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

INF01107 – Introdução à Arquitetura de Computadores –2012/1

**Trabalho Prático 2 - Simulador AHMES**

Escrever um programa para o simulador Ahmes que calcule a raiz quadrada inteira de um número inteiro positivo em 16 bits. Na prática, diversos métodos podem ser utilizados, mas existe uma maneira (baseada na Equação de Pell) que permite encontrar a parte inteira de uma raiz quadrada simplesmente subtraindo inteiros ímpares. Por exemplo, para calcular a parte inteira da raiz quadrada de 19, calcula-se a sequência:

1. 19 – 1 = 18
2. 18 – 3 = 15
3. 15 – 5 = 10
4. 10 – 7 = 3
5. Como 3 é menor que 9, a sequência para aqui. Como 4 subtrações foram efetuadas, então a resposta é 4

Ou seja, para calcular a parte inteira da raiz quadrada *n* de um número inteiro positivo *m*, pode ser usado o seguinte trecho de programa:

```
n = 0
i = 1
while (m >= i){
    m = m - i;
    i = i + 2;
    n = n + 1;
}
```

Observe-se que se *n* é a raiz exata, o valor final de *m* é zero.

O programa Ahmes deve receber como entrada um número inteiro positivo de 16 bits, representado em duas posições consecutivas de memória, e calcular a parte inteira da raiz quadrada, usando o laço iterativo descrito acima. Adicionalmente, o programa também deve informar se a raiz calculada é exata ou não.

Para o cálculo devem ser obrigatoriamente utilizadas as seguintes posições:

Endereço 128 – valor de *m*, bits mais significativos
Endereço 129 – valor de *m*, bits menos significativos
Endereço 130 – resultado *n*, isto é, a parte inteira da raiz quadrada de *m*
Endereço 131 – indicador do resultado: 1 se a raiz for exata, ou 255 caso contrário

Dicas e Observações:

- Como o número *m* está no intervalo [0, 65535], a parte inteira de sua raiz quadrada está sempre no intervalo [0, 255].
- Como o número *m* é representado em dois bytes, o número *i* (do programa acima) também deve ser representado em dois bytes.
- O Ahmes somente opera com um byte de cada vez. Assim, operações de soma, subtração e comparação de dois bytes devem ser desdobradas em operações sobre um byte. Dica: ao fazer isto, leve em consideração os indicadores de vai-um (C) e emprestou-um (B).

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.
- usar para variáveis adicionais ou para código extra os endereços de memória de 132 em diante.
- no cálculo, o valor de ***m*** (endereços 128 e 129) não deve ser modificado.
- o programa será executado 20 vezes de forma consecutiva (sem ser carregado de disco a cada vez); portanto é necessário inicializar todas as variáveis utilizadas.

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 (.mem), contendo o programa.
- um arquivo texto, gerado pelo Montador Daedalus no formato do Ahmes (.ahd), com documentação contendo uma breve descrição do método utilizado e uma listagem comentada do programa fonte. Não se esqueça 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.TXT`** e **`UmDeTresQuatro.ZIP`** (ou **`RAR`**).

Conforme estabelecido no plano de ensino, trabalhos copiados, independentemente do método usado para sua realização, são considerados como "trabalhos não entregues" e implicam em reprovação.

**Data de Entrega: 29/06/2012 via http://moodle.inf.ufrgs.br**

**Alguns casos de teste**

| Teste | End. 128 | End. 129 | m | n (End.130) | indicador (End. 131) |
|---|---|---|---|---|---|
| 1 | 40 | 160 | 10400 | 101 | 255 |
| 2 | 0 | 0 | 0 | 0 | 1 |
| 3 | 0 | 16 | 16 | 4 | 1 |
| 4 | 0 | 24 | 24 | 4 | 255 |
| 5 | 255 | 255 | 65535 | 255 | 255 |
| 6 | 4 | 64 | 1088 | 32 | 255 |
| 7 | 254 | 0 | 65024 | 254 | 255 |
| 8 | 0 | 1 | 1 | 1 | 1 |
| 9 | 1 | 0 | 256 | 16 | 1 |
| 10 | 254 | 1 | 65025 | 255 | 1 |
| 11 | 16 | 0 | 4096 | 64 | 1 |
| 12 | 39 | 17 | 10001 | 100 | 255 |