Learning Objectives:

Project

GS

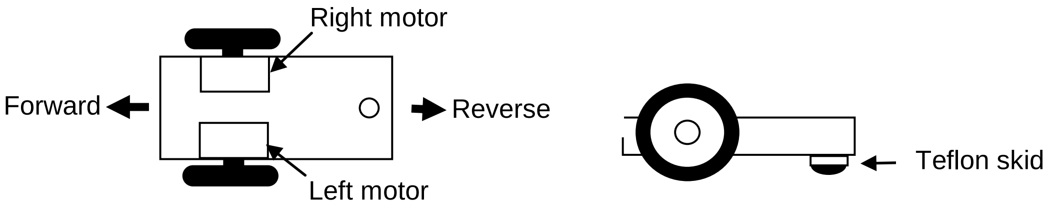
- Review logic simulation using *ModelSim*
- Practice coding Verilog for synthesis
- Introduce logic synthesis using Xilinx *XST*
- Introduce Digilent Nexys™ 3 development board
- Learn how to configure FPGA using Xilinx *iMPACT*

This project is provided to help you come up to speed on Xilinx ISE, Synthesizable Verilog and the Digilent Nexys™ 3 board

You do not need to demonstrate or submit deliverables for this project and it will not be graded

Project: RojoBot

This project models a very simple robot. The robot is a platform with two wheels, each driven by an independent motor. A third free rotating wheel or Teflon skid serves to stabilize the platform.



For this course we won't use a physical robot, instead we'll model the robot's operation using the Digilent Nexys 3 development board. In this Getting Started project we will use the Nexys 3 board's four pushbutton switches to control our virtual robot's wheel motors, and the 4-digit seven segment display to show information about the robot's motion.

You are given a working reference design to start with. It provides a design framework and example to help you build your own design. Use the reference design to verify the proper operation of your design software and Nexys 3 board. In the reference design the left and right wheel counter values are displayed on the Nexys 3 board's seven segment display. There is a decimal point ('.') between digits 2 and 1 to divide the left wheel counter displayed on digits 3-2 from the right wheel counter displayed on digits 1-0.

Included in the reference design are Verilog modules that can be used to debounce the pushbuttons and switches and drive the display.

Your task for this project is to design and implement two new motion indicators for the display: One digit will indicate Stop/Forward/Reverse/Right-Turn/Left-Turn; the other three digits will indicate the robot's compass heading. The object of this project is to introduce the logic synthesis software and FPGA hardware that will be used during the course.

Functional Specification of Bot

PUSHBUTTON MOTOR CONTROL

The four pushbutton switches control two wheel motors as specified in Table 1:

Table 1: Pushbutton Motor Control

Pushbutton	Motor Function
BTN_LEFT	Left Forward
BTN_UP	Left Reverse
BTN_RIGHT	Right Forward
BTN_DOWN	Right Reverse

If neither of the two buttons that control each motor is pushed, then that motor is stopped. If *both* of the two buttons that control each motor are pushed, or if all 4 buttons are pushed at the same time the actions cancel each other leaving the motor stopped.

MOTION MODES

The Bot's independently controlled wheels enable it to move forward, backward, turn left or turn right, as shown in Table 2.

Table 2: Robot Motions

Left Motor	Right Motor	Robot motion mode
Stop	Stop	Stop
Forward	Stop	Turn Right 1X Speed
Stop	Reverse	Turn Right 1X Speed
Forward	Reverse	Turn Right 2X Speed
Stop	Forward	Turn Left 1X Speed
Reverse	Stop	Turn Left 1X Speed
Reverse	Forward	Turn Left 2X Speed
Forward	Forward	Forward
Reverse	Reverse	Reverse

Note that the robot can turn at two speeds in either direction depending on whether both or only one wheel is moving during the turn.

MOTION INDICATION FUNCTIONS

The reference design provided with the project represents the motion of each wheel with an 8-bit up/down counter. The counters count up at 5 Hz when the wheel is moving forward, and count down at 5 Hz when the wheel is moving in reverse. The left wheel counter is displayed on the two leftmost digits of the four-digit display and the right wheel counter is displayed on the rightmost two digits of the display.

You will change the functionality from the reference design for this project. Instead of merely displaying the wheel count, use the leftmost digit to implement a new motion indicator and the three rightmost digits to implement a digital compass that continuously displays the robot's heading. These two functions are specified in more detail below.

MOTION INDICATOR

This indictor uses a single seven-segment digit to provide an animated display of the robot's current motion mode, as defined in Table 2. Table 3 defines the digit's action for each of the modes.

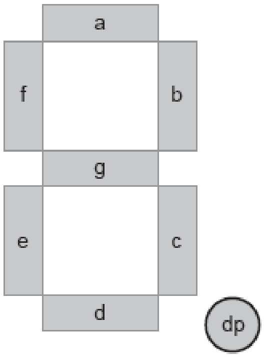


Figure 1: Seven-Segment Digit

Table 3: Motion Indicator

Robot motion mode	Display Action
Stop	Segment **g** on **steady**
Turn Right 1X Speed	Chase clockwise 5Hz
Turn Right 2X Speed	Chase clockwise 10Hz
Turn Left 1X Speed	Chase counterclockwise 5Hz
Turn Left 2X Speed	Chase counterclockwise 10Hz
Forward	Segment **a** blinks at about **1Hz**
Reverse	Segment **d** blinks at about **1Hz**

"Chase" means that one lit segment moves around the perimeter of the display digit. Only one segment is lit at a time. The lit segment advances one position at the rate of 5Hz or 10Hz depending on the turning speed. For a clockwise chase the segments are lit in the following sequence: a, b, c, d, e, f, a… For a counterclockwise chase the sequence is reversed.

COMPASS

This indictor uses the three rightmost seven-segment digits to provide a display of the robot's current heading from 0 to 359_{10} degrees. The compass heading is computed by dead reckoning, that is, by accumulating the robot's motions over time. When the robot is stopped or moving in a straight line (either forward or reverse) the compass heading does not change. The compass heading only changes when the robot is turning. The rate of compass change will, of course, depend on the turning rate. The compass can be implemented with an up/down counter. Table 4 defines the actions of the compass counter as a function of the motion mode.

Table 4: Compass Specification

Robot motion mode	Compass Action
Stop	Hold
Forward	Hold
Reverse	Hold
Turn Right 1X Speed	Increment at 5Hz
Turn Right 2X Speed	Increment at 10Hz
Turn Left 1X Speed	Decrement at 5Hz
Turn Left 2X Speed	Decrement at 10Hz

Bot Design Notes

The reference design is implemented using two 8-bit up/down counters, as shown in the block diagram in Figure 2. Each block (module) receives the global 100MHz clock generated by the oscillator on the Nexys 3 board and divides it down locally to make the slower timing signals that it needs.

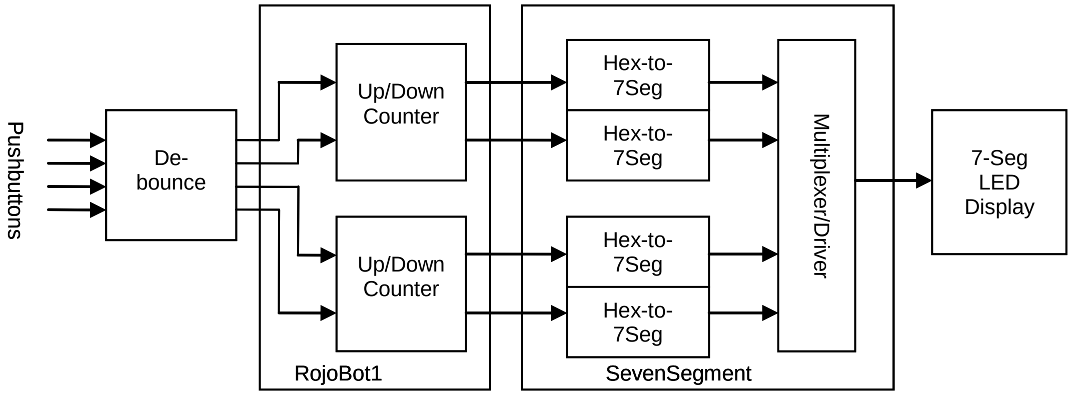


Figure 2: Reference Design Block Diagram

A block diagram for your new design with the Compass and Motion Indicator is shown in Figure 3. The debounce module and the display driver can be reused. Both the Motion Indicator and Compass modules use the robot's motion mode as inputs. The motion mode can be decoded from the debounced pushbutton inputs. The motion indicator (MI) can be implemented with a finite state machine (FSM). The MI output is not hexadecimal but there are special codes that cause each of the individual segments of a digit to display. Table 5 lists the display codes that are recognized.

Table 5: Display Character Codes

Code (binary)	Displays (ASCII code)
0 - 9	Characters 0 to 9
10 – 15	Upper case characters A to F
16 – 22	Single Segments a to g
23	Space (Blank)
24*	Upper case character H
25*	Upper case character L
26*	Upper case character R
27*	Lower case character L (l)
28*	Lower case character R (r)
29 – 31	Space (blank)

* Special characters used in the future project

The compass can be implemented with a 0-359 decimal up/down counter. There is, of course, additional control logic not described in detail here; rather its implementation is left as an exercise for the student.

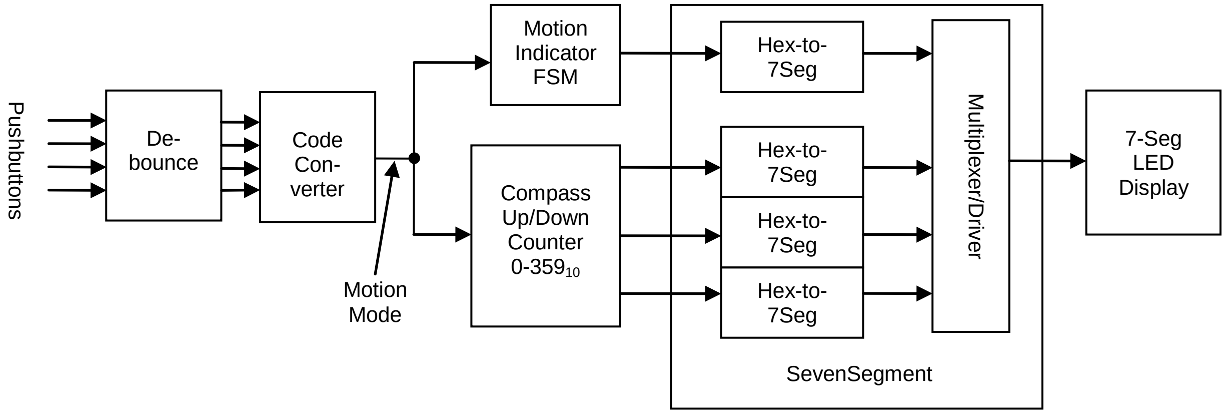


Figure 3: RojoBot with Compass and Motion Indicator Block Diagram

Hardware

The projects for this course will be implemented in a Xilinx Spartan 6 FPGA on the Nexys 3 development board. In addition to the FPGA, the board provides a variety of input and output devices. This project uses the pushbuttons for input and the seven segment display for output. Other I/O devices on the board will be used in later projects.

Design Software

The projects for this course will use electronic design automation (EDA) software tools from Mentor Graphics, Xilinx, and Digilent. Mentor *Graphics ModelSim PE Student Edition* digital simulator can be downloaded free from http://www.mentor.com/company/higher_ed/modelsim-student-edition.

Xilinx *ISE WebPACK* includes all the necessary Xilinx software tools, and can be downloaded free from http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools.html. You may have to download the entire ISE Design Suite but the only Xilinx tool needed in this course is the ISE Webpack. You may need a license, even to run the Webpack, but it is available for no charge.

ModelSim PE

You may be familiar with Mentor Graphics' ModelSim logic simulator from previous courses or from on-the-job usage. If you have not used ModelSim before we recommend doing the ModelSim Tutorial (From ModelSim's Help menu bar) to come up to speed.

NOTE: Xilinx ISE includes a logic simulator which may be used instead of ModelSim. However, it should be noted that the instructors are more familiar with ModelSim and may not be able to answer questions about the ISE simulator.

Xilinx ISE FPGA place & route software

Xilinx Integrated Software Environment (ISE) tools include XST, a state-of-the-art FPGA logic synthesis tool that synthesizes Register-Transfer Level (RTL) Verilog HDL into a network of FPGA primitives: LUTs and flip-flops, a Place & Route tool takes the netlist produced by synthesis and places the primitive cells in the FPGA, then routes the wires using programmable interconnect. The resulting implementation is then converted into a *bitstream* (.bit) file specifying the configuration of all programmable elements within the FPGA.

Bitstream download software

Xilinx iMPACT is used to download the bitstream file from a PC's USB port to the FPGA on the development board. iMPACT is included in the ISE WebPACK and can be launched directly from the Xilinx ISE Project Navigator or directly from Windows using the START menu shortcuts to the Xilinx ISE tools. iMPACT can also be used to create and download bit images to the configuration EEPROMs on the Nexys 3 development board. Be sure to include the cable drivers when ISE is installed.

Project Tasks

Download the reference design

Download *ece540gs_release.zip* from the course website. This .zip file contains the reference design that you can use to get started and examples for using the peripheral modules provided for the course. The zip file contains the following files:

Project 1 Reference Design	
Name	**Description**
`ece540gs_refdsgn\hdl\tb.v`	A testbench that instantiates `Nexys3fpga`. It provides clock signals that model the oscillator on the Nexys3 board and a

	7-segment display emulator.
ece540gs_refdsgn\hdl\nexys3fpga.v	Top level of the Nexys 3 FPGA. Includes all the ports needed and some utility logic. Instantiates bot1, debounce and SevenSegment.
ece540gs__refdsgn\hdl\bot1.v	A reference design using two up/down counters to indicate the position of the robot's two wheels.
ece540gs__refdsgn\sim\wave.do	Waveform setup file for ModelSim. This file can be used to display and format useful signals.
Common Files	
common\hdl\debounce.v	Debounces pushbutton and slide switch inputs.
common\hdl\sevensegment.v	Provides the drivers needed for the seven segment display.
common\synth\nexys3fpga.ucf	User constraints file. Contains pin assignment information so that the fpga's top level module ports are connected to the proper FPGA device physical pins. This file comments out the unused pins and can be modified for other projects as the pinout changes.
Documentation	
docs\ece540_GettingStarted.pdf	This document

Simulate the reference design

Simulate the reference design using ModelSim (files `tb.v Nexys3fpga.v, bot1.v and debounce.v`) or the ISE logic simulator. Learn the hierarchical design structure of the FPGA and test bench Verilog models. Observe the waves display to see how the design operates.

Synthesize the reference design

Learn about the synthesis process and Xilinx place and route tools.

- Start ISE (*Start → Programs → Xilinx Design Tools→ Xilinx ISE Design Suite 14.7→ ISE Design Tools → Project Navigator*). For more information about using ISE click *Help → Software Manuals*. If you installed 64-bit ISE you will have to select either the 32-bit or 64-bit Project Navigator to match your installation.
- Click *New Project* in the "Welcome to the ISE® Design Suite" pane. In the "New Project Wizard" window that pops up, enter "ece540gs" or a name of your choice for "Project Name", then enter the path to the directory that holds the downloaded Verilog files for "Project Location". Make sure the "Top-level source type is set to HDL. You can also enter a description if you'd like. Click "Next".
- Enter the following "Device Properties" into the wizard. The "Device" and "Package" values must match the FPGA chip on your board.

Evaluation Development Board:	None Specified
Product Category:	General Purpose
Family:	Spartan6
Device:	XC6SLX16
Package:	CS324
Speed:	-3
Synthesis Tool:	XST (VHDL/Verilog)
Simulator:	ModelSim-PE Mixed (or whichever simulator you

	are planning to use)
Preferred Language:	Verilog
Property Specification…	Store all values
Manual Compile Order	Unchecked
Enable Message Filtering	Checked

Click *Next* and then *Finish*.

- Click the “Add Source” button on the left or select Project→Add Source from the menu bar
- Select all of the .v files in the ece540gs_refdsgn\hdl directory except tb.v and all of the .v files in the common\hdl directory. The file tb.v is the test bench for simulation and is not part of the design that you are synthesizing.
- Click the “Add Source” button on the left or select Project→Add Source from the menu bar again
- Navigate to common\synth directory and add nexys3fpga.ucf to the project. nexys3fpga.ucf contains pin assignment information; it tells the synthesis tool which ports of nexys3fpga.v are connected to which physical pins of the FPGA device.
- In ISE’s “Sources” pane (upper left) click on “Nexys3fpga” to highlight it. Note that the contents of the “Processes” pane changes depending on which source is highlighted in the Sources pane.
- Now you’re ready to compile the design. Double-click “Synthesize-XST” in the Processes pane. Watch the “Console” pane at the bottom of the ISE window and view the log file. This shows a blow-by-blow log of each step in the compile process. Observe how the finite state machines in the design have been extracted. Keep an eye out for unexpected errors or warning messages in this log. NOTE: There will be a number of messages, including a few unconnected node messages. These are generally OK but you should check them to make sure they are not important. When I ran the reference design ISE reported 29 warnings.
- Click on “Synthesis Report” in the “Detailed Reports” section of the Design Overview Summary pane. Read the report that appears.
- Next click “Synthesize/View RTL Schematic” and view the schematic. The schematic is hierarchical and each level may have multiple pages. You can traverse the hierarchy by right-clicking in different places in the schematic and selecting the desired action from the menu that pops up. Note that the design has been mapped to generic flip-flops and gates, but not yet to FPGA primitive cells. Things to think about: In what ways is schematic hierarchy similar to the Verilog hierarchy? In what ways does it differ?
- Next, click “View Technology Schematic” and view the schematic of the design mapped to FPGA logic primitives. Experiment with the schematic viewer. You may need it for debugging your design.
- The next step is to place and route the design. In the Processes Pane double-click “Implement Design”. Then watch the implementation progress in the Console pane. Note that the constraints file is used during this step.
- Finish the design by double-clicking “Generate Programming File” in the Processes pane.

Download the reference design to the development board

After the programming file has been generated, double-click on "Configure Device (iMPACT)" icon the Processes pane to start the iMPACT utility that will download the programming file to the FPGA on the Nexys 3 board.

Make sure the Nexys 3 board is configured correctly:.

- ♦ Connect the USB cable from the PC to the USB PROG connector on the Nexys 3 board. The board can be powered through the USB cable. The optional power jack is not needed.
- ♦ Turn on the power to the Nexys 3 board (SW8 on). If the power does not come on make sure the Power jumper (located immediately to the right of the external power jack) is configured for USB power.

- o Double click on "Configure Target Device". This starts the iMPACT application. Click "OK" if you get a warning "No iMPACT project file exists…"
- o When iMPACT starts open a new project file (File → New Project) , select "create a new project (.ipf) ", browse to your project directory, enter a file name (or use the default name) and Click *OK*
- o In the next screen select "Configure devices using Boundary-Scan (JTAG)" and "Automaticaly connect to a cable and identify Boundary-Scan chain". click *OK*
- o You may get a dialog box saying "Do you want to continue and assign configuration files" click on Yes. You may want to set the preference to do this automatically.
- o If the cable is connected and the correct cable drivers are loaded, iMPACT will walk you through assigning configuration files. Click on "nexys3fpga.bit" for xc6slx16, and "Open". You may get the "Attach SPI or BPI PROM" dialog box. If so, click on "No".
- o If a window pops up warning that the "Startup Clock has been changed..." Ignore it - just click "OK"
- o iMPACT should now show a window showing the xc6slx16 FPGA icon. Below the xc6slx16 should be the programming file name, nexys3fpga.bit.
- o Right Click on xc6slx16 and select *Program (or double click the Program menu item on the left)*, then click *OK* in the "Programming Properties" window that pops up.
- o Programming will take a few seconds, then you should see the message "Program Succeeded" in a blue banner.
- o The FPGA on the Nexys 3 board has now been programmed with the reference design. Press any of the pushbuttons and watch what happens.

Design and simulate your Compass and Motion indicator

- o Code your new modules.
- o Modify `nexys3fpga.v` to instantiate your modules and connect them to clock and other signals. You may need to make other changes in `nexys3fpga.v`.
- o Change `tb.v` as needed to verify your design through simulation.
- o (Optional) Simulate your new design with ModelSim making the necessary changes to `tb.v`.

Synthesize your design using ISE

You'll have to add your new files to the project. Watch the log for synthesis errors and warnings.

Download your design to the board and test it

Verify that the indicators operate as specified in the functional specification. Verify that all buttons perform as specified.

<Finis>

References

[1] Nexys3 Board Reference Manual, April 2013.
[2] *Xilinx ISE Software Manuals*