

CSU22022 Computer Architecture I

Ninth Lecture - Testbench

Michael Manzke

2023-2024

Trinity College Dublin

Simulation

OR_XXXXXXX_TB

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity OR_XXXXXXX_TB is -- we don't need ports
end OR_XXXXXXX_TB;
architecture Simulation of OR_XXXXXXX_TB is
    COMPONENT OR_XXXXXXX -- Component Declaration for the Unit Under Test (UUT)
        Port ( x : in STD_LOGIC;
              y : in STD_LOGIC;
              F : out STD_LOGIC);
    END COMPONENT;
    signal x_TB, y_TB : STD_LOGIC := '0'; -- Inputs Signals
    signal F_TB : STD_LOGIC := '0'; -- Output Signal
begin
    uut: OR_XXXXXXX PORT MAP ( -- Instantiate the Unit Under Test (UUT)
        x => x_TB,
        y => y_TB,
        F => F_TB
    );
    stim_proc: process
    begin
        x_TB <= '0'; y_TB <= '0'; -- case A
        wait for 20 ns;
        x_TB <= '0'; y_TB <= '1'; -- case B
        wait for 20 ns;
        x_TB <= '1'; y_TB <= '0'; -- case C
        wait for 20 ns;
        x_TB <= '1'; y_TB <= '1'; -- case D
        wait for 20 ns;
    end process;
end Simulation;
```

OR_XXXXXXX

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

entity OR_XXXXXXX is
    Port ( x : in STD_LOGIC;
          y : in STD_LOGIC;
          F : out STD_LOGIC);
end OR_XXXXXXX;

architecture Behavioral of OR_XXXXXXX is
begin
    F <= x or y after 7ns;
end Behavioral;
```

x_TB → x

y_TB → y

F → F_TB

Figure 1: The Unit Under Test (UUT) OR_XXXXXXX and the testbench OR_XXXXXXX_TB

Design Entity - OR_XXXXXXXX

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity OR_XXXXXXXX is
5     Port ( x : in STD_LOGIC;
6           y : in STD_LOGIC;
7           F : out STD_LOGIC);
8 end OR_XXXXXXXX;
9
10 architecture Behavioral of OR_XXXXXXXX is
11
12 begin
13
14     F <= x or y after 7ns;
15
16 end Behavioral;
```

Listing 1: The design entity describes the hardware

Testbench Entity - OR_XXXXXXXX_TB - One

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity OR_XXXXXXXX_TB is
5 -- we don't need ports
6 end OR_XXXXXXXX_TB;
7
8 architecture Simulation of OR_XXXXXXXX_TB is
9
10     -- Component Declaration for the Unit Under Test (UUT)
11     COMPONENT OR_XXXXXXXX
12     Port ( x : in STD_LOGIC;
13           y : in STD_LOGIC;
14           F : out STD_LOGIC);
15     END COMPONENT;
```

Listing 2: The testbench **entity** instantiates the UUT and generates into to the UUT.

Testbench Entity - OR_XXXXXXXX_TB - Two

```
1  --Inputs Signals
2  signal x_TB : STD_LOGIC := '0';
3  signal y_TB : STD_LOGIC := '0';
4  --Output Signal
5  signal F_TB : STD_LOGIC := '0';
6  -- StudentID e.g. 26 33 57 25(DEC) = 1 91 D9 ED(HEX)
7  constant StudentID : STD_LOGIC_VECTOR (27 downto 0) := x"
   191D9ED";
8  begin
9  -- Instantiate the Unit Under Test (UUT)
10  uut: OR_XXXXXXXX PORT MAP (
11      x => x_TB,
12      y => y_TB,
13      F => F_TB
14  );
```

Listing 3: The testbench entity instantiates the UUT and generates into to the UUT.

Testbench Entity - OR_XXXXXXXX_TB - Three

```
1
2 stim_proc: process
3   begin
4     x_TB <= '0'; y_TB <= '0'; -- case A
5     wait for 20 ns;
6     x_TB <= '0'; y_TB <= '1'; -- case B
7     wait for 20 ns;
8     x_TB <= '1'; y_TB <= '0'; -- case C
9     wait for 20 ns;
10    x_TB <= '1'; y_TB <= '1'; -- case D
11    wait for 20 ns;
12  end process;
13 end Simulation;
```

Listing 4: The testbench entity instantiates the UUT and generates into to the UUT.

- We require a testbench for every entity
- These entities may have instantiated other entities.
 - Instantiated entities will have their own testbench entities
- A testbench allows us to generate input to the design entity (UUT)
- We can change the input to the UUT over time

Testbench Entity - Ports

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity OR_XXXXXXXX_TB is
5 -- we don't need ports
6 end OR_XXXXXXXX_TB;
```

Listing 5: The testbench entity has no ports

Testbench Entity - Component Declaration

```
1  -- Component Declaration for the Unit Under Test (UUT)
2  COMPONENT OR_XXXXXXX
3  Port ( x : in STD_LOGIC;
4         y : in STD_LOGIC;
5         F : out STD_LOGIC);
6  END COMPONENT;
```

Listing 6: The testbench entity must declare the component under test in its architecture

The component name and the component ports are those of the Unit Under Test (UUT)

Testbench Entity - signal Declaration

```
1  --Inputs  Signals
2  signal x_TB : STD_LOGIC := '0';
3  signal y_TB : STD_LOGIC := '0';
4  --Output  Signal
5  signal F_TB : STD_LOGIC := '0';
6
7  -- StudentID e.g. 26 33 57 25(DEC) = 1 91 D9 ED(HEX)
8  constant StudentID : STD_LOGIC_VECTOR (27 downto 0) := x"
    191D9ED";
```

Listing 7: The testbench **entity** must declare input and out **signals** for all the ports of the UUT in its **architecture**

For this coursework, we also define a constant with our studentID. This constant must be shown simulation output.

Testbench Entity - Instantiate the Unit Under Test (UUT)

```
1 begin
2   -- Instantiate the Unit Under Test (UUT)
3   uut: OR_XXXXXXXX PORT MAP (
4       x => x_TB,
5       y => y_TB,
6       F => F_TB
7   );
```

Listing 8: In the **architecture** after the **begin** we instantiate the previously declared component and connect the simulation **signals** to it.

Testbench Entity - stim_proc: process

```
1  stim_proc: process
2      begin
3          x_TB <= '0'; y_TB <= '0'; -- case A
4          wait for 20 ns;
5          x_TB <= '0'; y_TB <= '1'; -- case B
6          wait for 20 ns;
7          x_TB <= '1'; y_TB <= '0'; -- case C
8          wait for 20 ns;
9          x_TB <= '1'; y_TB <= '1'; -- case D
10         wait for 20 ns;
11     end process;
```

Listing 9: The stim_proc: process runs all the time and generates input signals for the Unit Under Test (UUT)

Testbench Entity - For Clocked Designs

```
1  Clk_TB <= not CLK_TB after PERIOD/2;
2  stim_proc: process
3  begin
4      wait until CLK_TB'event and CLK_TB='1';
5      D_TB <= '0' after PERIOD/4;           -- Case A
6      wait until CLK_TB'event and CLK_TB='1';
7      D_TB <= '0' after PERIOD/4;           -- Case B
8      wait until CLK_TB'event and CLK_TB='1';
9      D_TB <= '1' after PERIOD/4;           -- Case C
10     wait until CLK_TB'event and CLK_TB='1';
11     D_TB <= '0' after PERIOD/4;           -- Case D
12     wait until CLK_TB'event and CLK_TB='1';
13     D_TB <= '1' after PERIOD/4;           -- Case E
14     wait until CLK_TB'event and CLK_TB='1';
15 end process;
16 end Simulation;
```

Listing 10: Simulation for a clocked design