Andrea Smith

5423096

EE2361 – Lab 1 Report

1. In our bit-shifting algorithm, what is the purpose of the line of line of code: LATB |= 0x8000; that occurs immediately after the shift? What would happen if this line was not included?

The purpose of the logical operator OR is to set a specific bit to one while leaving the others unchanged. In the context of this lab, the line LATB |= 0x8000 is used to set the most significant bit to one after it has been shifted, which then will be reflected by the LED turning on. Without this, all four LED's would light up in order and then turn off in order instead of lighting up one at a time.

2. One important bitwise operation not discussed in the lab is bitwise negation (inversion). This can be performed using the "exclusive-or" operation (implemented with "^" in C). Suppose I want to invert the most significant bit in a byte of data but leave the other seven bits unchanged. Write a C function that takes a byte of data as an argument and uses a bitmask, and an exclusive-or to invert the most significant bit of that data.

```
short int exclusivemask(short int bit)
{
        return(bit^10000000)
}
```

3. What was the error that caused the delay function to hang? List two possible solutions to this problem.

The error was that there was a bit overflow in curCount because curCount was an int, which can only contain 16 bits, or a max of 2^{15} = 32,767, which is less than 40,000. One solution is to change the data type to a long. Another solution would be to let curCount go up to 20,000 and run the loop twice, which is the equivalent of letting curCount go up to 40,000.

4. Explain why the LED turns on when the value of the pin it is connected to is set LOW and off when it is set HIGH. Why isn’t it the other way around?

The LED turns on when the pin is low because the LED’s are connected to power instead of ground.