PostLab Lab 1

1. What are the GPIO control registers that the lab mentions? Briefly describe each of their functions.
MODER- Setting this controls what the pins in those positions will do, there are 4 different states for these pins and each pin is represented by two bits in the moder register, which is 32 bits long so it can contain the information for up to 16 pins. The 4 states are as follows:
00: Input mode (reset state)
01: General purpose output mode
10: Alternate function mode
11: Analog mode
OTYPER- This register selects the output mode we want for each pin.
OSPEEDR- This sets the speed at which the pin operates to cause lower power possibilities.
PUPDR – This connects the pins to internal pull-up or pull-down resistors.
IDR – sets the pins to be read only.
ODR – Sets the logical state of this specific pin.
BSRR – This sets the bits for specific output pins and is read only. It allows for faster modification of the output pins. This can simply only modify the specific output pin/bit that you want to affect.

LCKR – This locks a specific pin and the only way to unlock it is to follow a specific set of commands or to have the whole board reset.

These next two work together they are alternate functions low a high. It lets you set alternate functions to the different pins, and it lets each have 4 alternate functions so we need 64 bits which is why we need a low and high for this register type.
AFRL – (above)
AFRH – (above)

BRR – this resets the corresponding ODx bits but just pertains to the lower half. It is a write only register and we cannot read from it.

2. What values would you want to write to the bits controlling a pin in the GPIOx_MODER register to set it to analog mode?

You would need to write 11 to the bits corresponding to that GPIOx_MODER position.

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

We will want to set bit 19 to 1 if counting starts with 1. Otherwise, we would want to set bit 20 to 1 if we started counting from 0.

4. Perform the following bitwise operations:
• 0xAD | 0xC7 =
10101101 (or)
11000111 =
11101111

• 0xAD & 0xC7 =

10101101 (and)
11000111 =
10000101

• 0xAD & ~(0xC7) =
10101101 (and)
00111000 (not 11000111) =
00101000

• 0xAD ^0xC7 =
10101101 (xor)
11000111 =
01101010

5. How would you clear the 5th and 6th bits in a register while leaving the other's alone?

Register->reg &= ~((1 << 5) | (1 << 6));

6. What is the maximum speed the STM32F072R8 GPIO pins can handle in the lowest speed setting?
Low-speed external user clock generated from an external source. In bypass mode the LSE oscillator is switched off and the input pin is a standard GPIO, the max frequency for this pin is 1000 KHz.

However, for the OSPEEDR I/O AC characteristics then the max frequencies are shown below:
When CL = 50 pF, VDDIOx ≥ 2 V
- The max frequency is 2 MHz

When CL = 50 pF, VDDIOx < 2 V
- The max frequency is 1 MHz

7. What RCC register would you manipulate to enable the following peripherals: (use the comments next to the bit defines for better peripheral descriptions)
//look in header file
• TIM1 (TIMER1)
RCC->APB2ENR |= RCC_APB2ENR_TIM1EN ;

• DMA1
RCC->AHBENR |= RCC_AHBENR_DMAEN;

• I2C1

RCC->APB1ENR |= RCC_APB1ENR_I2C1EN;