ECE 3055 – Spring 2010

The following RISC assembly language program is executed on a 32-bit 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 0x02000 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, 0x02000
	sll	$4, $3, 10
	sub	$2, $4, $3
	xor	$3, $4, $2
	lui	$5, 0
	ori	$5, $5, 12373
	sub	$6, $4, $3
	bne	$3, $6, LABEL1
	addi	$6, $0, -25
LABEL1:	sw	$6, 0x02000

After execution of the MIPS code sequence above,

R2 = 0x 242ba3CF (*in hexadecimal*)

R3 = 0x 70ab67cF (*in hexadecimal*)

R4 = 0x 5480c400 (*in hexadecimal*)

R5 = 0x 00003055 (*in hexadecimal*)

Memory Location 0x02000 contains: 0x E3D55C31 (*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*
ADDI	Rd, Rs, *Immed*	-	Rd = Rs + *Immediate* value
ADD	Rd, Rs, Rt	-	Rd = Rs + Rt
ORI	Rd, Rs, *Immed*	-	Rd = Rs low 16-bits bitwise logical OR *Immediate* value
LUI	Rd, *Immed*	-	Rd = 16-bit *Immediate* value high 16-bits, 0's low 16-bits
BNE	Rs, Rt, *address*	-	Branch to *address*, only if Rs not equal to Rt
LW	Rd, *address*	-	LOAD - Rd gets contents of memory at *address*
SLL	Rd, Rs, *count*	-	Shift left 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