|  |  |
| --- | --- |
| **Computer Architectures 02LSEOV 02LSEOQ [AA-LZ]** | Delivery date:  Wednesday 12/12 |
| **Laboratory**  **7** | Expected delivery of lab\_07.zip must include:   * zipped project folder of the exercise 1 * this document compiled possibly in pdf format. |

Solve the following problem by starting from the *template.s* file.

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

Write, compile and execute a code that invokes a 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**?**, where **?** 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.

Lo stack è diviso in due parti, una usata come Main Stack e l’altra come Proces Stack.

A livello utente si ha un’esecuzione thread level, senza privilegi e quindi usando uno stack pointer di tipo PSP.

Quando si entra nella SVC l’esecuzione passa ad handler level e con privilege, usando uno stack pointer di tipo MSP. Questo implica che vengono usati due strutture stack diverse per l’utente e per l’SVC handler.

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.

Per passare ad un access level privilegiato del chiamante è necessario modificare il terzo bit del Control Register. Questo permette al chiamante di usare il Main Stack.

Ciò implica che, a livello del SVC Handler, si accede al Main Stack (quindi con uno stack pointer di tipo MSP) per ricavare i parametri passata via stack dal chiamante e modificare i registri R0-R3.

Per i parametri le uniche modifiche da effettuare sono I valori di offset usati per leggere dentro lo stack.

Per quanto riguarda la modifica dei registri basta sostituire la riga MRS R9,PSP con

MRS R9,MSP e modificare I valori di offset utilizzati per accedere ai registri.

Immagine che contiene screenshot

Descrizione generata automaticamenteImmagine che contiene testo, screenshot

Descrizione generata automaticamenteImmagine che contiene testo

Descrizione generata automaticamenteIl codice che implementa questa situazione è il seguente:

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

No, mancano i valori compresi tra 16 e 63.

**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 alternatively:

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

The main C language function takes care of declare an unsigned integer vector called V and composed of N elements. 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) | 297.96 | 564 | RO:204; RW:64; ZI:512 |
| Exercise 2) | 654.96 | 504 | RO:264; ZI:608 |