Universidade Federal do Rio Grande do Norte
Instituto Metrópole Digital
IMD0121 –Arquitetura de Computadores

Circuitos Combinacionais

Prof. Gustavo Girão
girao@imd.ufrn.br

Roteiro

- Definição de circuitos combinacionais

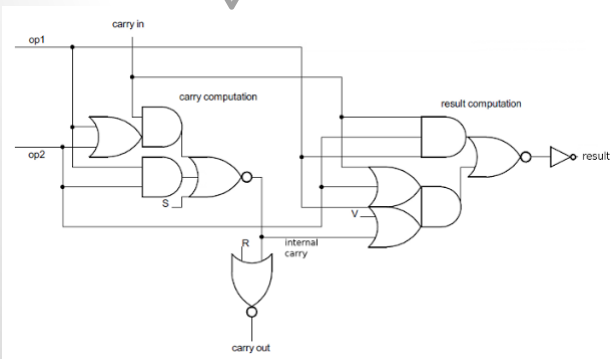
- Exemplos de circuitos combinacionais
  - Somadores
  - Multiplexadores
  - Comparadores
  - Deslocadores

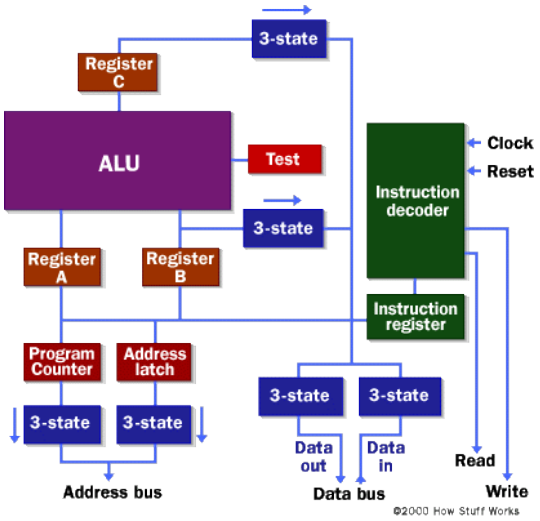
Circuitos Combinacionais

- Conhecemos portas lógicas básicas e portas lógicas compostas
  - AND, OR, NOT
  - NAND, NOR, XOR, XNOR
- Todos os exemplos que vimos até agora podem ser caracterizados como circuitos **combinacionais**
  - Circuitos simples de entender e com um fluxo de “dados“ unidirectional.
- Definição
  - **Um circuito digital cuja saída depende apenas dos valores atuais das entradas.**

Exemplos de Circuitos Combinacionais

- São utilizados como componentes basicos (acima das portas lógicas) na realização de tarefas maiores
- Um processador, por exemplo, pode ser visto como uma agregação destes componentes básicos





Exemplos de Circuitos Combinacionais

- Circuito Somador
- Comparador
- Multiplexador
- ULA

Circuito Somador

- Função
  - Adicionar dois números binários é talvez a operação mais comum executada nos sistemas digitais.
- Um somador de **N-bit** é um componente combinacional que adiciona **duas entradas** (A e B) de **N-bit** e gera uma **saída** de **N-bit**.
- **N** é a largura do somador.
  - Também chamada de **resolução**, porém é um termo mais usado em sistemas analógicos.
- Projetar somadores de forma eficiente (tamanho, velocidade) é um campo de pesquisa que tem recebido considerável atenção por muitas décadas.

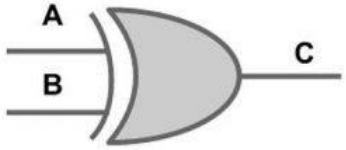
Começando do inicio

- Realizar a soma de dois bits é uma tarefa simples e já vimos como realizá-la manualmente

$$\begin{array}{r} 0111 \\ +\ 0110 \\ \hline 1101 \end{array}$$

- Vimos também a porta XOR que tem um comportamento similar a uma soma:

PORTA OU EXCLUSIVO (XOR) $C=A \oplus B$



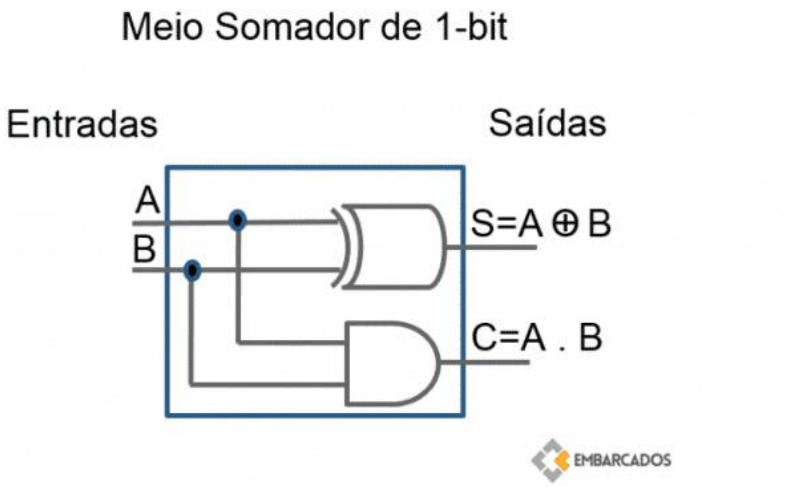
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

Começando do inicio

- Mas existe algo faltando….
- E o Carry?

```
  11
  0111
+ 0110
------
 1101
```

- A porta lógica XOR não é suficiente para representar uma soma de um bit. Precisamos de algo a mais:



Circuito Somador

- Adicionar dois números de 2 bits
  - 01 + 11 = 100 (1 + 3 = 4)
- Podemos implementar a lógica desse circuito usando mintermos ou maxtermos
  - Abordagem ineficiente!!!!
  - Por que?

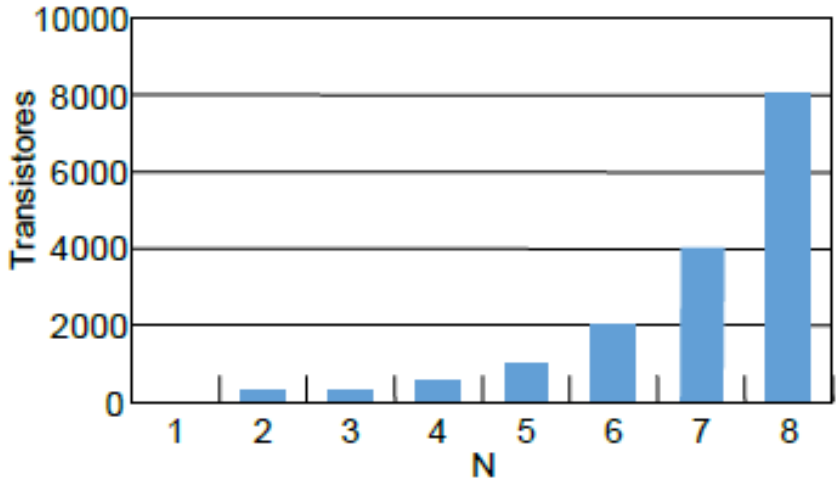
Entradas				Saídas		
a1	a0	b1	b0	c	s1	s0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

Circuito Somador

O tamanho da tabela da verdade cresce exponencialmente

- Somador de 2-bit: $2^{2+2} = 16$ linhas
- Somador de 4-bit: $2^{4+4} = 256$ linhas
- Somador de 8-bit $2^{8+8} = 65.536$ linhas
- Somador de 16-bit $2^{16+16} =$ 4 bilhões de linhas

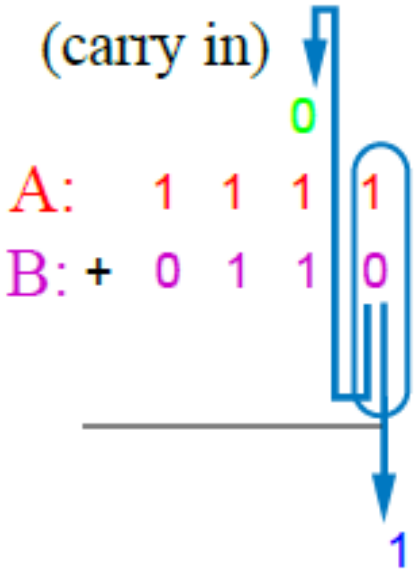
Circuito Somador



- N=5 $\rightarrow$ 1000 transistores
- Número de transistores dobra a cada aumento de N
- Qtd. transistores = $1000 \times 2^{N-5}$
  - N = 16 $\rightarrow$ 2.048.000 transistores
  - N = 32 $\rightarrow$ 100 bilhões de transistores para somar 2 números?

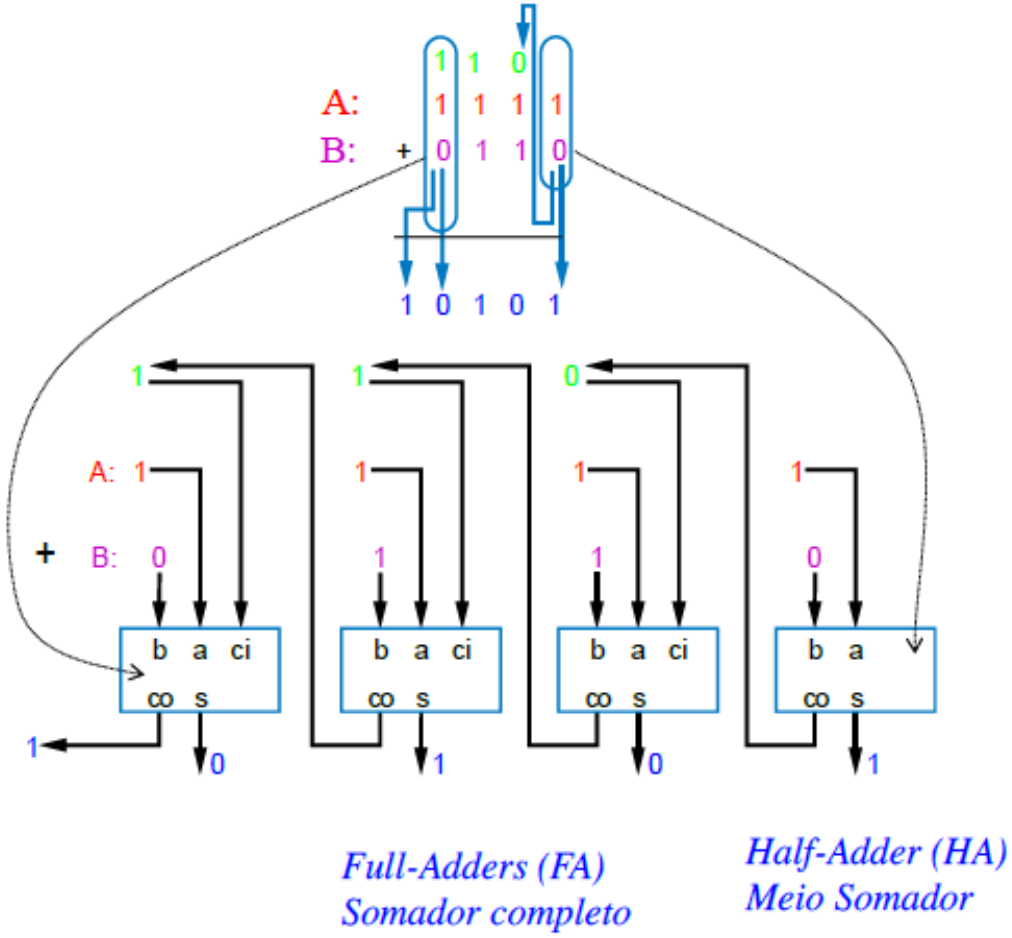
Modo alternativo de implementar somadores

- Imitar como as pessoas fazem a operação de adição manualmente



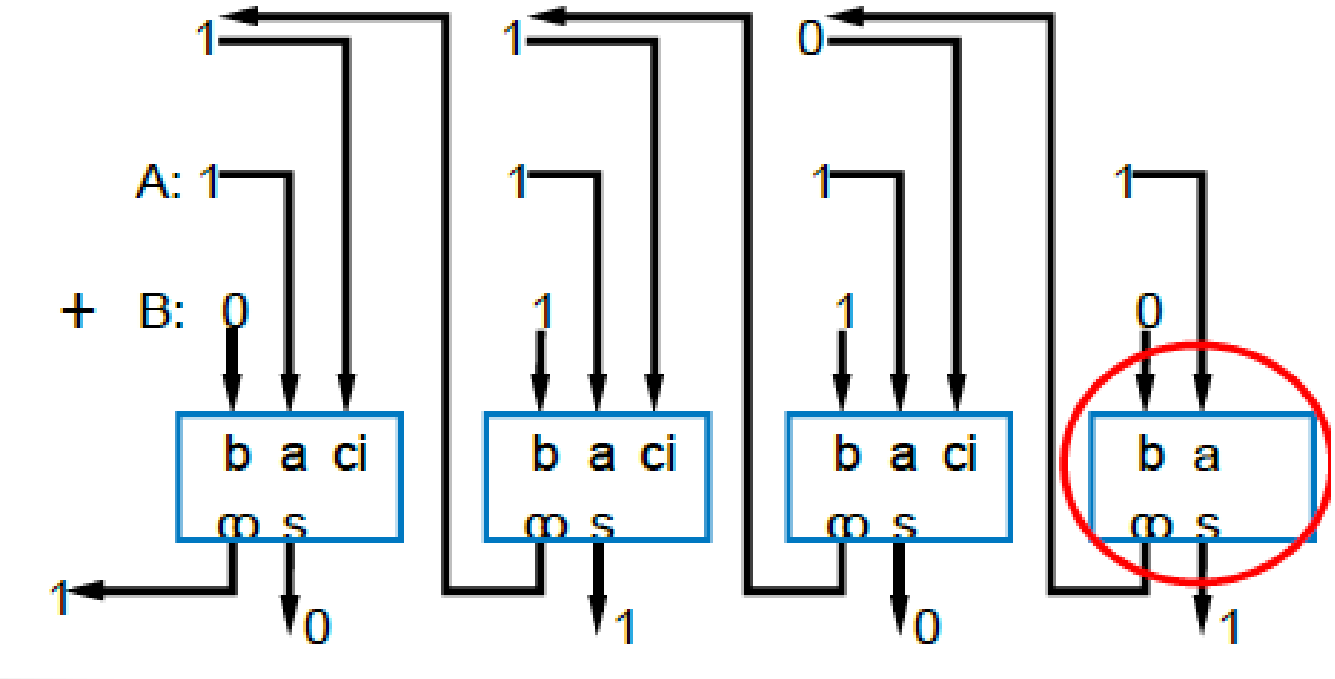
Modo alternativo de implementar somadores

- Criar um componente para cada coluna



Meio Somador (Half-Adder HA)

- Adiciona dois bits, gera a soma (s) e o carry out (co)
- Projeto é implementado na lógica combinacional clássica (mintermo ou maxtermo)



Fluxo de concepção do HA

- 1. Encontre a tabela da verdade

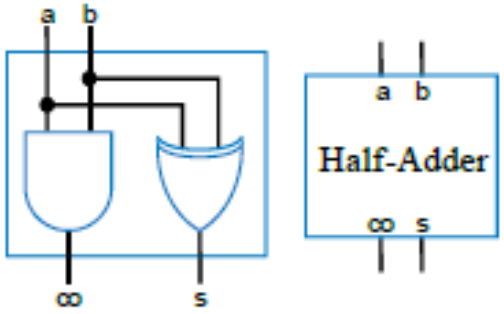
Entradas		Saídas	
a	b	co	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

- 2. Converter as saídas da tabela da verdade em equações

$$co = a.b$$

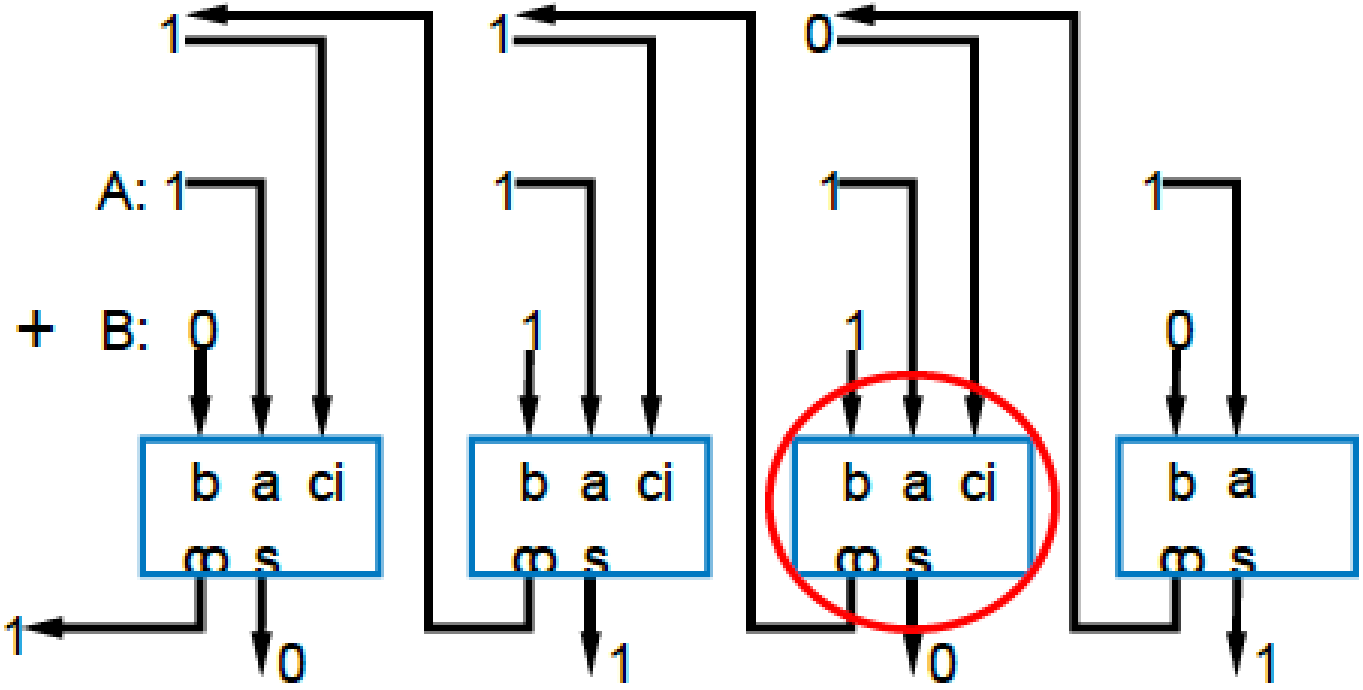
$$s = a \oplus b$$

- 3. Criar o circuito



Somador Completo (Full Adder – FA)

- Adiciona três bits, gera a soma (s) e o carry out (co)
- Projeto é implementado na lógica combinacional clássica (mintermo ou maxtermo)



Fluxo de concepção do FA

- 1. Encontre a tabela da verdade

Entrada			Saída	
a	b	ci	co	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

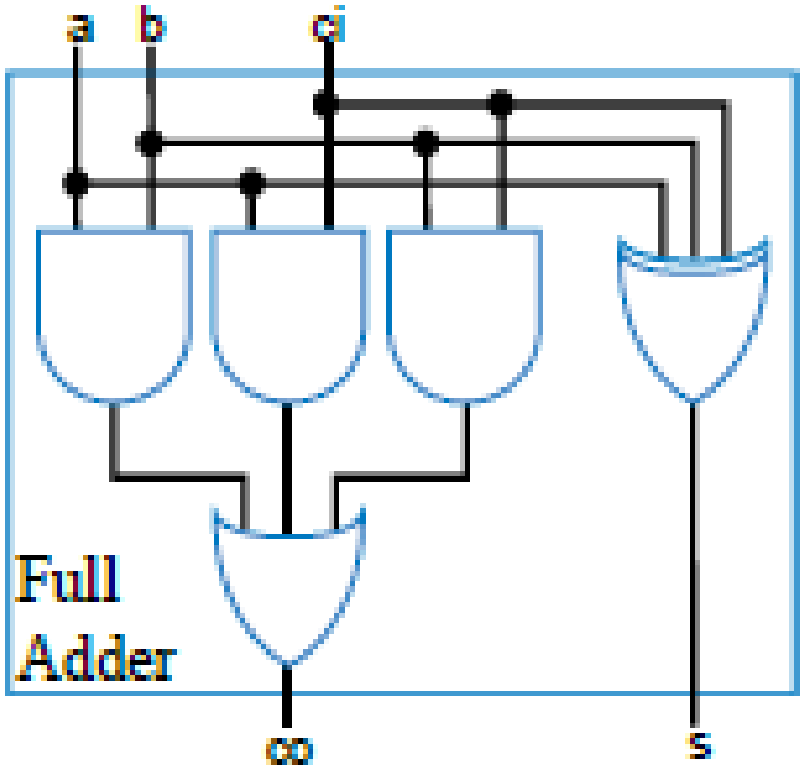
- 2. Converter as saídas da tabela da verdade em equações

$$co = b.ci + a.ci + a.b$$

$$s = a \oplus b \oplus ci$$

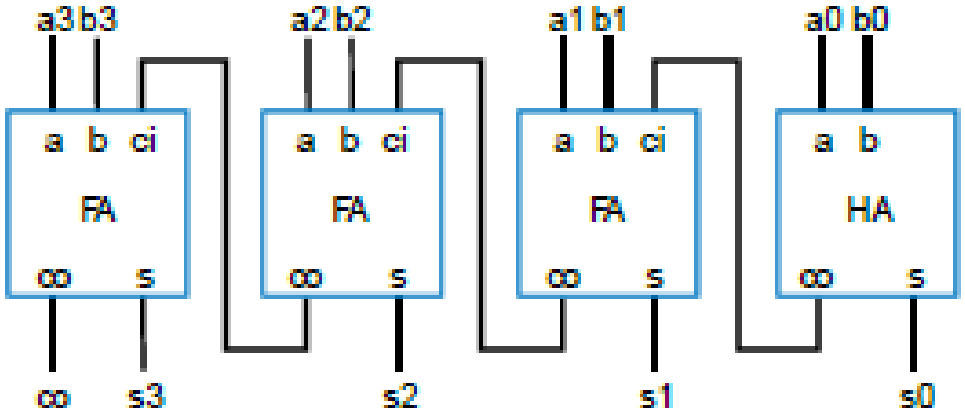
Fluxo de concepção do FA

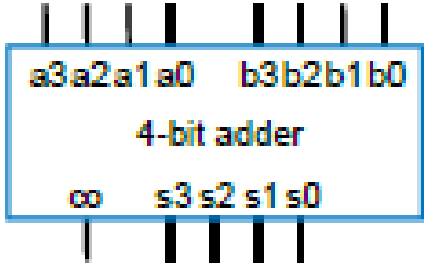
- 3. Criar o circuito



Somador com propagação do carry

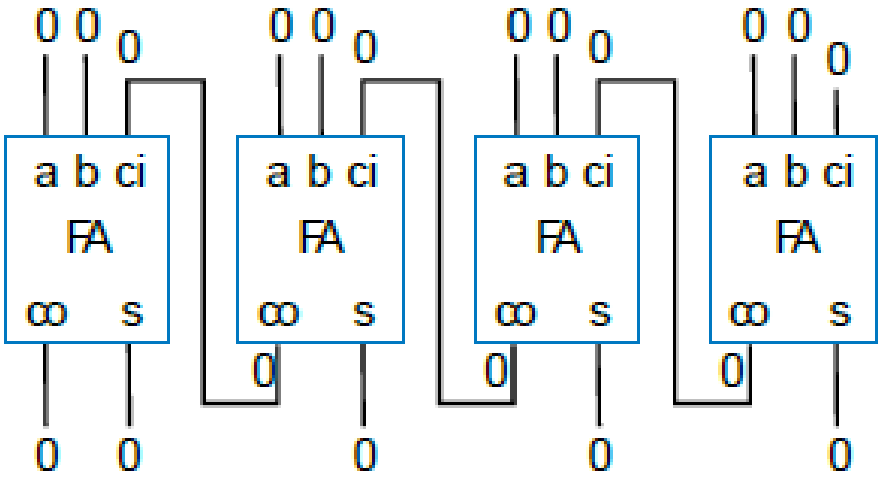
- Somador utilizando FA e HA (como se a soma fosse realizada à mão)
- Pode facilmente construir somadores de qualquer tamanho
  - Somador de 4 bits (2 palavras de 4-bit)
  - Gera uma saída de 4 bits e o bit de carry out





Atraso de porta

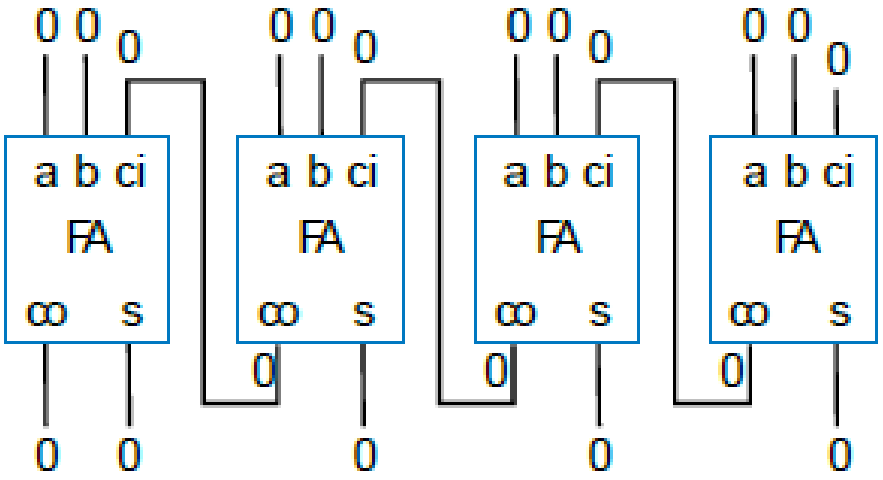
- Quais são as consequência de uma composição de elementos como esta?
- A maior delas é o acumulo do atraso
  - Que já existe em qualquer simples porta lógica



Assumir que todas as entradas são zero inicialmente

Atraso de porta

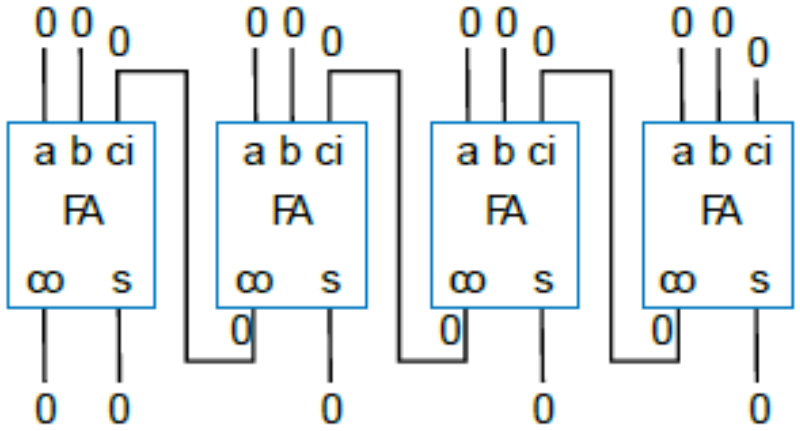
- Quais são as consequência de uma composição de elementos como esta?
- A maior delas é o acumulo do atraso
  - Que já existe em qualquer simples porta lógica
  - Vamos imaginar que cada FA abaixo, leva **2 nanosegundos (ns)** para realizer a soma de um bit (gerando o carry out)



Assumir que todas as entradas são zero inicialmente

Atraso de porta

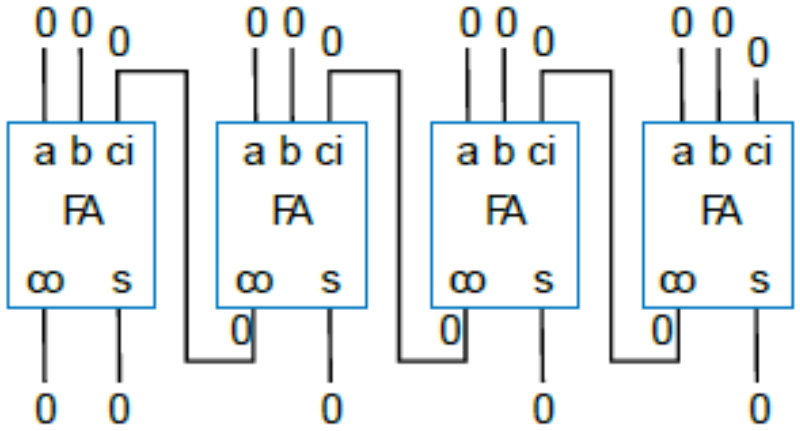
MOMENTO 1



Assumir que todas as entradas são zero inicialmente

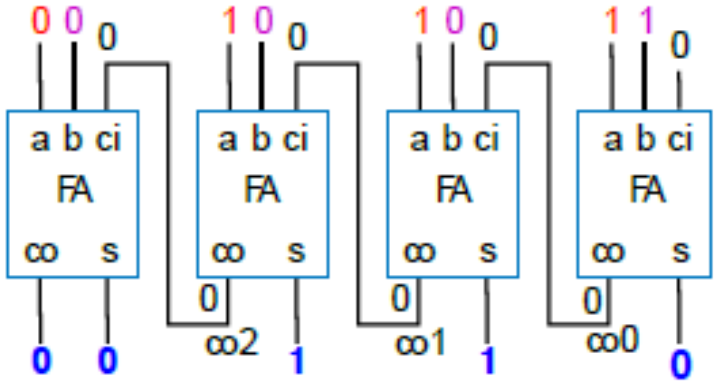
Atraso de porta

MOMENTO 1



Assumir que todas as entradas são zero inicialmente

MOMENTO 2

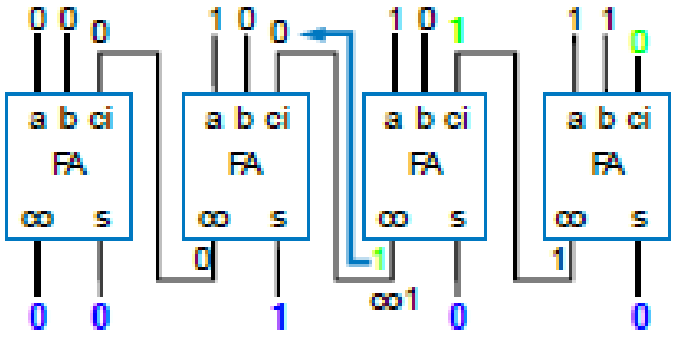


0111+0001
(reposta deveria ser 01000)

Saída após 2 ns (1 FA)

Atra

MOMENTO 3

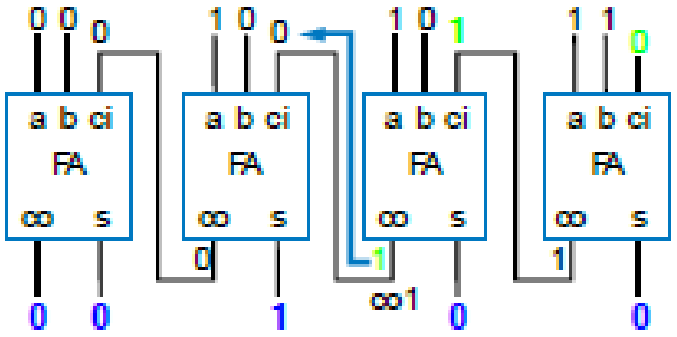


0111+0001
(saída deverla ser 01000)

Saída após 4ns (2 FA)

Atra

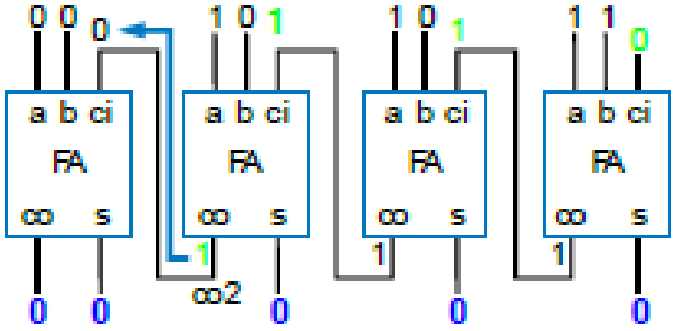
MOMENTO 3



0111+0001
(saída deverla ser 01000)

Saída após 4ns (2 FA)

MOMENTO 4

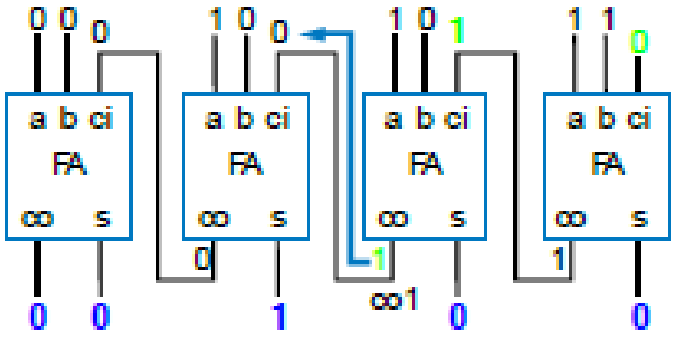


Saída após 6ns (3 FA)

Atra

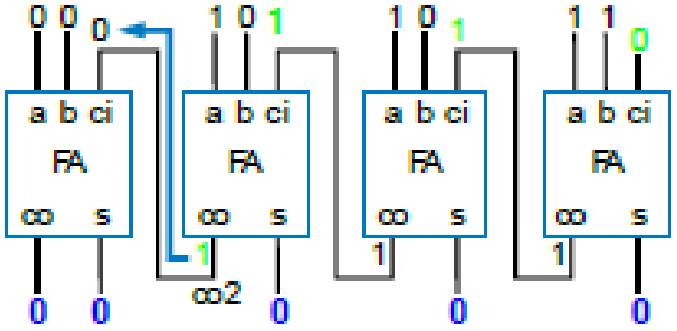
0111+0001
(saída deveria ser 01000)

MOMENTO 3



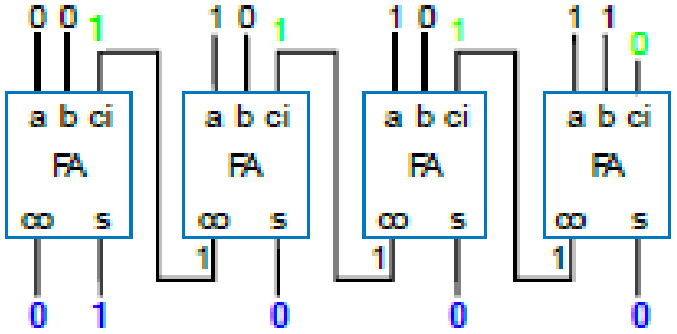
Saída após 4ns (2 FA)

MOMENTO 4



Saída após 6ns (3 FA)

MOMENTO 5

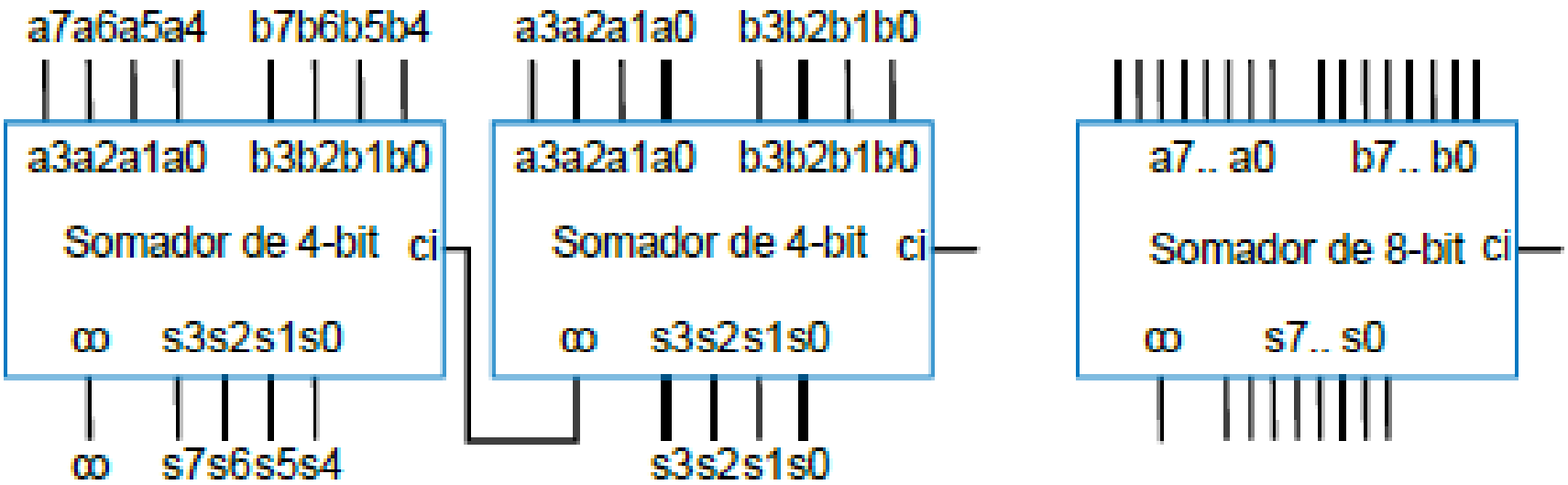


Saída após 8ns (4 FA)

Saída correta após 8ns

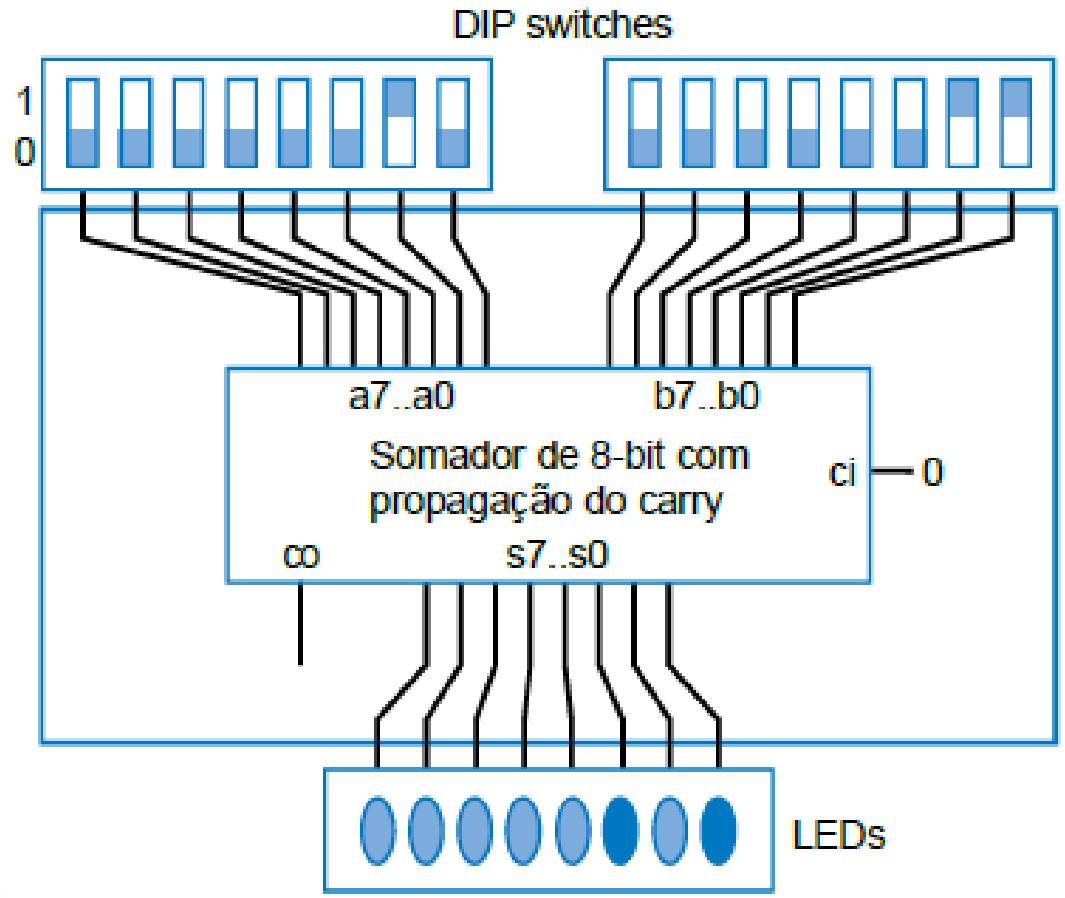
Cascateamento de somadores

- Como eu realizo a soma de 8 bits utilizando o somador de 4 bits anterior?



Somador

- Calculadora de 8-bits



Temporalidade

- No exemplo do somador, a temporalidade foi um fator importante na consolidação do valor final
- Externamente observamos que leva um tempo para o valor final “estabilizar”
- É importante notar também que no circuito do somador não existia nenhum componente que marcasse a **sincronização temporal** entre os outros componentes
- Assim, este componente é dito ASSINCRONO.

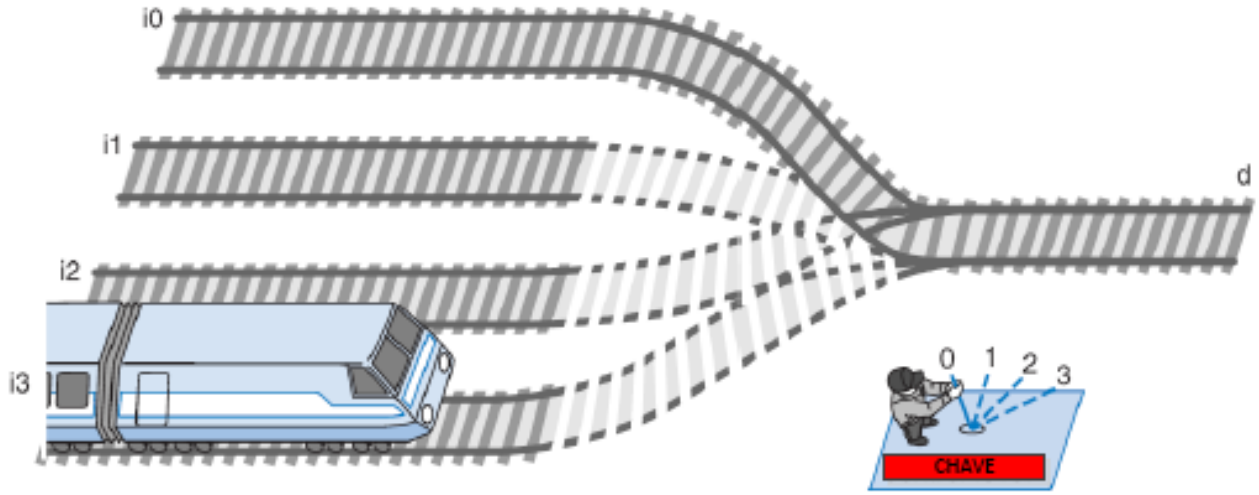
Multiplexador

- Outro bloco construtivo de nível mais elevado usado em circuitos digitais.
- Um MUX M x 1 possui **M entradas** de dados e **uma saída**.
- Permite que apenas uma das entradas seja passada para saída.
- Uma chave selecionadora informa qual entrada irá passar na saída.
- Mux de 4 entradas -> chave de duas posições
- Mux de 8 entradas -> chave de três posições
- Mux de M entradas -> chave de _____ posições

Multiplexador

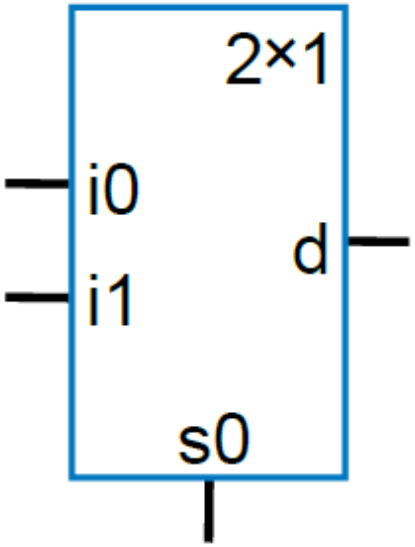
- Outro bloco construtivo de nível mais elevado usado em circuitos digitais.
- Um MUX M x 1 possui **M entradas** de dados e **uma saída**.
- Permite que apenas uma das entradas seja passada para saída.
- Uma chave selecionadora informa qual entrada irá passar na saída.
- Mux de 4 entradas -> chave de duas posições
- Mux de 8 entradas -> chave de três posições
- Mux de M entradas -> chave de $\log_2 M$ posições

Multiplexador (MUX)

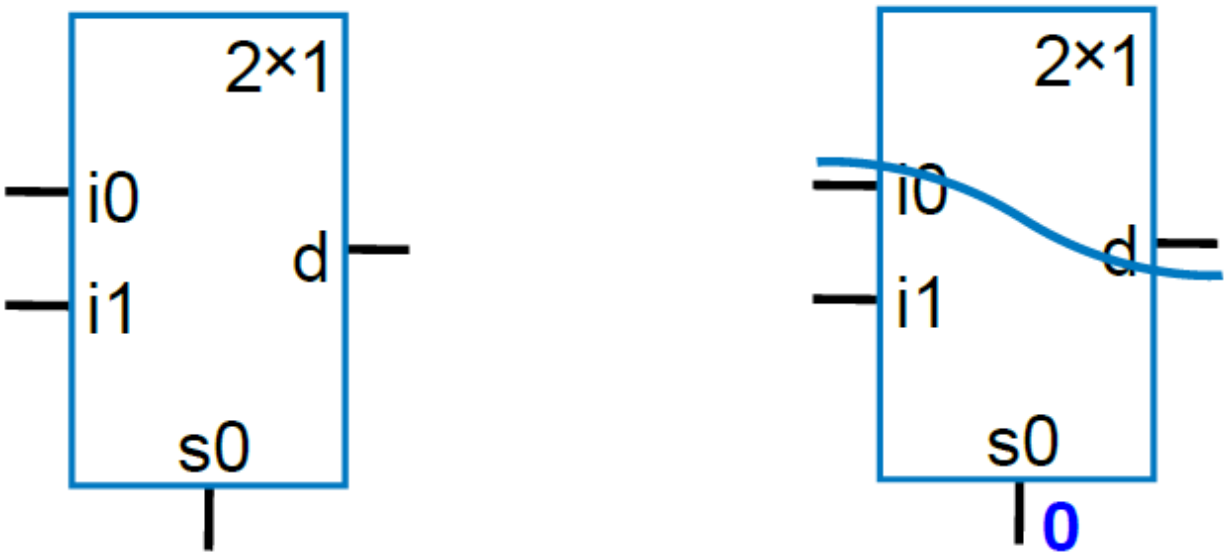


- Um multiplexador é como uma **aparelho de mudança de via** em um parque ferroviário de manobras.
- Ele determina qual via de entrada será conectada à única via de saída, de acordo com a alavanca de controle.

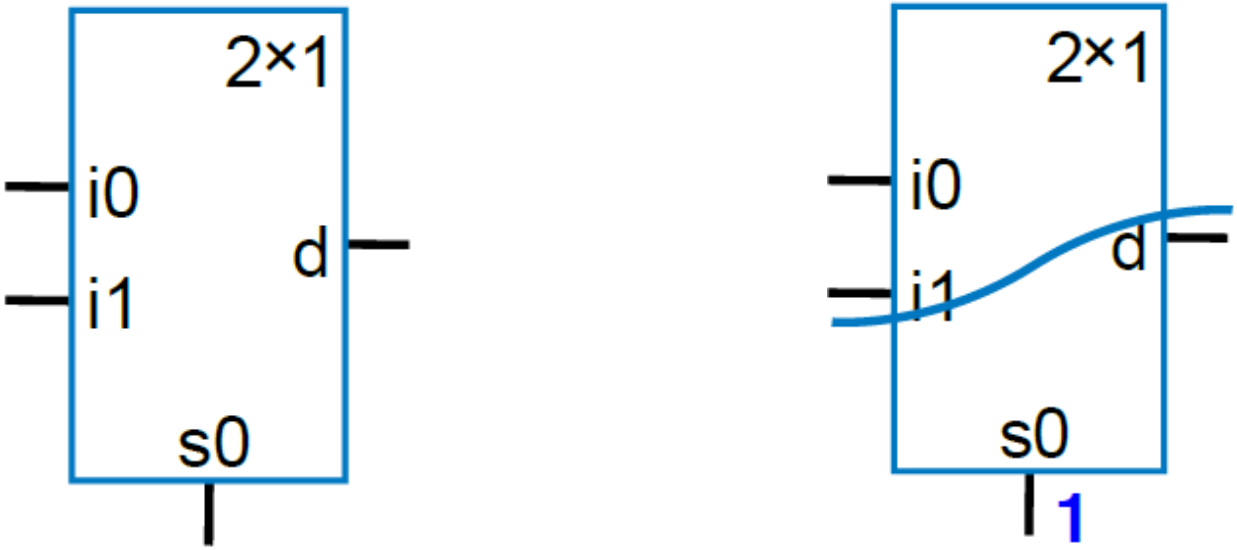
Multiplexador 2x1



Multiplexador 2x1



Multiplexador 2x1



Multiplexador 2x1

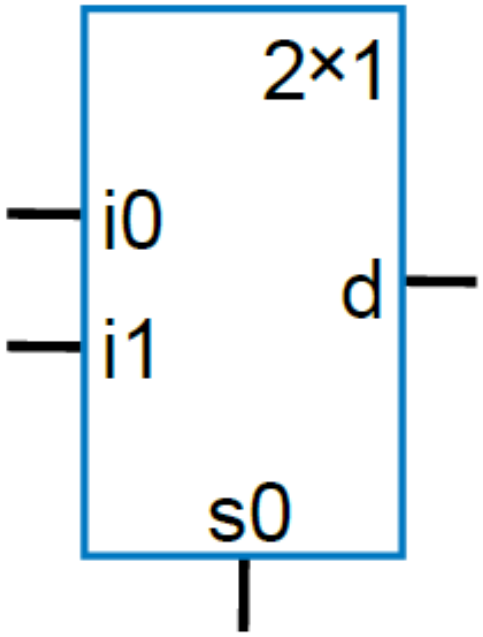


Tabela Verdade			
S0	I1	I0	**D**
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**0**
1	1	0	**1**
1	1	1	**1**

Multiplexador 2x1

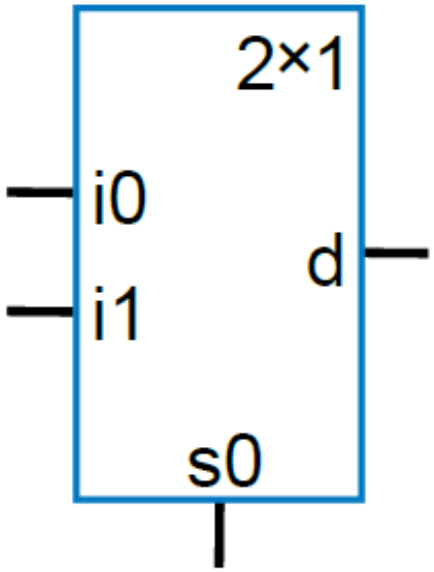
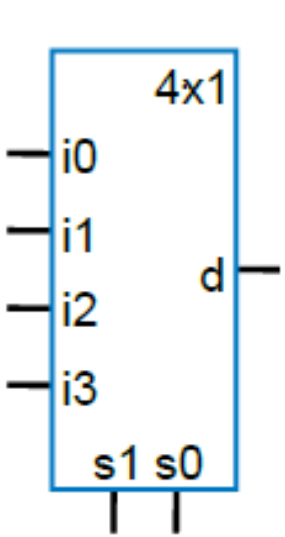


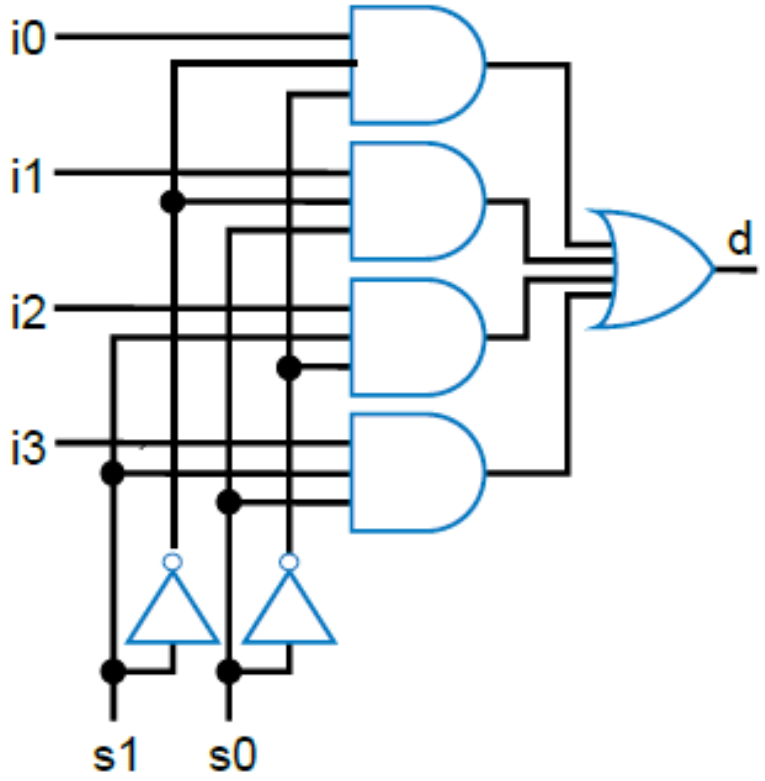
Tabela Verdade			
S0	I1	I0	**D**
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**0**
1	1	0	**1**
1	1	1	**1**

$$D = \overline{S0}.\overline{I1}.I0 + \overline{S0}.I1.I0 + S0.I1.\overline{I0} + S0.I1.I0$$

$$D = \overline{S0}.I0 + S0.I1$$

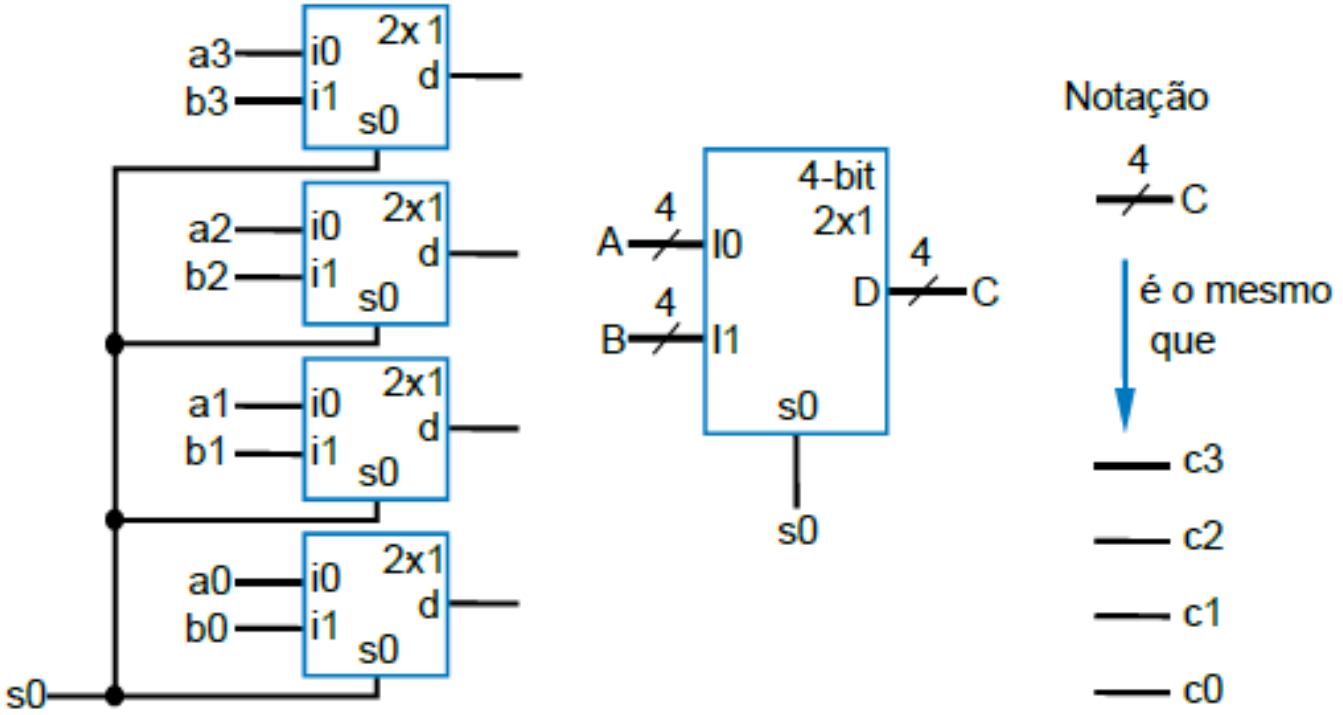
Multiplexador 2x1





Mux N-bit Mx1

- Um MUX é frequentemente utilizado para selecionar entradas de vários bits (N-bit)



Comparadores

- São circuitos que comparam dois números binários
- Vamos considerar palavras binárias de 1 bit

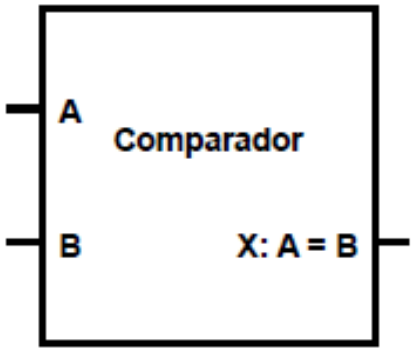


Tabela Verdade		
A	B	**X: A = B**
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**1**

Comparadores

- São circuitos que comparam dois números binários
- Vamos considerar palavras binárias de 1 bit

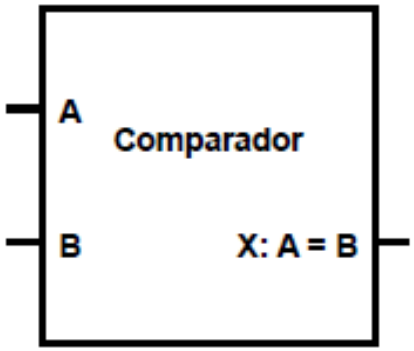


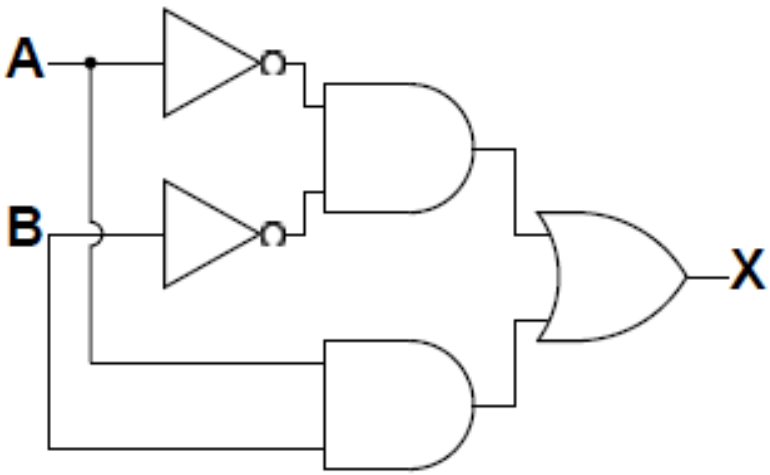
Tabela Verdade		
A	B	**X: A = B**
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**1**

$$X = \bar{A}.\bar{B} + A.B = \overline{A \oplus B}$$

Comparadores

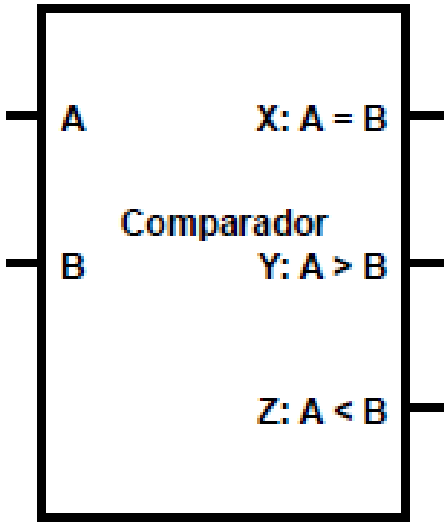
- São circuitos que comparam dois números binários
- Vamos considerar palavras binárias de 1 bit

$$X = \bar{A}.\bar{B} + A.B = \overline{A \oplus B}$$



Comparador com três saídas

- Três saídas:
  - X : A = B
  - Y : A > B
  - Z : A < B



- Como saber se Y : A > B = 1?
- Como saber se Z : A < B = 1?

Comparador com três saídas

- Como saber se Y : A > B = 1?
- Como saber se Z : A < B = 1?

Tabela Verdade				
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**

Comparador com três saídas

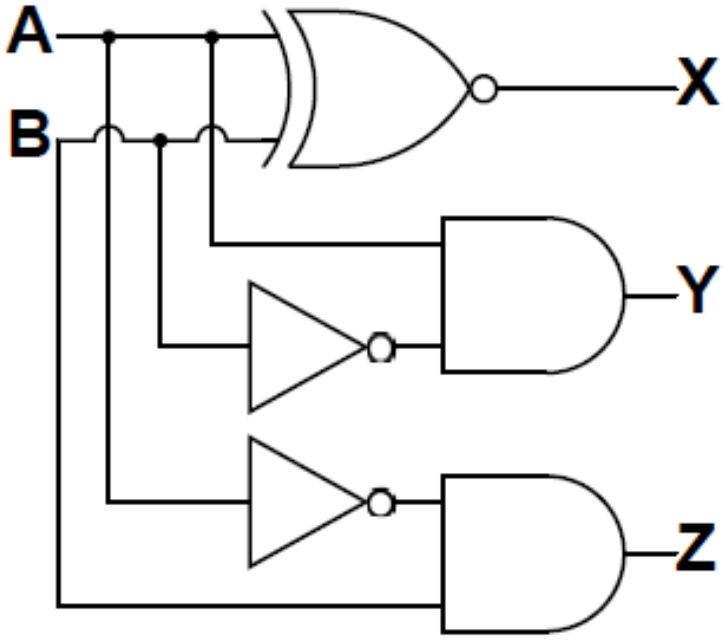
- Como saber se Y : A > B = 1?
- Como saber se Z : A < B = 1?

Tabela Verdade				
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**

$$Y = A.\bar{B}$$
$$Z = \bar{A}.B$$

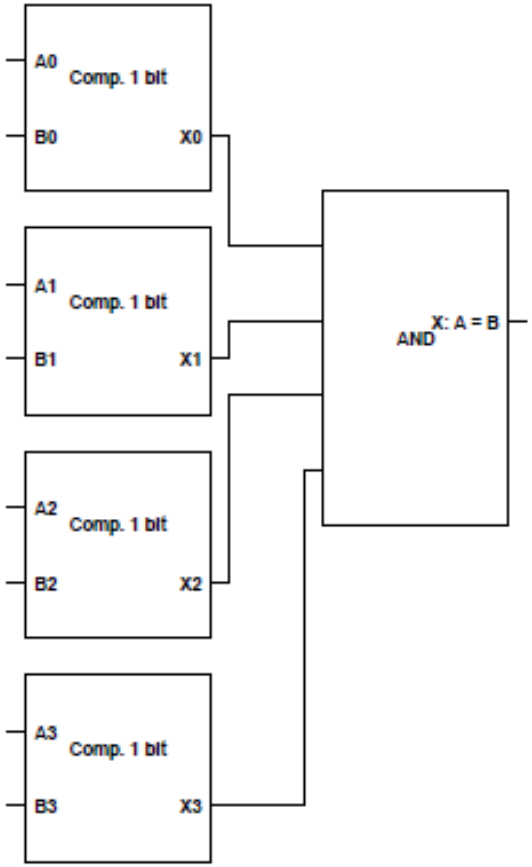
Comparador com três saídas

$$Y = A.\bar{B}$$
$$Z = \bar{A}.B$$



Comparador de N bits

- Para fazer um comparador de n bits, fazemos n comparações de 1 bit



Comparador de N bits com três saídas

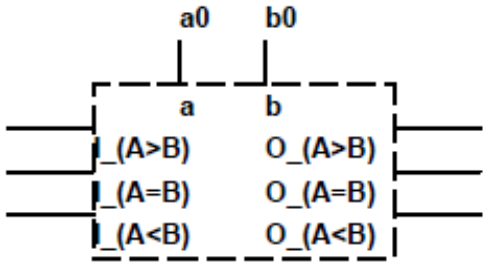
- Como saber qual das seguintes palavras binárias é maior?
  - A = 1001
  - B = 0110
- Passos:
  - Identificar o bit mais significativo das duas palavras binárias
  - Se $A_N > B_N$, então A > B
  - Se $A_N < B_N$, então A < B
  - Se $A_N = B_N$, então devemos testar o bit seguinte

**o índice N identifica o enésimo bit*

Comparador de N bits com três saídas

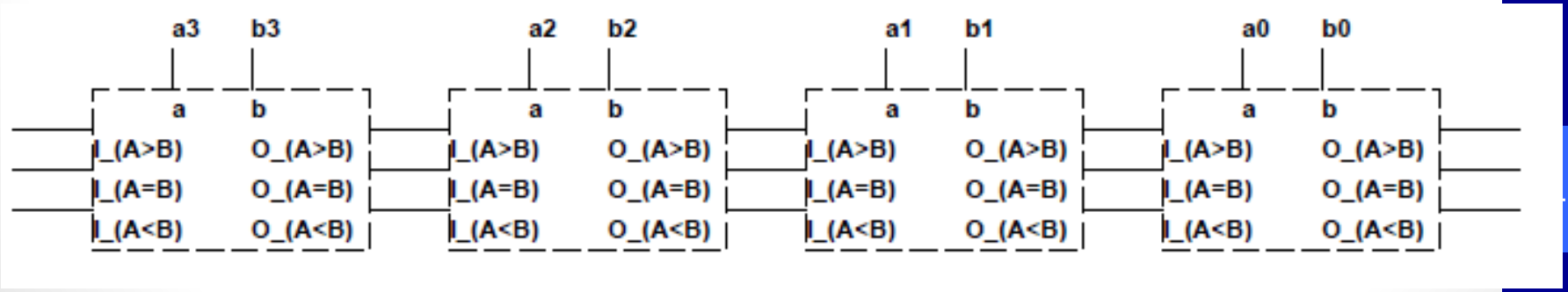
- Podemos usar o seguinte bloco lógico

Resultados dos bits anteriores



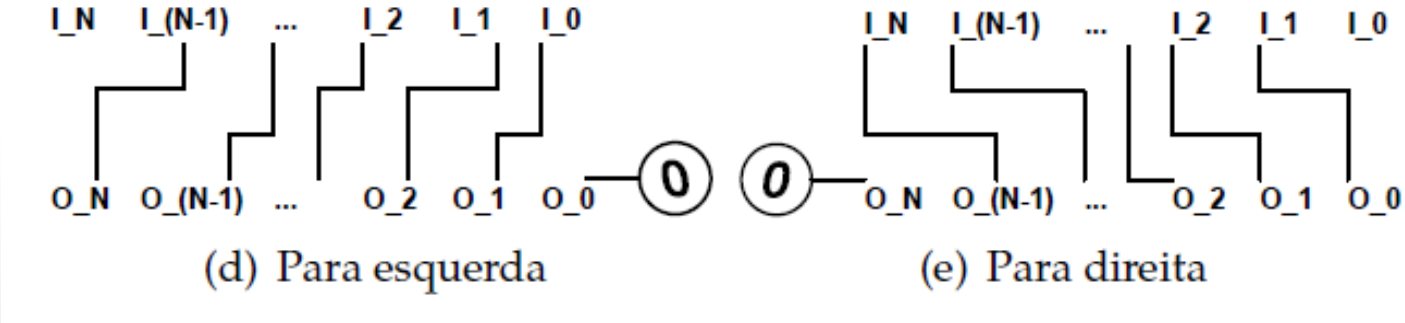
Resultados dos bits atuais

- Vamos concatenar os blocos para comparar palavras binárias com mais de 1 bit



Deslocadores

- Circuito com *n* entradas e *n* saídas
- Entrada: palavra binária de *n* bits
- Saída: palavra binária de *n* bits deslocada
- Circuito simples de deslocamento
  - Não utiliza nenhuma porta lógica
  - Apenas fios
  - Bits descartados: bit IN ou bit I0



(d) Para esquerda (e) Para direita

Aritmética utilizando deslocadores

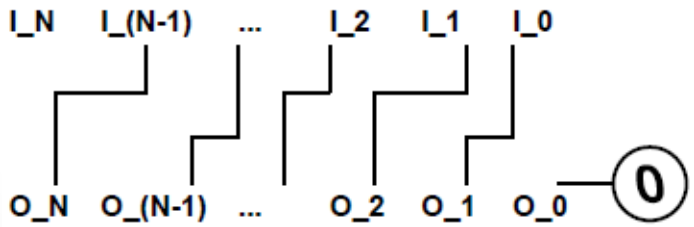
- Exemplos de deslocamentos à esquerda
  - 0110 -> 1100: 6 -> 12
  - 0101 -> 1010: 5 -> 10
  - 0010 -> 0100: 2 -> 4
- Exemplos de deslocamentos à direita
  - 0110 ! 0011: 6 -> 3
  - 0100 ! 0010: 4 -> 2
  - 1110 ! 0111: 14 -> 7
- **Resumindo:**
  - Deslocamento à esquerda: multiplica por 2
  - Deslocamento à direita: divide por 2

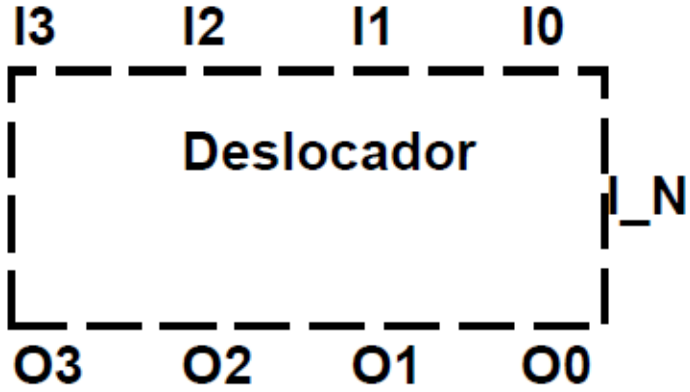
Aritmética utilizando deslocadores

- **O que acontece quando o deslocamento de 1 bit ocorre duas vezes?**
  - 001100 -> 110000: 12 -> 48
- Exemplos de deslocamentos à direita
  - 001100 -> 000011: 12 -> 3
- **Resumindo:**
  - Deslocamento à esquerda de 2 bits: multiplica por 4
  - Deslocamento à direita de 2 bits: divide por 4
- **Conclusão:**
  - Realizar *n* deslocamentos à esquerda de 1 bit, equivale a multiplicar o número por 2^n
  - Realizar *n* deslocamentos à direita de 1 bit, equivale a dividir o número por 2^n

Tipos de deslocadores

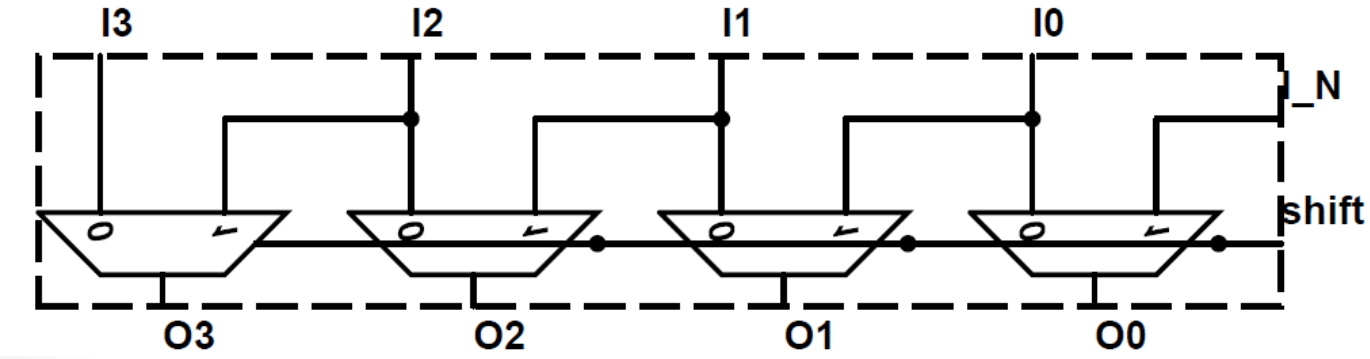
- Deslocador de 1 bit à esquerda
  - Adiciona 0 à I0
  - O mesmo deslocador pode ser feito para direita
- Deslocador de 1 bit à esquerda
  - Permite escolher o valor a ser adicionado em IN
  - O mesmo deslocador pode ser feito para direita





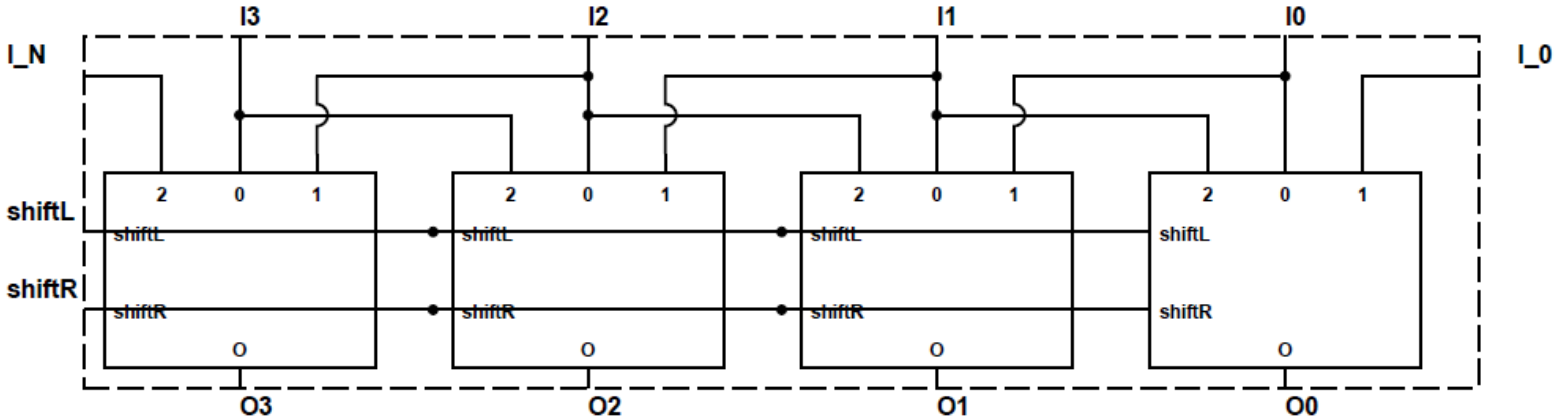
Tipos de deslocadores

- Deslocador com shift
  - A variável shift permite definir se o deslocamento vai ou não ser realizado
  - Se shift for igual a 1: deslocamento à esquerda
  - Se shift for igual a 0: a saída será igual à entrada
  - O mesmo deslocador pode ser feito para direita



Tipos de deslocadores

- Deslocador com escolha de lado
  - Permite escolher o lado do deslocamento (direita ou esquerda)
  - Possui variáveis de controle: shiftL e shiftR
  - Se shiftL for igual a 1: deslocamento à esquerda
  - Se shiftR for igual a 1: deslocamento à direira
  - Se shiftR e shiftL forem igual a 0: a saída será igual à entrada



Conclusões

- Vimos diferentes tipos de circuitos que podem ser construídos a partir de portas lógicas
  - São blocos básicos que representam um nível de abstração mais alta
- Os circuitos que vimos são todos chamados de circuitos combinacionais
  - Suas saídas dependem somente das entradas
  - O tempo necessário para realizar o processamento depende somente do atraso dos componentes, ou seja, do tempo que leva para o sinal chegar até a saída do circuito.
- Nem todos os circuitos são combinacionais!
  - Outros tipos de circuitos dependem do estado em que eles se encontravam até o inicio da computação.
  - Exemplo: elementos de memória

Referências

- **STALLINGS, William. Arquitetura e organização de computadores. 10. ed. São Paulo: Pearson, 2017. 814 p.**
  - **Capítulo 9**

- TOCCI, Ronald J; Widmer, Neal S. Sistemas Digitais: principios e Aplicações. 11. ed. São Paulo SP: Pearson, 2011, 817 p. ISBN 9788576050957
  - Capitulo 1

- PATTERSON, David A; HENNESSY, John L. Organização e projeto de computadores: A interface HARDWARE/SOFTWARE. Rio de Janeiro: Elsevier, 2005, 3ª edição.

Circuitos Combinacionais

Prof. Gustavo Girão
girao@imd.ufrn.br