1. Introduction

The field of digital signal processing (DSP) has experienced a considerable growth in the last two decades primarily due to the availability and advancements in digital signal processors (also called DSPs). Nowadays, DSP systems such as cell phones and high-speed modems have become an integral part of our lives.

In general, sensors generate analog signals in response to various physical phenomena that occur in an analog manner (i.e., in continuous time and amplitude). Processing of signals can be done either in the analog or digital domain. To perform the processing of an analog signal in the digital domain, it is required that a digital signal is formed by sampling and quantizing (digitizing) the analog signal. Hence, in contrast to an analog signal, a digital signal is discrete in both time and amplitude. The digitization process is achieved via an analog-to-digital (A/D) converter. The field of DSP involves the manipulation of digital signals in order to extract useful information from them.

There are many reasons why one wishes to process an analog signal in a digital fashion by converting it into a digital signal. The main reason is that digital processing allows programmability. The same processor hardware can be used for many different applications by simply changing the code residing in memory. Another reason is that digital circuits provide a more stable and tolerant output than analog circuits - for instance, when subjected to temperature changes. In addition, the advantage of operating in the digital domain may be intrinsic. For example, a linear phase filter or a steep-cutoff notch filter can be easily realized by using digital signal processing techniques, and many adaptive systems are achievable in a practical product only via digital manipulation of signals. In essence, digital representation (0s and 1s) allows voice, audio, image, and video data to be treated the same for error-tolerant digital transmission and storage purposes.

1.1 Digital Signal Processing Hands-on Lab Courses

Nearly all electrical engineering curricula include DSP courses. DSP lab or design courses are also being offered at many universities concurrently or as follow-ups to DSP theory courses. These hands-on lab courses have played a major role in student understanding of DSP concepts. A number of textbooks, e.g. [1-3], have been written to provide the teaching materials for DSP lab courses. The programming language used in these textbooks consists of either C, MATLAB®, or Assembly, i.e. text-based programming. In addition to these programming skills, it is becoming important for students to gain experience in a block-based or graphical (G) programming language or environment for the purpose of designing DSP systems in a relatively short amount of time. Thus, the main objective of this book is to provide a block-based or system-level programming approach in DSP lab courses. The block-based programming environment chosen is LabVIEW.

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) is a graphical programming environment developed by National Instruments (NI) which allows high-level or system-level designs. It uses a graphical programming language to create so called Virtual Instruments (VI) blocks in an intuitive flowchart-like manner. A design is achieved by integrating different components or subsystems within a graphical framework. LabVIEW provides data acquisition, analysis, and visualization features well suited for DSP system-level design. It is also an open environment accommodating C and MATLAB codes as well as various applications such as ActiveX and DLLs (Dynamic Link Libraries).

This book is written primarily for those who are already familiar with signal processing concepts and are interested in designing signal processing systems without needing them to be proficient C or MATLAB programmers. After familiarizing the reader with LabVIEW, the book covers a LabVIEW-based approach to generic experiments encountered in a typical DSP lab course. It brings together in one place the information scattered in several NI LabVIEW manuals to provide the necessary tools and know-how for designing signal processing systems within a one-semester structured course. This book can also be used as a self-study guide to design signal processing systems using LabVIEW.

In addition, for those interested in DSP hardware implementation, two chapters in the book are dedicated to executing selected portions of a LabVIEW designed system on an actual DSP processor. The DSP processor chosen is TMS320C6000. This processor is manufactured by Texas Instruments (TI) for computationally intensive signal processing applications. The DSP hardware utilized to interface with LabVIEW is the TI's C6416 or C6713 DSK (DSP Starter Kit) board. It should be mentioned that since the DSP implementation aspect of the labs (which includes C programs) is independent of the LabVIEW implementation, those who are not interested in the DSP implementation may skip these two chapters. It is also worth pointing out that once the LabVIEW code generation utility becomes available, any portion of a LabVIEW designed system can be executed on this DSP processor without requiring any C programming.

1.2 Organization

The book includes twelve chapters and twelve labs. After this introduction, the LabVIEW programming environment is presented in Chapter 2. Lab 1 and Lab 2 in Chapter 2 provide a tutorial on getting familiar with the LabVIEW programming environment. The topic of analog to digital signal conversion is presented in Chapter 3 followed by Lab 3 covering signal sampling examples. Chapter 4 involves digital filtering. Lab 4 in Chapter 4 shows how to use LabVIEW to design FIR and IIR digital filters. In Chapter 5, fixed-point versus floating-point implementation issues are discussed followed by Lab 5 covering data type and fixed-point effect examples. In Chapter 6, the topic of adaptive filtering is discussed. Lab 6 in Chapter 6 covers two adaptive filtering systems consisting of system identification and noise cancellation. Chapter 7 presents frequency domain processing followed by Lab 7 covering the three widely used transforms in signal processing: fast Fourier transform (FFT), short time Fourier transform (STFT), and discrete wavelet transform (DWT). Chapter 8 discusses the implementation of a LabVIEW-designed system on the TMS320C6000 DSP processor. First, an overview of the TMS320C6000 architecture is provided. Then, in Lab 8, a tutorial is presented to show how to use the Code Composer Studio™ (CCStudio) software development tool to achieve the DSP implementation. As a continuation of Chapter 8, Chapter 9 and Lab 9 discuss the issues related to the interfacing of LabVIEW and the DSP processor. Chapters 10 through 12, and Labs 10 through 12, respectively, discuss the following three DSP systems or project examples that are fully designed via LabVIEW: (i) dual tone multi-frequency (DTMF) signaling, (ii) software-defined radio, and (iii) MP3 player.

1.3 Software Installation

LabVIEW 8.0, which is the latest version at the time of this writing, is installed by running *setup.exe* on the LabVIEW 8.0 Installation CD. Some lab portions use the LabVIEW toolkits 'Digital Filter Design', 'Advanced Signal Processing', and 'DSP Test Integration for TI DSP'. Each of these toolkits can be installed by running *setup.exe* located on the corresponding toolset CD.

If one desires to run parts of a LabVIEW designed system on a DSP processor, then it is required to install the Code Composer Studio software tool. This is done by running *setup.exe* on the CCStudio CD. The most updated version of CCStudio at the time of this writing, i.e. CCStudio 2.2, is used in the DSK related labs.

The accompanying CD includes all the files necessary for running the labs covered throughout the book.

1.4 Updates

Considering that any programming environment goes through enhancements and updates, it is expected that there will be updates of LabVIEW and its toolkits. To accommodate for such updates and to make sure that the labs provided in the book can still be used in DSP lab courses, any new version of the labs will be posted at the website http://www.utdallas.edu/~kehtar/LabVIEW for easy access. It is recommended that this website is periodically checked to download any necessary updates.

1.5 Bibliography

[1] N. Kehtarnavaz, *Real-Time Digital Signal Processing Based on the TMS320C6000*, Elsevier, 2005.

[2] S. Kuo and W-S. Gan, *Digital Signal Processors: Architectures, Implementations, and Applications,* Prentice-Hall, 2005.

[3] R. Chassaing, *DSP Applications Using C and the TMS320C6x DSK*, Wiley Inter-Science, 2002.