Homework 2

- Chapter 2: 2.10, 2.12, 2.14, 2.16, 2.19, 2.23

11/14/09

2.10 [5] <§§2.2, 2.3> Translate the following MIPS code to C. Assume that the variables `f`, `g`, `h`, `i`, and `j` are assigned to registers `$s0`, `$s1`, `$s2`, `$s3`, and `$s4`, respectively. Assume that the base address of the arrays `A` and `B` are in registers `$s6` and `$s7`, respectively.

```
addi $t0, $s6, 4
add  $t1, $s6, $0
sw   $t1, 0($t0)
lw   $t0, 0($t0)
add  $s0, $t1, $t0
```

C: A[j] = A+1;
f = 2A;

2.12 Assume that registers `$s0` and `$s1` hold the values `0x80000000` and `0xD0000000`, respectively.

2.12.1 [5] <§2.4> What is the value of `$t0` for the following assembly code?

```
add $t0, $s0, $s1
```

2.12.2 [5] <§2.4> Is the result in `$t0` the desired result, or has there been overflow?

2.12.3 [5] <§2.4> For the contents of registers `$s0` and `$s1` as specified above, what is the value of `$t0` for the following assembly code?

```
sub $t0, $s0, $s1
```

2.12.4 [5] <§2.4> Is the result in `$t0` the desired result, or has there been overflow?

2.12.5 [5] <§2.4> For the contents of registers `$s0` and `$s1` as specified above, what is the value of `$t0` for the following assembly code?

```
add $t0, $s0, $s1
add $t0, $t0, $s0
```

2.12.6 [5] <§2.4> Is the result in `$t0` the desired result, or has there been overflow?

① $t0 = 0x00000000
② not the desired result, is overflow
③ $t0 = 0x60000000
④ Yes, it's not overflow.
⑤ $t0 = 0x00000000
⑥ not, overflow.

*, But with the "add" change to "addu"
we have: ① $t0 = 0x[0000000]
⑤ $t0 = 0x60000000

2.14 [5] <§§2.2, 2.5> Provide the type and assembly language instruction for the following binary value: $0000\ 0010\ 0001\ 0000\ 1000\ 0000\ 0010\ 0000_{two}$

R-type: add $1, $2, $3
即 $1 = 16, $2 = 16, $3 = 16

2.16 [5] <§2.5> Provide the type, assembly language instruction, and binary representation of instruction described by the following MIPS fields:

```
op=0, rs=3, rt=2, rd=3, shamt=0, funct=34
```

R-type: sub $1, $2, $3
即 $1 = 3, $2 = 3, $3 = 2

0000 0000 0110 0010 0001 1000 0010 0010

2.19 Assume the following register contents:

```
$t0 = 0xAAAAAAAA, $t1 = 0x12345678
```

2.19.1 [5] <§2.6> For the register values shown above, what is the value of `$t2` for the following sequence of instructions?

```
sll $t2, $t0, 44
or  $t2, $t2, $t1
```

2.19.2 [5] <§2.6> For the register values shown above, what is the value of `$t2` for the following sequence of instructions?

```
sll  $t2, $t0, 4
andi $t2, $t2, -1
```

2.19.3 [5] <§2.6> For the register values shown above, what is the value of `$t2` for the following sequence of instructions?

```
srl  $t2, $t0, 3
andi $t2, $t2, 0xFFEF
```

① $t1 = 0xbabefef8

② $t2 = 0xaaaaaaa0

③ $t2 = 0x00005545

2.23 [5] <§2.7> Assume `$t0` holds the value `0x00101000`. What is the value of `$t2` after the following instructions?

```
        slt  $t2, $0,  $t0
        bne  $t2, $0,  ELSE
        j    DONE
ELSE:   addi $t2, $t2, 2
DONE:
```

$t0 = 00101000

⇒ $t2 = 3