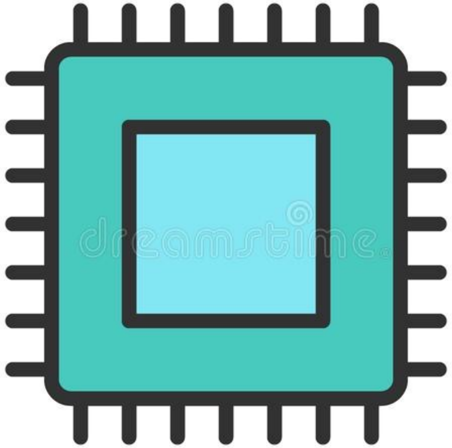
O mundo binário



TÉCNICO LISBOA

Bases de numeração

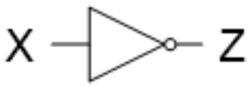
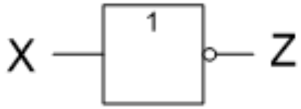
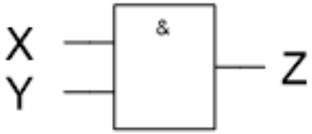
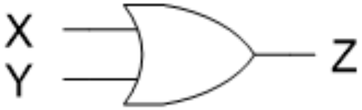
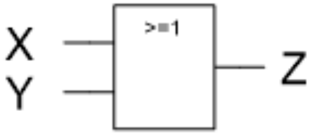
- Os seres humanos fazem contas em base 10. Porquê?
  - Resposta musical: https://www.youtube.com/watch?v=aEnfy9qfdaU
- Os circuitos eletrónicos funcionam em base 2
  - Cada símbolo na base 2 designa-se bit, 0 e 1 (**bi**nary digi**t**)
  - Mais **simples** de implementar (“com sinal elétrico = 1”, “sem sinal elétrico = 0”)
  - Permitem executar operações usando **álgebra booleana**
- Qualquer número pode ser representado em qualquer base, usando um ou mais símbolos
  - Exemplo: decimal -> binário -> hexadecimal
  - $3072_{10} = 110000000000_2 = C00_{16}$

Portas lógicas

Portas lógicas

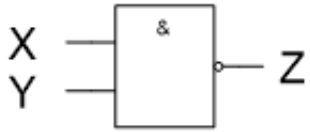
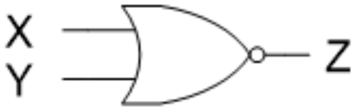
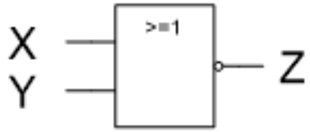
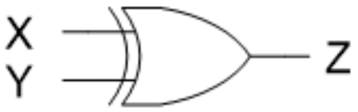
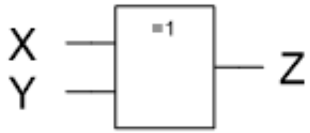
- São os circuitos que permitem executar as **operações básicas de álgebra booleana**, como o NOT, AND ou OR.

NOT, AND, OR

Porta	Símbolos		Função	Tabela de verdade
NOT			$Z = \overline{X}$	
AND			$Z = X \cdot Y$	
OR			$Z = X + Y$	

Uma **tabela de verdade** mostra, de forma tabelar, quais os valores de saída para cada uma das combinações dos sinais à entrada.

NAND, NOR, XOR

PORTA	SÍMBOLOS		FUNÇÃO	TABELA DE VERDADE
NAND			$Z = \overline{X \cdot Y}$	
NOR			$Z = \overline{X + Y}$	
XOR			$Z = X \oplus Y$	

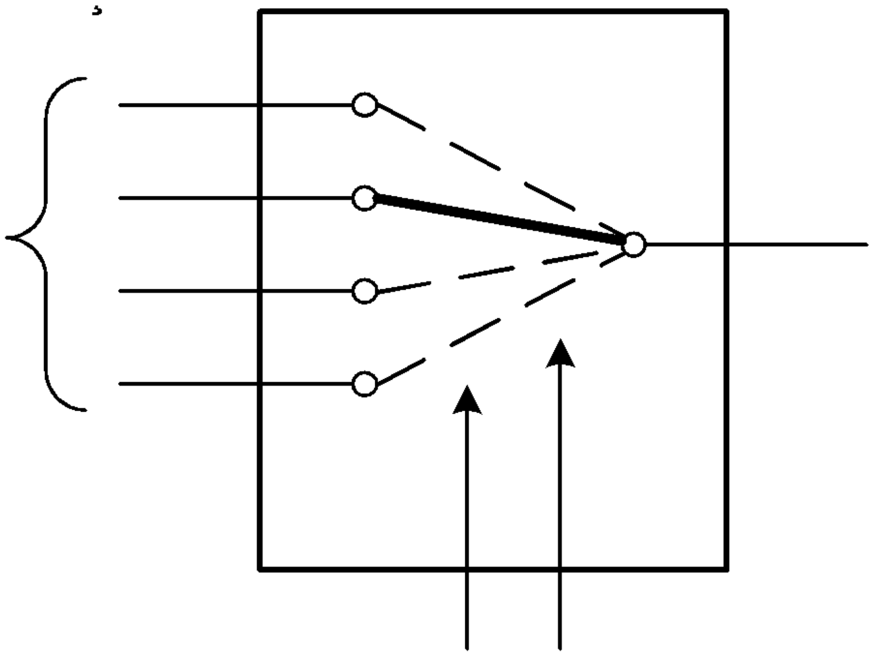
Circuitos combinatórios

Circuito combinatório

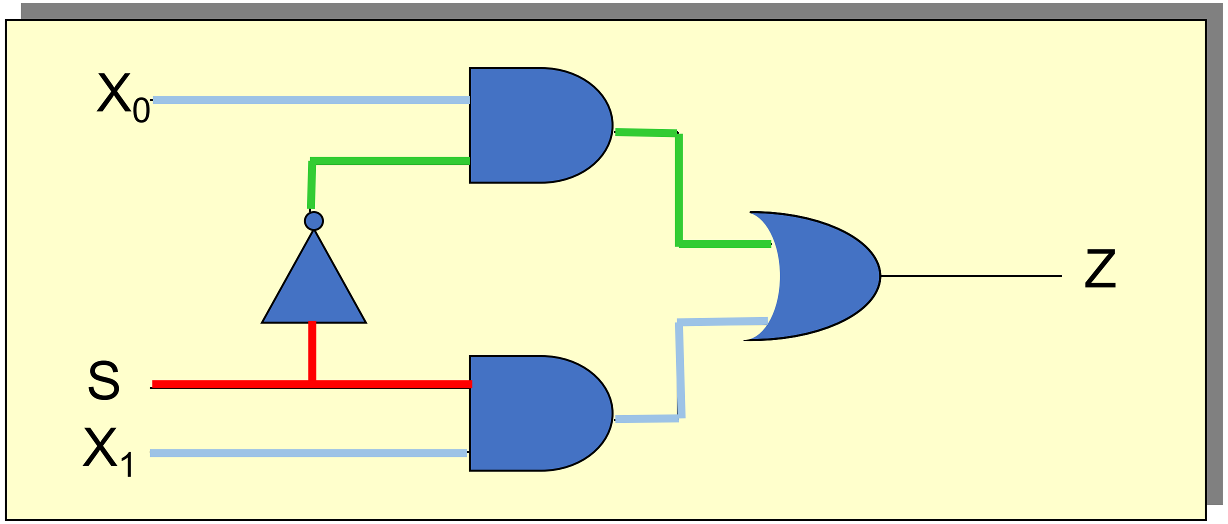
- Circuito digital **sem realimentação**
- Nenhuma entrada de uma porta lógica **depende** de nenhuma saída
- Para a mesma combinação de valores das variáveis de entrada os valores de saída serão sempre os mesmos, **independentemente da evolução passada** dos sinais de entrada
- Por outras palavras, o circuito **não tem estado interno**

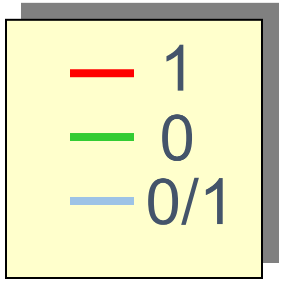
Multiplexer

- Circuito que permite **escolher entre uma de várias entradas** e trasnportar o seu valor para uma saída



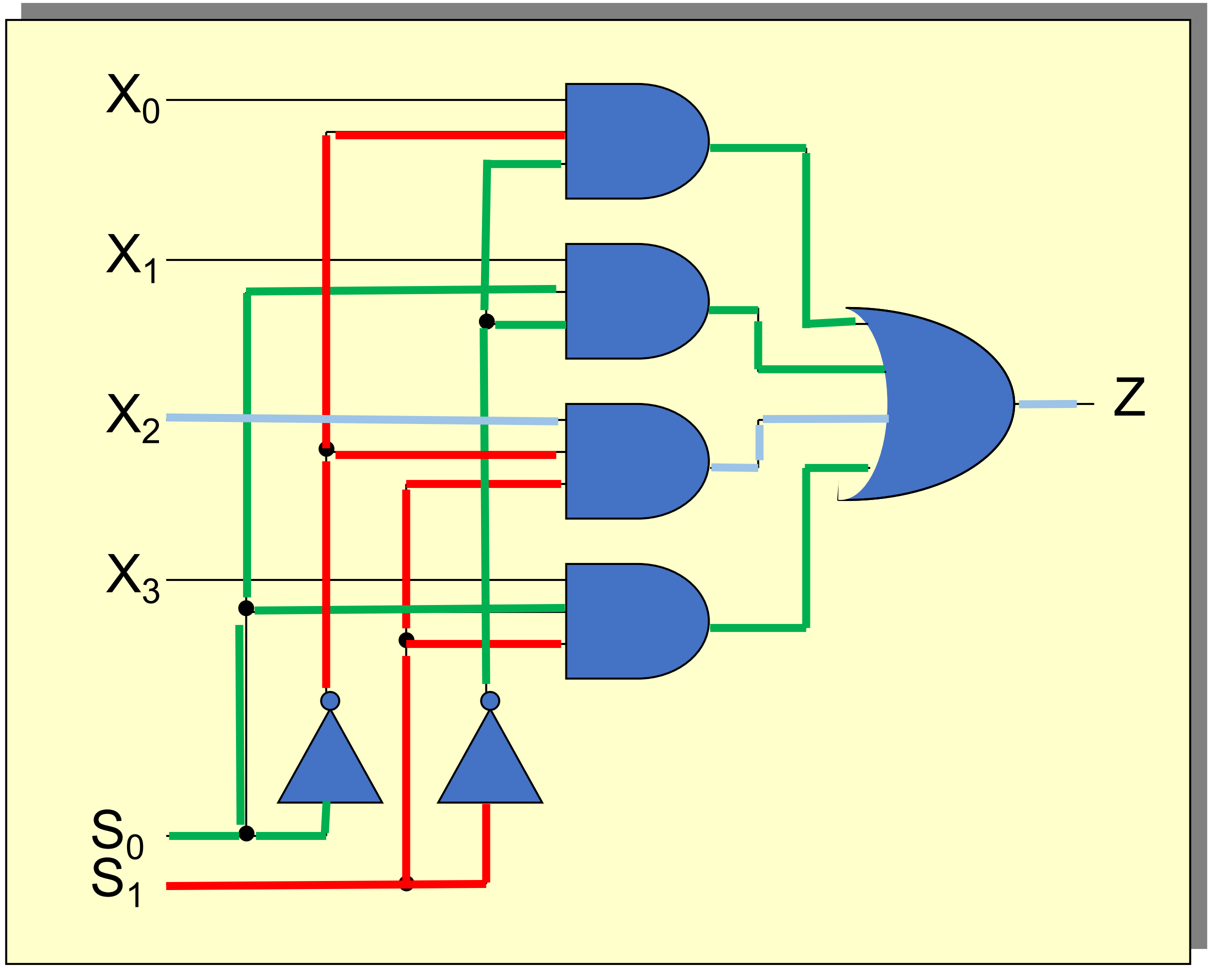
Multiplexer 2-para-1



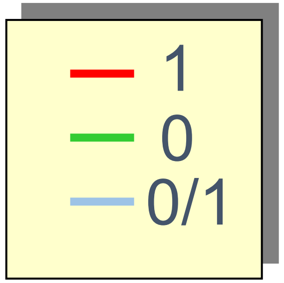


S	Z
0	X_0
1	X_1

Multiplexer 4-para-1



S_1	S_0	Z
0	0	X_0
0	1	X_1
1	0	X_2
1	1	X_3

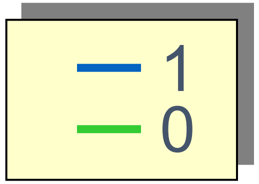


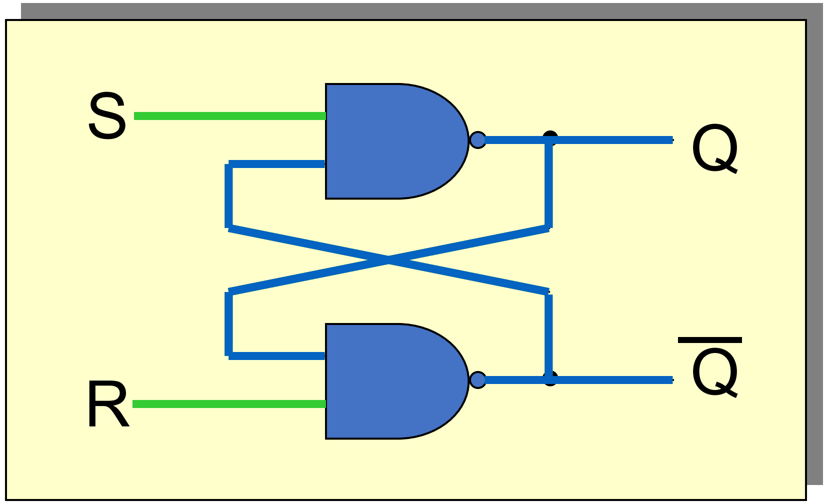
Circuitos sequenciais

Circuito sequencial

- Circuito digital **com realimentação**
- O valor das saídas depende não apenas das entradas mas **também dos valores anteriores** das saídas
- Por outras palavras, o circuito **tem estado** (uma combinação do valor das saídas do circuito)

Trinco *(latch)* SR



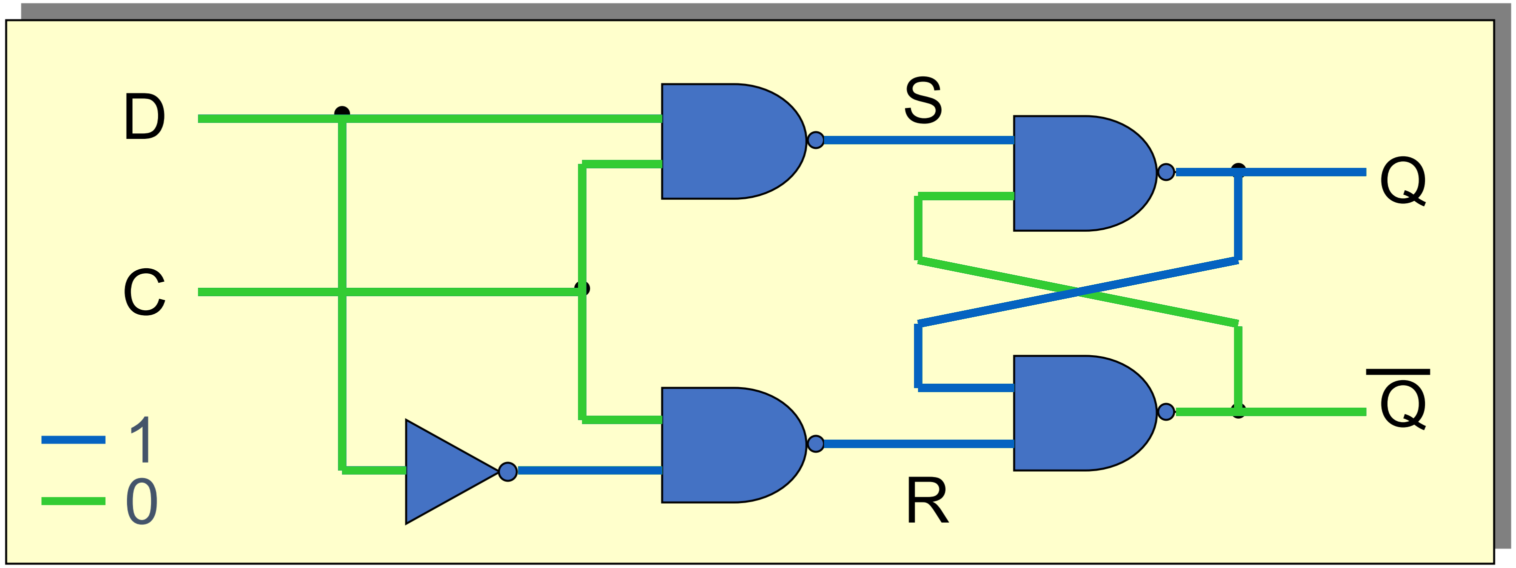


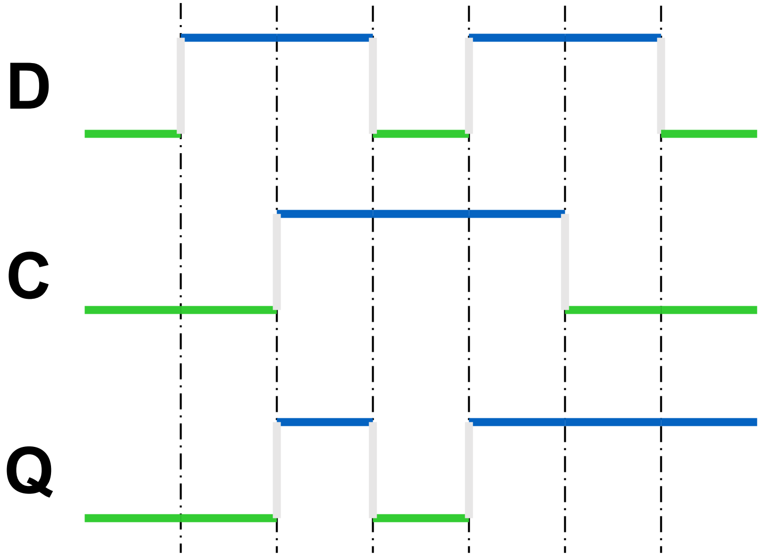
S	R	Q	$\overline{Q}$	
0	1	1	0	Força Q = 1 (set)
1	1	1	0	Mantém estado
1	0	0	1	Força Q = 0 (reset)
1	1	0	1	Mantém estado
0	0	1	1	Inválido

Possíveis limitações: necessita de 2 sinais para poder mudar de estado; entradas inválidas quebram semântica de negação entre as saídas.

TÉCNICO LISBOA

Trinco D

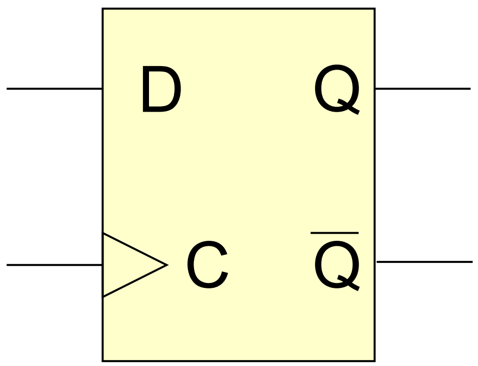




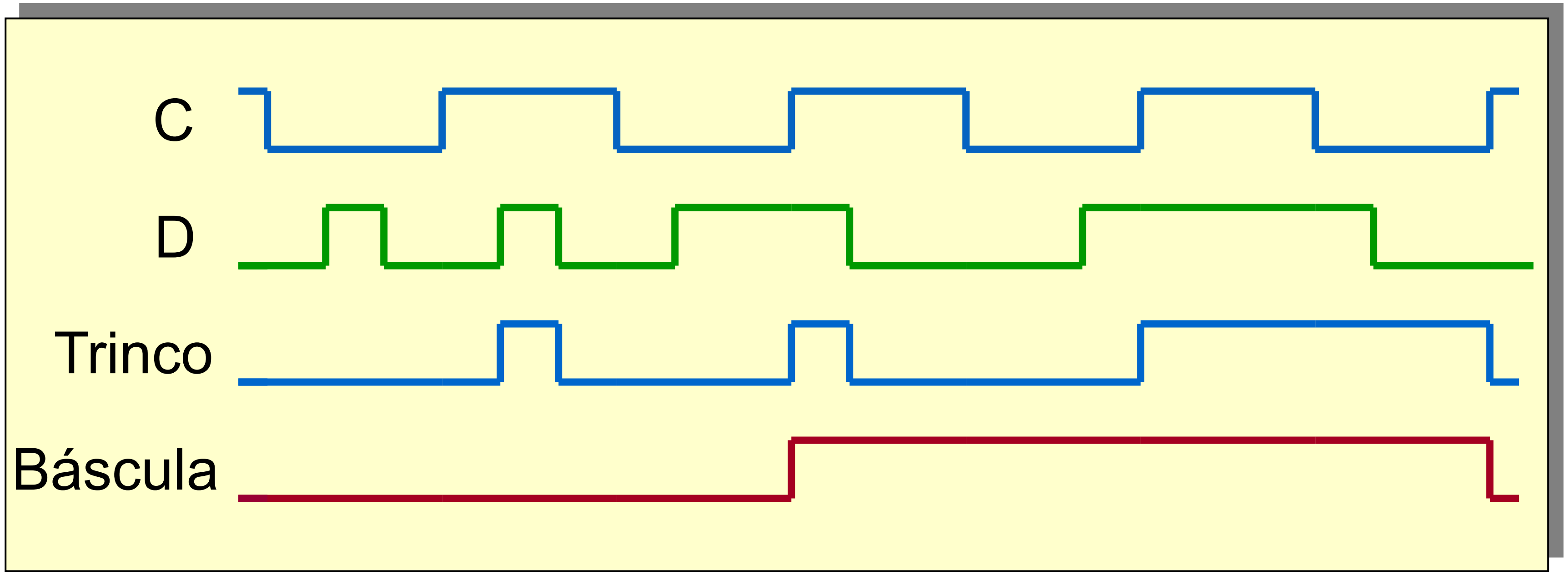
C	Q
0	Mantém estado
1	D (transparente)

Báscula (*flip-flop*) D (ativa no flanco)

- Memoriza o valor de D quando C **transita** de 0 para 1 (flanco ascendente)
- A principal diferença relativamente ao trinco é que a báscula mantém o valor D mesmo que este mude, e só memoriza o valor quando o C volta a transitar
  - Portanto é na transição do sinal C que se memoriza o valor do D
  - Pensem no C como o relógio ("clock") do sistema

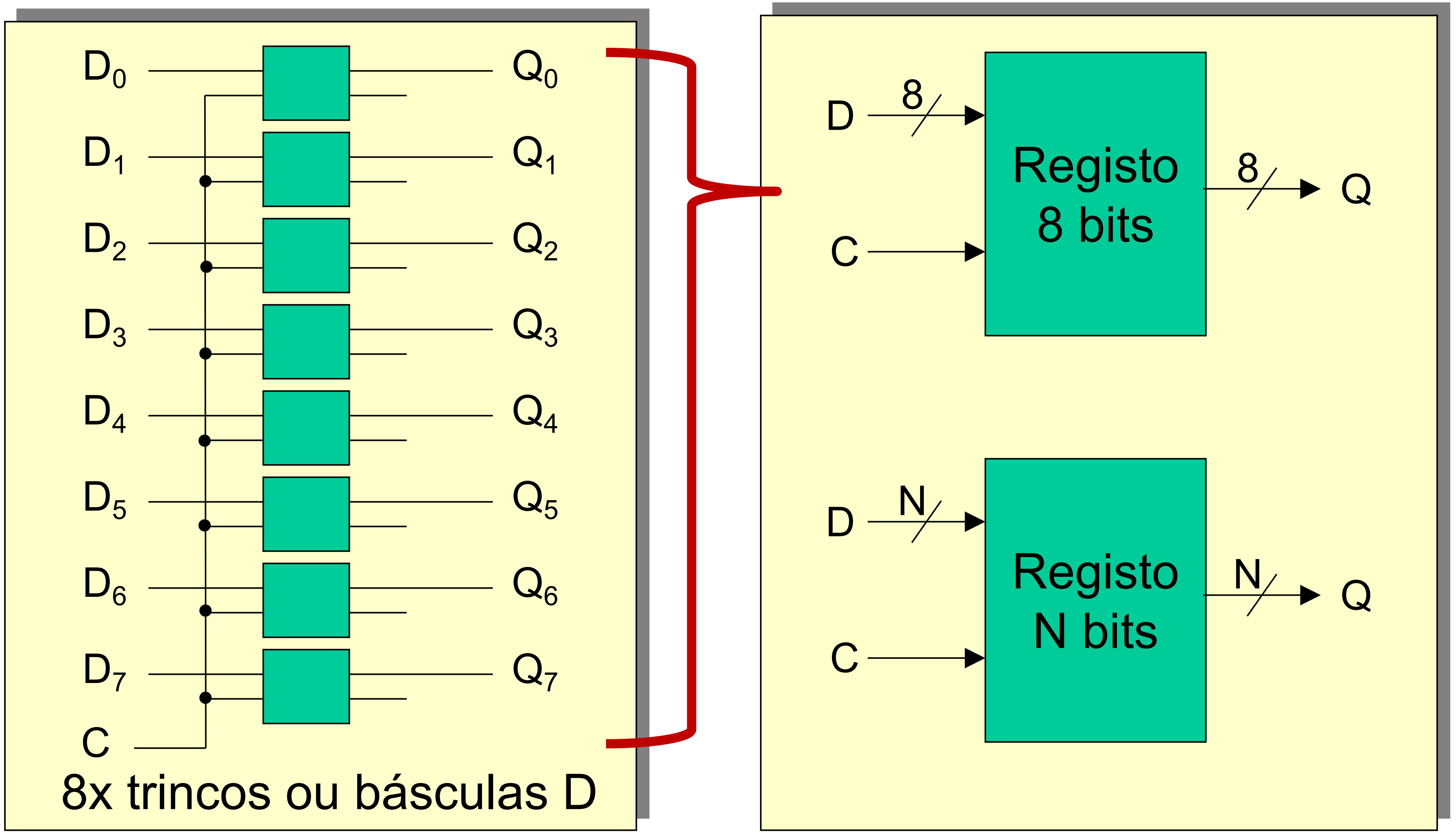


Trincos e básculas D (flanco ascendente)



TÉCNICO LISBOA

Registos



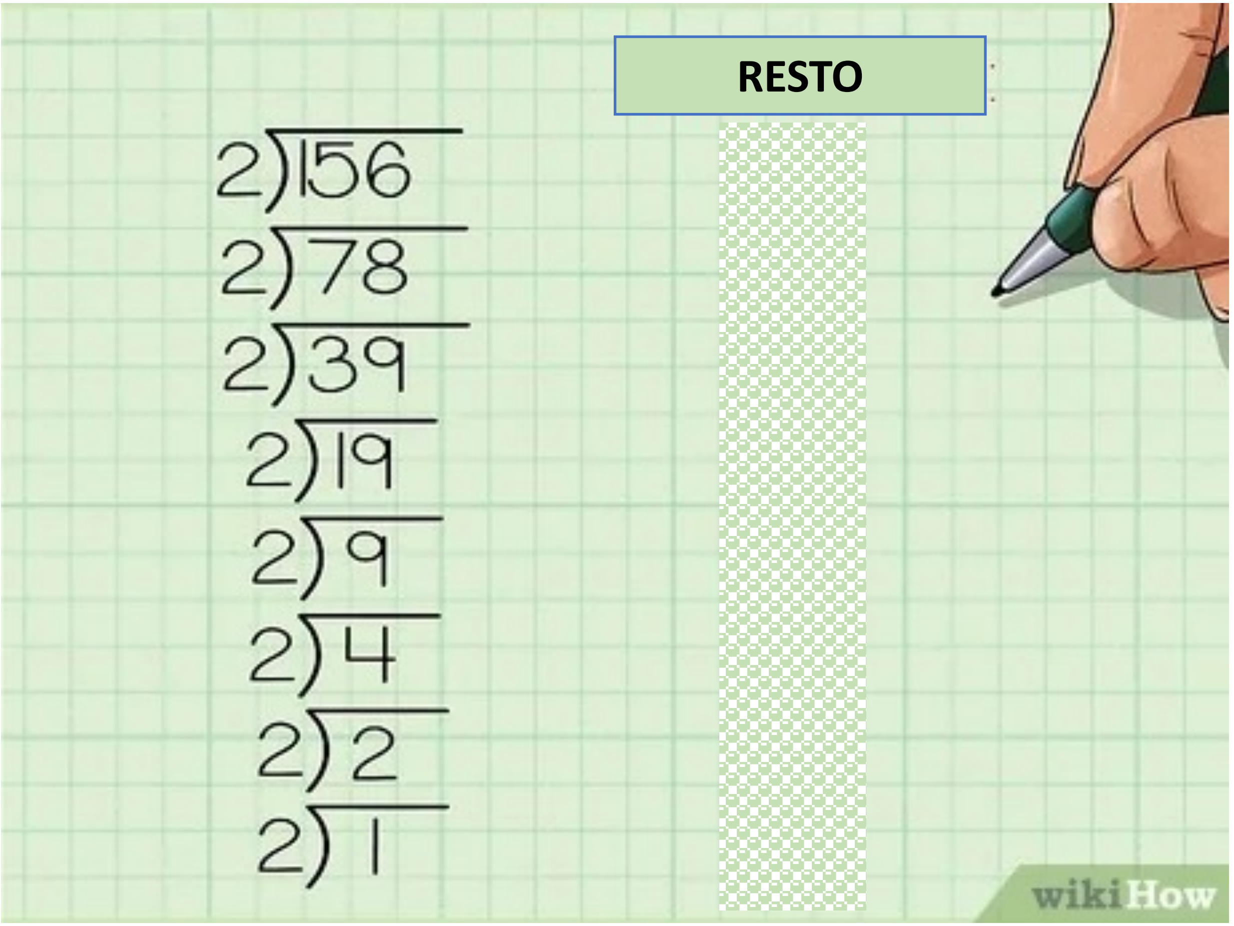
D0
D1
D2
D3
D4
D5
D6
D7
C
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
8x trincos ou básculas D
D
8
C
Registo
8 bits
8
Q
D
N
C
Registo
N bits
N
Q

Representação de números

Representação de números

Decimal	Binário	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Decimal -> Binário



$156_{10} = 10011100_2$

Binário –> Decimal

$$1011_2$$

$$\downarrow$$

$$1 \text{x} 2^3 + 0 \text{x} 2^2 + 1 \text{x} 2^1 + 1 \text{x} 2^0 = 11_{10}$$

Binário -> Hexadecimal

0110 1101 0100 1101 0011 0001 0101 0011

6 D 4 D 3 1 5 3_H

Hexadecimal -> Binário

$\mathbf{A\,6\,7\,F}_{\mathbf{H}}$

1010 0110 0111 1111

TÉCNICO LISBOA

Gama de números

- Com N bits consegue-se representar os números inteiros:

$$[0, 2^N - 1]$$

ou

$$[-2^{N-1}, 2^{N-1} - 1]$$

- Exemplo com 8 bits:

$$[0, 255]$$

ou

$$[-128, +127]$$

Notação complemento para 2

Sem sinal		Com sinal	
1111 1111	255	0111 1111	+127
1111 1110	254	0111 1110	+126
. . .	. . .	. . .	. . .
1000 0010	130	0000 0010	2
1000 0001	129	0000 0001	1
1000 0000	128	0000 0000	0
0111 1111	127	1111 1111	-1
0111 1110	126	1111 1110	-2
. . .	. . .	. . .	. . .
0000 0001	1	1000 0001	-127
0000 0000	0	1000 0000	-128

Obter simétrico em complemento para 2

transporte	0 0 0 0 0 0 1 1	
	0101 1100	número ($5C_H$)
	1010 0011	complemento para 1
+	1	
	1010 0100	complemento para 2 ($A4_H = -5C_H$)

TÉCNICO LISBOA

Extensão de sinal

bits	+2	-2
4	0010	1110
8	0000 0010	1111 1110
16	0000 0000 0000 0010	1111 1111 1111 1110

Operações aritméticas

TÉCNICO LISBOA

Soma

	0 1 0 0 1 1 1 0	transporte
binário	0 1 1 0 1 0 1 1	operando A
	+ 0 1 0 0 0 1 1 0	operando B
	1 0 1 1 0 0 0 1	resultado

	0 1	transporte
hexadecimal	6 B	operando A
	+ 4 6	operando B
	B 1	resultado

TÉCNICO LISBOA

Soma e subtração

A – B ≡ A + (-B)

Basta ter o simétrico de B em complemento para 2.

Exemplo: 5CH - 5CH ≡ 5CH + (-5CH) = 5CH + A4H

$$
\begin{array}{r}
0101\ 1100 \\
-\ 0101\ 1100 \\
\hline
0000\ 0000
\end{array}
\qquad
\begin{array}{r}
{\scriptstyle 1\ \ 1111\ 1\,0\,0\ } \\
0101\ 1100 \\
+\ 1010\ 0100 \\
\hline
1\ |\ 0000\ 0000
\end{array}
$$

Excesso (*overflow*)

	0 1 0 1 1 1 1 1	transporte
	0 1 0 1 1 1 0 1	operando A
+	0 1 0 1 0 1 1 1	operando B
	1 0 1 1 0 1 0 0	soma

Resultado negativo!

- **Solução:** aumentar número de bits
- **Deteção:**
  - se o sinal dos operandos for diferente, nunca ocorre
  - se for igual, e o resultado for diferente: **overflow!**

Referência rápida

Decimal	Binário	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

N	2^N (decimal)	Kibi (1024)	2^N (hexadecimal)
0	1		1
1	2		2
2	4		4
3	8		8
4	16		10H
5	32		20H
6	64		40H
7	128		80H
8	256		100H
9	512		200H
10	1024	1 Ki	400H
11	2048	2 Ki	800H
12	4096	4 Ki	1000H
13	8192	8 Ki	2000H
14	16384	16 Ki	4000H
15	32768	32 Ki	8000H
16	65536	64 Ki	10000H

- Fator multiplicador : K = 1000, Ki = 1024

Símbolo	Lê-se	Valor decimal
K	Kilo	10^3
M	Mega	10^6
G	Giga	10^9
T	Tera	10^{12}

Símbolo	Lê-se	Equivale a	Valor binário	Valor decimal
Ki	Kibi	1024	2^{10}	1 024
Mi	Mebi	1024 Ki	2^{20}	1 048 576
Gi	Gibi	1024 Mi	2^{30}	1 073 741 824
Ti	Tebi	1024 Gi	2^{40}	1 099 511 627 776

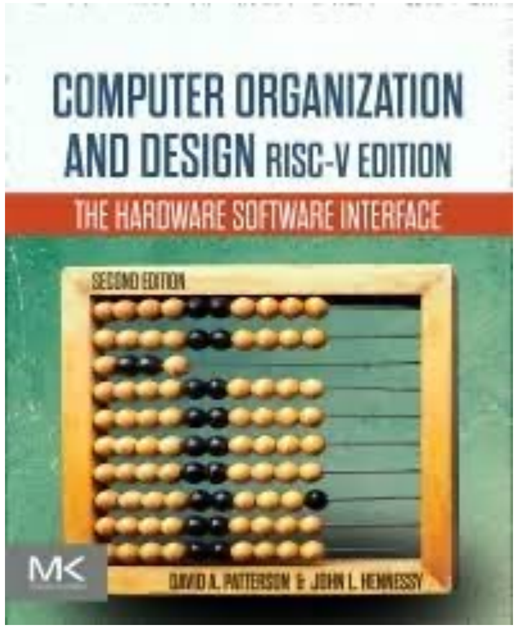
Bibliografia

Recomendada

- **[Delgado&Ribeiro_2014]**
  - **Secções 2.2.1, 2.2.3, 2.5.2, 2.6.1, 2.6.2, 2.7, 2.8.1-2.8.3**

Secundária/adicional

- [Patterson&Hennessy_2021]
  - Apêndice 1



TÉCNICO LISBOA