



# Simulation with VHDL



# Simulation with VHDL

---


## Index

- ❑ Simulation types in ISE simulator
- ❑ How to create a simulation *test-bench*
- ❑ *Test-bench* of a combinational entity
- ❑ Basics of the ISE simulator



# Simulation with VHDL

## Simulation types in ISE simulator

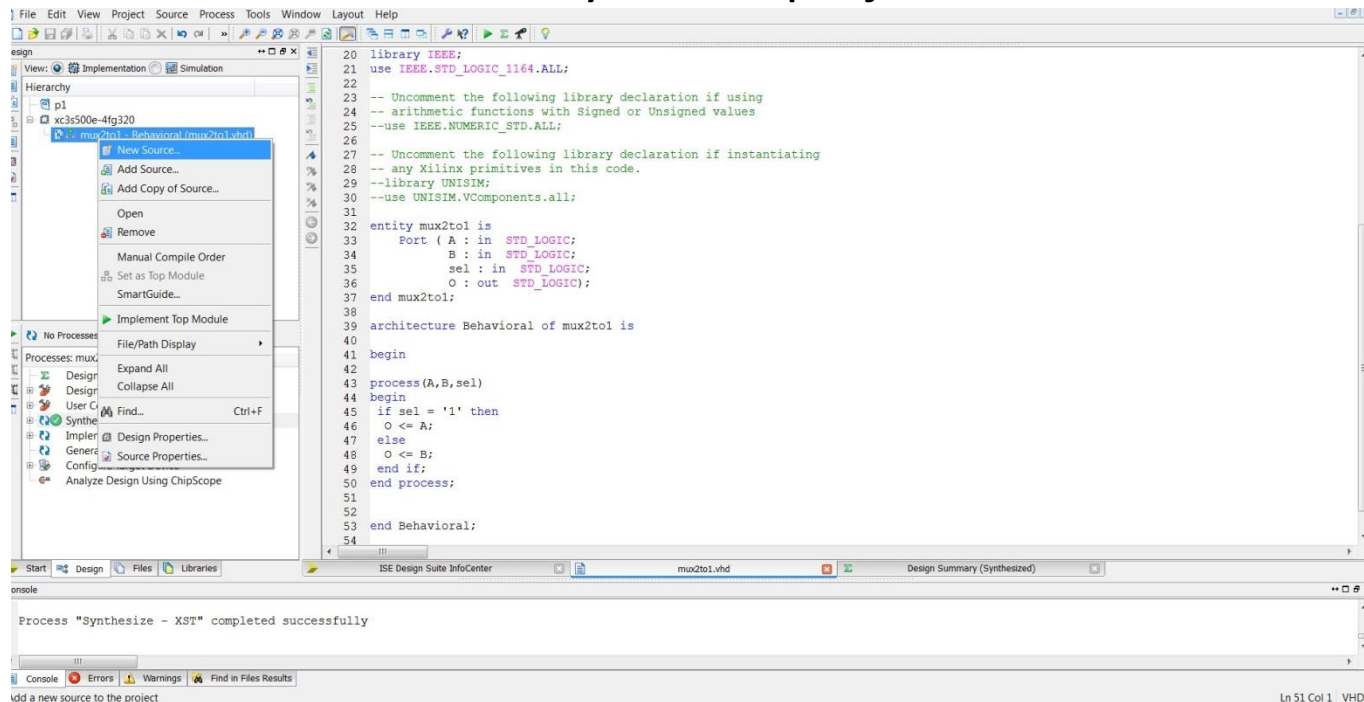
- ❑ Xilinx ISE Simulator features four different simulation types
    - ❑ *Behavioral* simulation
    - ❑ *Post-Translate* simulation
    - ❑ *Post-Map* simulation
    - ❑ *Post-Route* simulation
  - ❑ We will focus on the *behavioral* simulation that will verify the functional algorithm of the VHDL code
- 
- No delays are considered
- At each simulation more and more details about delays in logic components and routes are added
- Representative delay values of the implementation



# Simulation with VHDL

## How to create a simulation test-bench

- ❑ Test-bench files need to be created to simulate the behavior of an entity: its architecture
- ❑ A VHDL test-bench is code that is only used for simulation purposes
- ❑ How to add a test-bench to your ISE project

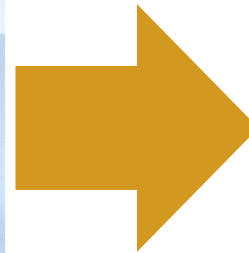
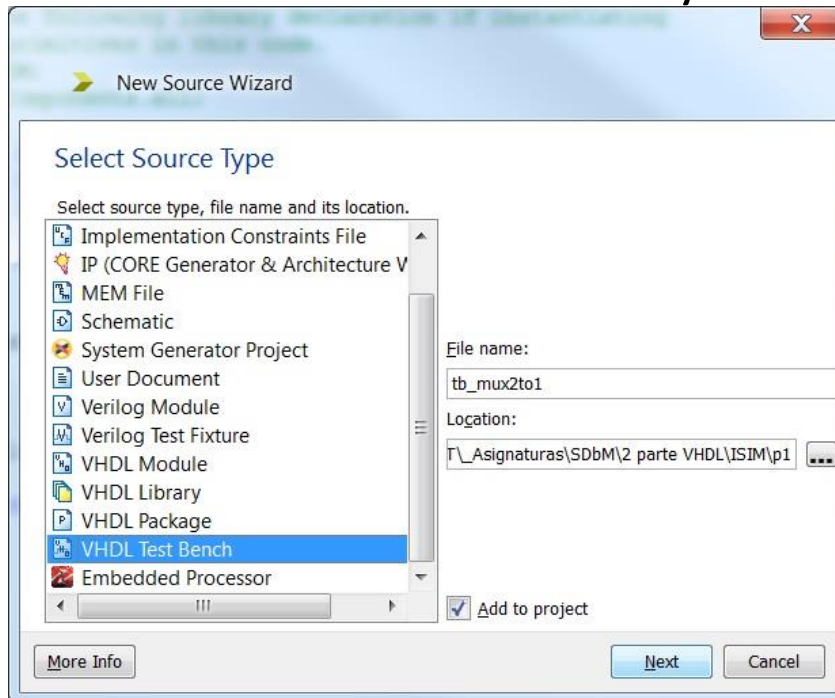




# Simulation with VHDL

## How to create a simulation test-bench

- ❑ Test-bench files need to be created to simulate the behavior of an entity: its architecture
- ❑ A VHDL test-bench is code that is only used for simulation purposes
- ❑ How to add a test-bench to your ISE project



```
64 BEGIN
65
66   -- Instantiate the Unit Under Test (UUT)
67   uut: mux2to1 PORT MAP (
68     A => A,
69     B => B,
70     sel => sel,
71     O => O
72   );
73
74   -- Clock process definitions
75   <clock>_process :process
76   begin
77     <clock> <= '0';
78     wait for <clock>_period/2;
79     <clock> <= '1';
80     wait for <clock>_period/2;
81   end process;
82
83   -- Stimulus process
84   stim_proc: process
85   begin
86     -- hold reset state for 100 ns.
87     wait for 100 ns;
88
89     wait for <clock>_period*10;
90
91     -- insert stimulus here
92
93     wait;
94   end process;
95
96
97 END;
```



# Simulation with VHDL

## Test-bench of a combinational entity

- ❑ Example: simulation of a comparator
- ❑ The VHDL code of the comparator is

```
-- 2 words comparator
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY comparator IS
    PORT ( A: IN std_logic_vector(7 DOWNTO 0);
          B: IN std_logic_vector(7 DOWNTO 0);
          enable: IN std_logic;
          AlessB: OUT std_logic;
          AequalB: OUT std_logic;
          AgreaterB: OUT std_logic);
END comparator ;

ARCHITECTURE Behavioral OF comparator IS
BEGIN

PROCESS(A, B, enable)
BEGIN
    IF (enable = '0') THEN
        AlessB<='0'; AequalB<='0'; AgreaterB <= '0';
    ELSE
        IF ( A < B ) THEN --A less than B
            AlessB <= '1';
```

```
        ELSE
            AlessB <= '0';
        END IF;
        IF ( A > B ) THEN --A greater than B
            AgreaterB <= '1';
        ELSE
            AgreaterB <= '0';
        END IF;
        IF ( A = B ) THEN --A equals B
            AequalB <= '1';
        ELSE
            AequalB <= '0';
        END IF;
    END IF;
END PROCESS;
END Behavioral;
```



# Simulation with VHDL

## Test-bench of a combinational entity

- ❑ The ISE wizard of the VHDL test-bench module generates code ready to “fill in the gaps”, including clock, **input** and **output ports** of the **unit under test** (uut)
- ❑ Note that for the simulation of combinational components the lines dedicated to the clock must be removed
- ❑ Set initial values for the input **ports**

```
--Inputs
-- Here initial values for signals can be given. By default 0

signal A : std_logic_vector(7 downto 0) := (others => '0');
signal B : std_logic_vector(7 downto 0) := (others => '0');
--signal enable : std_logic := '0';
signal enable : std_logic := '1'; -- I change the initial value of enable
```



# Simulation with VHDL

## Test-bench of a combinational entity

- ❑ Test-bench wizard generates two processes
  - ❑ One for the clock, not applicable for combinational systems
  - ❑ One for the rest of inputs

```
-- Stimulus process
stim_proc: process
begin
    -- hold reset state for 100 ns.
    wait for 100 ns;

    -- insert stimulus here

    wait; -- wait forever
end process;
```





# Simulation with VHDL

## Test-bench of a combinational entity

- ❑ Test-bench wizard generates two processes
  - ❑ One for the clock, not applicable for combinational systems
  - ❑ One for the rest of inputs

```
-- insert stimulus here

A <= "00001111"; -- I assign values to A and B
B <= "00000011";
wait for 100 ns; -- time interval between changes in the values

B <= "00100011"; -- A is not changed so no need to repeat
wait for 100 ns;

B <= "00000011";
wait for 100 ns;

B <= "00001111";
wait for 100 ns;

enable <='0';

wait; -- wait forever
```

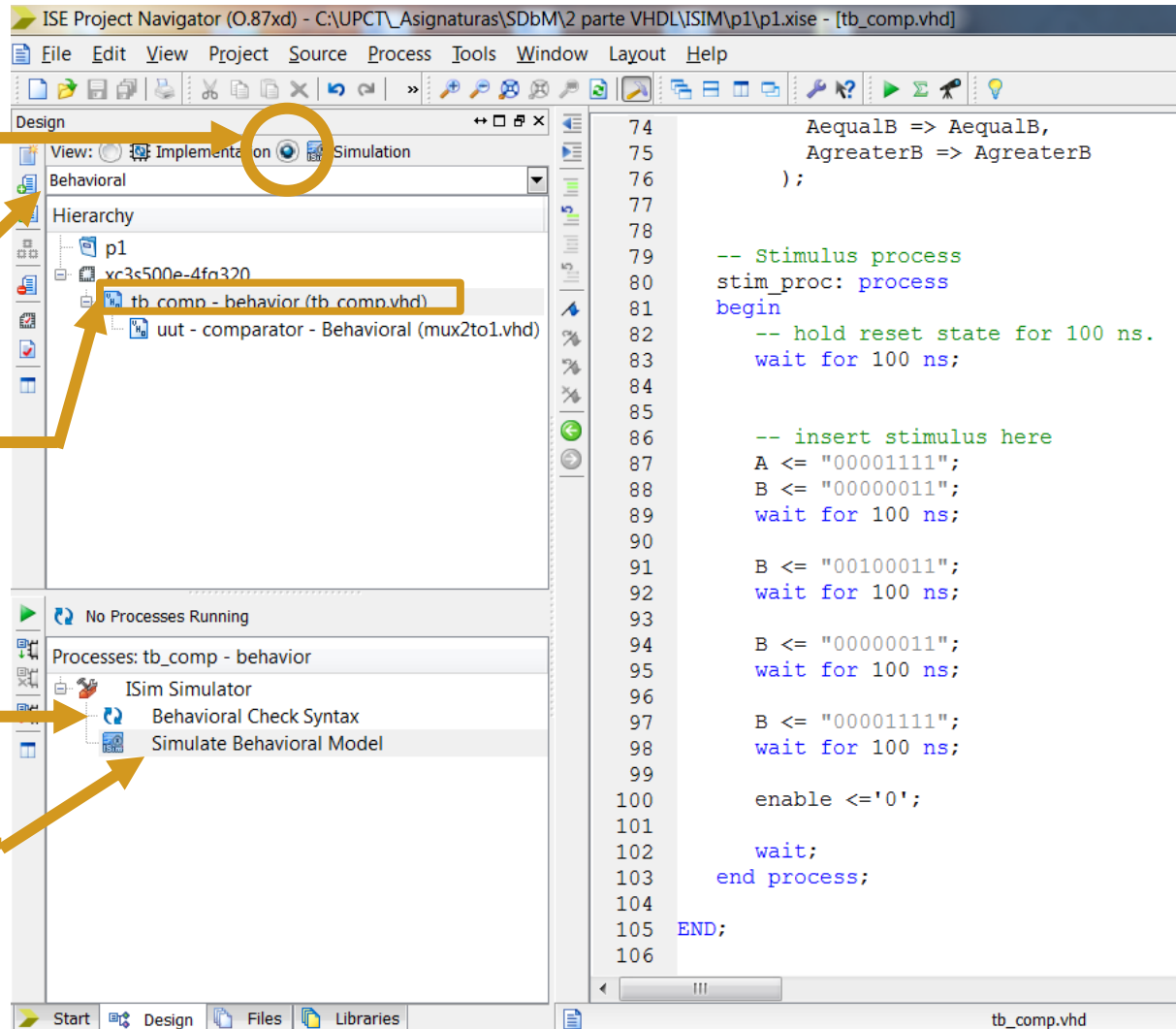


# Simulation with VHDL

## Test-bench of a combinational entity

### View of Xilinx ISE

- ☐ Option simulation must be selected
- ☐ Simulation type
- ☐ Test-bench file must be selected
- ☐ Syntax check
- ☐ Run behavioral simulation





# Simulation with VHDL

- ❑ Check syntax and then run the simulation
- ❑ *ISim* window will appear

The screenshot displays the ISim (O.87xd) - [Default.wcfg] window. The top menu bar includes File, Edit, View, Simulation, Window, Layout, and Help. Below the menu is a toolbar with various simulation controls. The main window is divided into several panes:

- Instances and Processes:** Shows the instance `tb_comp` and its associated process `std_logic_1164`.
- Objects:** A table titled "Simulation Objects for tb\_comp" listing object names and their values:

Object Name	Value
a[7:0]	00001111
b[7:0]	00001111
enable	0
alesb	0
aequalb	0
agreaterb	0
- Waveform:** A table showing signal values over time. The columns represent time points: 999,997 ps, 999,998 ps, 999,999 ps, and 1,000,000 ps. The rows correspond to the signals listed in the Objects pane:

	999,997 ps	999,998 ps	999,999 ps	1,000,000 ps
a[7:0]	00001111	00001111	00001111	00001111
b[7:0]	00001111	00001111	00001111	00001111
enable	0	0	0	0
alesb	0	0	0	0
aequalb	0	0	0	0
agreaterb	0	0	0	0

The bottom status bar indicates the simulation is running at 1.00us and shows the file name `Default.wcfg`. A console window at the bottom displays the message: "ISim O.87xd (signature 0xc3576ebc) This is a Full version of ISim."



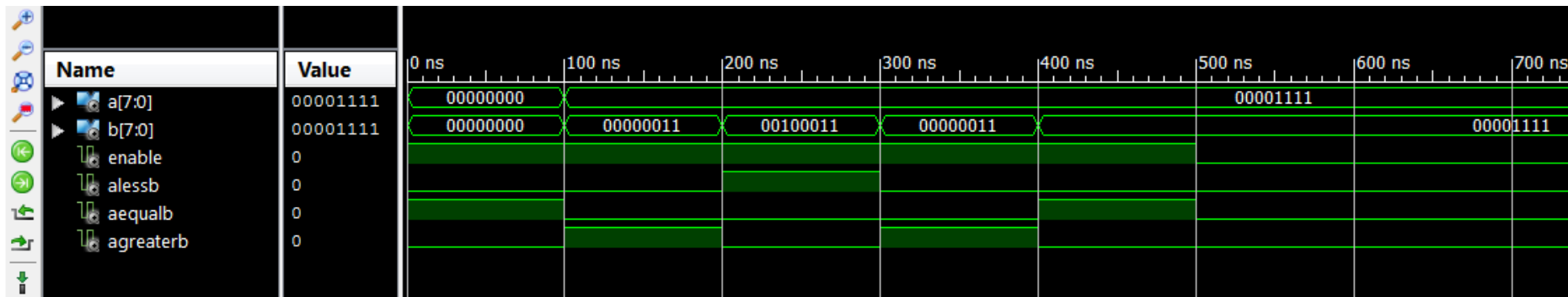
# Simulation with VHDL

## Test-bench of a combinational entity

- Click full zoom to see all the simulated time



- It can be seen how depending on the value of A and B, outputs change following the time sequence described in the test-bench



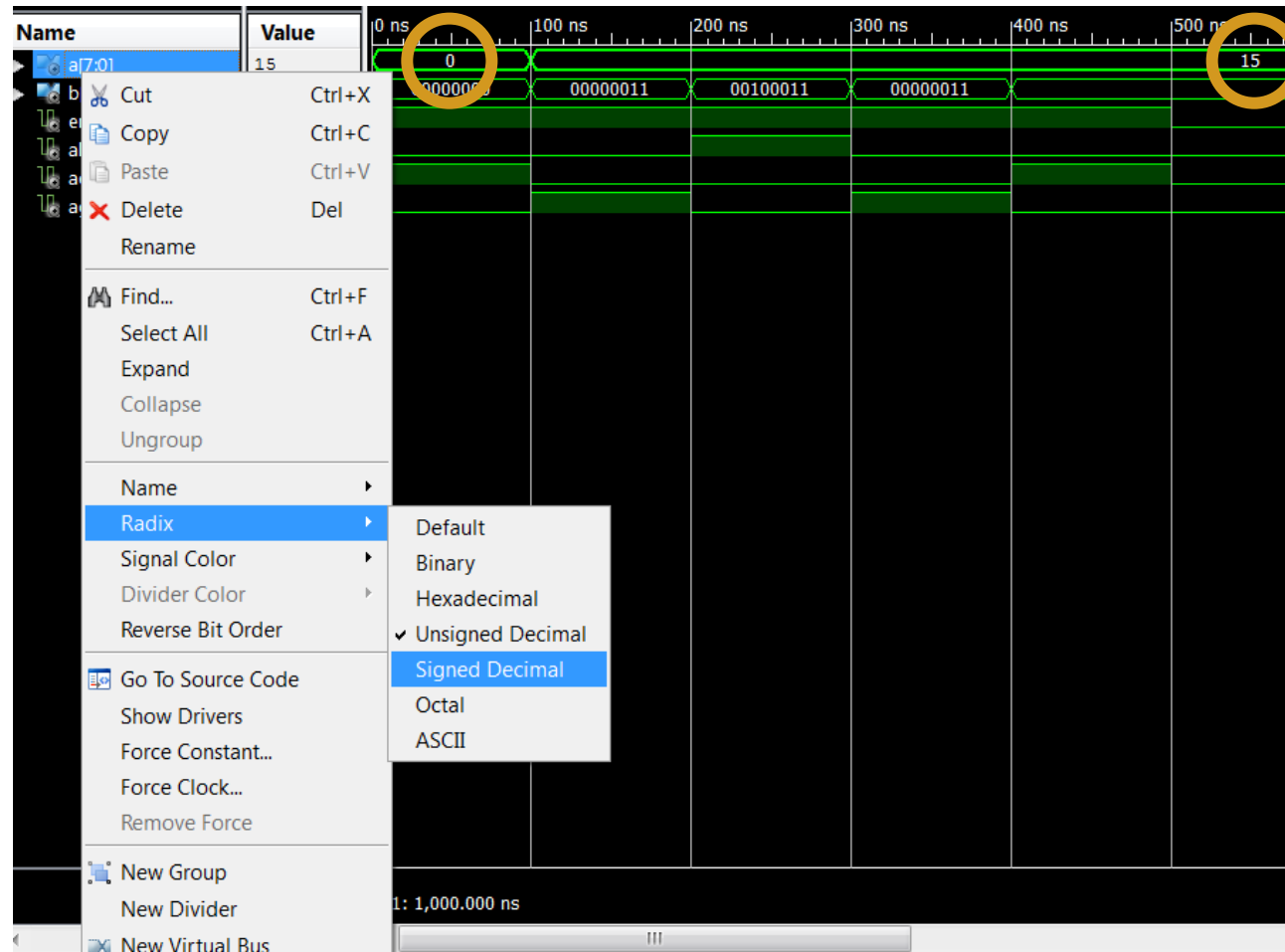
- Logical behavior of the entity is verified



# Simulation with VHDL

## Basics of the ISE Simulator

### □ Change radix of signal





# Simulation with VHDL

## Basics of the ISE Simulator

- ❑ If the duration of the simulated time is too short one can run extra time



- ❑ Go to previous transition (to see at what time it happened)



- ❑ Go to next transition



- ❑ Add marker (time stamp) at this instant of the simulation

