ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

O nível lógico digital

Prof. Me. Joseph Soares Alcântara

ENGENHARIA DA COMPUTAÇÃO

2018.2

UNIVERSIDADE FEDERAL DO CEARÁ

PROGRAMA DA DISCIPLINA

1) Introdução à Arquitetura e Organização de Computadores
2) Conceitos básicos de computadores
3) Organização de sistemas de computadores
4) O nível lógico digital
5) Conjunto de instrução
6) Programação em linguagem de montagem

O NÍVEL LÓGICO DIGITAL

- Introdução
- Portas e álgebra booleana
- Circuitos integrados
- Circuitos combinatórios
- Circuitos aritméticos
- *Clocks*
- Memórias de 1 bit
- *Flip-flops*
- Registradores
- ...

INTRODUÇÃO

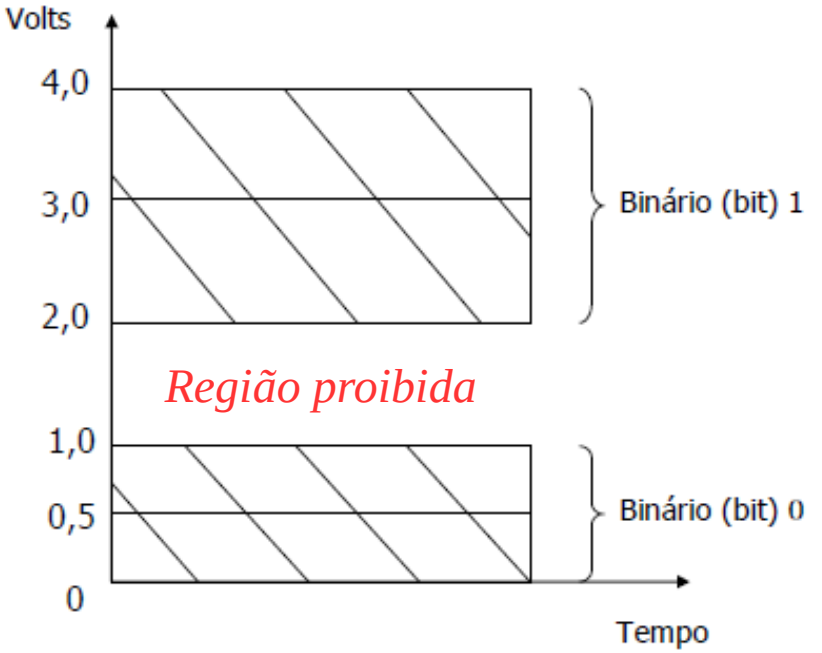
Circuito digital – dois valores lógicos
Transistores – Comutador binário

CIRCUITO DIGITAL

- A informação binária (0 ou 1) é representada em um sistema digital por sinais elétricos em dois níveis de intensidade, cada um correspondendo a um valor binário:

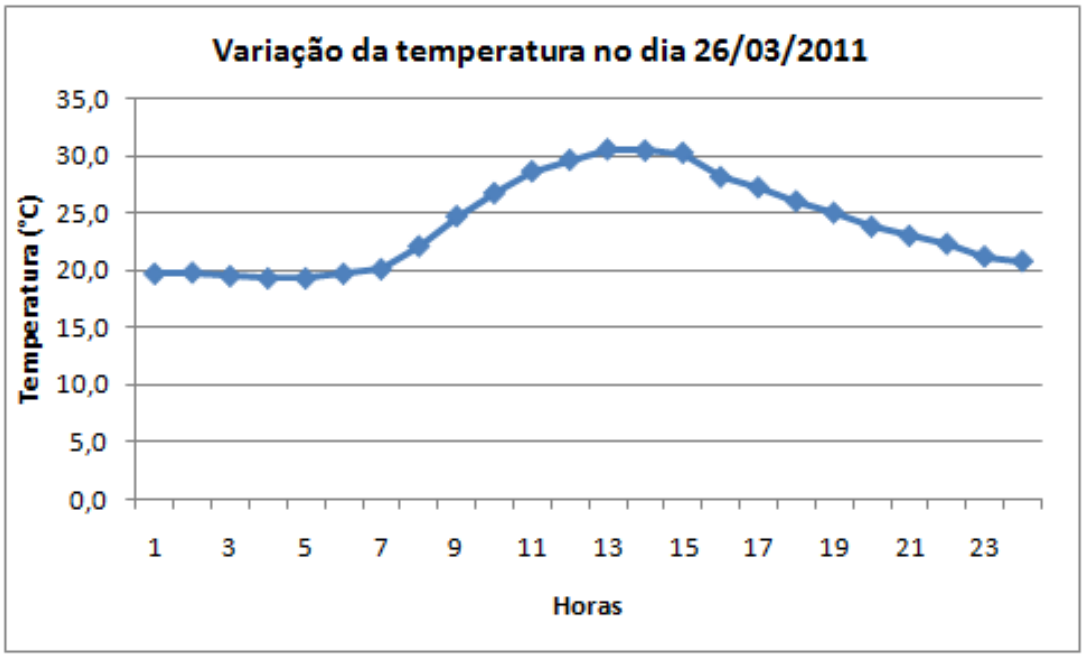
  0 → 0,5 V variando de [0,0:1,0] V

  1 → 3,0 V variando de [2,0:4,0] V.



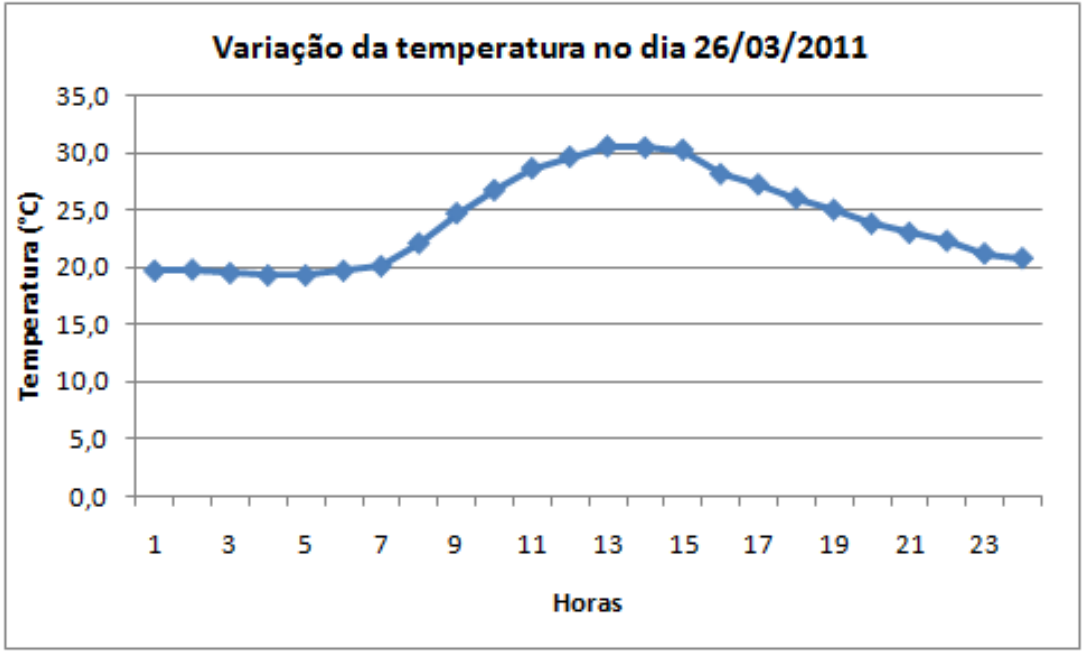
GRANDEZAS ANALÓGICA E DIGITAL

- Uma grandeza **analógica** é aquela que apresenta valores contínuos
  - Ex.: a temperatura do ambiente varia numa faixa contínua de valores – durante um determinado dia, a temperatura não passa, digamos, de 24ºC para 25º C instantaneamente; ela passa por uma infinidade de valores intermediários



GRANDEZAS ANALÓGICA E DIGITAL

- Uma grandeza **digital** é aquela que apresenta valores discretos
  - Ex.: Se fizermos a leitura da temperatura apenas a cada hora, aí sim podemos ter variações instantâneas (no instante da leitura) de um valor inteiro para outro – de 7 para 9 horas, a temperatura muda de 20º C para 25º C

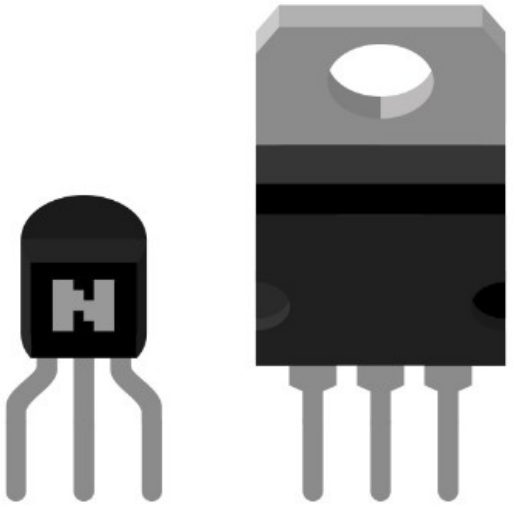


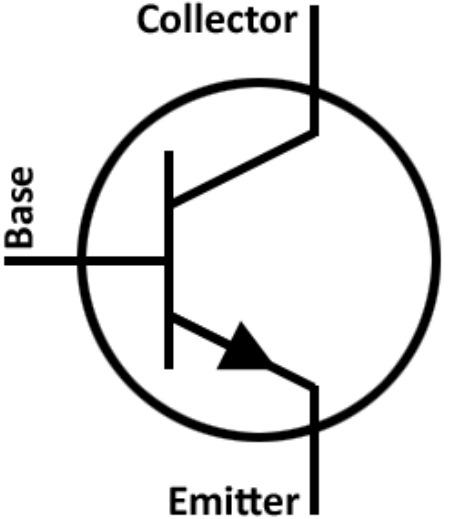
GRANDEZAS ANALÓGICA E DIGITAL

O termo digital é derivado da forma com que os computadores realizam operações: contando dígitos

TRANSISTOR

- Pode funcionar como um amplificador ou chave
  - No modo amplificador ele possui a capacidade de ampliar o nível de tensão
  - Como chave ele permite ligar cargas em sua saída
- Apenas tensão contínua



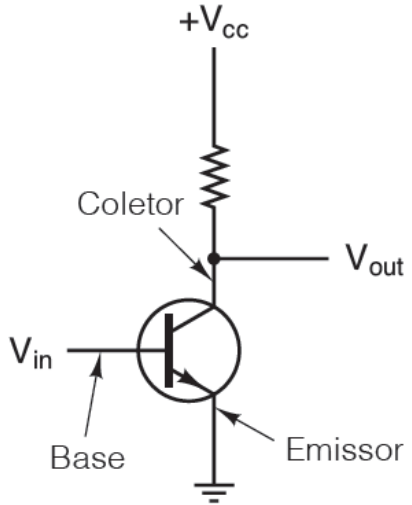


PORTAS E ÁLGEBRA BOOLEANA

Portas (*gates*) lógicas e a álgebra de Boole

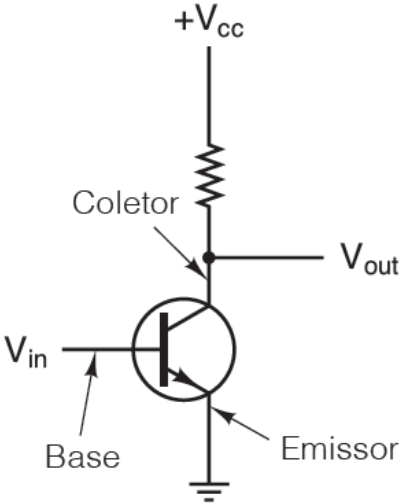
PORTAS

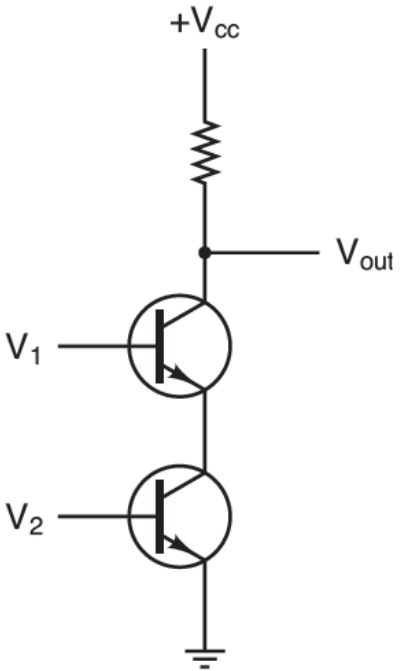
- Minúsculos dispositivos eletrônicos, denominados portas (gates), podem calcular várias funções dos sinais



PORTAS

- Minúsculos dispositivos eletrônicos, denominados portas (gates), podem calcular várias funções dos sinais

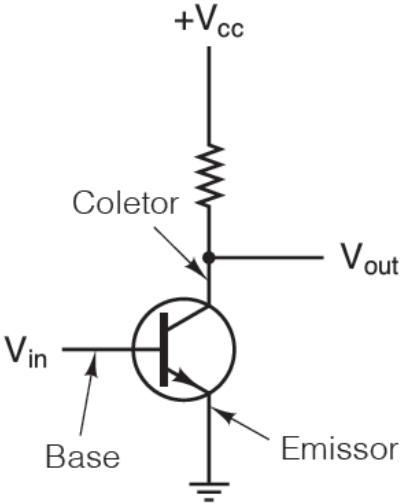


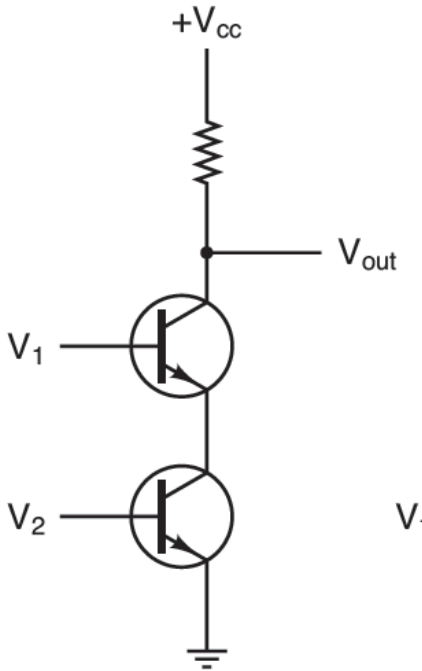


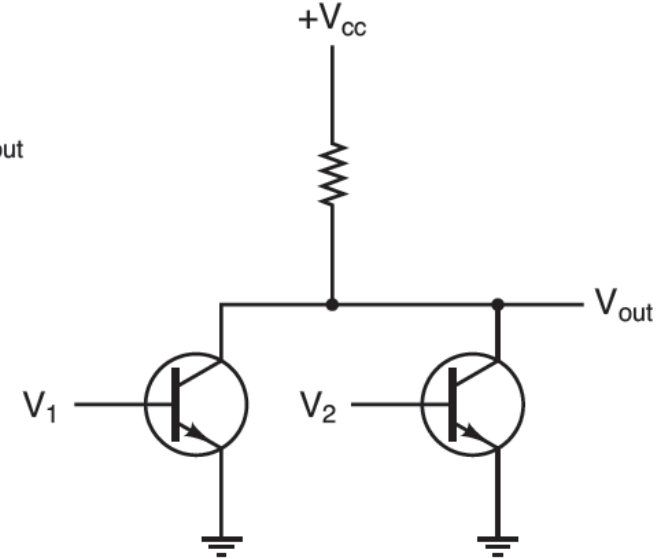
Inversor de transistor Porta NAND

PORTAS

- Minúsculos dispositivos eletrônicos, denominados portas (gates), podem calcular várias funções dos sinais



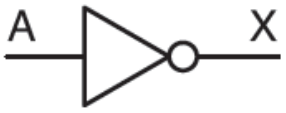




PORTAS

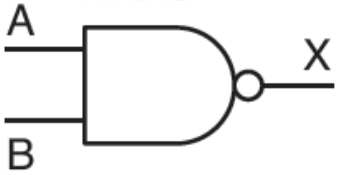
- Símbolos e comportamento funcional das cinco portas básicas

NOT



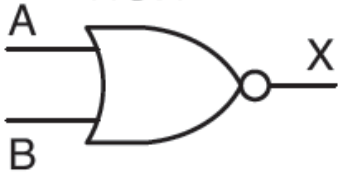
A	X
0	1
1	0

NAND



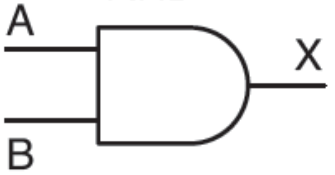
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

NOR



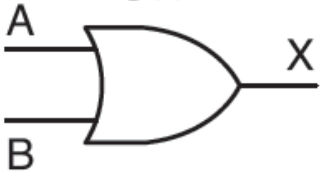
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

AND



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

OR



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

- Variável: A, B, C, ... que podem assumir valores Verdadeiros (1) ou Falsos (0)
- A negação de uma variável é representada por uma barra superior horizontal ou um apóstrofo. É possível ainda utilizar o til ~

  $\overline{A}$ B' ~C

- As operações AND e OR são representadas pelo sinal da multiplicação e da soma, respectivamente

  AB ou A·B → A AND B

  C+D → C OR D

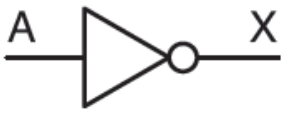
- Ex.:

  F = A + B·C

TABELA VERDADE

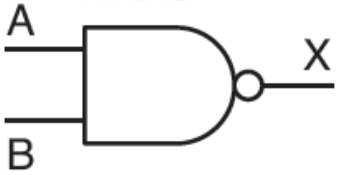
- Descreve as saídas para cada combinação em uma função booleana

NOT



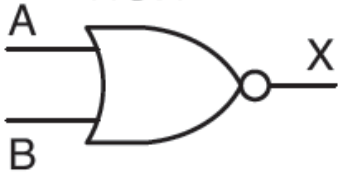
A	X
0	1
1	0

NAND



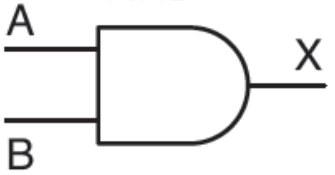
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

NOR



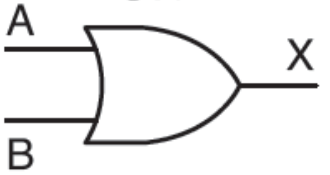
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

AND



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

OR

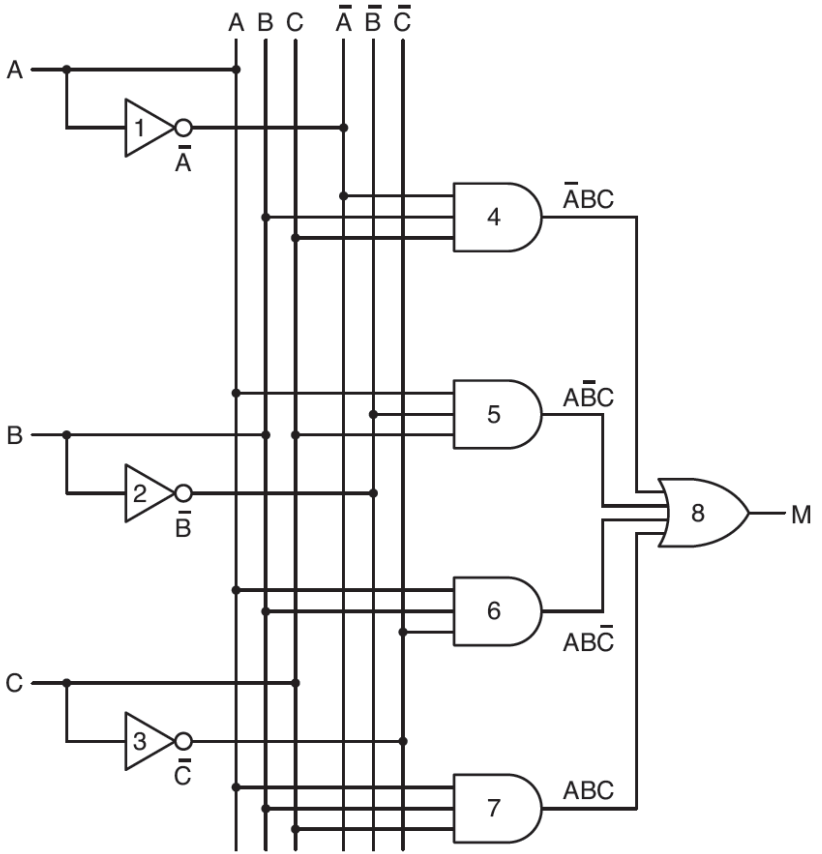


A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

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



DESENHO DE CIRCUITO A PARTIR DE FUNÇÃO

a) $F = A$

b) $F = A+B$

c) $F = AB$

d) $F = \overline{A}$

e) $F = \overline{A}+B$

f) $F = A+\overline{B}$

g) $F = \overline{A}+\overline{B}$

h) $F = \overline{A+B}$

i) $F = \overline{AB}$

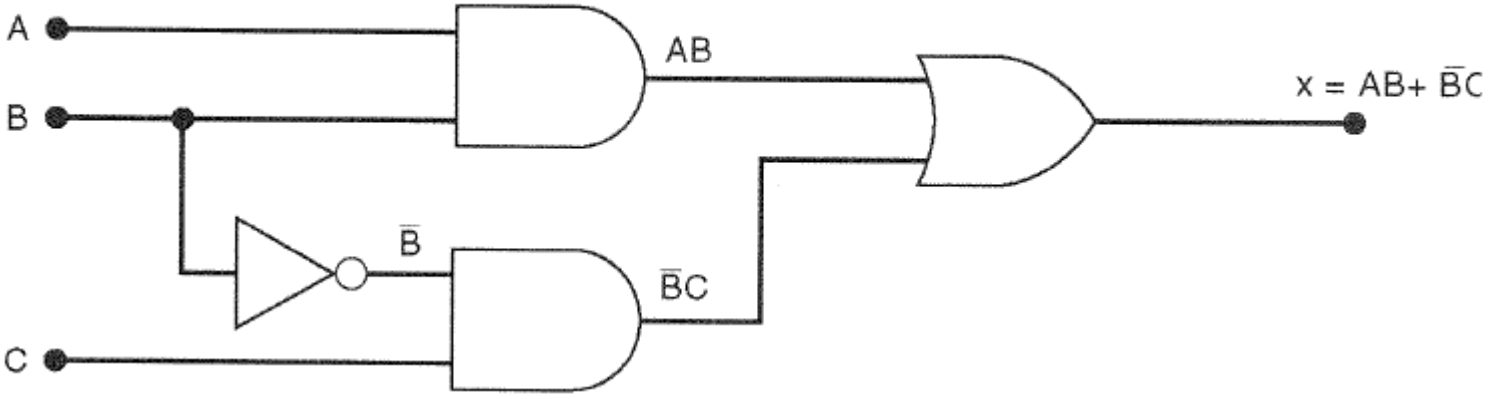
j) $F = A + B \cdot C$

k) $F = A + \overline{B} \cdot C$

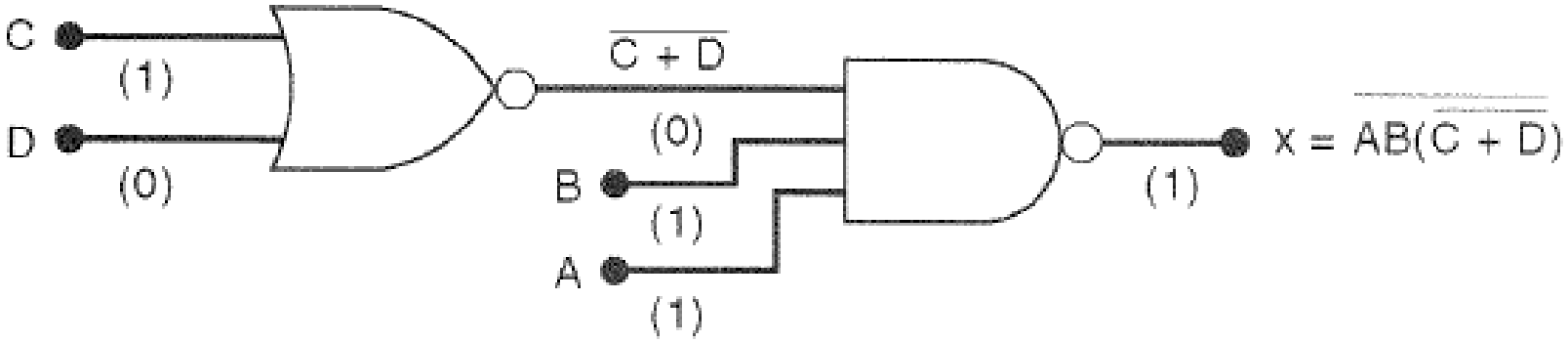
l) $F = \overline{AB+\overline{B}C}$

m) $F = \overline{AB(\overline{C+D})}$

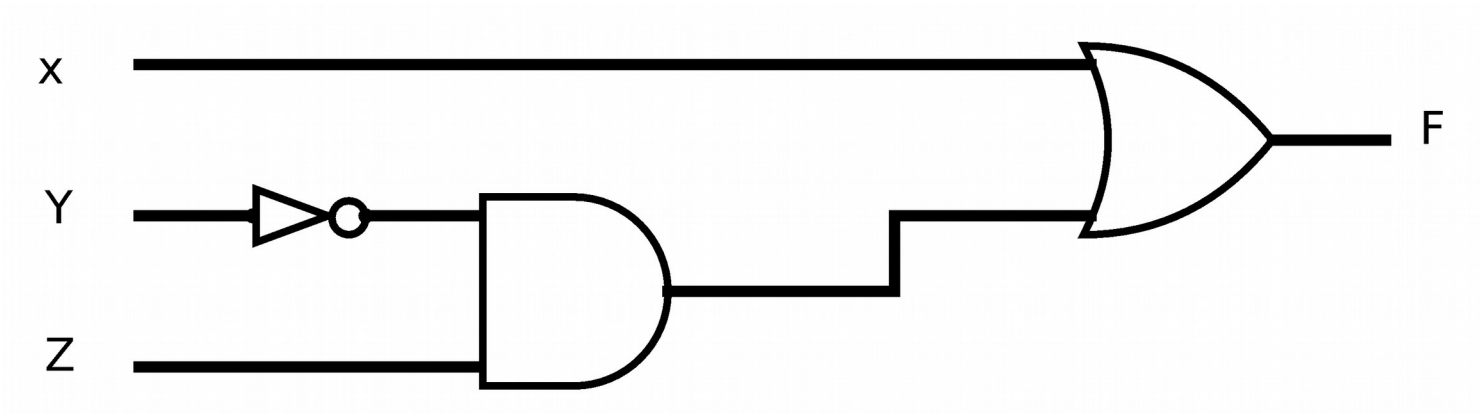
$F = AB+\overline{B}C$



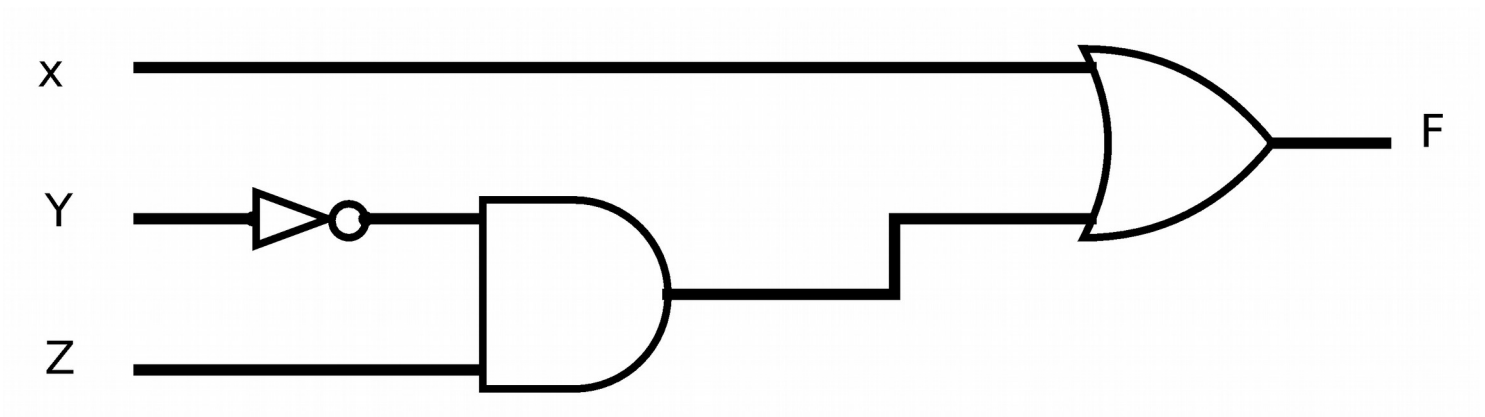
$$F = \overline{AB\overline{(C+D)}}$$



DESENHO DE FUNÇÃO A PARTIR DE CIRCUITO

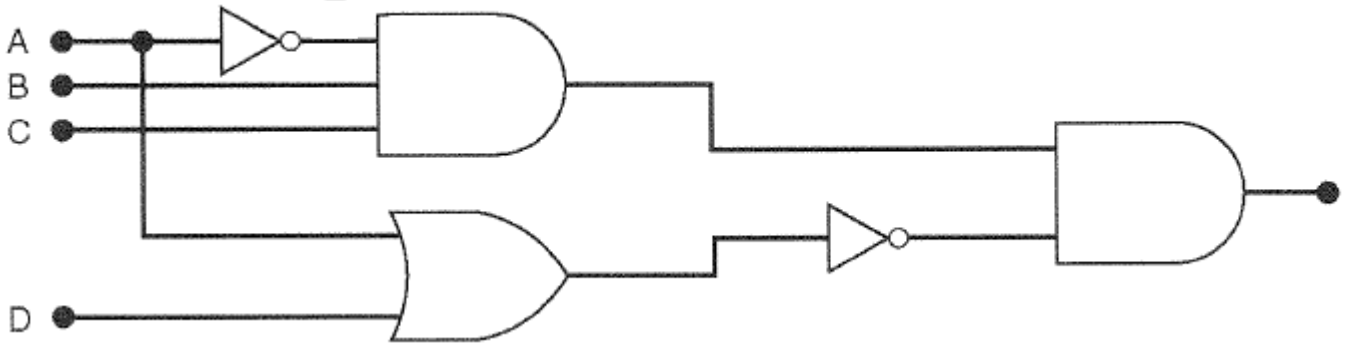


DESENHO DE FUNÇÃO A PARTIR DE CIRCUITO

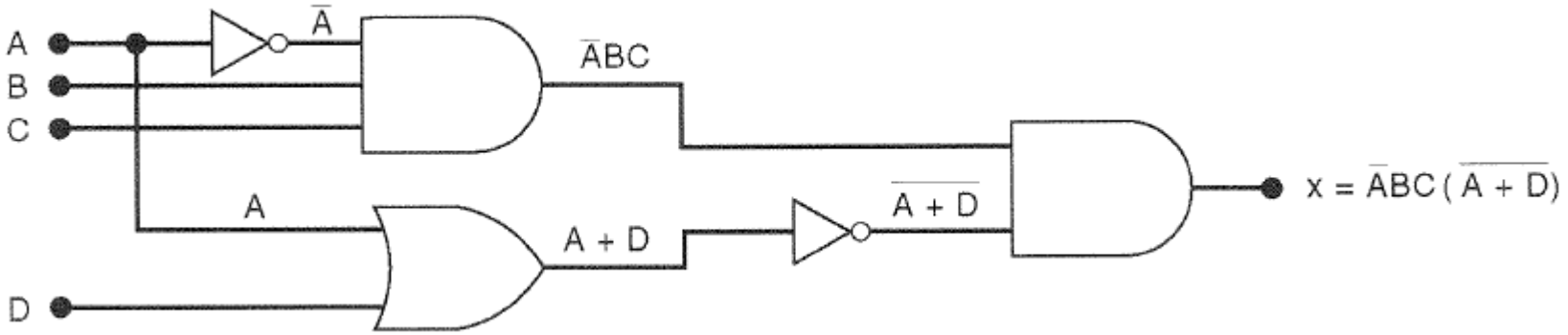


$$F = X + \overline{Y} \cdot Z$$

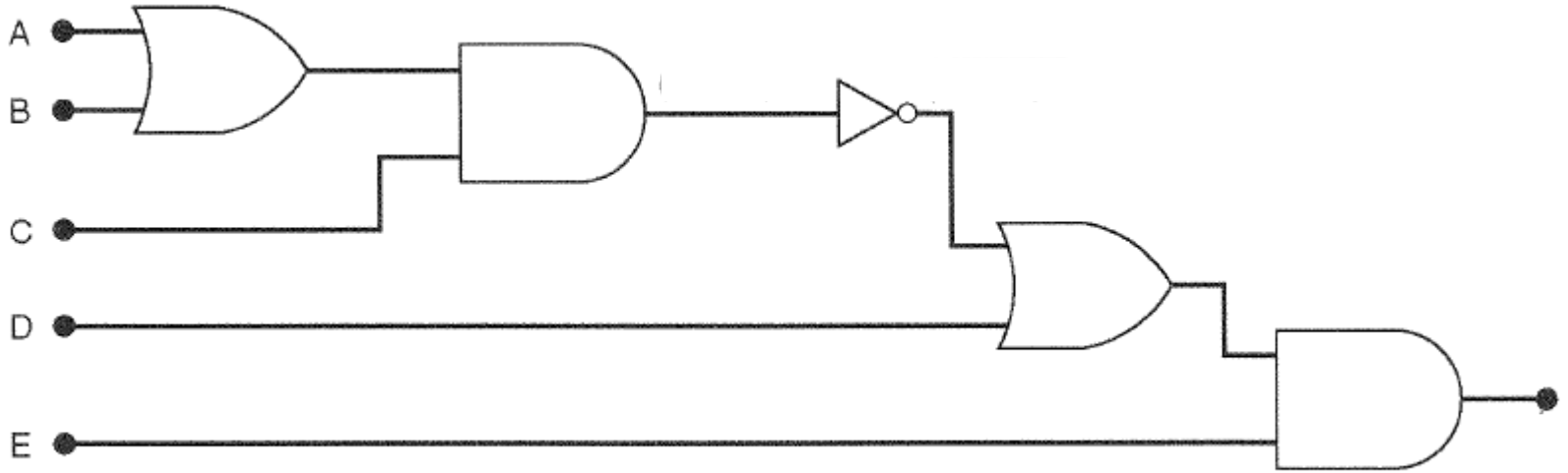
DESENHO DE FUNÇÃO A PARTIR DE CIRCUITO



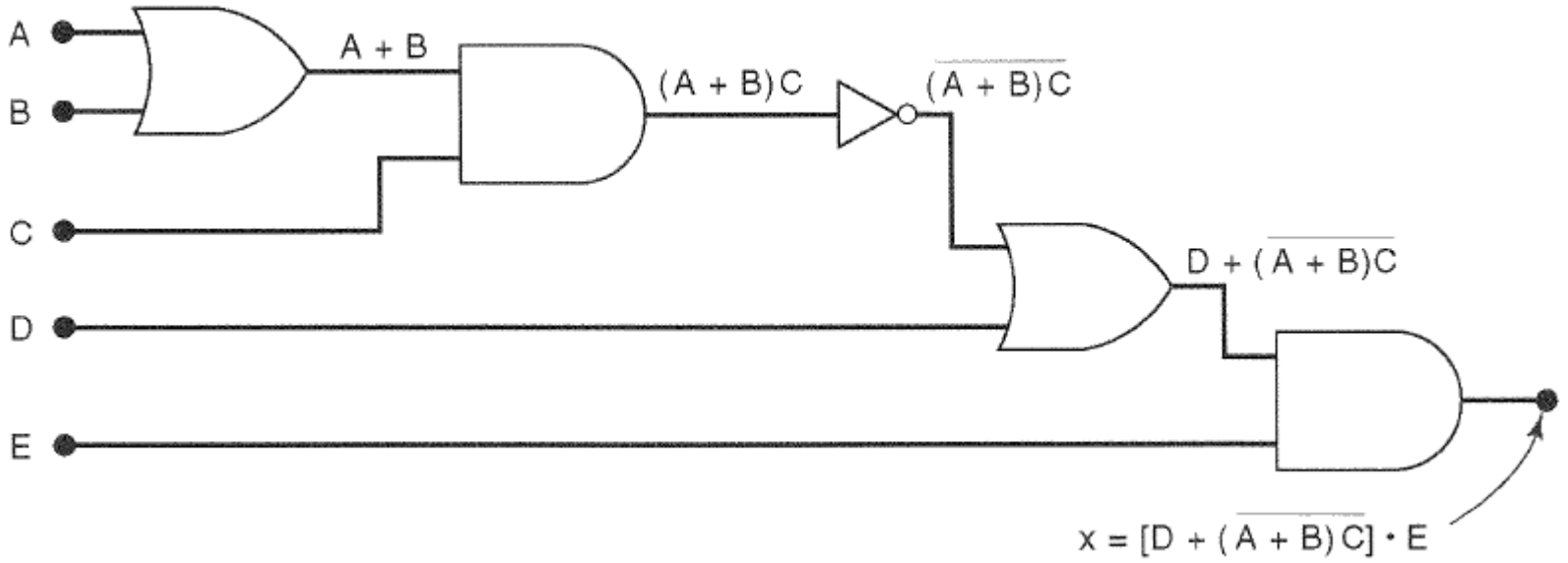
DESENHO DE FUNÇÃO A PARTIR DE CIRCUITO



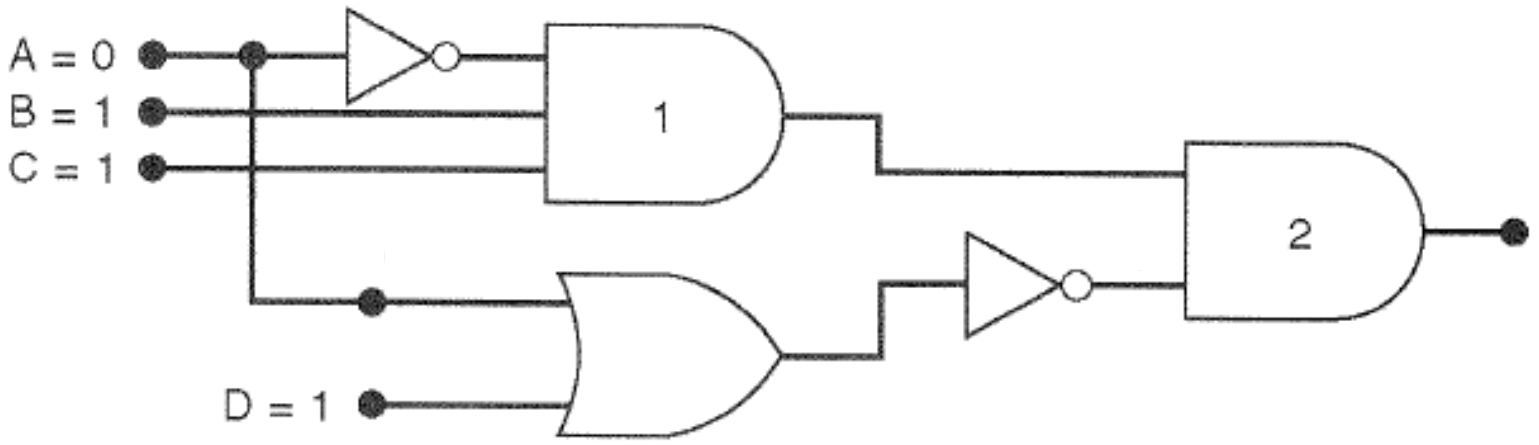
DESENHO DE FUNÇÃO A PARTIR DE CIRCUITO



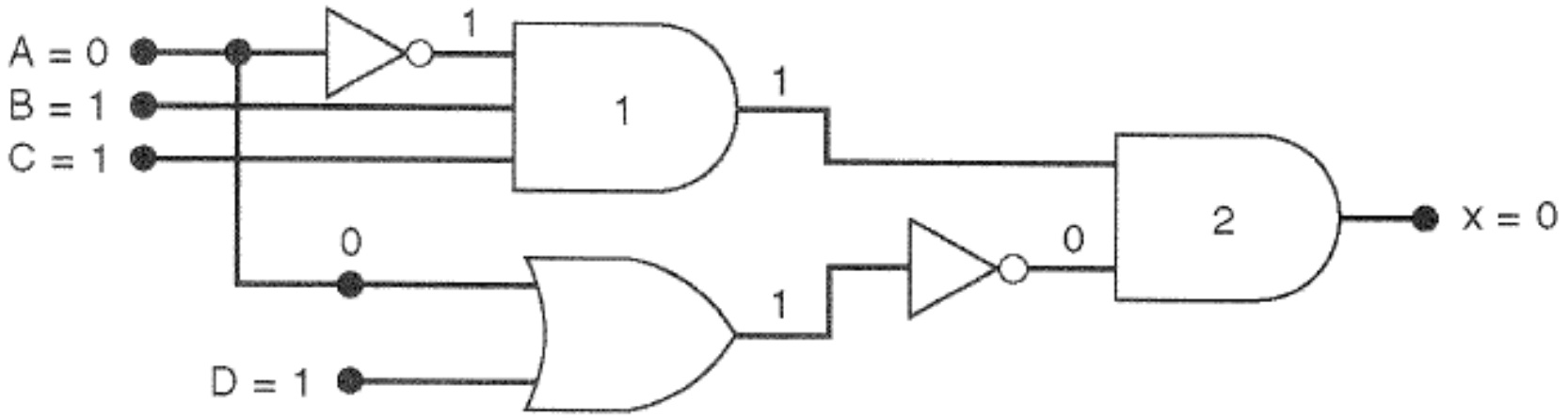
DESENHO DE FUNÇÃO A PARTIR DE CIRCUITO



DETERMINAR O NÍVEL LÓGICO DE SAÍDA A PARTIR DO DIAGRAMA



DETERMINAR O NÍVEL LÓGICO DE SAÍDA A PARTIR DO DIAGRAMA



EXPRESSÃO BOOLEANA A PARTIR DA TABELA VERDADE

X	Y	Z	F
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	1
1	1	1	1

EXPRESSÃO BOOLEANA A PARTIR DA TABELA VERDADE

X	Y	Z	F	
0	0	0	0	
0	0	1	1	$=> \overline{X} . \overline{Y} . Z$
0	1	0	0	
0	1	1	0	
1	0	0	1	$=> X . \overline{Y} . \overline{Z}$
1	0	1	1	$=> X . \overline{Y} . Z$
1	1	0	1	$=> X . Y . \overline{Z}$
1	1	1	1	$=> X . Y . Z$

$+$

EXPRESSÃO BOOLEANA A PARTIR DA TABELA VERDADE

X	Y	Z	F
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	1
1	1	1	1

$F = (\overline{X} \cdot \overline{Y} \cdot Z) + (X \cdot \overline{Y} \cdot \overline{Z}) +$

$(X \cdot \overline{Y} \cdot Z) + (X \cdot Y \cdot \overline{Z}) +$

$(X \cdot Y \cdot Z)$

$\Rightarrow \overline{X} \cdot \overline{Y} \cdot Z$

$\Rightarrow X \cdot \overline{Y} \cdot \overline{Z}$

$\Rightarrow X \cdot \overline{Y} \cdot Z$

$\Rightarrow X \cdot Y \cdot \overline{Z}$

$\Rightarrow X \cdot Y \cdot Z$

$+$

EXPRESSÃO BOOLEANA A PARTIR DA TABELA VERDADE

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

EXPRESSÃO BOOLEANA A PARTIR DA TABELA VERDADE

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

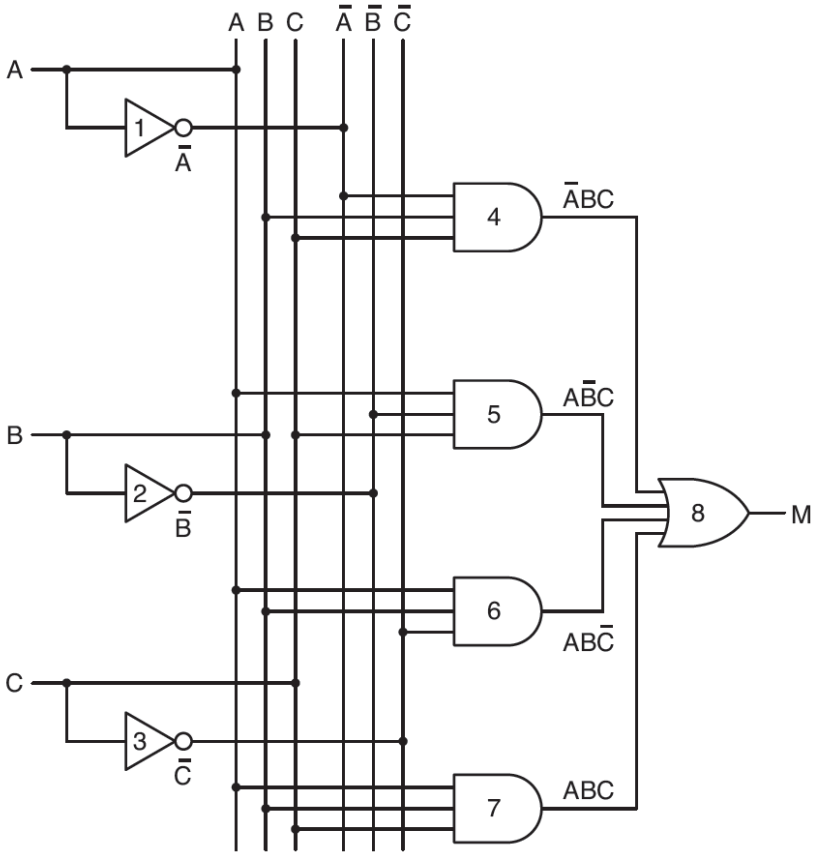
EXPRESSÃO BOOLEANA A PARTIR DA TABELA VERDADE

A	B	C	D	x
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

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

- **Leis Fundamentais e Propriedades da Álgebra Booleana**
  - As leis da álgebra Booleana dizem respeito ao **espaço Booleano** (isto é, valores que uma variável pode assumir) e às **operações elementares desse espaço**
  - As propriedades podem ser deduzidas a partir das definições das operações
  - Sejam A, B e C três variáveis Booleanas. Então, o espaço Booleano é definido:
    - se A ≠ 0, então A = 1
    - se A ≠ 1, então A = 0
  - O mesmo se aplica para B, C ou outra variável do mesmo espaço
  - Operações básicas desse espaço:
    - AND (e lógico)
    - OR (ou lógico)
    - NOT (negação / inversão lógica)

PORTAS E ÁLGEBRA BOOLEANA

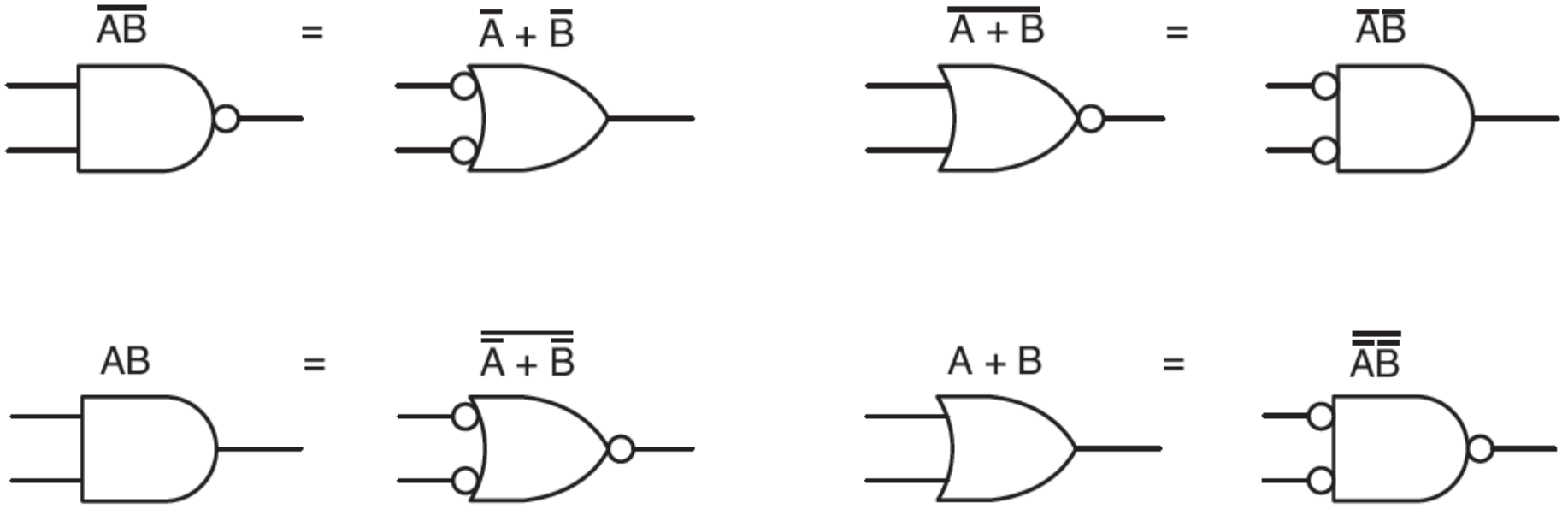
- Algumas identidades da álgebra booleana

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	$A\overline{A} = 0$	$A + \overline{A} = 1$
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	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}\,\overline{B}$

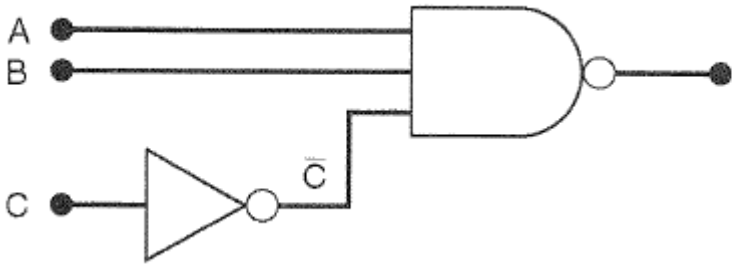
Identidade Auxiliar $A + \overline{A}.B = A + B$

PORTAS E ÁLGEBRA BOOLEANA

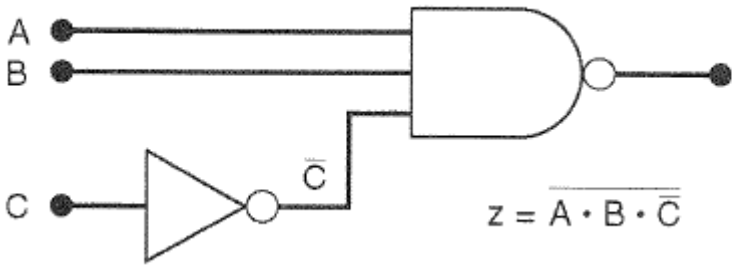
- Símbolos alternativos para algumas portas: NAND. NOR. AND. OR



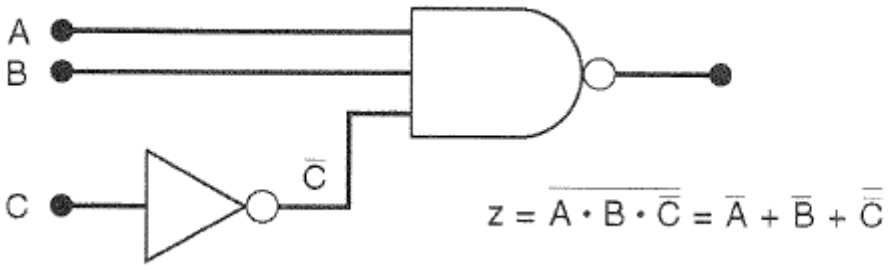
SIMPLIFICAÇÃO ALGÉBRICA – EXEMPLO 1



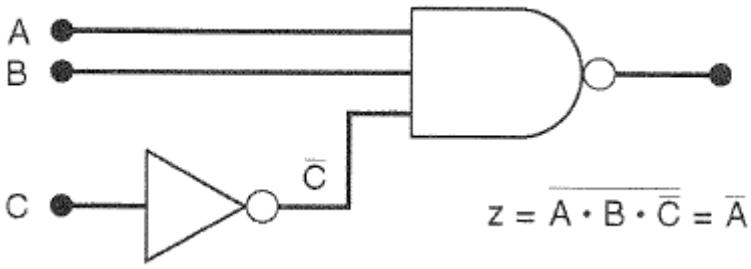
SIMPLIFICAÇÃO ALGÉBRICA – EXEMPLO 1



SIMPLIFICAÇÃO ALGÉBRICA – EXEMPLO 1

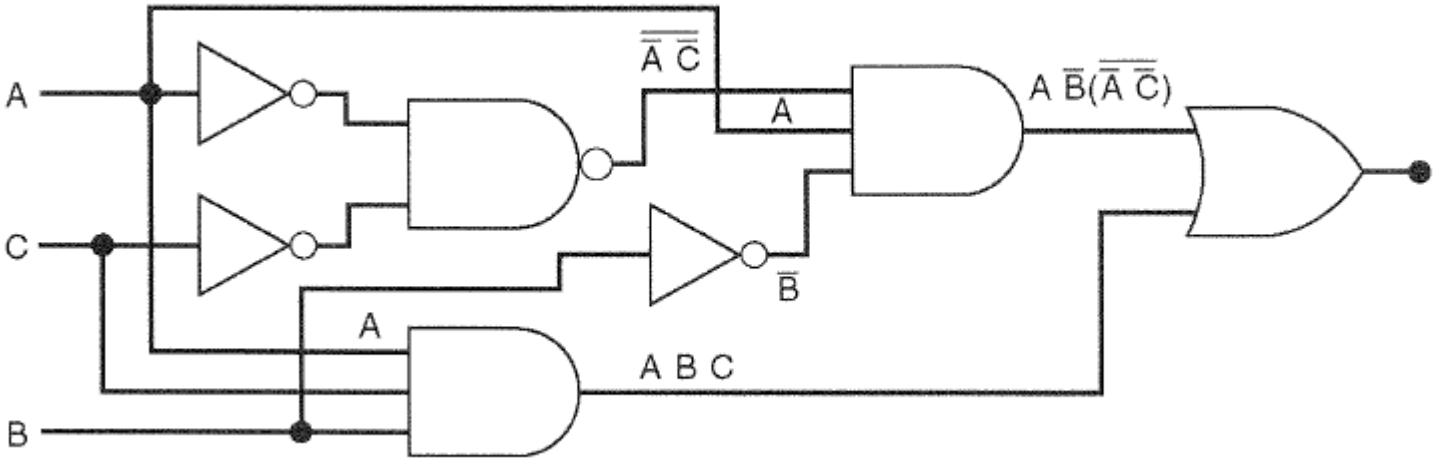


SIMPLIFICAÇÃO ALGÉBRICA – EXEMPLO 1

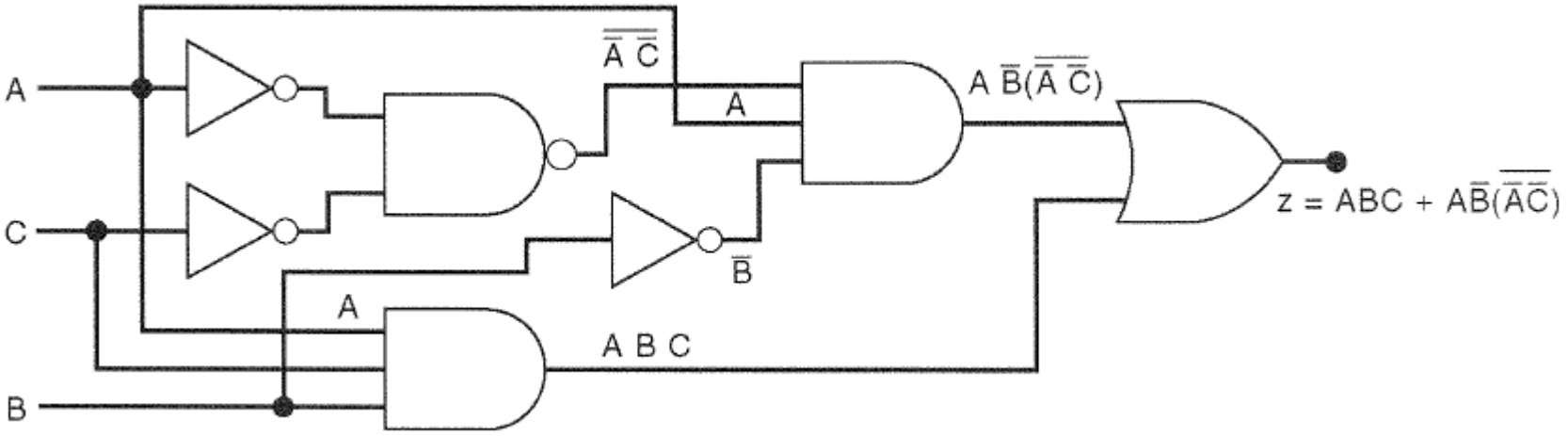


$$z = \overline{A \cdot B \cdot \overline{C}} = \overline{A} + \overline{B} + \overline{\overline{C}} = \overline{A} + \overline{B} + C$$

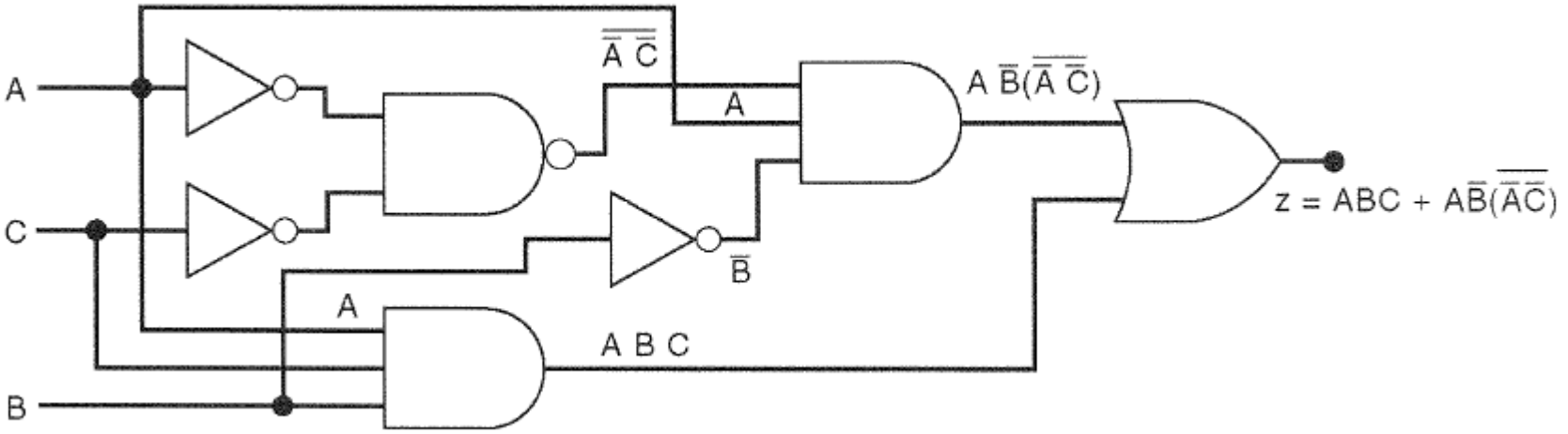
SIMPLIFICAÇÃO ALGÉBRICA – EXEMPLO 2

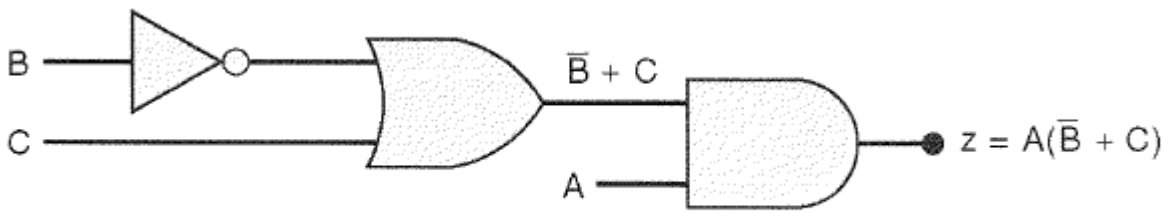


SIMPLIFICAÇÃO ALGÉBRICA – EXEMPLO 2



SIMPLIFICAÇÃO ALGÉBRICA – EXEMPLO 2





SIMPLIFICAÇÃO ALGÉBRICA

$$(\bar{A} + B)(A + B) =$$

SIMPLIFICAÇÃO ALGÉBRICA

$$(\bar{A} + B)(A + B) = B$$

SIMPLIFICAÇÃO ALGÉBRICA

$$(\bar{A} + B)(A + B) = B$$

ATIVIDADES

...

ATIVIDADE 01

- Circuitos analógicos estão sujeitos a ruído que pode distorcer sua saída. Os circuitos digitais são imunes ao ruído?

ATIVIDADE 02

- Um especialista em lógica entra em uma lanchonete *drive-in* e diz: “Quero um hambúrguer ou um cachorro-quente e batatas fritas”. Infelizmente, o cozinheiro não sabe (ou não se importa) se “e” tem precedência sobre “ou”. Para ele, tanto faz uma ou outra interpretação. Dessa forma, quais dos seguintes casos são interpretações válidas do pedido, do ponto de vista do cozinheiro?

  a) Apenas um hambúrguer
  b) Apenas um cachorro-quente
  c) Apenas batatas fritas
  d) Um cachorro-quente e batatas fritas
  e) Um hambúrguer e batatas fritas
  f) Um cachorro-quente e um hambúrguer
  g) Todos os três
  h) Nada – o especialista em lógica passa fome por ser muito gaiato ;)

SOLUÇÃO

Quero um **hambúrguer** ou um **cachorro-quente** e **batatas fritas**” – pode ser avaliado, pelo cozinheiro, como

hambúrguer ou (cachorro-quente e batatas fritas) ou
(hambúrguer ou cachorro-quente) e batatas fritas

a) Apenas um hambúrguer
b) Apenas um cachorro-quente
c) Apenas batatas fritas
d) Um cachorro-quente e batatas fritas
e) Um hambúrguer e batatas fritas
f) Um cachorro-quente e um hambúrguer
g) Todos os três
h) Nada – o especialista em lógica passa fome por ser muito gaiato ;)

SOLUÇÃO

Quero um **hambúrguer** ou um **cachorro-quente** e **batatas fritas**” – pode ser avaliado, pelo cozinheiro, como

hambúrguer ou (cachorro-quente e batatas fritas) ou
(hambúrguer ou cachorro-quente) e batatas fritas

a) Apenas um hambúrguer
b) Apenas um cachorro-quente
c) Apenas batatas fritas
d) Um cachorro-quente e batatas fritas
e) Um hambúrguer e batatas fritas
f) Um cachorro-quente e um hambúrguer
g) Todos os três
h) Nada – o especialista em lógica passa fome por ser muito gaiato ;)

SOLUÇÃO

Quero um **hambúrguer** ou um **cachorro-quente** e **batatas fritas**” – pode ser avaliado, pelo cozinheiro, como

hambúrguer ou (cachorro-quente e batatas fritas) ou

(hambúrguer ou cachorro-quente) e batatas fritas

a) Apenas um hambúrguer
b) Apenas um cachorro-quente
c) Apenas batatas fritas
d) Um cachorro-quente e batatas fritas
e) Um hambúrguer e batatas fritas
f) Um cachorro-quente e um hambúrguer
g) Todos os três
h) Nada – o especialista em lógica passa fome por ser muito gaiato ;)

SOLUÇÃO

Quero um **hambúrguer** ou um **cachorro-quente** e **batatas fritas**” – pode ser avaliado, pelo cozinheiro, como

hambúrguer ou (cachorro-quente e batatas fritas) ou
(hambúrguer ou cachorro-quente) e batatas fritas

a) Apenas um hambúrguer
b) Apenas um cachorro-quente
c) Apenas batatas fritas
d) Um cachorro-quente e batatas fritas
e) Um hambúrguer e batatas fritas
f) Um cachorro-quente e um hambúrguer
g) Todos os três
h) Nada – o especialista em lógica passa fome por ser muito gaiato ;)

ATIVIDADE 03

- Utilize a tabela verdade para mostrar que X = (X AND Y) OR (X AND NOT Y)

ATIVIDADE 04

- Mostre como uma porta **AND** pode ser construída com base em duas portas **NAND**

ATIVIDADE 05

- Mostre como uma porta **OR** pode ser construída com base em duas portas **NOR**

ATIVIDADE 06

- Construa uma porta **AND** usando apenas portas **NOR**

ATIVIDADE 07

- Construa uma porta **OR** usando apenas portas **NAND**

CIRCUITOS LÓGICOS

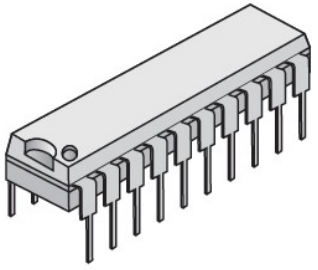
Circuitos integrados

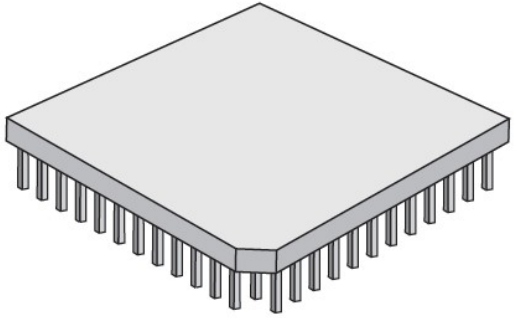
CIRCUITOS INTEGRADOS

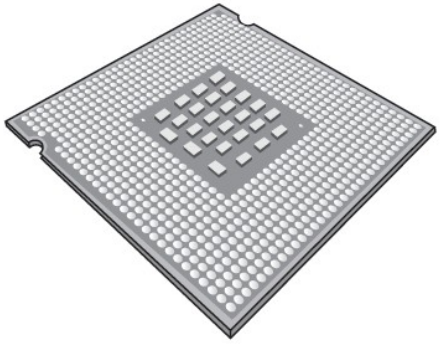
- Portas não são fabricadas nem vendidas individualmente, mas em unidades denominadas circuitos integrados, muitas vezes denominados ICs, CIs ou chips

CIRCUITOS INTEGRADOS

- Portas não são fabricadas nem vendidas individualmente, mas em unidades denominadas circuitos integrados, muitas vezes denominados ICs, CIs ou chips
- Um CI é um pedaço quadrado de silício de tamanho variado, dependendo de quantas portas são necessárias para executar os componentes do chip
  - Substratos de 2x2 mm a 18x18 mm







CIRCUITOS INTEGRADOS

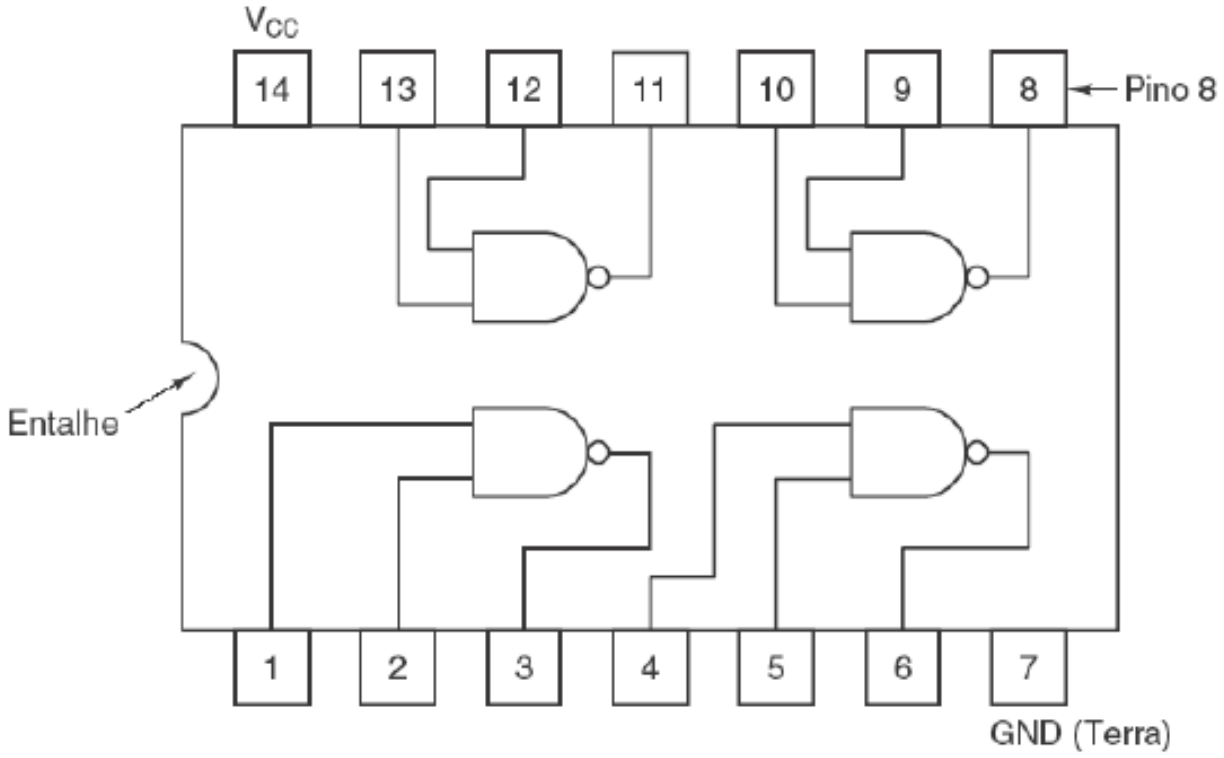
- Portas não são fabricadas nem vendidas individualmente, mas em unidades denominadas circuitos integrados, muitas vezes denominados ICs, CIs ou chips
- Um CI é um pedaço quadrado de silício de tamanho variado, dependendo de quantas portas são necessárias para executar os componentes do chip
- ICs costumam ser montados em pacotes retangulares (ou quadrados) de plástico ou cerâmica.

CIRCUITOS INTEGRADOS

- Níveis de integração
  - SSI (Integração em pequena escala):
    - Constituída de 1 a 10 portas;
  - MSI (Integração em média escala):
    - Constituída de até 100 portas;
  - LSI (Integração em larga escala):
    - Constituída de até 100.000 portas;
  - VLSI (Integração em larguíssima escala):
    - Chips com mais de 100.000 portas.

- A tecnologia moderna permite colocar mais de 1 bilhão de transistores em um único chip

CIRCUITOS INTEGRADOS



- Operação NAND
- Operação AND
- Operação NOT
- Operação OR
- Operação NOR

Chip composto por quatro portas NAND

CIRCUITOS INTEGRADOS

- Podem ser basicamente de dois tipos

- Circuitos combinacionais (combinatórios)
  - São circuitos em que a saída está diretamente relacionada aos valores de entrada, não possuindo elementos de memória em seu interior

- Circuitos sequenciais (de memória)
  - São aqueles em que o valor de saída não depende somente dos valores de entrada, mas também do “estado interno” do circuito – dependem também da memória do circuito

CIRCUITOS LÓGICOS

Circuitos combinacionais

CIRCUITOS COMBINACIONAIS

- 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 combinacionais (ou circuitos combinatórios)**

- Exemplos de circuitos combinatórios
  - Somador parcial
  - Somador completo
  - Decodificador
  - Multiplexador
  - Comparador

CIRCUITOS COMBINACIONAIS

- 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

- A entrada selecionada é dirigida para a saída

- Conversor de dados de paralelo para serial

- Exemplo: teclado – o toque em uma tecla define implicitamente um número (geralmente) de 7 bits que deve ser enviado serialmente para a CPU

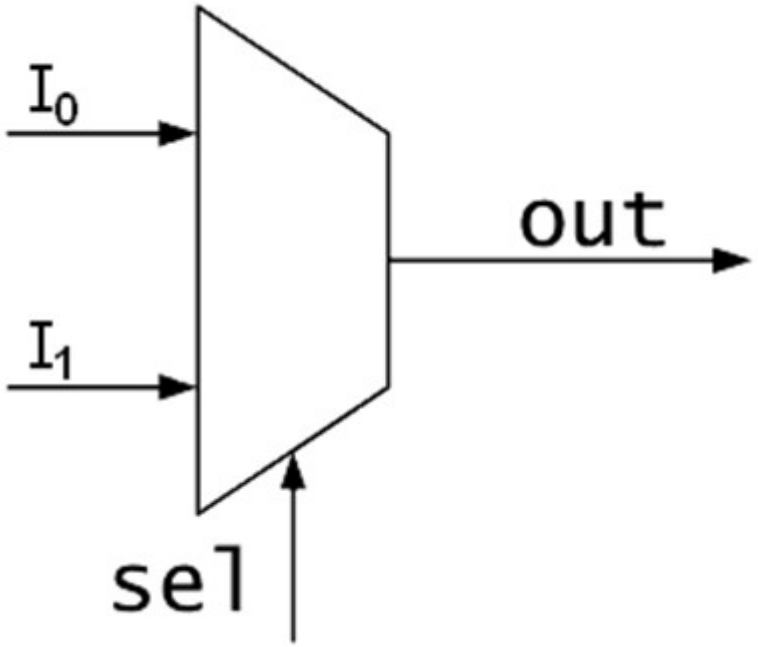
CIRCUITOS COMBINACIONAIS

- Configurações de multiplexadores

Multiplexador	Número de entradas	Número de linhas de seleção
2-para-1	2	1
4-para-1	4	2
8-para-1	8	3
16-para-1	16	4

CIRCUITOS COMBINACIONAIS

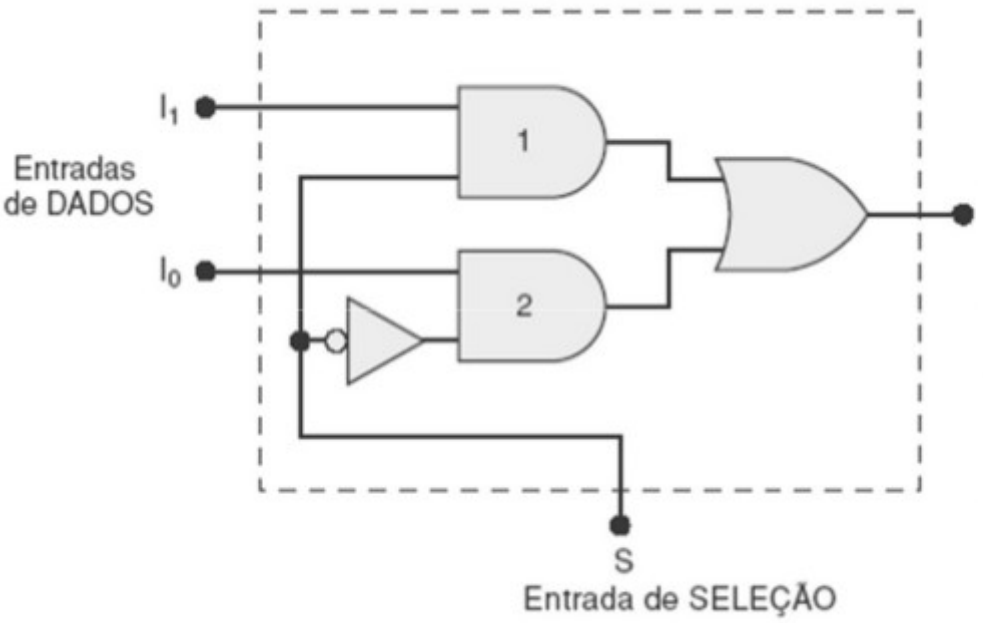
- Multiplexador de duas entradas



CIRCUITOS COMBINACIONAIS

- Multiplexador de duas entradas

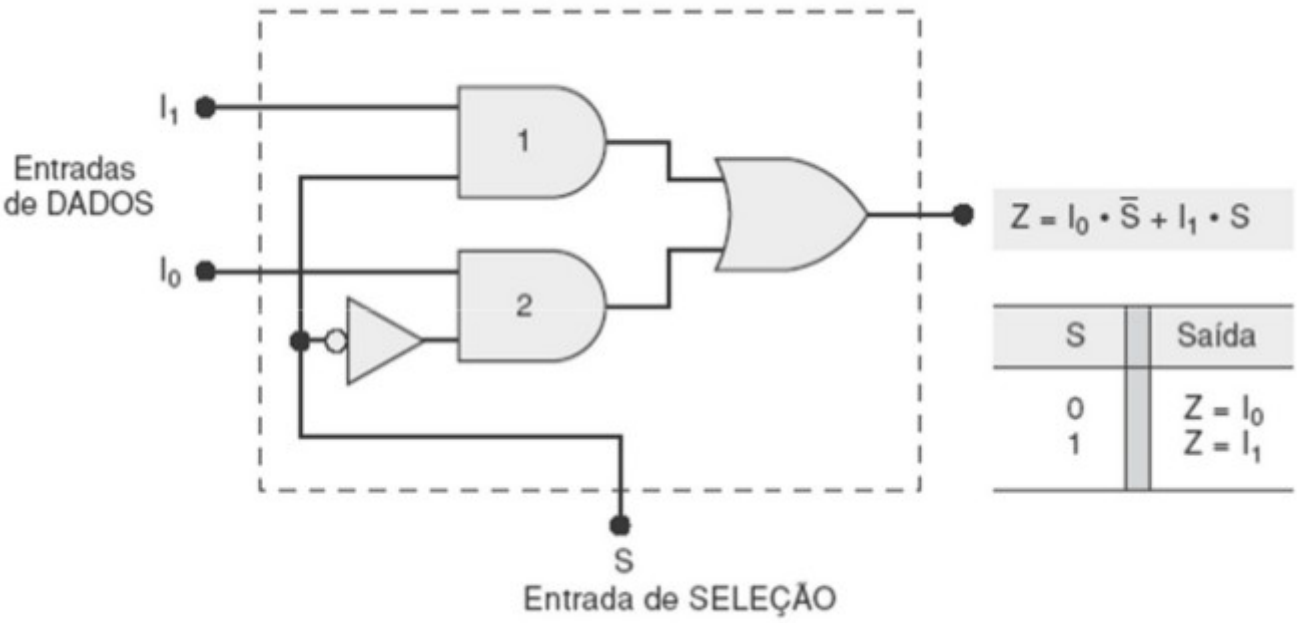
I_0	I_1	S	Z
0	0		
0	0		
0	1		
0	1		
1	0		
1	0		
1	1		
1	1		



CIRCUITOS COMBINACIONAIS

- Multiplexador de duas entradas

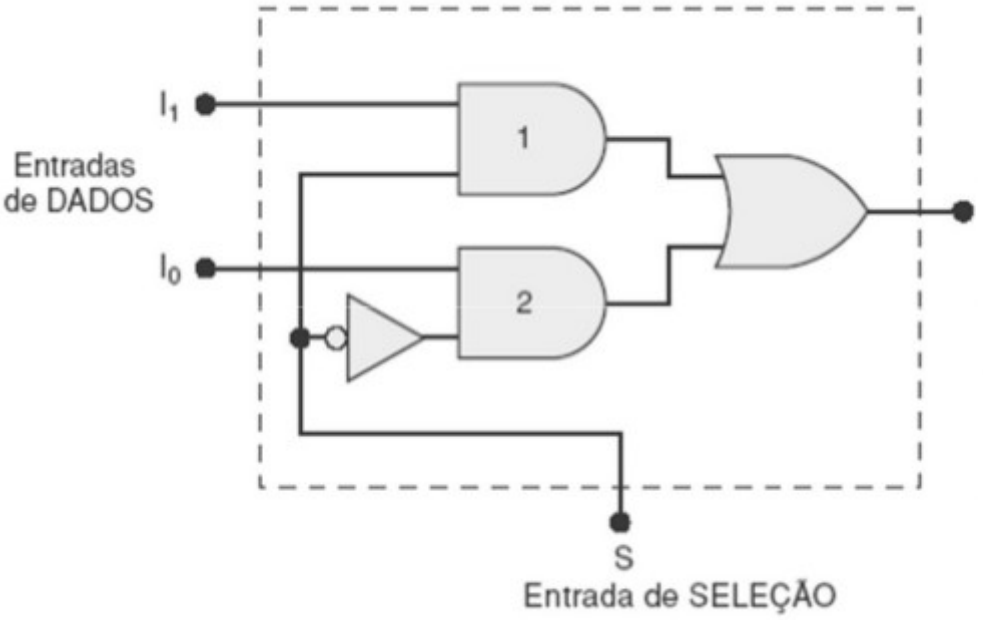
I_0	I_1	S	Z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



CIRCUITOS COMBINACIONAIS

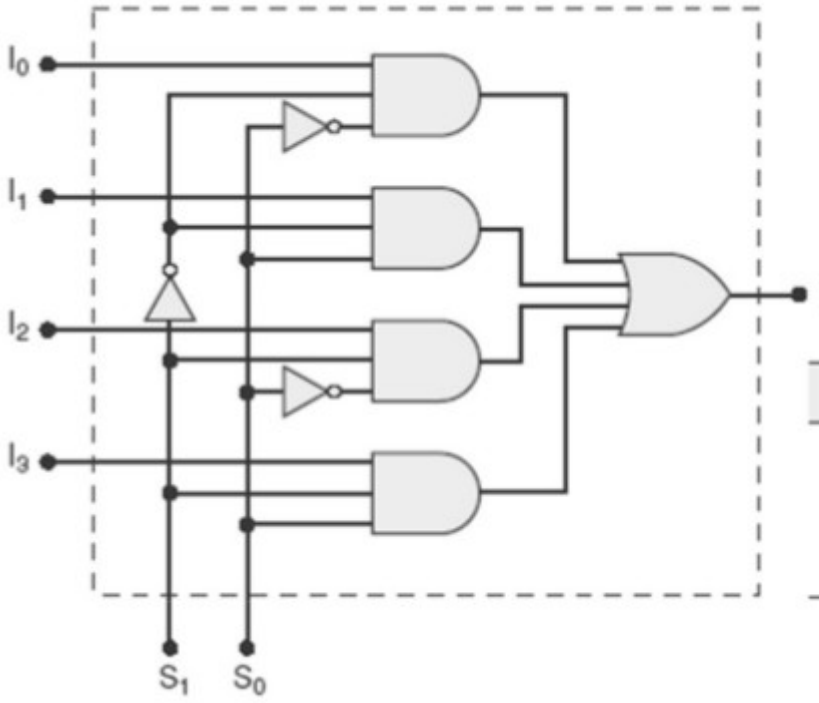
- Multiplexador de duas entradas

I_0	I_1	S	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



CIRCUITOS COMBINACIONAIS

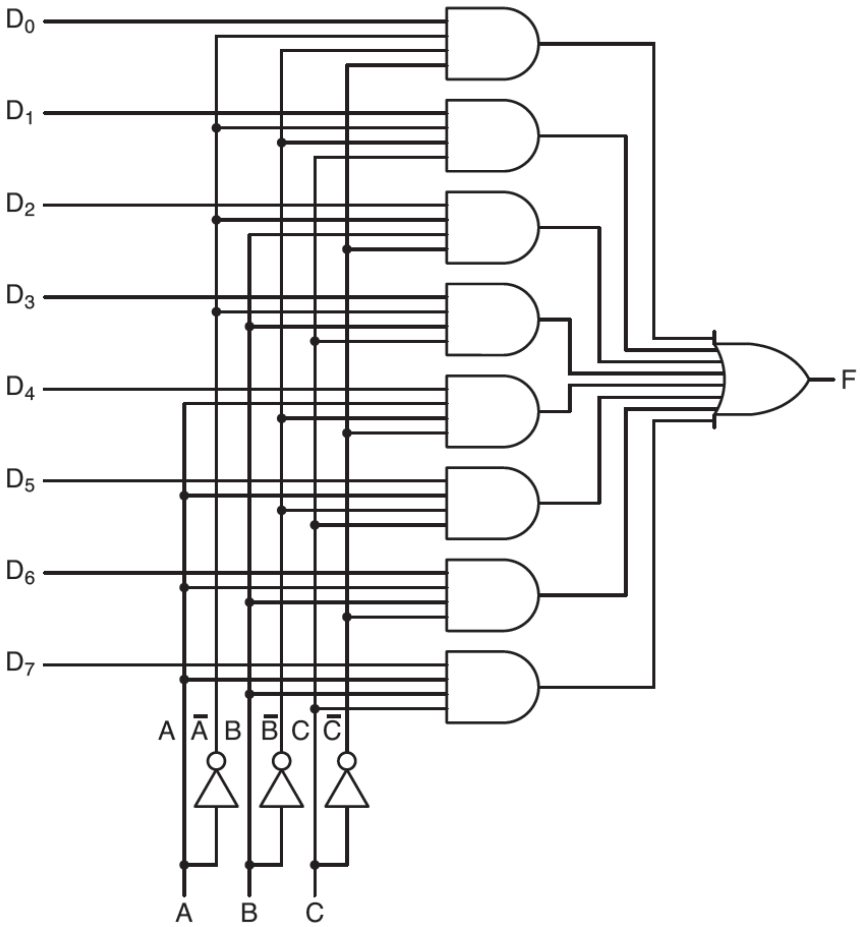
- Multiplexador de quatro entradas



S_1	S_0	Saída
0	0	$Z = I_0$
0	1	$Z = I_1$
1	0	$Z = I_2$
1	1	$Z = I_3$

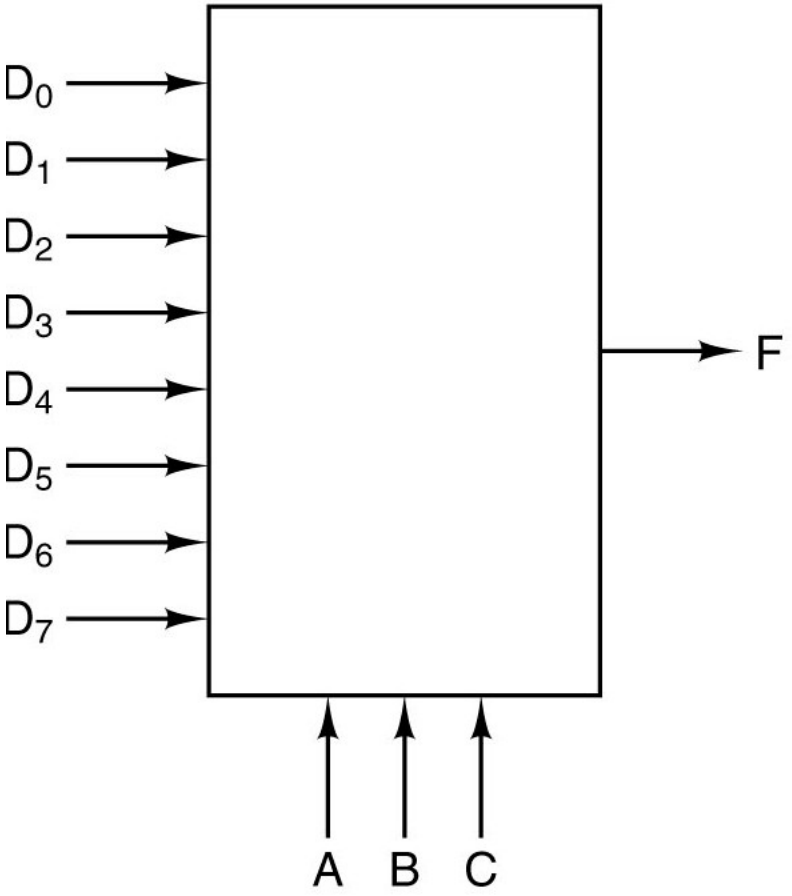
CIRCUITOS COMBINACIONAIS

- Multiplexador de oito entradas



CIRCUITOS COMBINACIONAIS

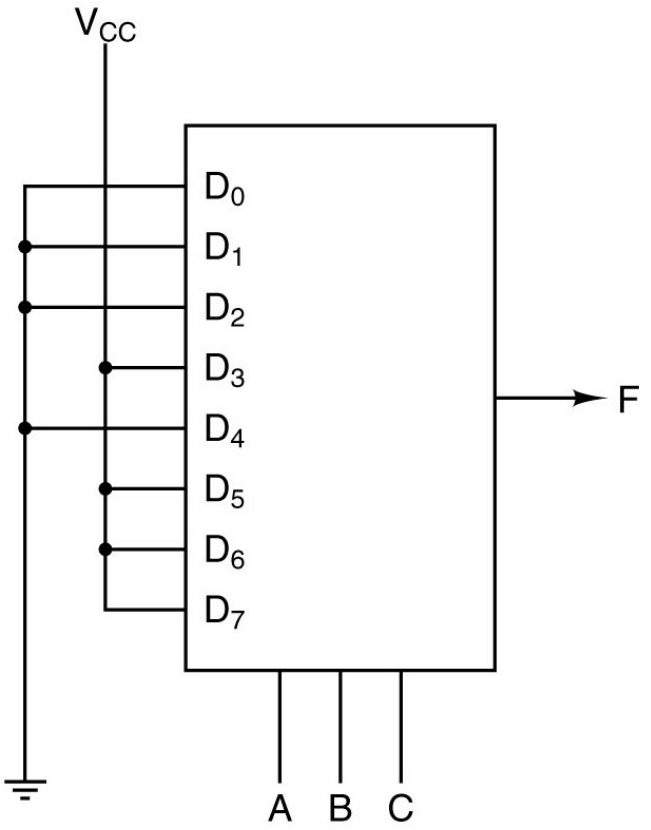
- Multiplexador de oito entradas



CIRCUITOS COMBINACIONAIS

- Multiplexador para a função majoritária de três variáveis

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

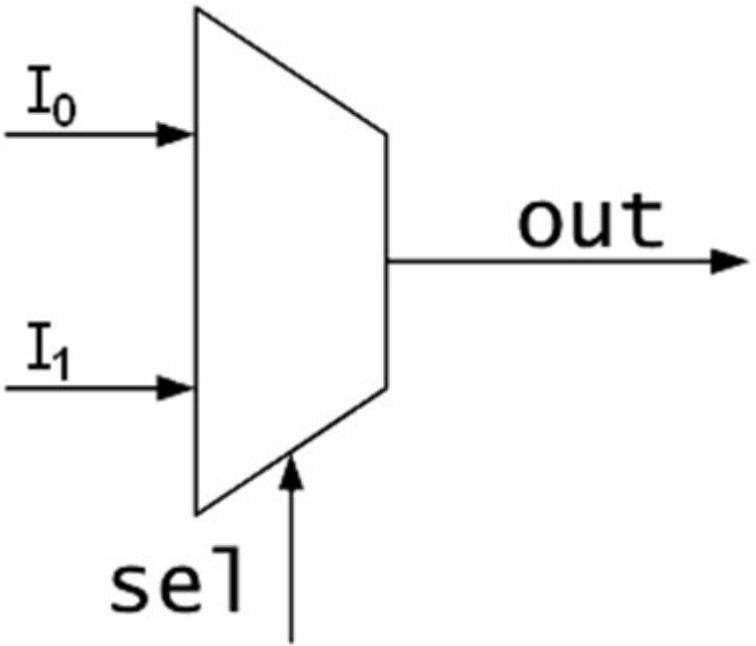


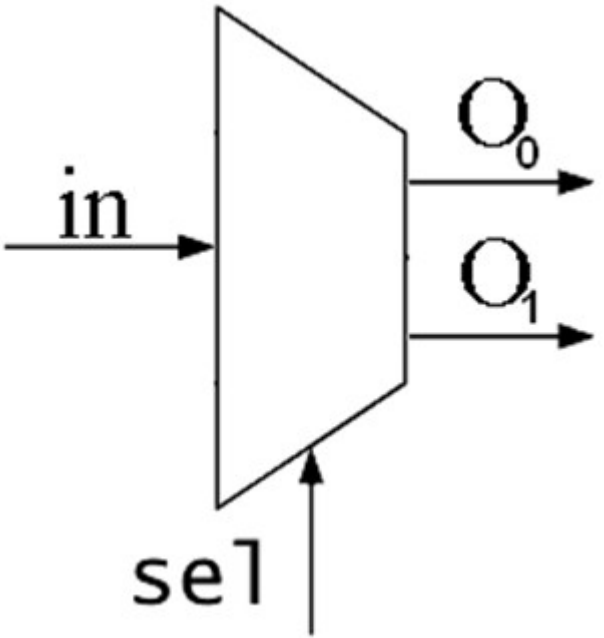
CIRCUITOS COMBINACIONAIS

- O Demultiplexador realiza a operação inversa – dirige uma entrada para uma das 2^n saídas

CIRCUITOS COMBINACIONAIS

- O Demultiplexador realiza a operação inversa – dirige uma entrada para uma das 2^n saídas

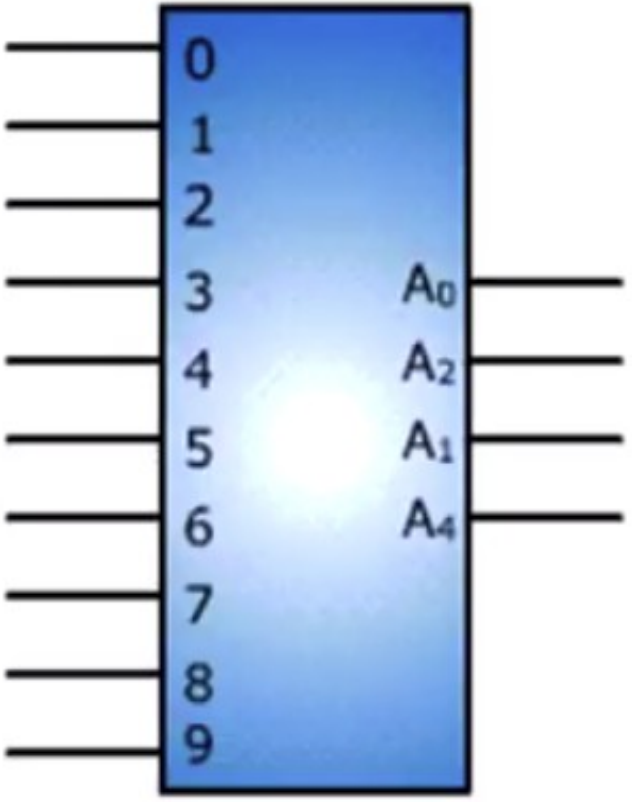




CIRCUITOS COMBINACIONAIS

- **Codificador**
  - Possuem várias linhas de entrada
  - Saída: tantas linhas quantas forem necessárias

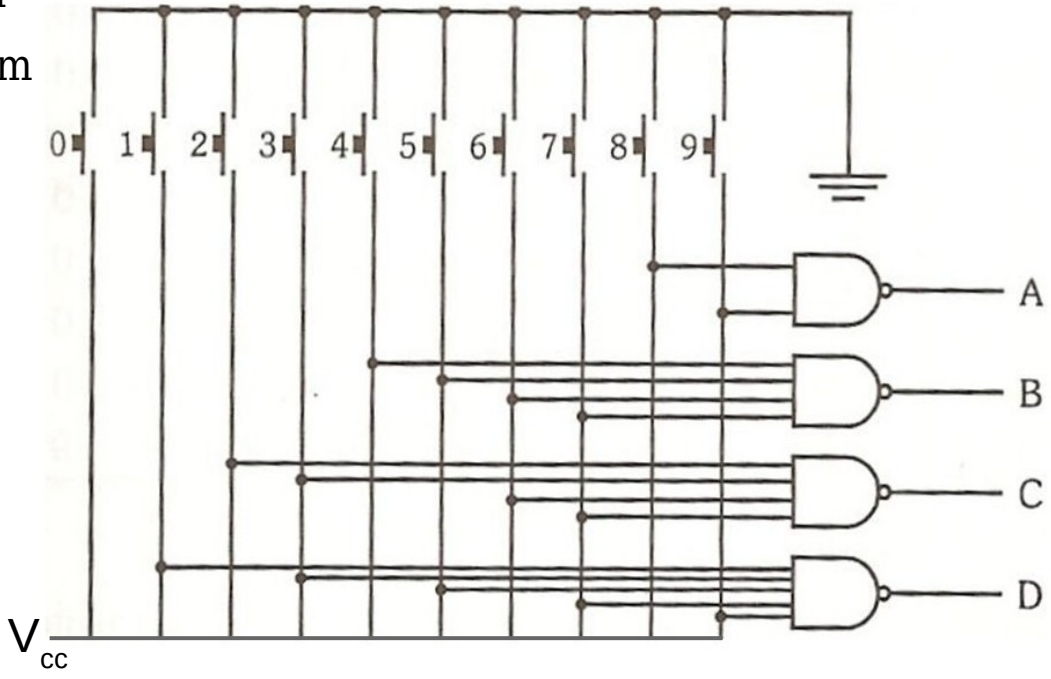
- Ex.: Codificar decimal em binário



CIRCUITOS COMBINACIONAIS

- **Codificador**
  - Possuem várias linhas de entrada
  - Saída: tantas linhas quantas forem necessárias

- Ex.: Codificar decimal em binário

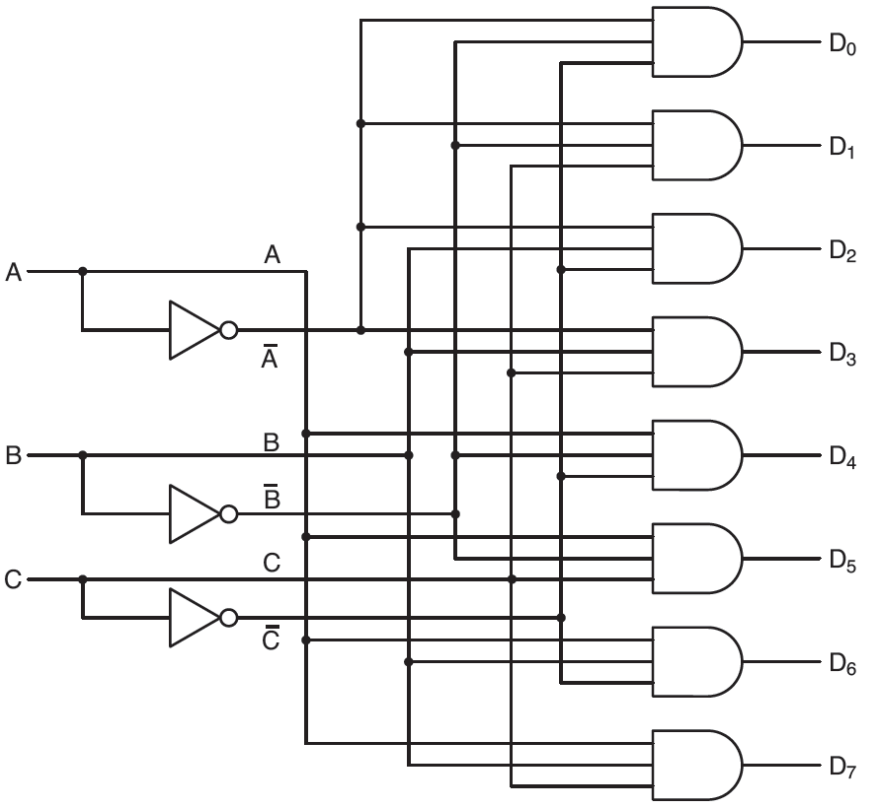


CIRCUITOS COMBINACIONAIS

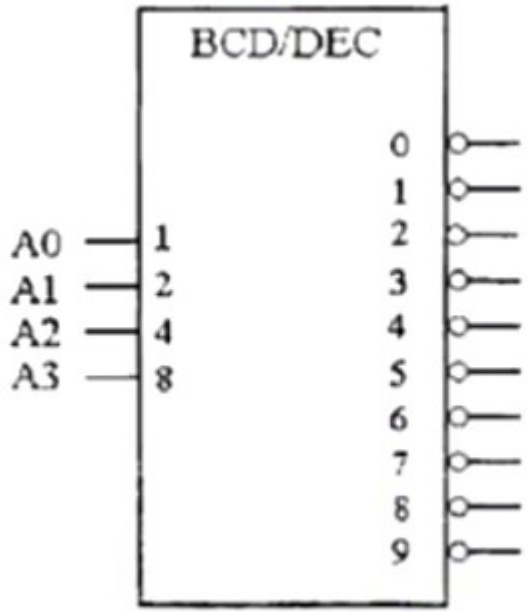
- **Decodificador** – circuito que toma um número de *n* bits como entrada e o usa para selecionar (isto é, definir em 1) exatamente uma das 2^n linhas de saída

- Cada porta **AND** tem três entradas, das quais a primeira é *A* ou *A'*, a segunda é *B* ou *B'* e a terceira é *C* ou *C'*

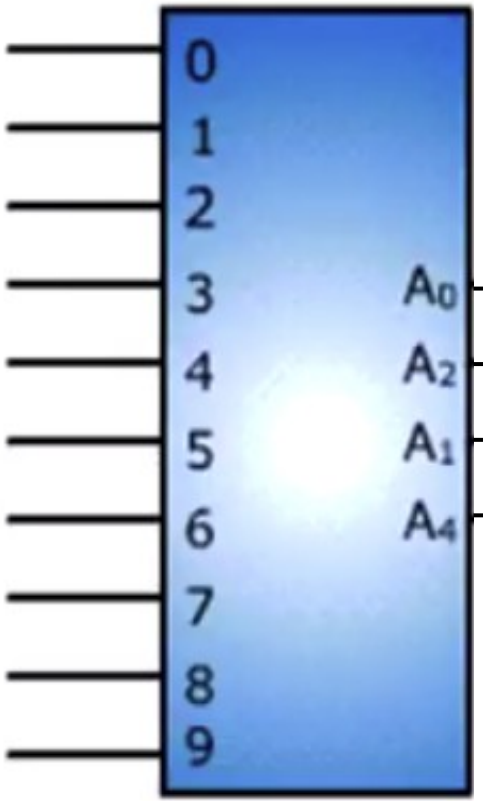
- Cada porta é habilitada por uma combinação (única) diferente de entradas
  - D_0: $A'\ B'\ C'$
  - D_1: $A'\ B'\ C$
  - etc



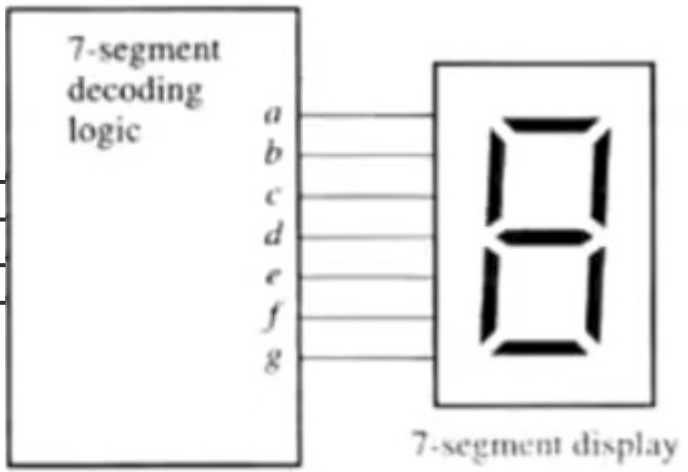
CIRCUITOS COMBINACIONAIS



CIRCUITOS COMBINACIONAIS

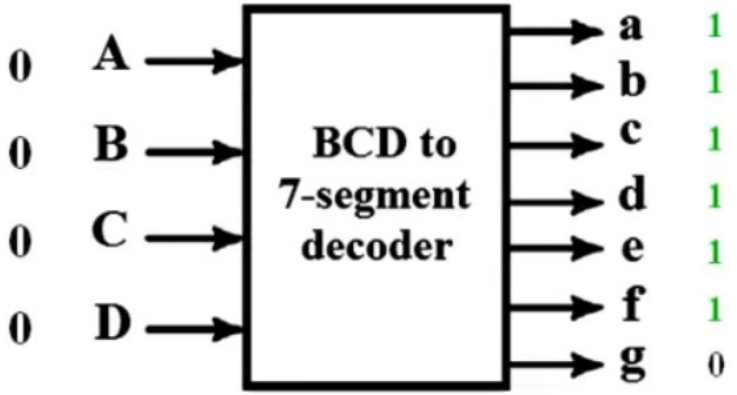


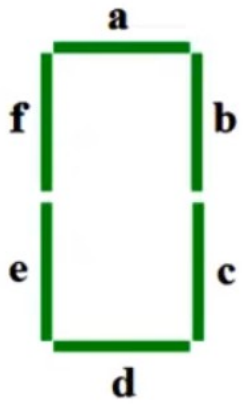
Codificar decimal em binário



Decodificador 7-segmentos

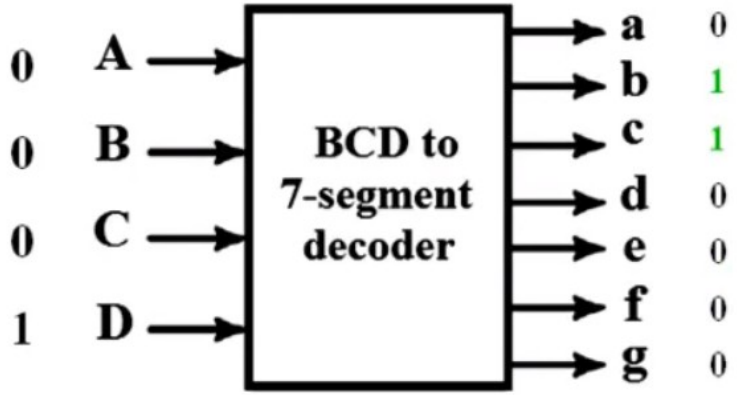
BCD to 7-segment decoder





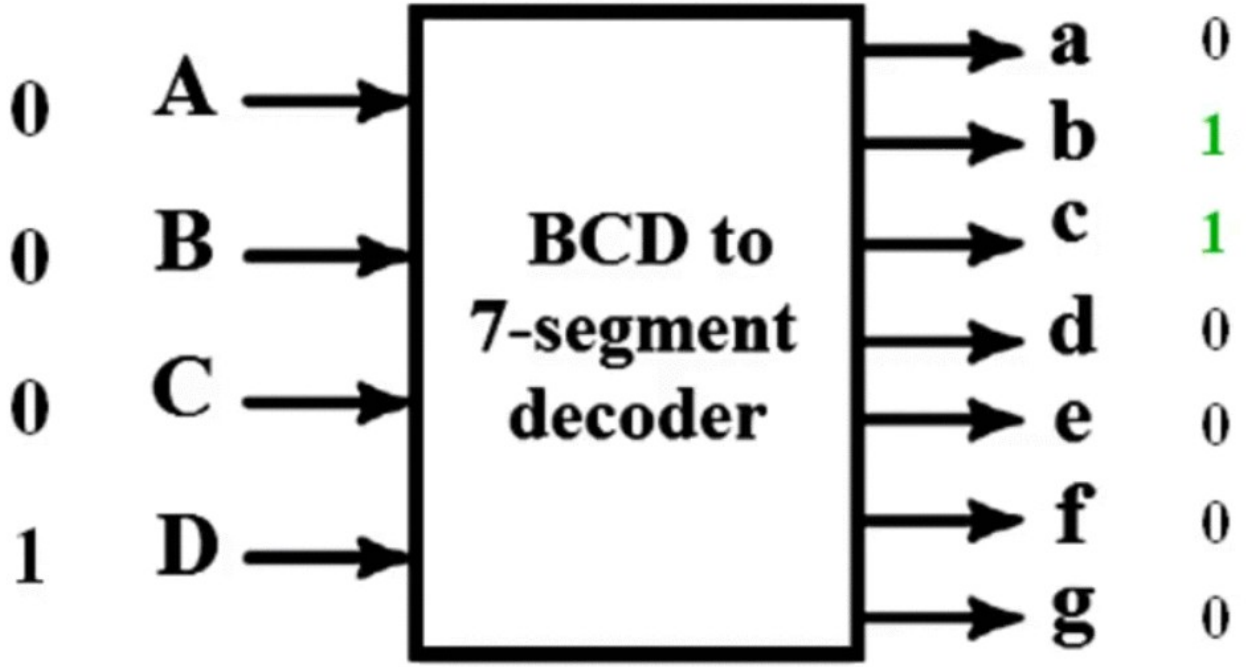
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

BCD to 7-segment decoder



A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

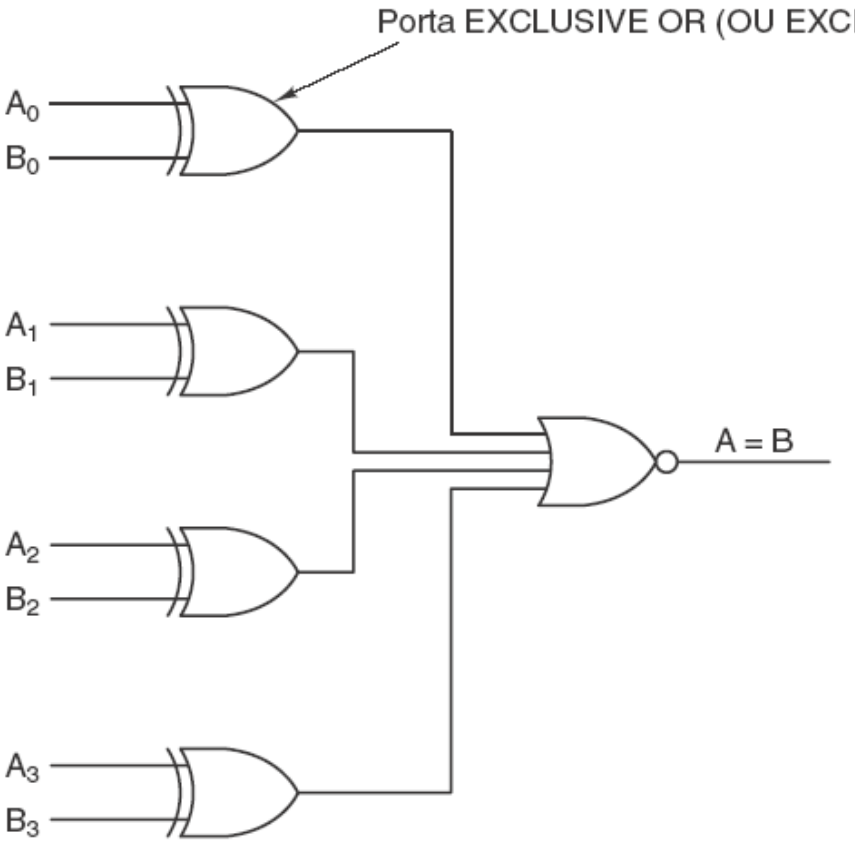
BCD to 7-segment decoder



A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0

CIRCUITOS COMBINACIONAIS

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



CIRCUITOS LÓGICOS

Circuitos aritméticos

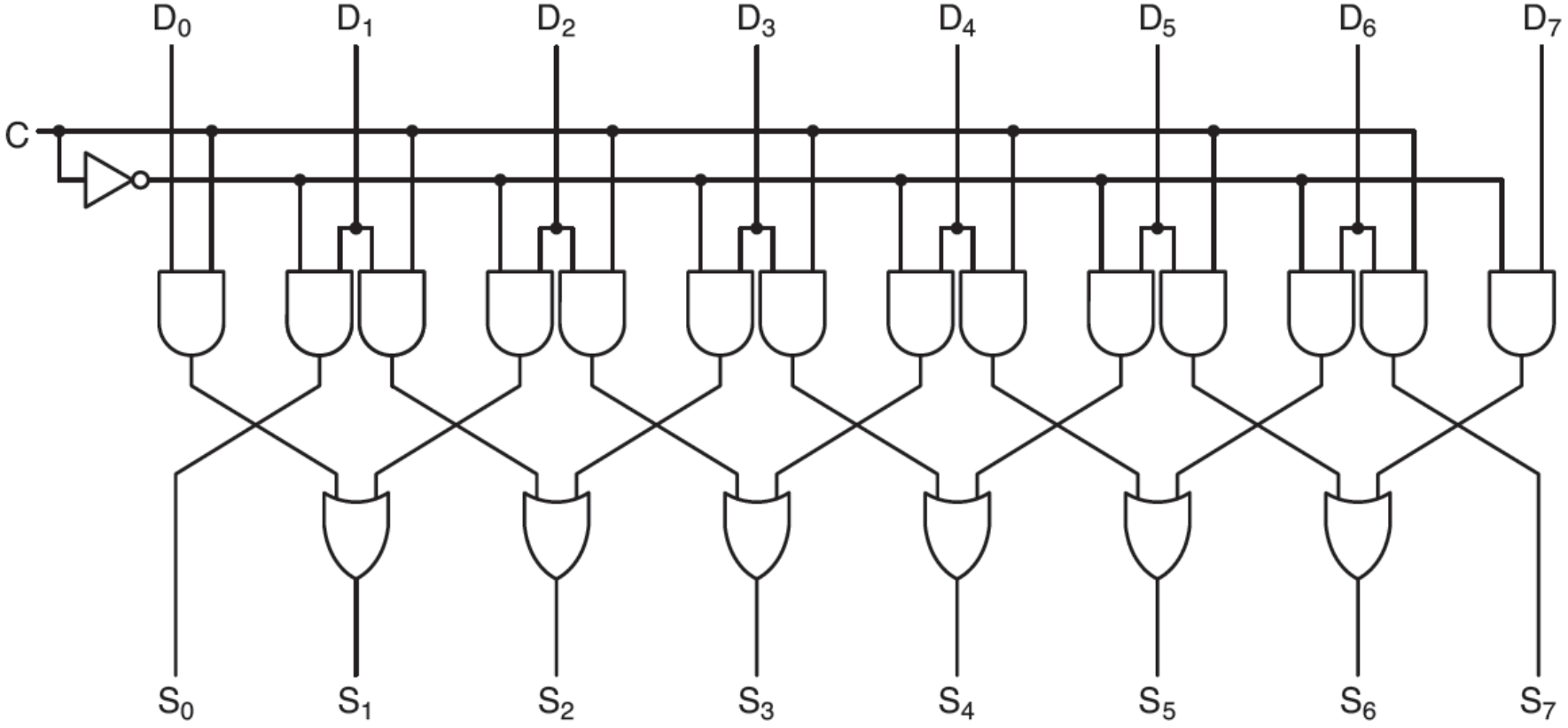
CIRCUITOS ARITMÉTICOS

- Deslocadores

- Somadores
  - Parcial
  - Completo

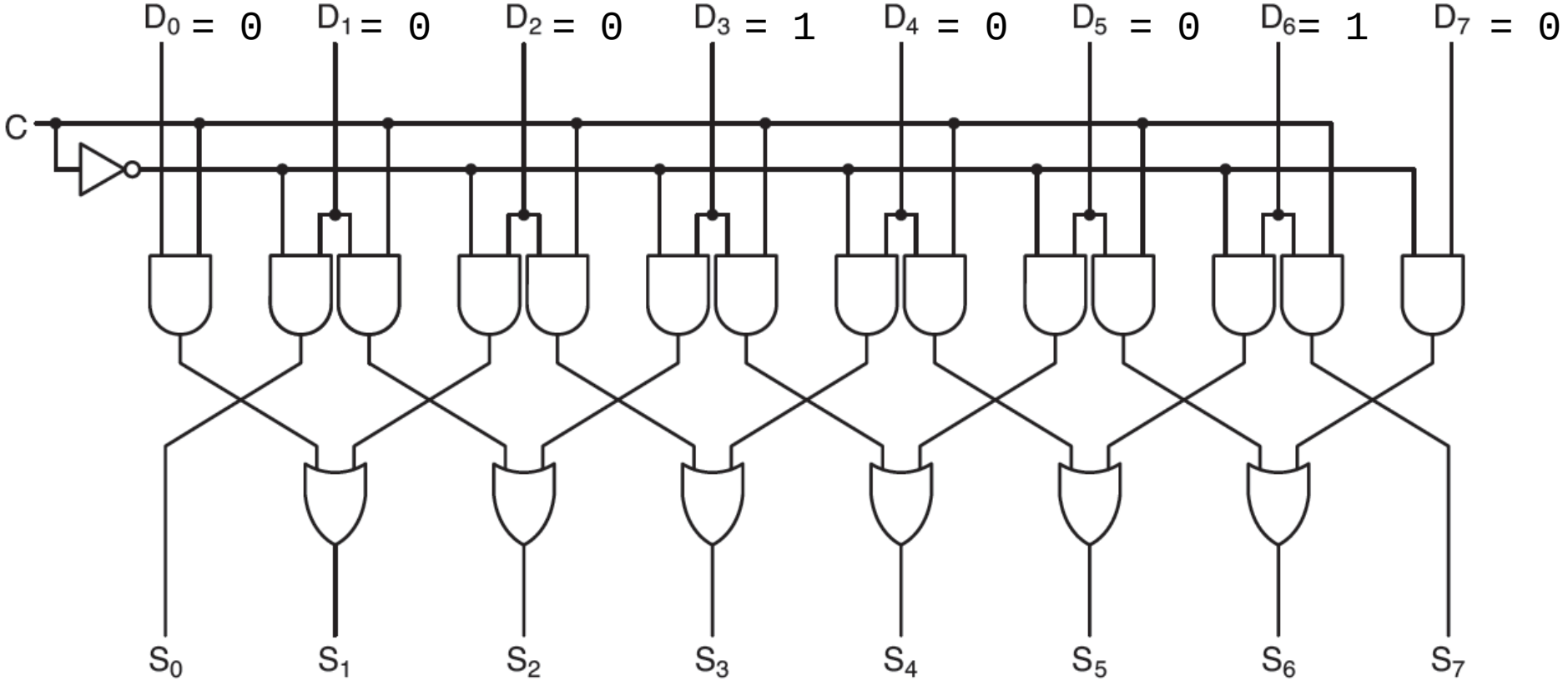
CIRCUITOS ARITMÉTICOS

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



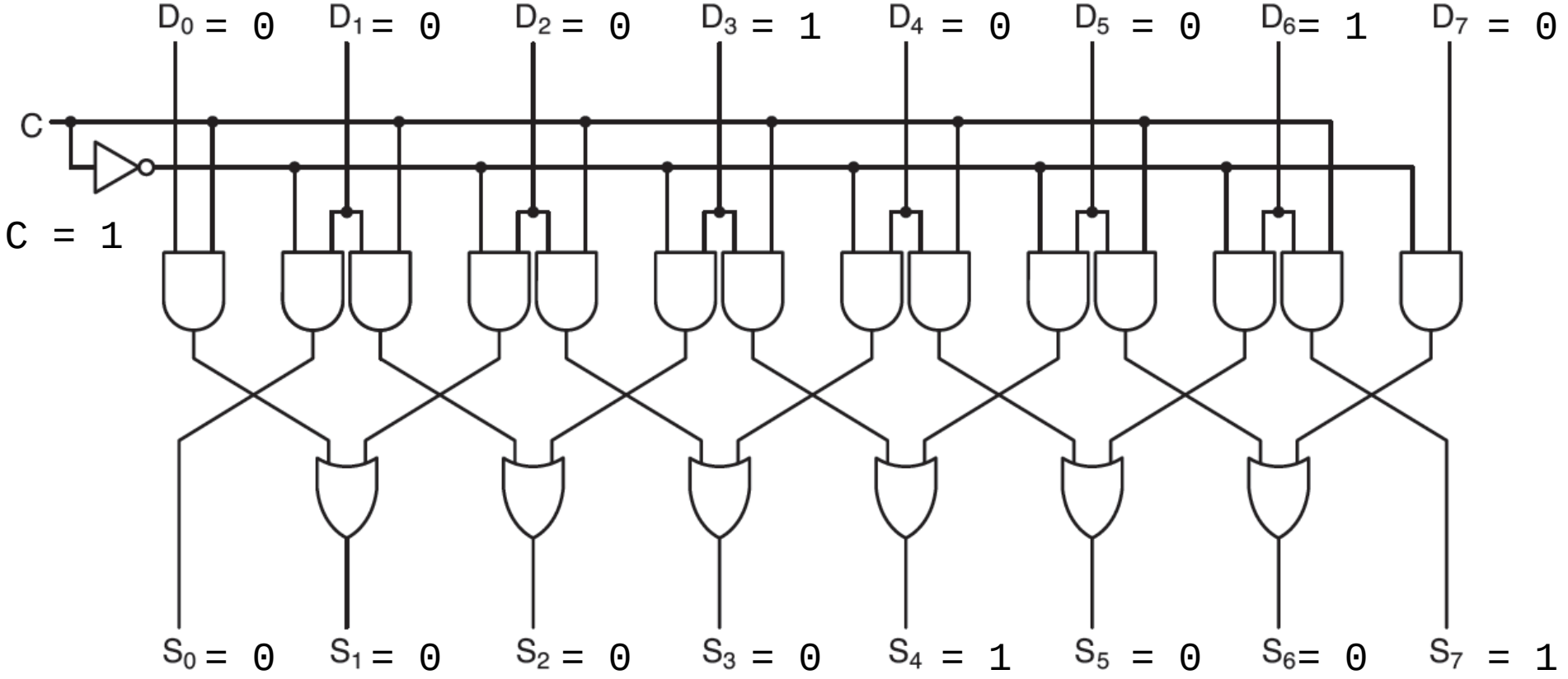
CIRCUITOS ARITMÉTICOS

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



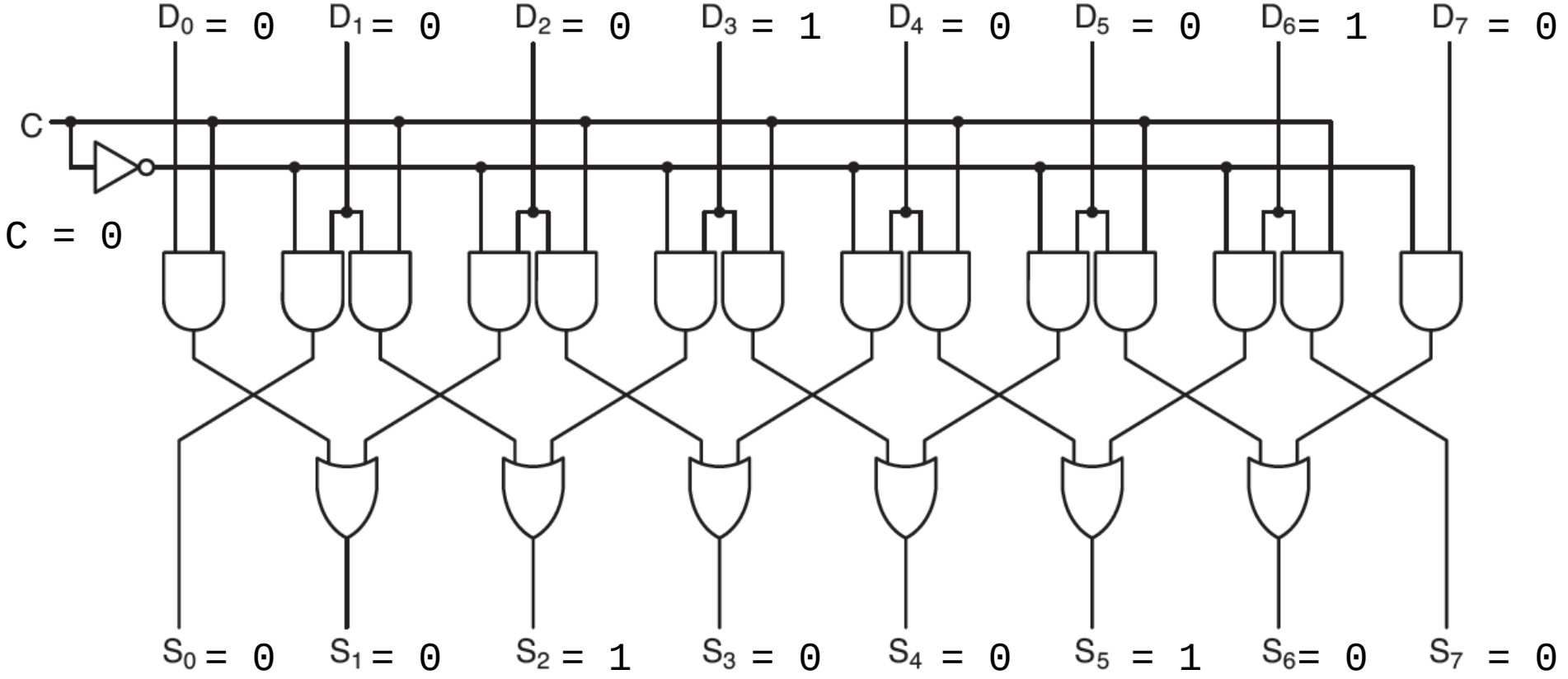
CIRCUITOS ARITMÉTICOS

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



CIRCUITOS ARITMÉTICOS

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



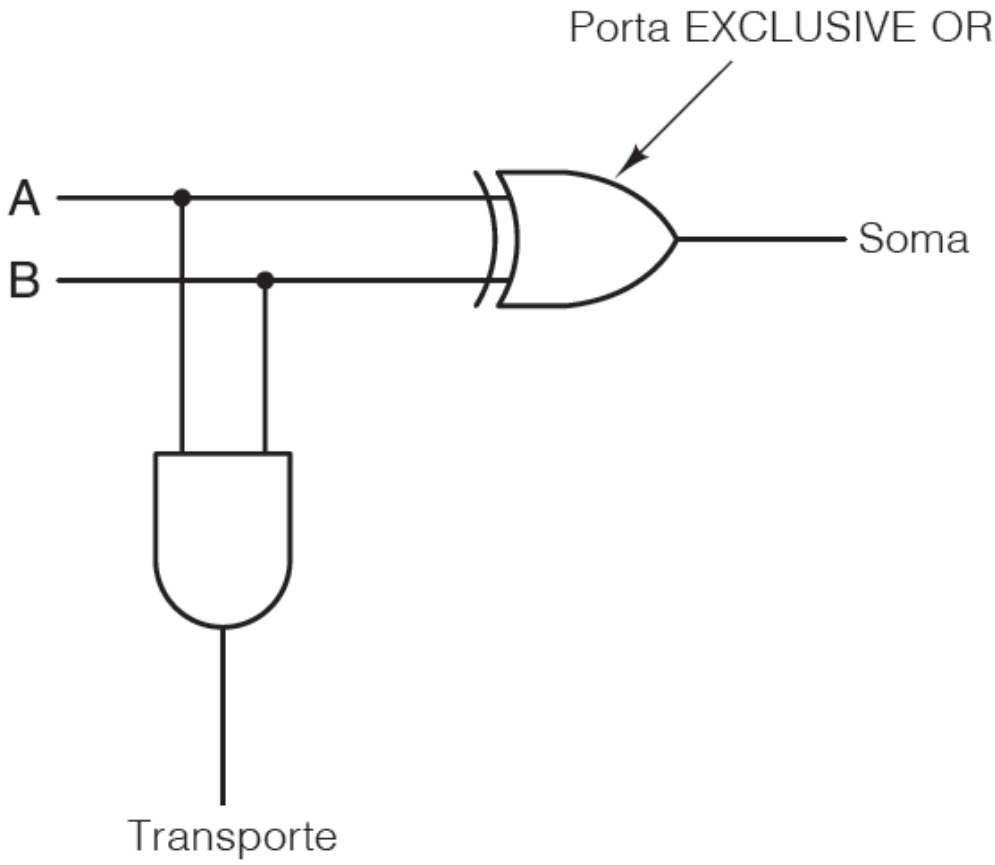
CIRCUITOS ARITMÉTICOS

- Um computador que não possa somar números inteiros é quase inimaginável
- Um *hardware* para efetuar adição é parte essencial em toda CPU
- Somador Parcial
- Somador Completo

CIRCUITOS ARITMÉTICOS

- Um circuito para calcular o bit de soma e o de transporte é conhecido como um meio-somador (somador parcial)

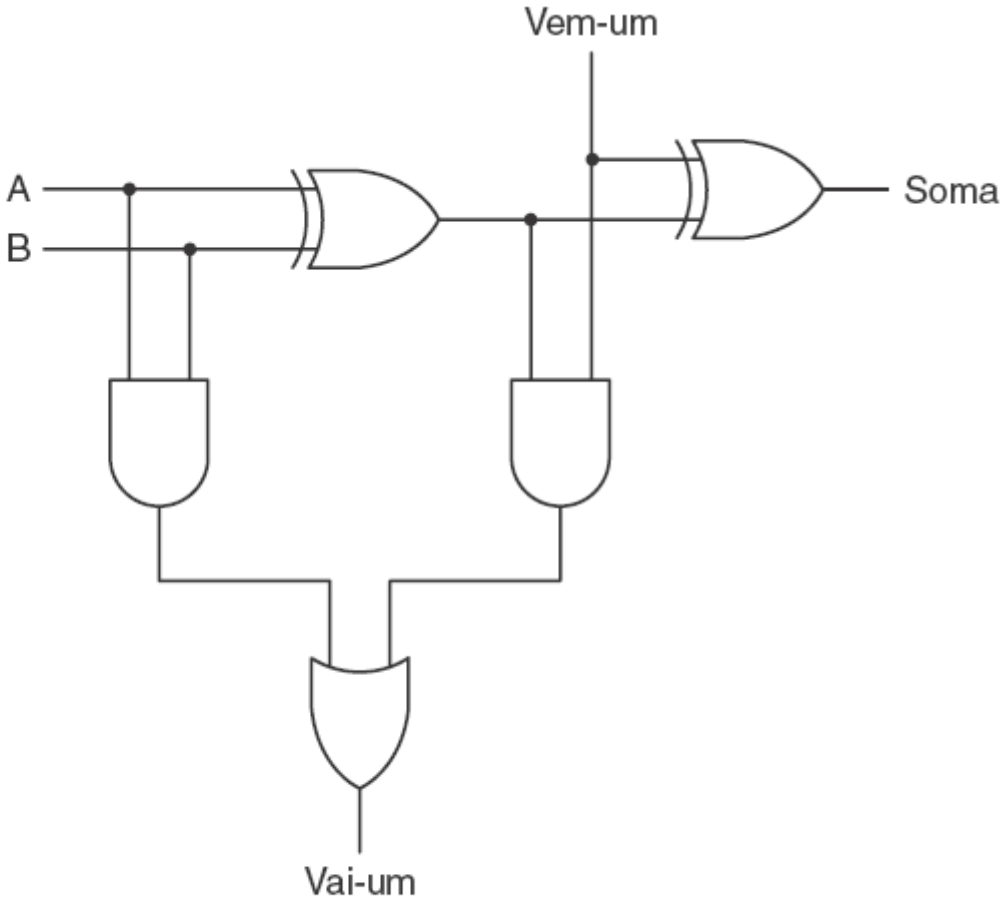
A	B	Soma	Transporte
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



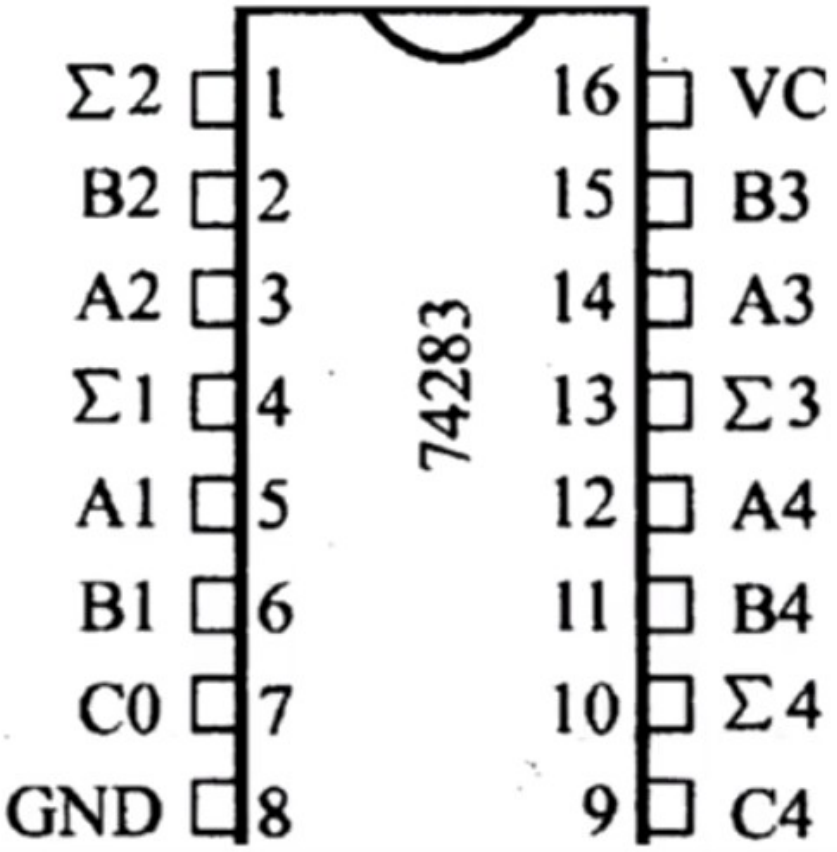
CIRCUITOS ARITMÉTICOS

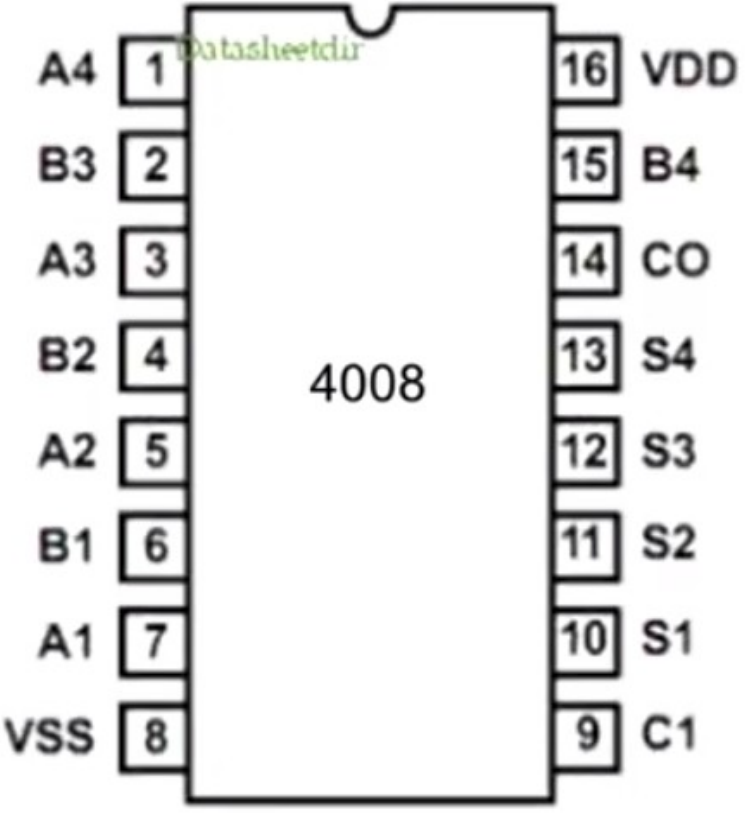
- Somador completo

A	B	Vem-um	Soma	Vai-um
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

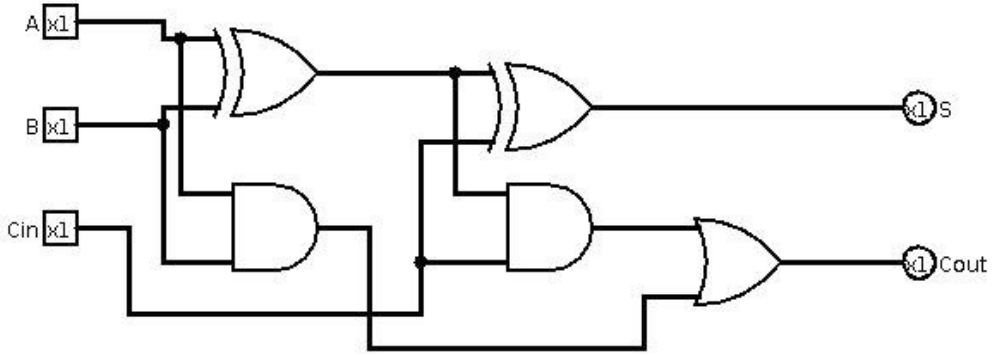


CIRCUITOS ARITMÉTICOS

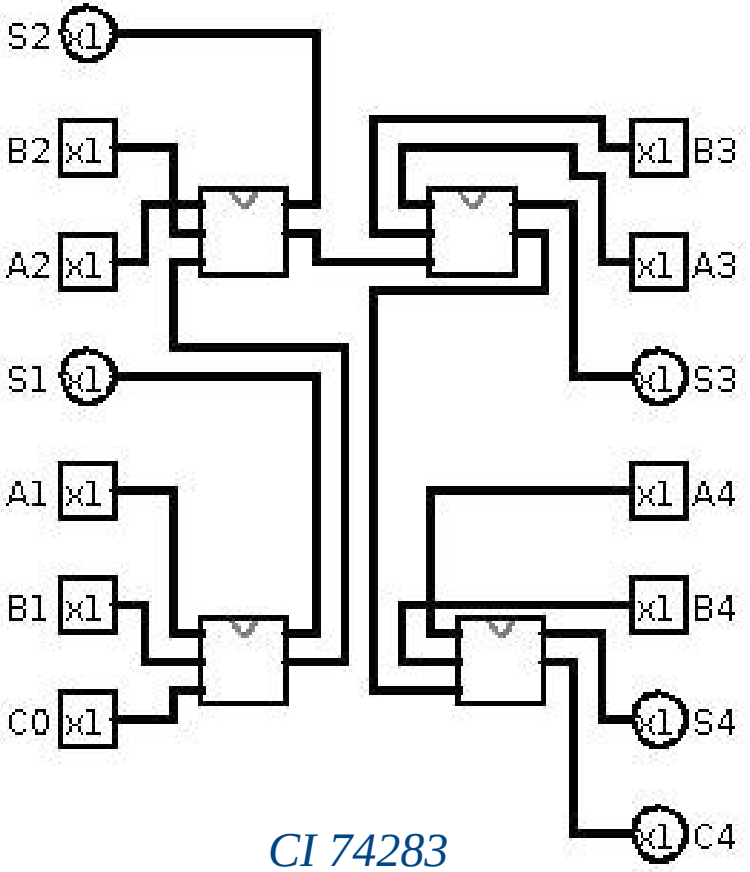




CIRCUITOS ARITMÉTICOS



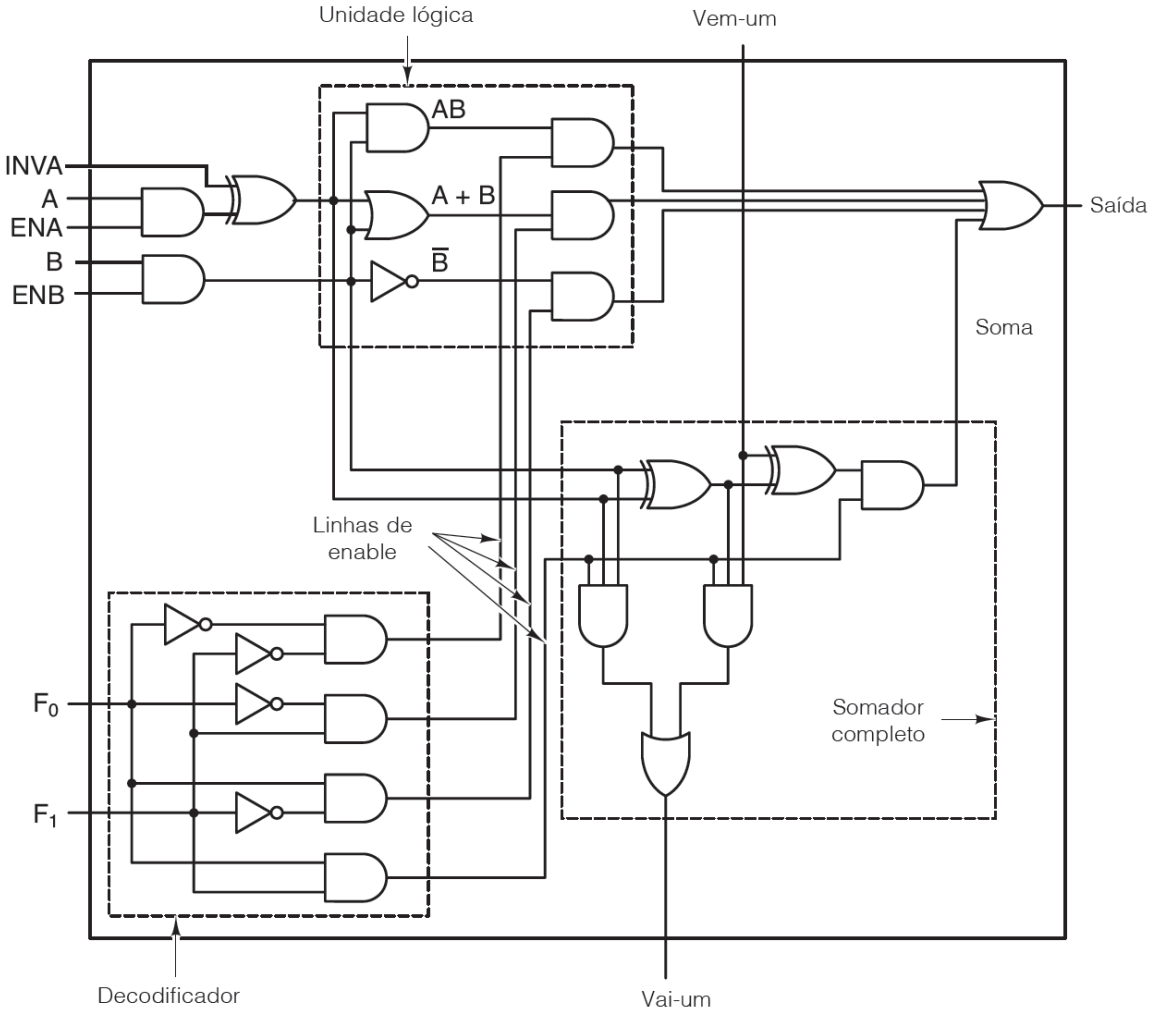
Somador completo



CI 74283

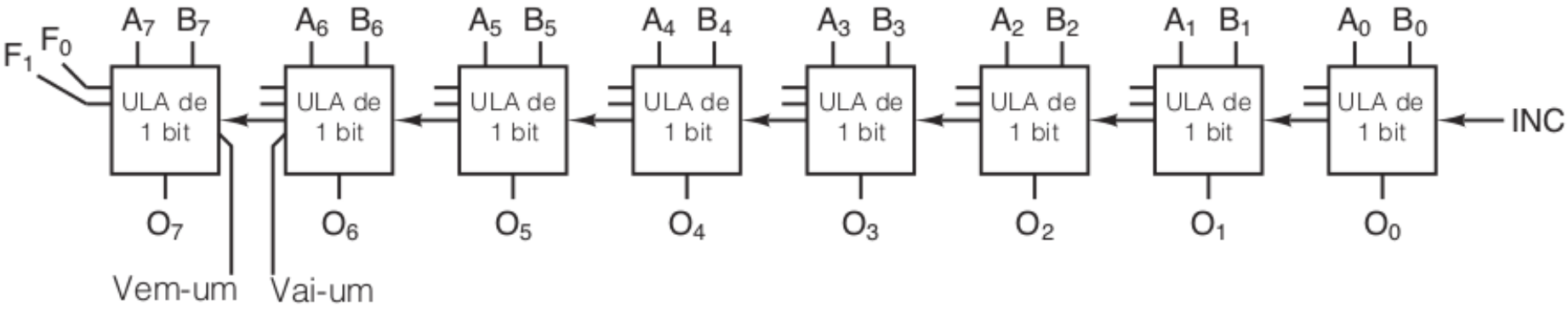
CIRCUITOS ARITMÉTICOS

- ULA de 1 bit.



CIRCUITOS ARITMÉTICOS

- 8 ULAs de 1 bit conectadas para formar uma ULA de 8 bits.
  - Por simplicidade, os sinais de habilitação e de inversão são omitidos
  - **INC** só é necessário para operação de soma. Para quaisquer outras operações, a variação de **INC** não altera o resultado

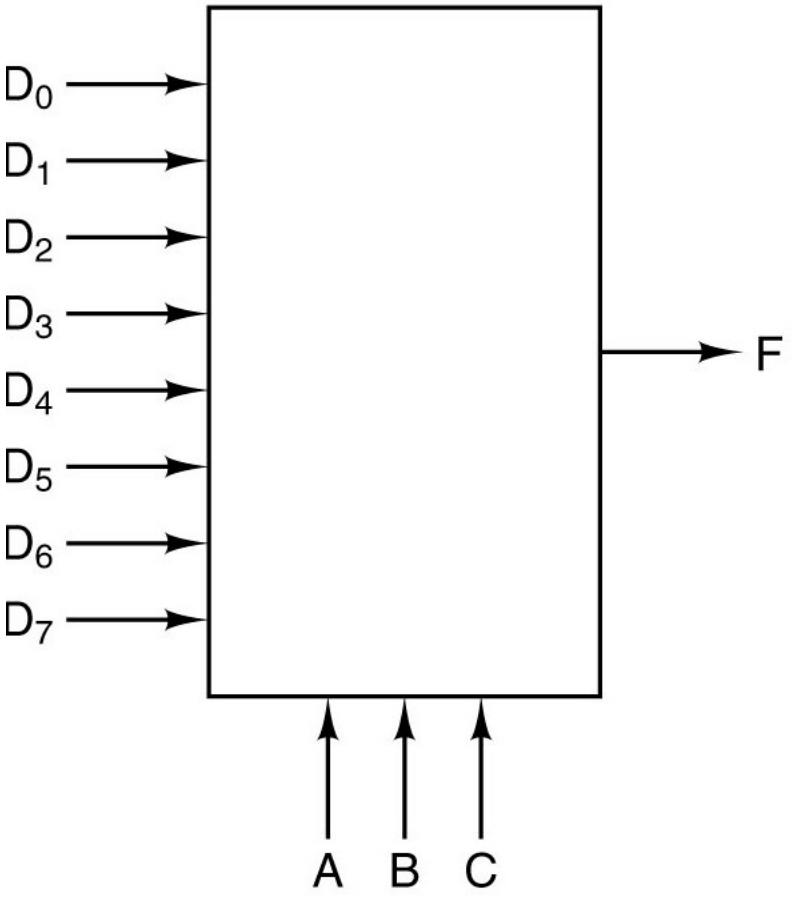


ATIVIDADES

...

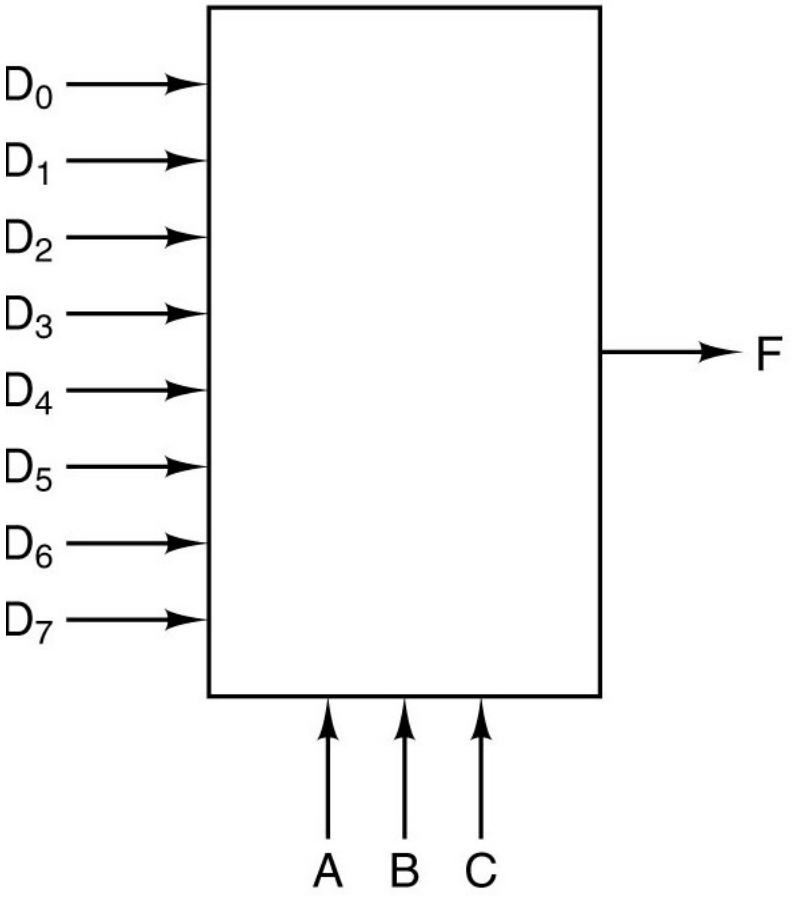
ATIVIDADE 08

- Usando o seguinte chip multiplexador, execute uma função cuja saída é paridade das entradas, isto é, a saída é 1 se, e somente se, um número par de entradas for 1
  - Quais entradas serão ligadas ao V_{CC} e quais serão ligadas ao GND para representar essa função?



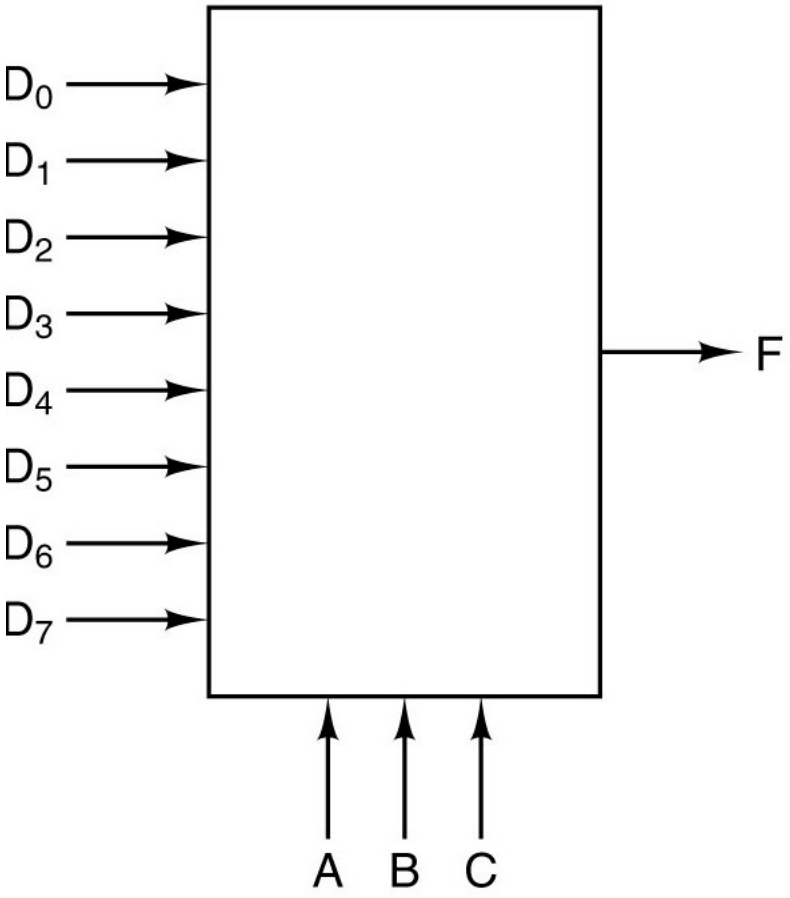
ATIVIDADE 08

- 000 par → 1
- 001 ímpar → 0
- 010 ímpar → 0
- 011 par → 1
- 100 ímpar → 0
- 101 par → 1
- 110 par → 1
- 111 ímpar → 0



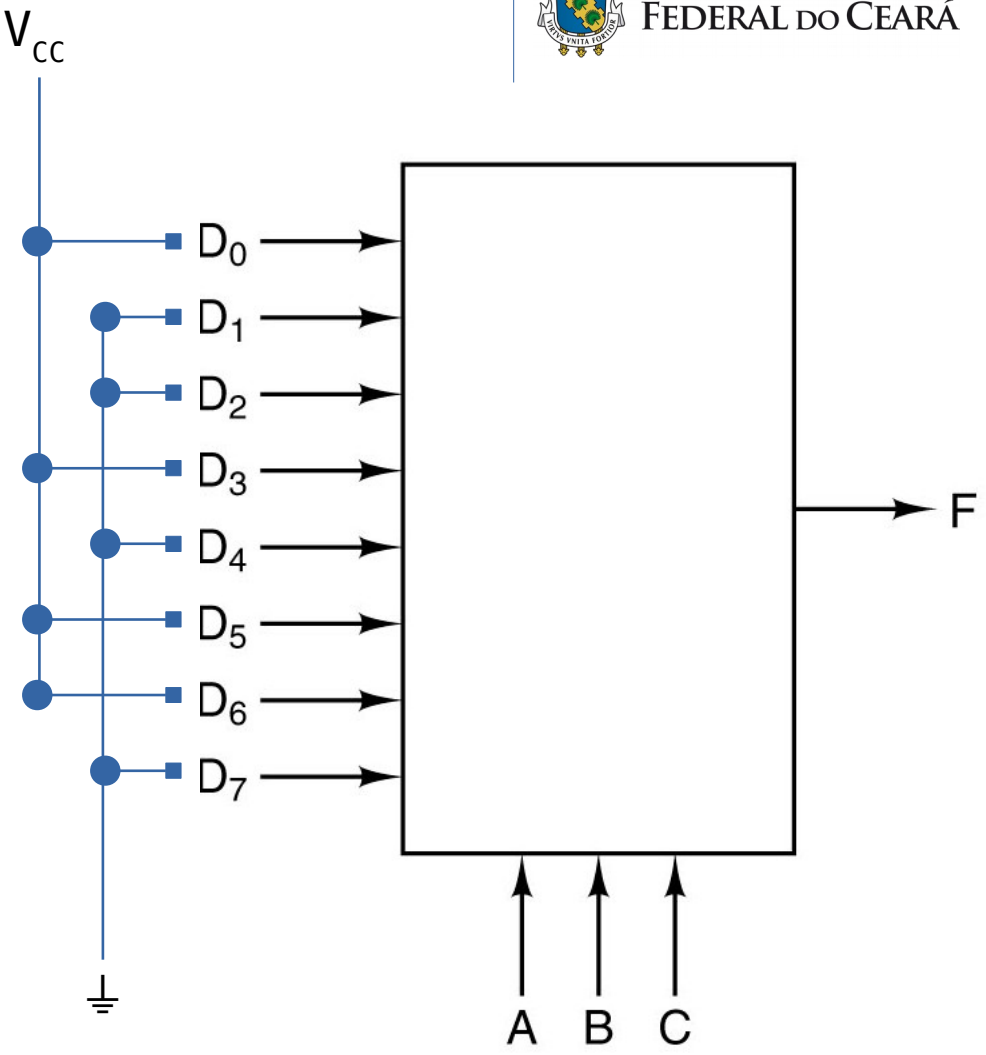
ATIVIDADE 08

- 000 par → V_{cc}
- 001 ímpar → GND
- 010 ímpar → GND
- 011 par → V_{cc}
- 100 ímpar → GND
- 101 par → V_{cc}
- 110 par → V_{cc}
- 111 ímpar → GND



ATIVIDADE 08

- 000 par → V_{cc}
- 001 ímpar → GND
- 010 ímpar → GND
- 011 par → V_{cc}
- 100 ímpar → GND
- 101 par → V_{cc}
- 110 par → V_{cc}
- 111 ímpar → GND

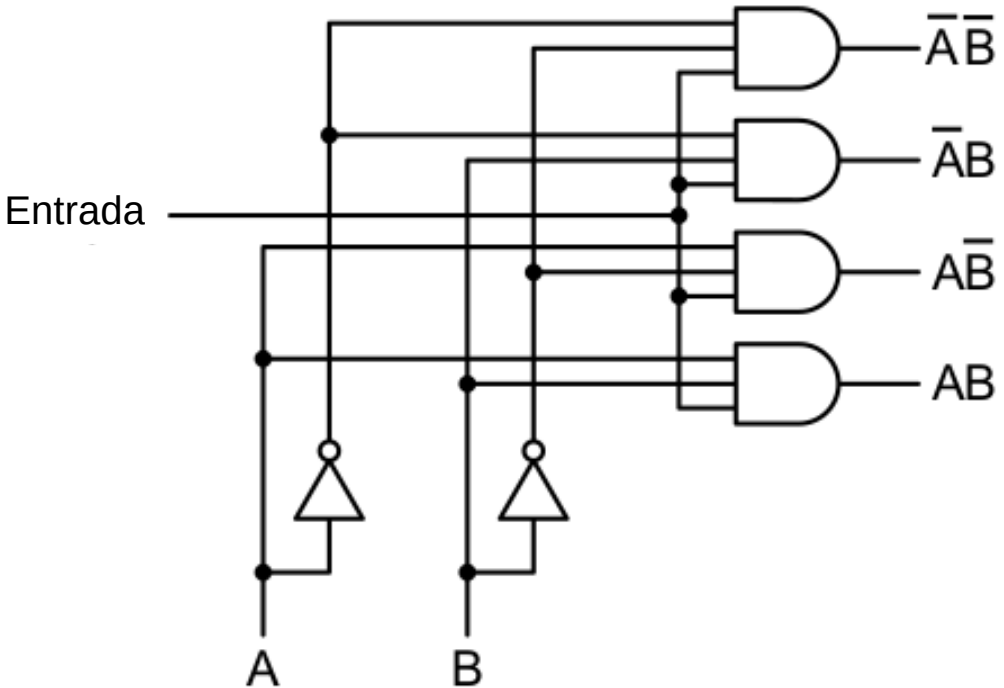


ATIVIDADE 09

- Desenhe o diagrama lógico para um demultiplexador de 2 bits, um circuito cuja única linha de entrada é direcionada para uma das quatro linhas de saída dependendo do estado das duas linhas de controle

ATIVIDADE 09

- Desenhe o diagrama lógico para um demultiplexador de 2 bits, um circuito cuja única linha de entrada é direcionada para uma das quatro linhas de saída dependendo do estado das duas linhas de controle



CIRCUITOS LÓGICOS

Clocks

CLOCKS

- Em inglês, significa relógio

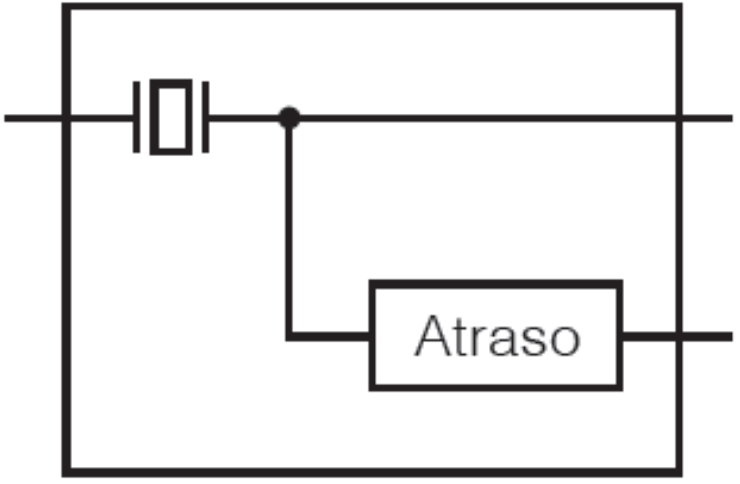
- Para que serve o relógio? Para que haja sincronia entre várias pessoas
  - Ex.: No caso em que uma pessoa que marca uma **consulta médica às 9h**, esse horário é um acordo em comum entre o paciente e o médico – quando o relógio do paciente e o do médico marcarem 9h, a pessoa estará no consultório e o médico terá de lhe atender. Às 9h, ambos deverão estar sincronizados – supõe-se que pouco antes, ambos se preparem para às 9h estarem disponíveis um para o outro

- No computador, o *clock* é uma espécie de **relógio que mantém os circuitos sincronizados**

CLOCKS

- Um ***clock*** é um circuito que emite uma série de pulsos com uma largura de pulso precisa e intervalos precisos entre pulsos consecutivos

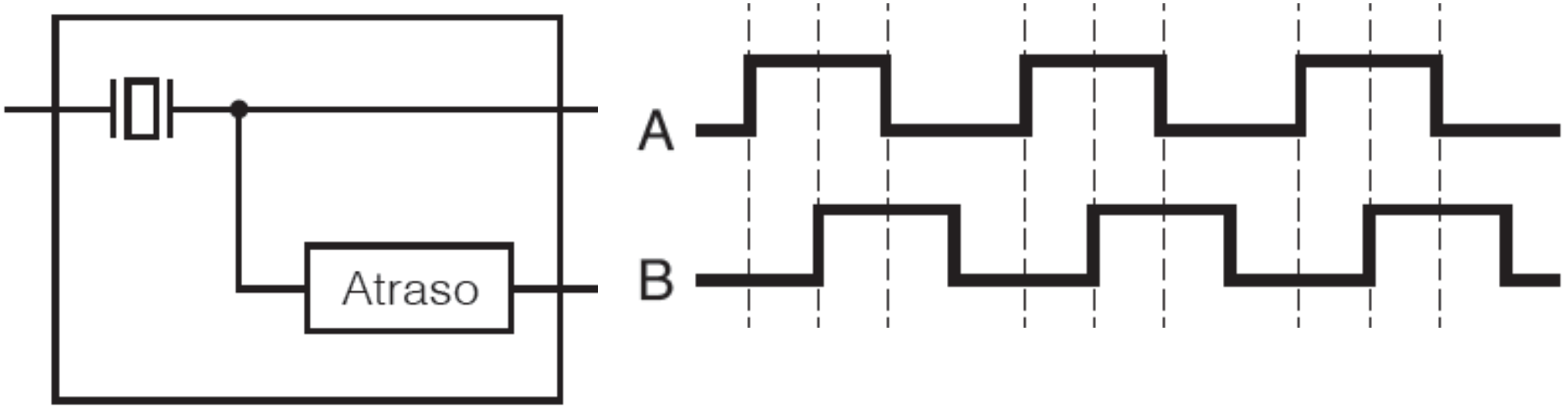
- O intervalo de tempo entre as arestas correspondentes de dois pulsos consecutivos é denominado tempo de ciclo de *clock*



CLOCKS

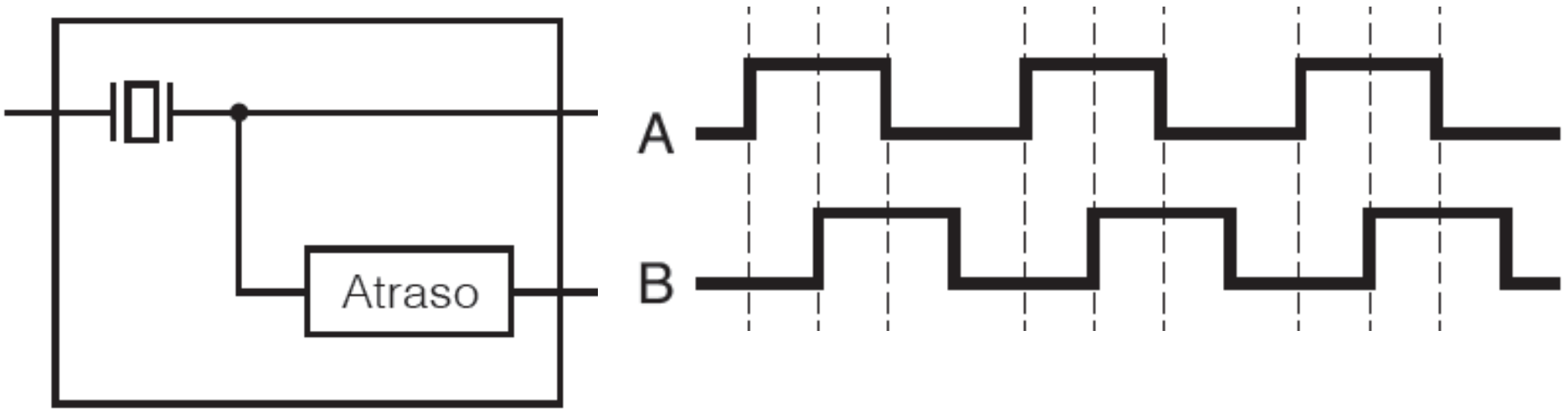
- Um ***clock*** é um circuito que emite uma série de pulsos com uma largura de pulso precisa e intervalos precisos entre pulsos consecutivos

- Durante um ciclo de *clock* diversos eventos podem ocorrer. Se esses eventos devem ocorrer em ordem específica, o ciclo deve ser dividido em subciclos



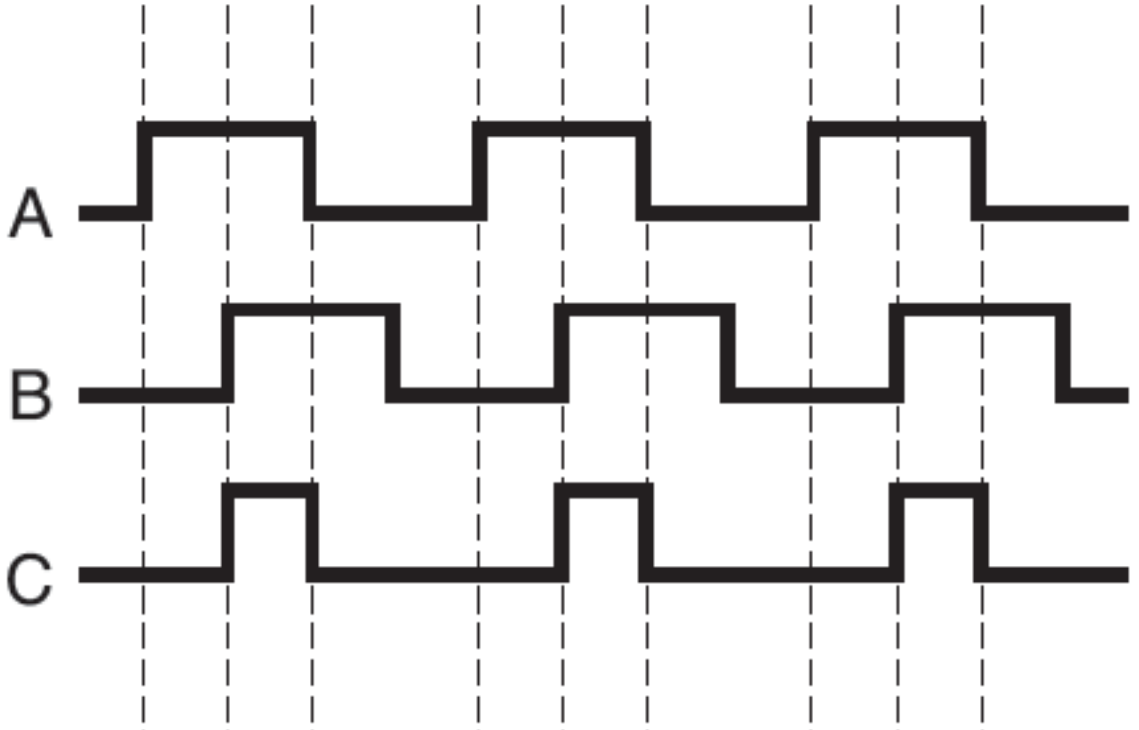
CLOCKS

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



CLOCKS

- Geração de um *clock* assimétrico.



CIRCUITOS LÓGICOS

Circuitos sequenciais

MEMÓRIAS DE 1 BIT

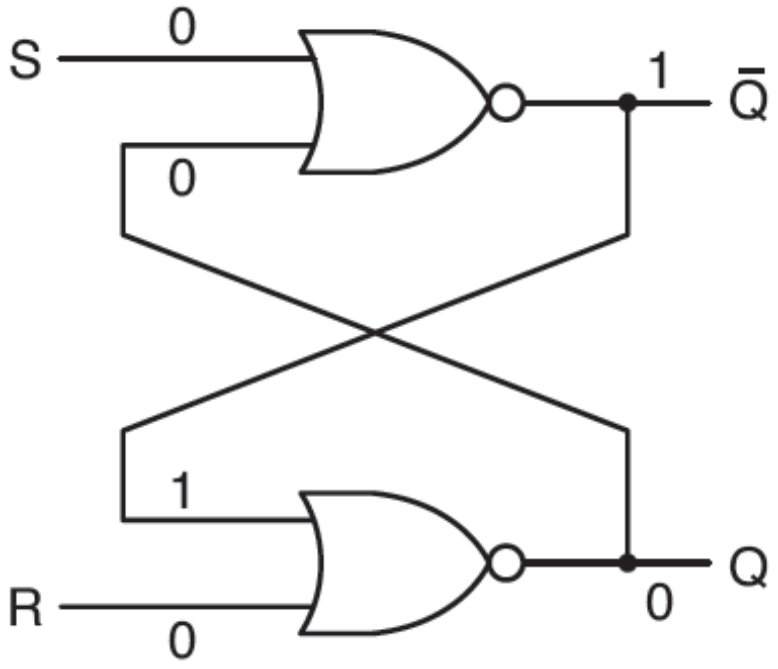
- Para criar uma memória de 1 bit (“*latch*”), precisamos de um circuito que “se lembre”, de algum modo, de valores de entrada anteriores

- Tal circuito pode ser construído com base em duas portas **NOR**:

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

MEMÓRIAS DE 1 BIT

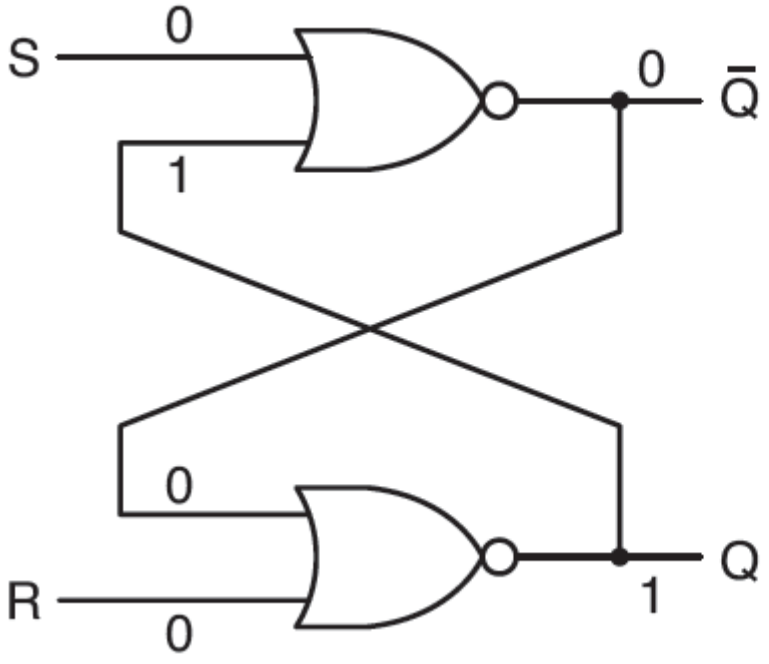
- *Latch* NOR no estado 0. Tabela verdade para NOR



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

MEMÓRIAS DE 1 BIT

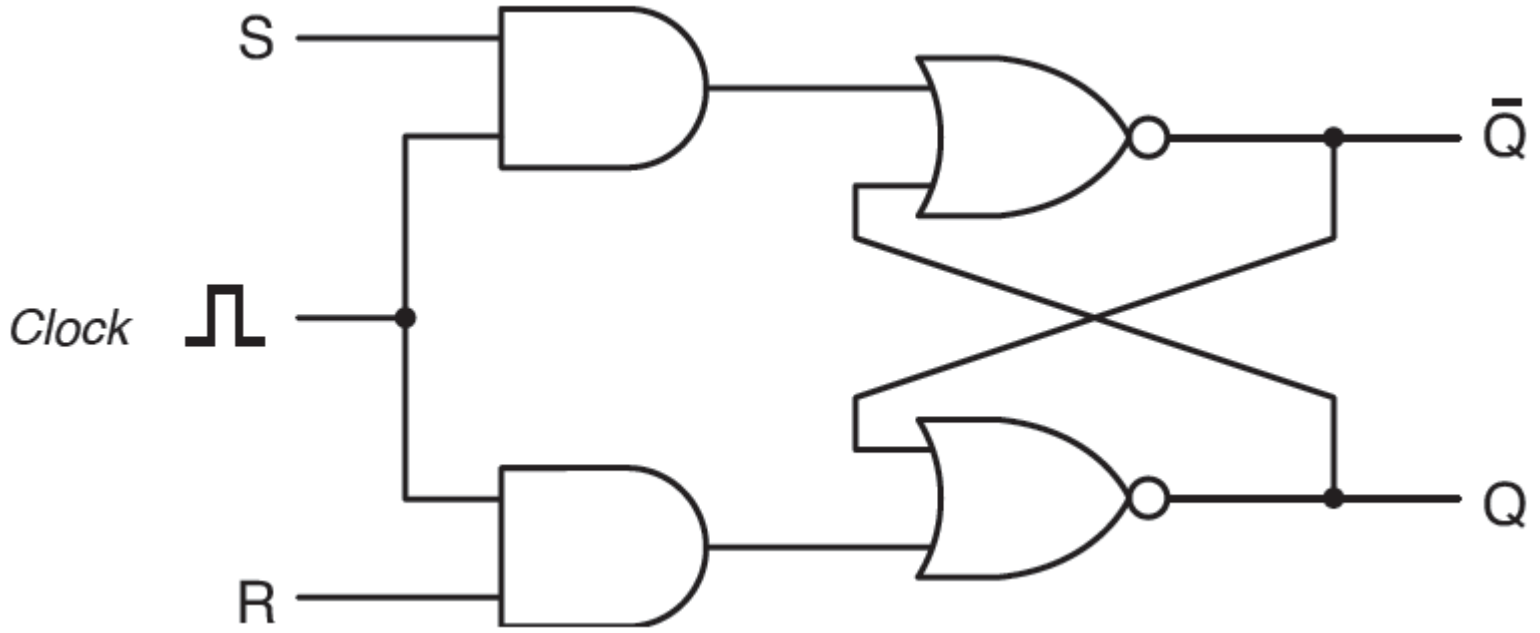
- *Latch* NOR no estado 1. Tabela verdade para NOR



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

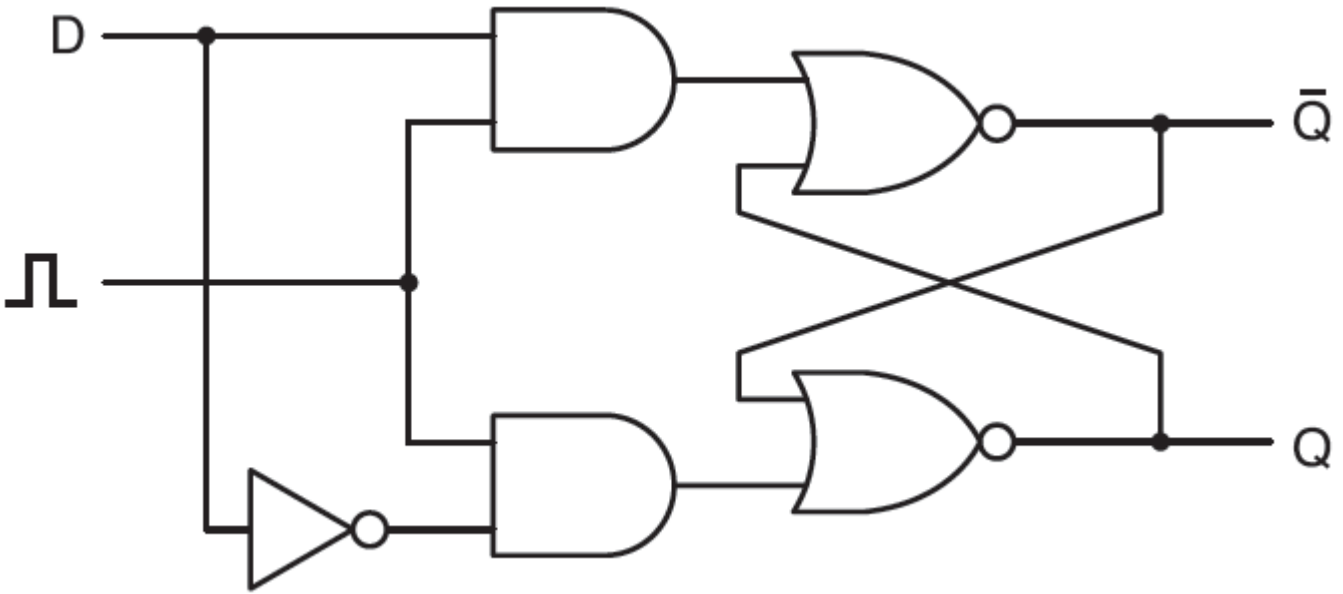
MEMÓRIAS DE 1 BIT

- 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***



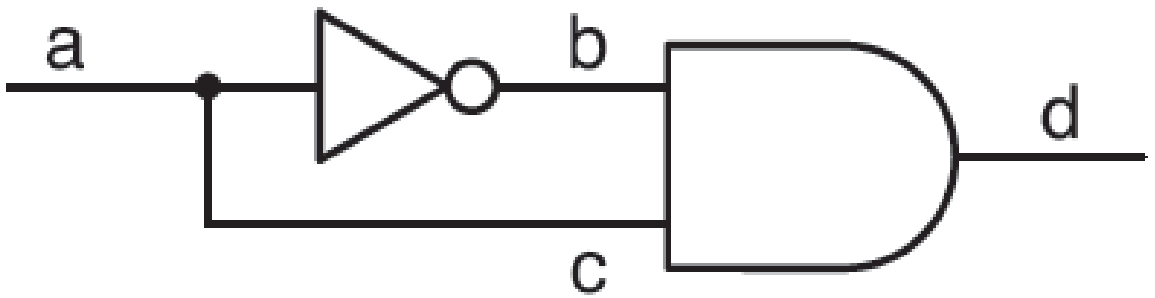
CIRCUITOS LÓGICOS

Flip-flops

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

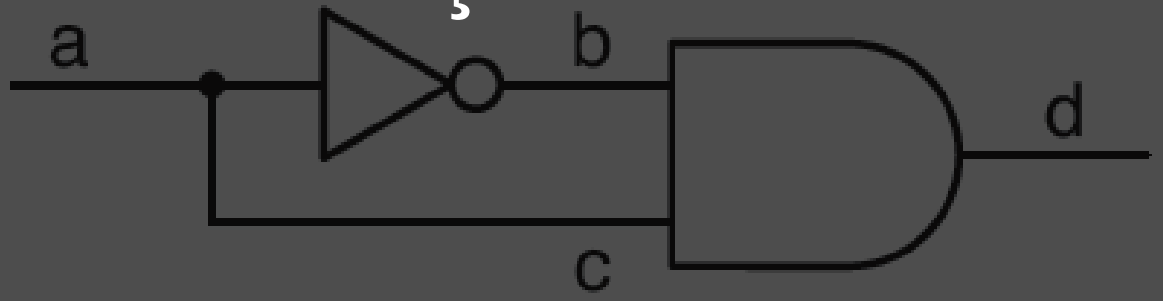


FLIP-FLOPS

- 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)

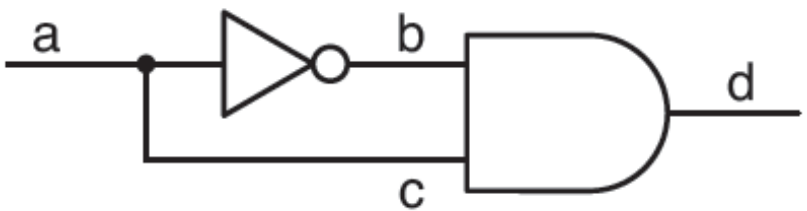
- Gerador de pulso

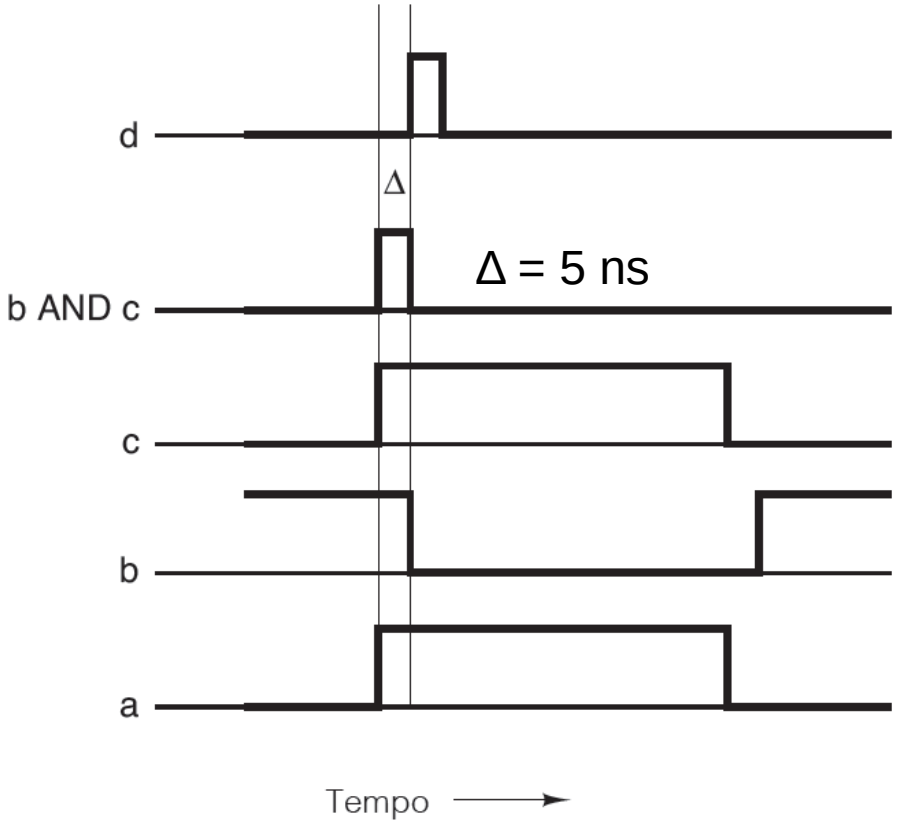
O ATRASO É OUTRA BOA RAZÃO PARA A SIMPLIFICAÇÃO DE CIRCUITOS



FLIP-FLOPS

- Temporização em quatro pontos do circuito.

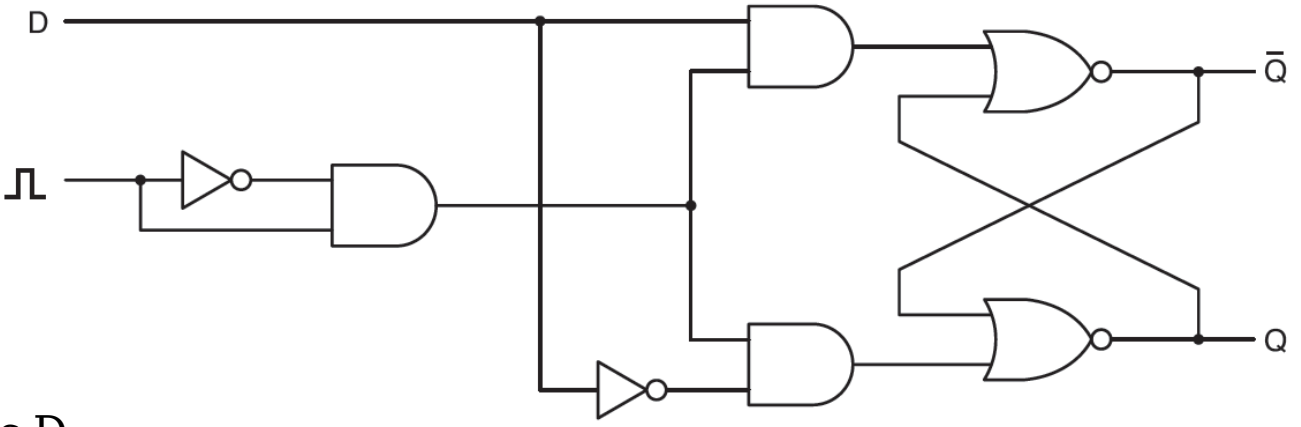




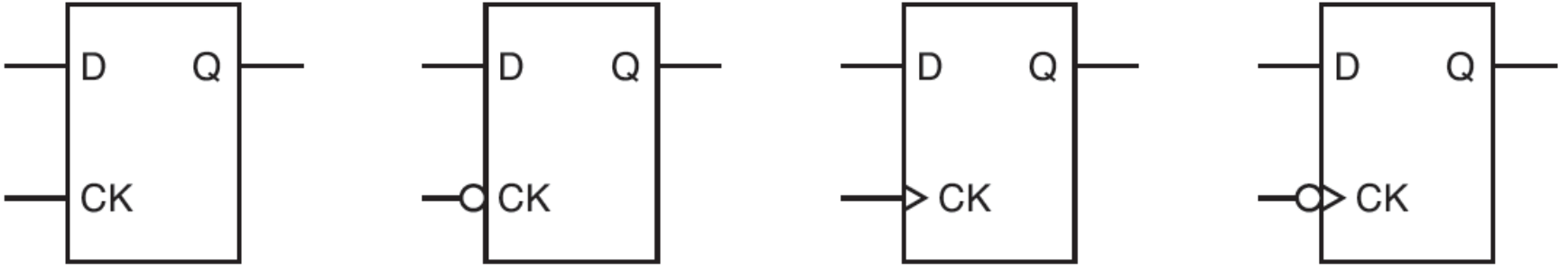
Na prática, o próprio sinal c possui um atraso em relação ao sinal a, porém é bem menor do que o atraso na inversora

FLIP-FLOPS

- *Flip-flop* D



- *Latches* e *flip-flops* D

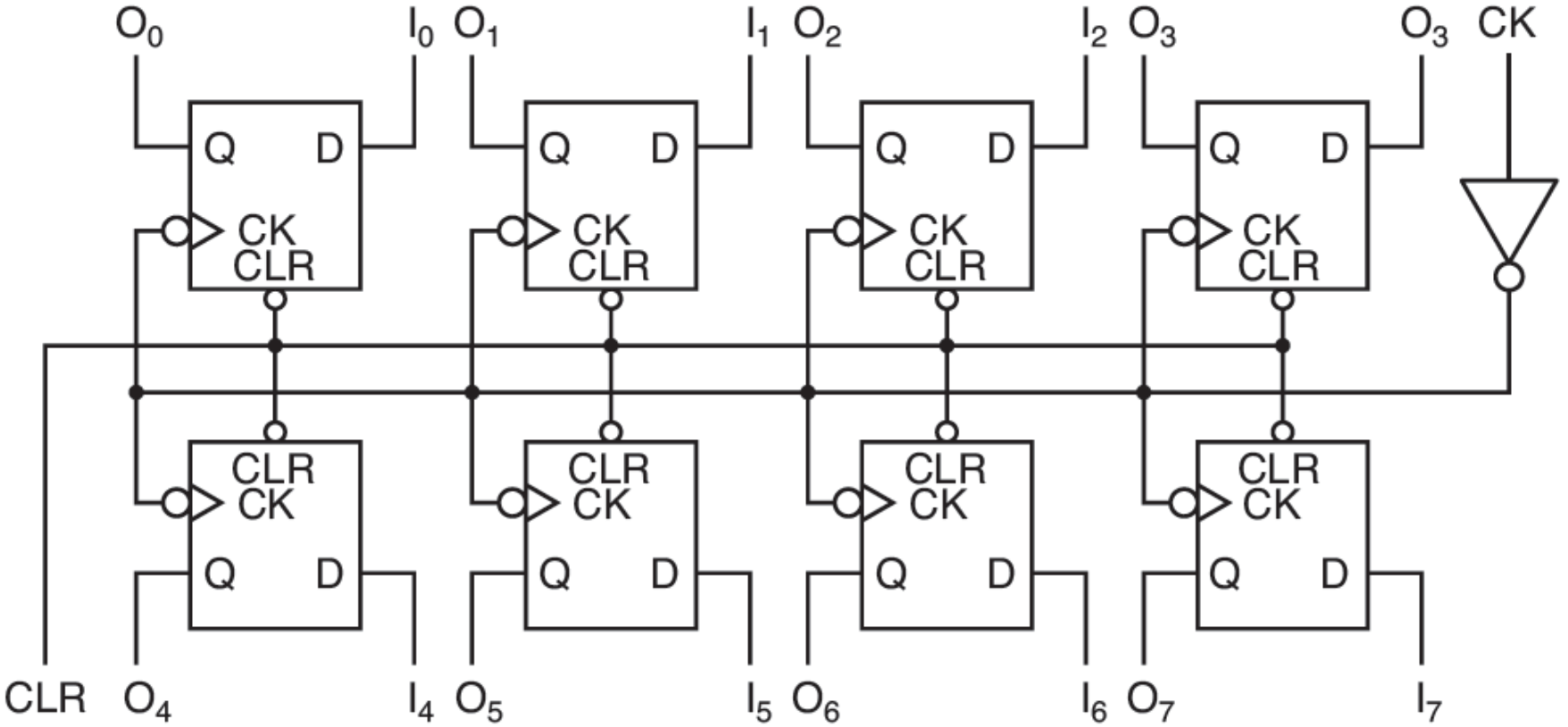


CIRCUITOS LÓGICOS

Registradores

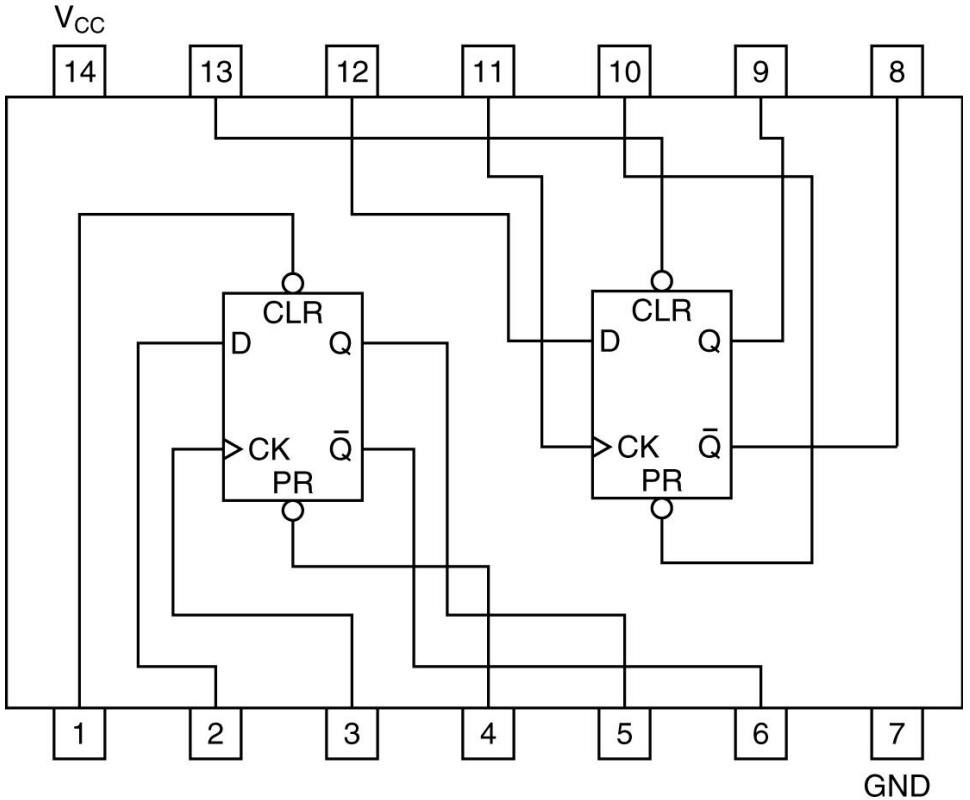
REGISTRADORES

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



REGISTRADORES

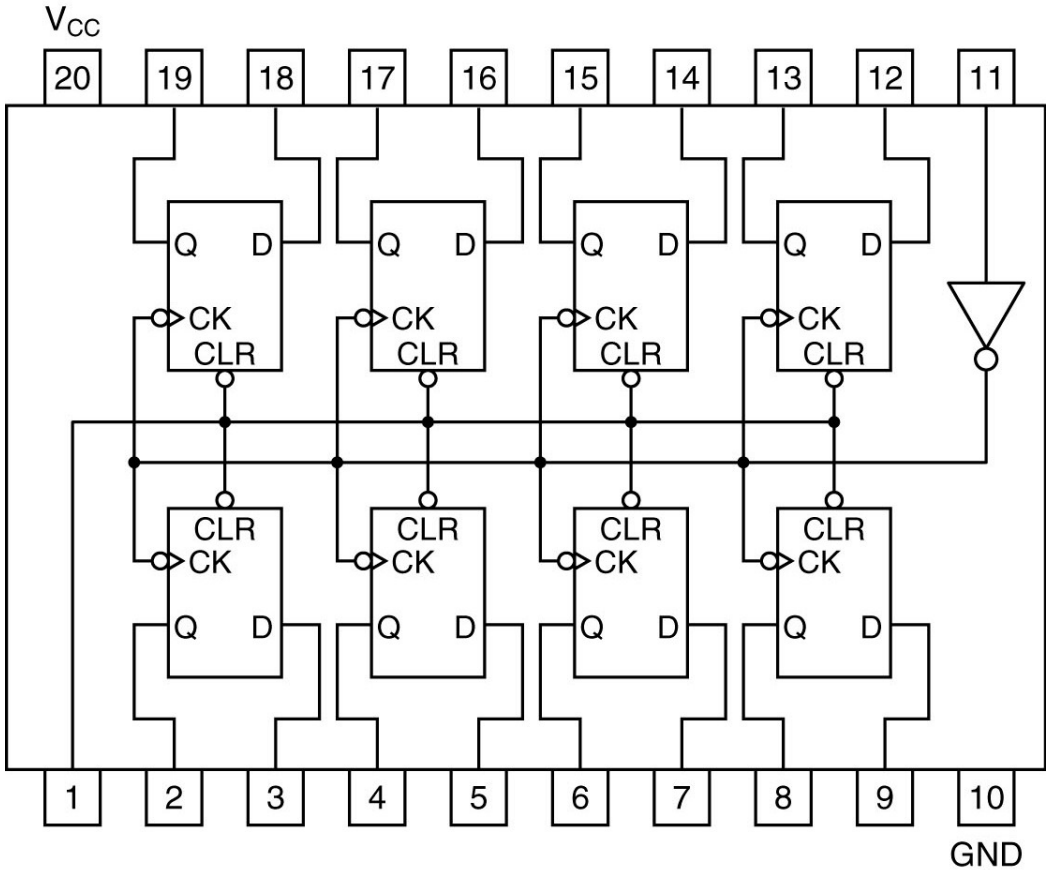
- *Flip-flop* do tipo D dual



(a)

REGISTRADORES

- *Flip-flop* do tipo D octal

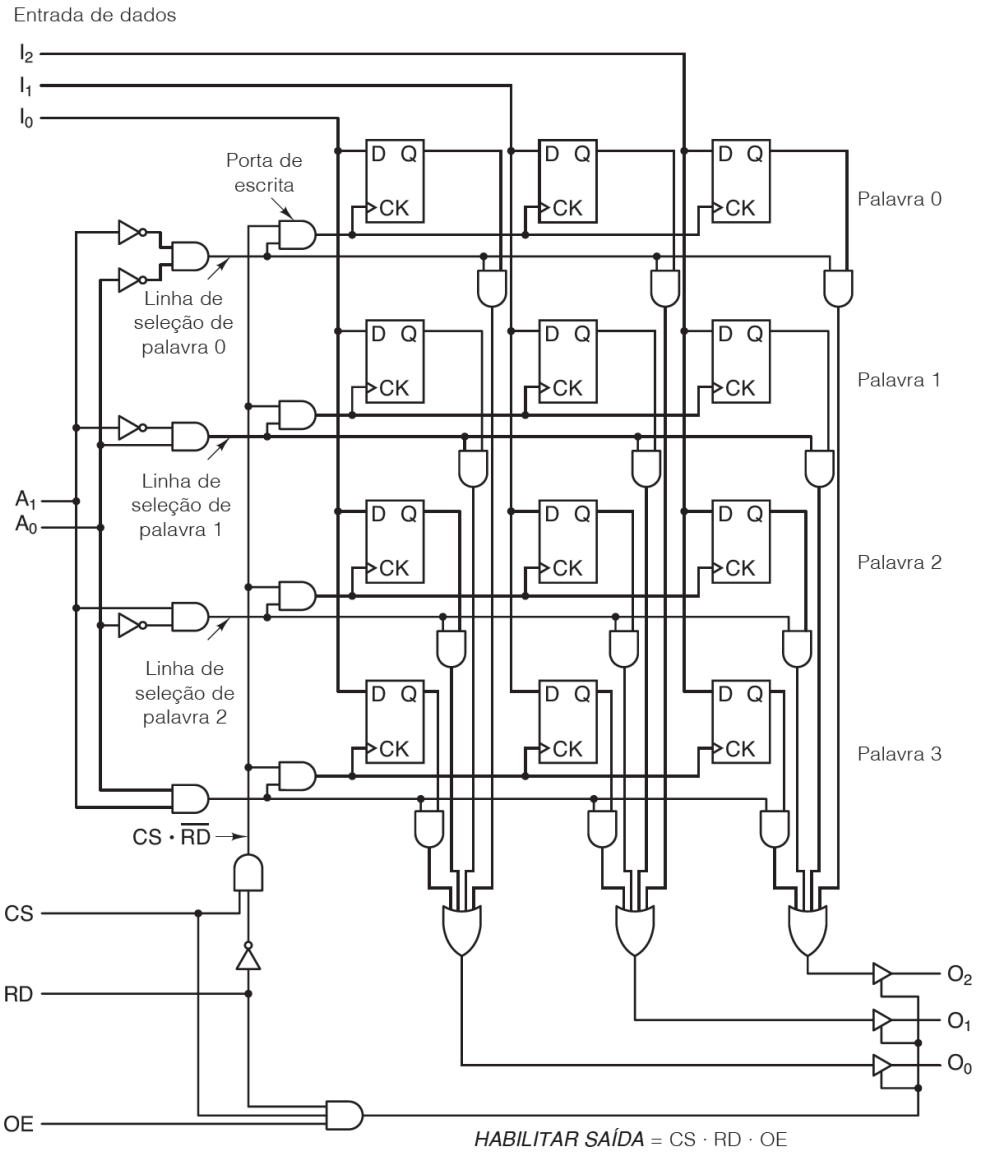


CIRCUITOS LÓGICOS

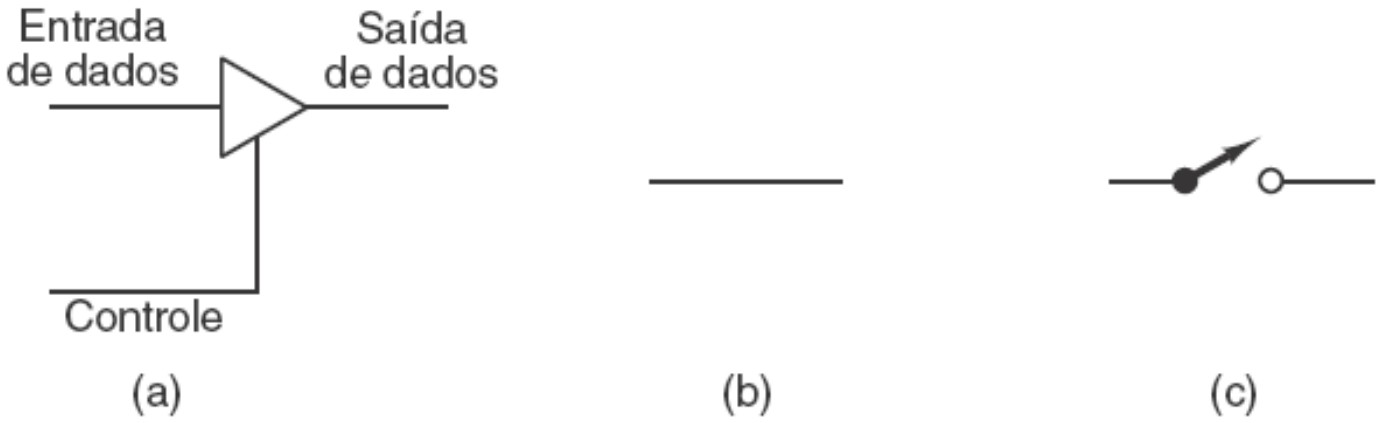
Organização da memória

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



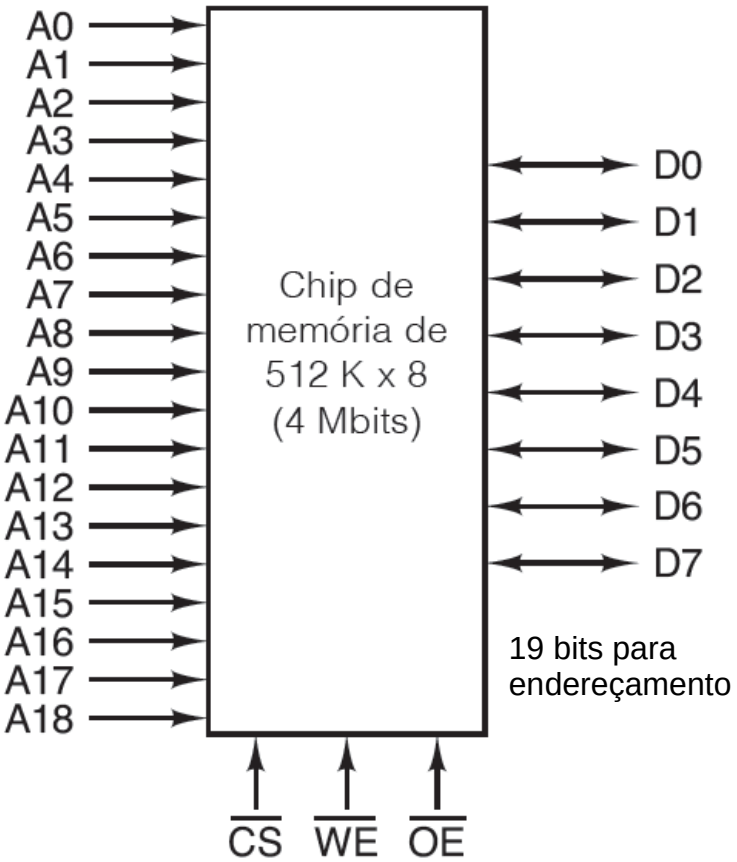
a) *Buffer*, não inversor

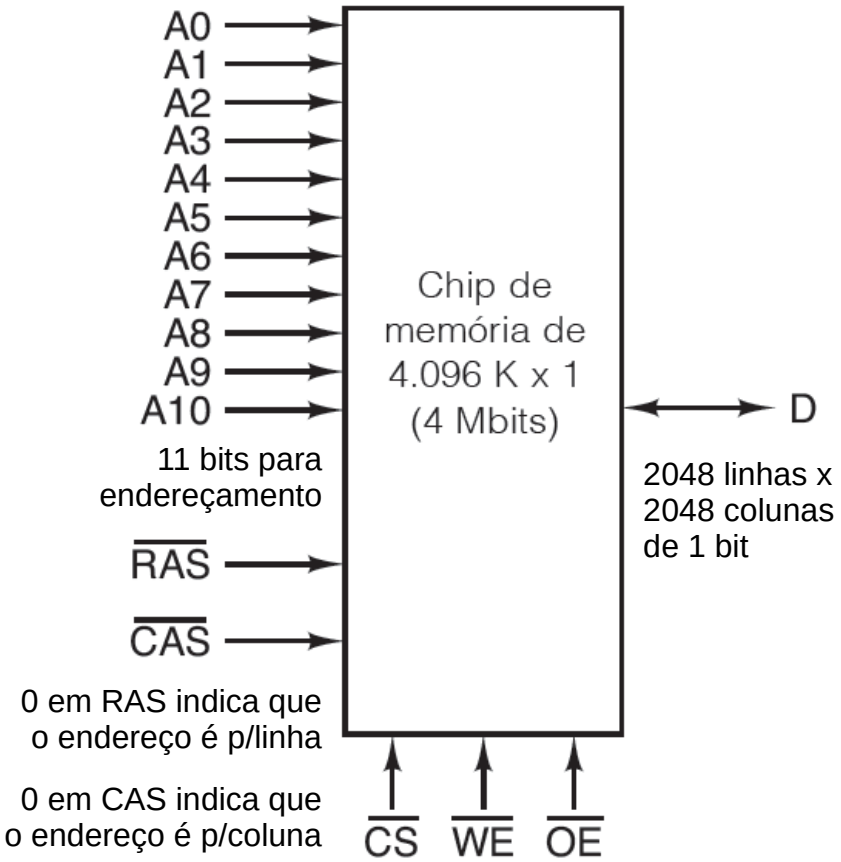
b) Efeito de (a) quando o controle está alto

c) Efeito de (a) quando o controle está baixo

CHIPS DE MEMÓRIA

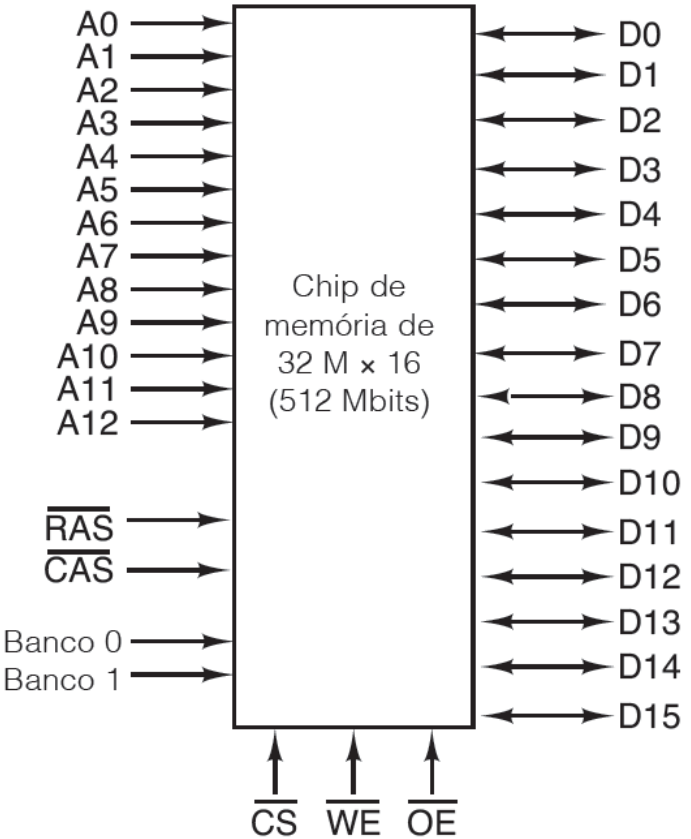
- Dois modos de organizar um chip de memória de 4 Mbits.

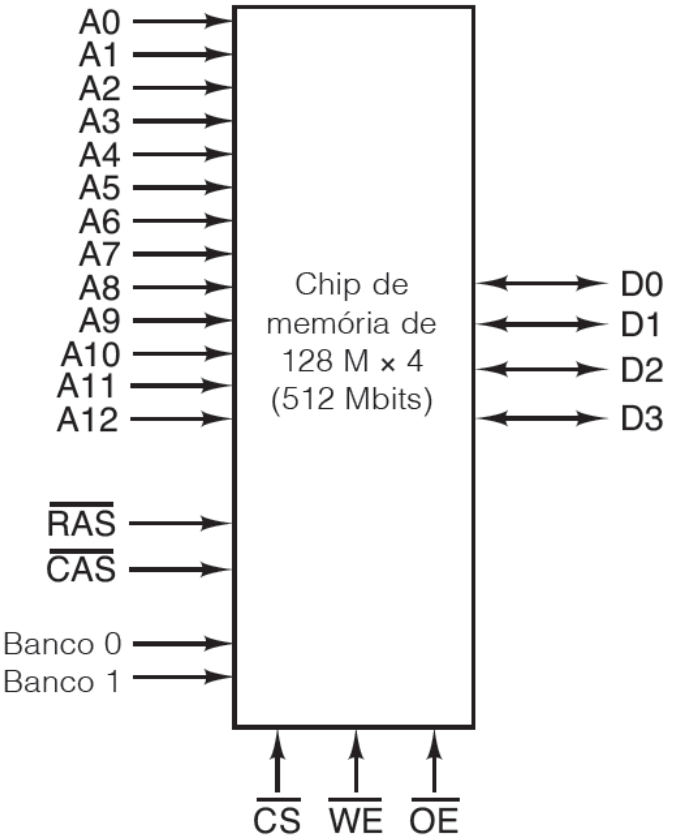




CHIPS DE MEMÓRIA

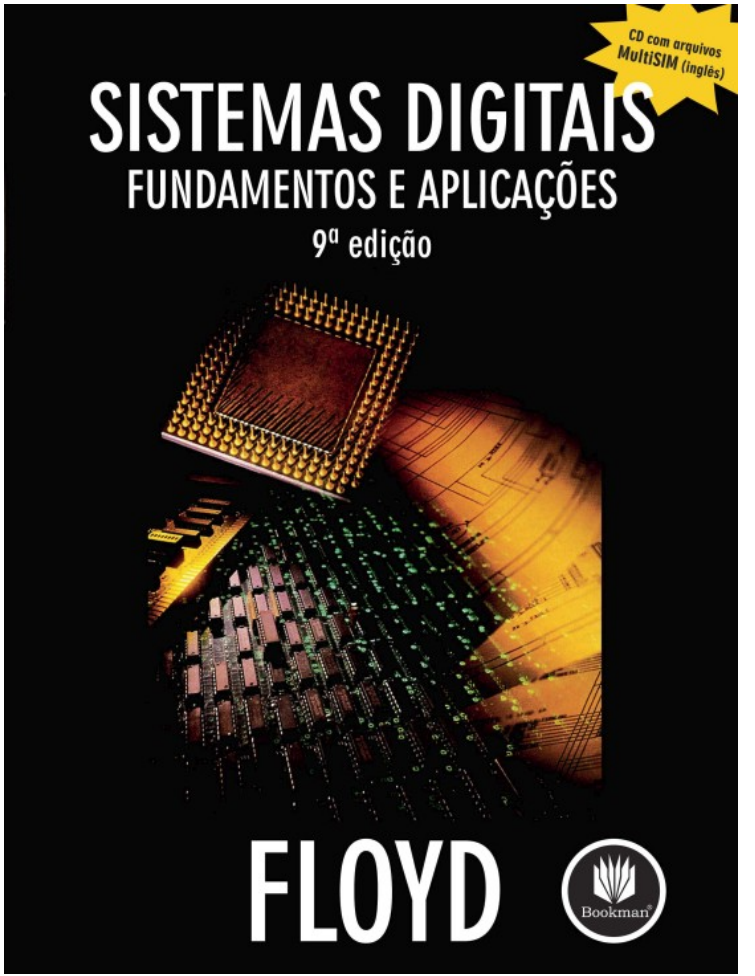
- Dois modos de organizar um chip de memória de 512 Mbits.





REFERÊNCIA SOBRE SISTEMAS DIGITAIS

- Para aprofundar no assunto, consulte
  - FLOYD, Sistemas Digitais



JOSEPH SOARES ALCÂNTARA

josephsoaresalcantara@gmail.com

ENGENHARIA DA COMPUTAÇÃO
2018.2

UNIVERSIDADE FEDERAL DO CEARÁ