Le SuperVisor Call sono delle eccezioni che vengono *triggerate* dall'istruzione **SVC**.
In un ambiente **OS**, permettono la chiamate a system call per accedere determinate funzioni del kernel etc.

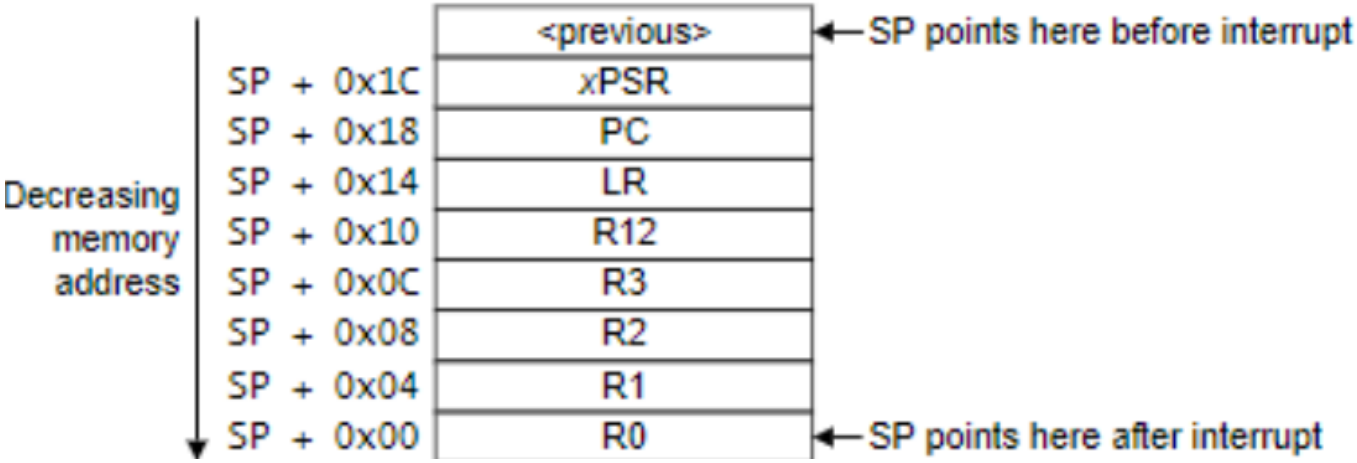
Stato delle eccezioni

Le *exceptions* possono trovarsi in uno dei seguenti tre stati:

- **inactive:** non è nè *active* nè *pending*
- **active:** *exception* che sta venendo gestita dal processore ma non completata
- **pending:** *exception* che sta aspettando di venir gestita dal processore.

Stack Frame

Quando il processore inizia a gestire una *exception*, prima di tutto *pusha* sullo stack vari registri.
Quest'operazione viene detta **stacking** e le informazioni vengono dette **stack frame**



Exception handler

Gli handler di default vengono dichiarati come **weak** per permettere l'overloading da parte del programmatore.
Di solito questi handler di default vengono implementati come cicli infiniti, quindi bisogna fare attenzione

SVC

L'istruzione ha la seguente sintassi **{label} SVC <immediate>** dove **immediate** è un numero a 8 bit(0-255) che identifica la system call.
Questo numero viene scartato dal processore in quanto quello che fa la SVC è *eseguire*

l'handler SVC.
Dunque spetta al programmatore, nell'handler SVC, estrapolare questo **immediate** e in base a ciò eseguire codice diverso.
Quindi nell'handler SVC avremo una sorta di switch case con i vari codici dell'SVC.

Execution

All'esecuzione di **SVC #imm**:

1. Lo **stack frame** viene salvato
   - Vengono **pushati** automaticamente **R0-R3,R12,LR,PC,xSPR**!!!!!
2. Il **LR** viene aggiornato con **EXC_RETURN**
   - **EXC_RETURN** è un codice che indica in quale modalità ritornare dalla gestione dell'eccezione

EXC_RETURN	Description
0xFFFFFFF1	Return to Handler mode. Exception return gets state from the main stack. Execution uses MSP after return.
0xFFFFFFF9	Return to Thread mode. Exception Return get state from the main stack. Execution uses MSP after return.
0xFFFFFFFD	Return to Thread mode. Exception return gets state from the process stack. Execution uses PSP after return.
All other values	Reserved.

3. Dal momento che lo **stack frame** include anche il **PC**, possiamo usarlo per prenderci l'istruzione **SVC** situata a **PC-4**, per ripescare **#imm**.
4. Attraverso una manipolazione di bit e shift, ci prendiamo il codice **#imm**.
5. Gestione del codice **#imm**
6. Alla fine si ritorna *exploitando* il valore **EXC_RETURN** in **LR**.

7. I valori dei registri vengono ripristinati

```
SVC_Handler
                LDR R0, [SP, #24]
                LDR R1, [R0,#-4]
                BIC R1, 0xFF000000
                LSR R1, #16




                BX LR
END
```

Template dell'handler SVC. La BX LR è speciale in quanto in LR è contenuto EXC_RETURN ma in qualche modo riesce ad usare il LR dello stack.

Cambiare modalità operativa del processore

E' possibile cambiare modalità operativa del processore e privilegi attraverso l'istruzione **MSR**. L'istruzione **MSR** permette di cambiare contenuto di un registro speciale, tuttavia è possibile usarla solo quando in uno stato *priviliged*.

```
MOV  R0, #3

MSR  CONTROL, R0
```

Questo pezzo di codice imposta a 1 i primi due bit(su 3) del registro **CONTROL**, portando il processore:

- in uno stato *unpriviliged*
- in *thread mode*
- ad utilizzare lo stack pointer alternativo **PSP**.
  Di solito viene usato per settare il processore in una modalità *user*.