Trabalho 1 de Organização e Arquitetura de Computadores

Bruno Justino Garcia Praciano (13/0006912)

4 de setembro de 2017

Resumo

Esse relatório é referente ao trabalho 1 da disciplina OAC, que tem como a finalidade demonstrar como foi construpido o simulador das funções do Mars Mips em liguagem C++.

1 INTRODUÇÃO

O problema apresentado consiste em desenvolver seis funções básicas do Assembly e simular sua execução no Mars Mips, considerando os registradores que podem estar em uso e, especialmente, as modificações na memória de instruções e dados. Para tal deveria se criar um array que correspondesse as 16384 posições da memória, tendo em vista que cada uma representa 1 byte e considerando que na simulação cada um dos 4096 índices terá 32 bits (ou 4 bytes).

Além dos detalhes apresentados anteriormente, também era necessário desenvolver códigos ASM para gerar instruções e dados ("text.bin"e "data.bin", respectivamente) que seriam salvos em binário e, após isso, na memória do simulador. Por fim, deve-se imprimir e comparar os resultados obtidos diretamente pelo MIPS e os simulados em C++.

2 Instruções Para Execução do Código

A cerca dos detalhes do código fonte, o programa foi compilado e executado através de um sistema ArchLinux x64, sem o uso de IDEs, apenas com o auxilio do editor de texto Atom. Ademais, o código foi escrito em linguagem C++ e compilado com o gcc version 7.1.1 20170630.

- Abrindo um terminal e indo para a pasta onde encontra-se os arquivos.
- Efetuar o seguinte comando no Terminal:
  $g++trabalho1_v2.cpp$
- para executar o arquivo compilado, deve-se usar o seguinte comando: ./a.out text.bin data.bin
- Em relação aos testes das funções, estes ocorrem em base as instruções passadas pelo binário e, por causa disso, o usuário deve utilizar-se dos arquivos "text"e "data"que já vem inclusos na pasta compactada, podendo também modificar o código ASM ou criar um novo para ser executado pelo simulador - gerando os binários necessários.

3 Metodologia

- Load Word (LW) Função que lê uma palavra na memória, ou seja, 4 bytes. Se o endereço ou o deslocamento não forem múltiplos de 4, a função retorna o código de erro -2 em decimal. Caso contrário ela retorna a palavra armazenado no endereço + deslocamento. Se o deslocamento for igual a 0, a palavra atual é lida, se o deslocamento for igual a 4, a próxima palavra é lida.

- Load Byte (lb) Função que lê um byte na memória, ela retorna o conteúdo armazenado no endereço+deslocamento. Se o byte for negativo, o retorno é uma palavra em complemento de 2. Ex: 0x10 = -2, retorno = 0x11111110 = -2. Se o deslocamento for igual a 0, o byte menos significativo é lido, se o deslocamento for igual a 1, o segundo byte menos significativo é lido, se o deslocamento for igual a 2, o segundo byte mais significativo é lido, se o deslocamento for igual a 3, o byte mais significativo é lido.

- Load Half World (lh)

  Função que lê uma meia-palavra na memória, ou seja, 2 bytes. Se o endereço não for múltiplo de 4 ou o deslocamento não for múltiplo de 2, a função retorna o código de erro 0xfffffffe, -2 em decimal. Caso contrário ela retorna a meia-palavra armazenada no endereço+deslocamento. Se a meia-palavra for negativa, o retorno é uma palavra em complemento de 2. Ex: 0x1000 = -8, retorno = 0x11111000 = -8. Se o deslocamento for igual a 0, a meia-palavra menos significativa é lida, se o deslocamento for igual a 2, a meia-palavra mais significativa é lida.

- Store Word (sw)

  Função que escreve uma palavra na memória, ou seja, 4 bytes. Se o endereço ou o deslocamento não forem múltiplos de 4, a função retorna o código de erro 0xfffffffe, -2 em decimal. Caso contrário ela armazena o dado no endereço+deslocamento. Se o deslocamento for igual a 0, o dado é armazenado na palavra atual, se o deslocamento for igual a 4,o dado é armazenado na próxima palavra.

- Store Byte (sb)

  Função que escreve uma palavra na memória, ou seja, 4 bytes. Se o endereço ou o deslocamento não forem múltiplos de 4, a função retorna o código de erro 0xfffffffe, -2 em decimal. Caso contrário ela armazena o dado no endereço+deslocamento. Se o deslocamento for igual a 0, o dado é armazenado na palavra atual, se o deslocamento for igual a 4,o dado é armazenado na próxima palavra.

- Store Half Word (sh)

Função que escreve uma meia-palavra na memória, ou seja, 2 bytes. Se o endereço não for múltiplo de 4 ou o deslocamento não for múltiplo de 2, a função retorna o código de erro 0xfffffffe, -2 em decimal. Caso contrário ela armazena o dado no endereço + deslocamento. Se o deslocamento for igual a 0, o dado é armazenado na meia-palavra menos significativa, se o deslocamento for igual a 2, o dado é armazenado na meia-palavra mais significativa.

4 Testes e Resultados

Os testes foram desenvolvidos no arquivo de testes, os resultados obtidos foram iguais aos resultados obtidos pelo MARS, portanto o programa desenvolvido foi considerado satisfatório.