Experiment

4

Analog to Digital & Digital to Analog Conversion

Objective

The aim of this lab experiment is to study the Analog to Digital conversion and Digital to Analog conversion.

Equipment

Flight 8086 training board, Application board, PC with Flight86 software, download cable.

Tasks to be Performed

- Simulation of a A/D conversion employing successive approximation method using D/A converter
- Use a D/A converter to perform the following:
    1. Sine wave generation (using look up table)
    2. Staircase waveform generation
    3. Saw-tooth waveform generation
- Read the DIL switches and output the digital values to the LEDs and DAC. The analog output of the DAC is to be represented by lighting up the bar-graph.

4.1 Background

In any computer controlled process it may be required to monitor analog values, i.e. the output of a voltmeter or strain gauge. In order to do this, the analog signal must be first converted into a digital value using an Analog-to-Digital Converter (A/D). On the other hand, Digital-to-Analog Converters (D/A) can be used to convert a digital output from the computer into an analog value. For instance, we could generate a series of tones by changing the digital output values in such a way that the analog signal is represented as a sine wave.

4.2 A/D Conversion

The Application Board provides four sources of analog inputs which can be selected using a four position switch (SW3). The analog source can be provided externally (P2-in), or from the output of a light dependent resistor (LDR1), or from the temperature sensor (Q1), or from an on board variable voltage (UR6).

With the Application Board we can simulate a simple A/D converter that reads the output of a certain analog source (e.g. variable voltage) and converts it into a digital value as shown in Figure 4.1. By means sending values to Port-B, and hence by means of the D/A converter we can generate analog voltages proportional to the digital value we output. If this analog voltage is now compared with the unknown analog voltage, we can gradually adjust the output value to Port-A, until the comparator finds the two analog voltages equal. The digital value output to the D/A converter must be the digital value equivalent of the unknown analog input.

The output of the comparator is bit 3 of the input Port-B. A logic 1 means the output on Port-B is too small, a logic 0 means it is too large.

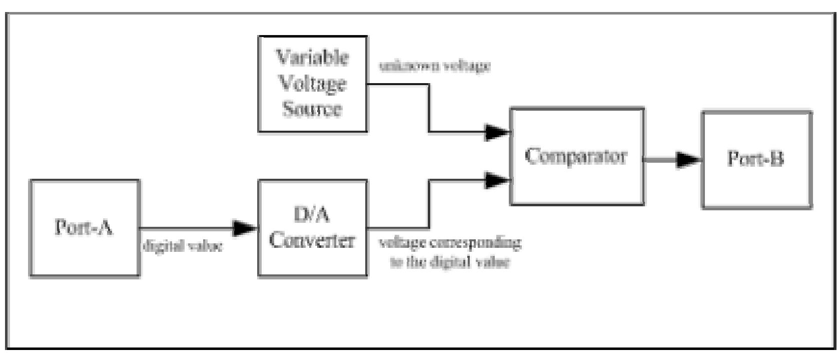

Figure 4.1: Simulation of an A/D Converter

Example 4.1: Write a program to simulate a simple A/D converter. Use the variable voltage source (UR6) as your analog source. The program should display one HEX digit (0-F) representing the digital value of the voltage input.

Set SW3 to VOLTS (Variable Voltage)
Set SW4-2 to DAC (enable D/A converter)
Set SW2-2, SW4-1, SW4-3, and SW4-4 OFF

```
COMSEG SEGMENT BYTE PUBLIC 'CODE'
ASSUME CS:COMSEG, DS:COMSEG, ES:COMSEG, SS:COMSEG
ORG 0100h
Start:
        MOV AL, 99h    ; initialize the 8255 PPI chip
        OUT 06h, AL    ; A input, B output, C input

        MOV AL,   0    ; turn off all LEDs
        OUT 02h, AL

ANAL:  MOV BL,    0    ; first digital value
SMALL: MOV AL,   BL    ; put in AL for output
       OUT 02h, AL     ; output value to D/A

       NOP             ; wait for D/A
       NOP
       NOP
       NOP
       NOP
       NOP

        IN   AL, 00h   ; get input Port-B
        AND AL,   08h  ; keep comparator bit (bit 3)

        JZ   LARGE     ; if value is large (bit3= 0)
                       ; then display the digital value
        INC BL         ; else increment the digital value
        JMP SMALL      ; and tray again

; display the digital value as a HEX digit
LARGE: MOV AL,    BL
        CALL display_voltage
        JMP  ANAL

INCLUDE display_voltage.asm
INCLUDE putc.asm
COMSEG ENDS
END    start
```

The previous code uses two functions, namely *display_voltage* and *putc*, to display the digital value corresponding to the input voltage. The first function converts the digital (binary) value of the input voltage into an ASCII character, and calls putc to display it on the screen. The two functions are given in two separate files *display_voltage.asm* and *putc.asm*, so that you can include these two files in your code using the INCLUDE directive.

4.3 D/A Conversion

The D/A converter on the Application Board produces an analog voltage proportional to the digital input. The digital input is provided by Port-A. The analog output corresponding to 00h is 0.00V, while the analog output corresponding to the digital input FFh (255) is approximately 2.55V.

The following example shows you how to generate a SIN wave using the D/A converter.

Example 4.2: Using a set of SIN tables for data, output a sine wave in 10 degree steps, observe the analog output (at P2-Out) with an oscilloscope and measure its frequency.

The first step is to construct a table containing the values of the SIN function for the following degrees: 0, 10, 20, …, 350 (see Table 4.1). Then, we assign a proper voltage to each value in the SIN table. As you know, the D/A converter can produce 256 different analog voltages (0.00V to 2.55V). Therefore, we can map a range of these voltages to the range of the SIN function (-1 to 1). Let us use the range 0.00V to 2.54V, such that 0.00V corresponds to -1 and 2.54V corresponds to 1. Since 1.27V is the mid point in the range 0.00V to 2.54V, it will be mapped to the mid point of the SIN range which is 0. Other voltage values can be mapped easily to the SIN values as shown in Table 4.1. Finally, we use the digital values corresponding to these analog voltages to generate the SIN wave as shown in the following program.

Set SW4-2 to DAC (enable D/A converter)
Set SW2-1 to SWITCH
Set SW4-1, SW4-3, and SW4-4 OFF

```
COMSEG SEGMENT BYTE PUBLIC 'CODE'
ASSUME CS:COMSEG, DS:COMSEG, ES:COMSEG, SS:COMSEG
ORG 0100h
Start:
        MOV AL, 99h  ; initialize the 8255 PPI chip
        OUT 06h, AL  ; A input, B output, C input
```

```
        MOV AL,  0 ; turn off all LEDs
        OUT 02h, AL

L1:     MOV SI,    OFFSET Table ; 1st element in the table
        MOV BL, 36 ; number of elements in the table

L2:     LODSB      ; load AL from the table
        OUT 02h, AL; and output the value to D/A

        DEC BL     ; count down table
        JZ  L2     ; loop if not zero
        JMP L1     ; if zero then return to the start
                   ; of the table

Table   DB 127     ; 1.27V -> sin(0)
        DB 149     ; 1.49V -> sin(10)
        DB 170     ; 1.70V -> sin(20)
        DB 191     ; 1.91V -> sin(30)
        DB 209     ; 2.09V -> sin(40)
        DB 224     ; 2.24V -> sin(50)
        DB 237     ; 2.37V -> sin(60)
        DB 246     ; 2.46V -> sin(70)
        DB 252     ; 2.52V -> sin(80)
        DB 254     ; 2.54V -> sin(90)
        DB 252     ; 1.27V -> sin(100)
        DB 246     ; 1.27V -> sin(110)
        DB 237     ; 1.27V -> sin(120)
        DB 224     ; 1.27V -> sin(130)
        DB 209     ; 1.27V -> sin(140)
        DB 191     ; 1.27V -> sin(150)
        DB 170     ; 1.27V -> sin(160)
        DB 149     ; 1.27V -> sin(170)
        DB 127     ; 1.27V -> sin(180)
        DB 105     ; 1.27V -> sin(190)
        DB 84      ; 1.27V -> sin(200)
        DB 64      ; 1.27V -> sin(210)
        DB 45      ; 1.27V -> sin(220)
        DB 30      ; 1.27V -> sin(230)
        DB 17      ; 1.27V -> sin(240)
        DB 8       ; 1.27V -> sin(250)
        DB 2       ; 1.27V -> sin(260)
        DB 0       ; 1.27V -> sin(270)
        DB 2       ; 1.27V -> sin(280)
        DB 8       ; 1.27V -> sin(290)
        DB 17      ; 1.27V -> sin(300)
        DB 30      ; 1.27V -> sin(310)
        DB 45      ; 1.27V -> sin(320)
        DB 64      ; 1.27V -> sin(330)
        DB 84      ; 1.27V -> sin(340)
        DB 105     ; 1.27V -> sin(350)
COMSEG ENDS
END     start
```

This program reads the digital values form Table and output them to Port-A, then the D/A converter converts them to the corresponding analog voltages. Notice that the values in Table can generate only one cycle of the SIN wave. Therefore, the digital values in Table are output continuously to the D/A converter to generate a continuous SIN wave.

Table 4.1: SIN Table

Degree	SIN(Degree)	Assigned Voltage	Corresponding Digital Value
0	0.000	1.27	127
10	0.174	1.49	149
20	0.342	1.70	170
30	0.500	1.91	191
40	0.643	2.09	209
50	0.766	2.24	224
60	0.866	2.37	237
70	0.940	2.46	246
80	0.985	2.52	252
90	1.000	2.54	254
100	0.985	2.52	252
110	0.940	2.46	246
120	0.866	2.37	237
130	0.766	2.24	224
140	0.643	2.09	209
150	0.500	1.91	191
160	0.342	1.70	170
170	0.174	1.49	149
180	0.000	1.27	127
190	-0.174	1.05	105
200	-0.342	0.84	84
210	-0.500	0.64	64
220	-0.643	0.45	45
230	-0.766	0.30	30
240	-0.866	0.17	17
250	-0.940	0.08	8
260	-0.985	0.02	2
270	-1.000	0.00	0
280	-0.985	0.02	2
290	-0.940	0.08	8
300	-0.866	0.17	17
310	-0.766	0.30	30
320	-0.643	0.45	45
330	-0.500	0.63	64
340	-0.342	0.84	84
350	-0.174	1.05	105

Exercises

6.1. Consider Example 4.1. Describe how you would minimize the number of digital values required to find the unknown voltage.

6.2. How could you vary the frequency of the SIN wave generated in Example4.2?

6.3. Use the D/A converter to perform the following:

a. Staircase waveform generation

b. Saw-tooth waveform generation

6.4. The bar-graph (U10) is essentially a bank of 10 LEDs. It is driven by U11 (LM3914) which samples a voltage input signal, as the signal exceeds certain preset levels it will output a signal to light one of the LEDs on the bar-graph. Hence, as the voltage increases, more LEDs will be turned on.

a. Write a program to read the DIL switches and output the digital values to the LEDs and DAC. The analog output of the DAC is to be represented by lighting up the bar-graph.

b. Plot a graph of switch digital value against the number of LEDs alight on the bar-graph.