

Educating and Training Digital Signal Processing Using TMS320 Processors

Claudia Mathis, Bernhard Rinner, Christian Steger and Reinhold Weiss *
[mathis, rinner, steger, rweiss]@iti.tu-graz.ac.at
Institute for Technical Informatics
Technical University Graz, AUSTRIA

Abstract

This paper presents (i) a lecture to introduce DSP architectures and DSP processors, (ii) a laboratory course to provide experience with DSP algorithms, and (iii) selected student projects that demonstrate the successful application of the acquired knowledge to real-world DSP problems. The goal of this teaching effort at our institute is to provide a solid education and hands-on experience in DSP applications for graduate students in Electrical Engineering and Telematics.

keywords:

*DSP education,
TMS320,
parallel processing*

1 Introduction

Digital signal processing (DSP) is one of the fastest growing markets, and the future for DSP-related research and applications looks very bright [2, 4]. Hence, the demand for researchers and engineers in DSP is steadily increasing [8].

A primary teaching goal of our institute is to provide a solid education in DSP algorithms and processors as well as hands-on experience in DSP applications for our students. In this paper, we present (i) a lecture to introduce DSP architectures and DSP processors, (ii) a laboratory course to provide experience with DSP algorithms, and (iii) selected student projects that demonstrate the successful application of the acquired knowledge to real-world DSP problems. With these courses and projects we cover a wide variety of topics in DSP and attract many students for this challenging area.

Lecture, laboratory course and projects are intended for graduate students in Electrical Engineering and Telematics¹. These students have a basic knowledge in signal processing and programming skills in assembler and C; some of them have also experience in parallel processing and distributed systems. Lecture and laboratory course have been completely revised during the last two years. TMS320C3x and TMS320C4x DSP processors are used as implementation platforms in the laboratory course and for the student projects.

¹The curriculum of Telematics combines Electronics, Communications Engineering and Computer Science.

*Authors in alphabetical order.

The remainder of this paper is organized as follows: Section 2 introduces the lecture; Section 3 presents an overview of our laboratory course. Section 4 briefly describes three student projects. A short discussion concludes the paper.

2 Lecture

The lecture aims at introducing architecture and performance issues of modern VLSI processors with the focus on DSP architectures. The lecture is organized in three chapters:

The first chapter deals with general processor architectures such as pipelines, scalar and super-scalar architectures as well as memory hierarchies and specialized instruction sets. The second chapter presents fundamentals of DSP algorithms and DSP processors. Special features of DSP architectures are described and illustrated by several examples. Various DSP processor architectures are compared. The third chapter introduces in detail the architecture (data path, memory architecture, instruction set etc.) of the TMS320C3x and TMS320C4x processors.

3 Laboratory Course

The lecture is a prerequisite for the attendance of the laboratory course. The laboratory course is organized in four different experiments ranging from simple audio processing with a DSP Starter Kit to parallel image processing using the Parallel Programming Development System [7]. All experiments are implemented using TMS320 DSP processors. The entire equipment of the laboratory course is shown in Figure 1.

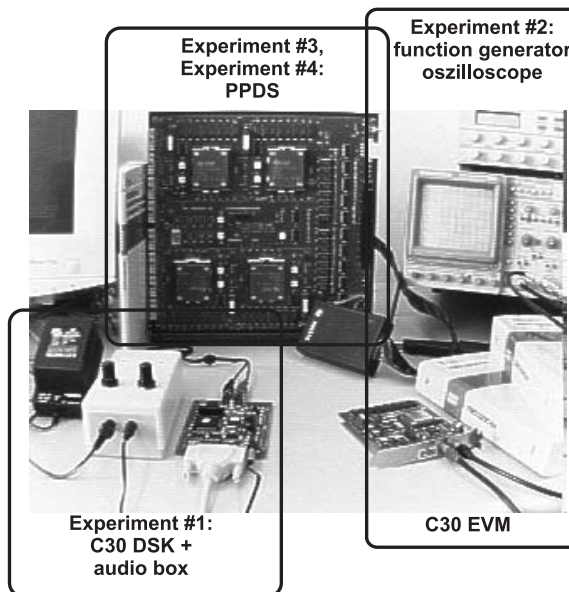


Figure 1: Equipment used for the laboratory course.

Experiment #1 introduces DSP processors and is based on the DSP Starter Kit C3x (DSK) and a simple "audio box" with sound generator and loudspeaker. The students experiment with simple audio signal processing, like rectification and filtering, by programming the DSK in assembler. This experiment is organized as home training – hence, audio box and DSK are made available for the students. The goal of this experiment is to get the students started with DSP hardware and software by executing and modifying sample programs and to encourage unsupervised education.

Experiment #2 is based on the TMS320C30 Evaluation Module (EVM) and the Code Composer as programming environment. This experiment demonstrates the exploitation of DSP processor features, e.g., circular addressing, hardware loops and delayed branching, using assembler programming. The students optimize typical DSP algorithms like FIR and IIR filters and compare them to code gen-

erated by a C compiler. The filter programs, i.e., code, data and stack segments, are mapped onto different memory locations and the effect on the runtime performance is tested using the Code Composer. The filter characteristics are measured using a function generator and an oscilloscope.

Experiment #3 deals with more complex DSP tasks in the area of image processing. The image processing algorithms are implemented in C using the Code Composer and the TMS320C40 Parallel Programming Development System (PPDS). Simple pixel manipulation functions like brightness and thresholding as well as more complex box-filter algorithms have to be developed. The students experiment with different coefficients of the box matrix in order to realize different filter characteristics. The image processing algorithms are evaluated with grey-scale images which are conveniently visualized using the Code Composer.

Experiment #4 introduces the parallel processing capabilities of the TMS320C40 processor based on the image processing algorithms of experiment #3. The box-filter algorithm is parallelized by exploiting its data-parallelism and is implemented on the PPDS using the Code Composer. This complex experiment is organized into three steps. First, the data transfer via the communication ports is introduced, and the image filtering is performed like a remote procedure call on two different processors. Second, the image filtering is parallelized on two processors using a master/slave structure. The master processor divides the image into two partitions, sends one partition to the slave processor and filters the other partition. After filtering the local partition, the master receives the filtered partition from the slave and merges both partitions. Finally, this

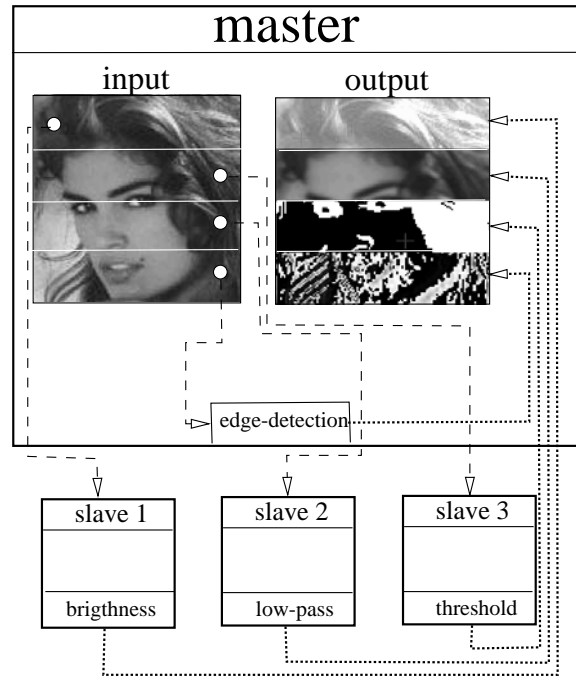


Figure 2: Parallel image processing in experiment #4. The box filter is parallelized onto 4 processors using a master/slave structure.

parallelization strategy is generalized to an arbitrary number of processors. Figure 2 shows the master/slave structure using 4 processors. Different box filters are used at the processors to ease the visualization of the parallelized algorithm.

4 Student projects

Students can specialize their DSP education and gain experience by collaborating in our student projects. During about one semester, students work in small teams to develop real-world DSP applications. Their tasks include problem analysis, implementation, evaluation and documentation of an application. Single- or multi-DSP systems based on TMS320C40 processors are mainly used as implementation platforms. In the following, three student projects are



Figure 3: The complete ANC demonstration system consists of a duct, a PC-rack equipped with a DSP and a data acquisition board, and an audio box including microphone and power amplifiers.

described as examples:

The goal of the first project is to develop an active noise cancellation (ANC) system for demonstrations and laboratory experiments [3]. This demonstration system consists of a long duct with mounted noise and cancellation loudspeakers and two microphones to record the noise and error signal. ANC is implemented as a narrow-band feed-forward system using the FXLMS adaptive digital filter. The implemented ANC attenuates noise at frequencies between 150 and 600 Hz by more than 20 dB. A picture of the ANC demonstration system is shown in Figure 3.

A speech recognition system for embedded applications has been developed by another student project [5]. The recognition system is comprised of a feature extractor and a classifier. The feature extractor is based on a 64-point Fast Fourier Transformation (FFT); the classifier is based on discrete-density Hidden Markov Models (HMM) with a variable codebook size. Training as well as classification are implemented using the Viterbi algorithm. The recognition rate and the performance are

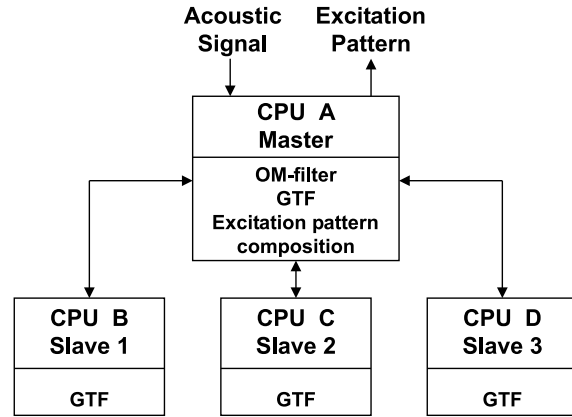


Figure 4: Master-slave structure of the peripheral auditory system simulator.

experimentally evaluated using a test vocabulary of 20 words.

The third student project is a parallel simulation of the peripheral auditory system [1]. Such a simulator can be applied as a preprocessor for acoustic applications, e.g., in speech analysis and recognition. This simulation is based on a non-linear functional model [6] and comprises the outer, middle and inner ear (cochlea). The outer and middle ear is modeled by a linear band pass filter denoted as OM-filter. The function of the cochlea is realized by means of overlapping auditory band pass filters. These auditory filters are non-linear Gammatone filters (GTFs) and form a filter bank. The output calculated from acoustic signals is a so-called excitation pattern, which corresponds to the stimuli to the auditory nerve. To meet the timing requirements for real-time applications a parallel implementation is necessary.

The simulator is implemented on the PPDS. Figure 4 shows the structure of the simulator and the task mapping for four processors. The master calculates the OM-filter. It then distributes the output data of the OM-filter to the slaves, which com-

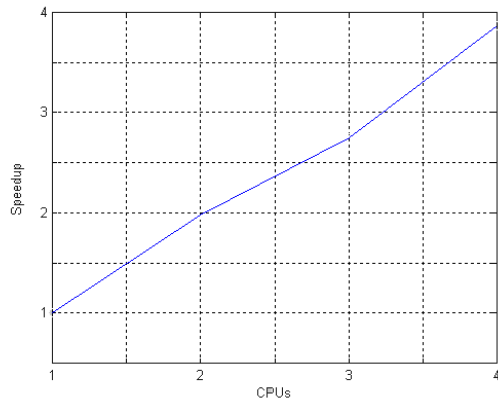


Figure 5: Speedup of the simulator dependent on the number of CPUs.

pute most of the Gammatone filters. The master calculates only the small remaining part of the filter bank and collects the results from the slaves to compose the excitation pattern. Speedup results of this simulator using up to 4 processors are depicted in Figure 5.

5 Conclusion

In this paper we have presented the activities of our institute in educating and training digital signal processing. The lecture, laboratory course and student projects have been evaluated very positively by our graduate students. Additionally, feedback from former students as well as employers confirms the effectiveness of our activities. We plan to intensify our teaching activities in the very near future by introducing a new and advanced DSP seminar/laboratory course.

Acknowledgments

The authors are grateful to Texas Instruments for the donations and the support in establishing the DSP laboratory course.

References

- [1] R. Buechel. Simulation of the Human Peripheral Auditory System on a Multi-DSP TMS320C40 System. Technical report, Technical University Graz, Oct. 1997.
- [2] J. Eyre and J. Bier. DSP Processors Hit the Mainstream. *IEEE Computer*, 31(8):51–59, 1998.
- [3] A. Haimayer, R. Jeza, M. Schmid, and R. Weiss. A Demonstration System for Active Noise Cancellation. Technical report, Technical University Graz, Oct. 1997.
- [4] E. A. Lee and D. G. Messerschmitt. Engineering an Education for the Future. *IEEE Computer*, 31(1):77–85, 1998.
- [5] B. Obermaier and B. Rinner. A TMS320C40 based Speech Recognition System for Embedded Applications. In *The Second European DSP Education and Research Conference*, pages 72–75, Paris, 1998.
- [6] M. Pflüger, R. Höldrich, and W. Riedler. A Non-Linear Functional Model of the Spectral Analysis Performed in the Peripheral Auditory System. In *Proceedings of the International Conference on Signal Processing Applications & Technology*, pages 236–240, San Diego, USA, 1997.
- [7] B. Rinner, R. Schneider, C. Steger, and R. Weiss. A Multi-DSP Laboratory Course. In *The Second European DSP Education and Research Conference*, pages 350–356, Paris, 1998.
- [8] F. J. Taylor. SPECTra: A Signal Processing Curriculum. *IEEE Transactions on Education*, 39:180–185, 1996.