Advanced Digital Circuit Design - Introduction to VHDL

Prof. Dr. Berna Örs Yalçın

Istanbul Technical University
Faculty of Electrical and Electronics Engineering
Department of Electronics and Communication Engineering
siddika.ors@itu.edu.tr

Outline

- VHDL Basics
- Simulation with Test Benches

VHDL Facts

- There are about 100 keywords (predefined lowercase identifiers) in VHDL, e.g. **entity**, **port**, **architecture**, **process**, **signal**, **and**, **or**, **not**, etc.
- -- is used for single-line comments
- Blank spaces are ignored but are not allowed in keywords, user-specified identifiers, operators or number representations.
- VHDL is case sensitive, e.g. **not** is not the same as NOT

VHDL Entity Declaration

- In VHDL, an entity is a fundamental descriptive unit and the term *entity* refers to the text enclosed by the keyword pair **entity** and **end**.
- The keyword **entity** is followed by a name (for identifying the entity) **is** and a list of ports
- Identifiers are composed of alphanumeric characters and the underscore (_) and are case-sensitive. They should start with an aphabetic character or underscore.
- A entity's port list is the interface between its environment and itself.
- The keywords **in** and **out** are used to denote which of the ports are inputs and which are outputs.

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  entity simple_circuit is
4      Port ( D : out STD_LOGIC;
5             E : inout STD_LOGIC;
6             A, B, C : in STD_LOGIC);
7  end simple_circuit;
8
9  architecture Behavioral of simple_circuit is
10 signal w1 : STD_LOGIC;
11 component and_gate
12   port (A, B: in STD_LOGIC;
13         C: out STD_LOGIC);
14 end component;
15 component or_gate
16   port (A, B: in STD_LOGIC;
17         C: out STD_LOGIC);
18 end component;
19 component not_gate
20   port (A: in STD_LOGIC;
21         C: out STD_LOGIC);
22 end component;
23 begin
24   G1: and_gate port map(A,B,w1);
25   G2: not_gate port map (C,E);
26   G3: or_gate  port map (E,w1,D);
27 end Behavioral;
```

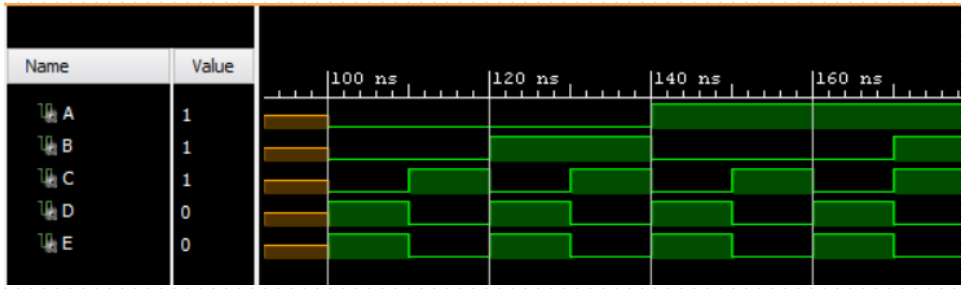
VHDL Entity Declaration

- Internal connections in a circuit are declared as signals using the keyword **signal**.
- Here a list of predefined gates (described with the keywords **and_gate**, **not_gate**, **or_gate**) are used. Each one is an instantiation of the gate and is called a gate instance.
- Each gate instantiation has an optional name (e.g. G1, G2, G3)
- Gate output and inputs are listed in parantheses and separated by commas.
- Note the difference between entity declaration and entity instantiation. The entity simple_circuit is declared here, but the primitive gates **and_gate**, **not_gate** and **or_gate** are instantiated

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  entity simple_circuit is
4      Port ( D : out STD_LOGIC;
5             E : inout STD_LOGIC;
6             A, B, C : in STD_LOGIC);
7  end simple_circuit;
8
9  architecture Behavioral of simple_circuit is
10 signal w1 : STD_LOGIC;
11 component and_gate
12   port (A, B: in STD_LOGIC;
13         C: out STD_LOGIC);
14 end component;
15 component or_gate
16   port (A, B: in STD_LOGIC;
17         C: out STD_LOGIC);
18 end component;
19 component not_gate
20   port (A: in STD_LOGIC;
21         C: out STD_LOGIC);
22 end component;
23 begin
24   G1: and_gate port map(A,B,w1);
25   G2: not_gate port map (C,E);
26   G3: or_gate  port map (E,w1,D);
27 end Behavioral;
```

Simulation with a Test Bench

- The **process** keyword is used to start execution of a set of statements enclosed between the **begin** and **end process** keywords.
- If a statement is preceeded by a delay value (e.g., wait for 10 ns;), the simulator postpones executing the statement until the specified time has elapsed.




```
1   library IEEE;
2   use IEEE.STD_LOGIC_1164.ALL;
3   entity simple_circuit_tb is
4   end simple_circuit_tb;
5   architecture Behavioral of simple_circuit_tb is
6   component simple_circuit
7       Port ( D : out STD_LOGIC;
8              E : inout STD_LOGIC;
9              A, B, C : in STD_LOGIC);
10  end component;
11  signal A,B,C,D,E : STD_LOGIC;
12  begin
13  DUT: simple_circuit Port map(D,E,A,B,C);
14  tb: process
15  begin
16  wait for 100 ns;
17  A <= '0'; B <= '0'; C <= '0';
18  wait for 10 ns;
19  A <= '0'; B <= '0'; C <= '1';
20  wait for 10 ns;
21  A <= '0'; B <= '1'; C <= '0';
22  wait for 10 ns;
23  A <= '0'; B <= '1'; C <= '1';
24  wait for 10 ns;
25  A <= '1'; B <= '0'; C <= '0';
26  wait for 10 ns;
27  A <= '1'; B <= '0'; C <= '1';
28  wait;
29  end process;
30  end Behavioral;
```