

Tutorial 2

Question 1: (6)

- a) What are two benefits of using the MOVS instruction rather than an LDR instruction to set a register to a value? (2)
- b) When are we able to use a MOVS instruction instead of a LDR? (1)
- c) Give a brief explanation of how breakpoints work. (2)
- d) Why might we want to switch from the internal RC oscillator to an external oscillator? (1)

Question 2: (8)

- a) Give a brief explanation of why pull resistors are necessary. (2)
- b) Which pull resistor should be set for our dev board. Explain why. (1)
- c) Why would it be a bad idea to write a word with the value 0 to the GPIOA_MODER? (2)
- d) Write some code which will set the upper byte (MSB) of word at memory address 0x20000100 to 0xAB while keeping the other bits unchanged. Note that the MSB is not necessarily 0 to start with. (3)

Question 3: (4)

- a) Assuming our micro is running at 8 MHz, how long will the following take to execute? Explain. (marks for explanation) (2)

```
LDR R1, my_data
MOVS R0, #0
BEQ some_label
```

- b) Explain why we add +1 to our reset vector, and why we are able to do this. (2)

Question 4: (5)

- a) For the following small program, what offset is applied to the PC by the load instruction? Explain. (marks for explanation). Assume the address of the LDR instruction is on a word boundary. I.e: it's address is a multiple of 4. (3)

```
LDR R0, foo
ADDS R0, R0, R0
B loop    @ assume 'loop' is defined elsewhere
.align
xyz: .word 0xAABBCCDD
abc: .word 0xFF
foo: .word 0x0
```

- b) How would the above be different if LDR was not on a word boundary? (2)

Marked out of: 23

Available marks: 23