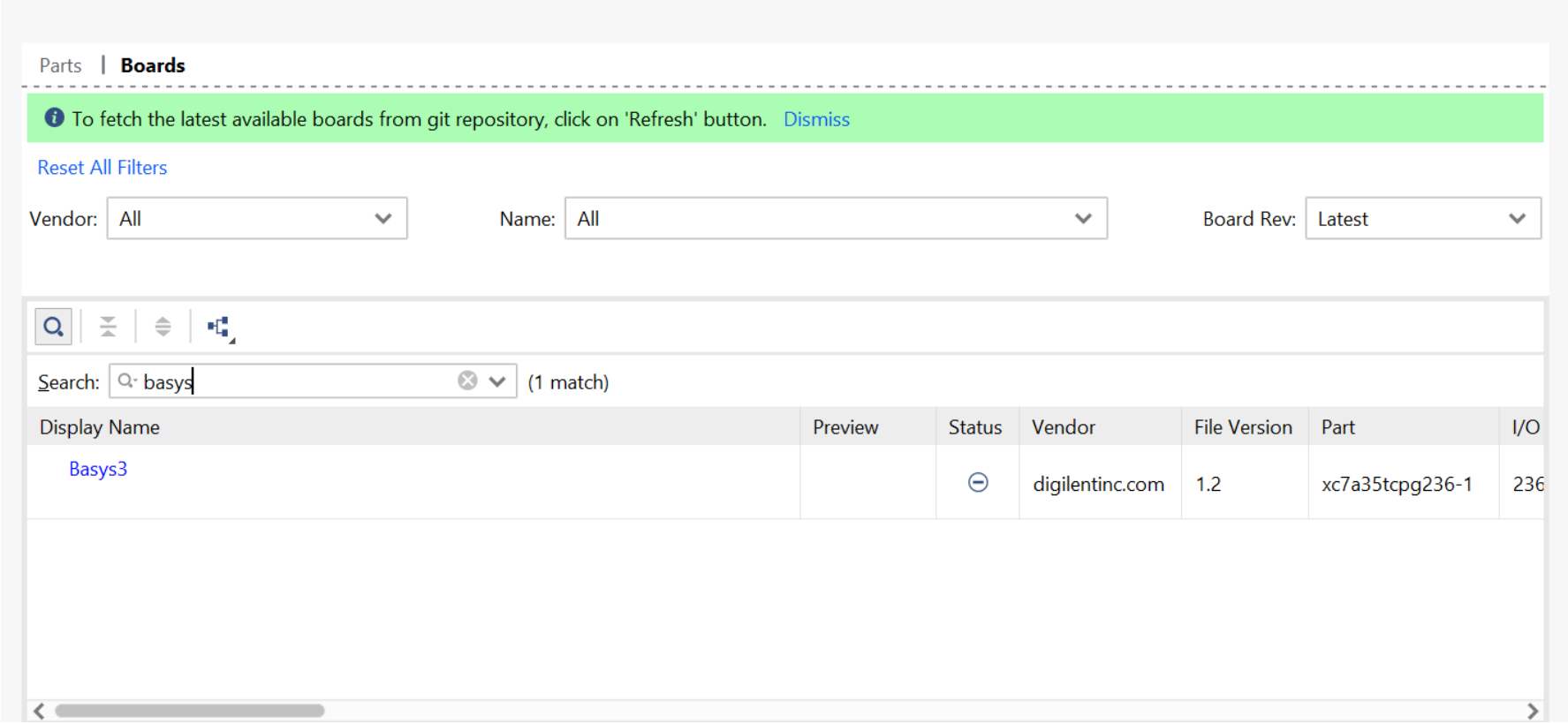
ELECH490 - Digital architectures and design

Intro to Exercise 1

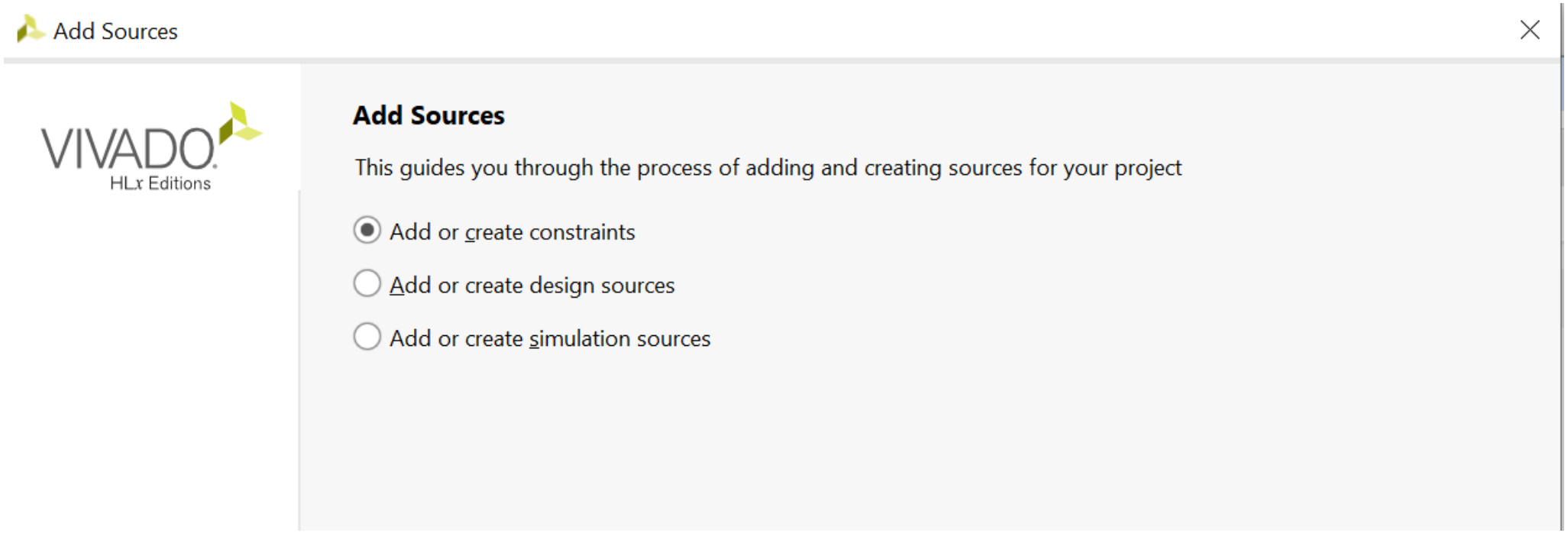
1. Create a new project

Create a new project and select the **Basys3** board as usually



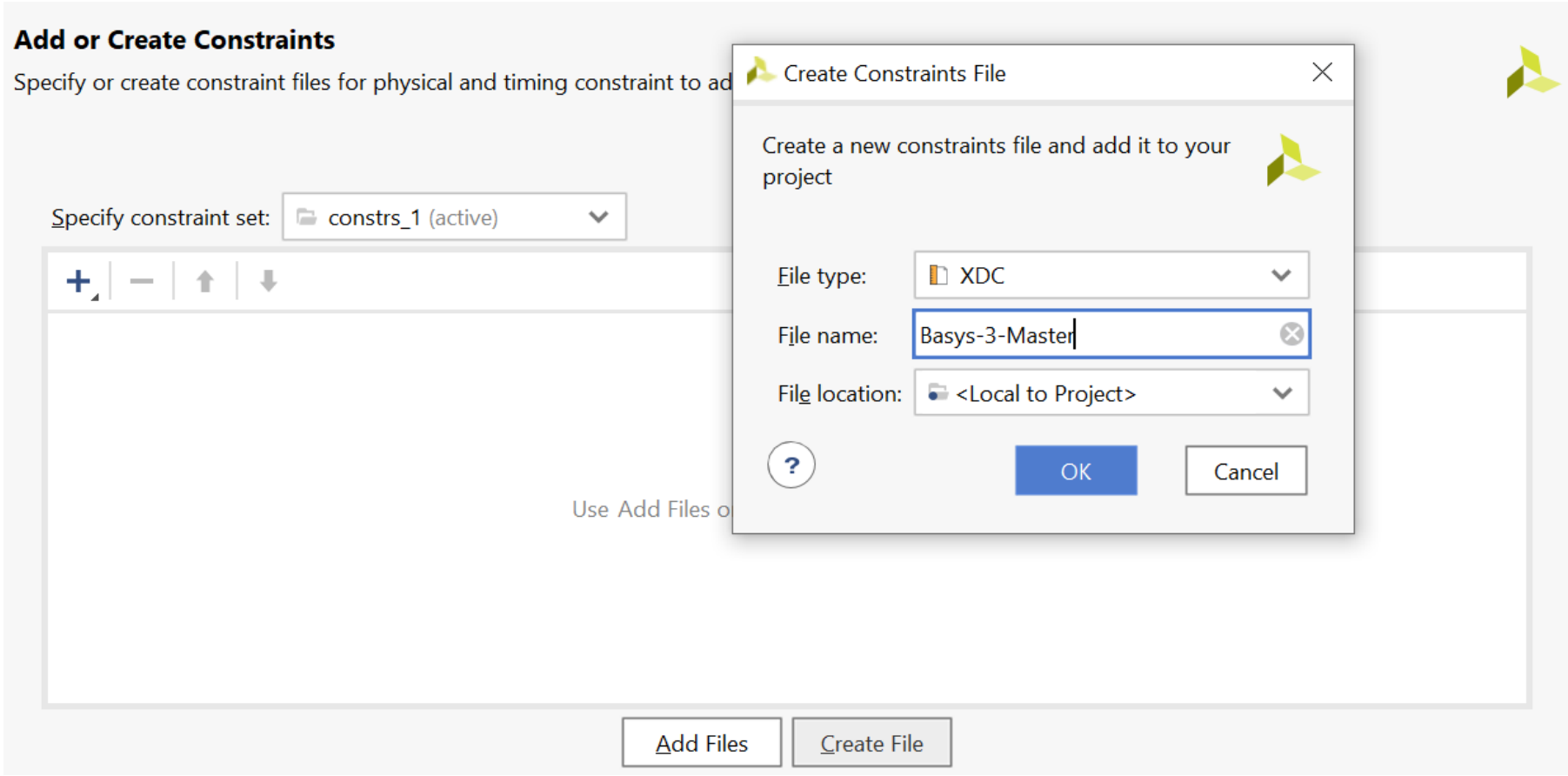
2. Add constraints

Go to **Add Sources** then select **Add or create constraints**



2. Add constraints

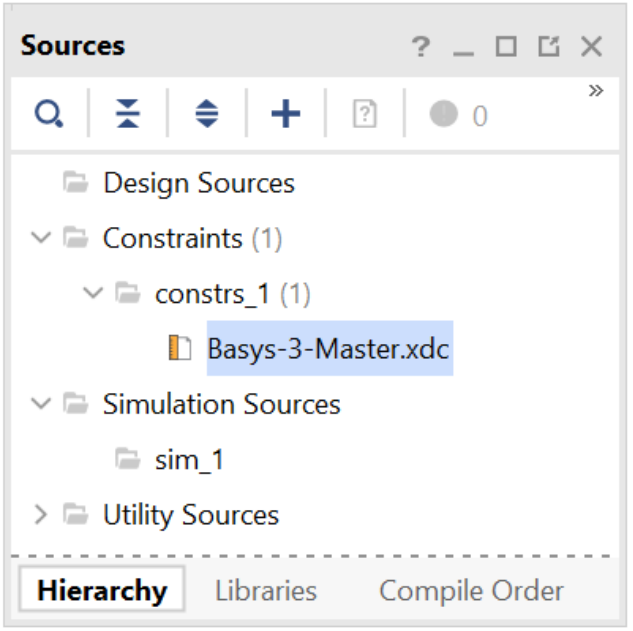
Create a new Constraints File



2. Add constraints

Copy the file given here : https://github.com/Digilent/digilent-xdc/blob/master/Basys-3-Master.xdc

Open the Constraints File you have created and then paste the file you just have copied.



3. Uncommenting lines of pins you use

Uncommend lines of pins that you need to use

In exercise 1 you need the clock, the LED0 and the right button

```
## Clock signal
set_property -dict { PACKAGE_PIN W5   IOSTANDARD LVCMOS33 } [get_ports clk]
#create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]


## LEDs
set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports {led0}]


##Buttons
#set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports btnC]
#set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVCMOS33 } [get_ports btnU]
#set_property -dict { PACKAGE_PIN W19   IOSTANDARD LVCMOS33 } [get_ports btnL]
set_property -dict { PACKAGE_PIN T17   IOSTANDARD LVCMOS33 } [get_ports btnR]
#set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVCMOS33 } [get_ports btnD]
```

4. Write your VHDL module

Create a new VHDL design Source File

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity LED_0_controller is
    Port (
    clk: in STD_LOGIC;
    btnR : in STD_LOGIC;
    led0 : out STD_LOGIC
    );
end LED_0_controller;
```

```
architecture Behavioral of LED_0_controller is

begin

process(clk) is
begin
    if rising_edge(clk) then
       led0 <= btnR;
    end if;
end process;

end Behavioral;
```

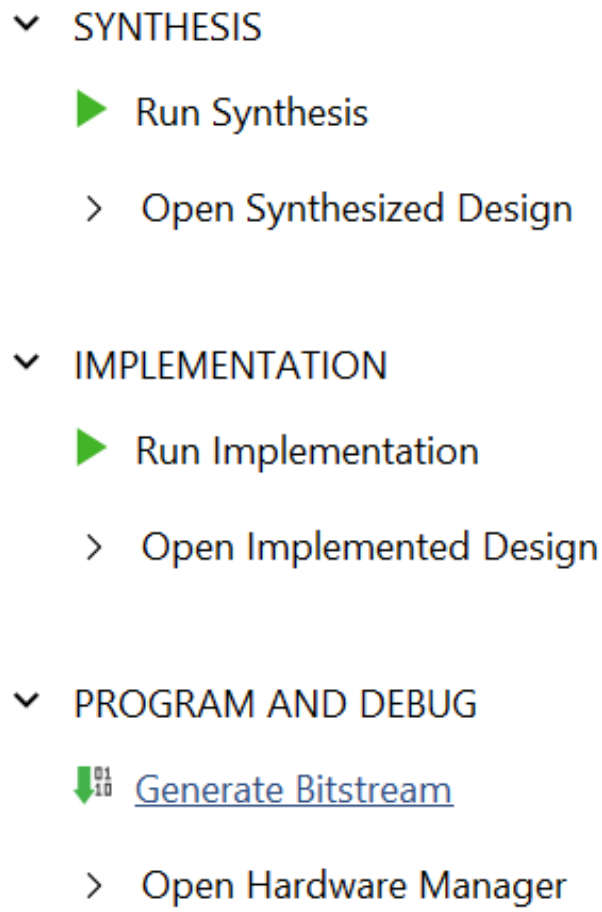
The names you give to the pins in your source file must match the name of the pins in the Constraints File

5. Generate bitstream

1. **Run Synthesis**

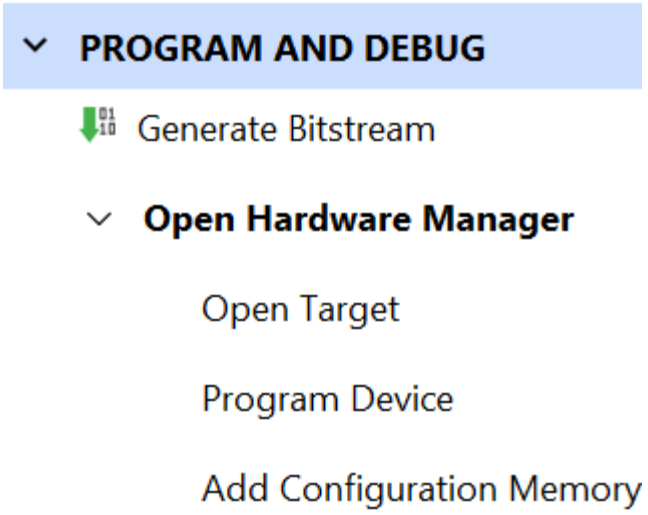
2. **Run Implementation**

3. **Generate Bitstream**



6. Program your FPGA

1. Connect your FPGA to your computer and power in ON.
2. Open the **Hardware Manager**
3. Click on **Open target** then on **Auto Connect**
4. **Program your device** (/!\ Make sure the Basys3 programming mode jumper is on the QSPI position)



VHDL non-synthetizable code

When you write VHDL code, you are writing code that will be translated into gates, registers, RAMs,… by the Synthesis Tool.

There are some parts VHDL that cannot be implement on an FPGA and can only be used for simulation : these parts are called *non-synthesizable*. For example, “the wait for 10 ns” statement is *non-synthesizable* because the FPGA has no direct concept of time.

Trying to synthesize *non-synthesizable* VHDL code will result in the following error:

```
[Synth 8-27] unsynthesizable attribute not supported
["C:/my_src/synth/my_entity_label.vhd":164]
```

Refer yourself to the Golden Reference Guide to check if you can synthesize a piece of code.