Architetture dei Sistemi di Elaborazione	Delivery date: 12 November 2024
Laboratory 6	Expected delivery of lab_06.zip must include: - Solutions of the exercises 1, 2, 3 and 4 - this document compiled possibly in pdf format.

Starting from the ASM_template project (available on Portale della Didattica), solve the following exercises.

1) Write a program using the ARM assembly that performs the following operations:
   a. Initialize registers *R1*, *R2*, and *R3* to random signed values.
   b. Subtract *R2* to *R1* (*R2* – *R1*) and store the result in *R4*.
   c. Sum *R2* to *R3* (*R2* + *R3*) and store the result in *R5*.

   Using the debug log window, change the values of the written program in order to set the following flags to 1, one at a time and when possible:
   - carry
   - overflow
   - negative
   - zero

   Report the selected values in the table below:

	Hexadecimal representation of the obtained values			
Updated flag	R2 – R1		R2 + R3	
	R2	R1	R2	R3
Carry = 1	1	-1	1	1
Carry = 0	2	-1	2	1
Overflow	0x7FFFFFFF	0xFFFFFFFF	0x7FFFFFFF	0x1
Negative	1	2	1	-2
Zero	1	1	1	-1

Please explain the cases where it is **not** possible to force a **single** FLAG condition:
Il caso dell'overflow porta anche il flag negativo perchè viene generato l'8° bit a 1.

2) Write a program that performs the following operations:
   a. Initialize registers *R6* and *R7* to random signed values.
   b. Compare the two registers:
      - If they differ, store in register *R8* the maximum among *R6* and *R7*.

- Otherwise, perform a logical right shift of 1 on *R6* (is it equivalent to what?), then subtract this value from *R7* and store the result in *R4* (i.e., *R4* = *R7* – (*R6* >> 1)).

Considering a CPU clock frequency (clk) of *16 MHz*, report the number of clock cycles (cc) and the simulation time in milliseconds (ms) in the following table:

	R2 == R3 [cc]	R2 == R3 [ms]	R2 != R3 [cc]	R2 != R3 [ms]
Program 3	14.72	0.00092	12	0.00075

Note: you can change the CPU clock frequency by following the brief guide at the end of the document.

3) Write a program that calculates the leading zeros of a variable. Leading zeros are calculated by counting the zeros starting from the most significant bit and stopping at the first 1 encountered: for example, there are five leading zeros in *2_00000*101. The variable to be checked is in *R10*. After counting, if the number of leading zeros is odd, subtract *R11* from *R12*. If the number of leading zeros is even, add *R11* to *R12*. In both cases, the result is placed in *R9*.

   Implement ASM code that does the following:
   a. Determine whether the number of leading zeros of *R1* is odd or even (with conditional/test instructions!).
   b. The value of R13 is then calculated as follows:
      - If the leading zeros are even, *R13* is the sum of *R11* and *R12*.
      - Otherwise, *R13* is the subtraction of *R11* and *R12*.

   a) Assuming a *15 MHz* clk, report the code size and execution time in the following table:

Code size [Bytes]	Execution time [*ms*]	
	If the leading zeroes are even	Otherwise
564	0.00350	0.00425

4) Create two optimized versions of program 4 (where possible!)
   a. Using conditional execution.
   b. Using conditional execution in IT block.

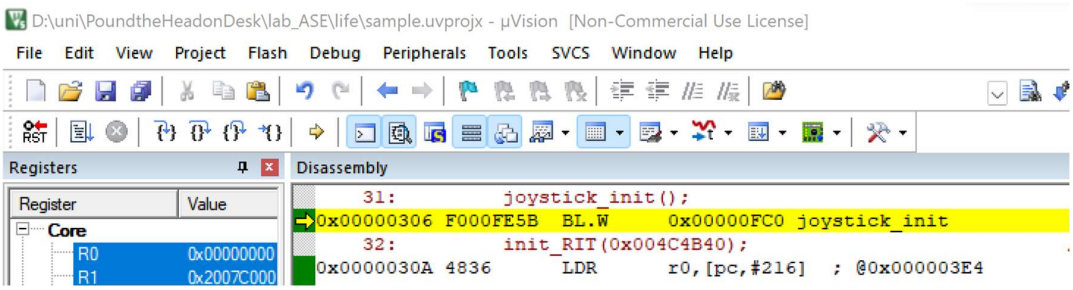
   Report and compare the execution Time

Program	Code size [Bytes]	Execution time [*ms*]	
		If the leading zeroes are even	Otherwise
Program 4 (baseline)	564	0.00350	0.00425
Program 4.a	564	0.00358	0.00408
Program 4.b	564	0.00375	0.00442

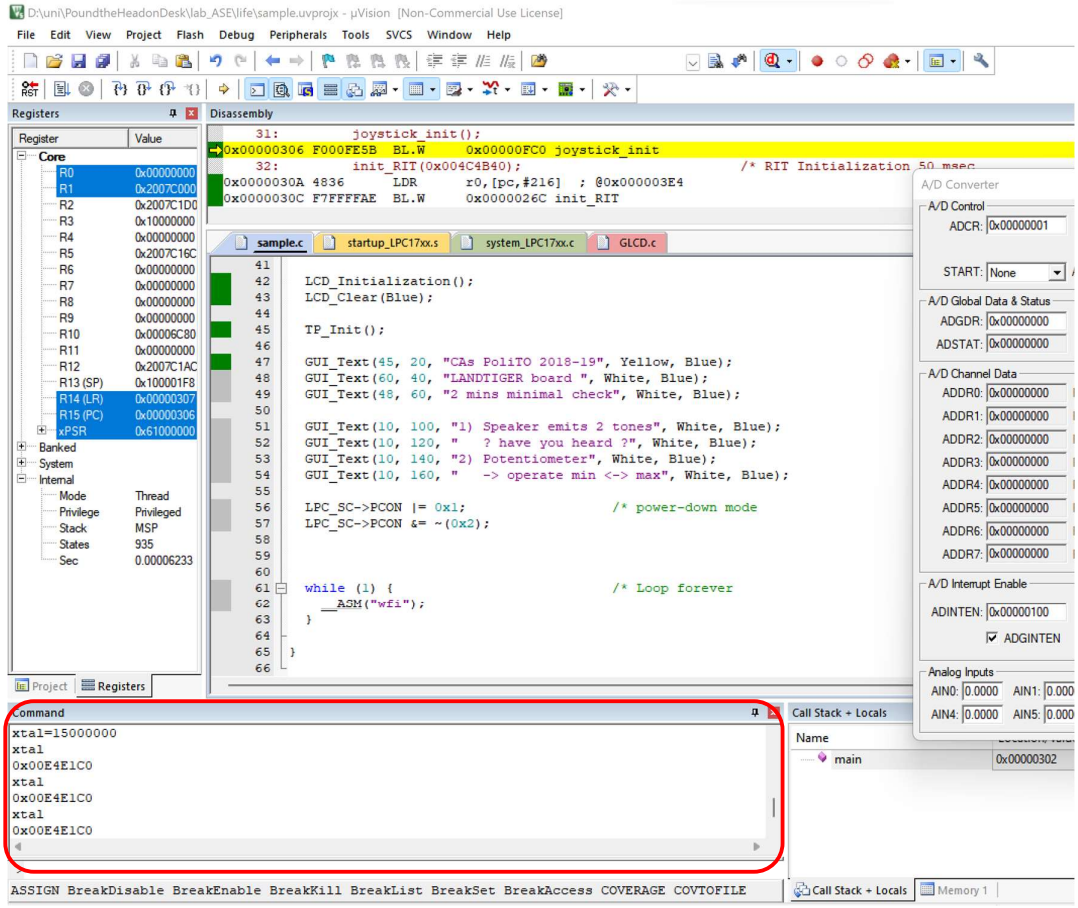
ANY USEFUL COMMENT YOU WOULD LIKE TO ADD ABOUT YOUR SOLUTION:
Nel 4.a nel caso del pari aumentano gli ms perchè usando il condizionale le operazioni che prima evitavo con la branch vengono fetchate e controllate; nel caso dispari ho un branch in meno in quanto le operazioni vengono direttamente messe con il conditional.

How to set the CPU clock frequency in Keil

1) Launch the debug mode and activate the command console.



2) A window will appear:



You can type *xtal* to check its value. To change its value, make a routine assignment, i.e., *xtal=frequency*, keeping in mind that frequency is in Hz must be entered. To set a frequency of *15 MHz*, you must write as follows: *xtal=15000000*.