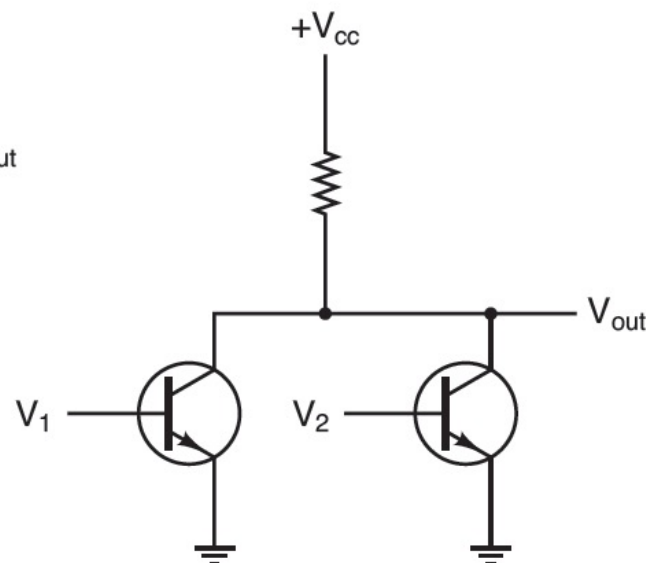
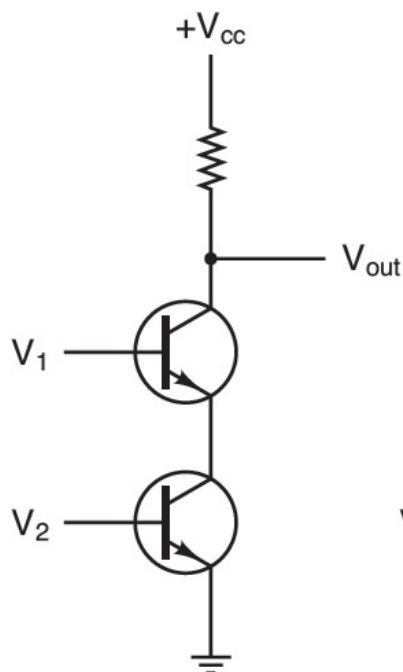
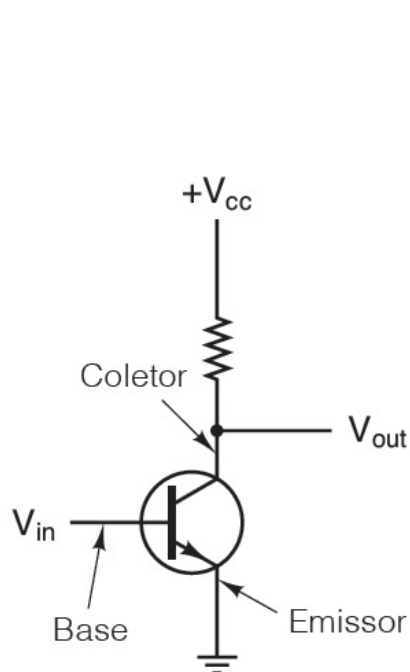


O nível lógico digital

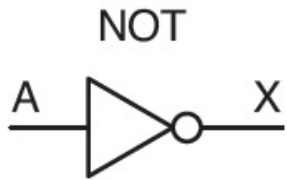
Portas e álgebra booleana

- Minúsculos dispositivos eletrônicos, denominados **portas** (*gates*), podem calcular várias funções dos sinais.
- Inversor de transistor. Porta NAND. Porta NOR.

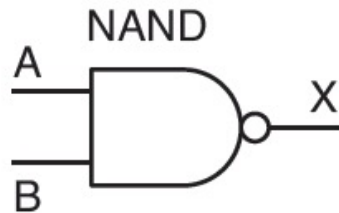


Portas e álgebra booleana

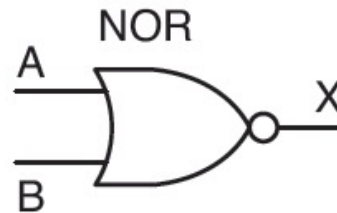
- Símbolos e comportamento funcional das cinco portas básicas.



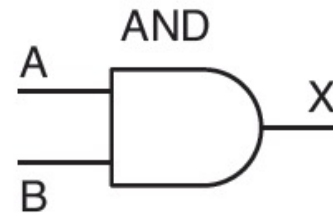
A	X
0	1
1	0



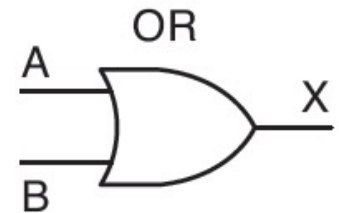
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0



A	B	X
0	0	1
0	1	0
1	0	0
1	1	0



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1



A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

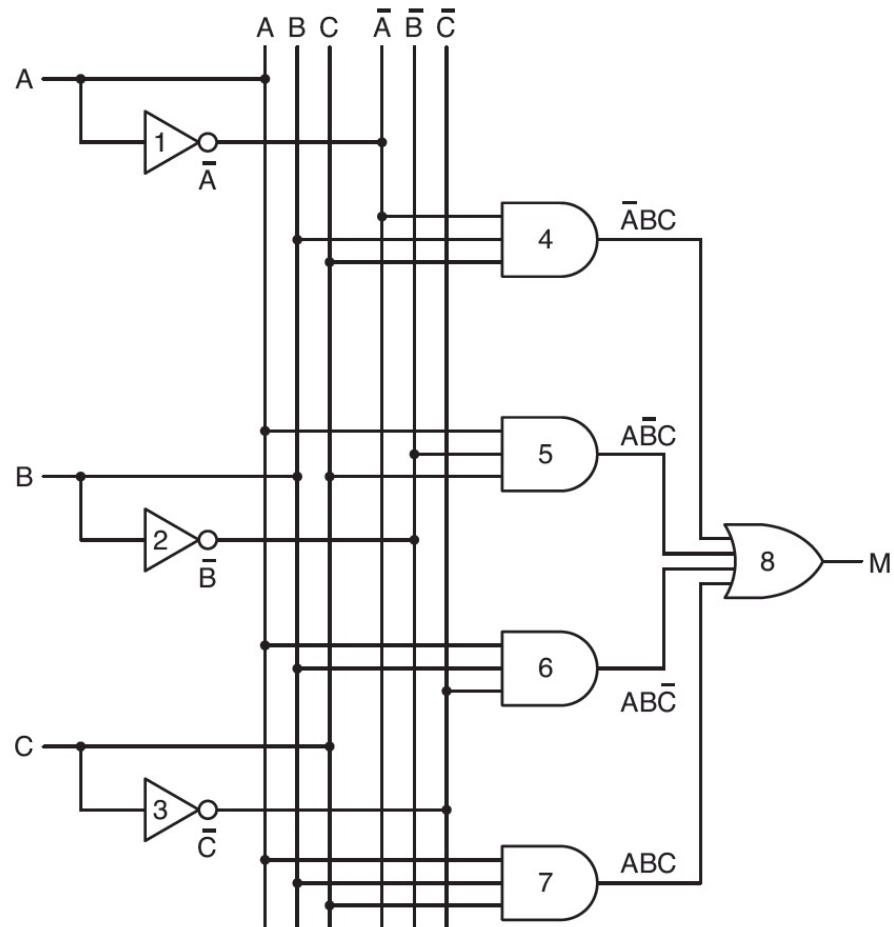
Portas e álgebra booleana

- Para descrever os circuitos que podem ser construídos combinando portas, é necessário um novo tipo de álgebra, no qual variáveis e funções podem assumir somente os valores 0 e 1.
- Essa álgebra é denominada **álgebra booleana**, nome que se deve a seu descobridor, o matemático inglês George Boole (1815–1864).
- Uma função booleana tem uma ou mais variáveis de entrada e produz um resultado que depende somente dos valores dessas variáveis.

Portas e álgebra booleana

- Tabela verdade para a função majoritária de três variáveis e respectivo circuito.

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



Portas e álgebra booleana

- Pelo exemplo da figura anterior deve ficar claro como colocar em prática um circuito para qualquer função booleana:
 1. Escreva a tabela verdade para a função.
 2. Providencie inversores para gerar o complemento de cada entrada.
 3. Desenhe uma porta AND para cada termo que tenha um 1 na coluna de resultado.
 4. Ligue as portas AND às entradas adequadas.
 5. Alimente a saída de todas as portas AND a uma porta OR.

Portas e álgebra booleana

- Para reduzir a complexidade de um circuito, o projetista tem de encontrar outro circuito que calcule a mesma função que o original, mas efetue essa operação com um número menor de portas.
- A álgebra booleana pode ser uma ferramenta valiosa na busca de circuitos equivalentes.
- Um projetista de circuitos começa com uma função booleana e depois aplica a ela as leis da álgebra booleana na tentativa de achar uma função mais simples.
- Um circuito pode ser construído com base na função final.

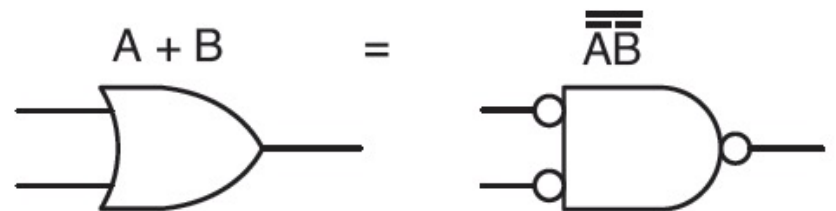
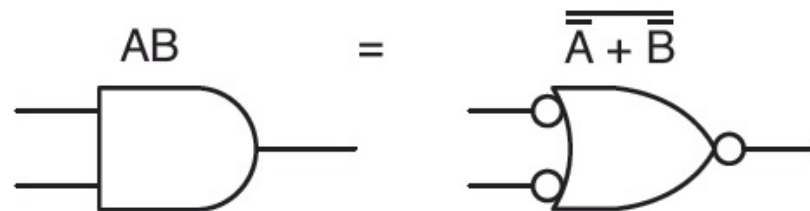
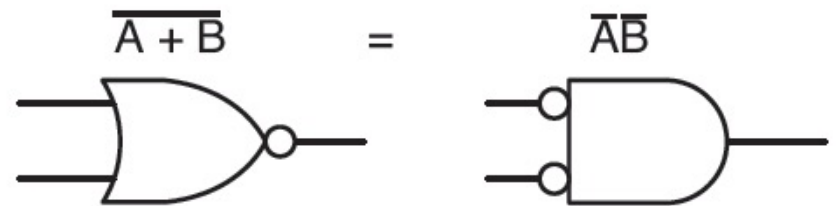
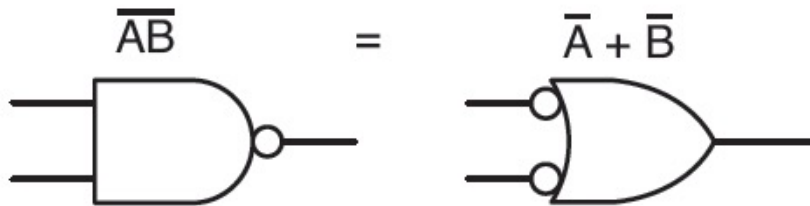
Portas e álgebra booleana

- Algumas identidades da álgebra booleana.

Nome	Forma AND	Forma OR
Lei da identidade	$1A = A$	$0 + A = A$
Lei do elemento nulo	$0A = 0$	$1 + A = 1$
Lei idempotente	$AA = A$	$A + A = A$
Lei do inverso	$\overline{A\overline{A}} = 0$	$A + \overline{A} = 1$
Lei comutativa	$AB = BA$	$A + B = B + A$
Lei associativa	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Lei distributiva	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Lei da absorção	$A(A + B) = A$	$A + AB = A$
Lei de De Morgan	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A} + \overline{B} = \overline{AB}$

Portas e álgebra booleana

- Símbolos alternativos para algumas portas: NAND. NOR. AND. OR.

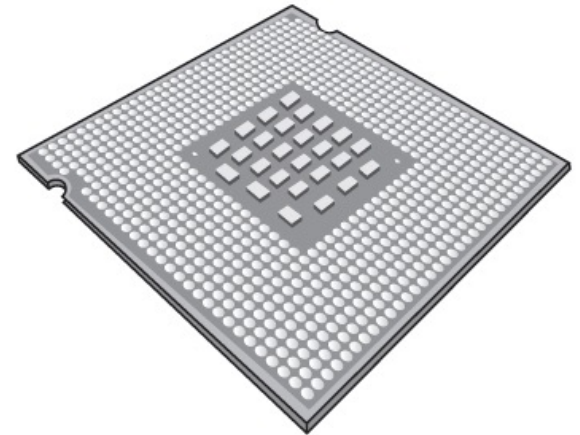
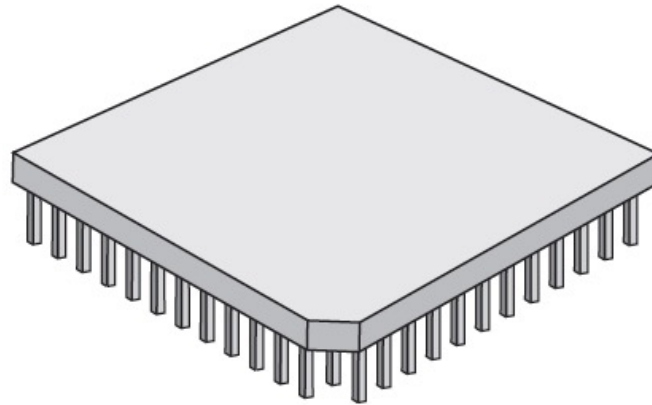
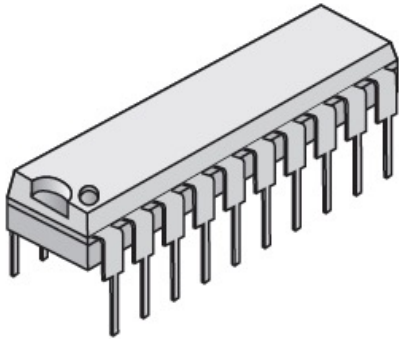


Circuitos integrados

- Portas não são fabricadas nem vendidas individualmente, mas em unidades denominadas circuitos integrados, muitas vezes denominados **ICs** ou **chips**.
- Um IC é um pedaço quadrado de silício de tamanho variado, dependendo de quantas portas são necessárias para executar os componentes do chip.
- Substratos pequenos medirão cerca de 2×2 mm, enquanto os maiores podem ter até 18×18 mm.
- ICs costumam ser montados em pacotes retangulares de plástico ou cerâmica.

Circuitos integrados

- Tipos comuns de pacotes de circuito integrado, incluindo um pacote *dual-in-line*, ou DIP, PGA e LGA.

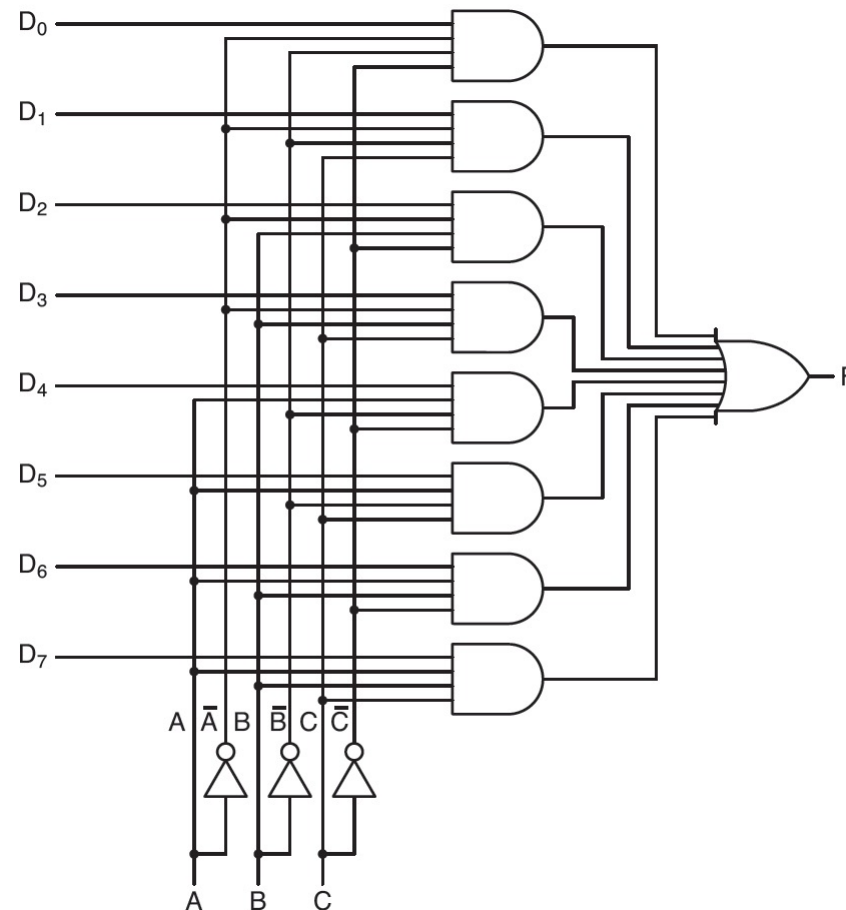


Circuitos combinatórios

- Muitas aplicações de lógica digital requerem um circuito com múltiplas entradas e múltiplas saídas, no qual as saídas são determinadas exclusivamente pelas entradas em questão.
- Esses circuitos são denominados **circuitos combinatórios**.
- No nível lógico, um **multiplexador** é um circuito com 2^n entradas de dados, uma saída de dados e n entradas de controle que selecionam uma das entradas de dados.

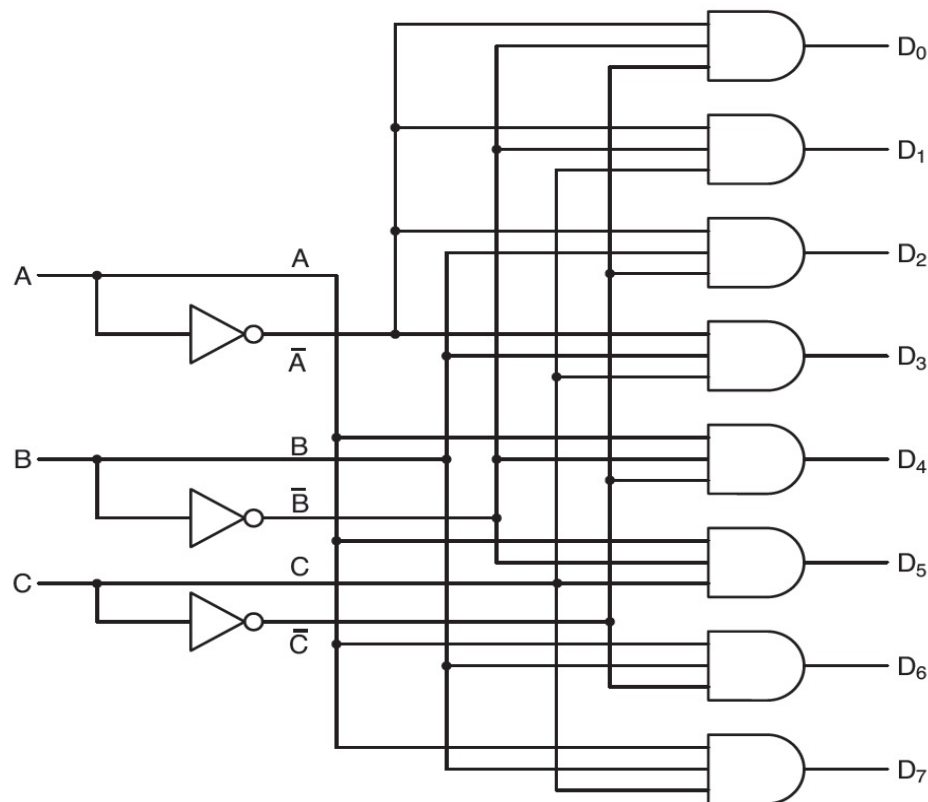
Circuitos combinatórios

- Circuito multiplexador de oito entradas.



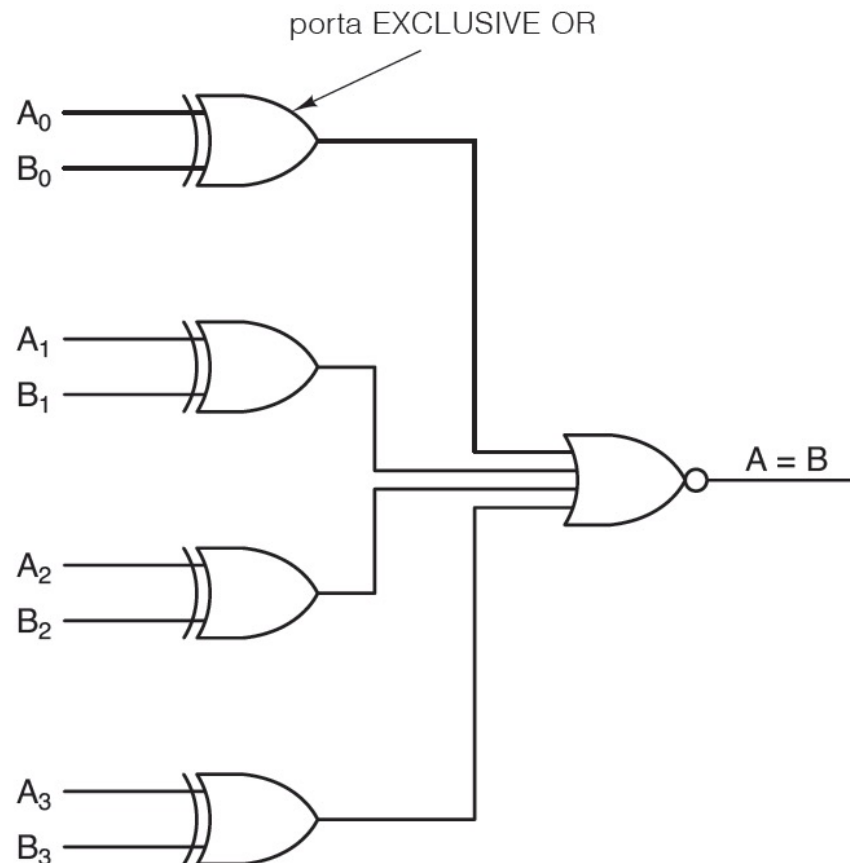
Circuitos combinatórios

- **Decodificador** - circuito que toma um número de n bits como entrada e o usa para selecionar (isto é, definir em 1) exatamente uma das 2^n linhas de saída.



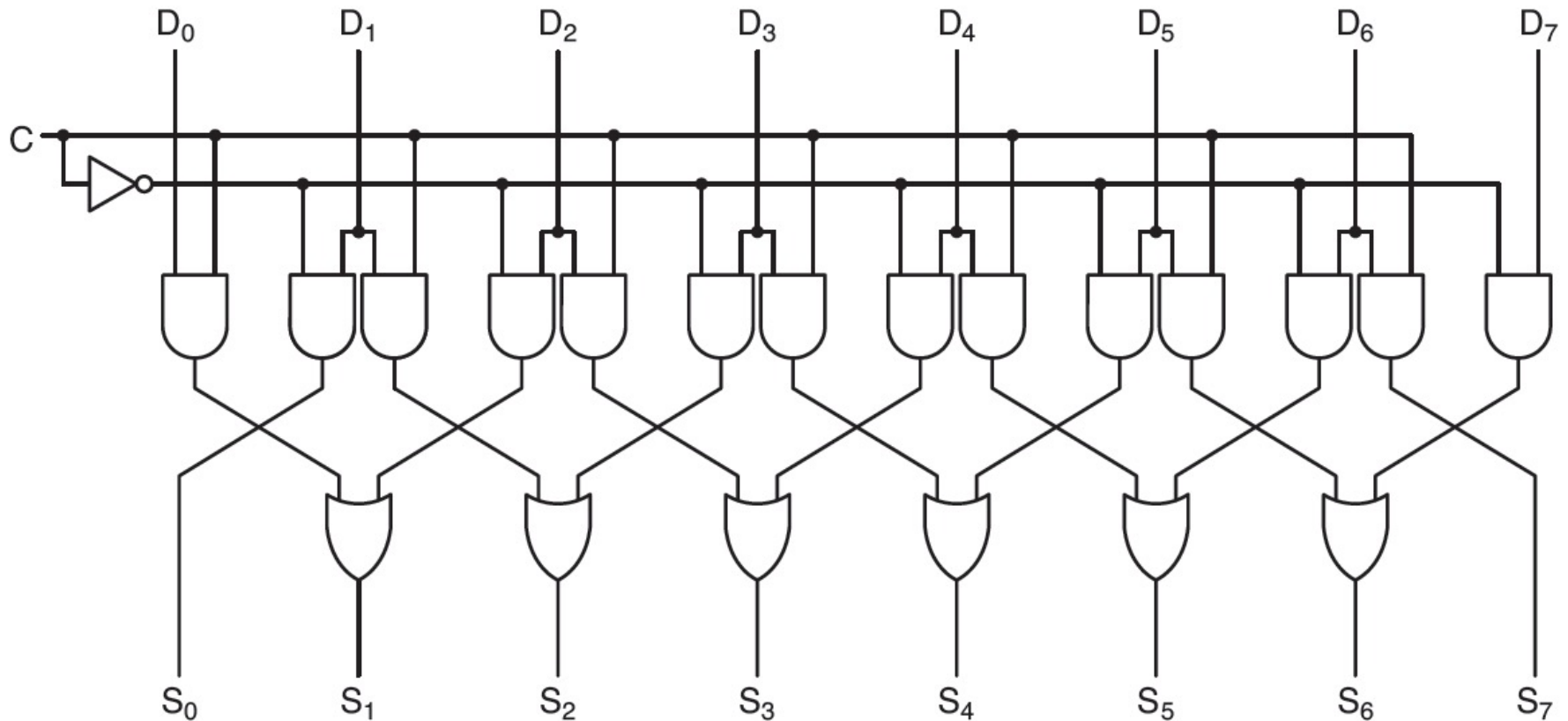
Circuitos combinatórios

- Outro circuito útil é o **comparador**, que compara duas palavras de entrada.



Circuitos aritméticos

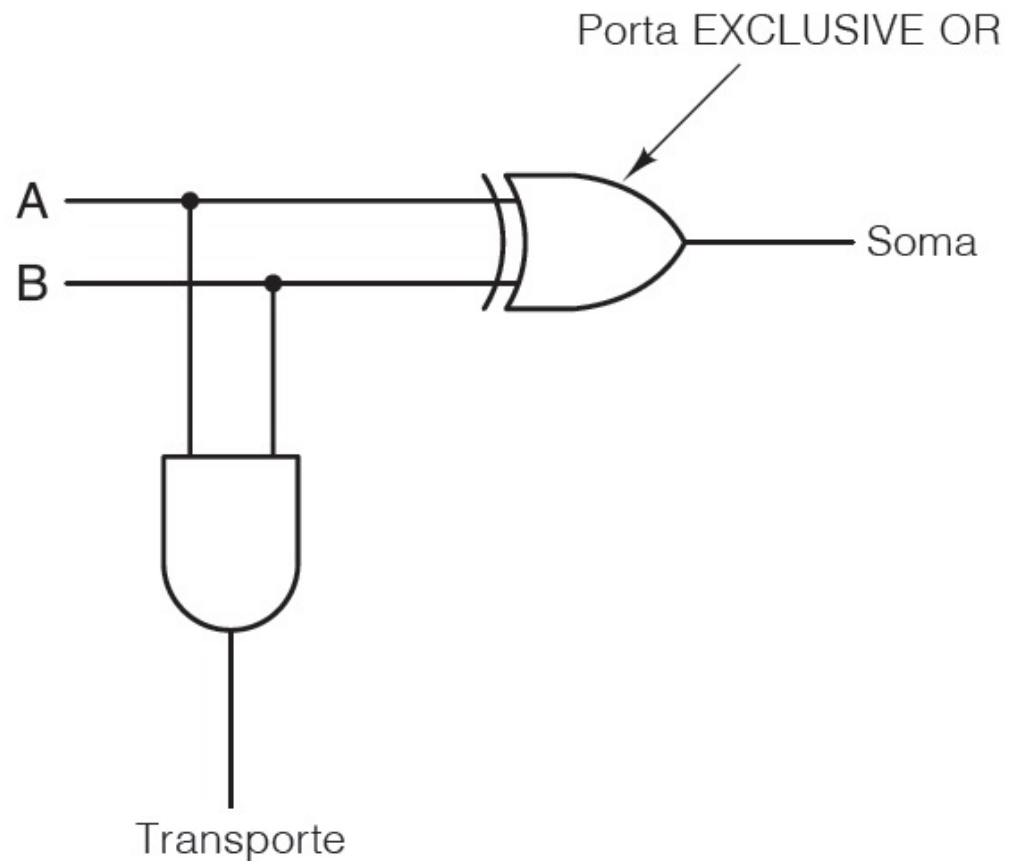
- Nosso primeiro circuito aritmético é um **deslocador** de oito entradas e oito saídas.



Circuitos aritméticos

- Um circuito para calcular o bit de soma e o de transporte é conhecido como um **meio-somador**.

A	B	Soma	Transporte
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

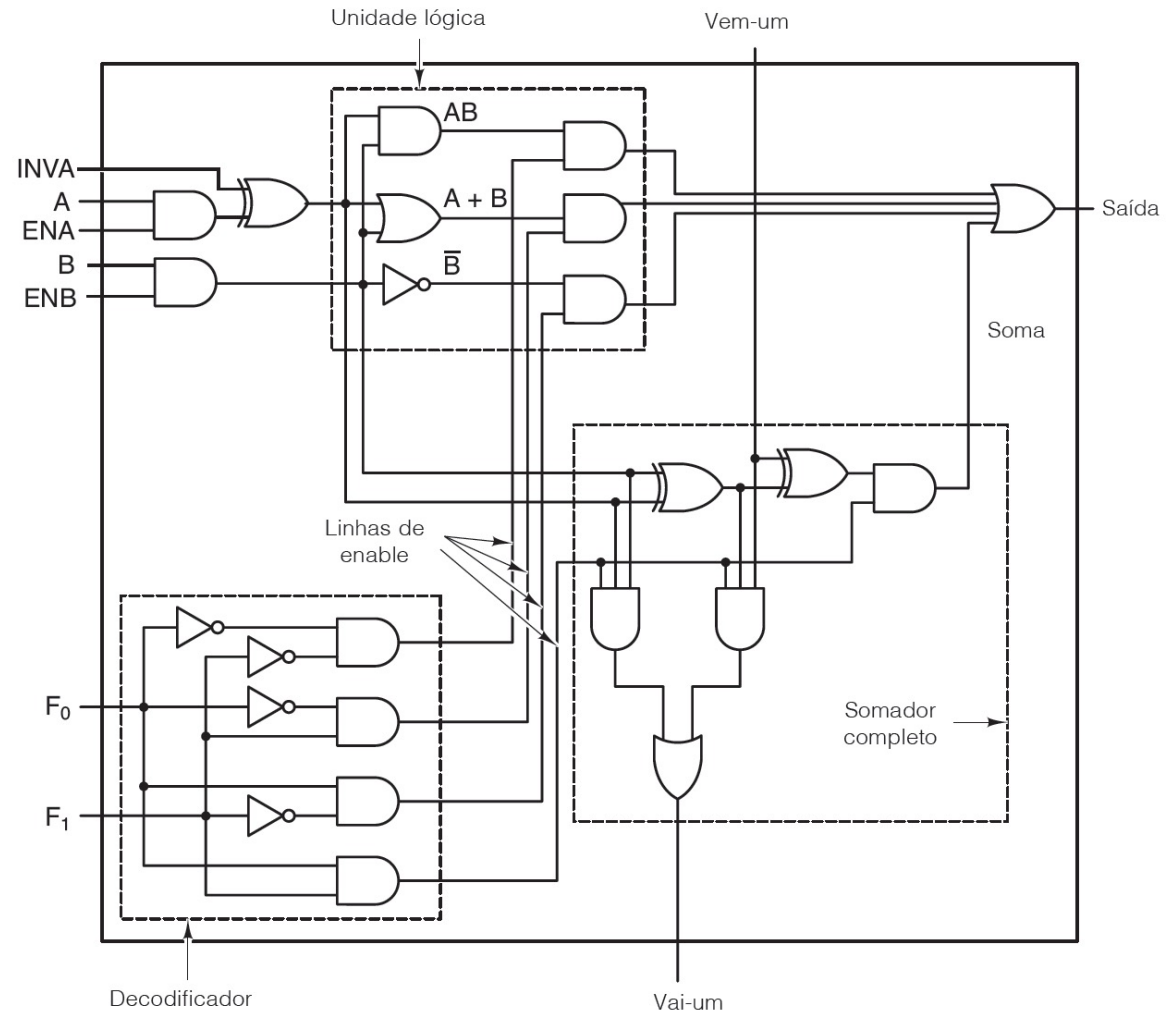


Circuitos aritméticos

- Grande parte dos computadores contém um único circuito para efetuar AND, OR e soma de duas palavras de máquina.
- No caso típico, tal circuito para palavras de n bits é composto de n circuitos idênticos para as posições individuais de bits.
- **Unidade lógica e aritmética (ULA)** pode calcular qualquer uma das quatro funções, dependendo de as linhas de entrada de seleção de função F_0 e F_1 conterem 00, 01, 10 ou 11 (binário).

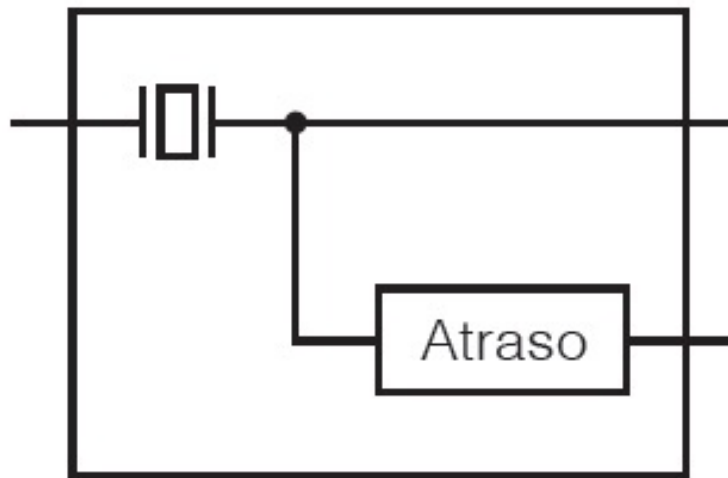
Circuitos aritméticos

- ULA de 1 bit.



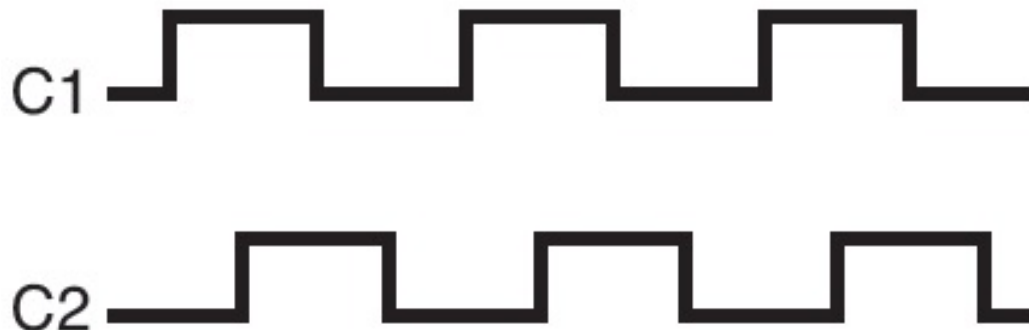
Clocks

- Um **clock** é um circuito que emite uma série de pulsos com uma largura de pulso precisa e intervalos precisos entre pulsos consecutivos.
- O intervalo de tempo entre as arestas correspondentes de dois pulsos consecutivos é denominado **tempo de ciclo de clock**.



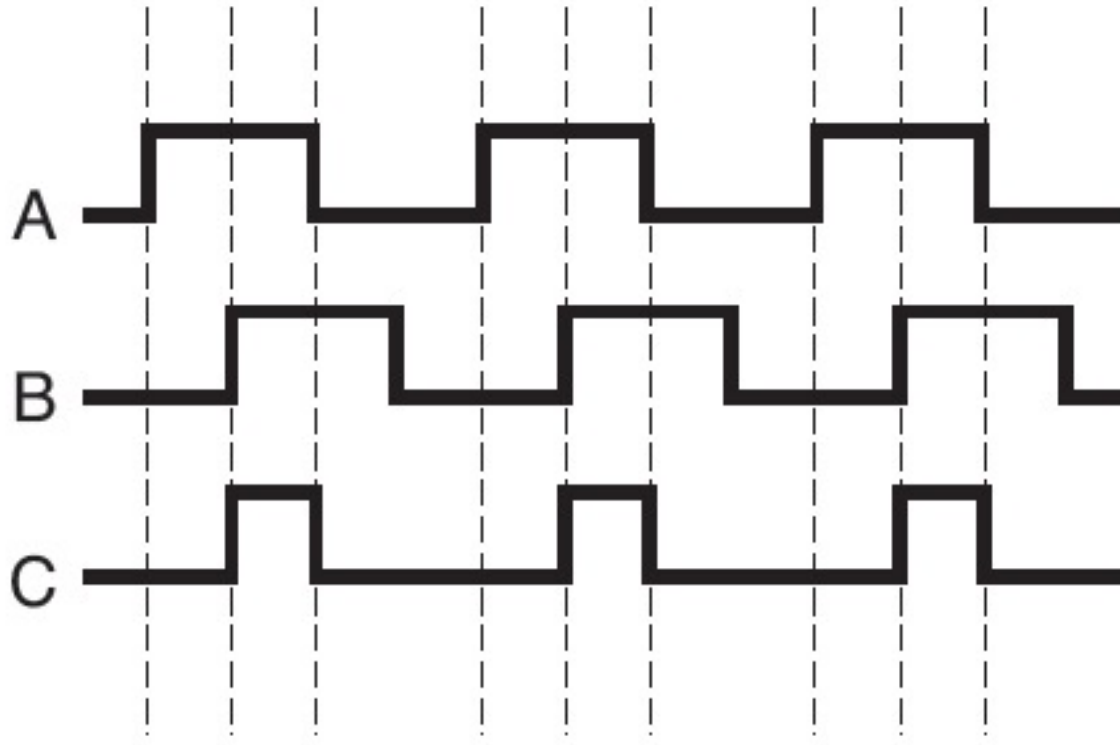
Clocks

- O diagrama de temporização dá quatro referências de tempo para eventos discretos:
 1. Fase ascendente de C1.
 2. Fase descendente de C1.
 3. Fase ascendente de C2.
 4. Fase descendente de C2.



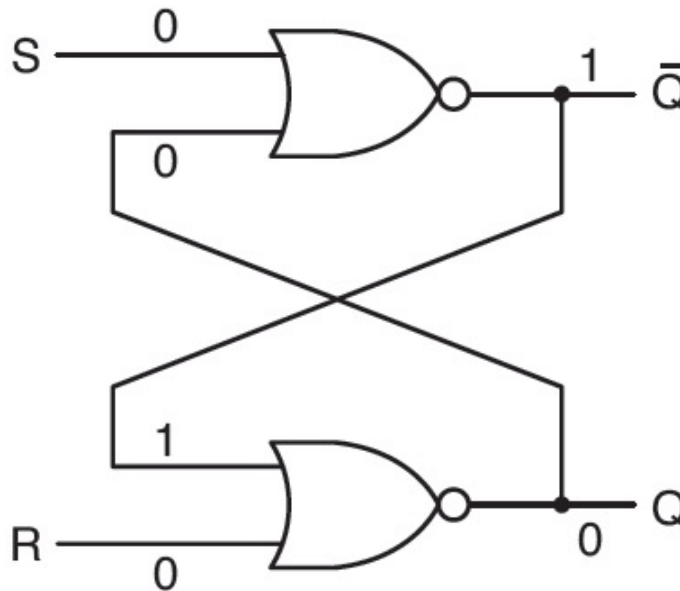
Clocks

- Geração de um *clock* assimétrico.



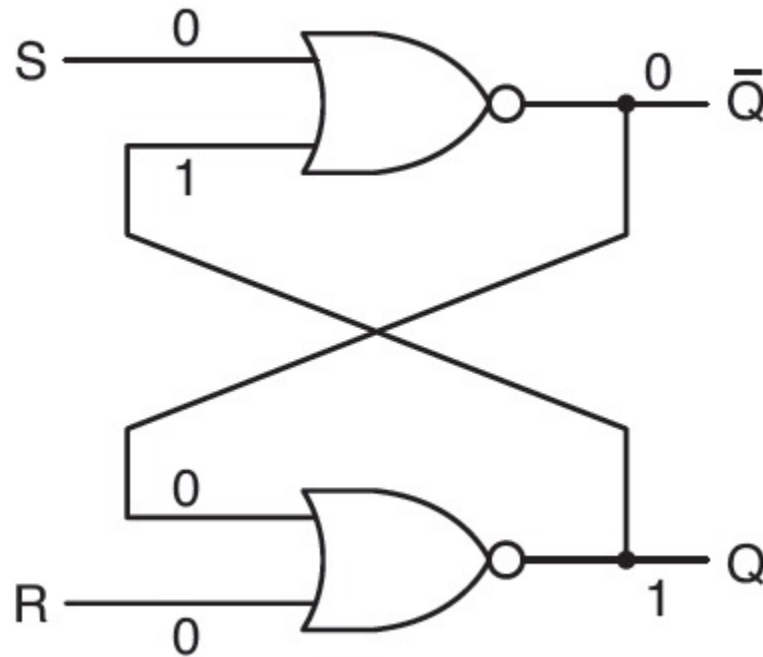
Memórias de 1 bit

- Para criar uma memória de 1 bit (“*latch*”), precisamos de um circuito que “se lembre”, de algum modo, de valores de entrada anteriores.
- Tal circuito pode ser construído com base em duas portas NOR:



Memórias de 1 bit

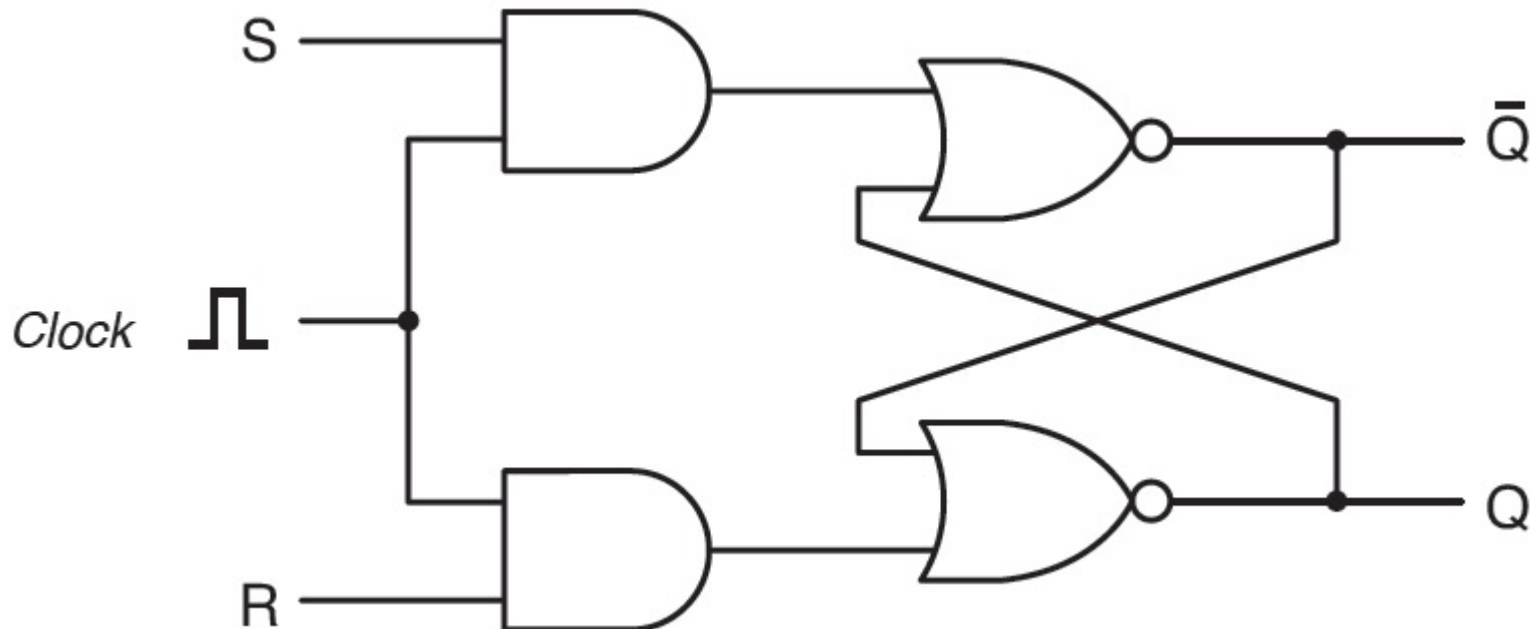
- *Latch* NOR no estado 1. Tabela verdade para NOR.



A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

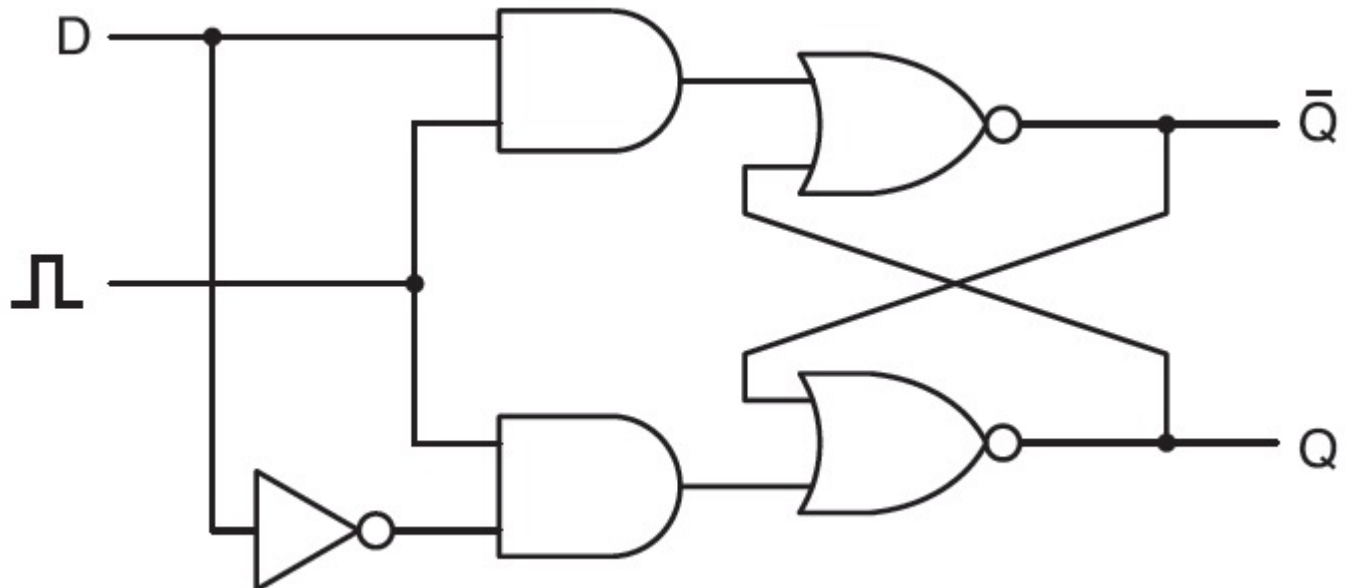
Memórias de 1 bit

- Muitas vezes é conveniente impedir que o *latch* mude de estado. Para atingir esse objetivo, fazemos uma ligeira modificação no circuito básico para obter um ***latch* SR com *clock***.



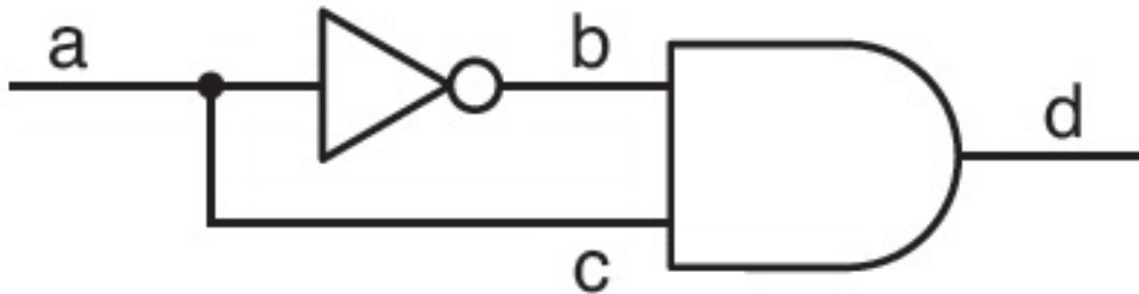
Memórias de 1 bit

- Uma boa maneira de resolver a instabilidade do *latch* SR (causada quando $S = R = 1$) é evitar que ela ocorra.
- Quando o *clock* for 1, o valor corrente de D é lido e armazenado no *latch*. Esse circuito, denominado ***latch D com clock***.



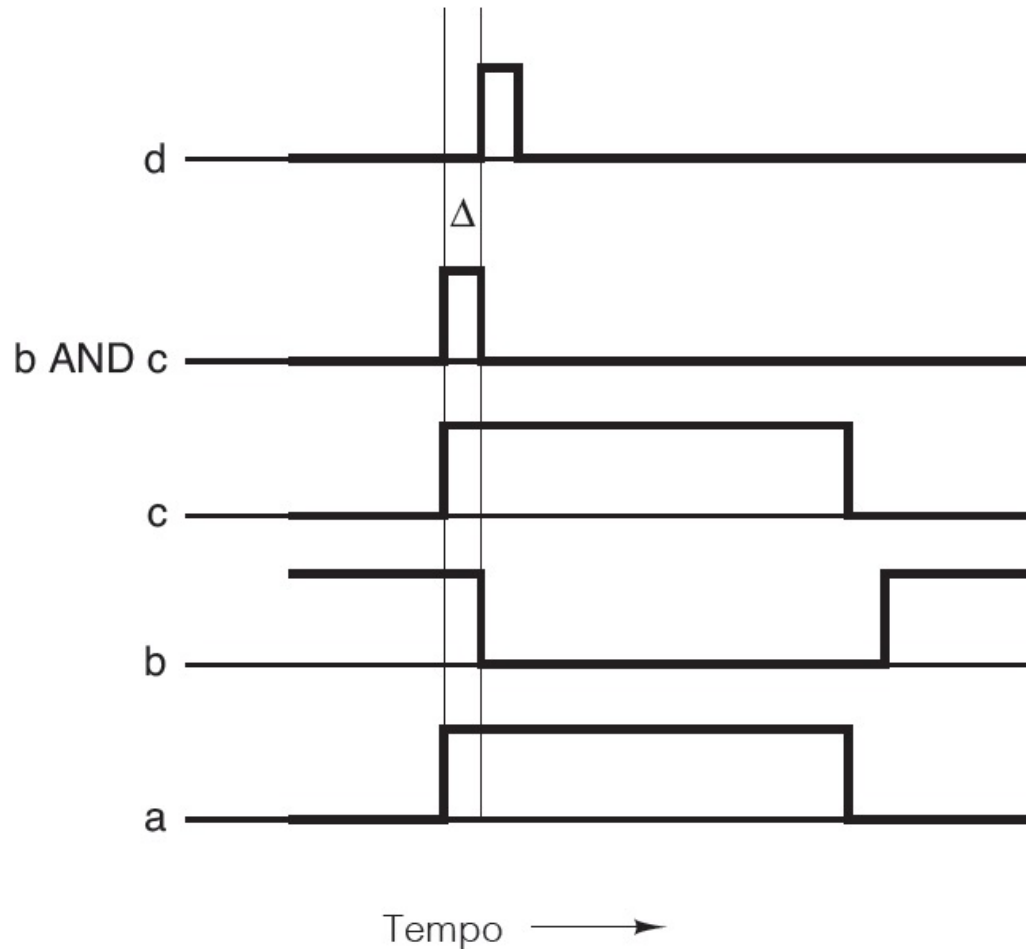
Flip-flops

- Na variante, denominada **flip-flop**, a transição de estado não ocorre quando o *clock* é 1, mas durante a transição de 0 para 1 (borda ascendente), ou de 1 para 0 (borda descendente).
- Gerador de pulso.



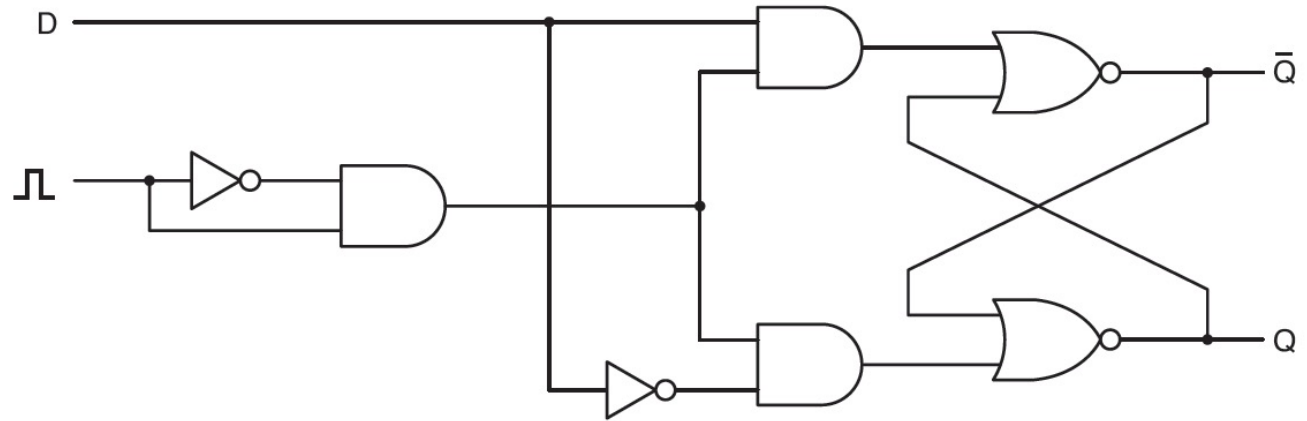
Flip-flops

- Temporização em quatro pontos do circuito.

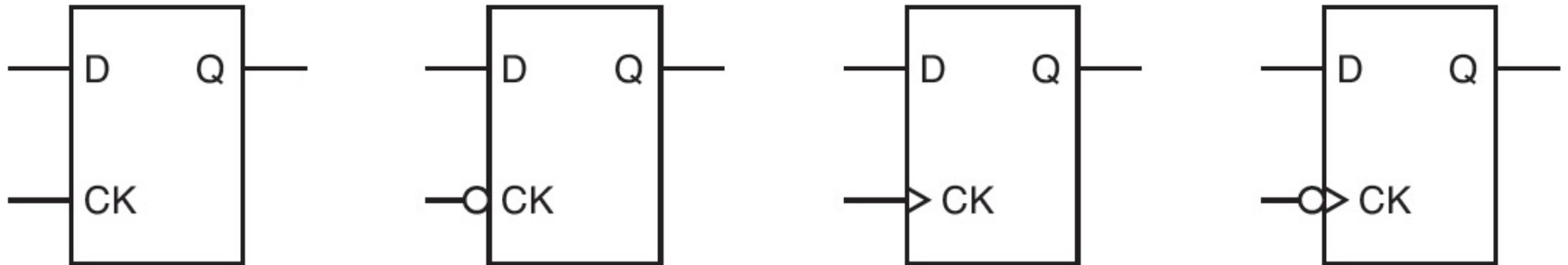


Flip-flops

- *Flip-flop D.*

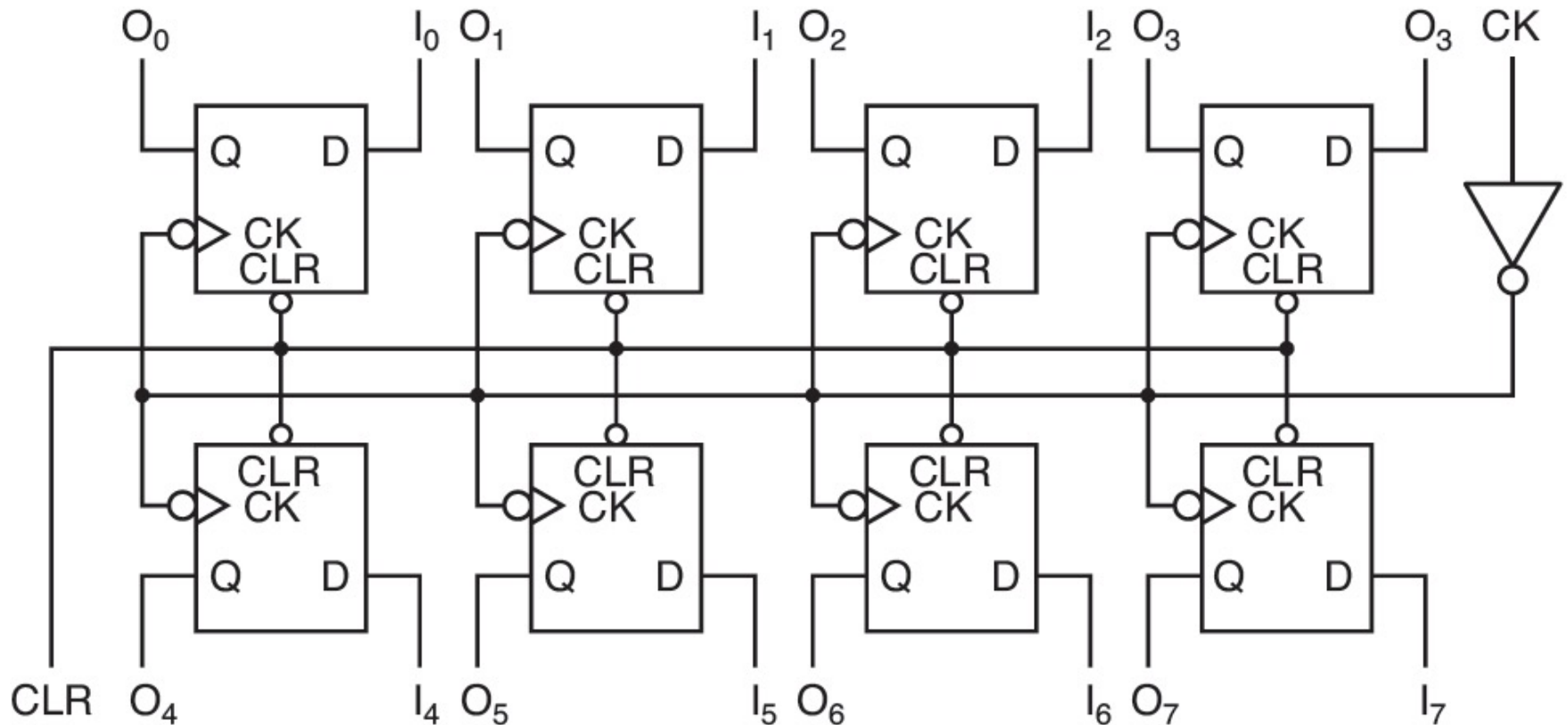


- *Latches e flip-flops D.*



Registadores

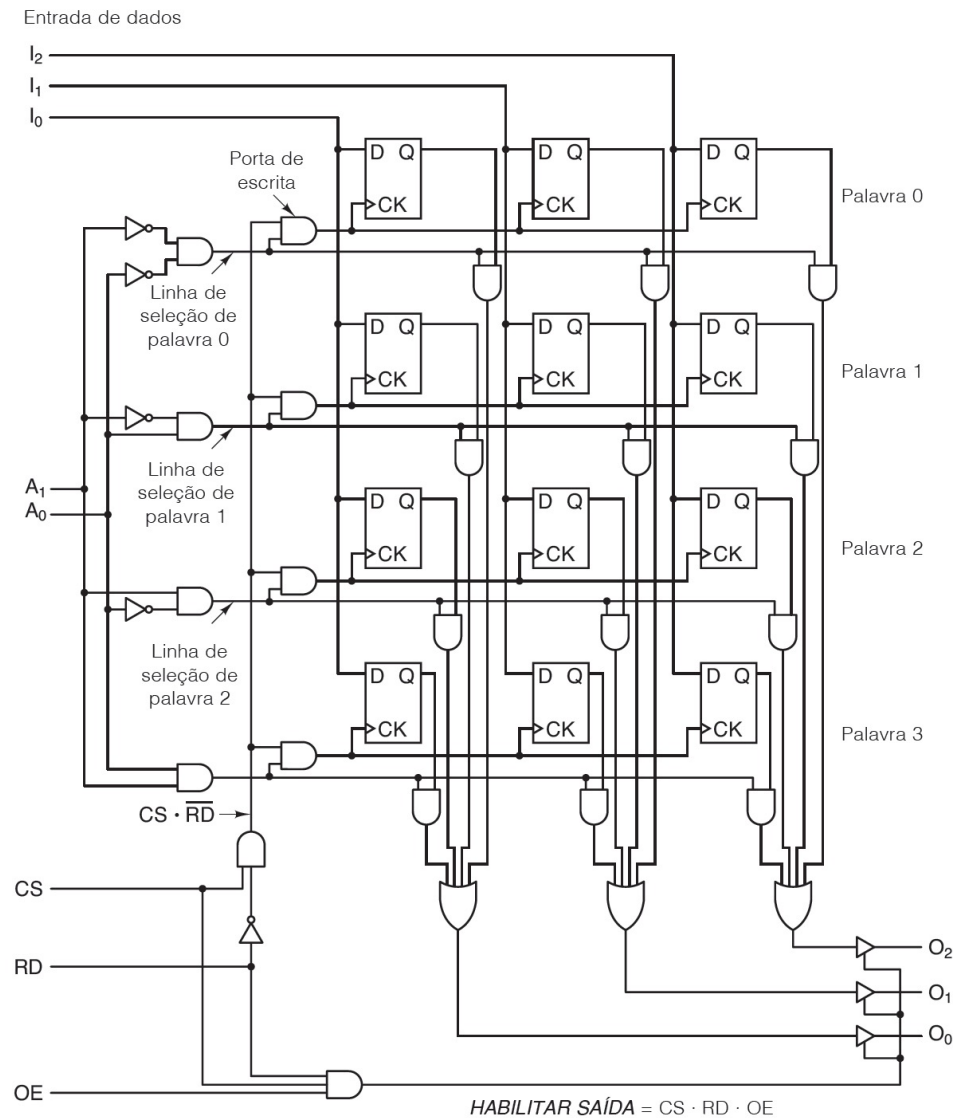
- O registrador abaixo mostra como oito *flip-flops* podem ser ligados para formar um registrador armazenador de 8 bits.



Organização da memória

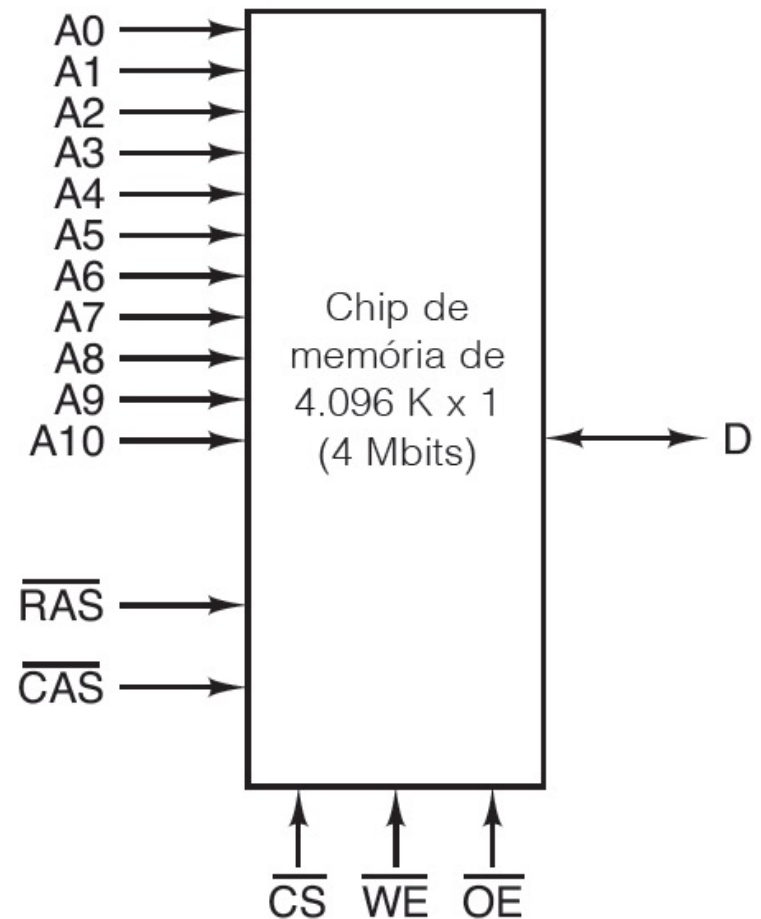
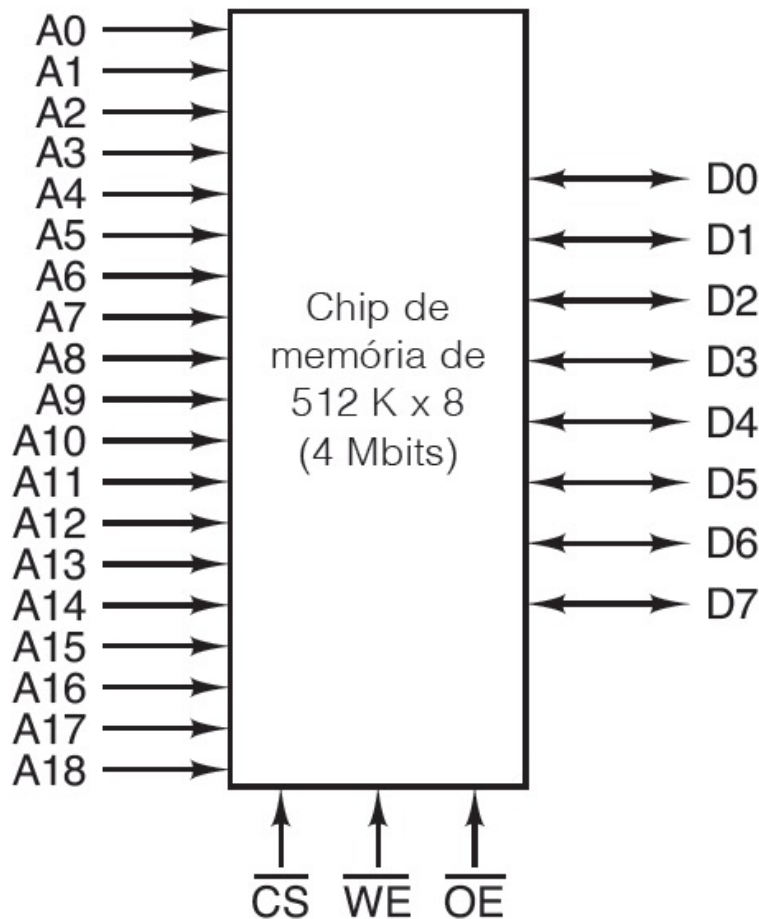
- A figura a seguir mostra um diagrama lógico para uma memória 4 x 3.
- Cada linha é uma das quatro palavras de 3 bits.
- Uma operação de leitura ou escrita sempre lê ou escreve uma palavra completa.
- Observe que o número de palavras é sempre uma potência de 2.

Organização da memória



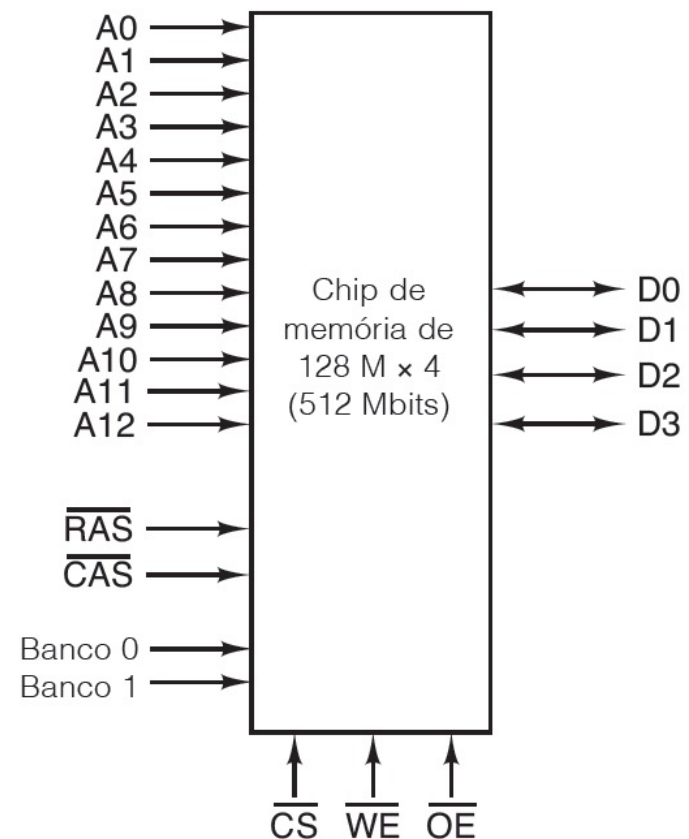
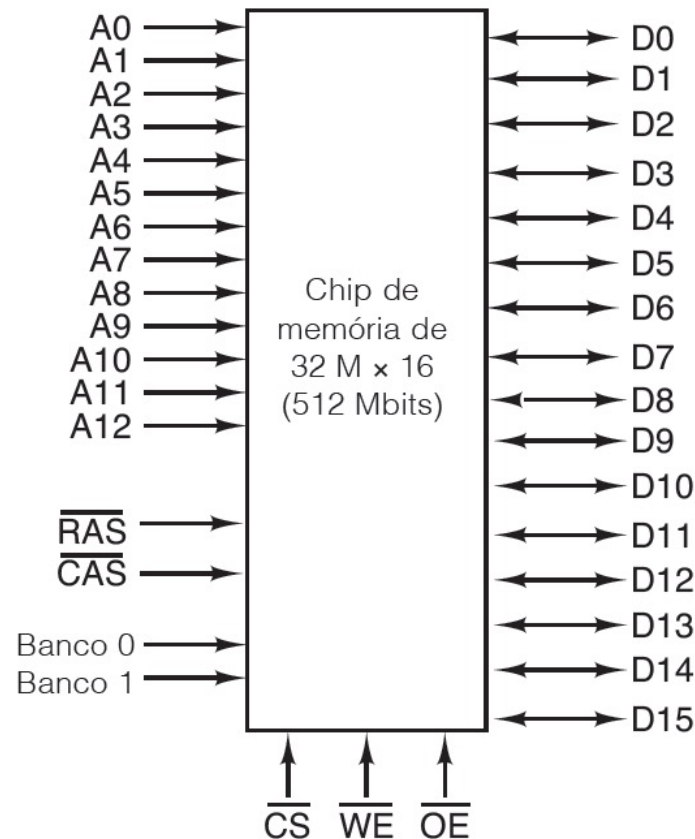
Chips de memória

- Dois modos de organizar um chip de memória de 4 Mbits.



Chips de memória

- Dois modos de organizar um chip de memória de 512 Mbits.



RAMs e ROMs

- **RAMs** podem ser de duas variedades, estáticas e dinâmicas.
- Nas estáticas (**Static RAMs – SRAMs**), a construção interna usa circuitos similares ao nosso *flip-flop* D básico.
- RAMs dinâmicas (**Dynamic RAMs – DRAMs**), ao contrário, não usam flip-flops.
- Em vez disso, uma RAM dinâmica é um arranjo de células, cada uma contendo um transistor e um pequenino capacitor.

RAMs e ROMs

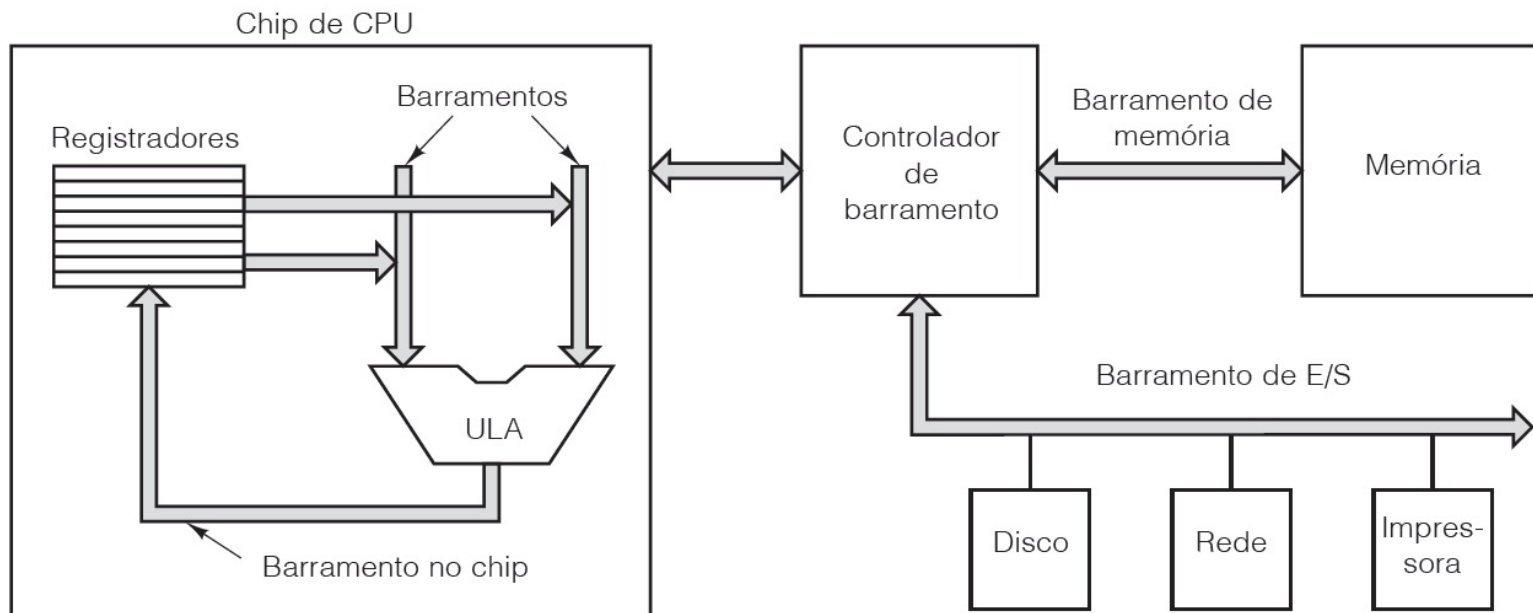
- Em muitas aplicações, como brinquedos, eletrodomésticos e carros, o programa e alguns dos dados devem permanecer armazenados mesmo quando o fornecimento de energia for interrompido.
- Uma vez instalados, nem o programa nem os dados são alterados.
- Esses requisitos levaram ao desenvolvimento de **ROMs (Read-Only Memories – memórias somente de leitura)**, que não podem ser alteradas nem apagadas, seja intencionalmente ou não.

Chips de CPU

- Cada chip de CPU tem um conjunto de pinos por meio dos quais deve ocorrer toda sua comunicação com o mundo exterior.
- Os pinos de controle podem ser agrupados aproximadamente nas seguintes categorias principais:
 1. Controle de barramento.
 2. Interrupções.
 3. Arbitragem de barramento.
 4. Sinalização de coprocessador.
 5. Estado.
 6. Diversos.

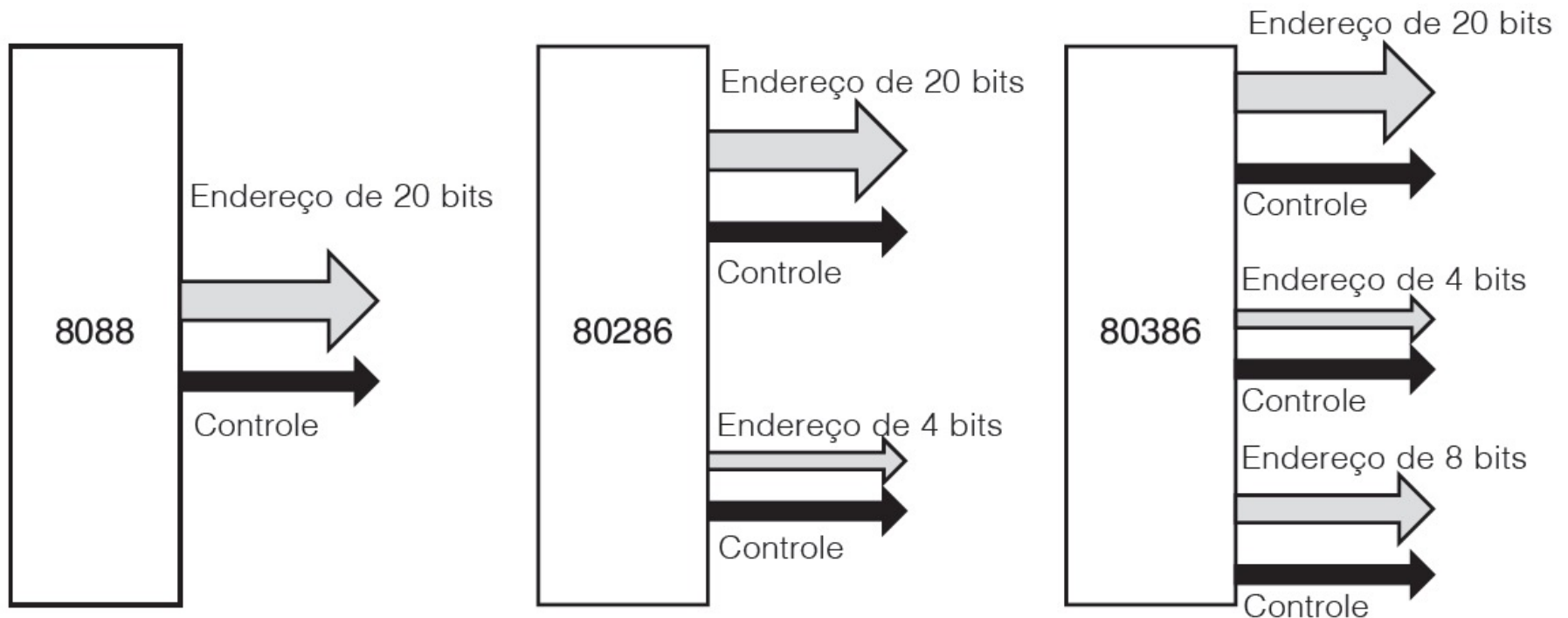
Barramentos de computador

- Um **barramento** é um caminho elétrico comum entre vários dispositivos.
- Sistema mínimo, com um barramento de memória e um barramento de E/S:



Barramentos de computador

- Crescimento de um barramento de endereços ao longo do tempo.



Barramentos de computador

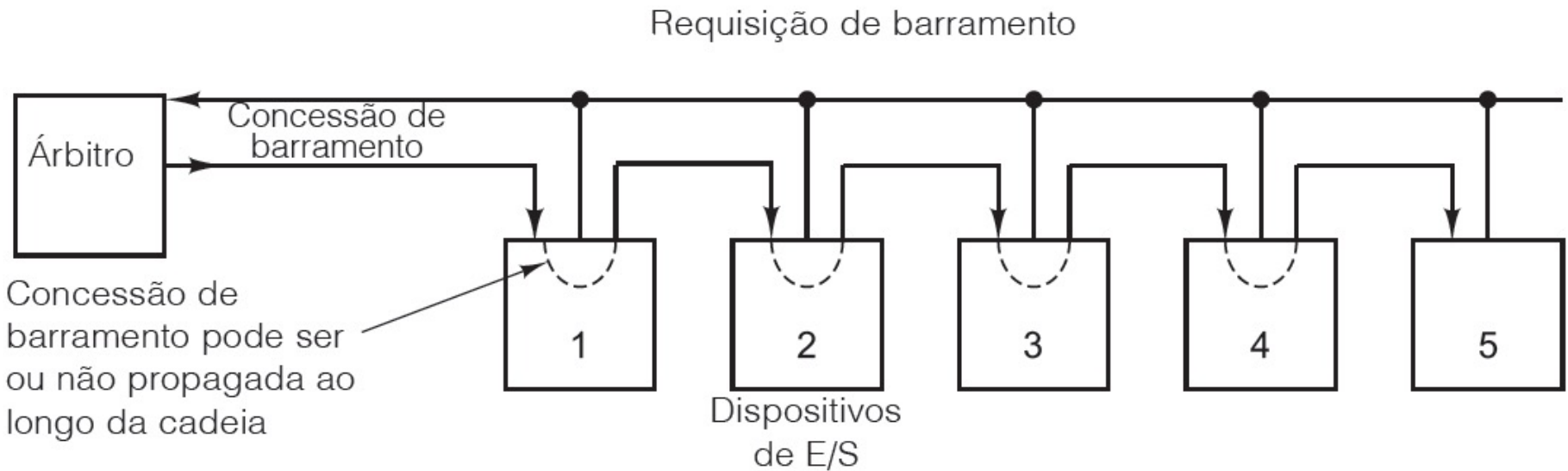
- Barramentos podem ser divididos em duas categorias distintas, dependendo de seu *clock*.
- Um **barramento síncrono** tem uma linha comandada por um oscilador de cristal. Todas as atividades do barramento tomam um número inteiro dos ciclos denominados **ciclos de barramento**.
- O **barramento assíncrono**, não tem um *clock* mestre.
- Ciclos de barramento podem ter qualquer largura requerida e não são os mesmos entre todos os pares de dispositivos.

Arbitragem de barramento

- “O que acontece se dois ou mais dispositivos quiserem se tornar mestres de barramento ao mesmo tempo?”
- É preciso algum mecanismo de **arbitragem de barramento** para evitar o caos.
- Mecanismos de arbitragem podem ser centralizados ou descentralizados.
- Se vários níveis de prioridade são requisitados ao mesmo tempo, o árbitro emite uma concessão somente ao de prioridade mais alta.

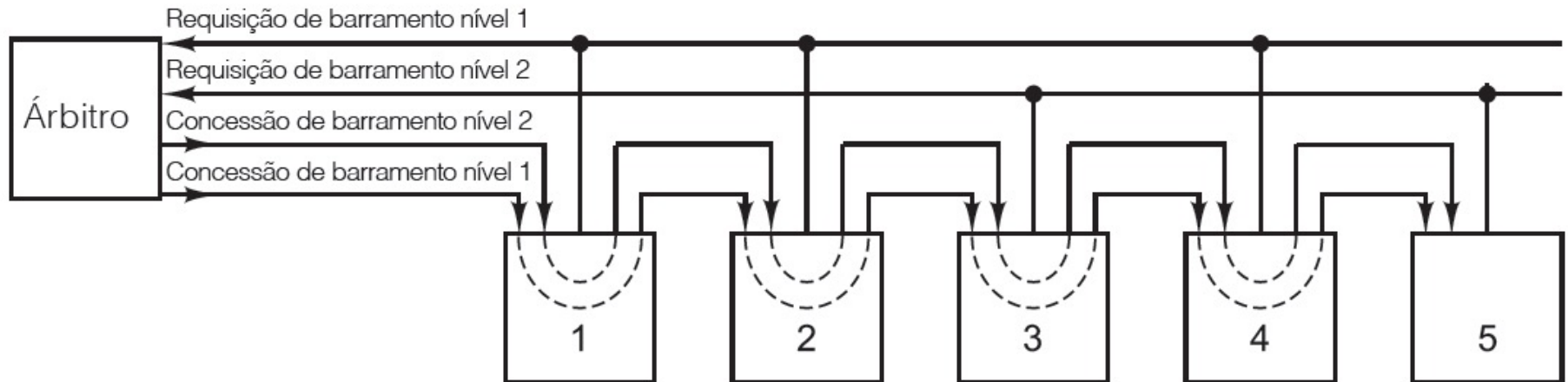
Arbitragem de barramento

- Árbitro de barramento centralizado de um nível usando encadeamento em série.



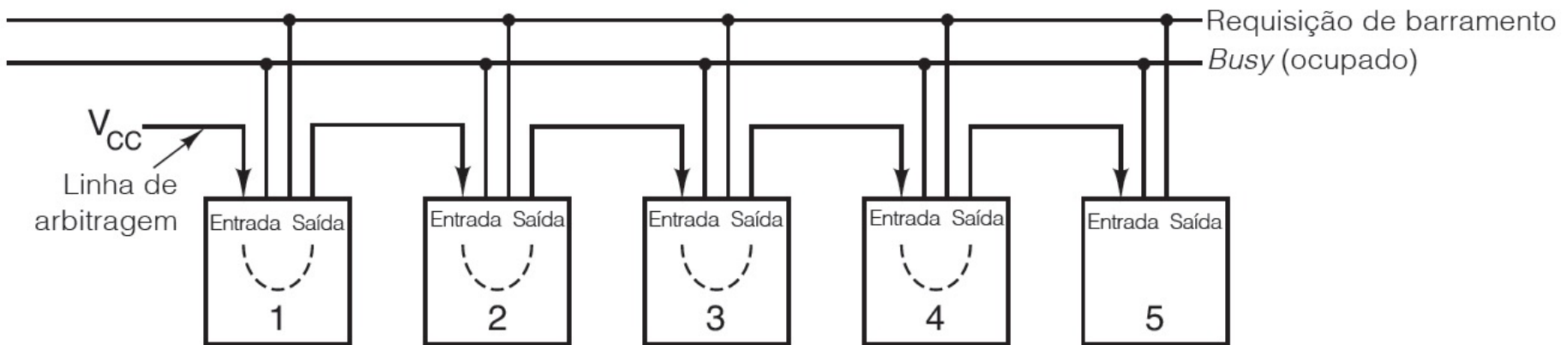
Arbitragem de barramento

- Mesmo árbitro, mas com dois níveis.



Arbitragem de barramento

- Outro tipo de arbitragem de barramento descentralizada, usa apenas três linhas, não importando quantos dispositivos estiverem presentes.



Exemplos de barramentos

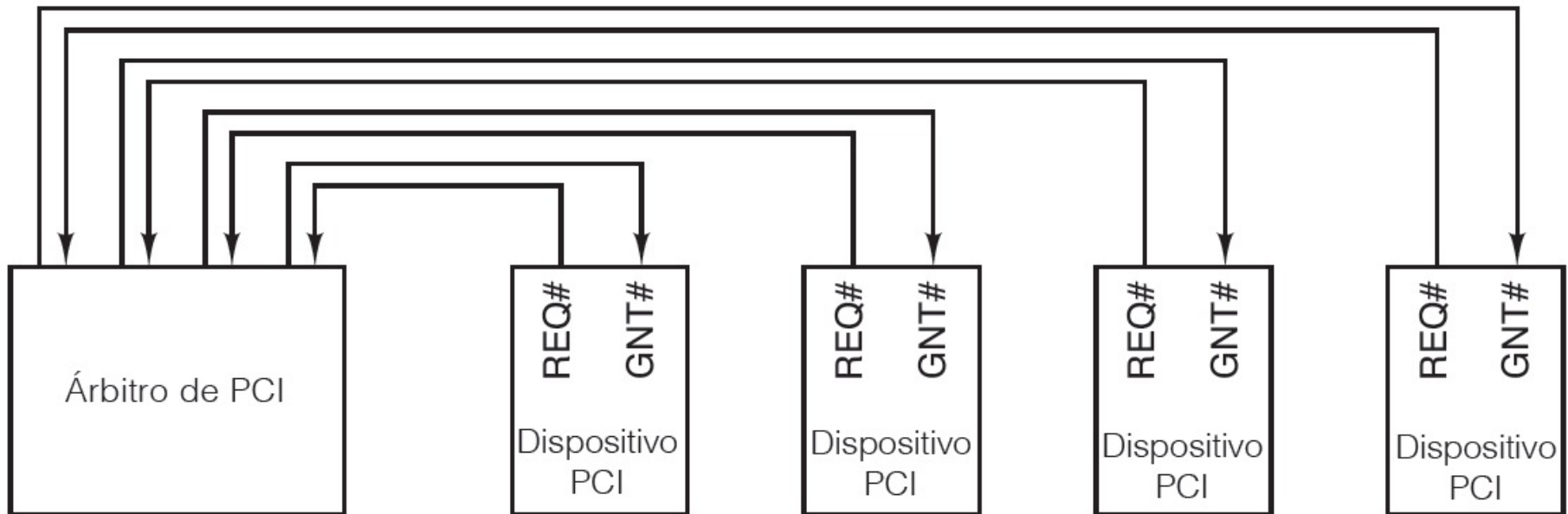
- Barramentos são a cola que mantém a integridade dos sistemas de computadores.
- Alguns barramentos populares são:
 - o PCI e
 - o USB.
- O PCI é o principal barramento de E/S usado hoje em dia nos PCs.

Exemplos de barramentos

- Ele pode ter duas formas:
 - o barramento PCI mais antigo, e
 - o novo e muito mais rápido barramento PCI Express (PCIe).
- O Universal Serial Bus é um barramento de E/S cada vez mais popular para periféricos de baixa velocidade, como mouses e teclados.
- Uma segunda e terceira versões do barramento USB rodam com velocidades muito mais altas.

Exemplos de barramentos

- O barramento PCI usa um árbitro de barramento centralizado.



Exemplos de barramentos

- Sinais obrigatórios de barramento PCI.

Sinal	Linhas	Mestre	Escravo	Descrição
CLK	1			<i>Clock</i> (33 MHz ou 66 MHz)
AD	32	×	×	Linhas de endereço e de dados multiplexadas
PAR	1	×		Bit de paridade de endereço ou dados
C/BE	4	×		Comando de barramento/mapa de bits para bytes habilitados
FRAME#	1	×		Indica que AD e C/BE estão ativadas
IRDY#	1	×		Leitura: mestre aceitará; escrita: dados presentes
IDSEL	1	×		Seleciona espaço de configuração em vez de memória
DEVSEL#	1		×	Escravo decodificou seu endereço e está na escuta
TRDY#	1		×	Leitura: dados presentes; escrita: escravo aceitará
STOP#	1		×	Escravo quer interromper transação imediatamente
PERR#	1			Erro de paridade de dados detectado pelo receptor
SERR#	1			Erro de paridade de endereço ou erro de sistema detectado
REQ#	1			Arbitragem de barramento: requisição de propriedade de barramento
GNT#	1			Arbitragem de barramento: concessão de propriedade de barramento
RST#	1			Restaura o sistema e todos os dispositivos

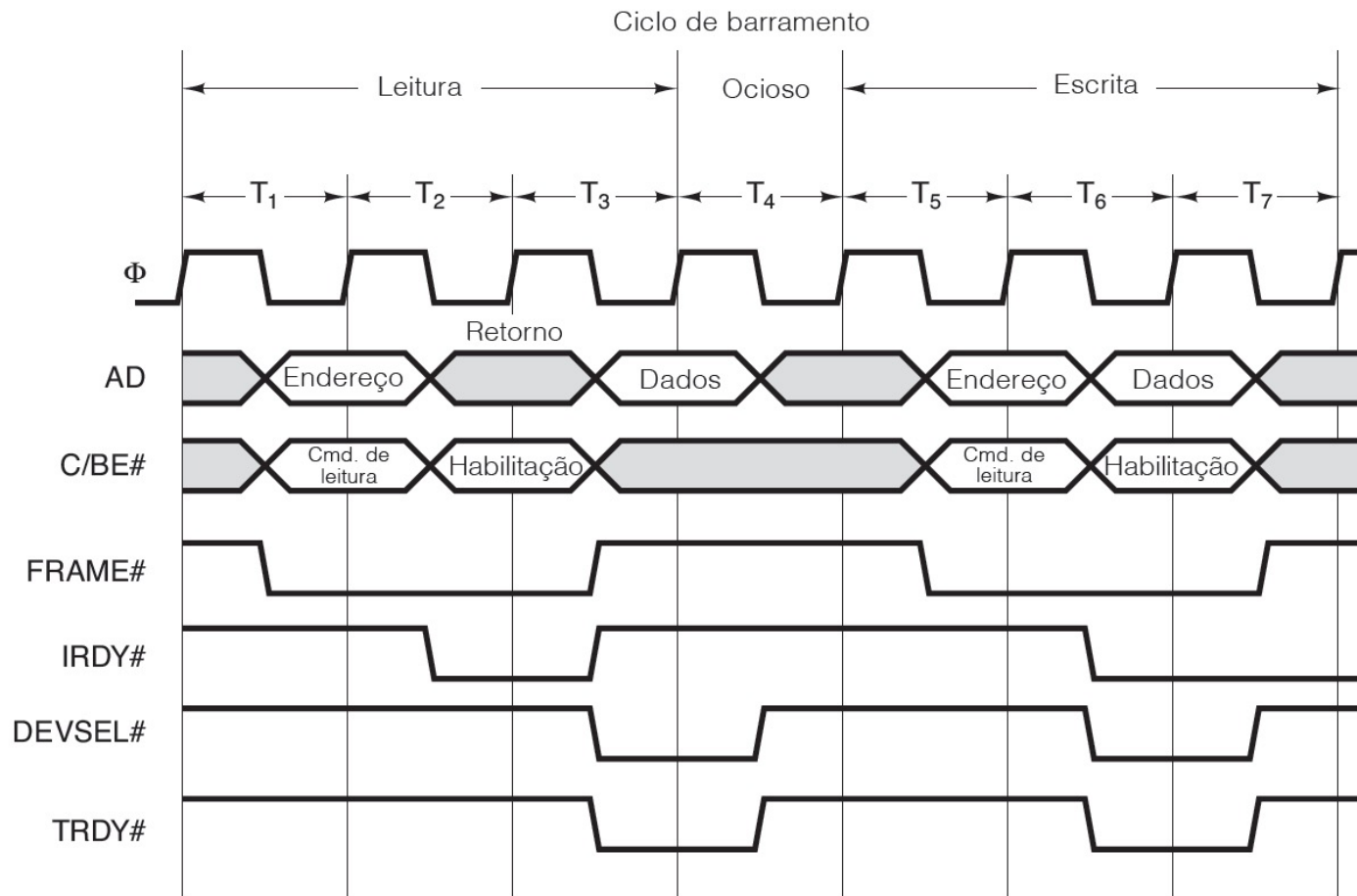
Exemplos de barramentos

- Sinais opcionais de barramento PCI.

Sinal	Linhas	Mestre	Escravo	Descrição
REQ64#	1	×		Requisição para realizar transação de 64 bits
ACK64#	1		×	Permissão concedida para uma transação de 64 bits
AD	32	×		32 bits adicionais de endereço ou dados
PAR64	1	×		Paridade para os 32 bits extras de endereço/dados
C/BE#	4	×		4 bits adicionais para habilitações de bytes
LOCK	1	×		Trava o barramento para permitir múltiplas transações
SBO#	1			Presença de dados em uma <i>cache</i> remota (para um multiprocessador)
SDONE	1			Escuta realizada (para um multiprocessador)
INTx	4			Requisição de uma interrupção
JTAG	5			Sinais de testes IEEE 1149.1 JTAG
M66EN	1			Ligado à energia ou ao terra (66 MHz ou 33 MHz)

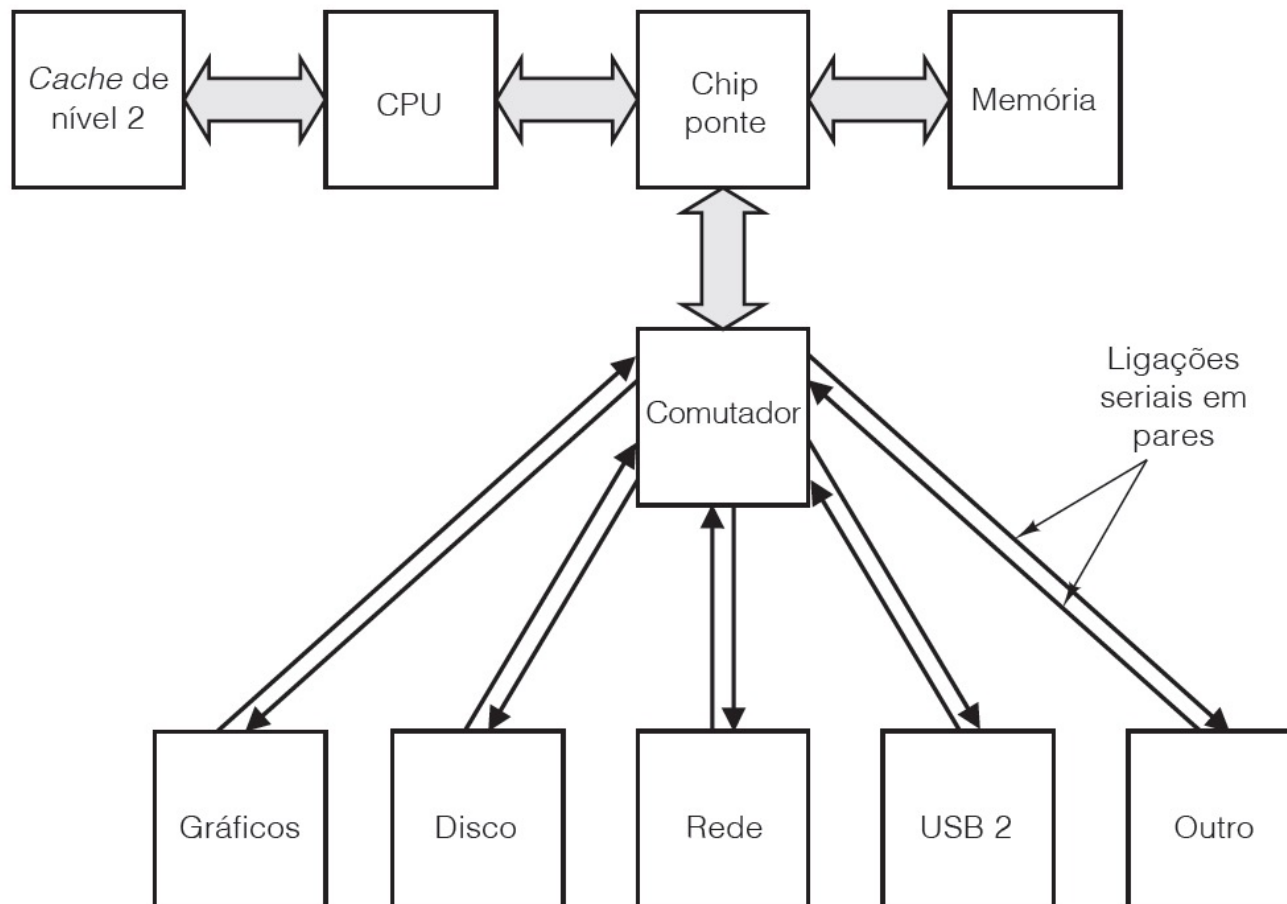
Exemplos de barramentos

- Exemplos de transações de barramento PCI de 32 bits.



Exemplos de barramentos

- Sistema PCI express típico.

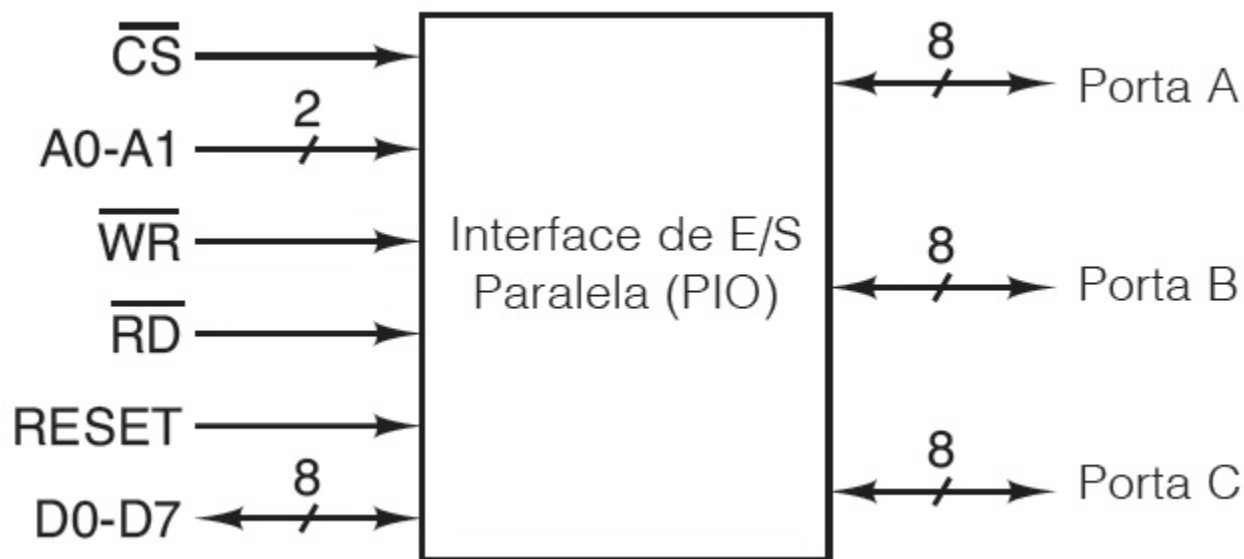


Exemplos de barramentos

- Um sistema USB consiste em um **hub-raiz** (*root hub*) que é ligado ao barramento principal.
- Em termos lógicos, o sistema USB pode ser visto como um conjunto de ramificações que saem do *hub-raiz* para os dispositivos de E/S.
- Exatamente a cada $1,00 \pm 0,05$ ms, o *hub-raiz* transmite um novo quadro para manter todos os dispositivos sincronizados em relação ao tempo.
- O USB suporta quatro tipos de quadros: de controle, isócrono, de volume e de interrupção.

Interfaces PIO

- Uma interface PIO (**Parallel Input/Output – entrada e saída paralela**) típica é o Intel 8255A:

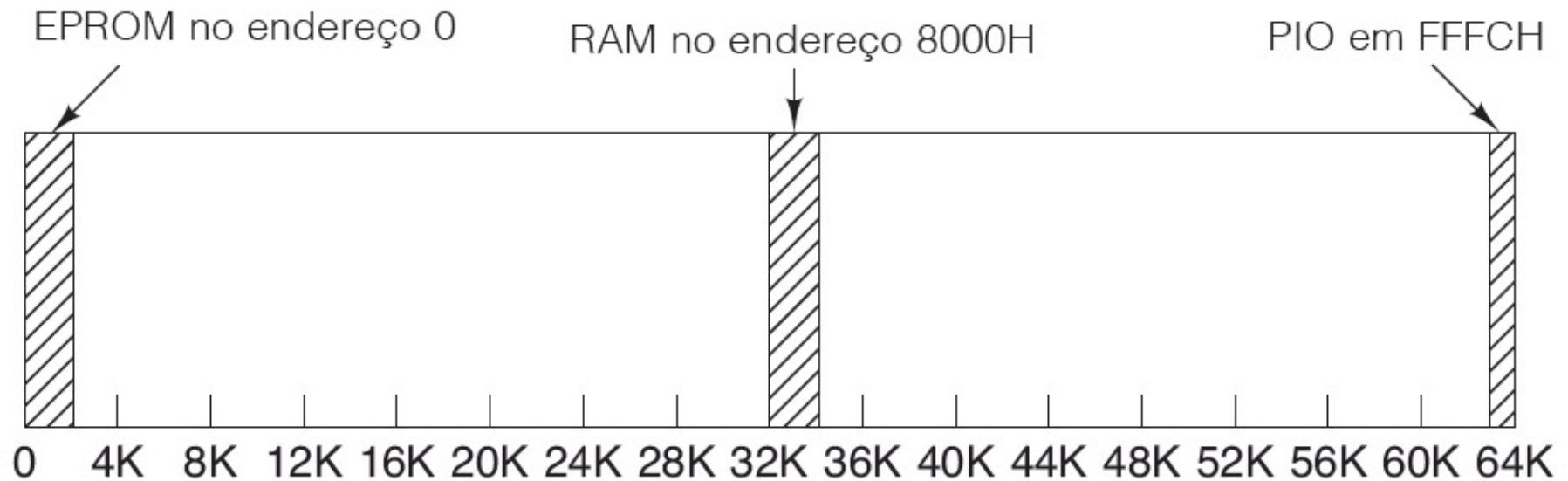


Interfaces PIO

- A interface PIO pode ser selecionada de um entre dois modos: como um verdadeiro dispositivo de E/S ou como parte da memória.
- Uma opção possível é mostrada na figura a seguir.
- A EPROM ocupa endereços até 2 K, a RAM ocupa endereços de 32 KB a 34 KB e a PIO ocupa os 4 bytes mais altos do espaço de endereço, 65.532 a 65.535.
- Com as designações de endereço da figura, a EPROM deve ser selecionada por quaisquer endereços de memória de 16 bits da forma 00000xxxxxxxxxxx (binário).

Interfaces PIO

- Localização da EPROM, RAM e PIO em nosso espaço de endereço de 64 KB.

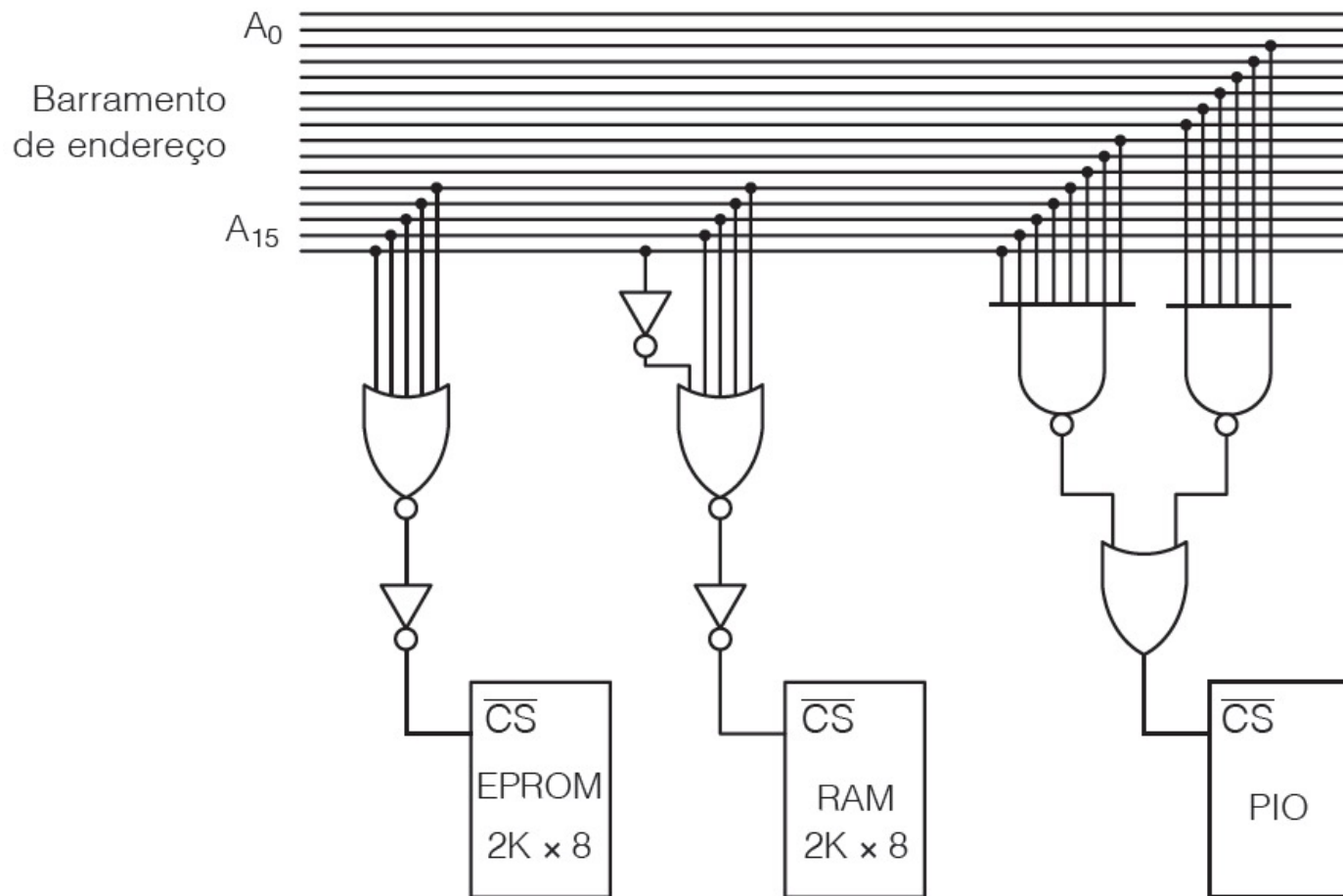


Interfaces PIO

- O mesmo princípio pode ser usado para a RAM.
- Contudo, a RAM deve responder a endereços binários da forma 10000xxxxxxxxxxxxx.
- Se o computador tiver apenas uma CPU, dois chips de memória e a PIO, podemos usar um truque para conseguir uma decodificação de endereço muito mais simples.
- Esse truque se baseia no fato de que todos os endereços da EPROM, e somente endereços da EPROM, têm um 0 no bit de ordem alta, A15. Por conseguinte, basta ligar \overline{CS} a A15 diretamente.

Interfaces PIO

- Decodificação total de endereço.



Interfaces PIO

- Decodificação parcial de endereço.

