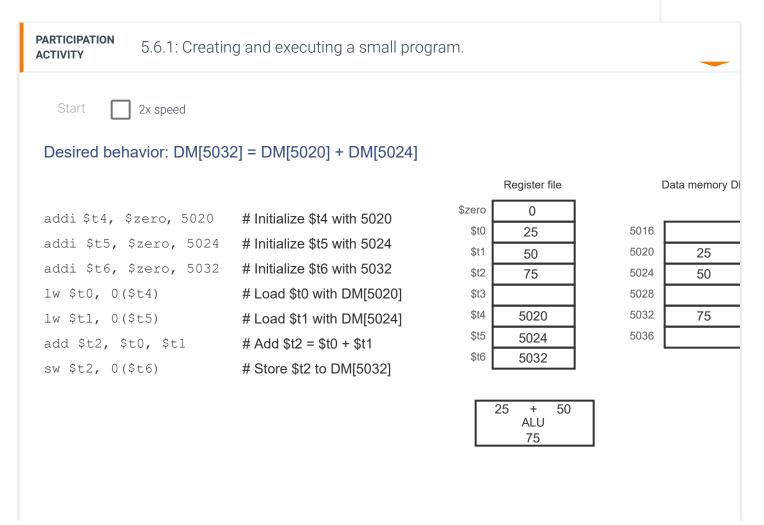
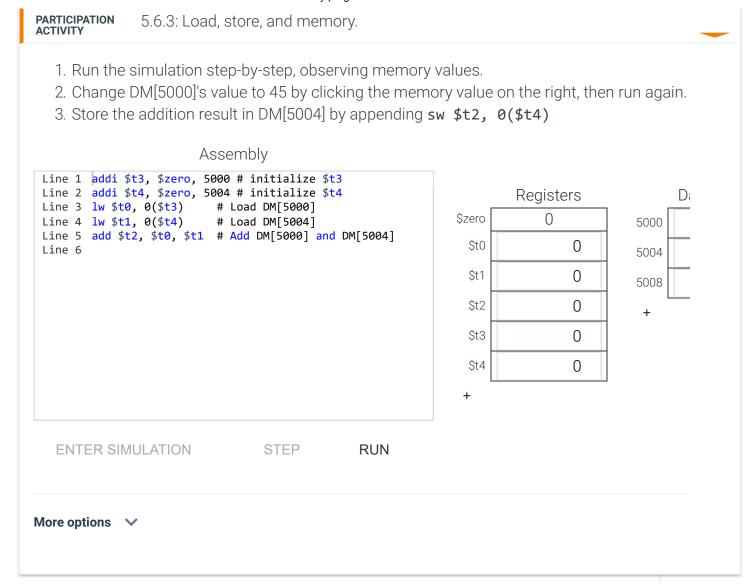
## 5.6 A small assembly program

## Creating and executing a small program

Given desired behavior, a programmer must create an instruction sequence implementing such behavior, using only the programmer must create an instruction sequence implementing such behavior, using only the programmer must create an instruction sequence implementing such behavior, using only the programmer must create an instruction sequence implementing such behavior, using only the programmer must create an instruction sequence implementing such behavior, using only the programmer must create an instruction sequence implementing such behavior, using only the programmer must create an instruction sequence implementing such behavior, using only the programmer must create an instruction sequence implementing such behavior.



```
PARTICIPATION
                 5.6.2: Creating small programs.
 ACTIVITY
Indicate the WRONG item. DM refers to data memory. Assume initial values:
$t0: 5000
$t1: 5004
$t2: 5008.
   Desired behavior: DM[5000] = DM[5004] +
1) DM[5008]
   lw $t3, 0($t1) # Load DM[5004]
   lw $t4,0($t2) #Load DM[5008]
   add $t5, $t3 , $t4 # Add DM[5004] + DM[5008]
   sw $t5, 0( $t1 ) # Store result into DM[5000]
   Desired behavior: DM[5000] = DM[5000] +
2) DM[5004]
   lw $t3, 0($t0)
   lw $t4, 0( $t2 )
   add $t3 , $t3, $t4
   sw $t3, 0( $t0 )
   Desired behavior: DM[5008] = DM[5004] +
3) DM[5004] + DM[5004]
   lw $t3, 0($t1)
    add $t4, $t3, $t3
add $t4, $t4, $t4
    sw $t4, 0($t2)
```



## Conserving registers in assembly programs

Registers are limited, so programmers should conserve registers. If a value in a register is not read later, the register can be writing another value. Ex: Assume \$t4 holds a memory address used in a lw instruction. If that memory address is not used instruction, \$t4 can be reused to hold a different memory address or used to hold the result of a computation.

5.6.4: Conserving registers. **ACTIVITY** Start 2x speed Desired behavior: DM[5032] = DM[5020] + DM[5024] addi \$t4, \$zero, 5020 addi \$t4, \$zero, 5020 addi \$t5, \$zero, 5024 lw \$t1, 0 (\$t4) addi \$t6, \$zero, 5032 addi \$t4, \$zero, 5024 lw \$t1, 0(\$t4) lw \$t2, 0(\$t4) lw \$t2, 0(\$t5) add \$t3, \$t2, \$t1 add \$t3, \$t2, \$t1 addi \$t4, \$zero, 5032 sw \$t3, 0 (\$t6) sw \$t3, 0(\$t4) 3 registers used for 3 1 register reused for all 3 memory different memory addresses addresses, conserving \$t5 and \$t6

PARTICIPATION ACTIVITY

5.6.5: Conversing register in assembly programs.

1) Which register can be reused in the addi and sw instructions to store \$t6 to memory address 5048?

addi \$t2, \$zero, 5000 lw \$t4, 0(\$t2) add \$t6, \$t6, \$t4 sw \$t6, 0(\_\_\_)

- O \$t2
- **O** \$t6

2) Which register can be reused in the addi and lw instructions to load \$t5 with data at memory address 5012?

```
addi $t1, $zero, 5000

lw $t4, 0($t1)

addi $t2, $zero, 5008

lw $t5, 0($t2)

addi ____, $zero, 5012

lw $t6, 0(___)

add $t4, $t4, $t5

sw $t4, 0($t2)
```

- O \$t2
- 3) If the add instruction's destination register is changed from \$13 to \$12, what other instruction must be updated?

addi \$t4, \$zero, 5020 lw \$t1, 0(\$t4) addi \$t4, \$zero, 5024 lw \$t2, 0(\$t4) add \$t3, \$t2, \$t1 addi \$t4, \$zero, 5032 sw \$t3, 0(\$t4)

- O lw \$t2, 0(\$t4)
- O sw \$t3, 0(\$t4)

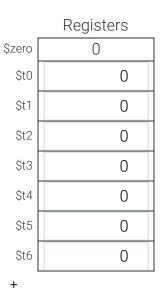
PARTICIPATION ACTIVITY

5.6.6: Conserving registers used for memory addresses.

The program below computes DM[5000] = DM[5000] + DM[5004] + DM[5008]

- 1. Run the simulation step-by-step, observing that program uses \$t4, \$t5, and \$t6 for the three memory addresses 5000, 5004, and 5008.
- 2. Revise the program to only use \$t4 for the memory addresses. Before each Iw or sw instruction, add an addi instruction to initialize \$t4 with the memory address. Then, use \$t4 in the Iw or sw instructions.

## Assembly



5000

5004

5008

ENTER SIMULATION

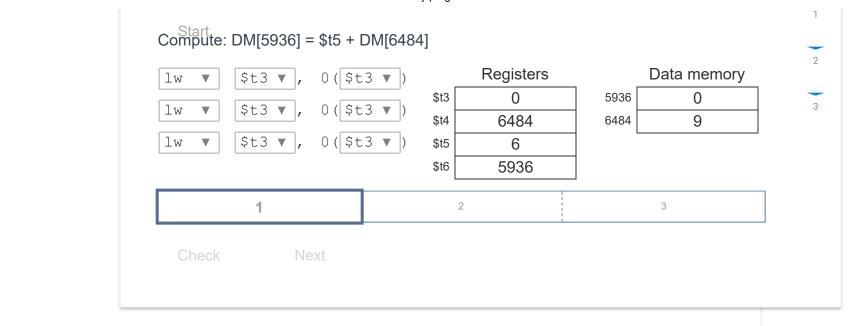
STEP

RUN

More options 💙

CHALLENGE ACTIVITY

5.6.1: Load, store, add, and addi instructions.



Provide feedback on this section