Universidade de Aveiro – Departamento de Electrónica, Telecomunicações e Informática

Laboratório de Sistemas Digitais

Ano Letivo 2013/14

Mini-teste 2

Nome:_____ N. Mec.:_____ Turma:_____

Grupo I

Desenhe o diagrama de estados de Mealy de uma máquina de estados finitos que compara as sequências binárias apresentadas nas suas entradas A e B. A sua função é activar a saída, Y, sempre que identificar a situação A=B em pelo menos 3 bits consecutivos. Analise o exemplo seguinte para melhor perceber o funcionamento pretendido.

A: 0 1 0 **1 0 1** 0 0 1 1 **0 0 1 1 0** 1 0

B: 0 1 1 **1 0 1** 1 0 1 0 **0 0 1 1 0** 0 1

Y: 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0 0

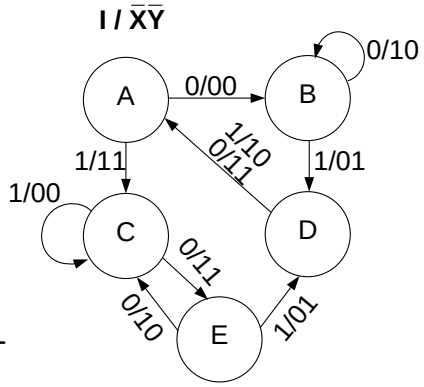
Sugestão: Pode usar notação algébrica para exprimir as condições de transição de estado.

Grupo II

1. Identifique, justificando, o modelo de máquina de estados representada no diagrama de estados abaixo.

---

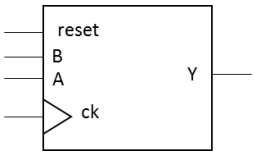
2. Complete o programa abaixo de acordo com o diagrama de estados (respeite os nomes indicados).




```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity MyStateMachine is
    port (clk     : in std_logic;
          ____    : in std_logic;
          _______ : ____ std_logic);
end MyStateMachine;

architecture STM of MyStateMachine is

  type MyST is (_____________________________

  signal _____,_____ : _____________________

  begin
    main : process(____)
    begin
      case ___ is
            when _____ =>

      end case;
    end process;

    control : process (clk)
      begin
        if (rising_edge(clk)) then
            CS <= NS;
        end if;
    end process;
end STM;
```

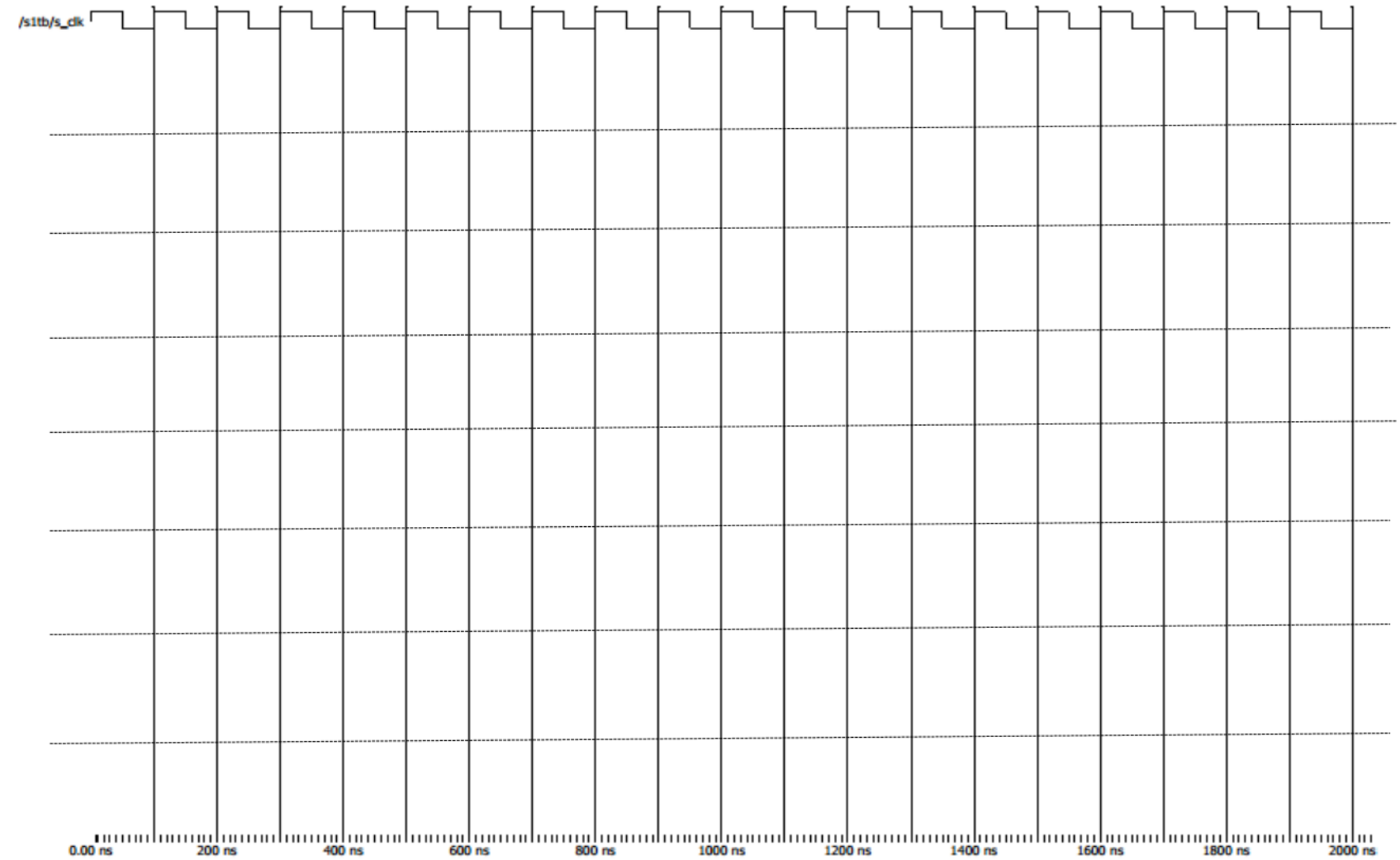
Grupo III

O sistema sequencial síncrono da figura implementado segundo o modelo de Moore compara duas sequências binárias A e B. A saída Y estará *active high* sempre se verificar a situação A=B em pelo menos 3 bits consecutivos. O sistema tem ainda uma entrada de reset assíncrona *active-low* que conduz o sistema ao estado inicial com a saída a "0".




```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity ABcompareTb is
end ABcompareTb;

architecture Stimulus of ABcompareTb is
  signal s_reset, s_clk: std_logic;
  signal s_A,s_B s_Y : std_logic;

begin
  uut : entity work.ABCompare(Behav)
        port map(reset  => s_reset,
                 ck     => s_clk,
                 A      => s_A,
                 B      => s_B,
                 Y      => s_Y);
  clock_proc : process
  begin
    s_clk <= '1'; wait for 50 ns;
    s_clk <= '0'; wait for 50 ns;
  end process;

  stim_proc : process
  begin
        s_reset <= '0'; s_A <= '0'; s_B <= 0; wait for 250 ns;
       s_reset <= '1'; wait for 100 ns;
        s_A <= '1'; s_B <= '1'; wait for 100 ns;
        s_A <= '0'; s_B <= '1'; wait for 100 ns;
        s_A <= '1'; s_B <= '1'; wait for 100 ns;
        s_A <= '0'; s_B <= '0'; wait for 100 ns;
        s_A <= '1'; s_B <= '1'; wait for 100 ns;
        s_A <= '0'; s_B <= '1'; wait for 100 ns;
        s_A <= '0'; s_B <= '0'; wait for 100 ns;
        s_A <= '1'; s_B <= '1'; wait for 100 ns;
        s_A <= '1'; s_B <= '0'; wait for 100 ns;
        s_A <= '0'; s_B <= '0'; wait for 200 ns;
        s_A <= '1'; s_B <= '1'; wait for 200 ns;
        s_A <= '0'; s_B <= '0'; wait for 100 ns;
        s_A <= '1'; s_B <= '0'; wait for 100 ns;
        s_A <= '0'; s_B <= '1'; wait for 100 ns;
  end process;
end Stimulus;
```

1. Complete adequadamente o diagrama temporal no sentido de demonstrar o funcionamento do detetor de sequências. Tenha em conta as especificações do código *testbench*, e justifique a evolução temporal dos sinais que considera relevantes para a simulação.



Entity:s1tb Architecture:stimulus Date: Wed Apr 30 15:41:40 GMT Daylight Time 2014 Row: 1 Page: 1