Application Report

SPNA118B–September 2011

ADC Source Impedance for Hercules™ ARM® Safety MCUs

Jim Childers

ABSTRACT

Unbuffered multiplexed ratiometric analog-to-digital converters (ADC) have strict requirements on driving source impedance that are not always obvious. This application report addresses the trade-offs between source impedance and sample rate. It includes both 10-bit and 12-bit examples using the TMS470 processors and the Hercules ARM Safety MCUs (TMS470M, TMS570 and RM4x families) in the GS30, GS40, and GS60 process nodes (a.k.a. F05/C05, F035/C035, and F021, respectively).

Contents

List of Figures

List of Tables

Hercules is a trademark of Texas Instruments.
ARM is a registered trademark of ARM Limited.
All other trademarks are the property of their respective owners.

1 Introduction

Unbuffered multiplexed ratiometric ADCs are commonly used in microprocessors due to their simplicity of design and inherent absence of circuits, which need trimming in production. Both multiplexed and unmultiplexed versions are commonly found in discrete form and are probably the most common ADCs in existence. They have no internal buffer amplifiers to introduce input offset and gain errors, and no internal voltage references that might induce scaling errors.

Designers are accustomed to using these ADCs for a variety of low-frequency applications. Over the last two decades, they have been included in virtually every microprocessor family from every company and have increased in conversion speed right along with the microprocessors that host them.

A side effect of their advantages is that the sample capacitor within the ADC is directly charged by the external signal and ever-increasing speed has made this a growing issue. While it may seem like a trivial problem to charge a 12 pF sample capacitor, at high conversion speeds it can be problematic to charge it to within ½ LSB in the allotted time.

Also, if the sample time is insufficient, then the charge left on the sample capacitor by the previous conversion of a channel can affect the accuracy of the channel currently being converted. This phenomenon is referred to as channel-to-channel crosstalk.

2 System Model

To start with, examine the overall environment in which the ADC is used. A model of the ADC system should include everything from the sensor or signal source to the ADC insides. Figure 1 partitions the system into four distinct blocks that can be discussed individually.

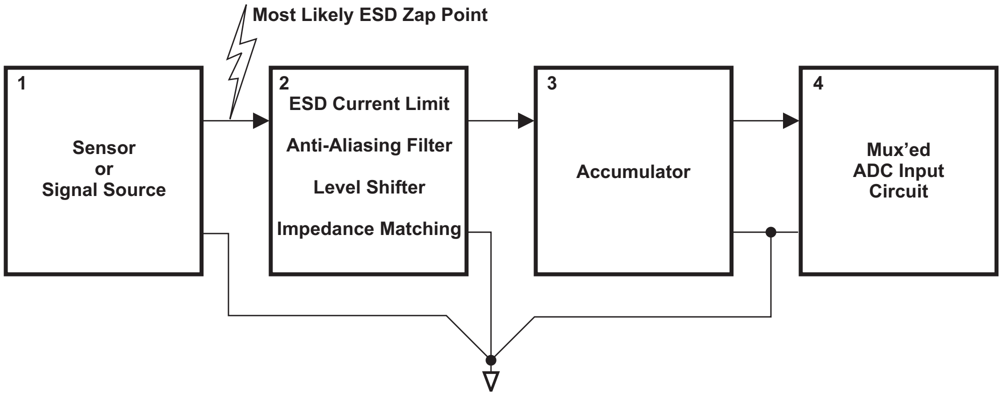

Figure 1. System Model

Block 1: The sensor can be virtually anything from a sophisticated mass air flow sensor to a brick striking a piezoelectric crystal. As such, the source voltage can range from microvolts (as from a thermocouple) to several thousand volts (the brick hitting a crystal). The source impedance and frequency can range similarly. With this in mind, not much can be said about the source except that it clearly sets the requirements for the input of Block 2.

Block 2: This might best be described as a matching circuit. It has many simultaneous requirements to fulfill as noted in Figure 1.

It must maintain at least enough series resistance between the ESD entry point (if applicable) and the ADC input pin to protect the ADC's input from being damaged. For example, to pass the 4KV Contact Model ESD test, about 3000 Ω minimum resistance is required between the zap entry point and the ADC pin (most of the ADCs have 2KV ESD protection).

Any time something is digitized, it is essential that no information above the Nyquist frequency greater than a *no effect* level be introduced into the sampled signal. That means -67dB for 10 bits or -79dB for 12 bits. Once that *noise* is digitized, it is indistinguishable from the desired signal, so it better be small! Therefore, the cutoff frequency of a low-pass filter (anti-aliasing filter) must be strategically positioned between the desired maximum signal frequency, f, and half of the input channel sampling frequency, fs/2 (Nyquist frequency). This filter is optional in some cases since some things just don't change very fast like the output of a thermistor, for example.

A level shifter is often required to match the peak signal level of the input to the nominal 3.3 V (or 5 V for some ADCs) swing of the ADC's input to optimize the overall dynamic range. This circuit may be as simple as two resistors acting as a voltage divider, an active circuit like an opamp, or a sophisticated automatic gain control (AGC) circuit like that used with a variable reluctance speed sensor.

Impedance matching is often necessary to match a higher impedance sensor or level shifter to the requirements of Blocks 3 and 4. The impedance requirement of Blocks 3 and 4 for a given channel is dictated by the sampling frequency, fs, of that channel. While the previous three items in Block 2 are generally well understood by designers; the true requirements for source impedance to the ADC inputs are sometimes misunderstood. Understanding the ADC's source impedance requirements are the crux of this application note.

Block 3: This block is optional depending on required sampling speed, cost, and other factors. If it exists, it is simply a capacitor. It accumulates charge in continuous time, which can then be charge-shared with the ADC's sample capacitor during the discrete-time sampling window of that channel.

Block 4: This is the ADC itself. Since the ADC is a single converter that is time-multiplexed with generally 16 or 32 input channels, it demands more attention at design time than if it was a converter-per-channel.

3 ADC Input Model

This is the ADC itself. Since the ADC is a single converter that is time-multiplexed with generally 16 or 32 input channels, it demands more attention at design time than if it was a converter-per-channel.

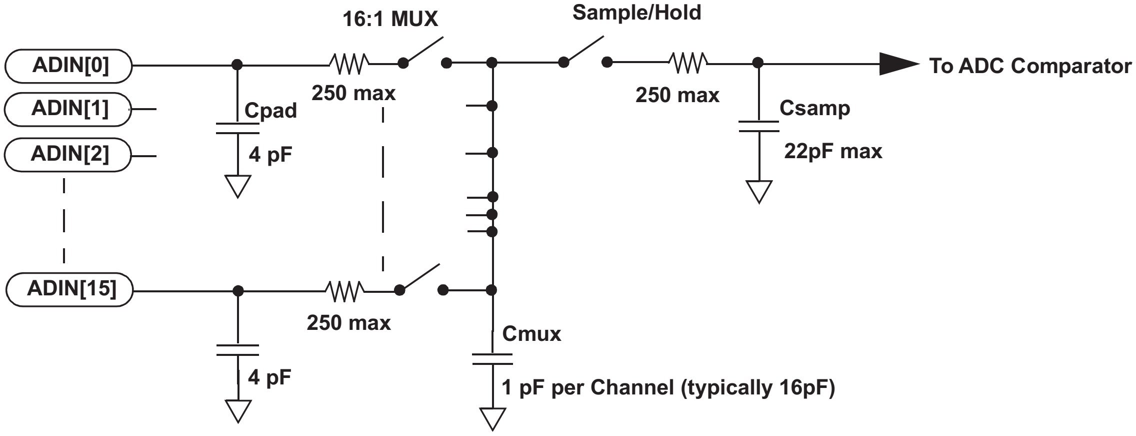

Figure 2. ADC Input Model

There are two CMOS switches in the path between the ADINx pin and the sample capacitor, Csamp. The first is an N-to-1 multiplexer that selects the channel to be converted; typically, N = 16 or 32. The second is the sample-and-hold gate that is controlled by the ADC's successive approximation state machine.

4 External Components

As mentioned previously with Blocks 2 and 3, it is common practice to add external components to the ADIN[X] pins that scale and filter the signal from the analog source. These components are determined by the requirements set by Blocks 1 and 4. A fairly typical circuit is shown in Figure 3.

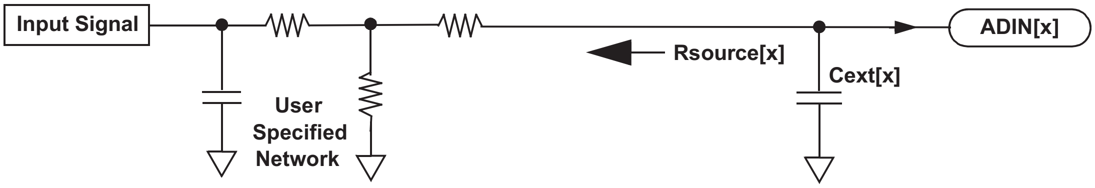

Figure 3. External Components

Generally, most users place a large capacitor (Block 3) from the ADC pin to ground ($\text{Cext}_{[X]}$ in Figure 3). This capacitor is used to lower the source impedance of the channel as seen by the ADC so that the internal sample capacitor can be charged quickly.

As noted earlier, this is a charge-sharing process between Cext and (Cmux + Csamp) (refer to Figure 2 and Figure 3), whose RC time constant is primarily determined by the maximum ADC input resistance (2 switches with 250 Ω max, each), the maximum multiplexer capacitance (16 pF), and maximum sample capacitance (13 or 22 pF) of the ADC.

In Figure 3, as Rsource is increased, the cutoff frequency created by Rsource and Cext will be lowered. This means that the response time increases between when the sensor changes and when the change settles on Cext.

5 Symptoms of Inadequate Settling Time

In Figure 4 through Figure 6, oscilloscope pictures are faked to demonstrate the effect of having insufficient settling time for Csamp. For example, say there are two channels that you are converting in sequence within a 10 µS Group Cycle time. The first channel has a 100 Hz squarewave on it, and the second channel is a DC signal.

For the first oscillograph, assume the settling time is adequate for the chosen sample frequency. In other words, this is what you expect to see on the two ADIN pins when everything goes right:

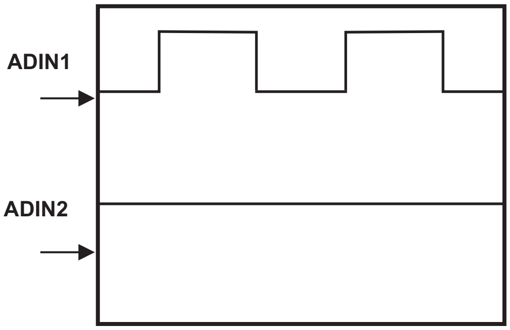

Figure 4. Adequate Settling Time

Now, reduce the allotted settling time so that the ADC only settles to within a few LSB rather than ½ LSB. After all, these two signals are very low frequencies: 100 Hz and DC. Why should they have to settle? Let’s see what happens:

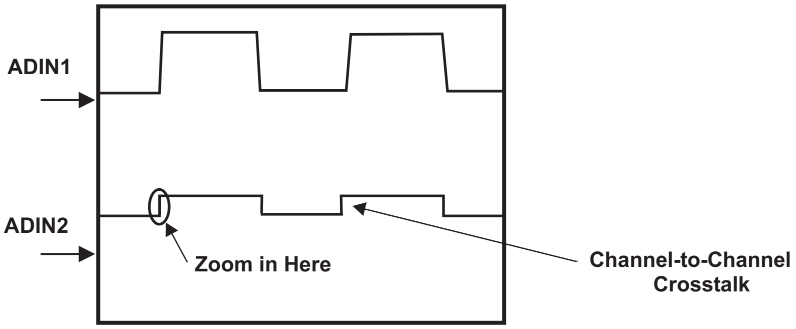

Figure 5. Insufficient Settling Time

The top waveform has lost some bandwidth so the corners aren't quite so square. But this may not be an issue if you are only interested in it's min and max values. But look at what happens to the second waveform. It has picked up crosstalk from the previous channel. If it was intended to be a DC level, this signal has been rendered just about useless. But what is the source of the problem?

Zoom in as indicated in the previous figure on the lower trace's rising edge:

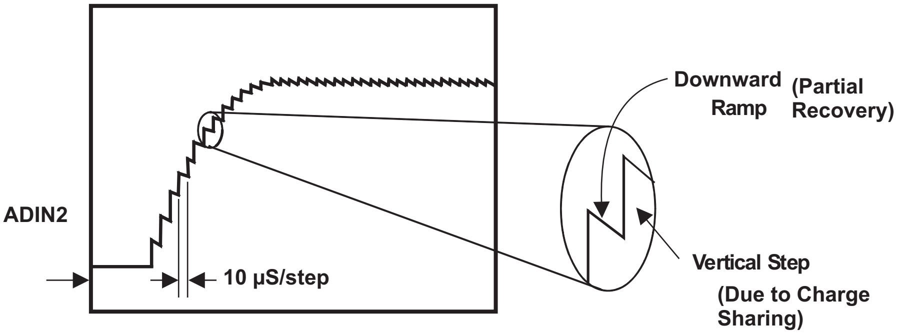

Figure 6. Close Examination of the Edges

You should see a saw-tooth pattern that makes up the transition edge. Actually in this example, the ratio of sample-frequency to signal-frequency is 1000:1, so probably there are a hundred or so steps rather than the dozen shown above. As shown, the step spacing will be at the channel cycle rate. In this case, 10uS. The vertical steps are caused by residual charge from the conversion of $\text{ADIN}_{[1]}$ left on Cmux and Csamp thus creating a very small undesirable offset in $\text{Cext}_{[2]}$ during the conversion of $\text{ADIN}_{[2]}$. The downward steps are caused by the source for $\text{ADIN}_{[2]}$ attempting to recover the error via the source impedance of $\text{ADIN}_{[2]}$. The difference between these two step-amounts is the error which accumulates on $\text{Cext}_{[2]}$ with time. As the error voltage across Rsource accumulates with each cycle, the error step becomes smaller and smaller until the vertical upward step and the downward step cancel each other.

On the oscilloscope, when zoomed out enough to see the 100 Hz waveform, the 100 KHz sampling artifacts of the crosstalk are completely invisible, and the basic exponential shape (in the above figure) looks like a clean squarewave. In the general case, crosstalk will look like a vertically scaled image of $\text{ADIN}_{[X-1]}$ superimposed on $\text{ADIN}_{[X]}$.

6 The Effect of Cext

Now, look at Block 3 and the rationale for selecting Cext, or for that matter, even having Cext. To do this, use SPICE to try several values of Cext and measure the time at which the voltage on Csamp settles to within ½ LSB of the exact value. For a 12-bit ADC at 3.3 V, this amounts to 403 μV. Plotting a curve of settling time versus Cext may tell us something about the nature of Cext.

Figure 7 is a graph plotted from 39 runs of SPICE assuming two channels of the ADC in continuous conversion mode. A mid-range value of 2000 Ω was chosen for Rsource.

The left-most point on the graph corresponds to Cext = 0 (but Cpad = 4 pF). As Cext is increased, it can be seen that the required settling time gets worse until Cext is around 200 nF. Then there is a sharp roll-off in settling time until Cext is around 622 nF at which point the slope of the graph settles to near zero for further increases in Cext. This graph clearly illustrates that there is an optimum range for Cext. It is shown in Section 8 how this optimum range is derived, but for now we need to understand the shape of the curve in the graph below calculated for 12-bits.

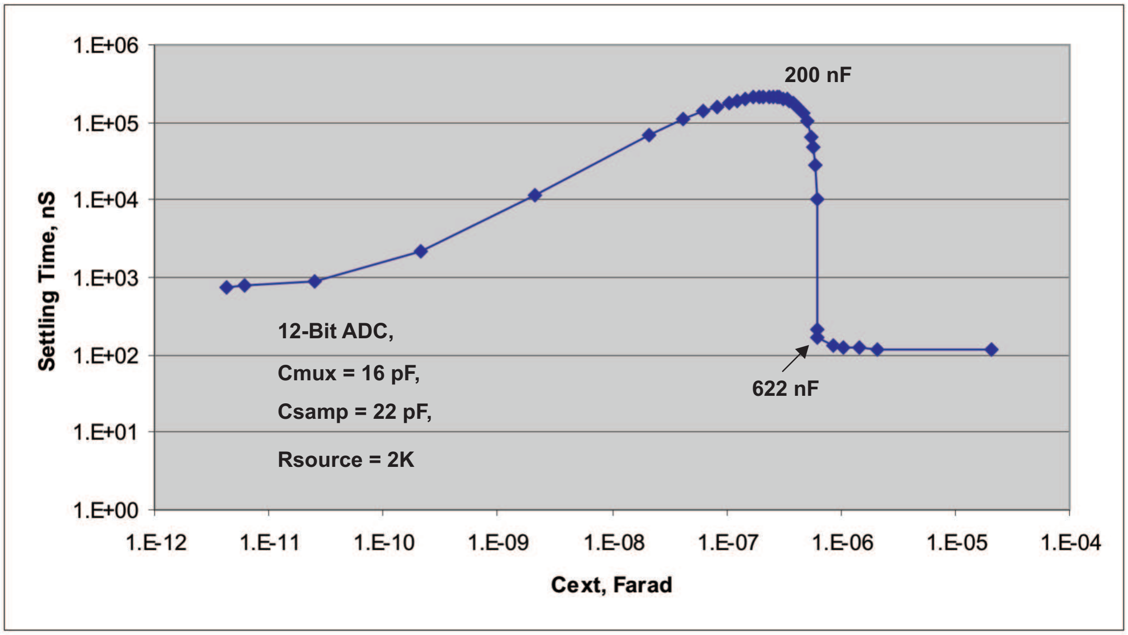

Figure 7. External Cap Value vs Settling Time

For a 10-bit ADC, the curve will be similar but Cext values will be 4 times smaller, assuming the same value for Csamp.

Figure 8 shows a hand-drawn time-domain plot of four SPICE runs on a 12-bit converter similar to the one made for Figure 7. These may help demonstrate the reason for the sudden drop in settling time as Cext increases; however, the waveforms have such hugh scale differences that not even a log-log graph does an adequate job of placing them on the same plot, so there is some graphic license taken in Figure 8.

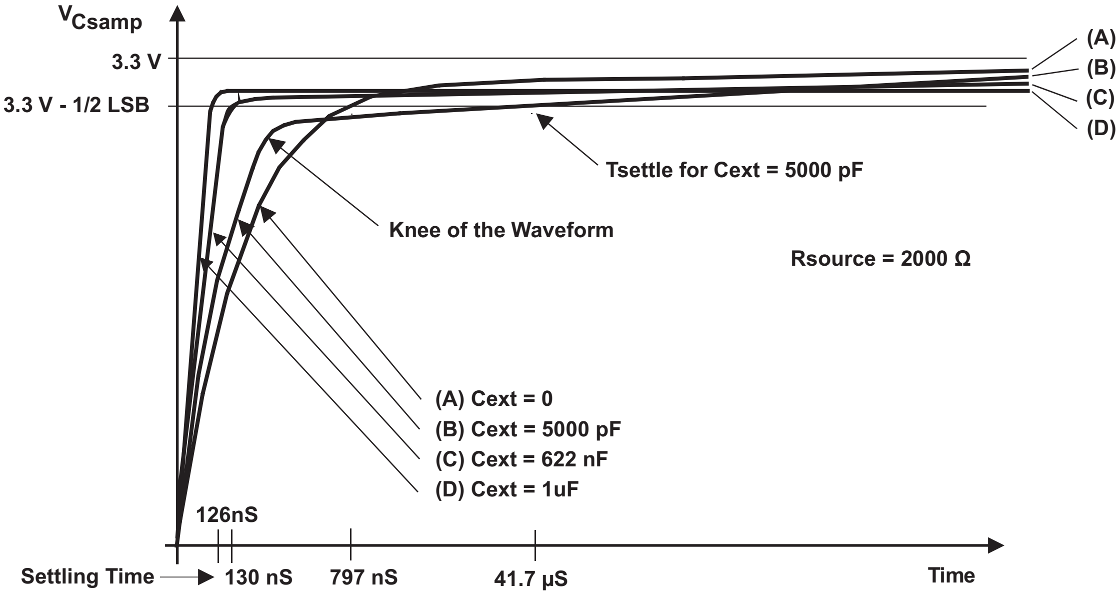

Figure 8. Four SPICE Runs on a 12-Bit Converter

The object is to get Csamp charged to within ½ LSB of 3.3 V before declaring Csamp settled. Note that as Cext becomes much larger than Csamp, the curve looks more like two straight lines joined by a knee. As Cext gets larger, the knee gets sharper.

Below are the dependencies of each of the curves above:

(a) With Cext = 0, there is effectively no discernible knee. The curve is a simple RC approximated by (Rsource + Rmux + Rsamp) * (Cmux + Csamp).

(b, c, d) For the other three curves, charge sharing jumps the voltage up to the knee with an RC time constant dominated by (Rmux + Rsamp) * (Cmux + Csamp). Then from the knee on, the RC time constant is dominated by Rsource * Cext.

(c, d) As the knee rises above the [3.3 V - ½ LSB] line with increasing Cext, there is a rapid reduction in time required for Csamp to settle since being within ½ LSB of the 3.3 V line is the definition of settled.

The vertical line segment is dominated by charge-sharing between Cext and (Cmux + Csamp). If, after charge-sharing, the knee falls short of the [3.3 V - ½ LSB] line, then it can take a very long time to finish charging Cext and Csamp the rest of the way via Rsource. However, if the knee occurs at or above the line, then charging Csamp is already done, and all that remains is to complete the recharging of Cext. However, once Csamp is settled, the analog-to-digital (A/D) conversion can proceed. Remember we have a full Group Cycle time to recharge Cext that can be two orders of magnitude longer than a single channel's cycle time.

7 Recharging Cext

During the charge-sharing process between Cext and [Cmux + Csamp], if you charge Cext by ½ LSB away from ideal, then in theory, you would have to wait an infinite amount of time for Cext to recover to its ideal value before you can attempt to charge-share again.

Try something more practical by waiting until Cext has recovered only to within 1/4 LSB before you charge-share again. Figure 9 shows that each successive charge-share pushes the error up until it settles in a range between ½ LSB and 1 LSB. This causes an effective 1 LSB of offset error in the voltage seen by the ADC.

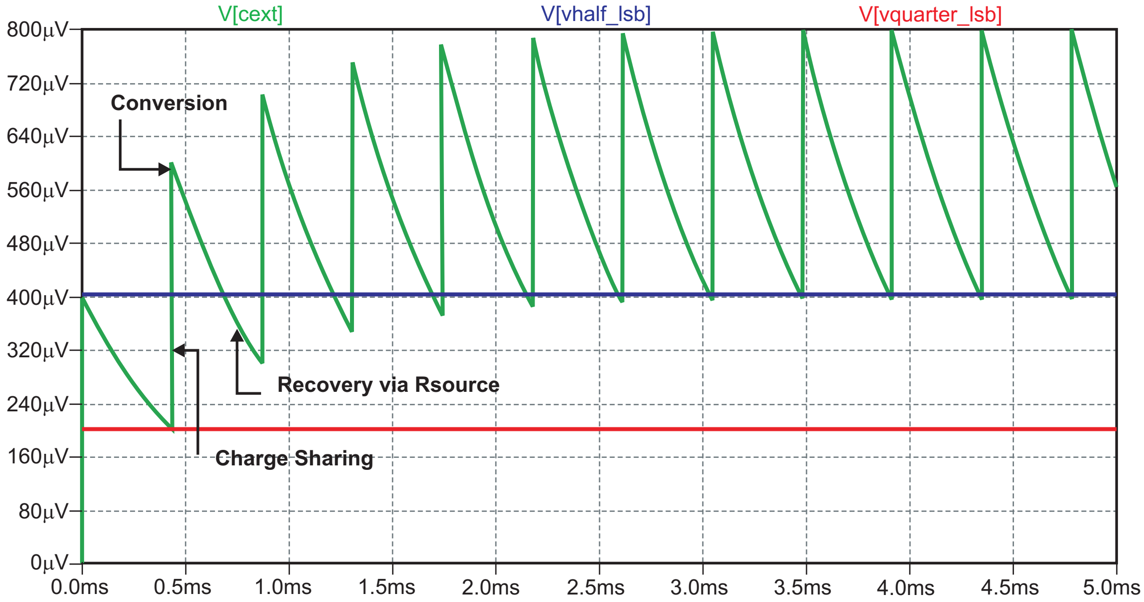

Figure 9. Recharging Cext

In the above simulation plot, the vertical edges are due to charge-sharing with [Cmux + Csamp] while the downward sloped edges are due to Cext recovering via Rsource. The assumption here is that the previous channel was at 3.3 V and the current channel is at 0 V.

The term error here refers to the deviation in Cext's voltage from ideal. The peak must never exceed ½ LSB to avoid crosstalk affecting the conversion result.

To solve this in such a way so as to improve offset error due to crosstalk, look what happens if Cext is only allowed to charge to 1/4 LSB during charge-sharing. This can be easily accomplished by doubling the size of Cext relative to the previous graph. Also, when Cext is recovered, recover it down to 1/8 LSB.

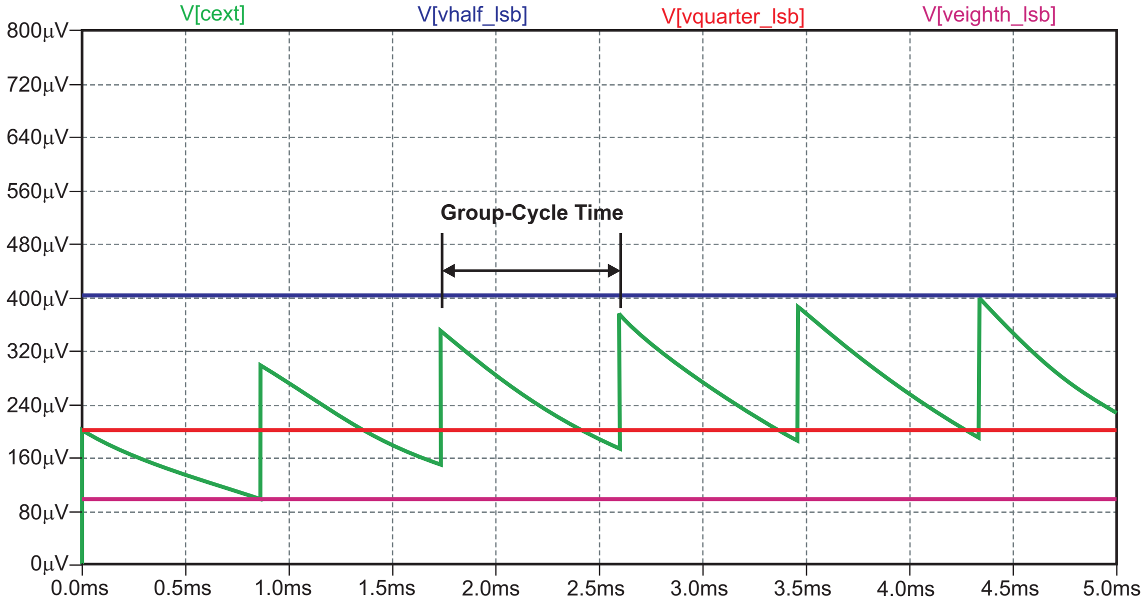

Figure 10. Cext Charging to 1/4 LSB during Charge Sharing

Now after several cycles, the time at which Cext begins charge-sharing with Csamp (i.e., the troughs) the error is only 1/4 LSB, and the maximum error on Cext never exceeds ½ LSB at the peaks. This causes an effective ½ LSB of offset error in the voltage seen by the ADC.

8 Calculating Cext

Look at why Cext should be greater than or equal to about 622 nF for a 12 bit converter having Cmux = 16 pF and Csamp = 20 pF. To do this, you need to examine the charge sharing between Cext and Csamp. Note that you will use Cmux + Csamp + tolerance (16 pF + (20 pF + 2 pF)) for charge sharing calculations. You can ignore Cpad as it is tiny and in parallel with Cext.

Develop the equation by recalling that conservation of charge says something like “the total charge before sharing is equal to the total charge after sharing”:

$$\text{before charge sharing,}\quad \text{Qsamp} = \text{Csamp} \times \text{Vsamp and Qext} = \text{Cext} \times \text{Vext}, \tag{1}$$

$$\text{conservation of charge, Qfinal} = \text{Qsamp} + \text{Qext}, \tag{2}$$

capacitors in parallel add,

$$\text{Ctotal} = \text{Csamp} + \text{Cext}, \tag{3}$$

after charge sharing,

$$\text{Qfinal} = \text{Ctotal} \times \text{Vfinal}, \tag{4}$$

substituting Equation 1, Equation 2 and Equation 3 into Equation 4,

$$(\text{Qext} + \text{Qsamp}) = (\text{Cext} + \text{Csamp}) \times \text{Vfinal} \tag{5}$$

$$\text{solving for Vfinal,}\qquad \text{Vfinal} = \frac{(\text{Qext} + \text{Qsamp})}{(\text{Cext} + \text{Csamp})}, \tag{6}$$

substituting Equation 1 into Equation 6,

$$\text{Vfinal} = \frac{(\text{Cext} \bullet \text{Vext} + \text{Csamp} \bullet \text{Vsamp})}{(\text{Cext} + \text{Csamp})}, \tag{7}$$

where Vfinal is the voltage remaining on Cext after charge sharing with Csamp.

It can be seen that for a 12-bit ADC settling to within ½ LSB, Vfinal would have to be:

$$\text{Vfinal} = \left(\text{Vin} - \frac{\text{Vin}}{2^{(12+1)}} \right) = 0.999878 \times \text{Vin} \tag{8}$$

where Vin is the desired input signal value.

It was concluded earlier that Csamp has to be settled to within 1/4 LSB to allow room to keep Cext's worst-case error below ½ LSB; therefore,

$$\text{Vfinal} = \left(\text{Vin} - \frac{\text{Vin}}{2^{(12+1+1)}} \right) = 0.999939 \times \text{Vin} \tag{9}$$

In a nominal 3.3 V system, this amounts to a worst-case value of 201 μV.

For a 10-bit ADC settling to within 1/4 LSB, Vfinal would have to be:

$$\text{Vfinal} = \left(\text{Vin} - \frac{\text{Vin}}{2^{(10+1+1)}} \right) = 0.999756 \times \text{Vin} \tag{10}$$

and in a nominal 3.3 V system, this amounts to 403 μV.

from Equation 7 and Equation 8,

Assume now that Csamp is discharged, Cext holds the value Vin, and Vfinal must end up at 0.999939 x Vin (that is, to within 1/4 LSB of Vin @ 12 bits):

$$\text{Vin} \times \left(1 - \frac{1}{2^{(12+2)}} \right) = \frac{(\text{Cext} \bullet \text{Vin} + \text{Csamp} \bullet 0)}{(\text{Cext} + \text{Csamp})} \tag{11}$$

$$(\text{Cext} + \text{Csamp}) \times \left(1 - \frac{1}{16384} \right) = \text{Cext} \tag{12}$$

$$\text{solving for Cext,} \quad \text{Cext} = 16383 \times \text{Csamp} \tag{13}$$

for a 12-bit converter with Cmux = 16 pF and Csamp = 20pF,

$$\text{Cext} = 16383 \times (16 + 20 + 2)\,\text{pF} = 622\,\text{nF} \tag{14}$$

for a 12-bit converter with Cmux = 16 pF and Csamp = 12pF,

$$\text{Cext} = 16383 \times (16 + 12 + 1)\,\text{pF} = 475\,\text{nF} \tag{15}$$

for a 10-bit converter with Cmux = 16 pF and Csamp = 20pF,

$$\text{Cext} = 4095 \times (16 + 20 + 1)\,\text{pF} = 156\,\text{nF} \tag{16}$$

for a 10-bit converter with Cmux = 16 pF and Csamp = 12pF,

$$\text{Cext} = 4095 \times (16 + 12 + 1)\,\text{pF} = 119\,\text{nF} \tag{17}$$

This is an absolute minimum value of Cext. Be sure when picking its value to include tolerance and aging factors. Larger values are fine but have practically no effect on sample time and only limited effect on group cycle time. This is discussed further in the following sections.

Assuming a 12-bit ADC, having Cext < 16383 * Csamp requires that Csamp be charged entirely during the Sample time rather than during the Group Cycle time. So Cext < 16383 * Csamp is essentially a different mode of operation from Cext > 16383 * Csamp.

From the standpoint of speed and Rsource requirements, you are actually better off with no external cap if Cext < 16383 * Csamp (assuming there is no anti-aliasing filter). This was shown dramatically in previous graphs.

9 Calculating Rsource

The total resistance feeding the external capacitor, Cext, is called Rsource. Namely, it is the Thevenin equivalent resistance of the driving source as viewed by Cext.

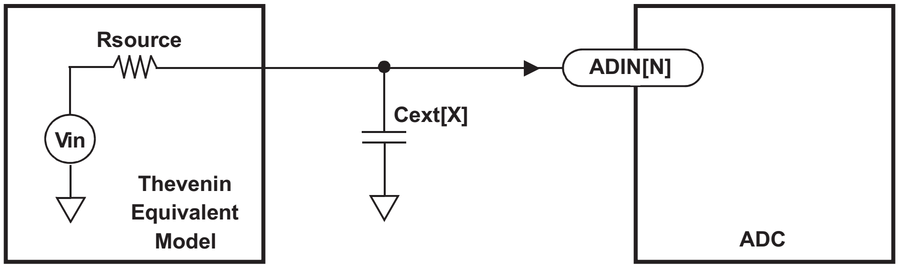

Figure 11. Thevenin Equivalent Model

The time constant required for an RC circuit to settle to within 1/4 LSB with 12 bits of resolution is:

$$\Upsilon = \ln\left(2^{(12+2)}\right) = 9.7 \quad \text{time constants} \tag{18}$$

For 10 bits:

$$\Upsilon = \ln\left(2^{(10+2)}\right) = 8.3 \quad \text{time constants} \tag{19}$$

Given a Group Cycle time, Tcyc, the value of Rsouce required to replenish the charge depleted from Cext is given by the relationship:

$$\text{Rsource} < \frac{\text{Tcyc}}{\Upsilon \times \text{Cext}} \tag{20}$$

But rather than do a lot of calculations, a table developed from a lot of SPICE runs is much easier, and besides, the calculation of ADC settling time gets messy due to the ADC having two closely spaced poles (see Section 3).

Table 1 shows settling times for different types of ADCs versus source impedance. To use this table, find your ADC in the first three columns based on data sheet specs. Secondly, if you use an external capacitor, Cext, make sure it is as big as the one listed in the fourth column after accounting for tolerance and aging. If your Cext is larger than the one listed, it will affect the result-columns favorably if at all.

Table 1. Settling Time vs. Source Impedance

ADC Channels	ADC Res, Bits	Datasheet Csamp, pF	Minimum Cext, nF	Rsource, Ω	Csamp Settling to ½ LSB, nS	Cext Recovery to 1/4 LSB, µS
16	10	20 ± 2	0	200	159	-
16	10	20 ± 2	0	2K	662	-
16	10	20 + 2	0	20K	5900	-
16	10	20 ± 2	156	200	112	21
16	10	20 ± 2	156	2K	112	213
16	10	20 ± 2	156	20K	112	2130
16	10	12 ± 1	0	200	113	-
16	10	12 ± 1	0	2K	508	-
16	10	12 ± 1	0	20K	4500	-
16	10	12 ± 1	119	200	77	17
16	10	12 ± 1	119	2K	77	165
16	10	12 ± 1	119	20K	77	1645
16	12	20 ± 2	0	200	187	-
16	12	20 ± 2	0	2K	797	-

Table 1. Settling Time vs. Source Impedance (continued)

ADC Channels	ADC Res, Bits	Datasheet Csamp, pF	Minimum Cext, nF	Rsource, Ω	Csamp Settling to ½ LSB, nS	Cext Recovery to 1/4 LSB, µS
16	12	20 ± 2	0	20K	7000	-
16	12	20 ± 2	622	200	130	90
16	12	20 ± 2	622	2K	130	900
16	12	20 ± 2	622	20K	130	9000
16	12	12 ± 1	0	200	134	-
16	12	12 ± 1	0	2K	600	-
16	12	12 ± 1	0	20K	5310	-
16	12	12 ± 1	475	200	89	68
16	12	12 ± 1	475	2K	89	682
16	12	12 ± 1	475	20K	89	6820

The settling time of Csamp becomes independent of source impedance if Cext is at least the minimum value previously calculated.

Next, looking at the settling time of Cext, the effects of varying Rsource are quite linear so you can easily interpolate using your timing requirements to obtain the required source resistance.

Look at an example of using Table 1. Keep in mind that the entire Group Cycle Time is used to recover Cext. Assume you have a 16-channel 10-bit ADC whose listed sample cap is 20 pF. You have a group of channels that must be converted every 15 µS. Using the data in the forth line of the table, if an Rsource of 200 Ω yields a 21 µS recovery time for Cext (i.e., Group Cycle time), then using linear interpolation, a 15 µs recovery time would require an Rsource of:

$$\text{Rsource} < \frac{200 \bullet 15}{21} = 143\Omega \tag{21}$$

This example shows that to speed up the recovery of Cext from 21 µS to 15 µS, it will be necessary to reduce source impedance from 200 Ω to 143 Ω or less.

Please note that if you, for example, double the size of Cext beyond the value in Table 1 , it is not necessary to recalculate anything, just interpolate as if you had used the exact value of Cext listed in the table. Either way, the same amount of charge is removed/recovered.

10 Consequences of Inadequate Settling Time

If Cext is at least as large as what is shown in Table 1, then the maximum time to settle Csamp is less than 130 nS, which is very small. There should never be a need to cheat on this parameter.

The only time it makes sense to omit Cext is when Rsource is very low, otherwise, the channel will be sensitive to noise. Generally, Rsource is low only with sensors that have outputs driven by opamps. In this case, Rsource is < 100 Ω and still there is no need to cheat on settling time.

Inadequate settling time leads to crosstalk. As discussed at the beginning, this is charge transferred from one channel to the next and accumulated on Cext over many conversion cycles. With crosstalk, each channel disturbs the next channel in the group to be converted. This phenomenon is due to the residue of charge on Csamp after converting channe$_{l[N]}$ which contaminates the charge on Cext$_{[X+1]}$ of channel$_{[X+1]}$.

A very good rule is don’t cheat on Settling Time.

11 Consequences of High-Source Impedance

If Rsource is too high for the desired conversion time, the only consequence is a time lag in the response of the channel. Full ½ LSB accuracy will occur, but it will be delayed according to Equation 18 and Equation 19.

If Cext is not present or if it is too small and you have cheated on Settling Time, then results will be inaccurate, there will be crosstalk, and full ½ LSB will never occur.

A very good rule is don't cheat on Settling Time.

12 Solutions

For those of you who insist on cheating, there is a software-selectable feature (included on newer Hercules MCU ADCs) that allows the application to discharge Csamp between conversions. The minimum time required to perform this discharging process is typically about 1 to 2 ADC clock cycles.

Using this feature is not a panacea. If you use it and cheat on settling time, it will provide a result that is scaled from the actual value. This scaled value will be predictable *only* if the Group Cycle time and channel-to-channel timing are well behaved. If the timing varies, the scaling factor will vary as well.

The only reason to use this discharge feature is to reduce the cost of using the correct value for Cext; however, it can be used successfully if ADC timing is well behaved.

13 Conclusions

Unbuffered multiplexed ratiometric ADCs are excellent in terms of cost and producibility, but careful consideration must be used in designing with them in order to obtain the expected results.

Most notable are the following points:

- Settling Time issues can be easily diagnosed by examining the waveforms at the ADIN$_{[X]}$ pins while the ADC is running in a continuous loop.
- In most cases, the best speed/impedance results will be obtained by including Cext if the proper value is selected.
- Given a specified number of bits of resolution, the proper value of Cext can be calculated independent of frequency and source impedance.
- Never cheat on Settling Time. It will only get you unpredictable results!
- If you do cheat, use the Discharge feature and continuous Group Conversion. Expect scaled results.

14 Terms in This Document

ADCLK	The internal on-chip clock of the ADC. The period of this clock is an integer multiple of the peripheral clock period, ICLK. This value is programmable from the ADC registers.
ICLK	The peripheral clock that drives the ADC.
Sample Time	The time during which the ADC's sample switch is closed for charging the internal sample capacitor, Csamp.
Settling Time	The time required to charge the sample capacitor, Csamp, to within ½ LSB. If Sample Time is greater than Settling Time, then the ADC is said to be settled.
Conversion Time	The time required for a single channel to be converted. It is the sum of the sample time plus one ADCLK clock cycle per bit of resolution (10 or 12).
Group Conversion	In the TMS470 and Hercules ARM Safety MCUs, all user-programmed channels in a group are converted sequentially and autonomously. Group conversions are set up and initiated by software. They may be programmed to run only once or continuously.
Group Cycle Time	Time measured from the start-of-conversion of channel[N] to the start of the next conversion of the same channel[N].
Channel Sample Frequency	This is the frequency at which a single channel is sampled and is equivalent to the reciprocal of the Group Cycle Time.
Offset Error	The translation from ideal of the ADC's transfer function due to an undesirable analog component in the input voltage.

15 References

- *Advanced CMOS Logic Data Book* (SCAD001)
- *Choosing an Anti-alias Filter*, Steve Hendrix, January 2001.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265