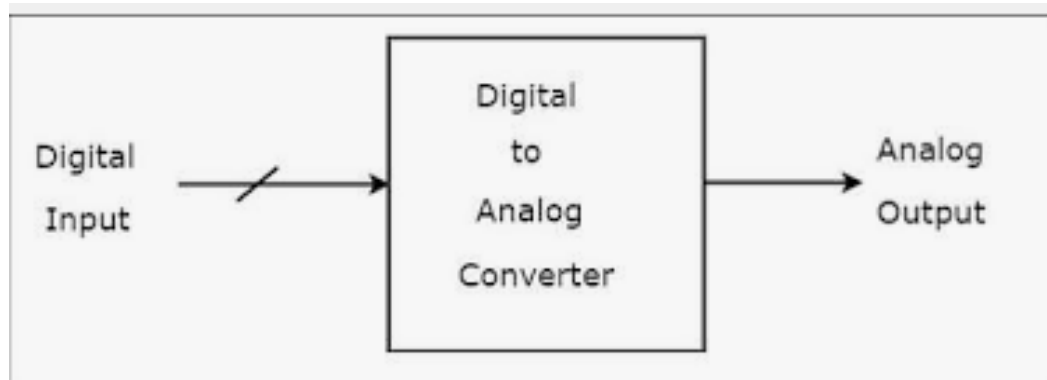


Interfacing DAC with 8051

- Engr. Shahzada Fahim Jan

- DAC (Digital to Analog Converter) is widely used to convert digital pulses to analog signal voltage or current.

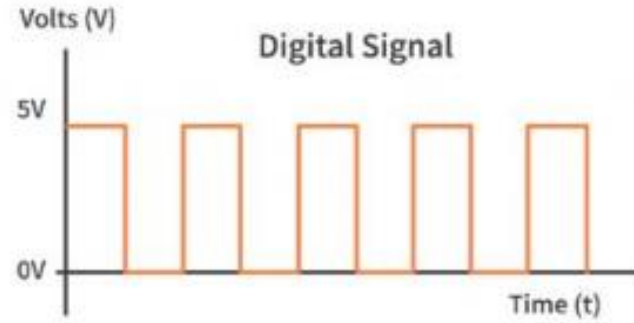
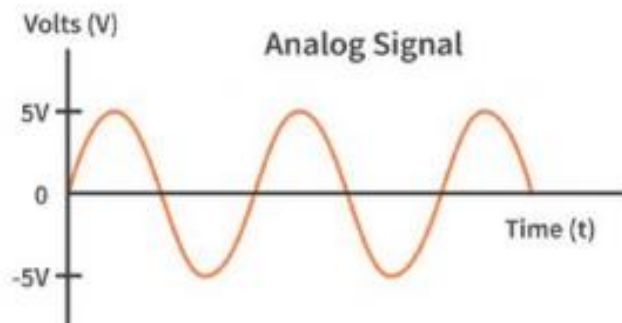


➤ Analog:

Analog signals are a type of signal sent in a continuous wave. These waves can vary in both amplitude and frequency.

➤ Digital:

Digital signals are signals that are represented in discrete values.

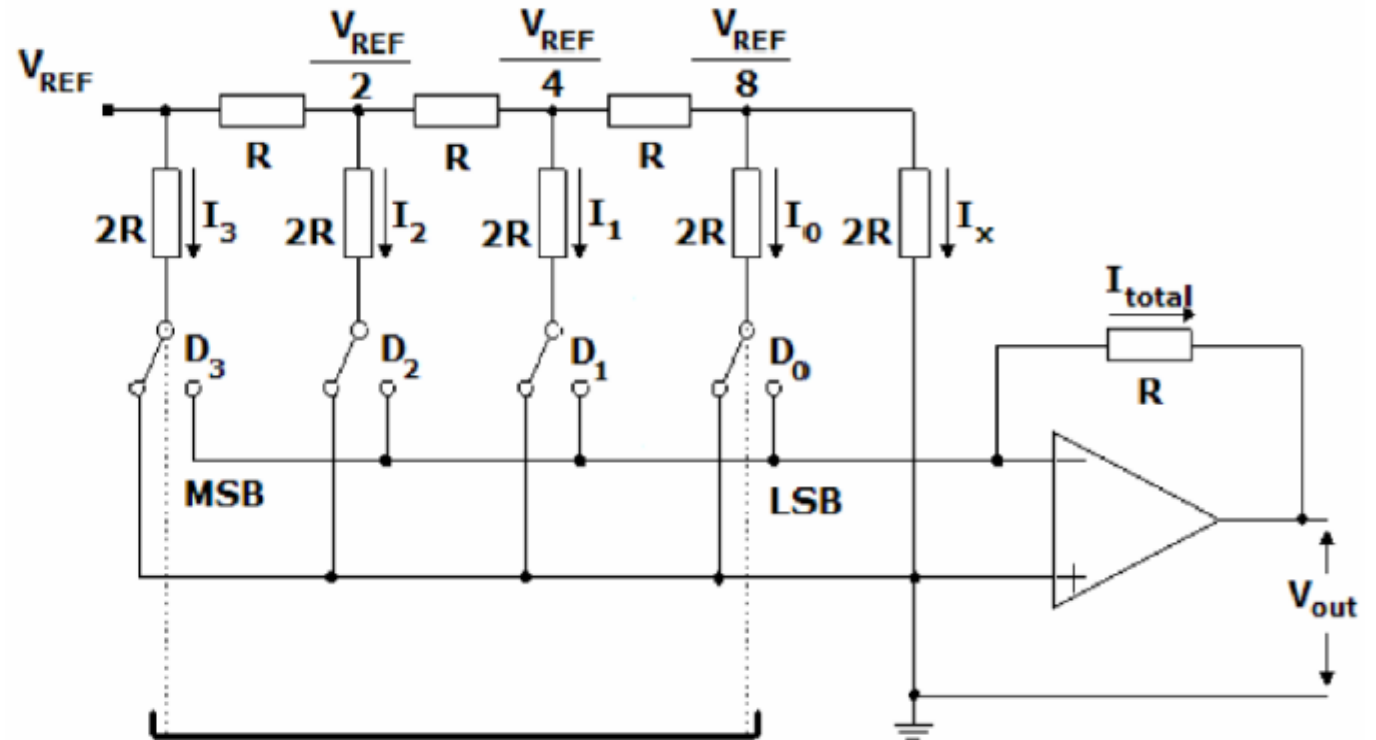


- Resolution:

The amount of changes in the output voltage for every single change of the LSB of the digital input.

R/2R 4-bit DAC Architecture (1/2)

- Uses a repeating cascaded structure of resistor values R and $2R$ to create a binary weighted DAC.
- The $R/2R$ ladder divides down a positive reference voltage by switching individual resistors between a positive reference voltage, V_{REF} , and the analogue ground, generating a current.
- The equivalent resistance between V_{REF} and ground is R .
- An operational amplifier converts this current to Voltage (V_{out})

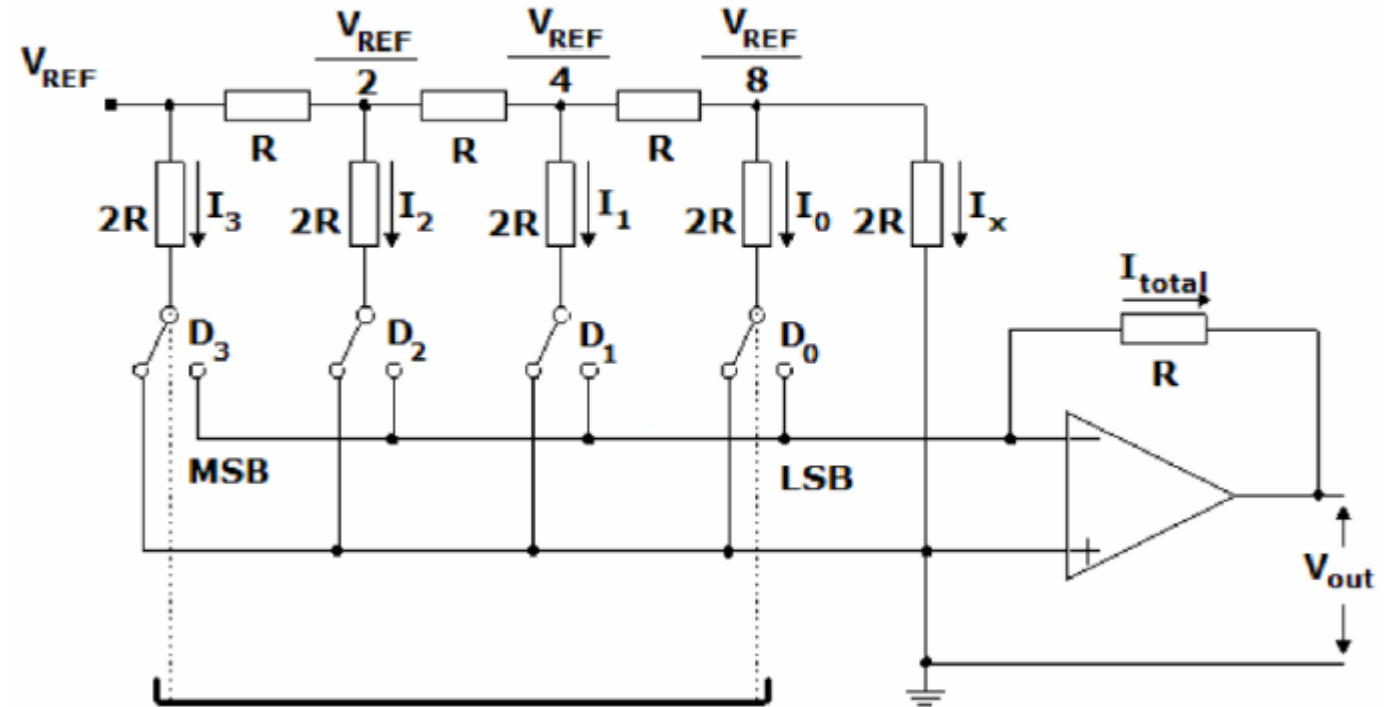


Data bit "Low" -> Switch current to ground

Data bit "high" -> Switch current to negative input of OpAmp

R/2R 4-bit DAC Architecture (2/2)

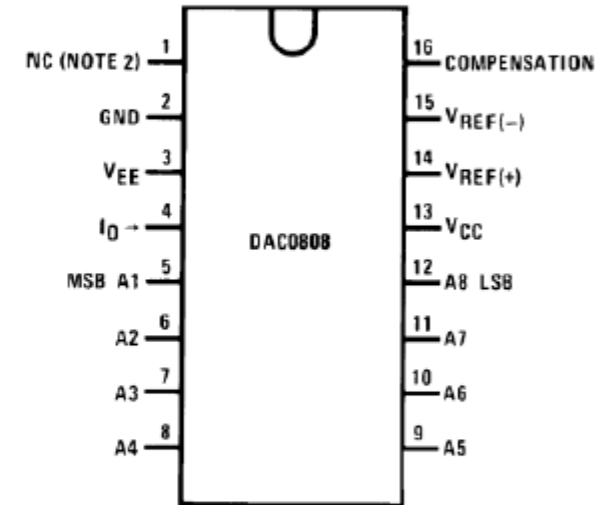
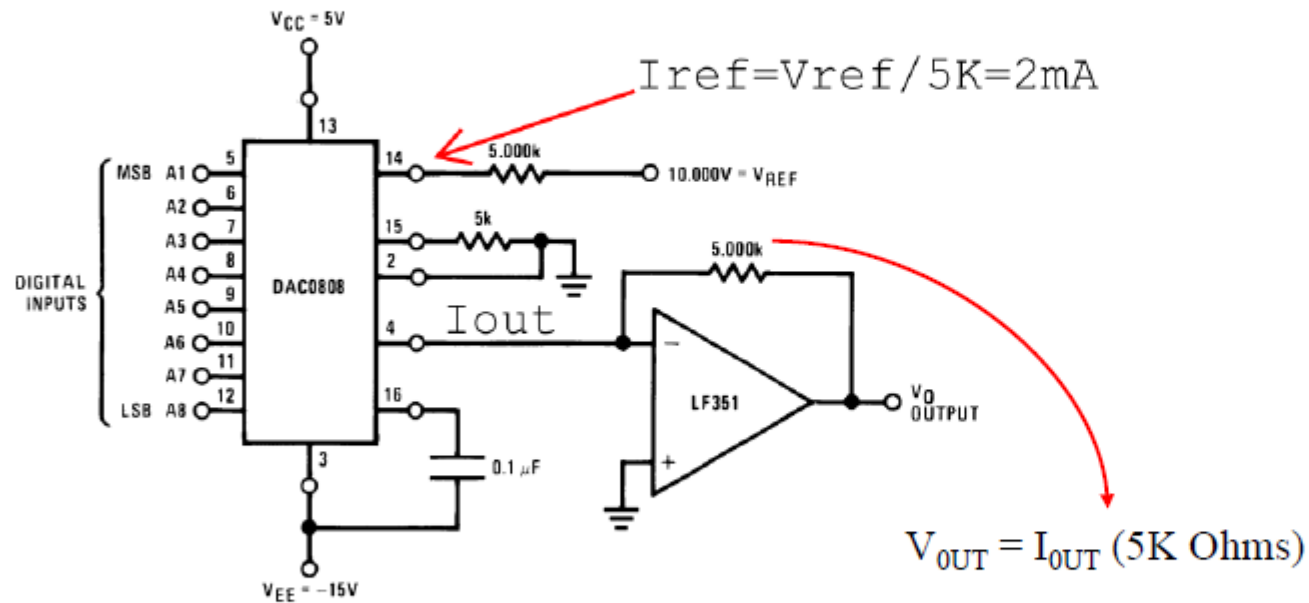
- $V_{out} = \frac{V_{ref}}{2^4} * \text{Value}$
- Example:
 - if Value = 15 & $V_{ref} = 2.5$ Volts.
 - $V_{out} = ?$
- Step size = ?



Data bit "Low" -> Switch current to ground

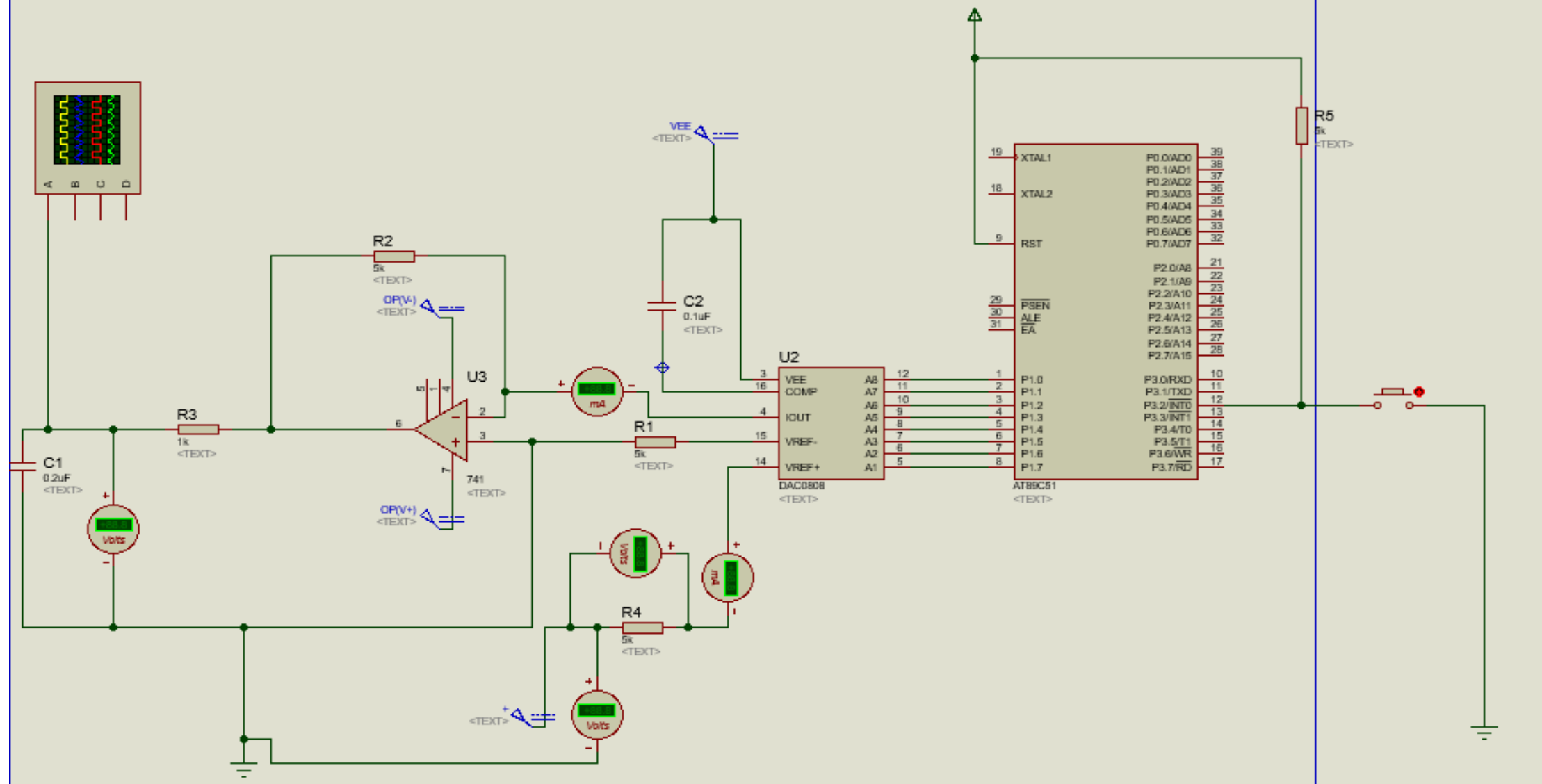
Data bit "high" -> Switch current to negative input of OpAmp

Typical Application (DAC0808)



$$I_{OUT} = I_{ref} (D_7/2 + D_6/4 + D_5/8 + D_4/16 + D_3/32 + D_2/64 + D_1/128 + D_0/256)$$

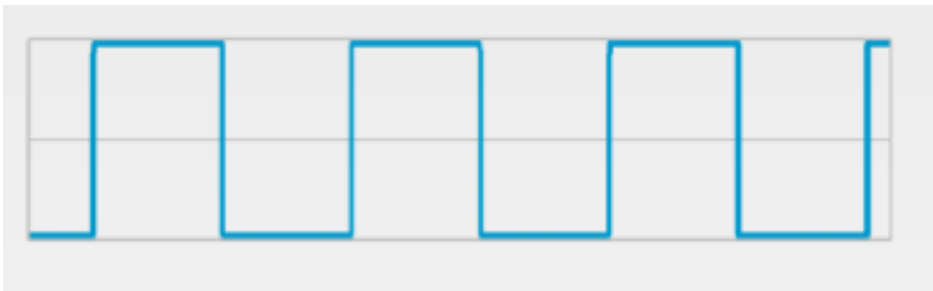
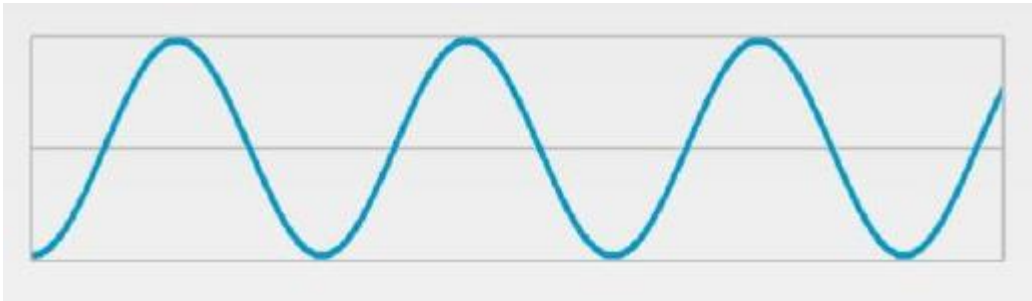
$$V_{OUT} = I_{OUT} (5K \text{ Ohms})$$




```
#include <reg51.h>
#include <stdio.h>
void delay(unsigned int x)
{
    unsigned int i;
    for(i=0;i<x;i++);
}
void sine_wave(){
    int sine[]={128,156,192,226,238,255,238,226,192,128,64,32,6,4,0,16,32,64};
    int i;
    for(i=0;i<18;i++){
        P1=sine[i];
        delay(15);
    }
}
void square(){
    P1 = 0x00;
}
void main(void)
{
    while(1){
        sine_wave();
    }
}
```

TASK

- Generate the following waveform



- You can use a button, when pressed once then generate triangular wave, when pressed again generate the square wave and so on.
Use Port-interrupt for button detection.