

Aritmética Computacional

Arquitetura e Organização de Computadores

Prof. Lucas de Oliveira Teixeira

UEM

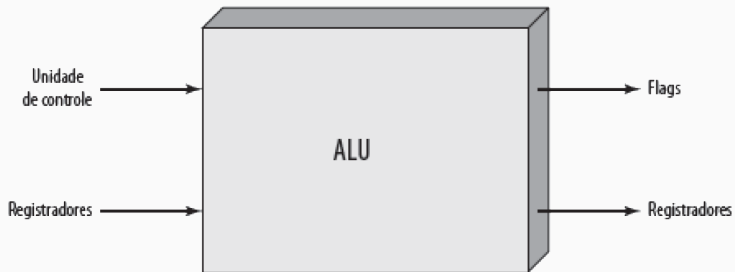
Introdução

- As operações lógicas e aritméticas são realizadas pela ULA (Arithmetic Logic Unit)
- Os demais dispositivos do computador existem para fornecer dados a ULA e depois levar os resultados de volta.
- É a parte do computador que de fato executa as operações sobre os dados.

- É baseada em dispositivos lógicos capazes de efetuar operações de lógica booleana.
- Os dados são fornecidos à ULA em registradores e os resultados de uma operação são armazenados também em registradores.
- A ULA pode também ativar bits especiais (registrador de flags) para indicar o resultado de uma operação.

Introdução

ULA:



- A aritmética computacional opera com dois tipos de números muito diferentes (em termos computacionais): números inteiros e números de ponto flutuante.
- Em ambos os casos, a escolha da representação é uma questão crucial de projeto.

- No sistema de numeração binária também é possível representar números usando vírgulas e sinais:

$$-1101,101_2 = -13,5_{10}$$

- Porém, para armazenar números no computador não é possível usar os sinais de menos e vírgula. Apenas dígitos binários (0 e 1) podem ser usados.

Números inteiros

- Existem diversas representações diferentes para números inteiros, sendo os mais usados: sinal-magnitude e complemento de dois.

Representação sinal-magnitude:

- Utiliza o bit mais significativo da palavra como um bit de sinal:
 - Se o bit mais significativo for 0, o número será positivo.
 - Se for 1, o número será negativo.
- É a forma mais simples de representação.

$$+18_{10} = 00010010_2$$

$$-18_{10} = 10010010_2$$

Representação sinal-magnitude:

- Existem dois problemas com essa representação:
 - Em uma operação aritmética é preciso considerar o sinal dos números.
 - Existem duas representações para o zero:

$$+0_{10} = 00000000_2$$

$$-0_{10} = 10000000_2$$

Representação complemento de dois:

- Também usa o bit mais significativo como bit de sinal.
- Os números positivos são representados na sua forma binária direta, e um bit de sinal 0 é colocado à esquerda.
- Por exemplo: 0001 (+1), 0100 (+4) e 0111 (+7).

Números inteiros

Representação complemento de dois:

- Os números negativos são representados usando a forma de complemento de dois, e um bit de sinal 1 é colocado à esquerda.
- Para o complemento de dois:
 - Invertemos os bits do número em binário:
 0100 invertendo temos 1011
 - Somamos um ao valor invertido:
 $1011 + 0001 = 1100$
 - Lembrando que ainda é necessário colocar o bit de sinal 1 à esquerda.

Representação complemento de dois:

- Essa representação resolve o problema das duas representações do zero.
- Ele é tratado como um número positivo e só existe uma representação.
- É possível representar $2^{n-1} - 1$ números positivos.
- É possível representar 2^{n-1} números negativos

Representação complemento de dois:

- A representação em complemento de dois parece pouco natural.
- Porém, ela também resolve o segundo problema do sinal-magnitude: não é mais necessário se preocupar com o sinal dos operandos.
- Assim, ela torna mais simples a implementação das operações aritméticas mais importantes (adição e subtração).

Números inteiros

Por exemplo, transforme o número 17_{10} para complemento de dois considerando uma arquitetura de 8 bits:

- Primeiro, precisamos transformar 17 para binário:

$$17_{10} = 10001_2$$

- Como o número é positivo basta completar os bits restantes com zero:

$$00010001_2$$

Números inteiros

Por exemplo, transforme o número -12_{10} para complemento de dois considerando uma arquitetura de 8 bits:

- Primeiro, precisamos transformar 12 para binário:

$$12_{10} = 1100_2$$

- Em seguida, precisamos aplicar o complemento de dois:

1100_2 invertendo, temos 0011_2 , somando um,

$$0011_2 + 0001_2 = 0100_2$$

- Finalmente, colocamos o bit de sinal à esquerda e preenchemos os bits restantes com um (por ser negativo):

$$11110100_2$$

Números inteiros

Por exemplo, transforme o número -50_{10} para complemento de dois considerando uma arquitetura de 8 bits:

- Primeiro, precisamos transformar 50 para binário:

$$50_{10} = 110010_2$$

- Em seguida, precisamos aplicar o complemento de dois:

110010_2 invertendo, temos 001101_2 , somando um,

$$001101_2 + 000001_2 = 001110_2$$

- Finalmente, colocamos o bit de sinal à esquerda e preenchemos os bits restantes com um (por ser negativo):

$$11001110_2$$

Operações aritméticas:

- A grande vantagem do complemento de dois é que não precisamos nos preocupar com o sinal dos operandos ao realizar a operação.
- Isso permite que a subtração seja feita como uma adição, ou seja, economia de circuito.

$$A - B = A + (-B)$$

Aritmética de inteiros

Adição:

- Simplesmente, fazemos a soma direta dos dois números:

$$\begin{array}{r} 0101(5) \\ + 0001(1) \\ \hline 0110(6) \end{array}$$

$$\begin{array}{r} 0100(4) \\ + 1100(-4) \\ \hline \textcolor{red}{1}0000(0) \end{array}$$

$$\begin{array}{r} 1100(-2) \\ + 1111(-1) \\ \hline \textcolor{red}{1}1101(-3) \end{array}$$

- Os bits em vermelho são ignorados.

Overflow da adição:

- O overflow acontece quando o resultado da operação de precisa de mais bits do que o hardware oferece.
- Não podemos confundir o overflow com o carry out. Os bits ignorados em vermelho são carry out.
- As regras para detectar overflow são simples:
 - A soma de dois números com sinal diferente nunca gera overflow.
 - A soma de dois números positivos não pode ser negativo.
 - A soma de dois números negativos não pode ser positivo.
- Em outras palavras, o overflow acontece quando dois números com o mesmo sinal forem somados e o seu resultado tiver sinal oposto.

Overflow da adição:

$$\begin{array}{r} 1001 \text{ (-7)} \\ + 1010 \text{ (-6)} \\ \hline 10011 \text{ (-13)} \end{array}$$

Aritmética de inteiros

Subtração:

- Para realizar a subtração, podemos simplesmente aplicar o complemento de dois no segundo operando e realizar a soma.

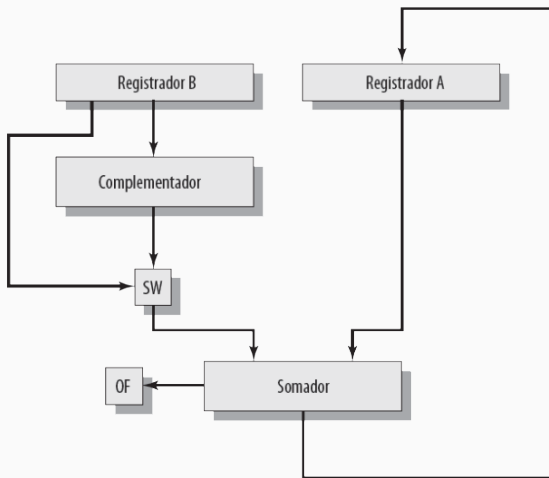
$$A - B = A + (-B)$$

- Por exemplo, para realizar a operação: $5 - 2$, vamos transformar a subtração em uma adição: $5 + (-2)$.
- Em seguida, convertemos para binário e usamos o complemento de dois: $0101 + (-0010) = 0101 + 1110$:

$$\begin{array}{r} 0101 (5) \\ + 1110 (-2) \\ \hline 10011 (3) \end{array}$$

Aritmética de inteiros

Diagrama de blocos da subtração:



OF = bit de *overflow* (do inglês *overflow bit*)

SW = seletor – multiplexador (seleciona adição ou subtração)

Exercício, faça as operações aritméticas abaixo utilizando números binários de 8 bits e complemento de dois:

- $9_{10} + 4_{10}$

- $9_{10} + 7_{10}$

Multiplicação:

- É uma operação complexa, seja implementada em hardware ou seja em software.
- Pode ser dividida em multiplicação de: números inteiros sem sinal e números em representação de complemento de dois.
- Existe uma grande variedade de algoritmos de multiplicação, cada projeto aplica o algoritmo mais conveniente.
- Veremos os mais utilizados.

Multiplicação de inteiros sem sinal:

- Envolve a geração de produtos parciais, que são facilmente determinados.
- O produto total é obtido através da soma dos produtos parciais.
- Forma natural que realizamos multiplicação.
- Obs.: A multiplicação de dois binários de n bits pode gerar um binário de até $2n + 1$ bits.

Multiplicação de inteiros sem sinal:

$$\begin{array}{r} 1010 \text{ (10)} \\ \times 1101 \text{ (13)} \\ \hline 1010 \\ 0000 \\ 1010 \\ + 1010 \\ \hline 10000010 \text{ (130)} \end{array}$$

Multiplicação de inteiros sem sinal:

- No computador a multiplicação é feita de modo mais eficiente do que na forma usual.
- Os produtos parciais são imediatamente acumulados.
- Para cada 1 no multiplicador, realiza-se uma operação de soma e um deslocamento.
- Para cada 0 no multiplicador, apenas um deslocamento é necessário.
- O computador usa alguns registradores auxiliares internos no processo.

Aritmética de inteiros

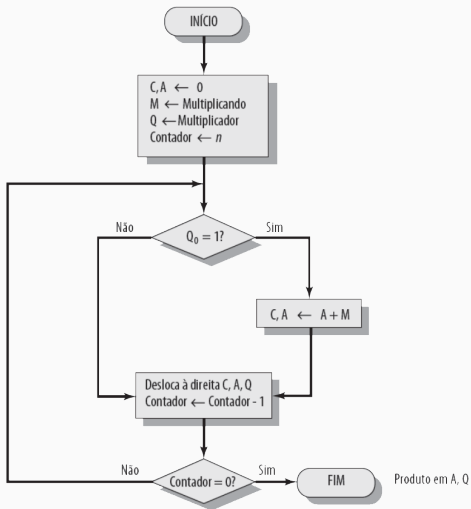
Por exemplo, a multiplicação de 13 por 11:

Carry	A	Q	M	
0	0000	1101	1011	Valores iniciais
0	1011	1101	1011	Adição ($C, A = A + M$)
0	0101	1110	1011	Deslocamento à direita de C, A, Q
0	0010	1111	1011	Deslocamento à direita de C, A, Q
0	1101	1111	1011	Adição ($C, A = A + M$)
0	0110	1111	1011	Deslocamento à direita de C, A, Q
1	0001	1111	1011	Adição ($C, A = A + M$)
0	1000	1111	1011	Deslocamento à direita de C, A, Q

- O resultado é a combinação de A e Q: $10001111_2 = 143_{10}$

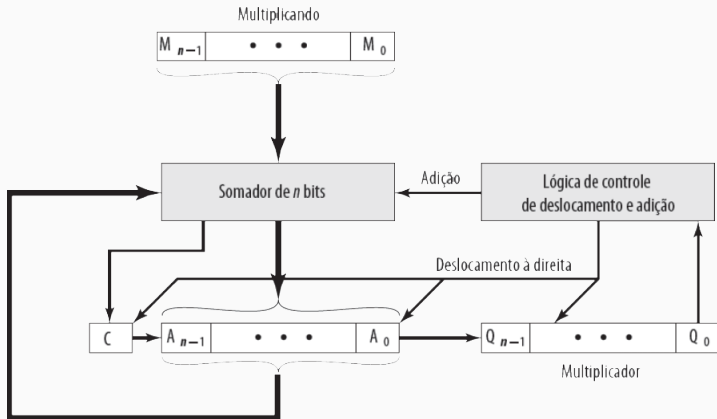
Aritmética de inteiros

Fluxograma da multiplicação de inteiros sem sinal:



Aritmética de inteiros

Diagrama de blocos da multiplicação de inteiros sem sinal:



Multiplicação de inteiros em complemento de dois:

- O método mais usado para multiplicação em complemento de dois é o algoritmo de Booth:
 - O multiplicador e o multiplicando são armazenados nos registradores Q e M.
 - Utiliza-se um registrador de 1 bit, posicionado logicamente à direita do bit menos significativo do registrador Q, o qual é designado como Q-1.
 - O resultado da multiplicação é dado nos registradores A e Q.
 - No início, A e Q-1 são inicializados com 0

Multiplicação de inteiros em complemento de dois:

- Será examinado o bit menos significativo de Q e o bit de Q-1:
 - Se os bits forem iguais (1-1 ou 0-0), então todos os bits dos registradores A, Q e Q-1 serão deslocados aritmeticamente para a direita.
 - Se os bits forem diferentes, o multiplicando será somado ou subtraído do registrador A e depois os bits serão deslocados aritmeticamente para a direita.
 - Se forem 0-1 soma-se.
 - Senão, se forem 1-0 subtrai-se.

Aritmética de inteiros

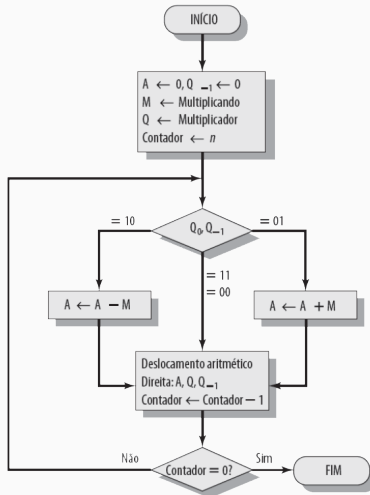
Por exemplo, a multiplicação de 7 por 3:

A	Q	Q-1	M	
0000	0011	0	0111	Valores iniciais
1001	0011	0	0111	Subtração ($A = A - M$)
1100	1001	1	0111	Deslocamento à direita de A, Q, Q-1
1110	0100	1	0111	Deslocamento à direita de A, Q, Q-1
0101	0100	1	0111	Adição ($A = A + M$)
0010	1010	0	0111	Deslocamento à direita de A, Q, Q-1
0001	0101	0	0111	Deslocamento à direita de A, Q, Q-1

- O resultado é a combinação de A e Q: $00010101_2 = 21_{10}$

Aritmética de inteiros

Fluxograma da multiplicação usando o algoritmo de Booth:

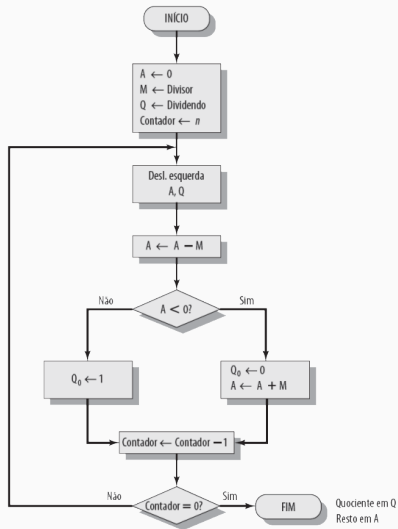


Divisão:

- É bem mais complexa que a multiplicação, embora seja baseada nos mesmos princípios gerais.
- Novamente temos os registradores A, M e Q.
- Também envolve repetidas execuções de adição, subtração e deslocamento.

Aritmética de inteiros

Fluxograma da divisão:



Divisão:

- Vamos pular o exemplo prático.

Exercício, faça as operações aritméticas abaixo utilizando multiplicação de números binários com complemento de dois:

- $0111_2 * 1101_2$
- $1100_2 * 1110_2$