PHYS-3360/ AEP-3630
LABORATORY MANUAL
2010-2011

E. J. Kirkland and R. Littauer
School of Applied and Engineering Physics and
Physics Department
Cornell University

July 28, 2010

Contents

Chapter 1

INTRODUCTION

This chapter is rather long but some of the material should be review. If you desire more background reading try some of the books listed at the end in the bibliography such as Brophy[1], Diefenderfer[2], Fortney[3], Hambley[4], Scherz[9], or Simpson[11].

1.1 Voltage Signals and Ground

Electrical signals such as voltage, current or charge can be used to encode information or control various devices. Electronic circuits are used to process these electronic signals. Appropriate *transducers* at the input and output of the circuit translate the information into and out of an electrical form. Chapter 10 briefly discusses transducers, but this chapter begins by discussing the handling of information within the circuits.

Perhaps because of the relative ease with which a voltage can be measured, there is a tendency to think of electrical signals primarily as voltages. However, currents are equally important. For some devices, such as the transistor, current plays the dominant role. You should therefore be prepared to consider signals either in the voltage domain or in the current domain, even though currents are rarely observe directly.

The word "voltage" normally means *voltage with respect to ground.* Ground plays an important role as a circuit reference potential. In many instruments the metal case is connected to the power-line ground via a three-prong power plug. Often the case is also connected permanently to one of the input or output terminals of the instrument. This is true for the oscilloscope and for many signal generators. The grounded terminal is called the *low, common, or neutral terminal.* The other is the *hot* or *live* terminal.

Most circuits are operated with a specified point connected to ground. Other, ungrounded circuits are *floating*, and are susceptible to all sorts of electrical *pickup* or unwanted signals introduced through proximity to other electrical apparatus.

Ground is also important for *shielding.* Sensitive circuits are enclosed in grounded metal cases to shield them from electric fields which could cause interference. External wires that carry low-level signals are usually shielded. Most often a shielded connection takes the form of a *coaxial cable*, complete with coax connectors at its ends. A coaxial cable has a single wire at its center that carries the signal.

This wire is surrounded by an insulator and both are inside an outer conductor shield, which is usually held at ground potential. The whole assembly is usually encased in plastic for easy handling. Plugging in such a cable takes care of the signal and the ground connection at the same time.

When part of the system is a "breadboard" circuit, grounding becomes less automatic. The circuit boards have coaxial connectors (type BNC) mounted on them to facilitate the transition between well engineered coax systems and the temporary circuit on our breadboard (see figure 1.1). When you use these connectors (for example, to bring in the signal from a generator) be sure to join both the signal (inner) and ground (outer) points of the connector to your circuit appropriately.

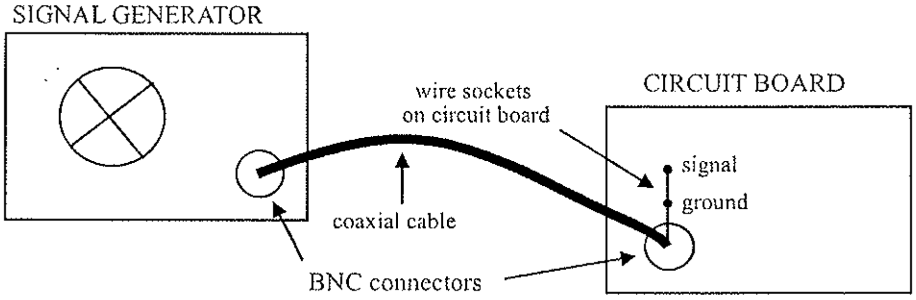


Figure 1.1: Equipment layout.

The signals to be displayed on an oscilloscope are almost always picked up via one of the special probes made for the scope. These probes have short ground wires attached to them, ending in an alligator clip. Grounding the scope is best done by attaching this clip to the ground line of your circuit.

Voltages measured with respect to ground are called *unbalanced or single-ended.* They form the majority by far. On the other hand, you sometimes need to observe the *voltage difference* between two *hot* terminals (i.e., where neither terminal is at ground potential). This is called the *differential mode*, such that the information is related only to the voltage difference between pairs of terminals. If the voltage signals on the two terminals are equal but opposite, the signal is called *balanced*, *push-pull*, or *double-ended.* Interfering pickup, to the extent that it is induced equally on both sides, is subtracted out when the difference signal is properly processed.

1.2 Measuring Voltages and Rise Time

Most electronics measurements involve observing a periodic waveform (sine wave, square waves, triangle waves, etc.) on an oscilloscope (see section 1.3). The amplitude of the voltage may be measured in several ways. Three common definitions are shown on a sine wave (with period T) in figure 1.2. The peak-to-peak voltage V_{pp} is the voltage difference between the maximum and minimum value of the waveform. The peak voltage V_p is the maximum value with respect to zero. With a symmetrical waveform (as shown) $V_p = V_{pp}/2$.

The rms voltage (rms is short for root-mean-square) is closely related to the power in the signal and is a little more complicated to define. If T is the period of the waveform then the rms voltage is

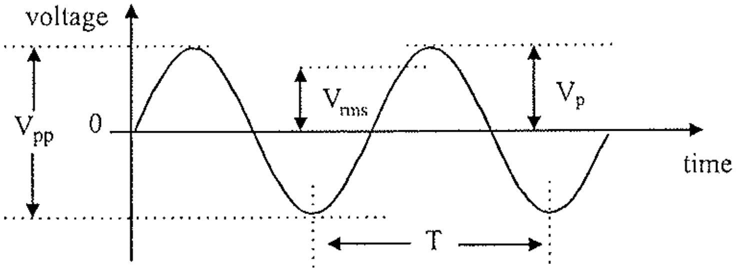


Figure 1.2: Measuring AC voltages.

mathematically defined as:

$$V_{RMS} = \sqrt{\frac{1}{T}\int_0^T |v(t)|^2 dt} \tag{1.1}$$

For the special case of sine waves (not square or triangle waves):

$$V_{RMS} = \frac{V_P}{\sqrt{2}} = \frac{V_{PP}}{2\sqrt{2}} \tag{1.2}$$

Although a square wave signal ideally has a perfect square edge that rises (and falls) in zero time, in practice all square pulses have a small *rise time* associated with them. ·Figure 1.3 shows an expanded view of a rising edge of a square wave signal, showing some non-ideal properties that are usually present in practical square wave signals. At the beginning and end of the edge there may be some small undershoot and/or overshoot and the signal may take a small time to actually rise from its initial value to its final value as illustrated in the figure.

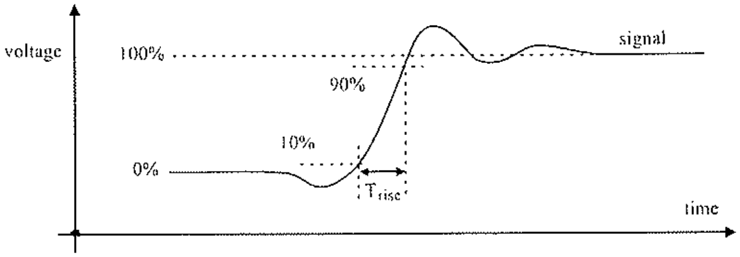


Figure 1.3: Measuring the rise time of square wave signals. This is an expanded view of the rising edge of a square pulse, showing undershoot and overshoot.

The presence of the overshoot and undershoot and the slow curve to a final value make it ambiguous as to what time the signal takes to rise. For this reason the *rise time*, $T_{rise}$is usually defined as the time the signal take to go from 10% to 90% of its total travel as drawn in the figure. These levels are defined as a percentage of the signal a long way away from the actual edge (the asymptotic values). The 10% and 90% voltage levels usually lie in the portion of the rising edge that is reasonably close to a straight line. A similar definition holds for the *fall time* , which is just the over edge (figure 1.3 drawn upside down).

1.3 Using the Oscilloscope

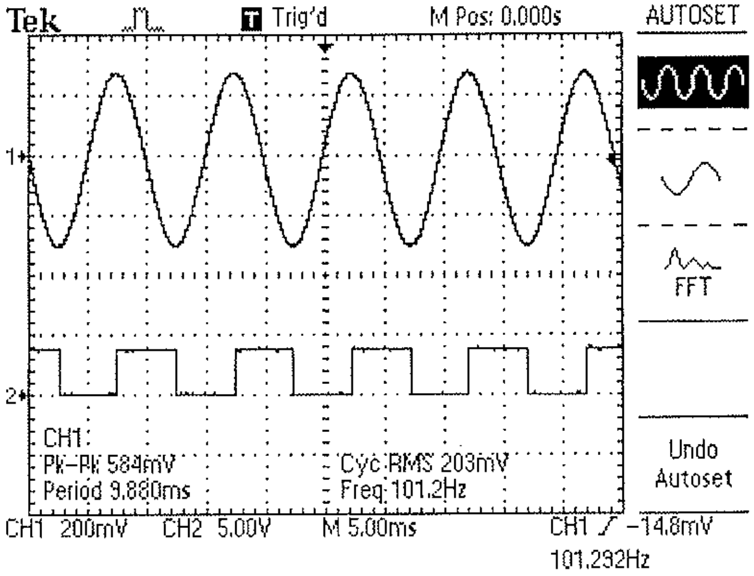
An oscilloscope (scope) is one of the primary measurement tools used in electronics. This section will go over some details of how to use the specific type of scopes that are in the lab and assumes that you have a general understanding of how a scope works. There are two types of digital sampling scope (may also be called a digital storage scope), the TDS-210 and the TDS-1002 both made by Tektronix. The TDS-1002 is a newer version of the TDS-210 and both work in a similar manner although the buttons on the TDS-210 work slightly differently than on the TDS-1002. This discussion will be for the TDS-1002 but you should be able to convert to the TDS-210 in a straight forward manner (ask you lab instructor for assistance if you have trouble).

Older analog scopes work by sweeping an electron beam across a phosphor screen in the horizontal direction (with a deflection in the vertical direction from the input signal) and the traces forms a graph of the time dependence of the input signal. Many analog scopes are still in use and you should be prepared to use them if necessary. Most new scopes use digital sampling techniques (as in the scopes here). This allows for a variety of other useful features in the scope although they are conceptually very similar to an analog scope in function. It takes a little bit of effort to learn how to use a digital scope but you will be rewarded with several useful measurement features. A digital scope will sample (or measure) the input voltage at a sequence of discrete times (using an internal ADC or analog to digital converter, as in chapter 12) and plot the results on the screen in graph form. The horizontal axis is time and the vertical axis is voltage. There is a small dedicated computer (microprocessor) inside the scope that controls its functions.

The TDS-1002 scope is a dual channel scope which means that it can measure two different voltages at the same time and plot both on the screen on the same time axis. The TDS-1002 has a bandwidth of 60 MHz and can sample both input channels at 1 GigaSamples per second (2 GS/s total). One very useful button on this scope is the "AUTO-SET" button. When you push this button with one or both input signals connected (two connectors labeled CH-1 and CH-2 for channel 1 and channel 2) then the scope will automatically find the correct settings for the time and voltage scales to display the input signals. You may have to change the finally setting but this usually gives you at least a good starting point to work from. Figure 1.4 shows the screen on the scope after performing an AUTO-SET function. The top curve (sine wave) is from the channel-1 (CH-1) input (indicated by a small '1' on the left of the curve) and the bottom curve (square wave) is from the channel-2 (CH-2) input (indicated by a small '2' on the left of the curve). At the bottom of the screen the scope tells you what sensitivity (in volts per division) it has used for each channel (may be different for each channel) and what time scale it has used (same for both channels). If you press the 'MEASURE' button it will also give you a variety of other measurements in digital form (i.e. numbers) for the input signals. Note that there are several levels of menu options to navigate through.

1.3.1 Vertical

The left hand section of the scope panel (just to the right of the screen) is devoted to the vertical-sensitivity controls. There are two input channels labeled CH-1 and CH-2. Normally both signals will be displayed, but one of them may occasionally be turned off. Each channel has an amplifier which



CH1:
Pk-Pk 584mV
Period 9.880ms
Cyc RMS 203mV
Freq 101.2Hz

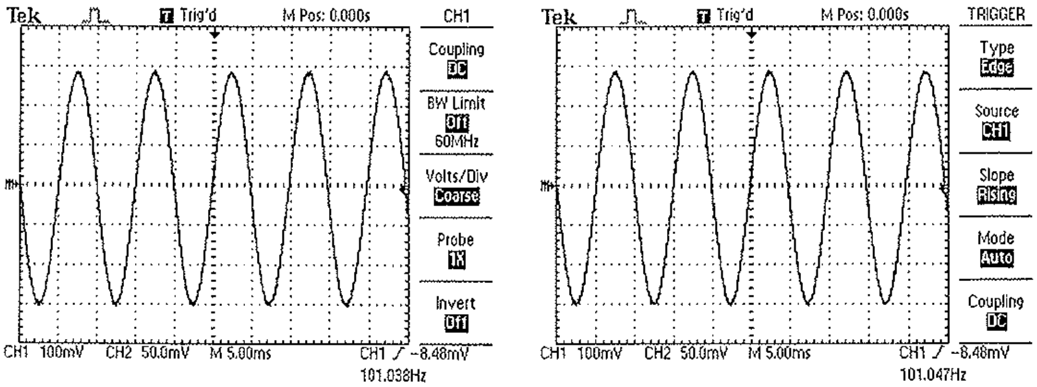
CH1 200mV CH2 5.00V M 5.00ms CH1 ∕ −14.8mV
101.232Hz

Figure 1.4: Screen view of the Tektronix TDS-1002 digital storage scope using the AUTO-SET feature. The CH-1 input was connected to the sine wave output of a function generator and CH-2 was connected to the TTL/CMOS or sync output.

produces an adjustable vertical-deflection factor, controlled by the knob labeled VOLTS/DIV meaning volts per division on the screen. You may rotate the knob to change the vertical sensitivity. The knob labeled POSITION controls the vertical position of each curve on the screen.

Each channel also has a menu button, which calls up a menu on the screen for each channel when it is pushed. If you push it twice then the channel is turned off (can be turned on again with another push of the button). The screen during set-up of the vertical amplifier section is shown on the left of figure 1.5. The button just to the right of each item controls the setting for that item. Normally the COUPLING should be set to DC to provide the best response (settings of AC and GROUND also possible). The BW-Limit should be set to OFF to give the largest possible bandwidth. The Probe item should be set to 1X for a direct connection and 10X when using a 10X probe (see later discussion). This affects the numerical readout.

With two independent vertical amplifier channels it is also possible to display both channels simultaneously. Typically, the input signal is on one channel and the output signal on the other. When both signals are displayed simultaneously you can observe their relationship in time (see figure 1.4). The scope can also measure differential signals using the both input channel, one for each input. Set the two channels to the same deflection sensitivity, then push the 'MATH MENU' button and set the operation to '-' or 'CH1-CH2'. The voltage difference between CH1 and CH2 will be displayed on the screen. In practice each individual signal voltage, with respect to ground, must not be so large as to overload the scope input. Should this happen, you get distortion in the display and you may damage the scope.



M Pos: 0.000s
CH1
Coupling DC
BW Limit Off 60MHz
Volts/Div Coarse
Probe 1X
Invert Off
CH1 100mV CH2 50.0mV M 5.00ms CH1 ∕ −8.48mV
101.038Hz

M Pos: 0.000s
TRIGGER
Type Edge
Source CH1
Slope Rising
Mode Auto
Coupling DC
CH1 100mV CH2 50.0mV M 5.00ms CH1 ∕ −8.48mV
101.047Hz

Figure 1.5: Screen of the Tektronix TDS-1002 digital storage scope during set-up. Left: set-up of the vertical amplifier section. Right: set-up of the trigger section.

1.3.2 Horizontal

The horizontal direction on the screen usually represents time and the sampling rate (in time) is selected by means of the SEC/DIV (seconds per division on the screen) knob. The current setting is displayed as a number on the bottom of the screen. The small knob labeled POSITION adjusts the horizontal position of the graph on the screen. Usually the horizontal axis is time and the vertical axis is voltage. However if you push the button labeled DISPLAY then you can also select the FORMAT=XY mode in which CH-1 is the x-axis (horizontal) and CH-2 is the y-axis (vertical). This will be used sometimes, but you normally should leave the format set to YT mode for voltage (vertical) versus time (horizontal).

1.3.3 Triggering

The scope has a *triggered* time base, which requires a specific triggering signal to initiate each sweep across the screen on the scope. On receipt of a trigger, the voltage is sampled and displayed on the screen. The vertical position on the screen tracks the voltage at the input to the scope. When properly triggered the waveform displayed on the scope appears to remain fixed in time even though the voltage is changing rapidly in reality. If the signal is repetitive and you use a trigger that bears a fixed time relationship to it (in the simplest case, a trigger derived from the signal itself), successive cycles of the display will follow a repetitive pattern that appears as a fixed pattern on the screen. The signals need not recur at regular intervals. Even if they arrive at random, successive traces of the display fall on top of each other with a proper trigger.

Selection of a proper trigger is very important. You proceed by three steps: (1) select the trigger SOURCE; (2) select the sweep MODE; and (3) determine at what point of the triggering signal you wish the actual trigger to be generated. Push the button labeled TRIG MENU (in the trigger section on the right hand side). Then you should see a screen display something like that shown on the right hand side of figure 1.5. The TYPE should be 'EDGE' for almost everything that you will need to do.

1. The SOURCE (of triggering) can be (internal) CH1 or CH2, which means that the same signal is

used for triggering as is being displayed in the Y direction (CH1 or CH2). EXT trigger uses the signal applied to the BNC connector labeled EXT TRIG in the bottom right. AC LINE source provides a 60Hz trigger signal derived from the power line (to test whether a signal comes from the power lines). The COUPLING lever selects AC or DC coupling for the trigger signal. DC coupling is usually the most appropriate. HF means high frequency and LF means low frequency.

2. There are two modes of triggering. AUTO or automatic mode is the default. In AUTO mode the time base improvises its own trigger whenever it has not been supplied with an official trigger for a certain time. This helps to keep a visible trace on the screen at all times even when there is no trigger. In the NORMAL mode there is only a trace when there is a valid trigger. You should usually use the AUTO trigger mode however you may need to switch to NORMAL whenever the occasional self triggering of the sweep interferes with your display.

3. Having selected the trigger source and sweep mode, you must still decide whether you want the trigger to occur when your signal moves in the positive or negative direction. The button labeled SLOPE chooses either a rising edge or a falling edge. The TRIG LEVEL knob (on the right) selects the voltage threshold at which the trigger fires. Beware, if the level is set to a voltage the triggering signal doesn't reach, no trigger occurs.

At this point you might play with the scope to familiarize yourself with its controls. Use it to display the output of the function generator in each of its three forms: sine waves, triangular waves, and square waves.

Exp. 1.1 Connect the 50Ω OUT socket of the function generator via coax to the channel-1 input of the scope. Adjust the generator to sine waves with maximum output (full clockwise). Set the scope VOLTS/DIV appropriately and obtain a good display of the signal. Vary the generator frequency over its complete range and follow suit by readjusting the scope's sweep speed appropriately.

Adjust the trigger LEVEL and SLOPE controls and observe their effect; also vary the amplitude of the signal to verify that the trigger is generated at a determined voltage threshold.

Measure the peak-to-peak (p-p) amplitude of the signal when the generator is set for maximum output. Measure the period and frequency of the signal when the generator dial is set to exactly 1 kHz.

With the generator delivering square waves, set up the scope to observe the rising edge of the waveform in detail (sketch it in your lab book). Measure the time it takes to go from 10% to 90% of its total voltage swing (this is the conventionally defined rise time of the signal). You may use the scopes cursors to measure the rise time and then compare to the built in rise time measurement in the scope. Does the rise time vary with frequency? **end**

For convenience and accuracy in using the scope it is important to learn the various ways of triggering the sweep. Simple use of the internal trigger mode is adequate for viewing one waveform, but other methods are often better.

When you want to see several related signals in succession, it is a nuisance to have to readjust the trigger controls each time the signal is changed. It is easier to let one of the signals do the triggering all the time. Then you are free to "roam" over the circuit with the scope input and view all the signals in turn. In this way you can also see the time relationship between the waveforms, since the sweep is triggered in a fixed manner. This would of course be impossible if each signal were allowed to provide its own trigger.

One way of obtaining a fixed trigger when your circuit is driven by a signal generator is to connect the generator to the *external trigger* input of the scope. Most generators provide a separate output for this purpose. On the function generator it is labeled SYNC OUT (or TTL/CMOS on the BK generators). This synchronizing signal is a constant amplitude square-wave, regardless of the waveform or amplitude delivered at the normal signal output. The edges of the SYNC signal coincide with those of the square-wave output, or with the peaks of the sine wave or triangle. The relative phase of the signal and the SYNC may be different on different function generators but it should remain constant in time for a given function generator. These sharp edges offer good time definition for stable triggering.

Another way of obtaining a fixed trigger is to feed the desired trigger signal into channel 1 of the scope, setting the trigger SOURCE to internal CH1. The scope will display both channels 1 and 2, but it will take its trigger *from channel 1 only.* Channel 2 can therefore be used to roam over the circuit. The simultaneous display of the "trigger" and "signal" waveforms shows their time relationship very beautifully.

> **Exp. 1.2** Connect the SYNC OUT (sometimes labeled TTL/CMOS) of the function generator to channel 1, and the signal output to channel 2. Then set up a dual-trace display as just described, and inspect the time relationship between the SYNC and signal waveforms for all three types of output. Verify that you can alter the signal amplitude without affecting the triggering of the scope. **end**

1.3.4 Input Probes

Connecting the scope to the signal source (via coax) is easy. More commonly the challenge is to pick up a signal from a relatively inaccessible point somewhere in the midst of a circuit. Here you have mechanical problems (inadvertently pulling the components out of their sockets, or shorting various wires). Also, you must take care of the grounding. Use of open clip leads is not recommended. Their mechanics are uncertain and they don't connect to the scope's BNC input without an adapter. Besides, they tend to pick up stray signals. The preferred method is to use a specialized scope probe. These probes attenuate the signal by a factor of 10 in transmitting it to the scope, but in return they present a modestly small capacitance (about 3 pF) and a high resistance (10 MΩ) to the circuit being measured. This *attenuation probe* also compensates for the frequency dependence due to the stray capacitance at the input of the scope. You should try to use the attenuated probe whenever possible to accurately measure fast signals. These probes also need to be adjusted for waveform fidelity, a skill you should pick up at once.

> **Exp. 1.3** Plug a probe (set to 10X) into channel 1 of the scope. Touch the tip to the PROBE COMP terminal (sometimes labeled CAL or PROBE ADJ on other scopes) (located in

the bottom right corner on the TDS-1002) and obtain a display. (You may clip the probes tip onto this terminal.) Adjust the sweep speed so that four or five complete cycles of the waveform are visible. (The PROBE COMP outputs delivers a square wave of about 5 V_{p-p} amplitude, at a frequency of about 1 kHz.)

There is an adjustable capacitor in the probe which must be set so that the square-wave signal is displayed correctly. To vary the capacitor, adjust the small screw near one end of the cable. The diagram in figure 1.6 shows the effects of this adjustment. Repeat the procedure with a second probe connected to channel 2. Leave both probes plugged in to the scope, they will be used constantly. NOTE: you will need to repeat this calibration procedure every time you get started int the lab. $\boxed{\text{end}}$

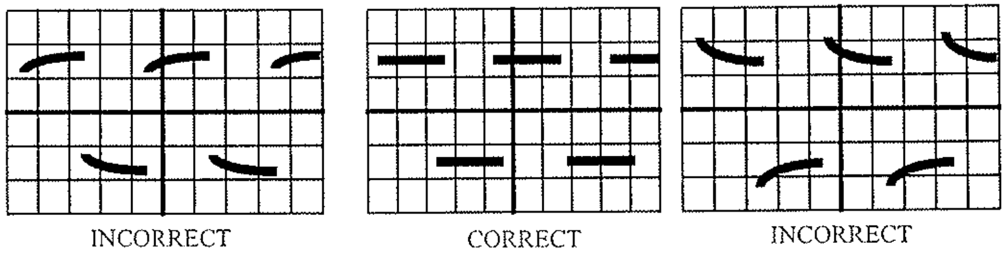


Figure 1.6: Scope probe adjust

1.4 Ideal Voltage and Current Sources

The function generator used above is an example of a voltage source. A battery is another example of a voltage source. An *ideal voltage source* produces an output voltage V_o which is independent of the load current I that is drawn from it. An ideal voltage source is shown schematically in figure 1.7 and is represented on a graph of V versus I by a horizontal straight line.

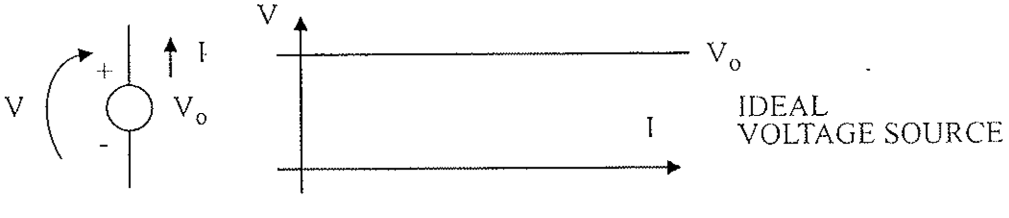


Figure 1.7: Ideal voltage source.

In a practical realization of a voltage source such as the function generator this model is only true if the physical limits of the source are not exceeded. Specifically you must keep the maximum current within the allowed range of the function generator (the straight line does not extend to infinity in practice).

Analogous to a voltage source is an ideal current source (see figure 1.8). The I-V curve of a current source is a nearly vertical straight line. A current source forces a constant current I_o through the load (connect between its two terminals), with the help of whatever voltage V is needed. In practice, you must of course stay within the range of validity of the current-source concept. If you extrapolate the curve far enough up, you must ultimately get into a region where the current falls well below I_o.

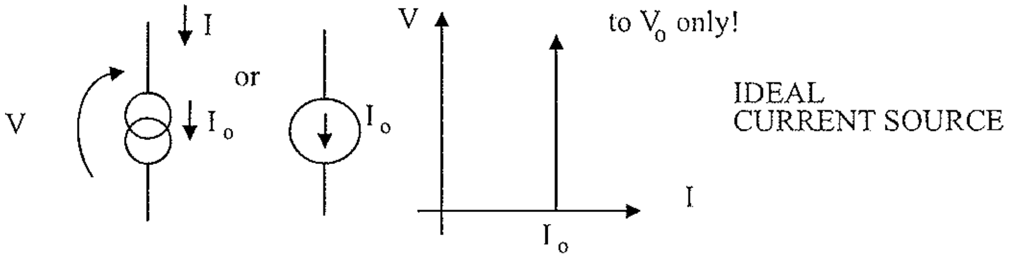


Figure 1.8: Ideal current source.

The simplest way to make a current source is to start with a voltage source much greater than the maximum variation to be expected in the load voltage, and then insert an appropriate resistor in series. In practice, this may be cumbersome and a transistor version of a current source if often used (see chp. 6).

1.5 Resistors

One of the most basic circuit components is the resistor. The current I (in Amps) flowing through the resistor, is related to the voltage V (in volts) across the resistor by Ohm's Law where R is the resistance of the resistor in Ohms.

$$V = IR \tag{1.3}$$

The schematic symbol for a resistor with the current I and voltage V is shown in figure 1.9.

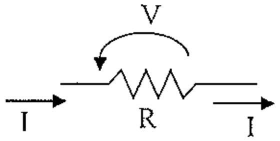


Figure 1.9: Schematic symbol for a resistor.

Resistors may be combined in various way. When put together in series, as in figure 1.10, the total resistance (i.e. the resistance between points A and B in the example) is just the sum of the resistances.

$$R_{AB} = R_1 + R_2 + \cdots + R_N \tag{1.4}$$

However when resistors are combined in parallel, as in figure 1.11, the total resistance (between points A and B) is the inverse of the sum of their inverse values.

$$\frac{1}{R_{AB}} = \frac{1}{R_1} + \frac{1}{R_2} + \cdots + \frac{1}{R_N} \tag{1.5}$$

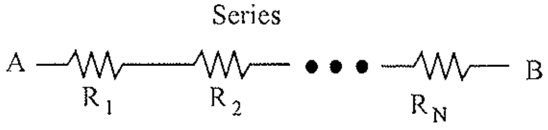


Figure 1.10: Series combination of resistors.

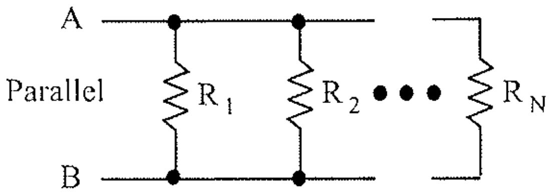


Figure 1.11: Parallel combination of resistors.

The combination of two resistors in parallel occurs frequently and is sometimes written as $R_1//R_2$ which has a total resistance of:

$$R_1//R_2 = \frac{R_1 R_2}{R_1 + R_2} \tag{1.6}$$

A common configuration of two resistors and one voltage source called a voltage divider is shown in figure 1.12. This configuration gets its name from the fact the output voltage is "divided down" from the input voltage (i.e. the output voltage is always lower than the input voltage). Using Ohm's law the output voltage is:

$$V_{OUT} = \frac{R_2}{R_1 + R_2} V_{IN} \tag{1.7}$$

assuming that no current is drawn by what is attached to the point labeled V_{OUT}.

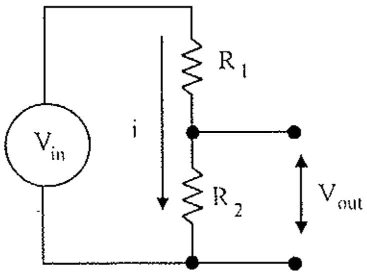


Figure 1.12: Voltage divider.

This configuration (the voltage divider) occurs so frequently that it is best to remember this result. Solving large new circuits is frequently a matter of recognizing parts of the circuit that are in a known configuration. The voltage divider is one of these small parts that you should learn to recognize.

You will frequently use one or more resistors in your circuits. Most resistors are marked by bands of different colors. The colors represent the decimal digits of the value of the resistor (in Ohms) as shown in figure 1.13.

0	Black
1	Brown
2	Red
3	Orange
4	Yellow
5	Green
6	Blue
7	Violet
8	Grey
9	White

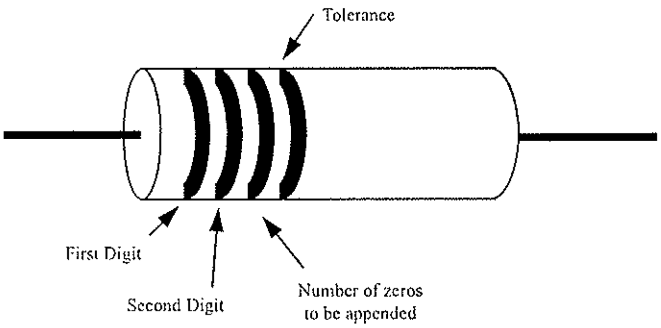


Figure 1.13: Resistor color code

This code is quite universally used and you should ultimately learn it by heart. Note that the range from 2 to 7 follows the rainbow, with the "extrapolations" on either end made plausibly by eye.

The resistor's value in ohms is marked in three bands, as indicated above. (When the third band is black, no zeros are to be added to the first two digits. When this band is gold a decimal point is placed between these two digits, and when it is silver the decimal point precedes them.) For example a resistor labeled Yellow-Violet-Red has a value of 4.7 KOhms. A fourth band, if present, indicates the accuracy (or *tolerance*) of the a resistor. Silver stands for ±10%, and gold for ±5%. A fifth band is sometimes used to encode the reliability rating of the resistor.

Commercial resistors are usually produced in standard values which are spaced so that ±10% or ±5% tolerance ranges just "touch." Thus the 10% series has the significant digits 1, 1.2, 1.5, 1.8, 2.2, 2.7, 3.3, 3.9, 4.7, 5.6, 6.8, 8.2, 10, and the 5% series, which is rarer, also fills in the gaps with extra values. It makes no sense to design a circuit around nominal resistor values, which do not coincide with these preferred discrete levels. (However, special high-precision resistors are available with many other nominal values. These will not be used in this lab.)

Most of the small resistors in our lab are rated to dissipate at most 1/4W or 0.25W of power. Larger sizes (up to 2 W in composition resistors, much higher in wire-wound types) are available, but these will not be needed here.

1.6 Kirchhoff's Laws

Kirchhoff formulated two important laws for voltages and currents. They are known as Kirchhoff's Voltage Law (KVL) and Kirchhoff's Current Law (KCL). The first says that *the sum of the voltages around a loop must be zero.* You need to be careful to measure the voltage in the same direction on each component around the loop. This law applied to a single voltage source and three resistors is shown in figure 1.14.

$$V_0 + V_1 + V_2 + V_3 = 0 \tag{1.8}$$

The second law (KCL) says that *the sum of the currents flowing into a node is zero.* (This is another way of saying that charge is neither created or destroyed at a node.) KCL applied to the junction of

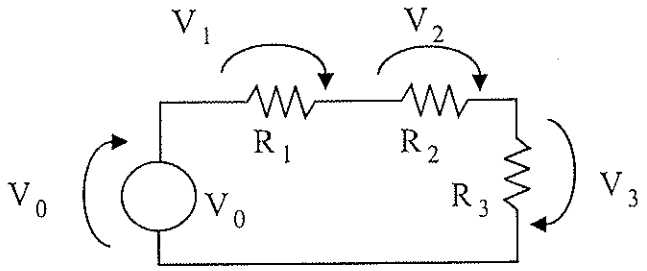


Figure 1.14: Kirchoff's voltage law.

four wires (i.e. a node) is (see figure 1.15):

$$I_1 + I_2 + I_3 + I_4 = 0 \tag{1.9}$$

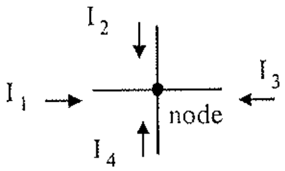


Figure 1.15: Kirchoff's current law.

1.7 Superposition

Another handy circuit solving tool is superposition. This rule says that the total effect of a large collection of voltage and current source in a linear circuit (i.e. resistors) can be found by adding the contribution of each source one at a time. To find the contribution for source i at some position x in a circuit:

1. replace all other voltage sources by a short circuit and all other current sources by an open circuit

2. solve for the voltage at point x due to source i

3. repeat for all other sources, adding each contribution to the total

This procedure is illustrated by applying it to the simple circuit shown in figure 1.16. This circuit has two voltage sources V_1 and V_2 and two resistors R_1 and R_2. Try to find the total voltage at point V_x with respect to ground at the bottom of the circuit.

First short out voltage source V_2 as shown on the left in figure 1.17 and solve for this contribution to the total voltage V_X. Note that this is just a voltage divider.

$$V_{X1} = \frac{R_2}{R_1 + R_2} V_1 \tag{1.10}$$

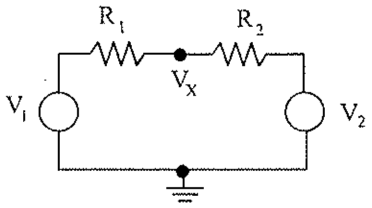


Figure 1.16: Circuit to illustrate superposition.

Next short out voltage source V_1 and solve for the contribution from V_2. This is again just a voltage divider.

$$V_{X2} = \frac{R_1}{R_1 + R_2} V_2 \tag{1.11}$$

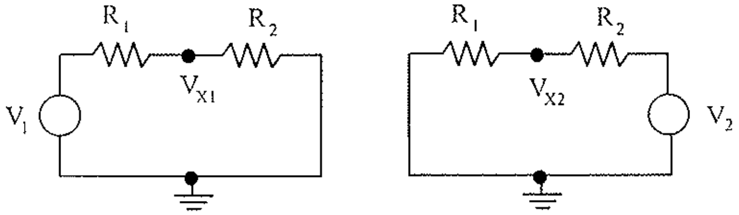


Figure 1.17: Superposition procedure applied to figure 1.16.

The total voltage is then:

$$V_X = V_{X1} + V_{X2} = \frac{R_2}{R_1 + R_2} V_1 + \frac{R_1}{R_1 + R_2} V_2 = \frac{R_2 V_1 + R_1 V_2}{R_1 + R_2} \tag{1.12}$$

You should verify that you get the same answer by solving this circuit without using superposition. The concept of superposition applies to all linear systems, and can be very useful in understanding what complicated circuits are doing.

1.8 Capacitors

A capacitor consists of two parallel conducting plates separated by an insulator. The basic definition of capacitance, C, states that the charge Q on a capacitor is proportional to the voltage V across it.

$$C = \frac{Q}{V} \tag{1.13}$$

The schematic symbol for a capacitor is shown in figure 1.18

When capacitors are combined in series (see figure 1.19) the inverse of the total capacitance is the sum of the inverses values.

$$\frac{1}{C_{AB}} = \frac{1}{C_1} + \frac{1}{C_2} + \cdots + \frac{1}{C_N} \tag{1.14}$$

When capacitors are arranged in parallel (see figure 1.20) the total capacitance is the sum of the individual capacitances.

$$C_{AB} = C_1 + C_2 + \cdots + C_N \tag{1.15}$$

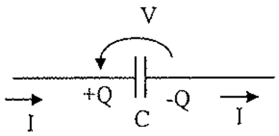


Figure 1.18: Schematic symbol for a capacitor.

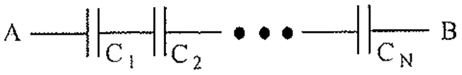


Figure 1.19: Series combination of capacitors.

This is just the opposite of the rules for combining resistances.

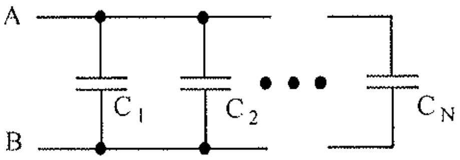


Figure 1.20: Parallel combination of capacitors.

Resistors behave essentially the same for AC signals and DC signal. However capacitors behave very differently for DC and AC signals. For perfect DC currents the capacitor looks like an open circuit and no DC current flows. However when the voltage across the capacitor changes a charge flows into and/or out of the capacitor. When the charge Q on the capacitor changes, a current I flows into one terminal of the capacitor and out of the other, with $I = dQ/dt$. Combining these equations gives:

$$\frac{dV}{dt} = \frac{I}{C} \quad \text{or} \quad I = C\frac{dV}{dt} \tag{1.16}$$

The current "through" a capacitor is proportional to the *rate of change* of the voltage across it. In any real, physical circuit the current always remains finite. It follows that dV/dt must remain finite, too. This amounts to saying that *the voltage across a capacitor cannot change instantaneously* (i.e. ΔV across the capacitor must go to zero as Δt goes to zero). In fact, for short enough intervals of time, the capacitor acts momentarily like a battery of fixed voltage.

There are several types of capacitors for various applications. The rolled plastic- film type has the most nearly ideal properties, but its bulk makes it somewhat awkward for general use. Ceramic disk capacitors are much more compact, but usually their value is subject to greater error. Ceramic disk capacitors usually have the best frequency response. Electrolytic capacitors are more compact still. Electrolytic capacitors are required for capacitance's upward of 1 μF. They are "polarized". The applied voltage must conform to the polarity markings (i.e. there will be a + or - near one end of the capacitor), or the capacitor will be destroyed quickly. They also have a significant leakage current and may have a rather large error in their marked value.

Capacitor markings follow various patterns. As a general rule, values marked as decimal fractions are in units of μF (i.e. microFarads), and values greater than 1.0 are in units of pF. You have to use your judgment a little. A large electrolytic capacitor marked +22 is obviously 22 μF, not 22 pF. (Also, the + is on the side to be connected to the more positive voltage.) The numerical value is sometimes followed by a letter which indicates the tolerance of the capacitor, as follows:

J	± 5%	R	+30%/-20%
K	± 10%	S	+50%/-20%
L	± 15%	Z	+80%/-20%
M	± 20%		

However, you have to be careful in reading these values because the manufacturers may include other markings using letters to indicate other properties of the capacitor. Capacitors (particularly electrolytic capacitors) are notorious for having a large error in their value.

The circuit diagrams used here follow the same convention. Fractional values are μF, values above 1 are pF unless a polarity symbol indicates that an electrolytic capacitor is intended.

New capacitors follow a different but increasingly common labeling scheme. The capacitor is marked with a series of three numbers. The last digit is the exponent and the first two digits are the mantissa. The number is in units of pfd. For example "103" means 10×10^3 pfd which is the same as 10 nfd or 0.01 μfd.

1.9 Circuit Assembly

Starting in the next few sections you will need to connect various components (resistors, capacitors, etc.) into a circuit. The quickest way to assembly a circuit for testing is with what is called a solderless breadboard. In the lab there is a breadboard called a Digi-Designer which has some special features (i.e. switches, connectors, etc.) appropriate for digital circuitry but can be used for analog circuits as well. The components (resistors, capacitors, etc.) are plugged into a white plastic matrix which houses recessed metal sockets (i.e. circles in the diagram below). These sockets are connected internally in the groupings shown in figure 1.21.

Components (resistors, capacitors, integrated circuits, etc.) and wires (not larger than #22 gauge) can be plugged directly into the holes. They become connected to everything else sharing the same group of contacts. The components and wires are not held very firmly, so you must be careful to avoid pulling them out as you handle the breadboard. An oscilloscope probe can be clipped directly to a component's wire as it emerges from the plastic matrix. More conservatively, you can plug a short extra wire into the group of holes and connect the scope probes to this wire. Do NOT plug the scope probe directly into the breadboard, because this will damage the breadboard.

The long horizontal groupings are intended for "bus lines" for distributing ground and the power-supply voltages to various places on the board. Note that each horizontal line has a break at the center, which you must bridge with a wire jumper if you wish your bus line to extend clear across the board.

The boards have binding posts on them, which can be used for the power-supply connections. They also have coaxial "BNC" connectors for signals to be brought in or taken out. Both the outer shield

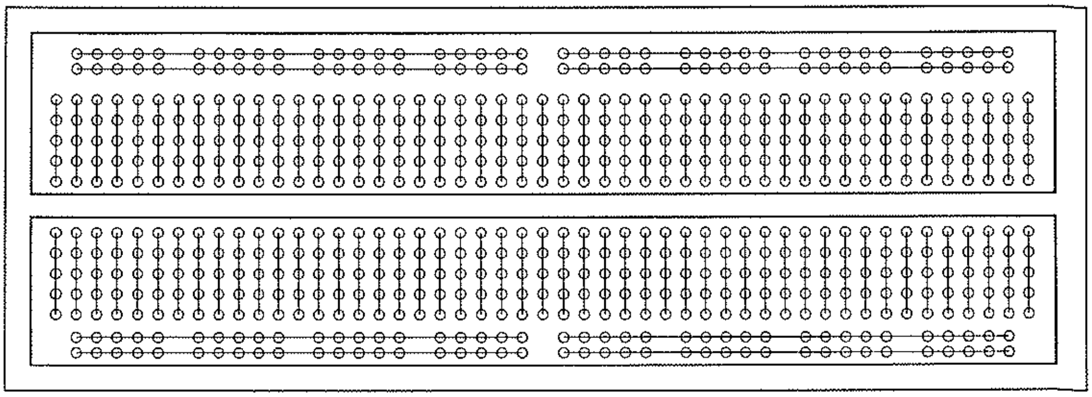


Figure 1.21: Solderless Breadboard. Each circle is a socket for a wire or component and the lines indicate how the sockets are connected on the bottom side of the board.

and the inner pin of the coax connector should be connected to your circuit. The shield is usually tied to ground (on the Digi-Designer the ground for the BNC connectors is already connected to the ground terminal internally).

Although these breadboards are particularly easy to use for quick testing of various circuits they are by no means permanent or durable. If you wish to use a circuit on a permanent basis you need to solder it on a printed circuit board. Unfortunately this can be very time consuming (but necessary).

1.10 Waveshaping by R-C Circuits

The two R-C combinations shown in figure 1.22 are of fundamental importance. They will change the shape of pulse passing through them and they will also act as frequency selective filters for sinusoidal AC signals (this gives rise to their names). In the next two sections you will investigate their effects on square pulses and later in chapter 3 you will investigate their effects on sinusoidal signals.

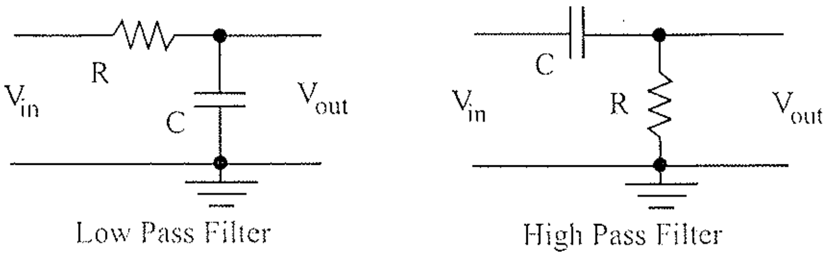


Figure 1.22: R-C filters.

If a square pulse of amplitude A and length T is applied to each filter the results are shown in figure 1.23. The low pass filter will allow a low speed signal (i.e. $T >> RC$) to pass nearly unchanged and the high pass filter will allow a high speed signal (i.e. $T << RC$) to pass nearly unchanged. It is interesting to note that the DC level or time averaged value (i.e. the integral of V_{out} over all time) of the output of the high pass filter is zero independent of the DC level of the input. This is because the capacitor does not pass the DC level of the input and the DC level (or time average) of the output is determined by the resistor and is thus zero. The trailing edge of the pulse is a small negative value but may last for a long time so that the integrated value (over time) of the output is zero.

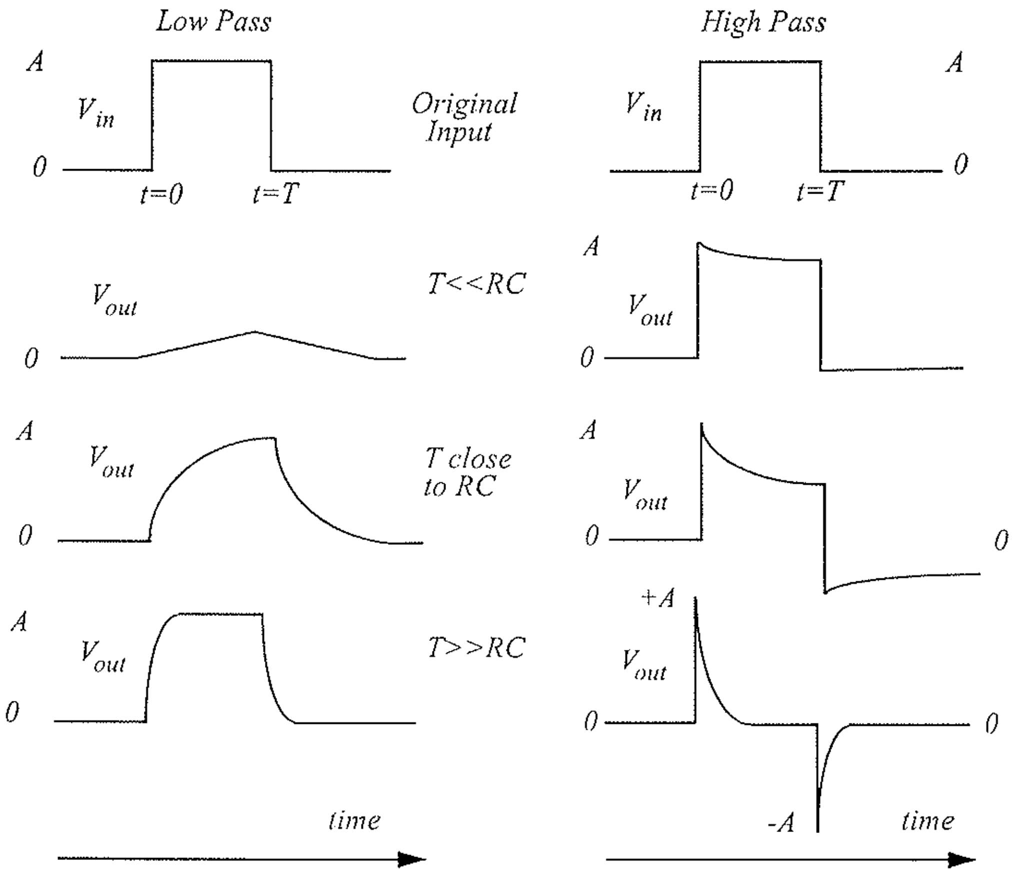


Figure 1.23: Effects of low and high pass filters on a rectangular pulse.

In the experiments to follow a square wave will be applied as the input instead of a single pulse to obtain a repetitive waveform that can be viewed on the oscilloscope. A square wave is really just a sequence of square pulses so the results will be similar to what is shown here. However, when the pulses come close together one pulse may not have time to decay completely before the next pulse arrives. This may shift the apparent DC (or average) level of the output.

1.11 High-Pass Filter

First lets analyze the high pass filter in more detail. Consider a single positive voltage step of the input waveform (see figure 1.24). Remember that *the voltage across a capacitor cannot change instantaneously*. During the short time in which the input signal changes its level, the voltage across the capacitor remains constant and therefore the whole of the step is transmitted directly to the output.

Afterwards the capacitor has time to gain or lose charge, and thus to let the output voltage relax back toward zero.

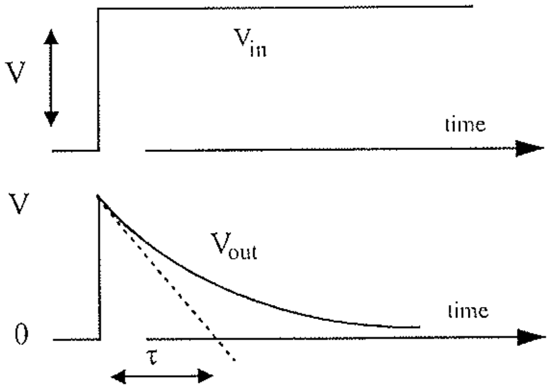


Figure 1.24: Effects of high pass filters on a rising edge.

Assuming that v_{out}=0 just before the step arrives, then just after the step $v_{out} = V$, where V is the height of the step. This momentarily puts a voltage V across R, starting a current $i = V/R$ through it. This current flows *out* of C, producing a rate of change of voltage.

$$\frac{dv}{dt} = -\frac{i}{C} = -\frac{V}{RC} \quad \text{[initially]} \tag{1.17}$$

This initial rate of change of voltage is shown by the broken line, in figure 1.24. It is such that, *if it were maintained constant*, the capacitor voltage would change by an amount equal to V in a time given by RC. This time is called the *time constant*, τ, of the R-C circuit: $\tau = RC$.

But the initial rate of discharge is not in fact maintained constant. As v_{out} decays, the current through R decreases in proportion ($i = v_{out}/R$). The rate of discharge thus decreases to:

$$dv/dt = -v_{out}/RC. \tag{1.18}$$

In other words, the rate of change of v_{out} is proportional to the amount *still left to go*.

Any curve whose slope is proportional to the amount "still left to go" is an *exponential*. A more formal analysis of our circuit shows that:

$$v_{out} = Ve^{-t/\tau} \tag{1.19}$$

(To show that this equation satisfies the conditions for our circuit calculate dv_{out}/dt and see that it equals $-v_{out}/RC$.)

Exp. 1.4 Assemble the circuit shown in figure 1.25. (If this is the first time you use a solderless breadboard, please refer to the description in section 1.9.) Bring the signal from the function generator to one of the coaxial BNC connectors on the board, and from there via a short wire to the input side of the capacitor. With a probe on channel 1, observe V_{in}. Set the generator to deliver square waves at 1 kHz, with a peak-to-peak amplitude of exactly 10 V.

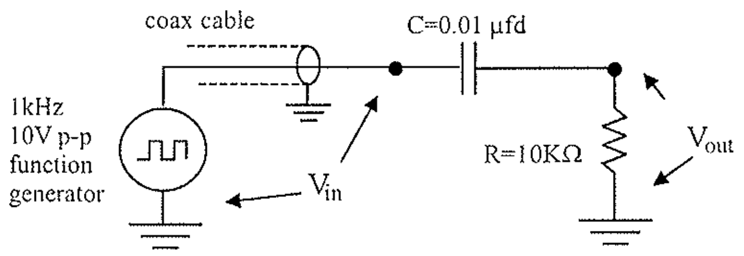


Figure 1.25: High pass filter experiment.

Trigger the scope from the channel 1 only. With another probe on channel 2, observe V_{out} and sketch both in your notebook. Calculate the time constant of your R-C circuit, and find by what factor v_{out} should decay in a time equal to one half-period of your square wave. **end**

The waveform you observed with a square-wave input (instead of a single step) is just a string of exponentially decaying pulses of alternating polarity, corresponding to the positive and negative steps of the square wave.

If the half-period of the square wave is made shorter, v_{out} may not have time to decay to almost zero. In that case the waveform for the subsequent step starts not from zero but from the level left by the previous step. When the half-period is very short, v_{out} has time for only a very small amount of decay. This then occurs at an almost constant slope given by $-v_{out}/RC$.

Exp. 1.5 Study and sketch the waveforms of v_{out}, with v_{in} a square wave of 10 V_{pp} amplitude, and f = 100 Hz, 1 kHz, 10 kHz, and 100 kHz (again using the circuit shown in figure 1.25). Be sure to record the actual voltage levels of v_{out} with respect to zero (ground) potential. The input for both channels must be set to DC coupling. You can check the zero level of the trace by moving this selector momentarily to the GND position.

For f= 10 kHz measure the value of dv_{out}/dt on the slightly sloped top of the waveform. You can do this more accurately if you increase the scope sensitivity and move most of the waveform off the screen with the VERTICAL POSITION control. Verify that $dv_{out}/dt = -v_{out}/RC$. **end**

In the action of this circuit, the average voltage of the input voltage doesn't actually matter. Only the height of the step, V, is important. This step is transmitted by C. A steady voltage at the input would merely cause a corresponding steady-state charge to accumulate on C, whereafter v_{out} would again be zero. In fact, v_{out} always tends to return toward ground potential. On the waveform you have just observed you will notice that v_{out} was *symmetrical about ground* (even if v_{in} was not). When clear, individual peaks were seen, their amplitudes were a symmetrical, ±10 V. When the square wave was transmitted without much change of shape, its upper and lower levels were ±5V. In other words: *the high-pass R-C circuit does not transmit the DC level of the input signal.*

If blocking the DC part of the signal is the intent, then the circuit is called AC coupling. The scope has the facility for inserting such an AC coupling in series with its input. You place the selector below the VOLTS/DIV switch in the AC position. This puts a capacitor in series with the input socket. The arrangement is useful when you wish to view a small change of signal level superimposed on a much larger steady voltage. (Unless you specifically want to do this, you should leave the input selector in the DC position because AC coupling also introduces a frequency dependent gain in the scope.)

1.12 Low-Pass Filter

To visualize the action of the low pass filter (as in figure 1.26), note first of all that v_{out} will always try to become equal to v_{in} because the capacitor can charge or discharge through R. However, this charging is slow. Only if v_{in} is maintained at a particular level long enough, does v_{out} become equal to v_{in}. Now suppose v_{in} has been kept at zero voltage for a long time, and then a step of height V suddenly occurs. The capacitor's voltage cannot change abruptly so v_{out} hovers near zero initially. Thus v_{in} appears entirely as a voltage drop across R (V_R). This starts a current $i = v_{in}$ /R flowing, and C then charges at such a rate that

$$\frac{dv}{dt} = \frac{i}{C} = \frac{v_{in}}{RC} \quad \text{[initially]} \tag{1.20}$$

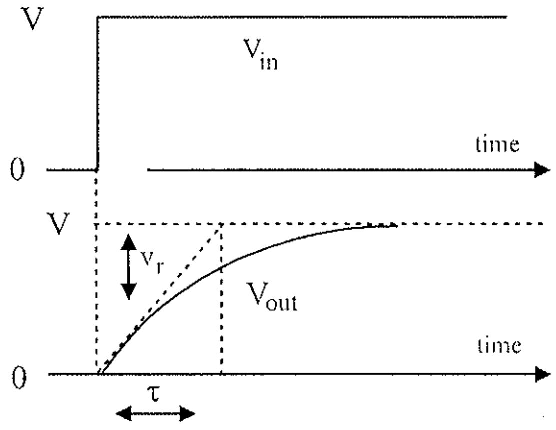


Figure 1.26: Low pass filter.

If this initial rate were maintained, v_{out} would reach its final level, v_{in}, in a time RC. In fact the charging rate decreases as V_R gets smaller. The slope of v_{out} is again proportional to the amount still left to go, and the curve is an exponential. The only difference from our previous case is that v_{out} rises toward a new level instead of decaying from a new level back toward zero voltage. The complete solution of this equation is:

$$\begin{aligned} V_{CAP}(t) &= V_F + (V_I - V_F)e^{-t/(RC)} & (1.21)\\ &= V_I + (V_F - V_I)[1 - e^{-t/(RC)}] & (1.22) \end{aligned}$$

where $V_{cap}(t) = V_{out}(t)$ is the voltage across the capacitor, V_I is the initial voltage on the capacitor and V_F is the final voltage on the capacitor (in this case V_I=0 and $V_F = V_{in}$). This will be called the RC charging equation and is valid when the capacitor is charged through a resistor to a constant voltage.

Exp. 1.6 Assemble the circuit shown in figure 1.27 and observe v_{in} and v_{out}, with the function generator set to deliver square waves at a frequency of 1 kHz. end

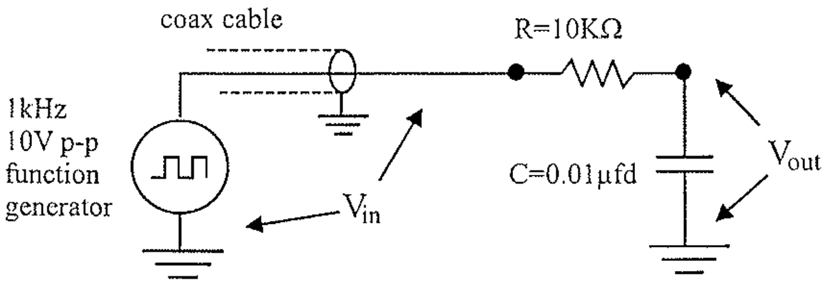


Figure 1.27: Low pass filter experiment.

In the low-pass filter (unlike the high-pass filter) the level of v_{out} is tied directly to the level of v_{in}. For example, if v_{in} starts from V_1 and switches suddenly to $V_2 = (V_1 + V)$, then (assuming enough time has elapsed) v_{out} starts from V_1 also and approaches V_2 with an exponentially decaying "distance to go" (as shown in figure 1.28). *The low-pass filter transmits the DC level of the input signal.* On the other hand, it does not transmit the sharp steps.

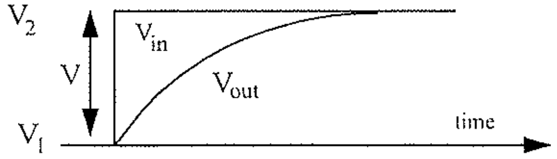


Figure 1.28: Low pass filter with offset voltage.

When v_{in} is a square-wave signal, v_{out} consists of a series of such rounded waveforms strung together. If the half-period of the square-wave is long compared to the time constant, RC, the capacitor has time to charge very nearly to the peak voltages of the generator. Using the capacitor charging equation it is possible to show that the 10% to 90% rise time of the output is:

$$T_{rise} = 2.2RC \quad (\text{only if } \; T >> RC) \tag{1.23}$$

when the amplitude is allowed to reach its full value. If the half-period is shorter, the capacitor does not charge fully, and the waveform is truncated. Each section starts from whatever level the previous piece had reached when the new step occurred. If the half-period becomes very much shorter than RC, v_{out} remains almost stationary. It doesn't get a chance to move much either way. However, *on average*, the amount by which it drifts up must equal the amount by which it drifts down. Hence the equilibrium level for v_{out} is the *time average* of v_{in} (also called the DC component).

Exp. 1.7 Sketch the waveform for v_{out} when the frequency of the function generator is 100 Hz, 1 kHz, 10 kHz, and 100 kHz (again using the circuit shown in figure 1.27). In each case measure the peak-to-peak amplitude of v_{out}. v_{in} should be set to deliver squares waves at 10 V_{p-p}.

With f= 100 kHz, measure the voltages at the top and bottom of the input square wave, and measure the average DC level that v_{out} stayes near. Verify that v_{out} is the time average of v_{in} Make a sketch of the waveform for v_R in this situation, and view this directly by displaying $v_{in} - v_{out}$ on the scope in the DIFFERENCE mode. Deduce the magnitude of the current which periodically charges and discharges C. From this find the instantaneous dv_{out}/dt, and calculate the peak-to-peak voltage swing of v_{out} that occurs in the time of one half-cycle of the square wave. $\boxed{\text{end}}$

A basic limitation on the performance of many circuits at high frequencies is set by the presence of parasitic (i.e. not wanted but unavoidable) capacitance associated with cables and connecting wires. In particular any coaxial cable has a well-defined amount of capacitance per unit length of cable.

$\boxed{\text{Exp. 1.8}}$ Rearrange the low-pass filter circuit of exercise [I.7] so that C is provided by a 3 to 5 ft length of coax cable with one end connected to a BNC jack on your circuit board. Set R= 100K. From the measured risetime of v_{out} with a suitable square wave drive, v_{in}, determine the value of capacitance/ft for your cable. Record the cable designation code stamped in the cable covering (probably RG58A/U or RG62/U). $\boxed{\text{end}}$

1.13 Source Impedance (or Resistance)

In the case of AC networks containing capacitance or inductance, the concept of internal or source resistance must be generalized to *internal impedance* (or *source impedance*). Although a full treatment of AC signals will wait for later, the usual practice of saying *impedance* instead of *resistance* will be followed here.

Given a source of emf (i.e. a voltage source), it is useful to know the actual voltage which exists across its terminals *under the loading conditions imposed by the circuit.* This applies to such sources as batteries, power supplies, or signal generators. For example think of the dimming of your car lights when you crank the engine. The battery voltage is reduced by the heavy load of the starter motor.

Practical sources are not ideal voltage sources. Their output voltage almost always decreases as the load current goes up. The decrease can be attributed to an effective *internal resistance*, R_i as shown in figure 1.29. Current drawn from the source flows through R_i, producing a voltage drop across it. The voltage available to the outside world is thus reduced. If, in this representation, the ideal source "hidden" behind R_i delivers a constant voltage v_0, the terminal voltage of the complete assembly is:

$$v = v_0 - iR_i \tag{1.24}$$

The I-V characteristic for this is a straight line falling below the horizontal plot of an ideal source. Many practical sources have I-V characteristics that are at least approximately straight. These can be described by their v_0 and R_i. For sources with nonlinear behavior it is often possible to fix attention on a restricted region of operation in which the linear approximation is good.

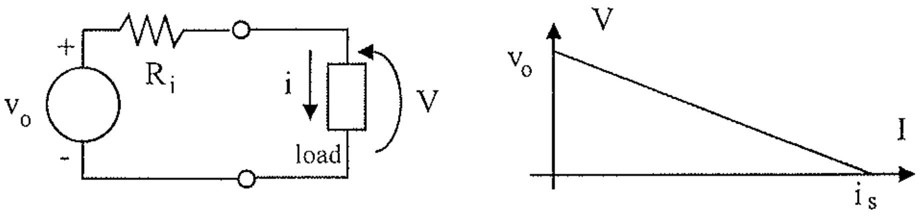


Figure 1.29: Source impedance.

To find v_0, first measure the voltage at the source terminals when zero load current is drawn, i.e., *an open circuit.* v_0 is the open-circuit voltage. At the other extreme, if the terminal voltage is forced down to zero (with a *short-circuit load*), a certain short-circuit current, i_s , flows. Evidently $i_s = v_0/R_i$. Thus, by measuring or calculating v_0 and i_s, R_i can be determined as (see figure 1.30):

$$R_i = v_0/i_s \tag{1.25}$$

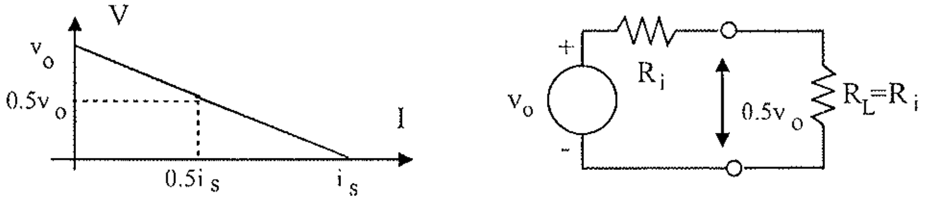


Figure 1.30: Effect of source impedance with an equal value of load resistance.

Many devices may be damaged by shorting their outputs and currents are more difficult to measure than voltages, so i_s is usually not determine directly. Instead, attach a finite load to the source and observe by how much the output voltage is depressed. The sketch shown in figure 1.30 illustrates the particular case where the output voltage is halved, which occurs when $R_L = R_i$. Some sources would not tolerate such heavy loading. Either they are damaged, or at least they cease to operate in their linear region. For such sources only relatively small changes of output voltage can be produced, and the measurements must be made with correspondingly greater accuracy.

Exp. 1.9 Measure the source impedance of the function generator by connecting a load resistor across it. Set the function generator to deliver about 1 V_{PP} at about 1 kHz. Find a value which reduces the function generators output voltage by approximately one-half, and from this calculate R_i.
OPTIONAL: Any resistor that significantly reduces the output voltage can be used to calculate the internal impedance using the voltage divider equation. Make a table of voltages for several resistors values and calculate the internal impedance for each resistor and compare these results.
end

Output devices (like signal generators) have an *output impedance* or resistance R_{out} and devices designed to sense an input signal (like voltmeters) have an *input impedance* or resistance R_{in}. Usually these devices are connected together through two terminals (using wires) with one side grounded as in figure 1.31 (grounding is not always necessary). To transfer the maximum power between devices requires $R_{in} = R_{out}$.

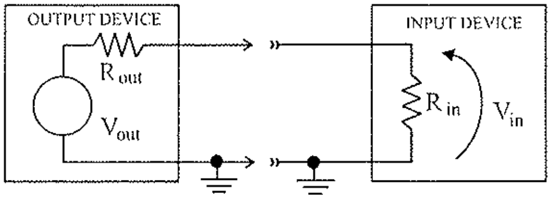


Figure 1.31: Connection between two terminal devices with input impedance R_{in} and output impedance R_{out}.

1.14 Thevenin's Theorem

Suppose now that, instead of a single voltage source with some internal resistance, you have a complicated network consisting of several circuit elements and several sources. Often it is possible to break off a section of this circuit by separating it at just *two* terminals. The piece of circuit is then a *two-terminal network*. For this network an I-V characteristic can be plotted just as before. If the network contains only *linear* elements (i.e. resistors and voltage and current sources), the I-V curve will also be a straight line. This is a remarkable result. However complicated the network, provided its elements are linear the overall result is a straight line! (A linear combination of linear components also has to be linear.) Such a line is defined by any two points on it, for example by v_0 and i_s. Thus there is a simple *equivalent circuit* for the network, consisting of a voltage source v_o and an impedance $R_i = v_0/i_s$ in series with it. This is known as the Thevenin equivalent circuit. Thevenin's theorem states that:

> *Any combination of linear circuit elements (i.e. resistors voltage sources and current sources) ending in two terminals may be replaced by a single voltage source and a single resistor.*

As far as the external behavior seen from outside the two terminals is concerned, the Thevenin equivalent and the network itself are indistinguishable. Replacing the network (in your mind) by its equivalent circuit can greatly simplify the analysis. More importantly, it sets up a way of visualizing the action of the network which is highly productive.

As an example let's consider the performance of a simple *resistive voltage divider*. In this very common circuit an input voltage v_{in} is applied across two resistors in series, and an output v_{out} is taken across the second resistor alone as in figure 1.32.

v_{out} is connected to subsequent circuitry, which draws some load current i_L from the divider. This load current changes v_{out}, which can be attributed to the finite source impedance of the divider. To construct the Thevenin equivalent of the divider, break off the part of the circuit enclosed by the dotted line on the diagram and ending in the two terminals labeled A and B. This is the two-terminal network.

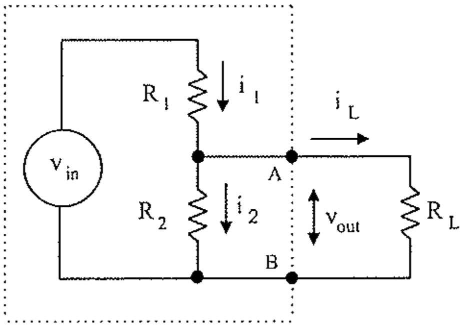


Figure 1.32: Thevenin analysis of a voltage divider.

It includes the source of v_{in} and the voltage divider. The "outside world" is represented by the load R_L (refer to figure 1.33).

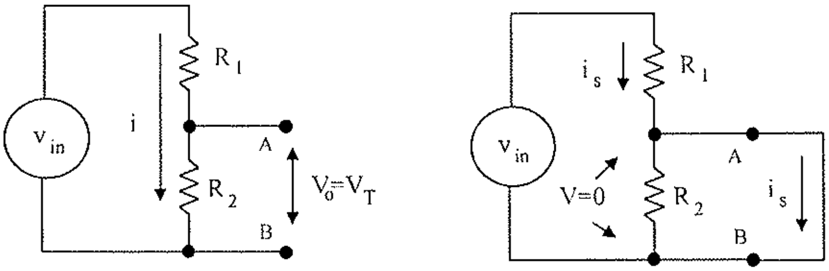


Figure 1.33: Thevenin analysis (continued).

Now let's determine the Thevenin equivalent circuit of the voltage divider by finding its open-circuit output voltage, v_0, and its short-circuit output current, i_s. The equivalent Thevenin voltage and resistance are labeled V_T and R_T respectively (instead of v_0 and R_i). The two calculations are illustrated in below. (To find i_s, it is important to realize that no current flows through R_2. All current is diverted away by the short-circuit load.)

$$i = \frac{v_{in}}{R_1 + R_2} \tag{1.26}$$

$$i_s = \frac{v_{in}}{R_1} \tag{1.27}$$

$$V_0 = V_T = iR_2 = \frac{R_2}{R_1 + R_2} V_{in} \tag{1.28}$$

From v_0 and i_s:

$$R_T = \frac{V_T}{I_S} = \frac{R_1 R_2}{R_1 + R_2} = R_1 // R_2 \tag{1.29}$$

This is the effective resistance of R_1 and R_2 connected in parallel. (The shorthand notation $R_1//R_2$ represents the parallel resistance.) The complete Thevenin equivalent is sketched in figure 1.34.

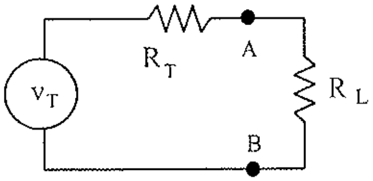


Figure 1.34: Thevenin equivalent circuit of figure 1.32.

To show there has been no sleight-of-hand in this rapid-fire derivation, go through a direct calculation for V_{out} as a function of I_L (cf. figure 1.32):

$$\begin{aligned} v_{out} &= v_{in} - v_{R1} = v_{in} - i_1 R_1 = v_{in} - (i_L + i_2)R_1 & (1.30) \\ &= v_{in} - (i_L + \frac{v_{out}}{R_2})R_1 & (1.31) \end{aligned}$$

Solving for v_{out}:

$$v_{out} = (\frac{R_2}{R_1 + R_2})v_{in} - (\frac{R_1 R_2}{R_1 + R_2})i_L \tag{1.32}$$

The first term of this result corresponds to V_T, while the coefficient of the second term, which is equal to $R_1 // R_2$, corresponds to R_T.

There is considerable insight to be gained by using the equivalent-circuit approach. It summarizes the results not only of a particular network calculation, but of all calculations involving different load impedances R_L. We would normally obtain numerical values for V_T and R_T immediately, and then use these values in further design steps. The relatively complicated algebra is thus reduced to "building-block" proportions. For example, you can tell at a glance what load resistance R_L would just halve the output voltage of the divider. It is equal to R_T. Similarly, you can predict the effect of a capacitance connected as load across the divider. This produces an equivalent low-pass R-C circuit, with an exponential time constant given by $R_T C_L$.

Beyond its practical benefits the Thevenin-equivalent picture creates in our minds the concept of *source impedance*, which leads to great progress in circuit visualization. Much of circuit design can be viewed as the problem of connecting a load to a source. Each of these can of course be a complicated network. The interaction of the load and the source is the fundamental problem and is characterized by the source and load impedances (i.e., by how each of the two "partners" relates voltage to current). Of course, once connected, the two partners are forced to make an accommodation about voltage and current!

There are three cases to consider:

1. $R_T << R$: The source impedance may be taken as negligible compared to that of the load. This is a good voltage source, which dictates its voltage V_T to the load, regardless of what (small) load current is drawn. The load in this situation is said to be voltage controlled.

2. $R_T >> R_L$: Now the load impedance is taken as negligible, and the current flowing in the circuit is determined almost entirely by R_T. This current is very close to $i_s = V_T/R_T$, the short-circuit current of the source. In other words, R_L acts roughly like a short circuit. The source, delivering

an almost constant is, is now called a current source, and it forces this current though the load regardless of what (small) voltage may develop. The load is current controlled.

3. $R_T \sim R_L$: Neither of these simplifying extremes is applicable. You are forced to work through the problem in greater detail, taking into account the values of both R_T and R_L.

So far R_T has been calculated only from V_T and i_s. This is often the best way. However, you can also proceed as follows. Ask yourself, what is the impedance you see when you *"look" into the two terminals of the network*? Because of the presence of internal sources within the network, this impedance cannot be determined simply by taking the ratio of output voltage to output current. To avoid this problem, consider what would happen if the internal sources were somehow disabled. The linearity of the network ensures that the output impedance is the same, regardless of the particular voltages of these sources. In particular, you can set these voltages to zero. If you do this, and then apply a voltage across the network's terminals from some external source, the ratio of v to i is indeed the network's internal impedance. Setting a source to zero is the same thing as replacing it with a *fixed zero voltage* (with a *short circuit*). So the prescription for finding the impedance looking into the pair of terminals is: first disable all internal voltage sources by replacing them with short circuits and disable all internal current sources by replacing them with an open circuit, then find the impedance in the usual way.

To illustrate the procedure, apply it to the voltage divider sketched in figure 1.35. It is clear that the impedance seen looking into the output terminals is $R_1//R_2$, just as was found above. This approach to finding R_T is instructive. Also, it's the only way that works if by any chance V_T= 0 (and therefore i_s = 0 also).

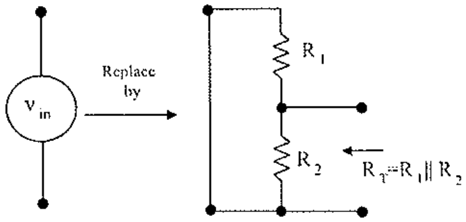


Figure 1.35: Thevenin equivalent circuit.

1.15 Norton 's Theorem

As mentioned above, a high-impedance source driving a low-impedance load is called a *current source*. Remember, though, that the idea of voltage or current sources derives from the circuit context (the load impedance) in which these sources live. The same network could well be a voltage source for one load and a current source for another.

The current-source concept gives us another method for drawing an equivalent circuit for a two-terminal network. It is the Norton equivalent shown at right. The double circle stands for an idealized current source delivering a fixed is. Norton's theorem states that:

Any combination of linear circuit elements (i.e. resistors voltage sources and current sources) ending in two terminals may be replaced by a single current source and a single resistor.

This differs from Thevenin's theorem only in the substitution of a current source for a voltage source (see figure 1.36, 1.37).

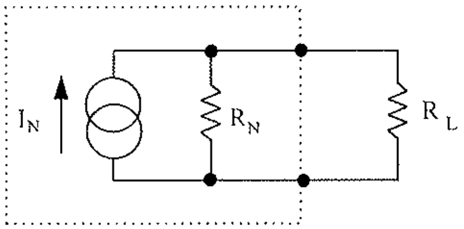


Figure 1.36: Norton equivalent circuit.

By itself, such an ideal current source has an infinitely large internal impedance. Then, by connecting R_N across it, this produces a combination which duplicates the internal impedance of the network to be simulated.

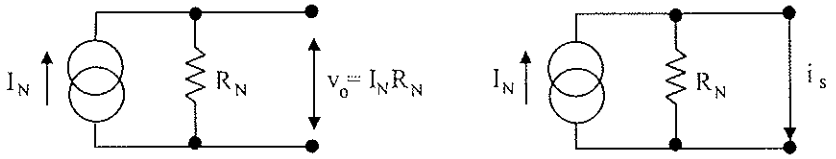


Figure 1.37: Norton equivalent circuit.

To convince yourself that the Norton equivalent circuit correctly represents the network, argue as follows. The circuit consists of linear elements, so its I-V curve is a straight line. Check two points on this line. With an open circuit, $v_o = I_N R_N$, as shown at left (in figure 1.37). With a short circuit, $i_s = I_N$, because R_N cannot take any current in competition with the external short circuit. These two points check correctly.

If the Thevenin and Norton equivalents are indistinguishable to the outside world, why have two different models? It's a matter of convenience in making approximations, where they are justified. If the practical configuration makes the source nearly a voltage source, it's easiest to use the Thevenin circuit and maybe even ignore the presence of R_T altogether. If the source is nearly a current source, the Norton circuit, again with R_N possibly omitted, is more appropriate. Many practical devices, such as transistors, have outputs that can be considered current sources in many circuits which would be well represented by a Norton equivalent circuit.

1.16 Source Impedance and Capacitive Loads

Suppose you have a signal source with a high source resistance, and you wish to connect it to a load with a large capacitance to ground. The signal must then effectively pass through a low-pass R-C circuit, which causes waveform distortion on such signals as square waves. As an example, suppose the source impedance is R_i= 10 kΩ, and the load capacitance is C_L= 10 nF. What is the exponential time constant of this combination?

If your application demands that the input square wave be delivered with a time constant of no more than 10 μs, what can you do? Assume that signal *amplitude* is of no concern–there's more than enough available. Then one possibility is to put a voltage divider across the source, with a Thevenin impedance much lower than R_i as in figure 1.38.

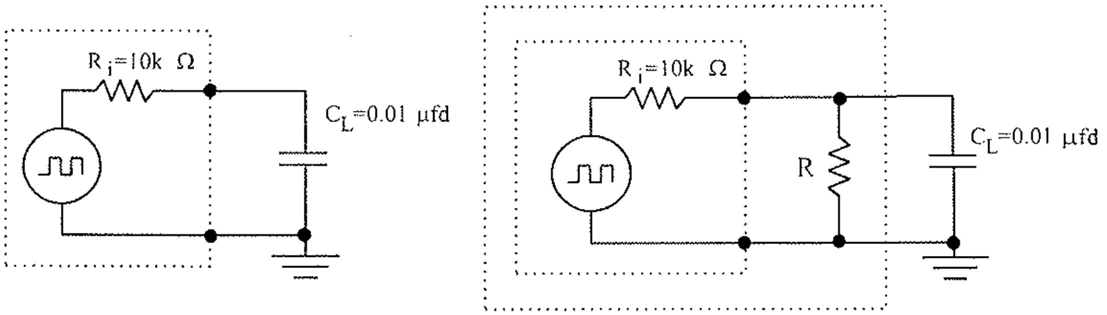


Figure 1.38: Thevenin equivalent circuit.

Exp. 1.10 Investigate this method experimentally as in figure 1.38. You can simulate the high-impedance source by connecting a 10K resistor in series with the function generator and considering this as the available signal source (inside the dashed box on the left above). Then connect C_L across the "output," and study the waveform across it when suitable square waves are used as input. Observe what happens when an additional resistor R is connected across C_L (see the circuit on the right of figure 1.38). Using Thevenin's theorem calculate a value of R that reduces the time constant of the waveform across C_L to the required 10μs? With this R in the circuit measure the rise time and amplitude of v_{out} and compare to what you expect from Thevenin's theorem. (Choose a frequency that is low enough so that the waveform has time to reach its maximum amplitude.)

Observe the initial rate of rise of the waveform across C_L with and without R. Does R affect this initial rate of rise? Sketch both waveforms and explain your observations of the initial rates of rise, and the final amplitudes of the two signals, in terms of the characteristics of the Thevenin sources with and without R. end

1.17 Practice Problems

[1] Find the currents I_1 and I_2 in the circuit shown in figure 1.39, with component values R_1 =10K, R_2 =3.9K, R_3 =4.7K, R_4 =4.7K, R_5 =6.8K, V_1 =10V, V_2 =5V, V_3 =5V. Express your answer in units

of mA. (Hint, it's much easier if you apply Thevenin's Theorem twice. Once on the left and once on the right.)

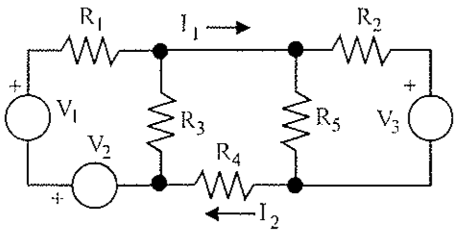


Figure 1.39: Circuit for problem [1]

[2] Show that the rise time of an RC low pass filter (with square wave input, T>>RC) is $2.2RC$ as in equation 1.22.

[3] Exp. 1.10 asks you to measure the *initial rate of rise* (i.e. the derivative at t=0 and NOT the rise time) of the voltage across C_L with and without R (with a square wave signal source). Derive an algebraic expression for the initial rate of rise and show that this initial slope must be the same with and without R for any value R. (Hint: Find the Thevenin equivalent of the signal source R_i and R and then find an expression for the slope at t=0.)

[4] A 1 kHz square wave with an amplitude of 2 V_{pp} is applied at the input V_{in} of the circuit shown in figure 1.40 (R_1 = R_2= 10K, C= 0.02μf, V_1= 10V).

a) What is the effective time constant of this circuit?

b) What is the minimum and maximum voltage reached at the output (V_{out})?

c) On the same time scale sketch V_{in} and V_{out}, clearly indicating their relationship in time.

(Hint: If V_1, R_1, and R_2 are replaced by their Thevenin equivalent circuit then this is just a high pass filter with an offset on the output.)

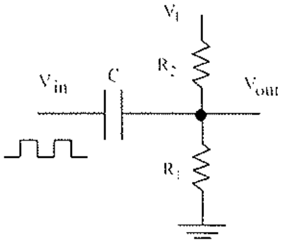


Figure 1.40: Circuit for problem [4]

[5] What is the effective time constant $\tau = RC$ of the circuit shown in figure 1.41, with component values R_1 =10K, R_2 =4.7K, R_3 =3.3K, and C =0.01μF. Express your answer in units of μSec. (Hint, it's much easier if you apply Thevenin's Theorem to everything except the capacitor.)

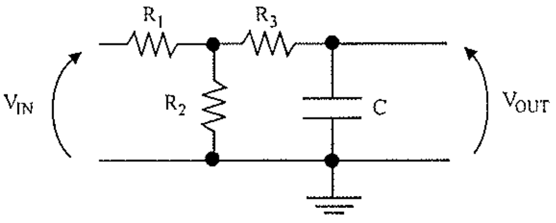


Figure 1.41: Circuit for problem [5]

[6] In the circuit shown in figure 1.42, calculate V_1 and V_2 for component values R_1 =3.3K, R_2 =4.7K, R_3 =10K, and V_{in} =10 V.

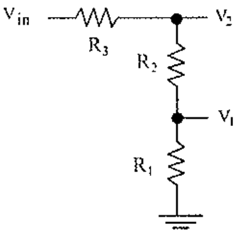


Figure 1.42: Circuit for problem [6]

Chapter 2

THE OPERATIONAL AMPLIFIER

2.1 Introduction

This chapter and chapter 4, will consider the amplification of *analog* signals which have a continuously variable amplitude directly representing the magnitude of a parameter of interest. This analog representation contrasts with *digital signals*, which are quantized signals restricted to a few discrete levels (usually two levels).

Analog signals often need to be *amplified* to increase the power in the signal. The added power is derived from a DC power supply, which in itself carries no information. It is the task of the amplifier to modulate this steady power in accordance with the information contained in the input signal. *Operational amplifiers* (or just *op-amps*) have evolved into a standard configuration available in integrated circuit (or IC) form (see figure 2.1). Besides just raising the power level, operational amplifiers can also perform certain mathematical *operations* on the signal, such as adding or subtracting two signals, differentiating or integrating with respect to time, taking the logarithm, and so on. The op-amp is a basic building block of many analog signal processing circuits. This chapter introduces the basic performance characteristics of the operational amplifiers and chapter 4 will return to the operational applications.

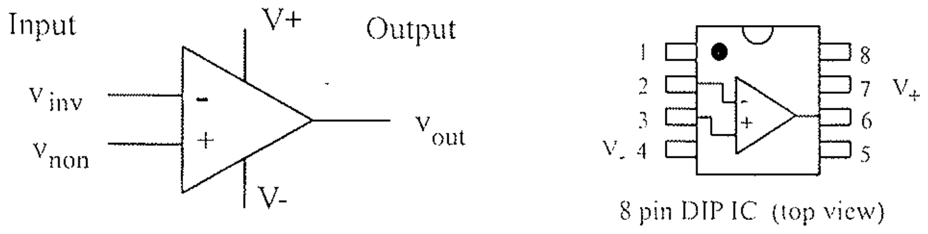


Figure 2.1: Operational amplifier IC.

The op-amp is represented by the circuit symbol shown at left in figure 2.1. It derives its (DC) power from a power supply via two *supply rails*, the voltages of which are called V_+ and V_-. Its output terminal delivers a voltage (v_{out}) that can swing over a specified range, often almost covering the span between V_+ and V_-. The amplifier has two voltage input terminals, one *inverting* (v_{inv}) and the other *noninverting* (v_{non}). These terms apply to the relative direction in which a voltage change applied at either input terminal drives v_{out}. Increasing v_{non} causes v_{out} to increase, however increasing v_{inv} causes v_{out} to decrease. The symbols used for the various voltages are by no means standard throughout the literature. You should be prepared to translate between different notations.

The op-amp integrated circuit (or IC) typically comes in an 8-pin DIP (Dual In- Line Package) format as shown at right in figure 2.1. This is the top view with the pins pointing down. Pin number 1 is identified with a small dot or by a small half circle depression at the top of the IC. The inputs are connected to pins 2 and 3 and the output is connected to pin 6. The power supply voltages comes in on pins 4 and 7 as shown. Be careful to get the polarity right. Connecting the supplies backwards will destroy the IC.

To a good approximation, the amplifier is a *difference amplifier* and is controlled only by the voltage *difference* between the two inputs. The effective controlling voltage is:

$$v_{dif} = v_{non} - v_{inv} \tag{2.1}$$

However, v_{inv} and v_{non} are individually restricted to remain within a certain range, which often comes close to spanning the gap between V_+ and V_-, and in some cases even exceeds it. In some amplifiers there is an additional restriction on the maximum absolute value for v_{dif}. It is important to remain within the specified ratings of the device (maximum voltages, maximum power dissipation) to avoid destroying it.

Modern general-purpose op amps are complicated devices containing many transistors and other circuit elements. A discussion of the internal structure of these integrated circuits (IC) will be postponed till Chapter 6. For now just try to make intelligent use of the specified op-amp terminal characteristics, which can be read from the *manufacturers' data sheets* (see section 2.3 of this chapter). Let's first consider an idealized op amp as shown in figure 2.2. As various applications are discussed, various ways in which practical op amps deviate from the ideal will be discussed.

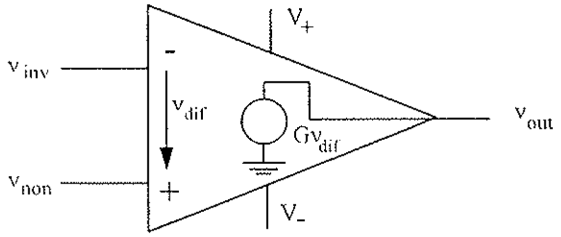


Figure 2.2: Operational amplifier circuit model.

The op amp is controlled by its voltage input signals, and it delivers a voltage at its output. Ideally the input terminals should have an infinite impedance (so that any source that drives them can act as voltage source) and the output terminal should have zero source impedance (so that it can act as voltage source in the face of any load). The internal working of the op-amp can approximately (i.e. more may be added to this model later) be modeled as a voltage source whose value is controlled by v_{diff} multiplied by a gain factor G as shown in figure 2.2.

The relationship between the effective input, v_{dif}, and the output, v_{out} is described by the *transfer characteristic*, a stylized version of which appears in figure 2.3. Because the amplification is high, a small v_{dif} suffices to drive v_{out} over its complete active range. With v_{dif} more positive or more negative, v_{out} saturates at the limiting voltages V_{SAT+} and V_{SAT-} respectively.

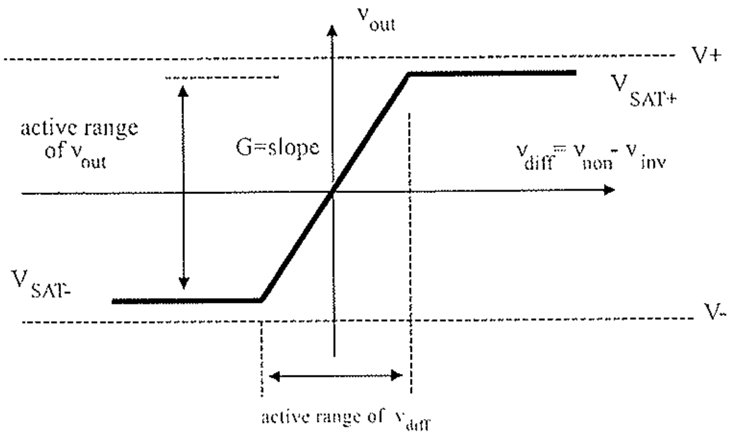


Figure 2.3: Voltage transfer characteristics for an operational amplifier.

The slope of the active portion of the curve defines the *voltage amplification* G:

$$G = \frac{\Delta v_{out}}{\Delta v_{dif}} \tag{2.2}$$

Defined this way, G is a positive number. In most op amps it is very large for low-frequency signals, perhaps in the range 10^4 - 10^7. Thus, for example, if V_{SAT-} = -15 V, V_{SAT+} = +15 V, and $G = 10^5$, a change in v_{dif} of only (30 V)/10^5= 300 μV suffices to drive the amplifier completely across its active range!

2.2 The Decibel

In practice it is convenient to measure signal amplification or attenuation (*gain* or *loss*) on a logarithmic scale. Successive gains experienced by a signal in passing through cascaded circuits can then be added algebraically (not multiplied). A negative logarithmic gain is of course just a loss or a signal attenuation.

The basic logarithmic unit is the bel, corresponding to a signal power ratio of 10:1. Two bels are a power ratio of 100:1, three bels 1000:1, and so on. The bel is subdivided into 10 decibels (or dB), so that:

$$\text{gain in dB} = 10 \log_{10}(\text{power out/power in}) \tag{2.3}$$

Given that signal power is proportional to V^2/R, the power ratio is $(V_{out}^2/R_{out})/(V_{in}^2/R_{in})$. If $R_{out} = R_{in}$ this leads to:

$$\text{gain in dB} = 10 \log_{10}(V_{out}^2/V_{in}^2) = 20 \log_{10}(V_{out}/V_{in}). \tag{2.4}$$

It is customary to retain this definition for the decibel gain even if $R_{out} \neq R_{in}$. In that case the unit is sometimes written dB_v to indicate that it refers to a V^2 ratio, not an actual power ratio.

Some examples: when a signal passes a circuit with no change of amplitude, the gain is 0 dB. When its amplitude is doubled, the gain is +6 dB ($\log_{10} 2 \sim 0.3$). When the amplitude is halved, the gain is -6 dB (or there is a loss of 6 dB). A factor of ten increase in voltage corresponds to 20 dB.

2.3 Manufacturers' Data Books and Spec-Sheets

Integrated circuits (IC's) and discrete devices (transistors and diodes) have many different parameters associated with them. The manufacturer of each device (for example National Semiconductor, Motorola, etc.) publish a list of the specifications of each device in what is called a "data book". Each data book usually contains the specifications for many different devices of a given type. For example there will usually be a data book on linear or analog devices and in this book there is usually a section on operation amplifier IC's. Other books may contain listings for all transistors or all digital devices in a particular family made by that manufacturer. Each manufacturer is free to organize these listings in whatever order they choose so you may have to look around a bit to find what you need.

The listing for each device can range from one page to several dozen pages. This listing is sometimes referred to as the "spec-sheets" for the device (short for manufacturers' specification sheets). Many devices are manufactured by several different manufacturers. Common devices such as you will be using have the same characteristics (or "specs") even though they can be made by different manufactures. For example a 741 op-amp made by National Semiconductor will behave the same as a 741 op-amp made by Motorola. This is not always true for exotic devices so in general you need to be careful in switching manufactures except when using simple components like used here.

Once you have found the listing for the device you are interested in you will usually see a short discussion of how the device works with a detailed numerical listing (usually in table form) and several graphs of the parameters for that device. Frequently you will see three different columns for each parameter labeled minimum, typical and maximum. This means that each parameter may fall within the range from the minimum value given to the maximum value given and the typical component will have the value labeled typical. This range of values results from the fact that electronic components are made in large quantities. To bring down the cost of the components the parameters are allowed to vary within some range. More expensive components have smaller ranges than cheaper ones and vice versa. As the semester goes on more discussion will be given on how to build circuits so that this variation will not effect their performance. Some parameters do not have a listing for all three columns. This usually means that one or more is columns is not important. For example the maximum bandwidth of an op-amp may not be specified. A larger bandwidth is good. As long as it meets its minimum bandwidth spec it is OK. In this case the minimum spec is the guaranteed minimum bandwidth of the amplifier. Other parameters may be missing a minimum spec but have only a maximum specification for similar reasons.

From time to time you will need to consult the manufactures' data books. There will be several copies of the appropriate books in the lab. Please consult the data books when you need to find out more about a specific component.

2.4 Connecting the Power Supplies

The op amps you will be using (types 741 and 3140) are commonly operated with supply rails at ±15 V with respect to ground. These rails are connected to two power supplies as shown in figure 2.4. The arrangement permits v_{out} to swing both positive and negative with respect to ground, a requirement in many applications. The input voltages can also move over a wide range. (A single power supply, for example with V_{-} grounded, is sometimes used when the amplifier voltages can be held in a more restricted range, see section 2.10.)

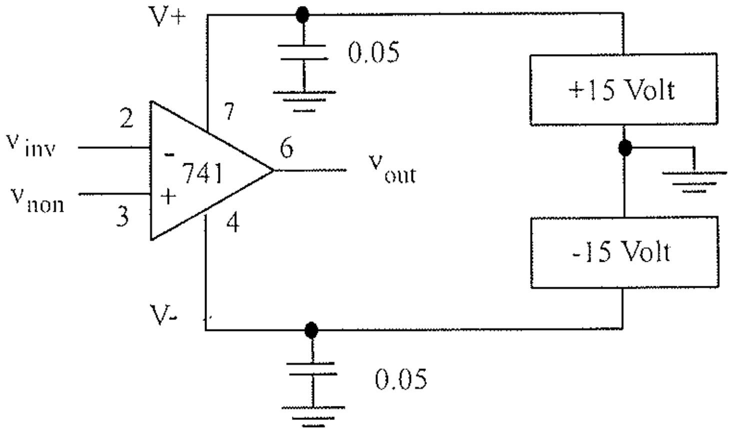


Figure 2.4: Connecting the power supplies to an operational amplifier.

Please make sure you know how to identify the pin numbers on the dual-inline package (DIP) of the op-amp (see figure 2.1). Also note that 0.05μF capacitors (anything in the range 0.01μF to 0.1μF is acceptable) are shown from each supply rail to ground. These bypass capacitors (use small ceramic disk types) should be mounted close to the amplifier. They help prevent spurious signals from appearing on the supply rails.

There is no ground terminal on the amplifier itself. It just rides between the supply rails. However, ground still plays its usual important role as reference point for all the signal voltages!

Op-amp circuit schematics are usually drawn without bothering to show the power supply and bypassing connections in detail. Simple arrows pointing to numbers may be drawn, which you should read as indicating the specified supply voltages. When you build the circuit, however, *you must pay careful attention to the supply-rail and bypassing requirements*. It is best to hook these items up first, with enough care so that they can be expected to stay put and not get in the way later. Please be very careful to see that the power supplies are connected *with the correct polarity*. Color coding (red for plus and black for negative) is helpful. as are a few labels on the circuit board itself. If you accidentally apply the wrong polarities. you will almost surely destroy the op amp.

The physical layout of the circuit components is also an important consideration. Although. in principle. many circuits will work with the components connected is strange physical positions. some will not. You will also find that an organized layout is easier to debug. Sensitive and/or high speed circuits can be very sensitive to the physical layout. Adjacent component can interfere with each other (this is called cross-talk) even though they are not electrically connected, and long wires can act as antennas (broadcasting as well as receiving). Cross-talk can produce an unexpected and confusing behavior in the circuits that is hard to debug. The first few circuits you will build may not be very complicated but you will eventually build more complicated circuits. It is probably best to start organizing your circuit layout now and continue for most circuits that you will be building in this course. One common requirement is for power and ground connections to be needed in many different places in the circuit. The solderless breadboards (see detailed drawing in figure 1.21) used in the lab have several long rows that are intended to distributes power conveniently to all portions of circuit. The diagram in figure 2.5 shows the breadboard with its long dimension placed horizontally. There are two long rows of contacts at the top and bottom that are connected all the way across except for a short spot in the middle (this short section is shown connected with a wire). It is usually easiest to simply make one whole row at the top be +15 volts and one whole row at the bottom be -15 volts (or whatever supply voltage you are using). The inner two rows can then just be connected as grounds. The IC can then conveniently straddle the slot running horizontally along the middle (so that opposite sides of the IC are electrically isolated)

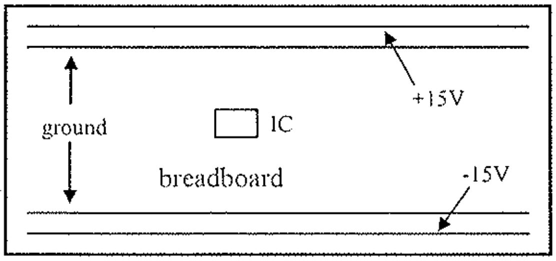


Figure 2.5: Breadboard layout.

2.5 Small-Signal Amplification and Input Offset Voltage

To study the amplification of the op amp without driving it beyond its active range, requires using a very small input signal. This can be obtained by passing the output of the function generator through a voltage divider, as indicated in figure 2.6. You can then observe v_{in} (a conveniently large signal) and from it infer the size of the voltage at the amplifier input, $v_{non} \sim v_{in}/500$.

Ideally, the active region of the op amp should be centered on the point $v_{dif}= 0$, which in this circuit would be the point $v_{non} = 0$, with v_{inv} set equal to zero. In practice there may be a small internal unbalance at the amplifier's input. This is called the *input offset voltage* , V_{os}, and in the 741 it is a few millivolts at most. The actual controlling voltage is:

$$v_{dif} = v_{non} - v_{inv} + V_{os} \tag{2.5}$$

The transfer characteristic is then centered on the point $v_{dif} = V_{os}$ rather than zero. The circuit below permits you to balance V_{os} with the small DC voltage injected at v_{inv}. The manufacturer provides an alternate scheme for balancing out V_{os}. Pins 1 and 5 for the 741 can be used to inject an "offset null" signal, to balance out V_{os}. The manufactures' spec sheets for the device usually show how to do this for each op-amp. For this measurement the direct injection of a balancing signal at v_{inv} will be used. With the components at our disposal, it permits a more sensitive control of the balance than use of pins 1 and 5. Frequently, the op-amp will be connected with other circuit elements in such a way that the total circuit will automatically correct for V_{os}, in which case a balance circuit can be omitted (see chapter 4).

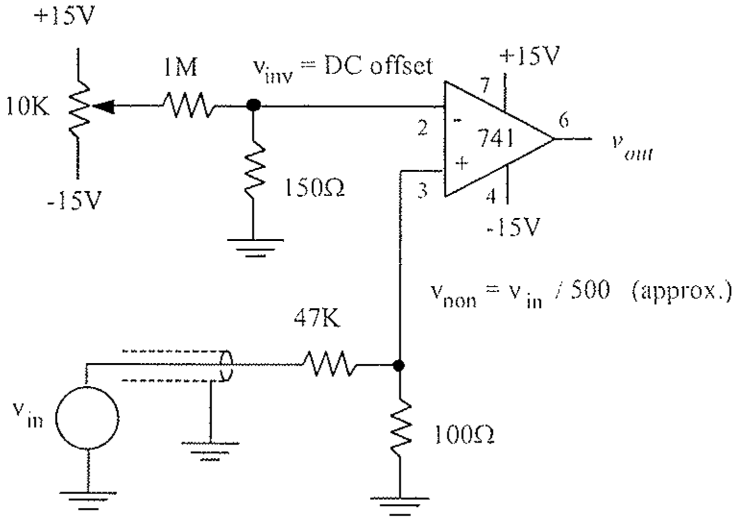


Figure 2.6: Measuring the open loop gain of the op amp.

Exp. 2.1 (Before proceeding with this experiment, check that the DC OFFSET control on the function generator is turned to its OFF position to ensures that the waveform it delivers is symmetrical with respect to ground.)

Set up the circuit shown in figure 2.6. With the generator delivering sine waves at 1 kHz, reduce the v_{in} signal amplitude progressively and readjust the DC offset to keep v_{out} symmetrical about ground. For many op amp IC's it is impossible to maintain a stable D.C. voltage level at v_{out} by use of this control. In this situation adjust the control as close to balance as you can and then adjust v_{in} until you can observe most of the v_{out} waveform in the active range (i.e. undistorted). An uncertainty of ±20 percent is acceptable here. When you have a value of v_{in} such that the amplifier clearly remains in the active range (i.e. if a sine wave is applied at the input an undistorted sine wave appears at the output), measure the amplification G of the amplifier. Repeat this measurement for frequencies in the range 10 Hz to 1 MHz, readjusting v_{in} and the DC offset as necessary. Measurements at intervals of factors of three should be satisfactory - i.e. 3 kHz, 10 kHz, 30 kHz, etc., and 300 Hz, 100 Hz, etc. You only need to sketch v_{in} and its corresponding v_{out} at one frequency, and then record the rest of the data in a table. With some op-amps at very high or low frequency, the measurement may be easier if you change the 100 Ohm resistor at the noninverting input (higher or lower values) to further increase or decrease the actual amplitude at the op-amp input. Measure and note the phase of

v_{out} relative to v_{in} at several frequencies - perhaps 1 kHz, 100 kHz, and 10 Hz.

Plot your results of G versus f on log-log paper (or G in db vs. f on semilog paper). This is referred to as the open loop gain curve. $\boxed{\textbf{end}}$

You no doubt found that the amplification of the op amp decreases at the higher frequencies. In fact, the op-amp is deliberately designed to make G proportional to 1/f above a *corner frequency* of around 10 Hz. By the time the frequency is as low as 1 Hz, G has attained its DC value. This *rolloff* in amplification is needed to ensure stability when the op amp is connected in a negative-feedback configuration (chapter 4). In some op amps the rolloff must be produced by external components. This permits the user some choice, but requires more work. The 741 and 3140, however, are *internally compensated* (see figure 2.7).

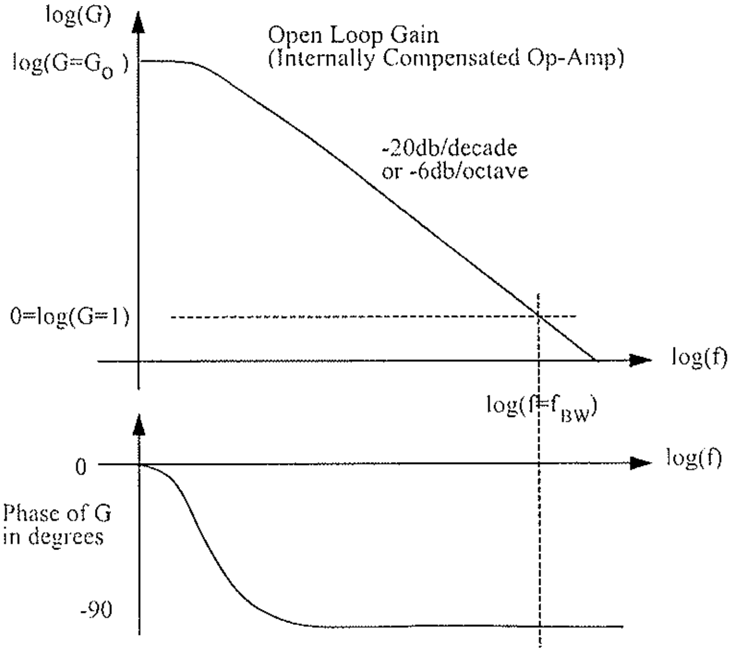


Figure 2.7: Op amp frequency response

A typical open loop gain curve for an internally compensated op amp is shown in figure 2.7. Note that a phase shift also accompanies this roll off of gain with frequency. Look up the specifications for the 741 op amp, and find the graph of open loop voltage gain versus frequency. The gain is plotted versus f (on a log-log scale), and the curve is a straight line above about 10 Hz. Check that the gain decreases by a factor of 10 (20 dB) for every factor of 10 increase in f, or equivalently by 6 dB for every factor of 2 in f. Thus G is proportional to f^{-1} , for f >10Hz. The 3140 has a similar curve but starts rolling off at a frequency of 100Hz. A gain rolloff inversely proportional to frequency is frequently encountered. It is described by saying that the gain rolls off at -20 dB/decade (of frequency), or by -6 dB/octave (an octave is a frequency ratio of 2:1).

2.6 The Overdriven Amplifier or Comparator

The op amp is not intended for the direct amplification of small signals in the manner you have just explored. It is not particularly stable, its gain is not particularly linear or constant, and its frequency response is tailored for feedback circuits. Also, any drift in its offset voltage V_{os} would make it very vulnerable in practice, as your experience with the measurement just made will have shown you. However this is the only way to measure the open loop gain curve (which will be needed in later chapters).

The op amp is, however, sometimes used "bare" (i.e. without negative feedback) to work with comparatively large signals. These overdrive the amplifier, swinging its output rapidly from V_{SAT+} to V_{SAT-}. If G is nearly infinite and the input signals are large then the following approximations can be made:

input condition	control voltage	output condition
$v_{inv} < v_{non}$	$v_{dif} > 0$	$v_{out} = V_{SAT+}$
$v_{inv} > v_{non}$	$v_{dif} < 0$	$v_{out} = V_{SAT-}$

In this use the exact shape of the input signal is not of interest. What concerns us is the time when the signal crosses a selected threshold. Since large voltages are being considered, ignore the presence of the small offset V_{os} and omit the offset-nulling circuit.

Exp. 2.2 Set up the circuit shown in figure 2.8, with v_{non} initially connected to ground. The triangle waveform from the generator drives the op amp across its active region, making v_{out} "switch" rapidly between V_{SAT+} and V_{SAT-}. Using a dual-trace scope, observe v_{in} and v_{out} and determine at what level of v_{in} this switching occurs. Measure V_{SAT+} and V_{SAT-}. Note that the finite propagation delay and slew rate of the op-amp (see figure 2.9) can cause an apparent shift in the switch points if the frequency of v_{in} is too high (i.e. keep the function generator frequency low).

Now connect v_{non} to the adjustable voltage which is called V_{ref}. (This voltage can change between -5 V and +5 V, and it is bypassed to ground to make sure no unwanted signals appear on it.) Vary V_{ref} and the amplitude of v_{in}, and observe at what level of the signal the op amp now switches. end

In this arrangement the op amp functions as a *voltage comparator*. It signals the moment when the input signal crosses a selected threshold, V_{ref}. v_{out} moves in the opposite direction to v_{in}. If you trade inputs to the amplifier ($v_{inv} = V_{ref}$, $v_{non} = v_{in}$) you obtain a comparator whose output moves in the same direction as the input signal.

The comparator marks the time of threshold crossing more accurately if its output transition takes place more rapidly. The rate at which v_{out} changes is determined, in the first place, by the op amp's amplification of that portion of v_{in} which traverses the active region. If the input signal changes fast, so does the output. However, the amplifier will take a small amount of time to respond (this is the propagation delay) and it can change v_{out} no faster than a certain maximum rate limited by its internal circuitry. This maximum rate is called the *slewing rate* (the 741 has a typical slew rate of 0.5 V/μSec and the 3140 has a typical slew rate of 9V/μSec).

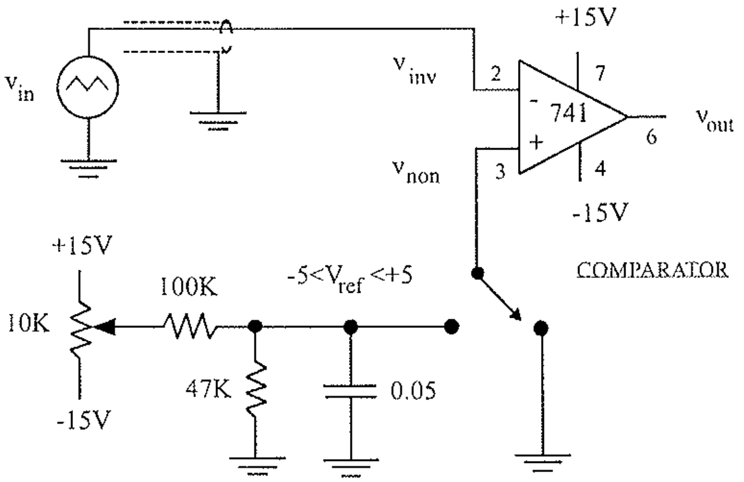


Figure 2.8: Comparator experiment.

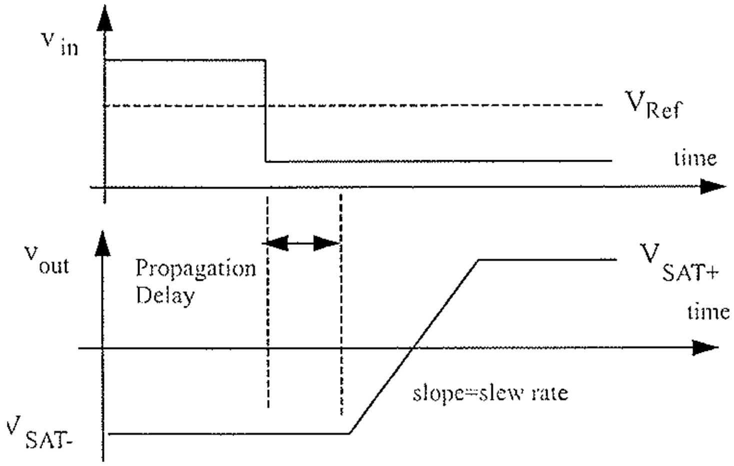


Figure 2.9: Op-amp response time.

Op amps are not designed to maximize the slewing rate. Their response is dominated by the gain rolloff at high frequencies. Specialized comparators are available which provide high slew rate and low propagation delay. Where the requirements are not too stringent, an op amp with a relatively high slewing rate (e.g., the 3140) is sometimes used as comparator anyway.

Exp. 2.3 Using the comparator circuit from figure 2.8, vary the amplitude and frequency of v_{in} (triangle wave) to see how the rate of change of v_{out} is affected. The referenece voltage V_{ref} should be left at one value (say GND). To obtain maximum slewing rate, change the function generator to deliver square waves. With such a signal, measure the slewing rate of your amplifier and compare to the value given in the spec sheets.

OPTIONAL: Measure the propagation delay of the op-amp. **end**

1. The inputs draw no current.
2. $V_{SAT+} = V_+$ and $V_{SAT-} = V_-$
3. $G = \infty$
4. $V_{OS} = 0$
5. Slew rate $= \infty$

Table 2.1: Some ideal op-amp rules

2.7 The Ideal Op-Amp

In practice it is difficult to think about all of the non-ideal properties of the op- amp (such as input offset voltage etc.) all of the time. It is easier to first think about the circuit as if the op-amp were ideal and then add in the non-ideal properties as small perturbations. The ideal op-amp rules are summarized in table 2.1.

These rules may be extended in later chapters. You should first try to understand a given op-amp circuit, assuming these rules and then add the non-ideal properties (such as finite gain or slew rate as small perturbations).

2.8 Positive Feedback and the Schmitt Trigger Circuit

If the input to the simple comparator crosses the threshold slowly, the output may become uncomfortably slow, too. Worse still, if the input should happen to hover for awhile right in the active region. v_{out} is neither V_{SAT+} nor V_{SAT-} but becomes instead a grotesquely amplified version of whatever small noise may be superimposed on the signal

The trick to force the comparator to have a clean sharp transition, no matter how slow the input signal may be, is to apply positive feedback (i.e. to return a fraction of the amplifier output back to its noninverting input), as shown in figure 2.10. This allows the amplifier to drive itself across its active region, regardless of how slowly the signal may be changing.

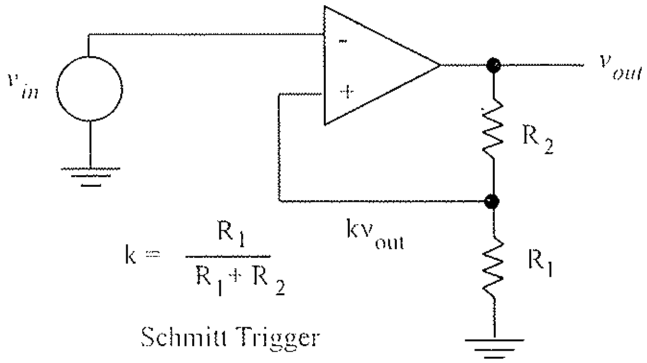


Figure 2.10: Schmitt trigger.

To visualize the action, suppose that v_{in} is negative and the op-amp is saturated at V_{SAT-}. Then

$v_{non} = kV_{SAT+}$, and this sets the threshold of the comparator. To be specific, assume $k = 1/20$ and $V_{SAT+} = -V_{SAT-} = 12$ V. Then in this initial state $v_{non} = +0.6$ V. Now let v_{in} increase slowly until it reaches the level of 0.6 V. At this point v_{out} begins to fall. As it does so, it takes v_{non} down with it. Therefore, even if v_{in} is perfectly stationary, the threshold is effectively whisking down past the signal, and the amplifier ends up switched to its other state. This description only requires that there be a signal returned to v_{non} which is substantially larger than what is required to make the switch. Since G is very large, this condition is easily met.

The switching action leaves v_{out} at V_{SAT-} (-12 V in this example), and the threshold is now determined by v_{non} which equal to -0.6 V. Clearly a very complete switch has taken place. Limited, in fact, only by the op amp's running into its other saturated state.

This new situation remains unchanged until v_{in} again approaches the threshold. In other words, v_{in} has to fall to -0.6 V before anything further can happen. If it does so, a similar snap action takes place in reverse, with v_{out} going to +12 V and v_{non} back up to +0.6 V. The circuit is then left in its original state.

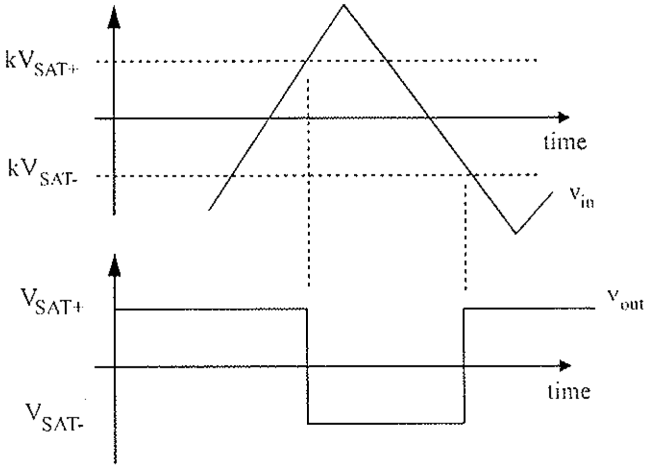


Figure 2.11: Response of the Schmitt Trigger

An important result of applying positive feedback, besides the speeding-up of the transition, is that the two switching thresholds are now at different levels as shown in figure 2.11. They differ by the amount of signal feed back to v_{non}, i.e. by $k(V_{SAT+} - V_{SAT-})$. In this example, by 1.2 V. This level difference is called hysteresis.

The use of hysteresis is common in many devices in which a definite, sudden action is to be provoked without hesitation on the threshold. For example, the thermostat which controls a house furnace has hysteresis. When the room temperature falls to the set point, the thermostat snaps to the on position, and the temperature has to rise to a significantly higher level before the thermostat snaps off again. Without this, the furnace would be turned on and off rapidly by minor temperature fluctuations near the threshold. These minor fluctuations can be likened to electrical noise (unwanted small signals). Hysteresis is a way of rejecting such noise.

Threshold circuits that exhibit hysteresis are usually called Schmitt Trigger circuits. This name can be applied to other implementations than the one just discussed. A graphical representation of the Schmitt trigger response is shown in figure 2.11. While v_{in} is below kV_{SAT-}, v_{out} is necessarily in its high state. While v_{in} is above kV_{SAT+}, v_{out} is necessarily low. In the region between these two levels, the circuit has a choice of states. Its condition will depend on which threshold was last crossed.

To design a Schmitt trigger circuit to switch at some particular value you will note that the value of k is determined once the power supply voltages are known. However, k only determines the *ratio* of R_1 to R_2, but you must still decide what absolute value to use. There are two other considerations which impose limits on the resistor values. On the low-resistance side you have to avoid loading the op amp output too heavily. Its output cannot supply much current. Perhaps a load of not less than 10 kΩ for $R_1 + R_2$ is appropriate. On the high side, the limit arises partly from the fact that the op-amp inputs draw a small amount of current, and partly from the desire that the R-C time constants formed by the resistance network and the stray capacitances should not be too long. You can get an idea about the R-C limit by assuming, for example, that the stray capacitance on the noninverting input will be of order 10 pF which interacts with the parallel combination of R_1 and R_2. Then, to keep the time constant below about 0.2 μs, what maximum resistance can be used?

$\boxed{\text{Exp. 2.4}}$ Construct the Schmitt trigger as shown in figure 2.10 with positive feedback using a 741 and +/-15 Volt supplies. Choose R_1 and R_2 to make $k \sim 1/10$ (see comments above for determining values for R_1 and R_2). Use triangle waves as input.

Measure the levels of v_{in} for which the upward and downward transitions (in v_{OUT}, fig. 2.11) occur (again remember to keep the input frequency low for this measurement so that the propagation delay does not affect your measurement), and verify that v_{out} moves at the maximum slewing rate of the amplifier, even when the input signal is very slow.

Substitute a type 3140 op amp for the 741. This has the same pin connections, but its slewing rate is considerably higher. The differential input voltage to the 3140 is limited to 8 V, so please make sure the amplitude for v_{in} is kept to a maximum of 5 V. Measure the maximum slewing rate for the 3140 and compare to the manufactures spec sheets.

OPTIONAL: Display the Hysteresis loop (as in figure 2.12) on the oscilloscope in XY mode (on the TEK1002 scope push the button labeled DISPLAY and set FORMAT=XY, then ch1=X and ch2=Y). Put one probe (x) on v_{in} and one probe on v_{OUT} and drive the circuit with a triangle wave. You may need to increase the scope persistence to see the whole curve. $\boxed{\text{end}}$

The amount of hysteresis in the comparator circuit is varied by adjusting k. The smaller k, however, the smaller the positive feedback, and the less forcefully does the amplifier drive itself. The transition becomes slower. On the other hand, if k is made too large there is so much hysteresis that the signal may be unable to swing across both threshold levels. In that case the circuit becomes stuck in one state.

The switching action of the Schmitt trigger can also be represented in a graph of V_{out} vs. V_{in} as shown in figure 2.12. The switching threshold depends on the recent history of the circuit as illustrated in the hysteresis loop. The arrows indicate the order that the input voltage is applied to the circuit.

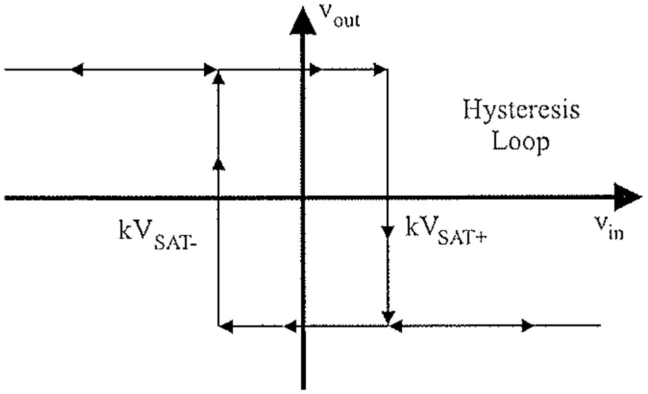


Figure 2.12: Hysteresis Loop

2.9 Free-Running (Astable) Multivibrator or Oscillator

Circuits that generate their own periodic waveforms are called *oscillators*, *astable multivibrators*, or *clocks*. They are free-running or astable. Often they serve as a primary source of signal. For example, the function generator is an oscillator. Most often the term oscillator is used for sine-wave generators. Such generators can be designed with good frequency stability and can thus serve as a basis for time measurement. The term *astable multivibrator* describes a circuit which delivers pulse waveforms, usually square waves which may be used to drive digital systems.

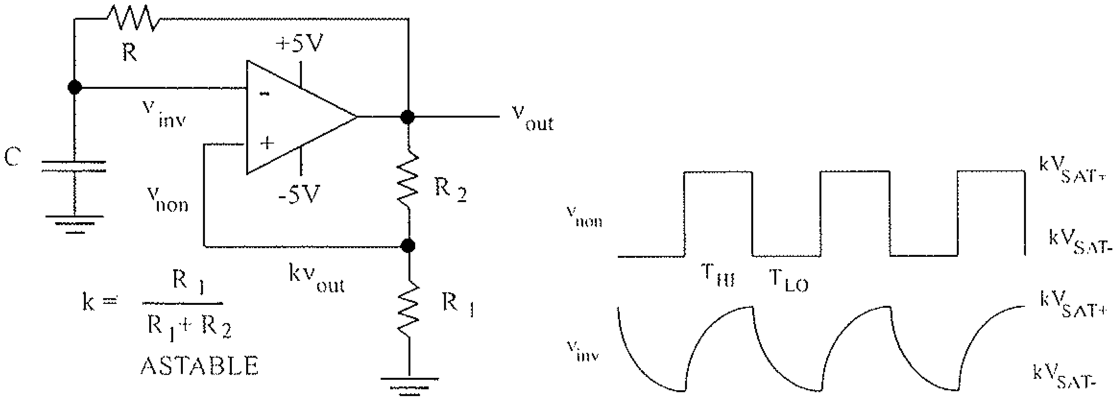


Figure 2.13: Astable Multivibrator (left) and its voltage waveforms (right)

The Schmitt trigger can be made to free-run by adding a loop which drives v_{inv} from v_{out} in such a sense as to try to force the circuit into the opposite state from the one it occupies. This *negative feedback* is slowed down by a low-pass R-C filter, and so it takes effect only after a certain time beyond each transition. The circuit shown in figure 2.13 produces square waves. The presence of positive feedback (through R_1 and R_2) drives the output into saturation. The noninverting input (v_{non}) will then be at kV_{SAT+} or kV_{SAT-}. The inverting input will be a low pass version of v_{out}. When v_{inv} reaches v_{non} then the output will switch to the other saturation level as shown in figure 2.13. v_{inv}

is effectively trapped between kV_{SAT-1} and kV_{SAT+} because when it reaches these points the output switches states.

The equation for RC charging and discharging (equation 1.22) can be used to find the time the output is high (T_{HI}) and the time the output is low (T_{LO}). To calculate T_{HI} the initial voltage across the capacitor is $V_I = kV_{SAT-}$ and the final voltage is $V_F = V_{SAT+}$ (see fig. 2.14). T_{HI} is found by solving for the time it takes the capacitor to charge to kV_{SAT+}. Similarly to find T_{LO}, $V_I = kV_{SAT+}$ and $V_F = V_{SAT-}$. T_{LO} is then the time the capacitor takes to reach kV_{SAT-}.

$$V_{CAP}(t = T_{HI}) = kV_{SAT+} = V_{SAT+} + (kV_{SAT-} - V_{SAT+}) \exp\left(\frac{-T_{HI}}{RC}\right) \tag{2.6}$$

$$V_{CAP}(t = T_{LO}) = kV_{SAT-} = V_{SAT-} + (kV_{SAT+} - V_{SAT-}) \exp\left(\frac{-T_{LO}}{RC}\right) \tag{2.7}$$

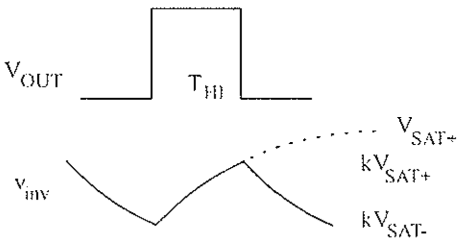


Figure 2.14: Astable multivibrator voltages.

If the saturation voltages of the op-amp are equal in value but opposite in sign (i.e. $V_{SAT+} = -V_{SAT-}$) then the time spent high is equal to the time spent low ($T_{HI} = T_{LO}$) and the period of oscillation is:

$$T = T_{HI} + T_{LO} = 2RC \ln\left[\frac{1+k}{1-k}\right] \quad \text{where} \quad k = \frac{R_1}{R_1 + R_2} \tag{2.8}$$

If the saturation voltages are not equal then the time high and the time low (of v_{out}) will not be precisely equal but it will still produce something similar to a square wave. You may find that this is true with low power supply voltages as in the next experiment. If you leave V_{SAT+} and V_{SAT-} as two separate variable in equations 2.6 and 2.7 you should find two different expressions for T_{HI} and T_{LO}.

Exp. 2.5 Assemble the circuit shown in figure 2.13, using a type 3140 op amp. to obtain a high slewing rate. To stay within the ±8 V limitation on the difference input, set the power supplies at ±5V. With reduced power-supply voltages, the op amp becomes less enthusiastic about driving its output load. You should therefore keep R_1, R_2 and R sufficiently large that the output current drain they require is not excessive (10K is reasonable).

Make the positive-feedback factor $k = 1/3$, and use an R-C time constant of about 100 μs in the negative-feedback path. Record the waveform at v_{out}. Measure the values of V_{SAT+} and V_{SAT-} and the time spent at each for the 3140 with supply voltages of ±5V. (You might want to momentarily substitute a type 741 in the same circuit to see how the V_{sat} values differ.)

View v_{inv} and v_{non} on a two-channel display. Sketch these waveforms and think about their relationships until you obtain a detailed understanding of how the circuit works. In particular, specify at what levels of v_{inv} the transitions occur, and think about how you would calculate the intervals of time the circuit spends in each of its two states. end

2.10 Single-Supply Operation

Sometimes it is necessary to use only a single power supply for an op-amp circuit for space or economical reasons. This can be done, for example, by setting V_- equal to ground. However, the voltage ranges for the inputs and output of the op amp are restricted by this choice. The 3140 is quite generous in this respect. Each input may actually go 0.5V below V_-, and the output can be brought down to within about 0.2 V of V_-. For the 741 the input and output limits lie at about 3V above V_-. Thus, unless you tread very carefully, you had best design the input and output voltage swings to stay well above ground, if you work with $V_-= 0$.

As an exercise, let's redesign the astable multivibrator circuit that was just used so that it operates from a single +10 V supply instead of ±5 V. The main concern has to be with v_{non}, since the two levels at this terminal fix the switching thresholds and therefore also "capture" the range of voltages over which v_{inv} swings. If the circuit were left as it stands, $v_{non} = v_{out}/3$, which places v_{non} in the lower third of the available voltage range. A symmetrical placement is better, however, because then the R-C circuit feeding v_{inv} can cover a more symmetrical part of its excursion, giving us more nearly equal "up" and "down" times.

Evidently v_{non} can be centered by connecting the lower end of R_1 to a potential of +5 V instead of to ground. Another way of seeing this is to note that +5 V occupies exactly the same relative place between V_+ and V_- as ground previously occupied. However this would still require two power supplies. How do you eliminate the other (+5V) power supply?

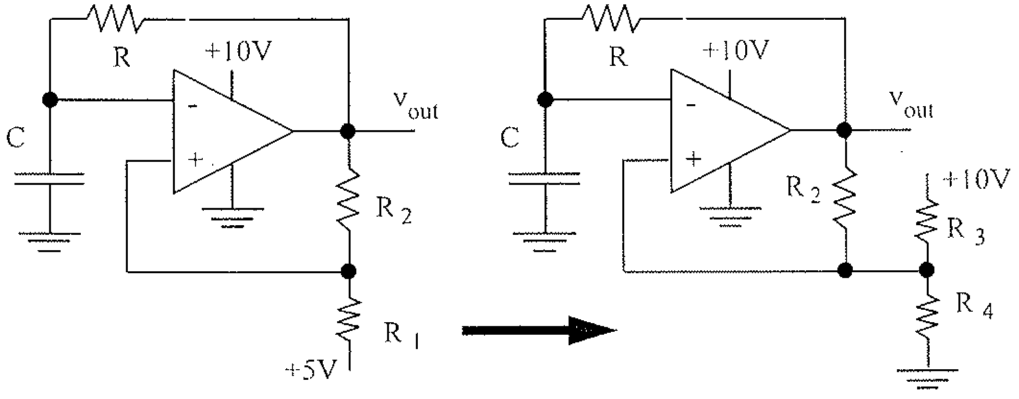


Figure 2.15: Astable multivibrator with a single supply

The evolution of the modified feedback circuit is illustrated above. Instead of a +5 V supply, use a voltage divider between +10 V and ground to give a Thevenin voltage of +5 V. Furthermore, choose R_3 and R_4 such that their combined Thevenin source resistance just takes the place of our previous R_1.

Op-Amp	741C	3140	LF156/LF356
DC gain, G_o	2×10^5	10^5	7×10^5
unity gain bandwidth, f_{BW}	1 MHz	4.5 MHz	4.5 MHz
slew rate	0.5 V/μSec	9 V/μSec	12 V/μSec
input impedance	2 MΩ	1.5×10^{12} Ω	10^{12} Ω
output impedance	75 Ω	60 Ω	
max output current	25 mA	+40/-18 mA	20 mA
input bias current, I_B	80 nA	10 pA	30 pA
input offset voltage, V_{OS}	2 mV	2 mV	3 mV
input capacitance	1.4 pf	4 pf	3 pf
max diff input voltage	30 V	8 V	40 V
min supply voltage $V_+ - V_-$	10 V	5 V	10 V

Table 2.2: Typical values of some op-amp parameters (from Manufacturers Data Books) Supply Voltage = +/-15 Volts.

$\boxed{\text{Exp. 2.6}}$ Calculate suitable values for R_3 and R_4 in figure 2.15 and modify your oscillator for single-supply operation from +10 V. (Hint: you may find Thevenin's theorem useful.) Test the circuit and record the waveform at v_{out} as in the previous experiment. $\boxed{\text{end}}$

2.11 Typical Values of Some Op-amp Parameters

Some typical op-amp parameters are shown in Table 2.2.

2.12 Practice Problems

[1] Find a value (in volts) for the typical and maximum input offset voltage for the 741C and the 3140A from the manufacturer's spec. sheets. Both pairs of numbers should be for an 8 pin DIP package and ±15V supplies at room temperature. You may get the full spec. sheets off the web or look at a hardcopy data book.

[2] Design an oscillator (astable multivibrator) to output a square wave that goes from -12 to +12 volts with a frequency of 1000 Hz using a single 741 op-amp. State a numerical value for each component and describe how it was calculated. Again you may assume that the op-amp is ideal. (Hint: Use a circuit similar to the one in Figure 2.13.)

First choose a set of supply voltages to give the desired output. Then choose an intermediate threshold for the op-amp inputs to oscillate between of ±2 volts. This fixes two more components. Then choose an appropriate time constant to give the correct frequency. You need to keep capacitor values significantly above a few picoFarads and to keep the resistors between about 1K and one megohm. To make things simpler choose a value of 10 K for the timing resistor.

[3] Analyze the Schmitt trigger op-amp circuit shown in figure 2.16, assuming that the op-amp is ideal. The power supply voltages are V_{+}=+15V and V_{-}=-15V and the resistor values are R_1=10K, R_2=4.7K and R_3=10K. a) What are the two possible values for v_{out}? b) At what value of v_{in} does the output switch from a high voltage to a low voltage? c) At what value of v_{in} does the output switch from a low voltage to a high voltage? (Hint you may find superposition helpful.)

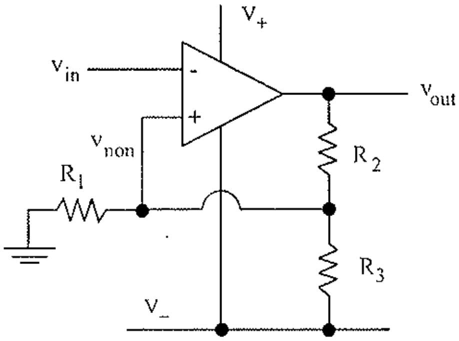


Figure 2.16: Schmitt trigger with offset switching points

[4] For the square wave oscillator circuit shown in figure 2.13 assume that the saturation voltage V_{SAT+} and V_{SAT-} are not equal in magnitude ($|V_{SAT+}| \neq |V_{SAT-}|$) and derive an expression for T_{HI} and T_{LO} as in equations 2.6 through 2.8 for an arbitrary value of V_{SAT+} and V_{SAT-}.

Chapter 3

SINUSOIDAL SIGNALS

The previous chapters described the behavior of RC circuits by treating the voltage and current as a function of time. This mode of analysis is referred to as being in the *time domain.* If instead, the voltage and current are assumed to be steady state sine waves (the *sinusoidal steady state*) then the circuit analysis can be greatly simplified in many cases. This type of AC circuit analysis is referred to as being in the *frequency domain.* When reactive circuit elements such as capacitors and inductors are included in the circuit then the voltage and the current no longer need to be in phase with one another. The important parameters for voltage and current sine waves are the amplitude, frequency and phase.

This chapter assumes some familiarity with the basic properties of sine wave signals, their representation by means of the phasor diagram, and with the performance of resistors, capacitors, and inductors in AC circuits.

> *There is very little laboratory work associated with the material of this chapter. You have a chance to catch up, or forge ahead to chapter 4 in the lab.*

3.1 Phasor Representation

AC time-dependent signals will be denoted by lower-case letters (v, i, etc.) while fixed values are shown in capitals. The general expression for an AC voltage in the sinusoidal steady state is:

$$v(t) = V_0 \cos(\omega t + \phi) \tag{3.1}$$

where V_0 is the *amplitude* or *peak value* of the voltage signal, ω is the angular frequency, and ϕ is the *phase.* ω is measured in units of *radians/second*, and is related to the (cycle) *frequency* f (measured in *Hertz = cycles/second*) by:

$$\omega = 2\pi f \tag{3.2}$$

The graph in figure 3.1 shows the voltage $v(t)$ as a function of time t. ϕ is the angle by which the peak of the cosine curve is phase advanced, i.e., shifted to the left from $t = 0$. The phasor diagram, on the right, is drawn for the moment $t = 0$. $v(t)$ is the horizontal projection of the phasor $\underline{V}$, which rotates at angular velocity ω. You can visualize how this projection $v(t)$ varies as the phasor rotates from the position shown, and compare with the graphical plot of $v(t)$ at the left.

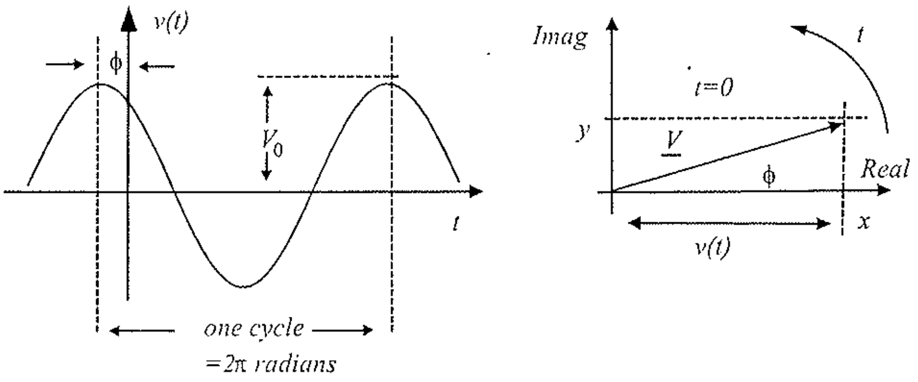


Figure 3.1: Voltage and phasors.

3.2 Complex Notation

Two parameters are needed to specify an AC signal of a given frequency: the amplitude and the phase. On the phasor diagram (figure 3.1) these two parameters appear as the length and angular position of the phasor. However, you could equally well describe the phasor by its two rectangular components, x and y (or the real and imaginary parts). The diagram on the right in figure 3.1 is similar to the one used for representing complex numbers, which also need two parameters (the real and imaginary parts). The apparatus of complex algebra turns out to be a convenient and powerful technique to handle phasor calculations.

To introduce the idea, the axes on the phasor diagram (figure 3.1) are labeled Real and Imag, as in the usual notation for the complex plane. To avoid confusion with the symbol for current, the symbol j (not i) is used for $\sqrt{-1}$. The complex number represented by V is then:

$$V = x + jy = V_0 \cos\phi + jV_0 \sin\phi = V_0(\cos\phi + j\sin\phi) \tag{3.3}$$

$$= V_0 e^{j\phi} \tag{3.4}$$

The expression in parentheses is equal to $e^{j\phi}$, by Euler's identity:

$$e^{j\phi} = \cos\phi + j\sin\phi \tag{3.5}$$

This method of writing the complex number V displays two important parameters: the magnitude, V_0, and the phase, ϕ. In the AC context these are the amplitude and phase.

The important factor $e^{j\phi}$ has the special function of a *rotator*. It does not change the magnitude of what it multiplies, because $|e^{j\phi}| = 1$, but it increases the angle by an amount ϕ. To see that this is so, consider what $e^{j\phi}$ does to a phasor $\underline{V}$, of length V_0 lying along the real axis. The product $V_0 e^{j\phi}$ has the same length, but it lies at an angle ϕ to the real axis. Or, supposing you start with a phasor $V_0 e^{j\alpha}$ which lies at an angle α, then multiplication by $e^{j\phi}$ yields $V_0 e^{j\alpha} e^{j\phi} = V_0 e^{j(\phi+\alpha)}$, again an extra rotation through ϕ without change of length.

At times other than $t = 0$ the phasor has rotated away from its position shown, to an angle $(\omega t + \phi)$. Thus, at time t;

$$V = V_0 e^{j(\omega t+\phi)} = V_0 e^{j\phi} e^{j\omega t} \tag{3.6}$$

The angular velocity of the phasor, $e^{j\omega t}$ can be factored out to separate the time dependence of the phasor. Recall that the instantaneous value of $v(t)$, at all times, is the horizontal projection of the phasor or, the *real part* of $\underline{V}$;

$$v(t) = Re[\underline{V}] \tag{3.7}$$

In the physical world the voltage and current are real quantities. Using complex notation is purely a mathematical convenience. The imaginary part of the voltage and current will usually be ignored at the end of the calculation (but not before) to compare to the measured real part.

Once the frequency is specified, it is often convenient to take the time dependent factor $e^{j\omega t}$ as *implied*, and to write only $V_0 e^{j\phi}$ to represent the signal. This is equivalent to drawing the phasor diagram only for $t = 0$.

3.3 Complex Impedance

The ratio of voltage to current in an AC network in the impedance. When you take the ratio of the amplitudes of voltage and current, you obtain a number for the impedance:

$$Z = \frac{V}{I} \tag{3.8}$$

(measured in ohms) to which you still have to append the phase difference between v and i before the description is complete. Instead, you can divide the *complex amplitudes* V and I, and obtain a *complex impedance* which describes both magnitude and phase:

$$Z = \frac{V}{I} = \frac{V_0 e^{j(\omega t + \delta)}}{I_0 e^{j\omega t}} = \frac{V_0}{I_0} e^{j\delta} = |Z| e^{j\delta} \tag{3.9}$$

where δ is the phase angle by which v leads i. This complex impedance concept permits a generalization of Ohm's law to AC circuits. With an AC impedance in place of a resistance the familiar circuit reduction methods can be applied directly. For example, adding impedances in series, or combining the reciprocals of impedances (*admittances*) for elements in parallel.

3.4 Working with Complex Numbers

First, a comment on the two ways a complex number can be written. The rectangular-coordinate notation:

$$Z = x + jy \tag{3.10}$$

and the polar-coordinate notation:

$$Z = |Z| e^{j\phi} \quad \text{where} \quad |Z| = \sqrt{x^2 + y^2} \tag{3.11}$$

The rectangular form is convenient when complex numbers are to be added or subtracted. You merely combine the several x's and y's separately.

$$Z_1 = x_1 + jy_1 \tag{3.12}$$

$$Z_2 = x_2 + jy_2 \tag{3.13}$$

$$Z_1 + Z_2 = (x_1 + x_2) + j(y_1 + y_2) \tag{3.14}$$

$$Z_1 - Z_2 = (x_1 - x_2) + j(y_1 - y_2) \tag{3.15}$$

The polar form is more appropriate for multiplication or division:

$$Z_1 Z_2 = (|Z_1|e^{j\phi_1})(|Z_2|e^{j\phi_2}) = |Z_1||Z_2|e^{j(\phi_1+\phi_2)} \tag{3.16}$$

$$\frac{Z_1}{Z_2} = \frac{|Z_1|e^{j\phi_1}}{|Z_2|e^{j\phi_2}} = \frac{|Z_1|}{|Z_2|}e^{j(\phi_1-\phi_2)} \tag{3.17}$$

Also,

$$\frac{1}{Z} = \frac{1}{|Z|e^{j\phi}} = \frac{1}{|Z|}e^{-j\phi} \tag{3.18}$$

When complex numbers are multiplied or divided, their magnitudes are multiplied or divided accordingly. Their phase angles, however, combine by addition or subtraction.

The transition from rectangular to polar form, or vice versa, is made according to

$$x = |Z|\cos\phi \quad ; \quad y = |Z|\sin\phi \tag{3.19}$$

$$|Z| = \sqrt{x^2 + y^2} \quad ; \quad \phi = \tan^{-1}(y/x) \tag{3.20}$$

The real part of the impedance is the resistance R and the imaginary part of the impedance is called the reactance X. Both are measured in units of Ohms.

$$Z = R + jX \tag{3.21}$$

3.5 The Inductor

If a current $i = I_0 \cos\omega t$ is sent through an inductor L, the voltage across the inductor is:

$$v(t) = L\frac{di(t)}{dt} \tag{3.22}$$

In complex notation, $i(t) = I_0 e^{j\omega t}$, and:

$$\frac{di(t)}{dt} = I_0\left(j\omega e^{j\omega t}\right) = j\omega \times i(t) \tag{3.23}$$

Thus *time differentiation of the factor $e^{j\omega t}$ is equivalent to multiplication by $j\omega$.* Reducing differentiation to an algebraic operation is what makes the exponential notation so powerful.

The voltage across the inductor is:

$$v(t) = L\frac{di(t)}{dt} = j\omega L i(t) \tag{3.24}$$

and the complex impedance is:

$$Z_L = \frac{V}{I} = j\omega L \tag{3.25}$$

The magnitude of this impedance is $|Z_L| = \omega L$. The factor j signifies a phase shift between v and i. If a unit vector lying along the real axis is multiplied by j, the result is a new unit vector lying along the imaginary axis. So multiplying by j represents a rotation through 90^o. (Alternatively, you can verify more formally, from Euler's identity, that $e^{j(90^o)} = j$.)

The complex impedance $j\omega L$ thus describes both the magnitude of the ratio V/I and the 90^o phase advance of v with respect to i (as shown in figure 3.2). (When the phase difference is exactly $\pm 90^o$, an impedance is called a reactance.)

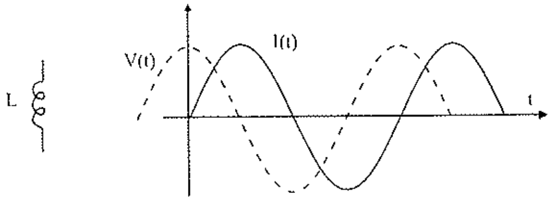


Figure 3.2: The inductor with sinusoidal signals.

At frequencies below a few megahertz, useful inductors are large and bulky. Capacitors can perform most of the required functions at low frequencies without inductors. Therefore it is rare to see an inductor for low frequency circuits (as is the case here). However inductors are important for radio frequency (RF) circuits where only a few turns of wire produce a significant reactance.

3.6 The Capacitor

The voltage across a capacitor C is:

$$v(t) = \frac{q}{C} \tag{3.26}$$

and charge is the time integral of current:

$$q(t) = \int i(t)dt \tag{3.27}$$

Strictly speaking, a constant of integration should be included with the integral. However, for present purposes, with only sinusoidally oscillating AC signals, the constant of integration can be ignored. (Other types of signals may require including a constant of integration.) Inserting the integral into the expression for C yields:

$$v(t) = \frac{1}{C}\int i(t)dt = \frac{1}{C}\int I_0 e^{j\omega t}dt = \frac{I_0 e^{j\omega t}}{j\omega C} = \frac{i(t)}{j\omega C} \tag{3.28}$$

Integrating $e^{j\omega t}$ is equivalent to division by $j\omega$. The complex impedance of the capacitor is:

$$Z_C = \frac{V}{I} = \frac{1}{j\omega C} \tag{3.29}$$

The factor $1/j = -j$ represents a 90^o phase lag of v behind i.

The values of the reactance (imaginary part of the impedance) for a simple capacitor and inductor are shown in fig 3.4 for a typical range of frequencies.

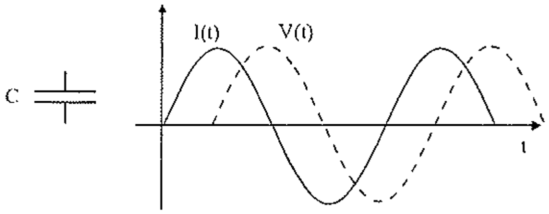


Figure 3.3: The capacitor with sinusoidal signals.

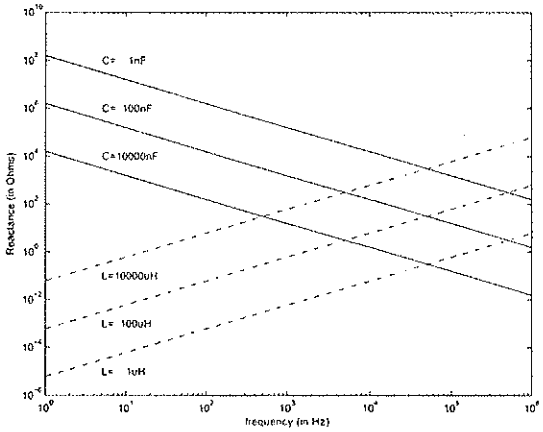


Figure 3.4: Reactance versus frequency for capacitance C and inductance L for sinusoidal signals.

3.7 Combinations of Elements

Complex notation allows us to generalize the circuit analysis procedures for DC circuits to steady-state AC circuits. Simply replace R by Z, make V and I complex, and keep track of the relative phase of each signal. The series and parallel combination of impedance is just the same as that for resistance:

$$\text{Series} \quad Z_T = Z_1 + Z_2 + Z_3 + \cdots \tag{3.30}$$

$$\text{Parallel} \quad \frac{1}{Z_T} = \frac{1}{Z_1} + \frac{1}{Z_2} + \frac{1}{Z_3} + \cdots \tag{3.31}$$

However unlike pure resistance each impedance element may vary with frequency. The high and low frequency limiting values of R,L,C elements are shown in table 3.1.

component	low frequency	impedance	high frequency
inductor L	short circuit	$j\omega L$	open circuit
resistance R	R	R	R
capacitor C	open circuit	$1/(j\omega C)$	short circuit

Table 3.1: Passive components in the frequency domain.

Example 1: What is the impedance of a series combination of R and C?

$$Z = Z_R + Z_C = R + \frac{1}{j\omega C} = R + j\left(-\frac{1}{\omega C}\right) \tag{3.32}$$

The magnitude and phase of Z are obtained by converting to the polar form:

$$\begin{aligned} Z &= |Z|e^{j\phi} && (3.33) \\ |Z| &= \sqrt{R^2 + \frac{1}{(\omega C)^2}} && (3.34) \\ \phi &= \tan^{-1}\left(\frac{-1}{\omega RC}\right) && (3.35) \end{aligned}$$

At low frequencies, Z approaches $1/\omega C$ and ϕ tends to -90°. This indicates that the impedance of the capacitor has become dominant, and R negligible in comparison. At high frequencies, on the other hand, Z tends to the value R and ϕ becomes zero. Now the impedance of C is negligible, and you are left with R alone.

Example 2: What is the impedance of a *parallel* combination of R and L? Recall that, in parallel circuits, the reciprocals of the impedances add, so that:

$$\frac{1}{Z} = \frac{1}{R} + \frac{1}{j\omega L} = \frac{1}{R} + j\left(-\frac{1}{\omega L}\right) \tag{3.36}$$

For the next step it is easiest to stay with 1/Z and convert it to polar form:

$$\begin{aligned} \frac{1}{Z} &= \frac{1}{|Z|}e^{j\phi} && (3.37) \\ \frac{1}{|Z|} &= \sqrt{\frac{1}{R^2} + \frac{1}{(\omega L)^2}} && (3.38) \\ \phi &= \tan^{-1}\left(-\frac{R}{\omega L}\right) && (3.39) \end{aligned}$$

Then the inverse of this expression gives Z.

Example 3: What is the impedance of a series R-L-C circuit?

$$\begin{aligned} Z &= R + j\omega L + \frac{1}{j\omega C} = R + j\left(\omega L - \frac{1}{\omega C}\right) && (3.40) \\ |Z| &= \sqrt{R^2 + \left(\omega L - \frac{1}{\omega C}\right)^2} && (3.41) \\ \phi &= \tan^{-1}\left[\frac{\omega L - \frac{1}{\omega C}}{R}\right] && (3.42) \end{aligned}$$

The magnitude of the impedance of a series combination of RLC is large at both high frequency and low frequency (see figure 3.5). The impedance reaches a minimum value of $|Z| = R$ when $\omega L = 1/(\omega C)$ or $\omega_0 = 1/\sqrt{LC}$. This frequency is called the *resonance frequency* of this circuit. A parallel combination of RLC elements also has a resonance but its impedance is a maximum at the resonance frequency. Resonant LC circuits are commonly used in radio circuits to select a particular frequency (radio or TV station).

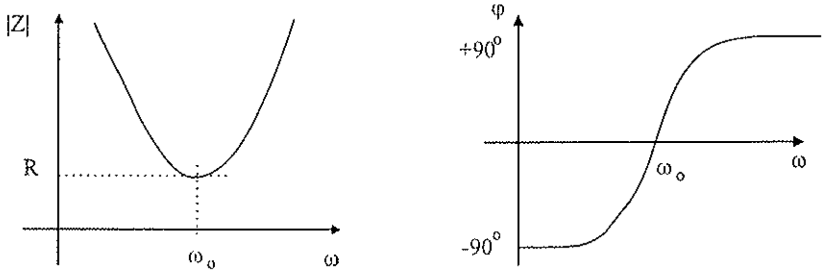


Figure 3.5: Series combination of RLC.

3.8 The Transfer Function and Bode Plots

The *transfer function* of a network is the ratio of its output to its input. Of course the form of the input must be known before the gain can be specified. Usually the input and output signals will be in the form of a sinusoidal voltage signal although many others forms are possible. The transfer function can also be referred to as the gain of the network and therefore has the symbol $G(\omega)$.

$$G(\omega) = \frac{v_{out}(\omega)}{v_{in}(\omega)} \tag{3.43}$$

A particularly useful way of displaying the frequency dependence of G is on a log-log plot. $\log|G|$ is plotted on the vertical axis and $\log\omega$ on the horizontal axis. Typically the frequency (and G) may vary over many orders of magnitude and the log-log scale compresses this range into a manageable size. Furthermore, G typically varies as some power of ω in some portion of the graph, which will become a straight line on a log-log scale, making it easy to draw. The transfer function, $G(\omega)$ is a complex quantity and also posses a phase. A Bode plot of $G(\omega)$ has two components, the magnitude and the phase. The magnitude is plotted on a logarithmic scale but the phase is plotted on a linear scale. Both use the same $\log\omega$ horizontal scale.

The decibel scale ("dB") is commonly used to measure power and voltage gains. Gain can also be expressed in dB form as:

$$G_{dB} = 20\log_{10}(V_{out}/V_{in}). \tag{3.44}$$

With this definition, the dB scale is automatically a log scale. A graph showing G_{dB} vs. $\log\omega$ is called a "Bode plot" (named for the engineer Henrik Bode, who introduced such plots). You may plot $|G|$ in dB on a linear scale or $|G|$ by itself on a log scale.

The next two sections discuss the transfer function $G(\omega)$ for the high-pass and low- pass R-C filters. These were discussed in chapter I in the time domain, now they will be discussed in the frequency domain.

3.9 High-Pass R-C Filter

The high pass filter is shown in figure 3.6. A conventional schematic is on the left and an equivalent circuit is shown on the right using generalized impedance elements for capacitance $Z_C = 1/(j\omega C)$ and resistance $Z_R = R$.

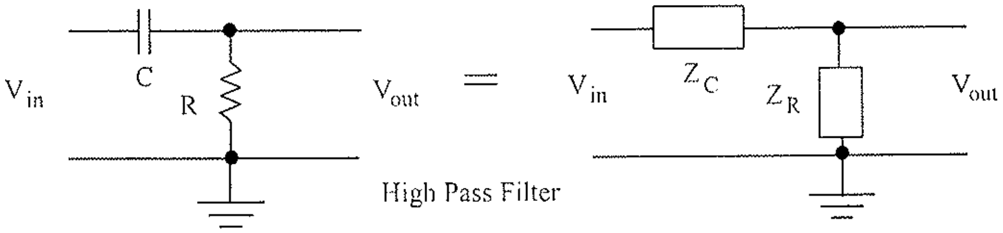


Figure 3.6: High pass filter with complex impedance.

By analogy with the resistive voltage divider, the transfer function is:

$$G(\omega) = \frac{v_{out}(\omega)}{v_{in}(\omega)} = \frac{Z_R}{Z_R + Z_C} = \frac{R}{R + \frac{1}{j\omega C}} = \frac{j\omega RC}{1 + j\omega RC} \tag{3.45}$$

$\tau = RC$ is the time constant of the RC combination. The magnitude and phase of the transfer function are:

$$|G(\omega)| = \frac{\omega RC}{\sqrt{1 + (\omega RC)^2}} \tag{3.46}$$

$$\phi[G(\omega)] = 90^o - \tan^{-1}(\omega RC) \tag{3.47}$$

At low frequencies ($\omega RC << 1$), $G \sim j\omega\tau$. Then $|G| << 1$, and v_{out} has a 90^o phase advance over v_{in}, from the factor j. At high frequencies ($\omega RC >> 1$), on the other hand, $|G| \sim 1$ and v_{out} is in phase with v_{in}.

A log-log plot (i.e., $\log|G|$ on one axis, $\log\omega$ on the other) turns out to be a useful way of displaying the ω dependence of G. The graph in figure 3.7 shows a Bode plot for the high-pass filter. The low-frequency part of the plot is a straight line, because $G \propto \omega$. The high-frequency part of the plot is also a straight line, with $\log G = 0$, because $G \sim 1$. At very high or low frequency asymptotic parts of the plots are straight lines. The log-log plot makes it easy to extrapolate the extremes in toward the point at which they meet. G deviates from the straight lines as this meeting point is approached, but it turns out that not much of an error is involved in pretending that G actually follows the asymptotes right into the corner.

The *corner frequency* , ω_c, is found by noting that the low-frequency asymptote represents $|G| = \omega RC$, which intersects the level $|G| = 1$ when $\omega RC = 1$.

$$\omega_c = \frac{1}{RC} \quad \text{or} \quad f_c = \frac{1}{2\pi RC} \tag{3.48}$$

At the corner frequency the transfer function has a particular values of:

$$G(\omega = 1/(RC)) = \frac{j}{1+j} \tag{3.49}$$

Which has a magnitude of $1/\sqrt{2}$ or -3dB and a phase of 45^o or $\pi/4$.

On the low frequency side of the high-pass filter plot, a change of one power of 10 in ω corresponds to a change in G_{dB} of 20. This asymptotic behavior corresponds to a rise in gain of "20 dB per decade". The rising portion (low frequency in the high pass filter) of the Bode plot is a straight line and is said

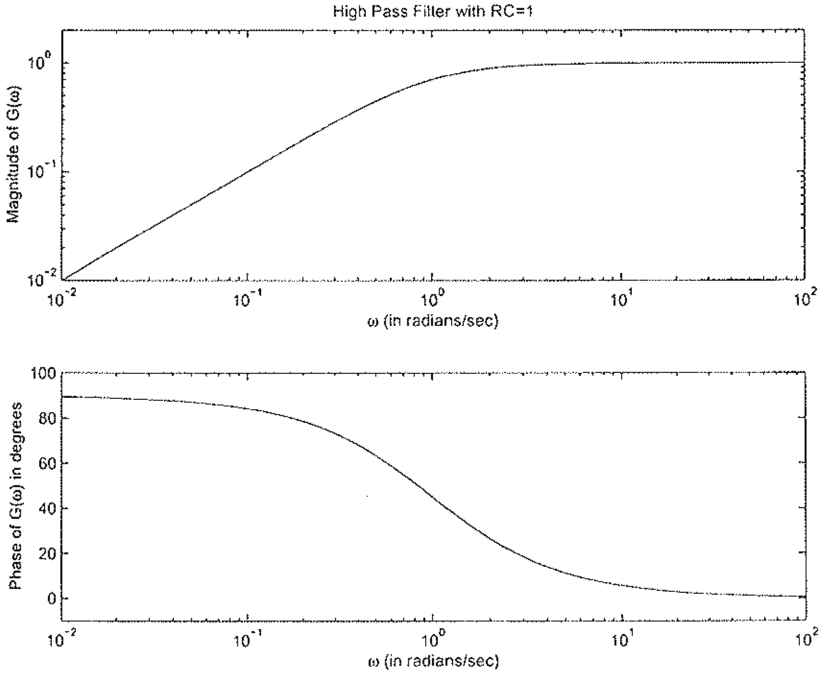


Figure 3.7: High pass filter Bode plot.

to have a slope of 20db/decade. On a semilog scale (G_{db} vs. $\log(\omega)$) the slope between two points (G_{db1}, ω_1) and (G_{db2}, ω_2) is:

$$\text{slope} = \frac{\Delta G_{db}}{\Delta \log_{10}(\omega)} = \frac{G_{db2} - G_{db1}}{\log_{10}(\omega_2) - \log_{10}(\omega_1)} = \frac{20 \log_{10}(G_2/G_1)}{\log_{10}(\omega_2/\omega_1)} \tag{3.50}$$

A one decade change in ω means that $\omega_2 = 10\omega_1$ leaving $\log_{10}(\omega_2/\omega_1)$=1. To get a 20 db change per decade change in frequency means that G must also change by a factor of 10 leaving $G_2 = 10G_1$ and $20\log_{10}(G_2/G_1)$=20. Saying the slope is 20 db/decade is just another way of saying $G \propto \omega$. 40 db/decade means $G \propto \omega^2$, 60 db/decade means $G \propto \omega^3$, etc.

Exp. 3.1 Construct a high-pass R-C filter, using the component values of experiment 1-4 (C= 0.01 μF, R= 10K). Drive it with sine waves from the function generator. Set up the two-channel scope to measure both the input and output p-p voltages (and calculate the ratio v_{out}/v_{in} to obtain the gain) and the relative phase of v_{out} and v_{in} (the ratio of time between peaks Δt to the period T multiplied by 2π). You should sketch one typical pair of waveforms (v_{in} and v_{out}) at one frequency and then make a table of numbers v_{in}, v_{out}, Δt, and T for the other frequencies.

First calculate the expected value of the corner frequency ω_c. Then make measurements at: (i) less than about $\omega_c/10$; (ii) ω_c itself; and (iii) greater than about $10\omega_c$. Make enough additional measurements above and below the corner frequency to extract a value of the slope from your graph. After you have plotted your measurement on semilog paper using the dB scale (or log-log paper using $|G|$ itself), you can draw in the high and low frequency straight line asymptotes. (Note that 20 dB/decade has a slope of 1 on the standard log-log plot.) The intersection of the high and low frequency asymptotes provides a more accurate value for ω_c, to be used for the second measurement. Determine the magnitude and phase of G at ω_c, and compare your results with those you calculate from the complex expression derived above. end

3.10 Low-Pass Filter

A schematic of the low pass filter is shown on the left in figure 3.8. An equivalent circuit with generalized impedance elements is shown on the right.

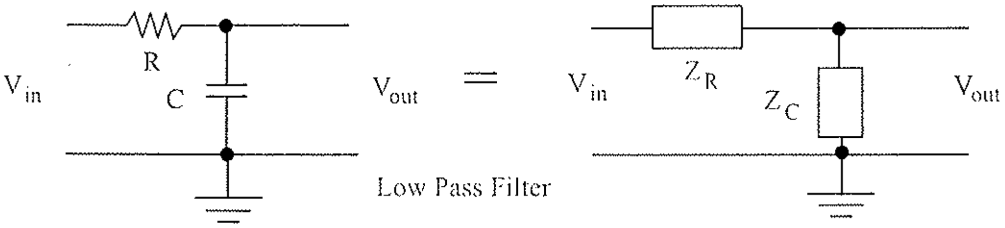


Figure 3.8: Low pass filter with complex impedance.

Again by analogy with the voltage divider the transfer function for the low pass filter is:

$$G(\omega) = \frac{v_{out}(\omega)}{v_{in}(\omega)} = \frac{Z_C}{Z_R + Z_C} = \frac{\frac{1}{j\omega C}}{R + \frac{1}{j\omega C}} = \frac{1}{1 + j\omega RC} \tag{3.51}$$

A Bode plot of the transfer function for the low pass filter is shown in figure 3.9. The low-frequency asymptote of the Bode plot of the low-pass filter is $G = 1$, with zero phase shift between v_{out} and v_{in}. The high-frequency asymptote is $G \sim 1/(j\omega RC)$, so that $|G| \sim 1/(\omega RC)$ and there is a 90^o phase lag of v_{out} and v_{in}. The relationship $G \propto \omega^{-1}$ appears as a straight line on the log-log plot, with a slope of -20 dB/decade. The two asymptotes once again intersect at a corner given by $\omega_c = 1/(RC)$.

Exp. 3.2 Rearrange your R-C filter in the previous experiment to be a low-pass and make measurements for the Bode plot in the same manner as before. Make the appropriate plot. Check that your values for G at ω_c agree with those calculated from the complex expression. end

3.11 Bode Plots, Poles and Zeros

The transfer function for many practical circuits can be put into a *pole-zero* form similar to:

$$G(\omega) = K(j\omega)^n \frac{(1 + j\omega/z_1)(1 + j\omega/z_2)\cdots(1 + j\omega/z_N)}{(1 + j\omega/p_1)(1 + j\omega/p_2)\cdots(1 + j\omega/p_M)} \tag{3.52}$$

where n is an integer and K is a constant. The constants z_1 through z_N are the zeros of the transfer function and the constants p_1 through p_M are the poles of the transfer function. The zeros are the

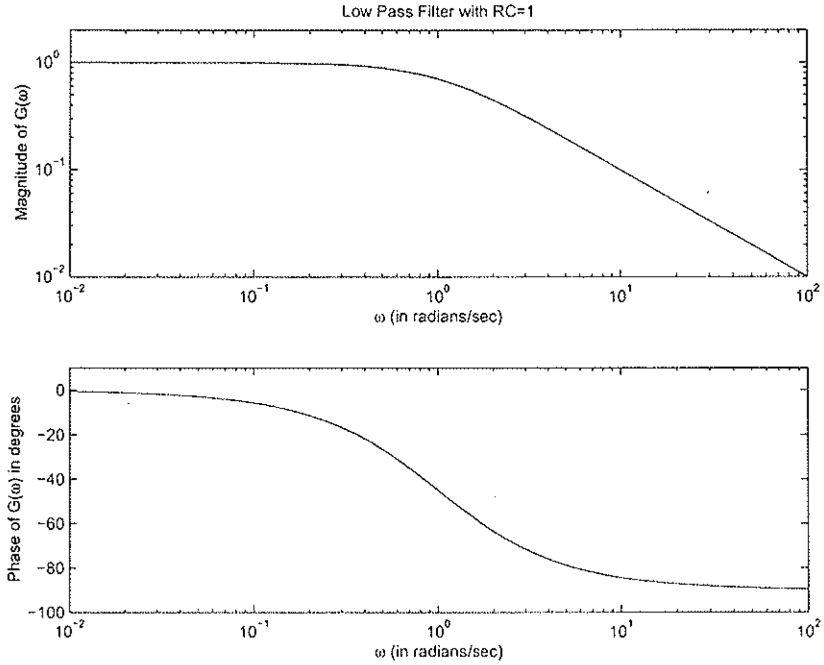


Figure 3.9: Low pass filter Bode plot.

	$\omega << \omega_c$	$\omega = \omega_c$	$\omega >> \omega_c$
zero: $(1 + j\omega/\omega_c)$	1	$(1 + j)$	$j\omega/\omega_c$
pole: $1/(1 + j\omega/\omega_c)$	1	$1/(1 + j)$	$\omega_c/(j\omega)$

Table 3.2: Asymptotic behavior of poles and zeros.

roots of the numerator and the poles are the roots of the denominator. In this expression the roots are actually imaginary. The leading factor of $(j\omega)^n$ (n is an integer) accounts for poles ($n < 0$) and zeroes ($n > 0$) at the origin. The poles and zeros can also be collectively referred to as corner frequencies, ω_c. The low pass filter has one pole at $1/(RC)$ and the high pass filter has one zero at the origin ($n = 1$) and one pole at $1/(RC)$. The factors of G (poles and zeros, etc.) become the terms of $\log |G|$.

$$\log |G(\omega)| = \log K + n \log \omega + \sum_{i=1}^{N} \log |1 + j\omega/z_i| - \sum_{i=1}^{M} \log |1 + j\omega/p_i| \tag{3.53}$$

On a log-log scale the factors of G simply add. The Bode plot for the magnitude of G can be obtained graphically by summing the plots for each term.

The poles and zeros are different for each circuit but once the transfer function is in this "pole-zero form" (equation 3.52) it is relatively easy to quickly sketch its Bode plot. The high and low frequency limiting values of poles and zeroes are shown in table 3.2, along with the values at the corner frequency (i.e. at a frequency ω_c equal to the pole or zero).

On a log-log plot ($\log |G|$ vs. $\log \omega$ or G_{dB} vs. $\log \omega$) both the pole and zero look like a straight line at high and low frequency. In a Bode plot the small curvature near the corner frequency (see the graphs given earlier for the high pass and low pass filter) is usually ignored. The high and low frequency asymptotes (straight lines) are simply drawn until they meet (at a corner). This makes is easy to quickly draw the Bode plot by hand with pencil and paper. At the corner frequency the actual response deviates from the straight line asymptote by a factor of $\sqrt{2}$ or 3 db (which is assumed but not usually drawn). On a log-log scale equal multiplicative factors appear as equal intervals. At a factor of 10X (in frequency) above or below the corner frequency (pole or zero) each pole or zero in the transfer function reaches its limiting value. The Bode plot for a simple transfer function with a single pole (on left) or zero (on right) is shown in figure 3.10. The phase of G is shown below the magnitude of G (both are vs. $\log\omega$). It may be helpful to visualize a tent pole to remember the shape of a pole in a Bode plot.

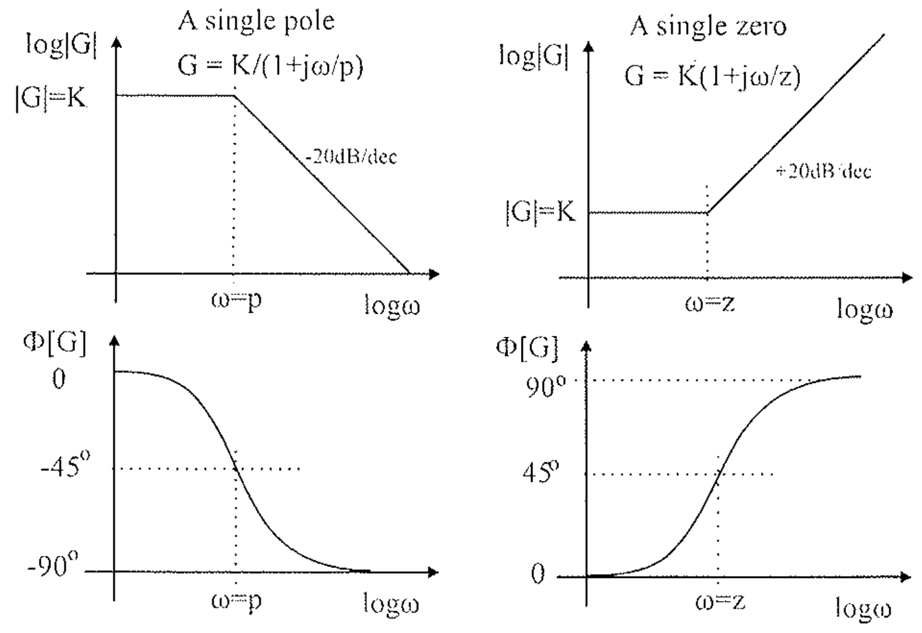


Figure 3.10: Bode plots of single pole and single zero.

To graph a complicated transfer function with many corner frequencies (poles and/or zeros), mentally sweep through all frequencies starting at very low frequency. At very low frequency $G \doteq K(j\omega)^n$. If $n > 0$ then G is rising at n(+20dB/decade). If $n < 0$ then G is falling with a slope of $|n|$(-20dB/decade). If $n = 0$ then $G = K$ with a slope of zero. As you mentally sweep through increasing frequencies, each time you pass a zero the slope of the transfer function changes by +20dB/decade and each time you pass a pole the slope changes by -20dB/dec. If the poles and zeroes are a factor of 10 or more apart then a graph of the transfer function is a sequence of straight line segments that bend up at each zero and down at each pole. (Poles and zeros closer than a factor of 10 in frequency require a more detailed analysis.) Another way to think of this graphing procedure is to count the number of zeros for which $\omega > z_x$ and the number of poles for which $\omega > p_x$. Then $G \propto (j\omega)^{n+N_z-N_p}$, where N_z =number of zeros and N_p =number of poles whose frequency is below the current frequency. Each additional power of ω gives an increase of 20db/dec in the slope of G_{dB}. You just have to count the number of active zeros and subtract the number of active poles to find the slope of G (active meaning $\omega >> \omega_c$).

For example the Bode plot in figure 3.11 is of a transfer function with a single zero and a single pole. At frequencies lower than both the pole and zero $G = K$ is a constant (zero slope). As frequency increases past the zero ($\omega = z$) its slope increases to +20dB/dec. When frequency increases further to the pole ($\omega = p$) the slope of G decreases by 20dB/dec. At high frequency the pole and zero cancel and G is again a constant with a slope of 0.

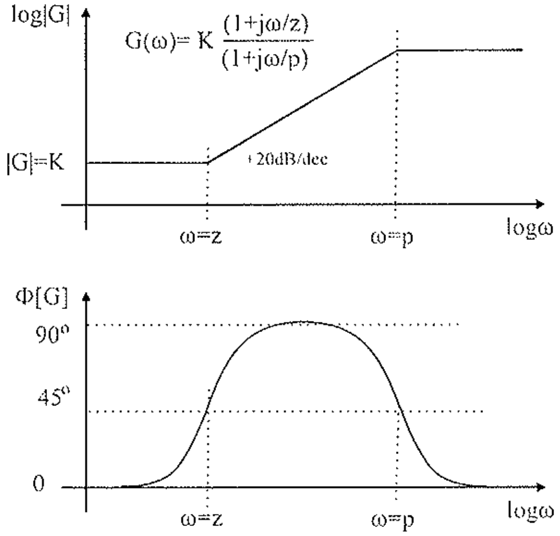


Figure 3.11: Bode plot with both a pole and a zero.

What Bode realized was that the slope of the magnitude of G is coupled to the phase of G because the factors of ω determine the slope and the factors of j determine the phase. The equations that determine the behavior of the circuit involve differentiation or integration with respect to time. In the sinusoidal steady state j and ω always appear together in $e^{j\omega t}$. This means that every +20dB/dec slope in G_{dB} produces a $+90^o$ phase shift (or $\pi/2$ radians) and every -20dB/dec slope in G_{dB} produces a -90^o phase shift. So once you know the graph for the magnitude of G, you can quickly sketch the phase of G as shown above.

The *passband* of a filter is that frequency region for which the gain is independent of frequency. Often, $G = 1$ in the passband. The regions where G varies steeply with frequency are called the *cutoff* regions.

Notice that each of the high pass and low pass filters has zero phase shift in their passband, but a phase shift of $\pm 90^o$ well into the cutoff region. A 90^o phase *advance* goes with a 20 dB/decade slope that is maintained over a wide region. A 90^o phase *lag* similarly goes with a slope of -20 dB/decade. These relationships between the phase and the amplitude are characteristic of a very large group of networks known as *minimum phase-shift (MPS) networks*. There exist networks with an excess phase *lag* over the minimum just discussed. They involve distributed parameters (e.g. a coaxial cable), transport phenomena (e.g., a light-beam signaling link), or bridge circuits. However MPS circuits are the main interest for now. A universal and painful consequence arises whenever a network's gain falls off at high frequencies (and they all do, if you go high enough). An inevitable phase lag accompanies the action. This phase lag can be very troublesome in negative-feedback circuits (chapter 4).

3.12 A Bandpass Filter

Consider the filter shown in figure 3.12. It has both a series combination (R_1 and C_1) and a parallel combination (R_2 and C_2).

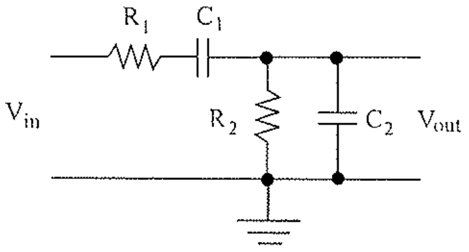


Figure 3.12: Band pass filter.

The presence of C_1 in series with the signal path must cause a cutoff at very low frequencies. Similarly, at very high frequencies, the fact that C_2 shunts the output signal must cause another cutoff. With suitable component values there exists a medium frequency region between these two cutoffs where the impedance of C_1 is negligible, and the impedance of C_2 is very large. In this intermediate region the filter acts as if it contains only R_1 and R_2, i.e., like a resistive voltage divider. This type of transfer function is sketched in figure 3.13, using a straight-line Bode plot. The filter thus has a passband defined by the corners at ω_1 and ω_2. This filter can be analysis graphically by considering each of three frequency regions, low, high and middle in approximate form.

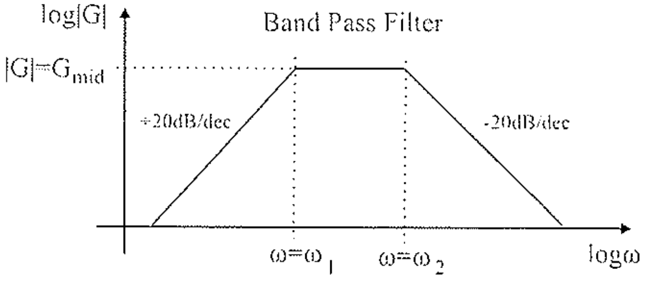


Figure 3.13: Band pass filter Bode plot.

(1) Low-Frequency Asymptote (figure 3.14): ($\omega < \omega_1$) At low frequency the impedance of both capacitors is much larger than the resistors. C_2 is negligible relative to the parallel R_2, and R_1 is negligible relative to the series C_1. What remains is a high-pass combination formed by C_1 and R_2. The asymptote is therefore a straight line at 20 dB/decade, intersecting the level of 0 dB at the corner frequency $\omega_{c1} = 1/(R_2C_1)$

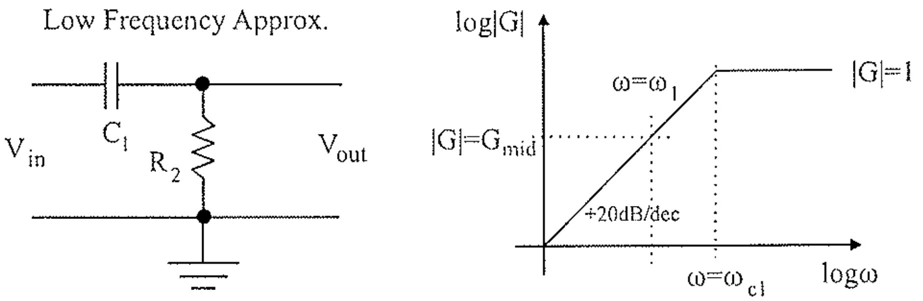


Figure 3.14: Low frequency approximation to the band pass filter.

(2) Midband: ($\omega_1 < \omega < \omega_2$) By assumption, the capacitors have negligible effect and the resulting resistive voltage divider yields:

$$G_{mid} = \frac{R_2}{R_1 + R_2} \tag{3.54}$$

(3) High-Frequency Asymptote (figure 3.15): ($\omega > \omega_2$) At high frequency the impedances of the capacitors are now very low, so that C_1 and R_2 are negligible. The result is a low-pass filter consisting of R_1 and C_2, whose Bode plot falls at -20 dB/decade from 0-dB with a corner frequency of $\omega_{c2} = 1/(R_1C_2)$

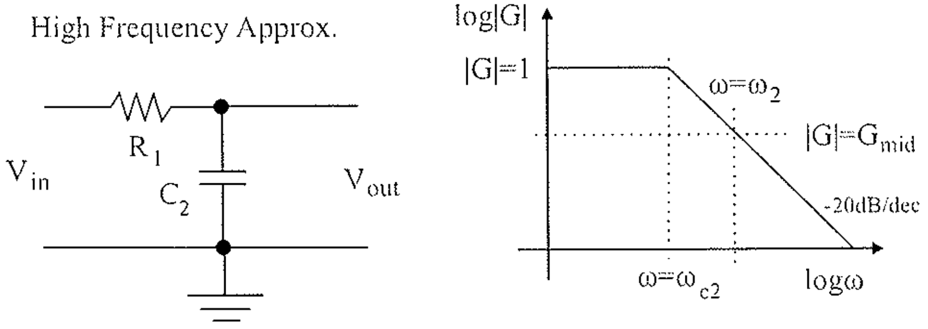


Figure 3.15: High frequency approximation to the band pass filter.

The sloping asymptotic lines for G at high and low frequencies intersect the level of G_{mid} before reaching the corner frequencies above. The resulting new corners lie at ω_1 and ω_2, which are readily calculated from the slope of G ($G \propto \omega$ at low frequency and $G \propto 1/\omega$ at high frequency).

Exp. 3.3 Design a band-pass filter of the type just discussed. Choose R_1 = 10K, arbitrarily, and then aim for the following specifications: $G_{mid} = 1/2$, $\omega_1 = 10^3$ rad/s, $\omega_2 = 10^5$ rad/s. (Note that ω_{c1} is NOT the same as ω_1 and ω_{C2} is NOT the same as ω_2.)

On semilog (G in db) or loglog paper, draw horizontal lines corresponding to 0 dB (G=1) and to G_{mid}. Draw the lower freq. and upper freq. asymptotes through the specified frequencies at G_{mid}, then extrapolate to 0 dB to find ω_{c1} and ω_{c2}. This determines the design of the filter.

Assemble the circuit with approximately the component values you have calculated, and measure G at several frequencies to see that it performs as expected (remember to use sine waves). Make a Bode plot (mag. and phase) of the performance of this filter. end

This example should not leave you with the false impression that every complicated network can be brought to its knees as easily as this, or that the results you obtain are necessarily watertight. It all depends on starting from a correct visualization of the circuit's action (how many segments to postulate for the plots, what approximations to make). Practice and experience do help.

An analytical derivation of the transfer function of the band pass filter starts with an expression using generalized impedance elements as:

$$G(\omega) = \frac{Z_2}{Z_1 + Z_2} \tag{3.55}$$

where Z_1 is the series combination of R_1 and C_1 and Z_2 is the parallel combination of R_2 and C_2 (yes this is just the voltage divider equation again).

$$\begin{aligned} Z_1 &= R_1 + \frac{1}{j\omega C_1} && (3.56) \\ Z_2 &= \frac{R_2 \frac{1}{j\omega C_2}}{R_2 + \frac{1}{j\omega C_2}} = \frac{R_2}{1 + j\omega R_2 C_2} && (3.57) \end{aligned}$$

After a bit of algebra the expression for the transfer function becomes:

$$G(\omega) = \frac{j\omega R_2 C_1}{j\omega R_2 C_1 + (1 + j\omega R_1 C_1)(1 + j\omega R_2 C_2)} \tag{3.58}$$

Unfortunately this is not in the preferred pole-zero form. However, with the assumption that $R_1C_1 >> R_2C_2$ an approximate transfer function for low and high frequency ranges can be obtained. At low frequency ($\omega R_1 C_1 \geq 1$ and $\omega R_2 C_2 << 1$)

$$G_{low}(\omega) \sim \frac{j\omega R_2 C_1}{j\omega R_2 C_1 + (1 + j\omega R_1 C_1)} = \left(\frac{R_2}{R_1 + R_2}\right) \frac{j\omega (R_1 + R_2) C_1}{1 + j\omega (R_1 + R_2) C_1} \tag{3.59}$$

This is just a high pass filter with a leading factor. Similarly at high frequency ($\omega R_1 C_1 >> 1$ and $\omega R_2 C_2 \geq 1$):

$$G_{hi}(\omega) \sim \frac{j\omega R_2 C_1}{j\omega R_2 C_1 + (j\omega R_1 C_1)(1 + j\omega R_2 C_2)} = \left(\frac{R_2}{R_1 + R_2}\right) \frac{1}{1 + j\omega R_3 C_2} \tag{3.60}$$

where $R_3 = R_1 // R_2$ is the parallel combination of R_1 and R_2. This is just a low pass filter with the same leading factor. Both of these analytical expressions are consistent with the graphical analysis done earlier.

3.13 Computer Aided Design and Analysis

The transfer functions for many different types of filters can be put into a pole zero form and analyzed with pencil and paper (as above). However there are some transfer functions that do not easily factor (such as the band pass filter). There are also a variety of computer software packages that can help in analyzing complicated filters (and other electronic circuits) that do not easily factor. Two software examples of making Bode plot are illustrated next. Although this software will not be used in this lab, it is available on many other computer on campus that are available for your use. The computer can produce Bode plots for arbitrarily complicated filters. This can be very helpful at times, however note that the computer will provide only the final graph. It will not provide any direct understanding of how the circuit works nor will it give you any insight into how you might design or modify a circuit to perform a given function (some new discoveries may be made by simple trial and error using the computer though). You should learn to use computer aided design tools but do not become dependent on them to do your thinking for you.

The commercial program package called MATLAB (www.mathworks.com) is common in many scientific and engineering settings (octave, www.octave.org, is a free alternative to matlab, with nearly the same syntax). It is primarily a general purpose numerical analysis and graphical package and can do many different types of operations. You program it by typing a sequence of commands at the command line or by executing a set of commands in a file. The following script produces a Bode plot of the Band pass filter (with different component values than used in exp.3.3). Please refer to the MATLAB documentation for a description of what each command is. The resulting graph is shown in figure 3.16.

```
%
%  Matlab script to make Bode plot of band pass filter
%
clf;
clear;
n = 100;                    % number of points
w = logspace( 0, +8, n );   % frequency of 1e0 to 1e+8
R1 = 1.0e4; R2 = 1.0e3;     % component values
C1 = 0.47E-6; C2 = 0.001E-6;
%
%  the transfer function G(w)
g = j*w*R2*C1./(j*w*R2*C1 + (1+j*w*R1*C1).*(1+j*w*R2*C2));
%
subplot(211);                %  plot magnitude on top graph
loglog( w, abs(g) );
axis( [1.0e0 1.0e8 1.0e-3 0.2 ]);
xlabel( '\omega (in radians/sec)' );
ylabel( 'Magnitude of G(\omega)' );
title( 'Band Pass Filter using Matlab' );
%
subplot(212);                % plot phase on bottom
semilogx( w, 180 *angle(g)/pi );
axis( [1.0e0 1.0e8 -100 100 ]);
xlabel( '\omega (in radians/sec)' );
ylabel( 'Phase of G(\omega) in degrees' );
%
print -deps bodeband.eps   %  print postscripit to file
```

The second software example is SCILAB (www.scilab.org or www-rocq.inria.fr/scilab/). It is very similar in function to MATLAB but it is free (you can download a copy for yourself and run it on you computer at home if you wish). SCILAB comes with signal processing and systems and control analysis tools built in (you have to pay extra for these in MATLAB). A sample script is shown below (comments

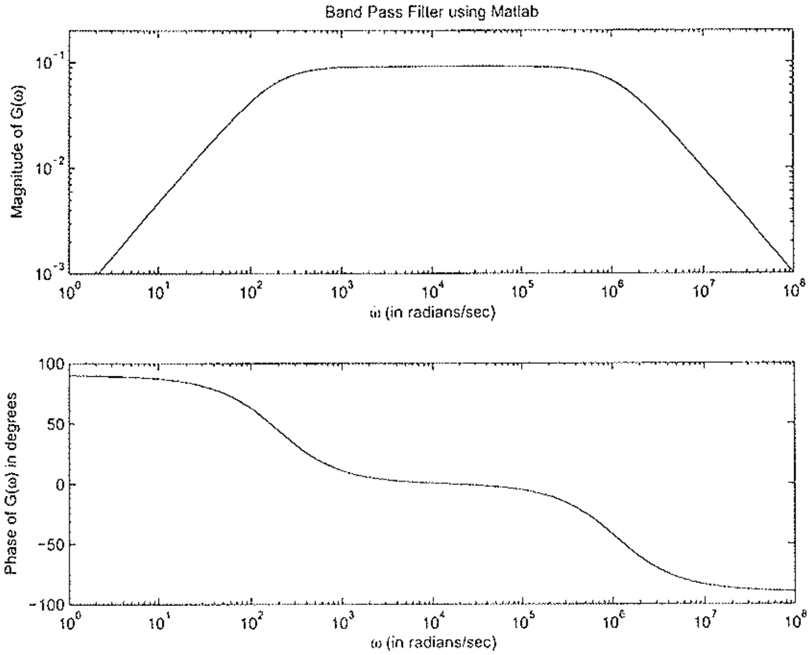


Figure 3.16: A Bode Plot of the band pass filter using MATLAB software (not the same component values as in exp. 3.3).

begin with a double backslash). You simply define a linear system to be the transfer function and then call the bode function which does all of the work. Please refer to the SCILAB documentation for more details. The resulting Bode plot is shown in figure 3.17.

```
//
//  scilab script to make Bode plot of bandpass filter
//  to run type exec('bodeband.sci');
//
s = poly(0,'s');             // define symbolic variable s=jw
R1 = 1.0e4; R2 = 1.0e3;      // component values
C1 = 0.47E-6; C2 = 0.001E-6;
g = syslin('c', s*R2*C1/(s*R2*C1 + (1+s*R1*C1)*(1+s*R2*C2)) );
bode( g, 1.0e-1, 5.0e7);  // plot over freq of 0.1 to 5x10^7 Hz
```

3.14 Fourier Analysis. The Frequency Domain

It is well known, in musical acoustics, that a periodic signal of arbitrary waveform can be analyzed into a series of *harmonic* components, each of which is a pure sine wave. The first harmonic, or fundamental, has a period equal to that of the signal itself. The other harmonics have frequencies that are integral multiples of the fundamental frequency. For example, if an oboe plays an A of 440 Hz, there are in its acoustic output components at 440 Hz, 880 Hz, 1320 Hz, etc. The relative amplitudes

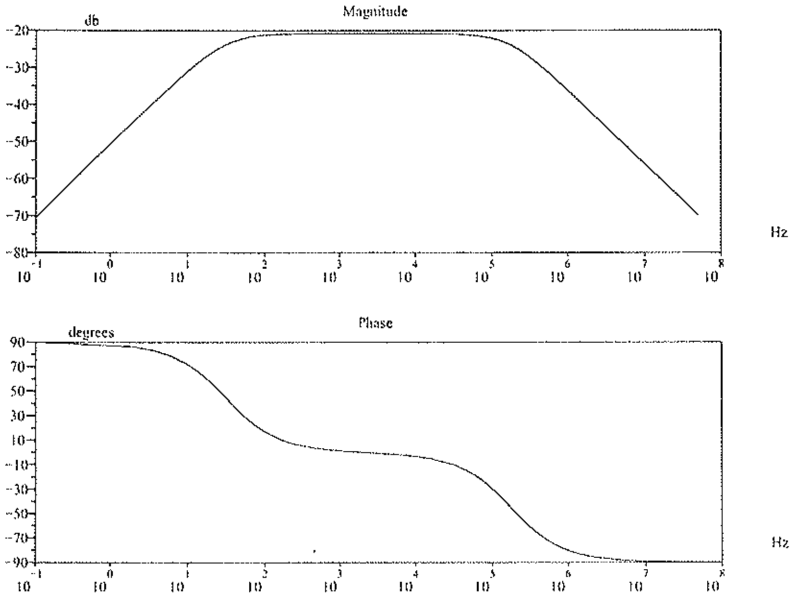


Figure 3.17: A Bode Plot of the band pass filter using SCILAB software (not the same component values as in exp. 3.3).

of these components depend on the exact wave shape of the oboe's sound. In fact, the sound quality is sometimes described by specifying the harmonic content.

Nonperiodic as well as periodic waveforms can be resolved into sine-wave components. One simply allows the fundamental period to become arbitrarily long. In the limit of this process, the fundamental frequency tends to zero and the harmonics that were previously a set of discrete frequencies merge into a continuous frequency *spectrum*. The process of decomposing a given signal wave shape into its sine-wave components is called *Fourier analysis*. (Fourier worked out the mathematics to formalize the procedure.) Without going into all of the details here, it suffices to say that any signal can be expressed as the superposition of different sine waves. This allows a different way of specifying a signal by stating the complex amplitude $A(\omega)$ of its Fourier components as a function of frequency. This description of a signal by its spectral distribution is completely equivalent to the more usual specification of the signal's direct time dependence, v(t). The latter description is in the *time domain*, the former in the *frequency domain*.

As shown earlier in this chapter it is relatively easy to work out a circuit's response to a sine-wave signal. In principle, any arbitrary signal can be converted into its Fourier components. If the circuit is linear (as are the filters discussed so far) each Fourier component is acted on separately by the circuit, and then combined at the output to obtain the resulting (modified) waveform. Each Fourier component of the signal, $A(\omega_n)$ is multiplied by the transfer function of the circuit for that particular frequency $G(\omega_n)$. This procedure can be mathematically tedious, but can provide some general insights.

When a network (for example, a hi-fi amplifier) is to pass a signal without significant waveform distortion, it is sufficient to require that all sine-wave components present in the signal be treated equally. In other words, the network's transfer function must be *flat* as a function of frequency over the required range of frequencies[1]. This is why audio amplifiers are specified to be "flat" to within 1 dB, for example, over the range from 20 Hz to 20 kHz (the range of the human ear). An idea of the frequency range covered by the spectrum of a given signal is all that is required to specify the frequency response of the amplifier.

Two rough guidelines can be worked out intuitively:

(1) The lowest frequency of importance is determined by the duration of the longest steady part of the signal. Consider, for example, a single rectangular pulse of duration T as shown in figure 3.18. Its Fourier spectrum extends from zero frequency all the way to infinity, but for a moment, imagine that it is related to a continuous square wave of period $2T$, whose lowest Fourier component is at frequency $f_1 = 1/(2T)$. It is plausible that the rectangular pulse will be transmitted recognizably by a network whose transmission (G is constant) extends at least down to f_1. (Depending on just how fast the transfer function cuts off below f_1, the pulse will suffer more or less drastic distortion.)

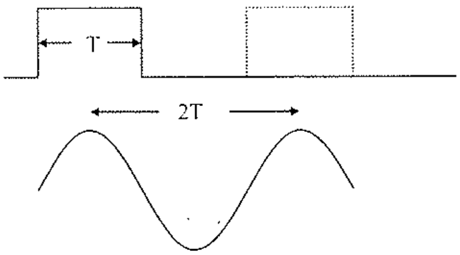


Figure 3.18: Low frequency components.

(2) The highest frequency of importance is determined by the shortest time during which any part of the signal changes. Suppose a pulse "switches" in a finite time t_r as shown in figure 3.19. Then, comparing it to a zig-zag wave of total time $4t_r$, it is plausible to think that the highest frequency of importance will be $f_2 = 1/(4t_r)$.

Although these guidelines only determine the order of magnitude of the frequency limits, the basic idea is valid. Low-frequency components yield long-maintained signals, and high-frequency components yield short rise and fall times (times of change).

3.15 R-C Networks in the Frequency Domain

The transfer functions of the high-pass and low-pass R-C networks were worked out earlier. In each case there is a passband ($\omega >> \omega_c$ for the high pass filter and $\omega << \omega_c$ for the low pass filter) in which $G = 0$ dB and $\phi = 0^o$. If the spectrum of a signal lies entirely within this passband then it is transmitted without significant distortion. On the other hand, in the cutoff region ($\omega << \omega_c$ for the

[1] The magnitude of $G(\omega)$ should be constant, but its phase can be a linear function of frequency. Such a linear phase shift corresponds to a simple *time delay* of the signal, each sine-wave component being delayed by the same amount of time.

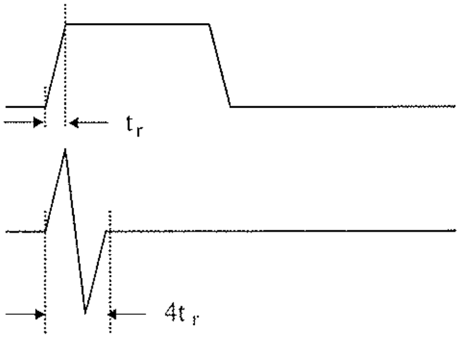


Figure 3.19: High frequency components.

high pass filter and $\omega >> \omega_c$ for the low pass filter) G varies as ± 20 dB/decade, and there is a $\pm 90^o$ phase shift. Signals whose spectrum lies in this cutoff region can be significantly distorted.

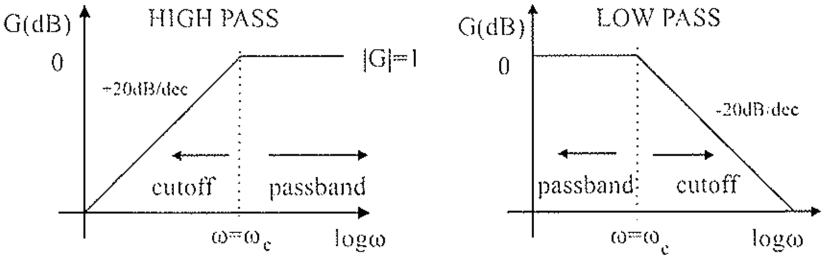


Figure 3.20: High and low pass filter frequency response.

In the frequency domain, the operation of time differentiation is equivalent to multiplication by the factor $j\omega$, i.e., to a transfer function that is proportional to frequency, and to a 90^o phase advance. This is just what the high-pass filter does in its cutoff region. Therefore, for signals with spectra falling in the cutoff region, the high pass filter acts as a *differentiator*. Similarly, the operation of time integration is equivalent to division by $j\omega$, which is what the low-pass filter does in its cutoff region. Thus a low pass filter *integrates* signals whose spectrum lies within the cutoff region.

Each filter thus has two modes of operation, depending on the placement of the signal spectrum relative to the corner frequency $\omega_c = 1/(RC)$. In the passband the signal is unchanged. In the cutoff region it is differentiated or integrated. Another way of specifying that the signal spectrum lies in the cutoff region is to say that the filter attenuates the signal greatly. *If the output is small compared to the input, the filter differentiates (or integrates).*

Differentiation and integration of the signal is sometimes a desirable operation. For example, suppose the signal indicates the position of a cutting tool on a milling machine, and you want to find the *speed* at which the tool is being advanced. Then differentiation of the position signal gives you the speed. Alternately, if you are given a signal proportional to the X-ray *intensity* in a test chamber, and you need to find the total dose given to a specimen there. Then you would need to integrate the intensity signal to obtain the total dose. The performance of R-C filters in their cutoff region, where they approximate these desired operations, is of practical interest.

Actually, the frequency-domain view gives a somewhat more general perspective, allowing the following general statements. Any network whose transfer function rises at 20 dB/decade, accompanied by a 90^o phase advance, performs as a differentiator for signals whose Fourier spectrum falls into that region. Any network whose transfer function falls at 20 dB/decade, with a 90^o phase lag, similarly performs as integrator.

3.16 Differentiation and Integration–the Time-Domain View

If the output voltage of the high-pass filter is very small compared to the input (the condition for accurate differentiation formulated previously) then there is a simple way of analyzing the circuit directly (refer to figure 3.21).

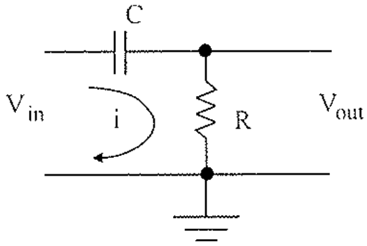


Figure 3.21: High Pass filter.

If v_{out} is very small the voltage across C is just v_{in}, and hence the AC current i through C is:

$$i = C\frac{dv_{in}}{dt} \quad \text{if} \quad v_{out} << v_{in} \tag{3.61}$$

Then, for v_{out} :

$$v_{out} = iR = RC\frac{dv_{in}}{dt} \tag{3.62}$$

Evidently the high-pass filter differentiates! You can see where the situation breaks down. As soon as v_{out} ceases to be negligible, the voltage across C is $(v_{in} - v_{out})$, not just v_{in} itself.

A similar argument can be applied to the low-pass filter. When $v_{out} << v_{in}$, the voltage across R is v_{in}, and $i = v_{in}/R$. Then:

$$v_{out} = \frac{q}{C} = \frac{1}{C}\int_0^t idt + Q(0) = \frac{1}{RC}\int_0^t v_{in}dt + v_{out}(0) \tag{3.63}$$

where $v_{out}(0) = q(0)/C$ is the output voltage at time $t = 0$.

It is instructive to examine the output of the low-pass filter in the light of this discussion (see figure 3.22). Suppose that a step-function input has been applied at $t = 0$ with the capacitor initially uncharged. The ideal integral of a step-function is a linear ramp, which is indicated by the dotted line. At first v_{out} follows this ideal output. But then, as soon as v_{out} ceases to be very small compared to v_{in}, it veers away and falls below the ideal curve. Of course it is clear that the ideal integral grows

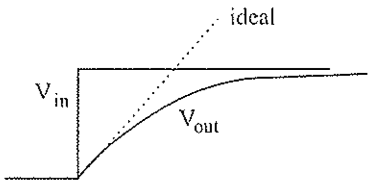


Figure 3.22: Low pass edge.

without bound, if you are willing to wait long enough. No practical circuit can hope to rise to such a challenge.

The restriction that *the output be negligible* is irksome in practice. That's the main reason why the operational amplifiers is used to improve the performance of R-C differentiating and integrating networks. This is the topic of the next chapter.

3.17 A Note on Dimensions

It is useful to keep an eye on the *dimensions* of all of the mathematical results. Rewrite the equations for the high-pass and low-pass filters (for $v_{out} << v_{in}$) using the time constant $\tau = RC$ which has units of time:

$$\text{high-pass:} \qquad v_{out} = \tau \frac{dv_{in}}{dt} \tag{3.64}$$

$$\text{low-pass:} \qquad v_{out} = \frac{1}{\tau} \int_0^t v_{in} dt + v_{out}(0) \tag{3.65}$$

Then the appearance of τ and dt in numerator and denominator makes it clear that the dimension of time cancels out, leaving only the voltage v_{in} on the right-hand side of the equations. All is well.

3.18 Practice Problems

[1] Derive an analytical expression for the total complex impedance of the circuit shown in figure 3.23. This circuit is frequently used to tune (or select) a specific radio frequency in a radio or television, because it has a resonance at a particular frequency. What is the impedance at $\omega = 1/\sqrt{LC}$?

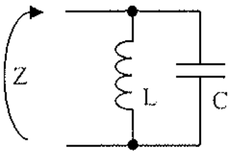


Figure 3.23: LC tuning circuit.

[2] Design a 10X probe for an oscilloscope (see schematic figure 3.24). The input of the scope has

some *parasitic capacitance* (i.e. unwanted but unavoidable) C_1 from its input amplifiers. This has the effect of loading the circuit being measured and makes the response of the scope vary with frequency. The 10X probes used in the lab use a circuit like that shown below to reduce the loading and frequency dependence of the scope measurements.

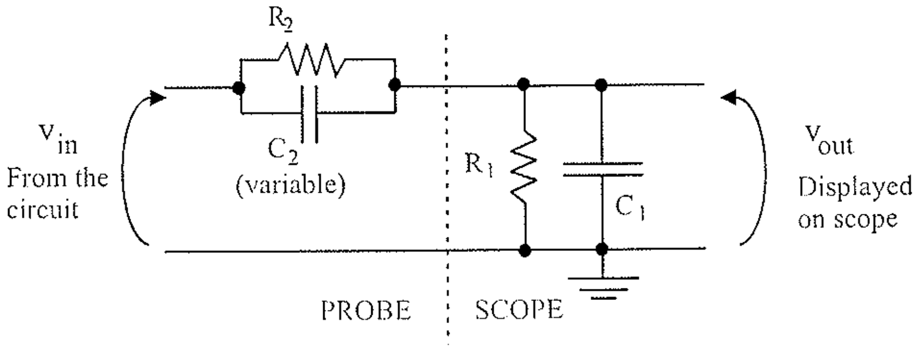


Figure 3.24: Scope probe compensation.

a) Show that $R_1C_1 = R_2C_2$ makes the total response independent of frequency. First find an analytical expression for v_{out}/v_{in} in terms of R_1, R_2, C_1, C_2. Then from this expression find another expression that makes v_{out}/v_{in} independent of frequency. These process is called compensation.

b) If R_1=1 MegΩ and C_1 = 40 pfd are inside the scope, find values for R_2 and C_2 (inside the probe) to produce a 20db or 10X attenuation. Assume $R_1C_1 = R_2C_2$ as in part a.

c) Find the equivalent load on the circuit from the combination of the probe and the scope. When connected to a circuit the combined probe plus scope looks like the parallel combination of a single resistor R_{eq} and a single capacitor C_{eq} if the expression given in a) is true. Find R_{eq} and C_{eq} using the values given in b). Assume $R_1C_1 = R_2C_2$ as in part a. (Hint write an expression for the total impedance Z_{eq} of each circuit, substitute the expression in part a and compare.)

[3] Figure 3.25 shows several possible filters with two resistors and one capacitor, R_1= 100 K. R_2= 1 K and C=680 pfd. Do the following for each of these filters:
i) Derive an analytical expression for $G(\omega) = v_{out}/v_{in}$ and put it in pole-zero form.
ii) Make a Bode plot (amplitude and phase) of $G(\omega)$. Label all asymptotic slopes and values.
iii) What is the corner frequency (or frequencies) (in Hz)?

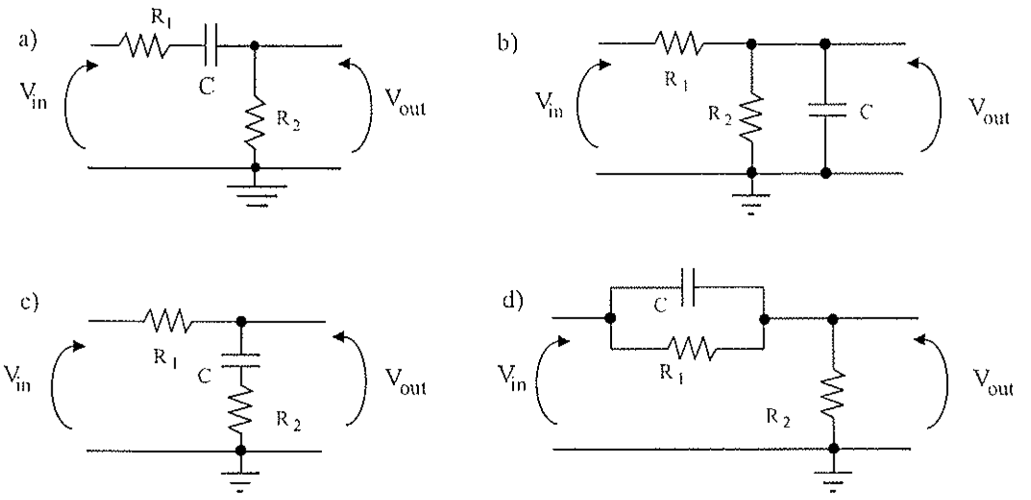


Figure 3.25: Problem 3.4.

Chapter 4

NEGATIVE FEEDBACK CIRCUITS

4.1 General Principles

Walking a straight line with your eyes open is not difficult. However, walking it blindfolded is an entirely different proposition. In the first case you are functioning as a *self-checking* and *self-correcting machine*. In the second case, you are attempting the same feat by *dead-reckoning*. A very similar distinction exists for an amplifier. If you merely apply the input signal and accept what appears at the output, the results are at the mercy of whatever nonlinearity, frequency dependence, or performance drift the amplifier happens to be afflicted with. If you build a self-checking system, however, you can correct for such defects.

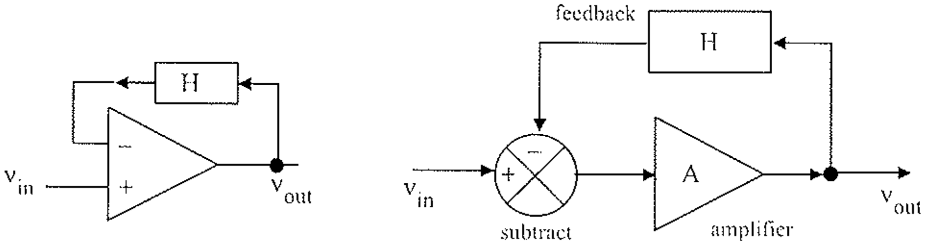


Figure 4.1: One form of negative feedback using an op-amp (left) and a signal flow diagram for negative feedback (right).

A self-checking amplifier is obtained by applying *negative feedback*. Part of the amplifier's output is returned back to its input, with such polarity as to subtract from the input signal. There are several negative-feedback configurations. The one shown in figure 4.1 uses the two input terminals of an op amp (a difference amplifier) to subtract the feedback signal from the input signal. The box labeled H is an external electronics network (such as a voltage divider) that takes some portion (labeled H) of v_{out} and returns it to the inverting input ($v_{inv} = HV_{out}$). A more formal abstract diagram for negative feedback is shown on the right of figure 4.1 .

There are two different gains that appear in this circuit. One is the gain of the op-amp by itself, which will be referred to as the open loop gain G_{OL} and the net gain of this circuit which will be referred to as the closed loop gain $G_{CL} = v_{out}/v_{in}$. The output voltage of the op-amp is related to the differential input voltage as:

$$v_{out} = G_{OL} v_{diff} \tag{4.1}$$

where the differential input voltage v_{diff} is:

$$v_{diff} = v_{non} - v_{inv} = v_{non} - Hv_{out} \tag{4.2}$$

Combining these two expressions yields:

$$v_{out} = G_{OL}(v_{non} - Hv_{out}) \tag{4.3}$$

However, in this configuration $v_{non} = v_{in}$, so the closed loop gain is:

$$G_{CL} = \frac{v_{out}}{v_{in}} = \frac{G_{OL}}{1 + HG_{OL}} = \frac{1}{H}\left[\frac{HG_{OL}}{1 + HG_{OL}}\right] \tag{4.4}$$

A first glance this appears to make the situation worse. The open loop gain is not a well- determined parameter. Its value can vary by a factor of five or ten between different op-amps of the same type. However, even though its value is not known precisely, it is always a large valve. When the loop gain satisfies $|HG_{OL}| >> 1$ the expression for the closed loop gain is approximately:

$$G_{CL} \sim \frac{1}{H} \tag{4.5}$$

This approximation can be called the infinite gain approximation. Note however, that it is the product HG_{OL} that must be large not just the op-amp gain. H is determined by an external circuit (such as a voltage divider) and is well controlled. This means that the closed loop gain is also well controlled (independent of the precise value of the open loop gain of the op-amp) and can be designed to have a precise value. $G_{OL}H$ is called the *loop gain* of the system, from the picture of a signal circulating around the loop formed by G_{OL} and H. (The name of this approximation might thus better be called infinite *loop-gain* approximation.)[1]

The equation for v_{out} can also be rearranged as:

$$\begin{aligned} v_{out} &= \frac{v_{inv}}{H} = G_{OL} v_{diff} = G_{OL}(v_{non} - v_{inv}) \qquad (4.7) \\ \frac{v_{inv}}{HG_{OL}} &= v_{non} - v_{inv} \sim 0 \qquad (4.8) \end{aligned}$$

Which results in another useful expression:

$$v_{non} = v_{inv} \tag{4.9}$$

if the open loop gain is nearly infinite. In this infinite-gain approximation, the effective difference signal at the input becomes negligibly small (provided only that the op-amp is in its active range).

[1] The symbols G and H are common in one branch of the business (servo mechanisms), but in others the gain is commonly called A and the feedback factor β. Moreover, the feedback signal is sometimes assumed to be added to v_{in}, rather than subtracted, so that the equation for the gain would take the form:

$$v_{out}/v_{in} = A/(1 - \beta A) \tag{4.6}$$

In the convention used here, the loop gain GH is positive; the feedback is negative because it is taken to the inverting input of the op amp.

This effective input can also be labeled the error signal, $v_{err} = v_{diff}$. With negative feedback the op-amp changes v_{out} to drive this error signal to zero. If v_{out} is too high then a portion of it is feed back to v_{diff} which drives v_{out} back down. The reverse is also true. If v_{out} is too low, then v_{diff} becomes larger and drives v_{out} up again. The only equilibrium condition is when v_{non} and v_{inv} are approximately the same. (In practice there is a small difference to maintain a non-zero v_{out} but the difference is of order $1/G_{OL}$ and it can be taken as zero in most calculations.)

The self-checking part of the circuit resides at the op amp input, where the feedback signal, Hv_{out}, is compared with the input, v_{in}. The feedback network, through which the checking takes place, is chosen such that H is the *inverse* of the desired overall gain, $G_{CL} = v_{out}/v_{in}$. This network normally consists of passive, linear, and stable components. Thus, H need not be subject to the nonlinearities and drifts which afflict an amplifier. The self-correcting comes about through the amplifier action because the feedback is taken to the inverting input terminal. Any deviation of v_{out} from the desired value produces an input in such a direction as to reduce the error.

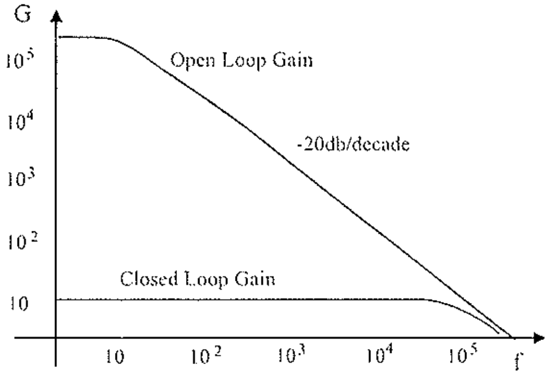


Figure 4.2: Magnitude of the Bode plot for an op-amp using negative feedback.

As a practical example, consider a type 741 op amp in a negative feedback loop with effective gain $G_{CL} = 10$. This could be obtained with a resistive voltage divider to produce the required H =0.1. The open and closed loop gains are plotted in figure 4.2. At a frequency of 10 Hz, $G_{OL} = 10^5$, the loop gain $G_{OL}H = 10^4$, and G_{CL} deviates from $1/H$ by only one part in 10^4. On the other hand, at a frequency of 10 kHz G_{OL} has fallen to 10, $G_{OL}H$ =10, and G_{CL} is in error by about 10%. The log-log plot for G_{OL} and G_{CL} in figure 4.2 illustrates that the closed loop gain G_{CL} is much more nearly constant than the bare op-amp gain G_{OL}. However, this is achieved at the cost of sacrificing much of the gain. This is not a serious sacrifice, since the gain starts out very large. [2]

[2] The price to be paid for the use of negative feedback is more subtle. It lies in the design effort required to ensure that the system is stable. Self-correcting systems may have an equilibrium point, but there is no *a-priori* guarantee that this equilibrium will in fact be reached. The system may overshoot and even go into uncontrolled oscillation. This problem is discussed in section 4.15. In this connection it is worth noting, from the equation:

$$G_{CL} = G_{OL}/(1 + G_{OL}H) \tag{4.10}$$

That G_{CL} approaches the value $1/H$ even when G_{OL} is negative and large. However, this corresponds to positive feedback and does not lead to stable equilibrium.

If the logarithm of the equation for the closed loop gain:

$$G_{CL} = \frac{v_{out}}{v_{in}} = \frac{G_{OL}}{1 + HG_{OL}} \tag{4.11}$$

is differentiated, treating H as constant, this results in:

$$\begin{aligned} \log G_{CL} &= \log G_{OL} - \log(1 + HG_{OL}) && (4.12) \\ \frac{dG_{CL}}{G_{CL}} &= \frac{dG_{OL}}{G_{OL}} - H\frac{dG_{OL}}{1 + HG_{OL}} = \frac{dG_{OL}}{G_{OL}}\left(\frac{1}{1 + HG_{OL}}\right) && (4.13) \end{aligned}$$

If the open loop gain, G_{OL}, varies, for any reason at all, then the fractional change in the closed loop gain, G_{CL}, is smaller than the fractional change in G_{OL} by the gain reduction factor, $(1 + HG_{OL})$. This is the same factor by which G_{CL} is smaller than G_{OL}, due to the feedback. This reduced fractional variation of G_{CL} applies to all causes, whether the variation is with frequency, with signal amplitude (=nonlinearity), with ambient conditions (temperature, supply voltages), or even with load. Connecting an external load to the amplifier reduces its output voltage, because of the finite source impedance of the amplifier. This reduction is smaller, by the gain reduction factor, when negative feedback is present. Thus the output impedance is effectively reduced by the gain reduction factor. This result is easily understood because the output voltage is being supervised and corrected as necessary, thus making it seem to come from a more nearly perfect voltage source.

From here on the subscripts OL and CL for the open loop and closed loop gain will be dropped. The distinction will usually be clear from the context.

4.2 Differential and Common-Mode Signals

In the negative-feedback configuration just considered, the op amp receives signals at its inverting and noninverting inputs, and is challenged to amplify only the difference signal, which plays the role of error voltage, v_{err}. Meanwhile the two inputs may actually swing over a considerable voltage range in common. To describe the action of a difference amplifier it is useful to develop the following terminology.

Signal components appearing in equal amounts at both terminals are called *common-mode signals*, v_{com}. The signal components appearing as equal but opposite changes form a *differential signal* v_{diff}. Two arbitrary voltages, v_{inv} and v_{non}, applied to the two inputs can be resolved into these components:

$$v_{com} = \frac{1}{2}(v_{non} + v_{inv}) \qquad v_{non} = v_{com} + \frac{1}{2}v_{diff} \tag{4.14}$$

$$v_{diff} = v_{non} - v_{inv} \qquad v_{inv} = v_{com} - \frac{1}{2}v_{diff} \tag{4.15}$$

The common-mode component is the average of the two inputs, and the differential component is their difference.

An ideal difference amplifier is completely insensitive to v_{com}. However, practical amplifiers do respond to this common-mode component to a small extent. The ratio of the output from a small difference signal to the output from an equal common-mode signal is called the *common-mode rejection ratio (CMRR)* of the amplifier. Usually this ratio is as high as 10^4 (80 dB) or more, but even then the effect of the common-mode component may need to be taken into account in some applications.

Amplifiers are limited to a stated common-voltage range, beyond which they may be damaged or at least will produce curious outputs. They are also limited to a maximum difference voltage, which however is always so large (several volts) as to be dramatically beyond the active range of the amplifier. Under overload conditions in a negative feedback loop, the difference voltage may exceed the ratings and produce unexpected results. Sometimes the amplifier output "locks-up" in response and the system may freeze in a pathological condition.

4.3 The Voltage Follower

The simplest negative-feedback circuit of the type being considered uses a feedback network with H=1 (a direct connection) as shown in figure 4.3. Then $v_{out} = v_{in}$ meaning that the output voltage follows the input voltage, and the device is called a *voltage follower*. Its purpose is to transport a signal from a high-impedance source to a low-impedance load. Thus, although there is only unity voltage gain, there may be considerable *power gain* due to the increased load current (i.e. large current gain).

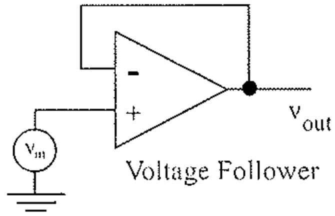


Figure 4.3: Voltage follower.

Exp. 4.1 Build a voltage follower, using a type 741 op amp with supply rails at ±15 V. Apply a sine-wave input at 1 kHz with 20 V_{p-p} amplitude and verify that $v_{out} = v_{in}$ within the error of measurement. Apply a square-wave input and measure and interpret the risetime of v_{out}.

Return to a sine-wave input and reduce the amplitude to 1 V_{p-p}. Then, while observing v_{out}, connect a 47 Ohm load resistor across the output (i.e from the output to ground) and estimate by how much (at most) v_{out} is reduced. Connect the 47 Ohm resistor directly across v_{in} to demonstrate that the voltage follower is doing some work. end

Had you attempted to drive the 47 Ohm load with the full ±10 V swing, the op amp would have been incapable of delivering the required maximum current. ±10 V into 47 Ohms corresponds to ±200mA. The 741 is limited to an output current of about 25mA. There is an important lesson in this. Although negative feedback lowers the output impedance and simulates an ideal voltage source *within the working range of the amplifier*, it cannot create output capabilities beyond those that were already there.

Exp. 4.2 Substitute a 3140 op amp in your voltage follower. Apply a square-wave input with 2 V p-p amplitude and no load resistor on the output. Measure the risetime of v_{out}.

OPTIONAL: Increase the amplitude of the input gradually and observe what happens to v_{out}. (Hint: this violates one of the maximum allowed conditions of the 3140 for a short period of time.) end

If you apply a large amplitude square wave to a 3140 voltage follower, strange effects can be produced at the output. These are caused by an overload of the differential signal ratings of the 3140. Since v_{out} cannot follow v_{in} exactly (it is limited by the slewing rate of the op amp), a large v_{in} causes momentary large differences between v_{in} and v_{out}, with unpleasant results.

4.4 Noninverting Amplifier

The schematic in figure 4.4 shows a resistive voltage divider as feedback network. This configuration produces an amplifier of "ideal" gain of:

$$G = \frac{1}{H} = \frac{R_1 + R_2}{R_1} \tag{4.16}$$

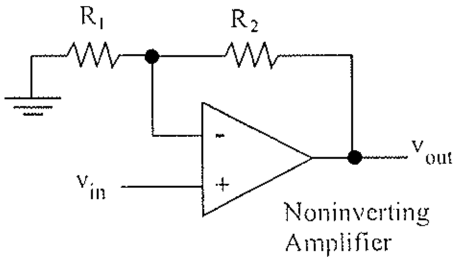


Figure 4.4: Noninverting configuration.

Exp. 4.3 Design and construct a noninverting amplifier with a voltage gain of G=30 (or as close as you can get with the available resistors) using a 741 op-amp driven with +/-15 volt supplies. Using a sine wave input, measure the voltage gain (magnitude and phase) at a frequency near 1kHz using a small enough input voltage such that the output is not distorted. What is the maximum input amplitude that can be used without distorting the output waveform? This circuit should have the same frequency response as the inverting configuration in the next experiment. To save time you do not have to investigate the frequency response of this circuit. end

4.5 The Inverting Configuration

The circuits used so far apply the input signal to v_{non} and result in noninverting amplifiers. Because the feedback has to go to v_{inv} to be negative, both amplifier inputs are "busy" and a common-mode input voltage equal to v_{in} is present.

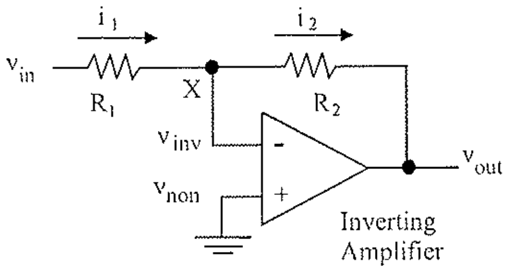


Figure 4.5: Inverting configuration.

You can also apply the signal to v_{inv} in the arrangement shown in figure 4.5. This produces an inverting amplifier and this configuration is also known as the operational configuration. v_{non} is freed and can be held at a constant potential, usually ground. The common-mode component is greatly reduced. The arrangement, known as the *operational configuration*, has many special features and it will be worth while to describe its action from several points of view. With v_{non} =0, the error signal is:

$$v_{err} = v_{non} - v_{inv} = -v_{inv} \tag{4.17}$$

Therefore, with the infinite-gain approximation $v_{inv} \sim 0$. Thus the circuit point marked X is almost exactly at ground potential, yet it isn't connected to ground. In fact, the amplifier's input draws a negligible current, so that X can be regarded as free-floating. What keeps it at ground potential is the effect of R_2 connected to v_{out}, which maneuvers so as to keep $v_{inv} = 0$ because of the negative feedback. Because of this special situation, X is known as a *virtual-ground* point.

Using the virtual ground concept, it follows directly that current entering through R_1 ($i = v_{in}/R_1$) must leave via R_2:

$$i_1 = \frac{v_{in} - v_{inv}}{R_1} = \frac{v_{in}}{R_1} = i_2 = \frac{v_{inv} - v_{out}}{R_2} = -\frac{v_{out}}{R_2} \tag{4.18}$$

This expression can be rearranged as:

$$G = \frac{v_{out}}{v_{in}} = -\frac{R_2}{R_1} \tag{4.19}$$

The input impedance of the operational configuration is evidently just R_1, since v_{in} must drive this element as if the other end were connected to ground.

Both the inverting and noninverting configurations should have the same frequency response as illustrated in figure 4.6. Each should produce a constant gain as calculated above for low frequencies. This gain should remain constant with increasing frequency until the open loop gain curve is crossed. The gain should then follow the open loop curve for further increases in frequency. The situation is further complicated for large amplitudes and high frequencies, where the output can also become slew-rate limited (output distorted). For small amplitude signals, the bandwidth may be calculated from the known unity gain bandwidth frequency at which $G = 1$, the slope of the open loop gain curve at high frequencies (-20db/dec) and the closed loop gain (i.e. find the frequency at which the open loop curve equals the closed loop gain). This simple explanation of the frequency response is only valid for small amplitude signals.

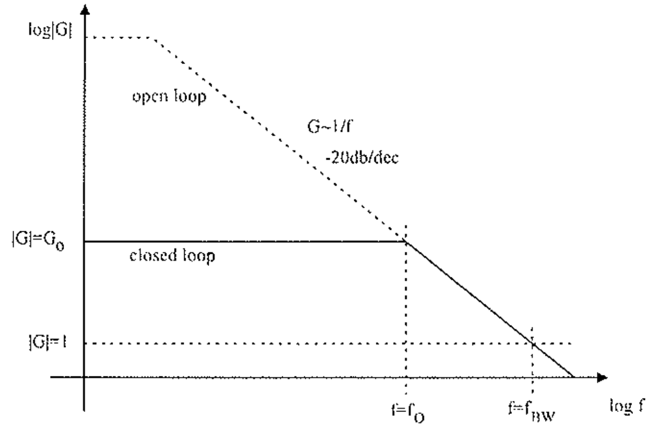


Figure 4.6: Closed loop Bode plot (magnitude only).

Exp. 4.4 Repeat Exp. 4.3 but for the operational or inverting configuration and a voltage gain of G= -30 (or as close as you can get with the available resistors). Next, measure the voltage gain (magnitude and phase) over a wide range of frequencies (up to about 1MHz) with a sine-wave input, v_{in} small enough such that the output is not distorted, but large enough to measure accurately. You should sketch v_{in} and v_{out} for one frequency and then make a table of numbers for the rest. As in the last chapter choose frequencies with logarithmic steps (10, 20, 50, 100, 200, 500,$\cdots$) with maybe a few extra points near the corner. Make a Bode plot of the magnitude of the closed loop gain of this amplifier. What is the effective bandwidth of this amplifier (the whole circuit not just the op-amp)?

OPTIONAL: measure and graph the phase portion of the Bode plot. **end**

What is the criterion for the applicability of the infinite-gain approximation? To formulate this, you must analyze the circuit with a finite amplifier gain, G_{OL}. You need to relate the voltage at X, v_{inv}, to the voltages at the two ends of the resistor chain made of R_1 and R_2. Though this is a straightforward resistance-network problem, it is easier and more instructive to treat it via the principle of superposition:

> *In any linear network, the signal at a given point, due to several sources, is just the sum of the signals that would be produced by letting each of the sources, in turn, act separately (the remaining sources being temporarily set to zero).*

In the present case there are two sources, v_{in} and v_{out}. The sketches shown in figure 4.7 illustrate how these sources act separately.

First let v_{in} act alone (diagram on the left), with $v_{out} = 0$, and then let v_{out} act while v_{in} =0 (diagram on the right). In each case the result at X is found from the voltage-divider formula. Now when v_{in} and v_{out} are present simultaneously, the combined result is:

$$v_{inv} = \frac{R_2}{R_1 + R_2} v_{in} + \frac{R_1}{R_1 + R_2} v_{out} \tag{4.20}$$

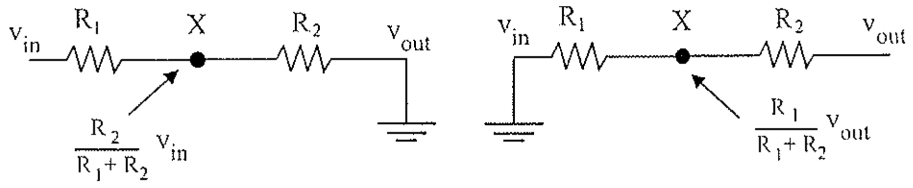


Figure 4.7: Using superposition to find the gain.

Using the relationship $v_{out} = -G_{OL}v_{inv}$ yields:

$$v_{out} = -G_{OL}v_{inv} = -\frac{G_{OL}R_2}{R_1+R_2}v_{in} - \frac{G_{OL}R_1}{R_1+R_2}v_{out} \tag{4.21}$$

Moving all terms with v_{out} to the left hand side leaves:

$$\begin{aligned} v_{out} &= -\frac{\frac{G_{OL}R_2}{R_1+R_2}}{1+\frac{G_{OL}R_1}{R_1+R_2}}v_{in} = -\frac{R_2}{R_1}\frac{HG_{OL}}{(1+HG_{OL})}v_{in} = -\frac{R_2}{R_1}\left[\frac{1}{\frac{1}{HG_{OL}}+1}\right]v_{in} \\ H &= \frac{R_1}{R_1+R_2} \end{aligned} \tag{4.22}$$

More picturesquely, note that the R_1-R_2 chain has two effects:

(i) It produces a feedback factor $H = R_1/(R_1+R_2)$. This tells us at once that the loop gain is $G_{OL}H = G_{OL}R_1/(R_1+R_2)$, and the infinite-gain approximation is good to the extent that this loop gain is much greater than unity.

(ii) It attenuates v_{in} by the factor $R_2/(R_1+R_2)$, so that the effective gain of the system is smaller, by this factor, than $G_{CL} = -1/H$ (the negative sign arises from the fact that v_{in} is connected to the inverting input of the op-amp.)

$$\frac{v_{out}}{v_{in}} = \frac{R_2}{R_1+R_2}\left(-\frac{1}{H}\right) = -\frac{R_2}{R_1+R_2}\frac{R_1+R_2}{R_1} = -\frac{R_2}{R_1} \tag{4.23}$$

This result agrees with the algebra above, and with the intuitive picture derived from the virtual-ground concept.

4.6 The Virtual-Ground Point: Current Control

Let's look at the properties of the virtual-ground point more closely, by removing R_1 and treating X as an input terminal directly as shown in figure 4.8. If a small voltage v_{in} is applied at X, then the other end of R_2 is driven to $-G_{OL}v_{in}$ by the amplifier, and thus the current through R_2 is:

$$i_{in} = \frac{v_{in} - (-Gv_{in})}{R_2} = \frac{v_{in}(1+G)}{R_2} \tag{4.24}$$

(Here G is the open loop gain of the op-amp.) This is the same current as would be drawn by a resistor $R_{in} = R_2/(1+G)$ connected directly between X and ground. R_{in} is the effective input impedance at X. A numerical example shows that R_{in} can be very low. If $G = 10^5$ and $R_2 = 10K\Omega$, then $R_{in} = 0.1\Omega$.

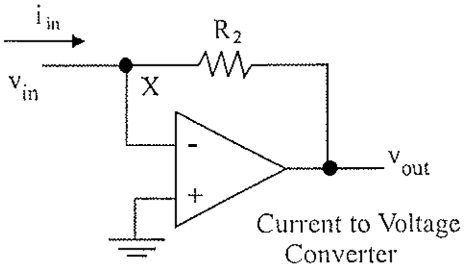


Figure 4.8: Current to voltage converter.

Thus it is fair to regard X as a virtual short circuit to ground in most contexts. The input signal is then more appropriately a *current*.

If instead, i_{in} is regarded as the input, then, the amplifier produces an output voltage given by:

$$v_{out} = -i_{in}R_2 \tag{4.25}$$

(in the infinite-gain approximation). R_2 plays the same role as the gain, but it has the dimensions of resistance, instead of being a pure number, because output and input are in different domains. V_{out}/i_{in} is called the *transfer impedance* of the system, sometimes abbreviated *transimpedance*.

You may ask what is so brilliant about the arrangement, seeing that you could produce a voltage $i_{in}R_2$ by merely passing i_{in} through a resistor R_2, without any amplifier at all. Once again it's a question of *power gain*. i_{in} is accepted into a virtual short-circuit, developing only a very small voltage v_{in}, whereas if it had to do all the work itself it would be producing a voltage drop v_{out}. The input current may be only a few μA but the output current may be a few mA (large current gain).

4.7 Non-Ideal Op-Amp Properties

In chapter 2 you have already seen that the transfer characteristic of a practical op-amp is not necessarily centered on the point $v_{non} = v_{inv}$, but may be offset by a voltage V_{io} which typically can be as large as several millivolts. V_{io} can be adjusted to zero by an *offset nulling* circuit, but the adjustment will thereafter *drift*, both as a function of temperature and age. The best op-amps for small and stable V_{io} are *chopper-stabilized* types, and they may achieve drifts as low as 0.1 μV/^{o}C. Ordinary integrated circuit op amps typically drift by approximately 1-10 μV/^{o}C.

Whether you adjust V_{io} to zero as best you can, or accept whatever V_{io} the amplifier happens to have, your circuit must ultimately work in the presence of some residual (and, in detail, unknown) offset. Because V_{io} changes only slowly, it is effectively a DC voltage and is therefore most troublesome when the signal to be amplified is also DC. In a DC amplifier, in fact, the offset voltage limits the smallest signal that can be distinguished[3]. If the signal is AC, the offset does not interfere with it directly, but it does affect the operating point of the amplifier (its DC or average condition).

[3] You may of course be able to turn on and off a very small DC signal to distinguish it from offset. That is what you do on the scope by setting the input selector momentarily to GROUND. However, such a switched signal is no longer really DC.

Practical op amps also require a finite current at their input terminals called the *input bias current.* The magnitude of this current is usually small and depends on the type of input element the op amp uses. The 741, which uses bipolar transistors, has a bias current of up to 0.5μA. The 3140, which uses an insulated-gate field-effect transistors (MOSFET), has a bias current of no more than 50pA (at 25°C). The bias currents at the two input terminals, while not necessarily equal, do tend to have similar magnitudes. The residual difference between them is called the *input offset current.* In suitably symmetrical circuits only this smaller offset current causes an error, not the individual bias currents.

The non-ideal properties of the op-amp can be summarized in the circuit model schematic in figure 4.9. The input offset voltage is labeled V_{io} and the input bias currents at the noninverting and inverting inputs are labeled I_{B+} and I_{B-} respectively. The input offset current is the difference between these two values. The effective input impedance of the op-amp is R_{in} and the output impedance is R_{out}. The output of the op-amp can be modeled as a voltage source that is controlled by v_{diff} and has a single pole at ω_0 to model the frequency dependence of the open loop gain. The values of these parameters are given in a table at the end of chapter 2.

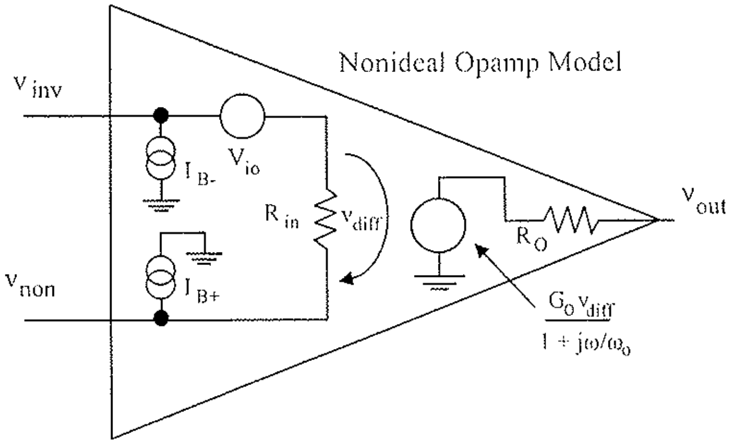


Figure 4.9: Nonideal op amp model.

The input bias currents can cause a variety of other problems when the op-amp interacts with the external components (forming the feedback loop for example). First of all, it is vital to realize that the presence of a bias current makes it mandatory that there be a *DC path to each input terminal* of the amplifier. It would thus be foolish to have an input terminal connected to nothing but capacitors. They would charge relentlessly on the bias current and ultimately drive the op amp out of range. (Capacitors that are discharged periodically by a switch are acceptable in some cases, however.)

The effective resistance of the DC path to the input terminal determines what effect the bias current has. If this resistance (the parallel combination of all available paths) is R_{dc}. I_{bias} produces a voltage drop which is equivalent to a bias voltage $V_{bias} = I_{bias}R_{dc}$. Consider the circuit shown on the left in figure 4.10. The input bias current I_{B-} produces a change in the voltage at the inverting of $\Delta v_{inv} = (R_1//R_2)I_{B-}$. This small voltage will then be amplified by the amplifier and produce a larger change in the output voltage. The input bias current at each of the op-amp inputs is typically about the same value, and more importantly drift with temperature by about the same amount. If both input terminals have the same R_{dc}, the voltage error at each input have a tendency to be equal and thus to cancel at the difference input, with only the amount $I_{offset}R_{dc}$ remaining effective ($I_{offset} = |I_{B+} - I_{B-}|$). Where bias current is a problem, therefore, the DC resistances of the two input circuits should be made equal, if necessary by inserting a dummy resistor in series with one input (see example shown in the right hand side of figure 4.10). Then the change in the differential voltage is approximately zero because Δv_{inv} cancels Δv_{non}. Note that the sign of the bias currents is rarely specified. It can flow into or out of each terminal although it is in the same direction for both terminals.

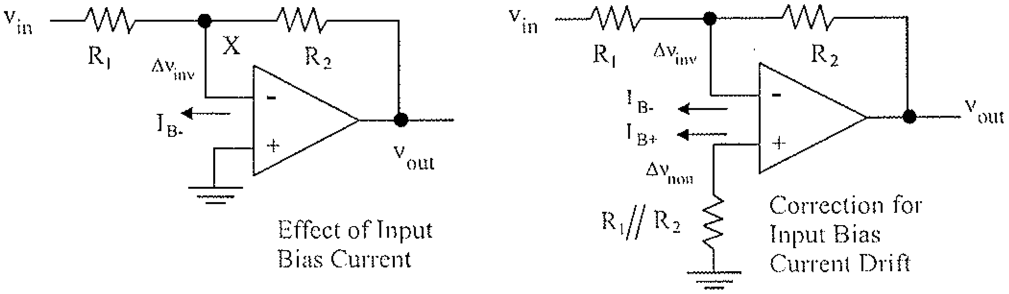


Figure 4.10: Correcting for input bias drift.

Summary: to find the worst likely effect due to offset voltages and currents, if the amplifier input is voltage-controlled then combine V_{io} with $I_{offset}R_{dc}$ to obtain the maximum unbalance voltage; if the amplifier input is current-controlled, then combine V_{io}/R_{dc} with I_{offset} to find the maximum equivalent unbalance current. Since both V_{io} and I_{offset} are as likely to be positive as negative, it is correct to consider their magnitudes only.

4.8 Summing with Op-Amps

The circuit shown in figure 4.11 produce an output that is proportional to the sum of its input signals. Each input signal (v_a, v_b, v_c) is converted into a current by means of its own input resistor (R_a, R_b, R_c). These currents are added at the virtual-ground point, X, of a transimpedance amplifier. The resulting output is:

$$v_{out} = -R_2\left(\frac{v_a}{R_a} + \frac{v_b}{R_b} + \frac{v_c}{R_c}\right) \tag{4.26}$$

Interaction between the signals (e.g., part of v_a becoming visible at the terminal for v_b) is minimized because of the low impedance at X. This point serves merely to collect all the currents from the inputs, and to send their sum on through R_2. For this reason X is often called the *summing junction* (even when there is only a single input). The summing circuit may be built with only two input signal or an arbitrary number of input signal (more than three is also possible).

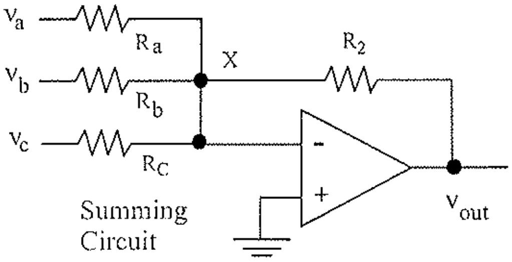


Figure 4.11: Summing amplifier.

4.9 Differential Amplifier

Although the op-amp is a differential amplifier, it's gain is not well controlled and can vary significantly from one IC to another. A differential amplifier using an op-amp in a negative feedback configuration is shown in figure 4.12. This configuration has a well controlled gain and the presence of negative feedback produces a nice stable amplifier.

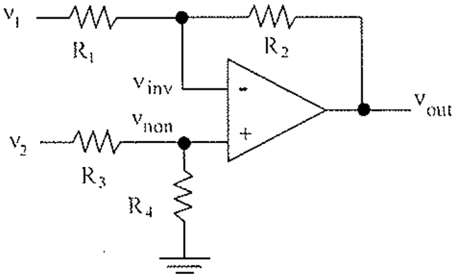


Figure 4.12: Well controlled differential amplifier using an op-amp.

The over all gain of this circuit can be calculated using the fact that negative feedback will make the two op-amp inputs equal:

$$v_{inv} = \frac{v_1 R_2 + v_{out} R_1}{R_1 + R_2} = v_{non} = \frac{R_4}{R_3 + R_4} v_2 \qquad (4.27)$$

Solving for v_{out} yields:

$$v_{out} R_1 = \frac{R_1 + R_2}{R_3 + R_4} R_4 v_2 - R_2 v_1 \qquad (4.28)$$

$$v_{out} = \frac{R_1 + R_2}{R_3 + R_4} \frac{R_4}{R_1} v_2 - \frac{R_2}{R_1} v_1 \qquad (4.29)$$

This circuit is commonly used with the special relationship between the resistors $R_1 = R_3$ and $R_2 = R_4$ in which case the gain takes the simple form:

$$v_{out} = \frac{R_2}{R_1}(v_2 - v_1) \qquad (4.30)$$

This is a differential amplifier with the gain determined by the external resistors and all of the benefits of negative feedback such as stability (as compared to the open loop configurations). One drawback of this circuit is that the input impedance is not very large (R_1 at v_1 and R_3+R_4 at v_2). A more elaborate version of this basic idea is illustrated in the instrumentation amplifier (figure 4.27). Op-amp A3 is the same as in figure 4.12 but two more op-amps have been added to produce a high input impedance at both inputs.

4.10 Time Integration Using an Op-Amp

This is among the most important of operations that can be performed on an analog signal. One possible implementation is shown in figure 4.13.

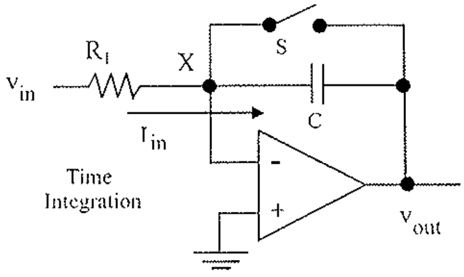


Figure 4.13: Integrating amplifier.

Let's describe the circuit from a different viewpoint (the time domain). R_2 is replaced by a capacitor, C, in the operational configuration. There again is an input current $I_{in} = v_{in}/R_1$ to the summing point X. This current flows into C and charges it. Using the definition of capacitance, the voltage across the capacitor is:

$$v_c = \frac{q}{C} = \frac{1}{C}\left[\int_0^t idt + q(0)\right] \tag{4.31}$$

which gives:

$$v_{out} = -\frac{1}{R_1C}\int_0^t v_{in}dt - \frac{q(0)}{C} \tag{4.32}$$

where $q(0)$ is the charge on C at the start of the integration period ($t = 0$). For example, you can make $q(0) = 0$ by momentarily closing switch S at $t = 0$, thus discharging the capacitor.

This equation for v_{out} looks as if v_{out} were the true time integral of v_{in}, but several approximations went into this derivation. For one thing, the amplifier open loop gain, G is assumed to be infinite. With finite G, the summing point does not remain at exactly zero voltage, and therefore I_{in} is not exactly v_{in}/R_1. In addition, there is an error caused by the input offset voltage V_{io} and bias current of the amplifier. Working with the net equivalent DC error current, $I_{dc} = (V_{io}/R_1) + I_{bias}$, causes C to charge at a constant rate even when $v_{in} = 0$. The magnitude of this rate is I_{dc}/C, and this determines the error in v_{out} that accumulates in a given integration time.

If v_{in} has a non-zero time average, its integral increases (or decreases) without bound. Evidently the amplifier must sooner or later run out of its active range! You should select component values such that v_{out} remains within the active range during the required integration time; and discharge C periodically by such means as the switch S.

You might think that, when v_{in} has an *exactly zero* average (for example, the AC output from a transformer), its integral would remain within bounds and there would then be no problem with the amplifier going out of its active range. In fact troubles still remain. The amplifier's offset voltage and bias current still contribute some effective I_{dc}, as mentioned above.

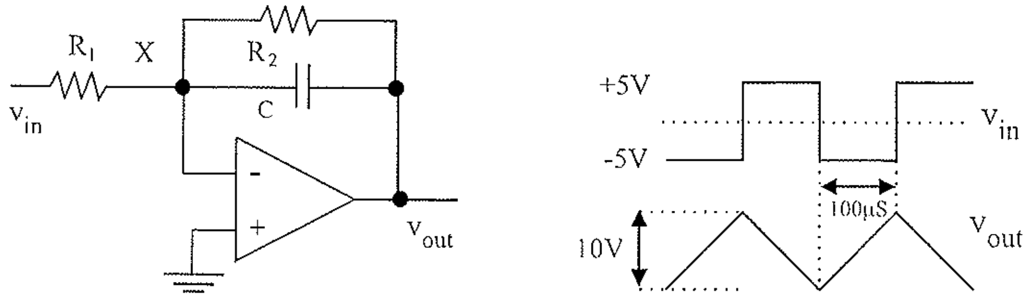


Figure 4.14: Integrating amplifier for experiment 4.5 and its desired waveforms for experiment.

The problem caused by I_{dc}, or by a slightly nonzero average of v_{in}, is not severe provided the desired input current, I_{in}, can be made much larger than the DC error current. In that case you can provide a DC path around C for the error current without upsetting the integrating action very much. Connect a resistor R_2 across C as shown in the circuit in figure 4.14. Choose its value such that the resulting output offset, $I_{dc}R_2$, remains small enough to allow the amplifier sufficient output swing. Integrating errors caused by the presence of R_2 and by the finite amplifier gain are quantified most easily in frequency-domain analysis, which will be considered after the following experiment.

Exp. 4.5 Design and build an operational integrator to produce a triangular wave, 10 V_{pp}, at f=5kHz from a square-wave input, v_{in}, as shown in figure 4.14. Use a 3140 op amp and +/-15volt power supplies. Observe v_{out} both with and without the extra resistor R_2.

The following reasoning may help you choose appropriate component values:

1. The bias current of the 3140, at most 50 pA, can be considered negligible if I_{in} is made much larger by selecting a small enough R_1.

2. The input offset voltage of the op-amp is probably 5 mV at most. There is little point in nulling this, however, because you must also consider the possible lack of symmetry in v_{in}. The function generator (even with its own OFFSET control set to OFF) is not exactly symmetrical. The average value of v_{in} is probably within 50 mV of zero. This offset, driving a DC current component through R_1, is the major source of error. Evidently changing R_1 does not affect the fractional amount of this error, since R_1 also changes the desired signal current in proportion.

3. Determine the appropriate integrating time constant, R_1C, to yield a 10 V_{pp} output from a ±5 V input at 5 kHz.

4. Pick a convenient value for R_1. Make it low enough that the resulting signal current is much larger than the bias current (item 1, above). Keep it large enough that it doesn't represent too much of a load on the generator. You will find the available range for R_1 is very wide (10K makes a convenient choice).

5. Select the corresponding value for C. (If this should turn out to be inconveniently large or small, you would change your choice of R_1.)

6. Pick a value for R_2 such that the maximum likely DC error at the input (50 mV) does not displace v_{out} by more than a volt or two, leaving ample room for the required 10 V_{pp} signal swing. Hint: think about the gain at DC with and without R_2.

When you have your circuit working as specified with the ±5V input at 5 kHz, vary the input amplitude and frequency to see over what range a good triangle output can be generated.
LEAVE THIS CIRCUIT SET UP FOR THE NEXT EXPERIMENT end

4.11 Active Filters

The simple RC filters discussed in chapter 3 are called *passive filters* because the circuit does not supply any energy (i.e. the output voltage is always less than or equal to the input voltage). You can also build similar filters using op-amp, but now the output power can be greater than the input power (the op-amp can supply power from the power supplies). Filters using op-amps are also called *active filters* (*active* means that it can add power and *passive* means it cannot).

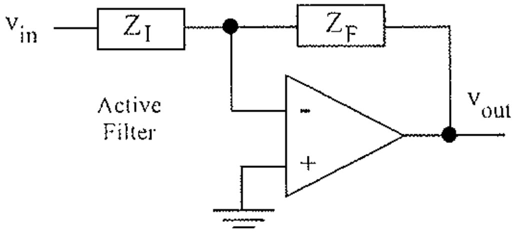


Figure 4.15: Active filter.

A general form of an inverting active filter is shown in figure 4.15. Active filters can also be made in the non-inverting configuration as well although this is not shown here. The feedback elements have been replaced with generalized impedance element Z_I and Z_F (which can be any combination of resistors, capacitors, etc.). Using the generalized impedance elements the overall gain of this circuit (figure 4.15) in the *frequency domain* is just:

$$G(\omega) = \frac{v_{out}(\omega)}{v_{in}(\omega)} = -\frac{Z_F}{Z_I} \tag{4.33}$$

(assuming the op-amp open loop gain is infinite). A similar expression exists for the non-inverting configuration.

Viewed in the *frequency domain*, the original integrator's performance (without R_2 or equivalently $R_2 = \infty$) can be obtained by merely substituting $Z_F = 1/(j\omega C)$ and $Z_I = R_1$ giving:

$$G(\omega) = \frac{v_{out}(\omega)}{v_{in}(\omega)} = -\frac{1}{j\omega R_1 C} \tag{4.34}$$

This transfer function, involving division by the factor $j\omega$, corresponds to time integration, as was shown in chapter 3. The Bode plot for $G(\omega)$ is a straight line of slope -20 dB/decade as shown in figure 4.16. It crosses the level $G = 1$ when $\omega = 1/(R_1C)$. The line would rise without limit at low frequencies, in the infinite-gain approximation and with $R_2 = \infty$. However, the closed loop gain G cannot exceed the open loop gain G_{OL} of the op-amp. Hence the Bode plot must fold over into a horizontal line at the level $G = G_{OL}$ which it does at a corner frequency given by:

$$\omega_c = \frac{1}{G_{OL}R_1C} \tag{4.35}$$

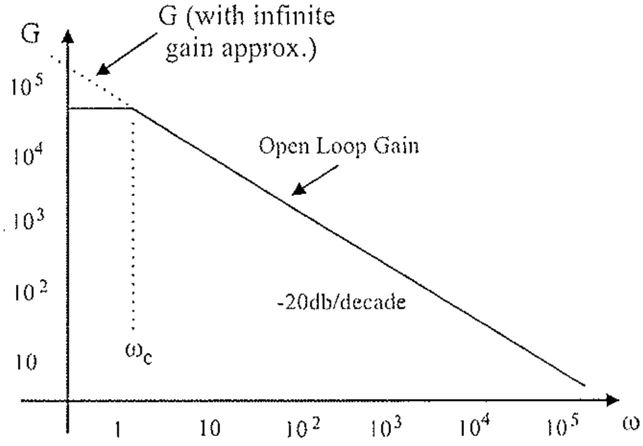


Figure 4.16: Simple integrator frequency response.

A more precise analysis starts from the expression towards the end of section 4.5 on the effects of finite loop gain in the inverting configuration. Substituting $Z_C = 1/(j\omega C)$ for R_2 in the final result yields:

$$G(\omega) = \frac{v_{out}(\omega)}{v_{in}(\omega)} = -\frac{G_{OL}Z_C}{R_1 + Z_C + G_{OL}R_1} = -\frac{G_{OL}}{1 + j\omega(1 + G_{OL})R_1C} \tag{4.36}$$

This has the same form as a transfer function for a low pass filter, $1/(1 + j\omega\tau)$ with an effective time constant of $(1 + G_{OL})R_1C$. The previous argument gave the same result except that $(1 + G_{OL})$ was replaced by G_{OL} (which is a negligible difference if G_{OL} is very large).

The integrator's performance is thus not perfect. As for the simple R-C low-pass filter in chapter 3, the low-frequency response ultimately fails to keep rising, as it should. However, the corner frequency (which signals the failure) is much lower (by a factor G, in fact) than the corner frequency of a passive low-pass filter using the same R and C.

You might ask why you couldn't merely use a low-pass filter with time constant G_{OL} times longer than that of the integrator, and amplify the small output voltage by a factor G_{OL} (as shown in figure 4.17). As far as signal amplitude is concerned, the two systems are similar. However, the amplifier on the left works without benefit of negative feedback, and all the imperfections inherent in the quantity G_{OL} appear in full. Also, the required size of the capacitor is inpractical (it is much too large). On the right is a feedback loop whose performance, in the interesting frequency region, is almost independent of G_{OL} and the required capacitor is a manageable size.

It is equally easy, with the help of a Bode plot, to see how inclusion of a resistor R_2 in parallel with C degrades the performance of the integrator. Even when G is arbitrarily large, the magnitude

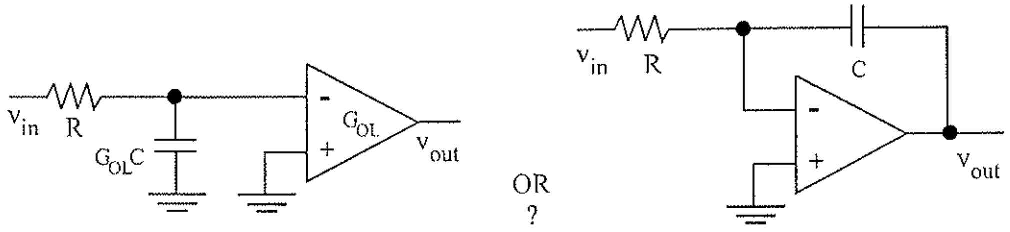


Figure 4.17: Passive filter plus op-amp versus active filter.

of V_{out}/V_{in} cannot exceed the value R_2/R_1. That's the effective gain of an inverting amplifier using R_1 and R_2 alone. In other words, R_2 lowers the low-frequency gain of the integrator, causing the Bode plot to become flat at a somewhat higher corner frequency. You can think of the effect of R_2 as lowering G to the value R_2/R_1. Evidently you should try to keep R_2/R_1 as large as possible, consistent with an acceptably small DC offset in v_{out}. For example, in experiment 4.5 you assumed a maximum source offset of 50 mV, and wanted to keep the output offset to 2 V. Thus you should have used R_2/R_1 = (2 V)/(50 mV)= 40. This obviously degrades the integrator's low-frequency performance by an enormous factor from what would have been available with $G = 10^5$.

The total transfer function is derived using the expression for the gain of the inverting active filter and the following expressions for the input impedance and the feedback impedance:

$$Z_I = R_1 \tag{4.37}$$

$$Z_F = R_2//Z_C = \frac{R_2\frac{1}{j\omega C}}{R_2 + \frac{1}{j\omega C}} = \frac{R_2}{1 + j\omega R_2 C} \tag{4.38}$$

Then the transfer function is:

$$G(\omega) = \frac{v_{out}(\omega)}{v_{in}(\omega)} = -\frac{Z_F}{Z_I} = -\frac{R_2}{R_1}\frac{1}{1 + j\omega R_2 C} \tag{4.39}$$

This result has a very similar form, with a single pole at $1/(R_2C)$ and a leading factor of (R_2/R_1) taking the place of G_{OL}. The Bode Plot of this active integrator is shown in figure 4.18. Note that the leading negative sign does not appear in the magnitude of G but instead comes through as an extra 180^o phase shift.

> $\boxed{\text{Exp. 4.6}}$ Make à Bode plot (i.e. measure it) for your integrator (from the previous experiment with both R_1 and R_2 in place), using sine waves instead of square waves. Measure the magnitude and phase of the gain as a function of frequency. Include frequencies of about a factor 10X-100X above and below the corner frequency so that the G reaches its asymptotic slope. $\boxed{\text{end}}$

4.12 Integration and Bias Currents

"Integration" is reduced to even simpler terms in the case of input signals which come directly from a *current source*. A common practical example is that of the photomultiplier tube, whose anode collects

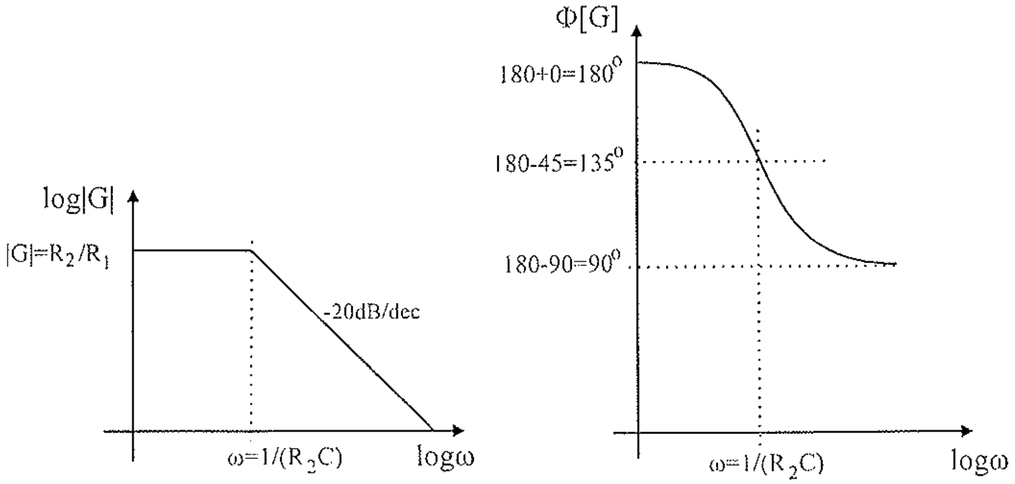


Figure 4.18: Frequency response of the active integrator (figure 4.14) with R_1 and R_2.

whatever electrons are supplied to it by the electron-multiplier structure (provided only that the anode voltage is not taken to "unreasonable" limits). The electron current is proportional to the light intensity striking the photomultiplier. If you wish to determine the *total* light flux in a given exposure, you need to integrate the output current over the time of interest. Alternatively, you need to measure the *charge* delivered to the anode. The circuit shown in figure 4.19 can therefore be considered as an integrator for i_{in} or as a device for measuring total input charge.

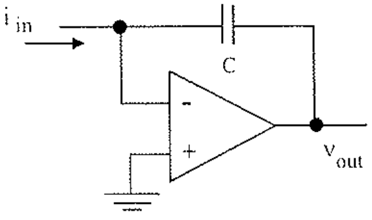


Figure 4.19: Current integrator.

The performance of this type of charge-sensitive circuit is limited by only two elements: the capacitor C and the op amp. If you use a low-leakage capacitor (ceramic, or plastic film type), the limiting parameter usually becomes the input bias current of the op amp. It is in this situation that the FET-input op amp has an enormous advantage. Let's do a short experiment to demonstrate this.

> **Exp. 4.7** Hook up the simple integrator shown in figure 4.19 using a 741 op-amp run from +/-15 volt supplies, and a ceramic or plastic-film capacitor of about 0.33μf for C. Do not connect any external input signals, the op-amp input bias current will produce an effective input current. Monitor v_{out} with a DC-coupled scope. Start the circuit from reasonable initial conditions by momentarily shorting C. Then observe the rate at which v_{out} drifts spontaneously. Calculate the input bias current of the op amp.

Now plug in a type 3140 op amp in place of the 741 and repeat the measurement, with C = 100 pfd. Some 3140's may have an input bias current too small to measure, however you can still establish an upper bound for the magnitude of the input bias current.

Compare both measurements to the manufacturer's spec. sheets. end

Beyond the difference in input bias currents, the 3140 and 741 differ in several other respects, of course. Chief among them are the higher slew rate of the 3140, and the fact that the 3140 can take its output right down to the negative supply-rail potential. On the other hand, the 741 is much more rugged and less likely to spring surprises on you, such as oscillations or locking-up in contorted conditions.

4.13 Time Differentiation using Op-amps

In the circuit shown in figure 4.20, i_{in} is the charging current for the input capacitor C, which, in the infinite-gain approximation means:

$$i_{in} = C\frac{dv_{in}}{dt} \tag{4.40}$$

$$v_{out} = -Ri_{in} = -RC\frac{dv_{in}}{dt} \tag{4.41}$$

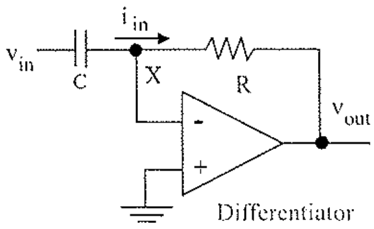


Figure 4.20: Op amp differentiator.

The circuit thus acts as differentiator. This can also be seen in the frequency domain by writing:

$$Z_C = \frac{1}{j\omega C} \tag{4.42}$$

$$\frac{v_{out}}{v_{in}} = -\frac{R}{Z_C} = -j\omega RC \tag{4.43}$$

The factor $j\omega$ implies differentiation (see chapter 3). The corresponding Bode plot (see figure 4.21) rises at 20 dB/decade. This rise is interrupted because of the finite open loop gain G, which, in fact, is falling at 20 dB/decade above the amplifier's cutoff frequency. The resulting Bode plot is even more complicated than would be implied by the intersection of these two lines, because of the effects of marginal instability, which will be touched on briefly in a moment.

Exp. 4.8 Build the differentiator shown in figure 4.22 (using a 3140 op-amp and +/-15 Volt supplies) and apply triangle waves from the function generator as input. Observe v_{out}, initially with R_1= 0. Check the edges of the waveform for oscillatory transients or overshoot.

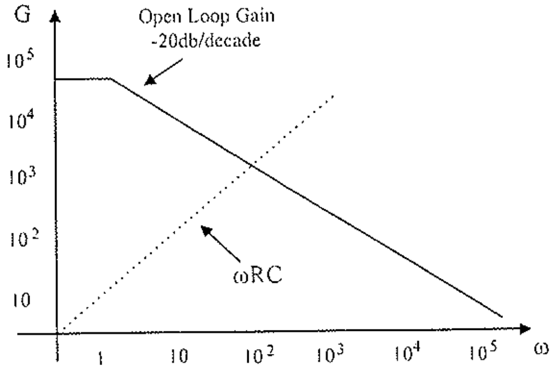


Figure 4.21: Op amp differentiator Bode plot (magnitude only).

Add a resistor R_1 in series with the input, trying several values in the range 100Ω to 1KΩ. Note the effect on the waveform for v_{out}. Select the smallest R that just eliminates overshoot in v_{out}.

OPTIONAL: Calculate the loop gain HG_{OL} with a non-zero R_1 and show that its phase does not easily get close to ±180° (a slightly modified version of eq. 4.22 may be used here). The phase of the fedback network (R_2, C, and R_1) adds to the phase of the op-amp (-90° at high freq.). The loop gain is the effective feedback factor for the whole circuit and should not change sign if the circuit is to remain stable. end

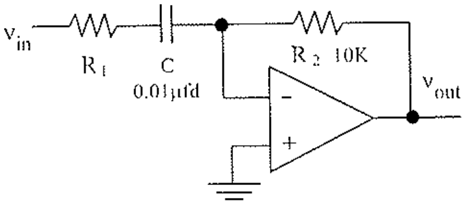


Figure 4.22: Stabilized op amp differentiator.

The transient misbehavior of the differentiator (in Exp. 4.8), with R_1=0, arises from the fact that the phase shifts around the negative-feedback loop add up to almost 180° at certain frequencies. The expression for the closed loop gain (eq. 4.22) contains $1 + HG_{OL}$ in the denominator. Both H and G_{OL} may be complex so this expression may go to zero giving an uncontrolled output even with no input (instability). A 180° phase shift converts negative feedback into positive feedback, and could thus give rise to self-sustaining oscillations. The two main contributions to the phase shift come from the amplifier and from the feedback network. The amplifier, above its corner frequency of about 10Hz, has a phase lag which approaches 90°. The feedback network, assuming v_{in} comes from a low-impedance source, represents a low-pass R-C filter and thus also introduces a phase lag which approaches 90°. At high frequency (with R_1=0):

$$HG_{OL} = \left(\frac{1}{1 + j\omega RC}\right)\left(\frac{G_o}{1 + j\omega/\omega_O}\right) \sim \frac{G_o}{(j\omega RC)(j\omega/\omega_O)} = \frac{-G_o}{\omega^2 RC/\omega_O} < 0 \qquad (4.44)$$

These two phase lags do not quite add up to 180^o, and so the circuit does not actually oscillate. However, it comes close enough to instability to produce the "ringing" transients you observed.

Introducing R_1 in series with v_{in} changes the high-frequency phase shift of the feedback network back to 0 deg. When the reactance of C is negligible, R_2 and R_1 act as a resistive voltage divider. A suitable choice of R_1 can thus avoid too close an approach to 180^o total phase lag, with a corresponding improvement in the transient response of the circuit. (Note that, even with $R_1 = 0$, the source impedance for v_{in} may provide some effective input resistance, in some cases enough to control the circuit's transient response.)

4.14 Nonlinear Operations

Either or both of the elements Z_1, Z_2 in the operational configuration can be made a nonlinear device, for example, a diode. Such arrangements give us wide latitude in the type of input-output characteristic that can be obtained. Among the most useful are the following: output proportional to $\log v_{in}$; output $= |v_{in}|$; output (dc) = peak value of v_{in}. Some examples will be given as the occasion arises. The analysis of such circuits, even with nonlinear elements, still follows the general pattern outlined above: first, find i_{in} from v_{in}, using the v-i law for the element in position Z_1; second, find v_{out} from i_{in}, using the v-i law for the element in position Z_2.

4.15 Stability

In the present context, *instability* of a circuit implies that the circuit produces an output without an external input. It is effectively supplying its own input. Two criteria must be satisfied if the circuit is to maintain oscillation *at a constant amplitude*: (i) the phase shift around the loop must be exactly $\Phi[HG_{OL}] = 180^o$ (or 0 deg. including the minus sign from the inverting inpt), and (ii) the loop gain must be exactly unity $|HG_{OL}| = 1$. Only if both criteria are met is the input provided by the feedback loop exactly that which is needed to produce the steady-state output.

These criteria presuppose that the oscillations are sinusoidal and the circuit is linear. To justify this assumption you can imagine that the circuit is starting to oscillate from very small amplitude, which would make at least its initial behavior linear. The frequency of oscillation is then determined by the phase-shift criterion, and the loop gain at this special frequency determines whether the oscillations grow or decay. If the loop gain is greater than one, the oscillations grow[4]. If the loop gain is less than one, the oscillations decay, but they may do so only gradually if the loop gain is close to unity. This situation corresponds to the oscillatory, "ringing," transients observed.

Negative feedback becomes positive if a 180^o phase change is introduced into the loop beyond the phase reversal originally used to make the feedback negative. This phase change can be a phase *advance*, produced typically at low frequencies by high-pass filters. More importantly (because its unavoidable)

[4] Growing oscillations ultimately drive the circuit into its saturated limits, where the effective loop gain is reduced and an equilibrium situation can be reached. The oscillations are then quite nonlinear, on account of the saturation, and "phase shift" and "loop gain" lose their precise meaning. The final equilibrium state of an unstable loop is not of concern here.

the change can be a phase *lag*, produced at high frequencies by low pass filters. (All practical circuit elements act effectively as low-pass filters, since they cannot transmit signals out to infinite frequency.)

The behavior of the feedback loop is displayed graphically in a pair of Bode plots (see figure 4.23), one for the loop gain, GH (in dB), and the other for the phase change, $\Delta\phi$. Look for the frequency at which $|\Delta\phi| = 180^o$ and find the loop gain there. It should be below unity, i.e., below 0 dB. The amount by which it is less than 0 dB is called the *gain margin*. Alternatively, look for the frequency at which GH has fallen to 0dB and find $\Delta\phi$ here. Its magnitude should not have reached 180^o, and the amount by which it falls short is called the *phase margin*.

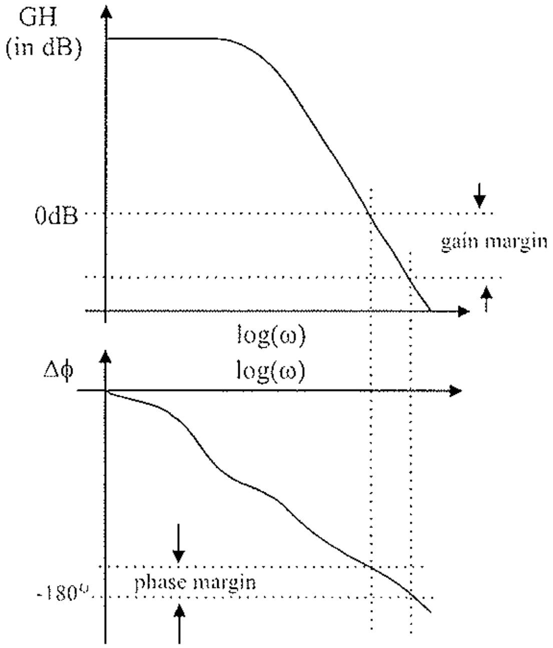


Figure 4.23: Stability and phase.

As already mentioned earlier a certain minimum phase lag is associated with gain rolloff; the phase lag is 90^o for every 20 dB/decade slope of the gain curve (assuming this slope is maintained over a wide frequency interval). To keep the phase lag from reaching 180^o, the rolloff must be slower than 40 dB/decade. Most internally compensated op amps use a rolloff of 20 dB/decade, producing a 90^o phase lag and thus leaving room for some additional phase lag in the feedback network which completes the loop. However, to maintain acceptable transient response, the feedback network should not introduce more than 30^o to 60^o of phase lag until the loop gain has fallen below 0 dB. This corresponds to a 60^o to 30^o phase margin (a useful engineering criterion.) There should also be a gain margin (gain less than 0 dB) of 6 to 15 dB for frequencies.

The central problem in maintaining stability is this: a large loop gain in a feedback system is nice, but you cannot afford to jettison this gain too fast at high frequencies. The larger the loop gain, the wider the frequency interval over which a slow, controlled rolloff must be engineered.

Internally compensated op amps are usually designed to be stable even if the feedback factor, H, is unity (as it would be in a voltage follower circuit). This connection gives the largest loop gain. Consequently, G must be rolled off slowly over a very wide range. This is why the corner frequency comes at such a surprisingly low value (about 10 Hz). Evidently the rolloff is far too conservative if the loop gain of the system is much lower. Also, if the feedback network does not introduce much phase lag of its own, the rolloff could be somewhat faster than 20 dB/decade and still leave adequate phase margin. By tailoring the rolloff for every given case we can therefore maintain a large loop gain out to higher frequencies, with improved performance of the circuit. This can be done with op amps using external compensation, but it involves significant design effort.

4.16 Stability Problems in Voltage Followers

As an example of how phase shifts within the feedback loop can cause problems with stability, consider the commonly used voltage-follower circuit when it is asked to drive a capacitive load, as in the circuit in figure 4.24. Remember the presence of the output impedance R_o of the op-amp

> Exp. 4.9 Investigate the stability of a voltage follower in the presence of a capacitive load as shown in figure 4.24. Use a type 741 op amp, with ±15 V supply rails, and drive it with a 0.5 V p-p square wave. Try several values of C_L from 50 pF up to 0.01μF or so and describe the changing behavior of v_{out}. To see how op amps may differ in performance characteristics, repeat the experiment with a 3140 substituted for the 741. end

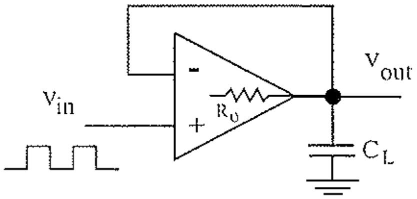


Figure 4.24: Stability test.

As the diagram shows, C_L acts as low-pass filter in conjunction with the internal output resistance, R_o, of the op amp. This filter introduces phase lag and can thus produce marginal instability or even self-sustaining oscillations.

The susceptibility of voltage followers to oscillation is a nuisance even in cases where the signals to be handled are only DC or at most low frequency. It is often necessary to connect a long, shielded cable to the output of the op amp. The capacitance of this cable can cause instability, as you have just seen. A practical remedy is to isolate the cable capacitance from the op-amp output with a resistor, R_1 (left side of figure 4.25), which is large compared to the op-amp output impedance R_o. The op-amp output voltage, v_{op}, is then almost unaffected by the presence of C_L and the feedback loop can be closed as shown.

> NOTE: Exp. 4.10 through 4.12 are optional. You should understand how these experiments work but you do not have to do them.

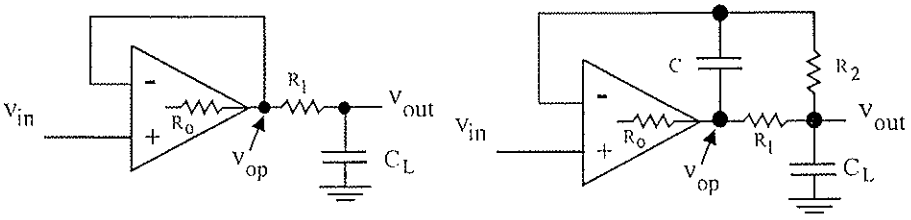


Figure 4.25: Steps to increase stability when driving a capacitive load.

Exp. 4.10 OPTIONAL: With the 3140 in place and C_L = 1000 pF, add the isolating resistor, R_1 =100Ω, as on the left in figure 4.25. Examine v_{out} and v_{op} and record the effect of R_1. **end**

A disadvantage of this arrangement is that the feedback no longer "supervises" v_{out}, but only v_{op}. Should the load have some resistance as well as capacitance, v_{out} will be lower than v_{op}, with the feedback unaware of this situation.

Exp. 4.11 OPTIONAL: Using the same circuit as in the previous experiment, and with v_{in} = $1V_{pp}$ add a 470Ω resistor in parallel with C_L to simulate a resistive load. Check v_{out}/v_{in} for degradation of voltage-follower action. **end**

The second sketch (on the right in figure 4.25) shows a way of tackling this problem. At high frequencies, where the phase lag is troublesome, feedback is taken directly from v_{op} via C ($|Z_C| << R_2$). At low frequencies the feedback comes from v_{out} via R_2 ($R_2 << |Z_C|$), and thus v_{out} is ultimately supervised and made equal to v_{in}, as desired.

Exp. 4.12 OPTIONAL: Modify your voltage follower as indicated on the right in figure 4.25. For C and R_2, whose values are not critical, use 1000 pF and 10 KΩ. Again, check v_{out} for faithfulness of replication of v_{in} (with the 470Ω load resistor). **end**

4.17 Op-amp Summary

The important elements of this chapter can be summarized in three op-amp configurations as shown in figure 4.26. There are three different configuration, the inverting, the non-inverting and the differential configuration. When each external circuit element is allowed to be a generalized impedance element (usually some combination of resistors and capacitors), most of the circuits discussed in this chapter are included.

4.18 Practice Problems

[1] A close friend of yours has an exotic (and expensive) plant that is sick and near death. After consultation with an eminent plant doctor, your friend finds out that the plant will need an exactly

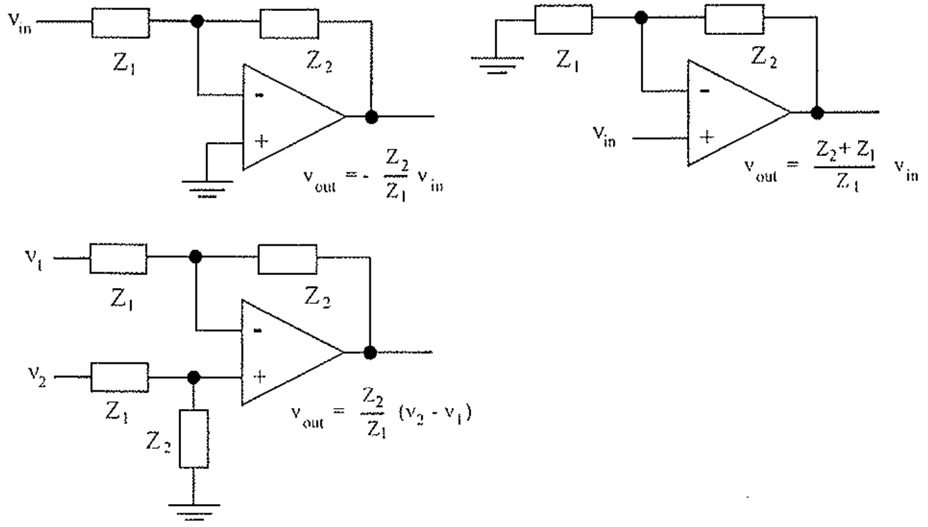

V_{in}
Z_1
Z_2
-
+
$V_{out} = -\frac{Z_2}{Z_1} V_{in}$
Z_1
Z_2
V_{in}
-
+
$V_{out} = \frac{Z_2 + Z_1}{Z_1} V_{in}$
V_1
Z_1
Z_2
-
+
V_2
Z_1
Z_2
$V_{out} = \frac{Z_2}{Z_1}(V_2 - V_1)$

Figure 4.26: Summary of op-amp configurations with general impedance elements Z.

constant amount of sun light every day. Unfortunately in Ithaca the sunlight varies greatly and the plant is near the window and can't be moved. However, there is an electric lamp nearby. Your job is to design a circuit that will control the lamp to produce a constant amount of light on the plant at all times so that the plant will get better.

The lamp can be roughly modeled as a resistor that produces a variable amount of light depending on the amount of voltage applied to it (the light output is not necessarily linearly dependent on the voltage because the filament heats up and changes resistance). Also available is a photovoltaic cell that produces a voltage output that varies linearly (between 0 and 1 volts) with the amount of light hitting it. Design a circuit using an op-amp with negative feedback that senses the amount of light arriving at the sick plant and adjusts the lamp voltage to keep a constant amount of light on the plant even in Ithaca. You may assume that someone will turn it on in the morning and off in the evening. The amount of light should be adjustable with a potentiometer (hint, set up a variable reference voltage similar to the 10k pot in experiment 2.1). The photovoltaic cell is small and can be placed near the plant. (You may also assume that the op-amp can supply enough current to drive the lamp although in practice a 741 probably would require some help.)

[2] In the circuit shown in figure 4.27 you may assume that the op-amps are ideal and driven from ±15 volt supplies. Derive an analytical expression (i.e. leave it in terms of the algebraic symbols R_1 through R_7) for a) through d). This configuration is known as an instrumentation amplifier.

a) How is v_4 related to v_1, and how is v_5 related to v_2?

b) Find an expression for I_2 in terms of $v_{diff} = v_2 - v_1$.

c) Find expressions for v_3 and v_6 as functions of v_1, v_2, and v_{diff}.

d) Find an expression for v_{out} as a function v_3 and v_6.

e) If $R_1 = R_3$ =20K, and $R_4 = R_5 = R_6 = R_7$, show that $\frac{v_{out}}{v_{diff}} = 1 + 2\frac{R_1}{R_2}$.

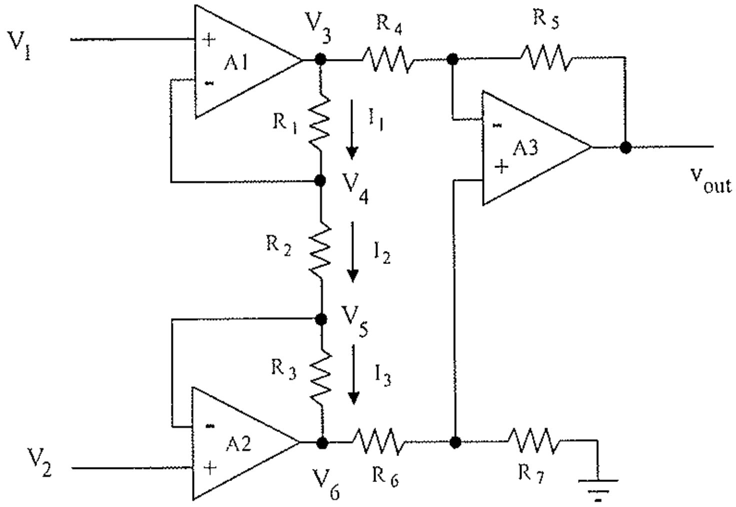


Figure 4.27: Instrumentation amplifier.

[3] Complete the design for the active bandpass filter shown in figure 4.28 assuming that the op-amp is ideal, that $R_I C_I \gg R_F C_F$ and $R_I = 2.7\text{K}$ and $R_F = 47\text{K}$.
a) Derive an algebraic expression for $G(\omega) = v_{out}(\omega)/v_{in}(\omega)$.
b) Make a Bode plot (magnitude and phase) of $G(\omega)$.
c) What is the maximum value of $|G(\omega)|$?
d) Find C_1 and C_2 to make the filter pass only audio frequencies between 20Hz - 20kHz (i.e. put the corner frequencies at these points).

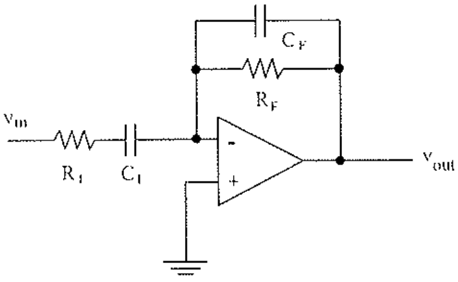


Figure 4.28: Active band pass filter.

[4] Consider a simple non-inverting amplifier (op-amp and two resistors, as in figure 4.4 of the lab manual), designed to have a gain of 5 using the approximation that the open loop gain of the op-amp is infinite. Calculate the magnitude of the actual gain using the actual. finite gain of a 741 op-amp (unity gain bandwidth of 1 MHz) at frequencies of 20 KHz 100 kHz and 200 kHz. How good is the infinite gain approximation? Where (in frequency) do you expect the corner of the closed loop gain curve to be? (Hint: you will need equations in section 4.1.)

[5] In the circuit shown in fig. 4.29, you may assume that the op-amp is ideal, and R_I= 1K, R_F= 100K, C_I= 0.01μfd.

a) Derive an algebraic expression for $G(\omega) = v_{out}(\omega)/v_{in}(\omega)$ and put it in pole-zero form.

b) Make a Bode Plot (magnitude and phase) of $G(\omega)$. Label all corners and give a value for all asymptotic slopes and limiting values of $G(\omega)$.

c) What are the corner frequencies (or frequency) in Hz? Give an expression and a number and label them on your graph.

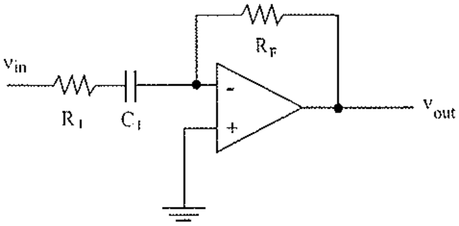


Figure 4.29: Active filter.

[6] The circuit shown in fig. 4.30 may be referred to as an *all-pass filter*. It has the novel property of passing all frequencies with the amplitude unchanged (if the op-amp is ideal), but the phase shifted by a specific amount. You may assume that the op-amp is ideal, and $R_1 = R_2$

a) Derive an algebraic expression for $G(\omega) = v_{out}(\omega)/v_{in}(\omega)$ and put it in pole-zero form.

b) Show that the magnitude of $G(\omega)$ is unity for all frequencies.

c) Find an expression for the phase shift of $G(\omega)$ as a function of ω, R_3 and C.

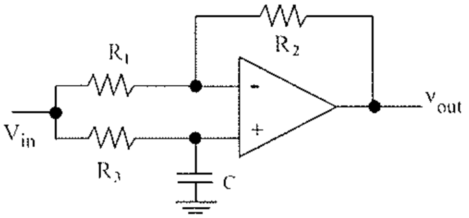


Figure 4.30: All-pass filter giving only a phase shift.

Chapter 5

DIODES

5.1 Ideal Diode

The *ideal diode* is a device which conducts perfectly (R = 0) for current flowing in the *forward direction*, but insulates perfectly ($R = \infty$) when voltage is applied in the *reverse* direction. This behavior is *nonlinear.* The schematic symbol for a diode is shown on the left in figure 5.1. It has two terminals, the anode and cathode. The arrow points in the direction of flow of positive current (from the anode to the cathode). The I-V characteristic of an ideal diode is shown on the right in figure 5.1.

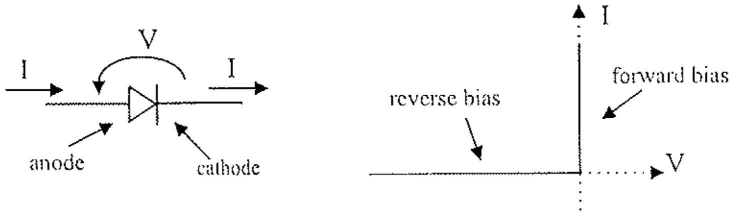


Figure 5.1: Ideal diode.

When the diode is forward biased and a large current is flowing through it, the diode is said to be "ON". When it is reverse biased and no current is flowing through it, the diode is said to be "OFF".

5.2 The Silicon Junction Diode

It is possible to make a diode from a variety of semiconductor materials as well as a vacuum tube. Most diodes in use today are made from the semiconducting material silicon. A typical diode looks similar to a resistor except that it has only one band (on the cathode end) as shown in figure 5.2. The direction of the diode is important, so you have to place the banded end (cathode) in the correct physical orientation in the circuit. Unfortunately, it is not possible to produce an ideal I-V curve. Instead the I-V curve of a semiconducting diode looks like the graph shown on the right in figure 5.2.

Pure silicon is essentially an insulator (high resistance) at room temperature, however it can be made to conduct by adding impurities in small amounts. Semiconducting materials, such as silicon.

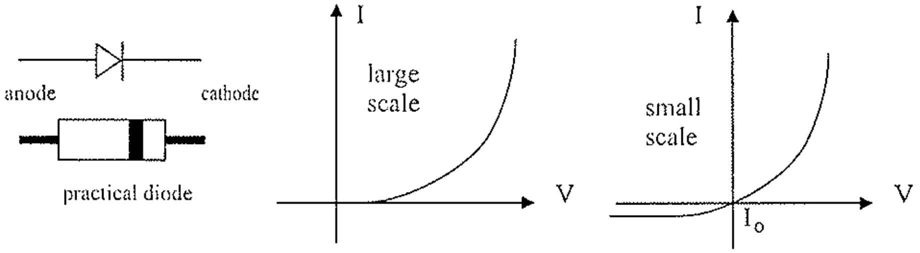


Figure 5.2: Practical diode.

can be doped as a p-type material or an n-type material by adding different types of impurities. A diode consists of a junction of p-type material and n-type material as shown in figure 5.3. In n-type material the charge carriers are electrons with negative charge (hence the name n-type). However in p-type material the charge carriers have a positive charge and are called holes. A hole is actually a nearly filled valence shell (of electrons). It is a result of solid state physics that this nearly filled valence shell can be treated as a single pseudo-particle with a positive charge. For most purposes p-type material can be thought of as having free positive charges that carry current.

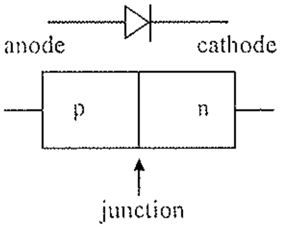


Figure 5.3: Semiconductor diode.

When the ends of the diode are connected together the holes diffuse into the n-type region and the electrons diffuse into the p-type region. The holes and electrons near the junction recombine with each other to yield a region near the junction that is devoid of charge carriers and is called a depletion region (see figure 5.4).

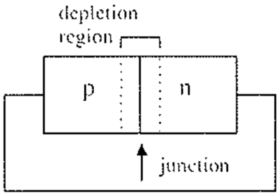


Figure 5.4: Semiconductor diode.

When a bias voltage is applied (as in figure 5.5) the external voltage produces an additional force on the holes and electrons. With reverse bias the force on both the holes and electrons (shown as horizontal arrows in the diagram) acts to pull the charge carriers (holes and electrons) further away from the junction. The depletion region grows larger and no current flows through the diode. The semiconducting material is inherently charge neutral (there is an equal number of positive and negative charges). When the holes and electrons are drawn away from their original positions (as in reverse bias) there is a net charge imbalance near the junction. This imbalance creates a large electrostatic field near the junction that exactly cancels the electrostatic field produced by the external voltage source. However, with forward bias, the force on the charge carriers pushes them towards the junction where they recombine. The external voltage source simply supplies more charge (positive and negative) at both ends of the diode and a large current flows through the diode.

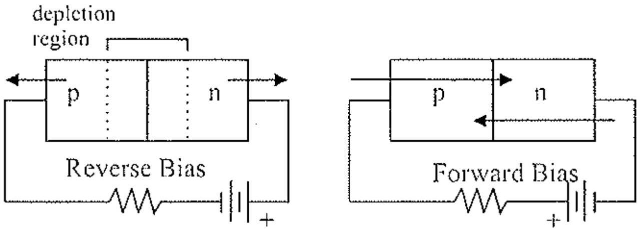


Figure 5.5: Bias voltage applied to a semiconductor diode.

5.3 The Diode Equation

The I-V characteristic of a semiconducting junction diode is described quite well by the diode equation:

$$I = I_0 \left[\exp\left(\frac{eV}{kT}\right) - 1 \right] \tag{5.1}$$

I_0 is a constant determined by how the particular diode was manufactured (i.e. it is a property of the diode) and is called the reverse bias saturation current. k is Boltzmann's constant and T is the absolute temperature. kT is the typical thermal energy at temperature T, and the combination kT/e (e = electronic charge) is the voltage, V_T, through which an electron would have to be accelerated to acquire energy kT. At room temperature, kT/e is about 26 mV.

Practical diodes deviate from the diode equation in several respects. Some of these deviations can be accounted for by using, instead of V_T, a corrected thermal voltage, V_T^*, which is typically about 40 mV. With this notation the diode equation becomes:

$$I = I_0 \left(e^{V/V_T^*} - 1 \right) \tag{5.2}$$

Because V_T^* is such a small voltage, the exponential in this equation becomes vanishingly small as soon as a negative V of only a small fraction of a volt is applied. Then the reverse current of the diode saturates at the level:

$$I \sim -I_0 \quad \text{[reverse bias, } V \leq -5V_T^*\text{]} \tag{5.3}$$

For this reason I_0 is called the *saturation current*.

Similarly, when V is positive the exponential soon becomes much larger than unity and the diode equation reduces to:

$$I \sim I_0 e^{V/V_T^*} \quad \text{[forward bias, } V \geq 5V_T^*] \tag{5.4}$$

At a typical forward voltage of about 600 mV, $V \sim 15V_T^*$, and the value of the exponential is about $3\text{x}10^6$. Thus I is many orders of magnitude larger than I_0. In a small diode I_0 might be about 1 nA.

To express V as a function of I in the forward region, take the logarithm of both sides of the last equation:

$$V = V_T^* \log\left(\frac{I}{I_0}\right) \tag{5.5}$$

Thus, in the region where this equation is valid, V is a logarithmic function of I.

The most notable deviation of a practical diode from the ideal is that it requires a small (but nonzero) forward voltage before significant forward conduction takes place. It is difficult to deal with the exponential diode equation all of the time, however the major features may be included in a simple circuit model as shown in figure 5.6. In many circumstances a real diode can be approximated as an ideal diode in series with a small voltage source, V_D. In other words, when forward current flows, the diode produces a finite *forward voltage drop*, V_D. To include this fact in a circuit model requires placing a battery of voltage V_D in series with an ideal diode. V_D, for silicon junction diodes, is about 0.6 V.

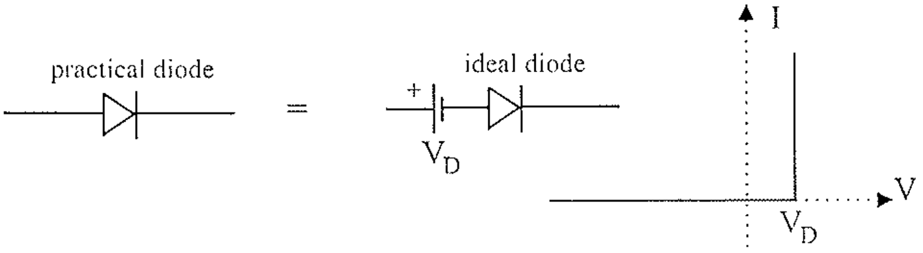


Figure 5.6: Simple model of a silicon diode.

5.4 Diode Ratings

There are two situations in which a diode (or transistor in future chapters) may be destroyed, which you should try to avoid:

1. Do NOT apply excessive *reverse voltage*. As the sketch in figure 5.7 indicates, the reverse current does not remain limited to the saturation current I_0, but increases (slowly at first, and then suddenly). The diode has entered *breakdown*. If the breakdown current and the reverse voltage together produce enough power to overheat the diode, the diode will be destroyed (see Table 5.2 for typical values). Even if the breakdown current is limited, however, there may still be local spots of high current density in the junction. Repeated episodes of minor breakdown stress may then degrade the diode progressively.

2. Do NOT permit excessive *forward current* to flow. This current, together with the associated voltage drop, heats the junction and may lead to its destruction. Note that the negative temperature coefficient of the diode voltage (V_D gets smaller as the temperature increases) can lead to *thermal*

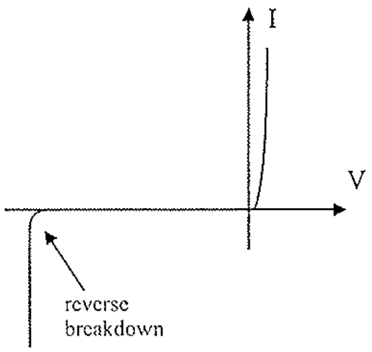


Figure 5.7: Reverse bias breakdown in a diode.

runaway if a fixed voltage is applied to the diode. As the temperature rises, the current goes up too, and the dissipation increases. Forward biased dissipation is more likely to be evenly distributed over the whole junction, so that it is permissible to send very large peak currents through the diode if they are of short enough duration. Note, however, that the thermal time constant of small diodes and transistors is very short, perhaps of the order of milliseconds. Thus, in many thermal accidents, the device appears to have been destroyed instantaneously.

In most applications it is easy to remain safely within the diode ratings published by the manufacturer. Most accidents occur because of a slip in working with the circuit. For example, putting the power-supply voltage across the diode in the forward direction. It takes only a moment's slip with a probe or component lead to have this happen.

5.5 Zener Diodes

Diodes can be produced with sharp breakdown characteristics and a stable breakdown voltage. Such diodes can be maintained safely in continuous breakdown, provided the power dissipation (i.e., the current) is limited. The common name, *Zener diode*, refers to one breakdown mechanism (the Zener effect) which is prevalent in the lower voltage ranges. These diodes are known as voltage-reference diodes or Zener diodes. A Zener diode can produce a constant, well-defined voltage from a variable voltage (as shown in figure 5.8) as long as the input voltage is larger than the Zener breakdown voltage. The symbol for a Zener diode is similar to a normal diode but with an extra diagonal bar on the cathode. Zener diodes are specified by their breakdown voltage, and the maximum power that they can handle. The type 1N4735 Zener diode (used below) has a nominal breakdown voltage of 6 V.

The most important use of Zener diodes is as *voltage reference* sources, for example to supervise the action of a stabilized power supply. In such use the breakdown current can be held quite constant. As an example, consider the rudimentary voltage stabilizer shown in figure 5.9. It can be viewed as a non-inverting amplifier whose input is the fixed voltage, V_Z. Then, if the amplifier is in its active range,

$$V_{OUT} = \left(\frac{R_1 + R_2}{R_1}\right) V_Z \tag{5.6}$$

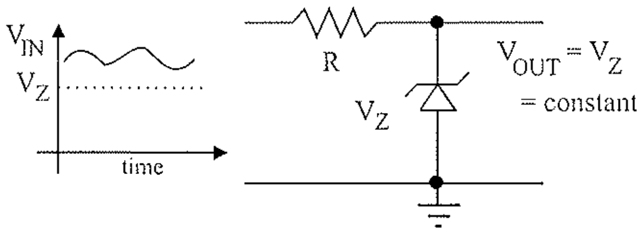


Figure 5.8: The zener diode generates a constant breakdown voltage.

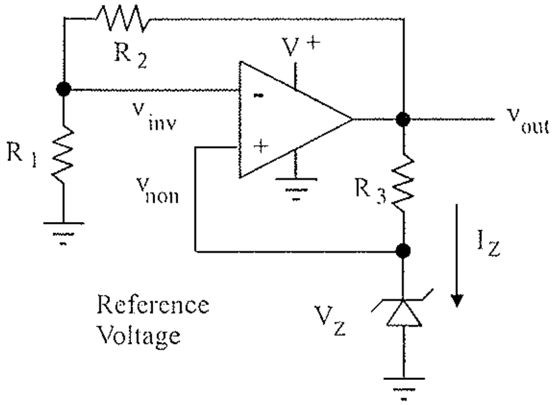


Figure 5.9: Zener diode voltage reference supply.

V_{OUT} is the stabilized output voltage, while the power supply for V^+ (which is the primary source) is permitted to vary by a small amount. The maximum output current of this circuit is limited by the capabilities of the op amp, but it is easy to insert a current boosting transistor (see chapter 6). Note that I_z, suitably determined by $(V_{OUT} - V_Z)/R_3$, is not affected by variations of V^+, thus enhancing the stability of V_Z.

5.6 Measuring the I-V Curve of a Diode

The characteristics of a semiconductor diode are best represented as an I-V curve which is a graph of the current I flowing through the diode as a function of the voltage V across the diode. You could measure this curve point-by-point using two of the hand held digital meters in the lab, however this procedure is a little tedious. If the diode current could somehow be converted into a voltage then the whole I-V curve could be displayed at the same time on the scope in XY mode using a triangle wave from the function generator. Forcing the diode current to flow through a series resistor will make the voltage across this resistor be proportional to the current through the diode. This voltage can then be observed on the oscilloscope.

There is a convenient form of integrated circuit amplifier called an *instrumentation amplifier* that will amplify the differential voltage between two points (i.e. the voltage across a resistor) with a well controlled, programmable gain. A common form of an instrumentation amplifier using three op-amps is shown in figure 4.27. A plain op-amp is also a differential amplifier but does not have a well controlled gain. The gain of an op-amp is just "very large" and you apply negative feedback to set it to a specific value. However, it is usually not a differential amplifier once you add simple forms of feedback. The AD622 instrumentation amplifier is shown in figure 5.10. The gain of this amplifier is determined by a single external resistor R_G connected between pins 1 and 8 as:

$$G = 1 + \frac{50.5\text{K}}{R_G} \tag{5.7}$$

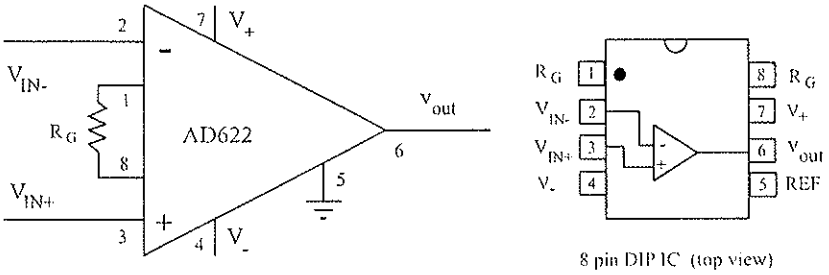


Figure 5.10: The AD622 instrumentation amplifier from Analog Devices

The AD622 is particularly convenient because, if you leave pins 1 and 5 unconnected (no R_G) then it has unity gain. You must also connect pin 5 (a reference voltage for the output) to ground. This IC can be run from supply voltages of ±2.6V to ±15V and has a bandwidth of about 800 kHz. As you might expect this amplifier is more expensive than a simple op-amp (about $4 each versus about $0.30 each for a 741, and $0.80 for a 3140), but can be worth it for some applications. Some other instrumentation amplifiers are the Linear Technologies LTC1920 and the Texas Instrument (Burr Brown) INA126 series.

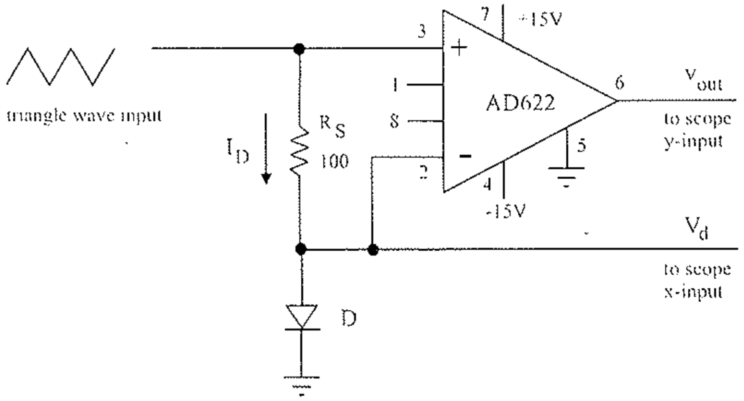


Figure 5.11: Display of the diode I-V curve on the scope.

Exp. 5.1 Set up the circuit shown in figure 5.11 using a 1N914 diode (note the orientation of the inverting and noninverting inputs). You should leave pins 1 and 8 of the AD622 unconnected to produce a differential gain of unity. Set the function generator to deliver a triangle wave going from -10V to +10V at a frequency of about 100 Hz to 1 kHz, and set the scope to XY mode (on the TEK1002 scope push the button labeled DISPLAY and set FORMAT=XY, then ch1=X and ch2=Y). Sketch the resulting diode I-V curve shown on the scope in you lab book. Record the voltage (in x and y) at several key points on the curve and convert each of these to an actual voltage and current in the diode. (You may find the origin of the graph on the scope by briefly connecting both probes to ground.)

Repeat this procedure for a 1N4735 Zener diode. Measure the Zener voltage. end

5.7 Rectification

The most common application of diodes is in the *rectification* of AC, to produce DC. Most power distribution systems use AC (alternating current) to transfer power across the country. You are probably very familiar with plugging appliances into a socket on the wall. The plug on the wall delivers 120 volts (rms volts) of AC power at a frequency of 60 Hz (in the USA, and 50 Hz in Europe). One of the primary reasons for using AC power is the existence of a device called a transformer (see schematic symbol in figure 5.12) that can easily transform AC voltages and currents.

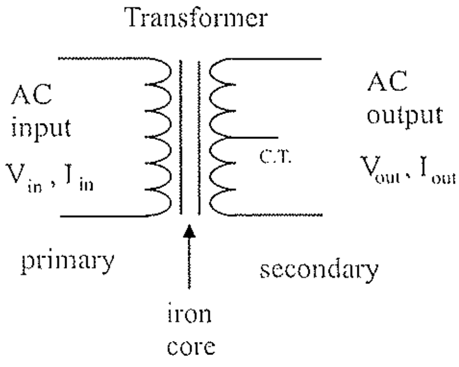


Figure 5.12: The transformer.

A transformer consists of two coils of wire wound around an iron core. An AC voltage and current (V_{in}, I_{in}) are applied to the primary winding (typically this is 120 VAC at 60 Hz). This produces an AC voltage and current on the output or secondary winding V_{out}, I_{out}. The power is transferred via the oscillating magnetic field in the iron core. If the primary has N_{in} turns and the secondary has N_{out} turns then the ratio of output to input voltage is:

$$\frac{V_{out}}{V_{in}} = \frac{N_{out}}{N_{in}} \tag{5.8}$$

Note that the total power must be conserved. This means that if the secondary voltage is lower than the primary voltage, then the output current will be proportionally higher than the input current. Some transformers may have an additional connection on the secondary winding called the center tap (labeled C.T.). The voltage at the center tap is half way between the voltage at the outer contacts of the secondary (similar to a voltage divider with equal resistor values).

The transformer is a convenient device for transforming a large voltage (120 VAC) back down to a low voltage (such as 15 volts) for an electronic circuit (using for example op-amps). However it is still an oscillating voltage and not a DC voltage as is required to power most electronic devices. The diode combined with the transformer secondary can convert the AC power back into DC power. Several diode rectifier circuits are presented next. Each rectifier circuit uses a simple resistor to simulate a load to make the circuits easier to test. In practice some other circuit or load could be substituted for this load resistor.

5.7.1 Half-Wave Rectification

A diode in series with the load (R_L) suppresses alternate half-cycles in v_{out} (figure 5.13). The result is a pulsating voltage of only one polarity. This can be smoothed to the required degree by adding a filter capacitor (see the section 5.7.5). When V_{SEC} is positive the diode conducts and v_{OUT} is equal to the secondary voltage less a diode drop. When V_{SEC} is negative the diode is off and v_{OUT} is zero (through the resistor). The peak value of v_{OUT} is the peak value of the secondary minus one diode drop.

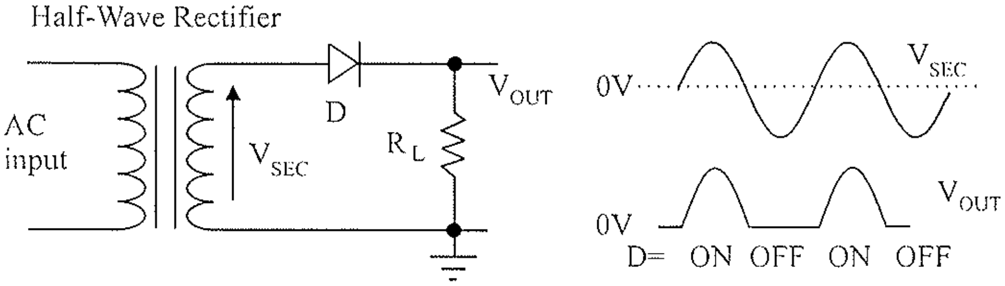


Figure 5.13: The half wave rectifier power supply.

The half-wave circuit (figure 5.13) has the major drawback that it only conducts for one half of the cycle. Also note that it drives current in only one direction in the transformer, which may cause the transformer to become magnetized in one direction and produce strange results.

5.7.2 Full-wave rectification with center-tapped transformer

This circuit (figure 5.14) consists of two half-wave rectifiers feed from AC voltages 180^o out of phase. They conduct alternately and produce a full-wave rectified waveform for v_{out}. Although a load current flows through the two halves of the transformer winding alternately, it goes in opposite directions and thus the magnetizing effect on the transformer cancels. The peak of the output voltage is now only one half of the peak value of the secondary voltage minus one diode drop. v_{out}, though still only pulsating DC, has a more solid look about it.

The center-tapped circuit has only a single diode's forward drop, however only half of the available voltage makes it to the output, which is not very efficient.

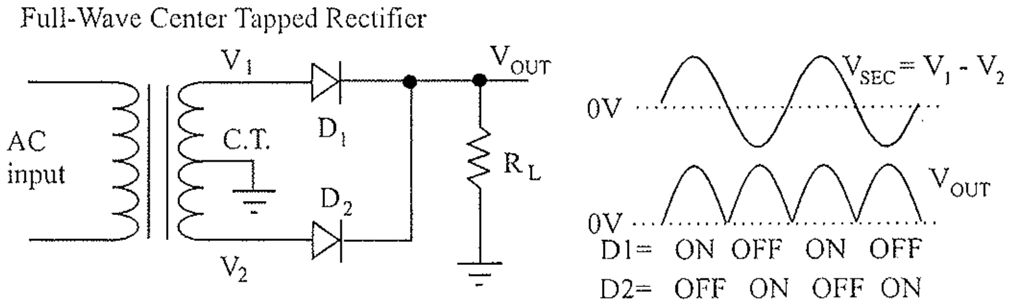


Figure 5.14: The full wave center-tap rectifier power supply.

secondary	ON	OFF
$V1 > V2$	D1, D4	D2, D2
$V1 < V2$	D2, D3	D1, D4

Table 5.1: State of the diodes in a full-wave bridge rectifier.

5.7.3 Full-wave bridge rectification

The full-wave bridge shown in figure 5.15 is probably the most common configuration. Diodes D1 and D4 conduct while the upper terminal of the transformer (V1) is positive, and diodes D2 and D3 conduct when the lower terminal is positive. The state of the diode are summarized in table 5.1. Figure 5.16 shows each half cycle of the transformer with the off-state diodes removed to illustrate the current path. For each half-cycle current is directed through R_L in the same direction. Now, however, the peak of the output is the same as the peak voltage of the secondary (instead of only one half the value as in the center tapped rectifier) minus *two* diode drops. This is the most efficient rectifier arrangement and also the most commonly used.

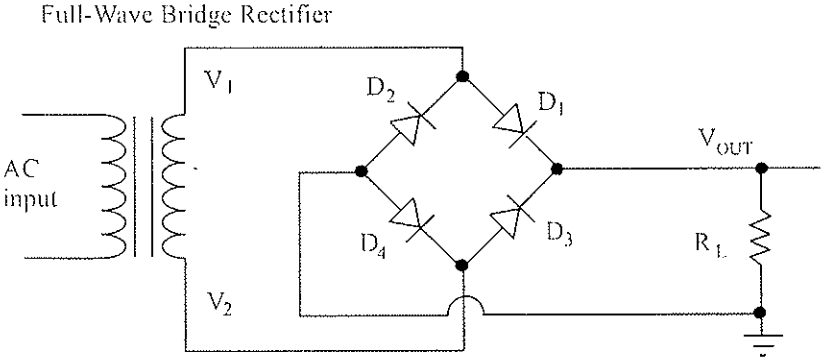


Figure 5.15: The full wave bridge rectifier power supply.

Note that, as in all rectifier circuits, the positive output terminal is the one which is supplied from the *cathode(s)* of the diodes. In the full-wave bridge circuit, current flows through the transformer

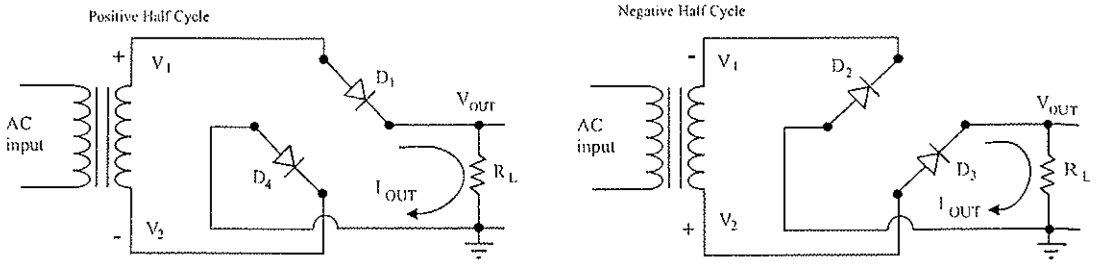


Figure 5.16: The full wave bridge rectifier power supply in each half cycle of the transformer with the off-state diodes removed.

during each half-cycle, so that there is no net DC component through the transformer. Note that both V_1 and V_2 have a significant voltage for alternating half cycles. To view the secondary voltage on the oscilloscope you should put one probe on V_1 and the other on V_2 and put the scope in differential mode.

A slight disadvantage of the bridge circuit is that there are two diodes through which the current must flow, and therefore a correspondingly larger forward voltage drop. Also, if one side of the load is grounded, neither side of the transformer can be connected to ground (since this would short one of the diodes). Use of a transformer actually becomes mandatory.

5.7.4 Voltage Doubler

Here the large capacitor, C, is in series with the input AC, but its output terminal looks into diode D1 to ground (see figure 5.17). Thus the capacitor is charged in such a way that its output never goes significantly below ground potential. v_{out} is then the whole sine wave, translated upward so that its negative peaks are at zero. (In the absence of a filter, D2 performs no useful function. A filter is added in the following section.) After several cycles the capacitor is charged to a voltage that is the peak value of V_{SEC}. If C is large this voltage (across C) is essentially constant and offsets the output by just the right amount to make the output always positive.

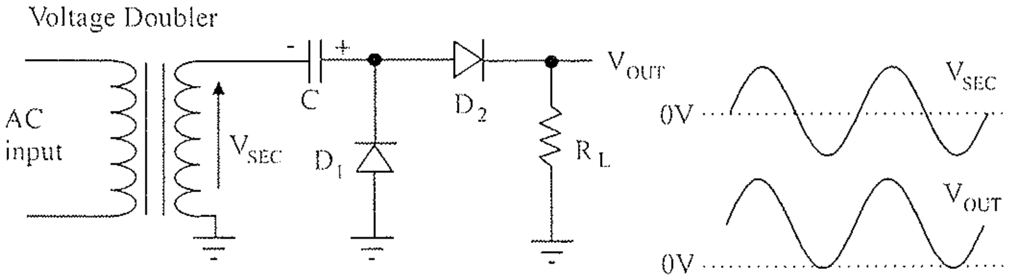


Figure 5.17: The voltage doubler.

5.7.5 Filter Capacitor

Pulsating DC (as above) is only part of the solution. A filter of some kind must be added to the output to convert it to a constant DC voltage. There are several types of filters, but the most common type uses only a single capacitor. A half wave rectifier with a filter capacitor is shown in figure 5.18.

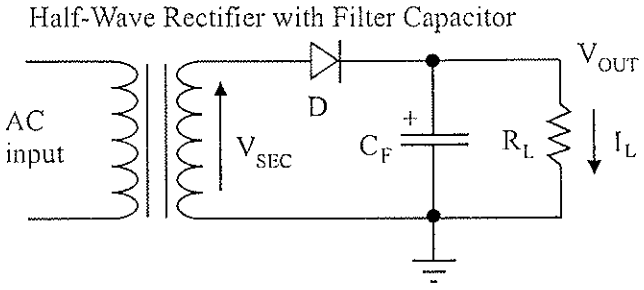


Figure 5.18: A half wave rectifier with filter capacitor.

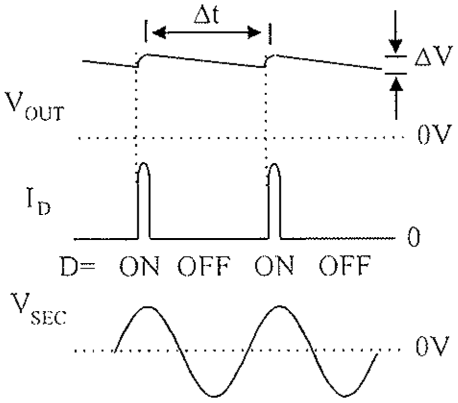


Figure 5.19: Voltages and currents in a half wave rectifier with filter capacitor.

A filter capacitor may be added to any of the rectifier circuits shown above. The distinguishing feature is that the capacitor acts as reservoir, sustaining v_{out} at an almost constant level between the peaks of the pulsating input (see figure 5.19). In practice C is usually made large enough so that only a small loss of voltage, ΔV, occurs between these peaks. Then the load current can be considered to be almost constant:

$$I_L \approx \frac{V_{out}}{R_L} \tag{5.9}$$

and the waveform for V_{out} decays linearly between the peaks at a slope of:

$$\frac{dV_{out}}{dt} = -\frac{I_L}{C} \approx \frac{\Delta V_{out}}{\Delta t} \Rightarrow |\Delta V_{out}| \approx \frac{I_L \Delta t}{C} \tag{5.10}$$

From this slope, and the time Δt between peaks, the peak-to-peak ripple, ΔV, in V_{out} can be calculated. Δt is $1/f$ for half-wave or voltage-doubler circuits, and $1/(2f)$ for full-wave rectifiers (f=60Hz). Full-wave rectifiers have relatively smaller ripple (for the same capacitance) because this frequency is twice as high as a half wave rectifier.

In each of the first three circuits, the peaks of the pulsating DC are at the peak voltage of the incoming AC. Hence the filter capacitor charges approximately to this peak voltage, and the DC output is the same. (The forward drop of one or two diodes must be subtracted, and the effect of the voltage loss ΔV taken into account, of course.) The rectifier diodes conduct only for a short part of each cycle, namely, when the AC voltage rises above the level to which v_{out} has decayed between cycles. Thus the charging current comes in short, intense bursts.

In the voltage doubler circuit, the peak of the pulsating DC is above ground by the peak-to-peak voltage of the AC, (i.e., by twice the amount of the previous arrangements). Hence the name.

$\boxed{\text{Exp. 5.2}}$ Using the small 6.3 V rms center-tapped transformer, construct the first three rectifier circuits (half wave, full-wave center tapped, and full wave bridge). The center tap should NOT be connected except in the full-wave center tapped rectifier circuit. Use type 1N4007 or 1N4001 rectifier diodes, and a 2.2 kΩ load resistor. Observe the input AC voltage(s) at V_{SEC} (noticeably distorted sine waves) and V_{out}, first without any filter. Account for the waveforms and amplitudes of each of your circuits. Note that, in the full-wave bridge rectifier both sides of the transformer have a non-zero voltage for part of each cycle, so you have to use two scope probes to observe the transformer voltage (one probe on each side of the transformer).

Choose one of the rectifier circuits and add a 100 μF electrolytic capacitor across R_L, being sure to observe the polarity markings. Measure the resulting DC output voltage and the peak-to-peak ripple, and compare with calculated values. (Comment: The capacitance of an electrolytic capacitor can be quite far from the marked value.)

OPTIONAL: Repeat this experiment using the voltage doubler rectifier circuit with a 100 μF electrolytic capacitor. Be sure to connect the terminal marked + to the diode side of the circuit. $\boxed{\text{end}}$

In many cases the ripple on the capacitor is small enough to permit use of the DC without further filtering. It is usual to include a voltage regulator (or stabilizer) in most power supplies, and this can be counted upon to perform further smoothing of the final voltage. If an additional passive filter is needed it is usually an L-C filter, but in some cases a low-pass R-C circuit can be used.

Voltage regulators are available in integrated circuit form. (e.g., the 7800 series for positive voltages and the 7900 series for negative voltages). These take a DC voltage with ripple (such as produced by a rectifier with a filter capacitor) and produce a stable output voltage with a well determined DC voltage. This is similar to a Zener diode but a regulator will provide a much higher current (one or more amps).

5.8 Diode Circuit Notes

Before discussing more examples of general diode applications, here are a few guidelines.

(i) To exploit the nonlinear (ON/OFF) characteristics of a diode, the signal must swing through a *large enough* voltage to drive the diode over an appreciable portion of the curve. The nonlinearity is characterized by the voltage V_T^*. Hence voltage swings small compared to this voltage will see the diode as a substantially linear element.

(ii) Since a diode conducts in one direction only, any useful current through it necessarily has a DC component. Hence there usually must be a DC *return path* around every diode. Other diodes can be included in this DC path, provided they all point the same way. Sometimes a switch which is closed at regular intervals can serve as part of the DC path. (If there is no DC path, series capacitors charge to such a level that no further diode current flows at all.)

(iii) Diode circuits are polarity invertible. All associated elements must, of course, be inverted at the same time: battery voltages and bias currents, and such devices as transistors. When an example of a diode circuit is shown, you should picture it yourself for the opposite polarities.

5.9 Clipping Circuits (Limiters)

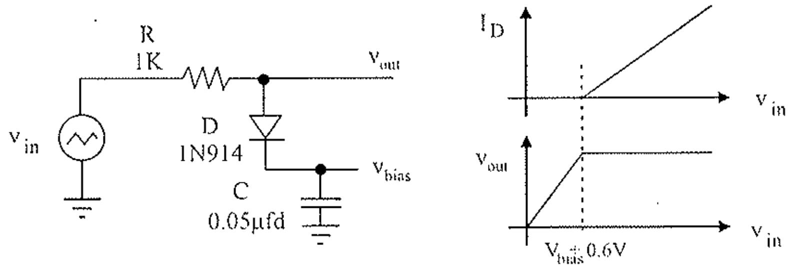


Figure 5.20: Diode clipping circuit (left). The diode is off (diode current I_D=0) until the input v_{in} reaches $V_{bias} + 0.6V$ and then the diode turns on (increasing current) and the output is clamped to be less than $V_{bias} + 0.6V$.

Exp. 5.3 Observe the performance of the circuit shown in figure 5.20. Apply a triangle wave of about 10 V_{PP} at v_{in} and use a positive adjustable DC power supply for V_{bias} in the range 2 to 10 volts. Correlate the portion of V_{in} that is clipped with the value of V_{bias}. **end**

The diode in figure 5.20 conducts when v_{in} exceeds V_{bias} by more than about 0.6 V. Then the forward resistance of D rapidly decreases to a low value, and v_{out} cannot rise further. The excess of v_{in} appears across R. The choice of this series-limiting resistor is often difficult. R should be large compared to the diode resistance to make the limiting action effective. In many applications it should also be large enough to avoid loading the source of v_{in} significantly once the diode conducts. (In other cases, loading of the source doesn't matter, and perhaps the internal resistance of the source is large enough to make an external R unnecessary.) On the other hand, R appears in series with v_{out}, causing a voltage drop if an external load resistor is to be driven, and producing a low-pass (slow rising) action if the load has significant capacitance to ground. These are reasons for keeping R small.

The source for V_{bias} has current driven into it, through the diode, in an unnatural direction. For example, a stabilized power supply might find it difficult to accept this current if it were not negligibly small. Evidently the size of R, combined with the fraction of the time that v_{in} spends above V_{bias}, and by how much, determines the average diode current. To make a stabilized supply happy, it would suffice to attach another load which draws at least as much current as the diode will ever deliver. The bypass capacitor, C helps smooth out the transcients and maintain a fixed DC level at V_{bias}. When you use a resistive voltage divider as a source of V_{bias}, the average diode current must be considered in the design as a load current which may falsify the divided voltage.

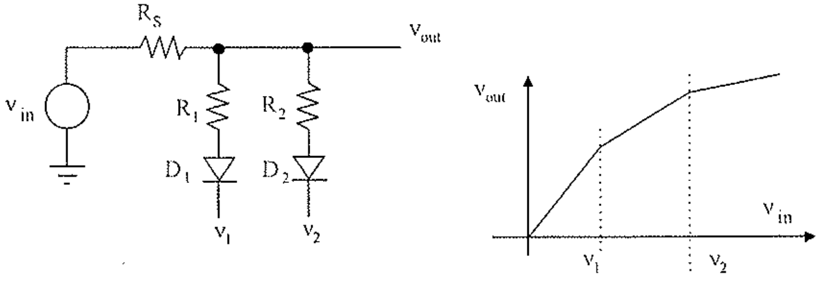


Figure 5.21: Diode waveshaping.

The diode limiting action can be "softened" by including a series resistor with the diode as shown in figure 5.21. Then the curve for v_{out} versus v_{in} changes its slope as soon as the diode conducts. Several such softened limiters can be included in parallel. Suitable choice of the bias voltages $V_1, V_2, \cdots$ and the resistors $R_1, R_2, \cdots$ allows the approximation of any desired curve (remember that any diode once conducting continues to conduct as v_{in} changes further). Of course reverse diodes and negative bias voltages can also be included. The circuit often serves as the sine wave portion of the function generator, where sine waves are made from triangles by this technique.

5.10 AC Coupling (Level Shifting), DC Restoration

The high-pass R-C circuit (see schematic in figure 5.22) can be used for *AC coupling*, to isolate the DC levels of the source and the load. The capacitor blocks the DC level of v_{in} but allows the AC portion of v_{in} to pass through to v_{out} (for sufficiently high frequencies). To make the coupling transmit the waveform as faithfully as possible, the time constant RC should be long compared to the variations of the signal. This may mean, for example, that the coupling transmits a single pulse quite well. Yet, when many such pulses follow each other with a significant *duty cycle* (= fraction of time high), a cumulative distortion of the waveform takes place. An example of this is sketched in Fig. 5.22 for rectangular pulses.

After the initial transient, the output pulses settle down to a level in which their *DC component* is zero (or whatever other voltage the lower end of R is tied to). This is true because, on average, no DC can be transmitted by C and the DC level of the output is zero because R is tied to ground. This is the reason why the circuit is said to have AC coupling.

The level shift caused by loss of the DC component can sometimes be accommodated, but often it is objectionable. It depends, moreover, on the duty cycle of the pulses. If these pulses arrive at a variable rate, the level for v_{out} varies in response (with a transient time of order RC).

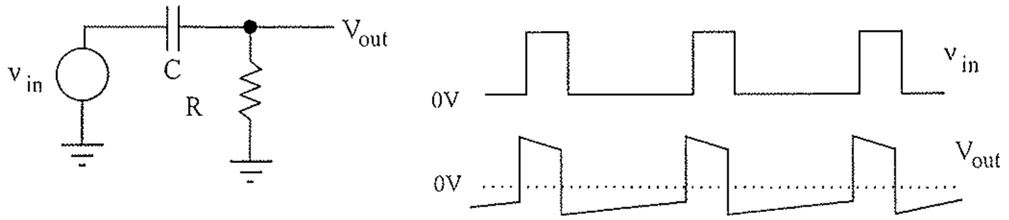


Figure 5.22: AC coupling.

When v_{in} has a well-defined lower (or upper) level, the DC component can be restored to v_{out} by a diode connected as shown in figure 5.23. Each individual pulse still suffers a small amount of distortion (tilt at the top and bottom), but when v_{out} attempts to undershoot at the trailing edge of the pulse, D starts conducting and recharges C so that v_{out} is once again above zero (actually a diode drop below ground). Then the next pulse starts from the same, standardized level, and no cumulative level shift occurs. The circuit is, in fact, a clipping circuit in which the series element is a capacitor instead of a resistor, permitting some DC isolation between v_{in} and v_{out}.

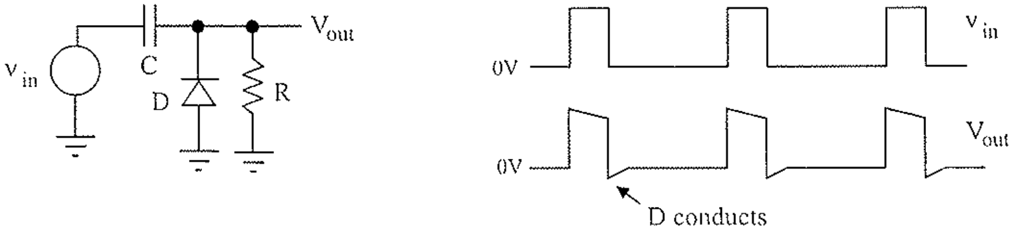


Figure 5.23: AC coupling with DC restoration.

Exp. 5.4 OPTIONAL: Assemble the circuit shown in figure 5.23, using C= 0.05 μF and R= 10 KΩ. Use square waves of about 2 volts peak-to-peak at high and low frequencies to check the circuit. With high frequencies vary the DC offset of the input and observe what happens to the output voltage. **end**

The two sketches in figure 5.24 show two cases of DC restoration to a non-zero value by adjusting the lower and upper levels of v_{out}. In each case the DC level of v_{in} can be arbitrary, and the level to which v_{out} is restored is chosen by means of V_{bias}.

If a signal is passed through several AC-coupled stages in succession, it may be necessary to add DC restoration at each stage. This avoids level shifts which might be locally harmful, and may also avoid the presence of such cumulatively compounded transients at the output as would make the life of a single DC restorer pretty hard. However, it is pointless to use DC restoration before the signal level is large enough (compared, as usual, to V_T^*)

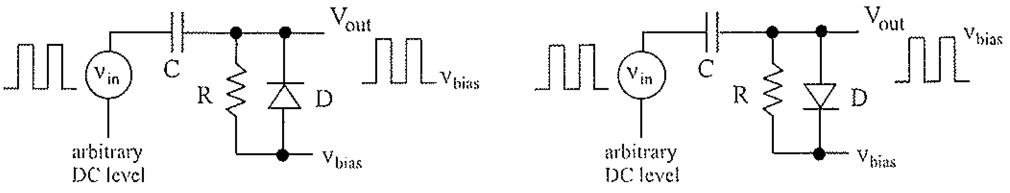


Figure 5.24: AC coupling with DC restoration.

5.11 Pickoff Circuits, Pulse Stretching

The arrangement shown in figure 5.25 is called a pickoff circuit because v_{out} picks off only that portion of v_{in} that exceeds V_{bias}. This would be useful as a threshold device to reject small noise voltages, for example. Note that V_{bias} appears at the output when v_{in} is below threshold.

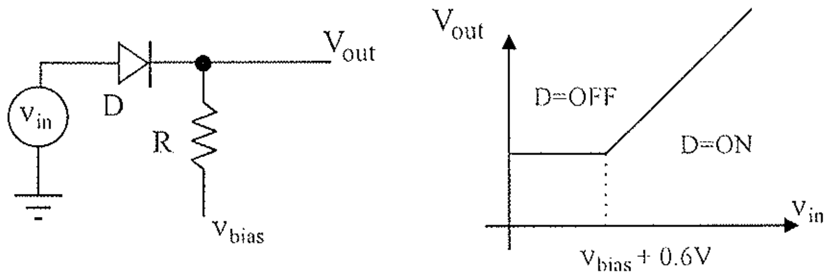


Figure 5.25: Diode pickoff.

If a capacitor is added to R, the pickoff diode charges it to the peak voltage of v_{in}, (less a diode drop of approx. 0.6 V). A large positive pulse of v_{in} charges the capacitor through the diode but when v_{in} returns to zero the diode is reversed biased so the capacitor cannot discharge through the diode but only through the resistor. This gives a long exponential tail, with time constant RC, to the pulse at v_{out}. This is called pulse stretching (see figure 5.26).

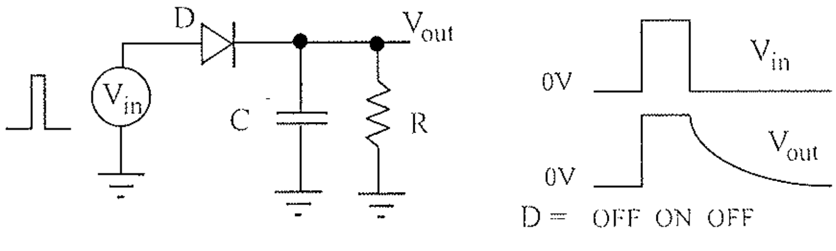


Figure 5.26: Pulse stretcher.

The shape of v_{out} may be changed by discharging the capacitor differently. In the next example (shown in figure 5.27) R is returned to a large negative voltage $-V_{bias}$, in effect producing a current source $I_{bias} = V_{bias}/R$. Then v_{out} slopes back to zero nearly linearly, with $dv_{out}/dt = -I_{bias}/C$. The output is nearly linear because it is only the initial portion of an exponential decay to a final value of

$-V_{bias}$ and not to 0 V. Before the output reaches $-V_{bias}$ the diode turns on and prevents v_{out} from going more than a diode drop below v_{in} (the capacitor is discharged through v_{in} and the diode at the end of the pulse). Note that v_{out} in this case rests about 0.6 V below ground potential, because of the continuous flow of I_{bias} through D.

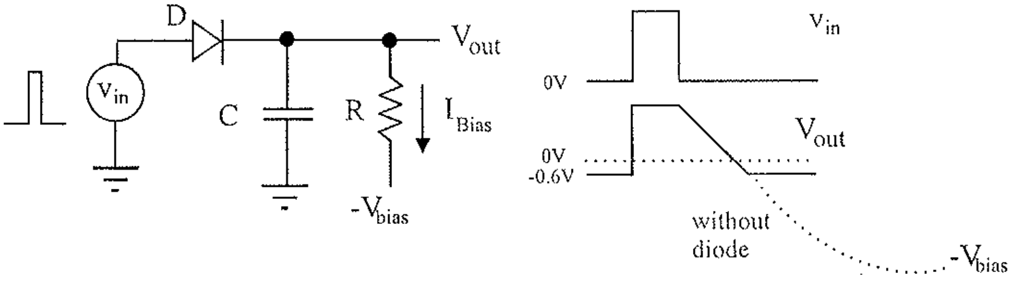


Figure 5.27: Pulse stretcher.

Pulse stretching requires a large burst of current from the source to charge C. Depending on the pulse duration and the size of C, there may also be a significant forward voltage drop in D. The final voltage on C may then be significantly lower than the (unloaded) peak voltage of v_{in}.

Suppose the input pulse is very narrow compared to the discharge time. The total pulse duration is then proportional to the pulse amplitude (if I_{bias} is really constant, as can be assured with the help of a more sophisticated current source). The pulse stretcher is then an elementary *amplitude-to-time converter*. v_{out} can be reshaped into a rectangular pulse by applying it to a comparator (see chapter 2), perhaps with $V_{ref} = 0$.

In the third example shown in figure 5.28, a switch S discharges C some time after the pulse has gone. S can be a mechanical switch or (more commonly) some form of transistor. The output charges to the peak value of v_{in} but when v_{in} decreases the capacitor cannot discharge through the diode (because it is reverse biased). The capacitor retains the peak value of v_{in} until the switch is closed (or the capacitor slowly discharges through some parasitic resistance in the circuit). The output is frequently buffered with a voltage follower to avoid loading the capacitor.

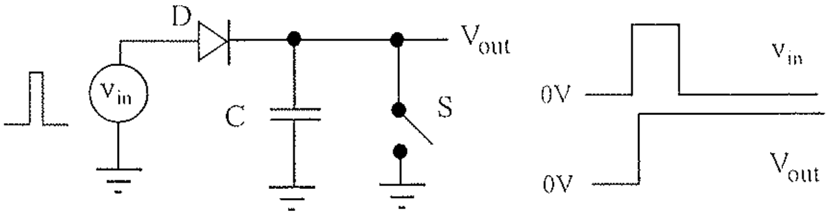


Figure 5.28: Peak detector.

5.12 Waveshaping with Circuits Containing Diodes

The sharp corners produced by the nonlinearity of the diode can be used to shape the waveform in a variety of ways. The pulse stretcher shown in figure 5.27 (with constant current discharge of the capacitor) is one example. A somewhat analogous action occurs in the circuit shown in figure 5.29.

Note that the position of the diode and capacitor have been reversed. Positive-going edges of v_{in} drive up v_{out}. When v_{out} is positive the diode is off and the capacitor C then discharges at a nearly constant rate, producing a pulse with sloping trailing edge. This edge returns to the base level (-0.6 V) smartly. The negative going step of v_{in} drives the diode hard into conduction, but produces only a small "wiggle" in v_{out}. Of course the source for v_{in} is heavily loaded at this time because it is charging C rapidly.

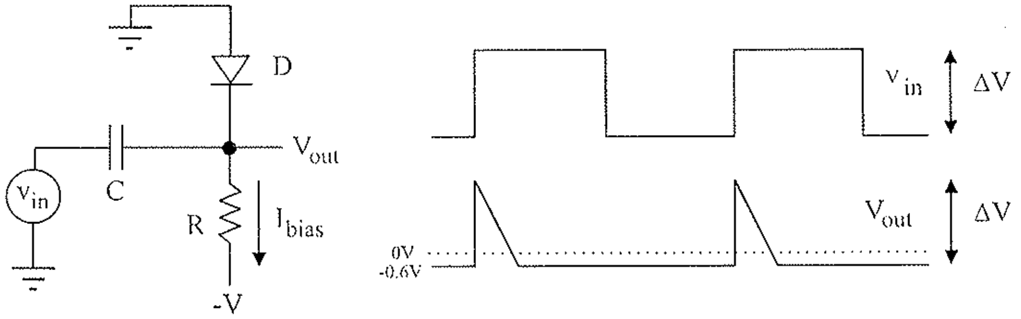


Figure 5.29: Edge detector.

If v_{out} is applied to a comparator ($V_{ref} = 0$) you can obtain a short, rectangular pulses whose duration is fairly well defined. This is a practical method for producing such pulses, synchronized with the positive-going edges of a given square wave.

$\boxed{\text{Exp. 5.5}}$ Design and build a circuit similar to that shown in figure 5.29 which accepts a 2 V p-p square wave at 200 Hz as input. V_{out} should be connected to the input of a 3140 op-amp comparator. Design the circuit to produces rectangular output pulses (output from comparator), going from *roughly* -5V to +5V, and 200 μs wide, synchronized with the negative-going edges of the input. (Note that this is the opposite of the example shown above.) Save this circuit for the next experiment. $\boxed{\text{end}}$

When you have this circuit working, you might as well use it as a pulse generator to try the action of a pulse stretcher:

$\boxed{\text{Exp. 5.6}}$ Apply the rectangular pulses from your circuit in the previous experiment to a pulse stretcher, as in Fig. 5.26, with exponential capacitor discharge, of time constant about 1 ms. Connect the lower end of R to -5 V to make sure the capacitor discharges to this level. Choose C small enough that it charges substantially to the peak voltage of the rectangle. (Experiment with different values of the resistor, leaving C fixed.) $\boxed{\text{end}}$

5.13 Diodes in the Operational Configuration

It is also possible to use a diode in the feedback path of an op-amp to produce a system in which v_{out} is proportional to $\log(v_{in})$ as in the circuit shown on the right in figure 5.30. The inverting input v_{inv} is still a virtual ground (if v_{in} is positive and v_{out} is negative) so v_{out} cannot be very large (in magnitude). In fact it must be within about a diode drop of ground. This is one instance where approximating the diode as a simple drop of 0.6 volts (when forward biased) is not sufficient. First consider the current I_R produced by a positive input voltage that flows into the resistor.

$$I_R = \frac{v_{in} - v_{inv}}{R} = \frac{v_{in}}{R} \tag{5.11}$$

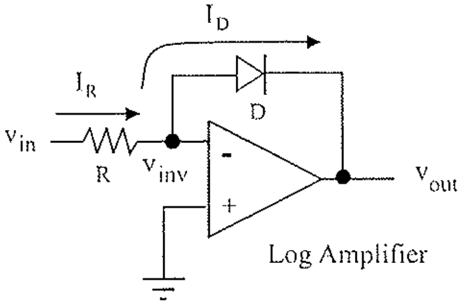


Figure 5.30: Log amplifier.

This current must also flow through the diode (the op-amp inputs draw no current). The current through the diode is given by the diode equation:

$$I_R = \frac{v_{in}}{R} = I_D = I_0 \left[\exp\left(\frac{v_{inv} - v_{out}}{V_T^*}\right) - 1\right] = I_0 \left[\exp\left(\frac{-v_{out}}{V_T^*}\right) - 1\right] \tag{5.12}$$

Taking the log of both sides and rearranging slightly gives:

$$v_{out} = -V_T^* \log\left[\frac{v_{in}}{I_0 R} + 1\right] \approx -V_T^* \log\left[\frac{v_{in}}{I_0 R}\right] \tag{5.13}$$

where the 1 inside the square brackets can be neglected because I_0 is usually a very small number. The output voltage is approximately the logarithm of the input voltage if the input is positive. This circuit can be used to multiply two or more analog signals together if it is combined with an op-amp summer and an anti-log circuit (switch the position of the resistor and diode in the log-amp circuit).

The simple log-amp circuit with a single diode is decidedly one-sided. It only works with a single polarity of v_{in}. The two-diode circuit shown in figure 5.31 produces more symmetrical results for both positive and negative input voltages. Investigate its performance in the next experiment. Note, if you carefully measure v_{out} vs. v_{in} and plot it on a semilog scale you may be able to extract the diode parameters I_0 and V_T^*.

> *NOTE: You only have to do one of the following two experiments. Choose one of Exp. 5.7 or Exp. 5.8. The other is optional. However you are responsible for understanding BOTH circuits.*

Exp. 5.7 Build the circuit shown in figure 5.31 using a 3140 run from ±15 volt supplies, and investigate its performance. Use various waveforms for v_{in}. Also observe v_{out} versus v_{in} directly on the scope using the xy display mode (one input channel drives the horizontal sweep and the other drives the vertical sweep). **end**

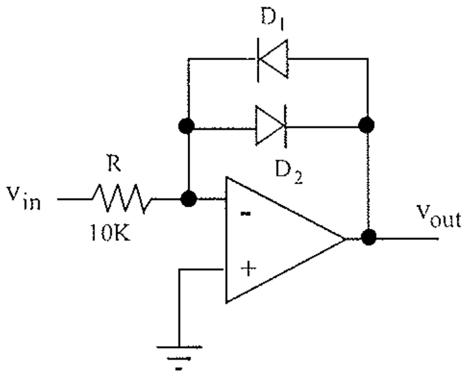


Figure 5.31: Bidirectional log amplifier.

A related but different circuit is shown in figure 5.32. This circuit simulates a single ideal diode (with an inverted output). One of the diodes is directly in the feedback path and the other diode supplies the output with a resistor in the feedback path. The output v_{out} is taken from the point that is "supervised" directly by the feedback. The 0.6 V forward voltage drop across D2 then becomes part of the amplifier and does not appear at v_{out}. The result is that the transfer function, v_{out} versus v_{in}, appears as shown, which is the performance curve for an ideal diode with an addition inversion due to the op-amp.

Exp. 5.8 Build the ideal diode circuit shown in figure 5.32 using a 3140 op-amp run from ±15 volt supplies. Check the operation of this "ideal diode" circuit by applying a sine wave for v_{in}. Observe carefully what happens to v_{out} at the "corners". Do you observe any departure from ideal performance. Try both low and high frequencies to see if there are response time limitations. Make a sketch showing v_{in}, v_{op}, and v_{out} on a common time axis. Describe how the circuit works. **end**

This circuit is easier to analyze starting from the point labeled v_{op} and working out to the input and output terminal (see fig. 5.33). When you see two diodes connected to a single node (v_{op}) and pointing in opposite directions (as D1 and D2 are) it usually means that the designer was trying to make the circuit do two different things depending on the sign of the common node (v_{op} in this case). When v_{op} is positive D1 if OFF (the other end is connected to a virtual ground) and D2 is ON. The output of the op-amp flows to v_{out} and then this is just an inverting amplifier with a gain of $-R_2/R_1 = -1$. When v_{op} is negative D1 is ON (and supplies the feedback for the op-amp) and D2 is OFF. Now v_{out} is set to zero via R_3 and R_2 (to virtual ground). You may wish to redraw this circuit for each case leaving out the diode that is OFF to visualize what is going on.

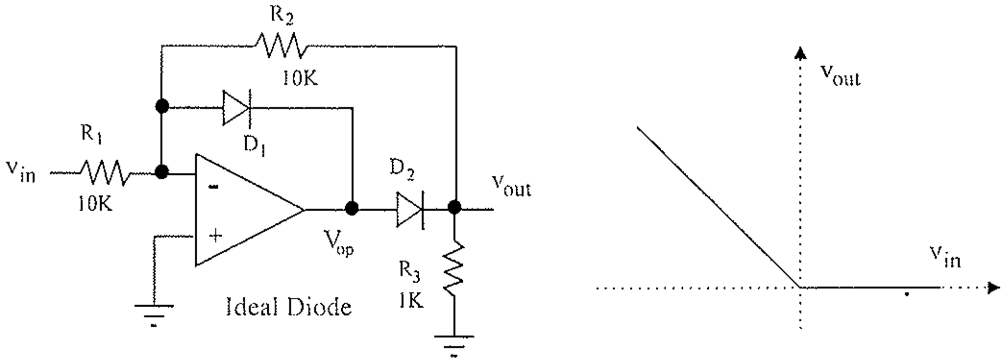


Figure 5.32: Precision rectifier circuit.

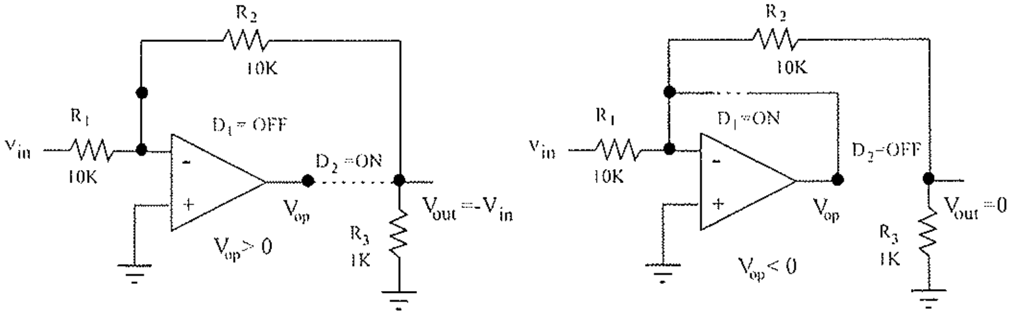


Figure 5.33: Precision rectifier circuit, showing diode action for each polarity of v_{op}.

5.14 Incremental Diode Resistance

Describing diode performance in a given circuit context is best done by specifying the forward voltage drop, V, associated with the required forward current, I. Usually it isn't particularly helpful to talk about the diode's DC resistance, V/I. The current I and the voltage V move over an extremely nonlinear curve (as in figure 5.34), and the ratio of V to I varies tremendously in the process. In some applications, however, the diode is placed at some specified *operating point* (also called a *quiescent point* or *Q-point*) by a certain forward current, I. In some circumstances it is necessary to know what happens when a small AC signal is superimposed on this quiescent current (or *bias* current).[1] The signal makes only small excursions on the V-I curve. Over these small excursions, the curve can be regarded as a straight line. The slope of this line then defines the incremental resistance (when forward biased) of the diode as:

$$r_f = \frac{\Delta V}{\Delta I} \approx \frac{dV}{dI} \tag{5.14}$$

Note that r_f can vary greatly with operating conditions. The diode obeys the diode equation:

$$I \approx I_0 e^{V/V_T} \tag{5.15}$$

[1] It is sometime more efficient to think of a forward bias *current* rather than a voltage, because the V-I curve is so very steep. Applying a fixed forward voltage would leave the current very poorly determined, since it changes by a factor of e for every 40 mV error in V!

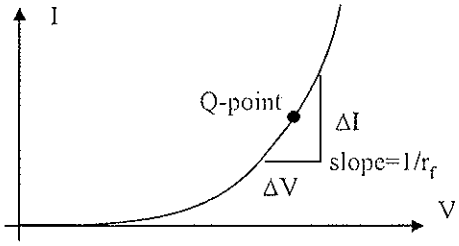


Figure 5.34: Dynamic diode impedance.

so that:

$$\frac{dI}{dV} = \frac{1}{V_T^*} I_0 e^{V/V_T^*} = \frac{I}{V_T^*} \qquad \text{or} \qquad r_f = \frac{dV}{dI} = \frac{V_T^*}{I} \tag{5.16}$$

For example, if a diode is forward-biased by a current of 1 mA, and a small AC signal is superimposed on this current, the incremental resistance will be about (40 mV)/(1 mA) = 40 Ohms. If the forward current is increased r_f should decrease in inverse proportion. Ultimately, however, the simple ohmic elements in the diode structure set a lower limit on r_f. These include the contacts and the bulk resistance of the semiconductor material. Together they account for what is called the *spreading resistance*, R_s.[2] The total forward incremental resistance is then:

$$r_f = \frac{V_T^*}{I} + R_S \tag{5.17}$$

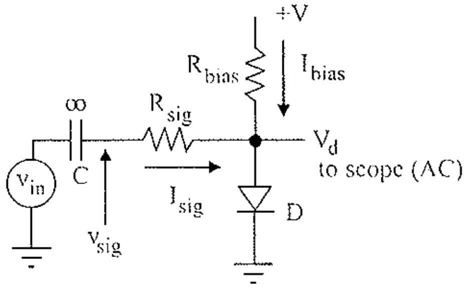


Figure 5.35: Measuring the dynamic diode impedance.

Although you will not be asked to verify every result experimentally, it is instructive to think about how you would measure r_f (see figure 5.35). First establish I_{bias} by suitable selection of V_{bias} and R_{bias}, preferably with $V_{bias} >> 0.6V$, so that I_{bias} comes effectively from a current source. Next add a signal current I_{sig}, again using a sufficiently large v_{sig} that you can consider the signal to come from a current source. Then observe v_d (the AC component of the diode voltage) on a scope. How large should you allow v_d to become? Note that the diode's nonlinearity is characterized by the voltage V_T^*. Hence v_d should be much less than this characteristic voltage, so that the AC excursions can be considered to be taking place on an almost straight piece of the curve. For example, v_d= 5 mV$_{p-p}$ would be a suitable choice. (To observe this on the scope you must make a direct connection, via coax cable, and not

[2] R_s is of order a few ohms in a small signal diode, such as the 1N914. It is much lower in power diodes.

	1N914	1N4001	1N4007
application	small signal	power rectifier	power rectifier
max. reverse voltage	75 V	50 V	1000 V
max. forward current	200 mAmp	1 Amp	1 Amp
max. reverse saturation			
current (25°C)	25 nAmp	10 μAmp	10 μAmp
response time	4 nSec	1 μSec	1 μSec

Table 5.2: Typical specification for some diodes

use the attenuator probe.) The capacitor marked "∞" is inserted to prevent I_{bias} from being diverted through the signal generator, although in practice the diode voltage is so low that not much of this would happen, anyway. The reactance of this capacitor at the signal frequency should preferably be much less than R_{sig}, but just to avoid error you can observe v_{sig} on the diode side of the capacitor. Then you see the voltage actually driving I_{sig} through R_{sig}, regardless of possible loss in the capacitor.

The voltage drop of a diode, at given forward current, exhibits a negative temperature coefficient. Though this varies somewhat among types of diode, it is usually close to:

$$\frac{dV_f}{dT} \approx -2.5\frac{mv}{^oC} \tag{5.18}$$

This implies that you should not rely on knowing V from dead-reckoning. For example, it takes a temperature change of only 15°C to change V by about 40 mV, with a consequent change in I by a factor of e. Often it is possible to arrange two diodes in a circuit in such a manner that the temperature-dependent changes of their voltage drops tend to cancel each other.

The above discussion about diode behavior applies in most respects to junction transistors, too, which makes these remarks doubly useful. Groups of diodes and transistors can be formed on a common semiconductor chip. In that case the temperature compensation that is obtained through the use of balanced circuits can become impressively accurate. The two "competing" elements are not only made of the same material, they are also in very close proximity and thus likely to be at the same temperature. This is how the temperature dependence of the offset voltage, V_{io}, in a balanced-input differential op amp is kept so low.

5.15 SOME DIODE CHARACTERISTICS

Some typical diode parameters are shown in figure 5.2 for a few types of diodes.

5.16 Practice Problems

[1] In the circuit shown in figure 5.36 find the voltages V_2 and V_3 if V_1= 5V R_1= 2.7K, and R_2= 3.3K. You may assume a diode drop of 0.6 V.

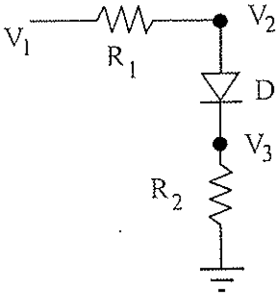


Figure 5.36: Problem 1.

[2] In the circuit shown in figure 5.37, R_1= 10K, V_1= 3V and V_2= 2V. The input is driven with a 10 V_{PP} triangle wave centered about ground. You may assume a diode drop of 0.6 V.
a) On the same time scale sketch several cycles of V_{in} and V_{out}, clearly indicating their relationship in time.
b) On your sketch of V_{out} indicate whether D_1 and D_2 are on or off during each portion of the waveform.
c) What is the minimum value of V_{out} (in volts)?
d) What is the maximum value of V_{out} (in volts)?

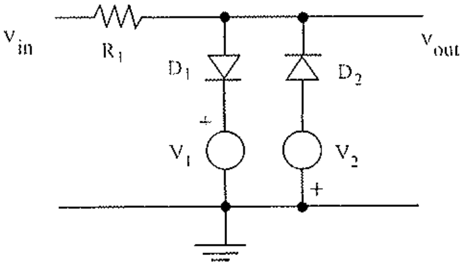


Figure 5.37: Problem 2.

[3] Design a full wave bridge rectifier power supply using a transformer at 60Hz. Your supply should deliver a positive output voltage of +10 volts (DC) with less than 0.1 volts ripple at a load current of 75 milliAmps. Sketch the whole circuit, label all components and calculate a value for the secondary voltage (in V_{RMS}) and the filter capacitor (in μfD). You may assume a diode drop of 0.6 volts.

[4] In the circuit shown in figure 5.38 you may assume a diode drop of 0.6 volts and that the op-amp is ideal and run from ±5 volt power supplies. The component values are R_1=100K, R_2=10K, R_3=47K, R_4=8.2K and C=0.01μfd.

a) Copy the adjacent table and fill in the four missing spaces to indicate whether each diode is ON or OFF.

V_{out}	D_1	D_2
+5V		
-5V		

b) How long does v_{out} stay high (in μSec)?

c) How long does v_{out} stay low (in μSec)?
d) Sketch v_{out} versus time.

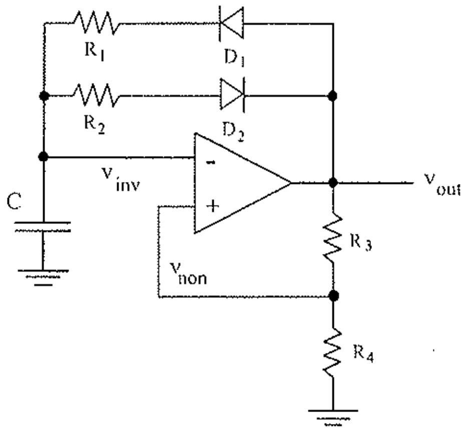


Figure 5.38: Asymmetrical oscillator.

[5] In the circuit shown in figure 5.39 is an improved peak detector (compare to figure 5.28). Assume that the switch S is initial closed for a long time and then opened just before the input pulse v_{in} of 0 to 2 volts is applied. The resistor R is to limit the maximum current drawn from the op-amp. You may assume that the op-amp is ideal and driven from ±15V power supplies, a diode drop of 0.6V, and C is small enough to charge almost immediately with the available current from op-amp $A1$ (its precise value doesn't otherwise affect this problem).
a) How is v_{out} related to V_2?
b) On the same time scale sketch v_{in}, V_1 and v_{out}. Op-amp $A1$ may be saturated for some portion of this time. What is the max. and min. value of V_1 and v_{out} (in volts)?
c) How does the diode drop influence the peak values detected at v_{out}? Why?

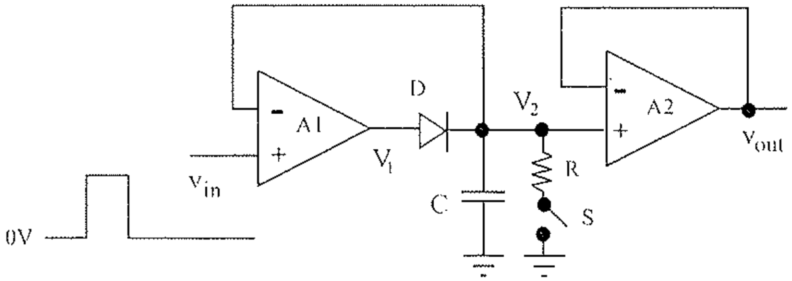


Figure 5.39: Improved peak detector.

Chapter 6

THE BIPOLAR JUNCTION TRANSISTOR(BJT)

NOTE: You will probably find that this is the longest and hardest chapter of the whole semester. Most people find that the material after this chapter gets easy.

Brinkman, Haggan, and Troutman (IEEE Journ. Solid-State Circuits, Vol. 32 No. 12, Dec. 1997, p. 1858-1865) has given a short history of the invention of the transistor and integrated circuit if you are interested.

6.1 Basic Mechanism

Picture a p-n junction diode with reverse bias voltage applied to it as shown in figure 6.1. This bias drives the *majority carriers* (electrons in the n region and holes in the p region) away from the junction. Electrons on the n side are drawn to the positive side of the battery, and holes on the p side are drawn to the negative side of the battery. There is essentially no current flowing because the majority carriers have been removed from the junction. A small number of minority carriers (electrons in the p region and holes in the n region) may be generated thermally, and may survive long enough to diffuse to the junction and be swept across it by the applied voltage. This causes a very small reverse current, I_0 to flow due to *minority carriers*.

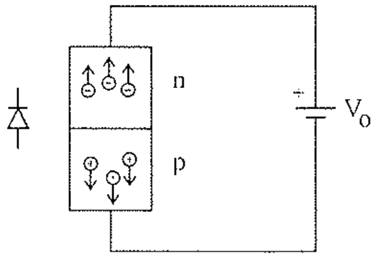


Figure 6.1: A reverse biased pn junction.

At room temperatures I_0 is a very small current, of order 10^{-9} A in small diodes. The reverse current could be increased if additional minority carriers were somehow created near the junction. For example, light of a suitable frequency creates minority carriers photoelectrically. Shining such light on the device increases the reverse current. This is how a *photodiode* works.

Minority carriers can also be *injected* into the reverse biased diode by placing another forward biased junction nearby. In the NPN example shown in figure 6.2, the second junction (labeled base and emitter on the bottom) is very close to the first (labeled collector and base on the top) and is attached to the p side of the diode. Forward current I_B (electrons traveling upward) is driven through the emitter-base junction on the bottom by the voltage V_{BB}. These electrons are injected into the p region of the reverse-biased diode (collector-base junction). Electrons are minority carriers in the p region. They can diffuse across the base, reach the reverse-biased collector-base junction, and get swept into the reverse biased junction on the top. If the voltage V_{CC} is made much larger than V_{BB} then most of the electrons that start in the emitter end up in the collector and not the base.

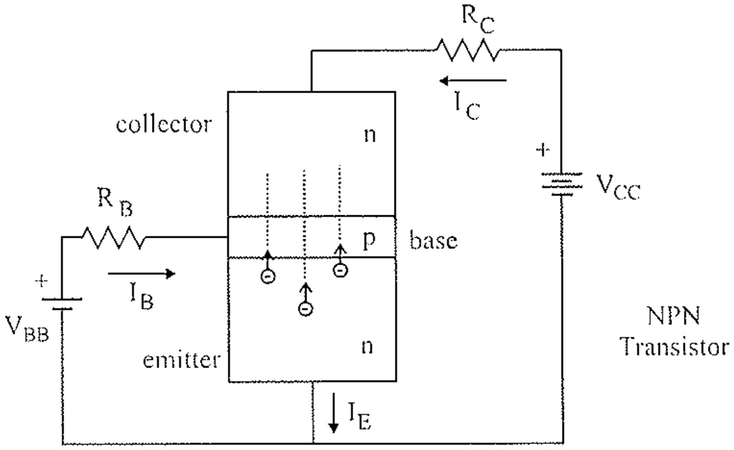


Figure 6.2: The NPN transistor.

This device is called an NPN junction transistor. The names of its elements are descriptive. The region that injects the minority carriers is the *emitter*, and the opposite side of the reverse-biased diode is the *collector*. The common region between them is called the *base*. It is usually thin and shaped in such a way that the injected carriers have the best chance of diffusing across to the reverse-biased junction at the collector. In most transistors the overall efficiency of the process is so high that all but a percent or two of the current I_E (electrons traveling upward in the diagram) ends up contributing to the reverse current of the collector-base diode.

If the injecting junction is placed on the n side of the reverse-biased diode, the device becomes a PNP transistor. The electrode names are the same as before, except that now the n side of the original diode is made thin and becomes the base. Injected minority carriers are now holes, as sketched in figure 6.3.

The circuit symbols for the two kinds of junction transistor are shown in figure 6.4. The base, emitter and collector are labeled B, E, and C respectively. In each case the emitter has an arrow

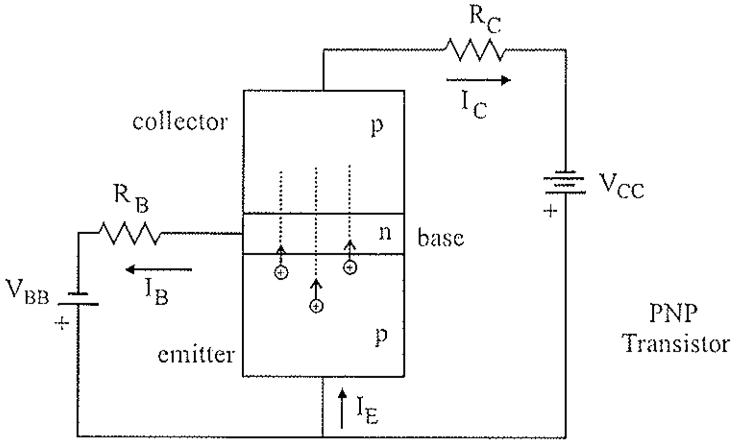


Figure 6.3: The PNP transistor.

denoting the direction of flow of positive current. The arrows for the currents I_e, I_b, and I_c are such that these currents are all positive. Some textbooks will uniformly define all currents as flowing into the terminal. Some currents would then be negative as well as positive. However the currents directions used here result in positive values. You should be careful to note the direction and sign of the current as well as its magnitude. Some specifications of these transistors are listed at the end of this chapter.

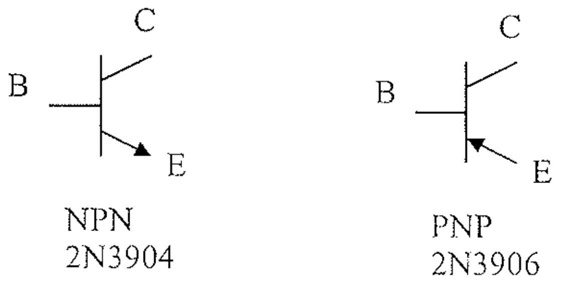


Figure 6.4: Transistor schematic symbols.

Transistors may be drawn in many different orientations. In particular some people prefer to draw PNP transistors with the emitter on the top to indicate that positive current is flowing down. The emitter is identified with an arrow not whether it is on the top or the bottom. You should be prepared to mix both NPN and PNP transistors in the same circuit in either orientation.

To summarize:

1. The collector base junction is reversed biased. Without further intervention no current would flow through the collector.

2. Forward current is forced into the emitter.

3. Almost all of this forward current finds its way to the collector.

4. The net base current is only that fraction of the emitter current that failed to reach the collector.

The fraction of emitter current transmitted to the collector is:

$$\alpha = \frac{|I_C|}{|I_E|} \tag{6.1}$$

The absolute value signs are included to sidestep the polarity conventions which tend to get underfoot (see below). On the basis of the mechanism outlined, α must lie between 0 and 1. Usually it falls in the range 0.97-0.995. For a numerical example, suppose α= 0.99, then an emitter current of 10 mA produces a collector current of 9.9 mA, with the small balance of 0.1 mA going to the base.

In many cases the base current can be regarded as the *control variable* for the transistor. The ratio of the collector current to the base current is:

$$\beta = \frac{|I_C|}{|I_B|} = \frac{\alpha}{1-\alpha} \tag{6.2}$$

This represents a current gain between base and collector which can be much greater than unity. In our example β is (9.9mA)/(0.1mA) = 99.

6.2 Common-Emitter Transistor Characteristics

The most common configuration of the bipolar transistor is the *common-emitter* configuration. The controlling input is the base current, I_b and I_c is the output. The emitter terminal is common to both the input and output circuits. The other two different types of configurations are the common base and the common collector configuration (also called the emitter follower). The NPN and PNP transistors behave in essentially the same manner but all of the polarities are reversed. For simplicity only the NPN will be described in detail in this section. The availability of complementary polarities provides us with an extremely useful degree of freedom in design. Some circuits may use both polarities to some advantage, but for the moment only the NPN is discussed.

The current ratio β depends on the small difference between α and unity. It is very sensitive to small variations in α. Thus β must be regarded as quite an "unreliable" parameter. The only thing that can be said about it is that it is much greater than unity. Operating conditions such as temperature and variations between transistors can vary β over a surprisingly large range. In some transistor types β is specified to fall anywhere from 30 to 300! Therefore you cannot rely upon a precise value of the current gain β.'

The actual value of β is unknown but $\beta >> 1$

With this realistic approach many minor distinctions and some major ones can be ignored. In particular, the difference between the DC current ratio, I_c/I_b, and the *incremental current gain*, $\Delta I_c/\Delta I_b$, can be ignored. Both will be denoted by the same symbol β. Although the symbols α and β are in fairly common use, there are several sets of more formal parameters to describe transistor properties in greater detail. One common set is the *hybrid* (or h) parameters. The name is derived from the fact that the input and output are in different domains (voltage and current). The current ratios α and β appear as forward transfer ratios (subscript f), and the configuration is designated by a second subscript (b or e for common base and common emitter respectively). Finally, incremental ratios use lower-case subscripts, while DC ratios use capitalized ones. Thus

$$h_{fe} = \frac{dI_c}{dI_b} \quad ; \quad h_{FE} = \frac{I_c}{I_B} \quad \text{[use } \beta \text{ for both].} \tag{6.3}$$

The relationship between the current and voltage in an NPN transistor in the common- emitter configuration is shown in figure 6.5. The vertical axis is the collector current and the horizontal axis is the voltage between the collector and emitter. Only the curve for the NPN transistor is shown. The PNP transistor has the same shape but all of the voltages and currents reverse sign.

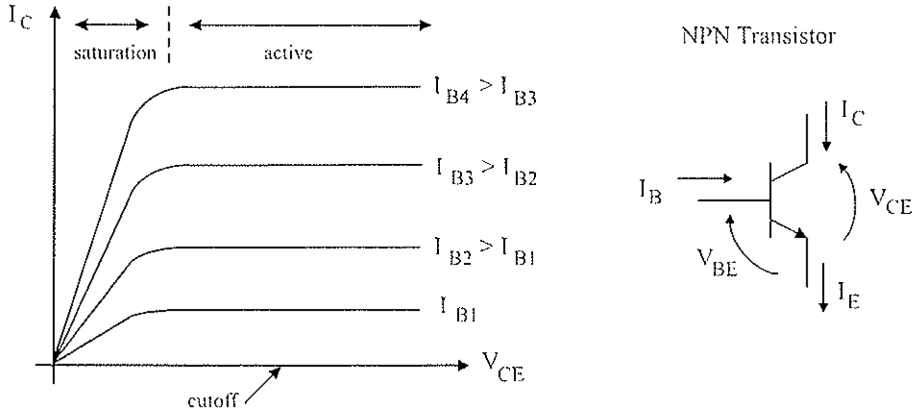


Figure 6.5: Current and voltage in an NPN transistor.

There are three different regions of operation that are defined in the characteristic curve of the transistor. In the *active region* the collector current is constant over a large range of collector-emitter voltages when the base current is held constant. When the collector-emitter voltage V_{CE} falls to a small value (typically a little less than a diode drop of 0.6 V) the transistor is said to be in the *saturation region*. The transistor enters saturation when V_c falls below V_b. In other words, when the collector-base junction becomes *forward biased*. The sharp downward bend in the collector-current curve can be attributed to forward current being sent from the collector back into the base, and subtracting from the normal transistor current. The third region is along the horizontal axis and is signified by a zero collector current. A fourth region that is not displayed in the graph above is the breakdown region. If the collector voltage is raised even higher the transistor will eventually breakdown and the collector current will start rising rapidly. This can damage the transistor and should be avoided. In summary these four regions (for the NPN transistor) are:

Cutoff: $I_c = I_B = 0$, usually caused by V_{BE} <0.6 V, the transistor is said to be "OFF".

Active region: V_{BE} = 0.6 V (a diode drop) and $I_C = \beta I_B$, the transistor is a current controlled current source.

Saturation: $V_{CE} = V_{CE-SAT}$ = 0.1 to 0.4 V, I_C is as large as permitted by the external circuit. The transistor is said to be "ON".

Breakdown: The collector voltage is too high and a large collector current flows. This may damage the transistor.

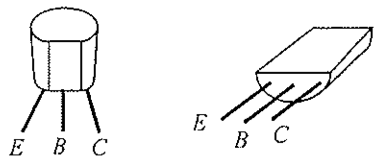


Figure 6.6: Physical appearance of the transistor (both 2N3904 and 2N3906).

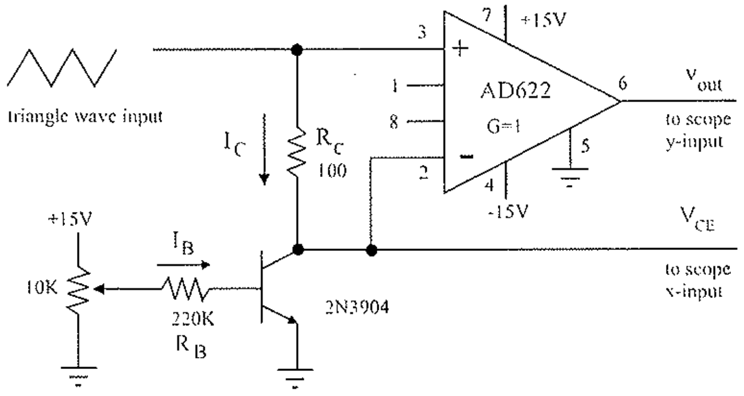


Figure 6.7: Circuit to display the I-V curve of a transistor on the oscilloscope.

Exp. 6.1 Setup the circuit shown in figure 6.7 to display the I-V characteristics (I_C versus V_{CE}) of a 2N3904 NPN transistor on an XY scope. The AD622 instrumentation amplifier (see figure 5.10) with a differential gain of unity (no resistor between pins 1 and 8) is used to convert the current flowing through the collector resistor R_C into a voltage to display on the Y axis of the scope. The collector-emitter voltage V_{CE} will be the X axis. Set the function generator to deliver a triangle wave of about 100 Hz to 1 KHz with an amplitude of -2 V to +10 V.

The base current in the transistor I_B can be varied with 10K potentiometer. Observe how the I-V curve changes on the scope as you vary I_B. Choose three different settings of base current to yield collector currents about equally spaced over the available range. At each of these base currents, record the collector current at a collector emitter voltage of 7 volts (on the scope), and the corresponding base current (use a handheld meter to measure the voltage across R_B).

At each of these three base currents calculate the DC current gain $\beta = h_{FE} = I_C/I_B$ of the transistor. From adjacent pairs of base currents calculate the AC current gain $\beta_{ac} = \Delta I_c/\Delta I_b$. What is α for your transistor? end

Exp. 6.2 OPTIONAL: Use the Tektronix 571 Curve Tracer located in the center room to record the common-emitter I-V characteristics of a 2N3904 NPN bipolar transistor. The curve tracer contains a microcomputer which can control various voltages and currents that are applied to the device being tested. It has the ability to perform and store many measurements and to plot the resulting curves on a printer.

The actual semiconductor device under test is inserted into the appropriate socket on the upper right hand portion of the control panel. Note that the curve tracer will not function with voltages above 30 V if the protection cover is not closed. The device type and operating parameters are set up by pushing the button labeled 'MENU'. The screen then will display a variety of options that may be addressed using the cursor keys.

In bipolar transistor mode, the curve tracer sweeps the collector voltage V_{CE} for a series of constant base currents I_b and measures the corresponding collector current I_c.

1. Common-emitter characteristics. First bring up the menu of options by pressing MENU, and make sure the following options are in effect:

Function = Acquisition

Type = NPN

Vce max = 50 V

Ic max = 20 mA

Ib/step = 10 μA

Steps = 10

Rload = 0.25 Ω

Pmax = 2 W

(Remember: To interrupt a test sequence manually, press the STOP button.)

Insert a 2N3904 transistor into the transistor terminals (see figure 6.6 for the pin positions), and close the protection cover. Press START. After a second or two, the characteristic curves should begin to appear on the screen. When the test is complete, press COPY to print the results. Can you identify the different regions of transistor operation (active, saturation, cutoff, breakdown)?

2. Next, examine the current gain of the transistor. You can use your characteristic curves to estimate the dc and ac current gains (β_{dc} and β_{ac}). First, press CURSOR to activate the two cursors. Use the $\Uparrow, \Downarrow, \Rightarrow$, and $\Leftarrow$ buttons to move the cursors to the middle of two adjacent curves (near 8 volts between collector and emittter). (To toggle between cursors, press the CURSOR button.) The values of I_c, I_b, and V_{CE} at the selected points are displayed to the left of the graph along with the DC current gain (hFE = $\beta_{dc} = I_c/I_b$) at the location of the active (blinking) cursor. Record the value of hFE for the other cursor as well. Are these values reasonable? Now print the screen by pressing COPY.

The AC current gain is defined as $\beta_{ac} = \Delta I_c/\Delta I_b$, where ΔI_c and ΔI_b represent corresponding small fluctuations in collector and base currents. Calculate β_{ac}, using current values kindly measured by the curve tracer (near Vce=8V and Ic=2mAmp).

What is α for your transistor?

3. Power dissipation. Repeat the above test, but with Pmax set to 0.1 W. This limits the power dissipation in the transistor accordingly, and the characteristic curves end when they reach this limit. Press COPY to print your results. Is the shape of the limiting envelope what you

expect? Does it actually correspond to 0.1 W? end

6.3 Circuit Model for the Bipolar Transistor

The characteristic curves of the transistor define the behavior of the transistor. However the curves are difficult to apply directly to analyze a circuit. In the active region the transistor produces a constant collector current independent of collector voltage. This means that the transistor behaves as a constant current limit device. When sufficient voltage is applied this behavior is identical to that of a current source so it is usually drawn as a current source although the current actually comes from the external circuit and not the transistor. The base current determines the value of the collector current. The transistor is a *current controlled current source*, and can be approximated in a circuit with the model in figure 6.8.

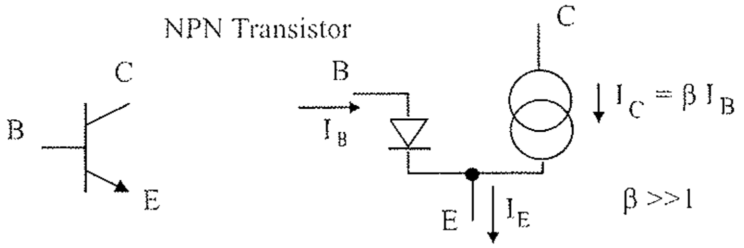


Figure 6.8: NPN transistor circuit model.

The base emitter junction is the controlling junction. It determines what the collector current will be. The base-emitter junction acts very much like a diode. When forward biased, V_{bc} is usually about 0.6 V (can be 0.5-1.0V in practice). However the current flowing through each lead of the diode can be dramatically different. A small base current I_B controls the much larger collector current source $I_C = \beta I_B$. The base current and the collector current combine in the emitter lead to produce the emitter current:

$$I_E = I_B + I_C = (1 + \beta)I_B \tag{6.4}$$

Although this model was justified from the active region, it is also useful in the saturation and cutoff region if these regions are thought of as the extreme limits of the active region. Also note that the PNP transistor has a similar model except that the direction of the diode and currents is reversed.

It is evident from this model that currents are the relevant signals when dealing with transistors. When working with op-amps, the important signals usually take the form of voltages that can be directly viewed on the oscilloscope. However with transistors the important signals (currents) cannot be viewed directly. The current must be passed through a resistor to produce a voltage to see it on the scope. This indirect nature is one thing that makes transistors more difficult to understand.

6.4 The Load Line

When the collector is connected via a *resistive* load R_C to the supply rail (V_{CC}), V_C and Ic are constrained by Ohm's law to fall on the straight line given by:

$$V_C = V_{CC} - I_C R_C \tag{6.5}$$

This line is called the load line. It is plotted in figure 6.9 along with the transistor characteristics. Two points on this line are easily spotted on the $V_C - I_C$ characteristics, at V_{CC} on the horizontal axis and at V_{CC}/R_C on the vertical. This permits us to easily draw the load line by connecting these two points. The actual operating point of the transistor is at the intersection of this load line with the collector curve appropriate for the base current supplied to the transistor. (This base current is determined, directly or indirectly, by the bias circuit for the transistor.)

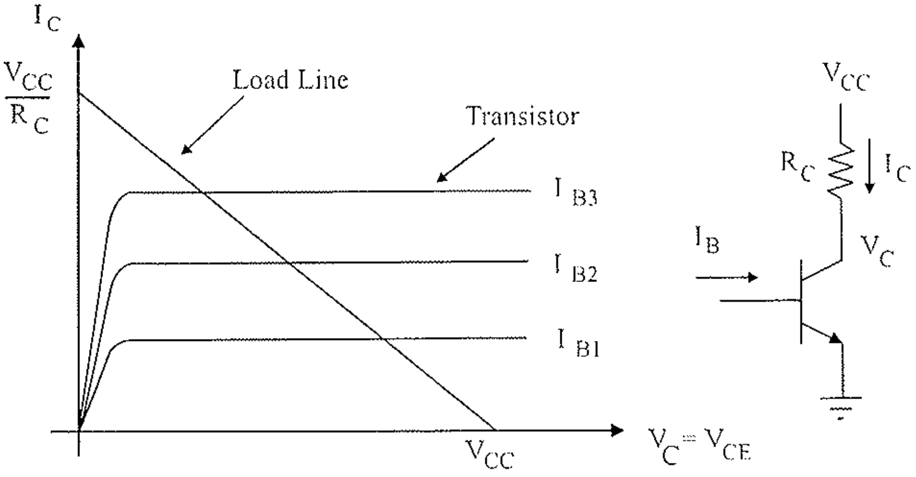


Figure 6.9: Transistor load line.

6.5 The Saturating Inverter

A transistor connected as a saturating inverter is shown in figure 6.10. The circuit schematic is shown on the left and the circuit model is shown on the right.

If the transistor is driven with a relatively large input signal then it is either cut-off or saturated, and it acts as an inverting *switch*. When v_{in} is large and positive, the transistor is 'ON' and v_c is low (near ground). When v_{in} is zero or negative the transistor is 'OFF" and v_c becomes V_{CC}. This transistor circuit has low voltage gain but can have a considerable current gain and hence power gain. It is often used as an output stage to drive such loads as lamps, relay coils, or motors (substituted for R_C in the circuit shown above). If the switching is clean, the transistor power dissipation is very low in either state. A relatively small transistor can control a large load power.

The graphs in figure 6.11 show the currents and voltages around the transistor as a function of the input voltage v_{in}. As the input voltage increases from negative values to large positive value the

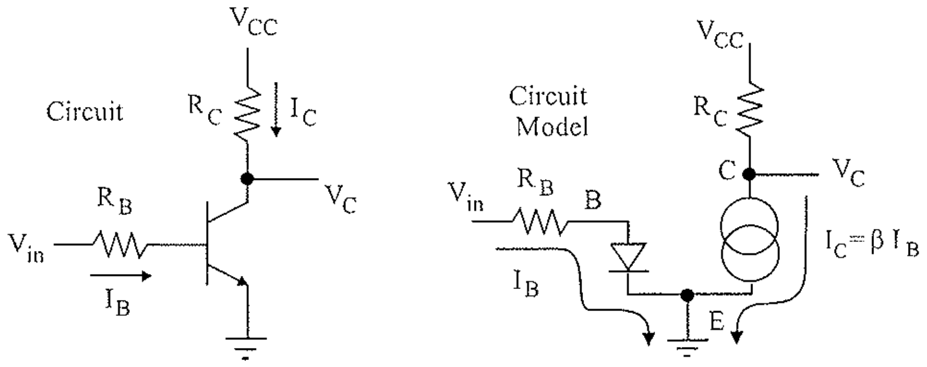


Figure 6.10: Saturating inverter.

transistor progresses from cut-off, through the active region and into the saturation region. If v_{in} has a large amplitude then the transistor crosses the active region relatively fast and stays only in the cut-off or saturation regions.

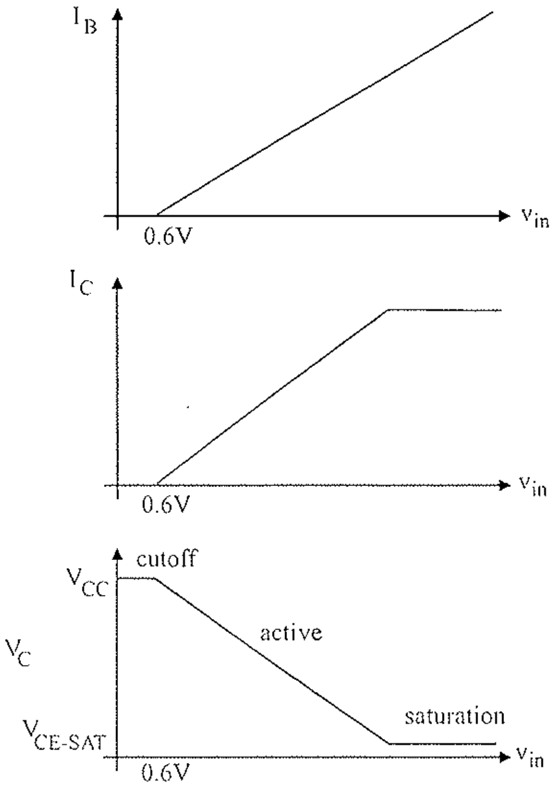


Figure 6.11: Current and voltage in the saturated inverter.

6.5.1 Cutoff

When the input voltage is negative the base is actually *reverse* biased, and the transistor is clearly in cut-off (for the npn polarity shown). There is no base current and hence no collector current. Without a collector current the voltage at the collector becomes V_{CC}. Only a fraction of a volt of reverse bias is required to reduce the emitter current to its lowest possible value. In most transistors the reverse-voltage rating for V_{be} is relatively small (often only a few volts), so excessive reverse bias should be avoided.

The transistor is also cut off when $v_{in}= 0$, but now the value of the base resistor R_b begins to become important. The residual collector current, I_{co}, should flow through R_b, not through to the emitter. The voltage drop in R_b is in the direction to apply some forward bias to the base, which should be kept small. Hence zero-bias cutoff should go with relatively low values of R_b. Preferably R_b should be 10 kΩ or less for small transistors, and lower still for power transistors or for high-temperature operation.

If v_{in} is slightly positive the transistor edges toward the forward conduction region of the base-emitter junction. However, a small forward bias (no more than about 0.1 V) may be acceptable if R_b is sufficiently small (perhaps 1 kΩ or less). Low R_b and reverse bias for cutoff are also helpful in securing a fast transition from the conducting to the cutoff state of the transistor. There is some stored charge in the base region of a conducting transistor, particularly when the transistor is saturated (collector forward- biased, also injecting carriers). This stored charge must be removed to turn the transistor off. It's much like discharging a fairly sizeable effective base capacitance.

6.5.2 Active

As v_{in} moves quickly from negative values (or small positive values) to large positive values the transistor briefly passes through the active region. In the active region the collector current is linearly proportional to the input voltage.

$$I_C = \beta I_B \quad \text{and} \quad I_B = \frac{v_{in} - 0.6V}{R_B} \tag{6.6}$$

The collector resistor R_C turns the collector current back into a voltage as:

$$V_C = V_{CC} - I_C R_C \tag{6.7}$$

6.5.3 Saturation

As v_{in} and hence I_B continue to increase the collector current reaches a maximum and the transistor is said to be in saturation. The minimum voltage between the collector and emitter is labeled V_{CE-SAT} and has a typical value of about 0.1 to 0.4 volts. Therefore the maximum collector current is $(V_{CC} - V_{CE-SAT})/R_C$. To guarantee saturation, the transistor has to deliver this much current even if the particular transistor happens to be as feeble as they come. Thus, to ensure that a particular transistor is guaranteed to be in saturation, decide on the lowest value of β that the transistor can have and apply some safety margin to account for component tolerances. Call this β_{min}. Then design your circuit to force a base current through the transistor of magnitude:

$$I_B(SAT) = \frac{V_{CC} - V_{CE-SAT}}{R_C \beta_{min}} \tag{6.8}$$

Then the transistor is forced to operate such that the ratio of the collector current to the base current, $I_c(sat)/I_b(sat)$, is less than the active β for that transistor. The effective current gain in saturation is defined as $\beta_{sat} = I_c(sat)/I_b(sat)$. An equivalent way of describing transistor operation in saturation is to say that β_{sat} falls off as the transistor enters saturation.

For a circuit to operate in saturation, prudent design then amounts to choosing β_{sat} less than the lowest active β to be expected. In many cases, when the requirements are loose enough to permit us the luxury of very conservative design, we might take $\beta_{sat} = 10$, which is much smaller than we expect the active β to be.

$I_b(sat)$ is found from:

$$I_B = \frac{v_{in} - 0.6V}{R_B} \tag{6.9}$$

Exp. 6.3 Build the saturating inverter shown in figure 6.12. Use square waves with various upper and lower levels to explore its behavior. The diode from the transistor's base to ground protects against excessive reverse voltage, and the load line is unconditionally safe as regards transistor power dissipation. You can vary v_{in} at will without damaging the transistor, even making it unsymmetrical by use of the function generator's OFFSET control. Record V_{in}, V_b and V_c. Find the conditions on the input that produce saturation of the transistor. With a large amplitude input (+/- 10V), measure the voltage to which V_c falls in saturation (i.e. measure V_{CE-SAT}). Also record the rise and fall times of V_c and its time relationship to v_{in}, and the behavior of v_b.

OPTIONAL-1: The transistor has a small switching delay (or propagation delay from v_{in} to V_c). Increase the input frequency (keeping the input amplitude large) until you can see this switching delay on the oscilloscope and measure it (time between the 50% amplitude point on each signal, which may be different on the low-to-high and high- to-low transitions). Add a small *speedup capacitor*, C (50pF or more) across R_B to see what effect it has on the response times (switching delay) of the inverter.

OPTIONAL-2: You should be able to reproduce figure 6.11 on the scope in xy mode by applying a triangle wave to the input (on the scope x input) and displaying the collector voltage on the y-input of the scope. Record the voltage for the various transition points in the curve. end

The output impedance of the inverter is just R_C when the transistor is cut off, but it is much lower when the transistor is saturated. (The incremental impedance of the saturated collector is only a few times larger than that of a forward-biased diode carrying the same current.) The change of output impedance between the two states can produce a very different behavior in the rising edge of V_C as compared to the falling edge.

Exp. 6.4 Using the circuit from the previous experiment (figure 6.12), connect a 0.01 μF capacitor from the inverter's output (V_C) to ground and observe the rise and fall times of V_C with this load. Interpret your observation. Which circuit element pulls the output high and which circuit element pulls the output low? end

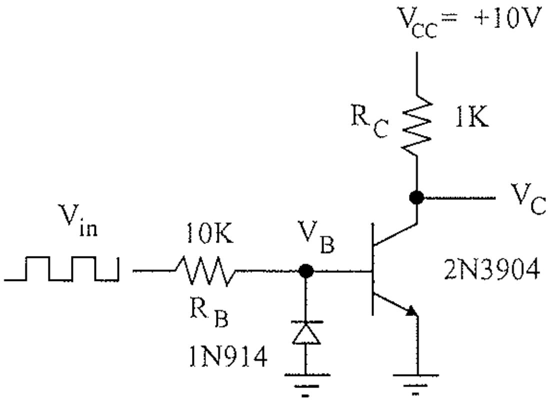


Figure 6.12: Experiment for the saturated inverter.

When a saturating inverter (or any other type of solid-state switch) drives an *inductive* load, such as a relay or motor winding, the load current cannot stop abruptly when the switch is turned off. Unless a specific path is provided for this inductive surge current, a large voltage "spike" is developed which will probably kill the transistor. The simplest remedy is to connect a *diode* as shown in figure 6.13. This diode is cut off while the transistor is on, but it accepts the load current upon turnoff until the inductive energy has been dissipated.

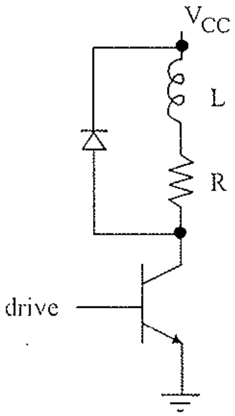


Figure 6.13: Driving an inductive load.

Because of the conservative design criteria for a saturated inverter, the current gain obtained from a single stage is quite modest. Two or more inverters can be cascaded directly as shown in figure 6.14. The output of the first stage (Q1) is feed into the input of the second stage (Q2).

Exp. 6.5 Construct the two-stage circuit shown in figure 6.14, using PNP transistors. v_{in} should be the maximum amplitude square wave from the function generator (-10V to +10V). Select a value of R_{C1} that will give values of β_{SAT} about the same for both transistor stages. For each of the two possible values for the input, think about which transistor is on and which transistor is off. Then calculate I_C and I_B for the transistor that is on. This gives a value for

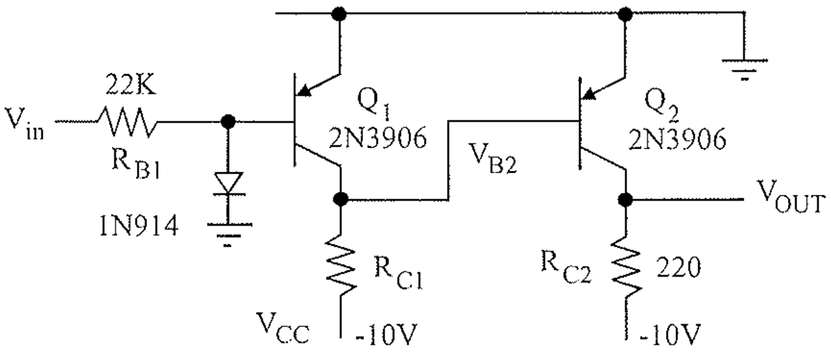


Figure 6.14: Cascaded saturating inverters.

β_{SAT} for that transistor. Equate the values of β_{SAT} for each transistor to find R_{C1}. Make a sketch showing v_{in}, v_{b2} , and v_{out}. If R_{C2} is interpreted as the load, what is the current gain of this circuit? $\boxed{\text{end}}$

6.6 DC bias and the Q-point

The saturated switch has only two output voltages (high and low) when driven with a large input signal. A linear amplifier should have a continuous range of output values (over some range of signal amplitudes). The output of the amplifier should be linearly proportional to the input signal. For example if the input is a sine wave of a given amplitude then the output should also be a sine wave with the same frequency and phase but a larger amplitude. A linear amplifier has the property that it preserves the shape of an arbitrary signal. Audio amplifiers designed for high fidelity music are linear amplifiers. As used above the saturated inverter always outputs a square wave. It works fine as a switch and cannot be used as a linear amplifier.

One way to make a linear transistor amplifier is to apply a large DC bias to the transistor to put it in its active region and then superimpose only a small AC signal on top of the DC bias. Consider the graph shown in figure 6.15 of the collector voltage versus the input voltage v_{in} for the saturated inverter. If the transistor were somehow held at the Q-point (or quiescent point) and a small AC signal Δv_{in} were applied, then the AC component of the output would follow the input with a net voltage gain that is just the slope of the curve in the active region (i.e. this would be a linear amplifier for the AC component and not the DC component of the signal).

It is useful to divide the voltages and currents in the transistor into two different types. The total values are the sum of the Q-point values (DC bias) plus a small AC signal:

$$V_{Total} = V_Q + \Delta V = V_Q + v \tag{6.10}$$

$$I_{Total} = I_Q + \Delta I = I_Q + i \tag{6.11}$$

Large DC values will be denoted by upper case letters and small AC signal will be denoted by lower case letter. If the small AC components are very small ($|\Delta V| << |V_Q|$ and $|\Delta I| << |I_Q|$) then the analysis of the circuit can be approximately treated as two separate and independent problems. First

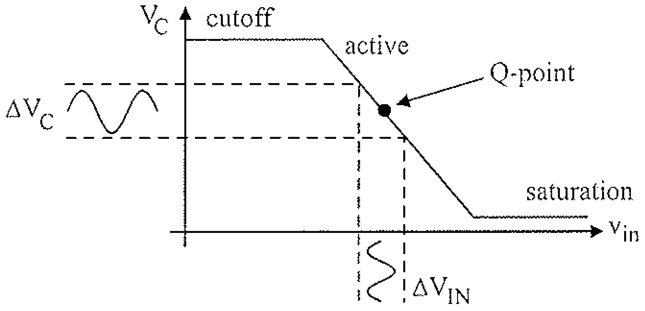


Figure 6.15: The Q point.

solve for the Q-point DC bias voltages and currents without any AC input signal, and then solve for the small AC component with suitable approximations. This will prove to greatly simplify the total analysis.

Which point in the active region to choose depends on the output swing needed from v_c. If signal excursions away from the quiescent point need be in one direction only, there are great advantages in keeping the transistor either cut off or in saturation. If bipolar signal swings are needed, the quiescent point must be near the center of the active region. Specific biasing conditions will be discussed later in the context of specific examples.

6.7 Voltage Control and Transconductance

Approximating the base-emitter junction as just a simple voltage drop of 0.6 volts is not very accurate. This diode drop is only accurate to about 0.1 volts, which is usually accurate enough to calculate the large DC bias voltage and currents at the Q-point. However small AC signals may also have this same magnitude. A better analysis of the small signal properties of a transistor circuit should take into account the non-linear variation of the base-emitter junction. The voltage and current in emitter circuit, I_E and the base-emitter voltage V_{BE} are related by a modified diode equation as:

$$I_E = I_0 \left[\exp(V_{BE}/V_T^*) - 1\right] \tag{6.12}$$

As in chapter 5 on diodes, I_0 is the reverse saturation current and $V_T^* = 40$ mV is a constant. Note that the current I_E is only in the emitter circuit and not the base circuit (the base current is a factor of β smaller), but that the voltage is between the base and emitter circuits. The effects of this equation on small AC signals may be approximated using the slope (or derivative) of the above expression near the Q-point (see figure 6.16).

The ratio of the output current to the input voltage is called the transconductance g_m which has units of 1/Ohms. It is also convenient to define a quantity called r_e that is the inverse of the transconductance and has units of Ohms.

$$g_m = \frac{1}{r_e} = \frac{dI_e}{dV_{BE}} \tag{6.13}$$

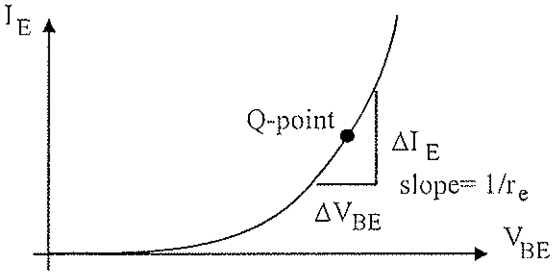


Figure 6.16: Dynamic base-emitter resistance.

The quantity r_e is the incremental resistance of the base-emitter junction. It is written as a lower case letter because it applies only to small AC signals. Taking the indicated derivatives:

$$\frac{dI_e}{dV_{BE}} = \frac{1}{V_T^*} I_0 \exp(V_{BE}/V_T^*) \approx \frac{I_E}{V_T^*} \tag{6.14}$$

which leaves:

$$r_e = \frac{V_T^*}{I_E} \tag{6.15}$$

This leads to a transistor model (figure 6.17) which can be helpful in summarizing voltage-controlled performance. It has an ideal diode at base-emitter junction with zero incremental resistance. Then, to account for the transconductance, include a resistor r_e in the emitter lead. Alternatively, a base resistance $r_b = \beta r_e$ can be included in the base lead. Note that r_e varies dramatically with quiescent current (I_E) and that you shouldn't include both r_e and r_b in the circuit explicitly. The presence of one automatically produces the corresponding impedance in the other place. Because the base current is $1/\beta$ times the emitter current the base resistance, r_b, is β times the emitter resistance r_e. Note that r_e and r_b are inside the transistor and cannot be removed. Only the points labeled EBC are accessible.

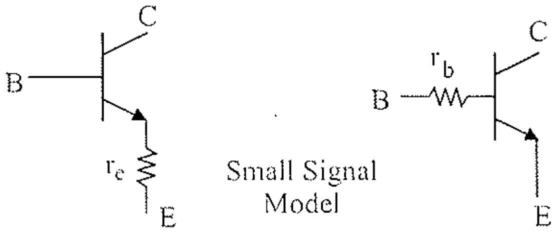


Figure 6.17: Small signal circuit model for the BJT.

Now it is possible to say when the transistor should be considered to be voltage controlled, and when current controlled. If the signal source has an impedance much lower than r_b, it exerts voltage control. If its inpedance is much larger than r_b, it exerts current control.

Given a small transistor with quiescent collector current of 10mA and β=100 leaves $r_e \approx 4\Omega$ or, $r_b \approx 400\Omega$. Under these conditions, the function generator (output impedance 50Ω) would voltage-control the transistor. The transconductance would be $1/(4\Omega) = 0.25$ S (Siemens). (Until recently the unit for conductance was called the mho which is Ohm spelled backwards.)

The transfer of impedances "through" the transistor from base to emitter applies not only to r_b and r_e, but also to any external impedances connected to the base or emitter leads.

6.8 The Emitter Follower

This configuration (figure 6.18) serves to drive a low-impedance load from a high-impedance source. It has a voltage gain of unity but a current gain of β (or equivalently a high input impedance and a low output impedance). The output capacitor C_{out} should be large enough to appear as a short circuit for AC signals. It serves to block the DC level but pass the AC signals. The resistor R_L represents a load. This is a linear amplifier for small AC signal.

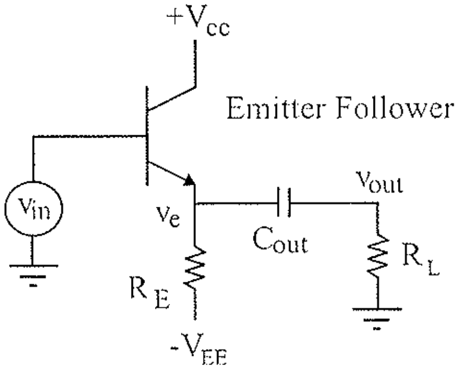


Figure 6.18: Emitter follower.

The input voltage v_{in} should be a small AC signal centered about ground and $(-V_{EE})$ is a large negative voltage. This means that the base emitter junction is always forward biased and the transistor is in its active region. The voltage at the emitter v_e follows the voltage at the base but with a DC offset of approximately 0.6 V. The output capacitor C_{out} looks like an open circuit for the DC signals but like a short circuit for AC signals so that $\Delta v_{out} = \Delta v_{in}$ when $R_L = \infty$. Note that this refers to incremental signal voltages, not the DC levels. As R_L becomes smaller it will load the output and the amplifier will eventually stop working. However for high to intermediate values of R_L the output voltage will follow the input voltage but with more available current (current gain approximately β).

First analyze the input impedance of the emitter follower for small AC signals as shown in figure 6.19. The small dynamic resistance r_e is also shown in the circuit. Remember that it should only be included for small AC signals. Consider a small change in voltage at the base ΔV_B. What is the corresponding change in base current ΔI_B? The voltage at the emitter V_E follows the voltage at the base V_B but is offset by a diode drop of about 0.6 V, so:

$$\Delta V_E = v_E = \Delta V_B = v_B \tag{6.16}$$

where again lower case letter will denote small AC signals. The current in the emitter circuit is the sum of the base current and the collector current and the collector current is equal to the base current multiplied by β:

$$\Delta I_E = i_E = \Delta(I_B + I_C) = i_B + i_C = (1+\beta)i_B \tag{6.17}$$

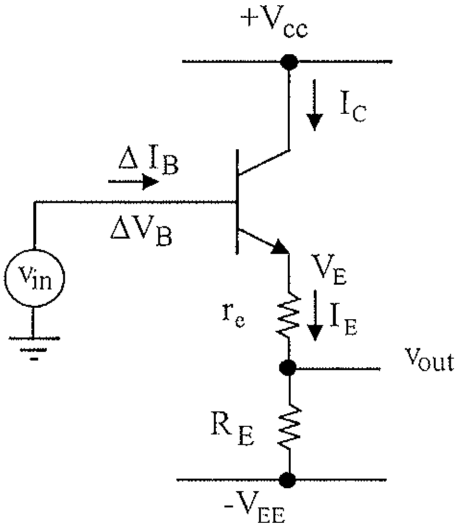


Figure 6.19: Emitter follower.

This current is related to the voltage at the emitter as:

$$\Delta V_E = v_e = i_E(r_e + R_E) = (1+\beta)(r_e + R_E)i_B \tag{6.18}$$

Combining these expressions yields the input impedance for small AC signal (or the dynamic input impedance):

$$Z_{in} = \frac{\Delta V_B}{\Delta I_B} = \frac{v_B}{i_B} = \frac{v_E}{i_B} = \frac{(1+\beta)(r_e + R_E)i_B}{i_B} = (1+\beta)(r_e + R_E) \tag{6.19}$$

The input impedance is just $(1+\beta)$ times the impedance in the emitter circuit. The value of β is large compared to unity so sometimes $(1+\beta)$ will be approximated as just β. In a similar derivation as above it can be shown that the impedance in the base circuit is transferred to the emitter circuit with a factor of $1/(1+\beta)$.

If the input voltage v_{in} is a perfect voltage source (with an output impedance of zero) then the voltage at the emitter ΔV_E also looks like a perfect voltage source so the output impedance of the emitter follower is r_e. If, instead, the input voltage source has an impedance of R_s, the output impedance becomes:

$$Z_{out} = r_e + \frac{R_S}{\beta} \tag{6.20}$$

The capacitor C_{out} (in above circuit) is sometimes referred to as a coupling capacitor. It is supposed to isolate R_L from the DC bias circuit of the transistor while allowing free passage of the small AC signal. This type of arrangement is present in many amplifier circuits because of a rather fundamental conflict between the aims of a good DC bias circuit and those of the AC signal amplification path. The coupling capacitor separates the DC bias circuit from the AC signal. It allows the bias circuit to establish a fixed DC level without being changed by an external DC resistance of the load. This allows us to treat the DC bias circuit and the AC signal paths as two separate problems (if the AC signal is only a small perturbation) that are easier to solve than one large problem. The actual voltage is just the superposition of the AC and DC signals. This means that the impedance of the capacitor must be negligible with respect to the other impedances in the circuit at the lowest signal frequency of interest. What "negligible" means depends, of course, on the context. In the circuit at the beginning of this section, the capacitor should have a reactance small compared to $(r_e + R_L)$.

It is not always convenient to have two power supplies as above. The emitter follower circuit may also be configured with a single supply by adding another AC coupling capacitor at the input. An example of an emitter follower working from a single power supply not using a DC path through the source of v_{in}, is shown in figure 6.20.

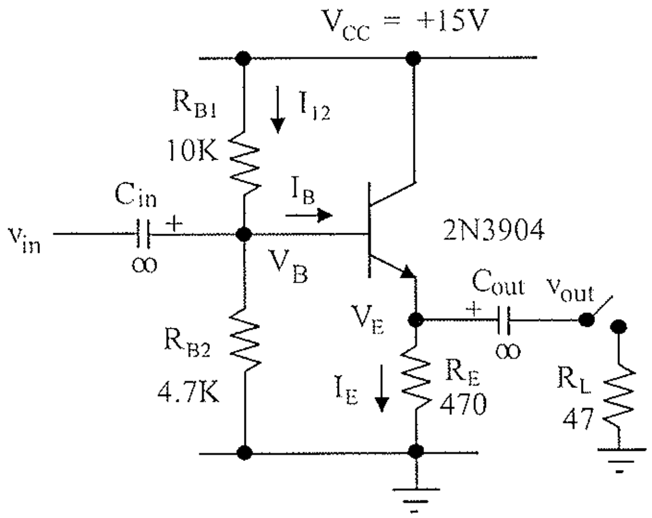


Figure 6.20: Single supply emitter follower experiment.

The first task is to calculate the Q-point (or DC bias) voltages and currents. The voltage at the base is related to the currents flowing in R_{B1}, R_{B2} and the base current I_B. However I_B depends on V_B and β which is not known accurately. This is a difficult calculation and involves the unknown parameter β. This calculation can be simplified by assuming that the base current is very small (i.e. β is large) so that the current $I_{12} = V_{CC}/(R_{B1} + R_{B2})$ is much larger than I_B. This means that R_{B1} and R_{B2} should be small. However you do not want to make them too small because that would seriously reduce the input impedance. You have to try to find something in between where the calculation is easy but the input impedance is not too low. If I_B is negligible then:

$$V_B \approx \frac{R_{B2}}{R_{B1} + R_{B2}} V_{CC} \tag{6.21}$$

and then the DC component of the emitter voltage and current are:

$$V_E = V_B - 0.6V \tag{6.22}$$

$$I_E = \frac{V_E}{R_E} \approx I_C \tag{6.23}$$

Now, using a typical value for β (100 is traditional for the 2N3904), you can find the actual I_B and correct your estimate for V_B if necessary. Given a new value of V_B you can then calculate a new value of

I_B etc. By this iterative approach you can get close enough to the actual conditions very quickly. Note, however, that if I_B perturbs the "natural" value of V_B significantly, you have opened a door through which β can influence the quiescent conditions of the circuit which would generally be considered bad news.

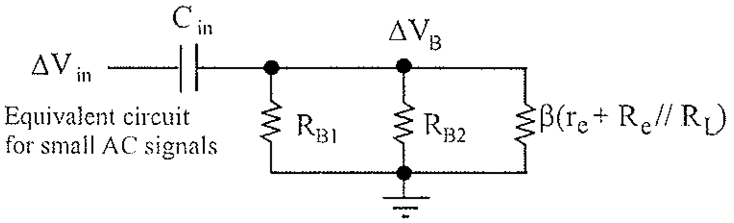


Figure 6.21: Equivalent input impedance for the emitter follower.

The next task is to determine a value for the coupling capacitors C_{in} and C_{out}. The input capacitor should pass the AC signal without attenuation but block the DC Q-point levels. An equivalent input circuit for small AC signals at is shown in figure 6.21. Note that the DC power supply (V_{CC}) looks like a ground for small AC signals because there cannot be any AC appearing on top of the supply voltage (which is held at a constant DC level). The input signal sees both R_{B1} and R_{B2} to ground along with β times the impedance in the emitter circuit. The goal is to make the AC signal that appears at the base (ΔV_B) be the same as that at the input (ΔV_{in}). Therefore for sufficiently large frequencies:

$$|Z_{Cin}| << R_{B1}//R_{B2}//\beta(r_e + R_E//R_L) \tag{6.24}$$

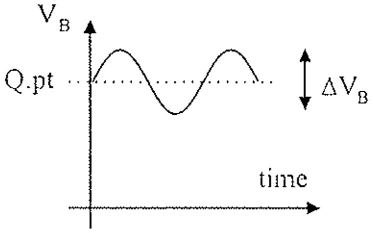


Figure 6.22: Superposition of the Q-pt bias voltage and the input signal.

You should calculate a specific value of C_{in} in the next experiment. Usually both coupling capacitors will be large. The end result for the total voltage at the base is shown in figure 6.22. The total voltage is the superposition of the DC Q-pt. value and the small AC voltage ΔV_B (which should be the same as the AC input voltage ΔV_{in}).

A similar expression exists for the other coupling capacitor C_{out}. In the small signal model shown in figure 6.23, the AC signal at the emitter is the same as that at the base, which in turn is the same as the input signal. The apparent source impedance that is driving the emitter signal is approximately $1/\beta$ times the impedance in the base (including the function generator output impedance of 50 Ohms). Therefore, the AC voltage at the emitter is essentially an ideal voltage source. To be precise you should replace ΔV_E, r_e and R_E with their Thevenin equivalent circuit and then calculate v_{out} including the impedance of the output capacitor C_{out}. C_{out} needs to be large enough so that the value of v_{out} is unaffected by this capacitor and as close to v_{in} as it can be. With the values in use here a rough approximation is to require that $|Z_{Cout}| << r_e$.

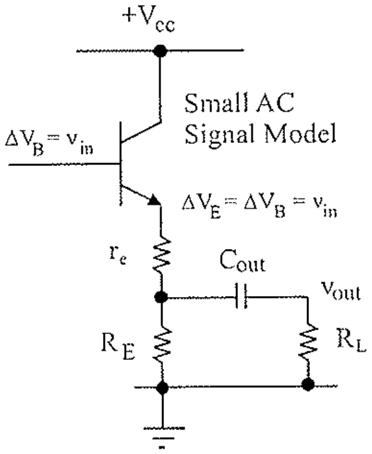


Figure 6.23: Small signal AC circuit model.

Exp. 6.6 Finish the design of the emitter-follower circuit shown in figure 6.20 by calculating values for C_{in} and C_{out} that meet the design criteria discussed above when R_L is in place. First build the circuit and measure the DC Q-pt. values with no input signal (and compare to what you calculate). Next explore the performance of this circuit using a 1 kHz sine-wave input signal. Test it with and without a load of $R_L = 47\Omega$.

With the input voltage small enough such that the output is undistorted, measure the AC voltage gain with and without R_L and compare to the theoretically expected value. Increase the amplitude of v_{in} gradually, again with and without R_L, and observe the ways in which the circuit departs from linearity. Try to understand the levels at which the circuit cuts off with R_L in place. (See following discussion).

[OPTIONAL] From the small signal AC gain (with undistorted output) you can obtain a measured value of the internal dynamic resistance r_e and compare to the theoretical value. **end**

The unsymmetrical limiting effects you have just seen are the consequence of the asymmetry of the base-emitter diode. When the output is being pulled to a higher voltage by the input, the transistor is on (base-emitter diode forward biased) and provides a significant current gain, however when the output is being pulled lower the base-emitter junction just gets reverse biased and the transistor is off completely. Thus for an NPN emitter follower, the negative load current can never be larger than the original quiescent transistor current. The positive load current is not subject to the same restriction, since the transistor can be driven very hard. With C_{OUT} large and R_L in place this effect is even more pronounced. In the limit as R_L goes to zero, a large C_{OUT} tends to hold the emitter at its Q. point value (just like the filter capacitor in a half wave rectifier), so even a small negative swing at the input will cause the base-emitter diode to be reverse biased. This asymmetry often determines whether an NPN or PNP transistor should be used. An NPN can pull the output high but not low and a PNP is just the opposite.

6.9 Emitter Bias

To put the transistor at a specific point in its active region requires fixing the collector current I_c to the appropriate value. Your first thought might be to supply the corresponding base current I_b to the transistor. Hold everything! This idea must be rejected because β is an unreliable parameter. Do not be misled by curves drawn for "typical" operating conditions into believing those curves really determine anything. The same curves for another transistor (of the same type) might look dramatically different. The only terminal that I_c can be effectively controlled from without the intervention of β is the emitter. *To determine I_c fix the current in the emitter lead.* This leaves only α, a highly predictable parameter, playing a part in the system.

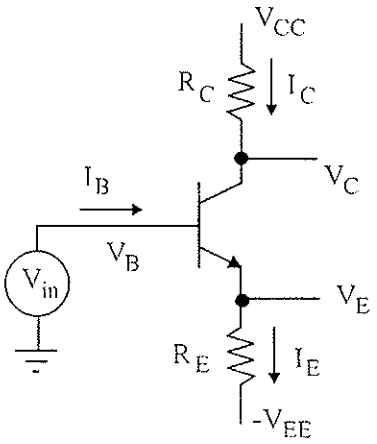


Figure 6.24: Emitter bias.

In the circuit shown in figure 6.24 the source v_{in} must have a DC path through it, fixing the quiescent value of V_B and permitting I_B to flow. For the moment, assume v_{in} is a small AC signal centered about ground, so that $V_B \approx 0$. Different DC levels could be chosen for V_B, which might avoid the need for two power supplies (V_{CC} and V_{EE}). The emitter current is formed from R_E and V_{EE} supplying a bias current of:

$$I_E = \frac{V_E - (-V_{EE})}{R_E} = \frac{-V_{BE} + V_{EE}}{R_E} \approx I_C \tag{6.25}$$

where V_{BE} (approximately 0.6 V) is predictable to within about 0.1 or 0.2 volts. Thus the magnitude of V_{EE} (a negative voltage) need not be much greater than three or four volts to make I_E a well-determined quantity.

When the base is held at a Q. pt. value of 0 volts, the Q. pt. value at the emitter is held a diode drop below the base (-0.6 V). The active range of the transistor is determined from the minimum and maximum values that the collector can have (in this case, when the transistor is cutoff and saturated). The collector cannot go below the emitter and it cannot go above V_{CC} in this circuit. A pure AC signal makes equal positive and negative excursions. The AC signal (coming from the input) is superimposed on the Q. pt. level at the collector as shown in figure 6.25. To allow for a maximum AC voltage swing at the collector, the transistor should be biased into the center of its active region(i.e. put the Q. pt. in the center), otherwise one end of the AC signal will hit the min. or max. range prematurely.

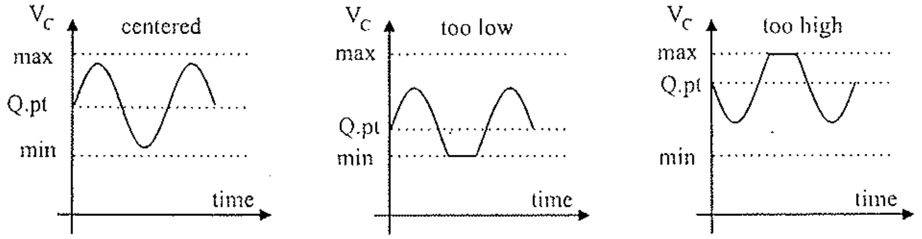


Figure 6.25: Effect of Q. point position on the maximum AC voltage swing.

$\boxed{\text{Exp. 6.7}}$ Build the circuit shown in figure 6.24, using a 2N3904 transistor, with V_{CC}= 10 V and R_C= 470Ω. Select V_{EE} and R_E to place the transistor in the center of its active region (i.e. half way between saturation and cut-off). Its better to choose R_E smaller than R_C to get a voltage gain (at the collector) that is greater than one. Then apply a *very small* signal v_{in} (sine wave centered about ground) and observe V_C and V_E. $\boxed{\text{end}}$

You should notice that v_e follows v_b faithfully, except for the small DC offset $V_{BE} \approx 0.6$ V. Thus, with the emitter as the output terminal, you obtain a voltage gain of almost exactly unity. Accompanying this there is a potentially large current gain involving β of the transistor. The output voltage at the collector, on the other hand, is determined by the ratio of R_C to R_E. Visualize the action of the circuit and check that your measurements agree with the anticipated results.

6.10 Constant-Current Source

The emitter-bias circuit is also an excellent method for making a practical current source (see figure 6.26). If V_{BB} and V_{EE} are constant voltages then:

$$I_C \approx I_E = \frac{V_{BB} - V_{be} - (-V_{EE})}{R_E} \tag{6.26}$$

and the collector current is αI_E. This is almost constant, regardless of V_c, provided only that the transistor remains in the active region. In other words, V_c must not try to go below V_{BB} (nor must it go so high as to break down the transistor).

To make the current source not only have as high an internal impedance as possible, but also to deliver a predictable *absolute* current level, various sources of drift must be considered. Among the most important is the temperature coefficient of V_{be} (-2.5 mV/°C, see chapter 5). V_{be} enters into the circuit determining I_E. First-order temperature compensation can be obtained with the help of a

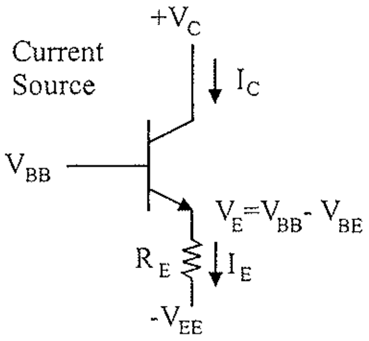


Figure 6.26: Transistor current source.

forward-biased diode, as in figure 6.27, so that $V_B = V_{BB} + V_{diode}$. Then, if the diode and the transistor stay at the same temperature, V_{diode} and V_{be} tend to change by the same amount and V_e remains more nearly constant.

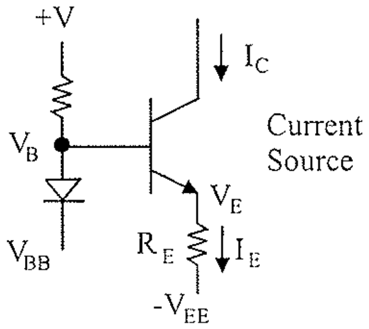


Figure 6.27: Temperature stabilized transistor current source.

6.11 Common Emitter Amplifier

This configuration has both current gain and voltage gain. It starts from the emitter follower configuration and adds a resistor in the collector circuit. Instead of taking the output voltage at the emitter, the voltage at the collector forms the output voltage signal. A schematic is shown in figure 6.28.

As with the emitter follower circuit a small change in voltage at the base ΔV_B is transferred to the emitter $\Delta V_E = \Delta V_B$. The voltage at the emitter in turn produces an emitter current of:

$$\Delta I_E = i_c = \frac{\Delta V_E}{r_e + R_e} = \frac{\Delta V_B}{r_e + R_e} \tag{6.27}$$

The same current, multiplied by α, shows up in the collector resistor R_C and is transformed into a voltage via Ohm's law as:

$$\Delta V_C = \Delta(V_{CC} - I_C R_C) = -\Delta I_C R_C \tag{6.28}$$

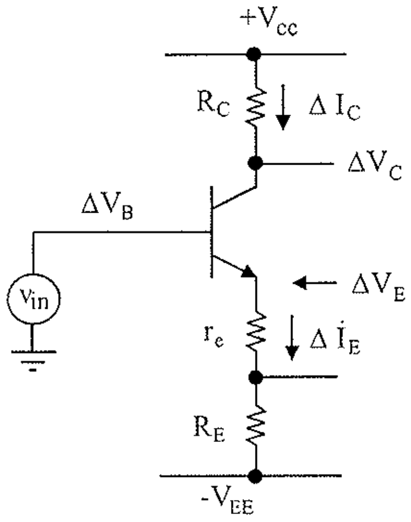


Figure 6.28: Common emitter amplifier.

which leaves:

$$\Delta V_C = -\alpha \Delta I_e R_C \approx -\frac{R_C}{r_e + R_e} \Delta V_b \tag{6.29}$$

(assuming that $\alpha = \beta/(1+\beta) \approx 1$). This means that the overall voltage gain for small AC signals is:

$$G = \frac{\Delta V_C}{\Delta V_B} = -\frac{R_C}{r_e + R_e} \tag{6.30}$$

For reliable determination of the quiescent current, R_e must certainly be much larger than r_e. In effect R_e reduces the gain. This is not all bad, however, because R_e can be a stable and linear component, whereas r_e is strongly dependent on operating conditions and therefore also nonlinear in the presence of large signals. R_e introduces *negative current feedback* into the amplifier, as you can see by considering that the voltage across R_e subtracts from the input signal, and is proportional to the output current. (Strictly speaking $\Delta i_e = \Delta i_c/\alpha$.) The linearizing and stabilizing effect of R_e is sometimes just right for a moderately critical application requiring only moderate voltage gain.

To get higher voltage gain, you can *bypass* R_e for high signal frequencies with a "large enough" capacitor C (as in figure 6.29). Then, for AC signals, the only effective resistance in the emitter circuit is r_e, and the voltage gain becomes:

$$G = \frac{\Delta V_C}{\Delta V_B} = -\frac{R_C}{r_e + R_e//Z_{Ce}} \approx -\frac{R_C}{r_e} = -\frac{R_C}{V_T^*/I_E} = -\frac{V_{CC} - V_C}{V_T^*} \tag{6.31}$$

This is limited only by the voltage, $V_{CC} - V_C$, that can be developed across the load resistor (effectively, by the available supply voltage and the transistor's ratings). As an example, suppose $V_{CC} - V_C$ is 10 V. Then the voltage gain of such an amplifier would be about (10 V)/(40 mV)= 250, with a negative sign to indicate the amplifier inverts.

As with the emitter follower the common-emitter amplifier can also be run from a single ended power supply as shown in figure 6.29. The calculation of the quiescent values of V_B, V_E and I_E follows that of the emitter follower circuit. The collector current is also easy to calculate because it is almost identical to the emitter current. Then the voltage at the collector is just $V_C = V_{CC} - I_C R_C$.

Exp. 6.8 Design and build the amplifier shown in figure 6.29. The design goals for the quiescent point are: a base voltage of V_B= 2 V, a collector voltage V_C= 8 V and emitter and collector currents of 2 mA. Pick capacitors which deserve the label "∞" for a signal frequency of 10 kHz. This amplifier has a very large voltage gain so the function generator signal must be reduced with a voltage divider to avoid saturating the amplifier. Measure the voltage gain and compare to the expected value. Also do some checking to verify that your capacitors are sufficiently large. (In your lab book please state what calculations you made for the capacitors, and what particular measurements you used to check the performance.) **end**

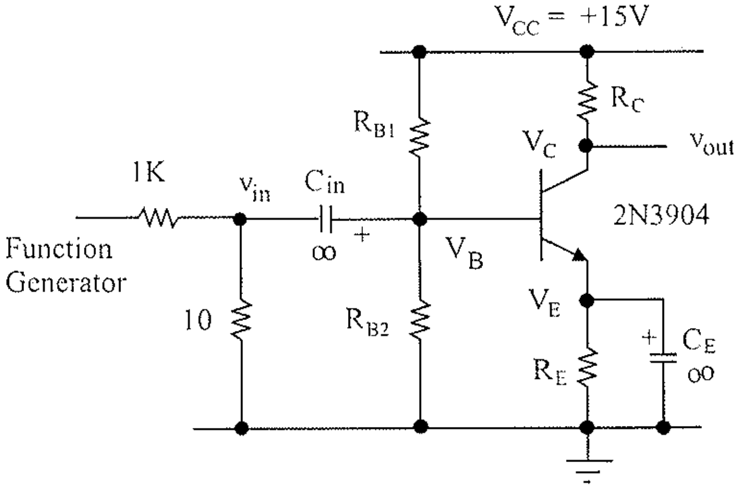


Figure 6.29: Common emitter amplifier with a single supply.

An amplifier like the one you just built, having *ac coupling* and *bypass capacitors*, is best suited for essentially symmetrical signals (zero *dc component*), such as are encountered in audio and rf communications. Where unsymmetrical transients occur, you should make every effort to use *direct-coupled* amplifiers in which problems of charging and discharging capacitors are sidestepped.

6.12 Darlington Pairs

Two transistors can be cascaded directly so that their effective β is the product of the individual β's, and their effective α is extremely close to unity. Such *Darlington pairs* (figure 6.30) are available in single packages, with just the terminals E, B, and C brought out. The resistor connected to the base of Q1 may or may not be present. A Darlington pair may be made from NPN transistors (shown) or from PNP transistors (not shown). In the NPN Darlington pair in figure 6.30, the current flowing into the first base (B) is amplified by factor $1 + \beta_1$ and flows out of the first emitter. This current than becomes the base current for the second transistor which is amplified by another factor of $1 + \beta_2$ flowing out of the second emitter E, with a net current amplification of about $\beta_1\beta_2$ for large β.

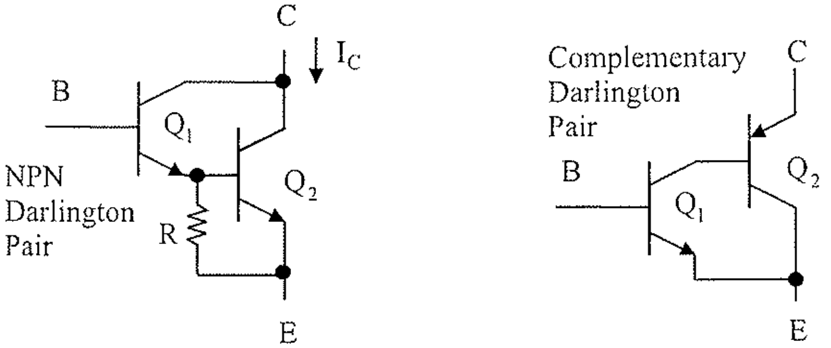


Figure 6.30: left: NPN Darlington pair ($\beta_{total} \sim \beta_1\beta_2$). right: Complementary Darlington pair.

Some points to bear in mind:

(1) There are two forward diodes between B and E, hence twice the V_{be}.

(2) Q_2 cannot be driven into saturation, since this would require V_{CE} for Q_1 to reverse its polarity.

(3) Depending on external connections (or on the possible inclusion of a base resistor R, for Q_2 in the package) the quiescent current in Q_1 is much lower than in Q_2, hence its g_m is much lower.

(4) It is more difficult to ensure clean cutoff for Q_2.

The *complementary* Darlington pair shown in figure 6.30 has only a single junction between B and E, so that its V_{be} is about 0.6V. In addition, because there is no cascading of junctions between B and E, the arrangement has a very large effective g_m, given in fact by $\beta_2 g_{m1}$, as you can see by shorting E to ground and applying a small voltage signal to B. This high effective g_m could be useful in obtaining a very low output impedance in an emitter follower.

6.13 Emitter-Coupled Pair

This configuration has two transistors whose emitters are connected together and tied to a single current source. The NPN version of this circuit is shown in figure 6.31. The input signals (V_{B1} and V_{B2}) are applied to the bases and the output signal is taken at the collectors (V_{C1} and V_{C2}). The input signal should be a small AC signal centered about ground. This circuit also requires that both transistors be a matched pair signified by saying that $Q_1 = Q_2$. This means that both transistors have exactly the same parameters. Producing a matched pair is relatively easy to do in an integrated circuit process because the transistors can be made close together on the same piece of silicon and have exactly the same manufacturing process. Using two transistors of the same type (as in the experiment below) usually works reasonably well too with a little care. The emitter-coupled pair is frequently found inside analog integrated circuits and is the fundamental input stage of almost all op-amps. It turns out to perform as a differential amplifier and is also sometimes referred to as a differential pair.

This first step is to find the Q-point of the amplifier. The Q-point is the DC voltages and currents with no input signal. The input is a small AC signal centered about ground. Note that the input is direct coupled so that the DC level of the input signal is important. Therefore to find the Q-point set the voltage at both bases to zero (or ground). This in turn implies that the voltage at the emitters

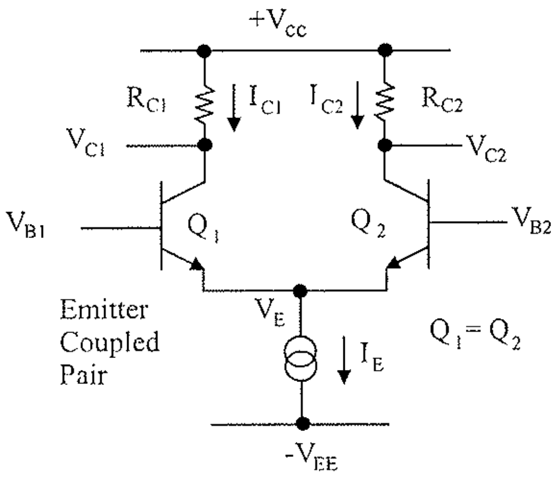


Figure 6.31: Emitter coupled pair.

is one diode drop below ground (V_E =-0.6V). The voltage across the base-emitter junction of both transistors is identically equal. If the transistors are matched this means that the collector current in each transistor is also the same because the base-emitter junction is the controlling junction. The collector currents must end up in the emitter, so:

$$I_E = I_{C1} + I_{C2} \quad \text{and} \quad I_{C1} = I_{C2} = I_E/2 \tag{6.32}$$

(β is assumed large so that the base currents can be neglected). Now knowing the collector currents yields the collector voltages:

$$V_{C1} = V_{CC} - I_{C1}R_{C1} \quad \text{and} \quad V_{C2} = V_{CC} - I_{C2}R_{C2} \tag{6.33}$$

This establishes the quiescent or Q-point values.

Next analyze the circuit for small AC signals. The emitter coupled pair is drawn again in figure 6.32 including the dynamic emitter resistances r_{e1} and r_{e2}. Apply a small AC signal (centered about ground) at the bases of the transistors. The difference of these base voltages appears across the two dynamic emitter resistances and forms a small current of:

$$\Delta I = \frac{\Delta V_{B1} - \Delta V_{B2}}{r_{e1} + r_{e2}} \tag{6.34}$$

This current flows around the loop containing the collector resistors as shown in figure 6.32. The current in the current source I_E is fixed so this AC current cannot flow down to V_{EE}, instead it flow in the upper loop as indicated. If the difference between the base voltages is labeled as the differential input voltage:

$$\Delta V_{diff} = \Delta V_{B1} - \Delta V_{B2} \tag{6.35}$$

then the AC signal at each collector is:

$$\frac{\Delta V_{C1}}{\Delta V_{diff}} = G_1 = -\frac{R_{C1}}{r_{e1} + r_{e2}} = -\frac{R_{C1}}{2r_e} \tag{6.36}$$

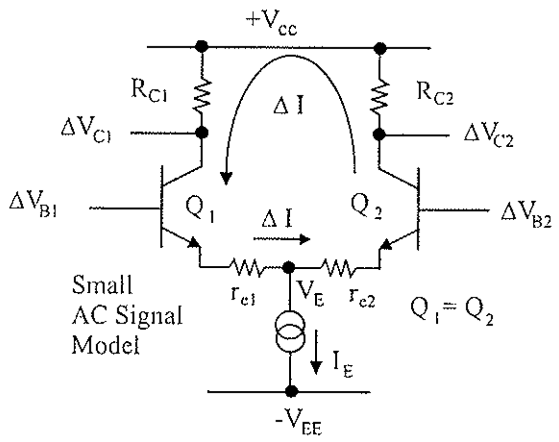


Figure 6.32: Small AC signal model of the emitter coupled pair.

$$\frac{\Delta V_{C2}}{\Delta V_{diff}} = G_2 = \frac{R_{C2}}{r_{e1} + r_{e2}} = \frac{R_{C2}}{2r_e} \tag{6.37}$$

$$\tag{6.38}$$

Note that one signal is inverted and the other is not (i.e. one gain is negative and the other is positive). Because the transistors are matched and $I_{C1} = I_{C2}$, $r_{e1} = r_{e2} = r_e = 2V_T^*/I_E$.

In an op-amp (such as the 741) the current source in the emitter circuit would be made from another transistor. However the circuit will work approximately the same if a large resistor is substituted in the emitter circuit as shown in figure 6.33. If R_E is big compared to r_e and the voltages at the bases do not deviate from zero by more than about 0.1 or 0.2 volts, then the current in the emitter circuit will remain approximately constant as if it were a current source. In addition the small AC signals will flow in the same loop as if there were a real current source in the emitter. Substituting a resistor greatly simplifies the construction of the circuit.

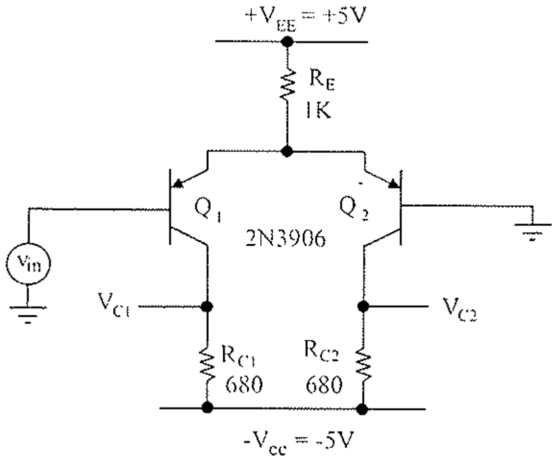


Figure 6.33: PNP emitter coupled pair experiment.

Exp. 6.9 Using the PNP emitter coupled pair shown in figure 6.33, and a small, sinusoidal input signal centered about ground, v_{in}, measure the DC Q-point voltages and the AC voltage gains at the two collectors and compare with values calculated from the expressions given above. (You may need to use the 10K potentiometer as a voltage divider, driven by the function generator, to reduce v_{in} sufficiently.) end

r_{e1} and r_{e2} vary with the quiescent current level in the transistors, so that the amplification is nonlinear for large input signals (it changes with signal level). However, in the vicinity of the point of equal transistor currents, an increase in r_{e1} is accompanied by a decrease in r_{e2}, so that there is first-order cancellation of the change. Where greater linearity is required, at the expense of some gain, negative current feedback can be introduced by including two fixed resistors R_e between the emitters, thus adding a constant amount to the term $(r_{e1} + r_{e2})$.

6.14 Power in Transistor Circuits

Consider trying to drive a load such as a speaker (typically 8 Ohms) using a common emitter amplifier (as in figure 6.34). Audio signals cover the approximate range of 20 Hz to 20 kHz so AC capacitive coupling should work reasonably well. This is officially called a class A amplifier because the transistor is biased so that it is always ON and drives both the positive and negative half of each AC signal. The output impedance of this amplifier is R_C, so to deliver the maximum amount of power to the load (speaker) the collector resistor R_C should also be about 8 Ohms. The quiescent value of the collector voltage V_C should be about half way between V_E and V_{CC} to allow the maximum symmetrical swing of the output voltage. Usually V_E will be a small voltage so this means that when no signal is applied (v_{in}=0) there is a substantial amount of power dissipated in the transistor (with none going to the speaker). This is not very efficient. Most of the power is thrown away in the bias circuit (in fact it can be shown that the maximum efficiency of a class A amplifier is 50%).

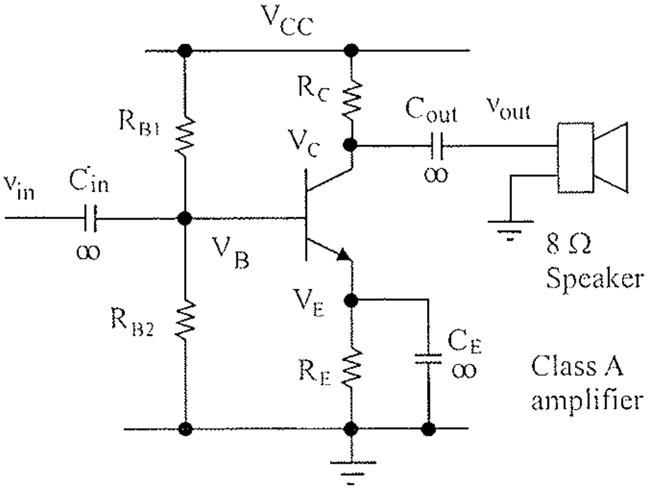


Figure 6.34: Class A amplifier.

The power dissipated in the transistor (and not to the load) is the product of the current through the transistor and the voltage across the transistor. The power in the transistor roughly follows the curve in figure 6.35. The voltage across the transistor goes to zero in saturation and the current through the transistor goes to zero at cutoff. Therefore the power in the transistor approximately goes to zero at cutoff and saturation and peaks in the center of its active region (i.e. where the class A amplifier is normally biased). Transistor power is what heats the device and threatens to destroy it. The simple class A amplifier is not very good for high power applications because a large fraction of the power is thrown away in heating the transistor and not delivered to the load.

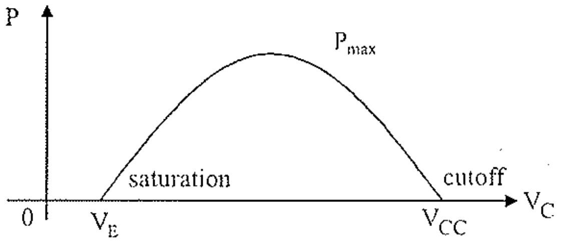


Figure 6.35: Power as a function of collector voltage V_C.

An emitter follower circuit on the other hand can be biased so that is in cutoff (low power) when no signal is applied. A single emitter follower can only supply power for one half of the AC signal (positive for NPN and negative for PNP). However if two complementary emitter followers are arranged as shown on in figure 6.36, both halves of the waveform can be amplified. This type of configuration is called a class B push-pull amplifier because each transistor is only on for one half of the cycle (one pulls and the other pushes). Q1 (NPN) is responsible for pushing the output high and Q2 (PNP) is responsible for pulling the output low. This is also called a complementary stage because it uses both an NPN and a PNP transistor. This amplifier draws no power when no signal is applied. It has a voltage gain of unity but a substantial current gain and hence power gain. It is well suited for driving a low impedance load such as a speaker. However it has one serious draw back. The base emitter diode drops in the NPN and PNP are in the opposite direction. When v_{in} rises above -0.6V the PNP (Q2) turns off but the NPN (Q1) does not turn on again until v_{in} rises above +0.6V. This is referred to as cross-over distortion or zero-crossing distortion and is illustrated in figure 6.36. The input is assumed to be a sine wave (dashed line) and the corresponding output is shown as a solid line.

Cross-over distortion may be approximately corrected by adding an op-amp with negative feedback taken from the output of the emitter followers, as in the circuit shown in figure 6.37. The op-amp can supply a large voltage gain but most op-amps cannot drive more than about 10 mA of output current (there are specialized power op amps that have large output currents, but these generally have extra power transistors internally). The output transistors can supply a much larger current to the load. The feedback for the op-amp is taken from the output of the transistors so the large voltage gain of the op-amp will correct for any nonlinearities in the circuit. An op-amp alone will not quite work because it takes a small amount of time to slew between -0.6V and +0.6 volts, however this circuit works reasonably well and is easy to build

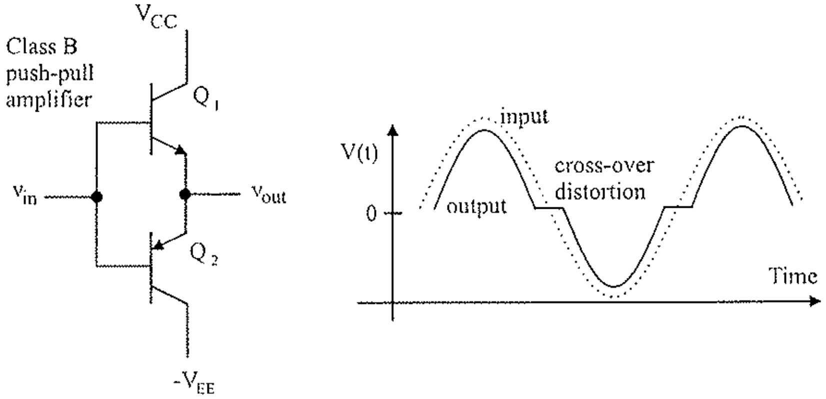


Figure 6.36: Simple class B push-pull amplifier (left) and its cross over distortion (right).

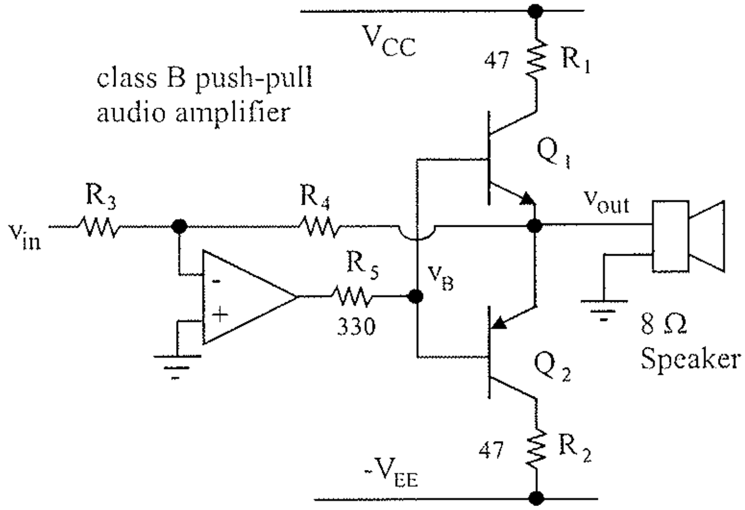


Figure 6.37: Push-pull amplifier with negative feedback to correct the cross-over distortion.

Exp. 6.10 Build the complementary emitter follower circuit shown in figure 6.37. Use a 741 op-amp driven from +/- 15 Volt supplies (also used for V_{CC} and $-V_{EE}$). Choose R_3 and R_4 to yield an overall voltage gain of about 10 and an input impedance of 1K. Because of the large currents involved, this circuit requires good bypass capacitors and a clean layout (avoid long wires in sensitive areas of the circuit) to prevent spurious oscillations. First test it with a 10 Ohm load resistor in place of the speaker and use a function generator set to deliver a 0.1 Vpp sine wave at 1 kHz as input. Sketch v_{in}, v_B and v_{out} on the same time scale. Does this circuit correct the cross-over distortion? Explain how this circuit works in your analysis. After the circuit is working with a load resistor and a function generator, connect a speaker to the output (in place of the load resistor) and drive the input from the earphones socket of one of the small radios provided in the lab. There should be a small adapter cable provided to do this. With the radio tuned to a nearby station you should be able to hear music coming form the speaker. Calculate approximately how much power can be supplied to the speaker (hint: the maximum peak power occurs when one transistor is saturated). end

The two 47 Ohm resistors (R_1 and R_2) serve to limit the maximum current that can flow through Q1 and Q2 and to protect the circuit in case of an accidental short circuit. R_5 also serves to protect the op-amp output. Because this circuit can deliver a significant output current, adequate bypass capacitors (not shown) are essential.

Cross-over distortion can be further corrected by adding two diodes in appropriate places to cancel out the base emitter diode drop in addition to a negative feedback circuit as shown below. Alternately you can use two more transistors in place of the diodes in a Darlington type configuration. An improved circuit using two addition transistors is shown in figure 6.38. This base-emitter diode drops of Q3 and Q4 just cancel the base-emitter diode drops of Q1 and Q2 so that v_{out} is at the same level as the op-amp output within a small fraction of a diode drop (i.e. the diode drops may not be exactly equal). The negative feedback of the op-amp (through R_3 and R_4) will compensate for the remaining non-linearities. The op-amp supplies voltage gain and corrects for non-linearities and the transistors supply current gain and increase the available output current. The additional two transistors also have the added advantage of increasing the overall current gain. This four-transistor combination is available in the LH0002 (National Semiconductor).

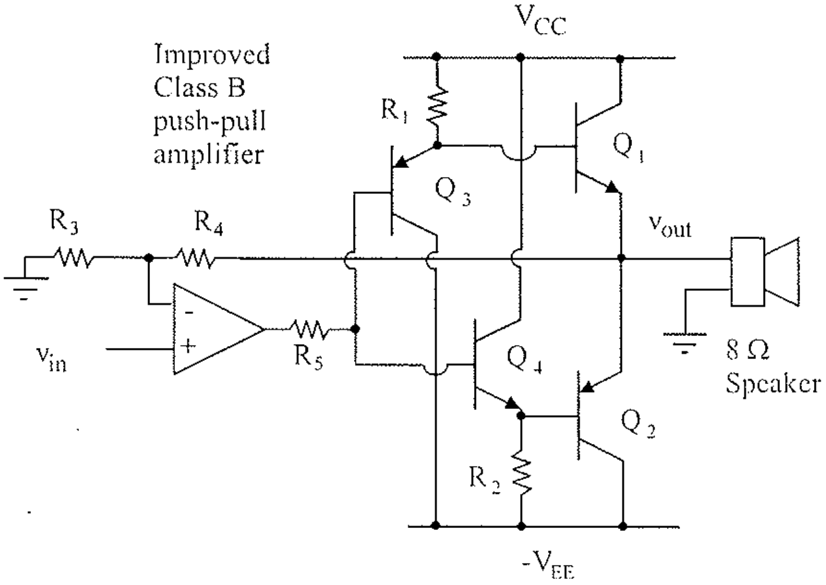


Figure 6.38: Push-pull amplifier with better correction for cross-over distortion.

Audio amplifiers with large output currents and some of the features discussed above are available in convenient IC form. For example see the following: LM12, LM380, LM383, LM384, LM386, LM675, and the LM1875. The LM1877 is even designed for stereo amplifiers and has two amplifiers with 2 watts each in a single 14 pin package (requires a heat sink though), and the Samsung KA2209B is a one Watt stereo amplifier in an 8 pin package that can be run from a single 9 volt supply or battery (it uses AC coupling).

	2N3904 (NPN)	2N3906 (PNP)	
maximum C-E voltage	40 V	40 V	
maximum C-B voltage	60 V	40 V	
maximum E-B (reverse) voltage	6 V	5 V	
maximum collector current	200 mA	200 mA	
maximum power	350 mW	350 mW	(at 25 °C)
C-E saturation voltage	0.2 V	0.25 V	(at Ic=10mA)
B-E saturation voltage	0.65-0.85 V	0.65-0.85 V	(at Ic=10mA)
current gain at audio frequencies	100-300	100-300	(at Ic=10mA)
current gain at 100 MHz	3	2.5	(at Ic=10mA)
delay time	35 nS	35 nS	(at Ic=10mA)
storage time	200 nS	225 nS	(at Ic=10mA)

Table 6.1: Some typical parameters for bipolar junction transistors (BJT).

6.15 Transistor Parameters

Some typical values of transistor parameters are shown in table 6.1.

6.16 Practice Problems

[1] The circuit in figure 6.39 is used to turn on an incandescent lamp that requires 50 mAmp to light up. Assume that v_{in} is a 10 V_{pp} square wave centered about ground, R_B= 10K, R_C = 1K, V_{CC} = 10 V, a base-emitter diode drop of 0.6 V and a collector-emitter saturation voltage of V_{CE-SAT}= 0.2 volts.
a) Find the minimum β for each transistor that will ensure that Q_1 saturates and that the lamp is turned on for one half cycle of v_{in} (not necessarily at the same time).
b) Is the lamp on when v_{in} is high or low?

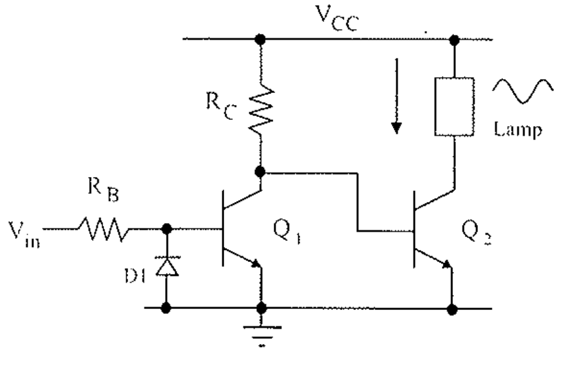


Figure 6.39: Problem 1.

[2] The circuit in figure 6.40 uses a transistor, Q_1 to turn on an LED (light emitting diode). The LED emits light (usually red) when a forward current is passed through it. Unlike a silicon diode with a forward voltage drop of 0.6 V, LED's typically have a larger diode drop of 1 to 2 volts. Assume that this diode has a forward diode drop of 1.8 volts at 5 mA and that 5 mA is required to produce light. Also assume that $V_{CC} = 5$ V, and that the transistor has a base-emitter diode drop of 0.6 V and a collector-emitter saturation voltage of V_{CE-SAT}= 0.2 V.
a) If the input voltage v_{in} is either 0 or 3 volts (i.e. one of two values) find the value of R_B that limits the current drawn from v_{in} to 0.2 milliAmps.
b) Find the value of R_C that allows the required current in the LED when Q_1 is saturated.
c) What is the minimum β for this transistor that will light the LED?
d) Is the LED on when v_{in} is high or low?

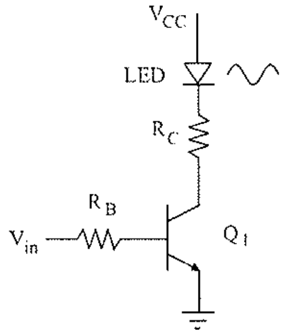


Figure 6.40: Problem 2.

[3] The transistor amplifier in figure 6.41 is a simple op-amp. The first stage is a differential amplifier. v_a and v_b are small AC input signals centered about ground. The second stage provides some additional gain and shifts the Q-pt of v_{out} back near ground. With a large β the loading of the first stage by the second may be roughly neglected. You may assume a base emitter diode drop of 0.6V, $\beta >>1$, V_T^*=40 milliVolts and $Q_1 = Q_2$.
a) Calculate the Q-pt values V_{E12}, I_{E12}, I_{C2}, V_{C2}, V_{E3}, I_{E3}, I_{C3}, and V_{C3}. There is a non-zero output voltage with no input voltage: This is related to the input offset voltage of this op-amp.
b) What is the small signal AC gain of the first stage $G = v_{C2}/(v_a - v_b)$ and the second stage $G = v_{out}/v_{C2}$ of the amplifier?
c) What is the total gain of the amplifier $G = v_{out}/(v_a - v_b)$?

[4] Assume a base emitter diode drop of 0.6 volts, $\beta \gg 1$, and that V_T^*=40mV. In Exp 6.6 (emitter follower, figure 6.20) in the Lab manual:
a) Calculate the quiescent voltages at the base and emitter of the transistor.
b) Calculate the internal emitter resistance r_e of the transistor.
c) Calculate the small signal AC voltage gain of the circuit for frequencies such that C_1 and C_2 may be replaced by short circuits. Assume R_L is not there. (Note, when you do this in lab the output will be

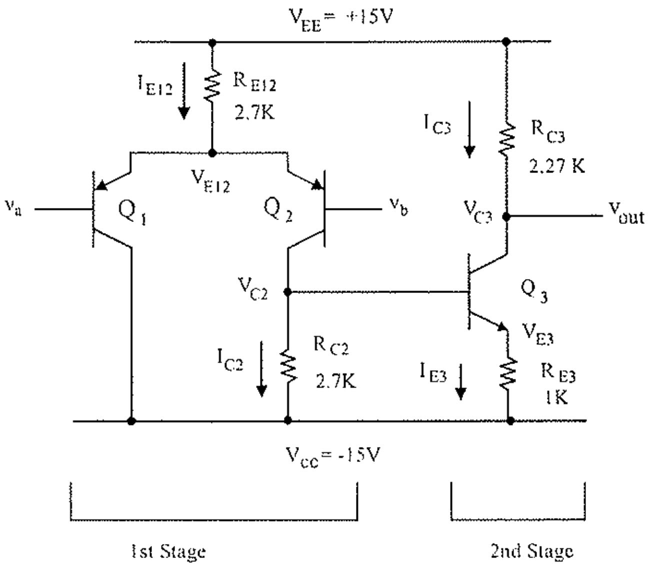


Figure 6.41: Problem 3.

rather distorted for large amplitude, however for this problem assume that the signals are very small and undistorted.) How much does this differ from the ideal gain of unity for an emitter follower?
d) repeat c) with $R_L = 47\Omega$.

[5] The circuit in figure 6.42 delivers a constant current I_{LOAD} to a variable load. It uses a 741 op-amp (A1) with negative feedback to stabilize the current in the load. The op-amp is driven from ±15 V supplies, V_{CC} = 15 V, and R_S = 10 Ohms. This type of circuit is frequently used to drive a large electromagnet (as the load) to generate a constant magnetic field.
a) A 741 op-amp cannot provide more than about 10 mAmp of output current. If the load needs 100 mAmp, what is the minimum β of Q_1 that will make this circuit perform properly?
b) Derive an expression relating the load current I_{LOAD} to the input control voltage V_{IN}.

[61] The circuit shown in figure 6.43 is to be used for frequencies of 10kHz or higher. V_{CC}= 15V, R_1= 4.7K, R_2= 10K, R_3= 1K, R_4= 1K, and $C_1 = C_2 = C_3 = 200\mu$fd. You may assume that $\beta >> 1$, V_T^*=40mV and a base-emitter diode drop of 0.6 volts.
a) Calculate the Q-pt values for V_1, V_3, V_4, I_3 I_4.
b) Calculate the small signal AC gain $G = v_{out}(\omega)/v_{in}(\omega)$.

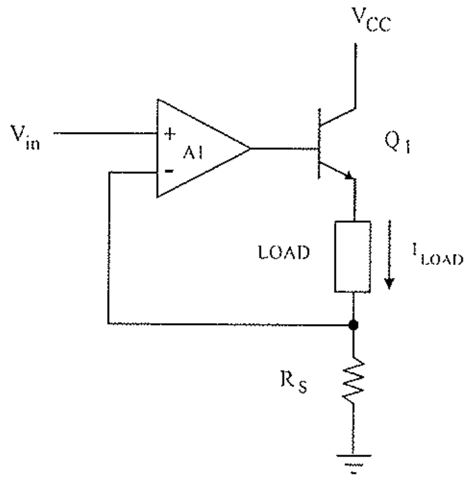


Figure 6.42: Problem 5.

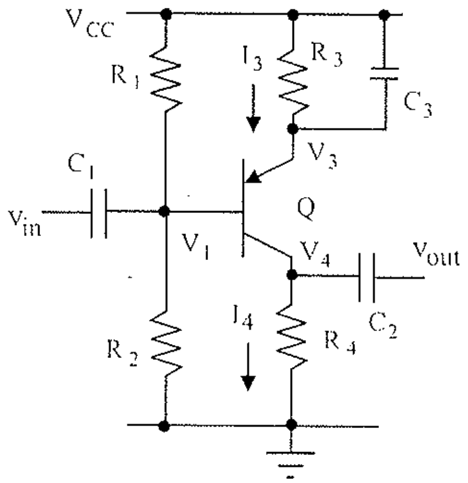


Figure 6.43: Problem 6.

Chapter 7

MOSFET TRANSISTORS

This chapter covers the Metal Oxide Semiconductor Field Effect Transistor, or MOSFET, which is one of several types of field effect transistors (FET's). There are probably more MOSFET's in the world than any other type of transistor because it is used in huge numbers in digital IC's. Microprocessors may have 100 million or more MOSFET's in a single (large) integrated circuit. The MOSFET generally has three terminals and amplifies just like a bipolar transistor. (Some MOSFET's may have four terminals but usually two of these are shorted together to make a three terminal device.) The MOSFET comes in two basic types, enhancement mode and depletion mode with two polarities each, n-channel and p-channel (like NPN and PNP). No current flows in an enhancement mode device with zero voltage applied at the gate and the current must be "enhanced" (or increased) by applying a voltage. Current normally flows in a depletion mode device with zero voltage applied to the gate, and must be "depleted" (or reduced) by applying a voltage. The depletion mode MOSFET behavior is also similar to the junction field effect transistor (JFET). This chapter will primarily focus on the enhancement mode MOSFET. For a more detailed discussion of FET's see (for example) chapter 5 of Sedra and Smith, *Microelectronic Circuits, fourth edit.*.

In general, field effect transistors have a very high input impedance compared to a bipolar junction transistor. In particular the input to the MOSFET (the gate) is an insulator (silicon dioxide) instead of a forward biased pn junction as in a bipolar (npn or pnp) transistor. This property can be very helpful when measuring low current signals. Unfortunately, FET's are also more difficult to use as linear amplifiers because they are not very linear, and their parameters are not easily controlled. There is a trade-off between ease of use and high input impedance. Frequently the input stage of an amplifier will be a FET (for high input impedance) and later stages may be BJT's for ease of use and controllability. This nonlinearity is not a problem with digital devices because only two states are used (ON and OFF). The nonlinearity may be corrected using negative feedback to make a linear amplifier using FET's.

The bipolar junction transistor (BJT, NPN or PNP) was found to be easiest to describe as a current controlled current source. The MOSFET (and JFET) will be easiest to model as a voltage controlled current source. A controlling voltage (on the gate) will produce an electrostatic field that controls a current.

7.1 MOSFET Structure and Circuit Symbols

A cross section of an n-channel enhancement mode MOSFET is shown in figure 7.1 and a cross section of a p-channel MOSFET is shown in figure 7.2. In both types of devices the main connections are from the drain and source which will function similarly to the collector and emitter in a bipolar transistor. The gate is electrically isolated from the rest of the transistor by a thin insulating oxide layer usually less than approx. 0.1 microns. The gate can be used to control the current flowing from drain to source by creating an electric field in the region between the drain and the source (hence the name field effect transistor). The thin oxide layer can be easily be destroyed by static discharge from handling it. You will need to be careful how you handle these transistor. If you drag your feet while walking across the room (particularly when the humidity is low), and pick up a MOSFET transistor you may inadvertently destroy it by producing a small (imperceptible) electrostatic discharge into the gate. Most MOSFET devices are shipped in special conducting plastic or other materials to reduce the possibility of static discharges.

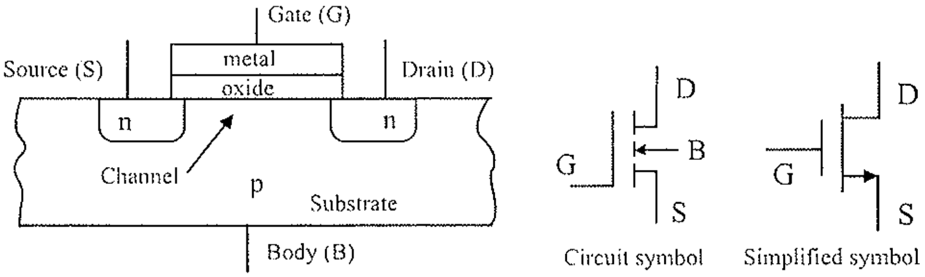


Figure 7.1: N-channel enhancement mode MOSFET (no applied voltage). The physical structure is on the left, the circuit symbol is in the middle and a simplified circuit symbol is on the right (body/substrate and source connected together).

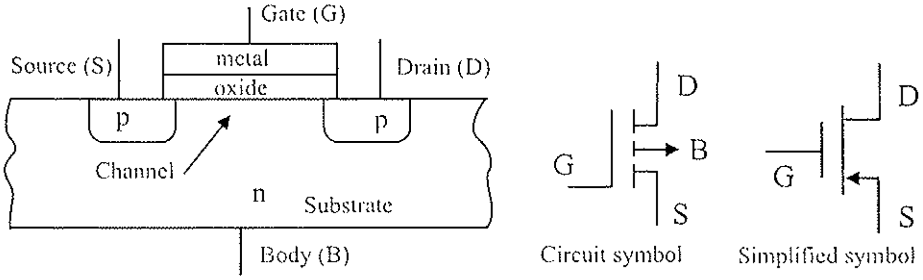


Figure 7.2: P-channel enhancement mode MOSFET (no applied voltage). The physical structure is on the left, the circuit symbol is in the middle and a simplified circuit symbol is on the right (body/substrate and source connected together).

The official circuit symbol for each device is shown near the center of each figure (labelled "Circuit symbol" in figure 7.1, 7.2). The gate is shown as a straight line physically separated from the rest of the device. The connection between drain and source is shown as a dashed line indicating that it is an enhancement mode device (a depletion mode device has a solid line between drain and source). The substrate or body of the device is shown as an arrow pointing in for N-channel and out for P-channel indicating the direction of the p-n junctions at the drain and source. The official symbols is tedious to draw and the simplified symbol shown on the right of each figure (labelled "Simplified symbol" in figure 7.1, 7.2) may be used for the case where the body or substrate is shorted to the source. The arrow points in the direction of flow of positive current in all four symbols (the substrate is normally reversed biased) just like the bipolar transistor and diode symbols.

7.2 N-Channel MOSFET Function

This discussion will focus mainly on the n-channel device. The p-channel device works in the same manner but with the polarity reversed, so if you understand one of them then you also should understand the other (just change the sign of all voltages and currents).

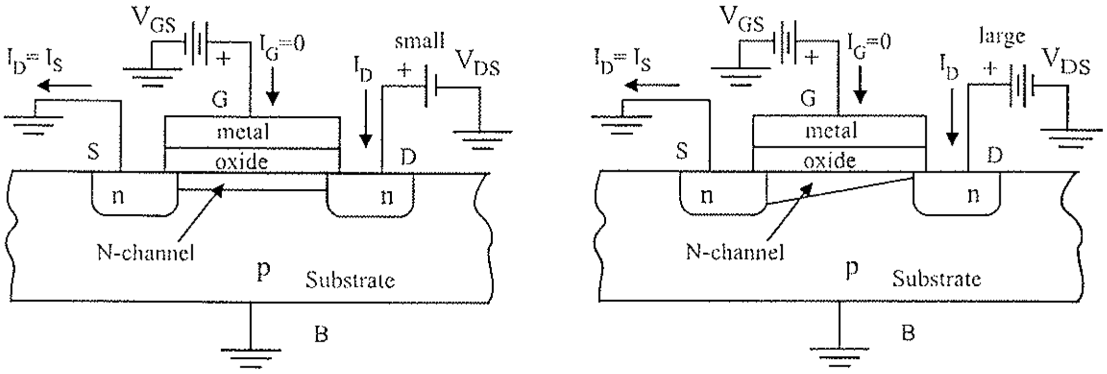


Figure 7.3: N-channel enhancement mode MOSFET (voltage applied). With a positive gate voltage above V_T there is a small uniform channel formed between the drain and source for small V_{DS} (left, triode or Ohmic region) and a large tapered channel formed for large V_{DS} (right, saturation region).

In the enhancement mode n-channel device (figure 7.1,7.3) there is a p-type region between the n-type drain and source (again referring to the n-channel flavor of this device). The source S and drain D are separated by two back-to-back diodes so no current can flow between them with zero voltage on the gate. This situation can be called *cutoff* as with the bipolar transistor. When a positive voltage is applied to the gate (with the body or substrate grounded), the positive holes in the p-type channel are repelled and any residual electrons (n-type carriers) are attracted into the channel (figure 7.3). When the gate voltage is large enough there will be more electrons than holes in the channel (assuming the materials have proper levels of dopants) and current may flow between the drain and the source. The value of V_{GS} (the voltage between the gate and the substrate or source in this case) at which a conduction channel is formed is called the threshold voltage V_T. V_T is positive for an n-channel device and negative for a p-channel device. Because the gate is separated from the rest of the device by an insulating oxide layer, no current flows into the gate for DC signals (the small capacitance of the gate may require a small current during a transition on the gate to charge and discharge).

If a small voltage (about 0.1 to 0.2 volts) is applied to the drain D with a positive voltage on the gate ($V_{GS} > V_T$) then a small uniform conduction channel is formed connecting the drain D and source S as on the left side of figure 7.3. The width of this channel varies approximately linearly with the applied gate voltage and the current between the drain and source varies as if these two points (D and S) were connected with a resistor whose value is controlled by the gate voltage. This is called the *Ohmic or Triode Region.* The MOSFET can be used as a voltage controlled resistor in this configuration.

If a large voltage (several volts) is applied to the drain D, then the channel depth is no longer uniform along the channel as shown on the right hand side of figure 7.3. If the drain D has a voltage of V_{DS} and the source S has zero volts, then the voltage along the channel varies between these to values. The field between the gate and the channel depends on the difference in voltage between the gate (constant of V_{GS}) and the voltage in the channel (zero near the source and V_{DS} near the drain). The channel depth (and hence conductivity) varies with this field giving a sloping channel depth as on the right of figure 7.3. This is called the *saturation region.* An increase in V_{DS} is balanced by a decrease in the channel conductivity producing a constant current approximately independent of voltage applied between the drain D and the source S, V_{DS}. In saturation the MOSFET can be though of as a current source that is controlled by the gate voltage V_{GS}. If the drain voltage is large enough so that the conduction channel depth goes to zero (at the drain end) then the device is said to be in *pinch-off.*

7.3 N-MOSFET Current-Voltage Characteristics

With $V_{GS} < V_T$ no current flows in the MOSFET which is said to be cutoff. With $V_{GS} > V_T$ and $V_{DS} > 0$ the n-channel MOSFET can be in either the Ohmic/Triode region or in the saturation region. The boundary between these two regions is defined as:

$$V_{DS} = V_{GS} - V_T \quad \text{Boundary} \tag{7.1}$$

$$V_{DS} < V_{GS} - V_T \quad \text{Ohmic} \tag{7.2}$$

$$V_{DS} > V_{GS} - V_T \quad \text{Saturation} \tag{7.3}$$

7.3.1 Ohmic or Triode Region

In the Ohmic or Triode region ($V_{DS} < V_{GS} - V_T$) the current flowing between the drain and source is given by:

$$I_{DS} = K\left[2(V_{GS} - V_T)V_{DS} - V_{DS}^2\right] \quad \text{Ohmic Region} \tag{7.4}$$

where V_T is the *threshold voltage* of the MOSFET and K is a constant.

$$K = \frac{1}{2}\mu_n C_{OX}\frac{W}{L} \tag{7.5}$$

where W and L are the width (distance into the paper in figure 7.1) and length (distance between drain and source) of the gate region, μ_n is the electron mobility and C_{OX} is the capacitance per unit area of the gate and channel. The constant K has units of mAmp/V^2 or Amp/V^2. For small V_{DS} of less than about 0.2 volts I_{DS} and V_{DS} are approximately proportional to each other as in a resistor as in figure 7.4. The symbol K is not very standard in the literature. Some books may use different symbols and some may or may not include W and L in the definition of this constant. You should be prepared to see different symbols in other places although this discussion will stick to a consistent definition.

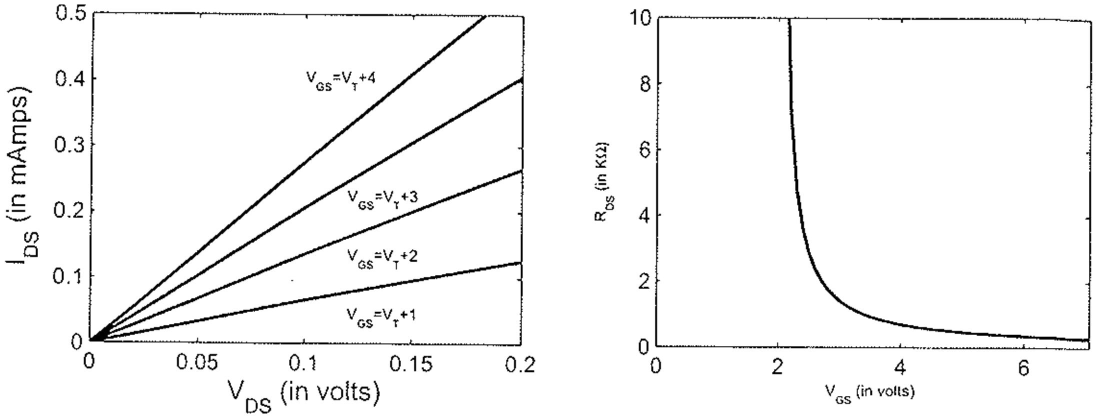


Figure 7.4: Current-voltage characteristics (left) of an n-channel enhancement mode MOSFET in the Triode or Ohmic region (small V_{DS}, K=0.35 mA/V^2). The drain-source resistance (equation 7.6) is shown on the right (V_T=2.0 V).

If the drain-source voltage is very small then the resistance between the drain and source is approximately:

$$R_{DS} = \frac{V_{DS}}{I_{DS}} \sim \frac{1}{2K(V_{GS} - V_T)} \quad \text{Ohmic Region} \qquad (7.6)$$

which is shown on the right side of figure 7.4. This voltage controlled resistance might typically be used to form an automatic gain control (AGC) circuit in which the gain of an amplifier is automatically adjusted using a control voltage to change the resistance in the feedback circuit of an op-amp amplifier.

7.3.2 Saturation Region

In saturation with $V_{GS} \geq V_T$ and $V_{DS} \geq V_{GS} - V_T$, the drain-source current is approximately given by:

$$I_{DS} = K(V_{GS} - V_T)^2 \quad \text{Saturation Region} \qquad (7.7)$$

The constant K is the same for both the Ohmic and saturation regions. A set of theoretical curves for an n-channel MOSFET including both the Ohmic region and the saturation region is shown in figure 7.5 and the drain current I_{DS} is shown in figure 7.6 as a function of the gate voltage V_{GS}.

Both constants K and V_T are properties of the transistor (determined by the manufacturer) and both will vary slightly with temperature. The magnitude of $|V_T|$ decreases by about 2 mV for every one degree C rise in temperature however K decreases with temperature and the net drain current also decreases with temperature.

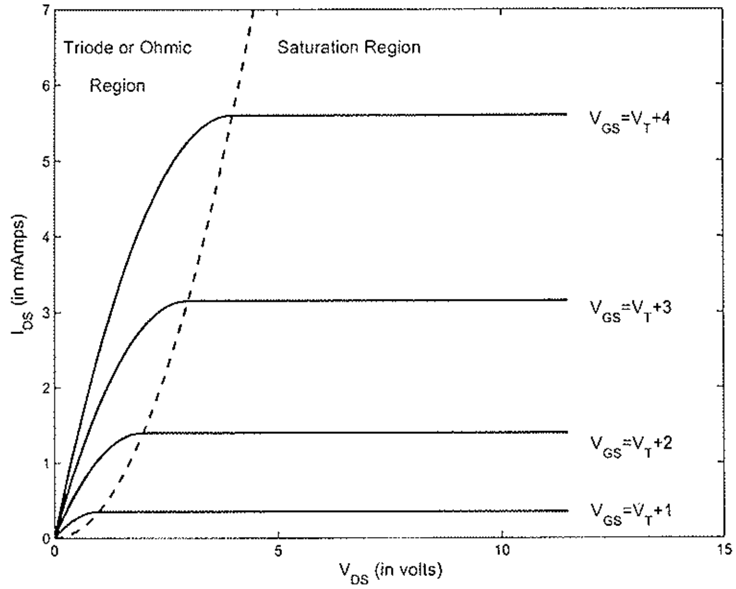


Figure 7.5: Current-voltage characteristics (I_{DS} vs. V_{DS}) of an n-channel enhancement mode MOSFET (K=0.35 mA/V^2). The dashed line is the boundary between the Ohmic and saturation regions (eq. 7.1).

If the drain-source voltage becomes large (in the saturation region) then the point at which the conduction channel is pinched off (right hand side of fig. 7.3) can move away from the drain toward the source. This effect causes the flat portion of the drain current versus drain-source voltage curve (fig. 7.5) to rise slightly with drain-source voltage and manifests itself as a finite output resistance in the saturation region (i.e. the transistor is not a perfect current source). This channel length modification can be modeled as an extra factor on the end of equation 7.7:

$$I_{DS} \quad = \quad K(V_{GS} - V_T)^2(1 + \lambda V_{DS}) \quad \text{Saturation Region} \qquad (7.8)$$

$1/\lambda$ has units of volts and is typically in the range 20 to 200 volts.

In saturation the MOSFET current is approximately independent of the voltage across the drain and source so it can be modeled as a voltage controlled current source as shown in figure 7.7.

7.3.3 Measurement

In the following experiments you will be using a 2N7000 n-channel MOSFET. The pin-out for this device is shown in figure 7.8. This MOSFET has a typical threshold voltage of 2.1 V with a range of 0.8 to 3.0 volts. It has a maximum rating of 60 V (drain to source), a maximum continuous current of 200 mA and a switching time of 10 nS. There is a (normally) reverse biased diode connected between the drain and the source resulting from the internal connection between the source and the substrate. This diode also protects the transistor from being damaged from transient voltage spikes as might occur when driving an inductive load such as a relay. Please consult the manufacturer's data sheet for more details on this device.

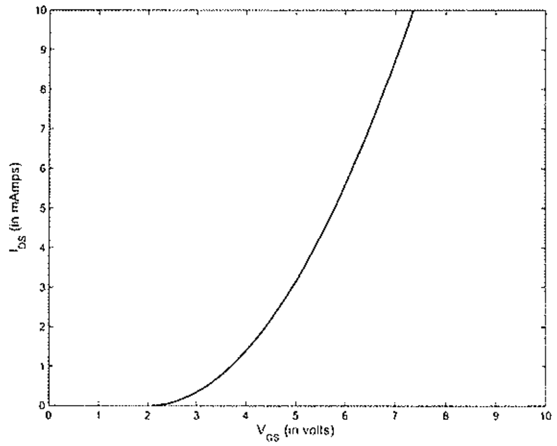


Figure 7.6: Current-voltage characteristics (I_{DS} vs. V_{GS}) of an n-channel enhancement mode MOSFET (K=0.35 mA/V^2, V_T=2.0 V) in the saturation region (eq. 7.7).

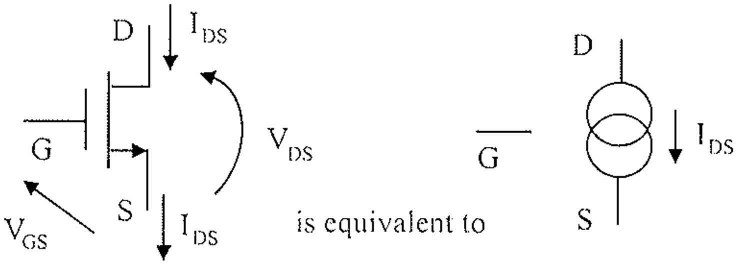


Figure 7.7: The n-channel MOSFET circuit model in the saturation region (right) and definitions of currents and voltages on the circuit symbol (left).

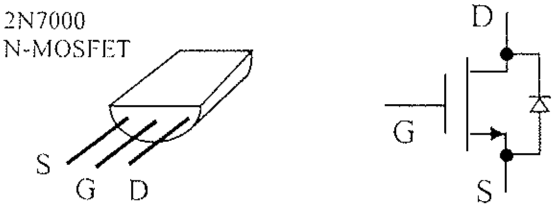


Figure 7.8: 2N7000 N-MOSFET pin-out (D=drain, S=source, G=gate). This particular MOSFET has an extra diode resulting from the internal connection between the source and the substrate or body (shown in schematic symbol). This diode also protects against reverse voltage spikes when switching inductive loads.

Most MOSFET's are inside integrated circuits and the few that are available as discrete components (single transistors) tend to be designed for use as high power (current and voltage) switches and amplifiers. There are a few medium current devices such as this MOSFET that can also be used as small signal amplifiers although their properties may not be ideal for this application. In particular you may find that the MOS parameter K is much higher than the value used in fig. 7.4, 7.5, 7.6 and the on-state resistance is much lower.

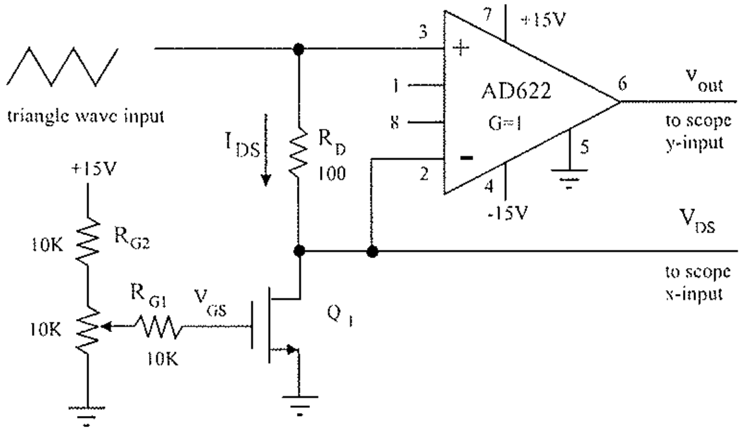


Figure 7.9: Circuit to display the I-V curve of a N-MOSFET transistor on the oscilloscope.

Exp. 7.1 Setup the circuit shown in figure 7.9, to display the I-V characteristics (I_{DS} versus V_{DS}) of a 2N7000 MOSFET transistor on an XY scope. The AD622 instrumentation amplifier (see figure 5.10) with a differential gain of unity (no resistor between pins 1 and 8) is used to convert the current flowing through the drain resistor R_D into a voltage to display on the Y axis of the scope. The drain-source voltage V_{DS} will be the X axis. Set the function generator to deliver a triangle wave of about 100 Hz to 1 KHz with an amplitude of -0.5 V to +10 V to start (you may have to increase the range a little to see the whole curve for some MOSFETs, but keep it within ±14V for the AD622).

The gate-source voltage of the transistor, V_{GS}, can be varied with 10K potentiometer. Observe how the I-V curve changes on the scope as you vary V_{GS}. Sketch a typical curve of I_{DS} vs. V_{DS} for 2 or 3 different value of V_{GS} in your lab book.

1. Saturation Region You will need to find a value for the MOSFET parameters K and V_T to use in the experiments that follow. The MOSFET behavior is a little more complex than the BJT, so this will require a little more effort than the BJT measurements you did in the last chapter. Choose about five (5) different settings of the gate-source voltage to yield drain currents about equally spaced over the available range. At each of these voltages, record the drain current at a drain-source voltage of about 6 to 8 volts, in the saturated region, (measured on the scope) and the corresponding gate-source voltage (use a handheld digital voltmeter to measure the voltage V_{GS}).

Plot your measured values of I_{DS} vs. V_{GS} in saturation, which should yield a parabolic curve. The point where I_{DS} drops to zero is V_T. However it is difficult to get an accurate value from this curve because the slope of the curve goes to zero here as well (also, I_{DS} will never be exactly zero because of various small leakage currents that have not been accounted for in our circuit model). To get accurate values of both MOSFET parameters plot $\sqrt{I_{DS}}$ vs. V_{GS} which should yield a straight line with an intercept on the horizontal axis of V_T and slope $\sqrt{K}$ with a reasonable accuracy. Make this plot and extract a value for K and V_T.

2. Ohmic Region Measure the slope of I_{DS} vs. V_{DS} in the Ohmic region ($0 < V_{DS} < 0.5$ volt) for a few values of the gate voltage V_{GS}. This slope is related to the inverse of the resistance between the drain and the source.

NOTE: Be sure to save this MOSFET and use it for the next few amplifier experiments in this chapter. You will need to refer to the curves measured in this experiment.

OPTIONAL-1: Measure the small slope of the I_D vs. V_{DS} curve in the saturation region near $V_{DS} \sim 8$V to get a value for the λ parameter (equation 7.8).

OPTIONAL-2: You can verify that the MOSFET is voltage controlled and not current controlled by measuring the current flowing into the gate. Connect a large resistor in series with the gate and measure the voltage across it with the handheld digital voltmeter. There should be no current and hence no voltage across this series resistor.

OPTIONAL-3: Setup the circuit shown in figure 7.10, to directly display the I_{DS} vs. V_{GS} curve of a MOSFET transistor on the XY-scope. Set the input triangle wave to deliver an amplitude of about 0 V to +4.0 V (or up to about +7.0 V if your MOSFET has a high V_T) at about 100 Hz to 1 kHz. The AD622 instrumentation amplifier has a maximum input voltage that is approx. 1.4 volts less than the supply voltage. Diodes D_1, D_2 and D_3 (use a 1N914 signal diode) serve to reduce the input voltage to the AD622 to an acceptable level. end

It is best to keep the same MOSFET for the next three amplifier experiments in this chapter to use the parameter you just measured.

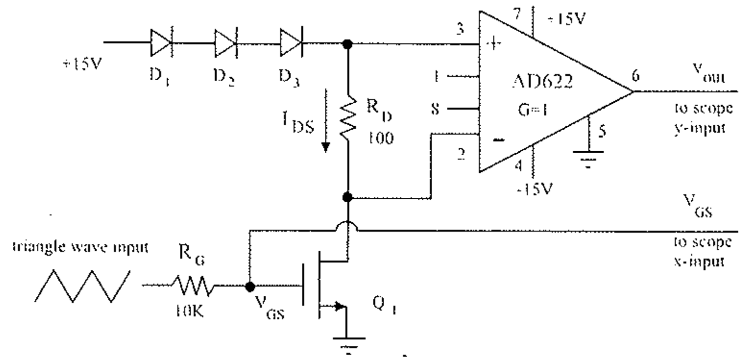


Figure 7.10: Circuit to display the I_{DS} vs. V_{GS} curve of a N-MOSFET transistor (in the saturation region) on the oscilloscope.

Exp. 7.2 Test the use of your MOSFET as a voltage-controlled resistance (in the Ohmic region) using the circuit shown in figure 7.11. Choose a drain resistor R_1 about the same as a typical R_{DS} of your MOSFET in the Ohmic region ($0 < V_{DS} < 0.5$ volt). This transistor is designed for medium power application so this resistance is very small, hence R_1 will also be small. v_{in} is a small AC signal between ground and about 0.5V. Test the circuit using sine waves. Is the MOSFET behaving as a linear resistor? If so then the shape of the waveform should be unchanged (do NOT use a square wave). Measure R_{DS} vs. V_{GS} and compare to the values you measured in the previous experiment. end

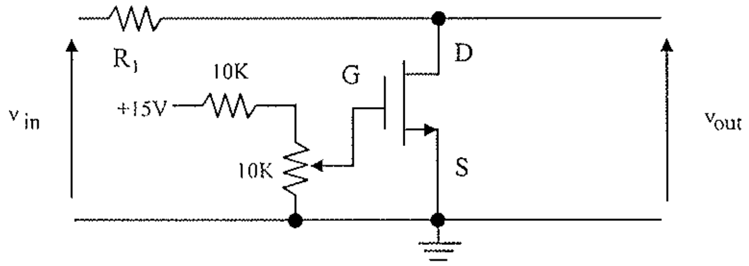


Figure 7.11: MOSFET Ohmic region experiment.

7.4 The MOSFET Amplifier

Generally speaking if you want a small signal amplifier you should use an op-amp. There are a vast number of different types of op-amps available now with a wide variety of capabilities and costs. It is only in the extreme cases of high frequency, high voltage, high power, etc. where you really have to design an amplifier using discrete transistors. However, it is worthwhile understanding how MOSFET's work as amplifiers to understand and use the op-amp's better. Understanding MOSFET's and how to use them will also better able you to understand new devices and circuits that will inevitably appear in the future.

7.4.1 One Transistor MOSFET Amplifier

The general idea to make an amplifier using a MOSFET is much the same as in a BJT amplifier. First the MOSFET must be biased into a region in which current normally flows (the active region in the BJT and the saturation region in a MOSFET), and then a small input signal is added to the bias voltage to wiggle the current about its quiescent or Q-point value. The MOSFET is more nonlinear than the BJT so this procedure is a little more difficult. A typical current-voltage transfer curve for a MOSFET in the saturation region is shown in figure 7.12. The drain-source current is quadratic in the gate-source voltage (in the saturation region). To make an amplifier the MOSFET is held near a Q-point in the middle of the curve using some external bias circuit and a small AC input voltage signal is applied at the gate. This produces a corresponding AC current variation in the drain current. If this signal is small enough then the curve is approximately linear over the range of the signal and the output is a faithful reproduction of the input signal. The slope of the curve near the Q-point is called the *transconductance* g_m of the MOSFET and will determine the overall gain of an amplifier.

$$g_m = \left.\frac{\partial I_{DS}}{\partial V_{GS}}\right|_{Q-pt} = 2K(V_{GS} - V_T)\Big|_{Q-pt} = 2\sqrt{KI_{D-Qpt}} \tag{7.9}$$

The value of g_m may vary with the position of the Q-point. The output signal is a current which may be turned back into a voltage using a resistor and Ohm's Law.

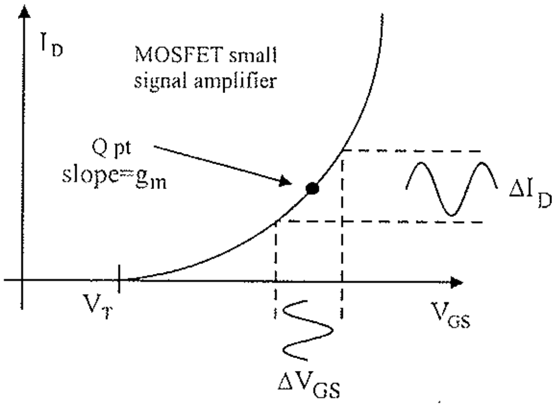


Figure 7.12: Current-voltage characteristics of a MOSFET in saturation near its Q-point.

The enhancement mode MOSFET can be biased in much the same way as the bipolar NPN and PNP transistors. Two possibilities are shown in figure 7.13. A single supply circuit (left figure 7.13) uses a voltage divider (R_1 and R_2) to offset the gate voltage to some positive value. The gate voltage is essentially zero so these two resistors can be very large (several megohms). The input impedance is the parallel combination of R_1 and R_2 and can be very large which is good for amplifying low current signals. The single supply circuit has AC coupling via capacitors C_{in} and C_{out}. Calculating the bias currents is a little more difficult than the bipolar circuit because of the quadratic dependence of drain current I_{DS} on gate voltage V_{GS}.

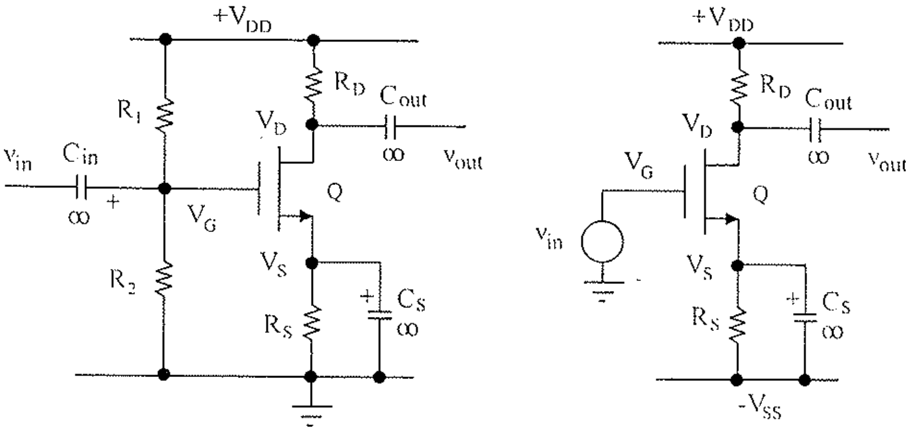


Figure 7.13: Some possible ways to bias a MOSFET amplifier using a single power supply (left) or a dual power supply (right).

A dual supply MOSFET bias circuit is shown on the right of figure 7.13. This allows the input signal v_{in} to be directly coupled for better low frequency response. The input impedance is still nearly infinite because v_{in} is driving an insulator (the gate). Calculating the bias currents and voltages is still complicated by the quadratic dependence of drain current I_{DS} on gate voltage V_{GS}.

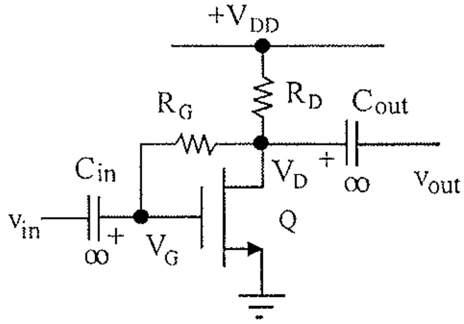


Figure 7.14: Simple MOSFET amplifier using a single power supply and self bias resistor R_G. C_{IN} and C_{OUT} allow the AC signal to pass without effecting the DC bias signal. Because the gate current is essentially zero, resistor R_G can be very large (1MΩ to 10MΩ).

Enhancement mode MOSFET's allow a simpler self-biasing circuit as shown in fig. 7.14. The resistor R_G connecting the drain to the source forms a self bias circuit. Because the gate is an insulator it draws no current and R_G may be very large. In practice the solderless breadboards you will be using have stray resistances of the order of several Megohms (2 to 20 MegOhms) which limits R_G to a maximum of about 1 MΩ but tens of Megohms are possible on a good printed circuit board. The final Q-point voltage V_D at the drain will reach an equilibrium value. If V_D is too high then the gate voltage will be large which causes the drain current I_D to be large. A large drain current produces a large voltage drop across the drain resistor R_D, which decreases the drain voltage. Conversely if the drain voltage is too small then the drain current will be small which increases the drain voltage. There is a single drain voltage that is an equilibrium point which is also the Q-point for this circuit.

First find the DC Q-point for the circuit in figure 7.14. In the saturation region (assumed) the drain current is (neglect the λ factor):

$$I_{DS} = K(V_{GS} - V_T)^2 \tag{7.10}$$

The voltage at the drain (and hence the gate, $V_{GS} = V_D$) is:

$$\begin{aligned} V_D &= V_{DD} - I_D R_D = V_{DD} - R_D K(V_{GS} - V_T)^2 \\ &= V_{DD} - R_D K(V_D - V_T)^2 \\ &= V_{DD} - R_D K(V_D^2 - 2V_D V_T + V_T^2) \end{aligned} \tag{7.11}$$

Grouping terms yields a quadratic equation for V_D:

$$R_D K V_D^2 + (1 - 2R_D K V_T)V_D + (R_D K V_T^2 - V_{DD}) = 0 \tag{7.12}$$

with a tedious solution of:

$$V_D = \frac{(2R_D K V_T - 1) \pm \sqrt{(1 - 2R_D K V_T)^2 - 4R_D K(R_D K V_T^2 - V_{DD})}}{2R_D K} \tag{7.13}$$

There are two solutions and one of them should not be possible. The other two bias circuits in figure 7.13 also lead to quadratic equations so there is no particular advantage to either of them in this regard.

The Q-pt. values can also be found graphically as in fig. 7.15. The MOSFET gives one equation or curve for I_D vs. V_{GS} (eq. 7.7) labelled NMOS in the graph. With the transistor's source grounded $V_{DS} = V_D$ and I_D is also determined by Ohm's law in the drain resistor R_D as $I_D = (V_{DD} - V_D)/R_D$ (dashed line in fig. 7.15). The maximum current of $I_{MAX} = V_{DD}/R_D$ occurs when V_D=0 and the maximum voltage of $V_D = V_{DD}$ occurs when I_D=0. These two points are connected with a straight line as shown. The drain current I_D must satisfy both equation so the only possible value is the intersection of these two curves which is the Q-pt. You can also find the Q-pt by constructing this graph using the data you measured from the above experiment on the current-voltage curves for this MOSFET.

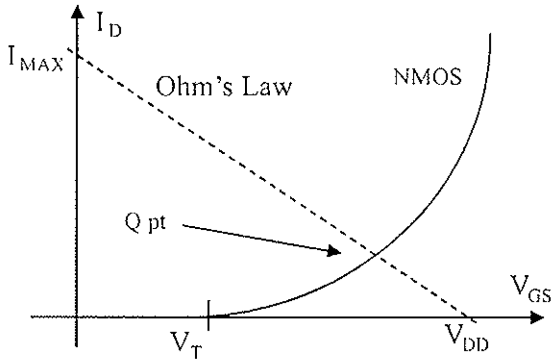


Figure 7.15: Graphical solution of the NMOS Q-point for the one transistor NMOS amplifier in fig. 7.14.

Once the Q-pt has been determined the actual small signal AC gain can be calculated from the transconductance.

$$\begin{aligned}
\Delta I_D &\sim \frac{\partial I_D}{\partial_{GS}}\Delta V_{GS} = g_m \Delta V_{GS} = g_m \Delta V_{in} && (7.14)\\
\Delta V_{OUT} &= \Delta(V_{DD} - I_D R_D) = -\Delta I_D R_D && (7.15)\\
G_{AC} &= \frac{\Delta V_{OUT}}{\Delta V_{IN}} = -g_m R_D = -2K(V_{GS}|_Q - V_T)R_D && (7.16)
\end{aligned}$$

where $V_{GS}|_Q$ is the DC Q-pt voltage between the gate and the source. At first glance it appears that the AC voltage at the gate should be the same as the AC voltage at the drain as is true for the DC voltages. The negative sign in the gain would then produce a large negative feedback and produce nearly zero AC gain. However the low pass filter formed by R_G and C_{IN} through the input voltage source effectively prevents the AC component of the drain voltage from reaching the gate while allowing the DC component to pass, so there is a significant AC gain.

$\boxed{\text{Exp. 7.3}}$ Construct a simple one transistor amplifier using a 2N7000 MOSFET as in fig. 7.14. If K is relatively large then the Q-pt. voltage at the drain will be only slightly larger than V_T (typically 2.1 V), so there is no point in making the supply voltage much larger than this if the output is to have a symmetrical swing positive and negative. Therefore choose a supply voltage of V_{DD}= +5V. Select $R_G \sim$ 1MΩ to get a large input impedance as is desirable for a small signal voltage amplifier. For R_D= 270Ω select values for the input and output capacitors such that they look like short circuits for frequencies near 1 kHz. Do NOT use an electrolytic capacitor for the input because they have too much leakage current and may disturb the gate bias circuit.

First measure the Q-pt voltage at the drain with no input signal applied and compare to what you calculate. Next apply a small input signal using a sine wave near 1 kHz and measure the small signal AC gain for this amplifier and compare to what you calculate (be careful that the output sine wave is not clipped or distorted).

OPTIONAL: Vary the drain resistor R_D (in the range 100 to 1K) and see what happens to the gain and the Q-pt voltage at the drain. $\boxed{\text{end}}$

If the MOSFET parameter K is large (as for this transistor) and the drain resistor R_D is not too small then a simple approximate solution exits. In this situation the gate voltage is only slightly above V_T so approximate $V_G \sim V_D \sim V_T$ then:

$$I_D = \frac{V_{DD} - V_D}{R_D} \sim \frac{V_{DD} - V_T}{R_D} \tag{7.17}$$

$$g_m = \frac{\partial I_D}{\partial V_{GS}} = 2K(V_{GS} - V_T) = 2\sqrt{KI_D} \sim 2\sqrt{\frac{K(V_{DD} - V_T)}{R_D}} \tag{7.18}$$

$$G_{AC} = -g_m R_D = -2R_D\sqrt{\frac{K(V_{DD} - V_D)}{R_D}} = -2\sqrt{KR_D(V_{DD} - V_D)} \tag{7.19}$$

$$\sim -2\sqrt{KR_D(V_{DD} - V_T)} \tag{7.20}$$

Because K is not dimensionless (units of mA/volt^2), it should be compared to typical voltages (V_{DD} max.) and currents (V_{DD}/R_D max.) in the circuit to determine whether or not it is "large".

7.4.2 Two Transistor Differential Amplifier

A two transistor differential amplifier can be built from MOSFET's in a manner similar to that of the BJT. Two examples are shown in fig. 7.16. The circuit on the left has a current source in the bottom which can be built from other transistors. The circuit on the right has a resistor that may approximate a current source if R_S and V_{SS} are both large. It is easier to build but some of its properties are not quit as good as an actual current source.

For the differential pair to work properly both transistors must be matched ($Q_1 = Q_2$) as in the BJT version. This means that:

$$\begin{aligned} K &= K_1 = K_2 \\ V_T &= V_{T1} = V_{T2} \end{aligned} \tag{7.21}$$

First analyse the circuit assuming a real current source in the bottom (left of fig. 7.16). If the transistors are matched then the drain currents for both transistors at the Q-point ($V_1 = V_2 = 0$) must satisfy:

$$\begin{aligned} I_{D1} + I_{D2} &= I_0 = \quad \text{constant} \\ I_{D1} = I_{D2} &= K(-V_S - V_T)^2 = \frac{1}{2}I_0 \end{aligned} \tag{7.22}$$

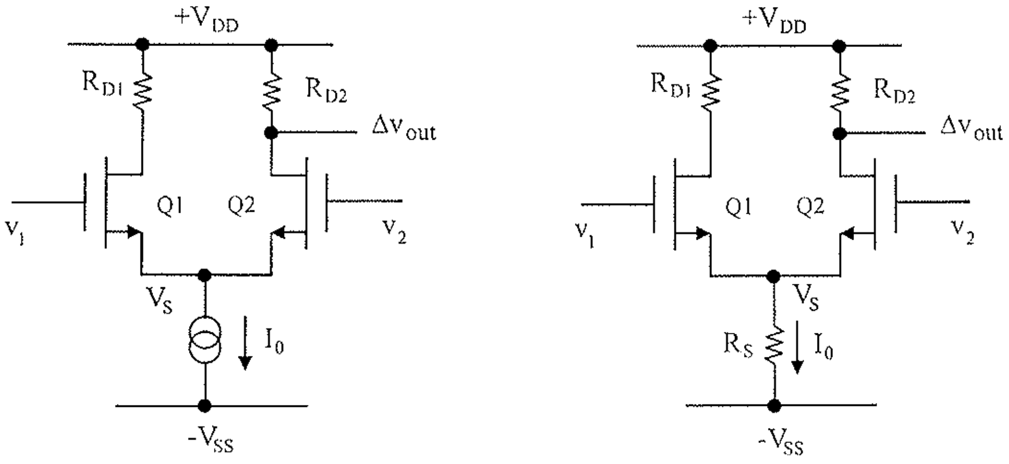


Figure 7.16: MOSFET differential amplifier. v_1 and v_2 are the small signal differential input voltages and Δv_{out} is the output. Both transistors are assumed to be matched ($Q_1 = Q_2$). A current source in the bottom (left) provides better performance but a simple resistor can be substituted (right) in some cases.

This yields a value for the Q-point voltage at the drains of:

$$\begin{aligned} V_{D1} &= V_{DD} - I_{D1}R_{D1} = V_{DD} - \frac{1}{2}I_0 R_{D1} \\ V_{D2} &= V_{DD} - I_{D2}R_{D2} = V_{DD} - \frac{1}{2}I_0 R_{D2} \end{aligned} \tag{7.23}$$

Now solve for the effects of a small incremental voltage change at the input of ΔV_1 and ΔV_2 near the Q-pt. ($V_1 = V_2 = 0$). The corresponding small incremental change in drain current is:

$$\begin{aligned} \Delta I_{D1} &\sim \frac{\partial I_{D1}}{\partial V_{GS1}}\Delta V_{GS1} = g_m \Delta V_{GS1} = g_m \Delta(V_1 - V_S) \\ \Delta I_{D2} &\sim \frac{\partial I_{D2}}{\partial V_{GS2}}\Delta V_{GS2} = g_m \Delta V_{GS2} = g_m \Delta(V_2 - V_S) \end{aligned} \tag{7.24}$$

where the transconductance of both transistors is the same:

$$g_m = g_{m1} = g_{m2} = 2\sqrt{KI_0/2} = \sqrt{2KI_0} \tag{7.25}$$

because the transistors are assumed to be matched and the input control voltages (Q-pt. values of V_{GS1} and V_{GS2}) are identical. Subtract these two equations (7.24) to obtain:

$$\Delta I_{D1} - \Delta I_{D2} = g_m \Delta(V_1 - V_2) = g_m \Delta V_{DIF} \tag{7.26}$$

where $V_{DIF} = V_1 - V_2$ is the differential input voltage. From equation 7.22:

$$\Delta I_{D1} + \Delta I_{D2} = \Delta I_0 = 0 \tag{7.27}$$

$$\Delta I_{D1} = -\Delta I_{D2} \tag{7.28}$$

which produces:

$$\begin{aligned}\Delta I_{D1} &= \frac{1}{2} g_m \Delta V_{DIF} \\ \Delta I_{D2} &= -\frac{1}{2} g_m \Delta V_{DIF}\end{aligned} \tag{7.29}$$

The drain resistors turn this current back into a voltage via Ohm's law:

$$\begin{aligned}\Delta V_{D1} &= \Delta(V_{DD} - I_{D1}R_{D1}) = -\Delta I_{D1}R_{D1} = -\frac{1}{2} g_m R_{D1} \Delta V_{DIF} \\ \Delta V_{D2} &= \Delta(V_{DD} - I_{D2}R_{D2}) = -\Delta I_{D2}R_{D2} = \frac{1}{2} g_m R_{D2} \Delta V_{DIF}\end{aligned} \tag{7.30}$$

The net voltage gain is:

$$\frac{\Delta V_{D2}}{\Delta V_{DIF}} = \frac{\Delta V_{OUT}}{\Delta V_{DIF}} = \frac{1}{2} g_m R_{D2} \tag{7.31}$$

This output signal could be capacitively coupled or applied to later stages of an op-amp.

If a resistor is substituted for the current source then there is another quadratic equation that must be solved. However, if K and V_{SS} are large then there is a simple approximation as in the previous section. With $V_1 = V_2 = 0$ the source voltage V_S will be slightly more than V_T below the gate. V_S may vary by probably less than one volt for a large K which will not change I_0 very much if $|V_{SS}| >>1$ V leaving:

$$I_0 \sim \frac{-V_T - (-V_{SS})}{R_S} = \frac{V_{SS} - V_T}{R_S} \tag{7.32}$$

> $\boxed{\text{Exp. 7.4}}$ Construct a simple two transistor differential amplifier using two 2N7000 MOSFET's and three resistors as shown on the right of fig. 7.16. Use supply voltages of ±15 V, and resistors of R_S = 1K, and $R_{D1} = R_{D2}$ = 1K. MOSFET parameters can vary significantly from one to another. It may be necessary to find a matched pair by trial and error. Try to get the DC voltages at the drain to be within a volt of two of each other. Measure the Q-pt. voltages (at source and drains) and the gain of this amplifier. Compare to the theoretically calculated values. Observe the phase of the signal at both drains relative to the input signal. The function generator does not easily produce a differential signal so ground one input and apply a sine wave signal to the other input (to measure the gain). Be sure to keep the input small enough so that the amplifier does not clip the output.
>
> OPTIONAL: Try replacing the resistor with a current source as in practice problem 2. Its easier to start with a potentiometer in place of R_3 and R_4 with V_{GS} slightly above V_T, and adjust to get a good Q-pt. at the drains. $\boxed{\text{end}}$

This MOSFET differential amplifier has a very high input impedance because the input signals are driving an insulator. High input impedance is very useful if you are trying to amplify very small, low current signals as is frequently required. The 3140 op-amp input is in fact two MOSFETs connected as a differential amplifier which accounts for the extremely low input bias current of the 3140. The output of this differential stage usually drives further amplifier stages in an op-amp which is then used in a negative feedback configuration which also serves to correct the nonlinearities of the transistors.

MOSFET	V_{DS} (volts)	I_{DS} (Amps)	on-R_{DS} (Ohms)	Δt_{on} (nSec)
n-channel				
2N7000	60	0.2	5.0	10
BS170	60	0.5	5.0	10
FQA170N06	60	170.0	0.0056	180
IRF510	100	5.6	0.54	7+
IRF540	100	28.0	0.077	11+
IRF640	200	18.0	0.18	14+
IRF840	500	8.0	0.85	14+
FQA13N80	800	12.6	0.75	130
p-channel				
BSS84	50	0.13	10	
IRF9520	100	6.8	0.60	10+
IRF9640	200	11.0	0.50	14+
FQP1P50	500	1.5	10.5	30

Table 7.1: Maximum ratings of some MOSFET's. (Delay times with a '+' are typical not max. because max. time not specified.)

7.5 Power MOSFETS

One very interesting and useful aspect of MOSFET technology is the ability to control very large currents and voltages with relatively small control voltages in the class of devices called *power MOSFETs*. A short list of a few different high and medium power MOSFET's is given in table 7.1. Generally speaking there are many more type of n-channel devices than p-channel devices. Possible uses for power MOSFET's include control of motors, lamps and other high current or voltage devices.

7.6 Digital Signals

Huge numbers of MOSFET's are used in digital integrated circuits (IC's) such as microprocessors and memory IC's (100's of millions of transistors may be on one single, albeit large, IC). It is worth a little effort here to think about how MOSFET's can be used for digital signals.

Digital signals and logic circuits will be discussed in detail in the next chapter. However it suffices, for the moment, to say that digital signals have only two discrete levels called a logic 0 and a logic 1 (or equivalently FALSE and TRUE). Each level will be represented as a high voltage and a low voltage, for example logic 0 will be near 0 volts and logic 1 may be near 5 volts. The voltage does not have to be exactly these values but only within a certain tolerance of these levels (maybe with 0.5 to 1.0 volts of these levels). A square wave is a good example of a digital signal, having two levels, high and low. There are a variety of new and useful things that can be done with digital signals which will be investigated in the next chapter.

The nonlinearity of the MOSFET causes trouble when used as a linear amplifier (as in the previous experiments). There is always this funny quadratic equation that must be solved. However if the device is used only for digital signals this difficulty essentially goes away. The device is either ON or OFF with nothing in between. As long as these two levels are clearly defined and separated then there is no concern about what happens in between these two levels (as long as there are no singularities). This probably accounts for the MOSFET's popularity as a digital device (it is also better for low power CMOS devices as in the next chapter).

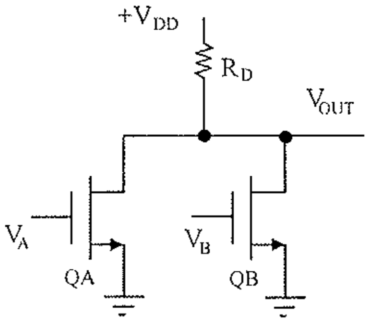


Figure 7.17: Simple digital logic gate using two n-channel MOSFETs.

The circuit shown in fig. 7.17 is a simple digital logic gate. A gate is the basic build block of digital electronics analogous to the op-amp in analog electronics. There are two digital inputs V_A and V_B that can be either high or low. Each of the two MOSFET's is used as a switch. If the input (V_A or V_B) is significantly higher than V_T then the MOSFET is said to be "ON" and the transistor looks like a short circuit to ground. If the input is lower than V_T then the transistor is "OFF" and the transistor looks like an open circuit. The two MOSFET's are connected to combine the two input digital signals V_A and V_B is a logically significant way. Investigate the function of this gate in the next experiment.

Exp. 7.5 Construct the simple two transistor digital logic gate using two 2N7000 MOSFET's as shown in fig. 7.17. Use a supply voltage of $V_{DD} = +5$ V, and $R_D = 1$K.

First observe the switching response of one MOSFET. Connect V_B to ground and drive V_A with a square wave (0V to 5V) from the function generator. Observe V_A and V_{OUT} on the scope and sketch both. Record the voltage levels of V_{OUT} and its rise and fall time. If you look on a small enough time scale you should be able to see the small switching time of these transistors.

Next record the logic function of this gate. Each input (V_A and V_B) can have two possible values, 0V or +5V, giving four possible input conditions. Observe the output V_{OUT} in volts for each of these four possible state and record the results in table form as in table 7.2. Translate these voltage levels into digital logic levels (0/1).

OPTIONAL: Record the logic function with the transistors in series (as in fig. 7.22) instead of parallel (as in fig. 7.17). end

V_A	V_B	V_{OUT}
0V	0V	
0V	+5V	
+5V	0V	
+5V	+5V	

Table 7.2: Logic table for the gate in fig. 7.17 to be filled in.

7.7 JFET's and Depletion Mode MOSFET's

There are two other related types of field effect transistor, the depletion mode MOSFET and the junction field effect transistor or JFET. The n-channel deletion mode MOSFET is shown in fig. 7.18. It is very similar to the enhancement mode MOSFET shown in fig. 7.1, except that there is normally an n-channel region connecting the drain and source even when there is zero voltage applied to the gate. Current normally flows between the drain and source with zero volts on the gate and when a negative voltage is applied to the gate the current is turned off or depleted. The circuit symbol on the right of the figure is also similar to that of the enhancement mode device except the line connecting the drain and source is a solid line rather than a dashed line and the simplified symbol is draw with a thick or double line connecting the drain and source. Again only the n-channel version will be discussed the p-channel version works in a very similar manner except all of the polarized quantities switch sign.

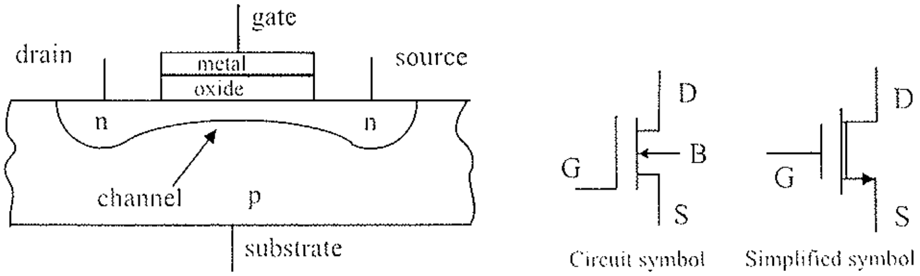


Figure 7.18: Depletion mode n-channel MOSFETs.

An n-channel JFET is shown in figure 7.19. The source and drain are connected with a continuous channel of n-type material. Current normally flows between the drain and source with zero voltage on the gate much like the depletion mode MOSFET. The gate is directly connected to the channel between the source and drain and forms a pn junction which is normally reversed biased. The depletion region formed at this reverse biased junction decreases the size of the conduction channel between the drain and source. Increasing the reverse bias on this junction increases the size of the depletion region and decreases the drain source current. Because the gate is reversed biased there is essentially no current flowing into the gate so the JFET input impedance is also very high just like a MOSFET. However there is still a small reverse saturation current so the JFET input current is slightly higher than that of the MOSFET (slightly lower input impedance). In recent years the MOSFET seems to be replacing the JFET in most applications.

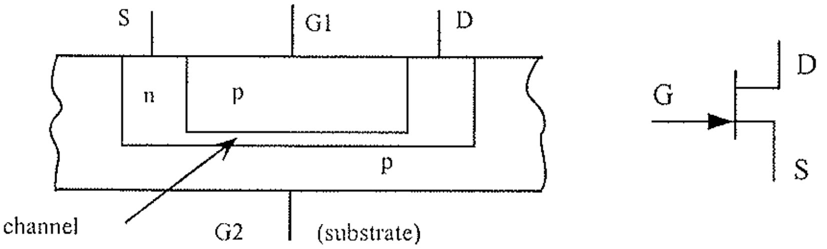


Figure 7.19: N-channel junction field effect transistor (JFET). The gate connections G1 and G2 are assumed to be connected together

Both the JFET and the depletion mode MOSFET have very similar current-voltage characteristics. A plot of the drain current versus gate-source voltage is shown in fig. 7.20 and compared to the enhancement mode MOSFET. The primary difference can be summarized by saying the the threshold voltage has an opposite sign. The I_D vs. V_{DS} curves for both the depletion mode MOSFET and the JFET are nearly identical to fig. 7.5 for the enhancement mode MOSFET except the values of V_{GS} change (the equations are also very similar). The JFET is sometimes a better choice for a voltage controlled resistor because it has a much higher resistance than a discrete MOSFET used above.

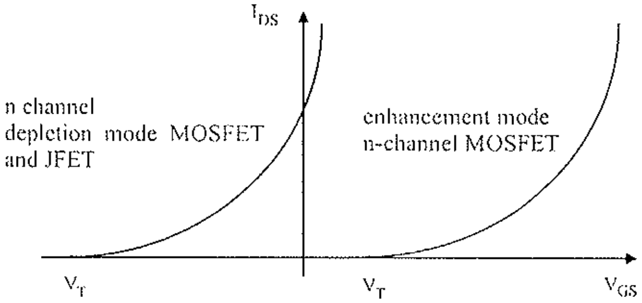


Figure 7.20: Depletion mode n-channel MOSFET and JFET current voltage relationship compared to that of an enhancement mode n-channel MOSFET.

7.8 Practice Problems

[1] If the single MOSFET amplifier shown in figure 7.14 is connected with a power supply voltage of +5V, R_G= 1MΩ and R_D= 330Ω find the following. Assume MOSFET parameters K= 100 mA/V^2, V_T= 2.0 V.

a) The Q-pt voltage at the drain (in volts).

b) The small signal AC gain.

[2] The differential amplifier shown in fig. 7.21 uses a transistor as a current source in the bottom of the circuit. Assume that V_T = 2V, K= 100 mA/V^2 and power supply voltages of ±5V.

a) Find R_3 and R_4 to make I_0 = 10mA.

b) Find R_{D1} and R_{D2} to make the Q-pt. at the drain of $Q1$ and $Q2$ be 2.5V using I_0 = 10mA.

c) What is the small signal AC gain $V_{out}/(V_1 - V2)$ if C_{out} is sufficiently large?

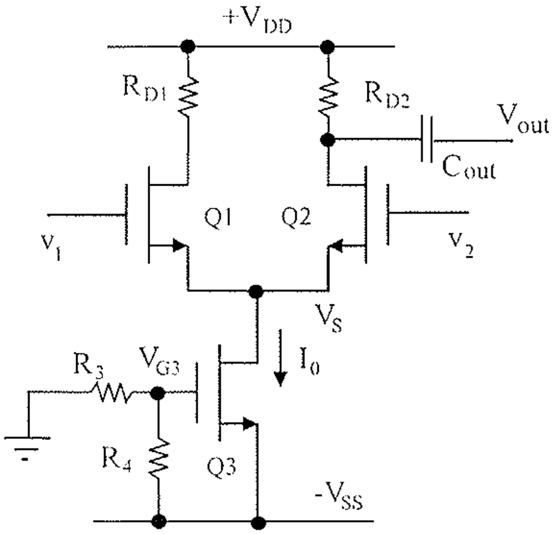


Figure 7.21: Problem 2.

[3] The circuit shown in figure 7.22 uses two enhancement mode MOSFET's to perform a Boolean logic function. You may assume K= 100mA/V^2 and V_T= 2.0V, and R_D= 2.7K. V_{DD}=+5V Complete table 7.2 for this circuit by filling in the missing lines. State whether each transistor is ON, OFF or undetermined and give a value for V_{OUT} (in volts).

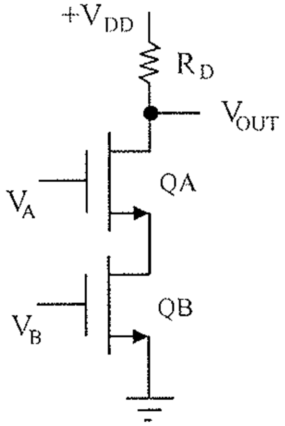


Figure 7.22: Problem 3.

Chapter 8

DIGITAL ELECTRONICS AND LOGIC GATES

8.1 Digital Signals

Information can be encoded into electrical signals as *quantized* or discrete levels. These kinds of signals are referred to as *digital signals.* Usually there are just two distinguishable levels, which are called HI and LO, (or 1 and 0) respectively. For example a HI value can be defined as a certain range of voltages and a LO value as another range of voltages that does not overlap with the HI range as in figure 8.1 In principle it is also possible to define a digital logic system with more than two levels, but such multi-level systems have not found much practical use so far.

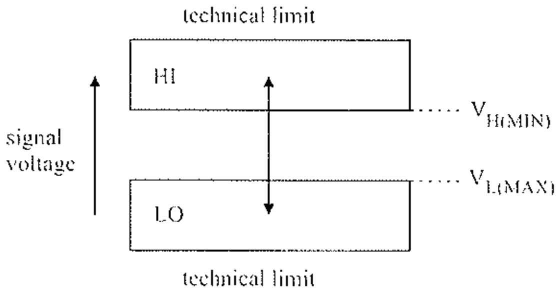


Figure 8.1: Digital signals.

To define the ranges over which the signal level can vary and still be uniquely identified as being either HI or LO, the two "inner" limits $V_{H(MIN)}$ and $V_{L(MAX)}$ are specified (see figure 8.1). Any signal falling into the dead band between these limits is ambiguous and should not appear in a well-designed circuit. The "technical" limits represent the maximum signal swing that can be delivered and accepted without damage by the circuit in use. These outer limits are not usually of great interest. All of the action takes place as the signal moves from within one zone to the other. One particular advantage of digital signals is that they are approximately insensitive to small amounts of noise. As long as the voltage is within the allowed *range* of HI or LO it can be interpreted as exactly a 1 or 0 respectively.

This is the reason that digital recording and communication is dramatically better than the old style analog versions.

Such two-level signals are called *binary*, (i.e. base 2 arithmetic) and they can be used to represent binary information. For example, ordinary logic has two "logical values". A statement (if meaningful) is either TRUE or FALSE. Similarly, arithmetic to base 2 has digits which can have only two values; 1 or 0. Logical or binary-arithmetic information can be processed with digital signals. The appropriate circuits are called *logic* or *digital circuits.*

There is a choice of how to associate signal levels with logic states. In positive logic, the HI state represents 1. In negative logic, the LO state represents 1. Note that HI and LO refer only to the relative signal levels. One or both of $V_{H(MIN)}$ and $V_{L(MAX)}$ could be negative voltages. Digital signals can also be encodes as current levels. The discussion in this chapter will stick to positive logic (with voltage levels) without loss of generality.

Logic operations can be described by *Boolean algebra*, in which the attribute TRUE is represented by 1, and FALSE by 0. The next few sections give a brief introduction to Boolean algebra, this notation and some of its uses. For a more detailed discussion of digital systems see, for example, Wakerly[19].

8.2 Logic Operations

8.2.1 Negation or Inversion $X = \overline{A}$

A	X
0	1
1	0

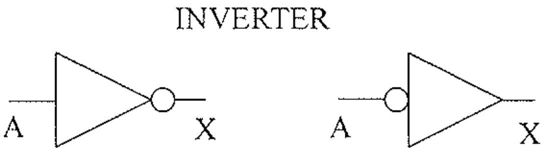


Figure 8.2: NOT gate or inverter (on right) with truth table (on left). $X = \overline{A}$

This operation generates a result which is the *complement* of the input variable. This symbol is read "not-A" or "A-bar". A is the input and X is the output. The *truth table* for a logic operation is a list of results for all possible inputs (shown in figure 8.2 on the left). The NOT operation has only a single input, so the table has only two lines.

A circuit which performs the NOT operation is called an *inverter*. In the symbol shown in figure 8.2, the inversion is indicated by the "bubble". The triangle stands for a buffer, which merely reproduces the input signal and provides it with more power for driving subsequent loads. The bubble could just as well have been shown at the input of the buffer. Two cascaded inverters yield a simple buffer: $\overline{(\overline{A})} = A$.

The notation $\overline{A}$ provides us with a clear way of identifying *active-low* signals in the positive-logic convention. The signal itself may indicate some condition to be TRUE when the LO state is asserted. Suppose a computer keeps a certain terminal at +5 volts while it is busy performing a calculation. Upon completion of the task it takes this terminal down to ground potential, signifying it is now ready for the next instruction. The signal on this "ready" terminal is in negative logic (it is *active-low*), but the positive-logic convention can be maintained by calling the signal $\overline{READY}$. It is true that the computer is not-ready when the signal is HI.

8.2.2 AND $C = A \cdot B = AB$

The result of this operation is 1 only if both inputs are 1. The "·" is read "and", and can sometimes be left out if the meaning is clear. A circuit performing the AND function is called an AND gate, as shown in figure 8.3 (A,B are inputs and C is the output).

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

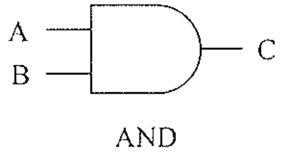


Figure 8.3: AND gate with truth table. $C = A \cdot B = AB$

These logic operations are often needed in implementing control systems. For example, "if the room thermostat calls for heat AND the pilot flame is known to be lit, turn on the gas to the furnace." The AND function can be extended to an arbitrary number of inputs. The result is 1 only if all the inputs are 1.

8.2.3 OR (Inclusive OR) $C = A + B$

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

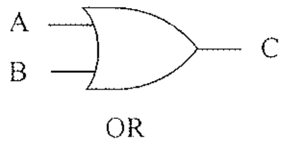


Figure 8.4: OR gate with truth table. $C = A + B$

Here the result is 1 if A or B (or both) are 1. The "+" is read "or". For example, "if the brake pedal is pushed OR the hand brake is pulled apply the rear brakes." The OR function, too, can be extended to an arbitrary number of inputs. The result is 1 if any of the inputs is 1. The circuit symbol for an OR gate is distinguished by its curved input edge (see figure 8.4) and pointed output edge.

It is possible to synthesize any logic function by suitably combining the NOT operation with the AND and OR functions. However, such a basic approach is cumbersome. Elementary gates are usually supplied in multiple groups of independent circuits on a single chip. Inverters typically come in groups of six ("hex inverters") and two input gates typically come in groups of four ("quad 2-input OR gates", etc.). If only one inverter is needed then it is inefficient. The NAND and NOR functions given below can perform all functions with just one type of gate which can be more efficient but a little harder to understand. Practical families of logic circuits contain many different functions, in some cases extending to complete instruments-on-a-chip.

8.2.4 NAND, NOR

Several technologies for logic gates lend themselves naturally to inverting the signal along with producing the desired operation. An AND gate whose output is inverted is called a NAND gate (NOT-AND). The bubble at the *output* of the circuit symbol signifies that the AND is formed *first*, and then the result is negated. The order makes a big difference. Similarly, the NOR function stands for NOT-OR, with NOT last. Both are shown below (A,B are inputs and C is the output).

A	B	C(NAND)	C(NOR
0	0	1	1
0	1	1	0
1	0	1	0
1	1	0	0

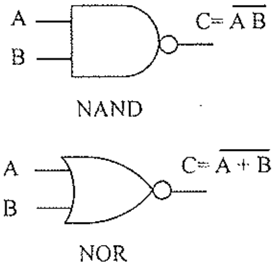


Figure 8.5: NAND and NOR gate with truth table.

Because gates often come in multiples within a package, it is sometimes convenient to use a leftover gate of one type to perform another function, rather than put in a separate package. Multiple-input gates can have their *width* reduced by tying some inputs together, or by fixing some of them appropriately at HI or LO potential. NAND and NOR gates can readily be used as inverters by this method. Some examples with two-input gates are shown in figure 8.6.

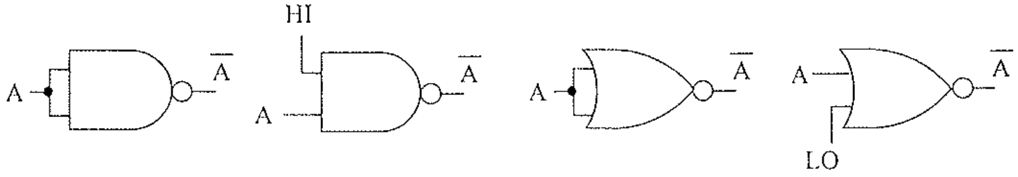


Figure 8.6: Making a single NOT gate from a NAND or NOR gate.

8.2.5 Exclusive-OR (XOR) $C = A \oplus B$

A slightly higher-level function is the exclusive-OR, which resembles OR except that it excludes the case when both inputs are 1. The truth table and the Boolean circuit symbol are shown in figure 8.7. Note that the complement of exclusive-OR is the equality function, which generates a 1 only when the two inputs are the same.

From the truth table (table 8.7) for the XOR, the Boolean expression for the XOR is:

$$C = A \oplus B = A \cdot \overline{B} + \overline{A} \cdot B \tag{8.1}$$

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

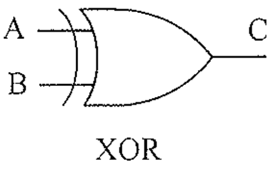


Figure 8.7: XOR gate with truth table. $C = A \oplus B$

The XOR gate is not really a fundamental gate because it can be made by connecting other gates together. There are many possible ways to make an XOR gate, and the most straight forward (although not the most efficient) method is shown in figure 8.8.

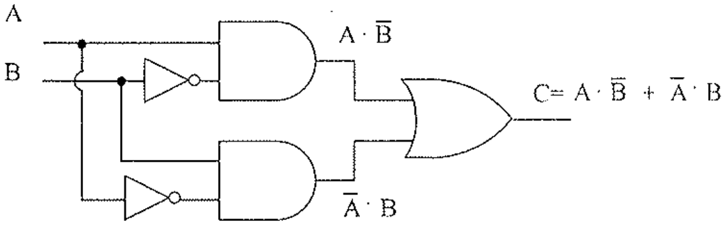


Figure 8.8: XOR gate made out of other gates.

8.2.6 Multiple Inputs

The number of inputs on the two input gates OR, AND, NAND, NOR can be expanded to more than two in the obvious manner. Two examples are shown in figure 8.9.

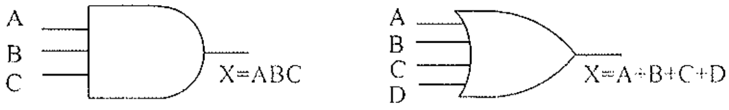


Figure 8.9: Multiple input gates.

8.3 CMOS Logic

Standardization of input and output characteristics is relatively easy to achieve when digital signals are used, and it is thus possible to design whole families of devices which can be easily interconnected. Several such logic families have evolved. This chapter will experiment with 4000 series CMOS logic (which uses Complementary MOSFET transistors), and the next chapter will experiment with TTL (bipolar transistor-transistor logic). Skipping the finer points for the moment, you can start using 4000 series CMOS logic on the basis of the following information:

1. The *supply voltage* is flexible (3 to 15 V). The Digi-Designer breadboards provide supply rails at ground and +5 V.

2. *Inputs* are effectively open circuits. The HI and LO states are symmetrically placed between the supply-rail potentials, with minor sample-to-sample variations. The input voltage should not go outside the supply-rail span. Because of the extremely high input resistance, no input must be left "floating". *All unused inputs should be tied to 0 or +5 V.*

3. *Outputs* swing very close to the supply rails in the two states. They can deliver about one mA in either direction without leaving the specified HI or LO zones.

Exp. 8.1 Measure the truth table of one of the AND gates in a CD4081 quadruple 2-input AND-gate circuit as shown in figure 8.10. The two inputs can come from the logic-level switches on the board. The state of the output can be monitored by connecting it to one of the logic-level LED indicators (they use op-amp comparators and thus have high input impedance).

The 4081 contains four independent 2-input AND gates. CMOS data books are available in each lab. You will need to use them to find the pin-out of each IC. Wire all unused inputs to ground or +5V (as convenient), but do not connect anything to unused outputs. The supply rail should be bypassed to ground, as usual.

Next move input 2 to the CLOCK source on the Digi-Designer, leaving input 1 on the switch. With the scope observe the waveforms for the clock and for the output, with the switch both high and low.

Finally, substitute a CD4071 quad 2-input OR gate (the pin connections are the same) and repeat the previous observations.

[OPTIONAL:] With the clock on one input and the other input tied high or low (as required to allow the output to follow the clock) measure the propagation delay from the input to the output on one or both of these gates (note that the high to low and low to high propagation delays may be slightly different). end

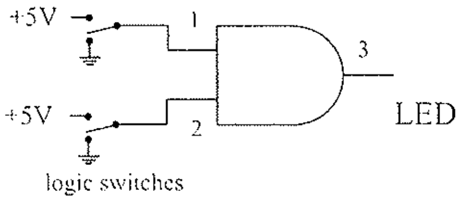


Figure 8.10: Measure the truth table for an AND gate.

As soon as your circuits start to contain several gates, it becomes important to be systematic in bookkeeping and drawing circuit diagrams to avoid confusion. Copy the pin numbers from the data sheets onto your diagrams and identify the packages (if there are several).

8.4 Logic Combinations

Surprisingly complicated logic combinations can be required in control systems or in numerical computation. There exists an extensive body of knowledge which formalizes the procedures for combining simple gates to achieve the desired functions with minimum hardware. Although this can be a difficult task, frequently someone has already designed what you need and it is available in integrated circuit form. The user is normally faced with the following possibilities:

(a) The required function is sufficiently "standard" that a complete package for it has already been marketed.

(b) The function is relatively simple and can be easily implemented, possibly at the risk of putting in one or two more basic gates than are strictly necessary.

(c) The function is complicated and non-standard. Some design effort may be required, or a more sophisticated device may be programmed to perform the function.

Frequently in a research environment the cost of the logic devices is not a significant factor in the construction of the circuits. More important considerations include the *design effort* and the difficulty of trouble-shooting ("*debugging*") the circuits when they are first built or later in their service life. The degree of *flexibility* of the circuit to permit functional changes to be produced at some later time (without complete reconstruction of the circuit) is also important. The primary goal in an industrial setting is frequently the opposite. You need to make the circuit in the most economically efficient manner, which usually translates into the smallest number of components.

Where complicated functions are needed, you would first look for a ready-made implementation. However, there are always situations in which a certain amount of logic manipulation needs to be done on a piecemeal basis. In many cases the required combinations can be discovered intuitively once the functions are clearly described in English. However, a slightly more systematic approach using Boolean algebra is worth exploring.

Most of the basic theorems of Boolean algebra seem readily apparent from common sense intuition. You should make sure that you can *verbalize* appropriate situations to which the following identities apply. A short list of basic Boolean identities is given here.

$$0 \cdot 0 = 0 \qquad 0 + \dot{0} = 0 \tag{8.2}$$

$$0 \cdot 1 = 0 \qquad 0 + 1 = 1 \tag{8.3}$$

$$1 \cdot 1 = 1 \qquad 1 + 1 = 1 \tag{8.4}$$

$$A \cdot \overline{A} = 0 \qquad A + \overline{A} = 1 \tag{8.5}$$

$$A \cdot 0 = 0 \qquad A + 0 = A \tag{8.6}$$

$$A \cdot 1 = A \qquad A + 1 = 1 \tag{8.7}$$

$$A \cdot A = A \qquad A + A = A \tag{8.8}$$

$$\begin{aligned}
A \cdot B &= B \cdot A = AB = BA && (8.9)\\
A + B &= B + A && (8.10)\\
A(BC) &= (AB)C && (8.11)\\
A + (B + C) &= (A + B) + C && (8.12)\\
A \cdot (A + B) &= A && (8.13)\\
A + (A \cdot B) &= A && (8.14)\\
A \cdot (B + C) &= AB + AC && (8.15)\\
(A + B)(C + D) &= AC + AD + BC + BD && (8.16)
\end{aligned}$$

The first step in implementing a logic function is to write done a Boolean expression for the function. This may be done from a truth table listing all possible states of the inputs and the resulting output for each or sometimes from a careful English description of the process. An example of the later follows. The output D of the circuit must be HIGH (or TRUE) when either C is FALSE or A is TRUE and B is not TRUE (i.e. FALSE). The Boolean expression for D is:

$$D = A \cdot \overline{B} + \overline{C} \qquad (8.17)$$

Now draw a schematic of the required circuit given A,B, and C as input. First form A and not-B. Because not-B is not available, first invert B and then AND it with A. Next OR this result with the inverse of C using an OR gate. Figure 8.11 shows this function using only NAND gates and DeMorgan's theorm for the OR function with an inversion. Combining gates in this manner is referred to as *combinatorial logic* as opposed to *sequential logic* as will be discussed in the next chapter on flip-flops.

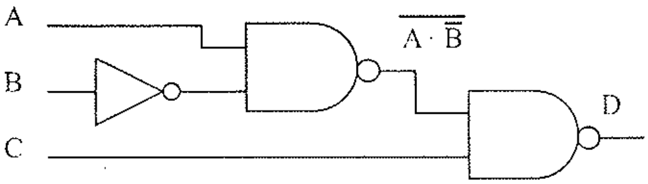


Figure 8.11: Logic circuit to test.

Exp. 8.2 Assemble the circuit shown in 8.11 using a 4011 NAND gate package. Measure its truth table (D as a function of A,B and C) and compare to what you predict from the Boolean algebra expression. It is easier to make the inverter from a NAND gate because then you only need one IC (less wiring). end

8.5 DeMorgan's Theorem

One Boolean theorem that is not so obvious is DeMorgan's Theorem. There are two common forms:

$$\overline{(A \cdot B)} = \overline{A} + \overline{B} \quad \text{and} \quad \overline{(A + B)} = \overline{A} \cdot \overline{B} \qquad (8.18)$$

When you "merge" the bar over two variables in an AND/OR combination, you must also interchange AND and OR. You can verbalize this theorem by noting that if unanimous consent is required to take some action (action = $A \cdot B \cdot C \cdots$), then a single "nay" vote suffices to veto the action (*action vetoed* = $\overline{A \cdot B \cdot C \cdots} = \overline{A} + \overline{B} + \overline{C} + \cdots$).

Using circuit symbols, DeMorgan's theorem says that *if you place "circles" on all inputs and outputs of a gate, you must interchange AND and OR at the same time.* The circles can later be mentally "slid" along the lines to nullify each other in pairs. All of the gates discussed above also have another representation due to DeMorgan's Theorem. The four examples shown in figure 8.12 are symbolic representations of DeMorgan's Theorem.

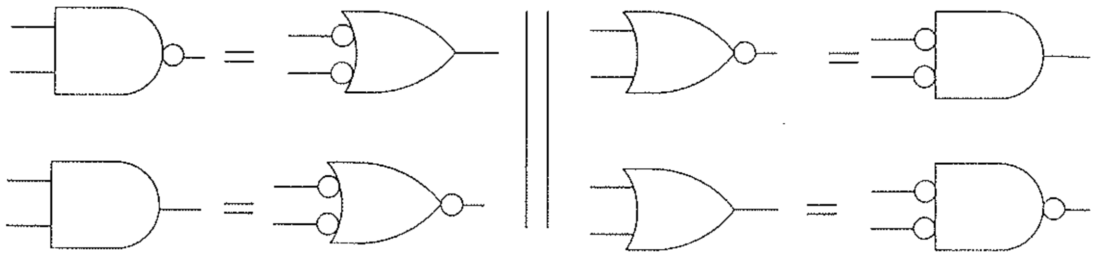


Figure 8.12: DeMorgan equivalent gates.

The logic functions AND, OR and invert can be performed using only NAND gates or only NOR gates (perhaps with a few more inverters than might seem necessary). NAND and NOR gates are very common in most logic families and are generally more useful, with the help of DeMorgan's theorem, than AND and OR gates. In fact, any logic function can be implemented using only NAND gates or only NOR gates. The extra inversions can be understood with a little practice. An example is shown in figure 8.13.

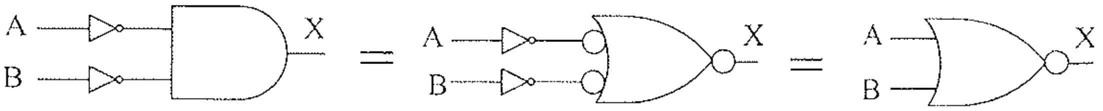


Figure 8.13: Obtaining a DeMorgan equivalent gate.

This shows the equivalence between different logic configurations (of course if you actually use two inverters and an AND gate you must still draw all three as shown on the left and not just the NOR gate on the right). The center logic diagram is obtained by DeMorgan's theorem and the diagram on the right is obtained by canceling adjacent circles (logic inversions).

Another use of Demorgan equivalent gates is shown in figure 8.14. Both circuits (left and right) use three gates and do the same thing. The one on the left uses conventional NOR gate symbols and is hard to figure out the total logic function. The circuit on the right uses a Demorgan equivalent gate and is easy to write down the final logic function of $E = (A + B)(C + D)$ (same logic for both left and right circuits). It is usually best to draw your circuit as on the right so that it is easy to see the final logic function. If *circles drive circles and lines drive lines* then the logic is much easier to figure out.

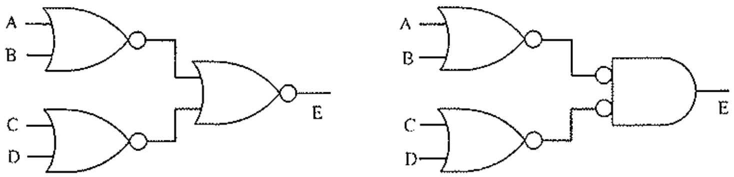


Figure 8.14: Using DeMorgan equivalent gates. Conventional gates (left) and equivalent gates (right). Both circuits perform the same logic function.

A	B	S	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table 8.1: Binary addition of two binary bits A and B yields a sum S and a carry. $A + B = S$

8.6 Addition of Binary Numbers

As an example of typical logic functions, let's consider the problem of adding two binary numbers. A binary number can be represented in parallel on as many separate wires as the number has digits. Alternatively, it can be represented serially on a single wire, if specified time slots are allocated sequentially to each digit. The parallel representation will be discussed here.

The rules for adding two single binary (base 2) digits (also called bits) are well known (see table 8.1), and you will see that the sum, S, is the same as $S = A \oplus B$ and the carry is $C = A \cdot B$. S and C can thus be generated by standard logic gates, as indicated in the circuit in figure 8.15.

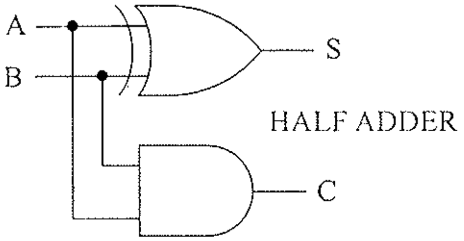


Figure 8.15: Half adder.

(The exclusive-OR function is not, strictly speaking an elementary logic function, but exclusive-OR gates are readily available.) You might occasionally want to synthesize this function from some "left-over" gates in other packages. Alternately, you might look for a package that performs the whole binary addition for you (e.g. the CD4008 4-bit full adder).

This single bit (digit) addition doesn't solve the N-bit problem completely, because it cannot accept a carry from a lower-order digit position. This can be accomplished by using a second, similar stage to add the previous carry to the sum S of A and B. Each individual stage is known as a *half adder*. The full adder, shown schematically in figure 8.16, produces a carry for the next-higher digit whenever either of its half-adder sections calls for a carry. (They never both produce a carry!)

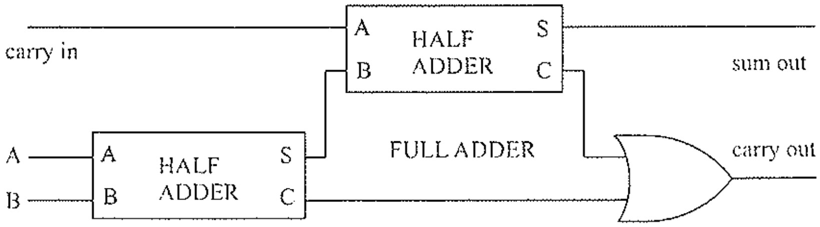


Figure 8.16: Full adder.

Exp. 8.3 Design and build a one bit full adder, using sections of the CD4070 quad exclusive-OR gate and the CD4011 quad NAND gate. First draw the whole circuit using XOR, AND and OR gates and then think about how to change the AND and OR gates into NAND gates using DeMorgan's theorem. Use logic-level switches to provide inputs A and B and the CARRY input, and display the SUM and CARRY out on the LEDs. Check through all eight lines of the truth table to verify that the circuit performs correctly. **end**

Full adder circuits can be *cascaded* into an n-bit adder of arbitrary depth (or number of bits). The term cascade means that the output of one stage is feed into the input of the next stage. A three-bit adder using three one bit full-adders is shown in figure 8.17.

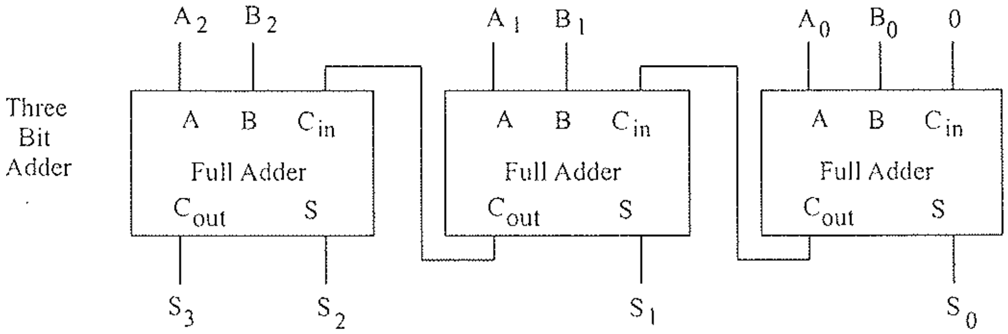


Figure 8.17: Cascading one bit full adders into a three bit adder.

$A_2A_1A_0$ is one three bit number and $B_2B_1B_0$ is a second three bit number (A_0 and B_0 are the least significant bits). The sum of these three bit numbers is produced in $S_3S_2S_1S_0$. Notice that the sum has one more bit than either of the two input numbers. Cascaded one-bit adders similar to this form the basis of many numerical calculations in a computer or calculator. A group of four full bit adders is available in a single IC package as the 4008 (or 74HC283 in the high speed CMOS family) as shown in figure 8.18. The carry in and carry out from each bit are already connected internally. Only the carry-in to the least significant bit and the carry out from the most significant bit are externally available, so that you can cascade several four bit adders together to make an adders of an arbitrary number of bits.

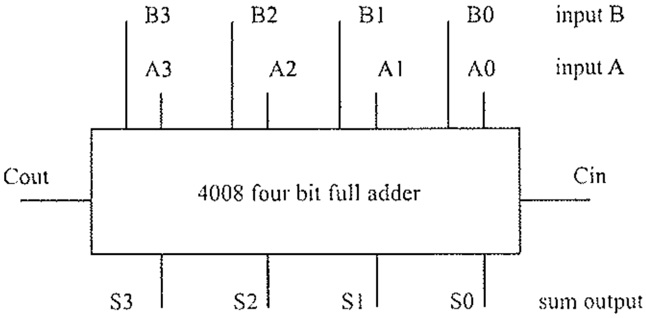


Figure 8.18: Four bit full adder in one IC package.

Exp. 8.4 Test a CD4008 four bit adder. This IC has four one bit full adders combined in one IC package. The Digi-designer does not have enough switches to test all four bits, but you can test it as a two bit adder (to add two 2-bit numbers). Use logic-level switches to provide inputs A_1A_0 and B_1B_0, and display the three bit sum on the LED's. Tie the unused inputs to '0'. Note that some manufacturers may start numbering with 0 and other with 1 (i.e. $A_3A_2A_1A_0$ or $A_4A_3A_2A_1$). Check through the truth table to verify that the circuit performs correctly. OPTIONAL: Vary the carry in and record what happens (you may use a wire if there are no more switches). **end**

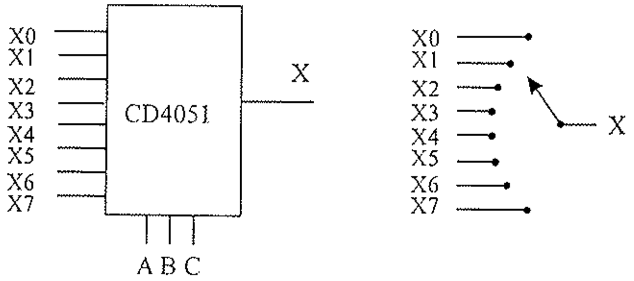
8.7 Multiplexers and Decoders

Although each gate contains several transistors it is not really very complicated in comparison to other digital IC's. Gates are referred to as SSI or *small scale integration*. More complicated IC's are available to perform commonly needed functions. Multiplexers and decoders are two examples of combinatorial MSI or *medium scale integration* functions. The name "demultiplexer" may also be used in place of the name "decoder".

Sometimes it is necessary to select one of several different inputs. This is the function of a multiplexer. The 4051 is a CMOS analog multiplexer. It has 8 inputs (which may be analog or digital signals) to be selected, one output and 3 digital address inputs. The three digital address inputs specify which one of the inputs is to be connected to the output. The multiplexer may be thought of as a single-pole eight-throw switch as shown in figure 8.19. The addressing method for this system is shown at the right. This address is expressed in binary form as CBA, and you can see that this binary number is simply the index attached to the X's in the table above: X = X(CBA). The single output is X, and the eight possible inputs are XO, X1, X2,...,X7. This example is a 1 of 8 multiplexer. Various other sizes are available (1 of 2, 1 of 4, etc.).

A decoder (or demultiplexer) performs the opposite function. It has many outputs and you may select one (and only one) of them to be true. A one of eight decoder would also have three address

1 of 8 Multiplexer



CBA	X
0 0 0	X0
0 0 1	X1
0 1 0	X2
0 1 1	X3
1 0 0	X4
1 0 1	X5
1 1 0	X6
1 1 1	X7

Figure 8.19: 1 of 8 multiplexer.

bits as in the multiplexer above. These address bits determine which of the eight outputs will be true. Sometimes a decoder will also have an additional input that allow you to change the state of the selected output (i.e. 0 or 1 on the selected output and all others 0).

Multiplexers and decoders may also be used to generate an arbitrary truth table describing a complicated logic function without constructing a large array of gates. If there are N independent logic inputs, the truth table has 2^N lines to cover all possible combinations, and each line must show as many entries for the results as there are required outputs. Now, instead of manufacturing the result X from A, B, and C by means of logic gates, you can simply treat A, B, and C as an *address specification* for looking up the output appropriate to the given inputs. If you wire terminals XO...X7 to fixed HI or LO levels, as appropriate to the truth table, you obtain an output X which obeys the desired logic function.

Given the ready availability of selector circuits, there are two advantages in using them instead of networks of gates: (1) the design effort is absolutely minimal, you need only know the truth table and (2) the circuit's function can be changed merely by reconnecting the wires which fix X0...X7. This flexibility is often appreciated when some design errors are discovered at a late hour! The selector circuit acts as a fixed memory for the truth table. This type of memory is called *read-only memory* (ROM). It is programmed, in our example, by suitably wiring some terminals: in more extensive systems, ROM's can be programmed by selectively burning out some tiny fusible links inside a circuit chip, or even by selectively placing charge (permanently!) in certain locations on the semiconductor. To duplicate the full adder of experiment [8.3] you would need two such memories, one to synthesize the sum S and the other for the carry output.

It is but a short step from the concept of read-only memory to that of *read-write memory*, where the information is not stored in such a permanent form as wires, but can be written at will by the action of the circuit itself. In this way the logic function can be modified from time to time, which leads to the powerful technique of *sequential programming*. This is how computers operate, and will be explored in the lab toward the end of the course. First, however, the various building blocks need to be studied in this chapter.

In typical applications, memory space is made available for storing large amounts of information, not just a handful of logic values. This information can serve to control the logic function of a system.

and it can also supply input data to the system, store intermediate results, and maybe even direct the sequence of logical steps in accordance with some of the results obtained in previous steps. Thousands of distinct bits of information can be stored on a single chip. A practical problem then occurs in the means by which any given location can be *addressed*, to read or write information there. If it is possible to select any desired address, in any sequence, the memory is called random-access memory (RAM). In some simpler systems, the information actually circulates continuously in a closed loop. Any particular location then waltzes past a fixed access port only once in a while, and you may have to wait for your answer. This would be called *sequential-access memory.*

8.8 Transmission Gates

The gates studied so far accepted binary logic-level signals at their inputs, and delivered similarly quantized signals at their output. Sometimes it is necessary to control an *analog* signal instead–authorizing or blocking its passage according to the state of an associated command signal. This is the task of a *analog transmission gate.* One of the inputs, and the output, are continuously variable voltages.

The simplest transmission gate consists of a MOSFET controlled in such a way that its channel is either fully open or completely closed. The MOSFET then acts as a voltage-controlled resistor (see section 7.3.1). When 'ON' the MOSFET looks like a small resistance or closed switch and when 'OFF' it looks like a large resistance or open switch. The gate-control mechanism must be quite sophisticated if v_{in} (and v_{out}) swing over a wide voltage range. The whole FET is carried up and down by the signal.

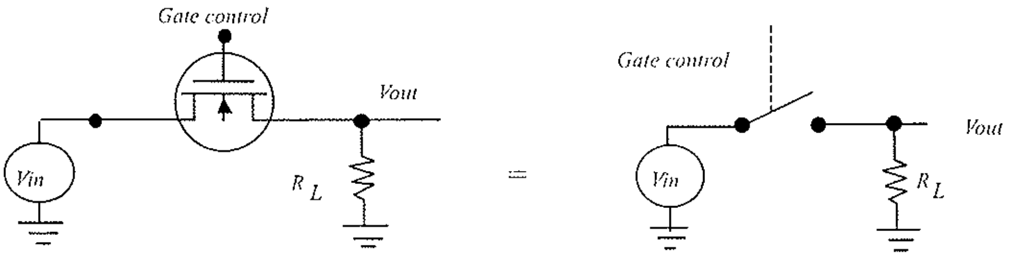


Figure 8.20: Analog transmission gate.

This sketch in figure 8.20 shows an n-channel MOSFET acting as a transmission gate. For wide-range operation it is common to have a p-channel MOSFET connected in parallel. The whole assembly, including the gate-control circuits, is available in integrated form. For example, the CD4066 *quad bilateral switch* contains four independent symmetrical switches (either terminal can be used as input). Gate control is under command of a standard logic-level voltage. The range for v_{in} is restricted to be between V_{SS} and V_{DD} .

$\boxed{\text{Exp. 8.5}}$ Investigate the properties of one section of the CD4066 quad bilateral switch. (Connect the control terminals for the other sections to 0 or +5 V, and likewise one terminal of each of these switches.)

The 4066 will be operating from the Digi-Designer supply, between 0 and +5 V, so the input signal must be kept within the same range. Set up the function generator, by using its amplitude and OFFSET controls, to deliver a signal that swings from +1 to +3V; thereafter do not disturb these settings!

Use the Digi-Designer's clock (set to about 100 HZ) to turn ON and OFF the 4066 switch (using its digital input). Connect a load resistor R_L, in the range 10K to 100K to the output of the transmission gate and the function generator to the input. Put one scope probe on the clock and set the scope to trigger off of this signal. Put the other scope probe on the output of the transmission gate. Set the function generator to deliver sine waves at a frequency that is an integer multiple of the Digi-designer clock (i.e. 100 Hz, 200 Hz, etc.). If the function generator's frequency is carefully adjusted you should observe a nearly stationary sine wave at the output of the transmission gate that is chopped into segments following the digital clock of the Digi-Designer. end

The CD4051 and CD4052 analog multiplexers contain eight (or four) bilateral switches connected, on a side, to a common terminal. They can thus handle analog signals as well as logic levels. Moreover, they can distribute a signal to several terminals as well as accept voltages from several different sources. This is a multiplexer/demultiplexer device. The next experiment will construct an analog multiplexer to understand how it works.

Exp. 8.6 Digital routing of analog signals (see figure 8.21). a) First design and construct a 1 of 4 decoder. This circuit should take 2 digital inputs (from the logic switches, shown as A,B above) that represent a binary address. This decoder should have 4 digital outputs that are addressed by this two bit address (i.e. the two digital inputs from the switches). One and only one of these 4 digital outputs should be high at a time and a different output should be high for each of the four possible combinations of the two inputs. (Hint try 4 2-input AND gates and 2 inverters.) Check this circuit by constructing a four line truth table and measuring all four lines.

b) Next connect each of these four digital outputs to four CD4066 transmission gates as shown below. One side of each of the CD4066 gates should go to an analog input and the other side of each CD4066 gate should be tied together and will form our output. This circuit can now select 1 of 4 possible analog inputs using a binary digital address. Test this circuit and record the output for each line of the truth table for the decoder. end

8.9 CMOS Circuitry

The basic constituents of CMOS logic blocks are complementary MOSFETs working in the *enhancement mode*. Each of these transistors is sketched in figure 8.22. An n-channel device is on the left and a p-channel device is on the right. Enhancement mode means that no current flows until a few volts of "forward" gate bias are applied (with opposite polarity for each type of MOSFET). The symbol for the MOSFET shows the channel broken into three segments, with the diode to the substrate attached

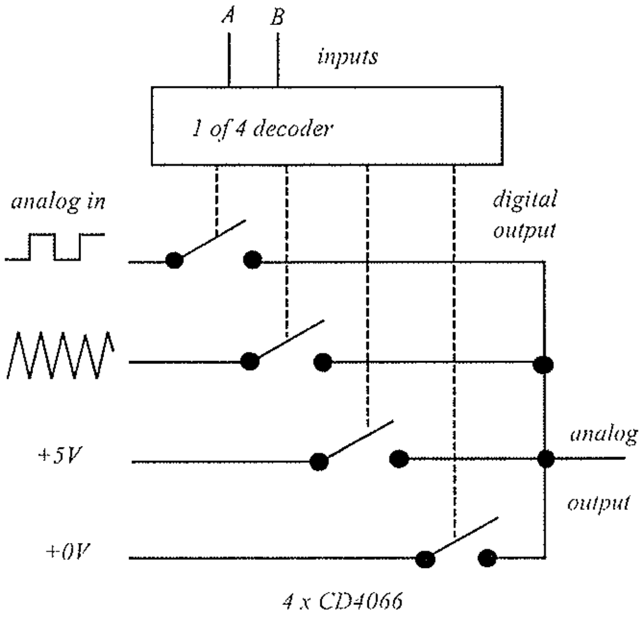


Figure 8.21: Analog multiplexer.

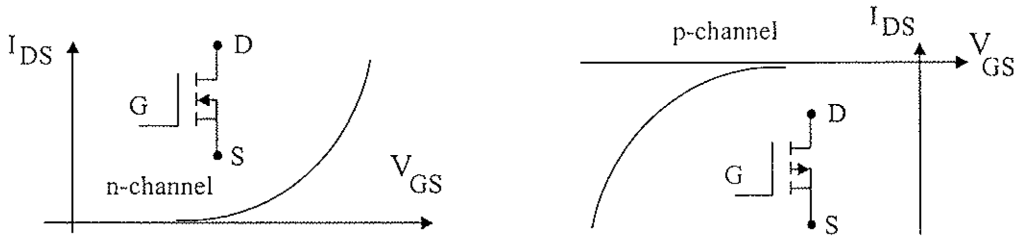


Figure 8.22: MOSFET transistors.

to the center segment. This diode points *in* for an n-channel device and *out* for a p-channel device. The gate, shown as a long line, has its connection at the source end.

The basic CMOS configuration functions as an *inverter*. A p-channel and an n-channel MOSFET are stacked in series, in a *totem-pole* circuit as in figure 8.23. v_{in} is connected to the two FET gates in parallel. The logic levels are close to the two supply rails (unfortunately the symbols for these two supply-rail potentials are taken over from n-channel FET technology: V_{SS} and V_{DD}). To simplify the discussion, assume that V_{SS} is at ground potential (a common choice).

The top transistor in the inverter (figure 8.23) is a p-channel MOSFET and the bottom transistor is an n-channel MOSFET. Note that the substrate and source of the top transistor are connected to the top supply voltage (V_{DD}) but the same connection in the bottom transistor goes to the bottom supply voltage (V_{SS}). If the inverter is driven by a digital signal with a high voltage of V_{DD} and a low voltage of V_{SS} then the state of both transistors for all possible values of V_{IN} can be summarized in the truth table given in table 8.2.

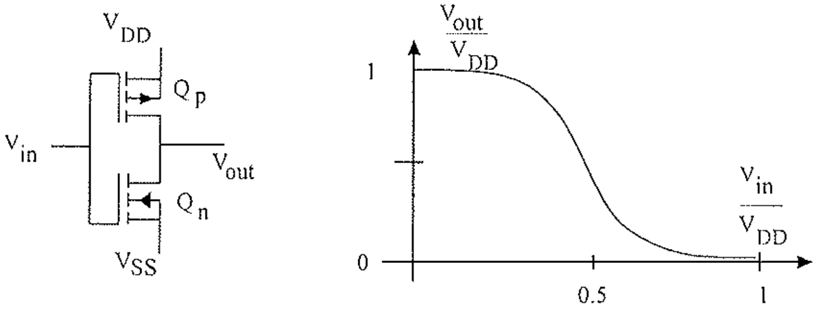


Figure 8.23: CMOS inverter (V_{SS} =0).

v_{in}	Q_n	Q_p	v_{out}
V_{DD}	ON	OFF	V_{SS}
V_{SS}	OFF	ON	V_{DD}

Table 8.2: Transistors states in a CMOS inverter (figure 8.23).

When v_{in} is at either logic level, the current through the circuit is negligible, since one or the other of the FET's is cut off. With v_{in} somewhere in the gray zone between logic levels, both transistors can conduct simultaneously. This condition should be avoided except for brief intervals of time. For this reason v_{in} is usually specified to cross between logic levels with a rise time no longer than about 10 microsec, and no input may be left completely disconnected. Because of the high resistance of the insulated-gate devices, such a disconnected input could "float" into the gray zone and remain there, causing the device to overheat.

With v_{in} at one logic level, the terminal for v_{out} is drawn to the opposite supply rail through the resistance of the MOSFET which is turned ON. If there is no external load, v_{out} comes to within about 50 mV of the supply rail. This action occurs symmetrically; v_{out} can be both pulled down or up. This gives rise to the name *active pull-up* for the configuration, as contrasted to the case where a resistor is the only thing pulling v_{out} up toward V_{DD} .

The transfer curve of the inverter is centered nominally on $v_{in} = 0.5V_{DD}$ (assuming that V_{SS} =0). Since the arrangement functions rather like a resistive voltage divider in which one or other resistor is switched, v_{in} and v_{out} scale roughly in proportion to V_{DD}. The allowable range for V_{DD} is from 3 V to 15 V.

Sample-to-sample variations cause the crossover point for v_{in} to vary, perhaps from $0.3V_{DD}$ to $0.7V_{DD}$. (Though sample-to-sample variations are large, the gates on a single multifunction chip tend to be quite similar to each other. They were manufactured under the same conditions. Occasionally this on-chip similarity can be put to good use.) However, the levels for v_{out} do not have such a variation. They remain almost indistinguishable from 0 and V_{DD} . Hence, the error margin is large: one device drives the next with voltages which fall clearly into the unique logic-level regions. This tells us that a voltage noise of up to $0.3V_{DD}$, peak, could be superimposed on the signals without producing an error. This is known as the dc noise margin, and it is particularly large in the CMOS logic family.

A	B	Q_1	Q_2	Q_3	Q_4	C
0	0	ON	ON	OFF	OFF	1
0	1	ON	OFF	OFF	ON	0
1	0	OFF	ON	ON	OFF	0
1	1	OFF	OFF	ON	ON	0

Table 8.3: Transistor state table for the CMOS NOR gate in figure 8.25.

External loads fall into two classes: those connected to V_{DD} (or a more positive potential), and those connected to ground or a more negative potential (see figure 8.24). The first call upon the output terminal to *sink* the load current, the second to *source* it. Because of the inherently symmetrical arrangement, the CMOS inverter acts in about the same manner as a sink or as a source. Typically, only a few mA of load current can be handled without taking v_{out} out of the defined logic-level regions. Since CMOS inputs draw negligible current, this restriction is immaterial as far as interconnections between devices are concerned. The limitation slows down the effective transition rates, because of the need for charging circuit capacitances. It also implies that separate output devices must be used if substantial load currents must be delivered to the outside world.

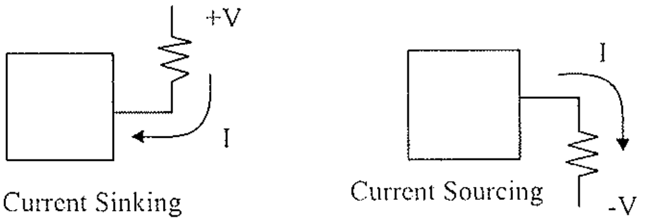


Figure 8.24: Source versus sink.

The insulated gate of a MOSFET can readily be punctured by excess voltage, and such a puncture, even if caused by a very low-energy source such as an electrostatically charged person, ruins the device. Hence all CMOS inputs are protected with diodes which conduct if v_{in} goes outside the allowable range. These diodes are themselves quite fragile, however, and are readily burned out if asked to carry large currents (perhaps more than 100 microAmp). Hence, in addition to the rule: *Keep v_{in} between the supply rails*, we have the additional: *Protect the protection!* In other words, when a circuit risks momentarily applying excess voltage to the input, include a series resistor to limit the resulting "protected" current to about 100 microAmp.

The basic totem-pole circuit can be modified to work with several inputs (see circuit in figure 8.25) by connecting FETs in *parallel* at one level (Q3 and Q4), but in series at the other (Q1 and Q2). In the example shown below, C can be pulled LO by either of the n-channel FETs (Q3, Q4), but it can be pulled HI only if both of the p-channel FETs collaborate. Thus C is LO if either A or B is HI (see table 8.3). This is a NOR gate. If, instead, the upper FETs were in parallel and the lower in series, this would be a NAND gate.

The system can be extended to a larger number of inputs. Each FET in the ON condition has

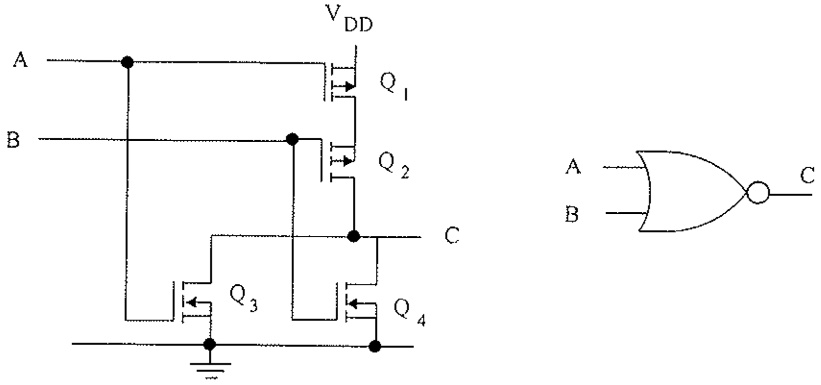


Figure 8.25: CMOS NOR gate.

only a small voltage drop across it, and thus doesn't affect the conditions for the other members in a "chain" too much. However, when there are many FETs in series on one side and many in parallel on the other, a certain asymmetry in output performance does develop; moreover, the effective logic threshold for each depends a little on what the other inputs are doing.

To standardize matters, buffer stages (simple inverters) can be inserted at the inputs and at the output. This produces an excellent output symmetry, and the large cascaded gain of the arrangement produces a very sharp, clean transfer curve. This type of CMOS gate (the buffered version) is identified by a suffix B. The buffered devices introduce a greater time delay between input and output (propagation delay), however, and their high gain makes it important that the signals do not linger near logic thresholds, otherwise some form of oscillation on the output may be observed.

8.10 Schmitt Trigger Inputs

The normal logic input responds correctly to signals in the defined HI and LO voltage regions; however, its own transition can occur anywhere within the band between these regions, and the width of its transition zone can be considerable. This causes no difficulty when the input is driven by a standard logic output. When less well-behaved input signals are applied, problems can arise during the time the input spends in the transition zone.

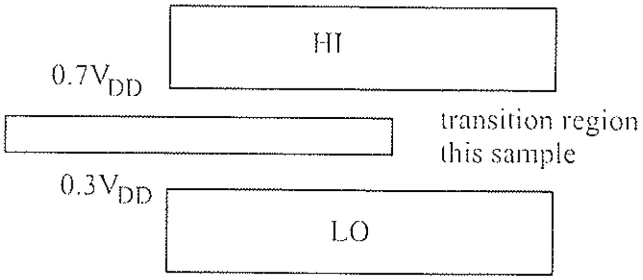


Figure 8.26: Schmitt trigger levels.

The response can be sharpened if the input exhibits *Schmitt trigger* characteristics. The change from LO to HI (or vice versa) is then made regeneratively, once the signal has reached a certain threshold. The speed of the transition is determined by the circuit's internal capabilities, not by the signal. The signal may even have some noise superimposed on it without causing problems, provided the noise is smaller than the *hysteresis* between the upper and lower thresholds. This type of input is known as a *Schmitt* input (cf. section 2-6),and is identified on the circuit by a small "hysteresis loop" symbol (see figure 8.27).

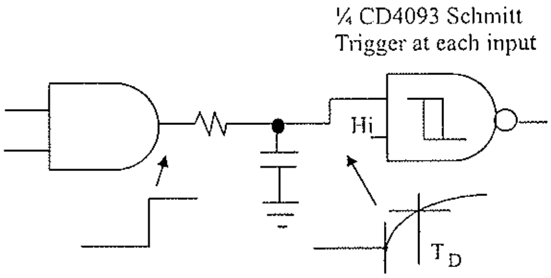


Figure 8.27: Gates with Schmitt trigger input.

The hysteresis at a CMOS Schmitt input is typically 0.9 V when V_{DD}=5 V. The signal applied to this input is permitted to move between logic levels at an arbitrarily slow rate. You must remember, however, that logic significance (i.e. valid data) is associated with it only at those times when it has safely reached either of the defined logic zones.

Beyond being tolerant to noisy, "unconditioned" signals, Schmitt trigger inputs are useful when a time delay is to be produced with the help of a low-pass R-C filter. The definition of the delay time, T_d, is subject to wide variations because the Schmitt thresholds are not held to close tolerance, but in many situations such a time delay is nevertheless useful.

8.11 Tri-State Outputs

Many information-handling systems are organized around *information buses* which are lines common to many devices, any of which can either transmit or receive information via the bus. Evidently some careful sequencing of operations is required, or all would be confusion.

A practical problem occurs because the CMOS output terminal represents a low impedance in each of its two states--there is always a conducting FET tying the terminal to a fixed potential. Thus several outputs cannot be connected to a common bus line without some provision for isolation. One simple expedient is to place a bidirectional switch between the CMOS output and the bus. Turning the switch on *enables* the output to transmit information; all other transmitters must be turned off, and any listener that wishes to pay attention can use the information.

To simplify the hardware, special *tri-state* outputs are available which make a direct bus connection possible. In addition to the LO and HI states the output terminal can be *inhibited*, in which case it presents a very high impedance. This is done by turning both the n- and the p-channel MOSFETs in the output stage off. The tri-state control signal is variously called ENABLE or INHIBIT, depending on whether a 0 or a 1 is needed to produce the open-circuit condition.

8.12 Simplifying Logic with Karnaugh Maps

Reducing complicated Boolean expression into their simplest form using Boolean algebra can be a tedious task. A Karnaugh map is a quick method of reducing a complicated logic expression to a short simplified form. Consider a function of two logic variable with three terms:

$$C = AB + A\overline{B} + \overline{A}B \tag{8.19}$$

This function is shown graphically in fig. 8.28. The variable A is mapped on the left vertical axis and the variable B along the top horizontal axis. A 1 means the variable appears in its true state and a 0 in its inverted state. There is a cell for all possible combinations of two logic variables. If there is a 1 in the cell the term is present, and blank (or 0) means the term is not present.

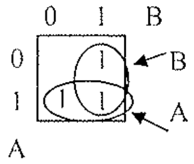


Figure 8.28: Karnaugh map of two variables in eq. 8.19. 1 means the term is present and blank or zero means the term is absent.

Adjacent cells contain both the true and inverted form of a given variable which cancel. The horizontal oval on the bottom shows that $A\overline{B} + AB$ may be converted into just A and the vertical oval on the right shows that $\overline{A}B + AB$ may be converted into just B leaving:

$$C = AB + A\overline{B} + \overline{A}B = A + B \tag{8.20}$$

One term is used twice. The Karnaugh map is a method of quickly finding ways to combine terms into a simpler form. You should verify for yourself that Boolean algebra yields the same result.

With more than two variables a Karnaugh map gets a little more complicated. Each side of the map may contain more than one variable in a sequence such that there is only one change between adjacent lines. This requirement generates a sequence similar to but different than counting in binary.

Consider a function of three variable A, B, C as in fig. 8.29a).

$$D = \overline{A}\cdot\overline{B}\cdot\overline{C} + \overline{A}\cdot\overline{B}\cdot C + \overline{A}\cdot B\cdot C + A\cdot\overline{B}\cdot\overline{C} \tag{8.21}$$

The four terms may be grouped in several different ways, one of which is shown. The sides and corners wrap around to the opposite sides and corners in a strange manner. This expression can be reduced to:

$$D = \overline{B}\cdot\overline{C} + \overline{A}\cdot C \tag{8.22}$$

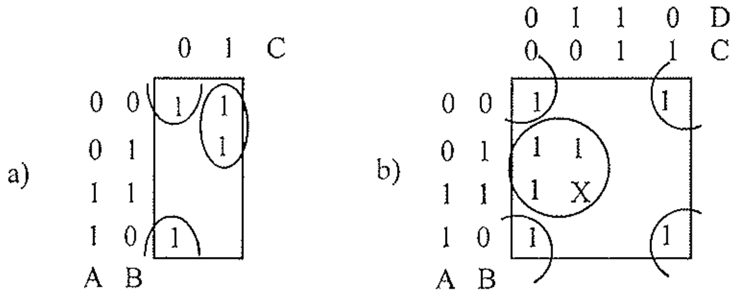


Figure 8.29: Karnaugh map of more than two variables. a) Three variable in eq. 8.21 and b) four variable as in eq. 8.23.

A function of four variable is shown in fig. 8.29b).

$$\begin{aligned} E = \quad & \overline{A}\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}+\overline{A}\cdot\overline{B}\cdot C\cdot\overline{D}+\overline{A}\cdot B\cdot\overline{C}\cdot\overline{D}+\overline{A}\cdot B\cdot\overline{C}\cdot D \\ & +A\cdot B\cdot\overline{C}\cdot\overline{D}+A\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}+A\cdot\overline{B}\cdot C\cdot\overline{D} \end{aligned} \tag{8.23}$$

In this case the term $A\cdot B\cdot\overline{C}\cdot D$ is known to never occur in the real circuit, and is marked as a *"don't care"* or X. This term may be used or ignored as needed. Using the Karnaugh map this expression can be reduced to:

$$E = B\cdot\overline{C}+\overline{B}\cdot\overline{D} \tag{8.24}$$

using the *"don't care"* term, which is a considerable simplification. This procedure may be expanded to more variables although it becomes less easy. Groups of even numbers of terms (2,4,8, etc.) may be combined (more than 4 not shown here).

8.13 Practice Problems

[1] a) Design a circuit to implement the two input AND function using only NAND gates. b) Repeat part b) but make the OR function.

[2] a) Design a circuit to implement $E = AB + CD$ using only NAND gates. b) Design a circuit to implement $E = (A + B)(C + D)$ using only NOR gates. (Hint: both require only three gates.)

[3] Design a Smart Alarm Clock. You have been forced to take a class at 9AM every day next semester and need a new alarm clock to wake you up on time. You need to have the alarm go off at 8AM every day during the week (Monday through Friday) but it should also let you sleep later on the weekends. However it should still go off at 10AM on the week end (you don't want to miss breakfast do you). You have found two digital alarm clocks. One clock has a CMOS digital output that becomes true (i.e. goes high and stays high until manually reset) when the set time is reached. The other has this same feature but in addition outputs seven CMOS outputs, one for each day of the week. These outputs are however inverted (for example the MON signal is 0 on Monday but 1 the other six days of the week). Design a logic circuit that has two time-inputs (T_8 for 8AM and T_{10} for 10AM) and seven

day-input (SUN, MON, TUE, WED, THUR, FRI, SAT) and decides when to sound the alarm.
a) Write a Boolean expression for the alarm signal A (i.e. sound the alarm when A is true).
b) Draw a logic diagram relating the above CMOS inputs to the CMOS output A. You may use any two input gate as well as inverters.

[4] Design a logic circuit that will set off a burglar alarm (logic signal B = true to sound alarm) only when the alarm is activated (logic signal I = true when inactive or false when activated) and one or both of the following happens; i) the window is open (signal W = true when window open), ii) the door is open (signal D = true when door closed). Write a logic equation for the output B and then draw the circuit using a minimum number of two-input NOR gates and the input signals U,D,W.

[5] Using only Boolean algebra show that the logic circuit shown in figure 8.30 implements the XOR function defined as $E = \overline{A} \cdot B + A \cdot \overline{B}$.

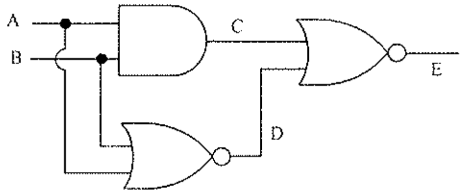


Figure 8.30: Problem 5.

[6] For the logic circuit shown in figure 8.31, write a Boolean expression for the output Y in terms of A, B, C0, C1, C2, C3 and E. This circuit is available in the 74xx TTL and 74HCxx CMOS logic families as the 74x153 multiplexer (two per package).

[7] Design a combinatorial logic circuit that will determine if two 2-bit binary numbers are equal. The two binary numbers can be written as $A = (a_1, a_0)$ and $B = (b_1, b_0)$ where a_0 and b_0 are the least significant bits and a_1 and b_1 are the most significant bits. The circuit should have four inputs (a_1, a_0, b_1, b_0) and one output, C, which should be a logic 1 if the two numbers are equal and a logic 0 otherwise.
a) Write a Boolean expression for C in terms of a_1, a_0, b_1, and b_0.
b) Draw the logic circuit using any two-input gates that were available in the lab.

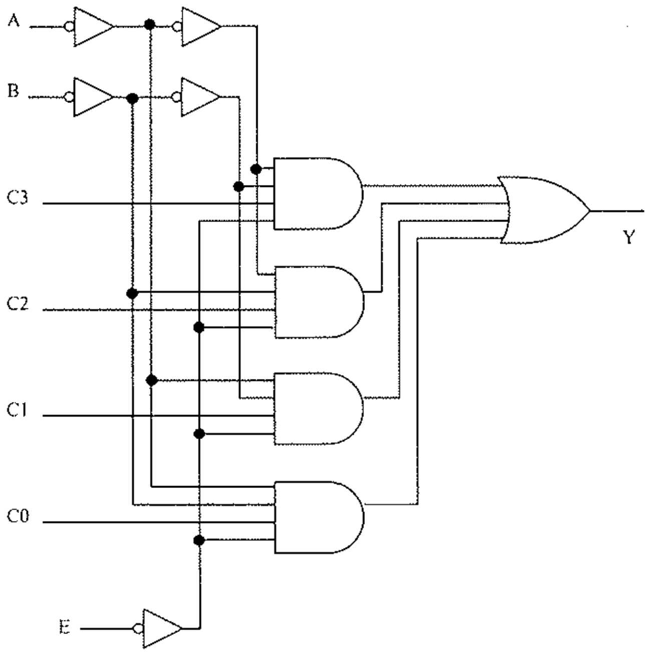


Figure 8.31: Problem 6.

Chapter 9

SEQUENTIAL CIRCUITS

9.1 Introduction

Logic circuits whose outputs depend only on the present state of their inputs are *combinatorial* logic (such as those studied in the last chapter). If the present output depends on both the present input and the past value of the input the circuit is *sequential.* Sequential circuits depend on the sequence of input values. The most common (and fundamental) sequential element is the flip-flop, which is also a one bit memory element. Several different types of flip flops will be studied in the next few sections. Counters, shift registers and memory elements are examples of complex sequential circuits. For a more detailed discussion of sequential and combinatorial digital systems see, for example, Wakerly[19].

9.2 R-S Flip-Flop

Two inverters connected in a loop as shown in figure 9.1 have two different equilibrium states: A=1, B=0; or A=0, B=1. In each state the output of one inverter supplies just what the other inverter requires for its input.

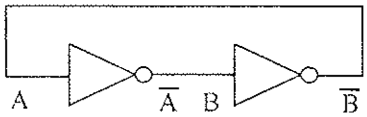
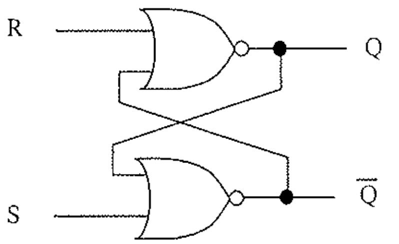


Figure 9.1: Simple bistable flip-flop.

This *bistable* circuit is a simple example of a *flip-flop* (or *latch*). If there is a way of forcing this circuit into a particular state, it can serve as *memory* to store one bit (i.e. a 0 or 1) of information. By examining its state at some later time the information can be retrieved. Flip-flops are distinguished mainly by the method adopted for forcing them into the desired state. The simplest method is to use NOR gates instead of the inverters.

A pair of *cross-coupled* NOR gates is shown in figure 9.2. It has two inputs (R and S) and two complementary outputs (Q and $\overline{Q}$). When both terminals R and S are LO, each NOR gate functions as a simple inverter for its other input and this is the original bistable circuit shown in figure 9.1. This



RS Flip-Flop

R	S	Q	$\overline{Q}$
0	0	old values	
0	1	1	0
1	0	0	1
1	1	not-allowed	

Figure 9.2: Cross-coupled NOR gate RS flip-flop.

situation (R=S=0) is the *standby* condition during which the flip-flop holds information in the form of Q=1 or Q=0. To force the flip-flop into the desired state, one of the terminals R and S (but not both) is momentarily made HI. For example, if S=1 (leave R=0) the output of the lower NOR gate goes LO ($\overline{Q}$=0). The upper gate then has two LO inputs and its output goes HI (Q=1). This state persists even after the signal S=1 is brought back LO (S=0). The HI input at S has SET the flip-flop. Similarly, if a HI level is applied to terminal R, the flip-flop is forced into the state Q=0. R is the RESET (or CLEAR) input. This simple flip-flop is called an R-S flip-flop. It responds directly to its two controlling inputs R and S, and provides complementary outputs, Q and $\overline{Q}$. Assuming that Q=0 initially then the sequence of R,S (vs. time) will produce the indicated response in Q shown in figure 9.3.

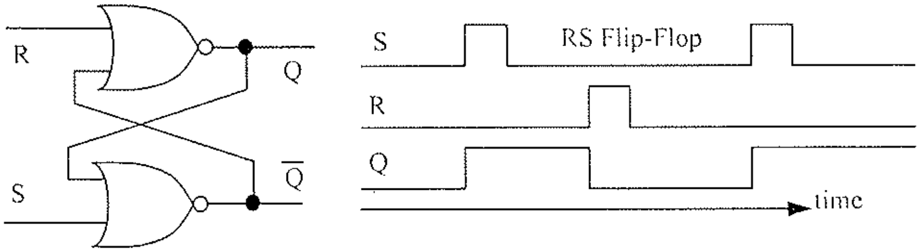


Figure 9.3: Time dependence of the RS flip-flop.

An important limitation of the R-S flip-flop is that it responds in an unpredictable way if both inputs R and S are simultaneously made HI. If R=S=1, then *both* Q and $\overline{Q}$ are LO. Not only does this conflict with their names, but it also can produce an undefined output when you try to return to the R=S=0 state. You cannot change two signal at exactly the same time so one of them will be 1 slightly longer than the other in a random order. You should avoid using this state with R-S flip-flops (indicated in the truth table by labeling this state as "not-allowed").

Frequently, digital inputs will be controlled by mechanical switches. Switches usually have two metal contacts that touch to close the switch. This mechanical contact will not be a clean on/off but may bounce many times (over several milliseconds) before settling down. The initial contact bounces but the initial separation is clean (this will be important for the switch debouncers discussed later). Digital circuits are fast enough to respond on this time scale and this bounce may cause a lot of problems. The next two experiments will investigate how an R-S flip-flop can be used to "clean up" (or condition) the signal from a mechanically operated switch. First study the switch action itself:

> $\boxed{\text{Exp. 9.1}}$ Connect a 1K resistor to one of the four switches on the Digi-Designer as shown in figure 9.4. Next set the scope to "normal" trigger (i.e. remove auto-trigger) so that there is only a trace when there is a valid trigger. Using about 0.1 to 1 mSec/div, adjust the trigger level with the probe connected to the clock output (i.e. the clock signal is about the same amplitude as the signal from the switch). Once the scope is triggering properly move the probe to the top of the 1K resistor connected to the switch. Now move the switch back and forth between 0 to 1 several times and observe its output waveform. You should see a generally random oscillation between 0V and 5V for a brief period after the switch is thrown (there should be something like a dashed line at both 0V and 5V at the same time, for a short period after the switch is thrown). Note that each switch will respond differently and you may need to try different switches to see a large bounce $\boxed{\text{end}}$

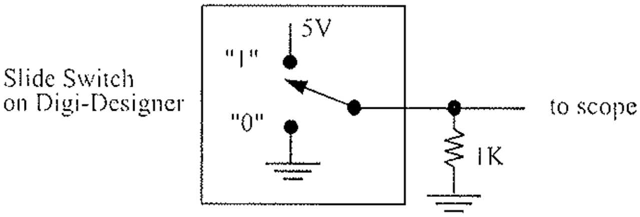


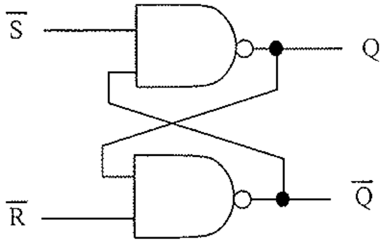
Figure 9.4: Mechanical switch.

This chapter will introduce another logic family called TTL which stands for transistor-transistor logic. For now, just use it "blindly" in the Digi-Designer with the 5V power supply (the 5V level is mandatory for TTL). The next section discusses how TTL works. Note that TTL inputs (unlike CMOS) may be left unconnected. An "open" TTL input floats to a HI level. However, since floating inputs are subject to noise pickup, it is better practice to connect a HI input firmly to +5V.

> $\boxed{\text{Exp. 9.2}}$ Build an R-S flip-flop from two TTL NOR gates in a 7402 IC as shown in figure 9.2. Connect two of the slide switches on the Digi-Designer to SET and RESET and connect the outputs to the LED display on the Digi-Designer. You may leave the inputs to unused TTL gates unconnected. First verify the truth table for this circuit. Next use the scope to verify that the transitions at Q and $\overline{Q}$ are clean. $\boxed{\text{end}}$

An R-S flip-flop can also be made from two NAND gates as shown in figure 9.5. In this case the standby state has the two inputs HI, so that the two NANDs can function as simple inverters for their inner inputs. By making either of the outer lines LO the flip-flop can be forced into the desired state. Note that the two input signals (outer lines) are labeled $\overline{S}$ and $\overline{R}$, to indicate (in positive logic) that the LO state produces the SET or RESET action.

The signal from one single-pole-double-throw switch (SPDT) can be conditioned or *debounced* by a circuit like the one shown in figure 9.6. The flip-flop remembers the first bounce of the switch and



RS Flip-Flop

$\overline{R}$	$\overline{S}$	Q	$\overline{Q}$
0	0	not-allowed	
0	1	0	1
1	0	1	0
1	1	old values	

Figure 9.5: Cross coupled NAND RS flip-flop.

produces a clean edge at its output Q. It is important that contact separation on the top or bottom does not bounce but only making contact on top or bottom bounces. This circuit works for both directions of the switch (i.e. the low to high and the high to low transition of Q). The logic-pulse push buttons on the Digi- Designer work on this principle and should be used in the experiments to follow when a clean switch is needed.

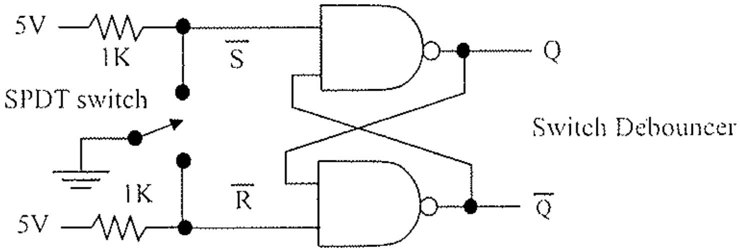


Figure 9.6: Switch debouncer.

9.3 TTL Logic

This is an extensive and widely accepted logic family that also contains several subgroups. The schematic for a standard 7400 two input NAND gate is shown in figure 9.7. There are four two-input NAND gates in one 14 pin IC package.

The output section of the gate contains a totem-pole (T2 and T3) driven by a saturating inverter (T1). The totem pole output provides a fast edge for both the rising edge (via emitter follower T2) and the falling edge (via saturated switch T3). When the base of T1 is HI it saturates and pulls down the base of T2 and cuts it off, but pulls *up* on the base of T3 and thus saturates T3 also. The base current of T3 is limited by the 1.6 K resistor from the supply rail (through T1), and is about 2 mA. Thus T3 can pass a considerable collector current while remaining in saturation. The TTL output is rated to *sink* at least 16mA in this LO state and is guaranteed to be no higher than 0.4 V.

For the opposite state, when the base of T1 is LO, T1 is cut off and so is T3. T2 now receives base current through the 1.6K resistor. The higher the output voltage rises, the smaller this base current becomes. Also, because of the forward voltage required by the base-emitter junction of T2 and the series diode, the output voltage at C cannot rise as high as the supply rail. Evidently the output is

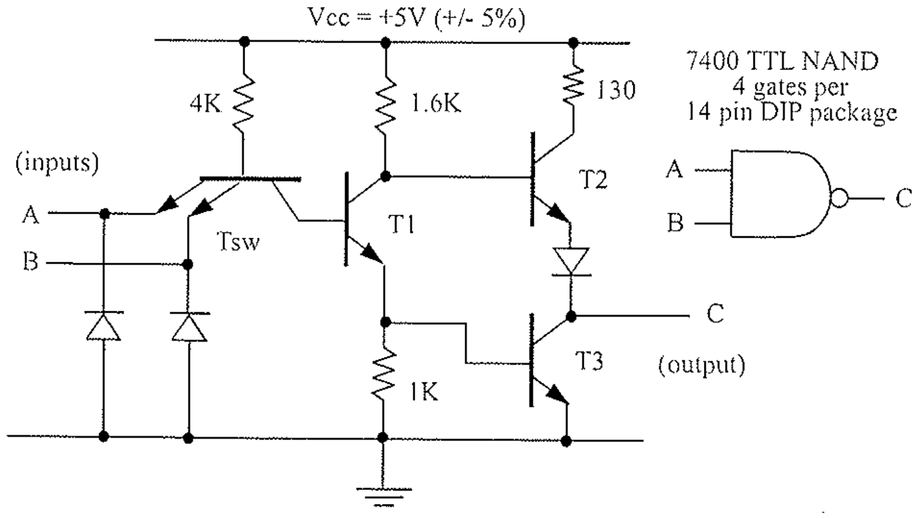


Figure 9.7: Internal working of a TTL NAND gate.

much less capable of *sourcing* current in the HI state than of *sinking* current in the LO state. This asymmetry matches the input requirements of the family. An unloaded output may typically rise as high as 3.5V. Under maximum rated load it is guaranteed to go at least to 2.4 V.

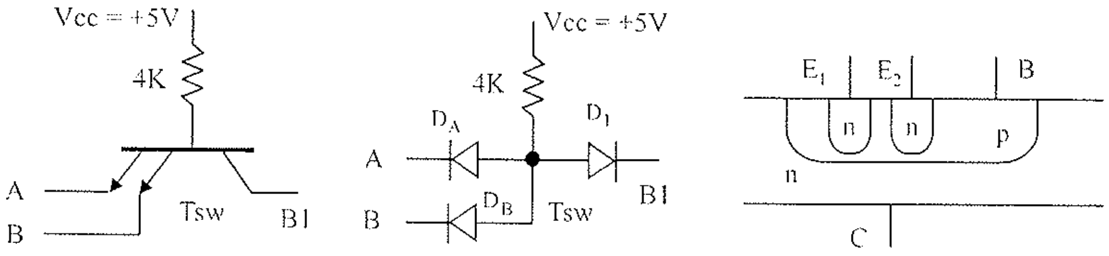


Figure 9.8: Dual emitter transistor in the TTL NAND gate (left). A conceptual model using diodes is shown in the center (the point labeled B1 is connected to the base of transistor T1 (figure 9.7). A possible realization in silicon is shown on the right.

The input transistor Tsw is used in a somewhat unusual manner and does the actual logic function. A conceptual version of this device using diodes is shown in figure 9.8. With inputs A and B both HI, no current flows through either emitter from the base but instead flows from the 4K resistor through the forward biased collector-base diode into T1 (the base=HI state above) in a rather non-standard manner for a transistor. With *either* A or B input LO, Tsw is saturated, taking the current from the 4K resistor through its emitter, and draws the base of T1 firmly down (base=LO above). The way current sloshes from one side to the other of the base in Tsw is strange, but it's efficient. The switching action is rapid and clean. Note that the input draws only a small leakage current in its HI state (Tsw OFF), but must sink all the current from the 4K resistor for the LO state. *TTL inputs must be pulled LO by sinking a substantial current* (rated to be at most 1.6 mA).

	74xx		74LSxx	
logic level	input	output	input	output
0	<0.8V (1.6mA)	<0.4V (16mA)	<0.8V (0.4mA)	<0.4V (8mA)
1	>2.0V (0.04mA)	>2.4V (0.4mA)	>2.0V (0.02mA)	>2.7V (0.4mA)

Table 9.1: TTL specifications, standard (74xx) and low power Schottky (74LSxx).

TTL thus differs from CMOS most importantly in this respect: the inputs are not open circuits and if left unconnected they "float" up to the HI state. Because of the input current needs, a TTL output cannot drive an unlimited number of inputs. TTL has a rated *fanout* of ten. This means that at most ten inputs can be driven from one output. TTL also differs from CMOS in that the logic levels are not as close to the supply rails. Table 9.1 shows the guaranteed limits for outputs and inputs, and indicates that there exists a DC noise immunity of about 0.4V (i.e. the difference between the input voltage and the output voltage). The HI output levels, particularly, tend to vary from circuit to circuit and depend on the combination of input signals present. This table also includes the specification for the low power Schottky series of TTL (74LSxx) which is designed to have the same pin-out and speed as standard TTL (74xx) but one fourth the power.

The NAND function (as shown above) is the most common form of TTL. The NOR, AND and OR functions can also be made but generally are not as fast. For example the 7408 AND gate just adds an extra inverter between T1 and the totem pole output T2 and T3. Please refer to the TTL data book for the circuit descriptions of these logic function.

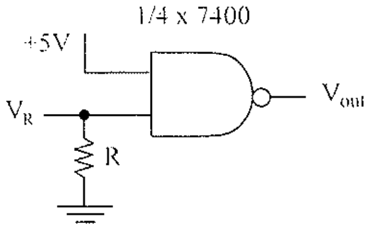


Figure 9.9: Input characteristics of a TTL NAND gate.

Exp. 9.3 Investigate the effect of pulling a TTL input LO through resistor R of various values 220, 470, 1K, and 2.2K ohms as shown in figure 9.9. For each resistor value measure V_R and v_{out} (the point labeled V_R is for measurement only and should not be connected to anything other than a voltmeter or scope input). end

Conservative design practice specifies that R, in the situation just explored, should be kept to 330Ω or less for a valid logic 0. TTL inputs usually have protective diode clamps, as shown, which means that they should never be driven negative or the diodes may be destroyed.

The totem-pole output transistors in TTL do not conduct simultaneously. However, during a transition one transistor may not have had time to turn off before the other comes on. There may be

TTL series	Input-low I(mA)	T_D (ns)	P (mW)	$T_D \times P$ (pJ)
7400	1.6	10	10	100
74H00	2.0	6	23	138
74LS00	0.4	8	2	16
74F00	0.6	3.5	8	28
74S00	2.0	3	20	60
74ALS00	0.1	3.5	1.3	4.6
74AS00	0.5	2.0	8	16

Table 9.2: Speed-power product for various TTL logic series (typical values quoted).

a short (10ns) but fierce (100 mA or more) current "spike", which can cause interference with other devices. *Good bypassing of the supply rail is important.* (Use a ceramic capacitor with short leads between +5V and ground.) A good rule-of-thumb is to have one capacitor (0.01 to 0.05μF) per 1 to 3 IC's. Altogether, TTL is considerably faster than 4000 series CMOS. Transition and delay times are of order 10ns. Hence it is much more important to build circuits neatly. With a "birdcage" you sometimes get crosstalk between signals which causes the logic to do unexpected things, yet you can't see the short crosstalk spikes on the scope. Even though one can often get away with messy circuit layout, the occasions when it causes trouble can be very mystifying and painful!

There are several different families of TTL. These differ mainly in the use of different types of bipolar transistors and different resistor values. The basic structure of the gate stays mostly the same, however the input and output current requirements are different for each TTL family. The functions in one family are mostly duplicated in another (the pin-out is usually identical to allow interchanging parts from different TTL families) and you can mix these families in a circuit with only minor precautions. The net effect is a variety of available speed and power per gate or IC. There is a tradeoff between speed (propagation delay per gate T_D in ns) and power (per gate in mW) as summarized in the table 9.2.

Note that the propagation delay for the low-to-high transition may not be the same as the propagation delay for the high-to-low transition (only the average delay is shown in the table). The overall performance is sometimes characterized by the speed-power product $T_D P$ (in pJ = pico-Joules). A lower $T_D P$ would indicate a better overall performance. The 7400 series was invented first and is often called "standard TTL". The 74LS00 series was designed to have lower power but approximately the same speed as standard TTL and is also quite popular. LS stands for low power Schottky. A Schottky clamped bipolar transistor has special features that allow it to transition between cutoff and saturation very quickly.

Comparing TTL to CMOS is a little complicated because the power required by CMOS varies with frequency. At DC, CMOS draws essentially no power (i.e. approx. < 0.02mW/gate) but the power increases linearly with frequency. The main power loss is due to the current required to charge and discharge the small capacitance at each MOSFET input. It is common to quote the power at a typical

CMOS series	T_D (ns)	P (mW at 1MHz)	$T_D \times P$ (pJ)
4011B (5 volts)	120	1.5	180
4011B (15 volts)	35	14	490
74C00 (5 volts)	50	2	100
74HC00	8	1.8	14
74HCT00	14	2.0	28

Table 9.3: Speed-power product for various CMOS logic series (typical values quoted).

frequency of 1MHz with a capacitive load of 50pfd as shown in the table 9.3.

The 74HC series is designed to have the same pin out and functionality as TTL but the power requirements of CMOS. It is increasingly replacing TTL in many circuits principally because it draws no power when it is idle. The 74HC series however cannot be connected directly to TTL (a 74HC output can drive one TTL input but TTL cannot drive a 74HC input because of different input voltage requirements). The 74HCT series is designed to interface between 74xx00 TTL and 74HC00 CMOS devices. However when a 74HCT device has a TTL voltage level as input its power requirements increase significantly. Please refer to the manufacturers data book for more details.

The propagation delay for TTL is so short that it can be difficult to measure. One method is to measure the average delay of several gates with a ring oscillator as shown in figure 9.10.

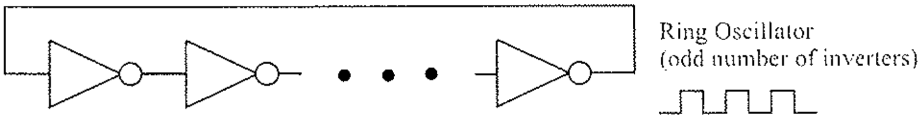
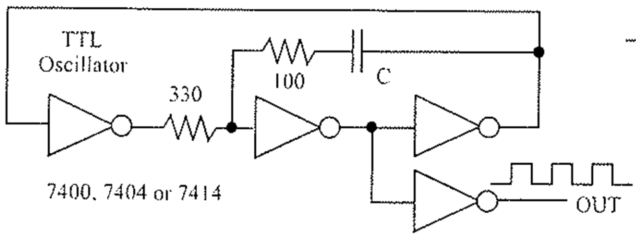


Figure 9.10: Ring oscillator to measure the average propagation delays.

Exp. 9.4 [OPTIONAL] Build a ring oscillator from about 5 or 7 TTL 7400 or 7402 gates connected as inverters. Measure the period of oscillation and calculate the average propagation delay of the gate. Compare your value to the manufacturers data. You might also repeat this experiment with a 4000 series CMOS NAND or NOR gates for comparison. end

The ring oscillator will work with any odd number of gates cascaded together with the output of the last gate feed back into the input of the first gate. The period of oscillation is then $2NT_D$ where N is the total number of gates. The ring oscillator can also be extended to make a general purpose oscillator by inserting an RC delay as shown in figure 9.11. Note that the frequency of oscillation is NOT precise (TTL does not specify a voltage threshold for switching) and may drift with time and temperature but this is compensated for by the simplicity of this oscillator. If you need a precise frequency then you should use a crystal oscillator as discussed in the next chapter.



C	approx. freq.
150 pf	5.5 MHz
1000 pf	1.4 MHz
0.01 μf	150 KHz
0.10 μf	17 KHz

Figure 9.11: Simple TTL oscillator.

9.4 D Flip-Flop

Several types of flip-flops with inputs that only have an effect when they are *synchronized* together have been developed. The simplest example is the *D flip-flop* as shown in figure 9.12.

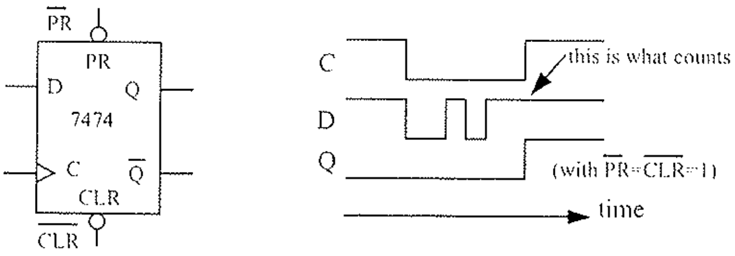


Figure 9.12: D flip flop.

It has a single logic-level input, D, whose state determines the flip-flop's action at a time determined by the clock input C. To be specific, consider the 7474 (all 74xx74's function the same), a dual D-type flip-flop in the TTL family (the 4013 is the CMOS version). Refer to the function table for this flip-flop in the TTL data book. You will see that the D input produces its effect only upon the positive-going edge (i.e. rising edge) of the clock signal, C (the clock signal may sometimes be labeled C, CK or CLK). This edge is symbolized by the notation ↑. There is a specified minimum required "setup" time (20 ns) during which D must hold a constant value prior to the rising edge of the clock. However, in our applications this setup time is so short that it does not need to be referred to explicitly all of the time. On the rising edge of the clock the output Q takes the value of D. D is said to be a synchronous input because its effect is synchronized with the clock.

The 7474 also has a pair of *direct* or *asynchronous* inputs, whose action is similar to R and S on the R-S flip-flop. Their names in this case are CLEAR (CLR) and PRESET (PR). These inputs have an *overriding* effect, providing a way of enforcing a desired state of the flip-flop regardless of what else may be going on, and without reference to the clock. A common application is for initialization of a system, either just after turning on the power or to return the system to a known state. The function table for the 7474 is shown in table 9.4. X="don't care" means that the value has no effect on the output.

When you use a certain type of flip-flop, you must first determine from its function table to what

$\overline{PR}$	$\overline{CLR}$	C	D	Q	$\overline{Q}$
0	1	X	X	1	0
1	0	X	X	0	1
1	1	↑	0	0	1
1	1	↑	1	1	0
1	1	0	X	Q_{old}	$\overline{Q}_{old}$
1	1	1	X	Q_{old}	$\overline{Q}_{old}$

Table 9.4: Truth table for the D flip-flop.

part of the clock waveform it is sensitive[1]. Then you must also inspect all the asynchronous inputs to see what their "rest" states are. In many cases these direct or asynchronous inputs are active-LO as indicated by a bubble on the block diagram. However, manufacturers are not always consistent in this regard. There are traps here for the unwary. Study the truth table (included in the specifications) carefully to discover which signal level is "active" and which is "standby".

There is an alternative notation used to describe the action of a sequential circuit. State Q_n is taken to be the state just following the n^{th} clock pulse. The next clock edge puts the flip-flop into state Q_{n+1}. For the D flip-flop the function table is then given in figure 9.13.

D_n	Q_{n+1}
0	0
1	1

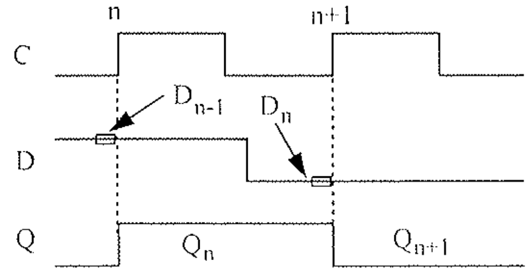


Figure 9.13: Time dependence of the D flip flop.

This can also be represented in equation form as:

$$Q_{n+1} = D_n \tag{9.1}$$

The D flip-flop holds one bit of data from the end of one clock period through the next period, as illustrated. The symbol "D" can be read to stand for "delay" or for "data."

$\boxed{\text{Exp. 9.5}}$ Study the action of one section of the type 7474 or 74LS74 dual D-type flip-flop. Supply D, $\overline{CLR}$, and $\overline{PR}$ from slide switches, and use a debounced push button pulser for C

[1] Edge-sensitive clocking (either ↑ or ↓) is the most convenient. However some devices use the DC clock level, as indicated by the notation drawing a whole square pulse in the function table. In that case the effective setup time extends back to the preceding clock transition, and you cannot permit the data inputs to change during this time without risking an unpredictable result. Fortunately these inconveniently clocked devices have been almost entirely replaced by edge-sensitive ones.

J_n	K_n	Q_{n+1}	Comment
0	0	Q_n	No change
1	0	1	Set
0	1	0	Reset
1	1	$\overline{Q}_n$	Toggle

Table 9.5: Truth table for the JK flip-flop.

Q_n	Q_{n+1}	
0	J_n	(regardless of K)
1	$\overline{K}_n$	(regardless of J)

Table 9.6: Excitation table for the JK flip-flop.

(=clock). Determine which edge of the clock signal triggers the flip-flop, and work through all combinations of the published function table to verify the results. $\boxed{\text{end}}$

9.5 J-K Flip-Flops

This type of flip flop is closely related to the R-S latch. However, it is clocked and it makes intelligent use of the previously "not-allowed" condition in which both R and S were simultaneously active (R=S=1). The two inputs are renamed J and K (arbitrarily). J is similar to S (set) and K is similar to R (reset). When J=K=1 (the previously disallowed state in the RS flip-flop) the JK toggles its output (i.e. the output goes to the opposite state). The truth table for the JK flip flop is shown in table 9.5.

The action of the JK flip-flop can also be represented in equation form as:

$$Q_{n+1} = J_n\overline{Q}_n + \overline{K}_nQ_n \tag{9.2}$$

Although it is easy to remember the J-K function table by its resemblance to R-S, another presentation may shed new light. This is the *excitation table*, (table 9.6) in which results are listed according to the preceding state of the flip flop rather than according to the inputs.

A *toggling* flip-flop, sometimes called a T flip-flop, goes through one cycle of states for every *two* clock pulses. It counts by twos and can be used as a binary counter or frequency divider. In the circuit shown in figure 9.14, the flip-flop is forced to toggle by wiring J=K=1. The "clock" input is used as a signal source, not a synchronizing input.

Note that, even if the input waveform does not have a 50% duty cycle, the subsequent waveform (at Q) has HI and LO sections of strictly equal duration, provided only that the input maintains a constant pulse spacing. A toggle or T flip-flop is thus a good way to produce an accurate 50% duty cycle. (Note: duty cycle is defined as the ratio of the time high to the total time in one period.)

TTL implementations of the JK flip flop usually have an inverted clock input (74xx70, 74xx73, 74xx76, 74xx78, 74LS109, 74LS113, 74LS113, 74LS114) and a preset and clear (PR and CLR as on the 7474). Some also have additional gating on the input. The standard TTL and LS TTL versions of the

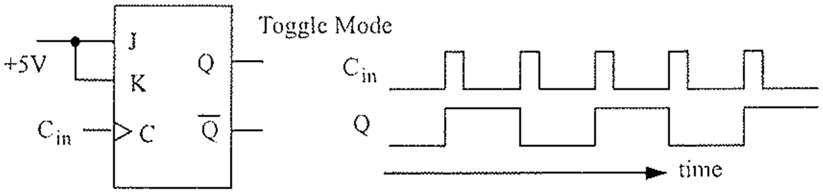


Figure 9.14: The JK flip flop connected to toggle.

JK flip flops differ in one important aspect. The standard series is all master-slave and the LS series is all edge triggered. A master-slave flip flop's input are sensitive during the whole time that the clock is high but the outputs only change at the falling edge (with an inverted clock input). This is signified by drawing a whole square pulse in the function table. Generally speaking edge-triggered devices are easier to use than master-slave devices. The symbol for the 74LS76 JK is shown in figure 9.15 (note that it has an active low clock, meaning its output changes on the falling edge of the clock).

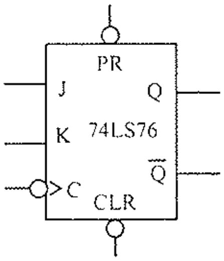


Figure 9.15: TTL JK flip flop 74LS76.

9.6 Synchronous Systems

The flip-flop has been shown to have *memory*. It can accept instructions at one time and hold the resulting information for use at a later time. Opening up the time dimension in this manner is an extremely significant step. Circuits can then be designed to process information *sequentially*, with the significance of various steps depending on their time of occurrence. Taking advantage of this idea, remarkably versatile sequential systems can be built. When such a system has the capability of changing its program of actions in response to external commands, or even as a consequence of the outcome of previous internal data handling, it is called a *computer*.

The picture of large arrays of circuit elements performing logic operations in a sequential manner raises the problem of *synchronization*. This is indeed a vitally important question. In most large systems the elements are synchronized with the help of a central *system clock*. Successive time slots (and sometimes subsections of slots) are defined by the cycles of this clock, and all events in the system are tied to this rhythm.

The R-S flip-flop discussed at the beginning of this chapter has the feature that it responds *immediately* to its input commands. These commands can thus produce action at any time they choose and

n	Q_1	Q_0	D_1	D_0
0	0	0	0	1
1	0	1	1	0
2	1	0	0	0
3	0	0		
4	0	1		

Table 9.7: Sequential values of the circuit in figure 9.16. n is the number of rising edges of the clock. The values repeat after n=2.

are called *asynchronous* (or direct) inputs. Supplementary gates must be added to the R-S flip-flop to synchronize it to the clock. The D and JK flip flops do this synchronization with their clock input. Large arrays of D and JK flip flops can thus be synchronized to a central clock

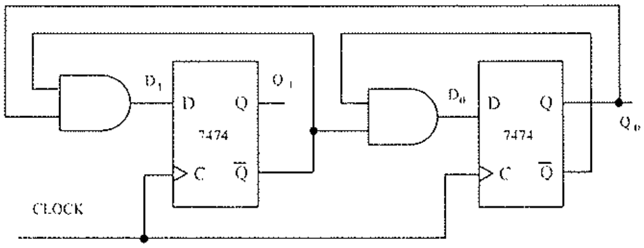


Figure 9.16: Simple synchronous circuit using two edge triggered D flip flops.

As an example of a simple synchronous circuit consider the circuit in figure 9.16. Note that both flip-flops are driven from the same clock. To find out how this circuit evolves in time, first find a Boolean expression for the two inputs (i.e. these control the outputs):

$$D_0 = \overline{Q}_1 \cdot \overline{Q}_0 \qquad (9.3)$$

$$D_1 = \overline{Q}_1 \cdot Q_0 \qquad (9.4)$$

Whenever there is a rising edge on the clock then Q_1 and Q_0 take the value of D_1 and D_0 respectively. Make a table of sequential values as in table 9.7. Start with an initial value of Q_1 and Q_0 at n=0 (assume both 0 here) where n is the number of rising edges of the clock . At each time step figure out D_1 and D_0 and then transfer these values to Q_1 and Q_0 on the next rising edge of the clock. This circuit counts up from 0 to 2 and then repeats (assuming that Q_0 is the LSB or least significant bit). This circuit can be called a divide-by-three circuit because the frequency of Q_1 is one third the frequency of the clock.

Integrated-circuit technology has produced a remarkable amount of progress with sequential circuits. A single IC package can now contain large arrays of flip-flops and logic gates. IC's are sometimes divided into different classes depending on their level of complexity:

SSI Small Scale Integration = several gates and flip flops

MSI Medium Scale Integration = several tens of gates and flip flops
= multiplexers, decoders, shift registers and counters (in the next sections)

LSI and VLSI (Very) Large Scale Integration
= hundreds or thousands or more of gates and flip flops
= microprocessors and other highly complex devices

Fortunately you do not need to understand all of the inner workings of complicated IC's to use them. Usually the manufacturer will supply a functional description that is sufficient to use the IC with only a general idea of how it works internally and an understanding of the basic principles discussed here. From here on, circuit function and not internal structure will be of interest to us.

9.7 Registers

A group of flip-flops that handles several bits of information is called a *register*. For example, the bits can represent a binary number in *parallel* notation with all digits available simultaneously, each on a separate terminal or wire. The 74LS374 is an example of a complete 8 bit latch or storage register in one 20 pin IC (see figure 9.17. There are eight data D-type inputs (D1 through D8) with corresponding outputs (Q1 through Q8). On the rising edge of the clock CLK the value of each input is clocked into the register. The outputs are tri- state for use in a bus organized system. The output control OC controls the state of the outputs. If OC=1 then all outputs are in a high impedance state. When OC=0 the outputs take on a specific value (0 or 1).

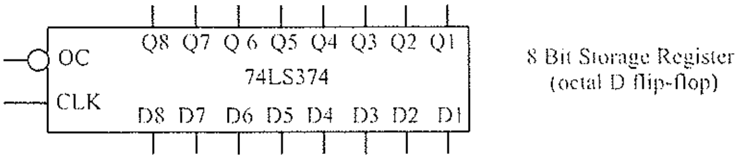


Figure 9.17: Storage register.

A few other examples are the 7475, 74173, 74174, 74175 or 74273 (see the TTL data book for details). Alternatively, the bits may represent independent but loosely related data; in that case they may be called *flags*, which can be set or cleared to signal certain conditions within a computer. An example is the *processor status register* present in most computers, whose flags concern such conditions as an overflow, a zero result, a negative result, etc. A register's function may be primarily for memory. Computers have several working registers in which numbers are stored temporarily while other procedures are carried out. A computer may also need input and output registers to facilitate data exchange with the outside world, whose marching orders may not match those of the computer's own clock.

Information can also be *processed* within a register, as for example in a *shift register* (figure 9.18). In this device, upon receipt of a shift command from a common clock, each stage sends its information on to its neighbor. Depending on the interconnections, shift registers can shift-left or shift-right, and in some IC versions the direction of shift may be chosen with a logic signal applied to a selector terminal.

The individual stages of the shift register can be type D or J-K. There is a vast amount of digital data transmitted on the internet, most of which is sent serially, one bit at a time. A shift register may be used to convert parallel data (many simultaneous bits at one time) into serial data (one bit at a time) and back again to send over the internet. Also inside a computer shift left or right represents a multiplication or division by two.

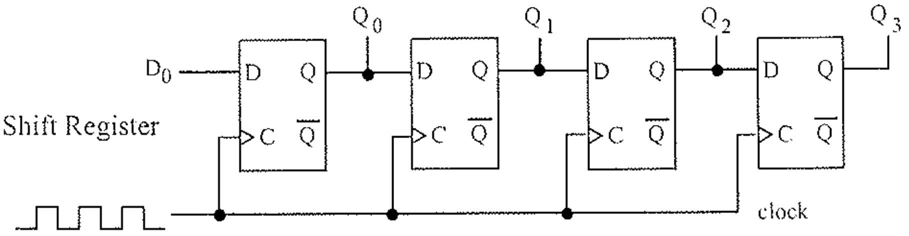


Figure 9.18: Shift register.

A D type shift register is shown in figure 9.18. This circuit has a four-stage shift-right register made of D-type stages. At each occurrence of the clock, the data shifts one step to the right: $Q_{3,n+1} = Q_{2,n}$, $Q_{2,n+1} = Q_{1,n}$, and $Q_{1,n+1} = Q_{0,n}$. $Q_{3,n}$, the information in the right most stage, is lost in the process. New data enters from the left: $Q_{0,n+1} = D_{0,n}$.

The most obvious application is in arithmetic processing of binary data, or to reposition bit-pattern information. In binary a shift right is equivalent to a dividing by 2 and a shift left is equivalent to multiplying by 2 (with D0 above being the most significant bit). There is also the important feature that the information entered at D0 is delayed by N clock cycles (N=number of stages) before it emerges at Q3. This has several possibilities for processing data during sequential time slots. Also if Q3 is connected back to D0, the four bits of data *recirculate* within the register. At any selected terminal, such as Q3, the four bits appear in turn at each edge of the clock and can be picked off by synchronization with the clock.

Finally, imagine that the register is loaded from four *parallel* data lines by use of the individual PRESET and CLEAR inputs of each flip flop. This data can then be right-shifted three times to appear, in *serial* form, at Q3 (one digit per clock period, right-most digit first). Alternatively, the register can be filled with serial data shifted in from the left-end input, D0. When all stages are filled, the data are available in parallel form at Q0...Q3. In this way shift registers are used for serial/parallel conversion. The next experiment will investigate how the 7495 shift register works. This IC is an MSI device and combines many of the features discussed above. A compressed diagram for this IC is shown in figure 9.19. QD is both a parallel output and the serial output. The SERIAL-IN input is the D input for the QA flip flop. During a normal shift-right operation QB will get QA, QC will get QB and QD will get QC. During a parallel load operation QA-QD will take the value of the inputs A-D.

Exp. 9.6 Study the operation of the 7495 4-bit shift register. This is a versatile and quite complicated device. It will be used only for shift-right operations (shift-left requires some external interconnections which will be omitted for simplicity).

The description of the inner workings of the 7495 given in the TTL data book, is rather

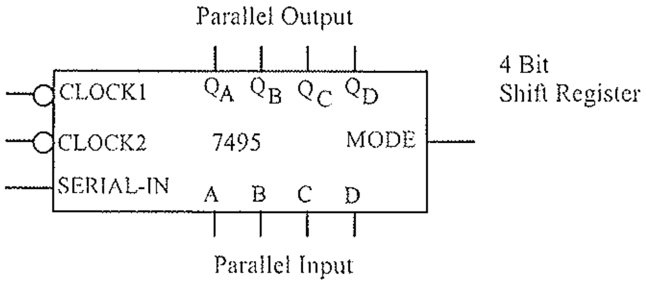


Figure 9.19: TTL 4-bit shift register with parallel load in a single IC.

forbidding at first. The 7495 has two D-type inputs for each stage. Each input is selected with two AND gates and one OR gate and is activated by a MODE CONTROL line. (Note that the signal names may vary from one manufacturer to another, you may have to translate these names.) The first input is connected internally for shift-right operation in conjunction with MODE=0 and the ↓ transition of $\overline{CLOCK1}$. This is just what the diagram in figure 9.18 shows. The second D input, synchronized by MODE=1 and ↓ of $\overline{CLOCK2}$, permits parallel loading of the flip-flops. This convenient parallel loading takes the place of the PRESET and CLEAR direct-load terminals described in the text above.

The MODE CONTROL must be LO to enable the shift-right operation with $\overline{CLOCK1}$, and it must be HI to enable the parallel-load with $\overline{CLOCK2}$. The two clocks can thus be driven in parallel, with the MODE CONTROL determining which operation is desired.

Many of the lines in the function table (in the TTL data book) merely go to show that the data in the register remain undisturbed by a large variety of input combinations. Thus you can afford to concentrate on the parallel load (MODE=H, $\overline{CLOCK2}$ =↓ and the shift right (MODE=L, $\overline{CLOCK1}$ =↓ portions of the table.

Wire up the 7495, using the four logic data switches for the parallel inputs A, B, C, and D. Drive the two clocks in parallel from one push button pulser, and use the other pulser to supply the serial input. You will have to connect the MODE CONTROL to +5V or ground by means of a movable wire, as required.

Load some arbitrary parallel data into the register and then watch the lamp on QD as the data are right-shifted and appear in serial form. Load the same data serially and observe them appearing in parallel on the lamps monitoring QA...QD. $\boxed{\text{end}}$

9.8 Counters

In section 9.5 the J-K flip-flop when set to toggle was shown to *divide-by-two*. Such stages can be cascaded to provide division by any power of 2. Each successive stage operates at a lower frequency, so the maximum counting frequency is determined primarily by the first stage. Sometimes it is desirable to choose a faster logic block for this stage (or the first few) than for the remainder. When each successive stage divides by two the combined output of all flip flops when viewed as a single binary

base-10	base-2	base-10	base-2
0	0000	8	1000
1	0001	9	1001
2	0010	10	1010
3	0011	11	1011
4	0100	12	1100
5	0101	13	1101
6	0110	14	1110
7	0111	15	1111

Table 9.8: Conversion from base 10 to base 2.

number forms a binary count sequence. A 4 bit *ripple counter* is shown in figure 9.20.

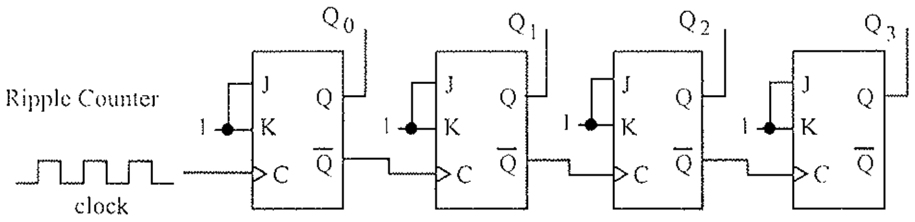


Figure 9.20: Ripple counter (asynchronous counter).

Q0 is the least significant bit (LSB) and Q3 is the most significant bit (MSB). The binary or base-2 counting sequence is given table 9.8.

A 4-bit binary counter can also be referred to as a modulo-16 counter. When it reaches its last value (15_{10}=1111_2) it returns to all 0's and then repeats the same sequence over again. The 74177 and the 74393 are examples of IC's with a whole TTL ripple counter (the 4020 and 4040 are CMOS ripple counters).

Since each stage is driven by the output of its predecessor, the propagation delays add cumulatively. If you picture the transition between the states $Q_3 \ldots Q_0$= 0111 and $Q_3 \ldots Q_0$= 1000, you see a domino effect–the stages do not change simultaneously. The transitions ripple through each successive stage giving rise to the name ripple counter. Thus there are brief intervals where such combinations as 0110, 0100, etc., are present; such unwanted combinations, caused by propagation delays (or other timing errors), are known as *glitches* and can cause errors. In applications where specific intermediate combinations are recognized and perform defined functions, it is preferable to use a *synchronous* counter instead of the ripple counter. However, for frequency division or simple event counting the ripple counter has the advantages of simplicity and high speed.

$\boxed{\text{Exp. 9.7}}$ Design and build a 4 bit (modulo 16) ripple counter using 7474 or 74LS74 D flip flops.

a) First study the truth table for the D flip flop and devise a way to connect it so that it toggles on every rising edge of its clock. (Its output changes to the opposite state on every rising edge.) Connect this circuit and apply a 100kHz clock and observe the clock input and Q output on the scope. This circuit will be called a T flip flop.

b) Next connect four of these T flip flops (two 7474 IC packages) into a four bit ripple counter (that counts up). Apply the 4 bit output to the four LED's on the Digi-designer and connect the clock input to a debounced switch. Manually push the switch and verify that your counter counts up every time the switch is pushed. Make a truth table of the LED outputs for a least 20 pushes of the switch.

c) Connect a switch that is not 'debounced' and observe the output of the LED display. Explain your observations.

d) How would you make the counter count down? end

Synchronous counters use the same clock to drive all their stages, and provide coded inputs to each stage which correctly bring it to the desired next state. (The coding can become quite complex.) For example, a simple synchronous binary counter might be constructed as shown in figure 9.21.

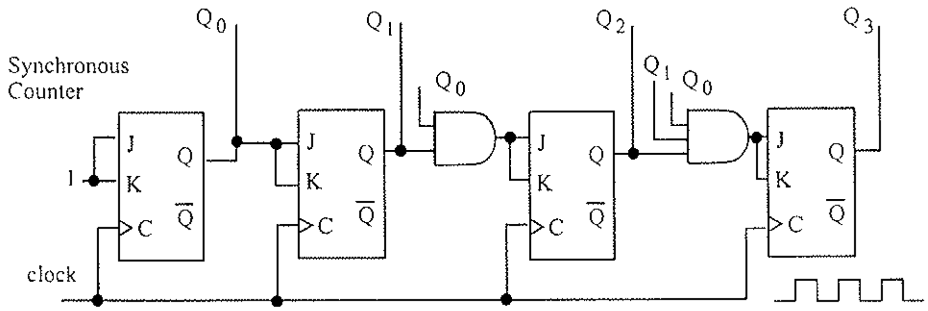


Figure 9.21: Synchronous counter.

The first stage toggles always. The second stage toggles when Q0 = 1; the third when both Q0 = 1 and Q1 = 1, and so on. Each stage toggles when all preceding stages are 1. Thus the N^{th} stage needs an AND gate with N-1 inputs, and there are many interconnections. The operating frequency is limited by the requirement that the AND gate for each stage must have arrived at the correct output by the setup time preceding the next clock. The propagation delay of the AND gate is added to that of the flip-flop. However, beyond increasing complexity, the counter's speed remains unchanged as the number of stages grows. Because the input to each stage comes all the way from the earliest stages of the counter, the system is said to use *look-ahead* generation.

Complete counters are available as single IC's. The 74161/74LS161 is an example of a synchronous four bit binary counter. (It is related to the 74160 that counts in BCD which stands for binary coded decimal or modulo-10.) A functional symbol for this counter is shown in figure 9.22. In addition to the normal clock input it has four additional controlling inputs. There are two active high enable inputs (ENABLE-P and ENABLE-T) that must both be HIGH to count, an asynchronous active-low clear (CLR) that resets the output to 0000, and a synchronous active low parallel load (LOAD). The four binary outputs (Q_D, Q_C, Q_B, Q_A) count up by one on each rising edge of the clock. A ripple carry out (RCO) goes HIGH on the last state (1111) of the counting sequence. RCO allows multiple IC's to be cascaded by connecting the RCO output of one IC to the ENABLE inputs of the next IC for counter lengths of more than four bits. QD is the most significant bit and QA is the least significant bit. Input A corresponds to output QA etc. QD through QA advance by one for each rising edge of the clock if ENABLE-P, ENABLE-T, CLR and LOAD are all high. IF LOAD=0 then QD through QA take the value of A through D on the next rising edge of the clock (assuming CLR=1). If CLR=0 then all outputs go to zero regardless of what else happens. Please refer to the TTL data book for a more complete description.

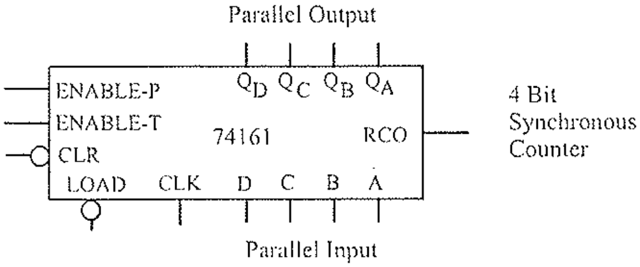


Figure 9.22: Four bit synchronous counter IC.

Exp. 9.8 Design and build a modulo-7 counter using the 74161 or 74LS161 synchronous 4 bit binary counter (i.e. the circuit should count 0 through 6 and then repeat this sequence over and over). Hint, if you use the synchronous parallel load input you only need one two-input gate to sense the correct end state. First connect the four output bits QD-QA to the LED's on the Digi-Designer. Drive the clock from one of the debounced push buttons and verify that your circuit counts properly. Next connect the clock input of the counter to the 100kHz clock on the Digi-Designer and measure the frequency and duty cycle of the next most significant bit QC. Correlate this with the clock and the function of this counter.

Note that some of the old style Digi-designers have a poor clock with a rise time of 100nS. TTL requires a 10nS rise time to work properly. If you have one of these then you should buffer the clock by passing it through a TTL gate before applying it to the counter. end

Counting patterns other than simple binary counting can be implemented by suitable choice of gating combination for each stage. For example the 74191 and 74193 can count either up or down (controlled with a single input line). The more commonly needed arrangements are available in integrated-circuit form, so there isn't much point in going into detailed design procedures here. Note, in particular, the availability of very flexibly coded counters known as *binary rate multipliers* (e.g., 7497, 74167, 4089, 4527).

9.9 Initialization

Complicated systems must be started on their careers in some controlled way. For example, when power is first applied to the circuit, most of its registers "power-up" in arbitrary states. The design of the system may not permit it to progress naturally from one or more of these random combinations, and it may find itself *locked out* of its intended functional cycle.

Power up initialization is frequently explicitly designed into a large system. A small circuit is especially designed to produce a master reset pulse some time after the supply rails are first turned on. It is assumed that all other components have had time to settle down; the reset pulse is routed appropriately to all places where a specific initial state is to be enforced. An alternative approach calls upon the user to press a RESET button whenever the system is first turned on. This reset button may also extricate the system from some locked-out state which it has reached because of a malfunction, or because someone produced an accidental transient while probing around with a voltmeter or oscilloscope.

It is preferable, at least in smaller circuits, to arrange matters by design in such a way that there are no locked-out states. Even if some combinations do not occur in normal service, the system's logic is so designed that it can progress away from them if ever they should accidentally arise. The possibility of locked-out states greatly complicates fault diagnosis and servicing. A system may appear to be broken, yet work perfectly well after some minor change which required the power to be turned off. If you then conclude that the change was related to the fault, you are off on a frustrating chase.

9.10 Practice Problems

[1] Using a positive edge triggered JK flip-flop and any of the gates discussed so far, design a circuit that performs like a positive edge triggered D flip-flop.

[2] The circuit shown in figure 9.23 uses a 7474 positive edge triggered D flip flop.

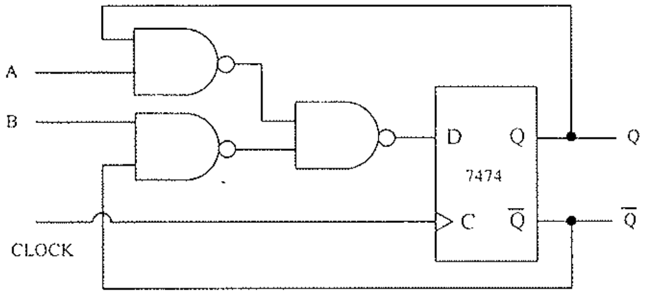


Figure 9.23: Problem 9-2.

a) Write a Boolean expression for D in terms of A. B. Q, and $\overline{Q}$.

b) Copy the following table and fill in the remaining values for Q_{n+1}. and $\overline{Q}_{n+1}$. A_n and B_n are the values just before the n^{th} rising edge of the clock and Q_{n+1} and $\overline{Q}_{n+1}$ are the values just after the n^{th} rising edge of the clock.

A_n	B_n	Q_{n+1}	$\overline{Q}_{n+1}$
0	0		
0	1		
1	0		
1	1		

c) This circuit, taken as a whole, is very similar to another type of flip flop discussed in in this chapter. What type of flip-flop is this similar to? Identify inputs A and B in terms of this other flip-flop.

[3] The circuit shown in figure 9.24 uses two 7474 positive edge triggered D flip flops. Q_0 is the least significant bit (LSB) and Q_1 is the most significant bit (MSB).

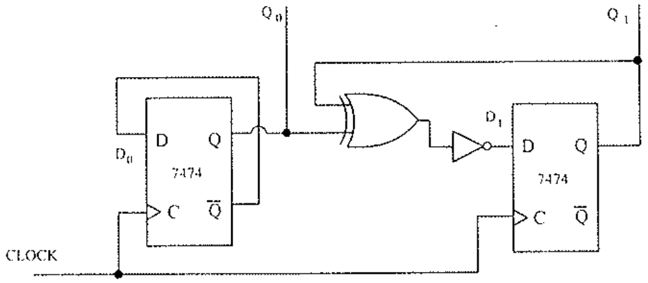


Figure 9.24: Problem 9-3.

a) Write a Boolean expression for D_0, and D_1 in terms of Q_0, and Q_1.

b) Copy the following table and fill in the remaining values for Q_1 and Q_0 after each rising edge of the clock C_{in} labeled n in the table. Note that the order of Q_0 and Q_1 is opposite to that in the schematic. (You may find it easier to add two more columns for D_1 and D_0 as calculated from part a and transfer these down one line to Q_1 and Q_0 at each clock edge.)

n	Q_1	Q_0
0	0	0
1		
2		
3		
4		
5		
6		
7		

c) If the clock has a frequency of 1 MHz, what is the frequency and duty cycle of Q_1?

[4] Complete the timing diagram for the circuit shown in figure 9.25. A, and C are inputs and Q is the output. Copy the diagram onto your homework and fill in the portion for Q (indicate if it is a 1 or 0 vs. time). Clearly indicate how A, C and Q are related in time (Q starts with a 0).

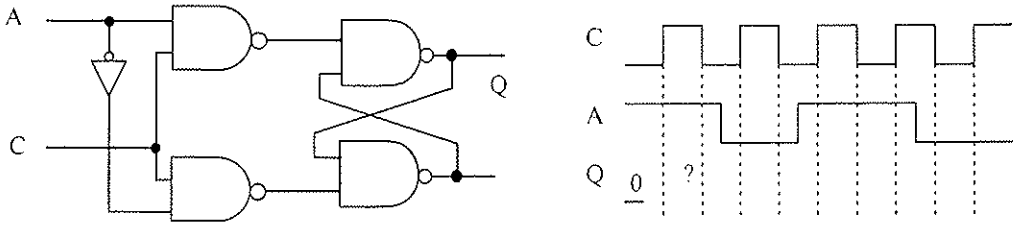


Figure 9.25: Problem 9-4.

[5] The circuit shown in figure 9.26 uses two 7474 positive edge triggered flip flops. (Hint, this is also a divide by three counter.)

a) Write a Boolean expression for D_0, and D_1 in terms of Q_0, and Q_1.

b) Make a table, stating the state of Q_0, Q_1 after the rising edge of each successive clock pulse. Assume that it starts with $Q_0 = Q_1 = 0$.

c) Make a timing diagram showing the clock, Q_0 and Q_1.

d) If the input frequency of the CLOCK is f_{in} what is the output frequency of Q_1, f_{out}?

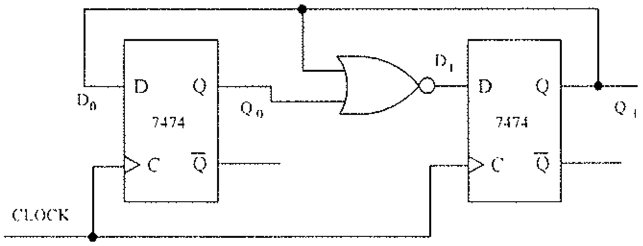


Figure 9.26: Problem 9-5.

Chapter 10

TIMING AND TRANSDUCERS

10.1 RC Timing

The simplest way to obtain a specific time duration is from the *time constant*, τ=RC, of an RC circuit using the low-pass or the high-pass configuration (see chapter 1). In each case, a *threshold* level, V_{thr}, needs to be defined between the initial voltage $V_I = V_1$ and the final voltage $V_F = V_2$ of an input step-function input (as in fig. 1.28 and equation 1.22). The output of the low-pass filter crosses V_{thr} after a time delay T relative to the input. The high-pass filter remains above V_{thr} for a time duration T. T is related to τ by a logarithmic factor, which depends on V_{thr}, V_1 and V_2:

$$\text{low-pass:}\;\; T = \tau \ln \frac{V_2 - V_1}{V_2 - V_{thr}} \qquad \text{high-pass:}\;\; T = \tau \ln \frac{V_2 - V_1}{V_{thr}} \tag{10.1}$$

To avoid using the flat, final portion of the exponential waveform, V_{thr} should not be too close to V_F for the low-pass filter, or too close to zero for the high-pass. To avoid obtaining a very short T relative to τ, V_{thr} should not be too close to zero in the low-pass case, or to $V_2 - V_1$ in the high-pass. Usually V_{thr} is best chosen to be about midway between the two levels of the input step-function.

An RC circuit also has a second exponential response to the trailing edge of the input waveform. This second exponential can usually be regarded as the *recovery time* of the circuit, a time which must elapse before the next timing operation can start from standard initial conditions. When the recovery time must be much shorter than the delay time, a nonlinear device (diode, or transistor switch) can be used to reduce τ on the trailing edge and obtain a faster recharge of the capacitor.

The high-pass filter produces an output of reverse polarity during the recovery time, which is often a nuisance. Usually the low-pass arrangement is easier to work with. However, if the DC levels of the input step and the exponential timing waveform must be different, then the high-pass filter is more natural.

V_{thr} is best defined by a comparator circuit with hysteresis (see the Schmitt trigger in chapter 2). Then the fact that the exponential waveform crosses the critical level quite slowly is not as important. If this slow waveform were applied directly to a logic gate, there is danger of multiple transitions on the output (due to noise) or oscillation (in some logic families).

10.2 The 555 Timer IC

This versatile IC (in an 8 pin DIP package) can serve several different kinds of timing functions and is good to illustrate many of the important timing principles. The 555 contains both analog and digital components as shown in the schematic in figure 10.1. It gets its name from the three 5K resistors that form a voltage divider to supply two different threshold voltages to two op-amp voltage comparators A1 and A2.

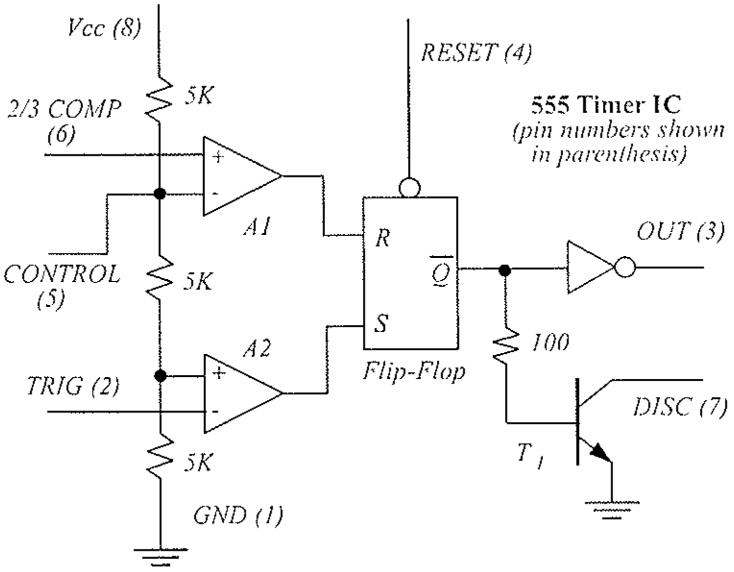


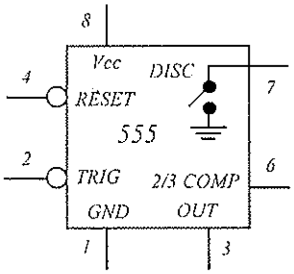
Figure 10.1: Internal configuration of the 555 timer IC.

The basic timing element is usually a low-pass R-C circuit which is connected external (i.e. not shown above) to the IC (see examples that follow). The saturating switch transistor T1 can be used to discharge the timing capacitor at the end of each cycle.

The 555 contains a flip-flop which controls its cycle and generates the ultimate output waveform (OUT on pin 3). When the flip-flop is RESET (OUT is low), T1 is closed and DISC (discharge) is connected to ground. If DISC is connected to the external capacitor its discharge becomes very rapid and non-exponential, being limited only by the current carrying capacity of the transistor. When the flip-flop is SET (OUT=high) then T1 is open and DISC is floating.

The inputs to the comparators, A1 and A2, RESET or SET the flip-flop. These will typically be connected to the external timing RC circuit. When pin 6 (2/3 COMP) is higher than $\frac{2}{3}V_{cc}$ then the flip-flop is reset (OUT=low). When pin 2 (TRIG) is below $\frac{1}{3}V_{cc}$ then flip-flop is SET (OUT=high). Note that the inputs to A1 and A2 are connected to the opposite polarity inputs.

It is inconvenient to draw this whole schematic every time so the abbreviated symbol shown in figure 10.2 will be used (pin 5 is normally left unconnected).



Short hand symbol for the 555 IC Timer

Figure 10.2: Condensed symbol for the 555 Timer IC.

10.3 Monostable Operation of the 555 (One Shot)

A timing circuit with two states (i.e. digital) that can rest in one state indefinitely, but will return from the other state spontaneously after a certain time, is called *monostable.* If it is forced into its temporary state by an *external trigger*, it generates a single pulse of fixed duration and then relapses into its rest state. A monostable can also be called a *one-shot.* A one-shot made from a 555 is shown in figure 10.3. R_A and C form the external timing element.

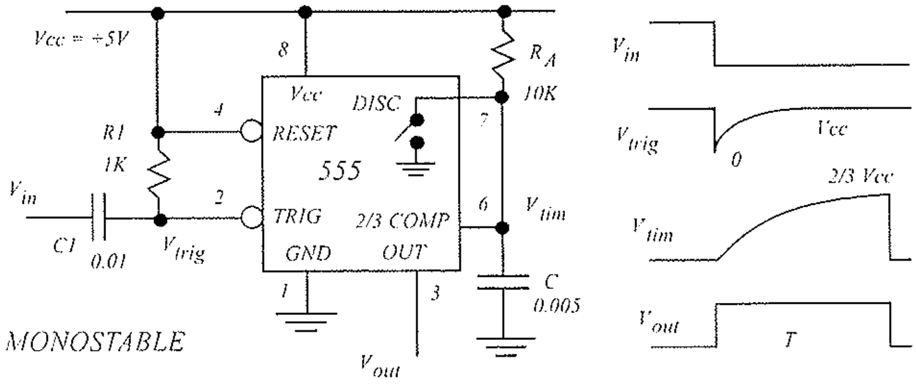


Figure 10.3: The 555 IC connected as a monostable (or one-shot).

When there is a high to low transition on the input V_{in}, TRIG momentarily crosses $\frac{1}{3}V_{cc}$. This sets the flip-flop and OUT goes high, causing DISC to open. V_{tim} is then free to rise and does so until it crosses $\frac{2}{3}V_{cc}$ at which point the flip-flop is reset and OUT goes low. The standard capacitor RC charging equation (see chapter 1) can be used to show that the output (OUT) stays high for a time

$$T = 1.1R_AC \tag{10.2}$$

(note that this is independent of the value of V_{cc}).

$\boxed{\text{Exp. 10.1}}$ Build a one-shot with the 555, using the circuit shown in figure 10.3. The 555 can operate with V_{cc} anywhere in the range 4.5V to 16V. Use the +5V supply of the Digi-Designer and drive the circuit with a 1kHz 5V square wave from the function generator.

To obtain a suitable, short triggering signal, the input square wave at V_{in} is sent through a 10μs high-pass filter returned to +5 V. Some Digi-Designer clock outputs do not go to the required +5V (i.e. they only have a TTL level) so use the function generator with the output set to $+5V_{PP}$ (instead of the Digi-Designer clock signal). The negative spikes fall briefly below the trigger level of $\frac{1}{3}V_{cc}$ Each negative spike should trigger the one-shot and produce a pulse at V_{out} (pin 3). Since DISC is directly connected to the timing capacitor C, the discharge (of the timing capacitor) should be rapid, permitting the 555 to be retriggered soon.

Record all the waveforms in the circuit and account for the duration of the pulse. $\boxed{\text{end}}$

A notable inconvenience in the use of the 555 as a one-shot lies in its inflexible trigger requirements. Not only is the downward passage of the trigger signal through the $\frac{1}{3}V_{cc}$ level strange, but the trigger signal can not remain below that level for longer than the desired output pulse! Otherwise the 555 will remain SET (i.e., it will refuse to *time out* in its normal manner). The presence of the $\overline{TRIG}$ signal, as it should properly be called, is overriding.

There is a RESET terminal (pin 4, more properly $\overline{RESET}$) which terminates the output pulse dictatorially and discharges the timing capacitor.

The CONTROL VOLTAGE terminal (pin 5) gives access to the resistor network which produces the $\frac{2}{3}V_{cc}$ threshold voltage. You may apply small perturbations to this terminal and vary the effective value of V_{thr}, giving, in effect, a voltage-controlled time interval. If you do not plan on this then you are encouraged to connect a bypass capacitor from this terminal to ground, so as to minimize the possibility of stray signal pickup into V_{thr}.

10.4 Astable Operation of the 555 (Oscillator)

A timing circuit which cannot remain in either of its states indefinitely is astable. It oscillates and can therefore serve as waveform generator. You have already seen such an *astable multivibrator* constructed from an op-amp Schmitt comparator, with low-pass R-C timing in chapter 2.

The 555 can be made astable by making it trigger itself as soon as V_{tim} falls below the trigger level. $\frac{1}{3}V_{cc}$. Obviously it is desirable to have V_{tim} pass through this level fairly slowly, so as to give the 555 time to trigger, release the discharge switch, and permit V_{tim} to rise smoothly from $\frac{1}{3}V_{cc}$. This implies that another resistor should be included to limit the rate at which C is discharged.

The astable configuration is obtained from the monostable by connecting V_{tim} to the TRIG terminal, and by including another resistor R_B between pin 6 and 7. V_{tim} is now "captured" between the levels $\frac{1}{3}V_{cc}$ and $\frac{2}{3}V_{cc}$ each time it reaches one or the other of these levels it triggers or resets the 555. The time constants of its rise and fall are different. During the rise $\tau = (R_A + R_B)C$, and during the fall $\tau = R_B C$ which can be used to find the high and low times as:

$$\begin{aligned} T_{LO} &= 0.7R_B C \\ T_{HI} &= 0.7(R_A + R_B)C \qquad (10.3) \end{aligned}$$

Thus the ratio of the ON and OFF portions of the waveform can be adjusted, but the ON time is always longer than the OFF.

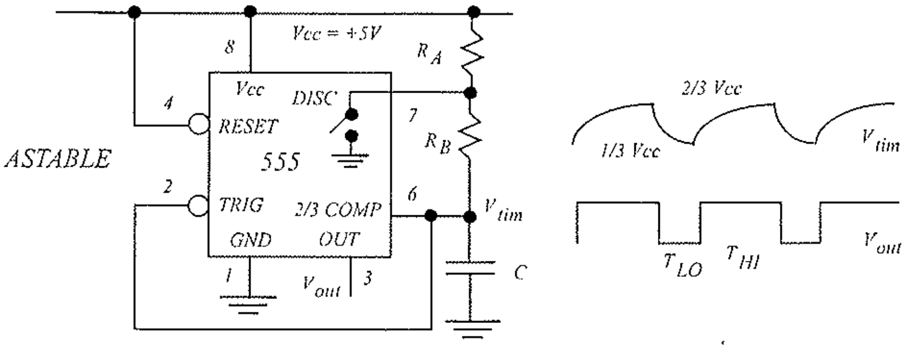


Figure 10.4: The 555 connected in the astable configuration.

Because the 555 does not make available the complement of its output, you need to use an inverter if you want to have a waveform whose HI level was of shorter duration than the LO level. Alternatively, you could connect diodes in such a way as to insert two different resistors R_B for the charging and discharging of C. However, the temperature dependence of the diode voltage drops would then impair the timing accuracy of the 555 which is normally impressive (timing defined to well within 1%).

To limit the dissipation in the discharge transistor, R_A is specified to be no less than 1 kΩ. On the other hand, because input and leakage currents to the DISCHARGE and THRESHOLD terminals are small (fractions of 1 mA), R_A and R_B can be large (a megohm is acceptable).

Exp. 10.2 Build a 555 multivibrator, selecting R_A, R_B, and C to yield (two different circuits):

(a) a square wave of very nearly 50% duty cycle, and frequency 0.5 kHz; and

(b) a negative going pulse (about 10μs long) at V_{out}, with a repetition frequency of about 1 kHz.

Leave the second circuit assembled for use in the next experiment. end

10.5 The TTL Family One-Shots

One of the first TTL-family one-shots was the 74121[1]. It has considerably more flexible triggering than the 555 and is entirely edge-sensitive, so that the input pulses can be longer than the desired output. Both positive-going and negative-going inputs are provided and the output and its complement are made available. The circuit has "rudimentary" timing components, R and C, built onto the chip, so that very short (about 40 ns) pulses can be generated without the help of external components. With external R and C, the one-shot duration can be as long as 28 sec. The positive-going trigger input has Schmitt trigger characteristics. Descendants of the 74121 include the 74221, 74122, 74123 and 74LS221. The x123 and x221 are dual one-shots (two per IC) and the x221 is available in both the TTL and CMOS 74HC series. A schematic symbols for two of these one-shots is shown in figure

[1] In 1969, when the 74121 first came out, it was advertised as *The One-Shot Heard Around the World*, actually a believable claim.

10.5. Each one-shot has a little different triggering set-up that is conveniently represented in logic form. Whenever there is a positive edge at the internal signal labeled T then a single output pulse is delivered at Q (see timing diagram in figure 10.6). The width of this pulse is determined by the external resistor R and capacitor C connected as shown. See the data book to calculate R and C (note that each has a max. and min. allowed range, as given in the spec. sheets). For the 74221 the output pulse width is $T_W = RC \ln 2$. The input labeled B on the 74121 and the 74x221 also has a Schmitt trigger feature that allows it to be driven from a slowly changing input.

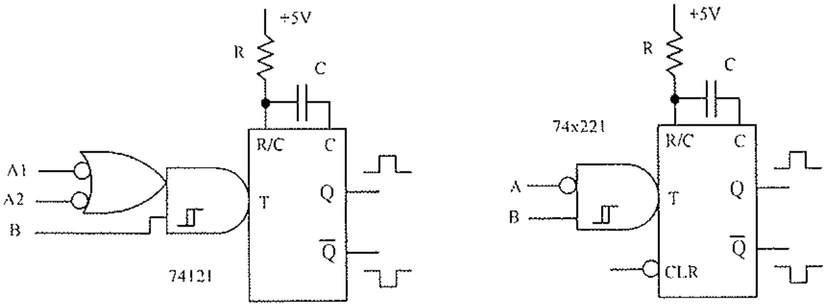


Figure 10.5: TTL one-shots.

Exp. 10.3 Design a pulse delay circuit using a 74LS221 or 74221 dual one shot. Your circuit should deliver a 1 microsec pulse 15 microsec after the rising edge of the input signal. (Hint you may need to use both one-shots in the IC.) Build your circuit and test its performance. You may use the output of the oscillator in the previous experiment or the function generator to drive this circuit.

Inspect the waveforms on both terminals of the timing capacitor, and convince yourself that it is using an RC discharge to get a pulse of a fixed length. **end**

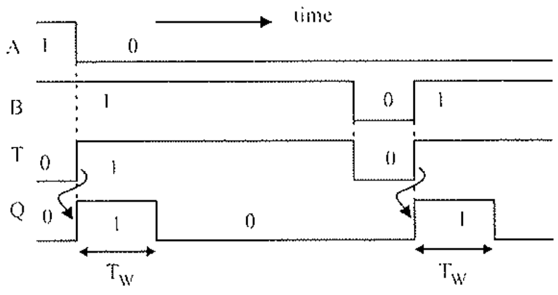


Figure 10.6: Triggering of the 74221 one-shot (CLR is held high).

Some of these one-shots are *retriggerable* and some are *non-retriggerable*. When an input trigger pulse is applied before the output pulse has ended, a retriggerable one-shot begins a new timing interval afresh without even momentarily resetting the output pulse. A non-retriggerable one-shot simply ignores input pulses while the output is active. In both types of one-shots if a trigger pulse arrives during the *recovery* time (directly after the natural end of the pulse) it may be treated in non-standard fashion.

A retriggerable one-shot is frequently used as a missing pulse detector. Suppose a system delivers pulses at a certain minimum rate while everything is functioning normally, but that you need to detect the moment when these pulses for some reason cease to arrive. A retriggerable one-shot, with pulse duration longer than the normal pulse interval, will remain in its SET state continuously while these pulses trigger it. If the pulses stop, the one-shot resets shortly thereafter.

It is tempting to connect two one-shots into a ring, each triggered by the trailing edge of the other. Such a circuit oscillates once it has been started. However, if the two one-shots are ever allowed to relapse into their stable states together then the output of both will remain low. A new starting pulse must be supplied from somewhere to get the system going again. You should avoid using such an arrangement in place of a truly astable multivibrator (as in chapter 9 or the following section).

10.6 Clocks

Digital or logic systems often use a central clock to define their timing and synchronize the components. The frequency stability required of such a clock source depends on the complexity of the system and the nature of its elements. Often a highly stable clock is chosen, even though the system could tolerate quite wide frequency changes. Stable clocks are inexpensive, and their use at least fixes one variable in the system very accurately.

The 555 timer, and such specialized clock sources as the 74124 and 4047, achieve a frequency stability of below 1%. For much higher accuracy, crystal-controlled oscillators are used (the 74124 can also be used with a crystal).

A simple crystal oscillator can be constructed from a CMOS gate which is used as a high-gain inverting amplifier. For this to work, the gate must be biased into its transition region, i.e., the region which is normally avoided in digital logic circuits. It would be hopeless to try to do this with a fixed voltage bias because the sample-to-sample variations of transition level are too large. Instead, negative-feedback is used. The input is connected to the output through a large resistor. Then the gate seeks to return to the condition where input and output are at the same voltage, which occurs roughly in the center of the transition between logic levels (see figure 10.7).

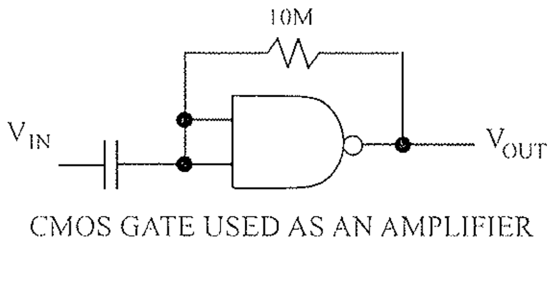


Figure 10.7: CMOS gate as an amplifier

The details of the crystal oscillator depend to a large extent on the properties of the crystal. Near the resonant frequency, a 180° phase change can be produced in conjunction with circuit capacitances and a resistor, R. This produces positive feedback, and the circuit oscillates. The amplitude grows until nonlinearities limit it. The output waveform is strongly non-sinusoidal (see figure 10.8). This sort of rudimentary crystal oscillator falls far short of the stability which can be achieved with crystals. Commercial oscillators are available in convenient IC size packages, which would normally be used when ultimate performance is needed. However, even a very careless crystal circuit is impressively stable. Some experimentation is usually substituted for rigorous design, to get it to oscillate in the first place.

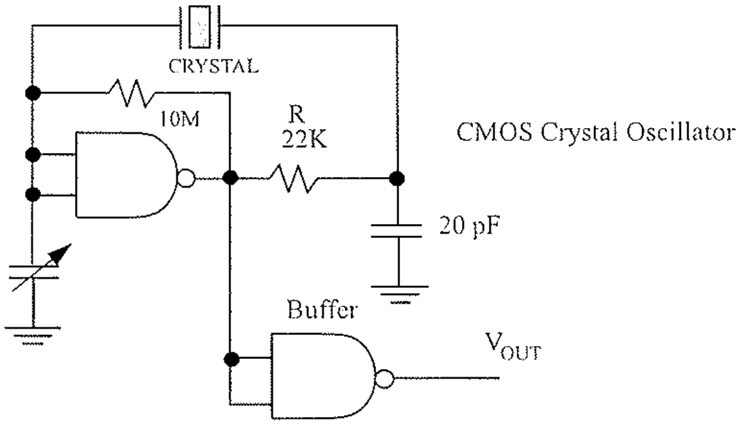


Figure 10.8: Crystal oscillator

10.7 Distance Measurement with Ultrasonic Transducers

As their name implies, transducers transform external physical variables to or from electrical signals. The term *sensor* applies to those transducers that sense physical parameters and convert them into electronic signals, but not the other direction. The term *actuator* applies to transducers that convert electrical energy into some other form of physical energy. A speaker is an actuator but a microphone is a sensor (both are transducers). A large range of transducers exits that can transform almost any physical parameter, such as temperature, pressure, sound, light, magnetic field, etc. to or from an electronic signal. Transducers provide a means of both measuring and controlling physical parameters using electronic devices and are one of the reasons why electronics is so important in physics and engineering. There is not enough time in the semester to study all the different types of transducers. This section will instead concentrate on one particular type of transducer that can transform ultrasonic or high frequency sound waves to and from electronic signals.

A piezoelectric crystal is a crystal that produces a voltage when it is strained (i.e. when the crystal is compressed or expanded). The reverse is also true. When a voltage is applied, it expands or contracts. Quartz (crystalline silicon dioxide) is a common form of piezo but many other types of materials can exhibit this behavior. This change in shape or size is relatively small but is large enough to couple to the air in the form of a sound wave. Piezoelectric crystals can be cut and shaped so that they resonate at audio or slightly high (ultrasonic) frequency (i.e. the human ear can distinguish sound in the approximate range 20 Hz to 20 kHz) and strongly couple to the mechanical vibrations in the air. Ultrasonic transducers are piezoelectric crystals made in such a way as to resonate above the normal audio range (i.e. we cannot hear them) and to couple their vibrational energy to the air. Piezoelectric crystals can also be cut and shaped so that they have a very precise resonant frequency in the MHz range. This resonance can be used to produce accurate stable oscillators as was shown in the last section.

The next experiment uses a pair of ultrasonic transducers to measure the speed of sound as shown in the diagram shown in figure 10.9. The transmitter is driven with a pulsed electronic signal of 40 kHz and produces an acoustic signal of 40 kHz (i.e. a sound wave conceptually similar to the sound from the speaker on a stereo except that it is at a frequency too high to hear). This frequency is fixed by the shape and size of the crystal and your circuit must match it fairly accurately. If the transmitter is sent a short burst of 40 kHz the resulting acoustic pulse will travel through the air. This pulse can be reflected back into the receiver, which is another piezoelectric crystal (transducer) manufactured to match the transmitter. The receiver will convert the acoustic energy in the air into an electronic signal that can be seen on the oscilloscope. The time it takes to travel a distance L and back can be used to measure the speed of sound in air given L or equivalently the distance L given the speed of sound. The transmitter and receiver are matched and will only respond to a frequency of 40 kHz so that talking next to the receiver should have no effect. The transmitter and receiver are both about 0.5 inches in diameter and should be mounted about one to two inches apart facing in the same direction. This separation prevents the transmitter from overwhelming the receiver while transmitting.

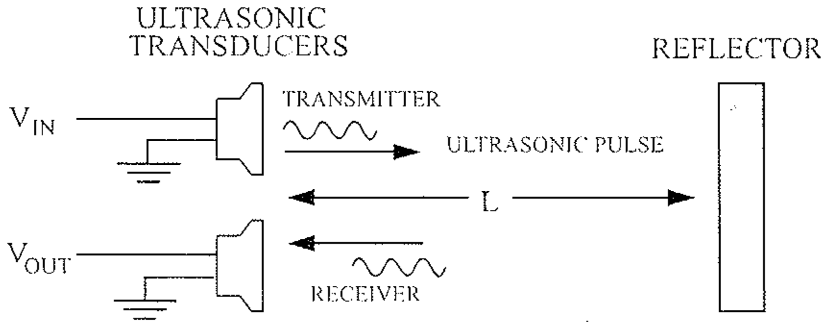


Figure 10.9: Rangefinder

The transmitter must be driven at a specific frequency of 40 kHz for a short period of time. Turning the 40 kHz signal on and off provides a means of distinguishing how long it takes to get from the transmitter to the receiver (i.e. you can't tell when a continuous oscillation starts or stops). Two 555 timers connected as shown in figure 10.10 can produce an appropriate signal.

The 555 on the left should generate a short pulse of about one millisec that is repeated every 15-20 millisec. The output of this oscillator is inverted and used to enable the second oscillator on the right. When STROBE is high the 555 on the right should oscillate at 40 kHz to drive the ultrasonic transmitter (the 555 can provide as much as 100 mAmp to drive the transducer). You will need to tune the 10K potentiometer RT to match the transmitter's frequency. The output of the receiver can be

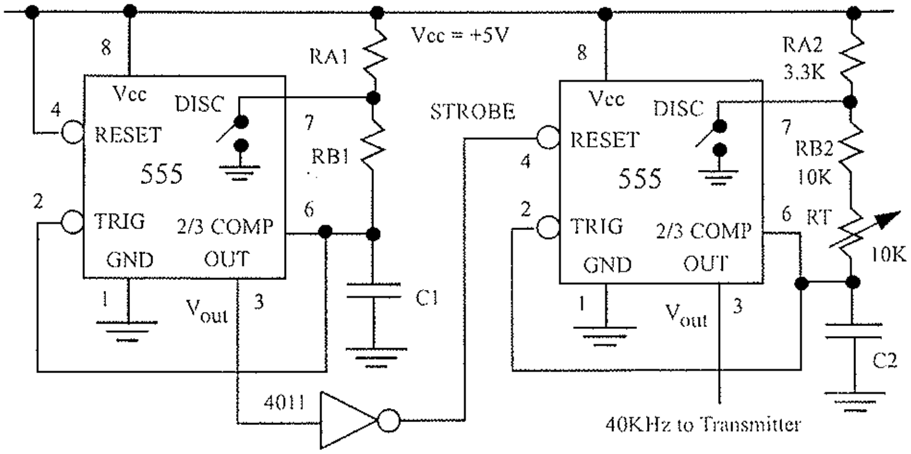


Figure 10.10: Rangefinder circuit.

viewed directly on the oscilloscope and the resulting waveform should look sometime like figure 10.11.

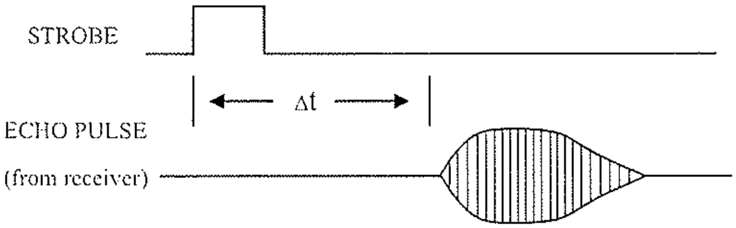


Figure 10.11: Scope trace for the ultrasonic rangefinder.

The delay time Δt between the transmitter STROBE signal and the echo pulse should be:

$$\Delta t = \frac{2L}{v_{sound}} + \Delta t_0 \tag{10.4}$$

where v_{sound} is the velocity of sound and Δt_0 is due to an unknown propagation delay in the transducer.

$\boxed{\text{Exp. 10.4}}$ Construct the transmitter driving circuit using two 555's as shown in figure 10.10.

(a) Calculate RA1, RB1, C1 to produce a pulse at STROBE that is high for about 0.5 to 1.0 millisec and repeated every 15 to 20 millisec (these times are not very critical).

(b) Calculate C2 to give an oscillation frequency of 40KHz when RT is in the middle of its range.

(c) Connect your 555 oscillator circuit to the ultrasonic transmitter and the receiver output to the oscilloscope. There should be two wires coming from each transducer that are twisted together. The case of each transducer should be either a black or green wire and should be connected to ground. The other wire is the signal. Hold a reflector (a large solid surface such as a textbook is good) an inch or two away from the transmitter so that a good strong signal is reflected back into the receiver. View the receiver output on the oscilloscope and tune RT until the signal is a maximum (200-300 millivolts peak-to-peak is typical). Be careful to hold the reflector still while tuning the oscillator frequency because small changes in the reflector angle and position can produce large changes in the receiver output.

(d) Now measure the time delay Δt between the beginning of the STROBE pulse and the beginning of the reflected pulse at the receiver for 4-5 different values of L between 4 inches and 20 inches (with a good reflector it may be possible to see the reflected pulse several feet away but the amplitude is very small). The transmitter and receiver send and receive in a very narrow cone of angles so you will have to adjust the orientation of the reflector carefully until you see the reflected signal (this may have to be repeated at each value of distance L). You will also find that the shape of the reflected pulse can vary with the shape and orientation of the reflector. The time delay from the start of the STROBE pulse to the start of the reflected pulse is however reproducible.

(e) Plot Δt versus L and extract the speed of sound from the slope of this line. Note that a finite propagation delay in the transducers etc. might prevent the line from going through the origin but should not effect the slope of the line.

end

10.8 Practice Problems

[1] Derive the expression for the one-shot pulse width time given in equation 10.2.

[2] Derive the expressions for the astable (oscillator) high and low times as given in equations 10.3.

[3] Design a digital capacitance meter. This circuit should produce a digital output proportional to the capacitance. Use a 555 monostable started from a single inverted pulse. The timing elements for the 555 should include the capacitor that is to be measured. The output of the 555 should enable an 8-bit binary counter (two 74LS161 four bit binary counter IC's). The counter clock should be a 100 kHz square wave from the Digi-Designer. Design the circuit so that a 10nF (0.010 μF) capacitor will produce a digital value of 10_{10}. You may assume that the counter is reset with another inverted pulse just prior to performing a capacitance measurement. Draw the whole schematic, label each component, and indicate where each of the two input pulses should be applied. Make a timing diagram showing *i)* the reset pulse to the counter, *ii)* the start pulse to the 555 timer, *iii)* the voltage across the capacitor, and *iv)* the enable to the counters. What is the accuracy and maximum range of this capacitance meter?

[4] Warble Alarm. The circuit in figure 10.12 produces a siren-like noise. The 555 on the left oscillates at a low frequency and controls the frequency of of the 555 on the right. The 555 on the right oscillates at two different audio frequencies and drives the speaker through the coupling capacitor C_{OUT}. $R_{A1} = R_{A2} =$ 10K, $R_{B1}=$ 1M, $R_{B2}=$ 3.9K, $R_{12}=$ 47K, C_1 =0.47μf, $C_2=$ 0.1μf, and $C_{OUT}=$ 10μf.
a) Calculate the frequency of the 555 on the left, f_1 in Hz.

b) When V_{OUT1} is low (0 volts) calculate the frequency of the 555 on the right, f_{2A} in Hz. (Hint, Mr. Thevenin might have something interesting to contribute to the solution.)
c) When V_{OUT1} is high (9 volts) calculate the frequency of the 555 on the right, f_{2B} in Hz.

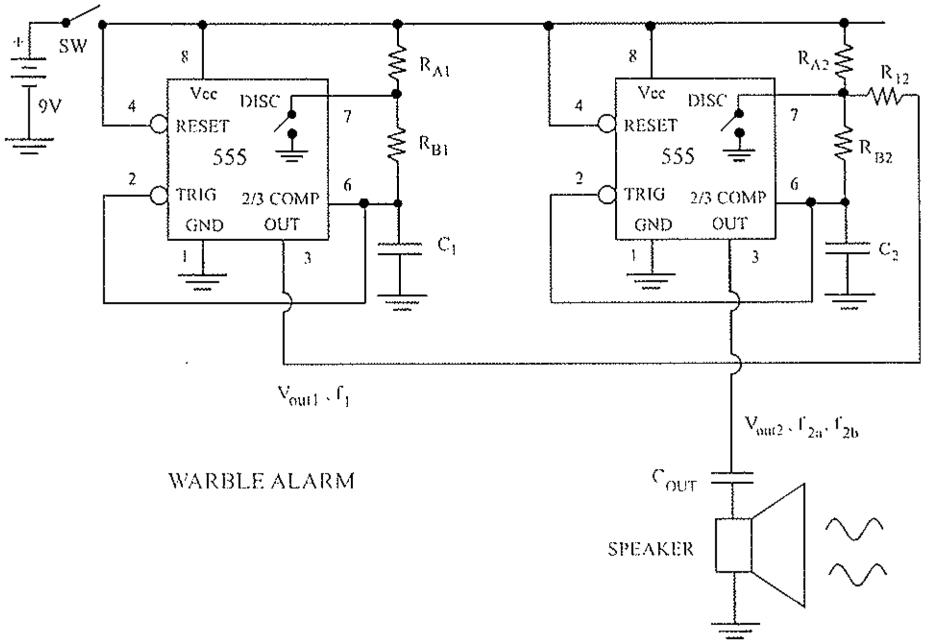


Figure 10.12: Warble alarm circuit problem 4.

[5] a) Design a digitally programmable non-retriggerable one-shot using a 74LS161 counter. There should be a four bit input determining the pulse width and it should be triggered by a single (positive-going) pulse at least one clock period long. You may assume that an appropriate clock input is available. (Hint: this can be done by adding just one gate and inverter.)

b) Give an expression relating the four bit digital input D_{IN} to the output pulse width and the clock frequency or period.

Chapter 11

LABORATORY COMPUTERS

The availability of fast and low cost computers has dramatically expanded the range of experiments that can be performed in almost all branches of science and engineering. The computer can record data and/or control a delicate experiment or other instrument (for example manufacturing equipment) on a time scale that is too long or too short for direct human control. Furthermore, the computer may provide highly nonlinear but precise control of an experiment or instrument that would be otherwise unattainable. For example, a computer may record a million or more data points over a period lasting several days while continually recalculating and adjusting various control voltages on a millisec. time scale to control an experiment or other manufacturing process. Using a computer in this manner is referred to as *real time control* and *data acquisition*. This mode of using the computer is very different from number crunching or simulation because it involves a direct electronic hardware interface to the real physical world. A typical configuration using a computer for control and data acquisition is shown in figure 11.1.

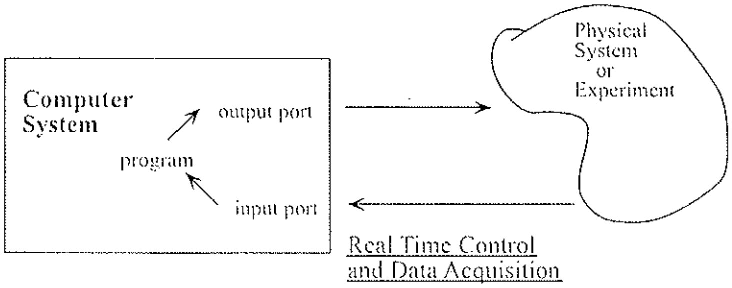

Computer System
output port
program
input port
Physical System or Experiment
Real Time Control and Data Acquisition

Figure 11.1: Computer usage in the laboratory.

The computer interfaces to the real world with one or more input/output (I/O) ports. These can be either digital or analog ports referring to the type of signal that is being used. This chapter will focus on controlling digital I/O ports and the next chapter will discuss analog I/O ports. A program running inside the computer will read and write to these ports to record data and/or control external physical electronic devices. From the programmer's point of view the IO port looks like a particular address inside the computer. Various program structures store or retrieve a digital number from or to the IO port. Inside the computer there is an electronics board that converts the actions of the program into real physical voltages and currents that appear on the cable coming out of the computer (usually hidden in the back). Additionally, sensors and transducers can be connected to the external cable of the IO port to convert electrical signals to or from other physical variables (such as temperature pressure, sound, etc.).

This chapter will start with some simple programs to control various external electronic circuits using a digital IO port. The primary goal is to get a sense of the range of useful electronic functions that the computer can perform and not on writing elaborate programs. Internally the computer has several million transistors. It is obviously not possible to get a detailed understanding of the inner workings of the computer but it suffices to say that computer contains many logic components (gates, flip-flops, etc.), which have already been discussed.

11.1 Computer System Description

All operations inside a computer (calculations, decision making, and communications) are performed in binary. Each *bit* of information has a logical value of 0 or 1, representing two different levels of voltages or currents. A *byte* consists of 8 bits of data. The basic organization of a microcomputer system is shown in figure 11.2.

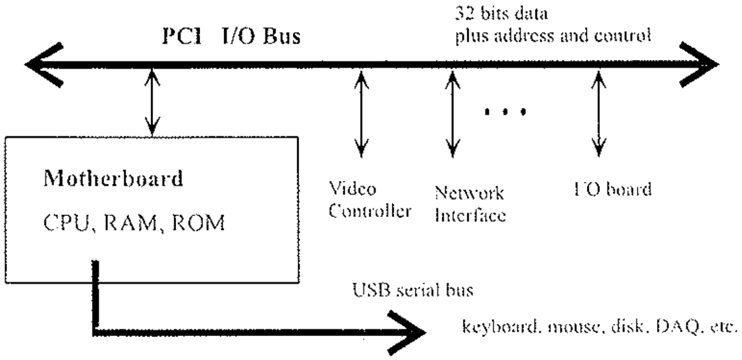


Figure 11.2: Computer system organization.

The *central processing unit* (CPU) of the PC is a large scale integrated circuit IC called a *microprocessor*. The microprocessor is the integrated circuit with the actual computer logic that is primarily responsible for the control of the system. It is by far the most complicated device in the system. The microprocessor and most of the associated electronics is mounted on a single large printed circuit board that is sometimes referred to as the *motherboard*. The motherboard, power supply and over peripherals such as the disk drives etc. are mounted inside a box that is collectively called the computer system. The microprocessor transfers information around the computer, performs arithmetic operations, writes data to output ports and reads data from input ports. A microprocessor can be used in a general purpose computer such as this one or as an embedded controller. For example most new cars have one or more embedded microprocessors that monitor the engine and make adjustments to improve fuel efficiency. These are referred to as embedded controllers because they are not generally accessible from the outside without specialized equipment.

The IBM-PC/AT (and its clones) was originally a 16-bit machine. This means that the internal organization is based on 16-bit registers, 16-bit parallel data paths, etc. Newer versions of the computer using the 80386 or later processors are 32 bit machines but generally retain backwards compatibility (i.e. they can still run the old programs etc.). The computer is a synchronous or sequential logic system. All internal operations are synchronized to a central oscillator or clock just like was used in chapter 9 for clocked flip-flops. When a microprocessor is said to be an xxMHz microprocessor this refers to the frequency of this clock or oscillator. The clock drives the flip flops inside the computer that are used to store data etc.

The main memory of the computer stores the sequence of instructions which comprises the program as well as the data. This memory may be either ROM (read only memory) or RAM (random access memory). ROM is nonvolatile meaning that it will remain when the power is turned off, and it cannot be written to, only read from. This is used to store various start-up programs. RAM is volatile meaning that whatever is stored in it will vanish when the power is turned off. RAM can be written to as well as read from.

11.1.1 Input and Output Connections (Buses)

Manufacturers of this style of computer have agreed on several standard configurations for connecting peripheral I/O (Input/Output) devices. The original IBM/PC had an 8 bit wide bus which was later upgraded to a 16 bit wide bus. This means that it could transfer digital data 8 or 16 bits at a time. This bus came to be called the the ISA or Industrial Standard Architecture bus. This bus is actually a series of slots (or sockets) on the motherboard. The connections on this connector support a common data path (either 8 bits or 16 bits wide) as well as a 16 bit address (24 bits on some slots). Each peripheral device has a unique address on the IO bus. This address is the means by which the microprocessor distinguishes between different peripherals. There are also various additional control signals that determine when an address is valid and when an I/O board should read or write its data from or to the ISA bus (controlled by the microprocessor). The ISA bus speed became a limiting factor as computers got faster. The EISA bus was popular for a brief time, and now the PCI 32 bit wide bus seems to be most common. Conceptually the PCI bus works in a similar manner to the ISA bus but is faster and generally better behaved (i.e. for plug-and-play features). Most new computers now come with PCI slots but no ISA slots. Some characteristics of these buses are summarized in table 11.1. Because the PCI bus is a well known standard, it is possible to buy different I/O boards to plug into this bus to perform different functions.

The PCI bus itself is being replaced by the PCI-express bus. At high speed it is difficult to keep all data bits synchronized in time, so very high speed buses, such as the PCI-express use serial data transmission (one bit at a time, but at exceptionally high speed). All of these buses are internal to the computer system, and required that the system be shut down and physically opened to be installed. The computer can also communicate digitally with external devices through lower speed external ports, such as the serial port (RS-232), the parallel printer port (8 bit wide).

Bus	Width (bits)	Max. Speed MHz)	Bandwidth (Mbytes/sec)	comments
ISA (8 bit)	8	8	8	original
ISA (16 bit)	16	8	16	
EISA	32	8	32	
PCI	32	33	132	
AGP	32	66	264	video only
PCI-express	1	2500	250	each direction per lane
PCI-express X4	1	2500	1000	each direction total
PCI-express X16	1	2500	4000	each direction toal
USB 1.0	1	12	1.5	external
USB 2.0	1	480	57	external

Table 11.1: Some peripheral buses for desktop computers. The bandwidth is a maximum theoretical bandwidth. Actual data transfers on the bus may involve several clock cycles, for example a typical bandwidth on the ISA bus is about 1 to 2 MBytes/sec.

The USB bus was designed for low cost and ease of use. It is a serial bus that has a simple connection on the outside of the computer system (external not internal). External USB devices can be connected and disconnected without powering down the computer system (hot swapped). The first version (USB 1.0) was intended for slow devices such as keyboards but the second version (USB 2.0) is suitable for high speed devices such as disk drives and video devices. The computer industry seems to be moving in the direction of using the USB buss (or equivalent) for all user connected devices and doing away with internal buses such as the PCI bus, with the computer becoming a closed box. The data acquisition device used in this chapter will be connected to the USB bus. It has a single simple USB cable connection to the computer. (See Axelson, *USB Complete, third edit.* 2005 for an extended discussion of USB.)

11.1.2 Lab Computer

The computer in the lab is one of a class labeled IBM-PC/AT clones or compatibles that are based on the original design of IBM. There are now numerous companies that make this style of computer. The microprocessor is a member of a 80x86 family made by Intel, that also includes the 8088, 8086, 80186, 80286, 80386, 80486, etc. After the 486 Intel started giving processors a name instead of a number. The next processor was called a Pentium, but then they added numbers to the name, so there is a Pentium-II, Pentium-III, Pentium-IV, etc. The Core and Core-2 series are the latest series from Intel. Likewise AMD started the Athlon series to be software compatible with the Pentiums. Currently both manufacturer make 64 bit processors that are also backward compatible with the 32 bit versions.

There are several types of computer systems in the lab are made by Dell and their basic features are summarized as:

Intel Pentium-4 (2.26,2.4 GHz) and Core 2 Duo (2.4GHz) processor

512 MB, 2 GB RAM
20 GB, 160 GB hard drive
Windows/XP and Windows Vista operating system
Microsoft Visual C/C++ compiler
National Instrument USB-6008 data acquisition device (DAQ)

The computers in the lab run the Windows operating system. This provides many high level features that make it easy to use but also complicates the low level I/O programming required for experimenting with electronics and I/O operations (as is the topic of this chapter). High level operating systems, such as UNIX (and Linux) or Windows do not allow the user to directly access the I/O operation for security reasons. A user program has to ask the operating system to perform the operations (and the operating system checks for security etc. in the I/O operation). Only those operations that have been programmed into the operating system can be performed. The manufacturer of the I/O device has supplied a system level device driver to talk to the board. This device driver runs as part of the operating system and is programmed to take care of the details of the board. The manufacturer has also supplied a library of subroutines that allow easy access to the DAQ device. This makes the whole process relatively transparent and easy to use, but there is a substantial performance penalty. Every programmed I/O operation must first ask the operating system to perform the operation and then wait for it to be completed. This can be a slow process, so the actual I/O bandwidth for single I/O operation is only a small fraction of the maximum. It will be fast enough for what is needed here however. In practice the trick to getting fast I/O is to perform each operation on large blocks of data for each call to the operating system. This is how disk drives work. Large blocks of data are transferred back and forth for each I/O operation. In less sophisticated operating systems (such as DOS) or in embedded controllers running without an operating system, low level I/O can be performed directly which is usually much faster, but you don't get the benefit of the operating system utilities, etc.

This lab assumes that you are generally familiar with how to use a desktop computer running the Microsoft Windows operating system. If you have not used Windows before, please ask your lab instructor to demonstrate it to you.

11.2 Microprocessor Evolution

Table 11.3 gives a brief history of the microprocessor "computer-on-a-chip" including which year each was introduced and its size. Both the clock speed and the number of transistors on an IC have been growing at an astounding pace since its first introduction in 1974. In the 1960's Intel co-founder Gordon Moore predicted that the number of transistors on a chip would double every 18 months. The microprocessor is a very complex IC and may not keep up with simpler IC's (like RAM) but has a similar growth pattern. This prediction seems to still be in effect and is now known as Moore's Law. Several companies (Intel. Motorola. IBM. etc.) also make a variety of microprocessors for general computing and embedded control although Intel and AMD seem to be the dominant supplier of desk top computers currently.

CPU	Company	Year	# Transistors	Clock (MHz)	# bits
4004	Intel	1971	2300	1	4
8008	Intel	1972	3500		8
8080	Intel	1974	6000	2	8
6800	Motorola	1974	4000		8
Z80	Zilog	1975	8500	2.5	8
6502	Mos-Tech	1976	9000		8
8086	Intel	1978	29 K	4-8	8/16
68000	Motorola	1979	68 K		16/32
80286	Intel	1982	134 K	8-12	8/16
80386	Intel	1985	275 K		8/16/32
80486	Intel	1989	1.2 M	50-100	8/16/32
Pentium	Intel	1993	3.1 M	60-200	8/16/32
PPC-601	Mot.,IBM	1993	2.8 M	50-100	32
PPC-603	Mot.,IBM	1994	1.6 M	80	32
Pentium-Pro	Intel	1995	5.5 M	200	8/16/32
PPC-604	Mot.,IBM	1995	3.6 M	100-200	32
Pentium-II	Intel	1997	7.5 M	233-450	8/16/32
Celeron A	Intel	1998	7.5M	300-533	8/16/32
Athlon	AMD	1999	22M	500-850	8/16/32
Pentium III	Intel	1999	28M	500-1000	8/16/32
Celeron II	Intel	2000	7.5M	533-1100	8/16/32
Athlon T'bird	AMD	2000	37M	500-850	8/16/32
Pentium IV	Intel	2000,2002	42M,55M	1400-3000	8/16/32
Athlon XP	AMD	2001	37.5M	1300-1533	8/16/32
Opteron	AMD	2003	105M	1800-2400	32/64
Pentium-M	Intel	2003	77M	1300-1700	8/16/32
Itanium-2	Intel	2003	221M	900-1500	64
Athlon-64	AMD	2003	106M	2000-2400	32/64
Pentium 4EE	Intel	2003	175M	3400	8/16/32
Pentium 4E	Intel	2004	125M	3400	8/16/32
Pentium-D 8xx	Intel	2005	233M	3400	8/16/32/64
Athlon-64 X2	AMD	2006	154M	2600	32/64
Core 2 Duo	Intel	2006	291M	2660	8/16/32/64
Core i7 quad	Intel	2009	731M	3200	8/16/32/64

Table 11.3: Brief History of the Microprocessor

11.3 Programming

The computer sequentially executes a series of very simple commands that together describe a specific task to be performed. The power of the computer results from its ability to rapidly perform millions of these simple commands over and over. This sequence of commands is called the program. At the most fundamental level the computer only understands its own language. This fundamental language is called machine language and is a sequence of binary 1's and 0's grouped into bytes (8 bits) or words (16 bits). It is almost impossible to program directly in machine language for everyday use. A close cousin to machine language is assembly language, which is a one to one symbolic representation of machine language. There is a short sequence of letters that corresponds to each sequence of 1's and 0's in machine language. Assembly and machine language are so closely related that the two names are frequently used interchangeably. Both assembly and machine language are unique to a given computer. Machine language programs from one type of computer do not in general run on any other type of computer. Even though assembler is much easier to program with, it is still awkward for simple programs. However a well written assembly/machine language program will execute the fastest and is capable of the most detailed control over the computer.

Compiled high level languages such as C/C++, Pascal, Fortran, etc. offer many advantages over assembler/machine language. These computer languages are composed of an English-like set of instructions. Programs written in a high level language are generally much easier to read, understand and make work. Furthermore, a standard form of the language exists so that a program written in a high level language on one type of computer may also be run on another type of computer. This feat is accomplished by using what is called a compiler. The compiler itself is also a program. Its job is to read the high level language program (also called source code) and translate it into raw machine code that the computer understands. A compiler is therefore unique to a given type of computer system. A good compiler will also provide various other services such as simple forms of error checking (to help get the program working), and various standard functions in a simple form to speed up the job of programming (i.e. writing characters to the screen or calculating the trigonometric sine of a number). The price that is paid for these features is a slight reduction in execution speed. A good compiler will typically produce machine code that executes only 2X to 10X times slower than a well written machine language version of the program. If the program only has to be run a few times, or if the computer is fast enough to execute the program without a significant delay, then the overall time, including the time it takes to write the program, is significantly shorter if a high level language is used. Furthermore a high level language program is easier to debug (i.e. to get working the first time) and to maintain over a period of time. Usually you will only need to use machine language as a last resort if you absolutely need the speed or some detailed function not available in the high level language.

Interpreted high level languages such as BASIC and some forms of Java go one step further. The interpreter continually translates each high level statement into machine language as it goes. This saves an intermediate step of compiling the program but the execution speed may be slowed down by a factor of 100X to 1000X.

One of the best ways to learn programming is to look at other people's programs and thoroughly understand how they work. The following experiments will explore simple programs in a high level

Binary	Hex	Decimal	Binary	Hex	Decimal
0000	0	0	1010	A	10
0001	1	1	1011	B	11
0010	2	2	1100	C	12
0011	3	3	1101	D	13
0100	4	4	1110	E	14
0101	5	5	1111	F	15
0110	6	6	10000	10	16
0111	7	7	10111	17	23
1000	8	8	10100010	A2	162
1001	9	9			

Table 11.4: Binary and hexadecimal numbers and their base ten (decimal) equivalents.

language called C/C++ and also look at assembly/machine language programs. A complete description of C/C++ or machine language is not possible in the limited time available. Instead a small subset of the features available to each language will be used to understand how the computer interacts with the external world through its electronic interface, and how it might be used in a scientific laboratory setting.

11.4 Hexadecimal Numbers

The computer uses binary numbers which can be easily represented using the *hexadecimal* system, with base 16 rather than base 10. Each hex digit counts from 0 to F (the normal base 10 numbers plus the first 6 letters of the alphabet) before returning to zero and incrementing the next most significant digit by one. Each group of four binary digits or bits can be represented by a single unique hex digit. The hexadecimal system is used because it relates so closely to *binary*. Four binary digits (four bits) are "packed" into one hexadecimal digit: $2^4 = 16$. The correspondence between binary, hexadecimal, and decimal numbers is illustrated in table 11.4. (An alternative notation commonly used is the *octal* system where three bits are packed into one digit to base 8.) The advantage of hex (as opposed to decimal or base 10) is that, with a little practice, you can convert back and forth quickly by inspection without doing any arithmetic. For example, the binary (base 2) number 111100100000_2 divided into groups of four bits is 1111-0010-0000 which is $F20_{16}$ in hex. When writing constants in C/C++ programs, hex numbers will be preceded by a 0x (i.e. 0x220 is 544 in base 10), and when programming in assembler add an H to the end of each hex number. Note that DEBUG assumes that all numbers are in hex so neither of these is necessary when using DEBUG (more on DEBUG later).

11.5 Programming in Assembly Language

The CPU or central processing unit of the computer performs all of the operations described by the program. The program and data are stored (in binary) in RAM (read/write random access memory).

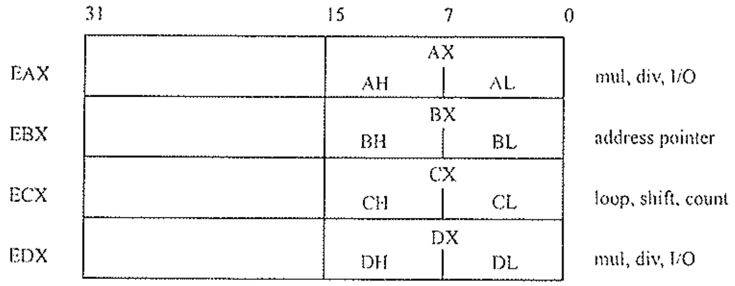


Figure 11.3: General purpose registers

This memory is physically placed on the main computer board and is essentially a collection of flip flops as in Chapter IX (albeit a vast number of them).

In the Intel family of computers memory (RAM) is organized as a sequence of 8 bit (or 1 byte) storage cells. Each of these storage cells has a unique 32 bit address. To store a byte of information in one of these cells requires knowing its 32 bit address. This means that there can be at most 2^{32} bytes (or 4 gigabytes) of memory. The earlier 8086 processors used a 16 bit address (which can only address 2^{16}=64K bytes at a time which caused a lot of problems). Some special versions of the Intel processor (Xeons) have a 36 bit addressing capability and the x86-64 or EMT64 processors use a 64 bit address. New versions of the Intel CPU's still have the capability of running in 16-bit mode for compatibility with older programs. Physical I/O (input/output) devices have their own set of 16 bit addresses. It is convenient to think of these two types of addresses as the memory space and the I/O space. Each type of space will have its own unique way of accessing the locations within it. It should be stressed that all of what is said in this section is very specific to the Intel family of microprocessors and does NOT apply to any other microprocessor, although the general concepts are relevant to most types of microprocessors.

Inside the CPU there is another set of fast memory cells that have very special uses. These are the CPU registers shown in figure 11.3. Almost all machine instructions refer to one or more of these registers. The four 32 bit wide general registers EAX, EBX, ECX, and EDX are used quite often. Inside the CPU, there is a bank of 32 D Flip Flops for each register that holds the value stored in these registers. Some of the specific uses of each register are also shown in figure 11.3

The lower significant 16 bits of each of these four registers can be accessed as the AX, BX, CX, and DX 16 bit wide registers and the individual high and low 8 bit portions of these registers can be accessed as the 8 bit wide registers AH, AL, BH, BL, CH, CL, DH, DL as shown in figure 11.3. The AH register is an eight bit register that is the high order byte of the AX register and the AL register is the low order byte of the AX register. Similar arrangements exist for the BX, CX, and DX registers. Although these are called general registers they cannot be used in a completely general manner. The 8 and 16 bit wide registers are compatible with earlier 16 bit versions of the Intel processors. There are a variety of other registers used to control the state of the processor and store the status of various things, but these can be ignored for the simplified discussion here.

There are several hundred different machine instructions that can be performed (each new version of the Intel processor seems to add new instructions). Many of these also have several different flavors or modes, and when taken all together there are many different operations that can be performed. Please refer to an official Intel data book for a complete description of all of the instructions. Fortunately, there is a small subset of commands that is sufficient for the simple programs considered here and these will be described next. Each assembly language instruction looks something like:

```
label:  op-code destination, source
```

The label is optional and, if preset, is a means of referring to the location of this instruction (note that the DEBUG assembler does not accept the label field). The op-code represents the operation to be performed. The destination and the source are referred to as the operands. There may be 0, 1, or 2 operands depending upon the instruction. The source operand is the source of data and the destination is the place the results of the operation will be stored. This format of assembly language is similar on most computers, however the order of the operands is often reversed from the Intel format.

Several instructions are summarized in table 11.5 where s=source, and d=destination. Although there are many other details and instructions not listed here, this short list is enough to do many useful things. In principle each instruction requires a specific number of clock cycles, however the number will vary depending on what type of address it is using and on a variety of other conditions in the processor. It is no longer practical to try to estimate the precise timing of an assembly language program due to the complexity involved. On a simpler processor running a simpler operating system (for example on embedded controller) the timing is routinely calculated to determine if a given procedure will finish in time. Also note that the IN and OUT instruction are listed, but are not allowed from a user program in Windows or UNIX. I/O in general is considered a privileged operation and can only be performed by the operating system due to security reasons.

The address of the source (s) and destination (d) may be specified in several way as listed here (this may not make any sense without specific examples):

direct an address is specified

immediate a value is specified

register a register name is specified

indirect a register (must be BX or EBX) is specified whose value will be used as an offset address

indexed add another value to the BX register to obtain an offset address

If both a source and destination are specified then usually one or both must be one of the general registers. Also, both the source and destination must be the same size. If one is a 16 bit value then the other must be also (or both may be 8 or 32 bit values). Please note the distinction between an address and a value that may be stored at this address. Each of these types of addressing may take a different amount of time (CPU clock cycles). Registers can be accessed quickly but memory takes longer to access. Some examples of addressing modes (using 16 bit mode registers as is consistent with DEBUG in the following section) are shown in table 11.6.

ADD d, s	add contents of s to d and store in d
AND d, s	bitwise and s with d and store in d
CMP d, s	compare the source and destination i.e. form d-s and save status but not result
DEC d	decrement the contents of d by 1
IN d, s	input from the I/O space IN AL, DX byte data IN AX, DX word data the I/O port address is in DX, and AL or AX will receive the data from the I/O port
INC d	increment contents of d by 1
JE loc	jump to location loc if the last instruction yielded a result of 0
JMP loc	jump to location loc
JNE loc	jump to location loc if the last instruction did not yield a result of 0
MOV d, s	move s into d
NEG d	negate d (d gets -d)
NOT d	invert all bits in d
OR d, s	bitwise or d and s and store in d
OUT d, s	output to the I/O space OUT DX, AL byte data OUT DX, AX word data the I/O port address is in DX and AL or AX will receive the data from the I/O port
SHL d, s	shift d left by s bits
SHR d, s	shift d right by s bits
SUB d, s	subtract s from d and store result in d
XOR d, s	d gets the bitwise exclusive OR of d and s

Table 11.5: Some assembly language instructions.

Assembler	Address Mode and Description
MOV BX, 101H	register mode, immediate mode move the hex value 101 to the 16 bit BX register
MOV BX, [101H]	register mode, direct mode move the value stored at location 101 into the BX register
MOV [BX], AX	indirect mode, register mode move the value stored in the AX register into the location whose offset address is in BX
MOV DX, [BX+2]	register mode, indexed mode move the value stored at the location specified by adding 2 to the value stored in the BX register into the DX register

Table 11.6: Addressing mode.

11.6 Using DEBUG

DEBUG is an old 16 bit DOS program that still runs in Windows. It is still useful to investigate a short assembly language program to see how the CPU works internally. Note that it only runs in 16-bit mode not 32 bit mode. Also note that its input and output instruction will not function under Windows. A short list of DEBUG commands is shown in table 11.7.

This simple 16-bit utility program can be used to explore simple machine language principles and to store and examine various memory locations inside the computer. To invoke DEBUG simply type DEBUG followed by a <RETURN> in the command prompt window. DEBUG will start up in command mode (with its - prompt). A short list of some of its useful commands (all values are in hex) is given in table 11.7.

11.7 A Sample Program in Assembly Language

The following program will familiarize you with the mechanics of creating a working assembly language program, assembling it, loading it into memory, running it both at normal speed and at single-step speed, and displaying and modifying the contents of memory locations using DEBUG. Below is a 16 bit mode program to interchange the contents of two memory locations.

a<address>	enter assembler input mode DEBUG will prompt you for code starting at your specified address. You may continue typing more lines of code until your program is completed. Type <RETURN> on a blank line to return to command mode.
d<loc1> <loc2>	dump the values stored at locations <loc1> through <loc2>
e<address>	start entering values at location <address> DEBUG will prompt you for hex values to store in memory. Typing <SPACE> after the value will continue the entering process. A <RETURN> key will exit enter mode.
g=<address>	Start the program (i.e. go) at location <address>
q	quit DEBUG and return to DOS
t=<add> <n>	trace the program, i.e. run the program starting at address <add> for <n> statements and display each statement as it is executed
u<address>	unassemble the machine code starting at location <address>

Table 11.7: DEBUG commands

Line	Program	Description of Operation
1	MOV BX, 200	store immediate value 200H into 16 bit BX register
2	MOV CH, [BX]	store 8 bit value at location 200H in the CH register
3	MOV CL, [BX+1]	store 8 bit value at location 201H in the CL register
4	MOV [BX], CL	store the value in the CL register into location 200H
5	MOV [BX+1], CH	store the value in the CH register into location 201H
6	INT 20	software interrupt to exit the program (DOS function)

Note that all numbers are assumed to be in hex when using DEBUG (also the line numbers and the description should not be typed into DEBUG). This program illustrates several modes of addressing. The first line uses immediate mode. This means that the actual value is specified (the value will appear in memory following the op-code). The second line uses indirect addressing. The offset address of the desired location has been loaded into the BX register. Enclosing BX in [..] indicates to the computer to use the value stored in BX as an address. Note that BX is the only general register that may be used for this. The third line uses indexed addressing. The value 1 is added to the value stored in the BX register and this new value is the address of the desired source location. Because memory is accessed 8 bits at a time and memory locations specify 8 bit units the address increment is 1. Lines 2,3 get values from memory and store them into internal registers and lines 4,5 reverse this procedure and store the values in the registers back into main memory but in the reverse order. This rather crude procedure is quite typical of machine/assembly language. Only very simple operations may be performed (very fast though). This is what makes assembly language programming tedious and time consuming.

Exp. 11.1 Start the computer and login as "user". Start a command prompt window and type DEBUG to start it running. Type in the 16 bit mode program shown above. Once DEBUG is running type 'a100' and then the program. When the program has been typed enter a <RETURN> on a blank line to return to command mode. Now type 'u100' to unassemble what you have typed and observe the raw machine code generated. Next enter the hex values 'A1' and 'B2' into locations 200H and 201H using the DEBUG 'e200' command. Dump these values with 'd 200 210', run the program with 'g=100', and then dump the locations 200 through 210 again. Has the program performed as expected? end

Exp. 11.2 Repeat the previous experiment, however run the program in trace mode using the command 't=100 5' so that you may observe the execution of each individual statement. Trace mode will give a detailed listing of each step in the program as it is performed. end

This illustrates the main ingredients of assembly language programming. It would require a large amount of programming practice to become proficient but this example gives a rough idea of what is going on inside the computer.

11.8 The Microsoft Visual C/C++ Programming Environment

The lab computers have Microsoft Visual C/C++ 2005 Express installed. This is a complete integrated development environment (IDE) that includes an editor, a compiler and a linker. It is relatively easy to type in a program and run it from within the Visual C/C++ IDE. First launch Visual C++ from the START button (in the lower left hand corner). A project is a description of the executable that you want to build. Pull down the 'File/New' menu and choose 'Project'. Then select 'Win32-Console-Application', give it a name and choose 'empty project' under 'application type' to start a new project. It also requests you to select a directory and project (i.e. executable program) name at this stage. Note that you may only store files in the directory C:\users directory (may be D:\user on some machines). Do NOT store files anywhere else. To enter the actual C/C++ program choose 'Project/Add-New-Item', select 'C++ file' and give it a name ending in '.cpp'. When you are done typing save the file. It should be inserted into your project automatically. If not, then insert it into the project using the 'Project/Add-Existing-Items' menu item. To compile your program, pull down the "Build/Rebuild-Solution" menu to compile the program. MSVC++ should add the appropriate libraries etc. into your project as well as the source code file. You can execute your program while in the Visual Studio using 'Debug/Start...' menu or invoke the command prompt and execute it manually (it will be inside the Debug or Release sub- directory).

11.9 A Sample Program in C/++

The following short program illustrates the general structure of a C/C++ program:

```
/*  example.cpp */

#include <stdio.h>  /* for printf() */
#include <math.h>   /* for sin() */

int main()
{   int i;
    double x;
    for( i=0; i<10; i++) {
        x = i;
        printf( "%f %f\n", x, sin(x) );
    }
    return(0);
}  /* end main() */
```

Each program in C/C++ is actually just a single function called main(). In this example main() is proceeded by int to indicate that it returns a value (in the return() statement at the end). The code between the curly brackets following main is the actual code for this program. Brackets create a block or group of statements that appear as a single entity. Each program statement can extend onto one or more line and must end with a semicolon (;). Comments (notes to the programmer that are not executed by the computer) begin with /* and end with */ and may flow across more than one line if necessary. C++ adds an additional comment structure with the double slash //, meaning that the rest of the line after // is a comment and ignored by the compiler. The two lines at the top that begin with a # in the first column are compiler directives. These are not compiled into executable code but are instructions to the compiler on how the program is to be compiled. In this example the header files **stdio.h** and **math.h** are to be included. These define the **printf()** function and the trigonometric function **sin()** respectively.

The actual program begins after the left bracket following **main()**. The next few lines declare the variables that will be used. C/C++ requires that all variables be defined before they are used. Some common variable types are:

short = 16 bit integer

long = 32 bit integer

int = a long or short integer (on these computer defaults to a long)

double = 64 bit double precision floating point

float = 32 bit single precision floating point

The program declares a variable **i** as a 32 bit integer and a variable x as a double precision floating point number. (Note that the size of these variable may vary from one compiler to the next and the sizes given here are for this particular compiler). Following the variable declaration is a **for()** statement. The **for** statement is one form of loop in C/C++ and is used to repetitively execute the code within the brackets following the for statement. There are three fields separated by semicolons inside the parenthesis following the **for** keyword. The first is the initial condition, the second is the continuation condition and the third is the repeat operation performed at the end of the loop. In this example the code in between the brackets is executed for each value of **i** starting with **i=0** and ending with **i=9** (note that there is an inequality in the continuation condition and not an equality). The **i++** statement means to increment **i** by 1 and is performed after each pass through the loop. The **x=i;** statement converts the integer **i** into a floating point number x. The **printf()** statement prints a formatted line to the screen. The first argument to **printf** is the format or how to print the following variable. It must be enclosed with double quote (" and not '). The field %f means a single floating point number. The \n at the end of the format means to print a newline. After the format the variables x and function sin(x) are listed separated by commas. This means to print x as a floating point number and to evaluate sin(x) and also print it as a floating point number. The end result of the program is to print a short table of value x and sin(x).

If you are unfamiliar with programming in C/C++ then you should type in this program and test it.

11.10 The Data Acquisition Port (DAQ)

The actual implementation of the electronics of the I/O interfaces can be quite sophisticated. Generally speaking it is cheaper and faster to buy a ready made device and use it instead of interfacing your circuits directly to the computer I/O bus. There are a great variety of boards and devices available for a relatively small amount of money. The I/O device that you will use here is plugged into the USB bus and was purchased from a company called *National Instruments* (see www.ni.com) and is identified as a USB-6008 (figure 11.4). It is relatively low cost and modest performance, but adequate for the experiments presented here. Generally speaking the more money you pay, the better the performance (speed and accuracy) you get. Although many of the features on this device are unique to this particular device, many of the general ideas are applicable to many different I/O devices. Unfortunately every I/O board or device you might buy is different so its not possible to discuss the experiments in a completely general manner, because one particular device is required to do the experiments.

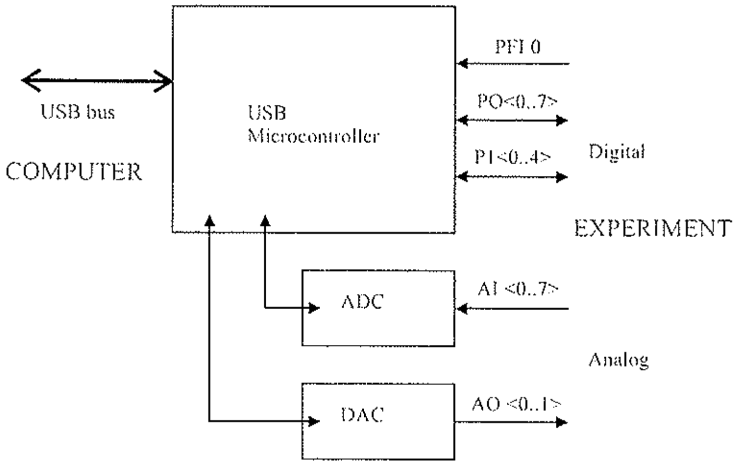


Figure 11.4: The National Instruments USB data acquisition (DAQ) USB-6008 device. There are two digital ports P0 (8 bits) and P1 (4 bits), an eight (8) channel multiplexed ADC and two (2) DAC analog ports.

This data acquisition (DAQ) device has 12 bits of digital I/O organized as one eight (8) bit port (P0) and one 4 bit port (P1), one 12 bit analog to digital convert (ADC) and two (2) digital to analog converters. Both digital ports are bidirectional (input or output), and the ADC is multiplexed into eight (8) single ended or 4 differential inputs with a maximum sampling rate of 10kHz (more money gets you faster sampling). Communicating over the serial USB bus involves a fairly complicated protocol which is best performed with another microprocessor as in this device. This microprocessor also controls the actual ADC etc. Internally there is a small amount of memory (512 byts RAM) to buffer the ADC process. The single digital input PF1 can be used to trigger the ADC process This chapter will concentrate on using the digital I/O ports. Chapter 12 will discuss analog I/O.

This device is a small rectangular plastic box about 3 inches in length. It is connected to the USB socket on the computer with a single long cable. Along each side of the device are a set of 16 small

signal	pin	pin	signal
GND	1	17	PO.0
AI 0/AI 0+	2	18	PO.1
AI 4/AI 0-	3	19	PO.2
GND	4	20	PO.3
AI 1/AI 1+	5	21	PO.4
AI 5/AI 1-	6	22	PO.5
GND	7	23	PO.6
AI 2/AI 2+	8	24	PO.7
AI 6/AI 2-	9	25	P1.0
GND	10	26	P1.1
AI 3/AI 3+	11	27	P1.2
AI 7/AI 3-	12	28	P1.3
GND	13	29	PFI10
AO.0	14	30	+2.5V
AO.1	15	31	+5.0V
GND	16	32	GND

Table 11.8: Pin assignments on the NI USB-6008. PO and P1 are digital IO ports. AI are the ADC inputs (single ended or differential), AO is the DAC output and PFI0 is a digital input for triggering. GND is ground. Do NOT connect the signals labeled +2.5V or +5.0V.

screw down terminals in black plastic for electronic connections. Each of these signals is labeled in table 11.8. To use this device, place it close to the breadboard, insert a short length of wire into each socket in the black plastic screw-down terminals on the side and tighten the screw to get good contact. Then insert the other end of the wire into your breadboard.

> *CAUTION: The software recognizes a unique serial number on each USB device. If you plug in a different USB-6008 then the computer tries to install a new device driver which doesn't work from a user account. This means that the USB-6008 devices are NOT interchangable. Please leave it connected to the computer and do not swap them around.*

For the digital IO experiment in this chapter use digital ports PO and P1. Individual bits in P0 are labeled PO.0, PO.1, etc. to PO.7 for a total of eight (8) bits. PO.0 is the LSB (least significant bit) and PO.7 is the MSB (most significant bit). Each wire is one bit in a byte of data. Each digital signal is a TTL logic level. The lines labeled AI0 through AI7 are the analog inputs (ADC) for single ended signals and AI0+/AI0- through AI3+/AI3- are for differential analog input signals. AO.0 and AO.1 are the analog outputs (DAC).

There is one set of 8 wires used for one port (used as output below) and another set of 4 wires for another port(used as input below). Reading a byte from this device records the electrical signals on each of the input wires as a 0 or a 1. Writing a byte to this device changes the voltage on the other set of wires corresponding to whether the associated bit is a 0 or a 1. The electrical signals are TTL levels. It is also important to connect the appropriate ground wire from this device to the ground of your circuit. The computer/device and your circuit MUST be on the same ground. Do NOT connect the lines labeled +2.5V and +5.0V. Be careful not to allow any of the other wires to inadvertently contact some other electrical signal or each other. The device may be damaged if this is allowed to happen. Make sure that they are left floating.

In a high level operating system such as UNIX or Windows/XP/Vista direct low level I/O operations are generally not allowed from a user mode program for security and stability reasons. I/O is considered to be a privileged operation and can only be done by the operating system itself. To access I/O from a user mode program (as you will do next) requires you to access what is called a device driver. This is a small segment of code that runs as part of the operating system and performs the actual I/O operation. The manufacturer of these I/O boards has provided a device driver for it as well as an easy-to-use (in a relative sense, compared to writing a device driver ourselves) subroutine library. In practice your program will call a few subroutines from this library, that in turn make the appropriate operating system calls to invoke the device driver that performs the I/O operation. In this manner the operating system gets to supervise all I/O operations and to prevent interference with other operating system functions. On the whole this makes for a more stable and secure operating system, but there is a large performance penalty. Every I/O operation has to go through the operating system which is a very slow process. Performing I/O one byte at a time (as you will do next) is easy to program but is slow (however computers keep getting faster so the speed is about the same as it was 10 years ago with slower processors but less sophisticated operating systems). To get better performance in practice, each I/O operation should be performed on a large block of data at one time, however this is not consistent with the type of operations in the next few experiments. Operation over USB has a very large performance penalty for single transfers for reasons that are not clear, so block IO is even more important. In return USB is much more convenient to use.

To access this library you need to do two things. First you have to include the description of the subroutines with the header file ni.h and secondly you have to link your program with the library file NIDAQmx.lib. Then you can call the subroutines from your program. There is an extensive help utility on-line under the National Instruments menu item (it is rather long and includes many items not needed here). The software interface provided by NI is intended to control all of the many different devices that NI makes, so is somewhat more general and complicated than needed for these experiment. The include file ni.h defines several simple, easy to use subroutine that call the appropriate NI subroutines, and also includes the actual NI header file called NIDAQmx.h. A full listing of ni.h is given at the end of this chapter.

ni.h	header file to include in each program
NIDAQmx.lib	device driver calling library to link each program with

The library file has been added to the appropriate compiler directories, so you don't have to do anything with it directly. There should be a copy of the header file ni.h in the user directory. All you have to do is add the appropriate include statement at the top of the program (see examples that follow) and add the library name to the project library list. Select the 'Project/Properties' menu item and choose "Configuration-Properties/Linker/Input". Under 'Additional-Dependencies' type in 'NIDAQmx.lib'. This will add the library to the file.

The file ni.h defines two easy-to-use subroutines that you can use in the following experiments.

- int DigitalOut(uInt8 data[], int n): output n bytes of data from array data[] and return number of bytes actually output
- int DigitalIn(uInt8 data[], int n): input n bytes of data to array data[] and return number of bytes actually input

11.11 Reading Data Into Computer From the Outside World

The next experiment will read the positions of switches SW0-SW3 that are connected to the least significant 4 bits of the input port (P1.0-P1.3) as shown in figure 11.5.

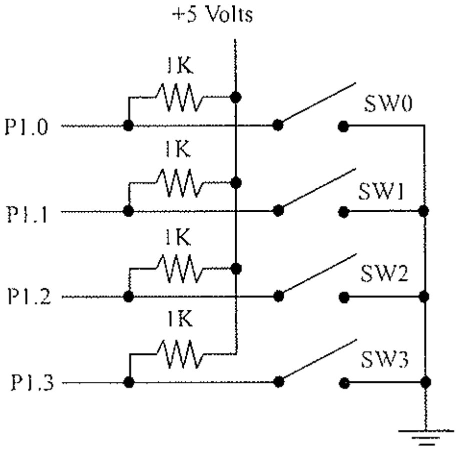


Figure 11.5: Reading the logic switches (the resistors are inside the Digi-designer, so you don't have to wire them).

The following program will read the input port and write its value to the screen. At the beginning of the program the header files "windows.h" and "ni.h" are included. The first defines the symbols for the windows operating system subroutines and the second defines the device driver subroutines. NI has defined a new data type called **uInt8** that is an unsigned 8 bit value or one byte which is probably equivalent to a C type of unsigned char. The actual I/O read operation is performed with a call to **DigitalIn()** to read a byte of data from the I/O device. The read operations are in a **do{...}while()** loop. This repeats everything in between the brackets {} as long as the Boolean expression inside the parenthesis is true. In this case the Boolean is true when a key is pressed on the keyboard other than a 'q'. **getchar()** causes the program to wait for the key (this is a standard C/C++ subroutine defined in stdio.h). Therefore the code inside the {} following the **do** will run every time you type the return key. To abort the program type a 'q' followed by a carriage return. The program prints out the value read from the I/O port using a format of %x to output the value in hex format. The program ends with a **return**.

```
/*  readsw.cpp : read the logic switches  */

#include <stdio.h>
#include <stdlib.h>
#include "windows.h"

#include "C:\user\ni.h" // Nat. Instr. I/O driver

int main()
{
    uInt8 data[1];

    do {
        DigitalIn( data, 1 );
        printf( "input= %x\n", data[0] );

    } while( getchar() != 'q' );

    return( 0 );
}
```

There are many different people that will have to use these computers. It is important that you keep the computer in good working order. You are asked to refrain from writing any files into directories other than C:\user and to refrain from installing any other programs or moving other than your own files around.

Exp. 11.3 Connect four of the 'Logic Switches' on the Digi-Designer (that look like the schematic in figure 11.5) so that P1.0-P1.3 can be made HI or LO. The resistors are already inside the Digi-designer, you don't have to wire the resistors, but just connect the switches. Be careful to connect the computer ground to your circuit ground. These are TTL inputs, do NOT connect voltage other than 0V to 5V to these inputs. Large voltages will destroy the board.

Type in the program above to read the data from the IO port and run it. Make a table of Hex values read versus switch positions. Do the values correctly represent the switch positions? end

11.12 Writing Data Out of Computer

The digital output port can be used to control the segments of a 7-segment LED display as shown in figure 11.6. This device has 7 diodes that will light up when current flows through them. Each segment of the display is a separate LED. The LED's may draw a lot of current so it is a good idea to add a 7400 or 7404 buffer with a current limiting resistor as shown. Note that the resistors and inverters are not included in the 7-segment display. The program shown below controls the individual LEDs directly.

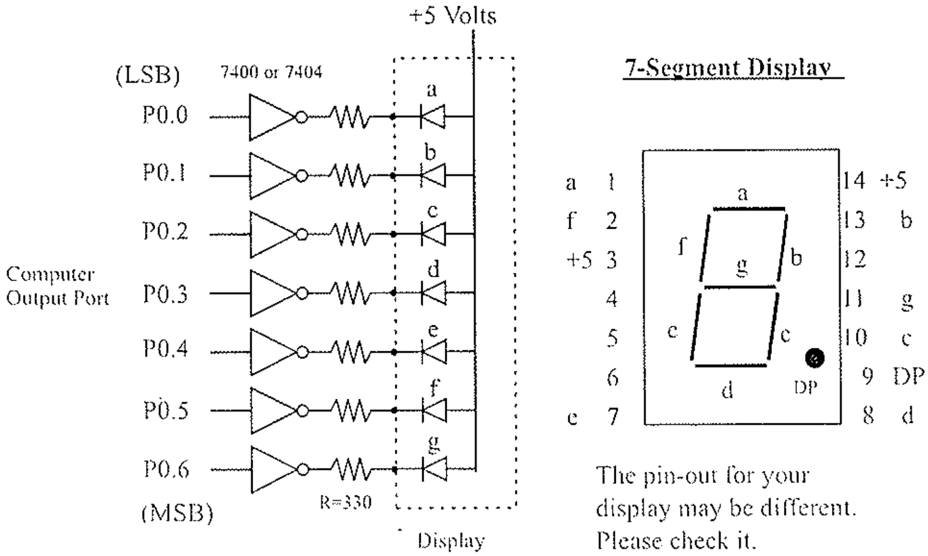
```
/*  led.cpp : write to the LEDs  */
```

```
#include <stdio.h>
#include <stdlib.h>
#include "windows.h"
#include "C:\user\ni.h" // Nat. Instr. I/O driver

int main()
{
    int i, led[16] ={0,1,2,4,8,16,32,64,255,1,2,4,8,16,32,64};
    uInt8 data[1];

    do {
        for( i=0; i<16; i++) {
            printf( "led[%d]= %d\r", i, led[i] );
            data[0] = led[i];
            DigitalOut( data, 1 );
            Sleep(200);
        }
     } while( getchar() != ’q’ );

    return( 0 );
}
```



The pin-out for your display may be different. Please check it.

Figure 11.6: Writing out data to drive an LED display.

This program is similar to the previous program that read the logic switches. However, it outputs a value (stored in the array **data**) and calls **DigitalOut()** to output data instead of input data. This program declares an array of data called led[] that has the bit patterns to write to the LED display. An array is really just a subscripted variable. The subscripts start with 0 and go to 15 in this program. C/C++ allows the array to be initialized by attaching an ={...} to the end of the declaration. The array elements are initialized to the list of values inside the brackets {}. For example led[0]=0, led[1]=1,

..., **led[8]=255**, etc. **led[]** is declared as a type **int**.

A delay of 0.2 sec has been inserted into the program to slow it down enough to read the display. The delay is made using the Windows subroutine **Sleep(dt)** where **dt** is the time in millisec. to wait (note that C/C++ is case sensitive). This is a useful feature of Windows and is not a standard C/C++ subroutine.

It is a lot easier to reuse the same project so you don't have to repeat all of the tedious procedure you just completed in the previous experiment. Also a significant portion of the program source code is repeated in each new program. Just keep the same project, save the old program source code under a new name (use the file/save-as menu). Highlight the old source name in the project menu and hit the delete key to remove it. Then choose Project/AddToProject and add the new file you just made. Edit this new file instead of retyping a lot of redundant code (and save the old program in case you need to go back to it). This should let you complete these experiments much faster.

Exp. 11.4 Wire the 7-segment LED display as shown in figure 11.6 so that the computer can be used to produce a digital display. The resistors and inverters are not part of the LED. You must wire BOTH of these externally (7 resistors and 7 inverters). Do NOT drive the LED directly from the computer output port. These are TTL outputs, do NOT connect any input voltages to these outputs and do NOT connect them to ground etc., otherwise you will burn out the device. It is easer to place the two IC packages (7400 or 7404) on opposite sides of the LED package so the resistors are less likely to short.

First, type in the program above and run it. Observe the pattern of lights shown on the LED. Correlate this with the program.

Next make a table of hex numbers which must be entered into the output port in order to light up the figures 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F on the LED display. Be careful that are symbols are different (you may have to use both upper case and lower case letters).

And finally, modify the program so that it shows these figures in this order with about 0.5 sec between characters, and verify your hex numbers by running the program. You can enter hex number by preceding the number with a 0x (number zero and letter x). end

11.13 Computer I/O Speed

The speed of the computer is very important if you want to control an experiment in the real world. A simple test of the basic I/O speed of the computer is to observe how fast the output voltages can be changed. The following program writes the sequence 0,1,2,3,...255 into the output port PO.0-PO.7. This sequence is repeated every time you type the return key.

```
/*  iospeed.cpp   */

#include <stdio.h>
#include <stdlib.h>
#include "windows.h"
```

```
#include "C:\user\ni.h" // Nat. Instr. I/O driver

int main()
{
    int i;
    uInt8 data[1];

    do {
        for( i=0; i<1000; i++) {
            data[0] = (i & 0xff);
            DigitalOut( data, 1 );
        }
        printf("done\n" );

     } while( getchar() != 'q' );

    return( 0 );
}
```

The statement "i & 0xff" is a bitwise AND function that just sets all bits in the 32 bit variable i to zero except for the lowest eight bits (see the next section on masking).

> Exp. 11.5 Type in the above program and observe the output of the least significant three bits on the oscilloscope. Record the period of the waveforms and hence estimate the I/O time of the computer. Is this what you would expect from the clock speed of the computer and USB bus? Why or why not?
>
> Repeat this experiment but use block IO. That is output a large array at one time instead of one element at a time. end

You may observe that the waveform has some jitter. This is caused by various other processes inside the computer that are competing for CPU cycles (i.e. screen refresh, DRAM refresh etc.). It's not always obvious what these are but you should be aware that CPU timing is not precise. This is a general feature of most general purpose computer systems. It is only with a dedicated microprocessor system that some degree of predictable time performance can be obtained.

This is a relatively simple program and the C/C++ compiler should do a good job at translating it. The speed is limited by the response time of the operating system and the device driver and not necessarily the real bandwidth of the hardware. (See discussion at the end of section 11.1.)

11.14 External Synchronization

The computer is executing its program with one particular clock frequency with other possible processes running in the background. In general the program will not be synchronized with an external electronics device or experiment. If the external experiment is changing its data the program must somehow synchronize itself to the external device. One method of doing this is called polling. The computer reads a ready bit (or equivalently a wait flag) to determine when the external circuit is ready. The computer keeps reading the ready bit until it has the correct value to indicate that the external data is valid. Another way of saying this is that the computer polls the ready flag and enters a wait loop until the other external circuit is ready. The next experiment will poll the debounced push button on the Digi-Designer and print the time when the button is pushed. The two **do {} while()** loops are the wait loops. **do{}** works much the same way as a **while{}** except the condition is checked after the code between the curly brackets is executed. The program reads the value of the IO port which is then bitwise AND'ed with 1 to mask out the low order bit. The single & performs a bitwise AND meaning that each bit of the mask is AND'ed with the input data.

DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	input byte
0	0	0	0	0	0	0	1	mask
0	0	0	0	0	0	0	DI0	bitwise AND result

If the high order bits (P1.7 through P1.1) of the input byte are not connected then they can be either 0 or 1 in a more or less random manner. The mask gets rid of the unused bits (P1.7 to P1.1) and allows the single bit P1.0 to be tested.

The first loop waits until P1.0 changes from 0 to 1. When this happens the program reads the system time using the standard C library function clock() and prints it to the screen. clock() actually reads the number of (integer valued) clock ticks and the symbol CLK_TCK contains the number of clock ticks per second.

```
/*  timer1.cpp   */

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "windows.h"

#include "C:\user\ni.h" // Nat. Instr. I/O driver

int main()
{
    int i;
    double time1;
    uInt8 data[1];

    for( i=0; i<10000; i++) {

        do{
            DigitalIn( data, 1 );
        } while( (data[0] & 0x01) == 0);

        time1 = (double) clock();
        printf("time = %f sec.\n", time1/CLK_TCK );

        do{
            DigitalIn( data, 1 );
```

```
       } while( (data[0] & 0x01) != 0);
    }

    return( 0 );
}
```

Exp. 11.6 Type in the program above. Connect one of the debounced push button on the Digi-Designer to the low order bit of the digital input port P1.0. Run the program and test it by pushing the button. It should print the time in seconds each time the button is pushed. Next modify the program so that it prints out the time that the button was held down. (Hint you will need to read the time twice and print the difference).

OPTIONAL: Build a digital capacitance meter using a 555 one-shot and the computer timer you just made. Connect the 555 as a one-shot using the unknown capacitor as the timing element. Trigger the one-shot from one of the debounced switches using a high pass RC filter as in chapter 10. Have the computer read the 555 output pulse (TTL digital level) and time its width. Modify the program so that it prints out the value of the unknown capacitor in μF and test it on several different capacitors. end

11.15 Listing of ni.h

```
/*    ni.h

   include file to use with National Instr. USB 6008
   data acquisition devices

   link with NIDAQmx.lib
*/

extern "C" {
#include <NatInstr\NIDAQmx.h>
}

TaskHandle taskHandle=0;    //  global task handler ID
float64 timeOut=10.0;       // in sec.
bool32 autostart=1;         // true

//----------- subroutine to check NI errors ---------------
void CheckError( int32 error )
{
    char errBuff[2048]={'\0'};

    if( !DAQmxFailed(error) ) return;
    else {
        DAQmxGetExtendedErrorInfo(errBuff,2048);
        if( taskHandle!=0 ) {
            DAQmxStopTask( taskHandle );
            DAQmxClearTask( taskHandle );
        }
        printf("DAQmx Error: %s\n",errBuff);
        printf("End of program, press Enter key to quit\n");
        getchar();
        exit( 0 );
    }
}

//----- output n digital values to NI device ----------
//
int DigitalOut( uInt8 data[], int n )
{
    int32 written;  float64 rate=1000.0;

    CheckError( DAQmxCreateTask("",&taskHandle) );   //  task

    CheckError( DAQmxCreateDOChan(taskHandle,"Dev1/port0/line0:7",
        "",DAQmx_Val_ChanForAllLines) );

    CheckError( DAQmxStartTask(taskHandle) );

    CheckError( DAQmxWriteDigitalU8(taskHandle, (int32) n,
        autostart, timeOut, DAQmx_Val_GroupByChannel, data, &written, NULL) );

    DAQmxStopTask(taskHandle);
    DAQmxClearTask(taskHandle);
    return( written );  // return number of values actually written
}


//------input n digital values from NI device -------
//
int DigitalIn(  uInt8 data[], int n )
{
    int32 read;
    CheckError( DAQmxCreateTask("",&taskHandle) );   //  task

    CheckError( DAQmxCreateDIChan(taskHandle,"Dev1/port1",
        "",DAQmx_Val_ChanForAllLines) );

    CheckError( DAQmxStartTask(taskHandle) );

    CheckError( DAQmxReadDigitalU8( taskHandle, (int32) n,
        timeOut, DAQmx_Val_GroupByChannel, data, 1, &read, NULL) );

    DAQmxStopTask(taskHandle);
    DAQmxClearTask(taskHandle);
    return( read );  // return number of values actually read
}
```

11.16 Practice Problems

[1] The circuit shown in figure 11.7 generates a pseudo-random sequence of 5 bit binary values to use as data to test a digital transmission system. Q_4 is the MSB or most significant bit and Q_0 is the LSB or least significant bit. It uses five positive edge triggered flip flops and is driven by a square wave CLOCK input. If it starts with a value of $1A_{16}$ (hexadecimal) find the next 5 values after successive rising edges of the clock n. Complete the following table and fill in the missing values in hexadecimal notation.

n	Hex
0	1A
1	
2	
3	
4	
5	

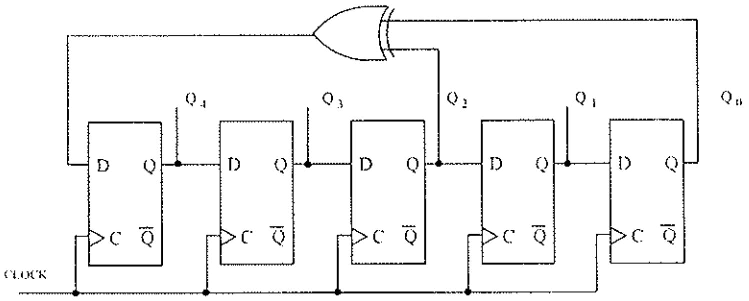


Figure 11.7: Circuit for problem [1]

Chapter 12

COMPUTER DATA ACQUISITION

The primary goal of this chapter is to explore some applications of the computer to perform analog data acquisition tasks common in the laboratory. The first half of this chapter will build a Digital to Analog Converter (DAC) and connect it to the computer, then use this DAC to construct an Analog to Digital Converter (ADC) driven from the computer. The second half of this chapter will use the ADC on the I/O board to sample external analog signals, display them on the computer screen and investigate how to average noisy signals to improve their signal to noise ratio (i.e. reduce the noise by averaging over many periods of the waveform).

12.1 Digital-to-Analog Conversion

Computers handle information in digital form, but most physical measurements appear first in analog form as a continuously variable voltage proportional (or at least related) to some physical quantity of interest (like temperature, pressure, etc.). In communication between a computer and the outside world, *analog-to-digital conversion* (ADC) and its converse, *digital-to-analog conversion* (DAC), become important.

One of several possible circuits for producing an analog output voltage, V_{DAC}, proportional to the value of the binary number represented by P0.7...P0.0 is shown in figure 12.1. This configuration is called the R-2R ladder. The output voltage is given by:

$$V_{DAC} = \frac{V_{ref}}{3 \times 2^7} \left[2^7 B_7 + 2^6 B_6 + \cdots + 2^0 B_0 \right] \qquad (12.1)$$

where $B_7 \cdots B_0$ are the values, 0 or 1, of the digital outputs P0.7...P0.0. The inputs are wired to the output connector of the computer, and are assumed to produce voltages of 0 or V_{REF}=+3.6V (TTL levels). For example, to obtain +3.6V at B_3, a 1 should be entered into the corresponding bit in the output port.

In practice this approach can only be used for a small number of bits (4 or 5). An eight bit DAC as shown above would require component values accurate to 1 part in 2^8 or 0.4%. The resistors typically used in the lab have tolerances of 5% or 10%. Furthermore, the above configuration requires that all of the digital input voltages be exactly the same value. A TTL level may be anywhere between 2.4

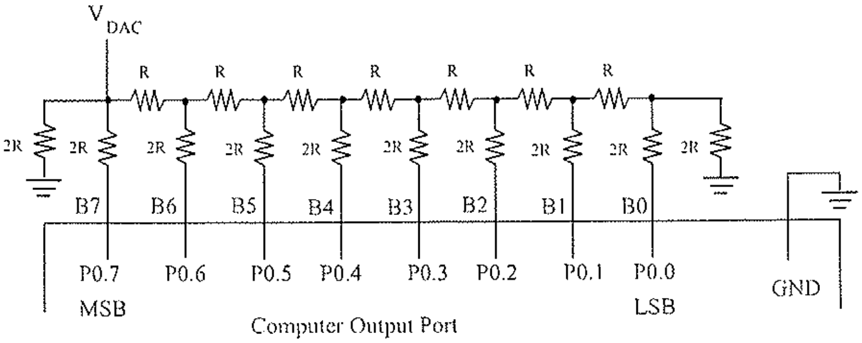


Figure 12.1: The R-2R ladder (DAC) connected to the computer.

and 5 volts for a logic 1 and 0.0 to 0.4 for a logic 0. Neither the resistors nor the digital voltages are accurate enough with off-the-self components.

Exp. 12.1 Obtain a resistor network like the one shown in figure 12.2 assembled on a 16-pin DIP plug. Connect the digital output port (8 bits) of the computer to the DAC.

You can write a short program to output digital values for testing, but it is easier to use the test program supplied by National Instruments. From the Start Menu launch the "National Instruments/Measurement and Automation" item. Select "Device and Interfaces/NI-DAQmx Devices/NI USB-6008" Then choose the "Test Panel tab and select "Digital Output". Set the following in the top two section; "Select Port=port0", "Sect Direction=All Output". In the bottom section labeled "Select State" you can output different digital values to port 0 by clicking on each button (one per bit) using the mouse (see figure 12.3).

Make a table in your lab book of the actual DAC output voltage with all bits 0 and all bits 1 and then a HI level at each successive bit (with all other bits held at 0). Compare to the value you expect from equation 12.1.

Save this circuit for the next experiment. end

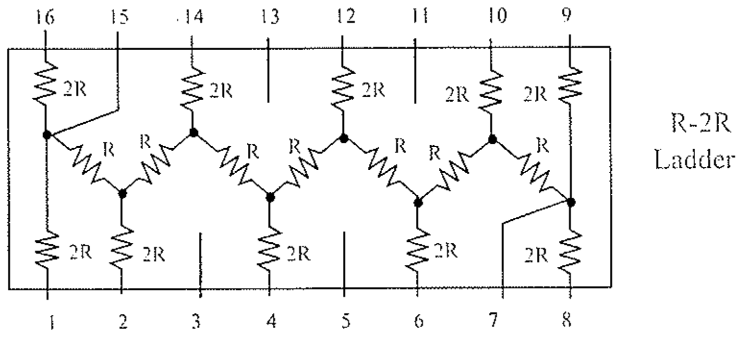
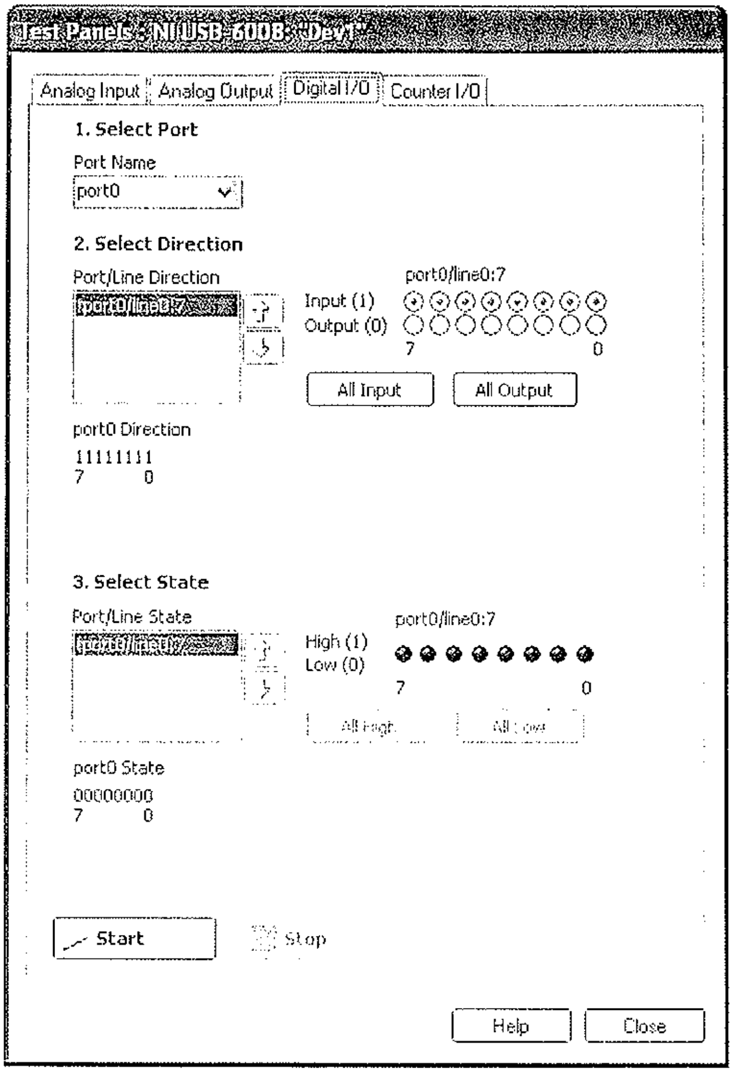


Figure 12.2: The R-2R ladder assembly.

A variation of the R-2R circuit is however ideally suited for integrated circuit manufacture. The R-2R ladder requires only that the ratio of two sets of resistors be fixed. Because the whole IC circuit



Test Panels : NI USB-6008: "Dev1"

1. Select Port

Port Name: port0

2. Select Direction

Port/Line Direction: port0/line0:7

port0 Direction
11111111
7 0

3. Select State

Port/Line State: port0/line0:7

port0 State
00000000
7 0

Figure 12.3: The National Instruments digital output test utility.

is made on one piece of silicon with the same manufacturing steps it is much easier to achieve the required accuracy in the ratio of resistor values (but not necessarily in their absolute values). Also, instead of trying to produce N accurate (and digitally controlled) reference voltages the manufacturers often turn the R-2R ladder upside down and interchange the inputs and outputs. What was the output (above) is driven from a single voltage reference and the current flowing out of what was the input is steered either to ground or a common output with current mode switches. (This inverted ladder is also slightly faster.) Most of the inexpensive DAC IC's have a current output as discussed here. This current may then be converted to a usable voltage with an external current to voltage Op-Amp circuit as in Chapter IV or a resistor if the load requirements are small.

12.2 The Ramp Generator

A computer program can be used to test a DAC by generating a repetitive ramp or staircase waveform shown in figure 12.4. The computer should cause the output port to step through the values 0..255, 0..255,.. in a regular sequence, and it should be able to run continuously. This is basically the IOSPEED program used in chapter XI repeated here and renamed ramp.cpp as below. This version will just hang the program in an infinite loop. To exit type control-C (hold down the CTRL key while pressing the C key) with the program window active. This will abnormally end the program. The inner loop runs until you type control-C. This approach makes it easier to see the waveforms on the scope. You will use this program in the next experiment. The output waveform can be observed on the scope if you trigger the sweep from the falling edge of the DAC output.

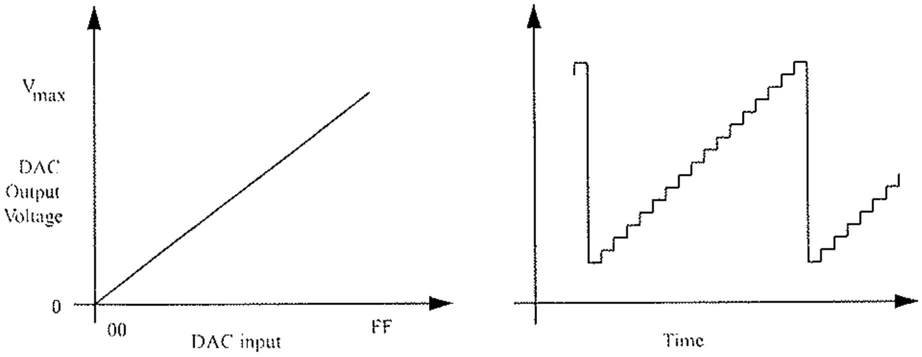


Figure 12.4: DAC ramp

```
/*  ramp.cpp   */
#include <stdio.h>
#include <stdlib.h>
#include "C:\user\ni.h"   //  NI subroutines etc.

int main()
{
    int i;
    uInt8 data[10];

    printf("start continuous ramp\n");
    while( true ) {
        for( i=0; i<256; i++) {
            data[0] = (i & 0xff);
            DigitalOut( data, 1 );
        }
    }
    return( 0 );
}
```

You may observe jitter in the staircase display on the oscilloscope even though it is properly triggered from the DAC output. This is due to the fact that every now and then, the computer steals some cycles for various other activities (such as DRAM refresh, or CRT refresh, etc). Whenever this occurs, execution of the program stops. The total time it takes to execute a staircase varies a little from one step to another. Since the time base of the oscilloscope sweep is fixed, the slight differences in time during the stepping of the staircase give rise to the observed jitter in the oscilloscope display.

Exp. 12.2 First test the performance of the ramp program using the resistor R-2R ladder circuit shown in figure 12.1 and 12.2 (from the previous experiment). Observe the DAC output voltage on the oscilloscope with the program running. If you have trouble getting a stable trace due to triggering problems, try putting a second probe on the most significant bit of the digital output from the computer and use this to trigger both traces. Note the total time and voltage range of one ramp cycle and the shape of the ramp. In particular, is it linear? Look carefully at the point where each bit changes, i.e. 1/2, 1/4, 1/8 etc of the full scale.

Next insert a DAC-0808 IC as shown in figure 12.5, run the ramp program and observe its output V_{DAC} on the scope. Record total time, voltage, and linearity of the ramp. Compare the linearity of the R-2R resistor ladder with that of the DAC-0808. Finally vary V_x and note what happens to the waveform at L.

Save this circuit for the next experiment. **end**

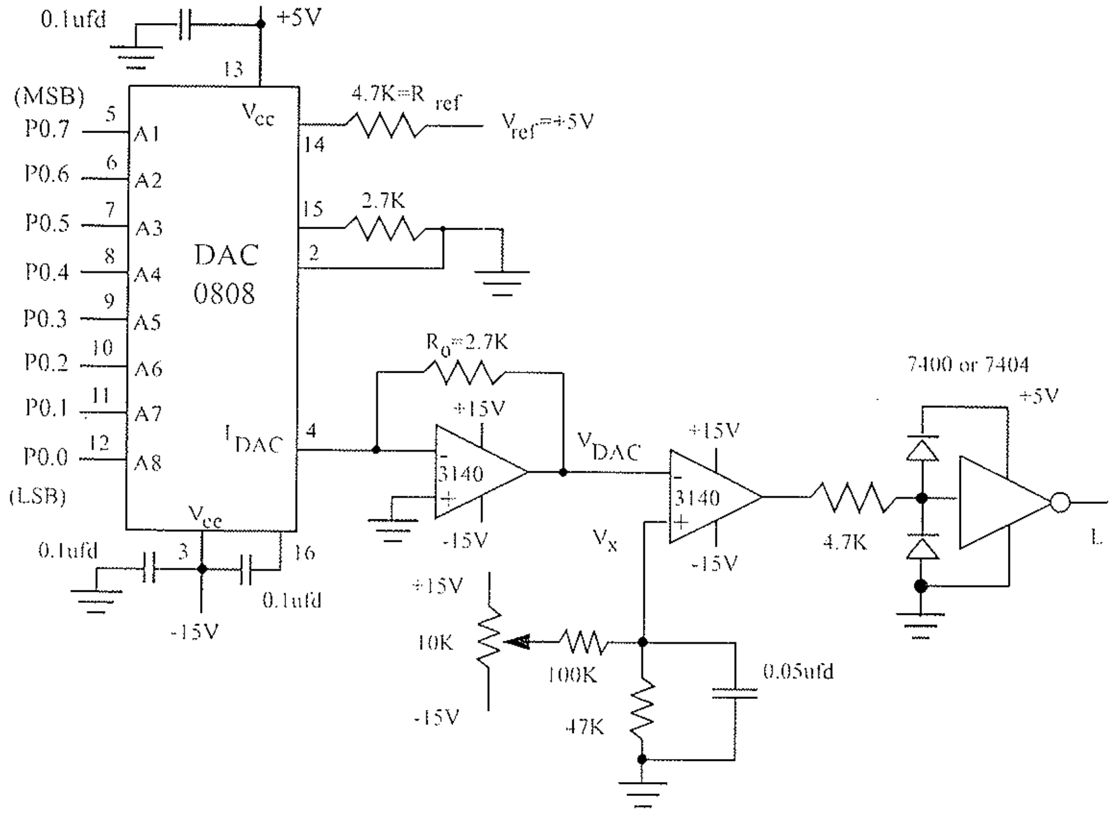


Figure 12.5: DAC circuit for Exp. 12.2. which will become the ADC circuit in Exp. 12.3 and 12.4.

12.3 Analog to Digital Conversion, the Digital Voltmeter

The staircase waveform can be used to measure an unknown voltage V_x by simply using a comparator to sense when the ramp crosses the level of V_x. The circuit shown in figure 12.5 illustrates this idea. The 3140 comparator delivers a negative-going output transition when the ramp crosses through the level V_x. The count in the output port when L becomes 1 for the first time measures the voltage V_x. This circuit will be used in the next few experiments to construct a *digital voltmeter.* In practice an eight bit ADC IC may be purchased for only a few dollars and it is generally more practical to simply buy one. However, if you have a DAC and a computer already then it is also a simple matter to construct an analog to digital converter as below. The National Instruments I/O device has an ADC IC circuit on it already and after learning how an ADC works, the on-board ADC will be used it in the last half of this chapter.

To display the voltage values measured by the ramp on the screen monitor, it is necessary to write another program that watches the signal L above using an input port. The procedure is to increment the value sent to the DAC and test whether the comparator has changed states. Bit 0 of the input port is connected to L and changes when the staircase voltage crosses above the unknown voltage. If it has crossed, there is no point in continuing the ramp to higher voltages: the necessary information has already been acquired. Hence the program is arranged so that it loops back to the incrementing instruction only while the staircase is still below V_x. Once the threshold is crossed, the incrementing is stopped and the value is displayed on the screen. The actual program is listed below.

```
/*  adc.cpp   */
#include <stdio.h>
#include <stdlib.h>

#include "C:\user\ni.h"  //  NI subroutines etc.

int main()
{
    int i;
    uInt8 data[1];

    while( true ) {
        for( i=0; i<256; i++) {
            data[0] = (i & 0xff);
            DigitalOut( data, 1 );  // write DAC value
            DigitalIn( data, 1 );   // read 3140 comparator
            if( (data[0] & 0x01) == 1 ) break;
        }
        printf("adc= %5d\n", i );
    }

    return( 0 );
}
```

The program repeats the main while(){} block until the ESCAPE key is pressed. The for() loop continually increments the variable i and checks the status of the L signal attached to bit zero of the input port. The expression (**data & 0x01**) is a bitwise logical 'and' (i.e. on a bit-by-bit basis) of the value read from the input port and 1. In other words, if the port contents is the 8-bit number $A_7 \cdots A_0$, and the logical-AND is taken with a number $M_7 \cdots M_0$ from memory, then the n^{th} digit of the result X will be:

$$X_n = A_n \cdot M_n \tag{12.2}$$

In this program, $M_7 \cdots M_0$ is the decimal number 1, which in binary notation is 0000 0001. Thus all binary digits of M are 0 except bit 0. Because $0 \cdot A_n = 0$ all digits of the result are forced to become 0 by the zeros in M, except for $X_0 = 1 \cdot A_0 = A_0$.

In this process, M= 1 is called a *mask* for A. Its binary form 0000 0001 makes it clear that this particular mask blocks out all digits of A except A_0, which it allows to shine through. In this manner, by choice of a suitable mask, you can pick out any bit (or combination of bits) from a many-digit number. The **break** statement causes the program to exit from the current loop, so when P0.0 changes state the program exits the for() loop.

$\boxed{\text{Exp. 12.3}}$ Modify the circuit of Exp 12.2 using the DAC-0808 and connect the output L back into bit 0 of the input port. With the voltmeter adc program (shown above) running, measure directly the voltages at the non-inverting input of the comparator, V_x. Make a table and a plot of V_x (the y-axis) versus the contents of the output port (the x-axis) displayed on the screen as decimal numbers. From the graph, obtain the best straight line fit to the data. Using the results of this fit, modify the program so that the computer displays the voltages measured by the staircase circuit in the units of volts. Measure the sampling error of your voltmeter and compare to your calculated value. Give explanations for the sources of errors and any nonlinearity observed in the V_x versus the output port content relationship.
Save this circuit for the next experiment. $\boxed{\text{end}}$

12.4 Digitization by Successive Approximations

The ADC algorithm used above for measuring a voltage V_x digitally is laborious. It steps through every possible value until it finally reaches and exceeds the level of V_x. If V_x is near the top end of the range and high resolution (i.e. many steps) is required, the process can be quite slow. It is much more economical to use the *successive-approximation* algorithm: try a value of X equal to half its maximum first, to see whether V_x is in upper or lower half of the range. Then explore the appropriate quarter, eighth, etc. portion of the range, until finally the nearest digital value to V_x itself is found. If X is an 8-bit number, for example, try the most significant bit first as in table 12.1. If V_x is still larger, leave that bit set at 1; otherwise return it to 0. Then work with the next lower bit, again leaving it so X is below V_x. After 8 such tests, the value of X is determined. In the staircase algorithm up to $2^8 = 256$ steps might be required. The ADC on the I/O board in fact uses a successive approximation technique (in hardware not software). If stated as an algorithm, this method is formally called bisection.

The successive-approximation algorithm is a bit trickier to write. A possible implementation is shown below.

D7(MSB)	D6	D5	D4	D3	D2	D1	D0 (LSB)	SA bits
T	0	0	0	0	0	0	0	step 0
S	T	0	0	0	0	0	0	step 1
S	S	T	0	0	0	0	0	step 2
	:				:			
S	S	S	S	S	S	T	0	step 6
S	S	S	S	S	S	S	T	step 7

Table 12.1: Steps in successive approximation. T=1 is a test and S=0 or S=1 is a bit that has already been testing and set to a specific value. MSB= most significant bit and LSB= least significant bit.

```
/*  adcsar.cpp   */

#include <stdio.h>
#include <stdlib.h>
#include "C:\user\ni.h"  //  NI subroutines etc.

int main()
{
    int i, dacbits, adcval;
    uInt8 data[1];

    while( true ) {
        dacbits = 0x80;
        adcval = 0;
        for( i=0; i<8; i++) {
            data[0] = (dacbits + adcval) & 0xff;
            DigitalOut( data, 1 );  // write DAC value
            DigitalIn( data, 1 );   // read 3140 comparator
            if( (data[0] & 0x01) == 0 ) adcval = adcval + dacbits;
            dacbits = dacbits/2;
        }
        data[0] = 0;
        DigitalOut( data, 1 ); // to sync the scope
        printf("adc= %5d\n", adcval );
    }

    return( 0 );

}
```

Exp. 12.4 Using the circuit of Exp 12.3 (with the DAC-0808) implement the successive approximation method using the program above.

Put one scope probe on the DAC output voltage V_{DAC}. Sketch the trace in your lab book and correlate V_{DAC} with the output of the program for different values of V_x. Why does a given sequence of 0's and 1's produce the trace shown for V_{DAC}?. (Remember that the program can output integers in Hex using the %x format.) end

12.5 The Digital Sampling Scope

An ADC can be used to sample a waveform and display it on the computer much like an oscilloscope. This has the added advantage that the resulting data may then be used directly for further analysis. The computer could precisely record many thousands of data points that would otherwise be impossible to record by hand. There are also commercial digital sampling scopes that are capable of sampling signals at several billion samples per sec. (several hundred GHz). These devices use a great deal of sophisticated hardware and very little software. The device to be built next is just the opposite. It is mainly implemented in software and can only sample signals at a few tens of thousand samples per sec (approx. 10kHz).

The basic idea behind digital sampling is to arrange for the ADC to convert an external analog signal at fixed time intervals and store the results in memory. The graph shown in figure 12.6 illustrates this principle. At times $n(\Delta t)$ where n=0,1,2,3... is an integer and Δt is the sampling interval the external analog signal is digitized and stored in a memory array with corresponding index n. The position in memory is proportional to the corresponding position in time. The NI USB device does this by itself and sends a block of data back to the computer. This is much faster and does not rely on the computer for accurate timing.

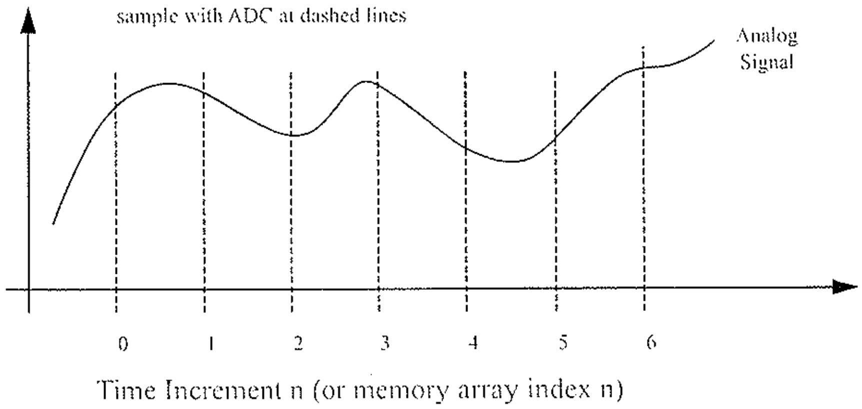

sample with ADC at dashed lines
Analog Signal
0
1
2
3
4
5
6
Time Increment n (or memory array index n)

Figure 12.6: Signal sampling

The program dscopeUSB.cpp implements a digital sampling scope on the computer and will be provided on computer disk. A complete listings is given here for your information. This program uses the OpenGL and GLUT (GL utility toolkit) libraries to draw a graph on the screen. OpenGL is included with Windows but the GLUT library must be downloaded separately and installed. It is actually easier to open a window using OpenGL/GLUT than with the actual Windows calls. OpenGL also has the advantage that it is portable to some extent. Nearly identical subroutine calls work the same on many different computer platforms (Apple Mac's, Linux, Unix, etc.) because various people have implemented the libraries on many different computer systems. For the most part you can ignore the graphics calls in the program and focus on how the ADC is used. For those who are interested some references to OpenGL are included in the beginning comments in the code.

The I/O device has built-in memory and control circuits to digitize a whole block of data at one time using a fixed sampling rate. This avoids the problem of timing jitter when executing a program in a general purpose computer (with various other programs running at the same time). The program has several new calls to the National Instruments subroutine library, but you can mostly ignore these as well.

```
/*
        dscopeUSB.cpp

   read the NI USB-6008 ADC and
   plot a graph using OpenGL graphics for Windows NT/2000/XP

   requires that the OpenGL Utility Toolkit be installed
   (downloaded from: http://www.xmission.com/~nate/glut.html)

   NOTE-1: Windows NT comes with the openGL library but NOT the
         utility toolkit "glut" which must be installed by hand

   OpneGL/GLUT references:
    [1] M. Woo, J. Neider, T. Davis and D. Shreiner, "OpenGL Programming
       Guide, third edit., the official guide to learning OpenGL, version 1.2",
       Addison Wesley 1999 (the "red book").

    [2] D. Shreiner, edit., "OpenGL Reference Manual, third edit., the official
       reference document to OpenGL, version 1.2", Addison Wesley 1999
       (the "blue book").

    [3] www.opengl.org

   started from dscope.cpp 11-nov-2006 E. Kirkland
   last modified 13-nov-2006 ejk
*/

#include <math.h>
#include <stdio.h>
#include "windows.h"    // Windows libraries
#include "C:\user\ni.h" // Nat. Instr. I/O driver

#include <GL/glut.h>    // for OpenGL and GLUT libraries

// ----- global data ----------

const int NXwindow=600;           // initial window size
const int NYwindow=300;

const int NPOINTS=512;                // number of data points
double xpt[NPOINTS], ypt[NPOINTS];    // normalized data for plotting
float64 sumdata[NPOINTS];             // to use in signal averager

//----------- readADC() --------------
//
```

```
//  read a block of data from the ADC on the NI USB-6008
//   and return values in data[]
//  using the Nat. Instr. subroutine library (requires ni.h)
//  return number of points actually read
int readADC(  float64 data[], int n )
{
    int32 read;

    CheckError( DAQmxCreateTask("",&taskHandle) );   //  task
    //  select voltage range from -10 to +10 volts
    CheckError( DAQmxCreateAIVoltageChan(taskHandle, "Dev1/ai0",
        "", DAQmx_Val_Cfg_Default, -10.0, 10.0, DAQmx_Val_Volts, NULL));
    //  select 10kHz sampling, rising edge trigger,
    CheckError( DAQmxCfgSampClkTiming( taskHandle,
        "", 10000.0, DAQmx_Val_Rising, DAQmx_Val_FiniteSamps, (uInt64) n));
    //  trigger from rising edge of PFI0
    CheckError( DAQmxCfgDigEdgeStartTrig(taskHandle,
        "/Dev1/PFI0", DAQmx_Val_Rising));

    CheckError( DAQmxStartTask(taskHandle) );

    CheckError( DAQmxReadAnalogF64( taskHandle,
        -1, timeOut, 0, data, (uInt32) n, &read, NULL) );

    DAQmxStopTask(taskHandle);
    DAQmxClearTask(taskHandle);
    return( read );  // return number of values actually read

} //  readADC()

//----------- OnKeyPressed() --------------
//
//  GLUT function to update the date when a key is pressed
//
void OnKeyPressed( unsigned char key, int xmouse, int ymouse )
{
    int i;
    float64 ydata[NPOINTS];
    double scaley, ymin, ymax;

    if( key == 'q' ) exit( 0 );      // quit if 'q' is pressed

    readADC( ydata, NPOINTS );          // get the data from the ADC

    //  transfer to graphing arrays
    for( i=0; i<NPOINTS; i++) {
        xpt[i] = ((double)i)/((double)(NPOINTS-1));
        ypt[i] = ydata[i];
    }
    //  scale the data for graphing
    ymin = ymax = ypt[0];
    for( i=0; i<NPOINTS; i++) {
        if( ypt[i] > ymax ) ymax = ypt[i];
        if( ypt[i] < ymin ) ymin = ypt[i];
    }
    printf( "ADC range = %f to %f\n", ymin, ymax );

    if( ymax>ymin) scaley = 1.0/((double)(ymax-ymin));
    else scaley = 1.0;
    for( i=0; i<NPOINTS; i++) {
        ypt[i] = scaley * ( ypt[i] - ymin );
    }

    glutPostRedisplay();        // update the window

} // end OnKeyPressed()

//
//---------  function to draw the graph in the window ----------
void display( void )
{
    int i;
    glClear( GL_COLOR_BUFFER_BIT );
    glColor3f( 0.0, 1.0, 0.0 );      // select green for drawing

    glBegin( GL_LINE_STRIP );
        for( i=0; i<NPOINTS; i++) glVertex2d( xpt[i], ypt[i] );
    glEnd();

    glFlush();
} // end display

//
//-------- main program --------------------
int main( int argc, char **argv )
{
    int i;

    printf( "dscope.cpp: plot ADC values in OpenGL\n"
        "press return to read ADC, q to quit\n" );

    for( i=0; i<NPOINTS; i++) sumdata[i] = 0;   // for signal averager

    // ------ initialize the OpenGL environment -----------------
    glutInit( &argc, argv );
    glutInitDisplayMode( GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH );

    glutInitWindowPosition( 100, 100 );
    glutInitWindowSize( NXwindow, NYwindow );
    glutCreateWindow( "dscope" );

    glClearColor( 0.0, 0.0, 0.0, 0.0 );         // clear to black
    glMatrixMode( GL_PROJECTION );
    glLoadIdentity();

    // set to simple 2D orthographic projection scaled to 0.0->1.0
    //  in both x and y with a small border
    gluOrtho2D( -0.05, 1.05, -0.05, 1.05 );

    glutDisplayFunc( display );         // hook the display function

    glutKeyboardFunc( OnKeyPressed );   // hook update on key-pressed

    glutMainLoop();    // let OpenGL take over

    return 0 ;

}  // end main()
```

This program looks rather long but it is mostly tedious bookkeeping. You do not need to understand all of the details of the program but a brief summary is presented here. There are several subroutines at the top and the main program is at the bottom. **main()** starts out with a lot of calls to subroutines beginning with 'gl...', which are the OpenGL subroutines. The first few initialize openGL and the last few set specific functions for this program. The function **glutDisplayFunc()** function sets the function to use for displaying the window, **glutKeyboardFunc()** sets the function to be called every time a key is pressed and **glutMainLoop()** passes control to OpenGL (this function never returns). To uses these routines you must also link your program with the libraries **opengl32.lib, glu32.lib** and **glut32.lib** using the same procedure as for adding **NIDAQmx.lib** in section 11.10

The **readADC()** function sets up the ADC and reads back the data into an array passed as its argument. The ADC is synchronized to the function generator with a digital transition on the PF1 input. This is the same as using the external sync input on the oscilloscope.

OnKeyPressed() is called every time a key is pressed. If you press the 'q' key then the program ends, otherwise this subroutine gets a new set of data using **readADC()** and the updates the window.

You will need to connect the function generator to one of the ADC channels on the I/O device as shown in fig. 12.7 (also connect the ADC ground to the ground on your breadboard and to the function generator ground). You will also need to synchronize the computer to the generator by connecting the sync-out from the generator to a circuit as shown in figure 12.7. This serves to convert the synch signal into a logic level signal that the digital input on the computer understands (i.e. the sync-out is not a TTL level on all of the function generators).

Exp. 12.5 Construct the circuit shown in figure 12.7. Connect the function generator signal (sine, square, triangle output) to the ADC input (labeled AI0 or AI0+). This device defaults to a differential input, so the analog input signal labeled AI4 (or A0-) should also be connected to ground.

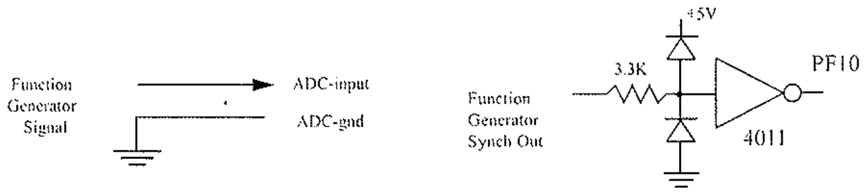


Figure 12.7: Circuit for Exp. 12.5.

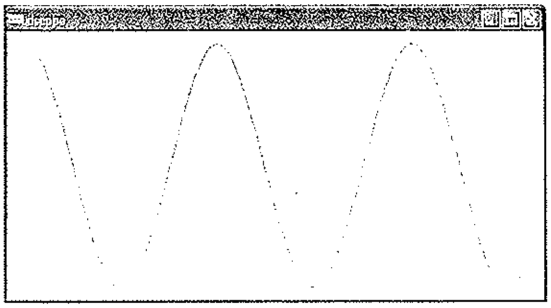


Figure 12.8: The output window of dscope.cpp

Next connect the synch out (from the function generator) to the CMOS gate (with +5V power) circuit to match the voltage levels going to the digital input PFI0 (used to synch or trigger the ADC). This circuit insures that the voltage levels are 0V to 5V for the TTL digital input of the DAQ device (anything larger than this will burn out the input). Do NOT connect ±15 V to the digital inputs on the device. (Some function generators have different voltages on the synch out.)

You will need this circuit for the next several experiments that involve building a large circuit. It is best to build this near one end of the breadboard to leave enough room for the next circuits (the final circuit uses almost all of the breadboard).

Obtain a copy of the program dscopeUSB.cpp (from C:\users). Work from a copy of this program and NOT the original. With the frequency set to about 100Hz and the amplitude set to 2 V_{pp}, run the program and observe the resulting computer display for sine, square and triangle waves. Every time you type a key other than 'q' the screen will redraw another waveform as in figure 12.8. Investigate the performance of the ADC for input voltages from the minimum amplitude to about 5 V_{pp}.

Calculate the voltage sampling error ΔV from the ADC trace on the computer screen and the amplitude in volts (measured on the scope). With a low input voltage the steps in the sine wave are the ADC sampling levels. (The ADC is sensitive enough to pick up noise from other electronics devices, such as the computer monitor so a careful circuit layout may be helpful.)

The ADC is set to sample at a rate of 10 kHz. This means that an input signal of up to nearly one half of this frequency may be digitized correctly. With the function generator set to deliver sine waves, increase the frequency to see what happens at high frequency. You should see that the digitized waveform picks up various low frequency components that are incorrect. (Hint: try input frequencies near the sampling rate and its integer multiples.) This phenomenon is called aliasing (as in figure 12.9). High frequencies are aliased as low frequencies with an inadequate sampling rate. When digitizing a signal it is important to limit the maximum input frequency to less than half of the sampling rate.

Leave this set up for the next experiments. $\boxed{\text{end}}$

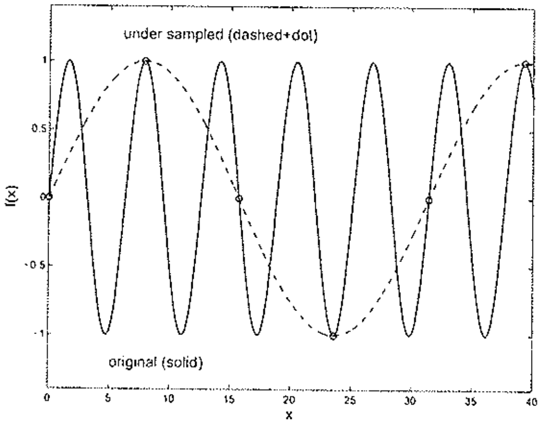


Figure 12.9: The effect of under sampling and aliasing. A high frequency (solid curve) sampled at a low frequency (dots) appears as if it were a low frequency (dashed line).

12.6 Signal Averager

A frequent problem in electrical measurements is that of retrieving a relatively feeble signal which is "buried" in noise. The noise can be caused by *interference* from a nearby electrical source, which in principle could be removed. However, an irreducible minimum amount of noise, due to *thermal excitation* of the signal-processing circuits themselves, is always present. In any resistor at absolute temperature T, this *Johnson noise* has power:

$$P_{noise} = 4kT(\Delta f) \qquad (12.3)$$

where k is Boltzmann's constant (1.38×10^{-23}J/K) and Δf is the *bandwidth* (in Hz) over which signals are accepted. Because thermal noise has a uniform *power density* (W/Hz) across the frequency spectrum, such noise is called *white*. A numerical example: a resistor operating at room temperature has a noise power of 1.6×10^{-12} W over a bandwidth of 100 MHz. For a 1kΩ resistor this corresponds to an *rms noise voltage* of 40 μV. Chapter 7 of Horowitz and Hill[6] has a good discussion of building low noise precision electronics and section 7.11 to the end of the chapter discusses noise in resistors, op-amps and other components.

Many devices, notably semiconductors, produce more noise than the thermal minimum. They do this particularly at low frequencies, where sometimes the noise power per unit bandwidth increases as 1/f. Such noise is often called *1/f noise, flicker noise, or excess noise.* Most noise is *random* in character. Thus if the instantaneous noise voltage were measured at a circuit terminal many times, the results would be as likely to be positive as negative. If you add N such results, their sum does not increase in proportion to N; instead, it tends to hover around zero, sometimes increasing, sometimes decreasing. However, because of statistical fluctuations, the magnitude of the sum does tend to grow. In the same way as happens for a random walk of N steps, the sum is likely to grow in proportion to $\sqrt{N}$.

Assume now that the *signal* of interest is *repetitive.* If you sample its instantaneous value *at the same relative point* in N cycles, the sum of these values is proportional to N. For a mixture of a repetitive signal and random noise, the sum of N samples taken at similar points of the signal will favor the signal relative to the noise: the signal contribution grows as N, while the noise contribution grows only as $\sqrt{N}$. By summing enough samples you can enhance the signal-to-noise ratio greatly (by the factor $\sqrt{N}$). Many experiments can be set up to take advantage of averaging as in figure 12.10. It is important that the probe signal occurs at known points in time so that the response of the experiment can be measured synchronized with the probe even though the response may be very noisy.

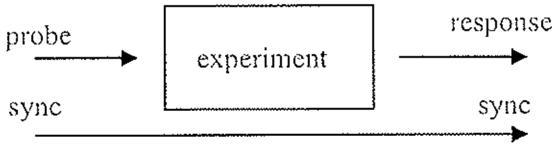


Figure 12.10: Averaging the response of an experimental system using a repetitive probe signal.

This procedure yields an improved measurement for the value of the signal at *one* particular point in its cycle. To reconstruct the whole cycle requires that you do the same thing, independently, for many different points. The number, k, of separate points needed to provide the desired detail within the cycle depends on many factors. Often k is restricted by the technical limits of the apparatus. Note, however, that one sample for each of the k points can be squeezed sequentially into a single cycle of the signal. In this way N signal cycles produce N samples at each of the k points.

A device which enhances the signal's contribution relative to that of the noise by such a process is called a *signal averager*. Its main components are:

1. a sampling device which can be set to measure the instantaneous voltage at a specified time
2. an arrangement for selecting the sampling times relative to the times of arrival of the beginning of the signal;
3. k channels of memory, in which the k independent sums of samples taken at selected times can be stored; and
4. a method for displaying the final results.

The next experiment will construct a rudimentary signal averager, by making small changes to the program **dscopeUSB**. First a noisy signal must be manufactured and then added to a signal (function generator) to model an experiment as in figure 12.11. The sync output of the function generator provides the sync input to the computer (signal averager) for averaging in software.

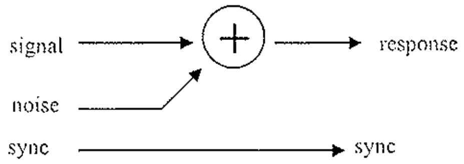


Figure 12.11: Making a noisy signal to test signal averaging.

12.6.1 Noise Generator

Many semiconductor devices will produce AC noise, particularly if they are biased in a marginal state, halfway between ON and OFF. A good noise source can be manufactured by picking off this AC component (capacitive coupled) and amplifying it. However, there is a large variation between different components (even of the same type), so it is not that reliable. You can also generate noise using digital methods. Digital circuits are inherently deterministic, so the signal will never be truly random, but instead it is pseudo-random. It will have many of the properties of random noise (good enough for our purposes at least) but it is predictable to some extent.

The specific noise generator method is modeled after that given in Horowitz and Hill, *The Art of Electronics 2nd edit.* (section 9.32 to 9.37). If two points in a long shift register, labeled M and N in figure 12.12, are combined in an XOR gate and feed back into the input, then the result is a pseudo-random sequence of 1's and 0's. Most choices of M and N will result is sequences that repeat in a few clock cycles and are not very interesting. There are a few special values of M and N, however that produce sequences that only repeat after approximately 2^N clock cycles. If N is large then the sequence appearing at the output (or any bit) is a psendorandom sequence. N=31 and M=18 are two such numbers (this can be shown using advanced number theory, but is somewhat beyond the scope of this course). This type of circuit is called a linear feedback shift register. If run from the 100 kHz clock on the Digi-Designer then this sequence repeats in $2^{31} \sim 2 \times 10^9$ clock cycles or about 6 hours!

The 4006 IC is an 18 bit shift register. The internal stages are arranged in a slightly strange order as shown in figure 12.13. This order, however works well for the pseudo-random noise generator. The final circuit is shown in figure 12.14.

There is one problem with this circuit. The state where all bits are 0's is a stable state and will produce a constant output. This circuit seems to get into this state at power-up more frequently than it should (it can't get into this state when running normally). The extra XOR gate at the top left of the schematic (connected to the logic switch) is to get it started if necessary. If the circuit has a constant output at power-up, just toggle the switch to get it started. The pseudo-random digital sequence is passed through a buffer gate and a low pass filter to get rid of its digital nature (in the upper right of the schematic). The voltage follower buffers this signal and makes a good voltage source VN with

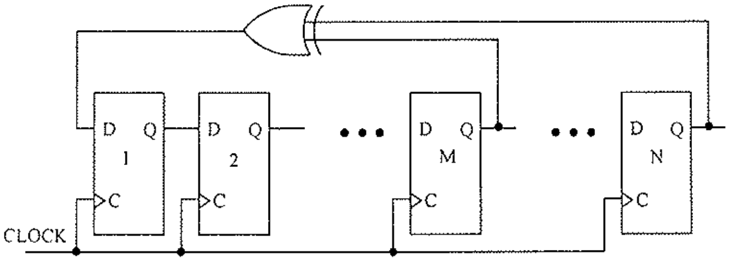


Figure 12.12: Linear feedback shift register. If N and M are chosen carefully, then the output is a pseudo-random sequence.

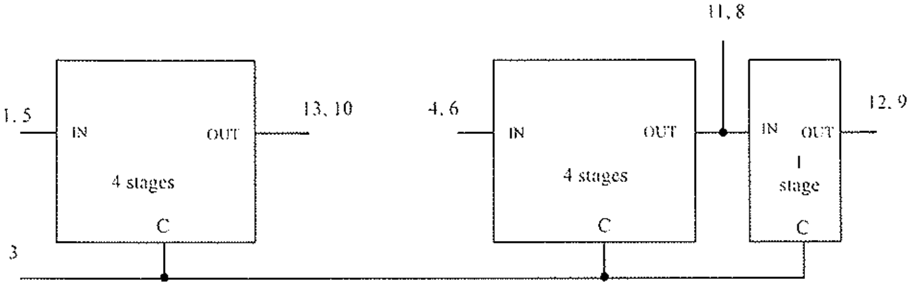


Figure 12.13: The 4006 shift register (18 bits). There are four sets of four stage shift registers. Two of them also have a one stage register attached as shown. Note that this diagram should be repeated twice. The first set of numbers are the pins for one set and the second set of numbers are the pins for the second set.

the noise signal. The noise VN is summed with the signal VS to produce a signal with a large noise component at $VN + VS$.

Exp. 12.6 Construct the digital noise generator circuit shown in figure 12.14. Use the Digi-Designer clock at 100 kHz. Keep the function generator input (VS) between about 1V to 3V in amplitude and about 100 Hz in frequency. Examine the digital noise and the analog signal plus noise $VN + VS$ on the scope. Adjust the function generator amplitude until its signal component is considerably smaller than the noise (i.e. signal not visible).
OPTIONAL: Most of the digital scopes (Tek TDS-1002) can calculate the spectrum of the input signal using an FFT (fast Fourier transform). Try using this option to measure the spectra of the noise. Is it a white noise source (flat spectra)? end

Exp. 12.7 Copy the dscopeUSB.cpp program into a new file with another name. You should edit only this new copy and not the original program. This way if you make a mistake you can always go back to the original. With the computer synchronized to the function generator as in Exp 12.5 connect $VN + VS$ to ADC channel 0 and run the copied dscopeUSB.cpp program. Do NOT connect voltages larger than 0V to 5V (i.e. TTL) to the digital inputs of the computer. Next modify the new program so that it will average the noisy signal.

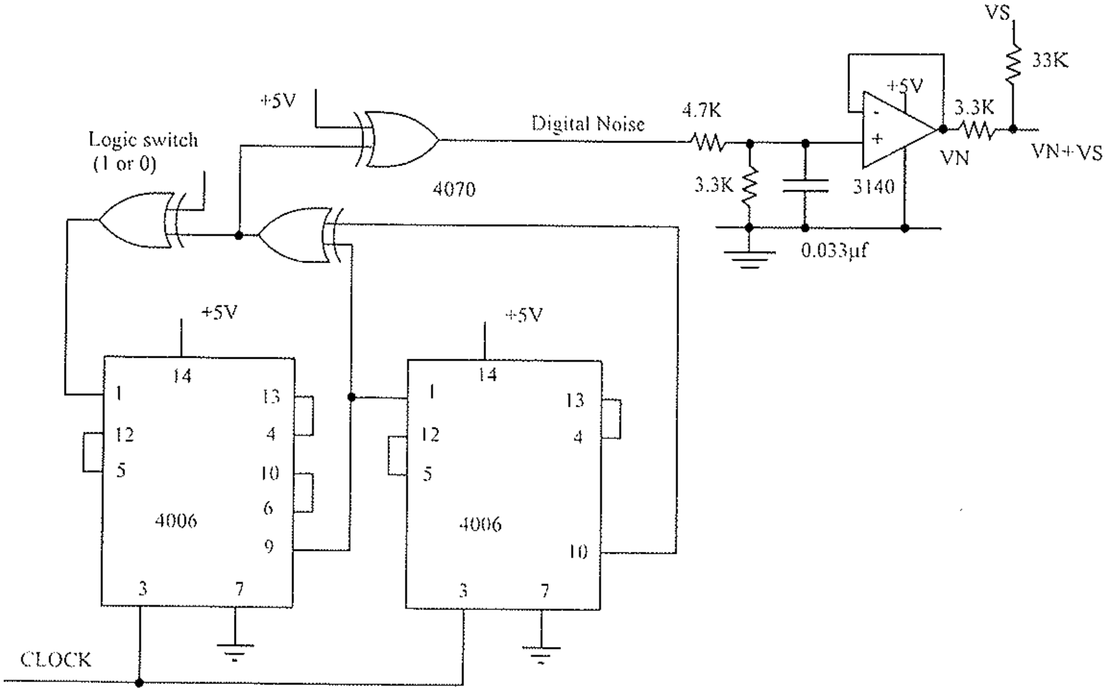


Figure 12.14: The digital noise generator plus summing circuit. The 4006 shift registers are shown in condensed form for simplicity. The 4006 on the left is 18 bits long and the 4006 on the right is 13 bits long for a total of 31 bits.

Hint: the global array sumdata[] is already set up to sum into (it is 64 bits deep so it won't overflow for a long time). It is zeroed in main(). You only have to make a small change inside OnKeyPressed() to sum successive passes of the ADC. You can add together about 10 passes of the ADC before displaying the results.

With the averaging program running investigate various waveforms etc. Your signal averager should be able to recover the original signal after it has been totally obscured by noise.

OPTIONAL: Try using the output of the radio (from chp. 6) set to some strong station as a noise source. The radio may pick up the sync. signal (if you are not very careful), which is a correlated noise source so would not average to zero, but other signals should be uncorrelated.
end

12.7 Practice Problems

[1] The circuit shown in figure 12.15 is a simple ADC circuit that converts the analog input voltage V_X into a digital output Q_5 through Q_0. Both the START and RESET pulses are narrow and RESET occurs before START (both are given once per conversion). The input CLOCK is a 1 MHz square wave. R_1= 2.2K. R_2= 4.7K. R_E= 10K. C_0= 0.01μF and V_{CC}= 5V. You may assume a base-emitter diode drop of 0.6 volts and $\beta >> 1$.

a) On the same time scale sketch, RESET, START, V_C and COUNT.

b) Calculate the charging current I_0.

c) If V_x=1.0 Volts, what is the value of $Q_5, Q_4, Q_3, Q_2, Q_1, Q_0$ a sufficiently long time after the start pulse?

d) What is the range (min and max) of voltages that this converter may accurately convert?

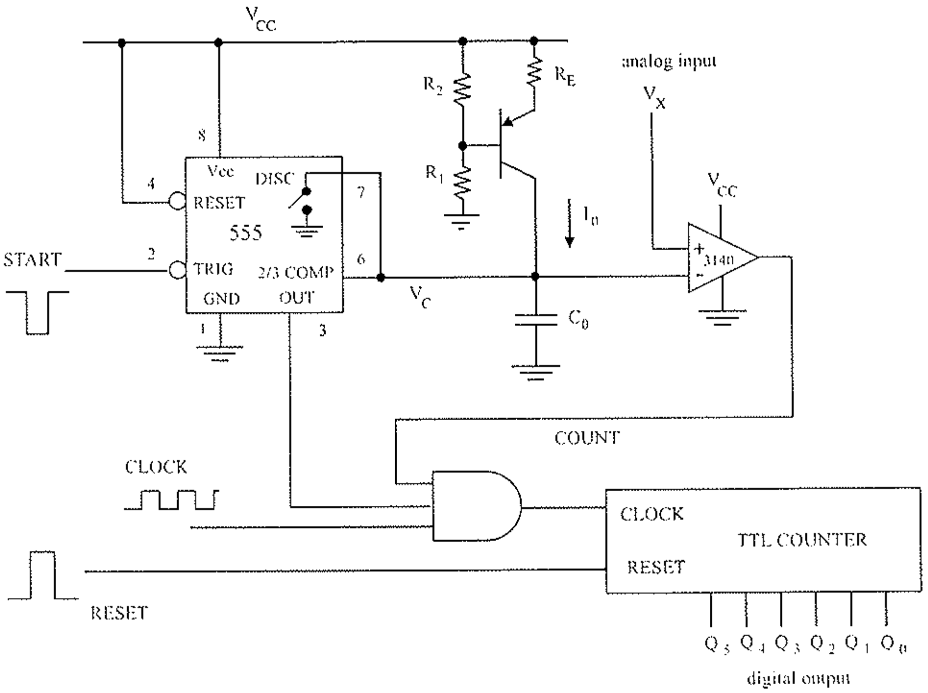


Figure 12.15: An ADC converter using a 555, a transistor current source and a counter.

Appendix A

Data Recording and Laboratory Notebooks

In general, the purpose of a laboratory notebook is to produce a record of your work that is complete enough so that someone else with a similar background may accurately reproduce the experiments you performed. Laboratory notebooks are important in many branches of engineering and science, and learning how to record your work in a laboratory notebook is a fundamental part of the lab work in this course. In many ways a lab notebook is like a diary of what you have done in the lab. There should be some informal statements about how things where set up and what happened when you did the experiment as well as an accurate quantitative record of the results.

For the purposes of this course, your lab book should be complete enough so that the grader does not need to refer to anything else (i.e. the lab manual) to understand exactly what you have done. Also note that you may be doing your analysis a week after doing the lab experiments, so a good lab notebook will help you to remember the experiments. To facilitate grading please use the following style: a) Data taken in lab should be recorded on the right hand page (left hand page left blank during lab). b) Analysis and discussion of your recorded data (from the right hand page) is written on the left hand page.

Your recorded data (right-hand page) should include the following:

1. A drawing of the circuit schematic with measurement points and components clearly indicated.

2. A list of measured component values (i.e. resistors and capacitors) and component types (i.e. which IC or transistor did you use) as well as a numerical calculation of the component values not specified in the lab manual.

3. A sketch of the input and/or output waveforms. Be careful to record the appropriate voltages and times and, in particular, indicate where ground potential is on your drawing.

4. Any other observation you consider important. In some ways, this portion of a lab notebook is like a diary of what you did in the lab. This does not have to be polished prose, but is an informal (but readable) recording of what you did in the lab.

5. Some errors and mistakes are expected in the data recording section. Just cross out errors (you never know they may turn out to have important data when you do the analysis).

6. You may want to perform some analysis in lab to check whether the experiment is working or not.

7. It's also a good idea to date each entry, to help remember what you did later on.

Your analysis (left hand page) should include the following:

1. Answer all questions explicitly asked for in the experiment description. You do not need to repeat any information already recorded in the data section.

2. Wherever possible try to compare quantitative and/or qualitative theoretical predictions and/or calculations with your experimental data and comment on any discrepancies. You should list the appropriate equations and plug in the numbers and maybe write a few sentences of explanation stating where the equations came from. You should also comment on whether the quantitative predictions of these equations agree with experiment.

3. Compare measured component properties with the manufactures data specification sheets. Try to be quantitative where possible.

4. DO NOT repeat verbatim the discussion lab manual. State a clear and concise explanation of the circuit (what it is supposed to do and whether or not it performed as expected) in your own words. This is usually a few sentences or a paragraph.

5. Analyze any other important data.

Some specific details of this style are unique to this course, but the general concepts are applicable to most forms of laboratory date recording. A sample of a experiment is shown on the following page. Obviously, what and how much you write will depend on the experiment. This example is a relatively simple experiment so there is not a lot to write about. Later experiment may require more discussion.

Exp. 1-X Analysis

The RC charging equation (from Lab Manual) is:

$$V_c = V_F + (V_I - V_F)\, e^{-t/\tau}$$

where
$$\begin{cases} \tau = RC : \text{ time constant} \\ V_c : \text{ Voltage across capacitor} \\ V_I : \text{ initial voltage} \\ V_F : \text{ final voltage.} \end{cases}$$

To get the initial rate of rise take the derivative (with respect to time) at $t=0$:

$$\frac{d}{dt} V_c \Big|_{t=0} = (V_I - V_F)\left(-\frac{1}{\tau}\right) e^{-t/\tau} \Big|_{t=0}$$

$$= \frac{V_F - V_I}{\tau} = \frac{9.84^V}{(9.86\,K)(9.65\,nFd)}$$

$$= 1.03 \times 10^5 \ \frac{\text{Volts}}{\text{Sec}} = \boxed{0.103 \ \frac{\text{Volts}}{\mu\text{sec}} \quad (\text{Theory})}$$

Compared to a measured value of:

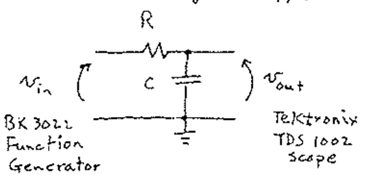
$$\frac{\Delta V}{\Delta t} = \frac{3 \times 500\,mV}{3 \times 5.00\,\mu\text{Sec}} = \boxed{0.100 \ \frac{\text{Volts}}{\mu\text{sec}} \quad (\text{Experiment})}$$

⇒ agreement between theory and experiment is very good. (~3% which is approx. accuracy of the scope) ✓

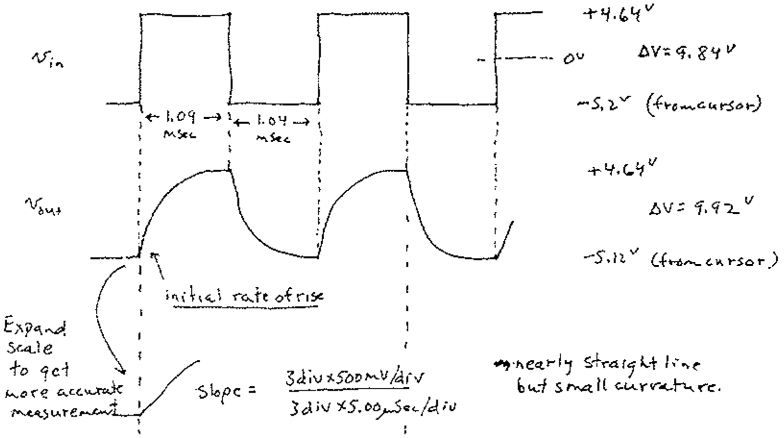
EXTRA: From the derivative of the RC charging equation (above) the falling time (initial rate of fall) has to be the same except for the sign. All that happens is that V_I and V_F get swapped.

⇒ agreement with experiment is good ✓

Exp. 1-X Measure the initial rate of rise of the output of a low pass filter (R=10K, C=0.01μfd) driven by a $10\,V_{pp}$, 0.5KHz square wave input.



Measured Values
$$\begin{cases} R = 9.86\,K\Omega \\ C = 9.65\,nfd \end{cases}$$
on BK 388 meter



$$\text{Slope} = \frac{3\text{div} \times 500\,mV/\text{div}}{3\text{div} \times 5.00\,\mu\text{Sec/div}}$$

→ nearly straight line but small curvature.

EXTRA: What is the initial rate of fall? Should this be the same except for sign?

$$\text{Slope} = \frac{-3\text{div} \times 500\,mV/\text{div}}{3\text{div} \times 5.00\,\mu S/\text{div}}$$

→ must change slope on Trigger to "falling" to get stable curve on scope.

Appendix B

Laboratory Equipment

Each lab station is set up to accomodate two people at one time and should have the following equipment (or its equivalent):

1. Oscilloscope (Tektronix TDS-1002, dual channel, 60 MHz, digital storage, 2 GS/s)
2. Function generator (sine, square and triangle waves, 0 to 2 MHz, B&K 3022)
3. Two variable power supplies (0 to ±15 volts, about 250 mAmp)
4. Solderless breadboard for circuit assembly
5. Assorted 1/4 Watt resistors (10 Ohms to 10 MegOhms)
6. Assorted capacitors (10 pfd to 220 μF)
7. Assorted wire (solid 22 guage), and tools (needle nose pliers, wire cutter, wire striper, and small screwdriver for scope probe adjustments)
8. Computer with data acquisition board (digital I/O, DAC, ADC)
9. Other semiconductor components (transistors, diodes, integrated circuits, etc.) obtained as needed

Appendix C

Manufacturers and Distributors

Electronics components, from discrete components like resistors and capacitors to sophisticated integrated circuits are manufactured by a variety of different companies. However, in most cases. you do not buy the components directly form the manufactures. Instead you purchase the components from a distributor. Distributors range from large scale industrial suppliers who will only sell components in large quantities to other companies to small mail order companies that will sell small quantities of components to individuals.

In most cases you must get the specifications for the components and decide want you want before contacting the distributor. This information is available in the form of data books from the manufacturer (discussed more in chapter 2) or in many cases can be found on web sites set up by the manufacturers (a general listing of different IC'c can be found at www.chipcenter.com). Some manufacturers web sites are listed in table C.1 (this is NOT a recommendation for or against any particular manufactures, but only a list of some sources of information). You may find the specifications for many types of electronics components on these sites.

Most distributors now have their whole catalogs on-line for easy access. Some distributors that will sell small quantities to individuals and their web sites are listed in table C.2 (again this is NOT a recommendation for or against any particular supplies. but only a list of some that are available):

There are also several companies that produce software for computer aided design of electronic

AMD:	www.amd.com
Analog Devices:	www.analog.com
Burr Brown:	www.burr-brown.com
Intel:	www.intel.com
Intersil:	www.intersil.com
Linear Technologies:	www.linear-tech.com
Motorola Semiconductor:	www.mot.com and mot-sps.com
National Semiconductor:	www.national.com
Texas Instruments:	www.ti.com

Table C.1: Some Manufactures

Allied Electronics:	www.allied.avnet.com
Digi-Key:	www.digikey.com
Jameco:	www.jameco.com
JDR:	www.jdr.com
Mouser Electronics:	www.mouser.com
Newark:	www.newark.com
Radio Shack	www.radioshack.com

Table C.2: Some Distributors

Circuit Maker:	www.circuitmaker.com
Electronic Workbench:	www.electronicsworkbench.com
Spectrum Software:	www.spectrum-soft.com/index.shtm
OrCad	www.orcadpcb.com
Spice	bwrc.eecs.berkeley.edu/Classes/IcBook/SPICE/

Table C.3: Some electronics CAD companies.

circuits. Some of these companies are listed in table C.3. These programs can be useful for analyzing complicated electronic circuits. Some of these companies offer free downloads of limited-use evaluation or student versions. Again this list is not a recommendation for or against any particular company but just a list of some things that are available.

Bibliography

[1] J. J. Brophy, *Basic Electronics for Scientists*

[2] A. J. Diefenderfer, *Principles of Electronic Instrumentation*, 3rd edit. Saunders 1994, ISBN 0-03-074709-0.

[3] L. R. Fortney, *Princ. of Electronics, Analog and Digital*, (Harcourt, 1987)

[4] Allan R. Hambley, *Electrical Engineering Principles, Prin. and Applic., 3rd edit.*, Prentice Hall, 2004.

[5] T. C. Hayes and P. Horowitz, *Student Manual for the Art of Electronics*, Cambridge Univ. Press, 1989, ISBN 0-521-37709-9.

[6] P. Horowitz and W. Hill, *The Art of Electronics*. 2nd edit.. Cambridge Univ Press, 1989, ISBN 0-521-37095-7.

[7] J. Millman and A. Grabel, *Microelectronics, 2nd edit.*, McGraw-Hill, 1987, ISBN 0-07-042330-X.

[8] C. J. Savant, M. S. Roden, G. L. Gordon, *Electronic Design*, 2nd Edit. Benjamin/Cummings, Calif., 1991, ISBN 0-8053-0285-9.

[9] Paul Scherz, *Practical Electronics for Inventors*, McGraw Hill, 2000.

[10] A. S. Sedra and K. C. Smith, *Microelectronic Circuits*. third edition. Saunders College Publishing. 1990, ISBN 0-03-051648-X.

[11] R. E. Simpson, *Intro. Electronics for Scientists and Engineers* 2nd edit., Allyn and Bacon, 1987.

Some books on analog electronics.

[12] P. R. Gray, P. J. Hurst, S. H. Lewis and R. G. Meyer. *Analysis and Design of Analog Integrated Circuits*, 4th edit, Wiley, 2001.

[13] M. N. Horenstein. *Microelectronic Circuits and Devices*. Prentice-Hall 1990.

[14] Edwin S. Oxner. *Power FETs and Their Applications*. Prentice-Hall. 1982.

[15] M. E. Van Valkenburg, *Analog Filter Design* (Holt Rinehart and Winston. 1982).

[16] Darold Wobschall, *Circuit Design for Electronic Instrumentation*, 2nd edit., (McGraw-Hill,1987).

Some books on digital electronics.

[17] R. L. Tokheim, *Digital Principles*, third edition, Shaums Outline Series, McGraw Hill 1994, ISBN 0-07-065050-0.

[18] R. Littauer, *Pulse Electronics*

[19] J. F. Wakerly, *Digital Design, Principles and Practices*, 2nd edit., (Prentice Hall 1990) ISBN 0-13-211459-3

Some books on computer interfacing.

[20] D. Abbott, *PCI Bus Demystified*, LLH Technology Publishing, 2000, ISBN 1-878707-54-X

[21] Jan Axelson, *Parallel Port Complete*, Lakeview Research, 2000, ISBN 0-9650819-1-5

[22] Lewis. C. Eggbrecht, *Interfacing to the IBM Personal Computer* 2nd edit. (SAMS/Macmillan, 1992), ISBN 0-672-22722-3.

[23] Jan Axelson, *USB Complete, 3rd edit.*, Lakeview Research, 2005, ISBN13 978-1-931448-02-4

Some books on semiconductor device physics and manufacturing.

[24] Stephen A. Campbell, *The Science and Engineering of Microelectronic Fabrication*, (Oxford Univ. Press, 1996).

[25] David J. Elliot, *Integrated Circuit Fabrication Technology* (McGraw-Hill, 1982).

[26] D. A. Fraser, *The Physics of Semiconductor Devices*, 3rd edit., (Oxford Univ. Press, 1983).

[27] D. A. Hodges and H. G. Jackson, *Analysis and Design of Digital Integrated Circuits*, (McGraw Hill, 1983).

[28] S. M. Sze, *Semiconductor Devices, Physics and Technology*, (Wiley 1985).

Some books on grounding, and noise reduction and instrumentation.

[29] Henry W. Ott, *Noise Reduction Techniques in Electronic Systems*, (Wiley, 1976).

[30] Ralph Morrison, *Instrumentation Fundamentals and Applications*, (Wiley, 1984).

[31] John H. Moore, Christopher C. Davis and Michael A. Coplan, *Building Scientific Apparatus*, (1st edit. Addison-Wesley, 1983; 3rd edit. Westview Press, 2002)

[32] Jacob Fraden, *Handbook of Modern Sensors*, (3rd edit. AIP Press, 2004)

Index