COMPUTER ARCHITECTURE & ASSEMBLY LANGUAGE
14:332:331
Rutgers University
Spring 2017

Homework 2: Due Friday March 3d (preferably in class or through sakai)

Total points: 100

1. 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:
```

a) Assume that the register $t1 is initialized to the value 10. What is the value in register $s2 assuming $s2 is initially zero?

b) For each of the loop above, write the equivalent C code routine. Assume that the registers $s1, $s2, $t1, and $t2 are integers A, B, i, and temp, respectively

c) For the loop written in MIPS assembly above, assume that the register $t1 is initialized to the value N. How many MIPS instructions are executed?

2.

a) Suppose that the current value of PC is 0x00004000. Can we use a single jump instruction to go to PC=0x20014924 ?(if yes, write the jump instruction and show the value of the immediate field in Hex. If not, use a combinations of instructions to do so and show the immediate values in Hex)

b) Suppose that the current value of PC is 0x00004000. Can we use a single branch instruction to go to PC=0x20014924? (if yes, write the branch instruction and show the value of the immediate field in Hex. If not, use a combinations of instructions to do so and show the immediate values in Hex)

c) Suppose that the current value of PC is 0x1FFFF000. Can we use a single branch instruction to go to PC=0x20014924? (if yes, write the branch instruction and show the value of the immediate field in Hex. If not, use a combinations of instructions to do so and show the immediate values in Hex)

3. Compile the assembly code for the following C code.

```
int func (int a, int b, int c){
if (a<=c)
   return 4;
else if (a<b)
   return 8
else
   return a+c
}
```

4. Compile the assembly code for the following C code.

```
int f1 (int m, int n){
        return f2(4*n+m);
}
```

5. Compile the assembly code for the following C code.

```
int f3 (int n){
   if (n>20)
      return 0;
   else if (n<=1)
       return 1;
   else return (4*f3(n-2)+2)
 }
```

6. Implement the following C code in MIPS assembly. What is the total number of MIPs instructions needed to execute the function?

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

7. Translate function f into MIPS assembly language. You may use temp registers as well.
The function declaration for func is: “int f(int a, int b);”
The code for function f is as follows:

```
Int f(int a, int b, int c, int d)  {
  Return func(func(a,b), c+d);
}
```

Can we determine the contents of registers $t1, $t2, $t3, $t4, $t5, $s3, $ra, $sp right before function f returns? Remark that for function func we only know its declaration.

8. Write the MIPS assembly code that creates the 32-bit constant 0010 0000 0000 0001 0100 1001 0010 0100 and stores that value to register $t0.
If the current value of the PC is 0x00000600, can you use a single branch instruction to get to the PC address shown above? What about if the current value of the PC is 0x1FFFF000?