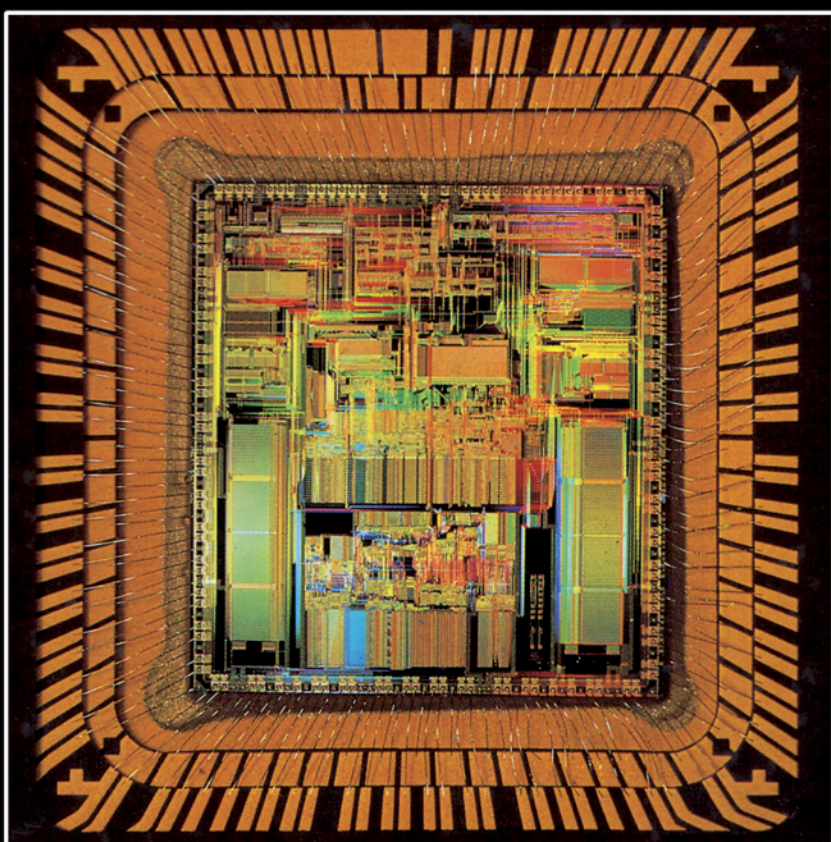

MICROPROCESSOR ARCHITECTURES AND SYSTEMS RISC, CISC & DSP



Steve Heath

*Microprocessor Architectures
and Systems*

This page intentionally left blank

*Microprocessor Architectures
and Systems*
RISC, CISC and DSP

STEVE HEATH

 NEWNES

Newnes

An imprint of Butterworth-Heinemann Ltd
Halley Court, Jordan Hill, Oxford OX2 8EJ



PART OF REED INTERNATIONAL BOOKS

OXFORD LONDON GUILDFORD BOSTON
MUNICH NEW DELHI SINGAPORE SYDNEY
TOKYO TORONTO WELLINGTON

First published 1991

© Steve Heath 1991

All rights reserved. No part of this publication may be reproduced in any material form (including photocopying or storing in any medium by electronic means and whether or not transiently or incidentally to some other use of this publication) without the written permission of the copyright holder except in accordance with the provisions of the Copyright, Designs and Patents Act 1988 or under the terms of a licence issued by the Copyright Licensing Agency Ltd, 33-34 Alfred Place, London, England WC1E 7DP. Applications for the copyright holder's written permission to reproduce any part of this publication should be addressed to the publishers

British Library Cataloguing in Publication Data

Heath, Steve

Microprocessor architectures and systems : RISC, CISC and DSP.

1. Microprocessor systems. Design

I. Title

004.256

ISBN 0 7506 0032 2

Typeset by Vision Typesetting, Manchester
Printed and bound in Great Britain

*For my Mother and Father
who bought me my
first soldering iron.*

This page intentionally left blank

Contents

<i>Preface</i>	xi
Acknowledgements	xiii
1 Complex instruction set computers	1
8-bit microprocessors: the precursors of CISC	1
8-bit microprocessor register models	3
Restrictions	4
Addressing memory	6
System integrity	7
Requirements for a new processor architecture	8
Software compatibility	9
Enter the MC68000	11
Complex instructions, microcode and nanocode	12
The MC68000 hardware	14
M68000 asynchronous bus	16
M6800 synchronous bus	19
Interrupts	19
Error recovery and control signals	20
Bus arbitration	22
Typical system	23
The register set	23
The USER/SUPERVISOR concept	25
Exceptions and the vector table	26
Addressing modes	28
Instruction set	29
Multitasking operating systems	30
High-level language support	34
The MC68010 virtual memory processor	36
MC68010 SUPERVISOR resource	40

Other improvements	41
The MC68008	42
The story continues	43
2 32-bit CISC processors	44
Enter HCMOS technology	44
Architectural challenges	45
The MC68020 32-bit performance standard	47
The programmer's model	49
Bus interfaces	52
Dynamic bus sizing	53
On-chip instruction cache	55
Debugging support	59
Coprocessor interface	60
MC68881 and MC68882 floating point coprocessors	63
The MC68851 paged memory management unit (PMMU)	64
The MC68030, the first commercial 50 MHz processor	66
3 The RISC challenge	72
The 80/20 rule	72
The initial RISC research	73
The M88000 family	76
The MC88100 programming model	83
The MC88100 instruction set	85
MC88100 external functions	88
MC88200 cache MMU	91
The MBUS protocol	94
4 Digital signal processing	98
Processor requirements	101
The DSP56000 family	102
The programming model	110
5 Memory, memory management and caches	119
Achieving processor throughput	119
Partitioning the system	121
Shadow RAM	122
DRAM v. SRAM	123
Memory management	127
Multitasking and user/supervisor conflicts	134
Cache size and organization	137
Cache coherency	143
Implementing memory systems	151
Conclusions	152

6	Real-time software, interrupts and exceptions	154
	What is real-time software?	154
	Responding to an interrupt	155
	Interrupting the processor	155
	Servicing the interrupt	155
	Locating associated tasks	156
	Context switches	157
	Improving performance	158
	Interrupting an MC88100	158
	MC88100 interrupt service routines	159
	Improving software performance	160
	Interrupting the DSP56000	163
	The M68300 family	169
	Conclusions	174
7	Multiprocessing	175
	SISD – Single instruction, single data	176
	SIMD – Single instruction, multiple data	176
	MIMD – Multiple instruction, multiple data	177
	MISD – Multiple instruction, single data	177
	Constructing a MIMD architecture	178
	Fault-tolerant systems	184
	Single- and multiple-threaded operating systems	187
8	Application ideas	189
1	MC68020 and MC68030 design techniques for high-reliability applications	189
2	Upgrading 8-bit systems	198
3	Transparent update techniques for digital filters using the DSP56000	202
4	Motor and servo control	205
9	Semiconductor technology	213
	Silicon technology	213
	CMOS and bipolar technology	215
	Fabrication technology	217
	Packaging	218
	Processor technology	221
	Memory technology	221
	Science fiction or not?	222
10	The changing design cycle	224
	The shortening design cycle	224
	The double-edged sword of technology	226

Make v. buy	226
Simulation v. emulation	232
11 The next generations	237
Enter the MC68040	237
The MC68300 family	246
Improving the instruction set	250
DSP96000 – combining integration and performance	256
12 Selecting a microprocessor architecture	259
Meeting performance needs	259
Software support	260
Development support	261
Standards	263
Built-in obsolescence	265
Market changes	265
Considering all the options	266
<i>Appendices</i>	267
<i>A Benchmarking</i>	267
<i>B Binary compatibility standards</i>	273
<i>Index</i>	280

Preface

‘Why are there all these different processor architectures and what do they all mean?’

‘Which processor shall I use and how should I choose it?’

There has been an unparalleled introduction of new processor architectures in recent years, which has widened the choice available for designs, but has also caused much confusion with the claims and counterclaims. This has resulted in questions concerning the need for several different processor types. The struggle for supremacy between Complex Instruction Set Computer architectures and those of Reduced Instruction Set Computer purveyors and the advent of powerful Digital Signal Processors has pulled the humble microprocessor into the realm of computer architectures, where the total system is the key to understanding and successful implementations. The days of separate hardware and software engineering are now numbered because of this close interaction between hardware and software. The effect of one decision now has definite repercussions with so many aspects throughout the design.

Given the task of selecting an architecture or design approach, both engineers and managers now require a knowledge of the whole system and an explanation of the design trade-offs and their effects. Such information rarely appears within data sheets or user manuals. This book fills that knowledge gap by closely examining the developments of Motorola’s CISC, RISC and DSP processors and describing the typical system configurations and engineering trade-offs that are made. The first part of the book provides a primer and history of the three basic microprocessor architectures. The second part describes the ways in which the architectures react with the system. Chapter 5 covers memory designs, memory management and cache memories. Chapter 6 examines interrupt and

exception handling and the effect on real-time applications. Chapter 7 examines basic multiprocessing ideas. Chapter 8 gives some application ideas which show how certain characteristics can be exploited. The third part of the book looks at some more commercial aspects: Chapter 9 covers semiconductor technology and what it will bring, while Chapter 10 examines the changing design cycle and its implications for the design process and commercial success. Chapter 11 looks at future processor generations and Chapter 12 describes the criteria that should be examined when selecting a processor. The appendices include further information on benchmarking and binary compatibility standards.

The examples have been based around Motorola's microprocessor families, but many of the system considerations are applicable to other processors with similar architectural characteristics. The material is based on several years of involvement with users who have gone through the decision-making process and who have frequently asked the questions at the beginning of this preface. To those of you that asked and inspired this book, I thank you.

The application note on high integrity MC68020 and MC68030 designs, the descriptions of the MC68040 processor and silicon technology in Chapters 8, 9 and 11 are based on articles that I have written for *Electronic Product Design*. Their permission to reprint the material is gratefully acknowledged.

In addition, I would like to say thank you to several of my colleagues at Motorola: to Pat McAndrew who has guided and educated me through DSP, to Ronnie Sutherland and Peter Crooks who have always been willing to discuss an architectural aspect over a beer, and to John Ralston, who has always tried to achieve the impossible whenever I have asked him! Special thanks must again go to Sue Carter for yet more editing, intelligent criticism and excellent coffee and for allowing me the use of our dining room to house my computer systems.

Steve Heath

Acknowledgements

The following trademarks mentioned within the text are acknowledged:

MC6800, MC6809, MC6801, MC68000, MC68020, MC68030, MC68040, MC68332, MC68302, MC68851, MC68881, MC68882, MC68008, MC68HC000, MC68HC001, DSP56000, DSP56001, DSP96000, MC88100, and MC88200 are all trademarks of Motorola, Inc.

UNIX is a trademark of AT&T

VRTX is a trademark of Ready Systems, Inc.

OS-9 is a trademark of Microware.

pDOS is a trademark of Eyring Research.

PDP-11, VAX 11/780 and DEC are trademarks of Digital Equipment Corporation.

iAPX8086, iAPX80286, iAPX80386 are trademarks of Intel Corporation.

While the information contained in this book has been carefully checked for accuracy, the author assumes no responsibility or liability for its use, or for any infringement of patents or other rights of third parties which would result.

As technical characteristics are subject to rapid change, the data contained are presented for guidance and education only. For exact details, consult the manufacturers' data and specifications.

This page intentionally left blank

1 *Complex instruction set computers*

8-bit microprocessors: the precursors of CISC

Ask what the definition of a CISC (Complex Instruction Set Computer) processor is and often the reply will be based around subjective comments like ‘a processor that has an over-complex and inefficient instruction set’, or even simply ‘an MC68000 or 8086’. The problem with these definitions is that they fail to account for the facts that more CISC microprocessors are used than any other architecture, they provide more direct hardware support for the software developer than any other architecture and are often more suitable for applications than either RISC (Reduced Instruction Set Computer) or DSP (Digital Signal Processor) alternatives. A CISC processor is better described as a mature design where software compatibility and help for software are the overriding goals. Many of today’s CISC processors owe their architectural philosophy to the early 8-bit microprocessors either as a foundation or as an example of how not to design a high-performance microprocessor. It is worthwhile reviewing these early designs to act as an introduction and a backdrop to further discussions.

A microprocessor can be described simply as a data processor: information is obtained, modified according to a predetermined set of instructions and either stored or passed to other elements within the system. Each instruction follows a simple set path of fetching an instruction, decoding and acting on it, fetching data from external memory as necessary and so on. In detail, such events consist of several stages based around a single or a succession of memory accesses. Figure 1.1 shows the basic functional blocks within a typical 8-bit microprocessor. The stages needed to execute an instruction to load an accumulator with a data byte are shown in italic type. Each stage is explained below.

Start of memory cycle 1

Stage 1 Here the current program counter address is used to fetch the