CPE 301
Microprocessor System Design Lab
Field Programmable Gate Arrays

Example-2: 2–Input Gates

Code-2:

```
module gate2 (
input wire a,
input wire b,output wire [5:0] z
);

assign z[5] = a & b;
assign z[4] = ~(a & b);
assign z[3] = a | b;
assign z[2] = ~(a | b);
assign z[1] = a ^ b;
assign z[0] = ~(a ^ b);

endmodule
```

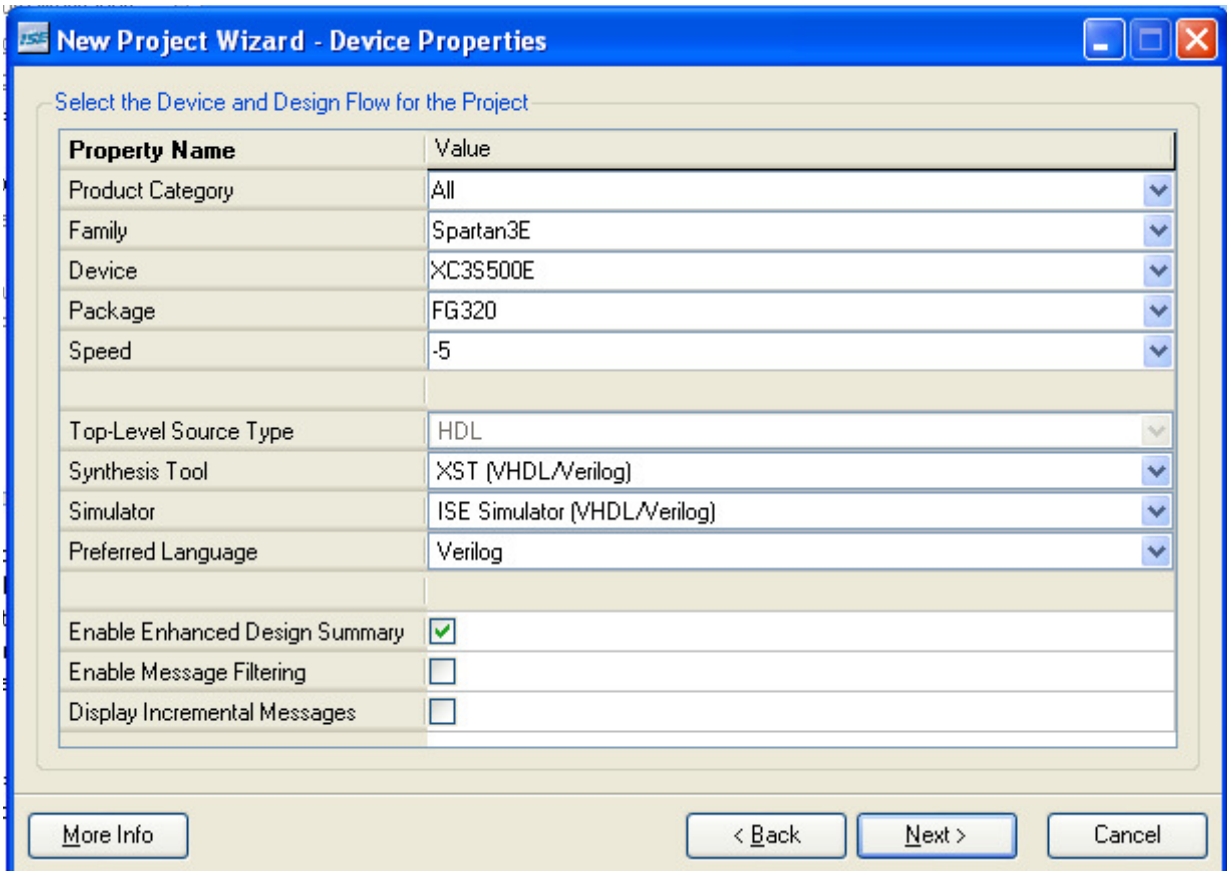
Instructions For Xilinx ISE Design Suite Program

1) Double click “Xilinx ISE, Project Navigator”

2) Click File - New Project.
Create a new folder on the desktop and create your projects in the folder.
Fill in the name field and put your folders path in the Project Location field.

Top level source type: HDL

3) Check your FPGA IC on the board. **Spartan, XC3S500E, FGG320**



4) Skip the create new source by clicking “next”

5) Skip the add source by clicking “next”

6) Click “finish”

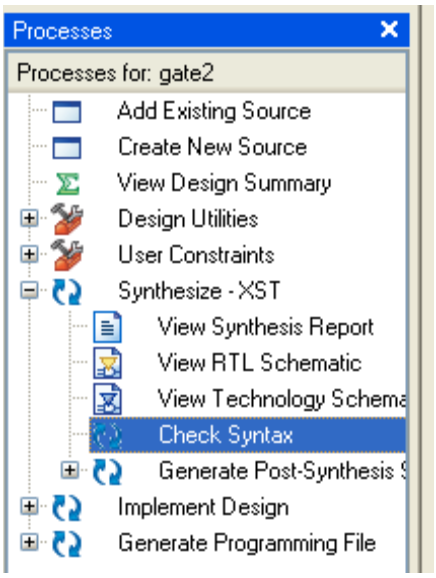
7) In the processes window, Click “Create new source”

8) Select “Verilog Module” and Filename: gate2. Make sure “add to project” option is selected

9) Skip the port names by clicking “next”. Then Click finish.

10) Now we have gate2.v verilog source file. Modify the source file as in **Code-1** given above and click save.

11) Double click “check syntax” in the processes window. Make sure that there are no errors.



12) Now, we can simulate our module.
In Sources window, select behavioral simulation from the combo box
Double click “Create New Source” .
Select “Verilog Text fixture” and the name field: gate2_top

13) Check that, top module has encapsulated the gate2 module in the Sources window. Text fixtures are used to apply logic levels to UUT (Unit under Test) and plot the outputs.

14) Consider the **Initial begin** block;
character represents the delay. Apply all combinations to the inputs. After each change make sure you put some delay so that it s effect can be observed on the outputs.

Example:

```
initial begin
        a = 0;
        b = 0;
        // Wait 100 ns for global reset to finish
        #100;
        a=1;
        #20;
        a=0;
        b=1;
        #20;
        a=1;
```

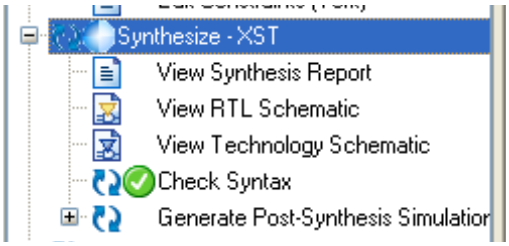
15) Save your file and double click check syntax in processes window.

16) If there are no errors double click “simulate Behavioral Model”

17) Analyze the waveforms in the simulation window. Are the output waveforms as expected? If not, follow (modify-simulate) cycle until you get the expected behavior.

18) From the Sources window, select “synthesis/implementation” and activate your source code by single click.

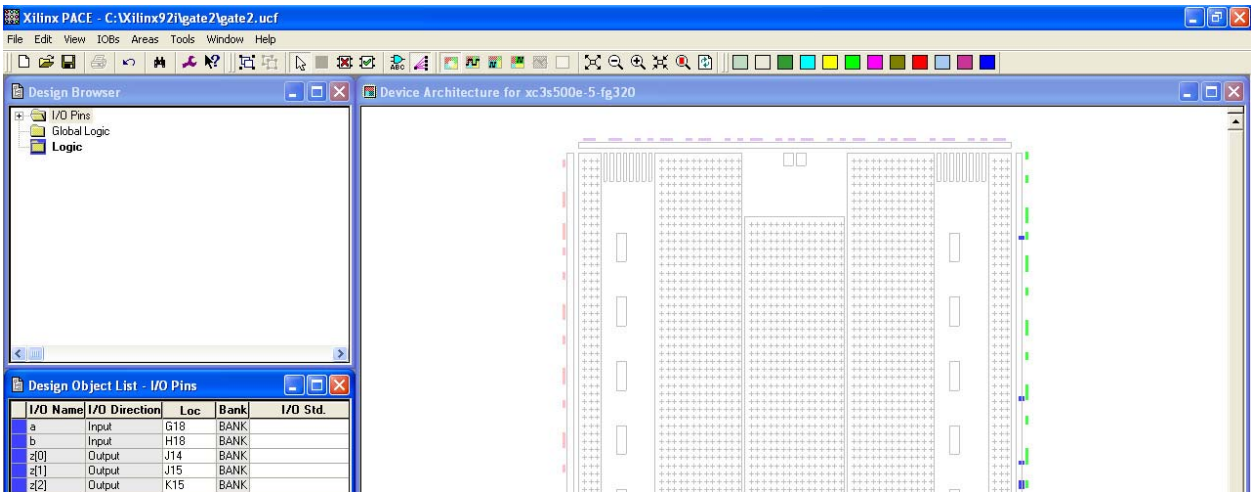
19) In the processes window, double click Synthesize-XST.



20) Select User Constraints and Double click “Assign Package Pins”. Click “Yes” to add ucf file.

21) PACE program is used to generate user constraint files. In this step we assign (i.e. match) necessary I/O circuit elements (switches, push buttons, LEDs) on the board to our I/O pins of our module gate2.

22) Click LOC field and **Type** necessary numbers for I/O. You can find the numbers next to the elements on the board.



23) Click save and OK. Synplify Verilog Default. Then close PACE.
Note: After closing PACE, if you want to make changes in the PACE program and face problems, Go to the project folder, delete **.ucf** file and restart PACE.

24) Rerun Synthesize and run Implement Design. Make sure all Translate, Map, and Place&Route steps are successfully completed.

25) Double Click “Generate Programming File”. In this step Xilinx ISE creates **.bit** file which we will download to our board. After this step make sure that **.bit** file is in the project folder.

26) Open Digilent-Adept program in the start menu.

27) Connect **ONLY USB** cable. Click “Initialize chain” in adept and make sure device is found.

28) Browse **.bit** file and click program for FPGA XC3S500E. Note that we do not program PROM.

29) Test your board by applying necessary inputs.