Score:_____ Name:_____

ECE 3055 A Quiz 1 – Spring 2009

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 0x01000 contains 0x30552031. Note: 0x indicates hexadecimal and all answers must be in hexadecimal, default is decimal in the MIPS assembly language source file. A MIPS memory word or register contains 32-bits. Use XXXXXXXX for an undefined value.

```
            LW      $3, 0x01000
            SRL     $4, $3, 4
            ADD     $3, $3, $4
            XOR     $2, $3, $4
            LUI     $5, 0x3035
            ORI     $5, $5, 100
            SUB     $6, $4, $3
            BLT     $6, $2, LABEL1
            ADDI    $6, $0, -8
LABEL1:     SW      $6, 0x01000
```

After execution of the MIPS code sequence above,

R2 = 0x 305F2037 (*in hexadecimal*)

R3 = 0x 335A7234 (*in hexadecimal*)

R4 = 0x 03055203 (*in hexadecimal*)

R5 = 0x 30350064 (*in hexadecimal*)

Memory Location 0x01000 contains: 0x CFA ADFCF (*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. Assume no branch delay slot is present.

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 = 16-bit *Immediate* value high 16-bits, 0's low 16-bits
BLT	Rs, Rt, *address*	-	Branch to *address*, only if Rs less than 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