

PCS 3115

Sistemas Digitais I

Circuitos Combinatórios **Blocos Básicos:** **Comparadores**

Prof. Dr. Marcos A. Simplicio Jr.

versão: 3.0 (Jan/2016)

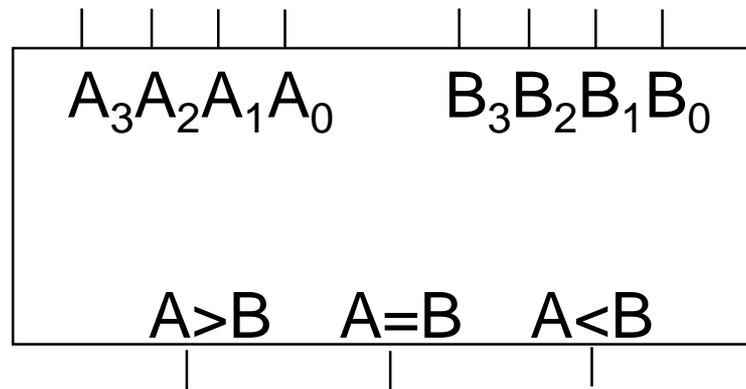
Adaptado por Glauber (2018)

Blocos básicos

- Codificadores e Decodificadores
- (De) Multiplexadores
- Portas tri-state
- Comparadores (**HOJE**)
- Somadores/Subtratores
- **16 de maio: P2**
- Multiplicadores
- ULA
- Gerador/Detector de Paridade

Comparadores

- Comparação entre palavras binárias é uma operação comum em sistemas digitais.
- Comparadores realizam essa função e podem indicar igualdade ($=$, \neq), e em alguns casos, relação aritmética ($>$, $<$).



Detour: portas XOR e XNOR

- Recap: portas lógicas de OU-exclusivo (XOR)
 - $X \oplus Y = (X' \cdot Y) + (X \cdot Y')$
 - Resultados (equivalentes)
 - 1 se **apenas uma** das entradas for 1, 0 caso contrário
 - 1 se **ambas as entradas são diferentes**, 0 caso contrário

▪ Tabela-verdade

| X | Y | $X \oplus Y$ |
|---|---|--------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

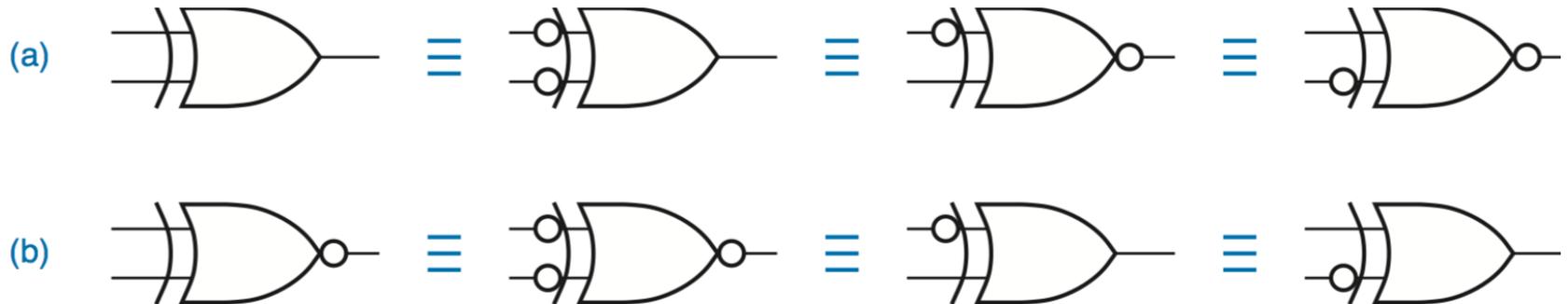
▪ Porta lógica

(representação gráfica)



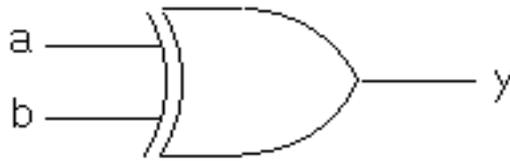
Portas XOR e XNOR

- Porta XNOR (b) é XOR com saída ativa em BAIXO
 - $(X \oplus Y)' = ((X' \cdot Y) + (X \cdot Y'))' = (X + Y') \cdot (X' + Y)$
 $= X \cdot Y + X' \cdot Y'$
- Símbolos alternativos para XOR (a) e XNOR (b)



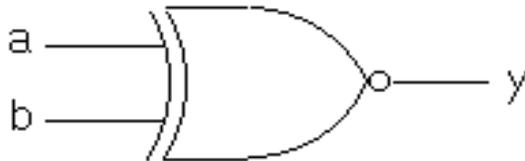
Comparadores

- XOR e XNOR são comparadores de 1 bit



$$Y = a \oplus b = \text{Dif}$$

Dif = 1, se entradas são diferentes



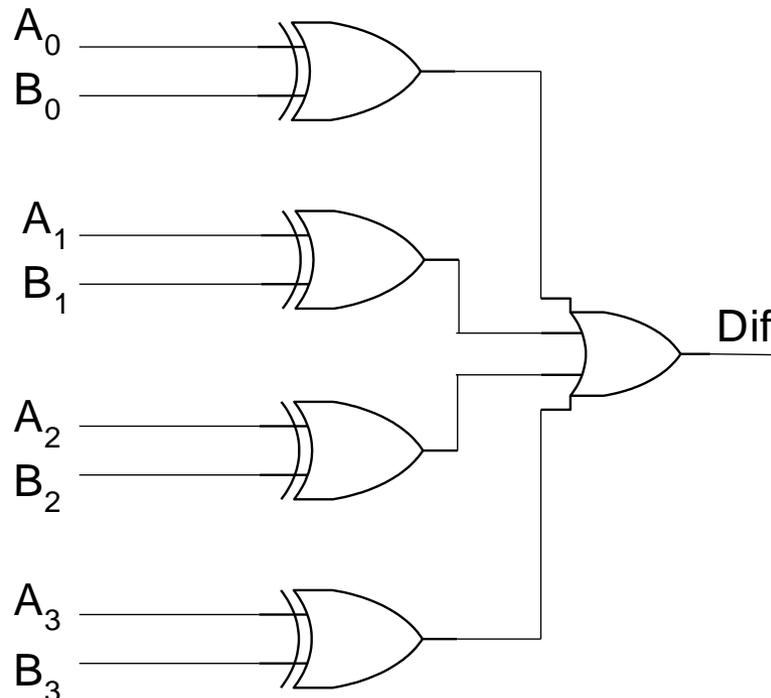
$$Y = (a \oplus b)' = \text{Eq}$$

Eq = 1, se entradas são iguais

- **Pergunta:** como fazer um comparador de n bits...?

Comparadores

- Comparador (paralelo) de n bits
 - Compara bit a bit, duas palavras de n bits,
 - Sumariza o resultado

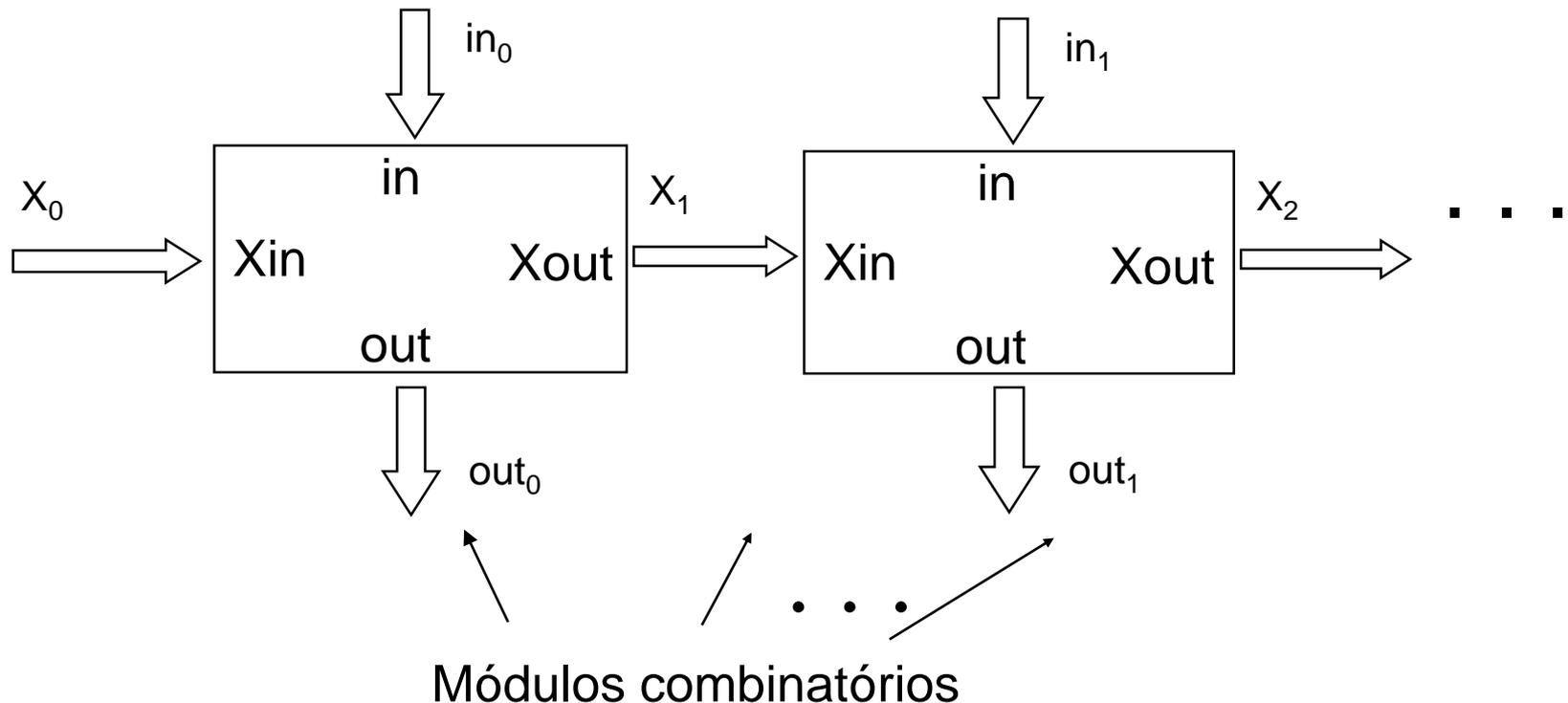


Comparadores “iterativos”

- Software iterativo:
 - Repete (itera) um bloco de código (for, while)
- Circuitos combinatórios “iterativos”:
 - n módulos idênticos cascadeados
 - Cada módulo possui:
 - Entradas e saídas primárias (in e out)
 - Entradas e saídas para **associação em cascata** (X)
 - Adequados para problemas que podem ser resolvidos com algoritmos iterativos.
 - Mas mais lentos que circuitos paralelos equivalentes

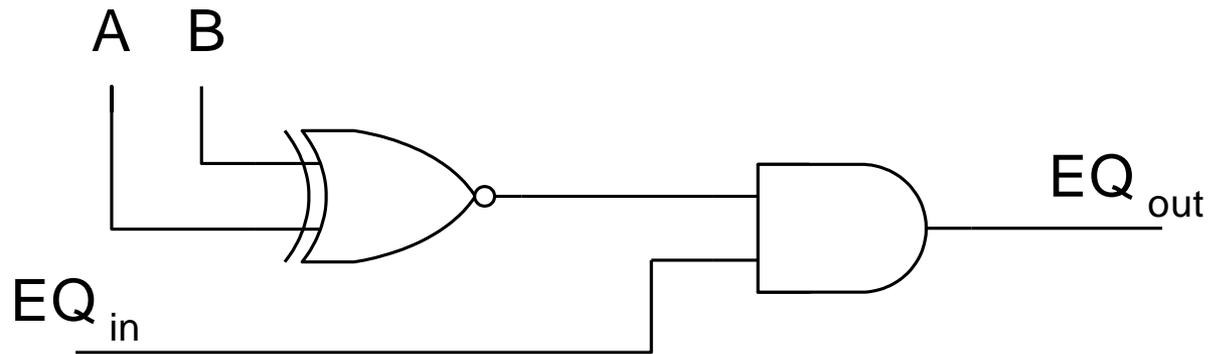
Comparadores “iterativos”

- Cada módulo possui:
 - Entradas e saídas primárias (in e out)
 - Entradas e saídas para **associação em cascata** (X)

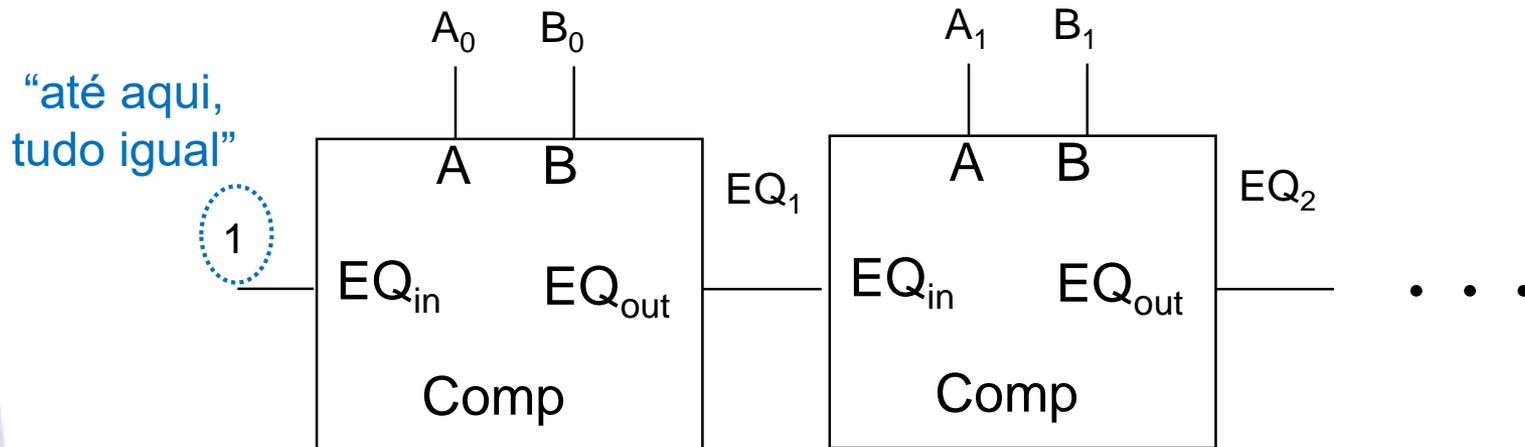


Comparadores “iterativos”

- Módulo combinatório básico do comparador



- Módulos do comparador associados em cascata



Comparador de magnitude

- Indica relação aritmética entre as palavras de dados comparadas
- **Pergunta** (comparação de 1 bit): quando $A_i > B_i$?
 - Resposta: quando $A_i = 1$ e $B_i = 0$
- **Pergunta** (comparação de n bits, sem sinal): como um humano faria essa comparação?
 - **Resposta**: analisar desigualdades dos bits, começando no mais significativo → $A > B$ se
 - $A_3 > B_3$ (i.e. $A_3=1$ e $B_3=0$)
 - Ou se $A_3=B_3$ e $A_2 > B_2$
 - Ou se $A_3=B_3$ e $A_2=B_2$ e $A_1 > B_1$
 - Ou se $A_3=B_3$ e $A_2=B_2$ e $A_1=B_1$ e $A_0 > B_0$

Comparador de magnitude

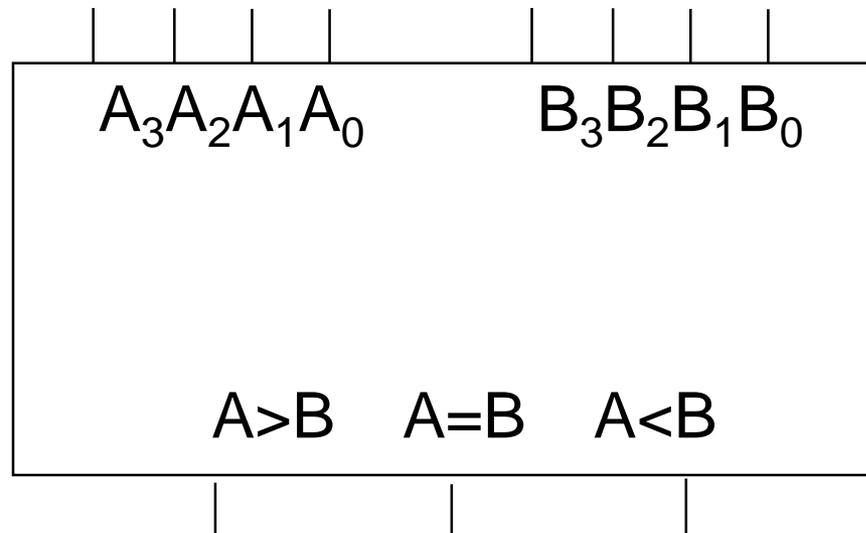
- $A=B$ se

- $A_3=B_3$ e
- $A_2=B_2$ e
- $A_1=B_1$ e
- $A_0=B_0$

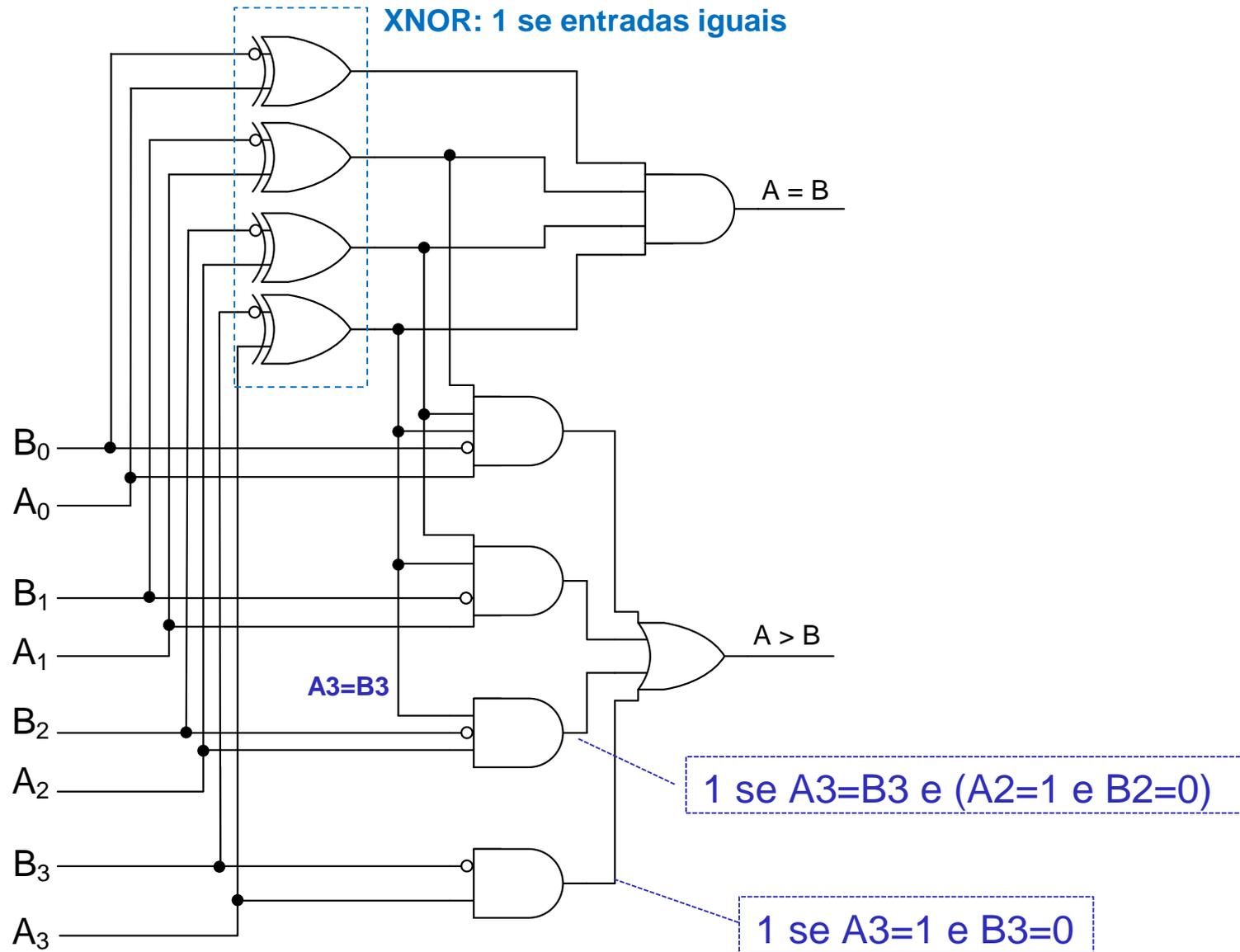
- $A>B$ se

- $A_3>B_3$
- Ou se $A_3=B_3$ e $A_2>B_2$
- Ou se $A_3=B_3$ e $A_2=B_2$ e $A_1>B_1$
- Ou se $A_3=B_3$ e $A_2=B_2$ e $A_1=B_1$ e $A_0>B_0$

$A_i > B_i$ se
 $A_i=1$ e $B_i=0$

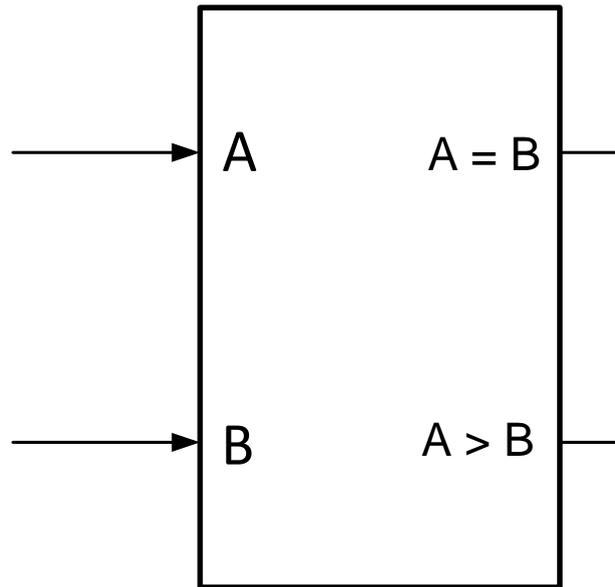


Comparador de magnitude



Comparador de magnitude

- Pergunta: como obter as demais relações?



A dif. B

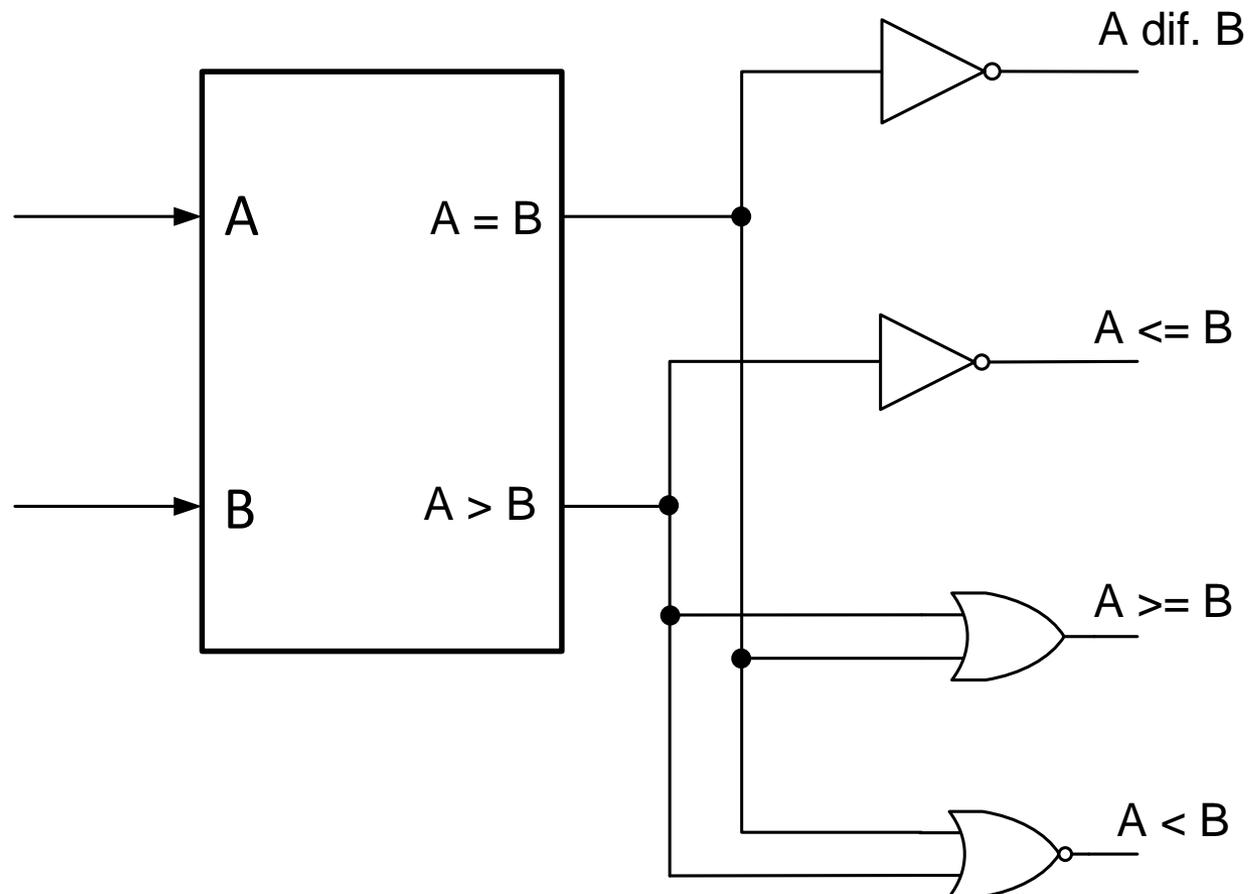
A <= B

A >= B

A < B

Comparador de magnitude

- Pergunta: como obter as demais relações?



Comparador 74x85

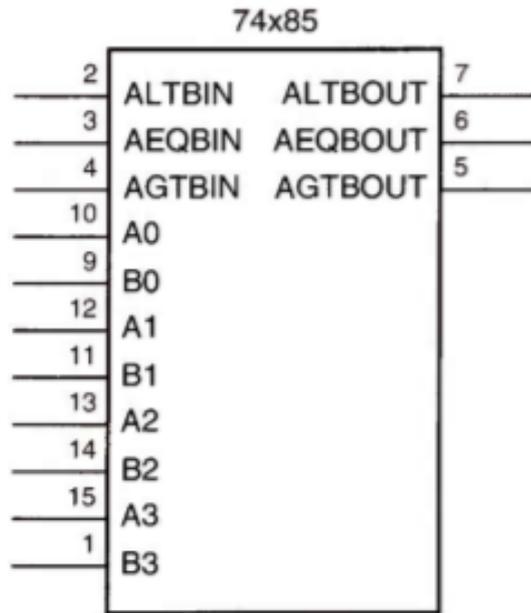


Figure 6-78

Traditional logic symbol for the 74x85 4-bit comparator.

$$\text{AGTBOUT} = (A > B) + (A = B) \cdot \text{AGTBIN}$$

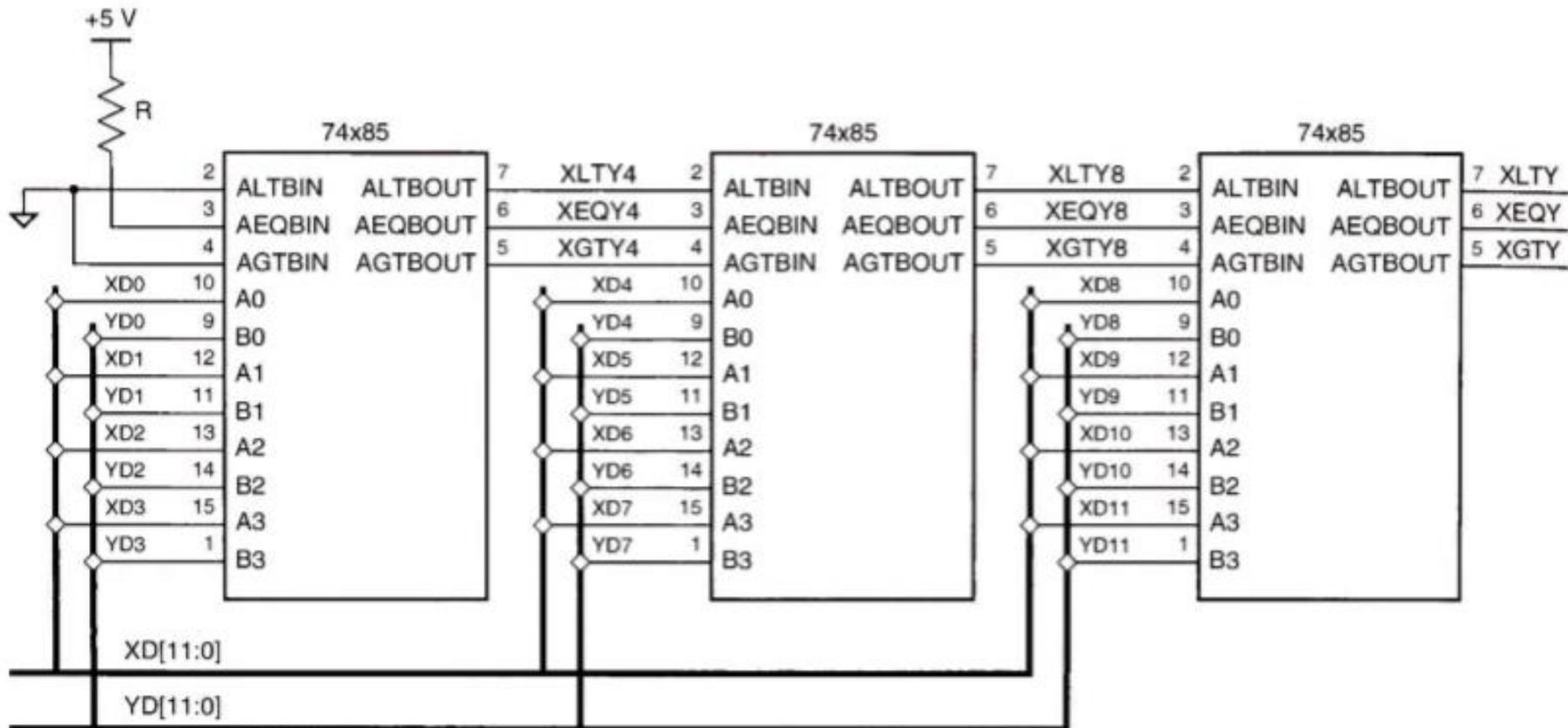
$$\text{AEQBOUT} = (A = B) \cdot \text{AEQBIN}$$

$$\text{ALTBOUT} = (A < B) + (A = B) \cdot \text{ALTBIN}$$

- Entradas ALTBIN, AEQBIN e AGTBIN: Cascadeamento!

Comparadores 74x85 em Cascata

Comparando 12 bits usando 3 comparadores de 4 bits:



Comparador de magnitude: VHDL

- ```
library IEEE;
use IEEE.std_logic_1164.all;

entity compmag is
 port (inpA,inpB : in std_logic_vector(3 downto 0);
 greater, equal, smaller : out std_logic);
end compmag;

architecture compmag_arch of compmag is
begin
 -- std_logic e std_logic_vector suportam comparações
 greater <= '1' when (inpA > inpB) else '0';
 equal <= '1' when (inpA = inpB) else '0';
 smaller <= '1' when (inpA < inpB) else '0';
end compmag_arch;
```

# Lição de Casa

- Leitura Sugerida:
  - Capítulo 6 do Livro Texto, Seção 6.9.
- Exercícios sugeridos:
  - Capítulo 6 do Livro Texto (Comparadores).

# Exercícios

- Projete um comparador análogo ao 74x85, porém com a ordem de cascadeamento oposta. Isto é, entradas de cascadeamento recebem o resultado da comparação de bits mais significativos.