ECCE 316 Microprocessor Systems
(3 Lectures, 3 Laboratories – 4 Credits)

Prerequisites: ECCE 210

Course Catalog Description:
Introduction to current microprocessor, microcontroller and microcomputer systems: basic components, memory map, organization and processor architecture. Hardware and software models of microprocessor and microcontroller systems. Processor instructions and assembly language programming. Exception handling: interrupts, traps and exception processing. Memory decoding, input/output interfaces and programming peripheral devices. Laboratory experiments provide hands-on experience in the use of cross-assemblers, C-programming, simulators and actual microprocessor/microcontroller hardware.

Textbooks:
1. Software and Hardware Engineering: Assembly and C programming for the Freescale HCS12 Microcontroller by Fredrick M. Cady, Oxford University Press, 2nd Intl Edition, 2009, ISBN 9780195391138.
2. KUSTAR ECCE 316 Laboratory manual

Reference Materials:
1. Han-Way (Han-Way Huang) Huang, The HCS12 / 9S12: An Introduction to Software and Hardware Interfacing, Delmar Cengage Learning, 2nd edition, 2009
2. Jonathan W. Valvano, Embedded Microcomputer Systems: Real Time Interfacing, CL-Engineering, 3rd edition, 2011
3. William Stalling, Computer Organization and Architecture, Prentice Hall, 9th ed, 2012
4. David Patterson and John Hennessy, Computer Organization and Design: The Hardware/Software Interface, Morgan Kaufmann, 4th ed., 2011

Course Topics:
1. Microcomputer Systems: general architecture of a microcomputer system, overall operation and the fetch-execute cycle, comparison of CICS and RISC based systems. (2 lectures)
2. Memory: types of memory devices, memory map of a system, memory space partitioning, full and partial decoding. (2 lectures)
3. Programmer model of a processor: registers types, sizes and functions. (2 lectures)
4. Processor hardware: classification of processor signals and their functions. Read and write cycles and their timing diagrams. (2 lectures)
5. Addressing modes: purpose of addressing modes, types of addressing modes and examples. (3 lectures)
6. Instruction set: instructions format, decoding instructions, instructions types and examples. (5 lectures)
7. Assembly programming: design process and structured programming in assembly language, and position-independent code. Complete assembly language programs with varied degrees of complexity. (4 lectures)
8. Subroutines: basic subroutine concepts, methods of parameters passing and programming examples. (3 lectures)
9. Programming in C and assembly: relationship between programming in assembly and the C language. (3 lectures)
10. Input/Output Circuits and operation: I/O interfaces, conditional and unconditional I/O, programming examples. (2 lectures)
11. Interfacing with Real world: sensors, actuators, signal conditioning. (3 lectures)
12. Data Converters: Analog to digital converters, Digital to analog converters. (3 lectures)
13. Peripheral devices: architecture, operation and interfacing of peripheral devices such as the SCI, ADC and Timers. (3 lectures)
14. Exception handling and interrupts (2 lecture)

15. Programmable Logic Controllers (PLCs) basics, different types of interfacing modules, ladder programming and industrial control using PLC's (6 lectures)

Course Structure:
Lectures: will either be 2 x 75 minutes per week. Laboratory sessions: 3 hours a week in the laboratory

Computer Resources:
Students use personal computers, microprocessor development systems and assembly programming tools.

Laboratory Experiments:
- Microprocessor/microcontroller cross-assembler and development system
- Addressing modes
- Arithmetic and data movement operations
- Program flow control & Subroutines
- Programming peripheral devices (with Interrupts)
- Assembly and C language programming
- PLC basic usage
- PLC programming and interfacing
- Microprocessor/microcontroller Interfacing design project (2-3 students design, build, test and write report)

Laboratory Resources:
Each workstation has PC with a microprocessor/microcontroller development system and application board, and circuit prototyping board and PLC unit with I/O modules.

Assessment:
All course learning outcomes are assessed using the following assessment tools.

Coursework (2 Tests)	20%
Semester Examination (Midterm Exam)	25%
Laboratory experiment, projects and report	20%
Final Examination	35%

Contribution to Engineering Students Outcomes:

a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**
-	L	M	-	H	-	M			M	H

H – High M – Medium L – Low

Contribution to Engineering Students Learning Outcomes:

Course Learning Outcome	**Student Outcomes**
1. Describe the general architecture and operation of a microcomputer system.	j
2. Differentiate the function and operation of the main building blocks of a microprocessor/microcontroller.	c,j
3. Examine the processor's memory space to blocks with various functions.	c,j,k
4. Construct a programs for microprocessor/microcontroller using assembly and C programming languages.	b,c,k
5. Investigate sensors, actuators and data converters	e,k
6. Examine Programmable Logic Controllers, learn how to interface and program them	c,k
7. Construct interfacing programs for microprocessors/microcontrollers to I/O devices.	b,c,k
8. Create laboratory reports, interpret experimental results and critically appraise them.	a,b,g

**This syllabus is augmented by a Syllabus Supplement for Students.*