Macmillan Computer Science Series

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

S.T. Allworth and R.N. Zobel, *Introduction to Real-time Software Design, second edition*
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*
Ian O. Angell, *High-resolution Computer Graphics Using C*
M. Azmoodeh, *Abstract Data Types and Algorithms*
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 Silver 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 Bomat, *Understanding and Writing Compilers*
Linda E.M. Brackenbury, *Design of VLSI Systems – A Practical Introduction*
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*
Andrew J.T. Colin, *Fundamentals of Computer Science*
Andrew J.T. Colin, *Programming and Problem-solving in Algol 68*
S.M. Deen, *Fundamentals of Data Base Systems*
S.M. Deen, *Principles and Practice of Database Systems*
C. Delannoy, *Turbo Pascal Programming*
Tim Denvir, *Introduction to Discrete Mathematics for Software Engineering*
P.M. Dew and K.R. James, *Introduction to Numerical Computation in Pascal*
D. England et al., *A Sun User's Guide*
A.B. Fontaine and F. Barrand, *80286 and 80386 Microprocessors*
K.C.E. Gee, *Introduction to Local Area Computer Networks*
J.B. Gosling, *Design of Arithmetic Units for Digital Computers*
M.G. Hartley, M. Healey and P.G. Depledge, *Mini and Microcomputer Systems*
Roger Hutty, *Z80 Assembly Language Programming for Students*
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*
E.V. Krishnamurthy, *Introductory Theory of Computer Science*

continued overleaf

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
 Tom Manns and Michael Coleman, *Software Quality Assurance*
 Brian Meek, *Fortran, PL/I and the Algols*
 A. Mével and T. Guéguen, *Smalltalk-80*
 R.J. Mitchell, *Microcomputer Systems Using the STE Bus*
 Y. Nishinuma and R. Espesser, *UNIX – First Contact*
 Pim Oets, *MS-DOS and PC-DOS – A Practical Guide*, second edition
 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 Systems*
 M.R. Tolhurst et al., *Open Systems Interconnection*
 J-M. Trio, *8086–8088 Architecture and Programming*
 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, *Planning for Data Communications*
 B.V. Cordingley and D. Chamund, *Advanced BASIC Scientific Subroutines*
 N. Frude, *A Guide to SPSS/PC+*
 Barry Thomas, *A PostScript Cookbook*

Microcomputer Systems Using the STE Bus

R.J. Mitchell

*Department of Cybernetics
The University of Reading*

M
MACMILLAN

© R.J. Mitchell 1989

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 Act 1956 (as amended), 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 1989

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

Laserset by
Ponting–Green Publishing Services, London

British Library Cataloguing in Publication Data

Mitchell, R.J.
Microcomputer Systems: using the STE Bus.-
(Macmillan computer science series).
1. Microcomputer systems. Buses
I. Title
004.6

ISBN 978-0-333-49649-7 ISBN 978-1-349-10972-2 (eBook)
DOI 10.1007/978-1-349-10972-2

Contents

	<i>Preface</i>	<i>ix</i>
1	Introduction to bus systems	1
	1.1 Introduction	1
	1.2 Bus systems	2
	1.3 Data transfer protocols	7
	1.4 Arbitration	11
	1.5 Basic interfaces	13
	1.6 Bus systems with many processors	19
	1.7 Conclusion	21
2	Survey of microcomputer buses	22
	2.1 Processor independent bus systems	22
	2.2 S-100	23
	2.3 MUBUS	24
	2.4 CYBUS	25
	2.5 STD	26
	2.6 STE	27
	2.7 G64 and G96	28
	2.8 Multibus I	28
	2.9 VME	29
	2.10 Multibus II	30
	2.11 Futurebus	31
	2.12 The IEEE-488 instrument bus	32
	2.13 Conclusion	33
3	Interfacing slave devices	34
	3.1 Principles of interfacing simple devices	34
	3.2 STE bus signals	36
	3.3 Interfacing display and keypad	39
	3.4 Alternative design	43
	3.5 Software	45
	3.6 Interfacing many devices	46

vi Contents

3.7	Interface using a PLA	51
3.8	Interface to non-asynchronous buses	51
3.9	Interfacing EPROM	52
3.10	Interfacing static RAMs	55
3.11	Interfacing dynamic RAM	56
3.12	Analog-digital conversion	63
3.13	Attention requests	67
3.14	Multiple attention requests	72
3.15	Read-modify-write cycles	82
3.16	Burst transfer sequences	83
3.17	Conclusion	84
4	Other slave devices in STE systems	85
4.1	Serial communication	85
4.2	Parallel communication and Input/Output devices	95
4.3	Disks	97
4.4	Graphics	100
5	Bus masters on STE	108
5.1	Bus masters on STE	108
5.2	Transferring control between master devices	108
5.3	The design of microprocessor-based master devices	109
5.4	Design of direct memory access interfaces	148
5.5	'Intelligent' slaves	153
5.6	Conclusion	161
6	Software and testing	162
6.1	System development	162
6.2	Testing a slave module	164
6.3	Testing a master module	168
6.4	Other test equipment	169
6.5	Monitor program	170
6.6	Software development	171
6.7	Conclusion	174
7	Practical considerations	175
7.1	Logic elements	175
7.2	Power supplies	179
7.3	Buses	182
7.4	Interfacing to the 'real world'	186
7.5	Conclusion	190

Appendix 1	Asynchronous sequential logic design	191
A1.1	Sequential logic	191
A1.2	The design method	192
A1.3	The DATAACK* problem	196
Appendix 2	Synchronous sequential logic design	200
A2.1	Synchronous sequential logic design method	201
A2.2	PLA in keypad and displays circuit	205
A2.3	6809 interface circuit	207
A2.4	Dynamic RAM controller	209
A2.5	Automatic design	211
	<i>Bibliography</i>	<i>213</i>
	<i>Index</i>	<i>215</i>

Preface

Computers are tools that help in the solution of problems. To do this, they must be given an appropriate set of instructions. If the computer is to be used in a 'real world' application, suitable interfaces are required for converting between the signals inside the computer and those needed outside. For any particular application an appropriate set of interfaces is required.

An ideal configuration which allows the connection of the necessary interfaces is a *bus* system. Here the computer consists of a number of modules which intercommunicate along a series of wires called a *bus*. These modules include microprocessors, memories and peripheral devices. For the particular application, the required interface modules are plugged into the bus.

Unfortunately, there is no standardisation among the manufacturers of microprocessors and their related devices as to the signals required for their interconnection. Hence interfacing these devices to the bus can be problematic. To remedy this a number of bus standards have been proposed. One of the most modern of these standards is the STE bus, now approved by the IEEE as the IEEE 1000 bus.

There are many outstanding features of the STE bus which make it eminently suitable for configuring computer systems.

First, STE was designed by a group of engineers independently of the manufacturers of microprocessors. Therefore, the products of most manufacturers can be connected easily to the bus.

Second, the bus contains a simple yet powerful set of well defined signals which make it very easy to design interfaces to the bus. Consequently, it is an excellent bus for use as the basis of teaching the principles of interfaces.

STE uses the reliable, standard 64-way connectors. One such connector is used for each board, so the system is much cheaper than, say, VME.

Also, STE uses the asynchronous protocol, so it is easy for different speed devices to communicate, and the system automatically allows the use of faster devices as and when they are produced.

In addition, the bus is very flexible. Systems can be produced with

only one microprocessor, but multi-processor configurations are also possible. Thus STE can be used for parallel processing.

Also, the STE bus has IEEE approval. This means that the bus is defined formally and that manufacturers will adhere to the specification. Thus it should prove a very simple matter to interconnect the products from the various manufacturers.

Finally, many manufacturers offer products that connect to STE, so off-the-shelf modules are available with which to configure STE systems. In a period of 18 months STE has jumped from nowhere to being one of the most popular bus systems.

The aim of this book is to describe bus systems and interfacing to microprocessors in general and to describe the STE bus in particular. In the first chapter the principles of bus systems are introduced, showing many of the different schemes which can be used. In chapter 2 a survey is given of many of the bus systems that have been proposed. This shows the advantages of the STE bus. In the third chapter the interfacing of simple peripherals and memory devices to the bus is considered, and many practical circuits are given. Using the principles explained in this chapter, any other peripheral or memory can be connected to the bus. In chapter 4 some other peripherals are described showing how they can be used in systems, rather than giving detailed interfaces. In the fifth chapter circuits are provided showing how some microprocessors and other bus masters may be connected to the bus. This includes methods of handling multiprocessor systems. In chapter 6 the important topics of testing and programming are considered. Chapter 7 is devoted to the practical problems associated with computer systems in general and bus systems in particular. Technical requirements of the STE bus are also described. Finally, there are two appendices describing algorithms for designing sequential logic circuits.

This book is based on three courses given to second/third year students studying the various courses in Cybernetics at the University of Reading. It is assumed therefore that the reader knows about boolean logic and the operation of computers. Any reader without such knowledge is referred to the books listed in the bibliography. The subject matter covered by this short book is quite large, so it has not been possible to cover all the aspects in as great a detail as one might like. However, the book should give enough information to allow both the use of bus systems like STE and to allow readers to design and test their own modules. References are given where appropriate for the reader who requires more detailed information.

Inevitably, in a book about computers many jargon technical terms are used. Unfortunately, when such jargon is used continually it no longer appears as such (to the user), and so TLA*s will inevitably abound without

* Three Letter Abbreviation

the author being aware. It is hoped that all such occurrences are sufficiently explained for the reader.

The author wishes to thank the many people who have helped in the production of this book. These include various people in the Cybernetics Department – Professor Kevin Warwick and other colleagues; my fellow collaborators on *CYBUS*, Dr Andrew Findlay, Paul Minchinton and Chris Williams; Geoff Pearce and Shawn Frazer who helped produce some departmental STE bus systems; thanks also to Dick Zobel for his comments on the manuscript, to Malcolm Stewart for agreeing to the suggestion for the book, to family and friends for encouragement, and to many Cybernetics students, who as part of their projects tested some of the circuits given in this book, or who attended the lectures where this material was tried out.

Reading 1989