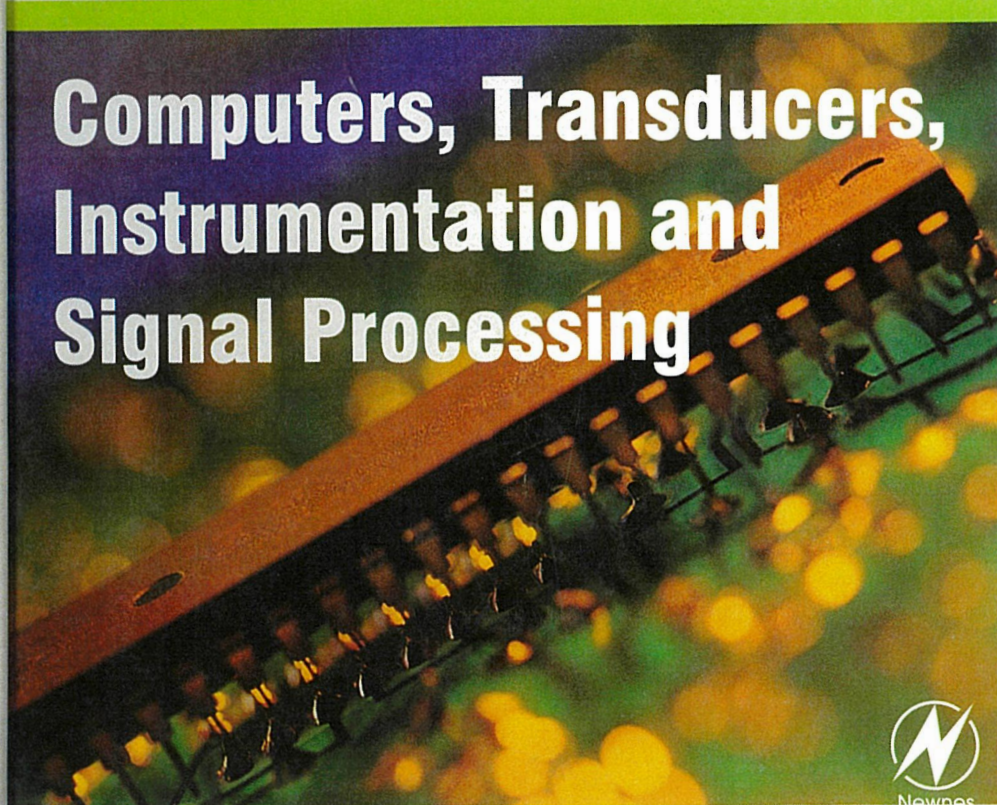


A. C. Fischer-Cripps

Newnes Interfacing Companion

**Computers, Transducers,
Instrumentation and
Signal Processing**



Newnes
An imprint of Elsevier Science
Linacre House, Jordan Hill, Oxford OX2 8DP
225 Wildwood Avenue, Woburn MA 01801-2041

First published 2002

Copyright © 2002, A. C. Fischer-Cripps. All rights reserved

The right of A. C. Fischer-Cripps to be identified as the author of this work
has been asserted in accordance with the Copyright, Designs and
Patents Act 1988

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, 90 Tottenham Court Road,
London, England W1T 4LP. Applications for the copyright holder's written
permission to reproduce any part of this publication should be addressed
to the publisher

British Library Cataloguing in Publication Data

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

Library of Congress Cataloguing in Publication Data

A catalogue record for this book is available from the Library of Congress

ISBN 0 750 65720 0

For information on all Newnes publications visit our website at www.newnespress.com

Printed and bound in Great Britain by Biddles Ltd. www.biddles.co.uk

Contents

Preface

Part 1: Transducers

1.1 Measurement systems

1.1.1 Transducers	4
1.1.2 Methods of measurement	5
1.1.3 Sensitivity	6
1.1.4 Zero, linearity and span	7
1.1.5 Resolution, hysteresis and error	8
1.1.6 Fourier analysis	9
1.1.7 Dynamic response	10
1.1.8 PID control	11
1.1.9 Accuracy and repeatability	12
1.1.10 Mechanical models	13
1.1.11 Review questions	14

1.2 Temperature

1.2.1 Temperature	16
1.2.2 Standard thermometers	17
1.2.3 Industrial thermometers	18
1.2.4 Platinum resistance thermometer	19
1.2.5 Liquid-in-glass thermometer	20
1.2.6 Radiation pyrometer	21
1.2.7 Thermocouple	22
1.2.8 Thermistors	24
1.2.9 Relative humidity	25
1.2.10 Review questions	26
1.2.11 Activities	28

1.3 Light

1.3.1 Light	35
1.3.2 Measuring light	36
1.3.3 Standards of measurement	37
1.3.4 Thermal detectors	38
1.3.5 Light dependent resistor	39
1.3.6 Photodiode	40
1.3.7 Other semiconductor photodetectors	41
1.3.8 Optical detectors	42
1.3.9 Photomultiplier	43
1.3.10 Review questions	44

1.4 Position and motion

1.4.1 Mechanical switch	46
-------------------------	----

1.4.2 Potentiometric sensor	47
1.4.3 Capacitive transducer	48
1.4.4 LVDT	49
1.4.5 Angular velocity transducer	50
1.4.6 Position sensitive diode array	51
1.4.7 Motion control	52
1.4.9 Review questions	53

1.5 Force, pressure and flow

1.5.1 Strain gauge	55
1.5.2 Force	57
1.5.3 Piezoelectric sensor instrumentation	58
1.5.4 Acceleration and vibration	59
1.5.5 Mass	60
1.5.6 Atmospheric pressure	61
1.5.7 Pressure	63
1.5.8 Industrial pressure measurement	64
1.5.9 Sound	65
1.5.10 Flow	66
1.5.11 Level	69
1.5.12 Review questions	70

Part 2: Interfacing

2.1 Number systems

2.1.1 Binary number system	74
2.1.2 Decimal to binary conversion	75
2.1.3 Hexadecimal	76
2.1.4 Decimal to hex conversion	77
2.1.5 2's complement	78
2.1.6 Signed numbers	79
2.1.7 Subtraction and multiplication	80
2.1.8 Binary coded decimal	81
2.1.9 Gray code	82
2.1.10 ASCII code	83
2.1.11 Boolean algebra	84
2.1.12 Digital logic circuits	85
2.1.13 Review questions	86
2.1.14 Activities	87

2.2 Computer architecture

2.2.1 Computer architecture	89
2.2.2 Memory	90
2.2.3 Segmented memory	91
2.2.4 Memory data	92

2.2.5 Buffers	93	2.4.11 Serial port operation	142
2.2.6 Latches	94	2.4.12 Parallel printer port	143
2.2.7 Flip-flop	95	2.4.13 Parallel port registers	144
2.2.8 Input/Output (I/O)	96	2.4.14 Parallel printer port operation	145
2.2.9 Microprocessor unit (MPU/CPU)	97	2.4.15 Review questions	146
2.2.10 Registers	98		
2.2.11 ROM	101	2.5 A to D and D to A conversions	
2.2.12 Interrupts	102	2.5.1 Interfacing	148
2.2.13 Memory map	104	2.5.2 The Nyquist criterion	149
2.2.14 Real and protected mode CPU operation	105	2.5.3 Resolution and quantisation noise	150
2.2.15 Review questions	107	2.5.4 Oversampling	151
2.2.16 Activities	108	2.5.5 Analog to digital converters	152
		2.5.6 ADC (integrating method)	153
2.3 Assembly language		2.5.7 ADC (successive approximation)	154
2.3.1 Instruction set	112	2.5.8 Aperture error	155
2.3.2 Assembly language	113	2.5.9 ADC08xx chip	156
2.3.3 Program execution	114	2.5.10 Sample-and-hold	157
2.3.4 Assembly language program structure	115	2.5.11 Sample-and-hold control	158
2.3.5 Assembler directives	116	2.5.12 Digital to analog conversion	159
2.3.6 Code segment	117	2.5.13 DAC0800	160
2.3.7 Assembly language shell program	118	2.5.14 Data acquisition board	161
2.3.8 Branching	119	2.5.15 Review questions	162
2.3.9 Register and immediate addressing	120		
2.3.10 Memory addressing	121	2.6 Data communications	
2.3.11 Indirect memory addressing	122	2.6.1 Communications	164
2.3.12 Indexed memory addressing	123	2.6.2 Byte to serial conversion	165
2.3.14 Interrupts	124	2.6.3 RS232 interface	166
2.3.15 Review questions	125	2.6.4 Synchronisation	167
2.3.16 Activities	126	2.6.5 UART (6402)	168
		2.6.7 Line drivers	170
2.4 Interfacing		2.6.8 UART clock	171
2.4.1 Interfacing	132	2.6.9 UART Master Reset	172
2.4.2 Input/Output ports	133	2.6.10 Null modem	173
2.4.3 Polling	134	2.6.11 Serial port BIOS services	174
2.4.4 Interrupts	135	2.6.12 Serial port operation in BASIC	175
2.4.5 Direct memory access (DMA)	136	2.6.13 Hardware handshaking	176
2.4.6 Serial port	137	2.6.14 RS485	177
2.4.7 Serial port addresses	138	2.6.15 GPIB	178
2.4.8 Serial port registers	139	2.6.16 USB	179
2.4.9 Serial port registers and interrupts	140	2.6.17 TCP/IP	181
2.4.10 Serial port baud rate	141	2.6.18 Review questions	182
		2.7 Programmable logic controllers	
		2.7.1 Programmable logic controllers	184

2.7.2 Timing	185	3.3 Instrumentation amplifier	
2.7.3 Functional components	186	3.3.1 Difference amplifier	247
2.7.4 Programming	187	3.3.2 CMRR	248
2.7.5 Ladder logic diagrams	188	3.3.3 Difference amplifier with	
2.7.6 PLC specifications	190	voltage follower inputs	249
2.7.7 Review questions	191	3.3.4 Difference amplifier with	
		cross-coupled inputs	250
2.8 Data acquisition project		3.3.5 CMRR cross coupled inputs	251
2.8.1 Serial data acquisition		3.3.6 Instrumentation amplifier	252
system	193	3.3.7 Log amplifier	253
2.8.2 Circuit construction	195	3.3.8 Op-amp frequency response	254
2.8.3 Programming	201	3.3.9 Review questions	255
2.8.4 Sample and hold	206	3.3.10 Activities	257
2.8.5 Digital to analog system	208		
Part 3: Signal processing		3.4 Noise	
3.1 Transfer function		3.4.1 Intrinsic noise	262
3.1.1 Instrumentation	214	3.4.2 Environmental noise	263
3.1.2 Transfer function	215	3.4.3 Signal-to-noise ratio	264
3.1.3 Transforms	216	3.4.4 Optical detectors	265
3.1.4 Laplace transform	217	3.4.5 Lock-in amplifier	266
3.1.5 Operator notation	218	3.4.6 Correlation	267
3.1.6 Differential operator	219	3.4.7 Review questions	268
3.1.7 Integrator – passive	220		
3.1.8 Differentiator – passive	221	3.5 Digital signal processing	
3.1.9 Transfer impedance	222	3.5.1 Digital filters	270
3.1.10 Review questions	223	3.5.2 Fourier series	271
3.1.11 Activities	224	3.5.3 Fourier transform	272
		3.5.4 Sampling	273
3.2 Active filters		3.5.5 Discrete Fourier transform	274
3.2.1 Filters	228	3.5.6 Filtering	275
3.2.2 T network filters	229	3.5.7 Digital filtering (ω -domain)	276
3.2.3 Twin-T filter	230	3.5.8 Convolution	277
3.2.4 Active integrator/		3.5.9 Discrete convolution	278
differentiator	231	3.5.10 Digital filtering (t-domain)	279
3.2.5 Integrator transfer function	232	3.5.11 Example	280
3.2.6 Low pass filter – active	233	3.5.12 Smoothing transfer function	281
3.2.7 2nd order active filter	234	3.5.13 Review questions	282
3.2.8 Double integrator	235	3.5.14 Activities	283
3.2.9 Bandpass filter – narrow	236		
3.2.10 Differentiator transfer		Index	285
function	237	Further reading	294
3.2.11 High pass filter – active	238	Parts list	295
3.2.12 High pass filter – ω domain	239		
3.2.13 Bandpass filter – wide	240		
3.2.14 Voltage gain and dB	241		
3.2.15 Review questions	242		
3.2.16 Activities	244		

An **interrupt vector table** contains the address pointer for the interrupt service routines associated with each of the 256 available interrupts. The interrupt vector table is usually located in low memory. Interrupt vectors 0 to 31 are usually reserved for microprocessor interrupts. The remainder can be used for hardware or software interrupts.

The interrupt **type number** determines its place within the interrupt vector table and its **priority** (with the exception of the NMI interrupt (2), but has the highest priority due to its direct connection with the CPU).

Microprocessor interrupts are divided into **fault**, **trap** or **abort** conditions. For fault conditions, the instruction that caused the fault is retried after the interrupt service routine has been executed. For trap conditions, the next instruction in the program being run by the CPU is executed after execution of the interrupt service routine. Abort conditions stop the main program execution entirely necessitating a restart of the program.

Vector	Interrupt
32–255	Available for software and hardware interrupts
17–31	Reserved
16	Coprocessor error
14–15	Reserved
13	General protection fault
12	Stack fault
10–11	Reserved
9	Hardware keyboard
8	Hardware timer
7	Coprocessor not available
6	Invalid opcode
5	Print screen
4	Overflow
3	Breakpoint
2	NMI
1	Debug/single step
0	Divide error

The **IF flag** is used to control whether or not **hardware interrupts** can be processed. When a hardware interrupt is recognised, the CPU clears the IF flag automatically, but this can be reset by the interrupt service routine if additional higher priority hardware interrupts are to be serviced during processing of the interrupt service routine.

The **non-maskable interrupt (NMI)** is a special hardware interrupt that is connected to the NMI pin of the CPU. The NMI is assigned an interrupt number of 2, although, since it cannot be masked by other interrupts, it effectively has the highest priority and is designed to be recognised in the shortest possible time. Conditions such as a power failure or memory read or write errors typically trigger this interrupt.

2.2.12 Interrupts

Servicing of I/O devices is usually done using **interrupts**. When an interrupt signal is received, the CPU suspends its activities and runs an **interrupt service routine** or **interrupt handler**. After the interrupt service routine has finished executing, normal execution is resumed.

There are three types of interrupts:

1. Microprocessor interrupts

These interrupts are initiated by various error conditions (such as a division by zero or arithmetic overflow). These interrupts are also known as **processor exceptions**.

2. Hardware interrupts

These interrupts are physically wired into the microcomputer. A special NMI interrupt has the highest priority and cannot be masked out by other interrupts. It is processed during critical hardware events such as a power loss.

3. Software interrupts

These interrupts are initiated by software to perform various operations such as writing to a disk file, reading from the serial port etc. These built-in interrupt handling routines are a part of the computer's **BIOS** – Basic Input/Output Services.

Interrupts are handled on a **priority** basis. The interrupt number determines its priority. High level interrupts cannot themselves be executed if a lower level or high priority interrupt is being processed.

The management of hardware interrupts is handled by a **programmable interrupt controller** chip: the 8259. This chip can be programmed to implement a variety of priority schemes and to accept level or edge-triggered interrupt signals. It determines which interrupt requires servicing and signals the CPU via the INTR line that an interrupt is pending. When an acknowledgement is received from the CPU, the 8259 places the interrupt number on the data bus and the CPU determines the address of the appropriate interrupt handler and the required interrupt service routine is then executed.

The 8259 controller can handle eight hardware devices. 8086-based microcomputers have one 8259 controller. 80286+ computers have two, with the second controller cascaded to interrupt channel 2 of the first giving access to 15 hardware devices.