

CAPÍTULO 5

ARITMÉTICA DIGITAL

- Introdução
- Números decimais
- Números binários positivos
- Adição Binária
- Números negativos
- Extensão do bit de sinal
- Adição e Subtração
- Overflow Aritmético
- Circuitos Aritméticos
- Somador Binário Paralelo
- Projeto de um Full Adder (FA)
- Somador completo com registradores
- Atraso de propagação do Carry
- Circuitos Integrados somadores
- Circuito somador/subtrator
- Somador/subtrator com registradores
- Circuitos Integrados ALU
- Biblioteca Altera – ALU de 8 bits
- Exercícios

Introdução

- Os computadores realizam diversas operações aritméticas diretamente nos números representados na sua forma binária
- É fundamental para programadores de computadores o bom entendimento das teorias por trás da aritmética digital
- Alguns objetivos deste capítulo:

Números decimais

- Para entender melhor o complemento de 2, vamos formalizar revendo a representação decimal
- No sistema decimal usamos 10 dígitos decimais (0,1,2,3,4,5,6,7,8,9). Ou seja, usamos a base 10.
- O número 4728 significa:
$$4728 = (4 \times 1000) + (7 \times 100) + (2 \times 10) + 8$$
- Para números fracionários:
$$472,83 = (4 \times 10^2) + (7 \times 10^1) + (2 \times 10^0) + (8 \times 10^{-1}) + (3 \times 10^{-2})$$
- Em geral, para $X = \{... x_2 x_1 x_0 ... x_{-1} x_{-2} x_{-3} ...\}$, o valor de X é

$$X = \sum_i x_i 10^i$$

Números binários positivos

- No sistema binário o mesmo formalismo vale, porém usando-se somente os dois dígitos “0” e “1”.
- Valores fracionários também usam a potência negativa da base
- Com 8 bits podemos representar números positivos de 0 a 255
- Exemplos:

$$10_2 = (1 \times 2^1) + (0 \times 2^0) = 2_{10}$$

$$11_2 = (1 \times 2^1) + (1 \times 2^0) = 3_{10}$$

$$100_2 = (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) = 4_{10}$$

$$1001,101 = 2^3 + 2^0 + 2^{-1} + 2^{-3} = 9,625_{10}$$

$$00000000 = 0$$

$$00000001 = 1$$

$$00101001 = 41$$

$$10000000 = 128$$

$$11111111 = 255$$

$$A = \sum_{i=0}^{n-1} 2^i a_i$$

Adição Binária

- A adição binária é feita exatamente como fazemos a adição decimal com lápis e papel:

$$\begin{array}{r} 3 \ 7 \ 6 \quad \text{LSD} \\ + 4 \ 6 \ 1 \\ \hline 8 \ 3 \ 7 \end{array}$$

- Na adição binária, quatro casos podem ocorrer:

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$1 + 1 = 10 = 0 + \text{vai um}$$

$$1 + 1 + 1 = 11 = 1 + \text{vai um}$$

- Alguns exemplos:

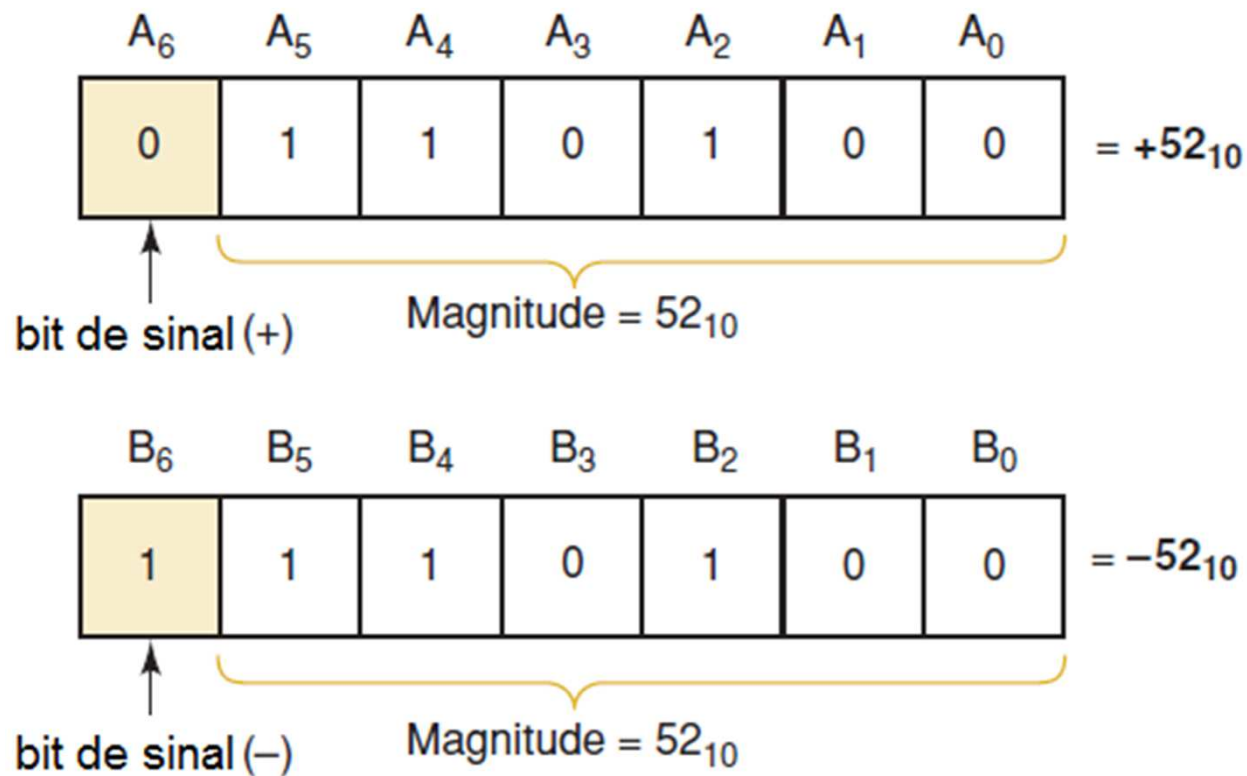
$$\begin{array}{r} 011 \ (3) \\ + 110 \ (6) \\ \hline 1001 \ (9) \end{array}$$

$$\begin{array}{r} 1001 \ (9) \\ + 1111 \ (15) \\ \hline 11000 \ (24) \end{array}$$

$$\begin{array}{r} 11.011 \ (3.375) \\ + 10.110 \ (2.750) \\ \hline 110.001 \ (6.125) \end{array}$$

Números negativos

- Sistema sinal/magnitude:



Números negativos

- Complemento de 1 ($\sim A$)

$$\begin{array}{rcccccc} A & 1 & 0 & 1 & 1 & 0 & 1 \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \sim A & 0 & 1 & 0 & 0 & 1 & 0 \end{array}$$

- Complemento de 2 ($\bar{A}$):

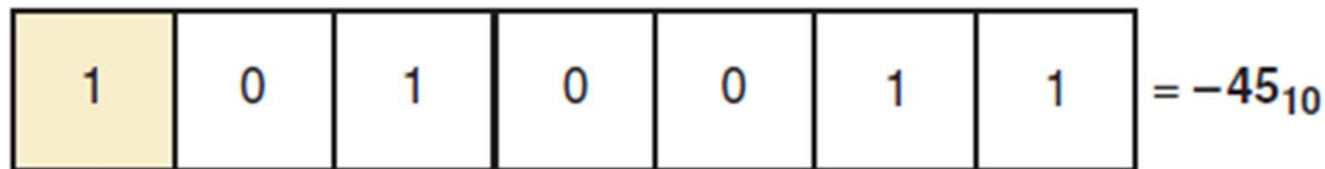
$$\begin{array}{rcccccc} A & 1 & 0 & 1 & 1 & 0 & 1 \\ \sim A & 0 & 1 & 0 & 0 & 1 & 0 \\ + & & & & & & 1 \\ \hline \bar{A} & 0 & 1 & 0 & 0 & 1 & 1 \end{array}$$

Números negativos

- O complemento de 2 é usado para representar números negativos na maioria dos sistemas



↑
bit de sinal (+)



↑
bit de sinal (-)

Números negativos

- Usa o bit mais significativo para representar o sinal
- Considere um número A de n bits representado em complemento de dois.
- Se o número é positivo o bit de sinal a_{n-1} será igual a zero e a magnitude do número será dada por:

$$A = \sum_{i=0}^{n-2} 2^i a_i$$

Números negativos

- Se o número é negativo, o bit de sinal a_{n-1} é 1
- Os $n - 1$ bits restantes podem representar até 2^{n-1} valores. Portanto a faixa de números negativos é de -1 até -2^{n-1} .
- Para agregar propriedades aritméticas, o bit mais significativo é multiplicado por -2^{n-1} .
- Então, o valor de um número negativo em complemento de dois é:

$$\bar{A} = -2^{n-1}a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i$$

- Esta equação também vale para os números positivos

Números negativos

- Podemos facilmente provar que $\bar{A}$ pode ser obtido pelo complemento de 1 de A mais 1:

$$\bar{A} = -2^{n-1}a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i$$

- Se $\bar{A}$ representa um número negativo, $a_{n-1} = 1$ e o valor positivo correspondente será:

$$-\bar{A} = 2^{n-1} - \sum_{i=0}^{n-2} 2^i a_i$$

$$-\bar{A} = 2^{n-1} - 1 + 1 - \sum_{i=0}^{n-2} 2^i a_i$$

$$-\bar{A} = 1 + \sum_{i=0}^{n-2} 2^i \times 1 - \sum_{i=0}^{n-2} 2^i a_i = 1 + \sum_{i=0}^{n-2} 2^i \times (1 - a_i)$$

$$-\bar{A} = 1 + \sum_{i=0}^{n-2} 2^i \times ! a_i$$

$$-\bar{A} = 1 + \sim A$$

Números negativos

Exercício:

Encontre os valores que os números abaixo representam, considerando que são representações em complemento de dois de 16 bits.

1. FFF0_{hex}

2. 8300_{hex}

Extensão do bit de sinal

- As imagens abaixo mostram o número -128 na calculadora do Windows, no modo programador e em hexadecimal
- O que está mudando em cada imagem é o tipo do dado: Qword, Dword, Word e Byte.



Adição e Subtração

- Vejamos alguns exemplos de adição e subtração com complemento de 2

Caso I – dois números positivos

$$\begin{array}{r} +9 \rightarrow \boxed{0} \ 1001 \\ +4 \rightarrow \boxed{0} \ 0100 \\ \hline \boxed{0} \ 1101 \end{array} \quad \begin{array}{l} \text{parcelas} \\ \text{soma} = +13 \end{array}$$

↑ bit de sinal

Caso II – número positivo e número negativo menor

$$\begin{array}{r} \phantom{\boxed{0}} \\ +9 \rightarrow \boxed{0} \ 1001 \\ -4 \rightarrow \boxed{1} \ 1100 \\ \hline 1 \ 0 \ 0101 \end{array} \quad \begin{array}{l} \text{bit de sinal} \\ \text{parcelas} \end{array}$$

↑ vai um descartado. O resultado é 00101 (soma = +5)

Adição e Subtração

Caso III - número positivo e número negativo menor

$$\begin{array}{r} -9 \rightarrow 10111 \\ +4 \rightarrow 00100 \\ \hline 11011 \end{array} \quad \text{soma} = -5$$

↑ bit de sinal negativo

Caso IV – Dois números negativos

$$\begin{array}{r} -9 \rightarrow 10111 \\ -4 \rightarrow 11100 \\ \hline 1 \ 10011 \end{array}$$

↑ bit de sinal negativo
↑ vai um descartado. O resultado é 10011 (soma = -13)

Adição e Subtração

Caso V – Dois números iguais e opostos

-9 → 10111

+9 → 01001

0 1 00000

↑ vai um descartado. O resultado é 00000

A subtração também cai em um desses 5 casos, pois é tratada como uma adição com o subtraendo em complemento de 2

Overflow Aritmético

- Nos exemplos vistos, os valores eram pequenos e todos os resultados couberam no número de 5 bits. Veja este exemplo:

$$\begin{array}{r} +9 \rightarrow \boxed{0} \ 1001 \\ +8 \rightarrow \boxed{0} \ 1000 \\ \hline \boxed{1} \ \underbrace{0001} \end{array}$$

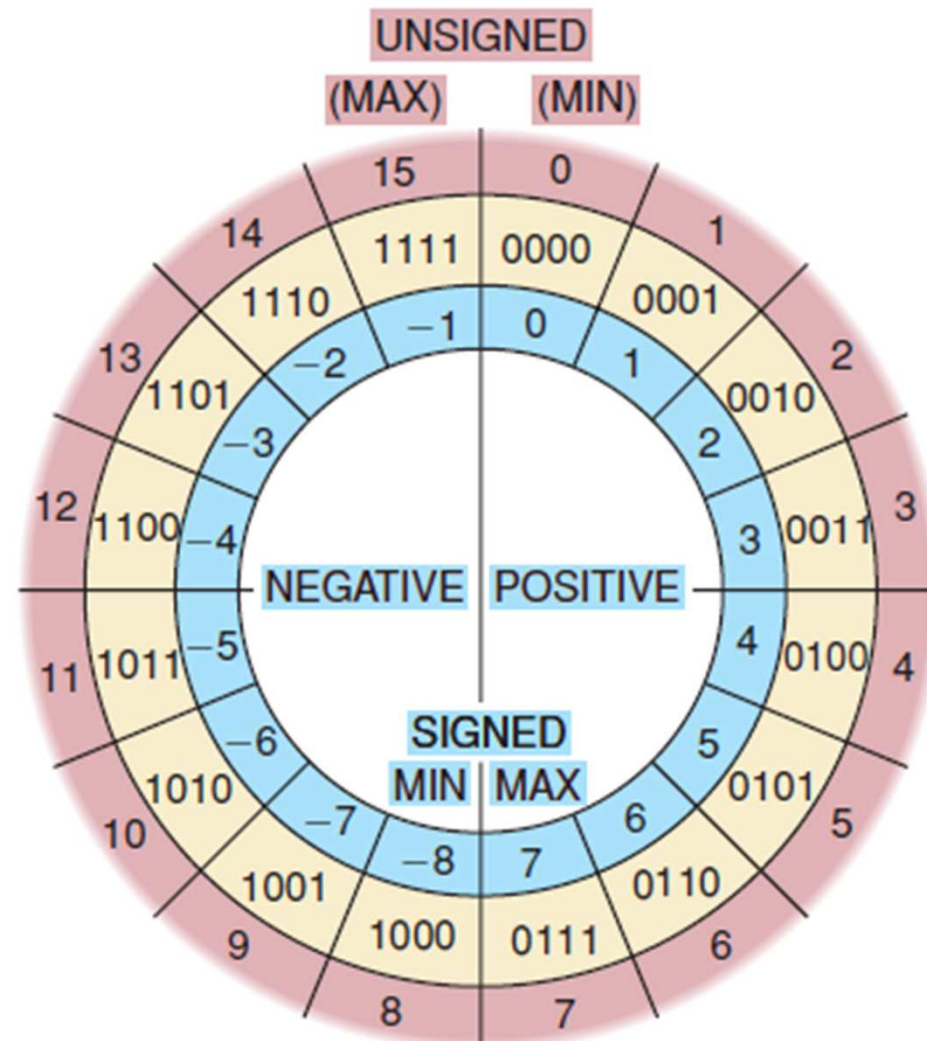
sinal incorreto $\uparrow$ $\uparrow$ magnitude incorreta

- O resultado está incorreto, que deveria ser +17. A magnitude do resultado sobrescreveu o bit de sinal. Ocorreu um **overflow**

Overflow Aritmético

- O overflow só pode ocorrer se os dois números somados tiverem o mesmo sinal
- Detectamos overflow quando o resultado se apresenta com bit de sinal diferente das parcelas da soma
- Na soma de números positivos não sinalizados, o último vai um (carry out) já significa overflow
- Nos demais casos é necessário analisar os sinais das parcelas e do resultado

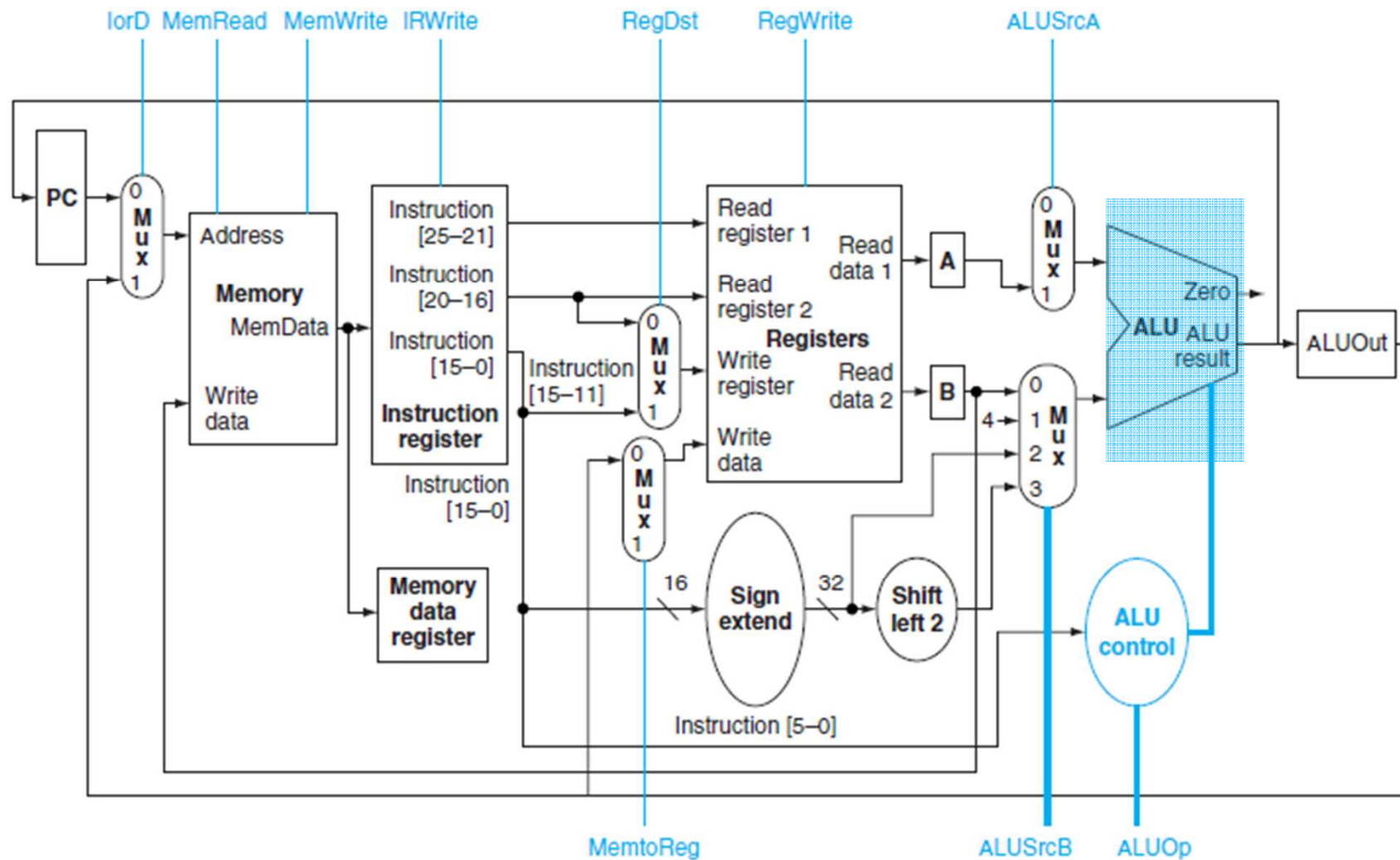
Círculo de números de 4 bits



Circuitos Aritméticos

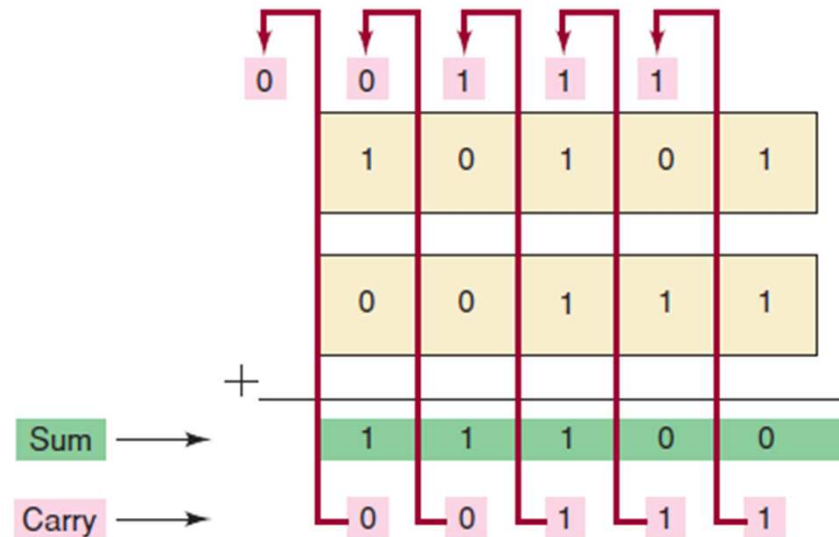
- Operações aritméticas e lógicas são realizadas em um computador pela unidade lógica e aritmética (ALU)
- A ALU possui todos os circuitos necessários para adicionar, subtrair, fazer operações and, or, etc bit a bit, deslocamentos, etc.
- Como são realizadas por circuitos digitais, o tempo dessas operações é da ordem de nanosegundo
- Em um processador, a operação é definida pela unidade de controle, através de bits de controle que são diferentes para cada instrução

Circuitos Aritméticos



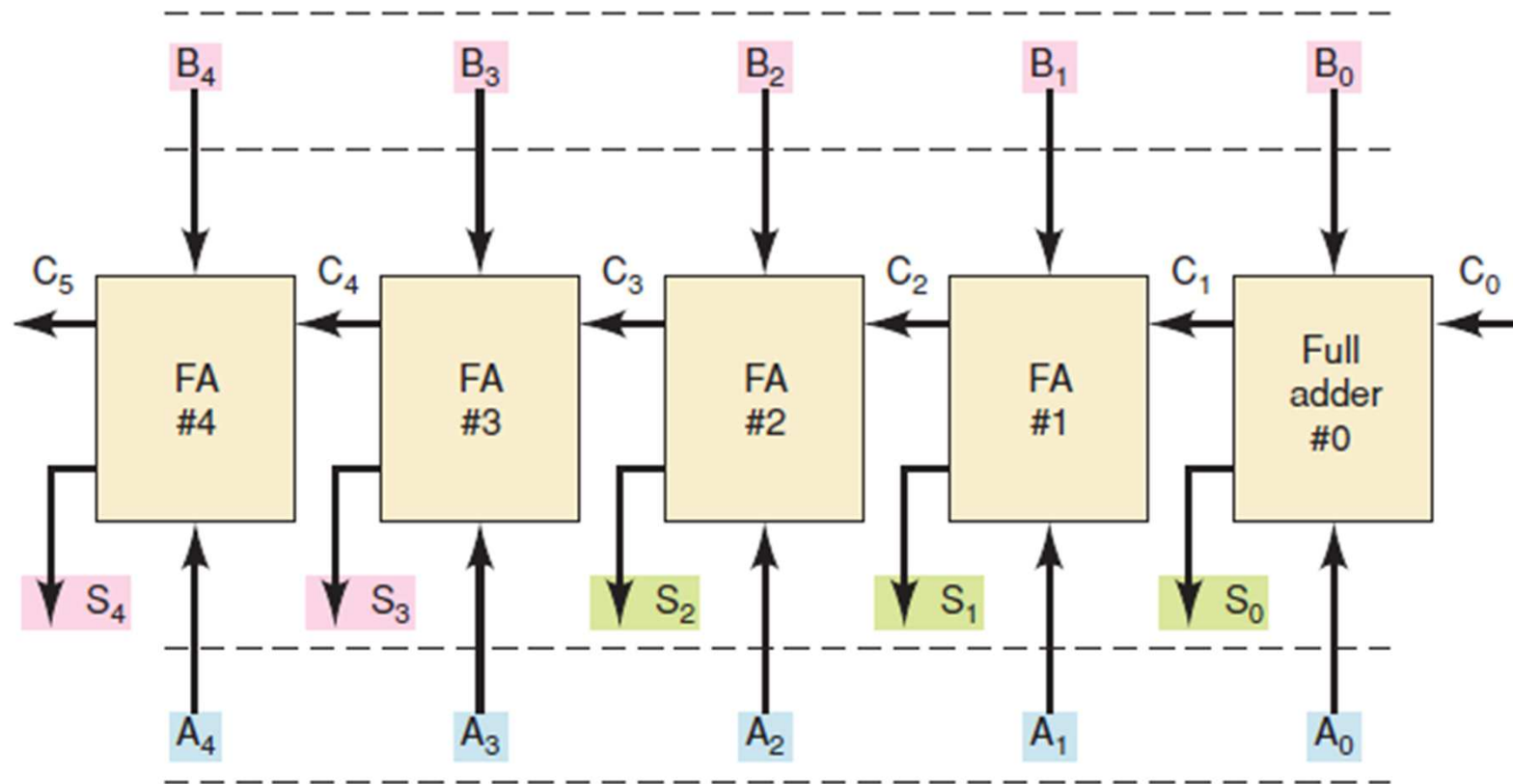
Somador Binário Paralelo

- A soma é realizada paralelamente, isto é, ao mesmo tempo para todos os bits



- Existem dois tipos: **Full Adder (FA)** e **Half Adder (HA)**. O somador completo FA tem carry-in e carry out e o meio somador HA não possui carry-in.

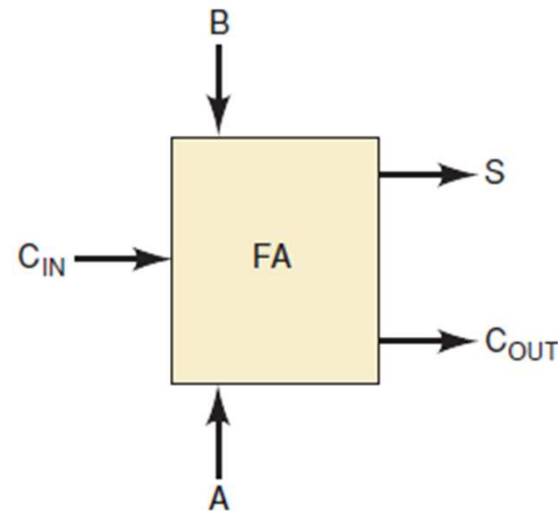
Somador Binário Paralelo



Sum appears at S₄, S₃, S₂, S₁, S₀ outputs.

Projeto de um Full Adder (FA)

A	B	C _{IN}	S	C _{OUT}
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



$$S = \overline{A}\overline{B}C_{IN} + \overline{A}B\overline{C}_{IN} + A\overline{B}\overline{C}_{IN} + ABC_{IN}$$

$$S = \overline{A}(\overline{B}C_{IN} + B\overline{C}_{IN}) + A(\overline{B}\overline{C}_{IN} + BC_{IN})$$

$$S = \overline{A}(B \oplus C_{IN}) + A(\overline{B \oplus C_{IN}})$$

$$X = B \oplus C_{IN}$$

$$S = \overline{A} \cdot X + A \cdot \overline{X} = A \oplus X$$

$$S = A \oplus [B \oplus C_{IN}]$$

$$C_{OUT} = \overline{A}BC_{IN} + A\overline{B}C_{IN} + AB\overline{C}_{IN} + ABC_{IN}$$

$$C_{OUT} = BC_{IN}(\overline{A} + A) + AC_{IN}(\overline{B} + B) + AB(\overline{C}_{IN} + C_{IN})$$

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

Projeto de um Full Adder (FA)

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

K map for S

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

(a)

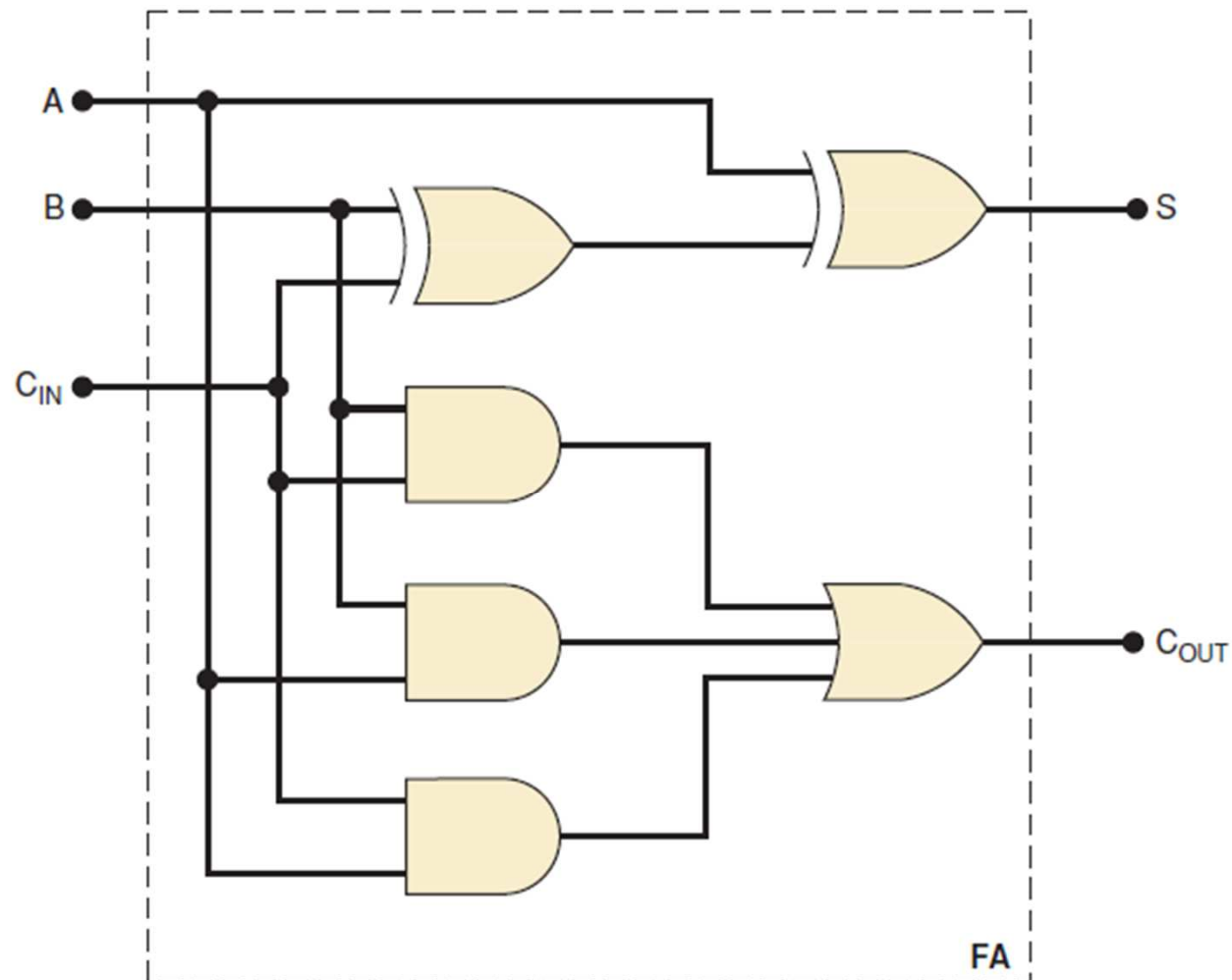
	$\overline{C_{IN}}$	C_{IN}
$\overline{A}\overline{B}$	0	0
$\overline{A}B$	0	1
AB	1	1
$A\overline{B}$	0	1

K map for C_{OUT}

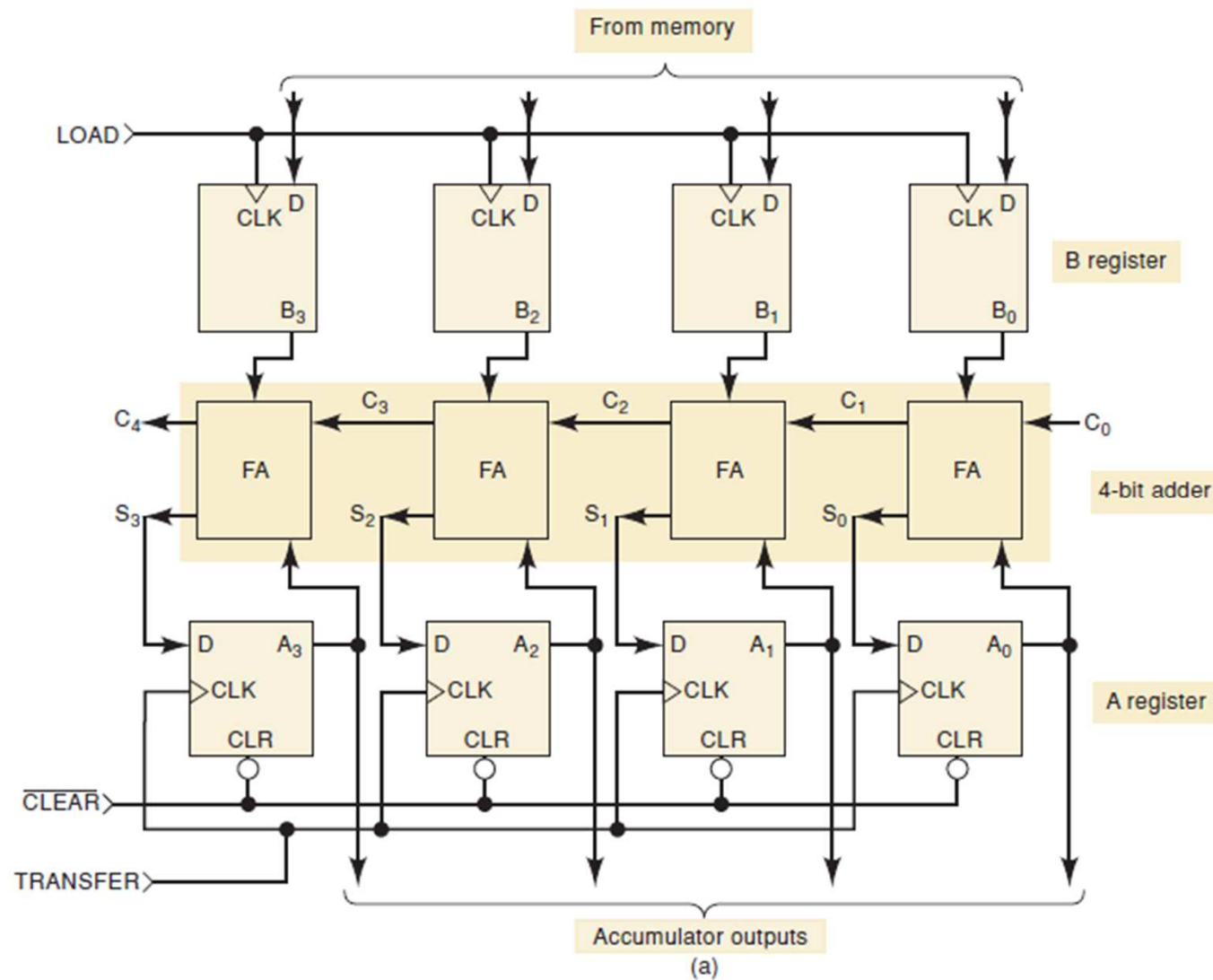
$$C_{OUT} = BC_{IN} + AC_{IN} + AB$$

(b)

Projeto de um Full Adder (FA)



Somador completo com registradores



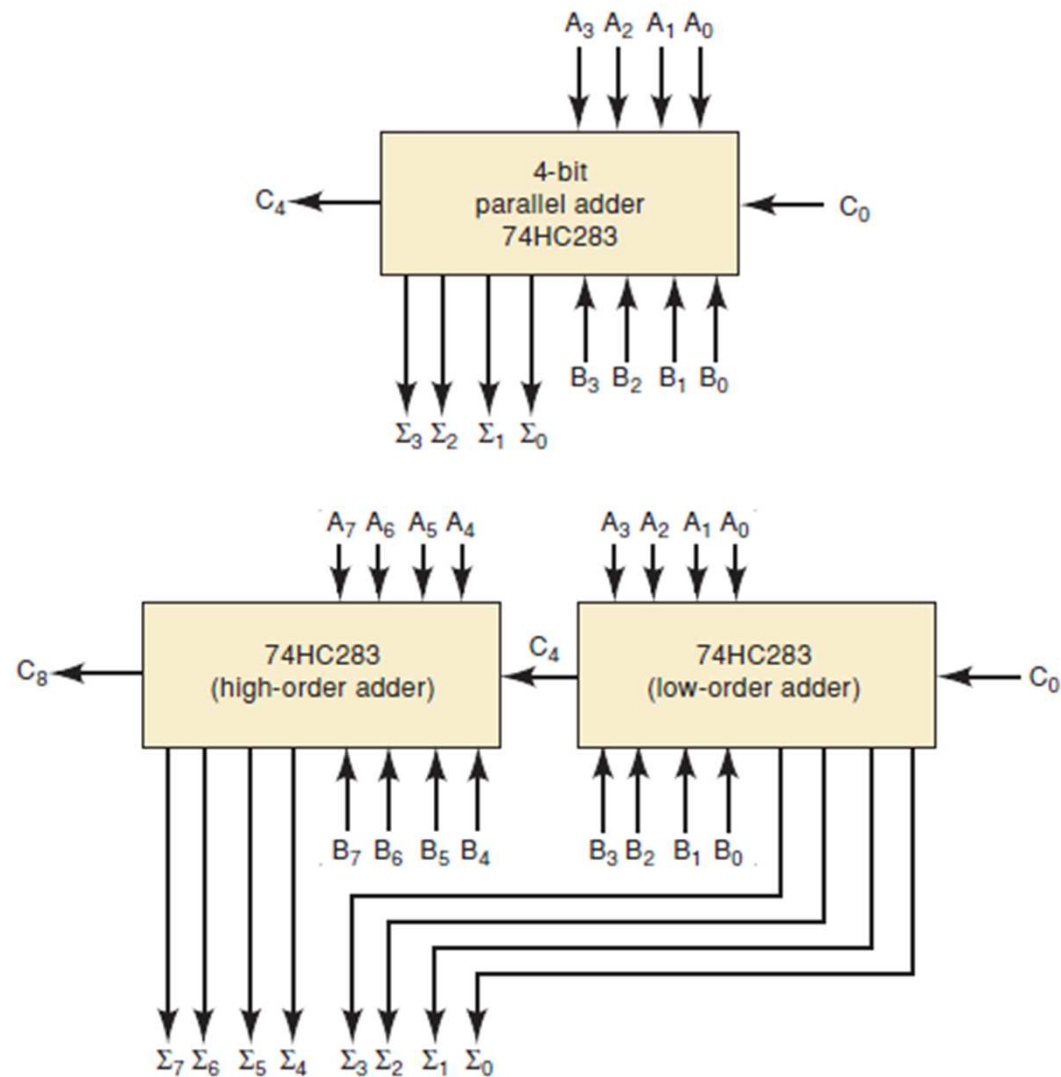
Atraso de propagação do Carry

- Os somadores apresentados podem gerar um tempo relativamente grande de propagação do carry. Por exemplo:

$$\begin{array}{r} 0111 \\ + 0001 \\ \hline 1000 \end{array}$$

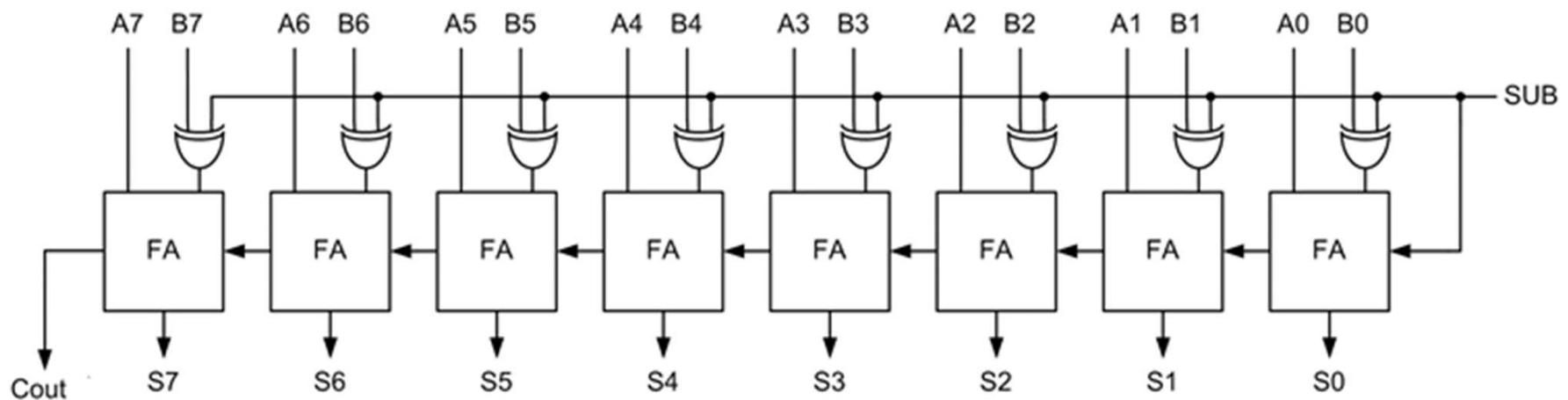
- A resposta correta terá um *delay* que será a soma de todos os atrasos de cada somador.
- É possível reduzir esse tempo. Por exemplo, é possível criar um circuito que gere C_3 em função de A_0, A_1, A_2, B_0, B_1 e B_2
- Essa solução é a chamada **look-ahead carry**
- Apesar de maior complexidade, resolve o problema, e é geralmente usada em circuitos integrados.

Circuitos Integrados somadores

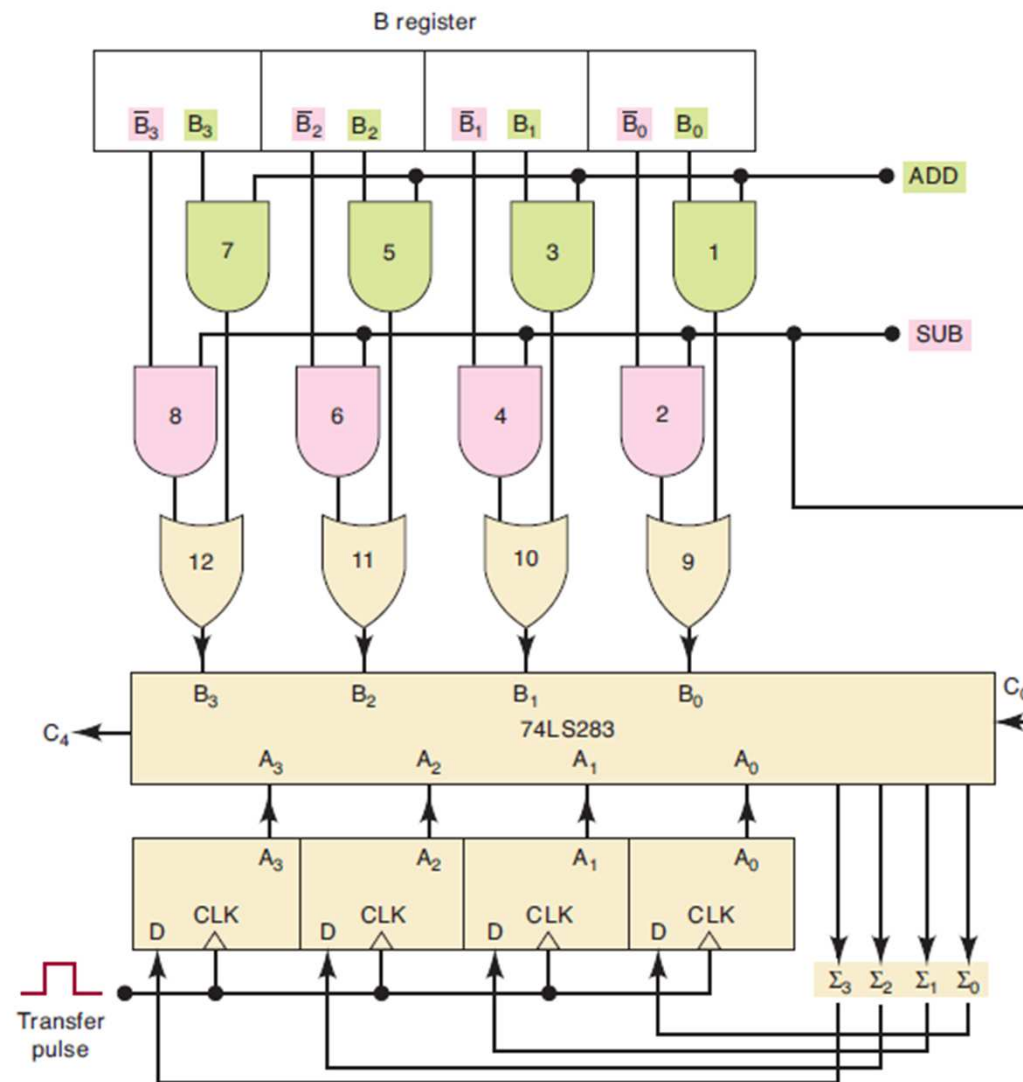


Circuito somador/subtrator

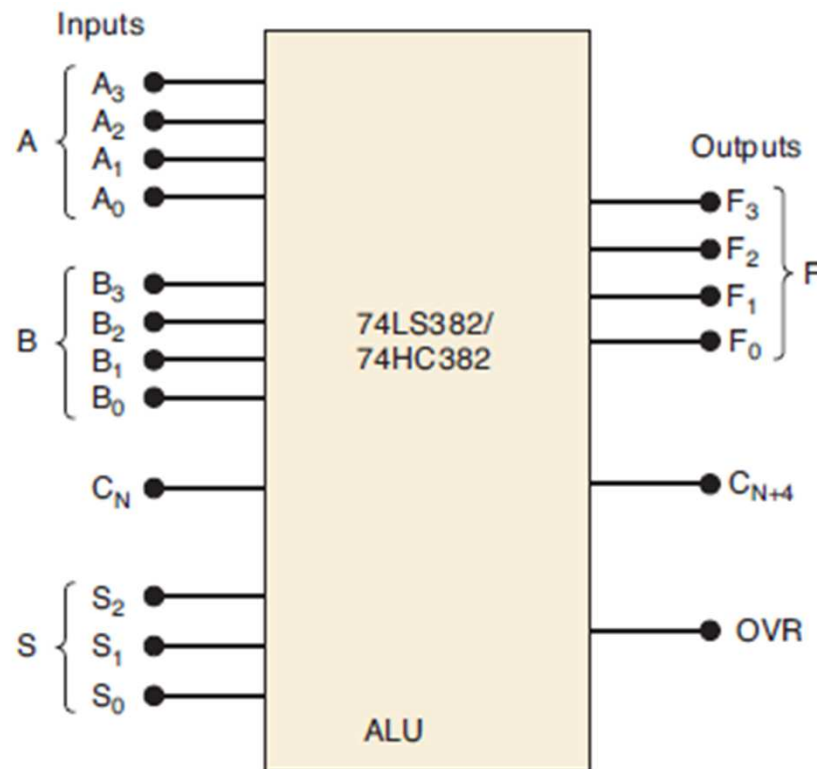
- Uso de portas ou exclusivo para calcular o complemento de dois de um dos operandos
- O “+1” do complemento de dois é obtido através do carry-in do primeiro FA.
- Na operação de soma, o carry in do primeiro FA recebe zero (SUB = 0)



Somador/subtrator com registradores



Circuitos Integrados ALU



A = 4-bit input number
 B = 4-bit input number
 C_N = carry into LSB position
 S = 3-bit operation select inputs

F = 4-bit output number
 C_{N+4} = carry out of MSB position
 OVR = overflow indicator

(a)

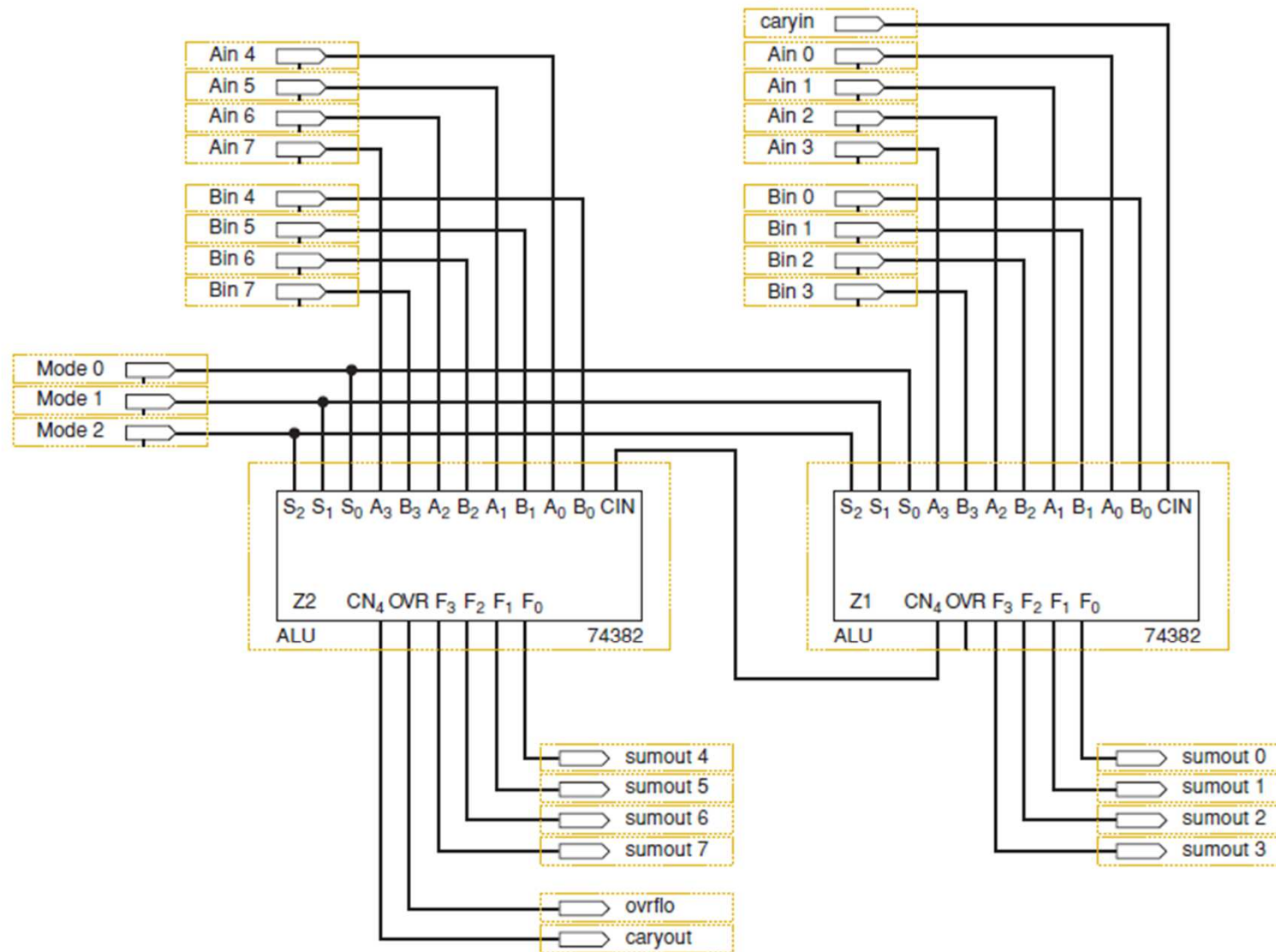
Function Table

S ₂	S ₁	S ₀	Operation	Comments
0	0	0	CLEAR	F ₃ F ₂ F ₁ F ₀ = 0000
0	0	1	B minus A	Needs C _N = 1
0	1	0	A minus B	
0	1	1	A plus B	Needs C _N = 0
1	0	0	A ⊕ B	Exclusive-OR
1	0	1	A + B	OR
1	1	0	AB	AND
1	1	1	PRESET	F ₃ F ₂ F ₁ F ₀ = 1111

Notes: S inputs select operation.
 OVR = 1 for signed-number overflow.

(b)

Biblioteca Altera – ALU de 8 bits



Exercícios

1. Represente cada um dos números decimais sinalizados abaixo na notação complemento de dois com 8 bits.

(a) +32

(e) +127

(i) -1

(m) +84

(b) -14

(f) -127

(j) -128

(n) +3

(c) +63

(g) +89

(k) +169

(o) -3

(d) -104

(h) -55

(l) 0

(p) -190

Exercícios

2. Cada um dos números a seguir representa um número sinalizado na notação complemento de dois. Determine o valor decimal em cada caso.

(a) 01101

(f) 10000000

(b) 11101

(g) 11111111

(c) 01111011

(h) 10000001

(d) 10011001

(i) 01100011

(e) 01111111

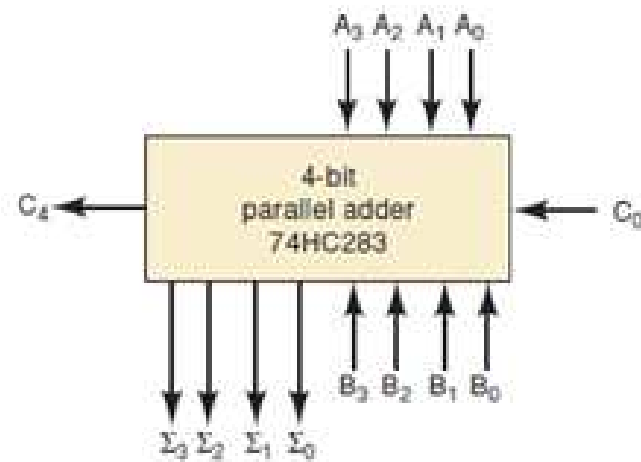
(j) 11011001

Exercícios

3. Uma variável de 16 bits sinalizada possui o valor 1 armazenado e uma outra variável de 8 bits, também sinalizada, possui o valor -128. Mostre bit a bit como seria a soma dessas duas variáveis.

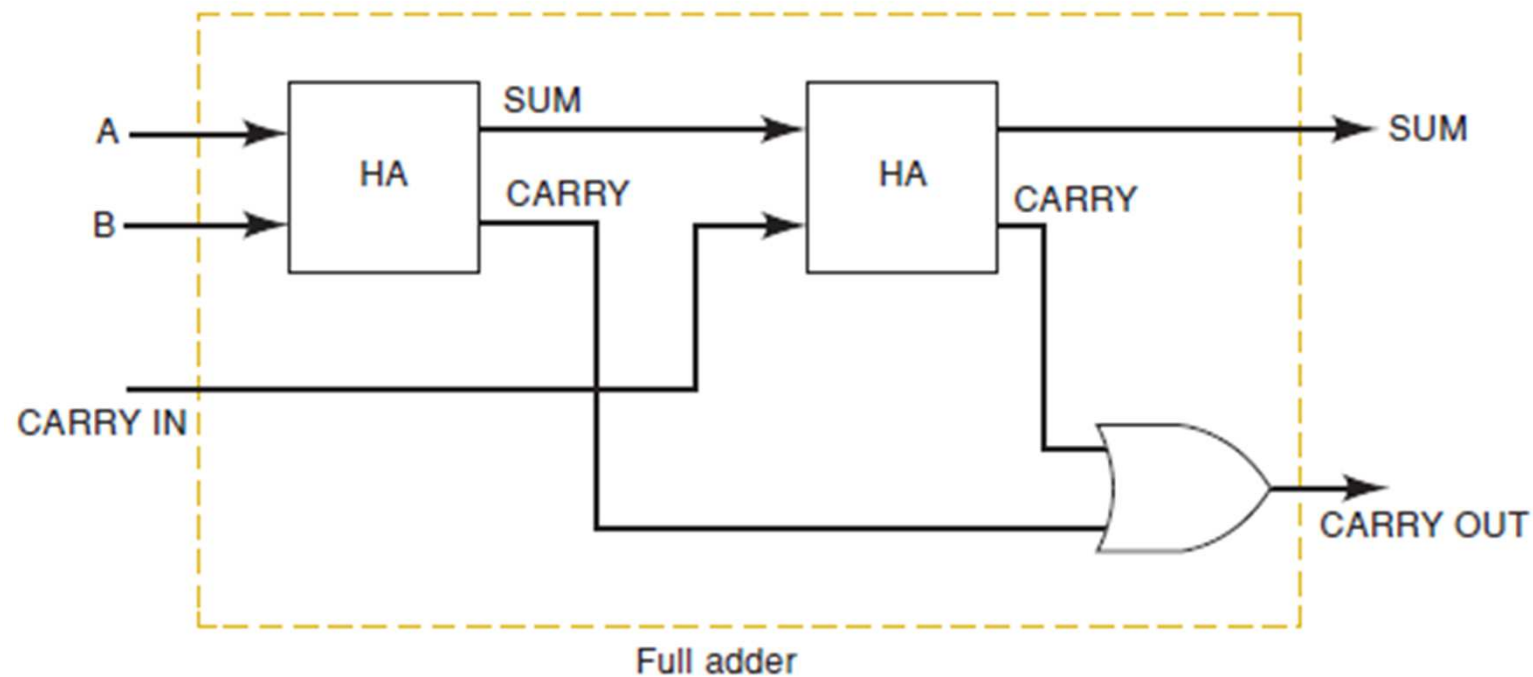
Exercícios

4. Mostre como dois circuitos integrados 74HC283 podem ser usados para somar dois números sinalizados, um de 8 bits e outro de apenas 4 bits.



Exercícios:

5. Mostre que o circuito abaixo é equivalente a um Full Adder (FA):



Exercícios

6. Com relação ao somador completo com registradores apresentado, determine o conteúdo do registrador A após a seguinte sequência de operações:

$[A] = 0000, [0100] \rightarrow [B], [S] \rightarrow [A], [1011] \rightarrow [B], [S] \rightarrow [A].$

Considere os dois casos: $C_0=0$ e $C_0=1$

Exercícios

7. No circuito somador/subtrator com o CI 74LS283 apresentado, determine a saída do somador para os casos abaixo. Determine também, em cada caso, se houve overflow.

- (a) Registrador A = 0101 (+5), registrador B = 1110 (-2);
SUB=1, ADD=0
- (b) Registrador A = 1100 (-4), registrador B = 1110 (-2);
SUB=0, ADD=1
- (c) Registrador A = 1100 (-4), registrador B = 1110 (-2);
SUB=0, ADD=0
- (d) Registrador A = 1101 (-3), registrador B = 0011 (+3);
SUB=1, ADD=0
- (e) Registrador A = 1100 (-4), registrador B = 0010 (+2);
SUB=0, ADD=1
- (f) Registrador A = 1011 (-5), registrador B = 0100 (+4);
SUB=1, ADD=0

Exercícios

8. Determine as saídas F , C_{N+4} e OVR para cada um dos seguintes conjuntos de entradas para o CI 74LS382:

- (a) $[S]=011$, $[A]=0110$, $[B]=0011$, $C_N=0$
- (b) $[S]=001$, $[A]=0110$, $[B]=0011$, $C_N=1$
- (c) $[S]=010$, $[A]=0110$, $[B]=0011$, $C_N=1$

9. Mostre como o CI 74LS382 pode ser usado para obter na saída F o complemento de um de A ($F = \bar{A}$).