VHDL Assignment 2

1- Names

Loïc Duchesne - 261052490
Yassine Mimet - 260980175

2- Executive summary

In this VHDL lab we first drew a schematic for the AeqB using the Block Diagram/Schematic file in Quartus which was just 4 XOR gates of input 2 regrouped by an AND gate of input 4. Then we converted the schematic to a VHDL description that we will use after in the simulation and we ran two simulations one with Testbench File using the Test Bench Writer in Quartus and the other one was an exhaustive test with all the cases with nested loops. In the second part of the lab, we simulated the 2-to-1 MUX so we wrote 2 VHDL files and 2 testbenches (you can find that in detail in question 1).

3- Questions:

(1) For the MUX: We wrote two VHDL codes for the structural and the behavioral description of the 2-to-1 multiplexer. And we also wrote 2 other testbench files to perform an exhaustive test for the VHDL description of the 2-to-1 multiplexer.

For the **structural** we started with the entity and we just declared our inputs(A, B and S) and our output Y, which were declared as 1 bit variables. Afterwards, in the architecture we wrote the structural function using this same set of inputs/output.

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity loic_duchesne_MUX_structural is
5      Port (A : in std_logic;
6            B : in std_logic;
7            S : in std_logic;
8            Y : out std_logic);
9  end loic_duchesne_MUX_structural;
10
11 architecture STRUCT of loic_duchesne_MUX_structural is
12 begin
13     Y <= (B and S) or ((not S) and A);
14 end STRUCT;
```

For the **behavioral** we also started with the entity and did the same thing for assigning the ports to their inputs/output. In the architecture our selection of the output depended on S so when S is 0 we assigned A to Y and when S is ‘1’ we assigned B to Y.

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity loic_duchesne_MUX_behavioral is
5      Port (A : in std_logic;
6            B : in std_logic;
7            S : in std_logic;
8            Y : out std_logic);
9  end loic_duchesne_MUX_behavioral;
10
11 architecture BEHAV of loic_duchesne_MUX_behavioral is
12 begin
13     with S select
14         Y <= A when '0',
15              B when '1',
16              'X' when others;
17 end BEHAV;
```

For the **testbench** files we had the same content in both of them except for the names being changed from structural to behavioral. We start by defining our 4 signals. Afterwards we assign the 4 ports to the MUX component. Then, we have the Begin statement where we assign the signals with their respective port. Finally, for the test we made 3 loops that iterates over all the input options for each ports.

```
27  LIBRARY ieee;
28  USE ieee.std_logic_1164.all;
29  USE ieee.numeric_std.all;
30
31 ENTITY loic_duchesne_MUX_testbench IS
32 END loic_duchesne_MUX_testbench;
33 ARCHITECTURE loic_duchesne_MUX_structural_arch OF loic_duchesne_MUX_testbench IS
34 -- constants
35 -- signals
36  SIGNAL A : std_logic;
37  SIGNAL B : std_logic;
38  SIGNAL S : std_logic;
39  SIGNAL Y : std_logic;
40 COMPONENT loic_duchesne_MUX_structural
41    PORT (
42       A : IN std_logic;
43       B : IN std_logic;
44       S : IN std_logic;
45       Y : OUT std_logic
46    );
47 END COMPONENT;
48 BEGIN
49    i1 : loic_duchesne_MUX_structural
50    PORT MAP (
51 -- list connections between master ports and signals
52       A => A,
53       B => B,
54       S => S,
55       Y => Y
56    );
57 generate_test: PROCESS
58 BEGIN
59    FOR i IN std_logic range '0' to '1' LOOP
60       A <= i;
61       FOR j IN std_logic range '0' to '1' LOOP
62          B <= j;
63             FOR k IN std_logic range '0' to '1' LOOP
64             S <= k;
65             WAIT FOR 10 ns;
66    END LOOP;
67    END LOOP;
68    END LOOP;
69    WAIT;
70 END PROCESS generate_test;
71 END loic_duchesne_MUX_structural_arch;
```

```
27  LIBRARY ieee;
28  USE ieee.std_logic_1164.all;
29  USE ieee.numeric_std.all;
30
31 ENTITY loic_duchesne_MUX_testbench IS
32 END loic_duchesne_MUX_testbench;
33 ARCHITECTURE loic_duchesne_MUX_behavioral_arch OF loic_duchesne_MUX_testbench IS
34 -- constants
35 -- signals
36  SIGNAL A : std_logic;
37  SIGNAL B : std_logic;
38  SIGNAL S : std_logic;
39  SIGNAL Y : std_logic;
40 COMPONENT loic_duchesne_MUX_behavioral
41    PORT (
42       A : IN std_logic;
43       B : IN std_logic;
44       S : IN std_logic;
45       Y : OUT std_logic
46    );
47 END COMPONENT;
48 BEGIN
49    i1 : loic_duchesne_MUX_behavioral
50    PORT MAP (
51 -- list connections between master ports and signals
52       A => A,
53       B => B,
54       S => S,
55       Y => Y
56    );
57 generate_test: PROCESS
58 BEGIN
59    FOR i IN std_logic range '0' to '1' LOOP
60       A <= i;
61       FOR j IN std_logic range '0' to '1' LOOP
62          B <= j;
63             FOR k IN std_logic range '0' to '1' LOOP
64             S <= k;
65             WAIT FOR 10 ns;
66    END LOOP;
67    END LOOP;
68    END LOOP;
69    WAIT;
70 END PROCESS generate_test;
71 END loic_duchesne_MUX_behavioral_arch;
```

(2)

	AeqB	2-to-1 MUX	
	Schematic	Structural	Behavioral
Logic Utilization (in ALMs)	2/32070 (< 1%)	1/32070 (< 1%)	1/32070 (< 1%)
Total pins	9/457 (2%)	4/457 (< 1%)	4/457 (< 1%)

Flow Summary

Flow Status	Successful - Wed Sep 21 17:39:38 2022
Quartus Prime Version	18.0.0 Build 614 04/24/2018 SJ Lite Edition
Revision Name	loic_duchesne_vhdl1
Top-level Entity Name	loic_duchesne_comp
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	2 / 32,070 (< 1 %)
Total registers	0
Total pins	9 / 457 (2 %)
Total virtual pins	0
Total block memory bits	0 / 4,065,280 (0 %)
Total DSP Blocks	0 / 87 (0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 4 (0 %)

Flow Summary

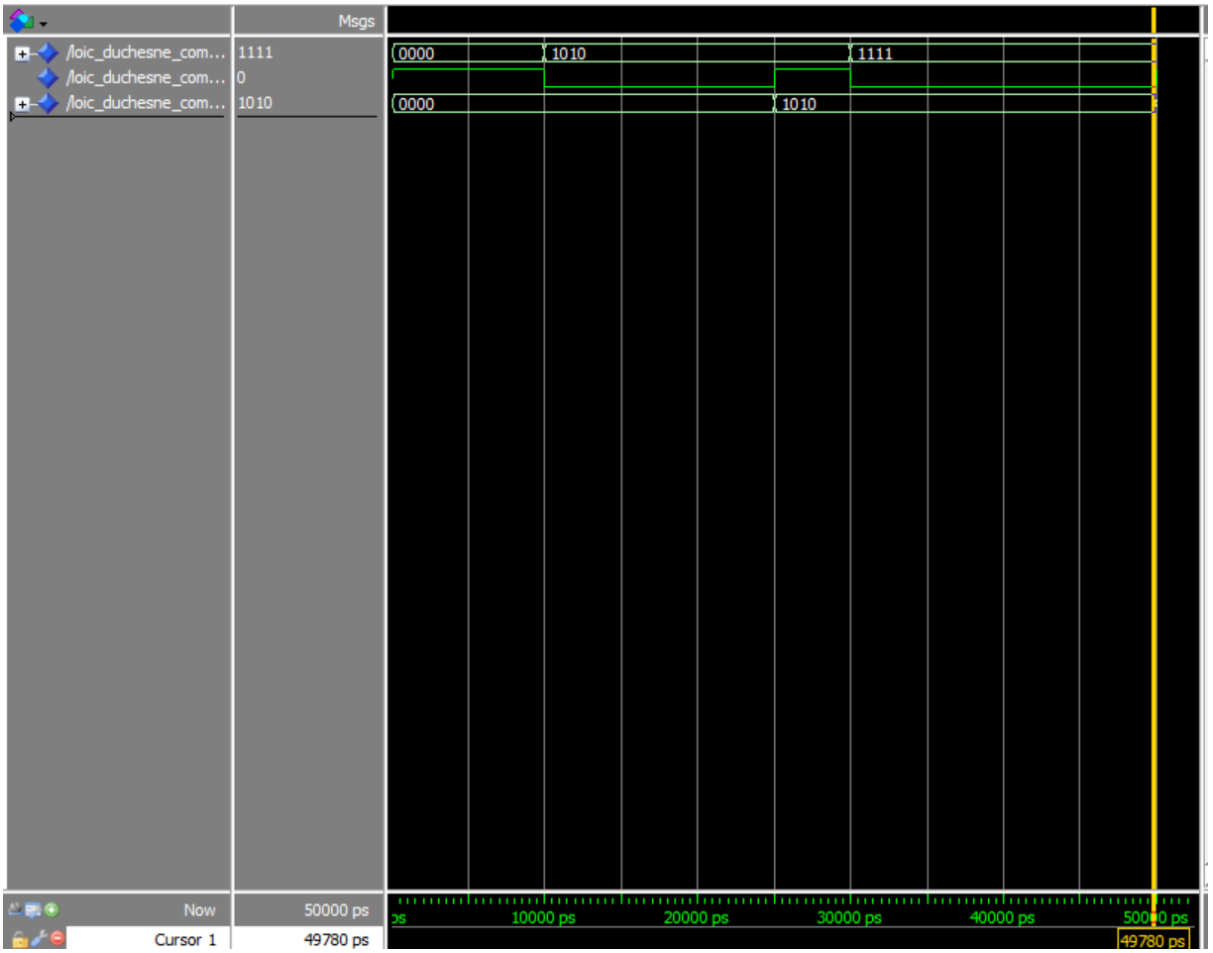
Flow Status	Successful - Wed Sep 21 17:17:48 2022
Quartus Prime Version	18.0.0 Build 614 04/24/2018 SJ Lite Edition
Revision Name	loic_duchesne_MUX
Top-level Entity Name	loic_duchesne_MUX_structural
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	1 / 32,070 (< 1 %)
Total registers	0
Total pins	4 / 457 (< 1 %)
Total virtual pins	0
Total block memory bits	0 / 4,065,280 (0 %)
Total DSP Blocks	0 / 87 (0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 4 (0 %)

Flow Summary

Flow Status	Successful - Wed Sep 21 17:20:56 2022
Quartus Prime Version	18.0.0 Build 614 04/24/2018 SJ Lite Edition
Revision Name	loic_duchesne_MUX
Top-level Entity Name	loic_duchesne_MUX_behavioral
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	1 / 32,070 (< 1 %)
Total registers	0
Total pins	4 / 457 (< 1 %)
Total virtual pins	0
Total block memory bits	0 / 4,065,280 (0 %)
Total DSP Blocks	0 / 87 (0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 4 (0 %)

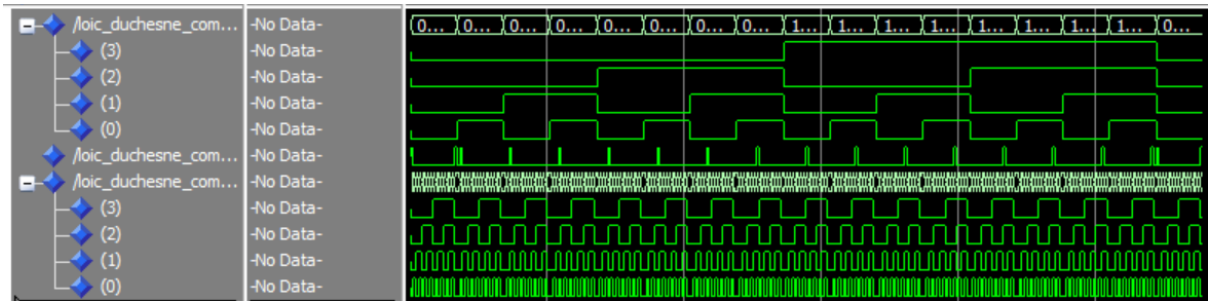
(3)

Introductory testing waveform results:



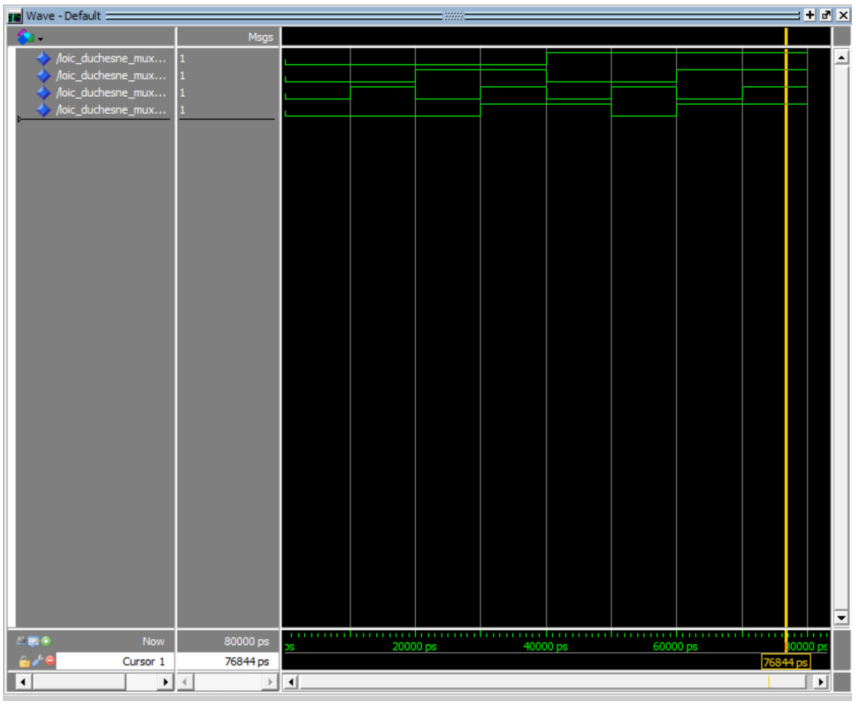
(4)

Exhaustive tests waveform results:

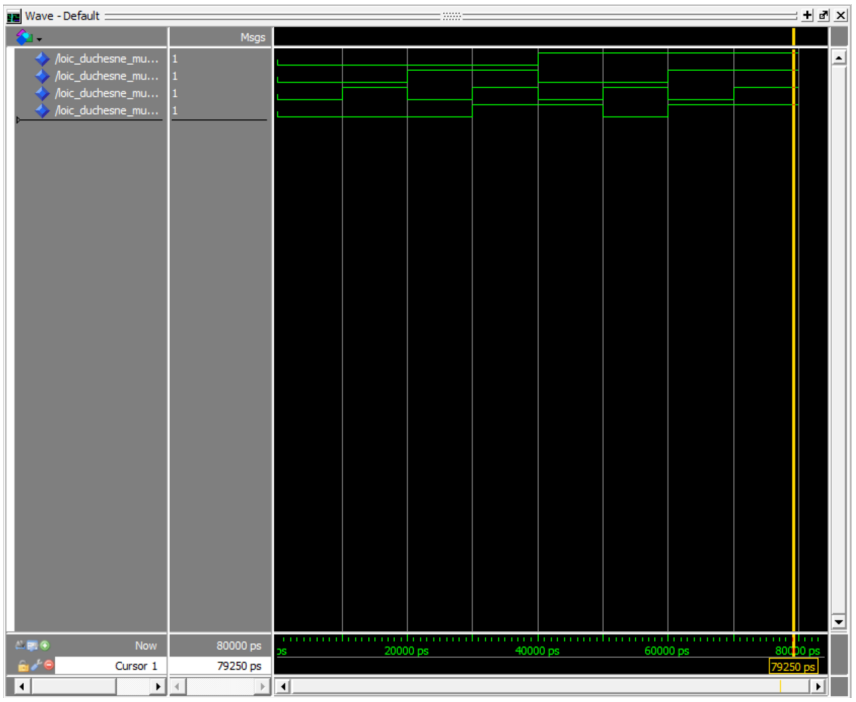


(5)

Structural waveform results:

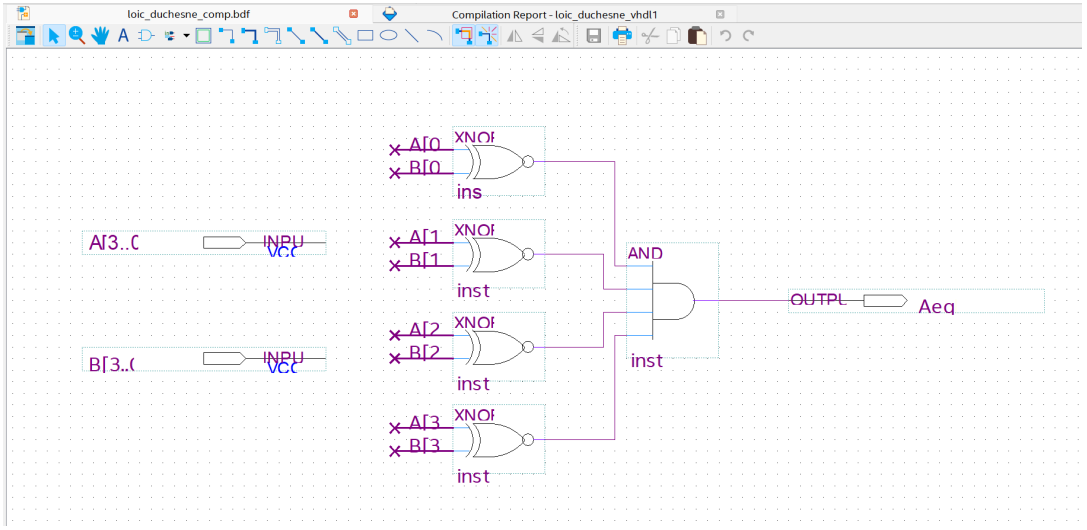


Behavioral waveform results:



4- Pictures (See part 3 for pictures of the results)

Schematic of AeqB:



5- Explanation of the results

For the first graph (*See Introductory testing waveform results p.3*) that we got the actual output after assigning A and B some of the values(not all of them because they are for bits) so we only have a 3 intervals for B because in the testbench file they didn't add a wait time after setting the value of B to 1111.
We got an output of '1' when A is '0000' and B is also '0000' also when A is '1010' and B is also '1010' is the two other cases we got an output value of '0'.

For the exhaustive (*See Exhaustive tests waveform results p.3*) we test for all the values of the inputs (A, B) using for loops so basically we have all the possible cases shown.

For the MUX (*See Structural waveform results & Behavioral waveform results p.4*), both behavioral and structural are the same in terms of their results. While their architectures are different, their underlying boolean expression remains the same. They are just describing the conditions of the function differently. In the results shown, we have 8 combinations of inputs and outputs. Our ultimate objective was to design a 2-to-1 MUX, where Y should be the same as input A if S is '0' or it should be B if S is '1'. While inspecting those exact conditions in the resulting waveform, we can clearly see that the MUX implementation was successful.

6- Conclusions

In conclusion, this lab allowed us to understand much more in depth how the VHDL design process works. Initially, we thought we had to compile everything through Quartus to run it on ModelSim, but quickly discovered that the testbench language we used was not actually compilable by Quartus. With the help of the the lab manual, we mostly used Quartus to actually write the files/design diagrams. We were also introduced to a new tool to convert diagrams to VHDL files. Afterwards, we opened ModelSim independently and built a workspace there to run our simulations.

In the end, we got the opportunity to familiarize with new workflows and techniques, which will be of great use in future VHDL projects.