电子科技大学
格拉斯哥学院
Glasgow College, UESTC

UESTC3020 – Digital Circuit Design

VHDL

LABORATORY

EXERCISE

BOOK

v 1.0

Frederic Surre **2024/25**

BEng in Electrical and Electronic Engineering (Communications Engineering)

SECTION 1: Use of Xillinx Vivavo Design environment

1- Introduction

In this first section, you will learn how to use Vivado Design environment to simulate combinational circuits.

In the first part, you are given the VHDL code and you will work on doing 'manual' simulations. Then, in the second part, you are given a circuit and you have to write two VHDL descriptions of this circuit

During this lab, you are expected to complete all tasks, write down detailed comments of your observations and practice.

The skills you are expected to acquire in this lab session will be critical for completion of the second part of your continuous assessment.

2- Preliminary remarks

- **Objectives:**
  - To learn how to use Xillinx Vivado Design suite.
  - To write VHDL programs to model digital circuits.
  - To perform 'manual' simulations of digital circuits.

- **Equipment required**
  - Xillinx Vivado Webpack 2018.3 or 2019.1

- **Videos & Documents (available on Moodle)**
  - How to install Xillinx Vivado.
  - Introduction to Xilinx Vivado.
  - **Video**: How to create a project in Xilinx Vivado.
  - **Video**: How to run a manual simulation in Xilling Vivado.

3- Tasks to be completed

PART I: Circuit I

In this part, you are given the VHDL code of a known circuit and you have to implement it in Xilinx Vivado. Then, you will simulate this circuit.
The circuit is the one from Tutorial #6 question 1.

Question 1:

Consider the following circuit:

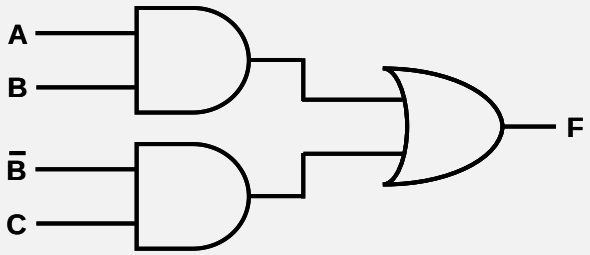

Figure 1-1: Combinational circuit to simulate in VHDL.

(a) Task 1

Start Vivado 2019.1 (or 2018.3 depending which version is available in the lab and/or which version you have installed on your own computer) and create a new project for the circuit on figure 1-1 (following the process presented in the video and extra documents available on Moodle). Make sure to include the correct inputs and output (see entity in code presented in figure 1-2).

(b) Task 2

Copy in your project the VHDL dataflow description code presented in figure 1-2.

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

entity Test2 is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : in STD_LOGIC;
           F : out STD_LOGIC);
end Test2;

architecture Behavioral of Test2 is
signal c1 : STD_LOGIC;
begin
    c1 <= not B;
    F <= (A and B) or (c1 and C);

end Behavioral;
```

Figure 1-2: Example of dataflow description of the circuit in fig 1-1.

(c) Task 3

Run a simulation of your VHDL file.

(d) Task 4

In the simulation window, restart the simulation and manually change the inputs following the sequence presented in table 1.1 (between each change you have to run the simulation for 200ns). Please refer to the video and extra documents on how to do this task in Vivado.

Table 1.1: Values of the three inputs, A, B, C as a function of time.

Variable	@t = 0s	@t = 200ns	@t = 400ns	@t = 600ns	@t = 800ns
A	0	1	1	1	0
B	0	0	1	1	1
C	0	1	1	0	0

(e) Task 5

Create a new project for the circuit on figure 1.1.

(f) Task 6

Copy in your project the structural description VHDL code presented in figure 1-3.

Be careful: for the structural description, you need to add the library UNISIM and the package VComponents. Generally, look carefully at the syntax of the code, especially how to reuse components.

(g) Task 7

Manually simulate this description using the input values for A, B and C presented in Table 1.1.

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

library UNISIM;
use UNISIM.VComponents.all;

entity test1 is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : in STD_LOGIC;
           F : out STD_LOGIC);
end test1;

architecture Structural of test1 is
signal S1, S2, S3 : STD_LOGIC;

component AND2
    port ( I0, I1 : in STD_LOGIC; O : out STD_LOGIC);
end component;

component OR2
    port ( I0, I1 : in STD_LOGIC; O : out STD_LOGIC);
end component;

component INV
    port ( I : in STD_LOGIC; O : out STD_LOGIC);
end component;

begin
    G1 : INV port map (B, S1);
    G2 : AND2 port map (A, B, S2);
    G3 : AND2 port map (S1, C, S3);
    G4 : OR2 port map (S2, S3, F);
end Structural;
```

Figure 1.3: Example of structural description of the circuit in fig 1-1.

PART II: Circuit 2

In this part, you are given a combinational circuit and you have to write two VHDL codes (using both data flow and structural descriptions). Then, you have to simulate the circuit to verify it is working as expected

Question 2:

This final part of the lab looks at using Boolean algebra to simplify Boolean expression and logic circuit.

Consider the circuit on fig. 1.4.

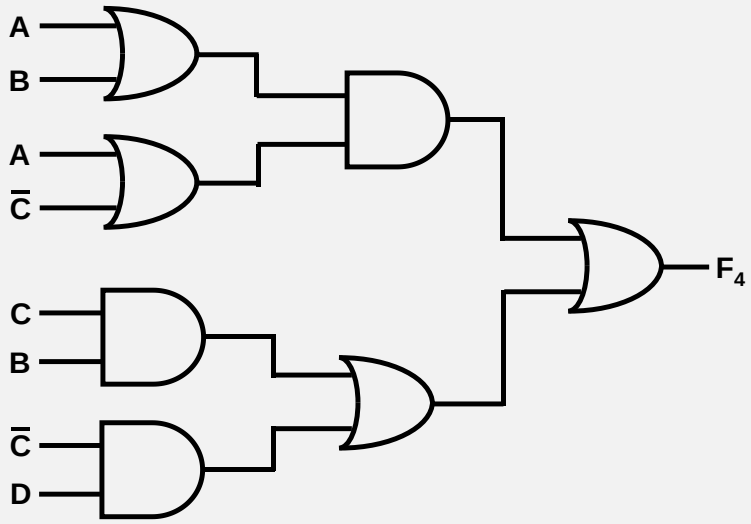

Figure 2.1: Second circuit to simulate.

(a) Task 8

Create a new project and identify the inputs and outputs of the circuit presented in figure 2.1.

(b) Task 9

From the circuit, find the Boolean expression for F4. This will be useful to check the result of your simulation in a later task.

(c) Task 10

Write a dataflow description of the circuit in figure 2.1.

(d) Task 11

'Manually' simulate your dataflow description and check that the truth table of the circuit is correct (Based on the Boolean function you have found in Task 9)

(e) Task 12

Create a new project for the circuit on figure 2.1 and write a structural description of the circuit.

(f) Task 13

Simulate the structural description from Task 12 and verify you found the same results as in Task 11 and Task 9

Optional Question:

Consider the 2-bit circuit you are designing in part one of the coursework (please, check the coursework 1 presentation document on Moodle if you are not sure which circuit your group has been asked to design). Create a new project and write both a dataflow and structural descriptions of one implementation of your design.

Simulate both descriptions and verify the working of your 2-bit circuit (either adder or subtractor as per the circuit you have been asked to design).

END OF LAB #3