Macmillan Computer Science Series

Consulting Editor: Professor F.H. Sumner, University of Manchester

A. Abdellatif, J. Le Bihan, M. Limame, *Oracle – A User's Guide*
S.T. Allworth and R.N. Zobel, *Introduction to Real-time Software Design, second edition*
Ian O. Angell, *High-resolution Computer Graphics Using C*
Ian O. Angell and Gareth Griffith, *High-resolution Computer Graphics Using FORTRAN 77*
Ian O. Angell and Gareth Griffith, *High-resolution Computer Graphics Using Pascal*
M. Azmoodeh, *Abstract Data Types and Algorithms, second edition*
C. Bamford and P. Curran, *Data Structures, Files and Databases*
Philip Barker, *Author Languages for CAL*
A.N. Barrett and A.L. Mackay, *Spatial Structure and the Microcomputer*
R.E. Berry, B.A.E. Meekings and M.D. Soren, *A Book on C, second edition*
P. Beynon-Davies, *Information Systems Development*
G.M. Birtwistle, *Discrete Event Modelling on Simula*
B.G. Blundell, C.N. Daskalakis, N.A.E. Heyes and T.P. Hopkins, *An Introductory Guide to Silvar Lisco and HILO Simulators*
B.G. Blundell and C.N. Daskalakis, *Using and Administering an Apollo Network*
T.B. Boffey, *Graph Theory in Operations Research*
Richard Bornat, *Understanding and Writing Compilers*
Linda E.M. Brackenbury, *Design of VLSI Systems – A Practical Introduction*
Alan Bradley, *Peripherals for Computer Systems*
G.R. Brookes and A.J. Stewart, *Introduction to occam 2 on the Transputer*
J.K. Buckle, *Software Configuration Management*
W.D. Burnham and A.R. Hall, *Prolog Programming and Applications*
P.C. Capon and P.J. Jinks, *Compiler Engineering Using Pascal*
J.C. Cluley, *Interfacing to Microprocessors*
J.C. Cluley, *Introduction to Low Level Programming for Microprocessors*
Robert Cole, *Computer Communications, second edition*
Derek Coleman, *A Structured Programming Approach to Data*
P. Davalo and E. Naim, *Neural Networks*
S.M. Deen, *Principles and Practice of Database Systems*
C. Delannoy, *Turbo Pascal Programming*
Tim Denvir, *Introduction to Discrete Mathematics for Software Engineering*
D. England et al., *A Sun User's Guide*
A.B. Fontaine and F. Barrand, *80286 and 80386 Microprocessors*
J.S. Florentin, *Microprogrammed Systems Design*
J.B. Gosling, *Design of Arithmetic Units for Digital Computers*
M.G. Hartley, M. Healey and P.G. Depledge, *Mini and Microcomputer Systems*
J.A. Hewitt and R.J. Frank, *Software Engineering in Modula-2 – An Object-oriented Approach*
Roger Hutty, *Z80 Assembly Language Programming for Students*
Roger Hutty, *COBOL 85 Programming*
Roland N. Ibbett and Nigel P. Topham, *Architecture of High Performance Computers, Volume I*
Roland N. Ibbett and Nigel P. Topham, *Architecture of High Performance Computers, Volume II*
Patrick Jaulent, *The 68000 – Hardware and Software*
P. Jaulent, L. Baticle and P. Pillot, *68020–30 Microprocessors and their Coprocessors*
J.M. King and J.P. Pardoe, *Program Design Using JSP – A Practical Introduction*

continued overleaf

E.V. Krishnamurthy, *Introductory Theory of Computer Science*
 V.P. Lane, *Security of Computer Based Information Systems*
 Graham Lee, *From Hardware to Software – An Introduction to Computers*
 A.M. Lister and R.D. Eager, *Fundamentals of Operating Systems, fourth edition*
 Elizabeth Lynch, *Understanding SQL*
 Tom Manns and Michael Coleman, *Software Quality Assurance*
 A. Mével and T. Guéguen, *Smalltalk-80*
 R.J. Mitchell, *Microcomputer Systems Using the STE Bus*
 R.J. Mitchell, *Modula-2 Applied*
 Y. Nishinuma and R. Espesser, *UNIX – First contact*
 Pim Oets, *MS-DOS and PC-DOS – A Practical Guide, second edition*
 Pham Thu Quang and C. Chartier-Kastler, *MERISE in Practice*
 A.J. Pilavakis, *UNIX Workshop*
 Christian Queinnec, *LISP*
 E.J. Redfern, *Introduction to Pascal for Computational Mathematics*
 Gordon Reece, *Microcomputer Modelling by Finite Differences*
 W.P. Salman, O. Tisserand and B. Toulout, *FORTH*
 L.E. Scales, *Introduction to Non-Linear Optimization*
 Peter S. Sell, *Expert Systems – A Practical Introduction*
 A.G. Sutcliffe, *Human-Computer Interface Design*
 Colin J. Theaker and Graham R. Brookes, *A Practical Course on Operating System*
 M.R. Tolhurst et al., *Open Systems Interconnection*
 A.J. Tyrrell, *COBOL from Pascal*
 M.J. Usher, *Information Theory for Information Technologists*
 B.S. Walker, *Understanding Microprocessors*
 Colin Walls, *Programming Dedicated Microprocessors*
 I.R. Wilson and A.M. Addyman, *A Practical Introduction to Pascal – with BS6192, second edition*

Non-series

Roy Anderson, *Management, Information Systems and Computers*
 I.O. Angell, *Advanced Graphics with the IBM Personal Computer*
 J.E. Bingham and G.W.P. Davies, *A Handbook of Systems Analysis, second edition*
 B.V. Cordingley and D. Chamund, *Advanced BASIC Scientific Subroutines*
 N. Frude, *A Guide to SPSS/PC+*
 Percy Mett, *Introduction to Computing*
 Tony Royce, *COBOL – An Introduction*
 Barry Thomas, *A PostScript Cookbook*

Microprogrammed Systems Design

J.S. Florentin

M
MACMILLAN

© J.S. Florentin, 1991

All rights reserved. No reproduction, copy or transmission of this publication may be made without written permission.

No paragraph of this publication may be reproduced, copied or transmitted save with written permission or in accordance with the provisions of the Copyright, Designs and Patents Act 1988, or under the terms of any licence permitting limited copying issued by the Copyright Licensing Agency, 33–4 Alfred Place, London WC1E 7DP.

Any person who does any unauthorised act in relation to this publication may be liable to criminal prosecution and civil claims for damages.

First published 1991

Published by
MACMILLAN EDUCATION LTD
Houndmills, Basingstoke, Hampshire RG21 2XS
and London
Companies and representatives
throughout the world

Typeset by
Ponting–Green Publishing Services, London

ISBN 978-0-333-54250-7 ISBN 978-1-349-21622-2 (eBook)
DOI 10.1007/978-1-349-21622-2

A catalogue record for this book is available from the British Library.

Contents

	<i>Preface</i>	<i>vii</i>
	<i>Acknowledgements</i>	<i>xi</i>
1	Introduction	1
	Necessary logic	2
	Gates	2
	Flip-flops	18
	Use of flip-flops	23
2	The First Computer	35
	Description of the First computer	35
	Implementation	41
	Data section	42
	Control section	56
	Practical points	66
3	The Bit-slice Chip	69
	Derivation of the bit-slice chip	69
	AMD Am2901 bit-slice chip	73
	Using the Am2901	80
	TI 74AS888 bit-slice chip	93
	Using the 74AS888	103
4	The Sequencer	105
	Derivation of sequencers	106
	Some real sequencer chips	130
	AMD Am2909 - outline of features	132
	Am2910	139
	Am2925 clock chip	151
	TI 74AS890 sequencer chip	153

5	Implementation of a Microprogrammed Computer	170
	Description of the Second computer	170
	Instruction set	171
	Implementation	177
	Data section	177
	Implementation of other features	196
6	Microprogramming	200
	Terminology	200
	Software tools	200
	The example	201
	Differences between assembler and microprogramming	211
7	Other Microprogrammable Devices	215
	Implementation of a hard disk controller	216
	Examples of microprogrammable controller chips	221
	A controller as a computer?	231
	RISC machines	234
8	VLSI Building Block Chips	239
	Building blocks	239
	Barrel shifters	243
	Funnel shifters	245
	Register file chips	246
	Arithmetic and logic units	246
	Combined register file/ALU chips	246
	Integer multipliers	249
	Floating point processors	254
	Examples of floating point processor chips	256
	Sequencers for use with building block chips	260
9	Conclusion	271
	<i>Bibliography</i>	274
	<i>Index</i>	275

Preface

To the man in the street a microprocessor is what is inside his home computer; however, the vast majority of microprocessors are not used in computers at all but are used to control devices like vending machines, lifts, security systems etc. These are all applications where once, if logic had been cheaper, the problem would have been solved using conventional logic. Microprocessors have essentially replaced this sort of logic.

The benefit of this is that it is possible to produce far more complex systems, far more easily, far more cheaply and in a shorter time. Whatever the system, a very limited number of different chips is needed - what makes these systems different are the programs in them. Furthermore, should the design need to be changed, usually all that has to be changed is the program, not the hardware.

What has really happened is that a system whose behaviour could be defined by means of a set of logic equations can now be defined by a program embedded in that system.

In many of these microprocessor systems the time between an input changing and causing a change in an output can be very long, without the user noticing. However, there are numerous types of logical problems where the speed of the system is of great importance. One such problem is the design of a microprocessor itself.

This book looks at the technique for designing digital systems which, if they were emulated on a microprocessor, would be too slow to be useful. At the same time, it shows how this technique must be suitable for designing systems of any degree of complexity.

This technique is called microprogramming. If we take an ordinary microprocessor we can program it in a high level language such as BASIC, we can also program it in assembler which is, in a sense, a lower level language. However, if we were able to look inside the chip itself we would usually find that each instruction, at the assembler level, is itself executed as a sequence of instructions at an even lower level. These lower level instructions are microinstructions. A sequence of microinstructions will form a microprogram.

Although the most obvious use of microprogramming is to construct

conventional computer systems, this technique can be used to construct almost any type of digital system.

In the same way that a microprocessor can be used to replace a logic design by a program, microprogramming can also be seen as replacing a logic design by a type of program.

This has similar advantages to those obtained when using a microprocessor as described above.

This book tries to cover the whole range of microprogrammable hardware currently available.

Wherever there is any risk of confusion, when a microinstruction is referred to in this book this will be done by using the term microinstruction; this is in contrast to conventional assembler level instructions that can be considered to be macro-instructions, which will usually be referred to as such. Similarly, microcode will be distinguished from macro-code and so on.

History

The concept of microprogramming was originated by Maurice Wilkes as long ago as 1951 in a paper called, "The best way to design an automatic calculating machine". Microprogramming only really became fashionable when the IBM 360 range appeared in the mid-sixties. All but the largest of these, when announced, were microprogrammed. Most computers and microprocessors made since then have used control logic that is, in principle, based upon Wilkes' original paper.

Readership

What would seem highly desirable would be a book that was suitable for a beginner, that would look at the material primarily in terms of the basic principles involved, that would provide a survey of the chips actually available and look at their properties in a comparative manner. It should also put these chips in context so that the reader can not only get them to work but should also realise whether they are the most appropriate technology to solve his problem. This is the author's aim.

The purpose of this material is not simply to show how to use the various chips but, because these techniques are used in the internal design of most microprocessors, to cover also their associated peripheral and controller chips, as well as almost all conventional computers and specialised processors, such as those used for signal processing.

Although many undergraduate courses may not include much of this material, many graduate electronics engineers who go on to work on digital hardware, apart from those who will only ever simply use microprocessors, will meet up with the chips and/or the techniques that the book describes.

Organisation

It is assumed that the reader has a good grasp of conventional logic. Some of this will nevertheless be described where certain concepts are absolutely crucial to the understanding of the subsequent material.

In order that the reader can appreciate what is happening, a simple computer rather like the Digital Equipment Corporation (DEC) PDP-8 computer will be constructed on paper from small scale integrated circuits. This is called the First computer. It will then be possible to analyse this computer in order to find what logic should be put on a chip to implement such a system, given that, as the technology advances, more and more logic can be put on a chip. This gives us two main types of chip, the sequencer chip and the bit-slice chip.

Unfortunately, having devised suitable chips and techniques for controlling them, the First computer then becomes too trivial to be interesting. At this point, a more complex computer, the Second computer, is used as the main example to show what can be done and how it can be done using these chips. This computer is based on the DEC PDP-11.

Usually, the simplest chips are covered first and other chips are then described, not in the same detail but sufficiently to explain any additional features. Two sets of chips are covered in detail; those from Advanced Micro Devices, AMD, and those from Texas Instruments, TI. Other chips are simply mentioned when they have particular features of interest.

A further example, a special processor for computing legal moves for a chess program, is used in chapter 6 to describe the software tools necessary for generating microcode.

Up to this point most of the emphasis has been placed on the design of conventional computer processors. At this point other types of micro-programmed systems are covered. The example introduced here is a hard-disk controller.

By now the basics have been covered, leaving a few other interesting cases and the Very Large Scale Integration (VLSI) building blocks. The example described at this point is a board that contains a flash floating-point chip, which is used to compute double-length floating-point sines.

Because there is more than enough material for any single course, the contents of this book have been organised so that, as far as possible, the material up to certain points is fairly self-contained. The book can thus be seen to consist of three parts. Part 1 covers the conventional logic and the implementation of the First computer and includes chapters 1 and 2. Part 2 presents an introduction to microprogrammable hardware and includes chapters 3, 4, 5 and 6 (excluding the coverage of the Texas Instruments chips). Part 3 covers more advanced microprogrammable hardware and the VLSI building blocks in chapters 7 and 8. This part also includes the Texas Instrument chips omitted from chapters 3 and 4.

Acknowledgements

The author would like to thank Advanced Micro Devices, Texas Instruments and Philips for permission to use diagrams from one or more of their data books and Mick Kelly for permission to use the microprogram written for his chess playing system. He would also like to thank Matthew Sillitoe for producing some of the drawings.

PDP-8, PDP-11 and DEC are trade marks of Digital Equipment Corporation.

Tri-state is a registered trade mark of National Semiconductors.