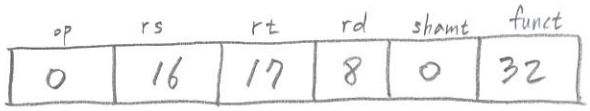
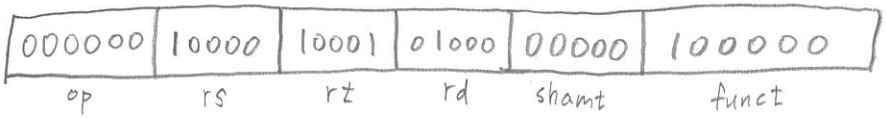
컴퓨터구조 HW 1

2016707079 하상천

Exercise 6.10

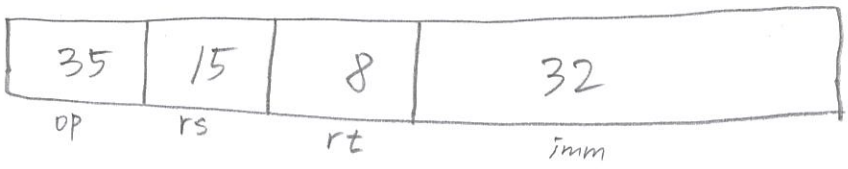
add $t0, $s0, $s1

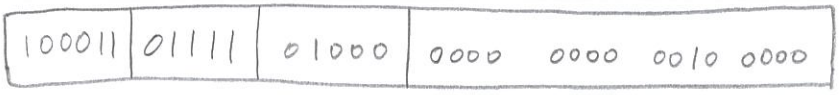




∴ 0x02114020

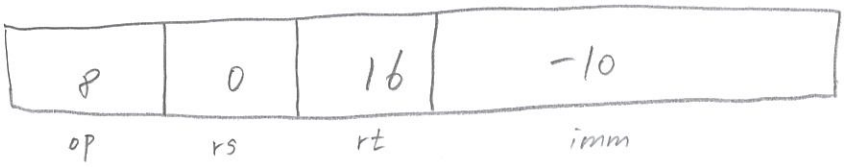
lw $t0, 0x20($t7)

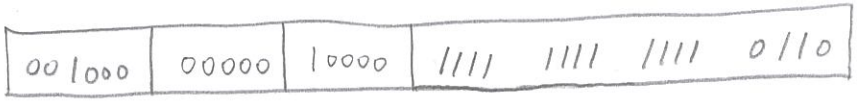




∴ 0x8DE80020

addi $s0, $0, -10

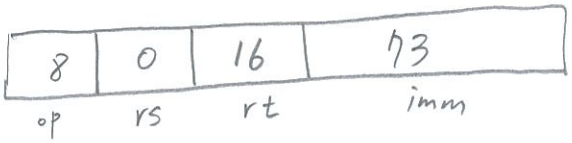


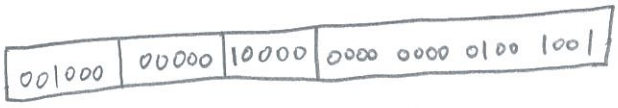


∴ 0x2010FFF6

Exercise 6.11

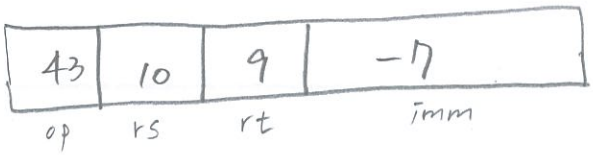
addi $s0, $0, 73

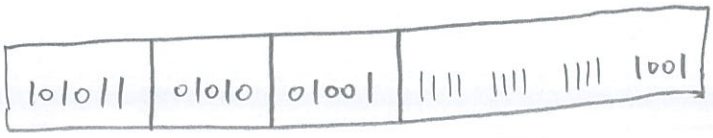




∴ 0x20100049

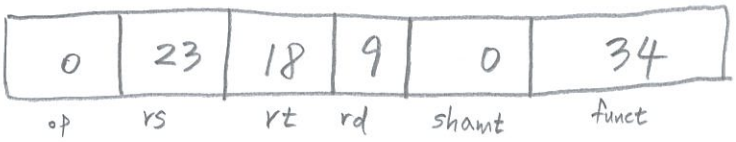
sw $t1, -7($t2)

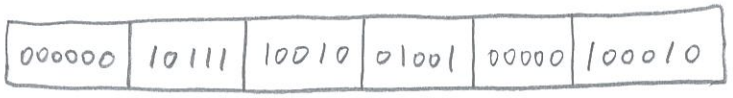




∴ 0xAD49FFF9

Sub $t1, $s7, $s2

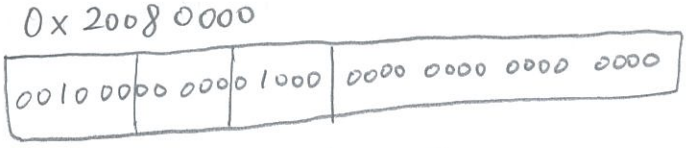




∴ 0x02F24822

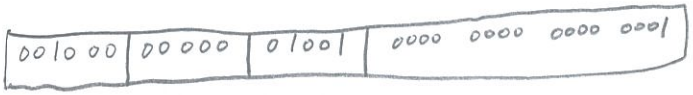
Exercise 6.14

0x20080000



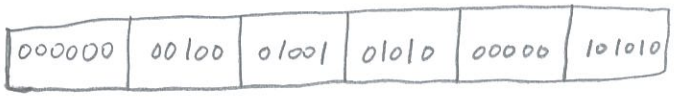
addi $t0, $0, 0

0x20090001



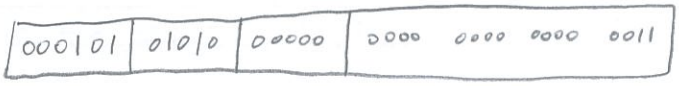
addi $t1, $0, 1

0x0089502A



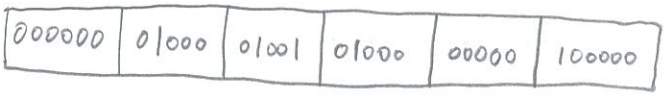
slt $t2, $a0, $t1

0x15400003



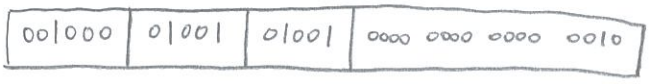
bne $t2, $0, 3

0x01094020



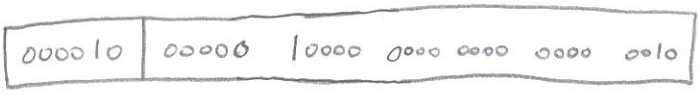
add $t0, $t0, $t1

0x21290002



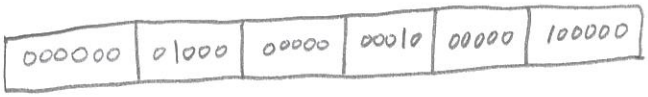
addi $t1, $t1, 2

0x08100002



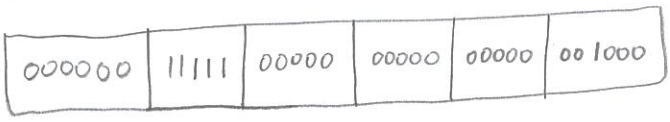
j 0x00400008

0x01001020



add $v0, $t0, $0

0x03E00008



jr $ra

∴

0x00400000	0x20080000		addi	$t0, $0, 0
0x00400004	0x20090001		addi	$t1, $0, 1
0x00400008	0x0089502A	loop:	slt	$t2, $a0, $t1
0x0040000C	0x15400003		bne	$t2, $0, finish
0x00400010	0x01094020		add	$t0, $t0, $t1
0x00400014	0x21290002		addi	$t1, $t1, 2
0x00400018	0x08100002		j	loop
0x0040001C	0x01001020	finish:	add	$v0, $t0, $0
0x00400020	0x03E00008		jr	$ra

temp = $t0, i = $t1, n = $a0,
result = $v0

```
temp = 0;
for(i=1; i<=n; i=i+2)
    temp = temp + i;
result = temp;
```

∴ 프로그램은 n까지의 홀수를 더하고
그 결과값을 $v0 에 넣는다.

Exercise 6.17

(a)
```
if (g > h)
    g = g+h;
else
    g = g-h;
```

```
# $s0 = g, $s1 = h
    slt  $t0, $s1, $s0
    beq  $t0, $0, else
    add  $s0, $s0, $s1
    j    done
else: sub  $s0, $s0, $s1
done:
```

(b)
```
if (g >= h)
    g = g+1;
else
    h = h-1;
```

```
# $s0 = g, $s1 = h
    slt  $t0, $s0, $s1
    bne  $t0, $0, else
    addi $s0, $s0, 1
    j    done
else: addi $s1, $s1, -1
done:
```

(c) if (g <= h)

g = 0 ;

else

h = 0 ;

```
# $s0 = g , $s1 = h

slt  $t0 , $s1 , $s0
bne  $t0 , $0 , else
add  $s0 , $0 , $0
j    done
else: add  $s1 , $0 , $0
done:
```

Exercise 6.21

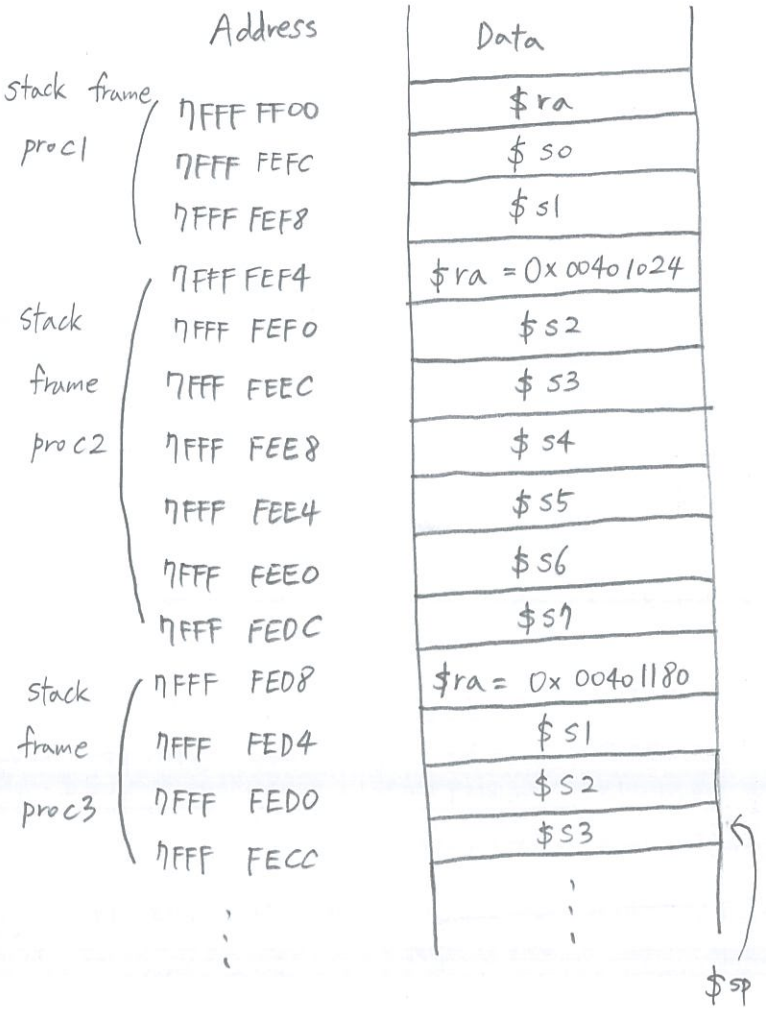
(a) proc1 : 3 words deep ($s0, $s1, $ra)

proc2 : 7 words deep ($s2 ~ $s7, $ra)

proc3 : 4 words deep ($s1 ~ $s3, $ra)

proc4 : 0 words deep

(b) 임의로 Stack pointer의 초기 값을 0x7FFFFF04 라고 하였다.



Exercise 6.26

(a)

0x00400028 add $a0, $a1, $0

000000 00101 00000 00100 00000 100000

0x00A02020

0x0040002C jal f2

000011 0000 0100 0000 0000 0000 0011 01

0x0C10000D

0x00400030 jr $ra

000000 11111 00000 00000 00000 001000

0x03E00008

0x00400034 SW $s0, 0($s2)

101011 10010 10000 0000 0000 0000 0000

0xAE500000

0x00400038 bne $a0, $0, else

000101 00100 00000 0000 0000 0000 0001

0x14800001

0x0040003C j f1

000010 0000 0100 0000 0000 0000 0011 00

0x0810000C

0x00400040 addi $a0, $a0, -1

001000 00100 00100 1111 1111 1111 1111

0x2084FFFF

0x00400044 j f2

000010 0000 0100 0000 0000 0000 0011 01 ⇒ 0x0810000D

∴ 0x00A02020

0x0C10000D

0x03E00008

0xAE500000

0x14800001

0x0810000C

0x2084FFFF

0x0810000D

(b)

```
        add  $a0, $a1, $0    # register only
        jal  f2              # pseudo-direct
f1:     jr   $ra             # register only
f2:     sw   $s0, 0($s2)     # base addressing
        bne  $a0, $0, else   # program counter-relative
        j    f1              # pseudo-direct
else:   addi $a0, $a0, -1    # immediate
        j    f2              # pseudo-direct
```

Exercise 6.36

(a)

```
0x00400000  main :  addi  $sp, $sp, -4
0x00400004          sw    $ra, 0($sp)
0x00400008          lw    $a0, x
0x0040000C          lw    $a1, y
0x00400010          jal   diff
0x00400014          lw    $ra, 0($sp)
0x00400018          addi  $sp, $sp, 4
0x0040001C          jr    $ra
0x00400020  diff :  sub   $v0, $a0, $a1
0x00400024          jr    $ra
```

(b)

Symbol	address
x	0x1000 0000
y	0x1000 0004
main	0x0040 0000
diff	0x0040 0020

(c)

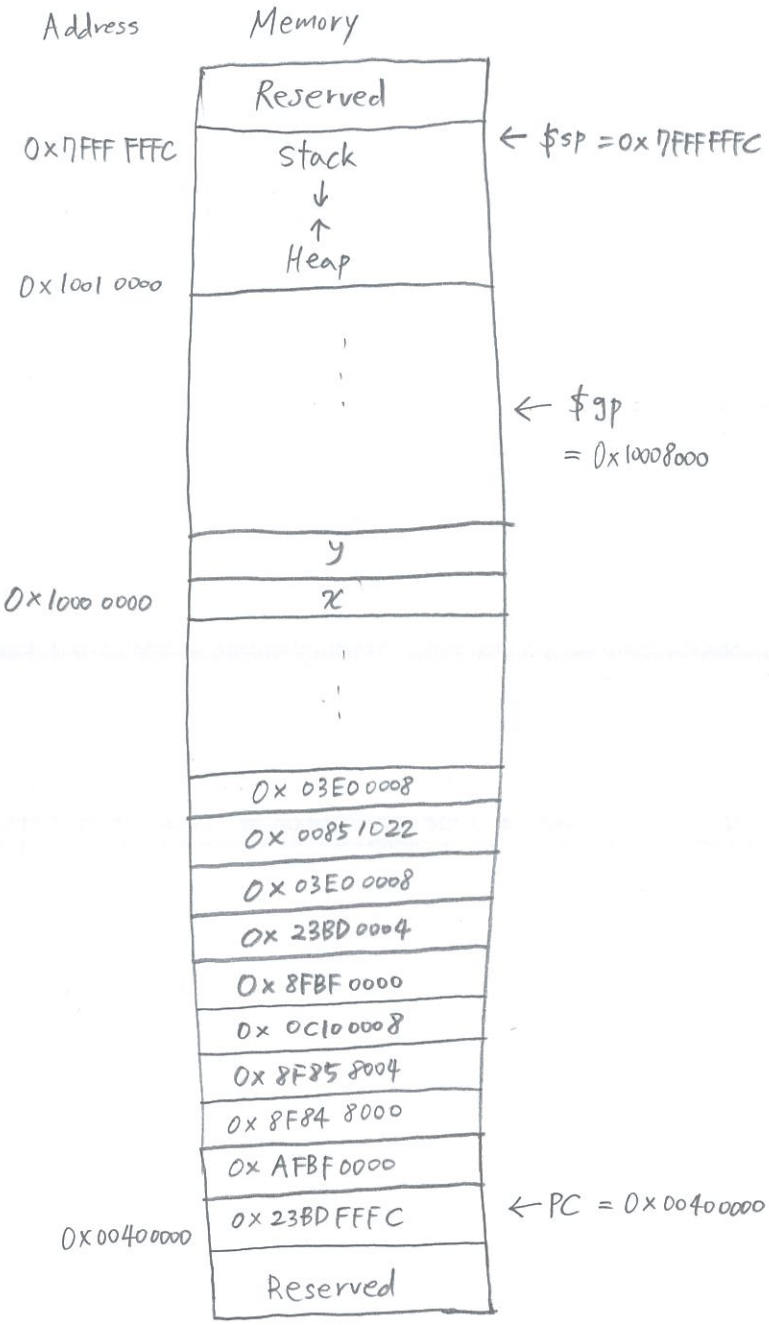
0x23BDFFFC
0xAFBF0000
0x8F848000
0x8F858004
0x0C100008
0x8FBF0000
0x23BD0004
0x03E00008
0x00851022
0x03E00008

(d)

data segment : 8 bytes

text segment : 40 bytes

(e)



Exercise 6.38

(a) addi $t0, $s2, $imm_{31:0}$

```
lui $at, imm31:16
ori $at, $at, imm15:0
add $t0, $s2, $at
```

(b) lw $t5, $imm_{31:0}$ ($s0)

```
lui $at, imm31:16
ori $at, $at, imm15:0
add $at, $at, $s0
lw $t5, 0($at)
```

(c) rol $t0, $t1, 5

```
srl $at, $t1, 27
sll $t0, $t1, 5
or $t0, $t0, $at
```

(d) ror $s4, $t6, 31

```
sll $at, $t6, 1
srl $s4, $t6, 31
or $s4, $s4, $at
```

Exercise 6.39

(a) beq $t1, $imm_{31:0}$, L

```
lui $at, imm31:16
ori $at, $at, imm15:0
beq $t1, $at, L
```

(b) ble $t3, $t5, L

```
slt $at, $t5, $t3
beq $at, $0, L
```

(c) bgt $t3, $t5, L

```
slt $at, $t5, $t3
bne $at, $0, L
```

(d) bge $t3, $t5, L

```
slt $at, $t3, $t5
beq $at, $0, L
```