Duration: 105 minutes.

It is possible to use:

- whatsoever paper medium

- slides downloaded from didattica website

- code from laboratories uploaded on didattica website in the “elaborati” section

Students caught talking among them will be immediately be asked to leave the lab.

Consider a natural number sequence in which, arbitrarily chosen the first number of the sequence , the following elements in the sequence are obtained in the following way:

The sequence terminates when the number 1 is obtained.

Example 1. If = 12, the sequence is: 12, 6, 3, 10, 5, 16, 8, 4, 2, 1. The sequence contains 10 elements.

Example 2. If = 19, the sequence is: 19, 58, 29, 88, 44, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1. The sequence contains 21 elements.

The Collatz conjecture states that, independently from the initial value , the sequence terminates every time, i.e. it always gets to one passing through a finite number of. The Collatz conjecture has never been theoretically demonstrated, but it has been verified experimentally for all natural numbers up to 87 \* 260 ≈ 1021.

You are asked to create a project with Keil, replace the contents of the startup\_LPC17xx.s file with the one in the **template** directory and add the other files in the directory. Finally, write **debugged and working subroutines** that meet the following 3 specifications.

Note 1: You should not change the code within Reset\_Handler that calls subroutines. It’s only required to implement subroutines. It is also recommended not to insert breakpoints in the instructions of Reset\_Handler, but to put them at the beginning of the implemented subroutines.

Note 2: Specifications must be run in order. You can only switch to Specification 2 after verifying that the solution to Specification 1 is working correctly. The same for Specification 3.

**Specifica 1** (8 punti). Scrivere in Assembly ARM una subroutine iterativeCollatz che riceve in input un numero naturale, tramite un ciclo calcola la sequenza di Collatz, e restituisce il numero di elementi che compongono la sequenza.

La subroutine deve essere conforme allo standard AAPCS (ARM Architecture Procedure Call Standard), in particolare per quanto riguarda il passaggio del parametro in input/output e il salvataggio dei registri.

Esempio 1: se il parametro in input è 12, il valore restituito è 10.

Esempio 2: se il parametro in input è 19, il valore restituito è 21.

Nota: Siccome l’unica operazione di divisione consiste nel calcolare la metà di un numero, non si possono usare le istruzioni UDIV e SDIV.

**Specification 1** (8 points). Write in ARM assembly an iterative subroutine iterativeCollatz that receives in input a natural number, through a cycle calculates the sequence of Collatz, and returns the number of elements that compose the sequence.

The subroutine shall comply with the ARM Architecture Procedure Call Standard (AAPCS) standard, in particular with regard to switching the parameter to input/output and saving the registers.

Example 1: If the input parameter is 12, the return value is 10.

Example 2: If the input parameter is 19, the return value is 21.

Note: Since the only splitting operation is to calculate half of a number, UDIV and SDIV instructions cannot be used.

**Specification 2** (6 points). Write in assembly ARM a subroutine recursiveCollatz that receives in input two parameters:

P1: a natural number corresponding to the element of the sequence

P2: the number *i*

The subroutine recursiveCollatz modifies the two parameters in input performing the following operations:

* increases the second parameter by 1: P2new = P2 + 1
* if the first parameter is equal to 1, the subroutine sets P1new = 1 and then ends (return to calling program)
* otherwise, the subroutine calculates the element of the sequence, sets P1new = and then calls itself up with P1new and P2new as ­­­­­­­­­parameters.

The subroutine shall comply with the ARM Architecture Procedure Call Standard (AAPCS) standard, in particular as regards the switching of input/output parameters and the saving of registers.

Example 1: If the input parameters are P1 = 12 and P2 = 0, the subroutine (after 9 recursive calls) returns to the calling program P1new = 1 and P2new = 10.

Example 1: If the input parameters are P1 = 19 and P2 = 0, the subroutine (after 20 recursive calls) returns to the calling program P1new = 1 and P2new = 21.

Note: Since the only splitting operation is to calculate half of a number, UDIV and SDIV instructions cannot be used.

**Specification 3** (5 points). Write appropriate instructions / functions in C for:

1. Start the timer 0 of LPC 1768 board. The timer must not trigger interrupt.

2. when the INT0 button is pressed:

* + read the current value of timer counter register of timer 0.
  + call the iterative subroutine iterativeCollatz by passing the timer counter value.
  + store in a variable the minimum between 255 and the value returned by the iterativeCollatz subroutine.
  + display the binary representation of the value of the variable on the led 4-11 (the led 11 corresponds to the least significant bit).