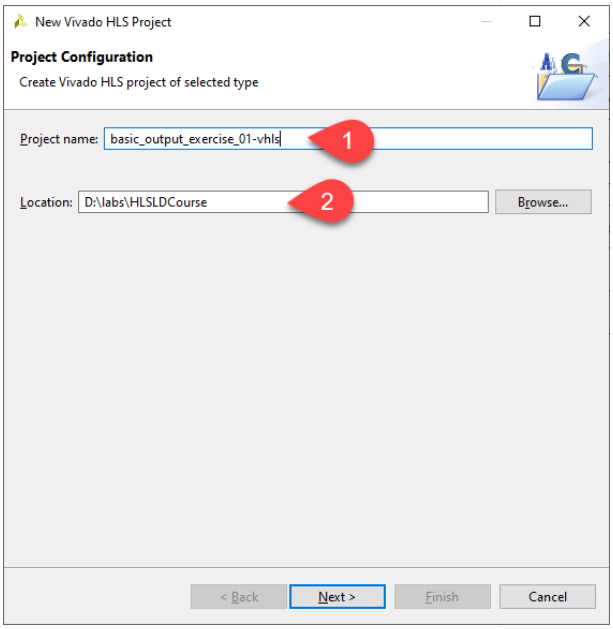
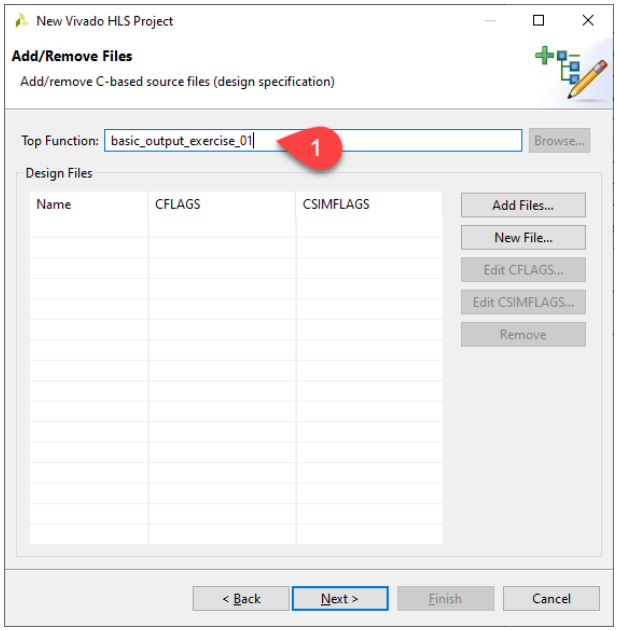
## Exercise 1

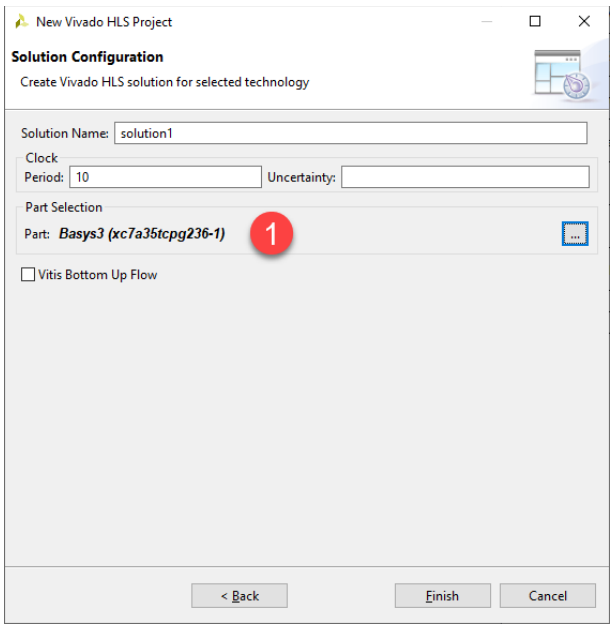
1- Create a Vivado-HLS project with the name of "*basic_output_exercise_01-vhls*".




2- Choose "*basic_output_exercise_01*" as the top-function name.



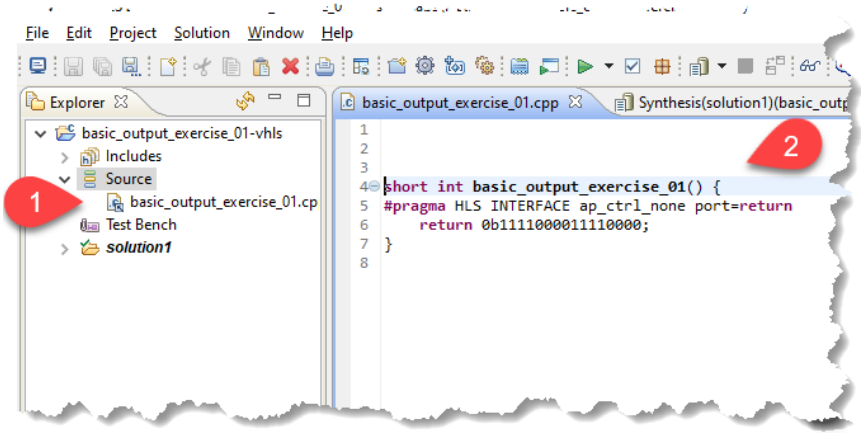
1- Choose the Basys-3 board as the target FPGA platform.



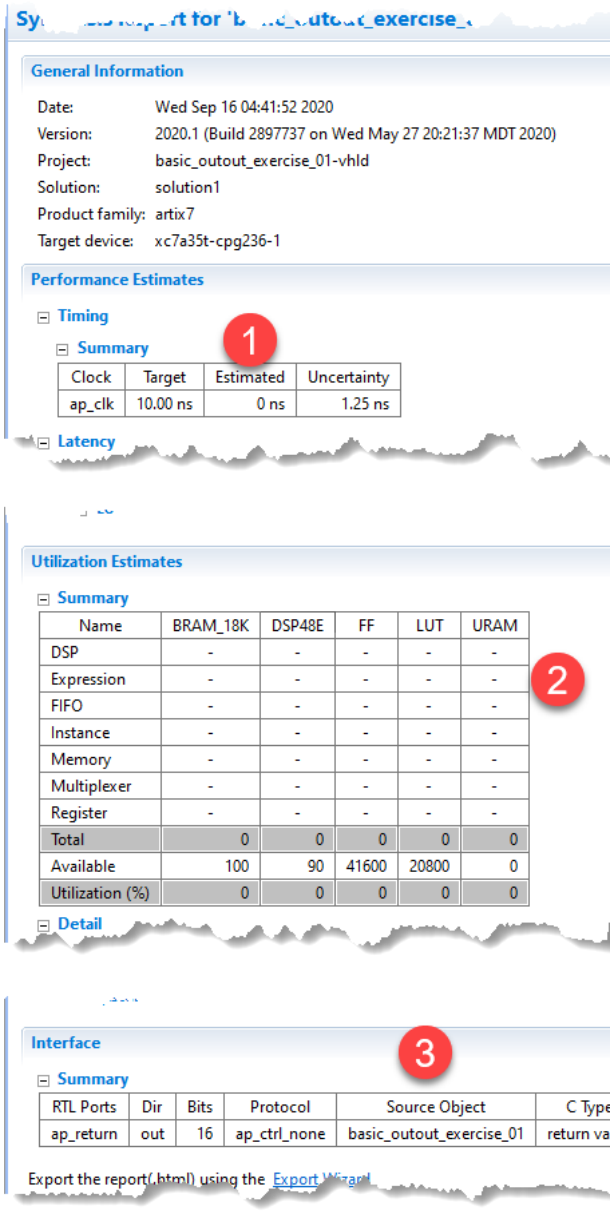

2- Create a new design file under the **Source** folder with the name of "*basic_output_exercise_01.cpp*".

3- Add the top-function as follows. As the function should return a 16-bit value, it has used short int as its return data type.

```
1  short int basic_output_exercise_01() {
2  #pragma HLS INTERFACE ap_ctrl_none port=return
3     return 0b1111000011110000;
4  }
```



4- Synthesise the code. The report should be as follows.



**General Information**

| | |
|---|---|
| Date: | Wed Sep 16 04:41:52 2020 |
| Version: | 2020.1 (Build 2897737 on Wed May 27 20:21:37 MDT 2020) |
| Project: | basic_outout_exercise_01-vhld |
| Solution: | solution1 |
| Product family: | artix7 |
| Target device: | xc7a35t-cpg236-1 |

**Performance Estimates**

Timing

Summary

1

| Clock | Target | Estimated | Uncertainty |
|---|---|---|---|
| ap_clk | 10.00 ns | 0 ns | 1.25 ns |

Latency

**Utilization Estimates**

Summary

| Name | BRAM_18K | DSP48E | FF | LUT | URAM |
|---|---|---|---|---|---|
| DSP | - | - | - | - | - |
| Expression | - | - | - | - | - |
| FIFO | - | - | - | - | - |
| Instance | - | - | - | - | - |
| Memory | - | - | - | - | - |
| Multiplexer | - | - | - | - | - |
| Register | - | - | - | - | - |
| Total | 0 | 0 | 0 | 0 | 0 |
| Available | 100 | 90 | 41600 | 20800 | 0 |
| Utilization (%) | 0 | 0 | 0 | 0 | 0 |

2

Detail

**Interface**

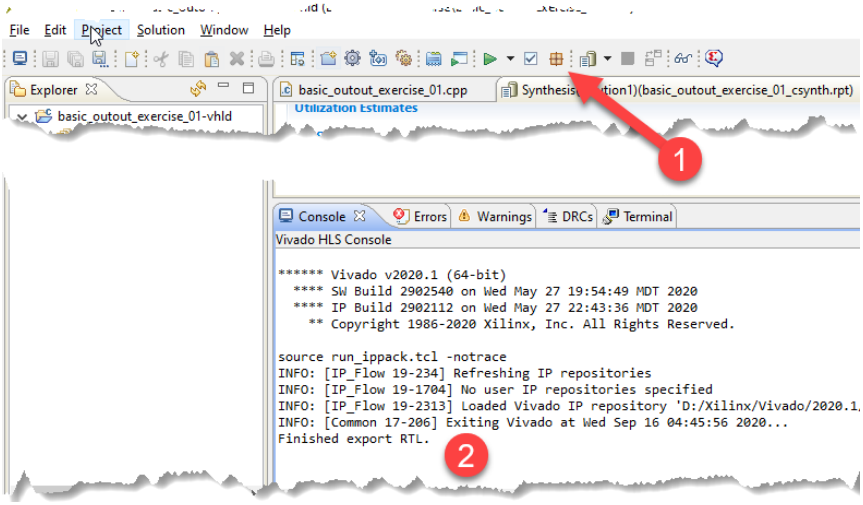
3

Summary

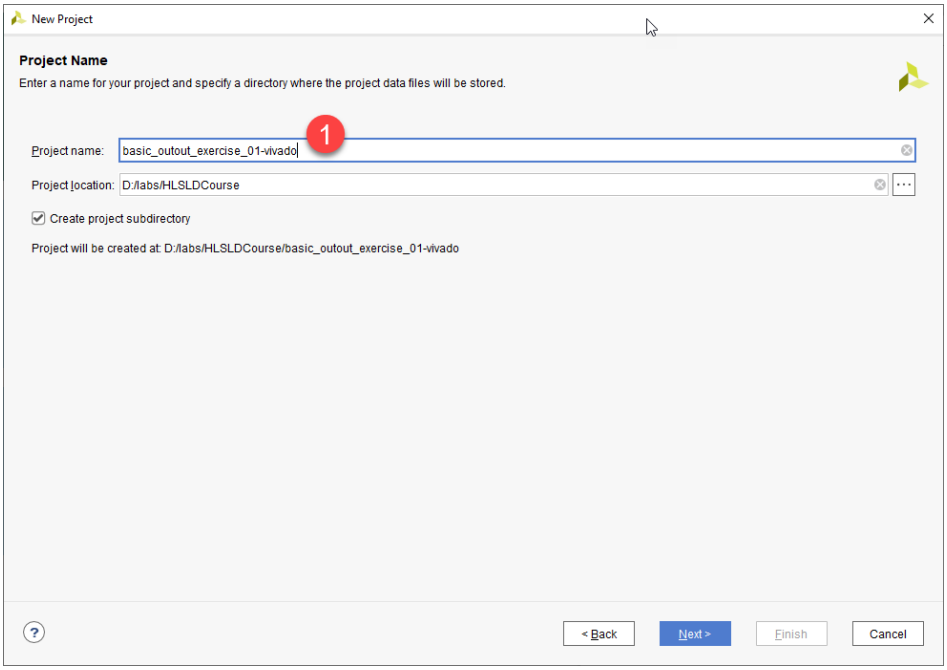
| RTL Ports | Dir | Bits | Protocol | Source Object | C Type |
|---|---|---|---|---|---|
| ap_return | out | 16 | ap_ctrl_none | basic_outout_exercise_01 | return va |

Export the report(.html) using the Export Wizard

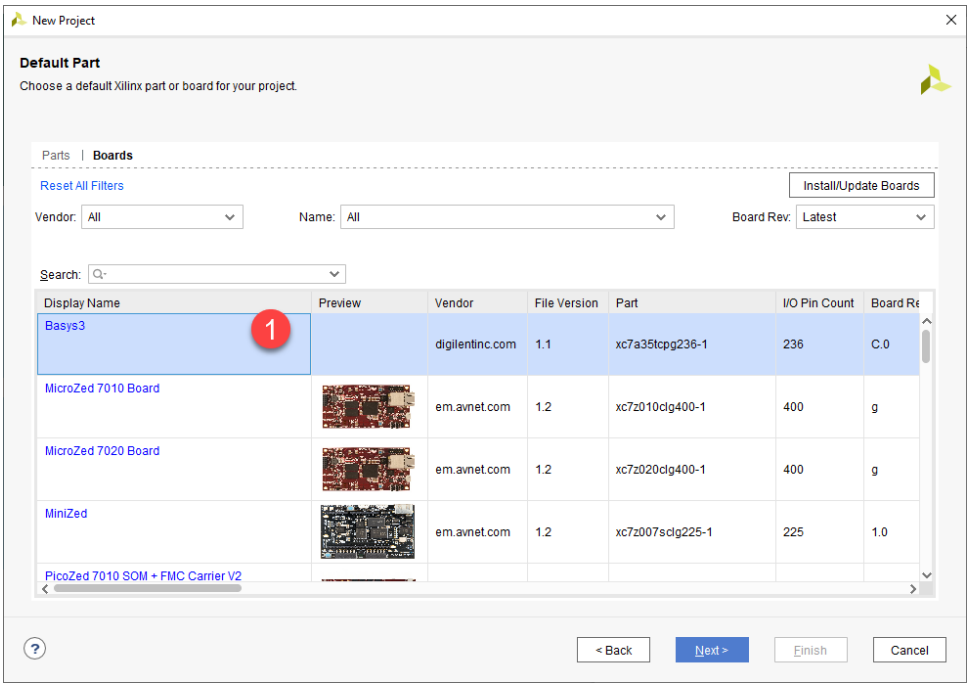
5- Generate the RTL IP and be ready for creating the Vivado project.



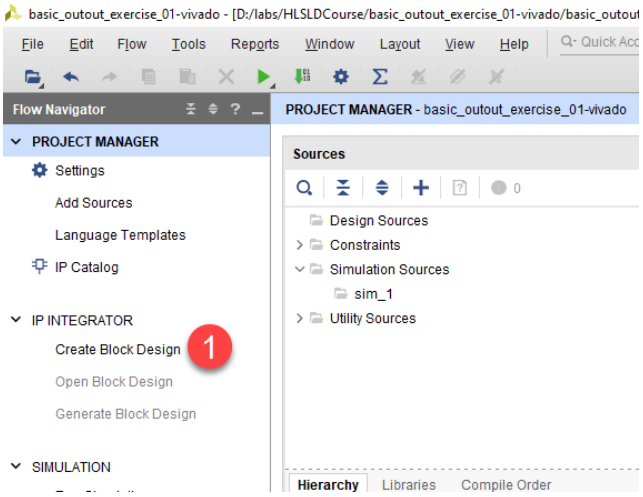

6- Create a Vivado project. Choose "*basic_output_exercise_01-vivado*" as the name.



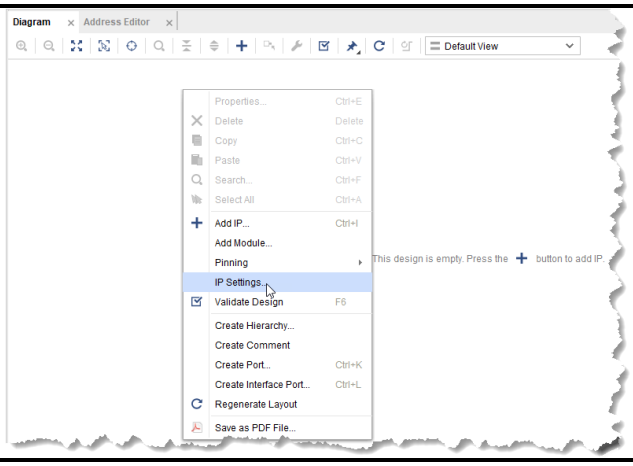
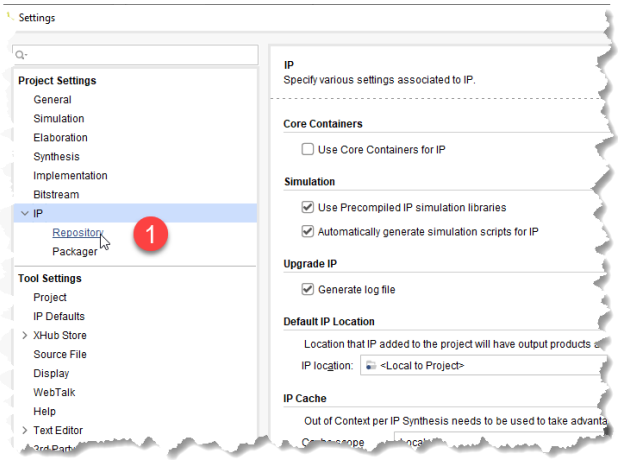
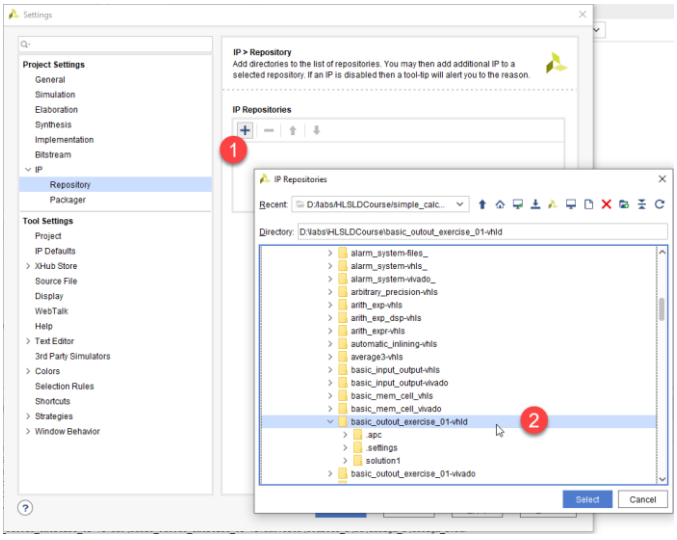

7- Choose the Basys3 board as the target FPGA.



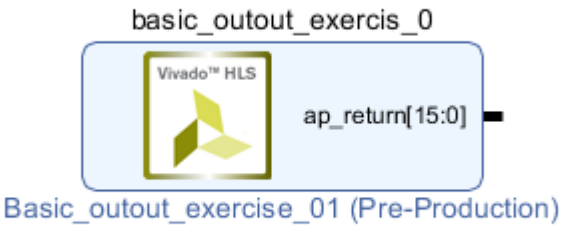

8- Create a block design.



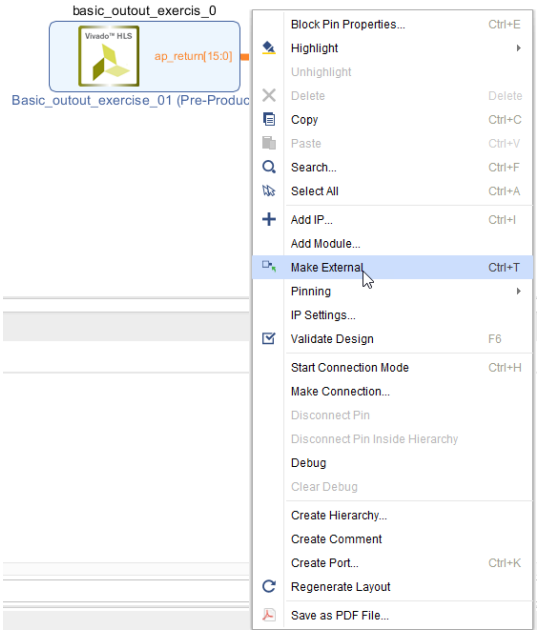
9- Add the generated RTL-IP to the Vivado repository.

| | |
|---|---|
| Right-click inside the Diagram area and select IP Settings |   |
| In the setting dialog, select Repository under the IP option. |   |
| Click on the plus icon and brows the Vivado-HLS project folder. The Vivado tool searches the folder and add all generated IPs to its repositoty. |   |

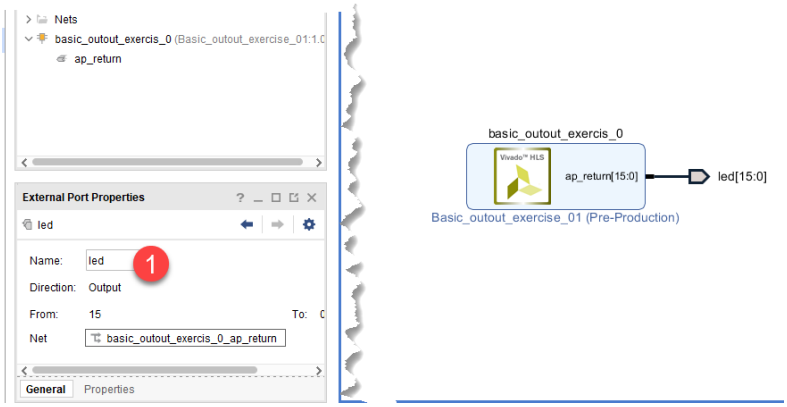
10- Add the IP to the Diagram area by clicking on the plug icon in the middle of the diagram area.



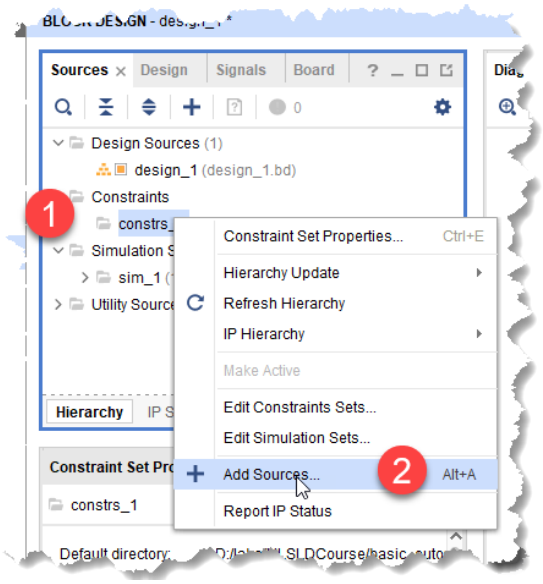
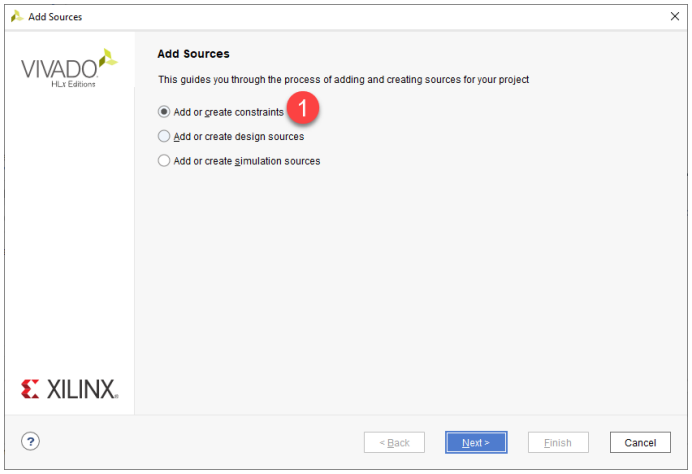

11- Select on the design port and make that external.



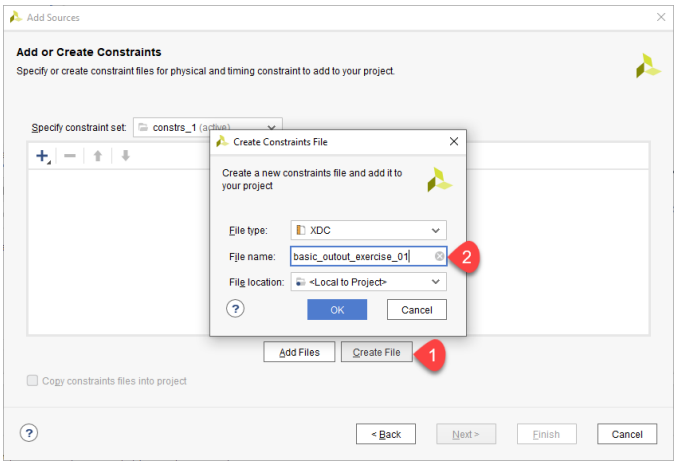
12- Change the port name to "*led*".




13- Now create a constraint file

| | |
|---|---|
| Right-click on the Constraints folder in the Sources tab and select the **Add Sources** option. |  |
| Make sure that the "Add or create constraints" option is selected. |  |

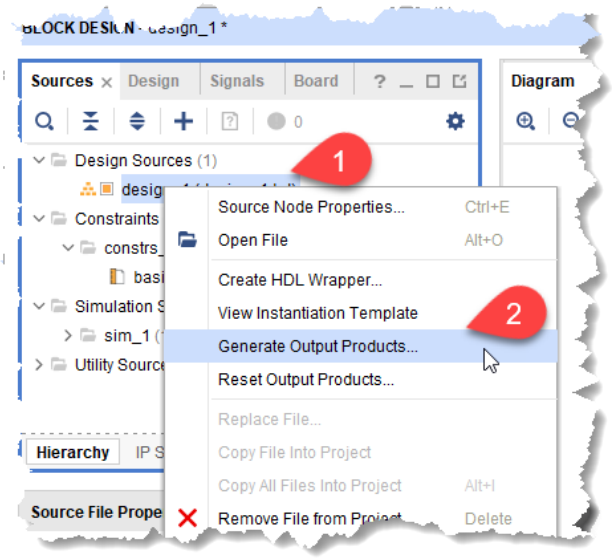
Click on the **Create File** button and select "basic_output_exercise_01" as the name.



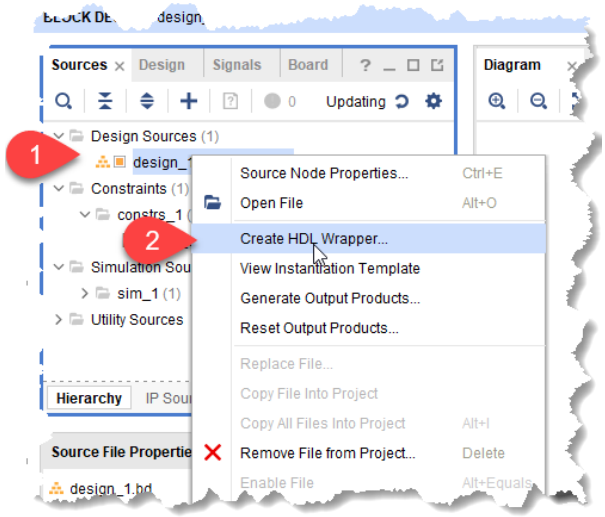

14- Add these constraints inside the created file

```
# LEDs
set_property PACKAGE_PIN U16 [get_ports {led[0]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {led[0]}]
set_property PACKAGE_PIN E19 [get_ports {led[1]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {led[1]}]
set_property PACKAGE_PIN U19 [get_ports {led[2]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {led[2]}]
set_property PACKAGE_PIN V19 [get_ports {led[3]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {led[3]}]
set_property PACKAGE_PIN W18 [get_ports {led[4]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {led[4]}]
set_property PACKAGE_PIN U15 [get_ports {led[5]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {led[5]}]
set_property PACKAGE_PIN U14 [get_ports {led[6]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {led[6]}]
set_property PACKAGE_PIN V14 [get_ports {led[7]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {led[7]}]
set_property PACKAGE_PIN V13 [get_ports {led[8]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {led[8]}]
set_property PACKAGE_PIN V3 [get_ports {led[9]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {led[9]}]
set_property PACKAGE_PIN W3 [get_ports {led[10]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {led[10]}]
set_property PACKAGE_PIN U3 [get_ports {led[11]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {led[11]}]
set_property PACKAGE_PIN P3 [get_ports {led[12]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {led[12]}]
set_property PACKAGE_PIN N3 [get_ports {led[13]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {led[13]}]
set_property PACKAGE_PIN P1 [get_ports {led[14]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {led[14]}]
set_property PACKAGE_PIN L1 [get_ports {led[15]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {led[15]}]
```

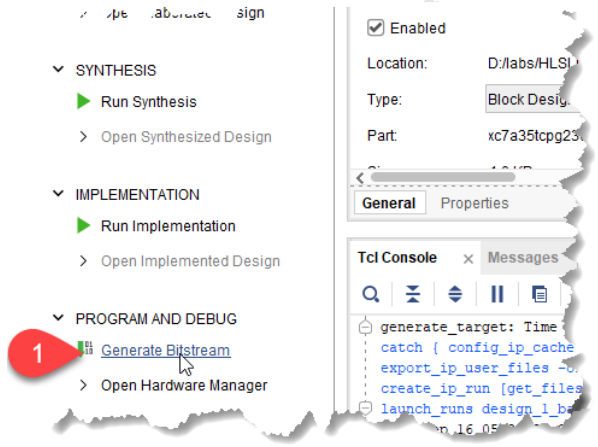
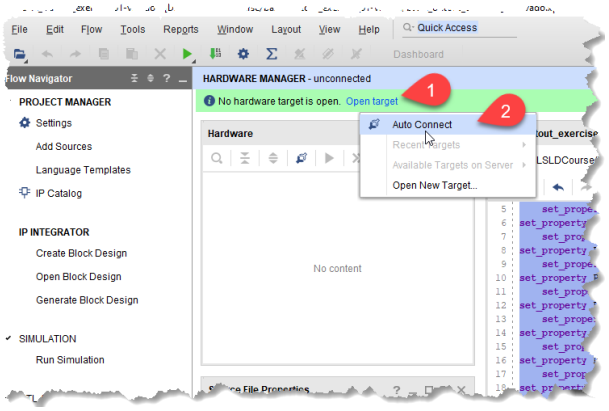
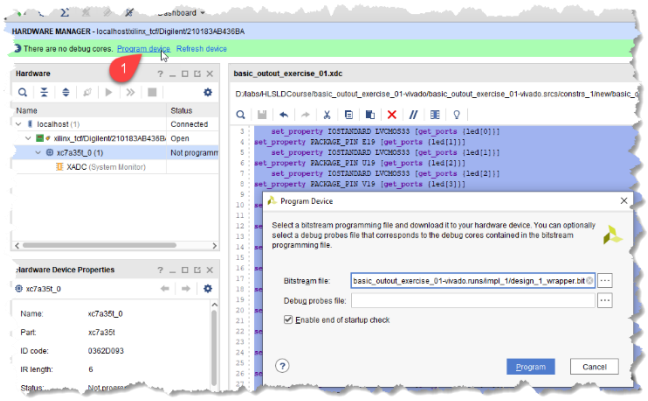
15- Right-click on the design_1 under the Design Source folder in the Sources tab and select “Generate Output Products”



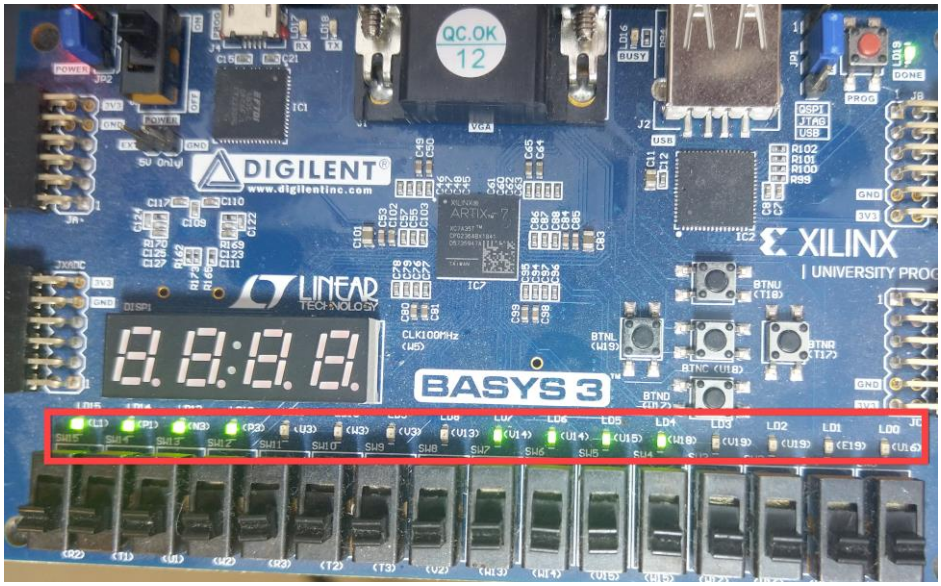

16- Right-click on the design_1 under the Design Source folder in the Sources tab and select “Create HDL Wrapper”



17- Now generate the bitstream and program the FPGA

| | |
|---|---|
| Click on the Generate Bitstream |  |
| Then click on the “Open Hardware Manager” | |
| Click on the “Open target” link and select the Auto-Connect |  |
| Finally click on the “Program device” link to program the FPGA. |  |

18- Now examine the board and check the ON and OFF patten on the LEDs.



### Exercise 2

Follow the steps of exercise 1 as follows

1- Create a new Vivado-HLS project with the name of "*basic_output_exercise_02-vhls*".
2- Choose "*basic_output_exercise_02*" as the top-function name
3- As all C/C++ data types are limited to 8-bit boundaries except the **bool** C++ data type, we use this data type to define five separate bits to drive five LEDs.

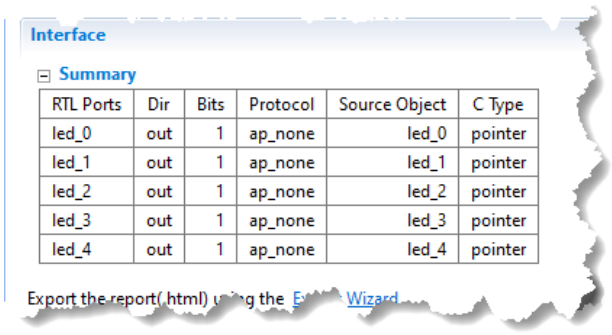
```
void basic_output_exercise_02(
            bool &led_0,
            bool &led_1,
            bool &led_2,
            bool &led_3,
            bool &led_4,) {

#pragma HLS INTERFACE ap_ctrl_none port=return
#pragma HLS INTERFACE ap_none port=led_0
#pragma HLS INTERFACE ap_none port=led_1
#pragma HLS INTERFACE ap_none port=led_2
#pragma HLS INTERFACE ap_none port=led_3
#pragma HLS INTERFACE ap_none port=led_4

      led_0 = 1;
      led_1 = 1;
      led_2 = 0;
      led_3 = 1;
      led_4 = 0;

}
```

4- The synthesis report shows eight one-bit output ports.


Interface

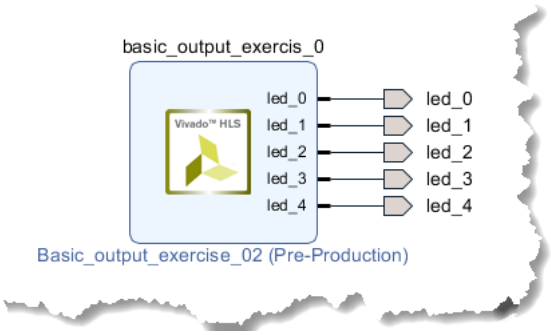
Summary

| RTL Ports | Dir | Bits | Protocol | Source Object | C Type |
|---|---|---|---|---|---|
| led_0 | out | 1 | ap_none | led_0 | pointer |
| led_1 | out | 1 | ap_none | led_1 | pointer |
| led_2 | out | 1 | ap_none | led_2 | pointer |
| led_3 | out | 1 | ap_none | led_3 | pointer |
| led_4 | out | 1 | ap_none | led_4 | pointer |

Export the report( html) using the Export Wizard

5- Generate the RTL IP.
6- Create a new Vivado project with the name of "*basic_output_exercise_02-vivado*".

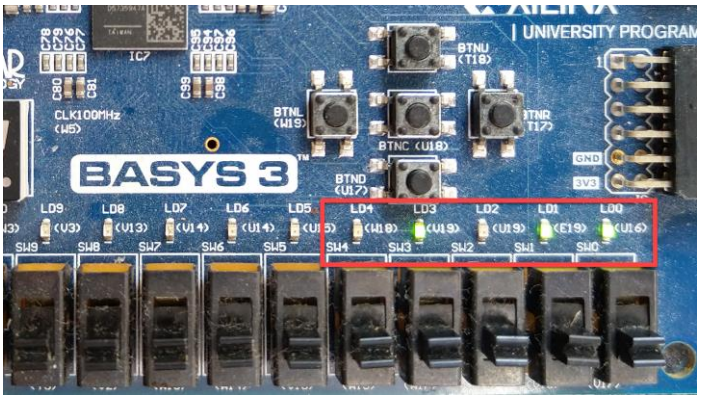
7- Add the generated IP to the Vivado repository. Also, add the IP into the Diagram area and make all its ports external and change their names as shown in the following figure.




8- Create a new constraint file and add these physical constraints

```
# LEDs
set_property PACKAGE_PIN U16 [get_ports {led_0}]
    set_property IOSTANDARD LVCMOS33 [get_ports {led_0}]
set_property PACKAGE_PIN E19 [get_ports {led_1}]
    set_property IOSTANDARD LVCMOS33 [get_ports {led_1}]
set_property PACKAGE_PIN U19 [get_ports {led_2}]
    set_property IOSTANDARD LVCMOS33 [get_ports {led_2}]
set_property PACKAGE_PIN V19 [get_ports {led_3}]
    set_property IOSTANDARD LVCMOS33 [get_ports {led_3}]
set_property PACKAGE_PIN W18 [get_ports {led_4}]
    set_property IOSTANDARD LVCMOS33 [get_ports {led_4}]
```

9- Generate the output product, then Create the HDL Wrapper. After that, generate the bitstream and finally program the FPGA and check the LEDs.


UNIVERSITY PROGRAM
BASYS 3
CLK100MHz
(W5)
BTNU
(T18)
BTNL
(W19)
BTNC (U18)
BTNR
T17
BTND
(U17)
GND
3V3
LD9
LD8
LD7
LD6
LD5
LD4
LD3
LD2
LD1
LD0
SW9
SW8
SW7
SW6
SW5
SW4
SW3
SW2
SW1
SW0

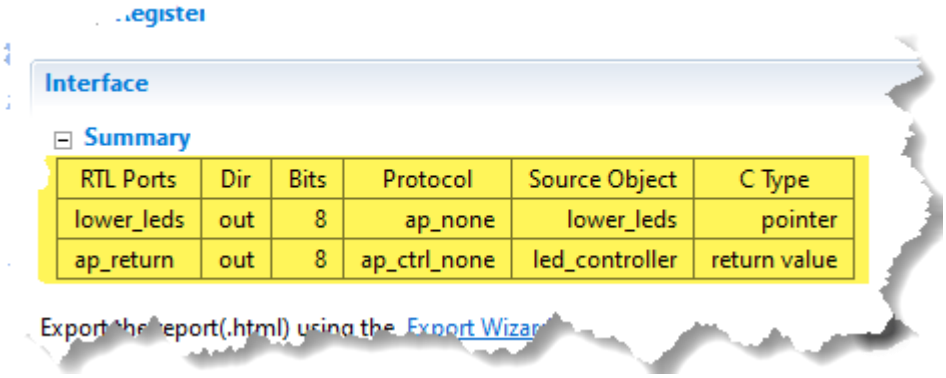
### Exercise 3

Follow the steps explained in the first exercise.

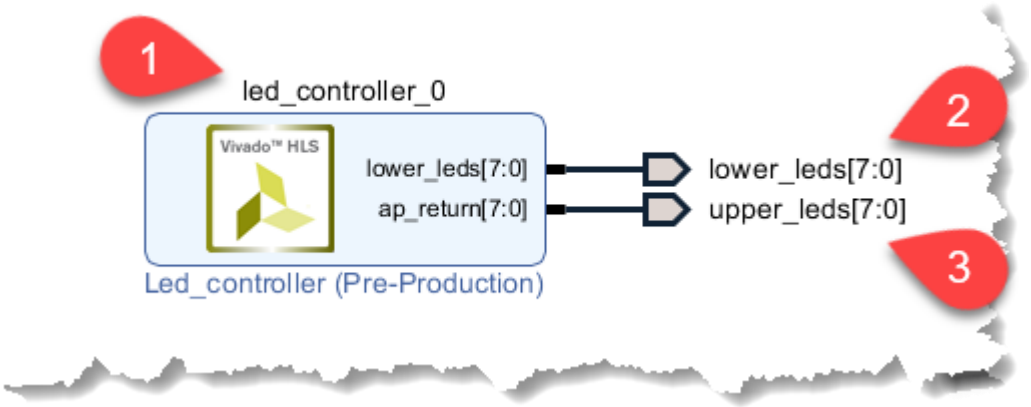
1- Choose *basic_output_exercise_03-vhls* as the Vivado-HLS project name.
2- Choose *led_controller* as the top-function name.
3- The top-function would be as follows.

```
1  char led_controller(char &lower_leds) {
2  #pragma HLS INTERFACE ap_ctrl_none port=return
3  #pragma HLS INTERFACE ap_none port=lower_leds
       lower_leds = 0b10110110;
4      return 0b01011;
5  }
```

4- The HLS report shows two output ports for the design.




5- Choose “basic_output_exercise_03-vivado” as the Vivado project name.
6- Note that the name of the IP is led_controller.
7- Change the port names as the following figure




8- The physical constraints are as follows.
Please look at the port names and the indices of the upper_leds in the constraints.

```
# LEDs
set_property PACKAGE_PIN U16 [get_ports {lower_leds[0]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {lower_leds[0]}]
set_property PACKAGE_PIN E19 [get_ports {lower_leds[1]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {lower_leds[1]}]
set_property PACKAGE_PIN U19 [get_ports {lower_leds[2]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {lower_leds[2]}]
set_property PACKAGE_PIN V19 [get_ports {lower_leds[3]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {lower_leds[3]}]
set_property PACKAGE_PIN W18 [get_ports {lower_leds[4]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {lower_leds[4]}]
set_property PACKAGE_PIN U15 [get_ports {lower_leds[5]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {lower_leds[5]}]
set_property PACKAGE_PIN U14 [get_ports {lower_leds[6]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {lower_leds[6]}]
set_property PACKAGE_PIN V14 [get_ports {lower_leds[7]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {lower_leds[7]}]
set_property PACKAGE_PIN V13 [get_ports {upper_leds[0]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {upper_leds[0]}]
set_property PACKAGE_PIN V3 [get_ports {upper_leds[1]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {upper_leds[1]}]
set_property PACKAGE_PIN W3 [get_ports {upper_leds[2]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {upper_leds[2]}]
set_property PACKAGE_PIN U3 [get_ports {upper_leds[3]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {upper_leds[3]}]
set_property PACKAGE_PIN P3 [get_ports {upper_leds[4]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {upper_leds[4]}]
set_property PACKAGE_PIN N3 [get_ports {upper_leds[5]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {upper_leds[5]}]
set_property PACKAGE_PIN P1 [get_ports {upper_leds[6]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {upper_leds[6]}]
set_property PACKAGE_PIN L1 [get_ports {upper_leds[7]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {upper_leds[7]}]
```

9- Generate the bitstream and program the board.