Course Number & Name: CSC 221 Computer Systems and Architecture

Credit Hours: 4.0    Contact Hours: 4.0    Lecture: 4.0    Lab: N/A    Other: N/A

Prerequisites: Grade of “C” or better in CSC 122

Co-requisites: None    Concurrent Courses: None

Course Outline Revision Date: Fall 2010

Course Description: This course provides a general introduction to the structure of computer systems and covers Assembly language for a specific computer. Topics discussed include machine components and cycles, assemblers, addressing techniques, macros, subroutines, program linkage, and input/output. A specific Assembly language is developed and implemented. Students must be prepared for extensive individual work in the computer laboratory.

Course Goals: Upon successful completion of this course, students should be able to do the following:

1. describe the basic hardware component of the digital computer;
2. describe the fetch-and-execute cycle as it is implemented in the hardware;
3. explain and use the binary and hexadecimal numbering systems;
4. express circuit function in terms of Boolean expressions; and
5. design, implement and test combinational and sequential circuits.

Measurable Course Performance Objectives (MPOs): Upon successful completion of this course, students should specifically be able to do the following:

1. Describe the basic hardware component of the digital computer:
   1.1 define terminology and explain concepts related to hardware;
   1.2 analyze hardware specification from advertisements; and
   1.3 draw schematic diagrams of the components and show their relationships

2. Describe the fetch-and-execute cycle as it is implemented in the hardware:
   2.1 use micro-computer instruction to describe the process; and
   2.2 trace the data path on logic diagrams to implement the process
Measurable Course Performance Objectives (MPOs) (continued):

3. Explain and use the binary and hexadecimal numbering systems:
   3.1 explain the digital nature of the modern computer;
   3.2 convert integers into binary representation; and
   3.3 convert floating point numbers into IEEE standard format

4. Express circuit function in terms of Boolean expressions:
   4.1 use Karnaugh maps to derive a Boolean function starting from a problem statement; and
   4.2 simplify Boolean expressions into sum of product form

5. Design, implement and test combinational and sequential circuits:
   5.1 compare and contrast combinational and sequential circuits;
   5.2 draw diagrams of various circuits including counters, multiplexers, decoders, and binary cells; and
   5.3 draw some basic ALU circuits such as binary adders including full and half

Methods of Instruction: Instruction will consist of lectures, projects and library assignments, and circuit diagrams.

Outcomes Assessment: Homework and exam questions are blueprinted to course objectives. Rubrics are used to evaluate the projects including the library assignment for the presence of course objectives. Data is collected and analyzed to determine the level of student performance on these assessment instruments in regards to meeting course objectives. The results of test and project analysis are used to guide necessary adjustments in the presentation of the material.

Course Requirements: All students are required to:
1. Maintain regular attendance and take part in class discussions.
2. Complete assigned homework and projects on time.
3. Take all exams as scheduled.

Methods of Evaluation: Final course grades will be computed as follows:

<table>
<thead>
<tr>
<th>Grading Components</th>
<th>% of final course grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Homework, quizzes, class participation, and attendance</td>
<td>10%</td>
</tr>
</tbody>
</table>

A perusal of homework problems and quizzes will provide evidence of student mastery of course objectives. Attendance and class participation are necessary for students to benefit from the guidance of the instructor.
Methods of Evaluation: Final course grades will be computed as follows:

<table>
<thead>
<tr>
<th>Grading Components</th>
<th>% of final course grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>• 9 or more projects including a library assignment</td>
<td>35%</td>
</tr>
<tr>
<td>(dates specified by the instructor)</td>
<td></td>
</tr>
<tr>
<td>Projects will show evidence of the extent to which</td>
<td></td>
</tr>
<tr>
<td>students meet course objectives. Students should show</td>
<td></td>
</tr>
<tr>
<td>that they have synthesized a combination of concepts</td>
<td></td>
</tr>
<tr>
<td>(See page 6 for suggested projects.)</td>
<td></td>
</tr>
<tr>
<td>• Midterm Exam</td>
<td>25%</td>
</tr>
<tr>
<td>The same objectives apply as with projects, but it is</td>
<td></td>
</tr>
<tr>
<td>anticipated that students will show mastery of content</td>
<td></td>
</tr>
<tr>
<td>covered in the first half of the course.</td>
<td></td>
</tr>
<tr>
<td>• Final Exam</td>
<td>30%</td>
</tr>
<tr>
<td>The same objectives apply as with the midterm exam, but</td>
<td></td>
</tr>
<tr>
<td>it is anticipated that students will provide increased</td>
<td></td>
</tr>
<tr>
<td>evidence of synthesizing a combination of concepts</td>
<td></td>
</tr>
</tbody>
</table>

Academic Integrity: Dishonesty disrupts the search for truth that is inherent in the learning process and so devalues the purpose and the mission of the College. Academic dishonesty includes, but is not limited to, the following:

- plagiarism – the failure to acknowledge another writer’s words or ideas or to give proper credit to sources of information;
- cheating – knowingly obtaining or giving unauthorized information on any test/exam or any other academic assignment;
- interference – any interruption of the academic process that prevents others from the proper engagement in learning or teaching; and
- fraud – any act or instance of willful deceit or trickery.

Violations of academic integrity will be dealt with by imposing appropriate sanctions. Sanctions for acts of academic dishonesty could include the resubmission of an assignment, failure of the test/exam, failure in the course, probation, suspension from the College, and even expulsion from the College.

Student Code of Conduct: All students are expected to conduct themselves as responsible and considerate adults who respect the rights of others. Disruptive behavior will not be tolerated. All students are also expected to attend and be on time for all class meetings. No cell phones or similar electronic devices are permitted in class. Please refer to the Essex County College student handbook, Lifeline, for more specific information about the College’s Code of Conduct and attendance requirements.

Additional required course materials: USB flash memory and a note book for taking notes.

<table>
<thead>
<tr>
<th>Week (3 meetings @ 80 minutes)</th>
<th>Topic/Chapter</th>
</tr>
</thead>
</table>
| 1                             | Basic Structure of Computers (Chapter 1 of text)  
|                               | Functional Units  
|                               | Operational Concepts  
|                               | Bus Structures  
|                               | Software  
|                               | Performance  
|                               | Multiprocessors and Multicomputers |
| 2                             | Binary Numbers and Codes (Sections 2.1 & 2.2 of text)  
|                               | 1.2 Number Systems  
|                               | 1.3 Arithmetic Operations  
|                               | 1.4 Decimal Codes  
|                               | 1.5 Alphanumeric Codes |
| 3 – 4                         | Digital Circuits (Appendices A1 – A5 of text)  
|                               | 2.1 Binary Logic and Gates  
|                               | 2.2 Boolean Algebra  
|                               | 2.3 Map Simplification  
|                               | 2.5 Map Manipulation  
|                               | 2.6 Nand and Nor Gates  
|                               | 2.7 Exclusive-Or Gate |
| 5                             | Combinational Systems (Appendices 7, 8, 9 & 10 of text)  
|                               | 3.1 Arithmetic Circuits  
|                               | 3.2 Decoders  
|                               | 3.3 Encoders  
|                               | 3.4 Multiplexers |
| 6                             | Sequential Logic (Appendices A6 & 13)  
|                               | 5.1 Latches  
|                               | 5.2 Flip-Flops  
|                               | 5.3 Design with D Flip-Flops  
<p>|                               | 5.4 Design with JK Flip-Flops |</p>
<table>
<thead>
<tr>
<th>Week</th>
<th>Topic/Chapter</th>
</tr>
</thead>
</table>
| 7 – 8                | **Midterm Exam**  
Computer Instructions and Addressing Modes (Section 2.3 and Chapters 4 & 5 of text)  
4.1 Addressing Modes  
4.2 Stack Organization  
4.3 Data Transfer Instructions  
4.4 Data Manipulation Instructions  
4.5 Floating-Point Operations  
4.6 Program Control Instructions  
4.7 Program Interrupt |
| 9 – 10               | **Registers and Counters**  
6.1 Registers  
6.2 Shift Registers  
6.3 Shift Register with Parallel Load  
6.4 Ripple Counter  
6.5 Synchronous Binary Counter |
| 11 – 12              | **Memory (Chapter 5 of text)**  
7.1 RAM  
7.2 Addressing and Coincident Selection  
7.3 ROM |
| 13                   | **Register Transfers and Datapaths (Section 2.4 of text)**  
8.1 Datapath and Operations  
8.2 Register Transfer Operations  
8.3 Microoperations  
8.4 ALU, the Shifter, and Datapath Representation  
8.5 The Control Word |
| 14                   | **Sequencing and Control**  
9.1 The Control Unit  
9.2 Microprogrammed Control, using Binary Multiplier as an Example  
9.3 A Simple Computer Architecture  
9.4 Multiple-cycle Microprogrammed Control |
| 15                   | **Review**  
**Final Exam** |

**NOTE:** 9 or more projects are assigned on an ongoing basis throughout the semester to correspond to the topics being discussed in class. Roughly one project is due each week (with the exception of exam weeks).
CSC 221 Suggested Projects

NOTE: The project topics given below are suggestions of what might be assigned to help students acquire basic concepts of the architecture of the digital computer. Other projects can be imagined and might serve the instructor and students better depending on the class. Projects are distributed through the course to correspond to the topics being discussed in class.

Project 1: Analyze an advertisement description of a computer to become acquainted with the hardware components, how they are described, and how they work together.

Project 2: Write a short program to convert decimal numbers to binary and the reverse and explore the capacity of finite machines for storing data.

Project 3: Demonstrate the use of Karnaugh maps to express Boolean algebraic expressions in standard normal form (sum of product form).

Project 4: Design an 8-bit adder using full adder circuits.

Project 5: Describe ISO 32-bit floating point standard representation for real numbers then use a spreadsheet program to demonstrate the precision of these numbers on a binary computer.

Project 6: Create a diagram showing how RAM is accessed using decoders arranged in row-column addressing.

Project 7: (Library assignment) Explore resources that describe computer topics such as sequential and combination circuits, assembly language, micro operations and micro operations notation. This project will require the student to visit a library and record the bibliographical data of books that cover one or more of these topics and find reference books that might be helpful in beginning to research one or two of these topics.

Project 8: Write a short program in a high-level language to demonstrate various addressing modes using pointer and reference variables.

Project 9: Write a short description and provide a diagram of the process used in the ALU to multiply floating point numbers using shifters and Boolean algebra.