

Computer Organization

Reference Solution

HW2

(5%)

2.10

$f = 2 * (&A);$

(5%)

2.11

	type	opcode	rs	rt	rd	immed
addi \$t0, \$s6, 4	I-type	8	22	8		4
add \$t1, \$s6, \$0	R-type	0	22	0	9	
sw \$t1, 0(\$t0)	I-type	43	8	9		0
lw \$t0, 0(\$t0)	I-type	35	8	8		0
add \$s0, \$t1, \$t0	R-type	0	9	8	16	

(10%)

2.13

2.13.1 $128 + x > 2^{31}-1, x > 2^{31}-129$ and $128 + x < -2^{31}, x < -2^{31} - 128$
(impossible)

2.13.2 $128 - x > 2^{31}-1, x < -2^{31}+129$ and $128 - x < -2^{31}, x > 2^{31} + 128$
(impossible)

2.13.3 $x - 128 < -2^{31}, x < -2^{31} + 128$ and $x - 128 > 2^{31} - 1, x > 2^{31} + 127$
(impossible)

題意未明確說明清楚作答方式，故本題批改標準會放鬆

(10%)

2.19

2.19.1 0xBABEFEF8

2.19.2 0xAAAAAAAA0

2.19.3 0x00005545

2.19.1 題目有誤，所以下兩種版本得到的答案皆對

題目	sll \$t2, \$t0, 44	sll \$t2, \$t0, 4
ANS	0x12345678	0xBABEFEF8

(10%)

2.20

```
srl $t0, $t0, 11
sll $t0, $t0, 26
ori $t2, $0, 0x03ff
sll $t2, $t2, 16
ori $t2, $t2, 0xffff
and $t1, $t1, $t2
or $t1, $t1, $t0
```

(10%)

2.25

2.25.1 i-type

```
2.25.2 LOOP: slt $t3, $zero, $t2
          beq $t3, $zero, DONE
          sub $t2, $t2, 1
          j LOOP
DONE:
```

(10%)

2.26

2.26.1 20

```
2.26.2 i=10;
        While(i>0){
        i=i-1;
        B=B+2;
        }
```

2.26.3 5N+2

(10%)

2.27

```
2.27   addi $t0, $0, 0
        beq  $0, $0, TEST1
LOOP1:  addi $t1, $0, 0
        beq  $0, $0, TEST2
LOOP2:  add  $t3, $t0, $t1
        sll  $t2, $t1, 4
        add  $t2, $t2, $s2
        sw   $t3, ($t2)
        addi $t1, $t1, 1
TEST2:  slt  $t2, $t1, $s1
        bne  $t2, $0, LOOP2
        addi $t0, $t0, 1
TEST1:  slt  $t2, $t0, $s0
        bne  $t2, $0, LOOP1
```

(10%)

2.31

```
2.31 fib:   addi $sp, $sp, -12      # make room on stack
        sw   $ra, 8($sp)        # push $ra
        sw   $s0, 4($sp)        # push $s0
        sw   $a0, 0($sp)        # push $a0 (N)
        bgt  $a0, $0, test2     # if n>0, test if n=1
        add  $v0, $0, $0        # else fib(0) = 0
        j    rtn                #
test2:  addi $t0, $0, 1          #
        bne  $t0, $a0, gen      # if n>1, gen
        add  $v0, $0, $t0       # else fib(1) = 1
        j    rtn                #
gen:    subi $a0, $a0, 1         # n-1
        jal  fib                # call fib(n-1)
        add  $s0, $v0, $0        # copy fib(n-1)
        sub  $a0, $a0, 1         # n-2
        jal  fib                # call fib(n-2)
        add  $v0, $v0, $s0       # fib(n-1)+fib(n-2)
rtn:    lw   $a0, 0($sp)         # pop $a0
        lw   $s0, 4($sp)         # pop $s0
        lw   $ra, 8($sp)         # pop $ra
        addi $sp, $sp, 12        # restore sp
        jr   $ra
```

```
# fib(0) = 12 instructions, fib(1) = 14 instructions,
# fib(N) = 26 + 18N instructions for N >=2
```

(10%)

2.43

```
2.43 trylk: li    $t1,1
          ll    $t0,0($a0)
          bnez $t0,trylk
          sc   $t1,0($a0)
          beqz $t1,trylk
          lw   $t2,0($a1)
          slt  $t3,$t2,$a2
          bnez $t3,skip
          sw   $a2,0($a1)
skip:    sw   $0,0($a0)
```

(10%)

2.46

2.46.1 Answer is no in all cases. Slows down the computer.

CCT = clock cycle time

ICa = instruction count (arithmetic)

ICls = instruction count (load/store)

ICb = instruction count (branch)

new CPU time = $0.75 * \text{old ICa} * \text{CPIa} * 1.1 * \text{old CCT}$

+ $\text{old ICls} * \text{CPIls} * 1.1 * \text{old CCT}$

+ $\text{old ICb} * \text{CPIb} * 1.1 * \text{old CCT}$

The extra clock cycle time adds sufficiently to the new CPU time such that it is not quicker than the old execution time in all cases.

2.46.2 107.04%, 113.43%

