Name: KEY **Id#**

ICS 233, Term 142

Computer Architecture & Assembly Language

Quiz# 3

Date: Tuesday, March 17, 2015

Q1. Assuming that functions F and G receive two arguments in $a0 and $a1 and return their results in $v0, implement the function F given below saving needed registers on the stack. Save changed registers according to the assumed programming convention.

```
int F(int a, int b) {
        return  a+G(b, G(a, b));
}
```

```
F:      addiu   $sp, $sp, -12    # frame = 12 bytes
        sw      $ra, 0($sp)      # save $ra
        sw      $a0, 4($sp)      # save argument a
        sw      $a1, 8($sp)      # save argument b
        jal     G                # call g(a,b)
        lw      $a0, 8($sp)      # $a0 = b
        move    $a1, $v0         # $a1 = g(a,b)
        jal     G                # call g(b, g(a,b))
        lw      $a0, 4($sp)      # $a0 = a
        addu    $v0, $a0, $v0    # $v0 = a+G(b, G(a, b))
        lw      $ra, 0($sp)      # restore $ra
        addiu   $sp, $sp, 12     # free stack frame
        jr      $ra              # return to caller
```

Q2. Given that **Multiplicand=1010** and **Multiplier=0111,** using the **refined signed multiplication** hardware, show the **signed** multiplication of **Multiplicand** by **Multiplier**. The result of the multiplication should be an 8 bit **signed** number in HI and LO registers. Show the steps of your work.

Iteration		**Multiplicand**	**Sign**	**Product = HI,LO**
0	Initialize (LO = Multiplier)	1010		0000 011**1**
1	LO[0] = 1 => ADD		1	1010 0111
	Shift Product = (HI, LO) right 1 bit	1010		1101 001**1**
2	LO[0] = 1 => ADD		1	0111 0011
	Shift Product = (HI, LO) right 1 bit	1010		1011 100**1**
3	LO[0] = 1 => ADD		1	0101 1001
	Shift Product = (HI, LO) right 1 bit	1010		1010 110**0**
4	LO[0] = 0 => Do nothing		1	1010 1100
	Shift Product = (HI, LO) right 1 bit			**1101 0110**

Q3. Given that **Dividend=1011** and **Divisor=0011**, Using the **refined unsigned division** hardware, show the **unsigned** division of **Dividend** by **Divisor**. The result of division should be stored in the Remainder and Quotient registers. Show the steps of your work.

Iteration		**Remainder (HI)**	**Quotient (LO)**	**Divisor**	**Difference**
0	Initialize	0000	1011	0011	
1	1: SLL, Difference	0001	0110	0011	1110
	2: Diff < 0 => Do Nothing				
2	1: SLL, Difference	0010	1100	0011	1111
	2: Diff < 0 => Do Nothing				
3	1: SLL, Difference	0101	1000	0011	0010
	2: Rem = Diff, set lsb Quotient	0010	1001		
4	1: SLL, Difference	0101	0010	0011	0010
	2: Rem = Diff, set lsb Quotient	**0010**	**0011**		