UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ELETRÔNICA E SISTEMAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DOCENTE RESPONSÁVEL: DR. MARCO AURÉLIO BENEDETTI RODRIGUES

DOCENTE ESTAGIÁRIO: MSC. MALKI-ÇEDHEQ B. C. SILVA

Informações da Disciplina

Curso: ENGENHARIA ELETRÔNICA – CTG

Disciplina: ELETRÔNICA DIGITAL 1A

Código: ES441, Turma: EB, Semestre: 2022.1

LISTA DE EXERCÍCIOS

AHDL, VHDL e Verilog HDL

Atenção: Não utilizar *softwares* para solução desta lista, deve-se resolver com papel e caneta e anexar a foto indicando a questão a que se refere.

ALUNO(A): _____

1. Analise a descrição de *hardware* apresentada na Figura 1, e esboce o diagrama de temporização, ao menos os dez primeiros ciclos de clock, para as portas (entradas e saídas) do circuito lógico modelado em AHDL. Considere o clock de entrada de 50MHz. Adicione no gráfico o rótulo, correto, de tempo para cada borda de subida e descida do clock.

Figura 1. Código em AHDL

```
SUBDESIGN q1_fsm_moore_ahdl
(
   clk_in : INPUT;
   saida  : OUTPUT;
)
VARIABLE
   fsm: MACHINE WITH STATES (s0,s1,s2);
BEGIN
   fsm.clk = clk_in;
   TABLE
      fsm =>  fsm, saida;
      s0  =>  s1 , GND;
      s1  =>  s2 , GND;
      s2  =>  s0 , VCC;
   END TABLE;
END;
```

2. Analise a descrição de *hardware* apresentada na Figura 2, e esboce o diagrama de temporização, ao menos os dez primeiros ciclos de clock, para as portas (entradas e saídas) do circuito lógico modelado em VHDL. Considere o clock de entrada de 50MHz. Adicione no gráfico o rótulo, correto, de tempo para cada borda de subida e descida do clock.

```
 1  LIBRARY ieee;
 2  USE ieee.std_logic_1164.ALL;
 3  USE ieee.numeric_std.ALL;
 4 ENTITY q2_fsm_moore_vhdl IS
 5 PORT( clk_in : IN STD_LOGIC;
 6       saida  : OUT STD_LOGIC);
 7 END q2_fsm_moore_vhdl;
 8 ARCHITECTURE bhv OF q2_fsm_moore_vhdl IS
 9    TYPE estados IS (s0, s1, s2);
10    SIGNAL estado_atual: estados;
11    SIGNAL proximo_estado: estados;
12 BEGIN
13    L1: PROCESS(clk_in)
14    BEGIN
15       IF rising_edge(clk_in) THEN
16          estado_atual <= proximo_estado;
17       END IF;
18    END PROCESS L1;
19    L2: PROCESS (estado_atual)
20    BEGIN
21       CASE estado_atual IS
22          WHEN s0 => proximo_estado <= s1;
23          WHEN s1 => proximo_estado <= s2;
24          WHEN s2 => proximo_estado <= s0;
25          WHEN OTHERS => proximo_estado <= s0;
26       END CASE;
27    END PROCESS L2;
28    L3: PROCESS (clk_in, estado_atual)
29    BEGIN
30       IF rising_edge(clk_in) THEN
31          CASE estado_atual IS
32             WHEN s0 => saida <= '0';
33             WHEN s1 => saida <= '1';
34             WHEN s2 => saida <= '0';
35          END CASE;
36       END IF;
37    END PROCESS L3;
38 END bhv;
```

3. Analise a descrição de *hardware* apresentada na Figura 3, e esboce o diagrama de temporização, ao menos os dez primeiros ciclos de clock, para as portas (entradas e saídas) do circuito lógico modelado em Verilog HDL. Considere o clock de entrada de 50MHz. Adicione no gráfico o rótulo, correto, de tempo para cada borda de subida e descida do clock.

```
1  module q3_fsm_moore_verilog (
2      output reg saida,
3      input clk_in);
4      localparam s0 = 2'b00;
5      localparam s1 = 2'b01;
6      localparam s2 = 2'b11;
7      reg [1:0] estado_atual, prox_estado;
8      always @ (posedge clk_in)
9      begin : L1
10        estado_atual <= prox_estado;
11     end
12     always @ (estado_atual)
13     begin : L2
14        case (estado_atual)
15           s0: prox_estado = s1;
16           s1: prox_estado = s2;
17           s2: prox_estado = s0;
18           default: prox_estado = s0;
19        endcase
20     end
21     always @ (posedge clk_in)
22     begin : L3
23        case (estado_atual)
24           s0: saida <= 1'b0;
25           s1: saida <= 1'b1;
26           s2: saida <= 1'b0;
27        endcase
28     end
29  endmodule
30
```

4. Analise o circuito lógico modelado em AHDL apresentado na Figura 4, e esboce o circuito RTL (*Register Transfer Level*) equivalente, ou seja, o circuito lógico a nível de registradores, multiplexadores, comparadores, operadores aritméticos e portas lógicas. Explique o circuito digital modelado, sua construção, modo de operação e função.

Figura 4. Código em AHDL

```
1   CONSTANT NumPulsos = 4;
2   CONSTANT Overflow_f = NumPulsos-1;
3   CONSTANT n_bits_f= LOG2(overflow_f);
4   SUBDESIGN q4_divisor_clock_ahdl(
5      clk_in   : INPUT;
6      clk_out  : OUTPUT;
7   )
8   VARIABLE
9      cnt[n_bits_f..0]  : DFF;
10     toggle      : TFF;
11  BEGIN
12     cnt[].clk   =  clk_in;
13     toggle.clk  =  clk_in;
14     IF cnt[] < Overflow_f THEN
15        cnt[].d  = cnt[]+1;
16     ELSE
17        cnt[].d  = 0;
18        toggle.t = VCC;
19     END IF;
20     clk_out = toggle;
21  END;
```

5. Analise o circuito lógico modelado em VHDL apresentado na Figura 5, e esboce o circuito RTL (*Register Transfer Level*) equivalente, ou seja, o circuito lógico a nível de registradores, multiplexadores, comparadores, operadores aritméticos e portas lógicas. Explique o circuito digital modelado, sua construção, modo de operação e função.

Figura 5. Código em VHDL

```
1  LIBRARY IEEE;
2  USE IEEE.std_logic_1164.ALL;
3  ENTITY q5_divisor_clock_vhdl IS
4     PORT (clk_in : IN std_logic;
5           clk_out: OUT std_logic);
6  END q5_divisor_clock_vhdl;
7
8  ARCHITECTURE bhv OF q5_divisor_clock_vhdl IS
9     CONSTANT NumPulsos_c : integer := 4;
10    CONSTANT Overflow_c : integer := NumPulsos_c-1;
11    SIGNAL Toggle_s : std_logic := '0';
12 BEGIN
13    PROCESS(clk_in)
14       VARIABLE Cnt_v : integer RANGE 0 TO Overflow_c;
15    BEGIN
16       IF rising_edge(clk_in) THEN
17          IF Cnt_v < Overflow_c THEN
18             Cnt_v := Cnt_v + 1;
19             Toggle_s <= Toggle_s;
20          ELSE
21             Cnt_v := 0;
22             Toggle_s <= not Toggle_s;
23          END IF;
24       END IF;
25    END PROCESS;
26    clk_out <= Toggle_s;
27 END bhv;
```

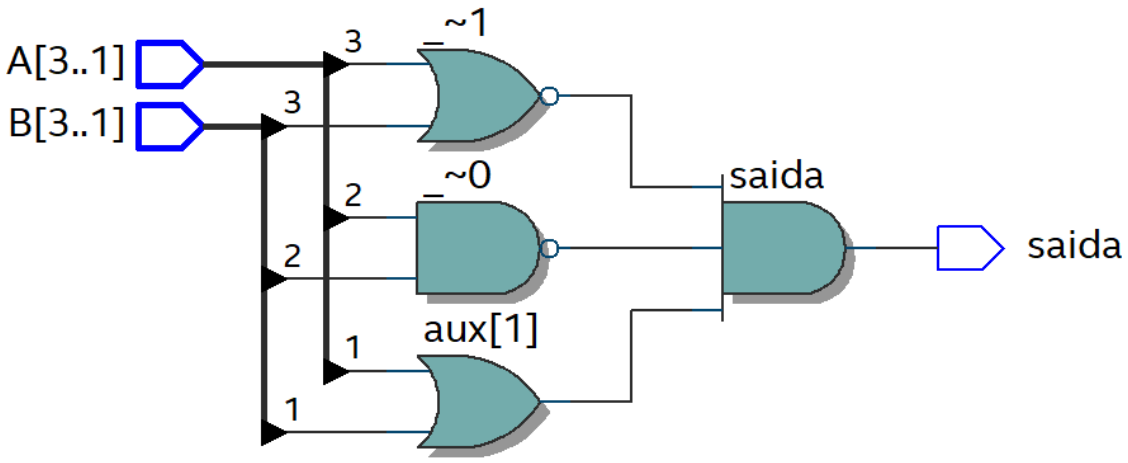
6. Analise o circuito lógico modelado em Verilog HDL apresentado na Figura 6, e esboce o circuito RTL (*Register Transfer Level*) equivalente, ou seja, o circuito lógico a nível de registradores, multiplexadores, comparadores, operadores aritméticos e portas lógicas. Explique o circuito digital modelado, sua construção, modo de operação e função.

Figura 6. Código em Verilog HDL

```
1  module q6_divisor_clock_verilog(
2      input clk_in,
3      output reg clk_out = 0 );
4
5      localparam NumPulsos = 4;
6      localparam Overflow = NumPulsos-1;
7      reg [$clog2(Overflow)-1:0] cnt;
8
9      always @(posedge clk_in)
10     begin
11        if (cnt < Overflow) begin
12           cnt <= cnt + 1'b1;
13           clk_out <= clk_out;
14        end
15        else begin
16           cnt <= 0;
17           clk_out <= ~clk_out;
18        end
19     end
20 endmodule
21
```

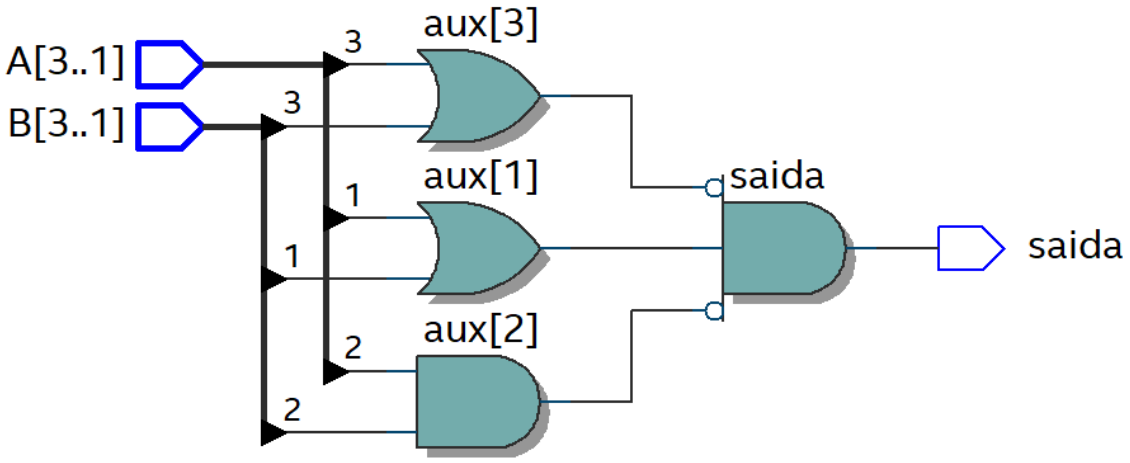
7. Analise o circuito RTL, em destaque na Figura 7, e modele-o através de um módulo utilizando a linguagem de descrição de *hardware* AHDL, utilizando o conceito de lógica gerada iterativamente. Explique o código.

Figura 7. Circuito Lógico Digital



8. Analise o circuito RTL, em destaque na Figura 8, e modele-o através de um módulo utilizando a linguagem de descrição de *hardware* VHDL, utilizando o conceito de lógica gerada iterativamente. Explique o código.

Figura 8. Circuito Lógico Digital



9. Analise o circuito RTL, em destaque na Figura 9, e modele-o através de um módulo utilizando a linguagem de descrição de *hardware* Verilog HDL, utilizando o conceito de lógica gerada iterativamente. Explique o código.

Figura 9. Circuito Lógico Digital