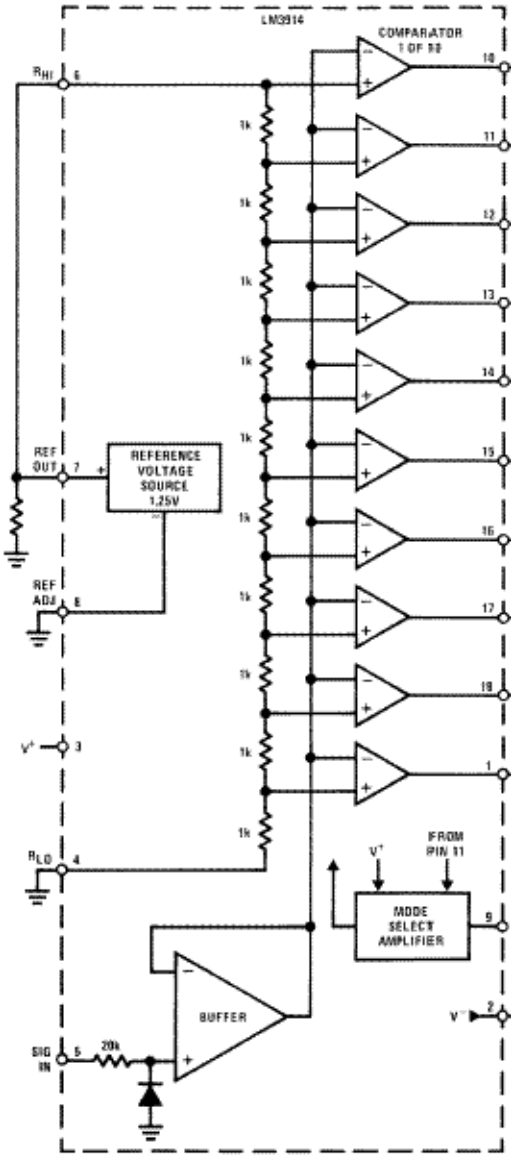
# Conceptual Design of an A to D Convertor

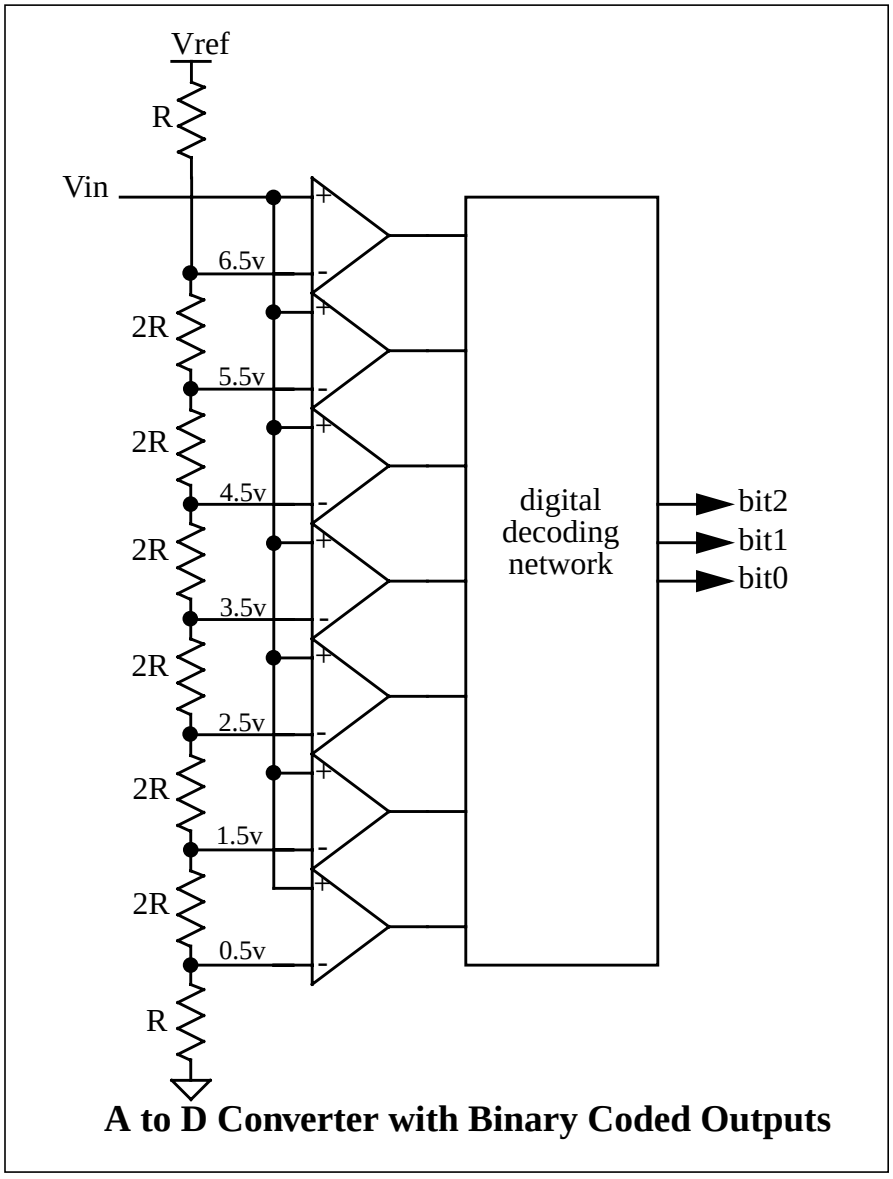
Using the comparator, how can we implement a A to D converter? Imagine if we could produce a set of voltages as reference inputs to multiple comparators that could tell us if we are above or below the different decision points. We could create the reference voltages using one big voltage divider with multiple “taps” where we get the decision voltages.

Below is shown the internal schematic diagram of the National Semiconductor LM3914. It has a resistor “tree” that creates ten reference voltages from a single internal reference voltage. The resistor tree divides the total reference voltage up into ten equally sized voltage steps. Thus each comparator has a reference voltage that differs by one tenth from its neighbors. In this way we can display ten different voltage levels that are applied to the other input to the comparators provided by the buffer amplifier at the bottom. The buffer amplifier provides current gain but no voltage gain. Thus, whatever voltage is presented to the chip pin “SIG IN” appears at each comparator’s non-inverting input.




**LM 3914 Internal Schematic Diagram**

As well as this scheme works, it is very limited as far as how many voltage levels can be represented. If instead we could have each output pin represent a power of two in ascending order, many more voltage levels could be represented without using so many package pins. Below is shown a different implementation of the A to D converter that corresponds to our initial example using binary weighted bits at the output..



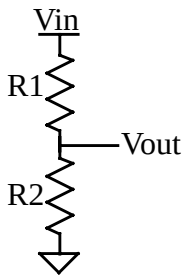

**A to D Converter with Binary Coded Outputs**

As before, the reference input to each comparator is connected to one of the voltage reference taps. The output of each comparator indicates if the input signal exceeds that reference input. The output of each comparator corresponds to a yes or no decision as to if the input signal exceeded the decision point. These digital outputs can be applied to a digital logic network that will *encode* the seven inputs to three outputs. Thus, we can represent eight possible output codes using only three digital bits for output. Likewise we could represent 256 levels of output

with eight output pins.

The resistor network creates the reference voltages required by using only 2 different values of resistor. Their absolute values are not critical as long as R is one-half of 2R. The voltage at the bottom tap follows the voltage divider equation we studied before.

$$V_{out} = V_{in}\frac{R1}{R1 + R2}$$ where;




For the bottom tap on the resistor network we have,

$$V_{out} = 7\frac{R}{14R} = \frac{7}{14} = 0.5V$$

At the next tap,

$$V_{out} = 7\frac{3R}{14R} = \frac{21}{14} = 1.5V$$

To design the binary encoder portion of the A to D converter requires a bit of knowledge of digital circuits. We will pursue that soon.