Cmpe 361 lab1 file

Melisa SUBAŞI id:22829169256

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	16	n/a	n/a
sw $t1, 0x10000000($0)	n/a	n/a	0x10000000	16
lw $t2, 0x10000000($0)	$t2	16	n/a	n/a
sub $t0, $t2, 1	$t0	15	n/a	n/a

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	n/a	n/a
sw $t1, 0x10000000($0)	n/a	n/a	0x10000000	0x7f6b2684
lb $t2, 0x10000002($0)	$t2	0x7f6b2684	n/a	n/a
sub $t0, $t2, 2	$t0	0x7f6b2684	n/a	n/a

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:

In the third row of the first table, $t2 is set to 16 because it was loaded from the memory location where we previously stored $t1.

Conversely, in the third row of the second table, $t2 is equivalent to the value stored at memory address 0x7f6b2684. When we convert this hexadecimal address to decimal, we obtain 107 as the value stored at that specific memory location.

As a result, it's evident that $t2 in the two tables is not equal.

Task 5:

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

Answer:

$t0 is currently stored at the memory address 0x10000005, which is not aligned correctly (we're skipping a memory location). Memory addresses should be multiples of 4 for proper alignment. To rectify this issue, we can make the following adjustment:

Change:

$t0, 0x10000005($0)

To:

sw $t0, 0x10000004($0)

This modification ensures that $t0 is stored at a memory location aligned to a multiple of 4, resolving the alignment problem in the code.