

Homework Assignment 2 Solution

Exercise 2.3 (5 points)

```
sub $t0, $s3, $s4
sll $t0, $t0, 2
add $t0, $s6, $t0
lw  $t1, 0($t0)
sw  $t1, 32($s7)
```

Exercise 2.4 (5 points)

$B[g] = A[f] + A[1+f];$

Exercise 2.5 (5 points)

```
Sll $t0, $s0, 2
add $t0, $s6, $t0
sll $t1, $s1, 2
add $t1, $s7, $t1
lw  $s0, 0($t0)
lw  $t0, 4($t0)
add $t0, $t0, $s0
sw  $t0, 0($t1)
```

Exercise 2.16 (5 points)

r-type

Assembly instruction: `sub $v1, $v1, $v0`

Hexadecimal representation: `0x00621822`

Binary representation: `00000000011000100001100000100010`

Exercise 2.17 (5 points)

i-type

Assembly instruction: `lw $v0, 4($at)`

Hexadecimal representation: `0x8C220004`

Binary representation: `10001100001000100000000000000100`

Exercise 2.18 (15 points)

2.18.1

opcode would be 8 bits, rs, rt, rd fields would be 7 bits each

2.18.2

opcode would be 8 bits, rs and rt fields would be 7 bits each

2.18.3

more registers → more bits per instruction → could increase code size

more registers → less register spills → less instructions

more instructions → more appropriate instruction → decrease code size

more instructions → larger opcodes → larger code size

Exercise 2.19 (15 points)

2.19.1

`0xBABEF8`

2.19.2

0x0000AAA0

2.19.3

0x00005545

Exercise 2.20 (5 points)

```
sll $t1, $t1, 6
srl $t1, $t1, 6
srl $t0, $t0, 11
sll $t0, $t0, 26
or $t1, $t1, $t0
```

Exercise 2.23 (5 points)

\$t2 = 3

Exercise 2.24 (5 points)

jump: no, beq: no

the jump target is limited by 28-bit byte address segment and the branch target has offset limited by 16-bit

Exercise 2.26 (15 points)**2.26.1**

20

2.26.2

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

2.26.3

$5 \times N$ or $5 \times N + 2$ are both acceptable answers