Teaching the TMS320C3X DSP as Individual Projects

Author: M. Otesteanu

ESIEE, Paris

September 1996
SPRA343

Texas Instruments

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain application using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

TRADEMARKS

TI is a trademark of Texas Instruments Incorporated.

Other brands and names are the property of their respective owners.

CONTACT INFORMATION

US TMS320 HOTLINE	(281) 274-2320
US TMS320 FAX	(281) 274-2324
US TMS320 BBS	(281) 274-2323
US TMS320 email	dsph@ti.com

Contents

Figures

Teaching the TMS320C3X DSP as Individual Projects

Abstract

The paper presents an efficient method of teaching the Texas Instruments (TI™) TMS320C3x digital signal processor (DSP) in the frame of a non-dedicated technical subject matter, during an academic semester. Individual tasks are defined, with similar difficulties and almost the same level of learning and understanding the processor capabilities. Some projects are presented, as examples of the method.

This document was part of the first European DSP Education and Research Conference that took place September 26 and 27, 1996 in Paris. For information on how TI encourages students from around the world to find innovative ways to use DSPs, see TI's World Wide Web site at www.ti.com.

Product Support on the World Wide Web

Our World Wide Web site at www.ti.com contains the most up to date product information, revisions, and additions. Users registering with TI&ME can build custom information pages and receive new product updates automatically via email.

Introduction

Actual Curricula

The "Politechnica" University of Timisoara prepares electronics specialists since 1970, in nine academic semesters. Now, about 180 students, each year, receive their diploma in Telecommunications or in Applied Electronics. The curricula for the two diplomas has about half of the topics in common. The other half represents the dedicated topics for their specialization.

The teaching in digital circuits is the same for both branches: Digital Circuits (3rd and 4th semesters), Microprocessors and Microcontrollers (6th semester), Digital Signal Processors (8th semester).

The DSP subject matter, similar to other dedicated subjects, is an introduction in the field, starting with theoretical aspects and continuing with examples to explain or to prove particular features of particular circuits. The goal of the actual teaching process is to keep this dedicated subject matter more general and not related to a particular DSP or DSP family.

Arguments

In my opinion, at the end of the teaching program, the graduates in electronics or telecommunications must be able to use a digital signal processor in new designs, not only to know how it works. So, I think that only writing and simulating algorithms is not enough for that goal.

The students have to design the interface with external devices, to use the interrupts, to initialize the CPU registers, to program the peripherals. And it is more important to design in details, using any particular DSP, than to generally design an application.

New Solution

One possible solution is to design an application after completing the DSP course. But this will be rather late, in the last semester. Further, the task of the project may not be so close to the student's field of interest, if the work is performed in the frame of a DSP dedicated subject matter.

The proposed solution is to perform this design in a non-dedicated subject matter. The topic must be a specializing one, of great interest for the student, the application must be more technical and the project has to solve precise tasks.

Learning DSP with a particular target is much more motivating than learning it as general circuits. Having such a particular and individual target, each student has to go into details, to solve his particular problem. Finally, the students know better the DSP, than after a very good DSP dedicated course.

Teaching Method

Frame Topic

My personal experience in teaching DSP as individual projects is related to Telephony Transmission Systems. This is a specializing subject matter for the students in Telecommunications, during the 7^{th} and 8^{th} semesters. The material covers, in courses and laboratories, frequency multiplexing techniques, for analogue transmission, and time division multiplexing techniques, for digital transmission.

In the frame of Telephony Transmission Systems topic, these students have to prepare a project, during the 8^{th} semester, with 14 hours of supervised activity. This is the opportunity to start the DSP teaching, according to the proposed method: individual projects.

At this point, our students already know analogue-to-digital and digital-to-analogue conversions, signal processing, the A and μ companding laws, the PCM frame structures in both American and European standards. So they are able to design digital transmission applications, in a variety of tasks. The new element is to use, for the first time, a particular DSP.

The Processor

We decided to use for these projects the TMS320C31. The main reasons for this choice are:

- the architecture is simple, consistent and friendly for the first application,
- the internal DMA controller offers useful features for real-time applications,
- the floating-point format can be used to increase the speed of particular algorithms,
- the experience of this project is complementary to what students are leaning during the DSP dedicated topic, where the fixed-point DSP family is studied.

Objectives

The project has two well defined objectives for the students:

- to better understand the principles of time-division multiplex transmission and the particular features of the international telephony standards in use,
- to learn the architecture, the features and the assembly language of a particular DSP.

The first objective is based on the frame topic. For the second objective, students have to:

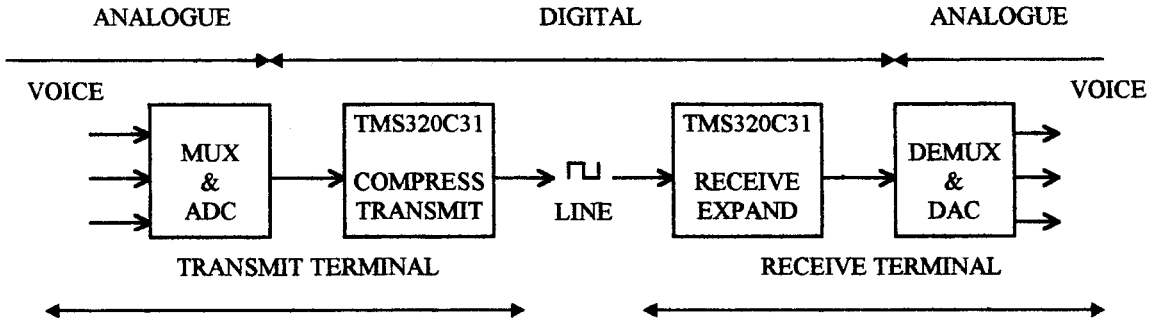
- program the timer, the serial port, the DMA controller,
- use external/internal interrupts and/or polling techniques,
- design the interface with external circuits and share de memory addressing space,
- initialize the CPU registers,
- write and simulate an algorithm.

Because this learning system is not based on a regular course, it is important that each task, for each student, includes all these steps.

Tasks

As general task we analyze a digital transmission system, multiplexing vocal signals, where each terminal must use the TMS320C31 DSP to perform the indicated functions, as shown in Figure 1.

Figure 1. General Task: Digital Transmission System



The individual tasks can be defined by selecting particular values or options for the following parameters.

- Terminal:
  - transmit terminal,
  - receive terminal.

- Standard:
  - n voice channels multiplexing (no frames),
  - 24 voice channels + 1 sync bit (American frame),
  - 30 voice channels + 2 channels (European frame).
- Number of multiplexed frames:
  - one frame, continuous transmission,
  - N frames, continuous transmission. N must be maximized in the corresponding project.
- Type of multiplexing/demultiplexing:
  - in analog form, with a single ADC or DAC for one frame,
  - in digital form, using ADC or DAC for each channel,
  - analogue in the first stage and digital in the second stage, if N frames are processed.
- The input/output is controlled by:
  - The CPU from an ADC or to a DAC,
  - the DMA controller from an ADC or to a DAC,
  - the serial port from a serial output ADC.

The tasks have to be individually defined by combining different options of the listed parameters, according to the following criteria.

- To be individual. Using these options, a lot of individual tasks can be created. Additional tasks as synchronizing word search, synchronizing strategy, signaling multi-frame can be defined. The variety of projects may be significantly increased by the different solutions given by the students.
- To assure the same level of understanding the processor capabilities. In spite of the variety of tasks, it is obvious that each student has to use a timer, a serial port, the DMA controller, some interrupts, internal memory addressing, external devices addressing, implementing a compression or expansion algorithm. Each student will use the same circuits or features of the same DSP, but in a particular way, to match the parameters of his own project.
- To have a similar difficulty level.

NOTE:

Obviously, some of the possible combinations of parameters don’t make sense or are not interesting from a technical point of view. Even so, there are tens of possible tasks to be individually solved.

The paper presents only three samples of these tasks. Using the experience of the selected projects, all other can be easily designed, too.

The Schedule

Because the point of the frame topic is telephony and not DSP, we don't go into details when introducing the TMS320C3x processor. The students are familiar with digital circuits and microprocessors, so the presentation is performed at a specialist level and at a brisk pace. It is almost a four hours seminar. Using transparencies, we are presenting internal architecture, peripherals, external interface, instructions and addressing modes.

At the beginning we are not interested to explain how each function or each circuit is working. The goal is to identify the functions and the circuits which might be used in the project, in order to solve, in an efficient way, each task. After identifying the possibility, each student will have the time to analyze, in detail, the particular mode for his individual project, based on the User's Guide

During the semester, the students have supervised activity in groups of about 10 students. Each class has a particular subject, as defined in the objectives, plus the external circuits design. In this way, general questions receive answers for all the students and particular aspects can be individually discussed.

Each student receives an individual task and has to prepare the corresponding project during the academic semester. The completed project consists of a written part (diagrams, external circuits design, peripheral programming, explanations, features, results) and the simulation of the program performing the task.

Example 1

Defining the Task

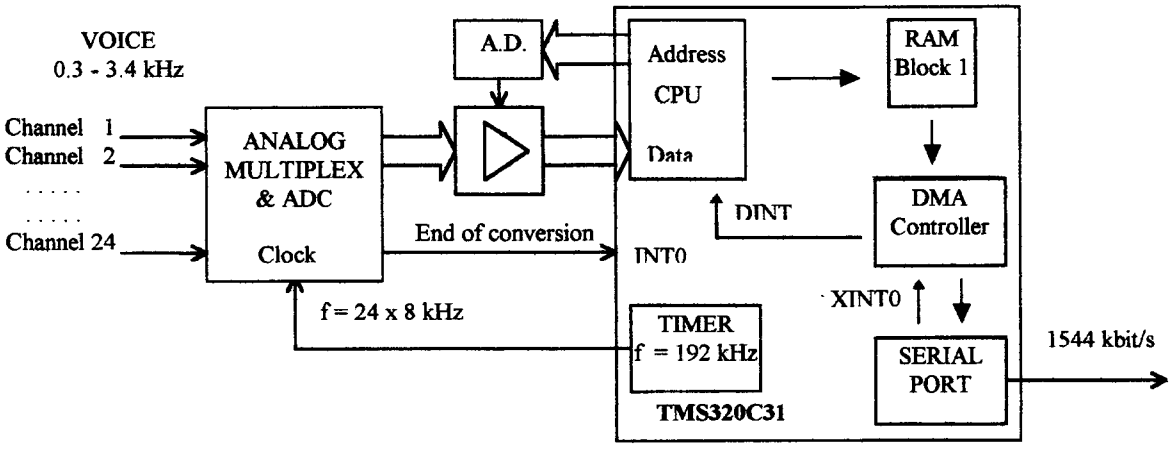
The first task, defined by setting the options for the general parameters, is to design the TRANSMIT terminal for the AMERICAN standard, processing a continuous data flow representing ONE FRAME. The multiplexing is performed by external ANALOG circuits followed by a single analogue-to-digital converter. The acquisition is performed by the CPU.

The analysis of the task offers the following conclusions:

- the number of channels to be processed is 24 (one frame), according to the American standard,
- the sampling frequency for the multiplexed signal is 24 x 8 kHz = 192 kHz, 8 kHz being the global standard for a voice channel,
- the ADC provides a parallel 13 bit word for each sample, to be compressed, according to the μ-law of the American standard, to an 8 bit word.
- the resulting data flow will be 8000 frames/s, where: a frame 1 = synchronizing bit + 24 channels x 8 bit, respectively 8000 x 193 = 1544 kbit/s.

In order to implement the task, all internal resources of the TMS320C31 must be used, as shown in Figure 2.

Figure 2. Transmit Terminal with Single ADC



External Circuits

The multiplexing of the 24 voice channels is performed in analogue circuits, not detailed in the diagram. The PAM signal is converted into digital by a single ADC. The student has to compute the necessary sample frequency (or start conversion) and program the TIMER to generate it.

When the conversion is completed and the output data is valid, the signal *End of conversion* is generated as external interrupt INT0.

The CPU is accepting (or waiting for) the interrupt and proceeds by reading from a fixed address, decoded by the address decoder A.D. The three-state buffer is enabled and the 13 bit word (one sample of one channel), is read by the CPU. The student has to choose an appropriate address in the external memory space.

Internal Circuits

The CPU performs the μ-law compression algorithm, designed by the student, resulting an 8 bit word, which is stored in the internal memory RAM Block 1.

The RAM can be organized with a 24 addresses table, corresponding to each input channel. The data is written by the CPU, synchronized with the acquisition, and read by the DMA controller, synchronized with the data flow transmission. On long term the data transfer is the same, but on very short term conflicts may occur. So, the student may choose, for safety reasons, to use the buffer memory.

The DMA controller must be set to read from an incrementing address, for 24 cycles, and write at a fixed address, of the serial port 0. Because the output data flow must be continuous, the write operation must be performed without delay, so the student must set the destination synchronization. At the end of the 24 transfers, the DMA controller has to send the DINT interrupt to the CPU, in order to be reprogrammed to repeat the task.

The SERIAL PORT 0 can be easily programmed to send the 8 bit words, in any format: 8, 16, 24 or 32 bits. For the transmission timing, the student can choose to use the internal corresponding timer or to design an external clock.

But the single synchronizing bit (1 of 193) cannot be handled in the same way. One solution the student can use is to put the serial port in the general-purpose output mode, for a single bit transmission, then restoring the serial output mode. But he also can solve the problem by software means, if the CPU has enough time to do it in real time.

The Figure 2 shows the steps of data processing and transfer by big arrows (CPU - RAM – DMA - SERIAL PORT) and the internal interrupts by small arrows (DINT and XINT0).

In order to design a synchronized operation inside the DSP, the student must set the indicated interrupts (in the Global-Control register of the serial port and of the DMA controller) and must enable them (in the Status Register and in the Interrupt Enable Register).

Finally, he must correctly load the Interrupt Vector Locations, organize the memory space (including the external circuits) and initialize the CPU registers.

In order to synchronize the functions of different circuits, the interrupts system was presented. But, the same effect can be obtained with the polling method, used by some students.

Example 2

Defining the Task

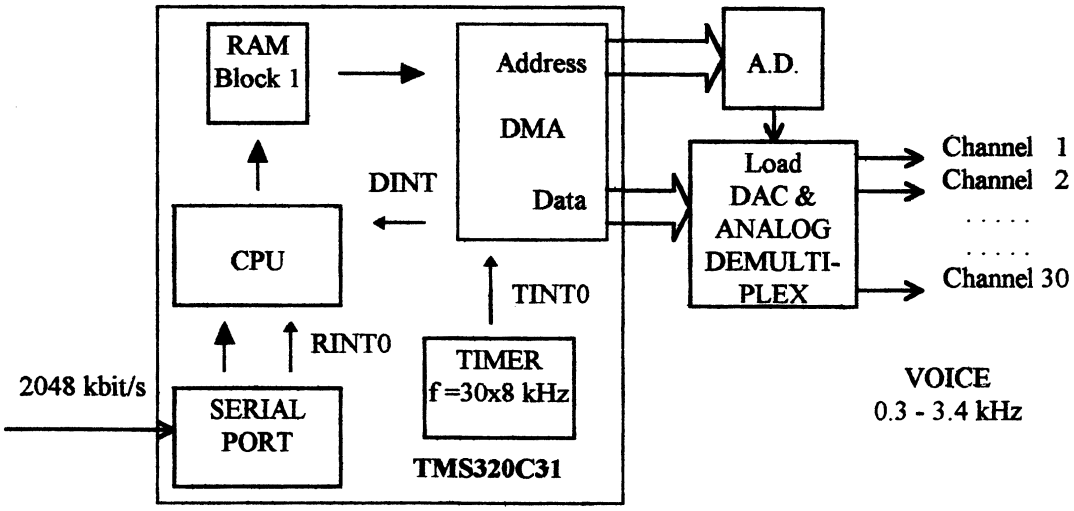
The second task, obtained for other options of the same parameters, is to design the RECEIVE terminal for the EUROPEAN standard, processing a continuous data flow representing ONE FRAME. The demultiplexing is performed by external ANALOG circuits, following the single digital-to-analogue converter. The analogue interface is controlled by the DMA controller.

The first conclusions of this task are:

- the number of channels to be processed during one frame is 30 (voice) + 2 (synchronization and signaling) = 32 according to the European standard,
- the standard sampling frequency is 8 kHz and the compressed word according to the A-law has 8 bit, so, the input data flow is 8000 frames/s x 32 channels/frame x 8 bit/channel, respectively 2048 kbit/s,
- the DAC receives a parallel 13 bit word, expanded according to the A-law of the European standard, at the rate 8000 frames/s x 30 channels/frame = 240 kHz

In order to use the TMS320C31 internal resources, the following solution, shown in Figure 3, is proposed.

Figure 3. Receive Terminal with Single ADC



Internal Circuits

The data flow of 2048 kbit/s is received by the SERIAL PORT 0, in preferred blocks of 8 bits.

The CPU performs the acquisition of the word and the expansion algorithm according to the A-law. To start the process, the internal interrupt RINT0 must be set. The result, a 13 bit word, is stored temporarily in the internal RAM Block 1.

Students may choose different block lengths (16, 24, 32), implying additional instruction to identify each 8 bit word, but reducing the number of interrupts.

The RAM contains a table of 30 locations to store the actual 13 bit words for each channel. The number of buffer locations may differ, depending on the ability of the student to program a real time application and to avoid conflicts and wait states.

The results are transferred to the DAC by the DMA controller. The students have to program the DMA controller to read from an incrementing internal address and to write to a fixed external address, for 30 cycles. Because the system uses a single DAC, the analogue signal will be the analogue multiplex of 30 channels. That's why the timing of the writing step must be precise. The destination synchronization of the DMA transfer must be used.

TIMER 0 can generate the necessary frequency for the external circuits: 30 channels x 8 kHz = 240 kHz. It must be set to send the interrupt request TINT0 to the DMA controller.

NOTE:

In a real transmission system, the clock is recovered from the data flow and the output rate must be 30/32 of the clock frequency. This is not the point of this project, so an asynchronous clock can be used.

In Figure 3 the data flow is represented with big arrows (SERIAL PORT - CPU - RAM - DMA) and the internal interrupts with small arrows (RINT0, TINT0, DINT).

As in Example 1, the students have to set correctly the interrupt condition bits in the peripheral registers and in the CPU registers, to fill the interrupt vectors table, to set the stack pointer.

External Circuits

The external DAC latches a new 13 bits word each time the DMA controller performs a transfer at the address of the DAC, decoded by the address decoder A.D. The resulting PAM signal must be demultiplexed using analogue circuits, not represented in details. To obtain the voice channels, low-pass filters must be used after demultiplexing.

Example 3

The Task

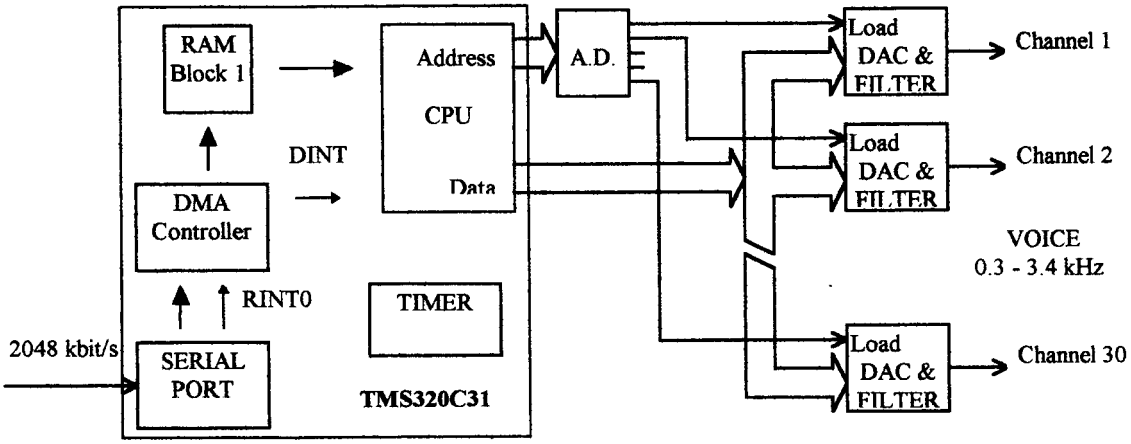
The third task is to design the RECEIVE terminal for the EUROPEAN standard, processing a continuous data flow representing ONE FRAME. The demultiplexing is performed in DIGITAL form, followed by individual digital-to-analogue converters. The analogue interface is controlled by the CPU.

In this case, the conclusions are similar to Example 2:

- 32 channels/frame, according to the European standard, with 8 bit words compressed according to the A-law,
- 2048 kbit/s input data,
- each of the 30 DAC's receive parallel 13 bit words, expanded according to the A-law, at the rate of 8 kHz.

A possible solution to implement the task is presented in Figure 4.

Figure 4. Receive Terminal with Individual DAC's for Each Channel



Internal Circuits

The data flow of 2048 kbit/s is received by the SERIAL PORT 0, as in the previous example.

The acquisition is performed, in this case, by the DMA controller. So, the interrupt request RINT0 will be used as signal for the source synchronization.

The internal RAM is storing the 8 bit compressed words, written by the DMA at 32 locations.

The CPU, after reading these locations, performs the A-law expansion program, elaborated by the student, for the 30 words corresponding to the voice channels. At the end of each processing loop, the 13 bit result is sent to the appropriate external DAC.

Because of the parallel digital-to-analogue conversion, each sample will last 125 μs. So the only timing requirement is to repeat the processes at the rate of 8 kHz. The internal interrupt DINT, indicating the frame rate, can be used to this end.

In this case, the TIMER may be used for replacing the unrecovered transmission clock or for generating the output transfer rate.

As in other cases, students have to set the interrupt system and to properly initialize the system.

External Circuits

The 30 DAC's must have different addresses, in order to be individually accessed. Students have to choose these addresses in the external memory space and to design the address decoder A.D. The decoded address acts as a *Load* signal and the 13 bits word from de data bus is latched in the corresponding converter. During a 125 μs period, all 30 converters receive new data. The analogue outputs must be low-pass filtered.

The point of all projects is the use of the TMS320C31, so the external circuits in all these examples are not detailed.

Conclusions

Results

Starting with the academic year 1993-1994, this method was used in the frame of a technical topic, not dedicated to the digital signal processors. Each year 50 to 80 students in Telecommunications had to solve an individual task, in digital communications, by using the TMS320C31.

My experience after several years, with more than 200 students, is positive, proving that designing applications with a DSP is possible, with good results, without an extensive study of the features of these processors. For this project, the students are able to deal with particular instructions, with particular registers and with particular bits to implement particular operation modes, when using the TMS320C3x. But after such a design, they will be ready to start any project, with any DSP.

The particular tasks of the projects are chosen for the Telecommunications students. But the teaching method can also be used, in appropriate topics, for the students in Power Electronics, in Instrumentation, in System Control. Every field can offer tasks to be solved with one member if the TMS320, as teaching target.

Notes

The paper presents a teaching alternative of the signal processor TMS320C3x, using digital communications as main applications to be designed.

Obviously, some of the tasks do not require all the computing power and all the features the TMS320C31 may offer. It is also generally known that applications such as those described are performed by specialized integrated circuits, better adapted to the application and at a lower cost.

The conclusion is that the solutions presented here do not try to compete with powerful applications with DSP’s or with dedicated communications integrated circuits. The applications presented are only proposed to students as their first project with a DSP, in order to LEARN how to use the great resources of the TMS320C3x.

Acknowledgement

This teaching experience would not be possible without the support of Max-Peter Gottlob, Texas Instruments European Customer Training Centre in Freising, and Tamas Biacs, Texas Instruments Office in Budapest.