PUBLISHED BOOK REVIEWS
Barry Wilkinson

October 19, 2008

BARRY WILKINSON AND MICHAEL ALLEN, *Parallel Programming: Techniques and Application Using Networked Workstations and Parallel Computers*, Prentice Hall Inc., 1st ed. 1999, 2nd ed. 2005.

D.1.3 **Concurrent Programming**

Distributed programming

See: 9903-0138 [D.1.5]

Parallel programming

See also: 9903-0131 [C.1.4—*Distributed architectures*]

WILKINSON, BARRY; AND ALLEN, MICHAEL 9903-0135
(Univ. of North Carolina, Charlotte)

Parallel programming: techniques and applications using networked workstations and parallel computers.

Prentice-Hall, Inc., Upper Saddle River, NJ, 1999, 431 pp., $59, ISBN 0-13-671710-1.

Teaching parallelism to undergraduates can be problematic, since access to real parallel machines is often impossible. By concentrating on networks of computers, such as are common in university computer laboratories, and on software specifically designed to distribute computations across a network, this book provides a real opportunity for students to experiment with parallelism.

This book is divided into two parts. In the first, basic concepts and algorithms are introduced, and in the second, a number of parallel algorithms, both numerical and nonnumerical, are analyzed. Competence in writing sequential programs is assumed, as is a basic understanding of data structures.

In chapter 1, the authors discuss the basic concepts of parallel programming, the terminology used, and the different types of parallel architectures. Chapter 2 examines the message-passing paradigm and introduces the parallel virtual machine (PVM) and message-passing interface (MPI) software tools. Computations that offer the possibility of massive parallelism are presented in chapter 3. Chapter 4 discusses divide-and-conquer and other basic parallelization strategies. The subject of chapter 5 is the pipelining of computations as a means of obtaining parallelism from sequential code. Chapter 6 examines synchronous computations, in which individual computations may need to wait for the results of other computations before continuing. In chapter 7, the authors discuss termination and load balancing. The final chapter of Part 1 is devoted to parallel programming on shared-memory systems.

Part 2 has only four chapters. Chapters 9 and 10 discuss sorting and numerical algorithms, including matrix multiplication and solutions to linear equations. Image processing is the subject of chapter 11. Chapter 12 covers searching and optimization, including branch-and-bound, genetic algorithms, and the hill-climbing technique.

At the end of each chapter, the authors include pointers to further reading, a bibliography, and several pages of exercises. There are four appendices. The first three provide the basic PVM, MPI, and Pthread routines that are needed for the exercises. The final appendix discusses the theoretical models of parallel computation. There is a comprehensive index, and an instructor's manual is available.

Although this book is aimed principally at undergraduate students, it would also be suitable for graduate students. It is well written and includes many code fragments and clear diagrams. The material is clearly and attractively presented, and the topics are chosen well. The problems are well designed and seem to be at the right level of difficulty. Online resources are provided, including extensive information about running parallel programs using the utilities discussed in the book. A complete lecture series based on the book is available on the Web.

My only concern is that the examples and code fragments are all in C or C++. This is for a variety of good reasons, not least that the parallel libraries PVM, MPI, and Pthread cannot easily be used with other languages. Many computer science students are no longer taught C or C++, however, and they may find some of the code difficult to follow.

This is a practical textbook, produced to a high standard. Although many textbooks have been written on parallel programming, to my knowledge none has taken a similar approach. I recommend this book strongly.

— *M. S. Joy,* Warwick, UK

BARRY WILKINSON, *Computer Architecture Design and Performance*, Prentice Hall Inc., 1st ed. 1991, 2nd ed. 1996.

C.1 PROCESSOR ARCHITECTURES
C.1.0 General

WILKINSON, BARRY (Univ. of North Carolina, Charlotte) 9706-0394

Computer architecture (2nd ed.): design and performance.

Prentice-Hall, Inc., Upper Saddle River, NJ, 1996, 463 pp., $62, ISBN 0-13-518200-X.

Parallelism at all levels of design as an architectural technique for enhancing computer performance is the focus of this book. The first part of the book deals with low-level opportunities for hardware concurrency at the instruction level and register transfer level; in cache memory organization and its mechanisms and algorithms; in memory management generally; and in pipelined processor design. Chapter 5, on the latter topic, and section 5.4, on superscalar processors, are especially good.

The second part of the book, chapters 6 through 8, turns to shared memory multiprocessor systems and deals with programming issues for MIMD multiprocessors, bus-based multiprocessors of various kinds and their related performance potentials, and, finally, interconnection networks in general.

The final third of the book, chapters 9 and 10, deals with message-based multiprocessor architectures and programming with examples such as hypercube derivative systems, workstation clusters, transputer-based designs, and the Occam programming language. Chapter 10 is devoted to the dataflow computational model and the architectural and programming language techniques based on it. On first reading, I thought the dataflow chapter completely out of place in a book of this generality and for its intended audience. I later reconsidered; the author seems to have been motivated to include this material as a sort of "looking beyond the current intellectual horizon" experience for his student readers as, for example, a contrast to the von Neumann architectural focus of the book up to this point. The author's pedagogical technique in doing so works well, regardless of the present or future importance or practicality of computer architectures based on the dataflow model. It will surely be a mind-stretcher for the typical student at this level.

This is an excellent textbook; the material is always well motivated and clearly presented, and the problems at the end of each chapter are excellent. One challenge I faced during my own experience of teaching computer architecture, before there were any actual textbooks on the subject, was finding graded problems that were not either descriptive and trivial or prescriptive and much too difficult for a learner in a first course. This book solves that problem for the instructor.

— *P. C. Patton,* Minneapolis, MN

PRENTICE HALL
BOOK REVIEW

Computing 13.6.90

EE - excellent

Book review

TOP OF THE CLASS

Computer Architecture Design and Performance
by Barry Wilkinson, published by Prentice Hall (01442) 881900, price £19.95

Reviewed by Chris Robbins, a freelance consultant

This is one of the best-written and most easily read books on the underlying concepts of computer design that I've encountered. When a book is this good, there's always the temptation to search out some fault, however small. But the only criticism I have is that the title is misleading, since it fails to correctly define the theme of the book, which concerns how a computer's architecture influences its performance, and, in particular, the use of parallelism to produce significant gains in computer performance. This is, however, adequately explained in the publisher's summary.

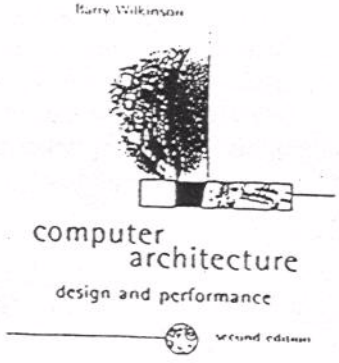
Starting with an excellent introduction to the 'conventional' Von Neumann stored-program computer, the author proceeds to give a brief historical outline and descriptions of basic concepts, such as addressing modes, microprocessor development and architecture, pipelining, memory hierarchies and reduced instruction set computing (Risc). All these topics and more are developed further in later chapters.

There is, for example, an excellent section focusing on the design of Risc instruction sets, including the associated microcode. This is followed by sections on other performance-influencing and enhancement techniques, such as cache memory, and memory management schemes, including paging and segmentation, pipelining and superscalar processors.

The second half of the book is devoted to parallel systems. In other words, multiprocessor designs and their interconnection schemes, including static and dynamic arrays and hypercubes, processing variants, dataflow and so on.

Since the target readership is undergraduate and graduate computer architecture students, each section concludes with a set of problems for readers to deal with, but no solutions are provided. The intention presumably being that these will be supplied by students' tutors.

Despite the ostensible readership, this book deserves to be read by a much wider audience — in fact, anyone with an interest in the subject of computer architecture and its development in the future.



PRENTICE HALL BOOK REVIEW

Computer Architecture — Design and Performance, 2nd Ed.: B. WILKINSON (Prentice-Hall, 1996, 463 pp., £19.95 paperback)

This book is divided into 3 parts. Part 1 deals with the internal design techniques of microprocessor circuits. Chapter 1 gives an overview of the development of computer architecture and, in particular, contrasts the RISC and CISC design approaches. Chapter 2 takes this theme further by developing an instruction set for a typical RISC processor and looks at how each instruction is executed, ending with the examination of a microprogrammed control-unit.

Chapters 3 and 4 cover memory-management issues. Chapter 3 deals with cache memory, examining the different types of cache memory organisation in detail and Chapter 4 considers paging the segmentation. Although reference is made to techniques applied by specific processor-families, the basic principles of these techniques are described rather than the details of specific implementations. Chapter 5 examines the use of pipelining and superscalar processors to speed up program execution.

Part 2 of the book covers shared-memory multiprocessor-systems. Chapter 6 explains Flynn's classification of multiprocessor systems and deals with array processors before going on to explain shared-memory multiple-instructions and multiple-data processors. Some of the software issues that arise with multiprocessor systems, such as mutual exclusion, are dealt with, but in a rather superficial way. Chapter 7 describes the use of various types of bus systems to interconnect processors and includes a discussion of cache coherence in such systems; such systems are usually only suitable for relatively small numbers of processors. Chapter 8 looks at networks suitable for connecting large numbers of processors including the hypercube configuration.

Finally, Part 3 explains multiprocessor systems which do not use shared memory, including message-passing multiprocessors, in Chapter 9, and data-flow techniques in Chapter 10. The transputer and occam, the language developed with the transputer, are described in some detail as perhaps the best example of the message-passing technique. However, a possible omission from the book is a coverage of the increasingly important area of digital signal processors and the approach to concurrency adopted by such processors as Texas Instruments' C40 and Analogue Devices' Sharc. With this exception, the book covers all the main topics relevant to modern computer-architecture. The explanations are clear and well illustrated and at about the right level of detail. Each chapter ends with a selection of interesting questions that could be used for tutorial sessions. Prentice-Hall will make a Solutions Manual available to lecturers who adopt the text.

Overall, this is an excellent book which could well provide the basis for final-year undergraduate or postgraduate courses and it can be thoroughly recommended.

G. F. BUTTERFIELD *Department of Electrical and Electronic Engineering, University of Hertfordshire*

INTERNATIONAL PRESS-CUTTING BUREAU
224-236 Walworth Road,
London SE17 1JE

Extract from
Times Higher Educational Supplement
– London.

27 SEP 1991

Constructing memory

BY **ALAN CLEMENTS**

COMPUTER ARCHITECTURE: DESIGN AND PERFORMANCE
BY BARRY WILKINSON
Prentice-Hall
354pp, £42.95 and £19.95
ISBN 0 13 173899 2 and 1739077
published January 1991

I TEACH COMPUTER architecture. It's a nice subject to teach for two reasons. First, it sounds good because it ties a modern high-tech plaything to an ancient and well-respected discipline, architecture. Second, no one actually knows what computer architecture is, so I get to teach what I want. However, over the past few years a consensus of what constitutes computer architecture has been emerging.

Three groups of people view a computer in radically different ways. Programmers and end-users regard the computer as a black box that executes their software. As for what goes on inside the black box, well frankly they don't give a damn. Computer technologists and electronic engineers are more concerned with how the computer is organised at the level of the logic element. They see the computer as a complex system of interconnected chips. The third group are the computer architects who are interested in what the computer does at the machine level. That is, they are concerned with the primitive operations that the computer carries out when it executes a program.

When authors write about computer architecture, they may veer towards the first group and concentrate on the way in which the computer implements high-level languages. Equally, they may veer towards the second group and emphasise how the computer is constructed in terms of its internal logic. Others writing on computer architecture take the middle path and describe the computer as seen by the assembly language programmer – that is, they cover the instruction set and addressing modes of one or more computers. Since you can take any of these approaches to computer architecture, the range of texts in this discipline is large.

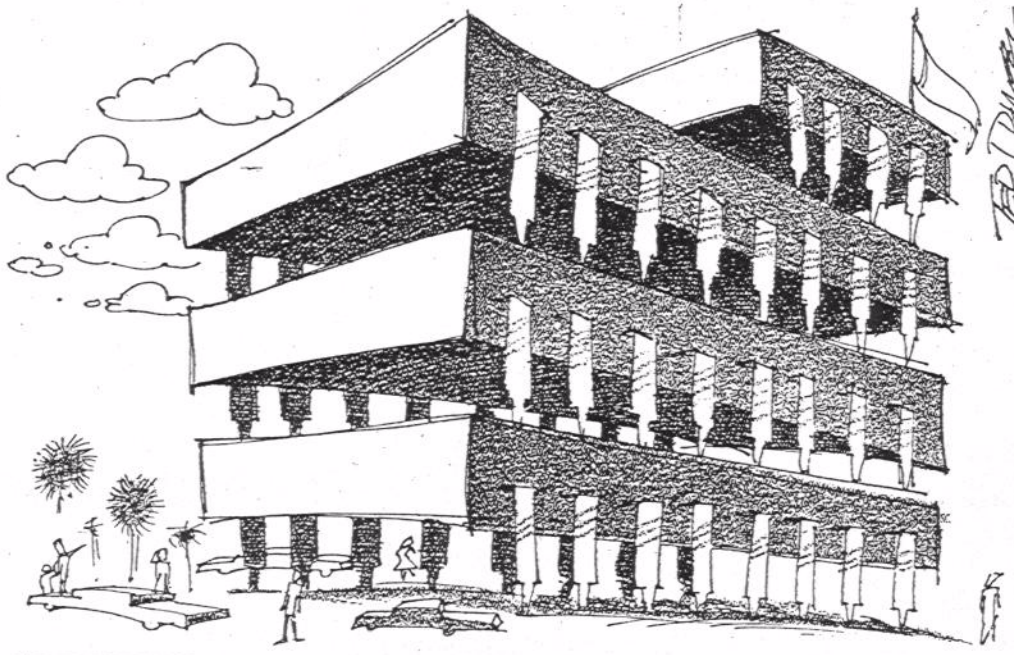
Barry Wilkinson's approach is biased towards the computer organisation/computer technology end of the spectrum, but the scope is widened to include many of the topics associated with computer systems rather than the central processing unit (CPU) itself. Of course, such a text is not intended for the beginner: it is more suited to the needs of the second-year or final-year degree student.

I found this text most useful because so many of its competitors focus on the CPU at the heart of a computer and devote much of their space to instruction sets and addressing modes. These works neglect the important systems that convert the CPU into a computer. Some of the topics Wilkinson has selected are: memory management and cache memories, pipelined systems and RISC processors, multiprocessors and their interconnection, and dataflow architectures. All these topics are of considerable interest today as they are closely associated with high-performance computers.

The section on memory systems includes coverage of: paging, address translation, multilevel paging, page-replacement algorithms, segmentation, and all aspects of cache memory – topics that are frequently either skipped over so lightly that the student is mystified, or dealt with at such length that the student is lost in the detail. Any student reading this text will be able to absorb all the fundamental principles and be in a position to tackle more specialised higher-level texts.

At least half of *Computer Architecture* is concerned with multiprocessing. Indeed, the book could probably have been retitled *An Introduction to Multiprocessing*. Some of the many topics covered are: pipelining, a survey of actual multiprocessor systems (for example, Illiac V, GF-11), parallel programming, buses and interconnection networks, and finally the non-von Neumann data flow machine. Mathematics is used in the description of multiprocessor systems and their performance, as you would expect at this level. However, the mathematics is not particularly difficult and should not present the typical student with any problems.

Apart from Wilkinson's selection of important topics, *Computer Architecture* makes a complex subject accessible to second-year or final-year students. Wilkinson writes in a clear and uncluttered style that can readily be appreciated by students; and he is to be congratulated for satisfactorily covering a large number of important topics in only 350 pages. I would strongly recommend this book for second-year or third-year courses in computer architecture, as it is both wide ranging and well written.



Alan Clements is reader in information engineering at Teesside Polytechnic.

PRENTICE HALL

INTERNATIONAL PRESS-CUTTING BUREAU
224-236 Walworth Road,
London SE17 1JE

Extract from
Computing – London

HARDWARE BOOKS

Computer Architecture — Design and Performance by Barry Wilkinson, published by Prentice Hall (0442) 231555, price £19.95.

If you want to remind yourself how virtual memory works, discover what a dataflow computer is, see how a transputer is put together or design a new chip, then this is definitely the book for you.

Furthermore, for such a dry-as-dust subject, it is remarkably readable and well written.

On the other hand you will need a background equivalent to two years of university computer studies to understand it fully. Computer architecture concerns the logical structure of hardware and will not help you understand chips at the physical level. I think you need a small understanding at this level, and this is why the background is needed. Conversely, a knowledge of high-level languages is also necessary — in order to follow such points as the occam examples.

The book meets its target as a text for third-year and graduate students very well, being comprehensive and accurate.

Coverage includes memory management, pipelining, Risc architectures, parallel and multiprocessor systems of all kinds. Special emphasis is placed on cache memory, transputers and dataflow architectures. Performance calculations are emphasised throughout, and comprehensive exercises provided.

A workmanlike, useful, entertaining book, on what I still regard as a dull subject, but then I'm not a systems programmer.

Reviewed by Ian Graham.

PRENTICE HALL

International Journal of
Electrical Engineering Education

ijeee

Volume 28 No. 4 - 1991

UKO cc. Allison King
Good.

Computer Architecture—design and performance: B. WILKINSON
(Prentice-Hall, 1991, 375pp., £19.95)

Barry Wilkinson is to be congratulated on his second-level text intended for final-year and graduate students of computer science. The book is divided into three parts. The first discusses the fundamental techniques to improve the performance of computer systems, the second looks at multi-processor systems with shared memory, while the final part considers multiprocessor systems not employing shared memory.

Consideration of the first section of the text indicates the overall methodology. The first chapter gives an excellent overview of computer systems from the earliest times while the second and third go on to consider memory management schemes. The work of both the pioneers and the current generation of innovators is acknowledged with appropriate references. Frequent reference is made to the products of current manufacturers comparing and contrasting relevant features. There is no hint of slavish reliance on manufacturers' catalogues.

A similar approach is adopted with the remainder of the book. Each chapter is followed by a set of exercises. The author is in the Department of Computer Science in the University of North Carolina at Charlotte. While the majority of his numerous references are to work in the USA, he is well aware of the European contribution since the earliest days. In the year of the bicentenary of the birth of Babbage it is good to see a generous reference to his work. Conrad Zuse too receives honourable mention for his pioneer work in Germany in the 1930s and 40s. The transputer, largely developed in the UK, receives careful attention as is appropriate in a text largely devoted to parallel processing. The book is recommended with enthusiasm.

M. G. HARTLEY, *Information Systems Engineering Group, UMIST*

Microprocessors & Microcomputers, May, 1991

An introduction to multiprocessing

Wilkinson, B
Computer architecture: design and performance Prentice Hall (1991) pp 375, ISBN 0-13-173899-2, 0-13-173907-7 pbk

The field of computer architecture and computer organization covers a multitude of topics, each of which can be taught at an introductory, an intermediate or an advanced level. Or, to put it another way, it is easy for a writer to 'pick and match' topics in this field to produce a unique text. Barry Wilkinson has selected his own set of topics from the rich palette of computer architecture. He has chosen to write at an intermediate level.

Many students taking a degree in computer science or computer engineering take at least one second-level course in computer architecture. Perhaps these students should no longer be taught a single module in computer architecture, as the subject has grown so much that it could easily encompass an entire degree scheme. A few years ago a course in computer architecture would have probably concentrated on the programming model of a computer and its implementation. Wilkinson assumes a basic knowledge of the central themes of computer architecture and concentrates on some of the areas that have become important in recent years. For example, this text includes very little on the design of instruction sets and addressing modes — a topic that lies at the heart of computer architecture.

The topics Wilkinson has selected are: memory management and cache memories, pipelined systems and RISC processors, multiprocessors and their interconnection, and dataflow architectures. All these topics are of considerable interest today.

Computer Architecture is pitched at the level of a second or third year degree student. Although its contents are not revolutionary in the sense that radically new material is introduced or a new insight into the subject is shown, this book makes a complex subject accessible to students. Wilkinson writes in a clear and uncluttered style that can readily be appreciated by students.

This section on memory systems covers topics including: paging, address translation, multilevel paging, page-replacement algorithms, segmentation, and all aspects of cache memory. These are topics that are frequently either skipped over so lightly that the student is mystified, or dealt with at such length that the student is lost in the detail. Any student reading this text will be able to absorb all the fundamental principles and be in a position to tackle more specialized higher level texts. Incidentally, Wilkinson provides a lot of embedded references to provide the reader with pointers to original source material and further reading.

At least half the book is concerned with multiprocessing. Indeed, it could probably have been retitled 'An introduction to multiprocessing'. Some of the many topics covered by Wilkinson are: pipelining, a survey of actual multiprocessor systems (e.g., Illiac V, GF-11), parallel programming, buses and interconnection networks, and finally the non von Neumann data flow machine. Mathematics is used in the description of multiprocessor systems and their performance, as you would expect at this level. However, the maths is not particularly difficult and should not present the typical student with any problems.

I would normally give a book like this a poor review, since it is impossible to treat the large number of topics covered by Wilkinson adequately. Inevitably, such a book serves little purpose since it confuses the beginner and fails to satisfy the more experienced reader. However, Wilkinson has achieved the impossible by satisfactorily covering a large number of important topics in only 370 pages. I would strongly recommend this for second or third year courses in computer architecture, as its contents are both wide ranging and well written.

Alan Clements
School of Computing, Teesside Polytechnic, Middlesbrough, UK

BARRY WILKINSON, *Digital System Design*, Prentice Hall Inc., 1987, 2nd ed. 1992.

PRENTICE HALL BOOK REVIEW

Zentralblatt für Mathematik und ihre Grenzgebiete –
Mathematics Abstracts ..... (B) Band 256/93

Chris Glenie

V.E.E. Good

Wilkinson, Barry; Makki, Rafic
Digital system design. 2nd ed.
Hemel Hempstead, Hearts.: Prentice Hall. (ISBN 0-13-220286-7). XIV, 538 p. (1992).

This textbook presents the fundamental topics in digital system designs. This is no book on mathematics or theoretical computer science, but on computer engineering. The fundamental mathematics is used to describe digital systems (hardware), to formalise the design process of them, and to study their behaviour and reliability.

The book is divided into three large parts. Part 1 is devoted to logic design, Part 2 is devoted to the components of a microprocessor system, and Part 3 is devoted to distinct aspects of digital system design. More precisely, Part 1 consists of Chapters 1 to 5. Chapter 1 presents the basic standard ways of the representation of numbers in digital systems. Chapter 2 introduces Boolean functions and some of the possibilities of their representation. The combinational logic circuits and sequential logic circuits resp. are defined in Chapter 3 and 4 respectively, where also the basic knowledge about them is given. Electronic fundamentals of circuits are presented in Chapter 5. Part 2 consists of Chapters 6 to 10. Chapter 6 provides the fundamentals of architectures of microprocessors including the concepts of stored programs and formats of instructions. Chapter 7 is devoted to basic instructions and Chapter 8 is devoted to memory devices. Chapter 10 shows how to manage the memory.

Part 3 consists of Chapters 11 to 15. Chapter 11 is a continuation of Chapter 4 on sequential circuits, and it discusses both synchronous and asynchronous circuit designs into more details. While Chapter 12 considers the design of a central processor. Chapter 13 is devoted to VLSI logic decision. Chapter 14 is devoted to the engineering aspects (transmission line reflections, cross-talk, etc.) of creating a working system. In Chapter 15, the reliability of a digital system is calculated, and methods to increase the reliability are discussed.

This textbook provides a very well written introduction into computer engineering covering all important, basic parts of this topic. All notions, tasks, and problems are carefully explained and exhaustively illustrated in a lot of examples. Some exercises are added to each Chapter of the book.

J. Hromkovič (Paderborn).

BARRY WILKINSON & DAVID HORROCKS, *Computer Peripherals*, Hodder and Stoughton, 1st ed. 1980, 2nd ed. 1987.

Computer Management, April, 1981

Computer Peripherals
by B. Wilkinson & D. Horrocks, Hodder & Stoughton, £12.95 (hb), £6.75 (pb)

This book is aimed at two types of readership the first being students studying HTC, HTD and BSC courses needing a supplementary book, and the second being personnel working with, or, interested in computers.

No specialist knowledge is required, but a knowledge of basic computing and simple electronics is desirable. All types of peripheral — some not even marketed yet are described — from the good old model 33 teletype and drum store to optical storage and bubble memory. In all cases physical and internal characteristics are described backed up with a large number of diagrams and drawings.

That the book fulfils its role for the students I do not doubt, and the adequate references at the end of each chapter should guide the student requiring further information.

As to the second readership category, the book is useful for looking up specific peripherals, as each chapter is free standing making the book a useful reference work. The exception was the chapter on the connection of peripherals to the computer which was rather more technical but nevertheless a relevant inclusion.

A useful addition to the DP library (especially at £6.75 in paper back) and all in all a good book — the authors just about succeeding in the difficult job of appealing to two classes of readership.

● **Circle No. 203** M. REES

Computer Management April 1981

in "The Principles of Computer Hardware" by A. Clements
Oxford University Press, 1985

Wilkinson, B., and Horrocks, D. (1982). *Computer peripherals*. Hodder and Stoughton. Devoted entirely to input and output devices, backing stores, and (in less detail) computer communications. All these topics are treated fully at an introductory level. This book is strongly recommended to those who wish to learn more about the structure, operation, and characteristics of peripherals.

Computing, April, 1981

BOOK REVIEWS

Clear and unbiased approach

Computer Peripherals by Barry Wilkinson and David Horrocks, published by Hodder and Stoughton, price £12.95.
reviewed by Tom Wearden

Anyone who wants to find out how computer peripherals work should read this well organised and thorough textbook.

In the preface the authors point out: 'Anyone coming into contact with computers, even if only superficially, cannot fail to become aware of the importance of peripherals.'

It is not unusual for the money spent on peripherals in new computer installations to be greater than that spent on processing data, so that an increasing number of people who are not necessarily 'experts', have to make critical decisions which may well involve several thousands of pounds.

Such people will find this book useful in explaining the fundamental workings of peripherals so that they can have informed discussions with sales people and their own specialists and technical advisers.

Almost every type of peripheral involves some mechanical movement. And a feature of this book is the large number of simple line diagrams which explain the mechanisms and operation of a wide variety of equipment in a way that cannot be equalled by words alone, but is adequately supplemented by the text.

The three-dimensional line drawings of needle print heads and various tape drive mechanisms are particularly effective.

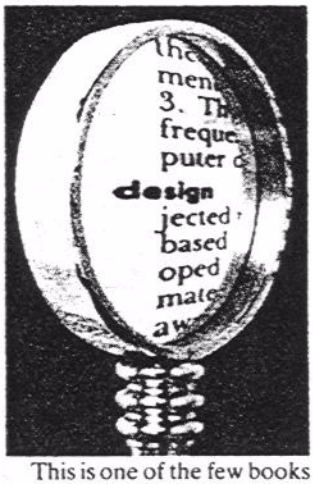
The opening chapter concentrates on the role of peripherals, and a later chapter deals with the ways in which information and data are passed between computers and peripherals. The main part of the book covers various peripheral devices, mechanisms and techniques under major headings, which include displays, printers, punched cards and tape, analogue-digital converters, graphic systems, data entry, and a wide ranging list of present and future backing stores.

The whole book is especially well organised so that it can easily be used as a reference work by a busy businessman who needs a quick insight into a specific problem, as well as a software specialist or system analyser who may sometimes see the electromechanical and engineering content of computer peripherals as an undesirable but necessary evil!

The authors are both lecturers at University College, Cardiff, and point out that 'this text should be of value to students studying engineering and computer technology'.

But this book deserves a far wider readership.

An 'outsider' approaching almost any aspect of computer work is immediately bewildered by the vast amount of written work that is on the market in various forms.



This is one of the few books which seems to recognise the problem, and point out that the 'first approach' to a wide range of techniques encountered in computer-peripherals is, in fact, surprisingly simple.

Computer Peripherals is recommended, especially, to those who have been rather baffled by the overpowering professionalism of so many 'experts'.

One omission in an otherwise excellent look at 'new' technologies, is reference to fibre optic technology.

Bubble domain memories and holographic stores are treated clearly, but there is no mention of the importance that optical fibres are likely to achieve in linking peripherals to computers.

It is to be hoped that these authors will produce further books in which their clear-thinking, unbiased approach will help the many people who need to have a fundamental understanding of computer-related subjects.

Tom Wearden is a freelance journalist.

The Times Higher Education Supplement, May 1st, 1981

Computer Science: an instructional manual
by C. S. French
D. P. Publications, £5.75 and £3.75
ISBN 0 905435 14 81 and 13 3

Fundamentals of Computer Science
by Andrew J. T. Colin
Macmillan, £14.00 and £6.95
ISBN 0 333 27552 7 and 30503 5

Computer Peripherals
by Barry Wilkinson and David Horrocks
Hodder & Stoughton, £12.95 and £6.75
ISBN 0 340 23649 3 and 32652 3

Without peripheral devices, however, no computation is possible. They already embrace a very wide range of electromechanical concepts; and new aspects of technology are being pressed into service at an alarming rate. However, books devoted to peripherals are few in number, a neglect which would seem surprising in view of the fact that they account for a good slice of the cost of computer installations.

Wilkinson and Horrocks have therefore provided a valuable contribution to the literature in their book entirely devoted to this topic—an excellent companion to Professor Colin's book. Professor Babbage himself would have delighted in the descriptions of the mechanical devices which had been just beyond his grasp throughout his life and marvelled at the electronic data-capture systems, especially those which are still at the limit of current technology.

After the first two chapters have set the scene with a comprehensive survey of computers and their organization (coupled with interface methods, standard interfaces, protocols and bus systems), the volume embarks on a systematic approach to peripherals. All the traditional digital devices, such as keyboards and VDUs, printers and punched-card devices, are discussed. Then comes comprehensive coverage of analog-signal input-output in which basic principles are followed by specific applications. The authors then move on to graphic systems and modern data-entry devices such as optical-mark-readers and optical-character-readers. Even voice input finds a place in the text; and the book concludes with a treatment of backing stores and data communication.

Michael Hartley

Michael Hartley is senior lecturer in the digital processes group of the department of electrical engineering and electronics at the University of Manchester Institute of Science and Technology.