FPGA Design

Part II - Xilinx Design Tools

Thomas Lenzi

Vrije Universiteit Brussel

UNIVERSITÉ LIBRE DE BRUXELLES ULB

Objective

- In this lecture we will learn to design for FPGAs using the Xilinx Development Tools.

Development Tools

Xilinx Development Tools

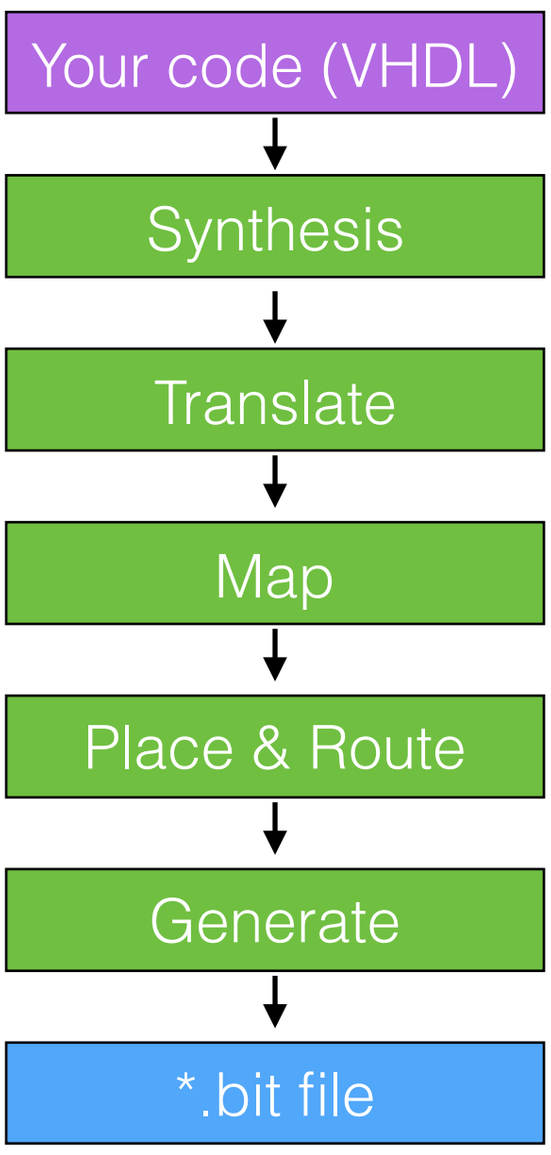
- ISE Design Suite: for Spartan3, Spartan6, and Virtex6 FPGAs
- Vivado Design Suite: for new generations of FPGAs
- Impact: programming of the FPGA
- ISim: simulate VHDL and Verilog code
- ChipScope: in-system debug and analysis. Allows you to monitor and control signals inside the FPGA
- Core Generator: gives you access to pre-build components
- PlanAhead: floorplanning, pin assignment, …
- Xilinx Software Development: develop C/C++ code for embedded systems
- etc

Design workflow

1. ISE: write the VHDL and Verily code
2. *ISim: perform software simulations
3. *PlanAhead: allocate ressources on the FPGA
4. Impact: program the FPGA
5. *ChipScope: in-system debug of the code

ISE Design Suite

Workflow



Transforms your VHDL code in hardware components

Include the user constraints in the synthesised code

Map the translated code to the device ressources

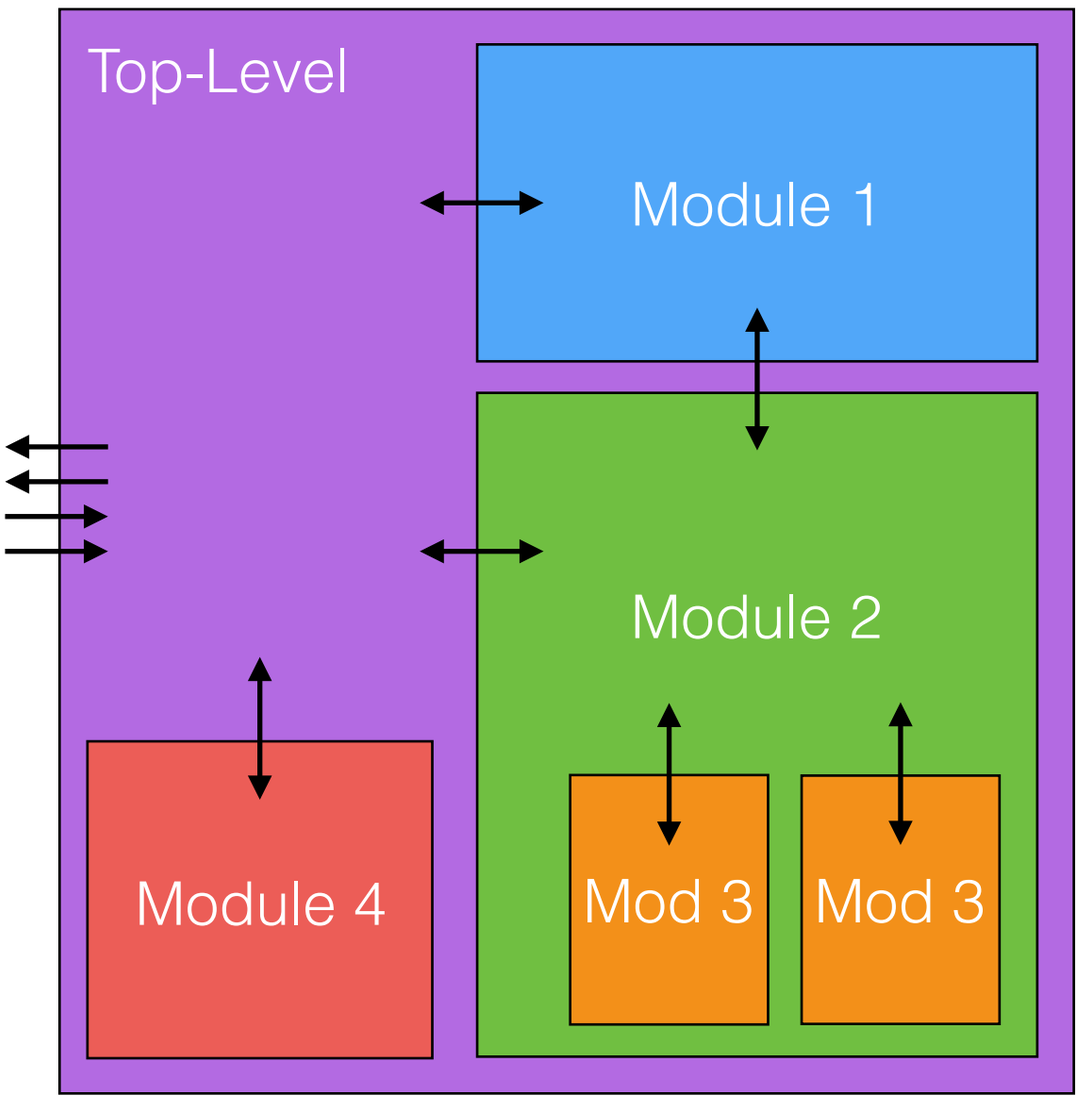
Place the design in the FPGA and form the routes

Generate the BIT file that can be loaded on the FPGA

Structure of the project

- Every design includes two things:
  - VHDL files that describe the behaviour of the code;
  - UCF (user constraints) files that constrain the design (IO pins for signals, timing requirements, …).

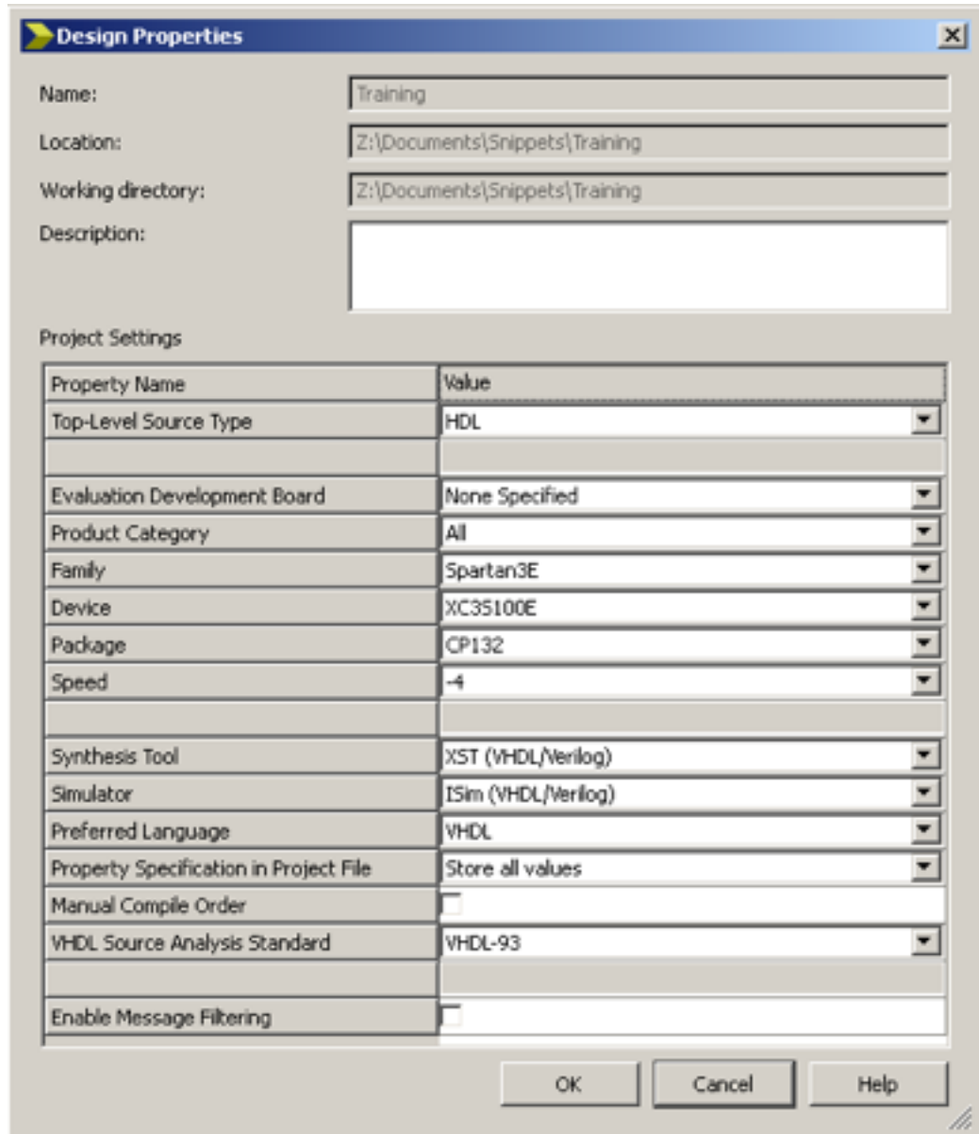
Structure of the code



- The structure of a VHDL code can be compared to the structure of an FPGA.
- The Top-Level file describes the FPGA itself. It regroups the signals that will leave or enter. That module is connected to the world via the IO pins.
- In that module, you can create sub-modules to use regions of the FPGA.
- There is no limitation to the number of sub-modules you can create.

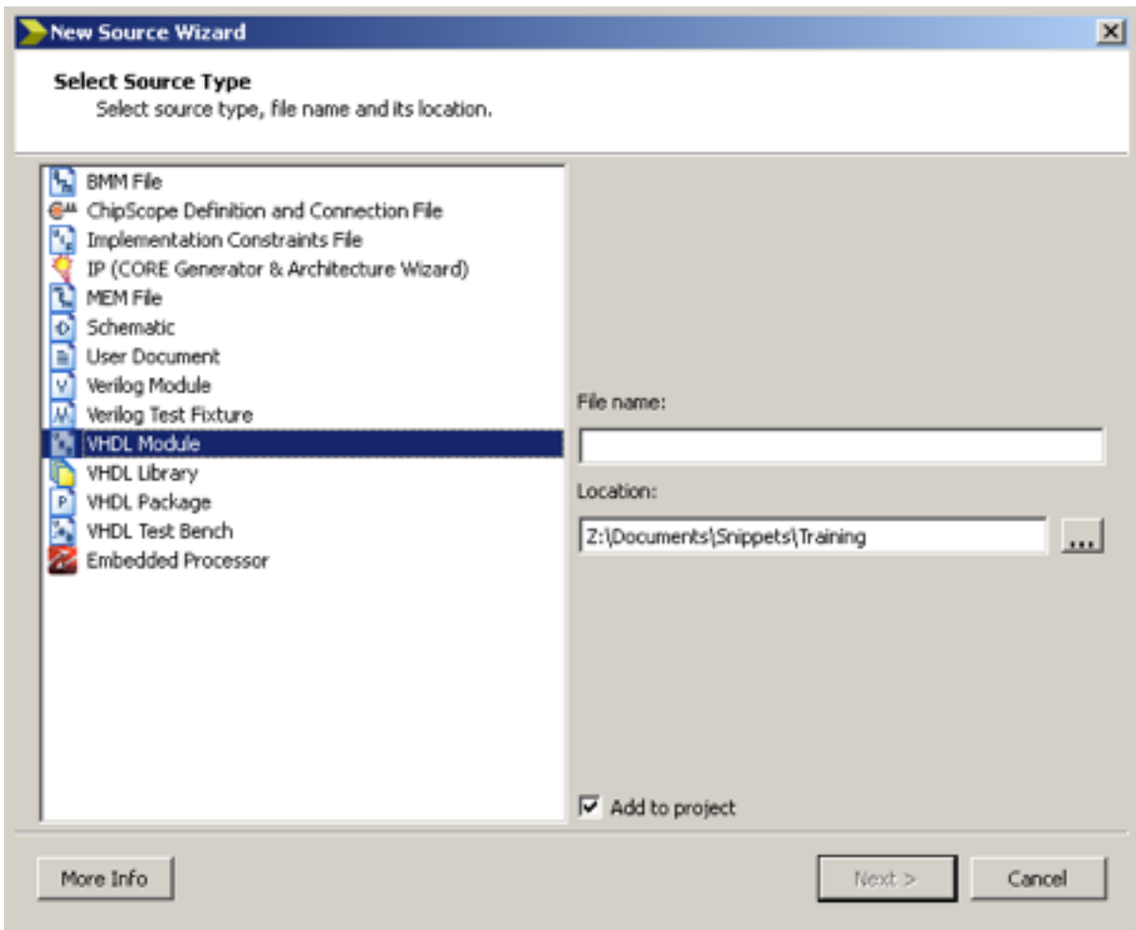
Create a project

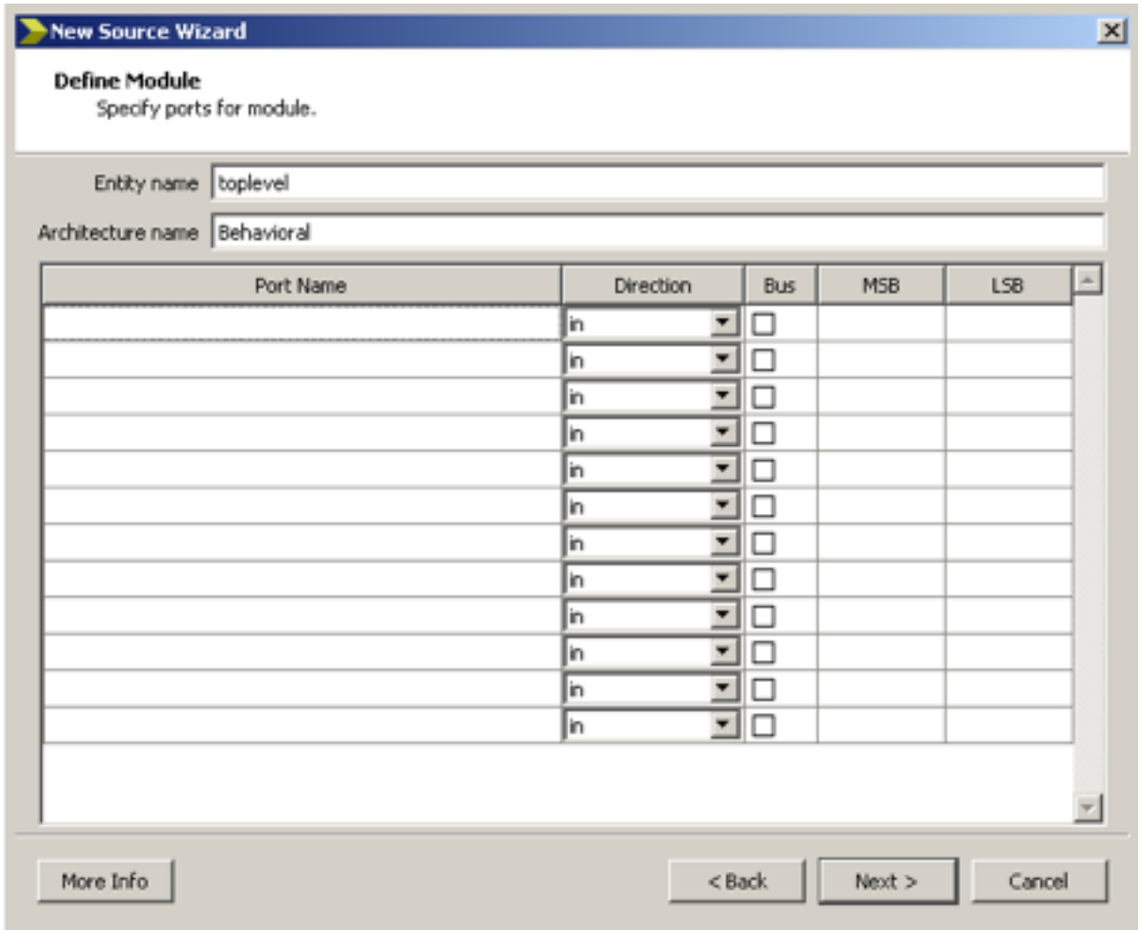
- Go to “File > New Project…”
- Give it a name and a location, set the “Top-Level source type” to “HDL”, and click “Next”
- Select the attributes corresponding to your board and click “Next”
  - Family: Spartan3E
  - Device: XC3S100E
  - Package: CP132
  - Speed: -4
  - Preferred language: VHDL
- Finally click “Finish”



Create a VHDL file

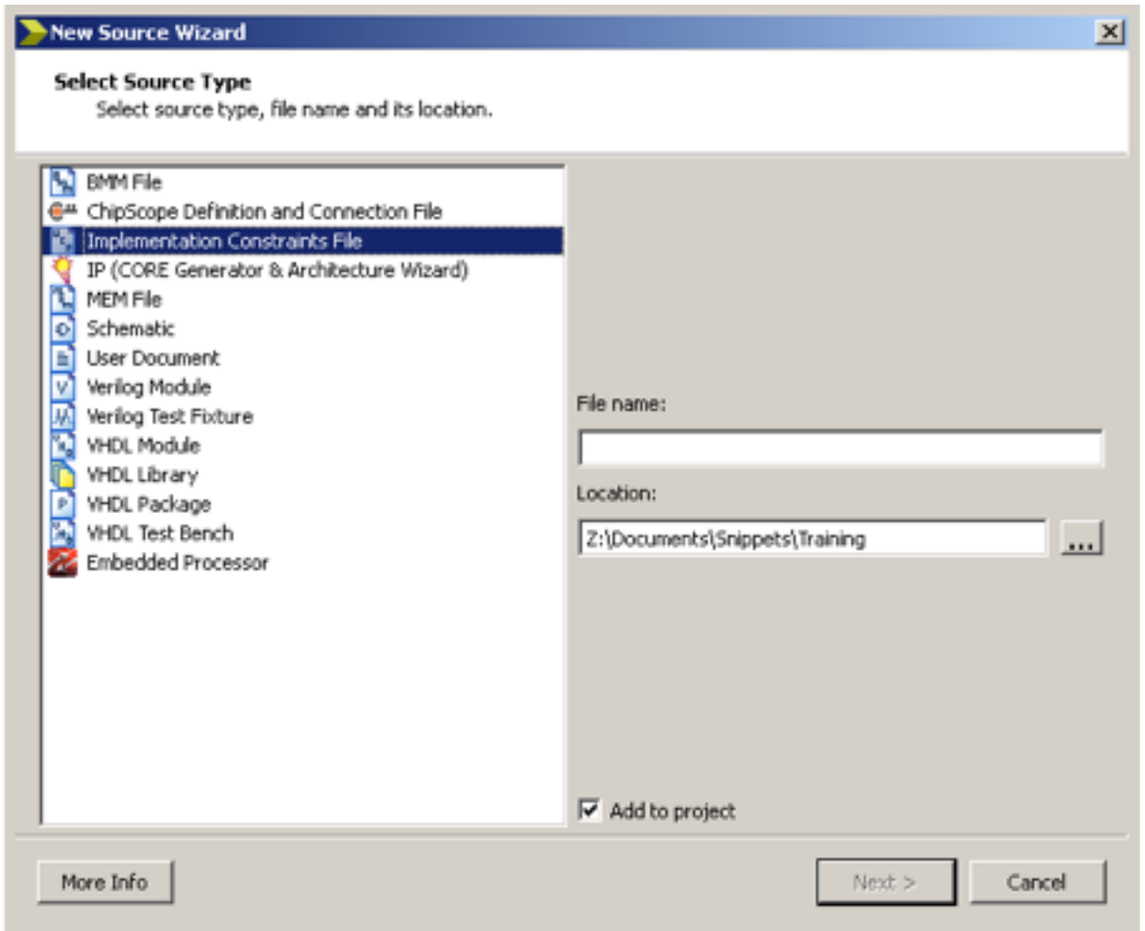
- Go to “Project > New Source…”
- Select “VHDL Module” and give it a name with a .vhd extension, then click “Next”
- On the next window, leave everything empty and click “Next” and then “Finish”



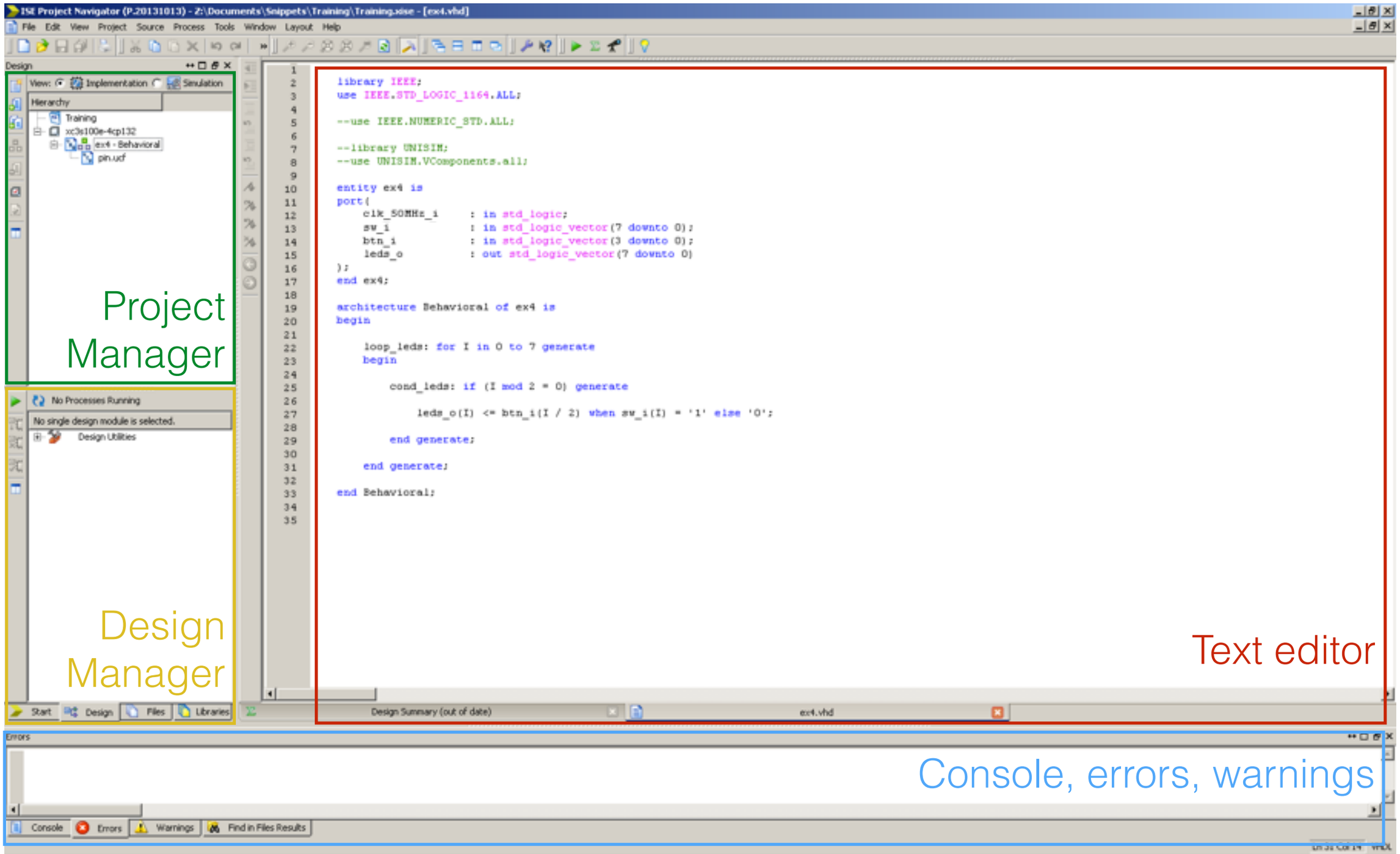


Create a UCF file

- Go to “Project > New Source…”
- Select “Implementation Constraints File” and give it a name with a .ucf extension, then click “Next”
- On the next window, click “Finish”



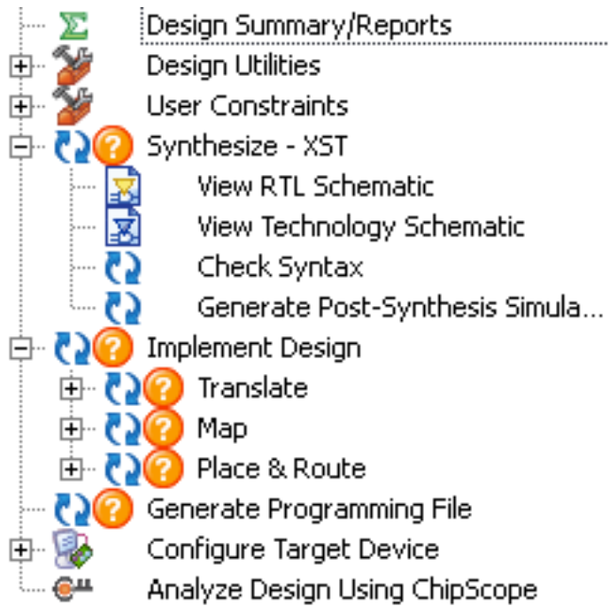
ISE interface




```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  --use IEEE.NUMERIC_STD.ALL;
5
6  --library UNISIM;
7  --use UNISIM.VComponents.all;
8
9  entity ex4 is
10 port(
11     clk_50MHz_i     : in std_logic;
12     sw_i            : in std_logic_vector(7 downto 0);
13     btn_i           : in std_logic_vector(3 downto 0);
14     leds_o          : out std_logic_vector(7 downto 0)
15 );
16 end ex4;
17
18 architecture Behavioral of ex4 is
19 begin
20
21     loop_leds: for I in 0 to 7 generate
22     begin
23
24         cond_leds: if (I mod 2 = 0) generate
25
26             leds_o(I) <= btn_i(I / 2) when sw_i(I) = '1' else '0';
27
28         end generate;
29
30     end generate;
31
32 end Behavioral;
33
34
35
```

Design management

- This region allows you to generate your design and flash it on the FPGA.



- Various tools are also present to analyse and optimise your design.

- To compile a design, simply double-click on “Generate Programming File”. All the necessary steps will be done automatically.

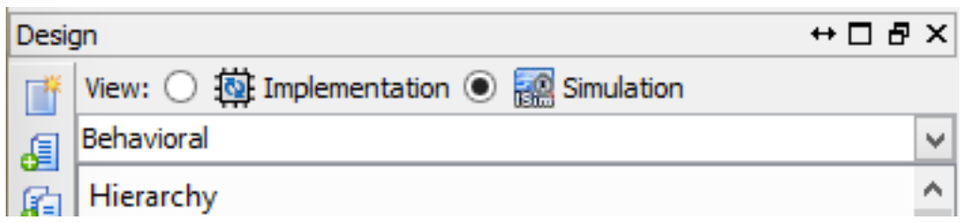
ISim

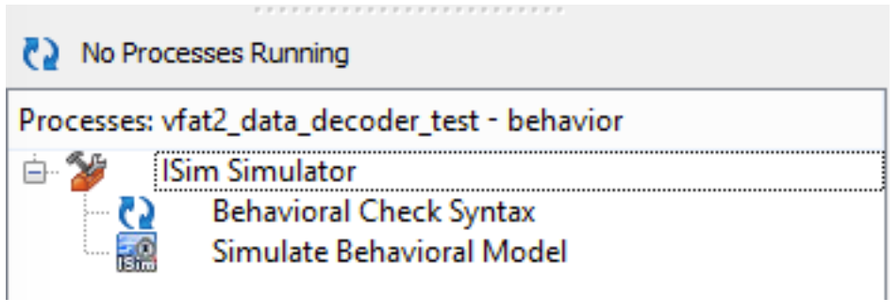
Simulation environment

- ISE makes a difference between Implementation (code that will run on the FPGA) and Simulation (code that will be run on the software side only).

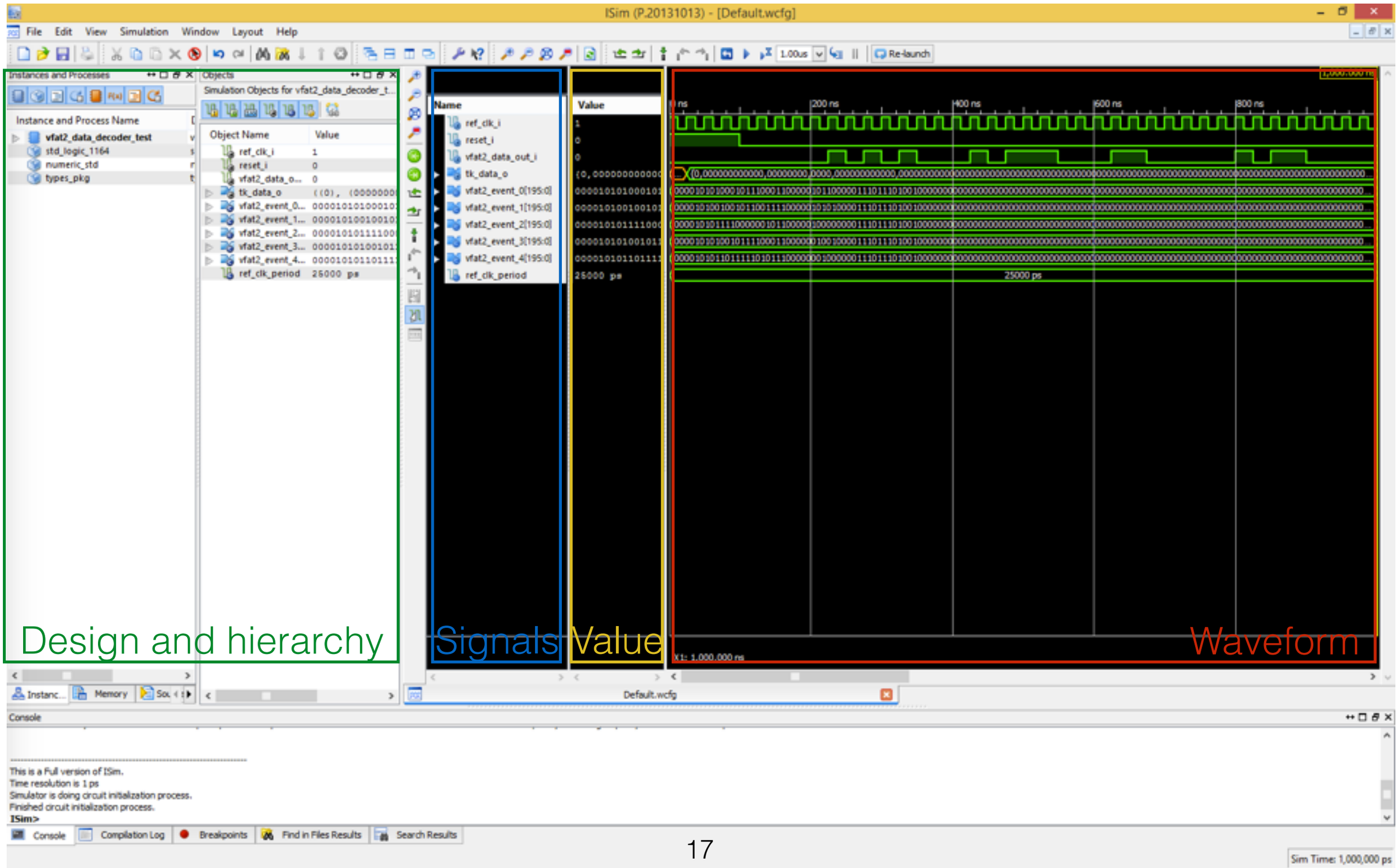
- Simulation often re-uses implementation code but provides it with signals that are generated in the simulation code.

- For example, instead of being connected to a button on the PCB, the signal would be manually defined (first high, then low after x ns, …) by a VHDL simulation code.



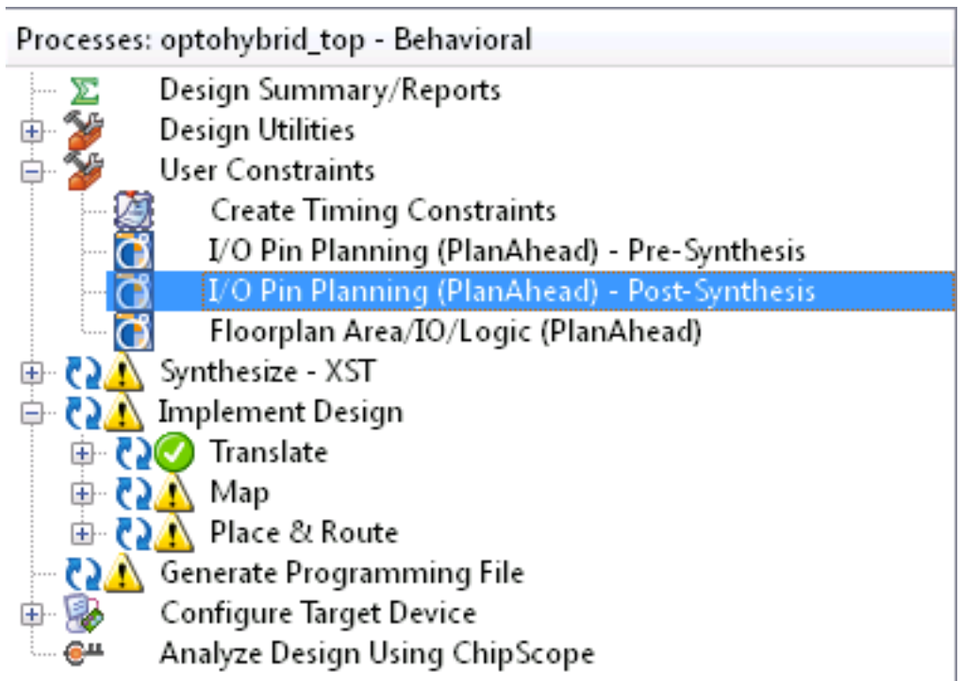


ISim


ISim (P.20131013) - [Default.wcfg]
File Edit View Simulation Window Layout Help
Re-launch
Instances and Processes
Instance and Process Name
vfat2_data_decoder_test
std_logic_1164
numeric_std
types_pkg
Objects
Simulation Objects for vfat2_data_decoder_t...
Object Name
Value
ref_clk_i 1
reset_i 0
vfat2_data_o... 0
tk_data_o
vfat2_event_0...
vfat2_event_1...
vfat2_event_2...
vfat2_event_3...
vfat2_event_4...
ref_clk_period 25000 ps
Name
Value
ref_clk_i
reset_i
vfat2_data_out_i
tk_data_o
vfat2_event_0[195:0]
vfat2_event_1[195:0]
vfat2_event_2[195:0]
vfat2_event_3[195:0]
vfat2_event_4[195:0]
ref_clk_period
25000 ps
200 ns
400 ns
600 ns
800 ns
25000 ps
X1: 1,000,000 ns
Design and hierarchy
Signals
Value
Waveform
Default.wcfg
Console
This is a Full version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
Console
Compilation Log
Breakpoints
Find in Files Results
Search Results
Sim Time: 1,000,000 ps

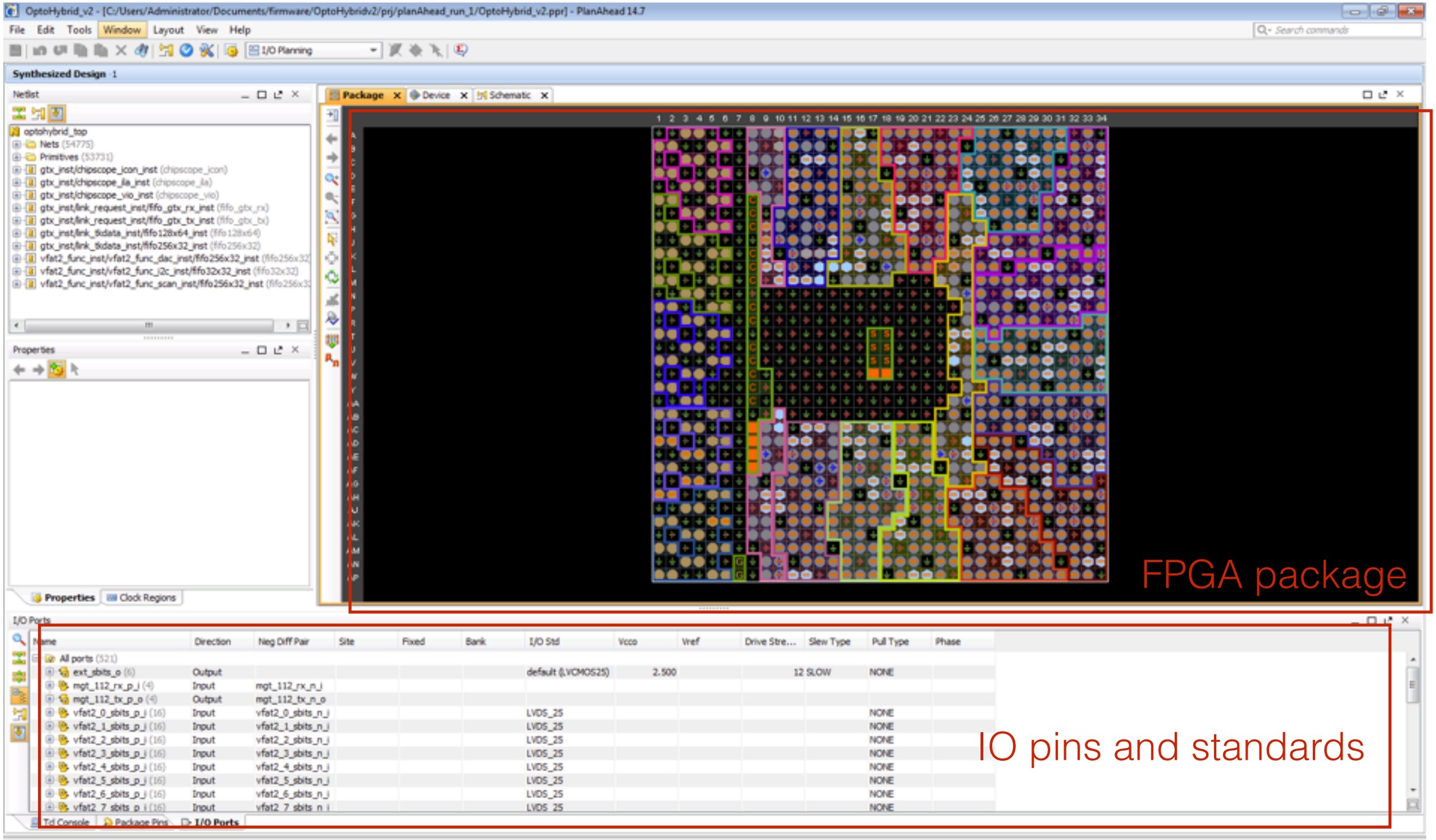
PlanAhead

Pin placement



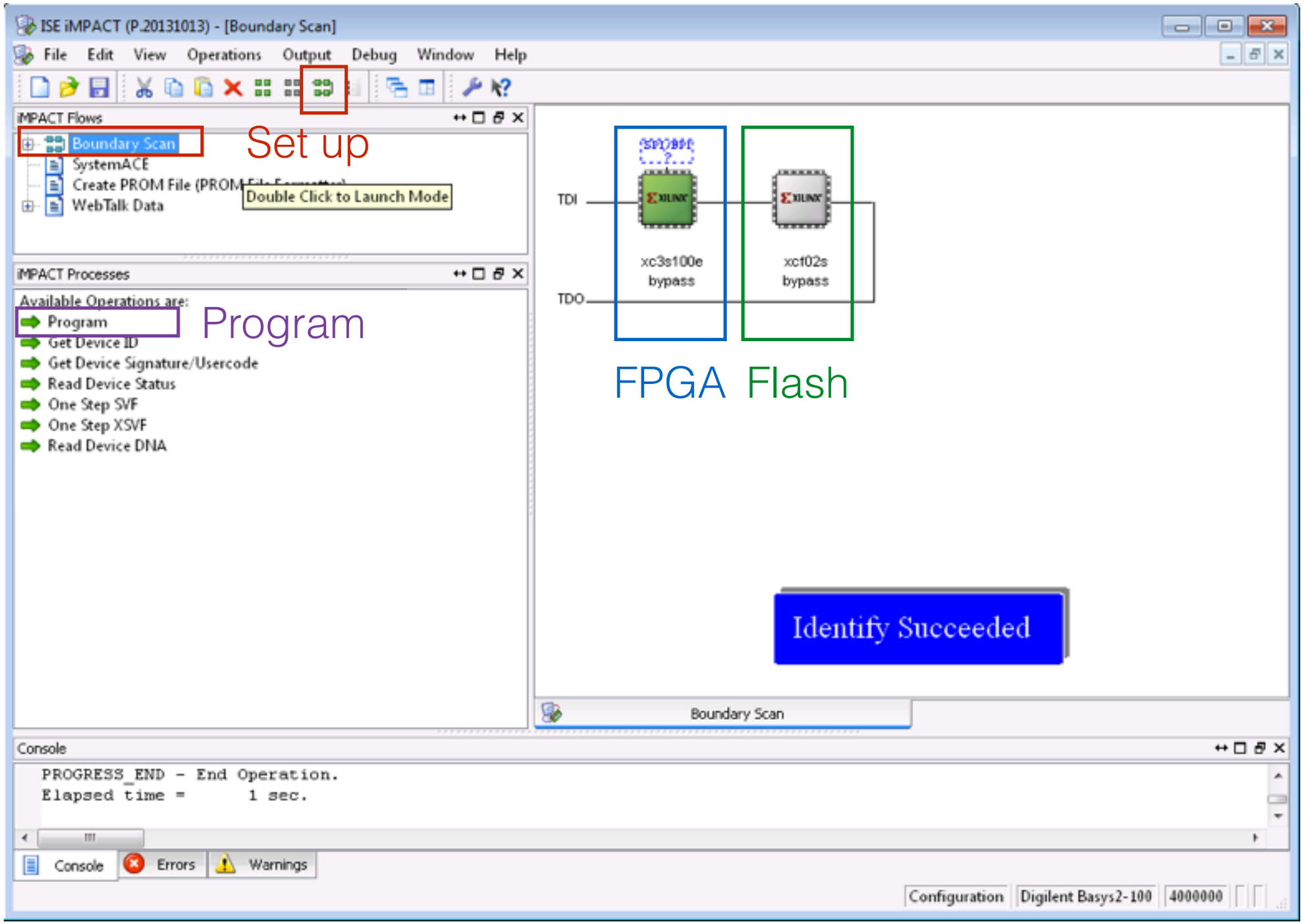
- PlanAhead can be used for pin placement in the design. It gives a visual representation of the FPGA and allows you to point and click to place signals.

PlanAhead interface



Impact

JTAG chain


ISE iMPACT (P.20131013) - [Boundary Scan]
File Edit View Operations Output Debug Window Help
iMPACT Flows
Boundary Scan
SystemACE
Create PROM File (PROM
WebTalk Data
Double Click to Launch Mode
Set up
iMPACT Processes
Available Operations are:
Program
Get Device ID
Get Device Signature/Usercode
Read Device Status
One Step SVF
One Step XSVF
Read Device DNA
Program
TDI
TDO
xc3s100e
bypass
xcf02s
bypass
FPGA
Flash
Identify Succeeded
Boundary Scan
Console
PROGRESS_END - End Operation.
Elapsed time = 1 sec.
Console
Errors
Warnings
Configuration
Digilent Basys2-100
4000000