

Reset

System privileged
Stack pointer used MSP

When come Interrupt or exception

- Finish current instruction (except for lengthy instructions)
- Push context (8 register 32bit) onto current stack(MSP or PSP)
- xPSR, Return Address, LR(R14), R12, R3, R2, R1, R0
- Switch to handler/ privileged mode, use MSP
- Load PC with address of exception handler
- Load LR with EXC_Return code
- Load IPSR with exception number
- Start executing code of exception handler

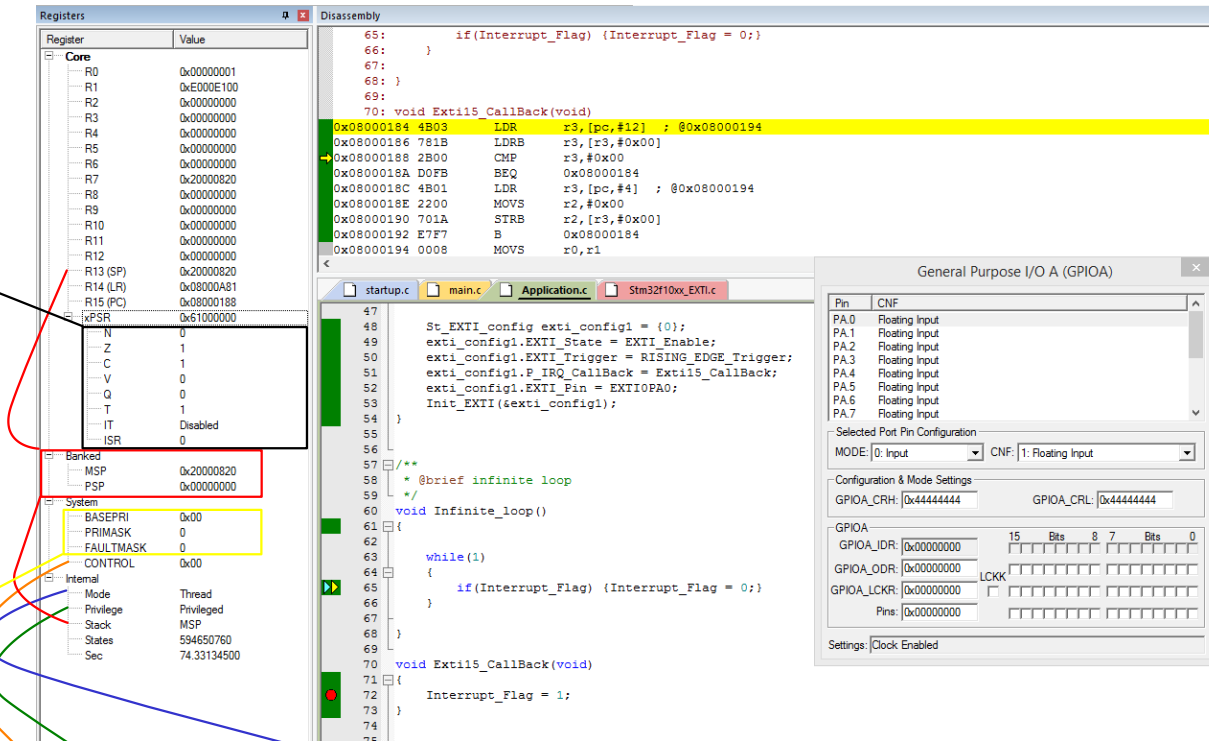
Exiting an exception handler

- execute instrcutin triggering exception return processing
- select return stack, restore context from that stack
- resum execution of code at resoted address

Bit	Description
N	Negative flag
Z	Zero flag
C	Carry (or NOT borrow) flag
V	Overflow flag
Q	Sticky saturation flag (not available in ARMv6-M)
GE[3:0]	Greater-Than-or-Equal flags for each byte lane (ARMv7E-M only; not available in ARMv6-M or Cortex-M0)
ICVT	Instruction-Controllable Instruction (ICI) bits. If-THEN instruction status bit for conditional execution (not available in ARMv6-M).
T	Thumb state, always 1; trying to clear this bit will cause a fault exception.
Exception Number	Indicates which exception the processor is handling.

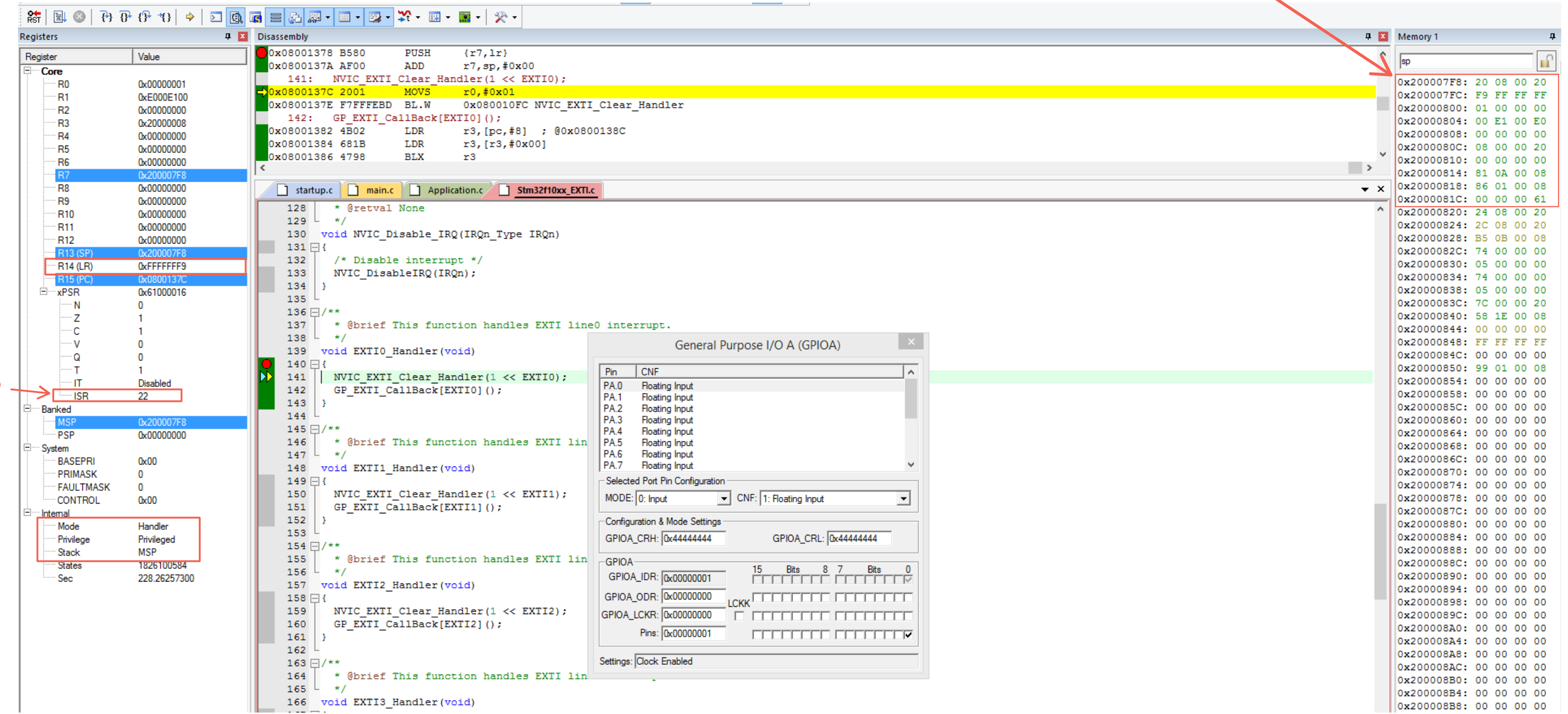
- **Base priority mask register (BASEPRI)** defines the priority threshold, and the processor disables all interrupts with a higher priority **value** than the threshold.
 - A lower priority **value** represents a higher priority (or urgency).
- **Priority mask register (PRIMASK)** is used to disable all interrupts excluding hard faults and non-maskable interrupts (NMI).
 - If an interrupt is masked, the processor disables this interrupt.
- **Fault mask register (FAULTMASK)** is used to disable all interrupts excluding nonmaskable interrupts (NMI).

Bit	Name	Position
[17:0]	Reserved	
[1]	SPSEL	Defines the currently active stack pointer. In Handler mode this bit needs to zero and again when the Vector Catch-MP enables the Vector Catch-MP for exception return.
0	MSP	0 = MSP is the current stack pointer
1	PSP	1 = PSP is the current stack pointer
[0]	SPRDY	Defines the Thread mode privilege level: <ul style="list-style-type: none">0 = Privileged1 = Unprivileged



5. Load LR With EXC_RETURN Code			
► EXC_RETURN value generated by CPU to provide information on how to return			
► Which SP to restore registers from? MSP (0) or PSP (1)			
► Previous value of SPSEL			
► Which mode to return to? Handler (0) or Thread (1)			
► Another exception handler may have been running when this exception was requested			
► With bit [4:7], using floating point unit or not (E, yes; F, no)			
EXC_RETURN	Return Mode	Return Stack	Description
0xFFFF_FFE1	0 (Handler)	0 (MSP)	Return to handler mode (always Main Stack)
0xFFFF_FFE9	1 (Thread)	0 (MSP)	Return to thread with MSP
0xFFFF_FFE5	1 (Thread)	1 (PSP)	Return to thread with PSP

Interrupt number to calculate
 $0x16 - 0x16 = 0x00$ first IRQ



Saved data when come Interrupt