



1 Objetivo

Analisar questões referentes à extensão de sinal em representações binárias sinalizadas.

2 Preliminar

A extensão de sinal consiste em se preservar o sinal (e conseqüentemente o valor) de um número binário quando o número de bits usado em sua representação é aumentado. Esse procedimento depende da representação sinalizada adotada.

Se a representação não usa sinal (*unsigned*), os bits extras trivialmente recebem o valor binário 0. Na representação sinal/magnitude, o bit de sinal deve ser transferido para o bit mais à esquerda e o sinal original é zerado, enquanto que os bits de magnitude não sofrem alteração. Os demais bits recebem o valor 0. Nas representações em complemento (1 e 2), todos os bits à esquerda do bit de sinal recebem o mesmo valor deste bit.

Exemplos de extensão de sinal (8 para 16 bits) para as representações sem sinal, sinal/magnitude e complemento de 2 são dados pela tabela 1.

Tabela 1: Exemplos de extensão de sinal

8 bits	sem sinal	sinal/magnitude	complemento de 2
0010 1011	0000 0000 0010 1011	0000 0000 0010 1011	0000 0000 0010 1011
1011 0101	0000 0000 1011 0101	1000 0000 0011 0101	1111 1111 1011 0101

3 Atividade 1

Salve o arquivo lab03.asm (presente na página do curso) no seu diretório local. Esse arquivo é um fonte em linguagem de montagem de um programa que lê dois dígitos pelo teclado, realiza a soma e escreve o resultado no vídeo. O programa principal está dividido em duas partes:

Trecho inicial - Lê dois números (em decimal) pelo teclado e realiza a soma (figura 1).

Isso é feito utilizando-se uma subrotina chamada *LeEntrada* (linhas 3 e 8), que devolve no nibble menos significativo do registrador AL o valor em complemento de 2 (valores negativos devem ser digitados prefixando-se o caracter -). O valor da entrada é validada de acordo com o número de bits usados para representá-la (no caso, 4 bits) em complemento de 2. Uma mensagem de erro é apresentada caso o número digitado não se encontre nesta faixa.

No final desse trecho (linhas 10 e 11), o resultado da soma sinalizada de 4 bits é armazenado no nibble menos significativo do registrador BL.

Trecho final - Realiza a escrita no vídeo do resultado da soma (figura 2).

A escrita é feita movendo-se o valor da adição do registrador BL para o registrador DH (linha 4) e chamando a rotina *ImprimeResultado* (linha 5). Essa rotina imprime no vídeo um número sinalizado de 8 bits contido no registrador DH. É checado se os números estão na faixa esperada (considerando o resultado de uma soma de 4 bits), e uma mensagem de resultado inválido é mostrada caso o número esteja fora desta faixa.

```
; Trecho inicial - Leitura dos digitos pelo teclado (4 bits)

1  mov  ax, data      ; seta segmento de dados
2  mov  ds, ax

3  call LeEntrada    ; le primeiro numero
4  mov  bl, al        ; bl <- 1o digito

5  mov  ah, 2         ; imprime caracter '+' (soma)
6  mov  dl, '+'
7  int  0x21

8  call LeEntrada    ; le segundo numero
9  mov  bh, al        ; bh <- 2o digito

10 add  bl, bh        ; soma os dois digitos -> resultado em bl
11 and  bl, 0x0f      ; limpa nibble alto (aritmetica modulo 16)
```

Figura 1: Trecho inicial do programa

```
; Trecho final - Escrita do numero no video (8 bits)

1  mov  dl, '='       ; imprime caracter '=' (resultado)
2  mov  ah, 2
3  int  0x21

4  mov  dh, bl
5  call ImprimeResultado

6  Fim:
7  mov  ax, 0x4C00    ; retorna ao DOS
8  int  0x21

; FIM
```

Figura 2: Trecho final do programa

Não se preocupe em entender as rotinas *LeEntrada* e *ImprimeResultado* internamente nesse ponto.

Use o montador (NASM) e o linkeditor (TLINK) para transformar o arquivo `lab03.asm` em um programa executável (`lab03.exe`), conforme procedimento visto no laboratório 1. Sabendo que são usados 4 bits para representar números sinalizados no processo de captura dos dígitos, responda: qual a faixa de números válidos que deve ser aceita pelo programa?

Execute o programa e entre com os pares de números pelo teclado conforme a tabela 2, anotando o resultado do programa.

Tabela 2: Programa lab 03

1o valor	2o valor	resultado do programa
8	1	
2	4	
7	3	
3	-9	
-2	5	
7	-6	
-1	-1	
-3	-2	
-6	-7	

Você consegue explicar os resultados obtidos? Lembre-se que a operação de entrada de números e soma são feitas usando-se 4 bits, e que a operação de escrita no vídeo entende números sinalizados de 8 bits. Não passe para a atividade 2 enquanto não entender o que está acontecendo!

4 Atividade 2

Altere o programa `lab03.asm` de forma que os resultados negativos sejam corretamente mostrados (desconsidere overflow). Para isso, abra o arquivo `lab03.asm` em um editor de texto ASCII e localize o trecho com o seguinte comentário:

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Insira seu codigo neste espaco

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

Você deve escrever suas instruções no espaço indicado. Use o fato de que o resultado da soma está no nibble menos significativo do registrador BL. Você deve estender o sinal de 4 bits para 8 bits (complemento de 2). O resultado da extensão deve ficar no registrador BL. Você pode utilizar os registradores de uso geral na sua implementação.

Dica: é possível implementar a extensão de sinal utilizando apenas as instruções MOV, AND, SHL e OR. Primeiro pense em um algoritmo em alto nível para extensão de sinal e, em seguida, tente mapear esse algoritmo para as instruções assembly. Para informações adicionais sobre as instruções do 8086 você pode consultar a lista de instruções na página do curso.

Para depurar o programa você pode fazer uso do Turbo Debugger, conforme visto no laboratório

1. Após ter implementado a extensão de sinal, torne a gerar o executável e preencha novamente a tabela 3 a seguir (mesmo pares de números da tabela 2).

Tabela 3: Programa lab 03 (após alterar programa)

1o valor	2o valor	resultado do programa
8	1	
2	4	
7	3	
3	-9	
-2	5	
7	-6	
-1	-1	
-3	-2	
-6	-7	

Os resultados estão agora corretos? Explique o que está acontecendo na soma com os pares $(7, 3)$ (linha 3 da tabela) e $(-6, -7)$ (última linha da tabela).