

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 pro available instructions.

**PARTICIPATION
ACTIVITY**

5.6.1: Creating and executing a small program.

Start ☐ 2x speed

Desired behavior: $DM[5032] = DM[5020] + DM[5024]$

```
addi $t4, $zero, 5020    # Initialize $t4 with 5020
addi $t5, $zero, 5024    # Initialize $t5 with 5024
addi $t6, $zero, 5032    # Initialize $t6 with 5032
lw  $t0, 0($t4)          # Load $t0 with DM[5020]
lw  $t1, 0($t5)          # Load $t1 with DM[5024]
add $t2, $t0, $t1        # Add $t2 = $t0 + $t1
sw  $t2, 0($t6)          # Store $t2 to DM[5032]
```

Register file	
\$zero	0
\$t0	25
\$t1	50
\$t2	75
\$t3	
\$t4	5020
\$t5	5024
\$t6	5032

Data memory D	
5016	
5020	25
5024	50
5028	
5032	75
5036	

25	+	50
ALU		
75		

**PARTICIPATION
ACTIVITY**

5.6.2: Creating small programs.

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)`

PARTICIPATION
ACTIVITY

5.6.3: Load, store, and memory.

1. Run the simulation step-by-step, observing memory values.
2. Change DM[5000]'s value to 45 by clicking the memory value on the right, then run again.
3. Store the addition result in DM[5004] by appending `sw $t2, 0($t4)`

Assembly

```

Line 1 addi $t3, $zero, 5000 # initialize $t3
Line 2 addi $t4, $zero, 5004 # initialize $t4
Line 3 lw $t0, 0($t3)      # Load DM[5000]
Line 4 lw $t1, 0($t4)      # Load DM[5004]
Line 5 add $t2, $t0, $t1   # Add DM[5000] and DM[5004]
Line 6

```

Registers		DM
\$zero	0	5000
\$t0	0	5004
\$t1	0	5008
\$t2	0	+
\$t3	0	
\$t4	0	
+		

ENTER SIMULATION

STEP

RUN

More options ▼

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 in a later instruction, \$t4 can be reused to hold a different memory address or used to hold the result of a computation.

ACTIVITY

5.6.4: Conserving registers.

Start ☐ 2x speedDesired behavior: $DM[5032] = DM[5020] + DM[5024]$

```

addi $t4, $zero, 5020
addi $t5, $zero, 5024
addi $t6, $zero, 5032
lw $t1, 0($t4)
lw $t2, 0($t5)
add $t3, $t2, $t1
sw $t3, 0($t6)

```

3 registers used for 3
different memory addresses

```

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)

```

1 register reused for all 3 memory
addresses, conserving \$t5 and \$t6

PARTICIPATION
ACTIVITY

5.6.5: Conserving 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(____)

```

- ☐ \$t2
☐ \$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)
```

☐ \$t1

☐ \$t2

- 3) If the add instruction's destination register is changed from \$t3 to \$t2, 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)
```

☐ lw \$t2, 0(\$t4)

☐ sw \$t3, 0(\$t4)

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 lw or sw instruction, add an addi instruction to initialize \$t4 with the memory address. Then, use \$t4 in the lw or sw instructions.

Assembly

```

Line 1 addi $t4, $zero, 5000 # initialize $t4
Line 2 addi $t5, $zero, 5004 # initialize $t5
Line 3 addi $t6, $zero, 5008 # initialize $t6
Line 4 lw $t0, 0($t4)      # Load DM[5000]
Line 5 lw $t1, 0($t5)      # Load DM[5004]
Line 6 lw $t2, 0($t6)      # Load DM[5008]
Line 7 add $t3, $t0, $t1    # Add DM[5000] and DM[5004]
Line 8 add $t3, $t3, $t2    # Add sum and DM[5008]
Line 9 sw $t3, 0($t4)      # Store sum to DM[5000]

```

Registers

\$zero	0	5000
\$t0	0	5004
\$t1	0	5008
\$t2	0	+
\$t3	0	
\$t4	0	
\$t5	0	
\$t6	0	

ENTER SIMULATION

STEP

RUN

More options ▼

CHALLENGE ACTIVITY

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

Start
Compute: $DM[5936] = \$t5 + DM[6484]$

lw ▾

\$t3 ▾

, 0 (

\$t3 ▾

)

lw ▾

\$t3 ▾

, 0 (

\$t3 ▾

)

lw ▾

\$t3 ▾

, 0 (

\$t3 ▾

)

Registers	
\$t3	0
\$t4	6484
\$t5	6
\$t6	5936

Data memory	
5936	0
6484	9

1

2

3

Check Next

 Provide feedback on this section