09/03/2024

IFT-1227

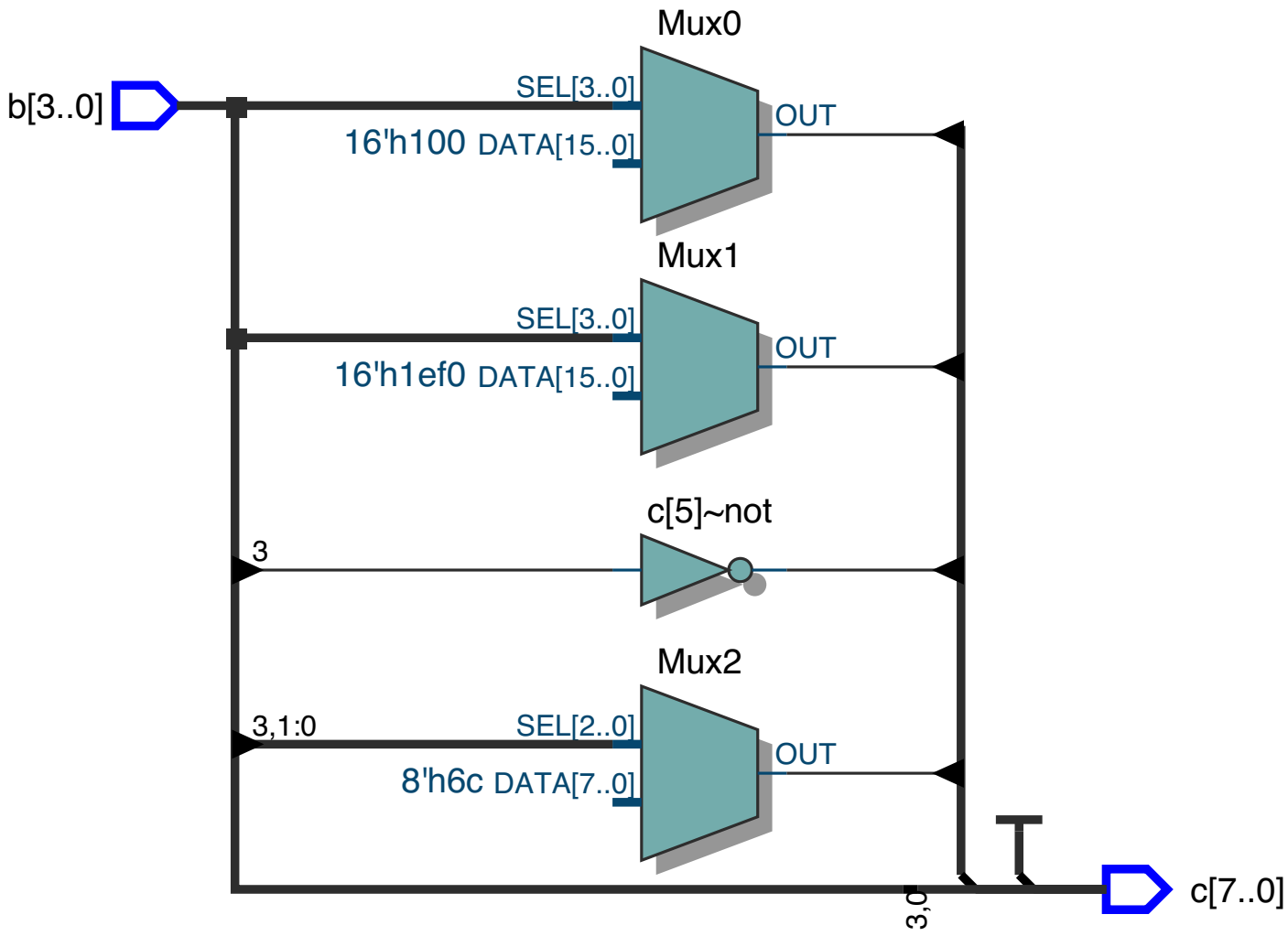
Rapport du devoir 2

Tarik Hireche 202 301 89

Ilyesse Bouzammita 202 761 43

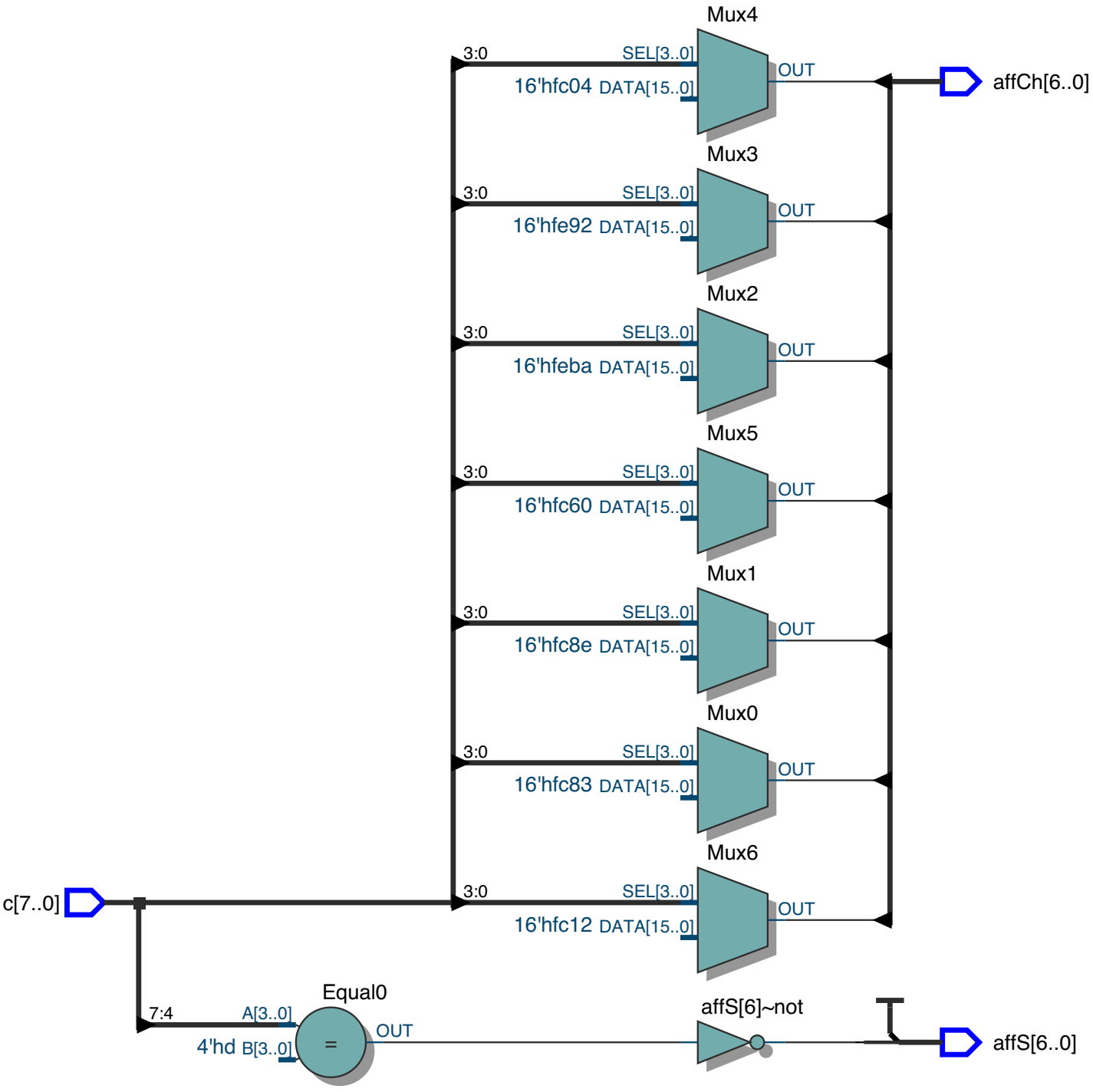
1. a) Premier circuit

Modélisation comportementale: Convertisseur C à 2 vers DCB



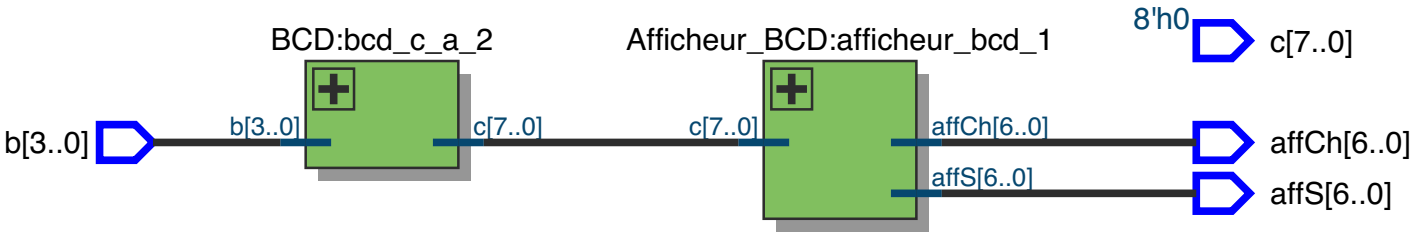
1. b) Deuxième circuit:

Module comportemental: Afficheur DCB

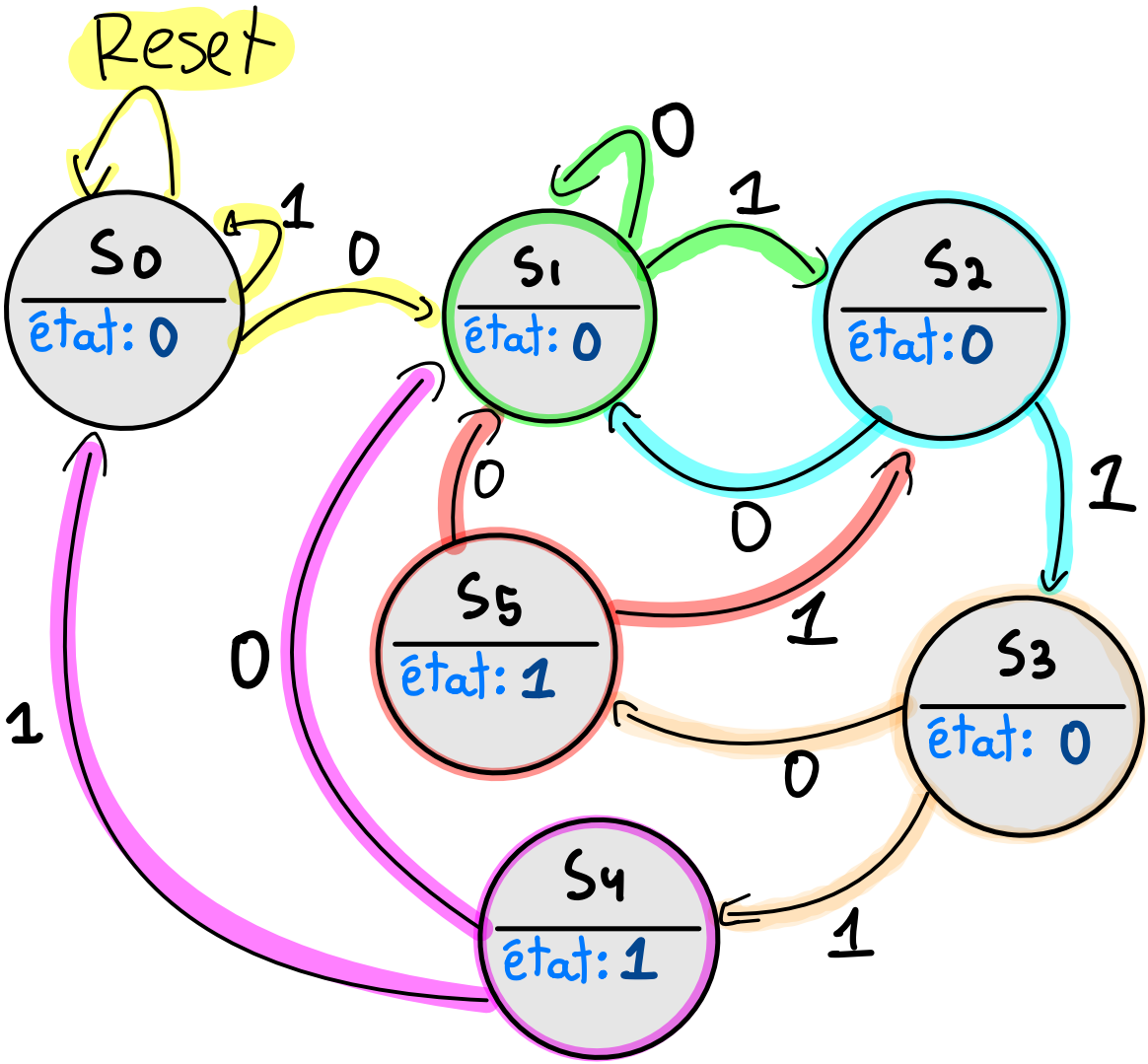


1.c) Troisième circuit:

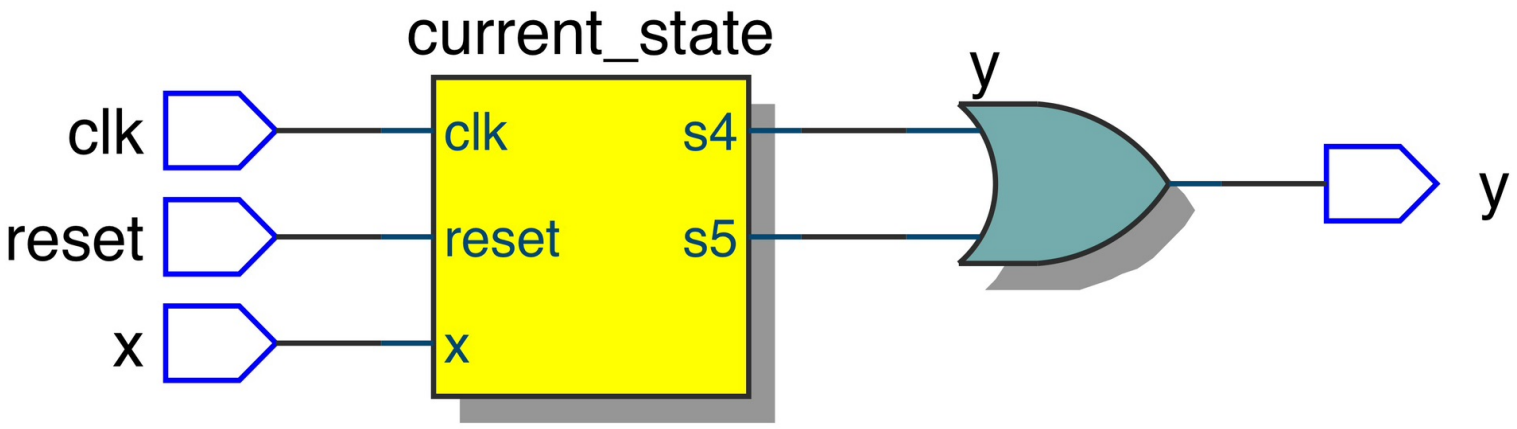
Modélisation structurelle du troisième circuit en utilisant les circuits de 1. a) et 1. b):



2. a) Diagramme de transition d'états:



2. b) Image du circuit synthétisé (MooreFSM):



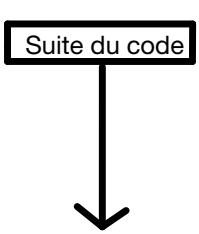
Capture d'écran TestBench:

Transcript

```
# Loading project Devoir2_sim
# Compile of MooreFSM_tb.vhd was successful.
ModelSim> vsim work.moorefsm_tb
# vsim work.moorefsm_tb
# Start time: 13:59:12 on Mar 12,2024
# Loading std.standard
# Loading std.textio(body)
# Loading ieee.std_logic_1164(body)
# Loading work.moorefsm_tb(tb_architecture)
# Loading work.moorefsm(fast)
add wave -position insertpoint sim:/moorefsm_tb/*
VSIM 3> run
# ** Note: All tests passed
#    Time: 130 ns  Iteration: 0  Instance: /moorefsm_tb

VSIM 4>
```

Code testbench:

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

entity MooreFSM_tb is
end entity;

architecture tb_architecture of MooreFSM_tb is
    component MooreFSM
        port(clk, reset: in STD_LOGIC;
             x: in STD_LOGIC;
             y: out STD_LOGIC);
    end component;

    signal clk, reset, x, y: STD_LOGIC;

begin
    -- Instantiating the MooreFSM component along with its inputs
and outputs
    dut: MooreFSM port map (
        clk => clk,
        reset => reset,
        x => x,
        y => y);

    -- Clock generation for the FSM
    process
    begin
        clk <= '0';
        wait for 5 ns;
        clk <= '1';
        wait for 5 ns;
    end process;
```



```
 -- Test process
    process
    begin
        -- Initial reset, state to S0
        reset <= '1';
        wait for 10 ns;
        reset <= '0';
        wait for 10 ns;

        -- Test #1: Sequence to check for 0110
        x <= '0'; wait for 10 ns; -- Bit 1 S0 to S1
        x <= '1'; wait for 10 ns; -- Bit 2 S1 to S2
        x <= '1'; wait for 10 ns; -- Bit 3 S2 to S3
        x <= '0'; wait for 10 ns; -- Bit 4,S3 to S4: should match 0110
        assert y = '1' report "Test for sequence 0110 failed";

        -- Reset again to S0 for the next test
        reset <= '1';
        wait for 10 ns;
        reset <= '0';
        wait for 10 ns;

        -- Test #2: Sequence to check for 0111
        x <= '0'; wait for 10 ns; -- Bit 1 S0 to S1
        x <= '1'; wait for 10 ns; -- Bit 2 S1 to S2
        x <= '1'; wait for 10 ns; -- Bit 3 S2 to S3
        x <= '1'; wait for 10 ns; -- Bit 4,S3 to S5: should match 0111
        assert y = '1' report "Test for sequence 0111 failed";

        -- Reset back to S0
        reset <= '1';
        wait for 10 ns;
        reset <= '0';

        report "All tests passed";
        wait;
    end process;
end tb_architecture;
```