

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

Grupo I

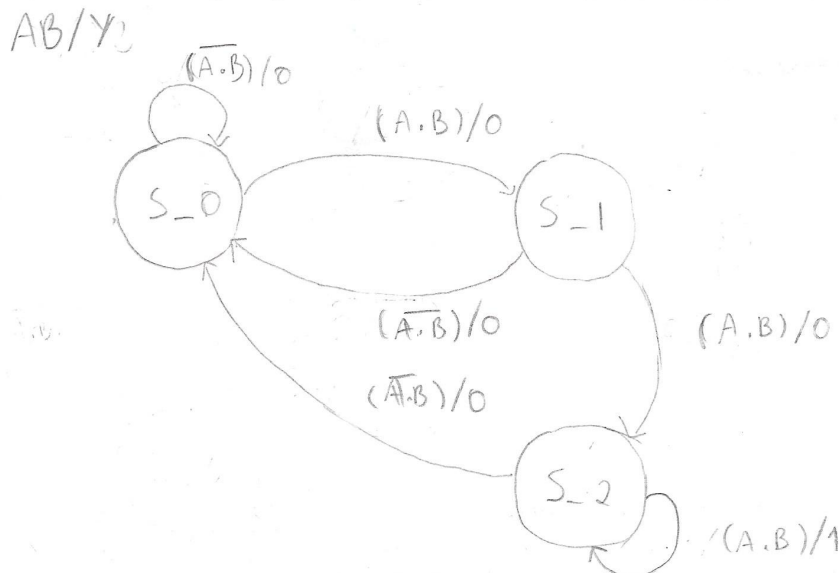
Desenhe o diagrama de estados de Mealy de uma máquina de estados finitos que compara as sequências binárias apresentadas nas suas entradas A e B. A sua função é activar a saída, Y, sempre que identificar a situação $A=B$ em pelo menos 3 bits consecutivos. Analise o exemplo seguinte para melhor perceber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

A: 01010100110011010

B: 01110110100011001

Y: 0000010000001100

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.

Maly, o valor de saída depende das entradas e do estado atual.

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;
          I        : in std_logic;
          X,Y      : out std_logic);
end MyStateMachine;
```

```
architecture STM of MyStateMachine is
```

```
    type MyST is (A, B, C, D, E);
    signal PS, NS : MyST;
```

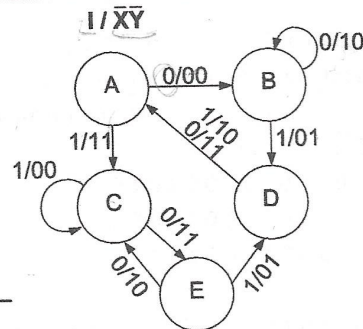
```
begin
    main : process(PS, I)
    begin
```

```
        case PS is
            when A =>
                if (I = '0') then
                    NS <= B;
                    X <= '1';
                    Y <= '1';
                else
                    NS <= C;
                    X <= '0';
                    Y <= '1';
                end if;
            when B =>
                if (I = '1') then
                    NS <= D;
                    X <= '1';
                    Y <= '0';
                else
                    NS <= B;
                    X <= '0';
                    Y <= '1';
                end if;
        end case;
```

```
    end process;
```

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

```
        end if;
    end process;
end STM;
```

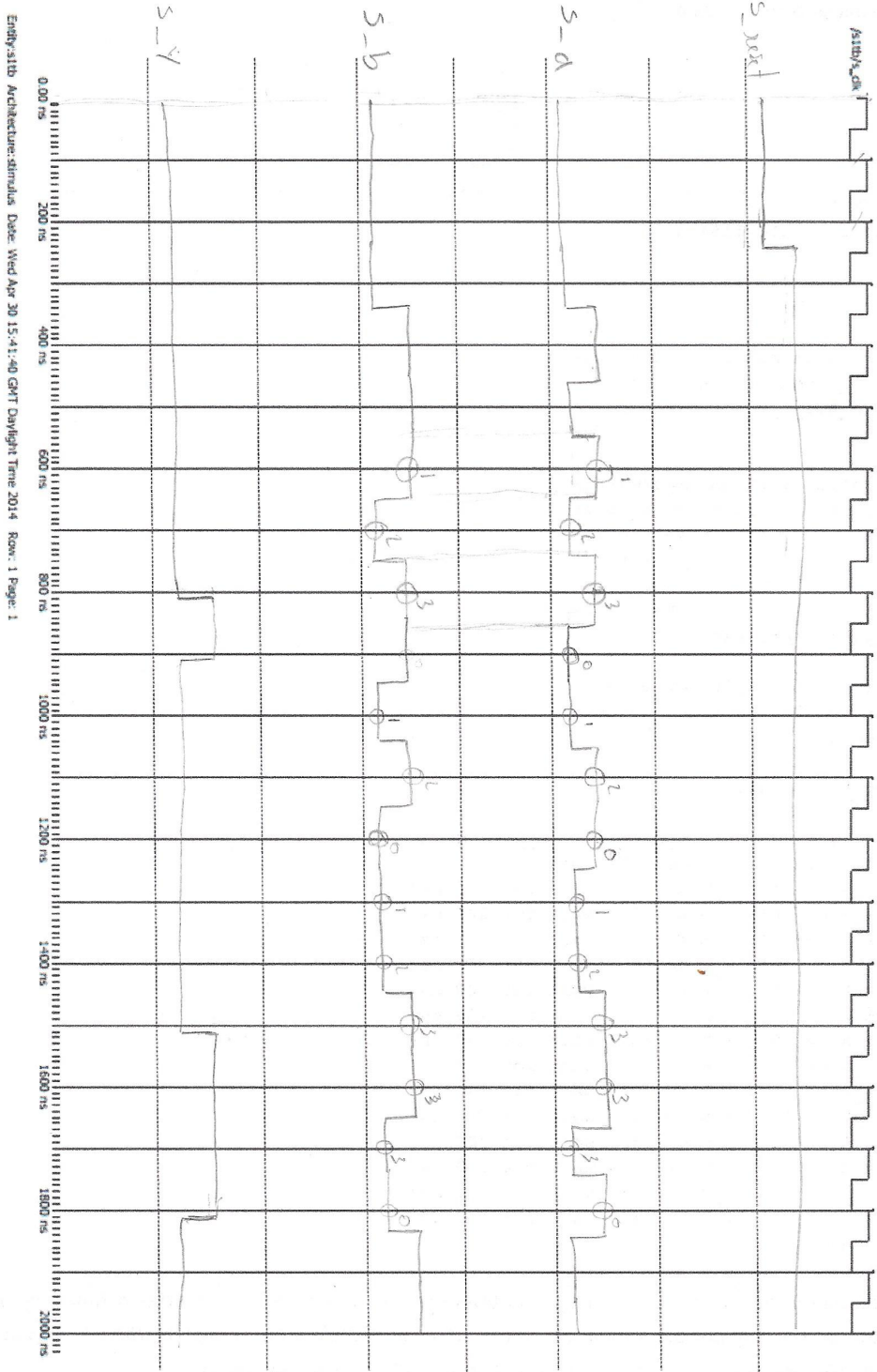


```

when C =>
    if (I = '1') then
        NS <= E;
        X <= '0';
        Y <= '0';
    else
        NS <= C;
        X <= '1';
        Y <= '1';
    end if;

when D =>
    NS <= A;
    if (I = '1') then
        X <= '0';
        Y <= '1';
    else
        X <= '0';
        Y <= '0';
    end if;

when E =>
    if (I = '1') then
        NS <= D;
        X <= '1';
        Y <= '0';
    else
        NS <= C;
        X <= '0';
        Y <= '1';
    end if;
    
```



Universidade de Aveiro – Departamento de Electrónica, Telecomunicações e Informática
Laboratório de Sistemas Digitais

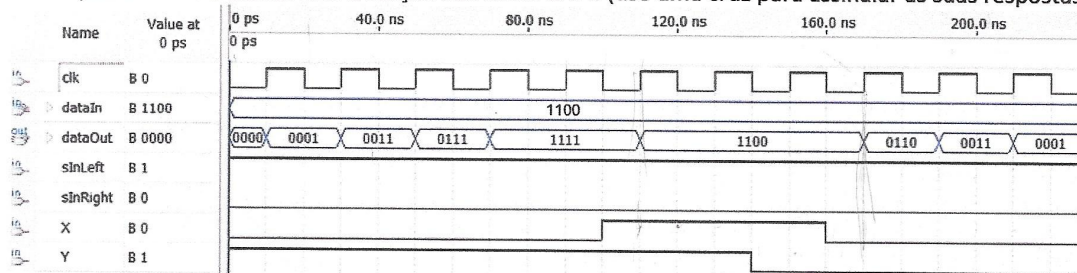
Ano letivo 2014/15

Mini-teste 2 – 8 de maio de 2015

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

Grupo I

1. [2 valores] Analise o seguinte diagrama temporal que ilustra o comportamento de um registo de deslocamento com entrada paralela de dados **dataIn** e saída paralela de dados **dataOut**. Preencha adequadamente a tabela com a função dos sinais **X** e **Y** (use uma cruz para assinalar as suas respostas).



Sinal	reset?	load?	deslocamento para a esquerda quando está a '0'?	deslocamento lógico para a direita quando está a '0'?	deslocamento aritmético para a direita quando está a '0'?
X		X			
Y				X	

2. [4 valores] Complete o código VHDL da *test bench* que permite simular o comportamento do registo de deslocamento e leve ao diagrama temporal tão próximo quanto possível do ilustrado.

```

entity ShifterTb is
end ShifterTb;
architecture Stimulus of ShifterTb is
    signal s_clk, s_X, s_Y, s_sInLeft, s_sInRight : std_logic;
    signal s_dataIn, s_dataOut : std_logic_vector (3 downto 0);
begin
    uut: entity work.Shifter(RTL)
        port map (clk => s_clk,
                  X => s_X, Y => s_Y,
                  sInLeft => s_sInLeft,
                  sInRight => s_sInRight,
                  dataIn => s_dataIn,
                  dataOut => s_dataOut);

    clock_proc: process
    begin
        s_clk <= '0';
        wait for 10ns;
        s_clk <= '1';
        wait for 10ns;
    end process;

    stim_proc: process
    begin
        s_dataIn <= "1100";
        s_sInLeft <= '1';
        s_sInRight <= '0';
        s_X <= '0';
        s_Y <= '1';
        wait for 100ns;
        s_X <= '1';
        wait for 110ns;
        s_Y <= '0';
        wait for 20ns;
        s_X <= '0';
    end process;
end Stimulus;

```


Grupo II

1. [1 valor] Identifique, justificando, o modelo de máquina de estados representada no diagrama de estados abaixo (com uma entrada, X , e uma saída, Z).

Moore, pois o valor da saída só depende do estado atual.

2. [4 valores] Complete o código 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;
          X : in std_logic;
          Z : out std_logic);
end MyStateMachine;
```

```
architecture STM of MyStateMachine is
    type TState is (J, K, L);
    signal pState, nState : TState;
begin
```

```
    sync_proc: process (clk)
    begin
        if (rising_edge(clk)) then
```

```
            end if;
        end process;
```

```
    comb_proc: process (pState, X)
    begin
```

```
        case pState is
            when J =>
```

```
                Z <= 0;
                if (X = '0') then
                    nState <= J;
                else nState <= K;
                end if;
```

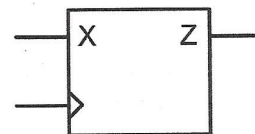
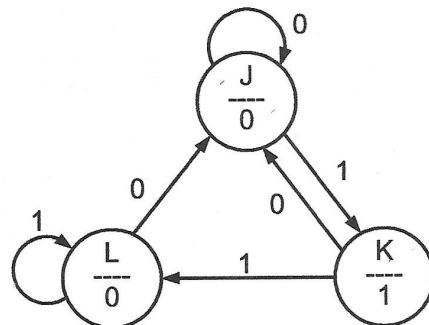
```
            when K =>
```

```
                Z <= 1;
                if (X = '0') then
                    nState <= J;
                else
                    nState <= L;
                end if;
```

```
            when L =>
```

```
                Z <= '0';
```

```
        end case;
    end process;
end STM;
```

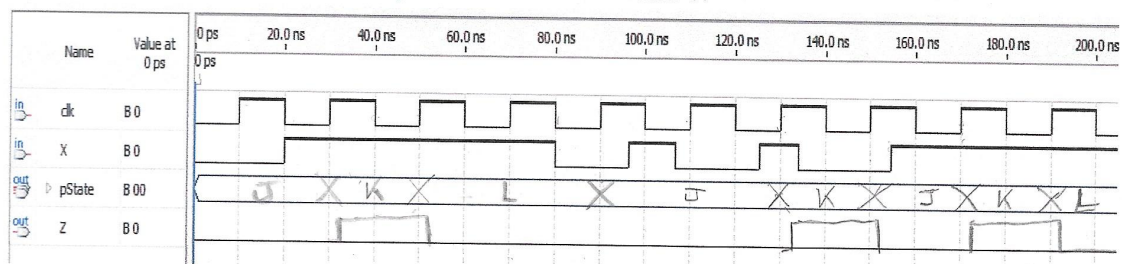


```

if (X <= '0')
    nState <= J;
else nState <= L;
end if;
when others =>
    nState <= J;

```

3. [2 valores] Analise o diagrama de estados da figura anterior e complete o diagrama temporal seguinte (sinais *pState* e *Z* – as linhas representadas servem apenas de referência visual para a sua resposta). Assuma que no momento 0 a máquina se encontra no estado *J*.



Grupo III

1. [2 valores] Analise os extratos de código seguintes e determine que *hardware* será sintetizado. Preencha a tabela com os resultados, assinalando as respostas com uma cruz.

<pre>-- código A process (enable, dataIn) begin if (enable = '1') then dataOut <= dataIn; end if; end process;</pre>	<pre>-- código C validOut <= '1'; output <= "00"; if input(3) = '1' then output <= "11"; elsif input (2) = '1' then output <= "10"; elsif input (1) = '1' then output <= "01"; elsif input (0) = '1' then output <= "00"; else validOut <= '0'; end if;</pre>
<pre>-- código B if (sel = '0') then output <= input0; else output <= input1; end if;</pre>	<pre>-- código D outputs <= "0000" when (enable = '0') else "0001" when (inputs = "00") else "0010" when (inputs = "01") else "0100" when (inputs = "10") else "1000" when others;</pre>

Código	Latch	Flip-flop	Multiplexer	Descodificador	Codificador de prioridade	Somador
A	X					
B			X			
C					X	
D				X		

Grupo IV

Analise o código VHDL seguinte.

1. [2 valores] Desenhe o diagrama de blocos que corresponde ao circuito descrito. Identifique claramente entradas/saídas externas, o nome de cada bloco, os seus portos e os nomes de sinais que interligam os blocos entre si. Use o espaço disponível na última página.

2. [1 valor] Determine a função do circuito.

3. [1 valor] Determine o número de ciclos de relógio (*N*) necessários para calcular o resultado. Justifique adequadamente a sua resposta.

4. [1 valor] Assumindo que *SW*="1111 0000 0011 1111", determine o valor da saída *LEDG*, após *N* ciclos de relógio, em que *N* é o valor obtido na questão anterior.

04/09.

```

entity Circuit is
  port(CLOCK_50      : in  std_logic;
        KEY          : in  std_logic_vector(0 downto 0);
        SW           : in  std_logic_vector(15 downto 0);
        LEDG         : out std_logic_vector(7 downto 0));
end Circuit;

architecture Structural of Circuit is
  signal s_en        : std_logic;
  signal s_xored     : std_logic_vector(7 downto 0);
  constant s_value   : std_logic_vector(7 downto 0) := "00000001";
  signal s_acc, s_add : std_logic_vector(7 downto 0);
begin

  s_xored <= SW(15 downto 8) xor SW(7 downto 0);

  get: entity work.ShiftReg(RTL)      -- registo de deslocamento
    port map (clk    => CLOCK_50,
              load    => KEY(0),
              dataIn  => s_xored,
              serOut   => s_en);

  add: entity work.Adder(RTL)         -- somador: result = A + B
    port map (A      => s_value,
              B       => s_acc,
              result  => s_add);

  store: entity work.Reg(RTL)        -- registo
    port map (clk    => CLOCK_50,
              reset   => KEY(0), -- reset síncrono, põe o registo a 0
              en      => s_en,   -- enable
              dataIn  => s_add,
              dataOut => s_acc);

  LEDG <= s_acc;
end Structural;

```

