

TOPICS IN DIGITAL SIGNAL PROCESSING

Digital Signal Processing with the TMS320C25

**RULPH CHASSAING
DARRELL W. HORNING**

TN 911.72
C488

9064403

Digital Signal Processing with the TMS320C25

Rulph Chassaing
Darrell W. Horning



E9064403



A WILEY-INTERSCIENCE PUBLICATION

JOHN WILEY & SONS

New York Chichester Brisbane Toronto Singapore

Copyright © 1990 by John Wiley & Sons, Inc.

All rights reserved. Published simultaneously in Canada.

Reproduction or translation of any part of this work beyond that permitted by Section 107 or 108 of the 1976 United States Copyright Act without the permission of the copyright owner is unlawful. Requests for permission or further information should be addressed to the Permissions Department, John Wiley & Sons, Inc.

Library of Congress Cataloging in Publication Data:

Chassaing, Rulph.

Digital signal processing with the TMS320C25 / Rulph Chassaing,

Darrell W. Horning.

p. cm.

"A Wiley-Interscience publication."

Bibliography: p.

Includes index.

ISBN 0-471-51066-1

I. Signal processing—Digital techniques. I. Horning, Darrell W.

II. Title.

TK5102.5.C473 1989

621.382'2—dc20

90-34031

CIP

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Digital Signal Processing with the TMS320C25

60-420

TOPICS IN DIGITAL SIGNAL PROCESSING

C. S. BURRUS and T. W. PARKS: *DFT/FFT AND CONVOLUTION ALGORITHMS:
THEORY AND IMPLEMENTATION*

JOHN R. TREICHLER, C. RICHARD JOHNSON, Jr., and MICHAEL G. LARIMORE:
THEORY AND DESIGN OF ADAPTIVE FILTERS

T. W. PARKS and C. S. BURRUS: *DIGITAL FILTER DESIGN*

RULPH CHASSAING and DARRELL W. HORNING: *DIGITAL SIGNAL
PROCESSING WITH THE TMS320C25*

Digital Signal Processing with the TMS320C25

To Our Families

Linda, Amber, Otis, Alice, Andrew, and Chloe

Preface

Digital signal processors, made possible through advances in integrated circuits, have added a new element to the environment of digital signal processing (DSP). With this new technology, a student can appreciate the concepts of digital signal processing through real-time implementation of experiments and projects.

This book was developed out of a digital signal processing course and a senior project course taught at Roger Williams College, as well as a digital signal processing laboratory course taught at the University of Bridgeport. The background assumed is an electrical engineering systems course and a knowledge of assembly language programming. Each chapter begins with a theoretical discussion, followed by representative examples. Thirty examples, many with program solution, are included throughout the book; and a variety of students' projects are described in Chapter 9.

This text is intended primarily for senior and first-year graduate students in electrical and computer engineering and as a tutorial for the practicing engineer.

In Chapter 1 we introduce the software development tools. These tools are demonstrated through short programming examples. Chapter 2 covers the architecture and the instruction set of the TMS320C25. Structures and special instructions that are useful in DSP are included. Chapter 3 focuses on input and output (I/O) methods. Two I/O alternatives are presented: The analog interface board (AIB) and the analog interface chip (AIC).

In Chapter 4 we introduce the Z-transform. A digital oscillator example is implemented and can be useful for later experiments and projects. Finite impulse response (FIR) filters are discussed in Chapter 5. Several window functions to improve the characteristics of FIR filters are demonstrated. In Chapter 6 we discuss infinite impulse response (IIR) filters illustrating different structures. The effect of quantization on IIR filters is examined. Two software design tools are covered in conjunction with FIR and IIR filters.

Chapter 7 includes both the decimation-in-time and the decimation-in-frequency fast Fourier transform (FFT). Special instructions for the implementation of the FFT are covered. In Chapter 8 we introduce an intuitive approach to adaptive filtering using the linear combiner structure and the least mean squared (LMS) algorithm. Laboratory examples demonstrate the usefulness of the adaptive approach. Chapter 9 covers a variety of projects, including multirate filtering, modulation techniques, and the FFT. This chapter can be used as a source of experiments, projects, and applications.

We feel that the principles of digital signal processing can best be mastered through interaction in a laboratory setting, with real-time algorithm implementations. This interaction can serve to enhance and enrich a student's understanding of DSP.

This book can be used in a variety of ways, such as:

- 1** For a senior or first-year graduate project course, using Chapters 1 to 7 (Chapter 3 partially) to provide general background, and selected materials from Chapters 8 and 9.
- 2** For a DSP lab course, covering Chapters 1 to 6 (Chapter 3 partially) and selected materials from Chapters 8 and 9. The beginning of the semester can be devoted to short experiments and miniprojects and the remainder of the semester used for a final project.

We would like to thank all our digital signal processing students who have made our project- and laboratory-oriented courses very rewarding; in particular, Peter Martin, for his work on adaptive filtering with the AIC, and Ken Zemlok for his work on the additional AIB channel. The suggestions made by Dr. Kun-Shan Lin of Texas Instruments and Dr. David P. Morgan of Sanders Assoc. are appreciated. The authors would like to acknowledge the National Science Foundation's equipment support through grants CSI-8851272, CSI-8650204, and USE-8851147 and the support of the Roger Williams College Research Foundation. A special thanks to Carol Reineke for typing the manuscript.

*Bristol, Rhode Island
Bridgeport, Connecticut
January 1990*

RULPH CHASSAING
DARRELL W. HORNING

List of Examples and Exercises

Example

- 1.1 A Simple Addition Program
- 1.2 Loop Program
- 2.1 Multiplication by 10
- 2.2 Multiplication by -1.72416
- 2.3 Example Using MACD and RPTK
- 3.1 Loop Program of Chapter 1 Revisited
- 3.2 Triangular Waveform Generation
- 3.3 Pseudorandom Noise Generator Using Interrupt
- 3.4 AIC Loop Program
- 3.5 AIC Two-Input Program
- 4.1 Unit Sequence $x(n) = 1, n \geq 0$
- 4.2 Exponential Sequence $x(n) = e^{-knT}, n \geq 0$
- 4.3 Sinusoidal Sequence $x(n) = \sin(n\omega T)$
- 4.4 Digital Oscillator
- 5.1 FIR Lowpass Filter
- 5.2 Lowpass Filter Using LTD/MPY
- 5.3 Bandpass FIR Filter With 41 Coefficients
- 6.1 Lowpass Filter
- 6.2 Butterworth Bandpass Filter
- 6.3 TMS320C25 Implementation of Fourth-Order Bandpass IIR Filter
Using a Butterworth Design
- 6.4 Scaling a Transpose Second-Order Function
- 7.1 8-Point FFT, Radix-2, Decimation-in-Frequency
- 7.2 16-Point FFT, Radix-4, Decimation-in-Frequency
- 8.1 Two Weights
- 8.2 Adaptation with Basic

- 8.3 Adaptive Echo Cancellation
- 8.4 Signal Identification
- 8.5 Amplitude and Phase Adaptation
- 8.6 Noise Cancellation
- 8.7 Noise Cancellation in Speech Signals

Exercise

- 2.1 Calculation with Overflow
- 3.1 AIC Two-Input Program without Delays
- 5.1 Bandstop Filter Calculation
- 6.1 Coefficient Quantization and FIR Zeros
- 6.2 Coefficient Quantization and IIR Zeros
- 6.3 Simulation of Round-Off Noise
- 6.4 Small-Cycle Oscillation
- 6.5 Overflow Oscillation
- 9.1 Sine-wave Generation 1
- 9.2 Sine-wave Generation 2—Backward Difference Equation
- 9.3 Sine-wave Generation from a Square Wave
- 9.4 Pseudorandom Noise Generator

Contents

Preface	xiii
List of Examples and Exercises	xv
1 A Digital Signal Processing Development System	1
1.1 Introduction	1
1.2 Software Development System	2
1.3 Using the SWDS: A Simple Program	4
1.4 Analog Input/Output Alternatives	6
1.5 Testing the Analog Interface Board	8
1.6 Program Development Using SWDS Tools	10
1.7 Assembling and Downloading Using COFF	13
References	14
2 The TMS320C25 Digital Signal Processor	15
2.1 Introduction	15
2.2 TMS320C25 Architecture	16
2.3 Memory Organization	19
2.4 Addressing Modes	20
2.5 Instruction Set	21
2.6 Fixed-Point Arithmetic	26
	vii

viii Contents

2.7	Representation of Numbers Greater Than 1 and Overflow on the TMS320C25	32
2.8	Floating-Point Representation	34
2.9	Discrete Equation Programming	35
	References	38
3	Input/Output	39
3.1	Introduction	39
3.2	Analog Interface Board	43
3.3	Analog Interface Chip	52
	References	62
4	Introduction to the Z-Transform and Difference Equations	63
4.1	Introduction	63
4.2	Mapping from s -Plane to z -Plane	65
4.3	Solution of Difference Equations Using the Z-Transform	66
4.4	Digital Oscillator from the Inverse Z-Transform	68
4.5	Digital Oscillator Using Data Transfer from PM to DM	71
	References	73
5	Finite Impulse Response Filters	75
5.1	Introduction	76
5.2	Discrete Signals	77
5.3	Linear Time-Invariant Systems	78
5.4	Convolution	78
5.5	Frequency Response	80
5.6	Finite Impulse Response Filters	81
5.7	FIR Implementation Using Fourier Series	82
5.8	Improvement of FIR Filters with Window Functions	92
5.9	Filter Development Package	96
5.10	Digital Filter Design Package	99
	References	103
6	Infinite Impulse Response Filters	105
6.1	Introduction	106
6.2	Realization Forms	107
6.3	IIR Filter Design	111
6.4	Bilinear Transformation	112
6.5	Bilinear Transformation Design Procedure	114
6.6	Scaling and Overflow	121

6.7	Quantization Effects	131
	References	146
7	Fast Fourier Transform	149
7.1	Introduction	149
7.2	Discrete Fourier Transform Algorithm	150
7.3	Development of the FFT Algorithm: Radix-2	151
7.4	Decimation-in-Time	159
7.5	Inverse Fast Fourier Transform	162
7.6	Radix-4 Decimation-in-Frequency	163
	References	171
8	Adaptive Filters	173
8.1	Introduction	173
8.2	Structures	174
8.3	Adaptive Linear Combiner	176
8.4	Performance Function	179
8.5	Searching for the Minimum	181
8.6	Applications of the LMS Algorithm	184
	References	208
9	Digital Signal Processing Applications: Student Projects and Laboratory Exercises	211
9.1	Multirate Filtering: A Shaped Pseudorandom Noise Generator	211
9.2	Modulation Techniques	223
9.3	Pressure Measurement Using a Resonant Wire Technique	234
9.4	Pressure-to-Voltage Conversion	236
9.5	Additional Input Channel for the AIB	239
9.6	Fast Fourier Transform Implementation	244
9.7	An FFT Algorithm with a Fixed Geometry	252
9.8	Real-Time FFT	255
9.9	Implementation of a Second-Order Goertzel Algorithm	259
9.10	Simulation of Hearing Impairments	262
9.11	Digital Touch-Tone Decoder/Generator	264
9.12	Dual-Tone Multifrequency Generation	268
9.13	Digitally Programmable Speech Scrambler	269
9.14	Bandlimiting Baseband Communications	274
9.15	Zero-Crossing-Based Spectrum Analyzer	277
9.16	Correlation of Two Signals	279
9.17	Voice Recognition	281

9.18	Voice Mail	282
9.19	Speaker Identification	287
9.20	Voice Control of a Rhino Robot	288
9.21	Voice Control of a Rhino Robot through Fiber Optics	291
9.22	Graphic Equalizer	294
9.23	Harmonic Scanner	295
9.24	Adaptive Differential Pulse-Code Modulation	296
9.25	Adaptive Notch Filter for Cancellation of Sinusoidal Interference	302
9.26	Guitar Tuning with Adaptive Filtering for Noise Reduction	311
9.27	Musical Tone Octave Generator	313
9.28	Laboratory Exercises	316
	References	318

SWDS Installation and Conversion to COFF 323

Appendix B

B.1	Introduction	333
B.2	Sample Rate Clock	334
B.3	AIB Control Register	335
B.4	D/A Modes of Operation	335
B.5	A/D Modes of Operation	337
B.6	AIB Status Register	338
B.7	Testing the AIB Interface	338

Analog Interface Chip and Macros 341

Appendix D

D.1	FIR Filter Development Package	349
D.2	Magnitude and Phase Response	351
D.3	FIRDP Listing and Support Program	360

Appendix E**ASPI Digital Filter Design Package (FIR FILTERS) 379**

E.1	Introduction	379
E.2	FIR Filter Design	379
E.3	TMS320C25 Code Generation	387
E.4	Frequency Response of a DFDP-Generated FIR Filter	389
E.5	ASPI-Generated Filter Program FIR Filter Program Listing	390

Appendix F**Digital Filter Tools 393**

F.1	Conversion from s to z	393
F.2	Magnitude and Phase Calculation	394
F.3	Bilinear Transformation Program	396

Appendix G**ASPI Digital Filter Design Package (IIR Filters) 400**

G.1	IIR Filter Design	400
G.2	TMS320C25 Code Generation	405
G.3	Frequency Response of a DFDP-Generated IIR Filter	406
G.4	ASPI-Generated IIR Filter Program Listing	407

Appendix H**Multirate Filtering 410**

H.1	Introduction	410
H.2	Random Generator Algorithm	410

Appendix I**Pressure Measurement Using Resonant Wire 423****Appendix J****Bandlimiting Baseband Communications 433**

J.1	Alternative Encoding and Decoding Scheme	433
J.2	Program Listing for Encoding and Decoding Scheme	434

Appendix K**Guitar Tuning 441****Appendix L****Musical Tone Octave Generator 446****Index 455**

1

A Digital Signal Processing Development System

CONCEPTS AND PROCEDURES

- Use of the Software Development System (SWDS), commands and menus
- Creation, assembly, and execution of a TMS320C25 program
- Use of the debugging tools such as modify memory and single step

In this chapter we introduce the use of development tools for real-time signal processing. Those tools include an SWDS, based on the second-generation TMS320C25 digital signal processor, an analog interface board (AIB) and an analog interface chip (AIC). In this chapter we show how to create a source file as well as an object file that can be downloaded into the SWDS and executed. A short example program (LOOP) illustrates how an input signal is brought through the A/D unit into the processor and then sent out to the D/A unit and output filter, where it is reconstructed.

1.1 INTRODUCTION

Signal processing can be split into two areas: nonreal-time signal processing and real-time signal processing. *Real-time processing* means that the processing must keep pace with some external event; *nonreal-time processing* has no such timing constraint. For signal processing, the external event to keep pace with is usually the analog input. This book and digital signal processing (DSP) processors such as the TMS320 family are concerned primarily with real-time signal processing.

The processing speed is often the paramount consideration in choosing design primitives for DSP applications. Figure 1.1 gives a general picture of the relative speeds of various primitives. In progressing from left to right on the diagram, the speed increases, as does the design difficulty. The degree of parallel processing also increases from left to right. On the far right is the fastest technology, fiber optic