Physics 632, Digital Circuits

**Course Description**

(3 S.H.) A lecture-laboratory course in digital electronics and systems with emphasis on experimental measurement in physics.  Topics include Boolean algebra, logic circuits, counters, registers, arithmetic-logic units, sequential circuits, sequence detectors, and finite-state machines.  Students will implement designs on hardware including field programmable gate array (FPGA) systems using the Verilog design and test language

Prerequisites: Introductory Electromagnetism (WSU Physics 222 or 202 or equivalent course) and Calculus (WSU Math 213 or equivalent course). Instructor permission is also acceptable. The course is online, with one or two optional in-person class meetings.

**Physics Subfield: Experimental Physics**

**Learning Outcomes**

A successful student in this course should be able to:

1. Take an appropriate novel physical problem and abstract it to a logical system/expression, using with Boolean variables.
2. Use Boolean Algebra and Karnaugh Maps to simplify logical expressions and create basic digital components (XOR, adder, latch, etc).
3. Express an algorithm or control structure as a finite-state machine.
4. Build and troubleshoot digital circuits (or verilog code).
5. Maintain a laboratory notebook documenting the project requirements, hardware design and/or flowchart, and finished schematic/code.

These goals will be assessed via weekly homework and occasional exams.

Additionally, students will:

1. Use digital circuit components to build an original data acquisition and control device for classroom or laboratory work

This goal will be assessed via a final project.

Two-level outline

1. Number systems
   1. How do we measure things? Voltage and Logic levels
   2. Base-2, bits, and Analog to Digital (ADC) signal conversion
   3. Negative number systems
2. Physical Implementation
   1. Diode circuit logic
   2. Transistor circuit logic
   3. TTL (7400 chips)
   4. Field Programmable Gate Arrays and Verilog
3. Logic Gates
   1. NOT/AND/OR
   2. NAND/NOR
   3. XOR/XNOR
   4. Neg-AND, neg-OR
   5. Verilog implementations
4. Logic Expression Systems
   1. Truth tables
   2. Boolean algebra
   3. SOP/POS forms
   4. DeMorgan’s Theorum
   5. Karnaugh Maps
   6. Functional vs Behavioral Verilog modeling
5. Logical Machine Building Blocks
   1. Multiplexors
   2. Decoders
   3. Adders and subtraction
   4. Verilog Implementations
6. Memory
   1. Latches and Flip-Flops
   2. Pre/re sets
   3. Synch and asynch behavior, measuring on interrupt
   4. Counters, clocks
   5. Shift registers
   6. Verilog wire vs register
7. Finite State Machines
   1. States and state transition rules
   2. Clock synchronization and edge detection
   3. Verilog Always blocks
8. CPU architecture
   1. Registers, data and instruction “stacks”
   2. Arithmetic/Logic Units (ALU)
   3. Serial/parallel operation
   4. Connections to assembly language programming