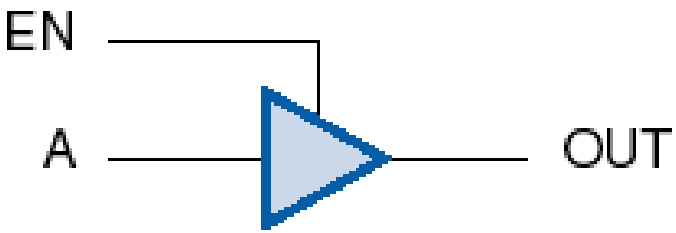
Sistemas Digitais

Buffers 3 estados (Three-state buffers)

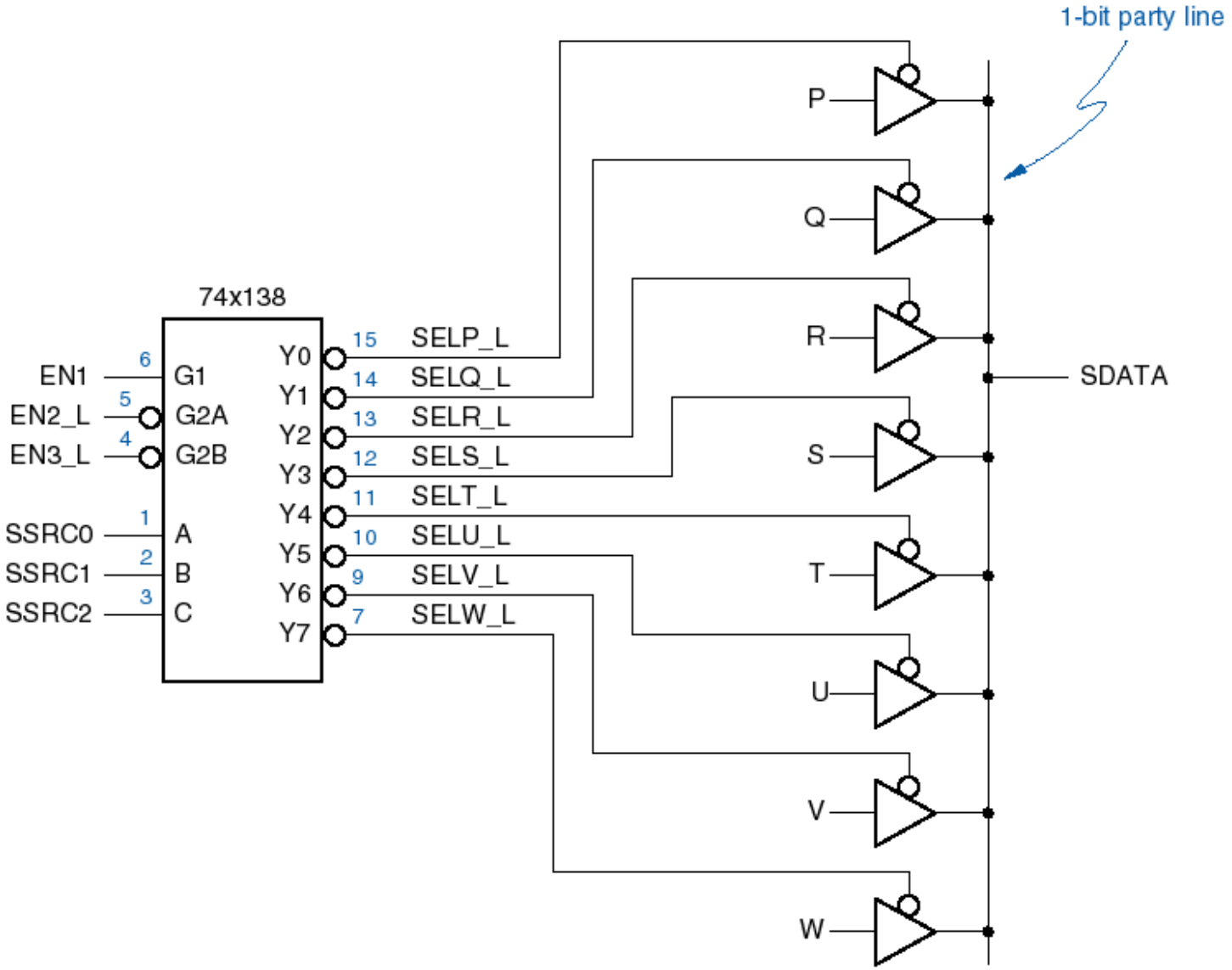
- SAÍDA = LOW, HIGH, or Hi-Z.

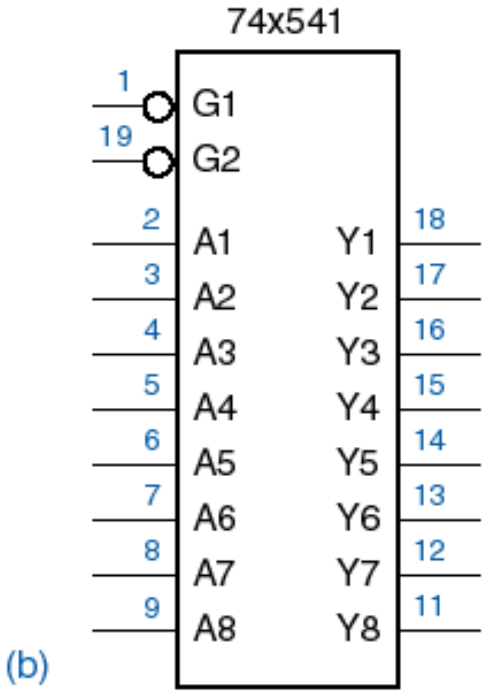


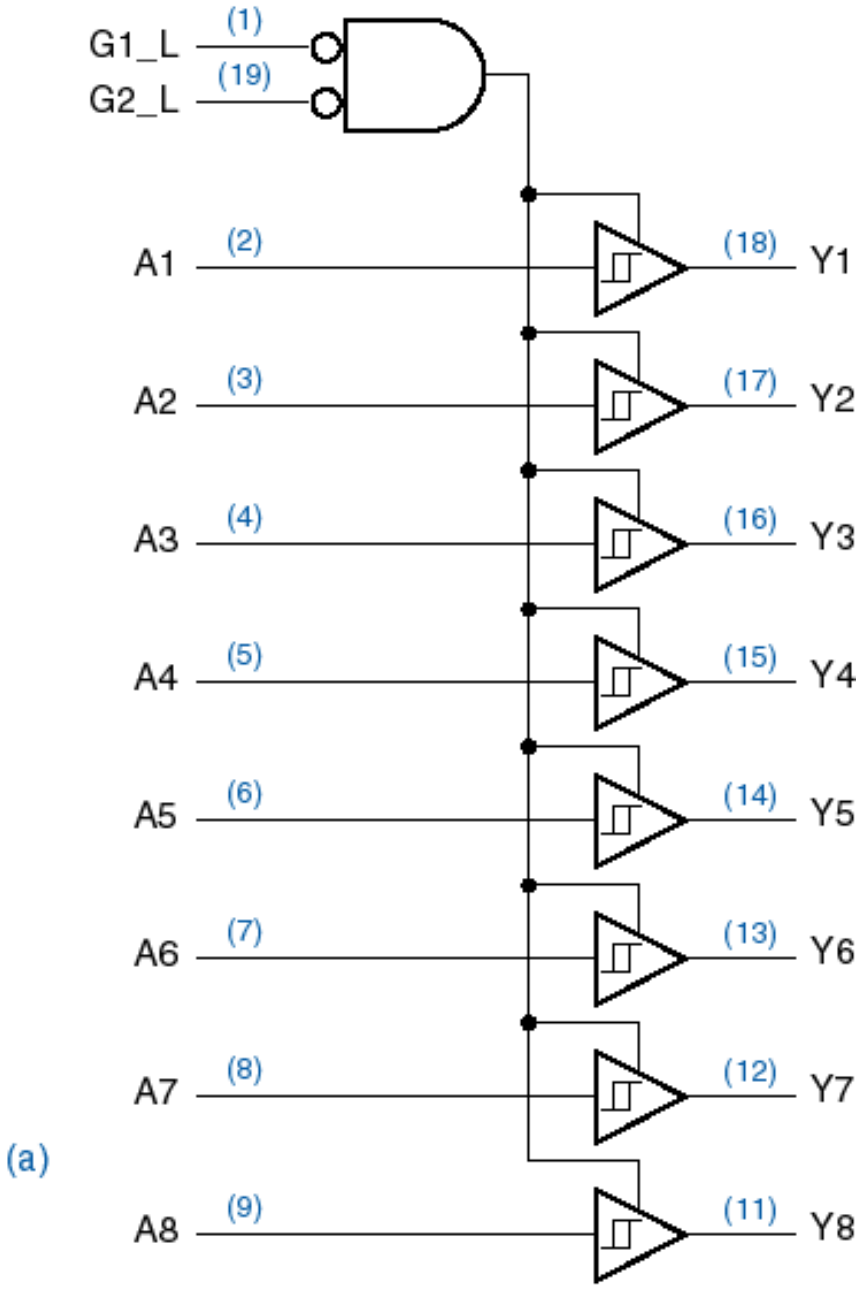
EN	A	OUT
L	L	Hi-Z
L	H	Hi-Z
H	L	L
H	H	H

- Várias saídas podem ser ligadas entre si, no entanto só uma delas pode estar activa.

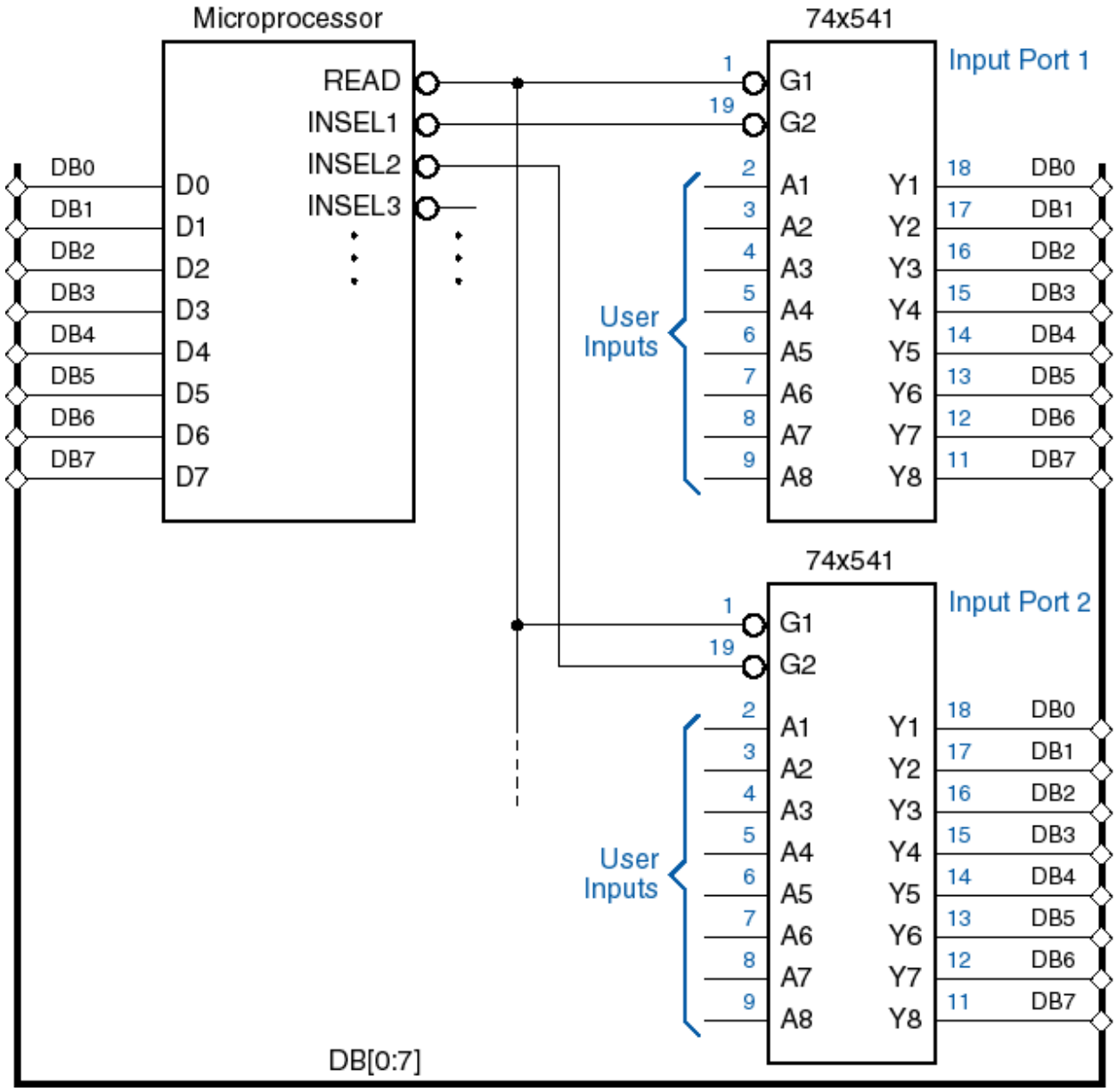
Aplicação Z-Buffers



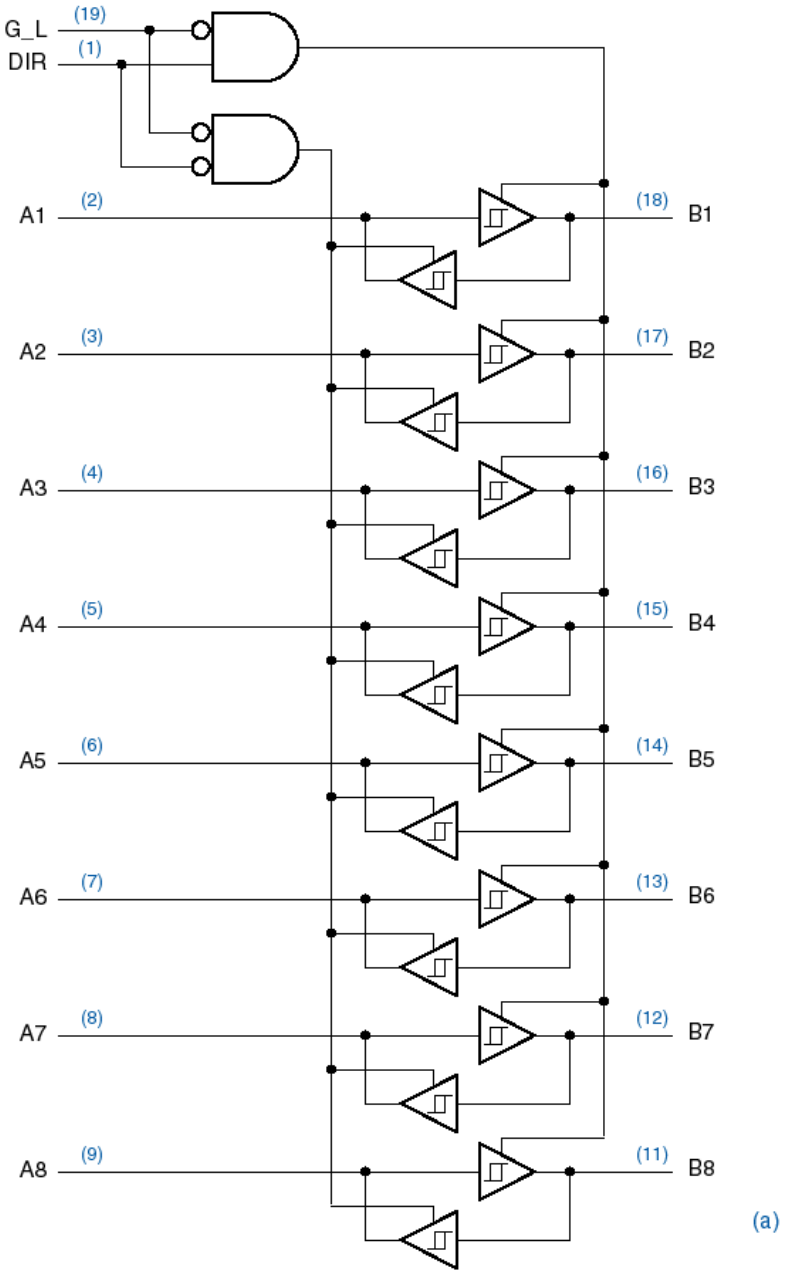




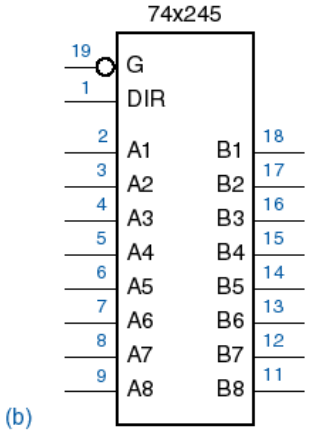
“Drivers” 3-estados



Aplicação de Drivers

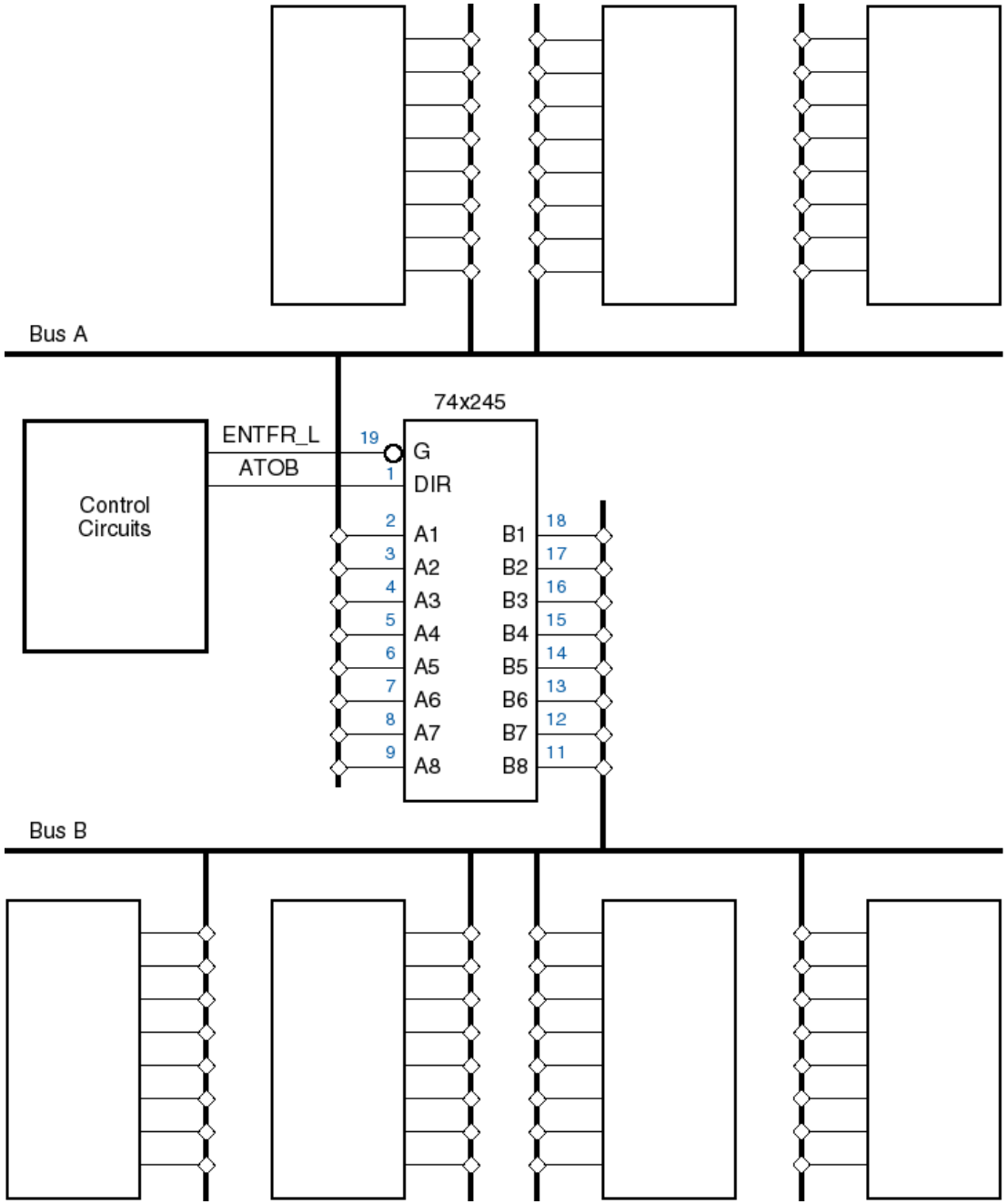


(a)



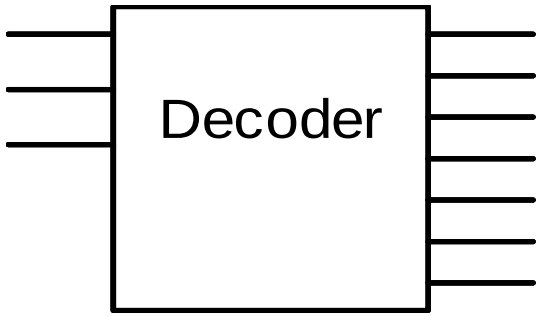
(b)

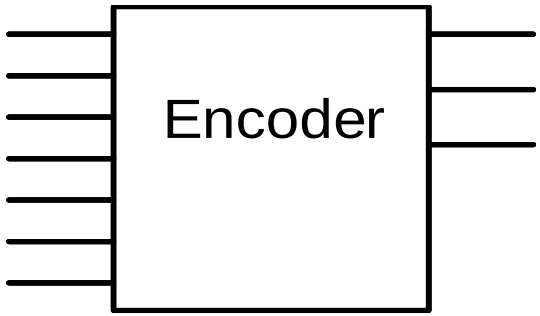
“Transceiver” 3 - estados



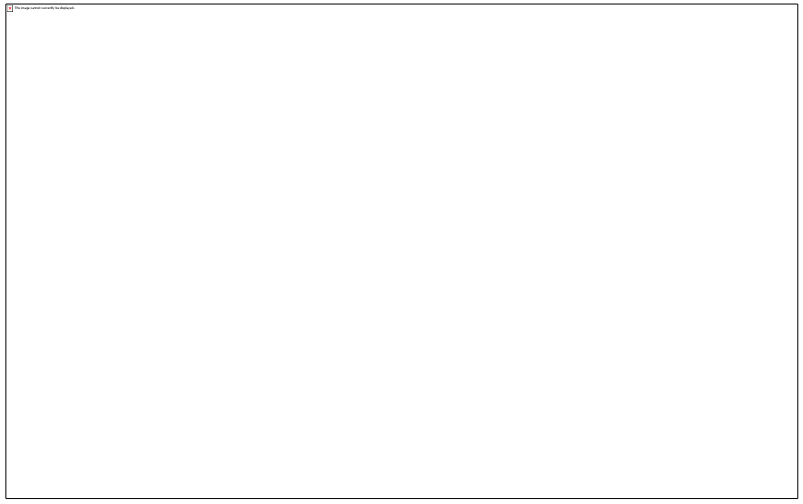
Aplicação de Transceivers

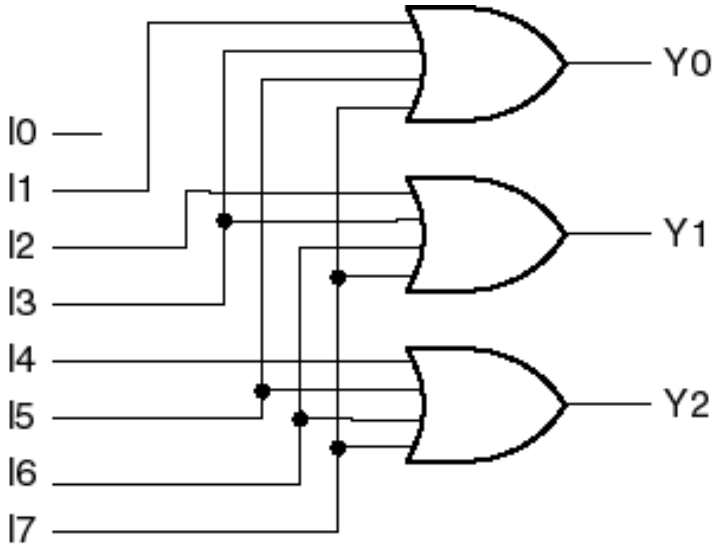
Encoders vs. Decoders



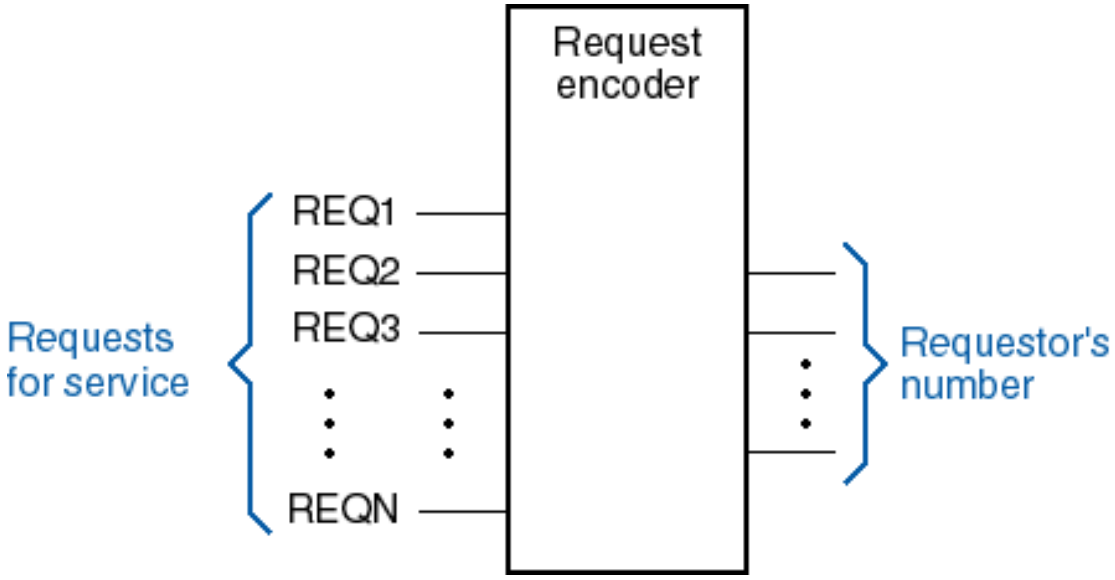


Binary encoders

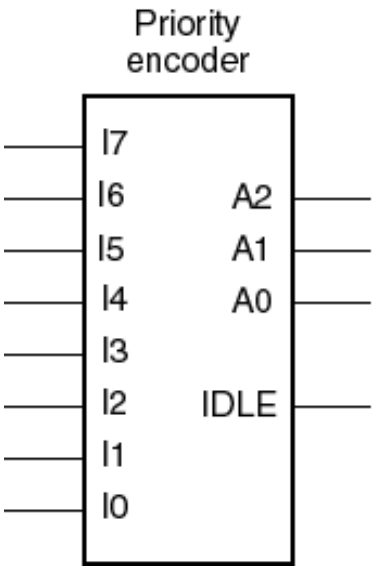




Muitas aplicações necessitam de sistema de definição de prioridades



Encoder de prioridade de 8 entradas



Equações lógicas dum “Priority-encoder”

$$H7 = I7$$

$$H6 = I6 \cdot I7'$$

$$H5 = I5 \cdot I6' \cdot I7'$$

$$\ldots$$

$$H0 = I0 \cdot I1' \cdot I2' \cdot I3' \cdot I4' \cdot I5' \cdot I6' \cdot I7'$$

$$A2 = H4 + H5 + H6 + H7$$

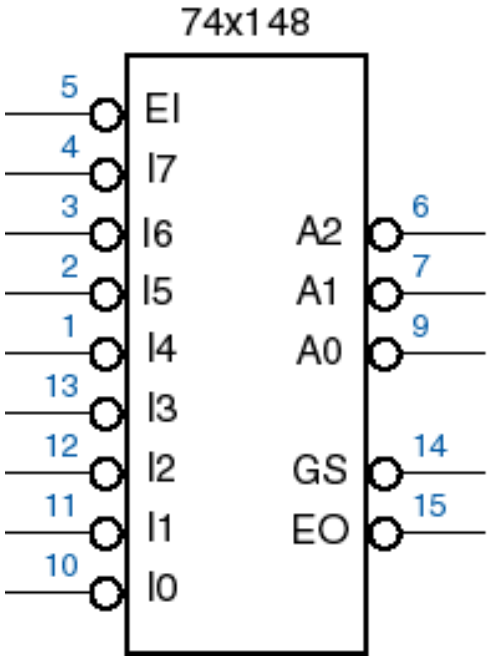
$$A1 = H2 + H3 + H6 + H7$$

$$A0 = H1 + H3 + H5 + H7$$

$$IDLE = (I0 + I1 + I2 + I3 + I4 + I5 + I6 + I7)'$$

$$= I0' \cdot I1' \cdot I2' \cdot I3' \cdot I4' \cdot I5' \cdot I6' \cdot I7'$$

“74x148: 8-input priority encoder”



- Active-low I/O
- Input Enable
- “Got Something”
- Enable Output

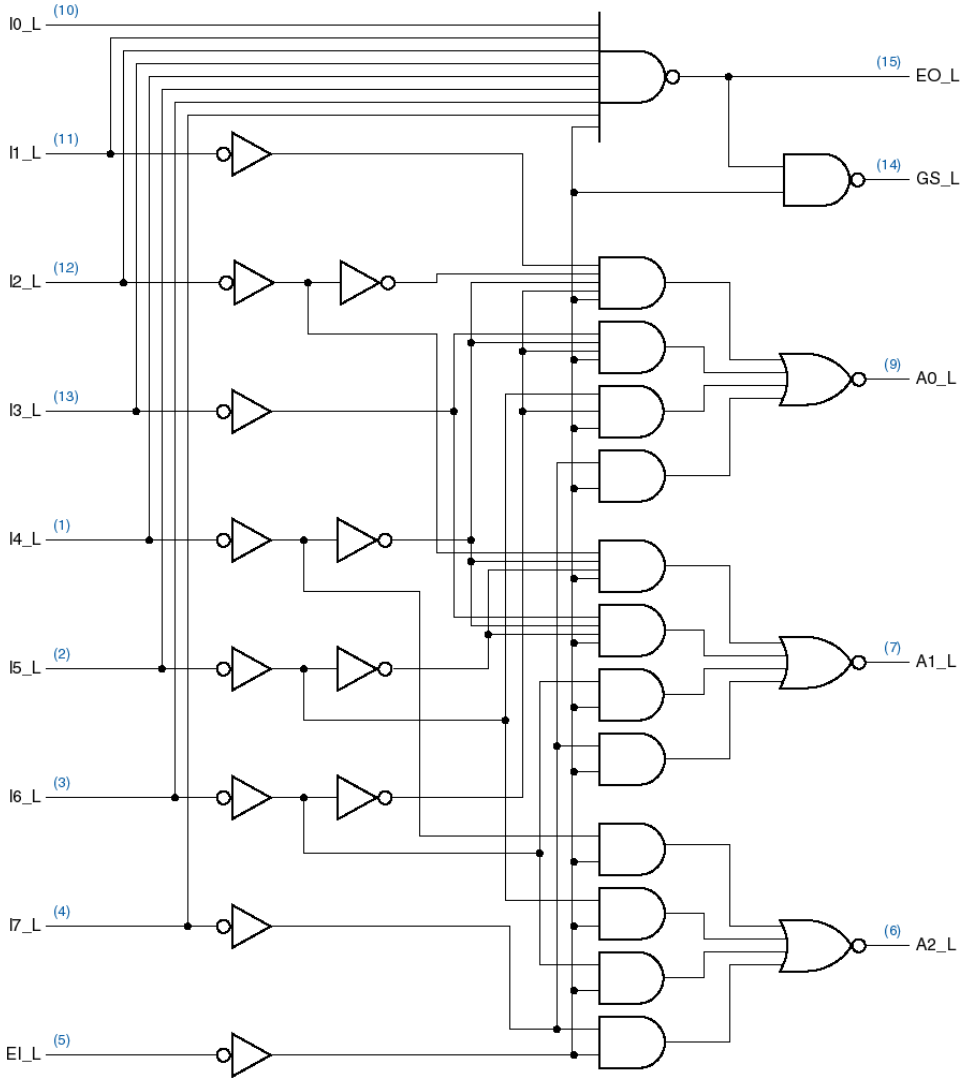
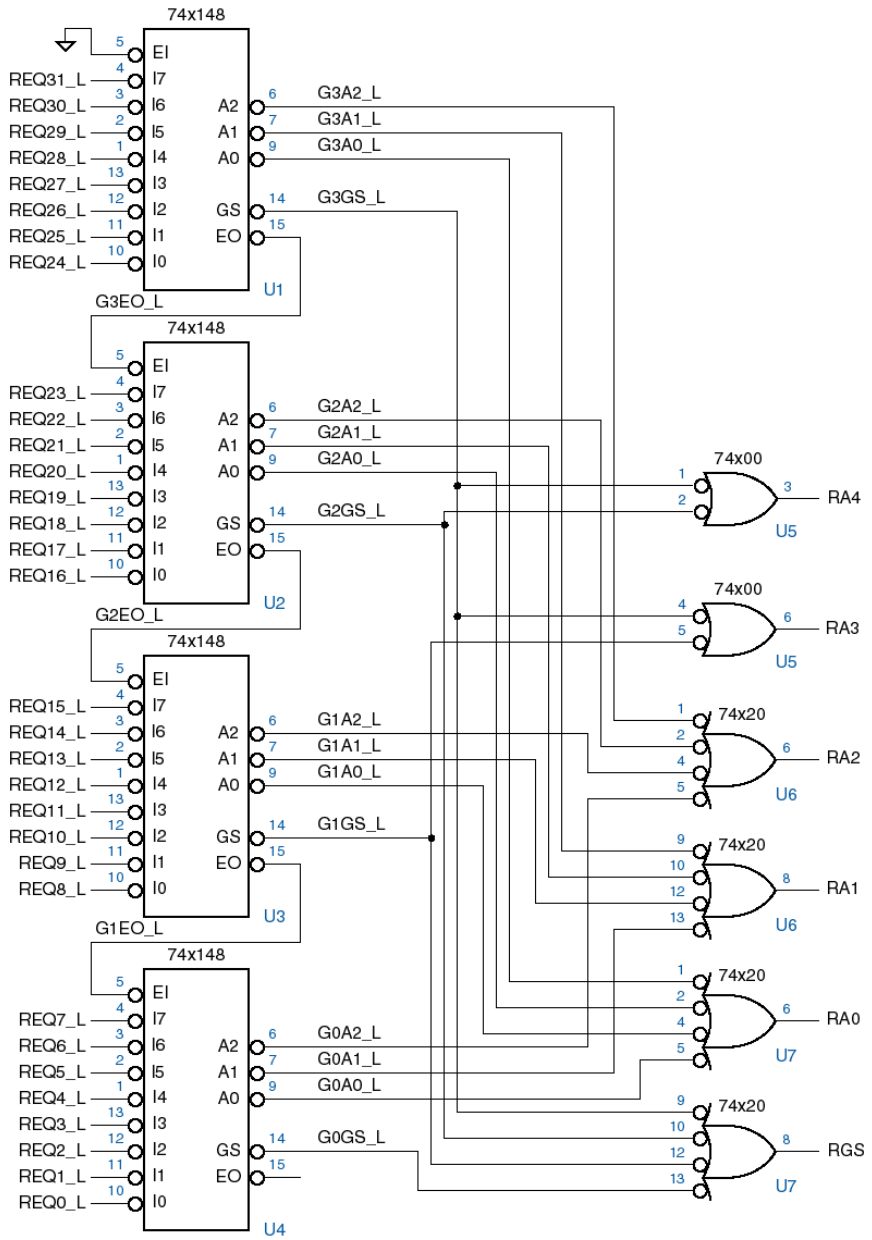


Tabela de Verdade do IC 74x148

Inputs									*Outputs*				
EI_L	I0_L	I1_L	I2_L	I3_L	I4_L	I5_L	I6_L	I7_L	A2_L	A1_L	A0_L	GS_L	EO_L
1	x	x	x	x	x	x	x	x	1	1	1	1	1
0	x	x	x	x	x	x	x	0	0	0	0	0	1
0	x	x	x	x	x	x	0	1	0	0	1	0	1
0	x	x	x	x	x	0	1	1	0	1	0	0	1
0	x	x	x	x	0	1	1	1	0	1	1	0	1
0	x	x	x	0	1	1	1	1	1	0	0	0	1
0	x	x	0	1	1	1	1	1	1	0	1	0	1
0	x	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0

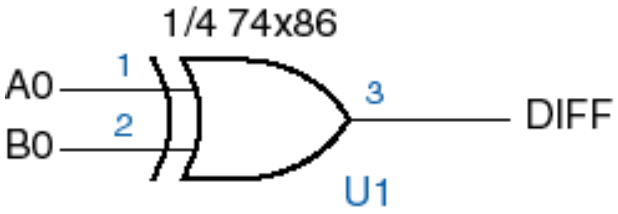


“Priority encoders” em cascata

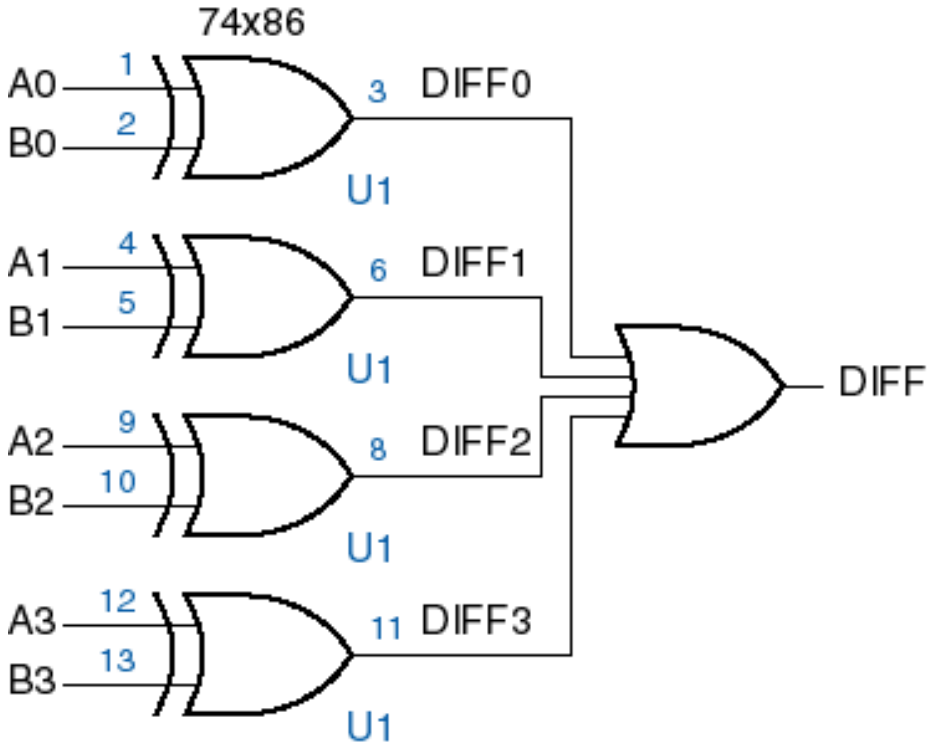
- “32-input priority encoder”

Comparadores de Igualdade

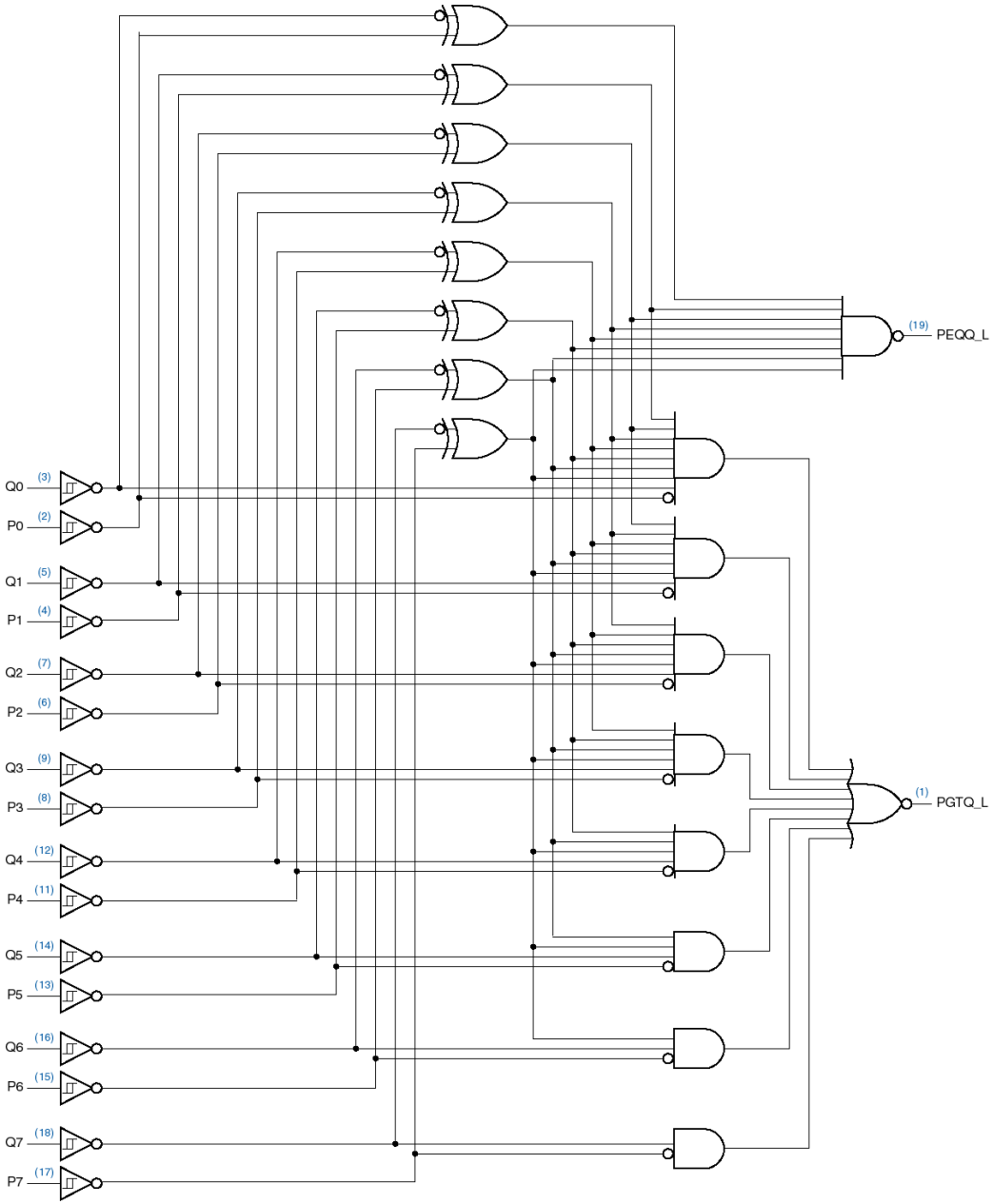
- comparador 1-bit

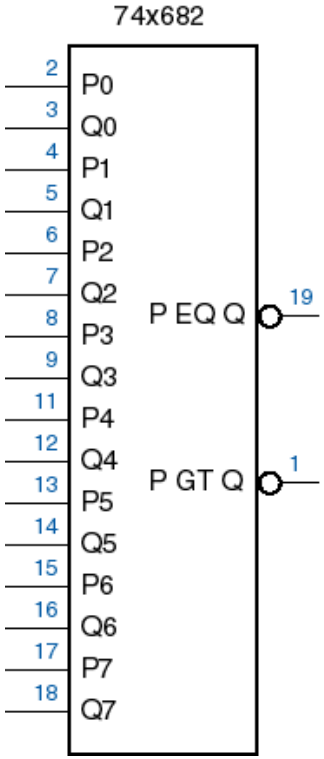


- comparador 4-bit



Comparador 8-bits





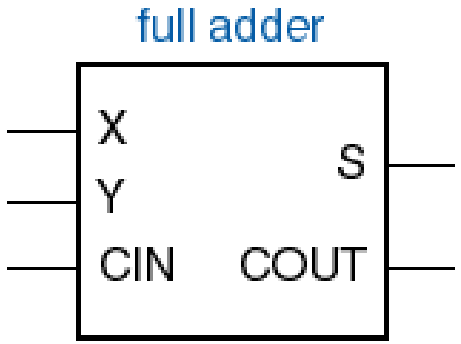
Comparador

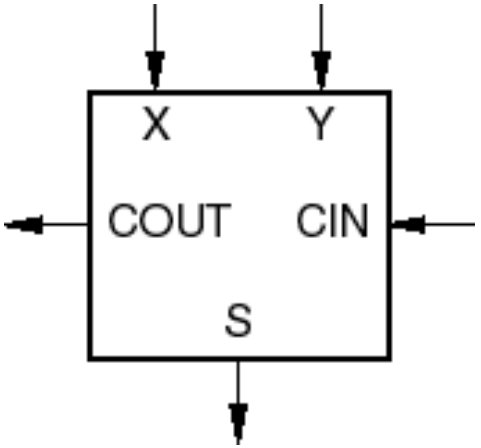
- Verificação Igualdade
  - PEQQ = ([P7..P0] == [Q7..Q0]);
  - PEQQ_L = !([P7..P0] == [Q7..Q0]);
  - 16 termos produto
- Comparação Magnitude
  - PGTQ = ([P7..P0] > [Q7..Q0]);
  - PGTQ_L = !([P7..P0] > [Q7..Q0]);
  - 255 termos produto

Somadores

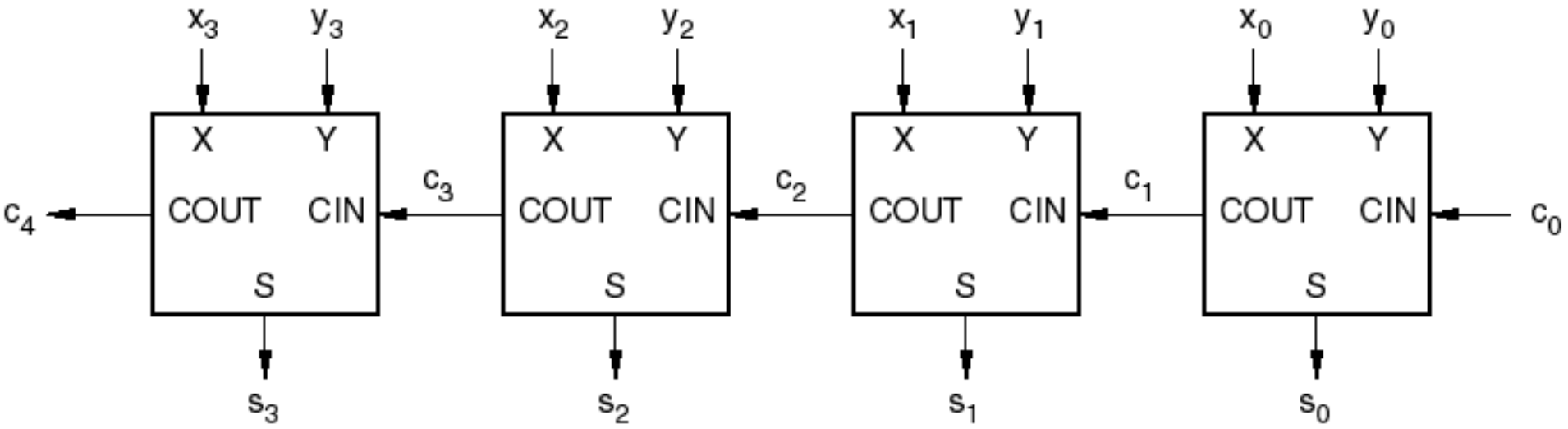
- Bloco básico é denominado “full adder”
  - Somador de 1-bit, produz soma e saídas carry
- Tabela de Verdade:

X	Y	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

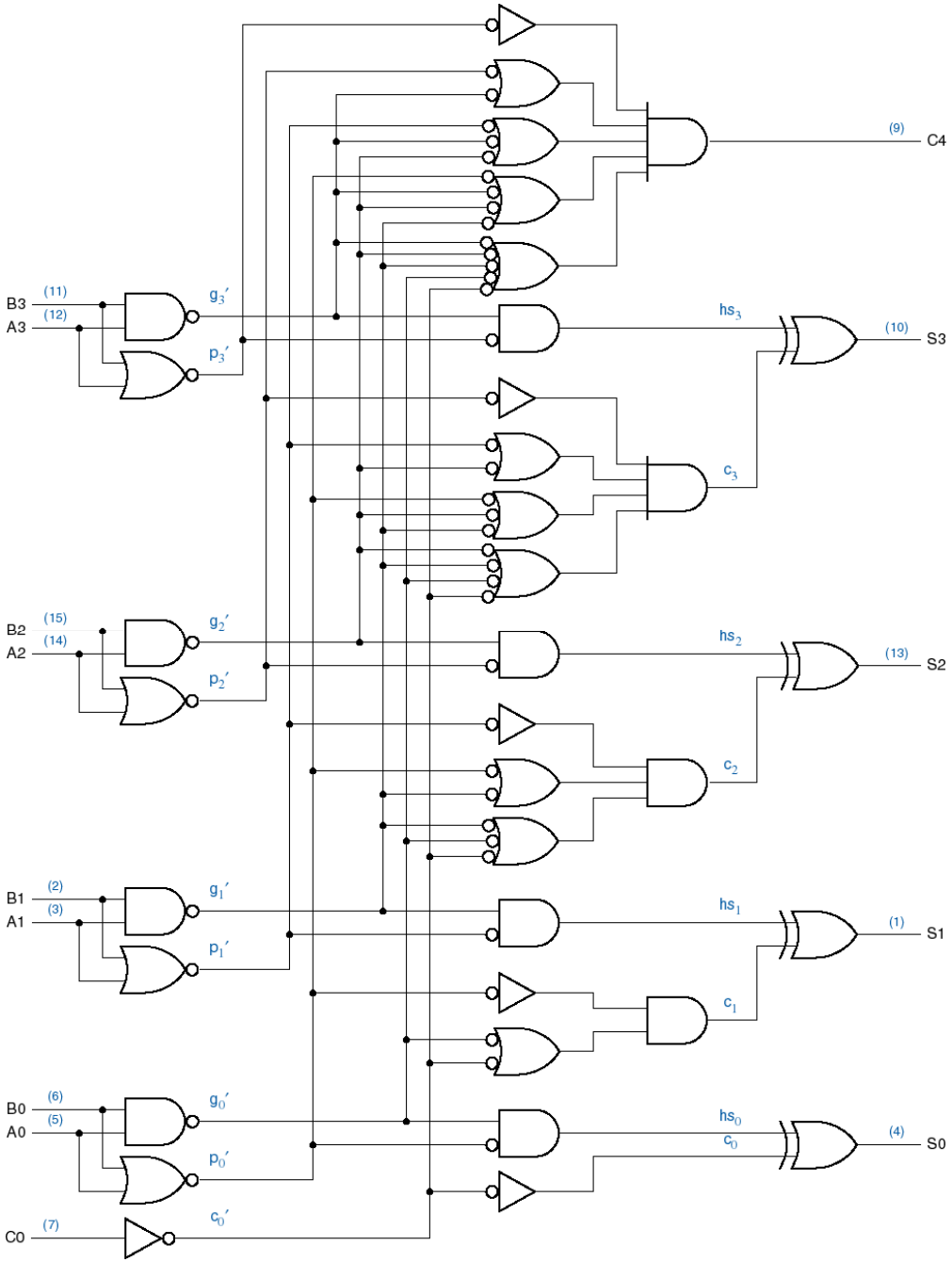




Somador “Ripple adder”

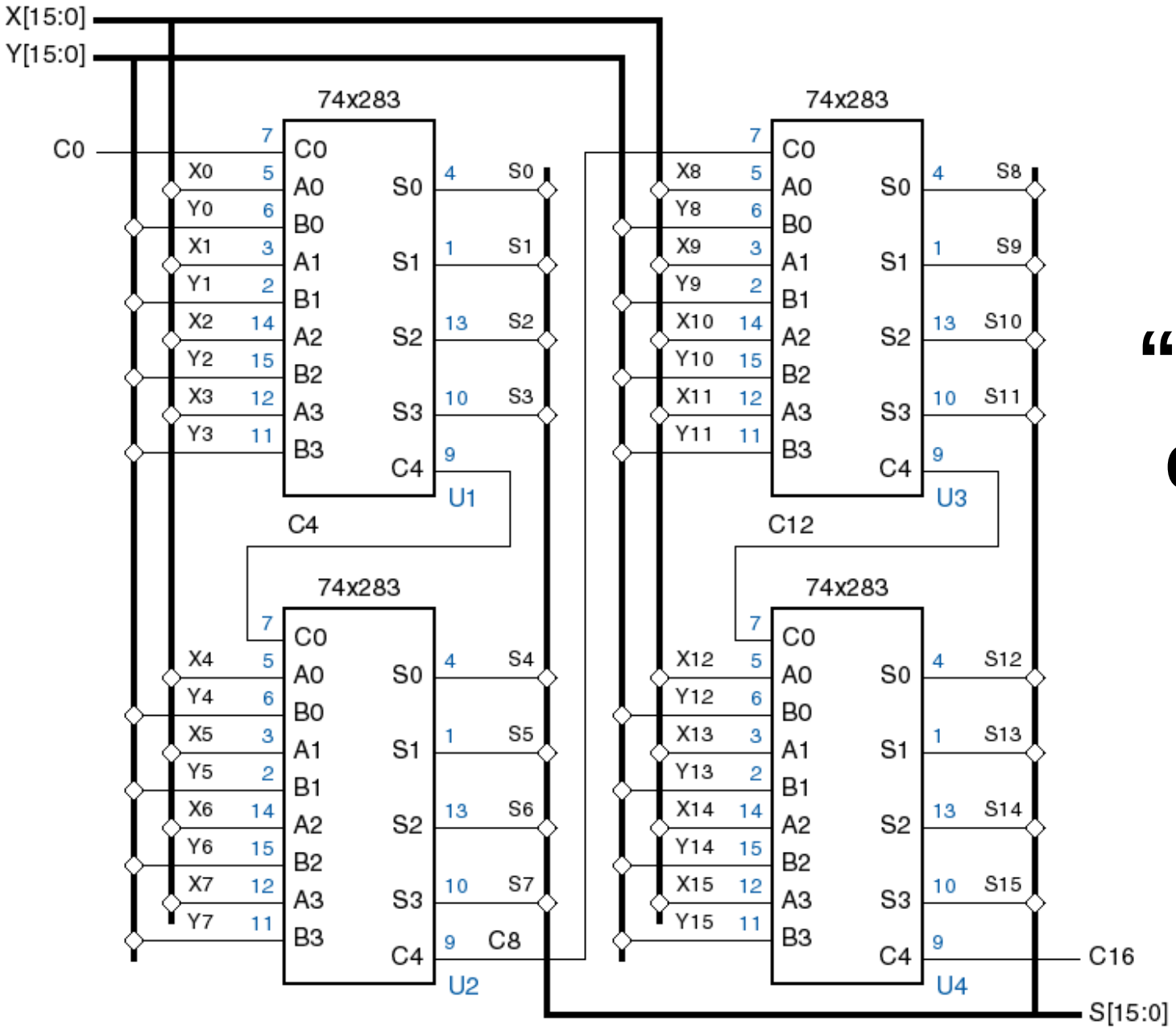


- Velocidade limitada pelo tamanho da cadeia



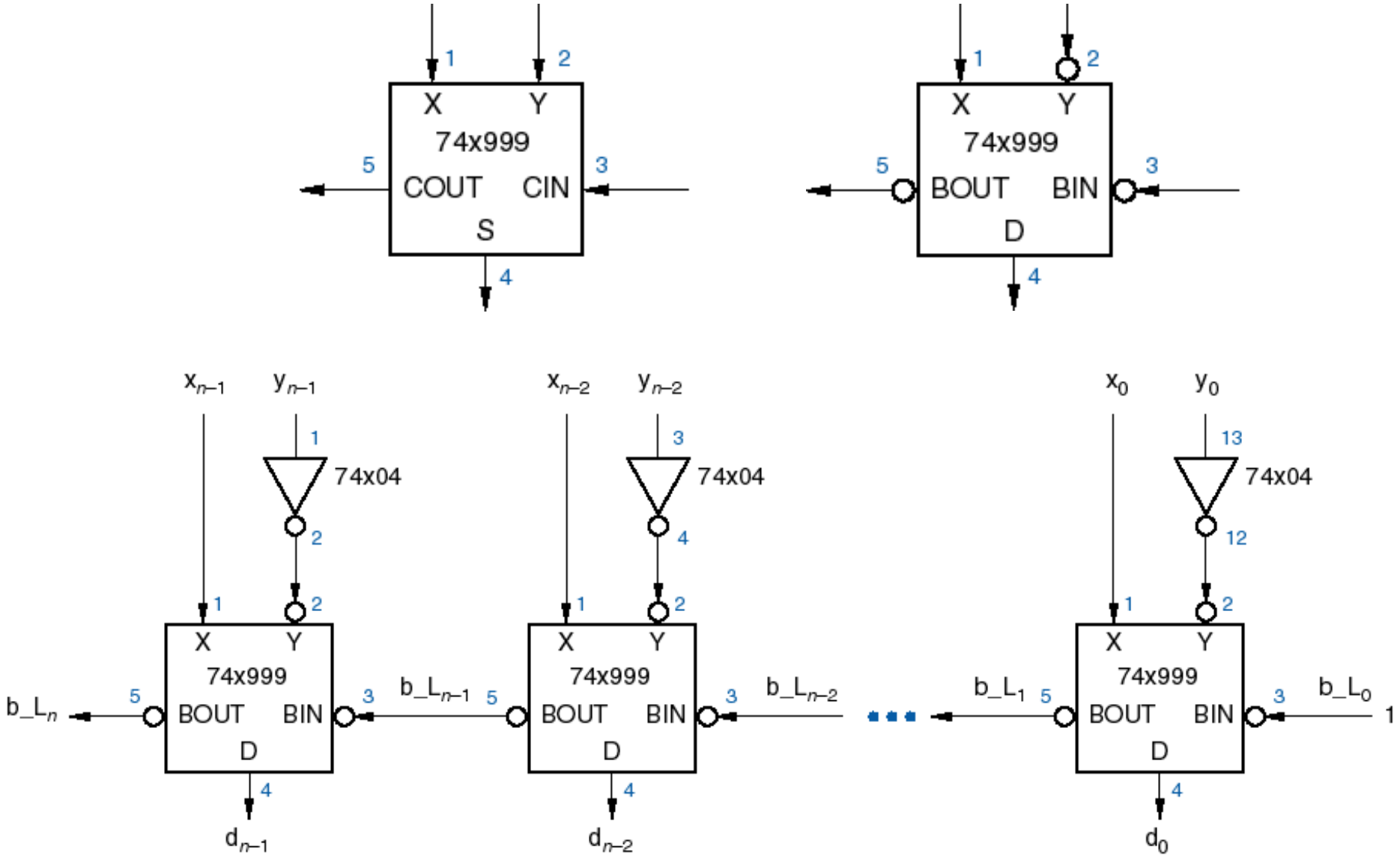
74x283 Somador 4-bit

- Usa “carry lookahead” internamente



“Ripple carry” entre grupos

“Full subtractor” = “full adder”, quase



Multiplicador

- multiplicador 8x8