

UNIVERSIDADE DE BRASÍLIA
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

116394 ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES

Trabalho I: Memória do MIPS

Aluno: Ian Nery Bandeira

Matrícula: 17/0144739

Descrição do problema

O trabalho apresentado requer ao aluno que simule as instruções de acesso a memória do MIPS, em C, utilizando da biblioteca `<stdint.h>`, para tipificação dos dados de 32 bits (word), 16 bits (half word) e 8 bits (byte); e de uma memória previamente definida com tamanho de 4096 words. As funções requeridas pelo trabalho são: *dump_mem*, *lw*, *lh*, *lhu*, *lb*, *lhu*, *sw*, *sh* e *sb*.

Descrição das funções

- *dump_mem*: Consiste em receber um endereço e um tamanho específico, para assim imprimir na tela os valores guardados na memória em formato hexadecimal, word por word.
- *lw*: Recebe um endereço e uma constante, para assim retornar e imprimir o valor guardado na posição determinada pelo endereço e constante, em hexadecimal e decimal.
- *lh*: Recebe um endereço e uma constante, para assim retornar e imprimir o valor de uma half word com sinal (16 bits) contida em uma posição da memória determinada pelo endereço e constante, sendo que a constante determina se a leitura é dos 16 bits menos significativos ou dos 16 bits mais significativos da word.
- *lhu*: O mesmo de *lh*, porém retorna um valor de half word sem sinal (16 bits) e imprime apenas um valor em decimal.
- *lb*: Recebe um endereço e uma constante, para assim retornar e imprimir o valor de um byte com sinal (8 bits) contido em uma posição da memória determinada pelo endereço e constante, sendo que a constante determina a posição dos 8 bits que serão lidos na word.

- *lbu*: O mesmo de *lb*, porém retorna um valor de byte sem sinal (8 bits) e imprime apenas o valor em decimal.
- *exit*: Definido para finalização do programa.

Testes e resultados

Os testes consistem em inicialização da memória, imprimir o conteúdo inteiro da memória por meio do comando *dump_mem*, e imprimir o conteúdo da memória por partes via os comandos de *lw*, *lh*, *lhu*, *lb* e *lbu*.

Começando por inicializar a memória, executamos a seguinte sequência de comandos:

```
sb 0 0 0x04; sb 0 1 0x03; sb 0 2 0x02; sb 0 3 0x01;
sb 4 0 0xff; sb 4 1 0xfe; sb 4 2 0xfd; sb 4 3 0xfc;
sh 8 0 0xffff0; sh 8 2 0x8c;
sw 12 0 0xff;
sw 16 0 0xffff;
sw 20 0 0xffffffff;
sw 24 0 0x80000000;
```

E então, após a inicialização da memória, utilizamos o comando *dump 0 28*, para resultar na impressão do conteúdo dos endereços de memória da posição 0 até 6 (28 bytes), e então recebemos na tela

```
mem[0] = 0x01020304
mem[1] = 0xfcfdfeff
mem[2] = 0x008cfff0
mem[3] = 0x000000ff
mem[4] = 0x0000ffff
mem[5] = 0xffffffff
mem[6] = 0x80000000
```

O que corresponde exatamente ao que foi gravado na memória pelos comandos *sb*, *sh* e *sw* previamente utilizados.

Logo, agora se faz necessário o teste das funções *lw*, *lh*, *lhu*, *lb* e *lbu*. Portanto, utilizamos os comandos e recebe-se a respectiva resposta impressa na tela :

<i>lw 0 0</i>	<i>mem[0] = 0x01020304</i> <i>mem[0] = 16909060</i>
<i>lw 4 0</i>	<i>mem[0] = 0xfcfdfeff</i> <i>mem[0] = -50462977</i>
<i>lw 20 0</i>	<i>mem[5] = 0xffffffff</i> <i>mem[5] = -1</i>
<i>lw 16 0</i>	<i>mem[4] = 0x0000ffff</i> <i>mem[4] = 65535</i>
<i>lw 24 0</i>	<i>mem[6] = 0x80000000</i> <i>mem[6] = -2147483648</i>
<i>lh 0 0</i>	<i>mem[0] = 0x0304</i> <i>mem[0] = 772</i>
<i>lh 0 2</i>	<i>mem[0] = 0x0102</i> <i>mem[0] = 258</i>
<i>lh 4 0</i>	<i>mem[1] = 0xfeff</i> <i>mem[1] = -257</i>
<i>lh 4 2</i>	<i>mem[1] = 0xfcfd</i> <i>mem[1] = -771</i>
<i>lhu 0 0</i>	<i>mem[0] = 772</i>
<i>lhu 0 2</i>	<i>mem[0] = 258</i>
<i>lhu 4 0</i>	<i>mem[1] = 65279</i>
<i>lhu 4 2</i>	<i>mem[1] = 64675</i>
<i>lb 0 0</i>	<i>mem[0] = 0x04</i> <i>mem[0] = 4</i>
<i>lb 0 1</i>	<i>mem[0] = 0x03</i> <i>mem[0] = 3</i>
<i>lb 0 2</i>	<i>mem[0] = 0x02</i>

	<i>mem[0] = 2</i>
<i>lb 0 3</i>	<i>mem[0] = 0x01</i> <i>mem[0] = 1</i>
<i>lb 4 0</i>	<i>mem[1] = 0xff</i> <i>mem[1] = -1</i>
<i>lb 4 1</i>	<i>mem[1] = 0xfe</i> <i>mem[1] = -2</i>
<i>lb 4 2</i>	<i>mem[1] = 0xfd</i> <i>mem[1] = -3</i>
<i>lb 4 3</i>	<i>mem[1] = 0xfc</i> <i>mem[1] = -4</i>
<i>lbu 0 0</i>	<i>mem[0] = 4</i>
<i>lbu 0 1</i>	<i>mem[0] = 3</i>
<i>lbu 0 2</i>	<i>mem[0] = 2</i>
<i>lbu 0 3</i>	<i>mem[0] = 1</i>
<i>lbu 4 0</i>	<i>mem[1] = 255</i>
<i>lbu 4 1</i>	<i>mem[1] = 254</i>
<i>lbu 4 2</i>	<i>mem[1] = 253</i>
<i>lbu 4 3</i>	<i>mem[1] = 252</i>

Portanto, conclui-se que todas as funções implementadas no programa possuem os resultados esperados.