

5.3 sub, mul: Subtraction and multiplication instructions

Subtract instruction: sub

A **subtract instruction** (**sub**) computes the difference of two register values, and writes the difference into a register. A MIF instruction format is shown below, which computes $\text{regA} = \text{regB} - \text{regC}$.

```
sub regA, regB, regC
```

PARTICIPATION ACTIVITY

5.3.1: sub instruction.

Assume initial register values of:

- \$t0: 30
- \$t1: 10
- \$t2: -5
- \$t3: 5

1) After the following, what is \$t4?

```
sub $t4, $t1, $t3
```

Check

Show answer

2) After the following, what is \$t0?

```
sub $t0, $t0, $t3
```

Check**Show answer**

3) After the following, what is \$t5?

```
sub $t5, $t1, $t2
```

Check**Show answer**

4) Type an instruction to subtract \$t3 from \$t4, writing the difference to \$t5.

Check**Show answer**

A brief note from your instructor:

Note: "mul" is the first of many MIPS pseudoinstructions that are used in the chapter. However, I believe it is important to use real MIPS instructions to accomplish these tasks (and will thus test that on examinations).

Therefore, it is important to know which instructions are real and which are pseudo. If you are ever in doubt, all real MIPS can be found in the MIPS Instruction Reference here: <http://www.mrc.uidaho.edu/mrc/people/jff/digital/MIPSir.html>

Multiply instruction: mul

A **multiply instruction** (*mul*) computes the product of two register values, and writes the product into a register. A MIPS multiply instruction format is shown below, which computes $\text{regA} = \text{regB} * \text{regC}$. The multiply instruction computes a 32-bit product, and ignores

that may result from multiplying two 32-bit values.

```
mul regA, regB, regC
```

**PARTICIPATION
ACTIVITY**

5.3.2: mul instruction.

Assume initial register values of:

- \$t0: 40
- \$t1: 20
- \$t2: 70000
- \$t3: 30

1) After the instruction `mul $t3, $t1, $t0` what is \$t3?

- ☐ 600
- ☐ 800

2) Which instruction multiplies \$t4 with \$t5, writing the product to \$t5.

- ☐ `mul $t5, $t5, $t4`
- ☐ `mul $t4, $t5, $t5`

3) Does the following instruction result in an overflow?

```
mul $t5, $t1, $t2
```

- ☐ Yes
- ☐ No

4) Does the following instruction result in an overflow?

mul \$t5, \$t2, \$t2

☐ Yes

☐ No

Pseudoinstructions

A **native instruction** is an assembly instruction directly supported by a processor's hardware. A **pseudoinstruction** is an assembly instruction that must be replaced by one or more native instructions before being executed. Pseudoinstructions are used to keep the number of native instructions small, which leads to more efficient processor hardware, while providing programmers a large set of instructions for common operations. The MIPS mul instruction is a pseudoinstruction implemented using mult and mflo native instructions, discussed elsewhere.

PARTICIPATION ACTIVITY

5.3.3: sub and mul instructions.

The assembly program below calculates the total taxi fare as \$15 plus \$2 per mile traveled. DM[5000] holds the total miles traveled, and the total fare is stored in DM[5004].

1. Run the simulation step-by-step, observing memory values.
2. Change DM[5000]'s value to 25, then run again.
3. Modify the program to subtract \$2 for a frequent rider discount, store the discounted total fare in DM[5008].

Assembly

Line 1

addi \$t5, \$zero, 5000

Line 2

lw \$t0, 0(\$t5)

Line 3

addi \$t1, \$zero, 2

Line 4

mul \$t2, \$t0, \$t1

Line 5

addi \$t2, \$t2, 15

Line 6

addi \$t5, \$zero, 5004

Line 7

sw \$t2, 0(\$t5)

Registers

\$zero

0

\$t0

0

\$t1

0

\$t2

0

\$t5

0

+

5000

5004

5008

+

D:

ENTER SIMULATION

STEP

RUN

More options

▼

Instruction format summary: sub, mul

Table 5.3.1: Instruction summary: sub, mul.

Instruction	Format	Description	Example
sub	sub \$a, \$b, \$c	Subtract: Subtracts \$c from \$b, and writes the difference into register \$a.	sub \$t3, \$t2, \$t5
mul	mul \$a, \$b, \$c	Multiply: Multiplies register \$b and \$c, and writes the lower 32-bits of the product into register \$a.	mul \$t3, \$t2, \$t1

**CHALLENGE
ACTIVITY**

5.3.1: Subtract and multiply instructions.

Start

Compute: $\$t4 = \$t6 - \$t5$, ,

Registers

\$t4	0
\$t5	9
\$t6	5

1	2	3	4
---	---	---	---

Check

Next

 [Provide feedback on this section](#)