Catalog Data:
CS/ECE 333: Computer Architecture. 3 Credits.
Topics include the organization and architecture of computer systems hardware; instruction set architectures; addressing modes; register transfer notation; processor design and computer arithmetic; memory systems; hardware implementations of virtual memory; and input/output control and devices.

Prerequisites:
ENGR 208: Digital Logic Design (Now CS/EE 230) – Grade of C- or higher
CS 201: Software Development Efforts – Grade of C- or higher

Textbook:
John L. Hennessy and David A. Patterson, Computer Organization and Design: The Hardware/Software Interface, 2nd edition

References:
Vincent P. Heuring and Harry F. Jordan, Computer Systems Design and Architecture,
Addison-Wesley, Menlo Park, California, 1997.
William Stallings, Computer Organization and Architecture: Designing for Performance,

Instructor:
Christopher W. Milner, Ph.D., Department of Computer Science

Goals:
This course is intended to provide students a solid background in the organization and architecture of computer systems hardware. The functions and interactions of the various components that create a computer will be examined in detail. Modern processors will be used to illustrate many of the concepts presented in this class. The students will gain a sufficient understanding of computers to pursue more advanced computer design and computer-based system design courses. Practical experience will be obtained by assembly language programming on a modern processor.

Prerequisites by topic:
1. Boolean algebra
2. Combinational logic and sequential circuit design
3. Software development methods

Topics:
1. History of computers
2. Review of digital logic design
3. Basic structure of computer hardware and software
4. Computer instruction sets
5. Assembly language programming
6. Addressing modes
7. Example processors
8. Processor data path
9. Hardwired control units
10. Microprogrammed control units
11. Pipelining
12. Computer arithmetic
13. Memory components
14. Memory systems
15. Input and output units
16. Interrupts
17. Direct memory access (DMA)
18. Peripheral devices
19. Computer communications
20. Computer networks

Computer Usage:
Students are required to write assembly language programs for execution on a modern processor. Students gain experience designing and implementing the computer data path using the SMOK tool. Students use the University of Washington Cebollita toolkit to write C and ASM systems programs for the processor.

Laboratory Projects:
We will use Cebollita to learn assembly language programming and SMOK to perform data path design work.

ABET category content
Engineering Science: 2.5 credits or 83.3%
Engineering Design: 0.5 credit or 16.6%

Grading:
Mid Term – 20%
Lab Exam – 30%
Homework, Labs – 20%
Class Participation – 10%
Final Exam (Cumulative) – 20%
Course Objectives:
1. Understand the classification of computers (accumulator machines, stack machines and general purpose register machines), instruction types (arithmetic, data movement and control), instruction formats (0,1,2, etc.-address machines) and addressing modes.
2. Understand formal notations for describing processors.
3. Evaluate the design and performance trade-offs between Complex Instruction Set Computers (CISC) and Reduced Instruction Set Computers (RISC).
4. Develop an understanding of processor design (the design process, data path implementation, control unit implementation, 1- 2- and 3-bus processor designs and machine exceptions, pipelining and instruction-level parallelism)
5. Develop an understanding of computer arithmetic and arithmetic units
6. Develop an understanding of the memory hierarchy, cache memory and virtual memory.
7. Gain practical experience in programming with assembly language.
8. Understand program vulnerabilities arising from computer architecture decisions.

Tentative schedule:

<table>
<thead>
<tr>
<th>Week</th>
<th>Dates</th>
<th>Topics</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8/27-8/29</td>
<td>Ch.1 Intro, Ch.2, Performance</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>9/3-9/5</td>
<td>Performance, operations, operands (addressing), encoding</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>9/10-9/12</td>
<td>Decisions, procedure calls, more addressing</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>9/17-9/19</td>
<td>Systems programs: compiler, asm, linker, loader, arrays v. pointers</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>9/24-9/26</td>
<td>80x86, 0-, 1-, 2-, etc. address machines,</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10/1-10/3</td>
<td>Review, Midterm</td>
<td>Midterm</td>
</tr>
<tr>
<td>7</td>
<td>10/8-10/10</td>
<td>Signed and unsigned numbers, addition, subtraction, logical, ALU,</td>
<td>Drop date 10/7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>multiplication, division, floating-point</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>10/15-10/17</td>
<td>Finish up maths, Single-cycle datapath, single-cycle control</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>10/22-10/24</td>
<td>Multi-cycle datapath and control, microprogramming</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10/29-10/31</td>
<td>Exceptions, pipelining datapath, control</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>11/5-11/7</td>
<td>Pipelining: data hazards, exceptions</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>11/12-11/14</td>
<td>Cache, performance, Virtual memory</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>11/19-11/21</td>
<td>I/O</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>11/26-11/28</td>
<td>Thanksgiving Recess</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>11/3-11/5</td>
<td>Microcontrollers</td>
<td>2 Hour Lab Exam this week</td>
</tr>
</tbody>
</table>