Syllabus for Computer Logic and Circuit Design: PHYS306/COSC3300

Dr. Jordan Hanson - Whittier College Dept. of Physics and Astronomy

January 22, 2020

Abstract

The fundamental concepts of digital logic and digital circuit design are presented, both theoretically and in laboratory exercises. In the beginning, core concepts like binary number systems, binary mathematics, and floating point representation are presented. The concepts of Boolean logic and logic gates, truth tables, combinatorial logic and Karnaugh maps follow, all with corresponding laboratory demonstrations or exercises. These core concepts prove useful in a wide range of engineering, business, and scientific situations. Laboratory exercises are designed to provide hands-on experience with the physical implementation of digital logic gates and programmable logic. Following an understanding of core modules, more complex digital elements are introduced. Examples include flip-flops (memory elements), counters, registers and shift registers. The course concludes with examples of digital systems useful for science and engineering, including binary adders (ALUs) and analog-to-digital converters (ADCs), digital-to-analog converters (DACs), digital signal processing (DSP) of audio, and HDMI video. Student-designed projects are created and presented near the end of the course that integrate programmable logic and peripheral devices that enrich the student experience.

***Pre-requisites**: PHYS180, COSC120. Students may receive permission to take these courses in the same semester as PHYS306/COSC330.*
***Course credits, Liberal Arts Categorization**: 3 Credits, None*
***Regular course hours**: Tuesdays and Thursdays from 13:30 - 14:50 in SLC 104. Laboratory exercises: SLC 102.*
***Instructor contact information**: jhanson2@whittier.edu, tel. 562.907.5130*
***Office hours**: Monday 8:00-12:00 in SLC 212*
***Attendance/Absence**: Students needing to reschedule midterms and exams should notify the professor a reasonable time beforehand. Further attendance issues are left to the discretion of the instructor.*
***Late work policy**: Late work is generally not accepted, but is left to the discretion of the instructor.*
***Text**: Digital Fundamentals, 11th ed., by Thomas L. Floyd (Pearson). This is a widely adopted text with a variety of online features, online chapters, code examples, and excellent homework sets*[1].
***Grading**: The assignment weighting for this course: two midterms worth 15% each, laboratory exercises worth 15%, homework exercises worth 25%, a final project design and presentation worth 15%, and a final exam worth 15%. The final exam will be held on May 8th, from 15:30 - 17:30.*
***Grade Settings**: $< 60\% = F$, $> 60\%, \leq 70\% = D$, $> 70\%, \leq 80\% = C$, $> 80\%, \leq 90\% = B$, $< 90\%, \leq 100\% = A$. Pluses and minuses: 0-3% minus, 3%-6% straight, 6%-10% plus (e.g. 79% = C+, 91% = A-)*
***Homework Sets**: Due weekly on Thursdays. Homework sets usually include textbook problems, subitted computer code, and other digital designs/plans.*
***ADA Statement on Disability Services**: The Americans with Disabilities Act (ADA) is a federal anti-discrimination statute that provides comprehensive civil rights protection for persons with disabilities. Among other things, this legislation requires that all students with disabilities be guaranteed a learning environment that provides for reasonable accommodation of their disabilities. If you believe you have a disability requiring an accommodation, please contact Disability Services: disabilityservices@whittier.edu, tel. 562.907.4825.*
***Academic Honesty Policy**: `http://www.whittier.edu/academics/academichonesty`*
***Course Objectives**:*

- To master the binary and hexadecimal number systems, fluency with various binary codes and checksums.
- To master Boolean logic and logic gates, Boolean algebra and logical simplification through Karnaugh maps.
- To practice the application of Boolean logic to scientific, engineering, and business scenarios.
- To train in the art of transforming a diagram into a working prototype.
- To train in the art of troubleshooting a digital design and remedy bugs and errors.
- To develop confidence in building digital prototypes that accomplish design goals.
- To learn how to control programmable logic firmware with computer code.

[1] As a general rule, Prof. Hanson does not require students to purchase expensive texts. However, it is generally recommended to have this book as a reference if you are planning on a a degree in engineering or science. Please make arrangements with Prof. Hanson if you need assistance acquiring the book.

Course Outline:

1. Week 0 - **Reading: Chapter 1 of Digital Fundamentals.**
   (a) First meeting - Thursday, January 30th, 2020.
      i. Course introduction, syllabus distribution
      ii. *Laboratory tour and introductory laboratory exercises*
         A. The PYNQ-Z1 System on a Chip (SoC)
         B. Jupyter Notebooks
         C. Breadboards, digital voltmeters, DC power supplies
         D. Oscilloscopes and signal generation
2. Week 1 - **Reading: Chapters 1-2 of Digital Fundamentals.**
   (a) First meeting - Tuesday, February 4th, 2020. *Reading: Chapter 1 of DF.*
      i. Introductory concepts in digital design
      ii. Logic functions: basic, combinatorial and sequential
      iii. Programmable logic and fixed IC logic
      iv. Test and measurement instruments
      v. *Laboratory exercise, part I*
         A. Logic signal creation and measurement with oscilloscope
         B. Capturing digital signals from PYNQ-Z1: the PL layer and logictools library
   (b) Second meeting - Thursday, February 6th, 2020. *Reading: Chapters 2-1 through 2-7*
      i. Decimal and binary numbers, conversions and arithmetic
      ii. Signed numbers and arithmetic with signed numbers
      iii. *Laboratory exercise, part II*
         A. Representing a binary number with LEDs
         B. PYNQ-Z1 and logictools: boolean generator, simple version.
3. Week 2 - **Reading: Chapters 2-3 of Digital Fundamentals.**
   (a) First meeting - Tuesday, February 11th, 2020. *Reading: Chapters 2-8, 2-10 through 2-12*
      i. Hexadecimal numbers
      ii. BCD and other numerical codes
      iii. Error codes: parity bits and CRCs
      iv. *Laboratory exercise, part I*
         A. Decimal, binary, and hexadecimal conversion
         B. Python programming, random number generation, and matplotlib
   (b) Second meeting - Thursday, February 13th, 2020. *Reading: Chapters 3-1 through 3-6*
      i. Logic gates: NOT, AND, OR, NAND, NOR, XOR, XNOR
      ii. *Laboratory exercise, part II*
         A. PYNQ-Z1 and logictools finite state machine (FSM) generator
         B. Creating a gray code counter
4. Week 3 - **Reading: Chapters 3-4 of Digital Fundamentals.**
   (a) First meeting - Tuesday, February 18th, 2020. *Reading: Chapters 3-8 and 3-9, 4-1 and 4-2*
      i. Programmable logic with gates
      ii. Troubleshooting
      iii. Boolean operations and expressions, algebra
      iv. Abstract applications and examples in business, engineering
      v. *Laboratory exercise, part 1*
         A. Complex boolean functions, simulation, programmable logic, verification, part I
   (b) Second meeting - Thursday, February 20th, 2020. *Reading: Chapters 4-3 through 4-10*
      i. DeMorgan's Theorems; further abstract examples in business and engineering
      ii. Boolean analysis of logic circuits, simplification
      iii. Truth tables, Karnaugh maps and SOP/POS minimization
      iv. *Laboratory exercise, part II*
         A. Complex boolean functions, simulation, programmable logic, verification, part II

5. Week 4 - **Reading: Chapter 5 of Digital Fundamentals**
   (a) First meeting - Tuesday, February 25th, 2020. *Reading: Chapters 5-1 through 5-3*
      i. Combinatorial systems of logic gates
      ii. Universal NAND and NOR gates
      iii. *Laboratory exercise, part I*
         A. Trace analyzer and pattern generator
         B. Use PYNQ-Z1 and pattern generator to multi-pin digital patterns
         C. View multi-pin digital patterns with trace analyzer
         D. Plot the trace of combinatorial logic, part I
   (b) Second meeting - Thursday, February 27th, 2020. *Reading: Chapters 5-4 through 5-7*
      i. *Laboratory exercise, part II*
         A. Plot the trace of combinatorial logic, part II
         B. System simplification and plot trace to show equivalency.
6. Week 5 - **Reading: Chapter 6 of Digital Fundamentals**
   (a) First meeting - Tuesday, March 3rd, 2020. *Reading: Chapters 6-1 through 6-6*
      i. Adders
      ii. Comparators
      iii. Encoders
      iv. *Laboratory exercise, part I*
         A. Testing fixed IC adders with scope, LEDs
         B. Testing fixed IC comparators with scope, LEDs
   (b) Second meeting - Thursday, March 5th, 2020. *Reading: Chapters 6-7 through 6-11*
      i. Multiplexers and Demultiplexers
      ii. Troubleshooting
      iii. *Laboratory exercise, part II*
         A. Adders in programmable logic with gates
         B. Comparators in programmable logic with gates
7. Week 6 - **Reading: Chapters 7 and 8**
   (a) First meeting - Tuesday, March 10th, 2020. *Reading: Chapters 7-1 through 7-4, 7-5*
      i. S-R latches, flip-flops, and applications
      ii. Timers and pulse generation
      iii. *Laboratory exercise, part I*
         A. Demonstration of fast pulse generation
         B. Measurements of RF pulses, digital signal processing with NumPy/Octave
   (b) Second meeting - Thursday, March 12th, 2020. *Reading: Chapters 7-5, and 8-1 through 8-4*
      i. One-shots (timers)
      ii. Shift registers
      iii. *Laboratory exercise, part II*
         A. Formation of final project teams
         B. Brainstorming for project ideas
8. Week 7 - **Spring Break: March 16th through 20th, 2020.**
9. Week 8 - **Reading: Chapters 8 and 9**
   (a) First meeting - Tuesday, March 24th, 2020. *Reading: Chapters 8-5, 9-1 through 9-5*
      i. Shift register applications
      ii. Finite state machines (FSMs)
      iii. Asynchronous and synchronous counters
      iv. *Laboratory exercise, part I*
         A. Testing fixed IC counters with scope
         B. Simple FSM and counter example with PYNQ-Z!
   (b) Second meeting - Thursday, March 26th, 2020. *Reading: 9-6 through 9-8*

i. Cascaded counters and counter decoding
ii. Applications of counters
iii. *Laboratory exercise, part II*
A. Pattern generator counting and combination with boolean generator via pynq logictools control
B. FSM counter generation and combination with boolean generator via pynq logictools control

10. Week 9 - **Reading: Chapter 12**

(a) First meeting - Tuesday, March 31st, 2020. **Reading: Chapters 12-1 through 12-3, with supplemental DSP reading**
i. Analog-to-Digital Conversion (ADC)
ii. Digital-to-Analog Conversion (DAC)
iii. *Laboratory exercise, part I*
A. PYNQ-Z1 PMOD (peripheral modules) ports
B. ADC PMOD - read in a voltage and plot it

(b) Second meeting - Thursday, April 2nd, 2020. **Reading: Chapters 12-4 through 12-5, with supplemental DSP reading**
i. Digital Signal Processing (DSP) - an introduction to sampling and digitization
ii. The DSP processer
iii. *Laboratory exercise, part II*
A. PYNQ-Z1 PMOD (peripheral modules) ports
B. DAC PMOD - produce a voltage and plot it on oscilloscope

11. Week 10 - **Reading:**

(a) First meeting - Tuesday, April 7th, 2020.
(b) Second meeting - Thursday, April 9th, 2020.

12. Week 11

(a) First meeting - Tuesday, April 14th, 2020.
(b) Second meeting - Thursday, April 16th, 2020.

13. Week 12

(a) First meeting - Tuesday, April 21st, 2020.
(b) Second meeting - Thursday, April 23rd, 2020.

14. Week 13

(a) First meeting - Tuesday, April 28th, 2020.
(b) Second meeting - Thursday, April 30th, 2020.

15. Week 14

(a) First meeting - Tuesday, May 5th, 2020.

16. Final Exam: Friday, May 8th, 2020.