MODULE 5

8051 INTERFACING

5.1 8255 Introduction:

The 8255 is a 40-pin DIP chip. It has three separately accessible ports. The ports are each 8-bit, and are named A, B, and C. The individual ports of the 8255can be programmed to be input or output.

Port A (PA0 - PA7) - The 8-bit port A can be programmed as all input, or as all out- put, or all bits as bidirectional input/ output.

Port B (PB0 - PB7) - The 8-bit port B can be programmed as all input or as all output. Port B cannot be used as a bidirectional port.

Port C (PC0 - PC7) - This 8-bit port C can be all input or all output. It can also besplit into two parts, CU (upper bits PC4 - PC7) and CL (lower bits PC0 - PC3). Each can be used for input or output.

RD and WR – these two are active-low control pins. The RD and WR signals fromthe 8051 are connected to these inputs.

D0 - D7 (data bus) – These pins are used for data transmission.

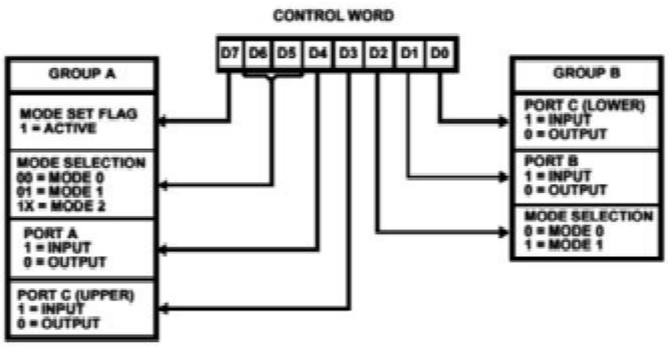
RESET - This is an active-high signal input into the 8255 used to clear the control register.

A0 and A1 – these two pins are used to access the A,B, C and CER as follows.

CS	A1	A0	Port/CWR
0	0	0	A
0	0	1	B
0	1	0	C
0	1	0	CWR
1	x	x	8051 not selected

Mode selection of 8255

These are three modes. The ports of 8255 can be programmed in any of thefollowing modes by CWR register.


CONTROL WORD
D7 D6 D5 D4 D3 D2 D1 D0
GROUP A
MODE SET FLAG 1 = ACTIVE
MODE SELECTION 00 = MODE 0 01 = MODE 1 1X = MODE 2
PORT A 1 = INPUT 0 = OUTPUT
PORT C (UPPER) 1 = INPUT 0 = OUTPUT
GROUP B
PORT C (LOWER) 1 = INPUT 0 = OUTPUT
PORT B 1 = INPUT 0 = OUTPUT
MODE SELECTION 0 = MODE 0 1 = MODE 1

Mode 0 : simple IO mode, in this any of the ports A, B, CL, and CU can be programmed as input or output

Mode 1: in this mode, ports A and B can be used as in put or output ports with handshaking capabilities. Handshaking signals provided by the bits of port C.

Mode 2 : In this mode, port A can be used as a bidirectional I/O port with handshaking capabilities whose signals are provided by port C.

Control word of PPI 8255A

Ex: Obtain the control word for the following configuration of the ports of 8255A.Port A – as input port,

Mode for Port A – mode 1

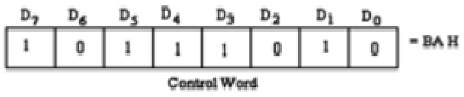
Port B – input port,

Mode for Port B – mode 0,

Port CLower – output port

The remaining pins PC6 and PC7 of port C are to be used input pins.

Sol: The control word for this case is given as



D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
1	0	1	1	1	0	1	0	= BA H

Control Word

Ex: Obtain the control word for the following configuration of the ports of 8255A.

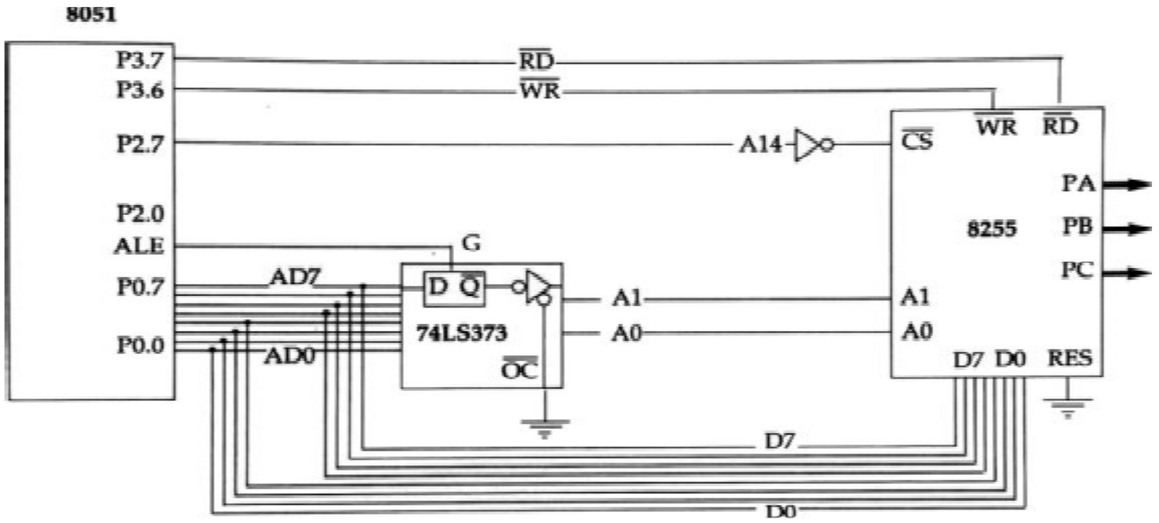
Port A – as bidirectional
Mode for Port A –
mode 2Port B – input
port
Mode for Port B – mode 1

Sol: The control word for this case is given as

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
1	1	X	X	X	1	1	X	= C6H

Control Word

Interfacing of 8255 to 8051



One can program the 8255 chip in any of the three previously stated modes by passing a byte to the chip's control register. The port addresses assigned to ports A, B, C, and CWR must first be located. We refer to this as mapping the IO port. As seen in Fig., the 8255 is connected to an 8051a. Because the IO chip is mapped into memory space, this type of IO chip connection to CPU is known as memory mapped IO. To access the 8255, we thus utilize instructions like MOVX.

Programming example:

4 switches connected to the upper 4 bits of port A (PA4 – PA7). Write a program to transfer status of switches to LEDs which connected to the lower 4 bitsof port B (PB0 – PB4)

CWR for above example –
99H

Address	Port
4000H	Port A
4001H	Port B
4002 H	port C
4003 H	CWR

```
MOV A, 99H
MOV   DPTR,   #4003H  MOVX   A,
        @DPTR
BACK:  MOV  DPTR,  #4000H MOVX
        A, @DPTR SWAP A
MOV DPTR, #4001H
MOV @DPTR, ASJMP BACK
```

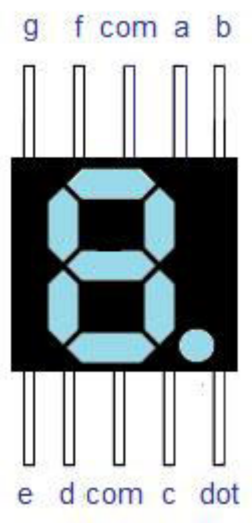
5.2 Interfacing 8051 to 7 segment LED:

This explain the process of connecting a seven-segment LED display to an 8051 microcontroller. Besides showing digits 0 through 9, these displays can also depict characters like A, B, C, H, E, e, F, n, o, t, u, y, and more. Understanding how to interface these displays with microcontrollers is crucial for embedded systems development.

A seven-segment display consists of seven LEDs arranged in the shape of a slightly inclined "8", with an additional LED for displaying a dot character. Different characters are displayed by selectively illuminating specific LEDs. These displays come in two types: common cathode and common anode.

In common cathode displays, all cathodes of the seven LEDs are connected together to a common terminal (often labeled "com"), while their anodes are separate and labeled as "a," "b," "c," "d," "e," "f," "g," and "h" (or dot). Conversely, in common anode displays, all anodes are connected together to a common terminal, and the cathodes are individually designated.

The pin layout and an illustration of a typical seven-segment LED display are depicted in the figure below.



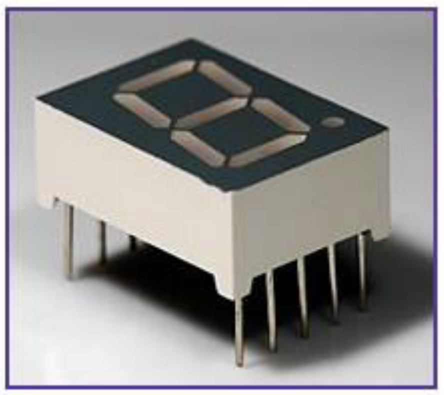


Fig 5.1.17 segment LED display

Digit drive pattern.

The 'a' to 'h' terminals of a seven segment LED display can be configured in various logic combinations to display different characters and digits, which is known as the "digit drive pattern." The table below displays a seven segment display's common digit drive patterns, numbered 0 through 9.

Digit	a	b	C	d	E	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

Interfacing seven segment display to 8051.

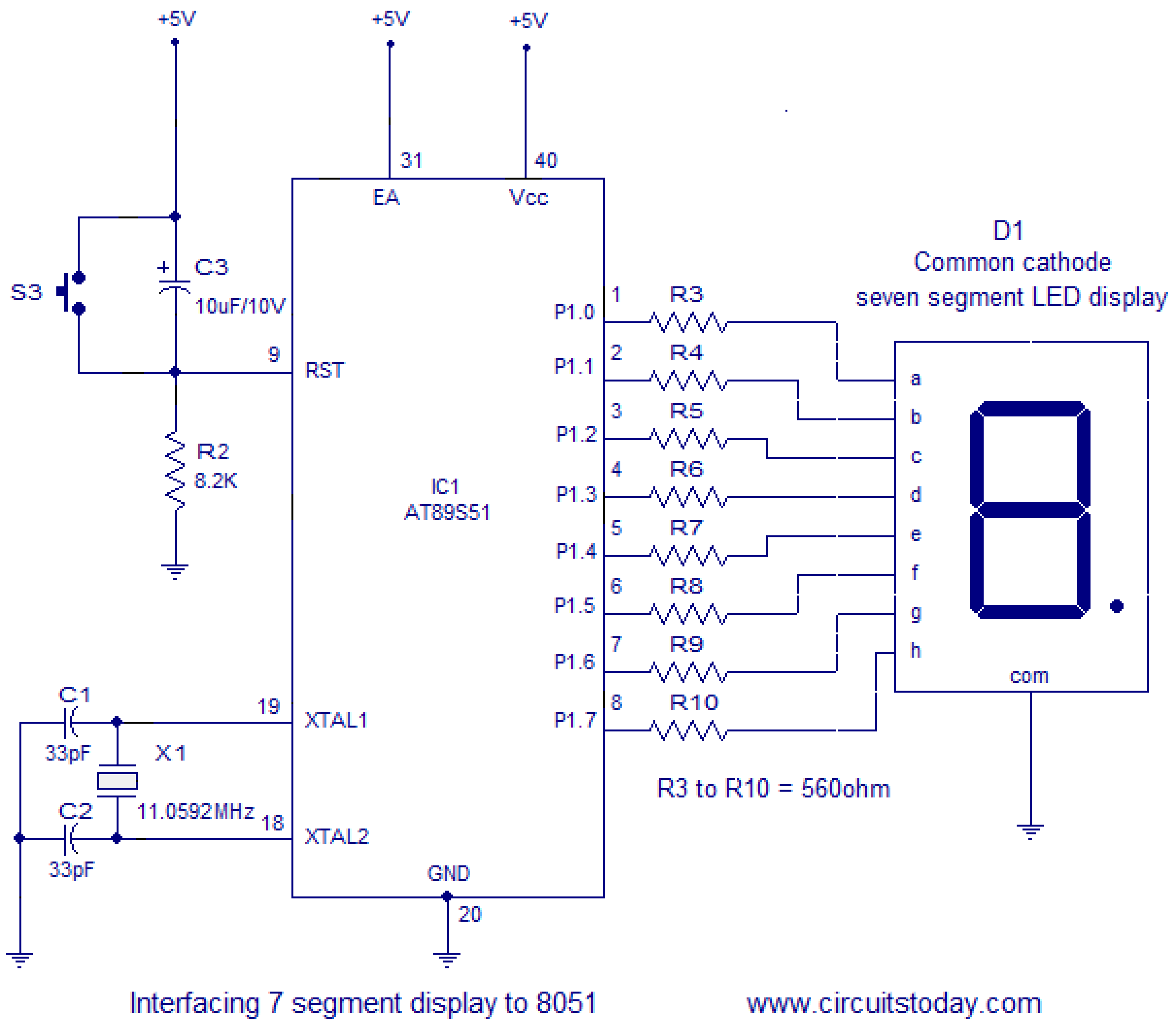


Fig 5.1.2 Interfacing 7 segment display to 8051

A 0 to 9 counter based on an AT89S51 microprocessor is interfaced to a 7 segment LED display in the circuit diagram above to display the count. This simple circuit illustrates two points. Build a simple 0 to 9 up counter with an 8051. More importantly perhaps, how to connect a seven-segment LED display to an 8051 so it can display a certain result. The connection between the common cathode seven segment display D1 and the microprocessor (AT89S51) is depicted in the circuit schematic. R3 through R10 are resistors that limit current. S3 is the reset switch, and R2 and C3 make up the debouncing circuitry. C1, C2, and X1 are components linked to clock circuits. The software component of the project must finish the following assignments.

- Set a delay of, let's say, thirty seconds and create a 0 to 9 counter.
- Using the existing count, create a digit drive design.

• Place the current digits' drive pattern into a port for display. All of the aforementioned responsibilities are fulfilled by the program that is stated below.

5.3 Stepper motor interfacing to 8051

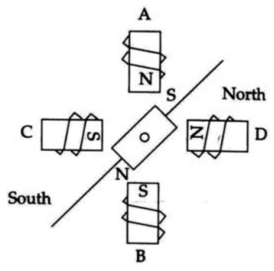
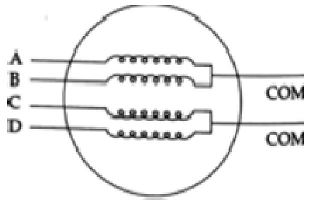


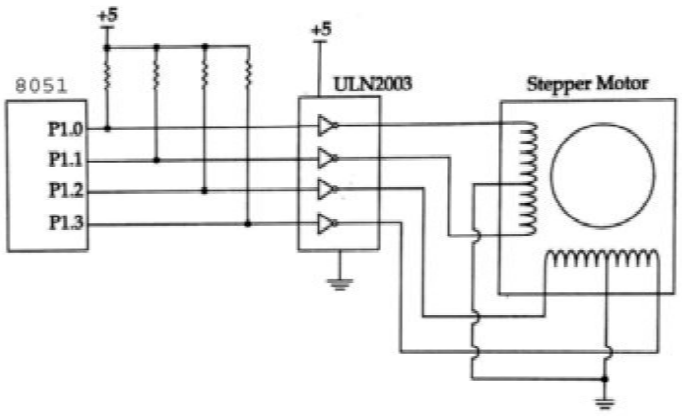
Fig 5.3.1 stepper motors

A stepper motor is a widely used device that translates electrical pulses into mechanical movement. In applications such as disk drives, dot matrix printers, and robotics, the stepper motor is used for position control. Stepper motors commonly have a permanent magnet rotor (shaft) surrounded by a stator sown in figure. There are also other steppers called variable reluctance stepper motors that do not have permanent rotor.

The most common stepper motors have four stator windings that are paired with a center tapped common as shown in figure.



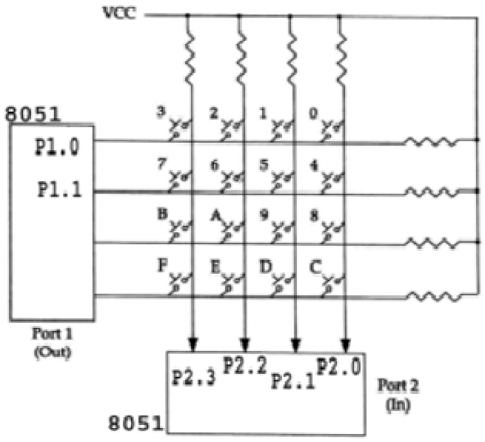
This type of stepper motor is commonly referred to as a four-phase stepper motor. The center tap allows a change of current direction in each of two coils when a winding is grounded, thereby resulting in a polarity change of the stator. The stepper motor rotor moves in a fixed repeatable increment, which allows one to move it to an accurate position. The stator poles are determined by the current sent through the wire coils. As the direction of the current is changed, the polarity is also changed causing the reverse motion of the rotor. Here the stepper motor has a total of 6 leads: 4 leads representing the four stator windings and 2 commons for the center-tapped leads. As the sequence of power is applied to each stator winding, the rotor will rotate.



5.4 Keyboard Interface

Once you add a keyboard for your system, you should allow the user to input information to the microcontroller in real time.

In general keyboards are organized in a matrix of rows and columns. Figure shows a 4 x 4 matrix connected to two ports, two sides of matrix are connected to VCC through resisters while 3rd side is connected to 8051 port 2 which is configured as input port, and last side connected to port 1 which is configured as an output port.



Microcontroller keeps scanning the keyboard, if all inputs are high, that means no key is pressed. If one bit is low that means there is a pressed key. System designer setup a look up table contained ASCII code for each key. In this keyboard we have 16 keys to represent the hexa number 0 to F arranged as shownin fig.

To detect a pressed key

The microcontroller grounds all rows by providing 0 to the P1, and then it reads the columns. If data is 1111, no key has been pressed and the process continues until a key press is detected. If one of the column bits has a zero, this means that a key press has occurred.

To identify exact key pressed

Microcontroller Start with the top row by grounding it, then it reads the columns. Ifthe data read is all 1s, no key in that row is activated and the process is moved to the next row until reach the row that has a pressed key. At this stage, microcontroller identifies the row that has a pressed key and can setup the starting address in the look-up table for that row. The last step is finding the column thathas a pressed key by rotating the column bits by using carry flag and RRC instruction, when 0 bit found in carry, microcontroller pulls the corresponding code from the look-up table.

The following code is for scanning the 4 X 4 keyboard.

```
            MOV P2, #0FFH  -  P2  as  a  i/p
    port BACK:  MOV P1, #00H -  P1  as  a
    o/p portGO:            MOV A, P2
          ANL A, #000D1111B
          CJNE  A,  #00001111B
  GOGO1:     ACALL DELAY
       MOV A, P2
       ANL A, #0000111lB
       CJNE  A,  #00001111B,  PRS
          SJMP GO1
PRS:      ACALL DELAY
       MOV A, P2
       ANL A, #000011l1B
       CJNE  A,  #00001111B,  OVER
          SJMP GO1
  OVER:   MOV          P1,
          #11111110BMOV A,
          P2
       ANL A, #00001111B
       CJNE  A,  #O00O1111B,  ROW0
          MOV P1, #11111101B
                                                  MOV
    A,P2
           ANL A,#000O1111B
           CJNE    A,#D00Oll1lB,ROW_1
              MOV Pl,#11111011B
           MOV A,P2
           ANL A,#00001111B
           CJNE  A,#00001111B,ROW_2
```

```
MOV P1,#11110111B
MOV A,P2
ANL A,#00001111B
CJNE   A,#00001111B,ROW_3
    LJMP K2
MOV         DPTR,#KCODEO
    SJMP FIND
MOV DPTR,#KCODE1SJMP
    FIND
MOV DPTR,#KCODE2SJMP
    FIND
MOV  DPTR,#KCODE3 RRC
    A
     JNC   MATCH INC
         DPTR     SJMP
         FIND CLR A
     MOVC       A,OA+DPTR
         MOV P0/A
     LJMP K1
```

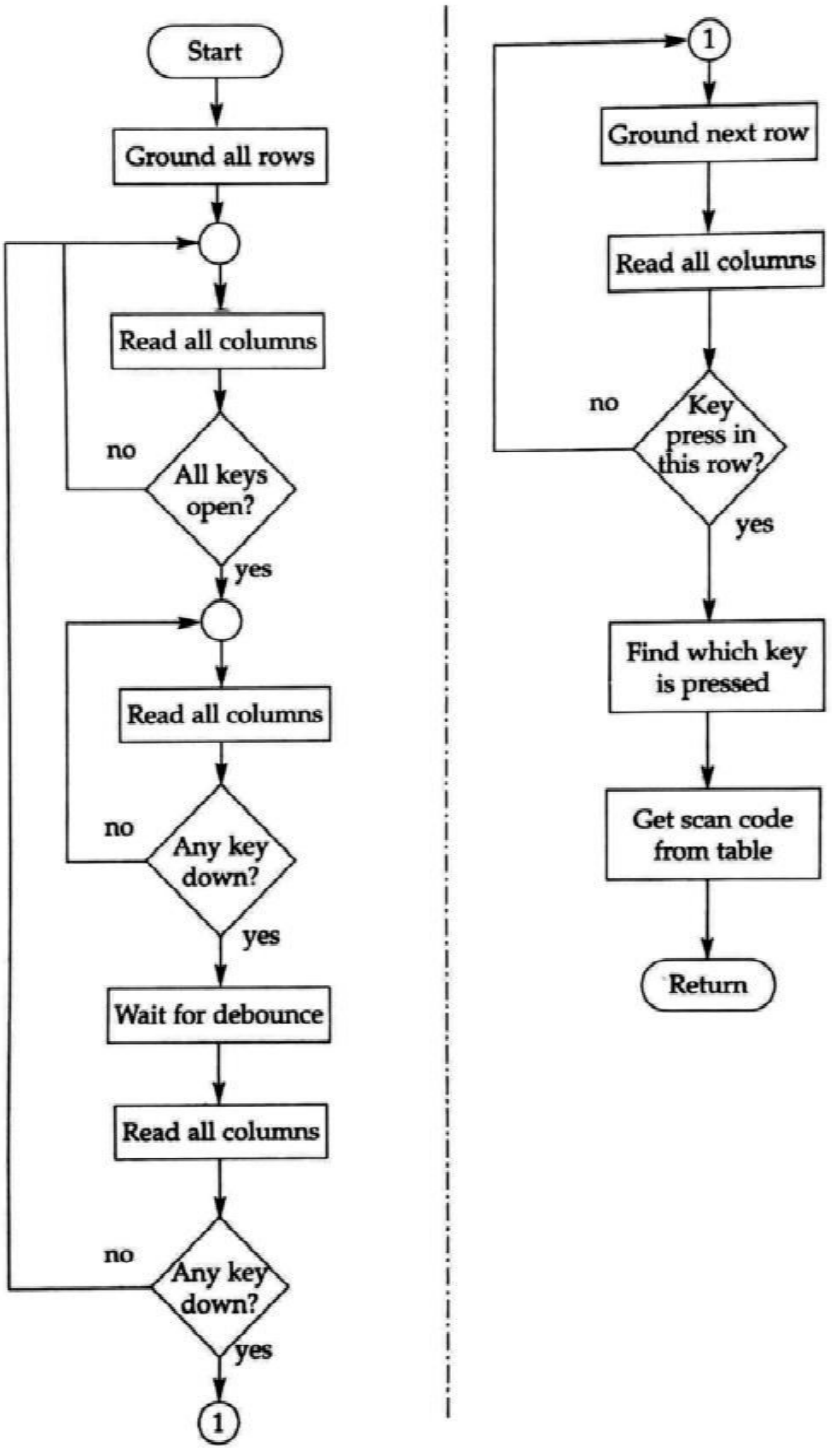
LOOK-UP TABLE POR EACH ROW

ORG 0000H DB ‘0’,’1’,’2’,’3’

DB ‘4’,’5’,’6’,’7’

DB ’8’,’9’,’A’,’B’

DB ‘C’,’D’,’3’,’F’END

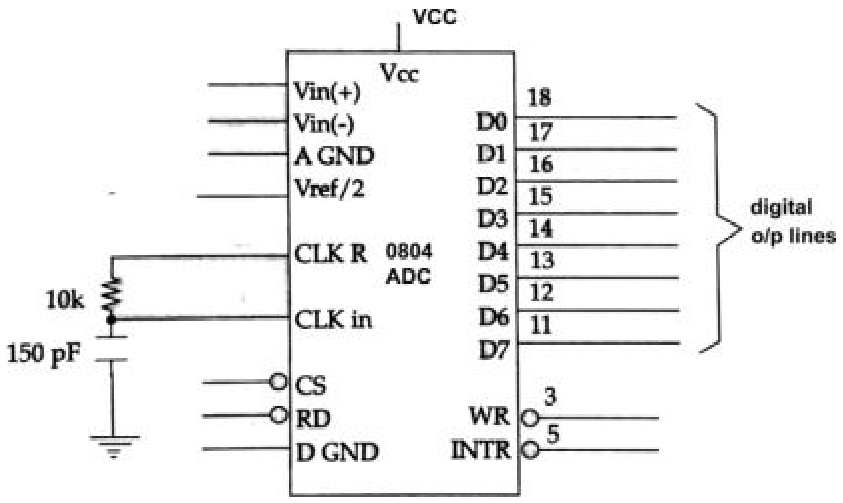

Start
Ground all rows
Read all columns
no
All keys open?
yes
Read all columns
no
Any key down?
yes
Wait for debounce
Read all columns
no
Any key down?
yes
1
1
Ground next row
Read all columns
no
Key press in this row?
yes
Find which key is pressed
Get scan code from table
Return

5.5Analog -to-digital converters (ADC)

Analog-to-digital converters are a commonly used data collecting equipment. Digital computers employ binary (discrete) values, but everything in the actual world is analog (continuous), including temperature, humidity, velocity, and pressure (liquid or wind).

An apparatus that transforms a physical quantity into electrical impulses is called a transducer. Sensors are another term for transducers. We need to use an ADC to convert the analog signals to digital values so that the microcontroller can read and interpret them. There are options for both parallel and serial ADCs.

Parallel ADC 0804



capacitor and a resistor, as shown in Figure. In this case the clock frequency is given by

f □ $\frac{1}{\quad}$

1.1 R C

Ex: Typical values are R = 10 Kohms and C = 150 pF. Substituting in the above equation, we get f = 606 kHz. In this case, the conversion time is 110 us.

INTR (end of conversion)

This is an output pin and is active low. It is a normally high pin and when the conversion is finished it goes low to signal the CPU that the converted data is readyto be picked up.

After INTR goes low, we make CS = 0 and send a high-to-low pulse to the RD pinto get the data out of the ADC0804 chip.

V_{in} (+) and V_{in} (-)

These are the differential analog inputs where V_{in} = V_{in} (+) – V_{in} (-). Often the V_{in} (-) pin is connected to ground and the V_{in} (+) pin is used as the analog input.

VCC

This is the +5 volt power supply. It is also used as a reference voltage when the V_{ref}/2 is open.

V_{ref}/ 2

This pin is used for the reference voltage. If this pin is open, the analog input voltage for the ADC0804 is in the range of 0 to 5V. V_{ref} / 2 is used to implementanalog input voltages other than 0 to 5 V. For example, if the analog input range needs to be 0 to 4 volts, V_{ref}/ 2 is connected to 2V.

D0-D7

D0 - D7 are the digital data output pins of a parallel ADC0804. The converted datais accessed only when CS = O and RD is forced low.

Digital and analog grounds

These are the input pins that provide ground to both analog and digital signals.

The following steps must be taken in order to convert data using the ADC0804.

- Set CS = 0 and pulse pin WR from low to high to start the conversion.
- Keep watching the INTR pin.
- If the INTR is low, the conversion procedure is finished.
- After the INTR decreases, we transmit high-to-low data to RD and set CS = 0 to retrieve the data from ADC0804.

Figure illustrates the process's time.

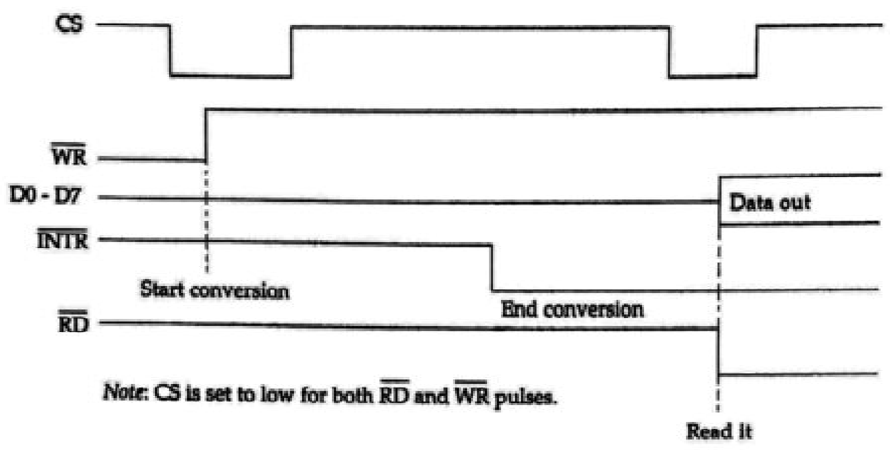
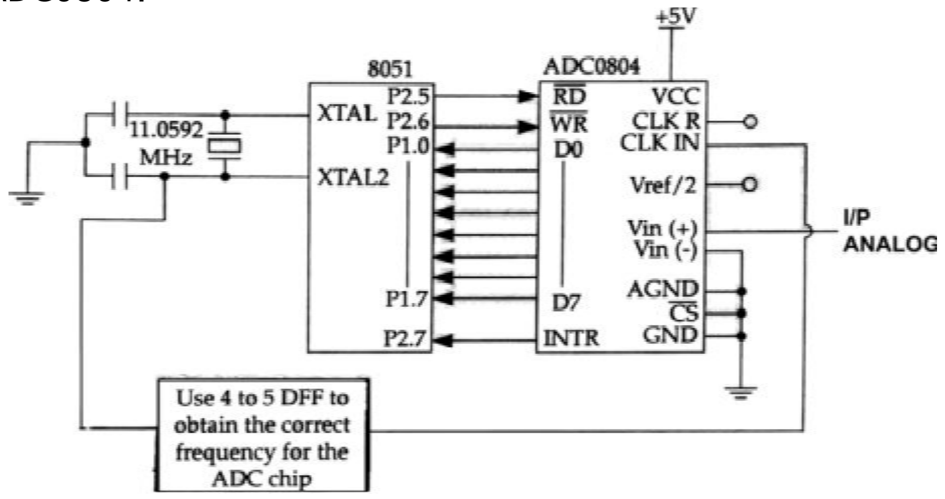


Fig shows interfacing connections between ADC0804 and 8051. Here CLKIN connected to the external CLK at 8051's crystal oscillator.

The program that follows enters an analog input into register A while keeping an eye on the INTR pin. Subroutines for data presentation and hex-to-ASCII conversion are then called. The pins D0-D7, which are the out data pins of the ADC0804, are linked to port 1 of the 8051. The pins P2.5, P2.6, and P2.7 of the 8051 are connected to the RD, WR, and INTR of the ADC0804.

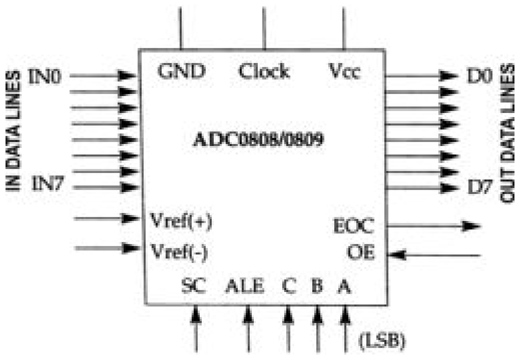


Program:

```
          RD BIT P2.5
          WR BIT P2.6
          INTR BIT P2.7
          MYDATA EQU P1
          MOV P1, #0FFH
BACK:     SETB INTR
          CLR WR
          SETB WR
HERE:     JB INTR, HERE
          CLR RD
          MOV A, MYDATA
          ACALL  CONVERSION
          ACALL  DISPLAY
          SETB RD
          SJMP BACK
```

Another useful chip is the ADC0808/0809 has 8 analog inputs. This chip allows 8 different analog inputs by using selection ckt and has 8 output data lines just like ADC0804.

Fig. shows pin configuration of ADC 0808/09 and table shows selection of analog input channel.



Selected i/p channel	C	B	A
IN0	0	0	0
IN1	0	0	1
IN2	0	1	0
IN3	0	1	1
IN4	1	0	0
IN5	1	0	1
IN6	1	1	0
IN7	1	1	1

In many applications, space is critical issue, using large no. of pins for data is not feasible. For this reason serial ADCs are becoming widely used than parallel ADC. MAX 1112 is a serial ADC chip from MAXIM Corporation.

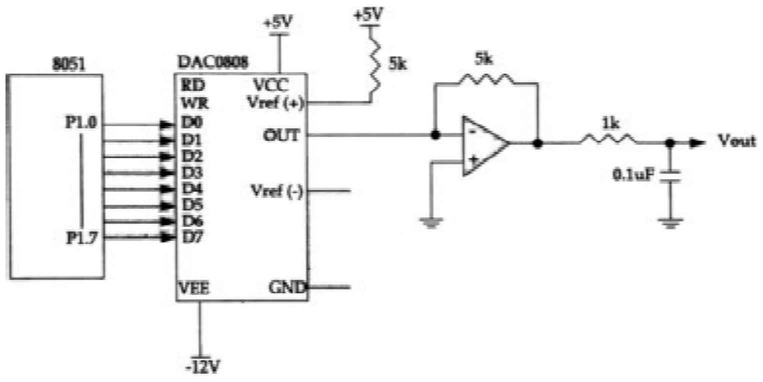
5.6 DAC interfacing

Digital-to-analog (DAC) converter

One common tool for converting digital pulses to analog signals is the digital-to-analog converter (DAC). We basically have two approaches for creating a DAC from digital electronics.
1. Weighted binary
2.Lattice R/2R network

The R/2R approach is used by the MC1408 (DAC0808). An 8bit DAC is used to transform digital data (of 8 bits) into analog signals. The number of data inputs determines the DAC's resolution. Given n, the number of data bit inputs, the number of analog output levels is therefore equal to 2n. Consequently, the output of an 8-input DAC (0808) is 256 discrete voltage or current values. The DAC0808 converts digital inputs to current (Iout). The output is then converted to voltage by attaching a resistor to the Iout pin.
The reference current (Iref) and the binary numbers D0 through D7 inputs of the DAC0808 determine the total current supplied by the Iout pin, which is determined by:

$I = I$



SJMP BACK

Where D0 is the LSB, D7 is the MSB for the inputs, and I_{ref} is the input current that must be applied. If I_{ref} = 2 mA, that all inputs to the DAC are high. The maximum current is 1.99 mA, which is as follows.

Digital i/p = FFh = 11111111b

$$I_{out} = 2\left(\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\frac{1}{16}+\frac{1}{32}+\frac{1}{64}+\frac{1}{128}+\frac{1}{256}\right) = 2\left(\frac{255}{256}\right) = 2(0.996) = 1.99\,mA$$

Ex: Program for generation of saw tooth wave form by using above interfacing ckt:

```
         MOV A, #00H
HERE:    MOVE P1, A
         INC A
         SJMP HERE
```

Ex: The following program is to set maximum required peak value in saw tooth wave.

```
         MOV R0, #7FH
BACK:    MOV A, #00H
HERE:    MOVE P1, A
         INC A
         CJNE A, R0 HERE
         SJMP BACK
```

5.7 Temperature sensor interfacing

Transducers convert physical data such as temperature, light intensity, flow, and speed to electrical signals. Depending on the transducer, the output producedis in the form of voltage, current, resistance, or capacitance. Temperature is converted to electrical signals using a transducer called a thermistor, it responds to temperature change by changing resistance, but its response is not linear.

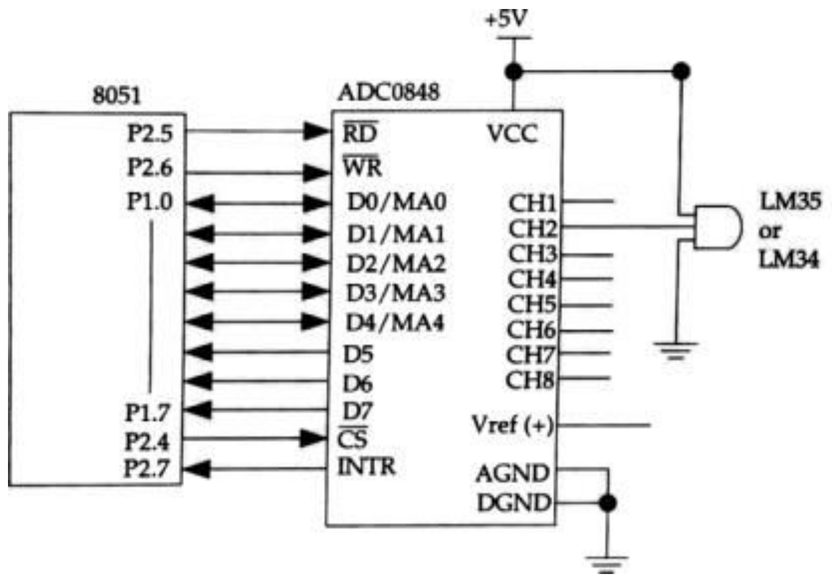
The complexity associated with writing software for such nonlinear devices. That's why manufacturers introduced linear temperature sensor. Simple and widely used linear temperature sensors are the LM34 and LM35.

LM34 and LM35 temperature sensors

The sensor LM34 is a temperature sensor whose output voltage is linearly proportional to the Fahrenheit temperature. The LM34 requires no external calibration since it is internally calibrated. It outputs 10 mV for each degree of Fahrenheit temperature.

The LM35 sensor is a temperature sensor whose output voltage is linearly proportional to the Celsius (centigrade) temperature. The LM35 requires no external calibration since it is internally calibrated. It outputs 10 mV for each degree of centigrade temperature.

Interfacing LM 35 to 8051



Signal conditioning

The most common transducers produce an output in the form of voltage, current, charge, capacitance, and resistance. However, we need to convert these signals to voltage in order to send input to an A-to-D converter. This conversion (modification) is called signal conditioning.

The thermistor changes resistance with temperature. The change of resistance must be translated into voltages by using signal conditioning process to give ADC.

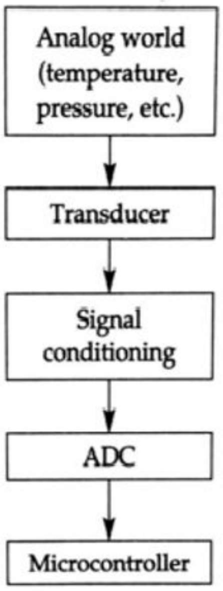
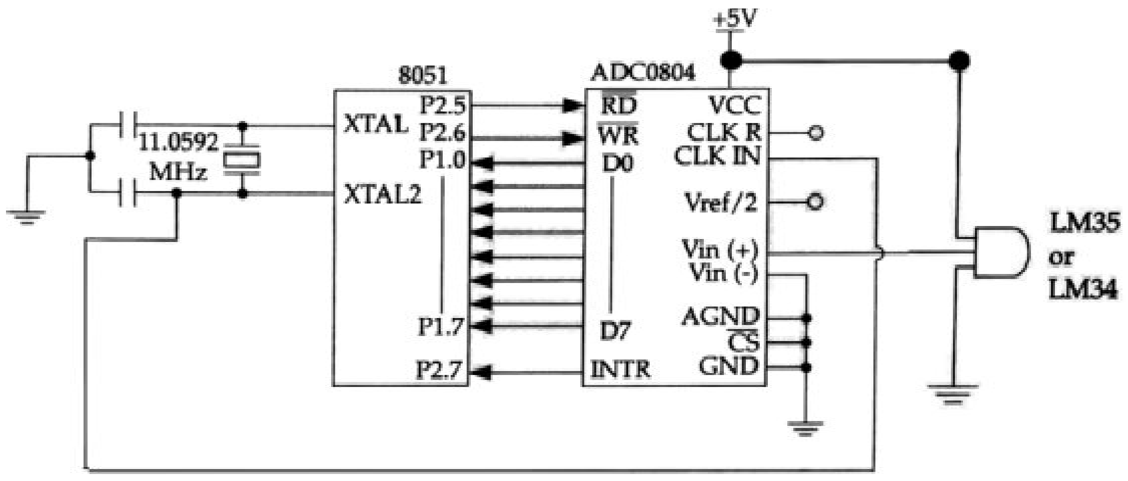


Figure shows connections of an LM35 to an ADCO848 which is interfaced to 8051. The ADC0848 has 8-bit resolution with a maximum of 256 steps. The LM35 (or LM34) produces 10 mV for every degree of temperature change, that means it produces maximum of 2560 mV (2.56 V) for full-scale output and is applied to ADC0848. Therefore, in order to produce the full-scale V_{out} of 2.56 V for the ADC0848, we need to set V_{ref} = 2.56V. in this case ADC0848 outputs directly to the temperature as monitored by the LM35.

The following program is for displaying temperature.

```
                    RD PIN P2.6 WR
                       PIN      P2.7
                       INTR      PIN
     P2.5CS PIN P2.4
MYDATA EQU P1CLR CS
SETB INTRBACK: CLR
     WR
SETB WR HERE:  JB    INTR
     HERE
CLR RD
MOV A, MYDATAACALL DIPLAY
     SJMP BACK
```

The interfacing ckt for temperature sensor to the microcontroller via ADC0804 asshown in fig.



+5V
8051
ADC0804
11.0592
MHz
XTAL
XTAL2
P2.5
P2.6
P1.0
P1.7
P2.7
RD
WR
D0
D7
INTR
VCC
CLK R
CLK IN
Vref/2
Vin (+)
Vin (-)
AGND
CS
GND
LM35
or
LM34