1. **What are the GPIO control registers that the lab mentions? Briefly describe each of their functions.**

Moder Register-> GPIO port mode register. Set the mode of the pin. Such as general purpose that we used in the lab

Output Type Register-> GPIO port output type register. Sets the output to push pull or open drain.

GPIO port output speed register-> GPIO speed register. Sets the speed of the GPIO pin. Low, medium, high speed.

GPIO port pull-up, pull-down register-> Sets the GPIO to pull up, pull down

GPIO port input data register: Outputs the value of the GPIO port

1. **What values would you want to write to the bits controlling a pin in the GPIOx\_MODER register in order to set it to analog mode?**

It should be set to 11.

1. **Examine the bit descriptions in GPIOx\_BSRR register: which bit would you want to set to clear the fourth bit in the ODR?**

I think that we should set bit 20 of the BSRR to reset the fourth bit in the output data register.

1. **Perform the following bitwise operations: (Operations)**

0xAD (OR) 0xC7 = 10101101 (OR) 11000111 = 11101111

0xAD (AND) 0xC7 = 10101101 (AND) 11000111 = 10000101

0xAD (AND) (NOT) (0xC7) = 10101101 (AND) (11000111) (NOT) -> 00111000 = 00101000

0xAD (XOR) 0xC7 = 10101101 (XOR) 11000111 = 01101010

1. **How would you clear the 5th and 6th bits in a register while leaving the others alone?**

EX) GPIOC -> MODER = ~ ((1 << 6) | (1 << 5)).

1. **What is the maximum speed the STM32F072R8 GPIO pins can handle in the lowest speed setting?**

First, OSPEEDR should be \_0 (\_ being Don’t Care Value). On maximum ratings on the Chip datasheet, if the I/O supply voltage is less than 2V then f (Frequency) at the low setting is 1M (in Hz) otherwise the max frequency is 2M (in Hz)

1. **What RCC register would you manipulate to enable the following peripherals: (use the comments next to the bit defines for better peripheral descriptions)**

TIM1 (TIMER1)-> RCC\_APB2ENR #bits 11

DMA1-> RCC\_AHBENR #bits 00

I2C1-> RCC\_APB1ENR #bits 21