# ENGR 210
Lab 12: Analog to Digital Conversion

In this lab you will investigate the operation and quantization effects of an A/D and D/A converter.

A. BACKGROUND

1. LED Displays

We have been using LEDs as indicators in the EWB simulations. Two ways to use LEDs as indicators are shown in Figure 1. Note that these circuits do not work with all logic families — many times the logic gates cannot supply the current which the LED needs. In those cases you have to use an external transistor to supply extra current. The technique of using an external transistor can also be used to operate relays and lights from a logic gate. Although only an inverter is shown in Figure 1 you can use any logic gate which has the necessary current output. The actual current is dependent upon the particular LED used. If we assume a current of 10 mA and a typical voltage across the LED of about 1.7 volts, we can calculate $R=(5-1.7)/0.01$ mA = $3.3/0.1=330\Omega$. Using resistors in Figure 1 which are much smaller than this value can burn out the output of the integrated circuit. Larger values are better but will dim the LED.
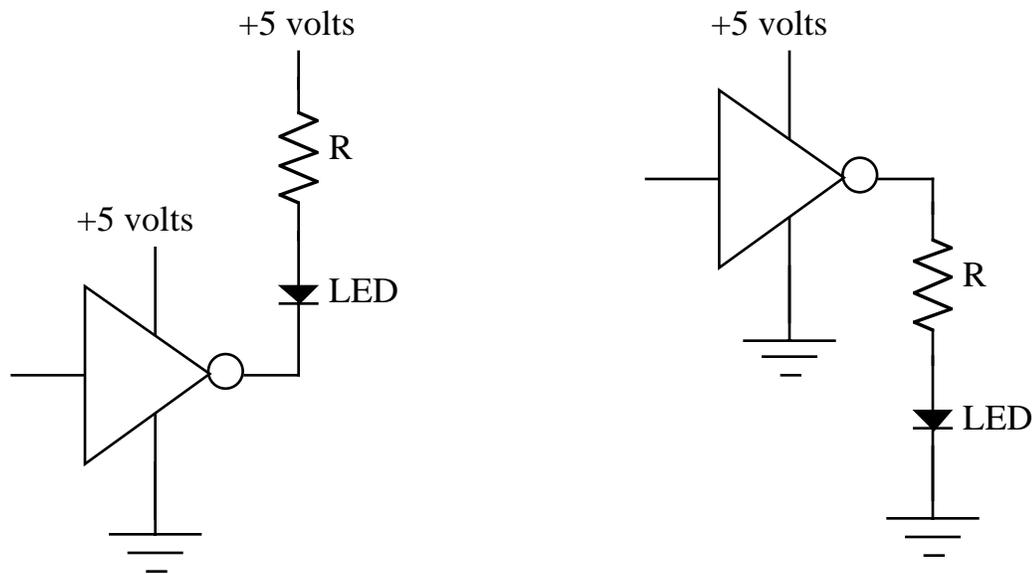
(a) Using TTL output as a current sink.   (b) Using TTL output as a current source

Figure 1. LED driver circuits

2. Analog to Digital (A/D) and Digital to Analog (D/A) conversion

Data that is obtained from a physical system is generally in analog form, i.e., it varies in time and can have any voltage between some maximum and minimum value. In order to do computer-based processing of that data it is necessary to first convert the information from analog to digital form. This is done by quantizing, or subdividing, the voltage range into a number of finite-size steps. The number of steps is usually 2^n-1, where n is the number of bits available for the binary word that will represent the analog voltage. An example of this quantization is shown in Figure 2 where n=3 has been chosen. As can be seen by comparing the two columns at the right of the figure,

there is always some error introduced in the conversion since no matter how many bits are chosen, there is always a step between levels.
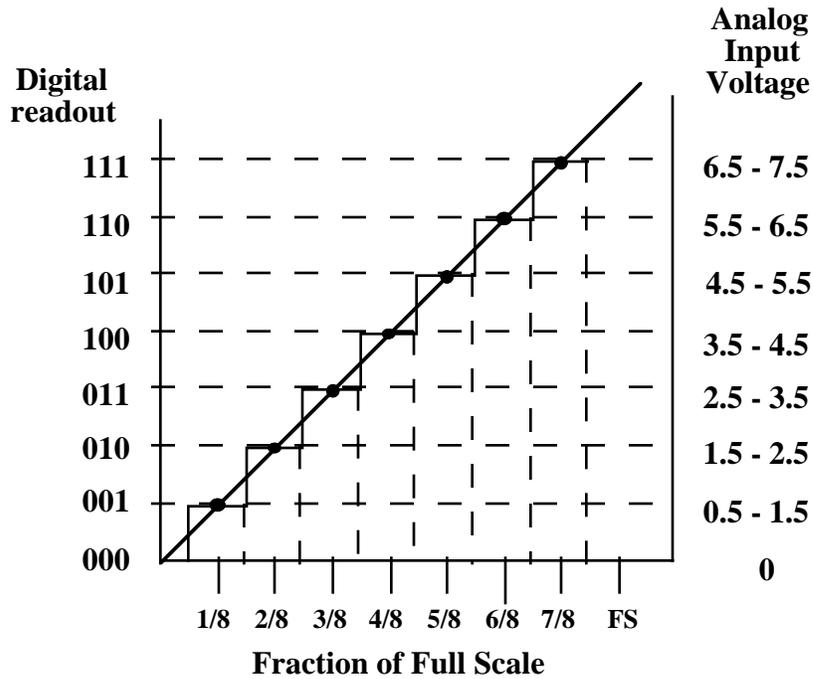

Figure 2. Three bit A/D quantization effect.

There are various approaches to the implementation of an A/D converter. One of these makes use of a set of comparators whose inputs are referenced to the quantized voltage levels. This is the fastest type of A/D converter available and is often called a "flash" converter. To demonstrate the quantization effect we will use an Electronics Workbench simulation. (This lab is a bit complicated to build and test in one lab period.) The fabricated circuit consists, in part, of an 8-bit, comparator-based flash A/D converter as shown in Figure 4. The output of the converter is an 8-bit code which is converted into a 3-bit binary code using the circuit of Figure 5. The output of this code converter is connected to the input of an op amp-based D/A converter (see Figure 3) whose output can, in turn, be displayed on a scope or a DMM.
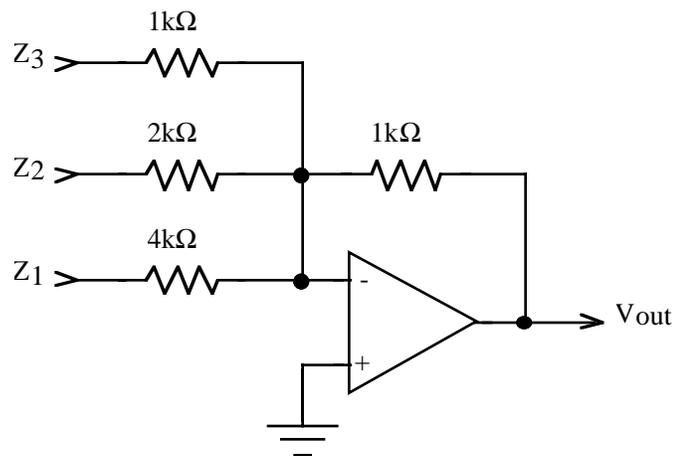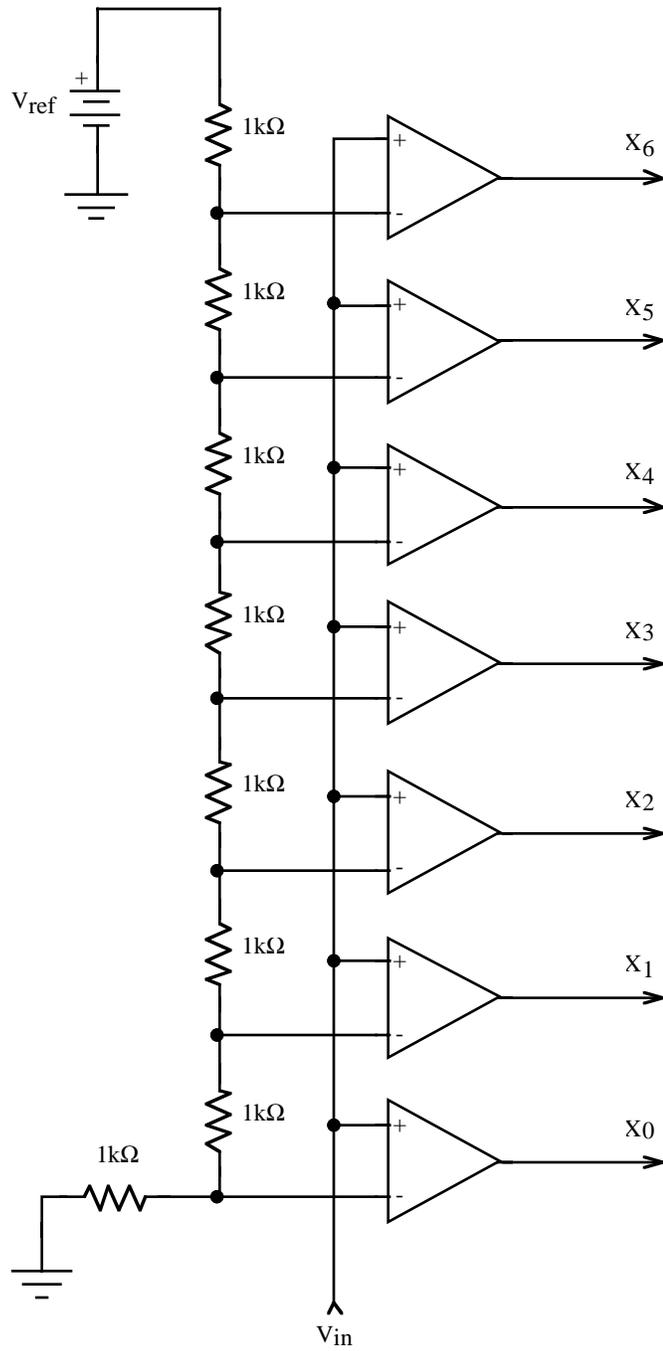

Figure 3. 3-bit D/A converter
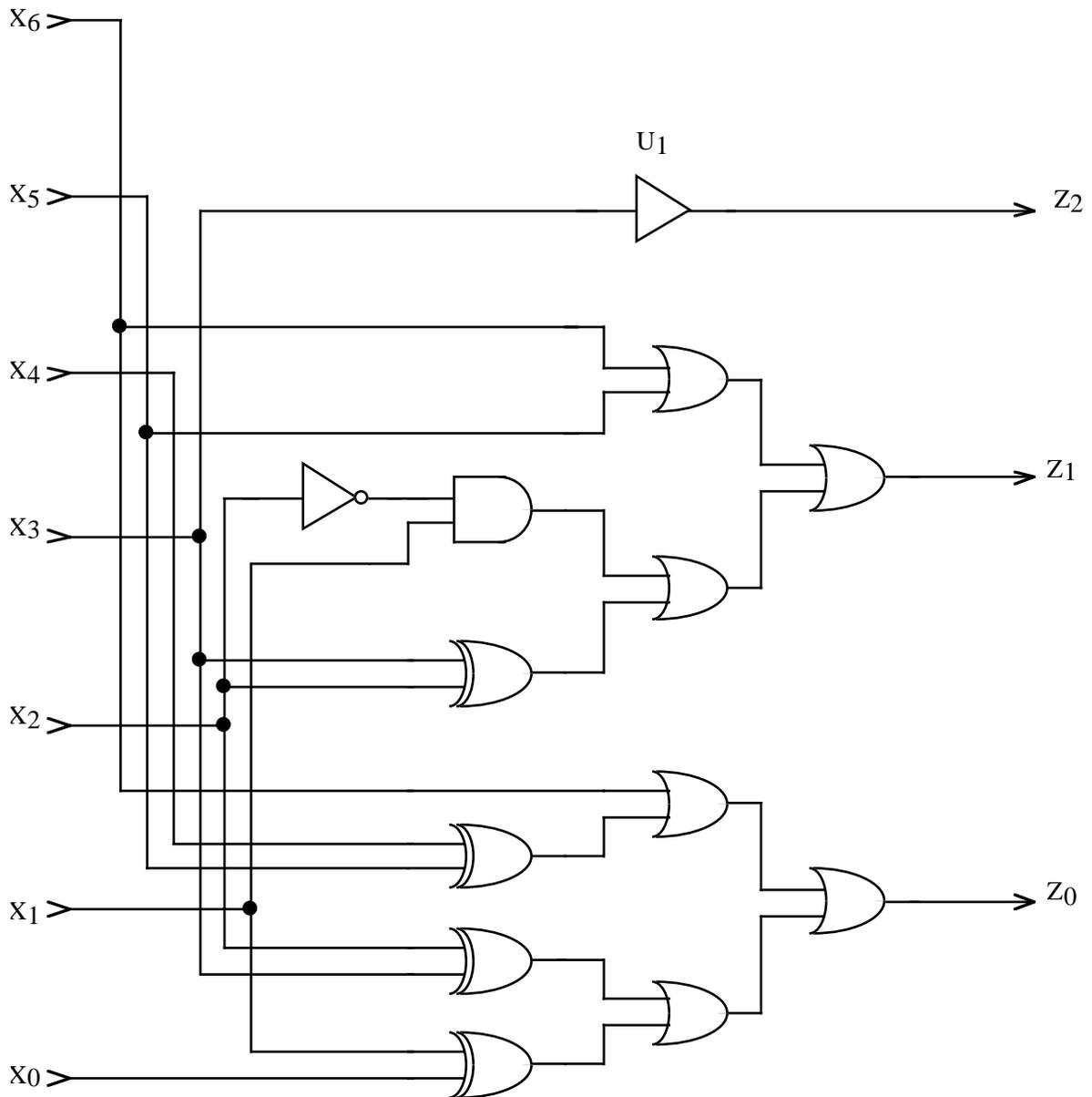
Figure 4. Flash A/D converter

Figure 5. 8-to-3 bit code converter

There are many other approaches to converting analog numbers to digital representations.

1. You have actually already seen one other type of A/D converter in the 555 timer lab — a 555 timer can be used as a V/F converter where an analog voltage controls the frequency of a 555 timer. A computer is then used to count the number of output pulses over a fixed period of time — the count is a digital representation of the analog input quantity.

2. Another common type of A/D converter uses an input voltage to charge up a capacitor. The time that it takes to discharge this voltage is timed by a computer or a digital counter. This is the type most often found in digital instruments such as a DMM and, depending upon the exact details of the implementation, is called a single or dual slope integrating A/D.

3. Yet another common A/D called the successive approximation A/D uses a computer to generate a digital code which is sent to a D/A converter. This digitally generated voltage is compared to

the input voltage with a comparator. If the input voltage is higher than the generated voltage the code number is increased, if the input voltage is less than the generated voltage the code number is decreased. The computer typically uses a binary search technique to rapidly find the largest code number which will generate the largest voltage less than or equal to the input voltage.

The difference between these converters lies in the details of how they work. The so-called "flash" A/D converter is the fastest. The dual slope integrating A/D converter is among the most accurate (especially using inexpensive components) and is among the slowest. The V/F converter is often the cheapest and is found in systems which use a microprocessor, such as computer controlled appliances.

B. LAB INSTRUCTIONS

Part 1: More logic circuits

(a) You will use a single 7402 logic gate to determine the actual LOW and HIGH logic levels of a logic gate. Select a single gate from the 7402. Tie its two inputs together. Connect these inputs to the positive adjustable power supply output using a 300Ω resistor — the exact value is not critical and you could use 330Ω, etc.   Make sure the adjustable power supply is set to zero volts. Measure the output of your selected gate with a DMM. It should be near +5 volts if the input is zero. Remember that you have it connected essentially as an inverter. Now slowly increase the power supply voltage while monitoring the output of the selected gate with a DMM. Record the voltage at which the output goes low in Data Table 1.
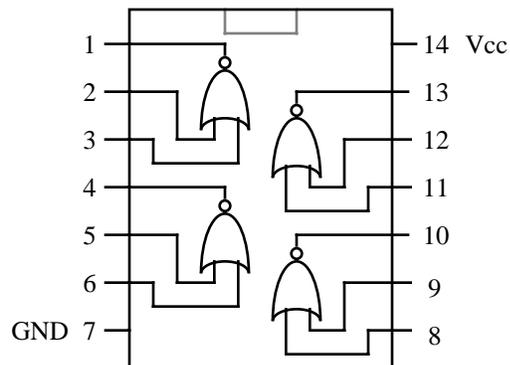

Figure 6. A 7402 IC contains 4 2-input NOR gates.

(b) Repeat Step 1 except now start the input voltage at +5 volts and decrease the input voltage. Since the input is initially HIGH the output should be LOW. At some value of the input voltage, the output will change. Record this voltage in Data Table 1.

(c) Add the circuit of Figure 1(a) to the output of your 7402 gate. Use the variable power supply to vary the input voltage to the 7402 gate between 0 and +5 volts. Does the LED function to indicate the output state of the gate? Measure the voltage drop across the LED and the resistor. Compute the current through the diode. Record these measurements in Data Table 2.

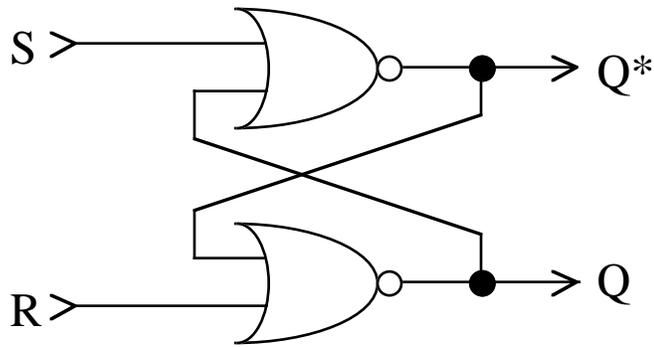(d) Using the gates on a 7402, build the circuit shown in Figure 7.

Figure 7. R-S Flip-flop

Determine the truth table for this circuit and record your measurements in Data Table 3. This is actually a R-S flip flop and can have very perplexing behavior. Unlike combinational logic it works based upon the two inputs (R and S) AND the current (called the previous) output. To help you understand this circuit the theoretical behavior of an R-S flip-flop is shown in Figure 8. Notice that the new output (Q_{n+1}) is described in terms of R, S and the previous value of Q (called Q_n). DO NOT CONFUSE Q_{n+1} with the Q* shown in Figure 7 — Q_{n+1} is the NEXT state (value) of the Q output of Figure 7. The actual operation of the R-S flip-flop can be more easily understood by a simple explanation. The R (also called the RESET) will always reset the Q output to zero (LOW) irregardless of the previous state of the output. The S (also called the SET) will always set the Q output to one (HIGH) irregardless of the previous state of the output. There will be no change in the Q output if both R and S inputs are zero (LOW). However, there is a problem if both R and S inputs are one (HIGH). Technically, you are trying to SET and RESET the output at the same time. As a result the output can be random and, consequently, undefined. In general, the Q* output of Figure 7 should behave as the NOT of Q and does not need to be separately explained.

S	R	Q_n (previous state)	Q_{n+1} (new state)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	undefined
1	1	1	undefined

Figure 8. Theoretical operation of R-S Flip-flop

Your final objective is to use the EWB simulation provided on the course Web page to observe and understand the operation of an A/D and D/A converter and the quantization effect that is produced by the overall circuit. The circuit diagram for the major components of this simulated circuit is shown in Figure 3, Figure 4 and Figure 5.

4. A voltage divider which can go from -0.5 volts to +5 volts is used to generate an input voltage for the A/D converter. This input voltage can be increased in 1% increments by typing a "r" on the keyboard; it can be decreased in 1% increments by typing "R" on the keyboard. The digital output of the A/D converter is shown by the horizontal row of LEDs labeled X0 to X6. This seven bit digital code is converted into a more conventional three-bit digital code by the logic shown. The three-bit code output, shown by the horizontal row of LEDs labeled Z0 to Z2, is input to a 3-bit D/A converter. Verify that you can change Vin by varying the resistor.

5. Using the EWB simulation complete the truth table for the A/D and D/A converter, i.e, what are the seven and three bit codes as a function of Vin? Record your results in Data Table 4. Also determine Vout as a function of Vin and the three-bit input code to the D/A. Also record these results in Data Table 4.

6. Replace the variable resistor R at the input of the A/D converter with the function generator. Adjust the function generator to produce a 0.5 Hz ramp of about 5 volts amplitude. None of these values are critical. Connect the oscilloscope to monitor Vin (the input to the A/D converter) and Vout (the output of the D/A converter). Run the simulation. The output of the D/A converter should be a waveform with discrete steps — this waveform is often called a staircase. You should also see some voltage "spikes" on the staircase output of the D/A converter. Adjust the oscilloscope to get a good display of Vin and the corresponding Vout for Vin>0. Make a printout of this voltage waveform.

DATA AND REPORT SHEETS FOR LAB 12

Student Name (Print): _____ Student ID: _____

Student Signature: _____ Date: _____

Student Name (Print): _____ Student ID: _____

Student Signature: _____ Date: _____

Student Name (Print): _____ Student ID: _____

Student Signature: _____ Date: _____

Lab Group: _____

Data Table 1. Voltage at which digital logic switches

	Input voltage at which output switches	Initial Measured V_{out}	Measured V_{out} after switch
Increasing V_{in} from 0 volts			
Decreasing V_{in} from +5 volts			

Data Table 2. LED indicator characteristics (for LED illuminated)

	Measured	**Calculated**
Voltage across R		
Voltage across LED		
Current through LED		

Data Table 3. Truth table for R-S flip-flop

R	S	Q_n	Q_{n+1}
0	1		
0	0		
1	0		
1	1		

Data Table 4. A/D and D/A Converter Characteristics

	X_6	X_5	X_4	X_3	X_2	X_1	X_0	Z_2	Z_1	Z_0	V_{OUT}
$V_{in} < 0V$								0	0	0	
$0 < V_{in} < 0.625$											
$0.625 < V_{in} < 1.25$											
$1.25 < V_{in} < 1.875$											
$1.875 < V_{in} < 2.5$											
$2.5 < V_{in} < 3.125$											
$3.125 < V_{in} < 3.75$											
$3.75 < V_{in} < 4.475$											
$V_{in} > 4.475V$											

Be sure to print out your simulation waveform and include it with your report.

Questions

1. Why does Data Table 4 only show seven input codes? HINT: Can the A/D converter generate a an output code $X_6X_5X_4X_3X_2X_1X_0 = 0001000$?

2. Using Data Table 4 explain how the digital logic converts the A/D converter outs $X_6X_5X_4X_3X_2X_1X_0$ into the three-bit code $Z_2Z_1Z_0$ used by the D/A converter.

3. What are the voltage levels at the output of the D/A converter? What range of voltage does each level of the input code $Z_2Z_1Z_0$ represent?

4. In step C.7 you recorded a time dependent response of the D/A output. This response showed some voltage spikes (often called D/A glitches). Why do these appear? HINT: Look at which bits are changing in the input code.

5. Suppose you wanted to redesign the A/D converter so that each input step corresponded to exactly 0.5 volts. An easy way to do this is to change the value of Vref. What should this new value of Vref be? Be sure to include your calculations.

6. The buffer labeled U1 is required in the simulated circuit for it to work properly. Why?