Al-Nahrain university

College of Information Engineering

Cyber Security Department

DSD Lab.2

- Implementation of Basic Logic Gates using VHDL

- Signal Statement

By

Dr. Donya Abbas Khalid

Asst.Lecture. Noora Hani

Aim:

First this lab aims to learn how to write a simple VHDL code by steps, then write a code for all logic gates.

Theory:

VHDL is one of the commonly used Hardware Description Languages (HDL) in digital circuit design. VHDL stands for VHSIC Hardware Description Language. In turn, VHSIC stands for Very-High-Speed Integrated Circuit.

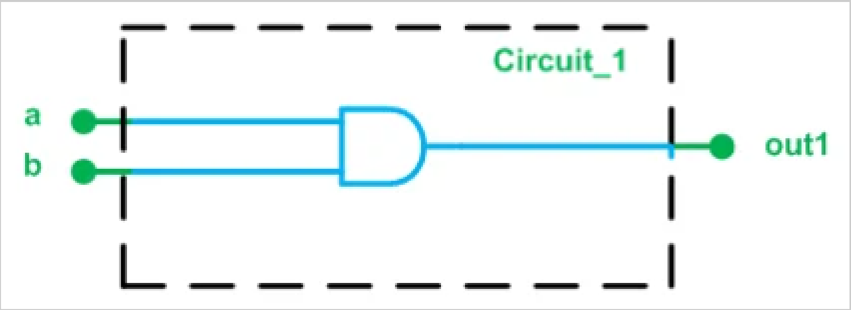


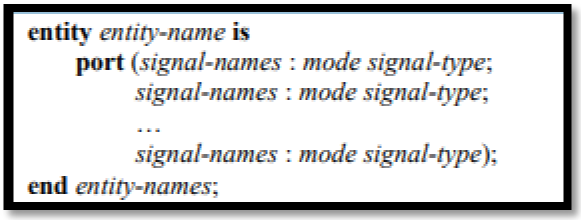
Figure 1: A simple digital circuit

This figure shows that there are two input ports, a and b, and one output port, out1.

A VHDL entity is simply a declaration of a module's inputs and outputs, i.e. its external interface signals or ports.

A VHDL architecture is a detailed description of the module's internal structure or behavior. You can think of the entity as a "wrapper" for the architecture, hiding the details of what's inside while providing the "hooks" for other modules to use it. VHDL actually allows you to define multiple architectures for a single entity, and it provides a configuration management facility that allows you to specify which one to use during a particular synthesis run.

A basic entity declaration has the syntax as shown below:



```
entity entity-name is
      port (signal-names : mode signal-type;
              signal-names : mode signal-type;
              . . .
              signal-names : mode signal-type);
end entity-names;
```

In addition to the keywords, an entity declaration has the following elements: **entity-name**: A user-selected identifier to name the entity.

signal-names: A comma-separated list of one or more user-selected identifiers to name external-interface signals.

mode: One of four reserved words, specifying the signal direction:

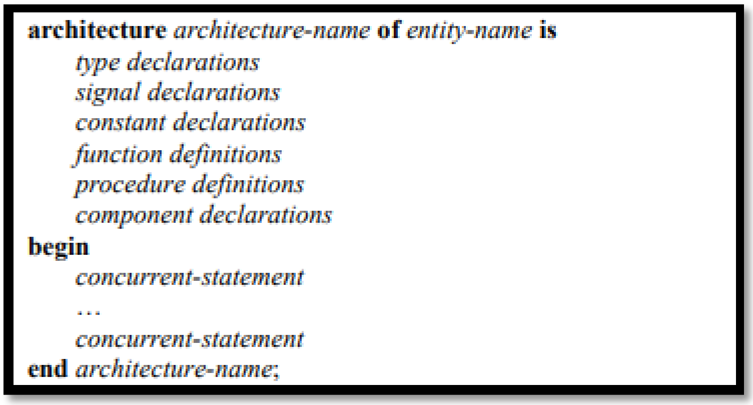
in – The signal is an input to the entity.

out – The signal is an output of the entity. Note that the value of such a signal cannot be "read" inside the entity's architecture, only by other entities that use it.

buffer – The signal is an output of the entity, and its value can also be read inside the entity's architecture.

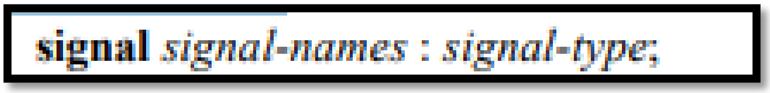
inout – The signal can be used as an input or an output of the entity.

A basic architecture definition has the syntax as shown below:



```
architecture architecture-name of entity-name is
      type declarations
      signal declarations
      constant declarations
      function definitions
      procedure definitions
      component declarations
begin
      concurrent-statement
      . . .
      concurrent-statement
end architecture-name;
```

Signal declaration.



```
signal signal-names : signal-type;
```

Procedure

Perform the following steps to implement a circuit corresponding to the code on the DE2-series board.

1. Create a new Quartus II project for your circuit. Select Cyclone IV EP4CE115F29C7 if using the DE2-115 board.

2. Create a VHDL entity for the code in Figure 3 and include it in your project.

3. Include in your project the required pin assignments for the DE2-series board, as discussed above. Compile the project.

4. Download the compiled circuit into the FPGA chip. Test the functionality of the circuit by toggling the switches and observing the LEDs.
5. Open ***Quartus*** II RTL ***Viewer (Tools→Netlist viewers→ RTL viewer)*** to display a schematic ***view*** of the design
6. By model Sim- Altera run a Functional Simulation to simulate your design based on a functional stand point.

Lab work:

1-Write VHDL code for XOR logic gate

Solution:

```
library ieee;
use ieee.std_logic_1164.all;
entity first is
port (a,b : in bit;
     x   : out bit);
end first;
```

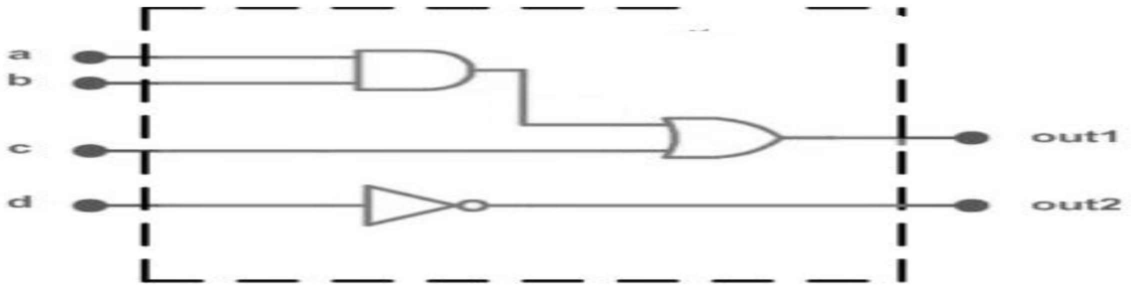
```
architecture simple of first is


 begin
 x<= a XOR b;
 end simple;
```

2-Write VHDL code for NOR logic gate

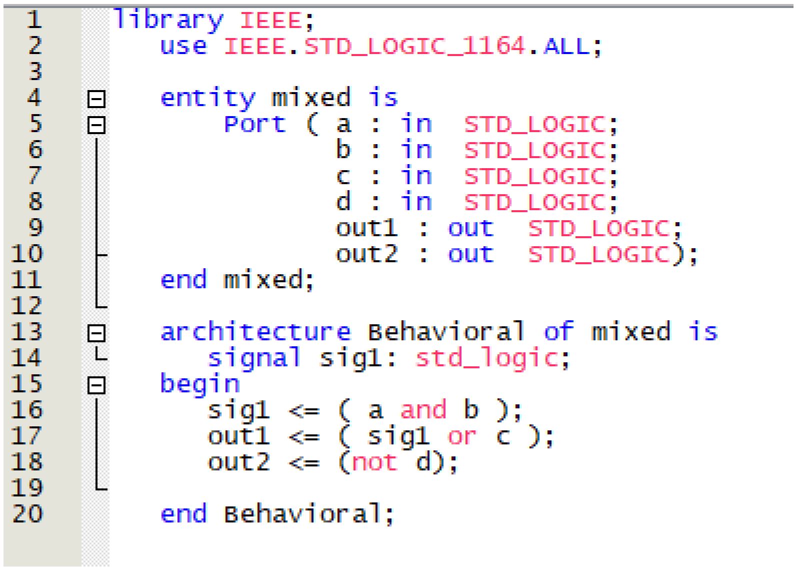
Solution:

```
library ieee;
use ieee.std_logic_1164.all;
entity first is
port (a,b : in bit;
      x   : out bit);
end first;
architecture simple of first is
 begin
 x<= a NOR  b;
 end simple;
```

3- Write the VHDL code for the circuit in Figure below



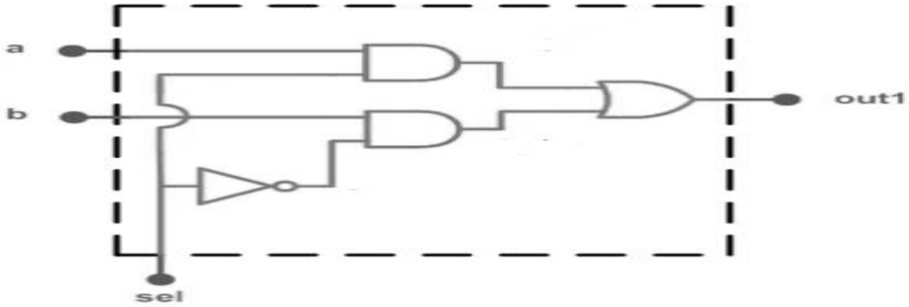
Solution:



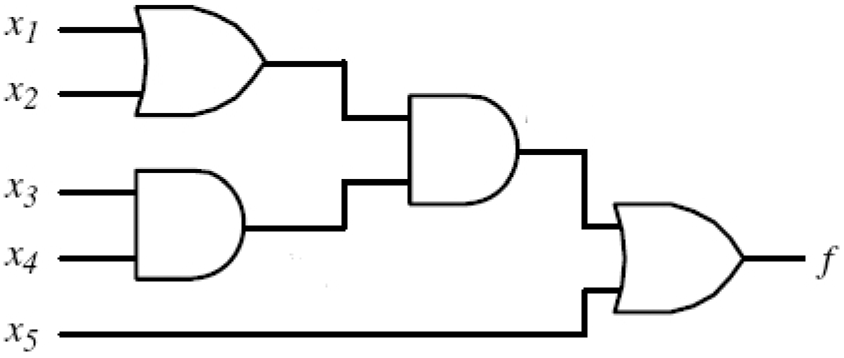
```
library IEEE;
   use IEEE.STD_LOGIC_1164.ALL;

   entity mixed is
       Port ( a : in  STD_LOGIC;
              b : in  STD_LOGIC;
              c : in  STD_LOGIC;
              d : in  STD_LOGIC;
              out1 : out  STD_LOGIC;
              out2 : out  STD_LOGIC);
   end mixed;

   architecture Behavioral of mixed is
      signal sig1: std_logic;
   begin
      sig1 <= ( a and b );
      out1 <= ( sig1 or c );
      out2 <= (not d);

   end Behavioral;
```

Home work:

1-Write a VHDL code for the Figure below and show its simulation result?



2- Write a VHDL code for the Figure below and show its simulation result?



3- Write a VHDL code to implement a 2-input XOR gate using SOP