22190009：周帛岑

计组第四次作业

3. $EA = A + (PC)$ $\quad PC = 200 \quad EA = 100$

则 $A = -100$ 又相对位移量长一字节，为补码表示

故第二字节内容 即A为 10011100

4. (1)： $EA = A + (I) = 1200H + 252$

$252 = FCH$

即 $EA = 1200H + FCH = 12FCH$

(2)： 1200H地址中的内容为12FCH

故 $EA = 12FCH$

(3)： 1200H地址中的内容为12FCH

故 $EA = 12FCH$

6. 由题知，二地址指令最多16条 且一地址指令相比二地址指令多6位，零地址指令相比一地址指令多6位，设一地址指令个数为k，

故 $K_0 = 2^6 \times [(16-K_2)2^6 - K_1]$

反解得 $K_1 = (16-K_2)2^6 - \frac{K_0}{2^6}$

考虑二地址，其占用了 K_2 个操作数，

剩余 $(16-K_2)$ 可用作 K_1 　 K_1 最多 $2^6 \times (16-K_2)$ 个

8. (1)： 指令最多有 $2^4 = 16$ 条 指令操作数占6位，其中寻址方式3位，故剩余的可用于通用寄存器的位数为3位

主存一共128KB，一字长16位 故主存有 $\frac{128KB}{2B} = 2^{16}$ 个单元，故两寄存器需至少16位

(2)： 由(1)，地址寄存器有16位，故范围为 0000H ~ FFFFH

(3)： 2315H

$(R4) = M[R[R4]] = M[1234H] = 5678H$

$(R5)+ = M[R[R5]] = M[5678H] = 1234H \quad R[5] = 5679H$

执行：R5中的内容 $= 5678H + 1234H = 68ACH$

9. lui \$s, 007FH

ori \$s1, \$t, FFE0H 　　　　007FFE0

add \$s2 \$s1 \$s0

构造一数，其5~22位为1，再与\$s0中数据按位与即可

10.

```
        add   $t0 , $zero , $zero   # 初始化 $t0
loop:   beq   $a1 , $zero , finish  # 若$a1中的值为0，则跳转到finish
        add   $t0   $t0 , $a0       # $t0中的值加上$a0中的值
        sub   $a1   $a1 , 1         # $a1中的值减1
        j     loop        # 循环
finish: addi  $t0   $t0   100       # $t0的值加上100
        add   $v0   $t0   $zero     # 输出结果至$v0寄存器
```

功能：计算 $a*b-100$

```
11.     sll   $a2 $a2  2    # 数组长度乘4(一个数组的每一元素占4个位)
        sll   $a3 $a3  2    # 同上
        add   $v0 $zero $zero # 初始化
        add   $t0 $zero $zero # 同上
outer:  add   $t4  $a0  $t0  # $t4的内容变化$a0+$t0的内容
        lw    $t4  0($t4)    # 读取数组值
        add   $t1  $zero $zero # $t1 初始化
inner:  add   $t3  $a1  $t1  # $t3的内容变为 $a1+$t1的内容
        lw    $t3  0($t3)  #读取数组值
        bne   $t3 ,$t4 ,skip #若数组1的值与数组2的值不相等就跳转至下一循环
        addi  $v0 ,$v0 ,1  #相等时结果+1
skip:   addi  $t1 , $t1 , 4   # $t1指向数组下一位
        bne   $t1 , $a3 , inner  #跳转 inner
        addi  $t0 , $t0 , 4     # t0指向数组下一位
        bne   $t0 , $a2 , outer  #跳转 outer
```

功能：检查两数组，有多少数相同,返回相同数个数

最坏情况为每次均相同，则不会执行skip。 add，addi sll的指令一共 4+2500×(2+(2500×3)+1) = 18757504

lw与bne 一共 1+2500×(1+2500×3+1) = 18755001

故时间为 $\frac{18757504+18755001\times 2}{2^9} \approx 0.028s$

```
12.  ori $t1 $t0 25
     换为 65525:   lui $t0 1   or $t1 $t0 $t1
```

13. 未改 v_0 值(即变化)循环无法跳出

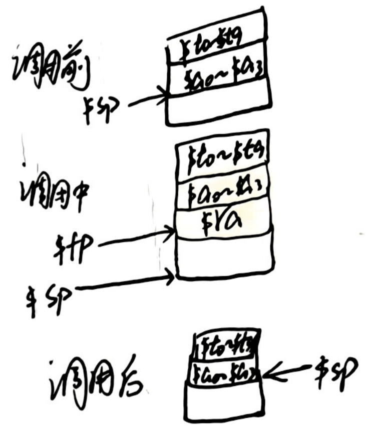
```
修改: addi $v0 $zero 0

loop:  lw   $v1  0($a0)
       sw   $v1  0($a1)
       beq  $v1  $zero exit
       addi $a0,$a0,4
       addi $a1,$a1,4
       addi $v0,$v0,1
       j    loop
exit:
```

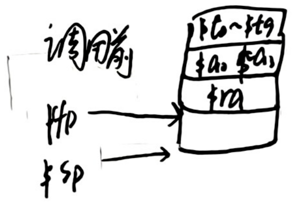
```
17.     andi t1 t0 31                 lui t2 0x00010
                                      and t1 t0 t2
```

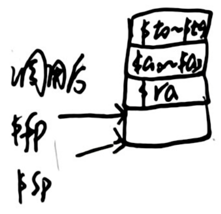
由于 RV32I 指令集的立即数为12位，故且 lui 指令操作高20位，故存在区别

[redacted]

```
15.  addi $s0 $zero 0
Sum_array :   addi $sp $sp 8
              sw   $ra 4($sp)
              sw   $fp 0($sp)
              addi $fp $sp 4
              add  $t0 $zero $a0   #数组初地址
              add  $t1 $zero $a1   # num在t1中
              addi $t2 $zero 0     # i 设为0
loop:         slt  $t3 $t2 $t1   若 i<num ,$t3=1 #标志用
               beq  $t3 $zero exit1
               add  $a0  $zero $t1 #  a0中的值变为num
               add  $a1  $zero $t2 #  a1的值变为i
               addi $a1  $a1  1  # 变为i+1
               Jal  compare #利用 a0, a1进行过程调用
               beq  $v0  $zero , continue
               sll  $t3  $t2  2 #ix4
               add  $t3  $t0  $t3  # $t3=array[i]
               lw   $t4  0($t3)
               add  $s0  $s0  $t4
continue       addi  $t2  $t2  1    # i=i+1
               J     loop
exit :         lw    $ra   4($sp)
               lw    $rp   0($sp)
               addi  $sp   $sp   8  #清空栈
               Jr    $ra
compare :      addi  $v0  $zero  0   #返回值为0
                beq  $a0  $a1   exit1
                slt  $t1  $a0   $a1  #a0<a1 $t1=1
                bne  $t1  $zero  exit1 #返回值为1
                addi $v0  $v0   1
exit1  :          jr  $ra
```


调用前
$t0~$t9
$a0~$a3
$sp
调用中
$t0~$t9
$a0~$a3
$ra
$fp
$sp
调用后
$t0~$t9
$a0~$a3
$sp


调用前
$t0~$t9
$a0~$a1
$ra
$fp
$sp


调用后
$t0~$t9
$a0~$a3
$ra
$fp
$sp