



UNIVERSIDADE ESTADUAL DE MARINGÁ

Centro de Tecnologia - CTC
Departamento de Informática - DIN

5178-31 – ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES
INFORMÁTICA – 1º sem 2016
Prof. Rodrigo Calvo

PROJETO SIMULAÇÃO DE INSTRUÇÕES DE MÁQUINA

Objetivo: Desenvolver um programa que simule o comportamento dos registradores de um processador em um ciclo de instrução ao executar uma sequência de comandos de linguagem de máquina. Os registradores devem armazenar dados temporariamente para que tais comandos sejam executados. Além disso, a quantidade de registradores deve ser suficiente para suportar a execução dos comandos solicitados.

Descrição: O simulador desenvolvido deve executar operações básicas de linguagem de máquina, como adição, subtração, divisão, multiplicação, mover dados, carregar e armazenar dados em memória principal e desvio de fluxo. Todas as instruções do algoritmo e os dados contidos nele devem estar armazenados em memória RAM. Para fins de implementação, os comandos devem ser armazenados na RAM sequencialmente em um local distinto de onde os dados serão armazenados.

O programa (simulador) deve receber como entrada um arquivo texto com os dados e a sequência de comandos. A linha 1 até a linha n do arquivo informam os dados e a posição na memória em que estão armazenados. A linha n+2 do arquivo indica o endereço na memória do primeiro comando a ser executado. A linha n+3 indica o primeiro comando da sequência. As linhas subsequentes indicam os demais comandos da sequência, sendo que a posição na memória está implícita de acordo com a posição do primeiro comando.

Após a leitura deste arquivo, o programa em linguagem de máquina e os dados utilizados nele estão representados em uma memória (ou vetor no programa). Este vetor deve simular o armazenamento de dados na RAM. Cada posição do vetor é endereçado por um valor no sistema hexadecimal. O mapeamento entre as posições do vetor da memória ocorre da seguinte maneira: a primeira posição do vetor equivale ao endereço 0x00, a segunda equivale ao endereço 0x01 e assim sucessivamente. Por exemplo, a décima primeira posição do vetor equivale ao endereço 0x0A da RAM. Um exemplo de arquivo texto de entrada é mostrado a seguir:

3	0x02
4	0x05
10	0x06
0x0A	
LOAD	A, 0x05
ADD	A, 0x02
MOV	B, A
MULT	B, 0x06
ADD	B, 5
STORE	B, 0x1B

Considere:

- Todos os dados envolvidos nos comandos devem ser inteiros;
- Os comandos sempre terão 2 argumentos (exceto os comandos de desvio):
 - LOAD A, 0x05 → carrega o valor que está no endereço 0x05 para o registrador A
 - LOAD B, 15 → carrega o valor 15 para o registrador A
 - ADD A, 0x02 → $A = A + \text{o valor que está no endereço 0x02}$
 - MULT B, 32 → $B = B * 32$
- Endereçamento Imediato e Direto;
- Os seguintes registradores
 - A: Acumulador
 - B: Uso geral
 - M1: Multiplicador
 - M2: Multiplicando
 - D1: Divisor
 - D2: Dividendo
 - R: Resto da divisão
 - C: Carry out (1: houve carry – 2: não houve carry)
 - Z: Resultado zero(-1: resultado negativo; 0: resultado igual a zero; 1: resultado positivo)
 - PC: *Program Counter*
 - IR: *Instruction Register*
 - MAR: *Memory Address Register*
 - MBR: *Memory Buffer Register*

- Não há chamada de procedimento;
- Não é necessário salvar o contexto do processador para restaurá-lo posteriormente;
- Não há interrupções;

Como o objetivo deste projeto é simular o comportamento dos registradores de um processador, o conteúdo de todos os registradores deve ser impresso em tela para cada comando da sequência. Para melhor visualizar a saída do programa, controle o fluxo de execução do algoritmo teclando <ENTER> entre a execução de dois comandos.

Antes de executar o primeiro, o conteúdo dos registrados deve ser exibido em tela. Lembre-se que neste instante somente o registrador PC possuirá conteúdo válido.

- Serão avaliados:
 - ✓ Vazamento de memória, referência a valores de variáveis não inicializadas;
 - ✓ Comentários em trechos de código;
 - ✓ Envie junto com o projeto um arquivo (readme.txt, os passos necessários executar os programas (comandos e parâmetros de entrada).
- Implemente usando a linguagem C;
- Os trabalhos deverão ser feitos individualmente ou em dupla;
- Serão verificados trabalhos nos quais haja quaisquer tipos de cópia ou plágio.

Modo de entrega: Fazer upload no Moodle. O nome do arquivo deve ter o seguinte formato: **cpu_ra1_ra2.zip**. A primeira linha do código do arquivo, deve haver o nome completo e RA dos alunos em comentário.

Prazo de entrega: 10/08/2016