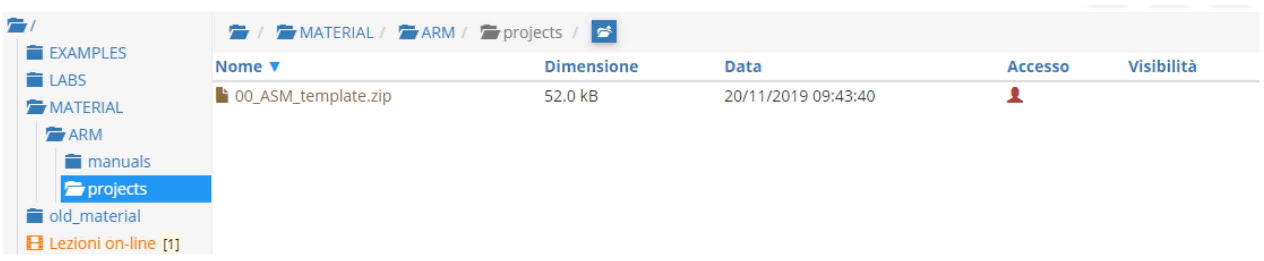
Architetture dei Sistemi di Elaborazione [AAA-GRA] Laboratory 6	Delivery date: 25th November 2022 Expected delivery of lab_06.zip must include: - Solutions of the exercises 1, 2 and 3 - this document compiled possibly in pdf format.

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



Nome ▼	Dimensione	Data	Accesso	Visibilità
00_ASM_template.zip	52.0 kB	20/11/2019 09:43:40		

1) Write a program using the ARM assembly that performs the following operations:
   a. Initialize registers R3 and R4 to random signed values
   b. Sum R0 to R3 (R0+R3) and store the result in R2
   c. Subtract R4 to R2 (R4-R2) and store the result in R5
   d. Force, using the debug register window, a set of specific values to be used in the program to provoke the following flag to be updated **once at a time** (whenever possible) to 1:
      - carry
      - overflow
      - negative
      - zero
   e. Report the selected values in the table below.

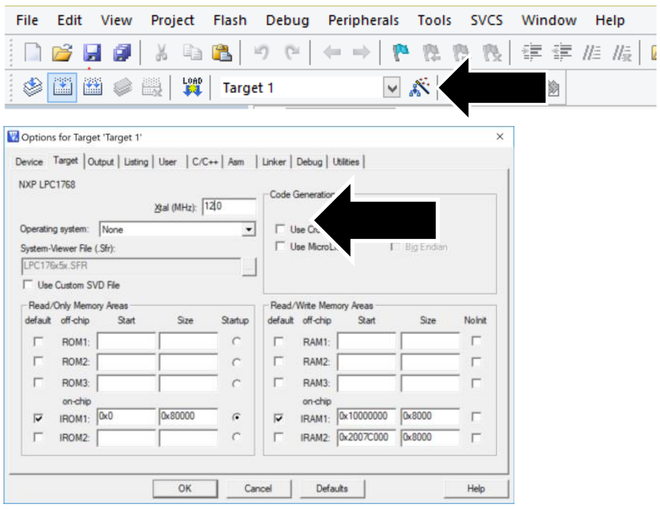
	Please, report the hexadecimal representation of the values			
Updated flag	R0 + R3		R4 – R2	
	R0	R3	R4	R2
Carry = 1	0x000000D1	-209	4	0x00000000
Carry = 0	0x000000D1	-212	4	0xFFFFFFFD
Overflow				
Negative	0x000000D1	-202	4	0x00000007
Zero	0x000000D1	-208	1	0x00000000

Please explain the cases when it is **not** possible to force a **single** FLAG condition:
- In the case of ZERO, as the last operation is a substraction the carry flag is always set to 1 as i had to substract from a positive number to have 0 as a result.

2) Write two versions of 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 the register R5 the maximum among R2 and R3
- Otherwise, perform a logical right shift of R3, sum R2 and store the result in R4

First, solve it by resorting to 1) a traditional assembly programming approach using conditional branches and then compare the execution time with a 2) conditional instructions execution approach.
Report the execution time in the two cases in the table that follows: **NOTE**, report the number of clock cycles (cc) considering a cpu clock (clk) frequency of 16 MHz, as well as the simulation time in milliseconds (ms).
Notice that the processor clock frequency is setup in the menu "*Options for Target: 'Target 1'*".



NXP LPC1768

Xtal (MHz): 12.0

Operating system: None

System-Viewer File (.Sfr): LPC176x5x.SFR

☐ Use Custom SVD File

Code Generation

☐ Use Cr

☐ Use MicroL

☐ Big Endian

Read/Only Memory Areas

default	off-chip	Start	Size	Startup
☐	ROM1:			○
☐	ROM2:			○
☐	ROM3:			○
	on-chip			
✓	IROM1:	0x0	0x80000	●
☐	IROM2:			○

Read/Write Memory Areas

default	off-chip	Start	Size	NoInit
☐	RAM1:			☐
☐	RAM2:			☐
☐	RAM3:			☐
	on-chip			
✓	IRAM1:	0x10000000	0x8000	☐
☐	IRAM2:	0x2007C000	0x8000	☐

	R2==R3 [cc]	R2==R3 [ms]	R2! =R3 [cc] (r2 as max/r3 as max)	R2! =R3 [ms] (r2 as max/r3 as max)
1) Traditional	14	0,00000087	12/13	0,00000075/0, 00000081
2) Conditional Execution	13	0,00000081	13/10	0,00000081/0, 00000063

3) Write a program that calculates the leading zeros of a variable. The leading zeros are computed by counting the number of zeros starting from the most significant bit and stopping at the first 1 encountered: e.g., the leading zeros of 0b00000101 are 5. The variable to check is in R1. After the count, if the number of leading zeros is odd, perform the sum between R2 and R3. If the number of leading zeros is even, perform the difference between R2 and R3. In both cases the result is placed in R4.

Implement the ASM code that performs the following operations:

a. Determines whether the number of leading zeros of R1 is odd or even.
b. As a result, the value of R4 is computed as follows:
    - If the leading zeros are even, R4 is the difference between R2 and R3
    - Else, R4 is the sum of R2 and R3
c. Report code size and execution time (with 15MHz clk) in the following table.

	Code size [Bytes]	Execution time [*replace this with the proper time measurement unit*]	
		If R2 is even	Otherwise
Exercise 3) computation	560	0,00000127	0,00000127

ANY USEFUL COMMENT YOU WOULD LIKE TO ADD ABOUT YOUR SOLUTION:

I USED THE CLZ INSTRUCTION TO COUNT THE LEADING ZERO. AFTER THAT, I TOOK THE QUOTIENT OF THE DIVISION BY 2 OF THE VALUE CONTAINED IN THE R1 REGISTER, I MULTIPLY IT BY 2 AND SUBTRACT WITH THE VALUE IN R1. IF IT’S EQUAL TO ZERO, THEN THE NUMBER OF LEADING ZERO IT’S EVEN OTHERWISE IT’S NOT.