25 – 2 Computer Architecture Homework #2

Due: 4/20 17:00 p.m.

1. 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

2. Assume the following register contents:

$t0=0xAAAAAAAA, $t1=0x12345678

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
```

3. Provide a minimal set of MIPS instructions that may be used to implement the following pseudoinstruction:

```
not $t1, $t2       // bit-wise invert
```

4. Consider the following MIPS loop:

```
Loop:   slt     $t2,    $0,     $t1
        beq     $t2,    $t0,    Done
        subi    $t1,    $t1,    1
        addi    $s2,    $s2,    2
        j       Loop
Done:
```

4-1. Assume that the register $t1 is initialized to the value 10. What is the value in register $s0 assuming the $s0 is initially zero?

4-2. 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.

4-3. 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?

5. Translate the following loop into C. Assume that the C-level integer i is held in register $t1, $s2 holds the C-level integer called result, and $s0 holds the base address of the integer MemArray.

```
        addi    $t1,    $0,     0
LOOP:   lw      $s1,    0($s0)
        add     $s2,    $s2,    $s1
        addi    $s0,    $s0,    4
        addi    $t1,    $t1,    1
        slti    $t2,    $t1,    100
        bne     $t2,    $s0,    LOOP
```

6. Assume for a given process the CPI of arithmetic instructions is 1, the CPI of load/store instructions is 10, and the CPI of branch instructions is 3. Assume a program has the following instruction breakdown: 500 million arithmetic instructions, 300 million load/store instructions, 100 million branch instructions.

   6-1) Suppose that new, more powerful arithmetic instructions are added to the instruction set. On average, through the use of these more powerful arithmetic instructions, we can reduce the number of arithmetic instructions needed to execute a program by 25%, and the cost of increasing the clock cycle time by only 10%. Is this a good design choice? Why?

   6-2) Suppose that we find a way to double the performance of arithmetic instructions. What is the overall speedup of our machine? What if we find a way to improve the performance of arithmetic instructions by 10 times?