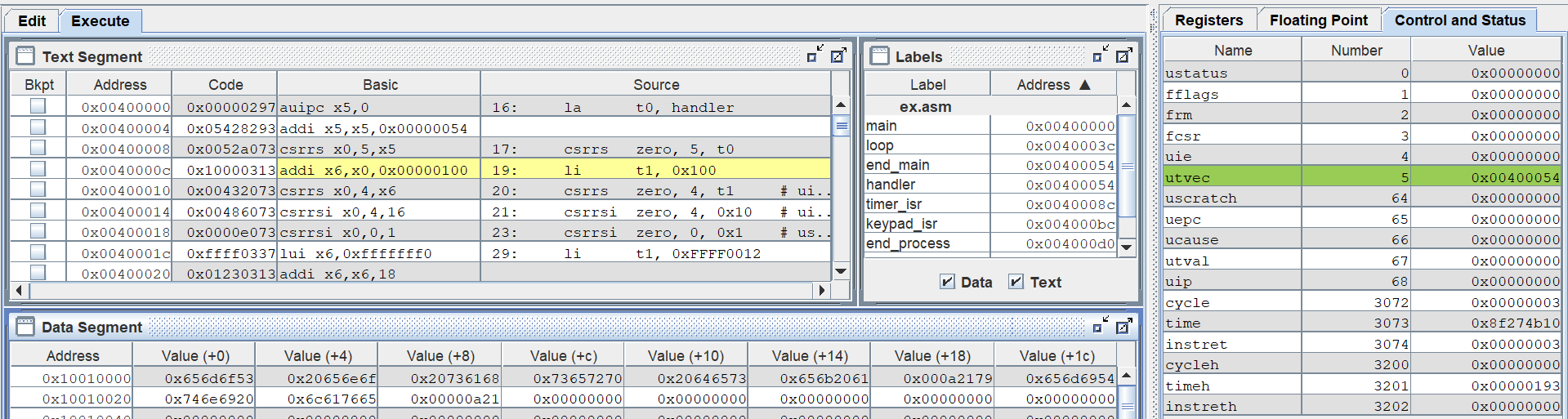
**Student name: Lê Ngọc Anh Vũ**

**Student ID: 20236014**

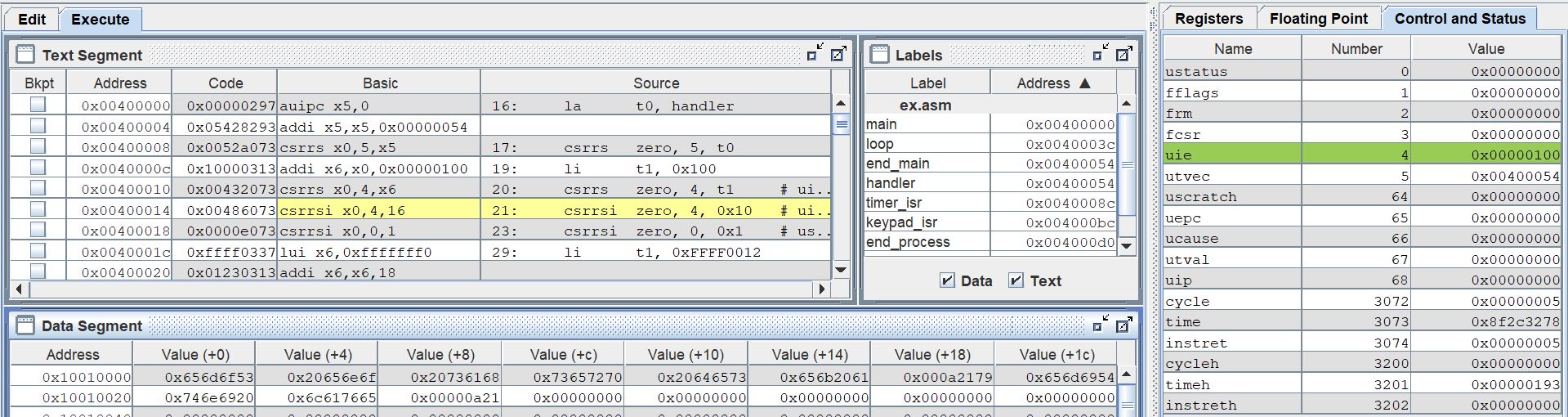
**Lab 11.2**

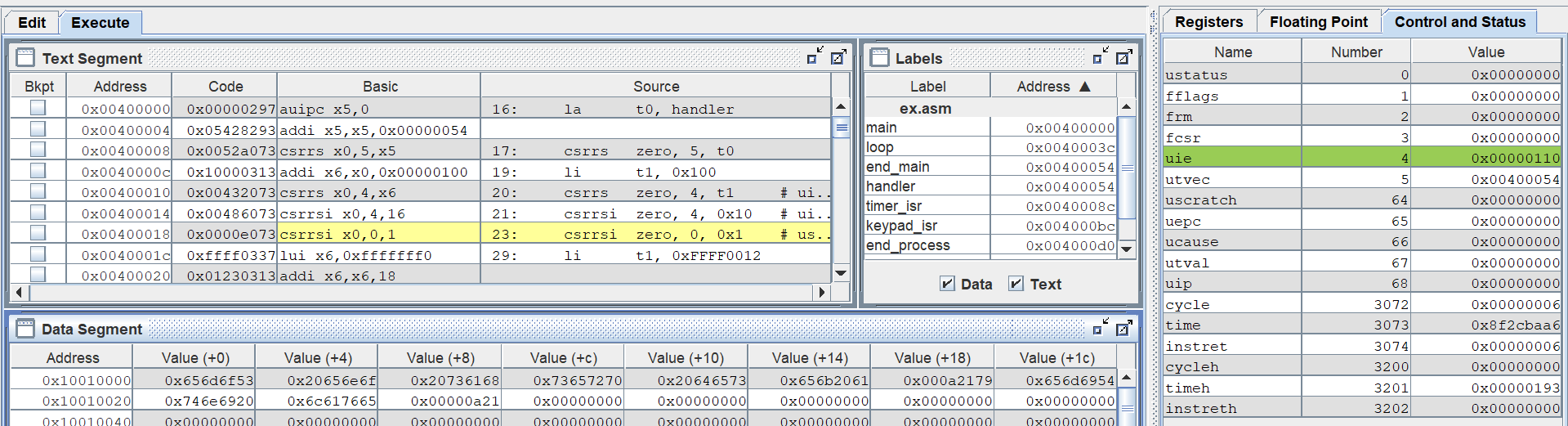
**Assignment 4: Create a new project, type in, and build the program of Home Assignment 4. Run the program step by step to understand each line of the source code.**

Load the interrupt handling routine address into the utvec register through register t0.

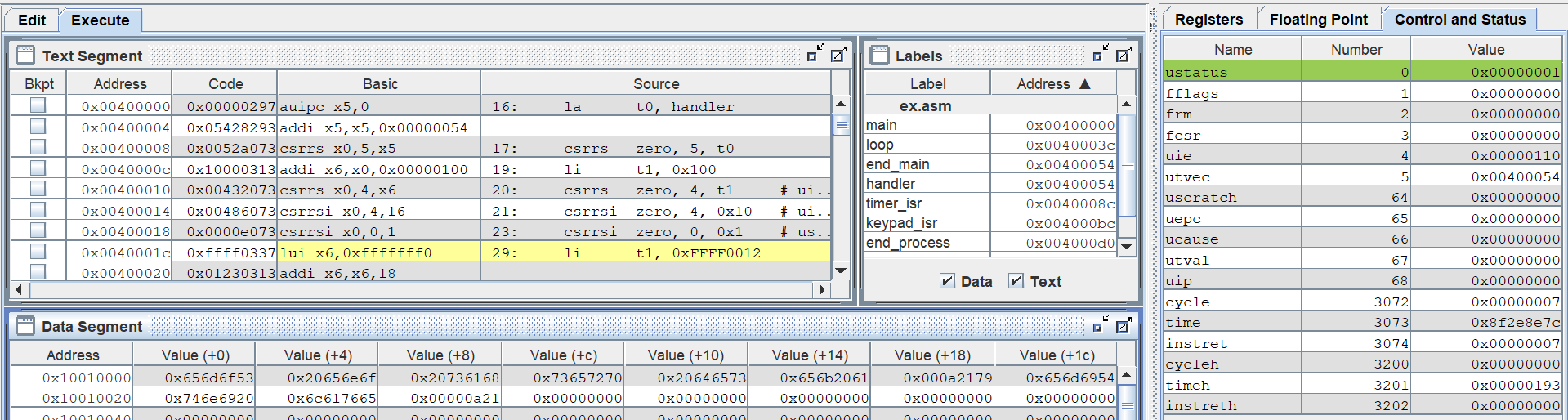


Set 1 to bit 8 (bit ueie) and bit 4 (bit utie) of register uie.

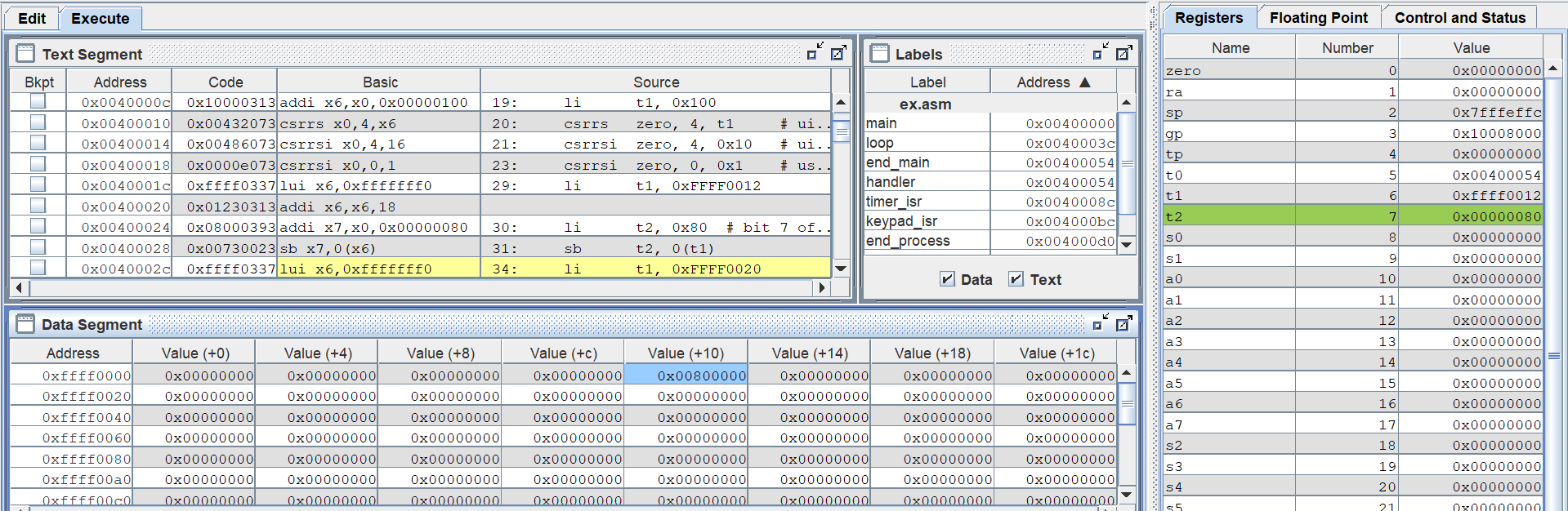




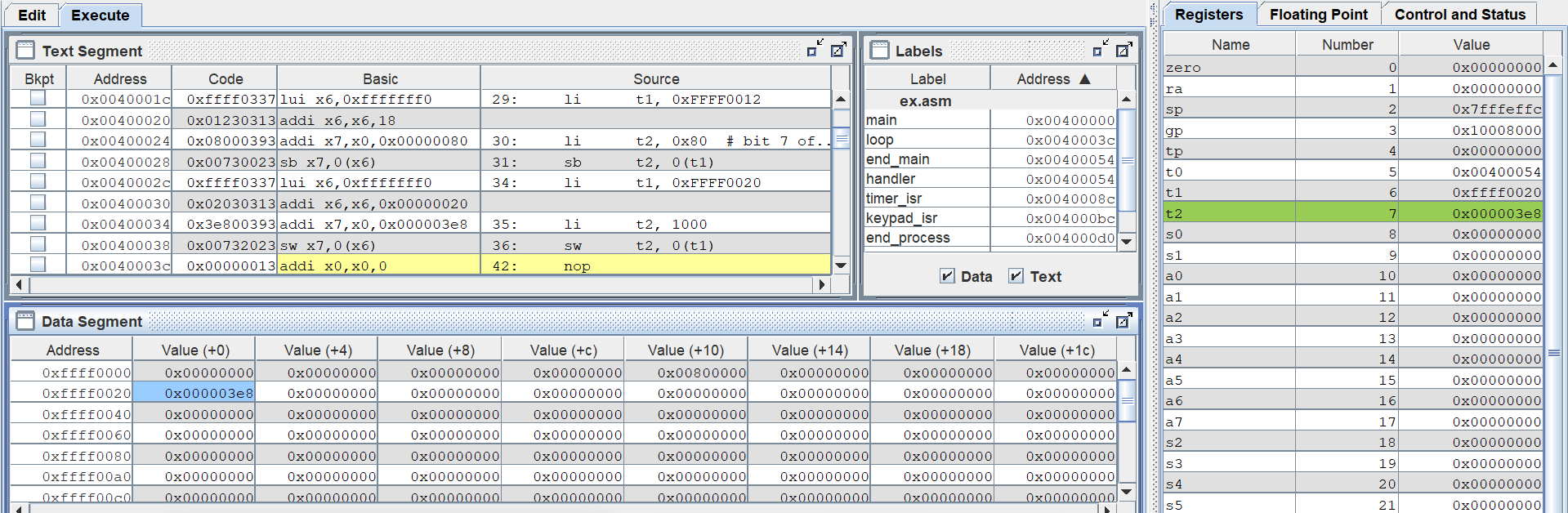
Set 1 to bit uie of register ustatus to enable interrupts.



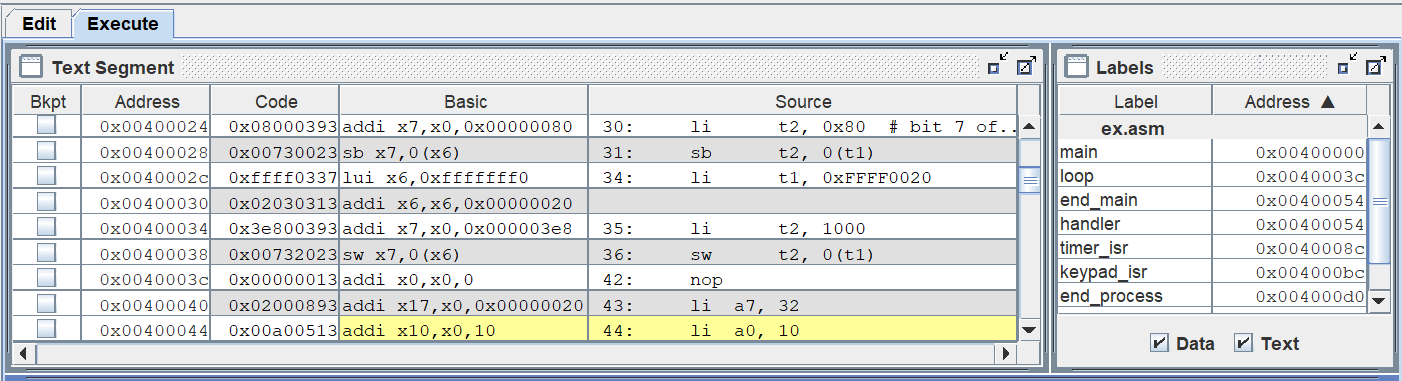
Set bit 7 of byte value at address 0xffff0012 to 1 in order to enable keyboard interrupt



Set the word at address 0xFFFF0020 containing the comparison value (in ms)

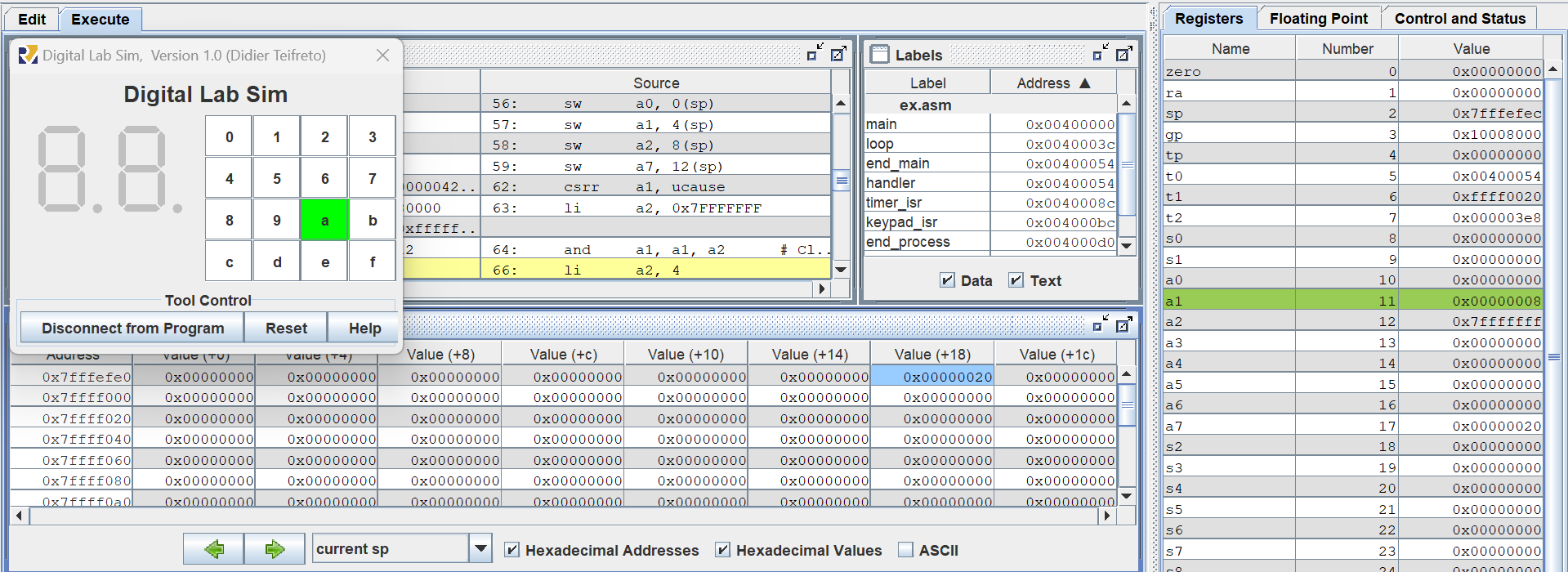


The main program executes the no-end loop until there exists interrupts.

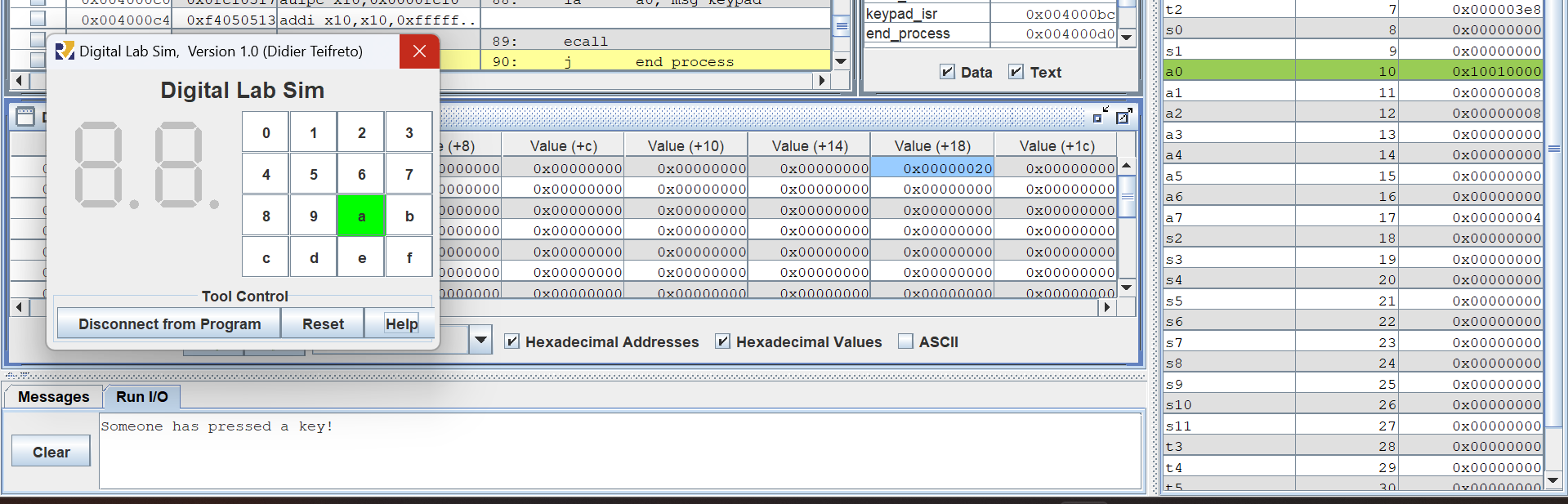


For interrupts from the keypad (Digital Lab Sim):

When an interrupt from keypad occurs, the program will jump to handler routine to handle this interrupt. The program will store the current context in stack. Get the value of EXCCODE from ucause and store in register a1 by executing operation “and” between value in ucause register and the value of 0x7fffffff stored in register a2.



Then, the program executes comparison value in register a1 with 4 and 8. In this case, a1 = 8, so this interrupt is from keypad (Digital Lab Sim). After that, the program jumps to msg\_kepad label to print message for this interrupt.



Finally, the program returns to the main and continue to execute the next instruction right after interrupt having occured and do the no-end loop to wait for the next interrupts.

For interrupts from the timer:

When an interrupt from the timer occurs which means the current value of timer has exceeded the comparison value of 1000 ms stored at address 0xffff0020, the program will jump to handler routine to handle this interrupt. The program will also store the current context in stack. Get the value of EXCCODE from ucause and store in register a1 by executing operation “and” between value in ucause register and the value of 0x7fffffff stored in register a2.

A screenshot of a computer

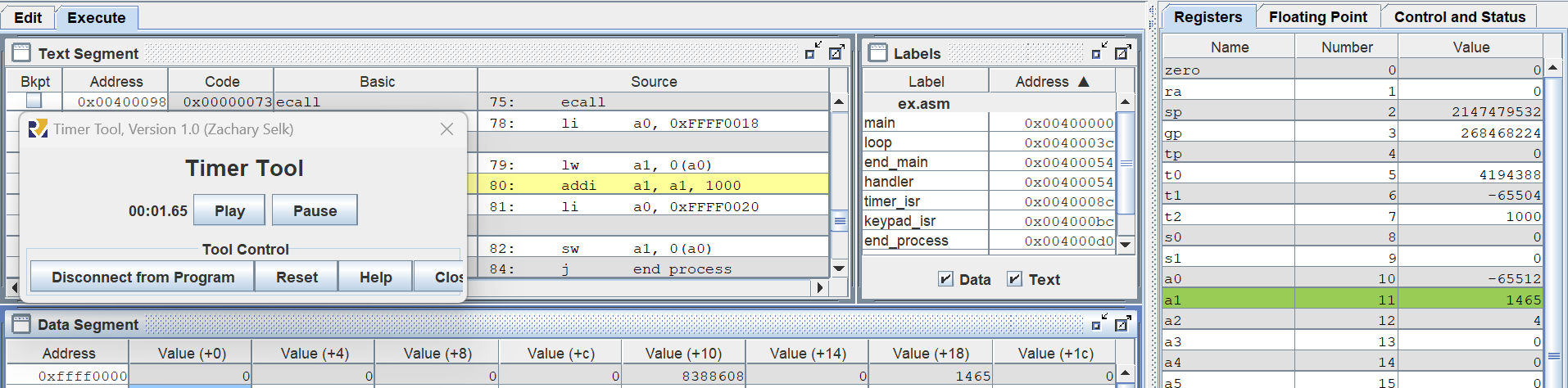
Description automatically generated

Then, the program executes comparison value in register a1 with 4 and 8. In this case, a1 = 4, so this interrupt is from the timer. After that, the program jumps to msg\_timer label to print message for this interrupt.

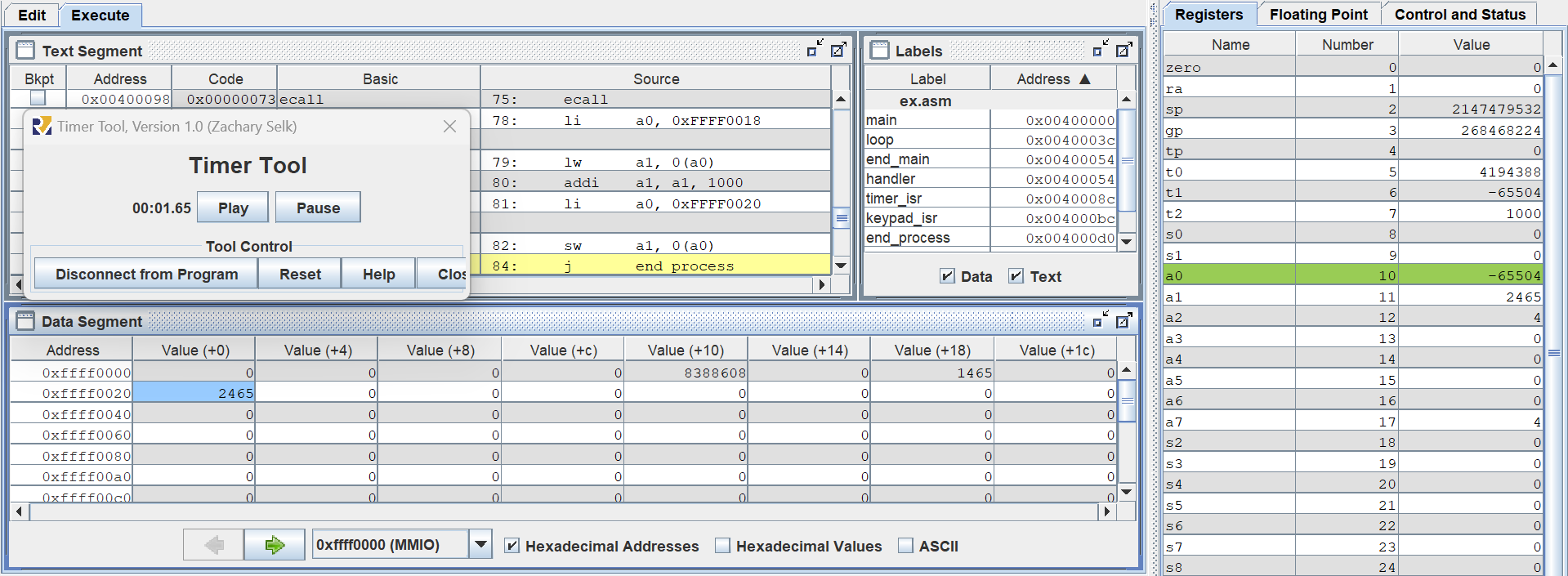
A screenshot of a computer

Description automatically generated

Then, set new value to comparison value at the address 0xffff0020. This new value is the sum of the current value and 1000. The current value is 1465 stored in register a1



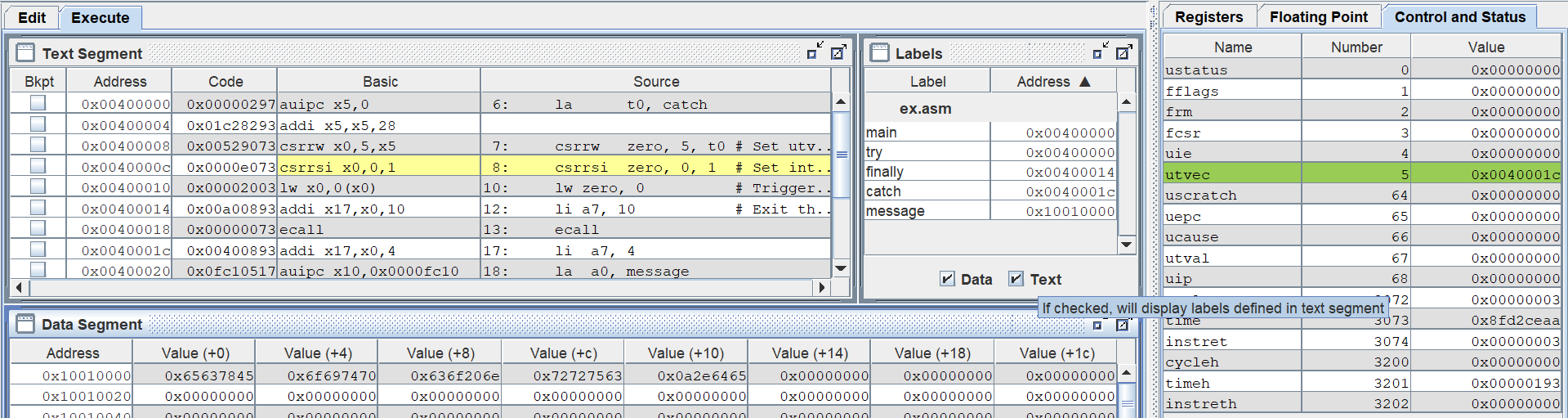
Then, a1 is increased by 1000 and stored in the address 0xffff0020



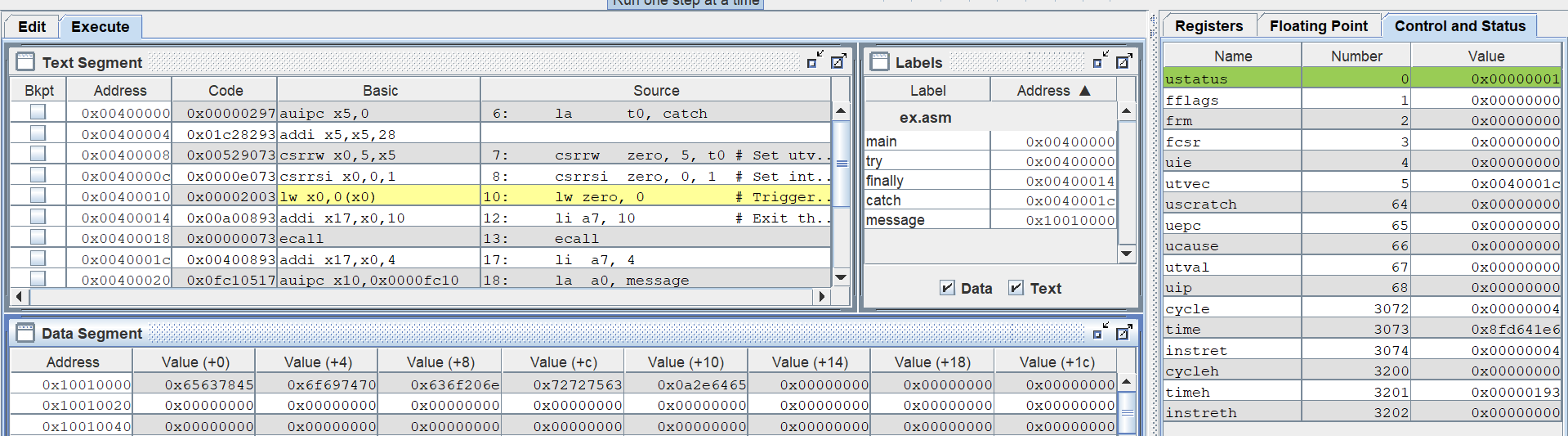
Finishing the process of handler routine and come back to the main program to continue to execute the next instruction right after the interrupt having occurred.

**Assignment 5: Create a new project, type in, and build the program of Home Assignment 5. Run the program step by step to understand each line of the source code.**

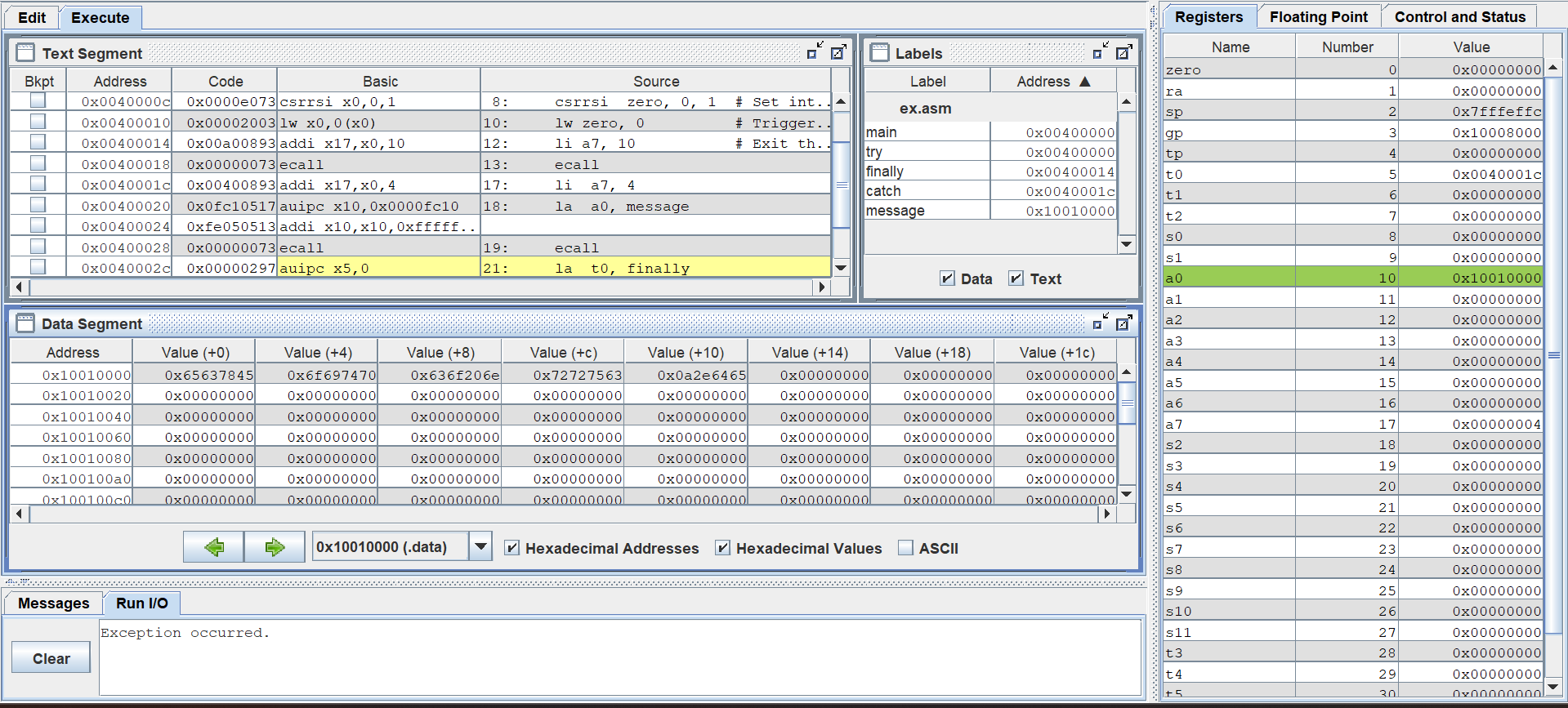
Load the interrupt handling routine address into the utvec register through register t0.

****

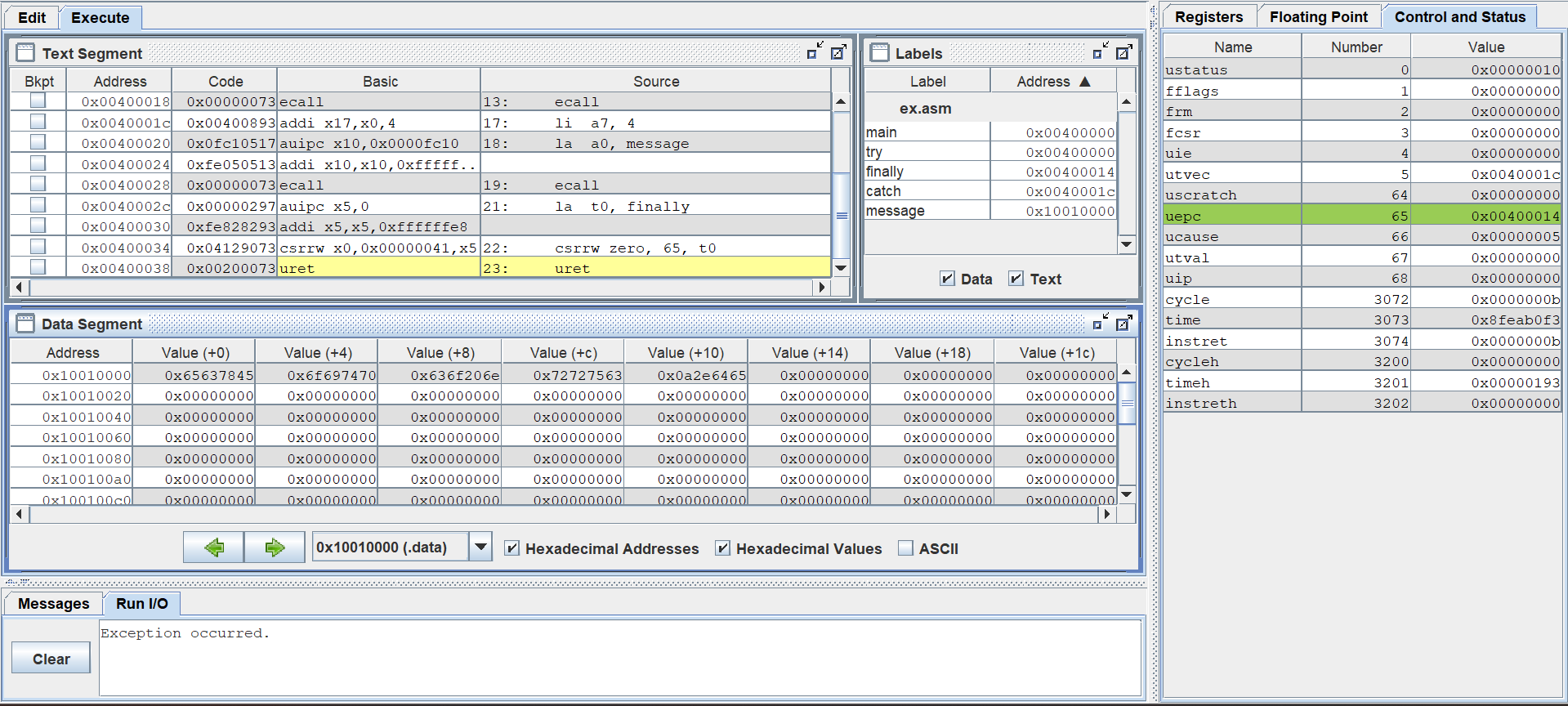
Set interrupt enable bit in ustatus (bit 0) to 1



In this case, instruction “lw zero, 0” is load access fault, so program jump to handling routine to handle this exception. Then, this routine prints the message for the interrupt.



Load finally label address to uepc register through register t0 in order to after handling the exception, the program executes the finally label and finishes running.



A screenshot of a computer

Description automatically generated

**Assignment 6: Software interrupts can be enabled by setting the USIP bit in the mip register. Write a program that enables a software interrupt when an overflow occurs while adding two signed integers (Lab 4). The interrupt routine will display a message in the console and terminate the program.**