CS230: Digital Logic Design and Computer Architecture
Tutorial 02 [Mon 26 Aug, Tue 27 Aug, Thu 29 Aug]

Concepts tested: Instruction Encoding, Function Call Support, HLL Code to Process

1. An assembly program has three functions as outlined below.

```
main:
...
# read $a0, $a1
...
# set $s0, $a0, $a1
# set $s1, $t0
...
jal F
# read $v0, $s0, $a0
...
jr $ra
```

```
F:
...
# read $a0, $a1
# set $s0, $s1
# set $t0, $t1
...
jal G
...
# read $t1
# read $s0, $s1, $a0
...
jr $ra
```

```
G:
...
# set $s3
...
# set $t1
# set $ra (unusual)
...
# read $s3, $t1
...
jr $ra
```

   (a) Which registers does main have to save as caller? As callee?

   (b) Which registers does F have to save as caller? As callee?

   (c) Which registers does G have to save as caller? As callee?

2. If the number of registers in MIPS is increased to 64, what implication does it have on the instruction encoding?

3. What is the maximum array index which can be supported as a constant in a single load instruction? Assume that the array is of 32-bit integers.

4. Suppose that program P is written in 2 files p1.s & p2.s. It has no other external library. And program Q is written in 2 files q1.s and q2.s. Q has to be linked with an external library lib1.o before being executed. Answer the following questions.

   (a) While generating the object files p1.o & p2.o for P, can the assembler exchange every instance of \$s0 with \$s1 (i.e. use \$s1 wherever \$s0 appears, and vice versa)? What about while generating the object files q1.o & q2.o? You can assume (for both P and Q) that no unresolved instruction (i.e. in the relocation table) uses the two registers in question. Explain your answer briefly.

   (b) Answer the above question for the case when the two registers being exchanged are \$s0 and \$t0. Explain your answer briefly.