

Circuitos Lógicos

DIM 0128

MATERIAL TEMPORÁRIO
[[[NÃO REVISADO]]]

Aula 14

Circuitos Combinacionais

Implementação de Multiplicador e Comparador

Edgard de Faria Corrêa

2022.1

Objetivos da Aula

◇ Multiplicador:

- Multiplicação binária.
- Multiplicação de N bits.

◇ Comparador:

- Comparador de igualdade.
- Comparador com 3 saídas (igualdade e desigualdades).

◇ Exercícios

MATERIAL TEMPORÁRIO
[[[NÃO REVISADO]]]

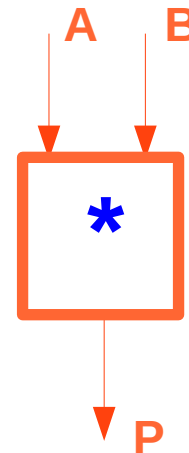
Multiplicador

MATERIAL TEMPORÁRIO
[[[NÃO REVISADO]]]

Multiplicação Binária

- ◇ Multiplicação entre 2 números binários:

A	B	P
0	0	0
0	1	0
1	0	0
1	1	1

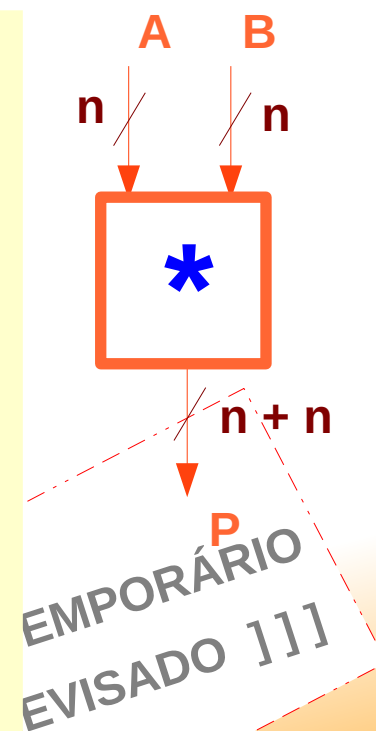


MATERIAL TEMPORÁRIO
[[[NÃO REVISADO]]]

Multiplicação de N bits

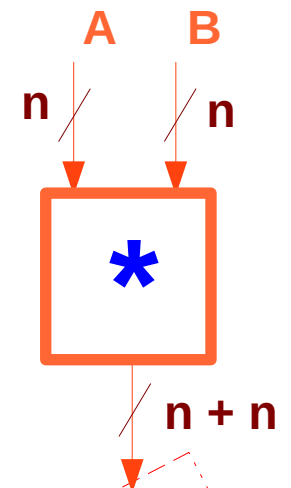
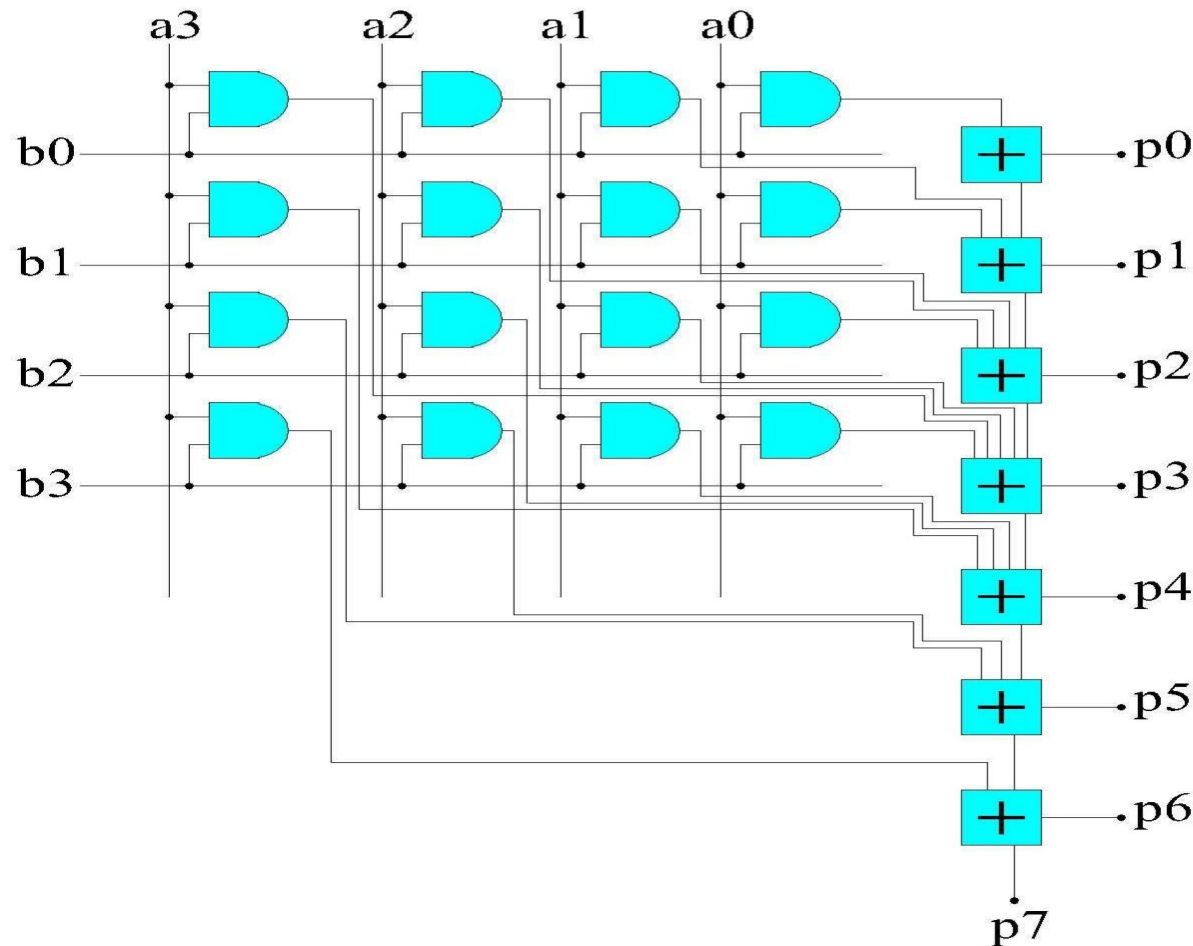
- ❖ Multiplicação entre 2 números binários de **n** bits resulta em um número de **$n+n$** bits:

				a3	a2	a1	a0
x				b3	b2	b1	b0
				b0.a3	b0.a2	b0.a1	b0.a0
				b1.a3	b1.a2	b1.a1	b1.a0
				b2.a3	b2.a2	b2.a1	b2.a0
+	b3.a3	b3.a2	b3.a1	b3.a0			
p7	p6	p5	p4	p3	p2	p1	p0



Multiplicação de N bits

- ❖ Multiplicação entre 2 números binários de **n** bits resulta em um número de **n+n** bits:

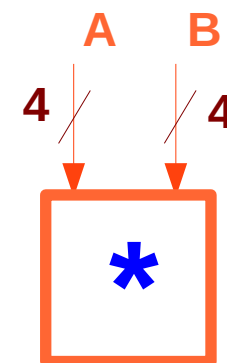


MATERIAL TEMPORÁRIO
[[[NÃO REVISADO]]]

Multiplicação de N bits

- ◇ Multiplicação entre 2 números binários de **4** bits, utilizando o pacote ARITH da biblioteca IEEE:

```
LIBRARY ieee;  
USE ieee.std_logic_1164.all;  
USE ieee.std_logic_ARITH.all;  
USE ieee.std_logic_UNSIGNED.all;  
  
ENTITY Multiplicador IS  
PORT ( A, B : IN STD_LOGIC_VECTOR (3 downto 0);  
        P : OUT STD_LOGIC_VECTOR (7 downto 0) );  
END;  
  
ARCHITECTURE m OF Multiplicador IS  
BEGIN  
    P <= A * B ;  
END m;
```



AL TEMPORÁRIO
AO REVISADO 111

Comparador

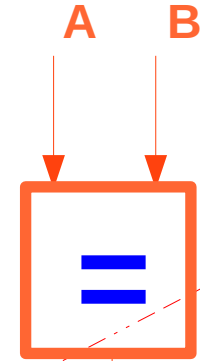
MATERIAL TEMPORÁRIO
[[[NÃO REVISADO]]]

Comparador de Igualdade

- ◇ Compara se duas palavras binárias são iguais.
 - Comparador de duas palavras binárias de **1 bit**:

A	B	C: A=B
0	0	1
0	1	0
1	0	0
1	1	1

$$C = \overline{A} \cdot \overline{B} + A \cdot B = \overline{A \oplus B}$$

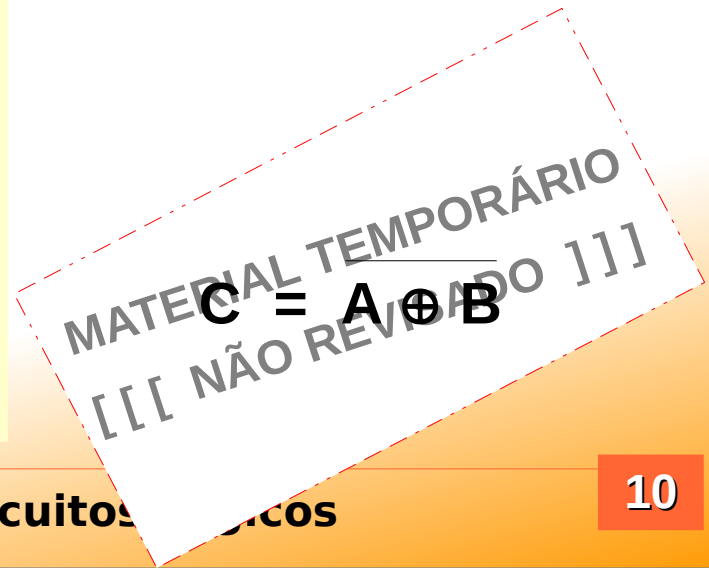


MATERIAL TEMPORÁRIO
[[[NÃO REVISADO]]]

Comparador de Igualdade

- ◇ Compara se duas palavras binárias são iguais.
 - Comparador de duas palavras binárias de **1 bit**:

```
LIBRARY ieee;  
USE ieee.std_logic_1164.all;  
  
ENTITY Comp_1bit IS  
PORT ( A, B : IN STD_LOGIC;  
        C : OUT STD_LOGIC );  
END;  
  
ARCHITECTURE comp OF Comp_1bit IS  
BEGIN  
    C <= A XNOR B ;  
END comp;
```

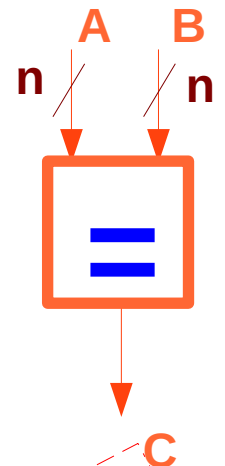


Comparador de Igualdade

◇ Comparador de duas palavras binárias de **N bits**:

■ Comparação bit-a-bit das palavras binárias:

- A_0 com B_0 , A_1 com B_1 , e assim por diante.



- Depois, os resultados passam por uma porta AND para saber se TODAS as comparações tiveram o resultado '1', ou seja se todos os bits eram iguais.

Comparador de Igualdade

- ◇ Comparador de duas palavras binárias de **4 bits**:

```
LIBRARY ieee;  
USE ieee.std_logic_1164.all;  
  
ENTITY Comp_4bits IS  
PORT ( A, B : IN STD_LOGIC_VECTOR(3 DOWNT0 0);  
        C : OUT STD_LOGIC);  
END;
```

```
ARCHITECTURE c4 OF Comp_4bits IS  
  
COMPONENT Comp_1bit  
PORT ( A, B: IN STD_LOGIC;  
        C: OUT STD_LOGIC );  
END COMPONENT;  
  
SIGNAL cp0, cp1, cp2, cp3: STD_LOGIC;
```

```
BEGIN  
  
i0: Comp_1bit PORT MAP (a(0), b(0), cp0);  
i1: Comp_1bit PORT MAP (a(1), b(1), cp1);  
i2: Comp_1bit PORT MAP (a(2), b(2), cp2);  
i3: Comp_1bit PORT MAP (a(3), b(3), cp3);  
C <= cp0 AND cp1 AND cp2 AND cp3;  
  
END c4;
```

Comparador com 3 saídas

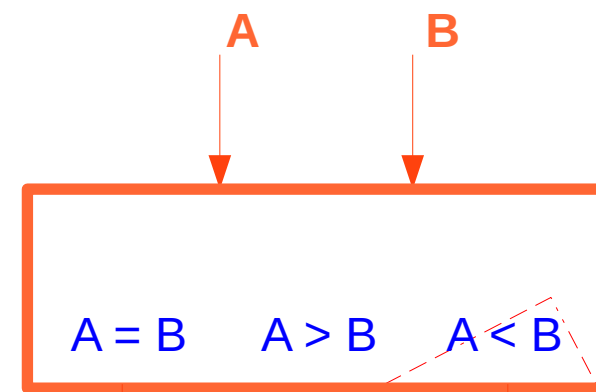
- ◇ Compara se duas palavras binárias são iguais.
- ◇ Se forem diferentes, indica qual é a maior.

A	B	X: A = B	Y: A > B	Z: A < B
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

$$X = \overline{A} \cdot \overline{B} + A \cdot B = \overline{A \oplus B}$$

$$Y = A \cdot \overline{B}$$

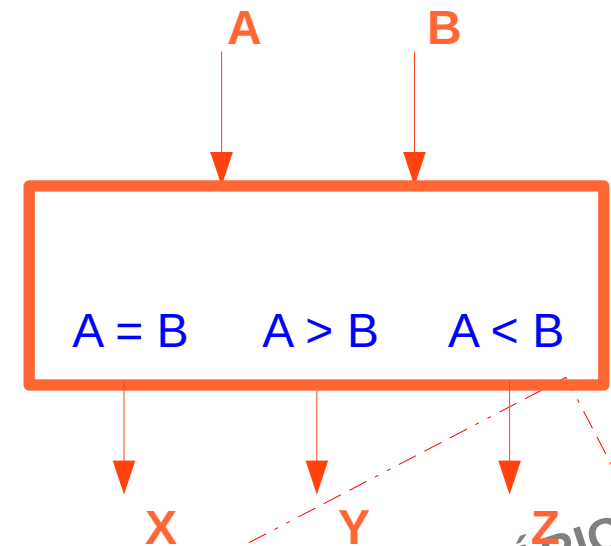
$$Z = \overline{A} \cdot B$$



Comparador com 3 saídas

- ◇ Compara se duas palavras binárias são iguais.
- ◇ Se forem diferentes, indica qual é a maior.

```
LIBRARY ieee;  
USE ieee.std_logic_1164.all;  
  
ENTITY Comp_3saidas IS  
PORT ( A, B : IN STD_LOGIC;  
        X, Y, Z : OUT STD_LOGIC );  
END;  
  
ARCHITECTURE comp OF Comp_3saidas IS  
BEGIN  
    X <= A XNOR B;  
    Y <= A AND (NOT B);  
    Z <= (NOT A) AND B;  
END comp;
```



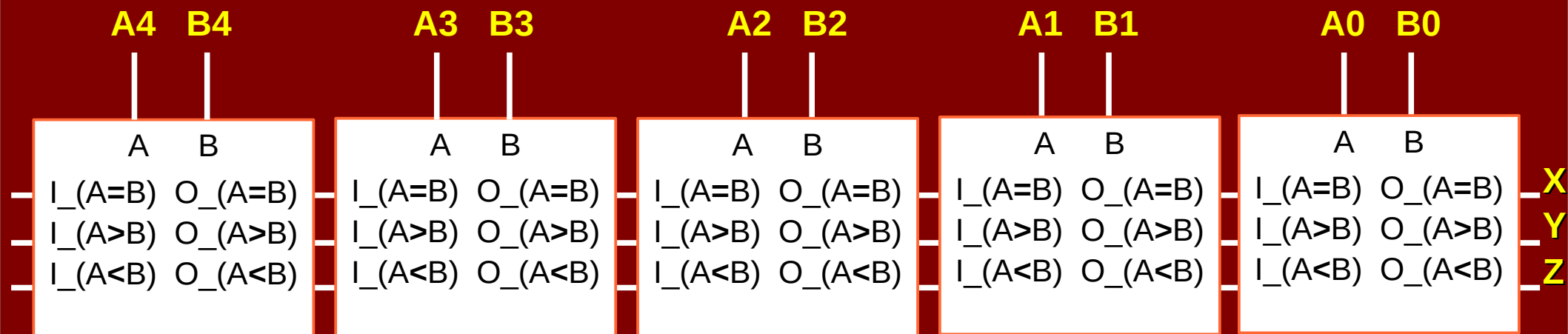
MATERIAL TEMPORÁRIO
[[[NÃO REVISADO]]]

Exercícios

MATERIAL TEMPORÁRIO
[[[NÃO REVISADO]]]

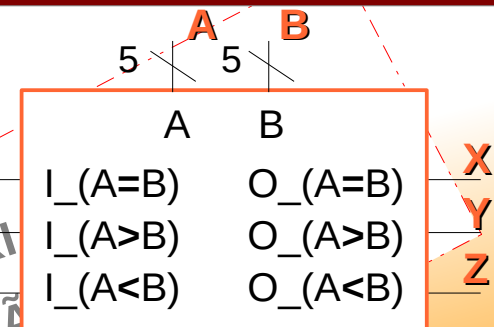
Exercício 1

- ◇ Implemente em VHDL o comparador de 5 bits com 3 saídas (X, Y, Z) da figura abaixo:



◇ DICA:

- Crie uma entidade num projeto que realize operação de um bloco comparador, como na figura ao lado.
- Use componentes com esta entidade no comparador de 5 bits.



Exercício 2

- ◇ Projete um circuito que determina o maior de 3 números de 8 bits.
 - Podem ser utilizados:
 - os operadores $>$, $<$, $=$
 - as estruturas de condição do VHDL.
 - Na simulação, garanta que qualquer possível situação seja testada:
 - 1 dos 3 números é o maior;
 - 2 números são iguais e são os de maior valor;
 - Todos tem o mesmo valor.

MATERIAL TEMPORÁRIO
[[[NÃO REVISADO]]]

Referências Bibliográficas

◇ Livros:

1. Frank Vahid. **Sistemas digitais: projeto, otimização e HDLS**. Artmed Bookman, 2008.
2. Volnei Pedroni. **Eletrônica Digital Moderna e VHDL**. Campus, 2010.
3. L.Mesquita; E.Pinheiro; C.Costa. **Elementos de Lógica Programável com VHDL e DSP: Teoria & Prática**. ERICA, 2011.
4. R. Tocci, N. S. Widmer. **Sistemas Digitais Princípios e Aplicações**. 10 ed. Pearson, 2007.

◇ Apostilas:

1. David Déharbe. **VHDL: notas de aula**. UFRN, 1998.
2. Ronaldo Hüsemann. **Apostila de VHDL**. UFRGS, 2001.
3. Hardi Electronics. **VHDL handbook**. 2000.

MATERIAL TEMPORÁRIO
[[[NÃO REVISADO]]]