2017 Pacific Southwest Section
Meeting: Tempe, Arizona Apr 20
Paper ID #20704

An Inverted Approach to Introductory Digital Design

Prof. bryan james mealy, Cal Poly San Luis Obispo

Bryan Mealy is an associate professor at Cal Poly State University in San Luis Obispo, California. Despite everything, he still finds the ultimate fulfillment in teaching.

An Inverted Approach to Introductory Digital Design

Bryan J. Mealy
California Polytechnic State University, San Luis Obispo

Abstract

Digital design courses generally start by introducing basic circuit elements such as logic gates. These gates form the foundation of all standard digital circuit modules such as multiplexors, decoders, registers, etc. The notion of digital design involves assembling and connecting a set of these basic circuit modules in such a way that it solves a given problem. This low-level first approach to digital design can be tedious, which can subsequently burden students as they are developing the skills to design more complex circuits. This paper describes a qualitative approach to digital design where we introduce circuits by first describing them at a high level and in the context of actual design problems. This inverted approach delays the presentation of low-level details until after students understand the basic functioning of the standard digital modules in actual designs. We provide the associated low-level details after students have complete many high-level digital designs.

Our novel approach to basic digital design utilizes two rarely considered qualities of digital circuits. First, we can model complex digital circuits as a set of various standard digital modules interfaced with each other and controlled by another digital circuit. We support this model by using a new digital design paradigm that classifies digital modules according to their basic purpose in a circuit. Second, we can describe the basic functions of standard digital modules without requiring knowledge of basic digital logic principles such as logic gates and Boolean algebra. Our approach has the benefit of enabling students to design relatively complex circuits at the beginning of the course. Additionally, we remove some of the traditional, but less important digital design topics, which gives students more time to complete complex designs throughout the course. This new approach also underscores our current emphasis on modular-based digital design techniques.

This digital design course includes traditional topics such as binary mathematics, logic gates, standard digital modules, and finite state machine design. The laboratory associated with the course requires students to design and implement circuits on FPGA-based development boards, which subsequently requires that students learn to model digital circuits using a hardware description language (HDL).

Introduction

Advances in digital technology in the last 40 years offer many new approaches to teaching digital design. But based on the approach taken in most digital design-based textbooks,[8,9] the path to teaching digital design remains virtually unchanged: digital design courses start with number theory, then go on to gates and gate-level designs, and later transition to more advanced design topics. This traditional approach to teaching digital design includes covering topics that modern digital design rarely use. A few of these low-value topics include an in-depth study of

Boolean algebra, function reduction using logic theorems and Karnaugh maps, in-depth study of flip-flops, and describing finite state machines (FSMs) that have no real-world applications.

Recent changes to teaching digital design have been generally focused on two areas. First, because field programmable gate arrays (FPGAs) have effectively freed students from the limitation of hand-wiring circuits, many digital design courses utilize FPGAs to implement complex circuits[1]. This approach necessarily requires digital design students to learn an HDL. Another trend in digital design courses is to essentially detach the instructors from the learning process with online courses[2] and various types of online support for courses[3], and generally developing courses that require less contact time with students[7]. While these efforts have positive results, we do not feel they are adequate substitutes for in-person student-instructor interaction.

After many years of teaching digital design using the traditional approach, we conclude that presenting rarely used topics reduces the overall efficacy of the approach. Specifically, spending course time on questionable topics reduces the time students can spend on more meaningful designs. Although many of the topics we refer to represent core digital principles, we rarely see them in modern digital design and feel they persist because they are easy for instructors to teach and later test on. Additionally, we find that students quickly forget knowledge that has no direct real-world applications. Another issue we find that presents obstacles to students is integrating the learning of digital concepts with learning a hardware description language (HDL). We feel this approach overwhelms students with details, which results in students not adequately learning either topic.

This paper describes our new approach to digital design. We focus this new approach on presenting topics in the context of digital designs with real-world applications, removing antiquated topics, separating the learning of an HDL from learning basic digital design, and presenting all circuits using a new digital design paradigm. We first provide an overview of our new approach, which describes our motivations behind moving away from the traditional approach. We base our approach on a new digital design paradigm, which we describe in the next section. The next two sections describe how we divide our approach into two distinct phases: the “Learning Design” and “Implementing Design” phases. We then address some of the practical aspects associated with the overall presentation of the course.

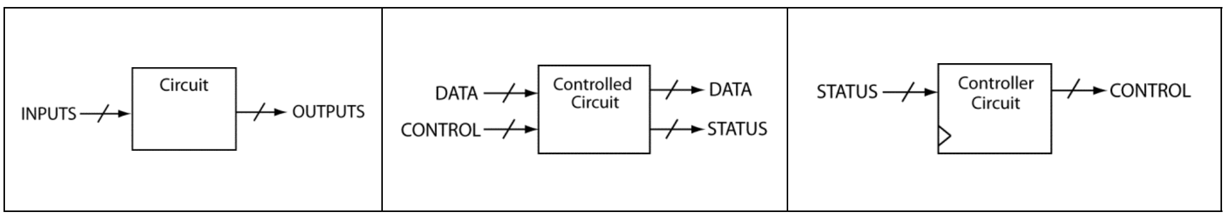
Course Overview

Our new approach to digital design retains most of the topics and learning objectives from the previous incarnation of this course, but presents the material in a unique order. The course topics include number representation and theory, basic logic gates, Boolean algebra, HDL concepts, and general “design concepts”. The associated laboratory requires students to implement their designs on FPGA-based development boards. We expect students to be able to design, model, and implement relatively complex digital circuits by the end of the course. This course is the first of three digital-type courses and is a required course for computer and electrical engineering students.

The underlying goals of our approach are to both simplify and unify the various levels of digital design. We base our simplification efforts on a new digital circuit paradigm, which separates digital modules into either “controlled” circuits or “controlling” circuits. Second, we use this new paradigm to unify the various levels of digital design in our curriculum. In particular, this new model places all levels of digital design into a common context, which is particularly important because our curriculums spread three digital-type courses over two academic years. The gaps between our digital course offerings underscore known problems with knowledge retention by students.

Our New Digital Design Paradigm

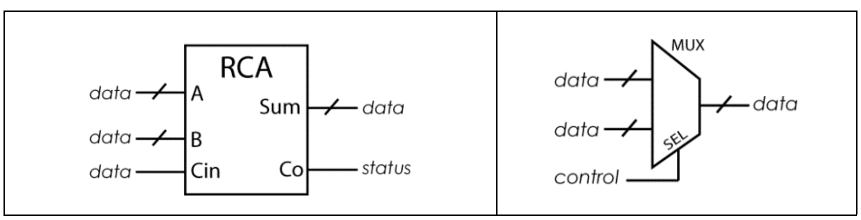
Figure 1(a) shows our previous approach to modeling digital circuits, where we classify all digital circuit signals as either inputs or outputs. Figure 1(b) and Figure 1(c) shows how we now separate digital modules into "controlled circuits" and "controller circuits". Note in Figure 1(b) and Figure 1(c) that the control outputs of the controller circuit provide the control inputs to the controlled circuit. Similarly, the status outputs of the controlled circuit form the status inputs of the controller circuit. Standard digital circuits such as registers and multiplexors (MUXes) are examples of controlled circuits while FSMs and microcontrollers are examples of controller circuits.



(a) (b) (c)

Figure 1: Old digital circuit model (a); models for controlled (b) and controller circuits (c).

Figure 2 shows our new digital design paradigm as it applies to a ripple carry adder (RCA) and a 2:1 MUX. While both circuits have data inputs and data outputs, the RCA has a status output (the carry-out) but has no control input, while the MUX has a control input (SEL) but has no status output.



(a) (b)

Figure 2: Examples of an RCA (a), and a 2:1 MUX (b), modeled as "controlled" circuits.

Our new paradigm allows us to model all digital circuits as controller that controls a set of controllable modules. We can then consider the solution to any digital design problem as a matter of using a controller to properly control the dataflow through a set of controllable modules. Figure 3 shows an example of many circuit modules controlled by a controller circuit. Figure 3 includes three different module shapes showing that controllable modules can either be combinatorial or sequential circuits, as well as more complicated circuits such as off-the-shelf computer peripherals. The model in Figure 3 unifies the presentation of digital circuits in our curriculum's three digital-type courses.

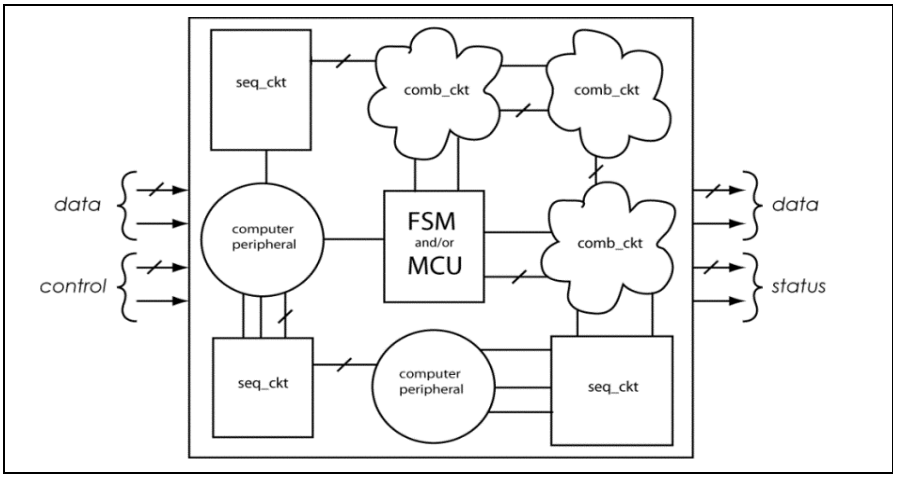


Figure 3: Our unifying digital circuit model.

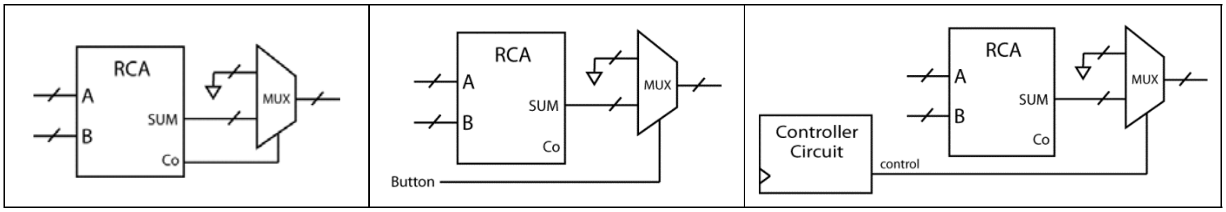
We further base our new digital design approach on the three following notions:

1) Digital circuit design can be done in a structured manner. This structured approach is analogous to structured program design, meaning that we can model any digital circuit using a relatively small set of standard digital modules. Additionally, we use our new digital paradigm to model each of these standard digital modules.

2) We can present the set of standard digital circuits at a high-level by primarily describing the functionality of the circuit in terms of its associated data, control, and status signals. This approach does not require knowledge of low-level implementation details.

3) We can model a digital circuit as a set of digital modules connected in such a way as to solve a problem under control of "something". Additionally, we group this controlling "something" into three categories: a) internal control (internal signals), b) external control (buttons, switches, etc.), or, c) circuit control (using FSM or computer).

Figure 4 shows an arbitrary circuit utilizing the three forms of control. The circuit contains an RCA and a 2:1 MUX; the MUX module has one control input to select which MUX data input appears on the output. Figure 4(a) shows an internally generated signal controlling the MUX (the Co is a status output from the RCA), Figure 4(b) shows an externally generated signal controlling the MUX, and, Figure 4 (c) uses a controller circuit to control the MUX.



(a) **(b)** **(c)**

Figure 4: Three forms of circuit control: (a) internal, (b) external, and (c) circuit controlled.

We divide the course into two distinct phases, which we refer to as the Learning Design and the Implementing Design phases. We dedicate the first half of the course to the presentation of standard digital circuits in the context of a seemingly never-ending stream of design problems. The general theme of the design problems is to use the standard modules in as many different circuit designs as possible. We dedicate the second half of the course to learning an HDL and using it to implement working circuits on an FPGA-based development board. The first half of the course exclusively uses paper-based designs to become familiar with the general operation of standard modules and the design process in general, while the second half of the course requires students to apply their design skills to modeling and implementing actual circuits.

Course Outline: The "Learning Design" Phase

We focus the first half of the course on learning the basic mechanics of digital design. Our approach is to first introduce FSMs (a controller circuit) followed by the basic set of standard digital modules (controlled circuits). The associated design problems use FSMs, internal, and/or external controls to control the associated designs. This FSM-first approach allows us to design meaningful circuits at every course meeting during the first half of the course. Table 1 shows the ordering of topics for the Learning Design portion of the course.

Week #	Topics
1	FSMs (State Diagrams), Counters, Multiplexors
2	Simple Registers, Adders, Comparators
3	Decoders (including basic logic functions), Shift Registers, Parity Generators
4	Numbers, Low-Level Implementations: RCAs, Comparators, Parity Generators
5	Mealy vs. Moore FSMs, Negative Logic, Boolean Algebra, Theorems, Latches

Table 1: Ordering of Topics for the "learning design" portion of the course.

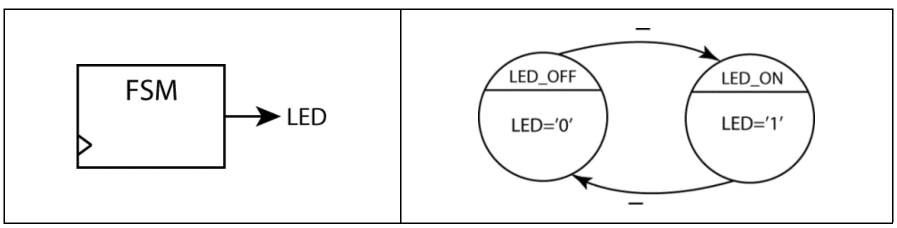
Our approach to presenting new modules starts with black box diagrams that clearly differentiate between data, status, and control signals. We then include the associated timing diagrams to show how the control and data inputs affect the status and data outputs. We complete the presentation with basic design problem examples that specifically utilize the new modules in ways that exercise the basic functions of the modules. The design examples typically utilize many of the previously presented modules in their designs. We then leave students with similar design problems that they complete in the remaining class time and/or as homework. Since students are not implementing circuits until the second half of the course, there is extra in-class time as the course is taught using a studio format that combines lecture and lab. This extra class time allows for more direct interaction between students and instructors.

As Table 1 indicates, we do not present the notion of logic or gates until Week 3, when we describe gates as a type of decoder. We define a decoder as any circuit that we can easily describe in a tabular format. We revisit the notion of gates in Week 4 by describing gate-level implementations of several standard modules previously presented at a block level. We use Week 5 to support basic topics that every digital design course should include. One of these topics includes the notion of Mealy and Moore FSMs, as we have only been using Moore-type FSMs up to this point. Week 5 officially introduces Boolean algebra and some of the more useful associated theorems. We spend a relatively small amount of time discussing function reduction using Boolean algebra, but we no longer include Karnaugh maps in our approach.

In an effort to show our presentation approach in the Learning Design phase, we provide the first two FSM design examples from the first day of the course. We omit many of the preliminary details for the sake of brevity in a potentially boring and verbose conference paper. We include both problems descriptions and their solutions to emphasize the incrementalism of our approach.

Design Problem #1:

Use an FSM to design a circuit that blinks an LED at a frequency that is half the FSM's clock frequency. Provide a black box diagram and a state diagram in your solution.

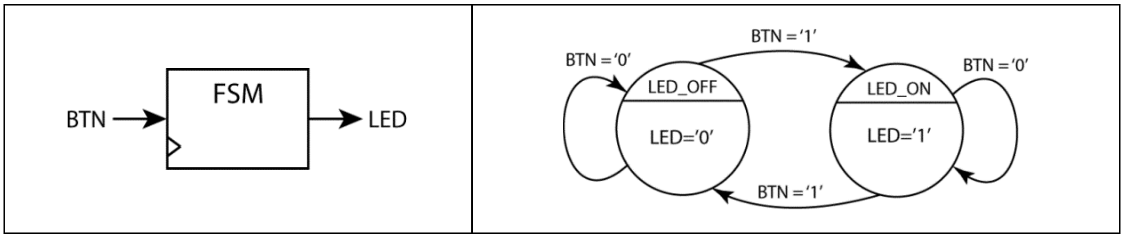


(a) **(b)**

Figure 5: Design Problem #1 solution: (a) the black box diagram, and (b) the state diagram.

Design Problem #2:

Use an FSM to design a circuit that blinks an LED at a frequency that is half the FSM's clock frequency. This circuit has a button input that prevents the LED from turning off if it is *on* or turning on if it is *off* when the button is pressed. Provide a black box diagram and a state diagram in your solution.



(a) **(b)**

Figure 6: Design Problem #2 solution: (a) the black box diagram, and (b) the state diagram.

Course Outline: The "Implementing Designs"

We dedicate the second half of the course to applying the knowledge and skills gained in the first of the course to implementing actual circuits. This approach thus requires that students learn an HDL they can use to model circuits and to become proficient using the development software associated with implementing those circuits on an FPGA-based development board. The second half of the course has students implementing circuits of increasing complexity in order to obtain the skills required to implement more complex circuits toward the end of the course.

After students complete the first half of the course, they will be familiar with basic digital design techniques, which allows them to focus their efforts on learning an HDL the can use to model and synthesize digital circuits. Thus, our students will no longer be simultaneously learning both digital design topics as well as HDL-based modeling techniques[4], which have had undesirable effects in our previous teaching experiences.

The format of the Implementing Designs portion of the course is to design and implement increasingly complex circuits. To support and expedite this process, we direct students toward a modular approach to HDL modeling. In this way, students spend a majority of their time modifying and interfacing HDL-based templates for their modules rather than spending a significant amount of time implementing modules from scratch. Students use the both modules they developed in earlier labs as well as provided template modules in the lab exercises.

Table 2 shows the sequence of laboratory exercises for the Implementing Designs phase. One of the underlying goals of these exercises is to have students develop modules they can use in later lab exercises or in their personal projects. Designing the sequence of experiments in this way supports the notion of modular design and reinforces the notion of module reuse. Students implement FSMs at a high-level using behavioral modeling; topics such as state variable assignment are no longer part of the course.

Week #	Laboratory Exercises
6	Full Adder (SOP & POS), Function Form Implementations, Ripple Carry Adder
7	BCD-to-7-Segment Decoder, BCD Comparator Module, Magnitude Comparator
8	Shift Register, Sequence Detector (FSM), Counter, 7-Seg Display Multiplexor
9-10	Multiplier, Divider, Interfacing with External Hardware Module

Table 2: Ordering of Experiments for the Implementing Designs portion of the course.

Practical Course Implementation Details

Because we are treating this course as a true design course and spend the first half of the course completing paper-based designs, students must constantly interface with the instructors and/or teaching assistants. Because there are rarely absolute solutions to digital design problems, students require direct feedback from mentors to verify the quality of their designs. This may be a detriment to instructors who rely on unqualified graders or "robot graders", but we know the time spent directly helping students is the best investment instructors can make, despite the fact that most academic administrators feel otherwise.

The fact that we have significantly redesigned the course sheds light on the issue of whether there is supporting courseware for the course. We counter this worry with a few comments. Because we are initially presenting only qualitative descriptions of circuits, we can do so with a minimum of course materials. The standard digital circuits associated with the Learning Design phase of the course are adequately describable using outlines of circuit inputs and outputs, which we support with timing diagrams. The actual learning takes place not so much by reading boring textbooks, but by designing circuits that utilize the modules they are studying. Additionally, we do have a more in-depth description of this material based on the fact that we previously authored a digital design textbook[5].

The main modeling tool in the Implementing Designs portion of the course is VHDL, which students use to implement designs based on techniques they learned in the first half of the course. As for learning VHDL, there are many online sources available. Once again, we authored one of these texts[6], which we can modify to support the unique needs of our new course approach. We also utilize some of the material in the existing lab manuals to fit the needs of the course.

Conclusion

Based on the current selection of digital design textbooks, the teaching of introductory digital design has changed little despite the many advances in digital technology. The most significant changes in modern digital design courses have seemingly been the inclusion of FPGAs, which allows students to implement more complex digital designs. This paper described a new approach to digital design courses that primarily uses design problems to reinforce the interaction between standard digital circuit modules. We base this approach to digital design on the notion that we can create complex digital circuits using a relatively small set of standard digital modules. We apply a new digital paradigm to describe digital circuits, which separates digital modules into "controlled circuits" or "controller" circuits. We present the course topics in reverse order compared to typical digital courses by taking advantage of the fact that we can describe standard digital modules using a qualitative approach that does not require knowledge of underlying implementation details. Our overall approach to the course extends the amount of "designing" that occurs in the course by having students work on designs starting on the first day of the course.

Bibliography

1. Areibi, S., 2001. “A First Course in Digital Design Using VHDL and Programmable Logic”. In *Frontiers in Education Conference,* 2001. 31st Annual (Vol. 1, pp. TIC-19). IEEE.

2. Benson, B. and Mealy, B., 2014, October. “Experience with Teaching Digital Design Online”. In *Frontiers in Education Conference*, 2014 IEEE (pp. 1-7). IEEE.

3. Donzellini, G. and Ponta, D., 2007. “A Simulation Environment for E-Learning in Digital Design”. *IEEE Transactions on Industrial Electronics*, 54(6), pp.3078-3085.

4. Marx, J., 2000. “A Proposal for Alternative Methods for Teaching Digital Design”. *Automation in Construction*, *9*(1), pp.19-35.

5. Mealy, B., “Digital McLogic Design”, www.freerangefactory.org

6. Mealy, B., Tappero, F., “FreeRange VHDL”, www.freerangefactory.org

7. Pérez, M.C., Hernández, Á. and Mateos, R., “Promoting Student Autonomy in the Electronic Design Course”, In *Design of Circuits and Integrated Circuits* (DCIS), 2014 Conference on (pp. 1-6). IEEE.

8. Vahid, F., 2010. “Digital Design with RTL Design, Verilog and VHDL”. John Wiley & Sons.

9. Wakerly, J.F., 2013. “Digital Design”, Pearson New International Edition. Pearson Education Limited.