AC 2009-217: A LABVIEW FPGA TOOLKIT TO TEACH DIGITAL LOGIC DESIGN

Troy Perales, Texas A&M University

Troy Perales graduated from the Electronics Engineering Technology (EET) Program in 2007 and is currently pursuing a Masters degree in Computer Science at Texas A&M University. He is graduate assistant for teaching within the EET Program and is responsible for the development and delivery of laboratories associated with digital design and analog electronic devices.

Joseph Morgan, Texas A&M University

Joseph A. Morgan has over 20 years of military and industry experience in electronics and telecommunications systems engineering. He joined the Engineering Technology and Industrial Distribution Department in 1989 and has served as the Program Director of the Electronics and Telecommunications Programs and as the Associate Department Head for Operations. He received his BS degree in electrical engineering (1975) from California State University, Sacramento, and his MS (1980) and DE (1983) degrees in industrial engineering from Texas A&M University. His education and research interests include project management, innovation and entrepreneurship, and embedded product/system development.

Jay Porter, Texas A&M University

Jay R. Porter joined the Department of Engineering Technology and Industrial Distribution at Texas A&M University in 1998 and is currently the Program Director for the Electronics and Telecommunications Programs. He received the BS degree in electrical engineering (1987), the MS degree in physics (1989), and the Ph.D. in electrical engineering (1993) from Texas A&M University. His areas of interest in research and education include product development, analog/RF electronics, instrumentation, and entrepreneurship.

© American Society for Engineering Education, 2009

A LabVIEW FPGA Toolkit to Teach Digital Logic Design

Abstract

National Instruments (NI) has added the ability to graphically design digital circuitry in its LabVIEW development environment that can directly interface to and program Xilinx FPGA devices. The Electronics and Telecommunications Engineering Technology (EET/TET) Programs at Texas A&M University selected the Xilinx Spartan-3E Starter Board available from Digilent as the platform for use in its sophomore-level digital design and microcontroller architecture courses. In addition, the EET/TET faculty decided to use the NI LabVIEW graphical development environment for both of these courses. The resources offered by National Instruments in their commercially available LabVIEW FPGA Module were found not to be intuitive enough to fully support the digital design requirements of these courses which included both combinatorial as well as sequential logic. During summer 2008, NI sponsored a development project in partnership with the EET/TET Programs. Under the direction of the EET/TET faculty, a graduate student (former EET undergraduate student) was funded by National Instruments to develop a digital logic design toolkit that integrated into the NI LabVIEW software and allowed students to design using typical combinatorial and sequential logic devices that ranged from AND, OR, NOT gates to Flip Flops, Counters, Comparators, Selectors, Decoders, etc. This newly developed toolkit was tested during the Fall 2008 semester and version 1.0 of the toolkit should be available to other universities in Spring 2009. This paper presents the toolkit developed by students and faculty at Texas A&M University and recommendations for integrating the toolkit into the digital design sequence of engineering and engineering technology programs.

Introduction

One of the primary focuses of the Electronics and Telecommunications Technology (EET/TET) Programs at Texas A&M University is providing each graduate with an ability to design hardware and software products and systems using industry-grade toolsets. Use of these tools to develop embedded data acquisition and control systems begins with the first technical courses at the sophomore level. The curricula seek to expose the students to a wide range of software (assembly, C, and graphical) languages as well as providing experience in utilizing multiple microcontroller development platforms. Recently, the programs, through the addition of a new faculty member, re-introduced field-programmable gate array (FPGA) technology to its hardware suite and Hardware Description Languages (specifically VHDL) to its programming languages. The intent is to provide students with a spectrum of hardware technologies and programming languages to choose from in implementing their Capstone Design Projects during their senior year of classes.

As with many engineering and engineering technology programs, the EET/TET programs recognized the importance of Field Programmable gate Array (FPGA) technology to the development of the digital curriculum.[1,2] The faculty decided to introduce the FPGA technology at the beginning of the technology courses required by both educational programs so that all students would have access to these tools. Working with the Xilinx university programs[3] and engineering support personnel, a Digilent Spartan-3E Development board[4] was chosen for use in

the introductory digital design class. This board had the ability to support both this course and the follow-on course in microcontroller architectures. Students in the digital design course used the board as the target platform for combinatorial and sequential logic design laboratory assignments as well as their implementation of an autonomous embedded controller for a small mobile platform. Although the FPGA-based board made design iteration and testing much faster and efficient, the Xilinx development environment had a rather steep learning curve and was much more than required to support digital design using basic gates and devices.

In discussions with National Instruments, a long time industry partner of the EET/TET Programs, it was agreed that the LabVIEW FPGA graphical programming environment[5,6] was non-intuitive and insufficient for teaching digital logic. However, the programming environment could be augmented with a toolkit that would allow students to design digital circuits using device icons and then have these circuits compiled for direct downloading to the Xilinx FPGA board. In the Summer 2008 semester, NI funded a curriculum development project at Texas A&M University where a graduate student working with the EET/TET faculty developed and tested a new toolkit for LabVIEW that would meet the requirements of the digital logic design course and provide a better basis of understanding for the students as they moved into the microcontroller architectures course.

This paper presents the toolkit developed by students and faculty at Texas A&M University and recommendations for integrating the toolkit into the digital design sequence of engineering and engineering technology programs. As a result of this development project, the EET/TET Programs are now able to introduce both FPGA technology and the LabVIEW programming environment to its sophomore-level students. Through the innovative use of VHDL to build the iconic digital gates and devices, the students have a direct transition to the follow-on course in the digital sequence which will utilize VHDL to implement embedded designs. Expanding the knowledge base of the students will better prepare them for their demanding Capstone Design Projects accomplished in the senior year.

VHDL-Based Toolkit

When starting this project, it was quickly recognized that while students in the introductory courses need to work at the gate level in order to realize a firm grasp of the fundaments of digital design, in follow-on courses, they would need to quickly migrate to the use of Hardware Description Languages (HDL) in order to fully use the potential of their FPGA devices. For this reason, it was decided that through the LabVIEW development environment, the students should graphically design their circuits but also have the capability of seeing the HDL equivalent of the devices they were using. Thus, each device in the library had to allow the students to drill down and view the implementation in VHDL.

Thus, the Digital Logic VI Library leverages the LabVIEW FPGA's HDL Interface Node. The HDL Interface Node provides the ability to use VHDL code rather than LabVIEW code within an FPGA VI. In order to interface properly with LabVIEW data flow, the HDL Interface Node is defined through its properties dialog box. The inputs and outputs are defined and configured in the Parameters Tab and the module is defined in VHDL under the Code Tab as shown in Figure 1. Using this methodology allows the interested student to "drill down" into each digital device

and see how it is implemented in VHDL. Therefore, in the introductory class, they can design systems graphically and start to learn VHDL at the same time.
Because the students are using a graphical interface, the iconic representation of digital devices is just as important as their functionality. While developers can create icons to represent their VI's using the built-in icon editor, the need for flexibility and advanced capabilities dictated the use of Microsoft Visio which includes image effects and supports scalable vector graphics. For the Digital Logic VI Library, each device is given an appropriate icon that conveys its functionality. Colors are used to distinguish one family of devices from another—combinatorial devices are yellow and sequential logic devices are orange. After creating the icons in large form, GIMP was used to add a drop shadow and resize to a 32x32 pixel image before importing into the LabVIEW icon editor.

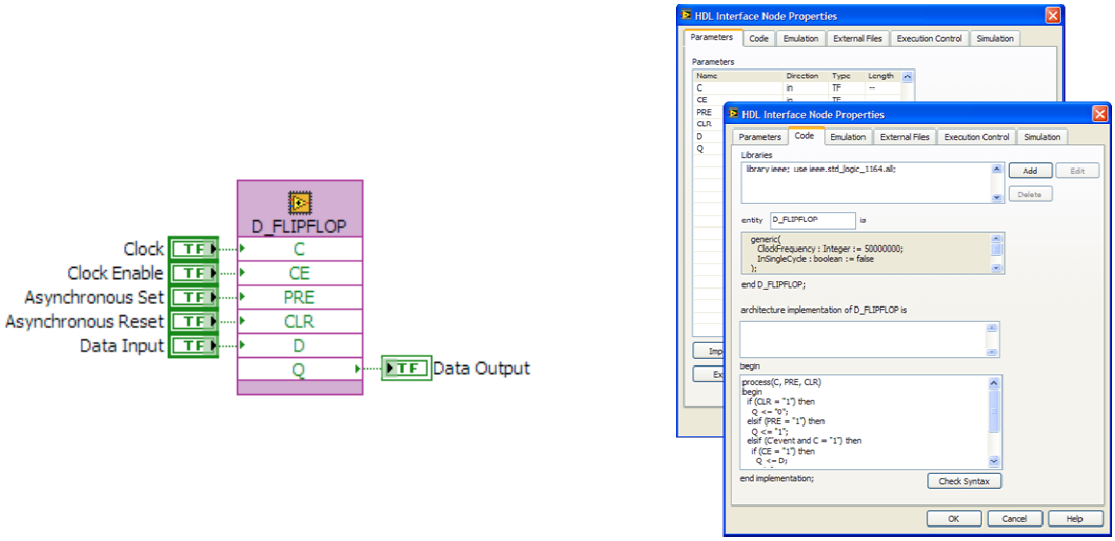


Figure 1. HDL Interface Node.

Using these techniques, each device is given full functionality. For example, the Bidirectional Counter device has a comprehensive set of controls and output signals including synchronous load, asynchronous reset, and terminal count. Although the footprint of this device is larger than a simple counter device, the students benefit more from having these I/O available and learning their functions. Additionally, most of the digital logic VI's are polymorphic which allow the student to easily change the device size and type as show in Figure 2. By using polymorphic VIs and giving full functionality to each device, students have the benefit of using a convenient library of fewer functional devices rather than an unwieldy library with an exhaustive list of devices.

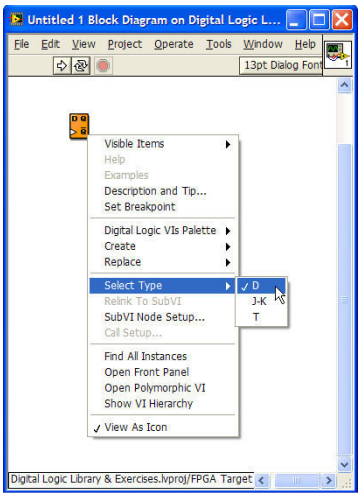


Figure 2. Polymorphic VI.

NI LabVIEW Toolkit Development

Once finished, the Digital Logic VI Library contained a complete set of logic devices necessary for developing laboratory assignments/projects for an introductory digital logic course including basic logic gates, combinatorial devices, sequential logic devices, and other general support devices. The HDL-based devices are represented as appropriate logic symbols in order to give the look and feel of a digital logic schematic while designing. As discussed previously, the implementation of all devices has been done assuming the Xilinx Spartan-3E Starter Board as the FPGA Target and the Onboard Clock (50 MHz, 50% duty cycle) as the FPGA Base Clock.

Basic Logic Gates

The basic logic gates are comprehensive and available in multiple sizes. Logic gates with less than five inputs have individual bit inputs as shown in Figure 3. Otherwise, logic gates have bus inputs and individual bits should be access with a Bus device.

Combinatorial Devices

The combinatorial devices contained in the latest version of the library include multiplexers, line decoders / demultiplexers, priority encoders, logical shifters, adders, and magnitude comparators. Each combinatorial device comes in different sizes as shown in the last column of Figure 4.

Device Name	Device Icon	Types / Sizes
Inverter	Input Output	1-Input
AND Gate	Input 0 Input 1 Output	2 to 9, 12, 16-Input

NAND Gate		2 to 9, 12, 16-Input
OR Gate		2 to 9, 12, 16-Input
NOR Gate		2 to 9, 12, 16-Input
XOR Gate		2-Input
XNOR Gate		2-Input

Figure 3. Basic Gates.

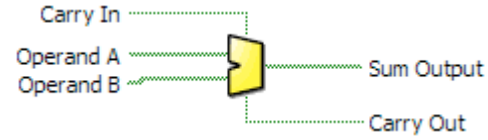
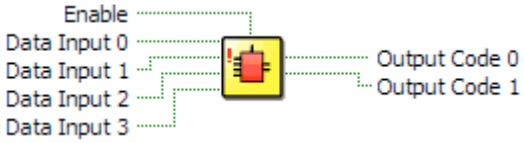
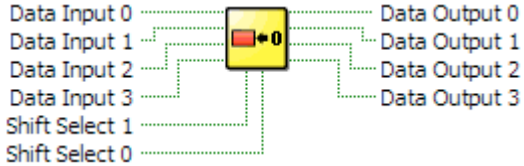
Device Name	Device Icon	Types / Sizes
Adder		4, 8, 16-Bit
Multiplexer		2:1, 4:1, 8:1 / 1-Bit
Line Decoder / Demultiplexer		2:4, 3:8, 4:16
Priority Encoder		2, 3, 4-Bit
Logical Shifter		4, 8, 16-Bit
Magnitude Comparator		4, 8, 16-Bit

Figure 4. Combinatorial Logic Devices.

Sequential Devices

The sequential devices include flip-flops, registers, and counters. Similar to the combinatorial devices, the register and counter devices come in different sizes. Additionally, the flip-flop device may be configured as a D, T, or J-K type as listed in Figure 5.

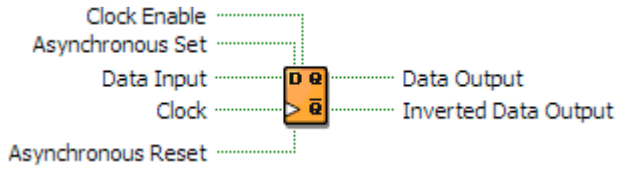
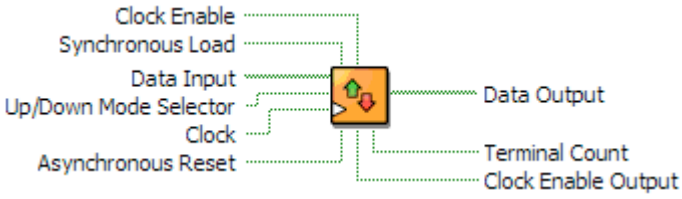
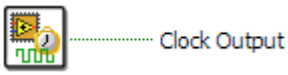
Device Name	Device Icon	Types / Sizes
Flip-Flop		D, J-K, T
Register		4, 8, 16-Bit
Bidirectional Counter		4, 8, 16-Bit

Figure 5. Sequential Logic Devices.

Support Devices

Finally, the library includes some general support devices necessary for complete functionality as shown in Figure 6. In the first revision of the library, the only general device that was necessary was the Clock device. This device provides access to the Onboard Clock and drives the sequential logic devices.

Based on student feedback, three more general devices have been created to assist in larger designs. These devices are based on Xilinx ISE's Schematic Tools. Students repeatedly access individual bits from multi-bit signals. In LabVIEW, this requires the wiring of a Bundle/Unbundle VI and Array To Cluster / Cluster To Array VI. While this is simple to do, it can become cumbersome for a larger scale design. Students will now be able to use a bus and bus tap device to access individual bits of an input and output, respectively. Likewise, a Constant device can be used to set an input to a constant value as shown in Figure 6.

Device Name	Device Icon	Types / Sizes
Clock		50 MHz @50%

		Duty Cycle
Bus		4 to 8-Bit
Bus Tap		4 to 8-Bit
Constant		4, 8, 16-Bit

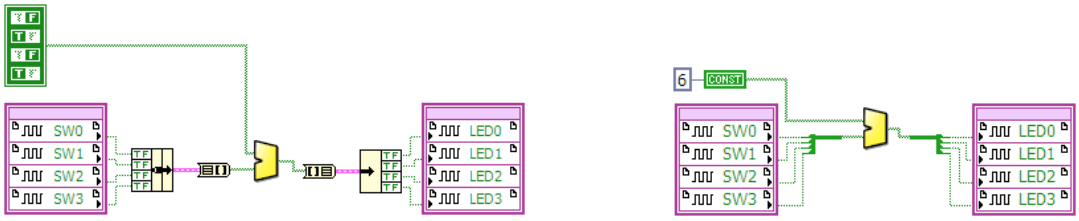


Figure 6. Support Devices.

Laboratory Assignments

Using the newly developed LabVIEW Digital Library VI toolkit, a comprehensive set of digital design laboratory assignments was created. The laboratory assignments were designed to not only reinforce the fundamental digital concepts and design principles being presented in the classroom, but also to provide the student teams with the basic building blocks necessary to successfully develop the autonomous control for a mobile platform. Digital concepts and embedded development topics are covered in a span of 11 weeks as outlined in Figure 7.

Based on student feedback, three more general devices have been created to assist in larger designs. These devices are based on Xilinx ISE's Schematic Tools. Students repeatedly access individual bits from multi-bit signals. In LabVIEW, this requires the wiring of a Bundle/Unbundle VI and Array To Cluster / Cluster To Array VI. While this is simple to do, it can become cumbersome for a larger scale design. Students will now be able to use a bus and bus tap device to access individual bits of an input and output, respectively. Likewise, a Constant device can be used to set an input to a constant value as shown in Figure 6.

Lab #	Lab Objective	Topics	Duration
1	Design a digital circuit from a truth table.	Basic Logic Gates, SoP/PoS	1 Week
2	Design a seven-segment display decoder.	Decoders, Karnaugh Maps	1 Week
3	Design a 4-bit 2:1 multiplexer.	Multiplexers, Cascading Devices	1 Week
4	Design a binary-to-BCD converter.	Adders, Magnitude Comparators	1 Week

5	Design a clock divider.	Flip-Flops, Counters	1 Week
6	Design a simple calculator.	Registers, Device Integration	1 Week
7	Design a pulse width modulation generator.	Pulse Width Modulation	2 Weeks
8	Design a quadrature decoder.	Quadrature Signals	1 Week
9	Design a counter as a state machine.	Finite State Machines	2 Weeks

Figure 7. Laboratory Assignments.

Each lab concludes with a demonstration of a fully functional design and FPGA implementation to the laboratory instructor. As the semester progresses, students develop their own library of devices with the ability to reuse them in later labs, as well as the course project.

Course Project

The culmination of the laboratory exercises is the course project (now the Xilinx Race of Champions) where a team of two students apply their digital design knowledge to develop an autonomous, line-following mobile platform using the Digital Logic VI Library. The controller must be capable of navigating a "drag race" (long and straight) as well as a "road race" (sharp turns). The lowest total time to complete both races is used to determine the winning team which then has one semester of bragging rights. The team's success is strongly determined by their lab participation, since topics from each lab are incorporated into the project. As indicated in Figure 8, students integrate the modules they have developed in the previous laboratories with the mobile platform hardware they been given and have assembled. Additionally, basic project kits based on the Digilent Robotic Starter Kit are provided; however, students have complete control over the platform mechanics and sensors and may use different parts at their discretion. Figure 9 is the winning platform from the Fall 2008 semester.

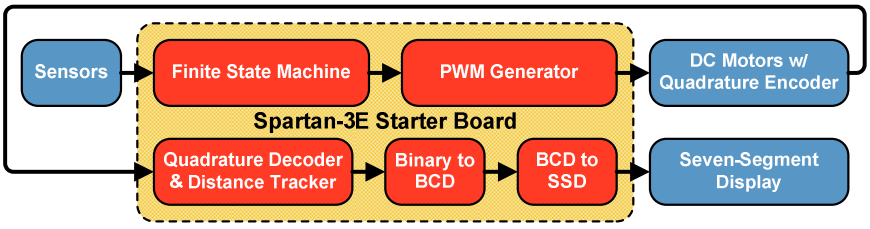


Figure 8. Block Diagram of Autonomous Platform Controller.

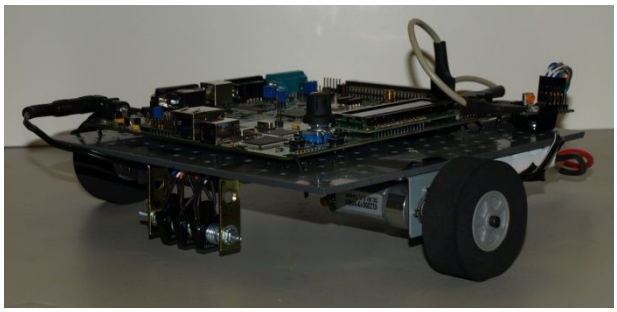


Figure 9. Spartan 3E Starter Kit and Mobile Platform.

Lessons Learned

Many lessons were learned during the initial implementation of the Digital Logic VI Library. Through student feedback and teaching experience, the advantages of the new development environment could easily be seen. Relative to the Xilinx ISE Schematic environment, students preferred dragging and dropping I/O quickly instead of assigning pins, referencing the built-in LabVIEW context help to understand devices more clearly, and creating their own reusable devices as well as their iconic representations. Towards the end of the semester, students were shown the HDL representation of some devices and realized the benefits of defining digital systems though a language interface.

In contrast, students expressed dissatisfaction with some aspects of the environment—some of which have been addressed with the creation of Support VI's. However, others are an inherent characteristic of all hardware schematic environments. For example, the disadvantage of designing large devices and complex systems is two-fold—wiring and compilation time. Wiring large schematic designs can be cumbersome but is unavoidable at times. Long compilation times can be shortened by upgrading the computers on which the environment run. Current students are encouraged to install LabVIEW on their laptops for convenient access and quick compilation times, assuming their laptops have sufficient capability.

This novel approach to the development of the Digital Logic VI Library has some limitations, since the LabVIEW FPGA Module's primary purpose is to help graphical programmers implement code on FPGAs *without* having any HDL experience. Because digital logic devices represent actual hardware, one of our goals during the development was to maintain a *schematic-like* environment. However, LabVIEW FPGA follows LabVIEW data flow which was developed with a programmer's perspective in mind. In this type of environment, a user needs software-defined hardware wrapped in an infinite loop structure to define its indefinite execution. Therefore, some form of a loop is required for the Digital Logic VI Library to function; otherwise, hardware execution runs for a single clock cycle and halts. The Single-Cycle Timed Loop as pictured in Figure 10 ensures that the logic design is implemented properly in hardware and is the recommended solution. Other loops may insert unwanted latches and flip-flops at compilation.

Figure 10. Single-Cycle Timed Loop.

Another structure implemented in LabVIEW FPGA is the Feedback Node. Feedback occurs when an output of a device is wired to the input of that same device or any device that affects it. LabVIEW automatically inserts a Feedback Node in order to stay true to LabVIEW data flow. In terms of hardware, a Feedback Node is simply a flip-flop that updates on the available clock's rising-edge. This presents a problem when using the Digital Logic VI Library—especially when designing state machines. A work around solution was created that is an extension of double data rate transfer. In the VHDL for the Clock device, we define the Clock Output as the *inverse* of the Onboard Clock. This means that all clocked devices in the library update on the *falling-edge* of the Onboard Clock and the effect of the Feedback Node is negligible for simple logic designs required in the lab course. Also, minimizing the number of Feedback Nodes that appear and placing them in consistent parts of any design is highly recommended. An example of this is shown above in Figure 11.

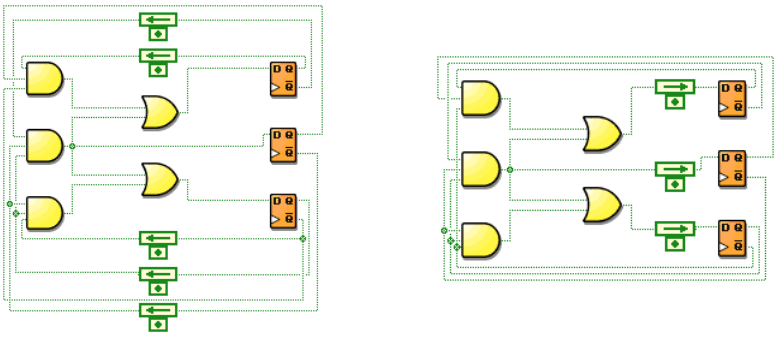


Figure 11. Minimizing Feedback Nodes.

Design Examples

To demonstrate the capability of the library, a state machine design for a 4-Bit Up-Counter is shown in Figure 12. The device has Run/Hold' and Initialize/Reset inputs and the count is displayed on discrete LEDs. A 1 Hz clock is derived from the onboard 50 MHz clock. Cascaded 4-bit counters are used instead of using larger counters to illustrate cascading devices. The design has been validated on the Spartan-3E Starter Board.

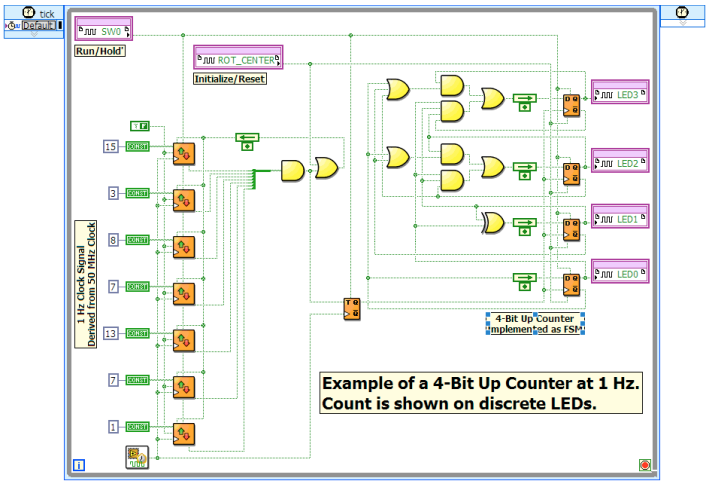


Figure 12. 4-bit Up Counter Example.

Figure 13 shows a student design for a 4-bit 8-to-1 multiplexer from cascading four 1-bit 8-to-1 multiplexers. Additionally, the student created a sub-VI from the design with an appropriate icon. Note the repetitive use of Bundling/Unbundling and Cluster-to-Array/Array-to-Cluster VIs. The new Bus/Bus Tap devices would be ideal for this design.

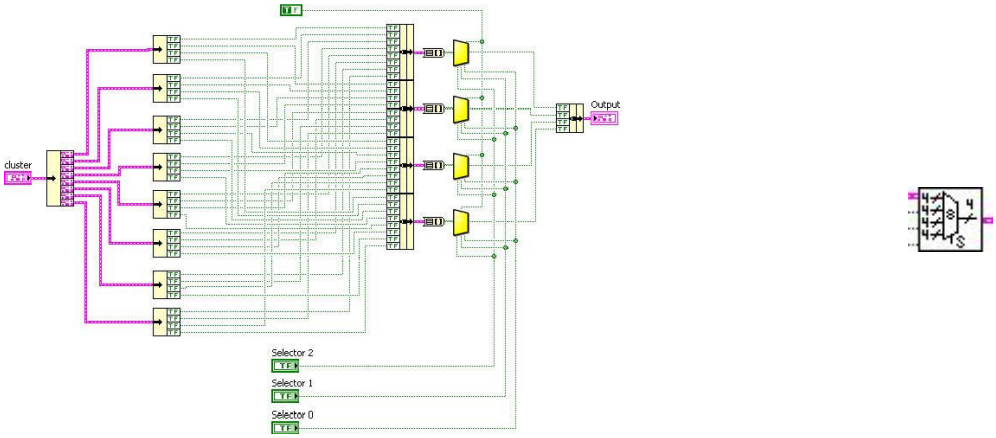


Figure 13. 4-bit 8-to-1 Multiplexer Example.

Finally, Figure 14 is a top-level view of a student team's design for the autonomous line-following platform project. The figure shows how LabVIEW supports a modular-based design. Additionally, one can see how the new Constant device would be beneficial, rather than the use of the Boolean constant arrays.

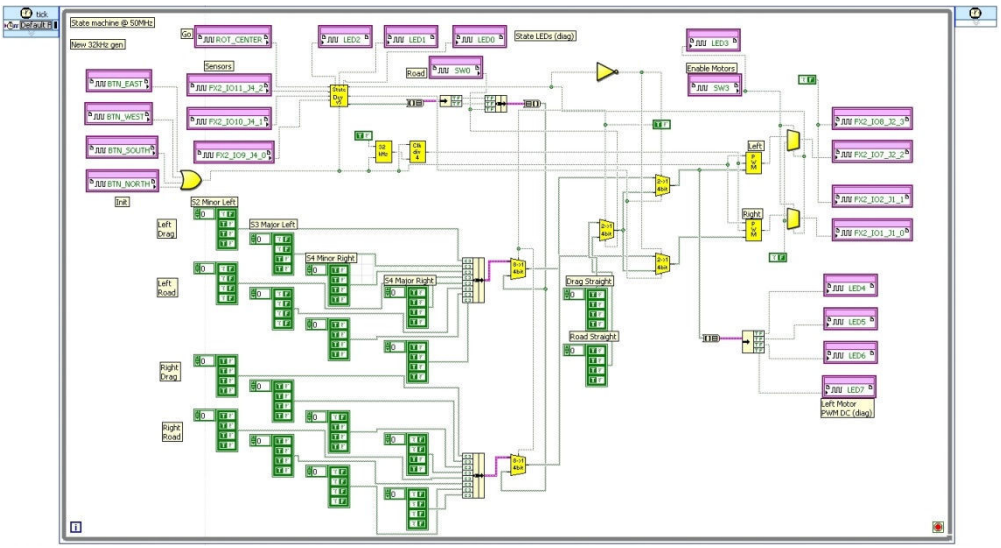


Figure 14. Top-level Design for Autonomous Platform Controller.

Follow-On Activities

While the programs have used FPGA's for several years, in particular with the more traditional graphical development tools offered by Altera and Xilinx, the capabilities offered by this LabVIEW toolkit are allowing the programs to expand on the use of programmable gate arrays throughout the curriculum. Because this toolkit creates a link between the use of discrete digital gates/devices and hardware description languages (HDL's), the students are well prepared for follow-on courses that rely on HDL to develop more complex digital systems such as digital controllers, microprocessors, and communication interfaces. For this reason, the Electronics and Telecommunication Programs are targeting two additional courses for follow-on activities related to FPGAs.

First, to build on what the students are now learning in the Introduction to Digital Logic course, the follow-on course in Advanced Digital Design is being modified. In the past, this second course has focused on the discrete design of a four-bit microcontroller with the goal of introducing students to the architecture and the programming of simple embedded systems. Starting in the Fall of 2008, the course objectives have been expanded to include:

- Learning the use of a hardware description language, specifically VHDL.
- Designing a simple microprocessor in VHDL, loading it on to an FPGA, and programming/testing the device.
- Learning to develop and work with FPGA soft cores, such as the PicoBlaze processor, and to integrate several cores together to create application-specific systems.

The expectation is that from a two course sequence in digital system design, students will be prepared to use FPGA design techniques in upper-level courses to solve complex product/system design problems.

Second, the Programs' capstone design sequence has a requirement for innovation, product/system design and implementation. Traditionally, students have used discrete microcontrollers with peripheral circuitry to create solutions for their specific design problem.

Armed with this new set of tools, the faculty expectation is that in the near future, student capstone teams will use FPGA hardware coupled with soft-core/firmware solutions as part of their product/system design.

Conclusions

In an introductory course such as digital design, students are being subjected to a significant number of new ideas and processes. In order to reinforce the classroom learning experience, the students must be able to, in a straightforward manner, transition from the blackboard design to a working circuit in a highly efficient and intuitive manner. Spending time inserting chips into protoboards and connecting a large number of wires to pins is not efficient and has a level of abstraction that reduces conceptual learning. Through the use of the new LabVIEW digital design toolkit, students can readily translate the combinatorial and sequential logic circuits discussed in class to a design in their computer. Once the design is completed, the students can quickly download the design to the FPGA-based development board for testing and debugging. This approach keeps all aspects of learning at the same level while making the design iteration and evaluation process fast and effective. Students complete the labs in a timely manner and have the theory and techniques learned in class fully reinforced in the laboratory assignment. In addition, the library allows students to self-teach HDL in preparation for follow-on courses. Finally, this new design environment allows the student teams to work in parallel to complete and test all stages of the digital design necessary to autonomously control the operation of a mobile platform. Assessing various design approaches as well as 'tweaking" the selected design to optimize performance is fully supported using the NI LabVIEW toolkit developed by the EET/TET Programs at Texas A&M University and the Xilinx development board. Student feedback indicates that the learning curve is manageable, and the value-add to understanding the basic concepts of digital design is significant. Minor adjustments and additions to the toolkit were made following the student feedback from the Fall 2008 semester. The Toolkit is now available for download and use by other educational institutions from the NI web site.

Bibliography

1. M. Radu, M. Alexandru Dabacan, S. Sexton , C. Cole, "Extensive Use of Advanced **FPGA** Technology in Digital Design Education," 2008 Annual Conference & Exposition, Pittsburgh, PA, 2008.

2. N. Alaraje, J. DeGroat, A. Oliveira, "Teaching Field Programmable Gate Array Design (**FPGA**) to Future Electrical Engineering Technologists: Course Development," 2007 Annual Conference & Exposition, Honolulu, HA, 2007.

3. http://www.xilinx.com

4. http://www.digilentinc.com

5. http://www.ni.com/fpga/

6. G. Crouch, R. Lee , B. Johnson , S. Sharad, "Fostering Design Across Multiple Disciplines With Graphical Programming and FPGAS," 2008 Annual Conference & Exposition, Pittsburgh, PA, 2008.