

```
/*
 * xor3_named_bits.c
 *
 * Created: 2/6/2022 4:32:55 PM
 * Author : Jason Tan
 */

#include <avr/io.h>
#include "data.h"

int main(void)
{
    //pointer to PIN5CTRL array of pin configuration registers
    uint8_t* ptr = (uint8_t*)&PORTA.PIN2CTRL;

    //DIR is what configure the port pins as inputs or outputs
    VPORTA_DIR = 0x00; //Configure PORTA pins ( PA7, PA6, PA5) as the inputs
    VPORTD_DIR = 0x80; //Configure (PD7) as output pin

    /*
    PA7 is "C"
    PA6 is "B"
    PA5 is "A"
    PD7 is "F"
    */

    //Configure PA7 - PA5 as input buffers with internal pull up resistors
    for(uint8_t i = 3; i < 8; i++){
        *(ptr + i) |= PORT_PULLUPEN_bm;
    }

    Named_bits data; //Data to be processed

    uint8_t temp_in = 0xFF; //Initialize the LED to be off
    uint8_t temp_out = 0x00; //Initialize the temp output to be all 0x00

    while (1)
    {
        uint8_t C = PORTA_IN & PIN7_bm;
        uint8_t B = (PORTA_IN & PIN6_bm) << 1;
        uint8_t A = (PORTA_IN & PIN5_bm) << 2;

        data.byte = temp_in; //Set that register as the initial output value

        //Canonical sum of products of the 3 input truth table and store that into
        data.byte = (((~C) ^ (~B) ^ (A)) ^ 0xFF ) | (((~C) ^ (B) ^ (~A)) ^ 0xFF ) |
            (((C) ^ (~B) ^ (~A)) ^ 0xFF ) | (((C) ^ (B) ^ (A)) ^ 0xFF );

        //Store that bit result after doing the canonical sum of product
    }
}
```

```
    temp_out = data.byte;

    //To output the result
    PORTD_OUT = temp_out;
}
}
```