Course Syllabus	**EECS 3100 – Microsystems Design**
Credits & Contact Hours	4 credit hours & 150 minutes lecture and 150 minutes lab contact hours per week.
Coordinator	Dr. Gursel Serpen
Textbook	Barry B. Brey, "The Intel Microprocessors", 8th Ed., Prentice Hall. 2009.
Course Information	Microprocessor systems design: basic computer system, cpu, embedded assembly programming, memory and peripheral interfaces, I/O techniques, interrupt structures, DMA, memory management, Hierarchies, and caches. Prerequisites: EECS 2110 and EECS 3400 Required course
Specific Goals-Student Learning Objectives (SLOs)	The students will be able to 1. Describe the meaning of an embedded system, the reasons for the importance of embedded systems, and how computer engineering uses or benefits from embedded systems. 2. Understand how assembly language programs convert into executable code through assembler, linker, locator and loader for an embedded system environment. 3. Write assembly code for an embedded system to function as system kernel, to perform setup, initialization, and built-in system testing. 4. Understand role of modern computer engineering hardware and software tools in system development and how to use these tools to support the design methodology. 5. Develop an understanding of the differences between a microprocessor and a microcontroller in regards to the hardware/software interface for communication with external devices. 6. Design a memory subsystem with both read-only memory and random-access memory for a microprocessor, develop read-only memory compliant random-access memory testing program in relevant assembly language, and program read-only memory with the memory testing program. 7. Design an interface for a programmable input/output device such as universal synchronous-asynchronous receiver-

transmitter and develop the device driver code in assembly/machine language.

8. Design an interface for a programmable interrupt controller and develop the code for device drivers in assembly/machine language.
9. Prototype a minimal system complete with microprocessor, both read-only and random-access memory, read-only memory resident random-access memory testing program, and system startup code developed in assembly or machine language.
10. Function effectively on a multidisciplinary team (as potentially composed of EE and CSE majors), with effectiveness being determined by instructor observation, peer ratings, and self-assessment.

Topics

1. Introduction to Embedded Systems
2. Intel ×86 Architecture: Microprocessor internal architecture, and segmented memory organization
3. Intel ×86 Hardware Specifications: Pins and signal descriptions and timing, bus buffering and latching, read & write timing, and minimum/maximum mode operation
4. Intel ×86 Memory Interface: Memory devices, ROM/SRAM/DRAM device, address decoding, ROM/RAM decoding subsystem design, and odd/even memory banks
5. Intel ×86 Assembly Language Programming: addressing modes, decision making, looping and control structures, procedures and parameter passing, and stack operation
6. Intel ×86 Input/Output (I/O) Interface: Assembly I/O instructions, isolated I/O, memory-mapped I/O, I/O port address decoding, data transfer between ×88/×188/×86/×186 and I/O ports, byte-wide and word-wide I/O ports, buffered input port, and latched output port
7. Intel ×86 Interrupt Structure: Interrupt processing, interrupt service routines, interrupt controller devices, interrupt interface expansion, and daisy-chained interrupt