Interfacing Analog to Digital Data Converters: (ADC)

→ We can interface 8 bit and 12 bit analog to digital converters with 8086.

→ We use PIO 8255 can be used to interface ADC's with 8086 microprocessor

→ The ADC is treated as an input device by the microprocessor, that sends an initializing signal to ADC to start the analog to digital conversion process.

→ The start of conversion signal is a pulse of a specific duration.

The process of A/D 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 about it and the result is ready at the output buffer of ADC.

Scanned with CamScanner

→ These tasks of issuing an SOC signal to ADC, and reading EOC signal from 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 pulse is called as the conversion delay of ADC.
It may range anywhere from a few microseconds in case of fast ADCs, to even a few hundred milliseconds in case of slow ADCs.

→ The selection of ADC required for a particular application is done, depending on required speed, resolution and cost factor.

→ The available ADCs in the market use different techniques for the conversion of analog signals to digital signals.
Successive approximation and dual slope integration techniques are the most popular techniques used in integrated ADC chips.

Scanned with CamScanner

→ A general algorithm for interfacing an ADC contains the following steps:

(1) Ensure the stability of analog input, applied to the ADC.

(2) Issue start of conversion (SOC) pulse to ADC.

(3) Read end of conversion (EOC) pulse to mark the end of conversion process.

Scanned with CamScanner

ADC 0808/0809 :

→ The ADC chips 0808/0809 are 8 bit cmos, successive approximation converters.

→ Successive approximation is one of the fastest technique used for the process of analog to digital conversion.

→ The conversion delay is 100 μs 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 taken care by internal circuits.

Scanned with CamScanner

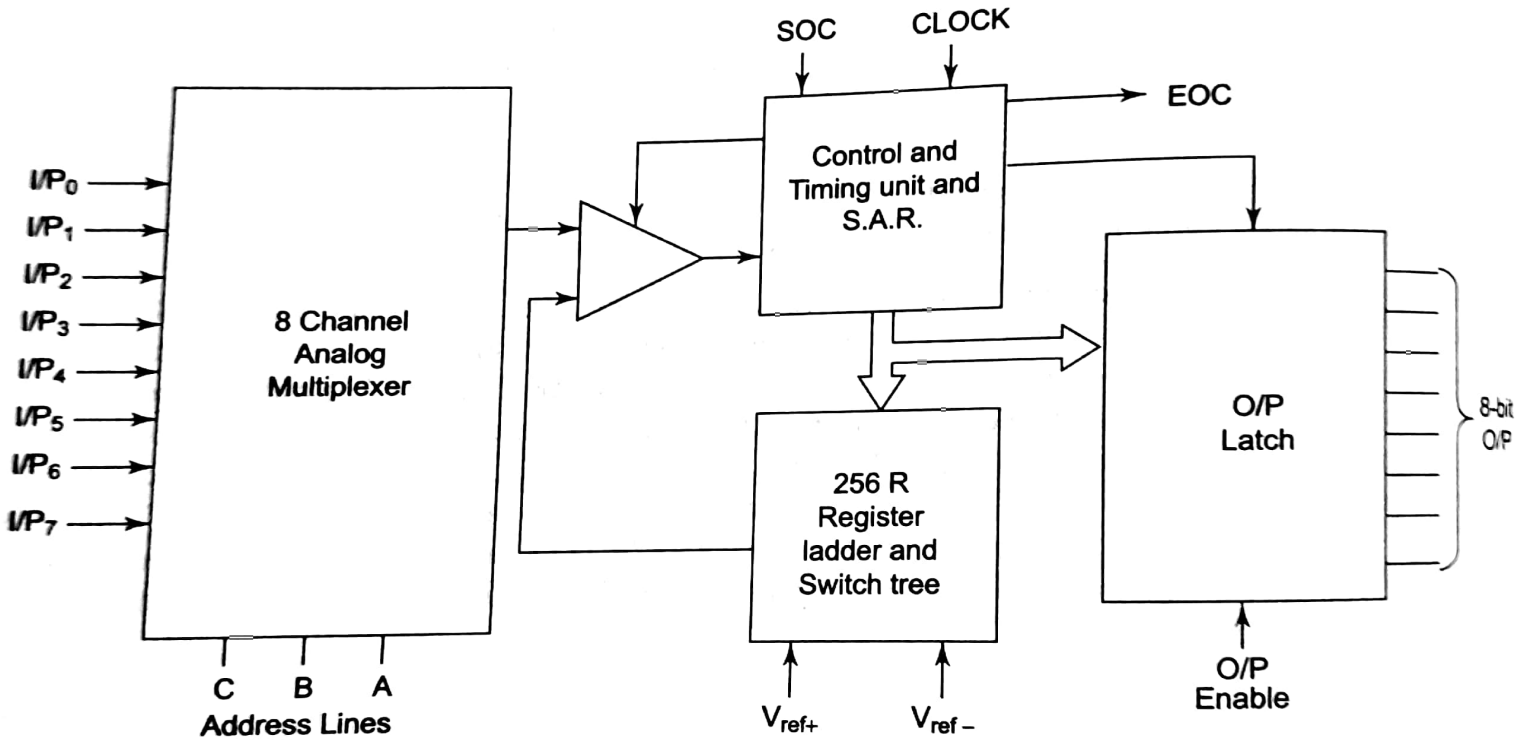
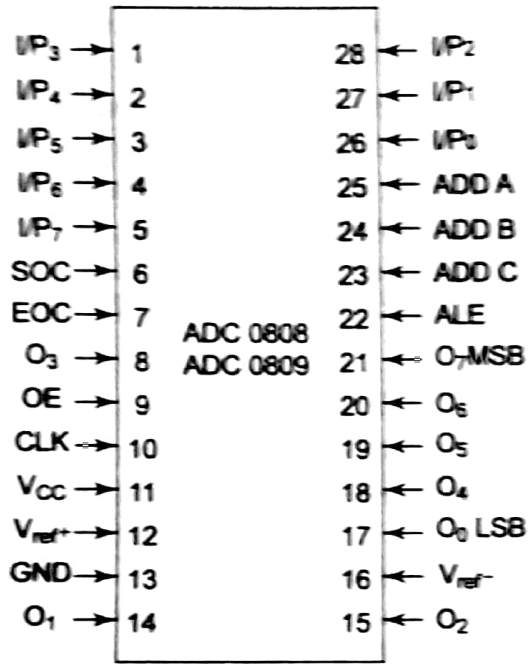


Fig. 5.37(a) *Block Diagram of ADC 0808/0809*

Scanned with CamScanner



I/P_0–I/P_7	Analog inputs
ADD A, B, C	Address lines for selecting analog inputs
O_7–O_0	Digital 8-bit output with O_7 MSB and O_0 LSB
SOC	Start of conversion signal pin
EOC	End of conversion signal pin
OE	Output latch enable pin, if high enable output
CLK	Clock input for ADC
V_{CC}, GND	Supply pins +5V and GND
V_{ref+} and V_{ref-}	Reference voltage positive (+5 Volts maximum) and Reference voltage negative (0V minimum)

Fig. 5.37(b) ***Pin Diagram of ADC 0808/0809***

Scanned with CamScanner

→ These converters have a 3:8 analog multiplexer so that at a time eight different analog inputs can be connected to the chip. Out of these eight inputs, only one can be selected for conversion by using address lines ADD A, ADD B and ADD C as shown.

The CPU can drive these lines using output port lines in case of multichannel applications.

In case of single input applications, these may be hardwired to select proper input.

Analog I/P selected	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

Scanned with CamScanner

→ These are unipolar ADC i.e. they are able to convert only positive analog input voltages to their digital equivalents.

→ These chips do not contain any internal sample and hold circuit.

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

Scanned with CamScanner

Ex:2

Interface 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 analog input is present at I/P 2 of ADC and a clock input of suitable frequency is available for ADC.

Draw the schematic and write the required ALP.

Sol

The analog input I/P 2 is used and therefore address pins ABC should be 010 respectively to select I/P 2.

The OE and ALE are already kept at +5V to select ADC and enable the output.

Port C upper acts as input port to receive EOC signal.

PORT C lower acts as output port to send SOC signal to ADC.

Port A acts as 8 bit input data port to receive digital data output from ADC.

Scanned with CamScanner

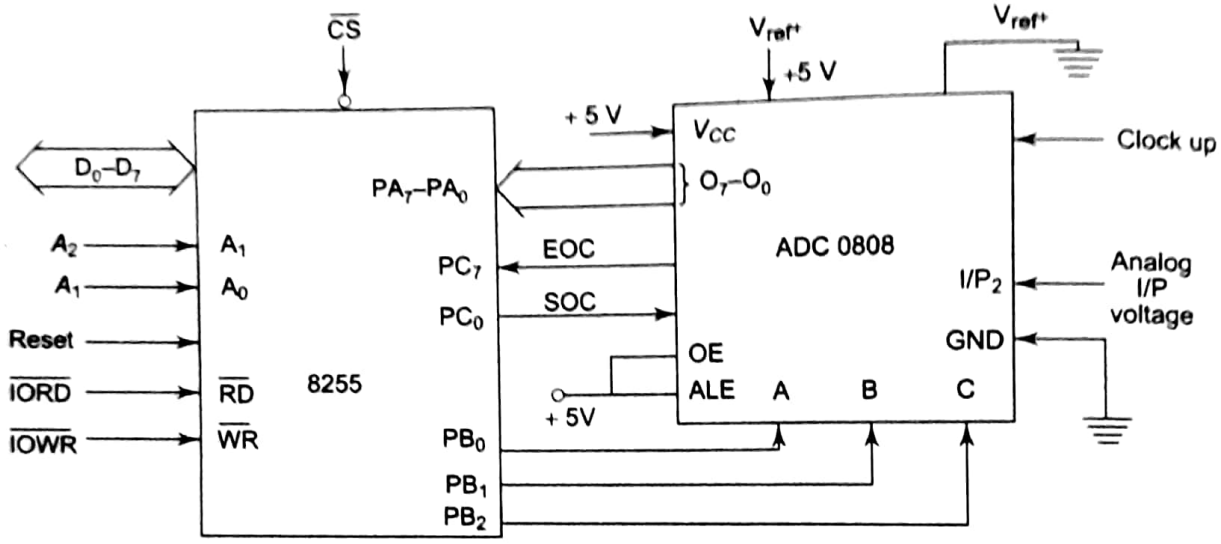


Fig. 5.39 *Interfacing 0808 with 8086*

Scanned with CamScanner

Solution Figure 5.39 shows the interfacing connections of ADC0808 with 8086 using 8255. The analog input I/P_2 is used and therefore address pins A,B,C should be 0,1,0 respectively to select I/P_2. 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:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Control word
1	0	0	1	1	0	0	0	= 98 H

The required ALP is given as follows:

```
        MOV AL,98 H              ; 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,01 H              ;
        OUT PORT C,AL            ;
        MOV AL,00H               ;
        OUT PORT C,AL            ;
WAIT :  IN AL,PORTC              ; Check for EOC by
        RCL                      ; reading port C upper and
        JNC WAIT                 ; rotating through carry.
        IN AL,PORTA              ; If EOC, read digital equivalent in
                                      AL
        HLT                      ; Stop
```

Program 5.11 *ALP for Problem 5.16*

Scanned with CamScanner

Interfacing Digital to Analog converters (DACs) :

→ The DAC converts binary numbers into their equivalent analog voltage.

→ The DAC finds applications in areas like digitally controlled gains, motor speed controls, programmable gain amplifiers etc.

Scanned with CamScanner

AD 7523 8 bit multiplying DAC :

→ Intersil's AD 7523 is a 16 pin DIP, multiplying DAC, containing R-2R (R=10k) for digital to analog conversion along with single pole double throw NMOS switches to connect digital inputs to the ladder.

→ Supply range +5V to +15V.
May can be between −10V to +10V.

→ max. Analog o/p voltage = +10V when all digital inputs are at logical 1 state.

→ A Zener diode is connected between OUT1, OUT2 to save DAC from negative transients

→ An Op Amp is need as a current to voltage converter at the o/p of AD 7523 to convert current o/p of AD 7523 to o/p voltage.

Scanned with CamScanner

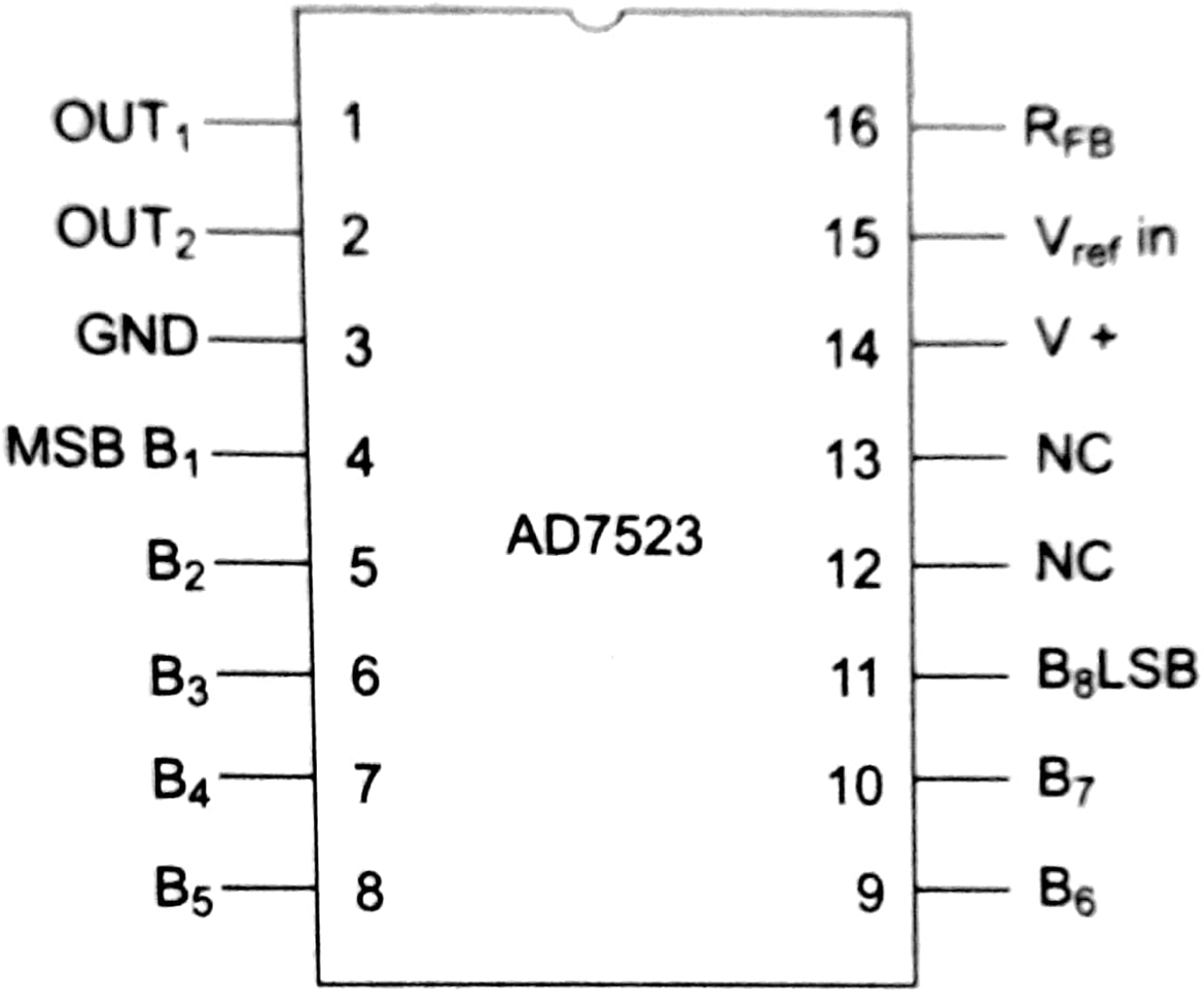


Fig. 5.45 *Pin Diagram of AD7523*

Scanned with CamScanner

Ex:-

Interface a DAC AD7523 with an 8086 CPU running at 8MHz and write an ALP to generate a saw-tooth waveform of period 1ms with Vmax 5V.

Sol

```
ALP :

        ASSUME  CS : CODE

CODE    SEGMENT.

START:  MOV AL, 80H ; port A as o/p port
        OUT CWR, AL ;

AGAIN:  MOV AL, 00H ; start the ramp from 0V

BACK:   OUT PORTA, AL ; o/p 00H to DAC
        INC AL
        CMP AL, 0F2H ; is upper limit reached?
        JB BACK ; if not, increment the ramp
        JMP AGAIN ; else start again from 00H

CODE    ENDS

        END START.
```

Scanned with CamScanner

→ port A is initialized as output port for sending digital data as input to DAC.

→ The ramp starts from 0V (analog), hence AL starts at 00H.

→ To increment the ramp, the content of AL is incremented during execution of the loop till it reaches F2H.

→ After that saw tooth again starts from 00H ie. 0V (analog) and procedure is repeated.

→ The ramp period given by this program is precisely 1.000625 ms.

count F2H is calculated by dividing the required delay of 1ms by the time required for execution of the loop once.

Scanned with CamScanner

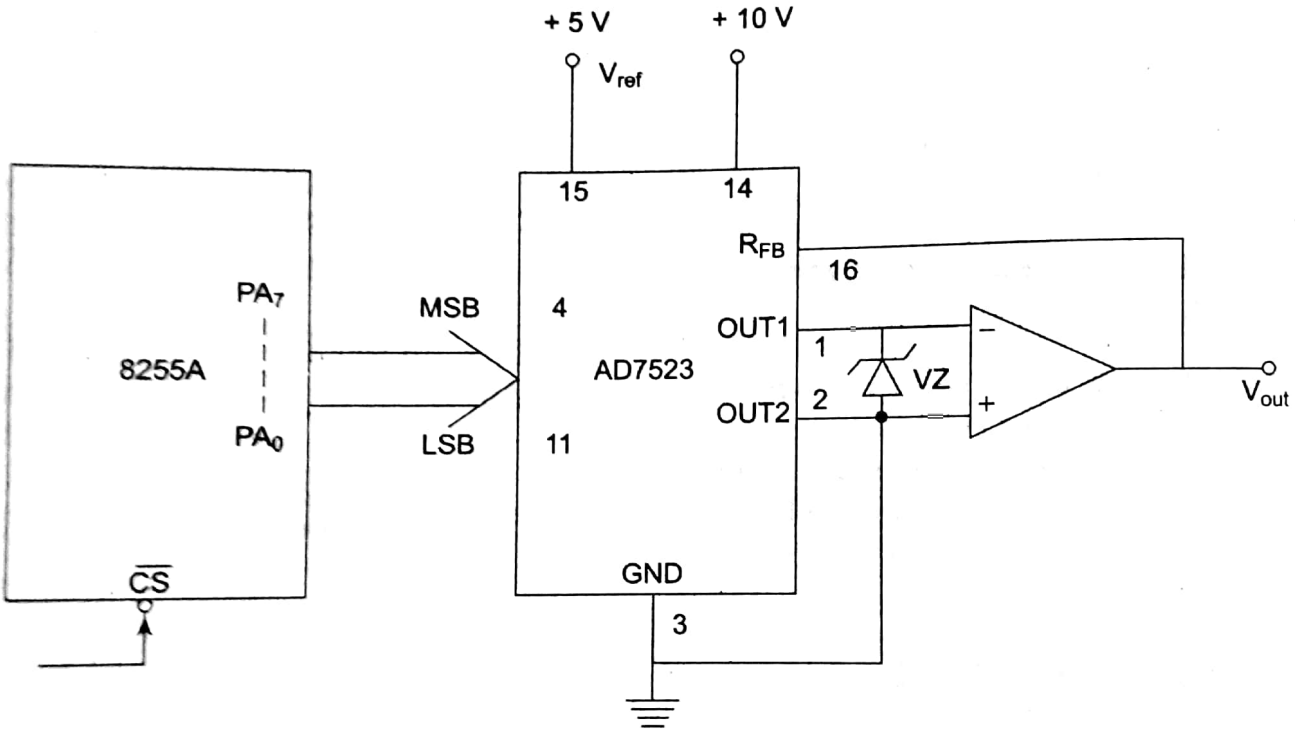


Fig. 5.46 *Interfacing of AD7523*

```
CODE        SEGMENT
START:      MOV AL,80 H          ; Initialise port A as output
            OUT CWR,AL           ; port
AGAIN:      MOV AL,00H           ; Start the ramp from 0V
BACK :      OUT PORTA,AL         ; Input 00H to DAC
            INC AL               ; Increment AL to increase ramp output
            CMP AL,0F2H          ; Is upper limit reached?
            JB BACK              ; If not, then increment the ramp
            JMP AGAIN            ; Else start again from 00H
CODE        ENDS
            END START
```

Program 5.13 *ALP for Generating Sawtooth Waveform Using AD 7523*

Scanned with CamScanner

DAC 0800 8 bit Digital to Analog converter

→ The DAC 0800 is a monolithic 8 bit DAC manufactured by National Semiconductor.

→ It has settling time around 100nS and can operate on a range of power supply voltages i.e. from 4.5V to 18V.

→ Usually supply voltage V+ is 5V or +12V.

The V- pin can be kept at a minimum of -12V.

Scanned with CamScanner

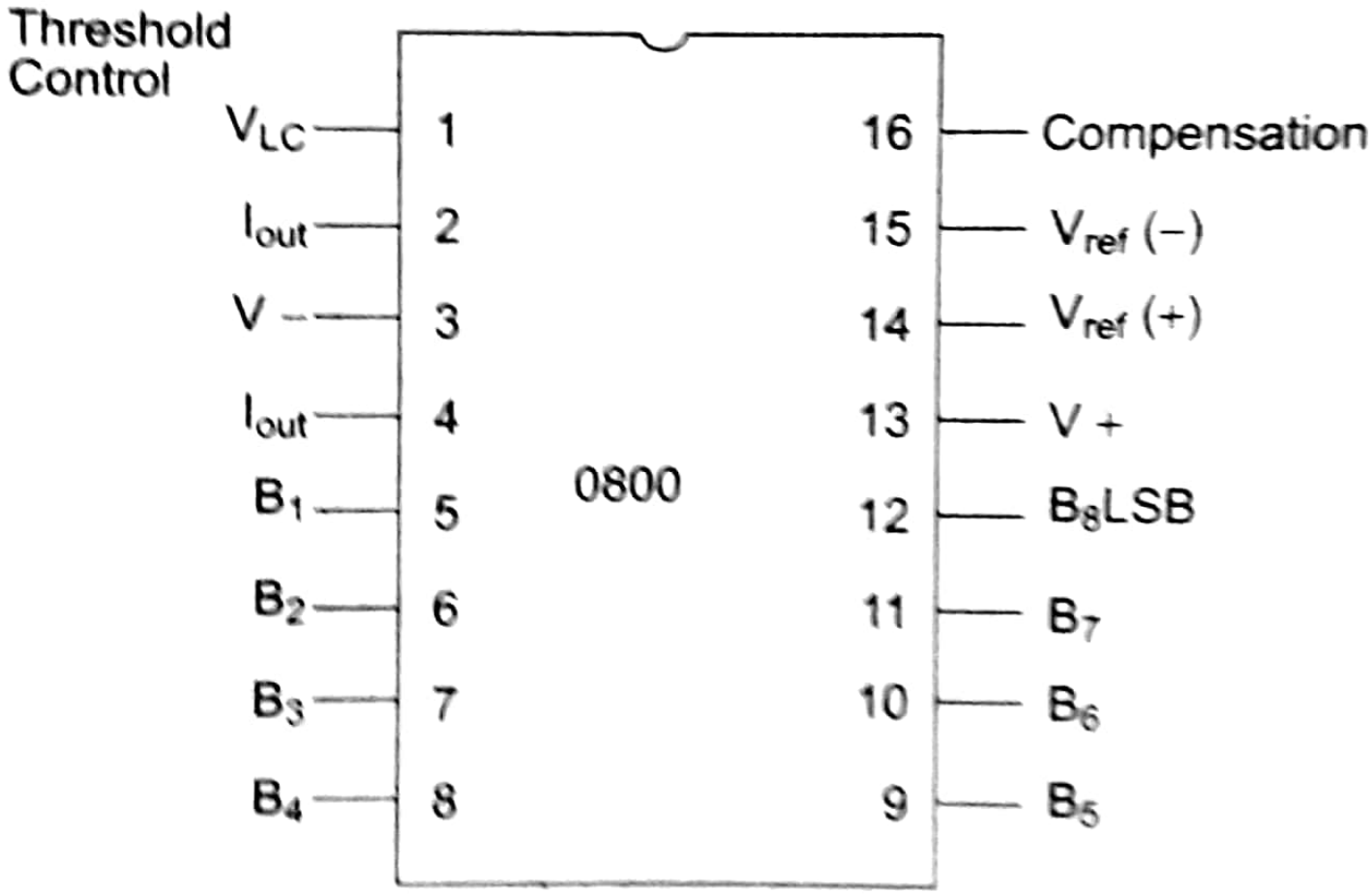


Fig. 5.47 *Pin Diagram of DAC 0800*

Scanned with CamScanner

Q: Write an ALP to generate a triangular wave of freq 500Hz using the interfacing circuit below. The ADC operates at 8MHz. The amplitude of triangular wave should be +5V.

Sol

→ The Vref+ should be tied to +5V to generate a wave of +5V amplitude.

→ The required frequency of output is 500Hz i.e. period is 2ms.

Assuming the wave generated is symmetric, the waveform will rise for 1ms and fall for 1ms.

This is repeated continuously.

Scanned with CamScanner

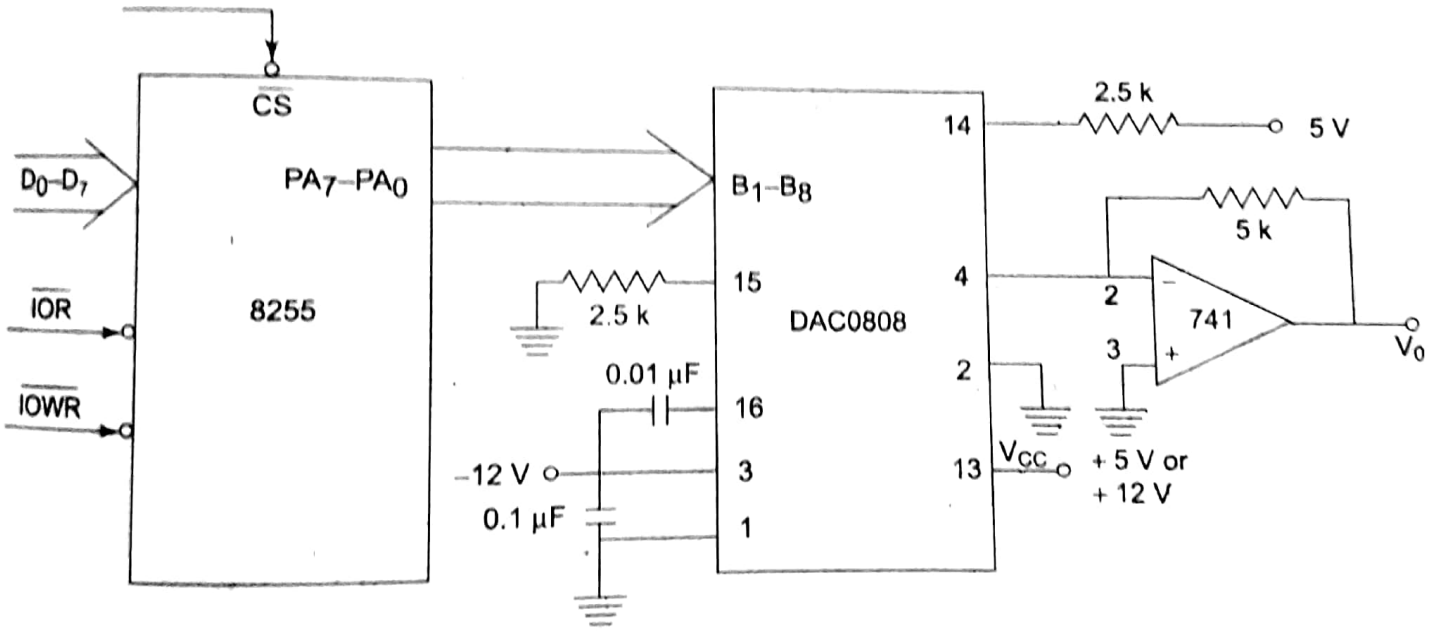


Fig. 5.48 *Interfacing DAC0800 with 8086*

Scanned with CamScanner

ALP:

```
ASSUME    CS: CODE
CODE      SEGMENT
START:    MOV. AL, 80H  ; Init. 8255 ports
          OUT CWR, AL   ;
          MOV AL, 00H   ; start rising ramp from
                        0V by sending 00H to
                        DAC
BACK:     OUT PORTA, AL ;
          INC AL
          CMP AL, FFH   ; If it is FFH then
BACK1:    OUT PORTA, AL ; output it and
                        start the falling ramp
          DEC AL        ; by decrementing AL
          CMP AL, 00    ; till it reaches 0.
          JA BACK1
          Jmp BACK:     ; Then start again
                        for next cycle
CODE      ENDS
          END START.
```

Scanned with CamScanner