Washington State University
School of EECS
Electrical Engineering Course Assessment Report

Course Number    EE 234
Course Title     Microprocessor Systems
Semester Offered Spring 2007
Instructor       Andrew O'Fallon
10th Day Enrollment 60 Number Completing Successfully (C grade or better) 52

I. Assessment Outcomes from the Course Syllabus – check those that apply

☒ (a) An ability to apply knowledge of computing and mathematics appropriate to the discipline
☒ (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
☒ (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
☐ (d) An ability to function effectively on teams to accomplish a common goal
☐ (e) An understanding of professional, ethical, and social responsibilities
☐ (f) An ability to communicate effectively
☐ (g) An ability to analyze the impact of computing on individuals, organizations, and society, including ethical, legal, security, and global policy issues
☒ (h) Recognition of the need for and an ability to engage in continuing professional development
☒ (i) An ability to use current techniques, skills, and tools necessary for computing practice.

II. List of Course Topics from the Course Syllabus

1. Microprocessor System Architecture
2. Assembly Language Programming
3. Assembler Directives
4. AVR Instruction Set and Instruction Mnemonics
5. Memory Organization and Memory Addressing Modes
6. AVR Interrupt Architecture and Interrupt Service Routines
7. Interfacing between C and Assembly Language
8. Stack Operations
9. Basic Algorithms and Data Structures
10. Peripheral Device Programming
11. Microprocessor Operation, Bus Cycles and Bus Timing
12. Memory and I/O Address Decoding
13. Intel 8086 and Later Processors Basic Architecture
### III. Course Assessment Summary Table: one row of the table should be devoted to each of
the checked outcomes in part I.

<table>
<thead>
<tr>
<th>Outcome</th>
<th>Topics</th>
<th>Specific Measures (These should be available in the course materials file for inspection.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>List the outcome letter name and full description</td>
<td>List topic numbers and/or names</td>
<td>List specific exams, homeworks, projects, presentations in which students demonstrate their achievement of this outcome</td>
</tr>
<tr>
<td>(a) An ability to apply knowledge of computing and mathematics</td>
<td>1 - 13</td>
<td>Exam 1: Understanding addressing modes and basic instruction set operation, Understanding operation of interrupt mechanism, Use of dumps of memory and processor registers to predict program behavior and stack contents, Implement simple subroutine in assembler given a detailed specification (Avg: 73)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Exam 2: General and external interrupt questions, Implement timers and PWM, Implement serial communication and USART control, Analog comparators, Analog to Digital converters (Avg: 71)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Final Exam: Understanding addressing modes and basic instruction set operation, Understanding operation of interrupt mechanism, Use of dumps of memory and processor registers to predict program behavior and stack contents, Implement simple subroutine in assembler given a detailed specification, General and external interrupt questions, Implement timers and PWM, Implement serial communication and USART control (Avg: 76)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lab 1: Use of programming tools</td>
</tr>
<tr>
<td>Lab 2: Manipulating I/O ports, peripheral modules, and bit masking (Avg: 90)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lab 3: String handling and subroutines (involving branch and jump instructions, pointers) (Avg: 80)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lab 4: Computer simulator for robotics applications (Avg: 81)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lab 5: Postfix calculator using external interrupts (Avg: 80)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lab 6: Implementing timers/counters to drive sounds from a speaker module (Avg: 85)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lab 7: Dynamic memory management. Creation of dynamic memory management system (Avg: 70)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lab 8: Serial communication and implementation of robotics language transferring between PC and microcontroller (Avg: 91)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| HW 1: Binary arithmetic, addressing modes and data transfer instruction operation, basic assembly language syntax (Avg: 87) |
| HW 2: Basic assembly language syntax, use of data transfer instruction, use of arithmetic and logical instructions (Avg: 83) |
| HW 3: Stack manipulation, program control flow (Avg: 65) |
| HW 4: Figure skating and shape tracing with robotic cars (Avg: 94) |
| HW 5: Line following with robotic cars (Avg: 91) |

(b) An ability to analyze a problem, and identify and define the
computing requirements appropriate to its solution

<table>
<thead>
<tr>
<th>Outcome Description</th>
<th>Score</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs</td>
<td>1 - 13</td>
<td>Same as above</td>
</tr>
<tr>
<td>(h) Recognition of the need for and an ability to engage in continuing professional development</td>
<td>1 - 13</td>
<td>Same as for (a). The applications that are supplied in this course require knowledge of programming, physics, and basic control systems</td>
</tr>
<tr>
<td>(i) An ability to use current techniques, skills, and tools necessary for computing practice.</td>
<td>1 - 13</td>
<td>Same as for (a)</td>
</tr>
</tbody>
</table>

IV. Using the table as a guide, for each outcome summarize your evaluation of the students’ achievement of that outcome; cite student performance on the identified measures as evidence to support your conclusions.

The assignments provided required students to think critically about how to solve problems given a problem description. On the exams, labs, and homeworks (as listed in the table above) most students received C’s or higher.

Problems included implementing standard C string library functions in assembly. Implementing linked lists and heap structures. Implementing assembly programs for driving speakers, serial communication, and robotics cars.

V. Qualitative Assessment of Student Performance: using the arguments above and other data support the claim that students who completed this course with a grade of C or better have achieved each of the intended outcomes of this course.

EXAMS

The students were allowed to use some notes for the first midterm. The second midterm and the final were open book, open note. This was to allow them to look up a detail that they may have forgotten in the behavior of the instructions. In order to complete these problems in the allowed time, the students required a good understanding of the basic behavior of the machine. I believe that the students this semester performed at a consistent or even higher level than with previous semesters.

The problems on the first exam stressed being able to understand sequences of instructions and to follow the data though the steps of processing. They were asked to examine memory dumps and a fragment of code and predict the contents of the stack and the behavior of the program. They were also asked to design and implement a simple subroutine (approximately 15 lines of code for a good solution) to a detailed input/output specification.

The second exam emphasized instruction execution time, instruction set encoding, and the operation of the hardware (timers, USART, etc). The students were asked to design small fragments of assembly language to produce USART communication. This required writing a
delay timing loop. There were also questions to demonstrate basic understanding of instruction set operation and system bus cycle timing.

The final exam was comprehensive and contained problems similar to those on the first two midterms. In addition, the final covered some concepts of 80x86 processors. More time is allowed for the final exam, and the exam was not designed to be as challenging relative to the amount of time allowed. Also by this time a number of marginal students had dropped the class, and so the average grade was much higher with a lower standard deviation. Most of the students demonstrated acceptable to very good understanding of the required material.

As far as I could tell students worked at the same level this semester as with previous semesters.

HOMEWORK
Homework assignments were designed to show the kinds of problems that would be asked on the exams and give the students practice working those kinds of problems. However, some of the homework required that students apply understanding of the material to robotic cars. They really enjoyed and excelled with homework 4 and 5.

LABORATORY

Lab 1: Use of programming tools (Avg: 85)

The first lab project is an introduction to use of the tools, assembly language syntax and operation of memory addressing modes. In this exercise, the students are given some code and answer various questions about its operation by observing it using the debugger.

Lab 2: Manipulating I/O ports, peripheral modules, and bit masking (Avg: 90)

The second lab project required that students learn how to interact with the microcontroller and peripheral devices.

Lab 3: String handling and subroutines (involving branch and jump instructions, pointers) (Avg: 80)

The third lab project introduces the use of data transfer instructions and basic control constructs. The students write several subroutine libraries of general purpose buffer and character string manipulation routines similar in nature to many of the functions in the C runtime library.

Lab 4: Computer simulator for robotics applications (Avg: 81)

Students had to develop a machine language that would eventually control a robotic car.

Lab 5: Postfix calculator using external interrupts (Avg: 80)

Students developed a postfix calculator that received input via interrupts of a switch peripheral module.

Lab 6: Implementing timers/counters to drive sounds from a speaker module (Avg: 85)

Students used the timer/counter PWM function to make sounds with a peripheral speaker module. Some were able to create “twinkle twinkle little star”
Lab 7: Dynamic memory management. Creation of dynamic memory management system (Avg: 70)

The seventh project has the students implement a dynamic memory heap and write a memory manager similar to the ‘malloc’, ‘realloc’ and ‘free’ heap routines in the C runtime library.

Lab 8: Serial communication and implementation of robotics language transferring between PC and microcontroller (Avg: 91)

This required that students create an assembly program that allowed for robotic machine language programs to be downloaded to the microcontroller via the RS232 module and USART.

The lab projects are designed to provide the students with significant real programming projects. It isn’t possible to learn assembly language programming without sufficient practice and these projects provide the students the opportunity to get sufficient practice in assembly language programming. Additionally, they are exposed to the low level details and implementation of subsystems that are fundamental to some of the runtime library systems in high level languages.

The students generally do well on the laboratory portion of the class. They have essentially unlimited time to work on the projects. It is through the directed, trial and error process of getting actual code to work that they develop real understanding of how the various parts of the computer work and develop actual mastery of the topics covered in the class. The student performance on the lab projects this semester was consistent with that of previous semesters.

VI. Concerns: state any concerns you may hold about this class – were the students adequately prepared coming into it? Are there topics or outcomes where (some) students were weak after completing the course? Other concerns? Were there any comments on students’ course evaluations that should be addressed in future instances of the course? This section is very important for improving our program: it provides critical input to the curriculum committee for identifying areas requiring attention.

Overall the success of the course was on par with previous semesters. Students really have seemed to enjoy applying these concepts to the robotics car.

Signature _Andrew O’Fallon_________________________ Date: _____ 5/10/07______