Technical Aspects of Creating and Assessing a Learning Environment in Digital Electronics for High School Students

Adam S. El-Mansouri, Herbert L. Hess, Kevin M. Buck, Timothy Ewers

Microelectronics Research and Communications Institute
Department of Electrical Engineering, University of Idaho
Moscow, Idaho

Abstract

To develop an interest and an understanding of digital electronics for high school students, we have created digital electronic projects using a Field Programmable Gate Array (FPGA). The approach is module-based; the origin and application of those modules is described. The educational process is segmented into small learning projects, where the function of the modules is presented, and a main project. Our main project example is a digital clock with alarm. This paper addresses the development of the modules, uses the clock as an example of a project, and demonstrates the technical aspects of the methods.

Introduction

Recently, engineering and education faculty members at the University of Idaho teamed with local high schools to create an integrated hardware/software environment to inspire high school students into electrical and computer engineering. Their intent was to develop a project-based learning environment that overcomes the learning curve often associated with introducing integrated software and digital hardware design. In that way, they could have the students creating new "things" quickly and with a "personal touch", a strong motivator for them to begin to understand what engineering is. As such, we began with a Field Programmable Gate Array (FPGA) learning system designed for college sophomores and juniors. In this environment, we developed some fundamental modules for a range of projects. Then we provided a few examples of increasing difficulty on how to assemble these modules to perform a useful function. After seeing these examples, the high school students are encouraged to try themselves, first at a specified project and then to create their own task and solution.

Hardware

The development environment consists of a hardware portion and a software portion. For hardware, we chose a circuit board, designed at Washington State University and sold by Digilent® Corporation [2], to help teach digital logic with applications. This board consists of a Field Programmable Gate Array (FPGA) mounted onto a standard-format printed circuit board. The pins of the FPGA connect to various input/output devices such as LEDs, switches, buttons, seven-segment displays, and standard connectors for interface to user-designed expansion interface boards. The Digilent® board contains a 50MHz oscillator which is used as the master clock signal. The Digilent® board has a flash memory to store user-generated programs. The user writes these programs in a software development environment provided by Xilinx® for installation in a host computer. (More on the software in the next section of this paper.) Circuits that the student designs in the development environment are automatically coded and then downloaded on command to the FPGA through an RS-232 interface. Downloading is simple: it is just a single press of a button on the host computer. The Digilent® board is shown on the left in Figure 1, ready for programming. User-designed peripherals, in this case a piezoelectric buzzer, wired onto a breadboard, are shown on the right in Figure 1.

Software

As mentioned in the previous section, the user generates appropriate programs in a software development environment. These programs are automatically coded for downloading into the FPGA. The software development environment that we chose is the Xilinx® ISE Webpack package [1]. This package has a user-friendly interface and three methods of entering the information to specify a circuit. These methods lend themselves nicely to a logical approach to learning digital electronic circuitry, software, and methods.

The software provides three methods of constructing a circuit: schematic capture, VHDL coding, and Verilog coding. Schematic capture allows for constructing circuits from predefined symbols that are part of the software package and also from user generated symbols. This is the preferred method for beginners, such as high school students, as it requires a short learning curve and provides a visual structure of the system. It also permits the instructor to assemble modules in advance, so the student can begin with a high-level, block diagram approach to circuit construction. (In the next section of this paper, we develop an appropriate set of modules for a nice range of projects.) The other two methods of circuit entry are VHDL and Verilog® code. Coding in either language is more efficient than schematic capture, making it the method of choice for veterans. However, it has a longer learning curve and, therefore, is an optional advanced topic for high school students.

As with similar development software environments, the Xilinx® software helps the user build the circuit, assigns the designated pins, checks the work for errors, synthesizes the circuit automatically for downloading to the FPGA, and engages a single-button download command. The only part required by the user is to assemble the appropriate modules that comprise the circuit and then assign the input/output (I/O) interfaces of the program to the pins of the FPGA. The software does the rest. The cost of one Digilent® Pegasus FPGA board system from Digilent® is $89. The software is free to students, provided by Xilinx®.

Module-based concept

The idea is to provide students with an ability to build something useful with a minimum of detailed technical knowledge of digital systems. This requires development of a set of blocks that each performs an appropriate function that tends to recur in digital designs. Such functions include arithmetic, counter, and interface functions. The software comes with basic digital blocks that are used along with customized blocks. We developed and documented each of these blocks ahead of time. In other words, each of these blocks is created in VHDL code before the project is introduced to the students. Students then use these schematic blocks as fundamental elements as they build their respective project. The idea is to provide students with the ability to construct real life systems with a minimum apparent complexity while requiring minimum programming knowledge.

To identify the blocks that we would need, we proposed about two dozen possible projects that the high school students might find of interest. We then subdivided these projects into fundamental elements. In doing so, several functions consistently recurred. From these recurring functions, we wrote VHDL code to make blocks for the high school students to use as their basic elements in their designs. We then introduced these functions with short examples to show how they could be used.

The blocks that we developed from our investigation and used in our example project are as follows:

1) Debounce: This block provides protection against debounce of switch and button signals.
2) Clkdivider: Reduces the clock frequency from a 50MHz to the desired frequency.
3) Count60cycles: This is counter which counts 60 cycles. The output is a clock signal every 60 count as well as the value of the count in decimal represented by two binary number for the ones and tens place.
4) Count24cycles: This is counter which counts 24 cycles. The output is a clock signal every 24 count as well as the value of the count in decimal represented by two binary number for the ones and tens place.
5) Segdriver: This block is a seven-segment driver which takes inputted four binary numbers and displays them on four seven-segment displays.
6) Comparator: This block compares to binary values and outputs a one if they are equal. It also contains a reset and enable input.
7) Pulsecontrol: This block outputs a clock frequency with a specific duty cycle when the input is set to one.
8) Mux2to1: This block is a simple 2 to 1 mux.
9) Btndemux: This block is composed of two demux's with a common control signal.
10) Displaycontrol: This block takes the eight binary numbers and outputs either the top four or the bottom four depending on the whether the control signal is set to one or zero. The block also outputs the eight numbers on combined into one vector as well.
11) Analog to Digital Converter: This block converts an analog voltage to an eight-bit digital number.
12) Digital to Analog Converter: This block converts an eight-bit digital number to an analog output voltage.

These blocks provide a basic function set that high school students can use to build a host of projects. In the next section, we describe how one of these projects is built.

Example Project: An alarm clock

Various projects can be constructed from this FPGA system. The approach is to show a good example project, give the students the opportunity to examine and modify the example first and then assign a second project that they can design from start to finish. This example also serves as a nice vehicle to show the technical aspects of how we created the module-based project development system.

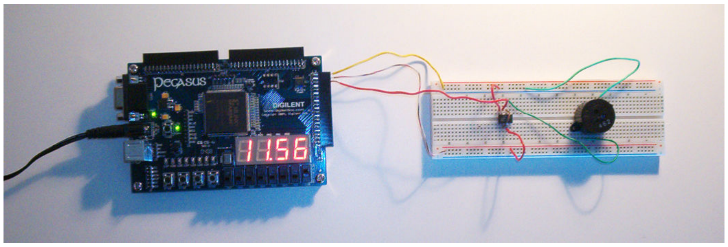

Figure 1: Digital clock system

The example is a digital clock with alarm which would provide a practical application, something that the average high school student identifies well. An alarm clock has a reasonably well-understood elementary function: it keeps accurate time and gives an alarm at a user-programmed point in time. The clock provides a practical application and physical system for the high school students, one that is well within their reach technically. An alarm clock presents the opportunity to add extra functions created by the student, in other words, a student's own "personal touches" that add spice to the project design process, such as unusual alarm sounds, different ways of indicating the time, creative alarm sequences, ways to speed up testing of the alarm, etc. This is important because it allows the students to investigate and learn design methods while "having fun". An alarm clock's fundamental concepts also lend themselves to application to future, more complex, projects that require timing. In Figure 1, the completed project is shown: the Digilent board, an LED output, and a piezoelectric buzzer.

The clock is a real time clock with an alarm. The clock displays the time on four seven-segment displays which display the hour and minute in decimal form. The clock can be set to run at two speeds either real time mode or fast mode determined by the switch "rate". The clock is also built with an alarm which outputs a signal to an LED as well as a buffer which drives a piezoelectric buzzer. For simplicity, the buffer used for our circuit is an OPAMP. Switches and buttons are used to set the clock and alarm time as well as disable and reset alarm.

The operation of the clock is shown in the block diagram in Figure 2. The FPGA contains a 50MHz clock which is inputted to a divider that provides an output clock signal of either 1Hz or at a higher rate determined by the input switch "rate". The divide clock signal is then inputted to the first counter which counts 60 cycles and outputs the minute's time and a signal to a second counter. The second counter counts 24 cycles from the output signal of the first counter and outputs the hour time. The clock and alarm time can be initialized by enabling the Clock setting switch or Alarm setting switch. The display output of either the clock or alarm time is controlled by the alarm setting switch. The desired output to be displayed is passed by the display control block to the display driver block. The comparator receives the initialized alarm time and current clock time and provides comparison. If the times are equal the Alarm is enabled then a signal is sent to the pulse generator which outputs a signal to a buzzer. The buzzer continues to sound until either the Alarm has been reset or disabled.

The students build the clock in the schematic editor of the Xilinx ISE Webpack software. The clock is divided into VHDL coded blocks. The blocks appear to the student in the schematic editor as symbols. Each block contains a description of its particular function. The blocks are then wired together as shown in the Figure 3. The students will be provided with the diagram shown in Figure 3 as well as an FPGA pin mapping table. After the circuit is constructed, it is then verified and downloaded to the FPGA which is then ready for testing.

The output of the clock time to the seven-segment driver (segdriver) is controlled by the DisplayControl. If the enable alarm setting is set to zero then the clock time is displayed. If it is set to one then the Alarm time is displayed. DisplayControl also passes the clock and alarm time to the comparator.

The comparator is used to compare the alarm time to the clock time and output a one to the PulseGenerator if they are equal. The output will remain a one until reset is set to one or enable Alarm is set to zero. Debounce blocks are used for protection against debouncing. The muxes are used to direct the input button signals to the clock or alarm blocks depending the desired setting.

Implementation

This project environment is being implemented in a high school in the 2004-2005 academic year. Therefore, formative assessments are ongoing at the time of presentation. This paper is intended to show the technical aspects of how we designed the project-based system. We will follow up with a paper on the pedagogical aspects and appropriate assessment in a paper at the next ASEE conference.

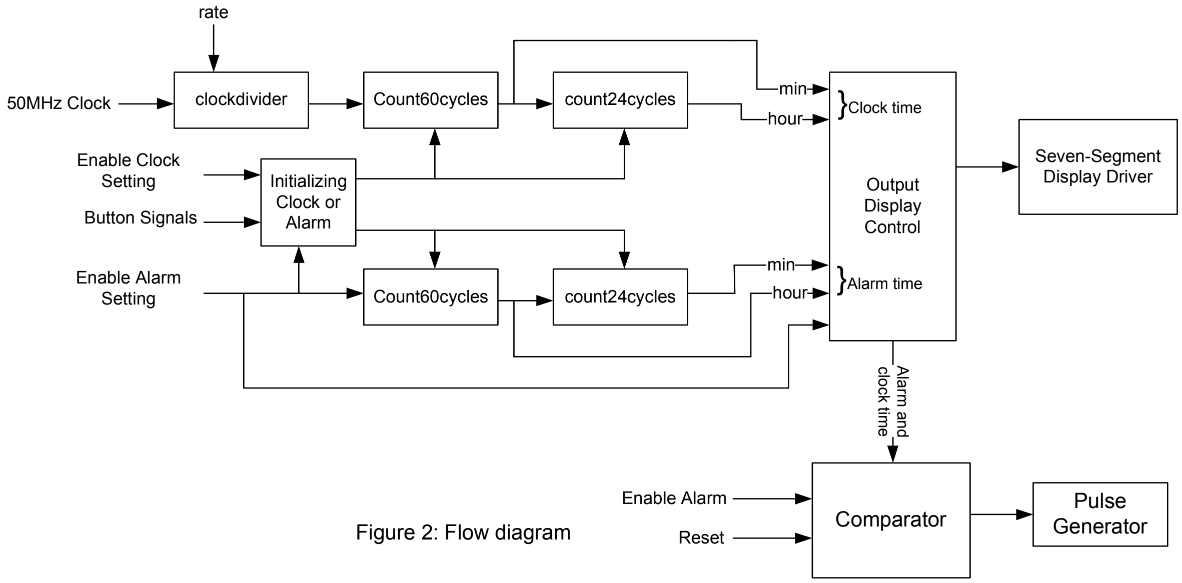

Figure 2: Flow diagram

Summary

The projects that can be constructed with these types of boards are legion. The first example project is a digital clock with alarm which can be set to run on real time or in a faster mode. The circuit is constructed as a schematic, using for circuit elements the modules represented as icons. The alarm system is a piezoelectric buzzer which is buffered by an operational amplifier which is connected to the FPGA. The FPGA provides a pulse signal at a certain frequency and duty cycle which causes the buzzer to sound. Varying the sound is a first modification to inspire student creativity. As the students completed the introductory project, they were encouraged to create their own project using the modules, including requests for new functions and modules.

This paper has presented technical details of creating a useful environment to introduce high school students to the fun of creative digital design. The FPGA system allows students to easily construct various projects at different levels of complexity. It would usually require students a fair amount of time to construct a large system such as a clock, but by segmenting large and complex projects into schematic blocks with particular functions it allows students to be able to take those blocks and build large practical projects. The goal is to segment the blocks in such away that would allow the system to be constructed with minimum amount of blocks but at the same time not limit those blocks to a particular project but rather to a general function which could be used in other projects which is demonstrated by the clock system.

This paper introduces technical design issues for this project environment. Pedagogical issues and assessment will be addressed in a paper at the next ASEE conference.

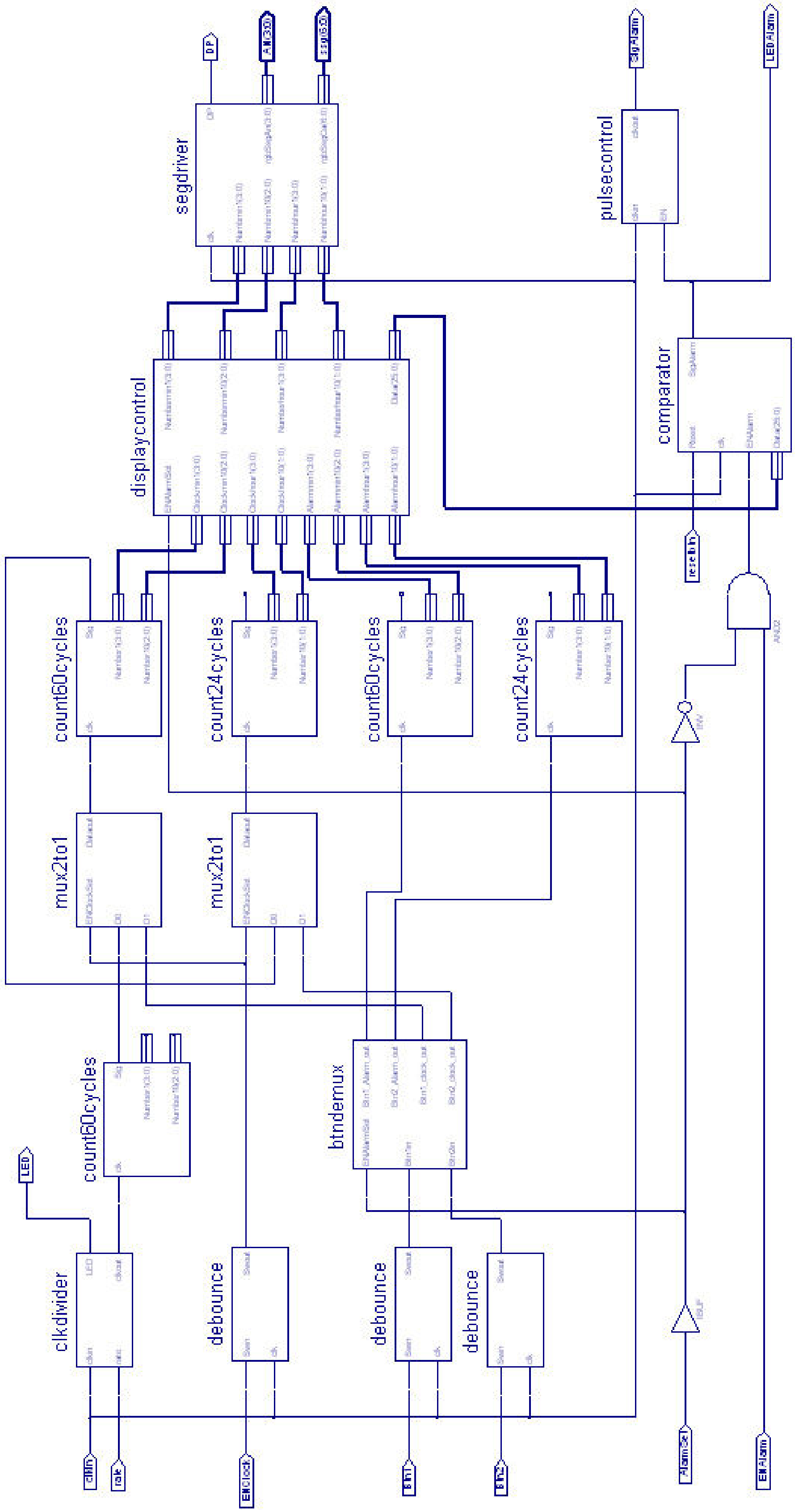

Figure 3: Clock circuit complete diagram.

BIBLIOGRAPHIC INFORMATION

[1] "Free ISE WebPack." Xilinx. 05 Jan. 2005 <http://www.xilinx.com/xlnx/xebiz/designResources/ip_product_details.jsp?sGlobalNavPick=PRODUCTS&sSecondaryNavPick=Design+Tools&key=DS-ISE-WEBPACK>.

[2] "Digilent Pegasus." Digilent Inc.. 05 Jan. 2005 <https://digilent.us/Sales/Product.cfm?Prod=PEGASUS>.

BIOGRAPHICAL INFORMATION

ADAM S. EL-MANSOURI
Adam El-Mansouri is a junior working toward a B.S. degree in electrical engineering at the University of Idaho. He is working as a lab assistant at the microelectronics research and communications institute at the University of Idaho. His interests include microelectronics power converter design and digital control systems.

HERBERT L. HESS
Herbert L. Hess received the PhD degree from the University of Wisconsin-Madison in 1993. He served on the faculty of the United States Military Academy from 1983-1988. In 1993, he joined the University of Idaho, where he is Associate Professor of Electrical Engineering. He received the Best Paper Overall Award for the 1999 ASEE Annual Conference. His interests are in device and circuit aspects of power electronic energy conversion.

KEVIN M. BUCK
Kevin M. Buck received the MS degree from the University of Idaho in 1999 where he is currently pursuing his PhD. In addition he is currently a full time research engineer for the Microelectronics Research and Communications Institute. His interests include high-voltage IC devices, FPGA design, ASICs, microelectronics power conversion techniques, and Extreme Environments.

TIMOTHY EWERS
Timothy Ewers is a scientist turned science educator. Formally trained as a toxicologist and then risk analyst he got interested in the science education while working with the general public on risk management issues. His interests include research into developing more authentic science experiences and science thinking in K-12 schooling and in promoting interest in math and science careers. He received a Ph.D. from the University of Idaho where he is an Assistant Professor in the College of Education.