A 0.8 V Asynchronous ADC for Energy Constrained Sensing Applications

M. Trakimas, S. Sonkusale

Tufts University

***Abstract*-This paper discusses the design of an asynchronous analog-to-digital converter targeted for low-power sensing applications. The asynchronous sampling scheme will save power because it only samples the input signal when it is changing. The idea of using an adaptive resolution to increase the maximum input frequency of the ADC is introduced. A prototype chip has been fabricated in a 0.18 µm CMOS process. Initial measurement results are presented.**

I. Introduction

In wireless sensor networks and portable medical devices, system power efficiency becomes increasingly important to increase the lifetime of the devices. Tsividis [1] suggested the use of continuous time digital signal processing as a viable alternative for aliasing-free high resolution low power signal processing. Such a system will also be energy efficient since their power consumption scales linearly with the input activity. For input signals exhibiting lower activity, synchronous sampling based digital signal processors are not power efficient due to its continuous sampling at a rate twice the highest frequency of the input signal, as stated by Shannon's sampling theory. When the input signal is changing at a rate lower than its maximum value, the synchronous system will continue to process the signal at the same rate burning unnecessary power during such idle periods.

A more efficient architecture uses an asynchronous sampling scheme where samples are taken only when the input signal changes [1]. The highest improvement in power efficiency for such architecture will be seen for signals which are relatively constant with brief periods of activity. This has been shown to be true in applications such as temperature sensors, pressure sensors, electro-cardiograms, and speech signals [2]. Another promising application is for brain implants which use electrodes to probe neurons. When a neuron fires, a voltage spike is seen at the electrode followed by little activity until the next time the neuron fires.

In order to realize an asynchronous system, analog-to-digital converters (ADCs) capable of processing signals asynchronously must be designed. In this paper we present the design of an asynchronous ADC which is targeted for use in energy constrained applications like implantable biomedical sensors. These applications are good candidates for an asynchronous architecture due to the nature of their signals and the critical importance of the system power consumption. In the following section we go into more detail about the asynchronous sampling scheme and its benefits. Section III presents the design of our ADC and introduces the concept of using programmable and adaptive resolution to improve the performance of asynchronous ADCs. We follow this with simulated and initial chip measurement results.

II. Asynchronous Sampling

The asynchronous sampling scheme can be explained using Fig. 1, which shows an asynchronously sampled signal (some examples include ECG or spiking neurons) The dashed lines in the figure represent 2^M-1 quantization levels which are evenly spaced across the full amplitude range, where M is the hardware resolution of the ADC. A sample is taken only when the signal crosses one of the quantization levels. As seen in the figure, the sampling rate adapts to the rate of change of the signal and drops to zero when the signal is not changing. This leads to a very efficient design which only takes samples if they will provide new information about the input signal. In order to preserve the shape of the signal, the time between samples, Δt, must also be kept. This value can be determined by either a local on chip timer or a time-to-digital converter which have been shown in [3] to add little to the overall power consumption of the ADC. The amplitude-time data pairs are output from the ADC and can be processed by an asynchronous DSP as demonstrated in [1]. The ADC can also be integrated with a synchronous DSP by using a synchronizer between the ADC and DSP. This will allow for increasing the power efficiency of ADCs in current systems without requiring a complete system redesign.

In order for asynchronous ADCs to be designed, the effect that the asynchronous sampling has on the system signal-to-noise ratio (SNR) must be analyzed. This is best explained by comparing the frequency responses of a sampled signal using a synchronous and an asynchronous sampling scheme. For a synchronous system, the signal harmonics will be aliased around the constant sampling rate back into the baseband. This effectively raises the noise floor and degrades the SNR which is limited to the theoretical maximum value of 6.02n + 1.76 dB, where n is the resolution of the ADC. The authors' of [1] have shown that in the case of an asynchronously sampled signal, the harmonics will no longer be aliased into the baseband since there is no longer a constant sampling rate. This theoretically eliminates the noise floor and as a result

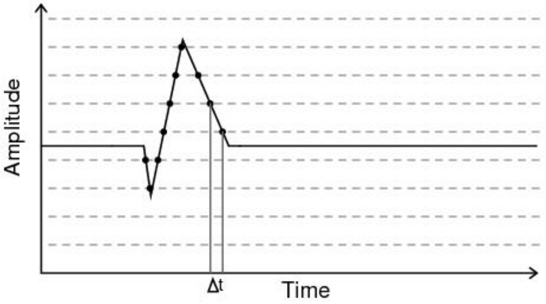

Fig. 1. Example of an asynchronously sampled signal.

only the signal harmonics within the baseband will affect the SNR. By setting the signal close to the system bandwidth, most of the harmonics will be out-of-band and the SNR can be significantly improved compared to the synchronous case.

In practical designs, there will still be quantization noise added to the signal due to the inexact time of the sampling instants resulting in jitter-like noise. This error in time, δt, will cause an error in the output voltage (1), where dV_{in}/dt is the slope of the input signal. Since the SNR of a system is dependant on the ratio of the power in V_{in} to the power in δV, the theoretic value for the asynchronous case can be calculated as (2), where T_C is the timer clock period which determines δt [2]. This shows that the SNR no longer depends on the bit resolution of the ADC but instead depends on the timer period T_C and the statistical properties of the input signal V_{in}. It has been shown in [3] that for speech signals, the ENOB of a 4-bit asynchronous ADC can be increased up to 10-bits by increasing the timer frequency to 1 MHz. In theory, the SNR will continue to increase as T_C increases, but in reality it is limited by the accuracy of the analog components in the ADC which determine the error in the quantization levels.

$$\delta V = \frac{dV_{in}}{dt} * \delta t \,, \qquad (1)$$

$$SNR_{dB} = 10\log\left(\frac{3P(V_{in})}{P\left(\frac{dV_{in}}{dt}\right)}\right) + 20\log\left(\frac{1}{T_C}\right), \qquad (2)$$

Another important parameter which must be calculated in the case of an asynchronous ADC is the maximum frequency which it can process. This value has been shown to be (3), where δ is the loop delay which is the minimum time the ADC requires between samples [2]. The loop delay is the inverse of the maximum sampling frequency of the ADC. Equation (3) shows that the maximum input frequency will be limited by the number of quantization levels since the ADC must process a sample every time the input signal crosses a level. In most applications, the input signal will never make a full-scale change at its maximum frequency and this requirement can be relaxed.

$$f_{\max} = \frac{1}{\delta * 2\pi\left(2^M - 1\right)}, \quad (3)$$

The asynchronous architecture has several other beneficial characteristics besides eliminating aliasing, improving quantization noise, and increasing power efficiency. The reduction in the sampling speed with low signal activity will reduce the electromagnetic interference caused by switching in the ADC. The ADC will also be immune to metastability since no sampling in the conventional sense takes place. The maximum sampling rate of the ADC can also be increased when compared to some ADCs such as a successive approximation (SAR) converter. The increase in speed is realized because the asynchronous ADC requires only one cycle per sample, while a SAR converter requires M cycles, where M is the resolution of the ADC [2].

III. Asynchronous ADC Design

Since the target application for our asynchronous ADC was low-frequency biomedical sensors, the main concern in our design was minimizing the power consumption. In order to do this we used a supply voltage of 0.8 V which biased the analog circuits in the moderate inversion region. This was a good compromise between the strong inversion region which allows for high bandwidths but wastes power, and the weak inversion region which offers the largest g_m/I_D ratio, but has absolute bandwidths too low for most applications.

A. ADC Architecture

The architecture of the designed asynchronous ADC is shown in Fig. 2. A fully-differential design was implemented in order to double the input signal swing which was limited by the low supply voltage. The operation of the ADC can be explained by first assuming that the digital logic has just taken a sample and has stored the digital value D_{OUT} which corresponds to the amplitude of the differential input signal. The logic has an internal timer which keeps track of the time between samples Δt and has been reset to zero at the time of the sample. After storing D_{OUT}, this value is sent to the two digital-to-analog converters (DACs) which output analog voltages equal to (4) and (5), where $\Delta V_{IN,diff}$ is the change in amplitude of the differential input signal since the last sample, and LSB is the least significant bit of the ADC. The input signal has been subtracted within the DAC to keep the output of the DAC < ±LSB. This will reduce the DAC settling time requirements and the loop delay of the ADC, allowing higher bandwidth signals to be processed. Assuming that the input signal amplitude has not changed while the DACs are being updated, *DAC1* and *DAC2* will initially output $+LSB$ and $-LSB$ respectively, and the outputs of both comparators will be low. When the differential input signal amplitude changes by ±LSB, one of the comparators will toggle high signaling the digital logic to increment or decrement the value of D_{OUT} by LSB, as well as to store the value of the timer. At this point the digital logic will update the DACs with the new value of D_{OUT} and reset the timer. This cycle will repeat as the ADC tracks the input signal. The values of D_{OUT} and Δt will be output at each sample for subsequent digital processing.

$$V_{DAC1} = +LSB - \Delta V_{IN,diff} \,, \; (4)$$

$$V_{DAC2} = -LSB - \Delta V_{IN,diff} \,, \; (5)$$

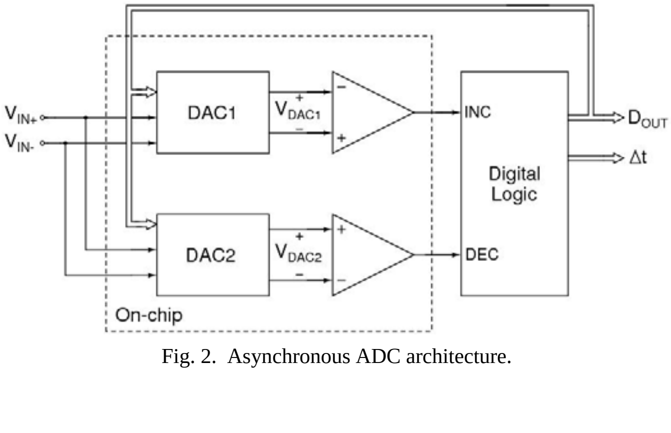

Fig. 2. Asynchronous ADC architecture.

B. DAC

The schematic of the 10-bit DAC designed for our ADC is shown in Fig. 3. A fully-differential operational transconductance amplifier (OTA) based hybrid charge-redistribution resistor string architecture was used for the design. The four least significant bits (LSBs) are resolved by the resistor string which is switched to the unit capacitor C_U closest to the OTA in Fig. 3. Both sides of the DAC share the same resistor string to save power. The next five bits are resolved by the capacitor array which includes 31 thermometer coded unit capacitors. These can be switched to either V_{DD} or GND depending on the most significant bit (MSB) of the DAC which determines the sign. The DAC has been over designed for 10 bits for added flexibility in the initial prototype. The unit capacitors C_U have been set to 150 pF in order to make the kT/C noise irrelevant. This sets the value of C_{FB} to 4.8 pF. The differential input signal is subtracted within the DAC using capacitor C_{IN} which has also been set to 4.8 pF for a gain of 1.

The operation of the DAC is similar to standard charge-redistribution DACs. It is reset by resetting the OTA and switching all the capacitors to V_{REF} which is $V_{DD}/2$. Once the DAC has settled to the required accuracy the OTA is taken out of reset and the capacitors are switched to either V_{DD} or GND and the first unit capacitor is switched to the resistor string. The number of capacitors switched and the location on the resistor string tapped depends on the value of the digital input. The difference between our DAC and a standard one is that the input signal is also switched to capacitor C_{IN} which adds its negative value to the DAC output. As discussed in the previous section, this will keep the DAC outputs within one LSB of V_{REF}, and will allow for a quicker settling time. An added benefit of this is that by keeping the outputs around $V_{DD}/2$, the leakage across the OTA switches is minimized. This reduces the voltage drift seen at the output of the DAC which can degrade the performance of an asynchronous ADC during extended periods of inactivity. The fully-differential architecture reduces the differential leakage seen at the output further and increasing the DAC performance. A second capacitor array and resistor string have been added in parallel with those shown in the schematic for compensating any offsets in the DAC.

C. OTA & Comparator

The schematic of the OTA designed for the DAC is shown in Fig. 4 with its common-mode feedback (CMFB) circuit omitted. The architecture used was a standard 2nd-order Miller compensated OTA. All the transistors were biased in the moderate inversion region to improve the power efficiency while still achieving a useful bandwidth. The CMFB circuit is essentially a replica of the input differential pair of the OTA and sets the bias point of the input stage pMOS loads. The OTA was designed with a bandwidth of 12 MHz for 5 pF loads on each output. The design of the comparator is shown in Fig. 5. It uses a Class-A amplifier structure followed by a pair of inverters to increase the gain and sharpen the transition between a ‘0’ and ‘1’ output.

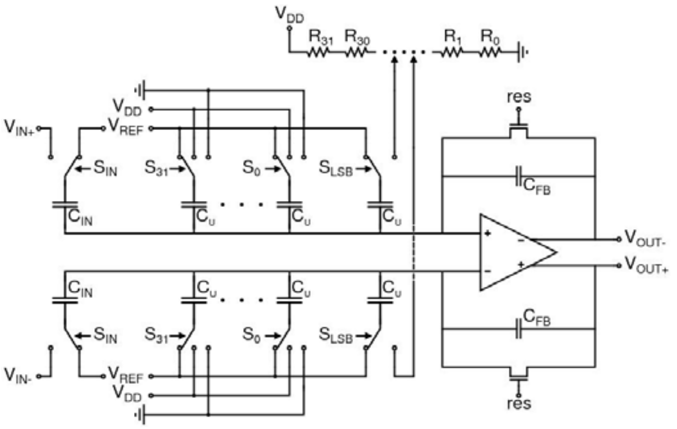

Fig. 3. Schematic of the hybrid charge-redistribution resistor string DAC.

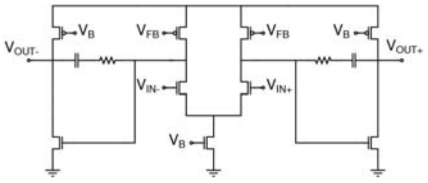

Fig. 4. Schematic of the OTA designed for the DAC (CMFB not shown).

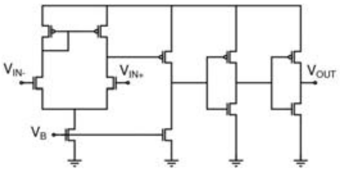

Fig. 5. Schematic of the continuous time comparator.

D. Programmable & Adaptive Resolution

As shown by (3), the maximum input frequency of the ADC is limited by the number of bits of resolution. Previous works have suggested that since the SNR of the ADC is theoretically independent of the bit resolution, the resolution can be set low while still achieving a high ENOB [2]. This is true for certain applications but will cause problems in applications where small amplitude signals must be processed. Take for example the case of an ADC processing the signals from a neural electrode. If the resolution is set too low then the ADC will miss some of the voltage spikes coming from the electrode. A tradeoff must be made between raising the resolution too high so that the ADC can not accurately track the high frequencies in the signal, and lowering it too much so that small pulses will be missed. Our designed ADC allows for programming the resolution anywhere from 2 to 8 bits which will allow the optimal resolution for the application to be used.

Another interesting concept which could be implemented in our ADC in the future is to have the resolution automatically adapt as the characteristics of the input signal change. This is possible since the resolution of the ADC is controlled by the digital logic. The resolution would initially be set high so that no small signals are missed. If the rate of change of the signal exceeds a set amount the resolution will decrease so that the ADC can quickly track the signal. The resolution will increase again as the signal slows down. This would effectively increase the maximum input frequency of the ADC. It also would not decrease the SNR of the ADC since it does not depend on the bit resolution.

IV. Results

A prototype chip of the asynchronous ADC was implemented in the IBM 0.18 µm 7RF CMOS process at the MOSIS fabrication facility. The digital logic was implemented off chip in an FPGA so that different control algorithms could be tested. The supply voltage of the DAC and comparators was set to 0.8 V to reduce the power consumption of the ADC. The digital logic which controlled the switches ran at 1.8 V in order to keep the on resistance of the switches low. The higher digital power supply will not affect the power efficiency of the ADC as the power consumption will be dominated by the analog circuits. A micrograph of the prototype chip is shown in Fig. 6. The die area for the analog section of the ADC was 0.96 mm^2.

Simulation results show that the ADC has a loop delay less than 1 µs, which leads to a maximum sampling rate of 1 MHz. This is achieved due to the low output swing required at the DAC which reduces its settling time requirement. Measurement results on the prototype chip confirm that the DAC reset and settling time will allow for a 1 MHz sampling rate. This will give a maximum input frequency of 10.5 kHz to 600 Hz for bit resolution from 4 to 8-bits.

The limiting factor in testing the chip has been the external FPGA which has limited the loop delay to 100 µs. This has only allowed us to test the prototype chip at low frequencies. A plot of the measured 6-bit ADC output for a 10 Hz 1.2 V peak-to-peak sinusoidal input is shown in Fig. 7. It can be seen that at the top of the sinusoid signal the sampling rate decreases. The ADC has been verified to function up to 8-bits. The maximum input of the ADC is 1.6 V since the input is subtracted within the DAC.

Initial ENOB measurements have been hampered by the large loop delay and the process used to capture the output spectrum. Despite this an ENOB of 7 has been measured for an 8-bit resolution. This number will increase as the FPGA problems are worked out and measurement techniques suited for asynchronous signals are used.

The static power consumption of the ADC has been measured to be 50 µW. The dynamic power could not be measured due to the digital logic being off chip. While the absolute value of the dynamic power could not be measured, the simulation of Fig. 8 shows that it will be much less than in a synchronous design for our targeted applications. Fig. 8 shows a plot of the analog power consumption for a train of pulse inputs which represent neuron voltage spikes. It is seen that dynamic power is only consumed when the signal is changing. This would not be the case for a synchronous design where we would see dynamic power consumed between the pulses.

V. Conclusion

A 0.8 V asynchronous ADC has been designed and implemented. Measurements show that the ADC will have a maximum sampling frequency of 1 MHz for a static power consumption of 50 µW. More testing will be required to fully characterize the ADC. Simulation results show that the ADC will save dynamic power in the case of signals with sparse activity. The designed ADC allows for an adaptive resolution control algorithm to be implemented in the future. This will increase the maximum input frequency that the ADC can process.

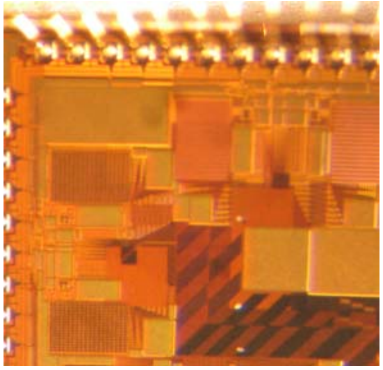

Fig. 6. Micrograph of the prototype chip (shows two copies of the ADC).

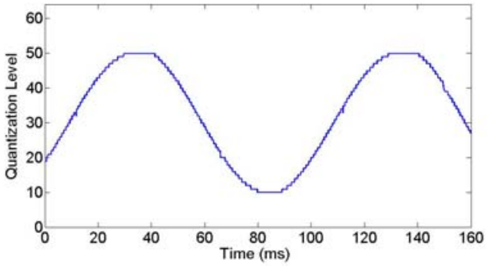

Fig. 7. Measured 6-bit ADC output for 10 Hz sinusoidal input.

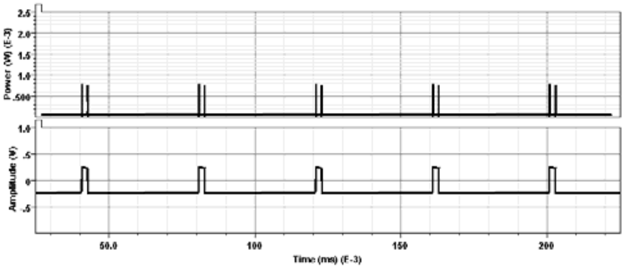

Fig. 8. Plot of ADC power in mW (top trace) for a train of pulses.

References

[1] Y. W. Li, K. L. Shepard, Y. P Tsividis, “Continuous-Time Digital Signal Processors,” IEEE International Symposium on Asynchronous Circuits and Systems, 2005.

[2] E. Allier, G. Sicard, L. Fesquet, M. Renaudin, “A New Class of Asynchronous A/D Converters Based on Time Quantization,” IEEE International Symposium on Asynchronous Circuits and Systems, 2003.

[3] E. Allier, J. Foulier, F. Sicard, A. Dessani, E. Ander, m. Renaudin, “A 120 nm Low Power Asynchronous ADC,” ISLPED, 2005.

[4] N. Sayiner, H. V. Sorensen, T. R. Viswanathan, “A Level-Crossing Sampling Scheme for A/D Conversion,” IEEE TCASII, April 1996.

[5] M. Renaudin, “Asynchronous Circuits and Systems: a Promising Design Alternative,” Microelectronic Engineering 54, 2000.

[6] L. Alacoque, M. Renaudin, S. Nicolle, “Irregular Sampling and Local Quantification Scheme A-D Converter,” Electronic Letters, Feb. 2003.

[7] F. Akopyan, R. Manohar, A. Apsel, “A Level-Crossing Flash Asynchronous Analog-to-Digital Converter,” IEEE International Symposium on Asynchronous Circuits and Systems, 2006.

FURTHER READING

Click any one of the following links to be taken to a website which contains
the following documents.

The following are some recent examples of Asynchronous ADC activity off the web.

6 bit Asynchronous December 2006
Asynchronous ADC In CAD Mentor Graphics
Asynchronous Data Processing System
ASYNCHRONOUS PARALLEL RESISTORLESS ADC
Flash Asynchronous Analog-to-Digital Converter
Novel Asynchronous ADC Architecture
LEVEL BASED SAMPLING FOR ENERGY CONSERVATION IN LARGE NETWORKS
A Level-Crossing Flash Asynchronous Analog-to-Digital Converter
Weight functions for signal reconstruction based on level crossings
Adaptive Rate Filtering Technique Based on the Level Crossing Sampling
Adaptive Level–Crossing Sampling Based DSP Systems
A 0.8 V Asynchronous ADC for Energy Constrained Sensing Applications
Spline-based signal reconstruction algorithm from multiple level crossing samples
A New Class of Asynchronous Analog-to-Digital Converters
Effects of time quantization and noise in level crossing sampling stabilization

Here is some more background information on Analog to Digital converters.

A 1-GS/s 6-bit 6.7-mW ADC
A Study of Folding and Interpolating ADC
Folding_ADCs_Tutorials
high speed ADC design
Investigation of a Parallel Resistorless ADC

Here are some patents on the subject.

4,291,299_Analog_to_digital_converter_using_timed
4,352,999_Zero_crossing_comparators_with_threshold
4,544,914_Asynchronously_controllable_successive_approximation
4,558,348_Digital_video_signal_processing_system_using
5,001,364_Threshold_crossing_detector
5,315,284_Asynchronous_digital_threshold_detector_
5,945,934_Tracking_analog_to_digital_converter
6,020,840_Method_and_apparatus_for_representing_waveform
6,492,929_Analogue_to_digital_converter_and_method
6,501,412_Analog_to_digital_converter_including_a_quantizers
6,667,707_Analog_to_digital_converter_with_asynchronous_ability
6,720,901_Interpolation_circuit_having_a_conversio2
6,850,180_SelfTimed_ADC
6,965,338_Cascade_A_D_converter
7,133,791_Two_mean_level_crossing_time_interval

11.19.10_1.20PM
dsauersanjose@aol.com
Don Sauer