

Virginia Western Community College
EGR 270
Fundamentals of Computer Engineering

Prerequisites

EGR 121 and either EGR 125 or CSC 221

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

Semester Credits: 4 Lecture Hours: 3 Lab/Clinical/Internship Hours: 3

Required Materials**Textbooks:**

Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog, 6th edition by Mano and Ciletti, ©2021, eBook format ISBN 9780137501984 or printed format ISBN 9780134549897, Publisher: Pearson. For example: <https://www.pearson.com/en-us/subject-catalog/p/digital-design-with-an-introduction-to-the-verilog-hdl-vhdl-and-systemverilog/P200000003241>

Other Required Materials:

none

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.

See: <https://courses.vccs.edu/courses/EGR270-FundamentalsofComputerEngineering/detail>

Course Outcomes

At the completion of this course, the student should be able to utilize the following:

- 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

Topical Description

- Number systems
- Combinational Logic
- Sequential Logic
- Datapath Components
- Hardware Description Language (HDL)
- Introduction to Assembly Language
- Lab implementation
- Design process

Notes to Instructors

- All instructors teaching this course in any given semester will use the same textbooks.
- The content of this course will be updated every few years in collaboration with engineering faculty from across the VCCS.

[ADA Statement](#) (PDF)

[Title IX Statement](#) (PDF)