

AN2239

Author: Dennis Seguire

Associated Project: No

Associated Part Family: CY8C24/27/29xxx

[GET FREE SAMPLES HERE](#)

Software Version: NA

Associated Application Notes: AN2095, AN2219

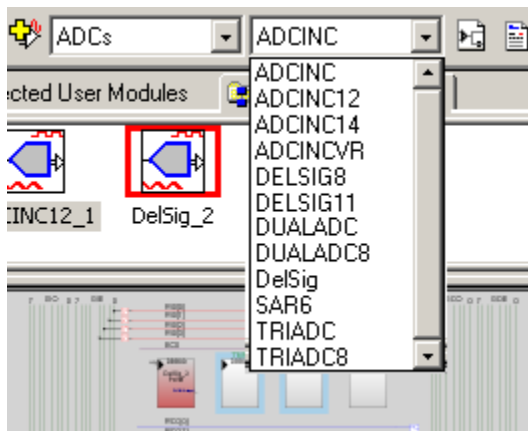
Application Note Abstract

This Application Note outlines features and performance of the wide array of PSoC® Analog-to-Digital Converters and provides guidance for selecting the most suitable converter for an application.

Introduction

Most PSoC applications utilize an Analog-to-Digital Converter (ADC). PSoC Designer presents dozens of ADCs with different resolutions, sampling rates, Signal-to-Noise Ratios (SNR), and resource utilization. This is evident in the ADC selection window of PSoC Designer Version 4.2, shown in Figure 1.

Figure 1. PSoC Designer v. 4.2 ADC Choices



This Application Note provides a brief review of the operation of each type of ADC and a guide for selecting which converter is most suitable to the PSoC-based system being designed. For an overall understanding, read the whole Application Note.

For an in-depth tutorial on ADC implementation and internal waveforms, review the ADC selection webcast of March 2004, available at www.cypress.com. For quick guidance to converter selection, jump to the Selection Guide on page ten of this Application Note.

ADC Basics

Conversion of an analog signal to a digital representation requires regularity and precision for useful data. We'll start with a few definitions.

Range

Range is the difference between the minimum and maximum values measured by the ADC. In PSoC, this is set by the references set in the Global Resource window of PSoC Designer.

Signals that are based on absolute voltages, such as power supply monitors or remote sensors, generally use a reference setting based on the PSoC's internal bandgap. This includes the reference $2 \cdot V_{BG} \pm V_{BG}$ (1.3 to 3.9V) for differential or analog ground referenced signals and $V_{BG} \pm V_{BG}$ (0 to 2.6V) for 3.3V systems with signals that are referenced to V_{SS} . Signals that come from sensors or systems where the sensitivity is determined by supply voltage generally use the reference setting of $V_{DD}/2 \pm V_{DD}/2$, so that when supply voltage changes, the ADC scale changes with it.

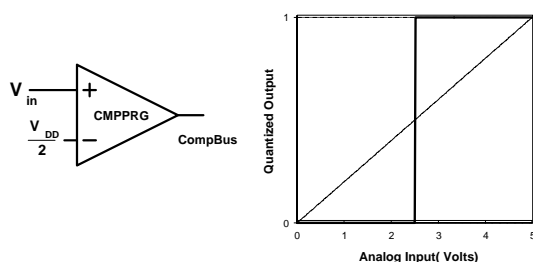
PSoC can also utilize user-supplied analog ground and reference signals. A more complete discussion of ground and references structures is contained in Application Note AN2219, "Selecting PSoC Ground and Reference."

Resolution

Resolution is the difference between successive measured levels, determined by the range of the ADC divided by the number of counts, typically $2^n - 1$. For example, a 10-bit ADC with scale from 0 to $V_{DD} = 5.0V$ has a resolution of $5.0 V / 1023 = 4.89 mV$ per bit. Some converters can be configured for resolution not equal to $2^n - 1$. These may be useful when precise sample rates are required.

To simply illustrate the low end of the resolution scale, a comparator, shown in Figure 2, is a 1-bit converter.

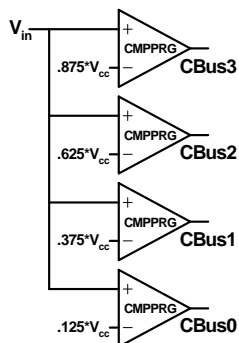
Figure 2. The Simplest 1-Bit ADC



The measured value is either above the threshold or it is not, a one-bit decision.

An ADC can be constructed from a number of comparators; this topology is commonly called a "flash" converter, not to be confused with Flash memory, and is shown in Figure 3. In PSoC, there is a maximum of four comparators, one for each column. This limits the resolution to one part in five, a 2.3-bit ADC.

Figure 3. Flash Converter Topology in PSoC



This converter is as fast as the comparator decision time, typically less than one microsecond. It can be extended to a very large number of comparators with sufficient chip real estate and power; an 8-bit comparator-based ADC requires 255 comparators plus priority encoder logic. This technique is used in ultra high-speed converters, typically for video applications. It is not an efficient use of PSoC resources, but it illustrates that very high speeds are possible with very dedicated resources.

Sample Rate

Sample rate is the number of times in a second that ADC data is obtained. When the ADC's data is gathered by triggering individual samples, the trigger rate must be less than the maximum sample rate. The sample rate is in "samples per second," not Hz. Sample rates vary widely depending on the topology of the converter, the resolution selected, and the power settings of the analog blocks used. The practical limit of a PSoC ADC is the ability to do something with the data. The PSoC's four MIPS will not allow a great deal of multiple channel parallel processing on high resolution (i.e., > 8 bit) data.

ADC Modulators

Incremental and delta sigma ADCs utilize a switched capacitor analog block configured as a combined integrator and comparator. One feature of this topology is the analog modulator that integrates the reference with a polarity based on the output state of the comparator. A second modulator block is added to some converters. This increases the effective integration gain and improves the SNR. This in turn allows a reduction in decimation rate and a corresponding increase in sample rate for a fixed resolution.

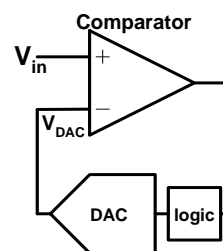
PSoC ADC Converter Types

Three basic types of ADCs are implemented in PSoC.

Successive Approximation Register (SAR)

The SAR is formed from a comparator and a Digital-to-Analog Converter (DAC), as shown in Figure 4.

Figure 4. SAR ADC Block Diagram

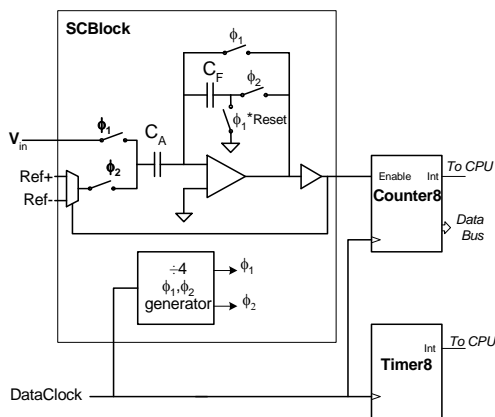


At the start of the conversion cycle, the DAC is set to half scale and a comparison is made between V_{IN} and the output of the DAC. With each step, the DAC is updated, the next bit selected, a subsequent comparison made, and the digital value found using a binary search (or "successive approximation") technique. This type of converter is fast but it uses 100% of the PSoC CPU during the conversion and requires the input to be stable throughout the complete conversion. In dedicated SAR ADCs, this is done with a sample and hold circuit, which is not available and not readily implemented in PSoC.

Incremental

The incremental converters utilize an integrator and comparator and a pair of references, as shown in Figure 5. The input is integrated on one phase of the clock cycle. The reference is then integrated in the opposite direction in the second phase of the clock cycle. The reference selection, positive or negative, is driven by the comparator, always integrating back to zero. The number of cycles in which the comparator output is positive is counted to obtain the digital result. This is algorithmically equivalent to a dual slope ADC.

Figure 5. Incremental ADC Block Diagram



The incremental ADC makes 2^n comparisons to form an n -bit conversion. The converter integrates the signal over the time of the sample. This has the result of integrating out high frequency noise terms. All PSoC incremental converters are clocked at four times the sample rate; the counter and timer clocks must match the divide by four of the phase clock generator in the analog column. PSoC incremental converters are available in a number of flavors:

ADCINC12 The ADCINC12 is clocked at four times the sample rate and requires a 14-bit timer and 14-bit counter to implement the 12-bit conversion.

The first two bits of both the timer and counter balance the divide by four in the column clock divider. Two digital blocks implement the lower eight bits of the timer and the lower six bits of the counter. The upper six bits of the timer and counter are implemented in interrupt-driven software.

ADCINC14 The ADCINC14 utilizes two more timer blocks than the ADCINC12. This enables a 16-bit timer function to be implemented without interrupt-driven software, eliminating the possibility of interrupt latency errors.

ADCINCVR This is the most flexible PSoC ADC, but it also requires the most code space for a single converter. The sample rate can be finely adjusted using the CalcTime parameter. This allows integration over a full cycle of a periodic waveform, typically a 50- or 60-Hz power cycle.

ADCINC The ADCINC utilizes the decimator normally associated with delta sigma converters in lieu of a counter. This saves one digital block and limits placement to a single instance. This converter is not as easily adjustable with regard to sample rate.

DualADC The DualADC is an expanded ADCINCVR. Each of the two timing-synchronized converters uses a separate integrator/comparator block and a separate 8-bit counter for the lower six bits of the result. Thus, the DualADC requires one more digital block than the ADCINCVR. The upper bits of the conversion are implemented in a software counter. Much of the firmware in the DualADC API implements the bit manipulation required for the variable resolution feature.

DualADC8 The DualADC8 is a simplified case of the DualADC. By using a single-byte returned value for each of the two results, it saves considerable code space and some calculation time in conversion.

TriADC The TriADC is an expanded ADCINCVR. Each of the three timing-synchronized converters uses a separate integrator/comparator block and a separate 8-bit counter for the lower six bits of the result. Thus, the TriADC requires two more digital blocks than the ADCINCVR. The upper bits of the conversion are implemented in a software counter. Much of the firmware in the TriADC API implements the bit manipulation required for the variable resolution feature.

TriADC8 The TriADC8 is a simplified case of the TriADC. By using a single byte-returned value for each of the two results, it saves considerable code space and some calculation time in conversion.

The DualADC and TriADC are especially useful in applications where multiple results must be converted simultaneously and multiplexing delays would result in inaccurate results. An example of this application is the calculation of AC power in the PMoCBasic1 Reference Design.

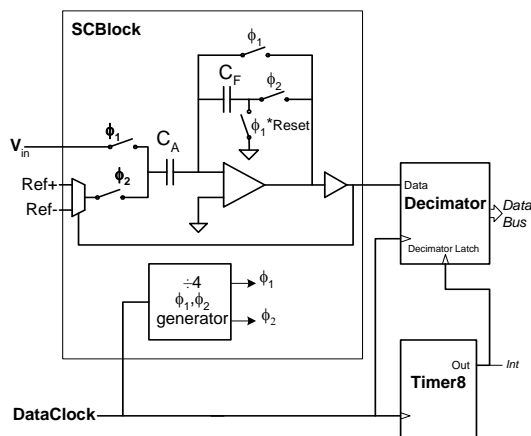
Delta Sigma

The delta sigma ADCs utilize the same integrator/comparator topology as the incremental converters. Rather than integrating the comparator output in a counter, the comparator result is processing in a decimator, as shown in Figure 6. The decimator double integrates the input at the sample rate. The output of the double integrator is sampled at the decimation rate (typically 1/64 times the sample rate) and subtracted from the last output value, yielding a differential. The differential process is repeated, still at the decimation rate, resulting in the transfer function:

$$H(z) = \frac{(1 - z^{-M})^2}{(1 - z^{-1})^2} \quad \text{Equation 1}$$

This is a sinc² filter. It has the advantage over the incremental converter of generating output data at the decimation rate times the sample rate instead of 2ⁿ times the sample rate. The output at any time includes data from two previous samples. When the input to the Delta Sigma converter is multiplexed, two samples must be processed through to get rid of old data before the current data is valid. This sets the minimum rate for scanned data.

Figure 6. Delta Sigma ADC Block Diagram



In CY8C25/26xxx and CY8C22/24/27xxx, the difference calculation is completed in software. CY8C29xxx has an improved decimator, which includes the difference function and reduces the calculation load.

DELSIG8 The DELSIG8 utilizes a single digital block for the timer function and decimates by 64. This ADC has an output rate of 31 ksp/s for a sample rate of 500 ksp/s (column clock equal to 2.0 MHz).

DELSIG11 The DELSIG11 utilizes the same hardware configuration as the DELSIG8, but decimates by 256. It has an output rate of 7.8 ksp/s for a sample rate of 500 ksp/s (column clock equal to 2.0 MHz). Faster output rates are possible but sacrifice linearity.

DelSig The DelSig converter provides a range of decimation rates with a different resolution for each. Topologies that decimate by 32 or 64 use a single digital block, topologies that decimate by 128 or 256 use two digital blocks. The digital blocks used for timing also implement a programmable PWM. The use of a double modulator improves the SNR and enables a resolution improvement of three bits at any given decimation rate. The improved decimator in the CY8C29xxx family substantially lowers CPU usage at all conversion rates.

ADC Tradeoffs

There are many reasons to select a specific ADC. Considerations, listed in the author's order of preference, include:

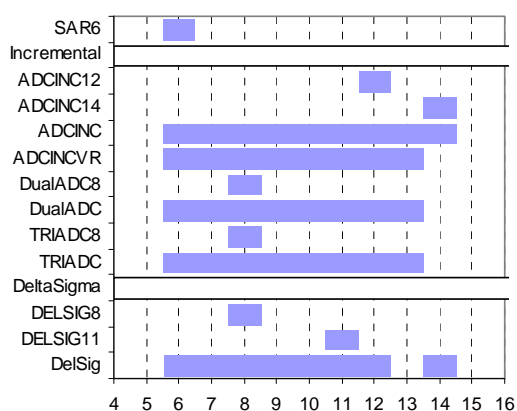
- Resolution
- Sample Rate
- Percent of CPU Usage
- Start Latency
- Block Count
- Power
- Interrupt Latency
- Gain Errors
- Linearity (INL, DNL)
- Noise
- RAM Consumption
- FLASH Consumption

Resolution

PSoC ADC User Modules range in resolution from 6 to 14 bits. A few ADCs, SAR6, ADCINC12, ADCINC14, DELSIG8 and DELSIG11, have specific resolution. Other converters have programmable resolution. At many resolution points, there are several selections, as shown in Figure 7.

An example of resolution choices is the 12-bit incremental converter, available as ADCINC12, ADCINCVR and ADCINC. All have the same sample rate, each has a significant advantage or disadvantage in applications. The ADCINC12 is subject to errors due to interrupt latency. The ADCINC runs at a fixed rate and uses the decimator in lieu of a digital block. It is the most resource efficient and eliminates the interrupt latency issue. The ADCINCVR consumes additional digital blocks but offers much adjustability in sample rate and eliminates the interrupt latency issue.

Figure 7. ADC Resolution Options

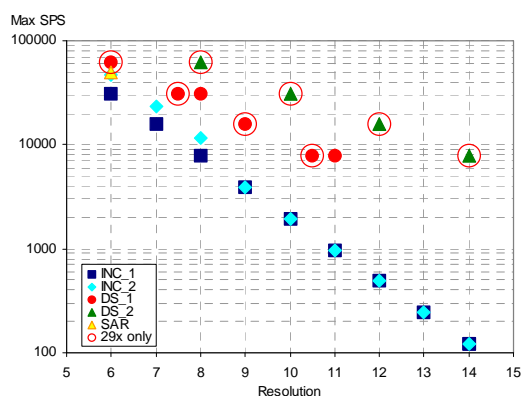


The user should make his selection of resolution based on system requirements; SNR is the key feature of accurate data. All of the incremental and delta sigma ADCs integrate noise and have lower aperture error than the SAR.

Sample Rate

The full range of PSoC ADC sample rate vs. resolution for a data clock frequency of 8.0 MHz is shown in Figure 6. The highest sample rates are attained using double modulator delta sigma converters. The sample rates shown in Figure 6 are maxima. PSoC ADCs are characterized with a data clock of 2.0 MHz and have better linearity at this clock rate than with a data clock of 8.0 MHz, but one-fourth the sample rate. The maximum sample rates for incremental converters are determined by the number of bits.

Figure 8. Sample Rate vs. Resolution Options



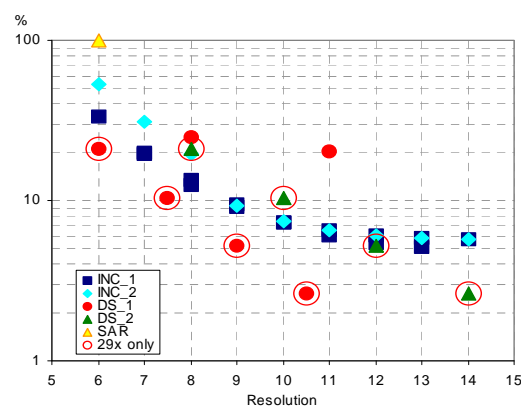
Delta sigma ADCs with double modulators can have significantly higher sample rates than single modulator implementations. While double modulators can yield improved SNR in incremental ADCs, they yield no improvement in sample rate, as it still takes 2^n samples to complete the conversion.

The ADCINCVR, DualADC and TriADC sample rates can be adjusted (downwards) by setting the CalcTime parameter in the User Module. DelSig, DELSIG8 and DELSIG11 sample rates can be adjusted by changing the resolution, setting the timer maximum count to a smaller number. An example of this is shown in AN2095, *mu-Law Voice Compression*.

Percent of CPU Usage

All ADCs require CPU time during data acquisition and for post-acquisition processing. The SAR6's logic stalls the CPU during the conversion. While it has a very short conversion time, it has 100% CPU usage during that time. The incremental converters use a software counter for the upper bits, triggered by the digital block (counter) interrupt. After the 2^n bit count sequence, incremental converters must scale the data, convert it to two's complement or unipolar if required, and direct the output to the A and X registers. In the case of DualADC and TriADC, the output is held in RAM. This data shuffling takes a fixed amount of time so that as the resolution increases and sample time slows, the percentage of time required for the post-acquisition processing drops. This is shown in Figure 9.

Figure 9. CPU Usage vs. Resolution



Low-resolution incremental converters have a higher percentage of CPU usage because they can have higher conversion rates. DelSig converters in the CY8C29xxx are exceptionally CPU-efficient because the type 2 decimator eliminates much of the need for post-acquisition processing.

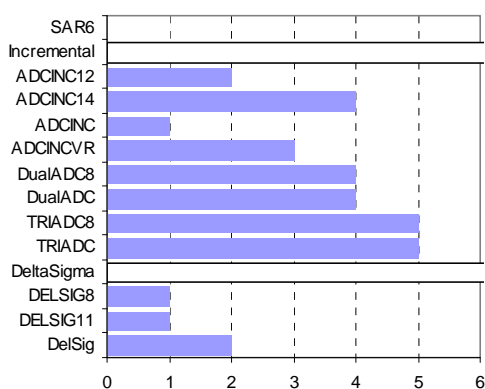
Start Latency

As mentioned earlier, the delta sigma ADC's output data includes the residue from two previous samples. When multiplexed, the first two samples after the input change will be inaccurate and should be ignored. This is usually done by adding a software sample counter that is used to delay usage of the data until the result is valid. Incremental converters, while considerably slower, do not have this problem. The ADCINC uses the decimator for the counting function but does not have the start latency since the decimator is reset at the start of the conversion, eliminating the residue from the previous sample.

Block Count

Incremental and delta sigma ADCs use a timer or PWM to establish timing and sample rate. The ADCINC12 uses a single timer block; the upper bits are implemented in interrupt-driven software. This allows for greater error than a full-length hardware timer, which is not based on interrupts. Higher resolution converters use 16-bit timers or PWMs. The ADCINC14 uses three blocks for the timer. The interrupt load from the upper byte of the counter function in these ADCs does not risk the same interrupt latency error of the ADCINC12. The digital block requirements for all converters are shown in Figure 10. The DelSig converter requires a second digital block for the timer (PWM) function in double modulator ADCs.

Figure 10. Digital Block Count



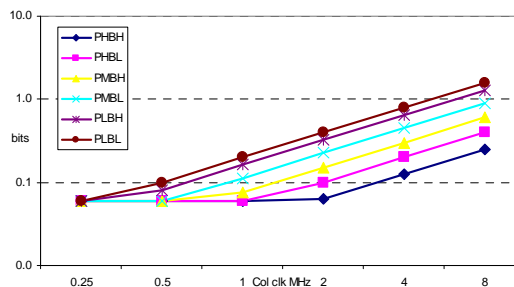
Single-modulator incremental and delta sigma ADCs require one switched capacitor analog block, double modulator ADCs obviously take two. Use of the double modulator improves the SNR. In the delta sigma ADCs, this SNR improvement enables higher resolution for a given decimation rate or faster conversion.

Power

The dominant power consumption in all PSoC ADCs is the analog blocks. Digital block and clock tree currents represent only a few percent of the analog power.

The analog block power setting must be sufficient to allow the opamp to slew and settle to the accuracy required for the selected conversion resolution; higher resolution requires longer settling. Settling time-induced error is a function of resolution and speed and follows the form of Figure 11. The graph in Figure 11 is a hypothetical representation and different for each type of converter and resolution, varying in scale and scope. At sample rate of 500 ksp/s (column clock = 2.0 MHz), setting power to HIGH and bias to Low is sufficient to keep settling time error below the level of the quantization noise for a 14-bit converter. This yields an ADS conversion rate of 122 samples per second.

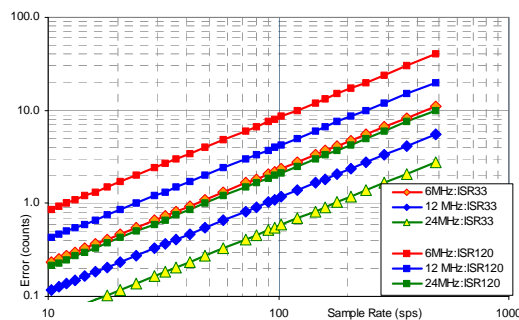
Figure 11. Bit Error vs. Clock and Power



Interrupt Latency

The ADCINC12 uses a 14-bit timer to count the number of times the input is sampled. The upper six bits of this timer are implemented in software. If the CPU is in the process of servicing an interrupt when the timer counts down to the end of the sample period, the timer continues to run and the counter continues to accumulate. When the timer interrupt is serviced and ADC output data is calculated, the counter may have counted past the correct value. Longer interrupts, higher sample rates and slower CPU clock speeds act together to increase the interrupt latency error, as shown in Figure 12. Again, this effect only exists in the ADCINC12.

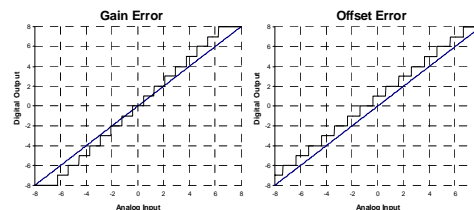
Figure 12. Interrupt Latency Error



Gain Errors

There are two types of gain error, shown in Figure 13. Signal error due to scale is caused by variations in the reference and the gain channel between the input and the ADC (for example, a PGA or INSAMP) and provides an error proportional to the signal level. Offset error is caused by a mis-match of input devices in input amplifiers and in the opamp used in the ADC's integrator/comparator.

Figure 13. Typical Gain Error



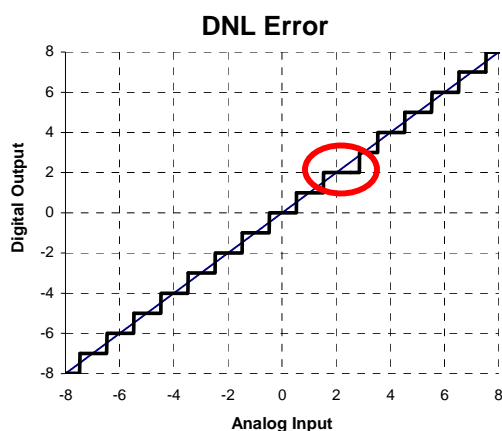
Both of these errors can be easily removed by calibration to known references.

Linearity

Linearity of ADCs is measured in terms of Differential Non-Linearity (DNL) and Integral Non-Linearity (INL).

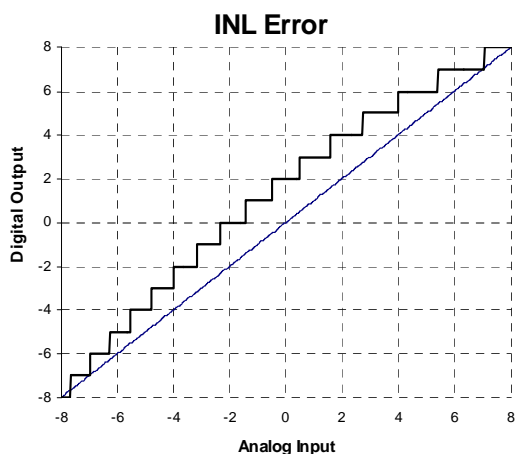
DNL is the deviation from a straight line from one bit to the next, as shown in Figure 14.

Figure 14. Differential Non-Linearity



INL is the summation of DNL errors and is characterized by a "bow" in the input/output curve, as shown in Figure 15.

Figure 15. Integral Non-Linearity

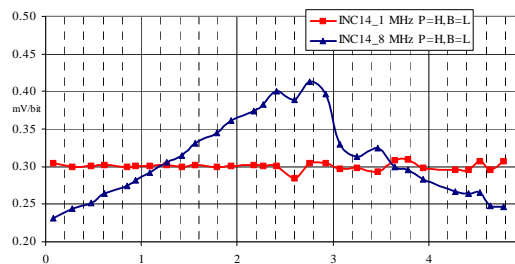


Both forms of non-linearity are functions of the topology and construction of the converter. They cannot be calibrated out as gain and offset errors can.

The accuracy of all converters is dependent on settling of the opamp in the integrator/ comparator. If the modulator is switched too fast and the opamp cannot keep up, substantial non-linearities can occur.

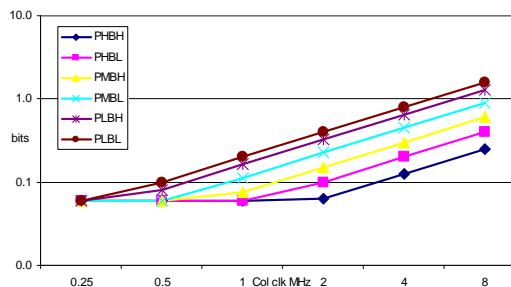
Figure 16 shows the typical linearity error of an ADCINC14 clocked at 2.0 MHz (500 kps, the frequency used for characterization) and then clocked at 8.0 MHz, four times the characterization frequency. This shows that the converter can be run this fast but it will suffer significant non-linearity.

Figure 16. ADCINC14 Non-Linearity for Sample Clock = 2.0 and 8.0 MHz



For lower-resolution converters, the settling time requirements are not as severe, so the converter can be clocked faster, or opamp power can be reduced, which slows down settling. For each converter at each resolution, the linearity error follows the shape of the curves of Figure 17.

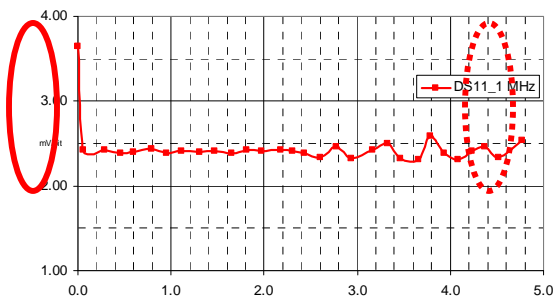
Figure 17. ADC Linearity vs. Power and Clock



PHBH means that the ADC User Module's power is set to HIGH and opamp bias is set to High in the Global Resources. Clearly, the ADC must be clocked at the proper (i.e., slow enough) rate to meet linearity specifications.

Delta sigma converters have increased non-linearity at specific output codes. This is an algorithmic function of the FIR filter used in the decimator. The non-linearity occurs most prominently at end-points of the scale, as shown in Figure 18.

Figure 18. Delta Sigma ADC Non-Linearity



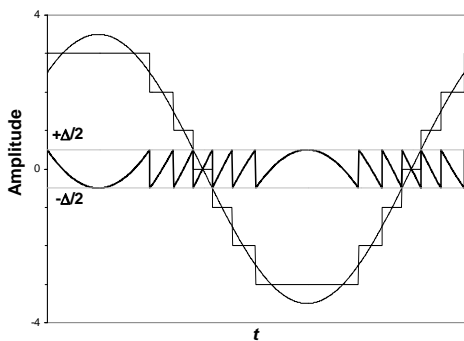
This non-linearity is NOT a circuit problem; it is a mathematical certainty of the FIR filter in the decimator. For most suitable applications of delta sigma converters (such as continuous data, as in audio), this non-linearity is not a problem. Systems that require linear measurement all the way to the rails should not use delta sigma converters.

Noise

Noise in ADCs comes from two sources, quantization (as a function of resolution) and the circuit itself. PSoC opamps have a noise level that is low enough for the circuit-induced noise to be at or below the quantization noise level for a 14-bit converter.

The quantization noise level represents the difference between the actual signal level and its digital representation, as shown in Figure 19.

Figure 19. Quantization Error



The quantization noise is easily calculated:

$$\epsilon_{RMS} = \sqrt{\frac{1}{\Delta} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \epsilon^2 d\epsilon} = \sqrt{\frac{1}{\Delta} \cdot \frac{\epsilon^3}{3} \Big|_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}}} = \frac{\Delta}{\sqrt{12}} \text{ Equation 2}$$

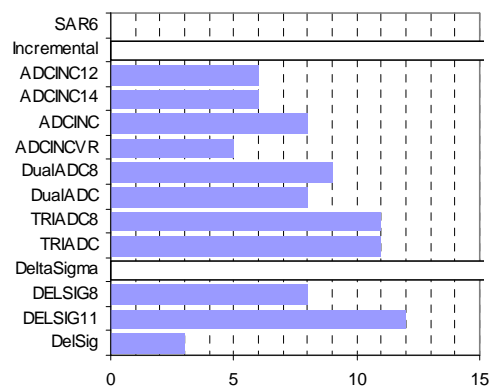
This comes out to 0.288 times the resolution. For example, a 12-bit converter with scale of 0 to 5.0V has a noise level of 0.35 Vrms.

Double modulators guarantee ADCs with noise levels well below the quantization noise limit.

RAM Consumption

All single-instance ADC User Modules return their data in the Accumulator for outputs of eight bits or less and in registers A and X for nine bits or more (MSB in X, LSB in A). Multiple-sample ADC User Modules store their data in variables within the API and return their data in X and A with individual call routines for each data point. Storage of this data in the API requires increased RAM for larger resolution, as shown in Figure 20.

Figure 20. ADC RAM Requirements

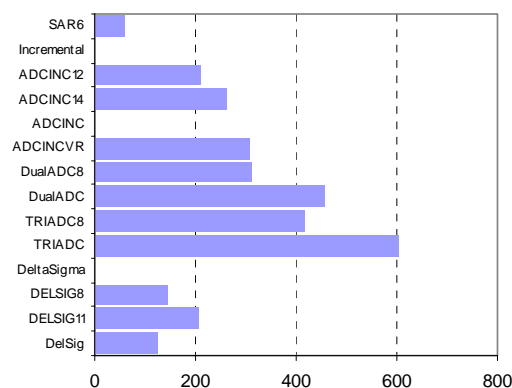


Note The SAR6 uses zero RAM, but 100% CPU capacity during the conversion.

FLASH Consumption

Flash consumption, shown in Figure 21, is usually the least important issue in selecting an ADC. While the SAR6 requires the minimum amount of Flash (and RAM), it disrupts CPU operation while acquiring data. Variable resolution ADCs include a good deal of code for adjusting the acquired value to the specified resolution. Note that the dedicated DualADC8 has 30 percent less code space than the more universal DualADC.

Figure 21. ADC Flash Requirements



Gotchas (Caveats)

- SAR input stability requirement.** If the input to the SAR changes during the conversion, the result can be corrupted. The SAR input bandwidth must be substantially less than its conversion rate.
- Only one decimator.** Since there is only one decimator, there can be only one instance of the ADCINC or any type of delta sigma converter. It is possible with dynamic reconfiguration to instantiate two delta sigma converters, but one of them will not work.
- Delta sigma not adapted to multiplexing.** The first two samples after the mux switch are lost, account for these and ignore them.
- ADCINC12 interrupt latency.** A system busy with interrupts has a high probability of corrupting ADC data.
- Delta sigma endpoint non-linearity.** These converters have substantial errors at the end-points and are not suitable for rail-to-rail or full reference-to-reference conversion range.
- Excess clock rate errors.** An excessively high column clock does not allow the integrator/comparator opamp to settle and results in large non-linearities across the full conversion range. Higher resolution converters are more sensitive to settling problems.

Selection Rules

- Use ADCINCVR.** It's initially a little tricky to set up, it takes more blocks, but it is clearly the most flexible.
- If fast data is required, use DelSig.** Make sure that routines calling for DelSig data are able to keep up with the sample rate.
- Use DualADC8 or TriADC8 in lieu of DualADC or TriADC if 8-bit resolution is sufficient.** This saves code space if internally controlled continuous sample rate is not required.
- Use ADCINC for resource minimization.**
- Use ADCINC12 only if it cannot be avoided.** There is nothing good about it that is not better in some other converter.
- Use SAR6 only if data is very slow.**
- If in doubt, use ADCINCVR.**

Summary

There is a flavor of PSoC ADC for every application. These converters are all well suited to the signal processing capabilities of the PSoC.

About the Author

Name: Dennis Seguire
Title: Senior Applications Manager
 Cypress MicroSystems

Background: BSEE, MSEE, University of California Berkeley.

More than 30 years experience in system and analog circuit design, holder of 5 patents for critical care medical devices.

Manager of Cypress MicroSystems applications engineering since 2000.

Contact: seg@cypress.com

In March of 2007, Cypress recataloged all of its Application Notes using a new documentation number and revision code. This new documentation number and revision code (001-xxxxx, beginning with rev. **), located in the footer of the document, will be used in all subsequent revisions.

PSoC is a registered trademark of Cypress Semiconductor Corp. "Programmable System-on-Chip," PSoC Designer, and PSoC Express are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
 198 Champion Court
 San Jose, CA 95134-1709
 Phone: 408-943-2600
 Fax: 408-943-4730
<http://www.cypress.com/>

© Cypress Semiconductor Corporation, 2005-2007. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.