125 YEARS AT THE HEART OF ENGINEERING EDUCATION
ASEE
2018 ASEE ANNUAL CONFERENCE & EXPOSITION
JUNE 24–27, 2018
SALT PALACE CONVENTION CENTER
SALT LAKE CITY, UT

Paper ID #21268

Design and Implementation of a Health-monitoring Design Project in an Introductory Digital Design Course

Dr. Matthew A. Watkins, Lafayette College

Matthew Watkins is an assistant professor of Electrical and Computer Engineering at Lafayette College. He received his Ph.D. and M.S. in Electrical and Computer Engineering from Cornell University and B.S. degrees in Computer Engineering and Electrical Engineering from the University at Buffalo. His research interests include engineering education and the design, use, and management of emerging computer architectures.

Design and Implementation of a Health Monitoring Design Project in an Introductory Digital Design Course

Abstract

Digital Design courses, which cover topics related to combinational and sequential logic, are a common element of most electrical and computer engineering programs. Introductory courses often employ only small "toy" examples that incorporate, at most, a few course topics in an example and fail to demonstrate the utility and power of modern digital systems. Additionally, at our institution, the Digital Design course is the first course that students take within the major, so there is an added interest to structure the course to motivate and retain a broad set of students.

With this in mind, we developed a design project that a) had a meaningful purpose, b) utilizes and integrates many of the topics from the course, and c) is likely to be of interest to a broad array of students. The project is an integrated health monitoring system. The health monitor determines and displays a user's heart rate and measures a user's reaction time (the latter of which can be useful for diagnosing diseases like Parkinson's disease). A complete, digital implementation of the health monitor will incorporate topics including combinational logic, common combinational building blocks, finite state machines, counters, adders, shift registers, and hierarchical design.

On the first day of the course, students are shown a functioning health monitor and as a class brainstorm a list of the capabilities such a system would need to achieve the demonstrated functionality. As new topics are discussed throughout the course, the class returns to the original list of capabilities and students are asked to identify what capabilities recent topics can address and then implement the new components. This immediately puts individual topics into a larger context. After all of the individual capabilities have been addressed, students work together in groups to develop a block diagram which combines the individual components to first realize the reaction timer functionality. Over multiple lab periods they implement the reaction timer design and, leveraging this experience, design and implement the pulse monitor. In the end, students create a meaningful system that incorporates multiple course concepts and that demonstrates the idea of hierarchical design that is common in many areas of engineering. A large majority of students (70-90%) report that the health monitor a) provided a socially relevant application of digital circuits, b) showed meaningful uses for individual course topics, and c) helped connect different course topics.

Introduction

Digital design courses are a common element of most electrical and computer engineering (ECE) programs and are recommended for many computer science programs [1, 2]. These courses cover topics related to combinational and sequential logic and how to design systems using these elements. The example problems used in introductory courses, both in class and in labs, are often small in scope, incorporating only a few different elements. The real power, and challenge, of digital systems comes from combining many simple elements into a larger whole (this is nicely illustrated by the fact that some modern systems have in excess of a billion logic elements on a single chip).

At our institution, the introductory digital design course is the first course students take within the ECE major. This adds a desire to structure the course to engage students, to pique their interest in ECE, and show them real applications of the major. This is also an opportunity to have a project with social relevance that is outside the “common” perceptions of ECE, which can help retain and even attract women and underrepresented groups in ECE.

With these elements in mind, we developed a design project with multiple goals:

- To have a meaningful purpose and realistic application,
- To utilize and integrate many course topics,
- To illustrate how to design a complex system from many simple components,
- To show an application of digital logic outside of what is likely to be perceived as its “common” application, and
- To be of interest to a broad range of students.

The project we developed is an integrated health monitor. The health monitor provides two diagnostic features. It can measure and display both a user’s heart rate and a user’s reaction time. While the usefulness of an integrated health assessment device is likely obvious to most, the broad interest of the idea and the fact that it is not already commonly available can be highlighted by the $10 million Qualcomm Tricorder XPRIZE that deals precisely with developing an integrated health monitoring and diagnostic tool [3]. XPRIZE targets innovative and unsolved challenges that can have a transformative impact. While the prize for this challenge was awarded in April 2017, it does not mean the problem is solved. There is still much work to be done before any of the proposed devices will be ready for widespread use.

Background

The ECE program at our institution includes a two course digital design sequence that is taken in the fall and spring semesters of students’ 2nd year. Semesters are 15 weeks long with 14 weeks of actual instruction. The courses meet three times a week for 50 minutes and have split weekly three hour lab periods. The first digital design course introduces the fundamentals of digital circuits and how to design basic digital systems. The course uses a physical textbook [4] and an interactive electronic textbook [5] to support the instruction. This is the first ECE course taken by students within the major and is the course targeted by the health monitor discussed in this paper. The second course looks at more complex digital systems, with a particular focus on computer organization and the design of microprocessors.

Prior to this new project, the first digital circuits course included nine labs and a multi-week final project. Each lab largely focused on a single element related to material recently discussed in class. In the final project, students implemented a version of the game Pong. In general, the document describing a particular lab or project provided the overall design of the system students had to implement. The students were primarily responsible for implementing the system. While this made the assignments more tractable, it hid what is often a critical piece of designing digital systems. Also, while the culminating Pong game project was fun, it utilized digital logic in a fairly classical domain and had limited broad relevance.

Existing research has identified a number of factors that can help attract and retain underrepresented groups, especially women, within engineering. Women more often choose majors that they perceive to have benefit to others [6]. They are attracted to and benefit from seeing applications that have real-life context, are socially relevant, and involve hands-on or problem-based learning [7, 8].

Design and Implementation

When selecting and developing a new project for the digital design course, there were a number of constraints and goals that influenced our decisions. We wanted a project that

- Was more than just a toy and served a meaningful purpose. The goal was to be able to connect class concepts to something realistic.
- Would incorporate many of the typical topics for a digital design course. The goal was to be able to show how different components could be used in concert to create something bigger and, through this, also walk through the design process of a system, particularly one without a single solution.
- Would be relevant and of interest to a broad range of students. Based on existing research, this suggested something that had a social impact and might not generally be thought of as a traditional engineering project.
- Would be tractable for students in their first ECE course without any expected background in the area, while also not having to provide large amounts of support or starter code. The goal was to have students develop the majority of the system themselves so that they would have a sense of ownership.

In our context, the goal was to have the pieces of the design and implementation process span the first 8-10 weeks of the course. This allowed time in the last third of the course to include a separate design project. The health monitor project is used to illustrate the steps in designing a moderately complex digital systems. The individual elements of the system are all discussed and sometimes designed as a class. The second project provides students the opportunity to take the knowledge and experience developed by creating the health monitor and apply it to another digital design project with less formal structuring. In our current implementation, the second project is a keypad-based PIN access system, although the specific project does not impact the health monitor.

The health monitor provides two pieces of functionality. It measures and displays a user's reaction time and a user's heart rate. Slow reaction time can serve, for example, as a possible indication of Parkinson's disease. The heart rate measurement is supported by an analog pulse sensor that is provided to students. The sensor outputs a digitized version of the pulsatile waveform sensed from a user's finger. Figure 1 shows an example output of the sensor. The sensor is discussed in more detail in the *Implementation Details* section below. As a whole, the system has the following functionality.

- The mode is selected by a slide switch.
- Reaction Timer
  - On the press of a Start Button, the system waits a random amount of time between roughly 1 and 9 seconds, with at least 8 possible different wait times.

    - After the random wait, the system turns on an LED and records the amount of time until the user presses an Enter Button. The LED should be off except when waiting for the user to press Enter.
    - Four 7-segment displays show one of four messages.
        - The reaction time in the format x.xxx (in seconds). The time is displayed after the Enter Button is pressed.
        - A '----' message if the Enter Button is pressed before the random wait has completed.
        - An ' Err' message if the user takes more than 9.999 seconds to press the enter button after the LED turns on.
        - Blank if currently evaluating the user's response time or after reset.
- Heart Rate Monitor
    - Receives a digital pulse signal from a pulse sensor and processes the signal to count each pulse once.
    - Samples the user's pulse over five seconds and maintains the last three samples to calculate the user's heart rate as a moving average.
    - Displays the user's heart rate in beats per minute (BPM) up to a maximum of 255 BPM.

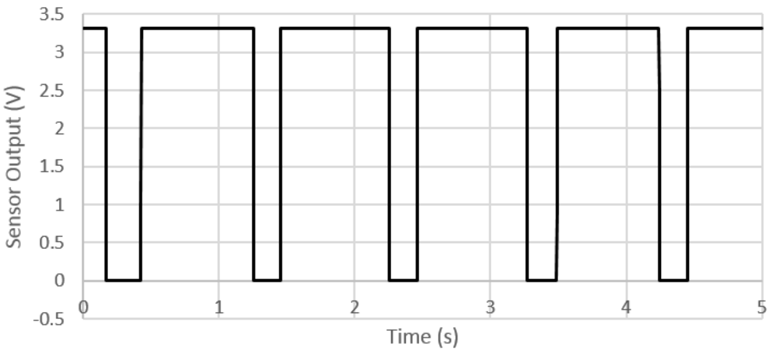


Figure 1. Example output of pulse sensor.

Figure 2 shows the hardware setup used to implement the health monitor and highlights the major input and output elements of the system. The display illustrates a correctly recorded reaction time. Figure 3 shows the other message types that the system can display.

The design and implementation of the health monitor and its subcomponents is incorporated in elements of lecture, lab, and homework throughout the first eight weeks of the course. It culminates with students pairs implementing the health monitor in lab over a two week period.

In-Class

On the first day of class, students are shown a working version of the health monitor to motivate them and show them what they will be working towards over the next two months. After seeing the demonstration, they are asked individually to think about and develop a list of capabilities such a system would need. Then, the class as a whole develops a complete list of necessary capabilities. This is done by getting ideas from individual students and providing prompts for yet unaddressed capabilities. It is important when developing this list that the elements be discrete capabilities needed for the system (not individual pieces of behavior) so that they can be translated into hardware elements in the future. As an example, one capability is to “track the state of the reaction timer” (ex., waiting, error state, timing). This is as opposed to, for instance, “displaying ‘----’ when an error occurs.” The former easily has a direct translation to hardware, while the latter does not. While developing the list, the instructor can choose to word some of these capabilities to include keywords, like “state” or “count,” that can be useful later in the course when connecting them to different topics. The final list includes 13 elements. Additional examples include “convert a binary number to 7-segment display signals” and “increment the heart rate count by one per non-fixed-width pulse input.”

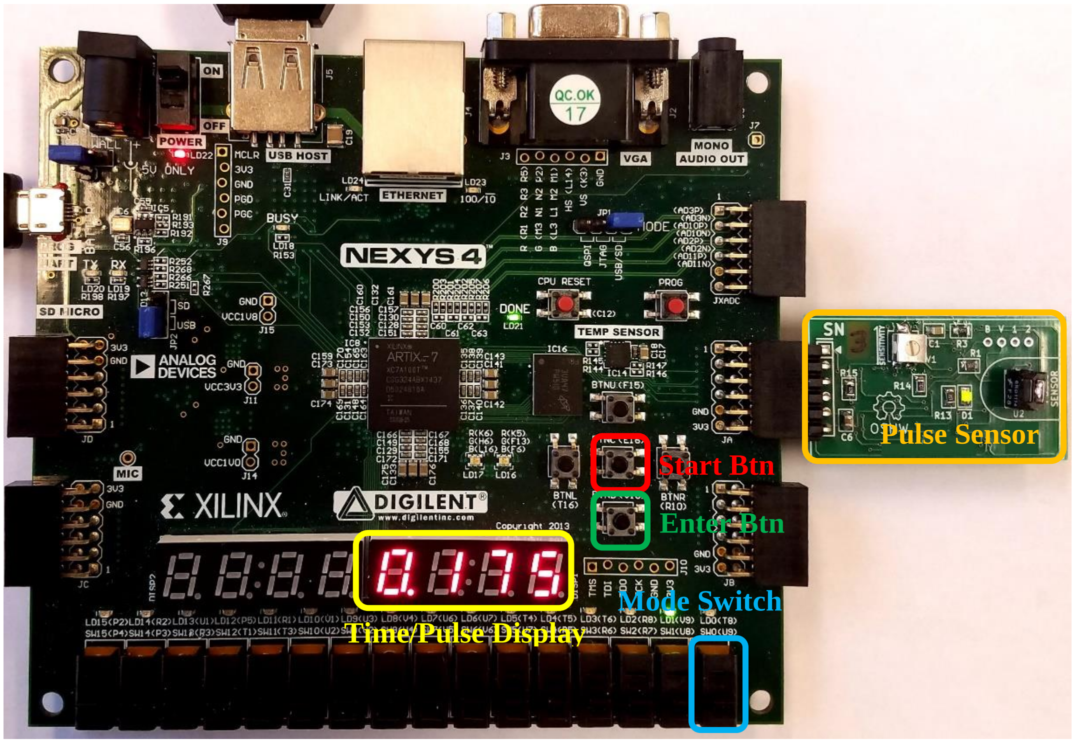
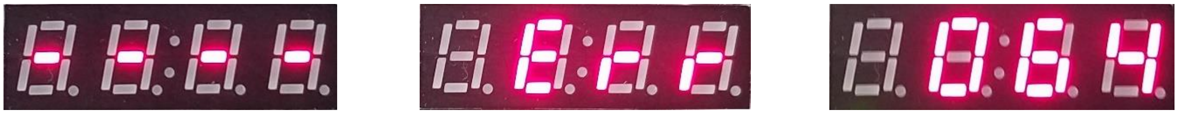


Figure 2. Implementation platform, showing a captured reaction time.



(a) Early press error (b) Exceed 10 seconds error (c) Heart rate

Figure 3. Additional system messages.

Future class periods are used to discuss common topics within digital design. After each individual topic, the class returns to the list of capabilities created during the first day of class and identifies those elements that can now be implemented based on the recently discussed topic. For example, after discussing combinational logic and minimization, students identify that with only combinational logic they can now convert a binary number into signals to drive a 7-segment display. After discussing finite state machines, the capability to “track the state of the reaction timer” can be addressed. For some elements, the design will then be developed in class either in small groups or as a class as a whole. Others will be left as elements for homework or lab assignments. Table 1 provides an overview of the topics discussed leading up to the final design and implementation of the health monitor. It also indicates the topics where the class returns to the list of needed capabilities and identify capabilities that can be addressed using the new concepts.

Week	**Topics**
1	Number Systems, Boolean Algebra, Logic Gates
2	Combinational Logic and Minimization*
3	Combinational Logic Elements*, System Verilog
4	Sequential Logic*, Finite State Machines*
5	FSM Implementation, Sequential System Verilog
6	Exam, Testing & Troubleshooting
7	ALUs & Binary Operations*, Counters*, Shift Registers*
8	Health Monitor Design (with focus on reaction timer)

*Table 1. In-class topics. (*Topics that directly address health monitor capabilities.)*

Lab

Weekly lab assignments provide students hands-on experience implementing and testing digital circuits. Table 2 shows the labs and the general topics they address. As indicated in the table, some labs have a direct application to the health monitor. In these labs, students create modules that they will directly use, or use with small modifications, in the final health monitor. Other labs indirectly help by developing proficiency in needed areas. The labs are implemented using SystemVerilog and are tested through a combination of simulation and deployment to an FPGA development board. More details on our specific development setup is discussion in *Implementation Details* below.

Week	**Lab**	**Title**	**Topic**
1	(1)	GIT and Vivado	Tool and development board introduction
2	(2)	Full Adder	Basic combinational Verilog and simulation
3	3	7-Segment Decoder	Minimized combinational logic
4	4	Binary-to-BCD Converter	Behavioral combinational logic
5	*No lab*	*Week with exam*	
6	(5)	Thunderbird FSM	Finite state machines
7	(6)	ALU	Multifunction ALU and automated testbenches
8	7	Stopwatch	Counters, time multiplexing
9	8	Health Monitor	*Reaction timer*
10			*Heart rate monitor*

Table 2. Course labs. Labs in '()' do not directly play a role in the health monitor.

Most of the labs can be completed within the scheduled three hour lab period. The two exceptions are the last two labs, the stopwatch and health monitor, which generally require some additional time outside of the lab period for most groups. Students at our institution can access the lab at any time without an instructor present so this is not an issue. A free version of the Vivado Design Suite is also available that students can use outside of the lab. At an institution where lab access is restricted, a prelab could be added for these two assignments to have students complete some work that does not require access to the physical hardware ahead of time. This would allow more time during the lab session for testing and debugging on the actual hardware.

Final Development and Implementation

Once all of the topics have been covered to address the individual capabilities, the next step is to develop an overall design for the system and then to implement the system. A 50 minute class period is dedicated to having students develop a block-level design for the reaction timer portion of the system. The period begins with a review and demonstration of the desired behavior of the reaction timer. The collection of elements that have been developed by students at previous points in the course is provided (on the board or as a handout). Table 3 lists these elements, including where in the course they are developed or discussed. Students are broken up into small groups (approximately four per group) and tasked with combining these components to achieve the functionality of the reaction timer. Given our class sizes of 20-30 students, we have the groups develop their diagrams on different sections of the board. This allows the instructor to easily observe how the groups are progressing and provide feedback and direction as necessary. Groups are generally able to create reasonable, complete or mostly complete diagrams in about 30 minutes.

After the groups have finished, we discuss some of the differences between designs developed by different groups. There are two differences that are fairly common. One revolves around how to count time and then have values to feed to the seven segment display. The two common approaches are to use either 1) a BCD counter or 2) a binary counter plus a binary-to-BCD converter. From the perspective of the general design, either is acceptable. For later implementation, the BCD counter approach is advantageous if the binary-to-BCD counter designed in lab does not produce four decimal digits (as is the case of the converter developed in our Lab 4). The second common discussion point is around how to display non-hexadecimal characters on the 7-segment display for the two error messages. In the earlier stopwatch lab (Lab 7), a block-level design to control a multi-digit display is provided that uses a single 7-segment decoder. This approach works for that lab since all of the values displayed are numbers supported by the 7-segment decoder from Lab 3. Based on this, one option is to retain the design with a single 7-segment decoder and modify the decoder to use certain input encodings for the new characters, '-' and 'r', and have the error producing logic output the appropriate 4-bit encoding as appropriate. A second option, which leaves the original 7-segment decoder unchanged, is to have multiplexers select 7-bit values that correspond to the 7-segments to be displayed. This requires multiple 7-segment decoders, but leaves the decoder implementation unchanged. We talk about the trade-offs with both options.

With the reaction timer design in place, the remainder of the class (5-10 minutes) is used to discuss how the reaction timer block just designed could be combined with a heart rate monitor block the students will design themselves to create an integrated system where a switch selects the outputs of one of the two elements to feed to the 7-segment displays.

Over the next two lab sessions, students create their health monitor. This includes implementing the reaction timer based on the block diagram developed in class and then designing and implementing the heart rate monitor and integrating the two elements. To keep the students on track, they must demonstrate a functioning reaction timer by the start of the second lab period. This requirement, which was not included the first semester using this project, led to more consistent progress on the project and better outcomes in the end.

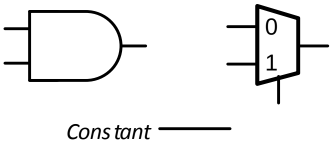
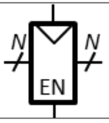
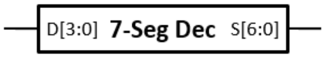
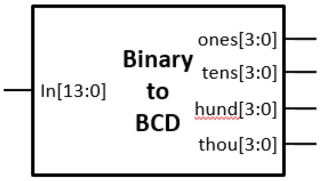
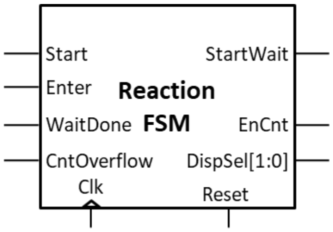
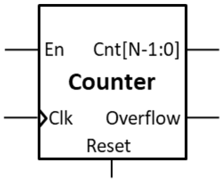
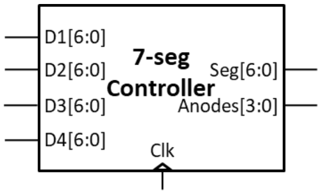
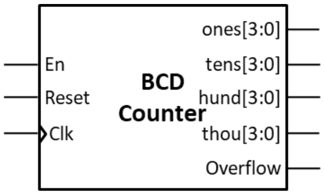
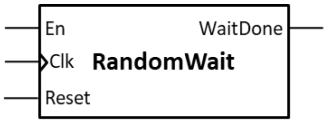
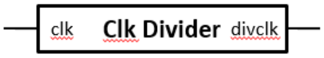
Element	Developed	Symbol
Basic logic elements (OR, AND, NOT, mux, constant, etc.)	In class (and homework)	
Flip-flops	In class	
7-Segment decoder	Lab 3	
Binary-to-BCD	Lab 4*	 In[13:0] Binary to BCD ones[3:0] tens[3:0] hund[3:0] thou[3:0]
Reaction FSM	Homework	 Start Enter WaitDone CntOverflow Reaction FSM StartWait EnCnt DispSel[1:0] Clk Reset
Counter	In class (and homework)	
7-Segment controller	Lab 7	 D1[6:0] D2[6:0] D3[6:0] D4[6:0] 7-seg Controller Seg[6:0] Anodes[3:0] Clk
BCD counter	Lab 7	 En Reset Clk BCD Counter ones[3:0] tens[3:0] hund[3:0] thou[3:0] Overflow
Random wait	Homework	
Clock divider	Provided (Lab 7)	

*Table 3. Individual elements of reaction timer. (*The binary-to-BCD converter developed in lab supports a 7-bit input, so only has ones, tens, and hundreds outputs.)*

Along with the actual implementation and demonstration, the students also submit a report for the lab. The report includes multiple diagrams that show the final state of the reaction timer, pulse monitor, and overall health monitor; commented SystemVerilog of their implementation; and a brief reflection on the difficulties experienced during the lab and how they would approach the lab differently if they were to repeat the design and implementation.

Implementation Details

We use a Digilent Nexys4 development board as the target platform and SystemVerilog and Xilinx Vivado to implement the design and configure the board. Students are introduced to the design tools and the development platform through the first lab (see Table 2) and utilize them in all of the other labs. In general, any HDL and target platform should work. The only elements needed, aside from the pulse sensor, are four 7-segment displays, two buttons, and a slide switch, which are available on almost any contemporary development board. Alternatively, external components could be used.

The only piece of HDL provided to the students is a clock divider. Everything else students create themselves at some point in the course (in homework, in class, in a prior lab, or during the health monitor lab).

The pulse sensor hardware is provided to the students. This is an analog component that is outside the scope of the course or the background of the students given the ordering of our curriculum. From our experience, the choice and function of this sensor can have a substantial impact on the quality of the heart rate that can be measured. Since the course is on digital logic, we wanted a sensor that provided a digital representation of a user's pulse (i.e., a signal that has a digital high and low section for each heart beat). We also wanted one that was simple to use and functioned reasonably across a range of users. We focused on reflective photoplethysmography-based options. In this approach, a light source (in our case infrared) illuminates a user's finger and the amount of light that is reflected from the finger is observed. The amount of light reflected changes as the volume of blood in the finger changes over the course of a heartbeat. The magnitude of the change, however, is small and so needs amplification to be easily used.

There are some inexpensive, hobby sensors available commercially, but the ones we found only provide an analog version of the pulse [9], so would have required additional external circuitry to create a digital signal. For this reason, we designed a sensor in house. This also allowed us to customize the interface so it nicely connected to the PMod interface on our development board.

Figure 4 shows the physical pulse sensor. The IR LED and phototransistor that a user rests their finger on is on the right and the pins that connect to the development board are on the left. The sensor includes an LED that shows the output of the digitized pulse signal to help with debugging and to aid assessment of the heart rate functionality. The Appendix includes the schematic for the sensor and a brief overview of its design. The sensor takes about five seconds to stabilize after a finger is placed on the sensor and then provides reasonable readings so long as the user keeps their finger still. An example of the single-bit output of the sensor is shown in Figure 1. The example shows five new pulses in the five second window. If this repeated for three consecutive five-second sampling windows, this would result in a calculated heart rate of 60 BPM.

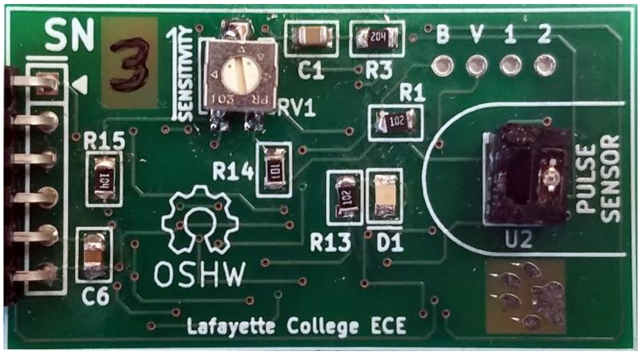


Figure 4. Custom pulse sensor.

Possible Variations & Extensions

In our context, the health monitor is the first notable design project completed by students in the major (this is the first course in the major). For this reason, we provide a large amount of scaffolding to guide students through the design process. More freedom and flexibility could be given if students already have exposure to designing systems with multiple elements.

The design of the pulse sensor itself could provide an interesting discussion or design problem for an analog electronics course. While the basic idea of amplifying an analog signal into one with digital logic levels is not especially complex, creating a design that was stable and functional across a range of users was more challenging than initially expected.

Results

We analyzed student achievement in implementing the health monitor, student perceptions of the health monitor relative to objectives for the design project, and student reflections on the design and implementation experience. The results are from the second time the health monitor project was used. There were a total of 24 students in the class, with lab sections of 11 and 13 students. While these class and lab sizes are relatively small, there is nothing that should prevent the project from being used in classes with more students or larger lab sections.

By the end of the two week implementation period, a majority of groups (69%) had a completely functioning health monitor. Another 15% had a system with all elements present except that the pulse monitor did not accurately count the pulse signals (a number for the heart rate was displayed and it changed, but it did not accurately reflect the BPM translation of the pulse signal observed on the sensor). One group had the reaction timer but not the pulse monitor functioning and a final group had only partial elements for both components. While not 100% success, we believe these results show that the health monitor design is a tractable problem that students in their 2nd year can address without prior knowledge in the area.

To assess the impact of the health monitor project relative to the objectives outlined at the beginning of this article, we surveyed students at two points within the course. The first occurred after the first month of classes. This was after the health monitor had been repeatedly revisited in class after individual concepts had been discussed, but before the development of the system-level design in class or the implementation in lab. The second survey occurred at the end of the semester, after students had completed both the health monitor and a second design project over the final third of the semester. Table 4 and Table 5 show the questions in each survey and the breakdown of responses for each. The second survey also asked if students previously saw the application of digital design to health related issues.

	Agree	**Neutral**	**Disagree**
The Health Monitor has helped provide context for the application of digital circuits.	67%	33%	0%
Periodically returning to the list of Health Monitor system elements and identifying elements that recently covered material address has helped connect different topics.	74%	26%	0%
The Health Monitor has helped show meaningful uses for individual course topics.	74%	26%	0%

Table 4. Mid-project student survey results.

In the middle of the health monitor development, a majority of students indicated that the project helped provide context for the course, helped connect different topics from the course, and showed a meaningful use of course topics. Each of these corresponds to one of the initial objectives and suggests that, for many students, the health monitor is achieving the goals we desire.

	Agree	**Neutral**	**Disagree**
The Health Monitor helped show a socially relevant application of Digital Logic and Electrical and Computer Engineering.	90%	10%	0%
The Health Monitor helped motivate you to learn course topics.	53%	42%	5%
The process of designing and building the Health Monitor through a combination of in-class discussion, homework, and lab assignments helped prepare you to design the final Pin-Based Access System with less direct supervision.	90%	10%	0%
Prior to the course, did you see the application of digital design to health related issues?	Yes – 26%		No – 74%

Table 5. End-of-semester student survey results.

At the end of the semester, three quarters of students indicated that prior to the course they had not considered the application of digital design to health related issues. At the same time, a large majority indicated that the health monitor showed a socially relevant application of the course material. This supports the conclusion that the health monitor is achieving the goal of showing applications of digital logic outside of preconceived ideas. Given that prior work has shown that underrepresented demographics are generally more interested in socially relevant applications [6, 7, 8], this also indirectly suggests that the project would be of interest to a broad range of students.

A large majority of students also indicated that the structured designing of the health monitor helped prepare them for the final design project were they had to design another multipart digital system as part of a small group. For the final design project, all groups created functioning implementations, providing additional indirect support to the students' responses.

Finally, while not an explicit goal when we developed the health monitor project, a slight majority of students indicated that it helped motivate them to learn course topics. Most of the remaining students were neutral on this point.

As part of the health monitor lab report, students were asked to reflect on the difficulties they had and what they would do differently if they were to repeat the design and implementation. The most common comment on what they would do differently was to test components individually before integrating them. The fact that the responses centered on implementation and testing makes sense to us as a large portion of the high-level design and partitioning was developed as a class. While the importance of implementing and testing components individually before integrating them was discussed multiple times in class and lab, and students even had structured experience with this in earlier labs, our experience has been that, when left to their own devices for the first time, many students skip individual testing. From informal discussions with students and our own observations, many of the modules students create appear so small and straight forward that students assume that they must work. This assumption can be especially problematic when developing in Verilog were small omissions or misunderstandings, such as around implementing combinational versus sequential logic, can result in problems that manifest in unpredictable ways when multiple elements are integrated. While there may be additional preemptive things that could be done to further encourage component level development and testing, our experience has been that this lesson is something that needs to be learned by experience for many students. This project seems to be a reasonably constrained setting where this lesson can be learned without disastrous consequences.

Conclusion

This paper presents a health monitor design project for an introductory digital design course that achieves multiple objectives. The health monitor provides a realistic, socially relevant application of digital logic. It incorporates many common elements of a digital design course into a single system and provides a setting to illustrate how to combine multiple simple elements to achieve more complex functionality. With appropriate structuring, it fits within an 8-10 week portion of a course and can be successfully implemented by students with no prior experience in the field. Student feedback and performance suggests that the health monitor project is achieving its goals and helping students learn and appreciate the application of digital logic.

References

[1] Association for Computing Machinery and IEEE Computer Society, "Computer Engineering Curricula 2016 (CE2016)," ACM/IEEE, 2016.

[2] Association for Computing Machinery and IEEE Computer Society, "Computer Science Curricula 2013," ACM/IEEE, 2013.

[3] XPrize, "Qualcomm Tricorder XPrize," 2017. [Online]. Available: https://tricorder.xprize.org/. [Accessed 5 December 2017].

[4] D. Harris and S. Harris, Digital Design and Computer Architecture e2, Morgan Kaufmann, 2012.

[5] F. Vahid, Digital Design, : zyBooks, 2017.

[6] I. Busch-Vishniac and J. Jarosz, "Can Diversity in the Undergraduate Engineering Population Be Enhanced Through Curricular Change?," *Journal of Women and Minorities in Science and Engineering*, vol. 10, no. 3, pp. 255-281, 2004.

[7] J. S. Brotman and F. M. Moore, "Girls and Science: A Review of Four Themes in the Science Education Literature," *Jounral of Research in Science Teaching*, vol. 45, no. 9, pp. 971-1002, 2008.

[8] D. Knight, E. Mappen and S. Knight, "A review of the literature on increasing the representation of women undergraduates in STEM disciplines through civic engagement pedagogies," *Science Education and Civic Engagement*, vol. 3, no. 1, pp. 36-47, 2011.

[9] PulseSensor.com, "PulseSensor Amped," [Online]. Available: https://pulsesensor.com/products/pulse-sensor-amped. [Accessed 12 December 2017].

Appendix – Heart Rate Sensor

Our initial pulse sensor design included three major element: a reflective object sensor, a pair of bandpass amplifiers to amplify the small sensor signal, and a comparator to produce a rail-to-rail digital logic-type signal. While this worked to some extent, it did not work reliably across a broad range of users. With this basic design as a starting point, our Lab Director set about creating a more consistent sensor. Two main issues were identified. One was that the nominal voltage output of the sensor varied non-trivially for different users under a fixed bias. The second was that the amplitude of the sensor output varied based on both the individual user and the state of the user's finger (ex., a cold finger resulted in a lower amplitude signal). These two issues created problems for the downstream amplification and comparison stages. To address these, an automated bias control loop and manual sensitivity control potentiometer were added to the sensor. Figure 5 shows the final design of the pulse sensor, with the reflective sensor on the bottom left and the digitized pulse signal on the right. After five seconds for stabilization, the sensor produces a good pulse signal so long as the finger remains steady on the sensor.

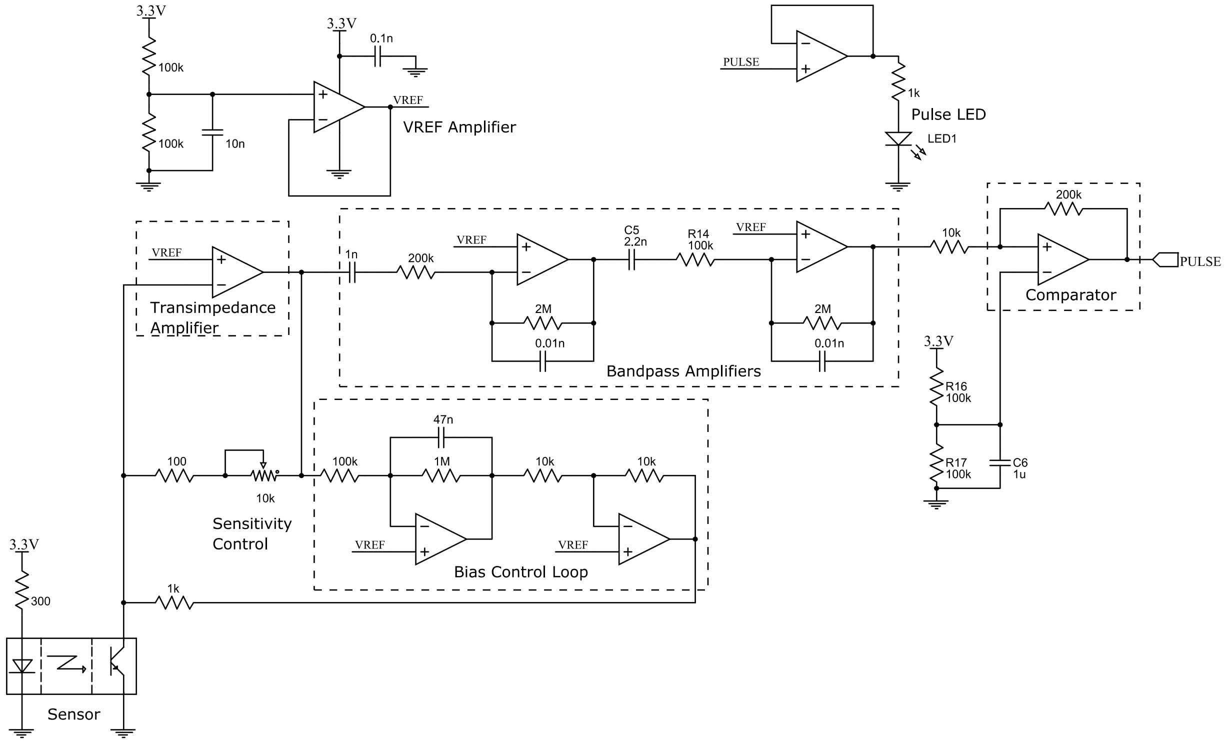


Figure 5. Pulse sensor schematic.