



TÉCNICO LISBOA

SISTEMAS DIGITAIS (SD)

MEEC

Acetatos das Aulas Teóricas

Versão 4.0 - Português

Aula Nº 04:

Título: Elementos de Tecnologia e Funções Lógicas

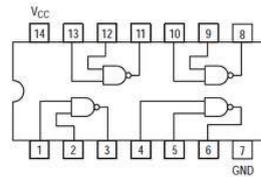
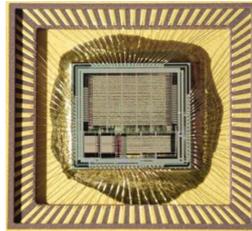
Sumário: Elementos de tecnologia (circuitos integrados, famílias lógicas); Funções lógicas (circuitos com portas NAND, circuitos com portas NOR).

2015/2016

Nuno.Roma@tecnico.ulisboa.pt

Sistemas Digitais (SD)

Elementos de Tecnologia Funções Lógicas



Aula Anterior

- **Na aula anterior:**
 - ▶ Álgebra de Boole:
 - Operações básicas
 - Propriedades
 - Portas lógicas
 - ▶ Leis de DeMorgan:
 - Simplificação algébrica

SEMANA	TEÓRICA 1	TEÓRICA 2	PROBLEMAS/LABORATÓRIO
14/Set a 19/Set	Introdução	Sistemas de Numeração e Códigos	
21/Set a 26/Set	Álgebra de Boole	Elementos de Tecnologia	P0
28/Set a 3/Out	Funções Lógicas	Minimização de Funções Booleanas (I)	L0
5/Out a 10/Out	Minimização de Funções Booleanas (II)	Def. Circuito Combinatório; Análise Temporal	P1
12/Out a 17/Out	Circuitos Combinatórios (I) – Codif., MUXs, etc.	Circuitos Combinatórios (II) – Som., Comp., etc.	L1
19/Out a 24/Out	Circuitos Combinatórios (III) - ALUs	Circuitos Sequenciais: Latches	P2
26/Out a 31/Out	Circuitos Sequenciais: Flip-Flops	Ling. de Descrição e Simulação de HW (ferramentas disponíveis no laboratório)	L2
2/Nov a 7/Nov	Caracterização Temporal	Registos	P3
9/Nov a 14/Nov	Revisões Teste 1	Contadores	L3
16/Nov a 21/Nov	Síntese de Circuitos Sequenciais: Definições	Síntese de Circuitos Sequenciais: Minimização do número de estados	P4
23/Nov a 28/Nov	Síntese de Circuitos Sequenciais: Síntese com Contadores	Memórias	L4
30/Nov a 5/Dez	Máq. Estado Microprogramadas: Circuito de Dados e Circuito de Controlo	Máq. Estado Microprogramadas: Endereçamento Explícito/Implícito	P5
7/Dez a 12/Dez	Circuitos de Controlo, Transferência e Processamento de Dados de um Processador	Lógica Programável	L5
14/Dez a 18/Dez	P6	P6	L6

■ Tema da aula de hoje:

- ▶ Elementos de tecnologia:
 - Circuitos integrados
 - Famílias lógicas
- ▶ Funções lógicas:
 - Circuitos com portas NAND
 - Circuitos com portas NOR

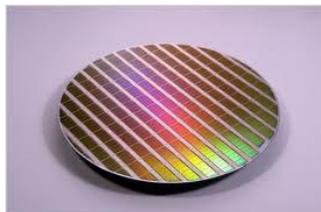
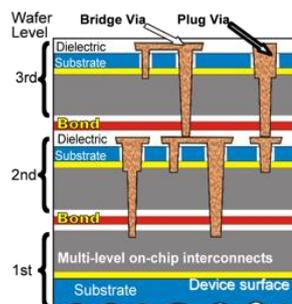
□ Bibliografia:

- **M. Mano, C. Kime:** Secções 6.1, 2.1 a 2.2, 2.8 a 2.9
- **G. Arroz, J. Monteiro, A. Oliveira:** Secções 3.1, 3.2 e 2.1

ELEMENTOS DE TECNOLOGIA

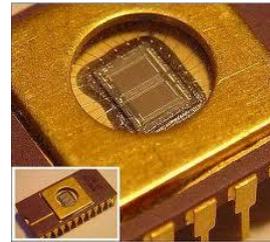
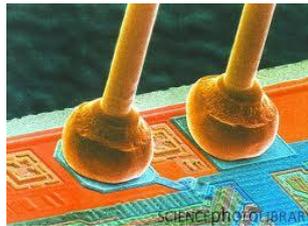
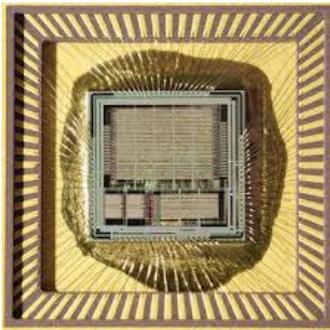
Circuitos Integrados

- Os circuitos digitais são realizados com circuitos integrados.
- Um circuito integrado é um cristal semiconductor, habitualmente de silício, que contém os componentes electrónicos que formam as portas lógicas. É habitualmente designado por chip.



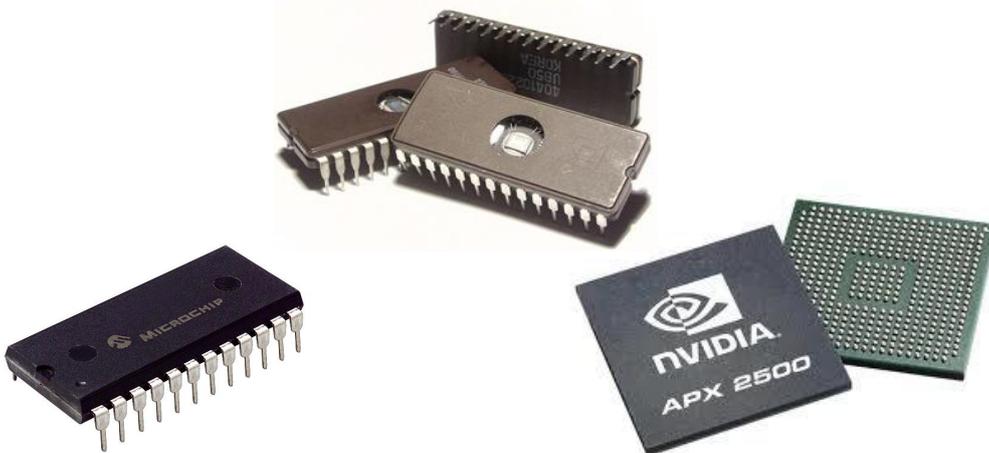
Circuitos Integrados

- O chip é montado dentro de um empacotamento cerâmico ou plástico e são construídas ligações do chip para os pinos externos do circuito integrado.



Circuitos Integrados

- O número de pinos pode variar entre 8 (para os empacotamentos mais pequenos) e várias centenas.



- Os circuitos integrados podem ser classificados segundo o nº de portas lógicas que utilizam:

SSI (Small-scale integration)	MSI (Medium-scale integration)	LSI (Large-scale integration)	VLSI (Very Large-scale integration)
≤10 Portas	≤100 Portas	≤10.000 Portas	>10.000 portas
Portas básicas (AND, OR, ...)	Funções elementares (ADD, SUB)	Processadores simples	Processadores complexos (ex.: microprocessadores)



- O desenho de circuitos VLSI tipicamente baseia-se em linguagens específicas de descrição de hardware, nomeadamente VHDL e Verilog.

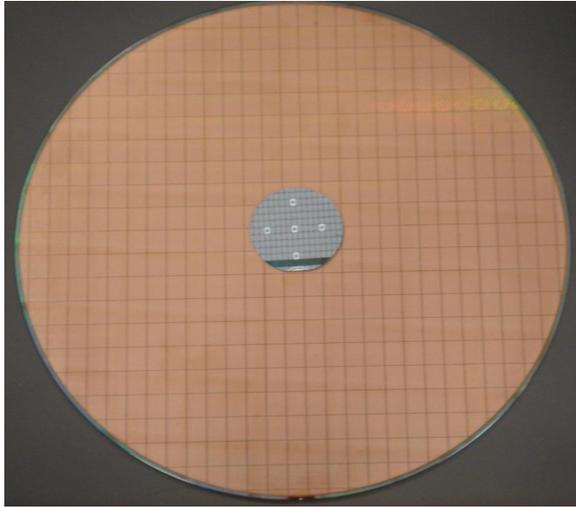
► Exemplo:

```
-- import std_logic from the IEEE library
library IEEE;
use IEEE.std_logic_1164.all;

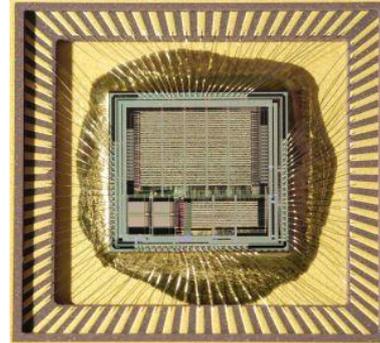
-- this is the entity
entity ANDGATE is
    port (
        I1 : in std_logic;
        I2 : in std_logic;
        O  : out std_logic);
end entity ANDGATE;

-- this is the architecture
architecture RTL of ANDGATE is
begin
    O <= I1 and I2;
end architecture RTL;
```

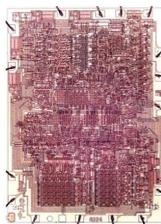
■ Bolachas de silício e interligações:



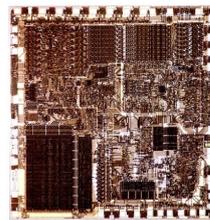
	Intel 4004	Intel Core i7
Diam. wafer	50 mm	300 mm
N. Transistores	$2,3 \times 10^3$	7×10^8
Pistas (fios)	10×10^{-6} m	22×10^{-9} m



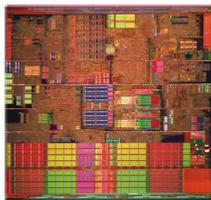
■ Processadores



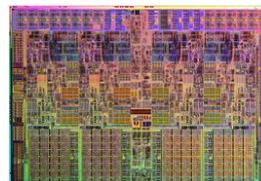
INTEL 4004
Ano: 1971
Freq.: 108 kHz
2.300 Trans.



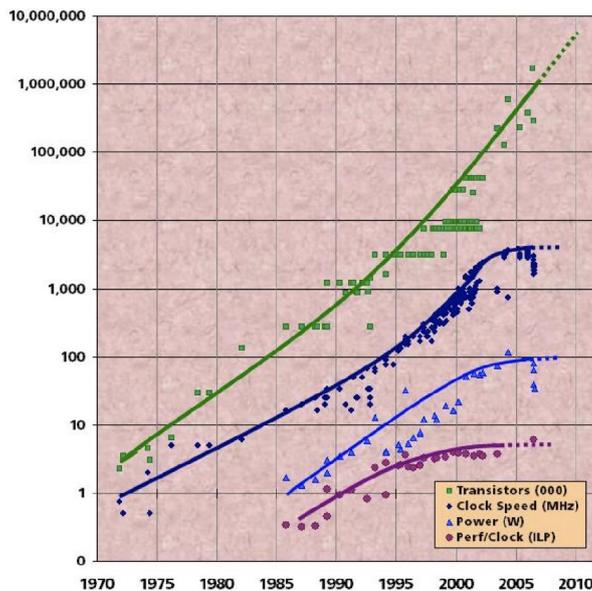
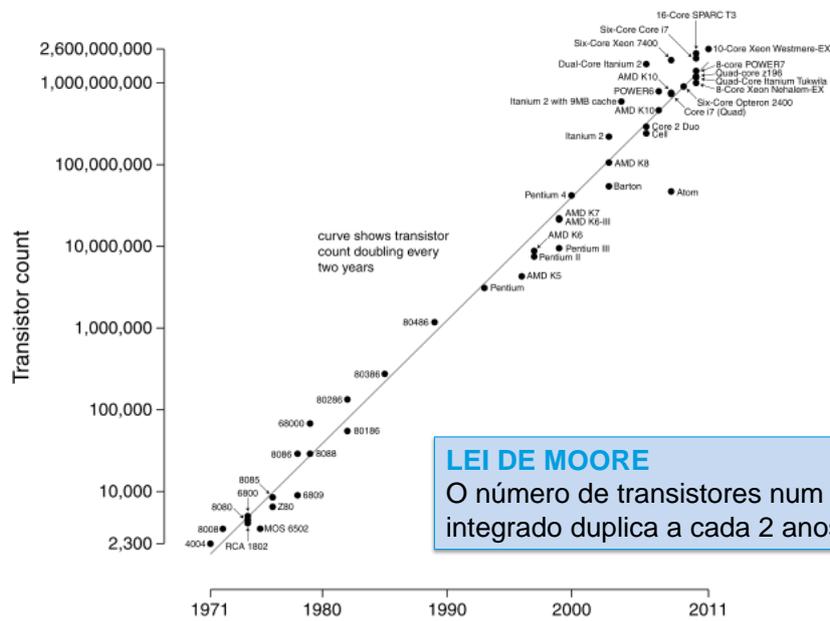
INTEL 8086
Ano: 1978
Freq.: 5 MHz
29.000 Trans.



INTEL Pentium 4
Ano: 2000
Freq.: 1,5 GHz
42.000.000 Trans.



INTEL Core i7 (quad)
Ano: 2008
Freq.: 3 GHz
731.000.000 Trans.



LEI DE MOORE
Durante a última década, limitações tecnológicas têm vindo a impor tectos difíceis de ultrapassar...

■ Tecnologias:

- ▶ Os CIs são também classificados de acordo com a tecnologia em que são fabricados.
- ▶ Os circuitos de uma dada tecnologia agrupam-se em famílias lógicas segundo os circuitos electrónicos que constituem as suas portas básicas.
- ▶ Historicamente, as tecnologias mais importantes são:

Abrev.	Nome completo	Observações
RTL	Resistor-Transistor Logic	Primeiras famílias lógicas - completamente obsoletas.
DTL	Diode-Transistor Logic	
TTL	Transistor-Transistor Logic	Popular e barata. Disponível um grande número de componentes SSI e MSI. Usada durante várias décadas.
ECL	Emitter-Coupled Logic	Usada em alguns circuitos que operam em alta frequência.
MOS (pMOS,nMOS)	Metal-Oxide Semiconductor	Grande capacidade de integração. Suscetíveis a ruído.
CMOS	Complementary Metal-Oxide Semiconductor	Baixo consumo de potência. Grande capacidade de integração. Tecnologia actualmente <u>dominante</u> .
BiCMOS	Bipolar Complementary Metal-Oxide Semiconductor	Combina CMOS com TTL, usada em casos em que só CMOS não garante capacidade de "drive".
GaAs	Gallium-Arsenide	Usada em circuitos que operam em muito alta frequência.

■ Componentes TTL:

▶ Designação:

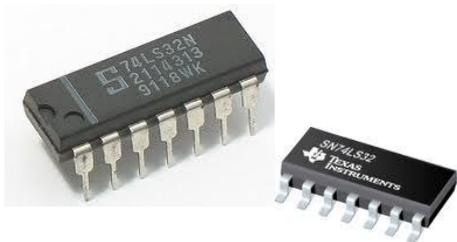
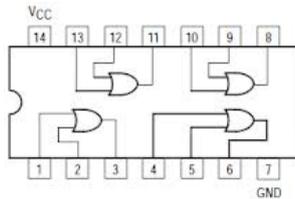
- 54 / 74 - Série 74 = standard
- Série 54 = aplicações militares
- L / LS / S / " " / H – tipos diferentes de transístores e compromissos diferentes de velocidade vs. potência dissipada.
- A família LS é actualmente a mais popular.



		Atraso	Potência Dissipada	Produto Pot.*Atr.
74	Standard	10 ns	10 mW	100 pJ
74S	Schottky	3 ns	19 mW	57 pJ
74L	Low-Power	33 ns	1 mW	33 pJ
74H	High-Power	6 ns	22 mW	132 pJ
74LS	Low-Power Schottky	9,5 ns	2 mW	19 pJ

Componentes TTL:

Exemplos:



Exemplo de Componentes Disponíveis	
Dispositivo	Função
'00	4 NAND2
'02	4 NOR2
'04	6 NOT
'08	4 AND2
'20	2 NAND4
'21	2 AND4
'27	3 NOR3
'30	1 NAND8
'32	4 OR2
'126	4 Buffers Tri-State
'136	4 XOR2

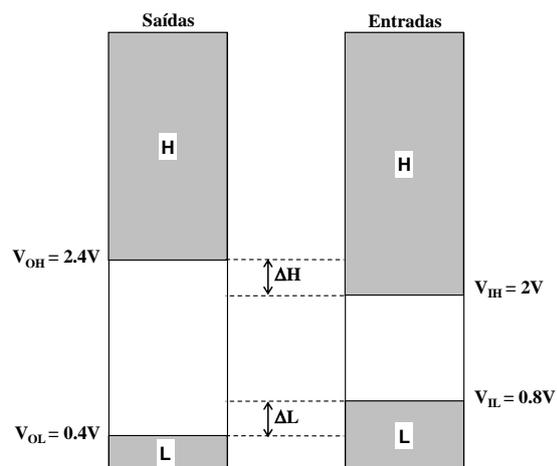
Níveis lógicos:

V_{OH} - Tensão mínima de saída fornecida pela porta lógica, quando a saída se encontra no nível lógico alto (HIGH)

V_{OL} - Tensão máxima de saída fornecida pela porta lógica, quando a saída se encontra no nível lógico baixo (LOW)

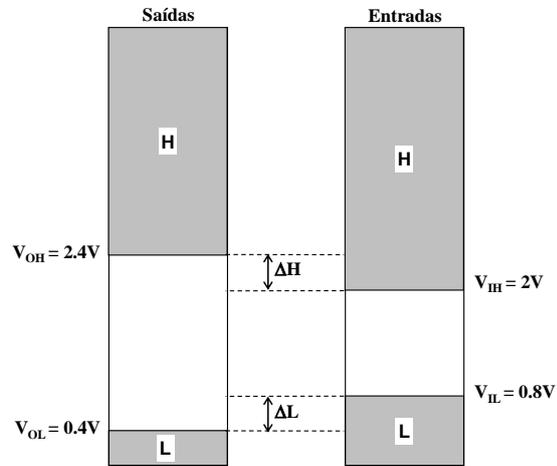
V_{IH} - Tensão mínima de entrada interpretada pela porta lógica como nível lógico alto (HIGH)

V_{IL} - Tensão máxima de entrada interpretada pela porta lógica como nível lógico baixo (LOW)



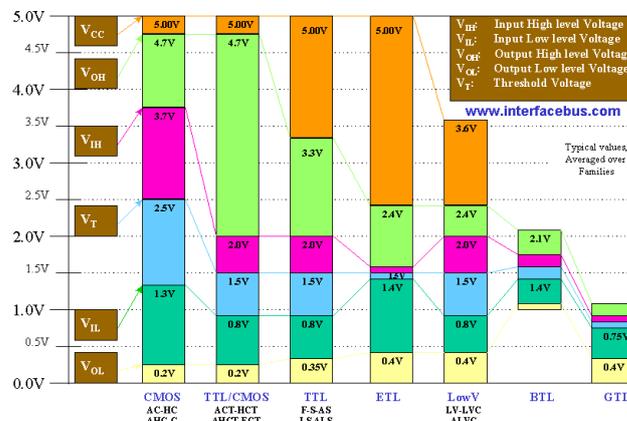
Níveis lógicos:

- ▶ ΔH e ΔL - Correspondem às **margens de ruído**, i.e., à diferença máxima entre os níveis de tensão fornecidos pelas saídas e os níveis de tensão admitidos nas entradas para uma interpretação correcta dos sinais.



Níveis lógicos:

- ▶ Famílias lógicas diferentes consideram, em geral, limites de tensão diferentes para a interpretação dos níveis lógicos





■ FAN-IN e FAN-OUT:

Fan-In: N^o de entradas disponíveis de uma porta lógica.

Fan-Out: N^o de entradas de portas lógicas a que podemos ligar a saída de uma porta, sem degradar o desempenho do circuito.

- ▶ Os circuitos MOS podem ser ligados a um n^o grande de outras portas MOS sem degradação do sinal, mas o atraso aumenta com o n^o de ligações.
- ▶ A velocidade de operação das portas TTL não depende do fan-out, mas a qualidade do sinal degrada-se a partir de um certo n^o de ligações.



■ Lógica positiva:

- ▶ Os catálogos dos fabricantes definem as portas lógicas em termos dos níveis H – “High”, e L – “Low”, e não dos valores lógicos 0 e 1.
 - Sistema de **lógica positiva**: $H \equiv 1, L \equiv 0$.
 - Sistema de **lógica negativa**: $H \equiv 0, L \equiv 1$.

■ Atribuição de nomes a sinais:

- ▶ Os nomes dos sinais são uma forma importante de documentação. Devem ser usados nomes que ajudem a perceber a função de cada sinal:
 - se um determinado sinal faz acender um LED deve ser designado, por exemplo, como **AcendeLED**, e não como X3.
- ▶ É também importante distinguir se o sinal faz acender o LED quando tem o valor H ou quando tem o valor L.
- ▶ Esta distinção é feita habitualmente usando os sufixos **_H** ou **_L**. Diz-se, no primeiro caso, que o sinal é activo a H (ou activo a 1) e, no segundo caso, que o sinal é activo a L (ou activo a 0).
 - **AcendeLED_H** – o sinal é activo a H, quer dizer que acende o LED quando o sinal vale H.
 - **AcendeLED_L** – o sinal é activo a L, quer dizer que acende o LED quando o sinal vale L.



Nota: quando o sufixo é omissivo considera-se habitualmente que o sinal é activo a H.

FUNÇÕES LÓGICAS

■ **Existem 16 funções de 2 variáveis Booleanas:**

x	y	f ₀	f ₁	f ₂	f ₃	f ₄	f ₅	f ₆	f ₇	f ₈	f ₉	f ₁₀	f ₁₁	f ₁₂	f ₁₃	f ₁₄	f ₁₅
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

► **Funções já conhecidas:**

$$f_0(x, y) = 0$$

$$f_3(x, y) = \bar{x}$$

$$f_5(x, y) = \bar{y}$$

$$f_8(x, y) = x \cdot y \quad \text{AND}$$

$$f_{10}(x, y) = y$$

$$f_{12}(x, y) = x$$

$$f_{14}(x, y) = x + y \quad \text{OR}$$

$$f_{15}(x, y) = 1$$

■ **Funções NOR e NAND:**

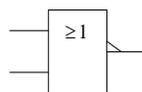
$$f_1(x, y) = \bar{x} \cdot \bar{y} = \overline{x + y} \quad \text{NOR}$$

$$f_7(x, y) = \bar{x} + \bar{y} = \overline{x \cdot y} \quad \text{NAND}$$

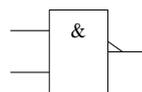
Funcionam como uma porta OR (ou uma porta AND) seguida de uma porta NOT

x	y	f ₁	f ₇
0	0	1	1
0	1	0	1
1	0	0	1
1	1	0	0

► **Simbologia:**



NOR



NAND

Nas tecnologias mais comuns (p.ex. CMOS), as portas NOR e NAND (portas inversoras) requerem menos transístores que as portas OR e AND (portas não inversoras).

De facto, as portas OR e AND é que são habitualmente realizadas com um porta NOR ou NAND seguida de uma porta NOT.

■ Funções OU-EXCLUSIVO:

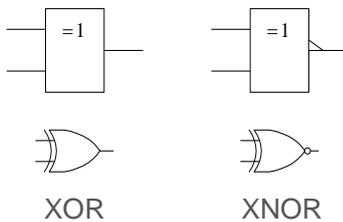
$$f_6(x, y) = \bar{x} \cdot y + x \cdot \bar{y} = x \oplus y \quad \mathbf{XOR}$$

$$f_9(x, y) = \bar{x} \cdot \bar{y} + x \cdot y = \overline{x \oplus y} \quad \mathbf{XNOR}$$

XOR é verdadeira se uma e apenas uma das 2 entradas for verdadeira.

x	y	f ₆	f ₉
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

► Simbologia:



■ Funções baseadas no operador Booleano IMPLICAÇÃO:

$$f_{11}(x, y) = \bar{x} + x \cdot y = \bar{x} + y = x \rightarrow y \quad : x \text{ implica } y$$

$$f_{13}(x, y) = \bar{y} + x \cdot y = y \rightarrow x$$

$$f_2(x, y) = \overline{y \rightarrow x}$$

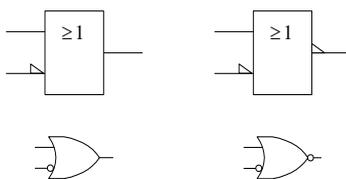
$$f_4(x, y) = \overline{x \rightarrow y}$$

x	y	f ₂	f ₄	f₁₁	f ₁₃
0	0	0	0	1	1
0	1	1	0	1	0
1	0	0	1	0	1
1	1	0	0	1	1

Exemplo:

Se partir, paga!!!

► Simbologia:

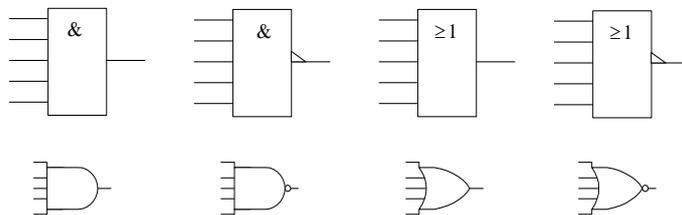


Estas funções não estão habitualmente disponíveis como portas lógicas básicas para realização de sistemas digitais.

Portas com mais de 2 entradas:

- ▶ As operações AND e OR (e consequentemente as portas NAND e NOR) são facilmente generalizáveis para N-entradas.
- ▶ Uma porta AND de N entradas tem a saída a 1 sse todas as entradas estiverem a 1.
- ▶ Uma porta OR de N entradas tem a saída a 1 se pelo menos uma entrada estiver a 1.

Simbologia:



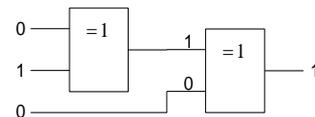
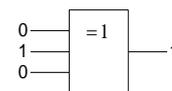
Função OU-EXCLUSIVO com mais de 2 entradas:

$$x \oplus y \oplus z = (x \oplus y) \oplus z$$

- ▶ A porta XOR de 3 entradas é verdadeira se uma e apenas uma das 3 entradas for igual a 1, ou se as 3 entradas forem iguais a 1.

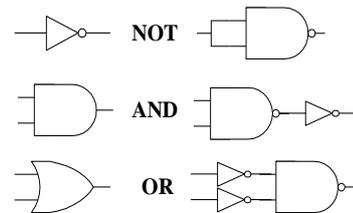
$$x_1 \oplus x_2 \oplus x_3 \oplus \dots \oplus x_N = (((x_1 \oplus x_2) \oplus x_3) \oplus \dots) \oplus x_N$$

- ▶ A porta XOR de N entradas é verdadeira se o número de entradas iguais a 1 for ímpar.
- ▶ De facto, e embora usada genericamente, a designação OU-exclusivo só é estritamente correcta para a função de 2 variáveis.
- ▶ As funções de paridade são muito utilizadas em sistemas de comunicação que requerem detecção de erros: um bit de paridade é habitualmente usado para detectar erros de transmissão.



■ Circuitos com portas NAND:

- ▶ A porta NAND é considerada uma porta universal porque qualquer circuito digital pode ser realizado apenas com portas NAND.
- ▶ Qualquer função booleana é realizável apenas com portas NAND por substituição directa das operações NOT, AND e OR.
- ▶ A operação NOT é normalmente considerada em sentido lato, como uma NAND de 1 entrada.
- ▶ Nalgumas tecnologias (p.ex. TTL) as portas NAND são as portas mais simples (portanto mais baratas), pelo que é vantajosa a realização de circuitos só com NANDs.

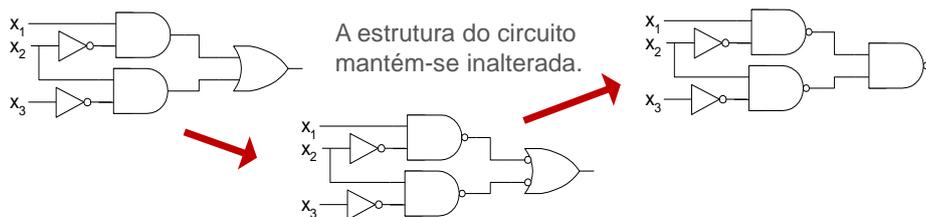


■ Circuitos com portas NAND (cont.):

- ▶ Uma função representada na forma de uma soma de produtos pode ser transformada numa forma directamente realizável apenas com portas NAND por simples aplicação da lei de DeMorgan.

Exemplo:

$$\begin{aligned}
 f &= x_1\bar{x}_2 + \bar{x}_3x_2 = \overline{\overline{x_1\bar{x}_2 + \bar{x}_3x_2}} = \overline{\overline{x_1\bar{x}_2} \cdot \overline{\bar{x}_3x_2}} \\
 &= (x_1 \text{ nand } \bar{x}_2) \text{ nand } (\bar{x}_3 \text{ nand } x_2)
 \end{aligned}$$



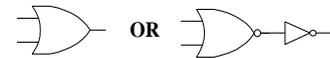
■ **Circuitos com portas NOR:**

Dual:

- ▶ Qualquer circuito pode ser realizado apenas com portas NOR.

- ▶ No caso de a função estar representada

como um produto de somas, a transformação mantém a estrutura.



$$g = (x_1 + \bar{x}_2) \cdot (\bar{x}_3 + x_2) = \overline{\overline{(x_1 + \bar{x}_2)} \cdot \overline{(\bar{x}_3 + x_2)}} = \overline{\overline{(x_1 + \bar{x}_2)} + \overline{(\bar{x}_3 + x_2)}} = \overline{(x_1 \text{ nor } \bar{x}_2) \text{ nor } (\bar{x}_3 \text{ nor } x_2)}$$



PRÓXIMA AULA



■ Tema da Próxima Aula:

- ▶ Funções lógicas:
 - Circuitos com portas NAND;
 - Circuitos com portas NOR;
- ▶ Representações normalizadas:
 - Soma de produtos;
 - Mintermos;
 - Produto de somas;
 - Maxtermos;
- ▶ Funções incompletamente especificadas.



Agradecimentos

Algumas páginas desta apresentação resultam da compilação de várias contribuições produzidas por:

- Guilherme Arroz
- Horácio Neto
- Nuno Horta
- Pedro Tomás