Training and Development Module for a Digital Signal Processing Laboratory

Módulo de Capacitación y Desarrollo para un Laboratorio de Procesamiento Digital de Señales

Juan García López[1], César Ortega Coral[2], Luis E. Palafox Maestre[3] y **José J. Esqueda Elizondo[3]**

[1]Instituto Politécnico Nacional

Centro de Investigación y Desarrollo de Tecnología Digital

Ave. Del Parque #1310 Mesa de Otay, C.P. 22320 Tijuana, B.C. México

Ph. +52(664)623-1344

jgarcia1@citedi.mx

[2]Universidad Iberoamericana Campus Tijuana

Departamento de Ingeniería y Ciencias

Ave. Centro Universitario #2501 Playas de Tijuana C.P. 22200 Tijuana, B.C. México

Ph. +52(664)630-1577 Ext. 308

cesarortega@hotmail.com

[3]Universidad Autónoma de Baja California

Facultad de Ciencias Químicas e Ingeniería

Calzada Tecnológico #14418 Mesa de Otay C.P. 22390 Tijuana, B.C. México

Ph. +52(664)682-1033 Ext. 5805

{lepalafox,jjesqueda}@uabc.mx

Abstract

A digital signal processing training module (DSPTM) is constructed around a TMS320C6211 starter kit. The DSP target system is enclosed in a portable case with the necessary accessories: A host port interface connector, power supply, function generator, audio amplifier, speaker, input/output audio sockets and selector switches. DSP demonstration software is bundled with custom windows 98/XP virtual instrumentation programs that run on a host personal computer (PC). An installation and exercise manual accompanies the trainer. The developed host-target software and suggested experiments assist students in understanding data acquisition, DSP and image processing fundamentals. The open source nature of this software helps those who need to develop their own real-time algorithms. This DSPTM is an inexpensive way of reducing the gap between simulation and real-time implementation of DSP algorithms.
Keywords: Digital Signal Processing Laboratory, Image Processing, Training Module, DSP, Virtual Instrumentation, System Identification.

Resumen

Un módulo de capacitación en procesamiento digital de señales (DSPTM) es construido en torno al paquete de inicio del TMS320C6211. El sistema DSP destino se encuentra en un estuche portátil con todos sus accesorios necesarios: un conector hacia la interfaz del puerto anfitrión (HPI), fuente de poder, generador de funciones, amplificador de audio, bocinas, conexiones de entrada/salida de audio y conmutadores de selección. Se incluye software de demostración de DSP así como programas personalizados para Windows 98/XP sobre instrumentación virtual que pueden ser ejecutados en una computadora personal (PC). Manuales de instalación y de ejercicios fueron desarrollados para el instructor. El software desarrollado y los experimentos sugeridos asisten a los estudiantes en el entendimiento de adquisición de datos, procesamiento digital de imágenes y principios de procesamiento de imágenes digitales. La naturaleza de "código abierto" de este software ayuda a aquellos que tienen la necesidad de desarrollar sus propios algoritmos en tiempo real. El DSPTM es una manera poco costosa de cerrar la brecha entre la simulación y la implementación de algoritmos de procesamiento digital de señales.

Palabras Clave: Laboratorio de Procesamiento Digital de Señales, Procesamiento de Imágenes, Módulo de Capacitación, DSP, Instrumentación Virtual, Identificación de Sistemas.

1   Introduction

Typical DSP undergraduate and graduate courses use simulation environments to make their point. In recent years, some articles have dealt with the issue of learning signal processing fundamentals through real-time implementations [1, 2, 3, 4], with the natural objective of reducing the gap between theory and practice. Different types of commercial processing platforms have emerged with that in mind, some are relatively easier to use than others.

On the other hand, from a 'first world' perspective, universities with engineering studies are able to spend considerable amounts of money in acquiring the necessary hardware-software for their labs. But in developing countries (that accounts for most of the world), in both public and private institutions, high quality practical education in engineering is limited due to budget constraints. Although there is an effort by some DSP suppliers to donate equipment and software, usually it is with an agenda of using students as testers of 'beta' versions, which in turn can report or fix their bugs. In this respect, one of the goals of the DSPTM is to offer a relatively inexpensive and portable DSP laboratory for Mexican public technological institutes and universities. Our choice of using the Texas Instruments 32 bit DSK320C6211 (that stands for DSP Starter Kit based on the TMS320C6211 fixed point processor) was made on account of its high throughput (1200 million instructions per second, MIPS). It comes with a compatible C language compiler-debugger, and ultimately this choice was made because of its low price.

Although it has been suggested worthy of comparison between the fixed point DSK6211 with its equivalent floating point platform, i.e. DSK6711, our limited budget made the choice easier on which platform to use for the DSPTM.

The developed exercises and the host-target shareware programs, included with the DSPTM, make an effort to speed up student comprehension of DSP concepts along with the host-target programs that are considered freeware.

2   The Training Module Construction and Software Development

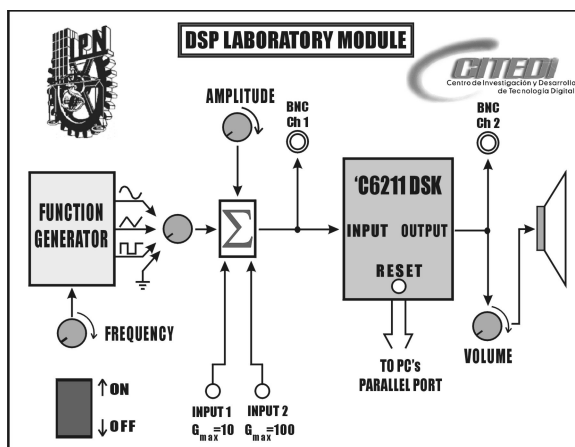Figure 1 illustrates the module front panel and fig. 2 shows its external connections.
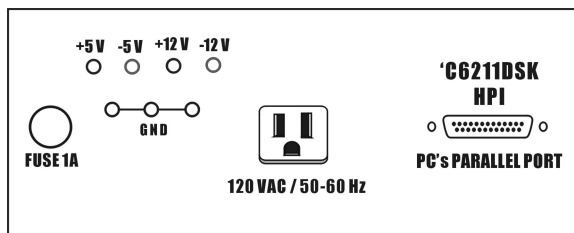

Fig. 1. Control panel

Fig. 2. External connections

The control panel shows a block diagram of the module interior: A function generator, a three input summing amplifier, followed by the DSK location in the signal path. An audio speaker with volume control is incorporated as a possible output for DSP experiments. Additional selector switches grant the user, control of the input signal waveform, amplitude and frequency. A couple of BNC connectors are included to permit oscilloscope observations of the input/output signals.

An 8038 integrated circuit (IC), resistors and capacitor make up the inexpensive function generator. Capable of outputting three basic shapes: sine, square and triangle waveforms. That range up to 5 V of peak amplitude and a maximum operating frequency of 5 kHz.

On the other hand, the TMS320C6211 starter kit, or DSK6211, is a low cost 32 bit general purpose platform for development, analysis and testing of DSP algorithms and applications. It runs at a 150 MHz frequency and is able of executing up to 1200 MIPS. A simplified DSK6211 block diagram is shown in fig. 3.
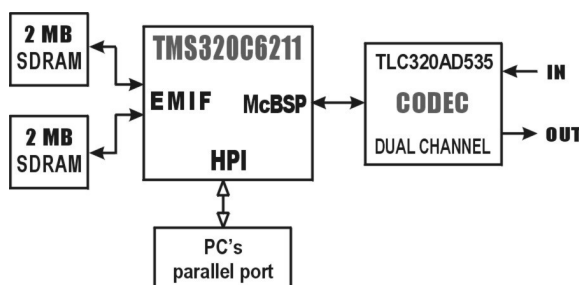

Fig. 3. DSK6211 block diagram

This target system comes with 4 Mbytes of external memory, a 16 bit stereo CODEC (with a fixed 8 kHz sampling rate) and a parallel host port interface (HPI). A fully assembled training module prototype is shown in fig. 4.
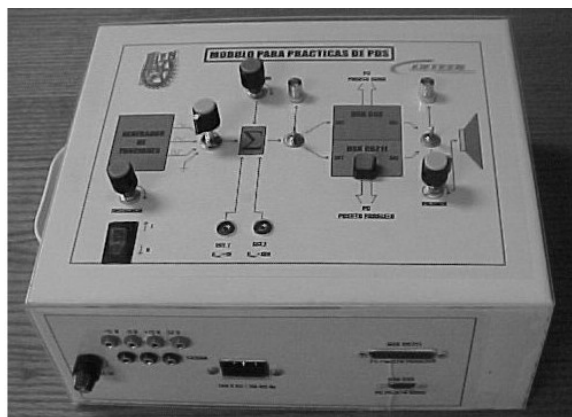

Fig. 4. DSP training and development module

In order to develop application software and be able to download it to the DSK memory, host communication is need. This is done connecting the target HPI to one of the PC parallel ports. For this, Texas Instruments (TI) distributes the DSK6211 along with TMS320C62xx C and assembly language code generation and source debugger, Windows 98/XP drivers, host PC and DSP software APIs (application programming interfaces), example applications with source code, and various utility programs. What the people of TI don't mention is the great deal of bugs that come with the software tools. Figure 5 shows a block diagram of the low-level host to target communications driver API that TI has developed, in the form of a Windows 98/XP DLL (i.e. Dynamic Link Library) object.
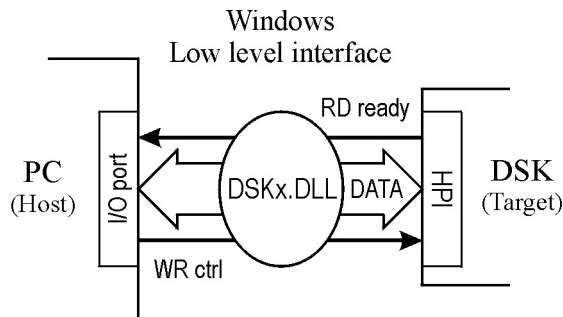

Fig. 5. Host - target software interface

To shorten development time of application software, Go-DSP Inc. (a TI affiliate) came up with an Integrated Development Environment (IDE) called Code Composer Studio (CCS). We use CCS in parts of the target software development process. The CCS includes the Code Generation Tools and the Real Time Data Exchange (RTDX) API that provides support for configuring, building, debugging, tracing, analyzing and editing the target's programs.

3   Demonstrations

A compact disc (CD) accompanies this training module that includes an installation and experiments manual, along with demonstration shareware. The manual describes the TI software and driver setup process. After which, the installation of our custom software is explained. Once both the TI software and the included shareware is correctly installed, any undergraduate or graduate student can follow the instructions on the use of the trainer and suggested DSP experiments.

Besides developing target software using CCS and its command line compiler-linker (finding and working around some of the bugs). The custom host programs we developed were done in different stages and with various PC programming tools like: Visual C++, Visual Basic (for which an intermediate DLL HPI had to be generated) and C++ Borland Builder.

The custom host-target shareware includes: A non parametric system identification, finite impulse response (FIR) digital filter implementations, virtual instrumentation programs, and software for loading and processing images.
For example, fig. 6 shows a virtual oscilloscope running on a host PC in a Win 98/XP format. In this case, a 1 volt 500 Hz sine wave is acquired and plotted with the corresponding sample lines. Making variations of the waveform frequency (as suggested in the manual) permits beginners firsthand understanding of the Nyquist theorem and the 'alias' phenomenon. Figure 7 shows the host spectrum analyzer plotting the sine wave in a logarithmic scale, where its frequency position is clearly observed. In this case a Hamming window is used and the discrete Fourier transform (DFT) is obtained with a radix 2 fast Fourier transform (FFT) scheme running on the DSK6211.
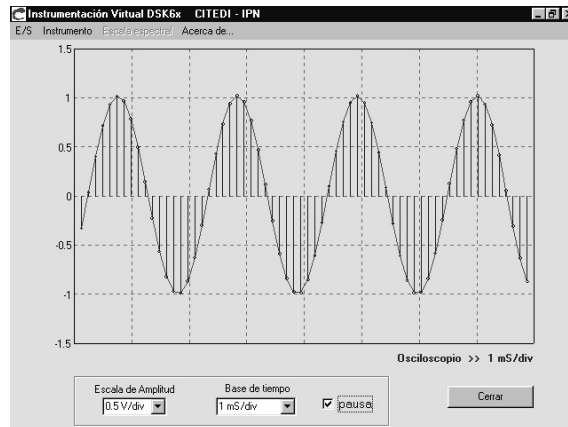
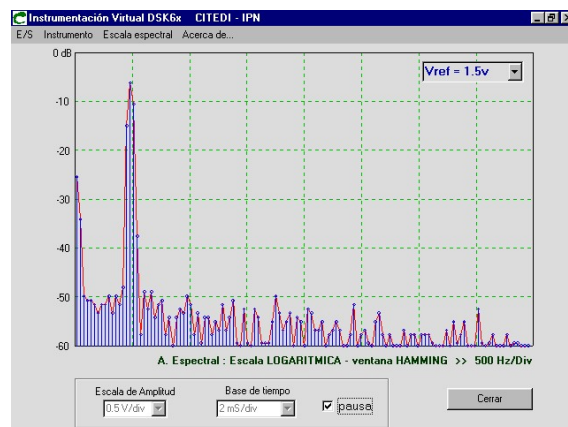**Fig. 6.** Sine wave plotted on the virtual oscilloscope

Fig. 7. Spectrum of the sine wave using a log scale and a Hamming window

These two virtual instruments replace the use of expensive conventional equipment. A drawback is the limited bandwidth (less than 4 kHz) of observable signals. This is due to the CODEC's sampling frequency of 8 kHz.

Figures 8 and 9 show the binarization and edge detection samples of the classic Lena image using a 128 threshold value [5]. The DSP processing time is negligible given the 1200 MIPS executed by the DSP.

However, the processed image took about three to four seconds to show up on the PCs monitor. The main reason is due to the limited bandwidth of the host parallel port (less than 10 kbits/s) and the large amount of the image data passing through it.

Fig. 8. Binarization of the Lena image

Fig. 9. Edge detection

Figure 10 shows the non parametric identification of a passband active filter with central frequency = 460 Hz, bandwidth = 200 Hz, on a linear frequency scale. The theoretical transfer function is given by eq. (1):

$$H(s) = \frac{-1.5 \times 10^{-3} s}{1.5 \times 10^{-6} s^2 + 3 \times 10^{-3} s + 11} \tag{0}$$

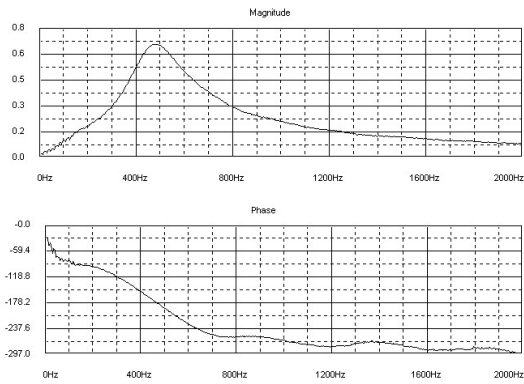The system is well identified for frequencies larger than 100 Hz.

Fig. 10. Non parametric identification of an active passband filter

Figure 11 shows the behavior of a 44th order low pass digital (FIR) filter, designed using the Fourier series method and a Hamming window with a 0.2 normalized cutoff frequency.

It is worth mentioning that currently our training module is being used by Mexican graduate students, developing real-time virtual instrumentation, adaptive filtering and image processing algorithms.
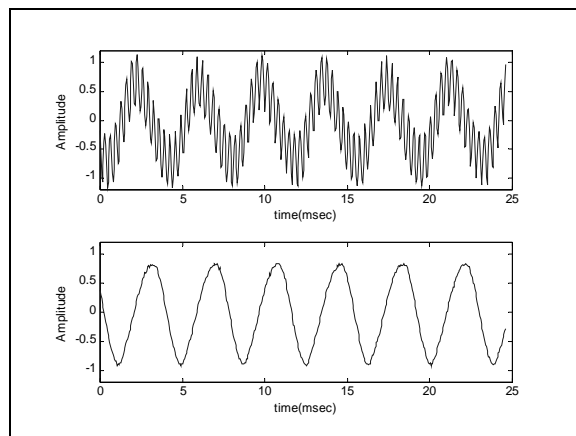

Fig. 11. Low pass filter

4   Concluding Remarks

This training module is useful to demonstrate and implement algorithms taught in Digital Signal Processing courses. Using the DSK6211 as the main block of our module lowers the total cost. Although use of Code Composer Studio reduces the time to develop applications, one can use the free of charge command line C compiler to develop DSP software for the DSK6211. In this case both methods were used. This module is a good example of technology convergence between hardware, DSP algorithms and embedded systems programming.

Acknowledgments

References

1. **Nahvi M**., "Design-Oriented DSP Courseware", IEEE Signal processing Magazine, pgs. 30-36, October 1992.
2. **Ingle V.K., Proakis J.G.,** "A DSP Based on Lecture/Lab Integration", IEEE Signal Processing Magazine, pag. 25-29 October 1992.
3. **Kann D.L., López J.G.,** Houts R.C., "Integrating a DSP system into a Telecommunications Laboratory", Frontiers in Education Conference Proceedings, 1988.
4. **Moreno L. et. al.,** "Digital Signal Processors for a Signal Processing Laboratory", IEEE Transactions on Education, Vol. 42, No.3, pgs. 192-199, August 1999.
5. **Palafox L.E.,** "Técnicas de Procesamiento Digital de Imágenes utilizando un Procesador Digital de Señales", CITEDI-IPN, pgs. 80-82, May 2002.
6. **Embree P.M., Kimble B.,** "C Language Algorithms for Digital Signal Processing", New Jersey, Prentice Hall 1991.
7. **Esqueda J.J.,** "Identificación no paramétrica de sistemas lineales e invariantes con el tiempo usando un Procesador Digital de Señales", pgs. 413-417, Electro 2002, ISSN 1405-2172, October 2002.
8. **Jamieson L.H.,** "The (R)Evolution in Signal-Processing Education", IEEE Signal Processing Magazine, pgs. 8-10, March 1998.
9. **McConnell D.,** "Digital Signal Processing An introduction", Circuit Cellar Ink, pgs. 30-41, February/March 1990.
10. **Texas Instruments,** "TMS320C5x User's Guide", 1997.
11. **Texas Instruments**, "TMS320C62x/C67x Programmer's Guide", 1998.
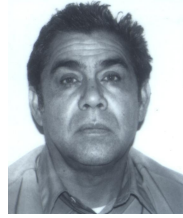
Juan García López. *He received the B.S.E.E. degree from the Universidad Nacional Autónoma de México (UNAM) in 1973. He obtained the Master and Ph.D. degree from the University of Alabama, USA, in 1982 and 1988 respectively. Areas of interest: Digital and analog electronics, telecommunications, control theory and DSP. Email: jgarcial@citedi.mx*

César Ortega Corral. *He was born in Chihuahua, México. In 1997 obtained his B.S. in Electronics and Communications Engineering degree at the Universidad IberoAmericana (UIA, Tijuana Campus). Finished a M.S. in Digital Systems during spring 2001 at the Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI) of the Instituto Politécnico Nacional (IPN). Tijuana, México. Currently, Ortega works in the private sector as an Information Technologies Consultant developing database frontend programs and web applications. Also, M.S. César Ortega is part of the faculty of the Undergraduate Engineering and Science Department at the UIA Tijuana. Email: cesarortega@hotmail.com*

Luis E. Palafox Maestre. *He obtained a Bachelor's degree in Computer Science from the Universidad Autónoma de Baja California (UABC) Tijuana campus, Mexico, in 1997. Obtained a M.S. degree in 2002 at the Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI) of the Instituto Politécnico Nacional (IPN) of México. Areas of interest: computer network development, object oriented programming and image compression. Email: lepalafox@uabc.mx*
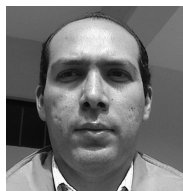
José J. Esqueda Elizondo. *He obtained his Bachelor's degree in Electronics from The Instituto Tecnológico de Ciudad Madero, Mexico, in 1999. Obtained a M.S. degree in 2001 at the Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI) of the Instituto Politécnico Nacional (IPN) of México. Since 2001, profesor at the Universidad Autónoma de Baja California (UABC) Tijuana Campus, México. Areas of interest: System Identification and Digital Signal Processing. Email:* jjesqueda@uabc.mx