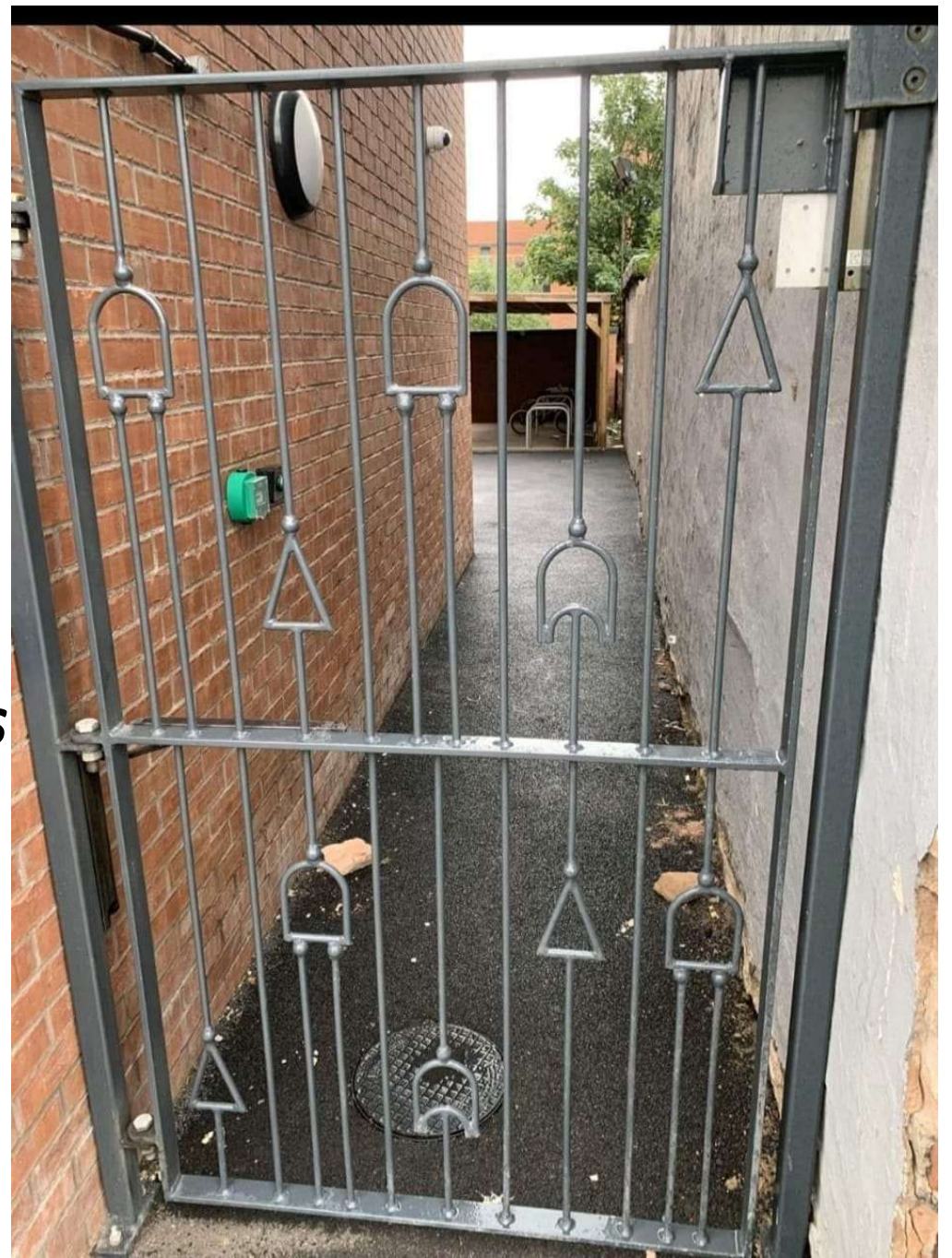




Aula 4

Portas lógicas e suas funções



Os elementos básicos de construção de circuitos digitais (ex.: microprocessador) são denominados portas lógicas.

As portas lógicas executam operações sobre operadores binários (0 e 1), podendo ser aplicadas para diversas finalidades. As principais portas são: AND, OR, NOT e XOR.

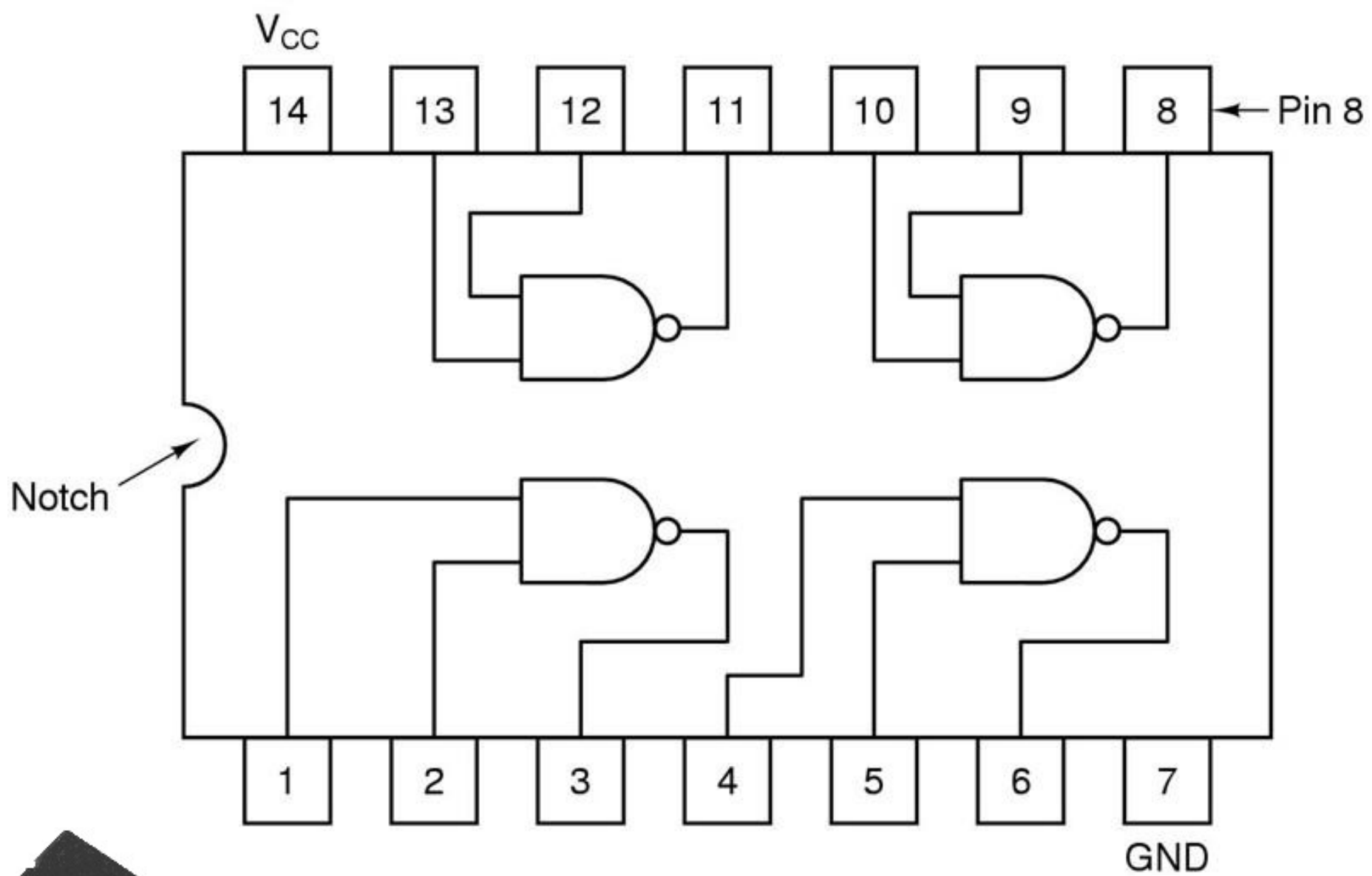
Quando agrupadas formam os chamados circuitos combinatórios que, quando possuem uma memória acoplada, são chamados de sequenciais. Esses circuitos digitais podem representar, por exemplo, a Unidade Lógica Aritmética (ULA).

Um computador é constituído de elementos eletrônicos, como resistores, capacitores e principalmente transistores.

Os transistores são, em geral, componentes de determinados circuitos eletrônicos que precisam armazenar os sinais binários e realizar certos tipos de operações com eles.

Esses circuitos, chamados de circuitos digitais, são formados por pequenos elementos capazes de manipular grandezas apenas binárias.

Eles são conhecidos como portas lógicas, por permitirem ou não a passagem desses sinais, e os circuitos que contêm as portas lógicas são conhecidos como circuitos lógicos.



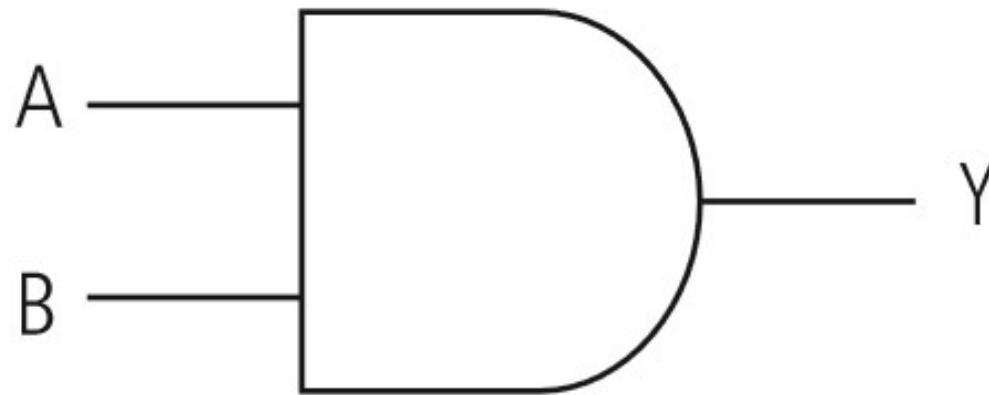
Em geral, os circuitos lógicos são agrupados e embutidos em um Circuito Integrado (CI). Esses dispositivos implementam uma determinada função com o objetivo de cumprir uma tarefa específica.

Uma operação lógica qualquer (ex.: soma ou multiplicação de dígitos binários) sempre irá resultar em dois valores possíveis: 0 (falso) ou 1 (verdadeiro). Assim, pode-se pré-definir todos os possíveis resultados de uma operação lógica, de acordo com os possíveis valores de entrada.

Para representar tais possibilidades, utiliza-se de uma forma de organizá-las chamada Tabela Verdade. Assim, cada operação lógica possui sua própria tabela verdade.

Porta AND

Trata-se de uma operação que aceita dois operandos ou duas entradas (A e B)

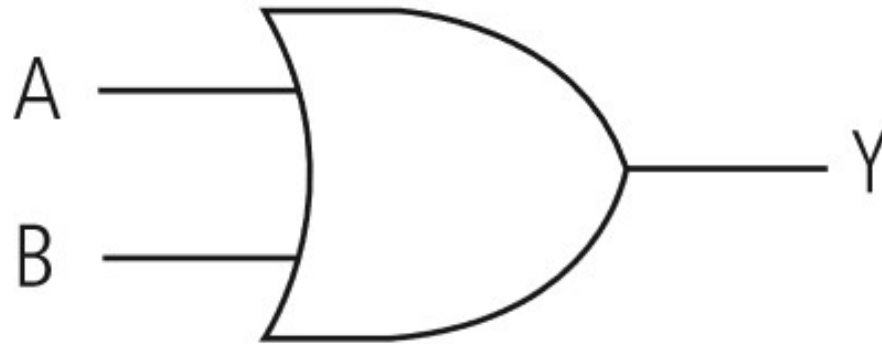


Pode-se dizer que a operação AND simula uma multiplicação binária, permitindo os possíveis resultados

Entrada		Saída
A	B	$X = A.B$
0	0	0
0	1	0
1	0	0
1	1	1

Porta OR

Trata-se de uma operação que aceita dois operandos ou duas entradas (A e B).



Podemos dizer que a operação OR simula uma soma de binários, permitindo os possíveis resultados.

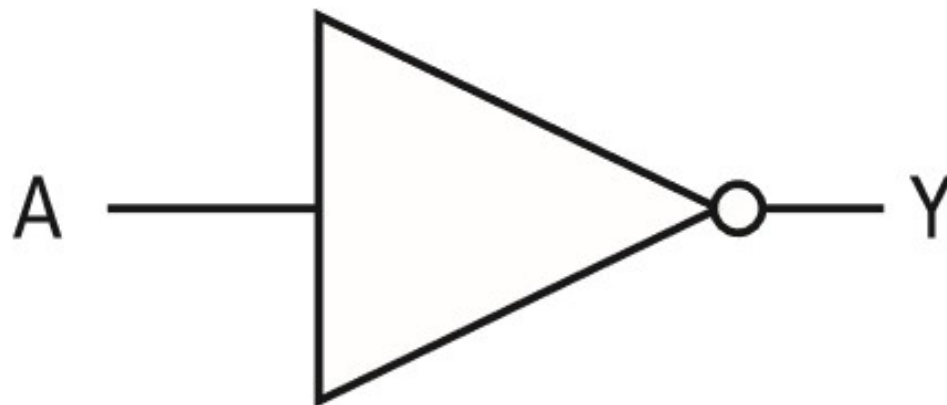
Entrada		Saída
A	B	$X = A+B$
0	0	0
0	1	1
1	0	1
1	1	1

Conforme podemos observar, a regra é: “se o primeiro operando é 1 ou o segundo operando é 1, ou se os dois operandos forem 1, o resultado é 1, senão o resultado é 0”.

As operações lógicas OR são muito utilizadas em lógica digital ou mesmo em comandos de decisão de algumas linguagens de programação (ex.: Se (X=1 OU Y=1) Então Executa uma ação).

Porta NOT

A porta NOT representa um inversor. Essa operação aceita apenas um operando uma entrada (A).



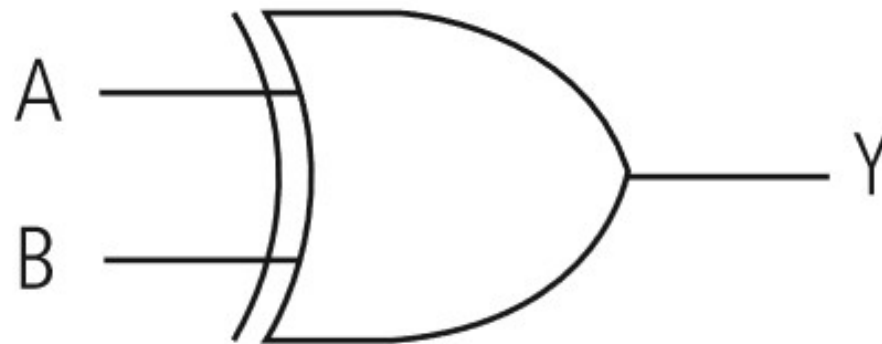
Entrada	Saída
A	$X = \bar{A}$
0	1
1	0

Conforme é possível observar, a regra é: “se o operando for 1, o resultado é 0, senão o resultado é 1”.

Entre as principais aplicações dos circuitos inversores está a representação de números negativos no formato binário, quando se usa o que é chamado de complemento a 1 ou complemento a 2, fazendo-se necessária a inversão de um grupo de bits representativos de um número negativo.

Porta XOR

A denominação XOR é a abreviação do termo EXCLUSIVE OR. Trata-se de uma operação que aceita dois operandos ou duas entradas (A e B). Os operandos são binários simples (0 e 1).



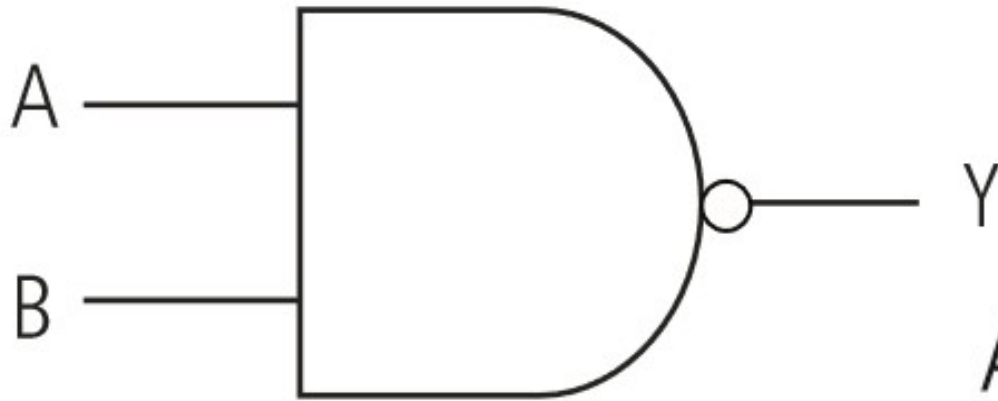
Entrada		Saída
A	B	$X = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Conforme é possível observar, a regra é: “se o primeiro operando ou o segundo operando for igual a 1, o resultado é 1; senão, o resultado é 0”. Ou seja, para entradas iguais a saída será 0 e para entradas diferentes a saída será 1.

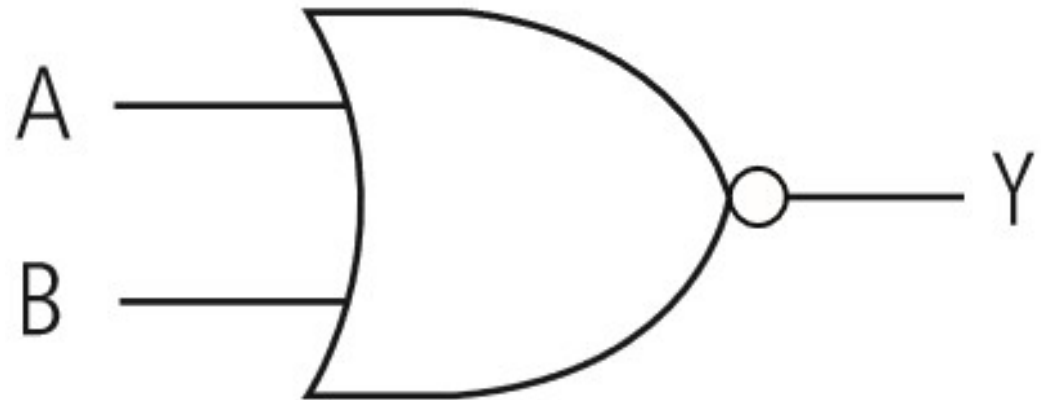
A porta XOR possui inúmeras aplicações, sendo um elemento lógico bastante versátil, permitindo, por exemplo, a fabricação de um testador de igualdade entre valores, para testar, de modo rápido, se duas palavras de dados são iguais.

Outras portas lógicas e circuitos combinatórios

Existem outras portas lógicas derivadas das portas lógicas apresentadas acima, tais como as portas NAND (porta AND invertida) e a porta NOR (porta OR invertida).



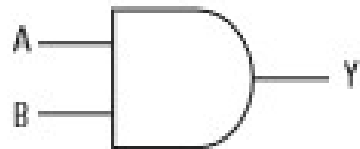
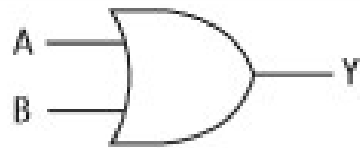
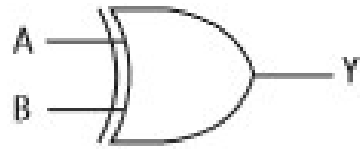
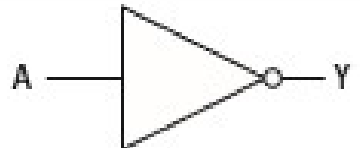
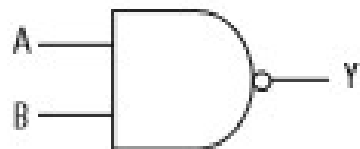

Porta lógica NAND (NÃO E)



Porta lógica NOR (NÃO OU)

As operações lógicas são realizadas em dois passos: primeiro a operação AND ou OR e, em seguida, o seu resultado é invertido. Esse tipo de portas lógicas também possui diversas aplicações, sendo utilizado para reduzir a complexidade e a quantidade de portas lógicas necessárias a um determinado circuito lógico.

Não esqueça de que um circuito lógico pode possuir diversas portas lógicas e, portanto, suas tabelas verdade poderão ter inúmeras entradas e inúmeras saídas (as quais poderão ser representadas por suas respectivas equações booleanas).

Função Lógica Básica	Símbolo Gráfico da Porta	Equação Booleana
AND		$Y = A.B$
OR		$Y = A+B$
XOR		$Y = A \oplus B$
NOT		$Y = \bar{A}$
NAND		$Y = \overline{A.B}$
NOR		$Y = \overline{A+B}$

A partir das portas lógicas básicas, é possível interligar diversas de suas unidades, de modo a construir redes lógicas, também chamadas de circuitos combinatórios.

Monteiro (2007) explica que um circuito combinatório é definido como um conjunto de portas lógicas cuja saída em qualquer instante de tempo é função somente das entradas. O autor afirma ainda que existe outra categoria de circuitos que combina portas lógicas, denominada circuitos sequenciais, os quais, além de possuir portas, contêm elementos de armazenamento (uma espécie de memória).

1. Desenhe os circuitos que implementam as seguintes expressões booleanas.

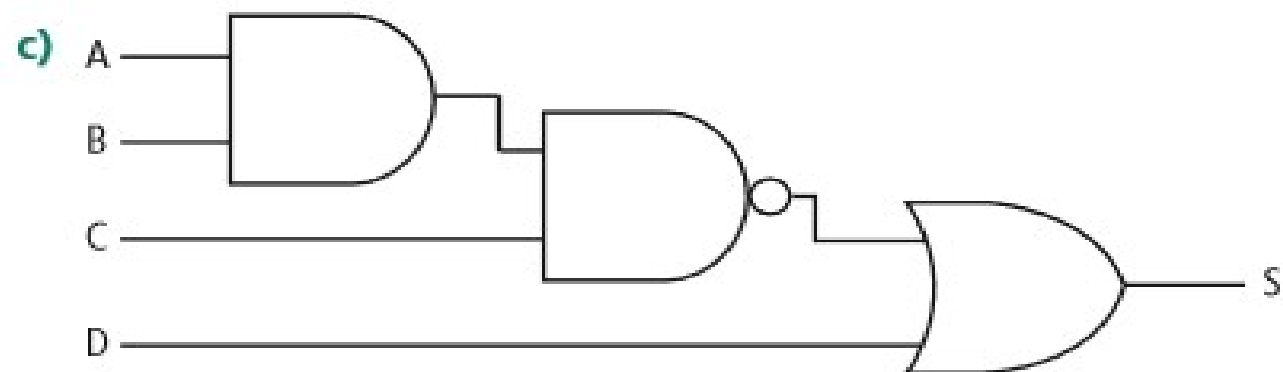
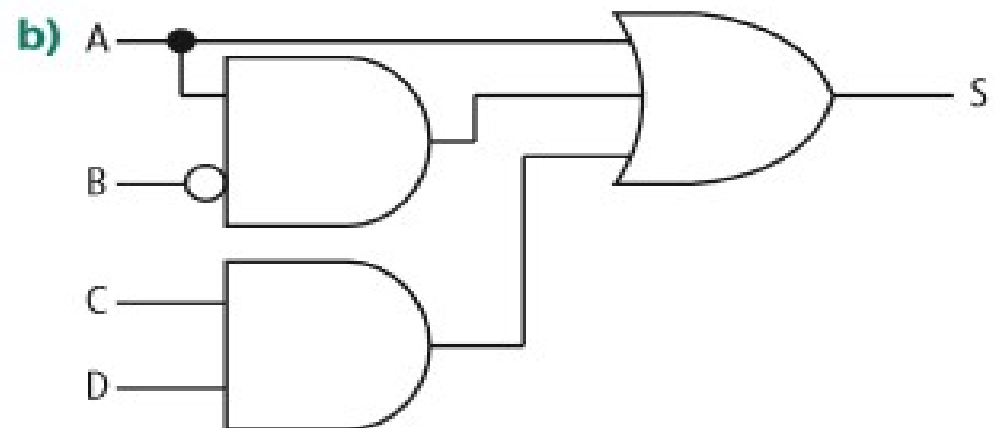
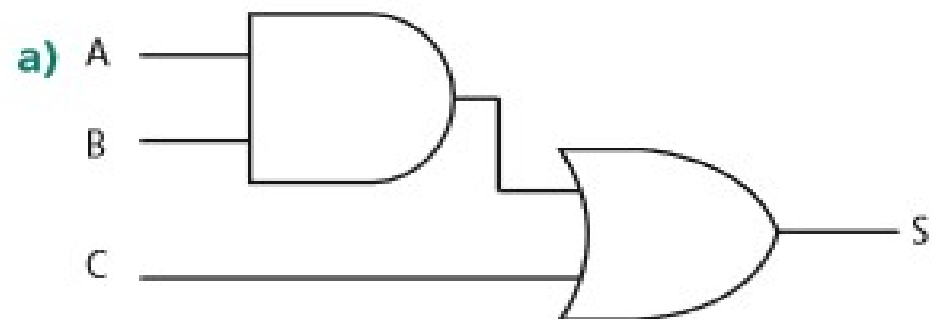
a) $S = \overline{A\bar{B} + \bar{C}D}$

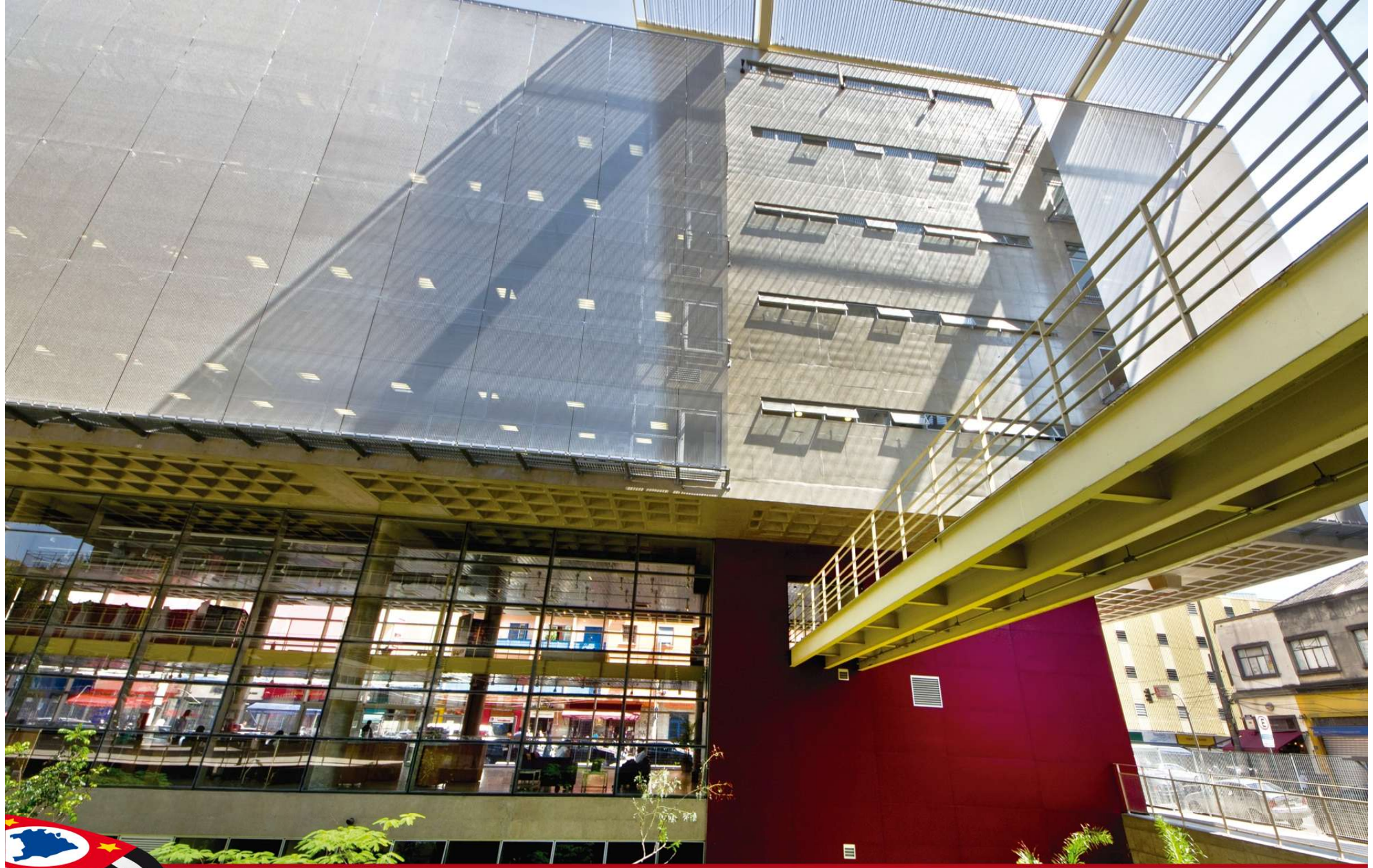
b) $S = \overline{(ABC\bar{C} + \bar{C}D) \oplus D}$

c) $S = A\bar{B} + (CD\bar{E})$

d) $S = A + (B + C\bar{D})(B + A)$

2. Escreva a expressão booleana executada pelos circuitos abaixo:





CPS
Centro
Paula Souza

 **GOVERNO DO ESTADO
SÃO PAULO**

Aula 5

Flip-Flops

Flip-flop

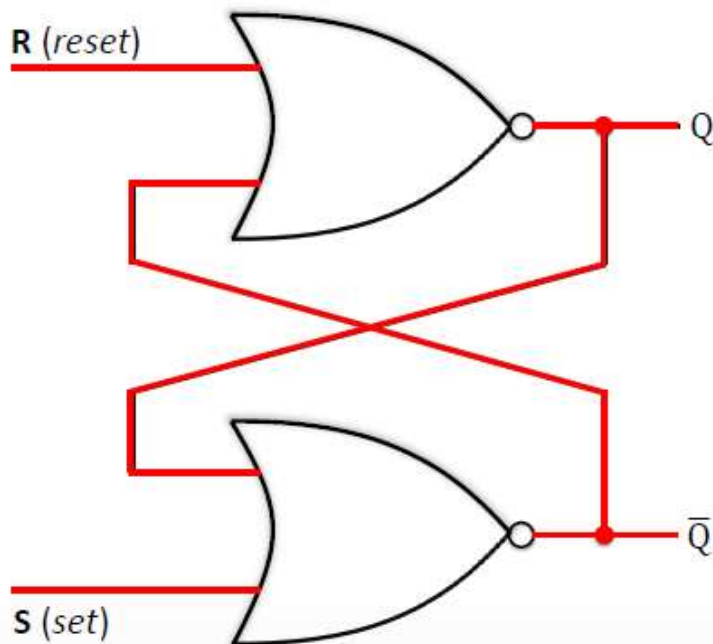
Para a adicionar a ideia de memória ao circuito combinatório, foi inventado o elemento de armazenamento chamado de Flip-Flop, que pode armazenar 1 bit de informação.

Um conjunto de flip-flops pode ser utilizado para armazenar um conjunto de bits.

Alguns tipos: SR, D, JK e T

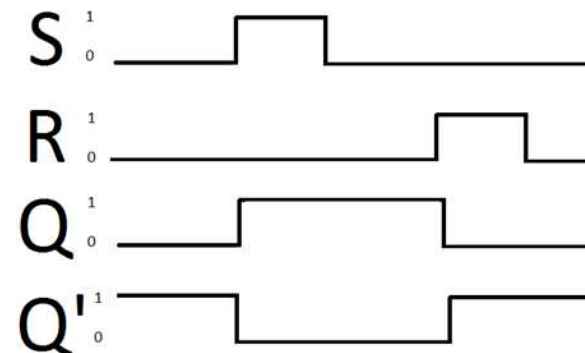
Flip-flop SR

É o mais simples. Possui duas entradas S (set) e R (reset). Possui duas saídas, Q e Q', que são sempre complementos uma da outra.



S	R	Q _{next}	Ação
0	0	Q	Mantém o estado das saídas
0	1	0	Limpa a saída Q
1	0	1	Set (ativa) Q
1	1	X	Não permitido

S	R	Q(T)	Q(T+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	INDEFINIDO
1	1	1	INDEFINIDO



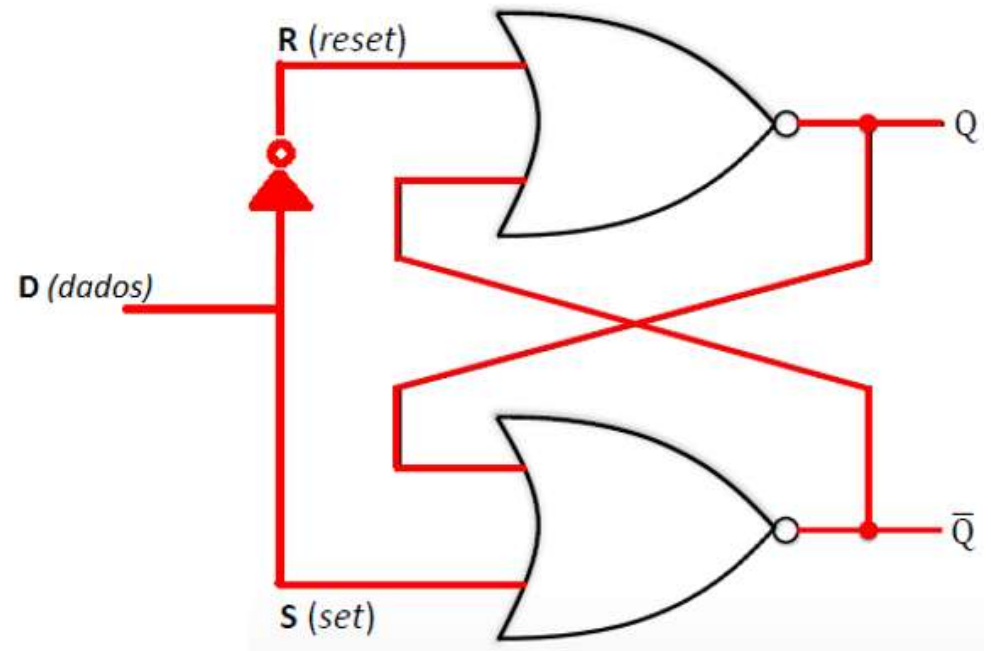
Flip-flop D

Flip-flops SR não podem ser utilizados como memória de 1 bit, pois precisa de 2 entradas. Uma pequena modificação no circuito é capaz de criar um flip-flop D – dados.

Note que a saída do flip-flop é igual a sua entrada. No entanto, esta saída permanece inalterada até que a próxima entrada seja fornecida.

Ou seja, é capaz de memorizar os estados de entrada

D	Q(T+1)
0	0
1	1

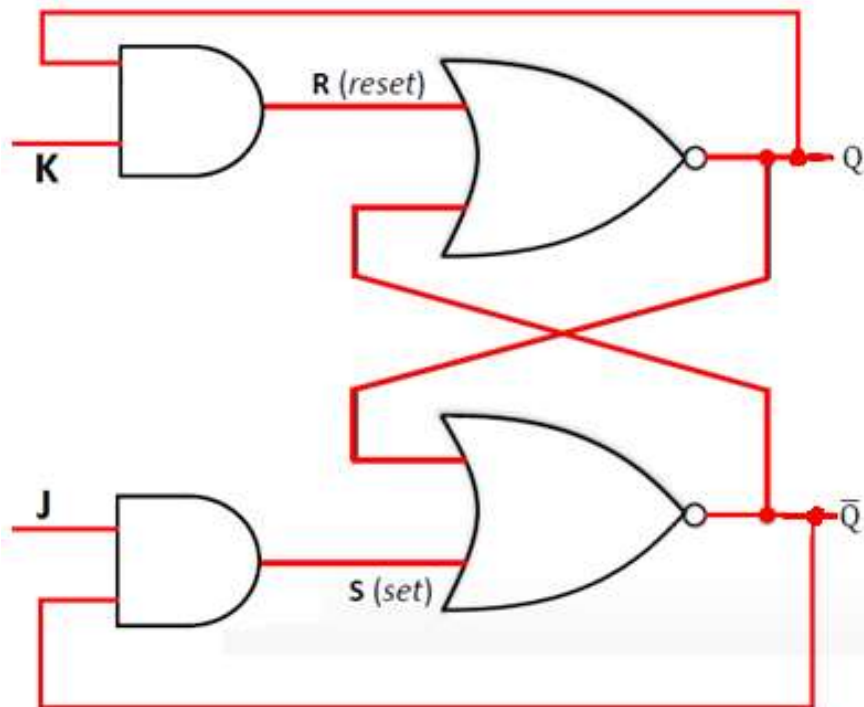


Flip-flop JK – Jack Kilby

Criado para remover o estado indefinido do flip-flop SR

É obtido com a adição de duas portas AND a um flip-flop SR

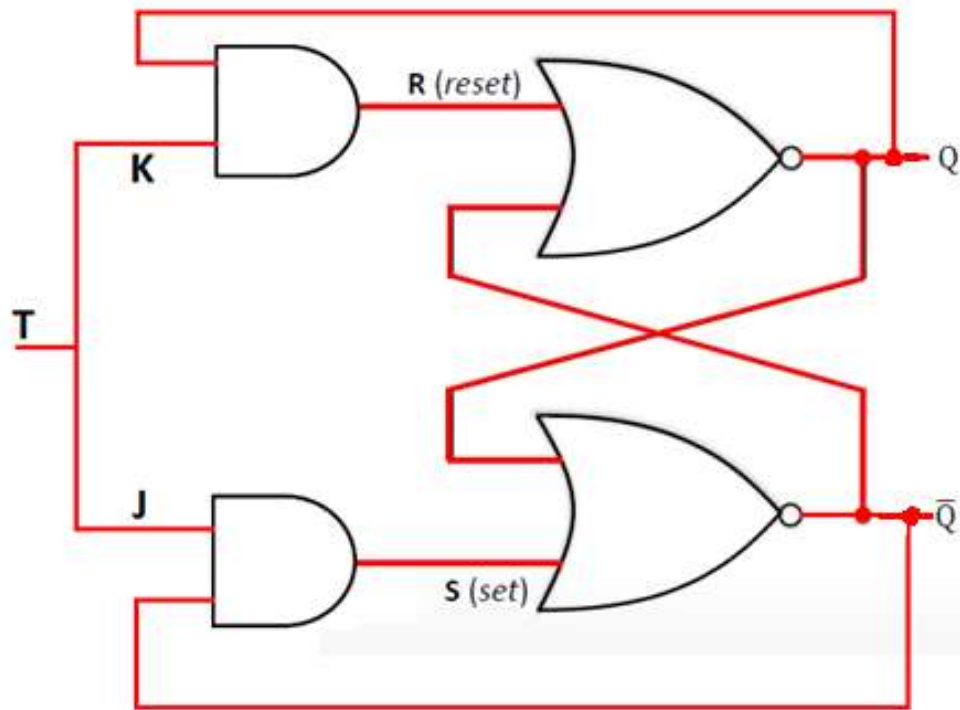
Jack St. Claire Kilby foi um físico e engenheiro eletricitista estadunidense. No final dos anos 50, ao ser contratado pela Texas Instruments, produziu um chip de cristal que reunia componentes antes separados, como transístores, resistores e capacitores



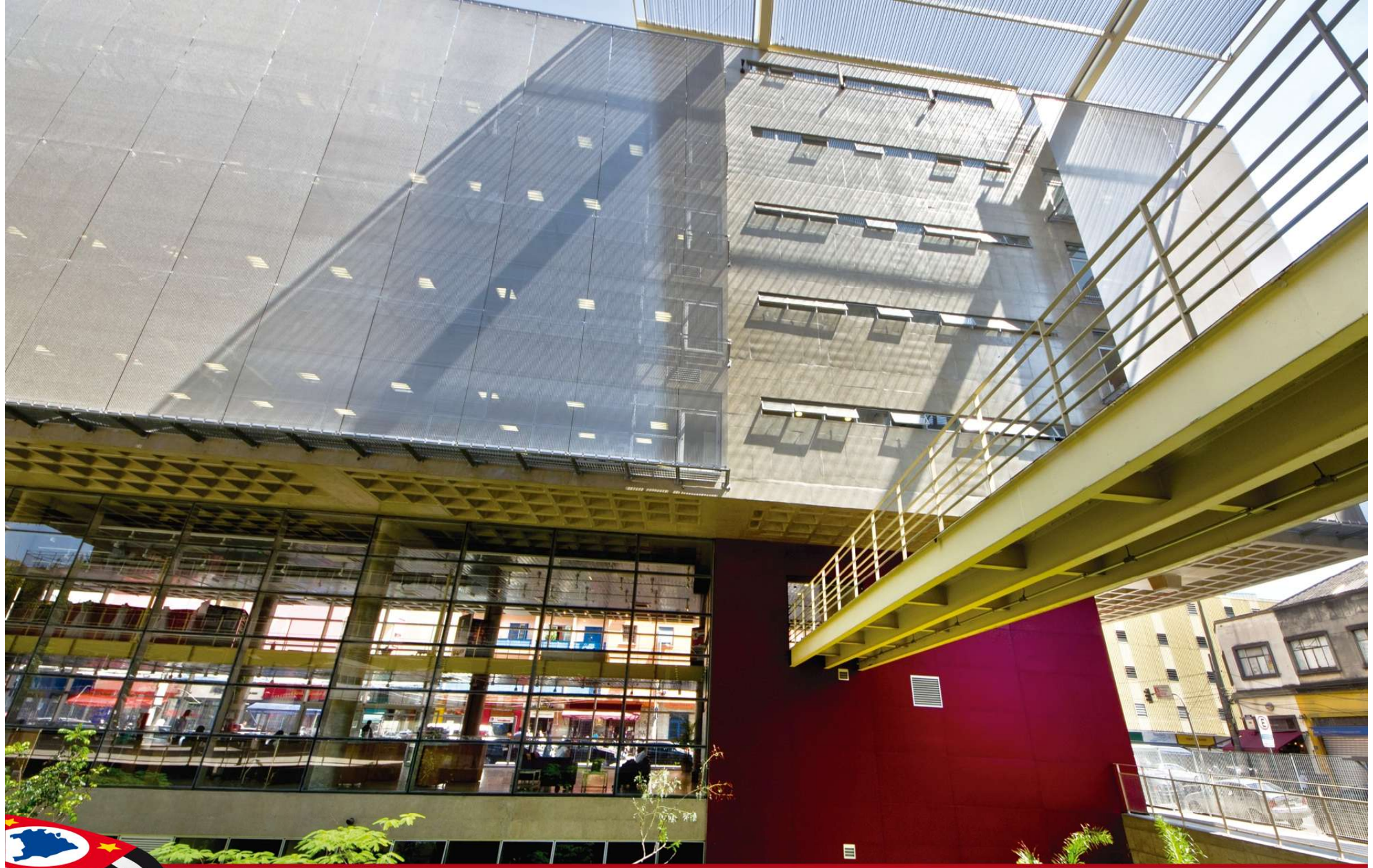
J	K	$Q(T+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$Q'(t)$

Flip-flop T – toggle ou alternar

- Se T estiver em estado alto, o flip-flop T (toggle) muda o estado. Se a entrada T for baixa, o flip-flop mantém o valor do seu estado. Seu comportamento é descrito pela tabela:



T	$Q(T+1)$
0	$Q(t)$
1	$Q'(t)$



CPS
Centro
Paula Souza

 **GOVERNO DO ESTADO
SÃO PAULO**

Aula 6

Modo de Endereçamento

Endereçamento de Memória

Em um sistema de computação temos a UCP se comunicando com a memória e os dispositivos de E/S. Podemos fazer um paralelo do acesso à memória com um carteiro entregando cartas ou um acesso à biblioteca.

Duas ações básicas podem ser executadas:

- armazenamento (escrita ou gravação (write))**
- recuperação (leitura (read))**

Em sistemas de computação a unidade básica é o bit que pode ter 2 valores. Mas precisamos de mais valores para representar informação: números, letras do alfabeto, símbolos. Para isso, os bits são agrupados em grupos de bits que são armazenados na memória e são acessados sempre em grupo. Para acessar um grupo de bits, utiliza-se um endereço que varia de 0 a (N-1) onde N é o número total de grupos existentes.

Endereçamento de Memória

end 0	M bits

end 1	M bits

end 2	M bits

	.

end N-1	b_{m-1} b_{m-2} ... b_7

Endereçamento de Memória

- Dados e instruções são armazenados na memória, lidos e escritos pela UCP
- Memória organizada em células de M bits, mas acesso depende do tamanho de palavra utilizado pela máquina
- Cada grupo de bits (célula) possui um endereço associado. Organização utilizada atualmente: cada célula possui 8 bits e o tamanho da palavra varia entre 16, 32 e 64 bits
- Tamanho da palavra: Quantidade de bits transferida entre a Memória e UCP em operações de leitura e escrita

Endereçamento de Memória

Neste exemplo temos $M = 8$ bits de palavras (células de memória)

Temos ainda 4 endereços, numerados de 0 a 3, que tem como objetivo endereçar cada célula de memória

N Endereços	M células de memória
-------------	----------------------

00	8 bits
----	--------

01	8 bits
----	--------

10	8 bits
----	--------

11	8 bits
----	--------



$2^2 = 4$ endereços

$e = 2$ bits de endereçamento

Quantos bits de endereçamento possui essa memória?

$2^e = 4$ endereços (células de memórias / linhas)

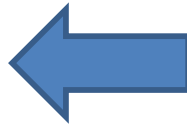
$2^2 = 4$ endereços, portanto, 2 bits de endereçar

Qual é a capacidade de memória?

$$T = N \times M$$

$$N = 2^e$$

Temos, $T = 4 \times 8 = 32$ bits ou 4 bytes



Endereçamento de Memória

Espaço de endereçamento = 8K células e cada célula pode armazenar 8 bits. Qual a capacidade da Memória e quantas linhas de endereço devem ser utilizadas?

$$T = N \times M = 2^{13} \times 2^3 = 2^{16} = 64K \text{ bits}$$

$N = 2^{13}$, logo teremos que ter 13 linhas para endereço

Um computador possui capacidade máxima de armazenamento de 512K células, cada uma de 16 bits.

Qual o número mínimo de bits que o barramento de endereços e de dados devem ter, supondo que duas células são acessadas em cada operação de leitura/ou escrita? Qual o maior endereço da Memória e qual a capacidade da Memória?

Se temos 512 K células = $2^9 \times 2^{10} = 2^{19}$, teremos 19 bits para o barramento de endereços.

Se em cada acesso à memória, duas células são transferidas, então o barramento de dados deve ter o tamanho igual a 32 bits.

O maior endereço da Memória é $N-1 = 2^{19}-1 = 524.287$

Capacidade da Memória = $N \times M = 2^{19} \times 16 = 2^{23} \text{ bits} = 2^{20} \text{ bytes} = 1M \text{ bytes}$

