Post Lab 1

1. Explanation of GPIO Control Registers

- RCC_AHBENR: Used for enabling peripheral clocks to GPIOs, ensuring that the GPIO ports are powered.
- GPIOx_MODER: Sets the mode of each GPIO pin as either input, output, alternate function, or analog.
- GPIOx_OTYPER: Determines the output type, such as push-pull or open-drain.
- GPIOx_OSPEEDR: Sets the speed at which a GPIO pin operates, with options including low, medium, high, or very high speeds.
- GPIOx_PUPDR: Configures pull-up or pull-down resistors for each pin to ensure proper logic levels when the pins are floating.
- GPIOx_IDR: The input data register, which holds the current input values read from the GPIO pins.
- GPIOx_ODR: The output data register, where data to be output on the GPIO pins is written.

2. Setting a Pin to Analog Mode

To configure a pin in analog mode, write '11' to the corresponding bits in the GPIOx_MODER register.

3. Clearing a Bit Using GPIOx_BSRR

To clear the fourth bit in the ODR, set bit 19 in the GPIOx_BSRR register.

4. Bitwise Operations

- 0xAD | 0xC7 = 0xEF
- 0xAD & 0xC7 = 0x85
- 0xAD & ~(0xC7) = 0x28
- 0xAD ^ 0xC7 = 0x6A

5. Clearing Specific Bits in a Register

To clear the 5th and 6th bits in a register while leaving others unchanged, use the mask 0x60 and apply the bitwise AND NOT operation: GPIOC->MODER &= ~0x60.

6. Maximum Speed in the Lowest Speed Setting

At the lowest speed setting, the maximum frequency for STM32F072R8 GPIO pins is 2 MHz, provided V_DDIOx >= 2V.

7. Enabling Peripherals in RCC Register

- TIM1 (TIMER1): RCC->APB1ENR |= RCC_APB1ENR_TIM1EN
- DMA1: RCC->AHBENR |= RCC_AHBENR_DMA1EN
- I2C1: I2C1->CR1 = I2C_CR1_PE