Washington State University
School of EECS
Electrical Engineering Course Assessment Report

Course Number  EE 234
Course Title  Microprocessor Systems
Semester Offered Fall 2006
Instructor  Andrew O'Fallon
10th Day Enrollment 49  Number Completing Successfully (C grade or better)  37

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. Microcontroller System Architecture
2. Data Representations
3. AVR Architecture
4. Memory Organization and Memory Maps
5. Data and Program Addressing Modes
6. AVR Assembly Instruction Set
7. Bit Masking and Stack Operations
8. AVR Interrupt Architecture and Interrupt Service Routines
9. Timers/Counters
10. Comparators
11. Analog to Digital Converters
12. Peripheral Device Programming
13. Basic Algorithms, Data Structures, and Algorithm Tracing
15. Basic Robotics, Figure Skating, and Line Sensing
### 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>
</tbody>
</table>
| (a) An ability to apply knowledge of computing and mathematics appropriate to the discipline | 1 - 14                                      | 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%)

Exam 2: Count clock cycles and determine execution time for sequence of instructions, Describe microprocessor control signal states for different types of bus cycles, Implement simple control subroutine in assembler given timing specification for required output, Implement solutions to control timers, comparators, and digital to analog comparators (Avg: 76%)

Final Exam: Understanding addressing modes and basic instruction set operation, Count clock cycles and predict execution time, Use dumps of memory and registers to predict program behavior and stack contents, Implement simple subroutine in assembler given a detailed specification, Understanding bus cycle timing Implement solutions to control timers, comparators, and digital to analog comparators (Avg: 73%) |
<table>
<thead>
<tr>
<th>Lab 1: Use of programming tools (Avg: 85%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab 2: Port Interfacing and basic interrupts (Avg: 83%)</td>
</tr>
<tr>
<td>Lab 3: String handling library functions (Avg: 91%)</td>
</tr>
<tr>
<td>Lab 4: Dynamic memory management. Creation of dynamic memory management system (Avg: 91%)</td>
</tr>
<tr>
<td>Lab 5: Linked List Management. Creation of subroutine library for management of doubly linked list structures (Avg: 72%)</td>
</tr>
<tr>
<td>HW 1: Binary arithmetic, addressing modes and data transfer instruction operation, basic assembly language syntax, basic ATmega64L features (Avg: 85%)</td>
</tr>
<tr>
<td>HW 2: Basic assembly language syntax, use of data transfer instruction, use of arithmetic and logical instructions High precision multiplication and division, algorithms (Avg: 82%)</td>
</tr>
<tr>
<td>HW 3: Memory windows, program tracing, calculation of addresses(Avg: 80)</td>
</tr>
<tr>
<td>HW 4:Robotics design (Credit for participation)</td>
</tr>
<tr>
<td>HW 5: Figure skating with robot (Credit for participation)</td>
</tr>
<tr>
<td>HW 6: Line following and competition (Credit for participation)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>2 - 11</th>
</tr>
</thead>
</table>

Same as above
| (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs | 1 - 14 | Same as above |
| (d) An ability to function effectively on teams to accomplish a common goal | 2 - 11 | Applies to labs mentioned above and HW4 - 6 |
| (f) An ability to communicate effectively | 2 - 11 | Applies to labs mentioned above and HW4 - 6 |
| (h) Recognition of the need for and an ability to engage in continuing professional development | 1 - 14 | Same as for (a) |
| (i) An ability to use current techniques, skills, and tools necessary for computing practice | 1 - 14 | Same as for (a) |

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 routines for robotics applications (using interrupts, sensors, timers, and USARTs)

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
For the first exam students were allowed 1 side of one standard sheet of paper only. For the second exam and final students were allowed to use the class text book, an AVR Instruction Set Reference and an instruction set timing reference during the exams. 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 level or higher than with previous semesters.

The problems on the first exam stressed being able to understand sequences of instructions and to follow the data through 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. The students were asked to write assembly fragments to perform interrupt processing and to control hardware devices such as USARTs, timers, and digital to analog conversion. There were also questions to demonstrate basic understanding of instruction set operation, instruction execution times, 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 Intel 80x86 processor instructions. Students were asked to describe various aspects of the Intel microprocessor. 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. The homework questions asked this semester were very similar to those asked on previous semesters and the performance on these assignments was very consistent with previous semesters. However I did add more hands on experience with robotics applications.

LABORATORY
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. The students also seemed to enjoy the robotics applications more than other applications provided in the past.

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. At times electrical engineering students seem to struggle with very basic assembly programming concepts than other computer science and computer engineering.

Signature  Andrew O’Fallon_________________________ Date: _____1/2/07_______