Sistemas Digitais

ET46B

Prof. Eduardo Vinicius Kuhn

kuhn@utfpr.edu.br
Curso de Engenharia Eletrônica
Universidade Tecnológica Federal do Paraná

Capítulo 6
Aritmética digital: Operações e circuitos

Conteúdo

Objetivos

- **Compreender e efetuar as operações de soma e subtração de dois números binários.**
- Introduzir formas de representação de **números binários com sinal** (e.g., sinal-magnitude, complemento de 1, e de 2).
- Converter e operar com números binários com sinal usando o **sistema de complemento de 2**.
- Construir o bloco de um **"somador completo"**, o qual é usado para formar um somador/subtrator (paralelo) de N-bits.
- Descrever as operações básicas de uma **unidade lógica/aritmética (ULA)**.
- Comentar sobre *overflow* e atraso de propagação de *carry*.

Introdução

Exemplo: Projete um somador de dois números binários de 8 bits cada (de 0–255), usando o ferramental visto até aqui?

Introdução

Exemplo: Projete um somador de dois números binários de 8 bits cada (de 0–255), usando o ferramental visto até aqui?

R: Uma possível solução se dá através da construção de um circuito com

- 16 **entradas** (i.e., 8 entradas para cada número); e
- 9 **saídas** (capaz de representar valores na faixa de 0–511);

produzindo assim uma **tabela-verdade com $2^{16} = 65.536$ linhas** (tamanho impraticável).

E, se os números contiverem 16 bits cada, produzindo assim uma tabela-verdade com $2^{32} = 4.294.967.296$ linhas?

Portanto, é fundamental compreender como as operações aritméticas com números binários são realizadas; a partir disso, circuitos podem então ser desenvolvidos.

Adição e subtração binária

A adição binária é realizada de forma similar a adição de números decimais, e.g.,

- $0 + 0 = 0$
- $1 + 0 = 1$
- $1 + 1 = 10$ (0 com *carry* de 1)
- $1 + 1 + 1 = 11$ (1 com *carry* de 1)

A operação de *carry* na adição ("vai um") ocorre quando há dois bits '1s' em uma mesma posição.

Exemplos:

011 (3)	1001 (9)	11,011 (3,375)
+ 110 (6)	+ 1111 (15)	+ 10,110 (2,750)
1001 (9)	11000 (24)	110,001 (6,125)

Quando mais de dois números devem ser somados, adicionam-se os dois primeiros e o resultado é acrescido ao terceiro número, e assim por diante.

kuhn@utfpr.edu.br | youtube.com/@eduardokuhn87

A adição é a operação aritmética mais importante já que a subtração, multiplicação e divisão usam a adição em suas operações.

Adição e subtração binária

A subtração binária é realizada de forma similar a subtração de números decimais, e.g.,

- $0 - 0 = 0$
- $1 - 0 = 1$
- $1 - 1 = 0$
- $0 - 1 = 10 - 1 = 1$ (1 com empresta 1)

A operação de empresta na subtração ("vem um") ocorre quando se deseja subtrair 1 de 0.

Exemplos:

110 (6)	11011 (27)	1000,10 (8,50)
− 010 (2)	− 01101 (13)	− 0011,01 (3,25)
100 (4)	1110 (14)	101,01 (5,25)

Quando mais de dois números devem ser subtraídos, subtraem-se os dois primeiros e o resultado é subtraído do terceiro número, e assim por diante.

Adição e subtração binária

Exemplo: Determine o resultado de

a)

$$\begin{array}{r} 10110_2 \\ +\ 00111_2 \\ \hline \end{array}$$

b)

$$\begin{array}{r} 011{,}101_2 \\ +\ 010{,}010_2 \\ \hline \end{array}$$

c)

$$\begin{array}{r} 101101_2 \\ -\ 010010_2 \\ \hline \end{array}$$

d)

$$\begin{array}{r} 10101{,}1101_2 \\ -\ 01110{,}0110_2 \\ \hline \end{array}$$

Adição e subtração binária

Exemplo: Determine o resultado de

a)

$$\begin{array}{r} 10110_2 \\ +\ 00111_2 \\ \hline 11101_2 \end{array}$$

b)

$$\begin{array}{r} 011{,}101_2 \\ +\ 010{,}010_2 \\ \hline 101{,}111_2 \end{array}$$

c)

$$\begin{array}{r} 101101_2 \\ -\ 010010_2 \\ \hline 011011_2 \end{array}$$

d)

$$\begin{array}{r} 10101{,}1101_2 \\ -\ 01110{,}0110_2 \\ \hline 00111{,}0111_2 \end{array}$$

Universidade Tecnológica Federal do Paraná
lattes.cnpq.br/2456654064380180
kuhn@utfpr.edu.br | youtube.com/@eduardokuhn87

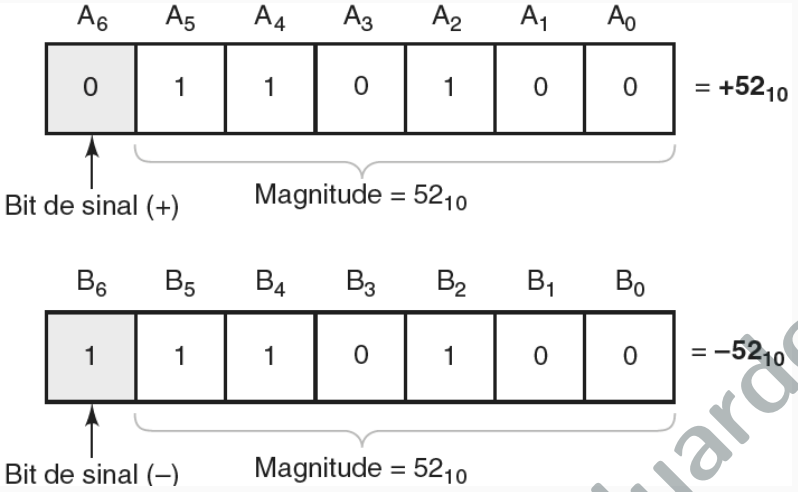
E, como representar números negativos?

E, como representar números negativos?

Existem três formas usuais, i.e.,
sinal-magnitude, complemento de 1, e de 2.

Representação de números com sinal

Forma de sinal-magnitude: ...um bit "de sinal" é **adicionado** ao número na posição MSB.



Especificamente:

- 0 no bit "de sinal" indica número **positivo**; e
- 1 no bit "de sinal" indica número **negativo**.

Com 7 bits, 128 valores podem ser representados; **todavia, agora, "metade" dos valores está no eixo positivo e a outra, no eixo negativo).**

Embora a forma sinal-magnitude seja uma representação direta, computadores normalmente utilizam a forma de complemento de 2.

Representação de números com sinal

Forma de complemento de 1: ...é obtida realizando a operação de "negação" (i.e., inversão).

1 0 1 1 0 1	número binário original
↓ ↓ ↓ ↓ ↓ ↓	
0 1 0 0 1 0	complementa-se cada bit para obter o complemento de 1

Forma de complemento de 2: ...é obtida tomando o complemento de 1 do número e somando 1 na posição do LSB.

	1 0 1 1 0 1	número binário de 45
	0 1 0 0 1 0	complementa-se cada bit para obter o complemento de 1
+	1	adiciona-se 1 para obter o complemento de 2
	0 1 0 0 1 1	complemento de 2 do número binário original

Forma de complemento de 2

Caso o número seja...

- **positivo,** a magnitude é representada na forma binária direta e um bit 0 é adicionado ao MSB; e
- **negativo,** a magnitude é representada na forma do complemento de 2 e um bit 1 é adicionado ao MSB.

Abordagem amplamente utilizada para representar números com sinal já que permite realizar a operação de subtração efetuando, na verdade, uma adição; consequentemente, poupando *hardware*.

Representação usando a forma de complemento de 2

Exemplo: Represente os números decimais como números binários com sinal na forma de complemento de 2, usando um total de 5 bits (incluindo o bit "de sinal").

a) $+13_{10}$

b) $+3_{10}$

c) -9_{10}

d) -8_{10}

Representação usando a forma de complemento de 2

Exemplo: Represente os números decimais como números binários com sinal na forma de complemento de 2, usando um total de 5 bits (incluindo o bit "de sinal").

a) $+13_{10} = 01101_2$

b) $+3_{10} = 00011_2$

c) $-9_{10} = 10111_2$

d) $-8_{10} = 11000_2$

Representação usando a forma de complemento de 2

Exemplo: Determine o valor decimal de cada dos números binários com sinal (de 5 bits) representados na forma de complemento de 2.

a) 01100_2

b) 01000_2

c) 11010_2

d) 10001_2

Representação usando a forma de complemento de 2

Exemplo: Determine o valor decimal de cada dos números binários com sinal (de 5 bits) representados na forma de complemento de 2.

a) $01100_2 = +12_{10}$

b) $01000_2 = +8_{10}$

c) $11010_2 = -6_{10}$

d) $10001_2 = -15_{10}$

Como acomodar um número (na forma de complemento de 2) em um registrador com maior capacidade (# de FFs)?

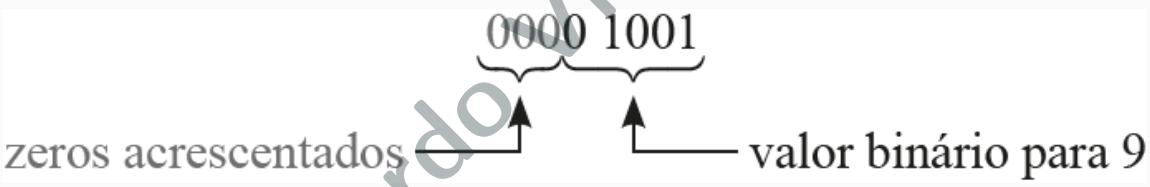
Como acomodar um número (na forma de complemento de 2) em um registrador com maior capacidade (# de FFs)?

Basta realizar a extensão do bit de sinal.

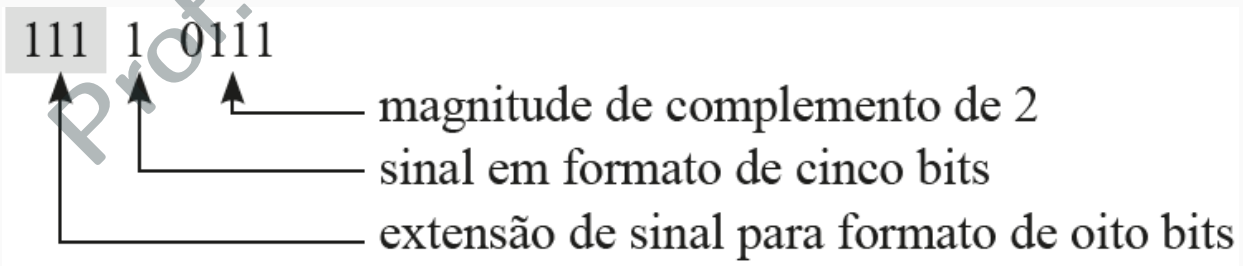
Extensão do bit de sinal

Para armazenar um número de 5 bits em um sistema com registradores de 8 bits (i.e., 7 bits para a magnitude mais o "bit de sinal"), **acrescentam-se**

- **'0s' à esquerda caso o número seja positivo**, i.e.,



- **'1s' à esquerda caso o número seja negativo**, i.e.,



Operação de "negação" (complemento de 2)

A **"negação"** (complemento de 2) é a operação de conversão de

- um **número positivo em seu equivalente negativo**; ou
- um **número negativo em seu equivalente positivo**.

Iniciar com	00001001	+9
Fazer o complemento de 2 (negação)	11110111	–9
Negar novamente	00001001	+9

Quando os números binários com sinal estão representados no sistema de complemento de 2, a "negação" é obtida pela operação de complemento de 2.

Faixa de valores usando a forma de complemento de 2

- **Faixa de valores** que pode ser representada:

  -2^{D-1} até $+(2^{D-1}-1)$

  onde D é o número de bits.

- Note que 2^D valores podem ser representados.

- **Por convenção, assume-se que**

  $$-2^{D-1} = 1000\ldots000_2$$

Valor decimal	Binário com sinal usando complemento de 2
$+7 = 2^3 - 1$	0111
+6	0110
+5	0101
+4	0100
+3	0011
+2	0010
+1	0001
0	0000
–1	1111
–2	1110
–3	1101
–4	1100
–5	1011
–6	1010
–7	1001
$-8 = -2^3$	1000

Faixa de valores usando a forma de complemento de 2

Exemplo: Represente cada um dos seguintes valores como um número com sinal usando 8 bits no sistema de complemento de 2.

a) 0_{10}

b) $+127_{10}$

Exemplo: Determine os equivalentes números decimais para os números binários representados no sistema de complemento de 2.

a) 1000000_2

b) 10000000_2

Faixa de valores usando a forma de complemento de 2

Exemplo: Represente cada um dos seguintes valores como um número com sinal usando 8 bits no sistema de complemento de 2.

a) $0_{10} = 00000000_2$

b) $+127_{10} = 01111111_2$

Exemplo: Determine os equivalentes números decimais para os números binários representados no sistema de complemento de 2.

a) $1000000_2 = -64_{10}$

b) $10000000_2 = -128_{10}$

Representação de números com sinal

Exemplo: Qual é a faixa de valores de números decimais com sinal que pode ser representada com 12 bits (incluindo o bit de sinal)?

Exemplo: Quantos bits são necessários para representar valores decimais na faixa de -50 a $+50$?

Exemplo: Qual é o maior valor decimal negativo que pode ser representado por um número de 2 bytes?

Representação de números com sinal

Exemplo: Qual é a faixa de valores de números decimais com sinal que pode ser representada com 12 bits (incluindo o bit de sinal)?
R: de -2048_{10} até $+2047_{10}$

Exemplo: Quantos bits são necessários para representar valores decimais na faixa de -50 a $+50$?
R: 7 bits

Exemplo: Qual é o maior valor decimal negativo que pode ser representado por um número de 2 bytes?
R: -32768_{10}

Como realizar a adição e a subtração usando o sistema de complemento de 2?

Operações no sistema de complemento de 2

No **sistema de complemento de 2, as operações de adição e subtração** podem ser realizadas apenas com a operação de adição.

- **Adição:** Segue tal como realizado na representação binária convencional.

$$[S] = [A] + [B]$$

- **Subtração:** É realizada usando a operação de adição e a representação de complemento de 2 do segundo operando.

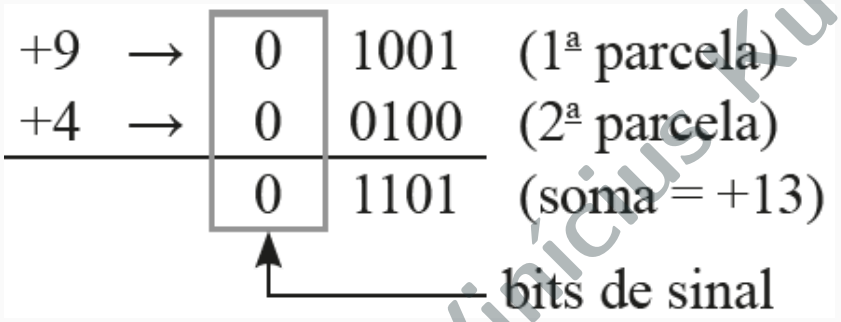
$$\begin{aligned}[S] &= [A] - [B] \\ &= [A] + (-[B]) \\ &= [A] + [\text{complemento de 2 de } B]\end{aligned}$$

A notação $[A]$ é usada para referenciar o conteúdo (de N bits) do registrador A.

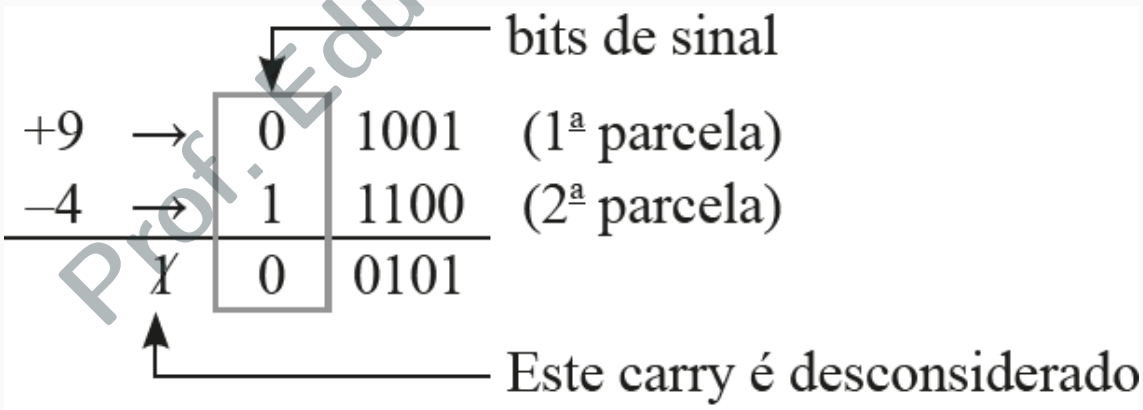
kuhn@utfpr.edu.br | youtube.com/@eduardokuhn87

Adição no sistema de complemento de 2

Caso I: Dois números positivos.

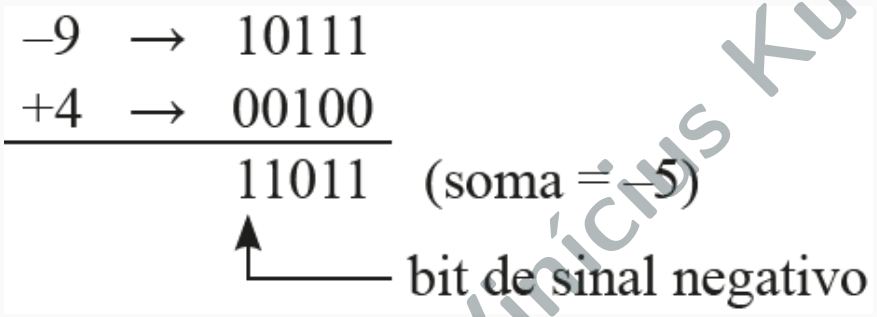


Caso II: Um número positivo e outro menor e negativo.

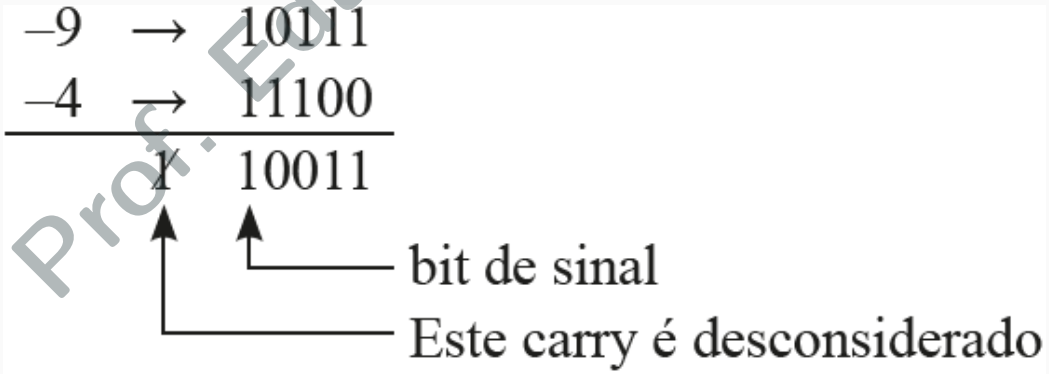


Adição no sistema de complemento de 2

Caso III: Um número positivo e outro maior e negativo.



Caso IV: Dois números negativos.



Adição no sistema de complemento de 2

Caso V: Dois números iguais e de sinais opostos.

OBSERVAÇÕES:

- A mesma operação feita sobre os bits de magnitude é, também, realizada sobre os bits "de sinal".
- Um *carry*, gerado na última posição da soma, é sempre desconsiderado.
- Tanto $[A]$ quanto $[B]$ devem estar devidamente representados usando a forma de complemento de 2.

Operações usando o sistema de complemento de 2

Exemplo: Usando a forma de complemento de 2 com 5 bits, efetue as operações e expresse o resultado como um número binário e seu equivalente decimal.

a) $5 + 10 = 15$

c) $6 - 12 = -6$

b) $6 - 4 = 2$

d) $-12 - 12 = -24$

Operações usando o sistema de complemento de 2

Exemplo: Usando a forma de complemento de 2 com 5 bits, efetue as operações e expresse o resultado como um número binário e seu equivalente decimal.

a) $5 + 10 = 15$

$$\begin{array}{r} 00101_2 \\ +\ 01010_2 \\ \hline 01111_2\ (15_{10}) \end{array}$$

b) $6 - 4 = 2$

$$\begin{array}{r} 00110_2 \\ +\ 11100_2 \\ \hline 00010_2\ (2_{10}) \end{array}$$

c) $6 - 12 = -6$

$$\begin{array}{r} 00110_2 \\ +\ 10100_2 \\ \hline 11010_2\ (-6_{10}) \end{array}$$

d) $-12 - 12 = -24$

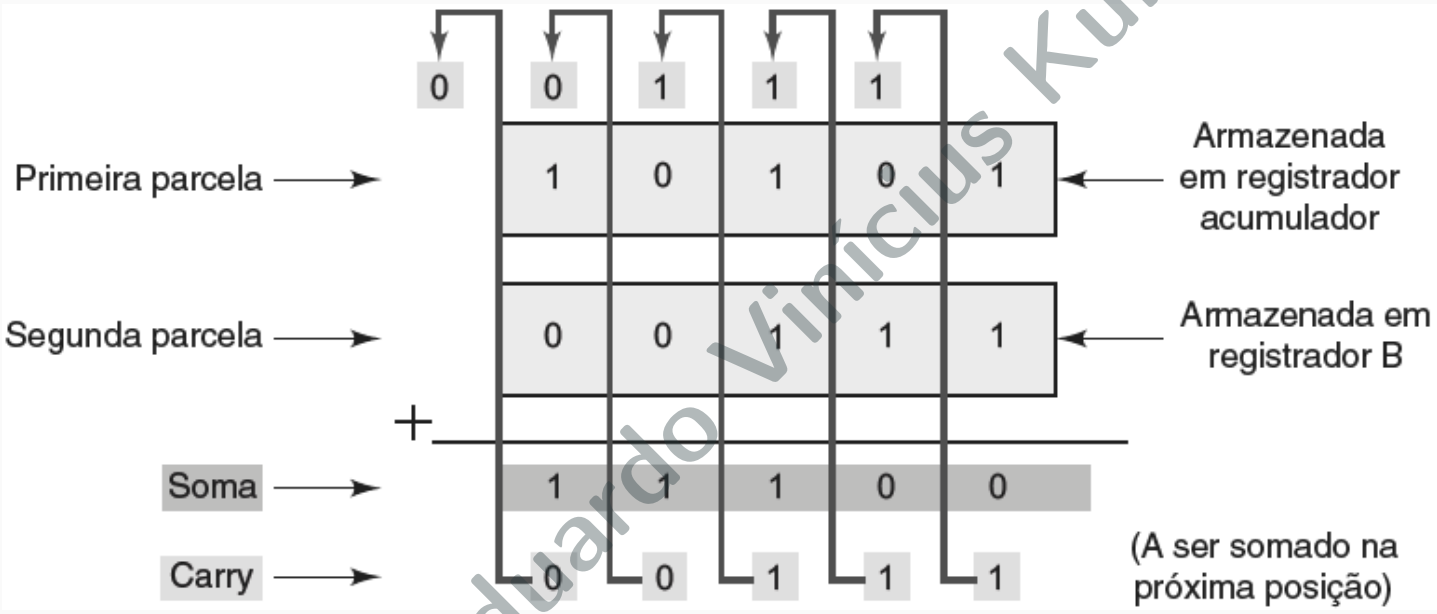
$$\begin{array}{r} 10100_2 \\ +\ 10100_2 \\ \hline \rightarrow 01000_2\ (\textit{overflow}) \end{array}$$

Overflow (transbordamento) ocorre quando o resultado de uma operação excede a capacidade de representação (# bits). E, pode ser detectado por circuitos especiais que informam a unidade de controle (UC).

Como estruturar o projeto de um somador?

Estrutura de um somador binário paralelo

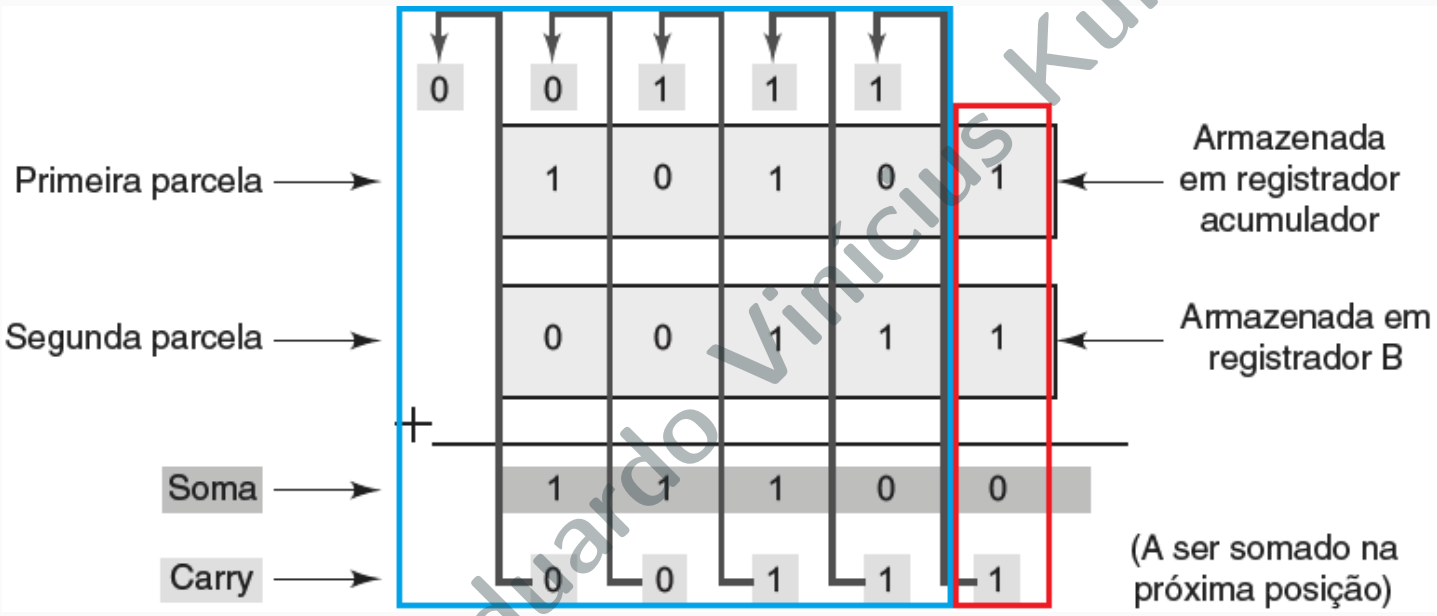
Exemplo: Adição de 10101_2 e 00111_2 (mesmo número de bits).



- Comece pelo LSB.
- Some os bits da posição para gerar soma e *carry*.
- *Carry* é adicionado aos bits da próxima posição.

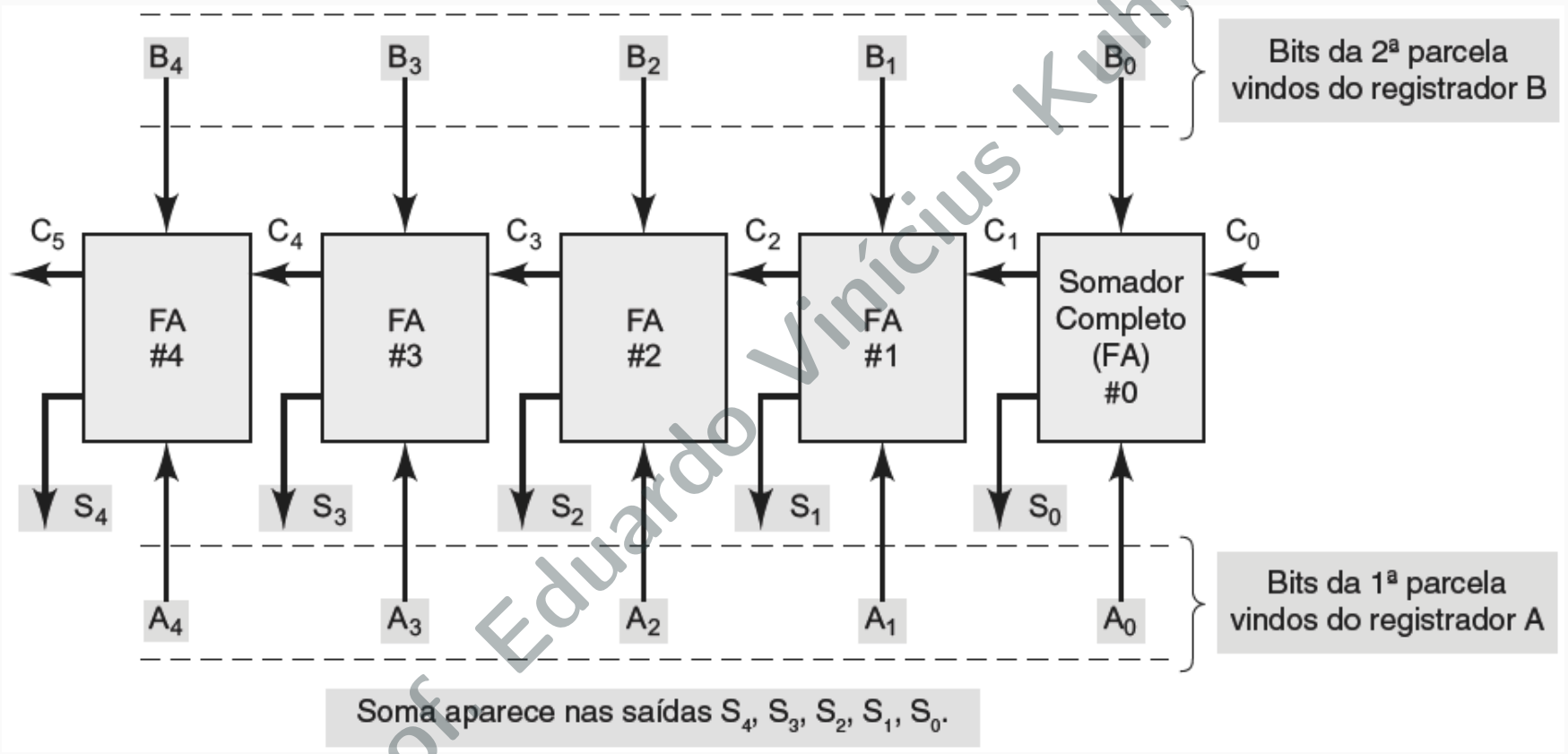
Estrutura de um somador binário paralelo

Exemplo: Adição de 10101_2 e 00111_2 (mesmo número de bits).



- Um "**meio somador**" (***half-adder***) opera com 2 entradas para gerar soma e *carry* como saídas.
- Um "**somador completo**" (***full-adder***) opera com 3 entradas para gerar soma e *carry* como saídas.

Diagrama de blocos de um somador binário paralelo (5 bits)



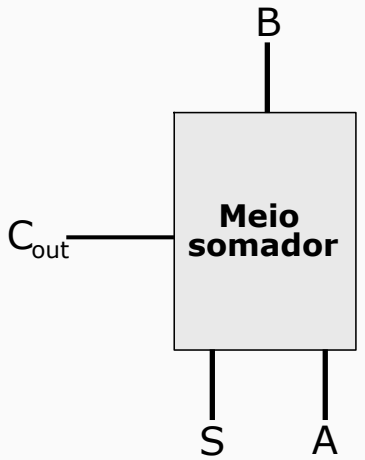
O circuito lógico de uma posição pode ser replicado para todas as posições.

Projeto de um "meio somador" (*half-adder*)

Em um **"meio somador" (*half-adder*)**, tem-se

- **Entradas:** A e B
- **Saídas:** S e C_{out}

A	B	S	C_{out}
0	0		
0	1		
1	0		
1	1		



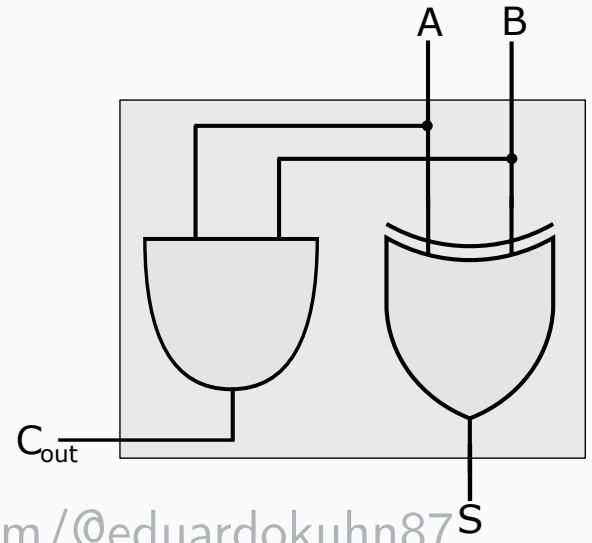
Circuito de um "meio somador" (*half-adder*)

Em um **"meio somador" (*half-adder*)**, tem-se

- **Entradas:** A e B
- **Saídas:** S e C_{out}

$$\begin{cases} S = A \oplus B \\ C_{\text{out}} = AB \end{cases}$$

A	B	S	C_{out}
0	0	**0**	**0**
0	1	**1**	**0**
1	0	**1**	**0**
1	1	**0**	**1**

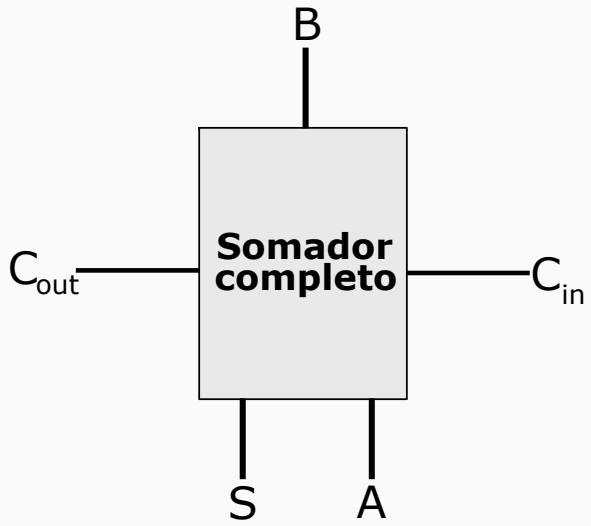


Projeto de um "somador completo" (*full-adder*)

Em um **"somador completo" (*full-adder*)**, tem-se

- **Entradas:** A, B e C_{in}
- **Saídas:** S e C_{out}

A	B	C_{in}	S	C_{out}
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



Projeto de um "somador completo" (*full-adder*)

Em um **"somador completo" (*full-adder*)**, tem-se

- **Entradas:** A, B e C_{in}
- **Saídas:** S e C_{out}

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

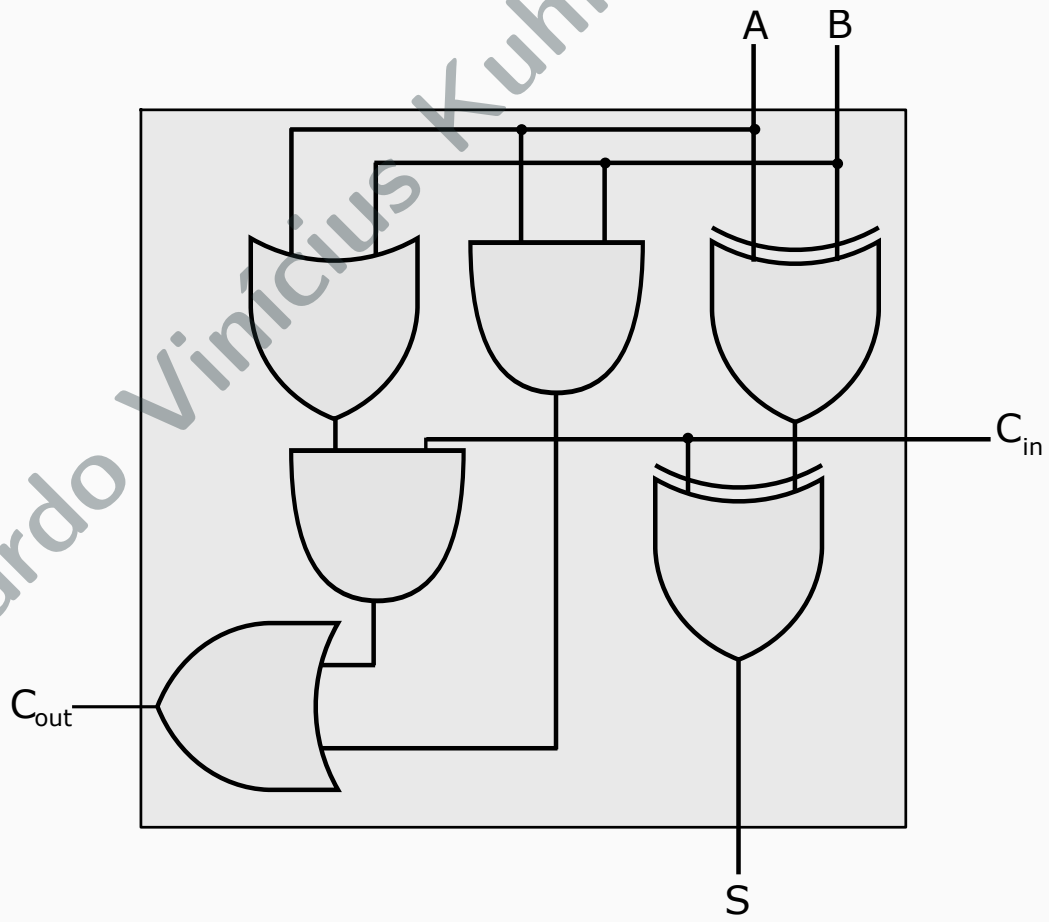
Expressões lógicas:

$$\begin{cases} S = (A \oplus B) \oplus C_{\text{in}} \\ C_{\text{out}} = (A + B)C_{\text{in}} + AB \end{cases}$$

Circuito de um "somador completo" (*full-adder*)

Expressões lógicas:

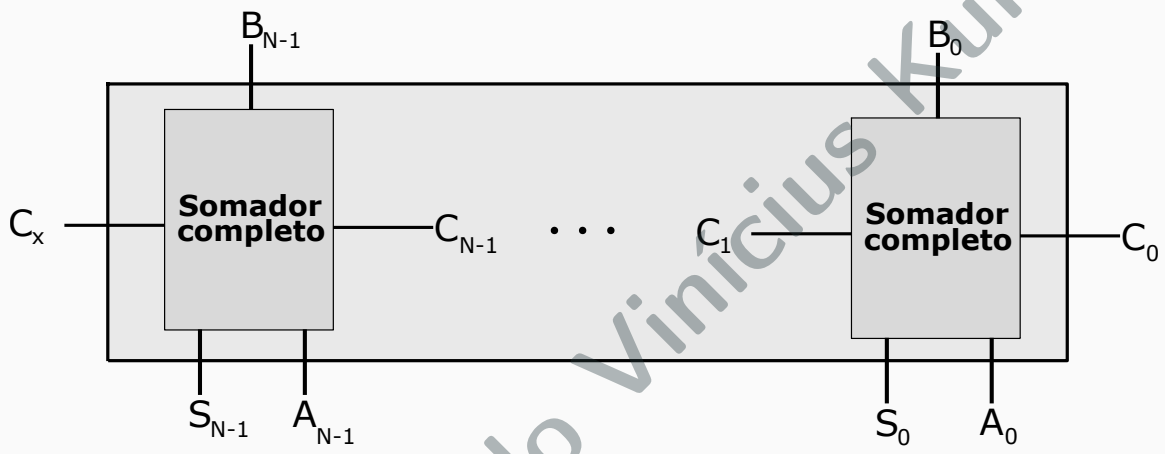
$$\begin{cases} S = (A \oplus B) \oplus C_{\text{in}} \\ C_{\text{out}} = (A + B)C_{\text{in}} + AB \end{cases}$$



Como implementar um somador de N bits?

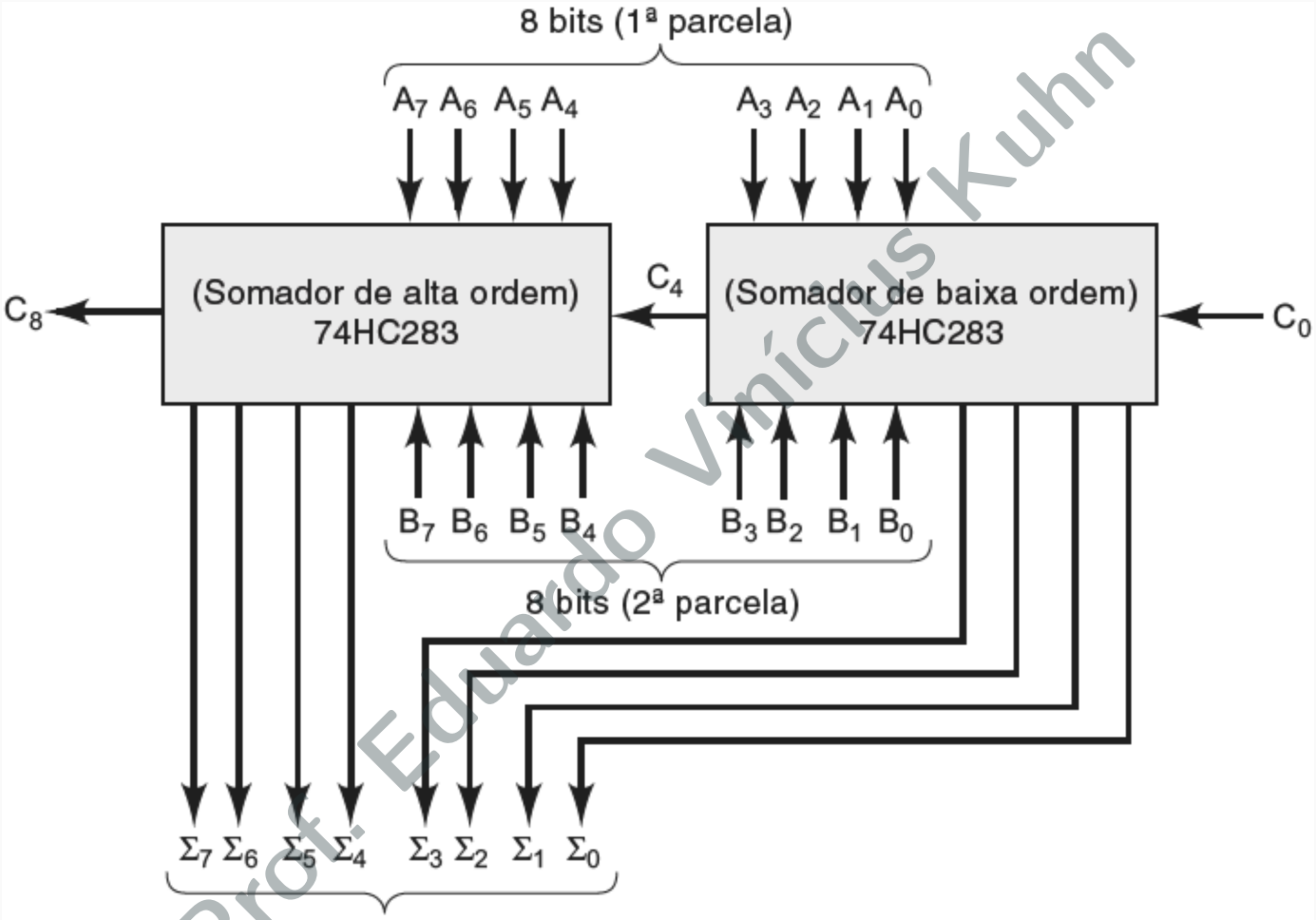
Implementação de um somador de N bits?

Diagrama de blocos de um somador de N bits:



Maior nível de abstração:

Somador paralelo de 8 bits usando somadores de 4 bits



Vale reforçar que o último *carry* (i.e., C_8) deve ser desconsiderado.

kuhn@utfpr.edu.br | youtube.com/@eduardokuhn87

Como combinar ambas as operações (adição e subtração) em um único bloco?

Adição e subtração combinadas

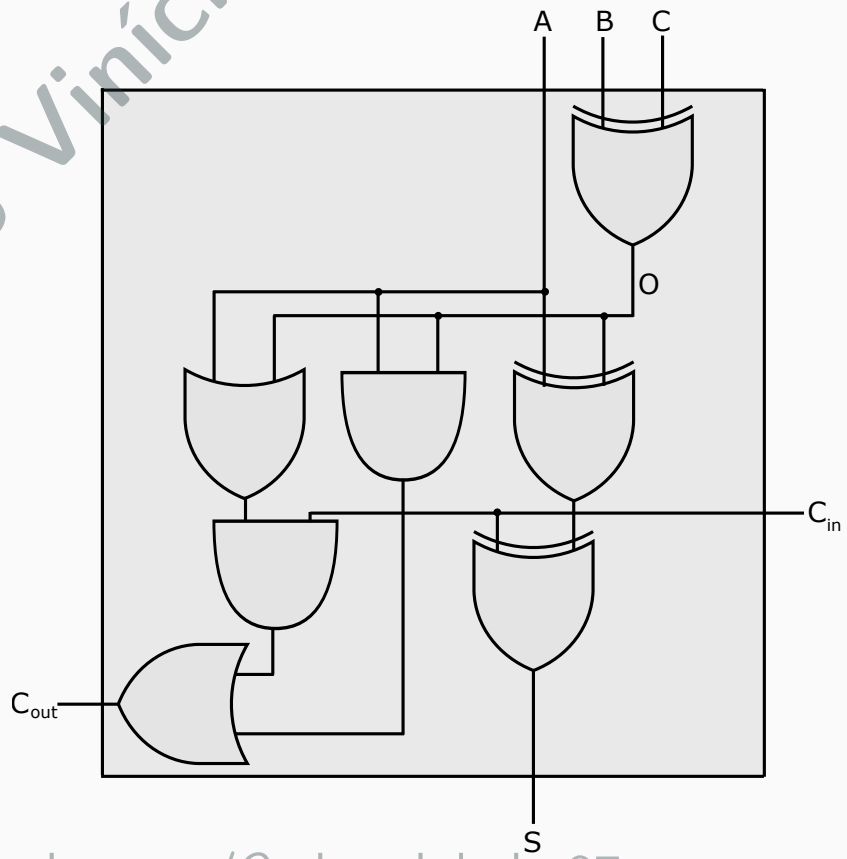
Visando **incorporar ambas as operações**, considera-se

- **Entradas:** B e C ($C = 0 \rightarrow +$ e $C = 1 \rightarrow -$)
- **Saídas:** O

C	B	O
0	0	0
0	1	1
1	0	1
1	1	0

Expressão lógica:

$$O = C \oplus B$$



(XOR é o inversor controlado)

kuhn@utfpr.edu.br | youtube.com/@eduardokuhn87

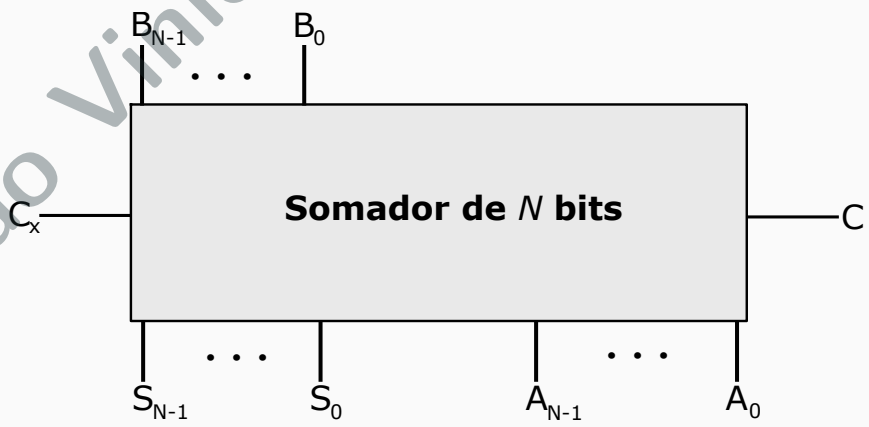
Adição e subtração combinadas

Visando **incorporar ambas as operações**, considera-se

- **Entradas:** B e C ($C = 0 \rightarrow +$ e $C = 1 \rightarrow -$)
- **Saídas:** O

Lógica de operação:

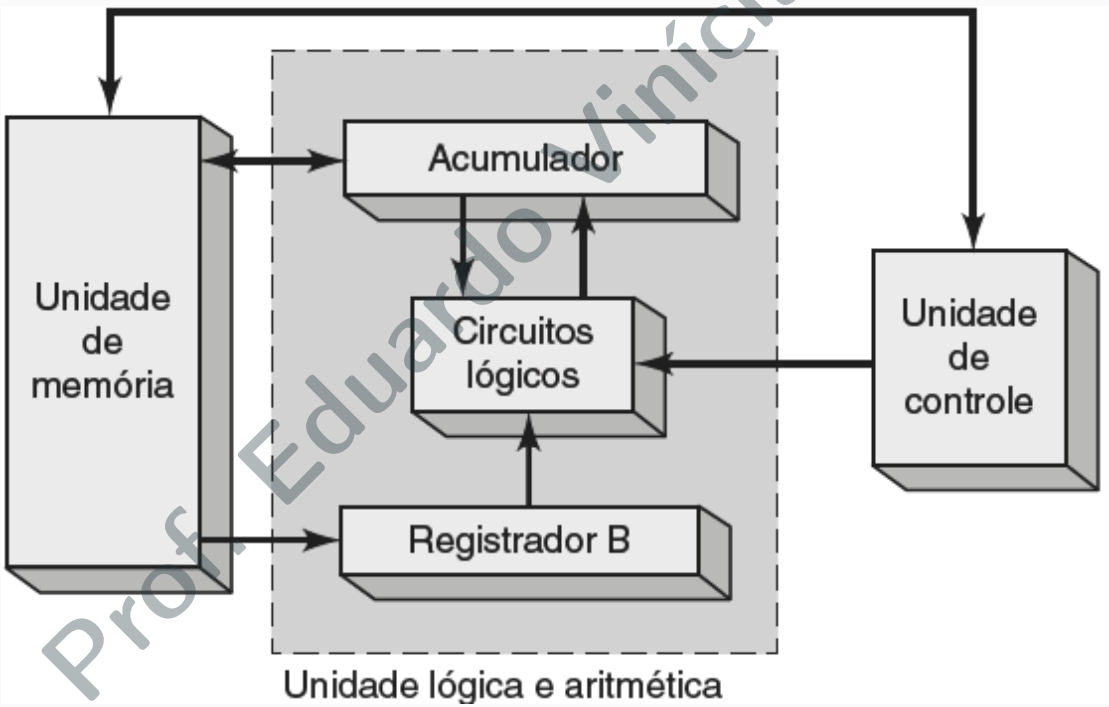
$$\begin{cases} C = 0 & \rightarrow & S = A + B \\ C = 1 & \rightarrow & S = A - B \end{cases}$$



Vale reforçar, mais uma vez, que tanto $[A]$ quanto $[B]$ devem estar devidamente representados usando o sistema de complemento de 2 (i.e., cuidado para não "negar" $[B]$ duas vezes erroneamente).

Circuitos aritméticos

A **unidade lógica e aritmética (ULA)** contém o circuito necessário para a realização de operações lógicas e aritméticas com os números binários armazenados na memória.

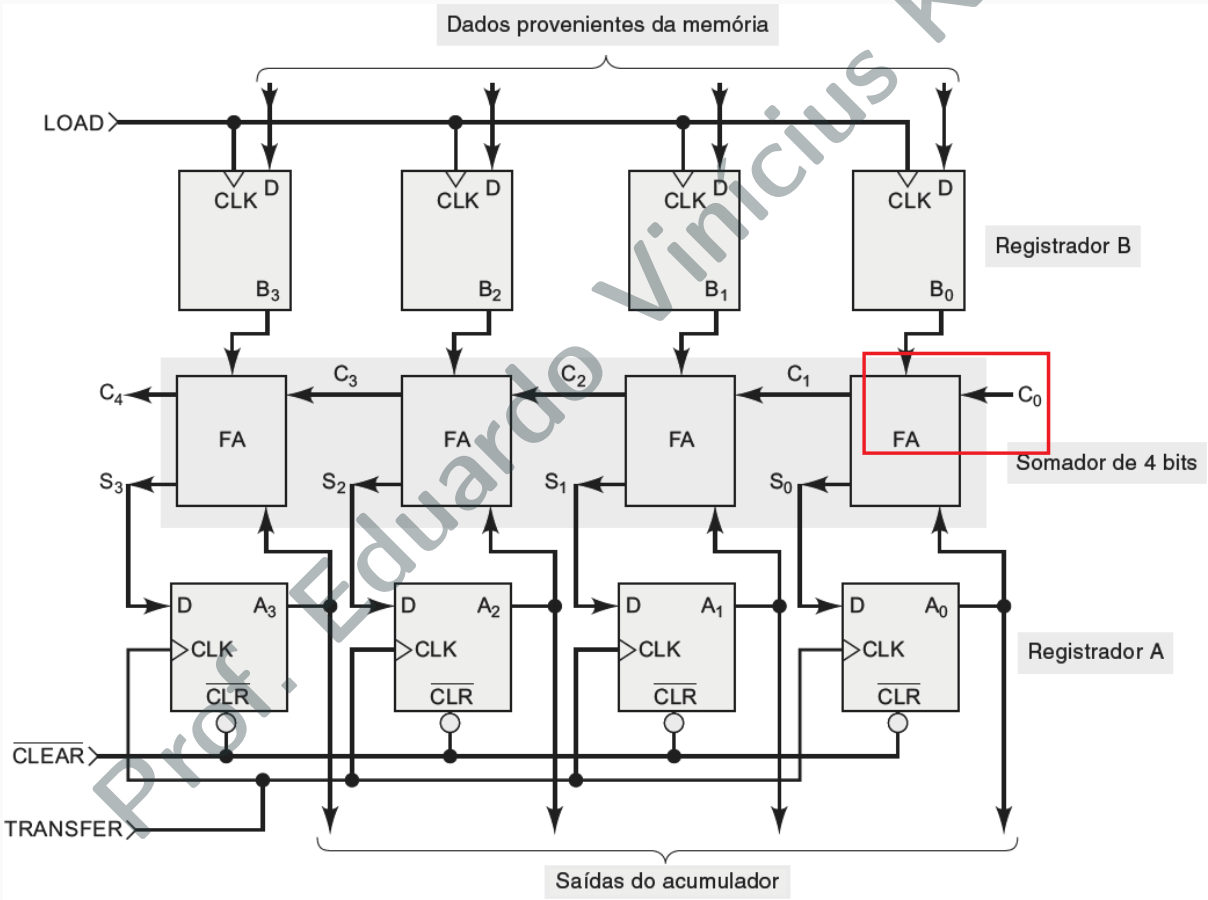


O tamanho de um registrador (# de FFs) define o número de dígitos binários (bits) armazenados para cada número (e.g., 4, 8, 16, 32 ou 64 bits).

Diversos CIs de ULAs estão disponíveis e podem ser usados para se realizar uma ampla faixa de operações lógicas e aritméticas sobre dois números.

Acumulador (de 4 bits)

O **registrador acumulador de uma ULA** armazena um dos números utilizados na operação assim como o resultado.



A soma em posições posteriores depende do *carry* gerado em posições anteriores.

Conforme o número de bits aumenta, o atraso de propagação do *carry* limita a velocidade dos somadores; para contornar isso, um circuito de *carry* antecipado pode ser incorporado.

Sugestão de leitura: Seções 6.5 e 6.6, as quais tratam sobre a multiplicação e a divisão envolvendo números binários.

Resumo

- Para representar números binários com sinal, **um bit "de sinal" é anexado como o MSB** (0 indica $+$ e 1 indica $-$).
- A manipulação de números binários (com sinal) **é mais conveniente usando o sistema de complemento de 2.**
  - O complemento de 2 de um número é obtido realizando-se a "negação" de cada bit e somando-se 1 ao resultado.
  - Números positivos são representados por um bit "de sinal" 0, seguido pelos bits de magnitude em sua forma binária direta.
  - Números negativos são representados por um bit "de sinal" 1, seguido pela magnitude na forma do complemento de 2.
  - Realizar o complemento de 2 sobre um número com sinal produz um número de mesmo valor, mas com sinal trocado.

Resumo

- A **operação aritmética de adição** envolvendo números binários é direta (bit-a-bit).
- A **subtração** pode ser realizada usando a operação de adição e a representação de complemento de 2 do segundo operando.
- **Um somador/subtrator binário (paralelo) de N-bits é construído conectando-se "somadores completos" em cascata.**
  - A construção dos blocos de um "meio somador" e de um "somador completo" foi discutida.
  - Basicamente, um somador completo realiza a adição de dois bits mais um *carry* de entrada.
- **A ULA contém circuitos para operar sobre e manipular os valores armazenados nos registradores.**

Considerações finais

Exercícios sugeridos:

6.1(a)-(c) e (j)-(l), 6.2(a)-(d), 6.3(a)-(d), 6.4, 6.5, 6.9(d) e (e), 6.10(c) e (d) e 6.20

de R.J. Tocci, N.S. Widmer, G.L. Moss, *Sistemas digitais: princípios e aplicações,* 12a ed., São Paulo: Pearson, 2019. $\longrightarrow$ (Capítulo 6)

Para a próxima aula:

R.J. Tocci, N.S. Widmer, G.L. Moss, *Sistemas digitais: princípios e aplicações,* 12a ed., São Paulo: Pearson, 2019. $\longrightarrow$ (Capítulo 7)

Até a próxima aula... =)