**COSC 2325 Computer Organization**

**Assignment 10**

**Due: 23:59:00, 11/14/2022 (Monday)**

**Andrew Kalathra, L20498001**

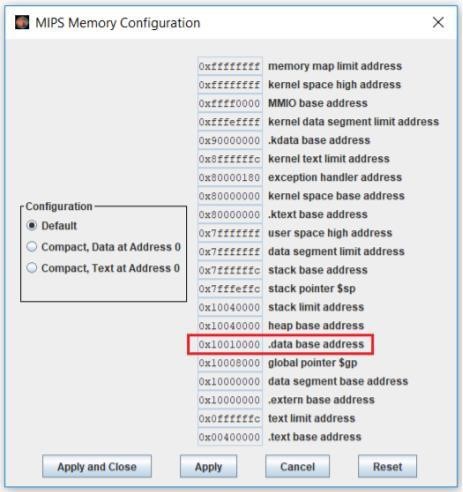
**If you use Qtspim:**

In the Settings menu of SPIM set Bare Machine OFF, Allow Pseudo Instructions ON, Load Trap File ON, Delayed Branches OFF, Delayed Loads OFF, Mapped IO ON, Quiet OFF. The starting address of Data Section is 0x10000000.

**If you use Mars:**

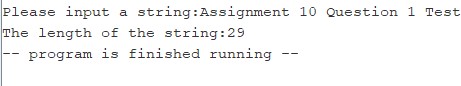
Menu → Settings → Permit extended (Pseudo) instructions and formats The starting address of Data Section may be 0x10010000

(Please check Settings->Memory Configuration… -> .data base address)



1. Please complete A10\_Q1.sm that asks the user for a string and then calculates and prints out the string length. (40 points)

Example (Mars)



Hint: Use indexed addressing learned in Chapter 24 (Chapter 24 PPT, slides 15 and 16) to scan through the string and use ‘syscall’ learned in Chapter 22 to read string (Chapter 22 PPT, slides 13 and 14) and print string and integer. Strings read in with the trap handler service include a '\n' character at the end, followed by the null termination. Don't count the '\n' or the null as part of the string length. The ASCII code of ‘\n’ is 10 (decimal). Use a loop and ‘lb’ (load byte) instruction to get the ASCII code of each character from the string. If the ASCII code of a character is not 10, increase index and counter and jump to next iteration; otherwise, stop counting.

Text, timeline

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated

Timeline

Description automatically generated with low confidence

I used an online compiler because mine (qtspim) weirdly does not take in anything. So by using this, I was able to tell that my program worked correctly.

1. Please complete A10\_Q2.asm to evaluate

18ab + 20bc – 9a + 36ac - 7

Prompt the user for the values a, b, and c. Try to use a small number of registers. Use the stack to hold intermediate values. Write the final value to the monitor. (40 points)

Hint: use ‘syscall’ learned in Chapter 22 to read string and print integer and use stack operations learned in Chapter 25 to hold intermediate values. You may also look at the example in Chapter 25.

Timeline

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Text

Description automatically generated with low confidence

1. Say that the jal instruction is at address 0x400000. The subroutine sub is at address 0x400300. What is in $ra after the jal instruction below executes? Why? (Chapter 26) (20 points)

jal sub

$ra will contain what is at the address of 0x400004 because at the execution step, the register $ra will update to the PC + 4.