

Management and analysis of physics datasets, Part. 1

First Laboratory

Antonio Bergnoli bergnoli@pd.infn.it

13/11/2019



Laboratory Introduction

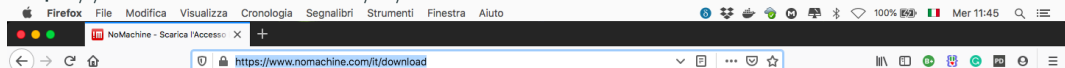


- Become familiar with the Xilinx Vivado IDE.
- Implement the VHDL **Hello World***. Then synthesize, simulate and download the project in the Artix-7 FPGA Development Board.



Installing NoMachine

https://www.nomachine.com/it/download



Scarica e installa il software per il tuo sistema operativo sul computer o sul dispositivo da cui ti connessi e sul computer Windows/Linux/Mac a cui vuoi connetterti. Puoi condividere un desktop remoto o controllare un server con facilità e su qualsiasi rete. Ovunque ti trovi, NoMachine in pochi secondi consente di accedere rapidamente a un desktop remoto in modo sicuro e semplice. Se stai cercando gli altri download della linea di prodotti Enterprise, fai clic [qui](#).



NoMachine per Windows

Windows i386/AMD64 XP/Vista/7/8/8.1/10/Windows Server 2008/2012/2016

 Download



NoMachine per Mac

Mac Intel 64-bit OS X 10.7 o successivo, macOS 10.12/10.13/10.14

 Download



NoMachine per Linux

Linux i386/AMD64, RHEL 4.4 o successivo, SUSE 10 o successivo, Fedora 10 o successivo, Debian 4 o successivo, Ubuntu 8.04 o successivo

 Download



NoMachine per iOS

Compatibile con iOS 8.0 o successivo



Connect the Nomachine Client to the NUC(1)

1. Try to connect to the wifi network named: **studentXX**
2. password: **\$unilab1**



Connect the Nomachine Client to the NUC(2)

Connessioni recenti

Vista Ordina

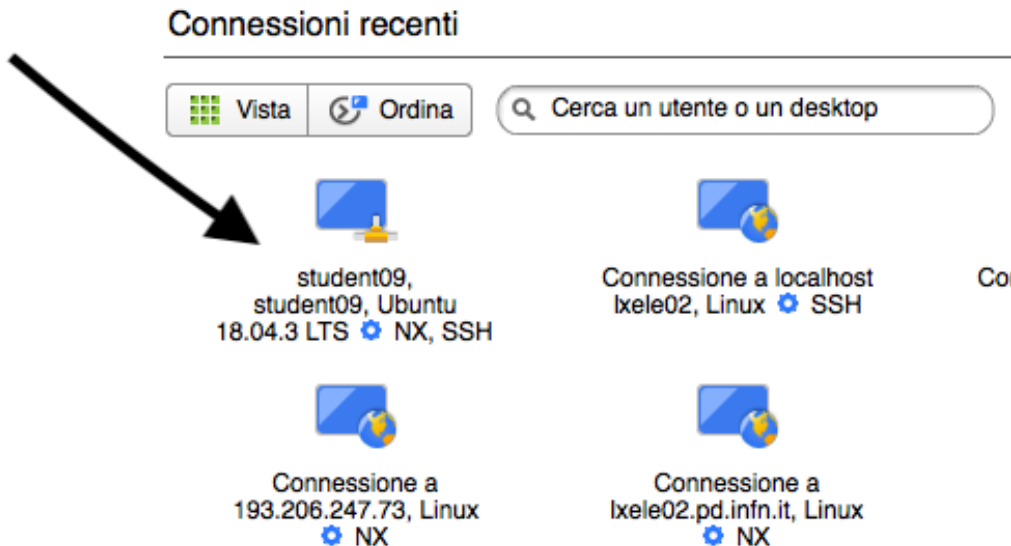
Cerca un utente o un desktop

student09,
student09, Ubuntu
18.04.3 LTS NX, SSH

Connessione a localhost
lxele02, Linux SSH

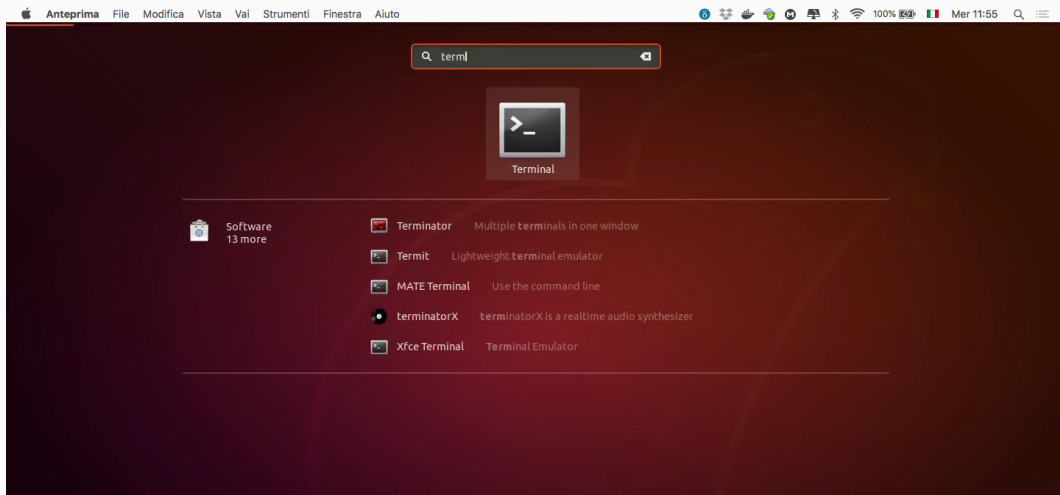
Connessione a
193.206.247.73, Linux
NX

Connessione a
lxele02.pd.infn.it, Linux
NX



Run Vivado

1. open a terminal
 - press *windows key* or *command key* (OSX)
 - type `term`
2. Type: `vivado`

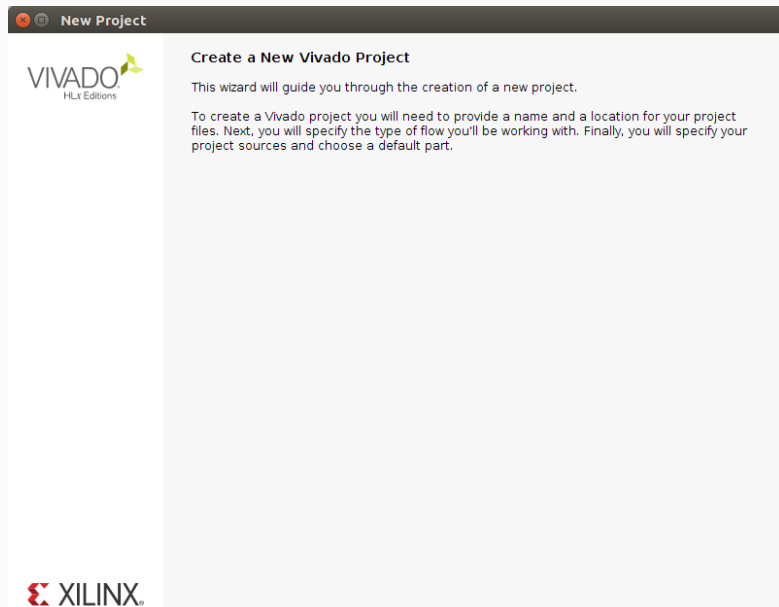


Presentation screen



Make a new project (1)

File → Project → New ...

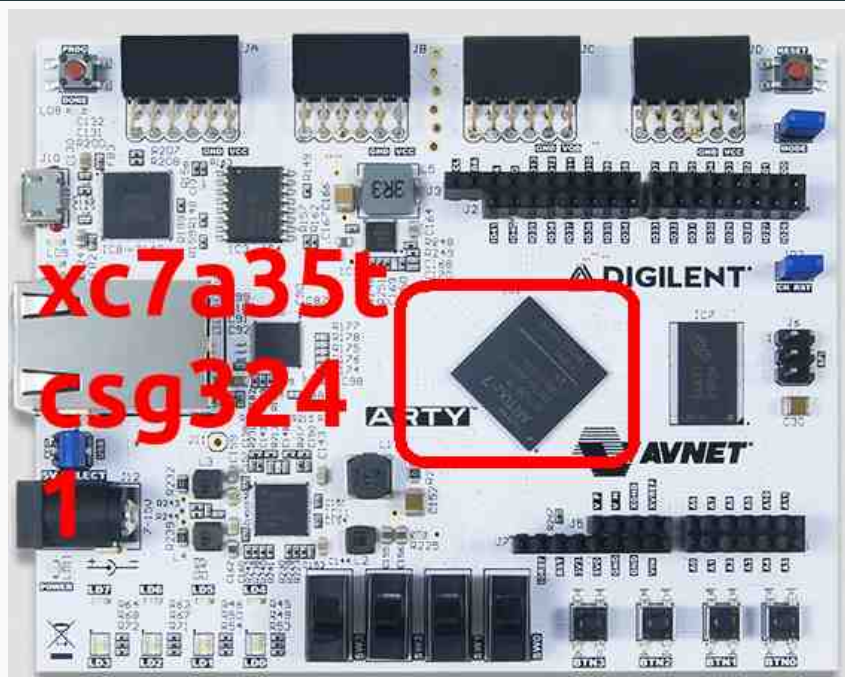


Make a new project (2)

1. Project name : **hello_world**;
2. Check "Create project subdirectory";
3. Next →;
4. Check "RTL Project";
5. Next →;
6. Target language : **VHDL**;
7. Simulator language : **VHDL**;
8. Next →;
9. Next →;
10. Search : **xc7a35tcsg324-1**;
11. Next →;
12. Finish →.



Evaluation Board - FPGA



Make a new project (3)

The screenshot displays the Vivado IDE interface for a project named 'hello_world'. The 'PROJECT MANAGER' window is the central focus, showing a tree view of sources. Under 'Design Sources', there are 'Constraints' and 'Simulation Sources'. 'Simulation Sources' contains a sub-entry 'sim_1'. Below the source tree, the 'Hierarchy' tab is selected, showing a list of libraries and compile order. The 'Properties' tab is also visible, but it is empty, prompting the user to 'Select an object to see properties'.

To the right, the 'Project Summary' window provides details about the project. The 'Settings' tab is active, showing the following information:

- Project name: hello_world
- Project location: /home/stefano/Desktop/hello_world
- Product family: Artix-7
- Project part: xc7a35tcsg324-1
- Top module name: Not defined
- Target language: VHDL
- Simulator language: VHDL

Below the settings, the 'Synthesis' and 'Implementation' tabs are visible. The 'Synthesis' tab shows the status of the synthesis process, and the 'Implementation' tab shows the status of the implementation process.

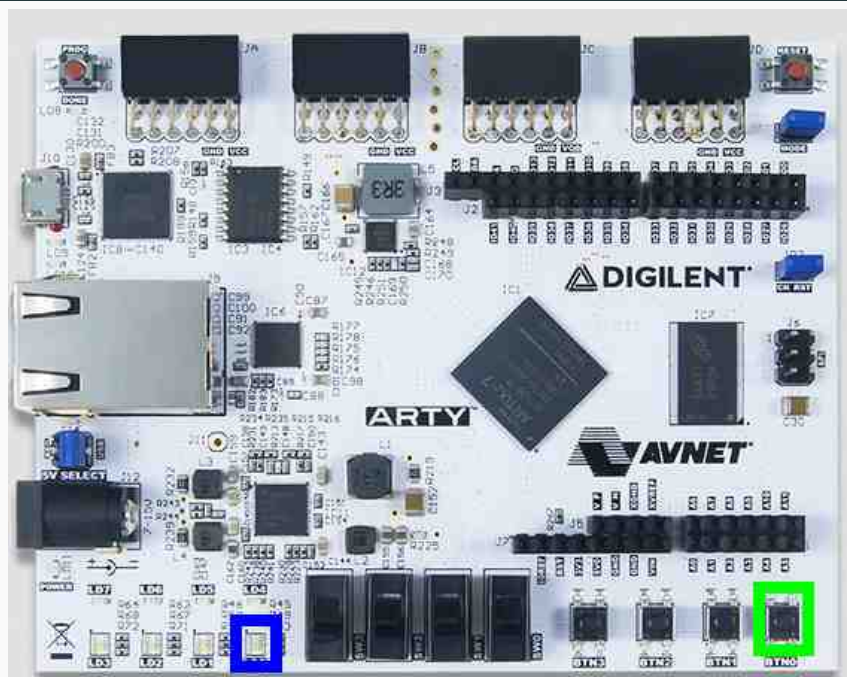
At the bottom of the interface, the 'Design Runs' table is displayed, showing the status of various design runs. The table has columns for Name, Constraints, Status, WNS, TNS, WHS, THS, TPWS, Total Power, Failed Routes, LUT, FF, BRAMs, URAM, DSP, Start, and End.

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	End
synth_1	constrs_1	Not started														
impl_1	constrs_1	Not started														

Hello World



Hello World



INFN

Make the source file (1)

The screenshot displays the Xilinx Vivado IDE interface for a project named "hello_world". The top menu bar includes File, Edit, Flow, Tools, Repgrts, Window, Layout, View, and Help. The "Flow Navigator" on the left shows the project hierarchy: PROJECT MANAGER (Settings, Add Sources, Language Templates, IP Catalog), IP INTEGRATOR (Create Block Design, Open Block Design, Generate Block Design), SIMULATION (Run Simulation), RTL ANALYSIS (Open Elaborated Design), SYNTHESIS (Run Synthesis, Open Synthesized Design), IMPLEMENTATION (Run Implementation, Open Implemented Design), and PROGRAM AND DEBUG.

The "PROJECT MANAGER - hello_world" panel shows the "Sources" list with "top(Behavioral) (top.vhd)" selected. The "Source File Properties" panel for "top.vhd" shows it is "Enabled" and located at "/home/stefano/Desktop/hello_world/hello_world". The "General" tab is active.

The "Project Summary" panel shows the following settings:

- Project name: hello_world
- Project location: /home/stefano/Desktop/hello_world
- Product family: Artix-7
- Project part: xc7a35tcsg324-1
- Top module name: top
- Target language: VHDL
- Simulator language: VHDL

The "Design Runs" panel shows the following table:

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMS	URAM	DSP	Start	Elas
synth_1	constrs_1	Not started														
impl_1	constrs_1	Not started														

Make the source file (2)

Double click on top.vhd.

The screenshot shows the Xilinx IDE interface. The **PROJECT MANAGER - hello_world** window is open, displaying the project hierarchy. Under **Simulation Sources**, **sim_1 (1)** is selected. The **top.vhd** file is highlighted in the **Hierarchy** view. The **top.vhd** source file is open in the editor, showing the following code:

```
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity top is
35 -- Port ( );
36 end top;
37
38 architecture Behavioral of top is
39
40 begin
41
42
43 end Behavioral;
```

The **Design Runs** table at the bottom shows the status of the synthesis and implementation runs:

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	E
synth_1	constrs_1	Not started														
impl_1	constrs_1	Not started														

Make the source file (3)

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity top is
    port (btn_in  : in  std_logic;
          led_out : out std_logic);
end top;
architecture Behavioral of top is
begin
    led_out <= btn_in;
end Behavioral;
```



VHDL naming convention

Signals/components	Name
Clock	<i>clk</i>
Reset	<i>rst</i>
Input Port	<i>port_in</i>
Output Port	<i>port_out</i>
VHDL file name	<i>entityname.vhd</i>
Test bench file name	<i>tb_entityname.vhd</i>
...	...



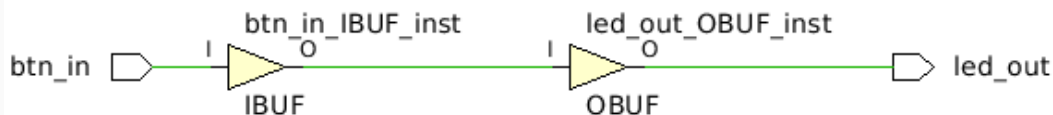
Why we need it?

- To check possible errors in the code.
- To translate the VHDL code in a netlist. A netlist is a list of the logic gates, flip-flops, other components and a list of the connections between them.



Synthesis (2)

1. Run Synthesis →;
2. Cancel →;
3. Open Synthesized Design →;
4. Schematic.



Implementation (1)

- In order to link the VHDL code to the FPGA pin, it is essential write a constraint file (.xdc);
- therefore we need two files: the schematic of the evaluation board and the xdc file of the FPGA mounted on the board;
- schematic
- constraint file

In this first example :

- The input (btn_in) has to be connected to the btn0 button.
- The output (led_out) has to be connected to the led0. (In particular to the blue led. → LED0_B)



Implementation (2)

Find in the *schematic* btn0.

IC1C

BANK 16

IO_L6N_T0_VREF_16
IO_L11P_T1_SRCC_16
IO_L11N_T1_SRCC_16
IO_L12P_T1_MRCC_16
IO_L12N_T1_MRCC_16
IO_L13P_T2_MRCC_16
IO_L13N_T2_MRCC_16
IO_L14P_T2_SRCC_16
IO_L14N_T2_SRCC_16
IO_L19N_T3_VREF_16

D9

BTN0

C9

BTN1

B9

BTN2

B8

BTN3

A8

SW0

C11

SW1

C10

SW2

A10

SW3

A9

UART_TXD_IN

D10

UART_RXD_OUT

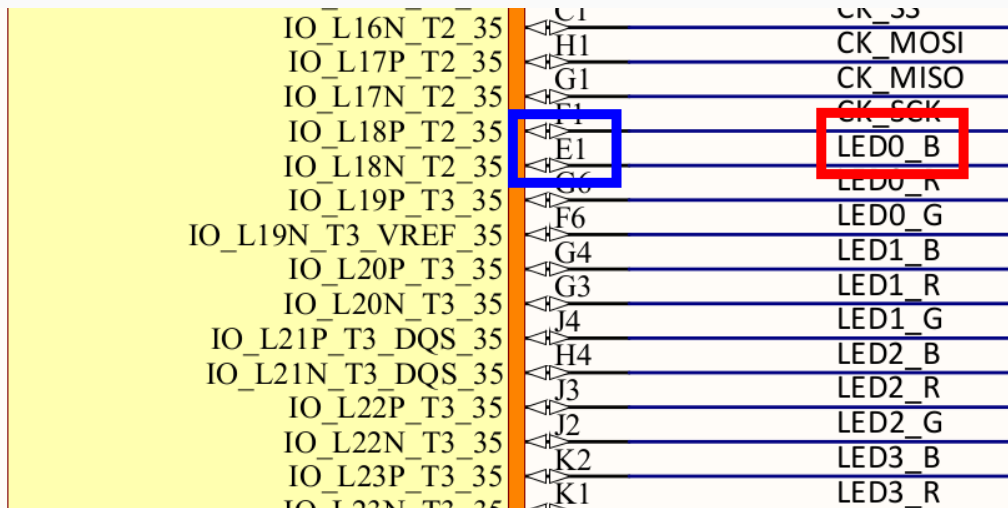
XC7A35T-L1CSG324I

It is connected to the FPGA pin **D9**.



Implementation (3)

Find in the *schematic* led0_b.



It is connected to the FPGA pin **E1**.

Implementation (4)

1. Add sources;
2. Check "Add or create constraints";
3. Next →;
4. Check "Create File";
5. File name: "mapping";
6. Finish →;



Implementation (5) Open the file mapping.xdc.

The screenshot shows the Xilinx IDE interface during the implementation phase. The top menu bar includes File, Edit, Flow, Tools, Repgrts, Window, Layout, View, and Help. The status bar at the top right indicates "Synthesis and Implementation Out-of-date" with a "details" link and a red warning icon. The "Flow Navigator" on the left lists various project tasks, with "SYNTHESIS" currently selected. The main workspace is divided into several panes: "Sources" (showing the project hierarchy), "Source File Properties" (for the selected 'mapping.xdc' file), and "Tcl Console" (showing the command to open the file). The "mapping.xdc" file is open in the editor, showing its path and content.

Sources x Netlist ? _ □ □

Design Sources (1)
• top (Behavioral) (top.vhd)
Constraints (1)
 constrs_1 (1)
 mapping.xdc
Simulation Sources (1)
 sim_1 (1)

Source File Properties ? _ □ □

mapping.xdc ← → ⚙

☒ Enabled

General Properties

Tcl Console x Messages Log Reports Design Runs ? _ □ □

```
close [ open /home/stefano/Desktop/hello_world/hello_world.srcs/sources_1/new/mapping.vhd w ]  
add_files /home/stefano/Desktop/hello_world/hello_world.srcs/sources_1/new/mapping.vhd  
<
```

Type a Tcl command here

1:0 Insert XDC

Implementation (6)

1. Open the file previously downloaded and find **D9** and **E1**;
2. Copy the equivalent lines in the file mapping.xdc;
3. Substitute led_out for led0_b;
4. Substitute btn_in for btn[0];
5. Uncomment the lines (Delete #).

```
set_property PACKAGE_PIN E1 [get_ports { led_out }];  
set_property IOSTANDARD LVCMOS33 [get_ports { led_out }];  
#IO_L18N_T2_35 Sch=led0_b  
set_property PACKAGE_PIN D9 [get_ports { btn_in }];  
set_property IOSTANDARD LVCMOS33 [get_ports { btn_in }];  
#IO_L6N_T0_VREF_16 Sch=btn[0]
```



Implementation (7)

Why we need it?

- To "merge" the netlist and the constraint file, creating a unique design project file.
- To map the components listed in the netlist, in the resources provided by the FPGA.
- To place the resources in the chip and to route them together according to the constraints.

1. Run Implementation →;



Generate Bitstream →;

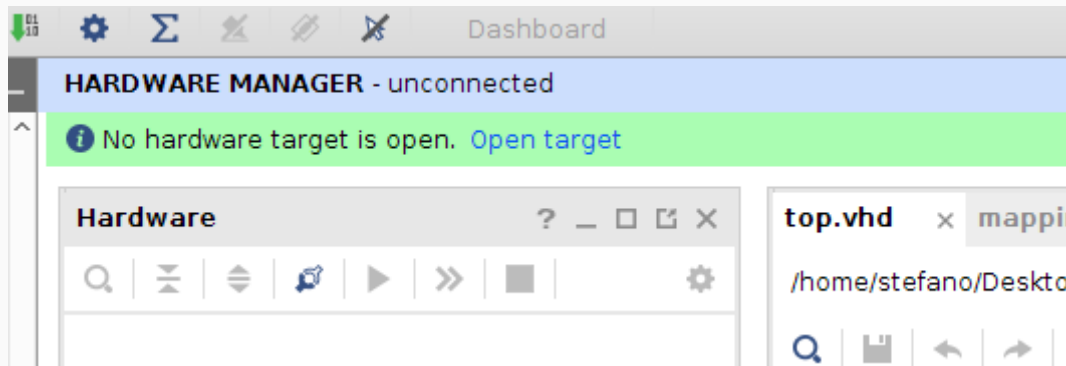
Why we need it?

- To generate the bitstream file (.bit);
- this file represent the final configuration to set the FPGA;
- the file is then downloaded into the FPGA.



Programming (2)

1. Connect the evaluation board to PC by the usb cable.
2. Open Hardware Manager →.
3. Open Target →.
4. Auto Connect.



Programming (3)

hello_world - [/home/stefano/Desktop/hello_world/hello_world.xpr] - Vivado 2018.2

File Edit Flow Tools Reports Window Layout View Help Quick Access

write_bitstream Complete ✓

Default Layout

Flow Navigator

- SYNTHESIS
 - Run Synthesis
 - Open Synthesized Design
- IMPLEMENTATION
 - Run Implementation
 - Open Implemented Design
 - Constraints Wizard
 - Edit Timing Constraints
 - Report Timing Summary
 - Report Clock Networks
 - Report Clock Interaction
 - Report Methodology
 - Report DRC
 - Report Utilization
 - Report Power
 - Schematic
- PROGRAM AND DEBUG
 - Generate Bitstream
 - Open Hardware Manager
 - Open Target
 - Program Device
 - Add Configuration Memory Device

HARDWARE MANAGER - localhost:11640 - tcf/Digilent/210319A43685A

There are no debug cores. Program device Refresh device

Hardware

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210319A...	Open
xc7a35t_0 (1)	Programmed
XADC (System Monitor)	

Properties

Select an object to see properties

top.vhd x mapping.xdc x

```
/home/stefano/Desktop/hello_world/hello_world.srcs/sources_1/new/top.vhd

29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity top is
35     Port (btn_in: in std_logic;
36           led_out : out std_logic );
37 end top;
38
39 architecture Behavioral of top is
40
41 begin
42
43     led_out <= btn_in;
44
```

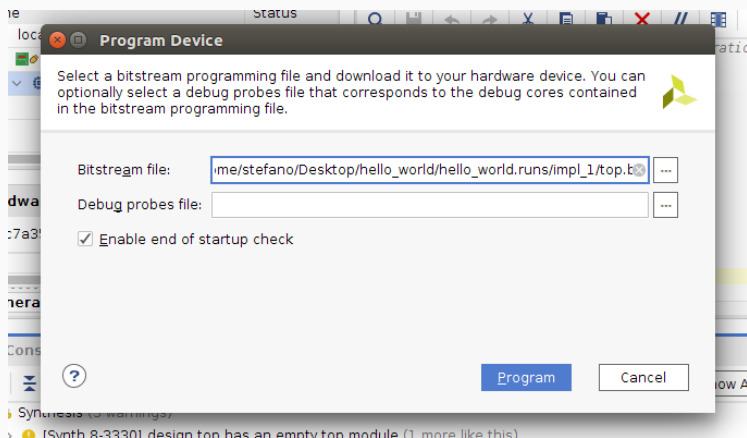
Tcl Console Messages x Serial I/O Links Serial I/O Scans

Warning (11) Info (206) Status (337) Show All

- Synthesis (3 warnings)
 - [Synth 8-3330] design top has an empty top module (1 more like this)
 - [Constraints 18-5210] No constraint will be written out.
- Implementation (3 warnings)
 - Route Design (2 warnings)
 - [Power 33-232] No user defined clocks were found in the design!
Resolution: Please specify clocks using create_clock/create_generated_clock for sequential elements. For pure combinatorial circuits, please specify a virtual clock, otherwise the vectorless estimation might be inaccurate

Programming (4)

Program device →.



(If not checked, check "Enable end of startup check".)

Program device →.



Try to press btn0.



Homework



Suggested exercises

- Redo 1,2,3, ..., N times the exercise "Hello World".
- Get the code more complicated (a little bit). For example instantiate more inputs. Hence repeat each step of "Hello World". An example is reported in the next slide.
- Other complications. More inputs and more outputs.



Hello World with two inputs

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity top is
    port (btn_in  : in  std_logic_vector(1 downto 0);
          led_out : out std_logic);
end top;
architecture Behavioral of top is
begin
    led_out <= btn_in(0) xor btn_in(1);
end Behavioral;
```

