Digital Signal Processing and Applications with the C6713 and C6416 DSK

Rulph Chassaing

Worcester Polytechnic Institute

WILEY-INTERSCIENCE

A JOHN WILEY & SONS, INC., PUBLICATION

Digital Signal Processing and Applications with the C6713 and C6416 DSK

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*

RULPH CHASSAING: *DIGITAL SIGNAL PROCESSING WITH C AND THE TMS320C30*

RULPH CHASSAING: *DIGITAL SIGNAL PROCESSING LABORATORY EXPERIMENTS USING C AND THE TMS320C31 DSK*

RULPH CHASSAING: *DSP APPLICATIONS USING C AND THE TMS320C6x DSK*

RULPH CHASSAING: *DIGITAL SIGNAL PROCESSING AND APPLICATIONS WITH THE C6713 AND C6416 DSK*

Digital Signal Processing and Applications with the C6713 and C6416 DSK

Rulph Chassaing

Worcester Polytechnic Institute

WILEY-INTERSCIENCE

A JOHN WILEY & SONS, INC., PUBLICATION

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data:
Chassaing, Rulph.
Digital signal processing and applications with the C6713 and C6416 DSK / by Rulph Chassaing.
p. cm.
Includes bibliographical references and index.
ISBN 0-471-69007-4
1. Signal processing—Digital techniques. 2. Texas Instruments TMS320 series microprocessors.
I. Title.

TK5102.9.C47422 2004
621.382′2—dc22

2004050924

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

Contents

Preface

Digital signal processors, such as the TMS320 family of processors, are used in a wide range of applications, such as in communications, controls, speech processing, and so on. They are used in cellular phones, digital cameras, high-definition television (HDTV), radio, fax transmission, modems, and other devices. 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.

Texas Instruments introduced the TM320C6x processor, based on the very-long-instruction-word (VLIW) architecture. This new architecture supports features that facilitate the development of efficient high-level language compilers. Throughout the book we refer to the C/C++ language simply as C. Although TMS320C6x/assembly language can produce fast code, problems with documentation and maintenance may exist. With the available C compiler, the programmer must "let the tools do the work." After that, if the programmer is not satisfied, Chapters 3 and 8 and the last few examples in Chapter 4 can be very useful.

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 students can appreciate the concepts of DSP through real-time implementation of experiments and projects. The background assumed is a course in linear systems and some knowledge of 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 105 programming examples, most using C code, with a few in assembly and linear assembly code. A list of these examples appears on page xvii. A total of 22 students' projects are also discussed. These projects cover a wide range of applications in filtering, spectrum analysis, modulation techniques, speech processing, and so on.

Programming examples 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. Many assignments are included at the end of Chapters 1–6.

This book can be used in the following ways:

1. For a DSP course with a laboratory component, using parts of Chapters 1–9. If needed, the book can be supplemented with some additional theoretical materials, since its emphasis is on the practical aspects of DSP. It is possible to cover Chapter 7 on adaptive filtering following Chapter 4 on finite impulse response (FIR) filtering (since there is only one example in Chapter 7 that uses materials from Chapter 5). It is my conviction that adaptive filtering should be incorporated into an undergraduate course in DSP.
2. For a laboratory course using many of the examples and experiments from Chapters 1–7 and Chapter 9. The beginning of the semester can be devoted to short programming examples and experiments and the remainder of the semester for a final project. The wide range of sample projects (for both undergraduate and graduate students) discussed in Chapter 10 can be very valuable.
3. For a senior undergraduate or first-year graduate design project course using selected materials from Chapters 1–10.
4. For the practicing engineer as a tutorial and reference, and for workshops and seminars, using selected materials throughout the book.

In Chapter 1 we introduce the tools through three programming examples. These tools include the powerful Code Composer Studio (CCS) provided with the TMS320C6713 DSP starter kit (DSK). It is essential to perform these examples before proceeding to subsequent chapters. They illustrate the capabilities of CCS for debugging, plotting in both the time and frequency domains, and other matters. Appendix H contains several programming examples using the TMS320C6416 DSK.

In Chapter 2 we illustrate input and output (I/O) with the AIC23 stereo codec on the DSK board through many programming examples. Chapter 3 covers the architecture and the instructions available for the TMS320C6x processor. Special instructions and assembler directives that are useful in DSP are discussed. Programming examples using both assembly and linear assembly are included in this chapter.

In Chapter 4 we introduce the z-transform and discuss FIR filters and the effect of window functions on these filters. Chapter 5 covers infinite impulse response (IIR) filters. Programming examples to implement real-time FIR and IIR filters are included. Appendix D illustrates MATLAB for the design of FIR and IIR filters.

Chapter 6 covers the development of the fast Fourier transform (FFT). Programming examples on FFT are included using both radix-2 and radix-4 FFT. In

Chapter 7 we demonstrate the usefulness of the adaptive filter for a number of applications with least mean squares (LMS). Programming examples are included to illustrate the gradual cancellation of noise or system identification. Students have been very receptive to applications in adaptive filtering. Chapter 8 illustrates techniques for code optimization.

In Chapter 9 we introduce DSP/BIOS and discuss a number of schemes (Visual C++, MATLAB, etc.) for real-time data transfer (RTDX) and communication between the PC and the DSK.

Chapter 10 discusses a total of 22 projects implemented by undergraduate and graduate students. They cover a wide range of DSP applications in filtering, spectrum analysis, modulation schemes, speech processing, and so on.

A CD is included with this book and contains all the programs discussed. See page xxi for a list of the folders that contain the support files for the examples and projects.

Over the last 10 years, faculty members from over 200 institutions have taken my workshops on "DSP and Applications." Many of these workshops were supported by grants from the National Science Foundation (NSF) and, subsequently, by Texas Instruments. I am thankful to NSF, Texas Instruments, and the participating faculty members for their encouragement and feedback. I am grateful to Dr. Donald Reay of Heriot-Watt University, who contributed several examples during his review of my previous book based on the TMS320C6711 DSK. I appreciate the many suggestions made by Dr. Mounir Boukadoum of the University of Quebec, Dr. Subramaniam Ganesan from Oakland University, and Dr. David Kozel from Purdue University at Calumet. I also thank Dr. Darrell Horning of the University of New Haven, with whom I coauthored my first book, *Digital Signal Processing with the TMS320C25*, for introducing me to "book writing." I thank al the students at Roger Williams University, the University of Massachusetts at Dartmouth, and Worcester Polytechnic Institute (WPI) who have taken my real-time DSP and senior design project courses, based on the TMS320 processors, over the last 20 years. The contribution of Aghogho Obi, from WPI, is very much appreciated.

The continued support of many people from Texas Instruments is also very much appreciated: Cathy Wicks and Christina Peterson, in particular, have been very supportive of this book.

Special appreciation: The laboratory assistance of Walter J. Gomes III in several workshops and during the development of many examples has been invaluable. His contribution is appreciated.

Rulph Chassaing
Chassaing@msn.com
Chassaing@ece.wpi.edu

List of Examples

Programs/Files on Accompanying CD

A list of the folders included on the accompanying CD is shown below. The folders contain the programs/files for the examples/projects covered in the book.

C:\myprojects

Adaptc	AdaptIDFIR	AdaptIDFIRW	AdaptIDIIR	Adaptnoise
Adaptnoise_2IN	Adaptpredict	Adaptpredict_2IN	Aliasing	Am
BeatDetector	bios_4led	bios_sine_ctrl	bios_sine_intr	bpsk
Code_casm	detect_play	Dft	Dotp4	Dotp4a
dotp4clasm	Dotpintrinsic	Dotpipedfix	Dotpipedfloat	Dotpnp
Dotpnpfloat	dotpopt	Dotpp	Dotppfloat	Dsk6416
Dsk6711	Dtmf	DTMF_Bios_FFT	Dtmf_Bios_Rtdx	echo
Echo_control	Emif_lcd	Encryption	Factclasm	Factorial
FastConvo	Fastconvo_sim	FFT256c	FFT256c_poll	FFTr2
FFTr4	FFTr4_filter	FFTr4_sim	FFTsinetable	Fir
Fir3lp	FIR4types	FIR4ways	FIRcasm	FIRcasmfast
FIRcirc	FIRcirc_ext	FIRinverse	Firpm	FIRPRNbuf
Flash_sine	G722	Graphic_fft	GraphicEQ	Iir
IIR_ctrl	IIRinverse	Loop_intr	loop_poll	Loop_print
loop_stereo	Loop_store	Modulation_schemes		MuLaw
Noise_gen	Noisegen_casm	Notch2	Pll	Psk
Ramp	Ramptable	record	rtdx_lv_filter	rtdx_lv_gain
rtdx_lv_sine	rtdx_matlab_sim	rtdx_matlabFFT	rtdx_matlabFIR	rtdx_vbloop
rtdx_vbsine	rtdx_vc_FFTMatlab	rtdx_vc_FFTr4	rtdx_vc_FIR	rtdx_vc_sine
Scram16k_sw	Scram8k_DMA	Scrambler	Sin1500MATL	Sine_led_ctrl
sine_stereo	Sine2sliders	sine8_buf	sine8_LED	Sinegen_table
Sinegencasm	SinegenDE	Soundboard	speaker_recognition	
Spectrogram	speech_syn	squarewave	sum	Support
Sweep8000	SweepDE	two_tones	Twosum	Twosumfix
Twosumfloat	Twosumlasmfix	Twosumlasmfloat	Viterbi	Readme

1

DSP Development System

- Testing the software and hardware tools with Code Composer Studio
- Use of the TMS320C6713 DSK
- Programming examples to test the tools

Chapter 1 introduces several tools available for digital signal processing (DSP). These tools include the popular Code Composer Studio (CCS), which provides an integrated development environment (IDE), and the DSP starter kit (DSK) with the TMS320C6713 floating-point processor onboard and complete support for input and output. Three examples illustrate both the software and hardware tools included with the DSK. It is strongly suggested that you review these three examples before proceeding to subsequent chapters.

1.1 INTRODUCTION

Digital signal processors such as the TMS320C6x (C6x) family of processors are like fast special-purpose microprocessors with a specialized type of architecture and an instruction set appropriate for signal processing. The C6x notation is used to designate a member of Texas Instruments' (TI) TMS320C6000 family of digital signal processors. The architecture of the C6x digital signal processor is very well suited for numerically intensive calculations. Based on a very-long-instruction-word (VLIW) architecture, the C6x is considered to be TI's most powerful processor.

Digital signal processors are used for a wide range of applications, from communications and controls to speech and image processing. The general-purpose

Digital Signal Processing and Applications with the C6713 and C6416 DSK By Rulph Chassaing
ISBN 0-471-69007-4

digital signal processor is dominated by applications in communications (cellular). Applications embedded digital signal processors are dominated by consumer products. They are found in cellular phones, fax/modems, disk drives, radio, printers, hearing aids, MP3 players, high-definition television (HDTV), digital cameras, and so on. These processors have become the products of choice for a number of consumer applications, since they have become very cost-effective. They can handle different tasks, since they can be reprogrammed readily for a different application. DSP techniques have been very successful because of the development of low-cost software and hardware support. For example, modems and speech recognition can be less expensive using DSP techniques.

DSP processors are concerned primarily with real-time signal processing. Real-time processing requires the processing to keep pace with some external event, whereas non-real-time processing has no such timing constraint. The external event to keep pace with is usually the analog input. Whereas analog-based systems with discrete electronic components such as resistors can be more sensitive to temperature changes, DSP-based systems are less affected by environmental conditions. DSP processors enjoy the advantages of microprocessors. They are easy to use, flexible, and economical.

A number of books and articles address the importance of digital signal processors for a number of applications [1–22]. Various technologies have been used for real-time processing, from fiberoptics for very high frequency to DSPs very suitable for the audio-frequency range. Common applications using these processors have been for frequencies from 0 to 96 kHz. Speech can be sampled at 8 kHz (the rate at which samples are acquired), which implies that each value sampled is acquired at a rate of 1/(8 kHz) or 0.125 ms. A commonly used sample rate of a compact disk is 44.1 kHz. Analog/digital (A/D)-based boards in the megahertz sampling rate range are currently available.

The basic system consists of an analog-to-digital converter (ADC) to capture an input signal. The resulting digital representation of the captured signal is then processed by a digital signal processor such as the C6x and then output through a digital-to-analog converter (DAC). Also included within the basic system are a special input filter for anti-aliasing to eliminate erroneous signals and an output filter to smooth or reconstruct the processed output signal.

1.2 DSK SUPPORT TOOLS

Most of the work presented in this book involves the design of a program to implement a DSP application. To perform the experiments, the following tools are used:

1. *TI's DSP starter kit (DSK)*. The DSK package includes:
   - **(a)** *Code Composer Studio* (CCS), which provides the necessary software support tools. CCS provides an integrated development environment (IDE), bringing together the C compiler, assembler, linker, debugger, and so on.

(b) A board, shown in Figure 1.1, that contains the TMS320C6713 (C6713) floating-point digital signal processor as well as a 32-bit stereo codec for input and output (I/O) support.

(c) A universal synchronous bus (USB) cable that connects the DSK board to a PC.

(d) A 5V power supply for the DSK board.

2. *An IBM-compatible PC*. The DSK board connects to the USB port of the PC through the USB cable included with the DSK package.
3. *An oscilloscope, signal generator, and speakers*. A signal/spectrum analyzer is optional. Shareware utilities are available that utilize the PC and a sound card to create a virtual instrument such as an oscilloscope, a function generator, or a spectrum analyzer.

All the files/programs listed and discussed in this book (except some student project files in Chapter 10) are included on the accompanying CD. Most of the examples (with some minor modifications) can also run on the fixed-point C6416-based DSK. See Appendix H for the appropriate support files along with five illustrative examples. Reference 1 contains examples implemented on the C6711-based DSK (which has been discontinued). A list of all the examples is given on pages xv–xviii.

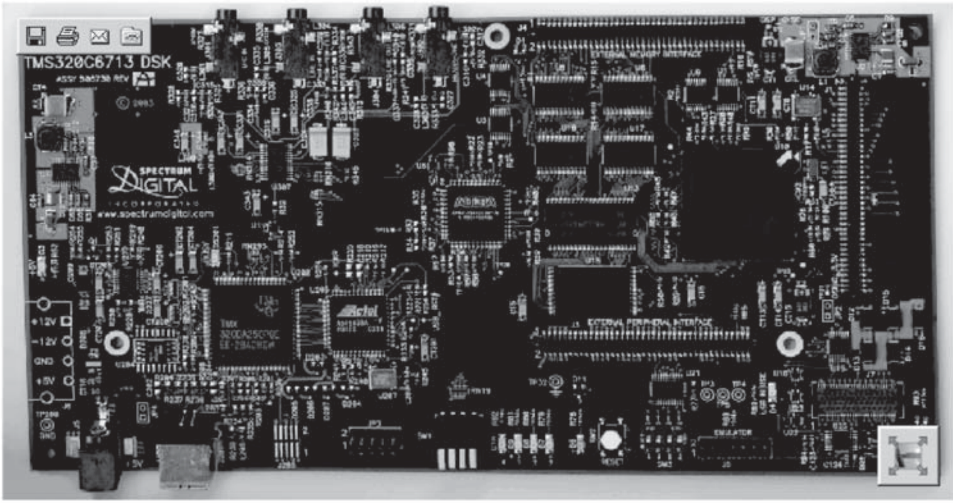
1.2.1 DSK Board

The DSK package is powerful, yet relatively inexpensive ($395), with the necessary hardware and software support tools for real-time signal processing [23–43]. It is a complete DSP system. The DSK board, with an approximate size of 5×8 in., includes the C6713 floating-point digital signal processor and a 32-bit stereo codec TLV320AIC23 (AIC23) for input and output.

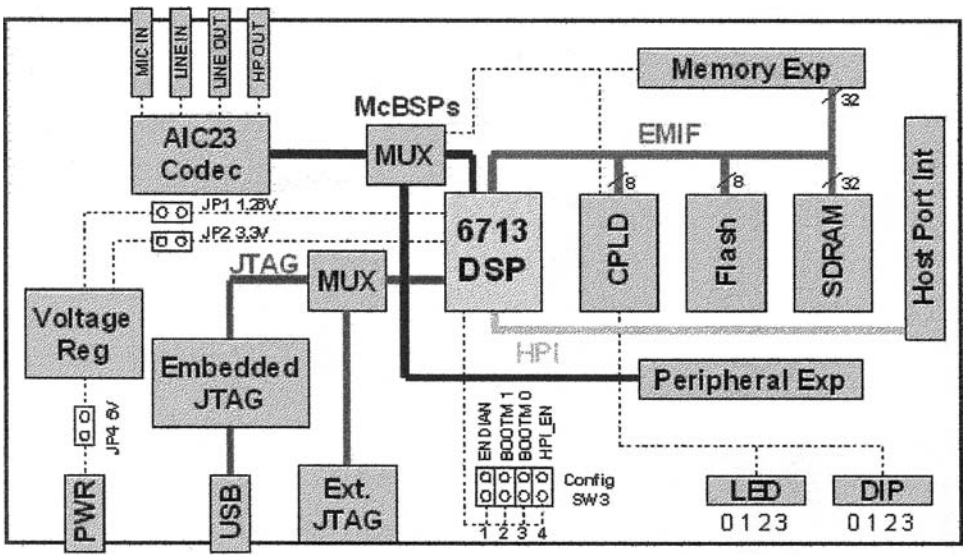
The onboard codec AIC23 [37] uses a sigma–delta technology that provides ADC and DAC. It connects to a 12-MHz system clock. Variable sampling rates from 8 to 96 kHz can be set readily.

A daughter card expansion is also provided on the DSK board. Two 80-pin connectors provide for external peripheral and external memory interfaces. Two project examples in Chapter 10 illustrate the use of the external memory interface (EMIF) with light-emitting diodes (LEDs) and liquid-crystal displays (LCDs) for spectrum display.

The DSK board includes 16 MB (megabytes) of synchronous dynamic random access memory (SDRAM) and 256 kB (kilobytes) of flash memory. Four connectors on the board provide input and output: MIC IN for microphone input, LINE IN for line input, LINE OUT for line output, and HEADPHONE for a headphone output (multiplexed with line output). The status of the four user dip switches on the DSK board can be read from a program and provides the user with a feedback control interface. The DSK operates at 225 MHz. Also onboard the DSK are voltage



(*a*)



(*b*)

FIGURE 1.1. TMS320C6713-based DSK board: (*a*) board; (*b*) diagram. (Courtesy of Texas Instruments)

regulators that provide 1.26 V for the C6713 core and 3.3 V for its memory and peripherals.

Appendix H illustrates a DSK based on the fixed-point processor C6416.

1.2.2 TMS320C6713 Digital Signal Processor

The TMS320C6713 (C6713) is based on the VLIW architecture, which is very well suited for numerically intensive algorithms. The internal program memory is structured so that a total of eight instructions can be fetched every cycle. For example, with a clock rate of 225 MHz, the C6713 is capable of fetching eight 32-bit instructions every 1/(225 MHz) or 4.44 ns.

Features of the C6713 include 264 kB of internal memory (8 kB as L1P and L1D Cache and 256 kB as L2 memory shared between program and data space), eight functional or execution units composed of six arithmetic-logic units (ALUs) and two multiplier units, a 32-bit address bus to address 4 GB (gigabytes), and two sets of 32-bit general-purpose registers.

The C67xx (such as the C6701, C6711, and C6713) belong to the family of the C6x floating-point processors, whereas the C62xx and C64xx belong to the family of the C6x fixed-point processors. The C6713 is capable of both fixed- and floating-point processing. The architecture and instruction set of the C6713 are discussed in Chapter 3.

1.3 CODE COMPOSER STUDIO

CCS provides an IDE to incorporate the software tools. CCS includes tools for code generation, such as a C compiler, an assembler, and a linker. It has graphical capabilities and supports real-time debugging. It provides an easy-to-use software tool to build and debug programs.

The C compiler compiles a C source program with extension `.c` to produce an assembly source file with extension *`.asm`*. The assembler assembles an *`.asm`* source file to produce a machine language object file with extension *`.obj`*. The linker combines object files and object libraries as input to produce an executable file with extension *`.out`*. This executable file represents a linked common object file format (COFF), popular in Unix-based systems and adopted by several makers of digital signal processors [25]. This executable file can be loaded and run directly on the C6713 processor. Chapter 3 introduces the linear assembly source file with extension `.sa`, which is a cross between C and assembly code. A linear optimizer optimizes this source file to create an assembly file with extension `.asm` (similar to the task of the C compiler).

To create an application project, one can "add" the appropriate files to the project. Compiler/linker options can readily be specified. A number of debugging features are available, including setting breakpoints and watching variables; viewing memory, registers, and mixed C and assembly code; graphing results; and monitoring execution time. One can step through a program in different ways (step into, over, or out).

Real-time analysis can be performed using real-time data exchange (RTDX) (Chapter 9). RTDX allows for data exchange between the host PC and the target DSK, as well as analysis in real time without stopping the target. Key statistics and performance can be monitored in real time. Through the joint team action group (JTAG), communication with on-chip emulation support occurs to control and monitor program execution. The C6713 DSK board includes a JTAG interface through the USB port.

1.3.1 CCS Installation and Support

Use the USB cable to connect the DSK board to the USB port on the PC. Use the 5-V power supply included with the DSK package to connect to the +5-V power connector on the DSK to turn it on. Install CCS with the CD-ROM included with the DSK, preferably using the `c:\C6713` structure (in lieu of `c:\ti` as the default).

The CCS icon should be on the desktop as "C6713DSK CCS" and is used to launch CCS. The code generation tools (C compiler, assembler, linker) are used with CCS version 2.x.

CCS provides useful documentations included with the DSK package on the following (see the Help icon):

1. Code generation tools (compiler, assembler, linker, etc.)
2. Tutorials on CCS, compiler, RTDX
3. DSP instructions and registers
4. Tools on RTDX, DSP/basic input/output system (DSP/BIOS), and so on.

An extensive amount of support material (*pdf* files) is included with CCS. There are also examples included with CCS within the folder `c:\C6713\examples`. They illustrate the board and chip support library files, DSP/BIOS, and so on. CCS Version 2.x was used to build and test the examples included in this book. A number of files included in the following subfolders/directories within `c:\C6713` (suggested structure during CCS installation) can be very useful:

1. `myprojects`: a folder supplied only for your projects. All the folders in the accompanying book CD should be placed within this subdirectory.
2. `bin`: contains many utilities.
3. `docs`: contains documentation and manuals.
4. `c6000\cgtools`: contains code generation tools.
5. `c6000\RTDX`: contains support files for real-time data transfer.
6. `c6000\bios`: contains support files for DSP/BIOS.
7. `examples`: contains examples included with CCS.
8. `tutorial`: contains additional examples supplied with CCS.