8086 Interfacing Examples

Interfacing To Alphanumeric Displays

- To give directions or data values to users, many microprocessor-controlled instruments and machines need to display letters of the alphabet and numbers. In systems where a large amount of data needs to be displayed a CRT is used to display the data. In system where only a small amount of data needs to be displayed, simple digit-type displays are often used.
- There are several technologies used to make these digit-oriented displays but we are discussing only the two major types.
- These are *light emitting diodes* (LED) and *liquid-crystal displays* (LCD).
- LCD displays use very low power, so they are often used in portable, battery- powered instruments. They do not emit their own light, they simply change the reflection of available light. Therefore, for an instrument that is to be used in low- light conditions, you have to include a light source for LCDs or use LEDs which emit their own light.
- Alphanumeric LED displays are available in three common formats. For displaying only number and hexadecimal letters, simple 7-segment displays such as that as shown in fig are used.

- To display numbers and the entire alphabet, 18 segment displays such as shown in fig or 5 by 7 dot-matrix displays such as that shown in fig can be used. The 7- segment type is the least expensive, most commonly used and easiest to interface with, so we will concentrate first on how to interface with this type.
- ***Directly Driving LED Displays:*** Figure shows a circuit that you might connect to a parallel port on a microcomputer to drive a single 7-segment , common-anode display. For a common-anode display, a segment is tuned on by applying a logic low to it.
- The 7447 converts a BCD code applied to its inputs to the pattern of lows required to display the number represented by the BCD code. This circuit connection is referred to as a *static display* because current is being passed through the display at all times.

- Each segment requires a current of between 5 and 30mA to light. Let's assume you want a current of 20mA. The voltage drop across the LED when it is lit is about 1.5V.
- The output low voltage for the 7447 is a maximum of 0.4V at 40mA. So assume that it is about 0.2V at 20mA. Subtracting these two voltage drop from the supply voltage of 5V leaves 3.3V across the current limiting resistor. Dividing 3.3V by 20mA gives a value of 168Ω for the current-limiting resistor. The voltage drops across the LED and the output of 7447 are not exactly predictable and exact current through the LED is not critical as long as we don't exceed its maximum rating.

Software-Multiplexed LED Display:

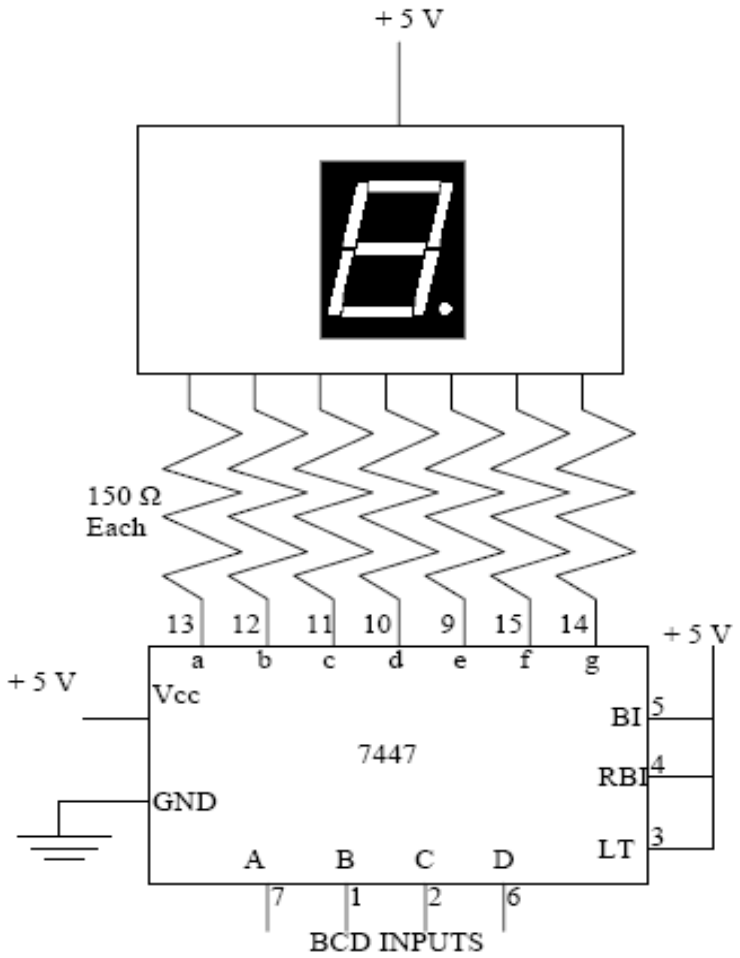
- The circuit in fig works for driving just one or two LED digits with a parallel output port. However, this scheme has several problem if you want to drive, eight digits.
- The first problem is power consumption. For worst-case calculations, assume that all 8 digits are displaying the digit 8, so all 7 segments are all lit. Seven segment time 20mA per segment gives a current of 140mA per digit. Multiplying this by 8 digits gives a total current of 1120mA or 1.12A for 8 digits.
- A second problem of the static approach is that each display digit requires a separate 7447 decoder, each of which uses of another 13mA. The current required by the decoders and the LED displays might be several times the current required by the reset of the circuitry in the instrument.

- To solve the problem of the static display approach, we use a *multiplex method*, example for an explanation of the multiplexing.
- The fig shows a circuit you can add to a couple of microcomputer ports to drive some common anode LED displays in a multiplexed manner. The circuit has only one 7447 and that the segment outputs of the 7447 are bused in parallel to the segment inputs of all the digits.
- The question that may occur to you on first seeing this is: Aren’t all the digits
- going to display the same number? The answer is that they would if all the digits were turned on at the same time. The tricky of multiplexing displays is that only one display digit is turned on at a time.

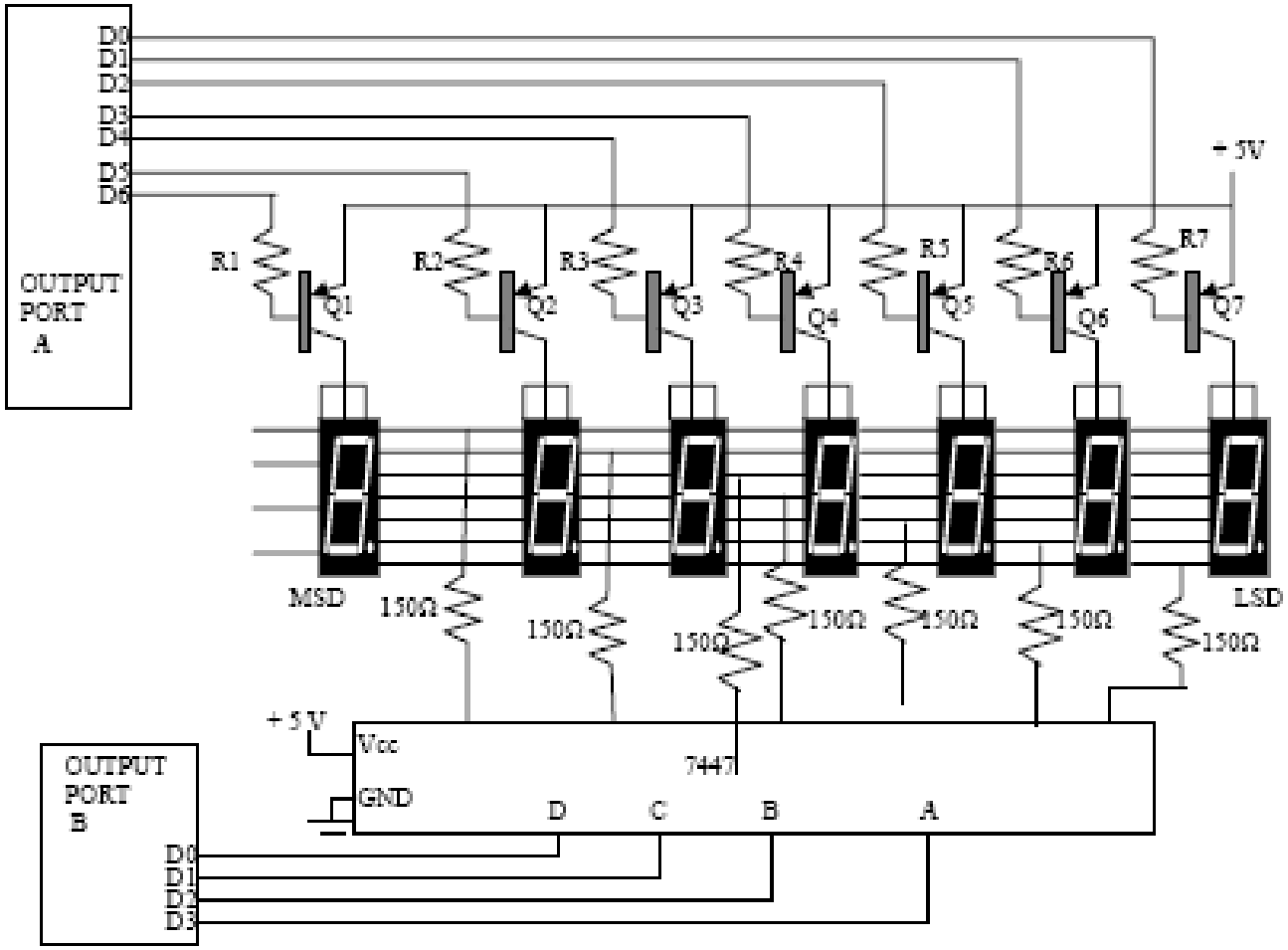
- The PNP transistor is series with the common anode of each digit acts as on/off switch for that digit. Here's how the multiplexing process works.
- The BCD code for digit 1 is first output from port B to the 7447. the 7447 outputs the corresponding 7-segment code on the segment bus lines. The transistor connected to digit 1 is then turned on by outputting a low to the appropriate bit of port A. All the rest of the bits of port A are made high to make sure no other digits are turned on. After 1 or 2 ms, digit 1 is turned off by outputting all highs to portA.
- The BCD code for digit 2 is then output to the 7447 on port B, and a word to turn on digit 2 is output on port A.

- After 1 or 2 ms, digit 2 is turned off and the process is repeated for digit 3. the process is continued until all the digits have had a turn. Then digit 1 and the following digits are lit again in turn.
- A procedure which is called on an interrupt basis every 2ms to keep these displays refreshed wit some values stored in a table. With 8 digits and 2ms per digit, you get back to digit 1 every 16ms or about 60 times a second.
- This refresh rate is fast enough so that the digits will each appear to be lit all time. Refresh rates of 40 to 200 times a second are acceptable.

- The immediately obvious advantages of multiplexing the displays are that only one 7447 is required, and only one digit is lit at a time. We usually increase the current per segment to between 40 and 60 mA for multiplexed displays so that they will appear as bright as they would if they were not multiplexed. Even with
- this increased segment current, multiplexing gives a large saving in power and parts.
- The software-multiplexed approach we have just described can also be used to drive 18-segment LED devices and dot-matrix LED device. For these devices, however you replace the 7447 in fig with ROM which generates the required segment codes when the ASCII code for a character is applied to the address inputs of the ROM.



Circuit for driving single 7-segment LED display with 7447


D0
D1
D2
D3
D4
D5
D6
OUTPUT
PORT
A
+ 5V
R1
R2
R3
R4
R5
R6
R7
Q1
Q2
Q3
Q4
Q5
Q6
Q7
MSD
LSD
150Ω
150Ω
150Ω
150Ω
150Ω
150Ω
150Ω
+ 5 V
Vcc
GND
7447
D
C
B
A
OUTPUT
PORT
B
D0
D1
D2
D3

Liquid Crystal Display

- Liquid Crystal displays are created by sandwiching a thin 10-12 µm layer of a liquid-crystal fluid between two glass plates. A transparent, electrically
- conductive film or backplane is put on the rear glass sheet. Transparent sections of conductive film in the shape of the desired characters are coated on the front glass plate.
- When a voltage is applied between a segment and the backplane, an electric field is created in the region under the segment. This electric field changes the transmission of light through the region under the segment film.
- There are two commonly available types of LCD : dynamic scattering and field- effect.
- The Dynamic scattering types of LCD: It scrambles the molecules where the field is present. This produces an etched-glass-looking light character on a dark background.

- Field-effect types use polarization to absorb light where the electric field is present. This produces dark characters on a silver- gray background.
- Most LCD's require a voltage of 2 or 3 V between the backplane and a segment to turn on the segment.
- We cannot just connect the backplane to ground and drive the segment with the outputs of a TTL decoder. The reason for this is a steady dc voltage of more than about 50mV is applied between a segment and the backplane.
- To prevent a dc buildup on the segments, the segment-drive signals for LCD must
- be square waves with a frequency of 30 to 150 Hz.

- Even if you pulse the TTL decoder, it still will not work because the output low voltage of TTL devices is greater than 50mV.
  - CMOS gates are often used to drive LCDs.
  - The Following fig shows how two CMOS gate outputs can be connected to drive an LCD segment and backplane.
- The off segment receives the same drive signal as the backplane. There is never any voltage between them, so no electric field is produced. The waveform for the on segment is 180 out of phase with the backplane signal, so the voltage between this segment and the backplane will always be +V.
- The logic for this signal, a square wave and its complement. To the driving gates, the segment-backplane sandwich appears as a somewhat leaky capacitor.

- The CMOS gates can be easily supply the current required to charge and discharge this small capacitance.
- Older inexpensive LCD displays turn on and off too slowly to be multiplexed the way we do LED display.
- At 0c some LCD may require as mush as 0.5s to turn on or off. To interface to those types we use a nonmultiplexed driver device.
- More expensive LCD can turn on and off faster, so they are often multiplexed using a variety of techniques.
- In the following section we show you how to interface a nonmultiplexed LCD to a microprocessor such as SDK-86.
- Intersil ICM7211M can be connected to drive a 4-digit, nonmultiplexed, 7- segment LCD display.

- The 7211M input can be connected to port pins or directly to microcomputer bus. We have connected the CS inputs to the Y2 output of the 74LS138 port decoder.

- According to the truth table the device will then be addressable as ports with a base address of FF10H. SDK-86 system address lines A2 is connected to the digit-select input (DS2) and system address lines A1 is connected to the DS1 input. This gives digit 4 a system address of FF10H.

A8-A15	A5-A7	A4	A3	A2	A1	A0	M/IO	Y Output Selected	System Base Address	Device
1	0	0	0	X	X	0	0	00	F F 0 0	8259A #1
1	0	0	1	X	X	0	0	1	F F 0 8	8259A #2
1	0	1	0	X	X	0	0	2	F F 1 0	
1	0	1	1	X	X	0	0	3	F F 1 8	
1	0	0	0	X	X	1	0	4	F F 0 1	8254
1	0	0	1	X	X	1	0	5	F F 0 9	
1	0	1	0	X	X	1	0	6	F F 1 1	
1	0	1	1	X	X	1	0	7	F F 1 9	
ALL OTHER STATES								NONE		

Fig : Truth table for 74LS138 address decoder

- Digit 3 will be addressed at FF12H, digit 2 at FF14H and digit 1 at FF16H.
- The data inputs are connected to the lower four lines of the SDK-86 data bus. The oscillator input is left open. To display a character on one of the digits, you simply keep the 4-bit hex code for that digit in the lower 4 bits of the AL register and output it to the system address for that digit.
- The ICM7211M converts the 4-bit hex code to the required 7-segment code.
- The rising edge of the CS input signal causes the 7-segment code to be latched in the output latches for the address digit.
- An internal oscillator automatically generates the segment and backplane drive waveforms as in fig . For interfacing with the LCD displays which can be multiplexed the Intersil ICM7233 can be use.

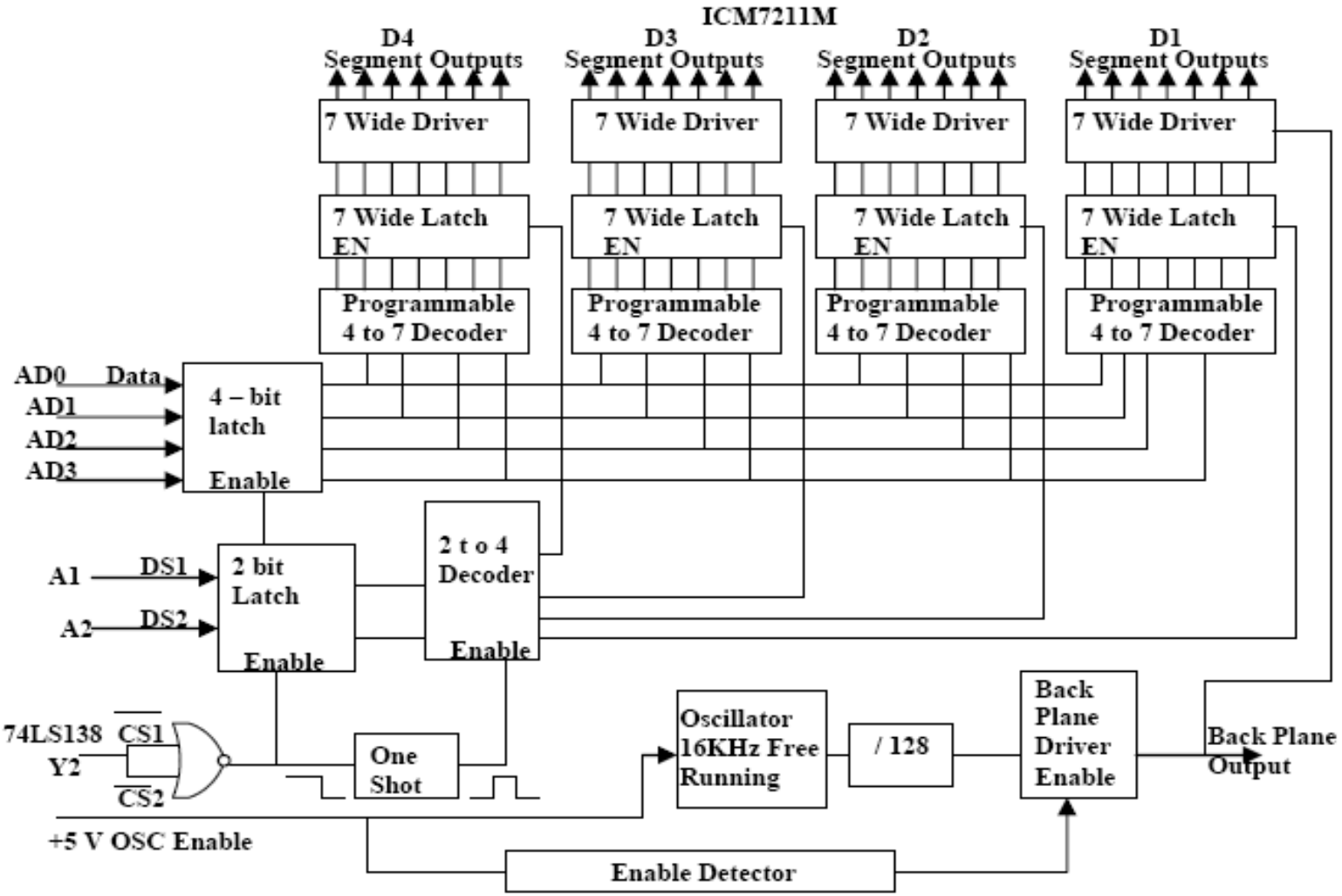


Fig : Circuit for interfacing four LCD digits to an SDK-86 bus using ICM7211M

Interfacing Analog to Digital Data Converters

- In most of the cases, the PIO 8255 is used for interfacing the analog to digital converters with microprocessor.
- We have already studied 8255 interfacing with 8086 as an I/O port, in previous section. This section we will only emphasize the interfacing techniques of analog to digital converters with 8255.
- The analog to digital converters is treaded as an input device by the microprocessor, that sends an initialising signal to the ADC to start the analogy to digital data conversation process. The start of conversation signal is a pulse of a specific duration.
- The process of analog to digital conversion is a slow process, and the microprocessor has to wait for the digital data till the conversion is over. After the conversion is over, the ADC sends end of conversion EOC signal to inform the microprocessor that the conversion is over and the result is ready at the output buffer of the ADC. These tasks of issuing an SOC pulse to ADC, reading EOCsignal from the ADC and reading the digital output of the ADC are carried out by the CPU using 8255 I/O ports.

- The time taken by the ADC from the active edge of SOC pulse till the active edge of EOC signal is called as the conversion delay of the ADC.
- It may range any where from a few microseconds in case of fast ADC to even a few hundred milliseconds in case of slow ADCs.
- The available ADC in the market use different conversion techniques for conversion of analog signal to digitals. Successive approximation techniques and dual slope integration techniques are the most popular techniques used in the integrated ADC chip.
- General algorithm for ADC interfacing contains the following steps:

1. Ensure the stability of analog input, applied to the ADC.
2. Issue start of conversion pulse to ADC
3. Read end of conversion signal to mark the end of conversion processes.
4. Read digital data output of the ADC as equivalent digital output.
5

5. Analog input voltage must be constant at the input of the ADC right from the start of conversion till the end of the conversion to get correct results. This may be ensured by a sample and hold circuit which samples the analog signal and holds it constant for a specific time duration. The microprocessor may issue a hold signal to the sample and hold circuit.
6. If the applied input changes before the complete conversion process is over, the digital equivalent of the analog input calculated by the ADC may not be correct.

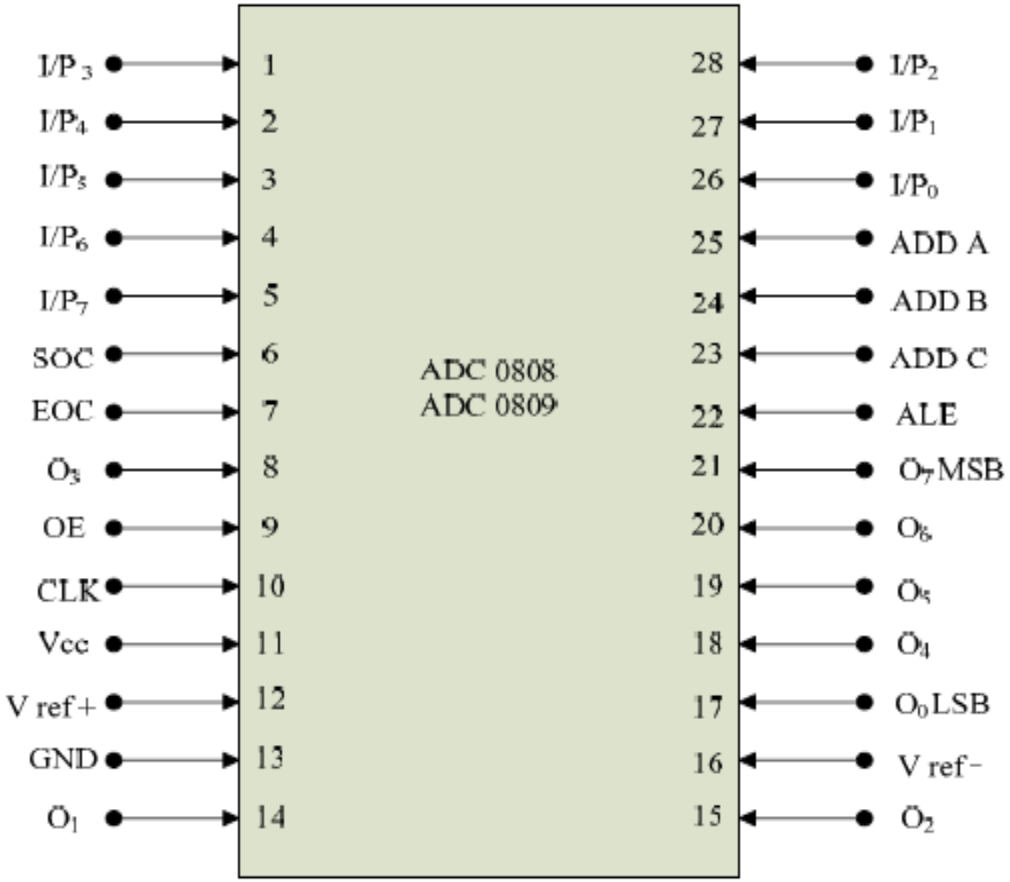
ADC 0808/0809 :

- The analog to digital converter chips 0808 and 0809 are 8-bit CMOS, successive approximation converters. This technique is one of the fast techniques for analog to digital conversion. The conversion delay is 100µs at a clock frequency of 640 KHz, which is quite low as compared to other converters. These converters do not need any external zero or full scale adjustments as they are already taken care of by internal circuits.
- These converters internally have a 3:8 analog multiplexer so that at a time eight different analog conversion by using address lines - ADD A, ADD B, ADD C. Using these address inputs, multichannel data acquisition system can be designed using a single ADC. The CPU may drive these lines using output port lines in case of multichannel applications. In case of single input applications, these may be hardwired to select the proper input.
- There are unipolar analog to digital converters, i.e. they are able to convert only positive analog input voltage to their digital equivalent. These chips do no contain any internal sample and hold circuit.

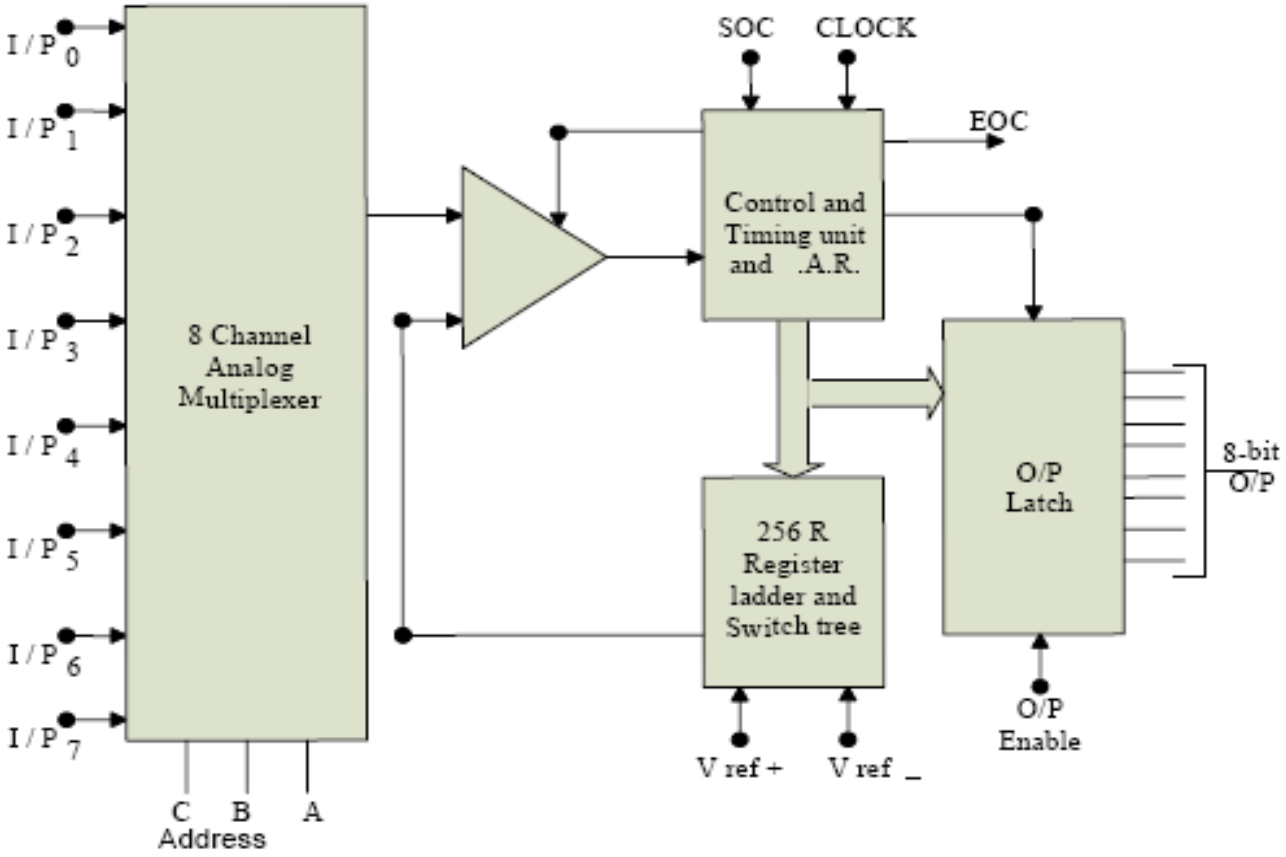
Analog /P selecte	Address lines		
	C	B	A
I/P_0	0	0	0
I/P_1	0	0	1
I/P_2	0	1	0
I/P_3	0	1	1
I/P_4	1	0	0
I/P_5	1	0	1
I/P_6	1	1	0
I/P_7	1	1	1

- If one needs a sample and hold circuit for the conversion of fast signal into equivalent digital quantities, it has to be externally connected at each of the analog inputs.

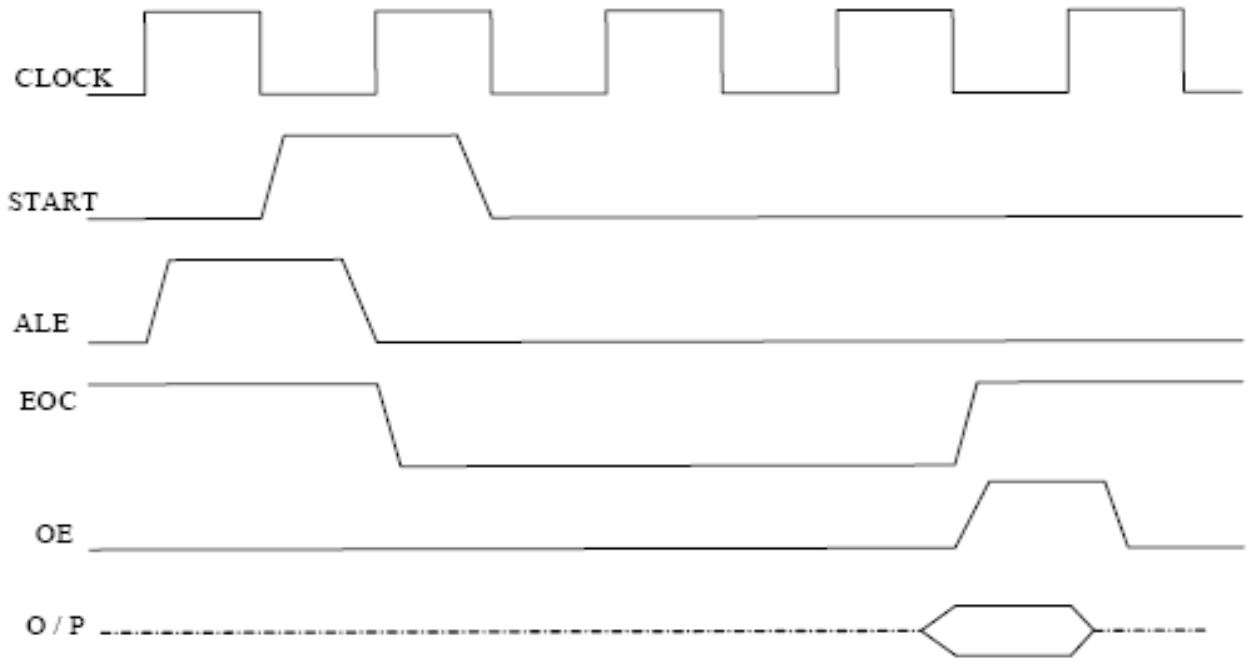
Vcc	Supply pins +5V
GND	GND
Vref +	Reference voltage positive +5 Volts maximum.
Vref _	Reference voltage negative 0Volts minimum.
I/P0 –$\bar{I}$/P7	Analog inputs
ADD A,B,C	Address lines for selecting analog inputs.
O7 – O0	Digital 8-bit output with O7 MSB and O0 LSB
SOC	Start of conversion signal pin
EOC	End of conversion signal pin
OE	Output latch enable pin, if high enables output
CLK	Clock input for ADC



PIN DIAGRAM OF ADC 0808 / 0809



Block Diagram of ADC 0808 / 0809



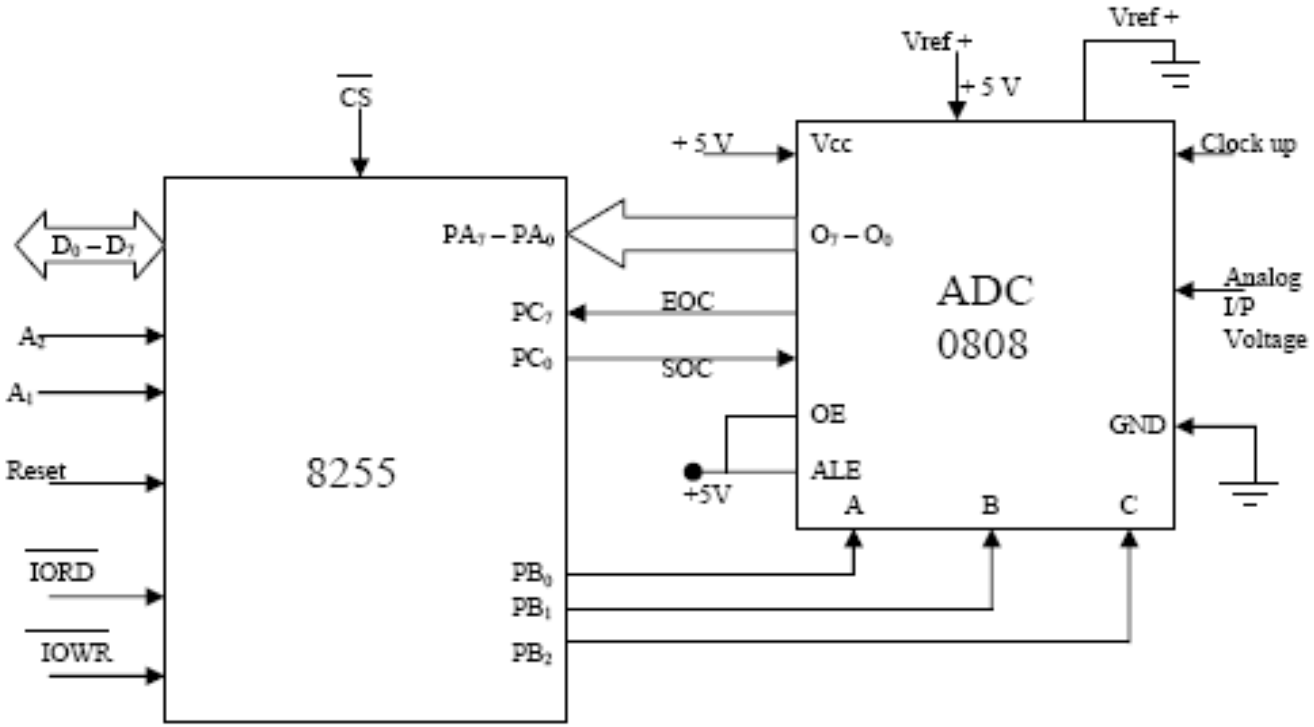
Timing Diagram of ADC 0808

- ***Example:*** Interfacing ADC 0808 with 8086 using 8255 ports. Use port A of 8255 for transferring digital data output of ADC to the CPU and port C for control signals. Assume that an analog input is present at I/P2 of the ADC and a clock input of suitable frequency is available for ADC.
- **Solution**: The analog input I/P2 is used and therefore address pins A,B,C should be 0,1,0 respectively to select I/P2. The OE and ALE pins are already kept at +5V to select the ADC and enable the outputs. Port C upper acts as the input port to receive the EOC signal while port C lower acts as the output port to send SOC to the ADC.
- Port A acts as a 8-bit input data port to receive the digital data output from the ADC. The 8255 control word is written as follows:

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	0	0

- The required ALP is as follows:

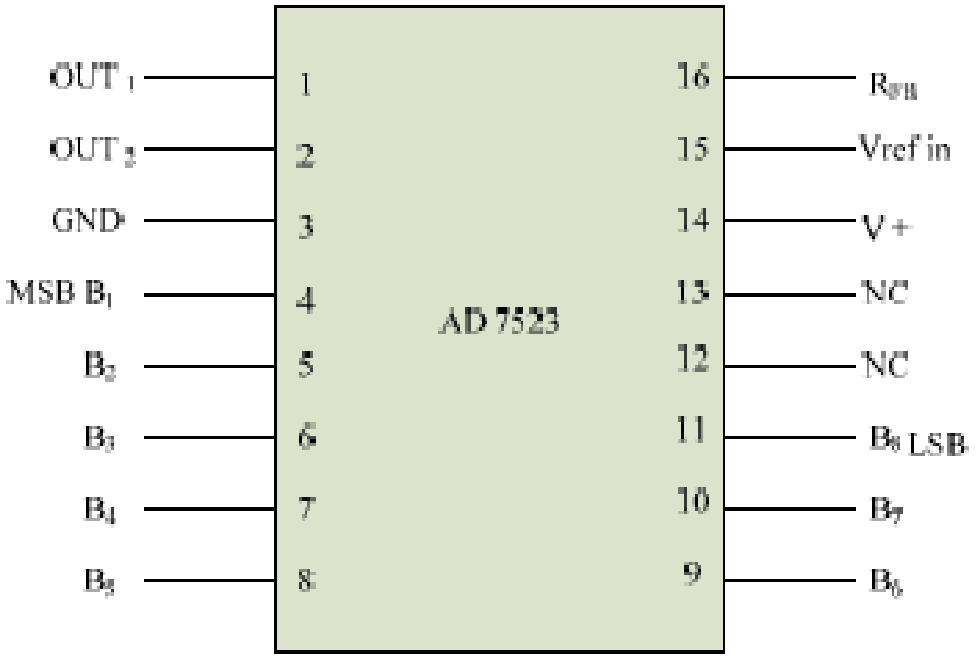
```
        MOV   AL, 98h        ;initialise 8255 as
        OUT   CWR, AL        ;discussed above.
        MOV   AL, 02h        ;Select I/P2 as analog
        OUT   Port B, AL     ;input.
        MOV   AL, 00h        ;Give start of conversion
        OUT   Port C, AL     ; pulse to the ADC
        MOV   AL, 01h
        OUT   Port C, AL
        MOV   AL, 00h
        OUT   Port C, AL
WAIT:   IN    AL, Port C     ;Check for EOC by

        RCR                  ; reading port C upper and
        JNC   WAIT           ;rotating through carry.
        IN    AL, Port A     ;If EOC, read digital equivalent in AL
        HLT                  ;Stop.
```



Interfacing 0808 with 8086

Interfacing Digital To Analog Converters

- *INTERFACING DIGITAL TO ANALOG CONVERTERS*: The digital to analog converters convert binary number into their equivalent voltages. The DAC find applications in areas like digitally controlled gains, motors speed controls, programmable gain amplifiers etc.
- AD 7523 8-bit Multiplying DAC : This is a 16 pin DIP, multiplying digital to analog converter, containing R-2R ladder for D-A conversion along with single pole double thrown NMOS switches to connect the digital inputs to the ladder.



Pin Diagram of AD 7523

- The pin diagram of AD7523 is shown in fig the supply range is from +5V to +15V, while Vref may be any where between -10V to +10V. The maximum analog output voltage will be any where between -10V to +10V, when all the digital inputs are at logic high state.
- Usually a zener is connected between OUT1 and OUT2 to save the DAC from negative transients. An operational amplifier is used as a current to voltage converter at the output of AD to convert the current out put of AD to a proportional output voltage.
- It also offers additional drive capability to the DAC output. An external feedback resistor acts to control the gain. One may not connect any external feedback resistor, if no gain control is required.

- ***EXAMPLE***: Interfacing DAC AD7523 with an 8086 CPU running at 8MHZ and write an assembly language program to generate a sawtooth waveform of period 1ms with Vmax 5V.
- Solution: Fig shows the interfacing circuit of AD 74523 with 8086 using 8255. program gives an ALP to generate a sawtooth waveform using circuit.

```
ASSUME     CS:CODE
CODE SEGMENT
START      :MOV AL,80h        ;make all ports output
           OUT  CW, AL
AGAIN      :MOV AL,00h        ;start voltage for ramp
BACK :     OUT  PA, AL
           INC  AL
           CMP  AL, 0FFh
           JB   BACK
           JMP  AGAIN
CODE ENDS
END        START
```

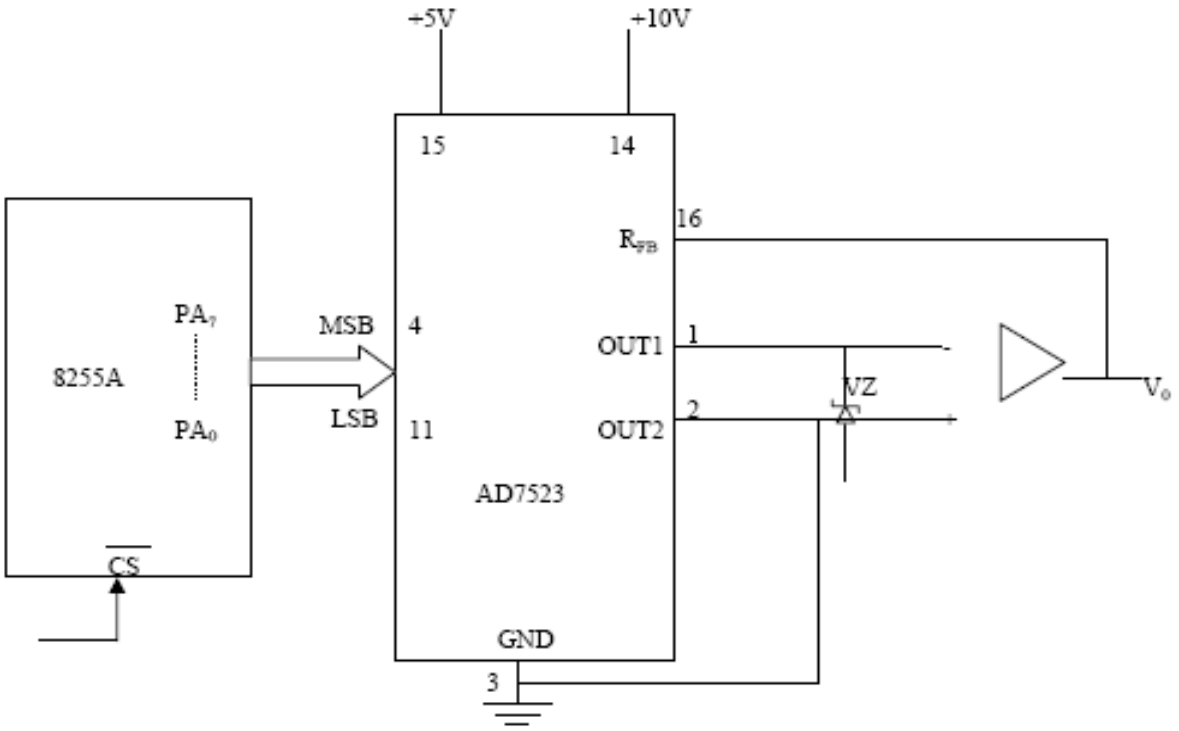


Fig: Interfacing of AD7523

- In the above program, port A is initialized as the output port for sending the digital data as input to DAC. The ramp starts from the 0V (analog), hence AL starts with 00H. To increment the ramp, the content of AL is increased during each execution of loop till it reaches F2H.
- After that the saw tooth wave again starts from 00H, i.e. 0V(analog) and the procedure is repeated. The ramp period given by this program is precisely 1.000625 ms. Here the count F2H has been calculated by dividing the required delay of 1ms by the time required for the execution of the loop once. The ramp slope can be controlled by calling a controllable delay after the OUT instruction.