Experiment step 5.1.1

After logging in to the PC with your NTU account, open File Explorer.

Go to the project location (e.g. drive J) specified by the lab executive, create a New Folder and name it **Lab3**.

Copy the 3 given files *vsevenseg.v, vsevenseg_tb.v* and *vsevenseg.xdc* from NTULearn and place them in the newly created folder **Lab3**.

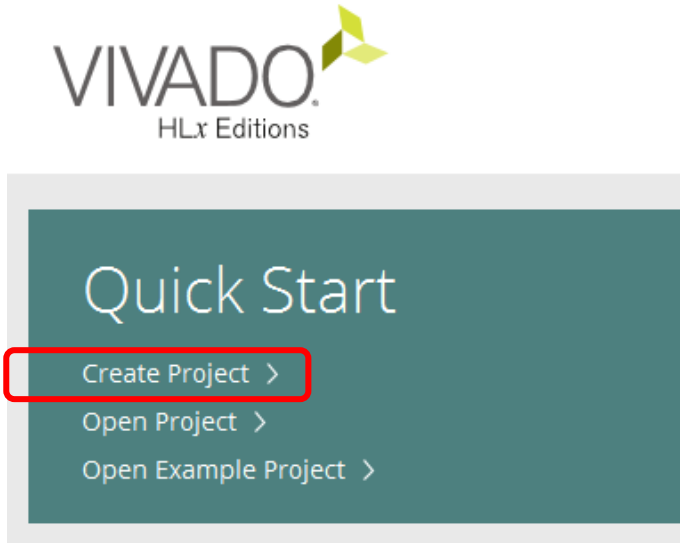
Check that the file extensions (e.g. *filename.**v**, filename.**xdc***) are not modified. Also, if any filename contains brackets, e.g. *filename(1).v*, rename the file to get rid of the brackets.

Lab 3 Vivado guide

Part A: Create project and add a design file

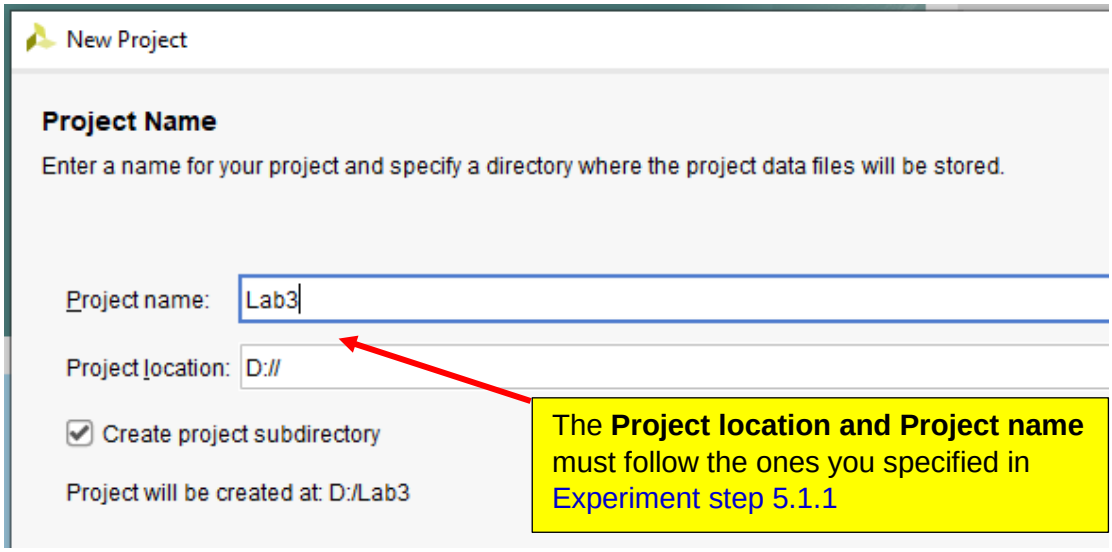
1. **Double click Vivado 2018.3 shortcut to begin**
   Wait patiently. The software may take a while to open

2. **Click Create Project on Quick Start**

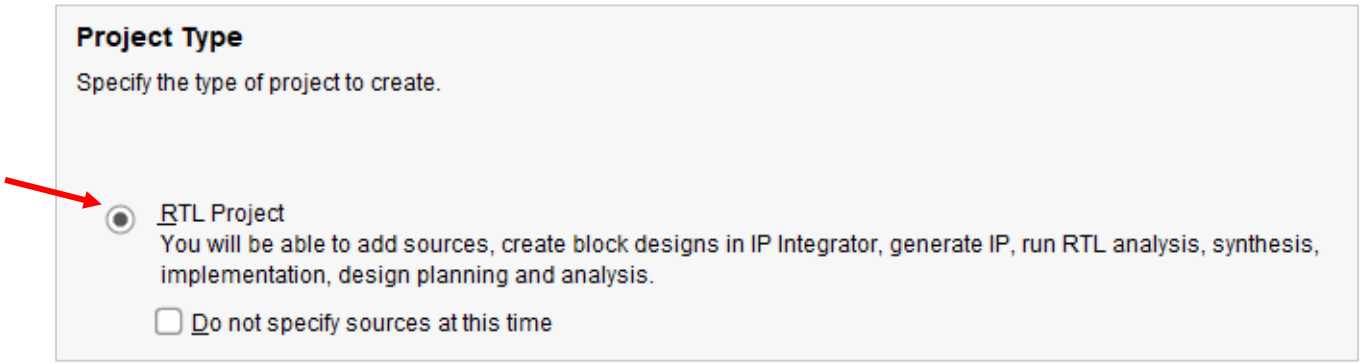


If this is your first time using Vivado, please **follow every step carefully** so that you do not run into any unexpected problem which may prevent you from completing the experiment

3. **Click Next on the "Create a New Vivado Project" wizard**
4. **Use the same Project location (e.g. J) and Project name (e.g. Lab3) specified above in** Experiment step 5.1.1
   Click Next

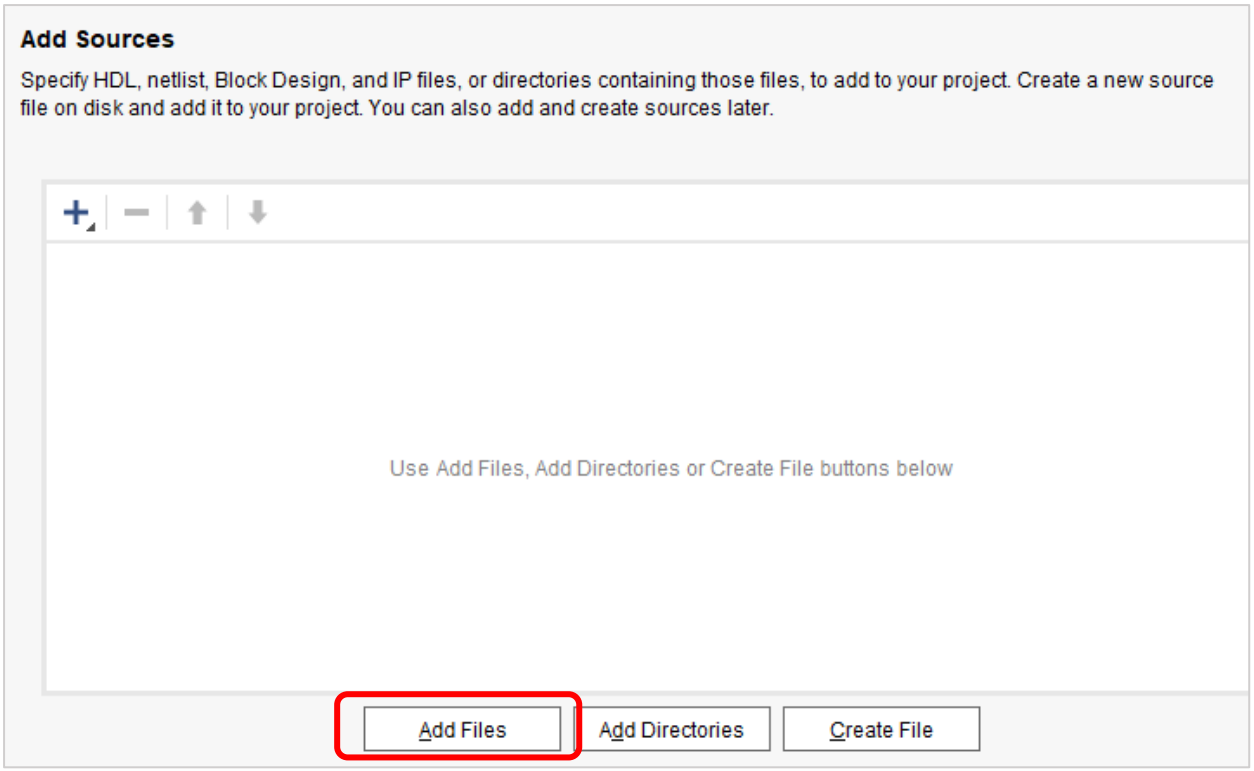


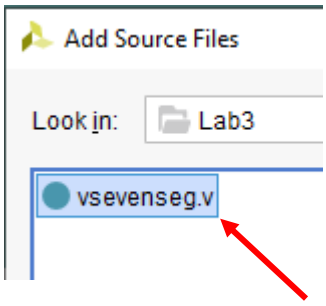
5. **Select RTL Project. Click Next.**



6. **Click Add Files. Select the file vsevenseg.v, click OK to add it to the project**

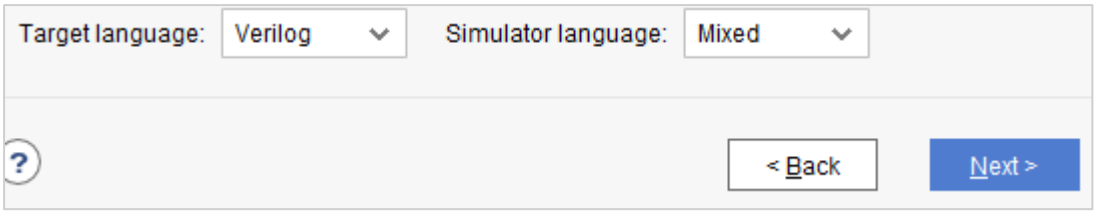
   You should have downloaded the necessary files from NTULearn and placed them in the project location and folder specified in Experiment step 5.1.1. If you are not able to find the files, open Windows Explorer, copy/move the files to the correct project location and folder before proceeding.



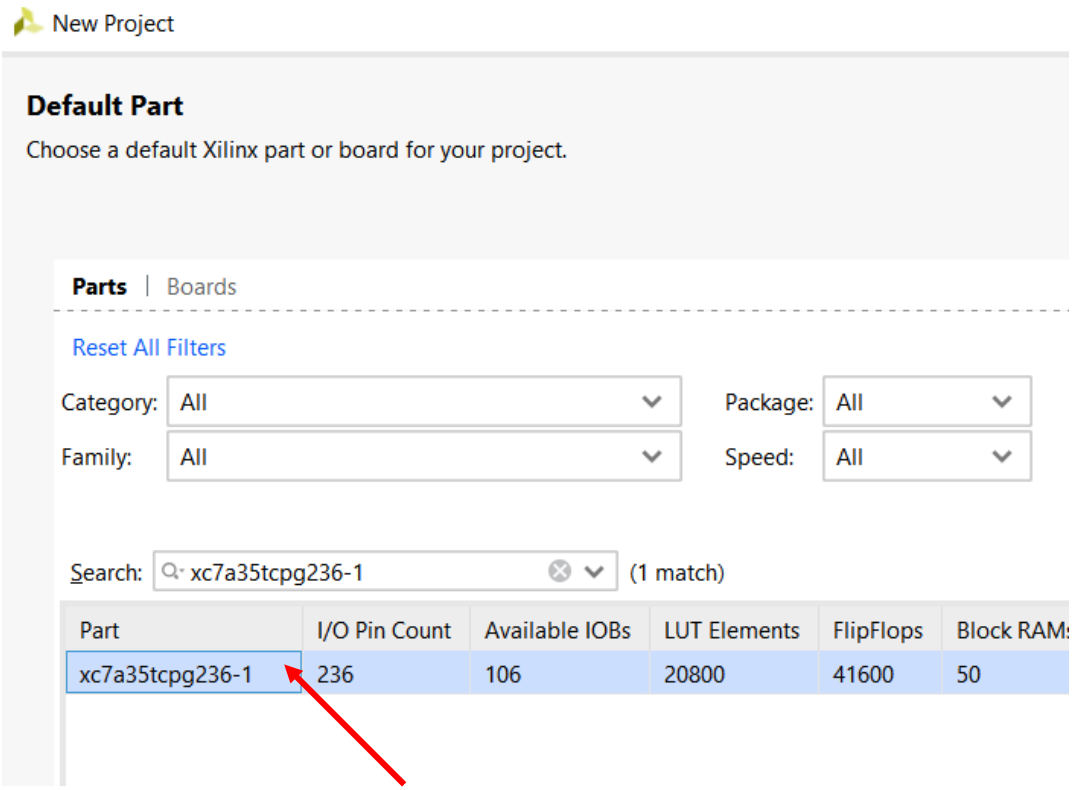


Select *vsevenseg.v* and click OK.

7. **Check that Verilog is selected for Target language. Click Next.**
   Click Next on the "Add Constraints (optional)" dialogue box.



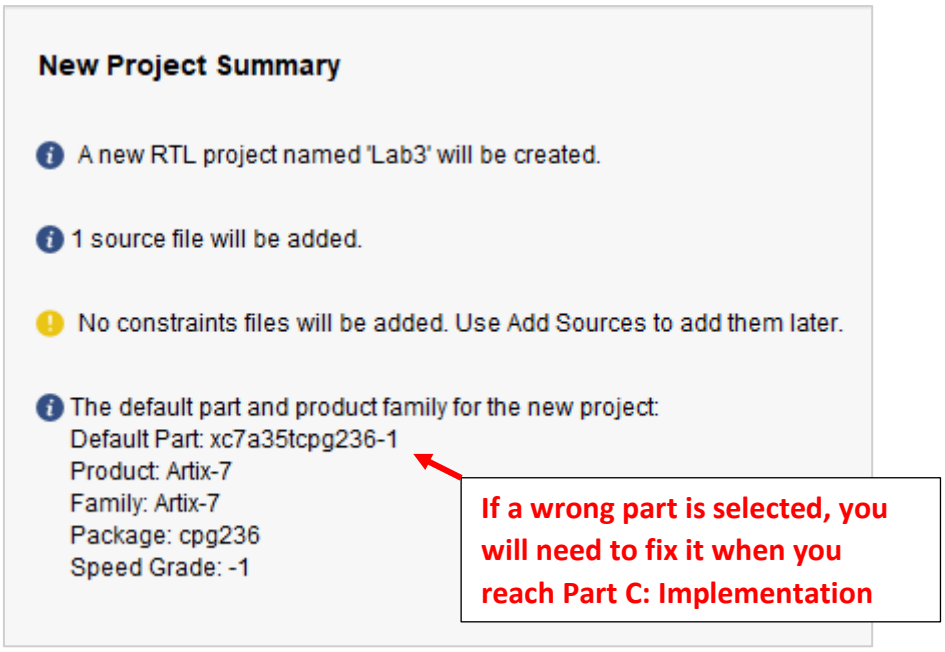
8. **For "Default Part", select Family (Artix7), Package (cpg236), Speed (-1)**
   Alternatively, copy and paste xc7a35tcpg236-1 into the search box
   Click to select the part xc7a35tcpg236-1 and click Next



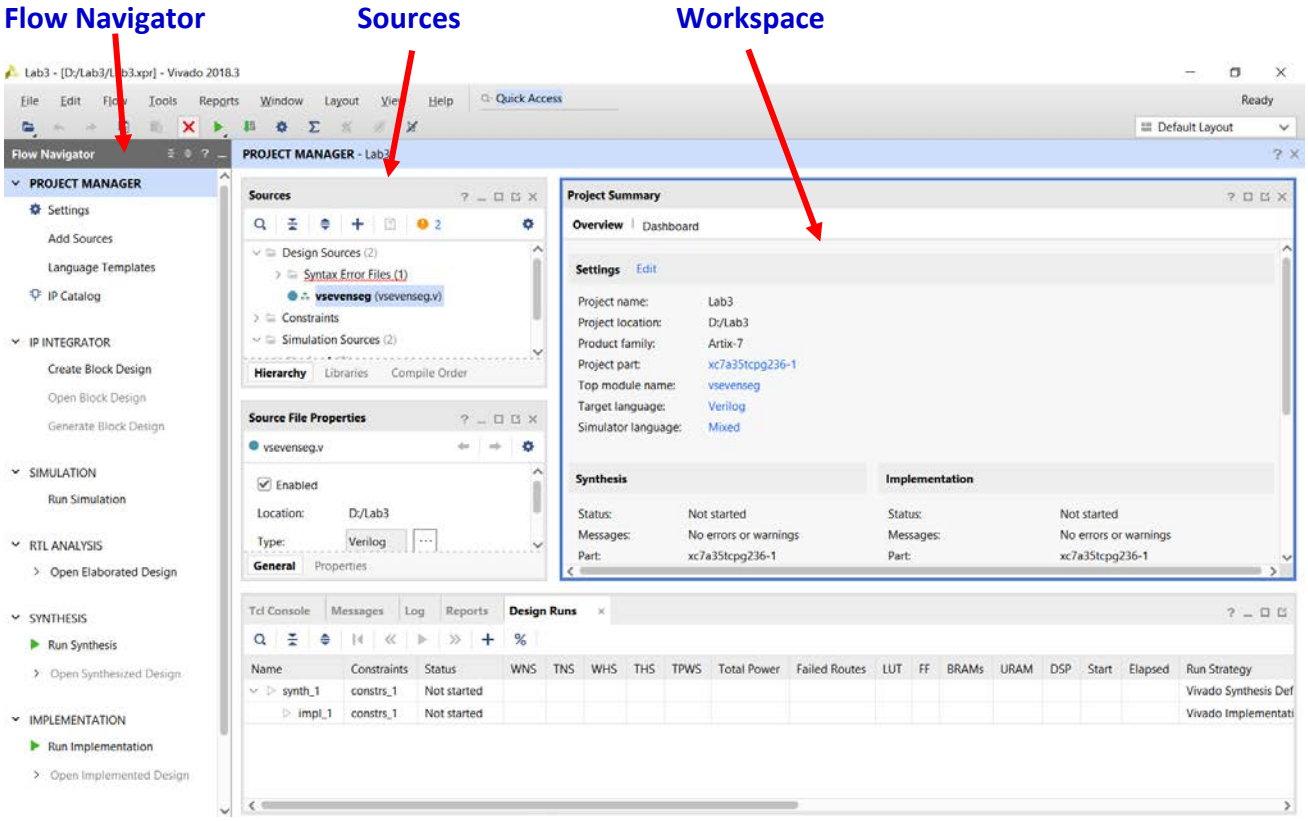
Click to select the correct part

If a wrong part is selected, you will need to fix it when you reach Part C: Implementation

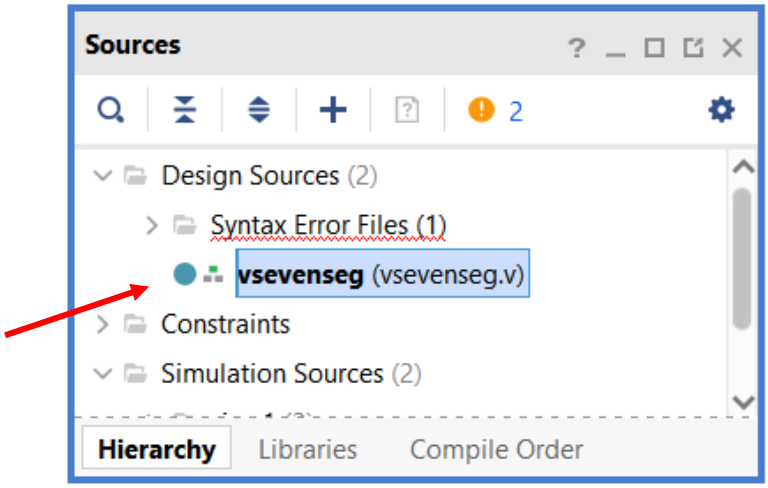
9. Check the New Project Summary and click Finish



Take note of three main areas of the Vivado application



10. In Sources, double click *vsevenseg.v* to open the design file in the workspace



11. Enter the expressions for segments c, d and f into the design file

(You should have obtained these minimum-cost SOP expressions using Karnaugh maps)

Project Summary | vsevenseg.v *

D:/Lab3/vsevenseg.v

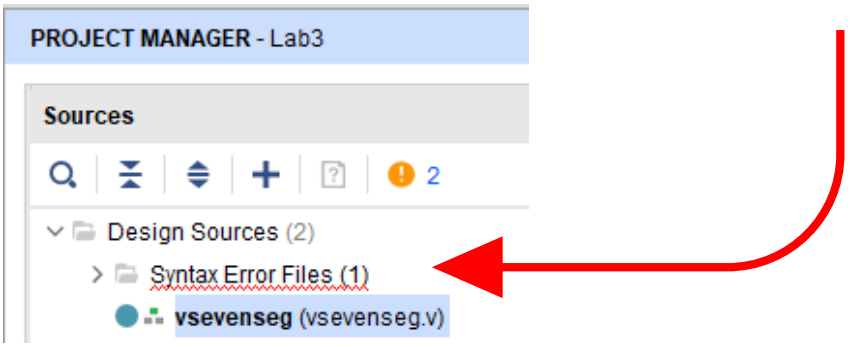
```
37  // turn on only the two rightmost digits - active low
38  assign anode_L = 4'b1100;
39
40  // pull cathode low to light up segment - active low
41  assign seg_L = ~seg;
42
43  // x format {msb, ., ., lsb}
44  // seg[6:0] format {g, f, e, d, c, b, a} - active high
45  // segment a, b, e, g expressions are given
46    // segment a
47     assign seg[0] = ~x[3]&x[2]&x[0]|x[2]&x[1]|~x[3]&x[1]|x
48    // segment b
49     assign seg[1] = ~x[3]&~x[2]|~x[2]&~x[0]|~x[3]&~x[1]&~x
50    // segment e
51     assign seg[4] = x[3]&x[2]|x[3]&x[1]|~x[2]&~x[0]|x[1]&~
52    // segment g
53     assign seg[6] = x[1]&~x[0]|x[3]&~x[2]|x[3]&x[0]|~x[2]&
54
55  ////////////////////////////////////////////
56  // students to fill in these 3 expressions
57  // remember to end each statement with a semicolon ";"
58    // segment c
59     assign seg[2] =
60    // segment d
61     assign seg[3] =
62    // segment f
63     assign seg[5] =
64
65  endmodule
```

Note that there is no signal x[4] in this circuit design

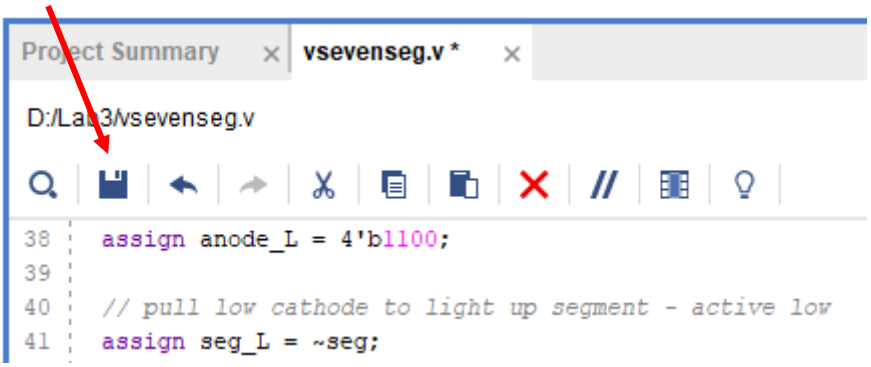
Enter the Boolean expressions for these 3 segments in Verilog syntax.

End each statement with a semicolon ";"

You must key in the Boolean expressions (in Verilog syntax) for the 3 remaining segments. Otherwise the circuit design is incomplete, error is flagged and you will not be able to proceed.

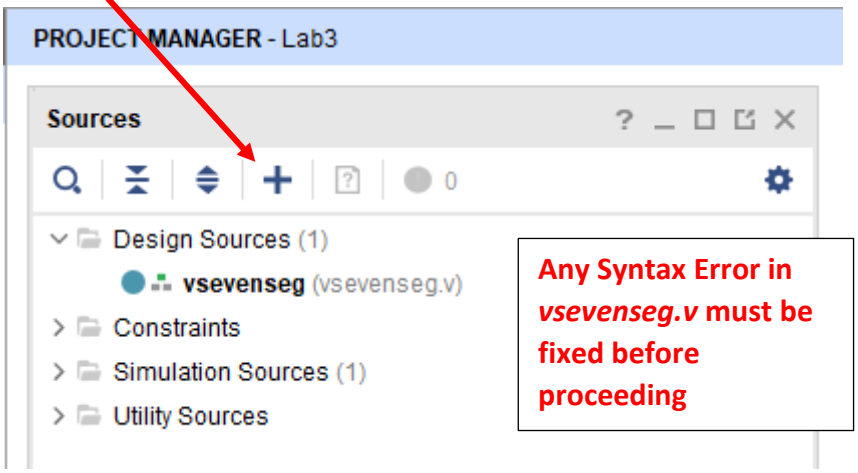


12. Click the Save button to save your design <u>after</u> inserting all three expressions

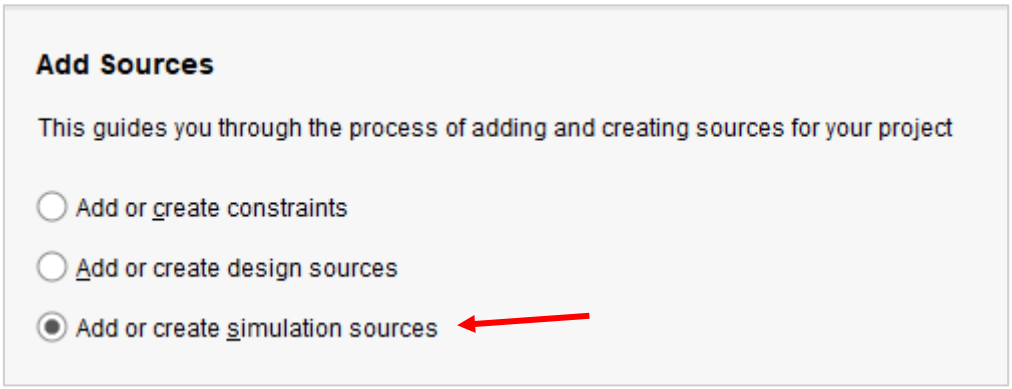


Part B: Add test bench for simulation

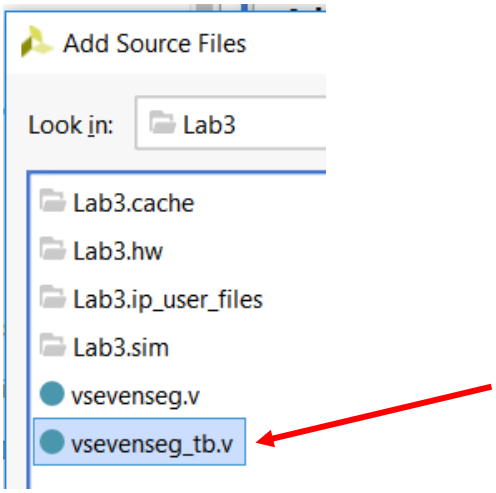
13. In Sources, click + to add a test bench for simulation



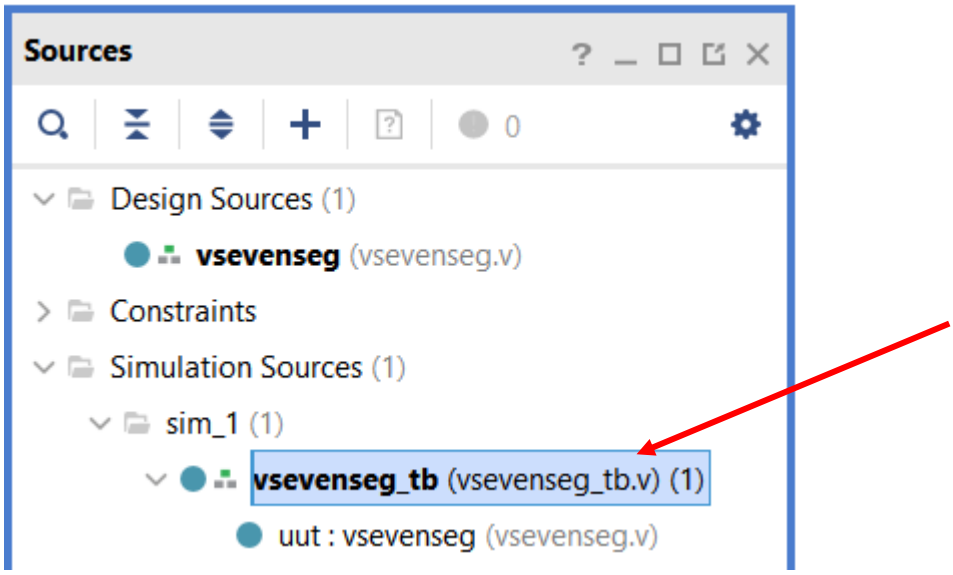
14. Select Simulation Sources, click Next



15. **Click Add Files, select *vsevenseg_tb.v* and click OK, click Finish**
(You should have downloaded the test bench file from NTULearn and placed it in the same project subdirectory or folder)



16. **Click > to expand Simulation Sources. Double click the file *vsevenseg_tb.v* to open it in the workspace**



Verilog design files and test benches have the file extension **.v**

The suffix **_tb** is a naming convention to indicate clearly that it is a **test bench**, not a design file.

17. Check that the module name, the input and output port names of the test bench match those of your design.

Project Summary | vsevenseg_tb.v

D:/Lab3/vsevenseg_tb.v

```
22
23  module vsevenseg_tb(
24      );
25
26      // Inputs injected to UUT
27      reg [3:0] x;
28
29      // Outputs observed from UUT
30      wire [6:0] seg_L;
31
32      // Instantiate the Unit Under Test (UUT)
33      // module name and port names must follow design source
34      vsevenseg uut (
35          .x(x),
36          .seg_L(seg_L)
37      );
38
39      initial begin
```

You will soon learn more about module instantiation in this course

In this case, the module name is vsevenseg

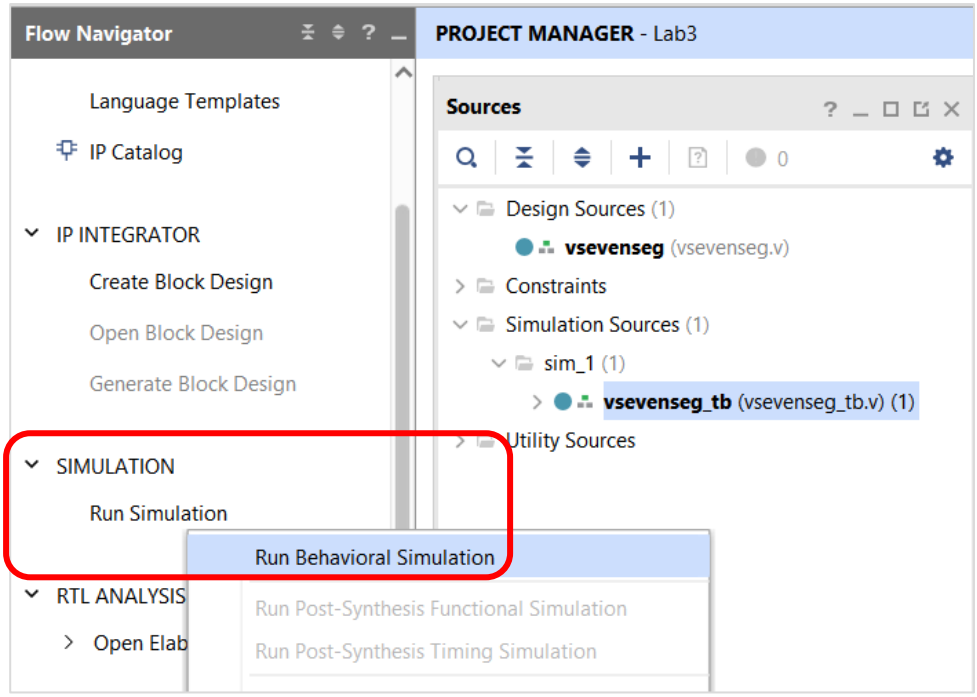
The input port name is x

The output port name is seg_L

18. Note that the test bench specifies a series of input values (each value lasts for 10 time units) for the unit under test

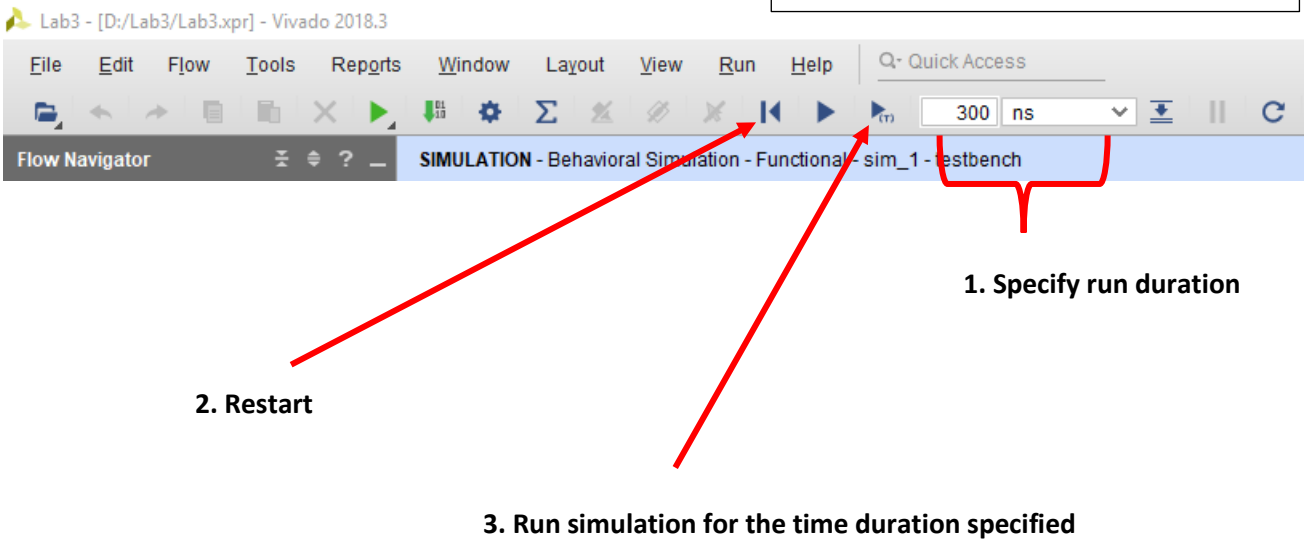
```
50          #10 x = 4'h0;
51          #10 x = 4'h1;
52          #10 x = 4'h2;
53          #10 x = 4'h3;
54          #10 x = 4'h4;
55          #10 x = 4'h5;
56          #10 x = 4'h6;
57          #10 x = 4'h7;
58          #10 x = 4'h8;
59          #10 x = 4'h9;
60          #10 x = 4'ha;
61          #10 x = 4'hb;
62          #10 x = 4'hc;
63          #10 x = 4'hd;
64          #10 x = 4'he;
65          #10 x = 4'hf;
```

19. On Flow Navigator, click Run Simulation > Run Behavioral Simulation

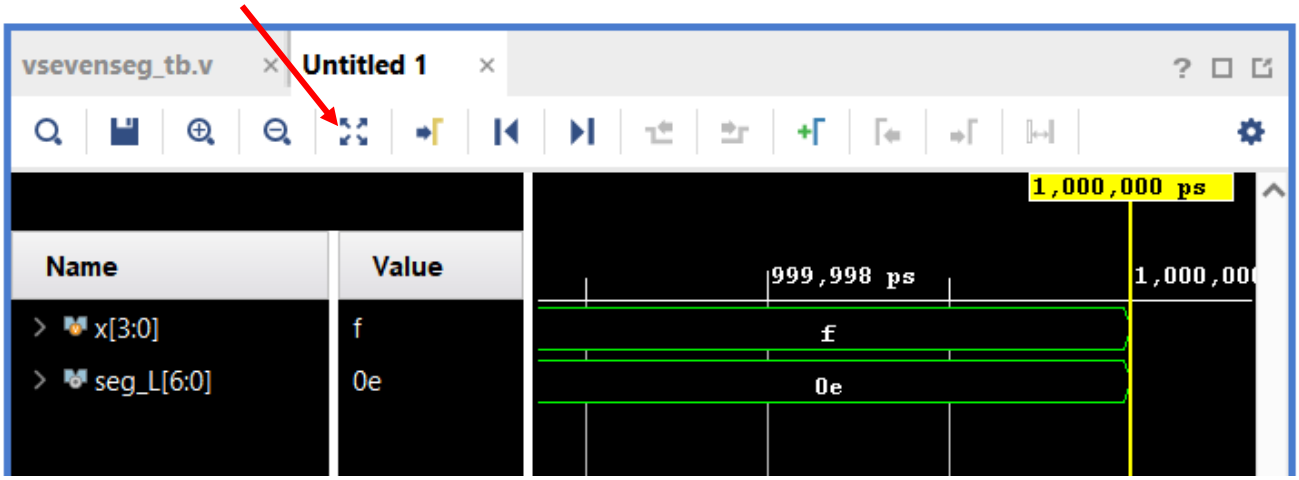


20. Specify 300ns for run duration
Click Restart
Run simulation for 300 ns

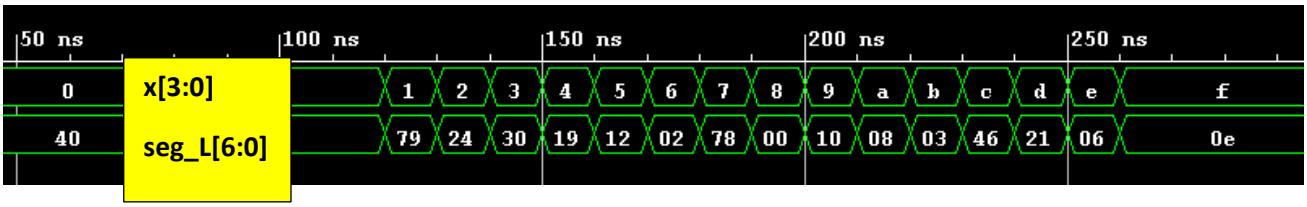
1 nanosecond = 10^(-9) second



21. Click Zoom Fit to get a good view of the simulation result

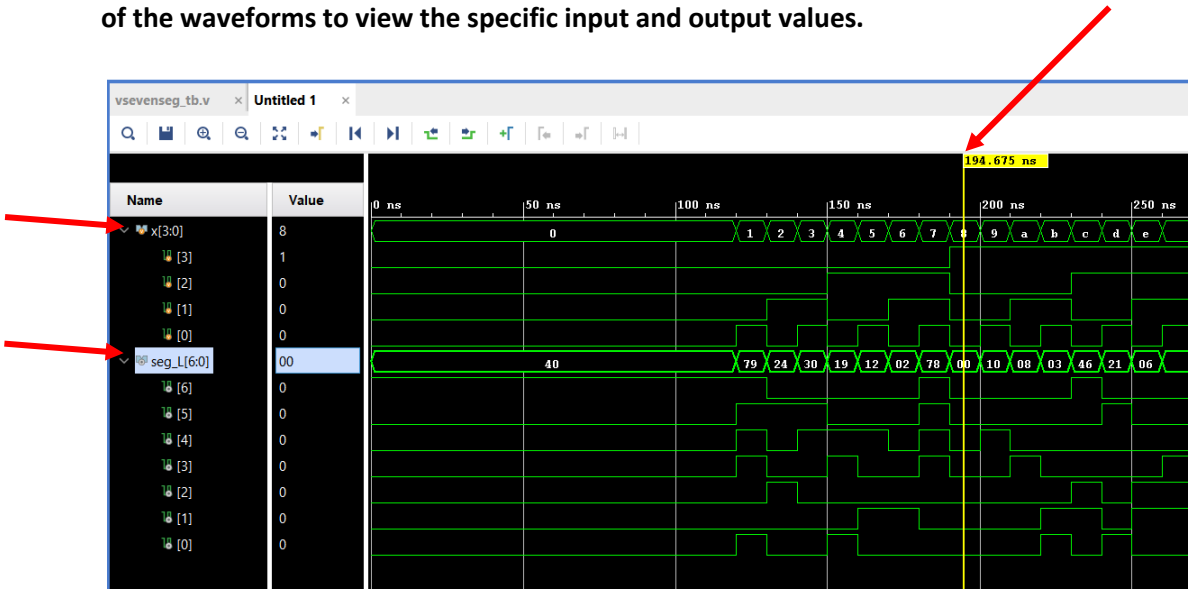


It is quite easy to verify the design using hexadecimal values:



Seg_L[6:0] segment order: g, **f**, e, **d**, **c**, b, a

22. Click > to expand the signals and separate the timing waveforms. Click on different time instances of the waveforms to view the specific input and output values.

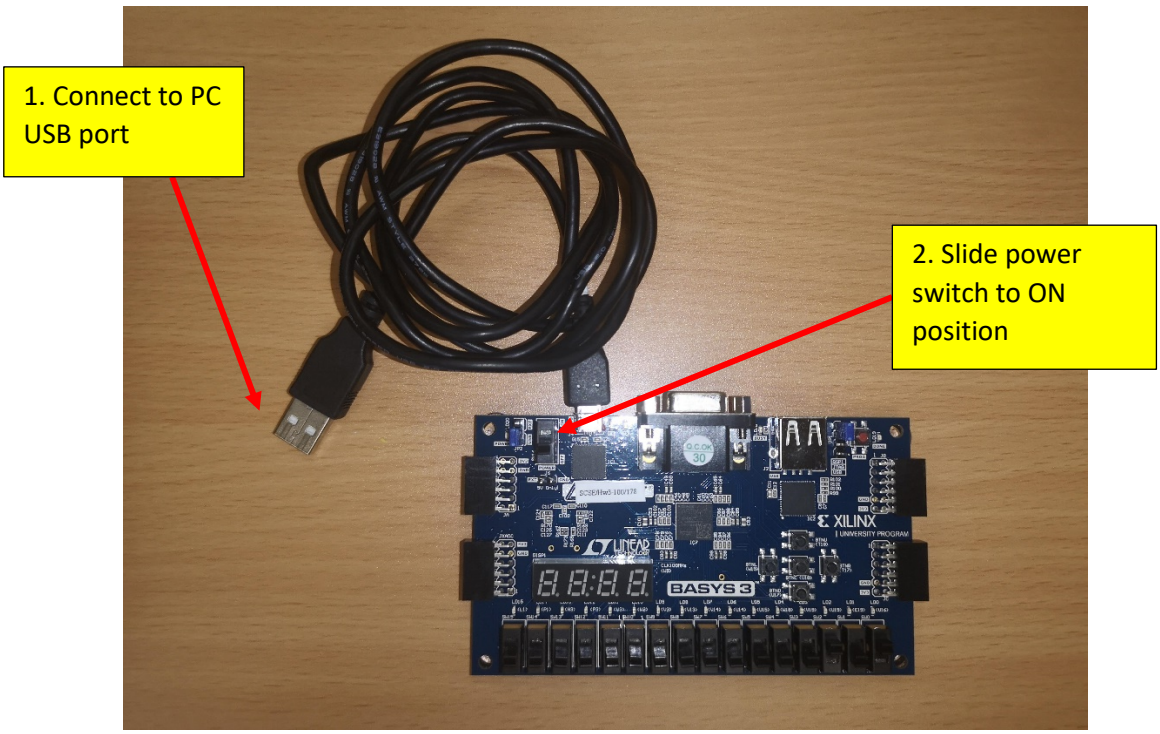


Note that the signal seg_L is active low, i.e. logic 0 (Low) means the segment should light up.

For example, when input=8, all 7 bits of seg_L are 0 since all 7 segments should light up.

Part C: Add in constraints for implementation on FPGA

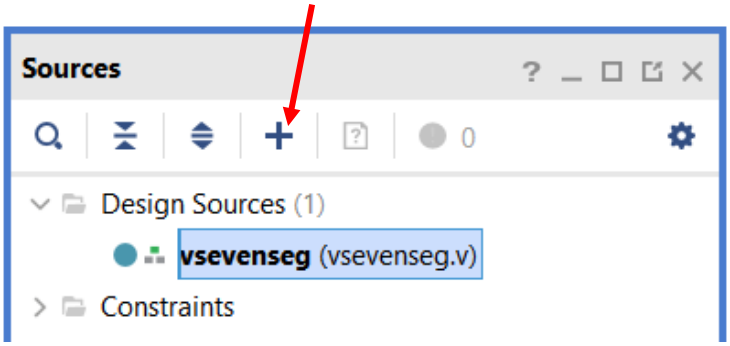
23. Plug the USB connector of the Basys3 board to the PC and turn on the power switch



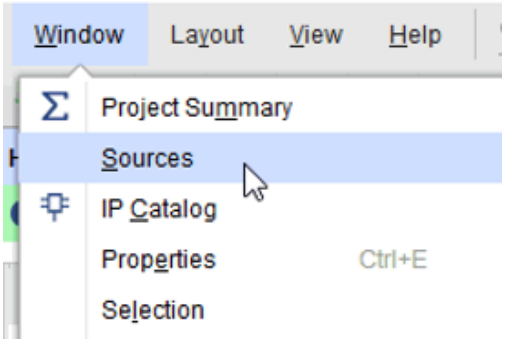
The power LED and the onboard 7-segment display will light up to show that it is working

24. Close the simulation window

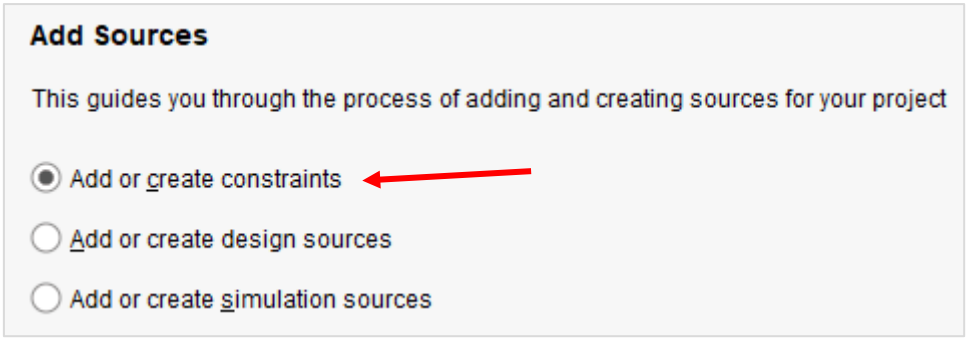
In Sources, click + to add the constraints file



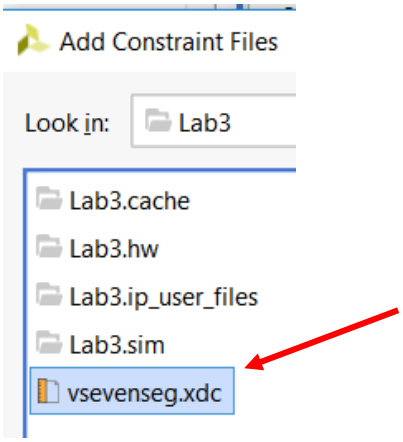
At any time, you may click **Window>Sources** on the top menu bar to call up the **Sources** window



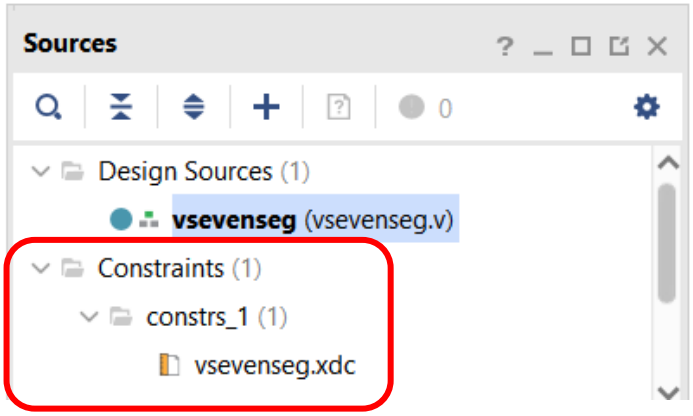
25. Select Constraints, click Next



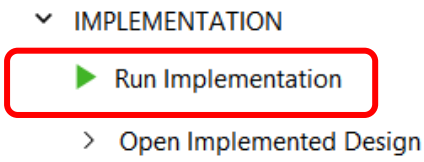
26. Click Add Files, select *vsevenseg.xdc*, click OK, click Finish
(You should have downloaded it from NTULearn and placed it in the same project subdirectory or folder)



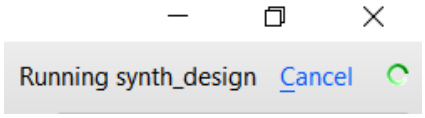
27. On Sources, click > to expand Constraints
You should see the constraints file *vsevenseg.xdc* added
Double click to open the file and view it in the workspace



28. On Flow Navigator, click Run Implementation
Click OK when prompted to launch synthesis first. Click OK and wait patiently.



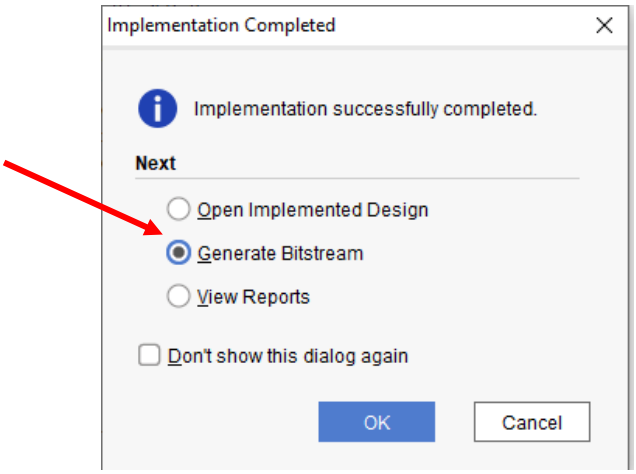
A green circle rotating on the top right corner of the application indicates that it's working



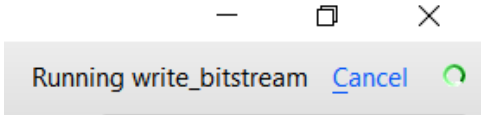
If Synthesis Failed, it could be due to a wrong part being selected in Step 8.

To correct it, click Project Summary, click Project part, select the correct part xc7a35tcpg236-1 and click OK, click Apply, click OK.
Repeat Step 28 after correcting the part.

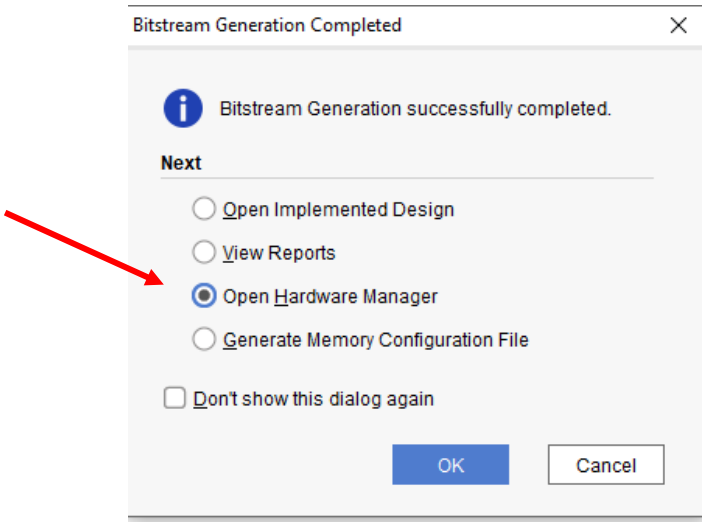
29. A dialogue box will pop up when implementation is successfully completed
Select Generate Bitstream and click OK, then click OK again.



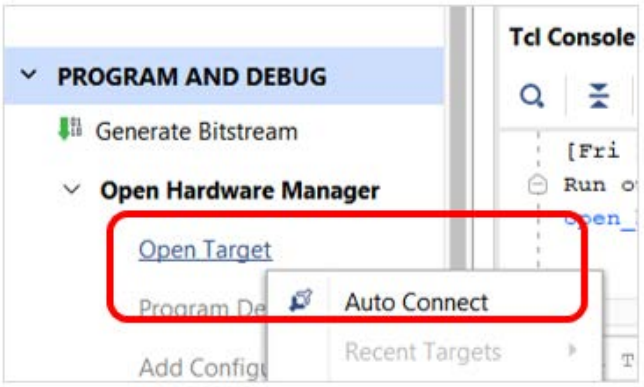
A green circle rotating on the top right corner of the application indicates that it's working



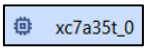
30. **This will pop up when the Bitstream is generated successfully**
    Select Open Hardware Manager and click OK

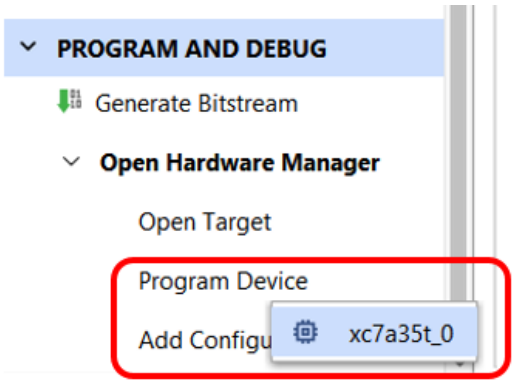


31. **On Flow Navigator, click Open Target and select Auto Connect**

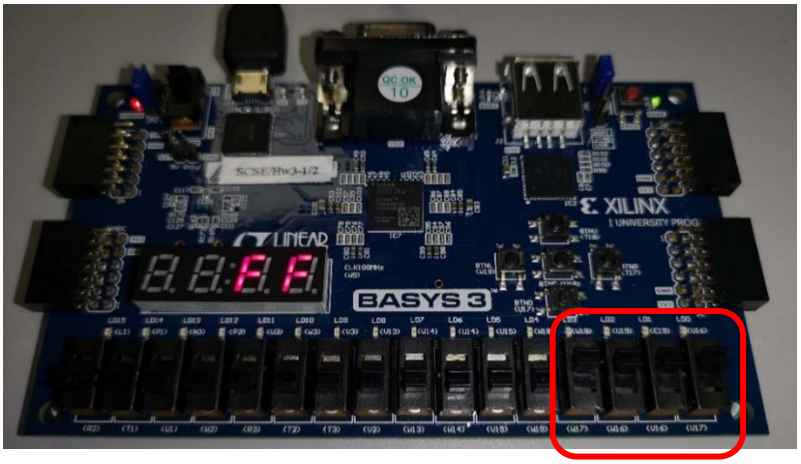


32.

33. **Click Program Device, click on the**  **device symbol**
    A dialogue box will pop up
    Check that the bitstream file name is correct and click Program



34. **When programming is done, you should see this if the switches SW3, SW2, SW1, SW0 are set to 1111:**

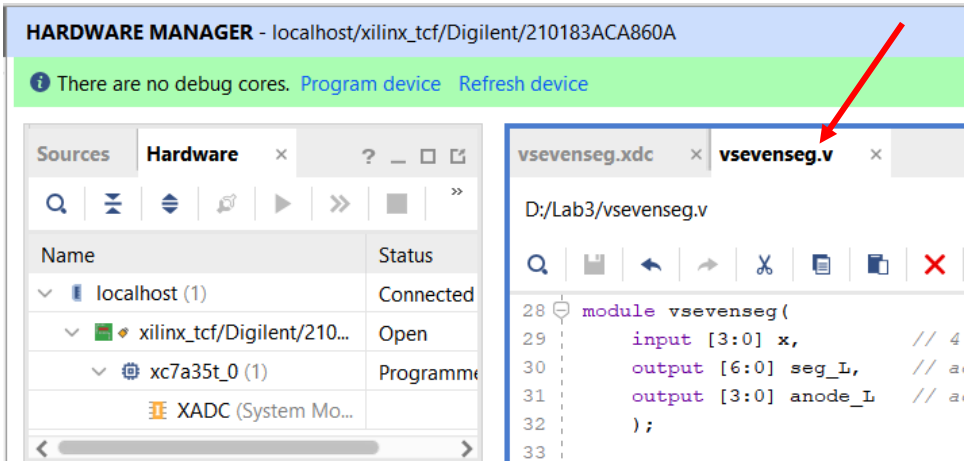


Input x[3:0] connected to SW3, SW2, SW1, SW0 which are set to **1 1 1 1**

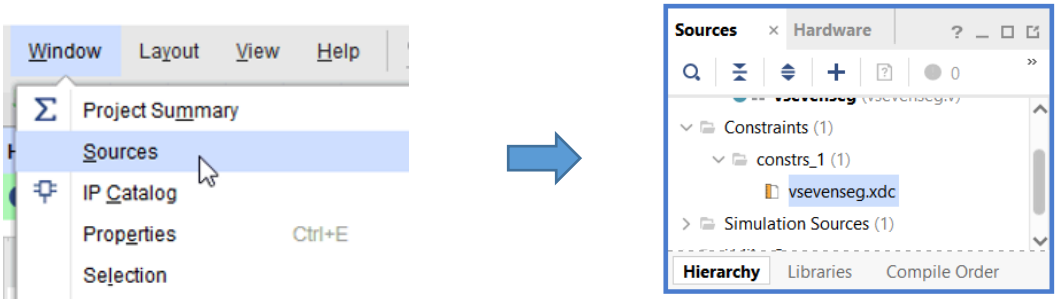
Set the switches to other values and verify that the displays are correct

Part D: Optional

35. **In the workspace, select the *vsevenseg.v* design file**



If the file is not already open in the workspace, you may click **Window>Sources** on the top menu bar to call up the **Sources** window, and double click on the file to open it



36. Edit the content of *vsevenseg.v* as follows and click Save (//comments need not be entered)

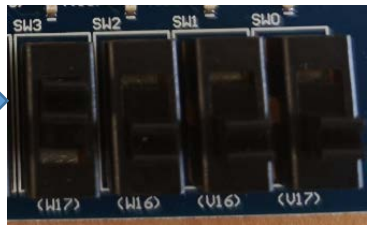
Note the changes required at these line numbers: 29, 30, 38, 41 and 43

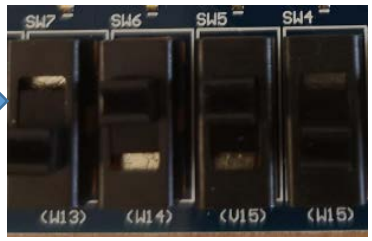
```
28    module vsevenseg(
29      input [3:0] a,b,        // 8 input switches: a left, b right
30      input left,             // select left digit if TRUE
31      output [6:0] seg_L,     // active low segment display
32      output [3:0] anode_L    // active low digit display
33      );
34
35  // declare internal active high segments
36  wire [6:0] seg;              // 1:on, 0:off
37  // value read from switches
38  wire [3:0] x;                // optional part
39
40  // turn on only the two rightmost digits - active low
41  assign anode_L = {2'b11, ~left, left};  //select left or right digit display
42  // select input a if left is TRUE, else select input b to display
43  assign x = left ? a : b;                // optional part
44
45  // pull cathode low to light up segment - active low
46  assign seg_L = ~seg;
```

No changes required below line 43

37. In the workspace, select the *vsevenseg.xdc* constraints file and modify its content as follows and then click Save. (Remove # at the start of a line to uncomment it. Port names must match design)

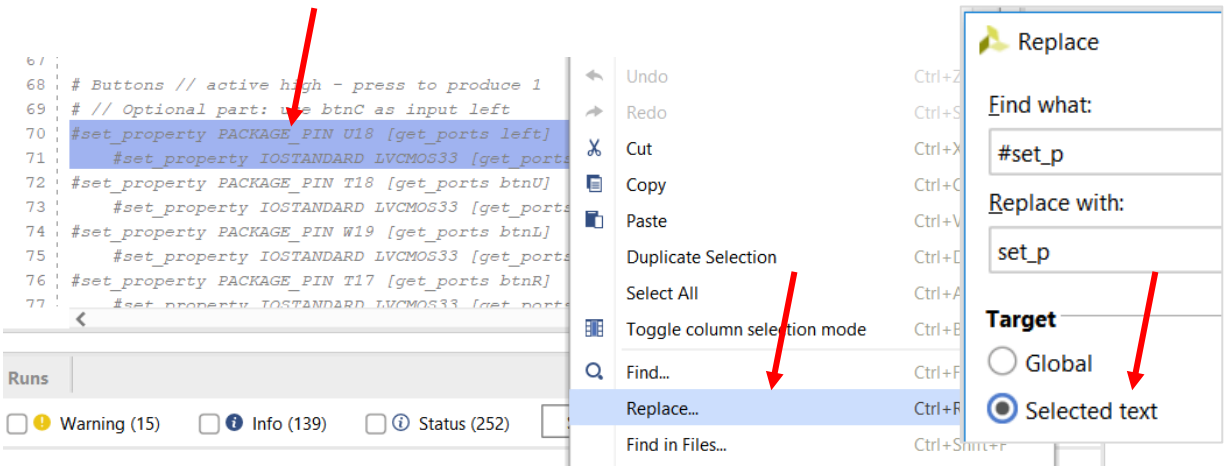
```
6   # Switches SW3-SW0
7   # // Optional part: rename input x to b
8   set_property PACKAGE_PIN V17 [get_ports {b[0]}]
9       set_property IOSTANDARD LVCMOS33 [get_ports {b[0]}]
10  set_property PACKAGE_PIN V16 [get_ports {b[1]}]
11      set_property IOSTANDARD LVCMOS33 [get_ports {b[1]}]
12  set_property PACKAGE_PIN W16 [get_ports {b[2]}]
13      set_property IOSTANDARD LVCMOS33 [get_ports {b[2]}]
14  set_property PACKAGE_PIN W17 [get_ports {b[3]}]
15      set_property IOSTANDARD LVCMOS33 [get_ports {b[3]}]
16
17  # Switches SW7-SW4
18  # // Optional part: use for input a
19  set_property PACKAGE_PIN W15 [get_ports {a[0]}]
20      set_property IOSTANDARD LVCMOS33 [get_ports {a[0]}]
21  set_property PACKAGE_PIN V15 [get_ports {a[1]}]
22      set_property IOSTANDARD LVCMOS33 [get_ports {a[1]}]
23  set_property PACKAGE_PIN W14 [get_ports {a[2]}]
24      set_property IOSTANDARD LVCMOS33 [get_ports {a[2]}]
25  set_property PACKAGE_PIN W13 [get_ports {a[3]}]
26      set_property IOSTANDARD LVCMOS33 [get_ports {a[3]}]
```



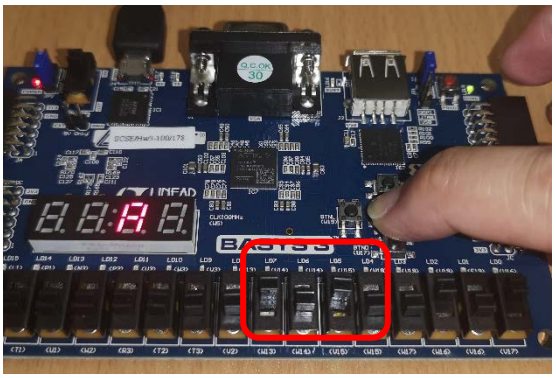



```
68  # Buttons // active high - press to produce 1
69  # // Optional part: use btnC as input left
70  set_property PACKAGE_PIN U18 [get_ports left]
71      set_property IOSTANDARD LVCMOS33 [get_ports left]
72  #set_property PACKAGE_PIN T18 [get_ports btnU]
73      #set_property IOSTANDARD LVCMOS33 [get_ports btnU]
```

Port name can be easily modified by selecting the required lines, then right click to **Replace.** Click **Selected text** so that other lines are not affected.

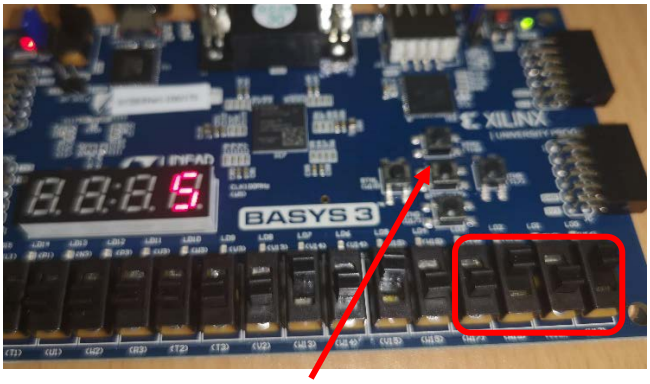


38. **On Flow Navigator, click Generate Bitstream and wait patiently**
39. **When Bitstream is successfully generated, click Cancel on dialogue box**
40. **On Flow Navigator, click Program Device to program the device with the newly generated bitstream. The circuit should behave as follows:**



Pressing BTNC makes input left=1 (i.e. TRUE)

Input a = SW7, SW6, SW5, SW4 = **1 0 1 0**



Not pressing BTNC makes input left=0 (i.e. FALSE)

Input b= SW3, SW2, SW1, SW0 = **0 1 0 1**

Notes:

1. Whenever you have modified the Verilog design file (e.g. vsevenseg.v) and/or the constraints file (e.g. vsevenseg.xdc), you need to generate the bitstream file (step 37) and program the FPGA (step 39) in order to observe the effect of your modifications on the circuit's behaviour.

2. Removing a file from a project is not the same as deleting the file from the project directory. You can safely remove a file from the project (and leave it in the project directory) and then add it back to the project later if needed.