Homework 6: Answers

Profs. John McLeod & Arash Reyhani

ECE3375, Winter 2022

Please note that the particular homework assignment that you answered will have the same options as listed here, but the order of these options may have been randomized for each quiz. So your option (b) may instead be written here as option (c), etc. Please read the questions and options in full.
The answers to these questions are all found in the course notes, textbook, and video lessons.

1. Problem: ARM GPIO ports have a data register at the base address (`[base]`), and a control register at 4 bytes above the base address (`[base+0x4]`). These ports can act as input or output, as configured in software.
Which of the following describes how to configure an ARM GPIO port?

(a). The pins in the port are individually configured for input by writing 0 to the corresponding control register bit, or for output by writing 1 to the corresponding register bit.

(b). The pins in the port are individually configured for input by writing 1 to the corresponding control register bit, or for output by writing 0 to the corresponding register bit.

(c). The entire port can be configured for input by writing 0 to the control register, or for output by writing 1 to the control register.

(d). The port acts as an input port whenever software attempts to read from that port, and acts as an output port whenever software attempts to write to that port. register.

Answer: Option (a).

2. Problem: You have an 8-bit piece of data stored in the variable `my_data`. You wish to clear every bit to zero except bits 1, 2, and 5: from `my_data = 0bxxxxxxxx` to `my_data = 0b00x00xx0`. This can be done with a bit mask:

```
my_data &= my_mask;
```

What is the correct decimal value for `my_mask`?

Answer: The appropriate mask is $2^5 + 2^2 + 2^1 = 38_{10}$.

3. Problem: Consider the following code.

```
// main code here
int x = 42;
int bit_mask = 19;
x |= bit_mask;
```

What is the final decimal value in `x`?

Answer: The initial value in `x` is `0b0010 1010`. The mask is `0b0001 0011`, so the bitwise-or sets bits 4, 2, and 1, while leaving all other bits the same. The value in `x` is then `0b0011 1011`, or 59_{10}.

4. Problem: What does the following code do?

```
*((unsigned int*)(0xFFAA1234)) = 0xFF100101;
```

(a). Writes the value `0xFF100101` to the memory address `0xFFAA1234`.

(b). Writes the value `0xFFAA1234` to the memory address `0xFF100101`.

(c). Applies the bit mask `0xFF100101` to the data stored at memory address `0xFFAA1234`.

(d). Reads the contents of memory address `0xFF100101` and writes that data to the memory address `0xFFAA1234`.

Answer: Option (a). The value `0xFFAA1234` is used as a pointer, so it is the memory address. The operation is assignment, without any bitwise logic, so the value `0xFF100101` is not a mask, rather it is an assigned value.

5. Problem: What does the following code do?

```
gpio_a->ctrl_reg = *((char *)(0x10AA));
```

(a). Writes the contents of the byte stored at memory address `0x10AA` to the control register of GPIO port A, this will only set the lowest 8 bits in the control register.

(b). Defines the variable `ctrl_reg` of the structure `gpio_a` as a pointer to a character.

(c). Tries to write data to the GPIO port, but generates a compiler error because the memory address is misaligned.

(d). Writes the value `0x10AA` to the control register of GPIO port A

Answer: Option (a). The value `0x10AA` is used as a pointer, so it is a memory address. The memory address is not evenly divisible by 4, hence it is defined as a byte (here obtusely referred to as a `char`, just to make your life difficult).