2.4 addi, add: Add instructions

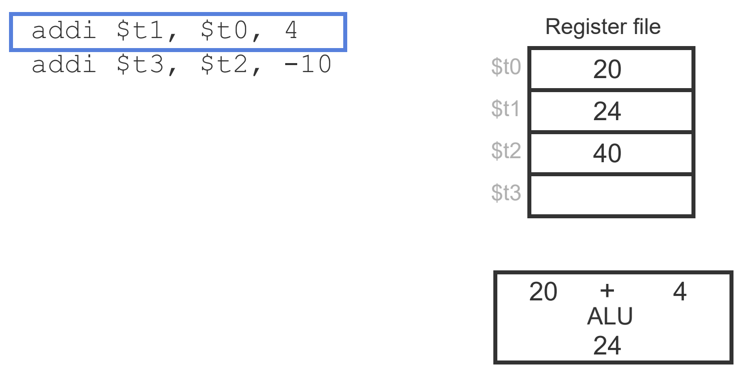
Add with immediate instruction: addi

A program often needs to add a specific value to a register, such as adding register $t3 and 4. An ***add immediate*** (***addi***) instruction adds a register's value and an immediate value. An ***immediate*** is a value specified within an instruction. In MIPS, the immediate is a 16-bit number that can range from -32,768 to 32,767. A MIPS addi instruction format is shown below, which computes regA = regB + immediate.

```
addi regA, regB, immediate
```

PARTICIPATION ACTIVITY | 2.4.1: Add immediate (addi) instruction.

1 2 2x speed



```
addi $t1, $t0, 4
addi $t3, $t2, -10
```

The add immediate instruction adds the immediate value 4 to the value held in $t0. The sum is written to register $t1.

Captions

1. The add immediate instruction adds the immediate value 4 to the value held in $t0. The sum is written to register $t1.
2. An immediate value can be negative. -10 is added to the value held in $t2, and the result 40 + -10 or 30 is written to $t3.

Feedback?

PARTICIPATION ACTIVITY

2.4.2: addi instruction.

For each question, assume initial register values of:

- $t0: 20
- $t1: 50
- $t2: 60

1) After the following, what is $t4?

```
addi $t4, $t2, 1
```

61

Check Show answer

✓ **Correct**

61

$t4 = 60 + 1 = 61

2) After the following, what is $t3?

```
addi $t3, $t1, -5
```

45

Check Show answer

✓ **Correct**

45

$t3 = 50 + -5 = 45. No subi instruction exists, because subtracting an immediate is easily done by adding a negative immediate.

3) After the following, what is $t2?

```
addi $t2, $t2, 6
```

66

Check Show answer

✓ **Correct**

66

$t2 = 60 + 6 = 66. The initial value of $t2, 60, is overwritten with the sum 66.

4) Type an addi instruction that writes $t5 with the sum of $t4 and 17.

addi $t5, $t4, 17

Check Show answer

✓ **Correct**

addi $t5, $t4, 17

The instruction adds the value in $t4 and the immediate value 17, writing the sum to $t5.

5) Type an instruction that adds 3 to $t4, writing the sum to $t4.

addi $t4, $t4, 3

Check Show answer

Correct

addi $t4, $t4, 3

The immediate value 3 is added to the value in $t4, and the sum is written back $t4.

Feedback?

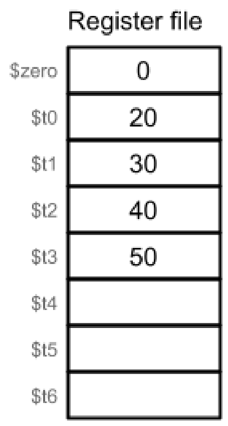
Commonly, a specific value needs to be written to a register. The addi instruction format below computes regA = immediate:

```
addi regA, $zero, immediate
```

Since $zero always holds the value 0, the sum is equal to the immediate value, and the immediate value is written to the register.

PARTICIPATION ACTIVITY 2.4.3: Initializing registers with addi.

Given the following register file contents, match the register to the value held in the register as the provided instructions.


Register file

$zero	0
$t0	20
$t1	30
$t2	40
$t3	50
$t4	
$t5	
$t6	

```
addi $t4, $zero, 40
addi $t3, $t0, 0
addi $t2, $zero, 50
```

If unable to drag and drop, refresh the page.

50	$t2 $t2 = 0 + 50 = 50	Correct

20	$t3 $t3 = $t0 + 0 = 20 + 0 = 20	Correct
40	$t4 $t4 = 0 + 40 = 40	Correct

Reset

Feedback?

Add instruction: add

An ***add instruction*** computes the sum of two register values, and writes the sum into a register. A MIPS add instruction format is shown below, which computes regA = regB + regC.

```
add regA, regB, regC
```

The register written by an instruction is called the ***destination register***. A register read by an instruction is called a ***source register***. For the add instruction, regA is the destination register, and regB and regC are source registers.

PARTICIPATION ACTIVITY | 2.4.4: Add instruction.

Assume initial register values of

- $t0: 20
- $t1: 30
- $t2: 40

1) After the following, what is $t0?

```
add $t0, $t1, $t2
```

70

Check Show answer

Correct

70

$t0 = 30 + 40, so $t0 = 70. The initial value of $t0, namely 20, is simply overwritten.

2) After the following, what is $t2?

```
add $t0, $t1, $t2
```

40

Check Show answer

Correct

40

The instruction reads $t1 and $t2, and writes $t0. Reading a register does not change the register's value. Thus, $t2's initial value of 40 does not change.

✓

3) After the following, what is $t2?

```
add $t2, $t1, $t0
```

50

Check Show answer

Correct

50

The destination register is listed first. So, the first register listed, $t2, gets the sum of the second and third registers listed. Thus, $t2 = 30 + 20, so $t2 = 50.

✓

4) After the following, what is $t2?

```
add $t2, $t0, $t1
```

50

Check Show answer

Correct

50

The first register listed, $t2, gets the sum of the second and third registers listed. Thus, $t2 = 20 + 30, or 50. The order of the second and third registers doesn't matter. `add $t2, $t1, $t0` yields the same result of 50 in $t2.

✓

5) Type an instruction that writes $t3 with the sum of $t5 and $t6.

add $t3, $t5, $t6

Check Show answer

Correct

add $t3, $t5, $t6 or add $t3, $t6, $t5

The register being written, $t3, must appear first. The second and third registers can be either $t5, $t6, or $t6, $t5.

✓

Feedback?

Table 2.4.1: Instruction summary: addi, add.

Instruction	Format	Description	Example
addi	`addi $a, $b, C`	Add immediate: Adds register $b and the immediate value C, and	`addi $t3, $t2, 7`

		writes the sum into register $a.	
add	`add $a, $b, $c`	Add: Computes the sum of registers $b and $c, and writes the sum into register $a.	`add $t4, $t1, $t2`

Feedback?

CHALLENGE ACTIVITY 2.4.1: Add immediate and add instructions.

547404.4091098.qx3zqy7

Jump to level 1

1 ✓ 2 ✓ 3 ✓ 4 ✓

Compute: $t5 = $t4 + $t3 + 8

```
add  $t5, $t4, $t3
add  $t5, $t5, $t5
```

Registers

$t3	5
$t4	3
$t5	0

1 2 3 **4**

Check Next

Done. Click any level to practice more. Completion is preserved.

✓ Your code with comments:

```
add $t5, $t4, $t3   # $t5 = $t4 + $t3

add $t5, $t5, $t5   # $t5 = $t5 + $t5
```

Your code resulted in:

Registers

$t3	5
$t4	3
$t5	16

Feedback?

How was this section?

Provide section feedback