Digital to Analogue Conversion

Very often a need arises of Converting digital output to Continuous Signal (Analogue) for Anlogue equipment, Signal generators etc.

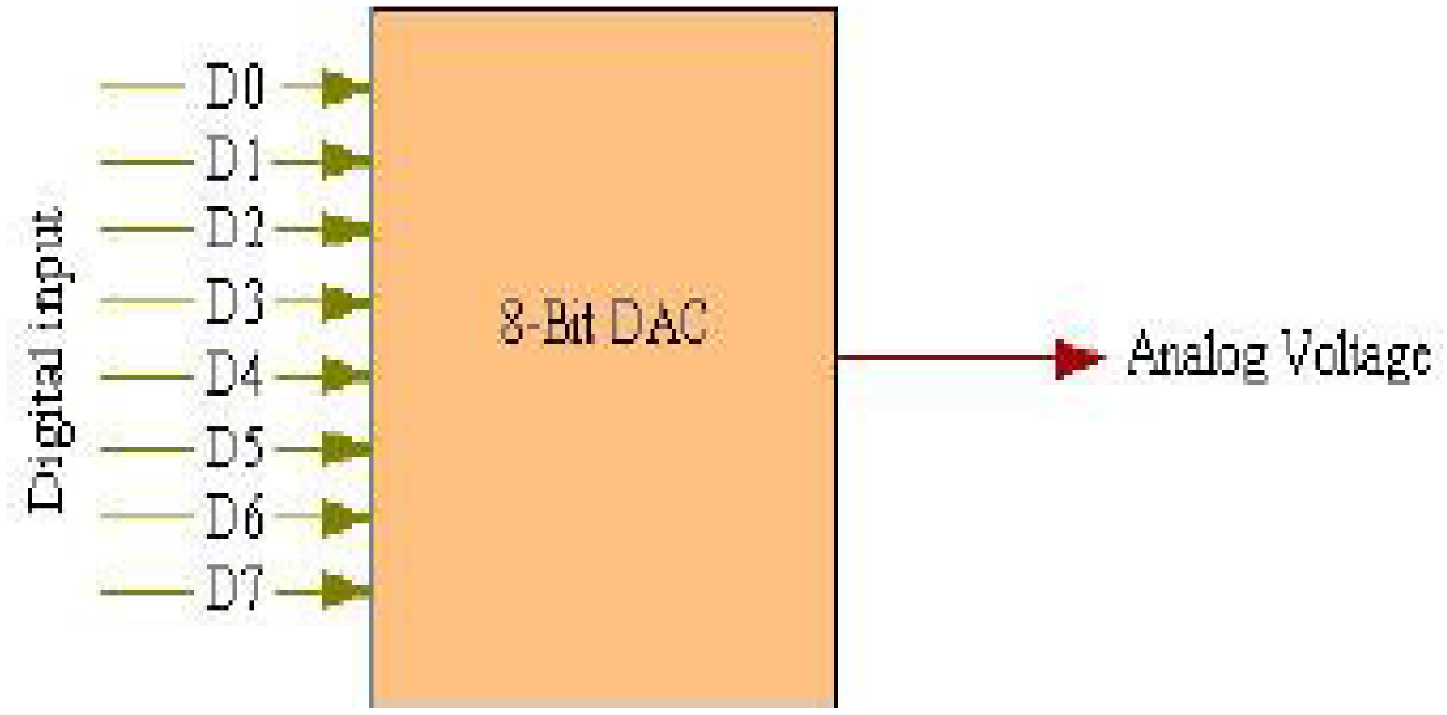
A DAC (Digital to Analog Converter) is used for this purpose. Various types of DAC exist i.e. 8-bit DAC, 10-bit DAC, 12-bit DAC, 16-bit DAC etc.

Digital to Analogue Conversion..

Basically a DAC is equipped with digital input lines for binary info, an LDAC (Load DAC) line, and an Analogue Output line.

These DACs can be interfaced to a microcomputer System through parallel output ports.

Digital to Analogue Conversion..



Digital to Analogue Conversion..

The DAC Resolution can be defined as change in output voltage resulting from a change of 1 LSB at the digital inputs.

For an n-bit DAC with a Full scale output Voltage, VFS, the resolution is given as:

$VFS/(2^n – 1)$

Digital to Analogue Conversion..

Generally Digital Input Value of 0 (all bits at logic 0) generates the minimum output voltage, while the maximum output voltage is generated when all input bits are at logic 1.

The output voltage at any time is:

$Vout = Vmin + (DV * VFS/(2^n – 1))$

Digital to Analogue Conversion..

Where Vmin = the lowest output voltage value, and DV = Digital input value.

For a DAC biased for a 0 to + 5v Voltage range:

$V_{out} = 0 + (DV * 5 /((2^n – 1))$

Digital to Analogue Conversion..

On the other hand for a DAC biased for the -5V to + 5V voltage range, VFS = 10V, as such:

$$Vout = -5 + (DV * 10 /((2^n – 1))$$

For an 8 bit DAC with 00H as input DV

$$Vout = - 5$$

For 0FFH inptut value:

$$Vout = - 5 + (255 * 10/255) = 5V$$

Digital to Analogue Conversion..

The algorithm for sending digital data to DAC is as follows:

1.Send digital data to the output port(s) where DAC input lines are connected.

2.Send a 1 to the LDAC bit

3.Send 0 to the LDAC bit

Steps 2 and 3 generate an LDAC pulse that commands the DAC to start Conversion. The pulse is like a Data Ready signal to DAC

Analogue to Digital Conversion

Provides a way of importing analogue data into a Computer System. The analogue data can come form sensors, transducers etc.

Analogue to Digital Converters (ADC) are used for this purpose.

Analogue to Digital Conversion

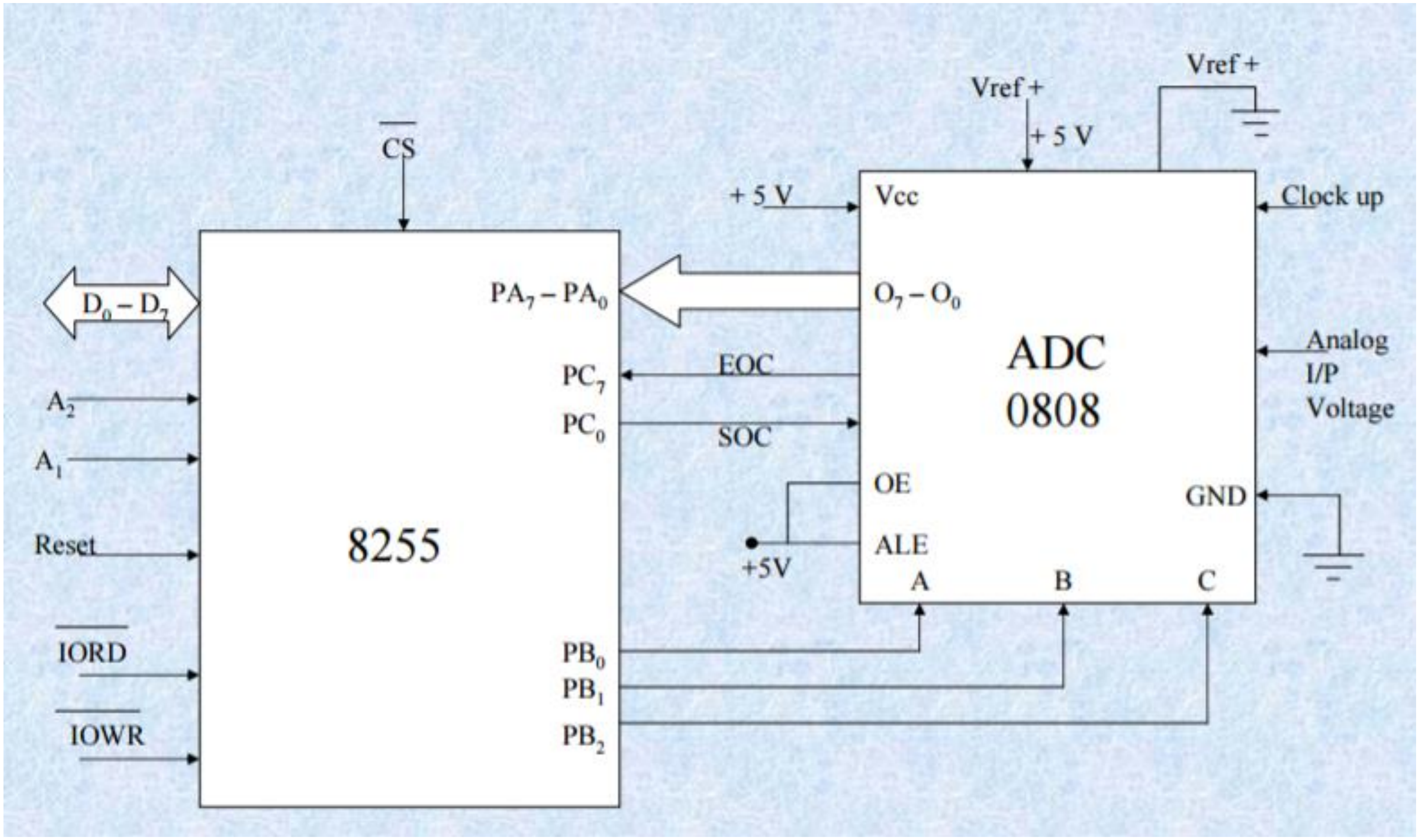
Typically an ADC is equipped with an Analogue input line, a STRT input line (Start of Conversion line SOC), a Busy output line (End of Conversion line, EOC) and Digital Output Lines.

The Digital Output lines, and the Busy line are connected to the Input ports of a computer system, and the STRT signal is provided by the computer through an output port line.

Analogue to Digital Conversion..

Generally the ADC generates digital value 0 for the Minimum input voltage, and has all Output bits at logic 1 for the Maximum input Voltage.

Analogue to Digital Conversion..

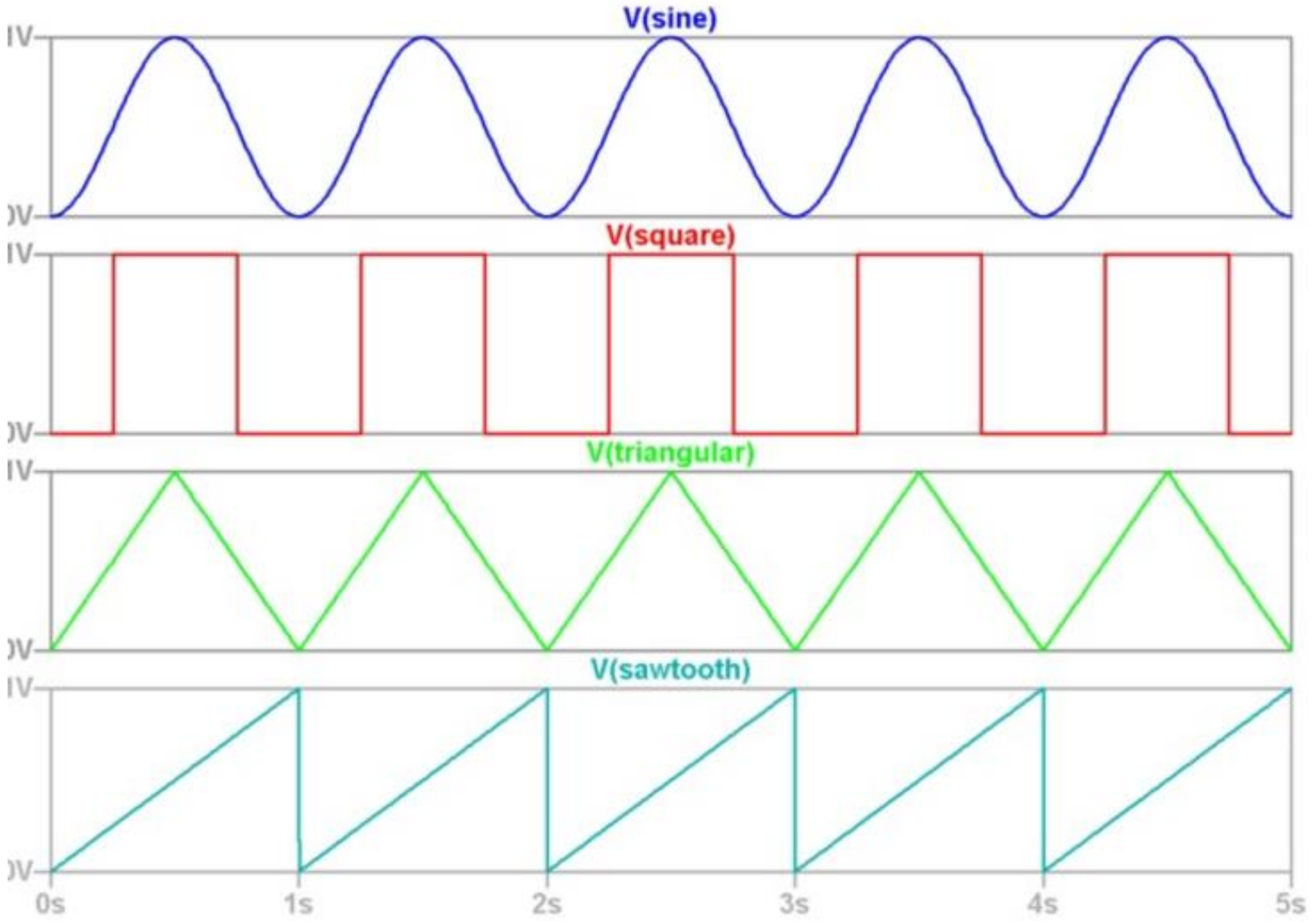


Analogue to Digital Conversion..

The algorithm for getting data from ADC is as follows:

1. Send logic 1 to the STRT (SOC)
2. Send logic 0 to the STRT (SOC) to complete the STRT pulse.
3. Monitor the BUSY line until Not Busy (EOC) is detected. i.e. the Conversion takes time.
4. Read Data from the input port.

Waveforms that can be Generated via DAC



Square Wave Generation via DAC

Example:

An Intel 8085 based microcomputer uses the 8155 PPI for its I/O Operations. An 8-bit DAC biased for 0 – 5V output voltage range gets its digital input from Port A of the PPI, while its LDAC signal is provided by bit 2 of the PPI Port B. The PPI occupies a memory block that starts at address 2800H. Write an assembly language program that continuously generates a 2 Hz square waveform at the DAC output with a peak to peak voltage of 0 to 5V.. The CPU operates at 2Mhz and Stack area is from 8250H to 8300H.

Square Wave Generation via DAC…

Step 1: Determine PPI Port Addresses

From the starting memory block address the 8155 PPI physical Port Addresses can be deduced as:

CSR	: 28H
Port A	: 29H
Port B	: 2AH
Port C	: 2BH

Square Wave Generation via DAC..

Step 2: Determine the Waveform Period

From the waveform frequency of 2Hz the time for 1 period is :

1/2Hz = 0.5 second

With a Square waveform half the period will be at 0V and the other half at 5V.

Square Wave Generation via DAC..

Step 3: Develop the Algorithm (or Flow Chart)

1. Define both Ports A and B as Output Ports
2. Output digital value for 0V output
3. Assert the LDAC Signal
4. Delay for half the period
5. Output digital value for 5V output
6. Assert the LDAC Signal
7. Delay for half the period
8. Repeat from Step 2

Square Wave Generation via DAC..

Step 4: Determine the DELAY Reference count

A ROM DELAY Routine has not been provided. As such we have to develop our own DELAY routine. This involves determining the number of iterations the count-down loop has to be executed in order to get the desired delay of 0.25 second.

Standard DELAY Routine

Label	Mnemonic		Number of States
DELAY:	LXI	D, xxxx	10
CTDOWN:	DCX	D	6
	MOV	A,D	4
	ORA	E	4
	JNZ	CTDOWN	10/7
	RET		10

Standard DELAY Routine…

Total number of States in the Routine:

10 + (6 + 4 + 4 + 10)*N – 3 + 10

= **24**N **+ 17**

Where N = Register pair content (iterations)

1 State → 1/f

24N + 17 States → DT

Standard DELAY Routine…

→ **N = (DT*F – 17) / 24**

Where :	DT	: Desired Delay Time
	F	: Operating CPU Frequency

In this case DT = 0.25 second and F = 2 MHz

➔ N = (0.25 * 2 * 10^6 – 17) / 24

➔ N = (500000 – 17) / 24 = 20832

= **5160H** (Register Pair D content)

Program Header & Port Defn Section

```
NAME      SQWAVE       ;Program Name
CSR       EQU  28H     :Command Status Register
                       ;Address
DACPT     EQU 29H      ;PPI Port A Address
LDACPT    EQU 2AH      ;PPI Port B Address


LXI       SP,8300H     ;Initialise Stack Pointer
MVI       A,03H        ;Bit Pattern to
OUT       CSR          ;Define Ports A and B as
                       ;Output ports
```

Generating Low level of Square Wave Section

```
LOW:    MVI     A, 00H          ;Digital value for DAC 0V output

        OUT     DACPT           ;Generate Low Level of
                                ;Square Wave
        MVI     A,04H           ;Prepare LDAC Port bit 2 = 1

        OUT     LDACPT          ;Assert the LDAC Pulse

        MVI     A,00H           ;Prepare LDAC Port bit 2 = 0

        OUT     LDACPT          ;Reset the LDAC Pulse

        CALL    DELAY           ;Delay for 0.25 second
```

Generating High level of Square Wave Section

```
MVI    A, 0FFH   ;Digital value for DAC 5V output

OUT    DACPT     ;Generate High Level of Square Wave
MVI    A,04H     ;Prepare LDAC Port bit 2 = 1
OUT    LDACPT    ;Assert the LDAC Pulse
MVI    A,00H     ;Prepare LDAC Port bit 2 = 0
OUT    LDACPT    ;Reset the LDAC Pulse
CALL   DELAY     ; Delay for 0.25 second


JMP    LOW       ;Restart with low level of waveform
```

User Defined DELAY Routine Section

```
DELAY:    LXI   D, 5160H   ;Reference Count for
                           ;0.25 second delay
CTDOWN:   DCX   D          ;Count down

          MOV   A,D

          ORA   E          ;Is 0.25 seconds over?

          JNZ   CTDOWN     ;No – Keep marking
                           ;time
          RET              ;Yes - Return to main
                           ; program
```

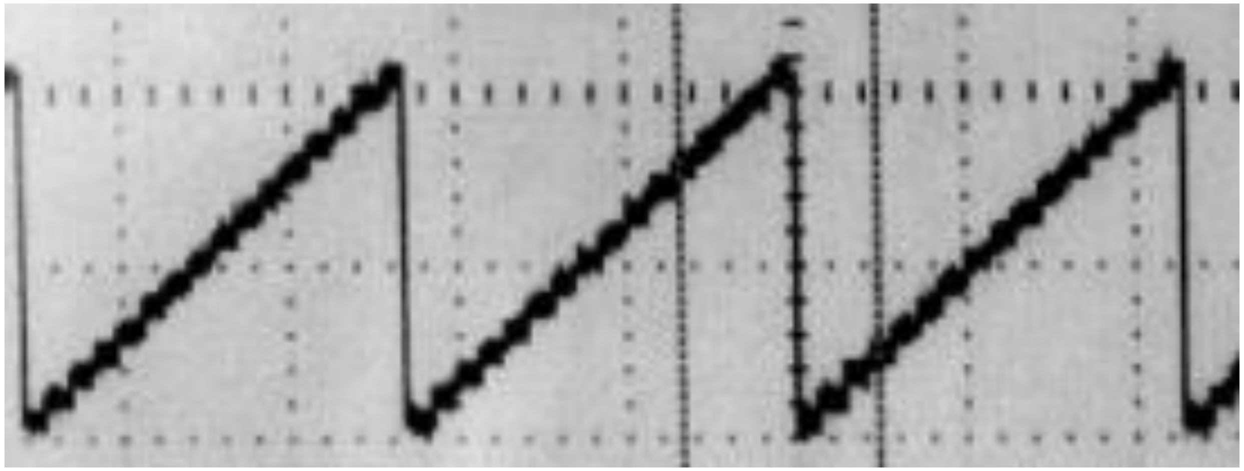
Sawtooth Waveform Generation

Example:

An Intel 8085 based microcomputer utilises the 8355 ROM chip I/O ports for its I/O Operations. An 8-bit DAC biased for 0 – 5V output voltage range gets its digital input from Port A of the ROM chip, while its LDAC signal is provided by bit 0 of the ROM chip Port B. The ROM chip occupies a memory block that starts at address 0000H. Write an assembly language program that continuously generates a 5 Hz Sawtooth waveform at the DAC output with a peak to peak voltage of 0 to 5V.. The CPU operates at 2Mhz and Stack area is from 8250H to 8300H.

Sawtooth Waveform Generation...

Can be generated using small incremental steps in the Digital Output values to DAC



Sawtooth Waveform Generation...

Step 1: Determine the Period

With a Frequency of 5 Hz One period is:

= 1 / f

= 1/5Hz

= 0.2 second

Sawtooth Waveform Generation...

Step 2: Choose Number of Samples

The 8-bit DAC does not provide the best of resolutions. As such we can use incremental steps of 1 for the digital output value.

Number of samples in one period is 256

Note: However for say a 12-bit DAC one can afford to go in incremental steps of 2 or even 4.

Sawtooth Waveform Generation...

Step 3: Determine the inter-sample interval

In one Period there are:

256 – 1 intervals

= 255

The inter-sample interval duration (delay):

T / 255 = (0.2 / 255) second

In other words after sending one digital value we delay for this duration before sending the next digital value.

Sawtooth Waveform Generation...

Step 4: Determine the Reference Count for inter-sample DELAY routine.

From the analysis we did on the Square waveform:

$$\mathbf{N = (DT*F - 17) / 24}$$

$$N = ((0.2 / 255) * 2 * 10^6 - 17) / 24$$

$$N = 64.651 = 64$$

N = 40H   Count cannot have fractions

When N has decimal points it must be truncated to take into account the CALL instruction time.

Sawtooth Waveform Generation...

Step 5: Determine the ROM Port Addresses

From the starting memory block address (0000H) the 8355 ROM chip occupies its physical Port Addresses can be deduced as:

Port A	: 00H
Port B	: 01H
DDRA	: 02H
DDRB	: 03H

Sawtooth Waveform Generation...

Step 6: Develop the Algorithm (or Flow Chart)

1. Define both Ports A and B bits as Output bits
2. Initialise digital value for 0V output
3. Output the Current digital Value
4. Assert the LDAC Signal
5. Delay for one inter-sample interval
6. Determine the next Digital value
7. Repeat from Step 3

Note: No need to initialise the total number of samples in this case. An increment on digital value 0FFH brings the value back to 00H.

Program Header & Port Defn Section

```
Label   Mnemonic                      Comment
        NAME       SAWTTH             ;Program Name
        DACPT      EQU  00H           ;ROM chip Port A Address
        LDACPT     EQU  01H           ;ROM Chip Port B Address
        DDRA       EQU  02H           :Port A Data Direction Register
                                      ;Address
        DDRB       EQU  03H           :Port B Data Direction Register
                                      ;Address

        LXI        SP,8300H           ;Initialise Stack Pointer
        MVI        A,0FFH             ;Bit Pattern to
        OUT        DDRA               ;Define All Port A Bits as Output
        OUT        DDRB               ;Define All Port B Bits as Output
```

Generating Waveform Section

```
        MVI     B, 00H      ;Initialise 0V output Digital Value
SEND:   MOV     A, B        ;Get current step Digital Value
        OUT     DACPT       ;Send Current Step digital value
                            ;to DAC
        MVI     A,01H       ;prepare LDAC Port bit 0 = 1
        OUT     LDACPT      ;Assert the LDAC Pulse
        MVI     A,00H       ; Prepare LDAC Port bit 0= 0
        OUT     LDACPT      ;Reset the LDAC Pulse
        CALL    DELAY       ; Delay for 1 inter-sample interval
        INR     B           ;Get Next Digital value
        JMP     SEND        ;Process new current digital value
```

Inter-Sample DELAY Section

```
DELAY:    LXI   D, 0040H   ;Reference Count for
                           ;inter-sample delay
CTDOWN:   DCX   D          ;Count down

          MOV   A,D

          ORA   E          ;Is inter-sample duration
                           ;over?
          JNZ   CTDOWN     ;No – Keep marking
                           time
          RET              YES - ;Return to main
                           ;program
```

Exercise

1. Modify the Sawtooth Example program such that it is the Triangular waveform that is generated with the same frequency and peak to peak amplitude.

2. The DAC for the Sawtooth program is now biased for the analogue output voltage range of -5V to + 5V. Modify the sawtooth program such that the generated sawtooth waveform has the same frequency but with a peak to peak amplitude of -2.5V to +2.5V.

Non-Linear Waveforms Generation

Non-linear waveforms such as the Sinusoidal Waveform can not be generated by just varying the digital value linearly in a loop as is the case for sawtooth and triangular waveforms.

To solve this problem lookup tables are employed.

Sinusoidal Waveform Generation

A 12-bit DAC biased for -5v to +5V is to be used to generate a sinusoidal waveform.

The digital values for generating a sinusoidal waveform with a peak to peak amplitude of –1V to +1V can be determined for the quarter of the waveform i.e. 0^o to 90^o.

Sinusoidal Waveform Generation..

For Amplitude of 1, and angle x:

Y = Sin(x)   Analogue output

Recall DAC output Vout (Y)

$Vout = Vmin + VFS * DV/(2^n - 1) = Y$

$\rightarrow DV = (Y - Vmin) * (2^n - 1) / VFS$

$DV = (Sin(x) - (-5)) * 4095/10$

Sinusoidal Waveform Generation..

DV = (Sin(x) +5) * 409.5

The Digital valuesa can therefore be determined and noted from 0^o to 90^o in steps of a quarter degree i.e. for 0, 0.25. 0.5,0.72,1.0,….,90^o

These Values are then placed in a lookup table with value for each degree requiring 2 bytes.

Sinusoidal Waveform Generation..

To generate the 0 to 90 degree section of the waveform one only has to make a loop that reads the digital values from the lookup table and sending them to the DAC Ports at appropriate intervals for the desired frequency.

The same values can also be used to generate the other sections of the Sinusoidal waveform with a little manipulation.

Exercise

A 12-bit DAC biased for -5V to + 5V Voltage range is to be used to generate a 2Hz Sinusoidal Waveform of peak to peak amplitude of -2V to +2V. Port A of the 8255 PPI provides the least significant 8 bits of DAC digital inputs, Port B bits 3 to 0 provide the most 4 significant bits of the DAC digital inputs, while bit 0 of Port C provides the LDAC signal. The memory block that starts at address DIGVAL contains the digital values for 0 to 90 degree section of a sinusoidal waveform with a peak to peak amplitude of -1V to +1V at steps of 0.25 degree. Each degree has two bytes in the lookup table, 1st byte hold the lower byte of the digital value and the second byte holds the Upper byte value. Write a program that continuously generates the described waveform using the digital values in the lookup table. CPU operates at 2Mhz and Stack area is from 8250H to 8300H. The PPI occupies a memory block that starts at 4000H