

Tutorial 5

Question 1: Exceptions (5)

In the event that an instruction tries to perform an unaligned data access, describe what happens...

- a) ... if we **have not** placed useful data at the HardFault vector location? (2)
- b) ... if we **have** placed useful data at the HardFault vector location? (2)
- c) What do we mean by 'useful data'? (1)

Question 2: Stack (6)

- a) Why is having a stack useful? (1)
- b) Why do we want the stack to start as close to the end of RAM as possible? (1)
- c) Is it not a problem that the initial SP value (0x2000 2000) is outside of RAM? (1)
- c) Write a sequence of instructions (excluding push and pop) which will perform the same job as: (should be somewhere between 7 and 11 instructions)

```
PUSH {R0}
PUSH {R1, R2, LR}
POP {R0} (3)
```

Question: Subroutines (3)

- a) What is a subroutine? (1)
- b) Why is the ability to use subroutines very useful? (1)
- c) If the instruction `BL foo` is located at address 0x0800 1000, what data will be placed into R14 when the instruction executes? Note that there is an unfortunate complexity to this. See page 123 of the ARMv6-M Architecture Reference Manual (1)

Question: Breakpoints (5)

<https://sourceware.org/gdb/current/onlinedocs/gdb/Breakpoints.html>

- a) Explain what breakpoints are, and why they are useful. (2)
- b) Give an example of three different ways (actual GDB command) which we can set breakpoints. (2)
- c) What GDB command should we write to delete all breakpoints we have set? (1)

Bonus: (2)

In the event that an NMI occurs due to the HSE failing, write a block of code (about 6 lines) which should be executed by the NMI_Handler in order to acknowledge the exception and return to running where the program was when the exception occurred.