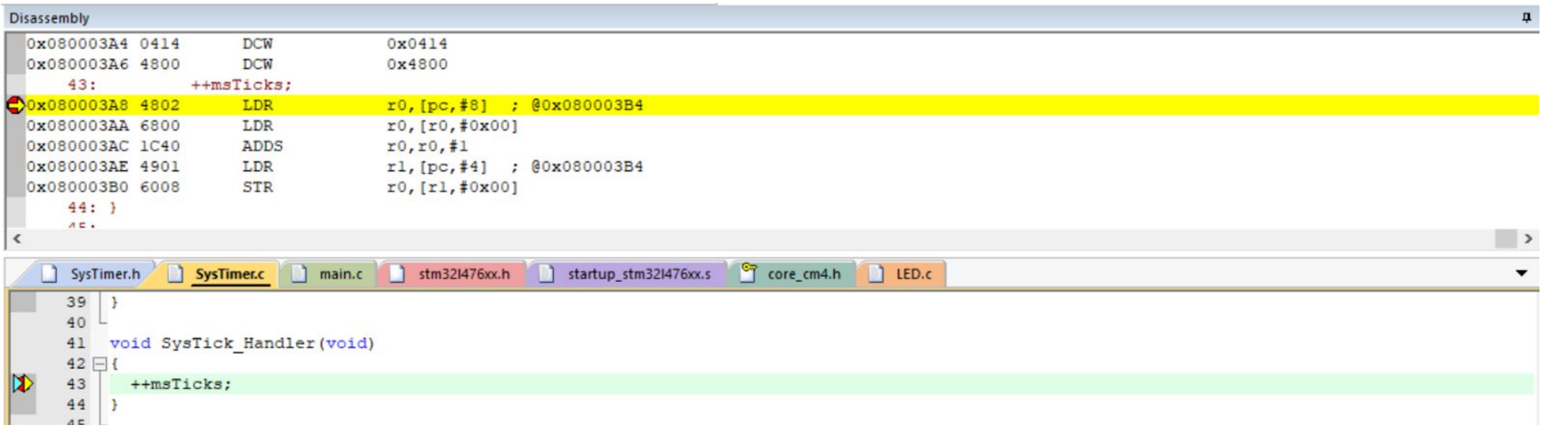
1. What is the address of the SysTick Handler() function? Verify it (i.e. take a screenshot) in the debug environment.

The address of `SysTick_Handler()` is `0x080003A8`.



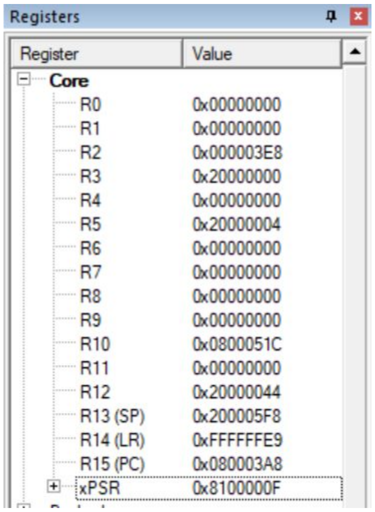
```
0x080003A4 0414      DCW          0x0414
0x080003A6 4800      DCW          0x4800
    43:          ++msTicks;
0x080003A8 4802      LDR          r0,[pc,#8]  ; @0x080003B4
0x080003AA 6800      LDR          r0,[r0,#0x00]
0x080003AC 1C40      ADDS         r0,r0,#1
0x080003AE 4901      LDR          r1,[pc,#4]  ; @0x080003B4
0x080003B0 6008      STR          r0,[r1,#0x00]
    44: }
```

```
}

void SysTick_Handler(void)
{
  ++msTicks;
}
```

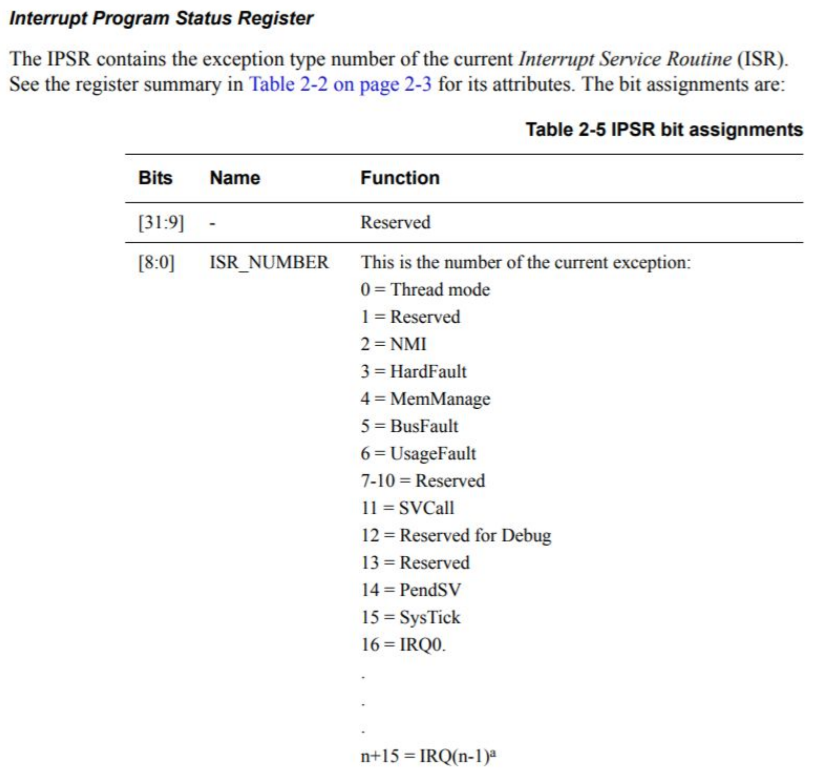
2. Set up a breakpoint within the SysTick Handler() function. In the debug environment, find out the exception number in the program status register when the program runs to the breakpoint. Explain what this number means.

The program status register has a value of `0x8100000F` as shown below



Register	Value
Core	
R0	0x00000000
R1	0x00000000
R2	0x000003E8
R3	0x20000000
R4	0x00000000
R5	0x20000004
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x0800051C
R11	0x00000000
R12	0x20000044
R13 (SP)	0x200005F8
R14 (LR)	0xFFFFFFE9
R15 (PC)	0x080003A8
xPSR	0x8100000F

According to the Cortex-M4 user guide bits [31:9] are reserved and bits [8:0] represent the ISR number as shown in the image on the next page. ‘0x0F’ is 15 in decimal which is interpreted as the SysTick exception type number. In the documentation, it says “A SysTick exception is an exception the system timer generates when it reaches zero. Software can also generate a SysTick exception.”



Interrupt Program Status Register

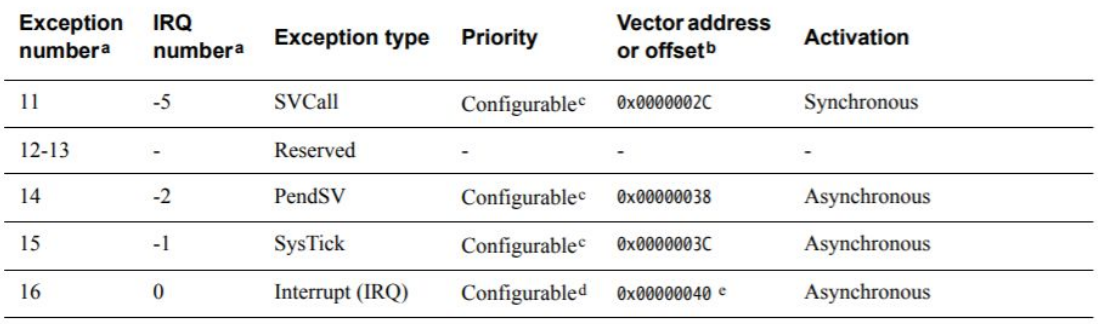
The IPSR contains the exception type number of the current *Interrupt Service Routine* (ISR). See the register summary in Table 2-2 on page 2-3 for its attributes. The bit assignments are:

Table 2-5 IPSR bit assignments

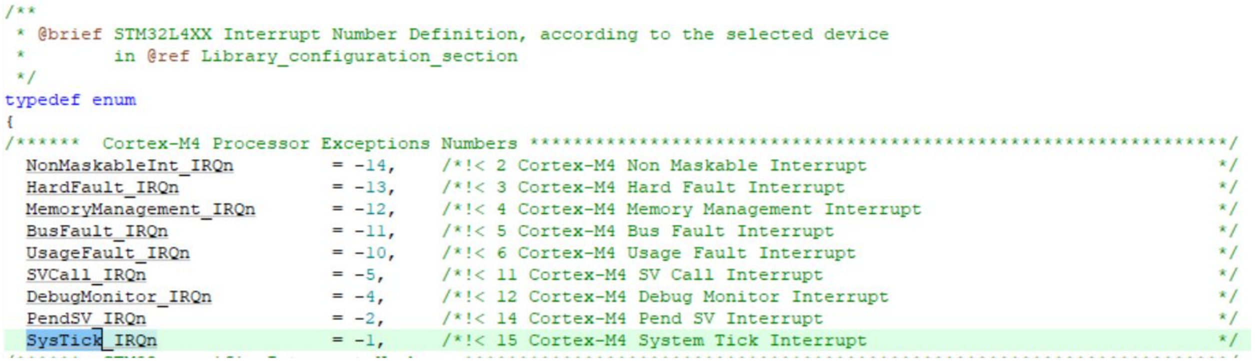
Bits	Name	Function
[31:9]	-	Reserved
[8:0]	ISR_NUMBER	This is the number of the current exception: 0 = Thread mode 1 = Reserved 2 = NMI 3 = HardFault 4 = MemManage 5 = BusFault 6 = UsageFault 7-10 = Reserved 11 = SVCall 12 = Reserved for Debug 13 = Reserved 14 = PendSV 15 = SysTick 16 = IRQ0. . . . n+15 = IRQ(n-1)[a]

3. Cortex-M series supports up to 256 interrupts. What is the interrupt number of SysTick that is defined in CMSIS?

As seen in the table, the interrupt number of SysTick is defined as **-1**. (Also in STM32l476xx.h)



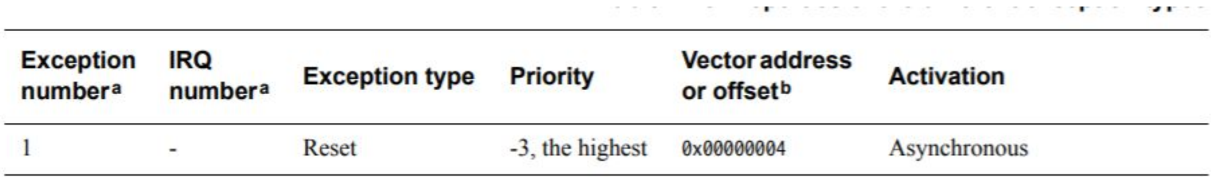
Exception number[a]	IRQ number[a]	Exception type	Priority	Vector address or offset[b]	Activation
11	-5	SVCall	Configurable[c]	0x0000002C	Synchronous
12-13	-	Reserved	-	-	-
14	-2	PendSV	Configurable[c]	0x00000038	Asynchronous
15	-1	SysTick	Configurable[c]	0x0000003C	Asynchronous
16	0	Interrupt (IRQ)	Configurable[d]	0x00000040 [e]	Asynchronous



```
/**
 * @brief STM32L4XX Interrupt Number Definition, according to the selected device
 *        in @ref Library_configuration_section
 */
typedef enum
{
/******  Cortex-M4 Processor Exceptions Numbers ****************************************************************/
  NonMaskableInt_IRQn         = -14,    /*!< 2 Cortex-M4 Non Maskable Interrupt                                */
  HardFault_IRQn              = -13,    /*!< 3 Cortex-M4 Hard Fault Interrupt                                  */
  MemoryManagement_IRQn       = -12,    /*!< 4 Cortex-M4 Memory Management Interrupt                           */
  BusFault_IRQn               = -11,    /*!< 5 Cortex-M4 Bus Fault Interrupt                                   */
  UsageFault_IRQn             = -10,    /*!< 6 Cortex-M4 Usage Fault Interrupt                                 */
  SVCall_IRQn                 = -5,     /*!< 11 Cortex-M4 SV Call Interrupt                                    */
  DebugMonitor_IRQn           = -4,     /*!< 12 Cortex-M4 Debug Monitor Interrupt                              */
  PendSV_IRQn                 = -2,     /*!< 14 Cortex-M4 Pend SV Interrupt                                    */
  SysTick_IRQn                = -1,     /*!< 15 Cortex-M4 System Tick Interrupt                                */
```

4. Does a higher priority value represent a higher urgency?

No, a lower priority value represents a higher urgency. For example, “Reset” has the highest urgency and has a priority value of -3 as shown.



Exception number[a]	IRQ number[a]	Exception type	Priority	Vector address or offset[b]	Activation
1	-	Reset	-3, the highest	0x00000004	Asynchronous

5. Suppose a clock of 16 MHz is used to drive the system timer. What is the maximum period between two consecutive SysTick interrupts that we can possibly obtain?

SysTick **Reload is 24 bits** so the max value is (2^24 **-**1) but it counts down to zero so we add one tick so that the largest amount of ticks between two interrupts is **2^24**. A tick happens every clock cycle or every **1/16MHz. Max period** = 2^24/(16*10^6) = **1.048576 seconds**