UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA - DEPARTAMENTO DE INFORMÁTICA APLICADA

INF01107 – Introdução à Arquitetura de Computadores –2015/1

Trabalho Prático 2 - Simulador AHMES

Escrever um programa para o simulador Ahmes que calcule o maior divisor comum (mdc) para números inteiros em 16 bits, ou seja, números inteiros armazenados em dois bytes consecutivos. O cálculo do maior divisor comum pode ser feito pelos métodos descritos no trabalho do Neander, descritos a seguir em pseudo linguagem:

```
function mdc(a, b)
    while b ≠ 0
       t := b
       b := a mod b
       a := t
    return a

function gcd(a, b)
    while a ≠ b
        if a > b
           a := a − b
        else
           b := b − a
    return a
```

Observe-se que, como no trabalho do Neander, o maior divisor comum também é definido para números negativos, ou seja, mdc(a,b) = mdc(|a|,|b|). Observe-se também que mdc(a,0) = a.

O programa Ahmes deve receber como entrada dois números inteiros de 16 bits, em complemento de dois, e calcular o maior divisor comum. Para o cálculo devem ser obrigatoriamente utilizadas as seguintes posições de memória:

Posição 128 – entrada **a,** bits mais significativos
Posição 129 – entrada **a,** bits menos significativos
Posição 130 – entrada **b,** bits mais significativos
Posição 131 – entrada **b,** bits menos significativos
Posição 132 – mdc(a,b), bits mais significativos
Posição 133 – mdc(a,b), bits menos significativos

Para os valores de entrada podem ser assumido que serão diferentes de --32768, ou seja, tanto **a** quanto **b** estarão sempre no intervalo [-32767,+32767].

Dicas e Observações:

- O Ahmes trabalha naturalmente com 8 bits. Para trabalhar com 16 bits (dois bytes), as operações devem ser programadas para tratar cada um destes dois bytes.
- Para trocar o sinal de um número em 16 bits, deve-se inverter cada um dos bytes, depois somar 1 no byte menos significativo e, caso seja gerado um carry, somar este carry no byte mais significativo.
- Para somar dois números de 16 bytes, deve-se inicialmente somar os dois bytes menos significativos, e depois os dois bytes mais significativos. Caso a soma dos bytes menos significativos tenha gerado um carry, somar este carry no byte mais significativo do resultado.

- Para a subtração procede-se de maneira análoga a da soma, mas caso a subtração dos bytes menos significativos tenha gerado um borrow, subtrair este borrow do byte mais significativo do resultado.
- Para dividir por 2, basta deslocar um número para a direita. Se o número for de 16 bits, deslocar primeiro o byte mais significativo, e depois o menos significativo, cuidando de inserir o carry do primeiro deslocamento como bit mais significativo do segundo deslocamento.
- Para testar se um número de 16 bits é negativo, basta testar o seu byte mais significativo.

Os trabalhos serão corrigidos de forma automática, com **20** valores diferentes. Portanto, devem ser observadas rigorosamente as seguintes especificações:

- o código do programa deve iniciar no endereço 0 da memória.
- a primeira instrução executável deve estar no endereço 0.
- os endereços para os dois operandos e para o resultado devem ser exatamente os especificados acima, inclusive na ordem dos bytes.
- usar para variáveis adicionais ou para código extra os endereços de memória de 134 em diante.
- no cálculo, os valores de **a** (endereços 128 e 129) e **b** (endereço 130 e 131) não devem ser modificados.

O trabalho deverá ser entregue no Moodle, na área de "Entrega do Trabalho Ahmes", na forma de um arquivo compactado (formato Zip ou Rar) composto por:

- um arquivo de memória do Ahmes, contendo o programa.
- um arquivo com programa fonte comentado, gerado pelo Daedalus (formato .ahd). Lembre-se de incluir seu nome completo e seu número de cartão nas primeiras linhas deste arquivo.
- Para nomear os arquivos, utilize todo o seu nome, usando maiúsculas e minúsculas, sem espaços em branco e sem acentos. Assim, por exemplo, o aluno Um de Três Quatro deve denominar os seus arquivos de **`UmDeTresQuatro.MEM`**, **`UmDeTresQuatro.AHD`** e **`UmDeTresQuatro.ZIP`** (ou **`RAR`**).

Data de Entrega: 15/06/2015, via Moodle

Alguns casos de teste

Teste	**a (16 bits)**	a, msb (end. 128)	a, lsb (end. 129)	**b (16 bits)**	b, msb (end. 130)	b, lsb (end. 131)	**mdc (16 bits)**	mdc, msb (end. 132)	mdc, lsb (end. 133)
1	**33**	0	33	**27**	0	27	**3**	0	3
2	**256**	1	0	**16**	0	16	**16**	0	16
3	**500**	1	244	**75**	0	75	**25**	0	25
4	**32000**	125	0	**25000**	97	168	**1000**	3	232
5	**-3**	255	253	**-99**	255	157	**3**	0	3
6	**4200**	16	104	**5680**	22	48	**40**	0	40
7	**0**	0	0	**1234**	4	210	**1234**	4	210
8	**2800**	10	240	**2000**	7	208	**400**	1	144
9	**-2500**	246	60	**-1500**	250	36	**500**	1	244
10	**32767**	127	255	**32766**	127	254	**1**	0	1