Experiment No. 6: Analog-to-Digital (A/D) and Digital-to-Analog (D/A) Conversion Interfacing

---

1. Aim

To understand the principles and practical implementation of Analog-to-Digital (A/D) and Digital-to-Analog (D/A) conversion, and to interface these converters with a microprocessor (e.g., 8085/8086) to perform data conversion operations.

2. Objectives

Upon completion of this experiment, the student will be able to:

- **Explain the concepts of D/A and A/D conversion and their parameters (resolution, accuracy, conversion time, full-scale voltage).**
- **Understand the internal working principles of common DAC (e.g., DAC0808) and ADC (e.g., ADC0804) ICs.**
- **Design and implement interfacing schematics for DAC and ADC with a microprocessor.**
- **Write assembly language programs to generate analog outputs using a DAC.**
- **Write assembly language programs to read analog inputs using an ADC and display the digital values.**
- **Observe and analyze the generated analog waveforms and converted digital readings.**

3. Theory

The real world is predominantly analog, characterized by continuous signals (e.g., temperature, pressure, sound). Microprocessors, however, operate on digital signals (discrete binary values). Analog-to-Digital (A/D) converters and Digital-to-Analog (D/A) converters are essential interface components that bridge this gap, enabling microprocessors to interact with the analog world.

3.1. Digital-to-Analog (D/A) Conversion

A DAC converts a digital input code (binary) into a proportional analog output voltage or current.

- **Resolution: The smallest change in analog output for a 1-bit change in the digital input. For an N-bit DAC, the number of discrete steps is 2N.**
    - **Resolution (Voltage) = Full Scale Output Voltage / 2N**
- **Full Scale Output Voltage (V_FS): The maximum analog output voltage the DAC can produce.**
- **Reference Voltage (V_REF): An external stable voltage source used by the DAC to determine its output range.**
- **Output Voltage (V_OUT) Calculation:**

- **For a unipolar DAC (0 to V_FS):**
  V_OUT = Digital Input Value (decimal) * (V_FS / 2N)
  or
  V_OUT = Digital Input Value (decimal) * Resolution (Voltage)
- **Example: For an 8-bit DAC with V_REF = 5V (resulting in V_FS = 5V if correctly configured) and digital input 80H (128 decimal):**
  Resolution = 5V / 28 = 5V / 256 = 0.01953125 V/step
  V_OUT = 128 * (5V / 256) = 128 * 0.01953125V = 2.5V

DAC0808 (8-bit Digital-to-Analog Converter):

The DAC0808 is a popular 8-bit monolithic multiplying DAC.

- **Input Pins: D0-D7 (Digital Data Inputs), V_REF+ / V_REF- (Reference Voltage Inputs), Compensation (for internal op-amp stabilization).**
- **Output Pins: I_OUT (Current Output), I_OUT (Complementary Current Output). A resistor and an external op-amp (Current-to-Voltage converter) are typically used to convert the output current into a measurable voltage.**
- **Control Pins: Usually no separate control pins for simple write operations; it's always ready to convert. The analog output changes immediately when the digital input lines change.**

3.2. Analog-to-Digital (A/D) Conversion

An ADC converts an analog input voltage into a proportional digital output code.

- **Resolution: The smallest analog voltage change that causes a 1-bit change in the digital output.**
  - **Resolution (Voltage) = (V_MAX - V_MIN) / 2N (where V_MAX and V_MIN define the input range).**
- **Conversion Time: The time taken by the ADC to convert an analog input into a digital output.**
- **Full Scale Voltage (V_REF or V_MAX): The maximum analog input voltage the ADC can convert.**
- **Digital Output Value Calculation:**
  - **Digital Output (decimal) = Analog Input Voltage * (2N / V_REF)**
  - **Example: For an 8-bit ADC with V_REF = 5V and analog input 2.5V:**
    Resolution = 5V / 28 = 5V / 256 = 0.01953125 V/step
    Digital Output (decimal) = 2.5V / (5V / 256) = 2.5V / 0.01953125 = 128 (approx.)
    So, 80H.

ADC0804 (8-bit Analog-to-Digital Converter):

The ADC0804 is a commonly used 8-bit successive approximation ADC.

- **Input Pins: VIN+ / VIN- (Analog Voltage Input), V_REF/2 (Reference Voltage Input), CLK R / CLK IN (Clock Resistor / Clock Input for internal or external clock).**
- **Output Pins: D0-D7 (Digital Data Outputs).**
- **Control Pins:**
    1. **overlineCS (Chip Select): Active low, enables the chip.**
    2. **overlineRD (Read): Active low, enables output buffers to place data on data bus.**
    3. **overlineWR (Write): Active low, initiates a conversion.**
    4. **INTR (Interrupt): Active low output, goes low when conversion is complete.**
    5. **CLK R: For internal clock, a resistor and capacitor are connected to this pin.**
- **Conversion Process:**
    1. **Microprocessor sends a overlineCS low and overlineWR low pulse to start conversion.**
    2. **The ADC converts the analog input. During conversion, INTR is high.**
    3. **When conversion is complete, INTR goes low.**
    4. **Microprocessor detects INTR low, then sends overlineCS low and overlineRD low pulse to read the converted digital data from D0-D7.**

3.3. Interfacing with Microprocessor (8085/8086)

Interfacing ADCs and DACs with a microprocessor involves connecting their data lines to the microprocessor's data bus, control lines to I/O or control signals, and address decoding logic (for I/O mapped I/O).

- **I/O Mapped I/O: DAC and ADC chips are treated as I/O ports. They are accessed using `OUT` (for DAC) and `IN` (for ADC) instructions, and their Chip Select is generated using I/O address decoding, which typically involves the IO/overlineM signal being high.**

4. Materials Required

- **Microprocessor Trainer Kit (8085 or 8086 based)**
- **DAC0808 (8-bit Digital-to-Analog Converter) IC**
- **ADC0804 (8-bit Analog-to-Digital Converter) IC**
- **LM741 or similar Op-Amp IC (for DAC0808 current-to-voltage conversion)**
- **Resistors: 10k Ohm, 5k Ohm, 1k Ohm, 220 Ohm (for DAC/ADC specific configurations)**
- **Capacitors: 150 pF (for ADC0804 internal clock)**
- **Potentiometer (e.g., 10k Ohm - for variable analog input to ADC)**
- **LEDs (8 for digital output display)**
- **Oscilloscope (for observing DAC output waveforms)**
- **Digital Voltmeter (DVM) / Multimeter (for measuring analog voltages)**
- **Connecting Wires/Jumper Cables**
- **Breadboard (if not using a dedicated trainer kit interface)**
- **DC Power Supply (+5V, -5V, +12V as required by ICs)**

5. Procedure

Part A: DAC0808 Interfacing and Waveform Generation

1. **Interfacing Schematic for DAC0808:**
    - **Connect DAC0808 D0-D7 to the microprocessor's D0-D7 data bus.**
    - **Connect the V_REF+ pin to +5V (or desired reference voltage).**
    - **Connect the V_REF- pin to ground.**
    - **Connect the Power Supply pins: VCC to +5V, VEE to -5V (or ground if using single supply mode).**
    - **Connect I_OUT to an external current-to-voltage converter circuit using an op-amp (e.g., LM741) and a feedback resistor. A common configuration uses a 5k Ohm feedback resistor with V_REF = +5V.**
        - **Output Voltage (V_OUT_OPAMP) = -I_OUT * R_F (where I_OUT is the DAC current output and R_F is the feedback resistor).**
        - **I_OUT = I_REF * (Digital Input / 256) where I_REF = V_REF / R_REF (R_REF is often an internal resistor or an external resistor on V_REF pin). For DAC0808, I_REF = V_REF / 10k Ohm.**
        - **So, V_OUT_OPAMP = - (V_REF / 10k Ohm) * (Digital Input / 256) * R_F.**
        - **If V_REF = 5V and R_F = 5k Ohm, V_OUT_OPAMP = - (5V / 10k) * (Digital Input / 256) * 5k = - (0.5mA) * (Digital Input / 256) * 5k = - (2.5 * Digital Input / 256).**
        - **To get positive voltage, either invert the op-amp output or use I_OUT (complementary output) with an op-amp. Most trainers have internal op-amps for DAC.**
    - **I/O Address Decoding: Design a simple I/O address decoder to generate a Chip Select (overlineCS) for the DAC. For example, assign DAC to I/O address 40H.**
        - **Connect A6 of 8085/8086 to a gate, and combine with IO/overlineM (high) and overlineRD (high, as it's a write operation) to create overlineCS.**
        - **For 8085: Use A6 and IO/overlineM (inverted) through a NAND gate, and then connect this to the DAC's CS. For a fixed port, say 40H, use A6=1, A5=0, A4=0, A3=0, A2=0, A1=0, A0=0. A simple decode could be A6 ORed with inverters of lower address lines. For simplicity, many trainers connect a DAC to a fixed port, say 40H, without complex external decoding. We'll assume a port address.**
2. **Assembly Program for Ramp/Staircase Waveform (8085/8086 example):**
    - **Aim: Generate a linearly increasing analog voltage output (staircase waveform) by sending incremental digital values to the DAC.**
    - **Microprocessor: 8085 (example, similar logic for 8086)**
    - **Port Address: Assume DAC is interfaced at Port 40H.**
    - **Assembly Code:**
    - **Code snippet**

```
; 8085 Assembly Code for Ramp Waveform Generation
 ORG 0000H

 MVI A, 00H      ; Initialize Accumulator with 0
LOOP:
 OUT 40H         ; Output A to DAC (Port 40H)
                 ; DAC converts this digital value to an analog voltage
 INR A           ; Increment A
 JNZ LOOP        ; Repeat until A overflows (goes from FFH to 00H)

 HLT             ; Halt
```

- **Execution and Observation:**
  - **Connect the DAC's analog output to an oscilloscope.**
  - **Run the assembly program.**
  - **Observe the oscilloscope display. A staircase waveform should be visible, starting from 0V and increasing incrementally up to the full-scale voltage, then resetting to 0V.**
  - **Measure the step size and peak voltage to verify DAC operation.**

Part B: ADC0804 Interfacing and Digital Display

1. **Interfacing Schematic for ADC0804:**
   - **Connect ADC0804 D0-D7 to the microprocessor's D0-D7 data bus.**
   - **Connect VIN+ to the output of a potentiometer (variable voltage source, 0-5V). Connect VIN- to ground.**
   - **Connect V_REF/2 to +2.5V (for a 0-5V input range, V_REF = 5V. If V_REF is tied to VCC, it sets the max input). Typically, V_REF/2 is connected to VCC/2 or a dedicated 2.5V source. If V_REF is left open, it defaults to VCC.**
   - **Provide a clock source: For internal clock, connect 10k Ohm resistor between CLK R and CLK IN, and 150 pF capacitor between CLK IN and ground.**
   - **Connect Power Supply pins: VCC to +5V, GND to ground.**
   - **Control Signals:**
     - **overlineCS (Chip Select): Connect to I/O address decode output (e.g., Port 41H).**
     - **overlineRD (Read): Connect to microprocessor's overlineRD signal (for I/O read).**
     - **overlineWR (Write): Connect to microprocessor's overlineWR signal (for I/O write).**
     - **INTR (Interrupt): Connect to a status bit that the microprocessor can poll (e.g., a data line from an input port). For simple polling, connect INTR to D0 or D7 of an input port and read that port. Alternatively, use a ready pin if the trainer kit has one.**

- **Output Display: Connect D0-D7 of the ADC to 8 LEDs with current limiting resistors, or to an LCD module for display.**

2. **Assembly Program for ADC Reading and Display (8085/8086 example):**
   - **Aim: Read analog voltage from potentiometer, convert to digital, and display on LEDs/LCD.**
   - **Microprocessor: 8085 (example, similar logic for 8086)**
   - **Port Addresses: Assume ADC overlineWR is at Port 41H (dummy write to start conversion), and ADC overlineRD data is at Port 41H (actual read). Assume INTR is connected to bit D7 of Input Port 42H.**
   - **Assembly Code:**
   - **Code snippet**

```
; 8085 Assembly Code for ADC Read and Display
ORG 0000H

; Initialize (if necessary, for LCD or LED segment drivers)
; For direct LED connection, no special init needed

START_CONVERSION:
MVI A, 00H      ; Dummy data
OUT 41H         ; Send pulse to ADC WR to start conversion.
                ; This OUT instruction asserts WR (low) and selects port 41H
                ; (assuming 41H is decoded for ADC WR)

WAIT_FOR_CONVERSION:
IN 42H          ; Read status of INTR (e.g., from D7 of Port 42H)
ANI 80H         ; Mask all bits except D7 (INTR connected to D7)
JNZ WAIT_FOR_CONVERSION ; Loop until INTR goes low (D7 becomes 0)
                        ; If INTR is active low, it means D7 should be 0.
                        ; So, JNZ means "If D7 is NOT 0, keep waiting"

READ_ADC_DATA:
IN 41H          ; Read digital data from ADC (Port 41H)
                ; This IN instruction asserts RD (low) and selects port 41H
                ; (assuming 41H is decoded for ADC RD)
MOV B, A        ; Store the digital value in B register (for observation/display)

; Optional: Display B on LEDs or LCD
; If LEDs connected directly to data bus for output (e.g., through a latch)
; OUT LED_PORT  ; Replace LED_PORT with actual output port for LEDs

HLT             ; Halt

; Consider a loop to continuously read and display if desired.
; JMP START_CONVERSION ; Uncomment for continuous operation
```

- **Execution and Observation:**
  - **Vary the potentiometer connected to VIN+ of the ADC.**
  - **Observe the LEDs (or LCD). The binary pattern on the LEDs should change, reflecting the analog voltage from the potentiometer.**
  - **Measure the analog input voltage using a DVM and compare it with the displayed digital output.**
  - **Numerical Example:**
    - **If potentiometer is set to 2.5V, and ADC is 8-bit with 5V V_REF, the expected digital output is 80H (128 decimal).**
    - **If LEDs are connected to D0-D7, you should see D7 glowing, and all lower bits off.**
    - **If potentiometer is set to 1.25V, expected digital output is 40H (64 decimal).**

6. Observations

Record your observations during the execution of each part.

- **Part A: DAC0808 Interfacing**
  - **Generated Waveform: Describe the shape of the waveform observed on the oscilloscope (e.g., "staircase increasing linearly").**
  - **Peak Voltage: Note the maximum voltage reached by the staircase (e.g., "Approximately 4.9V for a +5V reference").**
  - **Step Size: Estimate the voltage change per step (e.g., "Around 19.5 mV per step").**
  - **Frequency/Period: Note the approximate period or frequency of the waveform.**
  - **Analysis: Confirm that the observed waveform characteristics align with the theoretical calculations for the DAC.**
- **Part B: ADC0804 Interfacing**
  - **Potentiometer Min Position:**
    - **Analog Input Voltage (DVM reading): [e.g., 0.0V]**
    - **Digital Output (LEDs/LCD): [e.g., 00H (00000000b)]**
  - **Potentiometer Mid Position:**
    - **Analog Input Voltage (DVM reading): [e.g., 2.5V]**
    - **Digital Output (LEDs/LCD): [e.g., 80H (10000000b)]**
  - **Potentiometer Max Position:**
    - **Analog Input Voltage (DVM reading): [e.g., 4.9V]**
    - **Digital Output (LEDs/LCD): [e.g., FFH (11111111b)]**
  - **Varying Input: Describe how the digital output changes as the potentiometer is smoothly rotated from min to max (e.g., "LEDs show increasing binary counts as voltage increases").**
  - **Analysis: Verify that the digital output values are proportional to the analog input voltages, confirming the A/D conversion process. Note any discrepancies due to resolution or accuracy limitations.**

7. Deliverables

1. **Explanation of D/A and A/D Conversion Principles: (As covered in Theory, Sections 3.1 & 3.2, including formulas and parameters).**
2. **Detailed Interfacing Schematic for DAC0808: Showing connections to microprocessor data bus, control signals, power, reference, and op-amp circuit (if used externally). Clearly label all pins and components.**
3. **Detailed Interfacing Schematic for ADC0804: Showing connections to microprocessor data bus, control signals, power, reference, clock, potentiometer, and output display (LEDs/LCD). Clearly label all pins and components.**
4. **Assembly Code for DAC Waveform Generation: With comments explaining each step.**
5. **Assembly Code for ADC Reading and Display: With comments explaining each step.**
6. **Observed Waveform from Oscilloscope (DAC): A sketch or printout of the staircase/ramp waveform, clearly labeled with voltage and time axes.**
7. **Tabulated ADC Readings: A table showing at least 5-7 pairs of (Analog Input Voltage (measured by DVM), Corresponding Digital Output (observed from LEDs/LCD)).**
8. **Analysis and Conclusion: Discuss how the observed results match theoretical expectations and any practical limitations.**

8. Conclusion

This experiment provided hands-on experience in interfacing Analog-to-Digital and Digital-to-Analog converters with a microprocessor. We successfully interfaced a DAC0808 to generate a staircase analog waveform, observing its characteristics on an oscilloscope. Subsequently, we interfaced an ADC0804 to convert a variable analog input from a potentiometer into a digital value, which was then displayed. This practical exercise reinforced the understanding of A/D and D/A conversion principles, including resolution and conversion processes, and demonstrated the crucial role of these converters in enabling microprocessors to interact with real-world analog signals.
