

YOUR NAME:

YOUR SID:

YOUR PARTNER'S NAME:

YOUR PARTNER'S SID:

Pre-Lab \_\_/20

In-Lab \_\_/35

Post-Lab \_\_/45

Total \_\_/100

# Analog to Digital Converters

---

Lab 7: Analog To Digital Converters

ELECTRICAL ENGINEERING 40

INTRODUCTION TO MICROELECTRONIC CIRCUITS

University Of California, Berkeley

Department of Electrical Engineering and Computer Sciences

Professor Michel Maharbiz, Professor Vivek Subramanian, Vincent Lee, Linda Dai

---

## Lab Contents:

1. Lab Objectives
2. Pre-Lab Component
  - a. Background and Theory
  - b. Converting From Analog to Digital
  - c. The Analog to Digital Converter
    - i. The Comparators
    - ii. The Rest of the Circuit
3. Lab Components – Building the Circuit
  - a. Design an ADC with One More Output
4. Lab Report Submissions
  - a. Image Citations

## Lab Objectives

---

This lab is designed to introduce you to the reasoning behind and vital applications of **Analog-to-Digital Converters (ADC)** and **Digital-to-Analog Converters (DACs)**. In this lab we will be introducing how they work and why they're important.

In the pre-lab we will start by taking a look at a simplified ADC and simulate it. We will then build a simple 2 bit ADC in the lab and then scale it up to a 3 bit ADC.

## Pre-Lab Component

---

### Background and Theory

The ADC is a circuit that takes an analog signal and converts it into discrete measurement values which we can play with on a computer or signal processor. This is a process known as discretization and is at the heart of **Digital Signal Processing (DSP)**. The discrete values generally are limited to a specific set of values governed by bandwidth and resolution constraints. In most applications, the amount of data we can represent is limited to the number of binary digits available.

Recall that if we have  $n$  binary digits, we can represent  $2^n$  things. In ADC and DAC circuit applications, each of these "things" corresponds to a voltage level, where the voltage step size is given by the voltage range divided by the number of possible things we can represent. We will discuss this relationship more explicitly in a few paragraphs.

Similarly, a DAC takes a digital signal (suppose with  $n$  binary digits) and converts it into an analog waveform. How accurately the original waveform is reconstructed depends on the type of reconstruction technique applied. The most common reconstruction technique is the zero-order hold which simply outputs a constant voltage for the duration of each sample.

ADCs and DACs are fundamental circuits that allow us to go from the analog to digital world. As the name implies, the ADC converts an analog signal or continuous signal, into a digital signal or discrete signal. Often times when we convert from analog to digital we will sample the digital reading periodically before we store it on memory.

ADCs and DACs are also very important concepts at the hardware software layer boundary. Since computer systems run off of binary data signals, which are essentially discrete signals, being able to convert between the analog world and digital world is fundamental.

### Converting From Analog to Digital

As we discussed earlier, the purpose of an analog to digital converter is to take an analog signal and convert it to a discrete signal. This is because digital signals can be interpreted and processed by a computer or stored on memory. So how exactly do we do this?

There are several considerations that we need to account for to understand ADC conversion. First off, we need to consider the bandwidth available for our data or how many "things" we can represent. This is usually determined by the number of outputs we have from the ADC. From here on out, we're going to assume that we have  $n$  outputs from our ADC. As mentioned before, this allows us to represent  $2^n$  "things". In circuits, these "things" will correspond to a discrete set of voltage levels that we want to represent.

For this lab, we will assume that each of these voltage steps **are evenly spaced** apart over the entire range of voltages from 0 to  $V_{cc}$ .<sup>1</sup> This means that if we have  $n$  binary digits, the increment between each voltage step is given by:

$$\Delta V = \frac{V_{cc}}{2^n}$$

Now suppose we want to determine the numerical value  $N$  that corresponds to a certain voltage  $V_{in}$ . For a voltage  $V_{in}$ , the corresponding value  $N$  that the ADC will output is the largest multiple of  $\Delta V$  less than or equal to  $V_{in}$ . For  $0 < V_{in} < V_{cc}$ , this is given by:

$$N = \left\lfloor \frac{V_{in}(2^n)}{V_{cc}} \right\rfloor$$

Where  $\lfloor \cdot \rfloor$  is the floor function. Take some time to analyze and understand why this equation makes sense. Below is an example of how we would convert the input voltage to its digital representation for several values:

For  $V_{cc} = 5V$ ,  $n = 8$ , and  $V_{in} = 3.4523V$

$$N = \frac{V_{in}(2^n)}{V_{cc}} = \frac{3.4523 \times (2^8)}{5} \approx [176.758] \rightarrow N = 176$$

We take the floor function denoted  $\lfloor * \rfloor$  since the value for  $N$  must be an integer.

Now that you are familiar with the conversion formula, it's your turn to try a few of them yourself.

Suppose  $V_{cc} = 15V$ ,  $n = 16$ ,  $V_{in} = 9.54V$ , compute the value for  $N$  in the space provided below.

Score \_\_/10

<sup>1</sup> It is also common practice to distribute the step logarithmically or bias the voltage steps so that they occur closer together in the range where higher resolution is needed or preferable.

Now suppose we have an analog signal that is a sinusoid which is given by  $V_{in}(t) = 2.5[\sin(100\pi t) + 1]$ . If we use an ADC with  $V_{cc} = 5V$ , and  $n = 4$ , and sample the input at  $t = 0$ ,  $t = .0025$ ,  $t = .005$ , and  $t = .0075$ , what is the sequence of readings  $N_0, N_1, N_2, N_3$  that you will get. In addition to finding the values for  $N$ , convert these numbers into binary.<sup>2</sup>

Score \_\_/10

In this lab, we will use  $n = 2$  for simplicity. However, most practical ADCs have many more bits of resolution. For example, all audio converters have at least 16 bits, while some have as many as 24 bits. Image sensors generally use ADCs that have between 8 and 12 bits of resolution. In general, there is a tradeoff between how quickly an ADC can sample its input signal and how many bits of resolution it has. ADCs are almost always implemented as integrated circuits.

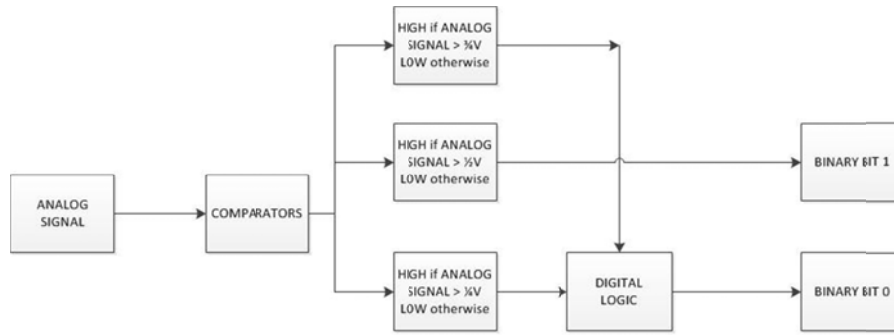
---

<sup>2</sup> These binary numbers are what would be transferred over the  $n$  output wires where the  $n$ th wire corresponds to the  $n$ th binary digit coming out of the ADC.

## The Analog to Digital Converter

Now that we have some understanding of how ADCs work, we can now move forward to the circuit that you will be building in lab. Obviously we'll be building an ADC, however to make your lives easier, we will be building a fairly simple ADC and then make you design a more complicated one as a thought experiment later.

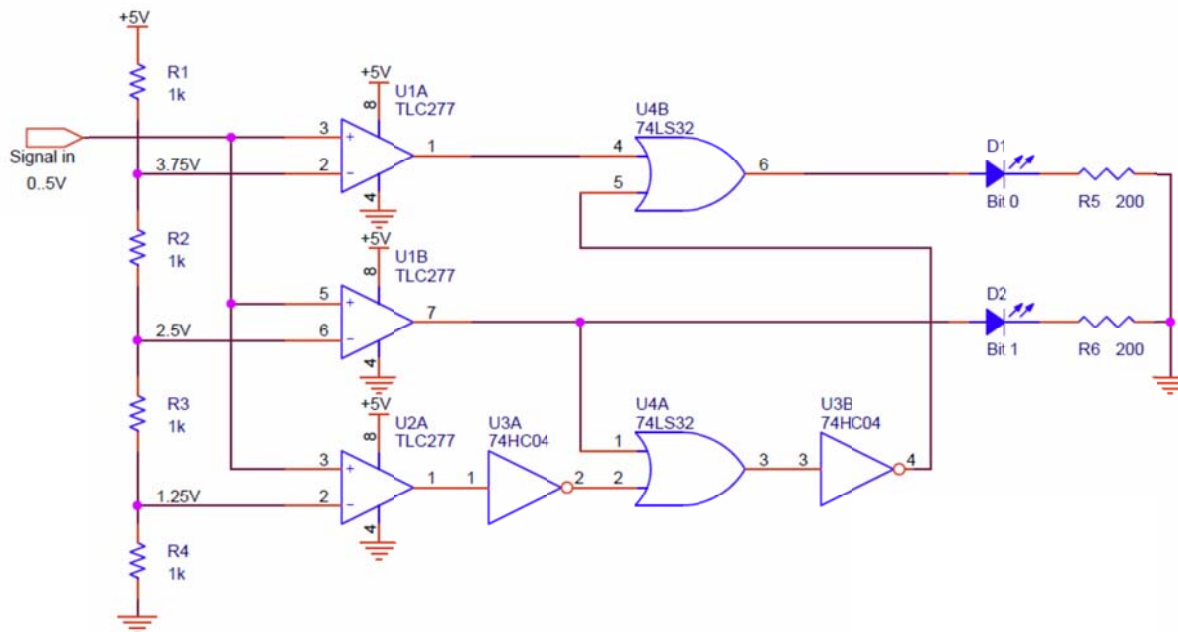
Below is a system diagram of what we want to implement.



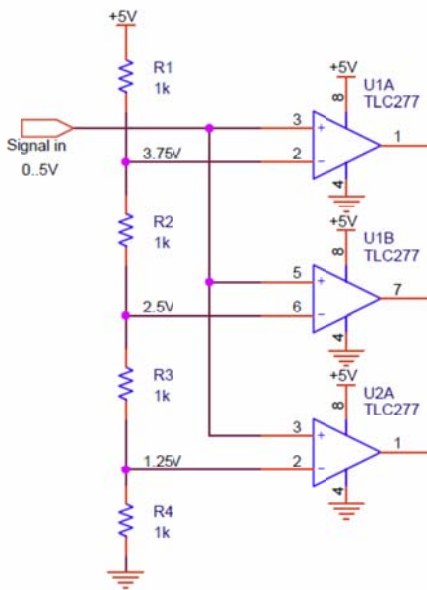
Since for this lab we will be using  $n = 2$ , we will need to figure out a way to represent the binary numbers 00, 01, 10, and 11. We refer to each digit position as Binary Bit 0 and Binary Bit 1 in the system diagram above.

You will also notice in the system diagram that we need to perform some digital logic since Binary Bit 0 should be on for 00, and 10 (not for 01, which is between the two). We do not need to perform any digital logic for Binary Bit 1 since it only needs to be on for 10 and 11 (which can be decided by a single comparator).

In terms of the schematic, we will provide you with a schematic this time ☺. You will notice that there are three operational amplifiers (configured in comparator mode), a resistor ladder that generates the reference voltages for each comparator, two inverters, and two OR gates. This configuration is known as a flash ADC, and its operation is described in the next section.



## The Comparators



In the circuit schematic, you will find that the analog signal is first fed into a bank of comparators which determine the voltage level. The comparators do this by comparing the input voltage with each possible voltage increment. In the skeletal circuit we gave you, each of the TLC277 operational amplifiers is configured as a comparator and takes in the analog signal as an input to the non-inverting terminal. This analog input is compared to voltages generated by the resistor ladder, which are  $\frac{3}{4}V_{cc}$ ,  $\frac{1}{2}V_{cc}$ , and  $\frac{1}{4}V_{cc}$ .

The comparator will output high (5V) only if the input analog signal is higher than the voltage provided by the voltage divider. It will output low (0V) otherwise. As the input voltage is increased from zero to 5V, the outputs of the comparators will transition between the 4 possible codes: 000, 001, 011, and 111. This is known as “thermometer code”, and must be converted to binary for further use.

## Thermometer Decoder

The digital logic implements a very simple 2-bit thermometer decoder. The operation is described by the following truth table. Note that there are 4 other possible input combinations, which will never occur in our circuit; by ignoring them, we can simplify the decoding logic. Input A is the output of the top comparator, and input C is the output of the bottom comparator.

| A | B | C | B1 | B0 |
|---|---|---|----|----|
| 0 | 0 | 0 | 0  | 0  |
| 0 | 0 | 1 | 0  | 1  |
| 0 | 1 | 1 | 1  | 0  |
| 1 | 1 | 1 | 1  | 1  |

The logical operation performed for the least significant binary bit (B0) is  $B0 = A + \bar{B} \cdot C$ , where the ‘.’ is the logical AND, ‘+’ is the logical OR and is the ‘ $\bar{x}$ ’ is the logical NOT operation. To minimize the number of different chips needed to implement this function, we can use De Morgan’s law to replace the AND operation with an OR:  $\bar{B} \cdot C = \overline{\overline{\bar{B}} \cdot \overline{\bar{C}}} = \overline{\bar{B} + \bar{C}}$ .

The most significant binary bit (B1) does not require any decoding logic, because it is the same as the input B.

**Lab Component - Building the Circuit**

---

Now that we've briefly taken a look at how the circuit works, we now have to build it.

We recommend that you build your circuit in modules and frequently check to make sure that your connections are correct. We labeled the pin outs for you on the schematic but it's your job to ensure that you connect everything correctly. A good way to split the circuit is to build the comparators and the digital logic as two separate blocks.

Note that the 74LS32 Logic OR gate is a QUAD chip. This means that the actual chip has 4 gates each, so 2 of these gates are not shown on our schematic. The 74HC04 inverter (NOT) gate is a hex chip. Also make sure to connect the power to the 74XX chips since that is also not shown on the circuit diagram (pin 14 is +5V, and pin 7 is GND). Google the other 74XX datasheets for finding out the pin connections.

--

With all this in mind, using the schematic, let's build the circuit.

We will be using the TLC277 operational amplifier and the 74XX logic ICs.

Since we are using many ICs<sup>3</sup> in this lab, we'll recap on some of the things that you should do to avoid the gods of frustration:

- SET CURRENT LIMITS
  - o Since we are now working with ICs, it is important to set current limits in the event that you unintentionally short something.<sup>4</sup> A reasonable current limit for this circuit is 75 mA.
  - o If you short-circuit your IC without setting current limits, it's very likely that you killed it and it no longer works. If your IC begins to feel like it can warm a cup of coffee, it's probably fried.
- CONNECT POWER
  - o ICs are NOT passive elements like resistors and capacitors. They require power and will not operate if you do not supply them power. The pin outs for power are usually specified on the datasheet.
  - o Be careful when connecting power supplies, if you connect power to the wrong terminals you can potentially burn out the chip. If you connect it backwards, you will definitely fry it.
- POLARITY AND IC ORIENTATION
  - o Components such as the electrolytic capacitors and LED diodes must be connected in a certain direction to function correctly. These components are said to be polarized or unidirectional and will only function correctly if connected the proper way. The LED diodes we will use in this lab are

---

<sup>3</sup> IC stands for Integrated Circuit

<sup>4</sup> We have extra parts in case you kill too many of your chips, however, after a reasonable number of disasters we reserve the right to start deducting points.

unidirectional and will only light if the positive lead voltage is higher than the negative lead. **The positive lead is always longer than the negative lead on polarized components.**

- Integrated circuits come in different sizes and have varying number of pin outs. However, the numbering scheme is always the same starting from the top left corner of the package. The “top” of an IC circuit is usually indicated by a half circle on the top of the IC package or a circle on the top left corner. Pin 1 is always located on the top left corner of the package and the numbering increases counter-clockwise around the package until reaching the top right pin. **Make sure when you place your ICs they’re not backwards.**

--

When you have finished building your circuit, test it to make sure that it performs as we expected from the pre-lab.

You can test it by either using a potentiometer to generate a constant DC analog signal, or you can use the function generator and try putting in a sine or triangle wave with a long enough period.<sup>5</sup>

It should perform approximately like the following:

- If the input voltage  $V_{in}$  is between  $0V$  and  $1.25V$ , neither LED should be lit.
- If  $1.25V < V_{in} < 2.5V$  the first binary digit LED should be lit and the other LED should not be lit.
- If  $2.5V < V_{in} < 3.75V$  the second binary digit LED should be lit and the other LED should not be lit.
- If  $3.75V < V_{in}$  the both LEDs should be lit.

Once you have verified that your circuit works, show your working ADC circuit to your GSI for check off.

Your GSI Signs Here **(35 pts)**



---

<sup>5</sup> If you use the function generator, we recommend that you use a frequency of 0.5Hz, a peak-to-peak amplitude of 5V, and a DC offset of 2.5V



### Design an ADC with One More Output

---

While building an ADC with two outputs ( $n = 2$ ) is nice, it's not incredibly useful. Your task is to design an ADC with  $n = 3$ . You must satisfy the following in your design and simulation.

- You must designate 3 LEDs as your outputs. Clearly label them on your schematic by using the Text tool located under Place -> Text.
- You must generate a test input of some kind using either a potentiometer or varying voltage source. Clearly label this as "ANALOG SIGNAL" in your design.
- There is no limit to the number of resistors and operational amplifiers that you can use.
- You may NOT use an ADC IC chip of ANY kind.
- You may use any number and type of logic gates.

Here are some hints to get you started if you need a starting point.

- First consider how you need to split up the range between  $0V$  and  $5V$ . What voltages do you need to supply to the comparators? What voltage dividers will you need to use?
- After you send the signal through the comparator bank, you will have to somehow break down the 8 outputs into its binary form since the output is only 3 binary digits. Think about how we formulated the reasoning behind our ADC for  $n = 2$ . Did we make any optimizations?
- Think about what logic gates you will need. We used an OR gate (74HC32) for our other ADC, will we need AND and/or NOT gate(s) for this design?

Once you have designed your 3-bit ADC, throw it into Multisim and verify that it works.

Once your circuit works, **attach a copy of your schematic** to this lab document. **(45 pts)**

### Lab Report Submissions

---

This lab is **due at the beginning** of the next lab section. Make sure you have **completed all questions** and **drawn all the diagrams** for this lab. In addition, attach any loose papers specified by the lab and submit them with this document.

These labs are designed to be completed in **groups of two**. Only one person in your team is required to submit the lab report. Make sure the names and student IDs of **BOTH** team members are on this document (preferably on the front).

### Image Citations

---

Textbook Images are courtesy of Fawwaz T. Ulaby and Michel M. Maharbiz and National Technology and Science Press.

Fawwaz T. Ulaby and Michel M. Maharbiz, Circuits © 2009 National Technology and Science Press