By: Abdelrahman matarawy

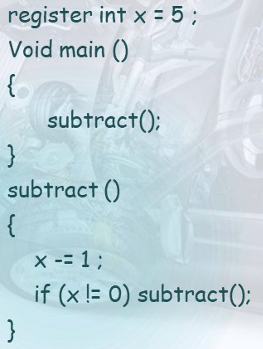
**Lecture2**

**Inline Assembly**

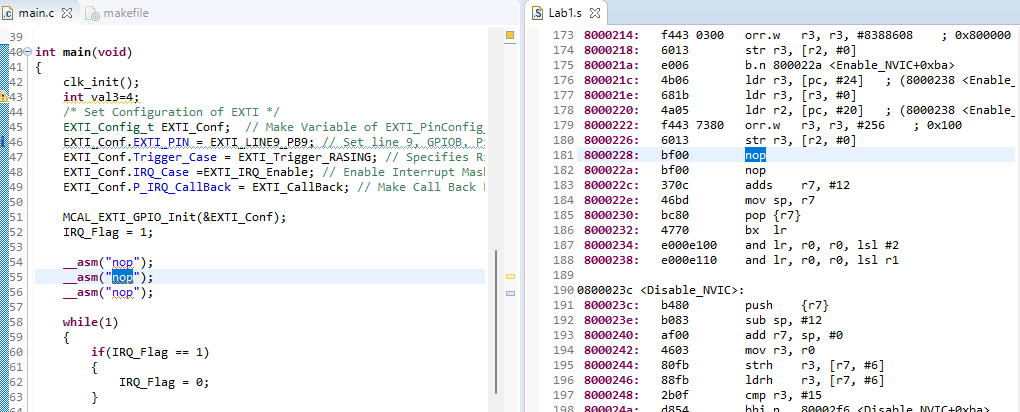
**ARM Assembly Basics**

* **Lab to achieve assembly to equivalent C code:**

**A screenshot of a computer

Description automatically generated**

* **INLINE Code:**
  + INLINE Function:
    - Making the Function Inline eliminates the BL and BX instructions and their execution but replace function code everywhere BL call which saving time at the possible expense of memory.
  + Not inline function:
    - Calling a regular function requires time to execute the BL and BX instructions but for large functions it saves memory at the expanse of speed.
* **Using Assembly in C source: (\_\_asm)**
  + Lab1:
    - Simple lab without input or output parameters.

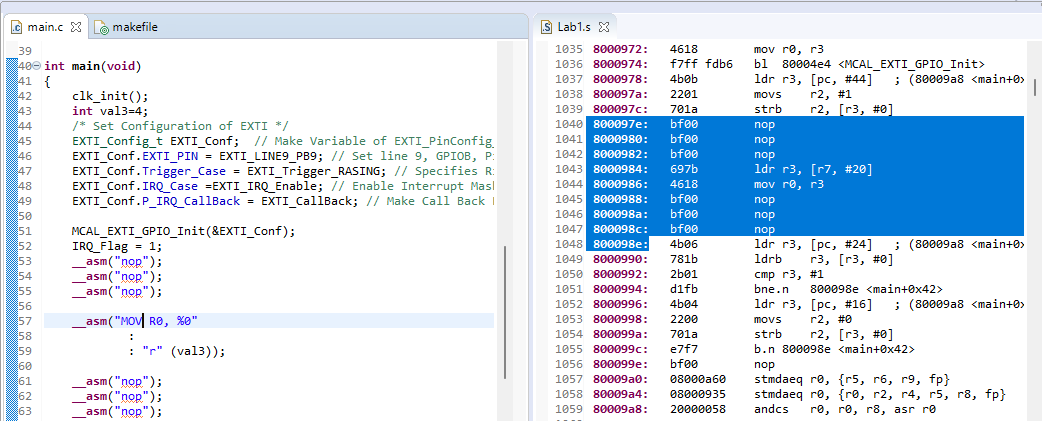


* + Lab2: (using output parameters)
    - The %0 in the instruction template is replaced with the name of this register.

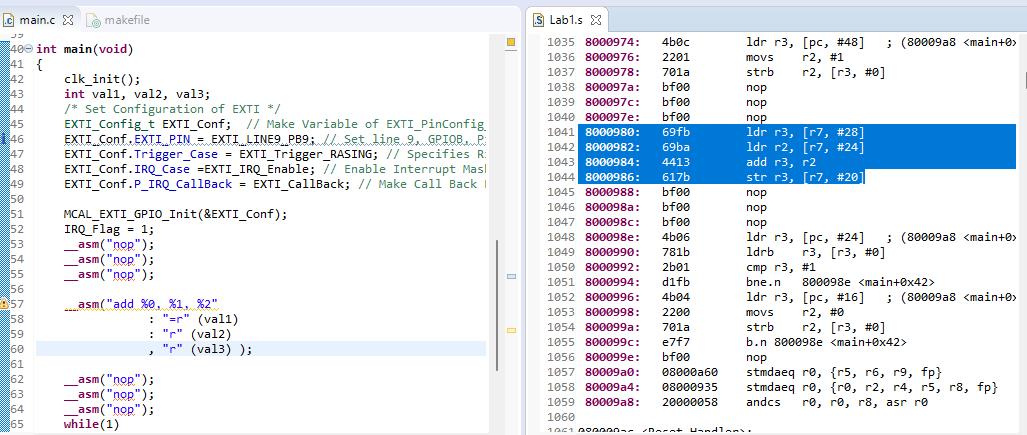
A screenshot of a computer

Description automatically generated

* + Lab3: (using output parameters)
    - [r7, #20] -> r7 is base of address and #20 is offset
    - the value in this address save in r3
    - “r” -> specifier for read and write



* + Lab4:



* + Lab6:

