

EE4313 COMPUTER ENGINEERING DESIGN PROJECT - I



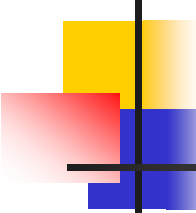
Lab 1

Lab 1: Data Dependent Rotations

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY leftrotate IS
    PORT (
        a      : in STD_LOGIC_VECTOR(31 DOWNTO 0);
        b      : in STD_LOGIC_VECTOR(31 DOWNTO 0);
        key    : in STD_LOGIC_VECTOR(31 DOWNTO 0);
        o      : out STD_LOGIC_VECTOR(31 DOWNTO 0));
END leftrotate;

ARCHITECTURE rtl OF leftrotate IS
    SIGNAL ab_xor : STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL ab_rot : STD_LOGIC_VECTOR(31 DOWNTO 0);
BEGIN
    ab_xor <= a XOR b;

    WITH b(4 DOWNTO 0) SELECT
        ab_rot <= ab_xor(30 DOWNTO 0) & ab_xor(31) WHEN "00001",
```


$$0 = (A \text{ XOR } B) \ll \ll B + \text{SKEY}$$

```
ab_xor(29 DOWNT0 0) & ab_xor(31 DOWNT0 30) WHEN "00010",
ab_xor(28 DOWNT0 0) & ab_xor(31 DOWNT0 29) WHEN "00011",
ab_xor(27 DOWNT0 0) & ab_xor(31 DOWNT0 28) WHEN "00100",
ab_xor(26 DOWNT0 0) & ab_xor(31 DOWNT0 27) WHEN "00101",
ab_xor(25 DOWNT0 0) & ab_xor(31 DOWNT0 26) WHEN "00110",
ab_xor(24 DOWNT0 0) & ab_xor(31 DOWNT0 25) WHEN "00111",
ab_xor(23 DOWNT0 0) & ab_xor(31 DOWNT0 24) WHEN "01000",
ab_xor(22 DOWNT0 0) & ab_xor(31 DOWNT0 23) WHEN "01001",
ab_xor(21 DOWNT0 0) & ab_xor(31 DOWNT0 22) WHEN "01010",
ab_xor(20 DOWNT0 0) & ab_xor(31 DOWNT0 21) WHEN "01011",
ab_xor(19 DOWNT0 0) & ab_xor(31 DOWNT0 20) WHEN "01100",
ab_xor(18 DOWNT0 0) & ab_xor(31 DOWNT0 19) WHEN "01101",
ab_xor(17 DOWNT0 0) & ab_xor(31 DOWNT0 18) WHEN "01110",
ab_xor(16 DOWNT0 0) & ab_xor(31 DOWNT0 17) WHEN "01111",
ab_xor(15 DOWNT0 0) & ab_xor(31 DOWNT0 16) WHEN "10000",
ab_xor(14 DOWNT0 0) & ab_xor(31 DOWNT0 15) WHEN "10001",
ab_xor(13 DOWNT0 0) & ab_xor(31 DOWNT0 14) WHEN "10010",
```


$$O = (A \text{ XOR } B) \ll\ll B + \text{SKEY}$$

```
ab_xor(12 DOWNT0 0) & ab_xor(31 DOWNT0 13) WHEN "10011",  
ab_xor(11 DOWNT0 0) & ab_xor(31 DOWNT0 12) WHEN "10100",  
ab_xor(10 DOWNT0 0) & ab_xor(31 DOWNT0 11) WHEN "10101",  
ab_xor(9 DOWNT0 0) & ab_xor(31 DOWNT0 10) WHEN "10110",  
ab_xor(8 DOWNT0 0) & ab_xor(31 DOWNT0 9) WHEN "10111",  
ab_xor(7 DOWNT0 0) & ab_xor(31 DOWNT0 8) WHEN "11000",  
ab_xor(6 DOWNT0 0) & ab_xor(31 DOWNT0 7) WHEN "11001",  
ab_xor(5 DOWNT0 0) & ab_xor(31 DOWNT0 6) WHEN "11010",  
ab_xor(4 DOWNT0 0) & ab_xor(31 DOWNT0 5) WHEN "11011",  
ab_xor(3 DOWNT0 0) & ab_xor(31 DOWNT0 4) WHEN "11100",  
ab_xor(2 DOWNT0 0) & ab_xor(31 DOWNT0 3) WHEN "11101",  
ab_xor(1 DOWNT0 0) & ab_xor(31 DOWNT0 2) WHEN "11110",  
ab_xor(0) & ab_xor(31 DOWNT0 1) WHEN "11111",  
ab_xor WHEN OTHERS;
```

```
o<=ab_rot + skey;  
END rtl;
```



Exercise 1

- Simulate Left Rotate VHDL Model
 - Functional Simulation using ModelSim
- Synthesize the Design
 - Perform Timing Simulation
- Draw and Discuss the Synthesized Circuit
- Understand use of “select” VHDL statement



Functional Simulation

- Create project folder
 - C:\ee4313\proj1
- Save your VHDL code in project folder
 - leftrotate.vhd
- Start ModelSim

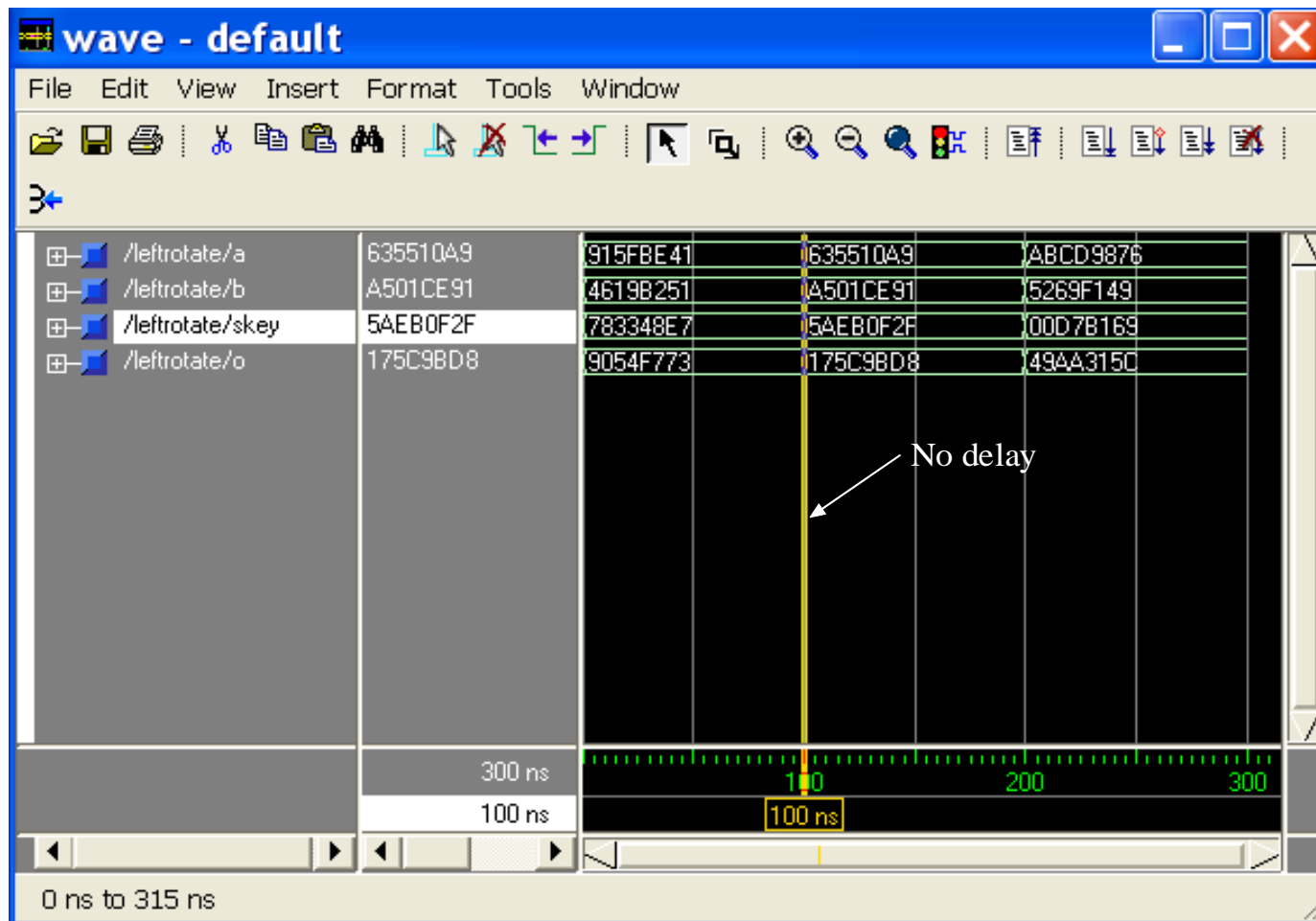


Functional Simulation

- In ModelSim console window, type following commands in sequence

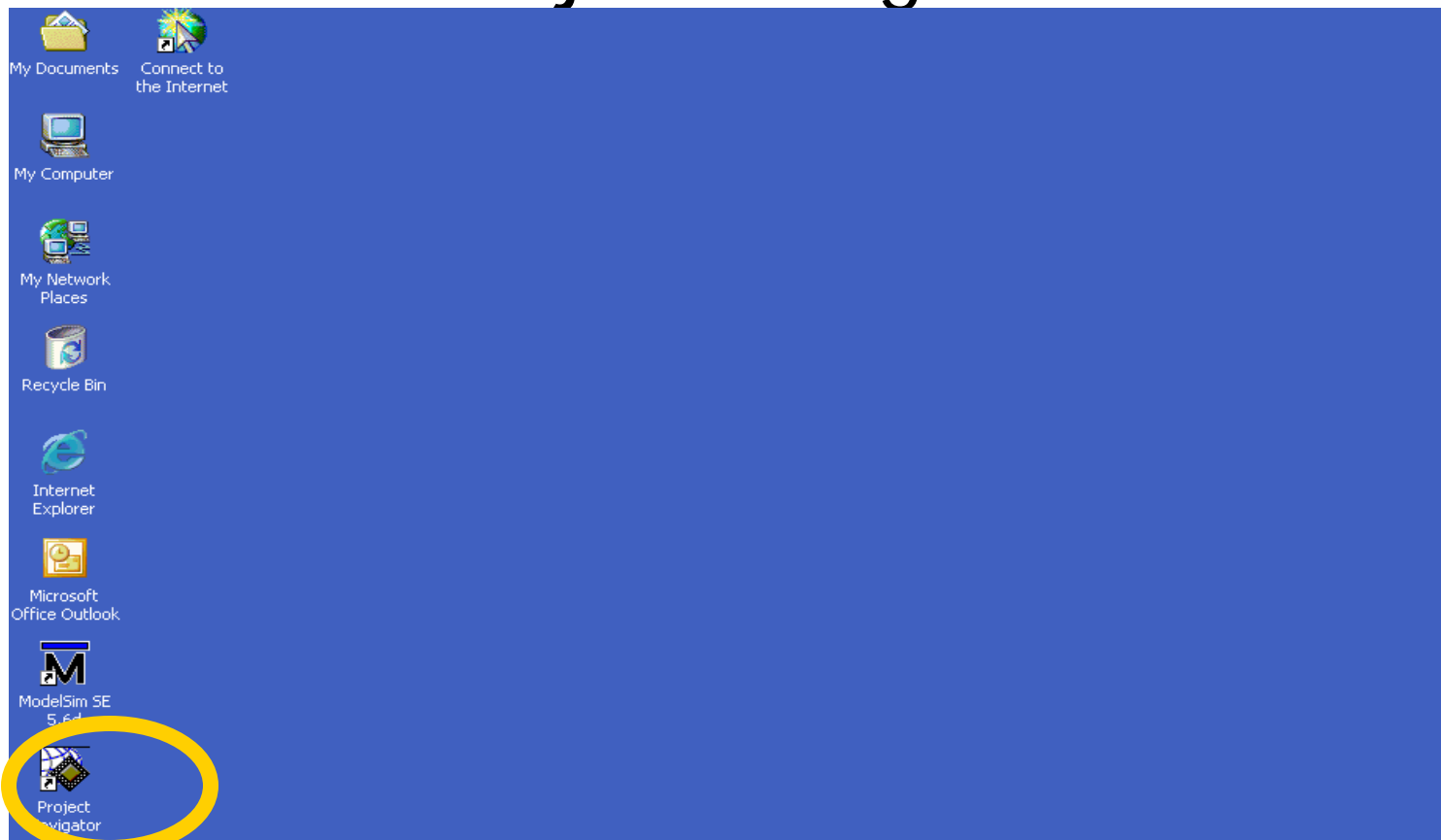
```
cd {c:/ee4313/proj1}
vlib work                -- create a work library
vcom leftrotate.vhd     -- "compile" your design
vsim leftrotate         -- simulate your design
view wave              -- start waveform viewer
add wave a b key o     -- add input/out signals in the waveform viewer
-- assign input signals (use hexadecimal, time unit ns)
force a 16#915FBE41 0, 16#635510A9 100, 16#ABCD9876 200
force b 16#4619B251 0, 16#A501CE91 100, 16#5269F149 200
force key 16#783348E7 0, 16#5AEB0F2F 100, 16#D7B169 200
run 300                 -- simulate 300ns
```

Functional Simulation



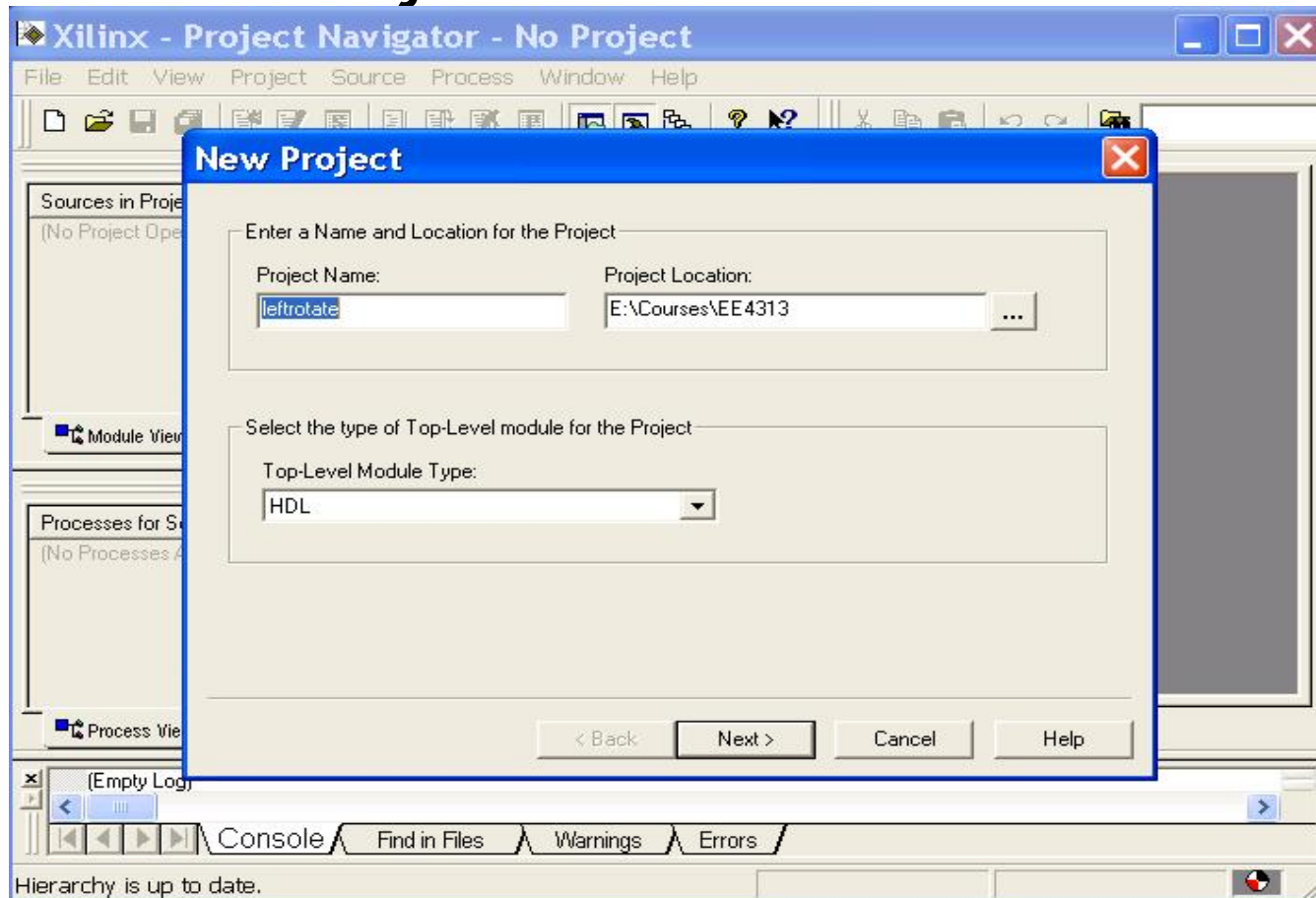
FPGA Synthesis and Implementation

- Start Xilinx Project Navigator



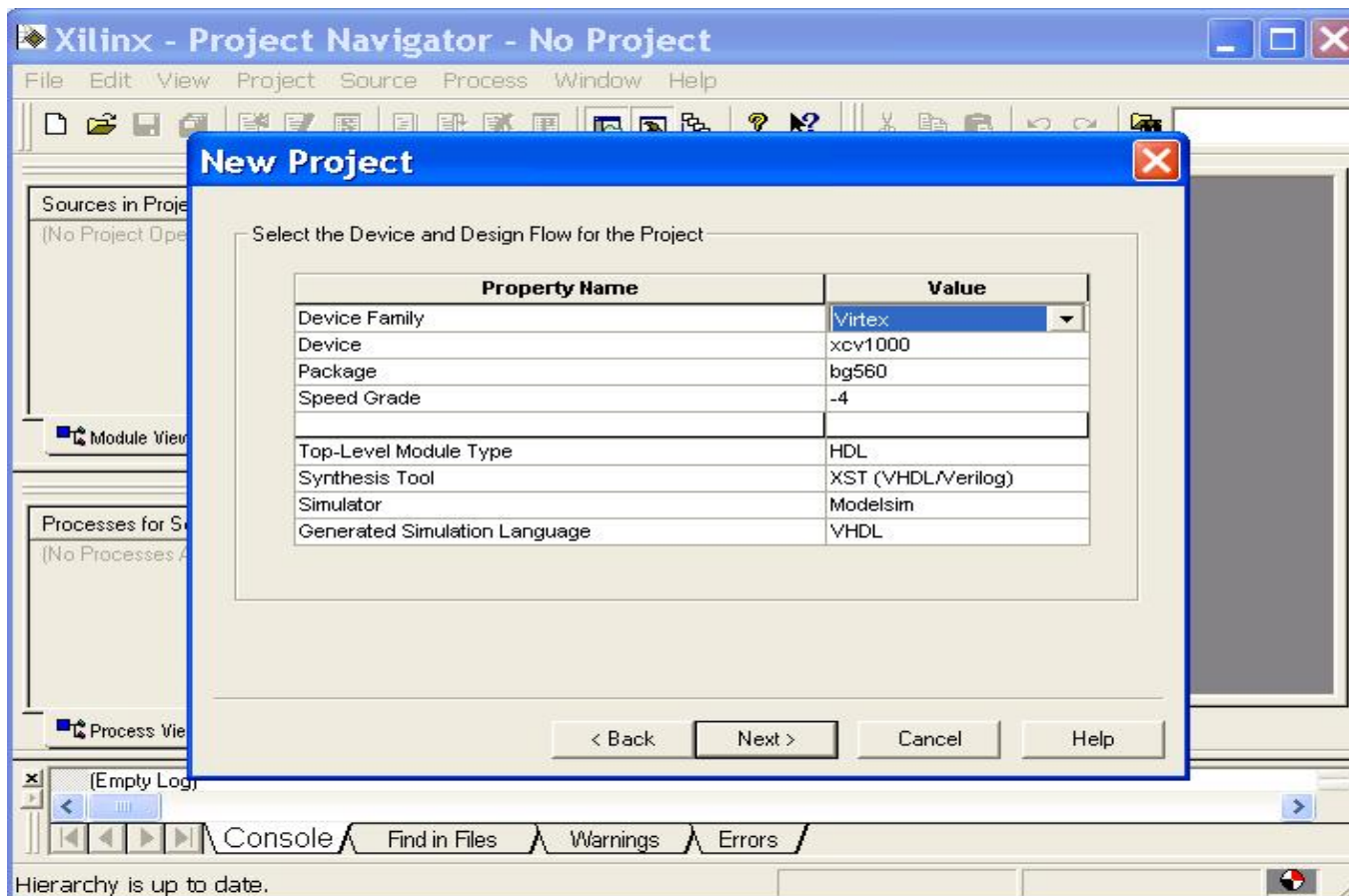
FPGA Synthesis

- File->New Project...



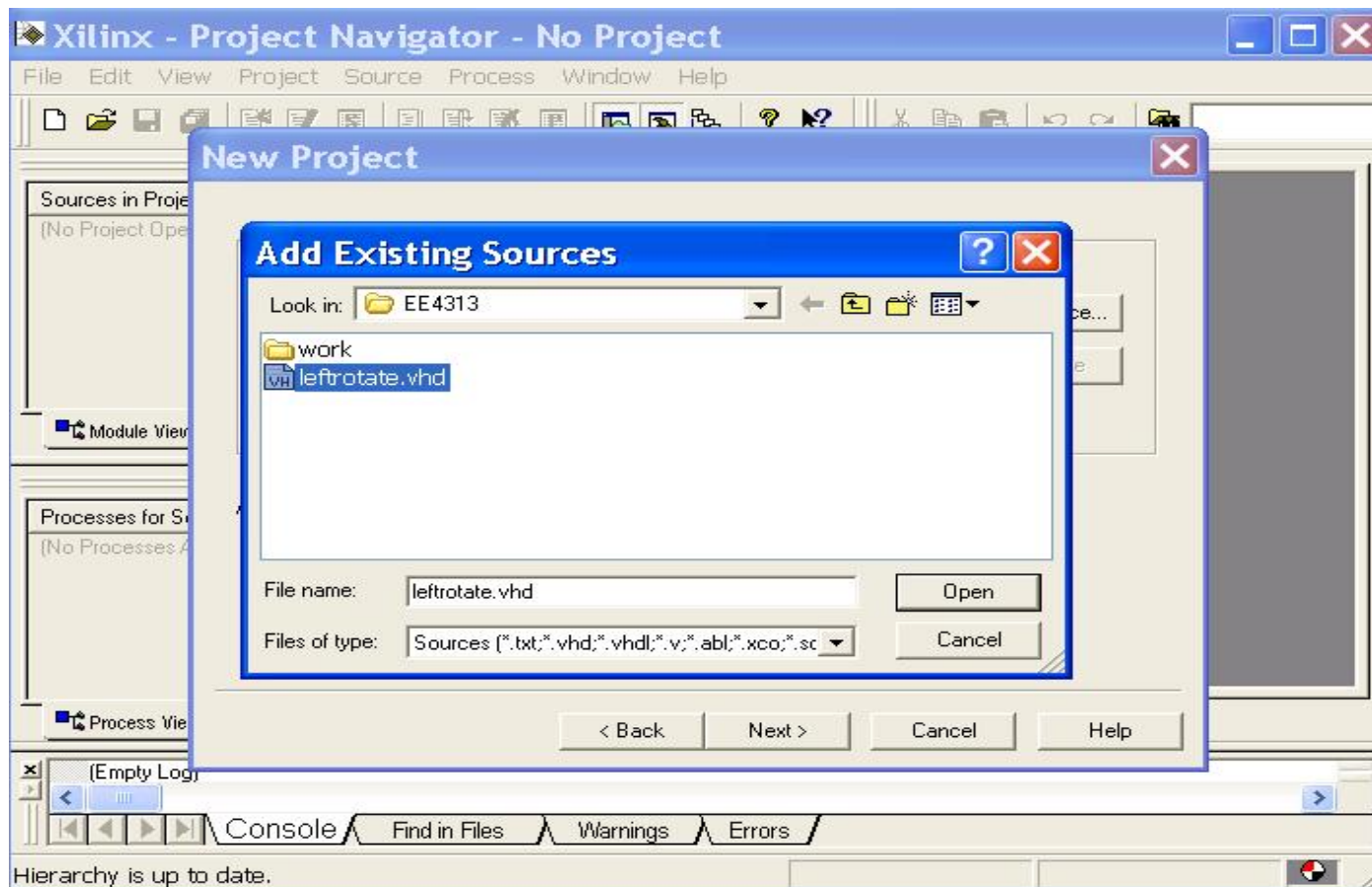
FPGA Synthesis

■ Next >



FPGA Synthesis

■ Next >



FPGA Synthesis

- Next > Simulation Model Properties

The screenshot shows the Xilinx Project Navigator interface with the 'Process Properties' dialog box open. The dialog box is titled 'Process Properties' and has several tabs: 'Post-Map Static Timing Report Properties', 'Post-Place & Route Static Timing Report Properties', 'Translate Properties', 'Map Properties', 'Place & Route Properties', 'Incremental Design Properties', and 'Simulation Model Properties'. The 'Simulation Model Properties' tab is selected, showing a table of properties and their values.

Property Name	Value
Simulation Model Target	Modelsim_VHDL
Post Translate Simulation Model Name	lefttotate_sim.vhd
Post Map Simulation Model Name	
Post Place & Route Simulation Model Name	
Change Device Speed To	-4
Correlate Simulation Data to Input Design	<input checked="" type="checkbox"/>
Retain Hierarchy	<input checked="" type="checkbox"/>
Generate Multiple Hierarchical Netlist Files	<input type="checkbox"/>
Use Automatic Do File for ModelSim Simulation	<input checked="" type="checkbox"/>
Bring Out Global Tristate Net as a Port	<input type="checkbox"/>
Global Tristate Port Name	N/A

Buttons at the bottom of the dialog: OK, Cancel, Default, Help.

FPGA Synthesis

The screenshot displays the Xilinx Project Navigator interface. The main window shows the source code for a VHDL file named `leftrotate.vhd`. The code defines an entity `leftrotate` with two input ports `a` and `b`, and one output port `o`. The architecture `rtl` implements a left rotation function using a 32-bit register `ab_xor` and a 4-bit select signal `b(4 DOWNTO 0)`.

The left sidebar shows the project structure with the following sources:

- leftrotate
- xcv1000-4bg560
 - leftrotate-rtl (leftrotate.vhd)

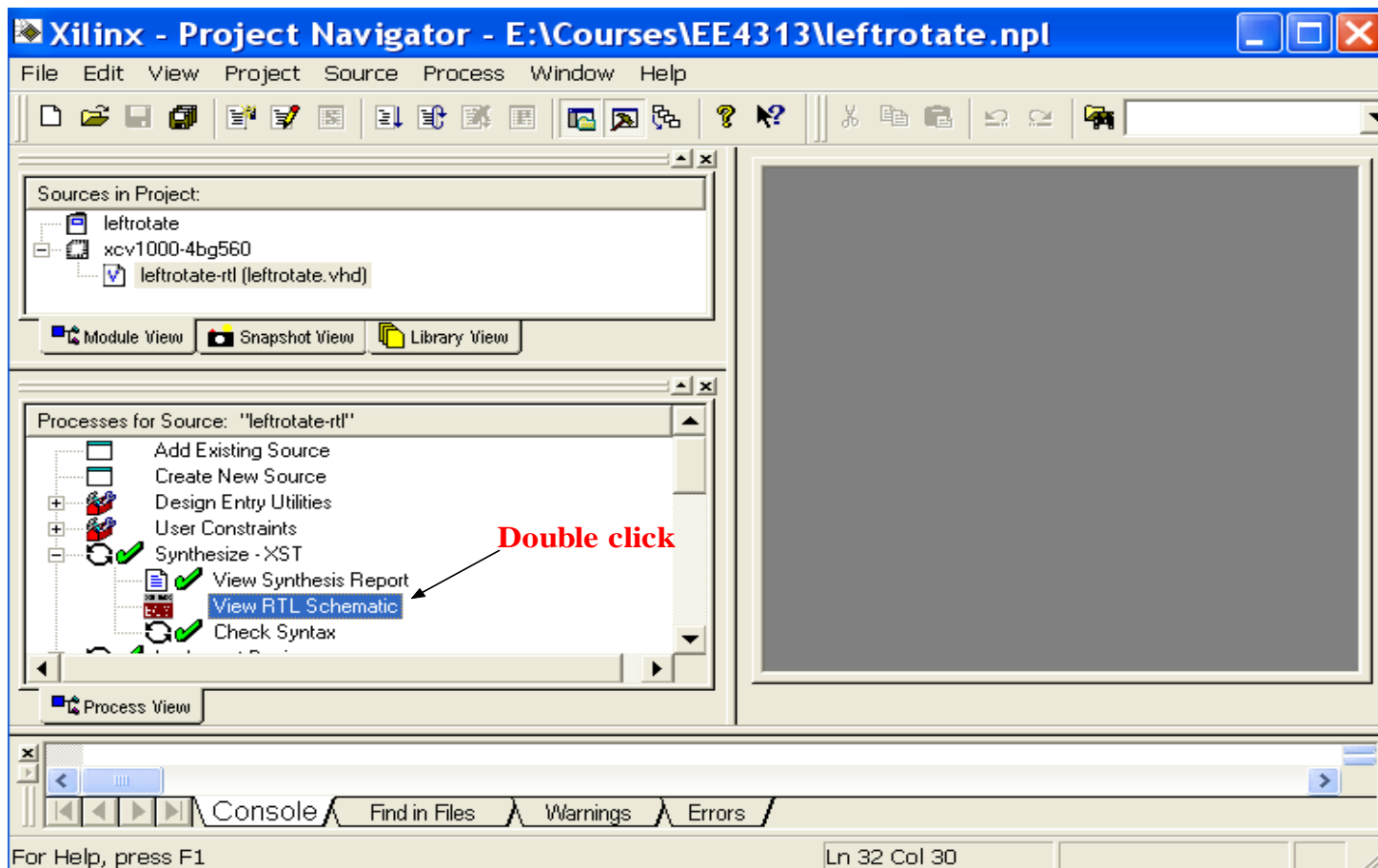
The bottom sidebar shows the processes for the source `"leftrotate-rtl"`. The process `Generate Post-Place & Route Simulation Model` is highlighted, and a double-click action is indicated by an arrow pointing to it.

```
1 LIBRARY IEEE;
2 USE IEEE.STD_LOGIC_1164.ALL;
3
4 ENTITY leftrotate IS
5 PORT ( a: in STD_LOGIC_VECTOR(31 DOWNTO 0);
6       b: in STD_LOGIC_VECTOR(31 DOWNTO 0);
7       o: out STD_LOGIC_VECTOR(31 DOWNTO 0);
8 END leftrotate;
9
10 ARCHITECTURE rtl OF leftrotate IS
11 SIGNAL ab_xor: STD_LOGIC_VECTOR(31 DOWNTO 0);
12 BEGIN
13     ab_xor <= a XOR b;
14
15     WITH b(4 DOWNTO 0) SELECT
16     o <= ab_xor(30 DOWNTO 0) & ab_xor(31) WHEN "0000";
17         ab_xor(29 DOWNTO 0) & ab_xor(31) WHEN "0001";
18         ab_xor(28 DOWNTO 0) & ab_xor(31) WHEN "0010";
19         ab_xor(27 DOWNTO 0) & ab_xor(31) WHEN "0011";
20         ab_xor(26 DOWNTO 0) & ab_xor(31) WHEN "0100";
21         ab_xor(25 DOWNTO 0) & ab_xor(31) WHEN "0101";
22         ab_xor(24 DOWNTO 0) & ab_xor(31) WHEN "0110";
23         ab_xor(23 DOWNTO 0) & ab_xor(31) WHEN "0111";
24         ab_xor(22 DOWNTO 0) & ab_xor(31) WHEN "1000";
```

Console output at the bottom shows the command `_runTOvxsim.tcl` and the message `Generating Post PAR VHDL Model`.

FPGA Synthesis

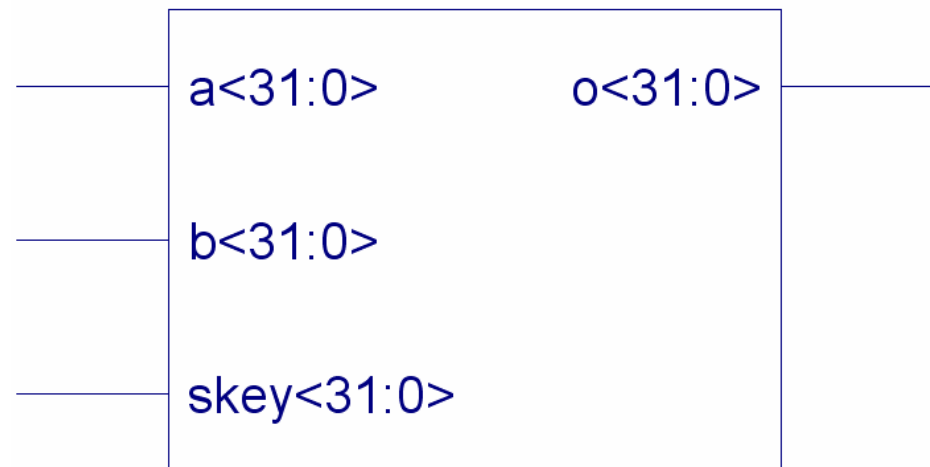
View RTL Schematic



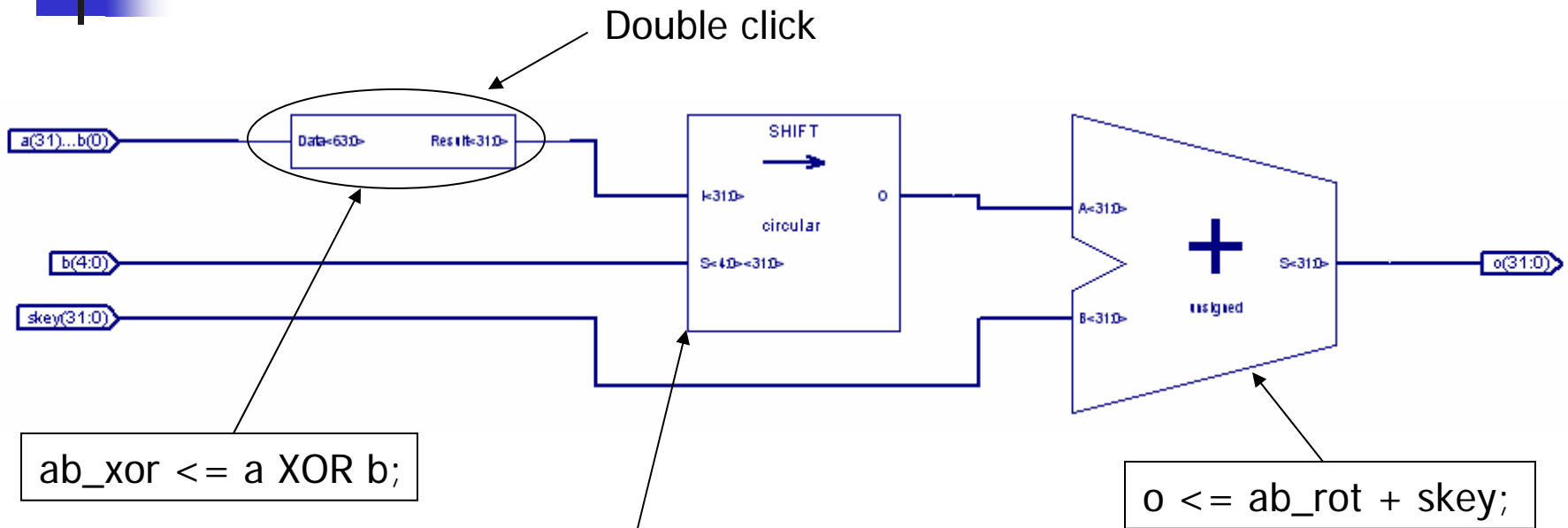
FPGA Synthesis

```
ENTITY leftrotate IS
  PORT ( a    : in STD_LOGIC_VECTOR(31 DOWNTO 0);
        b    : in STD_LOGIC_VECTOR(31 DOWNTO 0);
        skey: in STD_LOGIC_VECTOR(31 DOWNTO 0);
        o    : out STD_LOGIC_VECTOR(31 DOWNTO 0));
END leftrotate;
```

Double click

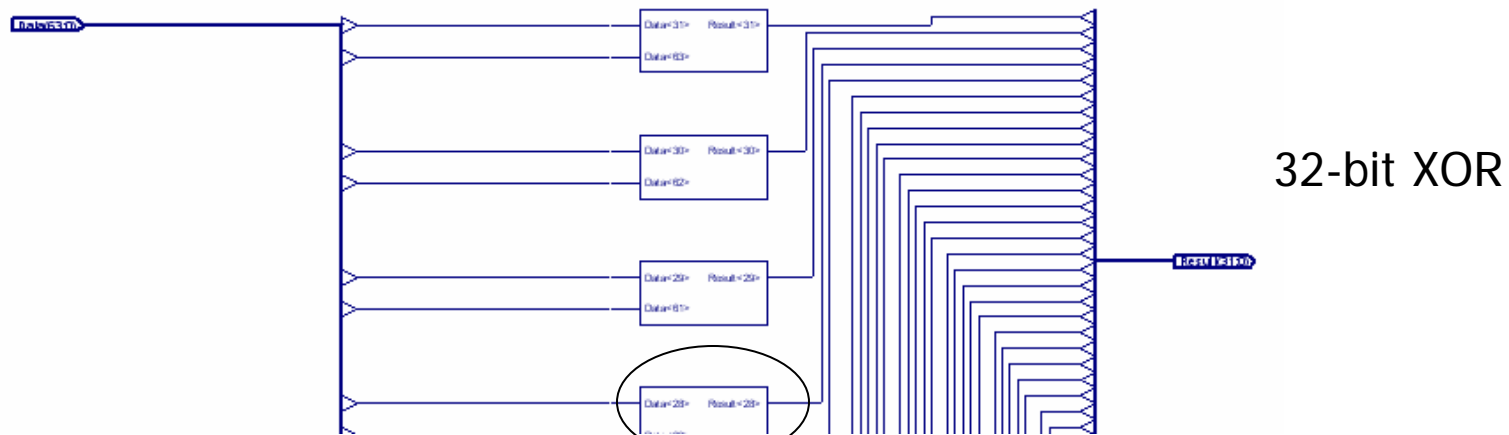


FPGA Synthesis



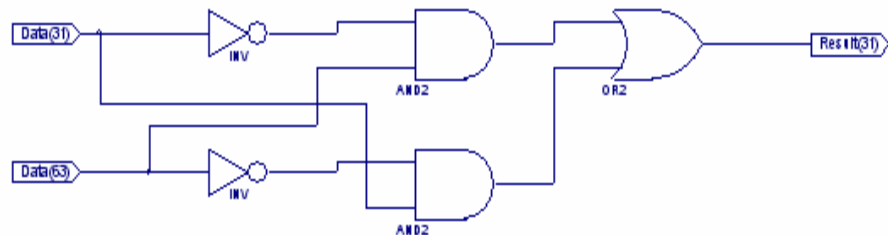
```
WITH b(4 DOWNT0 0) SELECT  
ab_rot<=ab_xor(30 DOWNT0 0) & ab_xor(31) WHEN "00001",  
      ab_xor(0) & ab_xor(31 DOWNT0 1) WHEN "11111",  
      .....  
      ab_xor WHEN OTHERS;
```

FPGA Synthesis



Double click

1-bit XOR





Timing Simulation

- Make sure timing simulation models are successfully generated
 - leftrotate_sim.vhd and leftrotate_sim.sdf
 - sdf stands for “standard delay format”
- Start ModelSim

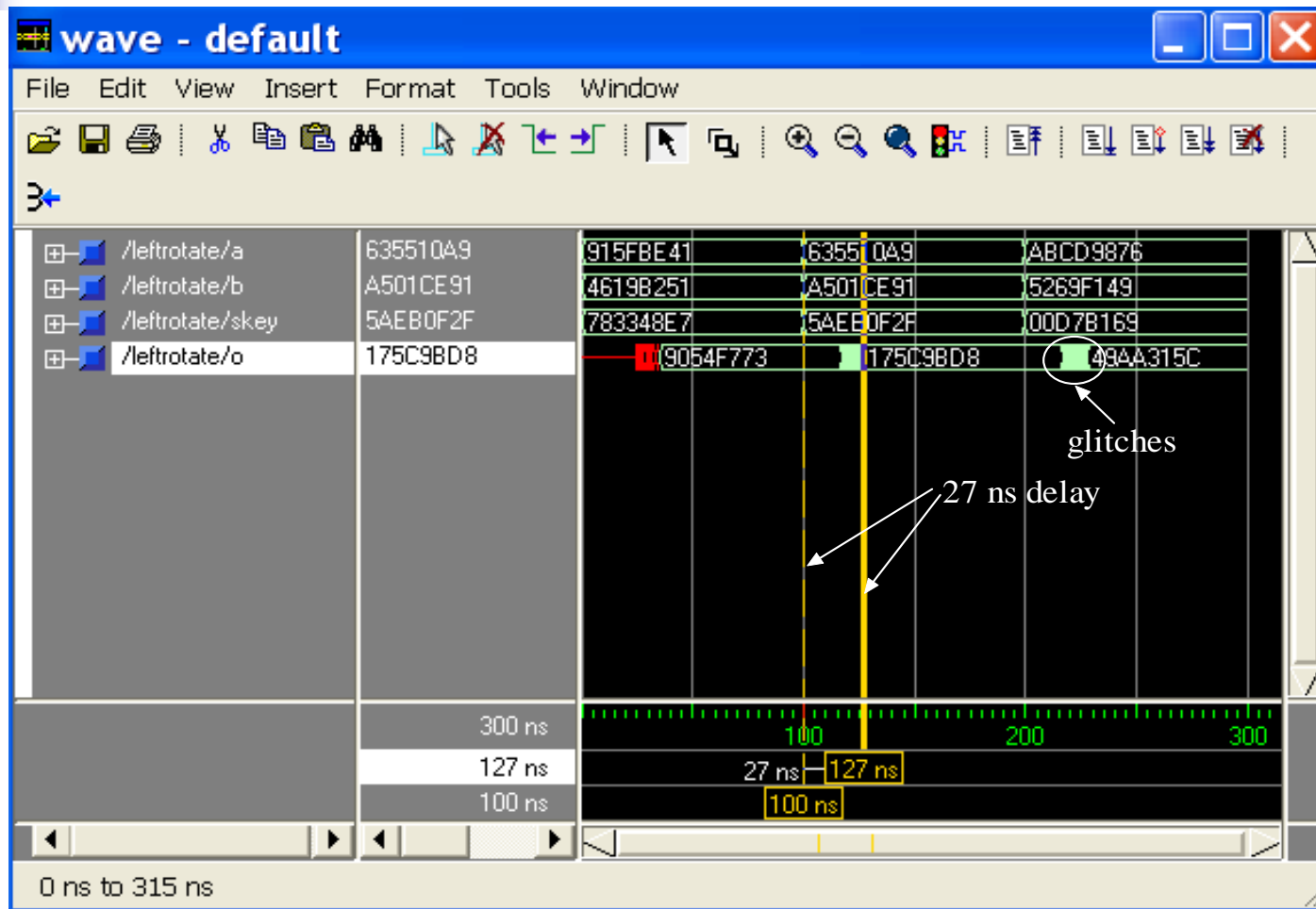


Timing Simulation

- In ModelSim console window, type in following commands in sequence

```
cd {c:/ee4313/proj1}
vlib work -- create a work library
- map simprim to pre-compiled standard library
Vmap simprim c:/modeltech_5.6d/xilinx/simprim
vcom leftrotate_sim.vhd -- "compile" simulation model
- run timing simulation with delay information (leftrotate_sim.sdf)
vsim -sdftyp /=E:/Courses/EE4313/leftrotate_sim.sdf work.leftrotate
view wave -- start waveform viewer
add wave a b key o -- add input/out signals in the waveform viewer
- assign input signals (use hexadecimal, time unit ns)
force a 16#915FBE41 0, 16#635510A9 100, 16#ABCD9876 200
force b 16#4619B251 0, 16#A501CE91 100, 16#5269F149 200
force key 16#783348E7 0, 16#5AEB0F2F 100, 16#D7B169 200
force 300 -- simulate 300ns
```

Timing Simulation



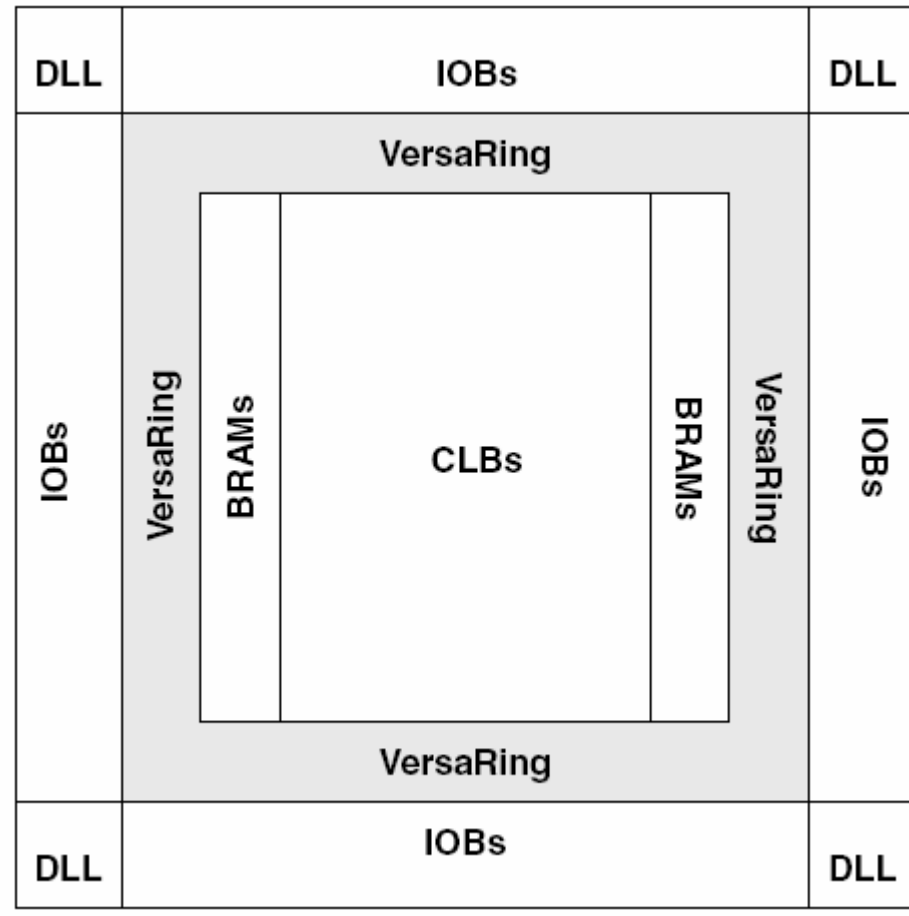


Results

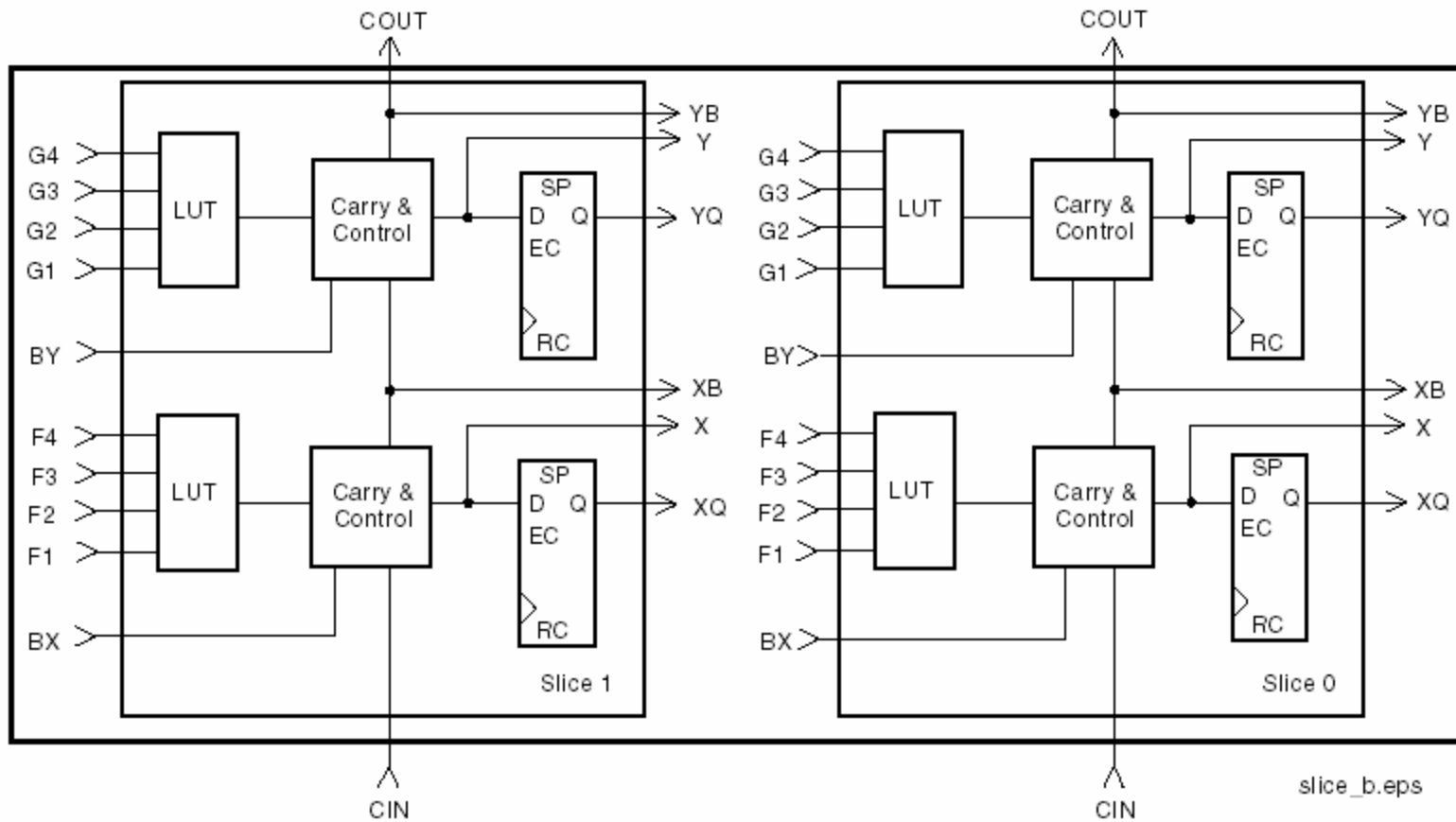
- Timing – 27ns delay (from ModelSim)
- Area
 - Logic Utilization: Number of 4 input LUTs: 188 out of 24,576 1%
 - Logic Distribution: Number of occupied Slices: 104 out of 12288 1%
 - Number of Slices containing only related logic: 104 out of 104 100%
 - Number of Slices containing unrelated logic: 0 out of 104 0%
 - Number of 4 input LUTs: 188 out of 24,576 1%
 - Number of bonded IOBs: 128 out of 404 31%
 - Total equivalent gate count for design: 1,314



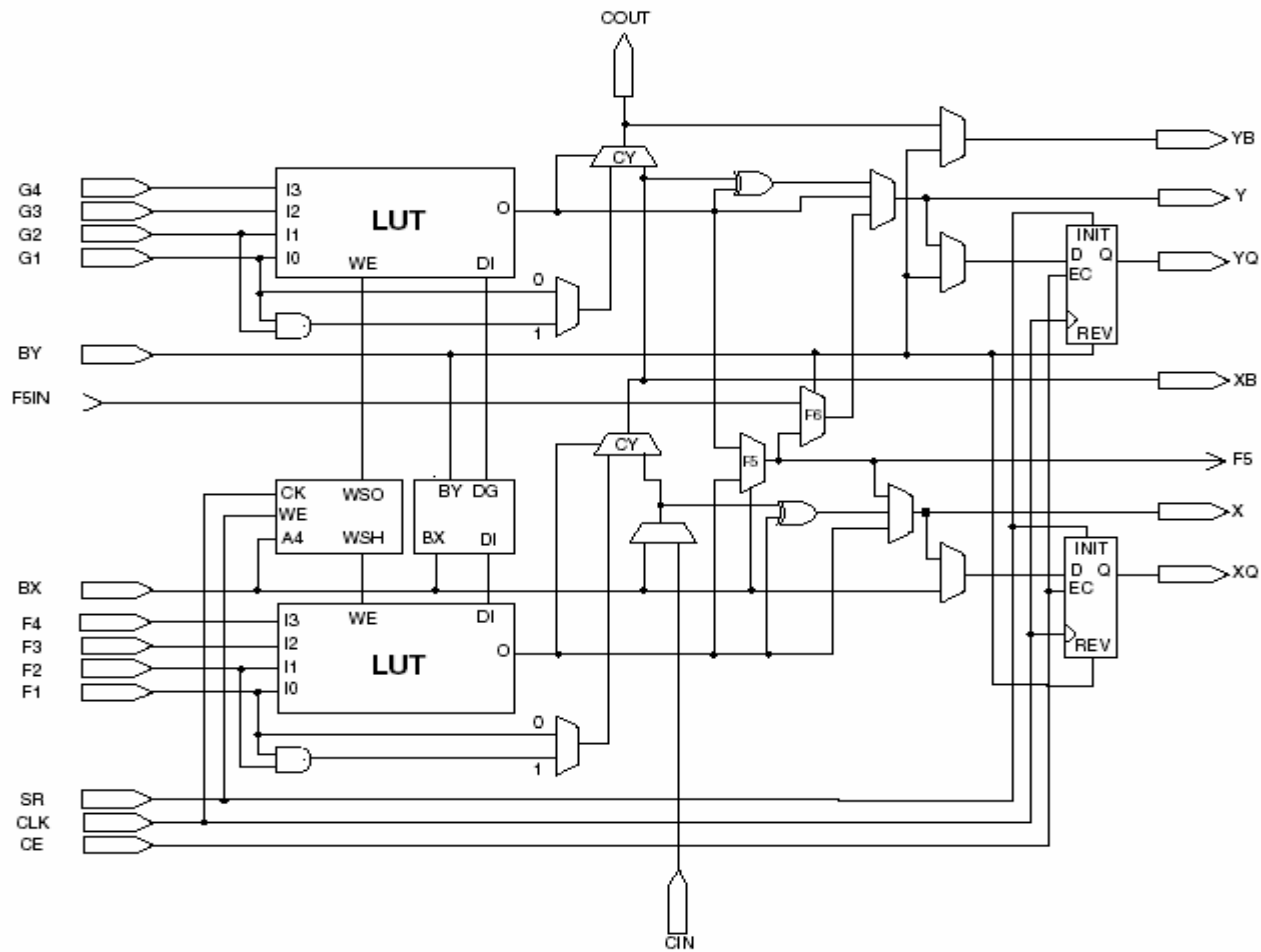
FPGA Architecture



FPGA Architecture



FPGA Architecture





Exercise 2

- Modify the previous model to implement $O = ((B - \text{SKEY}) \ggg A) \text{ XOR } A;$
 - Simulate the Model
 - Synthesize the Design
 - Draw and Explain the Synthesized Design