EE223 Laboratory #1

Digital to Analog Converter

Objectives:
1) Learn how superposition and Thevenin conversions are used to analyze practical circuits
2) Become familiar with ground bus and power bus notation
3) Understand D/A conversion concepts
4) Be able to design a D/A converter to meet a particular specification
5) Keep Spice skills current

Prelaboratory

In this experiment, you will use your knowledge of superposition and Thevenin conversions to analyze a popular digital to analog conversion circuit. This circuit is similar to those found in CD and DVD players and in computer sound cards. It has the ability to convert a set of bits (which each occupy one of two states) into a continuous analog signal (which may occupy many intermediate states) representing sound. The bits are stored as logic levels on data buses inside a computer, as pits on the surface of a CD, or as the state of switches in this lab.

Analog to digital conversion

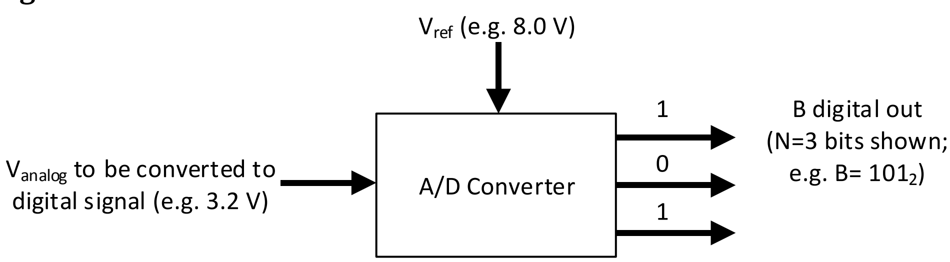

Figure 1- Block diagram of generic A/D Converter

In most of EE223, we deal exclusively with analog signals. An analog voltage or current is continuous and can occupy an infinite number of possible values (e.g. an analog voltage that varies between 0 and 5 volts may be 2.83623V). In contrast, a digital number must equal one of a finite number of values. For example, a 2 bit number can have binary values of {00_2, 01_2, 10_2, or 11_2} (where the subscript 2 indicates binary) corresponding to decimal numbers {0,1,2,3}. These values are usually written in binary and can be converted to an analog voltage by scaling the binary number by a constant.

To complete an analog to digital conversion (i.e. to take an analog voltage of 2.83623V and represent it with a binary number such as 101_2), we must know both the maximum possible range of the input signal (from 0V to a maximum traditionally called V_{ref}) and the number of bits that will be used to represent it (called N).

If there are N bits, there are 2^N unique voltages that can be encoded. For example, 3 bits can encode 000, 001, 010, 011, 100, 101, 110, and 111. A circuit powered by V_{ref} volts will divide that V_{ref} voltage into equal 2^N steps, so each voltage step will be $V_{ref}/2^N$.

You might expect that if the maximum possible input voltage was 4V (i.e. V_{ref} = 4) and we wanted to represent the input with 2 bits then the minimum number, 00_2, would correspond to the minimum voltage, 0V, and the maximum number, 11_2, would correspond to the maximum voltage, 4V. It makes sense, but that's wrong! The minimum voltage would be 0V, but the maximum is one step lower than V_{ref}, or $V_{ref} - V_{ref}/2^N$.

For example, to code {0V to 4V} into binary with N=2 bits, the largest value possible 11_2 (i.e. 3 in decimal) maps to 3V. If you instead could use a more precise N=8 bit encoder, 1111111111_2 (i.e. 255 in decimal) maps to 3.98V, and for N=16 bits, 11111111 1111111111_2 (65535_{10}) maps to 3.99994V. The scaling factor that is used to multiply the digital number to recover the analog voltage is therefore not arbitrary; it is a number such that an analog voltage range of {0V to V_{ref}} maps to a binary number equal to {0 to 2^N-1}, where N is the number of bits.

This is summarized in the following general formula that converts from an analog voltage V_{analog} to a number B, represented in binary using N bits:

$$B = \left\lfloor \frac{V_{analog}}{V_{ref}} 2^N \right\rfloor$$

The odd-looking braces indicates a *floor* function that always rounds *down* to an integer. As an example, in a digital recording system that uses a V_{ref} of 8V and uses groups of 4 bits to represent each sound level, a value of 4.63V coming in from the mike would be converted into $\left\lfloor \frac{4.63}{8}\left(2^4\right) \right\rfloor = 9$ (decimal) which would be recorded on the CD as a binary 1001_2 (i.e. N=4 bits, and so bit0=1, bit1=0, bit2=0, and bit3=1).

Digital to analog conversion

The CD player must accomplish the inverse action and produce an analog voltage level given the binary number. The equation it implements is:

$$V_{analog} = \frac{B}{2^N} V_{ref}$$

As an example, a 3-bit CD player using a V_{ref} of 8.0V and reading binary values 111, 100, 001 would output analog voltages of

$\frac{7}{2^3}8, \frac{4}{2^3}8, \frac{1}{2^3}8$ = (7, 4, 1) V. (Do you see that V_{ref} is simply a scaling factor with units of Volts?) A circuit that can accomplish this conversion is shown below, where binary digits are represented by switches.

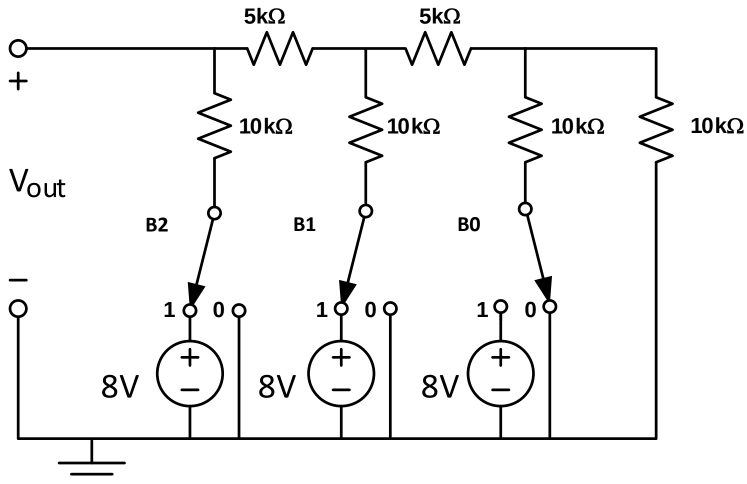

Figure 2 – D/A Converter Circuit. The switches are set to 110 binary (110 means B2=1, B1=1, B0=0) and are read left to right with the most significant bit on the left.

By hand analysis (not Spice), analyze the above circuit but alter the switches so that the circuit input is a binary 111 (i.e. all switches are set to 8.0V) and find V_{out} (show your work). **Use superposition to solve** (refer back to your EE122 notes if necessary). This will require first drawing **three separate circuits** (001, 010, 100). **For each of these three circuits**, solve using Norton/Thevenin source transforms and resistor simplification. Then, the result of 111 is simply the sum of the solutions for the 001, 010, and 100 cases. These hand calculations should be in your prelab.

Verify using Spice (you will have to "hard code" the switches as wired connections, so a 0 means a wire connected to ground and a 1 would mean the wire is connected to an 8V source), and attach a printout. Forgotten how to use Spice? Refer to the Spice walkthrough on my website (www.jimsquire.com) under Teaching – EE223 – Lab 1.

By superposition, you now know the contribution each bit makes to V_{out} !
For instance, the output for an input of 011 by superposition is your output for 001 plus your output for 010 (i.e. 001 + 010 = 011). For your prelab, finish filling out the table below (and copy it to your prelab – your prelab has to be complete by itself).

B2	B1	B0	Binary	Decimal	V_{out} predicted
GND	GND	GND	000	0	
GND	GND	8V	001	1	
			010	2	
			011	3	
				4	
				5	
				6	
				7	

In summary, for your prelab:

1) Read this entire packet and be ready to start breadboarding.

2) Your prelab document should have 3 circuits analyzed corresponding to 001, 010, and 100 by hand as discussed above, showing your work. Analyze the switches by pretending they are either opens or shorts, depending on whether the switch is open or closed, and sum the results to find 111. Each lab partner has to have their own, individual prelab – although you work the lab during the lab hour in pairs, the prelabs you must do on your own. You may consult your lab partner for help working the prelab problems if you don't understand a particular part, but if you find yourself copying his/her work because (although you may understand their work) you cannot do it independently, then you are setting yourself up for disaster on the first test when you won't be able to rely on their help.

3) Complete the above table in your individual prelab.

4) Attach a Spice printout of the circuit, again, for each student, not a single one shared among lab teams, for an input signal of 111. Model the switches as a open or short wires (i.e. replace each switch with either an 8V source or a ground). Check the output to make sure it agrees with your hand calculations in step 2 (you can screen-capture the Spice results).

Have your individual prelabs out at the start of the lab for grading in the first five minutes of class. You need your prelab results to check your lab work; if you do not have it complete you will have to leave the lab to complete them, and then complete the lab with whatever time you have remaining. You do not want to fall behind this power curve – complete the prelab completely, individually, and you will ace the lab.

Laboratory

Procedure:
Create the circuit shown in Figure 2 and find the experimental voltages for each binary combination (Think: how will you obtain a 5kΩ resistor when the closest standard value is 5.1kΩ? Consider using two 10kΩs…). Use the switches at the bottom of your protoboard to provide +8V and ground. Don't forget to measure and report the true values of the resistors.

Part II (OPTIONAL: 5 points) 6-bit D/A converter

Construct a 6-bit D/A converter and show the instructor. Answer these additional questions in your lab report:
How many discrete output voltages can this circuit render?
What is the smallest increment by which you can change the output voltage?
Make a table for at least 6 switch combinations. Predict and measure the output voltage. How well do they compare?

Discussion questions:
Always answer the discussion questions quantitatively. For instance, not "My errors were caused by resistor tolerance" but "The maximum errors were less than 4%. Since the resistors are only toleranced to 5%, resistor tolerance is the major source of error."

1. Common CD players record 16 bits of information for every voltage reading.
   a) How many discrete output voltages can 16 bits of information render?
   b) What is the smallest increment by which the output can vary given a V_{ref} = 1.25V?
   c) What are the lowest and highest voltages it can output?

2. The circuit you constructed can output anywhere from 0 to 7V. An engineering firm requests you design a different 4 bit D/A converter that can output from 0 (with an input of 0000) to 10V (an input of 1111).
   a) What V_{ref} would you select? (Hint: It is *not* 8, 10, 11, 15, or 16V)
   b) What is the resolution (smallest voltage step)?

3. (difficult: use Spice to help) Say you desire to construct an 8 bit D/A converter with a V_{ref}=5V using 5k and 10k resistors. 10k resistors in a lab are not exactly 10k; any one chosen may be anywhere within 5% of nominal (e.g. from 9.5k to 10.5k).
   a) How precise, in %, must the resistor be that has the most control over the output (e.g. the leftmost 10k in the 3bit converter of Figure 1) so that, assuming all the other resistors are perfect, the output value varies less than one least-significant bit's worth? It makes no sense to design a bit converter whose output varies more from resistor tolerances than from its least significant bit.
   b) Could you design it using our 5% resistors? It would be pointless to add additional bits to the design unless better resistors could be found.