

NOVA COLLEGE-WIDE COURSE CONTENT SUMMARY

EGR 270 – FUNDAMENTALS OF COMPUTER ENGINEERING (4 CR.)

Course Description

Covers digital system analysis, design, and implementation. Includes digital logic, Boolean algebra, combinational and sequential circuits, hierarchical design, and introduction to computer organization and assembly language. Features in laboratory work the use of discrete logic, programmable logic devices, and hardware description language to design, simulate, implement, validate, and document digital circuits
Lecture 3 hours. Laboratory 3 hours. Total 6 hours per week. 4 credits

General Course Purpose

EGR 270 introduces second year Electrical and Computer Engineering students to digital logic design including basic register transfer level, and hardware description language. This course covers techniques for designing, validating, implementing, and debugging digital circuits which can be integrated together to make larger, more complex digital systems.

Course Prerequisites/Corequisites

Prerequisites: EGR 121 and either EGR 125 or CSC 221 (or CSC 200).

Course Objectives

Upon completing the course, the student will be able to:

Number systems

- Represent and manipulate information in digital systems
- Apply these concepts to perform computer arithmetic

Combinational Logic

- Express, simplify, and minimize Boolean functions, using truth tables, Boolean algebra, canonical forms, and Karnaugh maps, including don't cares
- Utilize technology mapping techniques, such as NAND only
- Use decoders and multiplexers to implement a combinational logic function or system
- Identify design tradeoffs
- Analyze and design combinational logic circuits

Sequential Logic

- Explain and analyze the behavior of basic sequential logic elements
- Represent sequential systems using finite state machines, including Moore and Mealy
- Produce state diagrams, state tables, excitation tables and state equations, including one hot encoding
- Use timing diagrams to explain basic sequential circuit timing issues
- Analyze and design synchronous sequential logic circuits

Datapath Components

- Analyze and design circuits using Datapath components: registers, adders, shifters, comparators, and counters
- Analyze and design multi-function registers and counters
- Explain arithmetic logic units (ALUs), registers, and register operations
- Explain elementary register transfer level designs
- Describe the organization and operation of a basic digital computer

Hardware Description Language (HDL)

- Describe the difference between HDL and software

- Develop structural and dataflow models of logic circuits
- Verify designs using a testbench
- Use behavioral models of logic circuits
- Map HDL models to logic circuits on field programmable gate arrays (FPGAs)

Introduction to Assembly Language

- Explain the hierarchical relationship between transistors, digital logic, and microprocessors
- Explain the operation of machine language vis a vis computer architecture
- Analyze and revise basic assembly language computer programs using a 5-10 instruction set

Lab implementation

- Implement and debug combinational and sequential logic circuits using standard integrated chips
- Implement and debug combinational and sequential logic circuits using HDL and FPGAs
- Implement and debug basic systems using introductory level assembly language computer programs using a limited instruction set

Major Topics to be Included

Number systems

Combinational Logic

Sequential Logic

Datapath Components

Hardware Description Language (HDL)

Introduction to Assembly Language

Lab implementation

Design process