

GRUPO 5 - TURMA A

---

---

## Segundo laboratório de Circuitos Digitais

*MC 613 - Primeiro Semestre de 2010*

---

---

PROFESSOR: GUIDO ARAÚJO

HENRIQUE SERAPIÃO GONZALES	RA: 083636
MARCELO GALVÃO PÓVOA	RA: 082115
TIAGO CHEDRAOUI SILVA	RA: 082941

*28 de março de 2010*

## **1 Questão 1:**

## 2 Questão 2:

Listing 1: Descriptive Caption Text

```
1  LIBRARY ieee ;
2  USE ieee.std_logic_1164.all ;
3
4  ENTITY q2a IS
5      PORT (a,bi: IN STD_LOGIC_VECTOR(0 to 3);
6              z: BUFFER STD_LOGIC_VECTOR(0 to 3);
7              s: IN STD_LOGIC;
8              ovf: OUT STD_LOGIC);
9  END q2a;
10
11 ARCHITECTURE struct OF q2a IS
12     SIGNAL cint, b: STD_LOGIC_VECTOR(0 to 3);
13 COMPONENT somador IS
14     PORT ( c0, a0, a1 : IN STD_LOGIC ;
15             b0, c1 : OUT STD_LOGIC ) ;
16 END COMPONENT somador;
17 BEGIN
18     PROCESS (bi, s)
19     BEGIN IF s = '0' THEN
20         b(3)<=not bi(3) xor (not bi(0) and not bi(1) and not bi(2));
21         b(2)<=not bi(2) xor (not bi(0) and not bi(1));
22         b(1)<=not bi(1) xor not bi(0);
23         b(0)<=bi(0);
24     ELSE b<=bi;
25     END IF;
26 END PROCESS;
27
28     s0: COMPONENT somador
29         PORT MAP ('0', a(0), b(0), z(0), cint(0));
30     s1: COMPONENT somador
31         PORT MAP (cint(0), a(1), b(1), z(1), cint(1));
32     s2: COMPONENT somador
33         PORT MAP (cint(1), a(2), b(2), z(2), cint(2));
34     s3: COMPONENT somador
35         PORT MAP (cint(2), a(3), b(3), z(3), cint(3));
36
37     ovf<=(b(3) and a(3) and not z(3)) or (not b(3) and not a(3) and z(3));
38 END struct;
```

### 3 Questão 3:

Utilizando o chip 74284 projetou-se um circuito que executasse a multiplicação de 2 números de 4 bits. O chip 74284 possuía como entrada 4 bits de um número e 4 bits de do outro número, contudo retornava somente os 4 bits mais significativos da multiplicação. Para o cálculo dos 4 bits menos significativos utilizando componentes básicas do circuito lógico. A idéia central era analisar a relevância de cada multiplicação de bits para os bits da saída. Em que temos:

				x4y0	x3y0	x2y0	x1y0	x0y0
			x4y1	x3y1	x2y1	x1y1	x0y1	
		x4y2	x3y2	x2y2	x1y2	x0y2		
	x4y3	x3y3	x2y3	x1y3	x0y3			
x4y4	x3y4	x2y4	x1y4	x0y4				
-----								
z8	z7	z6	z5	z4	z3	z2	z1	z0

Como queríamos z3,z2,z1 e z0 deveríamos ter portanto:

$$z0 = x0y0$$

$$z1 = x0y1 + x1y0$$

$$z2 = x2y0 + x1y1 + x0y2$$

$$z3 = x3y0 + x2y1 + x1y2 + x0y3$$

Utilizando o componente somador, usamo-lo como somadores totais (com carry na entrada) e parciais (sem carry na entrada), para o calculo dos bits menos significativos. Por exemplo, para o z1. Inicialmente como entrada no somador colocamos (x0 and y1) e (x1 and y0) sendo a saída em cm e z1 (Obs: carry e saída são usados em outros somadores, sendo o primeiro para o calculo do próximo z - nesse caso z2 - e o zm para o calculo do próprio z).

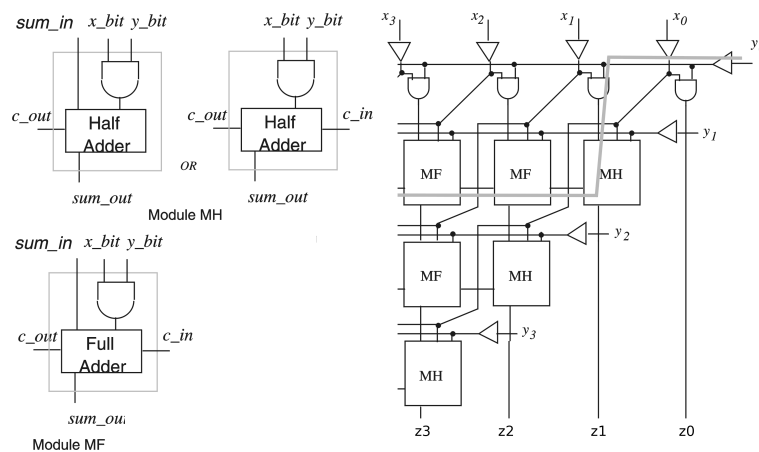


Figura 3.1: Multiplicação de 4 bits, saída dos bits menos significativos

Na segunda parte do projeto desenvolvemos em VHDL um multiplicador utilizando a função pré-definida  $y=a*b$ , sendo  $a$  um inteiro de 4 bits,  $b$  um inteiro de 4 bits e  $y$  um vetor de 8 bits.

Posteriormente fizemos a comparação de todos os resultados possíveis através de um circuito testbench no qual simulamos todas as entradas possíveis.

Listing 2: Multiplicador de 4 bits

```

1  LIBRARY ieee ;
2  USE ieee.std_logic_1164.all ;
3
4  ENTITY q3a IS
5      PORT (x,y: IN STD_LOGIC_VECTOR(3 downto 0);
6           z: OUT STD_LOGIC_VECTOR(7 downto 0));
7  END q3a;
8
9  ARCHITECTURE struct OF q3a IS
10     SIGNAL cm: STD_LOGIC_VECTOR(0 to 2);
11     SIGNAL zm: STD_LOGIC_VECTOR(0 to 2);
12     SIGNAL hi: STD_LOGIC_VECTOR(3 downto 0);
13 COMPONENT somador IS
14     PORT ( c0, a0, a1 : IN STD_LOGIC ;
15           b0, c1 : OUT STD_LOGIC ) ;
16 END COMPONENT somador;
17 COMPONENT \74284\
18     PORT ( gan: IN STD_LOGIC;
19           gbn: IN STD_LOGIC;
20           a: IN STD_LOGIC_VECTOR (4 downto 1);
21           b: IN STD_LOGIC_VECTOR (4 downto 1);
22           y: OUT STD_LOGIC_VECTOR (8 downto 5));
23 END COMPONENT;
24 BEGIN
25     —multiplicador para bits menos significativos
26     z(0)<=x(0) and y(0);
27     s1: COMPONENT somador
28         PORT MAP ('0', y(1) and x(0), y(0) and x(1),      z(1), cm(0));
29     s2: COMPONENT somador
30         PORT MAP (cm(0), y(1) and x(1), y(0) and x(2), zm(0), cm(1));
31     s3: COMPONENT somador
32         PORT MAP ('0', y(2) and x(0), zm(0),      z(2), cm(2));
33     s4: COMPONENT somador
34         PORT MAP (cm(1), x(3) and y(0), y(1) and x(2), zm(1));
35     s5: COMPONENT somador
36         PORT MAP (cm(2), zm(1), y(2) and x(1), zm(2));
37     s6: COMPONENT somador
38         PORT MAP ('0', zm(2), y(3) and x(0),      z(3));
39
40     mul: COMPONENT \74284\
41         PORT MAP ('0', '0', x, y, hi);
42
43     z(7 downto 4)<=hi;
44 END struct;

```

Listing 3: Funcao pre-defenida a\*b

```

1 LIBRARY ieee ;
2 use ieee.std_logic_1164.all;
3 use ieee.std_logic_arith.all;
4 use ieee.std_logic_unsigned.all;
5
6 ENTITY q3b IS
7     PORT (x,y: IN integer range 0 to 15;
8           z: OUT integer range 0 to 255 );
9 END q3b;
10
11 ARCHITECTURE struct OF q3b IS
12
13 BEGIN
14     z<=x*y; —multiplicacao de inteiros
15 END struct;

```

Listing 4: Comparador de resultados

```

1  LIBRARY ieee ;
2  use ieee.std_logic_1164.all;
3  use IEEE.numeric_std.all;
4
5  ENTITY q3c IS
6      PORT (x,y: IN STD_LOGIC_VECTOR(3 downto 0);
7            ok: OUT STD_LOGIC);
8  END q3c;
9
10 ARCHITECTURE struct OF q3c IS
11     SIGNAL za: STD_LOGIC_VECTOR(7 downto 0);
12     SIGNAL zb: integer range 0 to 255;
13     SIGNAL x_int, y_int: integer range 0 to 15;
14 COMPONENT q3a IS
15     PORT (x,y: IN STD_LOGIC_VECTOR(3 downto 0);
16           z: OUT STD_LOGIC_VECTOR(7 downto 0));
17 END COMPONENT q3a;
18 COMPONENT q3b IS
19     PORT (x,y: IN integer range 0 to 15;
20           z: OUT integer range 0 to 255 );
21 END COMPONENT q3b;
22 BEGIN
23     x_int<=to_integer(unsigned(x));
24     y_int<=to_integer(unsigned(y));
25
26     mul_str: COMPONENT q3a
27         PORT MAP (x, y, za);
28
29     mul_com: COMPONENT q3b
30         PORT MAP (x_int, y_int, zb);
31
32     ok<='1' WHEN (to_integer(unsigned(za)) = zb) ELSE '0';
33 END struct;

```



## 4 Questão 4:

Listing 5: Comparador de resultados

```
1  LIBRARY ieee ;
2  USE ieee.std_logic_1164.all ;
3
4  ENTITY q2a IS
5      PORT (a,bi: IN STD_LOGIC_VECTOR(0 to 3);
6              z: BUFFER STD_LOGIC_VECTOR(0 to 3);
7              s: IN STD_LOGIC;
8              ovf: OUT STD_LOGIC);
9  END q2a;
10
11 ARCHITECTURE struct OF q2a IS
12     SIGNAL cint, b: STD_LOGIC_VECTOR(0 to 3);
13 COMPONENT somador IS
14     PORT ( c0, a0, a1 : IN STD_LOGIC ;
15             b0, c1 : OUT STD_LOGIC ) ;
16 END COMPONENT somador;
17 BEGIN
18     PROCESS (bi, s)
19     BEGIN IF s = '0' THEN
20         b(3)<=not bi(3) xor (not bi(0) and not bi(1) and not bi(2));
21         b(2)<=not bi(2) xor (not bi(0) and not bi(1));
22         b(1)<=not bi(1) xor not bi(0);
23         b(0)<=bi(0);
24     ELSE b<=bi;
25     END IF;
26 END PROCESS;
27
28     s0: COMPONENT somador
29         PORT MAP ('0', a(0), b(0), z(0), cint(0));
30     s1: COMPONENT somador
31         PORT MAP (cint(0), a(1), b(1), z(1), cint(1));
32     s2: COMPONENT somador
33         PORT MAP (cint(1), a(2), b(2), z(2), cint(2));
34     s3: COMPONENT somador
35         PORT MAP (cint(2), a(3), b(3), z(3), cint(3));
36
37     ovf<=(b(3) and a(3) and not z(3)) or (not b(3) and not a(3) and z(3));
38 END struct;
```