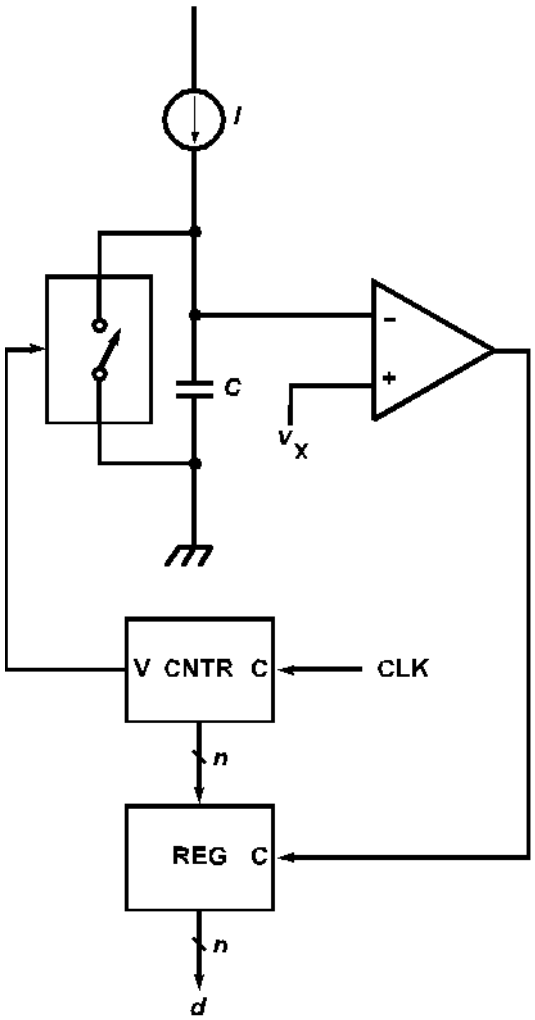
Simple µC-Based ADCs -- Part I
by Dennis L Feucht

Microcontrollers (µCs) have become a common circuit element -- albeit a rather involved one -- in instruments, consumer and office electronics, automobiles, motion control, and nearly everything else. The µC manages the interface between the user and the electronics, setting parameters of analog and discrete digital circuitry. The increasing speed of µCs extends their use to dynamic (as in real-time) control, as control elements in feedback loops.

One useful dynamic application of µCs is as ADCs. ADCs over the years have gone from complete DVMs as instruments to embedded subsystems, to a few components. This two-part article presents some minimalist techniques for implementing µC-based ADCs.

Minimalist A/D Conversion

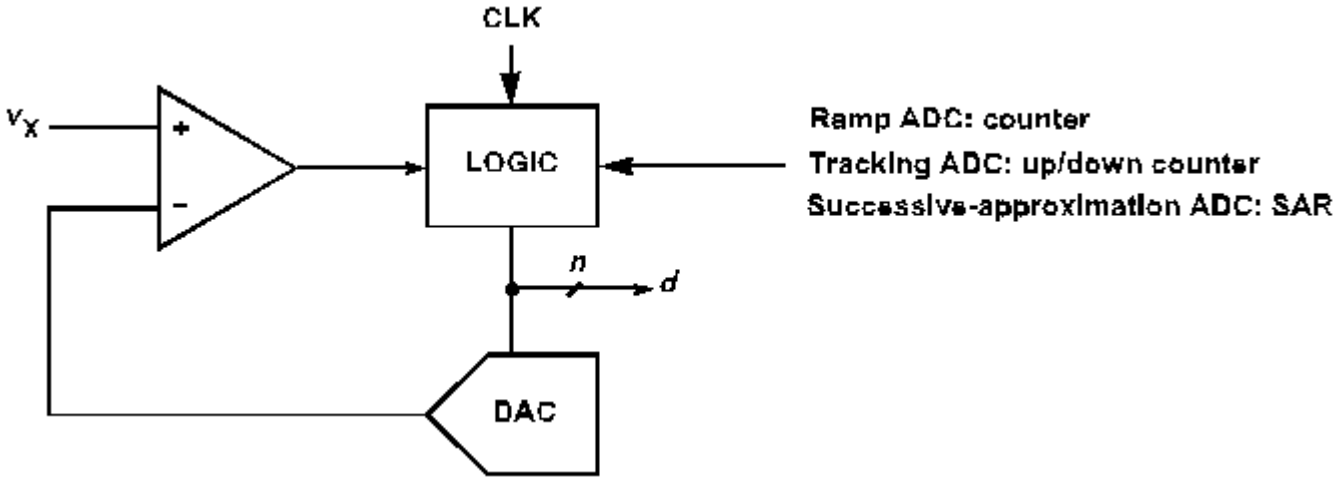
Multiple schemes have been devised over the years for A/D conversion. Some are simple, not very fast, and not very precise, such as the single ramp converter, as shown: charge a capacitor with a current source, thereby generating a ramp.



When the ramp crosses the unknown voltage, v_x, to be measured, the time interval from the start of the ramp to v_x, as detected by the comparator, is measured. For a (linear) ramp, the time is proportional to v_x. Timing

can be done with a counter that is reset (or overflows) at the start of the ramp and counts an accurate clock, such as a prescaled crystal-controlled μC clock. The ramp voltage is reset by a single transistor. This ADC is easily implemented using a μC, and requires one counter, an input bit-line from the comparator and an output bit-line to reset the ramp. The ramp ADC is subject to inaccuracy because the ramp slope can vary due to ramp current source variability, capacitor drift, and comparator input-voltage offset and bias-current errors. It is simple, but we can do far better than this with fewer components.

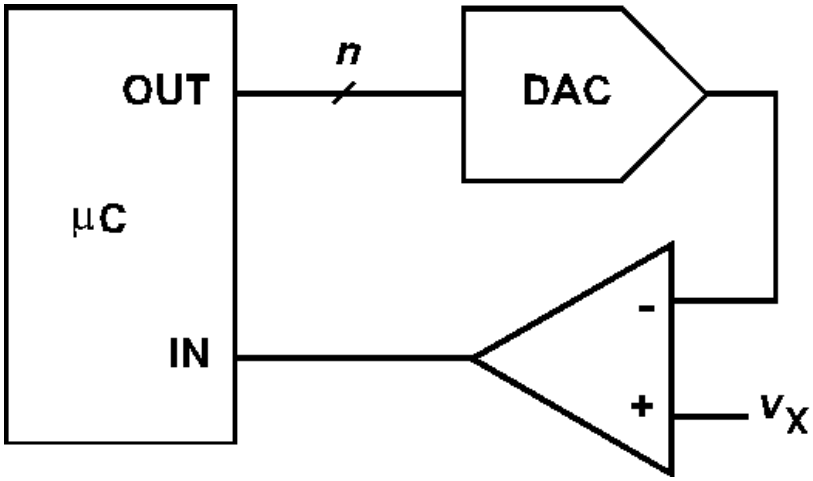
Another category of ADCs are the parallel-feedback types. They use a DAC and comparator to set bits of the converted measurement. The general scheme is shown below.



Depending on the logic, the converter can be one of multiple types. The successive-approximation scheme is predominant because it is a bit-per-iteration technique which takes *n* cycles to determine *n* bits, independent of the converted value of v_x. This scheme is well-suited for μC-based A/D conversion.

μC-Based Parallel-Feedback ADCs

The parallel-feedback converter can be adapted to most μCs as shown below.



The μC performs the logic. For the simplest and least desirable ADC, the ramp converter, the μC logic is given in algorithmic form below:

0. Ramp ADC
1. Set OUT to zero: OUT ← 0.
2. Input the IN bit.

3. If IN = 0, then VX ← OUT; go to 1.
   Else increment OUT: OUT ←□ OUT + 1.
4. Output OUT; go to 2.

The advantage of the tracking converter is that it follows the analog waveform, though it is subject to digital slew-rate limitations for fast changes in v_x. The tacking algorithm is given below.

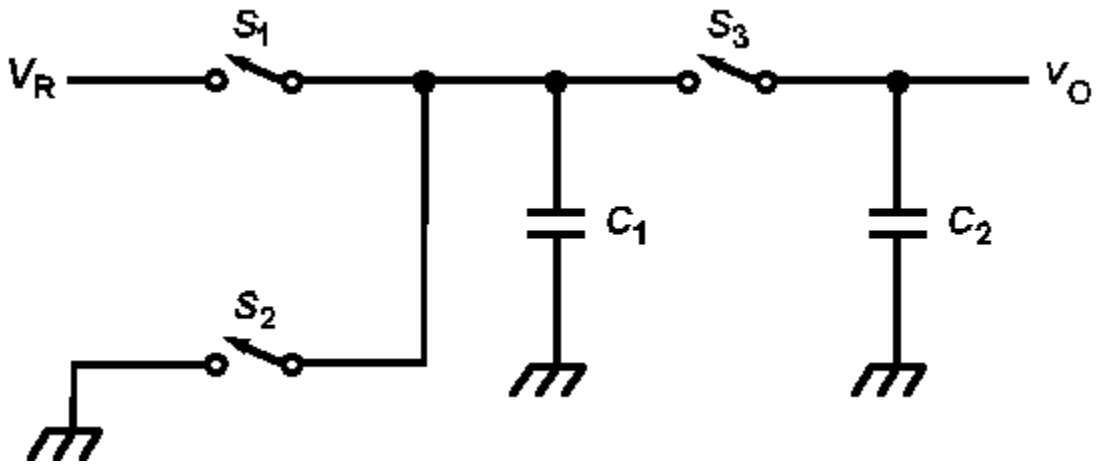
0. Tracking ADC
1. Output OUT.
2. Input IN.
3. If IN = 0, then decrement OUT: OUT ← OUT – 1.
   Else, increment OUT: OUT ← OUT + 1.
4. Set VX to OUT: VX ← OUT.
5. Go to 1.

This algorithm is not more complicated than that of the ramp converter but has the tracking advantage. The SA algorithm is somewhat more involved but has the advantage of constant, input-independent conversion time. It uses two memory locations, SAR and SR. C is the processor carry bit.

0. Successive-Approximation ADC
1. Clear SR and SAR: SR ← 0; SAR ← 0.
   Set C to one: C ← 1.
2. Rotate SR right, with C.
3. If C = 1, then return with VX ←SAR.
4. Output SR OR SAR to OUT: OUT ← SR OR SAR.
5. Input from IN.
6. If IN = 1, then go to 2.
7. Else, set SAR to SAR AND /SR: SAR ← SAR AND (NOT SR).
   (Alternative: SAR ←□ SAR AND (SR EOR 1111...).
8. Go to 2.

The 1 bit, initially in C, is shifted right, into SR, one bit per iteration. When it gets back to C (step 3 checks this), the procedure is done. Step 4 sets the SR 1 bit in the SAR. If the comparator (IN) is high, v_x is still greater than the SAR value, and this test bit remains set. If IN is low, the set bit made SAR too large, and it is cleared in step 7. Each bit, beginning at the MSB, is tested and then left set or cleared in SAR. Step 7 can use the clear-bit command for μCs that have it, with SR as the mask.

To reduce hardware in minimalist fashion, the DAC can be implemented as a μC-based serial DAC, below. Analog switch S3 requires a μC output-port bit. C1 and C2 are matched capacitors. A second port-bit output



can implement S1, S2, and VR, but must have a high-impedance state, or a three-state output; S1 and S2 must both be able to be set to off. The DAC reference supply, VR, uses the logic supply of the μC if S1, S2 are implemented by the bit-line output from the μC. If that is not good enough, then two external analog switches, switched by the output port bit-line are required for the more accurate implementation.

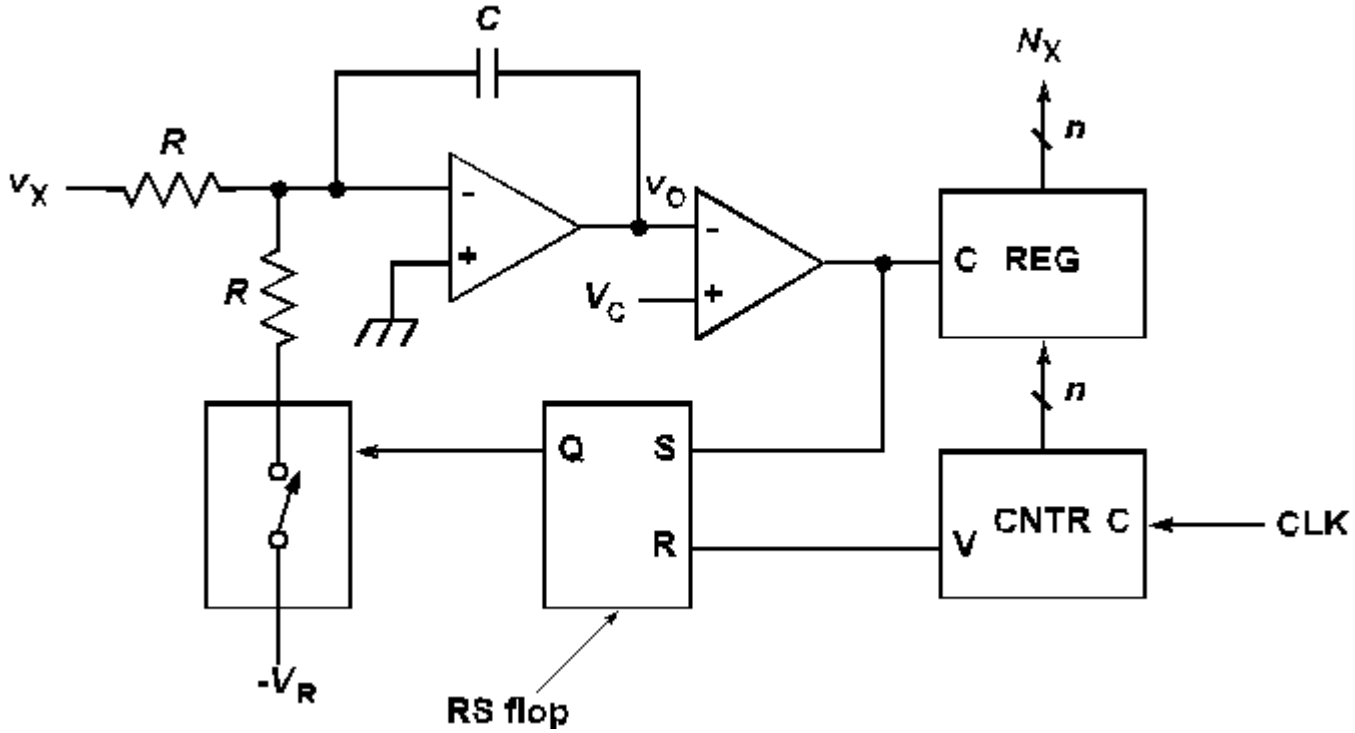
The serial DAC algorithm, implemented in the μC, is given below, where SW is the state of S3. OUT is the state of the C1 node, driven by S1, S2.

0. Serial DAC
1. From MSB to LSB for n bits: SW = 0 (open)
2. OUT = bn long enough to charge C; then OUT → hi-Z (open)
3. SW = 1 long enough for charge transfer between C1 and C2 to equalize
4. Decrement n. Go to 1.

Another way to implement a simple DAC is to use a μC PWM output. Filter it through an RC integrator, where the RC time constant is much larger than the PWM switching period. The result is a constant voltage with slight PWM ripple. The PWM DAC speed is limited by the low-pass RC filter but uses only two additional, low-cost parts, R and C. For μCs without a hardware PWM generator, the PWM also can be generated in software using a timer and an output-port bit, though the cycle period will be much longer than hardware PWMs.

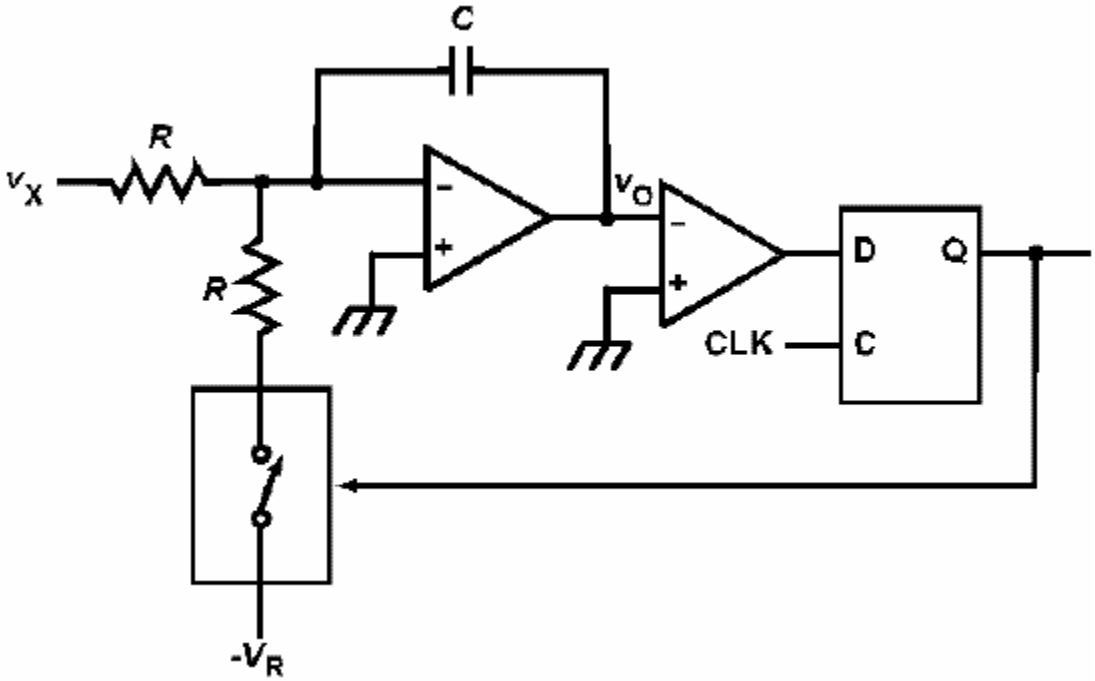
μC-Based Σ-Δ ADCs

A major improvement historically over the single-slope converter was the dual-slope converter, often



simplified to the modified dual-slope converter, as shown above in its non-μC instantiation. It has a minimalist appeal in that it has one analog switch, the SPST reference input switch.

Based on modified-dual-slope converter circuitry, an algorithmic variation that appeared in the late 1960s was the charge-balancing or sigma-delta (Σ-Δ) converter. (Some call it the delta-sigma converter.) Instead of making a current-source switching decision once each measurement cycle, decisions are made every clock cycle instead, in an attempt to maintain a virtual ground at the input. The fraction of cycles requiring the opposing reference current to maintain virtual-ground "balance" gives the acquired count. A non-μC Σ-Δ circuit is shown below.



The circuit topology is very similar to the modified dual-slope ADC. The circuit difference is that the flop driven by the comparator is clocked, and is a D-type flop instead of an RS flop. On a given cycle of the clock, the reference is switched in or out of the integrator to keep v_o near ground. The comparator output is the sign of the error. In other words, v_o is nulled by discrete-time feedback. The number of clock cycles that the flop was high, N_X, over the total number of conversion counts N, is the measure of v_x.

The transfer characteristic is derived by constructing the charge-balance equation for the total charge from the v_x and $-V_R$ inputs to the integrator. For $v_o = 0$, they must be equal, or:

$$Q_X = Q_R$$

These charges are the sums of the per-cycle charges:

$$q_X = \left(\frac{v_X}{R}\right) \cdot T_{\text{CLK}}, \qquad q_R = \left(\frac{V_R}{R}\right) \cdot T_{\text{CLK}}$$

The total charge of each depends on the number of cycles each is integrated. Then:

$$Q_X = q_X \cdot N = \left(\frac{v_X}{R}\right) \cdot N \cdot T_{\text{CLK}}, \qquad Q_R = q_R \cdot N_X = \left(\frac{V_R}{R}\right) N_X \cdot T_{\text{CLK}}$$

Then substituting and solving for the measured output count:

$$N_X = \left(\frac{v_X}{V_R}\right) \cdot N = \left(\frac{v_X}{V_R}\right) \cdot 2^n$$

for an n-bit conversion-time counter. This result is, incidentally, the same as for the modified dual-slope converter.

The charge-balancing circuit is also used as a modulator for serial digital telecommunications (in CODECs) and speech processing and can be used in its μC form for these applications too, if the conversion rate is high enough. Parallel-feedback-derived ADCs for μCs are faster than Σ-Δ ADCs, but require more external hardware. When a minimum external-parts-count converter is all that is needed, however, the Σ-Δ ADC is hard to beat.

The next part of this article presents some minimal parts count μC-based Σ-Δ converters and software algorithms for implementing them.