Interfacing Types

There are two types of interfacing in context of the 8085 processor.

1. Memory Interfacing.   2. I/O Interfacing.

Memory Interfacing:

While executing an instruction, there is a necessity for the microprocessor to access memory frequently for reading various instruction codes and data stored in the memory. The interfacing circuit aids in accessing the memory. Memory requires some signals to read from and write to registers. Similarly the microprocessor transmits some signals for reading or writing a data.

But what is the purpose of interfacing circuit here?

The interfacing process involves matching the memory requirements with the microprocessor signals. The interfacing circuit therefore should be designed in such a way that it matches the memory signal requirements with the signals of the microprocessor. For example for carrying out a READ process, the microprocessor should initiate a read signal which the memory requires to read a data. In simple words, the primary function of a memory interfacing circuit is to aid the microprocessor in reading and writing a data to the given register of a memory chip.

I/O Interfacing:

The keyboard and Displays are used as communication channel with outside world. So it is necessary that we interface keyboard and displays with the microprocessor. This is called I/O interfacing. In this type of interfacing we use latches and buffers for interfacing the keyboards and displays with the microprocessor. But the main disadvantage with this interfacing is that the microprocessor can perform only one function. It functions as an input device if it is connected to buffer and as an output device if it is connected to latch. Thus the capability is very limited in this type of interfacing.

Programmable peripheral devices were introduced by Intel to increase the overall performance of the system. These devices along with I/O functions, they perform various other functions such as time delays, counters and interrupt handling. These devices are nothing but a combination of many devices on a single chip. A programmable device can be set up to perform specific function by writing a code in the internal register. As this code controls the function of the device it's called **control word** and internal register in which it is stored is called **Control Register**.

I/O INTERFACING: The two methods of data transfer are: ·

Serial I/O : One bit is transferred using one data line ·

Parallel I/O : Data can enter (or exit) in groups of eight bits using the entire data bus.

Thus the I/O devices **(Keyboards and displays**) can be interfaced using two methods namely: ·

Peripheral-mapped I/O: The device is identified with an 8-bit address and enabled by I/O – related control signals. ·

Memory-mapped I/O: The device is identified with a 16-bit address and enabled by memory related control signals.

Concepts for peripheral mapped I/O

a) When an I/O instruction is executed, 8085 places the device address (Port number) on the de multiplexed low – order and high – order address bus

. b) The address is decoded to generate the pulse corresponding to the device

. c) The device address pulse is AND ed with the appropriate control signals like IO/M, RD and WR to assert the I/O device

. d) A latch is used for an output port and a tri-state buffer is used for an input port..

e) The address bus can be decoded by using either the absolute or the linear select decoding reduces the component cost but the I/O device ends up with multiple addresses.

Memory Mapped I/O : Here, the input and output devices are assigned and identified by 16-bit addresses.

To transfer data between 8085 and I/C devices, memory – related instructions like LDA, STA etc are used.

The control signals MEMR and MEMW should be connected to I/O devices.

STA 8000H; Address 2050H; stores the contents of A to 8000.

2050 32;

2051 00;

2052 80; Thus 8085 requires 4 Machine cycles to execute STA;

Instruction fetch and decode in M1;

read 2051 and 2052 in M2 and M3;

In M4 8085 places the entire address (8000H) on the address lines, the contents of the accumulator on data bus and generates MEMW Characteristics

Characteristics	Memory-Mapped I/O	Peripheral I/O
1. Device address	16 Bit	8 Bit
2. Control Signals	RD / WR (MEMR / MEMW)	IO/M , RD / WR (IOR, IOW)
3. Instructions	Memory related instructions LDA, STA, MOV, ADD, SUB	IN, OUT
4. Data transfer	Between any register and I/O	Between I/O and Accumulator
5.Maximum number of I/O	64K memory shared between I/O and system memory.	256 Input; 256 Output
6. Execution	13 T States (LDA, STA) 7 T States (MOV)	10 T States
7. Hardware	More hardware required to decode 16 bit address	Less hardware to decode 8 bit address
8. Other features	Arithmetic or Logical operations can be performed	Not Possible.

1. I/O Mapped I/O

In I/O mapped I/O, the 8085 uses $IO/\overline{M}$ signal to distinguish between I/O read/write and memory read/write operations. The 8085 has separate instructions IN and OUT for I/O data transfer. When 8085 executes IN or OUT instruction, it places device address on the demultiplexed low order address bus as well as the high order address bus, *i.e.,* the higher order address bus duplicates the contents of demultiplexed low-order address bus, when 8085 executes IN or OUT instruction.

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0
								Device address = 60_H							

IN and OUT Instructions

IN : It is used to read 8-bit data from I/O device into the accumulator. This two byte instruction has, the first byte as an opcode and the second byte specifies the device address or a port number.

OUT : It is used to send 8 bit data from accumulator to the output device. This two byte instruction has the first byte as an opcode and the second byte specifies the device address or a port number.

5.1 INTERFACING OF INPUT AND OUTPUT DEVICES

The transfer of data between keyboard and microprocessor, and microprocessor and display device is called input/output data transfer or I/O data transfer. The transfer of data between microprocessor and input device is done with the help of input port. The data transfer between microprocessor and output device is done with the help of output port.

Input Port

It is used to read data from the input device such as keyboard. The simplest form of input port is a buffer. The input device is connected to the microprocessor through buffer as shown in Fig. 5.1. The buffer is a tristate buffer and its output is available only when enable signal is active. When microprocessor wants to read data from i/p device, the control signals from the microprocessor activates the buffer by asserting enable input of the buffer.

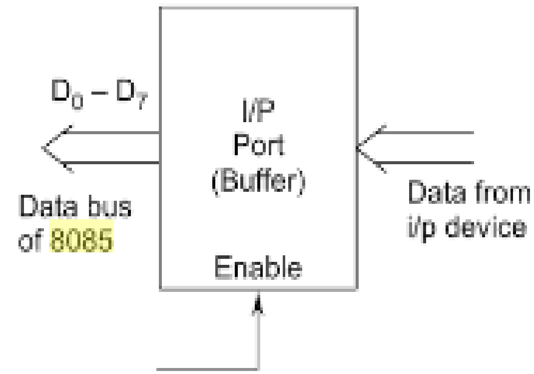


Fig. 5.1

Output Port

It is used to send data to the output device such as display from the microprocessor. The simplest form of output port is a latch. The output device is connected to the microprocessor through latch as shown in Fig. 5.2. When microprocessor wants to send data to the output device, it puts the data on the data bus and activates the clock signal of the latch, latching the data from the data bus at the o/p of the latch. It is then available at the output of latch for the o/p device.

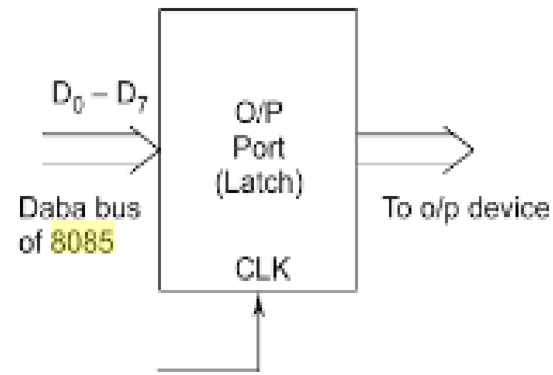


Fig. 5.2

I/O Device Selection

8085 gives 8 bit I/O address in I/O mapped I/O. This means it can select one of the 256 I/O ports. To select an appropriate I/O device, it has to do following:

1. Decode the address to generate unique signal corresponding to the device address on the bus.
2. When device address signal and control signal $(\overline{IOR} \text{ or } \overline{IOW})$ both are low, generate device select signal.
3. Use device select signal to activate the I/O port.

To generate device select signal (Y_0) low,

$$A_2\,A_1\,A_0 = 000_H$$

$$A_3\,A_4\,A_5\,A_6 = 0000_H \text{ (makes } \bar{G}_1 \,\&\, \bar{G}_2 \text{ low)}$$

$$A_7 = 1_H \text{ (makes G high).}$$

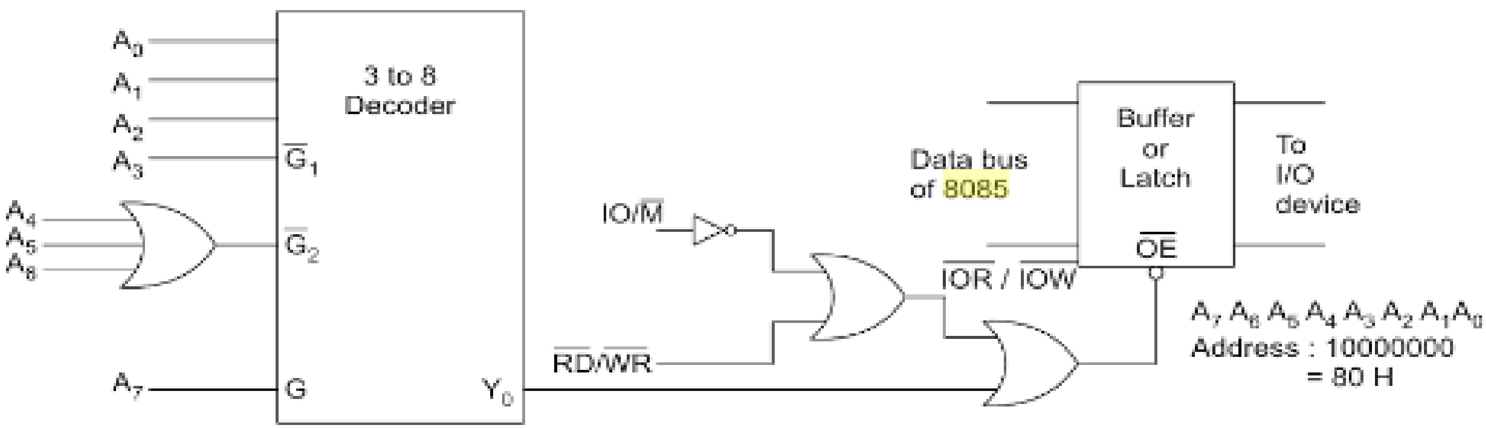


Fig. 5.3 Decoding circuit for I/O

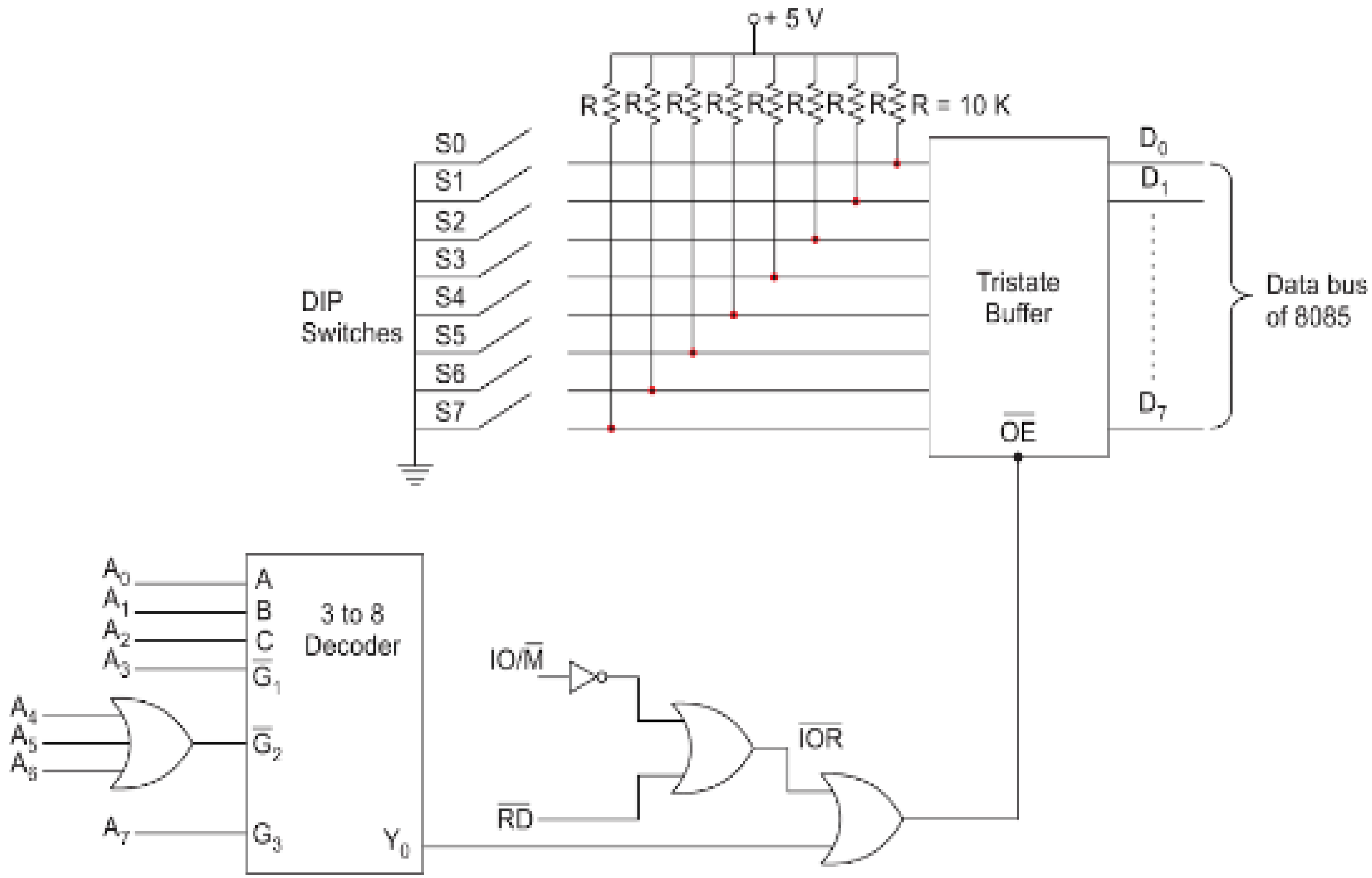
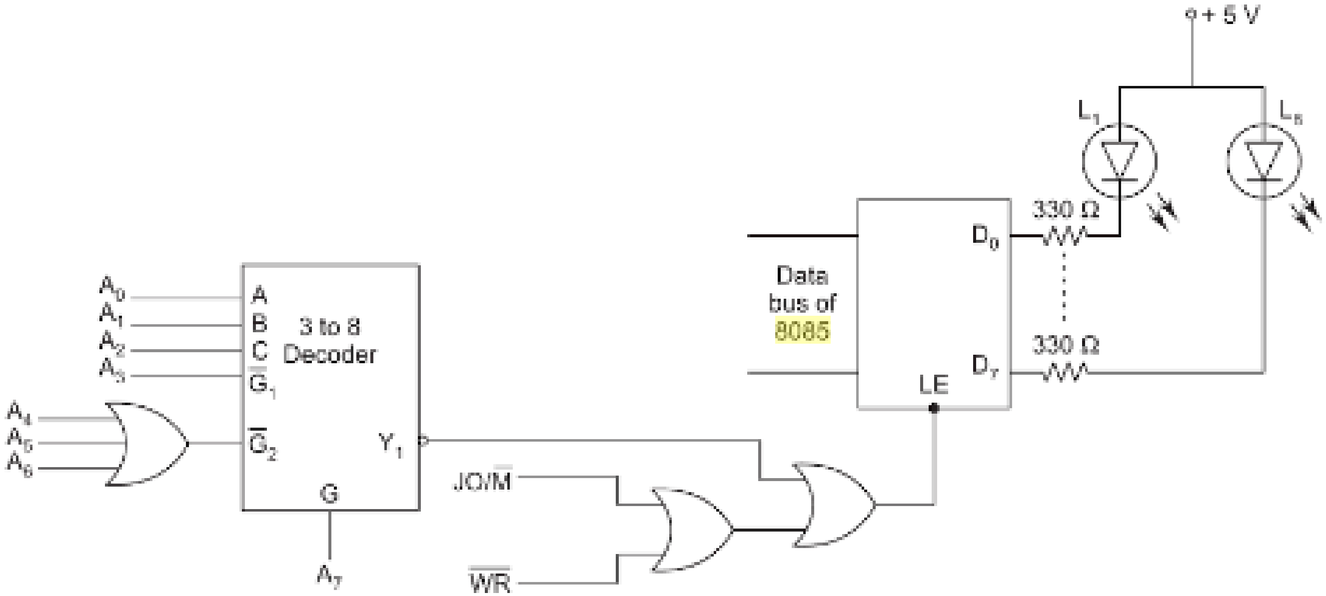


Fig. 5.4 Circuit diagram to interface input port

When, the switch is in released position, the status of line is high otherwise status is low. With this information microprocessor can check a particular key is pressed or not.

Interfacing Output Device

The microprocessor 8085 sends 8 bit data to the output device such as 7 segment displays, LEDs, printer etc. Figure 5.5 shows the circuit diagram to interface output port (latch) which is used to send the signal for glowing the LEDs. LED will glow when output pin status is low.



Address of O/P Device : 81 H

Fig. 5.5 Circuit diagram to interface output port

I/O Interfacing Using I/O Mapped I/O

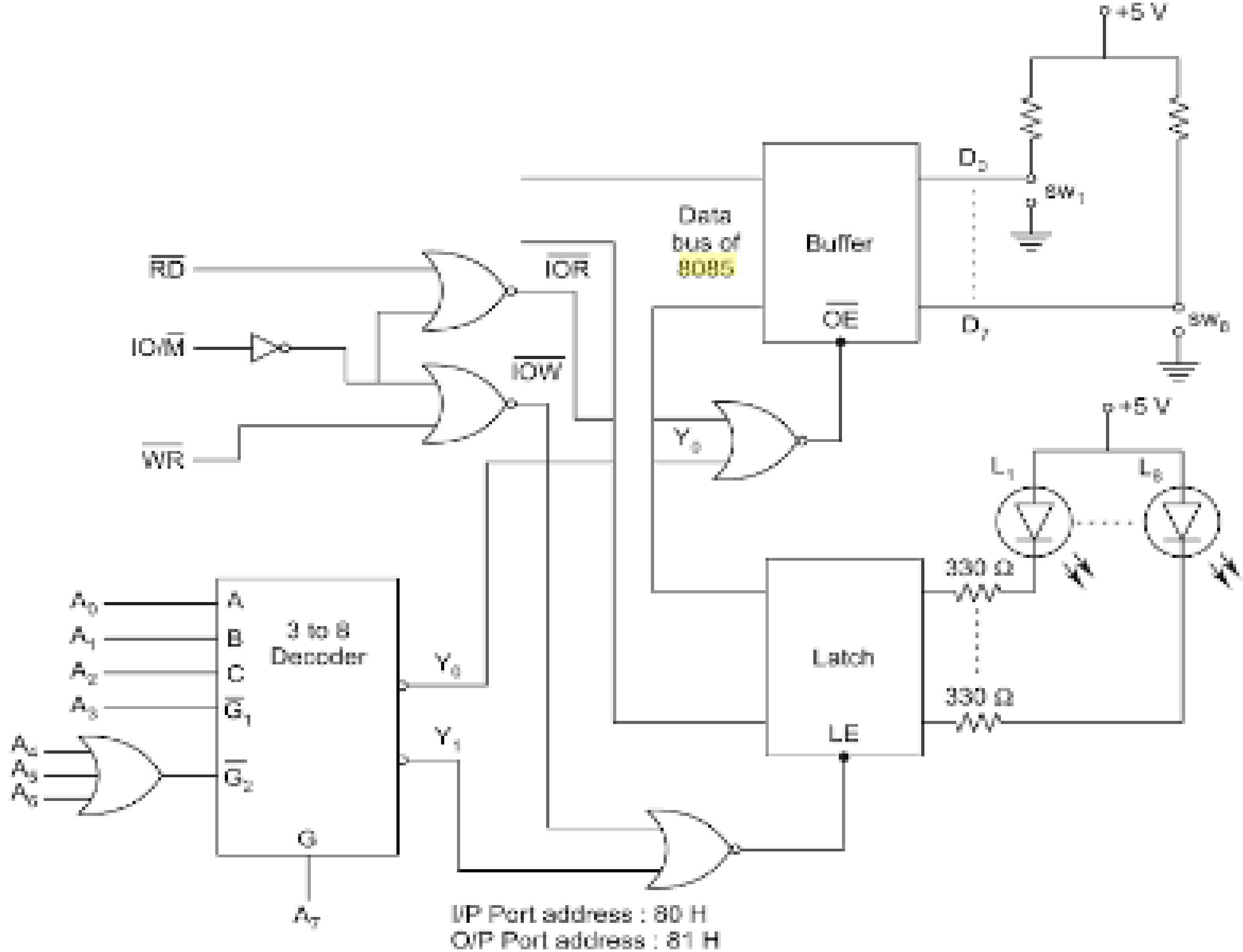
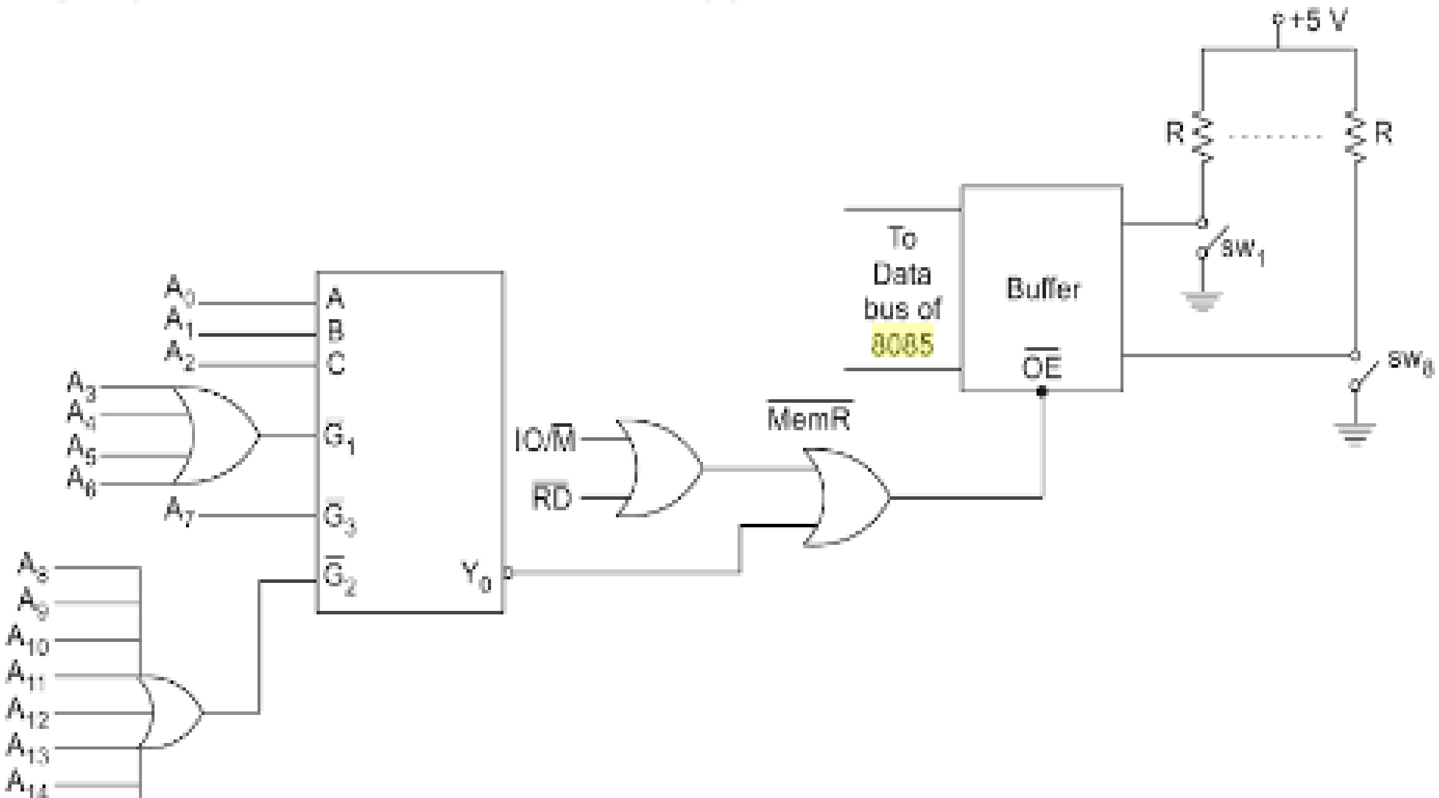


Fig. 5.6 I/O interfacing

2. Memory Mapped I/O

In memory mapped I/O, the I/O devices are assigned and identified by 16 bit addresses. The memory related instruction transfer the data between an I/O device and the microprocessor, as long as I/O port is assigned to the memory address space rather than to the I/O address space. Thus I/O device becomes a part of the system's memory map and hence its name. In memory mapped I/O every instruction that refers to memory location can control I/O.

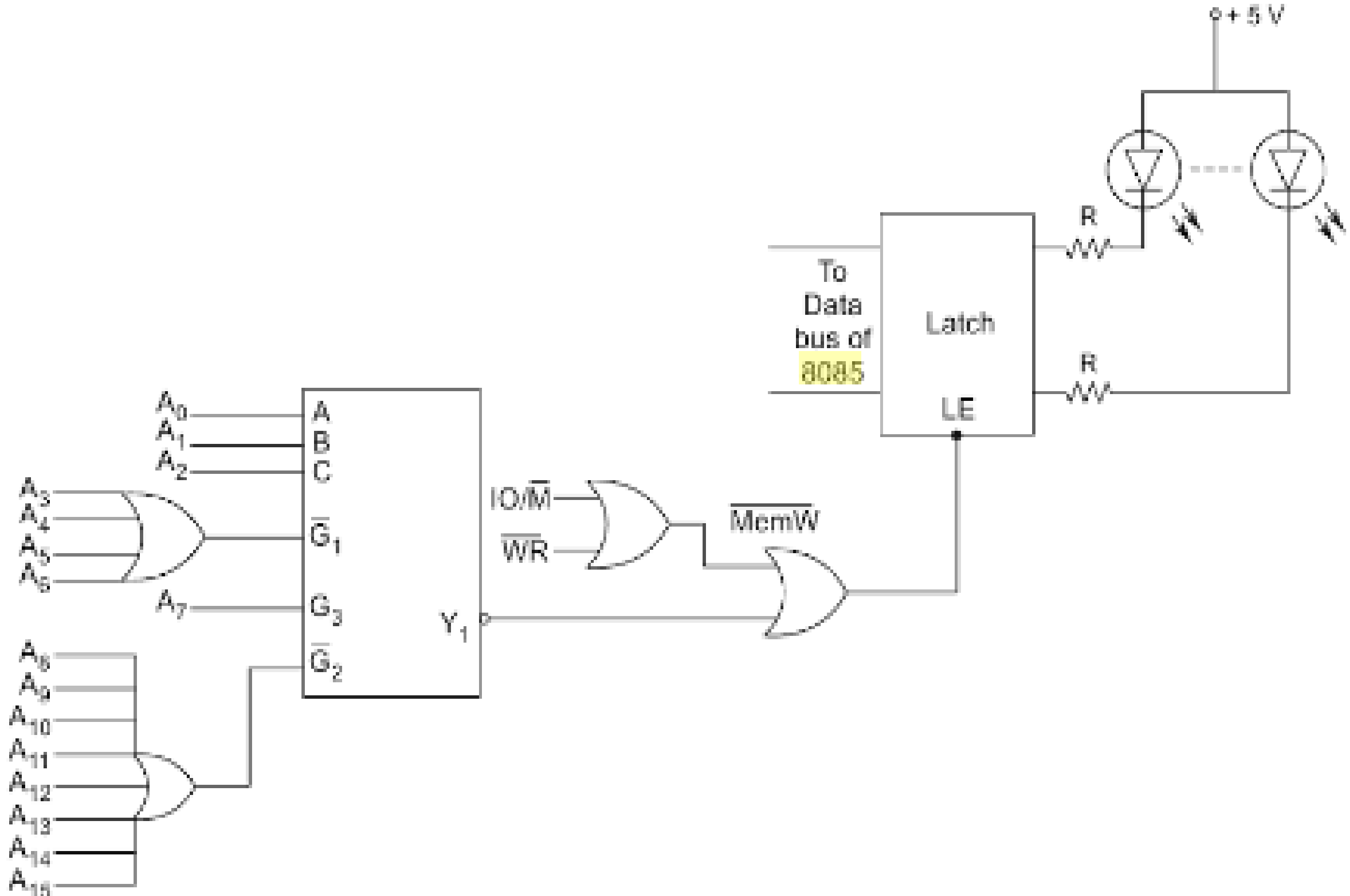
Interfacing Input Device with Memory Mapped I/O



Address of I/P Device : 0080H

Fig. 5.7 Circuit diagram for I/P interfacing with memory mapped I/O

Interfacing Output Device using Memory Mapped I/O



Address of O/P device : 0081H

Fig. 5.8 Circuit diagram for O/P device interfacing using memory mapped I/O

Digital To Analog Convertor (DAC)

The converter which converters the digital form of data in to analog form is called digital to analog converter. In this the digital data in the form of 1's and 0's are used to control the switches which are placed in a analog circuit with reference voltage, based on this switch condition (ON/OFF) and position (MSB or LSB) the output analog amplitude is calculated.

Types of Digital to Analog Converters

Binary Weighted Resistor DAC or R-2^nR DAC

R-2R Ladder DAC

Specifications of a Digital to Analog Converter (DAC)

Resolution: The resolution of a DAC is the smallest change in the output of the DAC for any change in digital input .i.e. if a input to DAC changes one bit, how much analog output has changed in full scale deflection.

% resolution = [Step size / Full scale output (FSO)] * 100

In other way the resolution is the number of states into which the full scale output is divided. i.e if a 8 bit DAC can resolve the FSO up to 255 levels. Each level of output is called step size and for higher number of bits the resolution will be better.

% resolution = [1/(2^N-1) *100 Normally the resolution will be in Milli volts.

Accuracy The Accuracy of a DAC is the difference between output practical analog output to the ideal expected output for a given digital input. The DAC is contains electronic components where the gain plays a major role which can introduce gain error in the output. Due to the the full scale output may differ compared to ideal one. For an example if a DAC of 10 V is said to have an accuracy of 0.01% there will be 10mv output deviation. The another factor which implicates the accuracy is the zero offset error i.e for a zero input the output of DAC reflects some offset value.

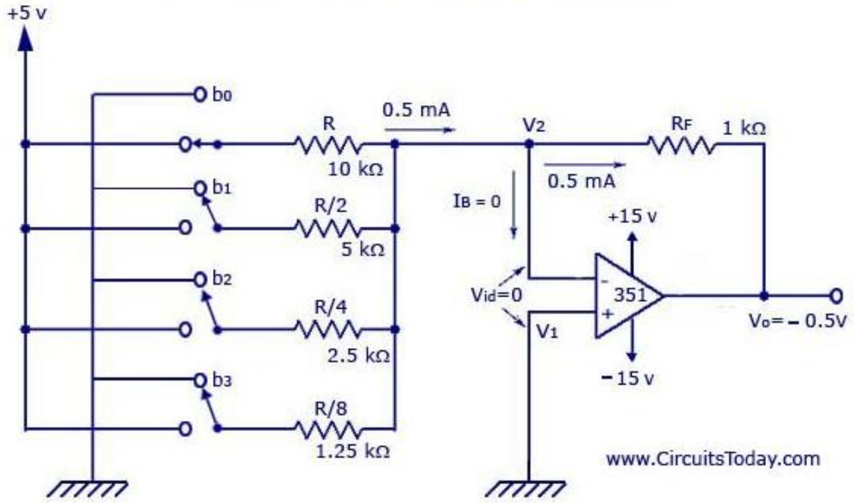
Conversion Speed The conversion speed of the DAC is output analog value settling time period for a change in the digital input. This is also called settling time period of DAC. Normally it will be micro seconds and in some advanced micro controller DAC it may be nano second

MonotonicityThe Digital to Analog Converter is said to be monotonic if its analog value is either increasing or equal to previous value for an LSB change in input digital signal.

Digital to Analog Converter using Binary-Weighted Resistors

A D/A converter using binary-weighted resistors is shown in the figure below. In the circuit, the op-amp is connected in the inverting mode. The op-amp can also be connected in the non-inverting mode. The circuit diagram represents a 4-digit converter. Thus, the number of binary inputs is four. We know that, a 4-bit converter will have 2^4 = 16 combinations of output. Thus, a corresponding 16 outputs of analog will also be present for the binary inputs. Four switches from b0 to b3 are available to simulate the binary inputs:

D/A Converter With Binary Weighted Resistors



Working

The circuit is basically working as a current to voltage converter.

b0 is closed It will be connected directly to the +5V.

Thus, voltage across R = 5V

Current through R = 5V/10kohm = 0.5mA

Current through feedback resistor, Rf = 0.5mA

output voltage = -(1kohm)*(0.5mA) = -0.5V

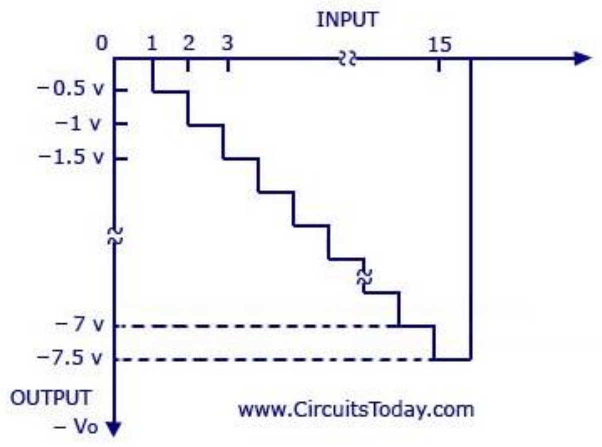
b1 is closed, b0 is open R/2 will be connected to the positive supply of the +5V. Current through R will become twice the value of current (1mA) to flow through Rf. Thus, output voltage also doubles.

b0 and b1 are closed Current through Rf = 1.5mA Output voltage = -(1kohm)*(1.5mA) = -1.5V

Thus, according to the position (ON/OFF) of the switches (bo-b3), the corresponding "binary-weighted" currents will be obtained in the input resistor. The current through Rf will be the sum of these currents. This overall current is then converted to its proportional output voltage. Naturally, the output will be maximum if the switches (b0-b3) are closed

V0 = -Rf *([b0/R][b1/(R/2)][b2/(R/4)][b3/(R/8)]) – where each of the inputs b3, b2, b1, and b0 may either be HIGH (+5V) or LOW (0V).

The output is a negative going staircase waveform with 15 steps of -0.5V each. In practice, due to the variations in the logic HIGH voltage levels, all the steps will not have the same size. The value of the feedback resistor Rf changes the size of the steps. Thus, a desired size for a step can be obtained by connecting the appropriate feedback resistor. The only condition to look out for is that the maximum output voltage should not exceed the saturation levels of the op-amp . Metal-film resistors are more preferred for obtaining accurate outputs The graph with the analog outputs versus possible combinations of inputs is shown\



Digital-to-Analog Converter Circuit - Binary-Weighted Resistors Method Graph

Disadvantages

If the number of inputs (>4) or combinations (>16) is more, the binary-weighted resistors may not be readily available. This is why; R and 2R method is more preferred as it requires only two sets of precision resistance values

Digital to Analog Converter with R and 2R Resistors

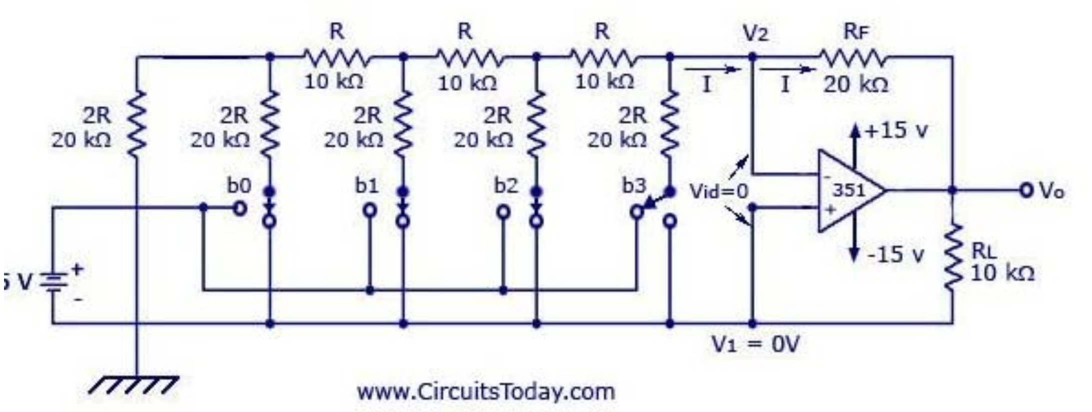
A D/A converter with R and 2R resistors is shown in the figure below. As in the binary-weighted resistors method, the binary inputs are simulated by the switches (b0-b3), and the output is proportional to the binary inputs. Binary inputs can be either in the HIGH (+5V) or LOW (0V) state. Let b3 be the most significant bit and thus is connected to the +5V and all the other switches are connected to the ground.

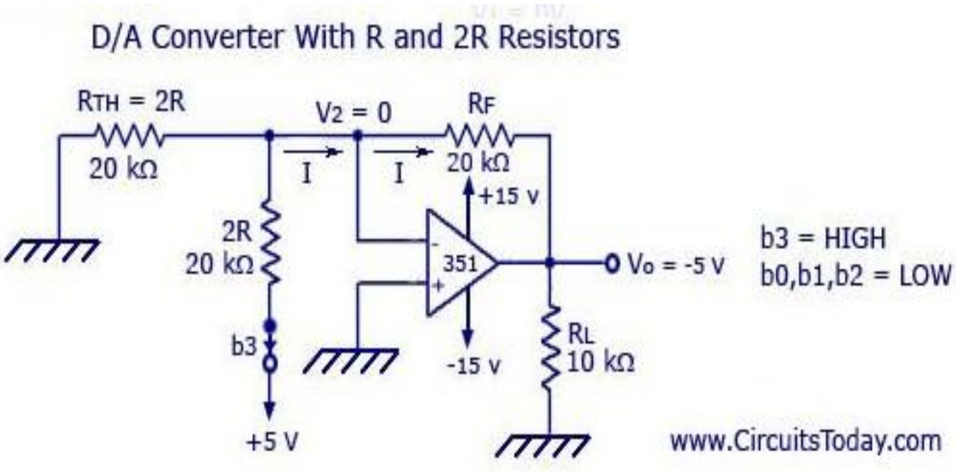
Thus, according to Thevenin's equivalent resistance , R_{TH}, = [{[(2RII2R + R)} II2R] + R}II2R] + R

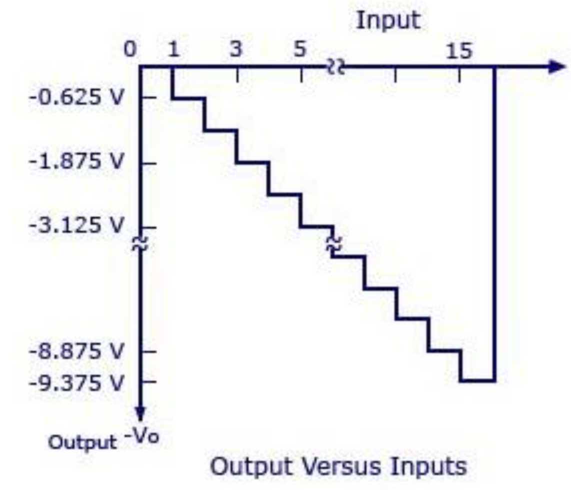
= 2R = 20kOhms.

The resultant circuit is shown in next slide.

D/A Converter With R and 2R Resistors







In the figure shown above, the negative input is at virtual ground, therefore the current through R_{TH}=0.

Current through 2R connected to +5V = 5V/20kohm = 0.25 mA

The current will be the same as that in Rf.

V_o = -(20kohm)*(0.25mA) = -5V

Output voltage equation is given below. $V_0 = -R_f \left(b3/2R + b2/4R + b1/8R + b0/16R \right)$

Analog to Digital Converter

The data converter which converts the data from analog values to digital values is called Analog to Digital converter in shot it is called ADC. In ADC the input signal value is sampled at a particular time interval and compared with the analog value produced by the combination of counter and DAC. If the output of the comparator is zero then the value of the counter will be the output digital value. These are very important converters as the environmental analog signals has too be converted in to digital for processing with the digital computer.

Types of Analog to Digital Converters:

1. Simultaneous or Flash A/D Converters
2. Counter-Type A/D Converter
3. Tracking-Type A/D Converter
4. Successive Approximation Type A/D Converter
5. Single-, Dual- and Multislope A/D Converters
6. Sigma-Delta A/D Converter

Specifications of ADC

Resolution of ADC: The resolution of an A/D converter is the quantum of the input analogue voltage change required to increment its digital output from one value to the next higher code value. i.e. if an n bit ADC then it needs $1/ 2^n-1$ of full scaled output to reflect at the output of ADC. For example The resolution of an eight-bit A/D converter can be expressed as one part in 255 or as 0.4% of full scale or simply as eight-bit resolution. If such a converter has a full-scale analogue input range of 10 V, it can resolve a 40 mV change in input.

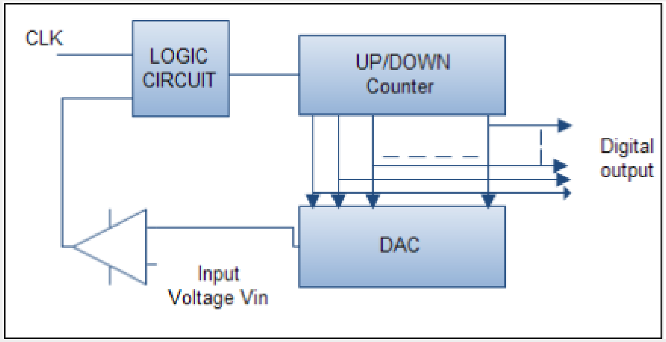
Aliasing or sampling frequency It is the rate at which the rate of the analog value is sampled for digitizing. If the sampling frequency is less than the signal frequency then due to aliasing the ADC output will be distorted and cannot reproduce the exact digital value. So to extract the nice digital output the ADC nyquist frequency should be at least 5 or 10 times greater than the signal frequency.

Step recovery of ADC: It shows that how quickly an ADC changes its output to match a large, sudden change in the analog input.

Tracking type ADC (Analog to Digital Converter)

In counter type ADC, the counter has to come to zero for every conversion which is a disadvantageous with respect to high conversion time. However the input analog signal is continuous in signal amplitude with respect to time. I.e. the difference between the two sampled analog values will be less, So for the first analog sampled value if the counter has reached to its equal value the counter has to be stopped there instead of coming to zero. The second analog value will be slightly higher or lower than the previous sampled value so the counter can increase or decrease the count for producing comparable analog output from DAC.

To perform this operation instead of normal counter Up/Down counter is used in tracking type ADC. After the first sampled value the up/down counter will tracks the input analog value so that it is called tracking type of ADC. The detailed block diagram of Tracking type ADC is shown in below figure.



Operation of tracking type ADC

The N bit up/down counter starts counting according to the clock pulse and provides a digital input to the DAC. The DAC converts the digital input into corresponding analog output which is applied to opamp. The opamp compares the DAC analog output with the input analog sampled value, if the input value greater than the DAC output it provides a clock pulse to increment the counter or otherwise if it is lesser than the DAC output it provides a clock pulse to decrement the counter. But normally for first sampled value the counter will incremented to match to the analog value and for subsequent samples only the counter may get decrement pulse.

When the DAC output is equal to the sampled value, the digital output will be taken from the up/down counter directly. Here shift register is not needed to capture the digital date because the counter will not go to reset state. Then onwards for the next sample value, the counter will increment or decrement according to the difference output of opamp.

This is also called as Derivative counter type ADC because the counter output depends difference to the previous and next sampled value like as differentiator

Advantages of Tracking type ADC

Shift register is not needed so the cost is less. Speed is high compared to digital ramp type as the counter will not reset

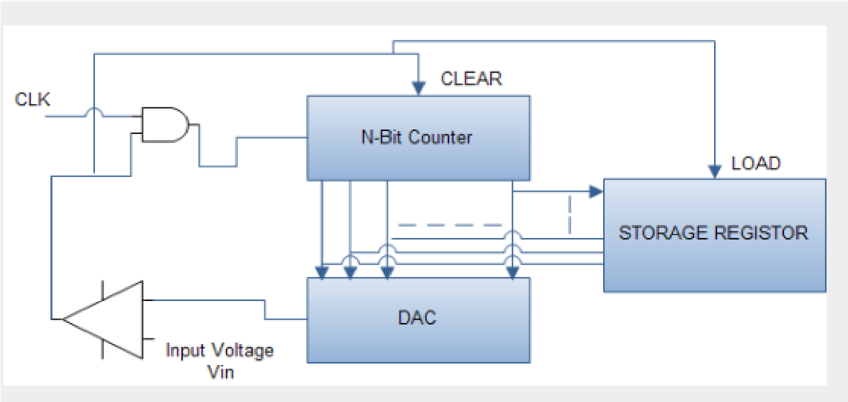
Disadvantages of Tracking type ADC

Up/Down counter leads to complexity of the circuit.

The digital output will never be constant because of differentiator effect.

I.e. for a constant analog value also the output will oscillating this is known as bit bobble.

Counter Type ADC (Analog to Digital Converter)

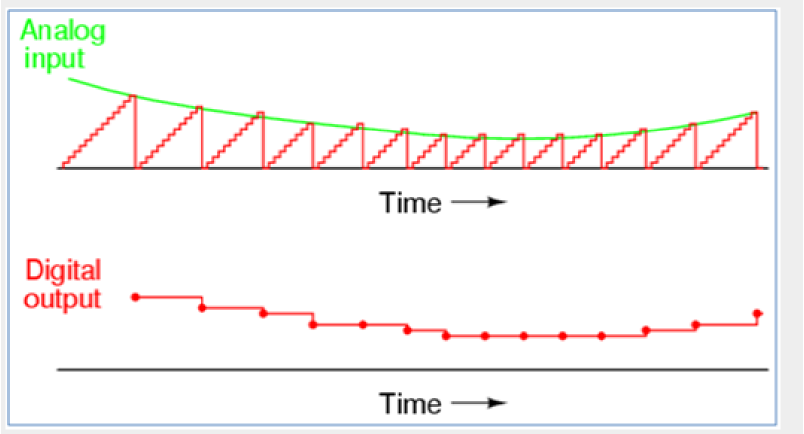


The Counter type ADC is the basic type of ADC which is also called as digital ramp type ADC or stair case approximation ADC. This circuit consists of N bit counter, DAC and Op-amp comparator as shown in figure.

Operation of counter type ADC

The N bit counter generates an n bit digital output which is applied as an input to the DAC. The analog output corresponding to the digital input from DAC is compared with the input analog voltage using an opamp comparator. The opamp compares the two voltages and if the generated DAC voltage is less, it generates a high pulse to the N bit counter as a clock pulse to increment the counter. The same process will be repeated until the DAC output equals to the input analog voltage.

If the DAC output voltage is equal to the input analog voltage, then it generates low clock pulse and it also generates a clear signal to the counter and load signal to the storage resistor to store the corresponding digital bits. These digital values are closely matched with the input analog values with small quantization error. For every sampling interval the DAC output follows a ramp fashion so that it is called as Digital ramp type ADC. And this ramp looks like stair cases for every sampling time so that it is also called as staircase approximation type ADC



Conversion time of Counter type ADC

Conversion time of ADC is the time taken by the ADC to convert the input sampled analog value to digital value. Here the maximum conversion of high input voltage for a N bit ADC is the clock pulses required to the counter to count its maximum count value. So The maximum conversion of Counter type ADC is = (2^N-1) T Where, T is the time period of clock pulse.

If N=2 bit then the T_{max} = 3T.   Ts >= (2^N-1) T

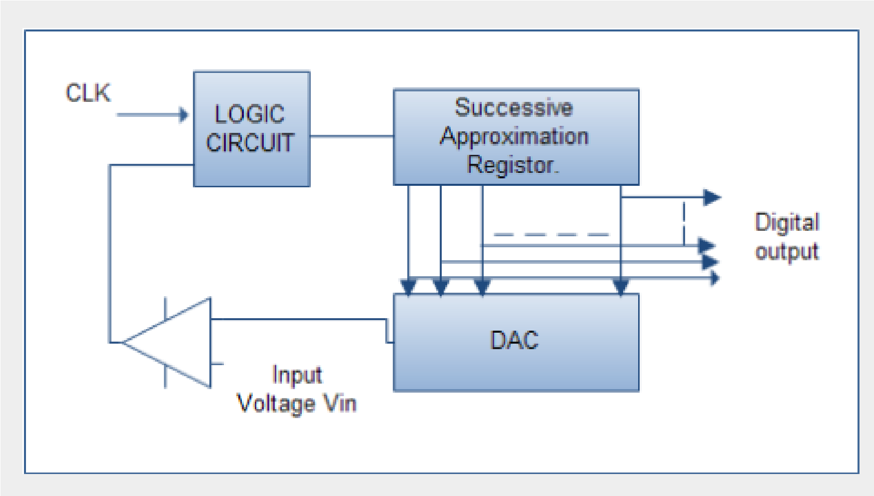
Advantages of Counter type ADC

Simple to understand and operate.

Cost is less because of less complexity in design.

Disadvantages or limitations of Counter type of ADC

Speed is less because every time the counter has to start from ZERO. There may be clash or aliasing effect if the next input is sampled before completion of one operation

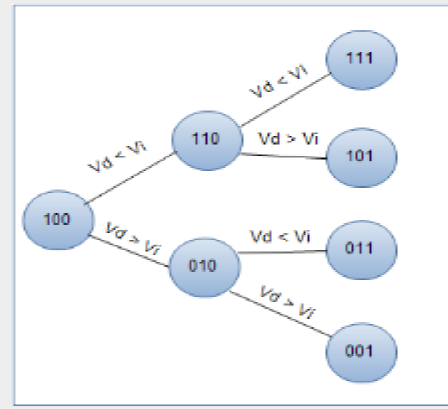


Successive Approximation ADC (Analog to Digital Converter)

Successive approximation ADC is the advanced version of Digital ramp type ADC which is designed to reduce the conversion and to increase speed of operation. The major draw of digital ramp ADC is the counter used to produce the digital output will be reset after every sampling interval. The normal counter starts counting from 0 and increments by one LSB in each count, this result in 2^N clock pulses to reach its maximum value. In successive approximation ADC the normal counter is replaced with successive approximation register as shown in below figure. The successive approximation register counts by changing the bits from MSB to LSB according to input. The detailed operation is shown below

Operation of 3 bit Successive Approximation ADC

The output of SAR is converted to analog out by the DAC and this analog output is compared with the input analog sampled value in the Opamp comparator. This Opamp provides an high or low clock pulse based on the difference through the logic circuit. In very first case the 3 bit SAR enables its MSB bit as high i.e. '1' and the result will be "100". This digita output is converted to analog value and compared with input sampled voltage (V_{in}). If the deference is positive i.e. if the sampled input is high then the SAR enables the next bit from MSB and result will be "110". Now if the output is negative i.e. if the input sampled voltage is less than the SAR resets the last set bit and sets the next bit and resultant output in this case will be "101" which will definitely approximately equal to the input analog value. The counting sequence is explained by the following counter flow chat as shown in below.



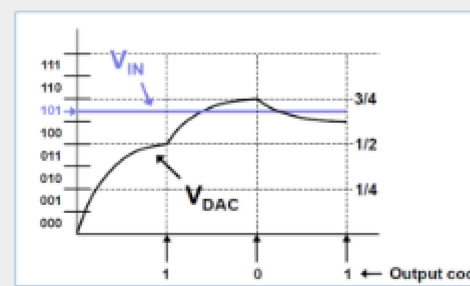
— Successive Approximation ADC Counter Flow chat

Conversion time of Successive Approximation ADC

By observing above 3 bit example it is illustrated for a 3 bit ADC the conversion time will be 3 clock pulses. Then; N bit Successive Approximation ADC conversion time = 3T (T- clock pulse). So to avoid aliasing effect the next sample of input signal should be taken after 3 clock pulses.

important note on Successive Approximation ADC

In Counter type or digital ramp type ADC the time taken for conversion depends on the magnitude of the input, but in SAR the conversion time is independent of the magnitude of the input sampled value.



— SAR ADC input output flow voltage graph

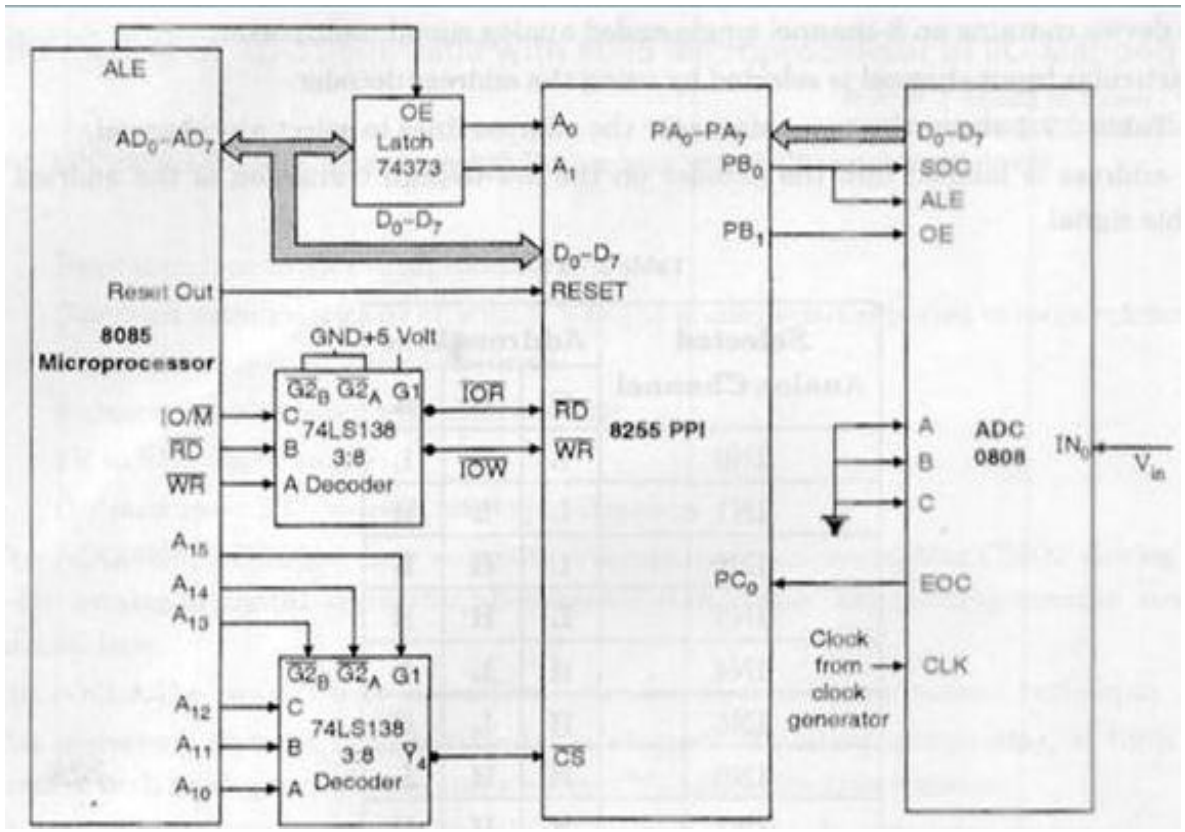
Advantages of Successive Approximation ADC

1. Speed is high compared to counter type ADC. 2. Good ratio of speed to power.
3. Compact design compared to Flash Type and it is inexpensive.

Disadvantages of Successive Approximation ADC

1. Cost is high because of SAR 2. Complexity in design.

Applications The SAR ADC will used widely data acquisition techniques at the sampling rates higher than 10KHz



ALE
AD_0-AD_7
OE
Latch
74373
D_0-D_7
Reset Out
8085
Microprocessor
GND+5 Volt
$\overline{G2}_B$ $\overline{G2}_A$ G1
IO/$\overline{M}$
$\overline{RD}$
$\overline{WR}$
C
B
A Decoder
74LS138
3:8
$\overline{IOR}$
$\overline{IOW}$
A_{15}
A_{14}
A_{13}
$\overline{G2}_B$ $\overline{G2}_A$ G1
A_{12}
A_{11}
A_{10}
C
B
A
74LS138
3:8
Decoder
$\overline{Y}_4$
A_0
A_1
D_0-D_7
RESET
$\overline{RD}$
$\overline{WR}$
8255 PPI
$\overline{CS}$
PA_0-PA_7
PB_0
PB_1
PC_0
D_0-D_7
SOC
ALE
OE
A
B
C
ADC
0808
IN_0
V_{in}
EOC
Clock from clock generator
CLK

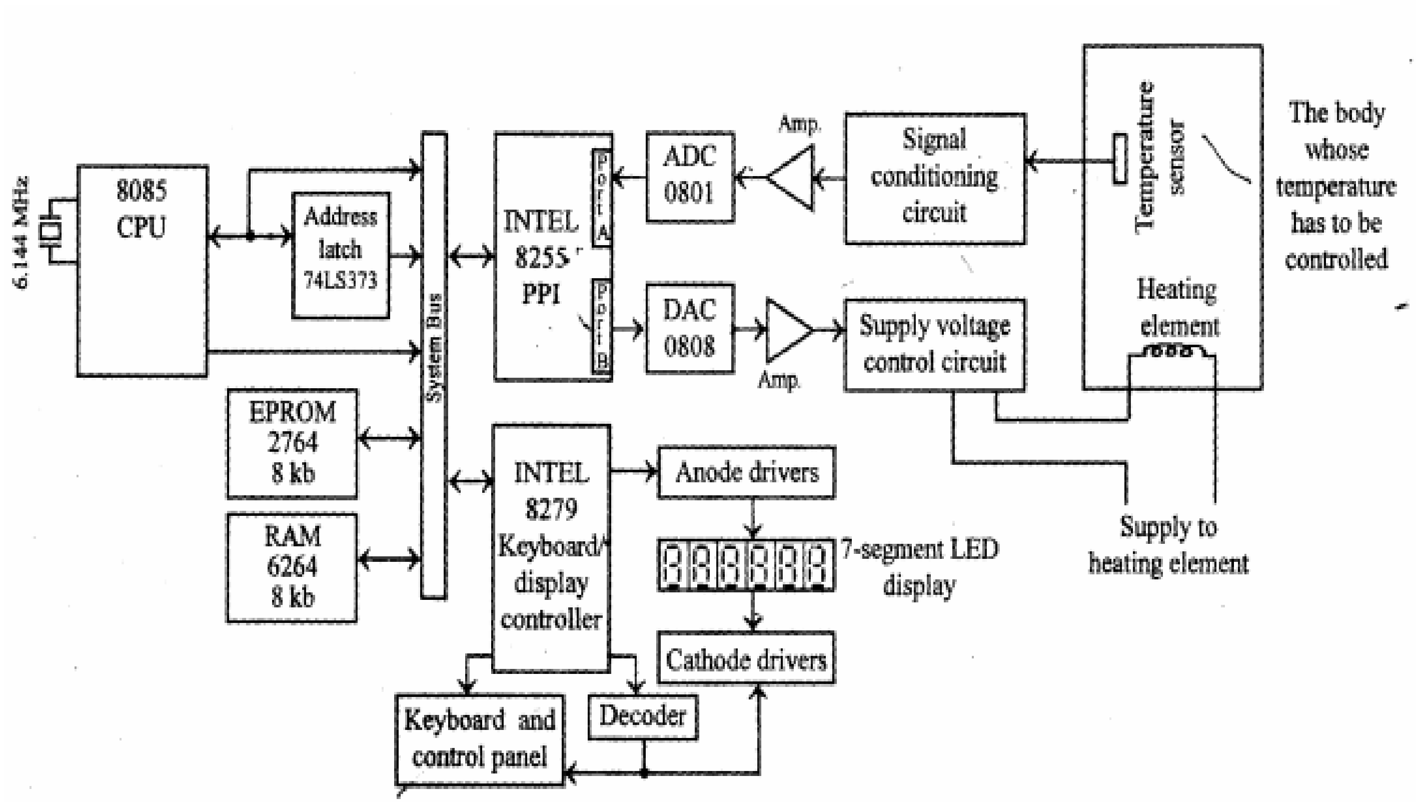
Block Diagram of Microprocessor Based Temperature Control

Transducer : For the measurement of physical quantities transducers are used. They convert them to electrical quantities. Here, for measuring temperature, sensors like thermocouple, thermistor, sensistor can be used.

Amplifier : If the electrical signal from transducer is small, it cannot be visualised or processed. Hence, it is amplified using amplifiers.

ADC : The electrical signal from transducer is an analog signal which a microprocessor cannot process. Hence, an analog to digital converter is used.

DAC : The signal from microprocessor will be digital signal which is going to control the analog elements like heater, cooler etc. For that the digital signal from the microprocessor has to be converted to analog by a DAC.



8085 Microprocessor based Temperature Control System

1. The system consist of 8085 microprocessor as CPU, EPROM & RAM memory for program & data storage, INTEL 8279 for keyboard and display interface, a keyboard and six numbers of 7-segment LEDs are interfaced to the system. ADC, DAC, INTEL 8255 for I/O ports, Amplifiers, Signal conditioning circuit, Temperature sensor and Supply control circuit. ·

2. The EPROM memory is provided for storing system program RAM memory for temporary data storage & stack operation.

3. The system has been designed to accept the desired temperature and various control commands through keyboard. ·

4. The 7-segment display has been provided to display the temperature of the body at any time instant. ·

5. The temperature of the body is measured using a temperature sensor. The different types of temperature sensors that can be used for temperature measurement are Thermo-couples, Thermistors, PNjunctions, IC sensors like AD590, etc. ·

6. The sensors will convert the input temperature to proportional analog voltage or current. The output signal of the sensor will be a weak signal and so it has to be amplified using high input impedance opamp. ·

7. The analog signal is scaled to suitable level by the signal conditioning circuit. ·

8. The microprocessor can process only digital signals and so the processor cannot read the analog signal from signal conditioning circuit directly. The system has an analog-to-digital converter (ADC) to convert the analog signal to proportional digital data. ·

9. In this system the ADC is interfaced to 8085 processor through port-A of 8255. The 8085 processor send signal to ADC to start conversion and at the end of conversion it read the digital data from the port-A of 8255. ·

10. The 8085 processor calculate the actual temperature using the input data and display it on the 7-segment LED. ·

11. Also, the processor compare the desired temperature with actual temperature (The operator can enter the desired temperature through keyboard) and calculate the error (the difference between actual temperature and desired temperature).

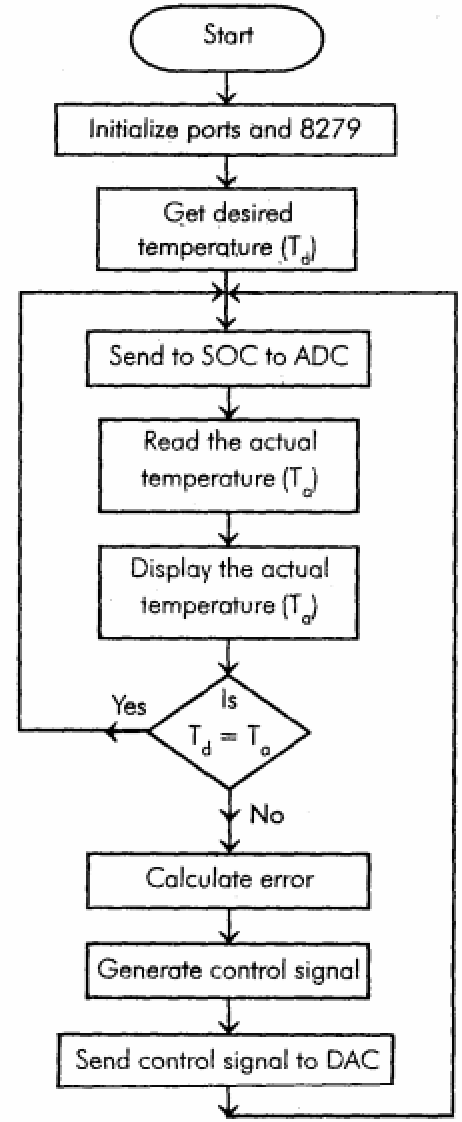
12. The error is used to compute a digital control signal, which is converted to analog control signal by DAC. The DAC is interfaced to the system through port-B of 8255.

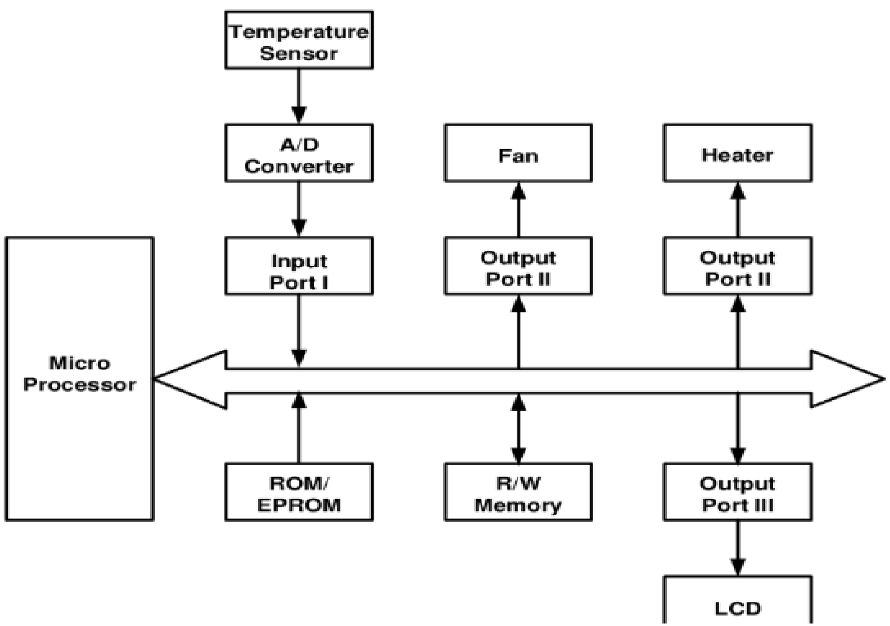
13. The analog control signal produced by DAC is used to control the power supply of the heating element of the body. ·

14. The digital control signal can be computed by the 8085 processor using different digital control algorithms (P/PL' PID/FUZZY logic control algorithms). ·

15. The control circuit for power supply can be either Thyristor - based circuit or relay. In case Thyristor control circuits the firing angle can be varied by the control signal to control the power input to the heater.

16. · In case of relay the control signal can switch ON/OFF the relay to control the power input to the heater.





For interfacing temperature control system with microprocessor, 8255 (PPI) and suitable ADC are connected between microprocessor and sensor output. Figure 5.10 shows the interface diagram of successive approximation A/D converter to the microprocessor through 8255 (PPI).

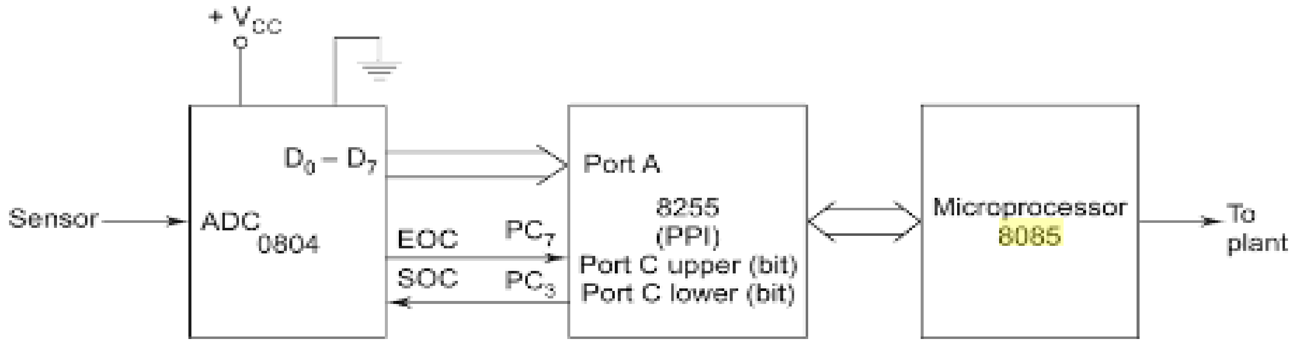


Fig. 5.10 Interfacing A/D converter with microprocessor

The microprocessor sends a Start of Conversion (SOC) signal to the A/D converter through Port C_{lower} of 8255. When A/D converter computes conversion, it sends as End of Conversion (EOC) signal to the microprocessor. Having received as EOC signal from A/D converter, the microprocessor reads the output of an A/D converter which is a digital quantity proportional to the temperature to be measured.

Algorithm:

Step 1. Initialize 8255 port A as an input port, port C_{upper} as an input port and port C_{lower} as an output port.

Step 2. Send SOC to an A/D converter (C_L port).

Step 3. Check for EOC from A/D converter (C_U port)

Step 4. Read the data from A/D converter

Step 5. Store the temperature (in °C) and corresponding digital voltage in memory locations.

Step 6. Compare the output of A/D converter with value in the memory location.

Step 7. Display the equivalent value (in °C)

Step 8. Repeat steps 2 to 7 for continuous temperature measurement and control.

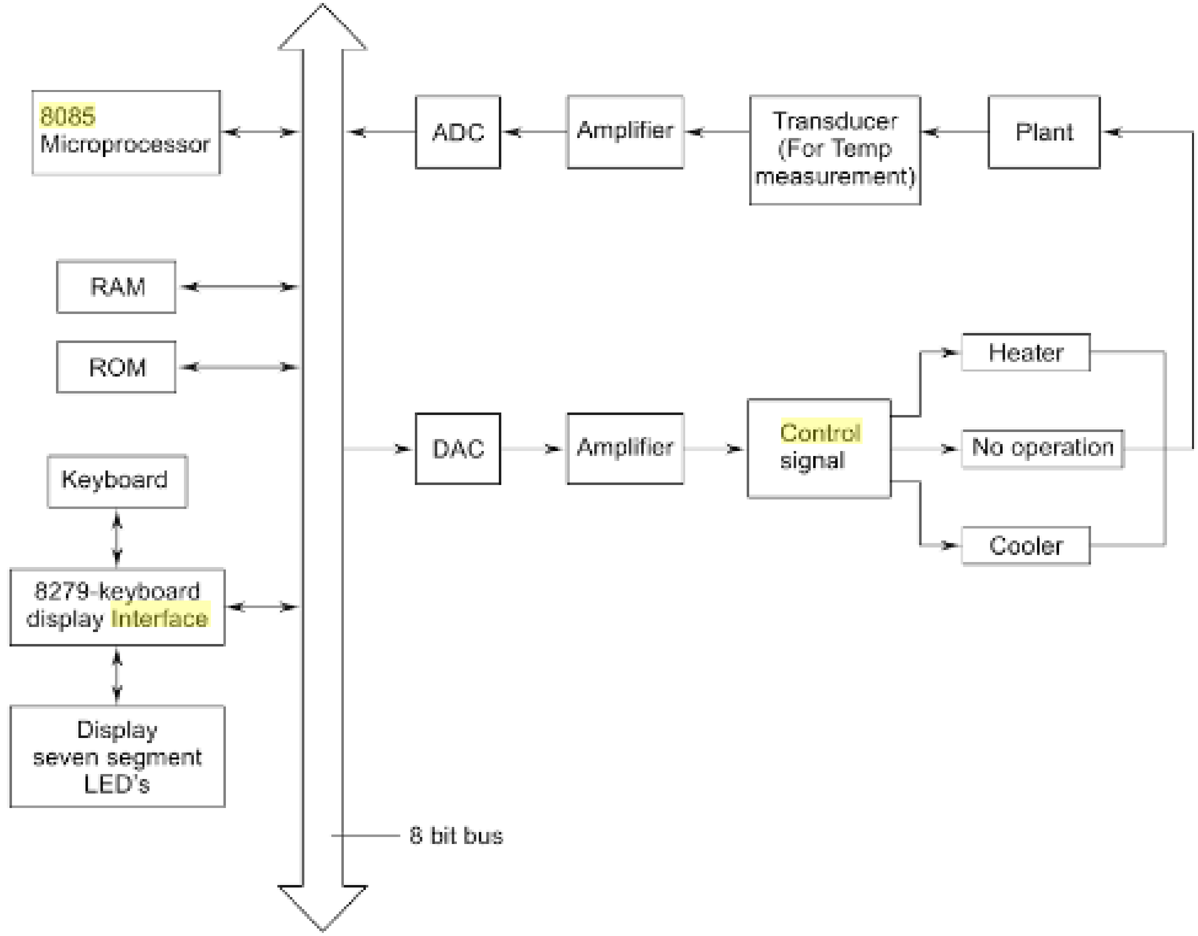


Fig. 5.9 8085 based temperature control system

Working : The plant may be an industry or an equipment or a furnace for which the temperature has to be monitored and maintained continuously at a particular temperature. The transducer measures current temperature and it is amplified by the amplifier which gets converted to hexadecimal digital value in ADC. This value is compared with the already set desired value in the microprocessor and the result of the comparison is as an error signal or null. The error may be positive meaning the temperature has to be increased by a heater arrangement. It may be negative meaning the temperature has to be reduced by some cooling set up. If there is no error, then no process has to be done and again the sensing activity has to be done repeatedly for continuous temperature maintenance.

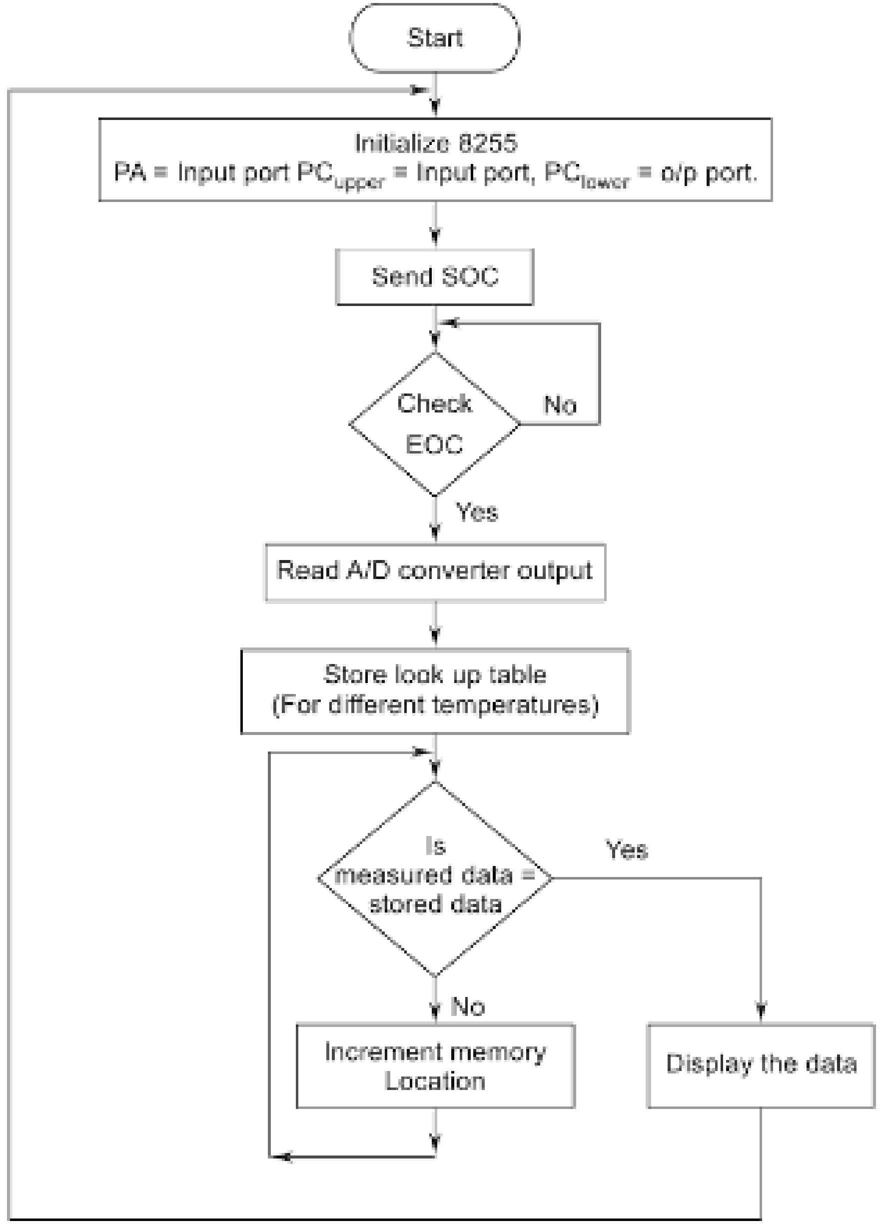


Fig. 5.11 Flow chart for temperature control system using 8085

4.10 PROGRAMS USING MICROPROCESSOR 8085

A program is a sequence of instructions to be executed for performing the task of a user. The program and data are fed into the microprocessor via input device and the program is stored in the memory of the microprocessor. Microprocessor has ALU which performs the necessary arithmetic and logic operations and the result is stored in memory which has to be displayed in output device. Micorprocessor instruction are written in Hex lode or by using Mnemonics. A source program written in the Mnemonic code is called assembly language and the software which translate this source program into machine language is called an assembler.

To write a program for performing a task, the following steps are required.

Step 1. Write as algorithm to solve the given problem. Algorithm is defined as a precise statement of the procedures required for solving a problem.

Step 2. Represent the steps in the algorithm and the sequence in the pictorial form called flow chart.

Step 3. Convert the blocks in the flow chart into 8085 operations which is called an Assembly Language Program.

Assembly Language Program has five sections :

1. Memory address : 16-bit addresses of the user to store the program.
2. Machine code : Hexadecimal numbers (instruction codes) to be entered.
3. Opcode : Opcodes are the symbols to indicate the operation of the function.
4. Operand : It is the data to be processed.
5. Comments : This play a critical role in the user's understanding of the logic behind a program.

5.7 REVIEW QUESTIONS

Part A

1. What are the different memory mapping schemes? Give one advantage for each.
2. Compare memory mapping types.
3. Differentiate I/O mapped I/O and memory mapped I/O.
4. What are IN and OUT instructions?
5. Explain about RIM and SIM instructions.
6. Define the term interfacing.
7. What is the role of microprocessor in stepper motor speed control?
8. What is the role of microprocessor in a traffic light controller?
9. How does 8085 control temperature variations of a particular system?
10. How many devices can be connected with 8085 in I/O mapped I/O?

Part B

1. Discuss in detail about I/O mapped I/O interfacing.
2. Explain briefly about memory mapped I/O technique with a neat example.
3. Interface a keyboard and 8 number of LED's with 8085 in I/O mapped I/O technique.
4. With neat block diagram explain temperature control system.
5. How does 8085 control speed of 4 phase stepper motor?

THANK YOU .

MAY GOD BLESS YOU ALL

FOR GOOD PROSPER

IN COMING YEARS