**Task 3:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Source code instruction | Register(s) modified | Value in the modified register(s) | Memory location(s) modified. | Value in modified memory location(s) |
| add $t1, $0, 16 | $t1 | 0x00000010 | None | None |
| sw $t1, 0x10000000($0) | - | - | 0x10000000 | 0x00000010 |
| lw $t2, 0x10000000($0) | $t2 | 0x00000010 | - | - |
| sub $t0, $t2, 1 | $t0 | 0x0000000f | - | - |

Table 1

**Task 4:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Source code instruction | Register(s) modified | Value in the modified register(s) | Memory location(s) modified. | Value in modified memory location(s) |
| add $t1, $0, 0x7f6b2684 | $t1 | 0x7f6b2684 | - | - |
| sw $t1, 0x10000000($0) | - | - | 0x10000000 | 0x7f6b2684 |
| lb $t2, 0x10000002($0) | $t2 | 0x0000006b | - | - |
| sub $t0, $t2, 2 | $t0 | 0x00000069 | - | - |

Table 2

**Question 1:** Comparing row 3 of Table 1 and Table 2, does register $t2 gets same or different values? Explain the reason for your answer.

**Answer:**

They change because the initial value that we write and subsequently read to memory in instructions 1 and 2 changes between the two programs.

**Task 5:**

**Question:** Explain the reason why Task5 terminates with error. Also, suggest a correction by rewriting the program below.

**Answer:**

It seems to be an attempt to exit the program by printing an integer value. However, there is a mistake in the code that would result in an error.

The issue is that you're trying to exit the program with the "exit" system call (syscall 10), but the $a0 register, which should contain the exit code, is loaded with the value from $t0. In this context, $t0 contains the value 0xff00ff00, which is not a valid exit code, and it may result in an error when calling syscall.

To solve this issue, we should set $a0 to the desired exit code, and then use syscall to exit the program. Here's the solution:

.text

add $t0, $0, 0xff00ff00 # Load a value into $t0

sw $t0, 0x10000005($0) # Store the value in memory at address 0x10000005

li $a0, 0 # Set the exit code to 0 (or any desired value)

li $v0, 10 # Load system call code 10 into $v0 (for exit)

syscall # Invoke the system call to exit

Because in the second instruction sw instruction is used with a target address which is not a beginning of a word (0x10000005). To solve the issue we can change the address to a value like 0x10000004 or parse the writing operation into multiple chunks and align the data using shift left and shift right operations.