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

GPIO port output type register (GPIOx_OTYPER)
This register selects the output mode you want for each pin. For pins without output mode configured, the bits in this register have no effect.

GPIO port output speed register (GPIOx_OSPEEDR)
Changes the speed of the registers to reduce power consumption and increase efficiency.

GPIO port pull-up/pull-down register (GPIOx_PUPDR)
Allows you to configure a pull up or pull down network using a transistor.

GPIO port input data register (GPIOx_IDR)
The data input register always reports the logical state of each pin in the GPIO port. If the pin is an input, then the matching bit will show the logical state to which the outside world is driving that pin; if configured as an output, then it will show whatever logical state that you have set in the output register for that pin.

GPIO port output data register (GPIOx_ODR)
This register sets the logical state of configured output pins. Writing a '0' to this register will pull the output low for that pin; writing a '1' to the bit will drive the output high.

GPIO port bit set/reset register (GPIOx_BSRR)
The bit set/reset register allows much faster modification to the output register because you only change the desired values. You may simply overwrite the entire register; it will only modify the output register on the bits that you have set.The lower half of this register sets bits in the output, and the upper half clears/resets them.

GPIO port configuration lock register (GPIOx_LCKR)
This will lock the desired register so that it cannot be edited or changed.

GPIO alternate function low/high registers (GPIOx_AFRL/GPIOx_AFRH)
The device specific datasheet contains a table which lists the possible alternate functions for each pin, as well as the number of each
function.

GPIO port bit reset register (GPIOx_BRR)
This reset-only register resets with the clearing bits in the lower half.

2. 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?
Set the bits to 11.

3. Examine the bit descriptions in GPIOx_BSRR register: which bit would you want to set to clear the fourth bit in the ODR?
20

4. Perform the following bitwise operations:
• 0xAD | 0xC7 = EF

• 0xAD & 0xC7 = 85
• 0xAD & ~(0xC7) = 28
• 0xAD ^0xC7 = 6A
5. How would you clear the 5th and 6th bits in a register while leaving the other's alone?

Bitwise and with 0xCF.

6. What is the maximum speed the STM32F072R8 GPIO pins can handle in the lowest speed setting?
• Use the chip datasheet: lab section 1.4.1 gives a hint to the location. You'll want to search the I/O AC characteristics table. You will also need to view the OSPEEDR settings to find the bit pattern indicating the slowest speed.

1000 kHz
7. 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_TIM1EN_Pos
• DMA1
RCC_AHBENR_DMAEN
• I2C1

RCC_APB1ENR_I2C1EN