HW2

Exercise: 2.3, 2.4, 2.7, 2.8, 2.9, 2.10, 2.17, 2.24, 2.25, 2.26, 2.29

2.3 [5] <§§2.2, 2.3> For the following C statement, write the corresponding MIPS assembly code. 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.

```
B[8] = A[i-j];
```

2.4 [5] <§§2.2, 2.3> For the MIPS assembly instructions above, what is the corresponding C statement? 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.

```
sll  $t0, $s0, 2     # $t0 = f * 4
add  $t0, $s6, $t0   # $t0 = &A[f]
sll  $t1, $s1, 2     # $t1 = g * 4
add  $t1, $s7, $t1   # $t1 = &B[g]
lw   $s0, 0($t0)     # f = A[f]
addi $t2, $t0, 4
lw   $t0, 0($t2)
add  $t0, $t0, $s0
sw   $t0, 0($t1)
```

2.7 [5] <§§2.2, 2.3> Translate the following C code to MIPS. 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. Assume that the elements of the arrays A and B are 8-byte words:

```
B[8] = A[i] + A[j];
```

2.8 [10] <§§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
```

2.9 [20] <§§2.3, 2.5> For each MIPS instruction in Exercise 2.8, show the value of the opcode (op), source register (rs) and funct field, and destination register (rd) fields. For the I-type instructions, show the value of the immediate field, and for the R-type instructions, show the value of the second source register (rt).

2.10 Assume that registers `$s0` and `$s1` hold the values `0x8000000000000000` and `0xD000000000000000`, respectively.

2.10.1 [5] <§2.4> What is the value of `$t0` for the following assembly code?

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

2.10.2 [5] <§2.4> Is the result in `$t0` the desired result, or has there been overflow?

2.10.3 [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
```

2.10.4 [5] <§2.4>Is the result in `$t0` the desired result, or has there been overflow?

2.10.5 [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
```

2.10.6 [5] <§2.4> Is the result in `$t0` the desired result, or has there been overflow?

2.17 Assume the following register contents:

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

2.17.1 [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
```

2.17.2 [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
```

2.17.3 [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
```

2.24 Consider the following MIPS loop:

```
LOOP: slt  $t2, $0,  $t1
      beq  $t2, $0,  DONE
      subi $t1, $t1, 1
      addi $s2, $s2, 2
      j    LOOP
DONE:
```

2.24.1 [5] <§2.7> Assume that the register `$t1` is initialized to the value 10. What is the value in register `$s0` assuming the `$s0` is initially zero?

2.24.2 [5] <§2.7> For each of the loops above, write the equivalent C code routine. Assume that the registers `$s1`, `$s2`, `$t1`, and `$t2` are integers A, B, i, and `temp`, respectively.

2.24.3 [5] <§2.7> For the loops written in MIPS assembly above, assume that the register `$t1` is initialized to the value N. How many MIPS instructions are executed?

2.25 [10] <§2.7> Translate the following C code to MIPS assembly code. Use a minimum number of instructions. Assume that the values of a, b, i, and j are in registers $s0, $s1, $t0, and $t1, respectively. Also, assume that register $s2 holds the base address of the array D.

```
for(i=0; i<a; i++)
    for(j=0; j<b; j++)
        D[4*j] = i + j;
```

2.26 [5] <§2.7> How many MIPS instructions does it take to implement the C code from Exercise 2.25? If the variables a and b are initialized to 10 and 1 and all elements of D are initially 0, what is the total number of MIPS instructions that is executed to complete the loop?

2.29 [30] <§2.8> Implement the following C code in MIPS assembly. Hint: Remember that the stack pointer must remain aligned on a multiple of 16.

```
int fib(int n){
    if (n==0)
        return 0;
    else if (n == 1)
        return 1;
    else
        return fib(n-1) + fib(n-2);
}
```