

# Homework2

张子康 PB22020660

2024 年 04 月 06 日

# 1

## 1.1

定义  $x1=es, x2=eax, x3=ecx, x4$  为计算后的地址

```
slli x3,3
add x4,x2,x3
addi x4,x4,0x5678
lw x1,x4
addi x1,x1,0x5678
sw x1,x4
```

## 1.2

$x1, x2$  为  $eax$ ,  $x3, x4$  为  $ecx$

$x10, x11$  存储  $eax + ecx \times 8 + 0x11223344$  结果

$x12, x13$  存储读取的结果与计算结果

```
// 计算 ecx*8
addi x5,x3,0
srli x5,5
slli x4,3
slli x3,3
add x3,x3,x5
//eax+ecx*8
add x11,x2,x4
sltu x6,x11,x2
add x10,x1,x2
add x10,x10,x6
//eax+ecx*8+0x11223344
addi x11,x11,0x11223344
sltiu x6,x11,0x11223344
```

```

add x10,x10,x6
// 读取寄存器
lw x12,x10
lw x13,x11
addi x13,x13,0x12345678
sltiu x6,x13,0x12345678
add x12,x12,x6
// 写寄存器
sw x12,x10
sw x13,x11

```

## 2

### 2.1

0x56

### 2.2

0x34

## 3

### 3.1

$-2147483648 \leq t_1 \leq -2147475420$

### 3.2

jal 可以到达  $pc + 1048574$ 或 $pc - 1048576$ 。blt 可以到达  $pc + 4094$ 或 $pc - 4096$ 。

## 4

### 4.1

#### 4.1.1

在基于栈的内存管理中，\$sp 寄存器用于指示栈顶位置，负责管理和维护栈内存的分配与回收，而 \$fp 寄存器则用于记录当前函数调用帧的底部地址，以便于访问局部变量和函数参数等栈帧内的数据。

#### 4.1.2

指针

#### 4.1.3

\$ra 占 32 字节，\$fp 占 32 字节，指针占 16 字节，alloca 的空间占 16 字节，第二个变量占 16 字节，一共 14Byte

#### 4.1.4

b \ \$ra

#### 4.1.5

b \ \$ra

### 4.2

#### 4.2.1

$\$fp = 0x24000000, \$sp = 23FFFFB0$

#### 4.2.3

jirl \ \$ra , main , 0

**4.2.4**

```
lu12i.w t0,0x23fff  
addi.w t0,t0,0xffc  
ld.w a1,t0,0  
jirl \($ra,main,0
```