## 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

lome:				
	Gr	upo I		
. [2 valores] Analise o seg deslocamento com entrad adequadamente a tabela c	a paralela de dados <b>d</b> o om a função dos sinais	ataIn e saída X e Y (use um	paralela de da a cruz para assi	ados <i>dataOut</i> . Preench nalar as suas respostas).
Name Value at 0 ps 0 ps	40.0 ns 80.0	ns 120,0	ns 160,0 r	ns 200.0 ns
S clk B 0				
> dataIn B 1100		1100		
	01 \ 0011 \ 0111 \	1111 X	1100	0110 0011 0001
sInLeft B 1				
sInRight B 0				
X 80				
Y B1				
Sinal reset? load?	deslocamento para a esquerda quando está a '0'?		eita quando	deslocamento aritmético para a direita guando está a '0'?
Х				
Υ				
deslocamento e leve ao d	agrama temporal tão ¡			portamento do registo d ustrado.
ntity ShifterTb is nd ShifterTb; rchitecture Stimulus o signal s_clk, s_X, s	of ShifterTb is _Y, s_sInLeft, s_	oróximo quanto	o possível do ilu	ustrado.
ntity ShifterTb is nd ShifterTb; rchitecture Stimulus o   signal s_clk, s_X, s   signal s_dataIn, s_d	of ShifterTb is _Y, s_sInLeft, s_	oróximo quanto	o possível do ilu	ustrado.
ntity ShifterTb is nd ShifterTb; rchitecture Stimulus of signal s_clk, s_X, s_ signal s_dataIn, s_d egin ut: entity work.Shift port map(clk => s_ X => s_X sInLeft	of ShifterTb is _Y, s_sInLeft, s_ ataOut : er(RTL) _clk, , Y => s_Y, => s_sInLeft,	sInRight :	o possível do ilu	ustrado.
ntity ShifterTb is nd ShifterTb; rchitecture Stimulus of signal s_clk, s_X, s_signal s_dataIn, s_dataIn egin ut: entity work.Shift    port map(clk => s_X	of ShifterTb is _Y, s_sInLeft, s_ ataOut : er(RTL) _clk, , Y => s_Y,	sInRight :	o possível do ilu	ustrado.
ntity ShifterTb is nd ShifterTb; rchitecture Stimulus of signal s_clk, s_X, s_signal s_dataIn, s_dataIn egin ut: entity work.Shift    port map(clk => s_X	of ShifterTb is _Y, s_sInLeft, s_ ataOut :  er(RTL) _clk, , Y => s_Y,	sInRight :	o possível do ilu	ustrado.
ntity ShifterTb is nd ShifterTb; rchitecture Stimulus of signal s_clk, s_X, s_ signal s_dataIn, s_d egin  ut: entity work.Shift port map(clk => s_	of ShifterTb is _Y, s_sInLeft, s_ ataOut :  er(RTL) _clk, , Y => s_Y,	sInRight :	o possível do ilu	ustrado.
ntity ShifterTb is nd ShifterTb; rchitecture Stimulus of signal s_clk, s_X, s_signal s_dataIn, s_dataIn egin ut: entity work.Shifter     port map(clk => s_X	of ShifterTb is _Y, s_sInLeft, s_ ataOut :  er(RTL) _clk, , Y => s_Y,	sInRight :	o possível do ilu	ustrado.
ntity ShifterTb is nd ShifterTb; rchitecture Stimulus of signal s_clk, s_X, s_signal s_dataIn, s_dataIn egin ut: entity work.Shifter     port map(clk => s_X	of ShifterTb is _Y, s_sInLeft, s_ ataOut :  er(RTL) _clk, , Y => s_Y,	sInRight :	o possível do ilu	ustrado.

end process;

end Stimulus;

end process;

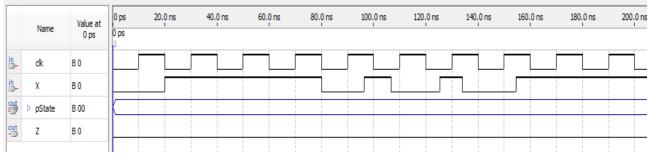
# **Grupo II**

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

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
nomb (all a in add logica
: in std_logic;
: std_logic);
end MyStateMachine; ( )
( 0 )
architecture STM of MyStateMachine is
type TState is ();
signal pstate, nstate: Tstate;
begin
( 7 L \ , Y K \
sync_proc: process (clk)
begin 0 / 1 /
if (rising_edge(clk)) then
$\longrightarrow$ x z $\longmapsto$
end if;
end process;
end process,
$\longrightarrow$
comb_proc: process(pState,)
begin
case is
when =>
end case;

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.

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

Código	Latch	Flip-flop	Multiplexer	Descodificador	Codificador de prioridade	Somador
A						
В						
С						
D						

#### **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.

```
entity Circuit is
      port(CLOCK_50 : in std_logic;
    KEY : in std_logic_vector(0 downto 0);
                         : in std_logic_vector(15 downto 0);
            SW
            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);</pre>
      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,
                        В
                               => s_acc,
                        result => s_add);
      store: entity work.Reg(RTL)
                                              -- registo
                               => CLOCK_50,
             port map (clk
                                => KEY(0), -- reset síncrono, põe o registo a 0
                        reset
                                => s_en,
                        en
                                              -- enable
                        dataIn => s_add,
                        dataOut => s_acc);
      LEDG <= s_acc;</pre>
end Structural;
```

