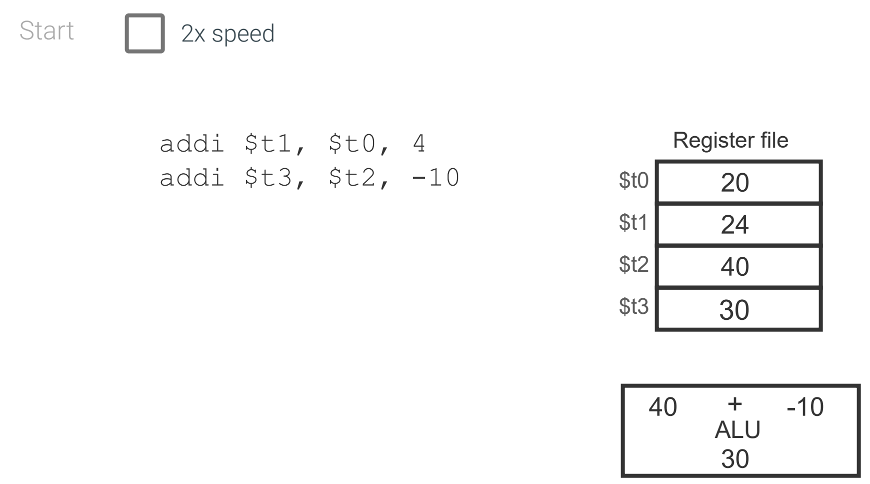
5.2 addi, add: Add instructions

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***) ins
register's value and an immediate value. An ***immediate*** is a value specified within an instruction. In MIPS, the immediate is
that can range from -32,768 to 32,767. A MIPS addi instruction format is shown below, which computes regA = regB + imm

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

PARTICIPATION ACTIVITY 5.2.1: Add immediate (addi) instruction.



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

Register file

$t0	20
$t1	24
$t2	40
$t3	30

PARTICIPATION ACTIVITY 5.2.2: 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
```

Check Show answer

2) After the following, what is $t3?

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

Check Show answer

3) After the following, what is $t2?

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

Check Show answer

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

Check Show answer

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

Check Show answer

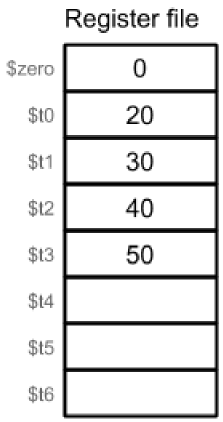
Commonly, a specific value needs to be written to a register. The addi instruction format below computes regA = immediat

```
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 re

PARTICIPATION ACTIVITY 5.2.3: 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
```

40 50 20

$t2

$t3

$t4

Reset

Add instruction: add

An ***add instruction*** computes the sum of two register values, and writes the sum into a register. A MIPS add instruction for 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 r*** add instruction, regA is the destination register, and regB and regC are source registers.

PARTICIPATION ACTIVITY 5.2.4: Add instruction.

Assume initial register values of

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

1) After the following, what is $t0?

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

Check Show answer

2) After the following, what is $t2?

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

Check Show answer

3) After the following, what is $t2?

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

Check Show answer

4) After the following, what is $t2?

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

Check Show answer

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

Check Show answer

Table 5.2.1: Instruction summary: addi, add.

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

CHALLENGE ACTIVITY 5.2.1: Add immediate and add instructions.

Start

Compute: $t6 = $t5 + 3

`addi ▼` `$t5 ▼`, `$t5 ▼`, `0`

Registers

$t5	2
$t6	0

1 2 3 4

Check Next

1
2
3
4

Provide feedback on this section