Embedded System Emphasis in an Introductory Microprocessor Course

Robert B. Reese

Electrical & Computer Engineering Department
Mississippi State University
reese@ece.msstate.edu

Abstract

In Fall 2003 within the Electrical & Computer Engineering Department at Mississippi State University, the introductory microprocessors course was shifted from a traditional approach using X86 assembly language and a software-only lab to a microcontroller emphasis using the PIC18F242 with a mixed software/hardware-based lab experience. This was done to better prepare our computer engineering and electrical engineering majors for our senior project course, which usually includes a microcontroller as a component. Assembly language labs are specified as *C* programs, with the students acting as human compilers for PIC18 implementation. This removes the mystery of the *C* to assembly language link, and prepares the students for the hardware labs that are implemented entirely in *C*. The hardware labs cover the onboard peripherals of the PIC18F242 such as the timer subsystem, I^2C interface, and analog-to-digital converter, as well as off-chip interfacing to devices such as a serial EEPROM, an I^2C digital-to-analog converter, and an infrared receiver. The challenges in this course design have included finding the correct mix of assembly language/hardware lab topics, a textbook with this particular topic coverage, lab exercise development, and suitably trained teaching assistants for the lab experience. Student response to the new course material has been positive, and senior project quality has improved as students no longer struggle with including a microcontroller component in their designs.

Motivation

Beginning Fall 2003, the Electrical & Computer Engineering Department at Mississippi State University shifted its introductory microprocessor course from a traditional assembly language orientation (X86-based) to one that emphasizes embedded system concepts and hardware/software prototyping skills [1]. The course is required for computer science (CS), electrical engineering (EE), computer engineering (CPE), and software engineering (SWE) majors and is 4 credit hours (3 lecture, 1 lab). The course prerequisites are digital logic design and introductory computer programming in either *C* or *C++*. This also coincided with changes to both our CPE and EE curriculums. Figure 1 summarizes these curriculum changes relative to the Microprocessor I course that is the subject of this paper. In addition to the Microprocessors I content change, the VLSI I required course for Computer Engineering majors (CPE) was replaced with an embedded systems course that features internet-enabled microcontroller systems as it was felt that this would benefit a wider range of our CPE students. This change was possible because many of the microcontroller topics previously taught in the optional microcontroller interfacing course of the old curriculum are now taught in the new Microprocessors I course.

These changes were made for the following reasons:

1. The senior design capstone course is now a very important component of our ABET 2000 assessment process. In that course, students are expected to build something "real" over two semesters, where the first semester is used for prototype development, and the second semester for refinement and packaging. Almost all senior designs have a microcontroller component regardless of the application, and microcontroller interfacing was an elective course under the old curriculum. This meant that EE/CPE majors without that course faced a very steep learning curve. There was a startling difference in quality and success rates between design projects created by students who had previous microcontroller exposure and those who did not. It was obvious that some of our students were at a disadvantage in project prototyping by not having mandatory coverage of microcontroller topics before the senior design course.

2. At the time when these changes were being discussed (Spring 2003), the strawman version of the IEEE/ACM Curriculum Guidelines for Computer Engineering [2] had just been released and recommended 20 core hours of embedded systems. The Computer Engineering Steering committee at MSU felt that the required courses in the old curriculum did contain enough embedded systems topics.

3. There was a strong desire to include a significant hardware component to the Microprocessors I lab experience, and this was not possible with the X86-approach because of the increasing hardware complexity of the X86-based PC platform. In the past, some simple bused-based interfacing could be done with an ISA interface card and programs executed from the DOS shell that allowed direct access to hardware components. However, the rapid changes in the Windows operating systems and the PC platform eventually forced us to drop all hardware interfacing in that lab experience.

4. Negative comments were consistently made about the applicability of X86-based assembly language programming in senior exit interviews, as students did not use this in any follow-on course or in their senior design projects.

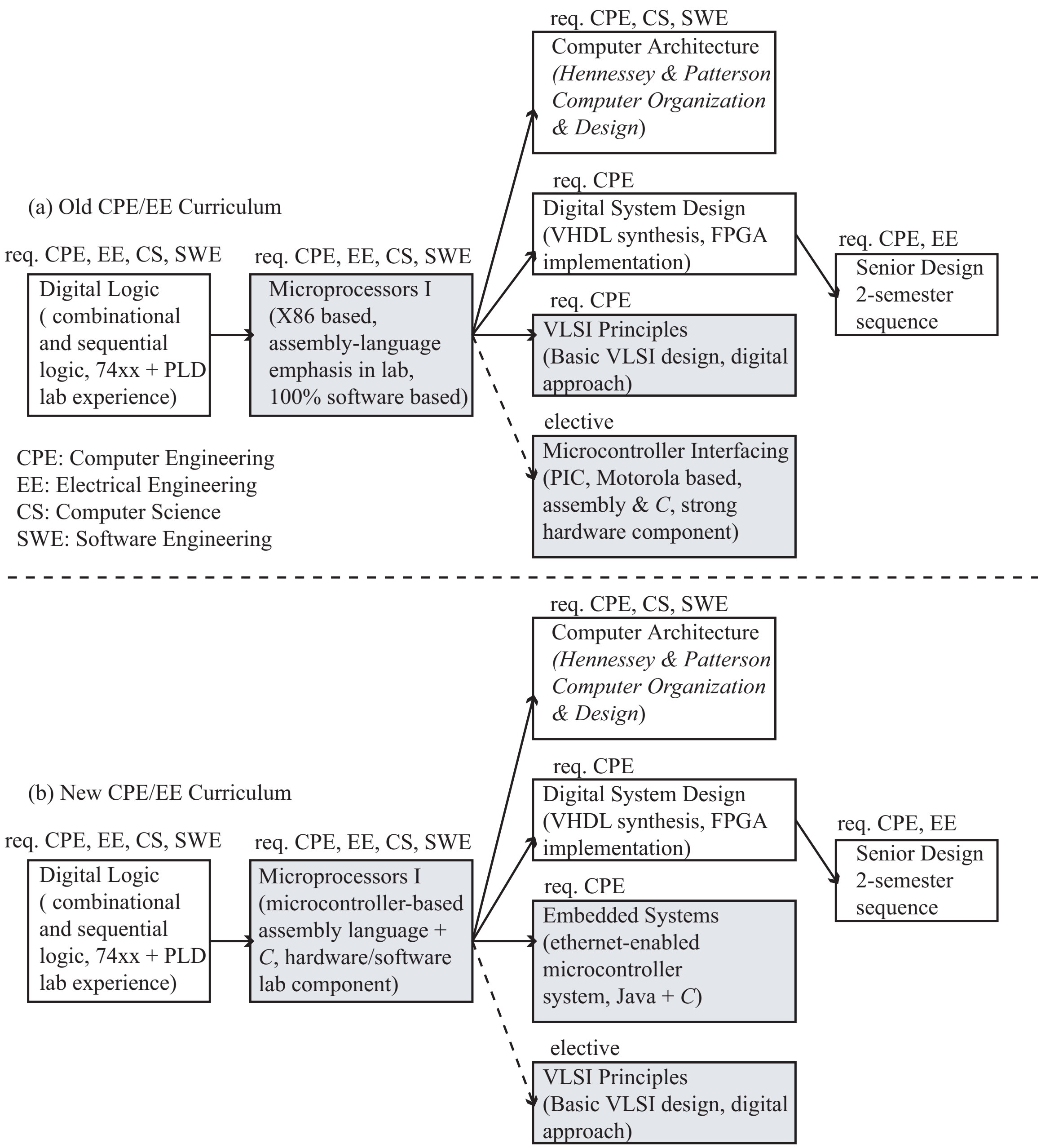

Figure 1 Old vs. New CPE/EE Curriculums.

Details

Table 1 compares the lecture and lab topics for the X86-based versus the microcontroller-based microprocessor courses. The microcontroller currently used is the PIC18F242, which comes in a 28-pin DIP package suitable for prototyping. In the new PIC18F242-

based course, students build a PIC18F242 system that contains an asynchronous serial interface, an external I^2C serial EEPROM, an external I^2C digital-to-analog converter, and a mini-jack input that allows an external PC to provide waveforms for sampling in the analog-to-digital converter lab experiments. A protoboard approach is used in which this system is built over a series of eight hardware labs. The parts kit for the labs including the protoboard and wiring kit cost $55 in Fall 2004.

The X86-based approach was first taught in the mid-90s (changed from a MC6800-based course) with the labs updated to the state of Table 1 in Spring 2000. In the X86-based course, a strong attempt was made to make the labs interesting by having them as interactive as possible in using the video, keyboard, and mouse inputs on the PC. However, this required a considerable amount of instructional effort in covering the DOS calls to these devices; effort that could be considered as wasted in that the lessons learned were peculiar to the DOS assembly language interface, which the majority of the students never used again after leaving this course. In senior exit interviews, many negative comments were received about the applicability of the X86-based material as students did not use this material in any follow-on courses or use these skills in their senior design projects. For the microcontroller-based course, negative comments about the lack of applicability of the material are no longer received. Instead, negative comments are usually about the amount of material covered (i.e., too much, too fast) and the debugging problems encountered in the hardware labs. However, these are offset by many positive comments when students reach senior design about how happy they are to have gained some microcontroller background before starting their projects. Comments have been received from seniors who have missed out on this material that they wished that they had been able to take the new microcontroller-based course. In a couple of cases, students have repeated the course simply because of the new content.

All of the hardware labs in the microcontroller-based course are done in *C* in order to reduce code complexity, improve code readability, and maximize understanding of the hardware topics being covered. The HI-TECH PICC-18 *C* compiler [3] is used for these labs. The assembly language labs are all specified as *C* language problems, with students having to play the part of a human compiler in converting them to assembly language. The coverage of *C* in these assembly language problems prepares them for the jump to *C* coding in the hardware labs, as well as making it clear the linkage between *C* statements/data types and their assembly language implementations. Each lab station has an oscilloscope, multimeter, and PC. The PC is used for assembling and compiling the assigned programs, or alternatively, a student can use their portable PC. A PICSTART programmer is shared between all lab stations and is used in the first hardware lab for programming the PIC18F242. Once the serial port is added to the board, a serial bootloader is used, allowing the PIC18F242 to be programmed in-circuit.

Table 1 Lecture/Lab Topics in X86-based/PIC18F242 Based Courses

Course	X86-based	PIC18F242
Lecture Topics	Assembly language topics focused on 16-bit subset. Considerable time spent on DOS software interrupts and utility functions for text IO because of the lab environment. Hardware topics included memory technologies, simple bus transfers (8088/8086), Serial IO (16550 UART), Interrupt vs. polled IO, floating point, disk storage basics, and video basics.	PIC18Fxx2 instruction set in *C* language context; 8-bit (`char`) vs 16-bit (`int`) data type operations, signed vs. unsigned operations, pointer (`*char`, `*int`) operations. Hardware topics include parallel port IO, LED/switch IO, timer usage, I^2C bus, ADC, DAC, interrupt driven IO vs. polled IO, power consumption in embedded systems.
Lab Topics	12 labs + practicum 1. DEBUG (hex/register intro) 2. DEBUG (data transfer) 3. MASM (assembler intro) 4. Addressing Modes 5. Arithmetic Operations 6. Subroutines, DOS calls 7. Text-based graphics 8. Bit-mapped graphics 9. Bit-mapped graphics (line drawing) 10. File IO 11. Interrupts (Mouse/keyboard, BIOS timer) 12. Floating point operations 13. Practicum	13 labs + practicum 1. Stored program machine introduction (Altera Maxplus) 2. MPLAB (assembler introduction, data movement) 3. 8-bit unsigned operations 4. 16-bit operations, signed ops 5. Subroutines, Pointers 6. Hardware startup (LED/switch IO) 7. Asynchronous Serial IO (USART) 8. Interrupt driven Serial IO 9. I^2C (Serial EEPROM) 10. ADC/DAC Intro (external I^2C DAC, onboard DAC) 11. Timers (waveform generation) 12. Timers (time measurement, IR waveform decoding) 13. Audio Sampling 14. Practicum

A picture of the protoboard at semester's end is shown in Figure 2. The LM386 audio amplifier and the CD4053 analog switch are used in the audio sampling lab that implements a digital voice recorder (lab 13). During playback, the 24LC515 serial EEPROM is used for storing voice samples that is digitized by the PIC18F242 on-chip ADC. The LM386 is used to amplify the voice that enters through the mini-jack input connected to a PC. During playback, samples are read from the 24LC515 serial EEPROM and converted back to an analog form by the MAX 517 DAC. Both the 24LC515 and MAX517 use the I^2C bus. The CD4053 analog mux allows the MAX 517 DAC to drive the mini-jack during playback; the mux setting is controlled by the

PIC18F242. The MAX202 is the RS232 transceiver chip, while the infrared (IR) receiver is used in the timer lab (lab 12) for decoding IR from a universal remote control. Use of the peripheral chips in Figure 2 such as the 24LC515 serial EEPROM and MAX 517 DAC are covered in lab exercises leading up to the digital voice recorder lab.

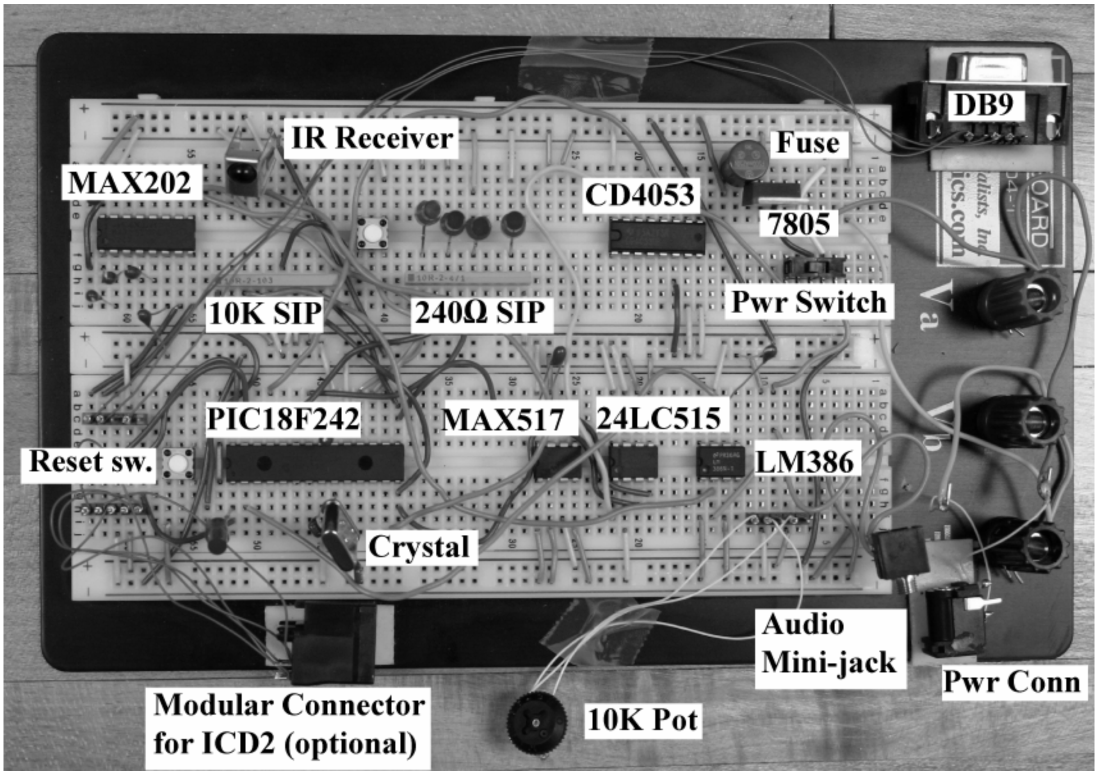

Figure 2 Completed Protoboard.

Challenges

There were several challenges in implementing this approach.

1. The lab Teaching Assistants (TAs) must be patient, dedicated, and knowledgeable in order to handle the debugging problems that arise during the hardware labs. Our first TAs were recruited from our traditional microcontroller course, and TAs are now recruited from our embedded system course. Step-by-step hardware debugging checklists have been created to assist students in developing a methodology for tracking down the inevitable hardware problems that occur.

2. Developing the right tradeoff between assembly language and hardware interfacing labs was difficult and is still an ongoing process. There are more hardware interfacing labs than assembly language labs as this is the only required course for EE majors in microprocessors, and the hardware interfacing labs are what prove to be the most useful to EE majors in the senior design course. Assembly language programming is reinforced for both computer science and

computer engineering majors in the computer architecture course, in which students perform some assembly language programming for the MIPs architecture [5]. The follow-on embedded systems course that is required of CPE majors reinforces the hardware interfacing topics. The embedded systems course also emphasizes memory technologies and external parallel bus protocols as these topics are only lightly touched upon within the Microprocessors I course.

3. Finding a textbook with the right topic coverage has proven difficult. An excellent text on using the PIC18F242 is [6]. Unfortunately, this text uses assembly language programming for all examples except in the appendix. After teaching the course for two semesters (Fall 2003/Spring 2004), the author developed a new text [4] that has the desired assembly/*C* emphasis. The course has been taught from an early version of this book in Summer 2004/Fall 2004, and will be taught again in Spring 2005.

Conclusions

The change in Microprocessor I course material from X86-based assembly language, 100% software lab experience to a microcontroller-based assembly/*C* language and mixed software/hardware experience has been in place at Mississippi State University for approximately four semesters. This better prepares both EE and CPE majors for senior design by providing them with needed skills in microcontroller programming, interfacing, and prototyping. This has also allowed our second course in embedded systems to begin at a higher level in emphasizing internet-enabled embedded systems.

Bibliography

[1] Microprocessors I Course Material, http://www.ece.msstate.edu/~reese/ece3724.

[2] Computing Curricula for Computer Engineering Joint Task Force, Computer Engineering 2005: Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering, IEEE Computer Society/Associate for Computing Machinery, 2004 October 12. Available online at *http://www.eng.auburn.edu/ece/CCCE/CCCE-FinalReport-Complete-2004Oct12.pdf.*

[3] HI-TECH Software, PICC-18 ANSI C Compiler. Available at *http://www.htsoft.com*.

[4] R. Reese, *Microprocessors: From Assembly to C using the PIC18Fxx2*, Charles River Media, ISBN 1-58450-378-5, Spring 2005, *http://www.charlesriver.com*.

[5] David A. Patterson, John L. Hennessy, *Computer Organization & Design: The Hardware/Software Interface 2nd edition*, Morgan Kaufmann, ISBN 1-55860-428-6, 1998.

[6] John B. Peatman, Embedded Design with the PIC18F452, Prentice Hall, ISBN 0-13046-213-6, 2002.

Biographical Information

ROBERT B. REESE received the Ph.D. degree in electrical engineering from Texas A&M University, College Station, in 1985. Since 1988, he has been with the Department of Electrical and Computer Engineering at Mississippi State University, where he teaches courses in VLSI systems and microprocessors. His research interests include self-timed digital systems and computer architecture.