An Advanced Microcontroller Systems Course for Upper-Level Undergraduate Curriculum

Teofilo A. Caceres, Zachary Combs and James A. Ochoa
Texas A&M University

Abstract

The Electronics Engineering Technology program at Texas A&M University prepares undergraduate students to "hit the ground running" in industry engineering positions. The program has a strong system-integration curriculum emphasizing the theory and application of both analog *and* digital electronics. Recognizing the popularity and importance of embedded processors, the EET program offers a Microcontroller Systems course (ENTC 349) that focuses on microcontroller architecture and interfacing.

Traditional microcontroller/microprocessor courses focus considerable attention on architectures, instruction sets and addressing modes for simple devices like the Motorola 68000 or Intel 8051/8088. However, these devices provide only simple interfacing resources and lack the ability to simultaneously connect to multiple sensors and actuators. Often, students are limited to integrating a parallel port and UART to the processor. In contrast, students in ENTC 349 work with the powerful 32-bit Motorola 68332 microcontroller.

The 68332 microcontroller is a multi-module processor with on-chip CPU, RAM, a serial communication interface, a serial peripheral interface, a time processing unit, and a system integration module. In the first several weeks of the semester, students study the processor's overall architecture and then focus on the CPU (i.e., instruction set and addressing modes). Students then explore the architecture and operation of each of the other modules. As the modules are covered in lecture, students are given laboratory problems that involve interfacing the 68332 to external circuitry. The individual labs lead up to a final project that is completed during the last four weeks of the semester. The project requires interfacing the microcontroller to multiple external devices (incl. analog-to-digital converter, H-bridge motor controller, etc) and creating a real-time user interface and motor control program.

This paper will discuss the course structure and will provide a detailed description of the final project, including required hardware and software resources.

I. Introduction.

The Electronics Engineering Technology Program at Texas A&M University teaches an advanced microcontroller systems course, ENTC 349: Microcontroller Systems, which utilizes a complex Motorola microcontroller. Students in the course learn about microcontroller architecture, instruction sets, and addressing modes as they would in most microcontrollers classes. The difference at Texas A&M University is that instead of using a simple microcontroller, such as the MC68000 or Intel 8051/8088, students are using the more complex MC68332. This powerful 32-bit microcontroller has a multi module complex architecture, larger instruction set, and a rich set of addressing modes. This allows students to be exposed to a broader set of capabilities that today's microcontrollers have. In the course lecture the students learn about the different modules, which they use in laboratory. The laboratory assignments include topics covering communications devices, interrupt handling, and interfacing to external devices. At the end of the course students will complete a final project that makes use of all modules and all capabilities previously demonstrated in lab. This project provides students a glimpse of what microcontrollers are capable of.

II. Course Overview

The course has a close relationship between the lecture and the laboratory. In lecture students learn how the MC68332 operates and in the laboratory students apply this understanding to gain practical applied knowledge. During the first few weeks of the semester, the students are taught the architecture and the operations of the CPU while the laboratory teaches students, assembly programming, debugging, and how to communicate with the MC68332 through the use of a resident debug program. Following introductions, the class splits the MC68332 into several pieces: the Periodic Interrupt Timer (PIT), the Serial Communication Interface (SCI), the Time Processor Unit (TPU), and Serial Peripheral Interface (SPI).

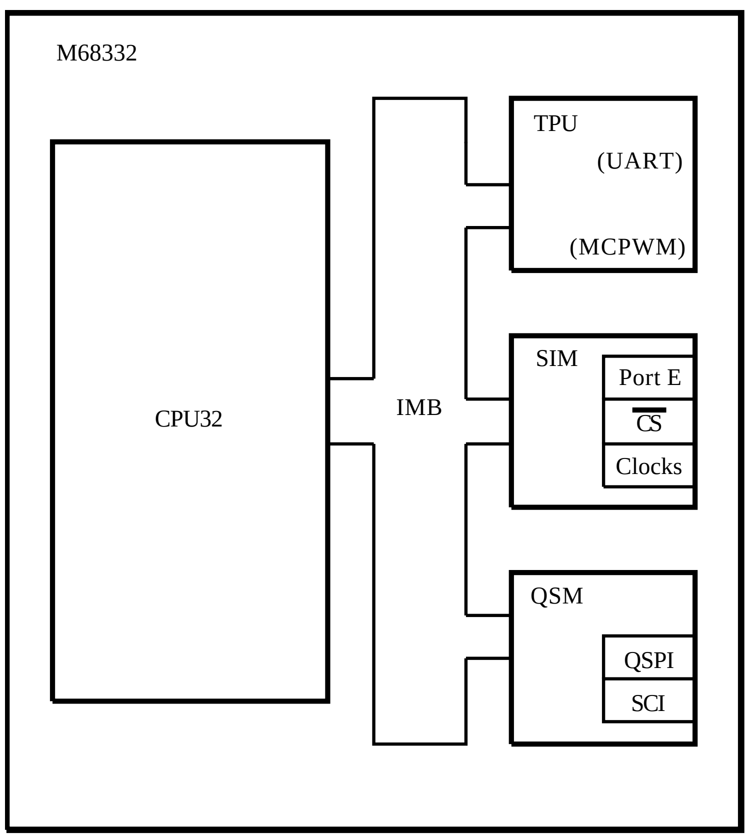

Figure 1: Overall MC68332 resources used in ENTC 349

The overall microcontroller *architecture* is the first topic introduced in the class. Following this, each of the modules are highlighted and are explained in terms of their role in the microcontroller. The MC68332 has four modules including the Central Processing Unit (CPU), the System Integration Module (SIM), the Time Processing Unit (TPU), and the Queued Serial Module (QSM) that includes the SCI and the SPI. Along with these modules is the Internal Module Bus (IMB) that allows for communication between modules.

The *operation* of the CPU is the second topic covered in the course. This section focuses on the instruction set and addressing modes of the MC68332. Students learn how to assemble and disassemble various instructions and addressing modes. Interrupts are also discussed in this section including concepts such as CPU interrupt processing, interrupt arbitration, and interrupt handling.

The third topic covered in the course is *Periodic Interrupt Timer* (PIT) within the M68332. This area builds on interrupt handling discussed previously by introducing students to time based interrupts. The PIT provides the real-time capability needed for the final project.

The goal of the next section is to provide the students an understanding of asynchronous serial communications using the *SCI*. Students develop a layered driver program, that interfaces the SCI at a hardware level, buffers the data, and provides the software interface to the buffer. This driver set is then used in a character echo application to demonstrate capability. This driver set will also be used in the final project.

The *TPU* module follows the SCI because the first lab assignment of this section is to use the TPU UART function to create a driver set similar to the SCI driver set. In doing so students learn the operation of the TPU. The next area in this topic is the Multi Channel Pulse Width Modulation (MCPWM) function of the TPU. The students learn briefly about what the MCPWM signal is and then write a program that implements MCPWM.

The next topic is the *serial peripheral interface* (SPI), which is a synchronous structure used to communicate with external devices. Students learn the different registers and configurations necessary to use an external device to gather data. In lab the students will use the SPI to interface to an analog to digital converter. This assignment is also used in the final project to gather information. This will complete the closed loop characteristic in the final project, as the MCPWM sends out signals to the environment the SPI will gather information about the environment.

The final topic is the *Final Project*. Students have a final project in which they utilize all of the modules within the MC68332 to implement a closed-loop motor speed controller. This is where the class brings together the knowledge gained in the classroom and the hard work in the laboratory. Students learn what a microcontroller can be used for that what it can do. Through this project, students also learn team dynamics and project management. This is not explicitly taught but rather is inherited by requiring students to select a team manager and turn in Gantt charts and updates on the progress of the project.

III. Final Project

The purpose of the final project is to use the MC68332 to provide real-time control of a DC motor with a user interface and a "debug" terminal on separate computers. Students are required to use motor tachometer to provide feedback to the controller. In addition the students are to program the motor control software to compensate for a load placed on the motor.

One valuable aspect of this project is that it requires the use of all communications interfaces of the 68332 using driver sets written by the students. It also requires the students to assemble external amplification and conversion circuitry to interface the motor and implement a closed loop control.

Students operate the motor using a Pulse Width Modulation (PWM) in the TPU and an external H-Bridge to amplify the signal. The tachometer provides feedback in the form of a signal with a varying frequency. This signal is converted to an analog voltage using an external frequency to voltage converter. This voltage was digitized using a 10-bit analog to digital converter. This data is communicated back to the microcontroller using the SPI interface.

The H-Bridge provide a brake and direction function which is controlled using a general-purpose digital port interface. It also provides a thermal signal indicating that the H-Bridge is overheating. This signal is read by another general purpose I/O, which triggers an auto vectored interrupt to stop the motor.

The communication medium employed by the students to communicate with the user and debug terminals used RS-232. Interrupt driven driver sets were written to create and manage a FIFO buffer and interface the hardware. There is, however, only one SCI on the MC68332. So the students configure the TPU UART function and use it as the second communication device. This created the need for a level converter to properly communicate using RS-232, which they also interfaced.

In order to provide real-time control of the motor a Periodic Interrupt Timer (PIT) was used to generate interrupts at predetermined time intervals to sample data returned by the A/D converter. This real-world data would then be compared to the desired speed of the motor and adjustments were to be made to the PWM output. However, the students were instructed to create a ramping function to change the speed of the motor so as to minimize current spikes caused by abrupt changes to the voltage input at the motor.

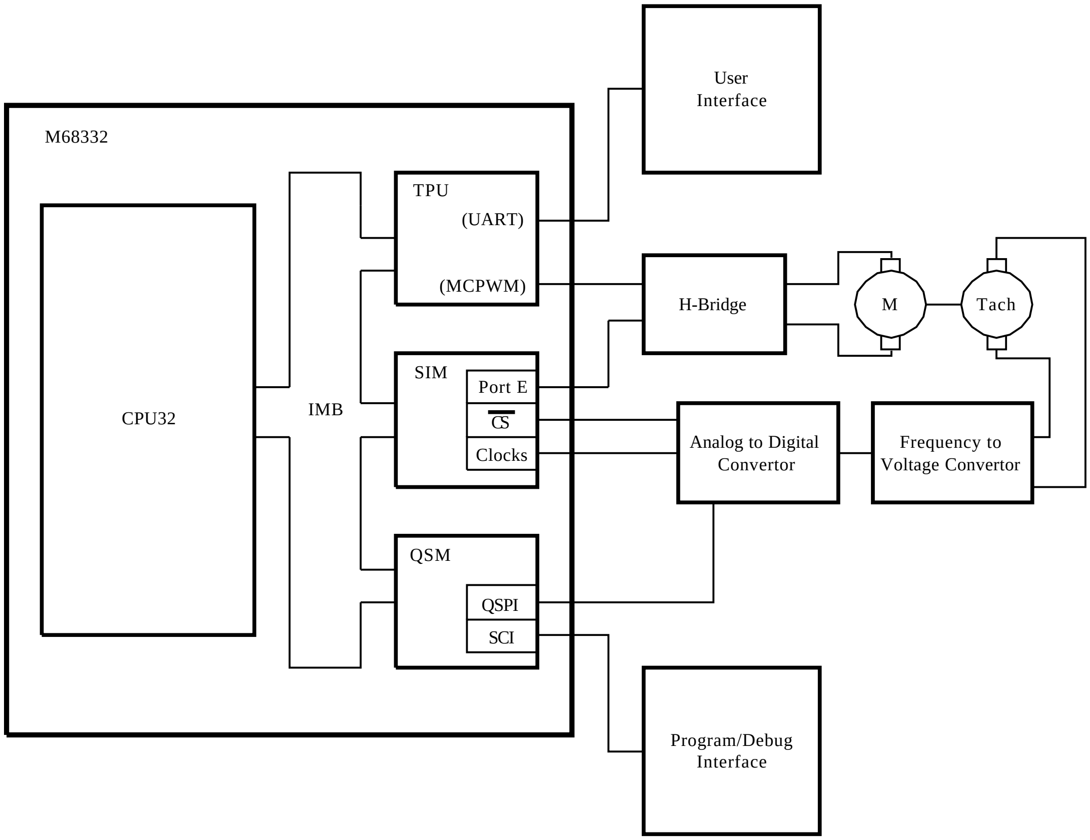

Figure 2: Final Project System Diagram

In the Fall 2001 semester, students were given the opportunity to experiment with wireless communication as an alternative medium to the standard RS-232. This proved to be a success as students interfaced the MC68332 to a wireless radio board. In doing so, the students found the need for an additional microcontroller, creating a network of microcontrollers.

With the addition of a second microcontroller, students were presented with the option of having the user interface controller receive keyboard commands from the user, process them, and send single commands to the microcontroller responsible for executing those commands or having the microcontroller on the user interface side act as a dumb device by only relaying user input. The students chose the later option and demonstrated a large degree of success with the wireless communication.

During the experimentation process students experienced some of the reasons for using handshaking techniques to reduce the effects of packet loss over the wireless medium. During testing, the students experienced a seemingly random loss of packets during communication. Through use of protocol analyzers, the problem was determined to be dropped packets. This was due either to poor signal quality or CRC errors resulting in the packets being dropped.

IV. Conclusion

The course provides an in depth study of the MC68332 architecture, covering all of its communications devices, interrupt handling, and interfacing external devices. All topics covered give the students a working knowledge of the fundamentals of microcontroller operation that can be applied to other architectures other than Motorola architecture.

Bibliography

1. The Motorola MC68332 Microcontroller, T.L. Harman, Prentice-Hall, 1991.
2. MC68332 User Manuel, Motorola, Inc.

TEOFILO CACERES
Teofilo is an undergraduate student in the Electronics Engineering Technology program at Texas A&M University. He is currently an undergraduate teaching assistant for the Department of Engineering Technology at Texas A&M University in the class ENTC 349. Mr. Caceres can be contacted at teofiloc@tamu.edu.

ZACHARY COMBS
Zachary is an undergraduate student in the Electronics Engineering Technology program at Texas A&M University. He is currently working as a research assistant whose duties include maintenance of Exchange Server and research lab maintenance. He also is currently working on the development of an embedded control system with Internet capabilities. Mr. Combs can be contacted at zach_combs@yahoo.com.

JAMES A. OCHOA
James Ochoa received a BS degree in electrical engineering at Texas A&M University – Kingsville in 1990 and a Ph.D. in electrical engineering at Texas A&M University in 1999. After completing his Ph.D., he joined the faculty in the Department of Engineering Technology at Texas A&M University. His research activities include digital electronics and circuit testing, system-on-a-programmable-chip, and micro-controller based control systems. Dr. Ochoa can be contacted via email at j-ochoa@tamu.edu.