

Relatório - Trabalho 1

Organização e Arquitetura de Computadores - Turma C

Gabriel Vieira de Arimatéa, 15/0126956

¹CIC – Universidade de Brasília (UnB)

vieira.arimatea@gmail.com

1. Descrição do problema

O trabalho consiste em construir um programa em linguagem C que simule as instruções de acesso à memória do MIPS.

2. Descrição sucinta das funções implementadas

As funções implementadas foram:

2.1. lw

Lê um inteiro alinhado na memória;

2.2. lh

Lê meia palavra (16 bits) com sinal;

2.3. lhu

Lê meia palavra (16 bits) sem sinal;

2.4. lb

Lê um byte com sinal;

2.5. lbu

Lê um byte sem sinal;

2.6. sw

Escreve um inteiro alinhado na memória;

2.7. sh

Escreve meia palavra (16 bits);

2.8. sb

Escreve um byte na memória;

2.9. dump_mem

Imprime o conteúdo da memória no formato hexa, palavra por palavra.

3. Testes e resultados

Para realizar os testes, foram executadas as seguintes funções:

- 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, 0xFFF0); sh(8, 2, 0x8C);

- `sw(12, 0, 0xFF);`
- `sw(16, 0, 0xFFFF);`
- `sw(20, 0, 0xFFFFFFFF);`
- `sw(24, 0, 0x80000000);`
- `dump_mem(0, 28);`
- `lb(0,0); lb(0,1); lb(0,2); lb(0,3);`
- `lb(4,0); lb(4,1); lb(4,2); lb(4,3);`
- `lbu(4,0); lbu(4,1); lbu(4,2); lbu(4,3);`
- `lh(8,0); lh(8,2);`
- `lhu(8,0); lhu(8,2);`
- `lhu(24,0); lhu(24,2);`
- `lw(0, 0); lw(4, 0); lw(12,0); lw(16, 0); lw(20,0);`
- `sb(28, 0, 0X11); sb(28, 1, 0X05); sh(28, 2, 0X8A1);`
- `sw(32, 0, 0X00100000);`
- `sw(32, 0, 0X000000F0);`
- `lw(32,0);`
- `sh(32, 2, 0XACDC);`
- `dump_mem(7, 12);`

Como resultado, obtivemos a seguinte saída:

```
mem[0] = 1020304
mem[1] = fcfdfeff
mem[2] = 8cfff0
mem[3] = ff
mem[4] = ffff
mem[5] = ffffffff
mem[6] = 80000000

Decimal: 4
Hexadecimal 4
Decimal: 3
Hexadecimal 3
Decimal: 2
Hexadecimal 2
Decimal: 1
Hexadecimal 1

Decimal: 255
Hexadecimal ff
Decimal: 254
Hexadecimal fe
Decimal: 253
Hexadecimal fd
Decimal: 252
Hexadecimal fc

Decimal: ff
Decimal: fe
Decimal: fd
Decimal: fc

Decimal: -16
Hexadecimal fff0
Decimal: 140
Hexadecimal 8c

Decimal: 65520
Decimal: 140
Decimal: 0
Decimal: 32768

Decimal: 16909060
Hexadecimal 1020304
Decimal: -50462977
Hexadecimal fcfdfeff
Decimal: 255
Hexadecimal ff
Decimal: 65535
Hexadecimal ffff
Decimal: -1
Hexadecimal ffffffff

Decimal: 240
Hexadecimal f0

mem[7] = 8a10511
mem[8] = acdc00f0
```

Figura 1. Saída no terminal