A Fixed-Point DSP Lab Using the TMS320C541

James Caffery, Jr., Tom Millikan, and Jeremy Wilch

University of Cincinnati
Dept. of Electrical & Computer Engineering & Computer Science
Cincinnati, OH 45221-0030

jcaffery@ececs.uc.edu

http://www.ececs.uc.edu/~jcaffery

Abstract

In this paper, we briefly describe the development and implementation of a digital signal processing laboratory at the University of Cincinnati which is based on the fixed-point processor TMS320C541. We examine the motivations behind the course along with its structure. A discussion of the various labs is given with a topical overview of the corresponding lectures. Finally, we provide an evaluation of the outcomes of the course with directions for future development.

1 Introduction

Digital signal processing (DSP) is a topic of great interest among students today. As the technology has advanced, a multitude of applications and uses for DSP technology have appeared. These include home stereo systems, cameras, telephones, household appliances, automobiles, children's toys and many more. Each of these commercial adventures and successes for DSP has broadened the awareness of the general public and of students (not just electrical engineering majors) to the incredible benefits and possibilities for digital design based on DSP. It has not just been the interest of students that has encouraged the development of solid DSP curriculum, but more importantly, it has been the need from industry to recruit and hire students with a solid foundation in the fundamentals of DSP. Any experience with DSP hardware only serves to enhance the student's marketability. To this end, the University of Cincinnati has had in place a DSP course which covered the theory of DSP as done at most every other university nowadays. Also, the University's relationship with Texas Instruments through its co-op program has made DSP courses valuable to students wishing to work with TI. When confronted with the possibility to take a laboratory course in DSP, the students were overwhelmingly in favor of such a course.

The question arises, "Why have a DSP lab?" The motivation for a DSP lab is to enhance the students' knowledge and understanding of DSP concepts. The lab gives the students a means to relate the (somewhat) abstract concepts of DSP theory to the physical world. Also, programming algorithms gives the students more detailed knowledge about an algorithm and how it performs than they might acquire in a lecture setting.

Our choice of TI DSP boards was motivated by trends in industry, namely, that fixed-point DSP boards are much more common for use in industry and commercial product development than floating-point DSPs. In fact, it is estimated that 92% of the world market uses fixed-point DSPs because of their lower cost. Many schools have DSP labs, but most (from our investigations) are based on floating-point processors (at least those using TI processors). By using a fixed-point processor, the students must deal with real-world limitations when designing and implementing DSP algorithms on the DSP board. This is a key component for educating our students for what they will encounter in the real world.

With support from TI, we were able to obtain the equipment necessary to create and sustain such a laboratory course. Eight TMS320C541 fixed-point DSP boards which are used in the laboratory with Code Composer software for evaluation and debugging.

In the following, we discuss the development of the DSP lab course at the University of Cincinnati. The laboratory environment is described and the course structure is outlined. We discuss some of the topics and labs as well as the design project that were part of the course. Finally, we end the paper with some observations on the course where we examine shortcomings of the course and focus on improvements for the future.

2 Course Description

The DSP course at the University of Cincinnati is offered to seniors and graduate students. Its focus is on the fundamental theory of DSP: sampling, convolution, filters and their structures, Fourier and z transforms, etc. These concepts are dealt with in an ideal setting: infinite precision to represent filter coefficients, no noise, perfect analog-to-digital conversion, perfect sampling, etc. Tools such as Matlab are used to enhance the students' understanding of the concepts that are learned in the classroom. However, the students get little or no knowledge of how to implement algorithms in a hardware environment, since most of the algorithms are already implemented in Matlab and the course is designed to develop an understanding of the theoretical concepts. In addition, students do not generally get an in-depth idea of the applications that DSP technology is useful for.

We wanted to develop a course not just to give students the opportunity to implement algorithms on a DSP board, but to give them practical real-world experience with DSP hardware and issues related to it. Consequently, the course was developed with certain goals in mind:

1. Enhance/strengthen the knowledge learned in the DSP course.

2. Provide the students with an understanding of the wide-range of applications that DSP technology supports.

3. Give the students hands-on experience with DSP technology.

4. Provide theory pertaining to real-world limitations (quantization, roundoff errors, overflow).

5. Give the students experience with these limitations first-hand to facilitate understanding of the theory.

No particular textbook was used for the course. As the textbook *Digital Signal Processing*, by Proakis and Manolakis [1], was used in the DSP theory course, it was used as a reference for the lab course. Unfortunately, it was difficult to find teaching materials related to the hardware and assembly language that could serve as a thorough introduction for the students. The TI reference books for the TMS320C541 were made available, but are not useful as a starting point for instruction. Consequently, lecture notes and handouts served as the "textbooks" of the course[1].

The laboratory for the course included eight PCs running Windows 98. The DSP boards were installed in each computer along with the Code Composer software for evaluation and debugging. Each of the computers is connected to a printer via the local network. The ideal size of the course is two to three students per computer, which limits class size to under twenty-four students.

2.1 Course Structure

The course was comprised of two parts: a two-hour lecture and a four-hour lab each week. The lectures were organized to be pertinent to that week's and future weeks' lab exercises. They were used to review concepts from the earlier DSP course as well as introduce them to new concepts such as the 'C54x architecture, assembly language, and theoretical analysis of quantization effects and overflow. In the next two sections, more details are given regarding the content of the weekly lectures and labs.

[1]There are a couple of instruction materials under development at this time. They include a DSP teaching kit (DTK) for the 'C54x which will provide tutorials on the assembly language and lecture topics for the course. In addition, a textbook is begin developed on the 'C54x that will be available in the future.

2.1.1 Lectures

The topics that were a part of the ten weeks of lecture are listed below:

- Hardware/architecture of ’C54x
- Assembly Language on the ’C54x
- Sampling, quantization, A/D and D/A conversion
- Representation of Numbers
- FIR and IIR filtering structures & implementation issues
- DFT/FFT and power spectra
- Effects of quantization, round-off and overflow in filters and DFT
- Decimation and interpolation
- Advanced topics (modulation/demodulation for communications systems, adaptive filtering, linear predictive coding, introduction to ADSL).
- Guest lectures

A brief introduction to the hardware and architecture of the ’C54x family of processors was given to the students. Focus was placed on the ’C541 when appropriate to point out differences among the ’C54x processors. The students were presented with the key features of the DSP board and given an understanding of its operation. Discussed in detail were addressing modes, status registers, program and data memory, memory-mapped registers, and a brief discussion of the pipeline and issues related to it.

A small amount of time was spent reviewing key DSP concepts since more than a quarter had passed since the DSP course was offered. A major focus were those topics which could find implementation on the DSP board such as digital filters, Fourier transforms, and any other algorithm leading to structures for implementation. The structures of the FIR and IIR filters were reviewed as a prelude to the discussion of the effects of quantization, round-off error and overflow in finite-precision DSP processors. The content of the lectures concerning these key finite-precision issues were obtained from [1] and [2]. The same is true for finite-precision effects on the DFT/FFT. Several weeks were spent on these topics including theoretical derivations and discussion of the practical meaning of the results. The labs and work in Matlab provided the students further exposure to these ideas. These topics received a considerable amount of attention since they were not covered in the previous DSP course. In addition, these are important concepts of which the students should be aware when implementing algorithms on hardware platforms.

The guest lectures provided a valuable experience for the students to interact and question people from industry with DSP experience. They provide the means of enlightening the students to the many uses and applications that employ DSP technology. Guest lectures from industry are an interesting part of the course that provides a relaxed atmosphere for learning.

As time permitted during the project phase of the labs, additional topics were covered that were of interest to the students. The depth of the coverage was only intended to be introductory to give the students an understanding of concepts in other topic areas.

2.1.2 Labs

The labs were chosen to focus on the theoretical structures, etc., that the students learned in their DSP theory course. The labs which were assigned are shown in Table 1. The first lab was written by two senior students with prior experience with the TMS320C541. It focused

Week 1	Introduction to the TMS320C541 EVM
Week 2	Digital Waveform Generations
Week 3	The Analog Interface Circuit (A/D, D/A)
Week 4	FIR Filter Design & Implementation ($1\frac{1}{2}$ weeks)
Weeks 5-6	IIR Filter Design & Finite Precision Effects (2 weeks)
Weeks 7-10	Project

Table 1: Organization of weekly labs.

on the Code Composer development interface and the basic instruction set of the 'C541 processor. This was designed to provide the students with an introductory knowledge of the EVM tools that were available. The students were introduced to the basic structure and format of `.asm` and `.cmd` files and also learned essential skills for debugging program files using Code Composer. The students were shown the use of GEL files, viewing data and registers, viewing dis-assembly, setting and using breakpoints and probe points, loading/saving data to/from memory and using I/O data streams. The students were also introduced to visualization of data using Code Composer graphing tools. The students were required to perform some simple tasks with assembly code that was provided to them such as compiling and linking, executing the code, and viewing/graphing data.

The second lab served to give the students more experience with programming the DSP board as well as learning some important lessons regarding quantization and overflow. From the basic theory regarding digital oscillators, the students were given the basic code structure and were required to compute required values for parameters to create a sinusoid of a particular frequency. From this, the students learned about oscillator theory, but more importantly, saw the effects of register overflows and experienced the sensitivity of the oscillator to quantization of the computed parameters.

The third lab provided the students with more knowledge concerning the DSP hardware itself, particularly the A/D and D/A circuits. Music was streamed in from the host computer's CD-ROM and out of the DSP to the computer's sound card. The students were required to compute register values to change the sampling rate, etc., from the TL-C320AC01C data manual [3]. This lab allowed the students to hear first hand, from their audio CDs, the effects of aliasing (by changing the sampling frequency) and quantization through various exercises. These are concepts the students learned in their DSP course, but using music in this way provided a simple means to reinforce those ideas.

With the filtering labs, the students were given more opportunity to program the DSP. Although the commands of the assembly language instruction set had not been covered in detail in the lectures, the previous labs had served as a means of introducing them to the instruction set. The FIR lab was designed to give the students experience in writing assembly code from scratch and debugging it. This lab also introduced the students to the concept of circular addressing.

The IIR labs were much more intensive in that many important concepts were stressed. Matlab was used to design the filters which the students were to implement on the DSP board. This lab stressed the effects of filter coefficient quantization, roundoff errors and overflow. From the theory discussed in the lectures, the students designed their filters using a cascade of second-order filters in Matlab. Most of the students also encountered overflow problems, and were required to justify any scale factors that they used based on the theory given in lecture. They also explored different structures that ease the scale factor requirements, as discussed in [4].

Finally, the students were required to complete a project of their choosing in the application areas of speech processing, image processing, controls, communications, etc. The students worked in larger groups than in the weekly labs and were required to build their own code projects from scratch and implement them on the DSP board. Projects included adaptive interference cancellation, equalization, a 256-point FFT, and linear predictive coding for speech applications.

3 Future Directions

The problem encountered most often in the course was the student's unfamiliarity with the 'C54x assembly language instruction set. A pre-requisite for the course was that the students have taken an assembly language course. However, the subtleties involved with many of the instructions caused great confusion for the students. With some commands, registers are incremented, data is moved and pointers are changed in addition to the basic operation of the instruction. A good example are the differences between the instructions `MACD` and `MACP`. Although the labs were designed to stress understanding of the instruction set along with DSP concepts, the students would benefit from more instruction during the lecture time to familiarize them with the subtleties of the instruction set. Consequently, more time will be devoted to teaching the 'C54x instruction set during the first few lectures. The tutorials presently being developed for the 'C54x DTK will be of tremendous help in that regard. In addition, some attention should be given to programming code structure since the

'C54x instruction set permits more versatility in branches and subroutine calls than typical microprocessors.

A further consideration is that the course should be more focused toward the hardware in terms of more details concerning the architecture (memory maps, etc.). The students should also be exposed more to system level design issues such as the operation of the AIC and serial ports.

A final consideration for future development is that the students would prefer more programming experience earlier in the course, rather than working through tutorials and exercises as the first three labs required. This can easily be accomplished by rewriting those labs and giving the students more responsibility for writing the programs.

4 Conclusions

We have described the DSP laboratory course that was developed this past spring at the University of Cincinnati. The course we designed to give the students knowledge and experience in implementing algorithms on the DSP boards. The course stressed real-world aspects such as quantization effects and overflow. These concepts were stressed both in the weekly lecture and also in the weekly labs. The students were faced with handling effects of quantization as well as accumulator overflow. Weekly lectures provided the additional theory that related to these real-world issues and provided information for completion of the labs as well as further education in advanced topics in DSP. For future development of the class, more emphasis will be spent on providing a better understanding of the architecture and assembly language used by the TMS320C541, which proved to be the main difficulty in the labs.

5 Acknowledgments

We wish to thank the TI for their generosity and helpfulness in supplying materials for the DSP course. We would also like to thank Dr. Tom Mantei at the University of Cincinnati for his help and encouragement in the development of this course. Finally, we would like to

thank the students who took the course who provided valuable feedback on which much of our future directions will be based.

References

[1] Proakis and Manolakis, “Digital Signal Processing,” 1994.

[2] Oppenheim and Schafer, “Discrete-Time Signal Processing,” 1989.

[3] Texas Instruments, “TLC320AC01C: Single-Supply Analog Interface Circuit,” TI Data Manual, 1996.

[4] Mandy Tsai, “IIR Filter Design on the TMS320C54x DSP,” Texas Instruments, Document SPRA079, 1996.