ECE428 Xilinx ISE Tutorial

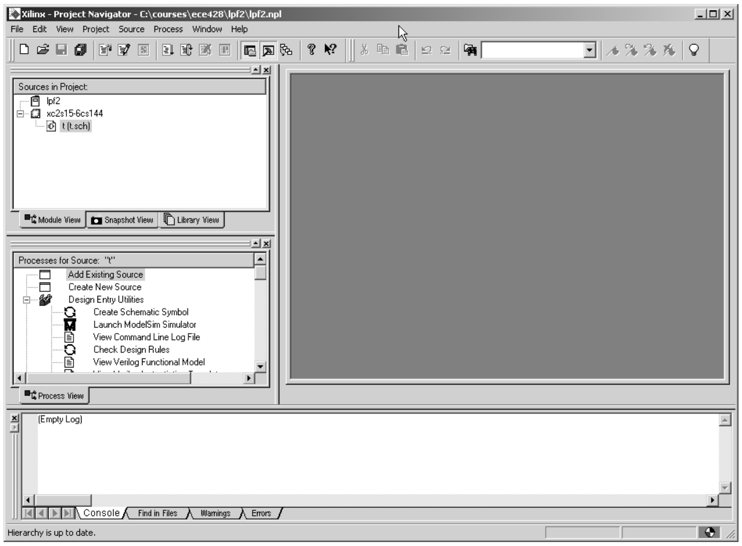
Haibo Wang
Southern Illinois University Carbondale

This tutorial explains the major steps in Xilinx ISE design flow. It consists of three sections. The first section describes how to enter a design through schematic capture, perform circuit simulation, assign pin locations, implement the design, and generate FPGA configuration data. Section 2 discuss how to specify timing constraints and perform static timing analysis. Section 3 explains how to use different design entry methods in a single design project.

1 Schematic-based FGPA Design flow

1.1 Creating a new project

Open Xilinx **Project Navigator** either from desktop icon or from windows Start Menu. The Project Navigator Window (PNW) is shown in Figure 1. It has four panels. The top panel in the left is the Source Panel that list all the design components (e.g. schematic, VHDL or Verilog code, and user constraint files) contained in the project. The middle pane in the left is the Process Panel. From this panel, users can start different design tasks (e.g. synthesize, implement, *etc.*) for selected design components. The bottom panel is the Transcript Panel for displaying ISE messages. The right panel in the Project Navigator Window is the Workspace Panel, which serves for difference purposes during the design process.

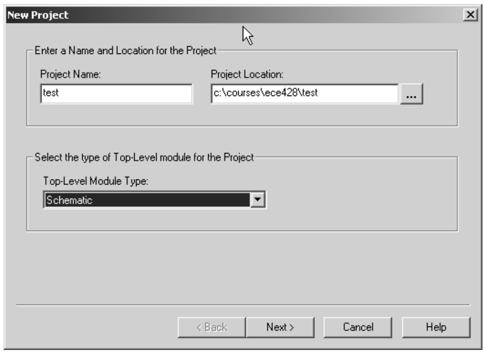


(Empty Log)

Hierarchy is up to date.

Figure 1. Project Navigator window.

From the Project Navigation Window, click **File** menu and select **New Project** to bring up a new window as shown in Figure 2 for specifying project name, directory, and the top level module type. After proper information is typed, click **Next** to bring up a new window as shown in Figure 3 to select FPGA devices (on which your design is intended to be implemented), synthesis tool, simulation tool (Modelsim), and simulation language. After this, three windows as shown in Figure 4, 5, 6 will pop up. The first and second windows are for adding new sources and adding already existing sources. The third window summaries the created design project. Next time when the Xilinx Project Navigator is opened, the designer can select **Open Project** from **File** menu to open the created project.



Enter a Name and Location for the Project

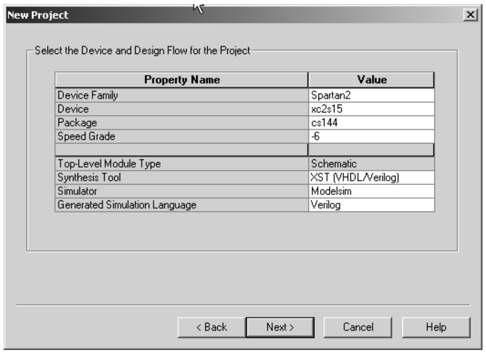
Project Name: test

Project Location: c:\courses\ece428\test

Select the type of Top-Level module for the Project

Top-Level Module Type: Schematic

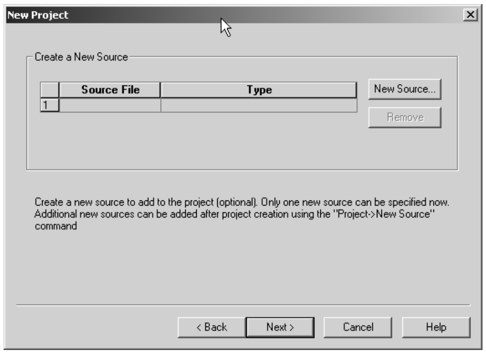
Figure 2. New project window.



Select the Device and Design Flow for the Project

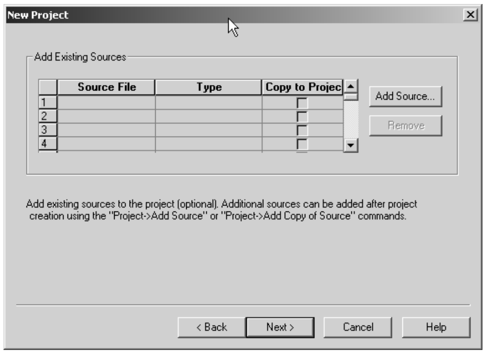
Property Name	Value
Device Family	Spartan2
Device	xc2s15
Package	cs144
Speed Grade	-6
Top-Level Module Type	Schematic
Synthesis Tool	XST (VHDL/Verilog)
Simulator	Modelsim
Generated Simulation Language	Verilog

Figure 3. Selecting FPGA device.



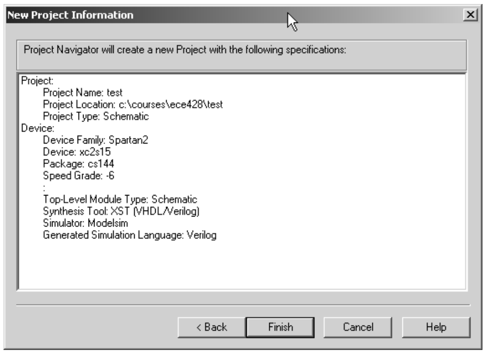
Create a new source to add to the project (optional). Only one new source can be specified now. Additional new sources can be added after project creation using the "Project->New Source" command

Figure 4. Adding new source to the project.



Add existing sources to the project (optional). Additional sources can be added after project creation using the "Project->Add Source" or "Project->Add Copy of Source" commands.

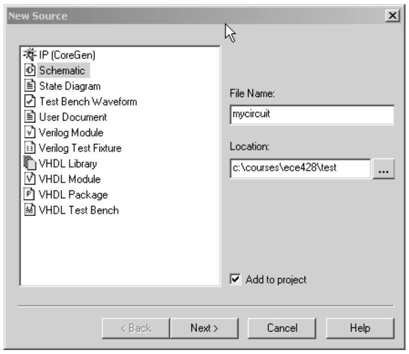
Figure 5. Adding existing source to the project.



Project Navigator will create a new Project with the following specifications:

```
Project:
      Project Name: test
      Project Location: c:\courses\ece428\test
      Project Type: Schematic
Device:
      Device Family: Spartan2
      Device: xc2s15
      Package: cs144
      Speed Grade: -6
      :
      Top-Level Module Type: Schematic
      Synthesis Tool: XST (VHDL/Verilog)
      Simulator: Modelsim
      Generated Simulation Language: Verilog
```

Figure 6. Project summary window.



File Name: mycircuit

Location: c:\courses\ece428\test

☑ Add to project

Figure 7. New source window.

1.2 Creating schematic

From the Project Navigator Window, click **Project** menu and select **New Source**. A window as shown in Figure 7 pops up. Select *schematic* from the left panel and type circuit name *mycircuit* in the right panel. The directory in which the schematic will be stored can be also changed in this window. After click **Next** and close the source summary window. An ECS window as shown in Figure 8 will pop up. In the left side of the ECS window, there are two overlapped panels, labeled by **Options** and **Symbols**. Click the tab of **Symbols** to bring the Symbol Panel to the front. The top of the Symbol Panel is the Category Window that lists all the available component libraries. The second window is the Symbol Window displaying component (symbol) names. Below these two windows are two blanks for symbol displaying filter and symbol orientation selection. The Symbol Information Button is located at the bottom of the Symbol Panel. Clicking this button will open the corresponding symbol description file which is in PDF format.

To place a component in the schematic area of the ECS window, first select the desired component name from the Symbol Window. Then, move the cursor (the symbol of the selected component will move with the cursor) to the desired location in the schematic area and left click the mouse to place it. To easily find desired components from the Symbol Window, the name of the desired component can be typed in the Symbol Name Filter blank to limit the symbol names

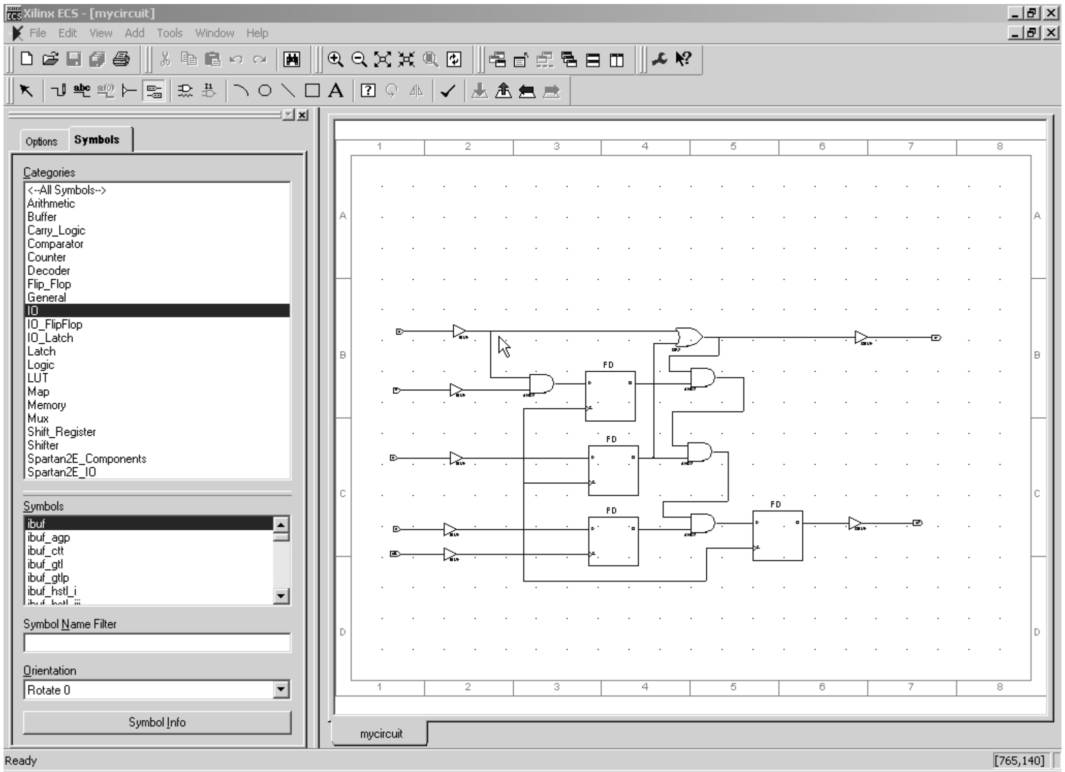


Figure 8. Xilinx ECS window and example schematic.

appearing in the Symbol Window. Place the following components in the schematic area as shown in Figure 8.

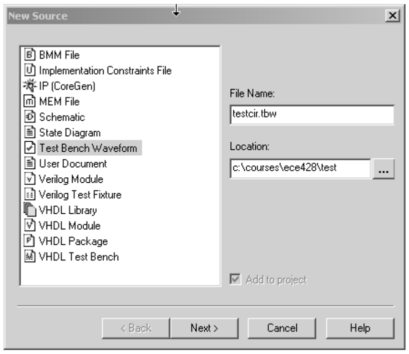
- **ibuf**, input buffer, five ibufs are located in the left side of the schematic
- **and2**, two-input AND gate
- **fd**, D Flip-Flop
- **or2**, two-input OR gate
- **obuf**, output buffer, two obufs are located in the right side of the schematic

Click **Add Wire** icon (the second icon from the left in the ECS window) to change the Xilinx tool into wiring mode. Move the mouse to connect the placed components as shown in Figure 8. Then, click **Add IO Marker** icon (the sixth icon from the left). From the Option Panel, select the type of IO markers before placing the markers into the schematic. Finally, save the design and close the ECS window. Schematic file *mycircuit.sch* should now be listed in the PNW Source Panel.

1.3 Simulating the created design

After drawing the schematic, the next task is to verify the function of the design through logic simulation. The current ISE tool in the ECE Department is configured to use ModelSim for logic simulation. To easily start the simulation process, a testbench can be created as follows.

From the Project Navigator window, click **Project** menu and select **New Source**. Configure the new source window as shown in Figure 9. In the next window, select *mycircuit* as shown in Figure 10. After these steps, a testbench summary window is displayed. Closing the testbench summary window, a timing initialization window as shown in Figure 11 will appear for specifying setup time, maximum delay as well as clock period and duty cycle (by specifying clock high time and low time). Thereafter, a waveform window as shown in Figure 12 will be displayed. Use the mouse to toggle input signal values. Finally, right click the mouse to bring up a pop-up menu and select **Set end of testbench**. This will set the time that the simulation will stop. In the waveform window, it is represented by a vertical blue time. The position of the blue vertical line (also the simulation stop time) can be change with using the mouse. After save the testbench waveform, testbench file *testcir.tbw* should be listed in the PNW Source Panel.



File Name: testcir.tbw

Location: c:\courses\ece428\test

Figure 9. New source window for testbench file.

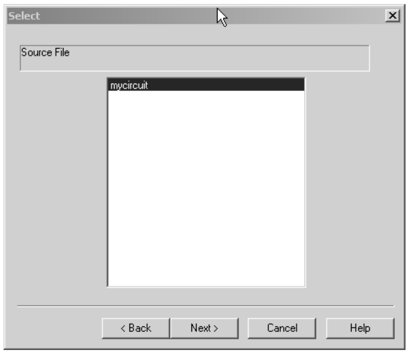
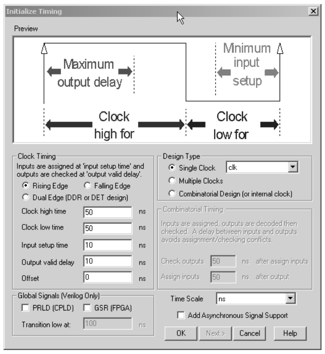


Figure 10. Selecting circuit to be tested.



Clock Timing

Inputs are assigned at 'input setup time' and outputs are checked at 'output valid delay'.

● Rising Edge ○ Falling Edge

○ Dual Edge (DDR or DET design)

Clock high time 50 ns

Clock low time 50 ns

Input setup time 10 ns

Output valid delay 10 ns

Offset 0 ns

Design Type

● Single Clock clk

○ Multiple Clocks

○ Combinatorial Design (or internal clock)

Combinatorial Timing

Inputs are assigned, outputs are decoded then checked. A delay between inputs and outputs avoids assignment/checking conflicts.

Check outputs 50 ns after assign inputs

Assign inputs 50 ns after output

Global Signals (Verilog Only)

□ PRLD (CPLD) □ GSR (FPGA)

Transition low at: 100 ns

Time Scale ns

□ Add Asynchronous Signal Support

Figure 11. Timing initialization window.

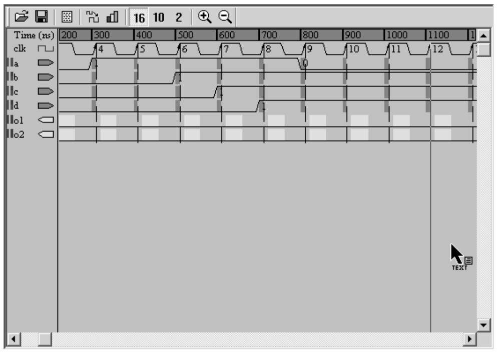
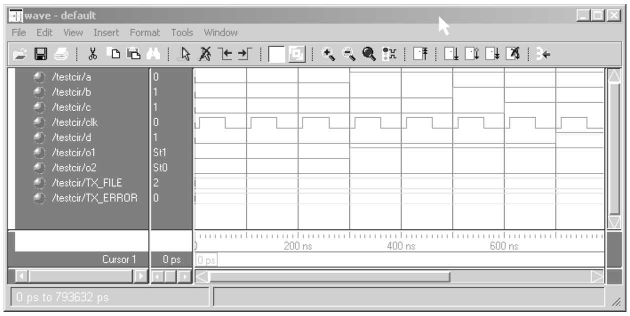


Figure 12. Waveform editing window.

In the PNW Source Panel, highlight testbench file *testcir.tbw*. In the Process Panel, double click **View Behavioral Test Fixture** (or **VHDL testbench** if VHDL is selected in the project) to see the corresponding HDL code of the created testbench. Double click the ”+” sign in front of **ModelSim Simulator** to display more options(if they are not displayed yet). Double click **Simulate Behavioral Model** to invoke ModeSim. The ModelSim HDL simulation will be automatically executed according to the testbench file and the simulation result will be displayed in a waveform window as shown in Figure 13



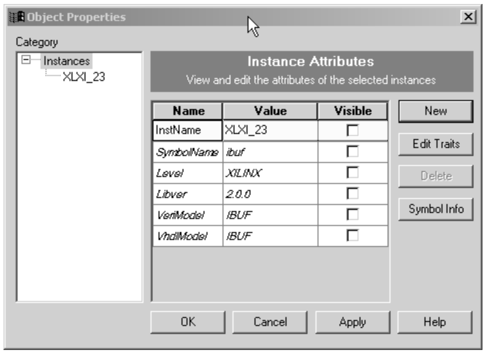
```
/testcir/a          0
/testcir/b          1
/testcir/c          1
/testcir/clk        0
/testcir/d          1
/testcir/o1         St1
/testcir/o2         St0
/testcir/TX_FILE    2
/testcir/TX_ERROR   0
Cursor 1            0 ps
0 ps to 793632 ps
```

Figure 13. ModelSim simulation result.

1.4 Specifying pin locations

There are multiple places in Xilinx ISE design flow that allow users to specify pin locations. In the circuit schematic, the designer can move cursor to select a input (or output) buffer and right click the mouse to bring up a pop-up menu. Select **Object Properties**. The Object Property Window pops up as shown in Figure 14. Click **New** to bring up the New Property Window as shown in Figure 15. Enter loc for the **Attribute Name** and A5 for the **Attribute Value**.

Another method to assign pin locations is through user constraint file. In the PNW Source Panel, highlight schematic file *mycircuit.sch*. In the Process Panel, click the ”+” sign in front of *User Constraints* to see more options. Double click



Object Properties

Category

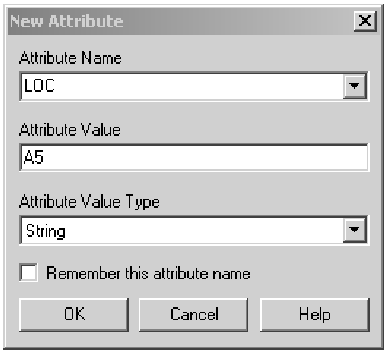
Instances
XLXI_23

Instance Attributes

View and edit the attributes of the selected instances

Name	Value	Visible
InstName	XLXI_23	□
SymbolName	*ibuf*	□
Level	*XILINX*	□
Libver	*2.0.0*	□
VeriModel	*IBUF*	□
VhdlModel	*IBUF*	□

Figure 14. Object property window.



New Attribute

Attribute Name: LOC

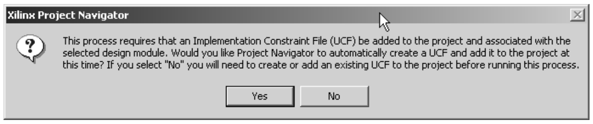
Attribute Value: A5

Attribute Value Type: String

□ Remember this attribute name

Figure 15. Adding location property.

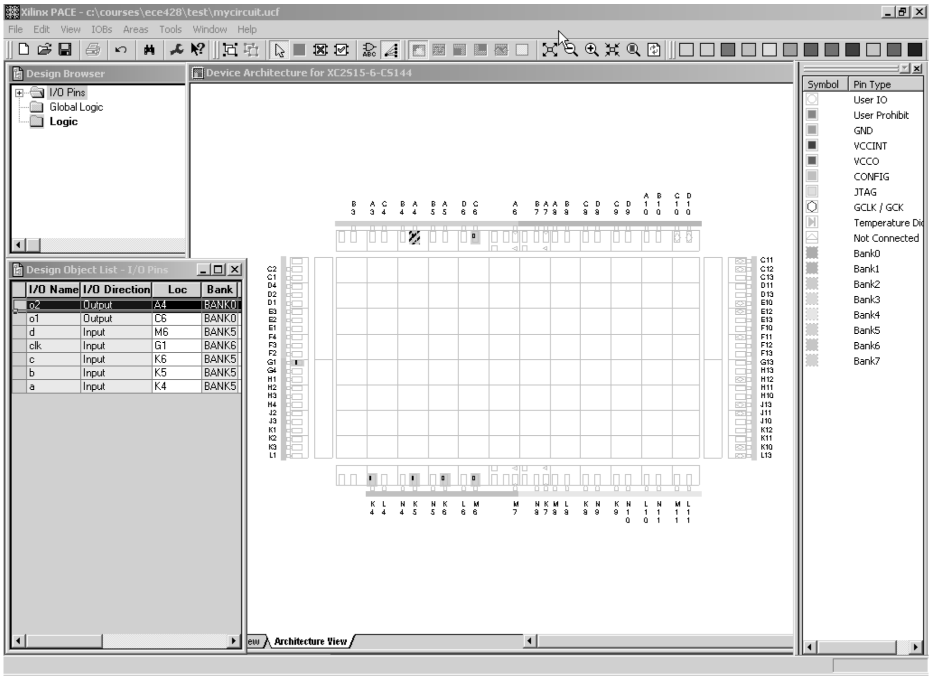
Assign Package Pins. It brings up a window as shown in Figure 16. Click **Yes**. Xilinx PACE window will pop up as shown in Figure 17. The PACE window consists of Design Browser Panel, Design Object List Panel, Device Architecture Panel, and Pin Type Panel. From the Design Object List Panel, select a pin, drag it to the Device Architecture Panel, and place it to the desired device pin location. The corresponding pin location will be displayed in the Design Object Panel. Repeat the above process for all the pins. Save the setup and close the PACE window. User constraint file *mycircuit.ucf* should now be listed in the PNW Source Panel.



Xilinx Project Navigator

This process requires that an Implementation Constraint File (UCF) be added to the project and associated with the selected design module. Would you like Project Navigator to automatically create a UCF and add it to the project at this time? If you select "No" you will need to create or add an existing UCF to the project before running this process.

Figure 16. Creating user constraint file.



Xilinx PACE - c:\courses\ece428\test\mycircuit.ucf

Design Browser: I/O Pins, Global Logic, Logic

Device Architecture for XC2S15-6-CS144

Design Object List - I/O Pins

I/O Name	I/O Direction	Loc	Bank
o2	Output	A4	BANK0
o1	Output	C6	BANK0
d	Input	M6	BANK5
clk	Input	G1	BANK6
c	Input	K6	BANK5
b	Input	K5	BANK5
a	Input	K4	BANK5

Figure 17. Xilinx PACE window.

In the PACE window, users can also specify area constraints. Please refer to Xilinx User manuals for more detail information. In the PNW Process Panel, double clicking **Edit Constraints (Text)** will open a text window to display user constraints in text format. For large designs, it is more convenient to specify users constraints (pin assignments, area, and timing) through editing this text user constraint file.

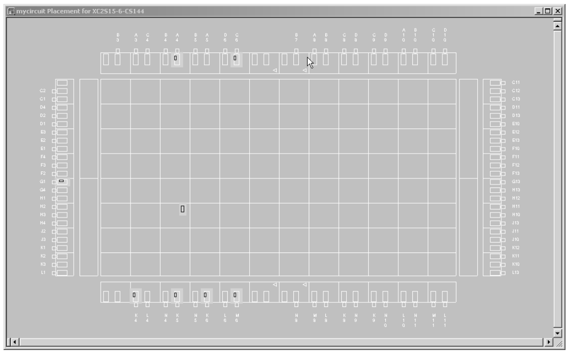


Figure 18. Circuit placement result.

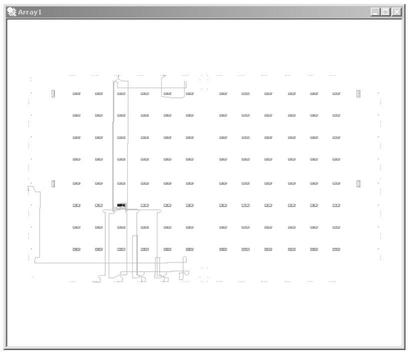


Figure 19. Circuit routing result.

1.5 Design synthesis

In the PNW Source Panel, highlight schematic file *mycircuit.sch*. In the Process Panel, double click **Synthesize-XST** to perform design synthesis. After the synthesis process, green marks appear to indicate the success of the synthesis process. Yellow marks indicate the existence of warning messages. Red marks indicate the occurrence of errors. Both warning and error messages will be displayed in the PNW Transcript Panel. Click the ”+” sign in front of **Synthesize-XST** to see more options. Double clicking **View Synthesis Report** will bring up a text window that summarizes the synthesis result, which includes device usage and timing information. Note that the timing information reported here does not consider effects of placement and routing parasitics. In addition to synthesis report, the synthesized schematic can also be viewed by double clicking **View RTL Schematic**. This is a useful feature when performing static timing analysis as described later.

1.6 Design Implementation

The Xilinx ISE implementation process consists of three major steps: *Translate*, *Map*, and *Placement and Routing (P&R)*. According to Xilinx user manual, the difference between Translate and Map is described as follows. ”Translate consists of a number of various programs that are used to import the design netlist and prepare it for layout. Mapping is the process of assigning design logic elements to the specific physical elements that actually implement logic functions in a device.” The mapped physical elements are placed and routed in the final P&R phase. In the ISE tool, users can simply double click **Implement Design** to let the tool automatically complete the above three steps. However, advanced features can be used in the implementation process to have more control over the final outcome.

After the implementation process is complete, double clicking **View/Edit Placed Design** or **View/Edit Routed Design** (Under **Place & Route** group in the PNW Process Panel) will open new windows to display the placement and routing results as shown in Figure 18 and 19, respectively. Also, the designer can double click **Generate Post-Place & Route Simulation mode** to create HDL netlist for post-layout simulation. After the netlist is generated, highlight testbench file *testcir.tbw* in the PNW Source Panel. Double click **Simulate Post-Place & Route Verilog Model** in the Process Panel to start simulation. Signals delayed will be observed in the simulation result because interconnect parasitics are considered in simulation.

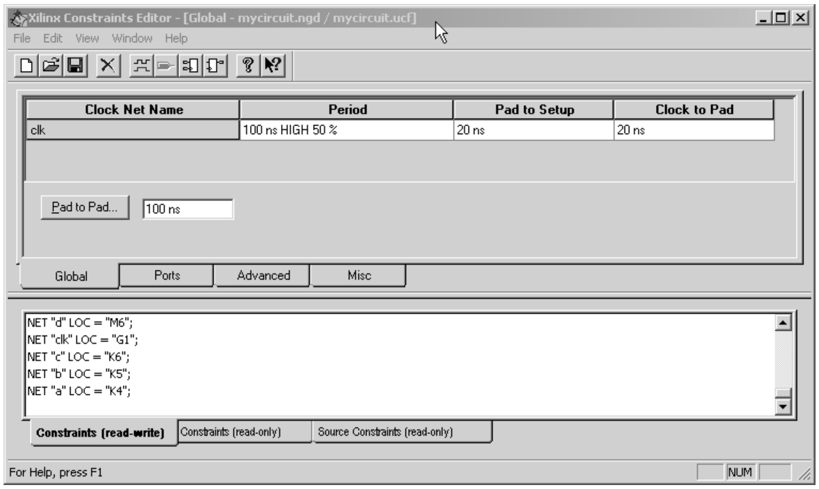
1.7 Generating configuration file

In the PNW Process Panel, double click **Generate Programming File**. After this process is complete, FPGA bit file *mycircuit.bit* will be stored in the project directory.

2 Working with Timing Constraints

2.1 Specifying timing constraints

Timing constraints are used to guide how designs are implemented (including translate, map, place & route) in order to meet desired performance. Timing constraints that are often used in Xilinx design flow include: **clock period, pad to setup time (offset in), clock to pad delay (offset out)**, and **pad to pad delay**. To specify timing constraints for the example design *mycircuit*, double click **Create Timing Constraints** (under **User Constraints** in the Process Panel). The timing constraint window pops up as shown in Figure 20. Type 100 at **period** and **Pad to Pad** blanks. Type 20 at **Pad to Setup** and **Clock to Pad** blanks. After saving timing constraints, the design needs to be re-implemented according to the specified timing constraints. If there are constraints that cannot be satisfied, error messages will be displayed in the PNW Transcript Panel and the implementation process cannot be completed. In such cases, the designer has to modify your design or re-consider if the timing constraints are feasible.


Xilinx Constraints Editor - [Global - mycircuit.ngd / mycircuit.ucf]
File Edit View Window Help
Pad to Pad...
Global
Ports
Advanced
Misc
Constraints (read-write)
Constraints (read-only)
Source Constraints (read-only)
For Help, press F1
NUM

Clock Net Name	Period	Pad to Setup	Clock to Pad
clk	100 ns HIGH 50 %	20 ns	20 ns

Pad to Pad...: 100 ns

```
NET "d" LOC = "M6";
NET "clk" LOC = "G1";
NET "c" LOC = "K6";
NET "b" LOC = "K5";
NET "a" LOC = "K4";
```

Figure 20. Specifying timing constraints.

2.2 Performing static timing analysis

After mapping or placement and routing process is complete, static timing analysis can be performed. The analysis result indicates what is the maximum frequency that the circuit can operate. It also reveals how the circuit can be modified for further performance improvement. In the PNW Process Panel, double click **Analyze Post-Map Static Timing (Timing Analyzer)** in the **Map** group (or **Analyze Post-Place & Route Static Timing (Timing Analyzer)** in the **Place & Route** group). The Timing Analyzer window appears as shown in Figure 21. Moving cursor around icons, their corresponding icon names will appear.

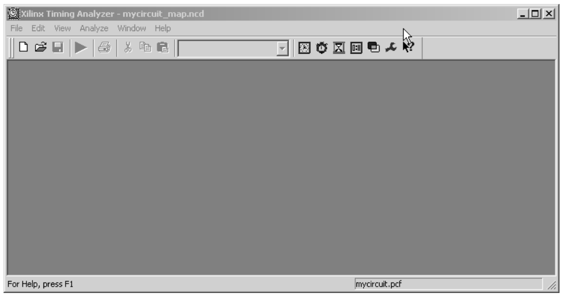

Xilinx Timing Analyzer - mycircuit_map.ncd
File Edit View Analyze Window Help
For Help, press F1
mycircuit.pcf

Figure 21. Timing Analyzer window.

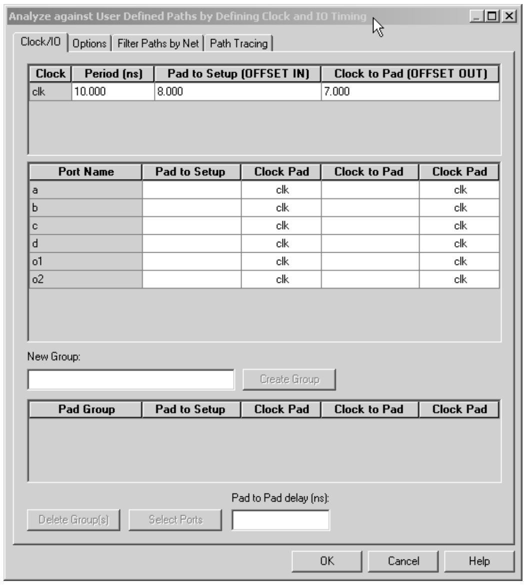
Click **Analyze against timing constraints** icon to generate a static timing report. Figure 22 shows a portion of the static timing report. The first text line states the timing constraint that is verified in the following portion of the report. The second line summarizes how many paths are examined and how many errors are detected. The lines started with *Source* and *Destination* indicate the starting and ending points (DFFs in this case, they can also be pads) of the currently reported path. To find the corresponding path in the synthesized schematic, open the synthesized RTL schematic (double click **View RTL Schematic** in the **Synthesize** group). The symbol of the synthesized circuit will first appear in the window. Double click it to descend to the low-level schematic. In the left Instance Content Panel, click the ”+” sign in front of **Instance** and select **XLXI_4**. The corresponding component will be highlighted in the schematic window. Similarly, the source element **XLXI_2** of the example path can be found.

The static timing report will list the minimum period and the maximum delays. Therefore, the maximum operating speed of the design can be estimated. In addition, static timing analysis can be performed with stringent timing constraints to see which path will fail first. (hence, the designer will know the critical path that need to be modified in order to further improve circuit performance.) Double click **Analyze Against User Specified Paths by Defining Clock and IO timing**

```
. . .
Timing constraint: TS_clk = PERIOD TIMEGRP "clk" 10 nS HIGH 50.000000 %;
3 items analyzed, 0 timing errors detected. (0 setup errors, 0 hold errors)
Minimum period is 2.484ns.
. . .
Slack: 7.516ns (requirement - (data path - clock path skew + uncertainty))
Source: XLXI_2 (FF)
Destination: XLXI_4 (FF)
Requirement: 10.000ns
Data Path Delay: 2.484ns (Levels of Logic = 1)
. . .
```

Figure 22. Static timing report.

icon. Configure the resulted window as shown in Figure 23 and click **OK**. A new timing analysis report will be generated. It will be observed that a timing violation occurs in an output path (from **XLXI_2** to **o1**).



Clock	Period (ns)	Pad to Setup (OFFSET IN)	Clock to Pad (OFFSET OUT)
clk	10.000	8.000	7.000

Port Name	Pad to Setup	Clock Pad	Clock to Pad	Clock Pad
a		clk		clk
b		clk		clk
c		clk		clk
d		clk		clk
o1		clk		clk
o2		clk		clk

Pad Group	Pad to Setup	Clock Pad	Clock to Pad	Clock Pad

Figure 23. Specifying new constraints for timing analysis .

2.3 Performing dynamic timing verification

In addition to static timing analysis, you also need to perform dynamic timing verification. This is done by performing **post-place & route simulation** (refer to Section 1.6).

3 Working with Different Design Entry Methods

Beside the schematic entry method, design components can be entered into the project by writing HDL (VHDL or Verilog) codes and drawing state diagrams. As a general approach, the top level design is usually entered using schematic method. Thereafter, the partitioned function blocks are designed by writing HDL codes. For finite state machines (FSMs), sometimes it is more convenient to draw state diagrams. The following example illustrates how to use these design approaches in a single design project. The function of the example circuit is to count how many "101" patterns appear in a serially transmitted data stream at a given time window. This simple circuit is partitioned into four functional blocks: reset delay circuit, counter circuit, pattern detection circuit, and register circuit. The implementation of these sub circuits are described as follows.

3.1 Reset delay circuit (rstgen)

Following the method discussed early, draw the schematic of the reset delay circuit as shown in Figure 24. Since the input and output signals of this circuit are internal signals, input and output buffers are not needed in the schematic. However, IO markers are still needed as shown in the schematic. After save the schematic, in the Process Panel double click **Create Schematic Symbol** to create a symbol for this circuit.

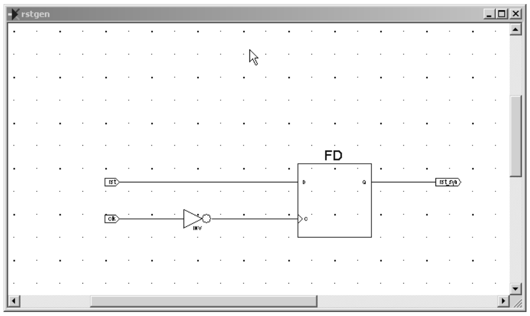
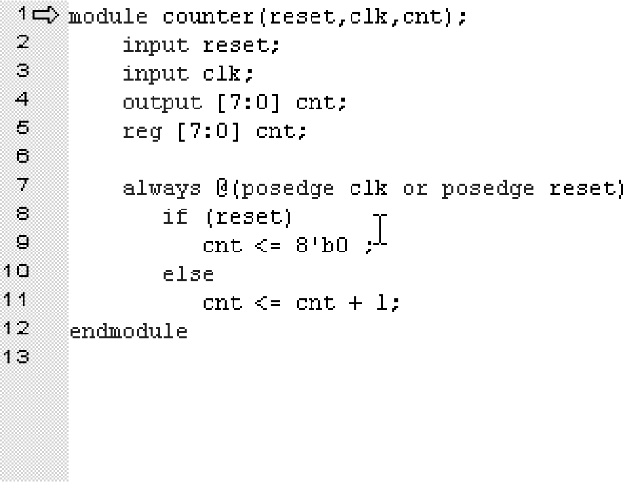


Figure 24. Reset delay circuit.

3.2 Counter Circuit

From the Project Navigator Window, click **Project** menu and select **New Source**. In the New Source Window, select *Verilog Module* and specify the file name as *counter*. In the Verilog edit window, type the code as shown in Figure 25. After save the code, create its schematic symbol as discussed early.



```
module counter(reset,clk,cnt);
    input reset;
    input clk;
    output [7:0] cnt;
    reg [7:0] cnt;

    always @(posedge clk or posedge reset)
       if (reset)
          cnt <= 8'b0 ;
       else
          cnt <= cnt + 1;
endmodule
```

Figure 25. Verilog code of a 8-bit counter.

3.3 Pattern detection circuit

From the Project Navigator Window, click **Project** Menu and select **New Source**. In the New Source Window, select **State Diagram** and specify the file name as *patdect*. In the opened **StateCAD** window, click **Help** and select **Tutorial**. Read the tutorial to learn how to draw state diagrams. Then draw the *patdect* state diagram as shown in Figure 26. (Its function is to detect 101 pattern. Once a pattern is found, the output of the FSM becomes logic 1 for one clock cycle.) Click **Optimization** and **generate HDL** icons to create Verilog code for the designed FSM. In the Project Navigator Window, click **Project** and select **Add Source** to add the created Verilog code into the project. Finally, create the schematic symbol for the FSM.

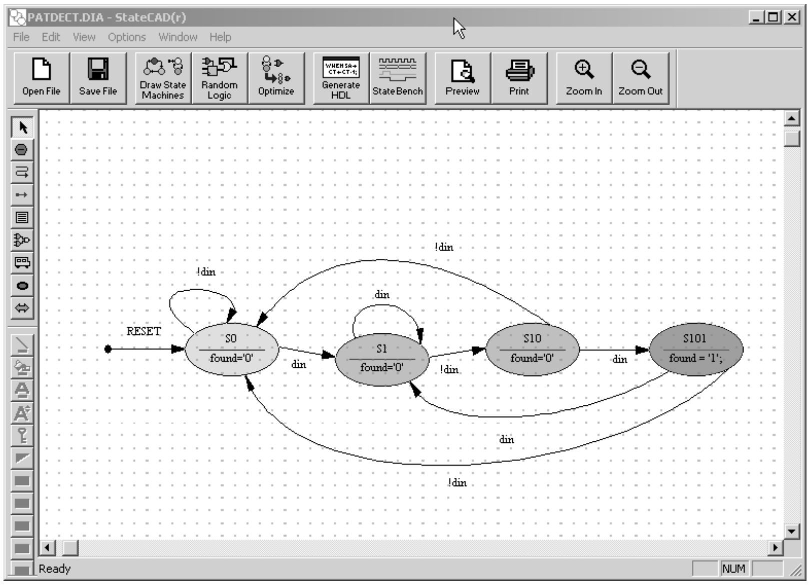


Figure 26. State diagram of the pattern detection circuit.

3.4 Top level circuit

The top view of the design is shown in Figure 27. After this schematic is complete, the entire design can be synthesized and implemented.

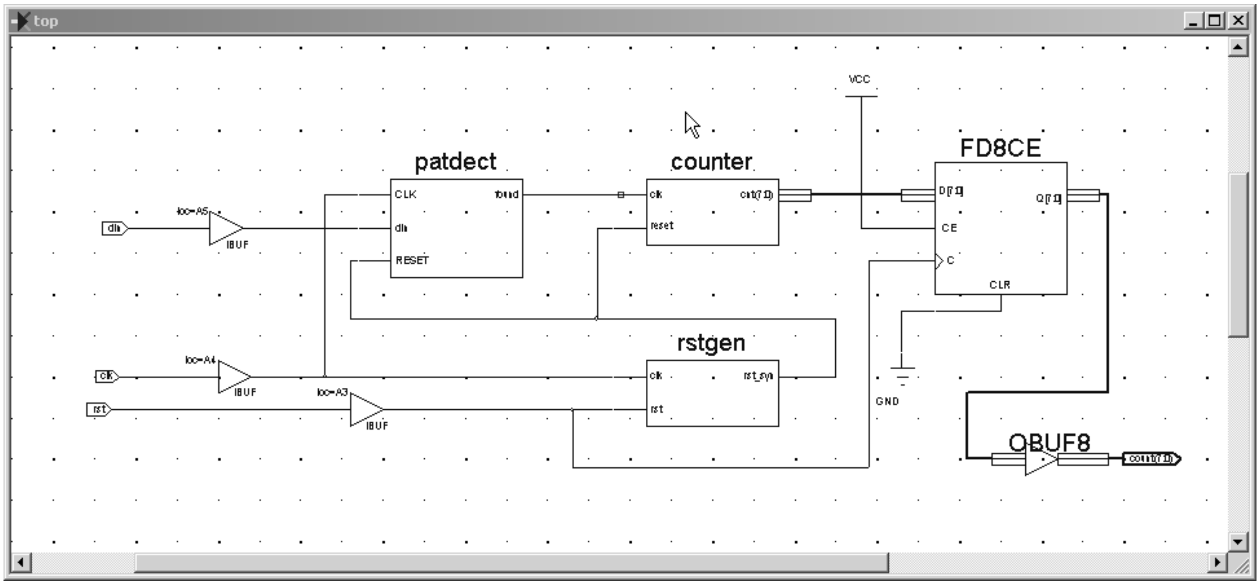


Figure 27. Top level schematic.