ECE260

Lab 4

Fall 2012

Using PLDs with ABEL HDL and ispLever

Revised: 12 Oct 2012 DAL

(intentionally left blank)

Objective

For this lab you are going to use the Lattice ispLever software package to generate and compile an ABEL program for a programmable logic device (PLD). You will test your design by programming a PLD, wiring the PLD into a circuit and then verifying the operation using either the logic analyzer or the Cadet workstation.

Equipment List

1. ispGAL22V10B
2. ispLever Software Package
3. ISP Programming Adapter
4. Agilent 16801 Logic Analyzer
5. Flying lead pod for Logic Analyzer

Discussion

The ispLever program is actually a suite of Windows based programs that are intended for developing programs for programmable logic devices from Lattice Semiconductor, including simple PLDs, CPLDs, and field programmable gate arrays (FPGAs). The user interface to ispLever is the Project Navigator, which allows you to call other programs that do such things as choose the target device, open the integrated editor program, compile the source files, and call the fitter program that converts the compiled program into an object file (.JED) that can be programmed into the device. The ispLever programs allow the user to define the source programs in one of three hardware description languages (HDL), specifically, ABEL, VHDL and Verilog. For now we will be using ABEL to describe our programmable logic functions.

The ispLever software can be downloaded from the Lattice Semiconductor website (www.latticesemi.com) so you can use the software on you own computers and not rely on just the computers in the lab. To use the downloaded program, you will have to register the software and obtain a free six month license that can be continually renewed.

Pre-Lab

In order to test the capabilities of the ispLever software you need to generate some ABEL source code. You will generate the logic, i.e. write an ABEL program, to detect the prime numbers from all possible combinations of 4 binary bits (0 – F_{16})

You will drive the four inputs to your circuit, Bit_0, Bit_1, Bit_2, and Bit_3, from a 4-bit binary counter which you will implement in the PLD and you will use either the Logic Analyzer or the Cadet workstation to test the operation of the circuit. The ABEL code for the counter is shown below so you will need to incorporate those line into the program you write.

When you get an error free compilation of your ABEL program you will also get a listing file that will list the pin numbers that have been assigned to the variable names you used in the PIN declarations. Figure 1. shows the pin out of the PLD we will be using for this lab. It has all of the I/O pins, power pins and the special pins that attach to the programming pod used to load the JED file generated by the ispLever software. You will need to attach pin 1 (ICLK) to the function generator and four pins of the in-system-programming port (ISP) to the programming adapter.

Before coming to lab do the following:

1. Use a text editor like Notepad to generate an ASCII text file that contains the ABEL program to implement the functions in your design. When you save the file, make the extension *.abl* so that ispLever will recognize it as an Abel source file.
2. Place the Abel source file on a USB memory stick or in a your working directory on the engineering server, Drive Z:

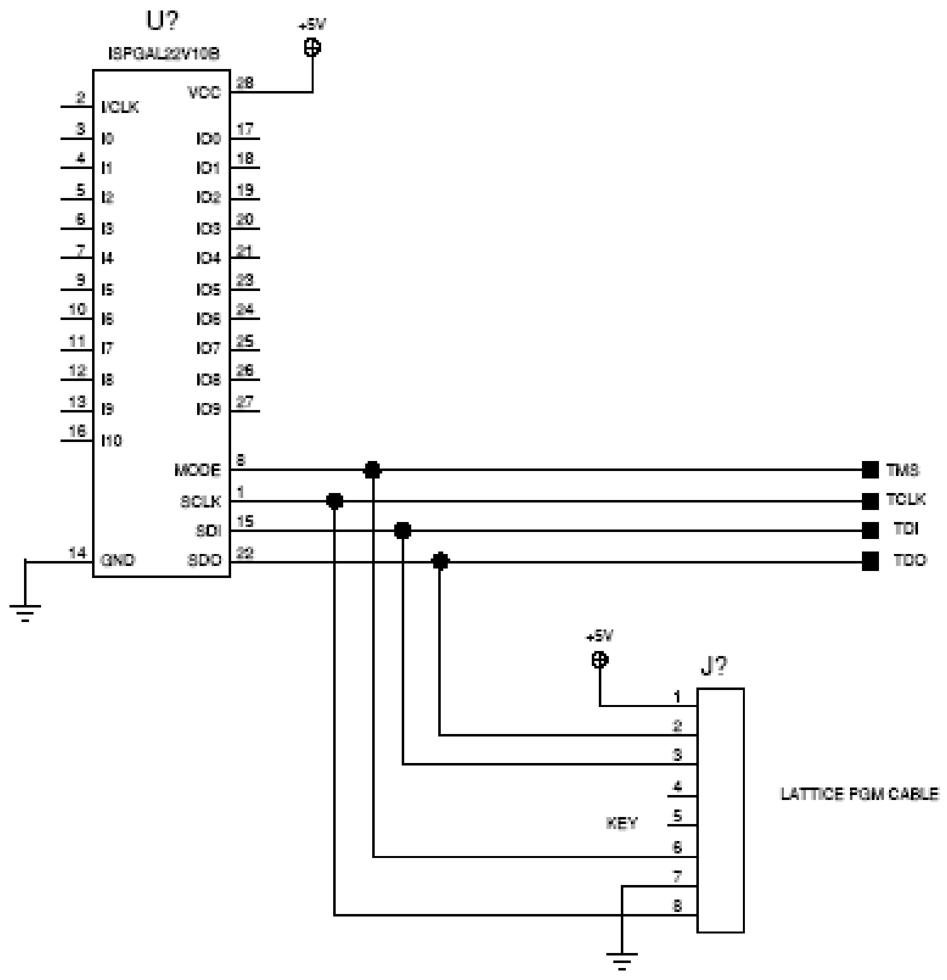


Figure 1. ispGAL22V10 and Programming Port

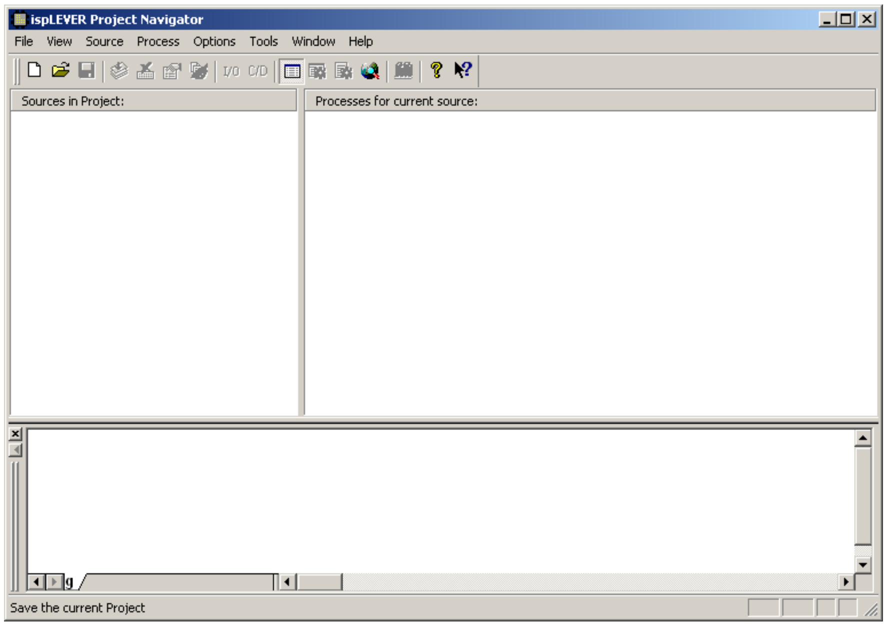
ABEL Code for Binary Counter

add the following lines of code to the appropriate section(s) of your ABEL program

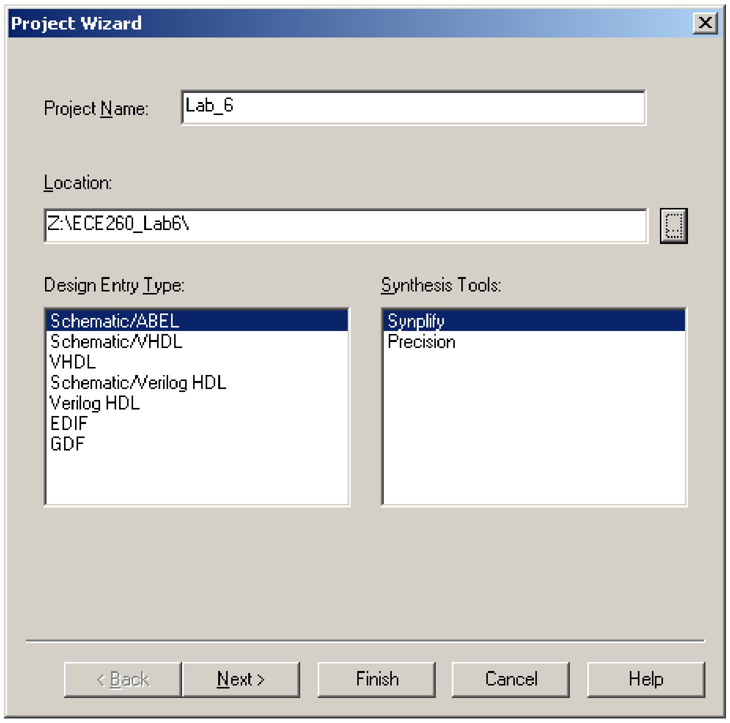
```
declatations
                                    iclk pin ;
        bit_3, bit_2, bit_1, bit_0 pin istype 'reg'  ;

         count = [bit_3, bit_2, bit_1, bit_0];                  "alias for counter

equations

     count.clk = iclk;
          count := count + 1;
```

Procedure

1. Open the ispLever Project Navigator. If there is no shortcut on the Desktop, look in the start menu under Lattice Software. You should see the following screen appear:


ispLEVER Project Navigator
File View Source Process Options Tools Window Help
Sources in Project:
Processes for current source:
Save the current Project

2. On the top line, left click on the File tab. In the pull-down menu select New Project. In the pop-up window, assign a project name of your choosing. In the box labeled *Location*, navigate to a sub-directory in your assigned workspace on Drive Z or on your USB memory stick. For *Design Entry Type*, select <u>ABEL/Schematic</u>. Finally select <u>Synplify</u> for the *Synthesis Tool*. When you have finished, click on Next.



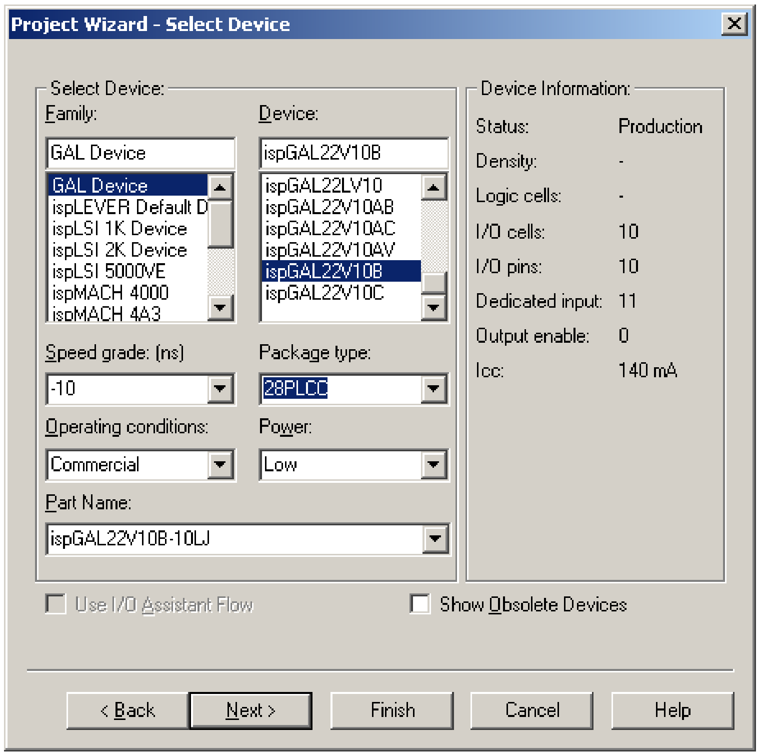
Project Name: Lab_6

Location: Z:\ECE260_Lab6\

Design Entry Type: Schematic/ABEL
Schematic/VHDL
VHDL
Schematic/Verilog HDL
Verilog HDL
EDIF
GDF

Synthesis Tools: Synplify
Precision

3. A new screen opens and asks you to select the device. In the section labeled *Family*, highlight the GAL Device selection. For *Device*, scroll down to the ispGAL22V10B. Under *Package type*, select 28PLCC. The other settings are not critical so click on the Finish button on the bottom of the screen.



Select Device:

Family: GAL Device
GAL Device
ispLEVER Default D
ispLSI 1K Device
ispLSI 2K Device
ispLSI 5000VE
ispMACH 4000
ispMACH 4A3

Device: ispGAL22V10B
ispGAL22LV10
ispGAL22V10AB
ispGAL22V10AC
ispGAL22V10AV
ispGAL22V10B
ispGAL22V10C

Speed grade: (ns) -10

Package type: 28PLCC

Operating conditions: Commercial

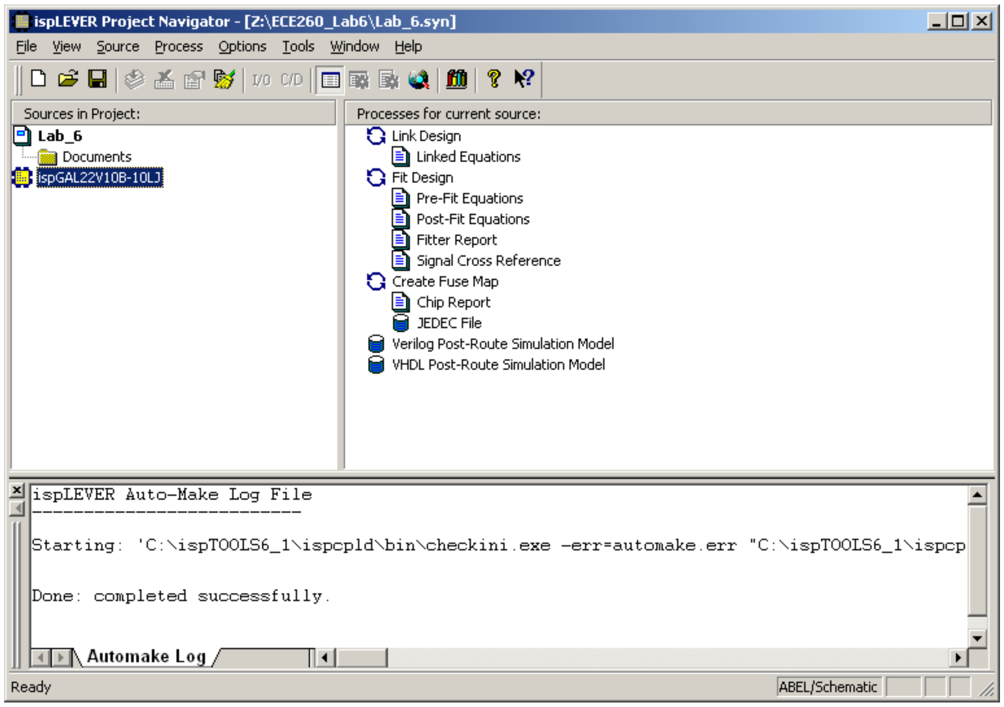
Power: Low

Part Name: ispGAL22V10B-10LJ

Device Information:

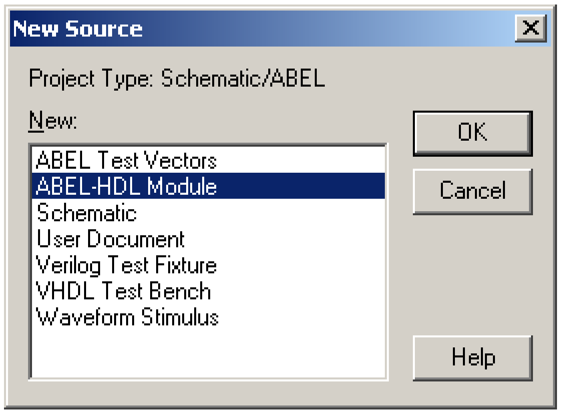
Status:	Production
Density:	-
Logic cells:	-
I/O cells:	10
I/O pins:	10
Dedicated input:	11
Output enable:	0
Icc:	140 mA

4. You should see the following screen. You have to enter the source code for your ABEL program. Click on the Source tab. You can either start a new ABEL source program by selecting *New* or you can import a source file you generated off-line with a text editor by selecting *Import.*

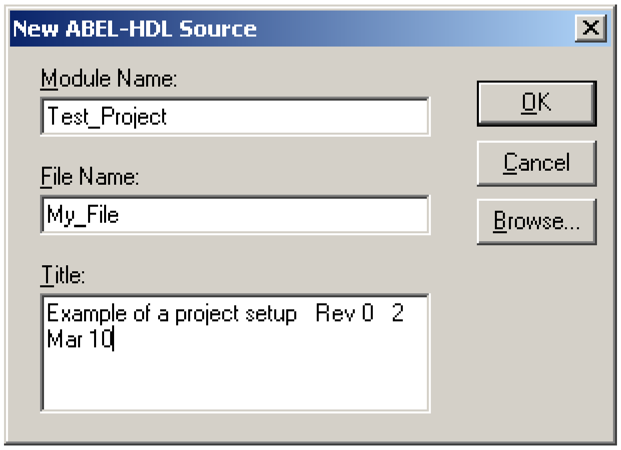


```
ispLEVER Auto-Make Log File
---------------------------

Starting: 'C:\ispTOOLS6_1\ispcpld\bin\checkini.exe -err=automake.err "C:\ispTOOLS6_1\ispcp

Done: completed successfully.
```

5. If you are starting a new file, you will see the following screen. Select ABEL-HDL Module and click OK.



6. In the pop-up window, enter the requested information and click OK. You will be sent to the integrated text editor to start entering the source code of your program.

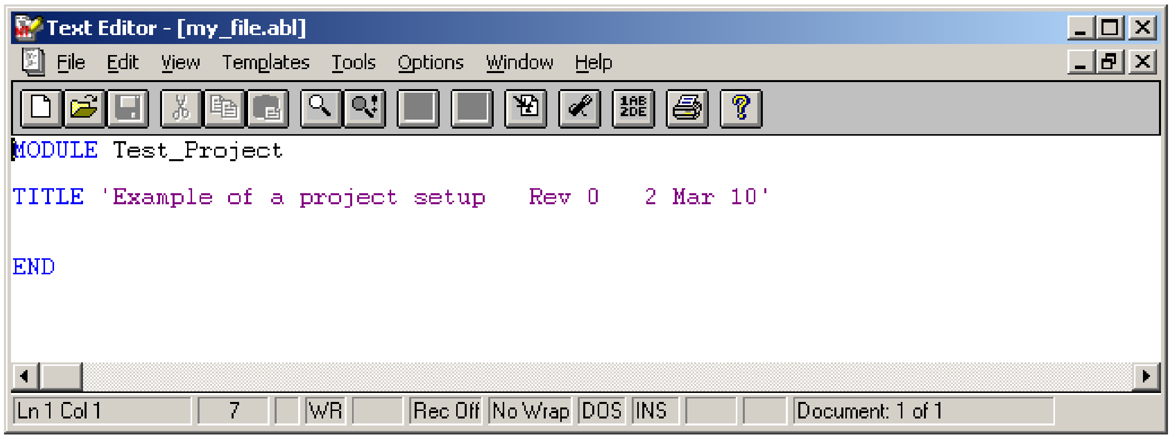


Module Name: Test_Project

File Name: My_File

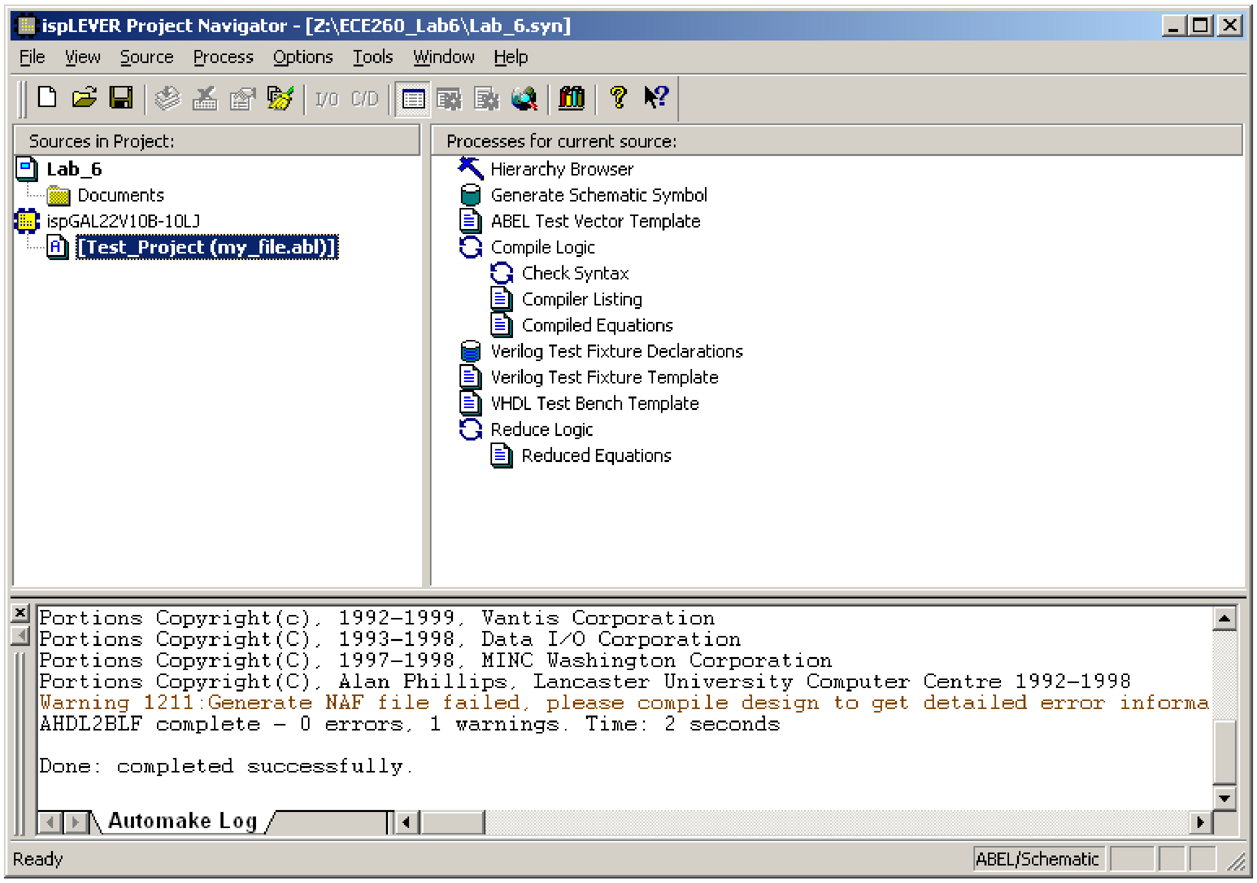
Title: Example of a project setup Rev 0 2 Mar 10

7. If you imported a file, then double click on the file name in the left hand pane of the Navigator window.


Text Editor - [my_file.abl]
File Edit View Templates Tools Options Window Help
Ln 1 Col 1
7
WR
Rec Off
No Wrap
DOS
INS
Document: 1 of 1

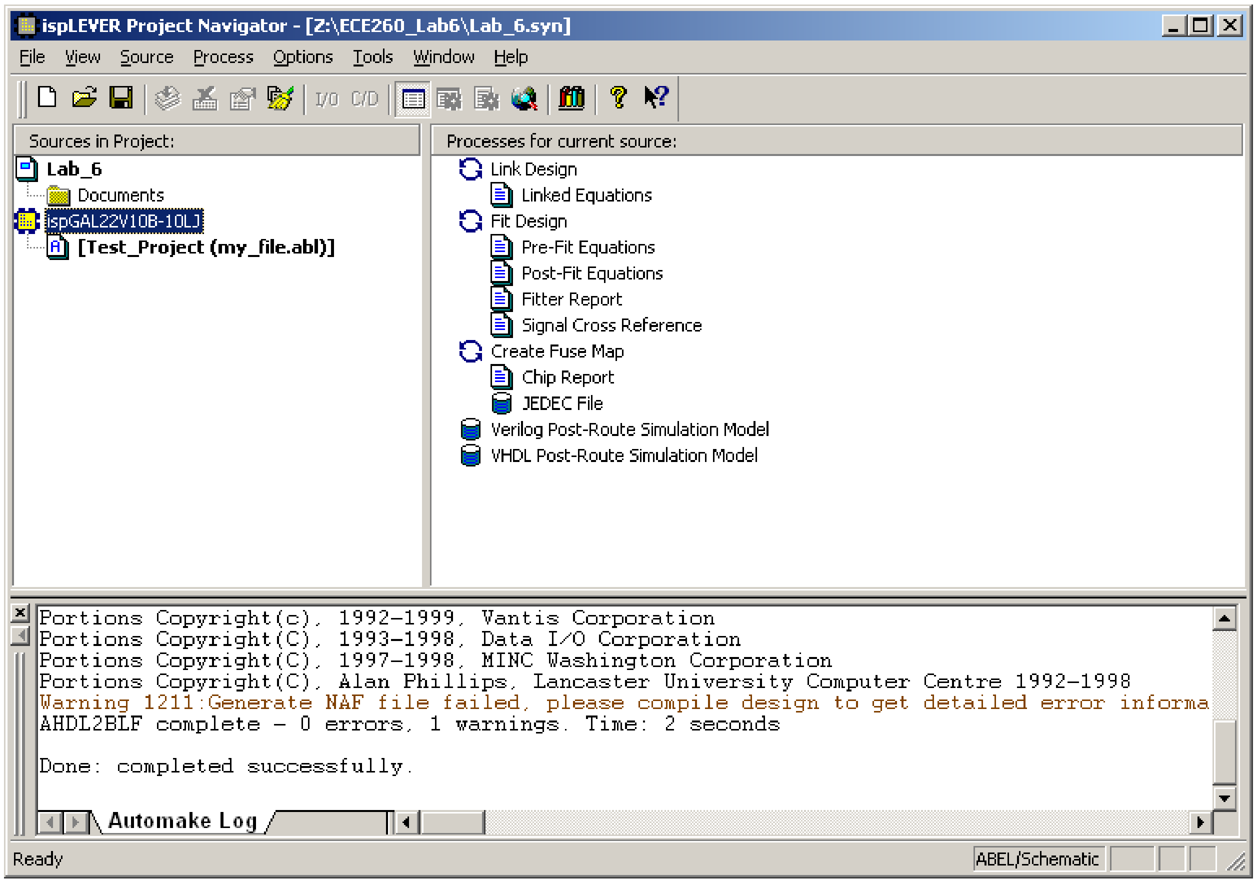
```
MODULE Test_Project

TITLE 'Example of a project setup   Rev 0   2 Mar 10'


END
```

8. In the editor, make any additions/changes and then save the file and go back to the Navigator window. You can either close the editor or leave it open – you will be back here to fix errors in your source code.
9. When the source file is highlighted in the left pane of the Navigator window, the right pane shows tasks related to the source file.


ispLEVER Project Navigator - [Z:\ECE260_Lab6\Lab_6.syn]
File View Source Process Options Tools Window Help
Sources in Project:
Lab_6
Documents
ispGAL22V10B-10LJ
[Test_Project (my_file.abl)]
Processes for current source:
Hierarchy Browser
Generate Schematic Symbol
ABEL Test Vector Template
Compile Logic
Check Syntax
Compiler Listing
Compiled Equations
Verilog Test Fixture Declarations
Verilog Test Fixture Template
VHDL Test Bench Template
Reduce Logic
Reduced Equations
Automake Log
Ready
ABEL/Schematic

```
Portions Copyright(c), 1992-1999, Vantis Corporation
Portions Copyright(C), 1993-1998, Data I/O Corporation
Portions Copyright(C), 1997-1998, MINC Washington Corporation
Portions Copyright(C), Alan Phillips, Lancaster University Computer Centre 1992-1998
Warning 1211:Generate NAF file failed, please compile design to get detailed error informa
AHDL2BLF complete - 0 errors, 1 warnings. Time: 2 seconds

Done: completed successfully.
```

10. The first task is to compile your source file and remove any syntax errors. Highlight the source file in the left window and then highlight Compile Logic in the right pane. At the bottom of that pane, left click on Properties. In the pop-up window, select Generate Listing – Expanded then close the window.
11. Next, left click on the Start box at the bottom of the right pane. This will compile your logic and open up a message window informing you of the status of the compile. If you have compile errors you need to go back to the editor window to correct them and retrace your steps.
12. After you get a compile pass with no errors, you should see green check marks in fron t=of some of the tasks listed in the right pane. Highlight the Compiled Equations line and hit the start button. A report window should open up and it should have a listing of the compiled equations.
13. Repeat these steps for the Reduced equations (if the selection exists).
14. Highlight the device in the left pane of the Navigator window. The right pane shows the tasks related to “fitting” the compiled files to the particular device selected.

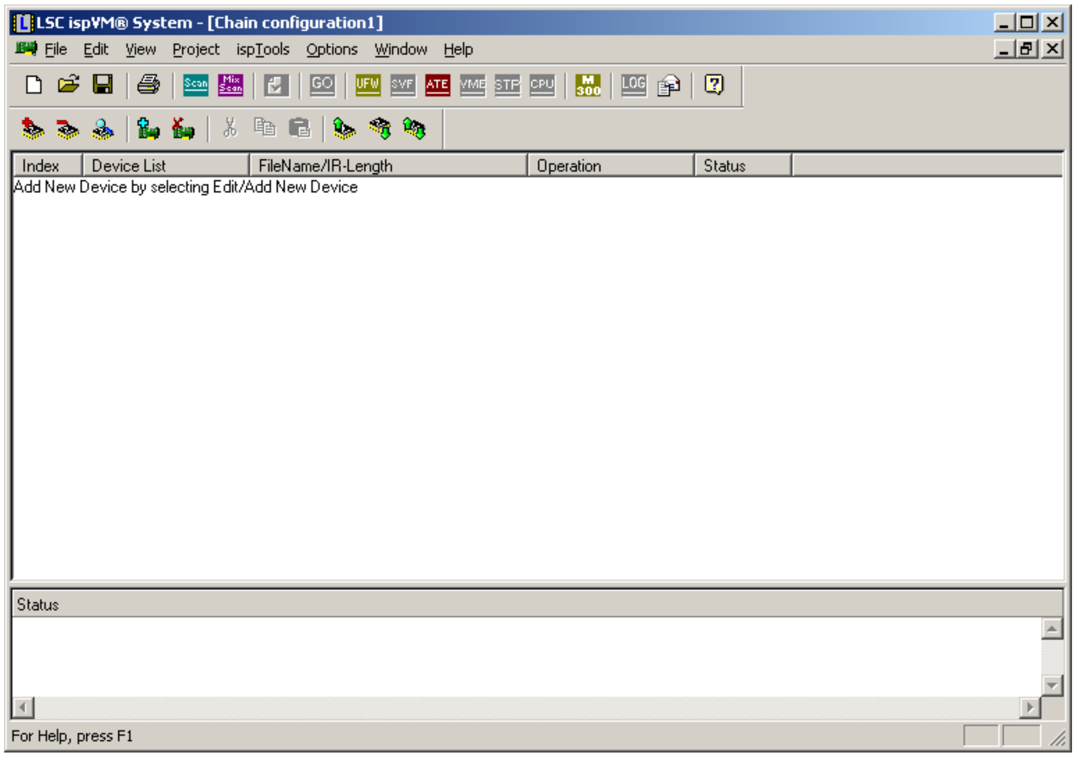

ispLEVER Project Navigator - [Z:\ECE260_Lab6\Lab_6.syn]
File View Source Process Options Tools Window Help
Sources in Project:
Lab_6
Documents
ispGAL22V10B-10LJ
[Test_Project (my_file.abl)]
Processes for current source:
Link Design
Linked Equations
Fit Design
Pre-Fit Equations
Post-Fit Equations
Fitter Report
Signal Cross Reference
Create Fuse Map
Chip Report
JEDEC File
Verilog Post-Route Simulation Model
VHDL Post-Route Simulation Model
Automake Log
Ready
ABEL/Schematic

```
Portions Copyright(c), 1992-1999, Vantis Corporation
Portions Copyright(C), 1993-1998, Data I/O Corporation
Portions Copyright(C), 1997-1998, MINC Washington Corporation
Portions Copyright(C), Alan Phillips, Lancaster University Computer Centre 1992-1998
Warning 1211:Generate NAF file failed, please compile design to get detailed error informa
AHDL2BLF complete - 0 errors, 1 warnings. Time: 2 seconds

Done: completed successfully.
```

15. Highlight Fit Design in the right pane. Again hit the start button and observe the status displayed in the pop-up report window. If there are any errors you will need to go back to the editor to correct them.
16. When the fitting process has completed successfully you are ready to program the device and then test it in the circuit.
17. If you did not assign pin numbers in your original source file, then you need to open the Chip Report to find out what pins have been assigned to your input and output signals. Using this information you can add it to the schematic you created for the prelab.
18. Wire up the circuit from your schematic and connect the wires of the programming pod to the four programming pins on the PLD and to power and ground.
19. When you have finished wiring the board, attach wires to the power supply which you have set to +5V. Attach the clock for the 74LS163 counter to the SYNC output of the function generator.

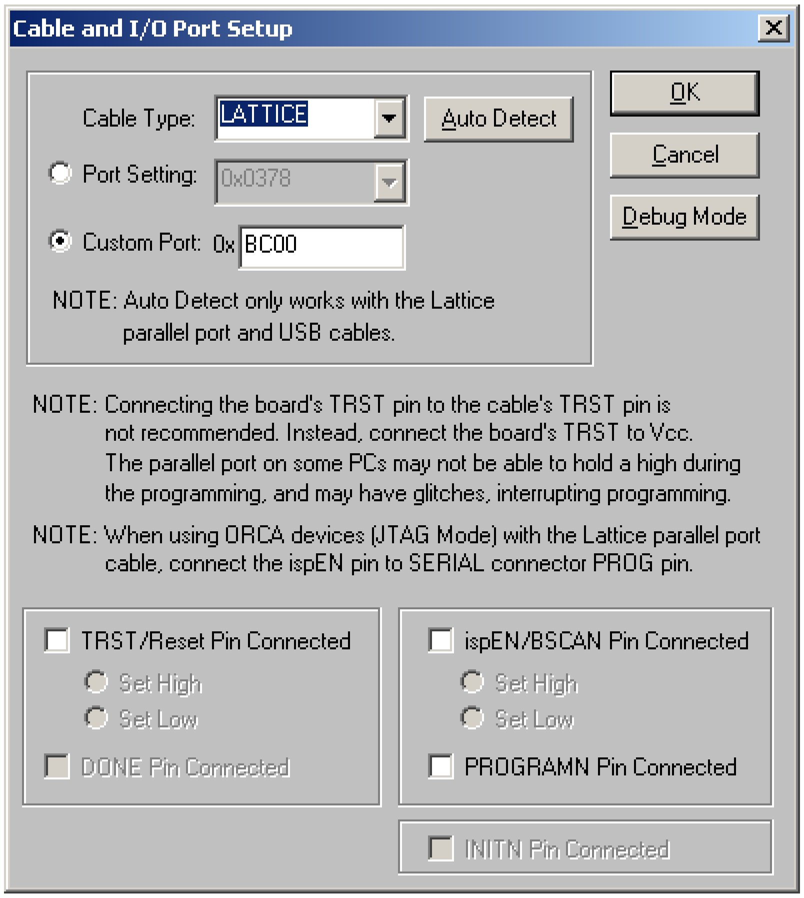
Device Programming:

20. You now need to program the PLD using the *ispVM System* software package. Click on *Start*, *Programs*, *Lattice*, *Accessories* and finally *ispVM*. The following screen should open up.


LSC ispVM® System - [Chain configuration1]
File Edit View Project ispTools Options Window Help
Index
Device List
FileName/IR-Length
Operation
Status
Status
For Help, press F1

Add New Device by selecting Edit/Add New Device

21. You need to set up the software to be able to attach to the pod so now click on the *Options* tab on the toolbar and then select *Cable and I/O Port Setup* in the drop-down window. You should now see the following:



Cable Type: LATTICE

Port Setting: 0x0378

Custom Port: 0x BC00

NOTE: Auto Detect only works with the Lattice parallel port and USB cables.

NOTE: Connecting the board's TRST pin to the cable's TRST pin is not recommended. Instead, connect the board's TRST to Vcc. The parallel port on some PCs may not be able to hold a high during the programming, and may have glitches, interrupting programming.

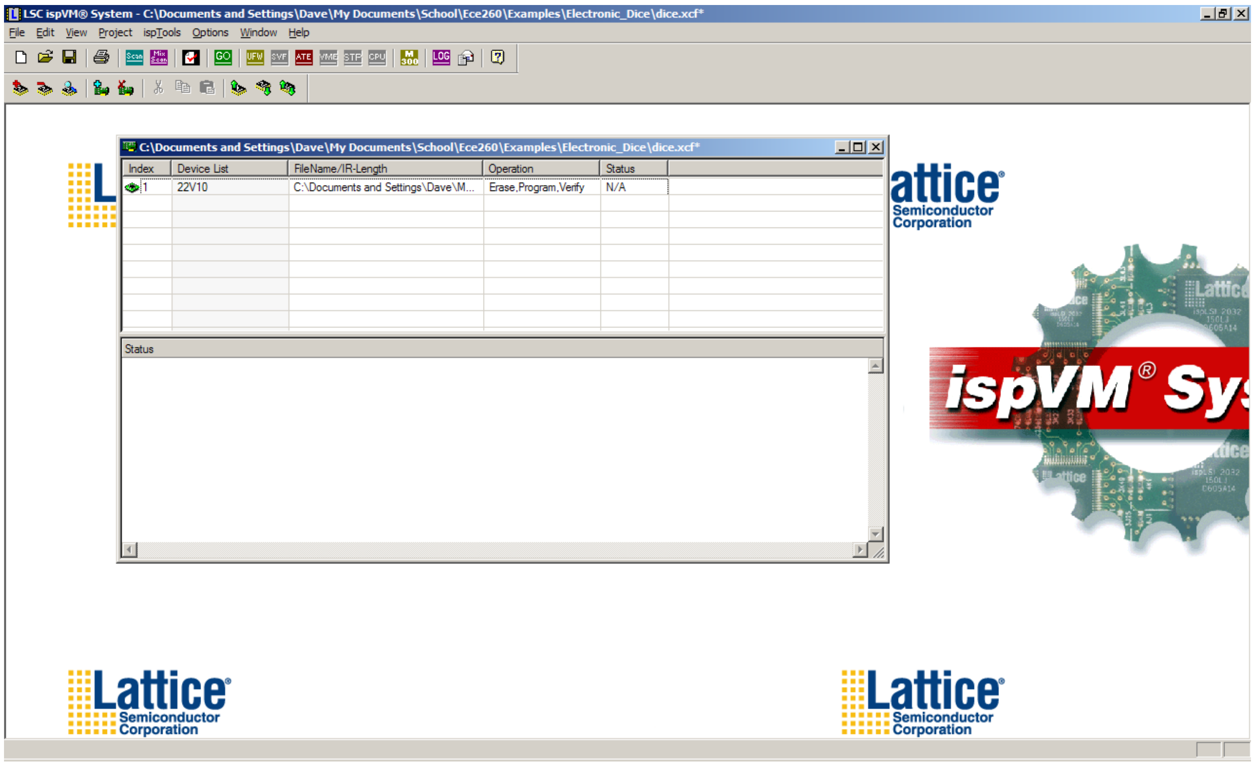
NOTE: When using ORCA devices (JTAG Mode) with the Lattice parallel port cable, connect the ispEN pin to SERIAL connector PROG pin.

☐ TRST/Reset Pin Connected
○ Set High
○ Set Low
☐ DONE Pin Connected

☐ ispEN/BSCAN Pin Connected
○ Set High
○ Set Low
☐ PROGRAMN Pin Connected

☐ INITN Pin Connected

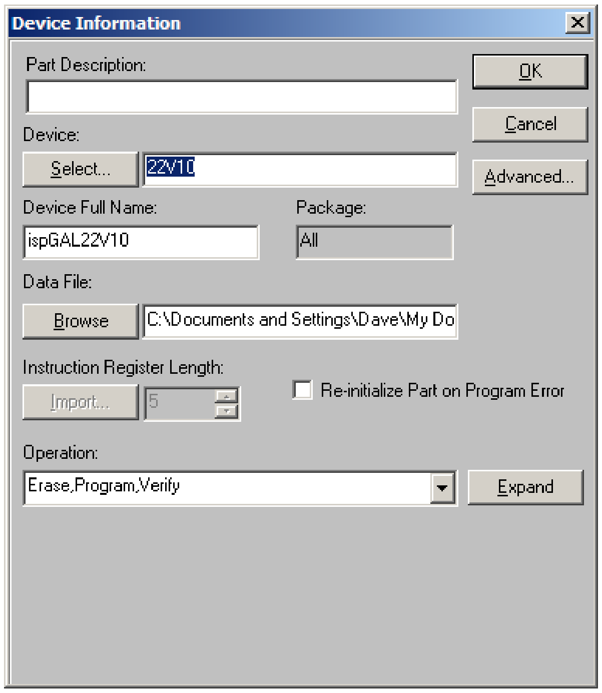
22. The selections used in this setup depend upon the type of programming pod used and will be discussed in the lab. After thee settings are made ckick OK.
23. Now back in the main window click on the *SCAN* tab in the toolbar . This will cause the software to check to see if any devices are attached to the test port and will report the results.


LSC ispVM® System - C:\Documents and Settings\Dave\My Documents\School\Ece260\Examples\Electronic_Dice\dice.xcf*

C:\Documents and Settings\Dave\My Documents\School\Ece260\Examples\Electronic_Dice\dice.xcf*

Index	Device List	FileName/IR-Length	Operation	Status
1	22V10	C:\Documents and Settings\Dave\M...	Erase,Program,Verify	N/A

24. In on the box below *Filename/IR Length.* In the pop-up window. Browse to the location of the JED file and then click OK.


Device Information

Part Description:

Device: 22V10

Device Full Name: ispGAL22V10

Package: All

Data File: C:\Documents and Settings\Dave\My Do

Instruction Register Length: 5

Re-initialize Part on Program Error

Operation: Erase,Program,Verify

25. Back on the main window, click on the tab labeled *GO* on the toolbar. If everything works correctly, you should get a message that states PASSED; if not ask for assistance from the TA or the instructor.

Testing with Cadet Workstation:

To use the Cadet workstation to test your design you use a very slow input clock to drive the counter and the outputs of the counter and results will be tied to LED indicators.

1. Connect a wire from the square wave output of the function generator of the CADET to pin 1 of the PLD (iclk) and to LED 7 for observation.
2. Connect Bit_3 thru Bit_0 to 4 the LED 0 – 3 on the CADET.
3. Connect the two result pins, PRIME_EQ and PRIME_TT, to two of the LEDs.
4. Now set the function generator to a very slow clock and verify that the corresponding LED is flashing on and off at a slow rate.
5. Verify the the 4 bits of the counter are counting up through all sixteen binary combinations.
6. Verify that the PRIME_EQ and PRIME_TT LED are turned on at the correct state of the inputs.
7. If not, try to figure out the problem and then edit your ABEL program and then re-compile and reprogram the chip and test it again.
8. After you have the program working correctly, demonstrate it to the TA or instructor.

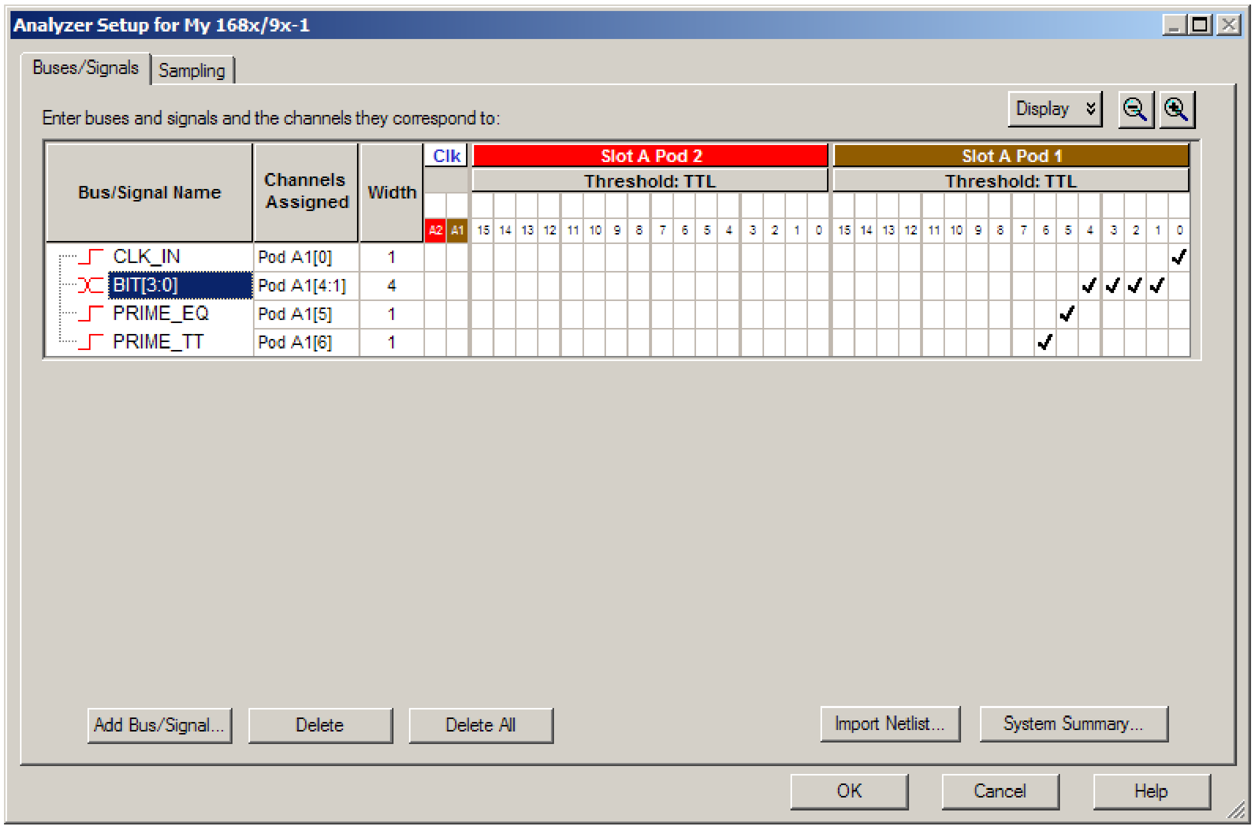
Testing with Logic Analyzer:

To use the Logic Analyzer to test your design you use high speed input clock from the Function Generator to drive the counter and then the outputs of the counter and results will be verified using the Analyzer .

1. Using the information you learned in Lab 4, attach leads of the flying lead pod of the logic analyzer to the following signals:

a.	Lead 0	CLK_IN	from function generator
b.	Lead 1	Bit_0	least significant bit of the four bit binary counter
c.	Lead 2	Bit_1	
d.	Lead 3	Bit_2	
e.	Lead 3	Bit_3	most significant bit of the four bit binary counter
f.	Lead 4	Prime_eq	output from the Boolean equation solution
g.	Lead 5	Prime_tt	output from the truth table solution

2. Set up the traces on the analyze as follows:



Enter buses and signals and the channels they correspond to:

Bus/Signal Name	Channels Assigned	Width
CLK_IN	Pod A1[0]	1
BIT[3:0]	Pod A1[4:1]	4
PRIME_EQ	Pod A1[5]	1
PRIME_TT	Pod A1[6]	1

3. Now set the function generator to approximately 1MHz and attach it to the CLK_IN signal in your circuit.
4. Capture some analyzer traces and use them to determine if your circuit is working as you expected.
5. Observe the timing on the counter output pins.
6. Have the TA or the instructor verify your results and sign the Verification Sheet.

Lab Report: The lab report is to contain the following sections: (see Lab Report Guideline)

1. Title sheet
2. Objective
3. Material List
4. Procedure
5. Experiment data
   a. If you used the Logic Analyzer, include the traces you printed out along with explanation of the data being displayed
   b. Copy of your schematic showing pin out
   c. Copy of the ABEL source code for your program
   d. Explanation of any problems and/or errors and how you would have corrected them
6. What you need to know section including a brief explanation of your program
7. Appendix
   a. Signed verification sheet