Experiment 10

American International University-Bangladesh
Department of Electrical and Electronic Engineering
EEE3120: Digital Logic & Circuits Laboratory

Title: Design of a Digital to Analog and Analog to Digital Converters.

Part I: Design of a Digital to Analog Converter

Introduction:

This lab describes the design of a Digital to Analog Converter (DAC). Two types of design are shown in this lab, binary weighted DAC and R/2R ladder DAC design. Finally student will compare both the design to conclude which design is efficient and why.

Theory and Methodology:

One common requirement in electronics is to convert signals back and forth between analog and digital forms. Most such conversions are ultimately based on a *digital-to-analog converter* circuit. Therefore, it is worth exploring just how we can convert a digital number that represents a voltage value into an actual analog voltage.

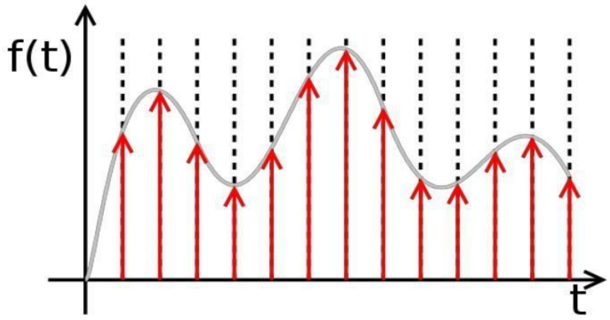
Digital-to-Analog Converters

In electronics, a digital-to-analog converter (DAC, D/A, D2A or D-to-A) is a function that converts digital data (usually binary) into an analog signal (current, voltage, or electric charge). An analog-to-digital converter (ADC) performs the reverse function. Unlike analog signals, Digital data can be transmitted, manipulated, and stored without degradation, albeit with more complex equipment. But a DAC is needed to convert the digital signal to analog, for example to drive an earphone or loudspeaker amplifier and produce sound (analog air pressure waves).

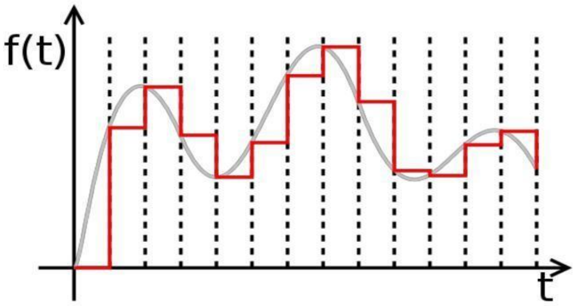
DACs and their inverse, ADCs, are part of an enabling technology that has contributed greatly to the 'digital revolution'. To illustrate this, consider a typical long-distance telephone call. The callers voice is converted into an analog electrical signal by a microphone. The analog signal is then converted to a digital stream by an ADC. That digital stream is then divided into packets where it will be mixed with other digital data, not necessarily audio. The digital packets are then sent to the destination, but each packet may take a completely different route and may not even arrive at the destination in the correct time order. The digital voice data is then extracted from the packets and assembled into a digital data stream. A DAC converts it into an analog electrical signal which drives an audio amplifier which in turn drives a loudspeaker which finally produces sound. Of course, this is a simplified and stylized description, but it does illustrate one vital role of ADCs and DACs.

There are several DAC architectures; the suitability of a DAC for a particular application is determined by six main parameters: physical size, power consumption, resolution, speed, accuracy, cost. Due to the complexity and the need for precisely matched components, all but the most specialist DACs are implemented as integrated circuits (ICs). Digital-to-analog conversion can degrade a signal, so a DAC should be specified that that has insignificant errors in terms of the application.

DACs are commonly used in music players to convert digital data streams into analogue audio signals. They are also used in televisions and mobile phones to convert digital video data into analog video signals which connect to the screen drivers to display monochrome or color images. These two applications use DACs at opposite ends of the speed/resolution trade-off. The audio DAC is a low speed high resolution type while the video DAC is a high speed low to medium resolution type. Discrete DACs would typically be extremely high speed low resolution power hungry types, as used in military radar systems. Very high speed test equipment, especially sampling oscilloscopes, may also use discrete DACS.



Ideally sampled signal



Piecewise constant output of an idealized DAC lacking a reconstruction filter. In a practical DAC, a filter or the finite bandwidth of the device smoothes out the step response into a continuous curve.

A digital-to-analog converter, or DAC for short, converts a digitally coded number to a voltage proportional to the number. For example, if a number N is supplied to a DAC, the output voltage will be proportional to N: $V_{out} = N \times B$ The constant of proportionality, B, is normally determined from the ratio of the reference voltage, V_{ref}, and the maximum value that N can have, N_{max}, $B = V_{ref} / N_{max}$ so that $V_{out} = V_{ref}\ N / N_{max}$ A common way to make a DAC is with an OpAmp circuit. Recall the circuit for the summing amplifier.

Binary Weighted Digital-to-Analog Converter:

The following circuit is a basic digital-to-analog (D to A) converter. It assumes a 4-bit binary number in Binary-Coded Decimal (BCD) format, using +5 volts as a logic 1 and 0 volts as a logic 0. It will convert the applied BCD number to a matching (inverted) output voltage. The digits 1, 2, 4, and 8 refer to the relative weights assigned to each input. Thus, 1 is the Least Significant Bit (LSB) of the input binary number, and 8 is the Most Significant Bit (MSB).

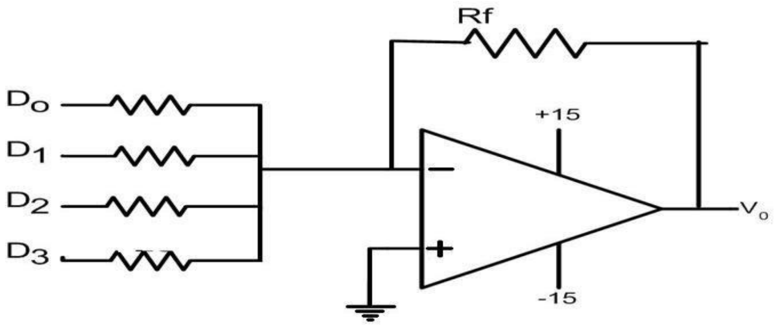


Fig1: Binary Weighted Digital to Analog converter.

If the input voltages are accurately 0 and +5 volts, then the "1" input will cause an output voltage of -5 × (4k/20k) = -5 × (1/5) = -1 volt whenever it is a logic 1. Similarly, the "2," "4," and "8" inputs will control output voltages of -2, -4, and -8 volts, respectively. As a result, the output voltage will take on one of 10 specific voltages, in accordance with the input BCD code.

Unfortunately, there are several practical problems with this circuit. First, most digital logic gates do not accurately produce 0 and +5 volts at their outputs. Therefore, the resulting analog voltages will be close, but not really accurate. In addition, the different input resistors will load the digital circuit outputs differently, which will almost certainly result in different voltages being applied to the summer inputs.

R/2R Ladder Digital-to-Analog Converter:

This improved circuit overcomes the problem of using many resistors. Instead it uses only two valued resistor.

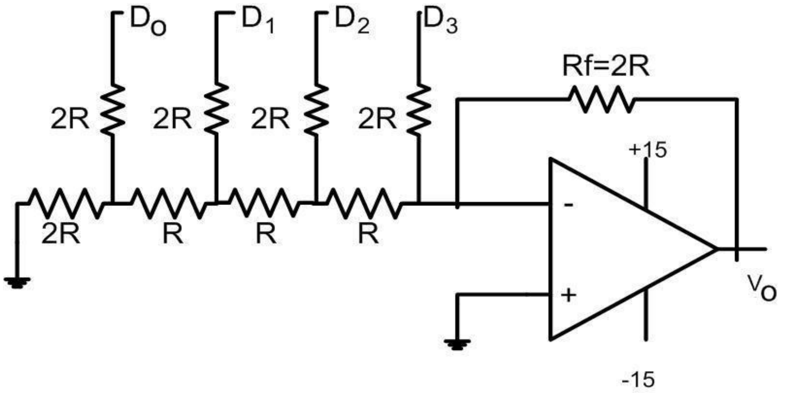


Fig 2: R/2R Ladder DAC

The circuit above performs D to A conversion a little differently. Typically the inputs are driven by CMOS gates, which have low but equal resistance for both logic 0 and logic 1. Also, if we use the same logic levels, CMOS gates really do provide +5 and 0 volts for their logic levels.

The input circuit is a remarkable design, known as an R-2R ladder network. It has several advantages over the basic summer circuit we saw first:

1. Only two resistance values are used anywhere in the entire circuit. This means that only two values of precision resistance are needed, in a resistance ratio of 2:1. This requirement is easy to meet, and not especially expensive.

2. The input resistance seen by each digital input is the same as for every other input. The actual impedance seen by each digital source gate is 3R. With a CMOS gate resistance of 200 ohms, we can use the very standard values of 10k and 20k for our resistors.

3. The circuit is indefinitely extensible for binary numbers. Thus, if we use binary inputs instead of BCD, we can simply double the length of the ladder network for an 8-bit number (0 to 255) or double it again for a 16-bit number (0 to 65535). We only need to add two resistors for each additional binary input.

4. The circuit lends itself to a non-inverting circuit configuration. Therefore we need not be concerned about intermediate inverters along the way. However, an inverting version can easily be configured if that is appropriate.

Pre-Lab Homework:

Why DACs has been an integral part of electronics for decades?

Where are DAC and ADC vastly used?

Apparatus:

1) IC741 OPAMP	1[pcs]
2) Resistors as required.	14[pcs]
3) Oscilloscope.	

Precautions:

Never turn on the DC source before the circuit is placed correctly and checked carefully.
Check for short circuits in the circuit**.**

Experimental Procedure:

1) First setup the Binary Weighted Digital to Analog converter as shown in Fig1 on the trainer board.
2) Put the following sequence 1010 to D0, D1, D2 and D3 respectively. See the output on the oscilloscope.
3) Again setup the R/2R ladder on the trainer board.
4) Repeat step 2 for R/2R DAC.

Simulation and Results:

1. Draw a plot on graph paper showing relationship between digital input and analog output of digital-to-analog-converter.

2. Calculate resolution and percentage resolution of each converter.

3. Use PSPICE for software simulation.

Questions for report writing:

1. Why R/2R Ladder Digital-to-Analog Converter is preferable than Binary weighted Digital-to-Analog Converter.

Discussion and Conclusion:

Interpret the data/findings and determine the extent to which the experiment was successful in complying with the goal that was initially set. Discuss any mistake you might have made while conducting the investigation and describe ways the study could have been improved.

Part II: Design of a flash Analog to Digital Converter

Introduction:

Flash analog-to-digital converters, also known as parallel ADCs, are the fastest way to convert an analog signal to a digital signal. Flash ADCs are ideal for applications requiring very large bandwidth, but they consume more power than other ADC architectures and are generally limited to 8-bit resolution. This tutorial will discuss flash converters and compare them with other converter types.

In this experiment, students will learn how flash ADC works by implementing a 2 bit flash ADC.

Theory and Methodology:

Flash converters are extremely fast compared to many other types of ADCs which usually narrow in on the "correct" answer over a series of stages. Compared to these, a Flash converter is also quite simple and, apart from the analog comparators, only requires logic for the final conversion to binary.

For best accuracy often a sample-and-hold circuit is inserted in front of the ADC input. This is needed for many ADC types (like successive approximation ADC), but for Flash ADCs there is no real need for this, because the comparators are the sampling devices.

A Flash converter requires a huge number of comparators compared to other ADCs, especially as the precision increases. A Flash converter requires $2^n - 1$comparators for an *n*-bit conversion. The size, power consumption and cost of all those comparators make Flash converters generally impractical for precisions much greater than 8 bits (255 comparators). In place of these comparators, most other ADCs substitute more complex logic and/or analog circuitry which can be scaled more easily for increased precision.

Implementation:

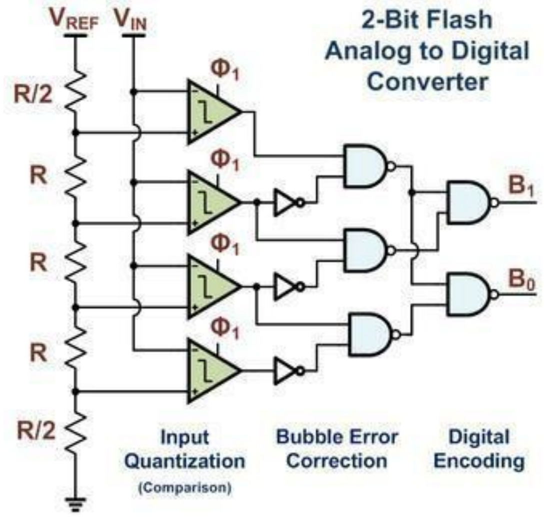


Fig1: A 2-bit Flash ADC Example Implementation with Bubble Error Correction and Digital Encoding

Flash ADCs have been implemented in many technologies, varying from silicon based bipolar (BJT) and complementary metal oxide FETs (CMOS) technologies to rarely used III-V technologies. Often this type of ADC is used as a first medium sized analog circuit verification.

The earliest implementations consisted of a reference ladder of well matched resistors connected to a reference voltage. Each tap at the resistor ladder is used for one comparator, possibly preceded by an amplification stage, and thus generates a logical '0' or '1' depending if the measured voltage is above or below the reference voltage of the resistor tap. The reason to add an amplifier is twofold: it amplifies the voltage difference and therefore suppresses the comparator offset, and the kick-back noise of the comparator towards the reference ladder is also strongly suppressed. Typically designs from 4-bit up to 6-bit, and sometimes 7-bit are produced.

Designs with power-saving capacitive reference ladders have been demonstrated. In addition to clocking the comparator(s), these systems also sample the reference value on the input stage. As the sampling is done at a very high rate, the leakage of the capacitors is negligible.

Recently, offset calibration has been introduced into flash ADC designs. Instead of high precision analog circuits (which increase component size to suppress variation) comparators with relatively large offset errors are measured and adjusted. A test signal is applied and the offset of each comparator is calibrated to below the LSB size of the ADC.

Another improvement to many flash ADCs is the inclusion of digital error correction. When the ADC is used in harsh environments or constructed from very small integrated circuit processes, there is a heightened risk a single comparator will randomly change state resulting in a wrong code. Bubble error correction is a digital correction mechanism that will prevent a comparator that has, for example, tripped high from reporting logic high if it is surrounded by comparators that are reporting logic low.

This circuit is the simplest to understand. It is constructed from a series of comparators, each one comparing the input signal to a unique reference voltage. The comparator outputs connect to the inputs of a priority encoder circuit, which then produces a binary output. The following illustration shows a Flash ADC 2-bit circuit:

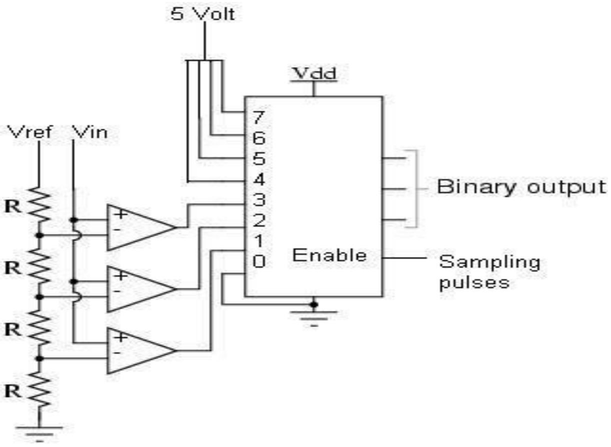
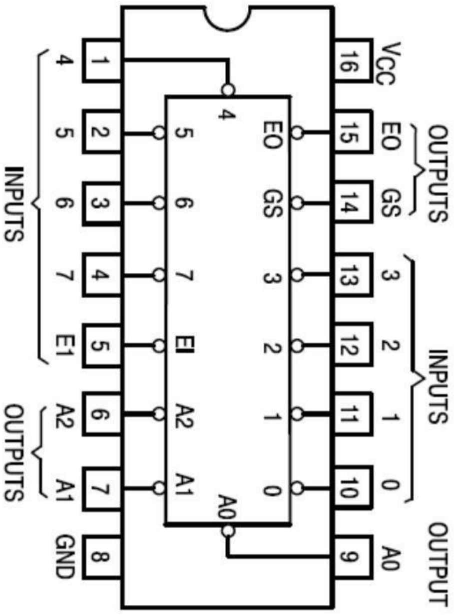


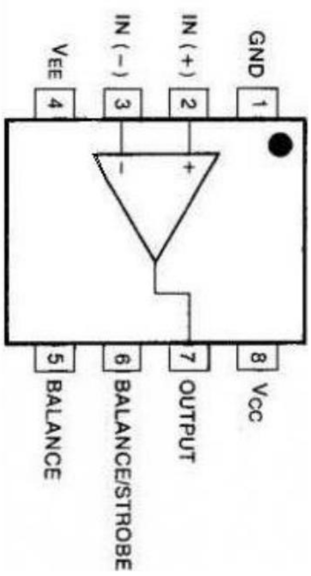
Fig 2: A 2 bit flash ADC.

An N bit flash ADC requires 2N – 1 number of comparators.

Vref is a stable reference voltage provided by a precision voltage regulator as part of the converter circuit, not shown in the schematic. As the analog input voltage exceeds the reference voltage at each comparator, the comparator outputs will sequentially saturate to a high state. The priority encoder generates a binary number based on the highest-order active input, ignoring all other active inputs.

IC Pin Configurations:





Pre-Lab Homework:

Students must study flash Analog to Digital Converter; perform simulation of the circuits mentioned in the lab manual using PSPICE and MUST present the simulation results to the instructor before the start of the experiment.

Apparatus:

1)	Trainer Board	:
2)	Op-Amp	: IC741
3)	Resistors	: 1KΩ
4)	8 to 3 bit priority encoder	: IC74148

Precautions:

Have your instructor check all your connections after you are done setting up the circuit and make sure that you apply only enough voltage to turn on the chip, otherwise it may get damaged.

Experimental Procedure:

Construct a 2 bit flash ADC. Document the output values for different input values. Draw the output wave shapes for different inputs.

Simulation and Measurement:

Compare the simulation results with your experimental data and comment on the differences (if any).

Discussion:

Interpret the data/findings and determine the extent to which the experiment was successful in complying with the goal that was initially set. Discuss any mistake you might have made while conducting the investigation and describe ways the study could have been improved.

Questions with answers for report writing:

Draw the wave shapes for binary output lines against analog input and sampling pulses.

Reference:

1. Thomas L. Floyd, *Digital Fundamentals,* 9th Edition, 2006, Prentice Hall.