Architecture for a High Speed SAR ADC

Joseph Palackal Mathew
Department of Electrical Engineering
University of California,Los Angeles
jpmathew@ucla.edu

July 25, 2011

Problem Definition

Derive a topology that will minimize power while achiveing 12 bits of resolution and 200MSPS speed . Make topology amenable for future interleaving that can improve speed.

Problem Analysis

We are given an input $x \,\epsilon\, [-1, 1]$ and converter tries to find K such that

$$-1 + K\Delta \leq x \leq -1 + (K + 1)\Delta.$$

If input is uniformly distributed then the problem will require a minimum $N = log_2(2/\Delta)$ "Yes or No" questions with precise answers. From a circuit perspective "Yes or No' is that of a comparator comparing input with a threshold with infinite precision. If the comparison has finite precision i.e it is unreliable for $|inp| < \epsilon$ then at the end of N questions input can be as far as ϵ away from the said range.

A typical SAR ADC conversion proceeds as follows . We compare the input with a threshold $V_{th}(k) = r(k) * (x_{max}(k) + x_{min}(k)) + \epsilon(k)$ where it's guranteed by previous knowledge that $x \,\epsilon\, [x_{max}(k), x_{min}(k)]$.At the end of comparison the input is guaranteed to be in $[x_{max}(k), V_{th}(k)]$ or $[x_{min}(k), V_{th}(k)]$ so by setting $x_{max}(k + 1)$ and $x_{min}(k + 1)$ accordingly we can gurantee that range in which input can lie $S(k + 1) = x_{max}(k + 1) - x_{min}(k + 1) < S(k)$ provided that we put the thresholds $V_{th}(k)\,\epsilon[x_{max}(k), x_{min}(k)]$. This by iteration can guarantee that we can locate input to a finite precision in required number of steps.

Sampling Circuit

Noise power due to Differential sampling referred to input

$$P_{noise} = 2 * \frac{KT}{Cs} * \frac{Cs + Cp}{Cs}$$

where Cs = sampling cap and Cp = parasitic cap on top-plate , assuming top plate switch is limiting bandwidth and contributing most of noise compared to bottom plate switch . This is a reasonable assumption as every attempt will be made to minimize the top plate switch to reduce charge injection offset. Input referred SNR assuming differential swing of $\pm V_{REF}$

$$SNR = {V_{REF}}^2/2 * P_{noise}$$

Assuming a shrink of .8 and $V_{REF} = 0.9$,for a noise performance equaling 12 bit quantization noise

$$P_{noise} = \Delta^2/12$$

$$Cs = 600fF$$

0.1 Clocked Comparator

In a latched comparator $V_{in} * e^{t/\tau} = VDD$. For a minimum resolution input V_δ it will need a time $T_\delta = \tau * ln(VDD/V_\delta)$.Input greater than V_δ needs time less time to resolve than T_δ.An asynchronous ADC uses this fact for optimization by making every comparison time just enough to make accurate decision by checking output differential level against VDD or timing out after T_{max}. Hence

$$T_{cmp} = \tau * ln(VDD/V_{in}).$$

This makes the conversion time input dependant and the worst case comparison time = $N^2/2 * \tau * ln(2)$ is a two fold improvement in comparison time. But power saving may be significantly higher if actives are powered down after conversion and negligible if all power is dynamic. For a latch $\tau = Gm/C = \beta * \sqrt{(w)}/C_{oxl} * W + C_{load}$. if offset is non critical (SAR) or calibrated (multibit) no optimization is possible in this end. At higher precision ($V_\delta = 10mV = 8bit$) noise (thermal/capacitive coupling noise $\propto \frac{1}{C}$) from this may become critical and a switch from a noisy to low noise latch may be required.This will increase W and hence τ.

A Way to improve the comparison time from $O(N^2)$ to $O(N)$ will be to add gain stages in front of the Clocked comparator to the tune of 2^k for k^{th} comparison . This will make compartison time constant for all clocks making comparison time $N * \tau * ln(2)$. But this will add to settling portion of converision time and will be a trade off . This will automatically improve noise performance w.r.t clocked comparator.

Another option is to add a pipelined clocked comparator to the first one but not wait for it . So output of second latch will be $V_{in} * e^{(t1+t2)/\tau}$ making its resolution better

Non binary SAR

In a SAR ADC without any redundancy k_{th} DAC step S(k) needs to settle to $1LSB = VREF/2^N$ for ADC to to converge correctly . In a constant clock environment this requires a time $T_{dac} = \tau * N * ln(2)$ corresponding to worst case first step and hence a total DAC time of $\tau * N^2 * ln(2)$. If each DAC step gets a settling time just enough to meet the settling requirement then $T_{dac}(k) = \tau * (N - k) * ln(2)$ and total dac time will be $\tau * N(N-1)/2 * ln(2)$. To make T_{DAC} a constant we need redundancy propotional to current step so that an error $S(k) * e^{-T_{dac}/\tau}$ can be tolerated .For this

$$\sum_{(k+1)}^{N} S(i) = (1+m) * S(k)$$

$$\sum_{(k+2)}^{N} S(i) = (1+m) * S(k+1)$$

$$S(k+1) = (1+m) * (S(k) - S(k+1))$$

$$S(k+1) = (1+m)/(2+m) * S(k)$$

This Points to a non binary radix $r = (1+m)/(2+m)$. since redundancy greater than a step is not required $m < 1$ and $r < 2/3$. Total conversion clock $= \tau * ln(1/m) * ln(2) * ln(m+1)/ln(2+m)$. Though this points to a near zero conversion time with m=1 its a mathematical anomally because of approximating error as $S(k) * e^{-T_{dac}/\tau}$. More correct Derivation is as follows (assumes non binary sar)

$$settlingerror = \sum_{1}^{k} S(i) * e^{(-(k-i)*T_{clk}+T_{dac})/\tau}$$

$$S(k) = 1/2 * r^{(k-1)}$$

$$settlingerror = 1/2 * r^{(k-1)} * e^{-T_{dac})/\tau} * \sum_{0}^{k-1} r^{-}i * e^{-iT_{clk}/\tau}$$

$$settlingerror = 1/2 * r^{(k-1)} * e^{-T_{dac})/\tau} * (1 - r^{-}k * e^{-k*T_{clk}/\tau})/(1 - r^{-1} * e^{-T_{clk}/\tau})$$

$$e^{-T_{clk}/\tau} < r$$

$$settlingerror \ = 1/2 * r(k-1) * e^{(-T_{clk}/\tau)}/(1 - r^{-1} * *e^{-T_{clk}/\tau}) = m * 1/2 * r^{(k-1)}$$

$$e^{-T_{dac}/\tau} = m/(1+rm) < r$$

$$r \ = 1/2$$

$$T_{dac} \ = 2 * ln(2) * \tau;$$

In a constant clock binary asynchronous ADC 's advantageous to constraint

$$T_{cmp}(k) + T_{dacNamp}(k+1) = T_{Clk}$$

This is because if $(k+1)^{th}$ decision is critical k^{th} decision is non critical and should take less time which can be used to make $(k+1)^{th}$ decision more precise.Also $(k+1)^{th}$ decision will take more time but since $(k+2)^{th}$ deciison is noncritical its dac settling error doesnt matter.