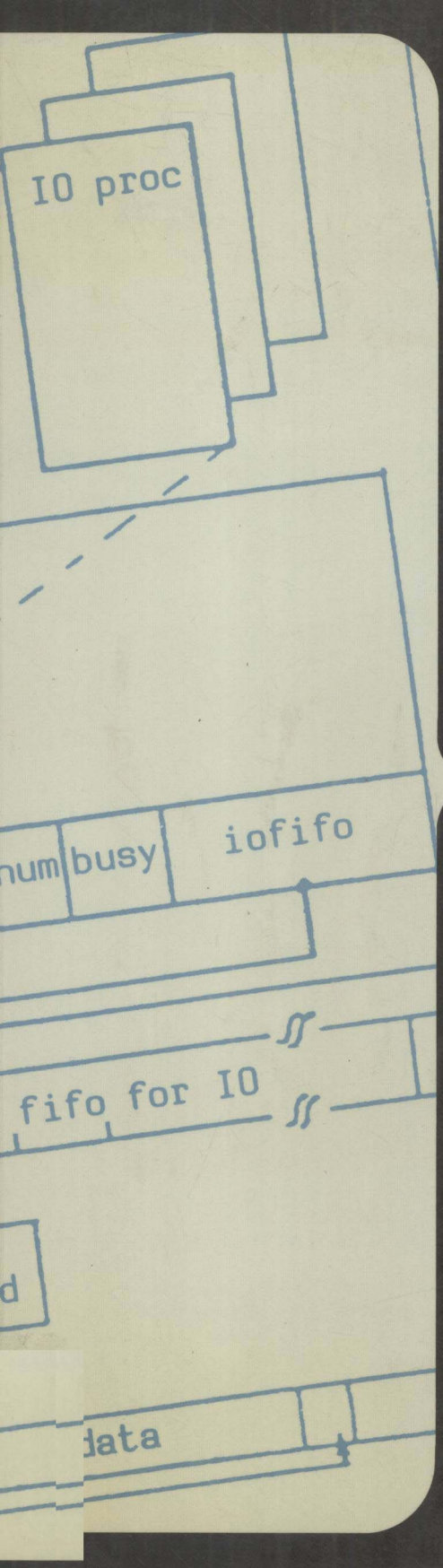

80286 and 80386 Microprocessors

New PC architectures

A.B. Fontaine and
F. Barrand

Translated by
A. Rawsthorne

80286 and 80386 Microprocessors

New PC architectures

A. B. Fontaine and F. Barrand

Translated by A. Rawsthorne
Department of Computer Science
University of Manchester

MACMILLAN

Authorised English language edition of *Les Microprocesseurs* 80286/80386 by A. B. Fontaine and F. Barrand.

First published 1989

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

Printed and bound in Great Britain at
The Camelot Press Ltd, Southampton

ISBN 0–333–49003–7

British Library Cataloguing in Publication Data
Fontaine, A. B.
80286 and 80386 microprocessors: new PC architectures. — (Macmillan computer science series)
1. INTEL 80286 & 80386 microprocessor systems
I. Title II. Barrand, F.
004.165

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*
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*
G. M. Birtwistle, *Discrete Event Modelling on Simula*
B. G. Blundell and C. N. Daskalakis, *Using and Administering an Apollo Network*
B. G. Blundell, C. N. Daskalakis, N. A. E. Heyes and T. P. Hopkins, *An Introductory Guide to Silvar Lisco and HILO Simulators*
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*
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*
M. R. M. Dunsmuir and G. J. Davies, *Programming the UNIX System*
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 Computers 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*

(*continued overleaf*)

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*
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/1 and the Algols*
A. Mével and T. Guéguen, *Smalltalk-80*
Y. Nishinuma and R. Espesser, *UNIX — First Contact*
Pim Oets, *MS-DOS and PC-DOS — A Practical Guide, second edition*
A. I. 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*
M. J. Usher, *Information Theory for Information Technologists*
B. S. Walker, *Understanding Microprocessors*
Peter J. L. Wallis, *Portable Programming*
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*

Acknowledgements

The translator wishes to thank Mr K Gallup, of Rapid Silicon, for making available facilities to compile the example programs in appendix C.

Contents

List of Figures

1 *Introduction*

The importance of sixteen-bit microprocessors has grown very significantly in the ten years since their introduction, and today only a small niche remains for the eight-bit processors. Apart from the Zilog Z80, which seems to go on and on, and excepting special-purpose sequencers and microcontrollers, the microcomputer market is dominated by 16- and 32-bit architectures. For a number of years now, the systems architect has had a number of suppliers at his disposal, and has been able to choose between numerous architectures; most current systems have, however, been based on one of two microcomputer families.

The two most significant families of microprocessor components in common usage today are the Intel 8086 and the Motorola 68000 architectures, although a number of less popular alternatives exist. The competition between these families is intense, and is based on the related bus and system architectures as well as on the capabilities of the semiconductor components themselves. Intel have developed the Multibus I and II architectures, while Motorola promote the VME bus, although it is possible to mix processors and bus architectures. Two significant influences on the adoption of these two families were the use of the Intel processor in the IBM PC and the appearance of the Motorola chip in the Apple Macintosh.

The financial rewards from a successful architecture are vast, since the investment in software for one architecture must be preserved, so semiconductor manufacturers need to hold on to their customers. One source inside Intel estimates that to date U.S. $6 billion have been spent on writing 8086 software!

In this book, we have chosen to describe in detail the iAPX 286, a successor to the 8086, with significant enhancements to help its users exploit its power. We also describe the significant enhancements brought to the Intel family by the introduction of the 32-bit iAPX 386.

The speed of progress in this area can be judged from a few figures: the 8086 processor, conceived in 1976 and introduced in 1978, has a raw performance of about 0.2 MIPS (million instructions per second) at 8 MHz. It can address 1 megabyte of memory. At the time of its introduction, this processor offered significant advantages over then-current 8-bit devices. Today, however, it has been largely superseded, for a number of reasons.

The first reason is the size of computer programs. Software sometimes seems

like an ideal gas filling all available memory. For example, the IBM PC, with a maximum of 640 kilobytes, is too small for a number of applications. This restriction has encouraged Intel and Lotus (of 1-2-3 spreadsheet fame) to develop a standard means of extending the 8086 address space to 8 megabytes, at the cost of some complexity in programs that need to manage this space explicitly.

The second reason is that of speed. Simple computer applications need only a limited computer power. Evolving applications tend to be much more sophisticated, with an optimised man-machine interface, with intensive use of graphics, in multiple windows, and extensive use of network communications. It is just not possible to provide these facilities on a less powerful machine. (The reader may test this assertion by comparing the usability of the MS-Windows system on a PC/XT and on a PC/AT.)

The third reason concerns the needs of protection. In order to use a multitasking or multi-user operating system, it is necessary to provide protection between processes. Protection facilities were introduced to the Intel family in the 80286 processor.

These considerations all played their part in Intel's development of the 80286, which also, of course, represents their attempt to keep a dominant position in the microprocessor market place.

The 80286 can be seen simply as an enhancement of the 8086. In real mode, or compatibility mode, it behaves as an 8086, but 4 or 5 times faster. It approaches a performance of 1 MIP, about the raw performance of a DEC VAX 11/780, for about $50. Used in this way, it offers a significant performance with no software modifications.

The two other problems mentioned above are addressed when the chip is used in protected mode, and need operating system modifications to exploit them. In this mode, the processor offers the same raw performance, but the real addressing capacity is increased to 16 megabytes, and an individual process can address a virtual space of 1 gigabyte. In this mode, the processor supports protection, which permits applications to run without corrupting others and, by catching erroneous accesses, eases debugging during program development. So far as compatibility is concerned, investment in existing 8086 software is preserved in that while applications needing to exploit new features will need rewriting, most current applications need very few, if any, changes to run on the 286. Even assembly-language programs can be ported to the 286 with minimal changes. Only an operating system that wishes to use the protected mode will need significant changes when ported from the 8086 to the 80286.

The story continues from here, since the 80286 now has a successor, the 80386, which retains the protected mode of the 80286, but extends its power: it can address 4 gigabytes of real memory, and has a raw performance of between 3 and 4 MIPS. A number of microcomputer manufacturers (including IBM) have introduced products based upon this processor: one of the first to be introduced was a portable computer by Compaq, offering a performance of about 3 MIPS, and

supporting 140 megabytes of disk, a streaming tape drive, and up to 6 megabytes of main memory. This machine showed a speed of about 3 times an IBM PC/AT, and was first offered in 1986 at a price of about U.S. $8000.

Software for the 80386 is also available now: the multitasking OS/2 is available, as are versions of Xenix and Unix System V. Some suppliers are offering concurrent UNIX and MS/DOS facilities, and one can predict the availability of "hypervisors" in the style of the VM/370 system for IBM mainframes.

1.1 8086 and 80286 Family Standard Architectures

The single most significant difference between the 8086 and the Motorola 68000 families is Intel's segmented memory addressing scheme. In order to address larger memory spaces, an address of 20 or more bits is necessary. If used everywhere, these addresses would lengthen the instructions in a program and slow its execution speed. Both architectures, therefore, allow the use of base registers, which can be loaded once with a full address, and thereafter permit data to be accessed using a short 8- or 16-bit displacement. The 68000 provides both of these and an absolute address form.

Intel, however, do not encourage the use of absolute addresses, and require the programmer to specify a Base Register and a displacement for every memory access. They call the base register a "segment register", and describe all memory accessible from that base as a "segment".

Segment sizes are limited, to 64 kilobytes in the case of the 8086 and the 80286 and to 4 megabytes in the 80386. This can be a significant constraint for an assembly-language programmer, but should be handled automatically by a high-level language compiler.

Segmentation can have advantages for the construction of large software applications, in that individual segments can be characterised by their access requirements: the 80286, for instance, monitors each access to a segment. If it is marked "read-only", no software can write to it: if it is not marked "executable" it is impossible to execute code from it. Forbidden accesses are trapped by the processor, enhancing the security of the software that is running.

A related topic to segmentation is providing virtual memory by "paging". This scheme allows an operating system to use memory and processor time efficiently by allocating memory in fixed-size "pages", which are usually between 1 and 4 kilobytes in size. Address translation hardware is needed to permit programs and their data to be loaded into non-contiguous real memory. Segments, however, are of variable size and may be much larger than an optimum page size. The choice of providing either segmentation or paging in an architecture is resolved in the 80386 by providing both.