

## ESD Lab Assignment No.1

Name: Mukund Vishwas Chavan

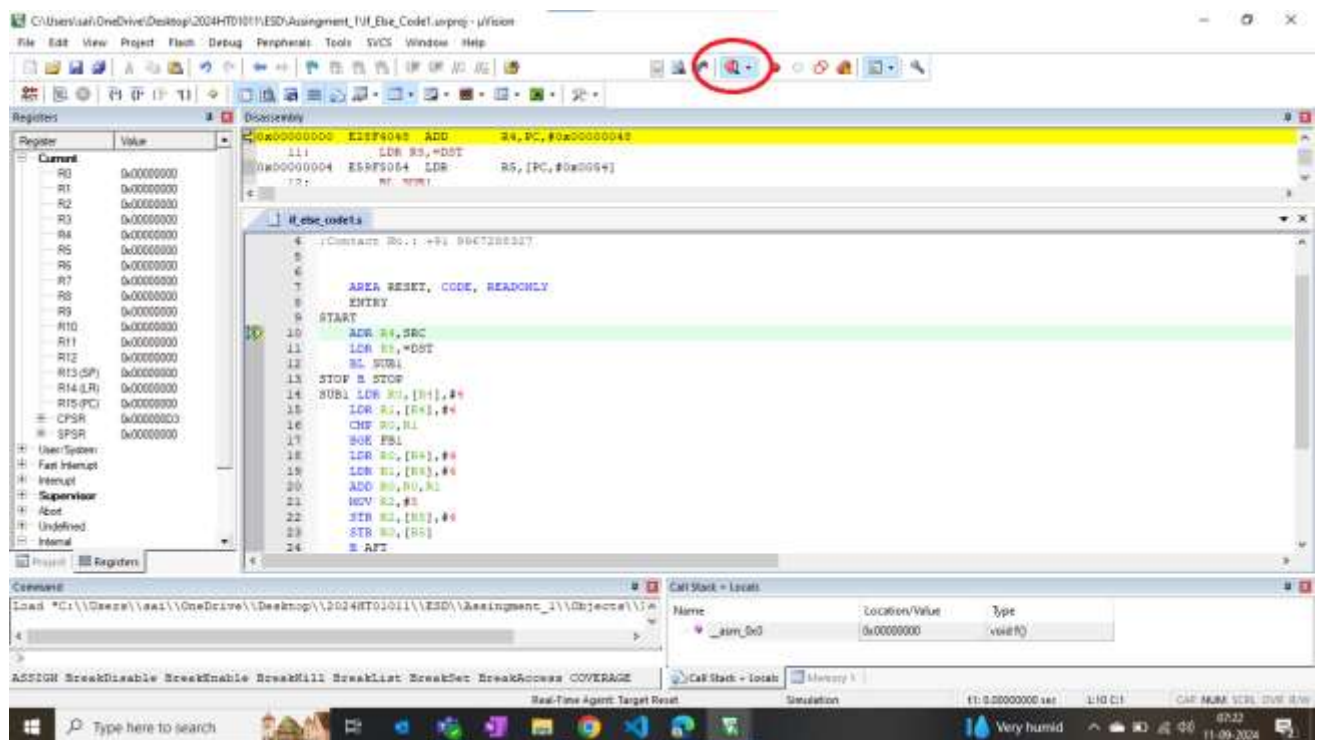
Bits-ID: 2024HT01011

Mail: 2024ht01011@wilp.bits-pilani.ac.in

Contact No.: +919967288327

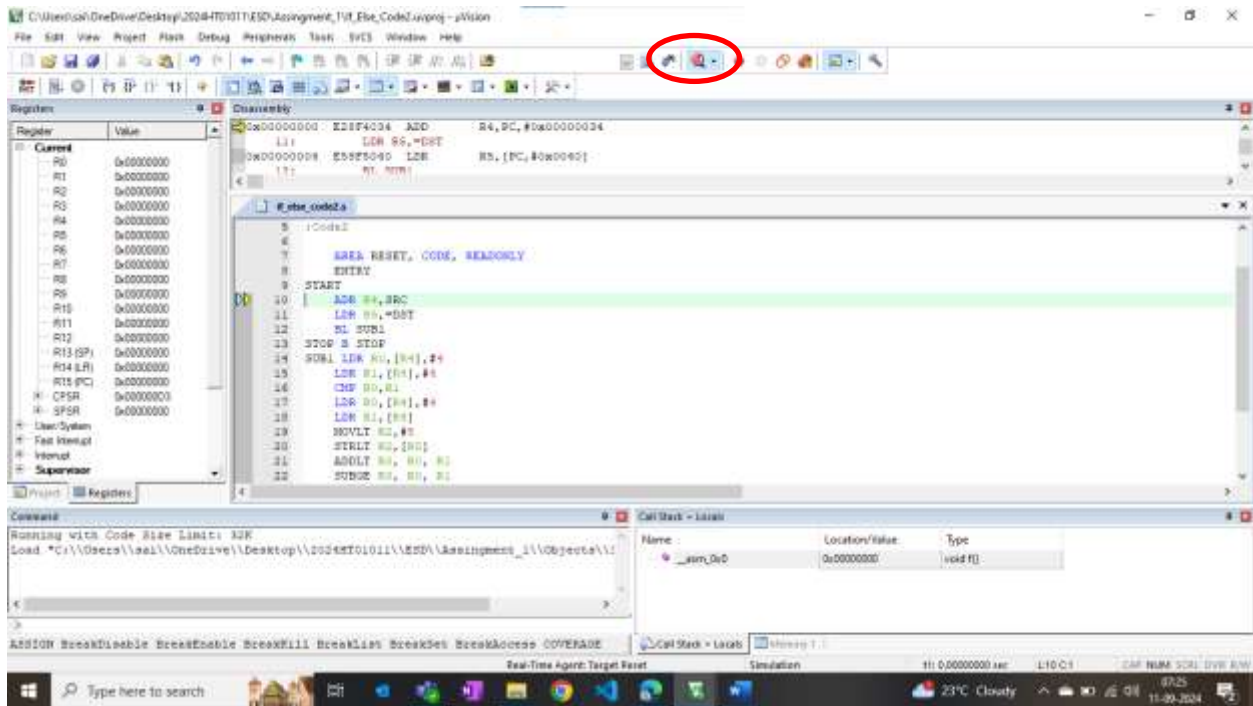
Question No 1: Assembly Language Programs (ALP) for an ARM7TDMI processor to implement following IF-ELSE statement?

Now in Debug mode code 1:

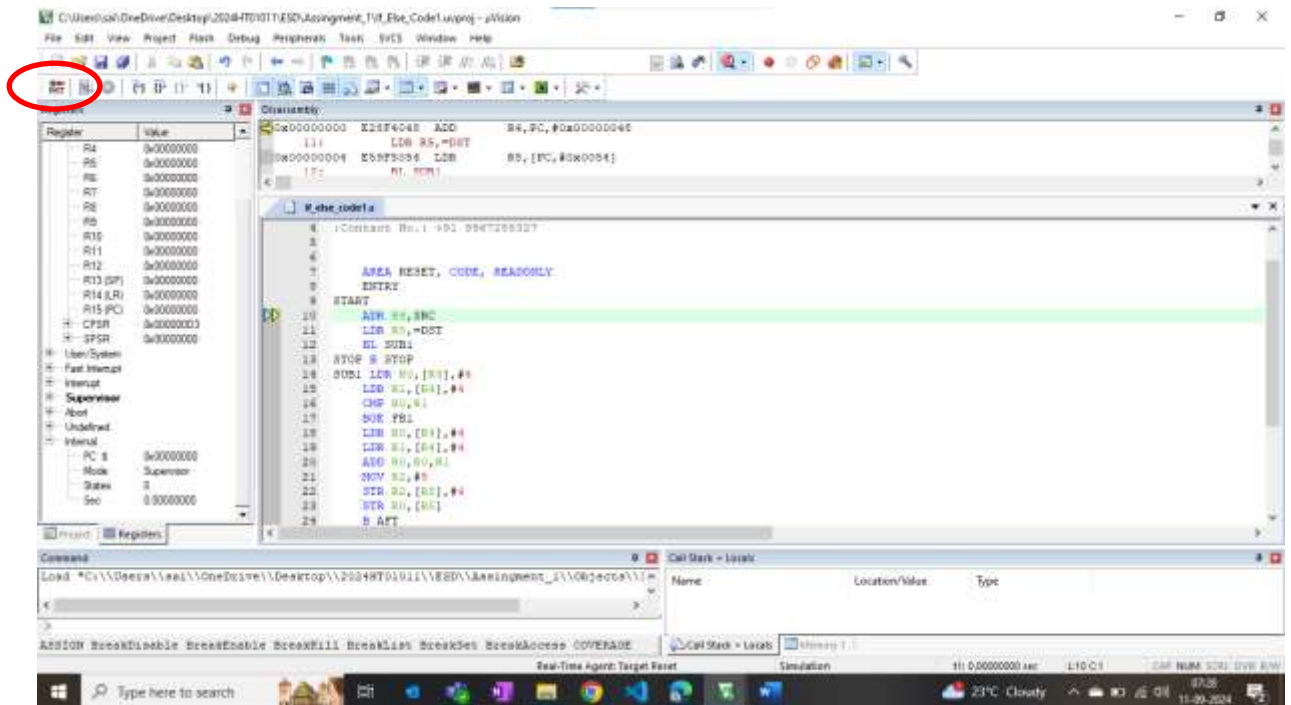


## ESD Lab Assignment No.1

Now in Debug mode code 2:

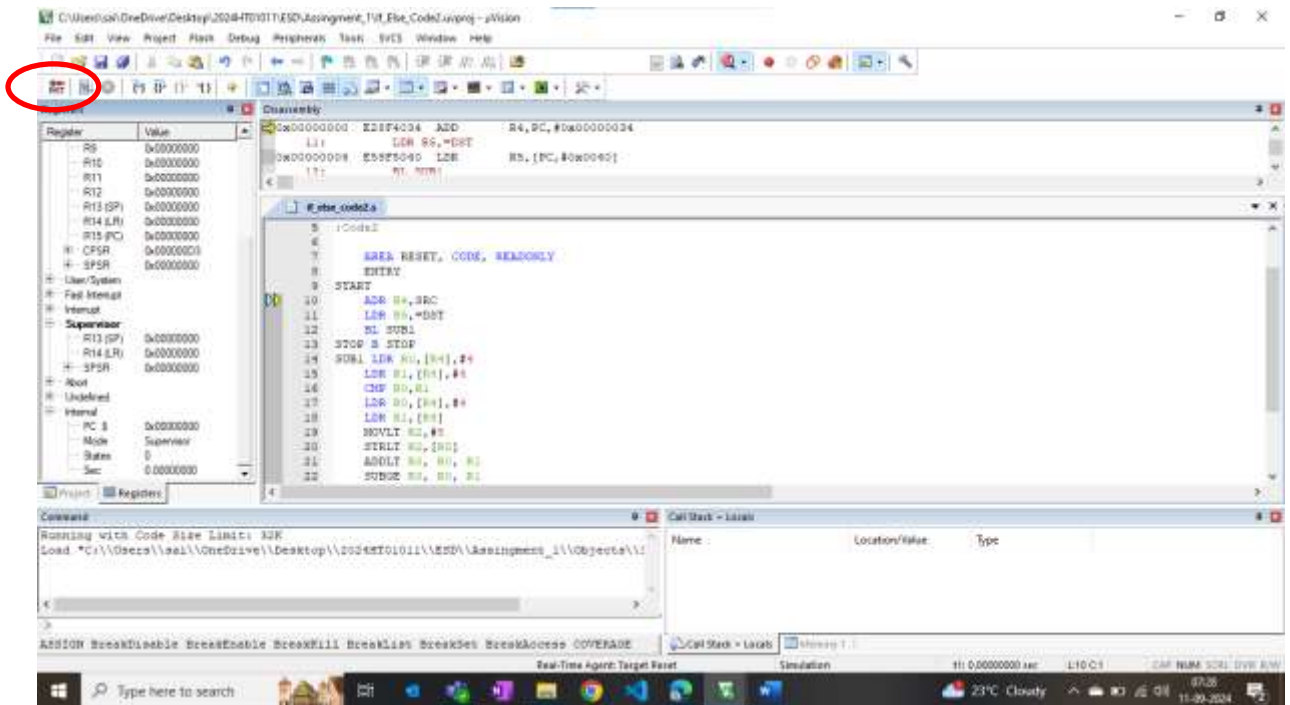


- a) On reset what is the ARM7TDMI processor's state and mode of operation?  
For Code 1 Processor's states = 0 >>



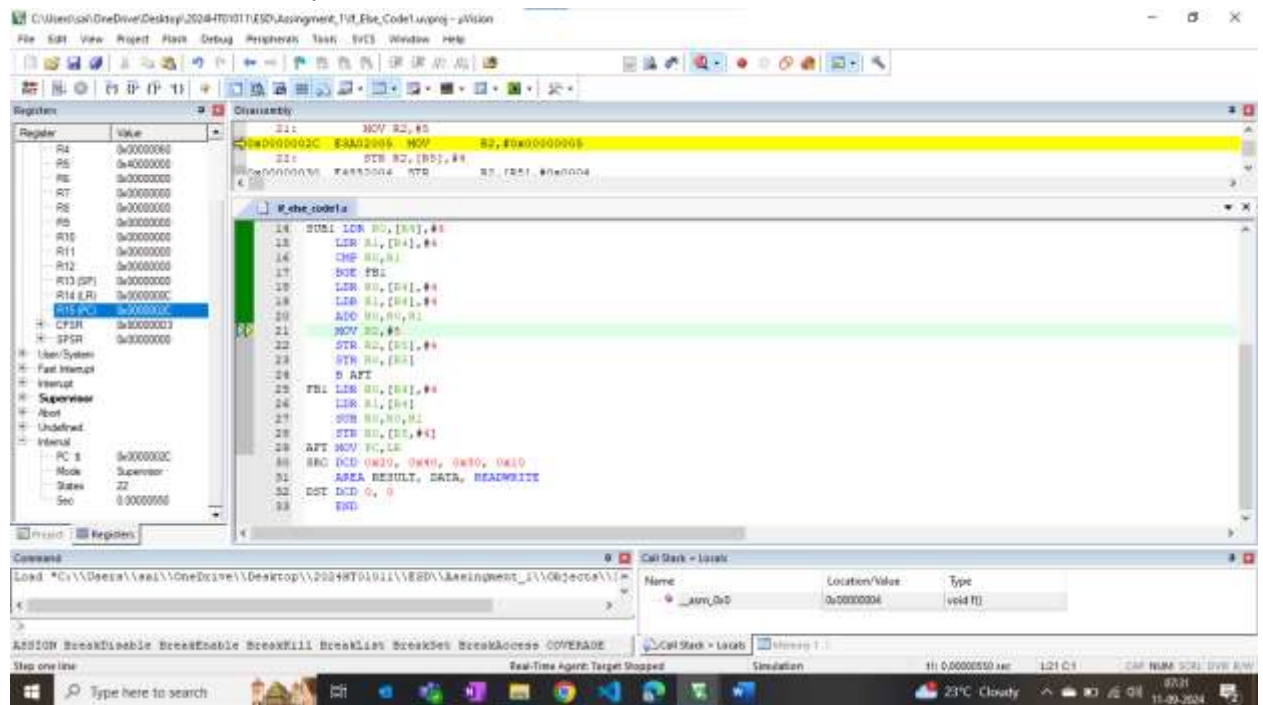
## ESD Lab Assignment No.1

For Code 2 Processor's states = 0 >>



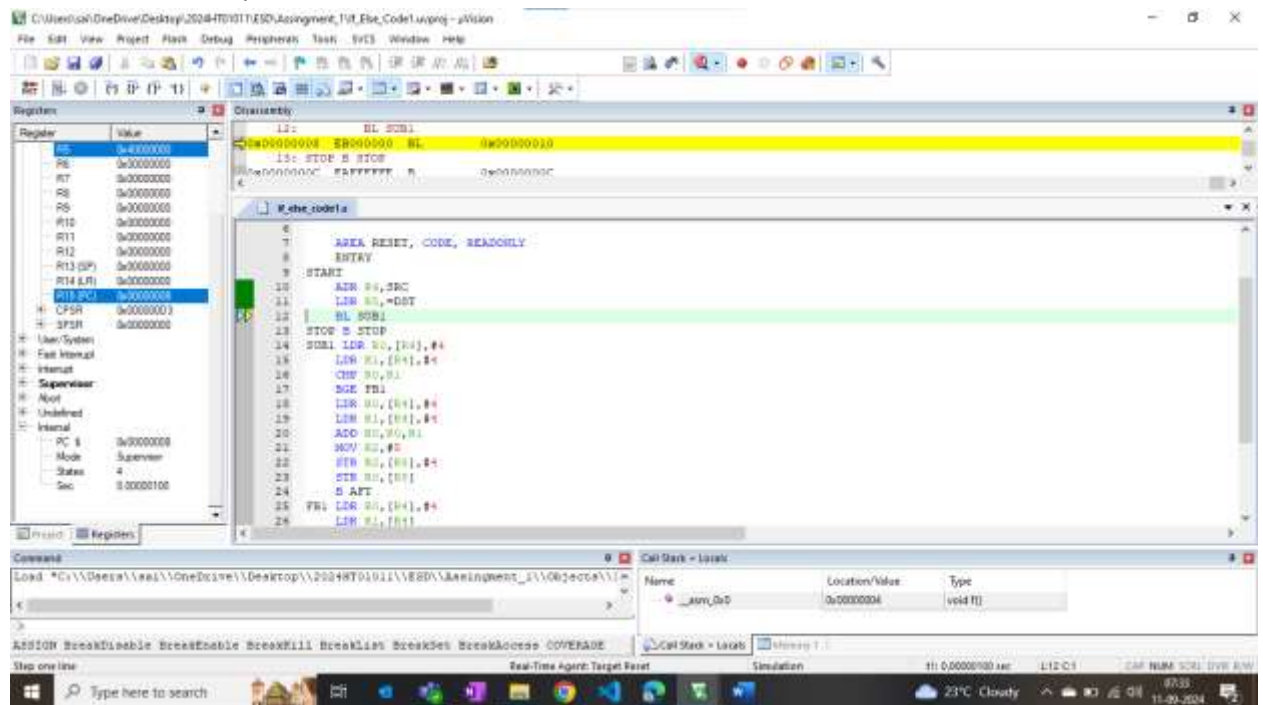
- b) How many states are taken for the execution of an Arithmetic instruction, Load and Store instruction respectively (For Code-1)

Arithmetic instruction required one state

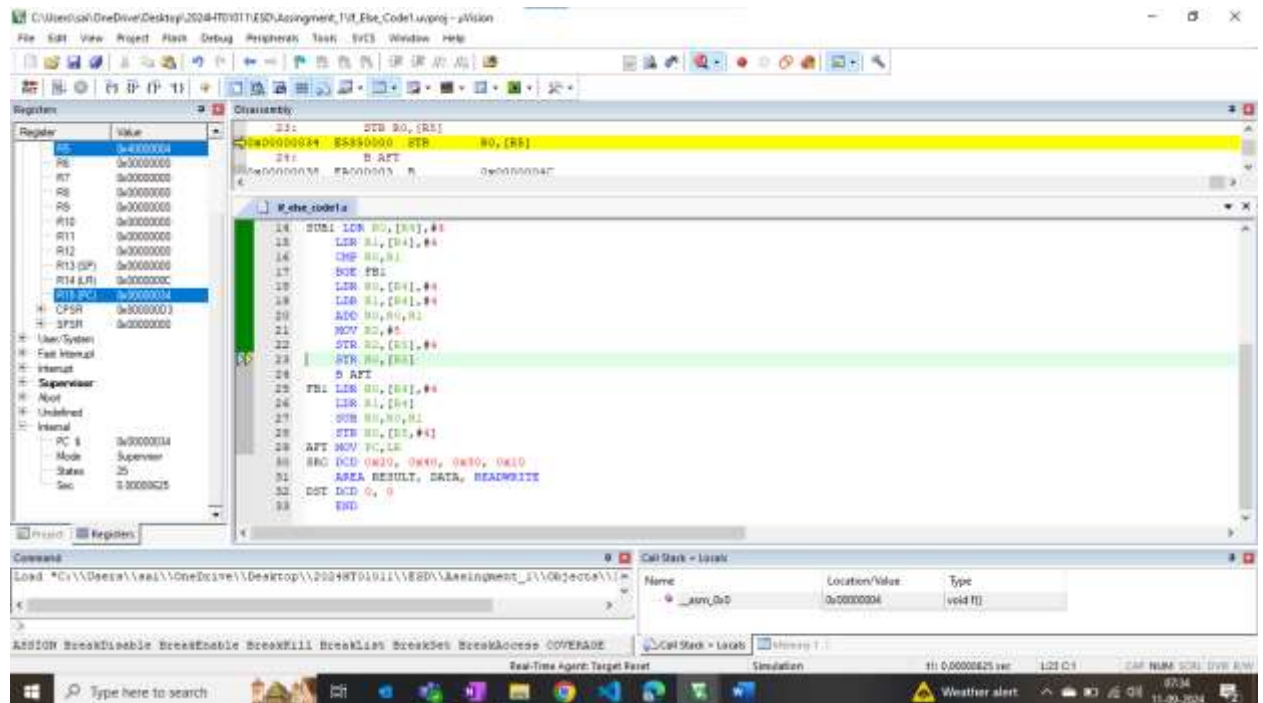


## ESD Lab Assignment No.1

Load instruction required 3 states to execute



Store instruction required 2 states

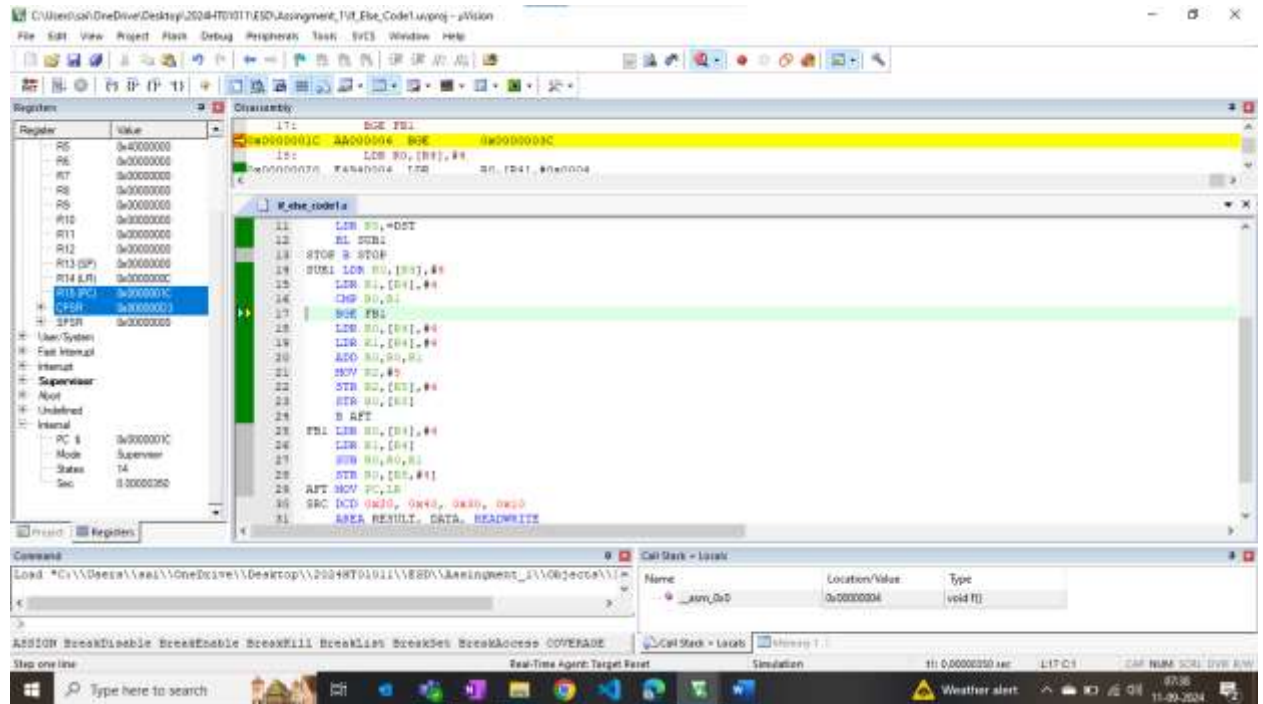




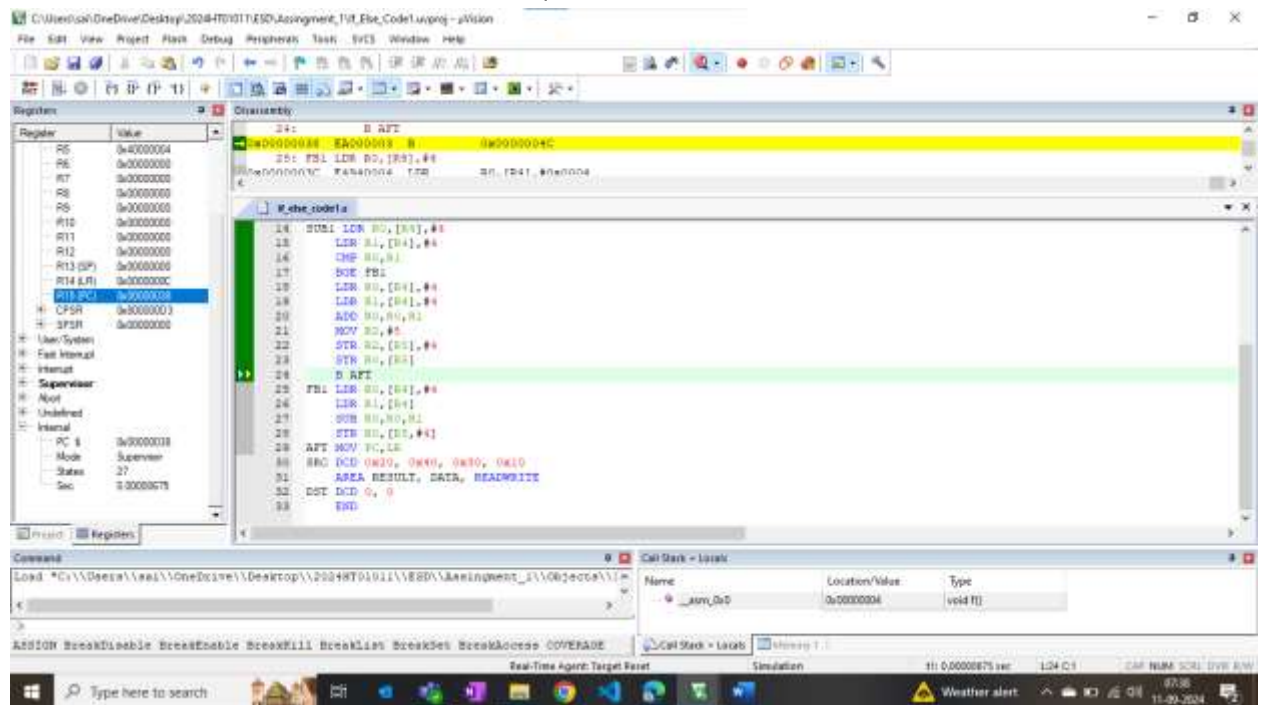
## ESD Lab Assignment No.1

- c) Are the number of states taken for completion the same for BGE instruction if the branch – (1) is taken (2) not taken? Please give the states are taken for each. (For Code-1)

branch – (1) is not taken when R0 value less than R1 (R0=20 and R1=40). Before branch BGE call states =14

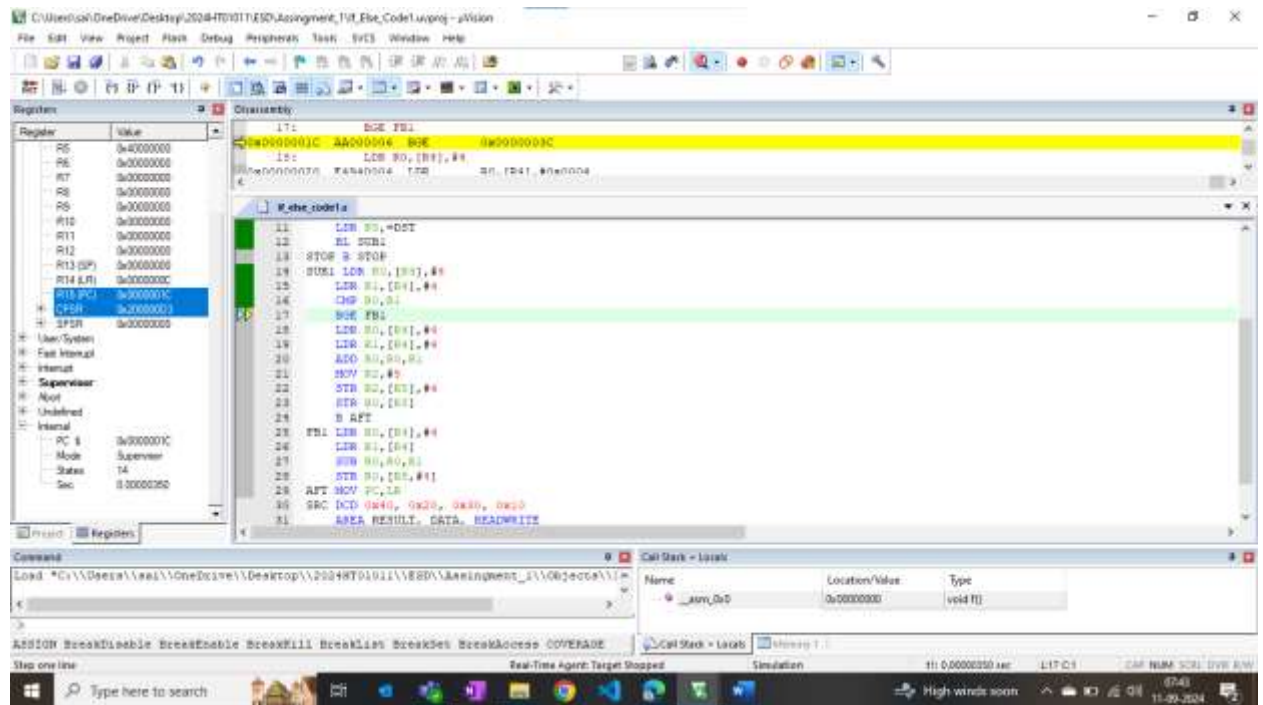


After branch BGE states =27 So the States required = 15

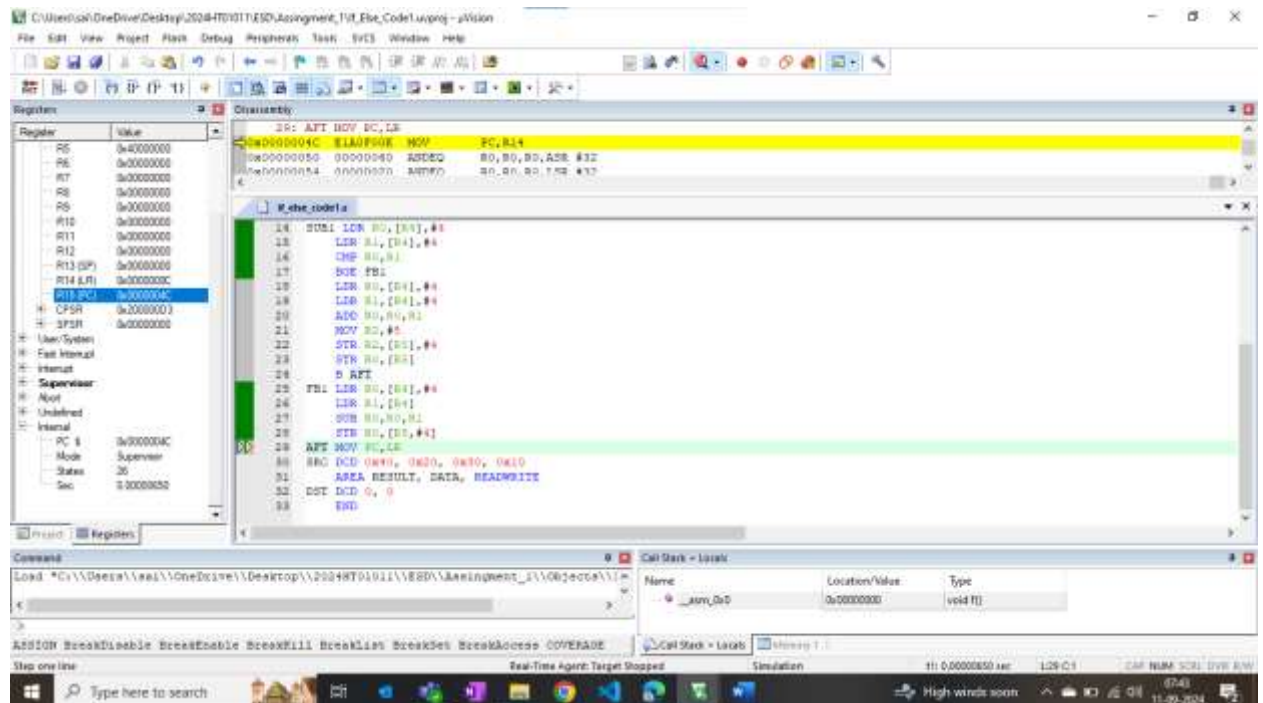


## ESD Lab Assignment No.1

Branch – (2) is taken where R0 value is greater than equal to than R1 (R0=40 and R1=20). Before branch BGE call states =14



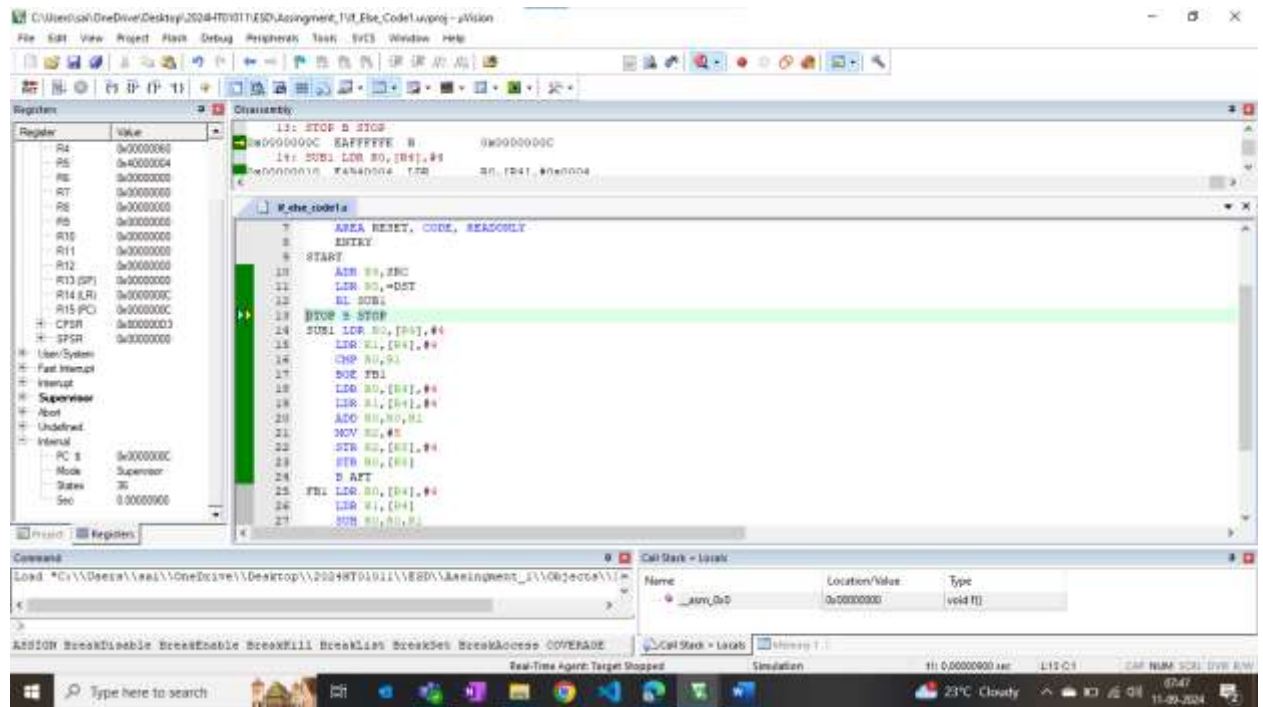
After branch BGE states =26 So the States required = 12



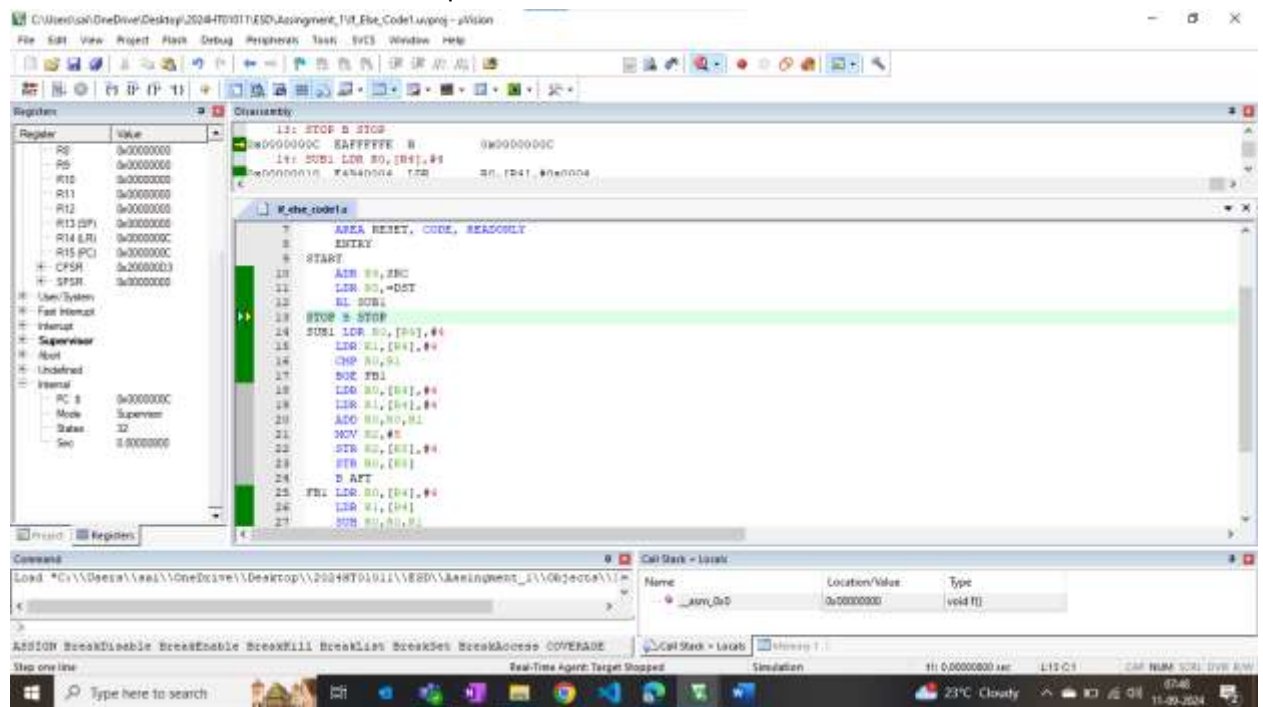
## ESD Lab Assignment No.1

- d) Measure the performance of code-1 and code-2 for the following conditions.

Code 1 : Condition 1 where  $a < b$  it required total 36 States

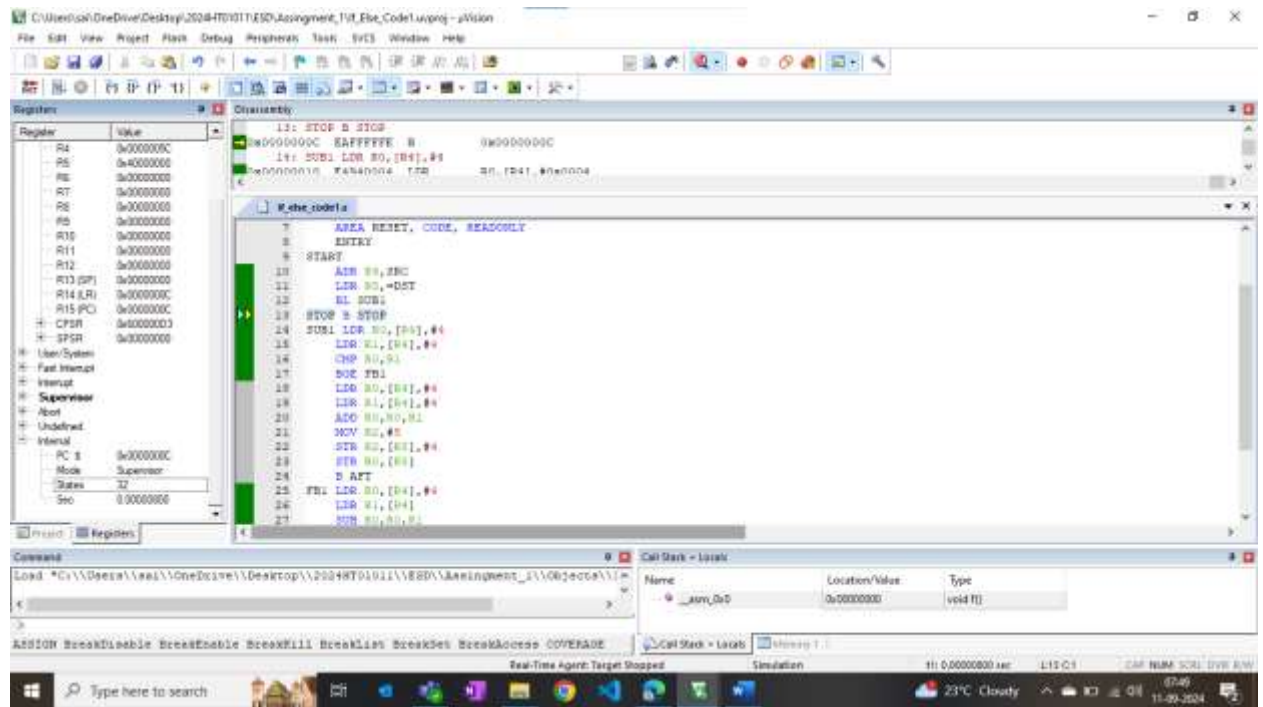


Code 1 : Condition 2 where  $a > b$  it required total 32 States

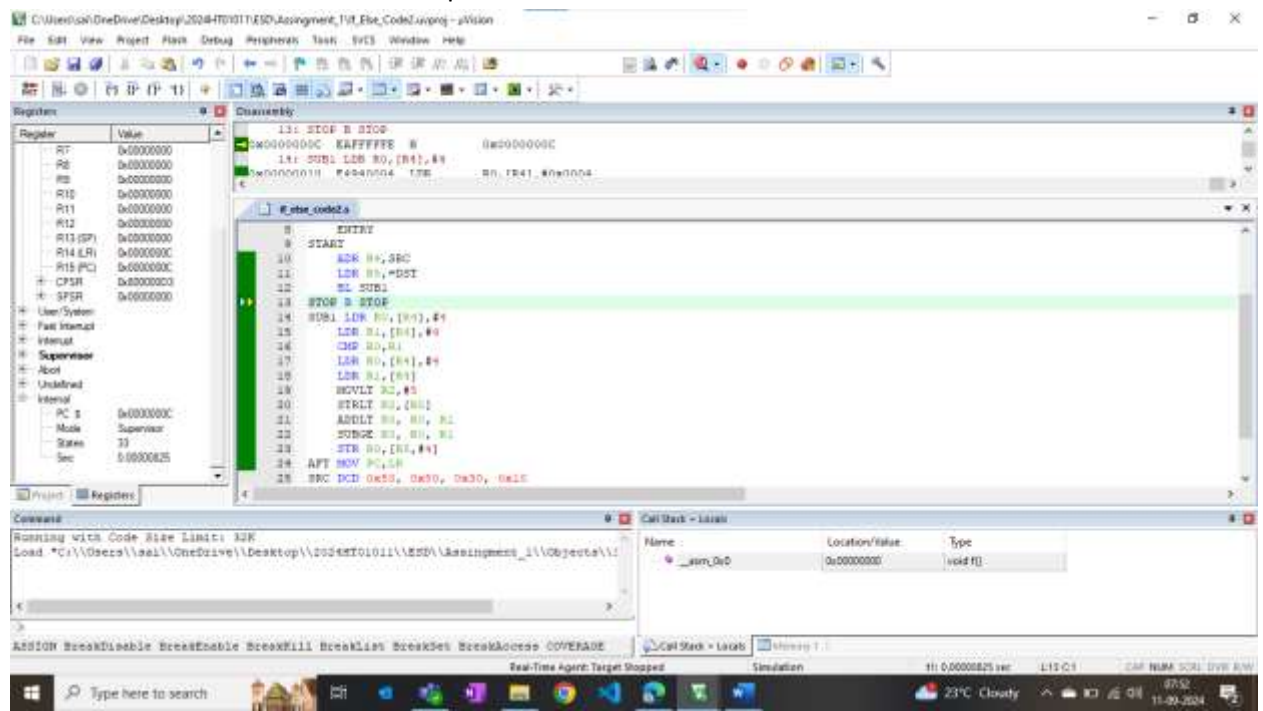


## ESD Lab Assignment No.1

Code 1 : Condition 3 where  $a=b$  it required total 32 States.



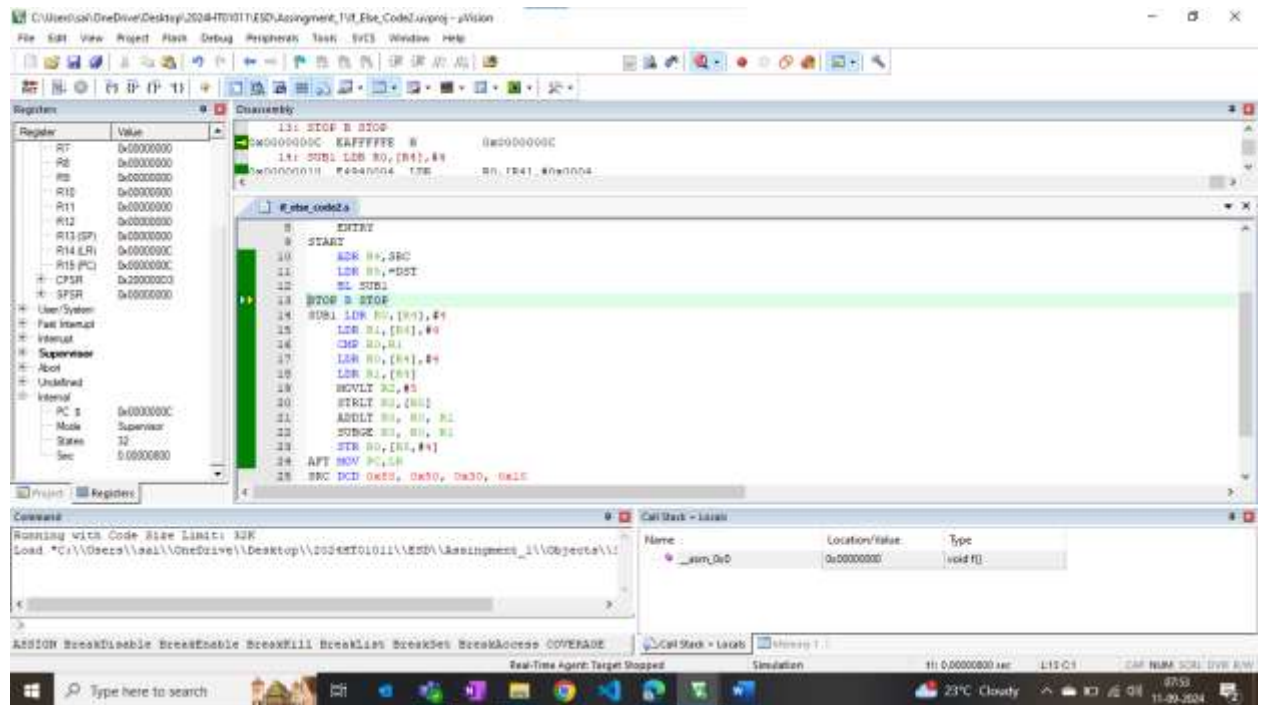
Code 2 : Condition 1 where  $a < b$  it required total 33 States



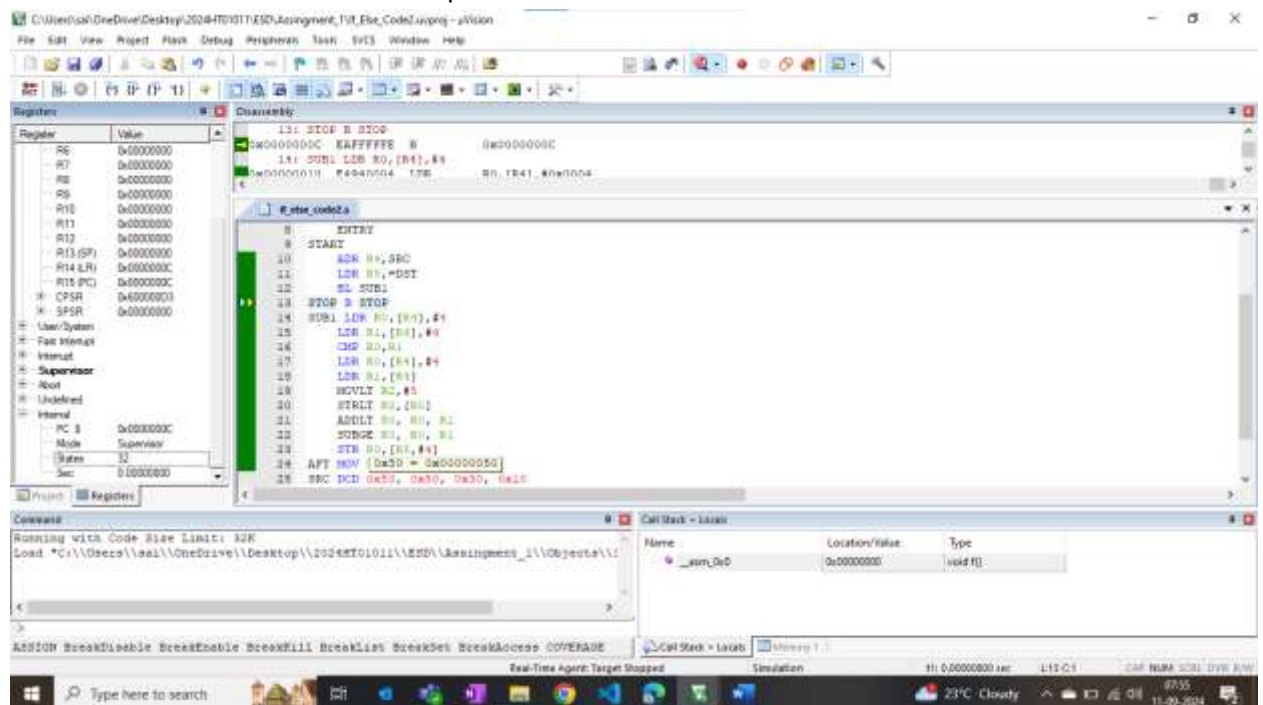


## ESD Lab Assignment No.1

Code 2 : Condition 2 where  $a > b$  it required total 32 States



Code 2 : Condition 3 where  $a = b$  it required total 32 States.

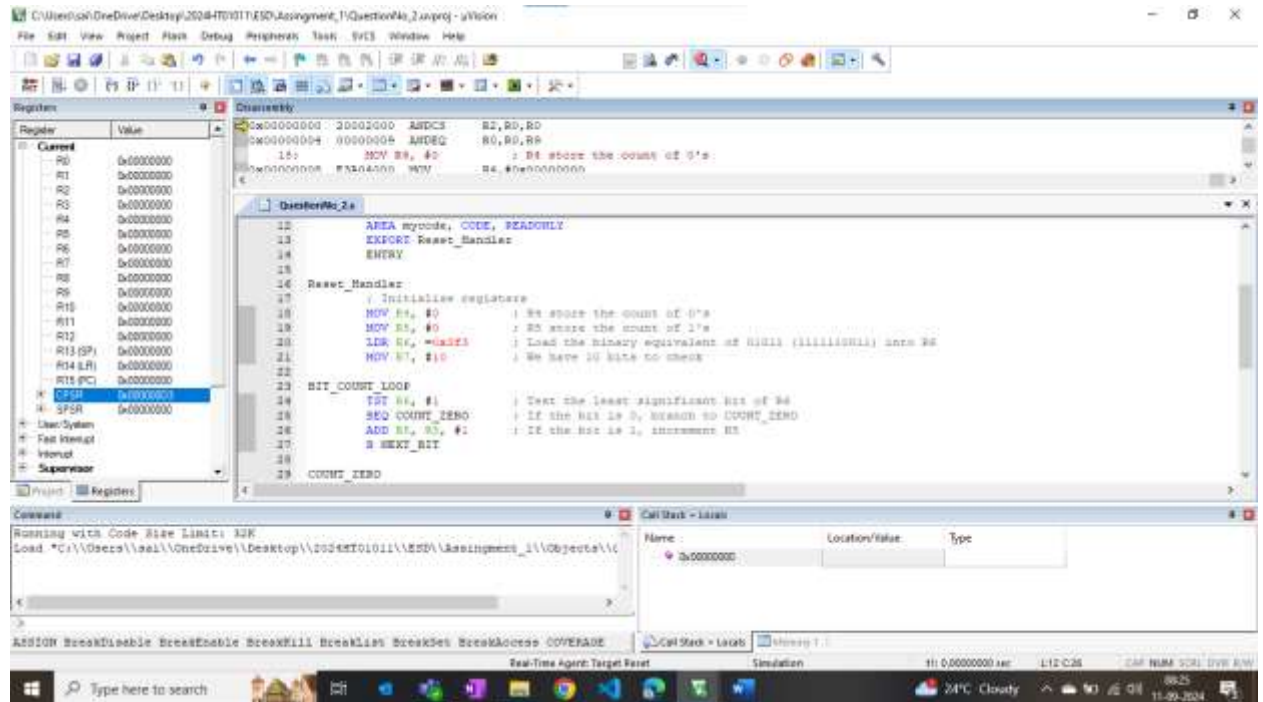


## ESD Lab Assignment No.1

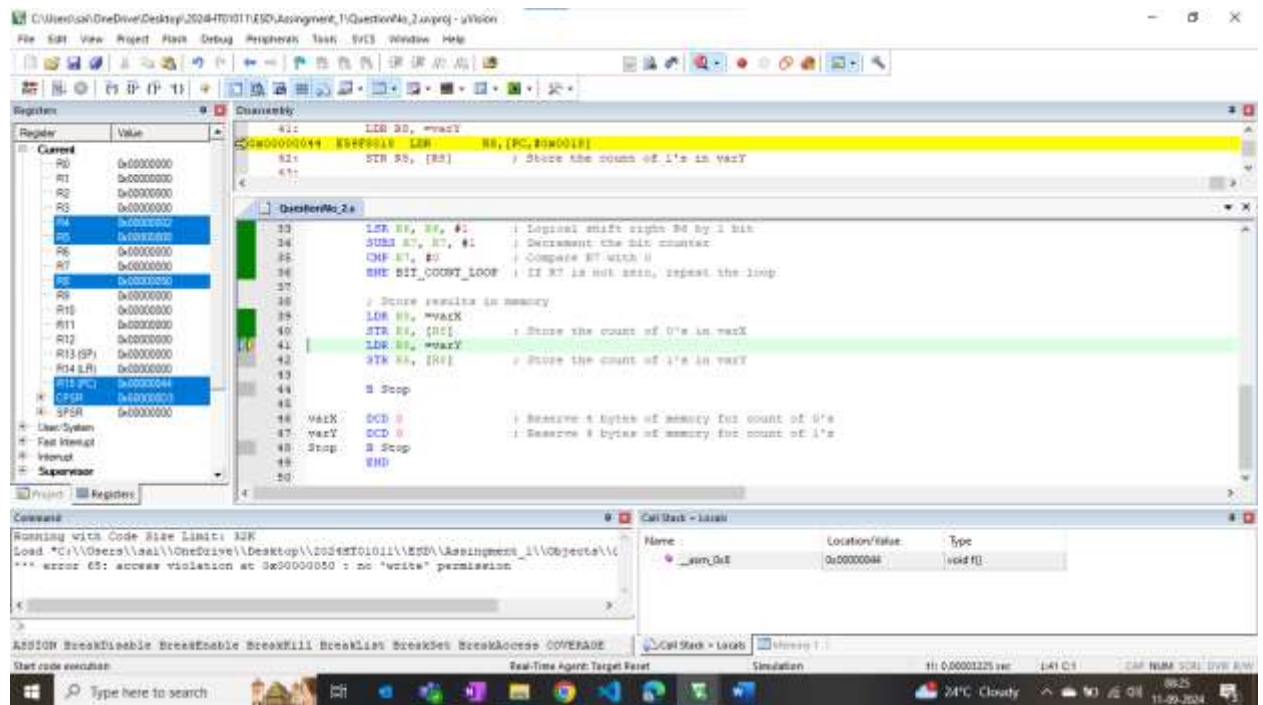
Question No 2: Write an assembly language program for ARM7TDMI to count the number of 0's and 1's in the last 5 digits of your BITS ID (number must be given in decimal number format). Store the number of 0's in register R0 and the number of 1's in register R1. Verify your result by performing manual calculation?

My BITS ID is 01011

Here initially R4 and R5 values are 0.



After execution the number of zero's in R4 are 3 and number of 1's in R5 are 8.

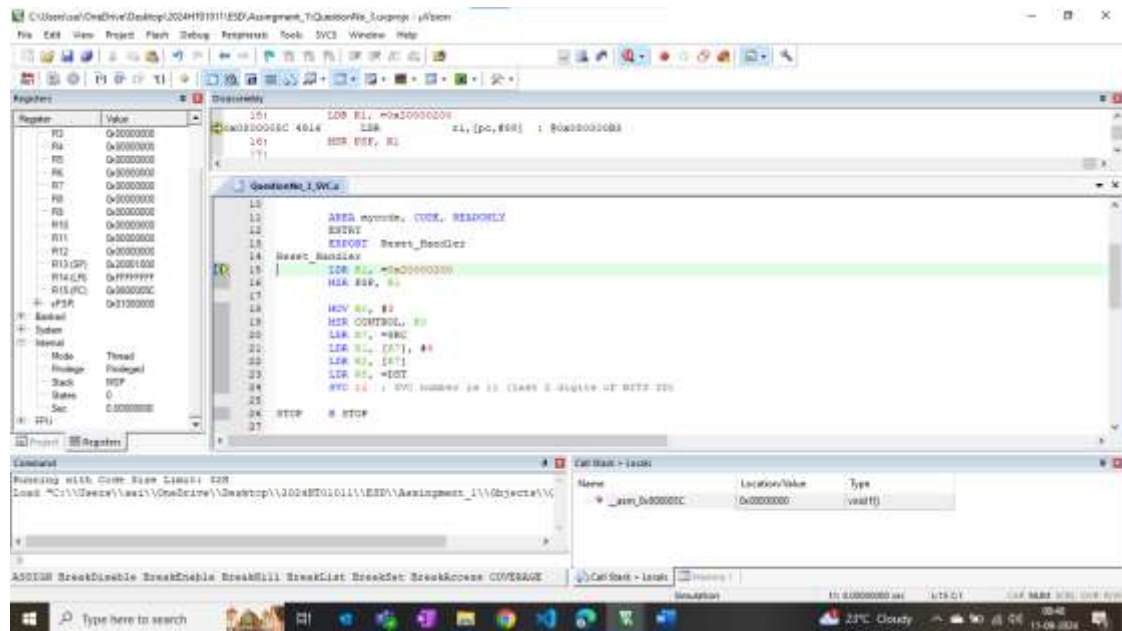


## ESD Lab Assignment No.1

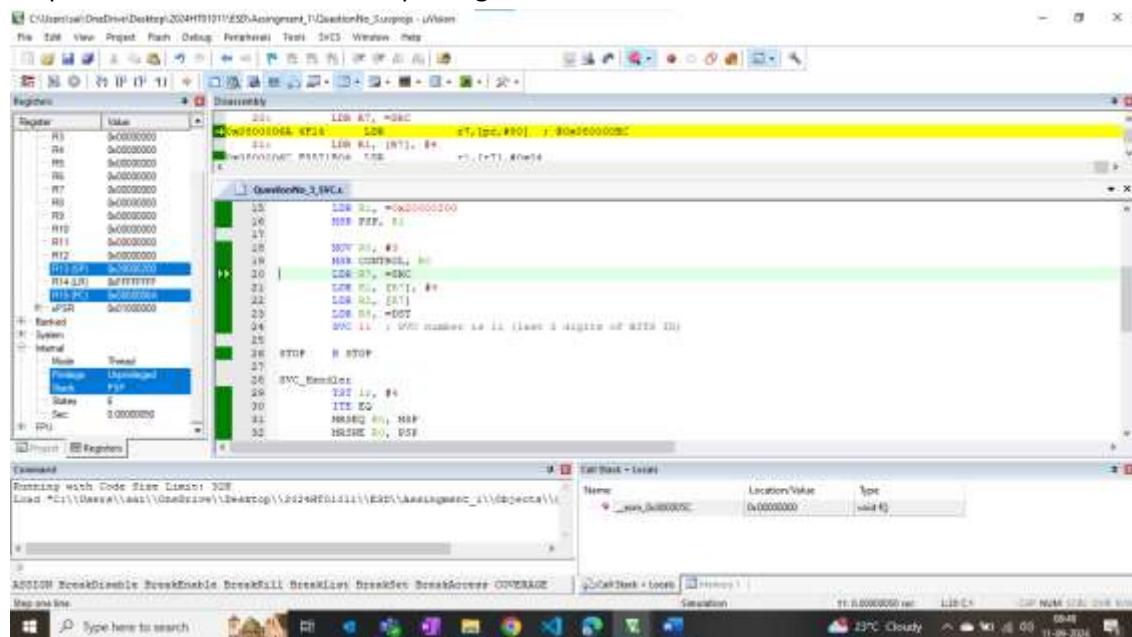
Here I have verified count values using Manual Calculation >> Binary representation of 01011 : 1111110011 and here Number of 0's: 2 Number of 1's: 3

Question No 3: Write an assembly language program for ARM Cortex M3/4 to handle Supervisor Call (SVC) exception. The program should meet the following requirements.

First Step: My BITS ID's last three digits are 011 so I'm taking SVC value as 11  
My Initially Thread was Privileged



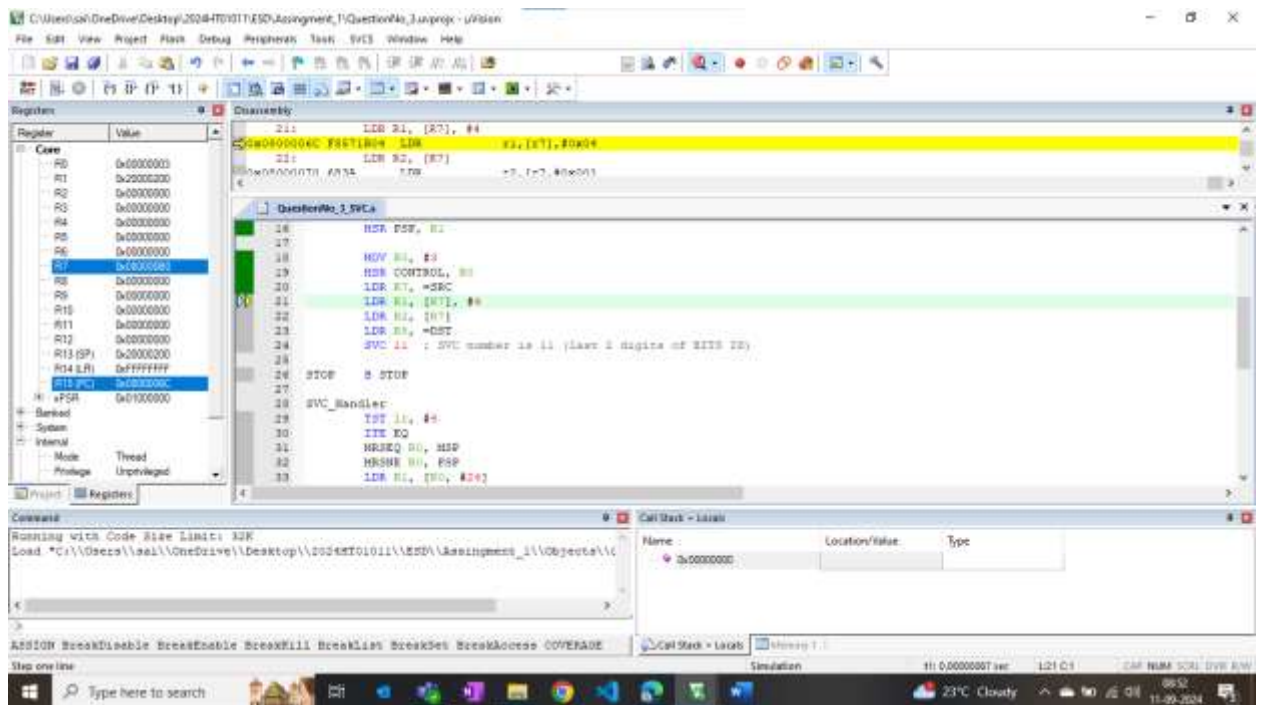
### Requirement 1: SVC Call from Unprivileged Mode



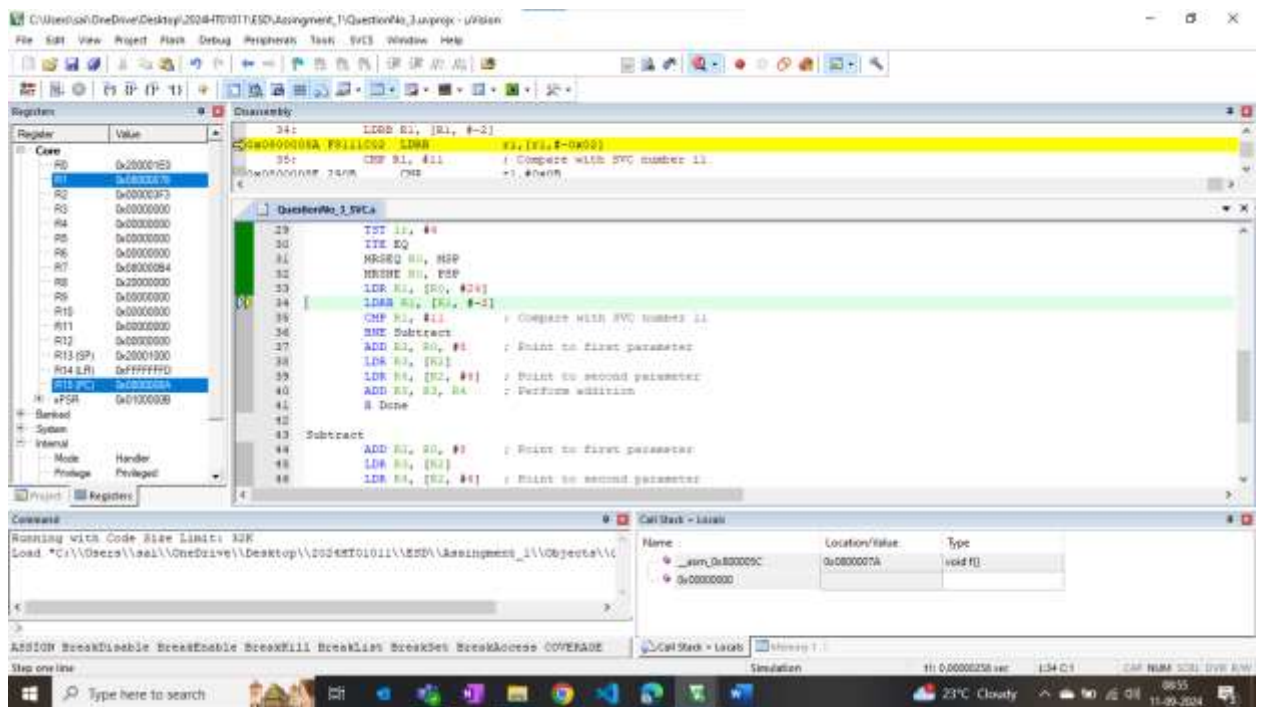
### Requirement 2: Parameters Passed to SVC Handler:

## ESD Lab Assignment No.1

LDR R7, =SRC: Loads the address of SRC into R7



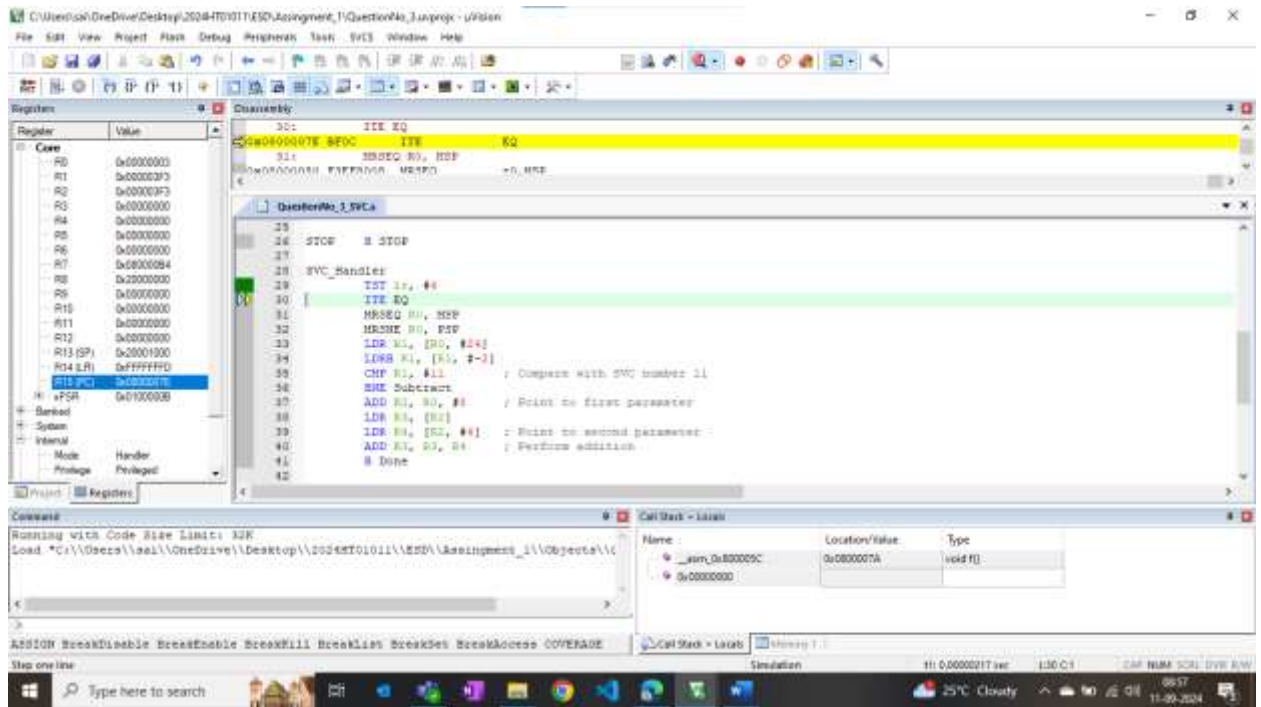
Requirement 3: Dynamic SVC Number Determination: LDR R1, [R0, #24]: Loads the stacked PC value into R1



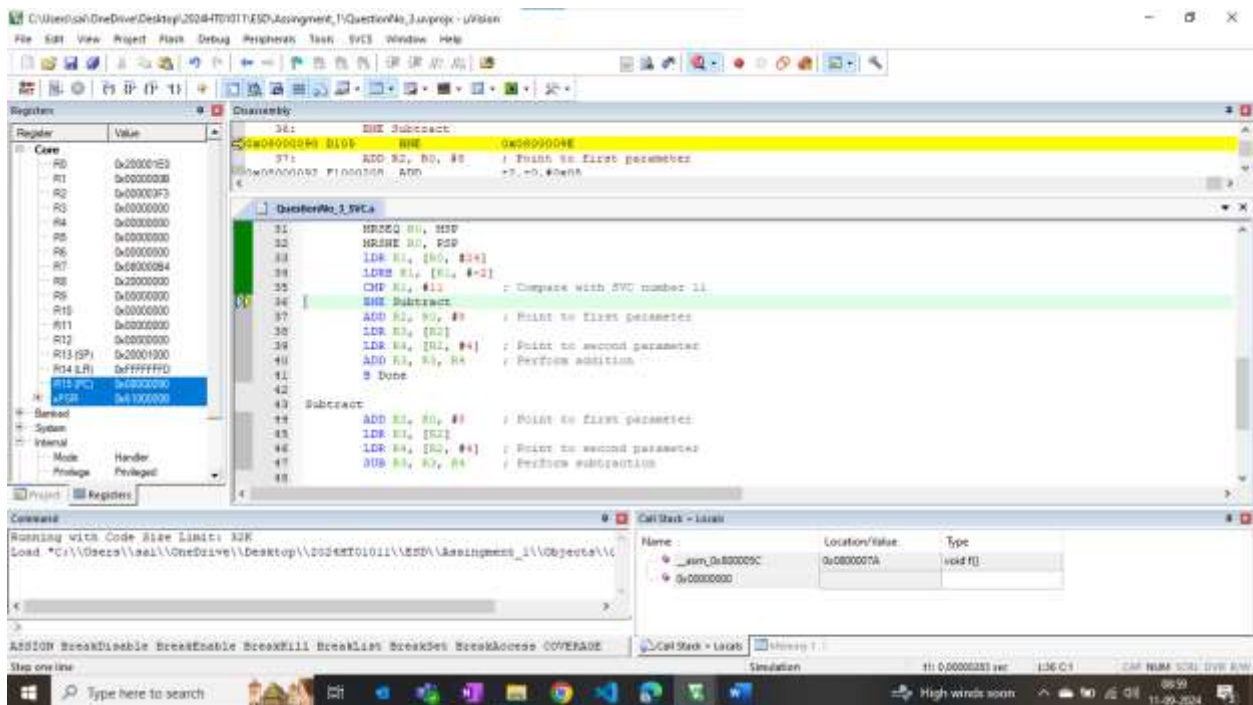
#### Requirement 4: Setting Up Stack Pointer in SVC Mode



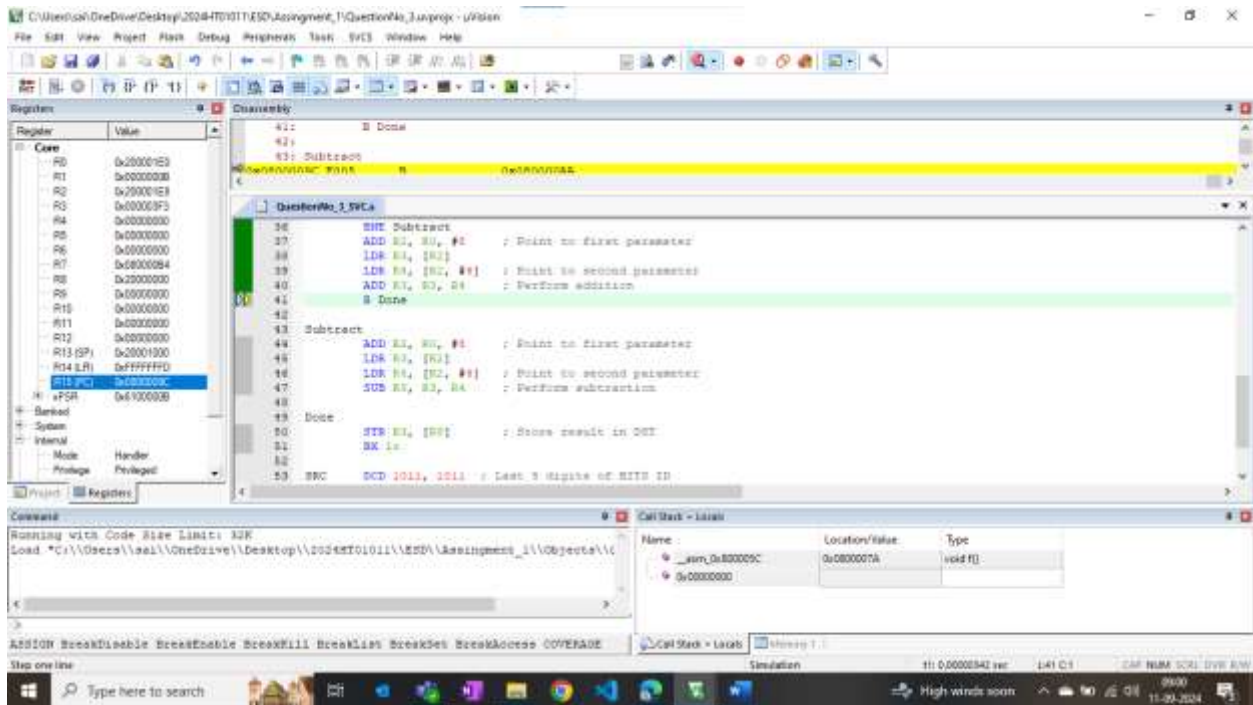
## ESD Lab Assignment No.1



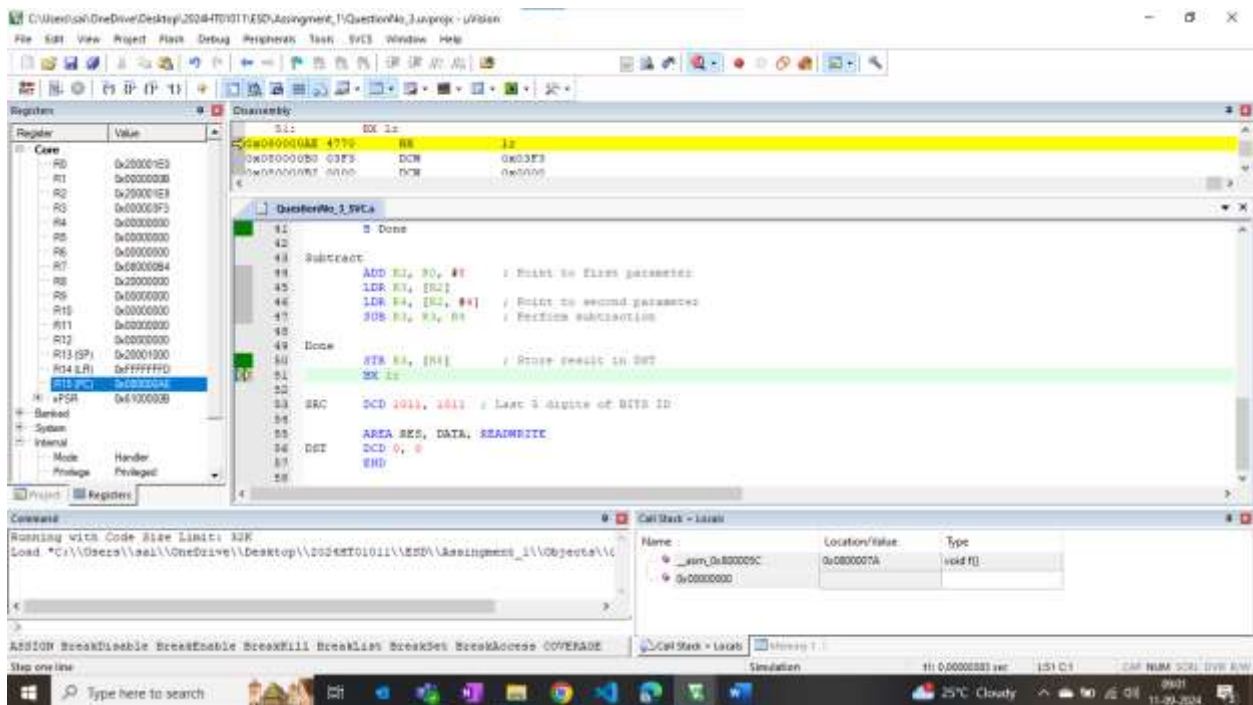
Requirement 5: Operation Based on SVC Number: CMP R1, #11: Compares the SVC number with 11



## ESD Lab Assignment No.1

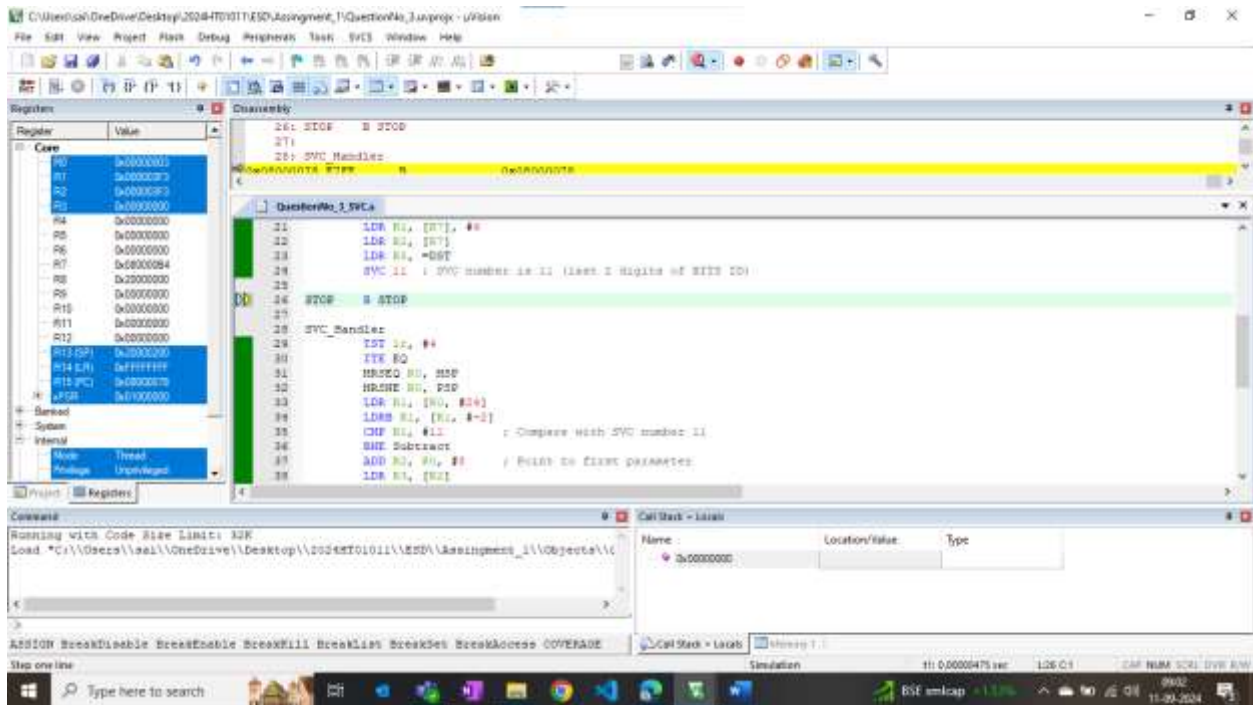


Requirement 6: Handling Stack Usage and Resuming Task: The result of the operation is stored in DST. The handler restores the context and returns from the exception using BX lr. STR R3, [R8]: Stores the result in DST

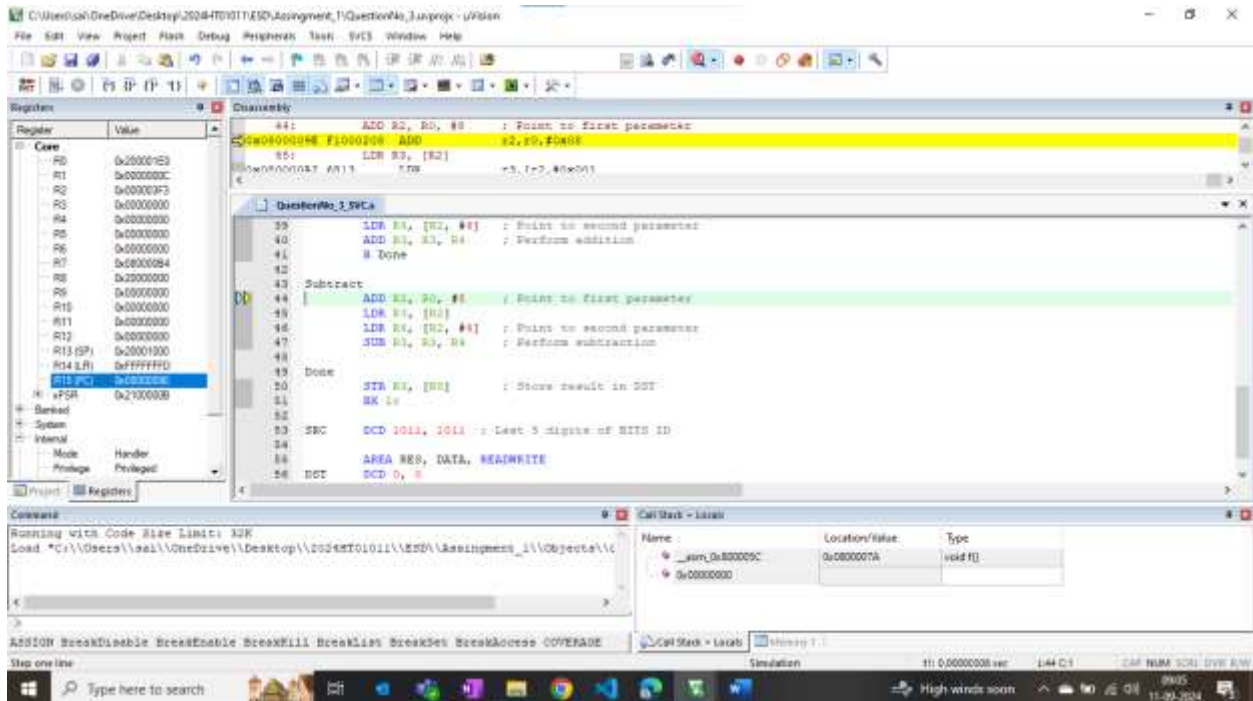


BX lr: Returns from the exception, STOP B STOP: Infinite loop to keep the program running >>

## ESD Lab Assignment No.1



Second Step: Rebuild the code and debug by changing SVC value to 12. SVC Changed to 12



## ESD Lab Assignment No.1

