**Task 1**

1. *What is the maximum value that the SysTick counter can start counting down from?*

or or

1. *What are the addresses of the CTRL, LOAD and VAL registers?*

CTRL: 0xE000E010

LOAD: 0xE000E014

VAL: 0xE000E018

1. *To use register names (CTRL, LOAD and VAL) in my assembly code, how to define them? (Hint: EQU)*

SYSTICK\_CTRL EQU 0xE000E010

SYSTICK\_LOAD EQU 0xE000E014

SYSTICK\_VAL EQU 0xE000E018

1. *What is the value to be written in the LOAD register to activate the SysTick interrupt every 2 ms, in case the Clock Source is 12MHz?*

or

1. *What are the clock sources that you choose from to configure the SysTick?*

MCK (value = 1) or MCK/8 (value = 0).

1. *Write down an assembly program that enables SysTick interrupt with a period of 5 ms*.

SYSTICK\_CTRL EQU 0xE000E010

SYSTICK\_LOAD EQU 0xE000E014

SYSTICK\_VAL EQU 0xE000E018

main

LDR R0, =SYSTICK\_CTRL

MOV R1, #0

STR R1, [R0]

LDR R0, =SYSTICK\_VAL

STR R1, [R0]

LDR R0, =SYSTICK\_LOAD

LDR R1, =49999

STR R1, [R0]

LDR R0, =SYSTICK\_CTRL

MOV R1, #0x07

STR R1, [R0]

STOP B STOP

END

**Task 2**

1. *What happens to the stack in the event of SysTick interrupt?*

It pushes registers R0-R3, R12, PC, PSR and LR to the current stack.

1. *What is saved on the stack in the event of SysTick interrupt?*

It saves the values in registers R0-R3, R12, PC, PSR and LR to the current stack.

1. *What is in the LR register when the SysTick interrupt handler executes, and what does it mean?*

0xFFFF’FFF9

It’s a specific value that will tell the program where we were when the interrupt happens.

If we write the value of LR to PC we will end the exception just like any other subroutine

1. *What instruction causes the SysTick interrupt to end, i.e. how do you return to the main program?*

*We use BX LR to write the address of LR to PC so we end the exception handler that was created by SysTick*

1. *Where is the return address stored?*

In the PC (Program Counter)

1. *What are the shared resources between the main program and the SysTick Handler?*

Registers R0, R4, R6

1. *What is the maximum value that the Binary counter of the main program reaches? Why?*

0x0000’34e9

Because the third time it leaves the interrupt loop it’s going to CMP R6, #0x0 and since R6 is always going to be 0x0 after the interrupt we are going to end the program.

1. *Do you recognise any sensitive code section in this program? Where is it?*

SUB R6, R0, R4

CMP R6, #0x0.

Since R6 is a global register and used both in the main and SysTick\_Handeler

1. *If you recognise a sensitive code section in your program, How can you protect it? (Hint: Do not use PUSH and POP)*

CPSID I

SUB R6, R0, R4

CMP R6, #0x0

CPSIE I