Digital Communications Lab

Lab #1

Experiment

Digital communication signals, like many signals used in digital electronics, often toggle back and forth between two voltage levels. We could use four levels, or eight, or more, but we often don't. Even if we do use more levels, we often try to understand what the circuits are doing by first looking at what happens with just two levels. It seems a bit old fashioned to talk about More code telegraph signals, but it turns out the digital signals we use today have pretty much the same format as the ones used in these old systems - on and off.

In this lab we will use 0 and 5 volts for the two voltage levels, just because those are the voltages used by the particular type of digital logic circuits we will be using (TTL). However, there is nothing special about these levels. You will see similar results if you were to use 0 and 3.3 volts, +1 and -1 volts, or any two voltage levels which are compatible with the electronics you are using.

We could generate digital signals by using the output of a computer, or even bring in an old telegraph key and let people tap out Morse code. But computer outputs like a USB connection turn out to use rather complicated signals now. And human generated Morse code varies a lot from one person to the next. We would like a simple and predictable source of digital signals. For this lab, our digital data source will be a function generator, set to generate a square wave. The square wave will need to toggle back and forth between 0 and 5 volts. This will simulate a computer sending a data signal that looks like an alternating series of zeros and ones, or 010101010101010...

The length of time required to send one bit of information is called the bit time, or Tb. It doesn't make any difference if the bit is a zero or one, either way the output needs to stay at a constant voltage for Tb seconds. It is often convenient to talk about the data rate, or fb, in units of bits per second. The bit time and data rate are reciprocals of each other, fb = 1/Tb.

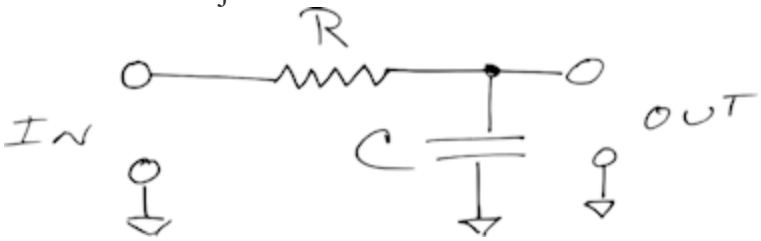
PART I. Set the square wave generator so it produces an alternating sequence of data bits, using a data rate of 1 kilo bit per second (1 kbps). Be careful, you do not want to set the square wave generator to 1 kHz. Display the simulated data signal on an oscilloscope.

Confirm that it toggles between 0 volts and 5 volts. Make sure it stays at each level for 1/1000 seconds. Note any errors, or imperfections, in the signal. Are the voltage levels exactly correct? Is the time it spends at each level correct? When the voltage is supposed to be exactly 0 volts, is it at exactly constant for the entire time, or does it drift up and down? Same thing for when it is at exactly 5 volts. If it does vary, does it do this in a predictable way or an unpredictable way? A predictable way would be something it did the same for every bit that was at that level. An unpredictable way is sometimes called noise, and is different for every bit. The best you can, describe what the predictable and unpredictable parts look like. The square wave is supposed to instantly jump between the two output levels. Does it do this, or does it take some time to transition? If it takes time, how much time?

PART II. In circuits classes we tell you the voltage is identical at all points of a wire, or what they may have called a circuit node. That is a good approximation when wires are short and voltages are constant or change slowly. However long wires that carry rapidly changing voltages don't work that way. The small amount of resistance, capacitance and inductance of the wire, or cable, causes the output voltage to be distorted. It does not look exactly the same as the voltage at the input. This can be a big problem in communication systems, and may ultimately limit how many bits per second can be sent through the cable.

Classes that talk about transmission lines will teach you about how to measure and model cables accurately. We don't want to have miles and miles of cable in the lab to show you what long cables do, so you are going to build a cable simulator. In this lab we are going to use an unrealistically simple model of a cable, but it will still distort the signals and let us do measurements to see how to handle this kind of problem.

The simulator is just an R/C circuit that looks like this



Use a 1 kohm resistor and a 0.1 uF capacitor. Put your simulated 1 kbps digital signal on the input. Use the scope to make sure the input still looks as good as it did in part I. Then use the scope to look at the output. Describe how it changed from what you saw in part I.

PART III. Vary the data rate in the previous part, and describe how the distortion of the digital signal changes with the data rate.

Part IV. People who build communication receivers, like those that build logic gates, recognize that their input signals might not look ideal. Even if they would like to see only 0 volts and 5 volts on the input, in practice it often falls in-between these two levels. So they typically build circuits to allow for some error. For example, they may say that anything below 1 volt will be treated as if it were 0 volts, and anything above 4 volts will be treated the same as if it were 5 volts. The middle range from 1 to 4 volts they would consider an invalid state, and expect you to move through that state quickly, and at a time that nobody was looking carefully at your signal. If a communications receiver does look at a signal in the 1 to 4 volt range, it will say an error occurred, and that bit of information is lost.

People who design communication systems will give you some length of time in which you can be in the prohibited/error zone. For example they might say that you have 10% of a bit time (Tb) to get from one logic level to the other. But then the other 90% of the bit time you must stay solidly in the high or low states, to allow them to determine which bit of information you are trying to transmit.

Using your cable simulator and square wave generator, find the highest possible data rate you can transmit through the simulated cable - and still meet the specification that you spend no more than 10% of a bit time between 1 and 4 volts.

Part V. Logic gates will usually have a hard specification on exactly what voltages you need to use for a logic high and low level. However communication receivers are often unsure exactly what voltages you are using. Sometimes they receive a powerful signal with a large voltage swing, other times they get a weak signal and the voltage difference between a logic high and low is very small. To handle this problem, the receiver can measure the peak voltage it receives, which we will call Vmax. The receiver will then decide a logic high level is anything that comes with 90% of Vmax and a logic low level is anything below 10% of Vmax.

Use the highest data rate you found in Part IV. Measure Vmax, and determine how long it takes for the signal to transition from 10% of Vmax to 90% of Vmax. Express this time as a fraction of Tb. Is it the same for a low-to-high transition as it is for a high-to-low transition?

Part VI. In the previous problem you probably measured Vmax by looking at the signal on an oscilloscope. While that's OK for a laboratory setting, it's not going to work so well in practice. We don't want to have a scope and person at every communication receiver to

measure Vmax. Try to design a circuit that will put out constant voltage at the level of Vmax - something that you could read on a conventional DC volt meter. Test your circuit to see how well it works. Things you should look for include: Did Vmax change when you connected your circuit to measure it? If so, by how much? Is the output of your Vmax measuring circuit constant, or does it fluctuate up and down? If it fluctuates, by how much does it change? Can your circuit tolerate Vmax becoming larger or smaller - as would happen if you change the output voltage of the square wave generator? If it does tolerate changes, how long does it take to respond to a change? There is no one right answer, or one optimal design, for this part. I suggest you consider using an R/C low pass filter, described in circuit analysis classes. If you have had a class on electronics, you may find some op-amp circuits useful