

CS6107 - Computer Architecture Integrated Lab

Batch : P

LAB 3 - Spot questions

Integer Addition and Subtraction Instructions

For these programming exercises, use only those instructions that have been discussed so far in these notes:

add	sll
addi	srl
addiu	sub
addu	subu
and	nor
andi	xor
or	xori
ori	

In the *Settings* menu of SPIM set Bare Machine ON, Allow Pseudo Instructions OFF, Load Trap File OFF, Delayed Branches ON, Delayed Loads ON, Mapped IO OFF, Quiet OFF.

Run the programs by setting the value of the PC to 0x400000 and then single stepping (pushing F10) or by multiple stepping (push F11 and enter a number of steps). Observing the results in the SPIM window.

Questions

1. Write an assembly language using arithmetic expression

Let register \$8 be x and register \$9 be y. Write a program to evaluate:

$$3x - 5y$$

Leave the result in register \$10. Inspect the register after running the program to check that the program works. Run the program several times, initialize x and y to different values for each run.

For these programming exercises, use only those instructions that have been discussed so far in these notes:

add	mfhi	sll
addi	mflo	sra
addiu	mult	srl
addu	multu	sub
and	nor	subu
andi	or	xor
div	ori	xori
divu		

In the *Settings* menu of SPIM set Bare Machine ON, Allow Pseudo Instructions OFF, Load Trap File OFF, Delayed Branches ON, Delayed Loads ON, Mapped IO OFF, Quiet OFF.

Run the programs by setting the value of the PC to 0x400000 and then single stepping (pushing F10) or by multiple stepping (push F11 and enter a number of steps). Observing the results in the SPIM window.

Integer Multiplication, Division, and Arithmetic Shift

2. Write a program that determines the value of the following expression:

$$(x*y)/z$$

Use $x = 1600000$ (=0x186A00), $y = 80000$ (=0x13880), and $z = 400000$ (=61A80). Initialize three registers to these values. Since the immediate operand of the `ori` instruction is only 16 bits wide, use shift instructions to move bits into the correct locations of the registers.

Choose wisely the order of multiply and divide operations so that the significant bits always remain in the **lo** result register.