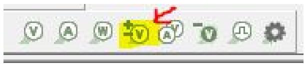
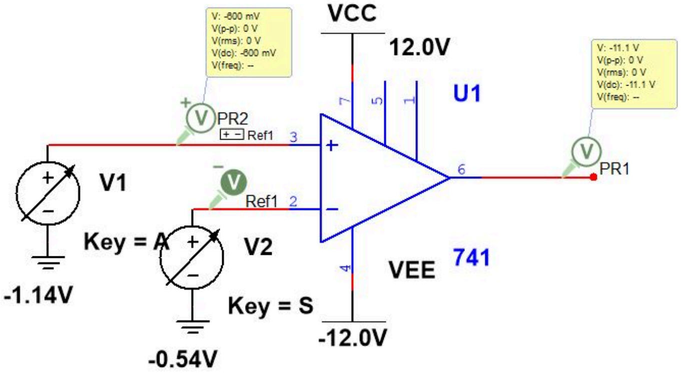
When you were introduced to op amps, you briefly investigated what happens with a high gain amplifier if there is a measurable differential input voltage. From that, you developed a model for op amps ***when they are acting as amplifiers***: the Virtual Short. You've been using that model for so long you may have come to believe that there can never be a difference in voltage between the inputs.

Using Multisim, create the following circuit.

- V1 and V2 are DC_INTERACTIVE_VOLTAGE, available under Sources->SIGNAL_VOLTAGE_SOURCES
- Set both V1 and V2 to Maximum Value = 5 V, Minimum Value = -5V, Increment =0.1%
- Monitor the output using a standard voltage probe, as shown
- Monitor the differential input voltage using a differential voltage probe , with the probe on the non-inverting input and its reference on the inverting input.
- Use +/- 12 VDC to power the 741 op amp.



Set V1 and V2 to the following values, and record the results seen on the voltage probes.

V1 (non-inv), V	V2 (invert), V	Vdiff, V	Vout, V
1.00	-1.00	2	11
-0.05	0.10	-0.15	-11
3.50	2.50	1	11
2.25	2.29	-0.04	-11
-0.01	-0.07	0.06	11

So, when there is a measurable difference in voltage between the two pins of an op amp, there are only two possible outputs: maximum positive voltage or maximum negative voltage. This is affectionately known as "nailed to the rails". Why does this happen? Let's look at the first example above. In this case, the differential voltage between the pins is +2 V. If the amplifier has an open circuit gain of 200,000, what would the theoretical output voltage be? Record the numeric part as a number between 1 and 999

400

and select the correct unit:

○ mV

○ V

● kV

Our amplifier is only powered from +/- 12 VDC, so there is no possible way it can achieve the voltage predicted by the gain formula. In this case, using terminology we learned with transistors, it is saturated, and its output is dictated by components external to the op amp itself, in this case the power supplies. As with transistors, op amps that are not operating in the active linear region are said to be in switch mode. Actually, since this is not a "rail to rail" op amp, it can only come to within about 1 V of the rails.

The good news is that, if there are only two possible outputs, the device is binary. That means it can be used as an interface between the real world and binary digital devices.

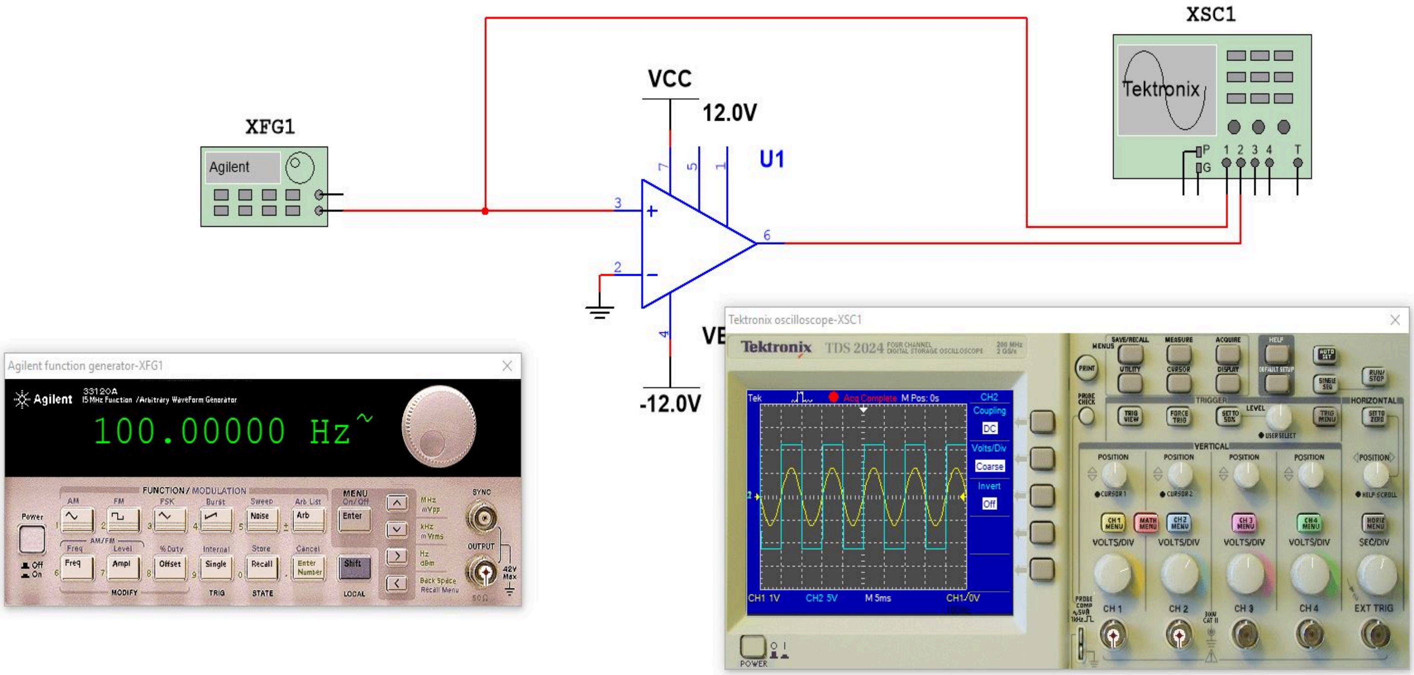
Comparators

The two possible outcomes, in fact, tell us which of the inputs is at a more positive voltage. If the non-inverting input is more positive, the output will be positive; if the inverting input is more positive, the output will be negative. So, in switch mode, the op amp "compares" the two signals and reports which one is more positive; therefore, when in this configuration, we don't call it an ampifier, we call it a ***Comparator***.

Zero Crossing Detectors

If one of the inputs is grounded, the output of the comparator will change states when the other input changes from positive to negative or *vice versa*. Therefore, the comparator detects when the input signal crosses the zero voltage line, and we call it a Zero Crossing Detector.

Modify your circuit in Multisim as shown. Please note that the Agilent Function Generator's true Output is the lower of the two "pins" shown -- the other on is called "Sync", and always generates a large unipolar square pulse, regardless of the waveform settings.



100.00000 Hz

CH1 1V CH2 5V M 5ms CH1 ⁄ 0V

Set up the function generator to produce a 100 Hz sine wave with an amplitude of 2.5 V_{p-p}. Adjust your oscilloscope to display this signal adequately, as shown above.

Notice that, even though the input is a sine wave, the output is square -- there are only two possible output voltages: max+ and max-.

Also notice that, with the input signal connected to the non-inverting terminal, the output is positive when the input is positive, and negative when the input is negative. This makes it a ***Non-Inverting Zero Crossing Detector***.

Now, decrease the amplitude of the input signal slowly, all the way down to 50 mV_{p-p}. The amplitude of the input signal

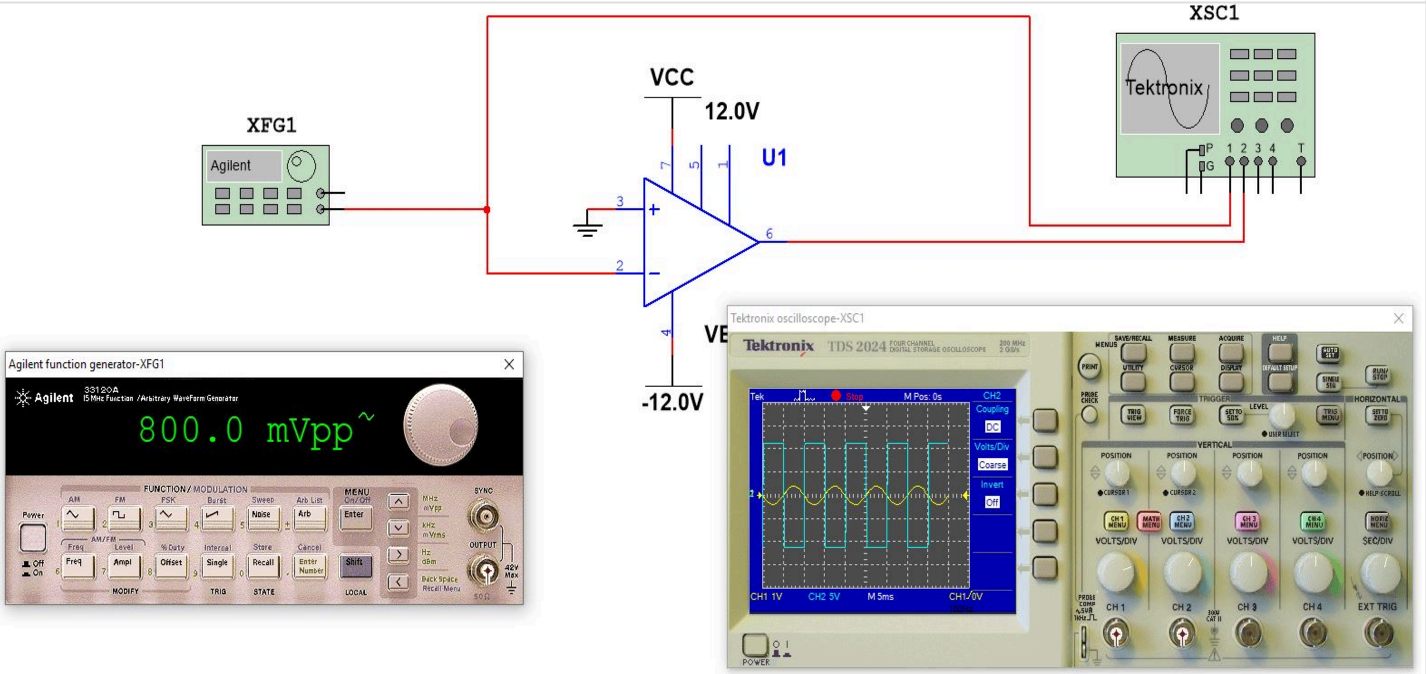
○ Directly affects the amplitude of the output signal.

● Does not affect the amplitude of the output signal.

○ Directly controls the shape of the output signal.

To prove that the last statement is not true, try switching to a square wave input, then a triangle wave input.

Now, ground the non-inverting input and connect the function generator to the inverting input, as shown. You can increase the amplitude back to where the input signal is clearly visible again.


XSC1
Tektronix
P
G
1 2 3 4
T
VCC
12.0V
U1
XFG1
Agilent
3
2
7
5
1
6
4
VEE
-12.0V
Agilent function generator-XFG1
Agilent 33120A 15 MHz Function / Arbitrary Waveform Generator
800.0 mVpp~
Tektronix oscilloscope-XSC1
Tektronix TDS 2024 FOUR CHANNEL DIGITAL STORAGE OSCILLOSCOPE
Tek
Stop
M Pos: 0s
CH2
Coupling
DC
Volts/Div
Coarse
Invert
Off
CH1 1V
CH2 5V
M 5ms
CH1 / 0V

Notice now that, although the output looks the same (rail to rail), it is positive when the input is negative and negative when the input is positive. This makes it an ***Inverting Zero-Crossing Detector***.

Dedicated comparators

While we have the previous circuit operational, increase the frequency slowly from the 100 Hz you started with up to 10 kHz. You should notice that the output is no longer a square wave. This is caused by

● Slew Rate

○ Small-Signal Bandwidth

○ Low-Pass Filtering

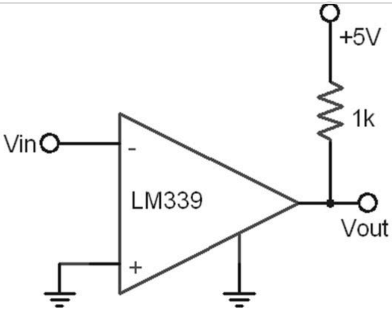
○ High-Pass Filtering

To overcome this problem, IC manufacturers have developed "dedicated comparators" -- op-amp-based ICs that can only be used as comparators. Most of these have transistor switches in their outputs to make sure the output level shifts are as rapid as possible. ***Dedicated Comparators*** come in two basic flavours:

- Open Collector (or Open Drain) Outputs
- Driven Outputs

The ones with driven outputs do not need any external components to make them work. However, the Open Collector or Open Drain output devices require an external pullup resistor in order to work. The Collector (or Drain, if it's a FET) of the internal transistor is simply presented to the user to connect as desired. If the desired logic levels are to be 0 and 5 V, the user would connect a pullup resistor to a $+5V_{DC}$ supply. Any other supply could be used, regardless of what is being used to power the comparator. Typically, comparators reference their outputs to ground, so they automatically generate "Unipolar" signals -- signals with only one non-zero voltage and zero volts as the other option.

The LM339 is one such open collector comparator, pictured below. This is actually just one of four comparators in an LM339 quad IC.

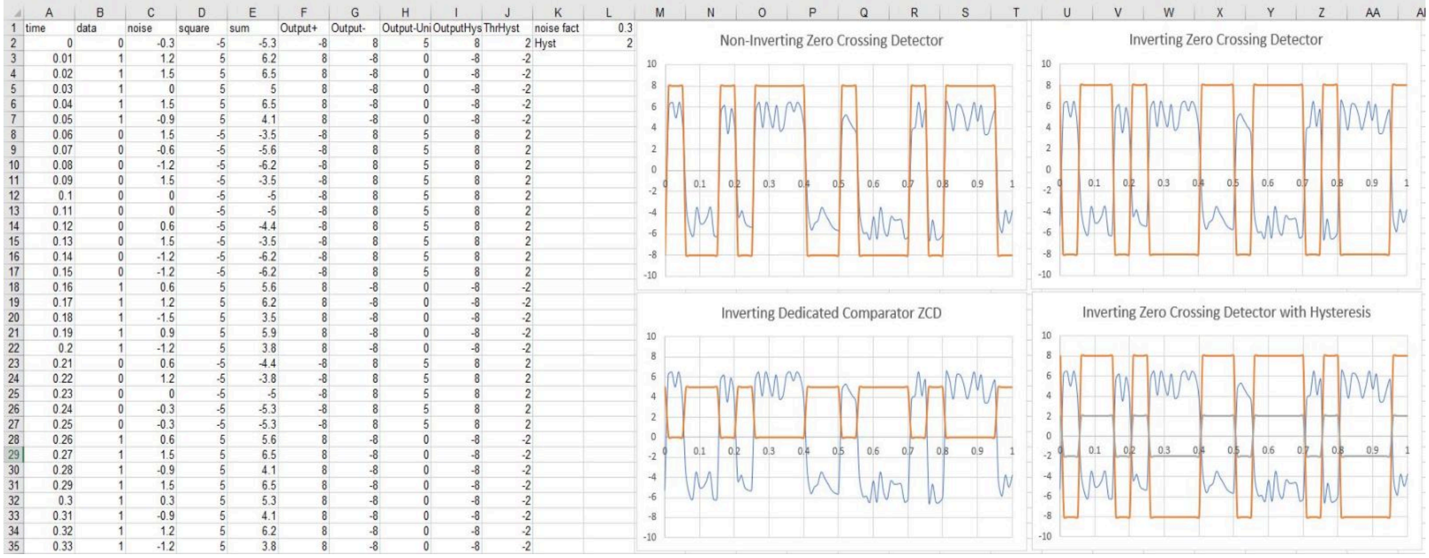


In this configuration, it is set up as an inverting zero crossing detector with TTL Logic outputs.

Since the dedicated comparators are designed to be very fast, and since their outputs are single transistors, the issue of Slew Rate is dealt with.

Data Regeneration

One application for Zero Crossing Detectors is in regenerating digital signals that have deteriorated during transmission due to amplitude attenuation and the introduction of noise. In Moodle, you'll find an Excel file called "Data Regeneration". Open it up, and you should see something like the following:



	A	B	C	D	E	F	G	H	I	J	K	L
1	time	data	noise	square	sum	Output+	Output-	Output-Uni	OutputHys	ThrHyst	noise fact	0.3
2	0	0	-0.3	-5	-5.3	-8	8	5	8	2	Hyst	2
3	0.01	1	1.2	5	6.2	8	-8	0	-8	-2		
4	0.02	1	1.5	5	6.5	8	-8	0	-8	-2		
5	0.03	1	0	5	5	8	-8	0	-8	-2		
6	0.04	1	1.5	5	6.5	8	-8	0	-8	-2		
7	0.05	1	-0.9	5	4.1	8	-8	0	-8	-2		
8	0.06	0	1.5	-5	-3.5	-8	8	5	8	2		
9	0.07	0	-0.6	-5	-5.6	-8	8	5	8	2		
10	0.08	0	-1.2	-5	-6.2	-8	8	5	8	2		
11	0.09	0	1.5	-5	-3.5	-8	8	5	8	2		
12	0.1	0	0	-5	-5	-8	8	5	8	2		
13	0.11	0	0	-5	-5	-8	8	5	8	2		
14	0.12	0	0.6	-5	-4.4	-8	8	5	8	2		
15	0.13	0	1.5	-5	-3.5	-8	8	5	8	2		
16	0.14	0	-1.2	-5	-6.2	-8	8	5	8	2		
17	0.15	0	-1.2	-5	-6.2	-8	8	5	8	2		
18	0.16	1	0.6	5	5.6	8	-8	0	-8	-2		
19	0.17	1	1.2	5	6.2	8	-8	0	-8	-2		
20	0.18	1	-1.5	5	3.5	8	-8	0	-8	-2		
21	0.19	1	0.9	5	5.9	8	-8	0	-8	-2		
22	0.2	1	-1.2	5	3.8	8	-8	0	-8	-2		
23	0.21	0	0.6	-5	-4.4	-8	8	5	8	2		
24	0.22	0	1.2	-5	-3.8	-8	8	5	8	2		
25	0.23	0	0	-5	-5	-8	8	5	8	2		
26	0.24	0	-0.3	-5	-5.3	-8	8	5	8	2		
27	0.25	0	-0.3	-5	-5.3	-8	8	5	8	2		
28	0.26	1	0.6	5	5.6	8	-8	0	-8	-2		
29	0.27	1	1.5	5	6.5	8	-8	0	-8	-2		
30	0.28	1	-0.9	5	4.1	8	-8	0	-8	-2		
31	0.29	1	1.5	5	6.5	8	-8	0	-8	-2		
32	0.3	1	0.3	5	5.3	8	-8	0	-8	-2		
33	0.31	1	-0.9	5	4.1	8	-8	0	-8	-2		
34	0.32	1	1.2	5	6.2	8	-8	0	-8	-2		
35	0.33	1	-1.2	5	3.8	8	-8	0	-8	-2		

M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
AA
Non-Inverting Zero Crossing Detector
Inverting Zero Crossing Detector
Inverting Dedicated Comparator ZCD
Inverting Zero Crossing Detector with Hysteresis
10
8
6
4
2
0
-2
-4
-6
-8
-10
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

This file is dynamic. If you press F9 (or do almost anything in the spreadsheet), the data and output graphs will change. The "noise" on the signal is controlled by a random generator that can be refreshed at any time.

The thin blue line in each of the graphs represents a degraded digital signal, and the heavier sepia (I think that's its colour) line represents the output of the particular comparator circuit described in each chart title.

Notice how, with the initial settings, the fairly noisy input signal is always "regenerated" to the same correct data, in all of the charts. Since the new signal could now be transmitted again without the noise, Zero-Crossing Detectors as applied to data are called ***Data Regenerators***.

The top two charts are very similar, except for the polarity. Satisfy yourself that one of them in non-inverting and the other is inverting.

The bottom left chart shows the output for a Dedicated Comparator, like the LM339, configured as an inverting zero-crossing detector, with a pullup resistor to +5V$_{DC}$. This is the circuit shown previously. Notice that this circuit generates logic levels like the ones you've become accustomed to in you Digital Logic course.

The "noise fact" number in the spreadsheet represents the ratio of the noise to the signal size. Slowly increase this from 0.3 to 1.4. When is the noise big enough to start to introduce transitions that aren't in the original data, for the three charts previously discussed?

○ 0.5

○ 0.8

● 1.0

○ 1.2

The reason these two Zero Crossing Detectors start to fail is that the noise is now big enough to cross the zero line even when the data is not supposed to.

But take a look at the bottom right chart. It doesn't show the false transitions the other ones are showing. With the "Hyst" setting at 2, further increase the noise factor until this one starts failing. This happens at

○ 0.8

○ 1.0

○ 1.2

● 1.4

Notice how badly the other ones are failing, but that the bottom right one is just catching the occasional noise spike. Set the "Hyst" to 1.35 to return to a clean output on this one. The bottom right chart represents the output from a comparator with positive feedback, also known as "Hysteresis", which is a topic in the next Online Lesson.

So, from this lesson, you've learned:

- Op amps with measurable voltages between the input pins will be in saturation, with only two output options: max+ and max-.
- Comparators indicate which input is more positive by the polarity of the output voltage.
- Zero Crossing Detectors compare an input signal to ground, and therefore report whether the input is positive or negative.
- Depending on which input is grounded, a zero crossing detector can be inverting or non-inverting.
- The slew rate of a standard op amp limits its usefulness; Dedicated comparators overcome this issue, and may provide additional external control if they are open collector or open drain. Typically, these devices produce Unipolar output signals.
- Zero Crossing Detectors can be used as Data Regenerators to clean up data that has been degraded due to attenuation and noise during transmission over a distance.
- If the noise is too great, even a Zero Crossing Detector will allow errors to be re-transmitted.

In a previous lesson, you learned how op amps are used to compare two voltages and report which of them is more positive. That seems like a pretty simple concept. As with any simple concept, humans will think of all sorts of complicated things to do with it. We've already seen how comparators can be used to regenerate digital information that has been degraded during transmission over a distance. However, there are times when the noise is too great to be rejected by a simple comparator.

Hysteresis

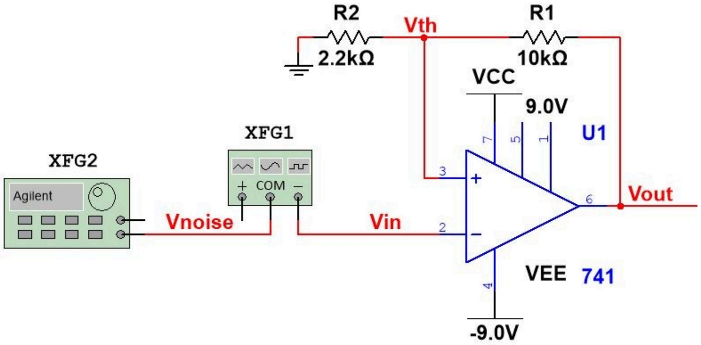
Hysteresis adds improved noise rejection to a comparator circuit. In a simple comparator, the voltage that's being compared to, typically called the "Threshold", is constant. For a zero crossing detector, that threshold is always zero.

Hysteresis introduces a changing threshold. The threshold moves "farther away" from the noise, and shifts depending on whether the output is positive or negative.

From Moodle, open up the Excel spreadsheet called "Data Regeneration". Notice that, in the bottom right chart, there's an extra grey line. That's the Threshold voltage. Notice also that the comparator is in Inverting mode -- that's a requirement for hysteresis.

Increase the noise factor to 1.3, and "refresh (F9)" the screen a few times. Notice that the other three windows are failing consistently, as the noise is crossing the zero line frequently. However, the Hysteresis screen doesn't show any failures. Although the noise is crossing the zero line, it isn't crossing the grey threshold line. When the signal is negative, the threshold line is positive, or "farther away" so the noise doesn't cross it; then the signal is positive, the threshold is negative, so again the noise doesn't cross it. In essence, the improvement in noise rejection is equal to the difference between zero and the threshold, in volts peak, or equal to the difference between the upper and lower thresholds, in volts peak to peak.

But how do we add hysteresis? The clue is in the phrase "move farther away". Unlike with amplification, where negative feedback made the variations smaller by reducing the gain, here we need ***positive feedback*** to increase the variations.



Notice the feedback network. It looks very like what we've used for linear amplifiers, except that it is connected to the non-inverting input. That means it is providing positive feedback instead of negative feedback, and will therefore be a comparator, not an amplifier.

Using Multisim, build the circuit shown above. The odd series arrangement of the two function generators allows us to produce the equivalent of a data pulse with variable noise.

Set up the generic function generator, XFG1 in the schematic, to produce a Square Wave with a frequency of 100 Hz, a duty cycle of 50%, an amplitude of 2.5 V_p (i.e. 5.0 V_{p-p}), and an offset of 0.

Set up the Agilent function generator, XFG2 in the schematic, to produce a Sine Wave with a frequency of 980 Hz, an amplitude of 1 V_{p-p} to begin with, and an offset of 0.

Use a Tektronix 4-channel oscilloscope to display the following:

Channel 1 to display V_{in} on the inverting pin of the op amp

Channel 2 to display V_{th} on the non-inverting pin of the op amp

Channel 3 to display V_{out}

Set all three channels to 5 V/div, and adjust the time base (Horizontal) to display about two cycles of the output signal.

Now, increase the noise signal (Agilent) slowly. At about 5 V_{p-p}, you should see that the noise touches the zero line from both top and bottom. However, the output signal remains unchanged. If this were a simple zero crossing detector, the output would, at this point, start to produce unwanted transitions.

Continue to increase the noise signal slowly. At some point, you should see the output signal start to produce unwanted transitions.

Record the noise voltage, in peak-to-peak volts. 9 V_{p-p}

Subtract the 5 V_{p-p} that you observed when the noise reached the zero line: 4 V_{p-p} This is the improvement in noise rejection resulting from the hysteresis, or positive feedback.

Now, use the measurement features of the Tektronix oscilloscope to measure the difference between the upper threshold and lower threshold voltages (i.e. peak-to-peak voltage of Channel 2). 4 $V_{p\text{-}p}$

The improvement in noise rejection correlates most closely to

○ The noise voltage

● The difference between the thresholds

○ The digital signal voltage

Look at the traces on your oscilloscope, with the noise set so the output is just beginning to produce unwanted transitions. You should be able to see that the noise is just touching the trace for the threshold voltage as it appears at the non-inverting terminal of the op amp.

Mathematically, the threshold voltage, and hence, the improvement in noise rejection, is easy to determine. Use the oscilloscope to determine the maximum and minimum voltages on the output signal.

$V_{out(max)}$ 8.1 V

$V_{out(min)}$ -8.1 V

Notice that V_{th} is in the middle of a voltage divider between V_{out} and ground. For the two possible output voltages, determine the associated threshold voltages:

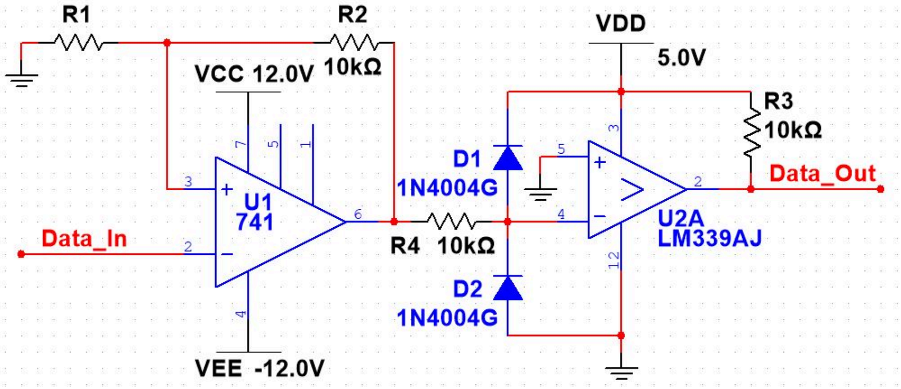
$V_{th(+)}$ = 2 V

$V_{th(-)}$ = -2 V

Find the difference between the upper and lower threshold voltages. This should be very close to the improvement in noise rejection seen earlier, so we'll make the jump and call this the noise improvement in peak to peak volts: 4 $V_{p\text{-}p}$

Worked Example

The following circuit is expected to increase the noise immunity in the output signal by 3.0 $V_{p\text{-}p}$ over a simple zero crossing detector.



First, let's figure out what this circuit does.

- The noisy data at Data_In arrives at the inverting pin of U1, and there's a positive feedback network on the non-inverting pin, referenced to ground. That makes the U1 circuit an inverting zero crossing detector with hysteresis.
- In order for this circuit to work, the noisy Data_In must be "bipolar" (i.e. positive to negative) in order to cross the threshold voltage and trigger the comparator.
- The output voltages from U1 will be approximately +/- 11 V, as the 741 isn't a rail-to-rail op amp.

- U2A is one of four comparators in an LM339 IC. The signal from U1 eventually arrives at the inverting pin of U2A, so, with the non-inverting pin grounded, U2A is an inverting zero crossing detector. With two inversions in the path, the polarity of the cleaned-up Data_Out is the same as the polarity of the noisy Data_In.
- U2A is an open-collector device, so R3 is installed to pull the signal up to +5 V when the internal transistor in U2A isn't pulling Data_Out to ground. So, the output logic is 0 V to +5 V, or standard TTL signalling.
- Notice that U2A is powered from +5 V to ground. This IC is picky about the input signal not exceeding the power rails by very much. That's what R4, D1, and D2 are about:
  - When the output from U1 is +11 V, D1 will be forward biased because current will flow from +11 V to +5 V through D1. That means D1 provides a 0.7 V drop to the +5 V supply, and the input signal to U2A will not exceed +5.7 V.
  - When the output from U1 is -11 V, D2 will be forward biased because current will flow from ground (0 V) to -11 V through D2. That means D2 provides a 0.7 V drop from ground, and the input signal to U2A will not go below -0.7 V.
  - In both directions, R4 provides a resistance across which the remaining voltage from the U1 output can be dropped.
  - This is referred to as either as a clipping circuit because it clips the input signals off at +5.7 V and -0.7 V or as a clamping circuit, because it clamps the input circuit to within 0.7 V of U2A's power rails.

Now, back to the problem at hand: How do we make this circuit improve the noise immunity by 3.0 $V_{p\text{-}p}$?

To do this, the Threshold voltage at the non-inverting pin would have to switch between +1.5 V and -1.5 V. Since the output of U1 switches between +11 V and -11 V, we can rework the voltage divider formula to provide a suitable value for R1:

$$V_{TH} = V_{out}\left(\frac{R_1}{R_1+R_2}\right)$$

so

$$R_1 = R_2\left(\frac{V_{TH}}{V_{out}-V_{TH}}\right) = 10\ k\Omega\left(\frac{1.5\ V}{11\ V\ -\ 1.5\ V}\right) = 1.58\ k\Omega$$

Since we don't have a 1.58 kΩ resistor in our kits, we'll pick the next biggest, because that will increase the noise rejection slightly. So, let's see what the theoretical results will be using a 1.8 kΩ resistor:

$$V_{TH} = V_{out}\left(\frac{R_1}{R_1+R_2}\right) = 11\left(\frac{1.8\ k\Omega}{1.8\ k\Omega+10\ k\Omega}\right) = 1.68\ V$$

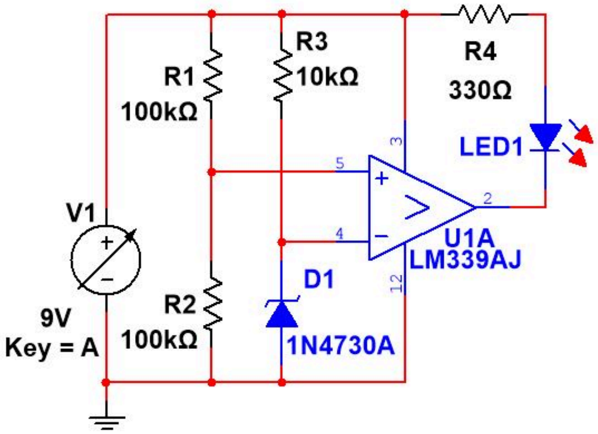
Therefore, the improvement in noise rejection is 1.68 V_p or 3.36 $V_{p\text{-}p}$.

Summary

In this lesson, you have learned that

- Noise rejection can be improved using hysteresis
- Hysteresis employs positive feedback to move the threshold voltage "away" from the noise; in other words, the threshold voltage is no longer simply zero, as it would be in a zero crossing detector. It becomes positive when the noisy input is negative, and it becomes negative when the noisy input is positive.
- The threshold voltages are determined from the voltage divider between the output signal and ground.
- The improvement in noise rejection is equal to the difference between the thresholds, in volts peak to peak.

As we saw in the introduction to comparators, we can compare an input signal to any voltage -- not just ground. The following circuit is a "Battery Low" indicator.



Build this circuit in Multisim. V1 is a DC_INTERACTIVE_VOLTAGE source. Make sure you select the "LED_red" from "LED" under "Diodes".

Start with V1 set to 9 V, to simulate a 9 V battery. Also set its "increment" to 1% for finer adjustment.

Run the simulator, and slowly decrease the "Battery" voltage until the LED lights. Record the voltage here. 7.8 V_{DC}

At this voltage, a 9 V battery would be considered "Dead".

Now to analyze this circuit to see why it works.

First, look up the 1N4730A Zener diode to see what its reverse breakdown voltage is. 3.9 V

Since the Zener diode is connected between the inverting terminal and ground, when the circuit is working, the voltage at the inverting terminal will be 3.9 V

When the battery is "good", i.e. 9.0 V_{DC}, what voltage will appear at the non-inverting terminal? 4.5 V_{DC}

(That's because of the voltage divider created by R_1 and R_2, since no current can flow into the input.)

When the battery is "good", the output will be HIGH

In this condition, no current can flow through the LED, so it will be OFF.

What would the battery voltage need to be to produce a voltage at the non-inverting input that matches the inverting input voltage?

7.8 V_{DC}

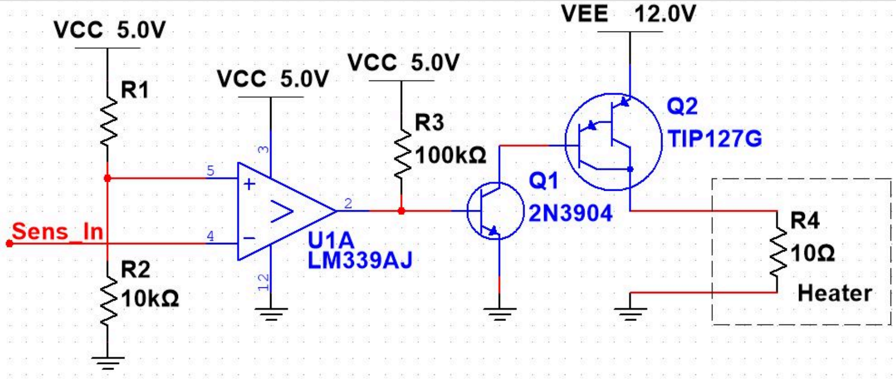
When the battery voltage is too low, the output will be LOW

Recall that the LM339 is an Open Collector device -- its output is the collector of an internal transistor. The output goes LOW when this transistor is turned ON, allowing current to flow through the LED.

Note that in this circuit the Zener diode provides a constant reference voltage regardless of the condition of the battery, as long as the battery voltage is high enough to put the Zener into reverse breakdown.

Worked Example

The following circuit receives an input voltage from a temperature sensor. When the temperature is too low (i.e. when the sensor voltage drops below 1.25 V), the Heater turns on.



First, let's analyze this circuit:

- The Sensor signal connects to the inverting input, so when the temperature is high, the output of U1A will be LOW, with its internal transistor turned on; when the temperature is too low, the output of U1A will be pulled up to +5.0 V. No current will flow into U1A's output, but R3 will act as the Base resistor for Q1.
- Q1 will be turned on, with a Base current of (5.0 V - 0.7 V)/100k = 43 μA. This means that the Collector of Q1 could draw up to βI_B = 100*43 μA = 4.3 mA from the Base of Q2.
- Q2 is a Darlington Pair PNP Power Transistor, which you'll learn more about in a subsequent course. Suffice it to say that it has a β of about 1200. The maximum Collector current that could be drawn through the heater is I=V$_{EE}$/R4 = 12 V/10 Ω = 1.2 A. Q2's Base current therefore would be 1.2 A/1200 = 1 mA, so Q1's Collector current is sufficient to turn Q2 on.
- So, when the Sensor voltage is lower than the Threshold voltage at the non-inverting input of U1A, the output of U1A goes high, allowing Q1 to conduct which draws Base current from Q2 which will turn it on and run about 1.2 A through the heater.
- When the Sensor voltage is higher than the Threshold voltage at the non-inverting input of U1A, the output of U1A goes low, turning off Q1 so no Base current is available to Q2, and Q2 will be off -- no current through the heater.

So, back to the original problem. The heater is supposed to come on if the Sensor voltage drops below 1.25 V. That means the Threshold voltage at the non-inverting input has to be set to 1.25 V. Reworking the voltage divider formula gives us the following:

$$V_{TH} = V_{CC}\left(\frac{R_2}{R_1+R_2}\right)$$

so

$$R_1 = R_2\left(\frac{V_{CC}-V_{TH}}{V_{TH}}\right) = 10\ k\Omega\left(\frac{5\ V-1.25\ V}{1.25\ V}\right) = 30\ k\Omega$$

In this case, we'll assume that the 1.25 V is critical, so we'll come up with a way to make 30 kΩ; and since we're doing that, we might as well make it adjustable! If we pick a standard potentiometer (with only 1, 2, and 5 available as multipliers) and a fixed resistor that's between 2x and 5x its size, it looks like we could use a 10 kΩ potentiometer and a 27 kΩ fixed resistor to provide a range of resistances from 27 kΩ to 37 kΩ.

When we discussed digital to analog conversion (D to A), we mentioned that D to A was the simpler process, and that one of the better known A to D conversion processes actually relies upon the DAC as one of its components. Now that we've been introduced to the comparator, a device that compares two voltages to indicate which is more positive, we have the tools necessary to tackle A to D conversion.

Approaches to A to D Conversion

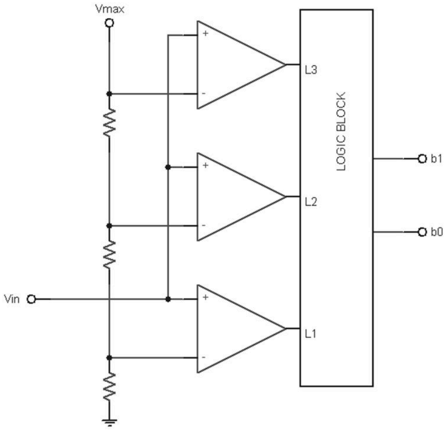
There are a number of different ways in which analog characteristics from the real world are digitized. The three main ways are:

1. Direct Conversion (also called Flash Conversion)
2. Fast Ramp Conversion
3. Successive Approximation Conversion

There are variations of the second one (Fast Ramp Conversion) which attempt to speed it up (contrary to its promising name, it is definitely the slowest of the approaches!) These are called things like Delta Conversion, Sigma Conversion, Delta-Sigma Conversion, etc. For this course, however, we will stick to the three main techniques.

Direct Conversion

Direct Conversion produces a binary result as a single event, or, in digital logic terms, in one clock cycle. As a result, this is a very fast technique, but the down-side to it is that it is very hardware intensive. Here's a simple 2-bit direct converter. (This is a pretty lousy A to D converter, as it only has four levels and three steps -- you're definitely not going to be listening to Tchaikovsky with this one!)



You'll get to build a 3-bit version of this circuit in the Lab, so instead of building and testing this circuit, we'll simply analyze it.

If V_{max} is 5.000 V and all three resistors are the same, we can analyze the thresholds for the comparators using a 3-resistor voltage divider as follows.

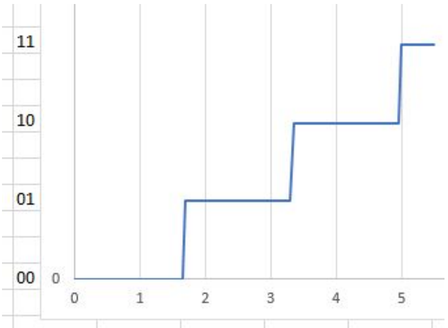
The current through the voltage divider would be

$I = \dfrac{V\text{max}}{3\ R}$, since the total resistance is the series combination of the three resistors.

The voltage for "L1" would then be

$$V_{L1} = R\left(\frac{V\text{max}}{3\ R}\right) \ = \ \frac{5}{3} = 1.667\ V$$

Following the same logic, the threshold for "L2" would be 3.333 V, and the threshold for "L3" would be 5.000 V. Here, then, is the relationship between the input voltage and the output binary code, given that we simply need to count the steps in binary:



Any voltage below 1.667 V is considered as 00. Any voltage between 1.667 V and 3.333 V is considered as 01. Any voltage between 3.333 V and 5.000 is considered as 10. Any voltage above 5.000 V is considered 11. This example really demonstrates that digitizing an analog signal can result in the loss of a lot of the original data! That's what we called "Quantization Error".

The "Logic Block" for this circuit needs to take the three comparator outputs and convert them into binary. Typically, this would be a Priority Encoder -- a circuit that generates a binary combination that represents the highest active input. By the way, all the lower inputs will also be active, because the input voltage is high enough to trigger them all.

This example is deceptively simple. Increasing the number of bits in the output increases the resolution of the circuit, but with an exponential increase in the hardware complexity. Notice that two bits were generated by four different voltage levels, three of which were supplied by comparators and the fourth was zero.

Let's consider what would be needed for more complex Direct Converters.

If we wanted three bits, we would need:

8 levels

7 comparators (one for each step), and a

7 input 3 output priority encoder.

Now, repeat for a sixteen bit Direct Converter.

65536 levels

65535 comparators and a logic circuit with

65535 inputs and 16 outputs. I'd like to see that circuit on a breadboard (or maybe a thousand breadboards!)

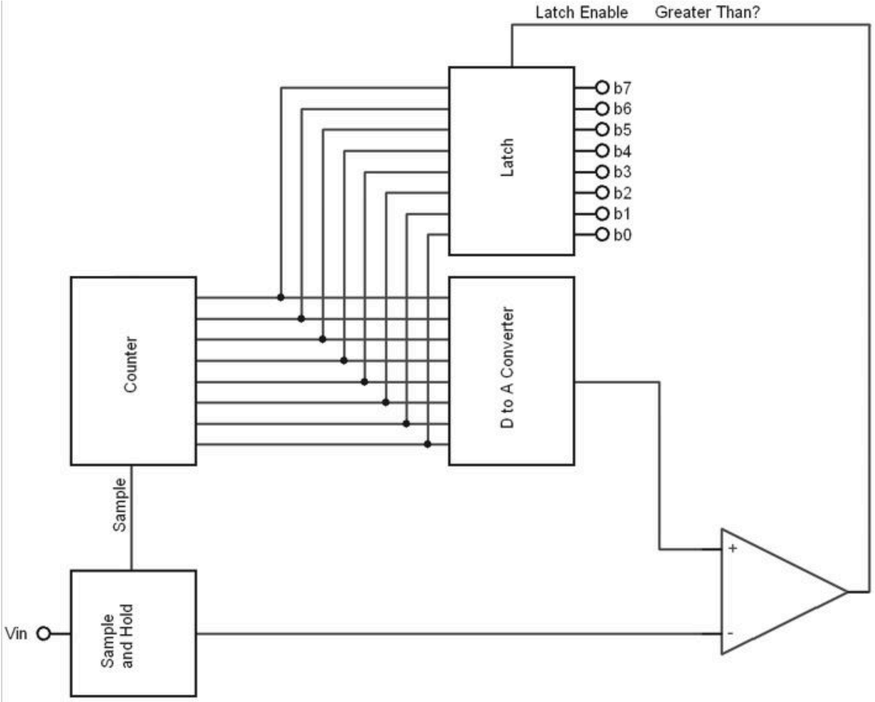
So, clearly, Direct Conversion is fast -- just one clock cycle -- but incredibly hardware intensive; and we haven't even talked about the digital logic involved in creating a Priority Encoder!

Fast Ramp Conversion

Clearly, there must be less hardware-intensive ways of digitizing characteristics from the real world. The Fast Ramp Converter is actually really slow. The name means you have to have a really fast binary ramp counter to make this even feasible.

With the Fast Ramp Converter, a DAC driven by a binary counter generates a constanty ramping output voltage until it reaches the top, then starts over again. The analog input signal is presented to one input of a comparatar, and the ramp is presented to the other input. When the ramp passes the analog signal, the value in the binary counter is stored as the binary representation of the analog input.

This introduces the need for a circuit called "Sample and Hold". At the beginning of each new ramp, a circuit at the input of the A to D converter grabs the present value of the input and keeps that value until the end of the ramp -- essentially an analog memory device. This is usually a capacitor, and the "switch" that temporarily connects the analog signal to it is usually a FET. The signal to get a new sample could be generated using the "Carry Out" of the counter. Here's the resulting block diagram of a Fast Ramp Converter.



In order for the sample rate to be constant (a definite requirement for proper digitization and playback), the Fast Ramp Converter needs to go through the entire counter ramp for each sample.

Just to see how slow this system is, let's compare it to a Direct Converter. Let's say that the clock running each of these devices issues a pulse every nanosecond.

How many samples per second would an 16-bit Direct Converter be able to make, in megasamples per second? 1000 MSa/s

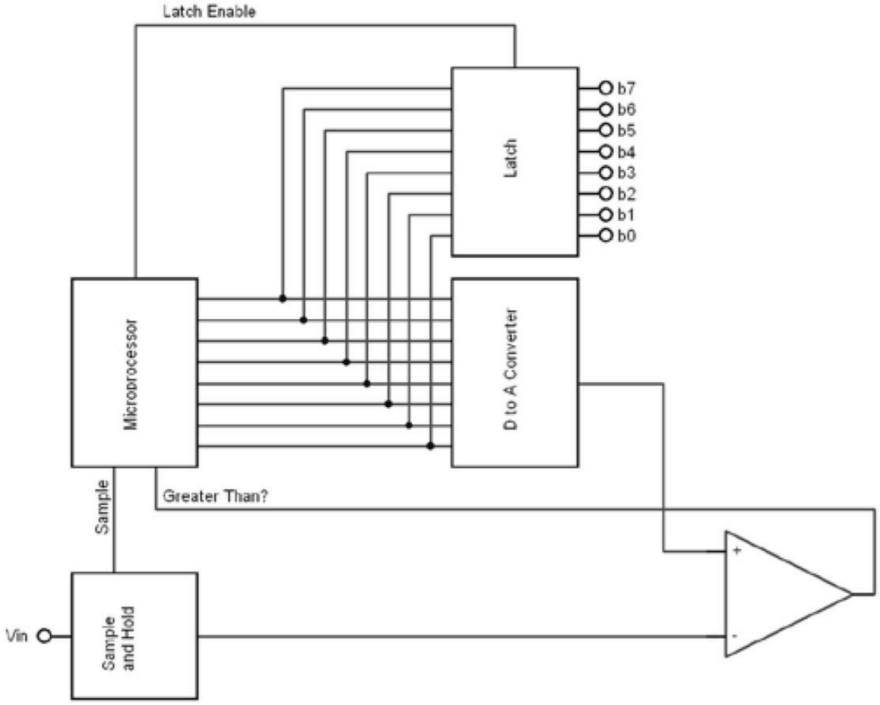
How many samples per second would an 16-bit Fast Ramp Converter be able to make, again in megasamples per second? (Start by determining how many different voltage levels there are in a 16-bit ramp.) 0.0152 MSa/s

That's significantly slower.

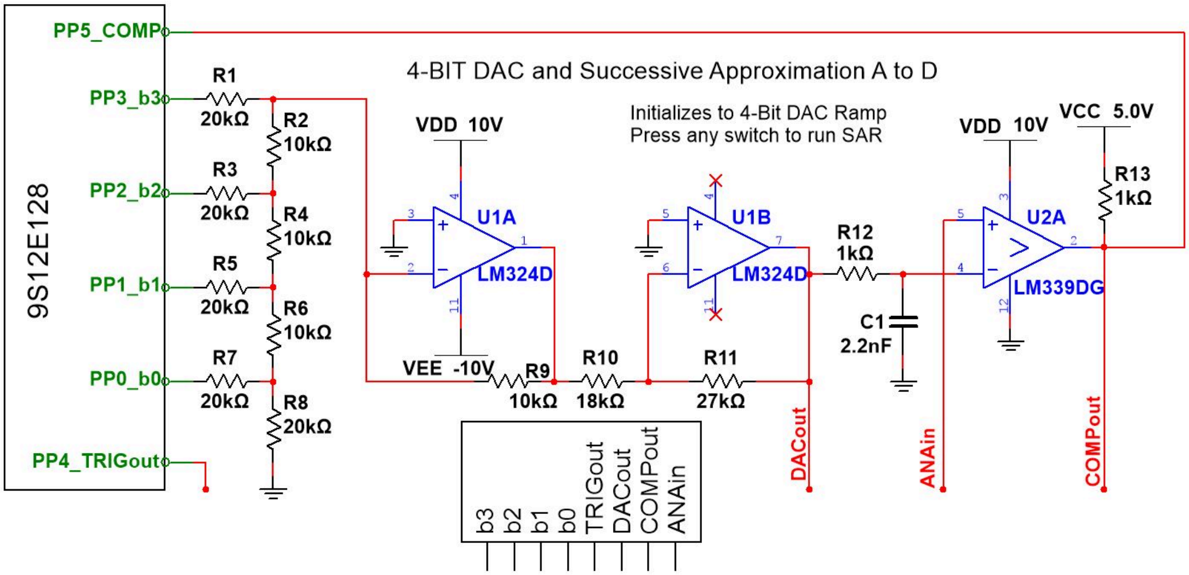
Successive Approximation

The happy medium between the two is a process called Successive Approximation, which is a smart guessing game. You could say that the Fast Ramp approach is a highly untelligent guessing game: Is it 0? No? Is it 1? No? Is it 2?, No? is it 3? Yes? Great; is it 4? No? is it 5? No? ....

The smarter way to go would be to split the field in half and pick the one with the answer in it, then split that half into quarters, etc. That's how a Successive Approximation A to D Converter works. The hardware looks pretty similar to the Fast Ramp, except for the addition of some "brains". By the way, a Successive Approximation A to D Converter is often referred to as an SAR, short for Successive Approximation Register, since the results are presented to the data bus for storage using a latching register.



Here's a 4-bit Successive Approximation DAC built up on one of the microcontroller boards from a few years ago.



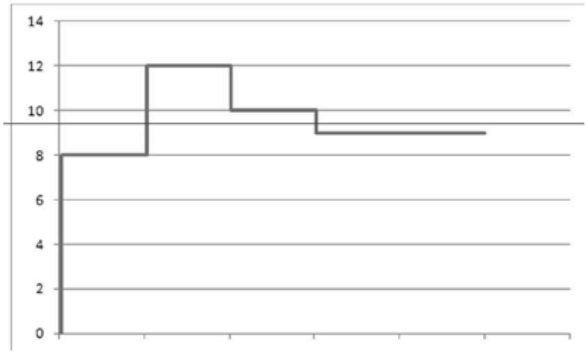
For this simple application, the Sample and Hold Circuit was left out, an omission which introduces significant error at higher frequencies.

The microprocessor on the left can generate any of the binary values from 0000 to 1111, which it sends into the R-2R ladder at the front end of the DAC. The R-2R ladder feeds into an inverting summing amplifier followed by a second inverting amplifier to generate a positive output. Next is a single-pole Low Pass Filter to reduce switching noise. After that comes the comparator. One of its inputs is the Analog signal, the other input is the DAC output. The comparator sends its binary output back to the microprocessor so it can make decisions based on whether the analog signal is greater or less than the DAC output.

For the fun of it, and for review, let's determine the step size. If all the outputs except the MSB are zero, then 5.0 V is presented to the summing amplifier through a 20 kΩ resistor. Since the feedback resistor is 10 kΩ, this means that the output of U1A will be -2.5 V. Multiplying that by the gain of the second amplifier, we calculate an output voltage of 3.75 V. Since this represents binary 1000 (only the MSB set), we divide by eight to get a step size of 468.75 mV/step. Bizarre value, but we'll live with it.

Successive Approximation Process

To simplify the analysis, let's instead start with an A to D converter with a step size of 1.0 V/step. The following picture shows the process of digitizing an input of 9.437 V.



To begin with, the microcontroller sets just the MSB to produce the binary code 1000 and, given a step size of 1 V/step, an output voltage of 8.0 V. Since this output is lower than the analog signal, the microcontroller keeps the MSB.

Now the microcontroller sets the next bit to produce the binary code 1100 and an output voltage of 12.0 V. Since this output is higher than the analog signal, the microcontroller clears this next bit.

Now the microcontroller sets the second last bit to produce the binary code 1010 and an output voltage of 10.0 V. Again, this output is higher than the analog signal, so the microcontroller clears this bit.

Finally, the microcontroller sets the LSB to produce the binary code 1001 and and output voltage of 9.0 V. Since this output is lower than the analog signal, the microcontroller keeps this bit, and presents its final guess as binary 1001. When played back by a DAC, this value will produce an output of 9.0 V instead of the original 9.437 V, again losing accuracy due to Quantization Error.

Let's compare the speed of a 16-bit Successive Approximation A to D Converter to the two previous converters, again assuming a clock pulse of 1 ns duration.

The Direct Conversion A to D could produce 1 billion samples in a second, or 1000 MSa/s

The Fast Ramp A to D took 65,536 clock cycles per sample, resulting in a pathetic 0.015 MSa/s

The Successive Approximation A to D takes 16 clock cycles for 16 bits, resulting in a rate of 62.5 MSa/s.

It's not super-speedy like the Direct Converter, but also it doesn't need a circuit with over 60,000 comparators and over 60,000 logic inputs; and it's definitely much faster than the Fast Ramp,with hardly any more circuit complexity.

Here are a couple of questions to finish up.

Match each of the following descriptions to the most likely 8-bit A to D converter.

2 ⌄ Digital output generated after eight conversion steps

1 ⌄ Digital output generated after one conversion step

3 ⌄ Digital output generated after 256 conversion steps

1. Direct Conversion
2. Successive Approximation
3. Fast Ramp Conversion

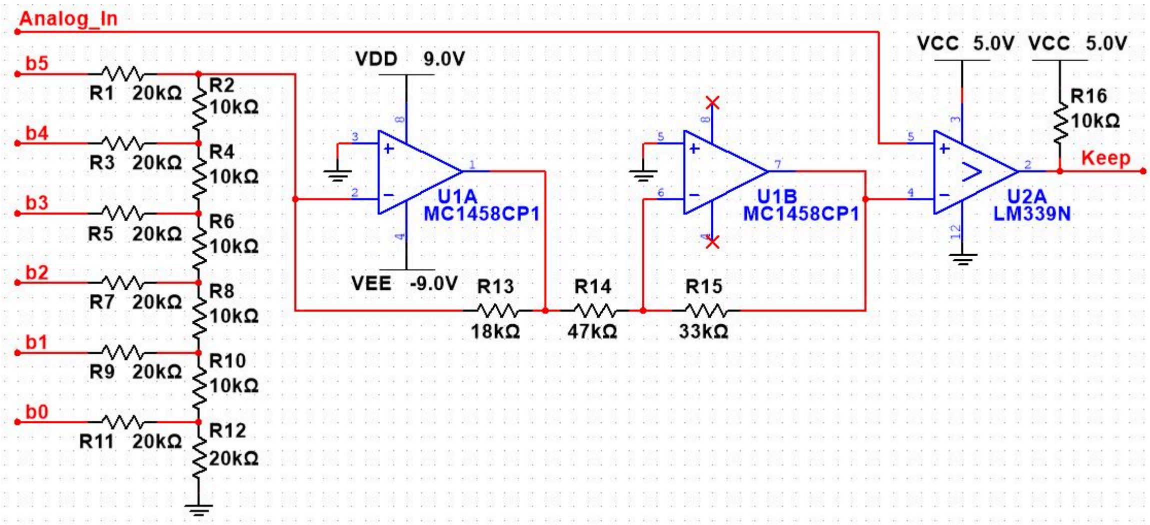
How many conversion steps would be required for a 12-bit A to D converter using each of the technologies listed?

Direct Conversion: 1

Fast Ramp Conversion: 4096

Successive Approximation: 12

Worked Example -- Successive Approximation A to D



We'll start with an overview of the circuit:

- There are six digital inputs, all of which are driven by 3.3 V LVTTL signals from a microcontroller (not shown)
- With six digital inputs, there are 2^6-1 steps, or 63 steps
- The inputs are handled by an R-2R ladder DAC, which should therefore be a linear binary-weighted DAC, with b5 as the MSB
- The two op amps in the DAC are both inverting, so the output voltage will always be zero or positive; hence, there is no need to clamp the input to the comparator
- When the analog signal is greater than the DAC output, the most recent "guess" should be kept; given that the analog input is connected to the non-inverting input, this means that a high output from the comparator indicates "keep"

To determine the step size for an R-2R ladder DAC, we set the MSB and clear all the rest, giving us a binary 100000, or decimal 32. In this state, none of the lower bits contribute any current to the circuit's input, so we can determine the expected output for 32 using the gains of the two amplifiers:

$$V_{32} = 3.3\left(-\frac{18\ k\Omega}{20\ k\Omega}\right)\left(-\frac{33\ k\Omega}{47\ k\Omega}\right) = 2.085\ V$$

The step size, therefore, is

$$Q = \frac{V_{32}}{32} = 65.2\ mV/\ step$$

The maximum output for 63 steps (binary 111111) would be 4.105 V

It's likely that the intent was for this DAC to have a range from 0 V to 4.095 V, and a better design would include a potentiometer in the op amp feedback to allow for some adjustment. For simplicity, let's work with the ideal step size:

Q=65 mV/step

From this, the voltage for binary 100000 would be 2.080 V and the output for binary 111111 would be 4.095 V.

Let's watch the process for the A to D converter digitizing the analog value 1.8702 V:

Binary (Decimal)	Voltage	Comparator
100000 (32)	2.080 (too high)	Don't Keep
010000 (16)	1.040 (too low)	Keep
011000 (24)	1.560 (too low)	Keep
011100 (28)	1.820 (too low)	Keep
011110 (30)	1.950 (too high)	Don't Keep
011101 (29)	1.885 (too high)	Don't Keep

Therefore, the final result is binary 011100, or 28 decimal, which produced the value 1.820 V -- a voltage that's less than one step below the analog input value of 1.8702 V.

In this example, it's fairly clear that 011101 or 29 produces 1.885 V which is actually closer to the analog input than the final result arrived at by this A to D converter. It's for this reason that a more sophisticated A to D converter would add half a step to the input voltage, bumping it up to 1.903 V, in which case the last bit would have resulted in "Keep", and the final answer would have been binary 011101, or decimal 29.