EET-1241: DIGITAL FUNDAMENTALS

Cuyahoga Community College

Viewing: EET-1241 : Digital Fundamentals

Board of Trustees:
December 2022

Academic Term:
Fall 2023

Subject Code
EET - Electrical/Electronic Engineer

Course Number:
1241

Title:
Digital Fundamentals

Catalog Description:
Introductory course to digital circuits. Logic and arithmetic operations are studied, designed and tested in a laboratory environment using discrete integrated circuit gates and programmable logic devices (PLD). Base 2 (binary) and base 16 (hexadecimal) number systems are used in conjunction with Boolean algebra and other theorems. Foundation for continued study of microprocessors/ microcontrollers.

Credit Hour(s):
3

Lecture Hour(s):
2

Lab Hour(s):
2

Requisites

Prerequisite and Corequisite
EET-1161 Direct Current Circuits, or concurrent enrollment; or departmental approval.

Outcomes

Course Outcome(s):
Demonstrate by using mathematics an understanding of base 2 and base 16 numbering systems and convert numbers from/to base 2, base 10 and base 16.

Objective(s):

a. Calculate and demonstrate base 2 addition.
b. Calculate and demonstrate base 2 subtraction.
c. Convert base 2 to base 16 and base 16 to base 2.
d. Convert base 2 to base 10 and base 10 to base 2.
e. Demonstrate binary subtraction by using the two's complement for the subtrahend and adding.
f. Calculate the product of two base 2 numbers.

Course Outcome(s):
Demonstrate logical operations using mathematics. Display the result in a lab environment using discrete logic gates and/or display the result using programmable logic.

Objective(s):

a. Demonstrate on paper and in the laboratory the operation of an AND gate, its truth table and logic symbol.
b. Demonstrate on paper and in the laboratory the operation of an OR gate, its truth table and logic symbol.
c. Demonstrate on paper and in the laboratory the operation of an INVERTER, its truth table and logic symbol.
d. Demonstrate on paper and in the laboratory the operation of a NAND gate, its truth table and logic symbol.
e. Demonstrate on paper and in the laboratory the operation of a NOR gate, its truth table and logic symbol.
f. Demonstrate on paper and in the laboratory the operation of an XOR gate, its truth table and logic symbol.
g. Demonstrate on paper and in the laboratory the operation of an XNOR gate, its truth table and logic symbol.

Course Outcome(s):

Simplify gate complexity by using rules of simplification and DeMorgan's theorem.

Objective(s):

a. Use the rules of Boolean algebra to simplify gates (reduce gate count) and demonstrate in a lab environment.
b. Use DeMorgan's theorem to simplify gates (reduce gate count) and demonstrate in a lab environment.
c. Use Karnaugh Maps to simplify circuits and prove that the simplified circuit is equivalent to the original circuit.
d. Use Karnaugh Maps to explain Sum of Products (SOP) and Product of Sum (POS) equations.

Course Outcome(s):

Demonstrate converting AND, OR and INVERT gates to universal gates and prove equivalence in a lab environment.

Objective(s):

a. Convert AND, OR and INVERT gates to NAND form. Design and test the circuit.
b. Convert AND, OR and INVERT gates to NOR form. Design and test the circuit.

Course Outcome(s):

Design and explain combinational logic circuits.

Objective(s):

a. Design solutions to logic problems such as driving a seven segment display using combinational logic gates.
b. Use gates and demonstrate design functionality in lab.
c. Use Field Programmable Gate Arrays (FPGAs) in simulation mode and program a FPGA to demonstrate functionality in a lab environment.

Course Outcome(s):

Explain binary adder integrated circuit operation using at least 4 bits.

Objective(s):

a. Build in a lab environment a binary adder and demonstrate the effect of carry input on the output.
b. Build in a lab environment a binary subtractor using a binary adder where the subtrahend is represented in two's complement form (uses carry in).

Course Outcome(s):

Explain the encoding or multiplexing operation and identify the part number(s) of integrated circuits that perform that function.

Objective(s):

a. Design and build an encoder/multiplexer circuit and demonstrate its output in lab.
b. Design the program for a FPGA that will perform the encoding/multiplexing function and show successful simulation and device function.

Course Outcome(s):

Explain the decoding or demultiplexing operation and identify the part number(s) of integrated circuits that perform that function.

Objective(s):

a. Design and build a decoder/demultiplexer circuit and demonstrate its output in lab.
b. Design the program for a FPGA that will perform the decoding/demultiplexing function and show successful simulation and device function.

Course Outcome(s):

Design and demonstrate various types of counters that are based upon flip-flops.

Objective(s):

a. Design and build a decade counter using flip-flops and gates or design a program using a FPGA that functions as a decade counter. Simulate and program the part to verify functionality.
b. Design and build a gray counter using flip-flops and gates or design a program using a FPGA that functions as a decade counter. Simulate and program the part to verify functionality. Explain the unique feature of a gray counter.

Course Outcome(s):

Design and build shift registers and explain how they can be used as binary multiply by 2 or binary divide by 2.

Objective(s):

a. Design and demonstrate a 4 bit shift right register using flip-flops and gates or by programming a FPGA.
b. Design and demonstrate a 4 bit shift left register using flip-flops and gates or by programming a FPGA.

Course Outcome(s):

Explain why a flip-flop is a single bit memory device and demonstrate its operation.

Objective(s):

a. Demonstrate the operation of set and clear on various types of flip-flops (clocking input is inactive).
b. Demonstrate an understanding of the transition controls inputs J and K (not acronyms) of a J-K flip-flop in a lab environment. If the physical integrated circuit is negative edge triggered, explain the output state if it were positive edge triggered and vice versa. This can also be done by using a FPGA.
c. Explain and demonstrate a data (D) latch and how its operation is different from a D flip-flop. This can be done by using a FPGA.

Course Outcome(s):

Explain the difference between static and dynamic memory devices including read-only memory and read-write memory.

Objective(s):

a. Construct and demonstrate a 4 bit read-write memory device using a static random access memory. Verify its operation by demonstrating that data read from an address is the same as data written to the address.
b. Use a flash drive and explain what type of memory it contains. That is, read-only or read-write, static or dynamic, volatile on power removal, or nonvolatile.
c. Explain write durability and load leveling in flash based jump memory.

Methods of Evaluation:

a. Tests
b. Quizzes
c. Laboratory Reports

d. Homework
e. Projects

Course Content Outline:

.

a. Number systems
    i. Binary (integer and fractional)
    ii. Hexadecimal (integer and fractional)
    iii. 8421 BCD code
    iv. Gray code
b. Boolean algebra
    i. Basic "AND" and "OR" operations
    ii. NOT gate
    iii. Boolean postulates
    iv. Boolean theorems
    v. Logic diagrams using standard logic gates
c. Logic gate equivalents
    i. AND, OR and INVERT gates
    ii. Troubleshooting AND, OR and INVERT gates
    iii. NAND and NOR gates
    iv. Troubleshooting NAND and NOR gates
    v. Positive logic and negative logic
    vi. Universal property of NAND and NOR gates
d. Boolean algebra applications
    i. Exclusive "OR" gate
    ii. Half adder and full adder
    iii. Simplifications using Karnaugh map
    iv. "Don't care" conditions
    v. BCD seven-segment decoder
e. Code converters
    i. Binary decoder
    ii. Seven-segment decoder
    iii. Encoders
    iv. Multiplexers (data selectors)
    v. Demultiplexers
    vi. Parity
f. Binary arithmetic
    i. Addition (binary, octal, and hexadecimal)
    ii. Subtraction (one's and two's complement)
    iii. Multiplication
    iv. Division
g. Switching circuits and data storage
    i. D or toggle flip-flop
    ii. J-K flip-flop
    iii. Monostable multivibrator
    iv. Astable multivibrator
    v. Latches
h. Counters
    i. Ripple or asynchronous counters
    ii. Binary up/down counter
    iii. Synchronous counters
    iv. Design of sequential counter
    v. Analysis of sequential counter
i. Registers
    i. Shift right register
    ii. Shift left register
    iii. Serial in-parallel out

iv. Ring counter
v. Bi-directional shift register

j. Integrated Circuit Technologies
i. Basic operational characteristics and parameters
ii. CMOS circuits
iii. TTL (bipolar) circuits
iv. Practical considerations in the use of TTL
v. Comparison of CMOS and TTL performance
vi. Emitter-Coupled Logic (ECL) circuits
vii. PMOS, NMOS, and E^2CMOS

Resources

Floyd, Thomas L. *Digital Fundamentals*. 11th ed. Prentice Hall, 2021.

Brown, Stephen and Zvonko Vranesic. *Fundamentals of Digital Logic with VHDL Design*. 4th ed. McGraw-Hill, 2023.

Chu, Pong P. *FPGA Prototyping by VHDL Examples: Xilinx Microblaze MCS SoC*. 2nd ed. Wiley, 2017.

Wakerly, John F. *Digital Design: Principles and Practices*. 5th ed. Pearson, 2018.

Patterson, David and John Hennessy. *Computer Organization and Design*. 6th ed. Elsevier, 2020.

Instructional Services

OAN Number:
Transfer Assurance Guide OET002 and Career Technical Assurance Guide CTEET002

Top of page
Key: 1625