DSP Implementation of Communication Systems

Raqibul Mostafa*, Nishith D. Tripathi+, and Jeffrey H. Reed*

* Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia
ramostaf@vt.edu; reedjh@vt.edu
+ Nortel Networks, 2201 Lakeside Blvd., Richardson, Texas

Abstract

The course "DSP Implementation of Communications Systems" provides undergraduate and graduate students with practical experience with communications concepts and algorithms using a TMS320C30 Evaluation Module (EVM) platform. This course is taught on a regular basis in the Electrical and Computer Engineering (ECPE) department at Virginia Polytechnic Institute and State University. The course is based on an effective combination of classroom lectures and laboratory experiments that renders itself to encouraging student creativity. The classroom lectures provide material on communications systems from a theoretical perspective. The theoretical discussion is supplemented by laboratory experiments, which provide an exposure to digital implementation by a DSP. The course concludes with a design project that requires students to propose and implement individual communications sub-systems. Past student projects include: Spread Spectrum Modems, LPC-10 Vocoders, Rake Receiver, Speech Recognition, Microphone Sensor Array, Angle-of-Arrival Measurement System, and a Multi-user CDMA System. The course presents a unique opportunity for students to experience the thrill of creating a practical implementation of communication sub-systems.

1.0 Introduction

Virginia Tech received an NSF grant as part of the NSF's Combined Research and Curriculum Development (CRCD) program for the development of a trilogy of wireless communications courses. These courses include DSP Implementation of Communications Systems, Simulation of Communication Systems, and Analog and Digital Communications. There is a similar lab at the University of Maryland, emphasizing wire-line modems, which has been a phenomenal success in teaching DSP-based systems. The course here started as an experimental independent study version in the fall of 1996 and the first regular class was offered in the spring of 1997. The course meets a growing need in industry to have people talented in DSP/Communications. Many of the lab experiments are based on a long running software radio work sponsored by DARPA. The educational value of this course is immense as the students obtain hands-on experience of practical implementation by using DSP.

This course involves presentation of communication theories in classroom lectures and hardware implementation through laboratory experiments and concludes with a final design project. The design project includes a formal oral presentation and a hardware demonstration, which is based on an industry-style design review with peer evaluations. The reference material for the course is [1], [2] and handouts.

TI played a major supporting role by donating the lab equipment. Students in this course are encouraged to participate in TI's global DSP design contest. In 1997 students from this class won the America's competition in DSP design sponsored by TI.

2.0 Course Description

The course "DSP Implementation of Communication Systems" is based on three main components:

- Classroom lecture
- Lab session
- Project design

Each of these items is described in the following subsections. The project design is presented in a separate section.

2.1 Class lectures:

The class lectures are designed to provide a theoretical background of the communication systems. A particular communication sub-system is discussed, the necessary theory developed, and algorithms are presented to illustrate the behavior of the system to the students. The lectures get the students familiarized with the system before hardware implementation. Different aspects of the system are assessed from a performance point of view.

The class lectures are designed to cover important sub-systems of communication systems. Lecture materials include:

Lecture 1: Overview of the DSP Microprocessor Development Systems and the TMS320C30
Lecture 2: Modulation Generation Using DSP
Lecture 3: Carrier Removal with DSP
Lecture 4: Symbol Timing and Synchronization
Lecture 5: Spread Spectrum Code Synchronization
Lecture 6: Equalization

Lecture 1: *Overview of the DSP μP Development Systems and the TMS320C30*
This lecture introduces to the students the realm of DSP and provides an overview of the development systems based on DSP. Comparison and contrast is drawn between a fixed-point and a floating-point processor. DSP from major vendors are introduced and the specifications in terms of speed, and cost are compared. DSP architecture, including standard and Harvard architectures, are presented. Pipelining issues are discussed with reference to examples. Then TMS320C30 and the evaluation module are discussed with respect to hardware components, peripherals, addressing, and instructions in a detailed manner.

Lecture 2: *Modulation Generation Using DSP*
This lecture provides generation of signals with digital methods and presents different techniques in this regard. These include digital filtering, phase accumulation techniques and phase locked loops. Pulse shaping and IIR oscillator are discussed in the digital filtering category. Direct digital synthesis (DDS) concepts are discussed in phase accumulation techniques.

Lecture 3: *Carrier Removal with DSP*
Carrier recovery and demodulation are taught in this lecture. Both coherent and non-coherent techniques are discussed. Discussions on coherent techniques include phase lock loop (PLL) with both analog and digital versions, Costas loop and Tanlock loop. Different performance metrics related to PLLs such as, cycle slipping and acquisition time are discussed.

Lecture 4: *Symbol Timing and Synchronization*
Lecture four presents symbol timing and synchronization. Types of synchronization (and usual order) can be described as (i) carrier synchronization, (ii) baud synchronization, and (iii) frame synchronization. Different classes of bit synchronizers are presented in this lecture that include nonlinear-filter synchronizer, data transition tracking loop (DTTL) and early-late bit synchronizer.

Lecture 5: *Spread Spectrum Code Synchronization*
This lecture begins with an overview of spread spectrum systems. It reviews the principle, receiver, spreading code generation (maximal length m-sequence) of such a system. Code synchronization or code acquisition is presented next including different acquisition techniques based on different search algorithms.

Lecture 6: *Equalization*
Lecture six introduces the concept of channel distortion and equalization as the counter-measure for this distortion. It illustrates different types of equalizer types, structures, and cost functions. Adaptive algorithms to implement equalization are presented. Both the linear and non-linear versions of the algorithms are discussed.

2.2 Lab experiments
The classroom lectures provide the necessary background to understanding and developing a communication sub-system. With this background the students enter the development phase where they learn hands-on experience working on a DSP platform. The laboratory section of this course takes place in the existing Control and DSP Lab at Virginia Polytechnic Institute and State University. The lab hosts six stations or PCs that have TMS320C30 EVM boards installed in them. The students work in groups of two and occupy the stations on a first-come first-serve or reservation basis. The lab is accessible to the students for most of the time through regular TA hours. Experience has shown that the blocking probability is quite low with the existing timing arrangement.

There are six regular experiments that span for most of the semester. The lab experiments are designed to include both simulation and DSP implementation. The simulation segment helps the students to understand the algorithms and expected performance. After a thorough investigation of the system by simulation, the students proceed to implement the system on the C30 platform.

The hardware implementation segment of the experiment requires the student to implement and test the system to ensure that pre-defined performance specifications are met. The students use standard C as the DSP language but they are encouraged to use Assembly language wherever necessary. In addition, they learn about benchmarking to compare the efficiency of their codes and the algorithms.

The six lab experiments are described briefly here.

Lab 1: *Introduction to TMS320C30 EVM board*
The lab experiment starts with an introduction to the evaluation module board and its different components. The students are introduced to the fundamentals of DSPs and their programming and the usage of different manuals in this regard. They are required to carry out short exercises that involve a walk through the compiler and its different components.

Lab 2: *Generation of a BPSK signal*
This lab experiment involves displaying a BPSK signal on an oscilloscope. The concepts of BPSK modulation and pseudo-noise (PN) sequence are briefly reviewed. Different ways of transferring data from the evaluation module (EVM) to the outside world (e.g., an oscilloscope) are taught in this experiment. The major objectives of this lab experiment are to:

(i) generate a PN sequence and verify the properties of the PN sequence,

(ii) create a signal source to be used in following labs, and

(iii) obtain a BPSK signal using the generated PN sequence and display it on an oscilloscope using three different methods of data transfer: polling method, interrupt method and direct memory access (DMA) method.

Lab 3: *Filtering: FIR & IIR*
This lab involves implementation of FIR and IIR digital filters. Efficient ways of performing convolution are introduced. Relevant theory of FIR and IIR filters is reviewed. Specific structures of the filters to be implemented are discussed. The usage of assembly functions in a C code is explained. An introduction to the usage of assembly language to perform major calculations is given.

The experiment starts with the design of FIR and IIR bandpass filters using MATLAB and verification of specifications of the filters. Then using the filter coefficients obtained from MATLAB program, the frequency response of the filters is evaluated on the C30 platform. The performance obtained from hardware implementation is compared with the theoretical values.

Lab 4: *Carrier Recovery*
This lab involves implementation of a carrier recovery system. A modified form of a phase-locked loop (PLL), called Costas loop, is introduced. The concept of Hilbert transform, important for the Costas loop implementation, is discussed. The major objectives are to:

(i) write a MATLAB code to design a filter that performs Hilbert transform, and verify the properties of the Hilbert transform filter.

(ii) design a Costas loop.

(iii) write C and assembly programs using the Hilbert transform filter coefficients and the Costas loop parameters to develop a carrier recovery system.

Lab 5: *Code Synchronization*
This lab involves implementation of a code synchronization technique. Code synchronization for spread spectrum systems is briefly explained. The matched filter synchronization technique is discussed. The students generate a PN code and proceed to implement a matched-filter based synchronization technique.

Lab 6: *Symbol Timing Recovery*
This lab involves implementation of a symbol timing recovery technique called early-late gate synchronizer. The students simulate the technique in Matlab and implement in the hardware.

Lab 7: *Equalization*
This lab involves implementation of several adaptive equalization techniques. The structure and implementation of an LMS adaptive equalizer are explained to the students. They implement the equalizer in software in Matlab. Once they study the convergence behavior and MSE performance of the equalizer, they implement this on the C30 platform.

2.3 Project description

The students are required to participate in a design project at the close of the semester. In this phase, the students demonstrate a hardware implementation of any communication sub-system and give an oral presentation. The experience that the students get through the regular lab experiments is put to full use as they propose a project of their choice and implement it. The project effort is initiated in the early part of the semester when the proposed projects are critically reviewed. At the end of the regular lab experiments, the students begin work on the project part of the course.

Previous student projects have demonstrated the implementation of many techniques or algorithms that are more sophisticated than the regular experiments. The C30 EVM is somewhat limited in terms of sophisticated wideband applications and the students get around this problem by working on a reduced data rate system. Sometimes the system requires front-end processing that is hard to design within the limited timeframe so the external signals are generated within the EVM itself.

The projects span a wide-range of applications. The range is from a model-based FM demodulator to a complete spread spectrum system. Different communication subsystems that were demonstrated include user authentication in a cellular system, different types of demodulators, rake receivers, and transmit diversity. System level demonstrations include spread spectrum modems, speech recognition system, QAM receivers, and microphone array system. A few of these are described briefly as follows.

The speech recognition system involves the implementation of a speech recognition system using Hidden Markov Models (HMM). This project is computationally intensive and involves a complex system. The project can be illustrated with the following flow diagram as shown in Figure 1. This system is based on training phase to prepare word models in a vocabulary, testing of the speech processing system with real time voice, speech recognition programming, and communication between host PC and EVM board. The project has demonstrated successfully the recognition of single-speaker and two-speaker case. In a two-speaker scenario, the same word spoken by both users is identified or the word along with the speaker is identified. This project may be made a regular lab in the future.

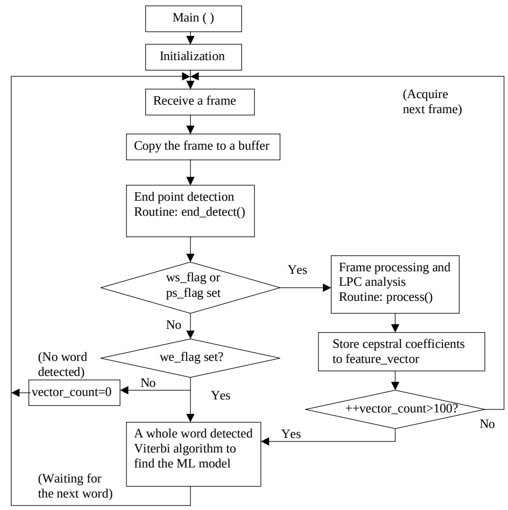

Figure 1. Flow diagram for Speech recognition system

The microphone array project demonstrates beamsteering with a two-microphone array. The microphone array will be able to maintain near unity gain for sounds coming from a given beam direction while diminishing sounds from other directions. It also demonstrates noise cancellation for a noise signal of known frequency from a given direction. Figure 2 shows the block diagram of the array front-end implementation.

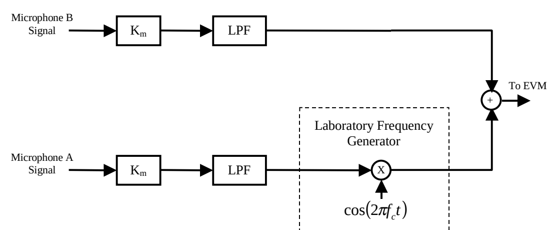

Figure 2. Block diagram for Microphone array system

The QAM receiver design develops a 16-ary QAM transmitter and receiver combination of a basic QAM radio. The transmitter converts a binary input signal into a QAM constellation signal, applies pulse shaping to match the channel passband, and modulates data in analog form. The receiver tracks the sampled QAM signal, equalizes for channel distortion, and extracts the original binary signal. The design principles follow from [1]. Figure 3 shows the block diagram of the receiver section.

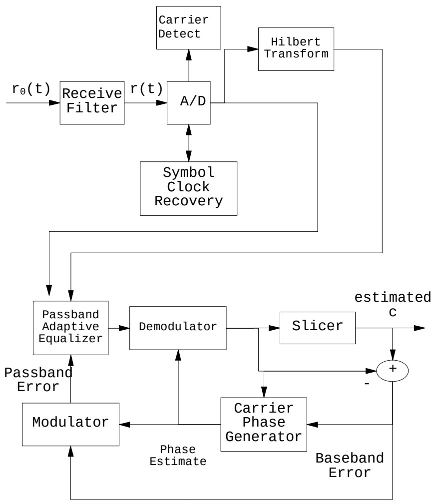

Figure 3. Block diagram of a QAM receiver [1]

3.0 Conclusions

A DSP-based communications system design and implementation course has been presented in this paper. Different aspects including the organization of the course are presented. Classroom lectures and laboratory experiments are discussed. Student projects have been discussed with some specific examples. This course has been useful to the students in understanding DSP architecture, design and applying these to implement important communication sub-systems. This course is in the process of upgrading to include TMS320C6X series DSP. With transition to new powerful DSPs, it is expected to include more sophisticated communications algorithms as regular laboratory experiments. This transition will enable students to carry out more complex projects.

4.0 Reference

[1]. Steven A. Tretter, *Communication System Design Using DSP Algorithms with Laboratory Experiments for the TMS320C30*, Plenum Press, 1995.

[2] Nishith D. Tripathi, and Jeffrey H. Reed, *Laboratory manuals: DSP Implementation of Communication Systems*.