Digital Signal Processing System Design: LabVIEW-Based Hybrid Programming

Nasser Kehtarnavaz

Digital Signal Processing System Design: LabVIEW-Based Hybrid Programming

by Nasser Kehtarnavaz
University of Texas at Dallas

With laboratory contributions by Namjin Kim
and Qingzhong Peng

ELSEVIER

Amsterdam • Boston • Heidelberg • London • New York • Oxford
Paris • San Diego • San Francisco • Singapore • Sydney • Tokyo
Academic Press is an imprint of Elsevier

ACADEMIC
PRESS

Academic Press is an imprint of Elsevier
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA
525 B Street, Suite 1900, San Diego, California 92101-4495, USA
84 Theobald's Road, London WC1X 8RR, UK

Copyright © 2008, Elsevier Inc. All rights reserved.

Cover image: supplied by author
Cover Design: Alisa Andreola
Cover Direction: Alisa Andreola

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the publisher.

Permissions may be sought directly from Elsevier's Science & Technology Rights Department in Oxford, UK: phone: (+44) 1865 843830, fax: (+44) 1865 853333, E-mail: permissions@elsevier.com. You may also complete your request online via the Elsevier homepage (http://elsevier.com), by selecting "Support & Contact" then "Copyright and Permission" and then "Obtaining Permissions."

Library of Congress Cataloging-in-Publication Data
Application Submitted

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

ISBN: 978-0-12-374490-6

For information on all Academic Press publications
visit our Web site at www.books.elsevier.com

Printed in the United States of America
08 09 10 9 8 7 6 5 4 3 2 1

Working together to grow
libraries in developing countries

www.elsevier.com | www.bookaid.org | www.sabre.org

ELSEVIER BOOK AID International Sabre Foundation

Contents

Preface xi

What's On the CD-ROM? xiii

Chapter 1: Introduction 1

1.1 Digital Signal Processing Hands-On Lab Courses 2

1.2 Organization 3

1.3 Software Installation 3

1.4 Updates 4

1.5 Bibliography 4

Chapter 2: LabVIEW Graphical Programming Environment 5

2.1 Virtual Instruments (VIs) 5

2.1.1 Front Panel and Block Diagram 5

2.1.2 Icon and Connector Pane 6

2.2 Graphical Environment 7

2.2.1 Functions Palette 7

2.2.2 Controls Palette 8

2.2.3 Tools Palette 8

2.3 Building a Front Panel 9

2.3.1 Controls 9

2.3.2 Indicators 10

2.3.3 Align, Distribute, and Resize Objects 10

2.4 Building a Block Diagram 11

2.4.1 Express VI and Function 11

2.4.2 Terminal Icons 12

2.4.3 Wires 12

2.4.4 Structures 13

2.4.4.1 For Loop 13

2.4.4.2 While Loop 14

2.4.4.3 Case Structure 14

2.5 MathScript 14
2.6 Grouping Data: Array & Cluster 16
2.7 Debugging and Profiling VIs 16
2.7.1 Probe Tool 16
2.7.2 Profile Tool 16
2.8 Bibliography 18

Lab 1: Getting Familiar with LabVIEW: Part I 19

L1.1 Building a Simple VI 20
L1.1.1 VI Creation 20
L1.1.2 SubVI Creation 25
L1.2 Using Structures and SubVIs 29
L1.3 Create an Array with Indexing 33
L1.4 Debugging VIs: Probe Tool 34
L1.5 Bibliography 36
L1.6 Lab Experiments 36

Lab 2: Getting Familiar with LabVIEW: Part II 37

L2.1 Express VIs Versus Regular VIs 37
L2.1.1 Building a System VI with Express VIs 37
L2.1.2 Building a System with Regular VIs 45
L2.2 Hybrid Programming 50
L2.2.1 MathScript Feature 50
L2.2.2 Call Library Function Feature 51
L2.2.2.1 Building C DLL Using MS Visual Studio 51
L2.2.2.2 Calling C DLL from LabVIEW 52
L2.3 Profile VI 54
L2.4 Bibliography 56
L2.5 Lab Experiments 56

Chapter 3: Analog-to-Digital Signal Conversion 57

3.1 Sampling 57
3.1.1 Fast Fourier Transform 60
3.2 Quantization 62
3.3 Signal Reconstruction 65
3.4 Bibliography 67

Lab 3: Sampling, Quantization, and Reconstruction 69

L3.1 Aliasing 69
L3.2 Fast Fourier Transform 76
L3.3 Quantization 80
L3.4 Signal Reconstruction 87
L3.5 Bibliography 90
L3.6 Lab Experiments 91

Chapter 4: Digital Filtering 93
4.1 Digital Filtering 93
4.1.1 Difference Equations 93
4.1.2 Stability and Structure 95
4.2 LabVIEW Digital Filter Design Toolkit 97
4.2.1 Filter Design 97
4.2.2 Analysis of Filter Design 98
4.2.3 Fixed-Point Filter Design 98
4.2.4 Multi-rate Digital Filter Design 98
4.3 Bibliography 98
Lab 4: FIR/IIR Filtering System Design 99
L4.1 FIR Filtering System 99
L4.1.1 Design FIR Filter with DFD Toolkit 99
L4.1.2 Creating a Filtering System VI 101
L4.2 IIR Filtering System 106
L4.2.1 IIR Filter Design 106
L4.2.2 Filtering System 110
L4.3 Building Filtering System Using Filter Coefficients 112
L4.4 Filter Design Without Using DFD Toolkit 113
L4.5 Building Filtering System Using Dynamic Link Library (DLL) 115
L4.5.1 Point-by-Point Processing 115
L4.5.2 Creating DLL in C 118
L4.5.3 Calling DLL from LabVIEW 119
L4.6 Bibliography 120
L4.7 Lab Experiments 121
Chapter 5: Fixed-Point versus Floating-Point 123
5.1 Q-format Number Representation 123
5.2 Finite Word Length Effects 127
5.3 Floating-Point Number Representation 128
5.4 Overflow and Scaling 130
5.5 Data Types in LabVIEW 130
5.6 Bibliography 132
Lab 5: Data Type and Scaling 133
L5.1 Handling Data Types in LabVIEW 133
L5.2 Overflow Handling 135
L5.2.1 Q-Format Conversion 137
L5.2.2 Creating a Polymorphic VI 138

L5.3 Scaling Approach 140
L5.4 Digital Filtering in Fixed-Point Format....................143
L5.4.1 Design and Analysis of Fixed-Point Digital Filtering System....................143
L5.4.2 Filtering System....................146
L5.4.3 Fixed-Point IIR Filter Example....................150
L5.5 Bibliography....................154
L5.6 Lab Experiments....................154
Chapter 6: Adaptive Filtering.................... 157
6.1 System Identification....................157
6.2 Noise Cancellation....................158
6.3 Bibliography....................160
Lab 6: Adaptive Filtering Systems.................... 161
L6.1 System Identification....................161
L6.1.1 Least Mean Square (LMS) Algorithm....................161
L6.1.2 Waveform Chart....................163
L6.1.3 Shift Register and Feedback Node....................163
L6.2 Noise Cancellation....................168
L6.3 Lab Experiments....................173
Chapter 7: Frequency Domain Processing.................... 175
7.1 Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT)....................175
7.2 Short-Time Fourier Transform (STFT)....................176
7.3 Discrete Wavelet Transform (DWT)....................178
7.4 Signal Processing Toolset....................180
7.5 Bibliography....................181
Lab 7: FFT, STFT, and DWT.................... 183
L7.1 FFT Versus STFT....................183
L7.1.1 Property Node....................189
L7.2 DWT....................190
L7.3 Bibliography....................195
L7.4 Lab Experiments....................195
Chapter 8: DSP Implementation Platform: TMS320C6x Architecture and Software Tools.................... 197
8.1 TMS320C6X DSP....................197
8.1.1 Pipelined CPU....................198
8.1.2 C64x DSP....................199

8.2 C6x DSK Target Boards 201
8.2.1 Board Configuration and Peripherals 201
8.2.2 Memory Organization 202
8.3 DSP Programming 203
8.3.1 Software Tools: Code Composer Studio 204
8.3.2 Linking 205
8.3.3 Compiling 205
8.4 Bibliography 206

Lab 8: Getting Familiar with Code Composer Studio 207

L8.1 Code Composer Studio 207
L8.2 Creating Projects 207
L8.3 Debugging Tools 214
L8.4 Bibliography 222

Chapter 9: LabVIEW DSP Integration 223

9.1 Communication with LabVIEW: Real-Time Data Exchange (RTDX) 223
9.2 LabVIEW DSP Test Integration Toolkit for TI DSP 223
9.3 Combined Implementation: Gain Example 224
9.3.1 LabVIEW Configuration 226
9.3.2 DSP Configuration 227
9.4 Bibliography 230

Lab 9: DSP Integration Examples 231

L9.1 CCS Automation 231
L9.2 Digital Filtering 233
L9.2.1 FIR Filter 233
L9.2.2 IIR Filter 238
L9.3 Fixed-Point Implementation 244
L9.4 Adaptive Filtering Systems 248
L9.4.1 System Identification 248
L9.4.2 Noise Cancellation 252
L9.5 Frequency Processing: FFT 254
L9.6 Bibliography 264

Chapter 10: DSP System Design: Dual Tone Multi-Frequency (DTMF) Signaling 265

10.1 Bibliography 268

Lab 10: Hybrid Programming of Dual Tone Multi-Frequency System 269

L10.1 DTMF Tone Generator System 269
L10.2 DTMF Decoder System 273
L10.3 Bibliography 275

Chapter 11: DSP System Design: Software-Defined Radio 277
11.1 QAM Transmitter 277
11.2 QAM Receiver 280
11.2.1 Ideal QAM Demodulation 280
11.2.2 Frame Synchronization 281
11.2.3 Decision-Based Carrier Tracking 281
11.3 Bibliography 284
Lab 11: Hybrid Programming of a 4-QAM Modem System 285
L11.1 QAM Transmitter 286
L11.2 QAM Receiver 289
L11.3 Bibliography 301
Chapter 12: DSP System Design: Cochlear Implant Simulator 303
12.1 Cochlear Implant System 303
12.2 Real-Time Implementation 305
12.2.1 Pre-Emphasis Filter 306
12.2.2 Filterbank for Decomposition and Synthesis 306
12.2.3 Envelope Detection 306
12.2.4 White Noise Excitation 307
12.3 Bibliography 308
Lab 12: Hybrid Programming of Cochlear Implant Simulator System 309
L12.1 Filter Design 310
L12.1.1 Bandpass Filter Design 312
L12.1.2 Lowpass Filter Design 314
L12.2 Real-Time Implementation 315
L12.3 Bibliography 320
Index 321

Preface

The previous edition of this book, titled *Digital Signal Processing System-Level Design Using LabVIEW*, showed how LabVIEW^{TM} graphical programming can be used to build and analyze digital signal processing (DSP) systems in an interactive manner and in relatively shorter times as compared to text-based programming. The motivation for writing the previous edition was derived from the observation that many students taking DSP lab courses, in particular at the undergraduate level, often struggle and spend a fair amount of their time debugging C and MATLAB® codes in lab sessions instead of placing more effort into analyzing and thus understanding signal processing systems.

In this second edition of the book, graphical and textual programming are combined to provide a hybrid programming approach toward achieving a more effective mechanism to build and analyze DSP systems. Textual programming and graphical programming have their own merits and demerits from a programming point of view. In general, math operations are found to be easier to code in textual mode. For example, MATLAB provides a rich set of built-in functions for performing signal processing vector and matrix-based math operations. On the other hand, graphical programming offers an easy-to-build interactive and visualization environment and a more intuitive approach toward building signal processing systems.

In an effort to bring together the preferred features of textual and graphical programming, the labs in the previous edition have been redesigned by incorporating MATLAB code blocks or modules into the LabVIEW graphical programming environment via its new MathScripting feature. In other words, the coding for math-oriented modules is now done using M-files, while interactivity, visualization, and modularity are maintained by using LabVIEW.

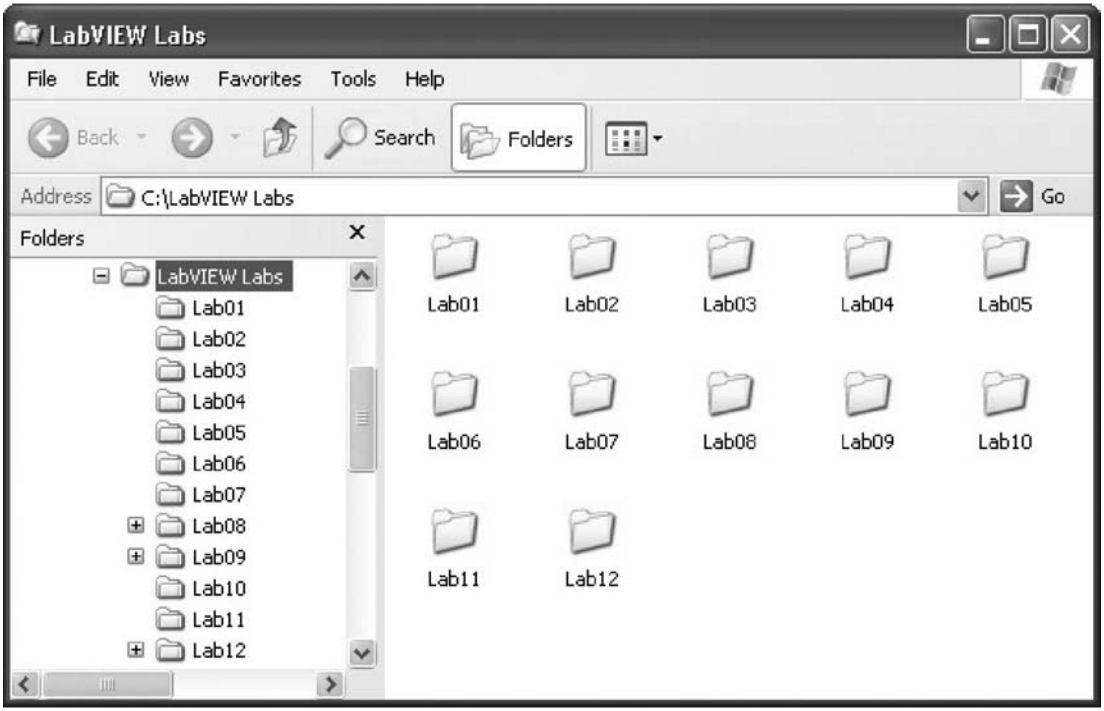
In addition to the hybrid programming approach adopted in this second edition, the labs have been redesigned based on the latest release of LabVIEW (LabVIEW 8.5) at the time of this writing instead of LabVIEW 7.1, which was utilized in the first edition.

I would like to express my appreciation and gratitude to National Instruments, in particular the Academic Marketing Division, for their support of this book.

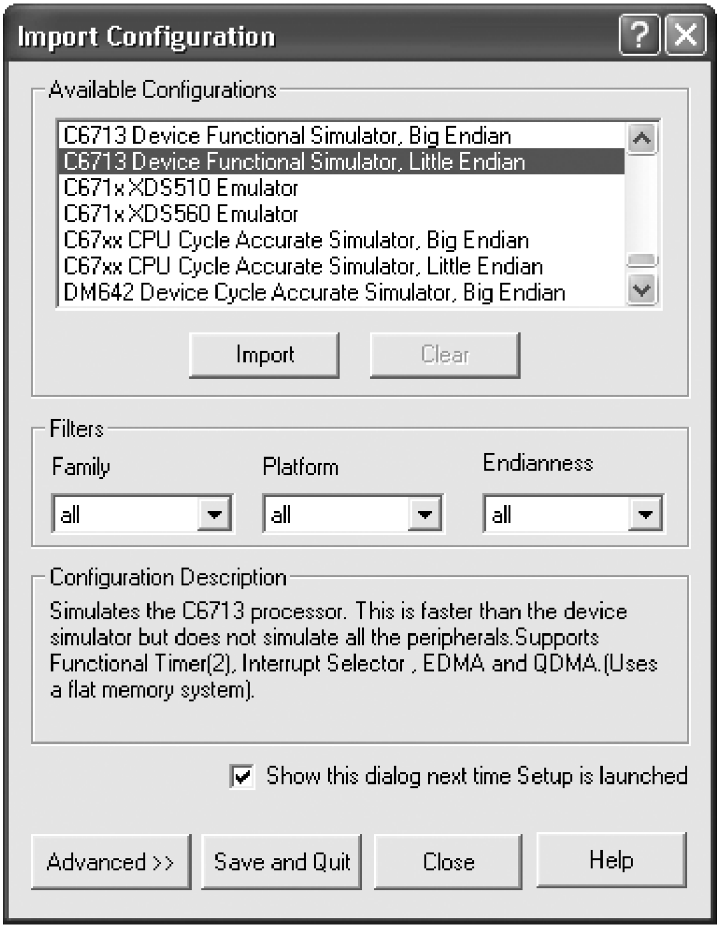
Nasser Kehtarnavaz
December 2007

What's On the CD-ROM?

- The accompanying CD-ROM includes all the lab files discussed throughout the book. These files are placed in corresponding folders as follows:
  - Lab01: Getting Familiar with LabVIEW: Part I
  - Lab02: Getting Familiar with LabVIEW: Part II
  - Lab03: Sampling, Quantization, and Reconstruction
  - Lab04: FIR/IIR Filtering System Design
  - Lab05: Data Type and Scaling
  - Lab06: Adaptive Filtering Systems
  - Lab07: FFT, STFT, and DWT
  - Lab08: Getting Familiar with Code Composer Studio
  - Lab09: DSP Integration Examples
  - Lab10: Hybrid Programming of Dual Tone Multi-Frequency System
  - Lab11: Hybrid Programming of 4-QAM Modem System
  - Lab12: Hybrid Programming of Cochlear Implant Simulator System
- To run the lab files, the National Instruments LabVIEW 8.5 is used and assumed installed. The lab files need to be copied into the folder "C:\LabVIEW Labs\", as shown in the following figure.



- For Lab 8 and Lab 9, the Texas Instruments Code Composer Studio^{TM} (CCStudio) version 3.0 is used and assumed installed in the folder "C:\CCStudio\". The subfolders correspond to the following DSP platforms:
  - DSK 6416
  - DSK 6713
  - Simulator (configured as DSK6713 as shown in the following figure)



Available Configurations

- C6713 Device Functional Simulator, Big Endian
- C6713 Device Functional Simulator, Little Endian
- C671x XDS510 Emulator
- C671x XDS560 Emulator
- C67xx CPU Cycle Accurate Simulator, Big Endian
- C67xx CPU Cycle Accurate Simulator, Little Endian
- DM642 Device Cycle Accurate Simulator, Big Endian

Filters

Family	Platform	Endianness
all	all	all

Configuration Description

Simulates the C6713 processor. This is faster than the device simulator but does not simulate all the peripherals.Supports Functional Timer(2), Interrupt Selector , EDMA and QDMA.(Uses a flat memory system).

☑ Show this dialog next time Setup is launched