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	0x07	0x0A	0X06	0XFFFFFFFF
Carry = 0	0x0	0xA	0X05	0X09
Overflow	0x7FFFFF88	0xFFFFFF80$_{(-128)}$	0x7FFFFFFF	0x7FFFFFFF
Negative	0x07	0xA	0x07	0xFFFFFF80$_{(-128)}$
Zero	0x07	0x07	0xFFFFFF80	0X80

Please explain the cases where it is **not** possible to force a **single** FLAG condition:

In ARM, non è in generale possibile forzare un singolo flag senza influenzare gli altri a causa della natura di come le operazioni aritmetiche e logiche aggiornano i flag.

2) Write a program that performs the following operations:
   a. Initialize registers *R2* and *R3* to random signed values.

b. Compare the two registers:
   - If they differ, store in register *R8* the maximum among *R2* and *R3*.
   - Otherwise, perform a logical right shift of 1 on *R3* (is it equivalent to what?), then subtract this value from *R2* 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	13	0.00108	15	0.00125

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 *R13*.

   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 [*replace this with the proper time measurement unit*]	
	If the leading zeroes are even	Otherwise
564	1.05us	0.92us

4) Create two optimized versions of program 3 (where possible!)
   a. Using conditional execution.
   b. Using conditional execution in IT block.

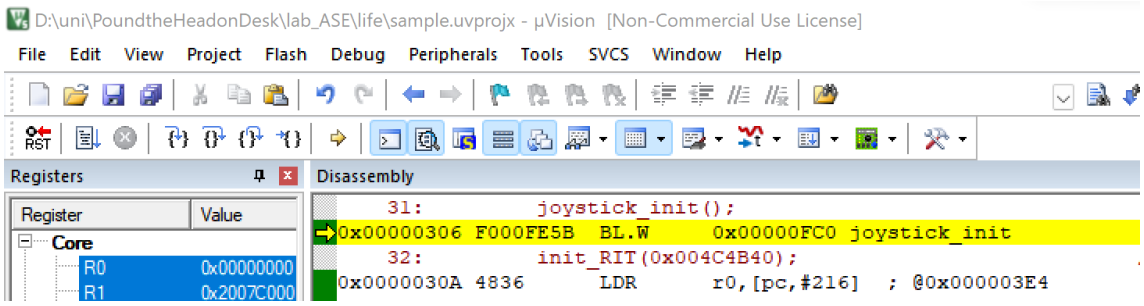
   Report and compare the execution Time

Program	Code size [Bytes]	Execution time [*replace this with the proper time measurement unit*]	
		If the leading zeroes are even	Otherwise
Program 3 (baseline)	564	1.05us	0.92us
Program 4.a	564	0.83us	0.83us
Program 4.b	564	0.83us	0.83us

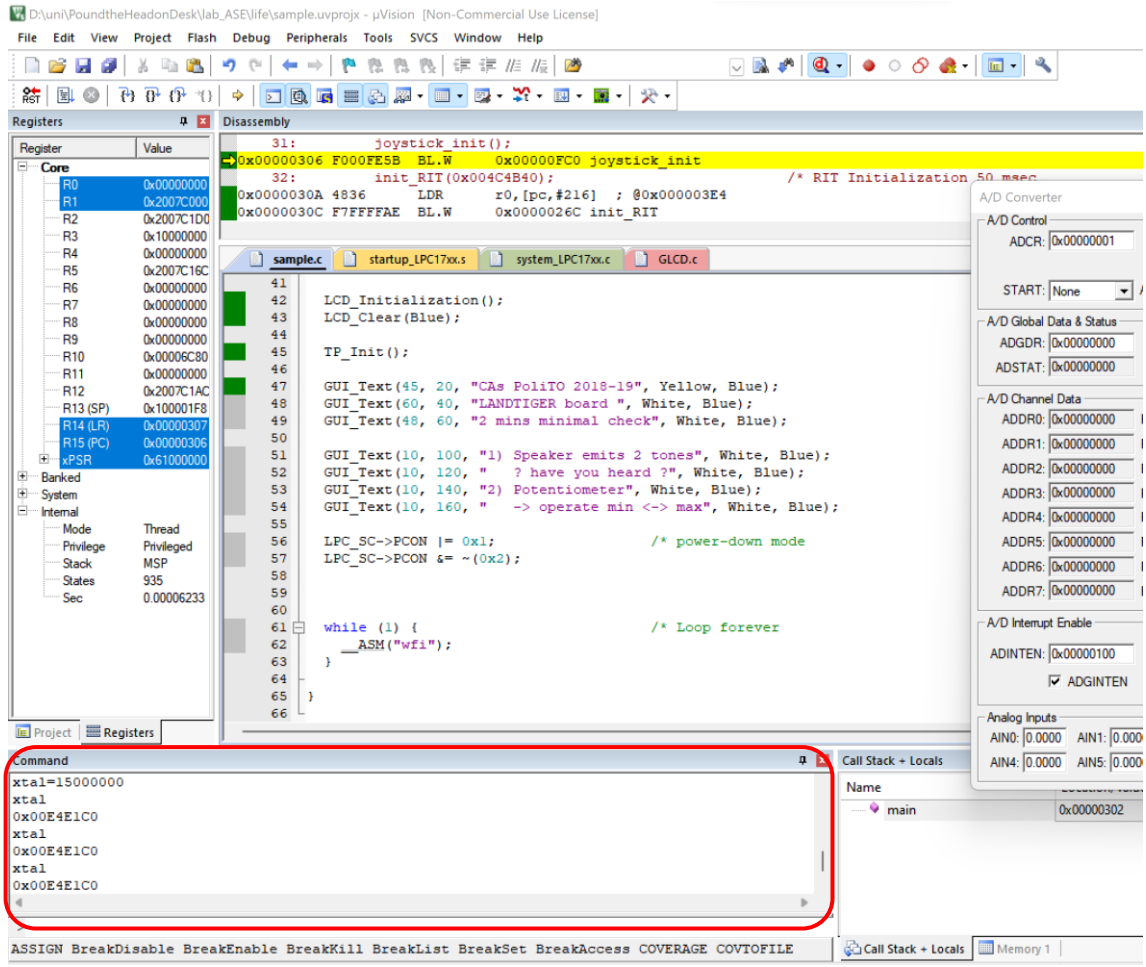
ANY USEFUL COMMENT YOU WOULD LIKE TO ADD ABOUT YOUR SOLUTION:
Nonostante la dimensione del codice non cambi, l'execution time diminuisce nei programmi ottimizzati. Tuttavia sia con il blocco IT sia senza, il tempo di esecuzione è lo stesso perché l'effettiva esecuzione delle istruzioni condizionali non cambia.

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*.