

## ECE2049: Homework 3

---

**Due:** Thursday, 9 June 2020 by noon EDT

**Submission notes:**

- For full credit, please show your work and denote your answers with a circle or a box.
  - Always write or draw your diagrams neatly! We cannot be expected to GUESS what you meant to write! Some problems (such as those involving code) must be typed to be graded—the others may be handwritten (neatly!) or typed.
  - Points for each problem are as indicated. Some portions of problems are marked as “BONUS,” which count as extra credit.
1. (5 pts) Consider the following code example, which configures 3 digital I/O pins on the MSP430F5529.

```
#include <msp430.h>
void configure_io(void)
{
    P2SEL &= ~(BIT5|BIT4|BIT0);
    P2DIR &= ~(BIT5|BIT4);
    P2DIR |= (BIT0);
}
```

*Note:* this example uses compound operators in C like `&=` and `|=`. These are shortcuts for assigning a variable to itself, eg. `P2DIR |= (BIT0)` is the same as writing `P2DIR = P2DIR | (BIT0);`

- a. Which I/O pins does this function configure? Express your answer as set of digital I/O pin names, ie. P1.2, P4.5, etc.
- b. Based on the code in `configure_io()`, which I/O pins are configured as inputs, and which pins are configured as outputs? Explain your reasoning.  
To answer this, you will need to work out which bits of the registers are set to 1 or 0—see PDF pages 7-14 of the notes from lecture 5 part 2 for details.

**(continued on the next page)**

- 2. BONUS (2 pts): Fun with memory-mapped peripherals:** Say you are using a different microprocessor that exposes the memory bus so that you can add new memory-mapped peripherals to it. Assume that you attach your peripheral device and that it has one 16-bit value that the CPU can read at address 0x1104.

How would you write code to read the value at this address? Like all register definitions in C, you can do this with a single `#define` statement. Complete the definition below, which includes an example of how the register should be used.

```
#define MY_REG    (/* Fill in your definition here! */)

void main(void)
{
    int val;
    val = MY_REG; // Read the value of the peripheral at address 0x1104

    // . . .
}
```