

Justin VanWinkle

A) Design C

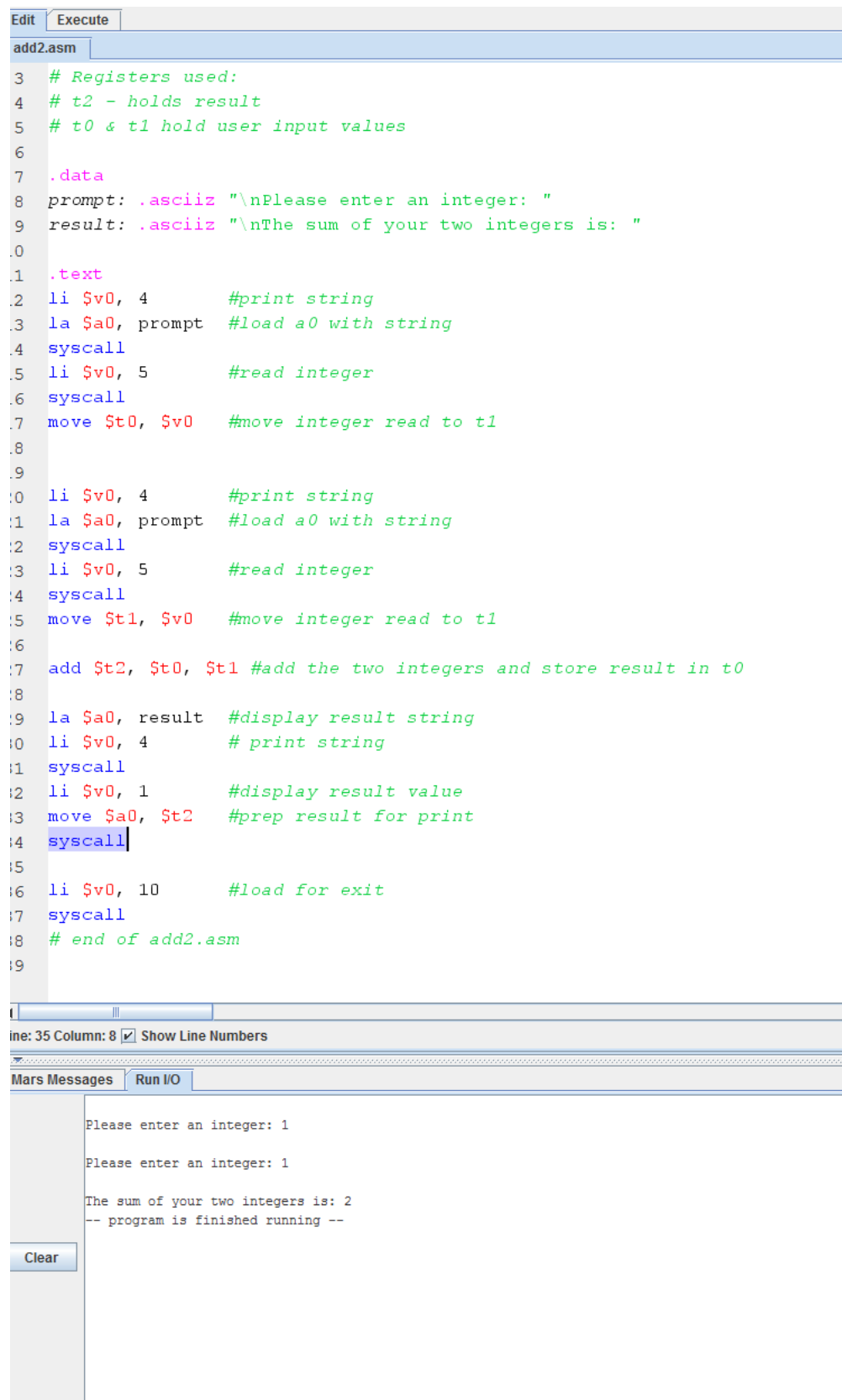
Pair	# instructions	Time/instr.	# r&w	Time/r&w	Total time
1&3	6	2nsec	4	7nsec	40nsec
1&4	6	2nsec	4	4nsec	28nsec
2&3	6	1nsec	4	7nsec	34nsec

The combination of the MIPS 34K CPU and the Toshiba DDR3-1333 memory provide the lowest possible execution time within the given parameters.

B) Design C

Pair	# instructions	Time/instr.	# r&w	Time/r&w	Total time
1&3	4	2nsec	3	7nsec	29nsec
1&4	4	2nsec	3	4nsec	20nsec
2&3	4	1nsec	3	7nsec	25nsec

The combination of the MIPS 34K CPU and the Toshiba DDR3-1333 memory provide the lowest possible execution time within the given parameters.



The screenshot displays the Mars IDE interface. The top section is the code editor, showing an assembly program named `add2.asm`. The code is as follows:

```
3  # Registers used:
4  # t2 - holds result
5  # t0 & t1 hold user input values
6
7  .data
8  prompt: .asciiz "\nPlease enter an integer: "
9  result: .asciiz "\nThe sum of your two integers is: "
10
11 .text
12 li $v0, 4      #print string
13 la $a0, prompt #load a0 with string
14 syscall
15 li $v0, 5      #read integer
16 syscall
17 move $t0, $v0  #move integer read to t1
18
19
20 li $v0, 4      #print string
21 la $a0, prompt #load a0 with string
22 syscall
23 li $v0, 5      #read integer
24 syscall
25 move $t1, $v0  #move integer read to t1
26
27 add $t2, $t0, $t1 #add the two integers and store result in t0
28
29 la $a0, result #display result string
30 li $v0, 4      # print string
31 syscall
32 li $v0, 1      #display result value
33 move $a0, $t2  #prep result for print
34 syscall
35
36 li $v0, 10     #load for exit
37 syscall
38 # end of add2.asm
39
```

Below the code editor is a status bar showing "Line: 35 Column: 8" and a checked "Show Line Numbers" option. At the bottom is the "Mars Messages" window, which contains the following output:

```
Please enter an integer: 1
Please enter an integer: 1
The sum of your two integers is: 2
-- program is finished running --
```

A "Clear" button is located at the bottom left of the Mars Messages window.

```
3  # Registers used:
4  # t2 - holds result
5  # t0 & t1 hold user input values
6
7  .data
8  prompt: .asciiz "\nPlease enter an integer: "
9  result: .asciiz "\nThe sum of your two integers is: "
10
11 .text
12 li $v0, 4      #print string
13 la $a0, prompt #load a0 with string
14 syscall
15 li $v0, 5      #read integer
16 syscall
17 move $t0, $v0  #move integer read to t1
18
19
20 li $v0, 4      #print string
21 la $a0, prompt #load a0 with string
22 syscall
23 li $v0, 5      #read integer
24 syscall
25 move $t1, $v0  #move integer read to t1
26
27 add $t2, $t0, $t1 #add the two integers and store result in t2
28
29 la $a0, result  #display result string
30 li $v0, 4      # print string
31 syscall
32 li $v0, 34     #display result value
33 move $a0, $t2  #prep result for print
34 syscall
35
36 li $v0, 10     #load for exit
37 syscall
38 # end of add2.asm
39
```

Line: 35 Column: 8 ☒ Show Line Numbers

Mars Messages Run I/O

Please enter an integer: 1

Please enter an integer: 1

The sum of your two integers is: 0x00000002

-- program is finished running --

Clear

The screenshot shows the Mars IDE interface. The top section is the code editor, titled 'increment.asm', containing assembly code. The code starts with a comment indicating it was created by Justin VanWinkle on 09/14/2015. It defines a loop that prints integers from 0 to 15. The code uses MIPS assembly instructions: `li` (load immediate), `la` (load address), `syscall` (system call), `addi` (add immediate), and `bne` (branch not equal). The bottom section is the 'Mars Messages' window, which shows the output of the program. The output consists of the numbers 0 through 15, each on a new line, followed by the message '-- program is finished running --'. The 'Run I/O' button is visible in the top right of the messages window.

```
1  # Justin VanWinkle -- 09/14/2015
2  # increment.asm -- A program that increments from 0 to 15
3
4  .text
5  li $t0, 0      #load 0 in t0
6  li $t1, 16     #load 16 in t1
7
8  loop:          #label
9  la $a0, ($t0)   # stage $t0 to be printed
10 li $v0, 1       #print integer
11 syscall
12 li $a0, 10      #load ascii code for LF
13 li $v0, 11      #print lower 8 bits of $a0
14 syscall
15 addi $t0, $t0, 1 #increment by 1
16 bne $t0, $t1, loop #loop if not equal
17
18 li $v0, 10      #exit
19 syscall
20 # end increment.asm
21
```

Line: 3 Column: 1 ☒ Show Line Numbers

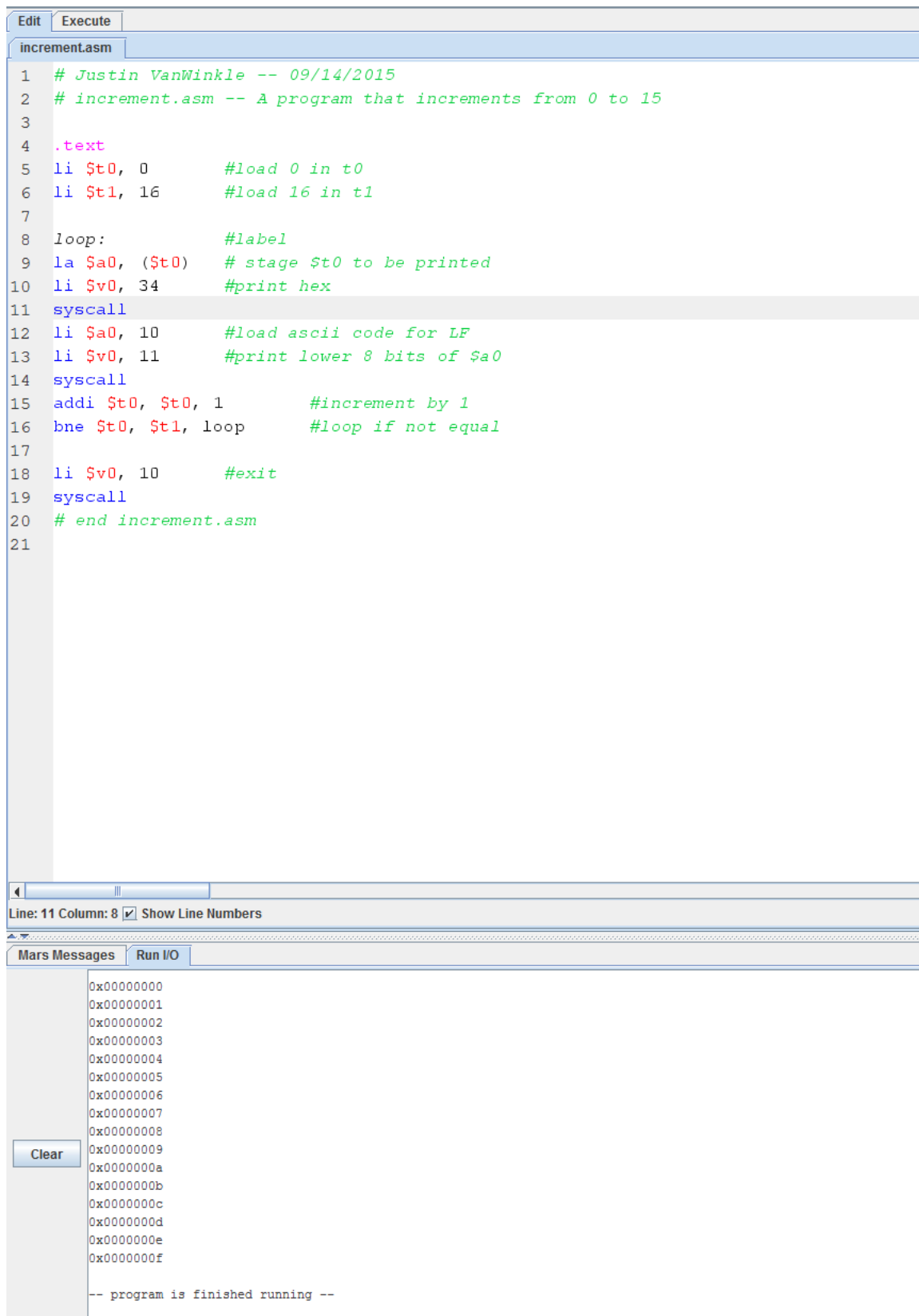
Mars Messages Run I/O

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

Clear

-- program is finished running --

Justin VanWinkle



The screenshot displays the Mars MIPS assembler interface. The top section, titled 'increment.asm', contains the assembly code for a program that increments a value from 0 to 15. The code is as follows:

```
1  # Justin VanWinkle -- 09/14/2015
2  # increment.asm -- A program that increments from 0 to 15
3
4  .text
5  li $t0, 0      #load 0 in t0
6  li $t1, 16     #load 16 in t1
7
8  loop:         #label
9  la $a0, ($t0)  # stage $t0 to be printed
10 li $v0, 34     #print hex
11 syscall
12 li $a0, 10     #load ascii code for LF
13 li $v0, 11     #print lower 8 bits of $a0
14 syscall
15 addi $t0, $t0, 1      #increment by 1
16 bne $t0, $t1, loop    #loop if not equal
17
18 li $v0, 10           #exit
19 syscall
20 # end increment.asm
21
```

Below the code editor, the status bar indicates 'Line: 11 Column: 8' and 'Show Line Numbers' is checked. The bottom section, titled 'Mars Messages', shows the output of the program, which is a sequence of hexadecimal values from 0x00000000 to 0x0000000f, followed by the message '-- program is finished running --'. A 'Clear' button is located to the left of the output list.