

Digital Signal Processing

WILEY SERIES ON TOPICS IN DIGITAL SIGNAL PROCESSING

- DFT/FFT and Convolution Algorithms and Implementation
by C. S. Burrus and T. W. Parks
- Digital Signal Processing: Laboratory Experiments Using C and the TMS320C31 DSK
by Rulph Chassaing
- Digital Signal Processing with the TMS320C25
by Rulph Chassaing and Darrell W. Horning
- A Simple Approach to Digital Signal Processing
by Craig Marven and Gillian Ewers
- Digital Filter Design
by T. W. Parks and C. S. Burrus
- Theory and Design of Adaptive Filters
by John R. Treichler and C. Richard Johnson

Digital Signal Processing

Laboratory Experiments
Using C and the TMS320C31 DSK

RULPH CHASSAING

University of Massachusetts, Dartmouth



A Wiley-Interscience Publication

JOHN WILEY & SONS, INC.

New York • Chichester • Weinheim • Brisbane • Singapore • Toronto

Designations used by companies to distinguish their products are often claimed as trademarks. In all instances where John Wiley & Sons, Inc., is aware of a claim, the product names appear in initial capital or ALL CAPITAL LETTERS. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic or mechanical, including uploading, downloading, printing, decompiling, recording or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the Publisher. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 605 Third Avenue, New York, NY 10158-0012, (212) 850-6011, fax (212) 850-6008, E-Mail: PERMREQ@WILEY.COM.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in rendering professional services. If professional advice or other expert assistance is required, the services of a competent professional person should be sought.

ISBN 0-471-20065-4.

This title is also available in print as ISBN 0-471-29362-8.

For more information about Wiley products, visit our web site at www.Wiley.com.

Contents

Preface	ix
List of Examples	xiii
List of Programs/Files on Accompanying Disk	xv
1 DIGITAL SIGNAL PROCESSING DEVELOPMENT SYSTEM	1
1.1 Introduction	1
1.2 DSK Support Tools	2
1.3 Programming Examples to Test the DSK Tools	3
1.4 Additional Support Tools	14
1.5 Experiment 1: Testing the DSK Tools	16
References	17
2 ARCHITECTURE AND INSTRUCTION SET OF THE TMS320C3x PROCESSOR	19
2.1 Introduction	19
2.2 TMS320C3x Architecture and Memory Organization	21
2.3 Addressing Modes	25
2.4 TMS320C3x Instruction Set	26
2.5 Assembler Directives	30
2.6 Other Considerations	32
2.7 Programming Examples Using TMS320C3x and C code	34
2.8 Experiment 2: TMS320C3x Instructions and Associated Tools	47
References	48
3 INPUT AND OUTPUT WITH THE DSK	51
3.1 Introduction	51
3.2 The Analog Interface Circuit (AIC) Chip	53

3.3	Interrupts and Peripherals	59
3.4	Programming Examples Using TMS320C3x and C code	60
3.5	PC Host - TMS320C31 Communication	80
3.6	External/Flash Memory and I/O with 16-bit Stereo Audio Codec	87
3.7	Experiment 3: Input and Output with the DSK	88
	References	89
4	FINITE IMPULSE RESPONSE FILTERS	91
4.1	Introduction to the z-Transform	91
4.2	Discrete Signals	96
4.3	Finite Impulse Response Filters	97
4.4	FIR Implementation Using Fourier Series	99
4.5	Window Functions	103
4.6	Filter Design Packages	106
4.7	Programming Examples using TMS320C3x and C Code	106
4.8	Experiment 4: FIR Filter Implementation	129
	References	131
5	INFINITE IMPULSE RESPONSE FILTERS	135
5.1	Introduction	135
5.2	IIR Filter Structures	136
5.3	Bilinear Transformation	143
5.4	Programming Examples Using TMS320C3x and C Code	150
5.5	Experiment 5: IIR Filter Design and Implementation	160
	References	163
6	FAST FOURIER TRANSFORM	165
6.1	Introduction	165
6.2	Development of the FFT Algorithm with Radix-2	165
6.3	Decimation-in-Frequency FFT Algorithm with Radix-2	167
6.4	Decimation-in-Time FFT Algorithm with Radix-2	174
6.5	Bit Reversal for Unscrambling	178
6.6	Development of the FFT Algorithm with Radix-4	179
6.7	Inverse Fast Fourier Transform	183
6.8	Programming Examples Using C and TMS320C3x Code	183
6.9	Experiment 6: FFT Implementation	193
	References	194

7	ADAPTIVE FILTERS	195
7.1	Introduction	195
7.2	Adaptive Structures	197
7.3	Programming Examples Using C and TMS320C3x Code	199
7.4	Experiment 7: Adaptive Filtering Implementation	221
	References	222
8	DSP APPLICATIONS AND PROJECTS	223
8.1	Banks of FIR Filters	223
8.2	Multirate Filter	228
8.3	Pass/Fail Alarm Generator	235
8.4	External Interrupt for Control	239
8.5	Miscellaneous Applications and Projects	242
	References	254
	APPENDIX A TMS320C3X INSTRUCTION SET AND REGISTERS	257
A.1	TMS320C3x Instruction Set	257
A.2	TMS320C3x Registers	257
	Reference	263
	APPENDIX B SUPPORT TOOLS	265
B.1	Code Explorer Debugger from GO DSP	265
B.2	Virtual Instrument Using Shareware Utility Package	269
B.3	Filter Design and Implementation Using DigiFilter	271
B.4	MATLAB for FIR/IIR Filter Design, FFT, and Data Acquisition	275
	References	281
	APPENDIX C EXTERNAL AND FLASH MEMORY	283
C.1	External Memory	286
C.2	Flash Memory	287
	References	289
	APPENDIX D INPUT AND OUTPUT WITH 16-BIT STEREO AUDIO CODEC	291
	References	298
	Index	299

Preface

Digital signal processors, such as the TMS320 family of processors, are found in a wide range of applications such as in communications and controls, speech processing, and so on. They are used in Fax, modems, cellular phones, etc. These devices have also found their way into the university classroom, where they provide an economical way to introduce real-time digital signal processing (DSP) to the student.

With the introduction of Texas Instruments' third-generation TMS320C3x processor, floating-point instructions and a new architecture that supports features which facilitate the development of high-level language compilers appeared. The C optimizing compiler takes advantage of the special features of the TMS320C3x processor such as parallel instructions and delayed branches. Throughout the book, we refer to the C/C++ language as simply C. Generally, the price paid for going to a high-level language is a reduction in speed and a similar increase in the size of the executable file. Although TMS320C3x/assembly language produces fast code, problems with documentation and maintenance may exist. A compromise solution is to write time-critical routines in TMS320C3x code that can be called from C.

This book is intended primarily for senior undergraduate and first-year graduate students in electrical and computer engineering and as a tutorial for the practicing engineer. It is written with the conviction that the principles of DSP can best be learned through interaction in a laboratory setting, where the student can appreciate the concepts of DSP through real-time implementation of experiments and projects. The background assumed is a system course and some knowledge of assembly language or a high-level language such as C.

Most chapters begin with a theoretical discussion, followed by representative examples that provide the necessary background to perform the concluding experiments. There are a total of 60 solved programming examples using both TMS320C3x and C code. Several sample projects are also discussed.

Programming examples using both TMS320C3x and C code are included throughout the text. This can be useful to the reader who is familiar with both DSP and C programming, but who is not necessarily an expert in both. Although the

X Preface

reader who elects to study the programming examples in either TMS320C3x or C code will benefit from this book, the ideal reader is one with an appreciation for both TMS320C3x and C code.

This book can be used in the following ways:

1. For a laboratory course using many of the Examples and Experiments from Chapters 1-7. The beginning of the semester can be devoted to short programming examples and experiments and the remainder of the semester used for a final project.
2. For a senior undergraduate or first-year graduate design project course, using Chapters 1-5, selected materials from Chapters 6-8, and Appendices C and D.
3. For the practicing engineer as a tutorial and for workshops and seminars.

Chapter 1 introduces the tools through three examples. These tools include an assembler and a debugger that are provided with the DSP Starter Kit (DSK). Program examples in C can be tested without a C compiler since all associated executables files are on the accompanying disk. Chapter 2 covers the architecture and the instructions available for the TMS320C3x processor. Special instructions and assembler directives that are useful in DSP are discussed. Chapter 3 illustrates input and output (I/O) with the two-input analog interface chip (AIC) on the DSK board through several programming examples. An alternative I/O with a 16-bit stereo audio codec that can be interfaced with the DSK is described.

Chapter 4 introduces the z-transform and discusses finite impulse response (FIR) filters and the effect of window functions on these filters. Chapter 5 covers infinite impulse response (IIR) filters. Programming examples to implement FIR and IIR filters, in both TMS320C3x and C code, are included.

Chapter 6 covers the development of the fast Fourier transform (FFT). Programming examples on FFT are included. Chapter 7 demonstrates the usefulness of the adaptive filter for a number of applications with the least mean square (LMS). Chapter 8 discusses a number of DSP applications.

A disk included with this book contains all the programs discussed in the text. See page xv for a list of the programs/files included on the disk.

During the summers of 1996-1998, a total of 115 faculty members from over 100 Institutions took my DSP and Applications workshops supported by grants from the National Science Foundation (NSF). I am thankful to them for their encouragement, participation and feedback on this book. In particular, Dr. Hisham Alnajjar from the University of Hartford, Dr. Armando Barreto from Florida International University, Dr. Paul Giolma from Trinity University, Dr. William Monaghan from the College of Staten Island—CUNY, and Dr. Mark Wickert from the University of Colorado at Colorado Springs. I also thank Dr. Darrell Horning from the University of New Haven, with whom I coauthored the text *Digital Signal Processing with the TMS320C25*, for introducing me to book-writing. I thank all the students who have taken my DSP and Senior Design Project courses. I am particularly indebted to two former students, Bill Bitler and Peter Martin, who have worked with me for many

years and have contributed to this book as well as to my previous book *Digital Signal Processing with C and the TMS320C30*.

The support of the National Science Foundation's Undergraduate Faculty Enhancement (UFE) Program in the Division of Undergraduate Education, Texas Instruments, and the Roger Williams University Research Foundation is appreciated.

RULPH CHASSAING

List of Examples

1.1	Matrix/vector multiplication using TMS320C3x code	4
1.2	Sine generation with 4 points using TMS320C3x code	8
1.3	Matrix/vector multiplication using C code	11
2.1	Addition of four values using TMS320C3x code	34
2.2	Multiplication of two arrays using TMS320C3x code	35
2.3	Background for digital filtering using TMS320C3x code	37
2.4	Matrix/vector multiplication using TMS320C3x code	42
2.5	Addition using C and C-called TMS320C3x assembly function	42
2.6	Matrix/vector multiplication using C and C-called TMS320C3x assembly function	45
3.1	Internal interrupt using TMS320C3x code	60
3.2	Sine generation with AIC data using TMS320C3x code	62
3.3	Loop/echo with AIC routines in separate file, using TMS320C3x code	65
3.4	Loop/echo with interrupt using TMS320C3x code	69
3.5	Sine generation with interrupt using TMS320C3x code	70
3.6	Pseudorandom noise generation using TMS320C3x code	70
3.7	Alternative pseudorandom noise generation with interrupt using TMS320C3x code	73
3.8	Loop/echo with AIC data using C code	75
3.9	Loop/echo Calling AIC routines in separate file, using C code	75
3.10	Loop/echo with interrupt using C code	79
3.11	PC-TMS320C31 communication using C code	82
3.12	Loop control with PC-TMS320C31 communication using C code	84
3.13	Data acquisition with the DSK using C and TMS320C3x code	85
4.1	FIR lowpass filter simulation with 11 coefficients using TMS320C3x code	108
4.2	FIR bandpass filter simulation with 45 coefficients using TMS320C3x code	111
4.3	Generic FIR filter specified at run-time, using TMS320C3x code	112
4.4	FIR filter incorporating pseudorandom noise as input, using TMS320C3x code	115

xiv List of Examples

4.5 Mixed-code FIR filter with main C program calling filter function in TMS320C3x code	117
4.6 FIR filter with data move using C code	121
4.7 FIR filter using C code	123
4.8 FIR filter with samples shifted, using C code	125
4.9 FIR filter design using filter development package	127
5.1 Sine generation by recursive equation using TMS320C3x code	152
5.2 Cosine generation by recursive equation using TMS320C3x code	154
5.3 Sine generation by recursive equation using C code	154
5.4 Sixth-order IIR bandpass filter using TMS320C3x code	156
5.5 Sixth-order IIR bandpass filter using C code	160
6.1 Eight-point complex FFT using C code	184
6.2 Eight-point FFT with real-valued Input, using mixed C and TMS320C3x code	187
6.3 Real-time 128-Point FFT using mixed code	191
7.1 Adaptive filter using C code compiled with Borland C/C++	200
7.2 Adaptive filter for noise cancellation using C code	203
7.3 Adaptive predictor using C code	206
7.4 Adaptive predictor with table lookup for delay, using C code	208
7.5 Adaptive notch filter with two weights, using TMS320C3x code	210
7.6 Adaptive predictor using TMS320C3x code	215
7.7 Real-time adaptive filter for noise cancellation, using TMS320C3x code	218
B.1 FIR filter using Code Explorer for debugging and plotting	265
B.2 FIR filter design and implementation using DigiFilter	272
B.3 IIR filter design and implementation using DigiFilter	274
B.4 FIR filter design using MATLAB	275
B.5 Multiband FIR filter design using MATLAB	276
B.6 IIR filter design using MATLAB	277
B.7 $H(z)$ from $H(s)$ using bilinear function in MATLAB	278
B.8 Eight-point FFT and IFFT using MATLAB	279
B.9 Data acquisition, plotting, and FFT using MATLAB	279
C.1 Multirate filter with 10 bands using external memory and TMS320C3x code	287
C.2 Sine generation with four points from flash memory, using C code	287
C.3 FIR bandpass filter from flash memory using C code	289
D.1 Loop programs for input and output with the Crystal 16-bit stereo audio codec using TMS320C3x code	297
D.2 FIR filter with the Crystal stereo audio codec using TMS320C3x code	298

List of Programs/Files on Accompanying Disk

README	TXT	169
EGAVGA	BGI	5363

Directory of CH1

MATRIX	ASM	1628
SINE4P	ASM	1118
MATRIXC	ASM	6860
MATRIXC	C	482
MATRIXC	CMD	750
MATRIXC	OUT	1901
AICCOM31	ASM	5308

Directory of CH2

ADD4	ASM	702
MULT4	ASM	1150
FIR4	ASM	3016
MATRIXMF	ASM	1369
ADDMFUNC	ASM	556
ADDM	ASM	4179
FIR11	ASM	2595
ADDM	C	393
MATRIXM	C	488
ADDM	CMD	804
ADDM	OUT	1878
MATRIXM	OUT	2053
FIR11L	DAT	190
FIR11X	DAT	242

Directory of CH3

INTERR	ASM	1915
SINEALL	ASM	3093
LOOP	ASM	838
LOOPI	ASM	1076
SINE8I	ASM	1539
PRNOISE	ASM	1829
PRNOISEI	ASM	2214
VECS_DSK	ASM	222
LOOPALL	ASM	8525
LOOPC	ASM	9635
PCLOOP	EXE	212306
C31COM	ASM	3169
DAQ	EXE	250093
DAQ	ASM	9627
LOOPALL	C	2488
AICCOMC	C	2271
LOOPC	C	610
LOOPCI	C	740
C31COM	C	439
C31LOOP	C	873
LOOPALL	CMD	991
LOOPCI	CMD	1029
C31COM	CMD	905
C31LOOP	CMD	905
LOOPALL	OUT	2100
LOOPC	OUT	2146
LOOPCI	OUT	2422

xvi List of Programs/Files on Accompanying Disk

C31LOOP	OUT	2856	BP45	COF	843
C31COM	OUT	1664	BP33	COF	706
PCCOM	CPP	1309	COMB14	COF	273
PCLOOP	CPP	1033	KBP53	COF	2426
DAQ	CPP	1632	RBP53	COF	2424
DAQ	DAT	3117	BP45COEF	H	721
DSKLIB	LIB	143872			
SYMBOLS	H	4190	<i>Directory of CH5</i>		
DSKLIB	H	293	SINEA	ASM	1767
VECS_DSK	OBJ	427	COSINEA	ASM	1833
SINEFM	ASM	2622	IIR6BP	ASM	2335
<i>Directory of CH4</i>			SINEC	C	1971
BP45SIM	ASM	2383	IIR6BPC	C	1057
LP11SIM	ASM	2385	IIR6BPC	CMD	1033
FIRNC	ASM	2147	SINEC	OUT	3986
FIRPRN	ASM	3550	IIR6BPC	OUT	3115
FIRMCF	ASM	2016	AMPLIT	CPP	17889
FIRMC	ASM	16714	BLT	BAS	5363
AICCOMC	C	2233	IIR6COEF	H	639
FIRDMOVE	C	1106	SINECMOD	C	2456
FIRERIC	C	1509	SINESW	ASM	2553
FIRMC	C	713	<i>Directory of CH6</i>		
FIRC	C	1376	TWID128	ASM	2096
FIRMC	CMD	1091	FFT_RL	OBJ	1011
FIRDMOVE	OUT	3018	FFT_RL	ASM	6358
FIRERIC	OUT	3113	TWID8	ASM	221
FIRMC	OUT	3040	FFT128C	C	2498
FIRC	OUT	3089	FFT	C	2294
BP45SIM	DAT	371	SINEGEN	C	540
LP11SIM	DAT	190	TWIDGEN	C	814
FIR	BAT	97	FFT8C	C	680
FIRPROGA	BAS	20237	FFT8MC	C	1124
FIRPROG	BAS	17752	FFT128C	CMD	1033
BP55	COF	1080	FFT128C	OUT	8327
PASS2B	COF	1083	FFT8C	OUT	5837
PASS3B	COF	1088	FFT8MC	OUT	2985
LP55	COF	1095	TWIDDLE	H	8557
BS55	COF	1082	COMPLEX	H	212
LP11	COF	578	FFT8C	CMD	922
HP55	COF	1079	<i>Directory of CH7</i>		
PASS4B	COF	1084	ADAPTP	ASM	4110
STOP3B	COF	1086	NOTCH2W	ASM	4072
BP23	COF	551			
BP41	COF	804			

List of Programs/Files on Accompanying Disk **xvii**

ADAPTER	ASM	3848	SINE4INT	CMD	947
ADAPTC	C	1684	SINE4INT	OUT	4035
ADAPTDMV	C	1600	SINE4C	C	959
ADAPTIVE	C	7783	SINE4C	CMD	959
ADAPTSH	C	1938	SINE4C	OUT	2317
ADAPTTB	C	1639	FIREXT	ASM	11292
ADAPTDMV	CMD	983			
ADAPTSH	CMD	746	<i>Directory of APPB</i>		
ADAPTDMV	OUT	3414	BP45SIMP	ASM	2573
ADAPTSH	OUT	5227	BP45SIMP	DAT	788
ADAPTTB	OUT	4543	DAQ	DAT	3117
SIN312		694	MATBP33	COF	594
SIN312A		776	MAT33	M	523
HCOS312		686	MAT63	M	544
HCOS312A		749	DAQ	M	752
COS312A		798			
DPLUSN		730	<i>Directory of APPC</i>		
DPLUSNA		840	SINEHEX	C	1254
SCDAT		3985	BP45HEX	C	1580
SIN1000		647	TESTMEM	CPP	3690
SHIFT	C	812	C31DLHEX	CPP	2087
ADAPTERC	ASM	4321	SINEHEX	CMD	1015
<i>Directory of CH8</i>			SINHEX30	CMD	448
MR7DSK	ASM	33624	BP45HEX	CMD	1048
FIR8SETS	ASM	10251	BPHEX30	CMD	471
FIRALL	ASM	10311	BP45HEX	OUT	3177
MR10SRAM	ASM	46118	BP45HEX	A0	4717
ALARMGEN	ASM	6053	SINEHEX	OUT	2632
SIM2	C	3803	SINEHEX	A0	3113
FIRALL	CPP	1226	SINEHEX	MAP	4439
FIR8SETP		3057	<i>Directory of APPD</i>		
FIRALL	EXE	212589	LOOPL_CS	ASM	865
EISINE	C	1521	LOOPR_CS	ASM	848
EISINE	CMD	1061	LOOPB_CS	ASM	1015
EISINE	OUT	4307	CSCOM	ASM	6646
VEC_DSK	ASM	215	BP45CS	ASM	2702
VEC_DSK1	ASM	290			
SINE4INT	C	1454			

Index

- Acoustic direction tracker, 242–245
- Adaptive filters:
 - background, 195–197
 - C and TMS320C3x code programming, 199–221
 - adaptive predictor, C code, 206–210
 - adaptive predictor, TMS320C3x code, 215–218
 - C code compiled with Borland C/C++, 200–201
 - interactive adaptation, 200, 202
 - noise cancellation, 203–206
 - notch filter with two weights, 210–215
 - real-time adaptive filter, noise cancellation, 218–221
 - table lookup for delay, adaptive predictor, 208–210
 - implementation, 221–222
 - structure, 197–199
 - TMS320C30 EVM, 250
- Adaptive notch filter:
 - TMS320C30 EVM, 250
 - two-weights, TMS320C3x code, 210–215
- Adaptive predictor:
 - adaptive filter system identification, 198
 - C code programming, 206–210
 - table lookup delay, 208–210
 - TMS320C3x programming code, 215–218
- ADDM.OUT execution, 45
- AIC master clock, sine generation, four points, loading and execution, 10–11
- Aliased sinusoidal waveform, DSK input/output, 52–53
- AM, 156, 163
- AMPLIT.CPP utility program, infinite impulse response (IIR) filters, 149–150
- Analog interface circuit (AIC) chip, DSK input/output, 53–58
 - control, 54–55
 - data configuration, 60
 - desired F_s and filter BW values, 55–58
 - loop/echo with C code, 75–79
 - loop/echo with TMS320C3x code, 65–69
 - sine generation with TMS320C3x code, 62–65
- ARn, TMS320C3x processor, 25
- ARn++(d), TMS320C3x addressing, 25
- ARn++(d)%, TMS320C3x addressing, 25–26
- +ARn(d), TMS320C3x addressing, 25
- ++ARn(d)B, TMS320C3x addressing, 25
- ARn++(IR0), TMS320C3x addressing, 26
- Assembler directives, TMS320C3x processor, 30–32
- Assembly function, TMS320C3x:
 - C code, addition with, 42–45
 - matrix/vector multiplication, 5, 45–47
- Bandpass filter:
 - finite impulse response filters, 111–112
 - infinite impulse response filters:
 - C code, 160
 - TMS320C3x code, 156–160
 - multirate filters, 228–235

- Bandstop filters:
 - finite impulse response (FIR), 117–118
 - infinite impulse response (IIR), 146–148
- Batch file for finite impulse response filters, generic filter, TMS320C3x code, 114–115
- Bilinear transformation (BLT), infinite impulse response (IIR) filters, 143–150
 - AMPLIT.CPP utility program, 149–150
 - BLT.BAS utility program, 148
 - design procedure, 143–145
 - first-order highpass filter, 146
 - first-order lowpass filter, 145–146
 - fourth-order bandpass filter, 147–148
 - second-order bandstop filter, 146–147
 - sixth-order bandpass filter, C code, 160–162
 - sixth-order bandpass filter, TMS320C3x code, 156–160
- Bit reversal, fast Fourier transforms (FFT), unscrambling applications, 178–179
- Blackman window, finite impulse response filters, 105
- BLT.BAS utility program, infinite impulse response (IIR) filters, 148
- Boot loader, sine generation, four points, loading and execution, 9–11
- Borland's C/C++ compiler:
 - adaptive filter programming, 200
 - PC host-TMS320C31 communication, 80–82
- Branch conflicts, TMS320C3x processor, 32
- Branch instructions, TMS320C3x processor, 28–29
- Cache, TMS320C3x processor, 33
- Cascade structure, infinite impulse response (IIR) filters, 140–141
- C code programming:
 - adaptive filters, 199–210
 - adaptive predictor, C code, 206–210
 - C code compiled with Borland C/C++, 200–201
 - interactive adaptation, 200, 202
 - noise cancellation, 203–206
 - table lookup for delay, adaptive predictor, 208–210
- DSK input/output, 60–80
 - loop/echo with AIC data, 75–79
 - loop/echo with interrupt, 79–80
- external/flash memory:
 - FIR bandpass filter, 289
 - sine generation, 287–289
- fast Fourier transform (FFT):
 - eight-point complex, 184–187
 - real-time 128-point FFT, mixed code, 191–193
 - real-valued input, 8-point mixed C and TMS320C3x code, 187–191
- finite impulse response filters:
 - data move, 121–122
 - filter convolutions, 123–124
 - flash memory, 285–289
 - sample shifting, 125–127
- infinite impulse response (IIR) filters:
 - sine generation, 154–156
 - sixth-order bandpass filters, 160–162
 - linking, 44
 - matrix/vector multiplication, 45–46
 - PC host-TMS320C31 communication, 80–87
- Circular buffering, TMS320C3x processor, 30
- Code Explorer, DSK support, 14
- Code Explorer debugger, GO DSP, 265–270
- Computer-aided approximation, finite impulse response filters, 105–106
- Conflicts, TMS320C3x processor, 32–34
- Convolution equation, finite impulse response filters, 98–99
 - TMS320C31 memory organization, 106–108
- Cosine generation, 152, 154
- Crystal CS4216/CS4218 stereo audio codec, DSK input/output, external/flash memory, 88
- C3x tools, DSK support, 14–16
- Data acquisition:
 - DSK communications, C and TMS320C3x code, 85–87
 - MATLAB design, 279–281
- Data move applications, finite impulse response filters, C code, 121–122

- Debugger windows, matrix/vector multiplication, TMS320C3x code, 5–7
- Decimation-in-frequency FFT algorithm, RADIX-2 development, 167–174
 - eight-point FFT, 170–172
 - sixteen-point FFT, 172–174
- Decimation-in-time algorithm, fast Fourier transform (FFT), RADIX-2 development, 174–176
 - eight-point FFT, 176–178
- Difference equations, finite impulse response filters, 95–96
- DigiFilter:
 - DSK support, 14
 - filter design and implementation with, 271–274
- Digital filtering, TMS320C3x code, 37–42
- Digital signal processing (DSP):
 - applications and projects:
 - acoustic direction tracker, 242–245
 - external interrupt for control, 239–242
 - FFT-based security system, 248–249
 - FIR filter banks, 223–228
 - harmonic analyzer, 246–247
 - multirate filters, 228–235
 - pass/fail alarm generator, 235–239
 - speech processing for identification, 248
 - TMS320C30 EVM projects, 250–254
 - development system, 1–2
 - DSK support tools, 2–3
 - support tools, 14–16, 265–281
 - tools testing, 16–17
- Direct form structures, infinite impulse response (IIR) filters, 136–140
- Discrete Fourier transform (DFT):
 - inverse discrete Fourier transform (IDFT), 183
 - RADIX-2 algorithm development, 165–179
 - RADIX-4 algorithm development, 179–182
- Discrete signals, finite impulse response filters, 96–97
- DMA, TMS320C3x processor, 33–34
- DSP Starter Kit (DSK), 2
 - components of, 3
- input/output, 51–89
 - analog interface circuit (AIC) chip, 53–58
 - external/flash memory with 16-bit stereo audio codec, 87–88
 - interrupts and peripherals, 59–60
 - PC host-TMS320C3x communication, 80–87
- Euler's formula, z-transforms of sinusoid, 93–94
- External/flash memory:
 - board construction, 283–286
 - DSK input/output, 87–88
 - memory decode ranges:
 - external memory, 286–287
 - FIR bandpass filter, 289
 - flash memory, 287
 - sine generation, 287–289
- External and flash memory, DSK support, 14–15
- Fast Fourier transform (FFT):
 - background, 165
 - bit reversal for unscrambling, 178
 - MATLAB design:
 - data acquisition, 279–281
 - eight-point FFT, 279
 - RADIX-2 algorithm development, 165–170, 174–176
 - decimation-in-frequency algorithm, 167–174
 - eight-point FFT, 170–172
 - sixteen-point FFT, 172–174
 - decimation-in-time algorithm, 174–179
 - eight-point FFT, 176–178
 - RADIX-4 algorithm development, 179–182
 - sixteen-point FFT, 181–182
 - TMS320C3x/C code programming:
 - eight-point complex, 184–187
 - real-time 128-point FFT, mixed code, 191–193
 - real-valued input, 8-point mixed C and TMS320C3x code, 187–191
- Filter design packages:
 - DigiFilter components, 271–274
 - finite impulse response filters, 106

- Filter development package (FDP), finite impulse response filters, 127–129
- Finite impulse response (FIR) filters:
 - adaptive filter structure and, 197
 - banks, implementation of, 223–228
 - Code Explorer debugging and plotting, 265–270
 - design criteria and techniques, 97–99
 - DigiFilter design and implementation, 271–274
 - discrete signals, 96–97
 - filter design packages, 106
 - filter development package, 127–129
 - generic, TMS320C3x code, 112–115
 - MATLAB design, 275–276, 281
 - multiband FIR filter, 276–278
 - s-plane to z-plane mapping, 94–95
 - window functions, 103–105
- Flash memory, construction, 283–286
- Floating-point tools, C compilation and linkage, matrix/vector multiplication, 12–14
- FM, 70, 88
- Four-channel multiplexer, TMS320C30 EVM, 253
- Fourier series, finite impulse response filters, 98–104
 - window functions, 104–106
- Four values addition, TMS320C3x code, 34–35
- Frequency shift modulation, TMS320C30 EVM, 253

- Generated output frequency, sine
 - generation, four points, loading and execution, 9–10
- Goldwave software:
 - DSK support, 14
 - virtual instrument shareware, 269–271

- Hamming window, finite impulse response filters, 104
- Hanning window, finite impulse response filters, 104–105
- Harmonic analyzer, 246–247
- Highpass filters, infinite impulse response (IIR) filters, 146

- Image processing, TMS320C30 EVM, 250
- Indirect addressing:
 - bit reversal, fast Fourier transforms (FFT), 178–179
 - TMS320C3x processor, 25–26
- Infinite impulse response (IIR) filters:
 - AMPLIT.CPP utility program, 149–150
 - background, 135–136
 - bilinear transformation (BLT), 143–148
 - BLT.BAS utility program, 148
 - cosine generation, recursive equation, TMS320C3x code, 154
 - cosine generation, TMS320C3x code, 154
 - design procedure, 143–145
 - DigiFilter design and implementation, 274
 - filter structures, 136–143
 - cascade structure, 140–141
 - direct form II structure, 137–139
 - direct form II transpose, 139–140
 - direct form I structure, 136–137
 - parallel form structure, 141–143
 - first-order highpass filter, 146
 - first-order lowpass filter, 145–146
 - fourth-order bandpass filter, 147–148
 - MATLAB design, 277–279
 - second-order bandstop filter, 146–147
 - sine generation, recursive equation, C code, 154–156
 - sine generation, TMS320C3x code, 152–154
 - sixth-order bandpass filter, C code, 160–162
 - sixth-order bandpass filter, TMS320C3x code, 156–160
- Input/output:
 - DSK support, 51–89
 - analog interface circuit (AIC) chip, 53–58
 - 16-bit stereo codec, 15, 88
 - interrupts and peripherals, 59–60
 - PC host-TMS320C3x communication, 80–87
 - TMS320C3x and C code programming examples, 60–87
 - 16-bit stereo audio codec design, 291–298
 - FIR filter, 298

- loop programs, 297–298
- TMS320C3x instruction, 28
- Instruction sets, TMS320C3x processor, 26–30, 257–260
 - branch instructions, 28–29
 - circular buffering, 30
 - input/output instructions, 28
 - load and store instructions, 27–28
 - math instructions, 27
 - parallel instructions, 260
 - repeat and parallel instructions, 29–30
- Interactive adaptation, adaptive filter, C code programming, 200–202
- Interactive implementation, FIR filter banks, 225–228
- Interrupts:
 - DSK input/output, 59
 - internal, with TMS320C3x code, 60–62
 - loop/echo using C code, 79–80
 - loop/echo using TMS320C3x code, 69–70
 - sine generation using TMS320C3x code, 70
 - external control, 239–242
- Inverse fast Fourier transform (IFFT):
 - eight-point IFFT, 183
 - MATLAB design, 279
- Kaiser window, finite impulse response filters, 105
- Laplace transform, finite impulse response filters, 91–92, 94
- Least mean square algorithm (LMS), adaptive filter structure, 196–197, 199
- Linking command, 13
- Load and store instructions, TMS320C3x, 27, 257
- Loop/echo programs, 75–80
 - AIC routines using C code, 75–79
 - AIC routines using TMS320C3x code, 65–69
 - crystal 16-bit stereo audio codec, 297–298
 - interrupt using TMS320C3x code, 69–70
- Lowpass filters:
 - finite impulse response (FIR) filter:
 - Fourier series, 101–103
 - TMS320C3x simulation, 108–110
 - infinite impulse response (IIR) filter, 145–146
- Math instructions, TMS320C3x processor, 27
- MATLAB software:
 - data acquisition, plotting and FFT, 279–281
 - eight-point FFT and IFFT, 279
 - filter design, 275–279, 281
 - infinite impulse response (IIR) filter design, 277–279
 - multiband FIR filter design, 276–278
 - real-time FIR/IIR filter design, 281
- Matrix/vector multiplication:
 - C and C-called function, 45–47
 - C code, 11–14
 - TMS320C3x, 4–7, 42
- Memory access:
 - external/flash memory, 286–289
 - organization, 21–25
 - TMS320C3x processor, 33
- Memory conflicts, TMS320C3x processor, 33
- Mixed-mode FIR filter, C and TMS320C3x code, 117–121
- Multiband FIR filter design, MATLAB design, 276–278
- Multirate filter, 228–235
 - design criteria, 228–233
 - external and TMS320C3x code, 287
 - implementation, 233–235
 - TMS320C30 EVM, 250
- Neural networks, TMS320C30 EVM, signal recognition, 253–254
- Noise cancellation:
 - adaptive filter:
 - C code, 203–206
 - real-time filter, TMS320C3x code, 218–221
 - adaptive filter structure, 197
 - TMS320C30 EVM, 250
- Oscilloscopes, DSK support tool, 2

- Parallel form structure, infinite impulse response (IIR) filters, 141–143
- Parametric equalizer, TMS320C30 EVM, 250
- Pass/fail alarm generator, programming for, 235–239
- PC host-TMS320C31 communication:
 - C code, 81–87
 - library support file DSKLIB.LIB, Borland's C/C++ compiler, 81
 - support header file DSKLIB.H, 80–81
- PID controller, TMS320C30 EVM, 251
- Pseudorandom noise:
 - finite impulse response filters, 115–117
 - pass/fail alarm generator, 238–239
- Pseudorandom noise generation, TMS320C3x code, 70–74

- RADIX-2 algorithm development, fast Fourier transform (FFT), 165–179
 - decimation-in-frequency algorithm, 167–174
 - decimation-in-time algorithm, 174–178
- RADIX-4 algorithm development, fast Fourier transform (FFT), 179–182
 - sixteen-point FFT, 181–182
- Read-only memory (ROM), 34
- Real-time adaptive filter, noise cancellation, TMS320C3x code, 218–221
- Recursive equation, infinite impulse response (IIR) filters:
 - cosine generation, 154
 - sine generation, 152–156
- Recursive least squares (RLS) algorithm, adaptive filter structure, 199
- Register conflicts, TMS320C3x processor, 32–33
- Registers:
 - interrupt enable (IE), 261
 - interrupt flag (IF), 263
 - status (ST), 261
- Repeat and parallel instructions, 29–30
- RIDE40, DSK support, 15

- Sample shifting, finite impulse response filters, C code, 125–127
- Sampling frequency (F_s), sine generation, four points, loading and execution, 10–11

- Security system design, fast Fourier transform (FFT), 248–249
- Serial port, DSK input/output, 59
- SigLab, DSK support, 15
- Signal generator, DSK support tool, 2
- Signal recognition, TMS320C30 EVM, neural networks, 253–254
- Signal/spectrum analyzer, DSK support tool, 2
- Sine generation:
 - C code, 154–156
 - DSK input/output:
 - AIC data using TMS320C3x, 62–65
 - interrupt using TMS320C3x, 70
 - external/flash memory, 287–289
 - four points, TMS320C3x code, 8–11
 - TMS320C3x code, 150–154
- 16-bit stereo/audio codec:
 - DSK input/output, 87–88
 - input/output, 291–298
- Speech processing for identification, 248–249
- s*-plane mapping, finite impulse response filters, *z*-plane mapping, 94–95
- Swept frequency response, TMS320C30 EVM, 250

- Taylor series approximation, finite impulse response filters, *z*-transforms of exponential functions, 92–93
- Timers, DSK input/output, 59
- TMS320C30 EVM projects, 250–254
- TMS320C3x:
 - adaptive filters, 210–221
 - adaptive predictor, TMS320C3x code, 215–218
 - real-time adaptive filter, noise cancellation, 218–221
 - addressing modes, 25–26
 - architecture and memory organization, 21–25
 - conflicts, 32–34
 - CPU registers, 22–25
 - crystal 16-bit stereo audio codec:
 - FIR filter, 298
 - loop programs, 297–298
 - developmental background, 19–21
 - digital filtering background, 37–42
 - DSK input/output, 60–74

- loop/echo with AIC routines, 65–69
- loop/echo with interrupt, 69–70
- pseudorandom noise generation, 70–74
- functional block diagram, 22–23
- sine generation with AIC data, 62–65
- sine generation with interrupt, 70
- external memory, multirate filter, 287
- finite impulse response filters, 106–121
 - generic filter specification, 112–115
 - lowpass filter simulation, 108–110
 - mixed-mode filter, C program calling, 117–121
- infinite impulse response (IIR) filters:
 - cosine generation, 152, 154
 - sine generation, 150–154
 - sixth-order bandpass filter, 156–160
- instruction set, 26–30, 257–260
- memory organization, 21–25
- registers, 22–25, 257, 259, 261–263
- TMS320 floating-point DSP assembly
 - language tools, DSK support, 2–3
- Tools testing, DSK experiments, 16–17
- Transfer functions, finite impulse response filters, 97–102
- Two arrays multiplication, TMS320C3x code, 35–37
- Two-weighted notch structure, adaptive filters, 198
- Unscrambling, bit reversal, fast Fourier transforms (FFT), 178–179
- Video line rate analysis, TMS320C30 EVM, 250–251
- Virtual bench, DSK support, 14
- Virtual instrument, shareware utility package, 269–271
- Wait states, TMS320C3x processor, 34
- Window functions, finite impulse response filters, 103–106
- Wireguided submersible, TMS320C30 EVM, 251–253
- z-transform, finite impulse response filters, 91–96
- ZT of exponential function, 92–93
- ZT of sinusoid, 93–94