

# Combined Signal Processing and Wireless Communications Laboratories at Iowa State University

J.A. Dickerson (julied@iastate.edu) and S. Koziol ([skoziol@iastate.edu](mailto:skoziol@iastate.edu))

Electrical and Computer Engineering Department

Iowa State University, Ames, Iowa, 50010

## I. Introduction

In recent years, there has been a convergence of technology within the fields of digital signal processing (DSP) and digital communications. To better reflect these trends and the needs of industry, Iowa State University (ISU) is evolving our existing laboratory courses in DSP and communications. Our goal is a complementary set of undergraduate DSP and communications laboratories that augments the current DSP and communications curriculum. The laboratory experiments contain elements of both DSP and communications systems design. The laboratory development projects described in this paper meet the need for students to have practical, real-world experience as part of their undergraduate education.

## II. Laboratory Overview

### A. *Systems approach*

The DSP and communications courses will be linked by several large-scale design experiments. Examples of these experiments include software radios (digital receivers), differentially encoded asynchronous biphase-shift keying (BPSK) receivers, spread spectrum RAKE receivers, and joint source-channel coding for wireless video codecs. Each experiment combines basic techniques such as filter design with advanced topics such as multi-rate signal processing or channel coding. The large-scale design experiments will allow the students to collaborate to build a working system. These projects require the students to work as a design team and debug and test a complex design. These courses build on successful courses developed at the University of Maryland [1] and Virginia Tech [2].

### B. *Theory to be emphasized*

The DSP laboratory course emphasizes projects in digital filter design, experiments related to Fourier and time-frequency signal analysis, and signal compression. It will also include projects in digital communication such as the generation of spread spectrum signals, waveform generation, and multi-rate processing, but it concentrates on DSP implementations. The communications laboratory will focus on modulation, receiver and transmitter design, and coding.

### *C. Hardware choice*

We plan to use the TMS320C542 fixed point processor for the DSP course and the TMS320C30 floating-point processor for the communications course. Using separate processors will expose the students to both fixed and floating-point processors. Although it would be easier to use a floating-point processor for both courses, we feel that experience with fixed point DSPs is important due to their wide use in commercial products. Fixed-point boards also introduce implementation considerations such as coefficient and input scaling and overflow. We chose the fixed-point C54x series because of its wide use in cellular phones, PCS terminals, pagers, IP phones, portable information systems, digital hearing aids, etc [3]. We chose the C3x series because it is a proven floating point DSP with many lab books and materials currently available to support it [1, 4, 5]. These resources will speed our lab's development time and will keep us from re-inventing the wheel.

The advent of inexpensive DSP boards such as Texas Instrument's (TI) C54x and C3x series boards that connect directly into PCs has made it possible to create low cost DSP workstations for students to get hands-on design experience with different types of signal processing [6]. Another popular approach for lab is to use software packages such as Matlab™ to simulate signal processing systems and work with a wide variety of signals [7], [8]. The problem with this approach is that students do not work with real-time signals and real-world problems of signal quality, quantization, and slow processors. One rarely has the precision and the chance to do processing off-line. At ISU, Matlab™ is used extensively for classroom demos and homework assignments in DSP, but hardware experience is essential for practicing engineers.

## III. Laboratory Goals and Objectives

We are reshaping our existing laboratory courses in DSP and communications to emphasize the convergence between digital signal processing and wireless communication technology. Our ultimate goal is to develop an integrated set of laboratory experiments that build on the skills learned in both disciplines. A secondary goal is to combine research and teaching in our capstone senior design course. This shift in emphasis is important since there is a strong demand for trained engineers in the areas of DSP, communications and computers throughout the world.

Our proposed labs coupled with the Iowa State College of Engineering goal to have all students graduate with co-op or intern experience will help provide the necessary engineers. The lab courses include hardware and software components combined with a change in teaching emphasis at the undergraduate level. We are actively soliciting feedback from industry on the material in the course and on student preparedness in the work world. This external evaluation has proved to be effective as the hardware and many of the experiments in the previous iteration of the laboratory were selected after consultation with a DSP design engineer at Rockwell Collins. The lab projects will focus on the integration of basic and advanced digital signal processing techniques with wireless communication applications. The projects will develop progressively and culminate in a complete system.

## IV. Experiment Sequence

The DSP lab consists of a basic section of experiments and an advanced section of experiments. All students are required to perform the basic experiments to gain familiarity with the lab equipment and DSP development tools, but each advanced project will not be performed by all groups due to time limitations. The basic experiments are:

- Introduction to the C54x hardware and Code Composer using audio effects filters.
- Creation of periodic waveforms and noise sequences using the DSP board using both wavetables and filters.

- Design and implementation of FIR and IIR filters [9].
- Image processing project using Matlab™

Each basic experiment is completed in one lab session. The advanced projects are each broken into modules, with each module taking one session to complete. Advanced projects include:

- Subsystems for a software radio
- Differentially encoded asynchronous BPSK communications

## Course Outline

1. Introduction to DSP development system and C compiler
2. Using the test equipment and direct memory access to monitor program performance
3. Using Matlab™ DSP Workbench and signal processing toolbox
4. DSP based waveform generation
5. Digital filter implementation and realizations, scaling
6. Musical Effects using digital filters
7. Image Processing Lab
8. Final Project
9. Final Project
10. Final Project

## V. Laboratory Setup and Equipment

A laboratory station consists of the following hardware: a personal computer (PC), DSP target board, emulation hardware, oscilloscope, function generator, speakers, and microphone. For the DSP laboratory, we are using the Texas Instruments TMS320C54x DSKplus for the target board. The DSKplus package is complete in that it has all the hardware and software necessary to debug and execute programs on the DSP. We chose to use the DSKplus solely as a target board and are not using the DSKplus's software package or emulation capabilities. This decision was made because *Code Explorer*, the code development environment included in the DSKplus package, will not function on a Windows NT 4.0 operating system. We replace *Code Explorer* with *Code Composer*, a similar but expanded software package.

In the conventional DSKplus system, the PC communicates with the DSP board via the parallel port. Our configuration uses the XDS510 emulation hardware as a link between PC and DSP. The XDS510 works in conjunction with a Joint Test Action Group (JTAG) system to load programs to the DSP and probe the DSP's registers with minimal effect on DSP board operation.

### A. Signal Processing Tools

**Code Composer (software):** Code Composer is software that functions as a fully integrated development environment (IDE). It provides a “Windows” environment in which DSP code can be edited, compiled, assembled, debugged, and loaded onto the DSP chip. The DSP’s registers can also be monitored using this software, and data can be plotted in graphical windows. Using the PC’s environment variables, Code Composer links itself to another software package called Code Generation Tools (CGT) that contains the C-compiler, assembler, and linker program [10].

**DSKplus evaluation board (hardware):** The DSKplus board includes a TLC320AC01 analog interface circuit (AIC). The AIC is an audio-band processor that provides an analog-to-digital and digital-to-analog input/output interface system [11]. This codec will be used in lab to input sound to the DSP via a microphone (or soundcard) and output sound to a speaker (or soundcard). The DSKplus also includes 10K words of on-chip Dual-Access RAM.

The DSKplus board is normally configured to communicate with a PC via a parallel port. In our lab set up, we are not using this method. Instead, we are using an emulation setup consisting of an XDS510 ISA board and a JTAG device, Figure 1. The XDS510 controls the flow of information to and from the DSP, and the JTAG device performs the actual scanning of the DSP’s memory and registers. The XDS510 sends its information to Code Composer where the data can be monitored and manipulated in a Windows environment. This system allows one to monitor the DSP in almost real time [13].

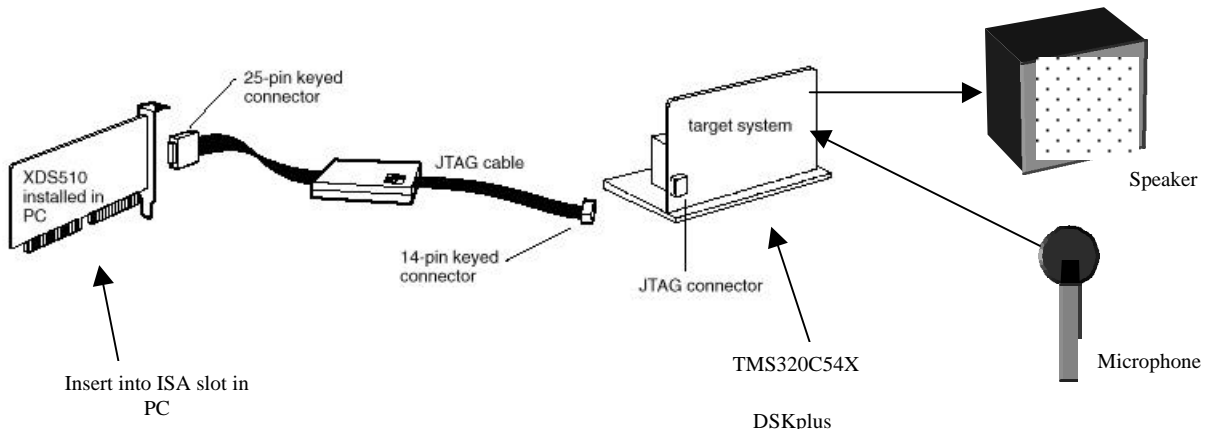


Figure 1. DSP equipment setup [12]

### B. Test and Measurement Equipment

Oscilloscopes will be used to monitor signals in and out of the evaluation board’s analog interface circuit. This will give the students an additional means of observing the results of their algorithms. Function generators will be used in some cases to produce predictable signals and in other cases they will produce noise for the experiments. FFT modules attached to the oscilloscopes will provide spectral information on the signals.

### C. Simulation Software

Matlab™ is a valuable tool for prototyping and simulation. The students use Matlab™ to familiarize themselves with algorithms before implementing them on the DSP board. The signal processing toolbox includes filter design and rate changing algorithms, which helps the students design more complex systems.

## VI. Laboratory Experiments

We have recently been updating our DSP laboratory with the support of a grant from the Rockwell Foundation. Our emphasis is on a systems approach to learning communications and signal processing. This approach has been very successful over the past few years in the basic signal processing class. The current labs can be seen at <http://www.eng.iastate.edu/~julied/classes/ee424/lab/labmain.html>.

Each new laboratory assignment will be composed of five parts: Objectives, Theory, Simulation, Implementation, and Evaluation. The lab assignments start by stating a few specific, measurable, and attainable objectives. Block diagrams and characteristic equations are then used in the Theory section to present a clear explanation of the DSP theory being studied. Each lab assignment then asks the students to simulate the algorithms using Matlab™ before beginning hardware implementation (formerly implemented on the Analog Devices ADSP-2181). This provides a good understanding of the expected results and effects of the algorithms. Some Matlab™ code is included in the lab to get the students started, but they are asked to modify it as an exercise in evaluating the effects on the signals when parameters are changed. After successfully simulating with Matlab™, the students begin the hardware part of the lab. C code and assembly code will be used for implementation. Because of the complexity of the code and the relative short time-span of the semester, skeleton code is supplied in the handouts. The skeleton code teaches important items such as: how to construct a linker command file, how to set up the AIC, how to use vector tables, and general DSP routines. Because this is an introductory course in DSP, the students will only be required to modify existing assembly and C-code. Once the code is loaded to the DSP, oscilloscope will be used to monitor and compare signals. This will provide the student with a different angle of viewing and measuring the results. Finally, the students answer some conceptual questions in their formal write-up of the experiment.

### A. Introductory laboratory experiments

- **DSP Introduction:** The purpose of this laboratory experiment is to teach the code development process for the C54x, and also how to use test equipment. The students gain a functional overview of the hardware and software by implementing echo, comb, and reverb filters.
- **Digital Waveform Generators:** Digital waveforms and pseudonoise sequences can be created using a filtering approach and a wavetable approach. This knowledge is put to practical use by creating a DTMF transmitter. The information taught in this laboratory experiment is used in subsequent experiments.
- **Digital Filter Design:** This laboratory experiment focuses on the development of transfer functions that meet certain frequency response specifications. Several methods of FIR and IIR filter design are investigated. The results of the filters are illustrated by implementing different musical sound effects and encouraging the students to bring in portable CD players as input sources. The experiment also covers the effects of coefficient quantization and scaling to prevent overflow and different filter structures.

## B. Advanced laboratory experiments

- Differentially encoded asynchronous BPSK communications: The implementation of DPSK modulators and demodulators concentrates on signal generation, mixing, digital logic switching, and filtering. The lab is composed of the following three modules that combine aspects of both Matlab™ and hardware implementation.

1. The students will first use Matlab™ to simulate the signal generator for DPSK signals, Figure 2. A defining equation for DPSK signals is:  $d_k = d_{k-1}b_k \oplus \bar{d}_{k-1}\bar{b}_k$ , where  $b_k$  is the input binary digit at time  $t_b$ , and  $d_{k-1}$  is the previous value of the differentially encoded digit. The amplitude level shifter changes a zero to the level  $-E_b$  and a one to the level  $+E_b$ .

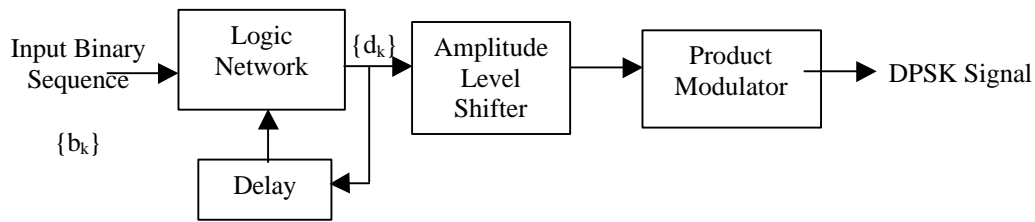


Figure 2. DPSK modulator [14]

After simulating, a DPSK sequence will be loaded to the DSP hardware. This sequence is mixed with a carrier that is produced using the wavetable approach.

2. The students will next simulate the components for an asynchronous differential BPSK receiver, Figure 3.

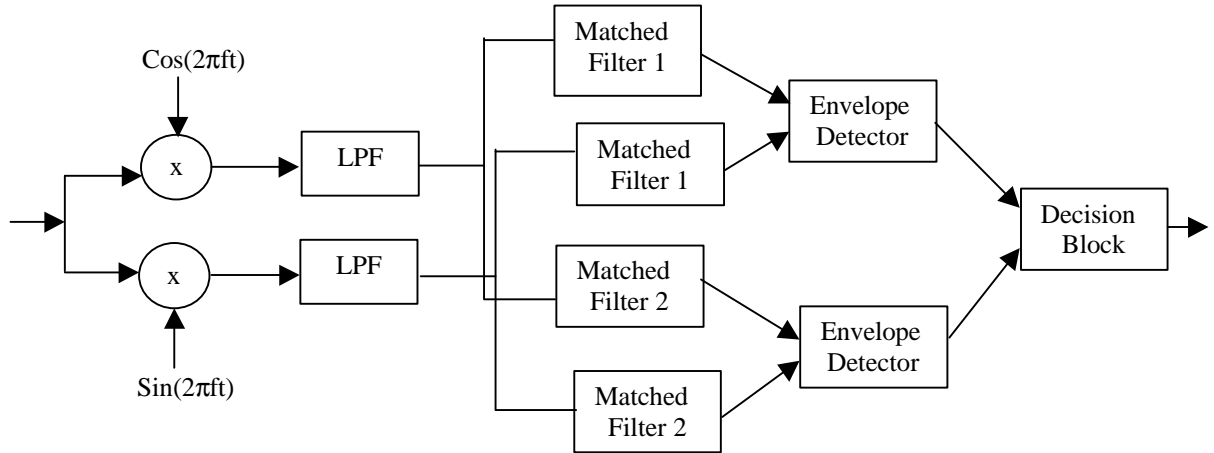


Figure 3. Asynchronous Differential BPSK Receiver

3. Finally, the students will implement the mixing and LPF block of Figure 3. on the C54x. To verify that their hardware implementation is correct, the output of the LPF will be sent through the Matlab™ coded matched filters, envelope detectors, and decision blocks generated in the previous lab.

- **Software Radio:** Introducing multi-rate processing is the key focus of the software radio experiment. In a series of three lab experiments, the students construct digital receiver components that replace their analog equivalents in a heterodyne radio receiver. The digital receiver takes a signal sampled at a high rate, down converts it, low pass filters it, decimates it, and then demodulates it for playback on a speaker [15].

The project is composed of the following experiments:

1. Using Matlab™, the students will simulate the entire software radio system, Figure 4.

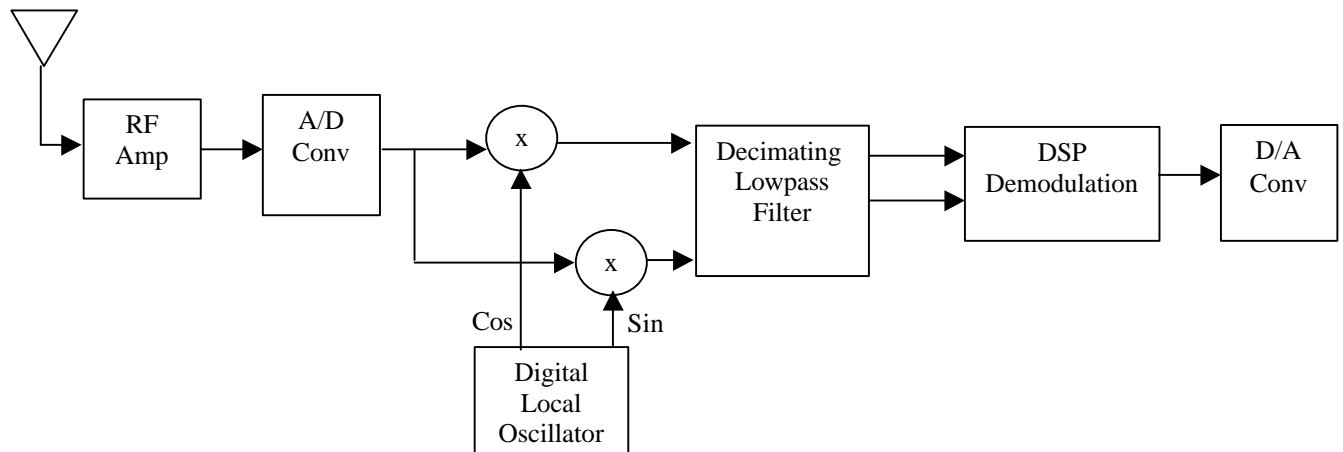


Figure 4. Software Radio Block Diagram [16]

2. Building on previous Matlab™ code, the students will prepare an input signal by modulating information with a carrier. This signal will be saved as a .wav file and used as the “RF” signal. This signal is sent to the C54x codec, and the processor mixes it with sinusoids produced using the wavetable approach.
3. Designing and implementing the decimating low-pass filter block and demodulation block are the final tasks.

## Summary and Conclusions

This paper has described an undergraduate DSP laboratory that is focused on combining experiments which contain elements of both DSP and communications systems design. We have shown that these two aspects can be successfully combined to meet the need for students to have practical, real-world communications and DSP experience as part of their undergraduate education.

## References

1. Tretter, S.A., *Communication System Design Using DSP Algorithms: With Laboratory Experiments for the TMS320C30*. Applications of Communications Theory, ed. R.W. Lucky. 1995, New York: Plenum Press.
2. Rappaport, T.S., et al. *Curriculum Innovation for Simulation and Design of Wireless Communications Systems*. in *American Society of Engineering Education (ASEE) Annual Conference*. 1996: ASEE.
3. Texas Instruments, <http://www.ti.com/sc/docs/products/dsp/c5000/index.htm>.
4. Sorensen, H.V. and J. Chen, *A Digital Signal Processing Laboratory Using the TMS320C30*. 1997: Prentice Hall.
5. Chassaing, R., *Digital Signal Processing Laboratory Experiments Using C and the TMS320C31DSK*. 1999, New York: John Wiley & Sons, Inc.
6. Taylor, F.J. and J. Mellott, *An Academic DSP Workstation*. IEEE Signal Processing Magazine, 1995. **12**(6): p. 33-37.
7. Ebel, W.J. and N. Younan, *Counting on Computers in DSP Education*. IEEE Signal Processing Magazine, 1995. **12**(6): p. 38-43.
8. Orsak, G.C. and D.M. Etter, *Collaborative DSP Education Using the Internet and MATLAB*. IEEE Signal Processing Magazine, 1995. **12**(6): p. 23-32.
9. Tsai, M., *IIR Filter Design on the TMS320C54x DSP*. Vol. SPRA079. 1996: Texas Instruments. 9.
10. Go DSP, *Code Composer Information*. Vol. <http://www.go-dsp.com/product/html/cc.htm>.
11. Texas Instruments, *TLC320AC01C Single-Supply Analog Interface Circuit*. Vol. SLAS057D. 1996, Dallas: Great Impressions Printing & Graphics.
12. Texas Instruments, *XDS51x Emulator Installation Guide*. Microprocessor Development Systems. Vol. SPNU070A. 1996.
13. Brokish, C., *Emulation Fundamentals for TI's DSP Solutions*. Texas Instruments Application Report. Vol. SPRA439. 1998.
14. Ziemer, R.E. and W.H. Tranter, *Principles of Communications: Systems, Modulation, and Noise*. 4 ed. 1995, New York: John Wiley & Sons, Inc.
15. Pentek, *Digital Receivers Bring DSP to Radio Frequencies*, . 1999, Pentek: New Jersey. p. 5.
16. Hosking, R., *Digital Receiver Handbook*, . 1998, Pentek: New Jersey. p. 32.