THE ART OF ELECTRONICS

PAUL HOROWITZ
WINFIELD HILL

SECOND EDITION

Microchip Ex. 1069, p. 1
Microchip v. Aptiv
IPR2024-00230

THE ART OF ELECTRONICS

Second Edition

Microchip Ex. 1069, p. 2
Microchip v. Aptiv
IPR2024-00230

THE ART OF ELECTRONICS

Second Edition

Paul Horowitz HARVARD UNIVERSITY

Winfield Hill ROWLAND INSTITUTE FOR SCIENCE, CAMBRIDGE, MASSACHUSETTS

CAMBRIDGE
UNIVERSITY PRESS

Microchip Ex. 1069, p. 3
Microchip v. Aptiv
IPR2024-00230

Published by the Press Syndicate of the University of Cambridge
The Pitt Building, Trumpington Street, Cambridge CB2 1RP
40 West 20th Street, New York, NY 10011-4211, USA
10 Stamford Road, Oakleigh, Melbourne 3166, Australia

© Cambridge University Press 1980, 1989

First published 1980
Second edition 1989
Reprinted 1990 (twice), 1991, 1993, 1994 (twice), 1995, 1996

Printed in the United States of America

Library of Congress Cataloging-in-Publication Data is available.

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

ISBN 0-521-37095-7 hardback

Microchip Ex. 1069, p. 4
Microchip v. Aptiv
IPR2024-00230

TO CAROL, JACOB, MISHA, AND GINGER

CONTENTS

List of tables xvi

Preface xix

Preface to first edition xxi

CHAPTER 1
FOUNDATIONS 1

Introduction 1

Voltage, current, and resistance 2

1.01 Voltage and current 2
1.02 Relationship between voltage and current: resistors 4
1.03 Voltage dividers 8
1.04 Voltage and current sources 9
1.05 Thévenin's equivalent circuit 11
1.06 Small-signal resistance 13

Signals 15

1.07 Sinusoidal signals 15
1.08 Signal amplitudes and decibels 16
1.09 Other signals 17
1.10 Logic levels 19
1.11 Signal sources 19

Capacitors and ac circuits 20

1.12 Capacitors 20
1.13 *RC* circuits: *V* and *I* versus time 23
1.14 Differentiators 25
1.15 Integrators 26

Inductors and transformers 28

1.16 Inductors 28
1.17 Transformers 28

Impedance and reactance 29

1.18 Frequency analysis of reactive circuits 30
1.19 *RC* filters 35
1.20 Phasor diagrams 39
1.21 "Poles" and decibels per octave 40
1.22 Resonant circuits and active filters 41
1.23 Other capacitor applications 42
1.24 Thévenin's theorem generalized 44

Diodes and diode circuits 44

1.25 Diodes 44
1.26 Rectification 44
1.27 Power-supply filtering 45
1.28 Rectifier configurations for power supplies 46
1.29 Regulators 48
1.30 Circuit applications of diodes 48
1.31 Inductive loads and diode protection 52

Other passive components 53

1.32 Electromechanical devices 53
1.33 Indicators 57
1.34 Variable components 57
Additional exercises 58

CHAPTER 2
TRANSISTORS 61

Introduction 61

2.01 First transistor model: current amplifier 62

Some basic transistor circuits 63

2.02 Transistor switch 63
2.03 Emitter follower 65

Microchip Ex. 1069, p. 6
Microchip v. Aptiv
IPR2024-00230

2.04 Emitter followers as voltage regulators 68
2.05 Emitter follower biasing 69
2.06 Transistor current source 72
2.07 Common-emitter amplifier 76
2.08 Unity-gain phase splitter 77
2.09 Transconductance 78

Ebers-Moll model applied to basic transistor circuits 79

2.10 Improved transistor model: transconductance amplifier 79
2.11 The emitter follower revisited 81
2.12 The common-emitter amplifier revisited 82
2.13 Biasing the common-emitter amplifier 84
2.14 Current mirrors 88

Some amplifier building blocks 91

2.15 Push-pull output stages 91
2.16 Darlington connection 94
2.17 Bootstrapping 96
2.18 Differential amplifiers 98
2.19 Capacitance and Miller effect 102
2.20 Field-effect transistors 104

Some typical transistor circuits 104

2.21 Regulated power supply 104
2.22 Temperature controller 105
2.23 Simple logic with transistors and diodes 107

Self-explanatory circuits 107

2.24 Good circuits 107
2.25 Bad circuits 107
Additional exercises 107

CHAPTER 3
FIELD-EFFECT TRANSISTORS 113

Introduction 113

3.01 FET characteristics 114
3.02 FET types 117
3.03 Universal FET characteristics 119
3.04 FET drain characteristics 121
3.05 Manufacturing spread of FET characteristics 122

Basic FET circuits 124

3.06 JFET current sources 125
3.07 FET amplifiers 129
3.08 Source followers 133
3.09 FET gate current 135
3.10 FETs as variable resistors 138

FET switches 140

3.11 FET analog switches 141
3.12 Limitations of FET switches 144
3.13 Some FET analog switch examples 151
3.14 MOSFET logic and power switches 153
3.15 MOSFET handling precautions 169

Self-explanatory circuits 171

3.16 Circuit ideas 171
3.17 Bad circuits 171 vskip6pt

CHAPTER 4
FEEDBACK AND OPERATIONAL AMPLIFIERS 175

Introduction 175

4.01 Introduction to feedback 175
4.02 Operational amplifiers 176
4.03 The golden rules 177

Basic op-amp circuits 177

4.04 Inverting amplifier 177
4.05 Noninverting amplifier 178
4.06 Follower 179
4.07 Current sources 180
4.08 Basic cautions for op-amp circuits 182

An op-amp smorgasbord 183

4.09 Linear circuits 183
4.10 Nonlinear circuits 187

A detailed look at op-amp behavior 188

4.11 Departure from ideal op-amp performance 189
4.12 Effects of op-amp limitations on circuit behavior 193
4.13 Low-power and programmable op-amps 210

Microchip Ex. 1069, p. 7
Microchip v. Aptiv
IPR2024-00230

A detailed look at selected op-amp circuits 213

4.14 Logarithmic amplifier 213
4.15 Active peak detector 217
4.16 Sample-and-hold 220
4.17 Active clamp 221
4.18 Absolute-value circuit 221
4.19 Integrators 222
4.20 Differentiators 224

Op-amp operation with a single power supply 224

4.21 Biasing single-supply ac amplifiers 225
4.22 Single-supply op-amps 225

Comparators and Schmitt trigger 229

4.23 Comparators 229
4.24 Schmitt trigger 231

Feedback with finite-gain amplifiers 232

4.25 Gain equation 232
4.26 Effects of feedback on amplifier circuits 233
4.27 Two examples of transistor amplifiers with feedback 236

Some typical op-amp circuits 238

4.28 General-purpose lab amplifier 238
4.29 Voltage-controlled oscillator 240
4.30 JFET linear switch with R_{ON} compensation 241
4.31 TTL zero-crossing detector 242
4.32 Load-current-sensing circuit 242

Feedback amplifier frequency compensation 242

4.33 Gain and phase shift versus frequency 243
4.34 Amplifier compensation methods 245
4.35 Frequency response of the feedback network 247

Self-explanatory circuits 250

4.36 Circuit ideas 250
4.37 Bad circuits 250
Additional exercises 251

CHAPTER 5
ACTIVE FILTERS AND OSCILLATORS 263

Active filters 263

5.01 Frequency response with *RC* filters 263
5.02 Ideal performance with *LC* filters 265
5.03 Enter active filters: an overview 266
5.04 Key filter performance criteria 267
5.05 Filter types 268

Active filter circuits 272

5.06 VCVS circuits 273
5.07 VCVS filter design using our simplified table 274
5.08 State-variable filters 276
5.09 Twin-T notch filters 279
5.10 Gyrator filter realizations 281
5.11 Switched-capacitor filters 281

Oscillators 284

5.12 Introduction to oscillators 284
5.13 Relaxation oscillators 284
5.14 The classic timer chip: the 555 286
5.15 Voltage-controlled oscillators 291
5.16 Quadrature oscillators 291
5.17 Wien bridge and *LC* oscillators 296
5.18 *LC* oscillators 297
5.19 Quartz-crystal oscillators 300

Self-explanatory circuits 303

5.20 Circuit ideas 303
Additional exercises 303

CHAPTER 6
VOLTAGE REGULATORS AND POWER CIRCUITS 307

Basic regulator circuits with the classic 723 307

Microchip Ex. 1069, p. 8
Microchip v. Aptiv
IPR2024-00230

6.01 The 723 regulator 307
6.02 Positive regulator 309
6.03 High-current regulator 311

Heat and power design 312

6.04 Power transistors and heat sinking 312
6.05 Foldback current limiting 316
6.06 Overvoltage crowbars 317
6.07 Further considerations in high-current power-supply design 320
6.08 Programmable supplies 321
6.09 Power-supply circuit example 323
6.10 Other regulator ICs 325

The unregulated supply 325

6.11 ac line components 326
6.12 Transformer 328
6.13 dc components 329

Voltage references 331

6.14 Zener diodes 332
6.15 Bandgap (V_{BE}) reference 335

Three-terminal and four-terminal regulators 341

6.16 Three-terminal regulators 341
6.17 Three-terminal adjustable regulators 344
6.18 Additional comments about 3-terminal regulators 345
6.19 Switching regulators and dc-dc converters 355

Special-purpose power-supply circuits 368

6.20 High-voltage regulators 368
6.21 Low-noise, low-drift supplies 374
6.22 Micropower regulators 376
6.23 Flying-capacitor (charge pump) voltage converters 377
6.24 Constant-current supplies 379
6.25 Commercial power-supply modules 382

Self-explanatory circuits 384

6.26 Circuit ideas 384
6.27 Bad circuits 384
Additional exercises 384

CHAPTER 7
PRECISION CIRCUITS AND LOW-NOISE TECHNIQUES 391

Precision op-amp design techniques 391

7.01 Precision versus dynamic range 391
7.02 Error budget 392
7.03 Example circuit: precision amplifier with automatic null offset 392
7.04 A precision-design error budget 394
7.05 Component errors 395
7.06 Amplifier input errors 396
7.07 Amplifier output errors 403
7.08 Auto-zeroing (chopper-stabilized) amplifiers 415

Differential and instrumentation amplifiers 421

7.09 Differencing amplifier 421
7.10 Standard three-op-amp instrumentation amplifier 425

Amplifier noise 428

7.11 Origins and kinds of noise 430
7.12 Signal-to-noise ratio and noise figure 433
7.13 Transistor amplifier voltage and current noise 436
7.14 Low-noise design with transistors 438
7.15 FET noise 443
7.16 Selecting low-noise transistors 445
7.17 Noise in differential and feedback amplifiers 445

Noise measurements and noise sources 449

7.18 Measurement without a noise source 449
7.19 Measurement with noise source 450
7.20 Noise and signal sources 452
7.21 Bandwidth limiting and rms voltage measurement 453
7.22 Noise potpourri 454

Microchip Ex. 1069, p. 9
Microchip v. Aptiv
IPR2024-00230

Interference: shielding and grounding 455

7.23 Interference 455
7.24 Signal grounds 457
7.25 Grounding between instruments 457

Self-explanatory circuits 466

7.26 Circuit ideas 466
Additional exercises 466

CHAPTER 8
DIGITAL ELECTRONICS 471

Basic logic concepts 471

8.01 Digital versus analog 471
8.02 Logic states 472
8.03 Number codes 473
8.04 Gates and truth tables 478
8.05 Discrete circuits for gates 480
8.06 Gate circuit example 481
8.07 Assertion-level logic notation 482

TTL and CMOS 484

8.08 Catalog of common gates 484
8.09 IC gate circuits 485
8.10 TTL and CMOS characteristics 486
8.11 Three-state and open-collector devices 487

Combinational logic 490

8.12 Logic identities 491
8.13 Minimization and Karnaugh maps 492
8.14 Combinational functions available as ICs 493
8.15 Implementing arbitrary truth tables 500

Sequential logic 504

8.16 Devices with memory: flip-flops 504
8.17 Clocked flip-flops 507
8.18 Combining memory and gates: sequential logic 512
8.19 Synchronizer 515

Monostable multivibrators 517

8.20 One-shot characteristics 517
8.21 Monostable circuit example 519
8.22 Cautionary notes about monostables 519
8.23 Timing with counters 522

Sequential functions available as ICs 523

8.24 Latches and registers 523
8.25 Counters 524
8.26 Shift registers 525
8.27 Sequential PALs 527
8.28 Miscellaneous sequential functions 541

Some typical digital circuits 544

8.29 Modulo-*n* counter: a timing example 544
8.30 Multiplexed LED digital display 546
8.31 Sidereal telescope drive 548
8.32 An *n*-pulse generator 548

Logic pathology 551

8.33 dc problems 551
8.34 Switching problems 552
8.35 Congenital weaknesses of TTL and CMOS 554

Self-explanatory circuits 556

8.36 Circuit ideas 556
8.37 Bad circuits 556
Additional exercises 556

CHAPTER 9
DIGITAL MEETS ANALOG 565

CMOS and TTL logic interfacing 565

9.01 Logic family chronology 565
9.02 Input and output characteristics 570
9.03 Interfacing between logic families 572
9.04 Driving CMOS amd TTL inputs 575
9.05 Driving digital logic from comparators and op-amps 577

Microchip Ex. 1069, p. 10
Microchip v. Aptiv
IPR2024-00230

9.06 Some comments about logic inputs 579
9.07 Comparators 580
9.08 Driving external digital loads from CMOS and TTL 582
9.09 NMOS LSI interfacing 588
9.10 Opto-electronics 590

Digital signals and long wires 599

9.11 On-board interconnections 599
9.12 Intercard connections 601
9.13 Data buses 602
9.14 Driving cables 603

Analog/digital conversion 612

9.15 Introduction to A/D conversion 612
9.16 Digital-to-analog converters (DACs) 614
9.17 Time-domain (averaging) DACs 618
9.18 Multiplying DACs 619
9.19 Choosing a DAC 619
9.20 Analog-to-digital converters 621
9.21 Charge-balancing techniques 626
9.22 Some unusual A/D and D/A converters 630
9.23 Choosing an ADC 631

Some A/D conversion examples 636

9.24 16-Channel A/D data-acquisition system 636
9.25 $3\frac{1}{2}$-Digit voltmeter 638
9.26 Coulomb meter 640

Phase-locked loops 641

9.27 Introduction to phase-locked loops 641
9.28 PLL design 646
9.29 Design example: frequency multiplier 647
9.30 PLL capture and lock 651
9.31 Some PLL applications 652

Pseudo-random bit sequences and noise generation 655

9.32 Digital noise generation 655
9.33 Feedback shift register sequences 655
9.34 Analog noise generation from maximal-length sequences 658
9.35 Power spectrum of shift register sequences 658
9.36 Low-pass filtering 660
9.37 Wrap-up 661
9.38 Digital filters 664

Self-explanatory circuits 667

9.39 Circuit ideas 667
9.40 Bad circuits 668
Additional exercises 668

CHAPTER 10
MICROCOMPUTERS 673

Minicomputers, microcomputers, and microprocessors 673

10.01 Computer architecture 674

A computer instruction set 678

10.02 Assembly language and machine language 678
10.03 Simplified 8086/8 instruction set 679
10.04 A programming example 683

Bus signals and interfacing 684

10.05 Fundamental bus signals: data, address, strobe 684
10.06 Programmed I/O: data out 685
10.07 Programmed I/O: data in 689
10.08 Programmed I/O: status registers 690
10.09 Interrupts 693
10.10 Interrupt handling 695
10.11 Interrupts in general 697
10.12 Direct memory access 701
10.13 Summary of the IBM PC's bus signals 704
10.14 Synchronous versus asynchronous bus communication 707
10.15 Other microcomputer buses 708
10.16 Connecting peripherals to the computer 711

Microchip Ex. 1069, p. 11
Microchip v. Aptiv
IPR2024-00230

Software system concepts 714

10.17 Programming 714
10.18 Operating systems, files, and use of memory 716

Data communications concepts 719

10.19 Serial communication and ASCII 720
10.20 Parallel communication: Centronics, SCSI, IPI, GPIB (488) 730
10.21 Local area networks 734
10.22 Interface example: hardware data packing 736
10.23 Number formats 738

CHAPTER 11
MICROPROCESSORS 743

A detailed look at the 68008 744

11.01 Registers, memory, and I/O 744
11.02 Instruction set and addressing 745
11.03 Machine-language representation 750
11.04 Bus signals 753

A complete design example: analog signal averager 760

11.05 Circuit design 760
11.06 Programming: defining the task 774
11.07 Programming: details 777
11.08 Performance 796
11.09 Some afterthoughts 797

Microprocessor support chips 799

11.10 Medium-scale integration 800
11.11 Peripheral LSI chips 802
11.12 Memory 812
11.13 Other microprocessors 820
11.14 Emulators, development systems, logic analyzers, and evaluation boards 821

CHAPTER 12
ELECTRONIC CONSTRUCTION TECHNIQUES 827

Prototyping methods 827

12.01 Breadboards 827
12.02 PC prototyping boards 828
12.03 Wire-Wrap panels 828

Printed circuits 830

12.04 PC board fabrication 830
12.05 PC board design 835
12.06 Stuffing PC boards 838
12.07 Some further thoughts on PC boards 840
12.08 Advanced techniques 841

Instrument construction 852

12.09 Housing circuit boards in an instrument 852
12.10 Cabinets 854
12.11 Construction hints 855
12.12 Cooling 855
12.13 Some electrical hints 858
12.14 Where to get components 860

CHAPTER 13
HIGH-FREQUENCY AND HIGH-SPEED TECHNIQUES 863

High-frequency amplifiers 863

13.01 Transistor amplifiers at high frequencies: first look 863
13.02 High-frequency amplifiers: the ac model 864
13.03 A high-frequency calculation example 866
13.04 High-frequency amplifier configurations 868
13.05 A wideband design example 869
13.06 Some refinements to the ac model 872
13.07 The shunt-series pair 872
13.08 Modular amplifiers 873

Radiofrequency circuit elements 879

13.09 Transmission lines 879

Microchip Ex. 1069, p. 12
Microchip v. Aptiv
IPR2024-00230

13.10 Stubs, baluns, and transformers 881
13.11 Tuned amplifiers 882
13.12 Radiofrequency circuit elements 884
13.13 Measuring amplitude or power 888

Radiofrequency communications: AM 892

13.14 Some communications concepts 892
13.15 Amplitude modulation 894
13.16 Superheterodyne receiver 895

Advanced modulation methods 897

13.17 Single sideband 897
13.18 Frequency modulation 898
13.19 Frequency-shift keying 900
13.20 Pulse-modulation schemes 900

Radiofrequency circuit tricks 902

13.21 Special construction techniques 902
13.22 Exotic RF amplifiers and devices 903

High-speed switching 904

13.23 Transistor model and equations 905
13.24 Analog modeling tools 908

Some switching-speed examples 909

13.25 High-voltage driver 909
13.26 Open-collector bus driver 910
13.27 Example: photomultiplier preamp 911

Self-explanatory circuits 913

13.28 Circuit ideas 913
Additional exercises 913

CHAPTER 14
LOW-POWER DESIGN 917

Introduction 917

14.01 Low-power applications 918

Power sources 920

14.02 Battery types 920
14.03 Wall-plug-in units 931
14.04 Solar cells 932
14.05 Signal currents 933

Power switching and micropower regulators 938

14.06 Power switching 938
14.07 Micropower regulators 941
14.08 Ground reference 944
14.09 Micropower voltage references and temperature sensors 948

Linear micropower design techniques 948

14.10 Problems of micropower linear design 950
14.11 Discrete linear design example 950
14.12 Micropower operational amplifiers 951
14.13 Micropower comparators 965
14.14 Micropower timers and oscillators 965

Micropower digital design 969

14.15 CMOS families 969
14.16 Keeping CMOS low power 970
14.17 Micropower microprocessors and peripherals 974
14.18 Microprocessor design example: degree-day logger 978

Self-explanatory circuits 985

14.19 Circuit ideas 985

CHAPTER 15
MEASUREMENTS AND SIGNAL PROCESSING 987

Overview 987

Measurement transducers 988

15.01 Temperature 988
15.02 Light level 996
15.03 Strain and displacement 1001

Microchip Ex. 1069, p. 13
Microchip v. Aptiv
IPR2024-00230

15.04 Acceleration, pressure, force, velocity 1004
15.05 Magnetic field 1007
15.06 Vacuum gauges 1007
15.07 Particle detectors 1008
15.08 Biological and chemical voltage probes 1012

Precision standards and precision measurements 1016

15.09 Frequency standards 1016
15.10 Frequency, period, and time-interval measurements 1019
15.11 Voltage and resistance standards and measurements 1025

Bandwidth-narrowing techniques 1026

15.12 The problem of signal-to-noise ratio 1026
15.13 Signal averaging and multichannel averaging 1026
15.14 Making a signal periodic 1030
15.15 Lock-in detection 1031
15.16 Pulse-height analysis 1034
15.17 Time-to-amplitude converters 1035

Spectrum analysis and Fourier transforms 1035

15.18 Spectrum analyzers 1035
15.19 Off-line spectrum analysis 1038

Self-explanatory circuits 1038

15.20 Circuit ideas 1038

APPENDIXES 1043

Appendix A
The oscilloscope 1045

Appendix B
Math review 1050

Appendix C
The 5% resistor color code 1053

Appendix D
1% Precision resistors 1054

Appendix E
How to draw schematic diagrams 1056

Appendix F
Load lines 1059

Appendix G
Transistor saturation 1062

Appendix H
***LC* Butterworth filters 1064**

Appendix I
Electronics magazines and journals 1068

Appendix J
IC prefixes 1069

Appendix K
Data sheets 1072

2N4400-1 NPN transistor 1073
LF411-12 JFET operational amplifier 1078
LM317 3-terminal adjustable regulator 1086

Bibliography 1095

Index 1101

Microchip Ex. 1069, p. 14
Microchip v. Aptiv
IPR2024-00230

TABLES

1.1 Diodes 43
2.1 Small-signal transistors 109
3.1 JFETs 125
3.2 MOSFETs 126
3.3 Dual matched JFETs 128
3.4 Current regulator diodes 129
3.5 Power MOSFETs 164
3.6 BJT-MOSFET comparison 166
3.7 Electrostatic voltages 170
4.1 Operational amplifiers 196
4.2 Recommended op-amps 208
4.3 High-voltage op-amps 213
4.4 Power op-amps 214
5.1 Time-domain filter comparison 273
5.2 VCVS low-pass filters 274
5.3 555-type oscillators 289
5.4 Selected VCOs 293
6.1 Power transistors 314
6.2 Transient suppressors 326
6.3 Power-line filters 327
6.4 Rectifiers 331
6.5 Zener and reference diodes 334
6.6 500mW zeners 334
6.7 IC voltage references 336
6.8 Fixed voltage regulators 342
6.9 Adjustable voltage regulators 346
6.10 Dual-tracking regulators 352
7.1 Seven precision op-amps 401
7.2 Precision op-amps 404
7.3 High-speed precision op-amps 412
7.4 Fast buffers 418
7.5 Instrumentation amplifiers 429
8.1 4-bit integers 477
8.2 TTL and CMOS gates 484
8.3 Logic identities 491
8.4 Buffers 560
8.5 Transceivers 560
8.6 Decoders 561
8.7 Magnitude comparators 561
8.8 Monostable multivibrators 562
8.9 D-registers and latches 562
8.10 Counters 563
8.11 Shift registers 564
9.1 Logic family characteristics 570
9.2 Allowed connections between logic families 574
9.3 Comparators 584
9.4 D/A converters 620
9.5 A/D converters 632
9.6 Integrating A/D converters 634
10.1 IBM PC bus 704
10.2 Computer buses 709
10.3 ASCII codes 721
10.4 RS-232 signals 724
10.5 Serial data standards 727
10.6 Centronics (printer) signals 730
11.1 68000/8 instruction set 746
11.2 Allowable addressing modes 748
11.3 68000/8 addressing modes 749
11.4 68008 bus signals 753
11.5 68000/8 vectors 788
11.6 Zilog 8530 registers 804
11.7 Zilog 8530 serial port initialization 806
11.8 Microprocessors 822
12.1 PC graphic patterns 839
12.2 Venturi fans 858
13.1 RF transistors 877
13.2 Wideband op-amps 878
14.1 Primary batteries 922
14.2 Battery characteristics 923
14.3 Primary-battery attributes 930

Microchip Ex. 1069, p. 15
Microchip v. Aptiv
IPR2024-00230

14.4 Low-power regulators 942
14.5 Micropower voltage references 949
14.6 Micropower op-amps 956
14.7 Programmable op-amps 958
14.8 Low-power comparators 966
14.9 Microprocessor controllers 976
14.10 Temperature logger current drain 983
15.1 Thermocouples 990
D.1 Selected resistor types 1055
H.1 Butterworth low-pass filters 1064

Microchip Ex. 1069, p. 16
Microchip v. Aptiv
IPR2024-00230

1.03 Voltage dividers

We now come to the subject of the voltage divider, one of the most widespread electronic circuit fragments. Show us any real-life circuit and we'll show you half a dozen voltage dividers. To put it very simply, a voltage divider is a circuit that, given a certain voltage input, produces a predictable fraction of the input voltage as the output voltage. The simplest voltage divider is shown in Figure 1.5.

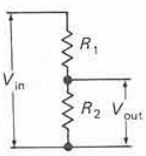


Figure 1.5. Voltage divider. An applied voltage V_{in} results in a (smaller) output voltage V_{out}.

What is V_{out}? Well, the current (same everywhere, assuming no "load" on the output) is

$$I = \frac{V_{in}}{R_1 + R_2}$$

(We've used the definition of resistance and the series law.) Then, for R_2,

$$V_{out} = IR_2 = \frac{R_2}{R_1 + R_2} V_{in}$$

Note that the output voltage is always less than (or equal to) the input voltage; that's why it's called a divider. You could get amplification (more output than input) if one of the resistances were negative. This isn't as crazy as it sounds; it is possible to make devices with negative "incremental" resistances (e.g., the tunnel diode) or even true negative resistances (e.g., the negative-impedance converter that we will talk about later in the book). However, these applications are rather specialized and need not concern you now.

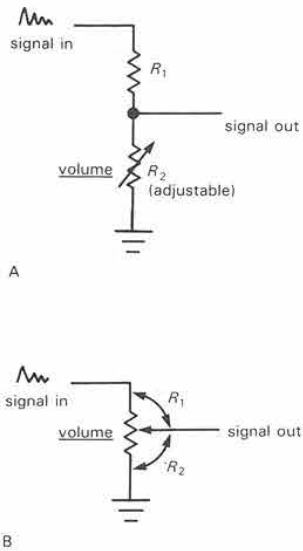


Figure 1.6. An adjustable voltage divider can be made from a fixed and variable resistor, or from a potentiometer.

Voltage dividers are often used in circuits to generate a particular voltage from a larger fixed (or varying) voltage. For instance, if V_{in} is a varying voltage and R_2 is an adjustable resistor (Fig. 1.6A), you have a "volume control"; more simply, the combination R_1R_2 can be made from a single variable resistor, or *potentiometer* (Fig. 1.6B). The humble voltage divider is even more useful, though, as a way of *thinking* about a circuit: the input voltage and upper resistance might represent the output of an amplifier, say, and the lower resistance might represent the input of the following stage. In this case the voltage-divider equation tells you how much signal gets to the input of that last stage. This will all become clearer after you know about a remarkable fact (Thévenin's theorem) that will be discussed later. First, though, a short aside on voltage sources and current sources.

Microchip Ex. 1069, p. 17
Microchip v. Aptiv
IPR2024-00230

EXERCISE 1.12
Derive the formula for the capacitance of two capacitors in series. Hint: Because there is no external connection to the point where the two capacitors are connected together, they must have equal stored charges.

The current that flows in a capacitor during charging ($I = CdV/dt$) has some unusual features. Unlike resistive current, it's not proportional to voltage, but rather to the rate of change (the "time derivative") of voltage. Furthermore, unlike the situation in a resistor, the power (V times I) associated with capacitive current is not turned into heat, but is stored as energy in the capacitor's internal electric field. You get all that energy back when you discharge the capacitor. We'll see another way to look at these curious properties when we talk about *reactance*, beginning in Section 1.18.

1.13 *RC* circuits: *V* and *I* versus time

When dealing with ac circuits (or, in general, any circuits that have changing voltages and currents), there are two possible approaches. You can talk about V and I versus time, or you can talk about amplitude versus signal frequency. Both approaches have their merits, and you find yourself switching back and forth according to which description is most convenient in each situation. We will begin our study of ac circuits in the time domain. Beginning with Section 1.18, we will tackle the frequency domain.

What are some of the features of circuits with capacitors? To answer this question, let's begin with the simple RC circuit (Fig. 1.29). Application of the capacitor rules gives

$$C\frac{dV}{dt} = I = -\frac{V}{R}$$

This is a differential equation, and its solution is

$$V = Ae^{-t/RC}$$

So a charged capacitor placed across a resistor will discharge as in Figure 1.30.

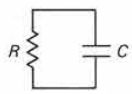


Figure 1.29

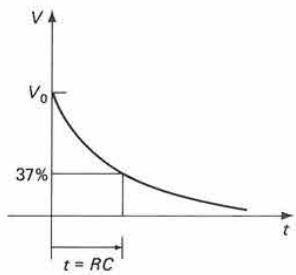


Figure 1.30. *RC* discharge waveform.

Time constant

The product RC is called the *time constant* of the circuit. For R in ohms and C in farads, the product RC is in seconds. A microfarad across 1.0k has a time constant of 1ms; if the capacitor is initially charged to 1.0 volt, the initial current is 1.0mA.

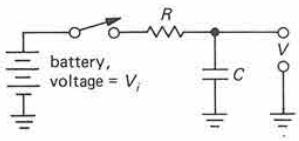


Figure 1.31

Figure 1.31 shows a slightly different circuit. At time $t = 0$, someone connects the battery. The equation for the circuit is then

$$I = C\frac{dV}{dt} = \frac{V_i - V}{R}$$

with the solution

$$V = V_i + Ae^{-t/RC}$$

Microchip Ex. 1069, p. 18
Microchip v. Aptiv
IPR2024-00230

(Please don't worry if you can't follow the mathematics. What we are doing is getting some important results, which you should remember. Later we will use the results often, with no further need for the mathematics used to derive them.) The constant A is determined by initial conditions (Fig. 1.32): $V = 0$ at $t = 0$; therefore, $A = -V_i$, and

$$V = V_i(1 - e^{-t/RC})$$

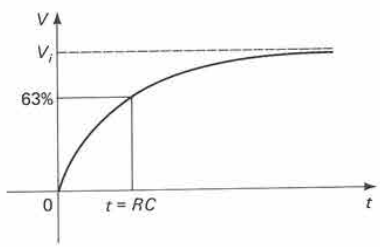


Figure 1.32

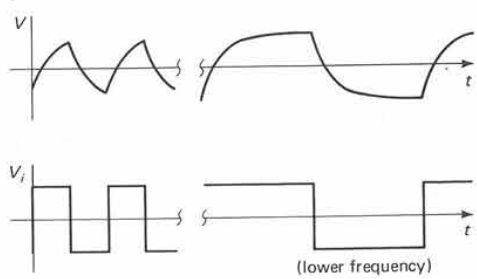


Figure 1.33. Output (top waveform) across a capacitor, when driven by square waves through a resistor.

Decay to equilibrium

Eventually (when $t \gg RC$), V reaches V_i. (Presenting the "$5RC$ rule of thumb": a capacitor charges or decays to within 1% of its final value in 5 time constants.) If we then change V_0 to some other value (say, 0), V will decay toward that new value with an exponential $e^{-t/RC}$. For example, a square-wave input for V_0 will produce the output shown in Figure 1.33.

EXERCISE 1.13

Show that the rise time (the time required to go from 10% to 90% of its final value) of this signal is $2.2RC$.

You might ask the obvious next question: What about $V(t)$ for arbitrary $V_i(t)$? The solution involves an inhomogeneous differential equation and can be solved by standard methods (which are, however, beyond the scope of this book). You would find

$$V(t) = \frac{1}{RC}\int_{-\infty}^{t} V_i(\tau)e^{-(t-\tau)/RC}d\tau$$

That is, the RC circuit averages past history at the input with a weighting factor

$$e^{-\Delta t/RC}$$

In practice, you seldom ask this question. Instead, you deal in the *frequency domain* and ask how much of each frequency component present in the input gets through. We will get to this important topic soon (Section 1.18). Before we do, though, there are a few other interesting circuits we can analyze simply with this time-domain approach.

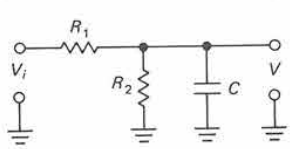


Figure 1.34

Simplification by Thévenin equivalents

We could go ahead and analyze more complicated circuits by similar methods, writing down the differential equations and trying to find solutions. For most purposes it simply isn't worth it. This is as complicated an RC circuit as we will need. Many other circuits can be reduced to it (e.g., Fig. 1.34). By just using the Thévenin equivalent of the voltage divider formed by R_1 and R_2, you can find the output

Microchip Ex. 1069, p. 19
Microchip v. Aptiv
IPR2024-00230

$V(t)$ produced by a step input for V_0.

EXERCISE 1.14

$R_1 = R_2 = 10\text{k}$, and $C = 0.1\mu\text{F}$ in the circuit shown in Figure 1.34. Find $V(t)$ and sketch it.

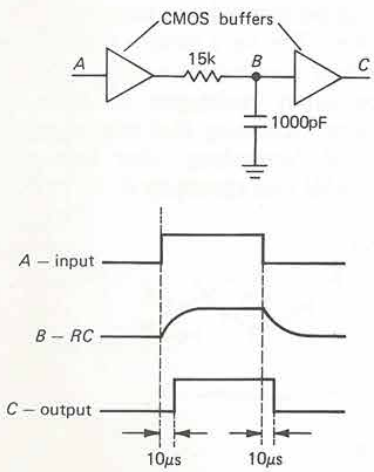


Figure 1.35. Producing a delayed digital waveform with the help of an RC.

Example: time-delay circuit

We have already mentioned logic levels, the voltages that digital circuits live on. Figure 1.35 shows an application of capacitors to produce a delayed pulse. The triangular symbols are "CMOS buffers." They give a HIGH output if the input is HIGH (more than one-half the dc power-supply voltage used to power them), and vice versa. The first buffer provides a replica of the input signal, but with low source resistance, and prevents input loading by the RC (recall our earlier discussion of circuit loading in Section 1.05). The RC output has the characteristic decays and causes the output buffer to switch $10\mu s$ after the input transitions (an RC reaches 50% output in $0.7RC$). In an actual application you would have to consider the effect of the buffer input threshold deviating from one-half the supply voltage, which would alter the delay and change the output pulse width. Such a circuit is sometimes used to delay a pulse so that something else can happen first. In designing circuits you try not to rely on tricks like this, but they're occasionally handy.

1.14 Differentiators

Look at the circuit in Figure 1.36. The voltage across C is $V_{\text{in}} - V$, so

$$I = C\frac{d}{dt}(V_{\text{in}} - V) = \frac{V}{R}$$

If we choose R and C small enough so that $dV/dt \ll dV_{\text{in}}/dt$, then

$$C\frac{dV_{\text{in}}}{dt} \approx \frac{V}{R}$$

or

$$V(t) = RC\frac{d}{dt}V_{\text{in}}(t)$$

That is, we get an output proportional to the rate of change of the input waveform.

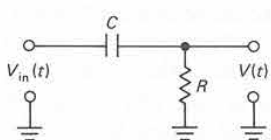


Figure 1.36

To keep $dV/dt \ll dV_{\text{in}}/dt$, we make the product RC small, taking care not to "load" the input by making R too small (at the transition the change in voltage across the capacitor is zero, so R is the load seen by the input). We will have a better criterion for this when we look at things in the frequency domain. If you drive this circuit with a square wave, the output will be as shown in Figure 1.37.

Microchip Ex. 1069, p. 20
Microchip v. Aptiv
IPR2024-00230

which might be useful if the complements A', B', and C' already exist somewhere in the circuit.

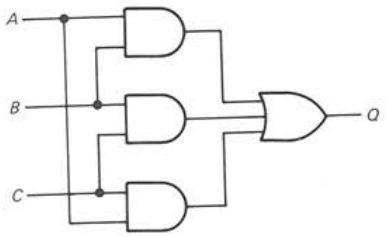


Figure 8.28

Some comments on Karnaugh maps

1. Look for groups of 2, 4, 8, etc., squares; they have the simple logic expressions.
2. The larger the block you describe, the simpler the logic.
3. The edges of the Karnaugh map connect up. For instance, the map in Figure 8.29 is described by $Q = B'C$.

CD \ AB	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	0	0	1
10	1	0	0	1

Figure 8.29

4. A block of 1's with only one or two 0's may be best described by the grouping illustrated in Figure 8.30, which corresponds to the logic expression

$$Q = A(BCD)'$$

5. Xs (don't care) are "wild cards." Use them as 1's or 0's to generate the simplest logic.

CD \ AB	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	0	0	1
10	0	0	1	1

Figure 8.30

6. A Karnaugh map may not lead directly to the best solution: A more complicated logic expression may sometimes have a simpler realization in gates, e.g., if some of its terms already exist as logic in your circuit, and you can exploit intermediate outputs (from other terms) as inputs. Furthermore, exclusive-OR realizations are not always obvious from Karnaugh maps. Finally, package constraints (e.g., the fact that four 2-input gates come in a single IC) also figure into the choice of logic used in the final circuit realization. When programmable logic devices such as PALs (Section 8.15) are used to construct logic functions, the internal structure (programmable AND; fixed OR) constrains the realization that can be used.

EXERCISE 8.13

Draw a Karnaugh map for logic to determine if a 3-bit integer (0 to 7) is prime (assume that 0,1, and 2 are not primes). Show a realization with 2-input gates.

EXERCISE 8.14

Find logic to perform multiplication of two 2-bit unsigned numbers (i.e., each 0 to 3), producing a 4-bit result. Hint: Use a separate Karnaugh map for each output bit.

8.14 Combinational functions available as ICs

Using Karnaugh maps, you can construct logic to perform rather complicated functions such as binary addition or magnitude comparison, parity checking, multiplexing

Microchip Ex. 1069, p. 21
Microchip v. Aptiv
IPR2024-00230

(selecting one of several inputs, as determined by a binary address), etc. In the real world the most frequently used complex functions are available as single MSI functions (medium-scale integration, upward of 100 gates on one chip). Although many of the MSI functions involve flip-flops, which we will get to shortly, lots of them are combinational functions involving only gates. Let's see what animals live in the MSI combinational zoo.

Quad 2-input select

The quad 2-input select is a very useful chip. It is basically a 4-pole 2-position switch for logic signals. Figure 8.31 shows the basic idea. When SELECT is LOW, the A inputs are passed through to their respective Q outputs. For SELECT HIGH, the B inputs appear at the output. ENABLE′ HIGH disables the device by forcing all outputs LOW. This is an important concept you will see more of later. Here's the truth table, which illustrates the X (don't care) entry:

Inputs				*Outputs*
E'	SEL	A_n	B_n	Q_n
H	X	X	X	L
L	L	L	X	L
L	L	H	X	H
L	H	X	L	L
L	H	X	H	H

Figure 8.31 and the preceding table correspond to the '157 quad 2-input select chip. The same function is also available with inverted output ('158) and with 3-state outputs (true: '257; inverted: '258).

EXERCISE 8.15

Show how to make a 2-input select from an AND-OR-INVERT gate.

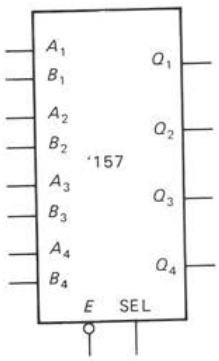


Figure 8.31. Quad 2-input select.

Although the function of a select gate can be performed by a mechanical switch in some cases, the gate is a far better solution, for several reasons: (a) it is cheaper; (b) all channels are switched simultaneously and rapidly; (c) it can be switched, nearly instantaneously, by a logic level generated elsewhere in the circuit; (d) even if the select function is to be controlled by a front-panel switch, it is better not to run logic signals around through cables and switches, to avoid capacitive signal degradation and noise pickup. With a select gate actuated by a dc level, you keep logic signals on the circuit board and get the bonus of simpler off-board wiring (a single line with pullup switched to ground by an SPST switch). Controlling circuit functions with externally generated dc levels in this manner is known as "cold switching," and it is a much better approach than controlling the signals themselves with switches, potentiometers, etc. Besides its other advantages, cold switching lets you bypass control lines with capacitors to eliminate interference, whereas signal lines cannot generally be bypassed. You will see some examples of cold switching later.

Transmission gates

As we discussed in Sections 3.11 and 3.12, with CMOS it is possible to make "transmission gates," simply a pair of

Microchip Ex. 1069, p. 22
Microchip v. Aptiv
IPR2024-00230

complementary MOSFET switches in parallel, so that an input (analog) signal between ground and V_{DD} is either connected through to the output through a low resistance (a few hundred ohms) or open-circuited (essentially infinite resistance). As you may remember, such a device is bidirectional and doesn't know (or care) which end is input and which end is output. Transmission gates work perfectly well with digital CMOS levels and are used extensively in CMOS design. Figure 8.32 shows the layout of the popular 4066 CMOS "quad bilateral switch." Each switch has a separate "control" input; input HIGH closes the switch, and LOW opens it. Note that transmission gates are merely switches, and therefore have no fanout; i.e., they simply pass input logic levels through to the output, without providing additional drive capability.

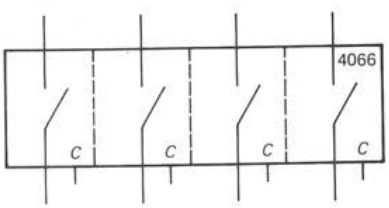


Figure 8.32. Quad transmission gate.

With transmission gates you can make 2-input (or more) select functions, usable with CMOS digital levels or analog signals. To select among a number of inputs, you can use a bunch of transmission gates (generating the control signals with a "decoder," as will be explained later). This is such a useful logic function that it has been institutionalized as the "multiplexer," which we will discuss next.

EXERCISE 8.16

Show how to make a 2-input select with transmission gates. You will need an inverter.

Multiplexers

The 2-input select gate is also known as a 2-input multiplexer. Multiplexers are also available with 4, 8, and 16 inputs (the 4-input variety comes as a dual unit, 2 in one package). A binary address is used to select which of the input signals appears at the output. For instance, an 8-input multiplexer has a 3-bit address input to address the selected data input (Fig. 8.33). The digital multiplexer illustrated is a '151. It has a STROBE (another name for ENABLE) input (negative-true), and it provides true and complemented outputs. When the chip is disabled (STROBE held HIGH), Q is LOW and Q' is HIGH, independent of the states of the address and data inputs.

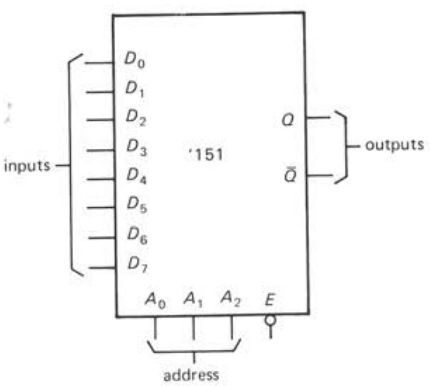


Figure 8.33. 8-input multiplexer.

In CMOS, two varieties of multiplexers are available. One type is for digital levels only, with an input threshold and "clean" regeneration of output levels according to the input state; that's also the way all TTL functions work. An example is the '153 TTL multiplexer. The other kind of CMOS multiplexer is analog and bidirectional; it's really just an array of transmission gates. The 4051–4053 CMOS multiplexers work this way (remember that logic made from transmission gates has no fanout). Since

Microchip Ex. 1069, p. 23
Microchip v. Aptiv
IPR2024-00230

transmission gates are bidirectional, these multiplexers can be used as "demultiplexers," or decoders. We will discuss them next.

EXERCISE 8.17

Show how to make a 4-input multiplexer using (a) ordinary gates, (b) gates with 3-state outputs, and (c) transmission gates. Under what circumstances would (c) be preferable?

You might wonder what to do if you want to select among more inputs than are provided in a multiplexer. This question comes under the general category of chip "expansion" (using several chips that have small individual capabilities to generate a larger capability), and it applies to decoders, memories, shift registers, arithmetic logic, and many other functions as well. In this case the job is easy (Fig. 8.34). Here we have expanded two 74LS151 8-input multiplexers into a 16-input multiplexer. There's an additional address bit, of course, and you use it to enable one chip or the other. The disabled chip holds its Q LOW, so an OR gate at the output completes the expansion. With three-state outputs the job is even simpler, since you can connect the outputs directly together.

Demultiplexers and decoders

A demultiplexer takes an input and routes it to one of several possible outputs, according to an input binary address. The other outputs are either held in the inactive state or open-circuited, depending on the type of demultiplexer.

A decoder is similar, except that the address is the only input, and it is "decoded" to assert one of n possible outputs. Figure 8.35 shows an example. This is the '138 "1-of-8 decoder." The output corresponding to (addressed by) the 3-bit input data is LOW; all others are HIGH. This particular decoder has three ENABLE inputs, all of which must be asserted (two LOW, one HIGH); otherwise all outputs are HIGH. A favorite use of the decoder is to cause different things to happen, depending on the state of a "counter" chip that drives it (more on this, soon). Decoders are commonly used when interfacing to microprocessor, to trigger different actions depending on the address; we'll treat this subject in detail in Chapter 10. Another common use of a decoder is to enable a sequence of actions in turn, according to an advancing address given by the output of a binary *counter* (Section 8.25). A close cousin of the '138 is the '139, a dual 1-of-4 decoder with a single LOW-true enable per section. Figure 8.36 shows how to use a pair of '138 1-of-8 decoders to generate a 1-of-16 decoder. No external gates are necessary,

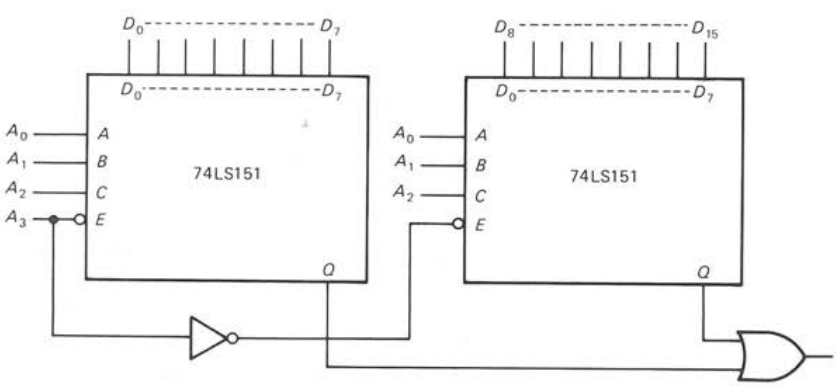


Figure 8.34. Multiplexer expansion.

Microchip Ex. 1069, p. 24
Microchip v. Aptiv
IPR2024-00230

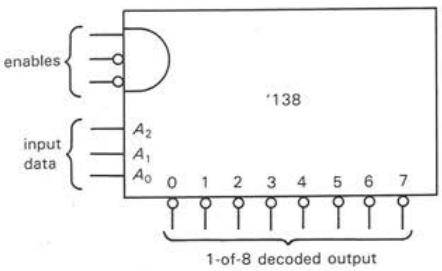


Figure 8.35. 1-of-8 decoder.

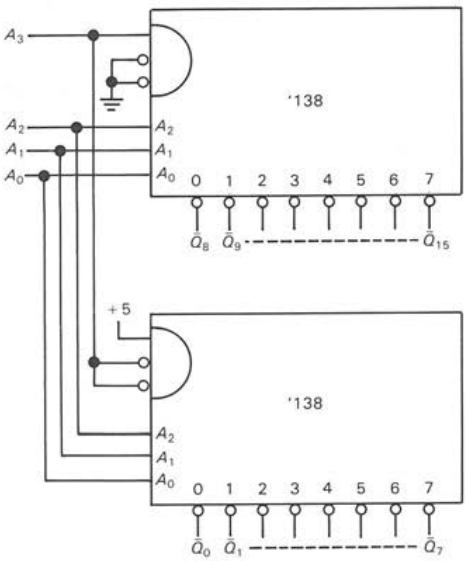


Figure 8.36. Decoder expansion.

since the '138 has enable inputs of both polarities.

EXERCISE 8.18

More expansion: Make a 1-of-64 decoder from nine '138s. Hint: Use one of them as an enabling switchyard for the others.

In CMOS logic, the multiplexers that use transmission gates are also demultiplexers, since transmission gates are bidirectional. When they are used that way, it is important to realize that the outputs that aren't selected are open-circuited. A pullup resistor, or equivalent, must be used to assert a well-defined logic level on those outputs (the same requirement as with TTL open-collector gates).

There is another kind of decoder generally available in all logic families. An example is the '47 "BCD-to-7-segment decoder/driver." It takes a BCD input and generates outputs on 7 lines corresponding to the segments of a "7-segment display" that have to be lit to display the decimal character. This type of decoder is really an example of a "code converter," but in common usage it is called a decoder. Table 8.6 at the end of the chapter lists most available decoders.

EXERCISE 8.19

Design a BCD-to-decimal (1-of-10) decoder using gates.

Priority encoder

The priority encoder generates a binary code giving the address of the highest-numbered input that is asserted. It is particularly useful in "parallel-conversion" A/D converters (see next chapter) and in microprocessor system design. An example is the '148 8-input (3 output bits) priority encoder. The '147 encodes 10 inputs.

EXERCISE 8.20

Design a "simple" encoder: a circuit that outputs the (2-bit) address telling which of 4 inputs is HIGH (all other inputs must be LOW).

Adders and other arithmetic chips

Figure 8.37 shows a "4-bit full adder." It adds the 4-bit number A_i to the 4-bit number B_i, generating a 4-bit sum S_i plus carry bit C_o. Adders can be "expanded" to add larger numbers: The "carry-in" input C_i is provided to accept the carry out of the next lower adder. Figure 8.38 shows how you would add two 8-bit numbers.

Microchip Ex. 1069, p. 25
Microchip v. Aptiv
IPR2024-00230

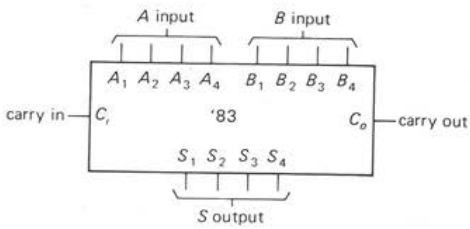


Figure 8.37. 4-bit full adder.

A device known as an arithmetic logic unit (ALU) is often used as an adder. It actually has the capability of performing a number of different functions. For instance, the '181 4-bit ALU (expandable to larger word lengths) can do addition, subtraction, bit shifts, magnitude comparison, and a few other functions. Adders and ALUs do their arithmetic in times measured in nanoseconds to tens of nanoseconds, depending on logic family.

Dedicated integer multiplier chips are available in configurations such as 8 bits times 8 bits, or 16 bits times 16 bits. A variation that is particularly suited to digital signal processing is the so-called MAC (multiplier-accumulator), which accumulates a sum of products; these, too, are available in sizes up to 32×32, with 64-bit product plus a few additional bits to keep the sum from overflowing. CMOS MACs and multipliers are available with typical speeds of 25–50ns; ECL multipliers are much faster – 5ns (typ) for 16×16 multiply.

Another arithmetic chip that's handy in digital signal processing is the *correlator*, which compares the corresponding bits of a pair of bit strings, calculating the number of bits that agree. A typical correlator chip compares a pair of 64-bit words, which can be shifted in internal shift registers. Any pattern of bits can be ignored ("masked") in the correlation. Typical speeds are 30ns, i.e., a bit stream can be clocked through at 35MHz, with a 7-bit correlation available at each clock tick. A variation (known as an FIR digital filter) calculates instead the true sum (with carry) of the true pairwise product of a pair of integer strings; typical

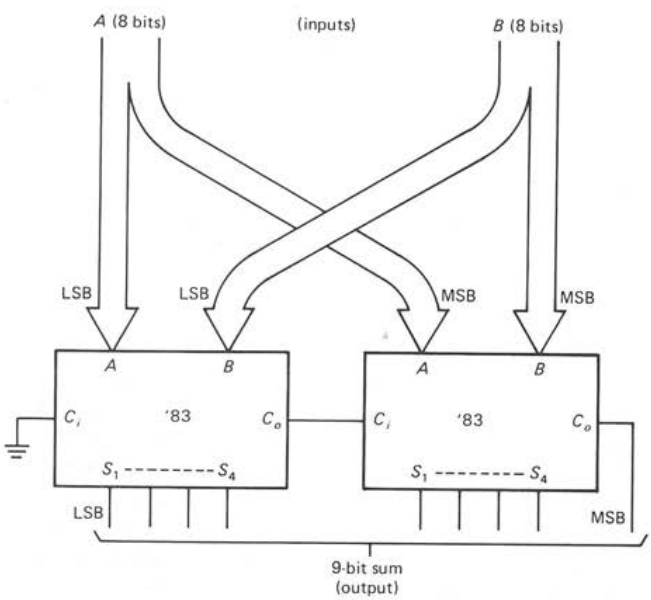


Figure 8.38. Adder expansion.

Microchip Ex. 1069, p. 26
Microchip v. Aptiv
IPR2024-00230

sizes are 4- to 10-bit integers, with string lengths of 3 to 8 words (expandable to greater length, of course).

The most complex arithmetic chips are the *floating-point processors*, which perform compares, sums, and products, as well as trigonometric functions, exponentials, and square roots. These are usually designed to work closely with particular microprocessors, and they generally conform to a nice standard known as IEEE P754, which specifies word size (up to 80 bits), format, etc. Examples (plus *matching microprocessors*) are the 8087 (*8086/8*), 80387 (*80386*), and 68881 (*68020*). These are truly stunning performers, with speeds of 10Mflops (million floating-point operations per second) or more.

Magnitude comparators

Figure 8.39 shows a 4-bit magnitude comparator. It determines the relative sizes of the 4-bit input numbers A and B and tells you via outputs whether $A < B$, $A = B$, or $A > B$. Inputs are provided for expansion to numbers larger than 4 bits.

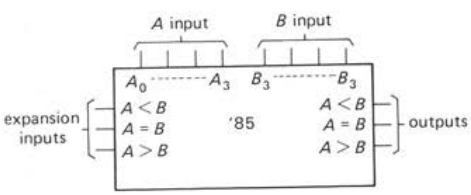


Figure 8.39. Magnitude comparator.

EXERCISE 8.21

Construct a magnitude comparator, using XOR gates, that tells whether or not $A = B$, where A and B are 4-bit numbers.

Table 8.7 at the end of the chapter lists most available magnitude comparators.

Parity generator/checker

This chip is used to generate a parity bit to be attached to a "word" when transmitting (or recording) data and to check the received parity when such data are recovered. Parity can be even or odd (e.g., with odd parity the number of 1 bits in each character is odd). The '280 parity generator, for instance, accepts a 9-bit input word, giving an even and an odd parity bit output. The basic construction is an array of exclusive-OR gates.

EXERCISE 8.22

Figure out how to make a parity generator using XOR gates.

Programmable logic devices

You can build your own custom combinational (and even sequential) logic on a chip, using ICs that contain an array of gates with programmable interconnections. There are several varieties, of which the most popular are PALs (programmable array logic) and PLAs (programmable logic arrays). PALs, in particular, have become extremely inexpensive and flexible and should form a part of every designer's toolbox of tricks. We will describe combinational PALs in the next section.

□ Some other strange functions

There are many other interesting MSI combinational chips worth knowing about. For example, in CMOS you can get a "majority logic" IC that tells you whether or not a majority of n inputs are asserted. Also available is a BCD "9's complementer," whose function is obvious. A "barrel-shifter" IC shifts an input word over by n (selectable) bits and can be expanded to any width.

Microchip Ex. 1069, p. 27
Microchip v. Aptiv
IPR2024-00230

transmit and receive *simultaneously* ("full duplex") on the same line (Fig. 9.41). When used this way, each chip can be transmitting to the other transceiver while receiving asynchronously from it at the same time, without cross-talk, at speeds of 100MHz or more.

You can drive a length of coax directly from +5 volt logic if you use a single *npn* emitter follower (Fig. 9.42). The 2N4401 is a husky little transistor with plenty of beta at high currents ($h_{FE} > 100$ at $I_C =$ 150mA). The 10 ohm resistor is for short-circuit protection. In comparison with the carefully designed and expensive 50 ohm cable driver chips that are available, this circuit is embarrassingly simple. Note that the "open-emitter" output must be loaded with a low-resistance return to ground in order to work, which is also true of some IC cable driver chips.

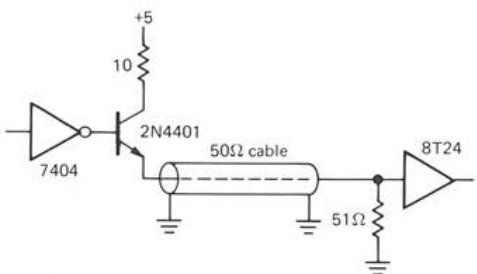


Figure 9.42. Simple 50 ohm cable driver.

Fiber-optics

An exciting new method of signal transmission is provided by fiber-optic cables. These come as nice plastic-coated cables, with matching connectors, emitters, and detectors. High-performance fiber-optic cables can transmit bandwidths of several gigahertz over distances of tens or hundreds of kilometers, with losses of fractional decibels per kilometer. Compared with coaxial cables, which are dispersive (different frequencies travel at different speeds, and with different amounts of loss, thereby corrupting waveforms), fiber-optic cables can be designed so that dispersion is negligible. And fiber-optic cables are insulators, so they can be used to send signals between devices with separate grounds, or at different voltages. Unlike ordinary cables, they do not act as antennas for radio interference or impulsive noise. They are lighter, safer, and more rugged than conventional cables, and they are potentially cheaper, too.

There are several types of fiber-optic cables, the choice involving a trade-off between cost and performance (length times bandwidth). The cheapest is the so-called step-index, multimode fiber, typically a plastic fiber 1mm in diameter; you would normally drive it with an infrared LED (as opposed to a laser diode) and use a phototransistor or pin-diode detector. Motorola makes an inexpensive set of driver/receivers (less than $1 each) that mate directly with jacketed fiber-optic cable (the MFOE71/MFOD71-73 series); with them you can send data at rates up to 1Mb/s over 30 feet of the plastic cable just described (see Fig. 9.43). Higher-performance cables use glass fibers – step-index multimode, graded-index multimode (better), or single-mode (best). With 200μm glass step-index fiber you can run 5Mb/s over a 1km path with standard off-the-shelf fiber-optic components, which include connectors, couplers, splitter/combiners, and detectors with built-in amplifiers. The current record for high-bandwidth long-haul fiber-optic communication is 4GHz over a 120km path, using no repeaters.

ANALOG/DIGITAL CONVERSION

9.15 Introduction to A/D conversion

In addition to applying the purely "digital" interfacing (switches, lights, etc.) discussed in the last few sections, it is

Microchip Ex. 1069, p. 28
Microchip v. Aptiv
IPR2024-00230

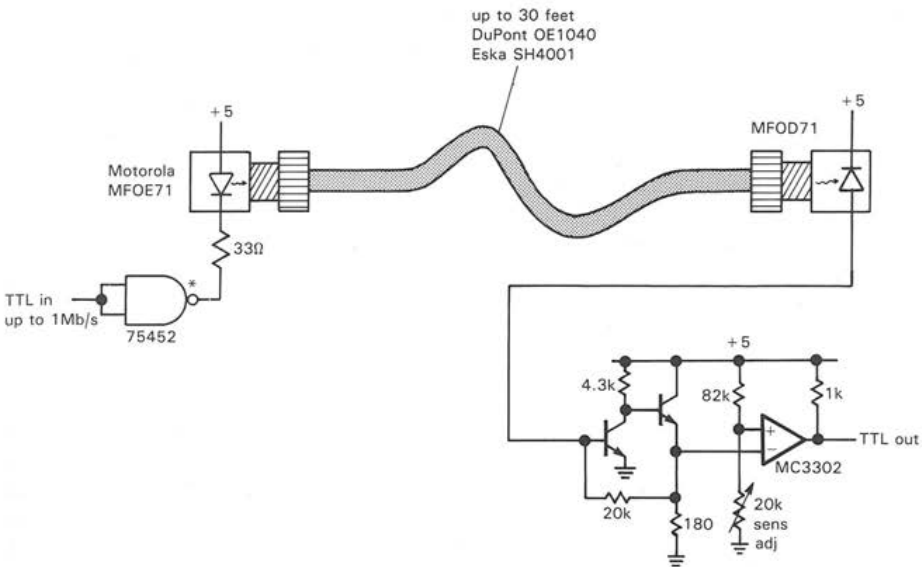


Figure 9.43. Inexpensive fiber-optic link. (Adapted from Fig. 7, MFOD71 Data Sheet, Motorola, Inc.)

often necessary to convert an analog signal to an accurate digital number proportional to its amplitude, and vice versa. This is essential in any application in which a computer or processor is logging or con trolling an experiment or process, or whenever digital techniques are used to do a "normally" analog job. Applications in which analog information is converted to an intermediate digital form for error- and noise-free transmission [e.g., "digital audio" or pulse-code modulation (PCM)] make heavy use of analog/digital conversion. It is necessary in a wide variety of measurement instruments (including ordinary bench instruments such as digital multimeters and more exotic instruments such as transient averagers, "glitch catchers," and digital memory oscilloscopes), as well as signal-generation and processing instruments such as digital waveform synthesizers and data encryption devices.

Finally, conversion techniques are essential ingredients for the generation of analog displays by a digital instrument, e.g., a meter indication or xy display (or plot) created by a computer. Even in a relatively unsophisticated electronic apparatus there are plenty of nice applications that call for A/D and D/A conversion, and it is worth developing familiarity with the various techniques and available modules used in analog/digital conversion, especially since the day of the $5 A/D and D/A converter is here.

Our treatment of the various conversion techniques will not be aimed at developing skill in converter design itself. Rather, we will try to point out the advantages and disadvantages of each method, since in most cases the sensible thing is to buy commercially available chips or modules, rather than build the converter from scratch. An understanding of conversion techniques and idiosyncrasies will guide

Microchip Ex. 1069, p. 29
Microchip v. Aptiv
IPR2024-00230

you in choosing among the hundreds of available units.

Codes

At this point you should review Section 8.03 on the various number codes used to represent signed numbers. Offset binary and 2's complement are commonly used in A/D conversion schemes, with sign-magnitude and Gray codes also popping up from time to time. Here is a reminder:

	Offset binary	*2's Complement*
+Full scale	11111111	01111111
+Full scale−1	11111110	01111110
↓	↓	↓
0+1 LSB	10000001	00000001
0	10000000	00000000
0−1LSB	01111111	11111111
↓	↓	↓
−Full scale+1	00000001	10000001
−Full scale	00000000	10000000

Converter errors

The subject of A/D and D/A errors is a complicated one, about which whole volumes could be written. According to Bernie Gordon at Analogic, if you think a high-accuracy converter system lives up to its claimed specifications, you probably haven't looked closely enough. We won't go into the application scenarios necessary to support Bernie's claim, but we will show the four most common types of converter errors. Rather than bore you with a lot of complicated talk, we'll just present self-explanatory graphs of the four most common errors: offset error, scale error, nonlinearity, and nonmonotonicity (Fig. 9.44).

9.16 Digital-to-analog converters (DACs)

The goal is to convert a quantity specified as a binary (or multidigit BCD) number to a voltage or current proportional to the value of the digital input. There are several popular methods:

Scaled resistors into summing junction

As you saw in Section 4.09, by connecting a set of resistors to an op-amp summing junction you get an output proportional to a weighted sum of the input voltages. (Fig. 9.45). This circuit generates an output from zero to −10 volts, with full output corresponding to an input count of 64. Actually, the maximum input count is always 2^n-1, i.e., all bits set to 1. In this case the maximum input count is 63, with output voltage of $-10 \times 63/64$. By changing the feedback resistor, you can generate an output of zero to −6.3 volts (i.e., output in volts numerically equal to −1/10 of input count), or you can add an inverting amplifier, or dc offset at the summing junction, to get positive outputs. By changing the input resistor values you can properly convert a multidigit BCD input code, or any other weighted code. The input voltages must be clamped to an accurate reference, and the smaller input resistors must be of correspondingly higher precision. Of course, the switch resistance must be smaller than $1/2^n$ of the smallest resistor, an important consideration since in actual circuits the switching is done with transistor or FET switches. This conversion technique is used only in fast, low-precision converters.

EXERCISE 9.2

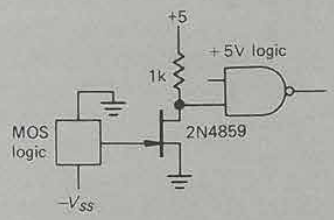
Design a 2-digit BCD D/A converter. Assume that the inputs are zero or +1 volt; the output should go from zero to 9.9 volts.

R-2R ladder

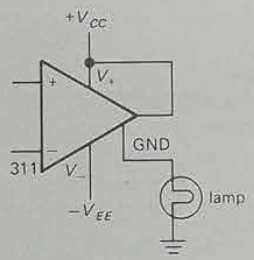
The scaled resistor technique becomes awkward when you have more than a few bits. For example, in a 12-bit converter you would need a range of resistor values of 2000:1, with corresponding precision in the small resistor values. An elegant solution is provided by the *R-2R* ladder,

Microchip Ex. 1069, p. 30
Microchip v. Aptiv
IPR2024-00230

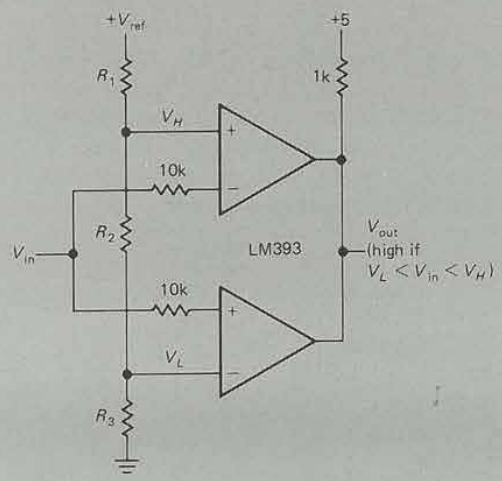
Circuit ideas



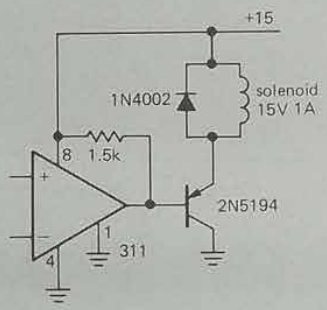
A. negative logic to TTL shifter



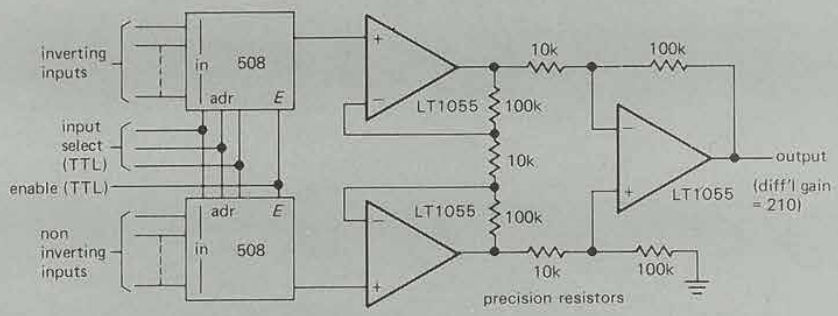
B. driving ground-returned load



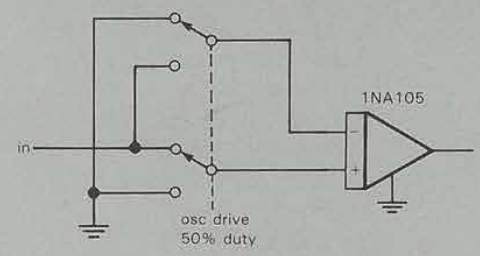
C. window discriminator



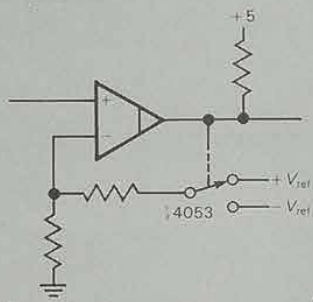
D. solenoid driver



E. 8-channel multiplexer with differential input



F. phase detector with 60dB osc rejection



G. precision Schmitt trigger

Figure 9.95

Microchip Ex. 1069, p. 31
Microchip v. Aptiv
IPR2024-00230

and operate at fixed gain), these are true op-amps (you use external feedback to determine the configuration and gain), and they are usable as closed-loop amplifiers to 100MHz or more. Many of these wide bandwidth op-amps use the vertical *pnp* process to get good performance. In a deviation from normal op-amp practice, you will often find unsymmetrical input impedances, with *current feedback*. Take a look at Table 13.2 for a sampling of these fast beasts.

RADIOFREQUENCY CIRCUIT ELEMENTS

13.09 Transmission lines

Before proceeding to the subject of communications circuits, it is necessary to deal briefly with the interesting subject of transmission lines. You have met these earlier in connection with digital signal communications in Chapter 9, where we introduced the ideas of characteristic impedance and line termination. Transmission lines play a central role in radiofrequency circuits, where they are used to pipe signals around from one place to another within a circuit, and often to an antenna system. Transmission lines provide one of the most important exceptions to the general principles (see Chapter 1) that a signal source ideally should have a source impedance small compared with the impedance of the load being driven and that the load should present an input impedance large compared with the source impedance driving it. The equivalent rule for transmission lines is that the load (and possibly the source) should present an impedance equal to the characteristic impedance of the line. The line is then "matched."

Transmission lines for signals of moderate frequency (up to 1000MHz, say) come in two major types: parallel conductors and coaxial line. The former is typified by the inexpensive molded 300 ohm "twin lead" used to bring the signal from a television antenna to the receiver, and the latter is widely used in short lengths with BNC fittings to carry signals between instruments. (Fig. 13.17).

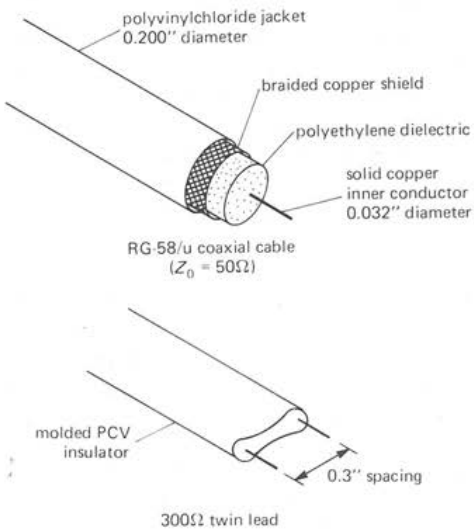


Figure 13.17

In the domain of ultra-high-frequency circuitry there are "stripline" techniques that involve parallel-conductor transmission lines as part of the actual circuit, and at the higher "microwave" frequencies (upwards of 2GHz, say) conventional lumped circuit elements and transmission lines are replaced by cavity and waveguide techniques, respectively. Except at these extremes of frequency, the familiar coaxial cable is probably the best choice for most radiofrequency applications. Compared with parallel conductor line, a properly matched coax line has the advantage of being totally shielded, i.e., there is no radiation or pickup of external signals.

Characteristic impedance and matching

A transmission line, whatever its form, has a "characteristic impedance" Z_0, meaning

(a) see footnote to Table 4.1. (b) input current (μA) if current-feedback type. (c) nV/√Hz at 10kHz. (d) full-swing output. (e) number of 10bit (monolithic-terminated video) loads with video levels. (f) E - external. (g) at 1kHz. (h) G_V = +2, 50Ω input. (i) ±3V out. (j) typical, over full temp. (k) 0.1%. (l) programmable. (m) min/max. (n) μV rms, 10Hz to 100kHz. (o) above 5MHz. (p) VA706 for G = 1; VA708 for G > 3. (q) 0.5MHz. (r) 6361 for G=1; 6365 for G>25. (t) typical.

Microchip Ex. 1069, p. 32
Microchip v. Aptiv
IPR2024-00230

that a wave moving along the line has a ratio of voltage to current equal to Z_0. For a lossless line, Z_0 is resistive and equal to the square root of L/C, where L is the inductance per unit length and C is the capacitance per unit length. Typical coaxial lines have impedance in the range of 50 to 100 ohms, whereas parallel-conductor lines have impedances in the range of 300 to 1000 ohms.

When used with high-frequency (or short-rise-time) signals, it is important to "match" the load to the characteristic impedance of the line. The important facts are the following: (a) A transmission line terminated with a load equal to its characteristic impedance (resistance) will transfer an applied pulse to the termination without reflection. In that case all the power in the signal is transferred to the load. (b) The impedance looking into such a terminated line, at any frequency, is equal to its characteristic impedance (Fig. 13.18).

This is surprising at first, since at low frequencies you tend to think of a length of coax as a small capacitive load, generally a pretty high (capacitive) impedance. Also, at low frequencies (wavelength $\gg$ length of cable) there is no need to match the line's impedance, provided you can handle the capacitance (typically 30pF per foot). If the cable is terminated with a resistor, on the other hand, it magically becomes a pure resistance at all frequencies.

Mismatched transmission lines

A mismatched transmission line has some interesting, and occasionally useful, properties. A line terminated in a short circuit produces a reflected wave of opposite polarity, with the delay time of the reflected wave determined by the electrical length of the line (the speed of wave propagation in coax lines is about two-thirds the speed of light, because of the solid dielectric spacing material). You can see the reason for this, since the short circuit enforces a point

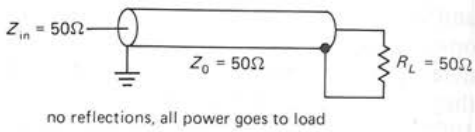


Figure 13.18

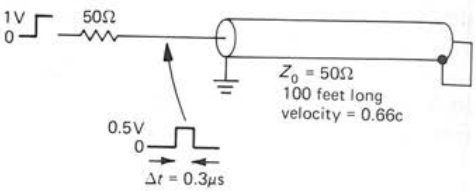


Figure 13.19. Pulse generation with shorted transmission line (inverted reflection).

of zero voltage at the end; the cable produces this obligatory condition by creating a wave of opposite phase at the short. In similar manner, an open-circuited cable (boundary condition of zero *current* at the end) produces a noninverted reflection of amplitude equal to the applied signal.

This property of a shorted cable is sometimes exploited to generate a short pulse from a step waveform. The step input is applied to the cable input through a resistance equal to Z_0, with the other end of the cable shorted. The waveform at the input is a pulse of width equal to the round-trip travel time, since the reflected step cancels the input (Fig. 13.19).

Cables terminated with a resistance R unequal to Z_0 also produce reflections, although of lesser amplitude. The reflected wave is inverted if $R < Z_0$ and uninverted if $R > Z_0$. The ratio of reflected wave amplitude to incident wave amplitude is given by

$$A_r/A_i = (R - Z_0)/(R + Z_0)$$

Transmission lines in the frequency domain

Looked at in the frequency domain, a transmission line matched at the far end looks like a load of impedance Z_0, i.e., a

Microchip Ex. 1069, p. 33
Microchip v. Aptiv
IPR2024-00230

pure resistance if line losses are neglected. That makes sense, because it just swallows any wave you apply, all the power going into the matching resistor. This is true independent of cable length or wavelength. It is when you deal with mismatched lines that things begin to get interesting in the frequency domain. Since, for a given line length, the reflected wave arrives back at the input with a phase (relative to the applied signal) that depends on applied frequency, the impedance seen looking into the input depends on the mismatch and on the electrical length of the transmission line, in wavelengths.

As an example, a line that is an odd number of quarter wavelengths long terminated in an impedance Z_{load} at the far end presents an input impedance $Z_{\text{in}} = Z_0^2/Z_{\text{load}}$. If the load is resistive, the input will look resistive. On the other hand, a line that is an integral number of half wavelengths long presents an input impedance equal to its terminating impedance (Fig. 13.20).

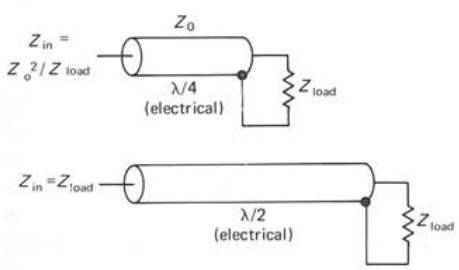


Figure 13.20

The presence of reflected signals on a transmission line is not necessarily bad. For operation at a single frequency, a mismatched line can be driven (through a line tuner) in such a way as to match its resultant input impedance, often with only negligibly greater line losses (due to higher voltages and currents for the same forward power) than with a matched load. But a mismatched line has different properties at different frequencies (the famous "Smith chart" can be used to determine transmission-line impedances and "standing-wave ratio," or SWR, a measure of the amplitude of reflected waves), making it undesirable for broadband or multifrequency use. In general, strive to terminate a transmission line in its characteristic impedance, at least at the receiving end.

□ 13.10 Stubs, baluns, and transformers

There are some interesting applications of transmission lines that exploit the properties of mismatched sections or generally use sections of line in an unconventional way. The simplest is the quarter-wave matching section, which exploits the relationship $Z_{\text{in}} = Z_0^2/Z_{\text{load}}$. This can be rearranged to read $Z_0 = (Z_{\text{in}}Z_{\text{load}})^{1/2}$. In other words, a quarter-wave section can be used to match any two impedances by choosing the characteristic impedance of the matching section appropriately.

In a similar manner, a short length of transmission line (a "stub") can be used to "tune" a mismatched load by simply putting the stub across or in series with the mismatched line, choosing the stub length and termination (open or shorted) and its position along the mismatched line correctly. In this sort of application the stub is really functioning as a circuit element, not a transmission line. At very short wavelengths the use of sections of transmission line as circuit elements is common (Fig. 13.21).

Sections of transmission line (or a transformer made with several interconnected windings) can be used to construct a "balun," a device for matching an unbalanced line (coax) to a balanced load (e.g., an antenna). There are simple configurations for making fixed-impedance transformations at the same time (1:1 and 4:1 are common). Perhaps the nicest circuit

Microchip Ex. 1069, p. 34
Microchip v. Aptiv
IPR2024-00230