LAB: OPERATION OF PASTEURIZATION PROCESS CONTROL
Using Xilinin ISE 9.2i Project Navigator and
Spartan 3E FPGA Development Board with Schematic and DHDL

Digital Fundamentals: {Insert Module Name}

Acknowledgements

Subject Matter Experts: Bassam Matar, Faculty at Chandler-Gilbert Community College, Chandler, Arizona and Ui Luu from Glendale College, Glendale, Arizona. *Funded by NSF*

Purpose

The goal of this lab is to learn the use of Spartan 3E FPGA development board from Xilinx and how to create the hardware connections between the development board and your PC.

Systems Rationale

In previous labs, you became familiar with TTL technology and bread boarding. In this lab, you will become familiar with a modern way of digital technology using one of the leading digital design software applications from "Xilinx" and interfacing this with the Spartan 3E FPGA development board. Combinational Logic Circuits are used to make decisions based on a series of true statements that can be laid out in a truth table. In previous years the 74/54xx, family (TTL) circuits were used to design and build basic combinational logic circuits. In this lab activity, we will use the software application Xilinx Project Navigator (ISE9.2i) to program the Operation of Pasteurization Process Control to the Spartan 3E FPGA development board.

System Concepts

This system covers the following system concepts (signified by an X):

_X_ S1. A system can be defined in terms of its functional blocks i.e., a "structured functional unit."

_X_S2. A system has a purpose, transforms inputs into outputs to achieve a goal.

___S3. A system is defined by the flow of materials, energy and information, between its functional units.

_X_S4. A system may be open or closed. In an open system additional inputs are accepted from the environment.

_X_S5. A system is more than the sum of its parts. Individual components can never constitute a system.

_X_S6. A system provides feedback to the operator and services to the user. Some system functions may involve operator action.

___S7. Systems have unique problems.

Learning Outcomes

{selected from the list from the SLO Tab from eSyst. Put the appropriate link in as well; i.e. http://www.esyst.org/Courses/DC-AC/_delivery/index.php}

For a full course SLOs, click the link and click SLO tab.

Learning Objectives

1. Getting familiar with Spartan 3E FPGA board. The advantage of this board is programmed through USB port. JTAG port is used to program previous versions.
2. Implement the VHDL and User Constraint File “ucf” that maps the input and output signals to the Spartan 3E FPGA using Xilinx® ISE 9.2i, compile and simulate for Xilinx Spartan 3E FPGA
3. Test the Results
4. Compare your results with the traditional way of TTL technology and bread boarding.

Grading Criteria

Your grade will be determined by your instructor.

Time Needed

Instructor Setup/Cleanup:
This will vary depending on the particular lab room arrangement and storage systems available at your institution.

Student Lab Performance:
It should take students approximately 2-3 hours to work through the entire lab.

Student Lab Deliverables:
It should take students approximately {insert time, e.g. 2.5 hours} of homework time to create the {insert deliverable name(s)}.

Materials, Equipment, & Supplies Needed

Item	Quantity
Instructor lab guide	1
Student lab guides	1/student
{List any other instructor materials, equipment, & supplies needed. Then, insert equipment and supplies text from student guide}.	

Special Safety Requirements

{Insert special safety requirements text from student guide}.

No serious hazards are involved in this laboratory experiment, but be careful to connect the components with the proper polarity to avoid damage.

Lab Preparation

- Review your PLD lecture from your class.
- Read the lab and the attached appendix to gain familiarity with the FPGA development board

Equipment and Materials

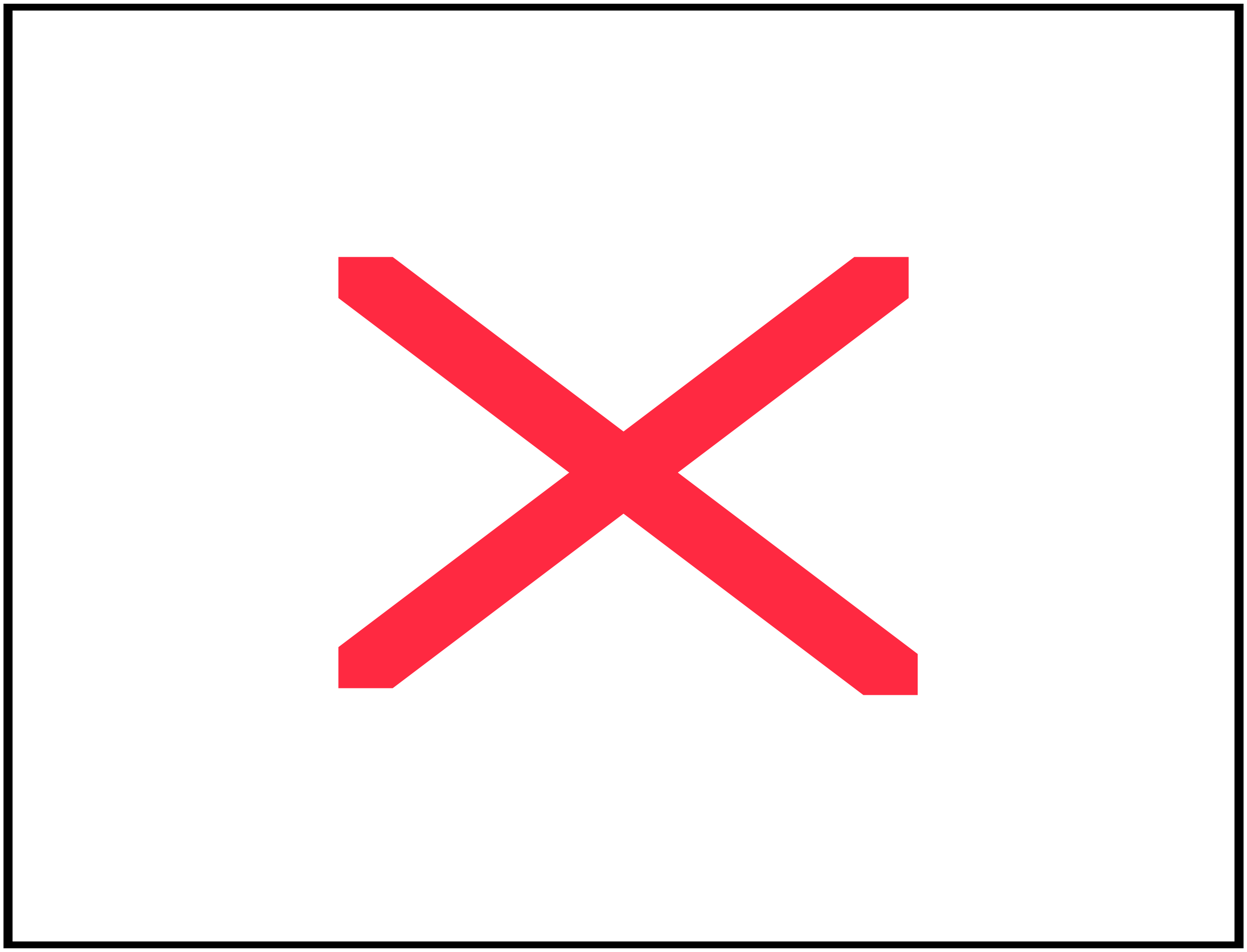
Each team of students will need the test equipment, tools, and parts specified below. Students should work in teams of two or three.

Test Equipment and Power Supplies	Quantity
The following items from the Xilinx: • Free software ISE *WebPACK* (**www.xilinx.com**) that can be installed on your personal computer or full version of Xilinx in your classroom • Spartan-3E Starter Kit, including download cable and power supply	1
Switch Module	2

Additional References:

1. Xilinx Spartan 3E FPGA Reference Manual and Schematic from: http://www.digilentinc.com/Products/Detail.cfm?Prod=S3EBOARD&Nav1=Products&Nav2=Programmable

System Interface Description



Consider the system block diagram / Beer Pasteurization Process Control using FPGA (Figure 1).

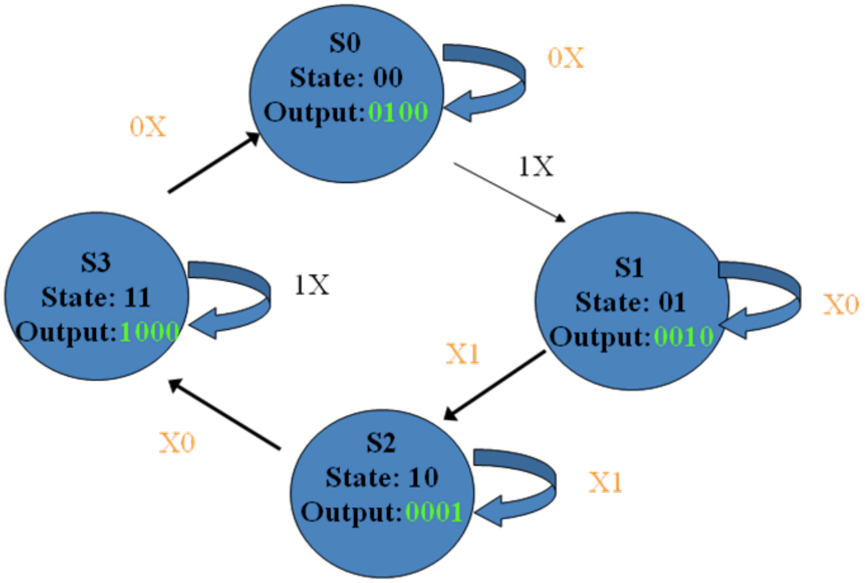
The system consists of a process controller implemented by FPGA interacts with a microbrewer model implemented by a PC with the following I/O interface:

- Inputs from Microbrewer model
  - Reset
  - Level Sensor
  - Temperature Sensor
- Output controls:
  - Inlet Valve
  - Heater
  - Chiller
  - Pump

The process controller operates in 4 states:

- State S0:
  - Turn OFF Pump
  - Turn on Inlet Valve
  - When Level =1 (container full), transition to State S1
- State S1:
  - Turn OFF Inlet Valve
  - Turn ON Heater
  - When Temp = 1 (Hot), transition to State S2
- State S2:
  - Turn OFF Heater
  - Turn ON Chiller
  - When Temp = 0 (Cold), transition to State S3
- State S3:
  - Turn OFF Chiller
  - Turn ON Pump
  - When Level = 0 (container empty), transition to State S0

State Diagram



Input: L T (L: Level Sensor) (T: Temperature Sensor)
Outputs: PVHC (P: Pump, V: Inlet Valve, H: Heater, C: Chiller)

State Table

Here is a truth table that should guide you with your test.

Reset	Level	Temp	Current State	Pump	Valve	Heater	Chiller
0	0	0	S_0	OFF	ON	OFF	OFF
0	0	1	S_0	OFF	ON	OFF	OFF
0	1	0	S_0	OFF	OFF	ON	OFF
0	1	1	S_0	OFF	OFF	ON	OFF
0	0	0	S_1	OFF	ON	OFF	OFF
0	0	1	S_1	OFF	ON	OFF	OFF
0	1	0	S_1	OFF	OFF	OFF	ON
0	1	1	S_1	OFF	OFF	OFF	ON
0	0	0	S_2	ON	OFF	OFF	OFF
0	0	1	S_2	OFF	OFF	OFF	ON
0	1	0	S_2	ON	OFF	OFF	OFF
0	1	1	S_2	OFF	OFF	OFF	ON
0	0	0	S_3	OFF	ON	OFF	OFF
0	0	1	S_3	OFF	ON	OFF	OFF
0	1	0	S_3	ON	OFF	OFF	OFF
0	1	1	S_3	ON	OFF	OFF	OFF

State Table

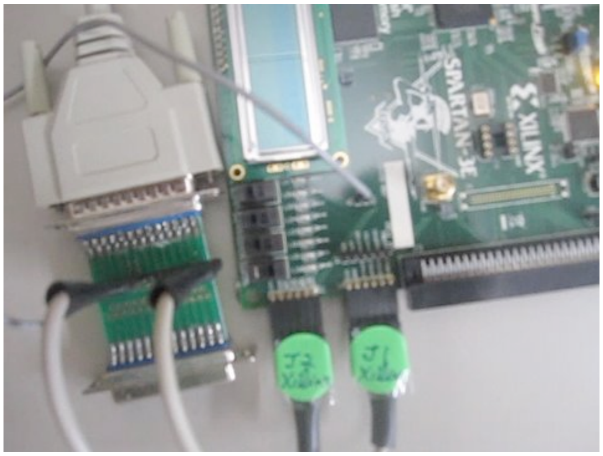
Components required

1. Xilinx Spartan 3E FPGA board

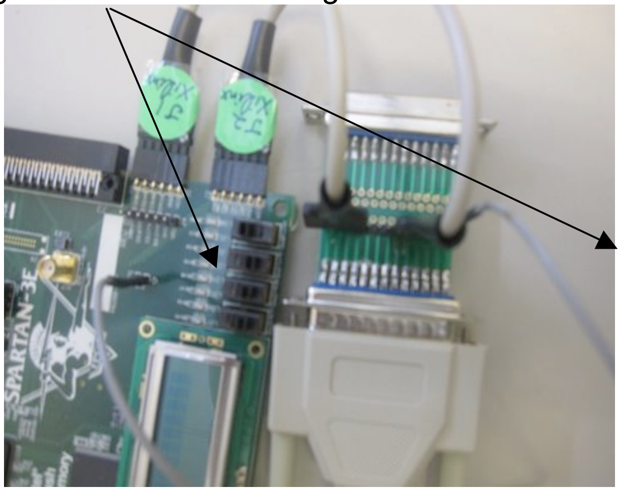
2. PC/Windows XP with parallel port
3. DB25 cable: M-FM
4. DB25 breakout box
5. J1: 6-pin adapter
6. J2 6-pin adapter
7. FPGA control program in BeerControl folder (electronic)
8. PC Beer model in BeerModel folder (electronic)

Hardware set up

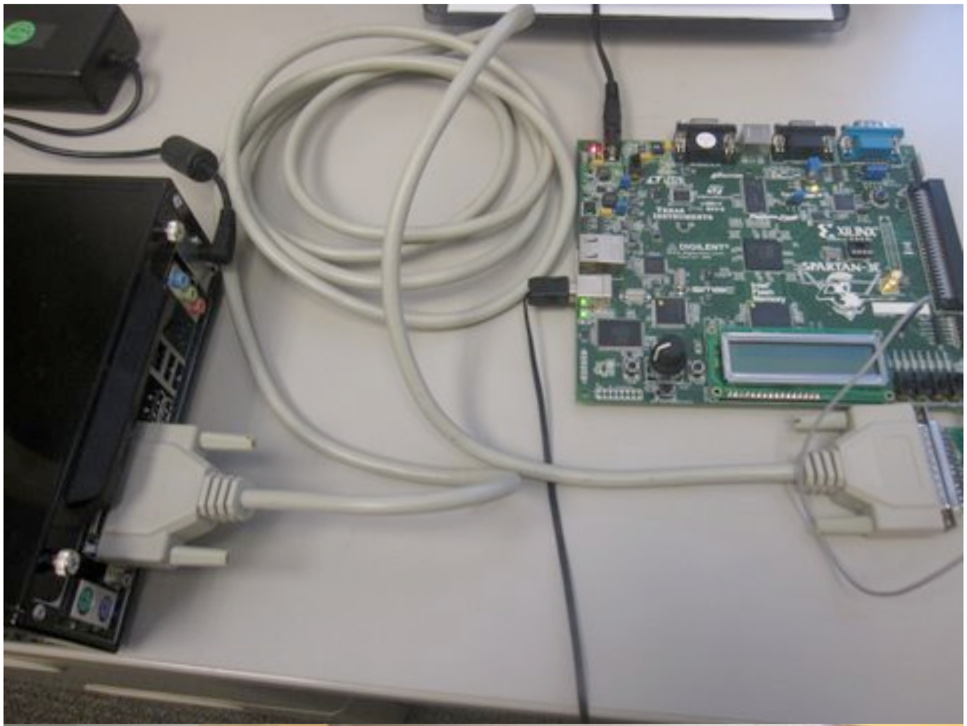
1. Attach USB connector of the FPGA board to your PC
2. Attached power cord to FPGA board.
3. Attach J1 and J2 connector to Xilinx board.

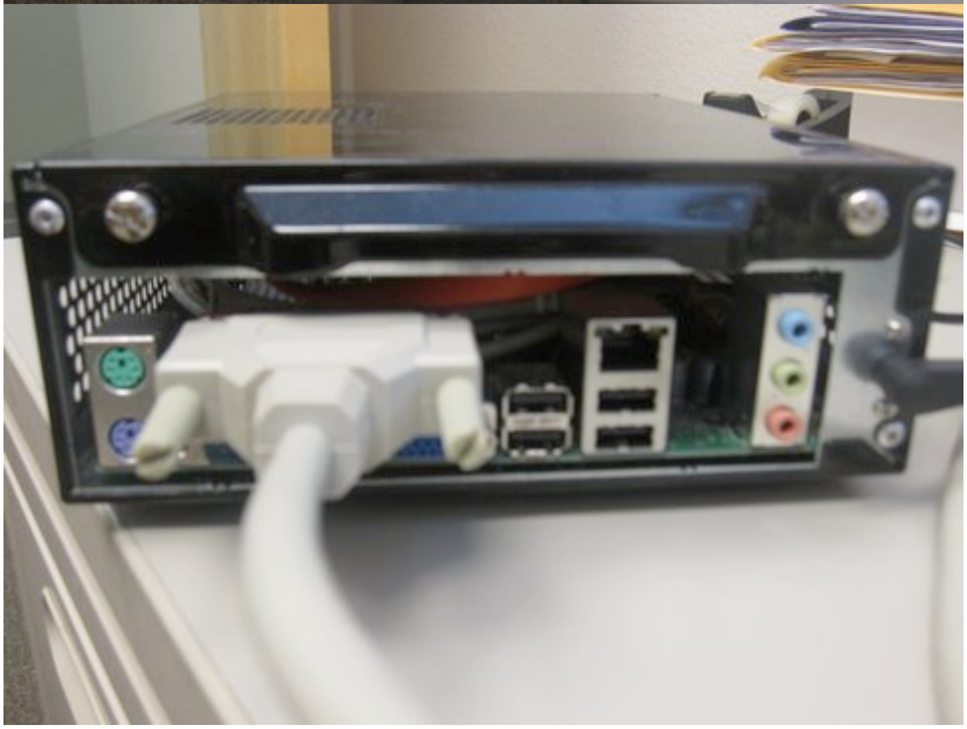


4. Attach system ground to FPGA board ground.



5. Attach DB 25 cable to model PC.





Beer Model Software set up at PC

1. Copy BeerModel folder to C: drive.
2. Copy: WinIO.dll, WinIO.sys, WINIO.VXD to Windows\system32 folder. These drivers allow the Beer Model to have direct access to digital I/O at parallel port.
3. To run the Beer Model, double click on BeerModel.exe.

Task 1:
Implement Beer Control using Xilinx ISE 9.2i tools for Spartan 3E FPGA board:

Part 1:

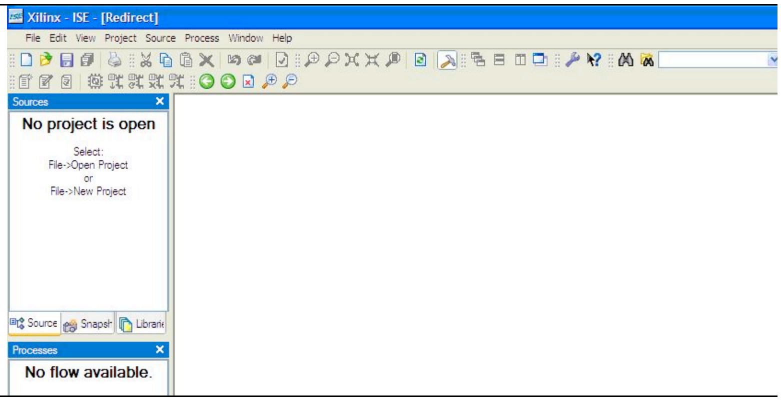
1. We need to set up our project correctly to reflect Spartan 3 FPGA board.

Open Xilinx ISE 9.2 edition software

a. Select Start
b. All Programs
c. Xilinx ISE 9.2 edition
d. Project Navigator

Or double click on the desktop icon:

1. The starting windows should look like this


Xilinx - ISE - [Redirect]
File Edit View Project Source Process Window Help
Sources
Source
Snapsh
Librarie
Processes

No project is open

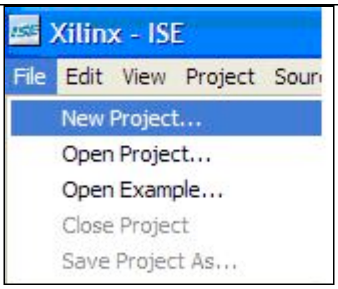
Select:
File->Open Project
or
File->New Project

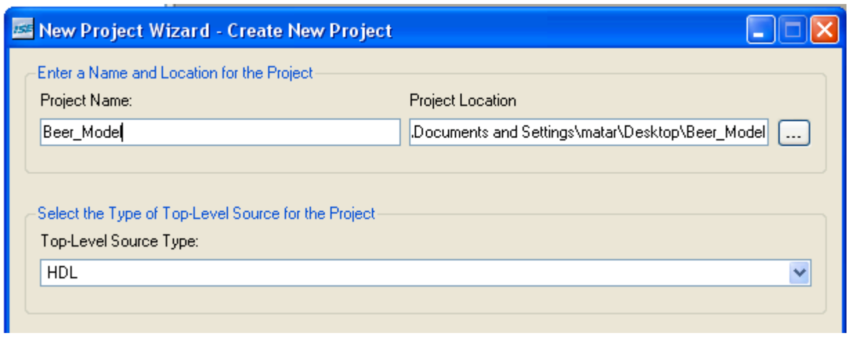
No flow available.

Figure 1: Xilinx Starting Window

2. Create a new project by selecting:
File from the main menu
New Project

a. In the New Project window, name your project *Beer Model* in the project name text box.
b. In the **Project Location** selection box, enter the folder or directory where your project will be saved. Use your name as for




New Project Wizard - Create New Project
Enter a Name and Location for the Project
Select the Type of Top-Level Source for the Project
...

Project Name: Beer_Model

Project Location: .Documents and Settings\matar\Desktop\Beer_Model

Top-Level Source Type: HDL

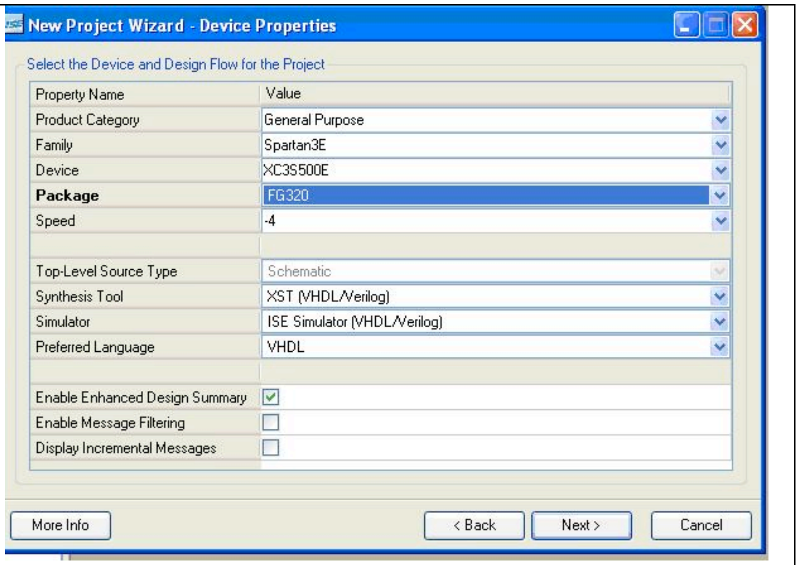
Student_Name and locate the place where you want to save all your files (i.e. C:\)

c. Under **Top-Level Source Type**, select **HDL** and click Next as shown in Figures 1-3

Figure 2: New Project Window

3. We will design our Beer Control for a particular device "Spartan 3E FPGA".

Product Category: General Purpose
Device Family: Choose Spartan 3E, the device we will be using.
Device: XC3500E, is the specific Spartan 3E device we use. This is actually printed (very small) on the FPGA core.
Package: FG320, this is the package type of our device (Ball Grid Array, 320 pins)
Speed Grade: The speed grade for this device is -4.



New Project Wizard - Device Properties

Select the Device and Design Flow for the Project

Property Name	Value
Product Category	General Purpose
Family	Spartan3E
Device	XC3S500E
Package	FG320
Speed	-4
Top-Level Source Type	Schematic
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISE Simulator (VHDL/Verilog)
Preferred Language	VHDL
Enable Enhanced Design Summary	☑
Enable Message Filtering	□
Display Incremental Messages	□

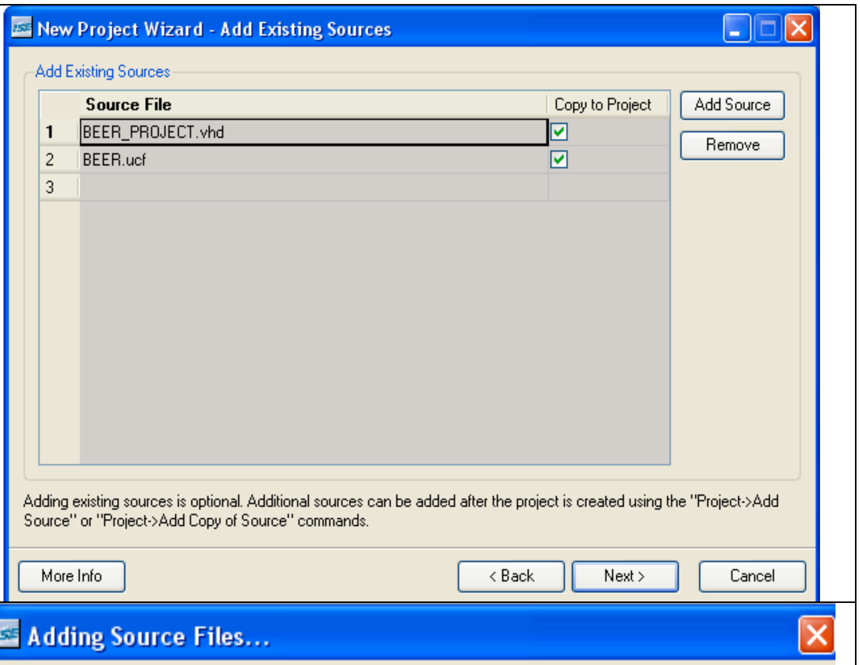
Figure 3: Device Properties

Hit the Next button ***TWICE*** to go to the next menu as shown in step 4.

4. Select **Add Source** and add the following existing files:

Beer_PROJECT.vhd
BEER.ucf

as shown and then click Next.



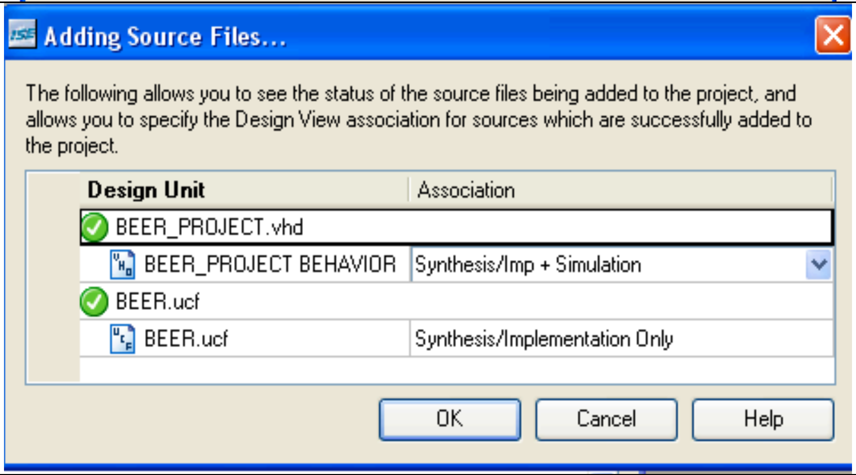
New Project Wizard - Add Existing Sources

Add Existing Sources

	Source File	Copy to Project
1	BEER_PROJECT.vhd	☑
2	BEER.ucf	☑
3		

Adding existing sources is optional. Additional sources can be added after the project is created using the "Project->Add Source" or "Project->Add Copy of Source" commands.

You should get the following green checkmarks if you selected the files correctly.



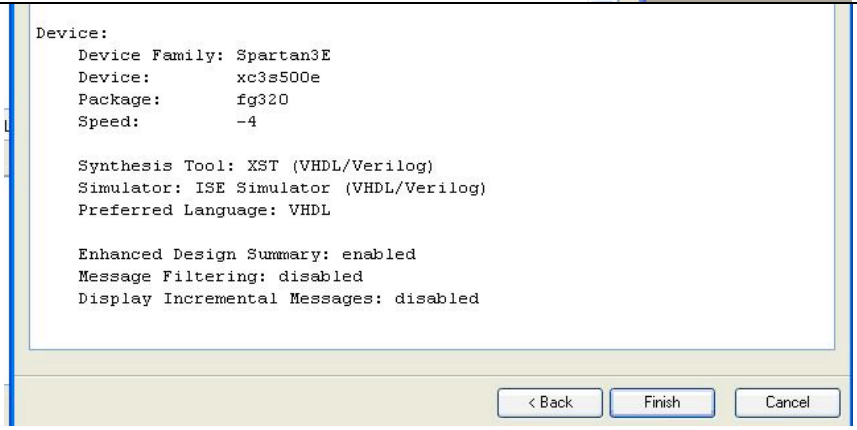
Adding Source Files...

The following allows you to see the status of the source files being added to the project, and allows you to specify the Design View association for sources which are successfully added to the project.

Design Unit	Association
✓ BEER_PROJECT.vhd	
BEER_PROJECT BEHAVIOR	Synthesis/Imp + Simulation
✓ BEER.ucf	
BEER.ucf	Synthesis/Implementation Only

5. Review the information listed in Figure 4 to insure it matches the information in the window.

Then click **Finish** to complete the process and verify the file name and type. Click **Next** once again to proceed and finish.



```
Device:
    Device Family: Spartan3E
    Device:        xc3s500e
    Package:       fg320
    Speed:         -4

    Synthesis Tool: XST (VHDL/Verilog)
    Simulator: ISE Simulator (VHDL/Verilog)
    Preferred Language: VHDL

    Enhanced Design Summary: enabled
    Message Filtering: disabled
    Display Incremental Messages: disabled
```

Figure 4: Project Summary

Part 2:
Program Beer Control into the Flash PROM FPGA Spartan 3 board

Generating the FPGA Configuration Bitstream File (i.e. Beer Project)

Before generating the PROM file, create the FPGA bitstream file. The FPGA provides an output clock, CCLK, when loading itself from an external PROM. The FPGA's internal CCLK oscillator always starts at its slowest setting, approximately 1.5 MHz. Most external PROMs support a higher frequency. Increase the CCLK frequency as appropriate to reduce the FPGA's configuration time. The Xilinx XCF04S Platform Flash supports a 25 MHz CCLK frequency.
Right-click **Generator Programming File** in the Processes pane, as shown in Figure 5. Left-click: **Properties**.

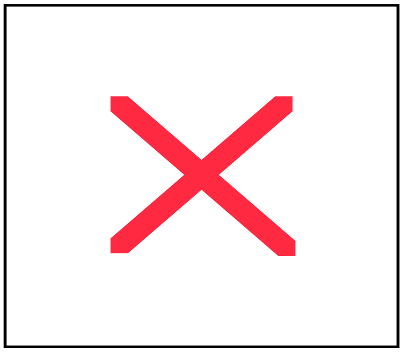


Figure 5: **Set Properties for Bitstream Generator**

Click **Configuration Options** as shown in Figure 6. Using the **Configuration Rate** drop list, choose **25** to increase the internal CCLK oscillator to approximately 25 MHz, the fastest frequency when using an XCF04S Platform Flash PROM. Click **OK** when finished.

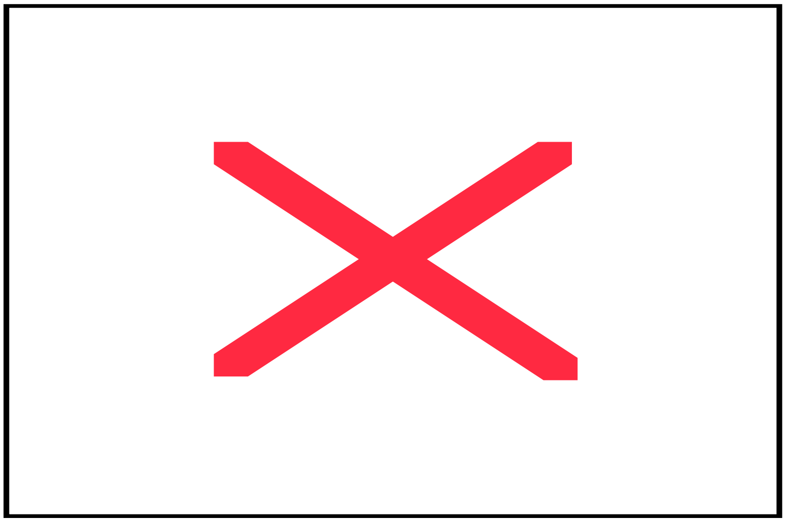


Figure 6: **Set CCLK Configuration Rate under Configuration Options**

Generating the PROM File

To generate the programming file, double-click: **Generate PROM, ACE, or JTAG File** in the Process pane to launch the iMPACT software, as shown in Figure 7.

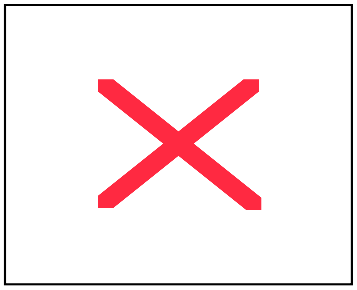
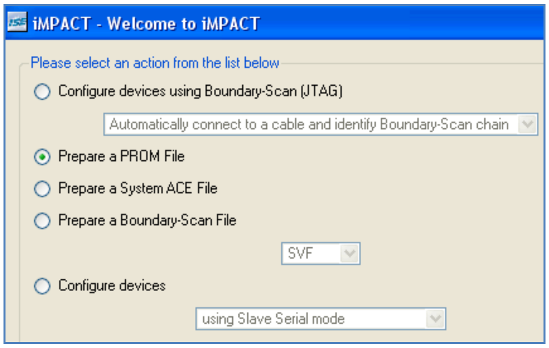


Figure 7: **Double-Click Generate PROM, ACE, or JTAG File**

After iMPACT process starts, select **Prepare a PROM file**, as shown in Figure 8.



iMPACT - Welcome to iMPACT

Please select an action from the list below

○ Configure devices using Boundary-Scan (JTAG)

Automatically connect to a cable and identify Boundary-Scan chain

● Prepare a PROM File

○ Prepare a System ACE File

○ Prepare a Boundary-Scan File

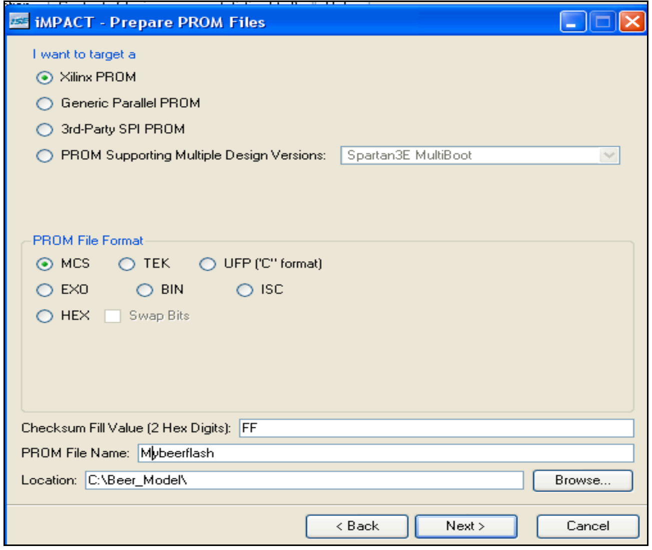
SVF

○ Configure devices

using Slave Serial mode

Figure 8: **Select Prepare a PROM File**

Choose Xilinx PROM as the target PROM type, as shown in Figure 9. Select from any of the PROM File Formats; the Intel Hex format (**MCS**) is a popular format. Enter the **Location** of the directory and the **PROM File Name (MyBeerFlash)**. Click **Next >** when finished.



iMPACT - Prepare PROM Files

I want to target a

● Xilinx PROM

○ Generic Parallel PROM

○ 3rd-Party SPI PROM

○ PROM Supporting Multiple Design Versions: Spartan3E MultiBoot

PROM File Format

● MCS ○ TEK ○ UFP ("C" format)

○ EXO ○ BIN ○ ISC

○ HEX □ Swap Bits

Checksum Fill Value (2 Hex Digits): FF

PROM File Name: Mybeerflash

Location: C:\Beer_Model\

Figure 9: **Choose the PROM Target Type, the Data Format, and the File Location**

The Spartan-3E Starter Kit board has an XCF04S Platform Flash PROM. Select xcf04s from the drop list, as shown in Figure 10. Click **Add**, and then click **Next**.

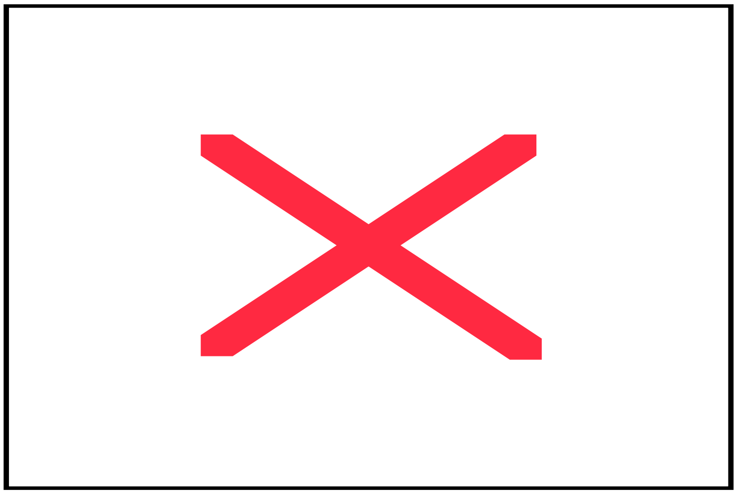


Figure 10: **Choose the XCF04S Platform Flash PROM**

The PROM Formatter then echoes the settings, as shown in Figure 11. Click **Finish**.

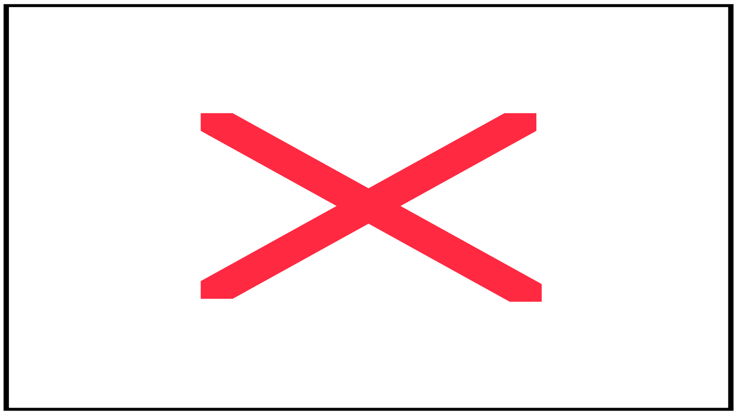
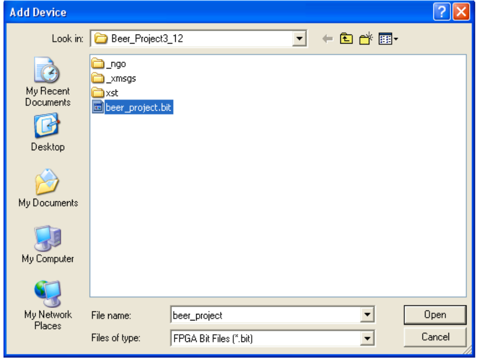


Figure 11: **Click Finish after Entering PROM Formatter Settings**

The PROM Formatter then prompts for the name(s) of the FPGA configuration bitstream file. As shown in Figure 12, click **OK** to start selecting files. Select an FPGA bitstream file (*.bit) (BeerModel.bit). Choose **No** after selecting the last FPGA file. Finally, click **OK** to continue.

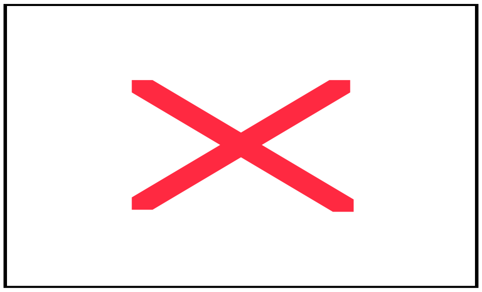


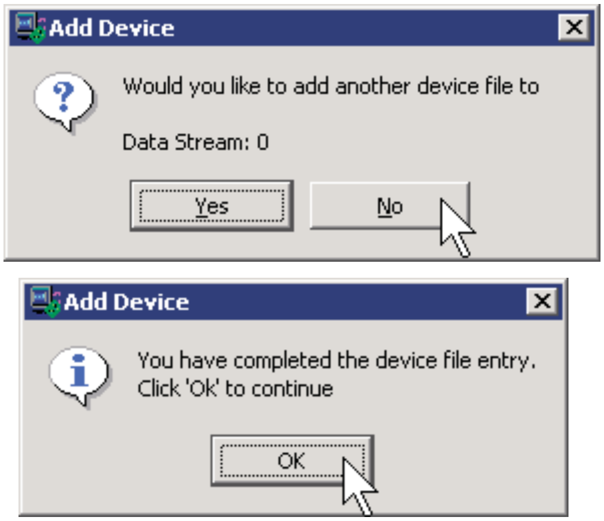
Add Device

Look in: Beer_Project3_12

File name: beer_project

Files of type: FPGA Bit Files (*.bit)





Would you like to add another device file to

Data Stream: 0

You have completed the device file entry. Click 'Ok' to continue

Figure 12

When PROM formatting is complete, the iMPACT software displays the present settings by showing the PROM, the select FPGA bitstream(s), and the amount of PROM space consumed by the bitstream. Figure 13 shows an example for a single XC3S500E FPGA bitstream stored in an XCF04S Platform Flash PROM.

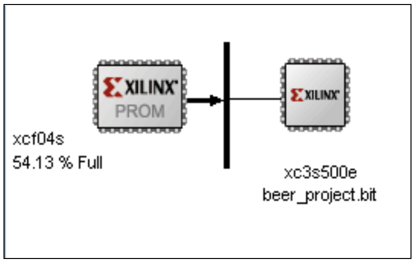


Figure 13: **PROM Formatting Completed**

To generate the actual PROM file, click **Operations** then **Generate File** as shown in Figure 14.

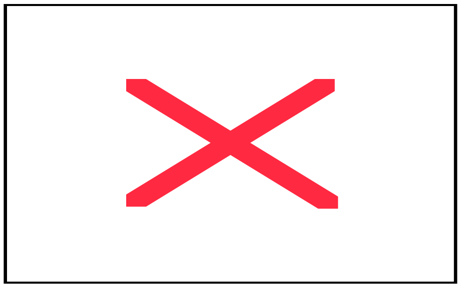


Figure 14: **Click Operations then Generate File to Create the Formatted PROM File**

The iMPACT software indicates that the PROM file was successfully created, as shown in Figure 15.

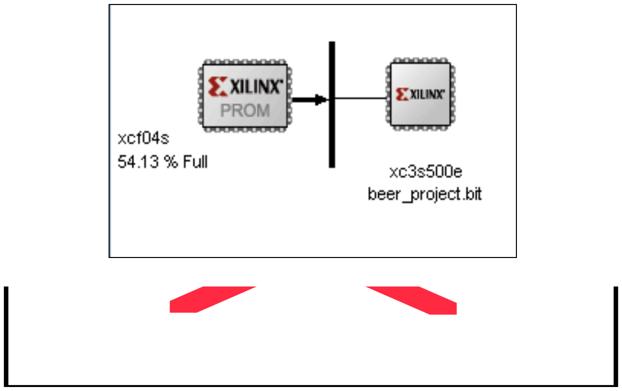


Figure 15: **PROM File Formatter Succeeded**

Program Beer Control into the FPGA Spartan 3 board Flash PROM via USB

Exit out of the Figure 15 screen, click on ***Process*** and ***Sources*** on the left hand side of the screen.

1. **Select the Sources tab** at the bottom of the screen page. **Click** to open **xc3.500e-4fg320** and select **Beer_Project-Behavior (BEER_PROJECT.vhd).**
2. Double click on **“Configure Device (iMPACT)”** under the implement design on the left hand side. If all goes well, you should get green check marks on **“Programming File Generation Report”** as shown Figure 16.

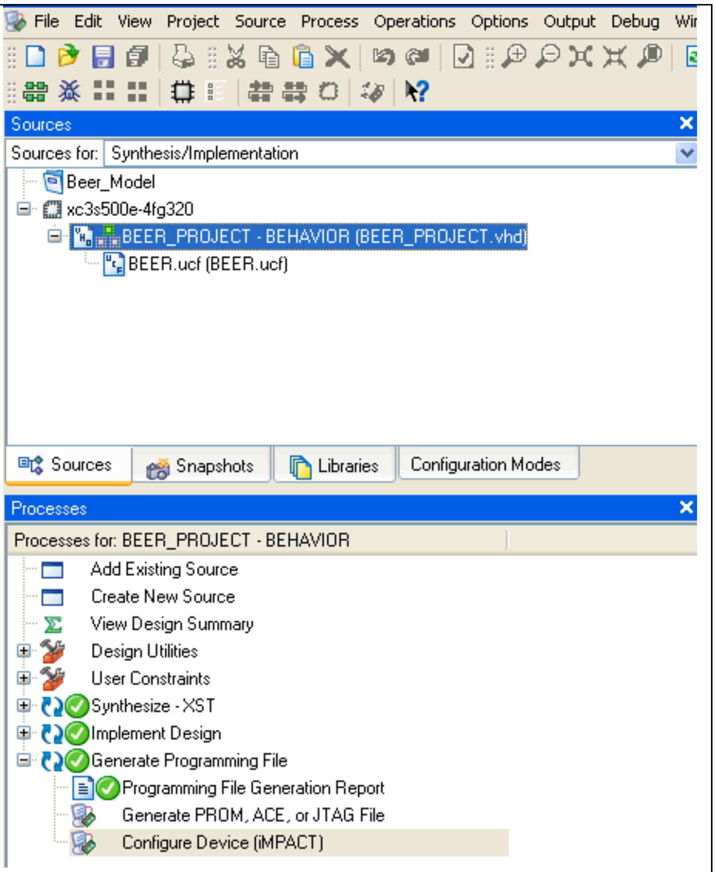
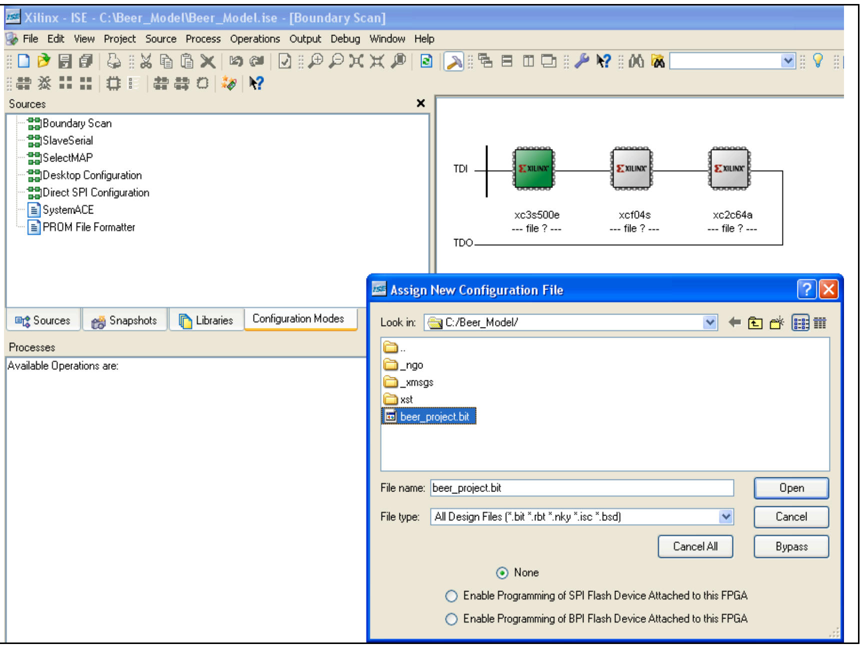

File Edit View Project Source Process Operations Options Output Debug Wi
Sources
Sources for: Synthesis/Implementation
Beer_Model
xc3s500e-4fg320
BEER_PROJECT - BEHAVIOR (BEER_PROJECT.vhd)
BEER.ucf (BEER.ucf)
Sources
Snapshots
Libraries
Configuration Modes
Processes
Processes for: BEER_PROJECT - BEHAVIOR
Add Existing Source
Create New Source
View Design Summary
Design Utilities
User Constraints
Synthesize - XST
Implement Design
Generate Programming File
Programming File Generation Report
Generate PROM, ACE, or JTAG File
Configure Device (iMPACT)

Figure 16: Generate Programming File

3. You should be prompted with the following screen.

As you can see in Figure 17 there are two devices listed in the window. One is xc3s500e FPGA and the xcf04s serial flash ROM. We will need to program only the (FPGA xc3s500e). We will do this by selecting **beer_project.bit** file from the window and click on open.


Xilinx - ISE - C:\Beer_Model\Beer_Model.ise - [Boundary Scan]

Look in: C:/Beer_Model/

File name: beer_project.bit

File type: All Design Files (*.bit *.rbt *.nky *.isc *.bsd)

● None

○ Enable Programming of SPI Flash Device Attached to this FPGA

○ Enable Programming of BPI Flash Device Attached to this FPGA

Figure 17: Beer_Project-bit

The second window is to program the flash ROM. Select **MyBeerFlash.mcs** and select open and then cancel the last window. Now you are ready to program your Spartan3E-Flash.

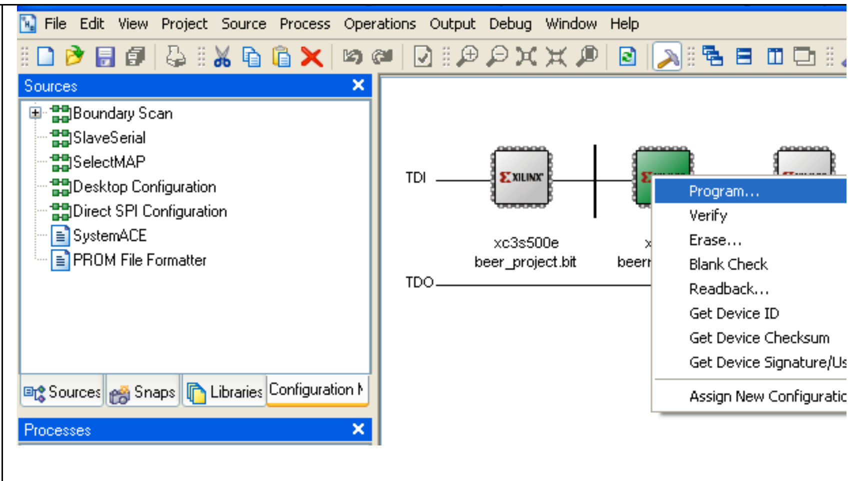
4. Right Click on the middle device (and select "**Program…**": as seen in Figure 18.	 File Edit View Project Source Process Operations Output Debug Window Help Sources Boundary Scan SlaveSerial SelectMAP Desktop Configuration Direct SPI Configuration SystemACE PROM File Formatter Sources Snaps Libraries Configuration Processes TDI TDO XILINX xc3s500e beer_project.bit beer Program... Verify Erase... Blank Check Readback... Get Device ID Get Device Checksum Get Device Signature/Us Assign New Configuratic

Figure 18: Program the board Flash

At this point the board flash should be programmed and ready for test.

Here is the User Constraint File "UCF" that maps inputs/outputs

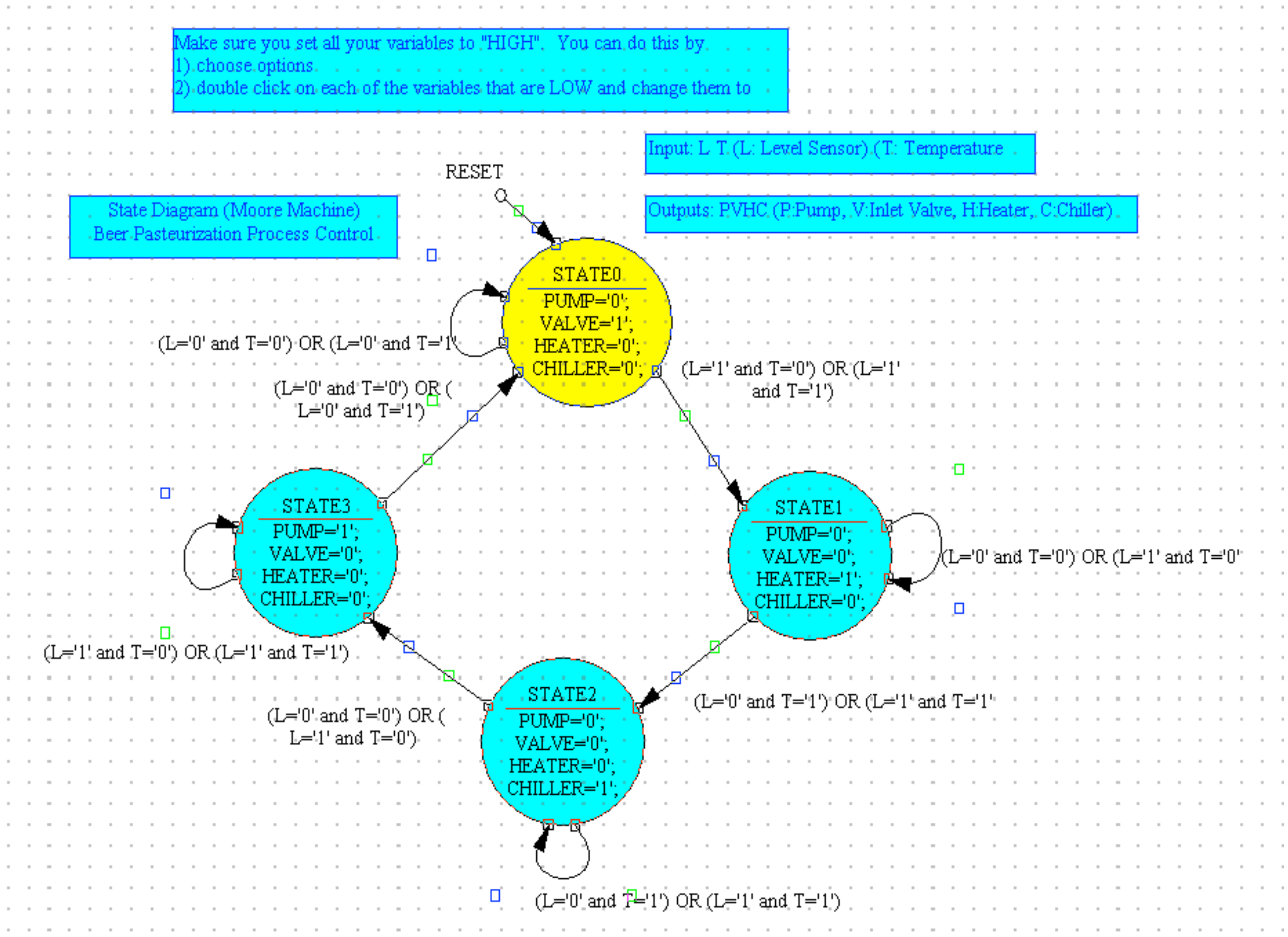
```
# Inputs
NET "L" LOC = "A4";              # LEVEL
NET "T" LOC = "B4";              # TEMPERATURE
NET "RESET" LOC = "D5";                # RESET
NET "CLK" LOC = "C9";            # Internal clock

# Outputs
NET "PUMP" LOC = "A6";           # PUMP
NET "VALVE" LOC = "B6";          # INLET VALVE
NET "HEATER" LOC = "F7";         # HEATER
NET "CHILLER" LOC = "E7";        # CHILLER
```

Demo your working hardware to your instructor and observe the difference between TTL, Xilinx schematic and VHDL implementation.

For instructors: Sources that are used in the lab

State Diagram that is used in the StateCAD editor of Xilinx:



Make sure you set all your variables to "HIGH". You can do this by
1) choose options
2) double click on each of the variables that are LOW and change them to

Input: L T (L: Level Sensor) (T: Temperature

State Diagram (Moore Machine)
Beer Pasteurization Process Control

Outputs: PVHC (P:Pump, V:Inlet Valve, H:Heater, C:Chiller)