Jaxon Parker
ECE 5780
1/30/24

Postlab 01 (Intro/GPIO)

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

    - GPIO Port Mode Register (GPIOx_MODER): Configures pin modes (input, output, alternate function, analog).
    - GPIO port output type register (GPIOx_OTYPER): Selects the output mode for each pin (open-drain or push-pull).
    - GPIO port output speed register (GPIOx_OSPEEDR): Sets the speed mode for GPIO pins (low, medium, or high).
    - GPIO port pull-up/pull-down register (GPIOx_PUPDR): Connects internal pull-up or pull-down resistors to a pin.
    - GPIO port input data register (GPIOx_IDR): Read-only register; reports the logical state of each pin.
    - GPIO port output data register (GPIOx_ODR): Sets the logical state of configured output pins.
    - GPIO port bit set/reset register (GPIOx_BSRR): Write-only register; quickly sets or clears bits in the output register.
    - GPIO port configuration lock register (GPIOx_LCKR): Locks other configuration registers for a pin to prevent accidental changes.
    - GPIO alternate function low/high registers (GPIOx_AFRL/GPIOx_AFRH): Configures alternate functions for each pin.
    - GPIO port bit reset register (GPIOx_BRR): Similar to BSRR but dedicated to clearing bits.

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?

You would want to write the values 11 to the bits controlling a pin in GPIOx_MODER resister to set it to analog mode. (11: Analog Mode)

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 would want to set bit 19 to 1 to clear the fourth bit in the ODR. This is assuming counting starts with 1. If counting started with 0 then we would set bit 20 to 1.

4. Perform the following bitwise operations:

    - 0xAD | 0xC7 = 0xEF
    - 0xAD & 0xC7 = 0x85

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

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

The following example with the MODER register shows how to clear the 5th and 6th bits while leaving the other's alone. (GPIOC->MODER &= ~((1 << 5) | (1 << 6));)

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

The maximum speed the STM32F072R8 GPIO pins can handle in the lowest speed setting is 1000 kHz. This is from Table 38. Low-speed external user clock characteristics where it talks about the user external clock source frequency.

From the Table 55 I/O AC characteristics table we know that the maximum frequency that the lowest speed can handle is 2 MHz for the condition CL = 50 pF, VDDIOx ≥ 2 V. If the condition is then changed to CL = 50 pF, VDDIOx < 2 V, the maximum frequency that the lowest speed can handle is then 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)

- TIM1 (TIMER1): RCC-> APB2ENR |= RCC_APB2ENR_TIM1EN;
- DMA1: RCC-> AHBENR |= RCC_AHBENR_DMA1EN;
- I2C1: RCC-> APB1ENR |= RCC_APB1ENR_I2C1EN;