University of California at Santa Barbara

ECE 154A Computer Organization

Homework #2

Due at 11:00 pm October 26\textsuperscript{th}, 2012

HOME WORK COVER SHEET

Name (Last, First)

Formatting guidelines:

1) A completed cover sheet must accompany all homework assignments
2) All homework must be stapled. Paper clips, folded corners, etc. are unacceptable.
3) All work must be eligible and clearly organized
4) No credit will be given for problems without necessary work

Turning-in homework:

1) Homework must be handed in homework box on 3\textsuperscript{rd} floor HFH
2) No late homeworks will be accepted
1. **(45 points)**

a. **(25 points)** The following program tries to copy words from the address in register $a0 to the address in register $a1, counting the number of words copied in register $v0. The program stops copying when it finds a word equal to 0. You do not have to preserve the contents of registers $v1, $a0, and $a1. This terminating word should be copied but not counted.

```
addi $v0, $zero, 0  # Initialize count
```

```
loop:  lw $v1, 0($a0)  # Read next word from source
sw $v1, 0($a1)   # Write to destination
addi $a0, $a0, 4  # Advance pointer to next source
addi $a1, $a1, 4  # Advance pointer to next destination
beq $v1, $zero, loop  # Loop if word copied != zero
```

There are multiple bugs in this MIPS program; fix them and turn in a bug-free version. Note that the easiest way check your code is to use SPIM simulator.

b. **(20 points)** Implement the following C code in MIPS:

```c
for (j = 0; j != 20; j++)
{
    a[j+2] = a[j+1] + a[j];
}
```

Assume that the base address of a array is stored in $a1. Write the code in MIPS assembly using the minimal number of registers.

2. **(10 points)**

The data below show the values stored in memory:

(i) 0xAE0B0008
(ii) 0x8D080040

(a) **(2 points)** Is it possible to say whether these values represent instructions or data?

(b) **(6 points)** Suppose that this data are MIPS instructions. What instructions do they represent? In other words show assembly code for each of them.

(c) **(2 points)** What type (I-type, J-type, or R-type) instruction do the hexadecimal entries above represent?
3. (20 points)

(a) (10 points) Convert the following MIPS instructions into corresponding hexadecimal representation

(i) add $t0, $t1, $0
(ii) lw $t1, -4($s3)
(iii) slti $s1, $v0, -1
(iv) jr $ra
(v) srl $a0, $t0, 10

(b) (5 points) What type (I-type, J-type, or R-type) instruction are shown above in (i)-(v)?

(c) (5 points) What addressing modes (immediate, register, base, PC-relative or pseudodirect) are used in (i)-(v)?

4. (25 points) Assume that the code is placed at location 0x0040 0000 in memory. Show the machine code (in hex) for this loop

```
LOOP:    lw $t0, 0($a0) 0x0040 0000:
        lw $t1, 0($a1) 0x0040 0004:
        beq $t0, $0, EXIT 0x0040 0008:
        addi $a0, $a0, 4 0x0040 000C:
        addi $a1, $a1, 4 0x0040 0010:
        bne $a0, $a2, LOOP 0x0040 0014:
EXIT:
```

5. (45 points) The sequence $F_n$ of Fibonacci numbers is defined using recurrence relation $F_n = F_{n-1} + F_{n-2}$ for $n > 2$ and assuming $F_1 = 1$ and $F_2 = 1$. In C language one can use the following recursive function to obtain any number in this sequence:

```
int fib_iter(int a, int b, int n) {
    if (n==2)
        return a;
    else
        return fib_iter(a+b, a, n-1);
}
```

(a) (25 points) Write a MIPS code for such recursive function. Assume that a, b, and n are passed with $a0$, $a1$, and $a2$, respectively, while the return value is in $v0$.

(b) (5 points) Show the context of the stack for the case $n = 4$, i.e. when MIPS function is called with $a = 1$, $b = 1$ and $n = 4$. Assume that the value of $sp$ is 0x7FFF FFFC before the function call, the function is called from the instruction at the location 0x0040 0F00 and the first instruction of the function itself is at the location 0x0040 F000.
(c) (10 points) Write more optimal code to obtain any number $F_n$, $n>2$ in a sequence, e.g., using c “do-while” style instead. Here assume that the first and the second numbers of the sequence are known already.

(d) (5 points) Derive the formula for the number of instructions which will be executed as a function of $n$ for two considered cases (assuming function is called with $a = 1, b = 1$ in the first case)