COURSE NUMBER: DP1110

COURSE TITLE: Digital Systems I (Logic)

COURSE DESCRIPTION:

This course introduces learners to the field of digital electronics. They will be taught design and diagnosis techniques applicable to digital electronics.

PREREQUISITES: ET1101 - Electrotechnology

CO-REQUISITES: None

CREDIT VALUE: Four (4)

COURSE HOURS PER WEEK: Three (3)

LAB HOURS PER WEEK: Two (2)

SUGGESTED TEXT:

One of:

Kleitz, W. (2012). *Digital electronics, A practical approach with VHDL*. Prentice Hall. ISBN 13: 978-0-13-254303-3, 10: 0-13-254303-6

Floyd, T. (2009). *Digital fundamentals* (10th ed.). Upper Saddle River, NJ: Prentice Hall. ISBN-13: 9780132359238

LEARNING RESOURCES: None

MAJOR TOPICS:

1.0 Introduction to Digital Circuits
2.0 Combinational Logic
3.0 Programmable Logic Arrays
4.0 Sequential Logic

LEARNING OBJECTIVES:

Upon completing this course, a proficient learner should be able to:

1.0 Introduction to Digital Circuits

1.1 Definition of Digital

1.1.1 Digital vs. Analog

1.1.1.1 Define both the meaning of analog and digital

1.1.1.2 Distinguish between continuously variable and discrete quantities

1.1.2 Examples

1.1.2.1 Give at least five examples of analog quantities

1.1.2.2 Give at least five examples of digital quantities

1.1.3 Advantages/Disadvantages

1.1.3.1 Explain how digital representation is robust in the presence of noise

1.1.3.2 Explain why digital information can be reproduced without degradation

1.1.3.3 Show that information must be lost when converting from analog to digital

1.1.3.4 Demonstrate how round-off errors in digital calculations accumulate

1.1.4 Binary Systems On-Off, High-Low, True-False, 1-0

1.1.4.1 Explain that a binary symbol set only requires two distinguishable states

1.1.4.2 Give four different methods to represent discrete information

1.1.4.3 Show how grouping binary bits can increase the range of representable values

1.2 Number Systems Used in Digital Systems

1.2.1 Decimal Review

1.2.1.1 State the 'rules' involved in addition, subtraction, multiplication and representation of decimal quantities

1.2.2 Counting in Various Bases

1.2.2.1 Generate an ascending or descending count sequence over any range in any base

1.2.3 Arithmetic Operations in Various Bases

1.2.3.1 Generalize the rules of decimal arithmetic to apply them to other bases

1.2.4 Conversion of Bases using Place Value

1.2.4.1 Convert a number of any base to decimal by forming a sum of the powers and digits

1.2.4.2 Convert a decimal number to any base by determining the maximum value each power should have

1.2.5 Conversion of Bases using Successive Multiplication or Division
- 1.2.5.1 Convert from decimal to any base by successive division by the base and the accumulation of remainders
- 1.2.5.2 Convert from any base to decimal by the addition of the digits and the successive multiplication by the base

1.2.6 Base Conversions where one Base is a Power of the other
- 1.2.6.1 Convert from base nm to n without the intermediate step of decimal representation
- 1.2.6.2 Convert from base n to nm without the intermediate step of decimal representation

2.0 Combinational Logic

2.1 Introduction to Boolean Algebra

2.1.1 Use of AND, OR, NOT Operators in English
- 2.1.1.1 Form English sentences which involve AND, OR and NOT relationships

2.1.2 Use of AND, OR, NOT as Boolean Operators
- 2.1.2.1 Express AND, OR and NOT operations in Boolean statements

2.1.3 Truth Table for Logic Representations
- 2.1.3.1 Generate the truth table for a NOT gate or an AND, OR gate of any number of inputs

2.1.4 Positive Logic Assignments - (low, 0, false) - (high, 1, true)
- 2.1.4.1 Define the terms active-high and positive logic

2.1.5 Equivalent Operations using Switches
- 2.1.5.1 Draw AND and OR circuits using equivalent switch circuitry

2.1.6 Circuit Diagram Symbols for AND, OR, NOT
- 2.1.6.1 Draw the symbols for a NOT gate or an AND, OR gate of any number of inputs

2.1.7 NAND, NOR Operations
- 2.1.7.1 Show how a NAND or NOR function can be generated with AND, OR and NOT gates
- 2.1.7.2 Generate the truth table of a NAND or NOR gate of any number of inputs

2.1.8 Circuit Diagram Symbols for NAND, NOR
- 2.1.8.1 Draw the symbols for a NAND or NOR gate of any number of inputs

2.1.9 XOR, XNOR Operations
- 2.1.9.1 Show how an XOR or XNOR function can be generated with AND, OR and NOT gates
- 2.1.9.2 Generate the truth table of an XOR or XNOR gate of two inputs

2.1.10 Circuit Diagram Symbols for XOR, XNOR
- 2.1.10.1 Draw the symbols for an XOR or XNOR gate of two inputs

2.1.11 Boolean Algebra Identities
- 2.1.11.1 Generalize the effect of the NOT function of a NOT
- 2.1.11.2 Generalize the output of a two input AND function when a variable is applied with '1', '0', itself and or with its complement
- 2.1.11.3 Generalize the output of a two input OR function when a variable is applied with '1', '0', itself and or with its complement

2.1.12 Boolean Algebra Properties
- 2.1.12.1 Demonstrate the commutative, distributive, associative and absorption properties of an AND or OR system

2.1.13 D'Morgan's Theorem
- 2.1.13.1 Use D'Morgan's theorem to replace a NAND gate with OR and NOT gates
- 2.1.13.2 Use D'Morgan's theorem to replace a NOR gate with AND and NOT gates

2.1.14 Deriving XOR Expressions in Boolean Algebra
- 2.1.14.1 State the XOR function in terms of AND, OR and NOT functions

2.1.15 Simplification of Boolean Algebra Expressions
- 2.1.15.1 Use Boolean identities, properties and D'Morgan's theorem to simplify a Boolean expression

2.1.16 Application of Boolean Algebra to Circuit Simplification
- 2.1.16.1 Use the simplification of a Boolean expression to reduce the number of gates required in a system

2.2 Combinational Logic Circuit Design

2.2.1 The 7400 Series of TTL Gates
- 2.2.1.1 Expand the 'TTL' abbreviation
- 2.2.1.2 Identify and explain the 74xx, 74LSxx, 74ALSxx and 74Fxx family of digital gates

2.2.2 Analysis of Circuit Diagrams - Generator Truth Table
- 2.2.2.1 Generate a truth-table from a circuit diagram containing multiple inputs and outputs

2.2.3 Deriving SOP Expression from Truth Table
2.2.3.1 Write the Boolean expression of a truth table output as a Sum of Products

2.2.4 Deriving POS Expression from Truth Table
2.2.4.1 Write the Boolean expression of a truth table output as a Product of Sums

2.2.5 Design of Efficient Circuits using Boolean Algebra
2.2.5.1 Simplify a SOP or POS Boolean expression and draw the resulting circuit diagram

2.2.6 Use of NANDs and NORs as Universal Gates
2.2.6.1 Explain why NAND and NOR gates are preferable to AND or OR gates
2.2.6.2 Draw a minimized SOP or POS function with NAND or NOR gates

2.2.7 Application to Summer and Subtractor Circuits
2.2.7.1 Generate the Truth-Table, POS representation and simplest circuit diagram of a Full or Half adder
2.2.7.2 Generate the Truth-Table, POS representation and simplest circuit diagram of a Full or Half subtractor
2.2.7.3 Reduce the component count of an Adder or Subtractor using XOR gates

2.2.8 Debugging/Troubleshooting Techniques
2.2.8.1 Logically trace a digital circuit to find wiring errors
2.2.8.2 Test a logic design to prove whether it performs as expected
2.2.8.3 Correct a flawed design by modifying the circuit diagram

2.3 Karnaugh Mapping
2.3.1 Completing 3 and 4 Variable K-Maps from Truth Table
2.3.1.1 Generate a 3 or 4 Variable K-Map from Truth Table
2.3.1.2 Generate a truth table from a 3 or 4 variable K-Map

2.3.2 Locating Groups of 1’s for SOP form of Expression
2.3.2.1 Find the rectangular groups of 1's in the K-Map and specify their most efficient Product term
2.3.2.2 Choose the most efficient groups of 1's to produce a circuit with the minimum number of gates

2.3.3 Locating Groups of 0’s for POS form of Expression
2.3.3.1 Find the rectangular groups of 0's in the K-Map and specify their most efficient Sum term

2.3.3.2 Choose the most efficient groups of 0's to produce a circuit with the minimum number of gates

2.3.4 Locating XOR Patterns in K-Maps
2.3.4.1 Recognize the "cross-hatch" XOR patterns in a K-Map and specify them as the most efficient XOR terms

2.3.5 Use of "Don't Care" Cells in K-Maps
2.3.5.1 Place "Don't Care" terms in a K-Map and use them to advantage when generating minimized SOP and POS expressions

2.3.6 K-Maps with more than 4 Inputs
2.3.6.1 Deal with K-Maps which have 5 or 6 inputs

2.4 Advanced Combinatorial Devices and Circuits
2.4.1 Multiplexers
2.4.1.1 Recognize, specify and design multiplexer circuits
2.4.1.2 Choose appropriate encoder circuits from a manufacturer's specification

2.4.2 DeMultiplexers
2.4.2.1 Recognize, specify and design demultiplexer circuits
2.4.2.2 Choose appropriate decoder circuits from a manufacturer's specification

3.0 Programmable Logic Arrays

3.1 Generic Combinatorial Circuits using SOP or POS Forms
3.1.1 Explain the significance of PAL fuses
3.1.2 Show the basic structure of a combinational PAL
3.1.3 Show how a logical function can be "burned" into a PAL

3.2 PAL Characteristics
3.2.1 Compare the speed and power capabilities of the PAL family with that of TTL

3.3 PAL Programming
3.3.1 Explain how a PAL can be programmed using a PAL Assembler package and a PAL programming device

3.4 PAL Applications
3.4.1 Give examples of how a PAL implementation reduces the package count of a circuit

4.0 Sequential Logic

4.1 Introduction to Sequential Logic
- 4.1.1 Latches using NAND and NOR Gates
  - 4.1.1.1 Draw the circuit of a latch implemented with NAND or NOR gates
  - 4.1.1.2 Show how a latch can be set or reset
  - 4.1.1.3 Explain why a latch retains its value
- 4.1.2 Timing Diagrams
  - 4.1.2.1 Show how the operation of sequential logic can be shown with a timing diagram
- 4.1.3 Gated Latches
  - 4.1.3.1 Show how additional gates enhance the performance of a basic latch
- 4.1.4 Applications of Latches
  - 4.1.4.1 Give examples of latch circuits and explain their operation
- 4.1.5 D-Type Flip-Flop Function Table
  - 4.1.5.1 Draw the function table of a D Flip-Flop and explain how it works
- 4.1.6 Edge vs. Level Activity
  - 4.1.6.1 Explain the difference between edge-triggered and level-sensitive latch operations
- 4.1.7 JK-Type Flip-Flop Function Table
  - 4.1.7.1 Draw the function table of a JK Flip-flop and explain how it works
- 4.1.8 Applications of Flip-Flops
  - 4.1.8.1 Give examples of flip-flop circuits and explain how they operate
- 4.1.9 Analysis of Sequential Logic Devices
  - 4.1.9.1 Demonstrate the operation of flip flops in a lab environment

4.2 Counters
- 4.2.1 Asynchronous Up-Counters
  - 4.2.1.1 Show how J/K or D Flip-Flops can be connected as up-counters
  - 4.2.1.2 Extend an up-counter system to any number of bits
  - 4.2.1.3 Generate a carry-out or overflow signal
- 4.2.2 Asynchronous Down-Counters
  - 4.2.2.1 Show how J/K or D Flip-Flops can be connected as down-counters
  - 4.2.2.2 Extend a down-counter system to any number of bits

4.2.2.3 Generate a borrow-out or underflow signal

4.2.3 Synchronous Up-Counters
- 4.2.3.1 Distinguish between a synchronous and asynchronous counter
- 4.2.3.2 Give examples where a synchronous counter must be used

4.2.4 Synchronous Down-Counters
- 4.2.4.1 Give examples where a synchronous counter must be used

4.2.5 Up-Down Counters
- 4.2.5.1 Employ an up-down counter in a circuit
- 4.2.5.2 Explain the significance of up and down counting on the carry/borrow signal

4.2.6 Modulo-X Counters
- 4.2.6.1 Form a counter of any modulus from an existing binary counter

4.2.7 MSI Counters (7490, 7493)
- 4.2.7.1 Show how a 7490 or 7493 counter must be wired for counting, and how it can be reset

4.2.8 Applications of Counters
- 4.2.8.1 List a number of applications which take advantage of the synchronous/asynchronous, up/down, and variable moduli features of counters

4.3 Sequential State Design

4.3.1 State Transition Diagrams
- 4.3.1.1 Generate and interpret state transition diagrams

4.3.2 Generation of a Function Table from the Transition State Diagram

4.3.3 D-Type Flip-Flop Excitation Table
- 4.3.3.1 Recall the Excitation table for a D Flip-Flop

4.3.4 JK-Type Flip-Flop Excitation Table
- 4.3.4.1 Recall the Excitation table for a JK Flip-Flop

4.3.5 Circuit Design
- 4.3.5.1 Develop the circuitry which executes a given function table using D or J/K Flip-Flops

4.3.6 Applications of Sequential State Machines
- 4.3.6.1 Describe the MicroProcessor in terms of a sequential state machine
- 4.3.6.2 List a number of other applications of sequential state machines

EVALUATION:

Laboratories	15 %
Assignments	5 %
Tests	30 %
Final Exam	50 %

Written communication skills and technical presentation will be included in the evaluation of all laboratory reports, projects and other written work.

DATE DEVELOPED: **DATE REVIEWED:** March 2011

REVISION NUMBER: 2 **DATE REVISED:** March 2012

Note to instructor: Check PIRS to ensure this outline is the most current version.