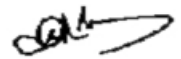


Circuitos Lógicos

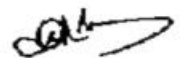
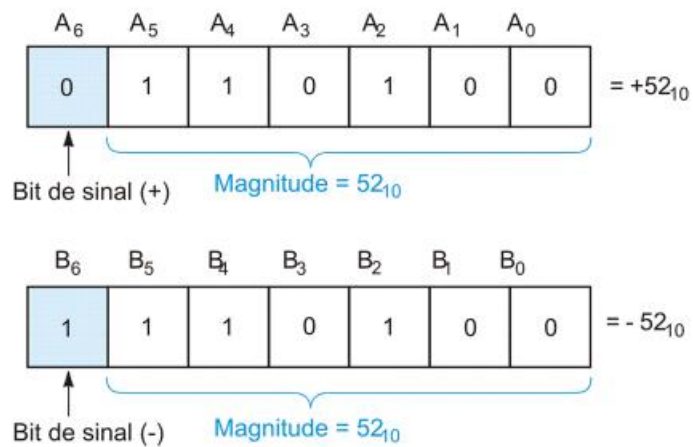


Módulo # 4

Circuitos Aritméticos e Programáveis



Representação de números com sinal na forma sinal-magnitude.



Representação de números com sinal na forma complemento de 1.



Representação de números com sinal na forma complemento de 2.

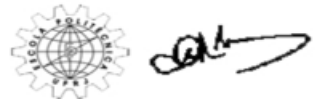
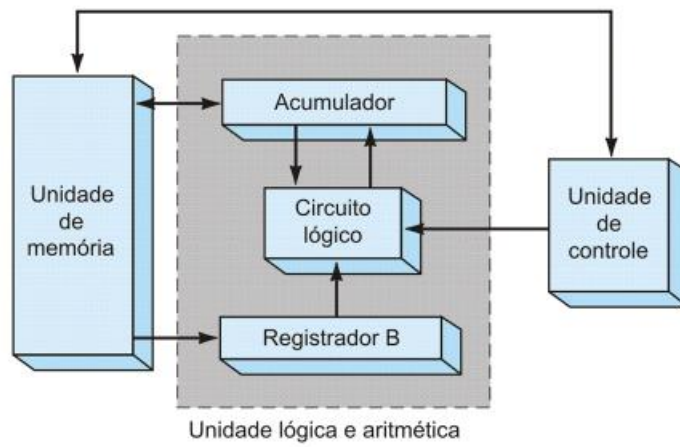
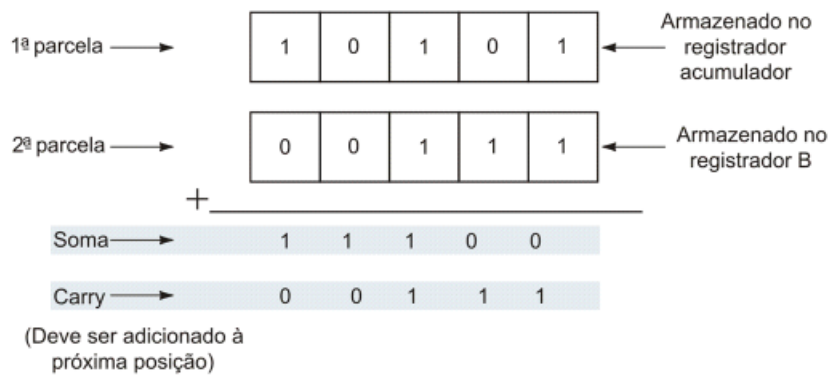


Diagrama básico de um processador digital



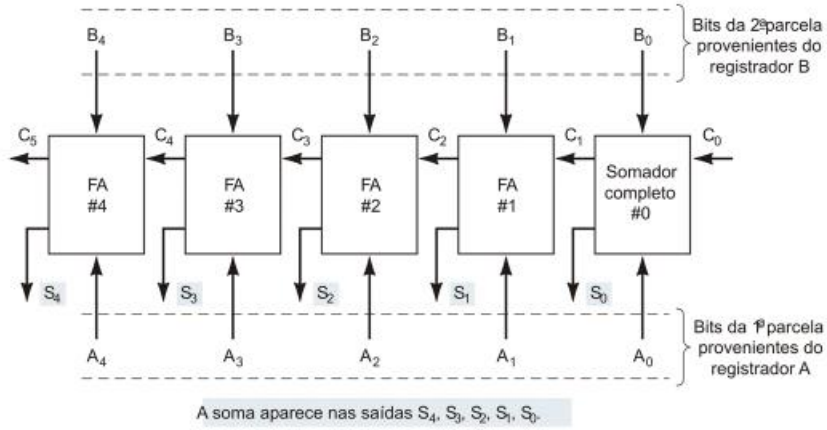
Assinatura

Processo típico de uma adição binária.



Assinatura

Circuito somador paralelo usando somadores completos



Handwritten signature

Mapas de Karnaugh para um somador completo

	$\overline{C_{IN}}$	C_{IN}
$\overline{A}B$	0	1
$A\overline{B}$	1	0
AB	0	1
$A\overline{B}$	1	0

Mapa K para S

$$S = \overline{A}B\overline{C_{IN}} + A\overline{B}\overline{C_{IN}} + AB\overline{C_{IN}} + A\overline{B}C_{IN}$$

	$\overline{C_{IN}}$	C_{IN}
$\overline{A}B$	0	0
$A\overline{B}$	0	1
AB	1	1
$A\overline{B}$	0	1

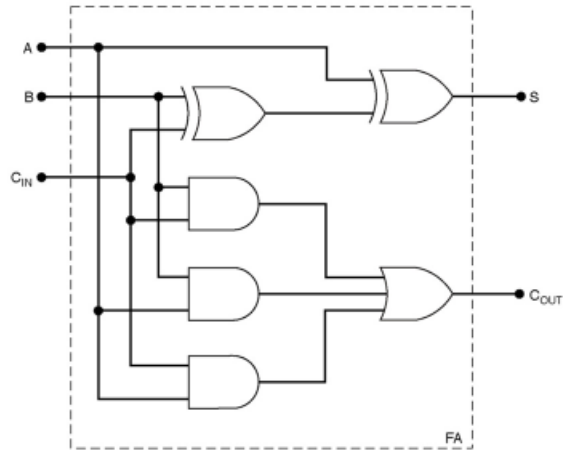
Mapa K para C_{OUT}

$$C_{OUT} = BC_{IN} + AC_{IN} + AB$$

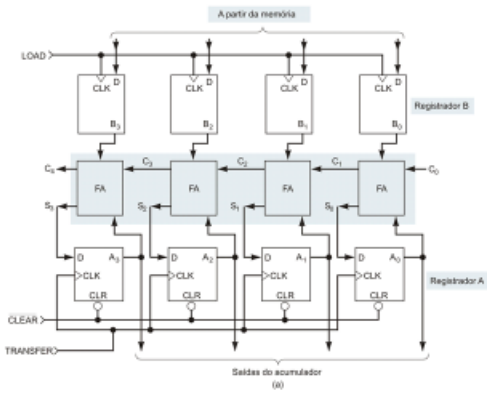


Handwritten signature

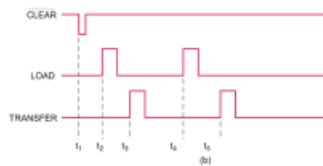
Circuito de um somador completo.



Handwritten signature



Somador de quatro bits completo com registradores

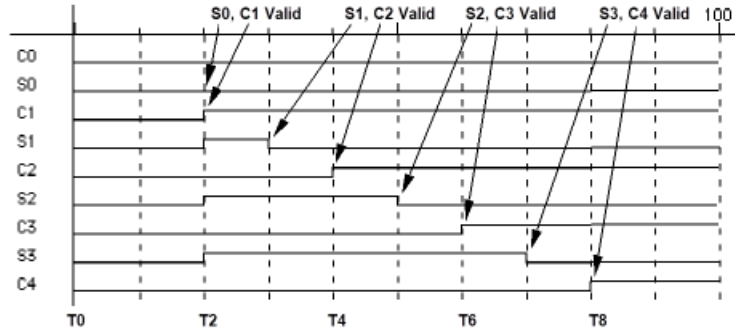
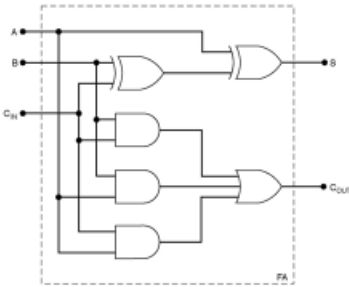
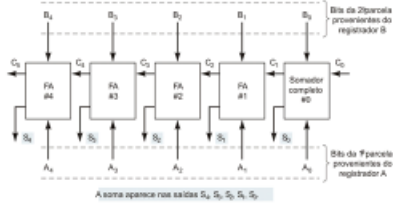


Sinal usado para somar os números binários provenientes da memória e armazenar o resultado no acumulador



Handwritten signature

Problema : propagação do carry



Handwritten signature

Carry lookdeahead

$G_i = A_i \cdot B_i$ gerador do carry
 $P_i = A_i \text{ xor } B_i$ propagador do carry

$S_i = A_i \text{ xor } B_i \text{ xor } C_i = P_i \text{ xor } C_i$

$C_{i+1} = A_i \cdot B_i + A_i \cdot C_i + B_i \cdot C_i$
 $= A_i \cdot B_i + C_i \cdot (A_i + B_i)$
 $= A_i \cdot B_i + C_i \cdot (A_i \text{ xor } B_i)$
 $= G_i + C_i \cdot P_i$

equivalentes

$$\begin{aligned}
 a b + c (a \oplus b) &= a b + c (a \bar{b} + a \bar{b}) \\
 &= a b + \bar{a} b c + a \bar{b} c \\
 &= a (b + \bar{b} c) + \bar{a} b c = a b + a c + \bar{a} b c \\
 &= a b + c (a + b)
 \end{aligned}$$

Dai:

$C_1 = G_0 + P_0 C_0$

$C_2 = G_1 + P_1 C_1 = G_1 + P_1 G_0 + P_1 P_0 C_0$

$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$

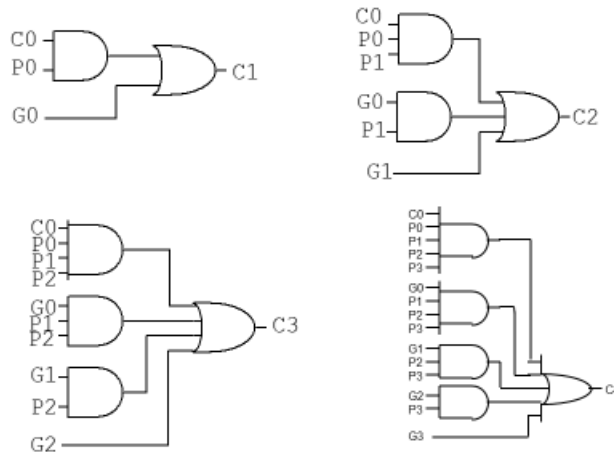
$C_4 = G_3 + P_3 C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$

A	B	C	C _{i+1}
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



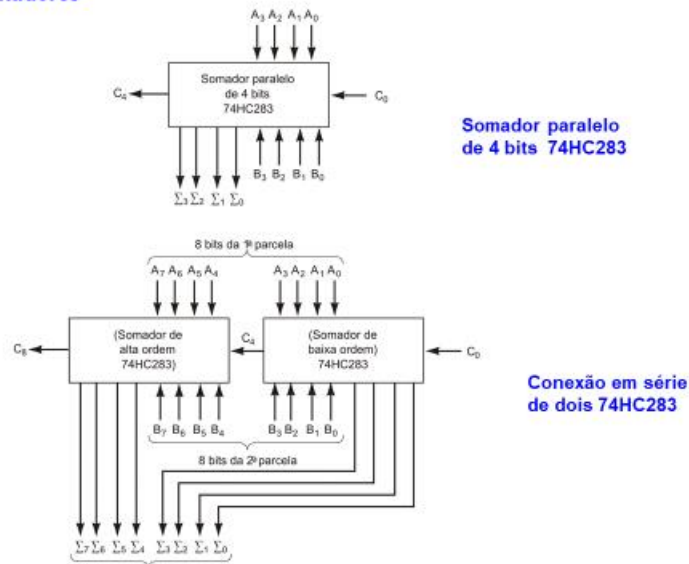
Handwritten signature

Carry lookdeahead , por ordem ...



Handwritten signature

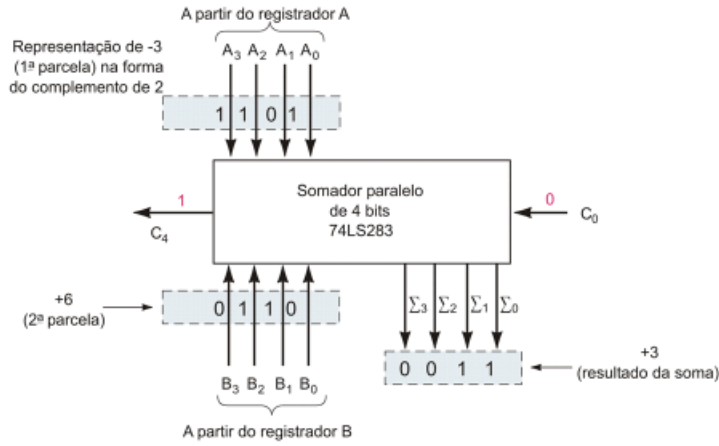
Circuitos somadores



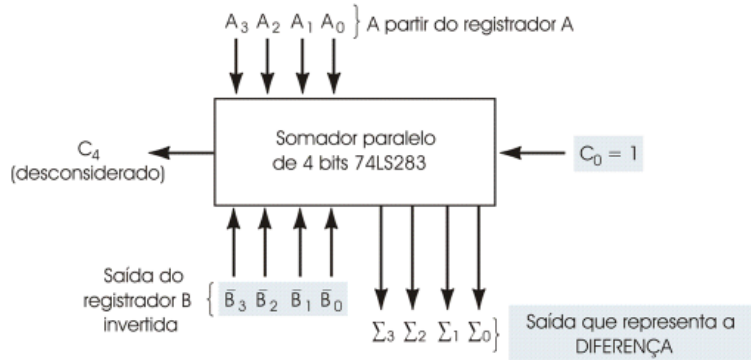
Handwritten signature



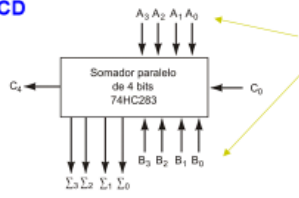
Adição de um número positivo (+) com um negativo (-) no sistema complemento de 2.



Subtração (A-B) usando o sistema complemento de 2.
Os bits do subtraendo (B) são invertidos e C₀ = 1 para gerar o complemento de 2 de um número positivo (+) com um negativo (-)

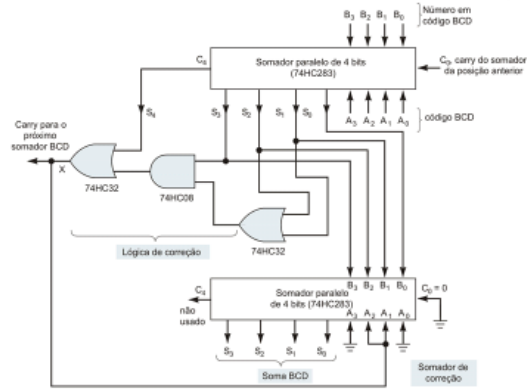


Somador BCD

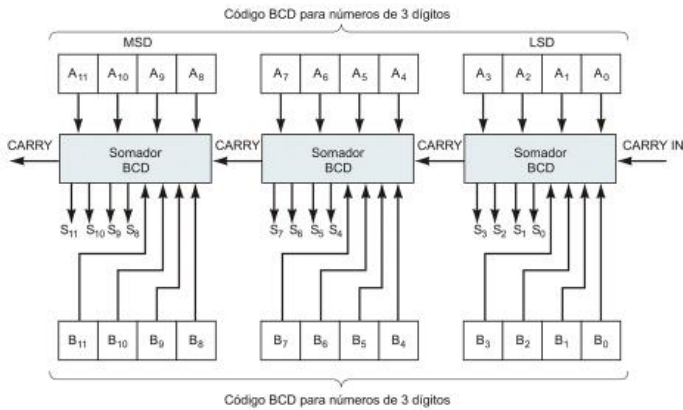


números em BCD

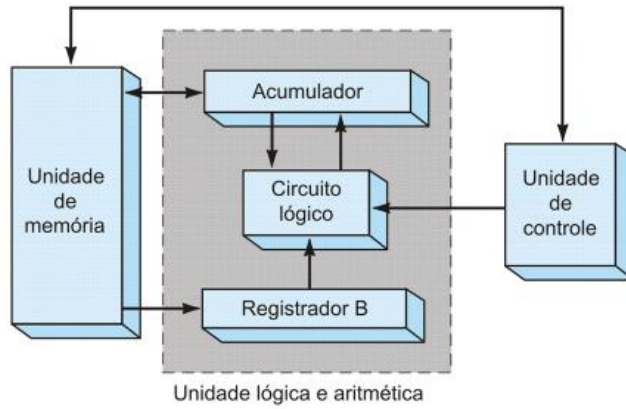
...contém dois somadores de quatro bits e um circuito detector para correção.



Conexão de somadores em cascata para somar dois números decimais de três dígitos.



ALU ou ULA

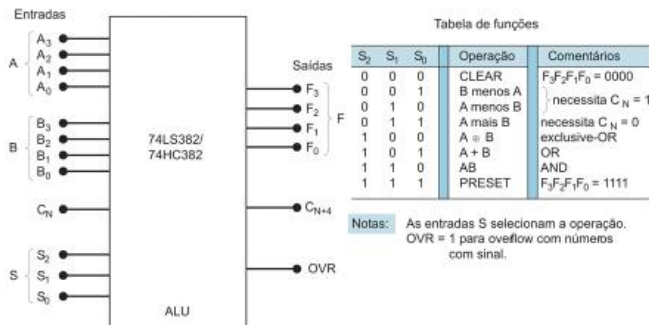


ALU ou ULA



Handwritten signature

ALU 74LS382/HC382



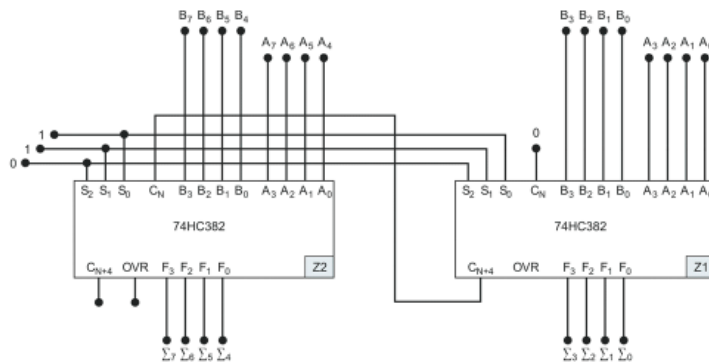
A = nomes de entrada de 4 bits
B = nomes de entrada de 4 bits
C_N = carry de entrada para a posição LSB
S = entrada de seleção de operação de três bits

F = nomes de saída de quatro bits
C_{N+4} = carry de saída da posição MSB
OVR = indicador de overflow



Handwritten signature

Dois chips ALU 74HC382 conectados com um somador de 8 bits.



Notas: Z1 soma os bits de baixa ordem
 Z2 soma os bits de alta ordem
 $\sum_7 \dots \sum_0 = 8$ bits de soma
 OVR de Z2 é o indicador de overflow de 8 bits



Algumas definições ...

- A maioria dos computadores manipulam e armazenam informações e dados binários em grupos de 8 bits. O conjunto de oito bits é denominado 1 byte.
- Um byte pode representar vários tipos de dados / informações. Uma palavra é um grupo de bits que representa uma determinada unidade de informação. O tamanho da palavra pode ser definido como o número de bits na palavra binária em que um sistema digital opera. O tamanho da palavra de um PC é de 8 bytes (64 bits).
- Números em códigos binários frequentemente são divididos em grupos de 4 bits, denominado *nibble*.



Deteção de erros ... método de paridade



Muitos sistemas digitais empregam métodos para detecção de erros e, por vezes, até para a correção dos mesmos.

Um dos métodos mais simples e mais utilizados para detecção de erros é o Método de Paridade, que requer a adição de um bit extra para cada grupo de dados (códigos).

Neste método, há duas formas possíveis : paridade par e paridade ímpar. Transmissor e receptor devem adotar um deles.

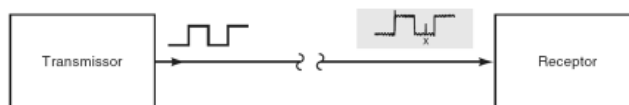


Deteção de erros ...



A transmissão de dados (códigos binários) de um local para outro acontece frequentemente em sistemas digitais. Como exemplos, temos as atividades de armazenamento / recuperação de dados em unidades externas de memória, e a comunicação entre sistemas computacionais através de canais telefônicos.

O ruídos elétricos e flutuações espúrias na tensão, eventos comuns em todos os sistemas eletrônicos, podem causar erros durante a transmissão dos dados.



Deteção de erros ... método de paridade PAR



Método de paridade PAR : o número total de bits em um grupo, incluindo o bit de paridade, deve ser um número par.

Por exemplo, o grupo binário 1011 requer a adição de um bit de paridade 1, formando o novo grupo 11011, que tem um número par de 1s.

Método de paridade ÍMPAR : o número total de bits em um grupo, incluindo o bit de paridade, deve ser um número ímpar.

Por exemplo, o mesmo grupo binário 1011 requer a adição de um bit de paridade 0, formando o novo grupo 01011, que tem um número ímpar de 1s.



PLDs

Dispositivos Lógicos Programáveis

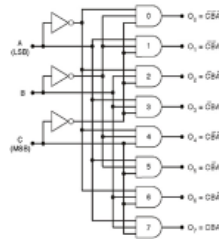


PLD (Programmable Logic Devices) : CI de uso geral
 CI semi-dedicado
 CI dedicado

Vantagens : espaço ocupado, confiabilidade, facilidade de projeto, tempo de projeto, modularidade

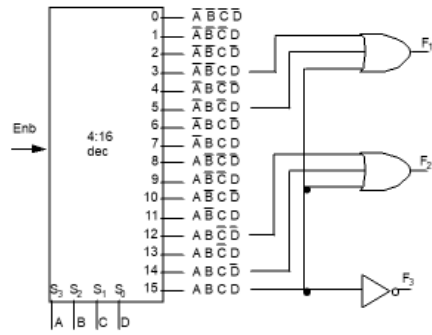


Dec 3:8

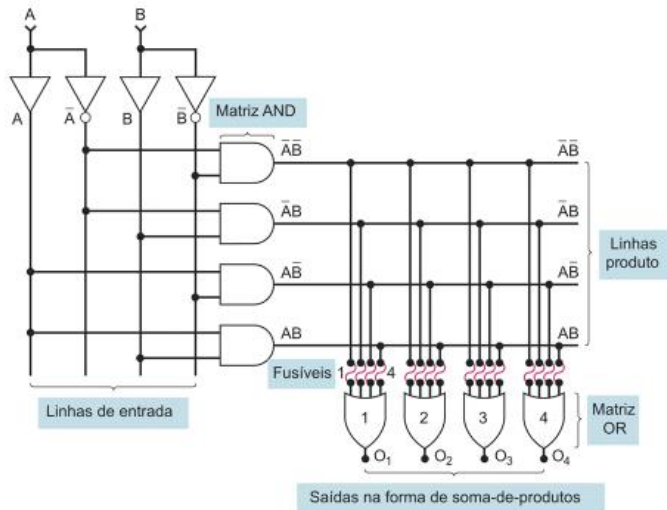


C	B	A	O ₀	O ₁	O ₂	O ₃	O ₄	O ₅	O ₆	O ₇
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	1	0	0	0	0
1	1	1	0	0	0	1	0	0	0	0

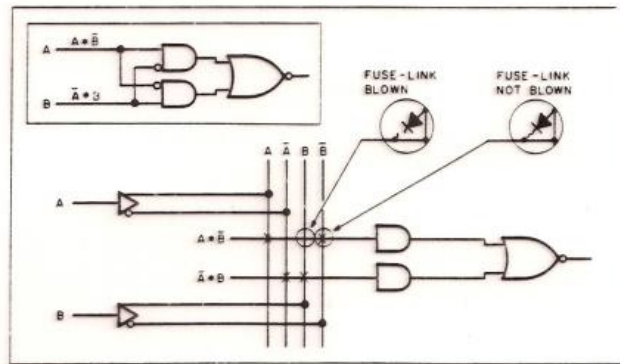
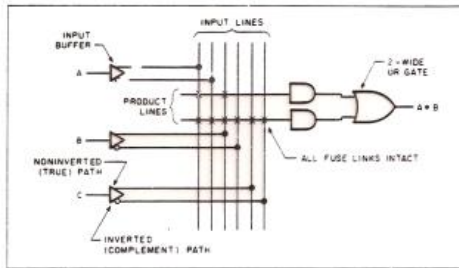
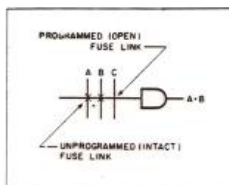
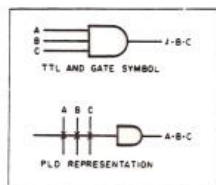
Dec 4:16



Handwritten signature



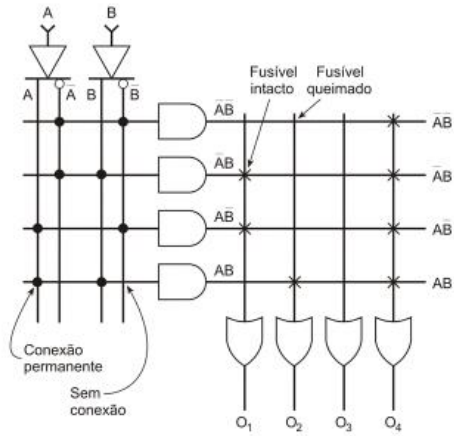
Handwritten signature



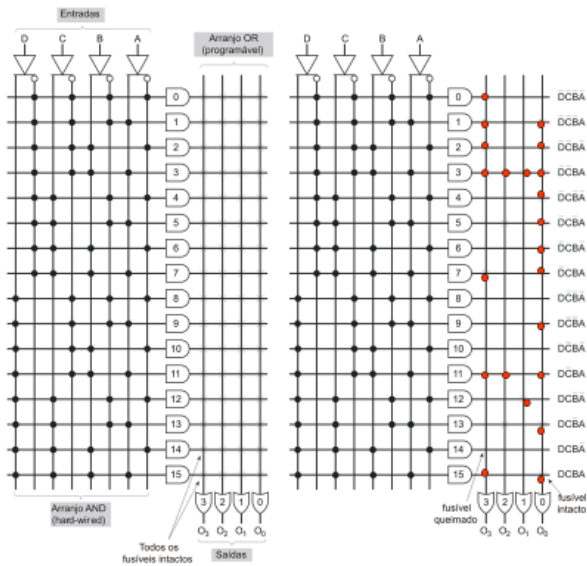
Handwritten signature



Handwritten signature



Handwritten signature



Quem são O₃, O₂, O₁ e O₀ ?



Handwritten signature



DISPOSITIVOS LÓGICOS PROGRAMÁVEIS

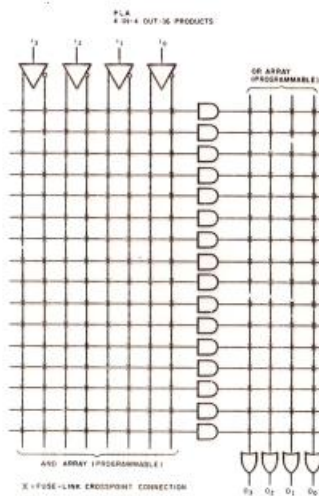
PLD (Programmable Logic Devices): CI de uso geral
 CI semi-dedicado
 CI dedicado

Vantagens : espaço ocupado, confiabilidade, facilidade de projeto, tempo de projeto, modularidade

PROM (programmable read-only memory) - matriz de ANDs fixa / matriz de ORs programável

pró : estrutura aproveitável para $6 < n < 10$, onde n é o no. de variáveis de entrada

contra : para $n = 10$ são 1024 mintermos/portas AND e muitos não serão necessários



PLA (programmable logic array) - matriz de ANDs programável / matriz de ORs programável

FPLA (field- programmable logic array)

pró : no. de mintermos $< 2^n \rightarrow n$ maior com os ANDs programáveis

pró : maior flexibilidade na programação (ANDs e ORs programáveis)

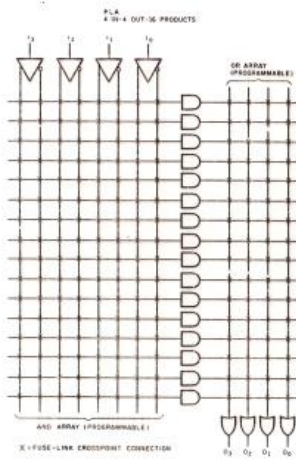
contra : exige circuitos adicionais para a programação da matriz OR

contra : elaboração de algoritmos de programação pode ser complexa e laboriosa

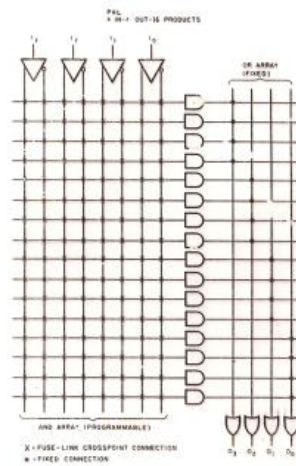


PLA





PLA



PAL



PLA (programmable logic array) - matriz de ANDs programável / matriz de ORs programável

FPLA (field-programmable logic array)

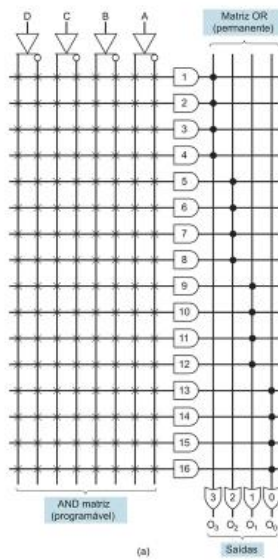
- pró : no. de mintermos $< 2^n \rightarrow n$ maior com os ANDs programáveis
- pró : maior flexibilidade na programação (ANDs e ORs programáveis)
- contra : exige circuitos adicionais para a programação da matriz OR
- contra : elaboração de algoritmos de programação pode ser complexa e laboriosa

PAL(D) (programmable-array logic device) - matriz de ANDs programável / matriz de ORs fixa

- pró : mesma flexibilidade da FPLA nos circuitos de entrada
- pró : baixo custo
- pró : arquitetura flexível



Handwritten signature



PLA (programmable logic array) - matriz de ANDs programável / matriz de ORs programável

FPLA (field-programmable logic array)

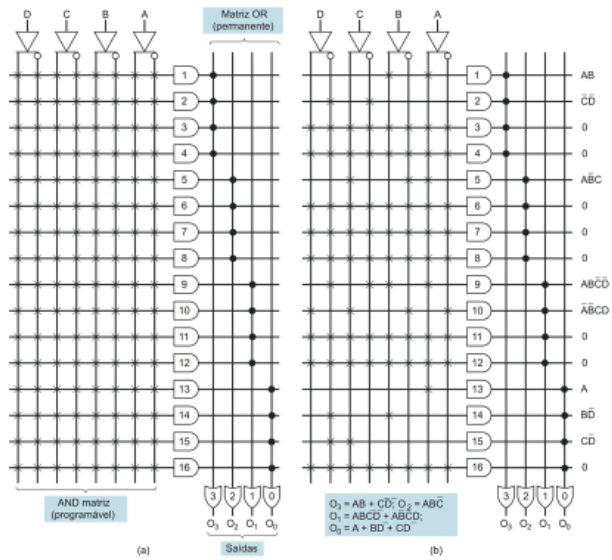
- pró : no. de mintermos $< 2^n \rightarrow n$ maior com os ANDs programáveis
- pró : maior flexibilidade na programação (ANDs e ORs programáveis)
- contra : exige circuitos adicionais para a programação da matriz OR
- contra : elaboração de algoritmos de programação pode ser complexa e laboriosa

PAL(D) (programmable-array logic device) - matriz de ANDs programável / matriz de ORs fixa

- pró : mesma flexibilidade da FPLA nos circuitos de entrada
- pró : baixo custo
- pró : arquitetura flexível



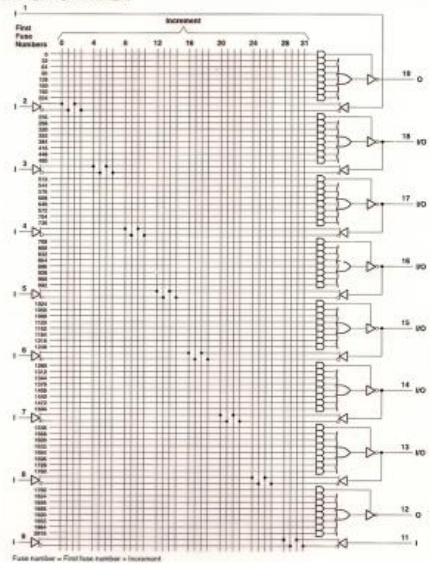
Handwritten signature



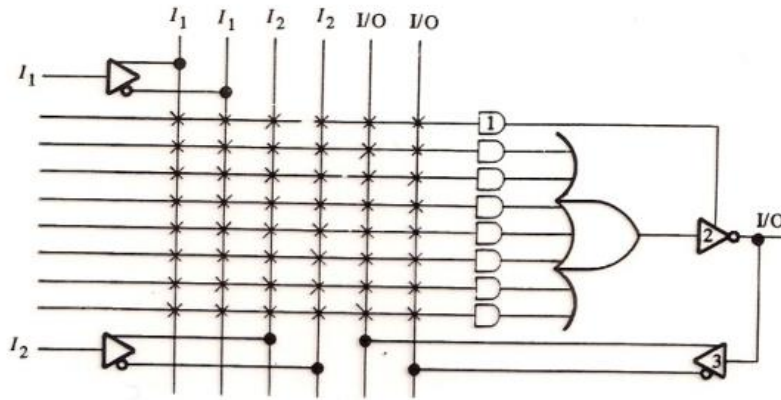
Handwritten signature

TIBPAL16L8-25C
TIBPAL16L8-30M
LOW-POWER HIGH-PERFORMANCE **IMPACT™** PAL® CIRCUITS

logic diagram (positive logic)



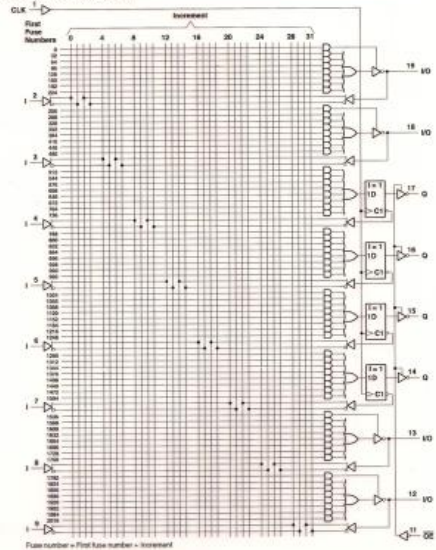
Handwritten signature



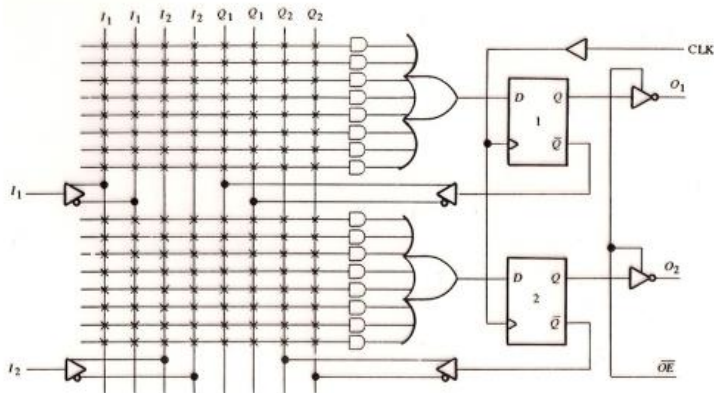
Handwritten signature

TIBPAL16R4-26C
TIBPAL16R4-30M
LOW-POWER HIGH-PERFORMANCE *IMPACT™* PAL® CIRCUITS

logic diagram (positive logic)



Handwritten signature



Handwritten signature

Leitura indicada

Maini, A.K. "Digital Electronics – Principles and Integrated Circuits"

a) Sec. 6.1 – 6.10 , pgs. 203 – 234

b) Sec. 11.1 – 11.11 , pgs. 423 – 462



6

Arithmetic Circuits

LEARNING OBJECTIVES

After completing this chapter, you will learn the following:

- Basic design steps for implementing combinational logic circuits.
- Basic building blocks of arithmetic circuits including half-adder, full adder, half-subtractor, full subtractor, and controlled inverter.
- Addition of BCD numbers.
- Carry propagation and look-ahead carry generation.
- Magnitude comparison.
- Application-relevant information on commonly used integrated circuits (ICs) for arithmetic operations.

11

Programmable Logic Devices

LEARNING OBJECTIVES

After completing this chapter, you will learn the following:

- Difference between fixed and programmable logic.
- Advantages and disadvantages of fixed and programmable logic devices.
- Using ROM as a PLD.
- Introduction to different programmable logic devices.
- Programmable logic array (PLA).
- Programmable array logic (PAL) and its variants.
- Complex programmable logic devices (CPLD).
- Field programmable gate array (FPGA).
- Introduction to PLD programming languages.



Handwritten signature