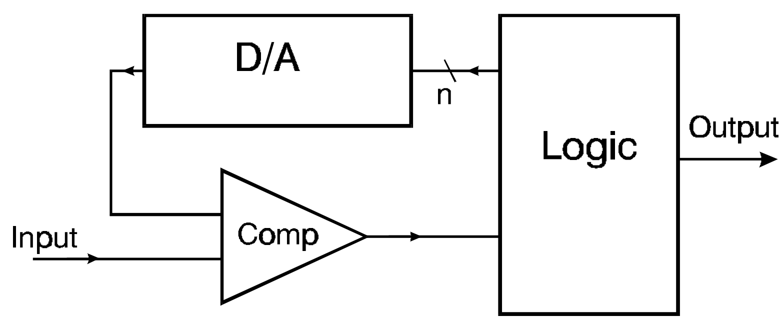
Analogue-to-Digital Conversion

The situation with Analogue-to-Digital Converters (ADCs) is the same as that for DACs, with some more tricks occasioned by the fact that the analogue signal might change at a faster rate than the ADC can handle, i.e. that the speed of the converter may not be infinite compared to the speed of the signal. There is therefore the problem of how to digitise a static signal (compared to the converter speed) and then the problem of converting a dynamic signal.

"DAC + Logic" Converters

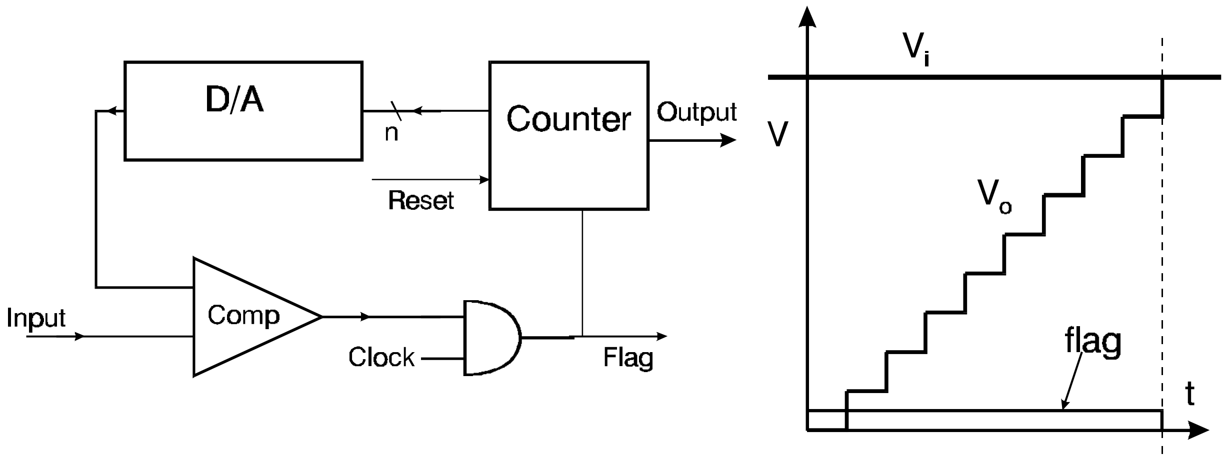
The most common breed of ADC runs by using a DAC and an analogue comparator[22]. The simplest converter is therefore as shown in this diagram where the block marked "LOGIC" can be one of a variety of designs which I shall discuss below. Essentially the strategy of all these converters is to put a value on the DAC, compare the value with the unknown and then update the DAC for another "guess". This is repeated until the optimum value is obtained. The differences between the converters essentially lie in the "guess-strategy" or conversion algorithm.



A Generalised "DAC + Logic" ADC

[22] An analogue comparator simply compares two input voltages and produces a logic "true" if A > B and "false" otherwise.

Counting Strategy

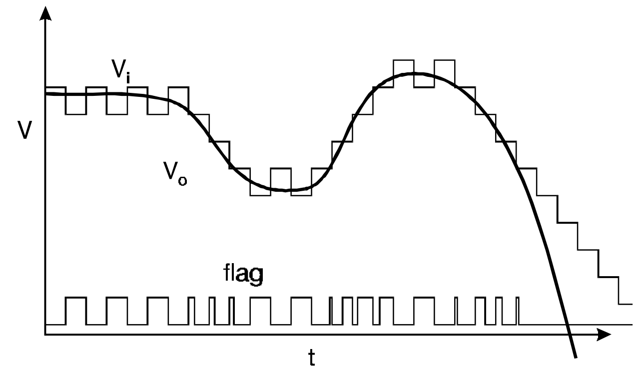


A "Counting" Converter

The simplest conversion algorithm is one which uses a counter which is zeroed and then counts up until the comparator output changes. Notice that the maximum allowed count rate is determined by the sum of the DAC settling time t_{da}, the comparator settling time t_s, and the logic reaction time t_l. The latter is usually negligible at all but the ultimate speeds. The converter is obviously slow, taking a maximum of 2^n cycles to convert to n bits. The conversion time is also variable - which can be a disadvantage. The maximum conversion rate is given by:

$$f = \frac{1}{2^n \left(t_{da} + t_s + t_l\right)}$$

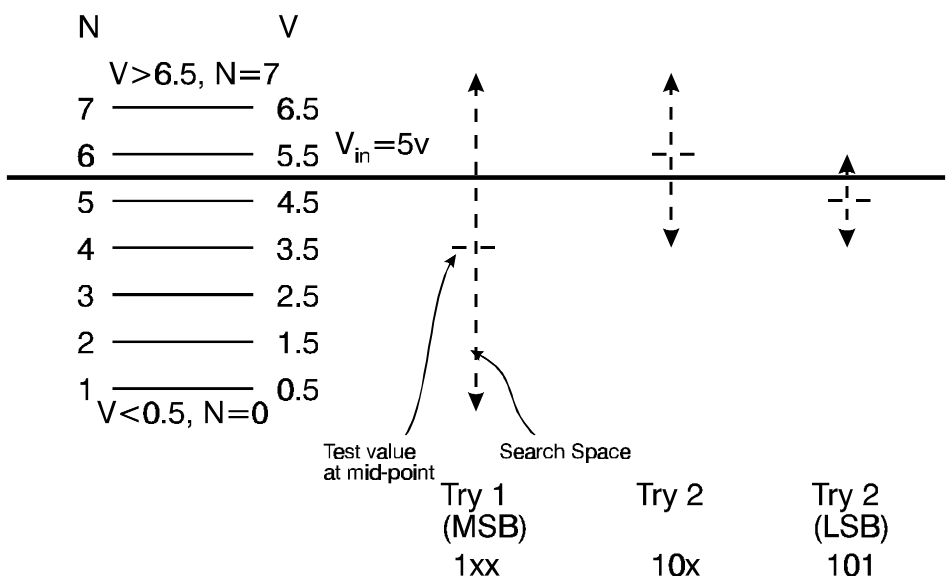
Putting some figures in for t_{da} = 200nS, t_s = 100nS and t_d = 10nS - reasonable figures - gives a maximum time of 256*310nS = 79µS for an 8-bit conversion or a rate of 126KHz. The general problem is that for more bits (ie more resolution) the settling times of the DAC and comparator go up, not down, and therefore for larger numbers of bits the conversion times go up faster than 2^n (which is already going up very fast).



Continuous Conversion Using a Bi-directional Counter

The primary disadvantage of this counter is that it has to count from zero every time a conversion is made. However if we allow the counter to be bidirectional and count "up" for comparator "true" and "down" otherwise, we have a tracking converter which will always be within 1 count of the correct answer assuming that the rate of change of the voltage input does not exceed the rate at which the counter can adjust. If we are accurately monitoring (slow converter rate) an output which changes very slowly, this turns into an advantageous arrangement since the value is always available without intervention, whereas if a full conversion is necessary there will always be some necessary delay before the results are available. However the popularity of the "continuous conversion" converter is not high at the moment because there are in general better solutions to the problem.

Successive Approximation

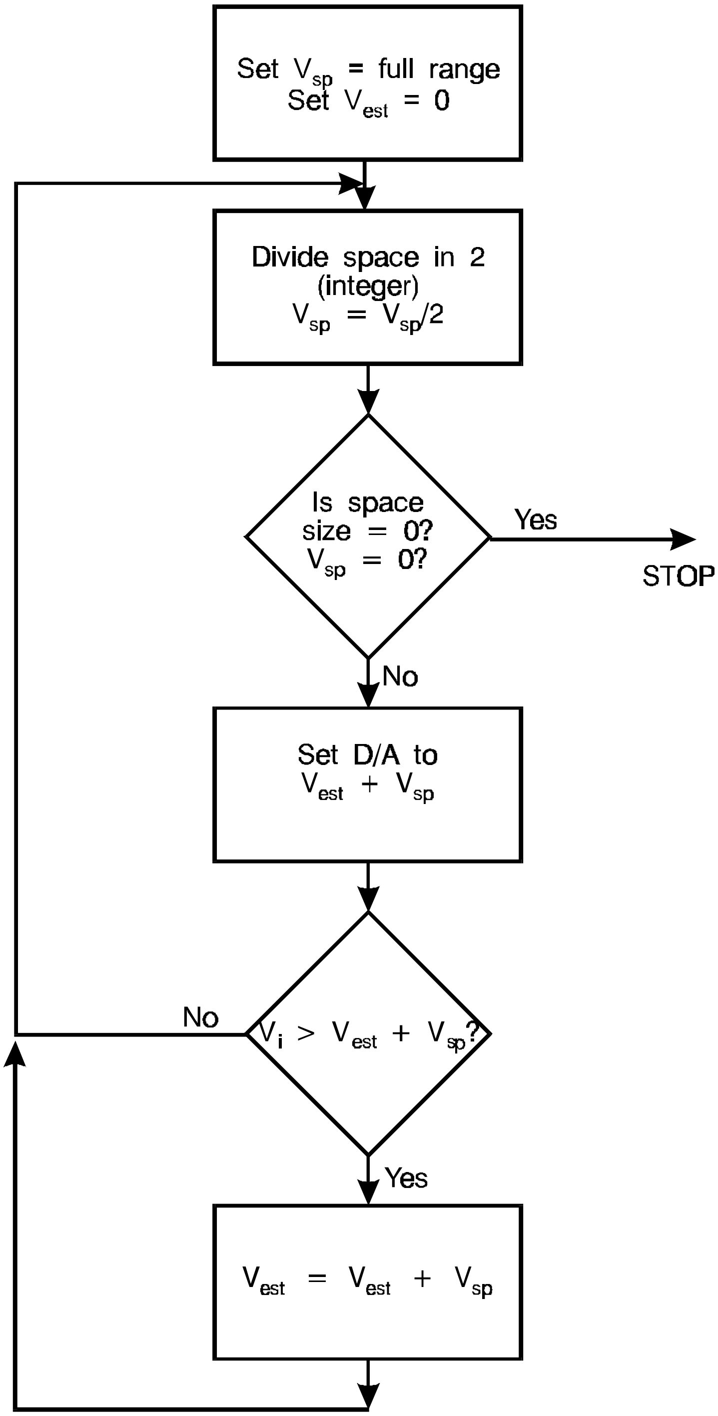


Binary Search Strategy

The second major type of algorithm is the so-called successive approximation, which is really a binary search of the space for the result. The value is first checked for presence in the upper half-space or the lower (>127 or <128 for an 8-bit converter). Having located the appropriate half-space, the process is then repeated on that space and so on. Thus to digitise a voltage of 5 volts in a space from 0-7 (actually levels of 0.5v - 6.5v remember) requires the steps as shown above.

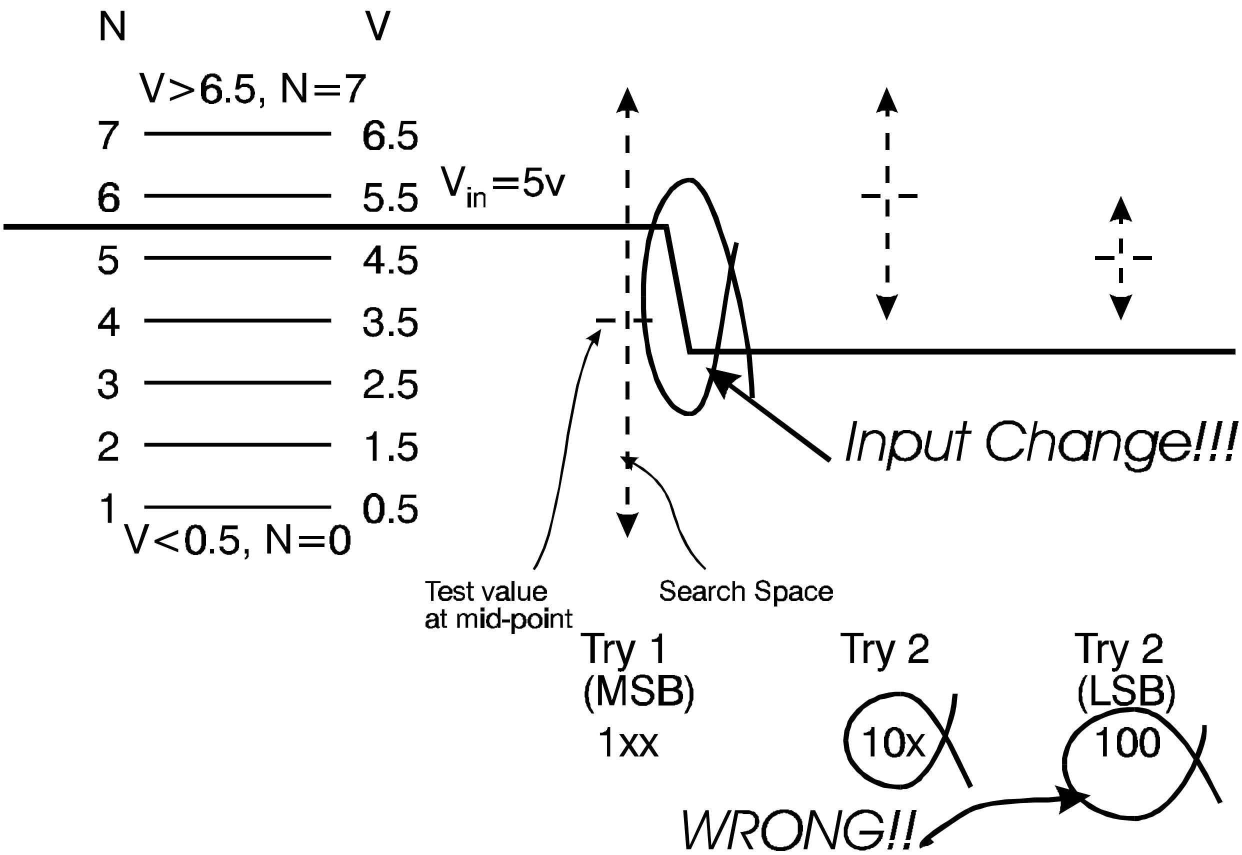
A flow chart of the operation is also given below where V_{sp} is the mid-point of the current space being tested, ie 0.5x the space size, and V_{est} is the current estimate of the voltage with zeros in the bit positions which have not yet been tested.

First the total possible range of values is divided into two and the unknown is compared to this value. If the comparator output indicates that the result is over the half-way value, then the half space is sub-divided into two equal spaces (at the 3/4 mark) and the value is compared to that. Depending on the result the upper or lower



Binary Search Strategy

quarter is divided and so on. The process terminates when the remaining space cannot be subdivided because it is only one bit wide



Binary Search Strategy - Problems

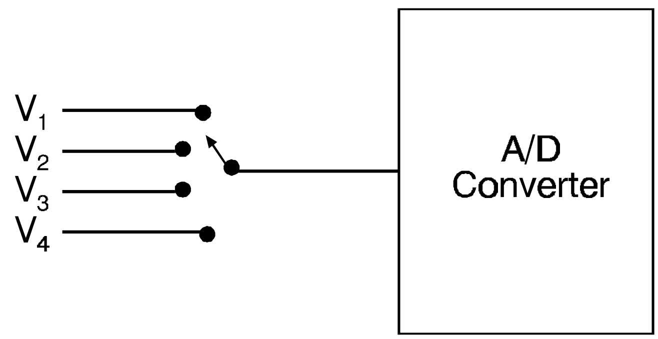
The advantage of this process is that it produces a result in a defined length of time, *n* cycles for an n-bit converter, and that this time is much smaller than the counting algorithm since $2^n >> n$. The disadvantage is that if the value to be measured changes during the measurement period, then the result can be wrong. this can be illustrated by the accompanying example where the voltage changes from 5 volts to 2 volts suddenly during the conversion.

"Successive approximation" converters are by far the most popular converters for microprocessors in circumstances where a small number of bits of accuracy are required (small $\leq$ 14). For larger numbers of bits a number of factors combine to make other solutions more attractive.

Calibration

One should never trust an ADC to be accurate because it might drift with time. ADC should therefore (like any other lab. instrument) be calibrated periodically. This can be accomplished in the traditional manner by manually connecting sources of known potential to it, or by incorporating a calibration cycle into the normal operation of the converter and installing calibrated sources in the system (which then need periodic checking against higher standard sources until one gets a traceback chain to the NRC standards)[23].

In circumstances where you cannot take the calibration on trust (ie always!), an input multiplexer and a calibration sequence is mandatory. The simplest input multiplexer consists of a set of relays which the computer can operate to select one of a number of voltages for monitoring. More complex solutions involve solid-state multiplexers.



Calibration of An A/D Converter

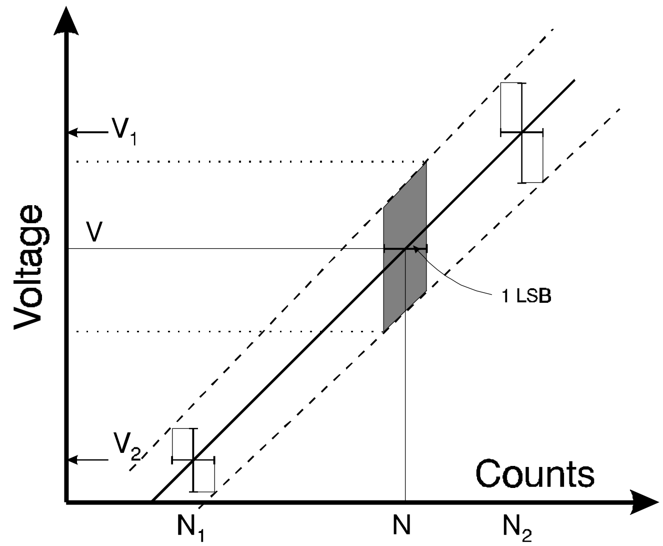
By making a couple of the voltages applied to the multiplexer known (i.e. standards) and assuming that the converter is linear, i.e. $N = mV + c$, then the use of two known voltages will allow a "2-point" calibration.

However note that there is a hidden problem here. If a number N is recorded then the number is really $N \pm 1/2$ since N has to be integer, so that the equation for calibration becomes:

$$V = \frac{(V_1 - V_2)N + V_2N_1 - V_1N_2}{N_1 - N_2}$$

$$\pm \frac{|V_1 - V_2| + \Delta V_1|N - N_1| + \Delta V_2|N - N_2|}{|N_1 - N_2|}$$

[23] Thought for the day: If I went into your laboratory and altered the calibration of all the voltmeters by 5% - how long would it be before you noticed?



Calibration of A A/D Converter - 2-point Plot

(Note: I'm assuming that the calibration voltages are on either side of the unknown as shown in the diagram, i.e. that we are *interpolating* not *extrapolating*.)

In order to minimise the errors and maximise the allowed range we should choose V_1 and V_2 as far apart as possible - preferably near the ends of the range i.e. just less than 2^n counts apart.

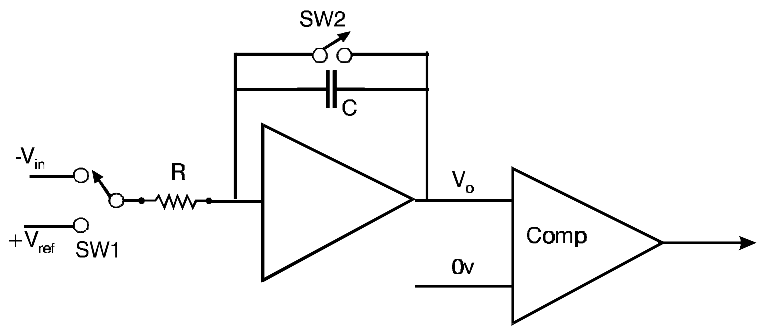
The calibration errors decrease the accuracy of the measurement and it may well be necessary to work to more bits than you think in order to get the answer that you want. Remember that I haven't included the inaccuracies of the converter yet!.

A practical consideration is not to let any calibration point be at a limiting value of the converter. Supposing we allowed zero volts to equal zero counts in a unipolar (only converts voltage of one sign, conceptual end of range at 0V and 0 counts) and then consider a zero drift. If a 0v input is indicated by a count of +1 instead of 0, then a drift in either direction can be monitored by putting 0v in as a calibration point and taking the result into the calibration equations. However if the drift is such that 0v should be represented by -1 counts we have a problem as we cannot represent numbers of less than zero. Thus the calibration points should not be extrema of the converter values under any conceivable set of drifts.

There are also other forms of error which can afflict an ADC but I want to delay discussing those until I have described another major form of converter - the dual-slope. This converter relies on the transformation of a measurement of voltage to a measurement of time.

Dual-Slope ADC

Consider this circuit.



Simple Dual Slope A/D Converter

In operation the integrator is first zeroed (close SW2), then attached to the input (SW1 up) for a fixed time *M* counts of the clock (frequency 1/*t*). At the end of that time it is attached to the reference voltage (SW1 down) and the number of counts *N* which accumulate before the integrator reaches zero volts output and the comparator output changes are determined.

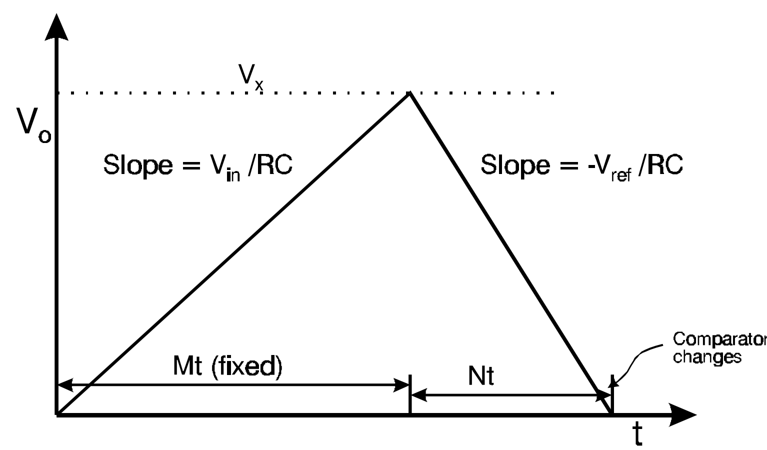
The equations of operation are therefore:

$$V_x = \frac{V_{in}(Mt)}{RC} = \frac{V_{ref}(Nt)}{RC}$$

$$V_{in} = V_{ref}\frac{N}{M}$$

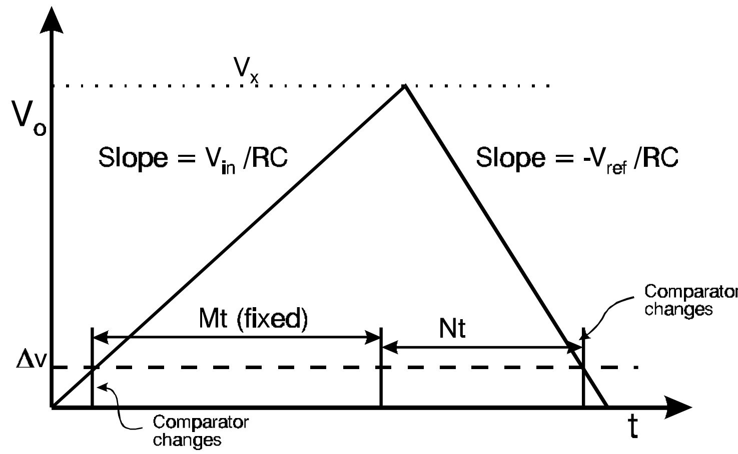
The unknown voltage is then just $V_{ref}*N/M$ and is reasonably independent of everything else.

The main problem with a simple dual slope ADC is in returning the converter to an exact zero before the start of each conversion as interpreted by an imperfect comparator.

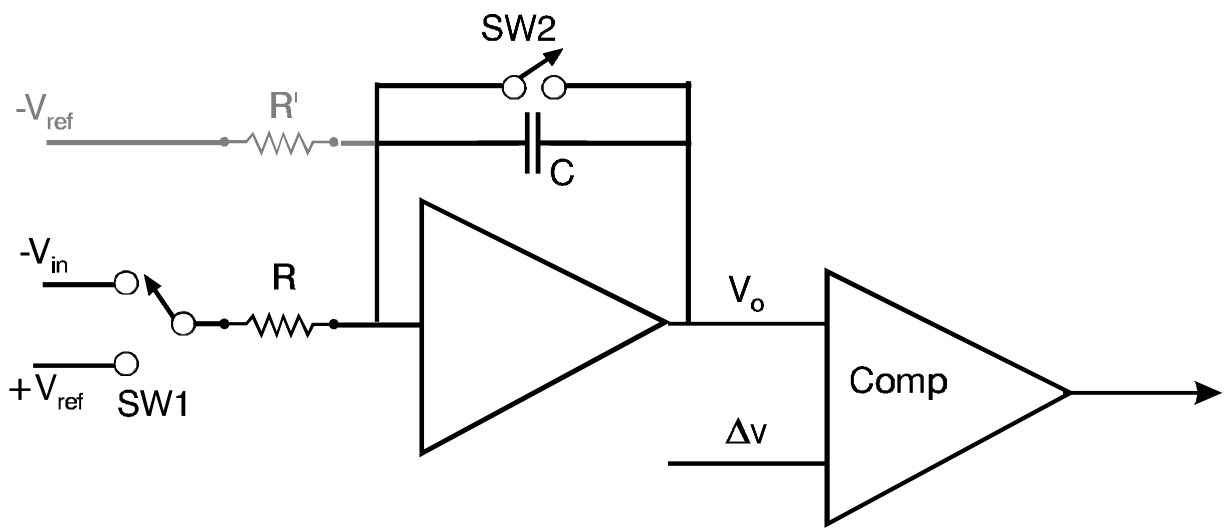


Dual Slope A/D Converter Output and Timing

This may be circumvented by changing the comparison level from zero volts to a small, stable, but not necessarily known, voltage (ΔV), which in our example must be more positive than any combination of imperfections in the comparison. The first timing interval then starts when the voltage crosses the level and the second interval finishes when the level is re-crossed at the end.

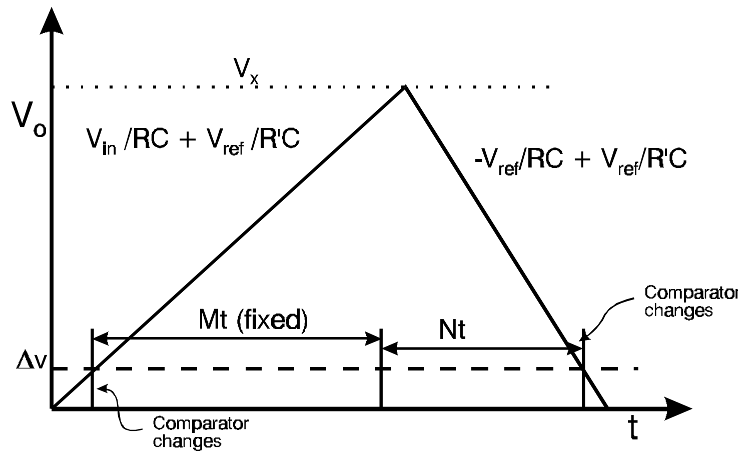


Dual Slope A/D Converter - Zero Offset



Dual Slope A/D Converter - Full Circuit

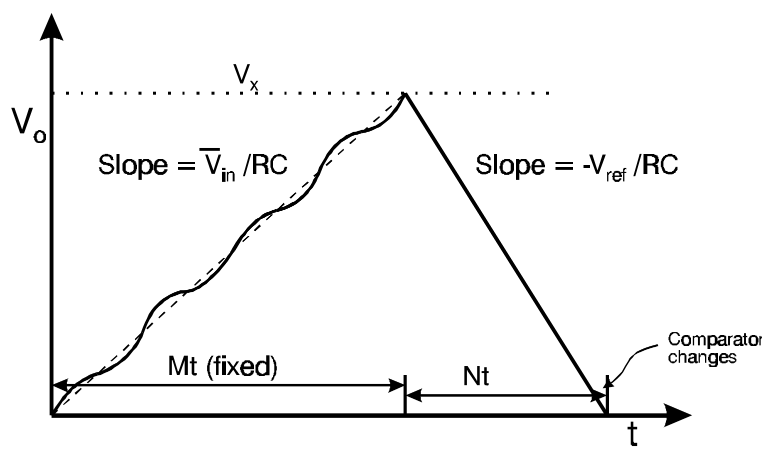
This brings with it a further problem that a very small input voltage will necessarily have a VERY long conversion time (takes a long time for the integrator output to reach ΔV), and a zero voltage an infinite time because the integrator output never reaches the comparator threshold. In order to circumvent that a small bias current is added to the integrator (using $-V_{ref}$ and R') to force the output to go positive even with an input of zero. Of course this means that zero volts is not zero output counts any more and a calibration step is required where a zero is established[24]. Notice however that despite the increase in complexity, all that is required out of all components except the reference voltage is stability over one measurement sequence, not accuracy. Those of you with an inventive mind might like to consider the problems of polarity of input.



Dual Slope A/D Converter - Full Circuit Timing

Further Considerations

There are two properties of the dual-slope converter (and a large number of related converters with slightly different but broadly similar operating philosophies) which make it peculiarly suitable for some purposes. The first is the *integrating* property and the second the *equal-interval* property.



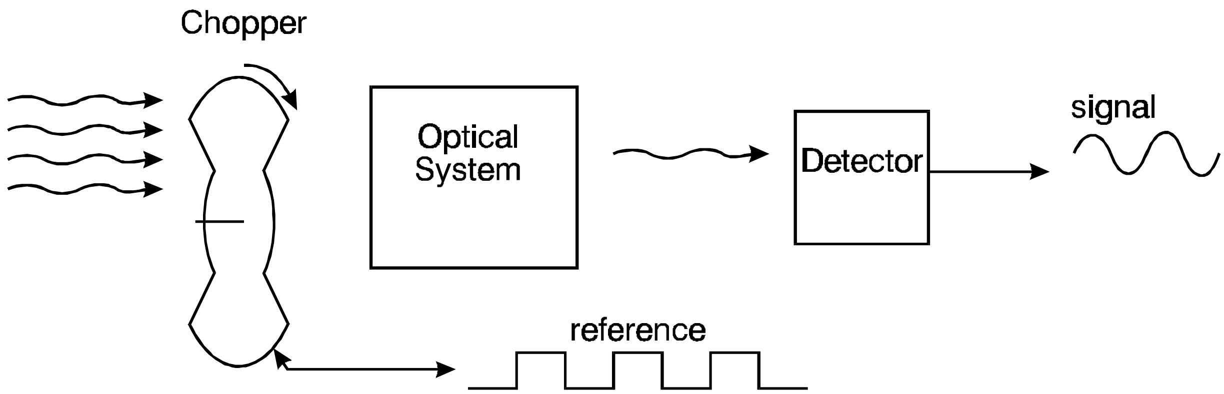
Dual-Slope Converter and a Varying Voltage

The *integrating* property distinguishes the dual-slope from successive approximation converters which digitise a

---

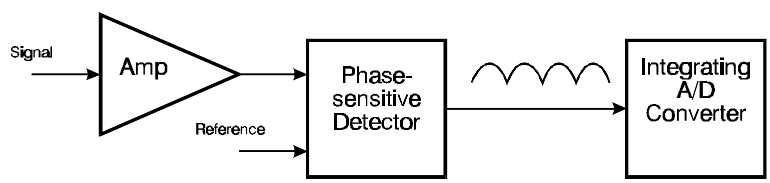
24 However remember that the system is still linear, we have only altered *c* in $V = mN + c$.

"spot" value. The input is integrated over a fixed time period (*N* clock cycles). This helps considerably with the problem of voltages changing whilst being measured because the converter effectively integrates for a fixed time period and then measures the result. This property can be turned to great advantage when measuring voltages which are fluctuating at a known precise frequency by making the first integration period an integer number of cycles long, e.g. if the input is d.c. + 60Hz ripple then a first integration time of 1/60S or any multiple thereof, will cause a true average to be taken because the component at 60Hz (actually any component at 60*n Hz) will average exactly to zero in 1/60S. The example is not trivial because most voltmeters use precisely this principle to eliminate line "hum" (60Hz) from their measurements. In physical experimentation the property may be turned to advantage as a "filter" to eliminate unwanted problems. To illustrate. Consider an optical system employing a chopper to interrupt the radiation.



A Chopped Optical System

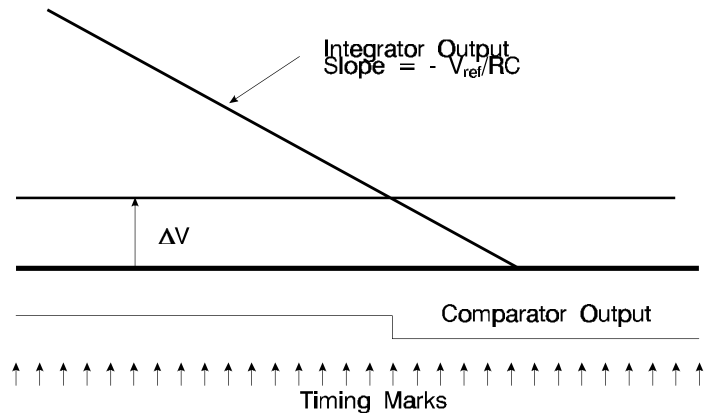
The signal from an optical detector is detected by a phase-sensitive detector and the processing system[25] is shown in



An Electronic Filter System

[25] *Phase Sensitive Detector* This is a common device to recover the amplitude of an ac signal where you know its frequency and phase. In its simplest form it consists of an amplifier whose gain can be switched between +1 and -1 by a logic level. If the logic input matches the ac signal in frequency and phase, then the output is a full-wave rectified waveform which can be averaged to recover the amplitude of the input. If the input does not match the logic signal in frequency and phase

the diagram. Using a dual-slope system whose first period is some integer number of cycles long (assuming that you know the frequency) will result in an exact d.c. average being output without further intervention.



Dual-Slope Converter at the End of the Cycle

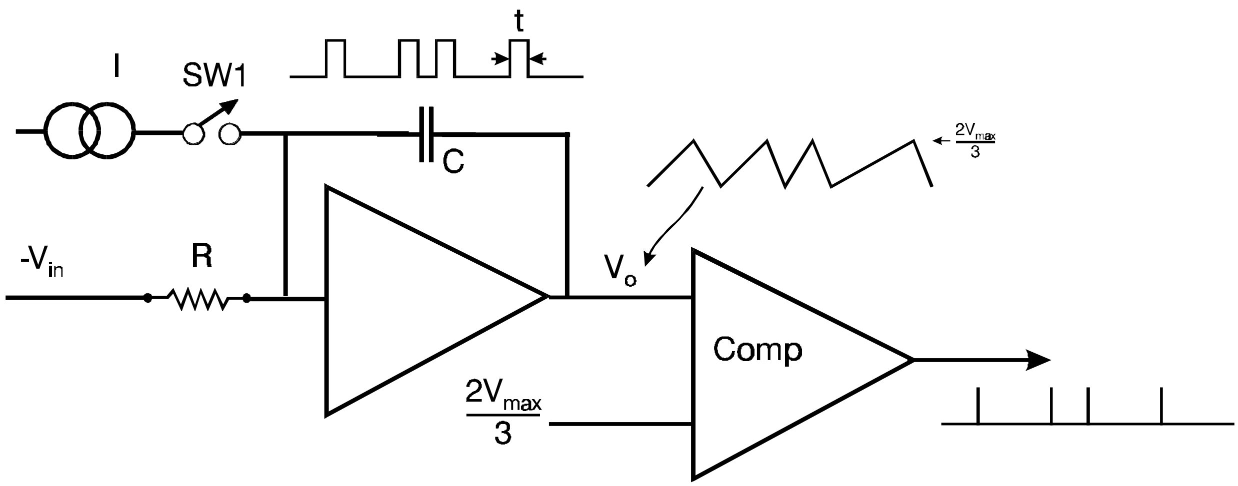
The second advantage is *equal interval*. This will be explained in more detail later but the important property to realise is that no count is preferred over any other count because the integrator does not know anything about the count period. Consider the output of the integrator at the end of the cycle as the comparator is about to change state: The slope of the integrator ramp-down is $-V_{ref}/RC$ which is independent of the input and the analogue circuit has no way of knowing the current count value. Therefore every count looks like every other count and they are indistinguishable from each other. Thus the input voltage change required to go from count 100→101 should be exactly the same as that for 1000→1001. This is in contrast to successive approximation ADCs where some states may be wider than others due to resistor tolerancing - the equivalent of DNL in a DAC. The equal interval property is equivalent to a DNL of zero.

---

then the signal is rejected to varying degrees depending upon the exact phase relationship and the degree of averaging of the output which is performed.

The Voltage-to-Frequency (V-f) Converter

A further example of an averaging converter is the voltage-to-frequency converter which uses a linear voltage controlled oscillator to produce a frequency output.



Schematic of a Voltage-to-Frequency Converter

Consider a simple integrating system with a switched current source as shown here. If, every time the comparator "flips" we switch on the current source for a time t and drain a charge It from the capacitor, the total charge collected in a time T is given by:

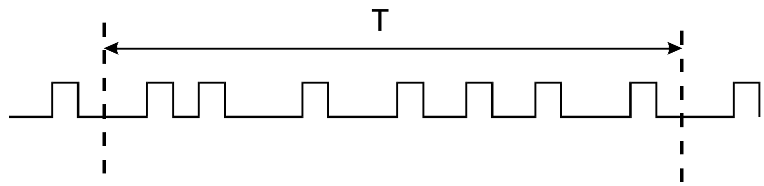
$$Q = \int_0^T \frac{V_i}{R} dt = \frac{\overline{V_i} T}{R} = NIt$$

where N is the number of "discharges" in the integration time T. The average frequency is therefore $f = N/T$.

We can express the relationship between the average voltage and the average frequency as:

$$f = \left(\frac{1}{RIt} \right) \overline{V_i} = mV + c$$

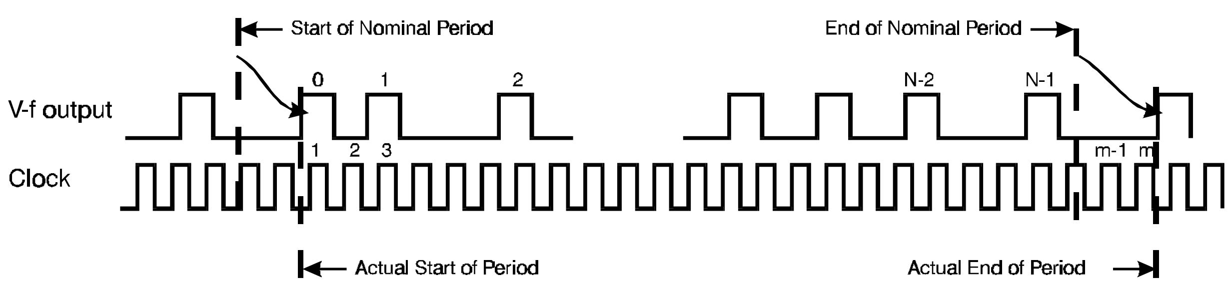
then, using calibration techniques we can easily recover the voltage by counting the output of the system for a fixed time. Note that I have included an offset term, c, in the equation, even though the ideal circuit does not have such a term - however ideal circuits do not exist in reality. If we record *N* counts then, because of the interaction of the "edges" of our count interval with the actual frequency, we count ± 1 of the average relationship. This imposes a resolution limit on our measurement but one which we can make arbitrarily small by making *N* as large as possible.



Counting a Simple V-f Output

So what limits the values of *N*? - boredom and the fact that we require some time relationship for our input, i.e. we want to measure the input this second and not this hour. V-f converters are normally limited in output frequency to about 100kHz which imposes a limit of $1:10^5$ in resolution (0.001%) for a one second measurement which is still inconveniently long. What is even more galling is that the converters can be linear to better than $1:10^6$ and we are losing precision due to the ± 1.

A solution to this problem is to measure the period of the waveform and then take a reciprocal to get the frequency. However we may run into the other problem that the period is too short for a good measurement. If we knew that the period was too short then we could measure say 10 periods and get the resolution that way. However there is a slightly more complex solution which solves all of these problems - if there is a handy computer.



Multiple Period Counting a V-f System

The trick is to make the total measurement period approximately constant but actually synchronous with the V-f output as shown in the diagram. The start of the nominal timing period opens a "gate". The actual timing starts from the first +ve transition of the V-f output after the gate opens. Similarly the end of the nominal timing period opens another "gate" and the actual timing ends at the next +ve edge of the V-f output. Within the actual timing period the number of V-f cycles N and the number of clock cycles m is counted. Then we know that there are $m \pm 1$ clock cycles of frequency F in exactly N cycles of the voltage-to-frequency converter and the output frequency is $NF/(m \pm 1)$. Since F may now be MHz, m can be extremely large and the resolution in a fixed time period is improved at the expense of some computing - hence the computer. As always there is a problem and the problem in this case is that N is actually variable and should not drop below 100 for a 1% uncertainty in the time location of the integration time. It is therefore necessary to offset the converter (make c in the linearity relation non-zero) in order to circumvent the problem. However here we come upon an interesting property of the converter - it is more linear in a restricted central range than it is over a very large dynamic range - and therefore by restricting the frequency range to say a 2:1 ratio (e.g. 1000-2000Hz) the potential resolution actually increases.

By the further use of multiple counters - and a computer - a continuous averager can be constructed which conserves all quantities (N and m) whilst allowing all the above properties. This allows a precise average over a longer time period to be determined in the post-processing phase of the data analysis.

The Delta-Sigma Converter

An alternative way of interpreting the output from a voltage-to-frequency converter is to regard the switch waveform (the set of pulses going to the current source switch) as continuous and sample it on a regular basis. Evidently if the frequency of switching is zero then the result will always be zero. If the output has a frequency of N pulses/sec each of length t, then the probability of getting a one as the result of a regular sampling at M samples/sec is Nt/M which is proportional to N which is in turn, the frequency of the output. If we regard the sampling output as the regular output of a one-bit averagable converter, then we can average the ones and zeros (as real numbers, of course) to get the average result.

Two things combine to make this a very interesting possibility:

a) The use of digital filters to "average" the output in a manner which gives the converter some very good properties, such as transient response and frequency response. We shall touch briefly on digital filtering later in the course.

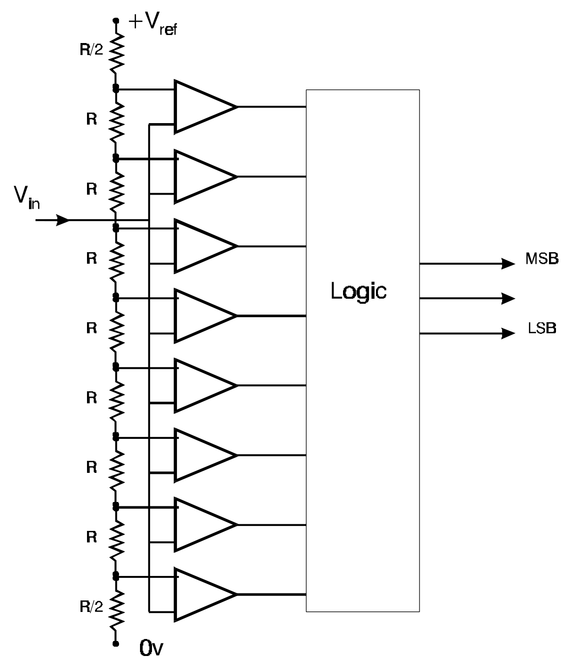
b) The fact that the input circuit can be pretty well anything so long as the "charge-balance" condition is maintained. Briefly stated, the charge-balance condition is that the charge being put onto a capacitor from a source must be balanced by the charge being pulsed off the capacitor by the converter. The use of a more elaborate circuit than a simple integrator and converter endows the converter with more flexibility in its response.

All this may seem somewhat esoteric, but these techniques are very powerful and an example is that 18-bit conversion is possible at 48kHz update rates for a very modest outlay using this method called "Delta-Sigma Conversion"

The "Flash Converter"

Every converter has its advantages - and the flash converter, converts in a flash, i.e. very fast. It is a brother to the successive approximation converter but whereas the successive approximation converter compares to a "guess" each time, the flash converter compares to all guesses simultaneously and then decides the "best" value from the results.

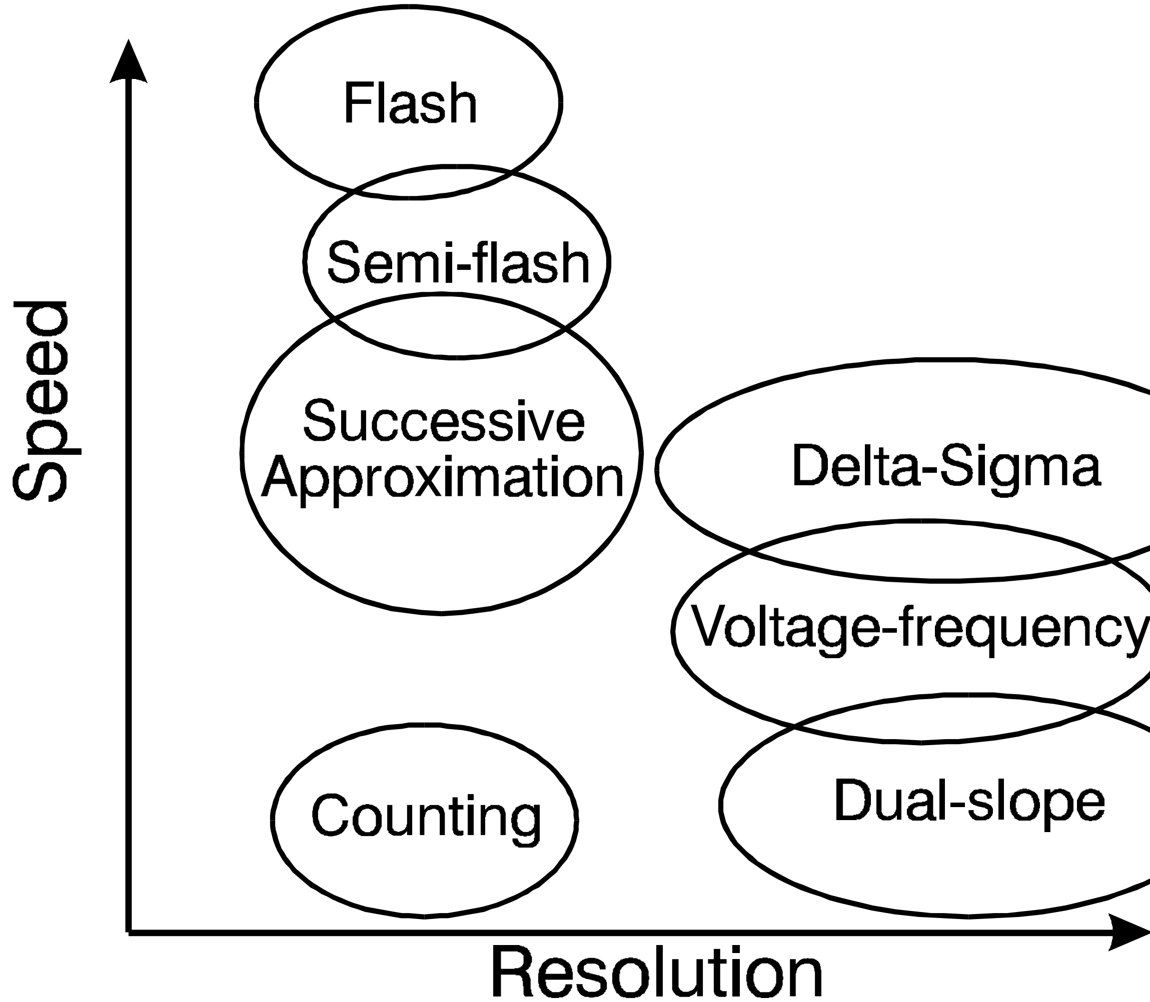
Note that a flash converter is very extravagant in hardware - but if you have to convert at 10Msamples/Sec you have to give up something!



A "Flash" Converter

Summary

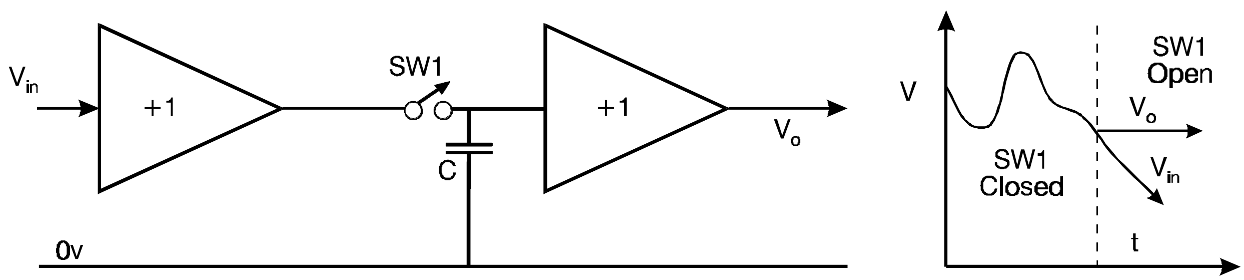
By now you will have gathered that specifying and selecting an ADC is by no means trivial - some of you will wonder if it can ever be done! However in practice you need to decide some basic parameters and then start checking the literature. The first thing to decide is to what extent you need the equal interval property. If you do, then this will tend (but there might be other ways round the problem) to select dual-slope, voltage-frequency and delta-sigma types. If you need the averaging property then dual-slope converters are the first (but not necessarily the only) choice. If these parameters are not important then the field is wide open. The accompanying diagram attempts to map the speed vs resolution space for converters to help with the problem.



Summary of A/D Converter Types

Sampling and Averaging

I have emphasised that various converters have various sampling methods and time-weightings ranging from the literal "spot value" of the flash converter to the "average" value of the dual-slope system with intermediate arrangements from other systems. It is really these other systems which cause the most trouble because they are designed with a constant input in view and do not take well to inputs changing during the conversion process.



Sample-and-Hold Schematic (Simplified)

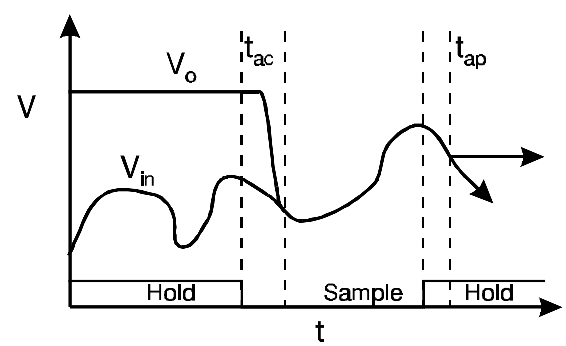
It has therefore become customary to "head-up" a converter with a "sample-and-hold" circuit which will allow the converter to get a "spot" value and then freeze it for conversion. A typical sample-and-hold is shown in the diagram and works as follows. When SW1 is closed the capacitor voltage follows the input and when SW1 is open it is frozen at the value at the instant of opening - almost.

Definition of Terms for Sample and Holds

There are a number of parameters associated with a sample-and-hold which need examining to see how far reality is close to ideality.

Acquisition time

The minimum time for the output to begin tracking the input voltage within a specified error band after the transition from hold to sample. Usually specified for a change in output corresponding to full scale. This tends to increase with increasing capacitor value



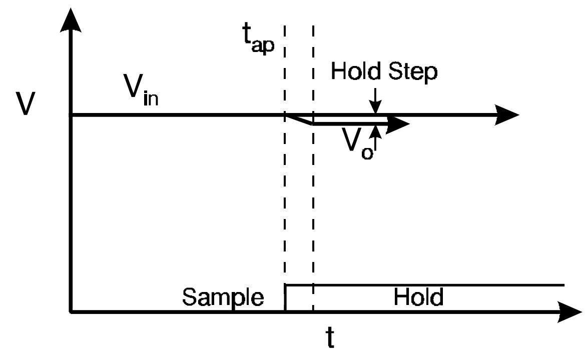
Sample and Hold - Acquisition Time

Aperture time

The reverse of acquisition time. The time after the transition of control signal from sample to hold that the output ceases to track the input.

Charge Transfer

The amount of charge transferred to the hold capacitor when the control level is switched. Charge transfer causes a change in the output given by $dV = dQ/C$ and therefore causes less trouble with large capacitors. The main cause of charge transfer is stray capacitance in the system. It may not seem a large effect but using a 1000pF hold capacitor, a 1pF stray causes a 0.1% change.



Sample and Hold - Charge Injection

Droop Rate

The rate of change of the output voltage when the control signal is in the hold state. The droop rate is due to leakage currents flowing on or off the hold capacitor and therefore the droop rate is less for large capacitors.

Feedthrough

The amount by which the output changes when the input changes when the system is in the hold state. Usually expressed as a ratio.

Full Power Bandwidth

Definition of bandwidth of the system for large signals when the system is in the sample state.

Settling Time

Time for the output to settle within a specified error band after initiation of the hold state.

Hold Step

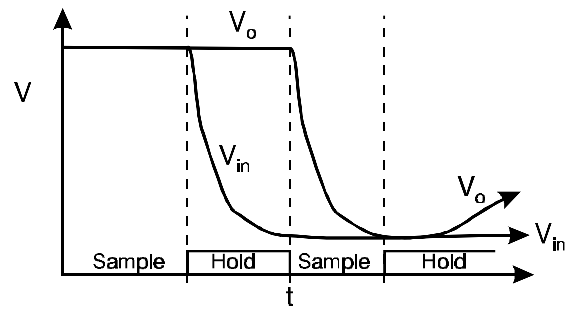
Change in output for steady input when control signal changes from sample to hold state.

Slew Rate

The maximum rate at which the output voltage can change. Often expressed in V/µSec or equivalent unit.

Memory Effects

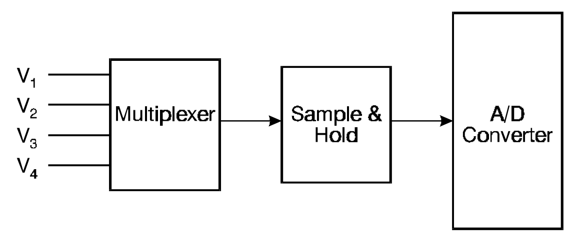
These are not specified on the spec. sheet. This occurs when a sample of +5v is followed by one of 0v. The output may be as shown in the diagram. Memory effects are caused by hysteresis in the capacitor dielectric and is minimised by the use of silver-mica, teflon or polycarbonate types in order of preference.



Sample and Hold - Memory Effects

This problem will particularly show up if the sample and hold is preceded by a multiplexer.

If the capacitor has "memory" then each value will be influenced by the last one. It is possible to take account of that in the data processing if need be - but it's easier not to introduce the effect in the first place.



Multiplexer - S/H - A/D

Summary

Notice that some parameters such as droop rate and hold step tend to suggest that larger capacitors are best, whereas others, such as acquisition time, suggest that smaller is better. Evidently there is going to have to be a compromise in practical cases.

Sample-and-hold tend to be simple in concept, usually OK in practice and have a nasty habit of causing trouble at inconvenient moments.