Filter Design and Implementation Using the TMS320C6x Interfaced with MATLAB

Walter J. Gomes III, Rulph Chassaing
University of Massachusetts Dartmouth

Abstract

This paper describes the design and real-time implementation of FIR and IIR filters using MATLAB interfaced directly with the TMS320C6x (C6x) digital signal processor. An FIR or IIR filter can be readily designed using MATLAB's graphical filter designer SPTOOL to generate a set of coefficients associated with a desired filter's characteristics. These coefficients are included into a generic filter program transparent to the user. Within SPTOOL, the filter's frequency response is plotted on the PC monitor. By modifying SPTOOL, the authors provide a button located on the toolbar to implement the desired filter in real-time on the C6x. Another button is provided to obtain a real-time implementation of the filter on the TMS320C31 on board a DSP Starter Kit (DSK)[1,2]. The authors have developed the support files required to duplicate these results, and they are available to anyone interested. Similar techniques can be developed to interface MATLAB with different digital signal processors.

Introduction

Digital signal processors are currently used for a wide range of applications from communications and controls to speech processing. They are found in cellular phones, fax/modems, disk drives, etc. They continue to be more and more successful because of the availability of low-cost support tools. DSP-based systems can be readily reprogrammed for a different application.

The C6x is Texas Instruments' (TI) highest performance processor based on the Very Long Instruction Word (VLIW) architecture. This type of architecture is very suitable for multitasking. The internal program memory of the C6x is structured as "fetch packets" with a 256-bit instruction and a 256-bit bus. As a result, a word (VLIW) can be fetched every cycle. For example, a C6x with a 200 MHz clock is capable of fetching eight 32-bit instructions, which forms a fetch packet, every 5 ns.

The C6x supports a 32-bit address bus to address 4G bytes, and two sets of sixteen 32-bit registers. It contains eight execution units composed of six ALU's and two multiplier units.

On-chip memory is available as program cache, data cache, and RAM/cache. The exact amount and configuration of memory depends on the specific member of the C6x family of processors. For example, the fixed-point C6211 (which is on-board TI's popular C6x DSK), has two Level-1 (L1)

program and data cache of 4K Bytes each and a Level-2 (L2) 64K Bytes which can be used either as RAM or cache. The C6x internal data memory can be accessed as bytes, 16-bit (half-word) or 32-bit words. A number of support documentation materials are available from TI[3-10] at www.ti.com.

Implementation

This project was tested on the following platforms: the fixed-point C6211-based DSK, the floating-point C6701-based evaluation module (EVM), the fixed-point C6201-based EVM, and the C31-based DSK. The C6xxx are all members of the C6x family of VLIW-based processors. The C31 is a member of the C3x family of floating-point processors based on the Harvard architecture.

The fixed-point C6211-based DSK[9] is TI's lowest cost development system based on the C6x processor. The DSK board includes TI's 16-bit AD535 data converter, which contains an A/D and a D/A. The AD535 on board the DSK has a sampling rate of 8 kHz.

The C6701 is a floating-point processor. Operating at a clock rate of 150 MHz, the C6701 is capable of performing over 1000 million floating-point operations per second (1GFLOPS). The C6701-based EVM includes Texas Instruments' C6701 processor, and the Crystal CODEC chip CS4231A for input and output. The CODEC has a maximum sampling rate of 48 kHz, 16-bit A/D and D/A converters, input (anti-aliasing) and output (reconstruction) filters, all on a single chip[11].

The fixed-point C6201 is pin-compatible with the C6701 floating-point processor. Operating at a clock rate of 150 MHz, the C6201 is capable of executing 1200 million instructions per second (1200 MIPS). The fixed-point C6201-based EVM also contains the CS4231A Crystal CODEC.

A DSK based on the floating-point C6711 is expected during the 2nd quarter of 2000. The C6711 is pin-compatible with the C6211 fixed-point processor.

The EVM and DSK packages include the popular Code Composer Studio (CCS)[12] which provides an integrated development environment (IDE), bringing together the necessary support tools such as the assembler, C compiler, editor, debugger, etc. A real-time data exchange (RTDX) capability with CCS, allows for the transfer of data between the PC host and the processor without stopping the application running on the target.

Filter Design with MATLAB's SPTOOL

Various FIR and IIR filter design techniques can be obtained with MATLAB's graphical filter designer SPTOOL. FIR filter designs based on the Kaiser window, equiripple (REMEZ) and least squares designs; and IIR filter designs using Butterworth, Elliptic, and Chebychev Types I and II are available within SPTOOL. In lieu of using an "M" file to specify the desired filter's characteristics[1], the GUI filter designer, SPTOOL, provides a more convenient alternative. The project was tested with MATLAB[13], version 5.3, along with the signal processing (SP) toolbox, which contains the necessary functions that can be used for different filter designs. Alternatively, the Student Edition of MATLAB[14] which includes the SP toolbox, can also be used.

FIR Filter Design

Figure 1 shows a window screen of the authors' modified version of MATLAB's SPTOOL. The authors' modifications include the three buttons in the upper right section of Figure 1 which illustrate the choices of platforms available for implementing the filter in real-time using either the C6211 DSK, the C31 DSK, or the C6701 EVM. Figure 1 also displays the frequency response of a FIR bandstop filter, with a notch frequency of 790 Hz. The filter's characteristics along with the coefficients of the designed filter with SPTOOL can be retrieved from MATLAB. The file FILTDES.M within MATLAB was modified in order to provide a direct interface to one of the three target processors for real-time implementation.

Within SPTOOL, a "rubber-band" procedure can be accomplished with a mouse to change the characteristics of the desired filter. For example, one can stretch the stopband frequency with the mouse, to specify a desired filter.

To implement the FIR bandstop filter on the C6211 DSK, select the button *C62DSK* in Figure 1. The authors developed an M file function, C62FIRMAT.M, which is called from FILTDES.M when the *C62DSK* button is pressed. After being passed the filter's transfer function and the sample frequency by FILTDES.M, C62FIRMAT.M generates a "coefficient" file containing the filter order, sampling frequency, and properly formatted coefficients.

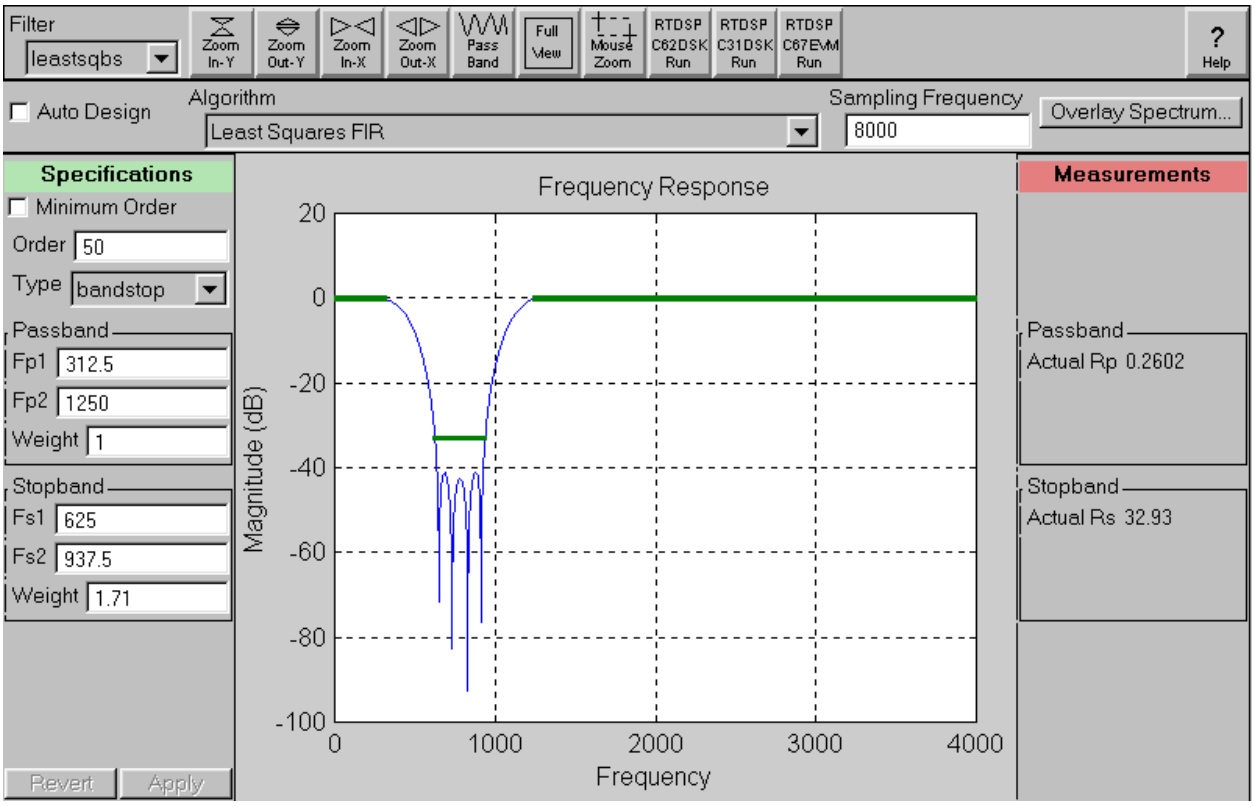

Figure 1. SPTOOL window screen with plot of frequency response of designed FIR filter.

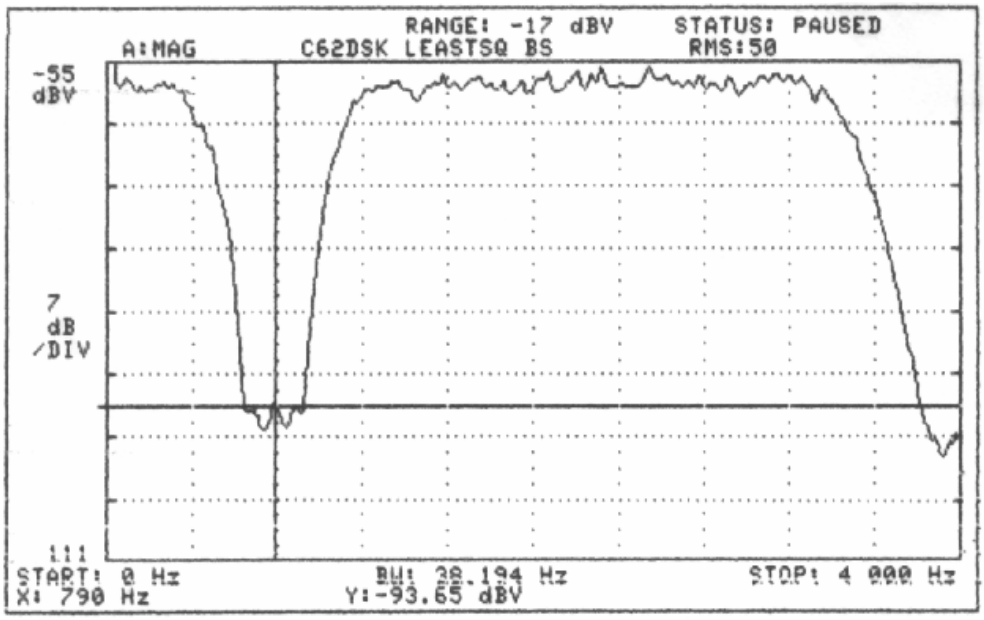

Figure 2. Real-time frequency response of FIR bandstop filter.

This coefficient file is then included in a C-coded generic FIR filter program. The FIR filter program also includes initialization and support files for the CODEC, and for input/output through the serial port. Within C62FIRMAT.M, the C-coded FIR program is compiled, assembled, and linked creating an executable common object file format (COFF) that is downloaded to run on the C6211-based DSK target. Figure 2 shows a plot of the frequency response of the FIR bandstop filter implemented in real-time and displayed on a HP 3561A dynamic signal analyzer. The roll-off frequency close to the Nyquist frequency of 4000 Hz is due to the input anti-aliasing filter on the CODEC.

In lieu of SPTOOL, one can write an "M" file to run on MATLAB[1], specifying the desired characteristics of the filter within the M file. The filter's characteristics are designated in this *M* file with specified magnitude at corresponding frequencies. Several MATLAB functions: *remez(), fir1(),* and *fir2()* are available for FIR filter designs.

All support programs are transparent to the user, since the target processor to implement the desired filter in real-time is selected from the SPTOOL window screen.

IIR Filter Design

Figure 3 shows a plot of the frequency response of an IIR bandpass filter using a Chebyshev design,

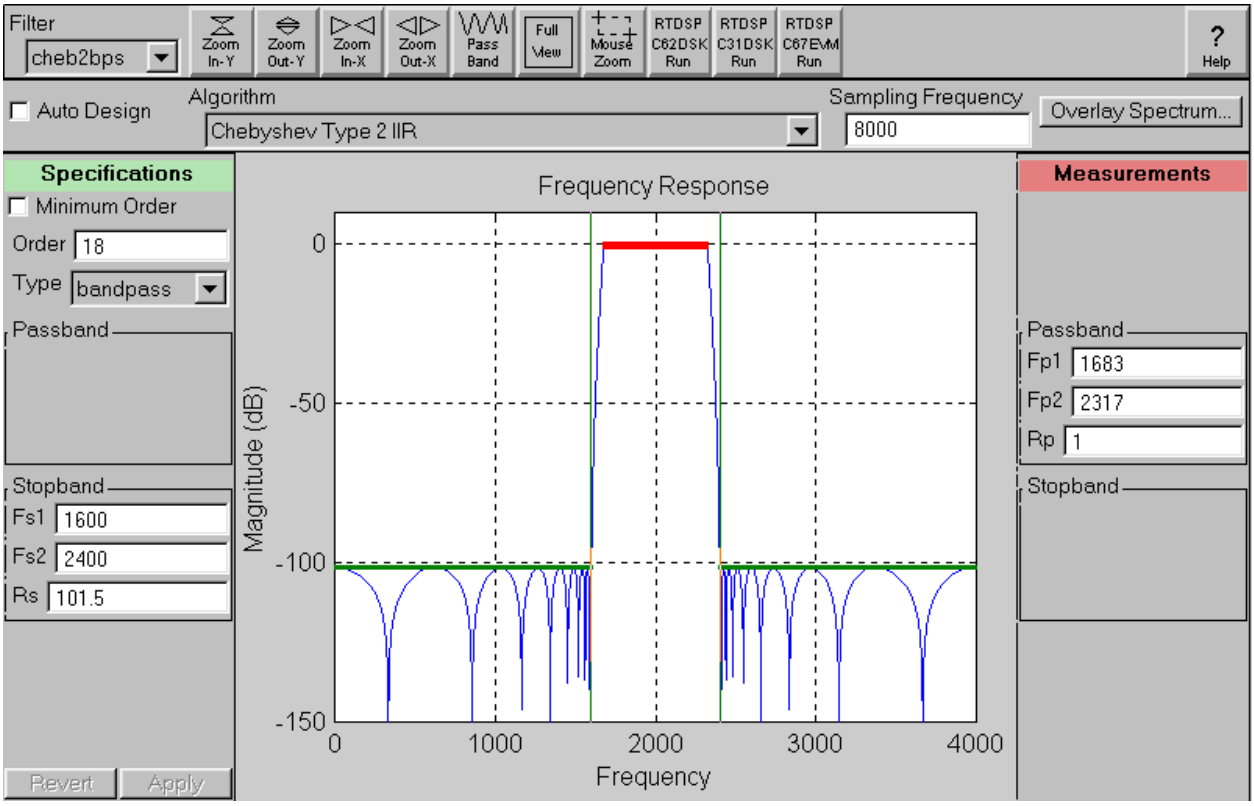

Figure 3. SPTOOL window screen with plot of frequency response of designed IIR filter.

displayed within *SPTOOL*. The filter has a center frequency of 2000 Hz.

The following MATLAB functions: *butter()* for a Butterworth design, *ellip()* for an elliptic design, *cheby1()* and *cheby2()* for Chebyshev Types I and II designs, respectively, are available for IIR filter design. The designed filter's coefficients are contained within *a* and *b* arrays, representing the denominator and numerator coefficients, respectively, of the transfer function of the designed IIR filter.

The authors created an M file function, C67IIR.M, which is called by FILTDES.M, and generates a coefficient file in a format appropriate to be included in a generic C-coded IIR filter program. It uses the following available MATLAB functions to format the IIR transfer function:

a. *tf2zp()* to find the zeros and poles of the transfer function expressed in terms of coefficients *a's* and *b's* associated with the denominator and numerator polynomials, respectively.

b. *zp2sos()* to express the transfer function in terms of second-order sections used by the generic C-coded IIR filter program. One set of coefficients, *a's* and *b's*, are obtained for each section.

Within C67IIR.M, the C-coded IIR filter program is compiled, assembled, and linked, which

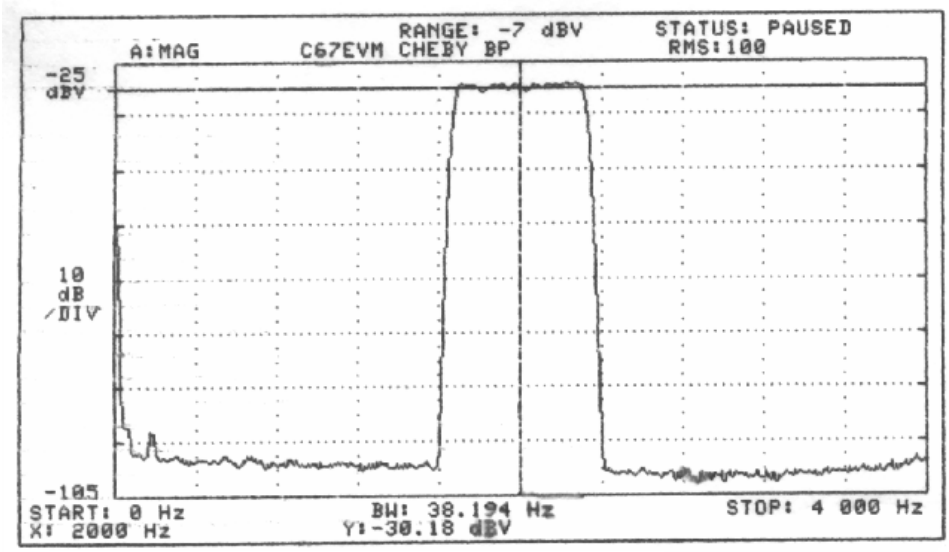

Figure 4. Real-time frequency response of IIR bandpass filter .

creates a COFF executable file that is directly downloaded and run on the selected target processor. Figure 4 shows the real-time frequency response of the IIR bandpass filter obtained from the HP dynamic signal analyzer.

Conclusion

MATLAB's SPTOOL is a powerful tool for the design of both FIR and IIR filters, and can be interfaced directly with the TMS320C6x (or the TMS320C3x) digital signal processor to implement filters in real-time. All support files/programs are transparent to the user while the desired filter is being implemented in real-time. Similar techniques can be developed to interface MATLAB's SPTOOL to other digital signal processors.

Acknowledgement

Grants from the National Science Foundation (NSF) over the last few years provided the support to offer several workshops on Applications in DSP for 113 faculty. Interfacing MATLAB with a digital signal processor evolved during a 1998 NSF-supported workshop given at the University of Massachusetts Dartmouth. The continued support of Texas Instruments is also appreciated.

Bibliography

1. W.J. Gomes III and R. Chassaing, “Real-Time FIR and IIR Filter Design Using MATLAB Interfaced with the TMS320C31 DSK,” in Proceedings of the 1999 ASEE Annual Conference.
2. R. Chassaing, Digital Signal Processing-Laboratory Experiments Using C and the TMS320C31 DSK, J. Wiley, 1999.
3. TMS320C6201/6701 Evaluation Module User’s Guide, SPRU269, Texas Instruments, Inc., 1998.
4. TMS320C62x/C67x CPU and Instruction Set Reference Guide, SPRU189, Texas Instruments Inc., Dallas, Tx., 1998.
5. TMS320C62x/C67x Programmer’s Guide, SPRU198, Texas Instruments, Inc., 1998.
6. TMS320C62x/C67x Technical Brief, Texas Instruments, Inc., Dallas, Tx.
7. TMS320C6x Assembly Language Tools User’s Guide, SPRU186, Texas Instruments, Inc., Dallas, Tx., 1998
8. TMS320C6x Optimizing C Compiler User’s Guide, SPRU187, Texas Instruments, Inc., Dallas, Tx., 1998
9. TMS320C6211 Fixed-point digital signal processor- Advance Information, SPRS073A, Texas Instruments Inc., Dallas, Tx., 1999.
10. TMS320C62xx Peripherals Reference Guide, SPRU190, Texas Instruments, Inc., 1997.
11. CS4231A Parallel Interface, Multimedia Audio Codec, Crystal Semiconductor Corp., Austin, Tx., 1994
12. Code Composer Studio Tutorial, SPRU301, Texas Instruments, Inc., Dallas, Tx., 1999.
13. MATLAB 5.3, The Mathworks, MA, 1998.
14. Student Edition of MATLAB, Prentice Hall, Englewood Cliffs, N. J., 1999.
15. C. H. G. Wright, T. B. Welch, W. J. Gomes III, and Michael G. Morrow, "Teaching DSP Concepts Using MATLAB and the TMS320C31 DSK," in Proceedings of the 1999 ICASSP.

WALTER J. GOMES III is a Senior Electrical Engineer at the Naval Undersea Warfare Center, Newport, RI. He received a BS in Computer Engineering and an MS in Electrical Engineering from the University of Massachusetts Dartmouth. He is a member of IEEE and Eta Kappa Nu. Email: jgomes@ieee.org.

RULPH CHASSAING received the PhD (EE) from the Polytechnic Institute of New York. He is the author of "*Digital Signal Processing-Laboratory Experiments Using C and the TMS320C31 DSK"* and *"Digital Signal Processing with C and the TMS320C30"*, and coauthored with Dr. D. W. Horning *"Digital Signal Processing with the TMS320C25"*, all published by Wiley (1999, 1992, 1990). Email: rchassaing@umassd.edu