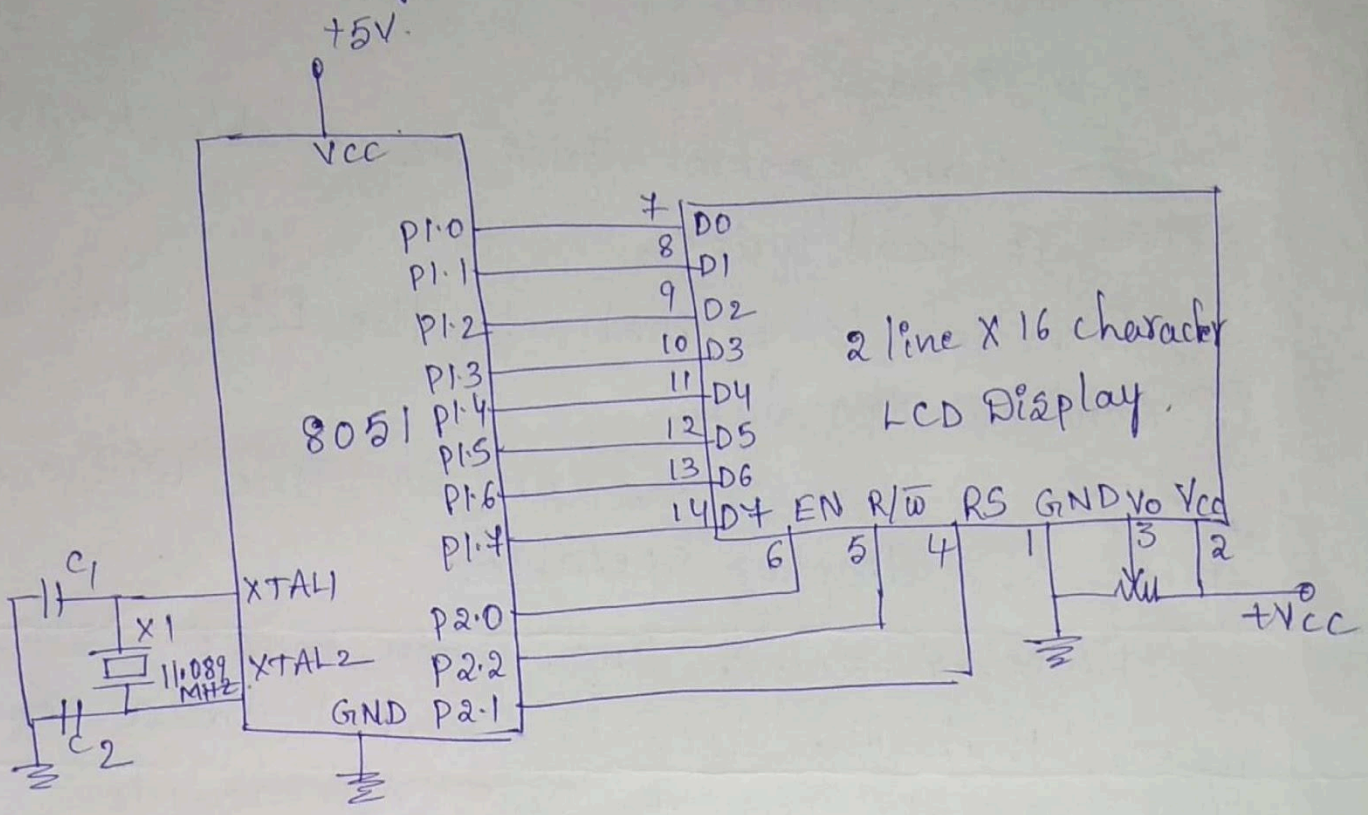
UNIT-III

I/O and Memory Interface.

* LCD Interfacing with 8051:



(1) LCD Pin description:

1. GND —— Ground
2. Vcc —— +5V power supply
3. VEE —— Contrast control power supply.
4. RS —— 0 = Select command register.
   1 = Select data register.
5. R/$\overline{W}$ —— 0 = write; 1 = Read.
6. EN —— Enable.

7-14 D0-D7 —— 8-bit data bus.

(2) Commands or instruction codes of LCD:

01H — clear display screen.

→ LCD instructions allow selecting the display functions, cursor positioning, blinking, selecting character font, number of lines etc.

* → In LCD module, there are 3 control lines & 8-data lines.

→ the three control lines are enable, Register select & Read/write.

→ EN line is to instruct the LCD that uc is sending the data.

→ 1 to 0 transitions on EN line will com -plete the display operation.

→ Register select line, when made low, means that the data is treated as a comm -and or instruction to the LCD module.

→ when RS line is high, the data is a text data to be displayed on the LCD.

→ the read/write line RW, when made low, the information on data bus is written to LCD & when RW line is high, it means that the data is being read from the LCD.

→ there is only one read command, namely Get LCD status.

→ the EN line must go from high to low for LCD to execute the instruction.

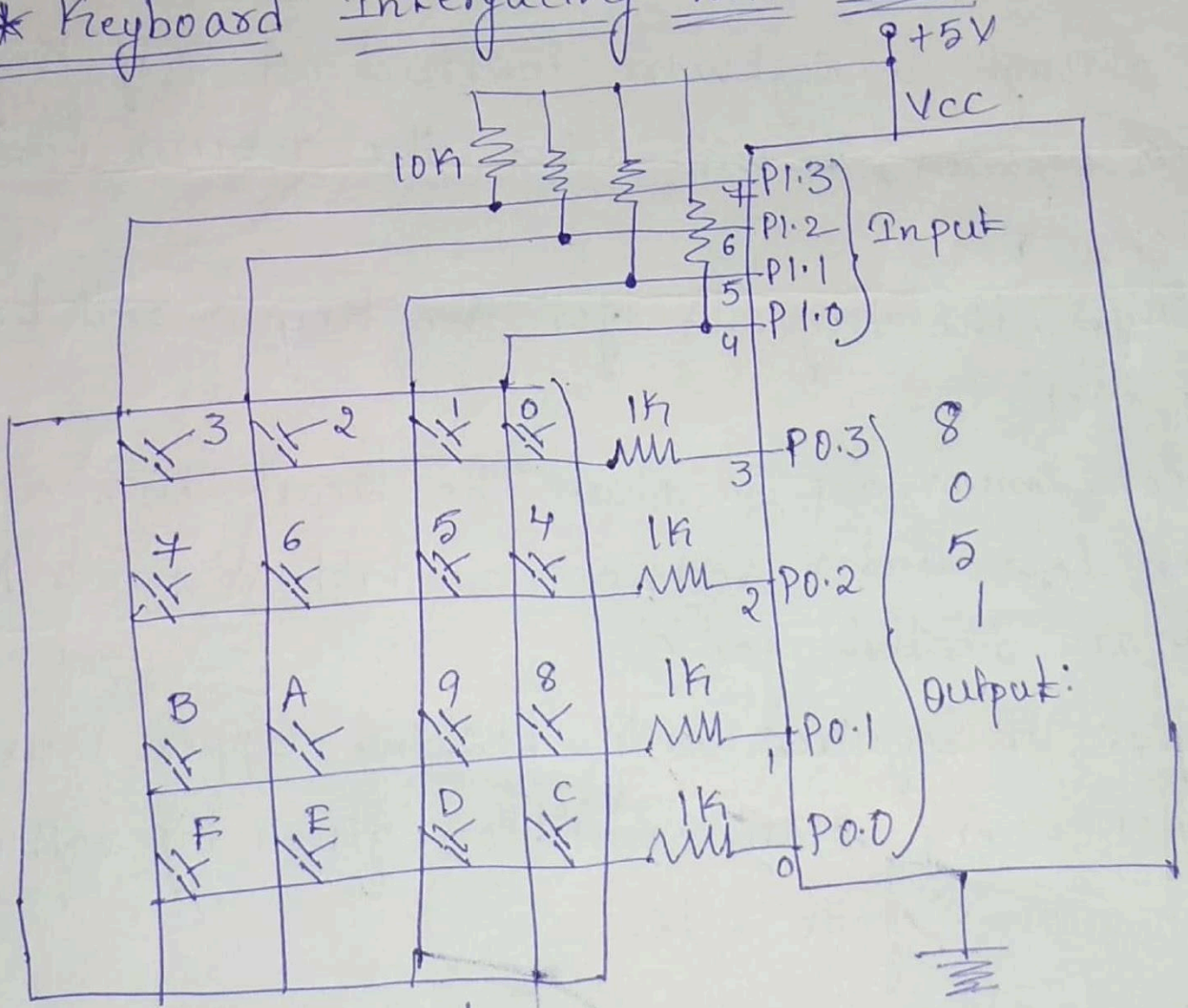
→ Get LCD status command will make bit 7 of data bus high if it is busy in executing the command & will make it low if the LCD is not busy.

→ RS line must be low, while sending any of the commands.

→ For writing the text, set line EN & RS. EN line must go from high to low for the execution of LCD operation.

---

* Keyboard Interfacing with 8051:



0	4	8	C
1	5	9	D
2	6	A	E
3	7	B	F

→ Port '1' pins 4, 5, 6, 7 are pulled high through 10K resistors and are configured as inputs.

→ 1K resistors are connected in each of the rows to pins 3, 2, 1, 0 to avoid any short circuit between these pins in case two keys are pressed simultaneously.

→ Pins 3, 2, 1, 0 are configured as outputs & are used as the scan lines.

→ Pins 7, 6, 5, 4 are the return lines.

→ For example, if a scan line '3' is made low through a software instruction & if key '0' is pressed, it will drive the return line '7' low.

→ Thus, the keycode for the key '0' will be "0111 0111".

→ More important is that the scan lines are driven low under software control & the return lines are tested for '0'.

→ Thus, it requires only testing upper four bits for every scan condition given by software driven lower four bits.

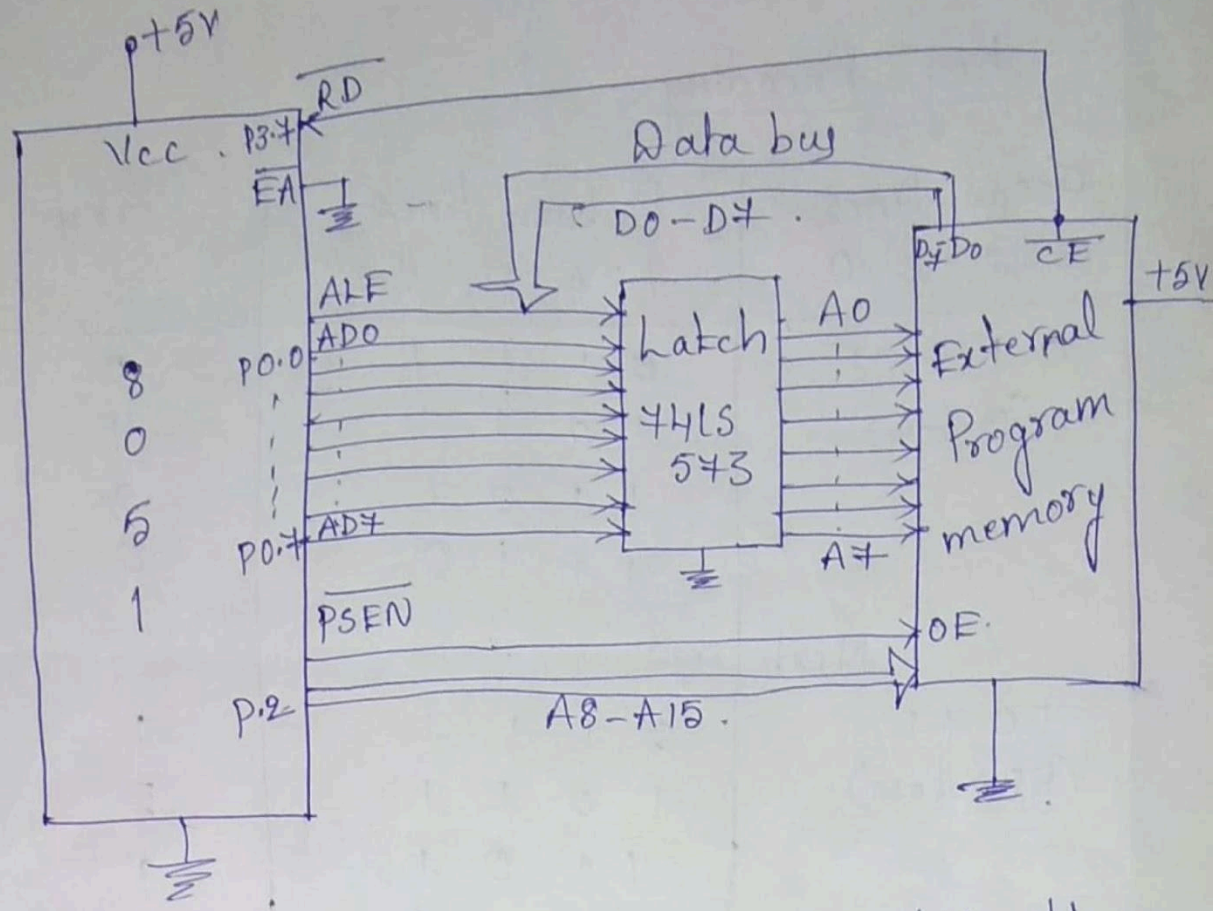
→ For testing any key closure in spite of which key is pressed all the scan lines may be driven low & test for any return line becoming low.

→ Initially, the program register must be loaded with B"1111 0000".

* Key Encoding:

Scan lines 3:2:1:0	Return lines 7:6:5:4	Key
0 1 1 1 (3 low)	0 1 1 1	0
	1 0 1 1	4
	1 1 0 1	8
	1 1 1 0	C
Next row		
1 0 1 1 (RB2 low)	0 1 1 1	1
	1 0 1 1	5
	1 1 0 1	9
	1 1 1 0	D
Next row		
1 1 0 1 (1 low)	0 1 1 1	2
	1 0 1 1	6
	1 1 0 1	A
	1 1 1 0	E
Next row		
1 1 1 0 (0 low)	0 1 1 1	3
	1 0 1 1	7
	1 1 0 1	B
	1 1 1 0	F.

* Interfacing External ROM with 8051:



→ In 8051 ports 0 & 2 provides 16-bit address to access the external memory.

→ P0 is multiplexed address/data bus.

→ By using ALE pin

ALE = 0 → Data path (access data from P0)

ALE = 1 → address path. (access address from P0)

→ O/p signal of 8051 must be connected to ($\overline{PSEN}$) the OE pin of ROM containing the program code.

→ $\overline{EA}$ pin (active low): 1 → pgm is on on-chip ROM

0 → pgm to be access from external pgm memory.

→ $\overline{PSEN}$: 0 → can access external memory.

1 → Cannot access external memory.

→ Latch is a 8-bit latch used for address & data multiplexing.

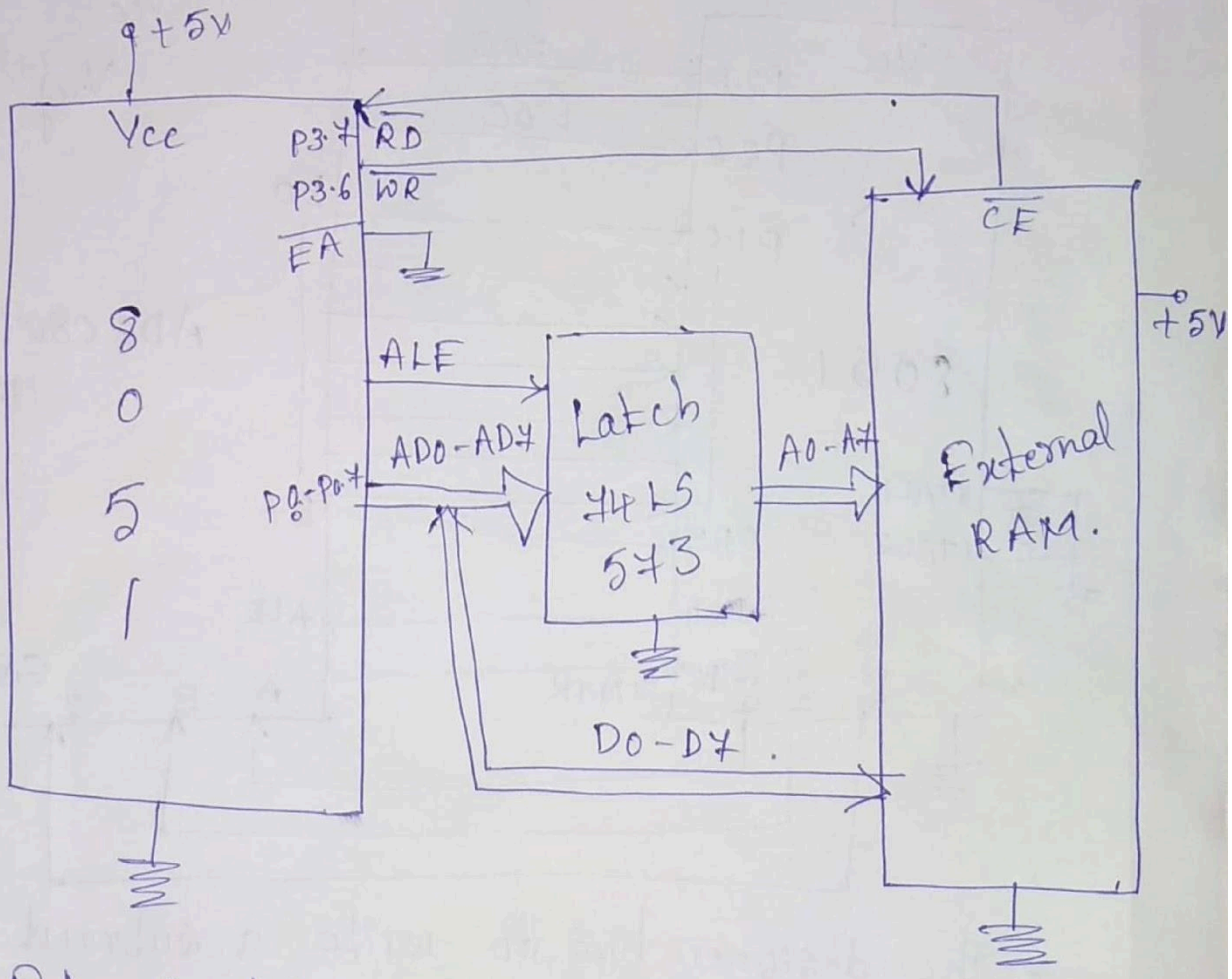
* → External program memory access is possible in 8051 if the $\overline{EA}$ pin is low.

→ It is necessary to latch the lower address byte when the external program byte is read on the data bus.

→ the lower order address byte is latched, as the ALE pulse goes low.

---

* Interfacing of External RAM to 8051:



→ External data memory up to 64K can be accessed dur using a 16-bit data pointer (DPTR).

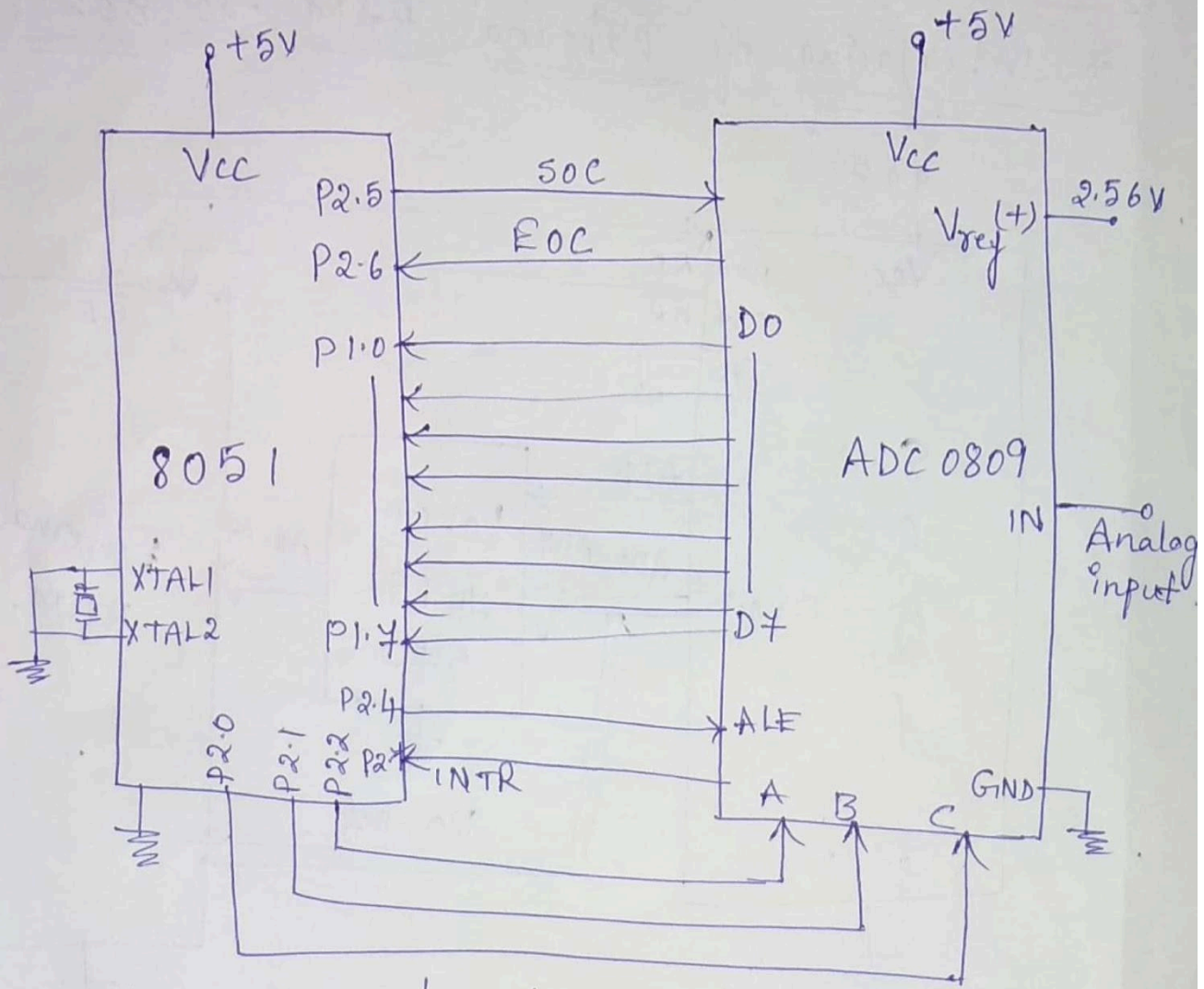
→ If the access is to data memory, then the signals $\overline{RD}$ & $\overline{WR}$ are generated by 8051.

→ During read cycle, $\overline{RD}$ signal goes low & a data byte from the external memory location is loaded onto the data bus.

→ $\overline{WR}$ signal is generated to write this byte into the external data memory.

---

* Interfacing ADC to 8051:



→ The designer has to write a subroutine for the control of ADC operation.

→ the analog to digital converter is treated as an input device by the microcontroller, that sends an initialising signal to ADC to start the analog to digital data conversion process.

→ the SOC signal is a pulse of a specific duration.

→ the process of ADC is a slow process, & the uc has to wait for digital data till the conversion is over.

→ After the conversion is over, the ADC sends EOC signal to inform the uc about it & the result is ready at the o/p buffer of ADC.

→ the time taken by the ADC from the active edge of SOC pulse till the active edge of EOC signal is called the conversion delay of ADC.

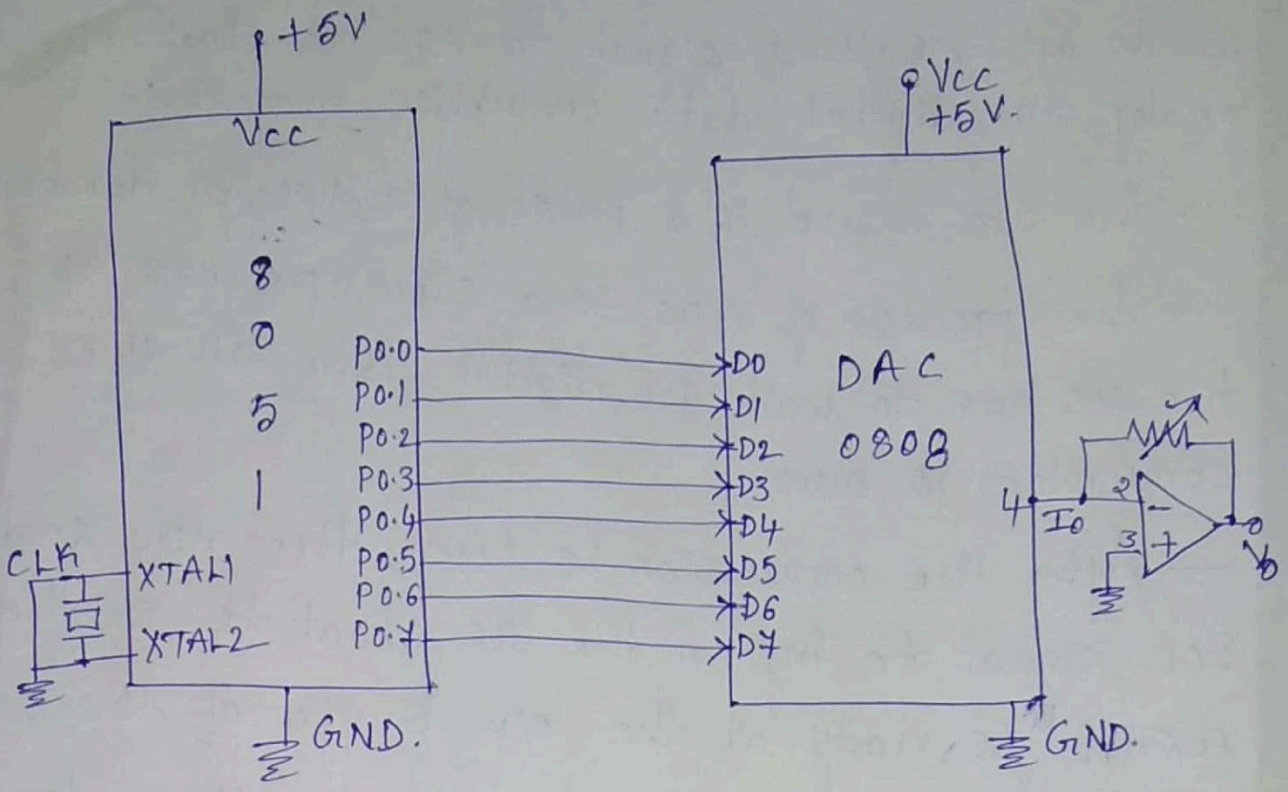
→ these converters initially have a 3:8 analog multiplexer so that at a time 8 different analog inputs can be connected to the chips.

→ Out of these 8 inputs only one can be selected for conversion by using address lines A, B & C.

Analog i/p selected	Address lines C	B	A
I/P 0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
7 6	1	1	0

1 1 1

* Interfacing of DAC to 8051:



→ the DAC convert binary numbers into their equivalent analog voltages.

→ An Operational amplifier is used as a current to voltage converter at the o/p of ~~AD~~ DAC. to convert current o/p to a proportional o/p voltage.

→ An external feedback resistor acts to control the gain.