

Management and analysis of physics datasets, Part. 1

First Laboratory

Stefano Pavinato

22/11/2018

1 Laboratory Introduction

2 New Project

3 Hello World

4 Homework

1 Laboratory Introduction

2 New Project

3 Hello World

4 Homework

- 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.

Windows

Double click on the icon created during the installation project.

Linux

```
source <installpath>/Vivado/2018.2/settings64.sh && vivado
```

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>
...	...

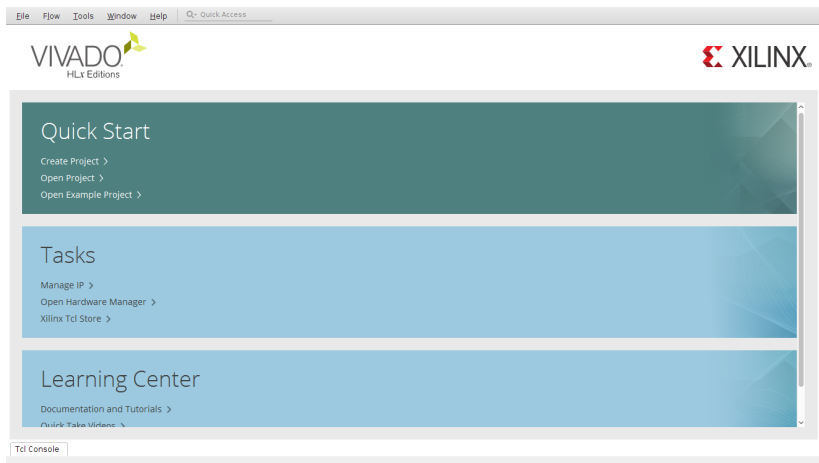
1 Laboratory Introduction

2 New Project

3 Hello World

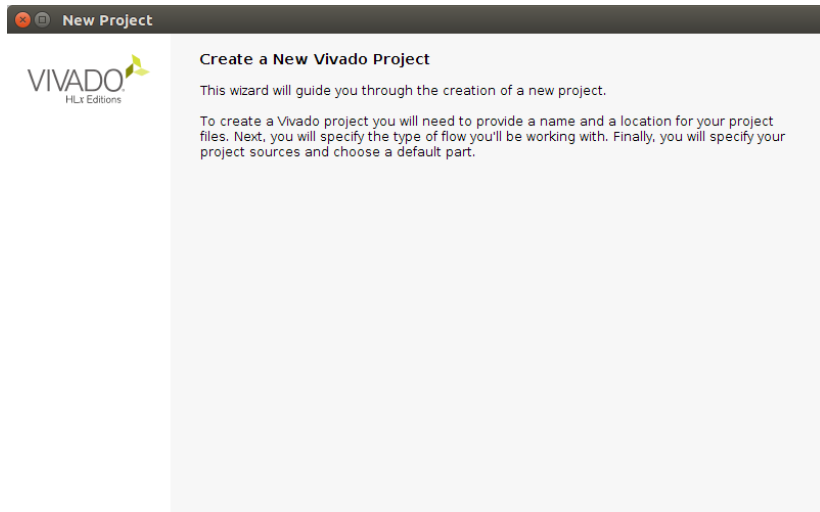
4 Homework

Presentation screen



Make a new project (1)

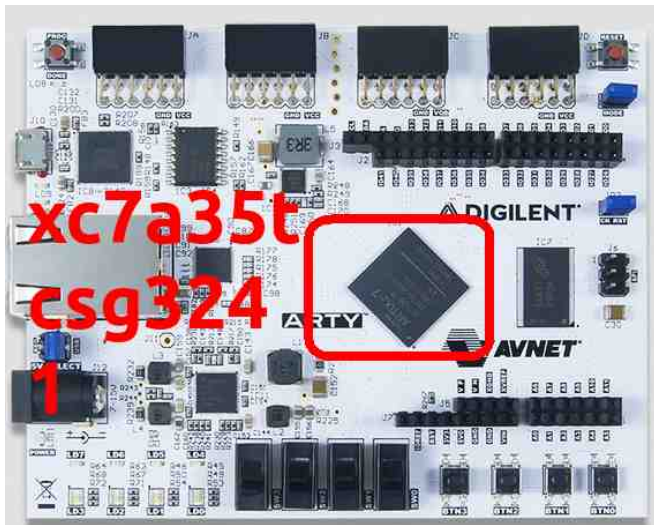
File → Project → New ...



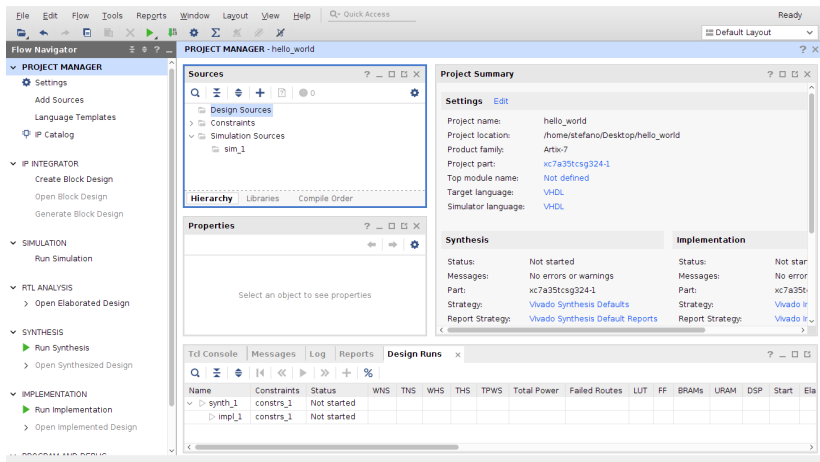
Make a new project (2)

- 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 Xilinx Vivado IDE interface. The 'PROJECT MANAGER - hello_world' window is active, showing the 'Sources' tab with a tree view containing 'Design Sources', 'Constraints', and 'Simulation Sources'. The 'Hierarchy' tab is selected, showing a tree view with 'sim_1'. The 'Properties' tab is also visible, showing a message: 'Select an object to see properties'. The 'Project Summary' window is open, displaying project details:

- 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

The 'Synthesis' and 'Implementation' tabs are also visible. The 'Synthesis' tab shows the status: 'Not started'. The 'Implementation' tab shows the status: 'Not started'. The 'Design Runs' table at the bottom provides a summary of the project's progress:

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														

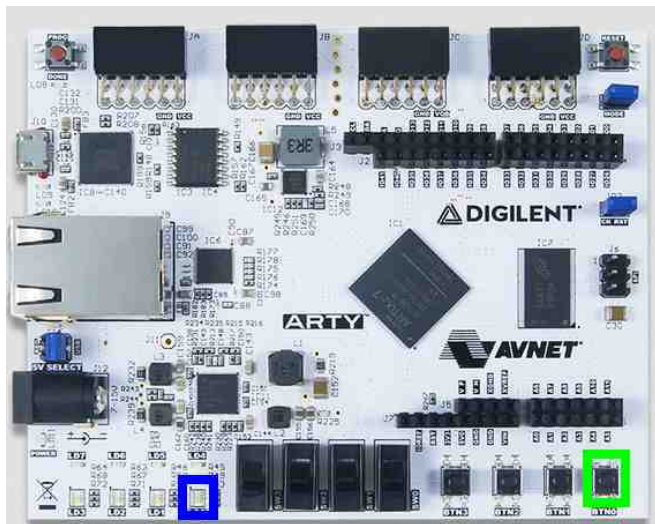
1 Laboratory Introduction

2 New Project

3 Hello World

4 Homework

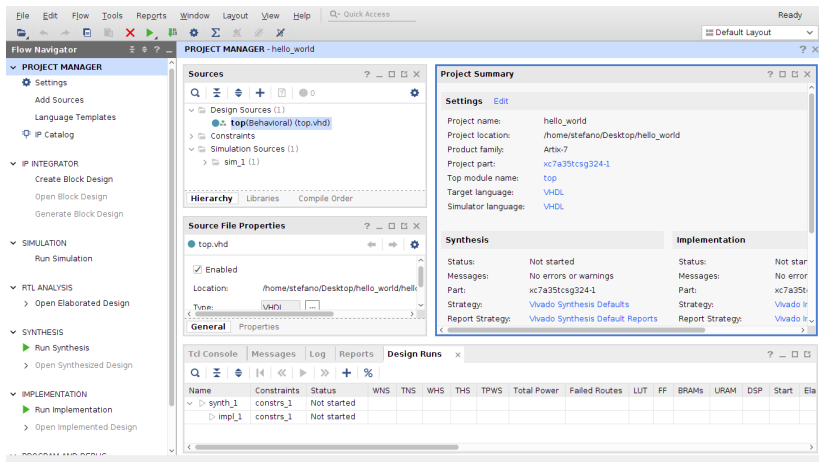
Hello World



Make the source file

- 1 Add sources;
- 2 Check "Add or create design sources";
- 3 Next →;
- 4 Check "Create File";
- 5 File name: "top";
- 6 OK →.
- 7 Finish →;
- 8 OK →.
- 9 Yes →.

Make the source file (1)



The screenshot displays the Vivado IDE interface for a project named "hello_world". The left sidebar shows the "Flow Navigator" with sections for PROJECT MANAGER, IP INTEGRATOR, SIMULATION, RTL ANALYSIS, SYNTHESIS, and IMPLEMENTATION. The main workspace is divided into several panes:

- Project Manager - hello_world**: Shows the project hierarchy with "top (Behavioral) (top.vhd)" selected under "Design Sources (1)".
- Source File Properties**: Shows properties for "top.vhd", including "Location: /home/stefano/Desktop/hello_world/hello_world" and "Type: VHDL".
- Project Summary**: Displays project details such as "Project name: hello_world", "Project location: /home/stefano/Desktop/hello_world", "Product family: Artix-7", "Project part: xc7a35tcsq324-1", "Top module name: top", "Target language: VHDL", and "Simulator language: VHDL".
- Synthesis and Implementation**: Shows the status of the synthesis and implementation processes. The "Synthesis" tab is active, showing "Status: Not started", "Messages: No errors or warnings", "Part: xc7a35tcsq324-1", "Strategy: Vivado Synthesis Defaults", and "Report Strategy: Vivado Synthesis Default Reports".
- Design Runs**: A table showing the results of the synthesis and implementation runs.

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)

Duoble click on top.vhd.

The screenshot displays the Quartus II software interface. The top menu bar includes File, Edit, Flow, Tools, Reports, Window, Layout, View, and Help. Below the menu bar is a toolbar with various icons. The main window is divided into several panes:

- Flow Navigator:** Shows the project hierarchy with 'PROJECT MANAGER' selected. It includes sections for Settings, Add Sources, Language Templates, IP Catalog, IP INTEGRATOR, SIMULATION, RTL ANALYSIS, SYNTHESIS, and IMPLEMENTATION.
- PROJECT MANAGER - hello_world:** Displays the project structure. The 'Design Sources' section shows 'top.vhd' selected. The 'Hierarchy' pane shows the project structure.
- Project Summary:** Shows the path to the source file: `/home/stefano/Desktop/hello_world/hello_world.srcs/sources_1/new/top.vhd`.
- Source File Editor:** Displays the content of `top.vhd`. The code includes library declarations for IEEE and UNISIM, followed by an entity declaration and architecture. The line `end Behavioral;` is highlighted in yellow.
- Design Runs:** A table showing the status of the design runs. The table has columns: Name, Constraints, Status, WNS, TNS, WHS, THS, TPWS, Total Power, Failed Routes, LUT, FF, BRAMS, URAM, DSP, Start, and E.

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														

The status bar at the bottom shows the time 43:00, the mode 'Insert', and the language 'VHDL'.

Make the source file (3)

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

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.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;

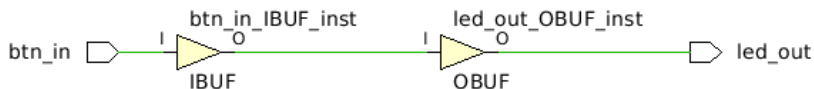
end Behavioral;
```

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.



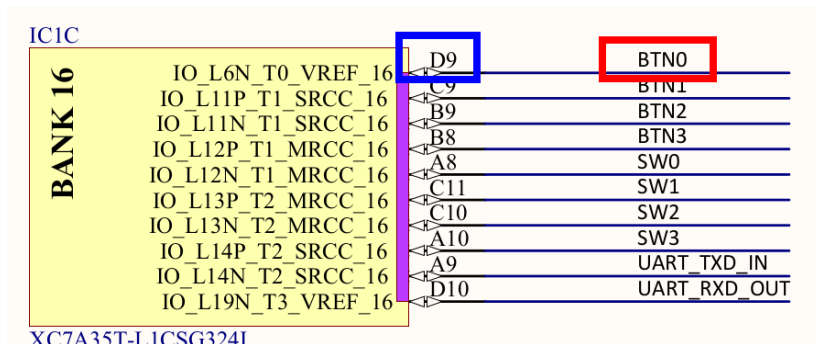
- In order to link the VHDL code to the FPGA pin, it is essential write a constraint file (.xdc);
- therefore we need two files: the schematic of the evaluation board and the xdc file of the FPGA mounted on the board;
- **SCHEMATIC.**
- **XDC;**

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.



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

Implementation (3)

Find in the schematic led0_b.

IO_L16N_T2_35	C1	CK_SS
IO_L17P_T2_35	H1	CK_MOSI
IO_L17N_T2_35	G1	CK_MISO
IO_L18P_T2_35	F1	CK_SCK
IO_L18N_T2_35	E1	LED0_B
IO_L19P_T3_35	G6	LED0_R
IO_L19N_T3_VREF_35	F6	LED0_G
IO_L20P_T3_35	G4	LED1_B
IO_L20N_T3_35	G3	LED1_R
IO_L21P_T3_DQS_35	J4	LED1_G
IO_L21N_T3_DQS_35	H4	LED2_B
IO_L22P_T3_35	J3	LED2_R
IO_L22N_T3_35	J2	LED2_G
IO_L23P_T3_35	K2	LED3_B
IO_L23N_T3_35	K1	LED3_R

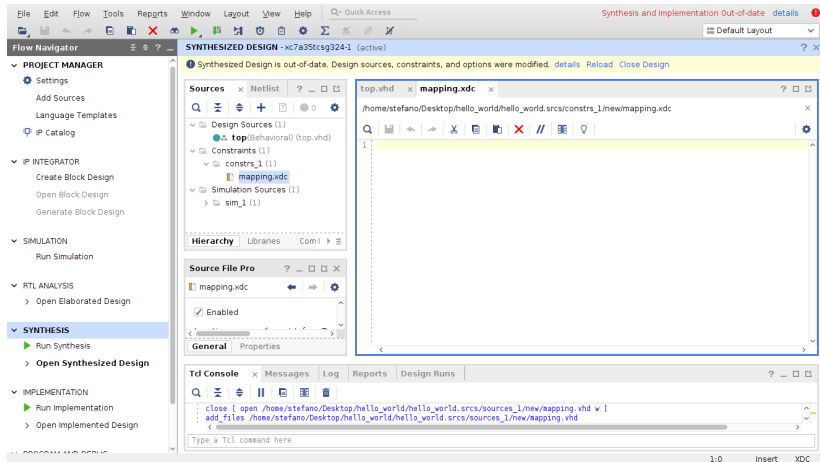
It is connected to the FPGA pin **E1**.

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.



Implementation (6)

- 1 Open the file **XDC** 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 -dict { PACKAGE_PIN E1      IOSTANDARD LVCMOS33 } [get_ports { led_out }];  
set_property -dict { PACKAGE_PIN D9      IOSTANDARD LVCMOS33 } [get_ports { btn_in }];
```

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 →;

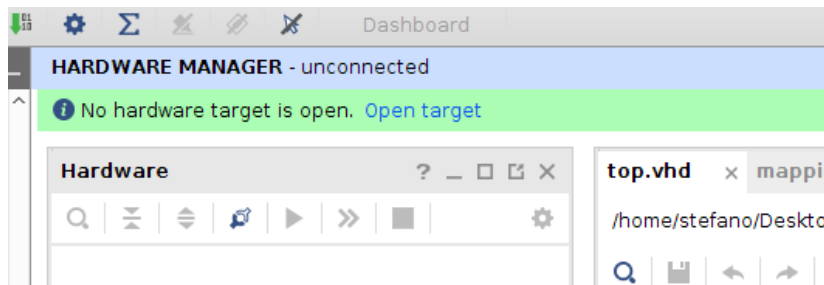
Generate Bitstream →;

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)

The screenshot displays the Vivado 2018.2 IDE interface for a project named 'hello_world'. The top toolbar shows the 'Program' button (a green play icon) highlighted with a red box. Below the toolbar, the 'Hardware Manager' tab is active, showing a table of hardware components:

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

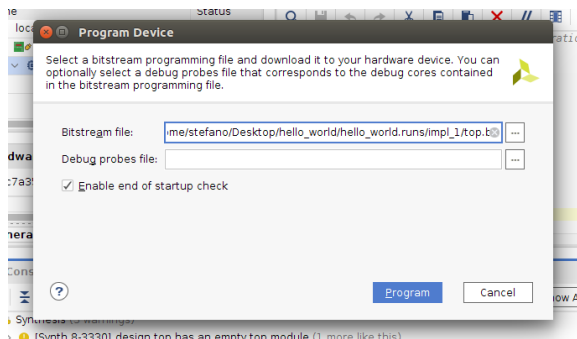
The 'Properties' tab is also visible, showing a message: 'Select an object to see properties'. The 'Flow Navigator' on the left shows the 'Open Implemented Design' step selected under the 'IMPLEMENTATION' section. The 'Messages' console at the bottom displays several warnings:

- [Synth 8-3330] design top has an empty top module (1 more like this)
- [Constraints 18-5210] No constraint will be written out.
- [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

The 'top.vhd' file is open in the editor, showing the following code:

```
//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: end;
```

Program device →.



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

Program device →.

Try to press btn0.

1 Laboratory Introduction

2 New Project

3 Hello World

4 Homework

- 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;
```

```
set_property -dict { PACKAGE_PIN E1      IOSTANDARD LVCMOS33 } [get_ports { led_out }]; #IO_L18N_T2_35 Sch=led0_b
set_property -dict { PACKAGE_PIN D9      IOSTANDARD LVCMOS33 } [get_ports { btn_in[0] }]; #IO_L6W_T0_VREF_16 Sch=btn[0]
set_property -dict { PACKAGE_PIN A8      IOSTANDARD LVCMOS33 } [get_ports { btn_in[1] }]; #IO_L12N_T1_MRCC_16 Sch=sw[0]
```