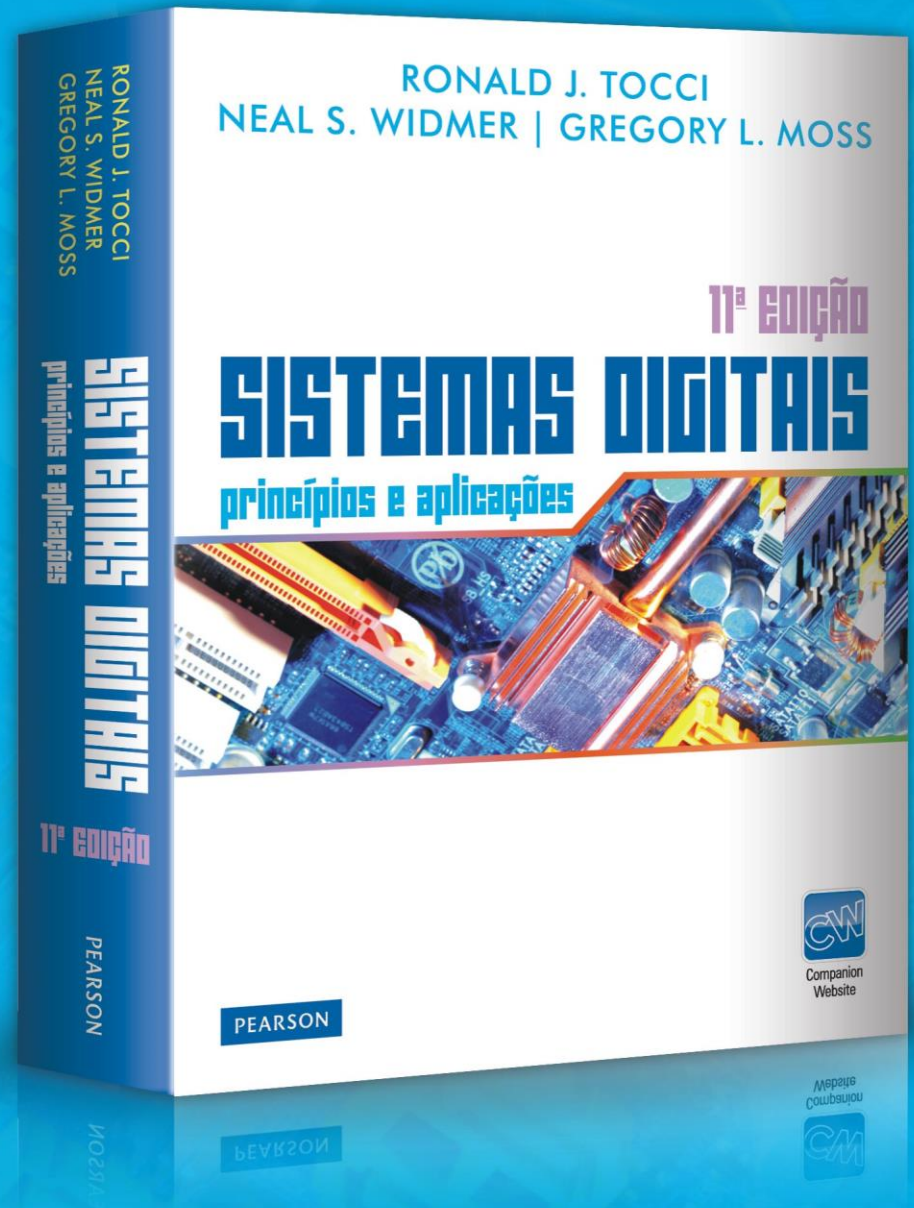


Capítulo 6

Aritmética Digital: Operações e Circuitos



Temas abordados nesse capítulo:

- Adição, subtração, multiplicação e divisão de dois binários.
- Diferença entre soma binária e soma OR.
- Vantagens e desvantagens entre três sistemas diferentes de representação de números binários com sinal.
- Descrição de operações básicas de uma unidade lógica e aritmética.
- Operação do circuito somador ou subtrador paralelo.
- Circuito integrado de ALU para realização de operações lógicas e aritméticas sobre dados de entrada.
- Funções de bibliotecas digitais para implementação de circuitos mais complexos.
- Descrição de equações booleanas para execução de operações em conjuntos inteiros de bits.

6.1 Adição e Subtração Binárias

- Os números binários são somados como os decimais:
 - Para decimais, quando um número soma mais do que 9, um carry (vai um) é resultante.
 - Para binários, quando números somam mais que 1, um carry (vai um) assume a posição.
- A adição é a operação aritmética básica utilizada pelos dispositivos digitais tanto para subtração como multiplicação e divisão.

6.1 Adição e Subtração Binárias

- A subtração binária é efetuada exatamente como a subtração de números decimais.

São quatro as possíveis situações ao se subtrair um bit de outro em qualquer posição de um número binário.

$$0 - 0 = 0$$

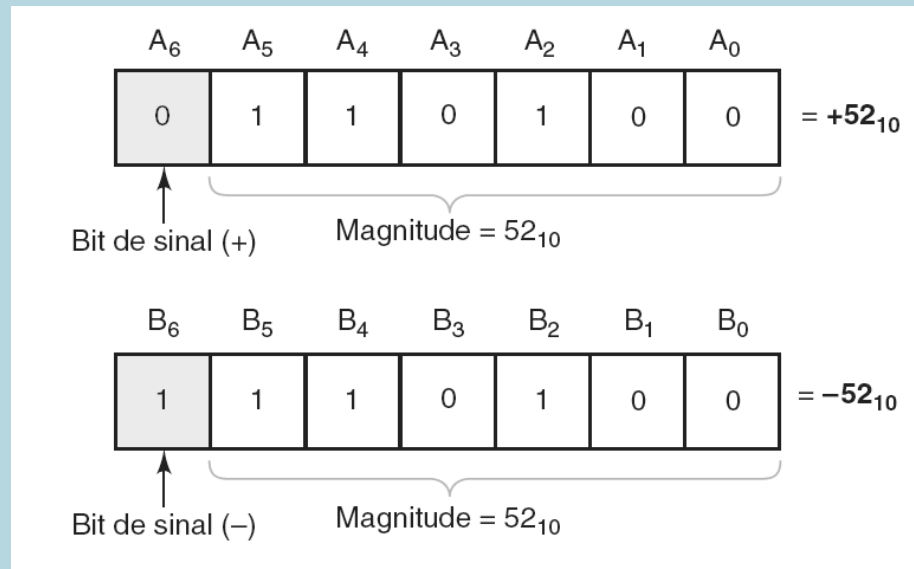
$$1 - 1 = 0$$

$$1 - 0 = 1$$

$$0 - 1 \rightarrow \text{precisa tomar emprestado} \rightarrow 10 - 1 = 1$$

6.2 Representação de Números com Sinal

- Como só é possível mostrar a magnitude com um número binário, o sinal (+) ou (-) é mostrado pela soma de um “bit de sinal” extra.
Um 0 no bit de sinal indica um número positivo.
Um 1 no bit de sinal indica um número negativo.



6.2 Representação de Números com Sinal

- O sistema de complemento de 2 é a forma mais comum usada para representar números binários com sinal.
- Para se determinar o complemento de 2 para um número binário, este deve ser inicialmente transformado para o complemento de 1:
 - Substitui-se cada bit do binário por seu complemento (oposto).
 - Somando-se 1 na posição do bit menos significativo do complemento de 1 este será convertido para seu complemento de 2.
- Negação é a operação de conversão de um número positivo em seu equivalente negativo ou de um número negativo em seu equivalente positivo. Quando os números binários com sinal estão representados no sistema de complemento de 2, a negação é obtida pela operação do complemento de 2.

6.3 Adição no Sistema de Complemento de 2

- Executa-se a soma binária normal de magnitudes.
Os bits de sinal são somados com os bits de magnitude.
- Se a soma resulta em um carry de bit de sinal, ele é desconsiderado.
 - Caso o resultado seja positivo, será em formato de binário puro.
 - Se for negativo, será em forma de complemento de 2.

6.4 Subtração no Sistema de Complemento de 2

- A subtração utilizando o sistema de complemento de 2 envolve, na verdade, a operação de adição.

Realiza-se a operação de negação do subtraendo (o 1º binário).

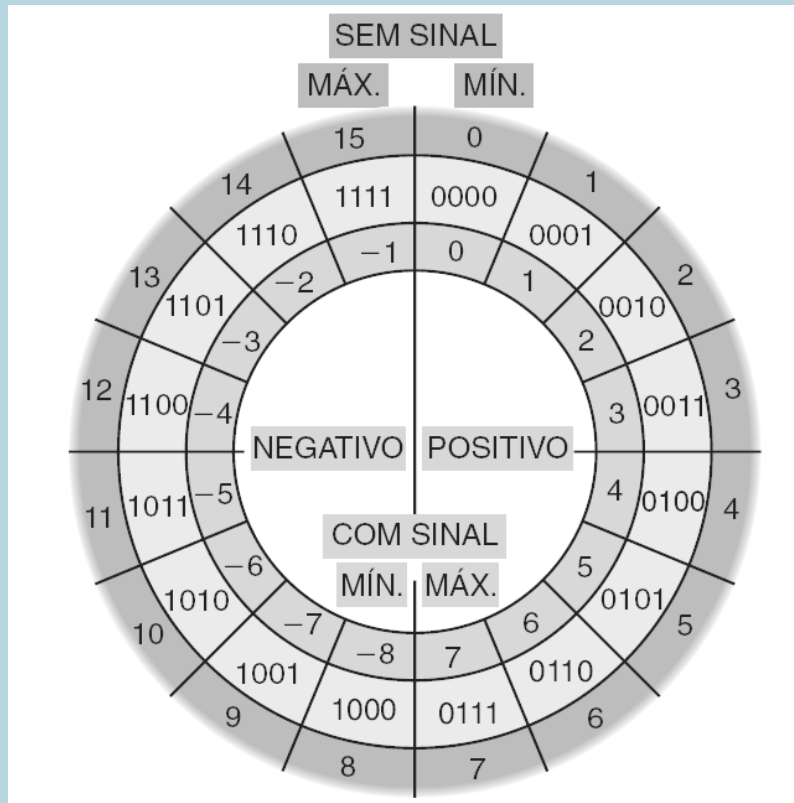
Adiciona-se o número obtido ao minuendo (o 2º binário envolvido).

A resposta representará a diferença entre ambos.

- Quando dois números positivos ou negativos são somados e sempre produzem um resultado incorreto, há a ocorrência de um *overflow* (transbordamento). Se a resposta exceder o número de bits de magnitude, há um overflow nos resultados

6.4 Subtração no Sistema de Complemento de 2

Uso de um círculo de números:



Para a adição, inicia-se no valor da 1ª parcela, avançando-se ao redor do círculo, no sentido horário, pelo número de casas da segunda parcela.

A forma mais natural para a subtração é movendo-se ao redor do círculo, no sentido anti-horário.

Qualquer operação de subtração entre números de quatro bits de sinal oposto produzindo um resultado superior a 7 ou menor do que -8 é um *overflow*.

6.5 Multiplicação de Números Binários

- É similar à multiplicação dos números decimais.
Cada bit no multiplicador é multiplicado pelo multiplicando.
- Os resultados são deslocados à medida que avançamos do LSB para o MSB no multiplicador.
Todos os produtos parciais são somados para geração do produto final.

6.6 Divisão Binária

- É similar à divisão entre decimais, mas mais simples, pois quando verificamos quantas vezes o divisor ‘cabe’ dentro do dividendo, somente 1 ou 0 são possíveis.
- A subtração que faz parte da operação de divisão normalmente é feita usando o complemento de 2.
 - Se os sinais do dividendo e do divisor forem os mesmos, a resposta será positiva.
 - Em caso contrário, a resposta será negativa.

6.7 Adição BCD

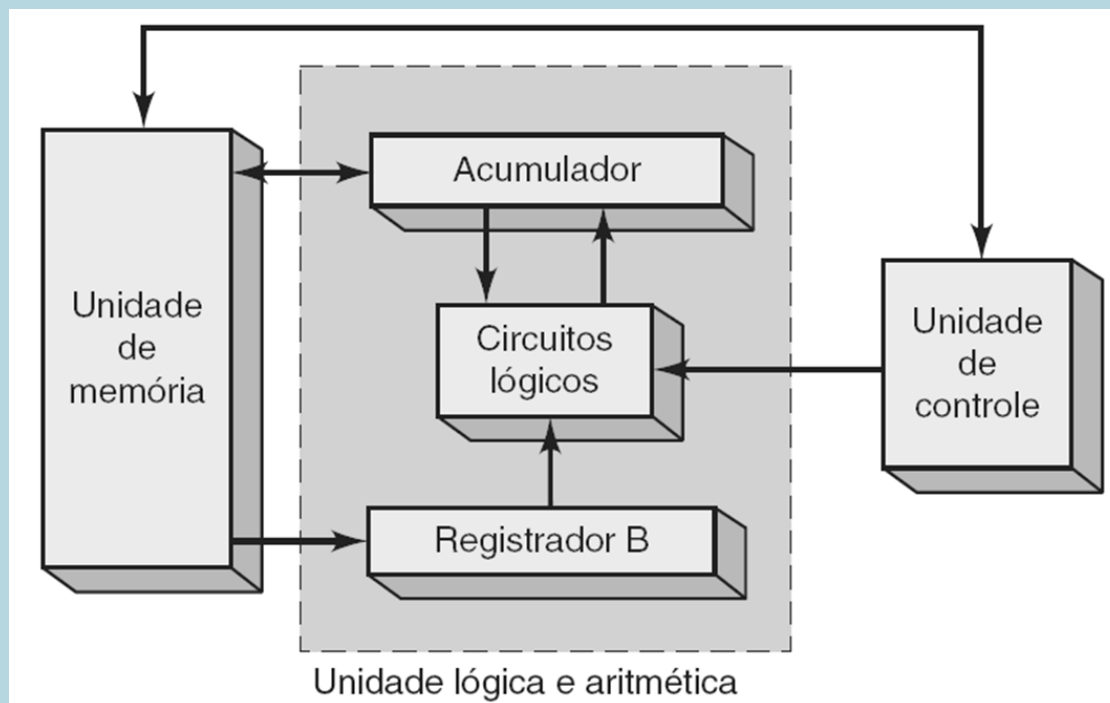
- Quando a soma de cada dígito decimal for menor que 9, a operação será feita como a adição binária normal.
- Quando a soma de cada dígito decimal for maior que 9, um 6 (binário) deverá ser adicionado.
Com essa adição, um carry (vai um) sempre será gerado.

6.8 Aritmética Hexadecimal

- Adição:
 - Somam-se os dígitos hex em decimais.
 - Se a soma for 15, ou menos, expressa-se diretamente em dígitos hex.
 - Se maior que 15, subtrai-se 16 com um carry 1 para a próxima posição.
- Subtração:
 - Usa-se o mesmo método utilizado para os números binários.
- Para um dado com valor negativo, o bit de sinal do número binário será 1, o que sempre tornará o MSD do número hexa maior ou igual a 8.
- Quando o dado tem valor positivo, o bit de sinal será 0, sendo o MSD do número hexa menor ou igual a 7.

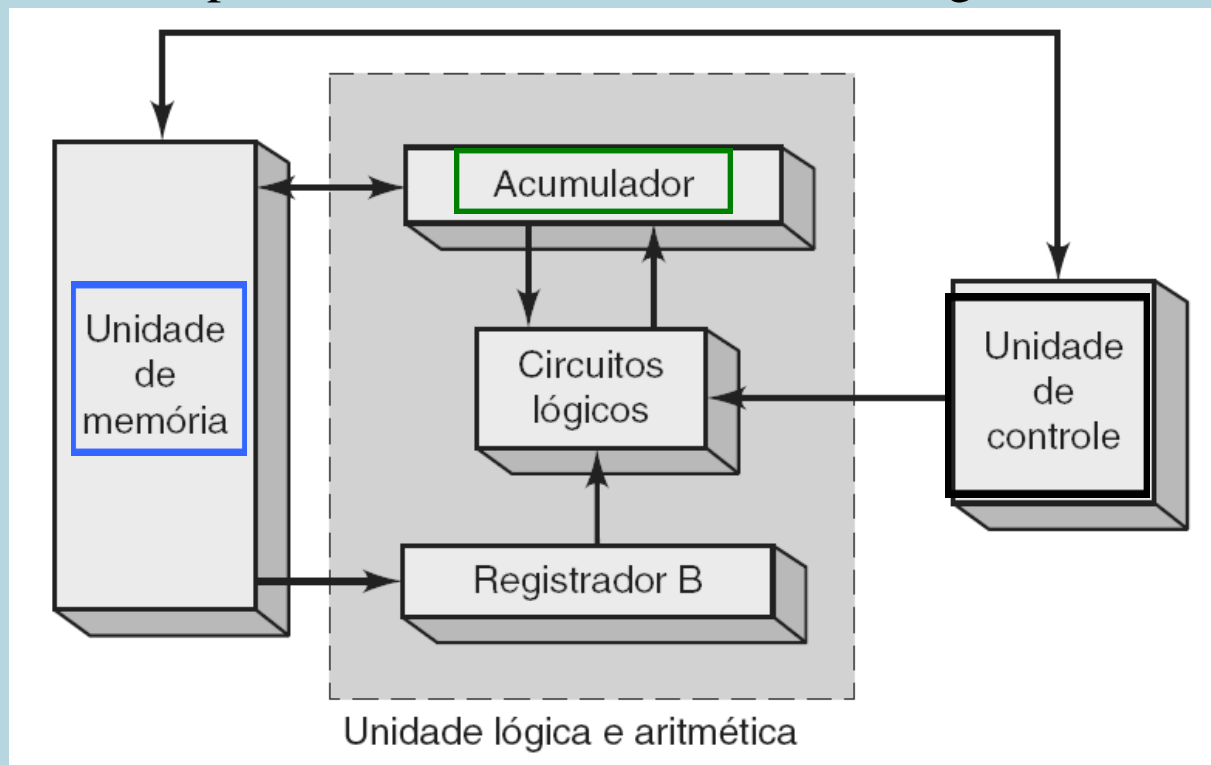
6.9 Circuitos Aritméticos

- Uma unidade lógica e aritmética (ULA) recebe os dados armazenados na memória e executa operações aritméticas e lógicas com instruções provenientes da unidade de controle.



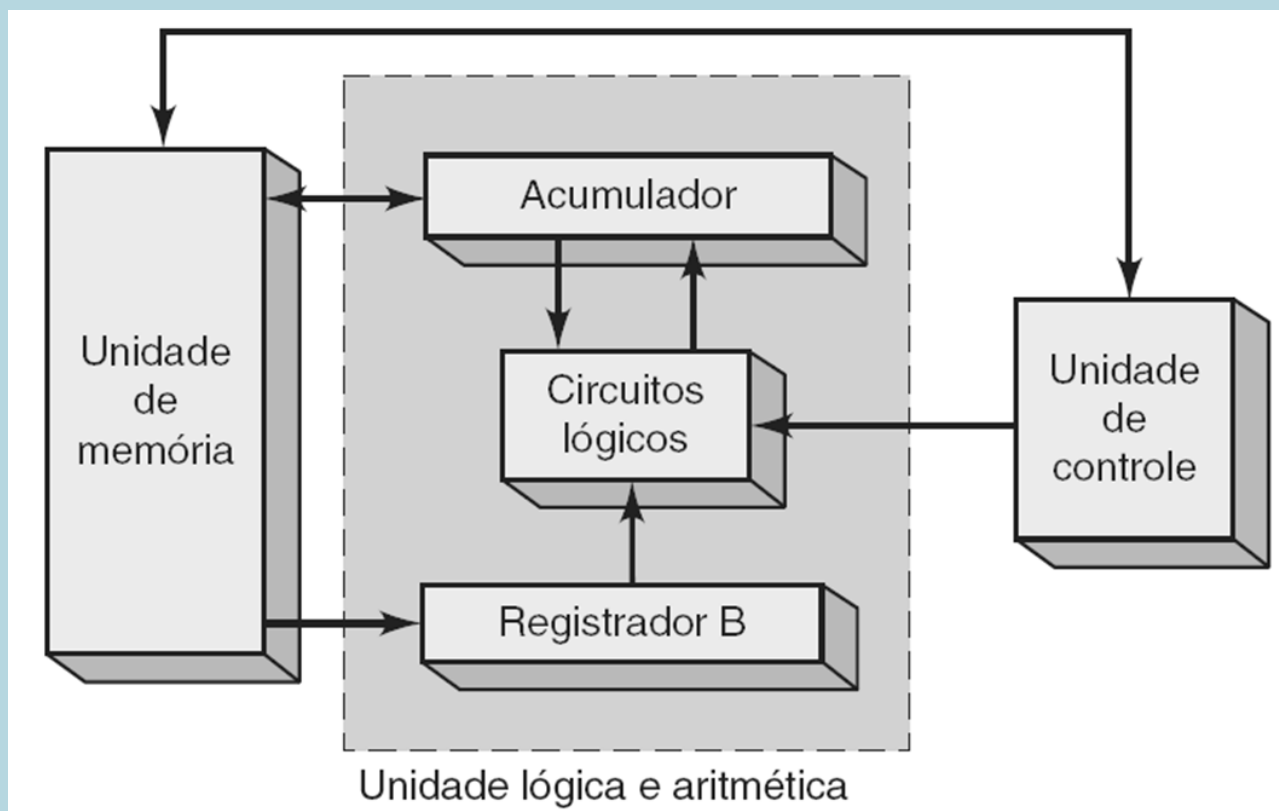
6.9 Circuitos Aritméticos

- A unidade de controle é instruída a adicionar um número específico de um local da memória para um número armazenado no registrador acumulador.



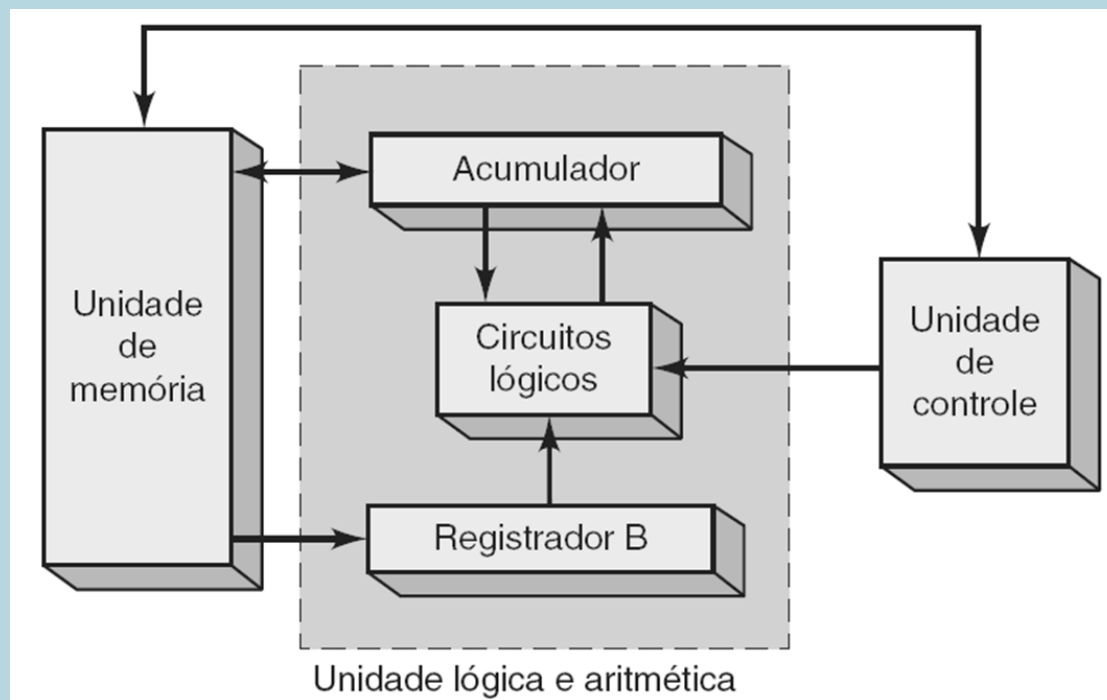
6.9 Circuitos Aritméticos

- O número é transferido da memória para o registrador B.



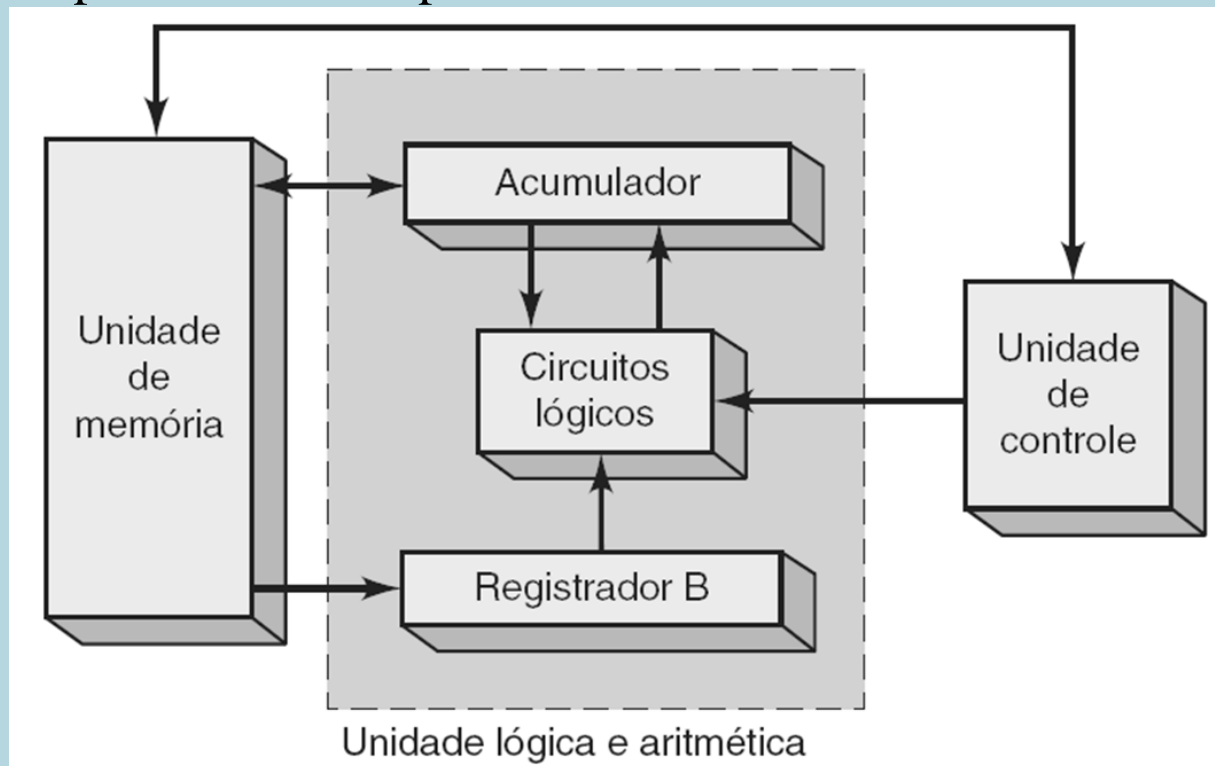
6.9 Circuitos Aritméticos

- O número no registrador B e o número do registrador acumulador são somados no circuito lógico e o resultado é enviado para o acumulador para ser armazenado.



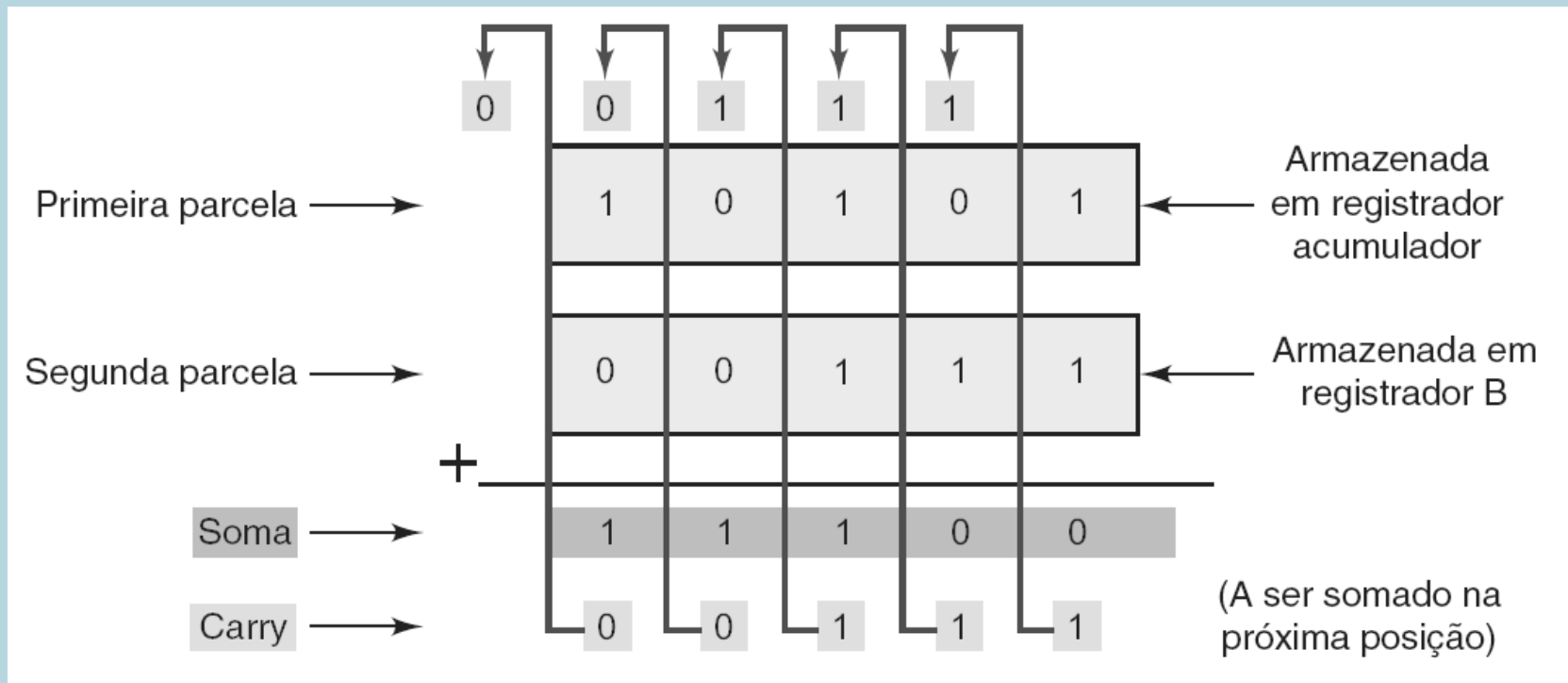
6.9 Circuitos Aritméticos

- O novo número permanece no acumulador para outras operações ou pode ser transferido para a memória para ser armazenado.



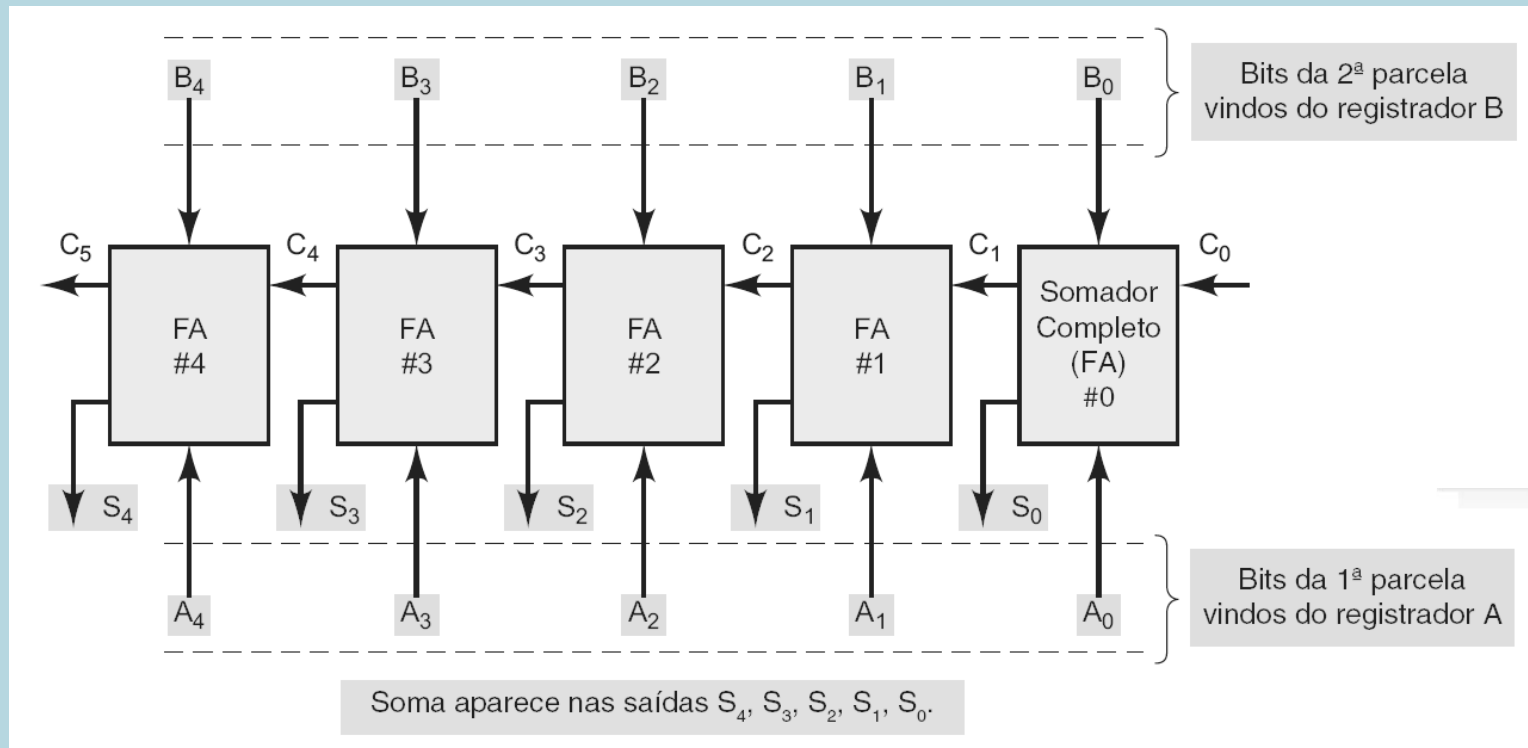
6.10 Somador Binário Paralelo

- Computadores e calculadoras realizam operações de adição sobre dois números de cada vez em que cada número binário pode ter vários dígitos binários



6.10 Somador Binário Paralelo

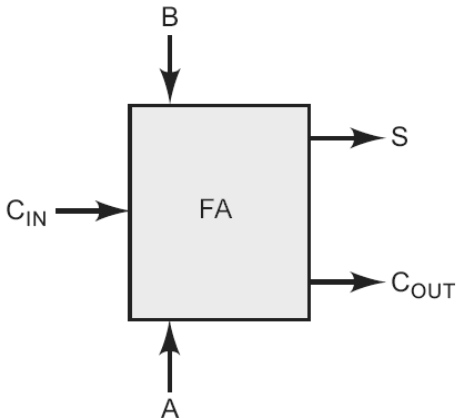
Diagrama em blocos de um circuito somador paralelo de cinco bits utilizando somadores completos.



6.11 Projeto de um Somador Completo

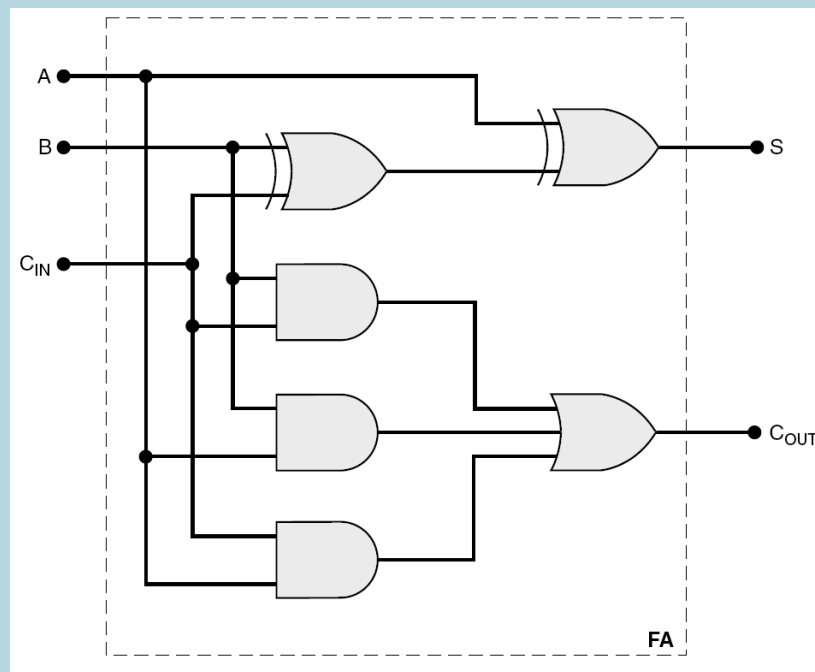
- Construindo uma tabela-verdade com:
 - 3 entradas (2 números para serem adicionados e carry in).
 - 2 saídas (soma e carry out).

Entradas de bits da primeira parcela	Entradas de bits da segunda parcela	Entradas de bits do carry	Saída de bits da soma	Saída de bits do carry
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



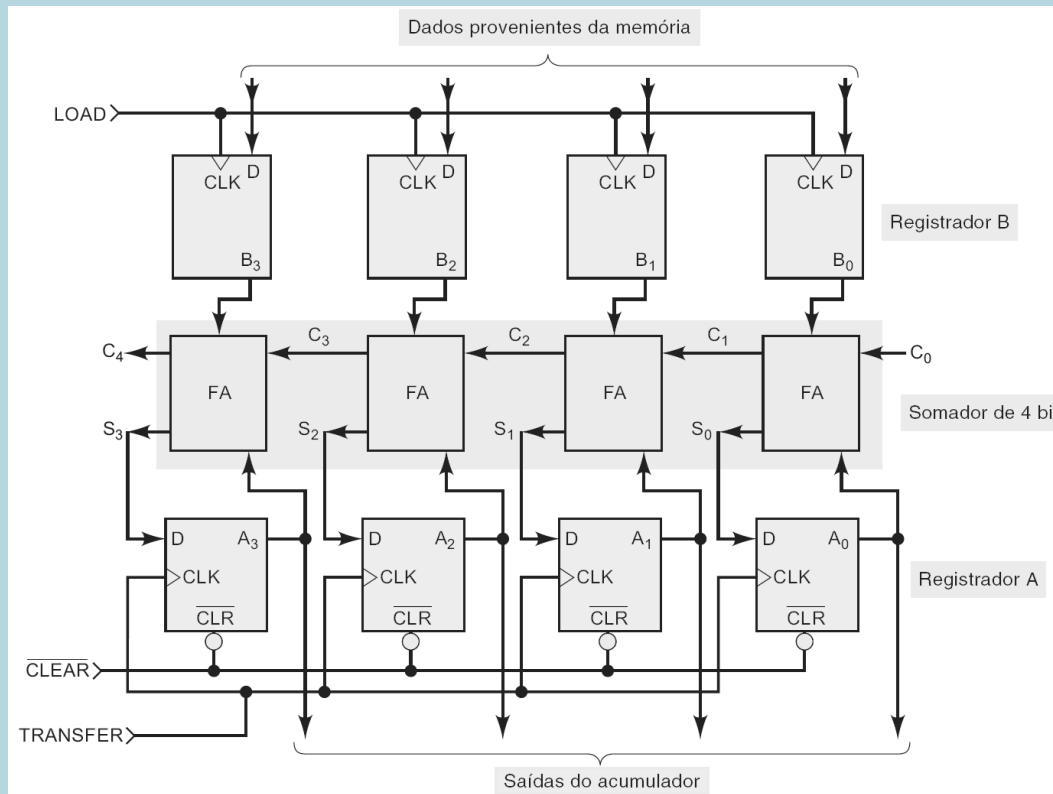
6.11 Projeto de um Somador Completo

- Usando métodos algébricos ou mapas K para simplificar a forma SOP resultante, o circuito completo com as entradas A, B e C_{IN} e as saídas S e C_{OUT} representará o somador completo (FA).



6.12 Somador Paralelo Completo com Registradores

Circuito somador paralelo de quatro bits, incluindo os registradores de armazenamento.



6.12 Somador Paralelo Completo com Registradores

- Processo pelo qual o circuito anterior soma os binários 1001 e 0101:
 - Um pulso CLR será aplicado nas entradas assíncronas, no instante t_1 .
 - O número binário 1001 será transferido da memória para o registrador B, em t_2 .
 - .
 - A soma do 1001 e 0000 será transferida para o registrador A, em t_3 .
 - O número binário 0101 será transferido da memória para o registrador B, em t_4 .
 - .
 - Os resultados da soma serão transferidos para o registrador A, em t_5 .
 - A soma dos dois números estará presente no acumulador.

6.12 Somador Paralelo Completo com Registradores

- Colchetes indicam o conteúdo de um registrador.
Por exemplo: $[A] = 1011$ equivale a $A_3 = 1, A_2 = 0, A_1 = 1, A_0 = 1$
Pode-se pensar em $[A]$ como o conteúdo do registrador A.
- A transferência de dados para um registrador ou a partir dele pode ser indicada com uma seta:

$$[B] \rightarrow [A]$$

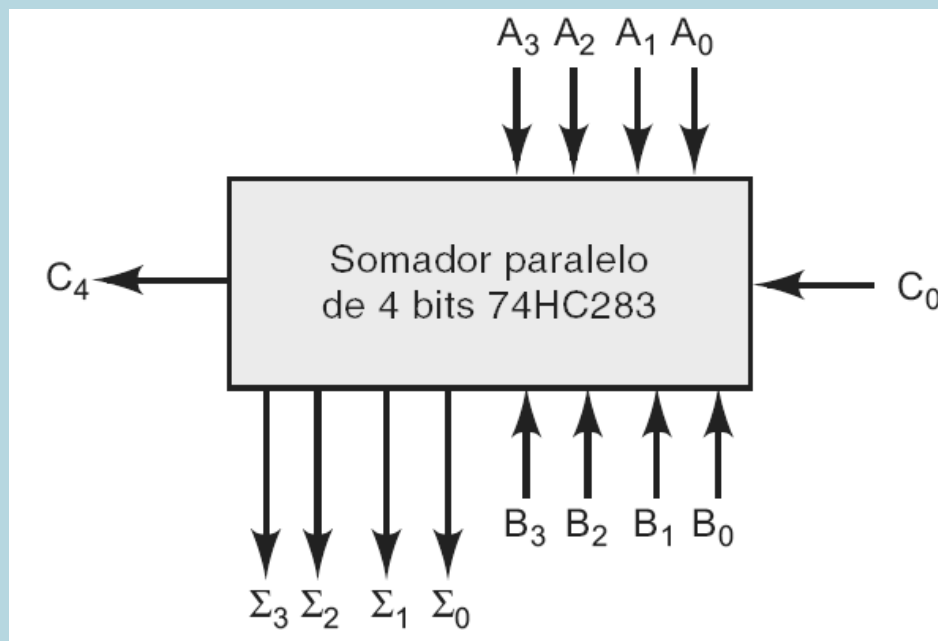
(O conteúdo do registrador B foi transferido para o A)

6.13 Propagação do Carry

- A velocidade de um somador paralelo é limitada por um efeito denominado propagação do carry (ou *carry ripple*):
 - A adição dos bits da 1ª posição gera um carry para a posição seguinte.
 - O último carry gera outro para a última posição (MSB).
 - A soma do bit gerado na última posição depende do carry que foi gerado na adição da primeira posição (LSB).
- O esquema do carry antecipado (ou *look-ahead carry*) é usado em dispositivos de alta velocidade para reduzir o atraso de propagação.

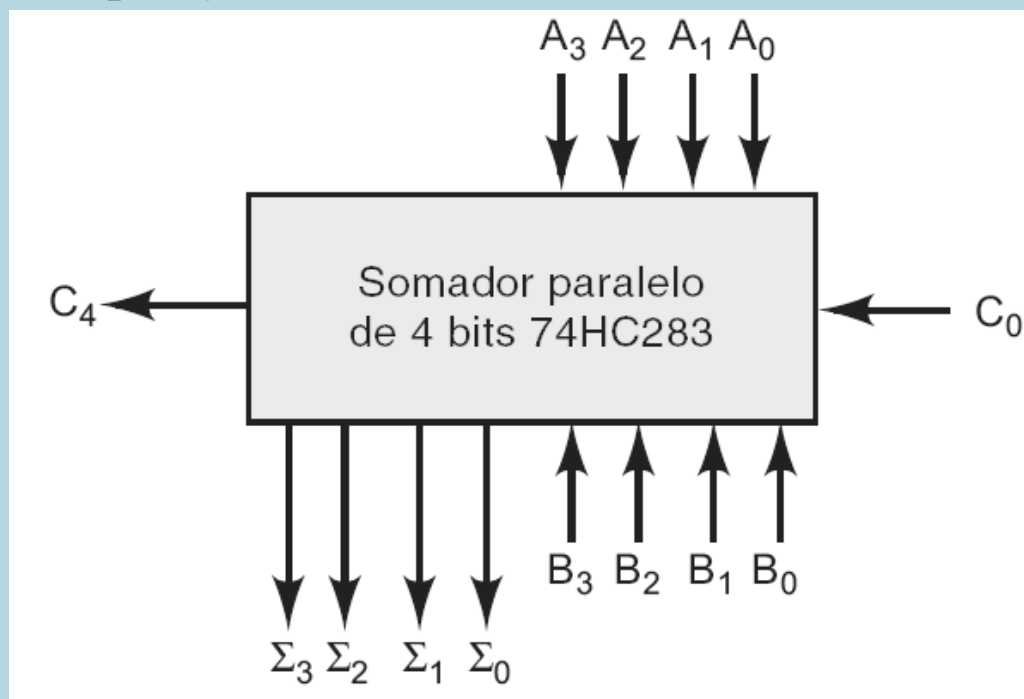
6.14 Somador Paralelo em Circuito Integrado

- O somador paralelo mais comum é um dispositivo de quatro bits, que contém quatro FAs interligados e um circuito de carry antecipado.
- CIs 7483A, 74LS83A, 74LS283, e 74HC283 são somadores paralelos TTL de quatro bits.



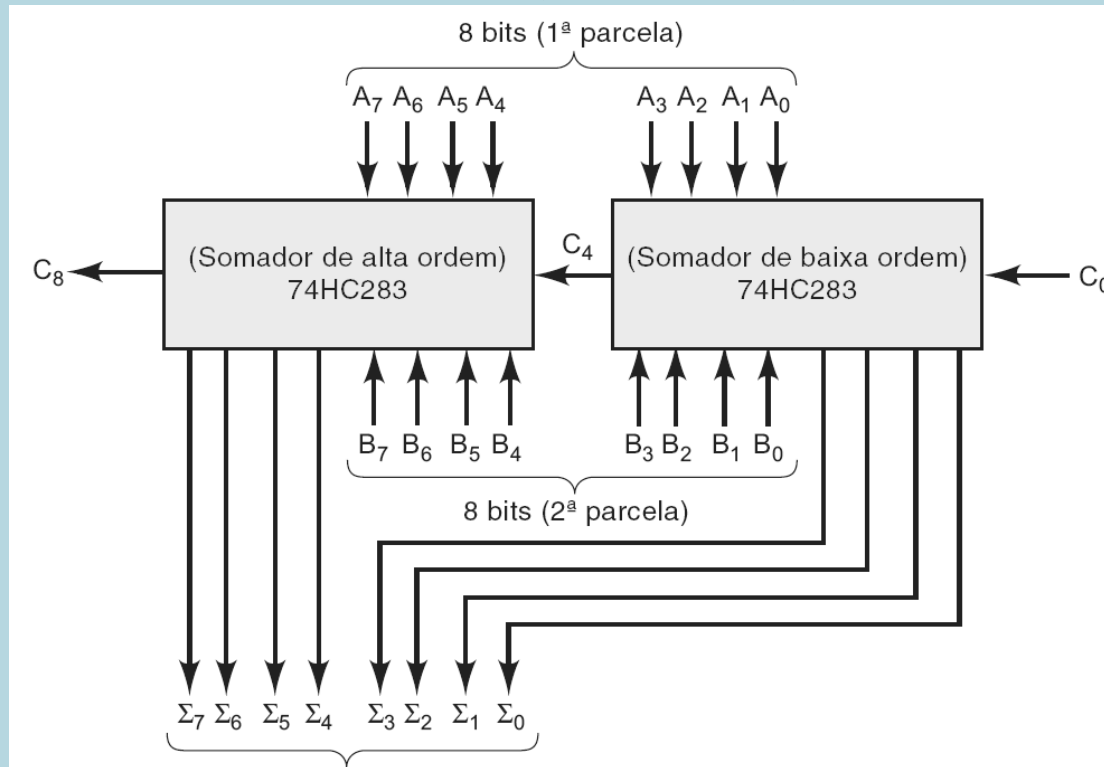
6.14 Somador Paralelo em Circuito Integrado

- As entradas do CI são dois números de quatro bits, $A_3A_2A_1A_0$ e $B_3B_2B_1B_0$, e o carry, C_0 , na posição LSB. As saídas são os bits do resultado da soma e o carry, C_4 , proveniente da posição MSB.



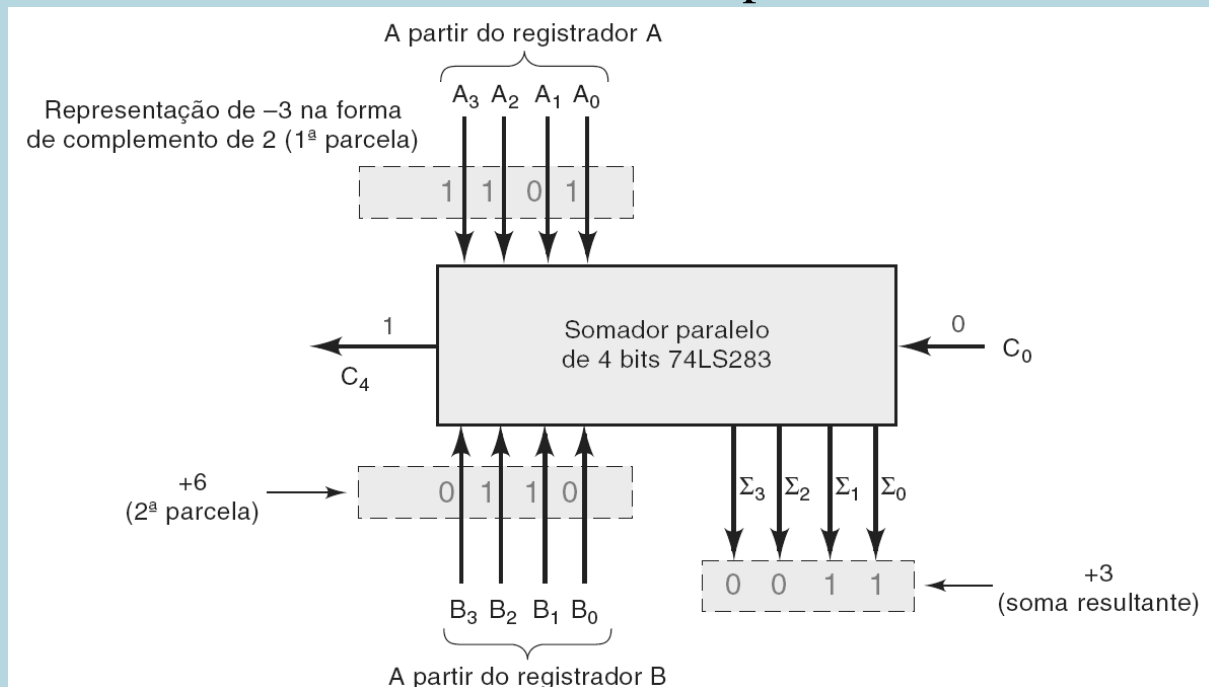
6.14 Somador Paralelo em Circuito Integrado

- Somadores paralelos podem ser conectados em cascata para implementar a adição de números binários maiores, nesse caso, dois números de 8 bits.



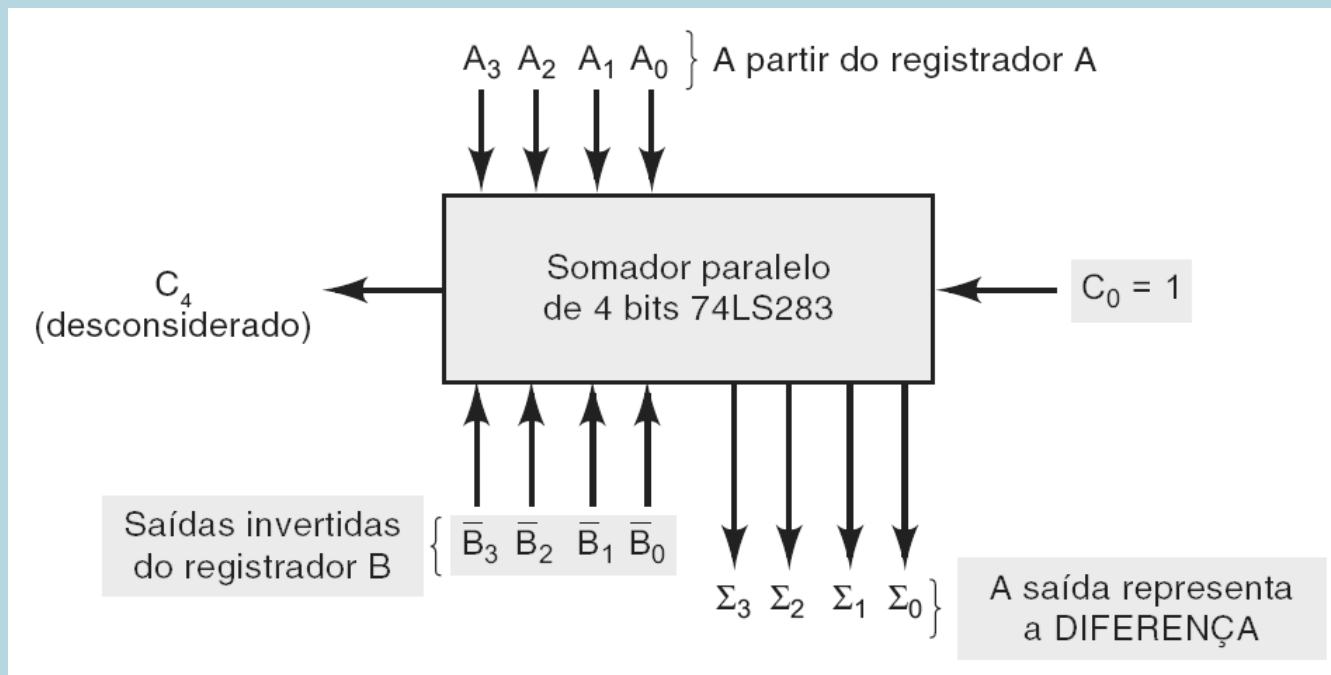
6.15 Sistema de Complemento de 2

- Os números positivos e negativos, incluindo os bits de sinal, podem ser somados em um circuito somador paralelo básico, quando os números negativos forem colocados na forma do complemento de 2.



6.15 Sistema de Complemento de 2

- O circuito do somador paralelo poderá ser adaptado para realizar a subtração, desde que se consiga obter o complemento de 2 do número armazenado no registrador B.

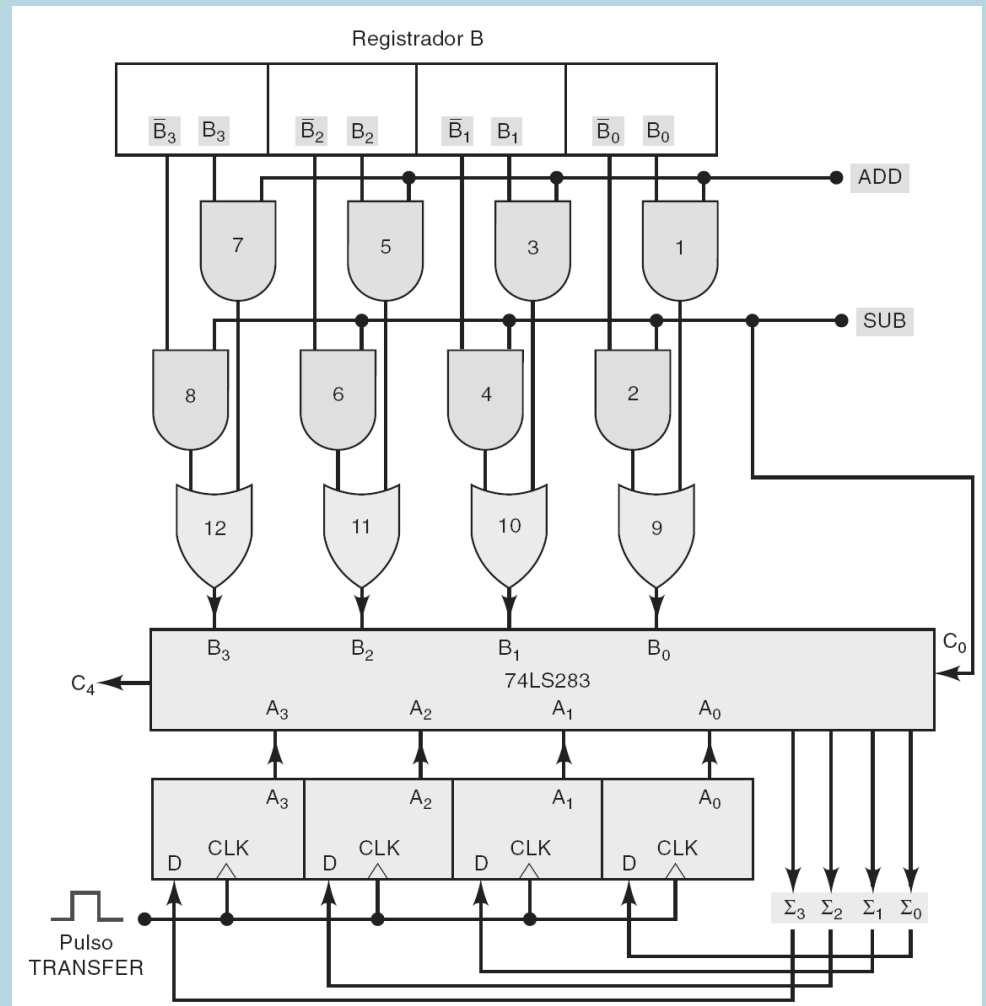


6.15 Sistema de Complemento de 2

- Um circuito completo pode realizar tanto a adição quanto a subtração.
- Esse circuito somador/subtrator é controlado por dois sinais de controle, chamados ADD e SUB:
 - Quando ADD está ALTO, o circuito realizará a adição dos números armazenados nos registradores A e B.
 - Quando SUB está ALTO, ele subtrairá o número que está em B daquele que está em A.

6.15 Sistema de Complemento de 2

Somador/subtrador paralelo, usando o sistema de complemento de 2.

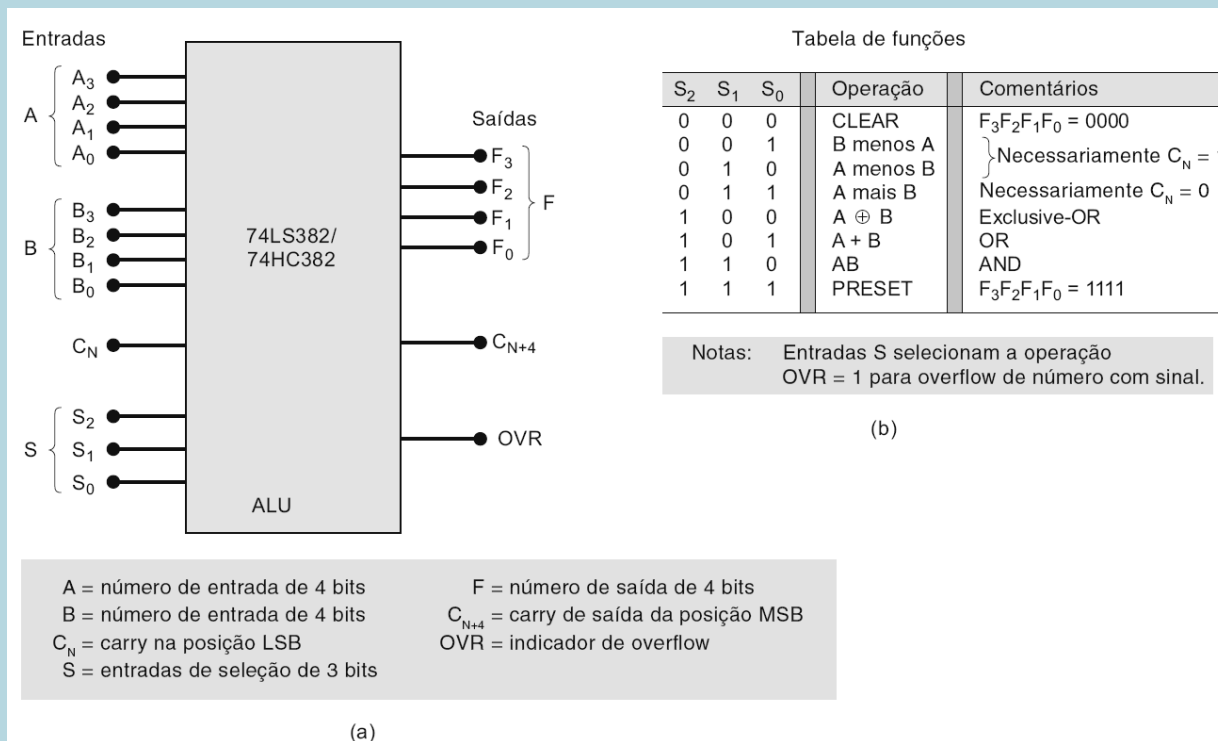


6.16 Circuito Integrado ALU

- Existem vários CIs disponíveis denominados unidades lógicas e aritméticas (*arithmetic/logic units* - ALUs), ainda que não tenham toda a capacidade de uma ALU de um computador.
- Esses chips ALU são capazes de realizar diversas operações lógicas e aritméticas sobre dados binários de entrada.
- A operação específica realizada pela ALU é determinada por um código binário específico colocado nas entradas de seleção de funções.
- ALUs podem realizar operações lógicas e aritméticas diferentes como determinado por um código binário nas entradas selecionadoras de função.

6.16 Circuito Integrado ALU

A ALU 74LS382 (TTL) e 74HC382 (CMOS) é um dispositivo que pode realizar oito funções diferentes.

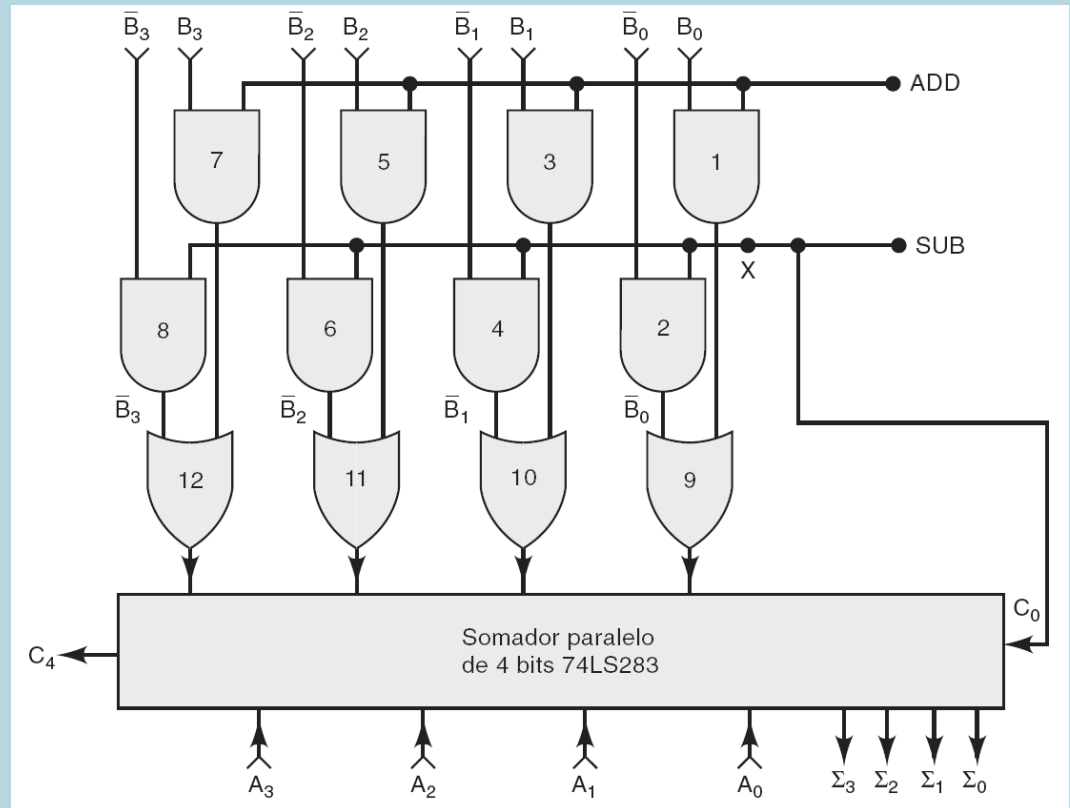


6.17 Estudo de Caso na Análise de Defeitos

- Determinação de um provável problema.

Modo 1: $ADD = 0$, $SUB = 0$.

Os saídas de soma são sempre iguais ao número no registrador A, *mais um*.

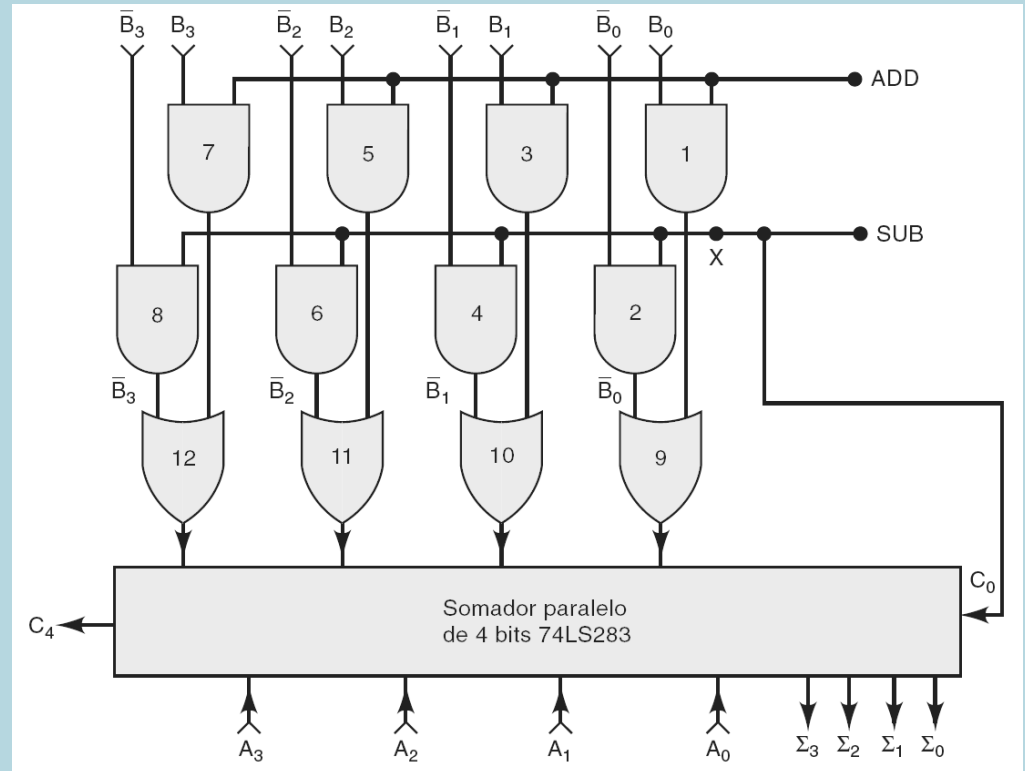


6.17 Estudo de Caso na Análise de Defeitos

- Determinação de um provável problema.

Modo 2: $ADD = 1$, $SUB = 0$.

A soma é sempre *um a mais* do que deveria ser.

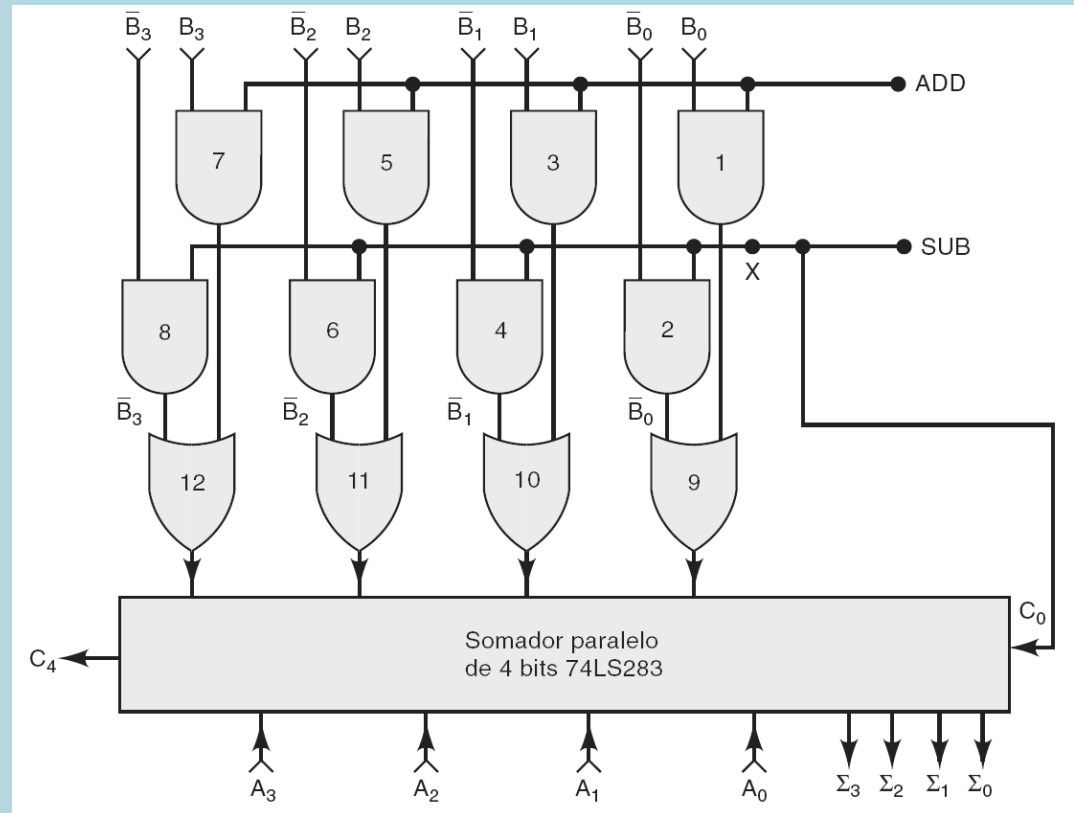


6.17 Estudo de Caso na Análise de Defeitos

- Determinação de um provável problema.

Modo 3: $ADD = 0$, $SUB = 1$.

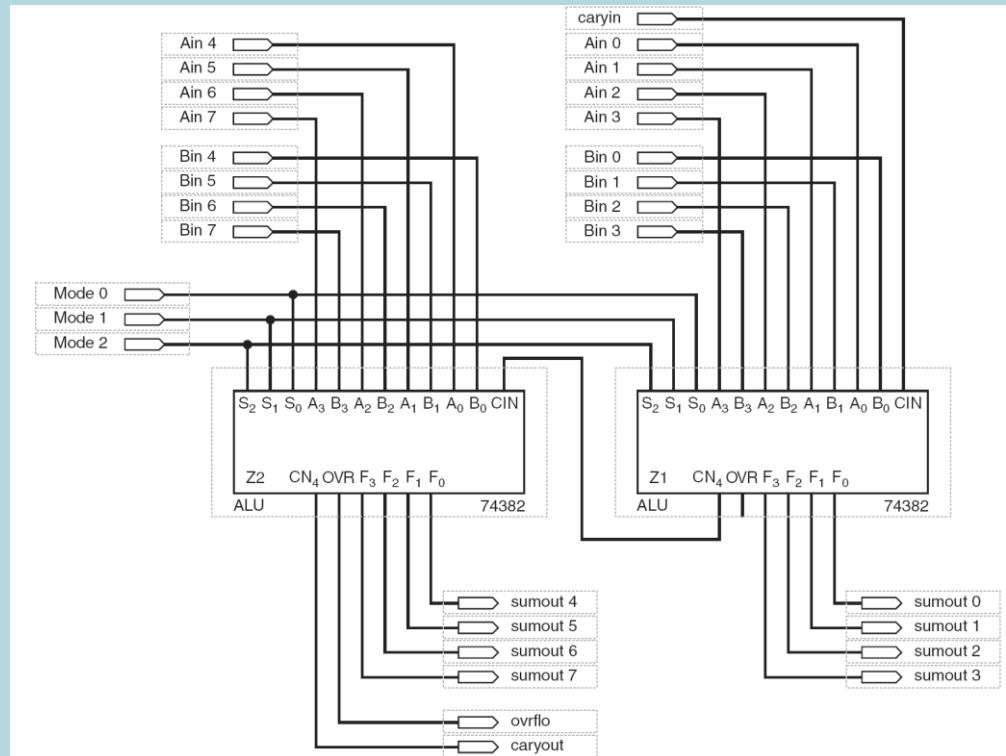
As saídas Σ são sempre iguais a $[A] - [B]$.



6.18 Usando Funções da Biblioteca Altera

- A Altera oferece circuitos pré-definidos lógicos em macrofunções. ALUs podem ser definidas graficamente.

Uma descrição gráfica pode parecer intuitiva, mas às vezes é mais fácil definir um dispositivo, usando texto e macrofunções.



6.19 Operações Lógicas em Vetores de Bits com HDLs

- Duas áreas principais de técnicas de HDL exigem compreensão:
 - Especificação de grupos de bits em vetores.
 - Utilização de operações lógicas para combinar vetores de bits, através de expressões booleanas.
- HDLs usam vetores de bits, um método semelhante à anotação de registro, para descrever sinais.
- O sinal de quatro bits chamado *d* é definido

Em AHDL: **VARIABLE d[3..0] :NODE**

Em VHDL: **SIGNAL d:BIT_VECTOR (3 DOWNT0 0)**

6.20 Somadores em HDL

- Um circuito somador paralelo de 8 bits, usando linguagem HDL, irá adicionar valores de 8 bits $A[8..1]$ e $B[8..1]$ para produzir a soma de 9 bits $S[9..1]$.
- No código AHDL

```
1  SUBDESIGN fig6_23
2  (
3      a[8..1]      :INPUT;          -- primeira parcela de 8 bits
4      b[8..1]      :INPUT;          -- segunda parcela de 8 bits
5      s[9..1]      :OUTPUT;         -- soma de 9 bits
6  )
7  VARIABLE
8      aa[9..1]     :NODE;           -- primeira parcela expandida
9      bb[9..1]     :NODE;           -- segunda parcela expandida
10 BEGIN
11     aa[9..1] = (GND,a[8..1]);     -- concatena à esquerda
12     bb[9..1] = (GND,b[8..1]);     -- para ambos os operandos
13     s[9..1] = aa[9..1] + bb[9..1]; -- adiciona operandos expandidos
14 END;
```

6.20 Somadores em HDL

- Um circuito somador paralelo de 8 bits, usando linguagem HDL, irá adicionar valores de 8 bits $A[8..1]$ e $B[8..1]$ para produzir a soma de 9 bits $S[9..1]$.
- No código VHDL:

```
1  ENTITY fig6_24 IS
2  PORT (
3      a      :IN INTEGER RANGE 0 TO 255;  -- primeira parcela de 8 bits
4      b      :IN INTEGER RANGE 0 TO 255;  -- segunda parcela de 8 bits
5      s      :OUT INTEGER RANGE 0 TO 511  -- soma de 9 bits
6  );
7  END fig6_24;
8
9  ARCHITECTURE parallel OF fig6_24 IS
10
11  BEGIN
12      s <= a + b;  -- adiciona operandos
13  END parallel;
```

6.21 Expandindo a Capacidade em Bits de um Circuito

- Um importante princípio da engenharia de software é a representação simbólica das constantes utilizadas ao longo do código.
- Constantes são simplesmente números fixos representados por um nome (símbolo).
- Pode-se definir um símbolo no topo do código-fonte ao qual for atribuído o valor para o número total de bits e usar-se esse nome ao longo de todo o código.
- Apenas uma linha do código precisa ser mudada para expandir a capacidade do circuito

6.21 Expandindo a Capacidade em Bits de um Circuito

- Adição de um atributo constante para o código HDL para circuito somador/subtrator: um bit de entrada único, chamado *add_sub*, controlará a função somador/subtrator.

```
1  PACKAGE const IS
2    CONSTANT n :INTEGER := 6;
3    CONSTANT m :INTEGER := 2**n;
4    CONSTANT p :INTEGER := 2**m;
5    CONSTANT q :INTEGER := 2**p;
6  END const;
7
8  USE work;
9
10 CONSTANT n = 6;
11
12 SUBDESIGN fig6_25
13 (
14   a[n..1] :INPUT;
15   b[n..1] :INPUT;
16   add_sub :INPUT;
17   result[n..1] :OUTPUT;
18   carryborrow :OUTPUT;
19 )
20 VARIABLE
21   aa[n+1..1] :NODE;
22   bb[n+1..1] :NODE;
```

-- usuário fornece número de bits de entrada
-- addsub
-- primeira parcela de n bits
-- segunda parcela de n bits
-- adiciona ou subtrai
-- resposta de n bits
-- carry de saída
-- primeira parcela expandida
-- segunda parcela expandida
-- adiciona ou subtrai
-- resposta
-- saída de carry ou borrow

O circuito adiciona os dois operandos quando *add_sub* = 0 e subtrai *b* de *a* quando *add_sub* = 1.

O código HDL completo de encontra nas páginas 296 e 297.

6.21 Expandindo a Capacidade em Bits de um Circuito

- Adição de um atributo constante para o código HDL para circuito somador/subtrator: Em AHDL, a palavra-chave CONSTANT é seguida por um nome simbólico e por seu valor atribuído.

```
1  PACKAGE const IS
2  CONSTANT n :INTEGER := 6;    -- usa número
3  CONSTANT m :INTEGER := 2**n; -- calcul
4  CONSTANT p :INTEGER := n+1; --
5  CONSTANT q :INTEGER := 2**p
6  END const;
7
8  USE work.const.all
9
10 ENTITY
11 PORT
12   CONSTANT n = 6;
13   SUBDESIGN fig6_25
14   (
15     a[n..1] :INPUT;
16     b[n..1] :INPUT;
17     add_sub :INPUT;
18     result[n..1] :OUTPUT;
19     carryborrow :OUTPUT;
20   ARCHITECTURE
21     VARIABLE
22     aa[n+1..1] :NODE;
23     bb[n+1..1] :NODE;
```

É necessária atenção com a ocorrência de *overflow* para uma operação de um número com sinal.

O código HDL completo de encontra nas páginas 296 e 297.

6.21 Expandindo a Capacidade em Bits de um Circuito

- Adição de um atributo constante para o código HDL para circuito somador/subtrator: Em VHDL, a palavra-chave CONSTANT é seguida por um nome simbólico, tipo e seu valor atribuído.

```
1  PACKAGE const IS
2      CONSTANT n  :INTEGER := 6;    -- usa número de bits de entrada
3      CONSTANT m  :INTEGER := 2**n; -- calcula o número de bits de saída
4      CONSTANT p  :INTEGER := n+1;  -- número de bits de saída
5      CONSTANT q  :INTEGER := 2**p; -- calcula o número de bits de saída
6  END const;
7
8  USE work.const;
9
10  ENTITY fig6_25
11  (
12      CONSTANT n = 6;
13      SUBDESIGN fig6_25
14      (
15          a[n..1] :INPUT;
16          b[n..1] :INPUT;
17          add_sub :INPUT;
18          result[n..1] :OUTPUT;
19          carryborrow :OUTPUT;
20      )
21  )
22  ARCHITECTURE fig6_26 IS
23      VARIABLE
24          aa[n+1..1] :NODE;
25          bb[n+1..1] :NODE;
```

A declaração geradora VHDL pode ser usada para replicar de forma concisa vários componentes conectados em cascata.

O código HDL completo de encontra nas páginas 296 e 297.

6.21 Expandindo a Capacidade em Bits de um Circuito

- A Altera oferece uma biblioteca de módulos parametrizados (LPMs) que oferecem soluções genéricas para os diversos circuitos lógicos usados em sistemas digitais.