

## Ex1:

- Chương trình:

```

/home/thuha/Desktop/TH_KTMT/W1/Exercise1.asm* - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute Exercise1.asm* Exercise2.asm Exercise3.asm Exercise4.asm Exercise5.asm Exercise6.asm
1 #laboratory Exercise 2, Assignment 1
2 .text
3 addi $s0,$zero,0x3007      # $s0 = 0 + 0x2110003d :I-type
4 add $s0,$zero,$0            # $s0 = 0 + 0 = 0 :R-type
5

Registers Coproc 1 Coproc 0
Name Number Value
$zero 0 0x00000000
$1 1 0x00000000
$y0 2 0x00000000
$y1 3 0x00000000
$y0 4 0x00000000
$y1 5 0x00000000
$y2 6 0x00000000
$y3 7 0x00000000
$y0 8 0x00000000
$y1 9 0x00000000
$y2 10 0x00000000
$y3 11 0x00000000
$y4 12 0x00000000
$y5 13 0x00000000
$y6 14 0x00000000
$y7 15 0x00000000
$y0 16 0x00000000
$y1 17 0x00000000
$y2 18 0x00000000
$y3 19 0x00000000
$y4 20 0x00000000
$y5 21 0x00000000
$y6 22 0x00000000
$y7 23 0x00000000
$y8 24 0x00000000
$y9 25 0x00000000
$y0 26 0x00000000
$y1 27 0x00000000
$gp 28 0x00000000
$sp 29 0x7fffffe
$fp 30 0x00000000
$ra 31 0x00000000
pc 32 0x00000000
hi 33 0x00000000
lo 34 0x00000000

```

Mars Messages Run I/O

Assemble: assembling /home/thuha/Desktop/TH\_KTMT/W1/Exercise6.asm  
Assemble: operation completed successfully.  
Step: execution terminated due to null instruction.

Clear Assemble: assembling /home/thuha/Desktop/TH\_KTMT/W1/Exercise1.asm  
Assemble: operation completed successfully.

```

/home/thuha/Desktop/TH_KTMT/W1/Exercise1.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute Text Segment Data Segment
Bkpt Address Code Basic Source
0x00400000 0x20103007 addi $16,$0,0x00003007 3: addi $s0,$zero,0x3007 # $s0 = 0 + 0x2110003d :I-type
0x00400004 0x00008020 add $16,$0,$0 4: add $s0,$zero,$0 # $s0 = 0 + 0 = 0 :R-type

Registers Coproc 1 Coproc 0
Name Number Value
$zero 0 0x00000000
$y1 1 0x00000000
$y0 2 0x00000000
$y1 3 0x00000000
$y0 4 0x00000000
$y1 5 0x00000000
$y2 6 0x00000000
$y3 7 0x00000000
$y0 8 0x00000000
$y1 9 0x00000000
$y2 10 0x00000000
$y3 11 0x00000000
$y4 12 0x00000000
$y5 13 0x00000000
$y6 14 0x00000000
$y7 15 0x00000000
$y0 16 0x00000000
$y1 17 0x00000000
$y2 18 0x00000000
$y3 19 0x00000000
$y4 20 0x00000000
$y5 21 0x00000000
$y6 22 0x00000000
$y7 23 0x00000000
$y8 24 0x00000000
$y9 25 0x00000000
$y0 26 0x00000000
$y1 27 0x00000000
$gp 28 0x00000000
$sp 29 0x7fffffe
$fp 30 0x00000000
$ra 31 0x00000000
pc 32 0x00000000
hi 33 0x00000000
lo 34 0x00000000

Mars Messages Run I/O
Assemble: operation completed successfully.  
Step: execution terminated due to null instruction.  
Clear Assemble: assembling /home/thuha/Desktop/TH_KTMT/W1/Exercise1.asm  
Assemble: operation completed successfully.  
Assemble: assembling /home/thuha/Desktop/TH_KTMT/W1/Exercise1.asm  
Assemble: operation completed successfully.

```

- Chạy từng dòng lệnh:

Step 1:

The screenshot shows the MARS 4.5 assembly debugger interface. The top menu bar includes File, Edit, Run, Settings, Tools, and Help. The title bar displays the path /home/thuhua/Desktop/TH\_KTMT/W1/Exercise1.asm - MARS 4.5. The main window is divided into several panes:

- Edit | Execute**: A toolbar with various icons for assembly editing.
- Text Segment**: Shows assembly code with two entries:
  - Line 3: addi \$0,\$zero,0x3007 ; addi \$0,\$0,0x3007 # \$0 = 0 + 0x3007 :I-type
  - Line 4: add \$0,\$zero,\$0 # \$0 = 0 + 0 = 0 ;R-type
- Data Segment**: Shows memory starting at address 0x10000000 with all values set to 0x00000000 across multiple bytes.
- Registers**: A table showing processor registers:
 

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t1	8	0x00000000
\$t2	9	0x00000000
\$t3	10	0x00000000
\$t4	11	0x00000000
\$t5	12	0x00000000
\$t6	13	0x00000000
\$t7	14	0x00000000
\$s0	15	0x00000000
\$s1	16	0x00000000
\$s2	17	0x00000000
\$s3	18	0x00000000
\$s4	19	0x00000000
\$s5	20	0x00000000
\$s6	21	0x00000000
\$s7	22	0x00000000
\$t8	23	0x00000000
\$t9	24	0x00000000
\$t0	25	0x00000000
\$k1	26	0x00000000
\$sp	27	0x00000000
\$fp	28	0x00000000
\$r1	29	0x7fffffefffc
\$r2	30	0x00000000
pc	31	0x00400000
hi		0x00000000
lo		0x00000000
- Run IO**: Buttons for running the program, including step, run, and stop.
- Mars Messages**: A pane showing log messages:
  - program is finished running (dropped off bottom) --
  - Reset: reset completed.
  - program is finished running (dropped off bottom) --
  - Reset: reset completed.
  - Reset: reset completed.
  - Reset: reset completed.

Step 2:

The screenshot shows the MARS 4.5 assembly editor interface. The top menu bar includes File, Edit, Run, Settings, Tools, Help, and a toolbar with various icons. The main window has tabs for Text Segment and Data Segment.

**Text Segment:**

Blkpt	Address	Code	Basic	Source
	0x00400000	0x20103007	addi \$16,\$0,0x00000307	3: addi \$16,\$0,0x00000307 # \$16 = 0 + 0x3007 ;I-type
	0x00400004	0x00000002	add \$16,\$0,\$0	4: add \$16,\$0,\$0 # \$16 = 0 + 0 = 0 ;R-type

**Data Segment:**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

**Registers:**

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$t10	26	0x00000000
\$t11	27	0x00000000
\$sp	28	0x10000000
\$fp	29	0x7ffffeff
\$ra	30	0x00000000
pc	31	0x00400004
hi		0x00000000
lo		0x00000000

- *Sự thay đổi giá trị của các thanh ghi:*

+ Sự thay đổi giá trị của thanh ghi \$s0 khi chạy từng lệnh :

0x00000000 → 0x00003007 → 0x00000000

+ Sự thay đổi giá trị của thanh ghi \$pc khi chạy từng lệnh:

0x00400000 → 0x00400004 → 0x00400008

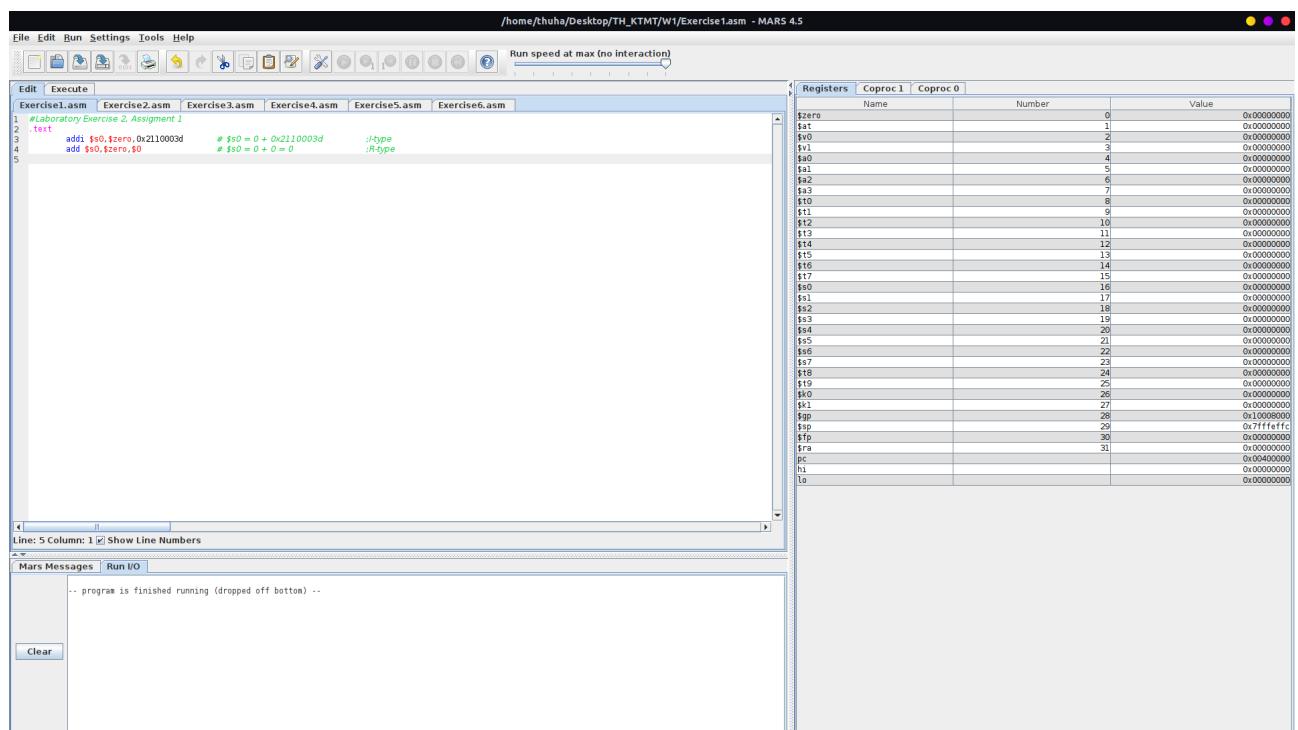
- So sánh mã máy của các lệnh:

addi \$s0, \$zero, 0x3007

rs: \$zero  
 rt: \$s0  
 imm: 0x3007  
 0010 0000 0001 0000 0011 0000 0000 0111 => 0x20103007

add \$s0, \$zero, \$0  
 op:0  
 rd: \$s0  
 rs: \$zero  
 rt: \$0  
 sham: 0  
 funct:32  
 0000 0000 0000 0000 1000 0000 0010 0000 => 0x00008020

- Sửa lại lệnh:



The screenshot shows the MARS 4.5 assembly debugger interface. The assembly code in the editor window is:

```

1 #LABORATORY Exercise 2, Assignment 1
2 .text
3 addi $s0,$zero,0x2110003d      # $s0 = 0 + 0x2110003d :I-type
4 addi $s0,$zero,$0                # $s0 = 0 + 0 = 0 :R-type
5

```

The Registers window on the right displays the following register values:

Registers	Name	Number	Value
	\$zero	0	0x00000000
	\$at	1	0x00000000
	\$v0	2	0x00000000
	\$v1	3	0x00000000
	\$a0	4	0x00000000
	\$a1	5	0x00000000
	\$a2	6	0x00000000
	\$a3	7	0x00000000
	\$t0	8	0x00000000
	\$t1	9	0x00000000
	\$t2	10	0x00000000
	\$t3	11	0x00000000
	\$t4	12	0x00000000
	\$t5	13	0x00000000
	\$t6	14	0x00000000
	\$t7	15	0x00000000
	\$s0	16	0x00000000
	\$s1	17	0x00000000
	\$s2	18	0x00000000
	\$s3	19	0x00000000
	\$s4	20	0x00000000
	\$s5	21	0x00000000
	\$s6	22	0x00000000
	\$s7	23	0x00000000
	\$t8	24	0x00000000
	\$t9	25	0x00000000
	\$t10	26	0x00000000
	\$t11	27	0x00000000
	\$gp	28	0x10000000
	\$sp	29	0x7fffffc
	\$tp	30	0x00000000
	\$ra	31	0x00000000
	pc		0x00400000
	hi		0x00000000
	lo		0x00000000

The Mars Messages window shows the message: ".. program is finished running (dropped off bottom) ..".

The screenshot shows the MARS 4.5 assembly editor interface. The top bar displays the file path: /home/thuhua/Desktop/TH\_KMT/WL/Exercise1.asm - MARS 4.5. The menu bar includes File, Edit, Run, Settings, Tools, and Help. The toolbar contains various icons for assembly operations. The main window is divided into several panes:

- Text Segment:** Shows assembly code with comments and assembly instructions. One instruction is highlighted in yellow: `addi $0,$zero,0x2110003d` at address 0x00000000.
- Data Segment:** Displays memory dump information for the data segment.
- Registers:** Shows the state of the processor registers (\$r0 to \$r31).
- Run IO:** A pane for monitoring I/O operations.
- Mars Messages:** A log of assembly and run-time messages.

At the bottom, there are buttons for step operations (Step Into, Step Over, Run, Stop), address selection (0x10010000 (.data)), and checkboxes for Hexadecimal Addresses, Hexadecimal Values, and ASCII.

## Step 1:

The screenshot shows the MARS 4.5 assembly editor interface. The top bar displays the file path /home/thuhu/Desktop/TH\_KTMT/W1/Exercise1.asm - MARS 4.5. The menu bar includes File, Edit, Run, Settings, Tools, Help. The toolbar contains various icons for file operations and assembly editing.

**Registers** pane (top right): Shows the processor state with all registers set to zero (0x00000000).

**Text Segment** pane (left): Displays assembly code in Basic format. The code includes instructions like ADDI and ADD, with comments indicating register usage.

**Data Segment** pane (middle left): Shows the memory dump starting at address 0x10010000, with values for each byte, word, doubleword, and quadword.

**Mars Messages** pane (bottom left): Logs messages such as 'Reset: reset completed.' and 'program is finished running (dropped off bottom)'.

**Run I/O** pane (bottom center): Contains buttons for running the program, selecting memory ranges (0x10010000 .data), and checkboxes for Hexadecimal Addresses, Hexadecimal Values, and ASCII.

---

## Step 2:

Step 3:

Name	Number	Value
\$zero	0	0x00000000
\$t0	1	0x2110003d
\$v0	2	0x00000000
\$a0	3	0x00000000
\$a1	4	0x00000000
\$a2	5	0x00000000
\$a3	6	0x00000000
\$t1	7	0x00000000
\$t2	8	0x00000000
\$t3	9	0x00000000
\$t4	10	0x00000000
\$t5	11	0x00000000
\$t6	12	0x00000000
\$t7	13	0x00000000
\$t8	14	0x00000000
\$t9	15	0x00000000
\$t10	16	0x00000000
\$t11	17	0x00000000
\$t12	18	0x00000000
\$t13	19	0x00000000
\$t14	20	0x00000000
\$t15	21	0x00000000
\$t16	22	0x00000000
\$t17	23	0x00000000
\$t18	24	0x00000000
\$t19	25	0x00000000
\$t20	26	0x00000000
\$t21	27	0x00000000
\$sp	28	0x01000000
\$fp	29	0x7fffffc1
\$ra	30	0x00000000
pc	31	0x00000000
hi		0x00000000
lo		0x00000000

Step 4:

Name	Number	Value
\$zero	0	0x00000000
\$t0	1	0x2110003d
\$v0	2	0x00000000
\$a0	3	0x00000000
\$a1	4	0x00000000
\$a2	5	0x00000000
\$a3	6	0x00000000
\$t1	7	0x00000000
\$t2	8	0x00000000
\$t3	9	0x00000000
\$t4	10	0x00000000
\$t5	11	0x00000000
\$t6	12	0x00000000
\$t7	13	0x00000000
\$t8	14	0x00000000
\$t9	15	0x00000000
\$t10	16	0x00000000
\$t11	17	0x00000000
\$t12	18	0x00000000
\$t13	19	0x00000000
\$t14	20	0x00000000
\$t15	21	0x00000000
\$t16	22	0x00000000
\$t17	23	0x00000000
\$t18	24	0x00000000
\$t19	25	0x00000000
\$t20	26	0x00000000
\$t21	27	0x00000000
\$sp	28	0x01000000
\$fp	29	0x7fffffc1
\$ra	30	0x00000000
pc	31	0x00000000
hi		0x00000000
lo		0x00000000

The screenshot shows the MARS 4.5 assembly debugger interface. The assembly pane displays the following code:

```

/home/thuha/Desktop/TH_KTMT/W1/Exercise1.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Bkpt Address Code Basic Source
0x00400000 0x3c012110 lui $1,0x00002110 3: addi $0,$zero,0x2110003d ;i-type
0x00400004 0x3421003d ori $1,$1,0x0000003d
0x00400008 0x00018020 add $16,$0,$1 4: add $0,$zero,$0 # $0 = 0 + 0 = 0 ;R-type
0x0040000c 0x00000020 addi $16,$0,$0

```

The Registers pane shows the following register values:

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x2110003d
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x2110003d
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$t10	26	0x00000000
\$t11	27	0x00000000
\$sp	28	0x00000000
\$fp	29	0x7fffffc
\$ra	30	0x00000000
pc	31	0x0040000c
hi		0x00000000
lo		0x00000000

The Data Segment pane shows the memory dump from address 0x10010000 to 0x100101f0.

The Mars Messages pane shows the following log:

```

Mars Messages Run I/O
Reset: reset completed.
Reset: reset completed.

-- program is finished running (dropped off bottom) --
-- program is finished running (dropped off bottom) --
-- program is finished running (dropped off bottom) --

```

Lệnh addi: 0x2110003d được chia thành 0x00002110 và 0x0000003d (Vì hằng số ở đây là 32 bit, máy phải tách ra thành 2 lệnh lui và ori để lần lượt chạy).

## Ex2.

- Chương trình:

The screenshot shows the MARS 4.5 assembly debugger interface. The assembly pane displays the following code:

```

/home/thuha/Desktop/TH_KTMT/W1/Exercise2.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Exercise1.asm Exercise2.asm Exercise3.asm Exercise4.asm Exercise5.asm Exercise6.asm
1 #Laboratory Exercise 2, Assignment 2.
2 .text
3 lui $0,0x2110          #put upper half on of pattern in $0
4 ori $0,0x003d          #put lower half on of pattern in $0
5

```

The Registers pane shows the following register values:

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$t0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$t7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$t0	26	0x00000000
\$k1	27	0x00000000
\$sp	28	0x00000000
\$fp	29	0x7fffffc
\$ra	30	0x00000000
pc		0x0040000c
hi		0x00000000
lo		0x00000000

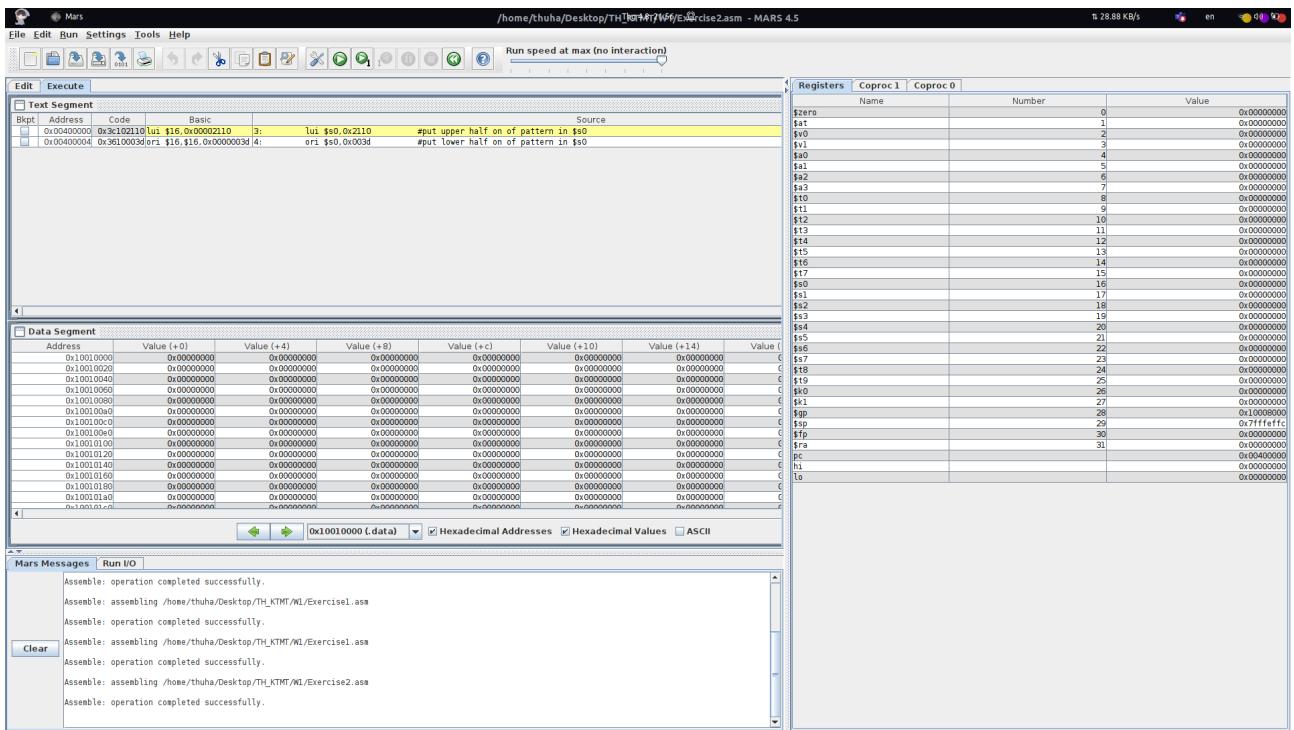
The Mars Messages pane shows the following log:

```

Mars Messages Run I/O
Step: execution terminated due to null instruction.
Assemble: assembling /home/thuha/Desktop/TH_KTMT/W1/Exercise1.asm
Assemble: operation completed successfully.

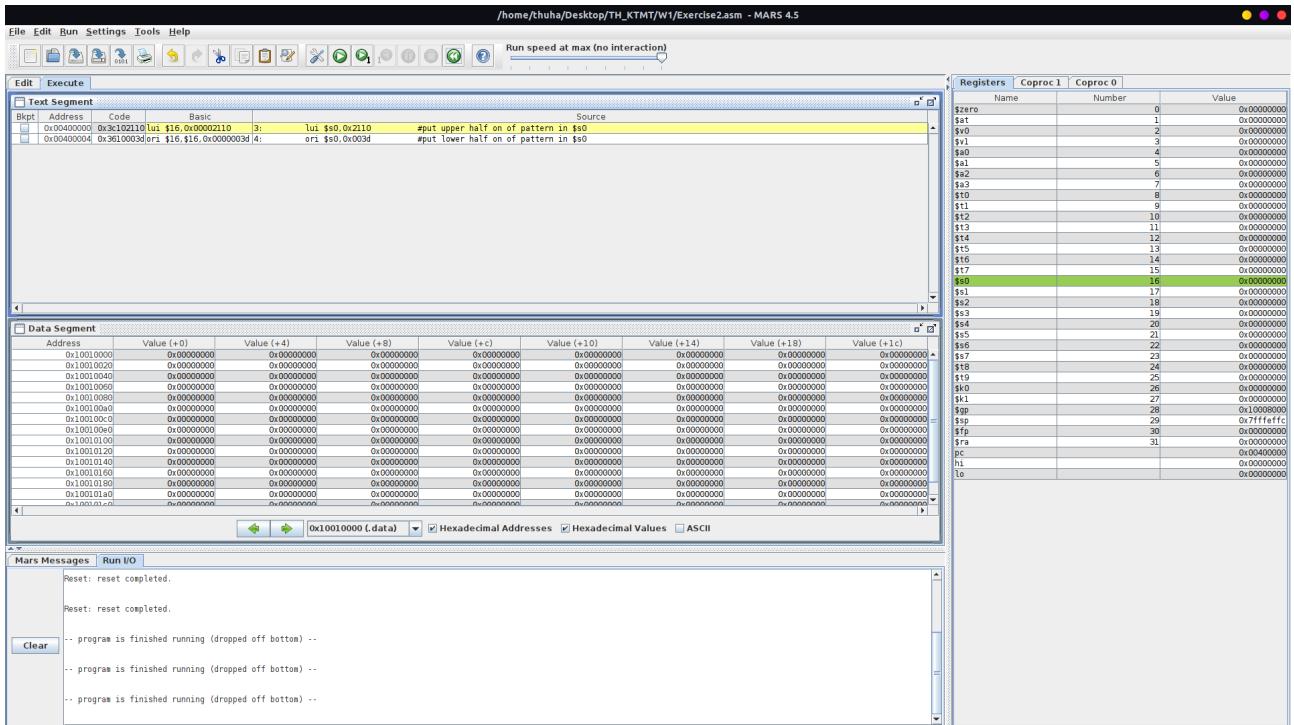
Clear Assemble: assembling /home/thuha/Desktop/TH_KTMT/W1/Exercise1.asm
Assemble: operation completed successfully.
Assemble: assembling /home/thuha/Desktop/TH_KTMT/W1/Exercise1.asm
Assemble: operation completed successfully.

```



- Chạy từng dòng lệnh:

## Step 1:



## Step 2:

The figure shows the MARS 4.5 assembly editor interface. The top status bar displays the file path: /home/tuhua/Desktop/TH\_KTMT/W1/Exercise2.asm - MARS 4.5. The menu bar includes File, Edit, Run, Settings, Tools, and Help. The toolbar contains various icons for file operations and assembly editing. The main window is divided into several panes:

- Text Segment:** Shows assembly code with comments. Two lines are highlighted in yellow:
  - lui \$16,0x00002110 #put upper half on of pattern in \$16
  - ori \$16,\$16,0x0000003d #put lower half on of pattern in \$16
- Data Segment:** Displays memory starting at address 0x10010000. The first 16 bytes are shown as 0x00000000. A dropdown menu below the table allows switching between Hexadecimal Addresses, Hexadecimal Values, and ASCII.
- Registers:** A table showing the state of registers. The \$16 register is highlighted in green, indicating it is the current register being edited.

- Sự thay đổi của giá trị của các thanh ghi:

+ Sự thay đổi giá trị của thanh ghi \$s0 khi chạy từng lệnh :

`0x00000000 => 0x21100000 => 0x2110003d`

+ Sự thay đổi giá trị của thanh ghi \$pc khi chạy từng lệnh :

`0x00400000 => 0x00400004 => 0x00400008`

+ Ở cửa sổ Data Segment của byte đầu tiên trùng với cột code trong Text Segment.

Data Segment									
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)	Value (+10)
0x00400000	0x3c102110	0x3610003d	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Text Segment									
Bkpt	Address	Code	Basic	Source					
	0x00400000	0x3c102110	lui \$16,0x00002110	3:	lui \$s0,0x2110	#put upper half on of pattern in \$s0			
	0x00400004	0x3610003d	ori \$16,\$16,0x0000003d	4:	ori \$s0,0x003d	#put lower half on of pattern in \$s0			

### Ex3.

- Chương trình:

Mars - /home/thuha/Desktop/TH\_KTMT/W1/Exercise3.asm - MARS 4.5

File Edit Run Settings Tools Help Run speed at max (no interaction)

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0x00000000
\$t1	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$t8	16	0x00000000
\$t9	17	0x00000000
\$t0	18	0x00000000
\$t1	19	0x00000000
\$t2	20	0x00000000
\$t3	21	0x00000000
\$t4	22	0x00000000
\$t5	23	0x00000000
\$t6	24	0x00000000
\$t7	25	0x00000000
\$t8	26	0x00000000
\$t9	27	0x00000000
\$gp	28	0x10000000
\$sp	29	0x7fffffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00000000
hi		0x00000000
lo		0x00000000

Line: 1 Column: 1 Show Line Numbers

Mars Messages Run I/O

```

Assemble: operation completed successfully.
Assemble: assembling /home/thuha/Desktop/TH_KTMT/W1/Exercise1.asm
Assemble: operation completed successfully.
Assemble: assembling /home/thuha/Desktop/TH_KTMT/W1/Exercise1.asm
Assemble: operation completed successfully.
Assemble: assembling /home/thuha/Desktop/TH_KTMT/W1/Exercise2.asm
Assemble: operation completed successfully.

```

/home/thuha/Desktop/TH\_KTMT/W1/Exercise3.asm - MARS 4.5

File Edit Run Settings Tools Help Run speed at max (no interaction)

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0x00000000
\$t1	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$t8	16	0x00000000
\$t9	17	0x00000000
\$t0	18	0x00000000
\$t1	19	0x00000000
\$t2	20	0x00000000
\$t3	21	0x00000000
\$t4	22	0x00000000
\$t5	23	0x00000000
\$t6	24	0x00000000
\$t7	25	0x00000000
\$t8	26	0x00000000
\$t9	27	0x00000000
\$gp	28	0x10000000
\$sp	29	0x7fffffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00000000
hi		0x00000000
lo		0x00000000

Text Segment

Bkpt	Address	Code	Basic	Source
0x00400000	0x3c012110	lui \$1, 0x00002110	3:	li \$0, 0x2110003d #pseudo instruction =2 basic instructions
0x00400004	0x3430003d	ori \$16, \$1, 0x0000003d		
0x00400008	0x24110002	addiu \$17, \$0, 0x0000... 4:		li \$s1,0x2 #but if the immediate value is small, one instruction

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value
0x10000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100000a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100000c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100001a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100001c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Mars Messages Run I/O

```

Assemble: operation completed successfully.
Assemble: assembling /home/thuha/Desktop/TH_KTMT/W1/Exercise1.asm
Assemble: operation completed successfully.
Assemble: assembling /home/thuha/Desktop/TH_KTMT/W1/Exercise2.asm
Assemble: operation completed successfully.
Assemble: assembling /home/thuha/Desktop/TH_KTMT/W1/Exercise3.asm
Assemble: operation completed successfully.

```

Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source
0x00400000	0x3c012110	lui \$1, 0x00002110	3:	li \$0, 0x2110003d #pseudo instruction =2 basic instructions
0x00400004	0x3430003d	ori \$16, \$1, 0x0000003d		
0x00400008	0x24110002	addiu \$17, \$0, 0x0000... 4:		li \$s1,0x2 #but if the immediate value is small, one instruction

- Giải thích:

- + Lệnh li là giả lệnh, được phân tách thành 2 lệnh là lui và ori.
- + Vì giá trị 0x2110003d là hằng số có giá trị 32 bit nên phải tách thành 2 phần 16 bit và dùng lệnh lui và ori để lấy địa chỉ.
- + lui \$1, 0x00002110: thực hiện gán địa chỉ 0x00002110 vào thanh ghi \$1.
- + ori \$16, \$1, 0x0000003d: thực hiện gán địa chỉ 0x0000003d vào thanh ghi \$16.

+ Lệnh addiu \$17,\$0,0x00000002 gán giá trị vào \$17.

Ex4:

- Chương trình:

```

1 #LABORATORY Exercise 2, Assignment 4
2 .text
3     #Assign X, Y
4     addi    $11,$zero,5      # X = $11 =?
5     addi    $12,$zero,-1     # Y = $12 =?
6     # Expression Z = 2X + Y
7     add    $10,$11,$11       # $10 = $11+$11 = X + X = 2X
8     add    $10,$10,$12       # $10 = $10 + $12 = 2X + Y
9

```

Registers	Coproc 1	Coproc 0
\$zero		0
\$t0		1
\$v0		2
\$v1		3
\$a0		4
\$a1		5
\$a2		6
\$a3		7
\$t0		8
\$t1		9
\$t2		10
\$t3		11
\$t4		12
\$t5		13
\$t6		14
\$t7		15
\$t8		16
\$t9		17
\$k0		18
\$k1		19
\$fp		20
\$sp		21
\$ra		22
pc		23
hi		24
lo		25

Mars Messages

```

Assemble: operation completed successfully.
Assemble: assembling /home/thuha/Desktop/TH_KTMT/W1/Exercise1.asm
Assemble: operation completed successfully.
Assemble: assembling /home/thuha/Desktop/TH_KTMT/W1/Exercise2.asm
Clear
Assemble: operation completed successfully.
Assemble: assembling /home/thuha/Desktop/TH_KTMT/W1/Exercise3.asm
Assemble: operation completed successfully.

```

```

1 #LABORATORY Exercise 2, Assignment 4
2 .text
3     #Assign X, Y
4     addi    $11,$zero,5      # X = $11 =?
5     addi    $12,$zero,-1     # Y = $12 =?
6     # Expression Z = 2X + Y
7     add    $10,$11,$11       # $10 = $11+$11 = X + X = 2X
8     add    $10,$10,$12       # $10 = $10 + $12 = 2X + Y
9

```

Registers	Coproc 1	Coproc 0
\$zero		0
\$t0		1
\$v0		2
\$v1		3
\$a0		4
\$a1		5
\$a2		6
\$a3		7
\$t0		8
\$t1		9
\$t2		10
\$t3		11
\$t4		12
\$t5		13
\$t6		14
\$t7		15
\$t8		16
\$t9		17
\$k1		18
\$k2		19
\$k3		20
\$k4		21
\$k5		22
\$k6		23
\$k7		24
\$k8		25
\$k9		26
\$k10		27
\$fp		28
\$sp		29
\$ra		30
pc		31
hi		0x00000000
lo		0x00000000

Mars Messages

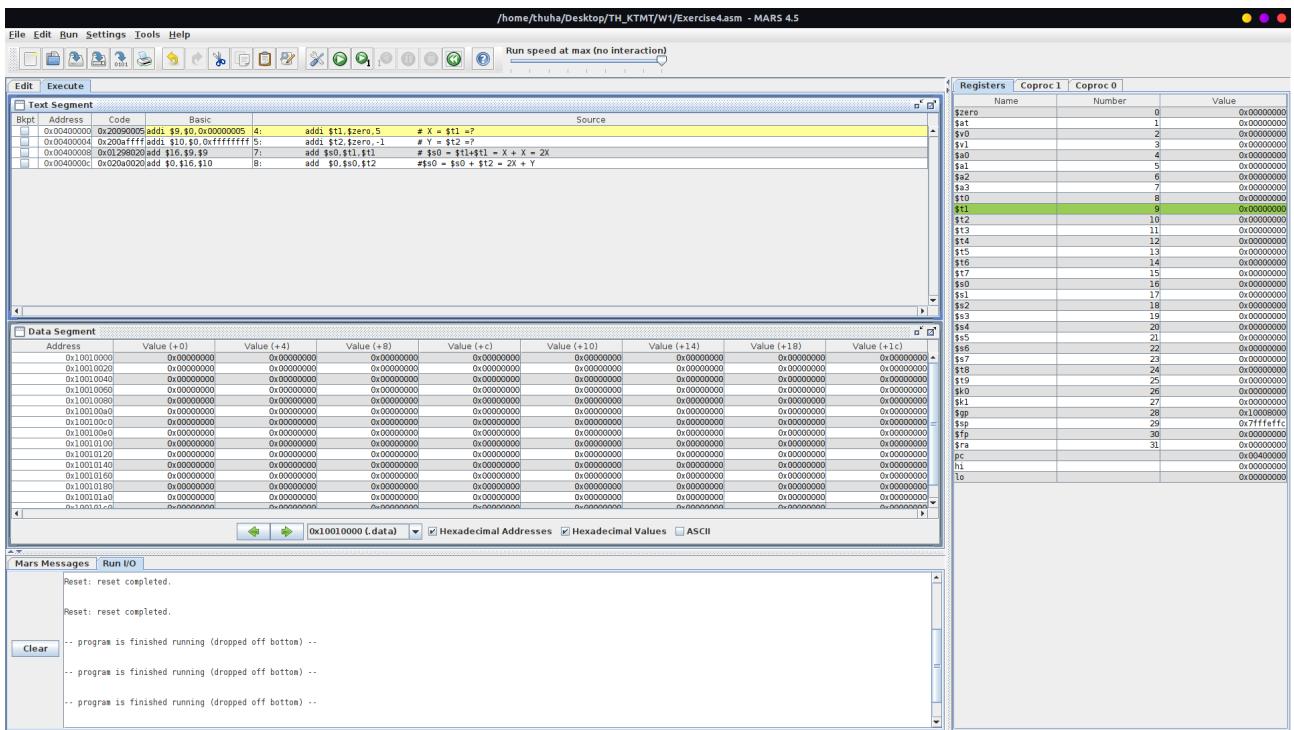
```

Assemble: operation completed successfully.
Assemble: assembling /home/thuha/Desktop/TH_KTMT/W1/Exercise2.asm
Assemble: operation completed successfully.
Assemble: assembling /home/thuha/Desktop/TH_KTMT/W1/Exercise3.asm
Clear
Assemble: operation completed successfully.
Assemble: assembling /home/thuha/Desktop/TH_KTMT/W1/Exercise4.asm
Assemble: operation completed successfully.

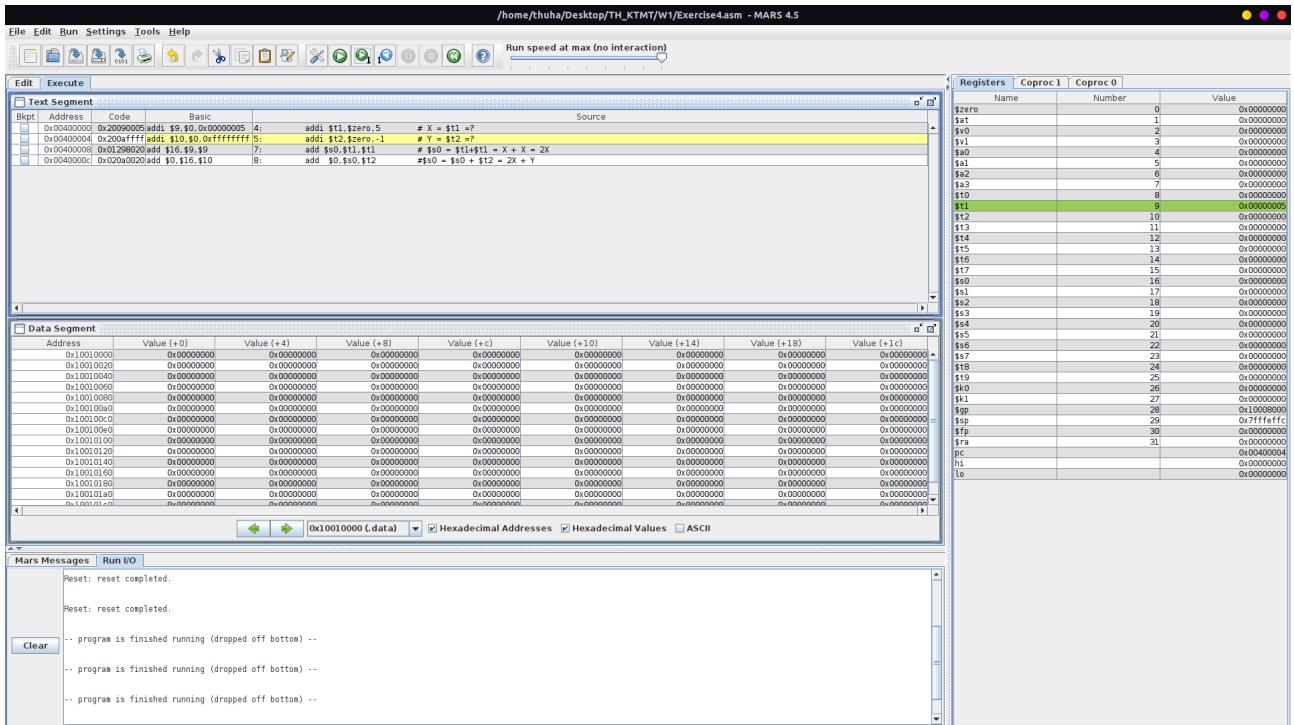
```

- Chạy từng dòng lệnh:

Step 1:



## Step 2:



### Step 3:

The figure shows the MARS 4.5 assembly editor interface. The top menu bar includes File, Edit, Run, Settings, Tools, Help, and a toolbar with various icons. A status bar at the bottom indicates the current memory location (0x10010000), address format (L(data)), and data type (Hexadecimal Addresses). The main window is divided into several panes:

- Text Segment:** Displays assembly code in columns for Blkpt, Address, Code, and Basic. The code includes add instructions for registers \$t1, \$t2, and \$t3.
- Data Segment:** Displays memory starting at address 0x10010000 with values for offsets 0, 4, 8, etc., all set to 0x00000000.
- Registers:** A table showing the state of various registers (\$zero through \$t3, \$t10, \$t11, \$t12, \$t13, \$t14, \$t15, \$t16, \$t17, \$t18, \$t19, \$t20, \$t21, \$t22, \$t23, \$t24, \$t25, \$t26, \$t27, \$t28, \$t29, \$t30, \$t31) and special purpose registers (\$sp, \$gp, \$fp, \$ra, \$pc, \$hi, \$lo).
- Mars Messages:** A log window showing messages such as "Reset: reset completed." and "program is finished running (dropped off bottom)".

## Step 4:

The screenshot shows the MARS 4.5 assembly debugger interface with the following panes:

- Text Segment:** Displays assembly code with addresses, opcodes, and comments. The current instruction is highlighted at address 0x00000004.
- Data Segment:** Displays memory starting at address 0x10000000 with values for offsets +0, +4, +8, +c, +10, +14, +18, and +1c.
- Registers:** Shows the CPU registers (r0 to r31, pc, sp, fp) and their current values.
- Mars Messages:** Displays log messages from the debugger, including "Reset: reset completed." and "program is finished running (dropped off bottom)" repeated multiple times.

Kết quả bằng 9 => chạy đúng kết quả.

- Sự thay đổi của các thanh ghi:

\$t1: 0x00000000 => 0x00000005

\$t2: 0x00000000 => 0xffffffff

`$s0: 0x00000000 => 0x0000000a => 0x00000009`

- Kiểm nghiệm với khuôn mẫu của kiểu lệnh I:

addi \$9, \$0, 0x00000005  
op: 8  
rs: \$0  
rt: \$9  
imm: 0x00000005  
0010 0000 0000 1001 0000 0000 0000 0101 => 0x20090005

addi \$10, \$0, 0xffffffff  
op: 8  
rs: \$0  
rt: \$10  
imm: 0xffffffff  
0010 0000 0000 1010 1111 1111 1111 1111 => 0x200affff

- Kiểm nghiệm với khuôn mẫu của kiểu lệnh R:

add \$16, \$9, \$9  
op: 0  
rs: \$9  
rt: \$9  
rd: \$16  
sh: 0  
fn: 32  
0000 0001 0010 1001 1000 0000 0010 0000 => 0x01298020

add \$16, \$16, \$10  
op: 0  
rs: \$16  
rt: \$10  
rd: \$16  
sh: 0  
fn: 32  
0000 0010 0000 1010 1000 0000 0010 0000 => 0x020a8020

**Ex5:**

- Chương trình:

The screenshot shows the MARS 4.5 assembly editor interface. The assembly code for 'Exercise2.asm' is displayed in the main window:

```
1 #Laboratory Exercise 2, Assignment 5
2 .text
3     # Assign XY
4     addi $11,$zero,4      #X=$11=?
5     addi $12,$zero,5      #Y = $12=?
6     subi $11,$12,0        #H=XY
7     mul $00,$11,$12       #H:LO = $11 * $12 = XY; $S0=LO
8     mul $00,$10,3         #$S0= $S0*3^X*Y
9     xzr                  #Z = J
10    mflo $11
11
```

The Registers window shows the state of the processor registers:

Registers	Coproc 1	Coproc 0
\$zero		0
\$t1		1
\$v0		2
\$v1		3
\$a0		4
\$a1		5
\$a2		6
\$a3		7
\$t0		8
\$t1		9
\$t2		10
\$t3		11
\$t4		12
\$t5		13
\$t6		14
\$t7		15
\$s0		16
\$s1		17
\$s2		18
\$s3		19
\$s4		20
\$s5		21
\$s6		22
\$s7		23
\$t8		24
\$t9		25
\$a0		26
\$k1		27
\$gp		28
\$sp		29
\$fp		30
ra		31
pc		0x04000000
hi		0x00000000
lo		0x00000000

The Mars Messages window shows the assembly process:

- Assemble: operation completed successfully.
- Assemble: assembling /home/thuha/Desktop/TH\_KMT/W1/Exercise2.asm
- Assemble: operation completed successfully.
- Assemble: assembling /home/thuha/Desktop/TH\_KMT/W1/Exercise3.asm
- Assemble: operation completed successfully.
- Assemble: assembling /home/thuha/Desktop/TH\_KMT/W1/Exercise4.asm
- Assemble: operation completed successfully.

The screenshot shows the MARS 4.5 assembly editor interface. The assembly window displays the following code:

```

Text Segment
Bkpt Address Code Basic Source
0x00400000 0x20090004 addi $9, $0, 0x00000004 4: addi $11,$zero,4 #X=$11=?
0x00400004 0x200a0005 addi $10,$0,0x00000005 5: addi $12,$zero,5 #Y=$12=?
0x00400008 0x200b0006 mul $10,$11,$12 6: mul $10,$11,$12 #H=I*D = $11 * $12 = X*Y; $0=4.0
0x0040000c 0x200c0003 addi $1,$0,0x00000003 7: addi $13,$0,0.00000003 #I=0=4.0*D*X*Y
0x00400010 0x20180002 mul $16,$16,$1 8: mul $16,$16,$1
0x00400014 0x400008812 mflo $17 10: mflo $1

```

The Registers window shows the following register values:

Registers	Coproc 1	Coproc 0
\$zero		0
\$t1		1
\$v0		2
\$v1		3
\$a0		4
\$a1		5
\$a2		6
\$a3		7
\$t0		8
\$t1		9
\$t2		10
\$t3		11
\$t4		12
\$t5		13
\$t6		14
\$t7		15
\$s0		16
\$s1		17
\$s2		18
\$s3		19
\$s4		20
\$s5		21
\$t6		22
\$t7		23
\$t8		24
\$t9		25
\$t10		26
\$t11		27
\$sp		28
\$fp		29
\$ra		30
pc		31
hi		0x00400000
lo		0x00000000

The Data Segment window shows the memory dump:

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

The Mars Messages window shows the assembly log:

```

Assemble: operation completed successfully.
Assemble: assembling /home/thuha/Desktop/TH_KMT/W1/Exercise3.asm
Assemble: operation completed successfully.
Assemble: assembling /home/thuha/Desktop/TH_KMT/W1/Exercise4.asm
Assemble: operation completed successfully.
Assemble: assembling /home/thuha/Desktop/TH_KMT/W1/Exercise5.asm
Assemble: operation completed successfully.

```

- Giải thích điều bất thường:

- + Lệnh mul đầu tiên thực hiện bình thường do là lệnh mul basic ( nhân hai biến).
  - + Lệnh mul thứ 2 không phải là lệnh basic ( vì nhân hằng với biến), nên thực hiện biến đổi thành hai lệnh ( gán hằng số cho thanh ghi tạm rồi mới thực hiện phép nhân).

- Chạy từng dòng lệnh:

## Step 1:

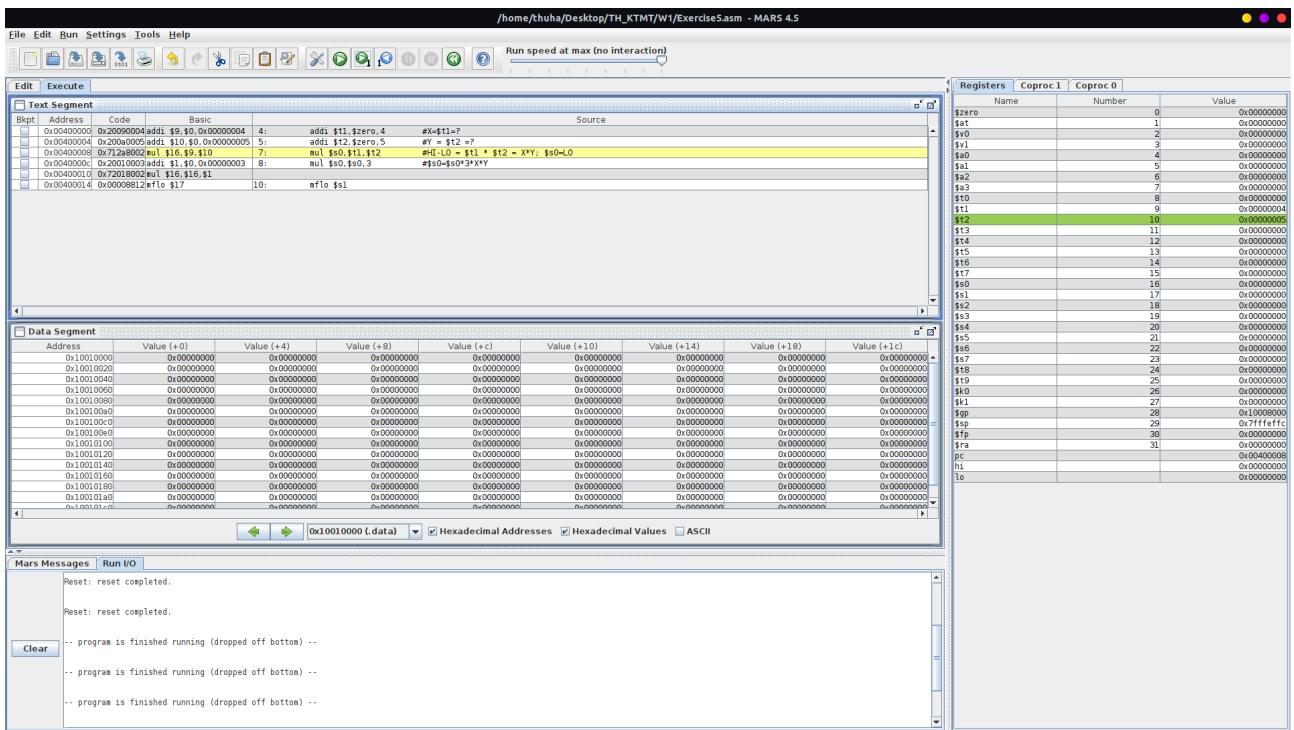
The screenshot shows the MARS 4.5 assembly debugger interface with several windows open:

- Registers** pane: Shows the CPU registers (\$r0 to \$r15, \$t0 to \$t15, \$s0 to \$s15) and floating-point registers (\$f0 to \$f15). The \$t1 register is highlighted.
- Stack** pane: Shows the stack contents from address 0x10010000 to 0x10010010, mostly filled with zeros.
- Mars Messages** pane: Displays log messages indicating a successful reset and program completion.

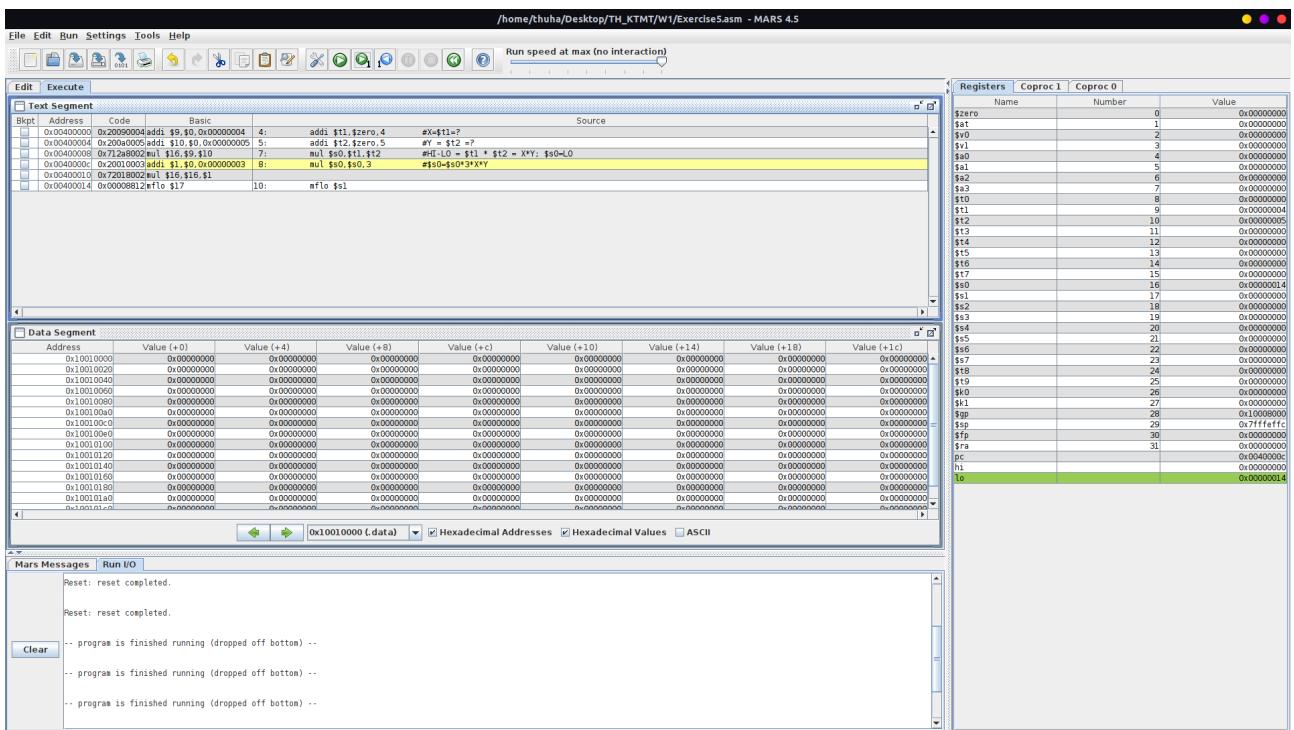
## Step 2:

The screenshot shows the MARS 4.5 assembly debugger interface. The Registers window on the right lists CPU registers from \$zero to \$t1L, with \$t1L highlighted in green. The Stack window below it shows memory starting at address 0x10000000, with values for \$t1L through \$t1H. The Registers window also shows the stack pointer (\$sp) at 0x00000000. The Registers window has tabs for Coproc 1 and Coproc 0, which are currently inactive.

### Step 3:



## Step 4:



## Step 5:

The screenshot shows the MARS 4.5 assembly editor interface. The top menu bar includes File, Edit, Run, Settings, Tools, Help, and a Run speed dropdown set to "Run speed at max (no interaction)".

**Edit | Execute**

**Text Segment**

Bkpt	Address	Code	Basic	Source
	0x00400000	0x20090004	addi \$9,\$0,0x00000004	4: addi \$11,\$zero,4 #X=11=?
	0x00400004	0x20090005	addi \$12,\$zero,5	5: addi \$12,\$zero,5 #Y=12=?
	0x00400008	0x20090006	mult \$9,\$11,\$12	6: mult \$9,\$11,\$12 #H=99 = \$11 * \$12 = X*Y: \$0=40
	0x0040000C	0x20090003	addi \$1,\$0,0x00000003	7: addi \$1,\$0,0x00000003 #I=0=40*2*X*Y
	0x00400010	0x7e020802	mflo \$17	10: mflo \$1
	0x00400014	0x00008812	mflr \$1	

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100A0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100C0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100E0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101A0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101C0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

**Mars Messages | Run I/O**

```

Reset: reset completed.

Reset: reset completed.

-- program is finished running (dropped off bottom) --

-- program is finished running (dropped off bottom) --

-- program is finished running (dropped off bottom) --

```

## Step 6:

The screenshot shows the MARS 4.5 assembly editor interface. The top menu bar includes File, Edit, Run, Settings, Tools, Help, and a toolbar with various icons. The main window has tabs for Text Segment, Data Segment, and Registers.

**Text Segment:** Displays assembly code with comments. The code includes additions, multiplications, and a division operation.

```

    .text
    .globl _start
_start:
    addi $t1,$zero,4      #t0=$1??
    addi $t2,$zero,5      #Y=t2=?
    mul $t0,$t1,$t2      #H-L=t1*t2=X/Y: $t0=d0.0
    addi $t0,$zero,3      #t0=$0.3
    mul $t0,$t0,$t0      #t0=$0*t0^3*X/Y
    mflo $t1

```

**Data Segment:** Displays memory starting at address 0x10010000 with values for offsets +0, +4, +8, +c, +10, +14, +18, and +1c.

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010004	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010008	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001000C	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010010	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010014	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010018	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001001C	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010024	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010028	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001002C	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010030	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010034	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010038	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001003C	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010044	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010048	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001004C	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010050	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010054	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010058	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001005C	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010064	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010068	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010070	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010074	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010078	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001007C	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010084	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010088	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001008C	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010090	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010094	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010098	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001009C	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100A0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100A4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100A8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100B0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100B4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100B8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100BC	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100C0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100C4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100C8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100D0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100D4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100D8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100E0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100E4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100E8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100F0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100F4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100F8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100FC	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

**Registers:**

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000003
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000004
\$t2	10	0x00000005
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$t8	16	0x00000000
\$t9	17	0x00000000
\$s0	18	0x00000000
\$s1	19	0x00000000
\$s2	20	0x00000000
\$s3	21	0x00000000
\$s4	22	0x00000000
\$s5	23	0x00000000
\$s6	24	0x00000000
\$s7	25	0x00000000
\$s8	26	0x00000000
\$s9	27	0x00000000
\$sp	28	0x10010000
\$gp	29	0x00000000
\$tp	30	0x00000000
\$ra	31	0x00000000
pc	32	0x00000004
hi	33	0x00000000
lo	34	0x00000000

- Sự thay đổi của các thanh ghi:

\$t1 : 0x00000000 => 0x00000004

\$t2: 0x00000000 => 0x00000005

\$lo: 0x00000000 => 0x00000014 => 0x0000003c

\$at: 0x00000000 => 0x00000003

\$s0: 0x00000000 => 0x0000003c

\$s1: 0x00000000 => 0x0000003c

Tham mưu không thay đổi giá trị VI kết quả dưới 32 bit, kết quả ở đây được gom vào tham mưu gom kít.

## Ex6:

- Chương trình:

<img alt="Screenshot of MARS 4.5 assembly editor showing Exercise6.asm code and registers. The code calculates Z = 2X + Y. Registers show \$zero=0, \$t0=2, \$t1=3, \$t2=4, \$t3=5, \$t4=6, \$t5=7, \$t6=8, \$t7=9, \$t8=10, \$t9=11, \$t10=12, \$t11=13, \$t12=14, \$t13=15, \$t14=16, \$t15=17, \$t16=18, \$t17=19, \$t18=20, \$t19=21, \$t20=22, \$t21=23, \$t22=24, \$t23=25, \$t24=26, \$t25=27, \$t26=28, \$t27=29, \$t28=30, \$t29=31, \$t30=32, \$t31=33, \$t32=34, \$t33=35, \$t34=36, \$t35=37, \$t36=38, \$t37=39, \$t38=40, \$t39=41, \$t40=42, \$t41=43, \$t42=44, \$t43=45, \$t44=46, \$t45=47, \$t46=48, \$t47=49, \$t48=50, \$t49=51, \$t50=52, \$t51=53, \$t52=54, \$t53=55, \$t54=56, \$t55=57, \$t56=58, \$t57=59, \$t58=60, \$t59=61, \$t60=62, \$t61=63, \$t62=64, \$t63=65, \$t64=66, \$t65=67, \$t66=68, \$t67=69, \$t68=70, \$t69=71, \$t70=72, \$t71=73, \$t72=74, \$t73=75, \$t74=76, \$t75=77, \$t76=78, \$t77=79, \$t78=80, \$t79=81, \$t80=82, \$t81=83, \$t82=84, \$t83=85, \$t84=86, \$t85=87, \$t86=88, \$t87=89, \$t88=90, \$t89=91, \$t90=92, \$t91=93, \$t92=94, \$t93=95, \$t94=96, \$t95=97, \$t96=98, \$t97=99, \$t98=100, \$t99=101, \$t100=102, \$t101=103, \$t102=104, \$t103=105, \$t104=106, \$t105=107, \$t106=108, \$t107=109, \$t108=110, \$t109=111, \$t110=112, \$t111=113, \$t112=114, \$t113=115, \$t114=116, \$t115=117, \$t116=118, \$t117=119, \$t118=120, \$t119=121, \$t120=122, \$t121=123, \$t122=124, \$t123=125, \$t124=126, \$t125=127, \$t126=128, \$t127=129, \$t128=130, \$t129=131, \$t130=132, \$t131=133, \$t132=134, \$t133=135, \$t134=136, \$t135=137, \$t136=138, \$t137=139, \$t138=140, \$t139=141, \$t140=142, \$t141=143, \$t142=144, \$t143=145, \$t144=146, \$t145=147, \$t146=148, \$t147=149, \$t148=150, \$t149=151, \$t150=152, \$t151=153, \$t152=154, \$t153=155, \$t154=156, \$t155=157, \$t156=158, \$t157=159, \$t158=160, \$t159=161, \$t160=162, \$t161=163, \$t162=164, \$t163=165, \$t164=166, \$t165=167, \$t166=168, \$t167=169, \$t168=170, \$t169=171, \$t170=172, \$t171=173, \$t172=174, \$t173=175, \$t174=176, \$t175=177, \$t176=178, \$t177=179, \$t178=180, \$t179=181, \$t180=182, \$t181=183, \$t182=184, \$t183=185, \$t184=186, \$t185=187, \$t186=188, \$t187=189, \$t188=190, \$t189=191, \$t190=192, \$t191=193, \$t192=194, \$t193=195, \$t194=196, \$t195=197, \$t196=198, \$t197=199, \$t198=200, \$t199=201, \$t200=202, \$t201=203, \$t202=204, \$t203=205, \$t204=206, \$t205=207, \$t206=208, \$t207=209, \$t208=210, \$t209=211, \$t210=212, \$t211=213, \$t212=214, \$t213=215, \$t214=216, \$t215=217, \$t216=218, \$t217=219, \$t218=220, \$t219=221, \$t220=222, \$t221=223, \$t222=224, \$t223=225, \$t224=226, \$t225=227, \$t226=228, \$t227=229, \$t228=230, \$t229=231, \$t230=232, \$t231=233, \$t232=234, \$t233=235, \$t234=236, \$t235=237, \$t236=238, \$t237=239, \$t238=240, \$t239=241, \$t240=242, \$t241=243, \$t242=244, \$t243=245, \$t244=246, \$t245=247, \$t246=248, \$t247=249, \$t248=250, \$t249=251, \$t250=252, \$t251=253, \$t252=254, \$t253=255, \$t254=256, \$t255=257, \$t256=258, \$t257=259, \$t258=260, \$t259=261, \$t260=262, \$t261=263, \$t262=264, \$t263=265, \$t264=266, \$t265=267, \$t266=268, \$t267=269, \$t268=270, \$t269=271, \$t270=272, \$t271=273, \$t272=274, \$t273=275, \$t274=276, \$t275=277, \$t276=278, \$t277=279, \$t278=280, \$t279=281, \$t280=282, \$t281=283, \$t282=284, \$t283=285, \$t284=286, \$t285=287, \$t286=288, \$t287=289, \$t288=290, \$t289=291, \$t290=292, \$t291=293, \$t292=294, \$t293=295, \$t294=296, \$t295=297, \$t296=298, \$t297=299, \$t298=300, \$t299=301, \$t300=302, \$t301=303, \$t302=304, \$t303=305, \$t304=306, \$t305=307, \$t306=308, \$t307=309, \$t308=310, \$t309=311, \$t310=312, \$t311=313, \$t312=314, \$t313=315, \$t314=316, \$t315=317, \$t316=318, \$t317=319, \$t318=320, \$t319=321, \$t320=322, \$t321=323, \$t322=324, \$t323=325, \$t324=326, \$t325=327, \$t326=328, \$t327=329, \$t328=330, \$t329=331, \$t330=332, \$t331=333, \$t332=334, \$t333=335, \$t334=336, \$t335=337, \$t336=338, \$t337=339, \$t338=340, \$t339=341, \$t340=342, \$t341=343, \$t342=344, \$t343=345, \$t344=346, \$t345=347, \$t346=348, \$t347=349, \$t348=350, \$t349=351, \$t350=352, \$t351=353, \$t352=354, \$t353=355, \$t354=356, \$t355=357, \$t356=358, \$t357=359, \$t358=360, \$t359=361, \$t360=362, \$t361=363, \$t362=364, \$t363=365, \$t364=366, \$t365=367, \$t366=368, \$t367=369, \$t368=370, \$t369=371, \$t370=372, \$t371=373, \$t372=374, \$t373=375, \$t374=376, \$t375=377, \$t376=378, \$t377=379, \$t378=380, \$t379=381, \$t380=382, \$t381=383, \$t382=384, \$t383=385, \$t384=386, \$t385=387, \$t386=388, \$t387=389, \$t388=390, \$t389=391, \$t390=392, \$t391=393, \$t392=394, \$t393=395, \$t394=396, \$t395=397, \$t396=398, \$t397=399, \$t398=400, \$t399=401, \$t400=402, \$t401=403, \$t402=404, \$t403=405, \$t404=406, \$t405=407, \$t406=408, \$t407=409, \$t408=410, \$t409=411, \$t410=412, \$t411=413, \$t412=414, \$t413=415, \$t414=416, \$t415=417, \$t416=418, \$t417=419, \$t418=420, \$t419=421, \$t420=422, \$t421=423, \$t422=424, \$t423=425, \$t424=426, \$t425=427, \$t426=428, \$t427=429, \$t428=430, \$t429=431, \$t430=432, \$t431=433, \$t432=434, \$t433=435, \$t434=436, \$t435=437, \$t436=438, \$t437=439, \$t438=440, \$t439=441, \$t440=442, \$t441=443, \$t442=444, \$t443=445, \$t444=446, \$t445=447, \$t446=448, \$t447=449, \$t448=450, \$t449=451, \$t450=452, \$t451=453, \$t452=454, \$t453=455, \$t454=456, \$t455=457, \$t456=458, \$t457=459, \$t458=460, \$t459=461, \$t460=462, \$t461=463, \$t462=464, \$t463=465, \$t464=466, \$t465=467, \$t466=468, \$t467=469, \$t468=470, \$t469=471, \$t470=472, \$t471=473, \$t472=474, \$t473=475, \$t474=476, \$t475=477, \$t476=478, \$t477=479, \$t478=480, \$t479=481, \$t480=482, \$t481=483, \$t482=484, \$t483=485, \$t484=486, \$t485=487, \$t486=488, \$t487=489, \$t488=490, \$t489=491, \$t490=492, \$t491=493, \$t492=494, \$t493=495, \$t494=496, \$t495=497, \$t496=498, \$t497=499, \$t498=500, \$t499=501, \$t500=502, \$t501=503, \$t502=504, \$t503=505, \$t504=506, \$t505=507, \$t506=508, \$t507=509, \$t508=510, \$t509=511, \$t510=512, \$t511=513, \$t512=514, \$t513=515, \$t514=516, \$t515=517, \$t516=518, \$t517=519, \$t518=520, \$t519=521, \$t520=522, \$t521=523, \$t522=524, \$t523=525, \$t524=526, \$t525=527, \$t526=528, \$t527=529, \$t528=530, \$t529=531, \$t530=532, \$t531=533, \$t532=534, \$t533=535, \$t534=536, \$t535=537, \$t536=538, \$t537=539, \$t538=540, \$t539=541, \$t540=542, \$t541=543, \$t542=544, \$t543=545, \$t544=546, \$t545=547, \$t546=548, \$t547=549, \$t548=550, \$t549=551, \$t550=552, \$t551=553, \$t552=554, \$t553=555, \$t554=556, \$t555=557, \$t556=558, \$t557=559, \$t558=560, \$t559=561, \$t560=562, \$t561=563, \$t562=564, \$t563=565, \$t564=566, \$t565=567, \$t566=568, \$t567=569, \$t568=570, \$t569=571, \$t570=572, \$t571=573, \$t572=574, \$t573=575, \$t574=576, \$t575=577, \$t576=578, \$t577=579, \$t578=580, \$t579=581, \$t580=582, \$t581=583, \$t582=584, \$t583=585, \$t584=586, \$t585=587, \$t586=588, \$t587=589, \$t588=590, \$t589=591, \$t590=592, \$t591=593, \$t592=594, \$t593=595, \$t594=596, \$t595=597, \$t596=598, \$t597=599, \$t598=600, \$t599=601, \$t600=602, \$t601=603, \$t602=604, \$t603=605, \$t604=606, \$t605=607, \$t606=608, \$t607=609, \$t608=610, \$t609=611, \$t610=612, \$t611=613, \$t612=614, \$t613=615, \$t614=616, \$t615=617, \$t616=618, \$t617=619, \$t618=620, \$t619=621, \$t620=622, \$t621=623, \$t622=624, \$t623=625, \$t624=626, \$t625=627, \$t626=628, \$t627=629, \$t628=630, \$t629=631, \$t630=632, \$t631=633, \$t632=634, \$t633=635, \$t634=636, \$t635=637, \$t636=638, \$t637=639, \$t638=640, \$t639=641, \$t640=642, \$t641=643, \$t642=644, \$t643=645, \$t644=646, \$t645=647, \$t646=648, \$t647=649, \$t648=650, \$t649=651, \$t650=652, \$t651=653, \$t652=654, \$t653=655, \$t654=656, \$t655=657, \$t656=658, \$t657=659, \$t658=660, \$t659=661, \$t660=662, \$t661=663, \$t662=664, \$t663=665, \$t664=666, \$t665=667, \$t666=668, \$t667=669, \$t668=670, \$t669=671, \$t670=672, \$t671=673, \$t672=674, \$t673=675, \$t674=676, \$t675=677, \$t676=678, \$t677=679, \$t678=680, \$t679=681, \$t680=682, \$t681=683, \$t682=684, \$t683=685, \$t684=686, \$t685=687, \$t686=688, \$t687=689, \$t688=690, \$t689=691, \$t690=692, \$t691=693, \$t692=694, \$t693=695, \$t694=696, \$t695=697, \$t696=698, \$t697=699, \$t698=700, \$t699=701, \$t700=702, \$t701=703, \$t702=704, \$t703=705, \$t704=706, \$t705=707, \$t706=708, \$t707=709, \$t708=710, \$t709=711, \$t710=712, \$t711=713, \$t712=714, \$t713=715, \$t714=716, \$t715=717, \$t716=718, \$t717=719, \$t718=720, \$t719=721, \$t720=722, \$t721=723, \$t722=724, \$t723=725, \$t724=726, \$t725=727, \$t726=728, \$t727=729, \$t728=730, \$t729=731, \$t730=732, \$t731=733, \$t732=734, \$t733=735, \$t734=736, \$t735=737, \$t736=738, \$t737=739, \$t738=740, \$t739=741, \$t740=742, \$t741=743, \$t742=744, \$t743=745, \$t744=746, \$t745=747, \$t746=748, \$t747=749, \$t748=750, \$t749=751, \$t750=752, \$t751=753, \$t752=754, \$t753=755, \$t754=756, \$t755=757, \$t756=758, \$t757=759, \$t758=760, \$t759=761, \$t760=762, \$t761=763, \$t762=764, \$t763=765, \$t764=766, \$t765=767, \$t766=768, \$t767=769, \$t768=770, \$t769=771, \$t770=772, \$t771=773, \$t772=774, \$t773=775, \$t774=776, \$t775=777, \$t776=778, \$t777=779, \$t778=780, \$t779=781, \$t780=782, \$t781=783, \$t782=784, \$t783=785, \$t784=786, \$t785=787, \$t786=788, \$t787=789, \$t788=790, \$t789=791, \$t790=792, \$t791=793, \$t792=794, \$t793=795, \$t794=796, \$t795=797, \$t796=798, \$t797=799, \$t798=800, \$t799=801, \$t800=802, \$t801=803, \$t802=804, \$t803=805, \$t804=806, \$t805=807, \$t806=808, \$t807=809, \$t808=810, \$t809=811, \$t810=812, \$t811=813, \$t812=814, \$t813=815, \$t814=816, \$t815=817, \$t816=818, \$t817=819, \$t818=820, \$t819=821, \$t820=822, \$t821=823, \$t822=824, \$t823=825, \$t824=826, \$t825=827, \$t826=828, \$t827=829, \$t828=830, \$t829=831, \$t830=832, \$t831=833, \$t832=834, \$t833=835, \$t834=836, \$t835=837, \$t836=838, \$t837=839, \$t838=840, \$t839=841, \$t840=842, \$t841=843, \$t842=844, \$t843=845, \$t844=846, \$t845=847, \$t846=848, \$t847=849, \$t848=850, \$t849=851, \$t850=852, \$t851=853, \$t852=854, \$t853=855, \$t854=856, \$t855=857, \$t856=858, \$t857=859, \$t858=860, \$t859=861, \$t860=862, \$t861=863, \$t862=864, \$t863=865, \$t864=866, \$t865=867, \$t866=868, \$t867=869, \$t868=870, \$t869=871, \$t870=872, \$t871=873, \$t872=874, \$t873=875, \$t874=876, \$t875=877, \$t876=878, \$t877=879, \$t878=880, \$t879=881, \$t880=882, \$t881=883, \$t882=884, \$t883=885, \$t884=886, \$t885=887, \$t886=888, \$t887=889, \$t888=890, \$t889=891, \$t890=892, \$t891=893, \$t892=894, \$t893=895, \$t894=896, \$t895=897, \$t896=898, \$t897=899, \$t898=900, \$t899=901, \$t900=902, \$t901=903, \$t902=904, \$t903=905, \$t904=906, \$t905=907, \$t906=908, \$t907=909, \$t908=910, \$t909=911, \$t910=912, \$t911=913, \$t912=914, \$t913=915, \$t914=916, \$t915=917, \$t916=918, \$t917=919, \$t918=920, \$t919=921, \$t920=922, \$t921=923, \$t922=924, \$t923=925, \$t924=926, \$t925=927, \$t926=928, \$t927=929, \$t928=930, \$t929=931, \$t930=932, \$t931=933, \$t932=934, \$t933=935, \$t934=936, \$t935=937, \$t936=938, \$t937=939, \$t938=940, \$t939=941, \$t940=942, \$t941=943, \$t942=944, \$t943=945, \$t944=946, \$t945=947, \$t946=948, \$t947=949, \$t948=950, \$t949=951, \$t950=952, \$t951=953, \$t952=954, \$t953=955, \$t954=956, \$t955=957, \$t956=958, \$t957=959, \$t958=960, \$t959=961, \$t960=962, \$t961=963, \$t962=964, \$t963=965, \$t964=966, \$t965=967, \$t966=968, \$t967=969, \$t968=970, \$t969=971, \$t970=972, \$t971=973, \$t972=974, \$t973=975, \$t974=976, \$t975=977, \$t976=978, \$t977=979, \$t978=980, \$t979=981, \$t980=982, \$t981=983, \$t982=984, \$t983=985, \$t984=986, \$t985=987, \$t986=988, \$t987=989, \$t988=990, \$t989=991, \$t990=992, \$t991=993, \$t992=994, \$t993=995, \$t994=996, \$t995=997, \$t996=998, \$t997=999, \$t998=1000, \$t999=1001, \$t1000=1002, \$t1001=1003, \$t1002=1004, \$t1003=1005, \$t1004=1006, \$t1005=1007, \$t1006=1008, \$t1007=1009, \$t1008=1010, \$t1009=1011, \$t1010=1012, \$t1011=1013, \$t1012=1014, \$t1013=1015, \$t1014=1016, \$t1015=1017, \$t1016=1018, \$t1017=1019, \$t1018=1020, \$t1019=1021, \$t1020=1022, \$t1021=1023, \$t1022=1024, \$t1023=1025, \$t1024=1026, \$t1025=1027, \$t1026=1028, \$t1027=1029, \$t1028=1030, \$t1029=1031, \$t1030=1032, \$t1031=1033, \$t1032=1034, \$t1033=1035, \$t1034=1036, \$t1035=1037, \$t1036=1038, \$t1037=1039, \$t1038=1040, \$t1039=1041, \$t1040=1042, \$t1041=1043, \$t1042=1044, \$t1043=1045, \$t1044=1046, \$t1045=1047, \$t1046=1048, \$t1047=1049, \$t1048=10410, \$t1049=10411, \$t10410=10411, \$t10411=10412, \$t10412=10413, \$t10413=10414, \$t10414=10415, \$t10415=10416, \$t10416=10417, \$t10417=10418, \$t10418=10419, \$t10419=10420, \$t10420=10421, \$t10421=10422, \$t10422=10423, \$t10423=10424, \$t10424=10425, \$t10425=10426, \$t10426=10427, \$t10427=10428, \$t10428=10429, \$t10429=10430, \$t10430=10431, \$t10431=10432, \$t10432=10433, \$t10433=10434, \$t10434=10435, \$t10435=10436, \$t10436=10437, \$t10437=10438, \$t10438=10439, \$t10439=10440, \$t10440=10441, \$t10441=10442, \$t10442=10443, \$t10443=10444, \$t10444=10445, \$t10445=10446, \$t10446=10447, \$t10447=10448, \$t10448=10449, \$t10449=10450, \$t10450=10451, \$t10451=10452, \$t10452=10453, \$t10453=10454, \$t10454=10455, \$t10455=10456, \$t10456=1045

The screenshot shows the MARS 4.5 assembly editor interface. The assembly window displays the following code:

```

Text Segment
Bkpt Address Code Basic Source
0x00000000 0x00100000 la $1, $00000100 B: la $18, X    # Get the address of X in Data Segment
0x00000004 0x00100004 la $2, $00000000
0x00000008 0x00100008 la $3, $00000000
0x0000000c 0x0010000c la $9, Y    # Get the address of Y in Data Segment
0x0000000e 0x0010000e la $10, Z   # Get the address of Z in Data Segment
0x00000010 0x00100010 lw $9, 0x00000000($24) 10: lw $11, 0($18)   # $11 = X
0x00000014 0x00100014 lw $12, 0($19)   # $12 = Y
0x00000018 0x00100018 add $11, $12, $11 11: add $11, $12, $11   # $11 = $11 + $12 - X + X - 2X
0x0000001c 0x0010001c add $10, $9, $10 13: add $10, $9, $10   # $10 = $10 + $9 - 2Y + Y
0x00000020 0x00100020 add $16, $10, $10 14: add $16, $10, $10   # $16 = $16 + $10 - 2X + Y
0x00000024 0x00100024 la $1, $00000001 16: la $17, Z   # Get the address of Z in Data Segment
0x00000028 0x00100028 add $16, $00000000($15) 17: sw $50, $0($17)   # Z = $50 - 2X + Y

```

The Data Segment window shows memory starting at address 0x10000000.

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10000000	0x00000005	0xffffffff	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000004	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000008	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1000000c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1000000e	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000010	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000014	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000018	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1000001c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000024	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000028	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1000002c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000030	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000034	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000038	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1000003c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000044	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000048	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1000004c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000050	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000054	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000058	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1000005c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000064	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000068	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000070	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000074	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000078	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1000007c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000084	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000088	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1000008c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000090	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000094	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10000098	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1000009c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100000a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100000a4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100000a8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100000ac	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100000b0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100000b4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100000b8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100000bc	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100000cc	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100000dc	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100000ec	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100000fc	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

The Registers window shows the following register values:

Name	Number	Value
\$zero	0	0x00000000
\$t1	1	0x00000000
\$v0	2	0x00000000
\$t3	3	0x00000000
\$s0	4	0x00000000
\$t1	5	0x00000000
\$t2	6	0x00000000
\$t3	7	0x00000000
\$t0	8	0x00000000
\$t5	9	0x00000000
\$t2	10	0x00000000
\$t4	11	0x00000000
\$t5	12	0x00000000
\$t6	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$t7	16	0x00000000
\$t8	17	0x00000000
\$t9	18	0x00000000
\$t9	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x00000000
\$sp	29	0x7fffffe0
\$fp	30	0x00000000
\$ra	31	0x00000000
pc	32	0x00400000
hi	33	0x00000000
lo	34	0x00000000

The Mars Messages window shows the assembly process completed successfully.

Text Segment				
Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c011001	lui \$1,0x00001001	8: la \$t8, X # Get the address of X in Data Segment
	0x00400004	0x34380000	ori \$24,\$1,0x00000000	
	0x00400008	0x3c011001	lui \$1,0x00001001	9: la \$t9, Y # Get the address of Y in Data Segment
	0x0040000c	0x34390004	ori \$25,\$1,0x00000004	
	0x00400010	0x8f090000	lw \$9,0x00000000(\$24)	10: lw \$t1, 0(\$t8) # \$t1 = X
	0x00400014	0x8f2a0000	lw \$10,0x00000000(\$25)	11: lw \$t2, 0(\$t9) # \$t2 = Y
	0x00400018	0x01298020	add \$16,\$9,\$9	13: add \$s0,\$t1,\$t1 # \$s0 = \$t1 + \$t1 = X + X = 2X
	0x0040001c	0x202a0820	add \$16,\$16,\$10	14: add \$s0,\$s0,\$t2 # \$s0 = \$s0 + \$t2 = 2X + Y
	0x00400020	0x3c011001	lui \$1,0x00001001	16: la \$t7, Z # Get the address of Z in Data Segment
	0x00400024	0x342f0008	ori \$15,\$1,0x00000008	
	0x00400028	0xadff0000	sw \$16,0x00000000(\$15)	17: sw \$s0, 0(\$t7) # Z = \$s0 = 2X + Y

Label	Address
<b>Exercise6.asm</b>	
X	0x10010000
Y	0x10010004
Z	0x10010008

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0x0000000000
\$at	1	0x0000000000
\$v0	2	0x0000000000
\$vl	3	0x0000000000
\$a0	4	0x0000000000
\$a1	5	0x0000000000
\$a2	6	0x0000000000
\$a3	7	0x0000000000
\$t0	8	0x0000000000
\$t1	9	0x0000000000
\$t2	10	0x0000000000
\$t3	11	0x0000000000
\$t4	12	0x0000000000
\$t5	13	0x0000000000
\$t6	14	0x0000000000
\$t7	15	0x0000000000
\$s0	16	0x0000000000
\$s1	17	0x0000000000
\$s2	18	0x0000000000
\$s3	19	0x0000000000
\$s4	20	0x0000000000
\$s5	21	0x0000000000
\$s6	22	0x0000000000
\$s7	23	0x0000000000
\$t8	24	0x0000000000
\$t9	25	0x0000000000
\$k0	26	0x0000000000
\$k1	27	0x0000000000
\$gp	28	0x10008000
\$sp	29	0x7ffffefffc
\$fp	30	0x0000000000
\$ra	31	0x0000000000
pc		0x00400000
hi		0x00000000
lo		0x0000000000

- Lệnh la được biên dịch bằng cách tách thành 2 lệnh basic là lui và ori.

- Ở cửa sổ Data Segment :

Địa chỉ của X = \$t8

Địa chỉ của Y = \$t9

Địa chỉ của Z = \$t7

Tương ứng với hằng số khi biên dịch lệnh la thành mã máy.

- *Chạy từng dòng lệnh:*

Step 1:

**Step 2:**

The screenshot shows the GDB interface with two panes. The left pane displays the assembly code for the program, and the right pane shows the register values.

**Registers (Coproc 0):**

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$t8	22	0x00000000
\$t9	23	0x00000000
\$t10	24	0x00000000
\$t11	25	0x00000000
\$k1	26	0x00000000
\$sp	27	0x00000000
\$fp	28	0x00000000
\$tfr	29	0x7ffffeffc
\$ra	30	0x00000000
pc	31	0x00000000
hi	32	0x00000000
lo	33	0x00000000

**Step 3:**

The screenshot shows the GDB interface with two panes. The left pane displays the assembly code for the program, and the right pane shows the register values.

**Registers (Coproc 0):**

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$t8	22	0x00000000
\$t9	23	0x00000000
\$t10	24	0x00000000
\$t11	25	0x00000000
\$k1	26	0x00000000
\$sp	27	0x00000000
\$fp	28	0x00000000
\$tfr	29	0x7ffffeffc
\$ra	30	0x00000000
pc	31	0x00000000
hi	32	0x00000000
lo	33	0x00000000

**Step 4:**

The screenshot shows the GDB interface with two panes. The left pane displays the assembly code for the program, and the right pane shows the register values.

**Registers (Coproc 0):**

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$t8	22	0x00000000
\$t9	23	0x00000000
\$t10	24	0x10010000
\$t11	25	0x00000000
\$k1	26	0x00000000
\$sp	27	0x00000000
\$fp	28	0x00000000
\$tfr	29	0x7ffffeffc
\$ra	30	0x00000000
pc	31	0x00000000
hi	32	0x00000000
lo	33	0x00000000

The screenshot shows the MARS 4.5 assembly debugger interface with several windows open:

- Registers**: Shows the CPU registers (r0 to r31) with their names, numbers, and current values.
- Stack**: Shows the stack contents from address 0x00000000 to 0x00000028.
- Registers**: Shows the CPU registers again, this time with the stack pointer (sp) at the top.
- Text Segment**: Displays the assembly code with comments and assembly mnemonics.
- Data Segment**: Displays the memory dump of the data segment.
- Mars Messages**: Displays the message "... program is finished running (dropped off bottom) .."

## Step 5:

### Step 6:

The figure shows the MARS 4.5 assembly debugger interface with the following windows:

- Registers**: Shows the CPU register state. The \$12 register is highlighted in green.
- Data Segment**: Shows memory starting at address 0x10000000. The value at address 0x10000000 is \$00000005.
- Mars Messages**: Displays the message "... program is finished running (dropped off bottom) .."

### Step 7:

The screenshot shows the MARS 4.5 assembly debugger interface with several windows open:

- Registers**: Shows the CPU register state. The \$r0 register is set to 0.
- Data Segment**: Displays memory starting at address 0x10000000. It includes columns for Address, Value (+0), Value (+4), Value (+8), Value (+c), Value (+10), Value (+14), Value (+18), and Value (+1c). The first few entries show values like 0x00000005, 0xffffffff, and 0x00000000.
- Text Segment**: Shows the assembly code. The code starts with a la \$t0, X instruction followed by a la \$t1, Y instruction. These are annotated with comments: "# Get the address of X in Data Segment" and "# Get the address of Y in Data Segment". The next instruction is add \$t0, \$t1, \$t1, which is annotated with "# \$0 = \$t0 + \$t1 - X + X - 2X". This is followed by add \$t0, \$t1, \$t2, annotated with "# \$0 = \$t0 + \$t1 + 2X + Y". The final instruction is la \$t7, Z, annotated with "# Get the address of Z in Data Segment".
- Mars Messages**: Displays the message: "program is finished running (dropped off bottom) .."

### Step 8:

The screenshot shows the MARS 4.5 assembly debugger interface with several windows open:

- Registers**: Shows the processor state with registers \$zero through \$t12, \$s0 through \$s3, \$t13 through \$t15, \$t16, \$t17, \$t18, \$t19, \$t20, \$t21, \$t22, \$t23, \$t24, \$t25, \$t26, \$t27, \$t28, \$t29, \$t30, \$t31, \$ra, \$pc, and \$hi. \$t17 is highlighted in green.
- Stack**: Shows the stack contents from address 0x00000000 to 0x0000000f.
- Registers**: Shows the processor state with registers \$zero through \$t12, \$s0 through \$s3, \$t13 through \$t15, \$t16, \$t17, \$t18, \$t19, \$t20, \$t21, \$t22, \$t23, \$t24, \$t25, \$t26, \$t27, \$t28, \$t29, \$t30, \$t31, \$ra, \$pc, and \$hi. \$t17 is highlighted in green.
- Text Segment**: Displays the assembly code for the basic section. The code includes instructions like `la $t8, X`, `la $t9, Y`, `lw $t1, 0($t8)`, `add $t1, $t1, $t1`, `add $t1, $t1, $t2`, `la $t7, Z`, and `sw $t0, 0($t7)`.
- Data Segment**: Displays the memory dump for the data section, showing values from 0x10000000 to 0x1000000f.
- Mars Messages**: Shows the message "... program is finished running (dropped off bottom) .."

### Step 9:

The screenshot shows the MARS 4.5 assembly debugger interface. The Registers window displays the CPU state with various registers containing values like \$zero=0, \$t0=1, \$v0=2, etc. The Stack window shows the stack contents from address 0x10000000 to 0x1000000f. The Data Segment window shows memory starting at address 0x10000000 with various fields like Value (+0) through Value (+11). The bottom pane displays Mars Messages and Run I/O, indicating the program has finished running.

### Step 10:

## Step 11:

The figure shows the MARS 4.5 assembly debugger interface. The Registers window displays the processor state with \$17 highlighted. The Data Segment window shows memory starting at address 0x10000000. The Mars Messages window indicates the program has finished running.

- Sự thay đổi giá trị các thanh ghi :

\$at : 0x00000000 => 0x10010000

\$t8 : 0x00000000 => 0x10010000

\$t9 : 0x00000000 => 0x10010004

\$t1 : 0x00000000 => 0x00000005

\$t2 : 0x00000000 => 0xffffffff

`$s0 : 0x00000000 => 0x0000000a => 0x00000009`

`$t7 : 0x00000000 => 0x10010008`

- Vai trò của lệnh lw và sw :

+ lw \$rt, imm(\$rs) : gán giá trị của thanh ghi \$rs vào thanh ghi \$rt ( $\$rt = M[\$rs+imm]$ )

+ Ở đây lw gán \$t1 = \$t8 và \$t2 = \$t9

+ sw \$rt, imm(\$rs) : gán giá trị của thanh ghi \$rt vào thanh ghi \$rs ( $M[\$rs+imm] = \$rt$ )

- Các lệnh lb, sb:

+ lb : chép 1 byte tại vị trí trong bộ nhớ RAM vào byte thấp của thanh ghi.

+ sb : lưu một byte thấp trong thanh ghi vào vị trí trong bộ nhớ RAM.