Xilinx Vivado tutorial (Version 2018.3)
Check **known issues.pdf** for any known issues.

CREATING A PROJECT

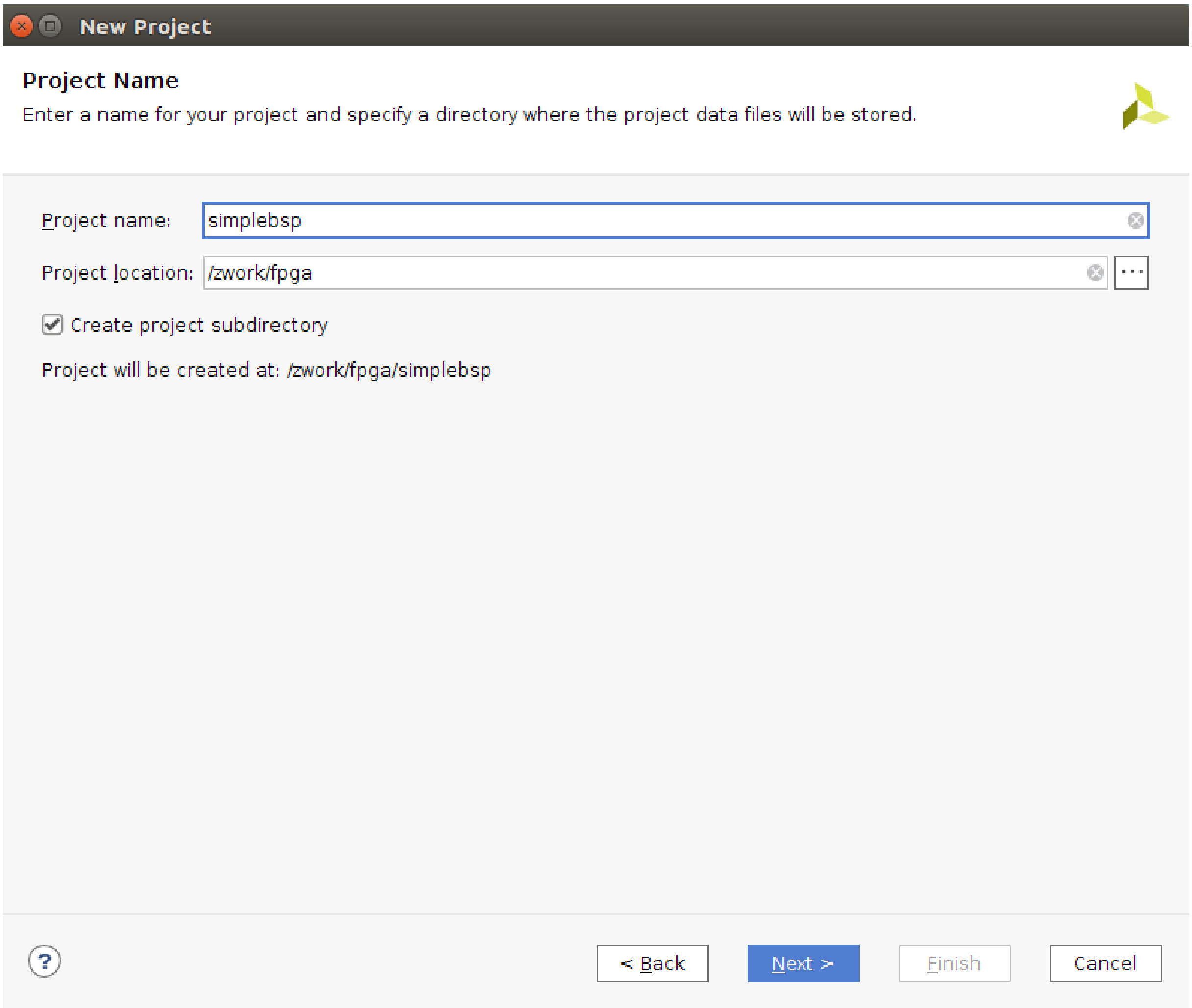
Copy and unzip the given source files to **/zwork/dd/simplebsp_src**

Open a Linux terminal and type:
./m1data.sh //if promted the password is ubupw

source Modelsim106c_settings64.sh
source XilinxVivado201803_settings64.sh

vivado&

File-> Project-> New…
Press Next



New Project

Project Name

Enter a name for your project and specify a directory where the project data files will be stored.

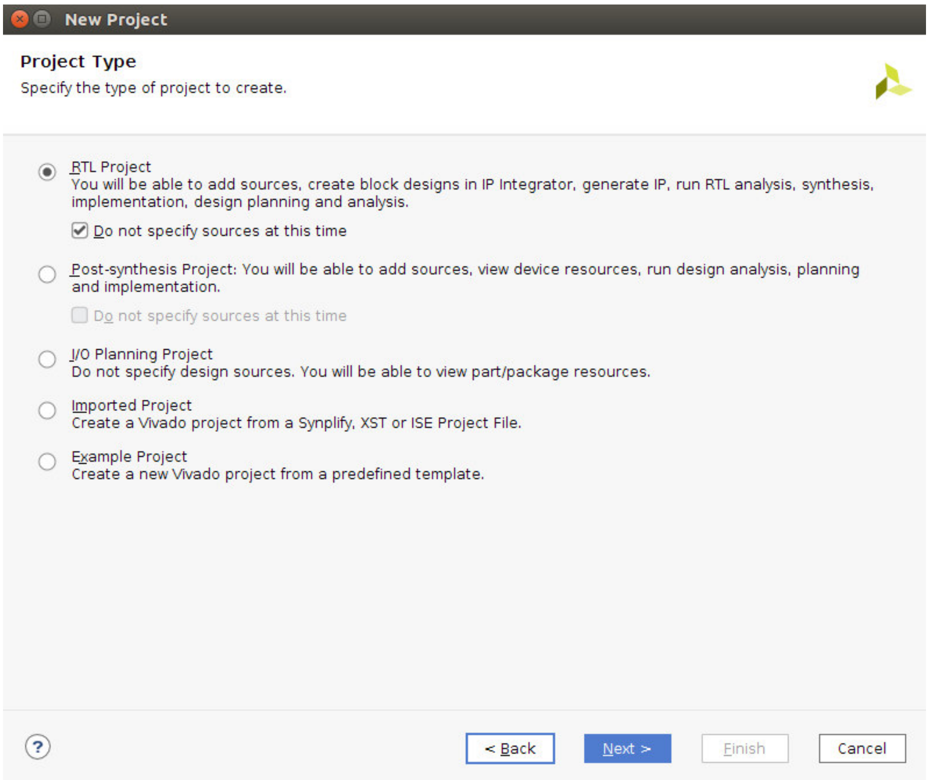
Project name: simplebsp

Project location: /zwork/fpga

☑ Create project subdirectory

Project will be created at: /zwork/fpga/simplebsp

Point to the correct folder destination (here **/zwork/dd/simplebsp**)
Press Next



Project Type

Specify the type of project to create.

● RTL Project
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.

☑ Do not specify sources at this time

○ Post-synthesis Project: You will be able to add sources, view device resources, run design analysis, planning and implementation.

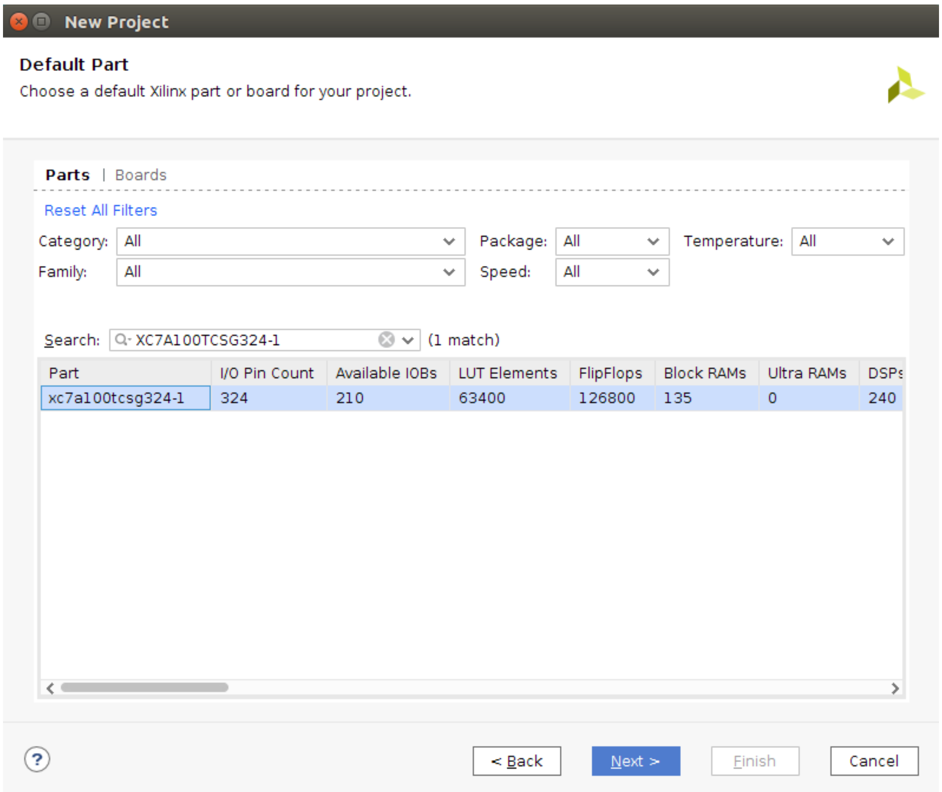
□ Do not specify sources at this time

○ I/O Planning Project
Do not specify design sources. You will be able to view part/package resources.

○ Imported Project
Create a Vivado project from a Synplify, XST or ISE Project File.

○ Example Project
Create a new Vivado project from a predefined template.

Choose RTL Project
Check the box “Do not specify sources at this time
Press Next



Default Part

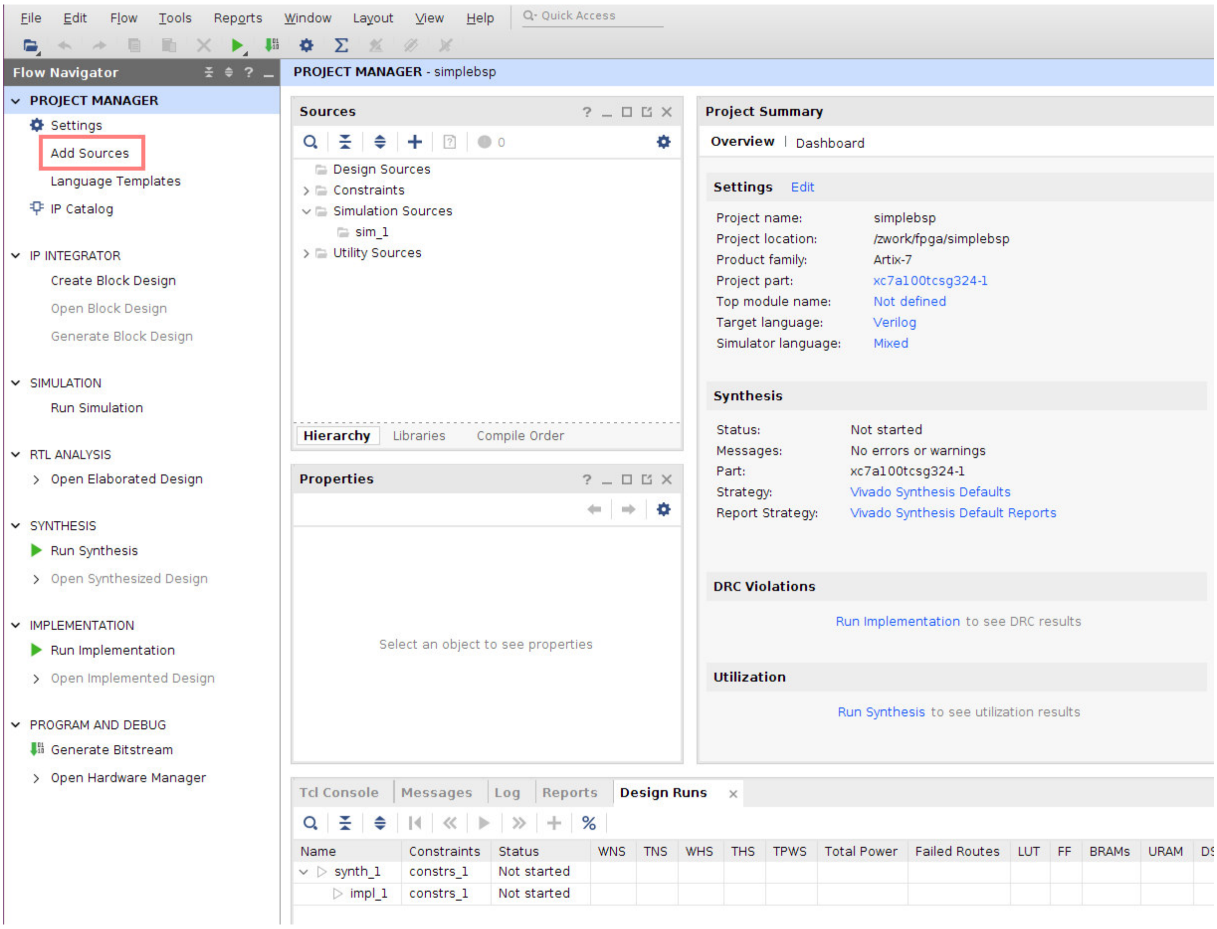
Choose a default Xilinx part or board for your project.

Category: All
Package: All
Temperature: All
Family: All
Speed: All

Search: XC7A100TCSG324-1 (1 match)

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs
xc7a100tcsg324-1	324	210	63400	126800	135	0	240

Select Board "Evaluation Development Board Arty A7 100" or FPGA-Part "XC7A100TCSG324-1"
Press Next and you get a summary of your project configuration
Press Finish



Project name: simplebsp
Project location: /zwork/fpga/simplebsp
Product family: Artix-7
Project part: xc7a100tcsg324-1
Top module name: Not defined
Target language: Verilog
Simulator language: Mixed

Synthesis

Status: Not started
Messages: No errors or warnings
Part: xc7a100tcsg324-1
Strategy: Vivado Synthesis Defaults
Report Strategy: Vivado Synthesis Default Reports

DRC Violations

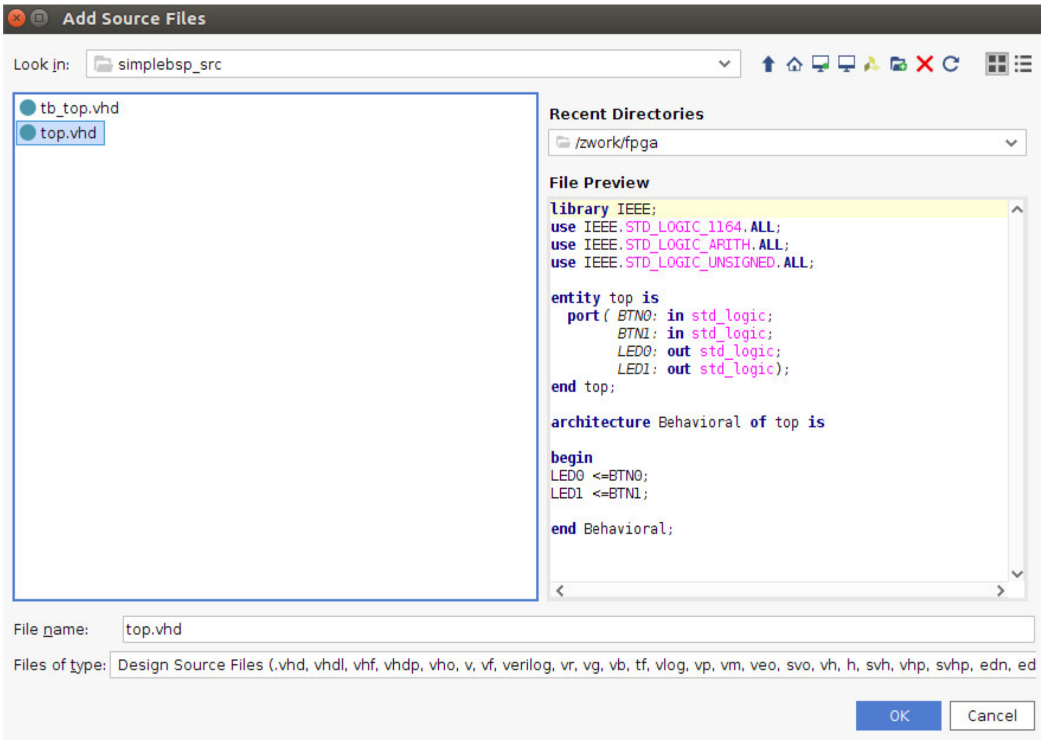
Run Implementation to see DRC results

Utilization

Run Synthesis to see utilization results

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

Project Manger->Add Soures-> Add or create design sources

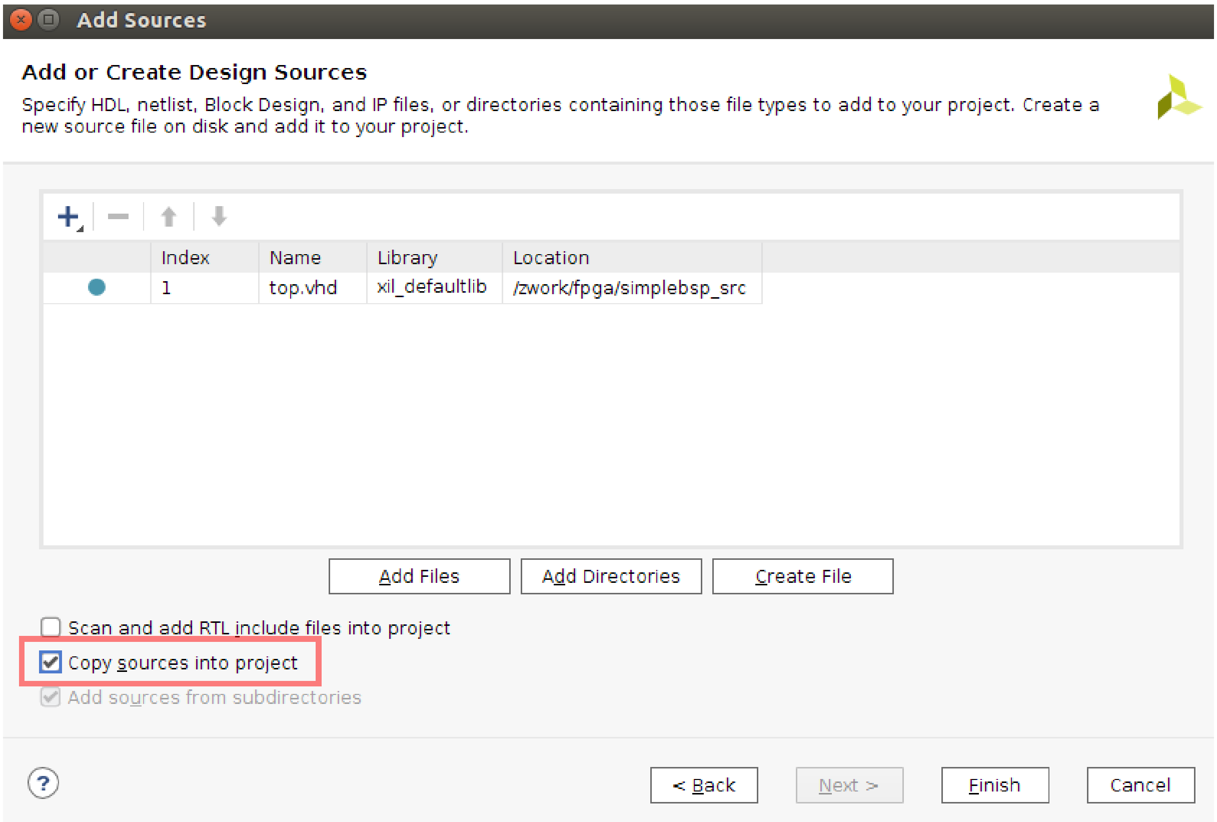


```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity top is
  port( BTN0: in std_logic;
        BTN1: in std_logic;
        LED0: out std_logic;
        LED1: out std_logic);
end top;

architecture Behavioral of top is

begin
LED0 <=BTN0;
LED1 <=BTN1;

end Behavioral;
```

File name: top.vhd

Files of type: Design Source Files (.vhd, vhdl, vhf, vhdp, vho, v, vf, verilog, vr, vg, vb, tf, vlog, vp, vm, veo, svo, vh, h, svh, vhp, svhp, edn, ed

Point to simplebsp_src and select (1 file: top.vhd)



Add Sources

Add or Create Design Sources

Specify HDL, netlist, Block Design, and IP files, or directories containing those file types to add to your project. Create a new source file on disk and add it to your project.

	Index	Name	Library	Location
●	1	top.vhd	xil_defaultlib	/zwork/fpga/simplebsp_src

☐ Scan and add RTL include files into project
☑ Copy sources into project
☑ Add sources from subdirectories

Check “Copy sources into project”
Press Finish

Project Manger->Add Sources-> Add constraints -> Point to folder simplebsp_src and select (1 file: Arty_ArtyA735_ArtyA7100_Master.xdc)
Check “Copy sources into project”
Press Finish

Project Manger->Add Sources-> Simulation Sources-> Point to folder simplebsp_src and select (1 file: tb_top.vhd)
Check “Copy sources into project”
Press Finish

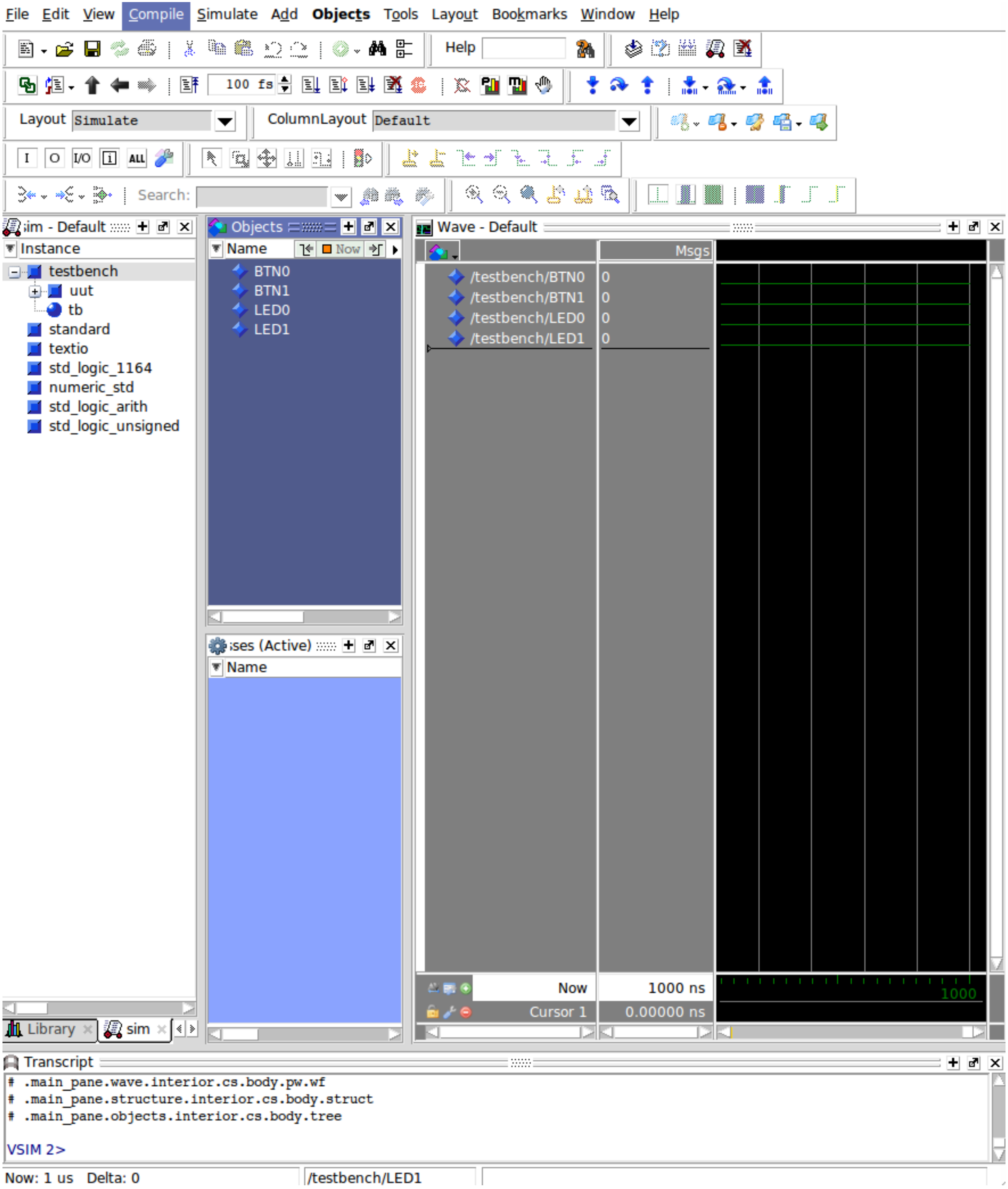
Project Manager -> Settings-> General ->Target language: VHDL
Press OK

SIMULATING WITH MODELSIM

Project Manager -> Settings-> Simulation ->Target Simulator: Modelsim Simulator
Project Manager -> Settings-> Simulation ->Compiled library location: /zwork/zsimlibmodelsimvivado201803
Press OK

Make sure you have saved any changes in your source code

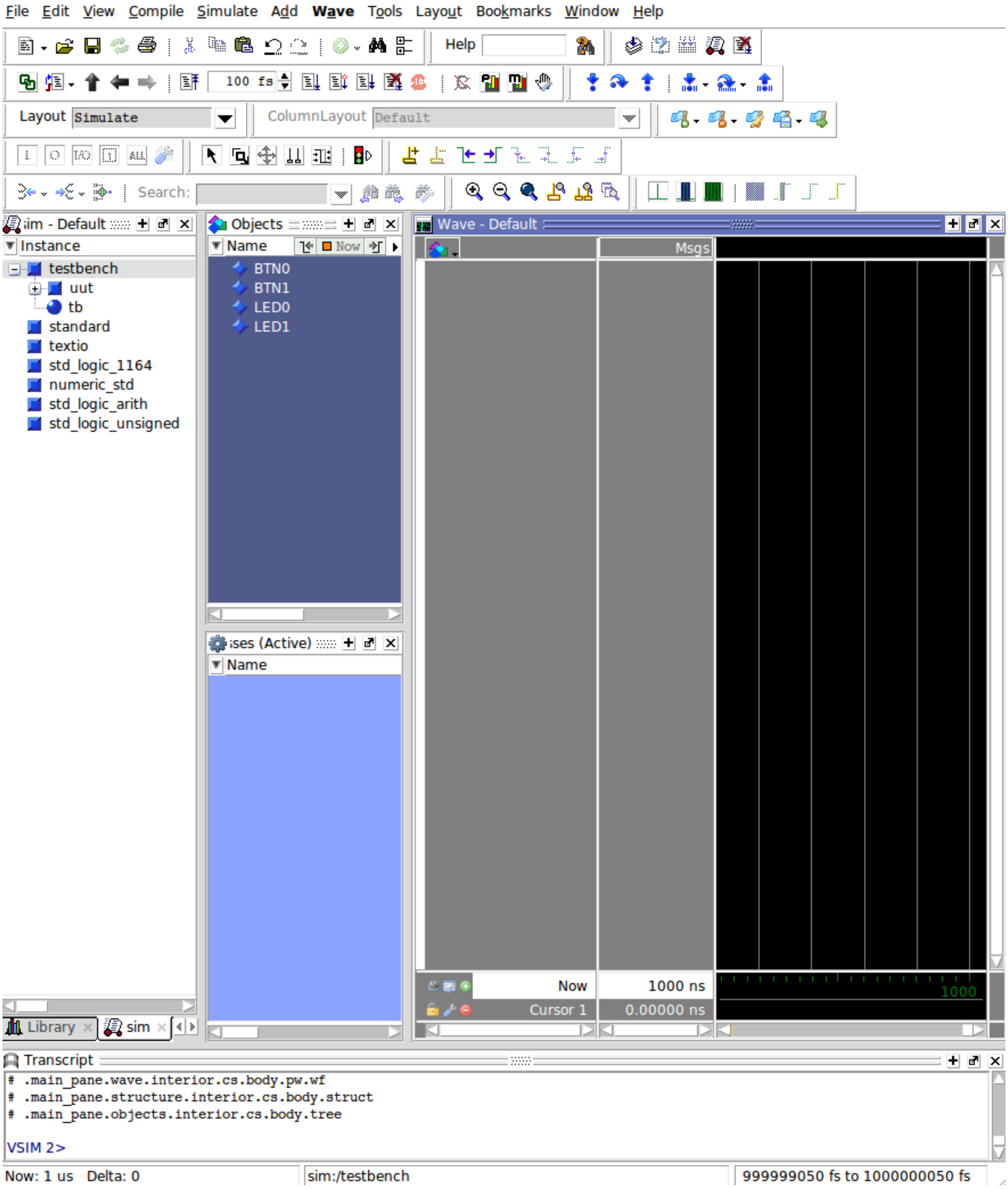
Project Manager-> Simulation->Run Simulation->Behavioral Simulation
Modelsim starts preconfigured



```
# .main_pane.wave.interior.cs.body.pw.wf
# .main_pane.structure.interior.cs.body.struct
# .main_pane.objects.interior.cs.body.tree

VSIM 2>
```

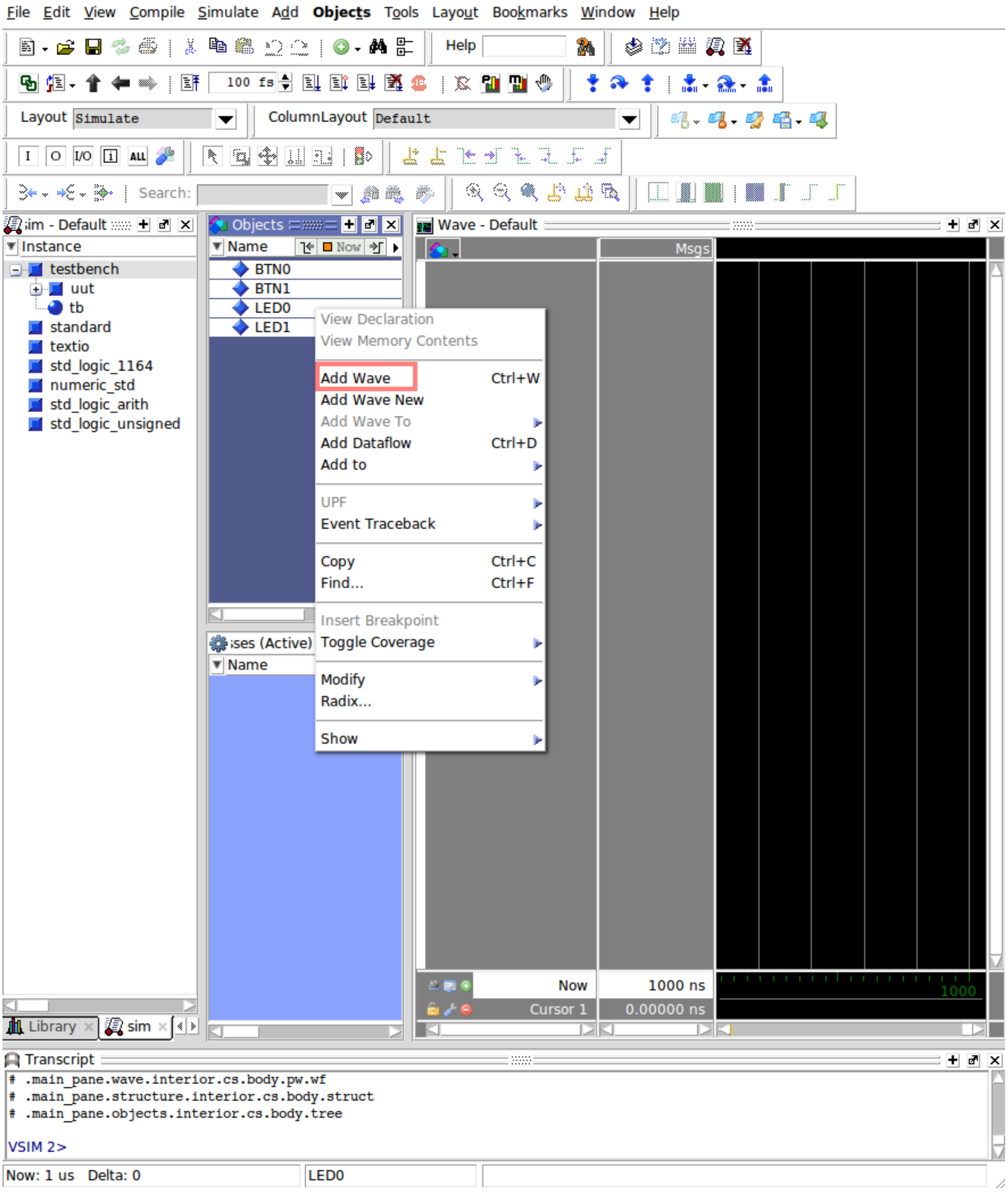
Now: 1 us Delta: 0 /testbench/LED1

The first thing we do is to reset the environment (by clicking Menu Wave -> Remove all)



```
# .main_pane.wave.interior.cs.body.pw.wf
# .main_pane.structure.interior.cs.body.struct
# .main_pane.objects.interior.cs.body.tree

VSIM 2>
```

Now: 1 us Delta: 0 | sim:/testbench | 999999050 fs to 1000000050 fs

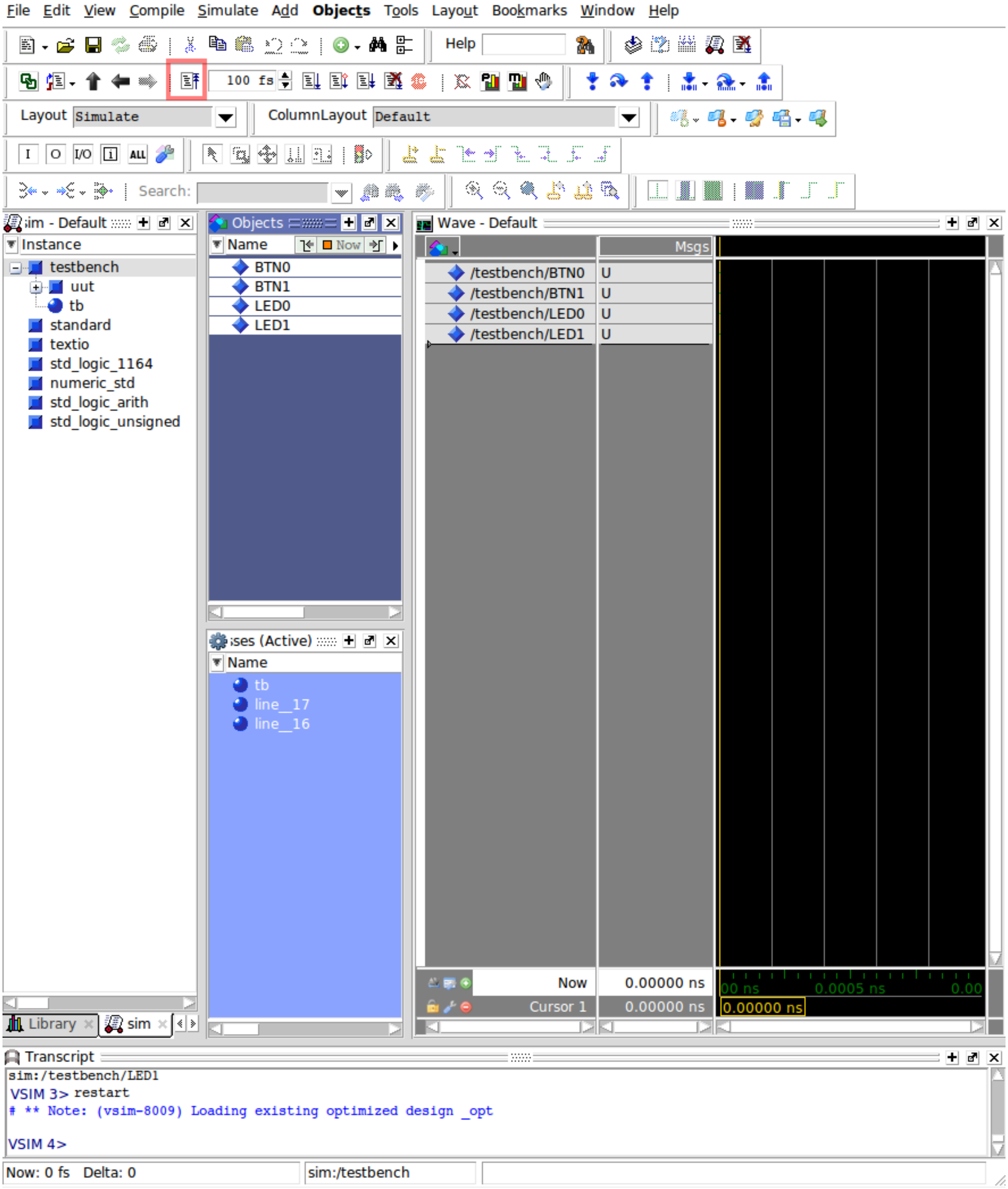


```
# .main_pane.wave.interior.cs.body.pw.wf
# .main_pane.structure.interior.cs.body.struct
# .main_pane.objects.interior.cs.body.tree

VSIM 2>
```

Now: 1 us Delta: 0   LED0

Now we select the signals we want to investigate (select the correct instance (here testbench) and then the signals BTN0, BTN1, LED0, LED1 (by keeping ctrl pressed while selecting these signals) Right click on these selected signals and click on “Add to wave”

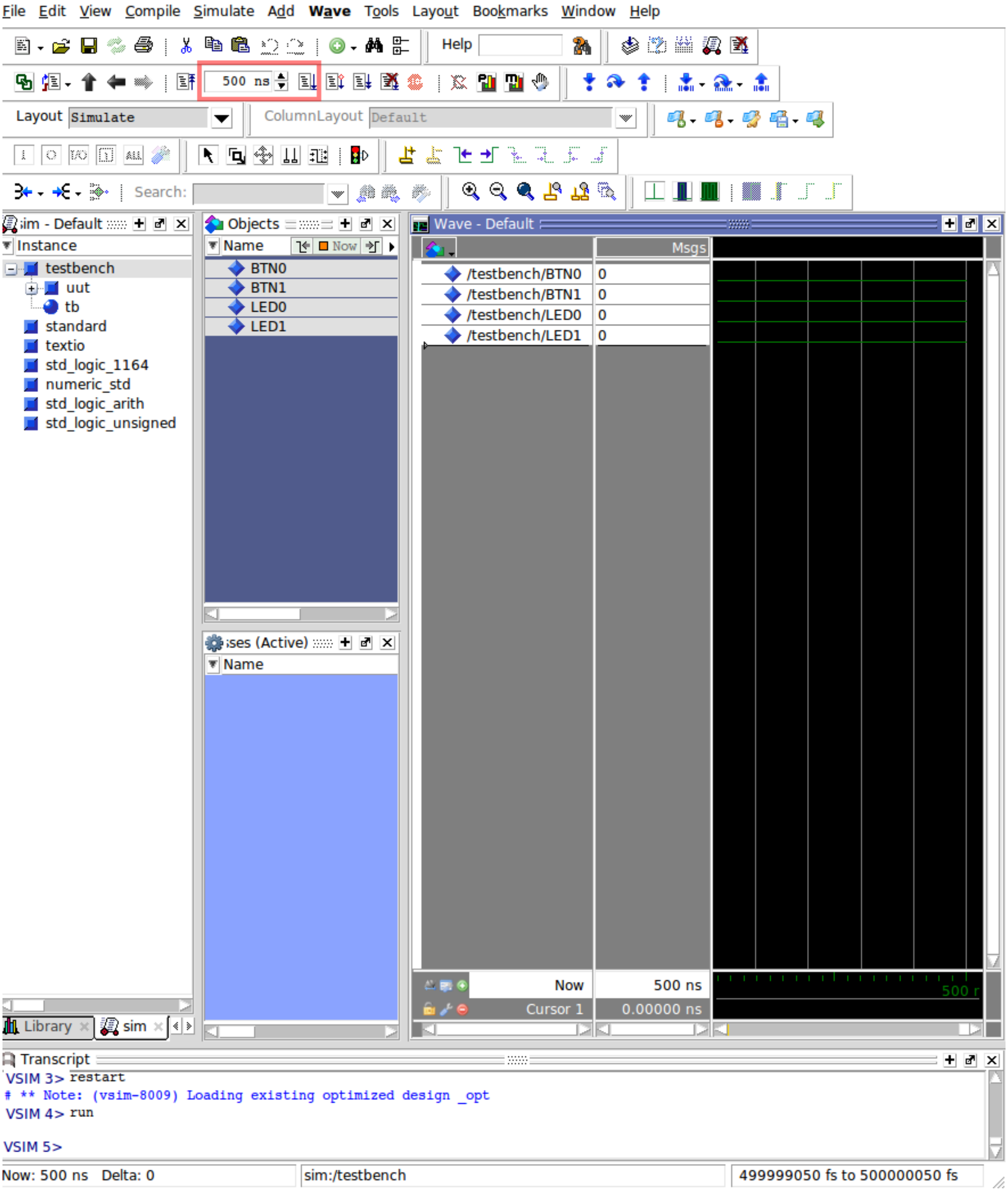
Clear the wave window by pressing the “restart button”(see marked in red in the toolbar
Confirm the next dialog with “ok



```
sim:/testbench/LED1
VSIM 3> restart
# ** Note: (vsim-8009) Loading existing optimized design _opt

VSIM 4>
```

Now: 0 fs   Delta: 0   sim:/testbench

Now choose the simulation time e.g. “500 ns” (left red circle) in the toolbar
To simulate press the “run button” (right red circle) in the toolbar

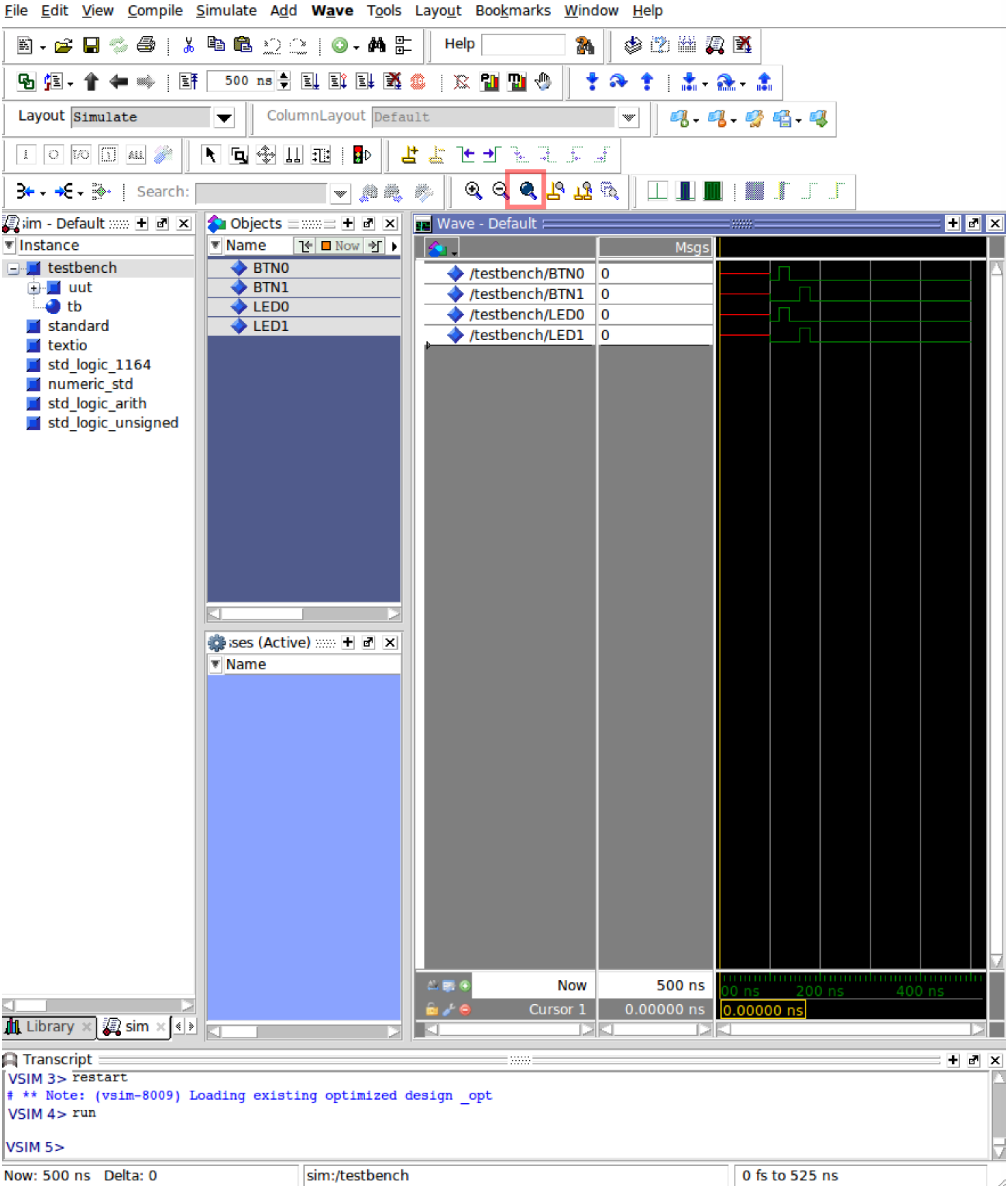


	Msgs
/testbench/BTN0	0
/testbench/BTN1	0
/testbench/LED0	0
/testbench/LED1	0
Now	500 ns
Cursor 1	0.00000 ns

```
VSIM 3> restart
# ** Note: (vsim-8009) Loading existing optimized design _opt
VSIM 4> run

VSIM 5>
```

Now: 500 ns   Delta: 0   sim:/testbench   499999050 fs to 500000050 fs

To adapt the wave window so that the whole simulation time is displayed with scrolling, select the wave windows so that its header is highlighted in blue and press the “zoom full” button (marked in red) in the toolbar.



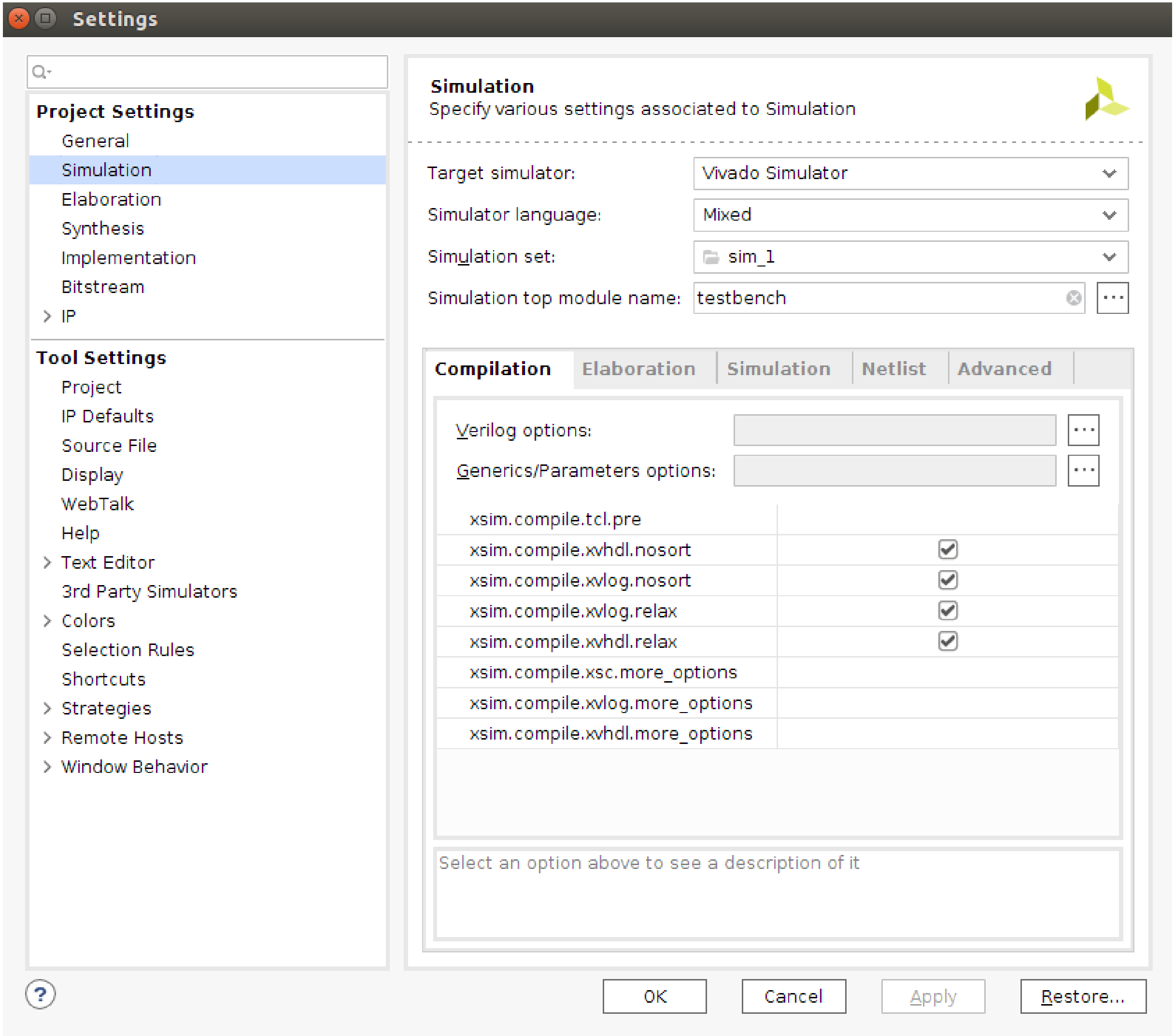
```
VSIM 3> restart
# ** Note: (vsim-8009) Loading existing optimized design _opt
VSIM 4> run

VSIM 5>
```

Now: 500 ns Delta: 0 | sim:/testbench | 0 fs to 525 ns

The simulated Waveform should show that led0 is high if btn0 is high and the same for btn1 und led1.

ALTERNATIVE: SIMULATING WITH THE INTEGRATED VIVADO SIMULATOR

At first make sure Vivado Simulator is selected as Simulator:
Project Manager -> Settings-> Simulation ->Target Simulator: Vivado Simulator



Simulation
Specify various settings associated to Simulation

Target simulator: Vivado Simulator

Simulator language: Mixed

Simulation set: sim_1

Simulation top module name: testbench

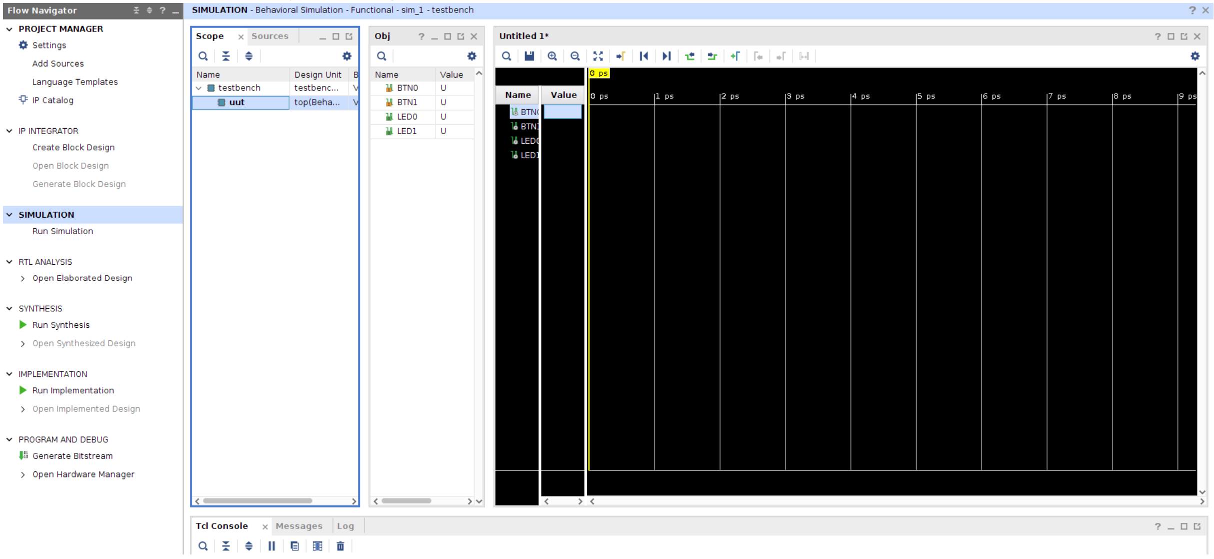
Verilog options:

Generics/Parameters options:

xsim.compile.tcl.pre	
xsim.compile.xvhdl.nosort	☑
xsim.compile.xvlog.nosort	☑
xsim.compile.xvlog.relax	☑
xsim.compile.xvhdl.relax	☑
xsim.compile.xsc.more_options	
xsim.compile.xvlog.more_options	
xsim.compile.xvhdl.more_options	

Select an option above to see a description of it

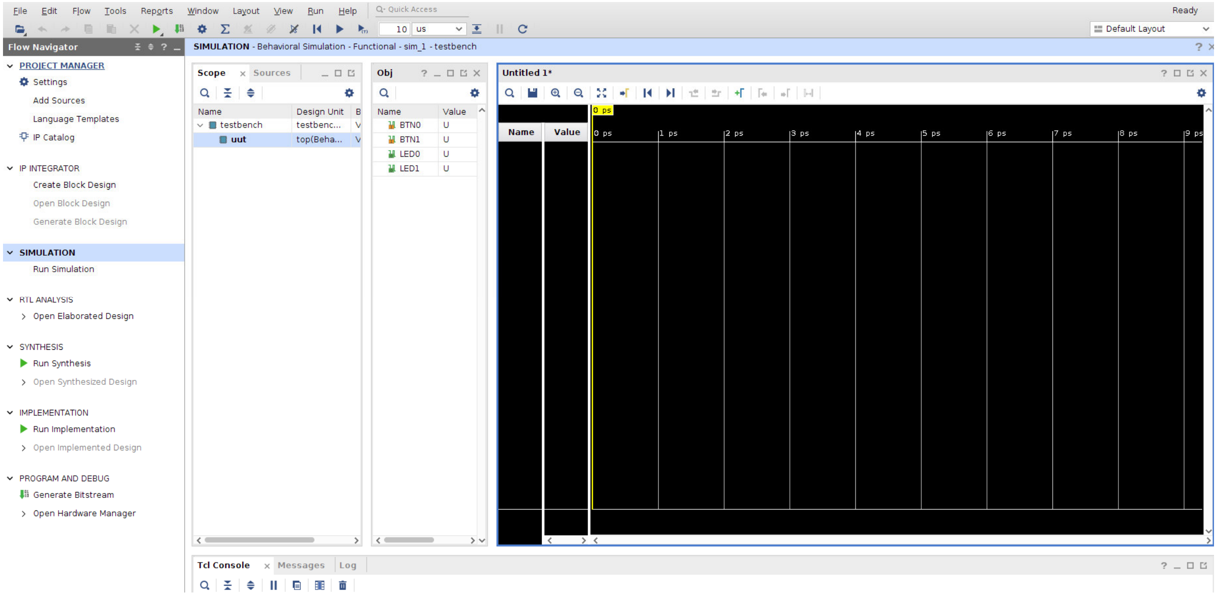
Project Manager-> Simulation-> Run Simulation-> Behavioral Simulation
Vivado Simulator starts preconfigured



Name	Design Unit
testbench	testbenc...
uut	top(Beha...

Name	Value
BTN0	U
BTN1	U
LED0	U
LED1	U

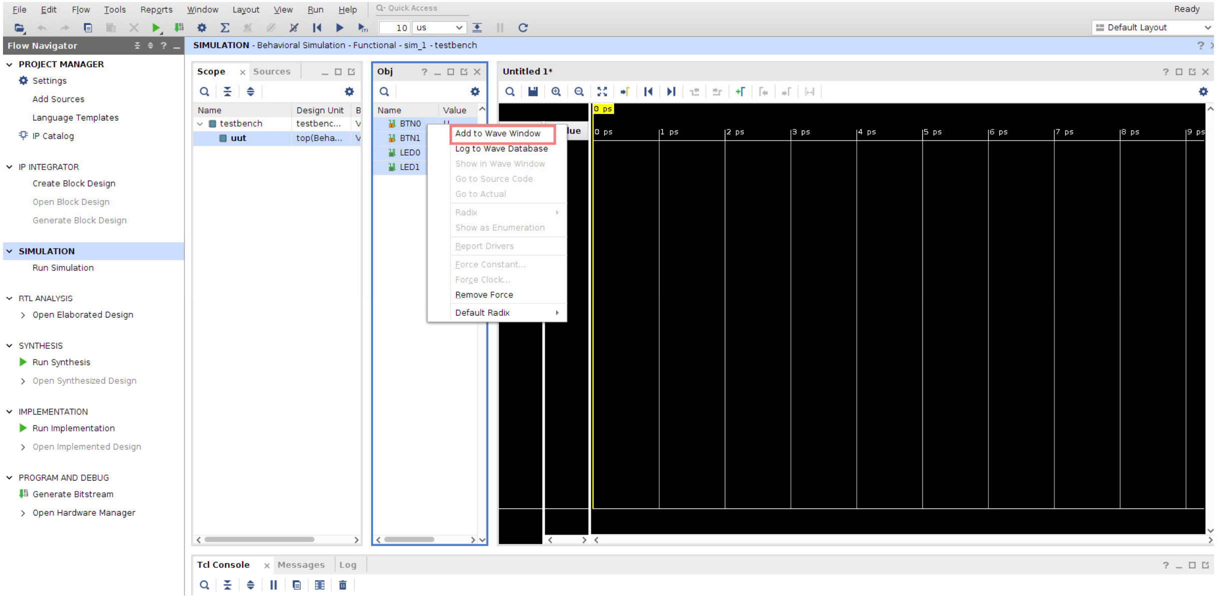
The first thing we do is to reset the environment (by right clicking on the selected signals in the wave window and clicking on “delete”)



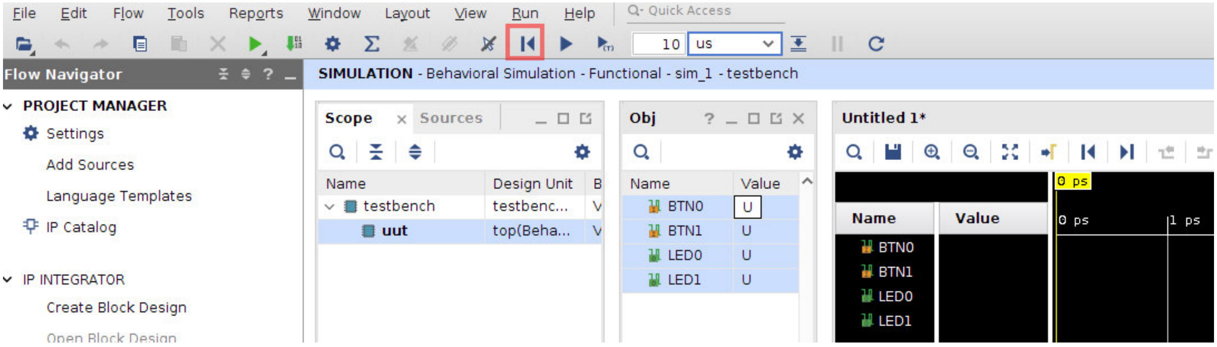
Name	Design Unit
testbench	testbenc...
uut	top(Beha...

Name	Value
BTN0	U
BTN1	U
LED0	U
LED1	U

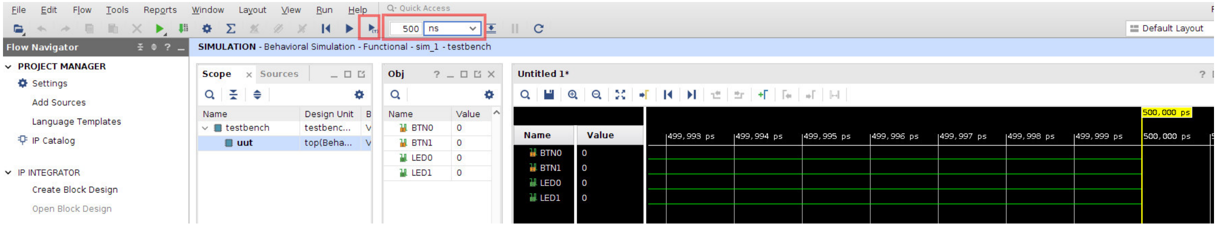
Select the correct instance (here testbench)
Then select the correct signals (btn0, btn1, led0, led1) (by keeping the ctrl key pressed while selecting the signals)
Right click on the selected signals and click on “Add to wave Window”



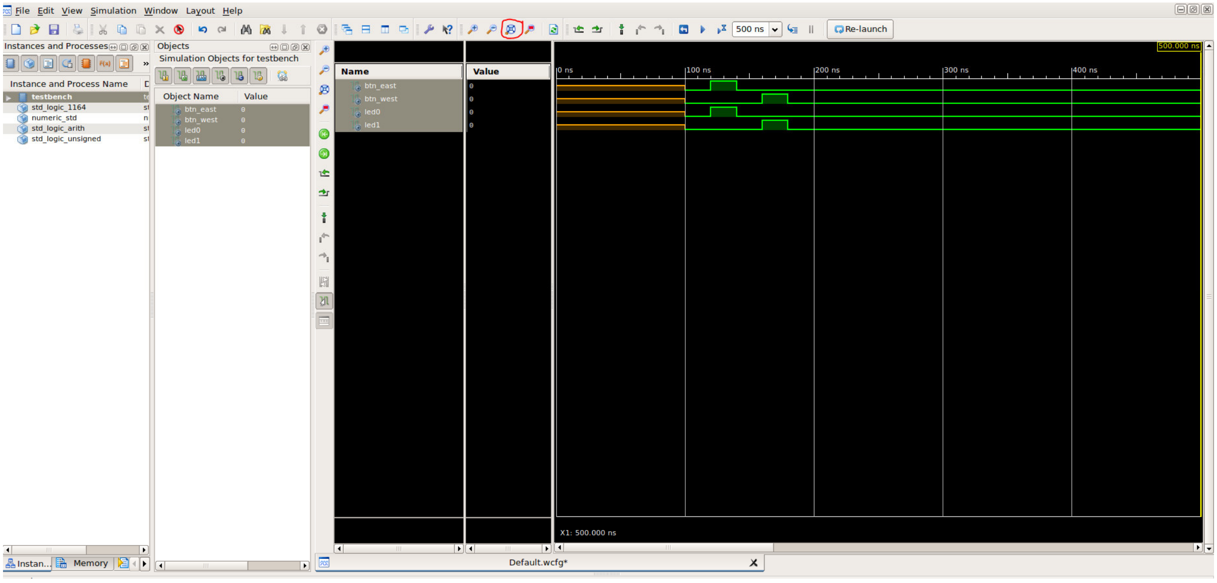
Clear the wave window by pressing the “restart button”(see marked in red) in the toolbar

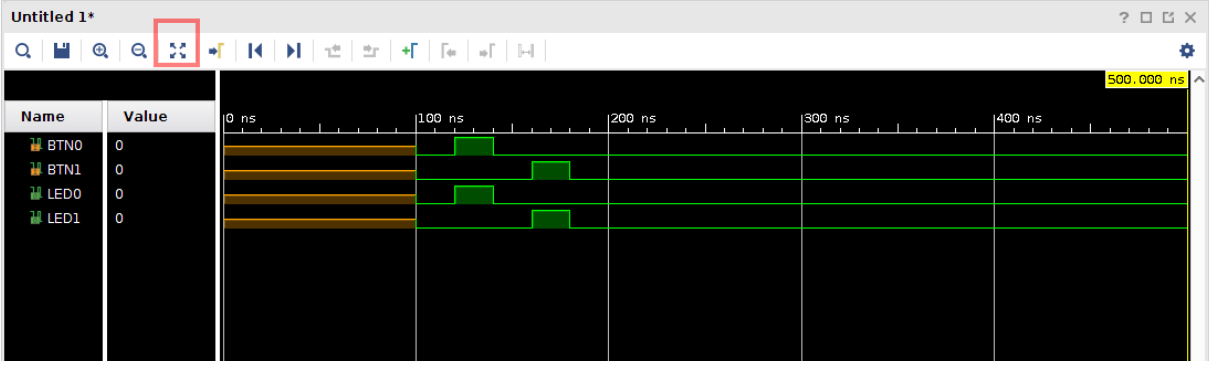


Now choose the simulation time e.g. “500 ns” (right red marking) in the toolbar
To simulate press the “run for xxx button” (left red marking) in the toolbar



To adapt the wave window so that the whole simulation time is displayed with scrolling, press the "zoom to fit" button (red marking) in toolbar

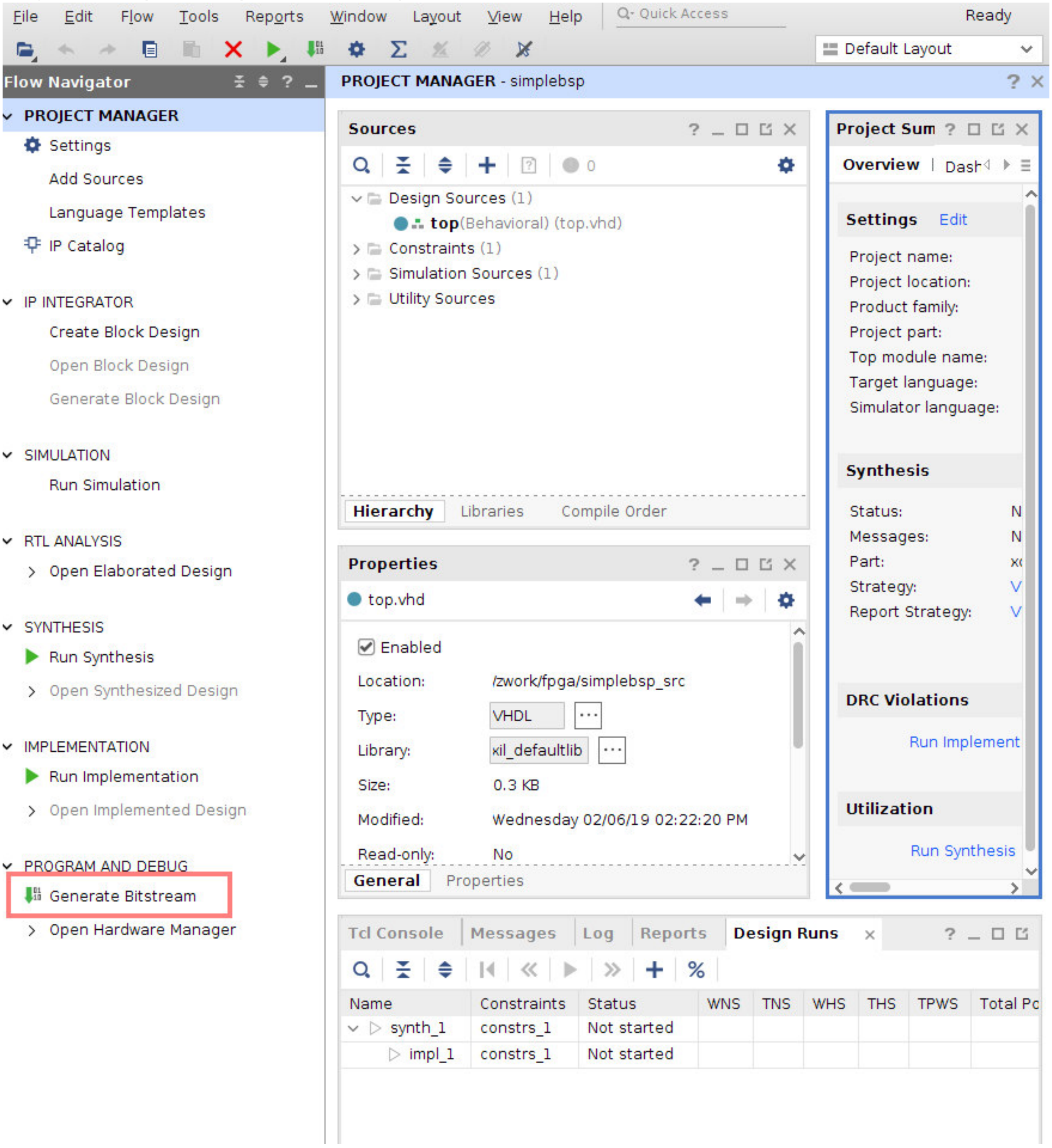




The simulated Waveform should show that led0 is high if btn0 is high and the same for btn1 und led1.

IMPLEMENTING AND LOADING A BITFILE

Project Manager-> Program and Debug -> Generate Bitstream



☑ Enabled

Location: /zwork/fpga/simplebsp_src

Type: VHDL

Library: xil_defaultlib

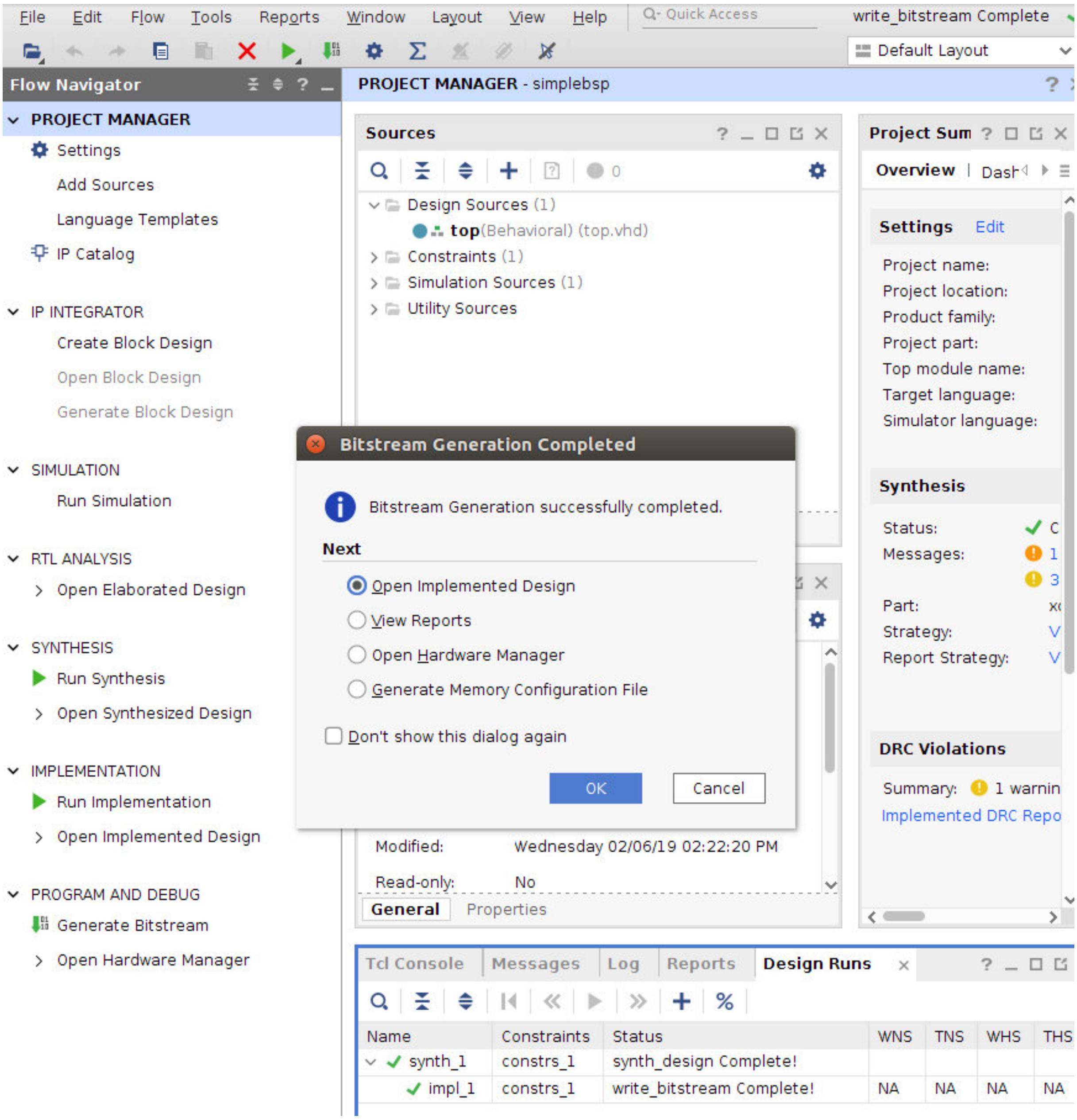
Size: 0.3 KB

Modified: Wednesday 02/06/19 02:22:20 PM

Read-only: No

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Pc
synth_1	constrs_1	Not started						
impl_1	constrs_1	Not started						

Conform the upcoming dialog “launch Runs” with OK
Now the process for synthesis, implementation and generate programming file is started
Keep an eye on warnings and errors (while waiting) in the tabs Messages and Tcl Console



File Edit Flow Tools Reports Window Layout View Help Quick Access write_bitstream Complete

Default Layout

Flow Navigator

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

PROGRAM AND DEBUG
- Generate Bitstream
- Open Hardware Manager

PROJECT MANAGER - simplebsp

Sources

- Design Sources (1)
  - top(Behavioral) (top.vhd)
- Constraints (1)
- Simulation Sources (1)
- Utility Sources

Project Sum

Overview | Dash

Settings Edit

Project name:
Project location:
Product family:
Project part:
Top module name:
Target language:
Simulator language:

Synthesis

Status: ✓ C
Messages: 1
3

Part: x
Strategy: V
Report Strategy: V

DRC Violations

Summary: 1 warnin
Implemented DRC Repo

Bitstream Generation Completed

Bitstream Generation successfully completed.

Next

● Open Implemented Design
○ View Reports
○ Open Hardware Manager
○ Generate Memory Configuration File

□ Don't show this dialog again

OK Cancel

Modified: Wednesday 02/06/19 02:22:20 PM
Read-only: No

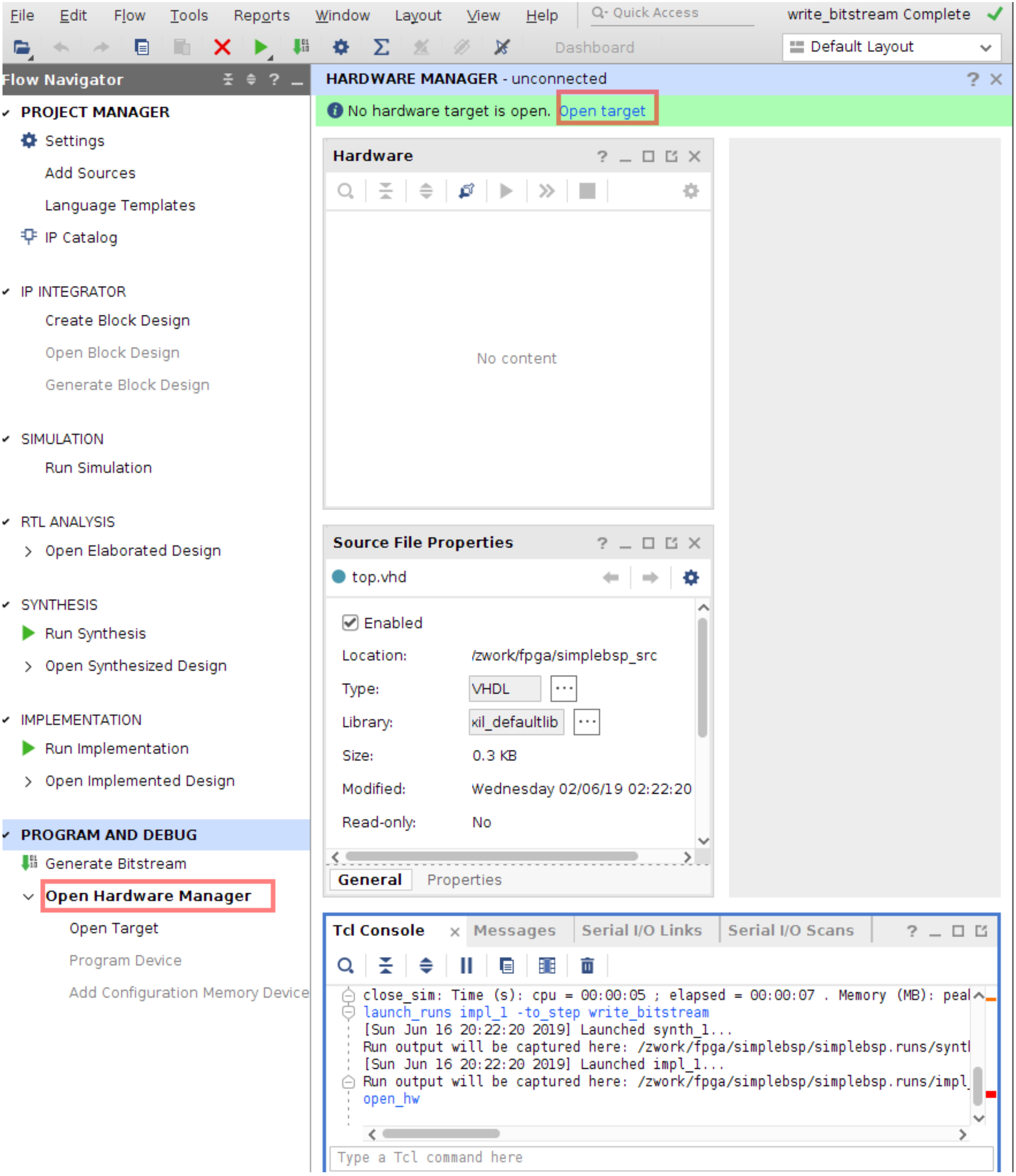
General Properties

Tcl Console Messages Log Reports Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS
✓ synth_1	constrs_1	synth_design Complete!				
✓ impl_1	constrs_1	write_bitstream Complete!	NA	NA	NA	NA

To load the bitfile to the FPGA, you have to make sure your FPGA board is connected and switched on.

Then Project Manager-> Open Hardware Manager



write_bitstream Complete ✓

HARDWARE MANAGER - unconnected

No hardware target is open. Open target

No content

top.vhd

☑ Enabled

Location: /zwork/fpga/simplebsp_src

Type: VHDL

Library: xil_defaultlib

Size: 0.3 KB

Modified: Wednesday 02/06/19 02:22:20

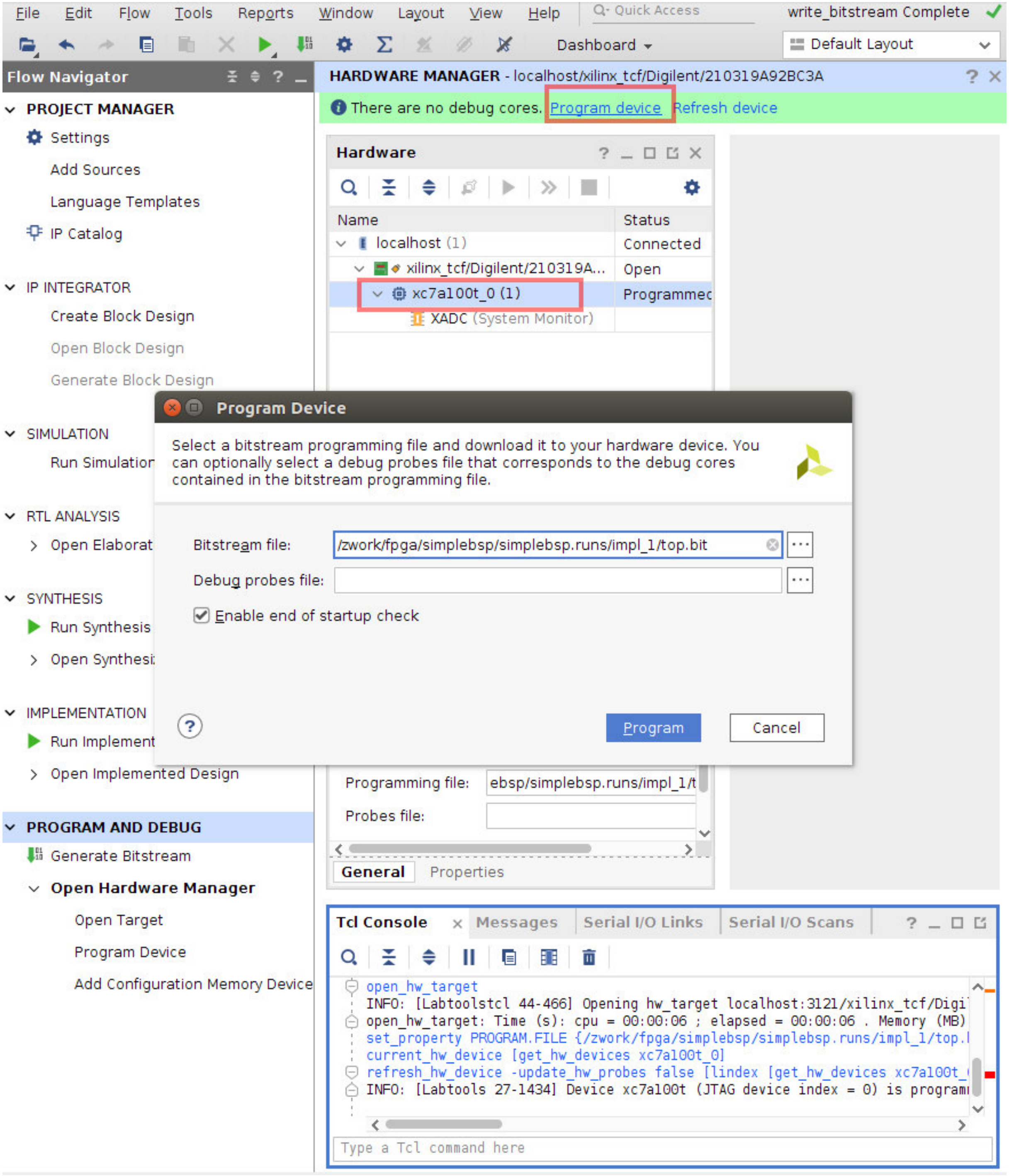
Read-only: No

```
close_sim: Time (s): cpu = 00:00:05 ; elapsed = 00:00:07 . Memory (MB): peal
launch_runs impl_1 -to_step write_bitstream
[Sun Jun 16 20:22:20 2019] Launched synth_1...
Run output will be captured here: /zwork/fpga/simplebsp/simplebsp.runs/syntl
[Sun Jun 16 20:22:20 2019] Launched impl_1...
Run output will be captured here: /zwork/fpga/simplebsp/simplebsp.runs/impl_
open_hw
```

Type a Tcl command here

Then double click on “Open Target” -> Autoconnect

Program Device (or right click on FPGA type)



There are no debug cores. Program device Refresh device

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210319A...	Open
xc7a100t_0 (1)	Programmec
XADC (System Monitor)	

Program Device

Select a bitstream programming file and download it to your hardware device. You can optionally select a debug probes file that corresponds to the debug cores contained in the bitstream programming file.

Bitstream file: /zwork/fpga/simplebsp/simplebsp.runs/impl_1/top.bit

Debug probes file:

☑ Enable end of startup check

Programming file: ebsp/simplebsp.runs/impl_1/t

Probes file:

```
open_hw_target
INFO: [Labtoolstcl 44-466] Opening hw_target localhost:3121/xilinx_tcf/Digi
open_hw_target: Time (s): cpu = 00:00:06 ; elapsed = 00:00:06 . Memory (MB)
set_property PROGRAM.FILE {/zwork/fpga/simplebsp/simplebsp.runs/impl_1/top.l
current_hw_device [get_hw_devices xc7a100t_0]
refresh_hw_device -update_hw_probes false [lindex [get_hw_devices xc7a100t_
INFO: [Labtools 27-1434] Device xc7a100t (JTAG device index = 0) is program
```

Type a Tcl command here

Select the correct bitfile **(e.g. /zwork/fpga/simplebsp/simplebsp.runs/impl_1/top.bit)** and conform with clicking on “Program”

You now get a progressbar showing the progress in programming the FPGA.

After the programming you can test the FPGA image. Press the buttons btn0 and btn1 -> The two leds led4 and led5 should illuminate while pressing the buttons btn0 and btn1.

ALTERNATIVE: Programming a bitfile outside a project

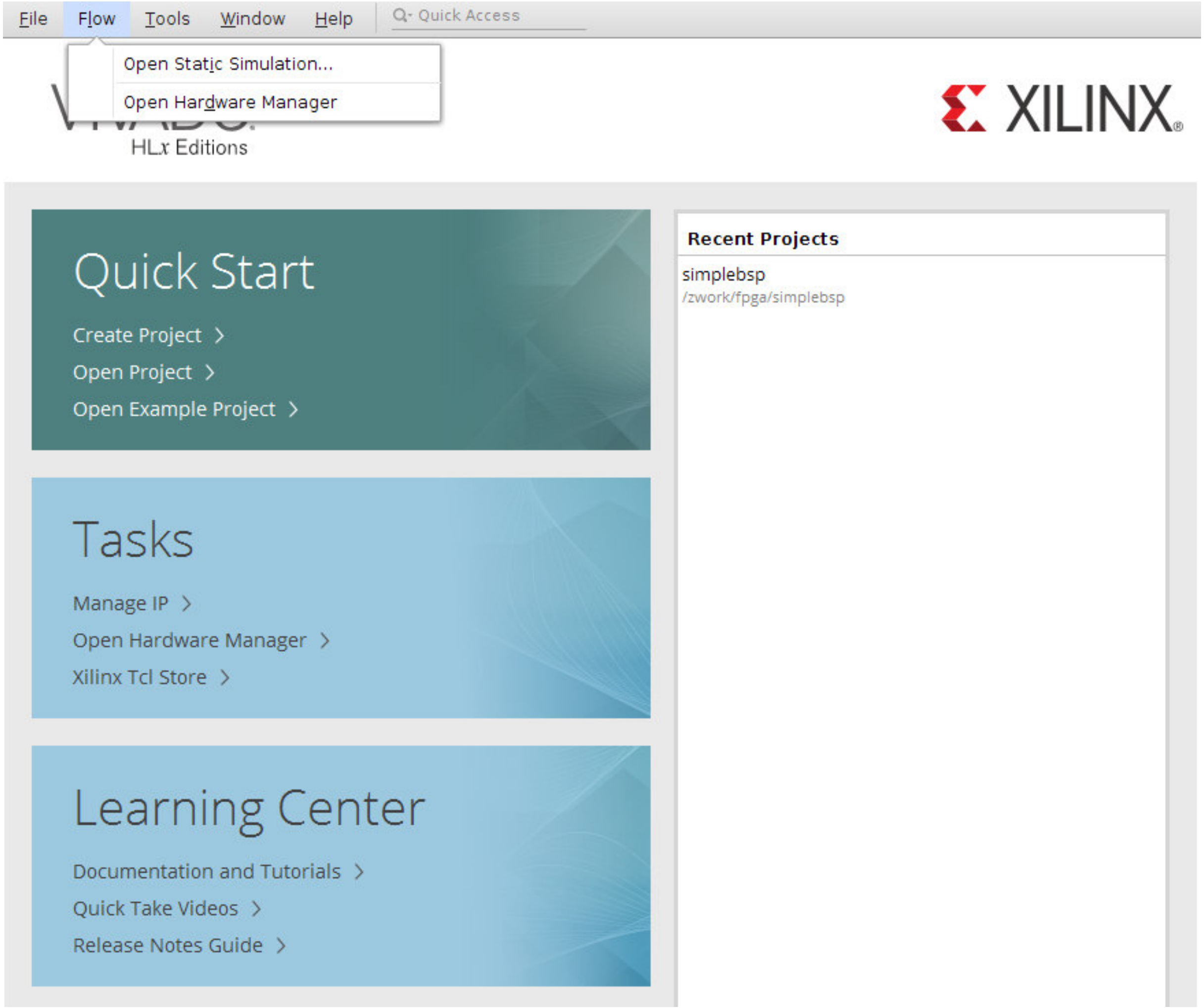
Copy and unzip the given source files to **/zwork/dd/simplebsp_src**

Open a Linux terminal and type:
./m1data.sh //if promted the password is ubupw

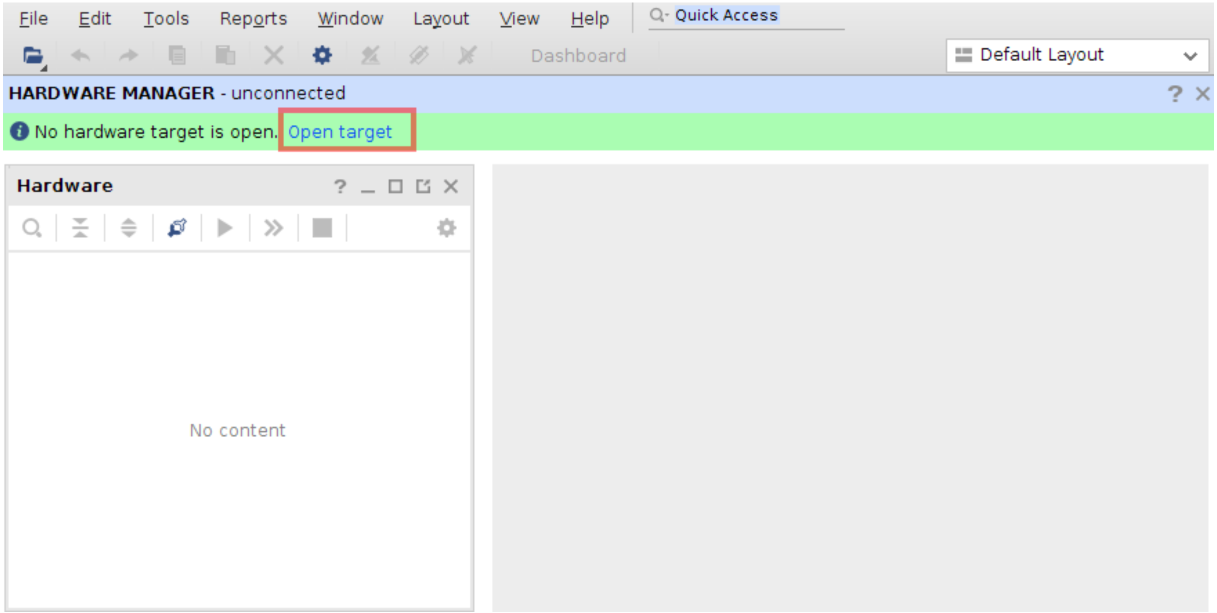
source Modelsim106c_settings64.sh
source XilinxVivado201803_settings64.sh

vivado&

To load the bitfile to the FPGA, you have to make sure your FPGA board is connected and switched on.
Then Flow-> Open Hardware Manager

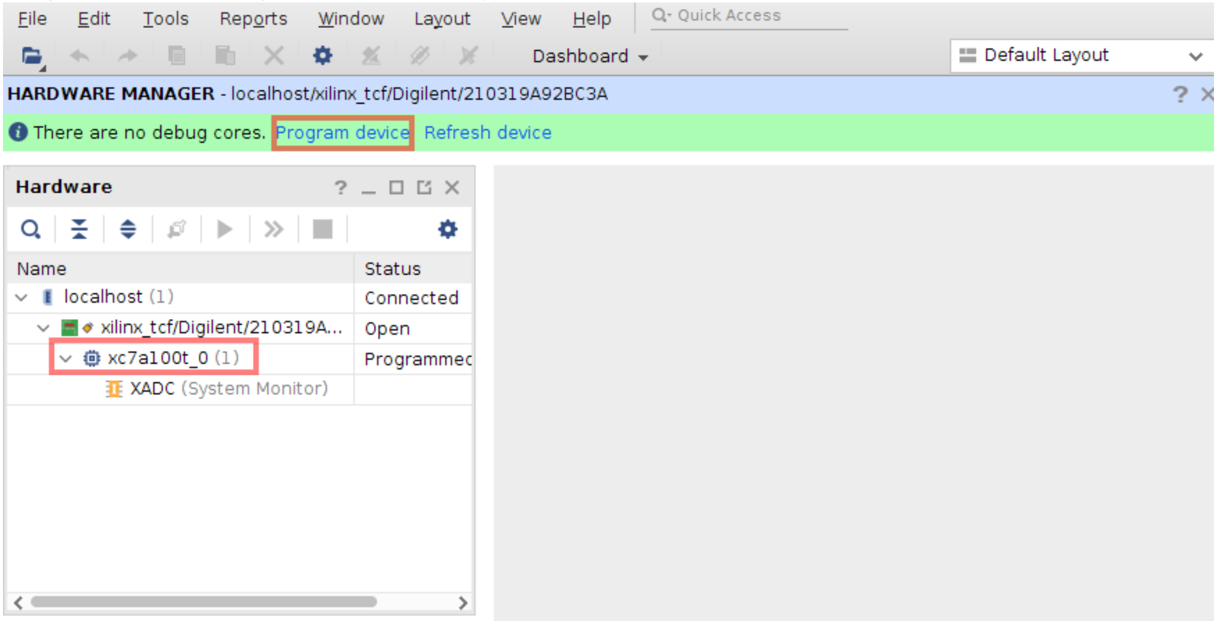


Then double click on “Open Target” -> Autoconnect



File Edit Tools Reports Window Layout View Help
Quick Access
Dashboard
Default Layout
HARDWARE MANAGER - unconnected
No hardware target is open. Open target
Hardware
No content

Program Device (or right click on FPGA type)



File Edit Tools Reports Window Layout View Help
Quick Access
Dashboard
Default Layout
HARDWARE MANAGER - localhost/xilinx_tcf/Digilent/210319A92BC3A
There are no debug cores. Program device Refresh device
Hardware

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210319A...	Open
xc7a100t_0 (1)	Programmec
XADC (System Monitor)	

Select the correct bitfile **(e.g. /zwork/dd/simplebsp/simplebsp.runs/impl_1/top.bit)** and confirm with clicking on “Program”

You now get a progressbar showing the progress in programming the FPGA.

After the programming you can test the FPGA image. Press the buttons btn0 and btn1 -> The two leds led4 and led5 should illumiate while pressing the buttons btn0 and btn1