



Prof. André Breda Carneiro
Prof. Rafael R. da Paz

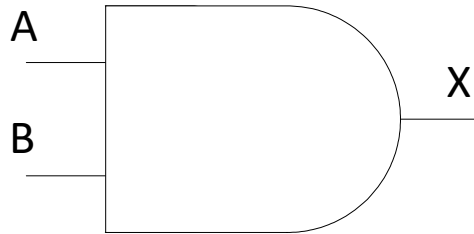
Organização Básica de Computadores

FACENS
Sorocaba/2020

RESUMO PORTAS LÓGICAS

Lógica booleana:

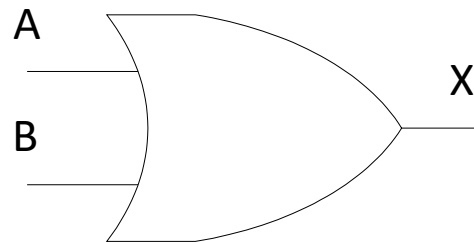
AND (E)



$$X = A.B \text{ ou } X = AB$$

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

OR (OU)

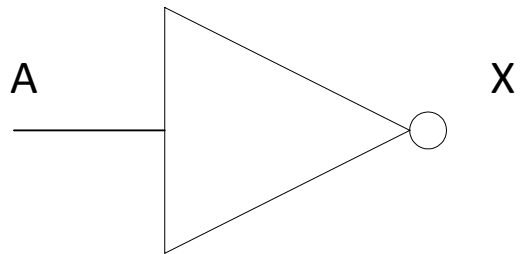


$$X = A+B$$

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

Lógica booleana:

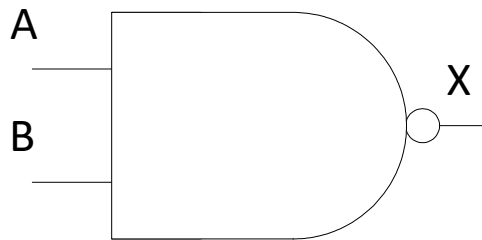
NOT (Não)



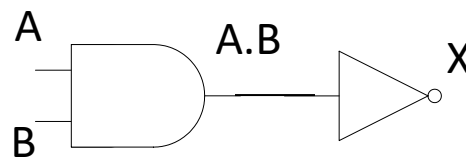
$$X = \overline{A}$$

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

NAND (Não-E)



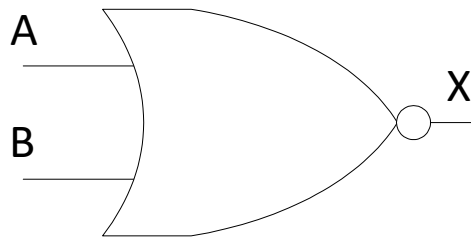
$$X = \overline{A.B} \quad \text{ou} \quad X = \overline{AB}$$



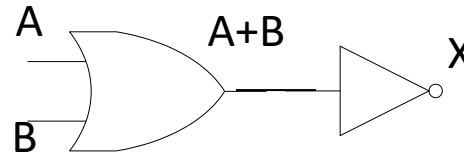
Entrada		Saída
A	B	$X = \overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

Lógica booleana:

NOR (Não-OU)

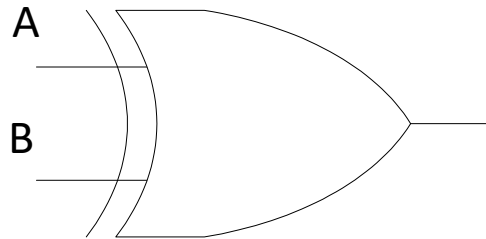


$$X = \overline{A+B}$$



Entrada		Saída
A	B	$X = \overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

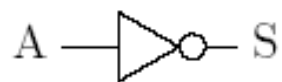
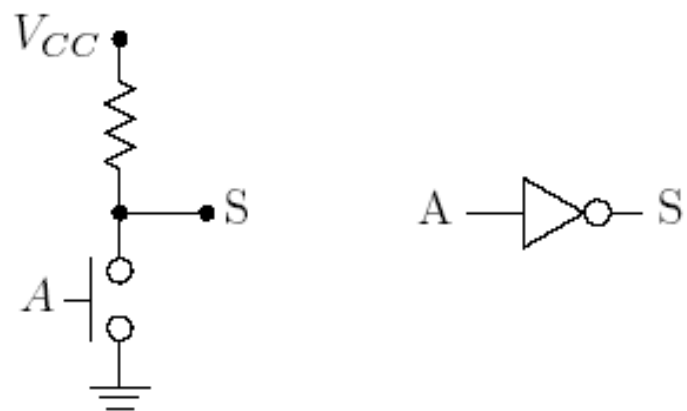
XOR (OU Exclusivo, ou seja só OU)



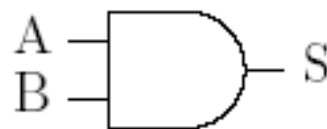
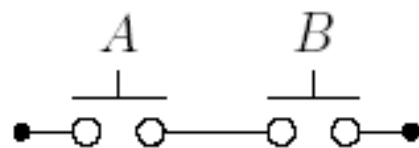
$$X = A \oplus B$$

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

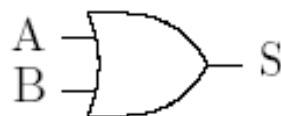
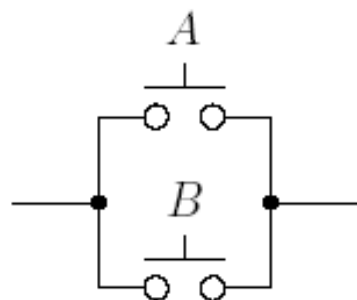
Analogia de porta lógica com lógica de contato



$$S = \text{Not } A$$



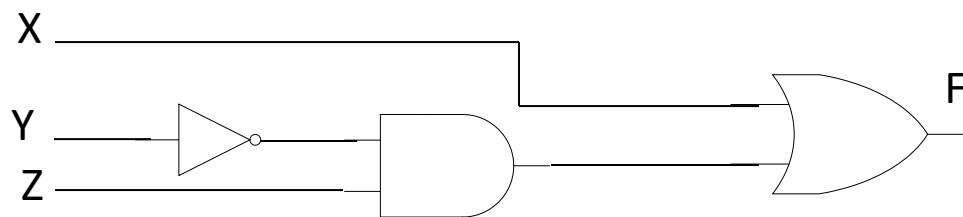
$$S = A \cdot B$$



$$S = A + B$$

Dada a expressão booleana crie o circuito e construa a tabela verdade.

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



Entrada			Saída
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

A partir de portas lógicas simples podemos montar circuitos mais complexos como: memórias, contadores etc...

Uma aplicação de portas lógicas é criar unidades de memória Para armazenar um bit ou uma sequência dos mesmo.
Esse circuito é chamado de Flip-Flop.

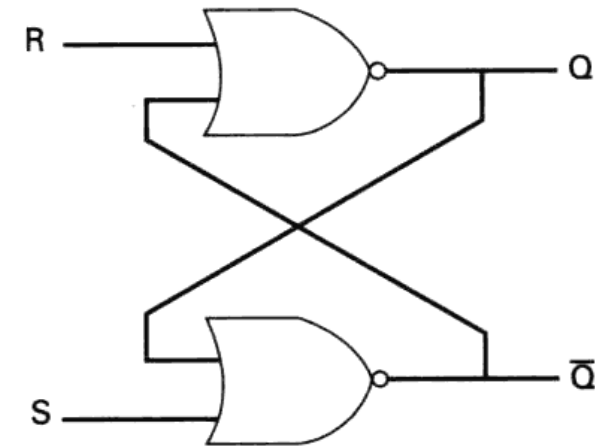
Um conjunto de flip-flops pode-se montar um registrador

Flip-Flop do tipo R-S

Duas entradas: S (set) e R (reset)

Duas saídas: Q e \bar{Q}

Feito a partir de duas portas NOR

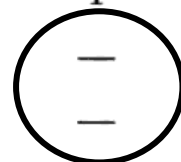


(a) Tabela característica

Entradas correntes	Estado corrente	Próximo estado
SR	Q_n	Q_{n+1}
00	0	0
00	1	1
01	0	0
01	1	0
10	0	1
10	1	1
11	0	—
11	1	—

(b) Tabela característica simplificada

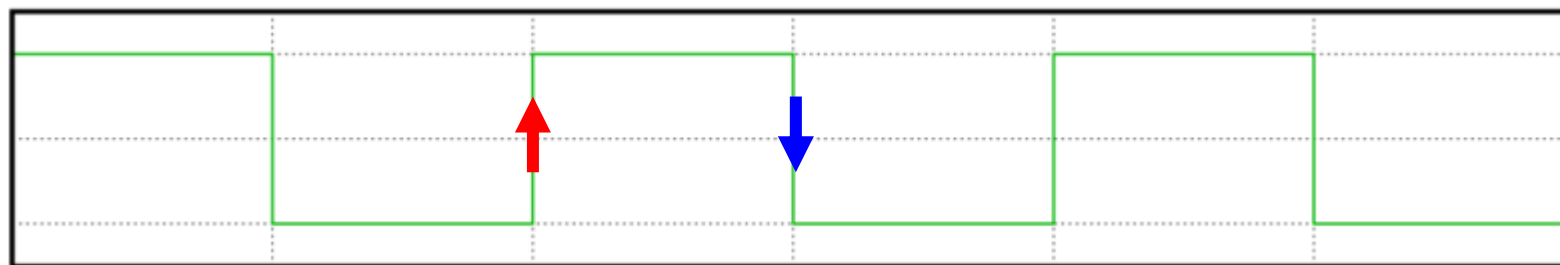
S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	—



Estado não permitido



O Flip-Flop indicado anteriormente não é sincronizado, ou seja não tem uma base certa de tempo para mudança de estado. Para resolver isso devemos colocar um sinal de clock. O clock é um sinal periódico em uma base de tempo.



Período ou base de tempo

Forma de ativação

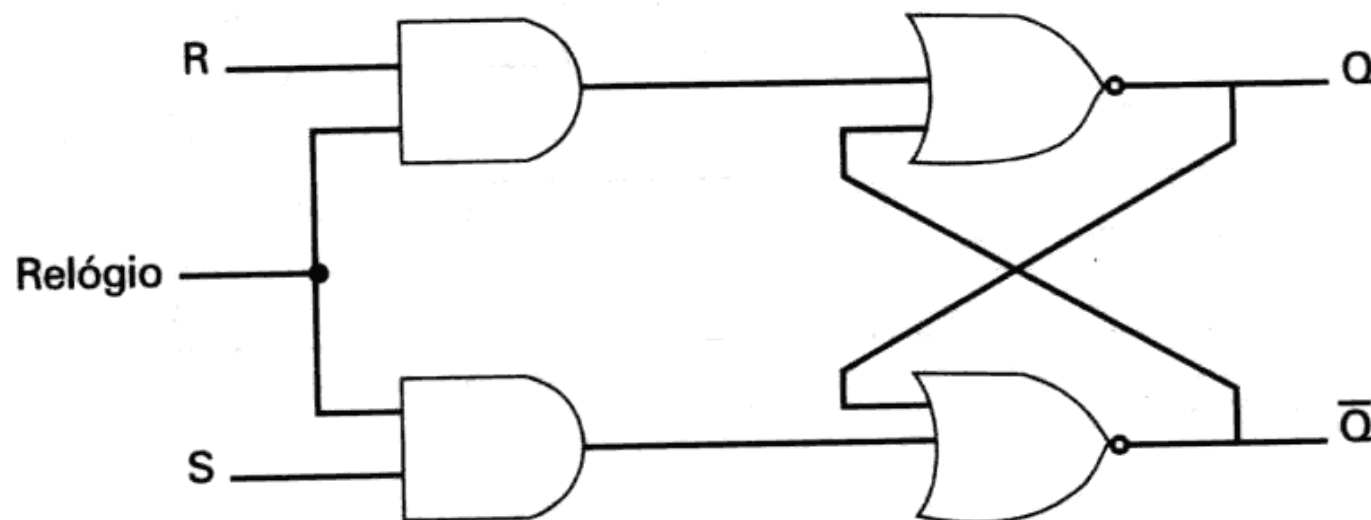


Borda de subida (caso geral)



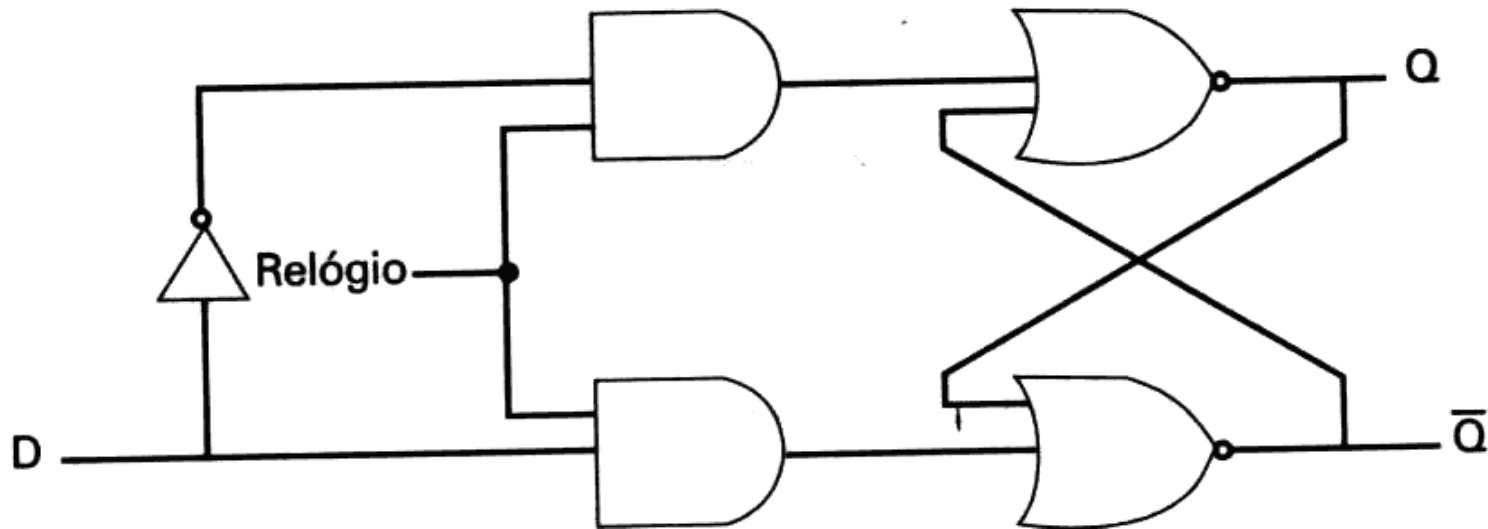
Borda de descida

- Como a saída de um flip-flop S-R muda após um atraso, e como os eventos de um computador são sincronizados, é necessário as entradas S e R somente sejam habilitadas quando ocorre um pulso de relógio(Clock).
- Gerar saída em paralelo de diversos flip-flops sem atraso de chaveamento(ou seja sincronizados).

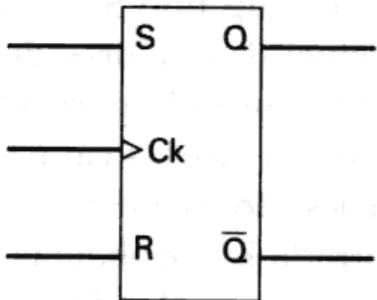
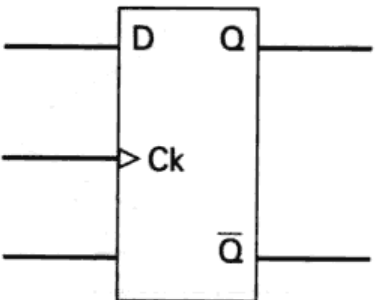


Flip-Flop D – Que armazena um bit

D	Q_{n+1}
0	0
1	1



Alimento pelo pino D e executo um pulso no Clock neste periodo.
O mesmo será copiado para o Q

Nome	Símbolo gráfico	Tabela característica															
S-R		<table><tr><th>S</th><th>R</th><th>Q_{n+1}</th></tr><tr><td>0</td><td>0</td><td>Q_n</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>—</td></tr></table>	S	R	Q_{n+1}	0	0	Q_n	0	1	0	1	0	1	1	1	—
S	R	Q_{n+1}															
0	0	Q_n															
0	1	0															
1	0	1															
1	1	—															
D		<table><tr><th>D</th><th>Q_{n+1}</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	D	Q_{n+1}	0	0	1	1									
D	Q_{n+1}																
0	0																
1	1																

Bloco Lógico que representar os flip-flops estudados até agora

Propriedades algébricas

Propriedade Comutativa:	$AB = BA$ $A + B = B + A$
Propriedade Associativa:	$A(BC) = (AB)C$ $A + (B + C) = (A + B) + C$
Propriedade Distributiva:	$A(B + C) = AB + AC$

Mapa de Karnaugh

O Mapa de Karnaugh é um diagrama utilizado na minimização de funções booleanas.

O método utiliza a tabela verdade de uma função booleana como base para as simplificações. Um mapa de Karnaugh é uma ajuda excelente para simplificação de funções de até 6 variáveis. Para funções de mais de 6 variáveis a simplificação é mais complexa pois torna-se uma tarefa árdua identificar as células adjacentes no mapa. Para funções de mais de 6 variáveis devem ser utilizadas soluções algorítmicas computacionais.

Mapa de Karnaugh

Exemplo: encontre a expressão lógica para a tabela abaixo

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

A partir da tabela
verdade monto essa
nova tabela

	AB	A \bar{B}	$\bar{A}B$	$\bar{A}\bar{B}$
C	0	0	0	0
\bar{C}	1	0	1	1

Mapa de Karnaugh

Sempre selecione os elementos em 1 nas quantidades, 2^N ou seja 1,2,4,8 ...

	AB	A \bar{B}	$\bar{A}B$	$\bar{A}\bar{B}$
C	0	0	0	0
\bar{C}	1	0	1	1

Analise o que não variou na seleção

Porção 1:
 $\bar{A} . \bar{C}$

Porção 2:
 $A . B . \bar{C}$

Solução 1:

$A . B . \bar{C} + \bar{A} . \bar{C}$ Simplificando: $\bar{C} . (A.B + \bar{A})$

Mapa de Karnaugh

Solução 2:

	AB	A \bar{B}	$\bar{A}B$	$\bar{A}\bar{B}$
C	0	0	0	0
\bar{C}	1	0	1	1

Porção 1:

$$\bar{A} \cdot \bar{C}$$

Porção 2:

$$B \cdot \bar{C}$$

Nova Solução:

$$B \cdot \bar{C} + \bar{A} \cdot \bar{C} \text{ Simplificando: } \bar{C} \cdot (B + \bar{A})$$

Mapa para duas variáveis

$A \backslash B$	0	1
0	0	1
1	2	3

Mapa para três variáveis

C / AB	00	01	11	10
0	0	1	3	2
1	4	5	7	6

Mapa para quatro variáveis

AB / CD	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

Na montagem do mapa só pode variar um elemento por vez

Exercícios:

1) Construa o circuito lógico e tabela verdade:

a) $F = A + \overline{B} \cdot C \oplus D$

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

2) Determine o circuito para as tabelas verdades abaixo:

a)

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

b)

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
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	0

c)

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Exercícios:

3) Desafio: Abaixo encontra-se o flip-flop JK, monte a tabela verdade para as entradas J e K com a saída Q

