Lecture 11

4.3 Examples of ADC and DAC interfacing:

4.3.1. DAC Interfacing

i. 8-bit DAC (1408) Interfacing Example

Problem Statement:

a. Design an output port with the address FFH to interface the 1408 DAC that is calibrated for 0-10V range.

Solution:

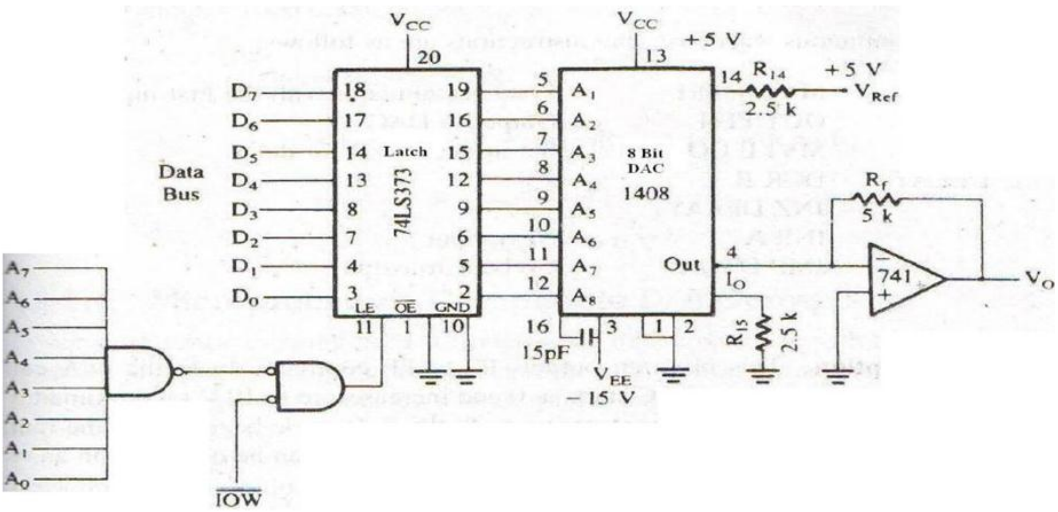


Figure 1: Interfacing 1408 DAC

Address lines (A7-A0) are decoded using the 8-input Nand gate and its output is combined with the control signal (IOW). Data bus (D7-D0) are connected to the latch (74LS373) and its output is connected to the input of 8- Bit DAC (1408).

b. **Write a program to generate a continuous ramp generator.**

Solution:

```
        MVI A,00H       ; Load accumulator with the first input
DTOA:   OUT FFH         ; Output to DAC (FF is port address)
        MVI B,COUNT     ; Setup register B for delay
DELAY:  DCR B
        JNZ DELAY
        INR A           ; Next input
        JMP DTOA        ; Go back to output
```

Program Description: This program outputs 00 to FF continuously to the DAC. The analog output of the DAC starts at 0 and increases up to 10 V (approx.) as a ramp. When the accumulator contents go to 0, the next cycle begins; thus the ramp signal is generated continuously. The ramp output of the DAC can be observed on an oscilloscope with an external sync [1].

The delay in the program is necessary for two reasons;

- The time needed for a microprocessor to execute an output loop is likely to be less than the settling time of ADC
- The slope of the ramp can be varied by changing the delay.

ii. 10-bit DAC (AD7522) Interfacing Example

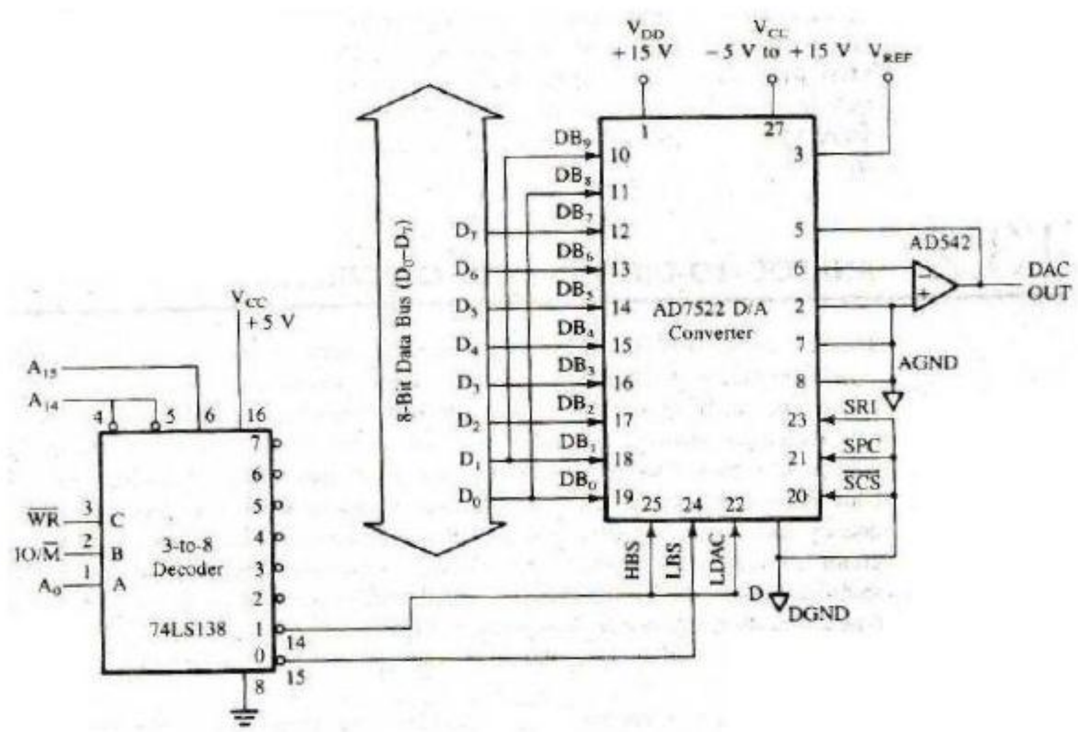


Figure 3: Interfacing 10 bit DAC with 8085 [1]

In many DAC applications, 10 or 12-bit resolution is required. But microprocessor has only 8-bit data lines. One method is to use two output ports on time shared basis; one for first eight bits and second for the remaining bits. Another method is to use double-buffered DAC such as AD7522, as shown below. AD7522 consist of an input buffer and a holding register. 10 bits are loaded into the input register in two steps using two o/p ports. The low-order 8-bits are loaded with the control line LBS and remaining 2-bits are loaded with the control line HBS. Then all 10-bits are switched into a holding register for conversion by enabling LDAC line [2].

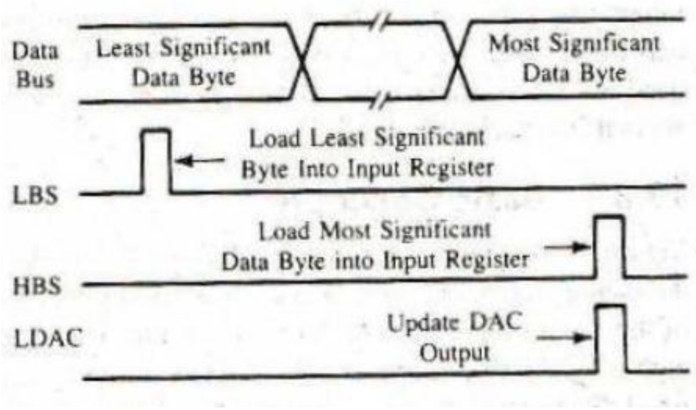


Figure 4: Timing Diagram [1]

For further explanation refer Gaonkar, page 413-414

4.3.2. ADC Interfacing

i. Interfacing an 8 bit ADC using status check

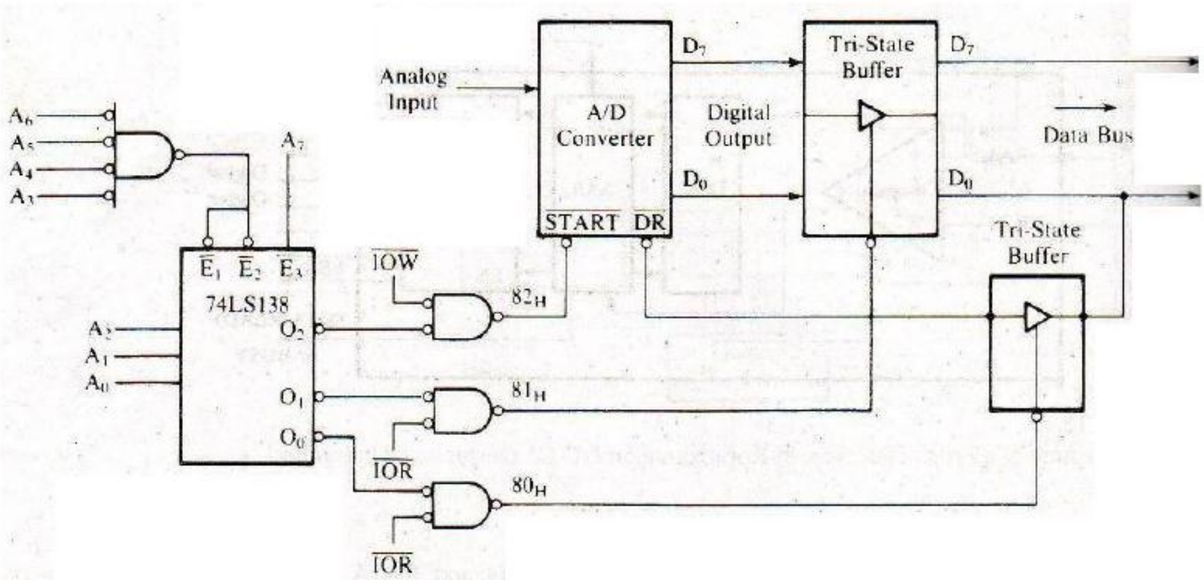


Figure 4: Interfacing an ADC using Status Check

ADC has one input line for analog signal and eight output lines for converted digital signal. Converter shows two lines START and DR (Data Ready). When an active low pulse is sent to the START pin, the DAC initiates conversion. When the conversion is complete, the DR goes low and data are made available on the output lines that can be read by the microprocessor.

To interface this converter, we need one output port to send a START pulse and another input port to check the status of DR line.

Subroutine:

```
        OUT 82H     ; start conversion

TEST:   IN 80H      ; Read data ready status
        RAR         ; Rotate D0 (LSB) into carry
        JC TEST     ; if D0=1, conversion is not
                      yet complete, go back and check

        IN 81 H     ; read output and save it in accumulator

        RET
```

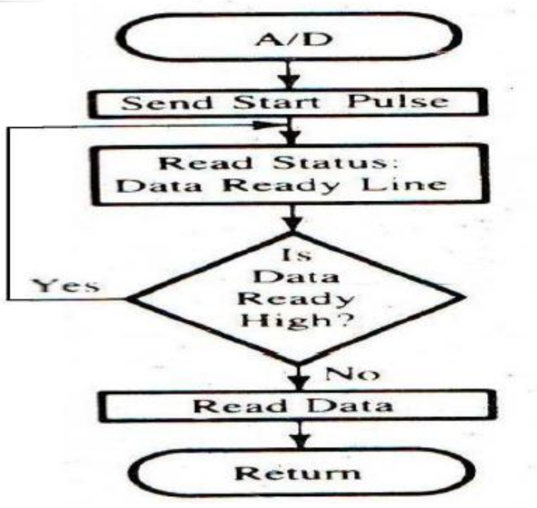


Figure 5: Flow chart of ADC using status checking

ii. Interfacing an 8 bit ADC using interrupt

Problem Statement:

a. Interface the ADC 0801 converter with 8085 MPU using memory mapped I/O and the interrupt RST 6.5

Solution:

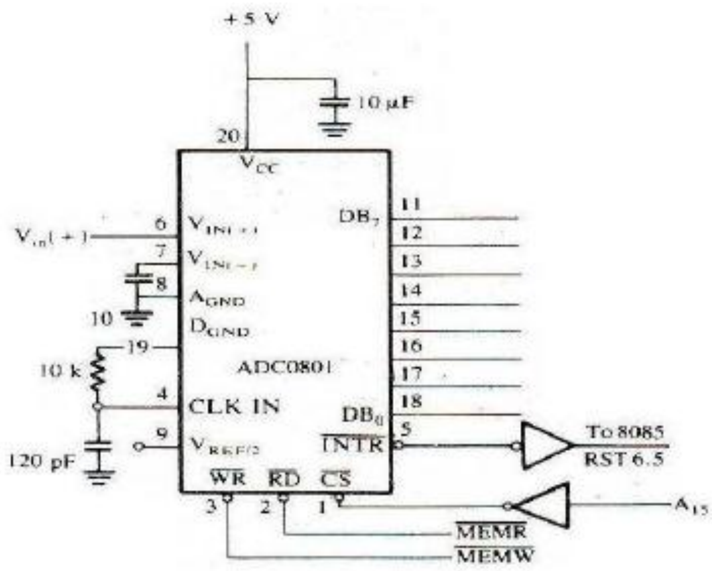


Figure 6: Interfacing ADC 0801 using Interrupt

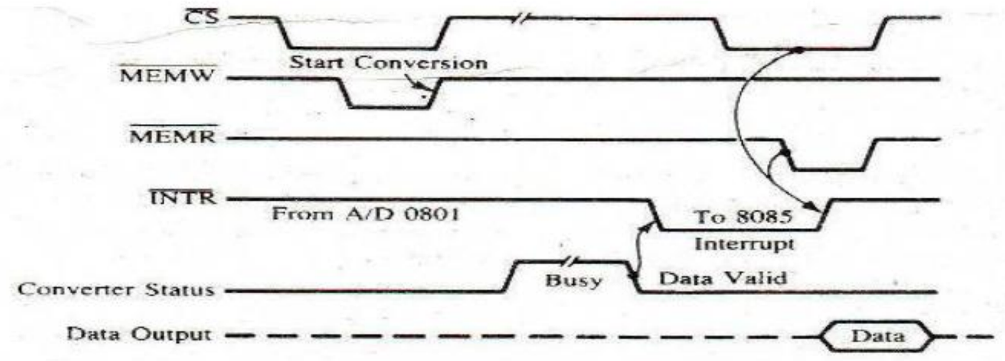


Figure 7: Timing diagram for Reading Data from ADC

In ADC interfacing using status check, we need external ports to access data and monitor the data ready signal. However in this configuration using Interrupt, the necessary logic is built inside the chip. It has three control signals: CS, WR and RD. To start conversion, CS and WR signals are asserted low. When the conversion is complete, the INTR is asserted low and the data are placed on the output lines. INTR signal can be used to interrupt the processor. When the processor reads the data by asserting RD, the INTR is reset.

b. Write an ISR to read the output data of the converter, store it in memory, and continue to collect data for the specified no. of times.

Solution:

Service Routine

LDA 8000H ; *Read data*, since A15 is high assuming all other lines at logic 0

MOV M, A ; *store data in memory*

INX H ; *Next memory location*

DCR B ; *Next count*

STA 8000H ; *start next conversion* by asserting MEMW signal

EI ; *Enable interrupt again*

RNZ ; *Go back to main if counter not equal to zero*

HLT ; *End*

References

[1] R. Gaonkar, Microprocessor Architecture, Programming, and Applications with the 8085, Mumbai: Penram International Publishing (India) Pvt.Ltd, 2010.

[2] H. Aryal, "Course Hands Out of Instrumentation II," Pulchowk Stationary, Pulchowk, Lalitpur, 2015.

Application of address decoding, 8085 interfacing with 8255, ADC & DAC in a single example:

Problem Statement:

Assume that your group has decided to make a PC based control system for a wine company. After studying the system, your group found out that the following to be implemented for controlling purpose:

- Pressure measurement (6 points)
- Temperature measurement (5 points)
- Weight measurement (1 point)
- Volume measurement for filling (5 points)

Your group also decided to use 8255A PPI card at base address **0550H**.

a) List out collected documents and components

Ans:

- Components: 8255A card, ADC, MUX, Memory, Processor, connecting wires, power supplies (+5V, GND), gates etc.
- Documents: Data sheets and technical documentation of above components

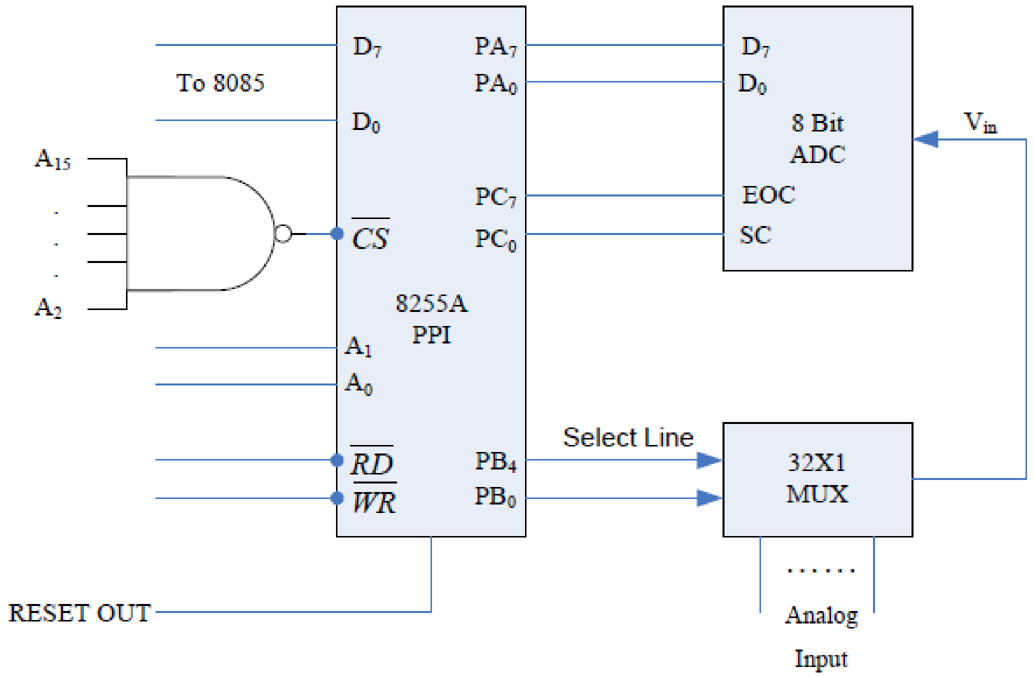
b) List out different signals you need to derive and or can be directly connected to your interfacing circuit.

Ans:

- A1, A2, Chip Select () for Port selection of of 8255A, RESET signal CS
- Read ( ) and Write () signals RD WR
- Start Conversion (SC) and End of Conversion (EOC)

c) Draw minimum mapping circuit for above system

Ans:



d) What are the address captured by card

Ans:

Since, The base address of card is 0550H, following are address captured by card :

Port	Address	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A	0550H	0	0	0	0	0	1	0	1	0	1	0	1	0	0	**0**	**0**
B	0551H	0	0	0	0	0	1	0	1	0	1	0	1	0	0	**0**	**1**
C	0552H	0	0	0	0	0	1	0	1	0	1	0	1	0	0	**1**	**0**
CR	0553H	0	0	0	0	0	1	0	1	0	1	0	1	0	0	**1**	**1**

The total numbers of monitoring points are 17. If we use 1 ADC for all of them, we need to select any one at given time. So, we can use 32X1 MUX which would then have 2^5=32 i.e. 5 selection lines (B_0 to B_4). These lines can have defined for any of the 17 lines.

In the above circuit,

Port A → Input port to read data from ADC in mode 0

Port B → Output port to select any one of 17 lines from MUX in mode 0

Port C → Output port (PC_0 as SC) and Input port (PC_7 as EOC)

e) Generate necessary control word

Ans:

Control word to set up port ports in above configuration:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	0	0

= 98H

BSR word to set PC_0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	1

= 01H

BSR word to reset PC_0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0

= 00H

Assuming that ADC starts the conversion process only when it receives SC signal and after conversion indicates via EOC line i.e. it has finished conversion and so ADC port data in its data lines which can be now be read through port A.

f) Write a program module for measuring the pressure of all the points

Ans:

```
        LXI H, MEMORY
        MVI A, 98H
        STA 0553H            ; write control word in CR
        MVI C, 06H           ; set counter to read 6 pressure points
        MVI B, 00H           ; selection of points for MUX
NEXT:   MOV A, B
        STA 0551H            ; select first pressure point
        MVI A, 01H           ; load A with BSR word to set PC0
        STA 0553H            ; set SC line
        CALL DELAY
        MVI A, 00H           ; load A with BSR word to reset PC0
        STA 0553H            ; reset SC line
READ:   LDA 0552H            ; read port C
        RAL
        INC READ             ; check for PC7
```

```
LDA 0550H           ; read data from port A
MOV M, A            ; store value in memory
CPI MAX_VALUE       ; compare with maximum value
JNC CONTROL         ; control value
CPI MIN_VALUE       ; compare with minimum value
JC CONTROL          ; control value
INC B
INX H
DCR C
JNZ NEXT
```

Problem Statement:
Interface a temperature sensor using an A/D converter and port A of the 8255. Interface a fan and a heater using optocouplers and triacs to drive the I/O devices. Write instructions to read the temperature; if the temperature is less than 10$_o$C, turn on the heater; and if the temperature is higher than 35$_o$C, turn on the fan.
Load temperature from temperature sensor LM135 and control fan and heater.
If temperature > 35$_o$ → Fan ON
If temperature < 10$_o$ → Heater ON

Solution:
(Refer Gaonkar 15.1.4 Illustration, pages 468-472)