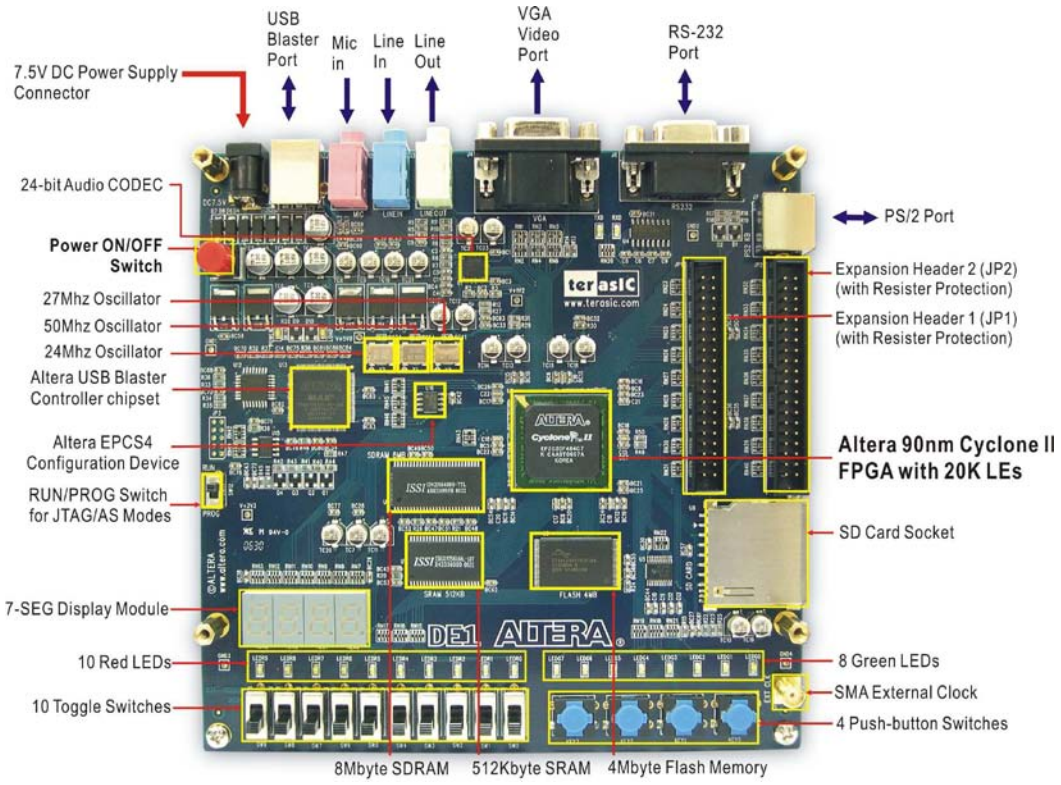
ALTERA QUARTUS II PROGRAMMING GUIDE
EE334

Learning digital logic can be difficult using just chalkboards. Sometimes a little hands-on training can bridge the gap between learning and understanding. For the next few PEs we will be designing digital logic circuits using the **Cyclone II FPGA Starter Development Boards (DE-1)** and the **Quartus II Software**.

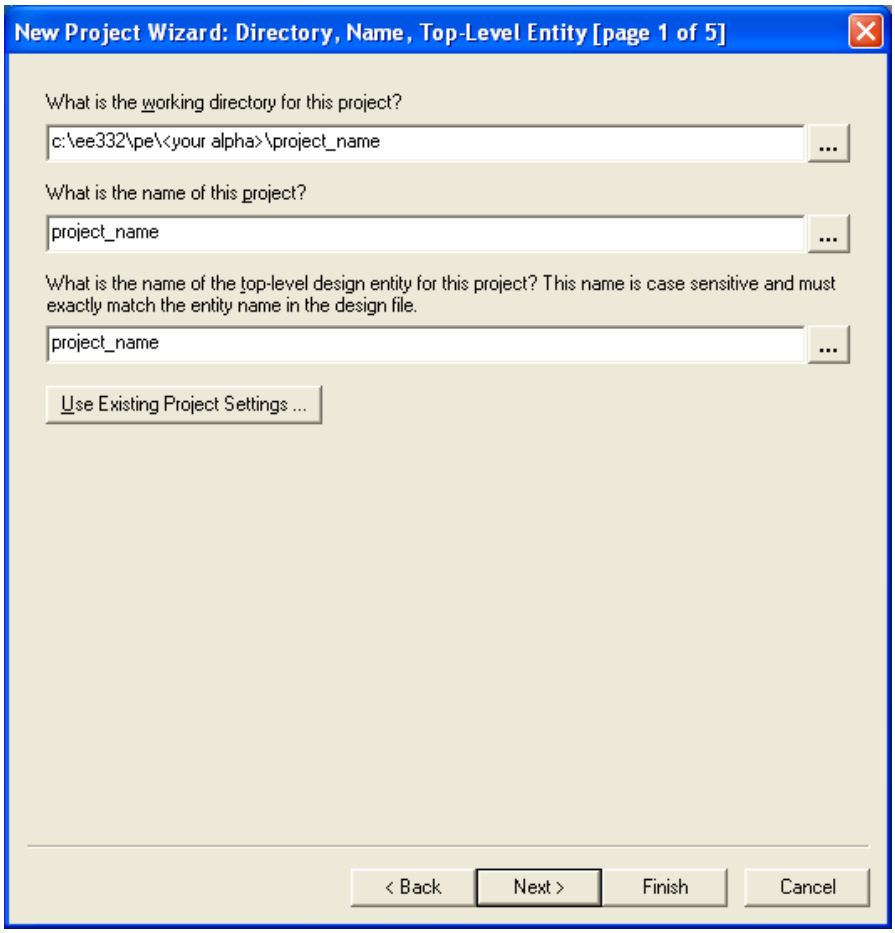
The heart of the Altera DE-1 Development Board is the Field Programmable Gate Array (FPGA). The FPGA consists of thousands of clocked logic blocks and communication lines. Each logic block can perform a variety of logic operations (AND, OR, NOT, NAND, etc). The logic blocks can be linked together via the communication lines. The communication lines also link logic blocks to external pins on the chip. By setting the functions of the logic blocks and configuring the connections of the communication lines, the FPGA can be programmed to perform an almost unlimited number of tasks. The remainder of the DE-1 board consists of helper circuits, peripheral devices, and I/O ports all of which can be controlled by the FPGA chip as the image below shows.

The Quartus II software enables engineers to design and simulate digital circuits. Once the design is complete, the Quartus II can load the design onto DE-1 board where it can actually operate on real hardware at near custom circuit speeds. Altogether, the DE-1 board and the Quartus II software provides a flexible system for designing and prototyping digital devices.

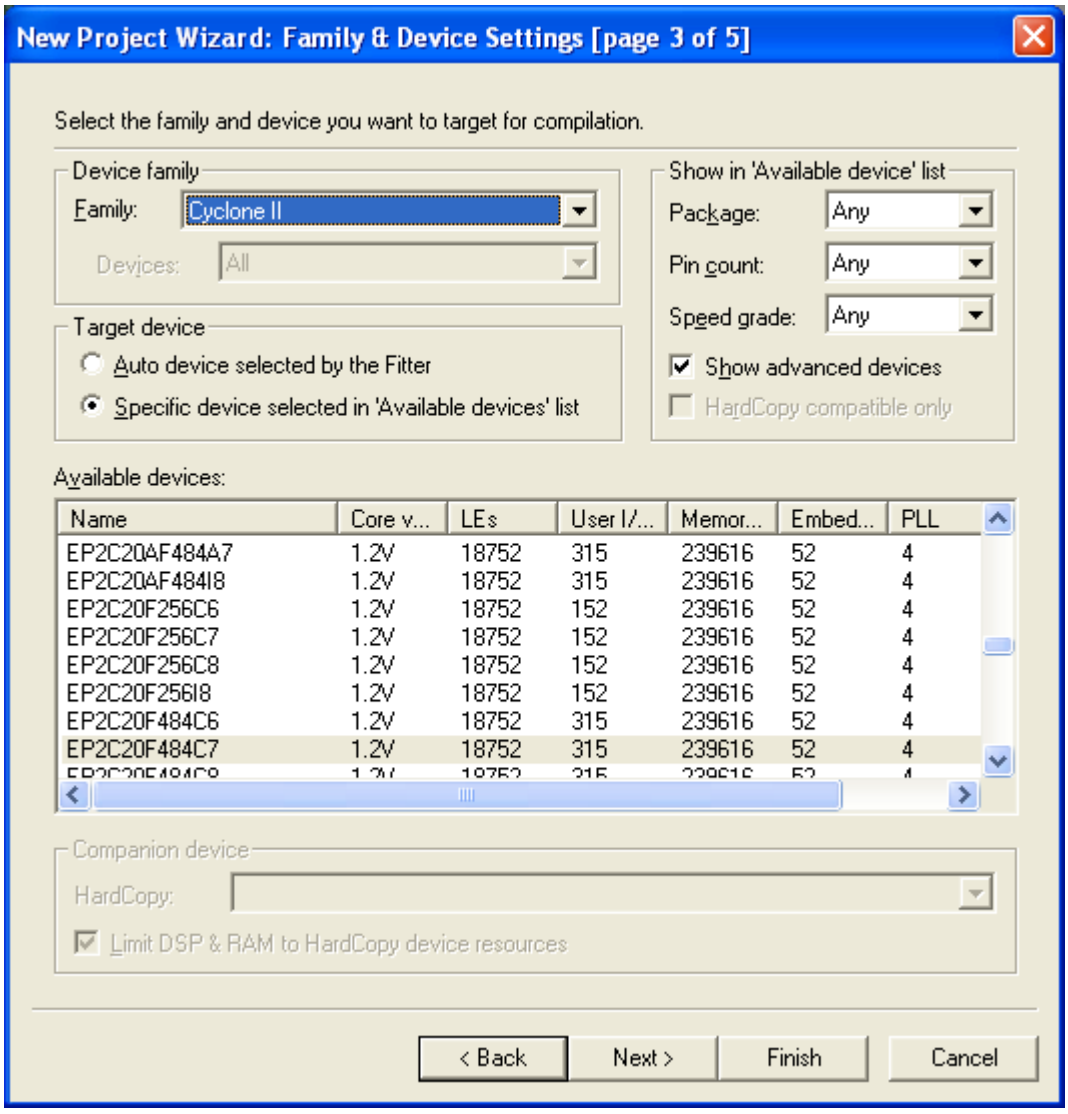


I. Setting up a New Project in Altera Quartus II

1. On the Getting Started With Quartus II Software Window, select **Create a New Project**.
2. If no window appears, select File>> New... then select New Quatrus II Project
3. Click **Next** on the introduction screen.
4. Enter the directory name and project name as shown below replacing <your alpha> with your alpha code and project_name with the lab you are working on.

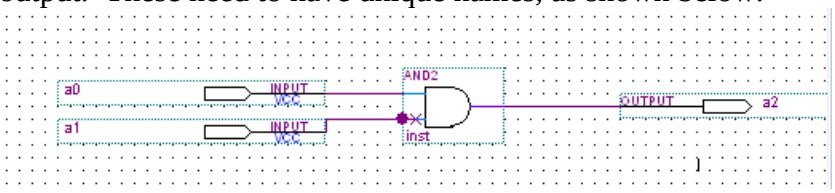


5. Click **Next**. Click **Yes** to make the directory.
6. Click Next.
7. Under “Device family” select Cyclone II. Under “Available devices” select **EP2C20F484C7** (See figure below). Click **Next**.
8. Click **Next**.
9. Click **Finish**.


New Project Wizard: Family & Device Settings [page 3 of 5]
Select the family and device you want to target for compilation.
Device family
Family: Cyclone II
Devices: All
Target device
Auto device selected by the Fitter
Specific device selected in 'Available devices' list
Show in 'Available device' list
Package: Any
Pin count: Any
Speed grade: Any
Show advanced devices
HardCopy compatible only
Available devices:
Name Core v... LEs User I/... Memor... Embed... PLL
EP2C20AF484A7 1.2V 18752 315 239616 52 4
EP2C20AF484I8 1.2V 18752 315 239616 52 4
EP2C20F256C6 1.2V 18752 152 239616 52 4
EP2C20F256C7 1.2V 18752 152 239616 52 4
EP2C20F256C8 1.2V 18752 152 239616 52 4
EP2C20F256I8 1.2V 18752 152 239616 52 4
EP2C20F484C6 1.2V 18752 315 239616 52 4
EP2C20F484C7 1.2V 18752 315 239616 52 4
Companion device
HardCopy:
Limit DSP & RAM to HardCopy device resources
< Back
Next >
Finish
Cancel

II. Drawing a block Diagram.

1. Add a new device schematic to the Project.
    a. Select **File>> New…**
    b. Under **Design Files** select **Block Diagram/Schematic File**.
    c. Click **OK**
2. Add parts to the schematic.
    a. Double click on the schematic.
    b. The directory c:/altera/81/quartus/libraries/ contains all the parts for the projects you will be working on. A table of parts you're likely to use in EE334 are in Table 1 below. **Note:** most of the parts you will need can be found in the primitives/pin and primitives/logic folders.
    c. Select the part you want and click **OK.**
    d. **NOTE** you will need to define input pins for each input and output pins for each output. These need to have unique names, as shown below.



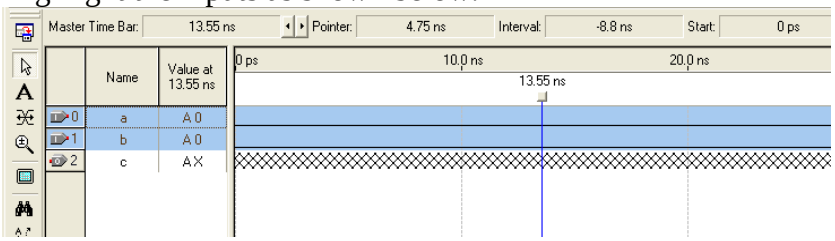
3. Connecting parts
    a. Connect parts by using the orthogonal node tool  .
4. Repeat Steps 3 and 4 until circuit is complete.
5. Save the project.
    a. Select **File>>Save Project**.
    b. Click **Save**.
6. Compile the Project.
    a. Press **Ctrl+L** or Select **Processing>>Start Compilation**.
    b. Correct any errors that are indicated and recompile.

Table 1. Common diagram parts and their Quartus names.

Device/Gate	Text Entry	Device/Gate	Text Entry
Input	`input`	JK FlipFlop	`jkff`
Output	`output`	D FlipFlop	`dff`
2 Input AND	`and2`	T Flip Flop	`tff`
3 Input AND	`and3`	Freq Divider	`freqdiv`
2 Input OR	`or2`	Nor Latch	`norltch`
Inverter	`not`	Ground (low)	`gnd`
XOR	`xor`	Vcc (high)	`vcc`
2 Input NOR	`nor2`		

III. Simulating the project

1. Open the Simulator Tool panel.
    a. Select **Processing → Simulator Tool**.
2. Identify which inputs and outputs are to be simulated.
    a. With the Simulator Tool pane open, click the **Open** button.
    b. Select Edit>>Insert>>Insert Node or Bus…
    (at the top of the screen, or right click)
    c. Click the **Node Finder** Button.
    d. Under **Filter** select **Pins: all**.
    e. Click **List**.
    f. Click "**>>**" to move the pins to the right text box.
    g. Click **OK**.
    h. Click **OK**.
3. Set up the pattern for the input signals
    a. Highlight the inputs as shown below.



    b. Select **Edit>>Grouping>>Group…**
    (at the top of the screen, or right click)
    c. Type in a group name and click **OK**.
    d. Highlight the input group you created and click the **Count** button.
    e. Click **OK**.
    f. Save the waveform file.
    g. You can close the waveform window now (not required).
4. Prepare for a functional simulation.
    a. Bring the Simulator Tool pane to the front.
    b. Set *Simulator Mode* to **Functional**.
    c. Click the **Generate Functional Simulation Netlist** button.
5. Simulate the Project.
    a. Press Ctrl+I or click on the **Start** button on the Simulator Tool pane.
    b. Once the simulation is complete, Click the **Report** button.

IV. Pin Assignments

Before you can load your design onto the Altera DE-1 board, you need to assign your design's inputs and outputs to physical connections (pins) on the FPGA chip. In turn, the pins connect to switches, lights, and other input/output devices on the DE-1 board. Thus, the pin assignments connect your design to the board so that you can interact with your design and vice-versa.

1. Press **Ctrl+Shift+N** or Select **Assignments>>Pin Planner**.
2. On the bottom of the screen, under Location, type in the desired pin for each input and output. The desired pin can be determined from tables 1 though 4.
3. Select **File>>Close.**
4. Compile the Project.
    a. Press **Ctrl+L** or Select **Processing>>Start Compilation**.
    b. Correct any errors that are indicated and recompile.

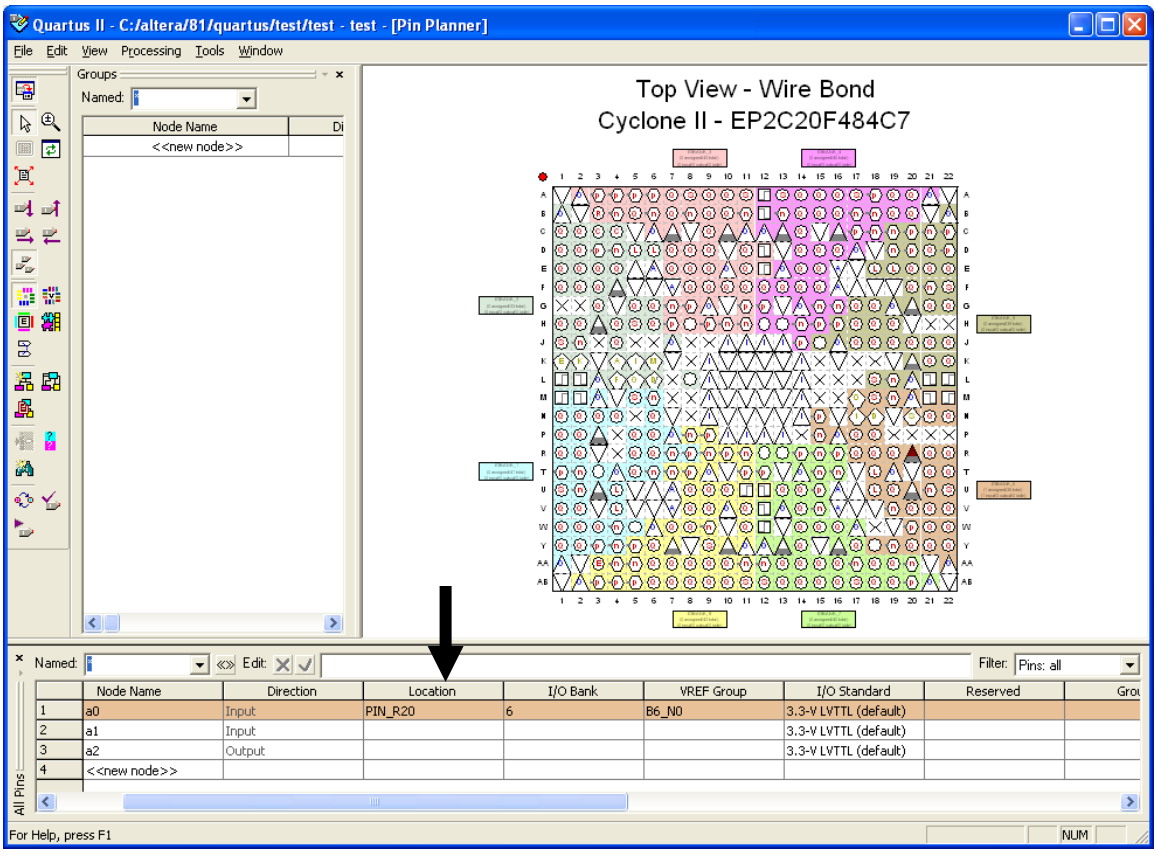


Table 1 Input: Toggle Switches

Signal Name	FPGA Pin No.	Description
SW[0]	PIN_L22	Toggle Switch[0]
SW[1]	PIN_L21	Toggle Switch[1]
SW[2]	PIN_M22	Toggle Switch[2]
SW[3]	PIN_V12	Toggle Switch[3]
SW[4]	PIN_W12	Toggle Switch[4]
SW[5]	PIN_U12	Toggle Switch[5]
SW[6]	PIN_U11	Toggle Switch[6]
SW[7]	PIN_M2	Toggle Switch[7]
SW[8]	PIN_M1	Toggle Switch[8]
SW[9]	PIN_L2	Toggle Switch[9]

Table 2 Input: Push Buttons Switches

Signal Name	FPGA Pin No.	Description
KEY[0]	PIN_R22	Pushbutton[0]
KEY[1]	PIN_R21	Pushbutton[1]
KEY[2]	PIN_T22	Pushbutton[2]
KEY[3]	PIN_T21	Pushbutton[3]

Table 3 Output: LEDS

Signal Name	FPGA Pin No.	Description
LEDR[0]	PIN_R20	LED Red[0]
LEDR[1]	PIN_R19	LED Red[1]
LEDR[2]	PIN_U19	LED Red[2]
LEDR[3]	PIN_Y19	LED Red[3]
LEDR[4]	PIN_T18	LED Red[4]
LEDR[5]	PIN_V19	LED Red[5]
LEDR[6]	PIN_Y18	LED Red[6]
LEDR[7]	PIN_U18	LED Red[7]
LEDR[8]	PIN_R18	LED Red[8]
LEDR[9]	PIN_R17	LED Red[9]
LEDG[0]	PIN_U22	LED Green[0]
LEDG[1]	PIN_U21	LED Green[1]
LEDG[2]	PIN_V22	LED Green[2]
LEDG[3]	PIN_V21	LED Green[3]
LEDG[4]	PIN_W22	LED Green[4]
LEDG[5]	PIN_W21	LED Green[5]
LEDG[6]	PIN_Y22	LED Green[6]
LEDG[7]	PIN_Y21	LED Green[7]

Table 4 Output: 7-Segment Displays for Figure 1

Signal Name	FPGA Pin No.	Description
HEX0[0]	PIN_J2	Seven Segment Digit 0[0]
HEX0[1]	PIN_J1	Seven Segment Digit 0[1]
HEX0[2]	PIN_H2	Seven Segment Digit 0[2]
HEX0[3]	PIN_H1	Seven Segment Digit 0[3]
HEX0[4]	PIN_F2	Seven Segment Digit 0[4]
HEX0[5]	PIN_F1	Seven Segment Digit 0[5]
HEX0[6]	PIN_E2	Seven Segment Digit 0[6]
HEX1[0]	PIN_E1	Seven Segment Digit 1[0]
HEX1[1]	PIN_H6	Seven Segment Digit 1[1]
HEX1[2]	PIN_H5	Seven Segment Digit 1[2]
HEX1[3]	PIN_H4	Seven Segment Digit 1[3]
HEX1[4]	PIN_G3	Seven Segment Digit 1[4]
HEX1[5]	PIN_D2	Seven Segment Digit 1[5]
HEX1[6]	PIN_D1	Seven Segment Digit 1[6]
HEX2[0]	PIN_G5	Seven Segment Digit 2[0]
HEX2[1]	PIN_G6	Seven Segment Digit 2[1]
HEX2[2]	PIN_C2	Seven Segment Digit 2[2]
HEX2[3]	PIN_C1	Seven Segment Digit 2[3]
HEX2[4]	PIN_E3	Seven Segment Digit 2[4]
HEX2[5]	PIN_E4	Seven Segment Digit 2[5]
HEX2[6]	PIN_D3	Seven Segment Digit 2[6]
HEX3[0]	PIN_F4	Seven Segment Digit 3[0]
HEX3[1]	PIN_D5	Seven Segment Digit 3[1]
HEX3[2]	PIN_D6	Seven Segment Digit 3[2]
HEX3[3]	PIN_J4	Seven Segment Digit 3[3]
HEX3[4]	PIN_L8	Seven Segment Digit 3[4]
HEX3[5]	PIN_F3	Seven Segment Digit 3[5]
HEX3[6]	PIN_D4	Seven Segment Digit 3[6]

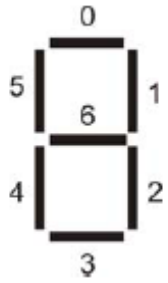
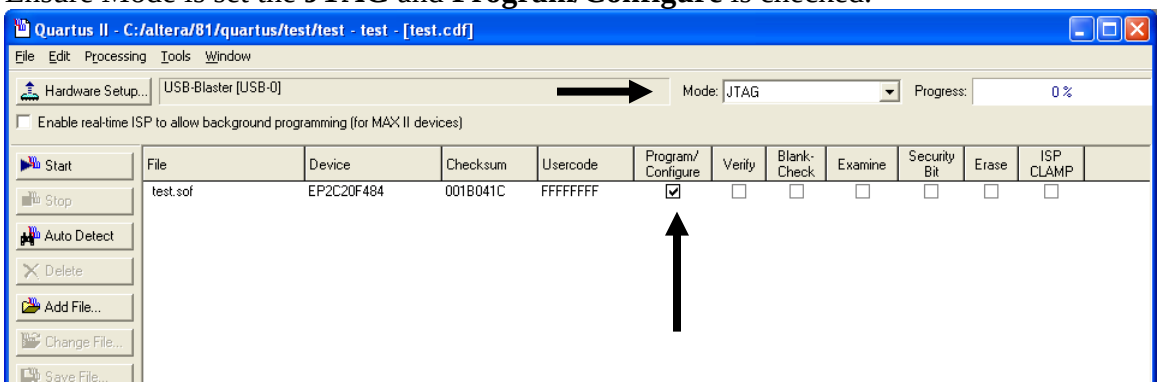


Figure 1: 7 Segment Display

V. Programming the Altera Board using JTAG

Using this method, the program will be erased when power is removed from the DE1 board.

1. Set Switch 12 to **RUN** (Located on the left hand edge of the board).
2. If the project has not been complied recently, compile the Project.
    a. Press **Ctrl+L** or Select **Processing>>Start Compilation**.
    b. Correct any errors that are indicated and recompile.
3. Connect the Power and USB cables
4. Select **Tools>>Programmer**.
5. Ensure Mode is set the **JTAG** and **Program/Configure** is checked.



6. Click **Start**.

VI. Programming the Altera Board using Active Serial Programing

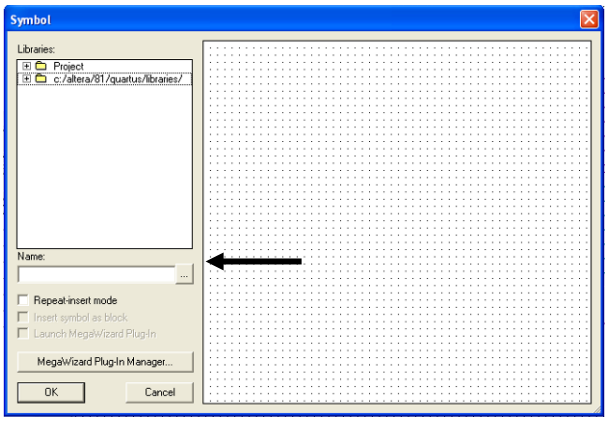
Using this method, the program will remain on the board, even after power is removed.

1. Set Switch 12 to **PROG** (Located on the left had edge of the board).
2. For each project file used, verify device is set to **EPCS4**, by **Assignments>>Device…** then click on the **Device and Pin Options…** button, then the **Configuration** Tab. Finally, under Configuration device, select **EPCS4** instead of Auto.
3. Compile the project by pressing Press **Ctrl+L** or Select **Processing>>Start Compilation.**
4. Connect the Power and USB cables.
7. Select **Tools>>Programmer**.
5. Ensure Mode is set the **Active Serial Programming**. If a warning screen pops up select **Yes**.
6. Click the **Add File…** on the left side of the window.
7. Select your project file with the extension **.pof** and select **Open**.
8. Select the **Program/Configure** checkbox.
9. Click **Start**.
10. After programming is complete, place Switch 12 to **RUN**.
11. Cycle power off then back on.

VII. Creating and Using a Symbol File

Symbol Files allow a user to create a specialized symbol for a circuit so that it can become a block in a larger project. Due to the small sizes of the EE334 projects, symbol files are optional. Advanced students may want to experiment with them, though.

1. To create a symbol, complete the following steps.
    a. Create a new **Diagram/Schematic File.**
    b. Design the desired circuit.
    c. Compile the circuit.
    d. Save the circuit using an appropriate name.
    e. Press **Ctrl+A**.
    f. Select **File>>Create/Update>>Create Symbol Files for Current File**.
    g. Close the project.
2. To use a symbol you have created.
    a. Double click on the schematic to add a part.
    b. Click on **…**



    c. Select the desired symbol name.
    d. If the imported symbol was part of another project, import the file using Project>>Add/Remove File in Project and select the file using the … on the file select screen to import it.