Enhancing Analog to Digital Converter Resolution using Oversampling Technique

Priyesh Pandya, Vikas Gupta

Abstract— This paper is going to expose a method that gives us the possibility to use a low-resolution Successive Approximation Register (SAR) Analog-to-Digital Converter (ADC) in high resolution measurements. Oversampling and averaging can increase the resolution of a measurement without resorting to the cost and complexity of using expensive off-chip ADC's. This paper discusses about oversampling method for increase the resolution of a 12-bits ADC to 16-bits ADC. Oversampling method also rejects the noise by using averaging or moving averaging method. Oversampling method provides a software-based technique, resulting in an improved effective number of bits (ENOB) in the conversion result. It can be only used for measuring very low frequency or continuous signals, but the costs are lower compared to the price of the same high-resolution converter. This paper discusses how to increase the resolution of ADC measurements by oversampling and averaging. Additionally, more in-depth description on types of ADC, theory of oversampling technique and example code on utilizing oversampling and averaging.

Index Terms— Analog to Digital Converter, Oversampling, Decimation.

I. INTRODUCTION

Successive Approximation Register (SAR) type converters are used in PC plug-in data-acquisition boards or PC external data acquisition systems. They are by far the most popular ADCs in today measurement products. One reason for their popularity is their outstanding linearity which comes from the fact that they usually exploit a 1-bit quantizer. Even with a tri-level quantizer, linearity performance up to 20 bits has been reported.

The oversampling is a method to improve the resolution of a converter, by software methods, with a little help from outside. It is useful when we have a microcontroller with SAR ADC and we want to measure output signals from a sensor with good resolution. The principle of oversampling is to take a great number of conversions, and calculate a mean value. It is useless to take lots of equal measurements and to calculate the mean value of them, because you will obtain the same result: the value. The whole idea is based on the presence of a white noise without a continuous component.

Many applications require measurements using an analog-to digital converter (ADC). Such applications will have resolution requirements based in the signal's dynamic range, the smallest change in a parameter that must be measured, and the signal-to-noise ratio (SNR). For this reason, many systems employ a higher resolution off-chip ADC. However, there are techniques that can be used to achieve higher resolution measurements and SNR. In this paper, we proposed the method which can do both the tasks.

SAR ADC

The SAR ADC is a type of analog-to-digital converter that transforms a continuous waveform into a discrete digital representation via a binary search through all possible quantization levels before finally converging upon a digital output for each conversion (Figure 1).

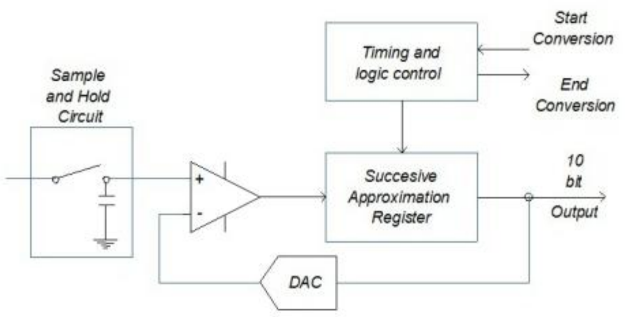

Figure 1: Internal diagram of the SAR ADC

The successive approximation analog to digital converter circuit typically consists of four main sub circuits.

A sample and hold circuit to acquire the input voltage (Vin).

An analog voltage comparator that compares Vin with the output of the internal digital-to-analog converter (DAC) and outputs the result of the comparison to the successive approximation register (SAR).

A SAR sub-circuit designed to supply an approximate digital code of Vin to the internal DAC.

An internal reference DAC that supplies the comparator with an analog voltage equivalent to the digital code output of the SAR for comparison with Vin.

Sampling frequency

The Nyquist Theorem states that a signal must be sampled at least twice as fast as the bandwidth of the signal to accurately reconstruct the waveform; otherwise, the high-frequency content will alias at a frequency inside the spectrum of interest (pass band). The minimum required sampling frequency, in accordance to the Nyquist Theorem, is the Nyquist frequency.

The Nyquist frequency:

$$\text{nyquist signal } f_{nyquist} > 2\cdot f_s$$

Where fs is the highest frequency of interest in the input signal. Sampling frequencies above fnyquist are called 'oversampling'.

This sampling frequency, however, is just a theoretical absolute minimum sampling frequency. In practice the user usually wishes the highest possible sampling frequency, to give the best possible representation of the measured signal, in time domain. One could say that in most cases the input signal is already oversampled.

The sampling frequency is a result of prescaling the CPU clock; a lower prescaling factor gives a higher ADC clock frequency.

Manuscript received April, 2014.

Priyesh Pandya, Department of Electronics and Communication, Technocrats Institute of Technology, R.G.P.V, Bhopal, M.P., India.
Vikas Gupta, Department of Electronics and Communication, Technocrats Institute of Technology, R.G.P.V, Bhopal, M.P., India.

At a certain point, a higher ADC clock will decrease the accuracy of the conversion as the Effective Number of Bits, ENOB, will decrease. All ADCs has bandwidth limitations.

Sources Noise of Data Converter

Noise in ADC conversions can be introduced from many sources. Examples include: thermal noise, shot noise, variations in voltage supply, variation in the reference voltage, phase noise due to sampling clock jitter, and noise due to quantization error. The noise caused by quantization error is commonly referred to as quantization noise. Noise power from these sources can vary. Many techniques that may be utilized to reduce noise, such as thoughtful board layout and bypass capacitance on the reference voltage signal trace.

However, ADC's will always have quantization noise, thus the best SNR of a data converter of a given number of bits is defined by the quantization noise with no oversampling.

Under the correct conditions, oversampling and averaging will reduce noise and improve the SNR. This will effectively increase the number of bits of a measurement's resolution.

II. METHODS TO INCREASE RESOLUTION OF AN ADC

Many applications measure a large dynamic range of values, yet require fine resolution to measure small changes in a parameter. For example, an ADC may measure a large temperature range, yet still require the system to respond to changes of less than one degree. Such a system could require a measurement resolution of 16 bits. Rather than resorting to an expensive, of f-chip 16-bit ADC, over sampling and averaging using Cygnal's on-chip, 12-bit ADC can measure a parameter with 16 bits of resolution. Some applications will use an ADC to analyze a signal with higher frequency components.

Such a system will also benefit from oversampling and averaging. The required sampling frequency in accordance with the Nyquist Theorem is the Nyquist Frequency: Sampling frequencies (fs) above fn is oversampling, and will increase the resolution of a measurement. .

A. Oversampling

For each bit of accuracy improvement, the signal must be oversampled by a factor of four, meaning that the relationship between the oversampling frequency fOS and sampling frequency fS is as shown in equation.

Oversampling Frequency $f_{os} = 4^x * f_s$

Where x is the desired improvement on the ENOB (for example, for two bits of improvement, x = 2).

Figure 2 shows the signal flow diagram of the oversampling method while figure 3 depicts how oversampling improves the accuracy of the conversion result. In this diagram, the input signal is oversampled by four (sample groups are shown in green and purple) and averaged. The sample points shown illustrate the difference between the raw, noisy signal and the average; the noise in this example affecting ± 3 bits of accuracy on an individual sample. Note that the averaged values (orange dots) are much closer to the ideal value than most of the single samples.

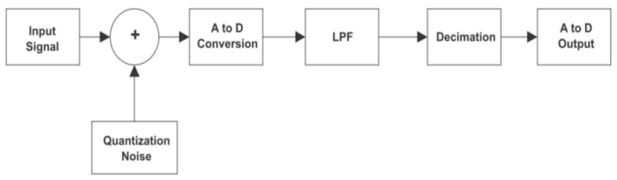

Figure 2: Signal-flow diagram for oversampling method

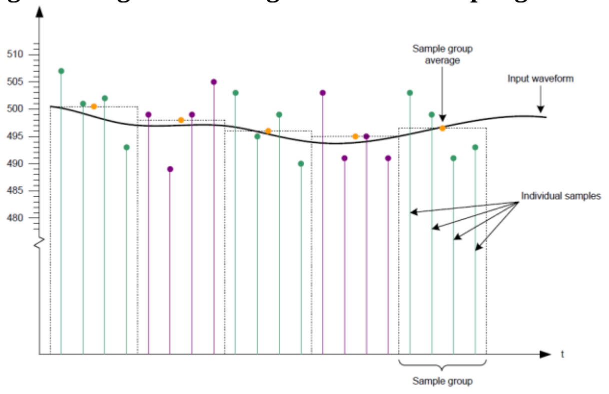

Figure 3: Averaged Conversion Results

B. Averaging

Averaging acts as a low-pass filter on the input signal, with the pass band of the filter narrowing as the sample size increases. When averaging conversion results, there are two approaches that can be taken: normal average or rolling average.

C. Normal Average

Taking n samples, adding them, and dividing the result by n is referred to as a normal average. When using normal averaging in an oversampling scenario, after the technique is applied, the sample data used in the calculation is discarded. This process is repeated every time the application needs a new conversion result.

In an application, the normal averaging approach is ideally used in cases where the sampling frequency is low compared to the sampling rate of the ADC.

Figure 4 shows a situation where normal averaging is used to oversample an input source by factor 4. For this example, the application requires that a new conversion value be ready (averaging complete) at each step of t (t0, t1, t2, and so on).

When using averaging techniques, there is a slight delay associated with the calculated conversion result since it corresponds to the average of the last n samples. The delay can be calculated using the formula shown in Equation.

Averaged Sample Delay

$t_{delay} = (t_{sn}-t_{s0})/2 + t_{process}$

Where tS0 is the time at which the first sample of the average occurs, and tSn is where the last sample occurs. The time required by the interrupt handler to process the sample data and calculate the average tprocess to supply to the application is also factored into the equation. In Figure 4, the delayed conversion result is highlighted in orange.

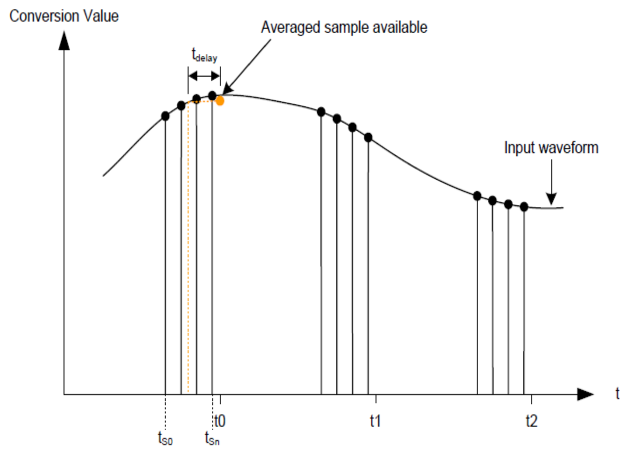

Figure 4: Normal Averaging

D. Rolling Average (Moving average)

A rolling average uses a sample buffer of the n most recent samples in the averaging calculation, allowing the ADC to sample at its maximum rate (the ADC sample rate is not reduced by n as in normal averaging), making it ideally suited for applications requiring oversampling and higher sample rates. The sample buffer can be prefilled with valid sample data (by taking n–1 samples prior to the first "real" data point), or can be left in an unknown state, depending on the application. The disadvantage of not prefilling the buffer is that the first n–1 samples contain invalid data and adversely affect the rolling average calculation. If acceptable by the application, buffer padding can be eliminated if the software can account for the possibility of the first n–1 samples being skewed.

Figure 5 shows an example of oversampling with a rolling average. The diagram shows a case where the input signal is oversampled by 4, meaning that the sample buffer uses the four most recent samples to calculate the average. In this example, the application requires a new sample at each step of t. Before the first oversampled result is calculated at t0, the sample buffer collects three samples so that the first data supplied to the application is valid.

When using a rolling average, the same sample delay calculated by previous equation applies. In Figure 5, the delayed values for t0, t1 and t2 (shown as d0, d1 and d2, respectively) are highlighted.

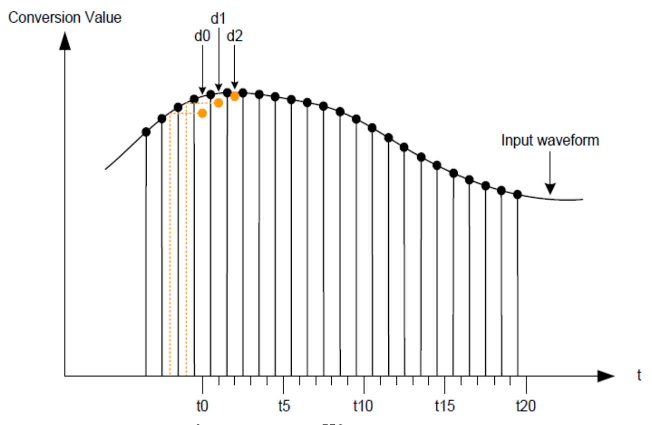

Figure 5: Rolling Average

III. IMPLEMENTATION

A. Hardware

Oversampling technique developed on Stellaris series evaluation kit. Evaluation Kit contains Microcontroller, LCD, analog input ranges from 0-20V and USB Mini-B cable for debug function. Figure 6 shows Evaluation Module used for developing oversampling technique.

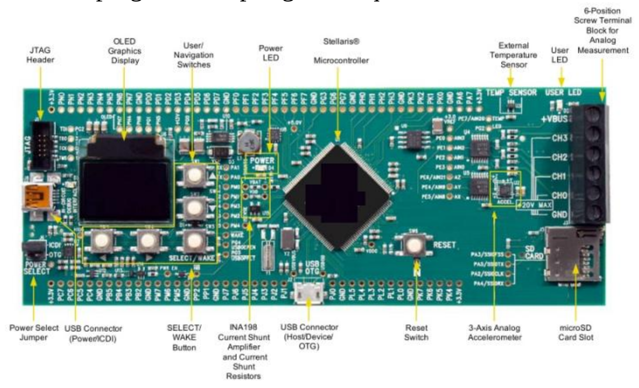
JTAG Header
OLED Graphics Display
User/ Navigation Switches
Power LED
Stellaris® Microcontroller
External Temperature Sensor
User LED
6-Position Screw Terminal Block for Analog Measurement
Power Select Jumper
USB Connector (Power/ICDI)
SELECT/ WAKE Button
INA198 Current Shunt Amplifier and Current Shunt Resistors
USB Connector (Host/Device/ OTG)
Reset Switch
3-Axis Analog Accelerometer
microSD Card Slot

Figure 6: Evaluation Module

B. Software

```
// Get the adc count data from the AD module
ADCSequenceDataGet(ADC0_BASE, 3, ulADC0_Value);

/ /Increment the count for Oversampling

count++;

/ /Accumulate the adc count data for Oversampling

ulADC0_OSDValue = ulADC0_OSDValue +
ulADC0_Value[0];

/ /Check for 16th sample

if(count >= 16)
{
    // Copy the accumulated data in buffer

  ulADC0_OSDValue = ulADC0_OSDValue;

    // Reset the counter for next cycle

  count = 0;

    ulADC0_Values = ulADC0_OSDValue;

//Range 0 to 20000 mV ,  Count range 0 to 65535
//20000/65535 = 0.301 mV resolution for one ADC count

    // Derive the Voltage value

  ulADC0_Values = (ulADC0_Values * 0.301);

// Display  first  digit

  font1 = ulADC0_Values % 10;
  display_string[5] = display_number[font1];
  ulADC0_Values = ulADC0_Values/10;

// Display  Second  digit

  font2 = ulADC0_Values % 10;
  display_string[4] =
display_number[font2];
```

```
    // Display  Third  digit

    ulADC0_Values = ulADC0_Values/10;
    font3 = ulADC0_Values % 10;
    display_string[3] = display_number[font3];

        // Display Fourth  digit

    ulADC0_Values = ulADC0_Values/10;
    font4 = ulADC0_Values % 10;
    display_string[1] = display_number[font4];

        // Display Fifth digit

    ulADC0_Values = ulADC0_Values/10;
    font5 = ulADC0_Values % 10;
    display_string[0] = display_number[font5];

}
```

IV. CONCLUSION

Oversampling technique is a combination of averaging and decimation. Adding extra samples and right-shifting the result by a factor n yield a result with resolution increased by n bits but also has the effect of averaging the quantization noise, which improves SNR. This will increase the ENOB and reduce the quantization error. With the availability of faster ADCs and with low memory cost, the advantages of oversampling are cost effective and desirable.

V. EXPERIMENT RESULTS

Oversampling technique developed on microcontroller evaluation kit. This evaluation kit has capability to take analog input on ADC channels ranging from 0-20VDC. Please find below test setup:

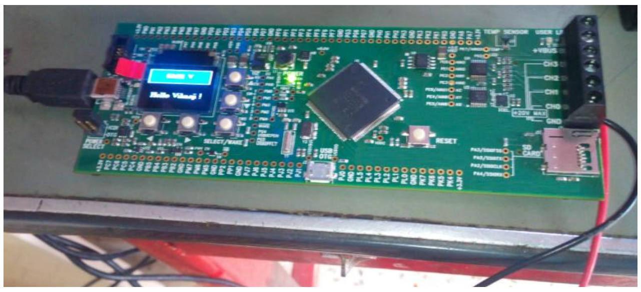
Figure 7: Test Setup with Evaluation kit

Below tables explains input voltage to ADC vs respective ADC counts. Table 1 shows ADC counts and voltage calculation without ADC gain calibration whereas Table 2 shows ADC counts and voltage calculation with gain calibration.

Table 1: 12bit and 16bit ADC count without calibration

ADC Input voltage	Original 12bit ADC Count	Without Calibration	
		Oversampled 16 bit ADC Count	Oversampled 16bit ADC counts to Voltage conversion
0.00V	2	36	0.01V
5.00V	1019	16318	4.98V
10.00V	1839	29435	8.98V
15.00V	2748	43973	13.42V
20.00V	3659	58548	17.87V

Table 2: 12bit and 16bit ADC count with calibration

ADC Input voltage	Original 12bit ADC Count	With Calibration	
		Oversampled 16 bit ADC Count	Oversampled 16bit ADC counts to Voltage conversion
0.00V	2.25	40	0.01V
5.00V	1019	18113	5.53V
10.00V	1839	32672	9.97V
15.00V	2748	48811	14.90V
20.00V	3659	64989	19.83V

This experiment shows that while increasing resolution of ADC from 12 bit to 16 bit its error gets multiplied by four. That's why it requires one time calibration to reduce error

REFERENCES:

1. Improving analog to digital converter's resolution using the oversampling technique by Flaviu Ilie BOB, Nicolae Cristian PAMPU, Liviu Teodor CHIRA.
2. Jayanath Murthy Madapura, Achieving Higher ADC Resolution using Oversampling, Microchip Technology Inc., 2008.
3. AVR121: Enhancing ADC resolution by oversampling, ATMEL 8-bit AVR Microcontrollers, Application Note.
4. Atmel AVR1629: XMEGA ADC Oversampling, Application Note.
5. AN1152: Achieving Higher ADC Resolution Using Oversampling, Application Note.
6. SPMA001A: ADC Oversampling Techniques for Stellaris Family Microcontrollers, Application Note.
7. SPMU272: Stellaris LM4F232 Evaluation Board, user manual.