EDC Experiment 5

EE23BTECH11035 - Kunal Thorawade
EE23BTECH110021 – Ganne Gopichandu

1 Build a 5 bit ADC

1.1 Aim

The aim of this experiment is to build a 5 bit Analog to Digital converter.

1.2 Components Required

- 3 Breadboards
- 4 10K ohm Resistors and 6 20K ohm Resistors
- Wires
- 6 7474 ICs
- 1 LM358 IC
- 5 LEDs

1.3 Theory

An Analog-to-Digital Converter (ADC) transforms a continuous analog signal into a discrete digital output. It samples the analog input, divides the input range into equal intervals called quantization levels, and assigns each level a binary code. The ADC's resolution, defined by the number of bits, determines how many discrete levels it can differentiate. A higher resolution results in finer detail but requires more bits. The conversion process introduces a small quantization error, as the analog value is approximated to the nearest level.

1.4 Circuit

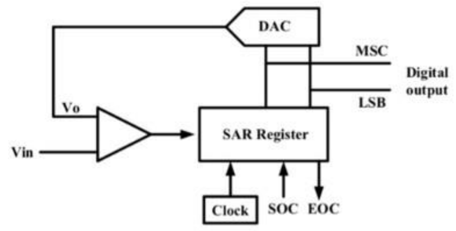


Figure 1: A 5-Bit ADC

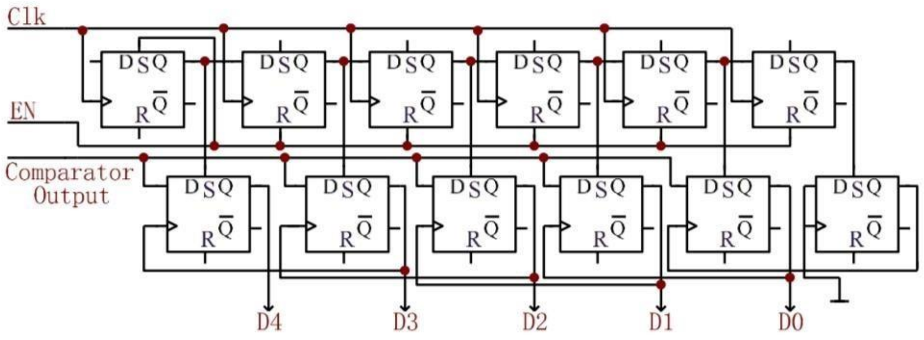


Figure 2: SAR Control Logic

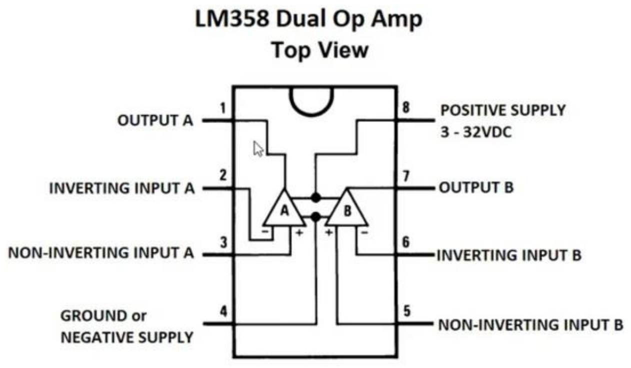


Figure 3: LM 358 Pinout

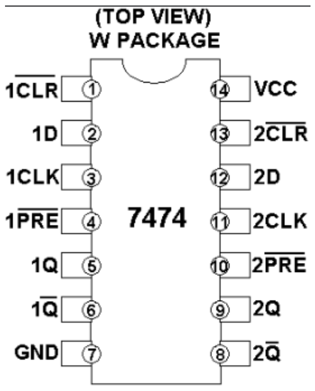


Figure 4: 7474 IC Pinout

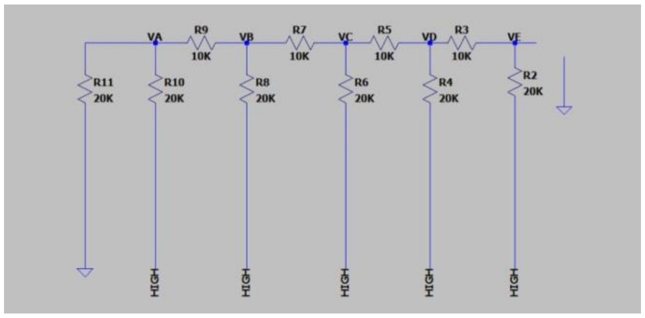


Figure 5: R-2R DAC Ladder

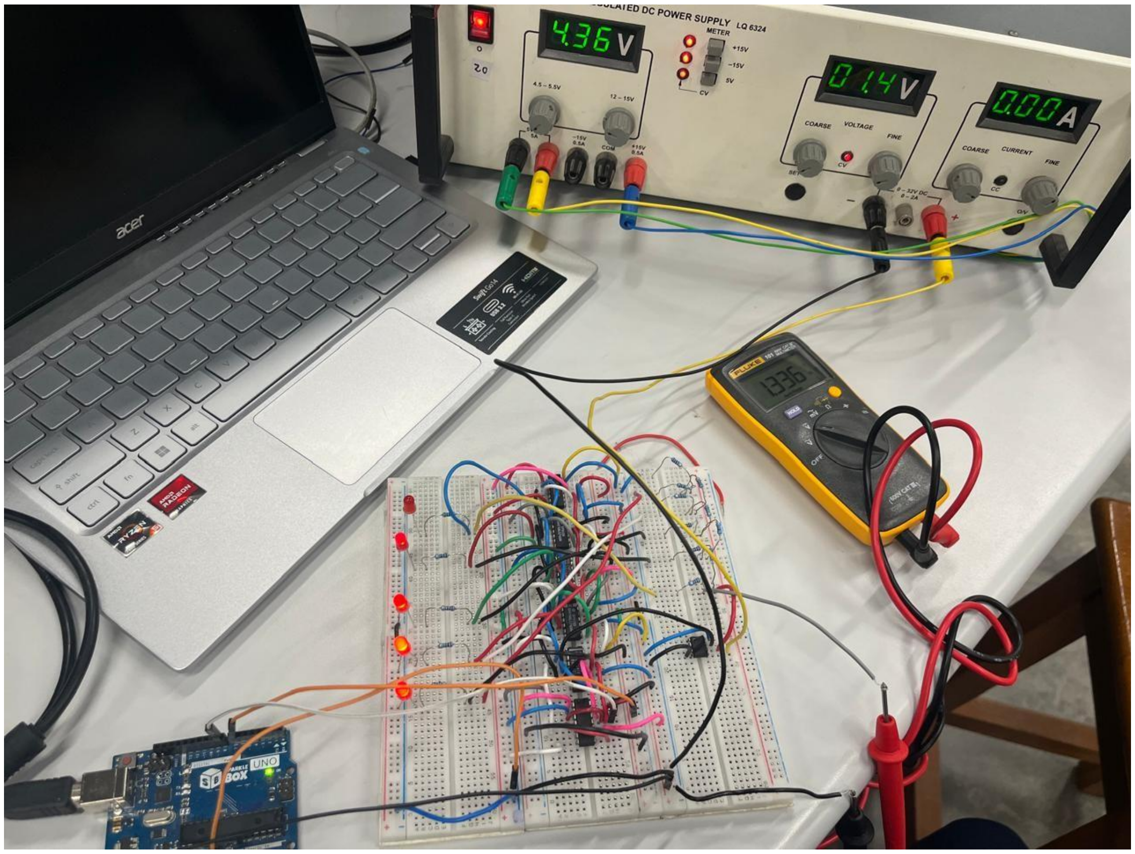
1.5 Procedure

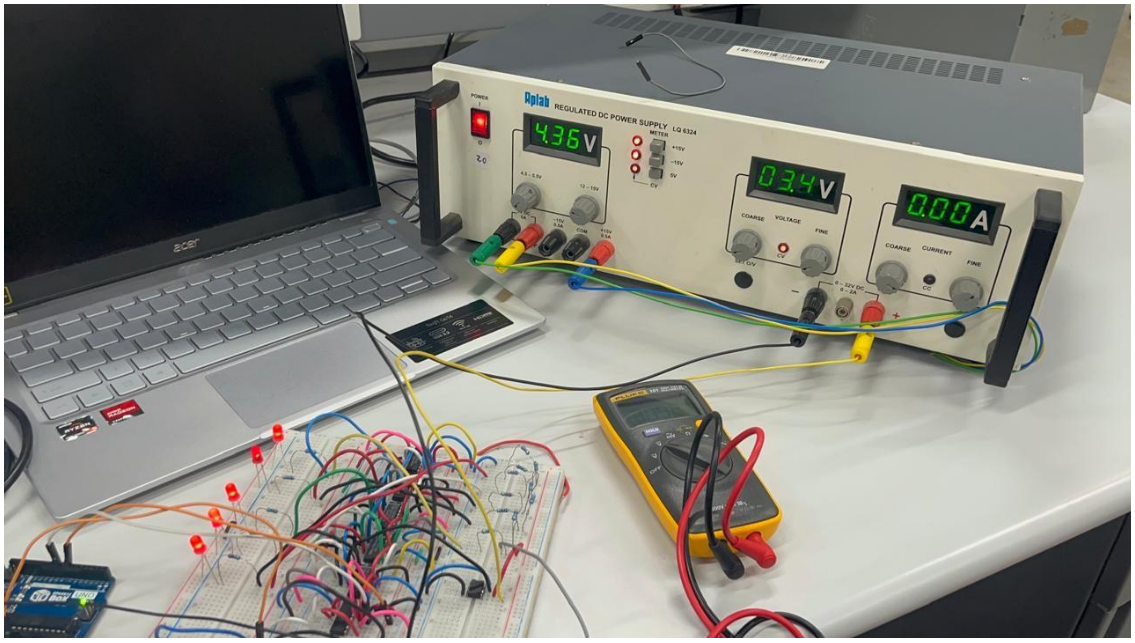
- **Prepare the Breadboard:** Set up a breadboard and insert the 7474 ICs and lm 358 into the appropriate sockets, ensuring the correct pin configuration.
- **Make the connections:** Complete the circuit by making appropriate connections for SAR , DAC and Comparator Circuit.
- **Verify Connections:** Double-check all connections to ensure they are secure and correct. Create an Arduino code for enable and clock. and feed them into the circuit.
- **Observe the Output after switching on Power:** Measure the binary output for different input voltages.

1.6 Working

In a 5-bit ADC using a comparator, DAC, and SAR, the Arduino provides clock and enable signals for the conversion. The process begins with the SAR setting the most significant bit (MSB) to 1, sending the digital code to the DAC, which converts it to an analog voltage. This voltage is then compared to the analog input using the comparator. The comparator outputs a high or low signal depending on whether the analog input is higher or lower than the DAC output. If the DAC voltage is lower, the bit stays 1; if higher, the bit resets to 0. The Arduino's clock signal advances the SAR to the next bit, continuing the comparison for each bit down to the least significant bit (LSB). The Arduino's enable signal starts the conversion cycle, and after all 5 bits are tested, the SAR holds the final 5-bit digital code representing the input. The comparator is crucial for determining the digital output bit-by-bit based on each comparison.

1.7 Observations





At nearly 3.4V we observe all the LEDs glowing which suggest that this voltage corresponds to 31 in binary through scaling. Similarly we see that at 1.4V (nearly half of 3.4V), we get 16 as our binary output which suggests that our ADC is working fine.