Course Name: Digital Circuits and Logic

Course Number: CST* E145

Credits: 4

Catalog description: A study of the elements of digital logic design, digital circuits, and the fundamentals of a modern digital system. Topics include binary number systems and data representation, Boolean algebra, analysis and design of combinational and sequential logic circuits, basic computer components, processor instruction set and assembly language. Logic design exercises and simulations are used to provide practical experience.

Prerequisite: MAT* E137 or higher. CSC *E105 or CSC* E223 or permission of the instructor

Corequisite or Parallel:

General Education Competencies Satisfied:

HCC General Education Requirement Designated Competency Attribute Code(s):

None

Additional CSCU General Education Requirements for CSCU Transfer Degree Programs:

None

Embedded Competency(ies):

None

Discipline-Specific Attribute Code(s):

☒ COMP **Computer Science**

Course objectives:

General Education Goals and Outcomes:

None

Course Specific Objectives:

1. Learn number systems

2. Learn Boolean logic, theorems, logic gates and Karnough Maps

3. Design combinational logic such as Multiplexer, Decoder, Adder and Subtractor

4. Analyze combinational circuits

5. Analyze and design sequential circuits

6. Describe the function of microcomputer's components

7. Learn characteristics of RAM, DRAM, ROM, EPROM and EEPROM

8. Perform experiments using breadboard and / or emulators such as LogicWorks or Logisim or Micromedia Logic. NOTE: Micromedia Logic is an open source product. It allows users to begin with simple gates and work their way up to larger components, such as memory and ALU, which are included as devices in the software.

9. Learn processor instructions including data transfer, arithmetic instructions, loops and conditional jumps

10. Assemble, link and run an assembler program

11. Learn different addressing modes

Course Content:

1 Digital Systems

2 Machine Representation of Data

1.2 Binary Numbers

1.3 Number-Base Conversions

1.4 Octal and Hexadecimal Numbers

1.5 Complements of Numbers

1.6 Signed Binary Numbers

1.7 Floating Point Representation

1.8 Binary Codes

1.9 Binary Storage and Registers

1.10 Binary Logic

3 Basic Computer Architecture

3.1 CPU architecture

3.2 Control unit

3.3 Virtual memory

3.4 Cache memory

3.5 Peripheral devices

3.6 Parallel systems (multiple cores)

4 Assembly-Level Programming

4.1 Instruction sets and types (data manipulation, control, I/O)

4.2 Assembly/machine language programming. Instruction formats, addressing modes, subroutine call and return mechanisms

4.3 I/O and interrupts

5 Memory system organization and architecture

6 Interfacing and communication

7 Boolean Algebra and Logic Gates

7.1 Basic Definitions

7.2 Axiomatic Definition of Boolean Algebra

7.3 Basic Theorems and Properties of Boolean Algebra

7.4 Boolean Functions

7.5 Canonical and Standard Forms

7.6 Other Logic Operations

7.7 Digital Logic Gates

7.8 Integrated Circuits

8 Gate-Level Minimization

8.1 The Map Method

8.2 Four-Variable K-Map

8.3 Product-of-Sums Simplification

8.4 Don’t-Care Conditions

8.5 NAND and NOR Implementation

8.6 Other Two-Level Implementations

8.7 Exclusive-OR Function

8.8 Hardware Description Language

9 Combinational Logic

9.1 Combinational Circuits

9.2 Analysis Procedure

9.3 Design Procedure

9.4 Binary Adder-Subtractor

9.5 Decimal Adder

9.6 Binary Multiplier

9.7 Magnitude Comparator

9.8 Decoders

9.9 Encoders

9.10 Multiplexers

9.11 HDL Models of Combinational Circuits

10 Synchronous Sequential Logic

10.1 Sequential Circuits

10.2 Storage Elements: Latches

10.3 Storage Elements: Flip-Flops

10.4 Analysis of Clocked Sequential Circuits

10.5 Synthesizable HDL Models of Sequential Circuits

10.6 State Reduction and Assignment

10.7 Design Procedure

Laboratory Projects

1. Microcomputer Components
2. Familiarization with equipment and safety
3. Combinational Logic
4. Decoder and Multiplexer
5. Half Adder and Full Adder
6. Flip Flops
7. Shift Register
8. Binary Counter
9. Arithmetic Logic Unit
10. Control unit

11. Memory I/O

12. Final project: build a rudimentary virtual CPU that can execute exactly one type of instruction

Date Course Created:

Date of Last Revision: 01/22/2018