|  |  |
| --- | --- |
| **Architetture dei Sistemi di Elaborazione 02GOLOV [M-Z]** | Delivery date:  12/12/2019 |
| **Laboratory**  **7** | Expected delivery of lab\_07.zip must include:   * zipped project folder of the exercise 1 * zipped project folder of the exercise 2 * this document filled in pdf format. |

Solve the following problems by starting from the *ASM\_template* project.

**Exercise 1)** Experiment the SVC instruction.

Write, compile and execute a code that invokes an SVC instruction when running a **user routine** with **unprivileged access level**. By means of invoking a SuperVisor Call, we want to implement a RESET, a NOP and a MEMCPY functions. The MEMCPY function is used to copy a block of data from a source address to a destination address and return information about the data transfer execution.

In the handler of SVC, the following functionalities are implemented according to the SVC number:

1. 0 to 7: RESET the content of register R**x**, where **x** can assume values from 0 to 7
2. 8 to 15 and >=128: NOP
3. 64 to 127: the SVC call have to implement a MEMCPY operation, with the following input parameters and return values:
   * the 6 least significant bits of the SVC number indicates the number of bytes to move
   * source and destination start addresses of the areas to copy are 32 bits values passed through stack
   * by again using the stack, it returns the number of transferred bytes



Example: the following SVC invokes MEMCPY from a given source to a destination

LDR R0, =SourceStartAddress

LDR R1, =DestinationStartAddress

PUSH {R0}  
PUSH {R1}

SVC 0x48 ; 2\_**01**001000 binary value of the SVC number

POP {R0}

Q1: Describe how the stack structure is used by your project.

Q2: What need to be changed in the SVC handler if the access level of the caller is privileged? Please report code chunk that solves this request.

Q3: Is the encoding of the SVC numbers complete? Please comment.

**Exercise 2)** Integrate ASM and C language functionalities

The following function, written in ASSEMBLY language, is invoked from a main C language function:

unsigned int average(unsigned int\* V, unsigned int n);   
/\* where n is the number of V elements \*/

The function returns exclusively one return value, which could be either:

* the integer average value of the values stored in V, or
* the value 0 if any significant error is encountered in the accumulation of the values.

The main C language function declares an unsigned integer vector called V and composed of N elements (**N chosen by you**). At declaration time, **the vector is statically filled** by random values.

Please fill the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| *F = 12MHz* | Execution time  (clock cycles) | Code size | Data size |
| Exercise 1) |  |  |  |
| Exercise 2) |  |  |  |