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

INF01107 – Introdução à Arquitetura de Computadores –2010/1

Trabalho Prático 2 - Simulador AHMES

Escrever um programa para o simulador Ahmes que calcule o maior divisor comum de dois números inteiros positivos (sem sinal) em 8 bits. Este cálculo deverá ser feito através do algoritmo de Euclides, que pode ser definido recursivamente como segue:

- Sejam *a* e *b* dois números naturais, onde *a* >= *b*
- O maior divisor comum é representado por mdc(*a*,*b*) e definido pela função

```
function mdc(a, b)
  if b = 0
     return a
  else
     return mdc(b, a mod b)
```

Por exemplo, para calcular mdc(45,27) recursivamente calcula-se mdc(27,18), depois mdc(18,9) e finalmente mdc(9,0), que retorna 9 como resultado. Esta função recursiva pode ser escrita através de um laço da forma:

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

Observe-se que a variável *t* é usada para trocar os valores de *a* e *b* entre si, de forma que sempre o valor de *a* seja maior (ou no máximo igual) que o de *b*. Ou seja, enquanto que *a* recebe o valor anterior de *b*, *b* recebe o resultado de *a mod b* (o resto da divisão inteira de *a* por *b*, que será garantidamente menor que *b*).

O programa Ahmes deve receber como entrada dois números inteiros positivos de 8 bits, representado em duas posições consecutivas de memória, e calcular o maior divisor comum destes dois números, usando o laço iterativo de divisão descrito acima. Adicionalmente, o programa também deve informar a quantidade de iterações realizadas (quantas vezes o laço foi executado).

Para o cálculo devem ser obrigatoriamente utilizadas as seguintes posições:

Posição 128 – valor de *a*
Posição 129 – valor de *b*
Posição 130 – resultado do cálculo, isto é, mdc(*a*,*b*)
Posição 131 – contador de iterações

Dicas e Observações:

- O método exige que o valor de *a* seja maior ou igual ao valor de *b*. Isto, entretanto, não é garantido nos valores de entrada.
- Por definição, mdc(n,0)=n, e por conseqüência mdc(0,0)=0.
- Pela faixa de valores utilizados, a divisão sempre poderá ser realizada, e não é necessário tratar nenhum caso excepcional (como divisão por zero, por exemplo).
- O programa deverá ser codificado para executar sob um número máximo de **instruções**. Enquanto não é exigida codificação eficiente, esta medida é adotada para impedir codificações ineficientes. O número máximo de **instruções** permitido para cada execução é dado por 40+contador*180.

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 valores de *a*, *b*, mdc(*a*,*b*) e contador devem ser exatamente os especificados acima.
- usar para variáveis adicionais ou para código extra os endereços de memória de 132 em diante.
- no cálculo, os valores de ***a e b*** (endereços 128 e 129) não devem ser modificados.

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

- um arquivo de memória do Ahmes, contendo o programa.
- um arquivo texto no formato do Daedalus, com documentação e comentários. Não se esqueça de incluir seu nome completo e seu número de cartão nas primeiras linhas deste arquivo.

Para nomear os arquivos (memória, texto e compactado), utilize os seus dois primeiros nomes, sem espaço em branco e sem acentos. Assim, por exemplo, o aluno Um Três Dois de Oliveira Quatro deve denominar os seus arquivos de **`UmTres.MEM`**, **`UmTres.AHD`** e **`UmTres.ZIP`** (ou **`RAR`**).

Data de Entrega: 16/06/2010, via Moodle

Alguns casos de teste

Teste	End. 128	End. 129	mdc (End. 130)	contador (End.131)	Limite de instr.
1	0	0	0	0	40
2	0	4	4	0	40
3	5	0	5	0	40
4	17	27	1	5	940
5	123	45	3	5	940
6	75	15	15	1	220
7	255	55	5	5	940
8	128	36	4	4	760
9	251	241	1	3	580
10	251	2	1	2	400
11	241	239	1	3	580
12	255	128	1	3	580