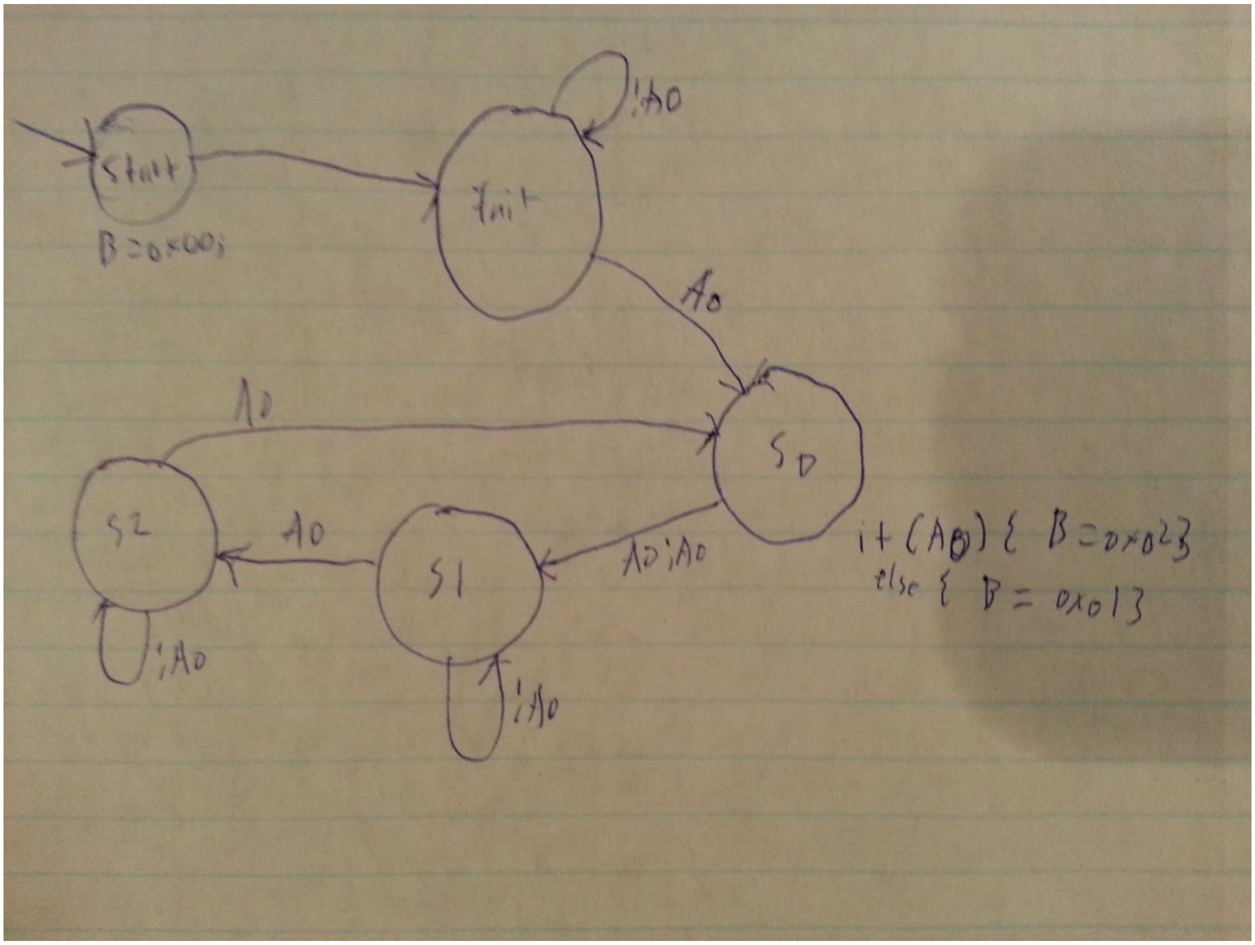
Lab3 Prelab



```
// code for state machine

#include <avr/io.h>

// Bit-access function

unsigned char SetBit(unsigned char x, unsigned char k, unsigned char b) {

        return (b ? x | (0x01 << k) : x & ~(0x01 << k));

}

unsigned char GetBit(unsigned char x, unsigned char k) {

        return ((x & (0x01 << k)) != 0);

}

enum States { Start, Init, s0, s1, s2 } State;
```

```
void Tick()
{
        unsigned char temp2 = 0x02;
        unsigned char temp1 = 0x01;
        unsigned char temp = PINA;
        unsigned char temp3 = PINB;
        switch(State)
        {
                case Start:
                State = Init;
                break;
                // Transitions
                case Init:              // Initial transition
                if ((GetBit(temp, 0) == 1))   // do button press here
                {
                        State = s0;
                }
                else
                {
                        State = Init;
                }
                break;

                case s0:
                if ((GetBit(temp, 0) == 1))
                {
                        State = s1;
                }
```

```
else if((GetBit(temp, 0) == 0))
{
        State = s1;
}
break;

case s1:
if(GetBit(temp, 0) == 1)
{
        State = s1;
}
else
{
        State = s2;
}
break;

case s2:            // Initial transition
if ((GetBit(temp, 0) == 1))   // do button press here
{
        State = s0;
}
else
{
        State = s2;
}
break;
```

```
        default:
        State = Init;
        break;
} // Transitions

switch(State)
{   // State actions
        case Start:
        PORTB = temp1;
        break;

        case Init:
        break;

        case s0:
        if(GetBit(temp3, 0) == 1)
        {
                PORTB = temp2;
        }
        else
        {
                PORTB = temp1;
        }
        break;

        case s1:
        break;
        case s2:
        break;


        default:
        PORTB = temp1;
        break;
    } // State actions
}



int main(void)
{
    DDRA = 0x00; PORTA = 0xFF; // Configure port A's 8 pins as inputs
    DDRB = 0xFF; PORTB = 0x00;


    unsigned char tmpB = 0x01; // intermediate variable used for port updates
    /* Replace with your application code */
    PORTB = tmpB;
    State = Start;
    while (1)
    {
        Tick();
    }
}
```