Score:_____ Name:_____

## ECE 3055 A Quiz 1 – Fall 2004

The following RISC assembly language program is executed on a MIPS processor. Fill in the register values that will be present, after execution of this program. A summary of MIPS instructions is included at the bottom of the page – for anyone unfamiliar with the MIPS instruction set. Prior to execution of the program, memory location 0x0300 contains 0x20303055. Note: 0x indicates hexadecimal and all answers must be in hexadecimal. A MIPS memory word or register contains 32-bits.

```
            LW        $3, 0x0300
            SRL       $4, $3, 4
            AND       $3, $4, $3
            XOR       $2, $3, $4
            LUI       $5, 55
            ORI       $5, $5, 0x50
            ADD       $6, $3, $4
            BEQ       $3, $6, LABEL1
            ADDI      $6, $6, -10
LABEL1:     SW        $6, 0x0300
```

After execution of the MIPS code sequence above,

R2 = 0x__02030300__ ( *in hexadecimal* )

R3 = 0x__0000005__ ( *in hexadecimal* )

R4 = 0x__02030305__ ( *in hexadecimal* )

R5 = 0x__00370050__ ( *in hexadecimal* )

Memory Location 0x0300 contains: 0x__02030300__ ( *in hexadecimal* )

The MIPS processor contains thirty-two 32-bit registers, $0 through $31. $0 always contains a zero. By default, all arithmetic operations use two's complement arithmetic.

| MIPS Instruction | | | Meaning |
|---|---|---|---|
| ADD | Rd, Rs, Rt | - | Rd = Rs + Rt  (R – register ($) ) |
| AND | Rd, Rs, Rt | - | Rd = Rs bitwise logical AND Rt  (R – register ($) ) |
| ORI | Rd, Rs, *Immed* | - | Rd = Rs bitwise logical OR *Immediate* value |
| LUI | Rd, *Immed* | - | Rd = *Immediate* value high 16-bits, 0's low 16-bits |
| BEQ | Rs, Rt, *address* | - | Branch to *address*, only if Rs equal to Rt |
| LW | Rd, *address* | - | LOAD - Rd gets contents of memory at *address* |
| SRL | Rd, Rs, *count* | - | Shift right logical (*use 0 fill*) by *count* bits |
| SUB | Rd, Rs, Rt | - | Rd = Rs - Rt |
| SW | Rd, *address* | - | STORE - memory at *address* gets contents of Rd |
| XOR | Rd, Rs, Rt | - | Rd = Rs bitwise logical XOR Rt |