Bit Manipulation

Bit manipulation is used to set up configuration registers and change the state of output pins to control devices connected to the microcontroller. The register we write to and the bits we change determint the functionality (refer to the datasheet for your chip).

Writing a register at once

Setting a group of bits using hex notation

```
REGISTER = 0xFF; //all high
REGISTER = 0xF0; //upper nibble high, lower nibble low
REGISTER = 0x00; //all low
REGISTER = 0xCA; //HHLL HLHL, 8+4=12=0xC, 8+2=10=A
```

Setting one bit high

```
REGISTER = 0x01; //R0 high, all other bits low, 0000 0001, LLLL LLLH
REGISTER = 0x04; //R2 high, all other bits low  0000 0100, LLLL LHLL
```

We have to use hex becuase there is no standard way of represeting binary in C, but there is an easier way...

Bit shift notation

Bit shifting is easier and more clear.

Let's do the same thing as above.

```
REGISTER = (1<<0); //R0 high, all other bits low, 0000 0001, LLLL LLLH
REGISTER = (1<<2); //R2 high, all other bits low  0000 0100, LLLL LHLL
```

Bit shifting is even more useful when we want to set multiple bits. We do this with the bitwise OR opeator which is "|" (the pipe key on the keyboard)

```
REGISTER = (1<<0)|(1<<2); //R0 and R2 high, all other bits low, 0000 0101
REGISTER = (1<<0)|(1<<2)|(1<<5)|(1<<7); //adding more bits is easy!
```

Note that above we are setting all the unspecified bits to '0'.

Setting a bit

Setting a bit means making it go to 1 (previous value doesn't matter).

It is possible to set one bit and leave the rest alone. This is very useful for controlling devices that are connected to your microcontroller one pin at a time.

To set one bit high we do a bitwise OR of current value and the bits we want to set and then write the result back to the register

Set R2 while leaving all the other bits as they are.

```
REGISTER = REGISTER | (1<<2); //xxxx xx1x (x means value unchanged)
```

This notation can be condensed using the bitwise OR assignment operator "|=".

```
REGISTER |= (1<<2);
```

You can set multiple bits and leave the rest alone; just like before but with "|=" insted of "='

```
REGISTER \= (1<<0)|(1<<2); //R0 and R2 high, others left alone, xxxx x1x1
```

Clearing a bit

Clearing a bit means making it go to 0 (previous value doesn't matter).

Clearing a bit is a little more tricky. We have to us the bitwise AND assignment operator "&=", but we also need the bitwise complement of the shifted expression (we're making 0's where we want the bits to turn off instead of 1's where we want the bits to turn on).

This is done with the bitwise NOT operator "~"

```
REGISTER = REGISTER & ~(1<<2); //xxxx xx0x
```

Once again, it can be shortened to save typing.

```
REGISTER &= ~(1<<2); //xxxx xx0x
```

The combination is a little more complicated (but not much). Just add extra brackets and the "~".

```
REGISTER &= ~((1<<0)|(1<<2)); //xxxx x0x0
```

Toggling a bit

Toggling is flipping a bit from 0 to 1 or from 1 to 0 (the previous value DOES matter this time, but we dont have to know it).

It's useful for lights or switching the direction of things

This is pretty much the same as setting a bit, but we use the exclusive or (XOR) assigmnent operator "^=" (carat on the keyboard), and it is shortened in the same way as before.

```
REGISTER = REGISTER ^ (1<<2); //toggle
REGISTER ^= (1<<2); //shortened
```

Multiple toggles

```
REGISTER ^= (1<<0)|(1<<2);
```

Summary

Use hex to set a whole register or port.

```
REGISTER = 0x00;
REGISTER = 0xFF;
```

Use bit shift notation when only setting a couple of bits. It will make your life much easier!

```
REGISTER = (1<<4)|(1<<5);
```

Use |= to set a bit without changing the others.

```
REGISTER |= (1<<3);
REGISTER |= (1<<2)|(1<<7);
```

Use &= with ~ for clearing bits

```
REGISTER &= ~(1<<3);
REGISTER &= ~((1<<2)|(1<<4));
```

Use ^= to toggle bits

```
REGISTER ^= (1<<5);
REGISTER ^= (1<<1)|(1<<3);
```

Tips

- Comment your code and use the clearest notation possible.
  Don't expect help if your code isn't organised and readable.
- The bits of the registers from the data sheet have names --- USE THEM! eg:

  ```
  //turn on the transmission and reception circuitry for USART0
  UCSR0B = (1<<RXEN0)|(1<<TXEN0);
  ```

- Initialise registers when using them (especially configuration registers),

  ```
  CONFIG_REGISTER = 0x00;
  CONFIG_REGISTER |= (1<<BIT_A)|(1<<BIT_B);
  ```

  or set them using "=" not "|=" on the first write.

  ```
  CONFIG_REGISTER = (1<<BIT_A)|(1<<BIT_B);
  ```

- Search for "bitwise operations" and "bit manipulation on the internet if you're having trouble.