Journal of
Electrical & Electronic Systems

Bengtsson, J Electr Electron Syst 2017, 6:1
DOI: 10.4172/2332-0796.1000219

Research Article OMICS International

Single-Chip Implementation of Level-Crossing ADC for ECG Sampling

Bengtsson L*

Department of Physics, University of Gothenburg, SE-41296 Gothenburg, Sweden

Abstract

This work demonstrates for the first time the implementation of a level-crossing analog-to-digital converter (LC-ADC) in a single, commercially available IC (that costs less than $2). The implementation utilizes adaptive threshold levels in order to prevent overload distortions for fast-changing signals. The entire design is based on a 20-pin PIC16F1769 microcontroller from Microchip and no external components are required. In fact, the only external circuitry required is a single jumper wire. This is due to the fact that the new generation of microcontrollers have integrated core-independent hardware, analog as well as digital. This design takes full advantage of the core-independent logic and analog blocks in a PIC16F17xx circuit to implement the LC-ADC technique that so far has required multiple-circuit designs or ASIC implementation. The design is demonstrated on a standard electrocardiogram (ECG) signal.

Keywords: Asynchronous sampling; Core-independent peripherals; Electrocardiogram; Level-crossing ADC; Microcontroller; Nyquist sampling; Overload distortions; Sparse signals; Synchronous ADC, Threshold levels

Introduction

Background

Traditional ADCs (Analog-to-Digital Converters) acquire samples at regular intervals T_S at a sample rate $f_S=1/T_S$ and needs to adhere to the Nyquist sampling theorem, i.e., $f_S>2\times f_B$, where f_B is the signal's bandwidth [1]. Due to the periodic nature of traditional ADCs, they are sometimes referred to as *synchronous* ADCs [2,3]. Synchronous ADCs are characterized by a periodicity in time and equidistant quantization levels as illustrated in Figure 1. Due to the fixed equidistant quantization levels, each sample will have an inherent uncertainty U, limited by the ADC resolution:

$$U_{\max}=\pm\frac{1}{2}\times\Delta U=\pm\frac{1}{2}\times\frac{U_{ref}}{2^N} \quad (1)$$

where U_{ref} is the ADC's reference voltage and N represents the ADC's number of resolution bits. This inherent uncertainty in the samples defines the limit of the SNR (Signal-to-Noise Ratio) of synchronous ADCs [4]:

$$\text{SNR}=6.02\times N+1.76\ \text{dB} \quad (2)$$

The disadvantage of a synchronous ADC is that it generates a great deal of samples that carry no information when "sparse and burst-like" signals are analyzed [5]. Sparse signals are, for example, radar and speech signals and electro cardiograms (ECG).

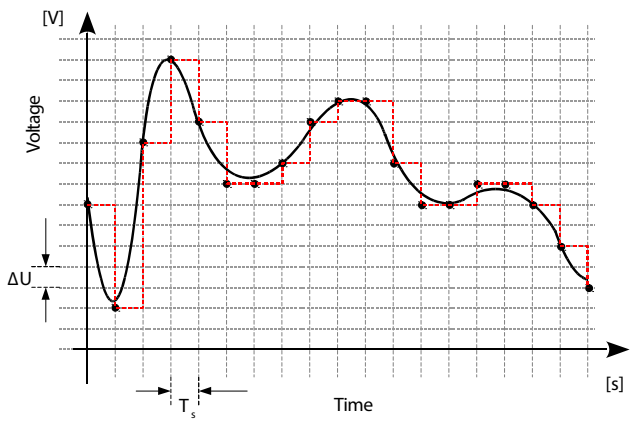

Figure 1: Synchronous ADC. Notice the periodic sampling interval and the inherent error in the sample values.

A sparse signal, like the one in Figure 2, has numerous regions with no, or very-low, activity, resulting in a corollary of identical samples which contain no net information.

In order to compress the data volume in sparse data sets, an *asynchronous* ADC may be used [6]. The asynchronous ADC is also referred to as the *level-crossing* ADC (LC-ADC) and was first suggested by Inose et al. in 1966 [7]. In an LC-ADC, the sampling is triggered by the signal activity rather than by a fixed time interval. Instead of periodically recording the signal's voltage level, the time between predefined level-crossings is recorded. Each sample becomes a 2-tuple:

$$u_n=D,T_n \quad (3)$$

where Tn is the time elapsed since the last sample and D is the "direction bit"; D indicates whether the upper or lower threshold was crossed. Figure 3 illustrates the same signal as in Figure 1 sampled with an LC-ADC.

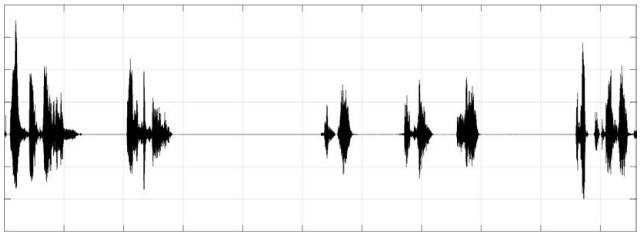

Figure 2: Speech signals are "sparse" and will generate a lot of dummy samples when sampled with traditional synchronous ADCs.

***Corresponding author:** Bengtsson L, Department of Physics, University of Gothenburg, SE-41296 Gothenburg, Sweden, Tel: +46317869128; E-mail: lars.bengtsson@physics.gu.se

Received March 03, 2016; **Accepted** March 24, 2017; **Published** March 29, 2017

Citation: Bengtsson L (2017) Single-Chip Implementation of Level-Crossing ADC for ECG Sampling. J Electr Electron Syst 6: 219. doi:10.4172/2332-0796.1000219

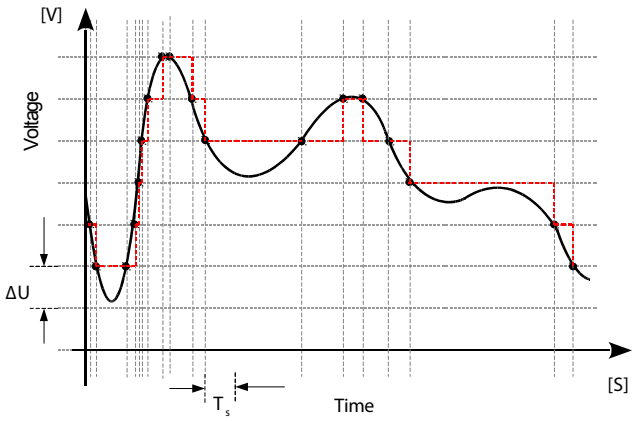

Figure 3: Asynchronous ADC. Notice the irregular sampling intervals and the lack of errors in the amplitude information of the samples.
Notice in Figure 3 how the sample density follows the signal derivative.

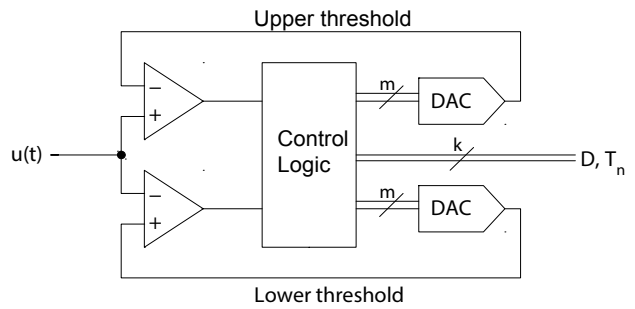

Figure 4: Basic LC-ADC design; an analog-digital hybrid.

Asynchronous sampling has several advantages over synchronous ADCs beyond the obvious inherent compressibility in sparse signal sampling. First of all, the data acquisition problem is transferred into one of quantifying time rather than voltage which is technically less complicated, less expensive and less power consuming. Second, high resolution time samples are easily accomplished; they depend on the reference clock frequency f_{clk} only. Third, the sample number reduction in sparse signals also suggests a reduced power consumption if the ADC host chip is retired to a low-power mode between samples. Finally fourth, the SNR is not limited to Equation (2); in an LC-ADC there is no uncertainty in the voltage levels. Instead, the SNR depends only on the "time resolution rate" R defined as [8]:

$$R = \frac{f_{clk}}{f_B} \tag{4}$$

(Reference clock's frequency/signal's bandwidth ratio). The SNR of an LC-ADC is [9]:

$$\text{SNR} = 20 \times \log R - 11.2 \text{ dB} \tag{5}$$

Hence for any given signal the SNR depends on the reference clock frequency only.

Basic design idea

Figure 4 illustrates the basic design on which most reported LC-ADCs are based [3,5,10-12]. The analog input signal $u(t)$ is compared to two reference levels (the upper and lower thresholds, respectively) in two analog comparators. If the signal crosses either one of the threshold levels, the control logic increases/decreases both levels in order to maintain the signal within the boundaries defined by the threshold levels. At every level-crossing a time stamp is generated that represents the data sample.

Related work

The level-crossing ADC technique was first suggested by Inose et al. [7] (but referred to as "asynchronous delta-modulation"). Mark and Todd picked it up in 1981 [6] with the pronounced objective of compressing data in sparse signals. In 2003, Allier et al. [3] implemented an "irregular sampling ADC" based on the LC-ADC technique. The main objective was to reduce power consumption in speech signal analysis and they implemented their design in a micro-pipelined architecture using STMicroelectronics 0.18 µm CMOS technology.

Implementation of "adaptive" LC-ADC algorithms was reported around 2010 [5,10]. In an adaptive LC-ADC the resolution of the threshold levels is reduced with increasing signal slope in order to reduce the overload distortion caused by fast-changing signals. Kózmin et al. [9] suggested a logarithmic distribution of threshold levels for ultrasound applications.

Tang et al. [13] suggested a "fixed window" design of the LC-ADC in 2013; instead of changing the threshold levels for every crossing, the thresholds stay fixed and instead the output from a DAC is subtracted from the signal. If the difference is outside the ± *Vth* thresholds the DAC input register is adjusted in order to keep the signal within the thresholds. This is illustrated in Figure 5. The objectives in Tangs et al. [13] work were to capture *in vivo* bio-potential signals. A similar design idea was suggested at the same time by Weltin-Wu and Tsividis [14]. The advantage of the fixed threshold window design is that it only requires one DAC and that the threshold levels are fixed. The disadvantage is that extra analog circuitry is required (for subtraction) and a negative reference voltage is required. Because of these adverse design issues, only the implementation scheme in Figure 4 was considered in this work.

The implementation of an LC-ADC is a synthesizing challenge. From Figure 4 it is obvious that the design has to be an analog/digital hybrid. Typical implementations have so far been separated into one digital part consisting of a microcontroller or an FPGA and an analog part consisting of external analog circuits [5,11,15]. Alternatively the entire design is implemented in an ASIC [3,12,13]. These solutions are either expensive or power consuming. This work will for the first time demonstrate a single-chip implementation of an LC-ADC in a low-power, commercially available integrated circuit that costs less than $2.

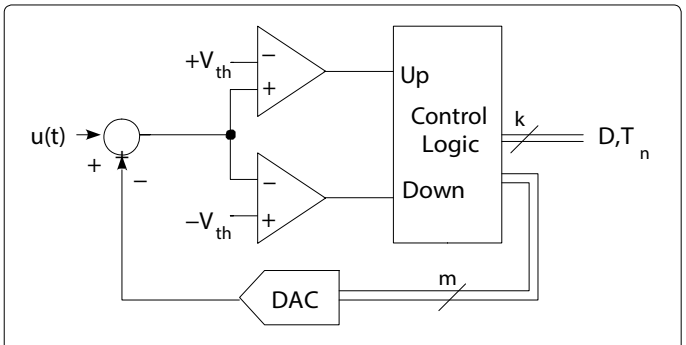

Figure 5: LC-ADC with fixed threshold window.

Method and Material

Hardware

The entire design is based on a PIC16F1769 microcontroller from Microchip [16] (with a list pricing of $1.92, October 2016). This controller has an 8-bit RISC architecture optimized for C programming. Apart from the "usual" microcontroller peripherals found in any commercially available controller (such as Timers, ADCs, UARTs, PWMs etc.) this circuit has "core-independent" peripherals that can run asynchronously and independently of the cpu. Also, these core-independent peripherals are both analog and digital. The digital blocks are 4-input/1-output configurable logic cells (CLCs) that can have one of eight predefined configurations with combinatorial and/or sequential digital electronics. The "intelligent analog peripherals" [16] comprise 10-bit ADCs, operational amplifiers, fast comparators, voltage reference generators and 5- or 10-bit DACs.

In order to implement the LC-ADC, two 10-bit DACs, two comparators and two CLCs are required from the core-independent peripherals. From the "standard" I/Os a 16-bit timer, a UART (Universal Asynchronous Receiver/Transmitter) and a Capture/Compare/PWM module (CCP) is used (for capturing). Figure 6 illustrates the implementation of the LC-ADC in a PIC16F1769 circuit.

If we compare Figure 6 and Figure 4 we can see that almost the entire design is implemented in core-independent hardware (once initiated they run autonomously with no cpu interference). CLC block 1 (CLC1) is configured to detect which threshold was crossed (upper or lower) by using an RS latch, and CLC block 2 (CLC2) is configured as a 2-input OR gate that triggers a capture event of Timer 1 whenever either threshold is crossed. Each CLC block has four inputs and only two are used in both CLC1 and CLC2; the remaining inputs are properly grounded or connected to V_{DD}.

Notice in Figure 6 that the only external hardware is limited to a single wire that connects pin 14 to pin 15; everything else is either configurable by registers or controlled by software. (The micro controller used here does not allow the negative comparator inputs to be connected internally).

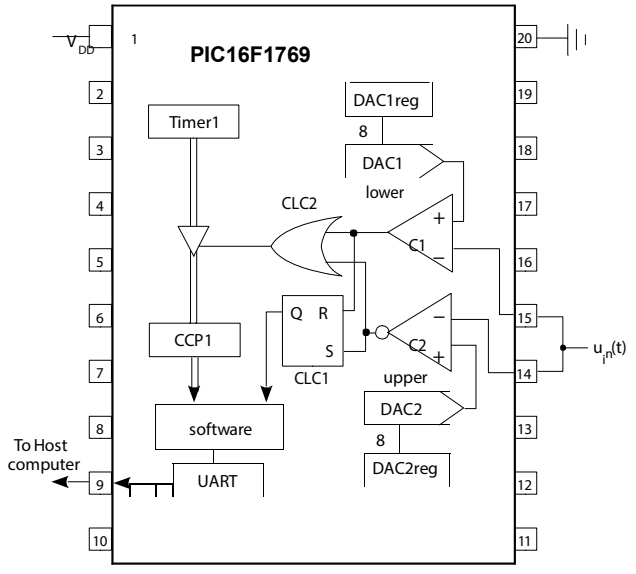

Figure 6: LC-ADC implementation in a PIC16F1769 20-pin circuit.

Software

The software is written in the C programming language and it is straight forward but depends on several subtle details. The following two major issues need to be considered:

- The available on-chip data memory is limited and only a limited number of samples can be stored on-chip; samples must eventually be transferred to a host computer. In the proposed design, the sample heap is off-loaded during the quiescent signal intervals.
- In order to prevent overload distortion caused by fast-changing signals, the distance between the threshold levels ("steps") must be dynamically adapted depending on the last sample value (i.e., the signal slope).

The work presented here was designed with the distinct objective of capturing transients (such as ECG signals) with an LC-ADC sampler. The hardware does nothing until the transient arrives and then samples are temporarily stored in on-chip RAM memory and transferred to the host computer only when the signal level has subsided below the lowest detectable level for some finite time; the data are transferred to the host computer during the quiescent intervals of the probed signal (the T-P interval in the ECG case).

The first part of the software is the main function which is responsible for off-loading the sample stack and transfers the samples to the host computer (Figure 7). The other part is the data sampler which is hosted in the interrupt service routine (isr) triggered by the capturing event. Samples are stored on the sample heap, the step size is determined depending on the size of the acquired sample and the threshold levels are adjusted accordingly. The ISR is illustrated in Figure 8.

The threshold levels are adapted to the signal's derivative as follows;

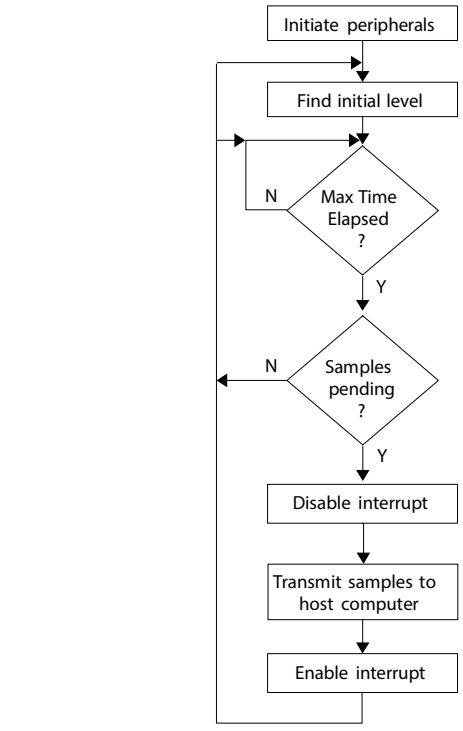

Figure 7: Main program flow chart.

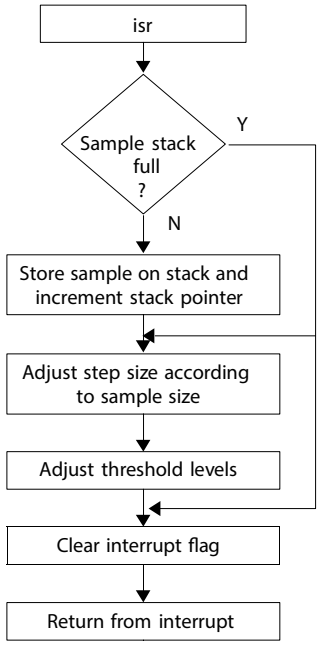

Figure 8: ISR flow chart. ISR is triggered by capturing events.

depending on the sample size (=the captured time value) the magnitude of the threshold levels are adjusted according to a 4-level logarithmic scale [9]. These levels were implemented as generic variables for easy adjustment.

The captured time samples are 16-bit integer numbers and the on-chip sample heap size was set to 100. This maximum heap size will depend on the particular controller used; the sample heap occupied 44% of the available data memory in the PIC16F1769 used in this work. Hence, a heap twice as large is possible to implement, but for this work 100 samples was sufficient.

The main program was designed to start off-loading the sample heap as soon as the signal line was "silent" or when the heap was full. In order to guarantee uncorrupted data transfers to the host computer, interrupt was disabled during data transfers.

Both DACs use 8-bit resolution (i.e., only the eight most significant bits are used). The entire software is hundreds of C code lines distributed over 23 source/header files (including *main.c*). However, by taking advantage of Microchip's Code Configurator (MCC), which is a graphical tool for initialization of peripherals, the designer's concern is the application code only. The application code is less than 100 lines of C code.

Setup/Data acquisition

In order to validate the design a Hewlett-Packard 33120A waveform generator was used to produce transient input signals to the signal input in Figure 6. The generated signal was captured on a digital oscilloscope and transferred via a thumb drive to the host computer and processed in MATLAB for presentation. The sampled data from the LC-ADC sampler was connected to a USB port on the host computer via an RS232-to-USB converter cable [17] and displayed in a standard terminal window (from which data could be easily imported to MATLAB). The baud rate used was 115,200 bits/sec.

Results

In order to adjust design parameters for optimal performance, some standard transients (with "decent" behavior) were initially generated. This made it possible to adjust threshold levels and step sizes in the software and the timer period in the hardware.

The first signal used was a triangular shaped transient; this is the simplest transient since it has constant slopes. Figure 9 illustrates an example of the generated signal and the signal reproduced by the samples produced by the LC-ADC. In Figure 9, the triangular waveform signal has a "frequency" of 15 Hz.

Next, the design was stressed by a transient with higher frequency and non-constant derivative; a sinusoidal signal with a "frequency" of 30 Hz was generated. Figure 10 illustrates the transient and the samples.

The experiments presented in Figures 9 and 10 made it possible to find the optimal design parameters (by a few trial-and-error adjustments) and finally a "realistic" ECG signal was injected into the system and the result is presented in Figure 11. The ECG signal in Figure 11 represents one heartbeat from a 1 Hz ECG (60 bpm) (from the HP33120A waveform generator) and the reproduced waveform from the LC-ADC samples.

Analysis

Figures 9-11 verifies the potential of the proposed design. Design parameters were optimized to accurately capture the vital information of an ECG signal (P, QRS and T waves [18]) and Figure 11 confirms that this is successfully achieved.

The time resolution in this design is 16 bits, but from Equation (3) it is clear that each sample will be 17 bits long; the time stamps are in the range ± 65,535. Hence, each sample consists of six ASCII characters that need to be transmitted by the UART interface. Also, a CR+LF character couple (Carriage Return, Line Feed) is appended

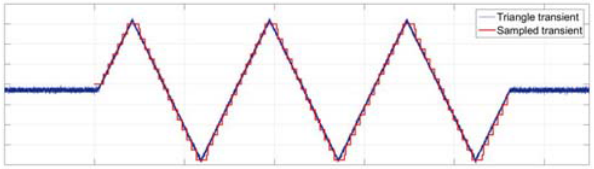

Figure 9: 3-Tuple triangular transient.

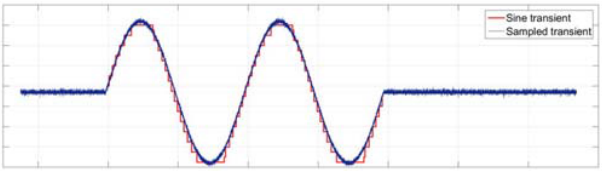

Figure 10: 2-Tuple sinusoidal transient.

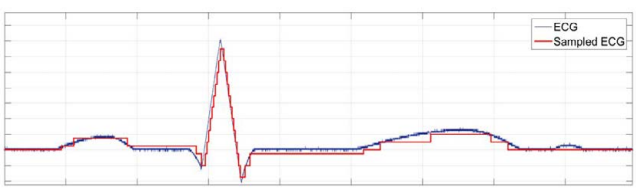

Figure 11: ECG signal (60 bpm rate).

to each sample in order to display the samples on separate rows on the receiving terminal. Every sample then consists of eight ASCII characters. The ECG signal in Figure 11 contains 34 samples that need to be transmitted to the host computer during the quiescent T-P [18] interval of the signal. In the RS-232 asynchronous serial interface, the transmission overhead is a single start bit and (at least) one stop bit (parity is optional). Hence, it will take 10 bits to transmit an ASCII character. The number of bits required to transmit the ECG signal in Figure 11 is therefore

10 bits × 8 ASCII characters × 34 samples=2,720 bits.

Using a bit rate of 115,200 baud means that it takes 2,720/115,200=2.36 ms to off-load the sample heap. At a heartbeat rate of 60 bpm (1 Hz), the quiescent time interval between the T wave and the next P wave is at least 200 ms, which means that there is plenty of time to transfer samples to the host computer; the off-loading of the sample heap occupies approximately 10% of the available "silent" time in a 60 bpm ECG signal.

In Figure 11, the average count number between samples is 3919.7 and the count rate (f_{clk}) was 2 MHz. This indicates an average sample rate of $2\times10^6/3919.7=510$ S/s.

Conclusions

Level-crossing ADCs have the potential of saving a great deal of data space in sparse-signal sampling and/or reducing the need for data transfers to a host computer [6]. They are straight-forward designs based on standard components. However, the analog/digital hybrid nature makes them complicated to implement on a printed circuit board and has so far required either a multi-circuit design using a combination of analog and digital circuits or an ASIC design. This work has proved that it can be implemented in a single, low-cost integrated circuit with an absolute minimum of external hardware (a single wire). In this work the objective was to sample an ECG signal of a specific rate but the design is not limited to this particular signal. The main design parameters are the size of the sample heap, the threshold levels, the time resolution and the time range. All of these parameters are configurable in software and can easily be adjusted for other signal types.

The "absolute limits" of this design have not (yet) been investigated since they depend on so many different parameter settings; they need to be determined for each signal to be sampled.

Discussion

Software structure

The flow charts in Figures 7 and 8 indicate some alternative ways to design the software (the dashed arrows). The software used assumes that the base line level (the quiescent signal interval) does not change. The software determines this level at the beginning of the program and then assumes that it stays constant for the rest of the measurement. If it does change, the small algorithm that finds the initial level will have to be included in the super loop of the main function; this is indicated by the dashed arrow in Figure 7). This design has also assumed a maximum number of samples in the transient; if the number of samples suddenly increases above that number, information is lost. If this is critical, the sample heap must be expanded or threshold levels must be adjusted in order to reduce the number of samples. However, if the sample heap is full, the software used here just ignores any additional samples. A compromise could be to keep tracking the signal (i.e., still adjusting the thresholds) without acquiring new samples. That would guarantee that the sampler is maintained at the right level when the sample heap is off-loaded and sampling resumes. This is indicated by the dashed arrow in Figure 8.

Sampling rate, SNR and reconstruction

Uniformly sampling ADCs must adhere to the sampling theorem [1] in order to be able to accurately reconstruct the sampled signal, i.e., the sample rate must exceed the signal bandwidth by a factor of 2. Since the LC-ADC produces non-uniform samples, the sampling theorem is not valid. However, it has been claimed that for LC-ADCs, the *average* sampling rate must exceed 2 × (signal bandwidth) for correct reconstruction [3]. In this work the average sample was 510 S/s. The main ECG power is concentrated between 0.5 and 30 Hz [19]. The "patient monitoring required bandwidth" for ECG signals is specified to 0.05 to 30 Hz [20] and even if we stretch that to "diagnostic grade monitoring", which is up to 100 Hz [20], the sampling rate still exceeds 2×(signal bandwidth) with a satisfactory marginal.

Assuming a signal bandwidth of 100 Hz, we can also calculate the time resolution rate R according to Equation (4):

$$R = \frac{2\times10^6}{100} = 20000$$

Hence we have an SNR of Equation (5):

$20 \times \log 20{,}000 - 11.2 = 74.8$ dB

Inserting this SNR into Equation (2) gives as the necessary resolution of 12 bits required for an equivalent synchronous ADC.

Uniformly sampled signals are reconstructed by interpolation of sinc functions [21]. The reconstruction of non-uniformly sampled signals is less straight-forward. Most reconstructions are based on straight line interpolation and smoothing filters [6]. Several other suggestions have been reported [8,22,23]. In 2009, Kozmin et al. [9] suggested the "Adaptive-weight conjugate Toeplitz method" for reconstruction of non-uniformly sampled signals which is based on trigonometric polynomials. So far though, no method for exact reconstruction of non-uniformly sampled signals has been reported.

Future work

The main objective of this work was to demonstrate the capability of the "core-independent" hardware available in low-cost microcontrollers and how they can be adapted to improve data acquisition of standard bio-metrical signals. A possible design flaw in the proposed design is that the sampler is "blind" for a short time during the off-loading of the sample stack (since the interrupt is disabled). The occurrence of asynchronous transients between the expected ECG transients could potentially lead to loss of information, which of course could be a serious problem. However, unless they are shorter than 2.34 ms (=the sample heap off-loading time) and occur exactly during the offloading window, at least some of the transient will be captured and would indicate a need for adjustment of the design parameters.

Future work is aimed at taking care of this permanently; the problem is only a matter of software design. Instead of waiting for the entire transient to pass before the off-loading starts (and disabling sampling during off-loading) the sample heap can be off-loaded continuously. New samples are continuously transferred to the host computer (without disabling the sampling). A pointer variable points to the next sample to be off-loaded and another pointer points to the next available location to store the next sample (on a circular sample heap).

That way no information will be lost and any transients exceeding the threshold levels will be captured since the UART transmission is core independent.

The next generation of this LC-ADC design will also encompass a 32-bit timer option in order to extend the range of the time intervals. Also, a separate 32-bit timer will be implemented to measure the period of the QRS wave.

References

1. Nyquist H (1928) Certain topics in telegraph transmission theory. Trans AIEE 47: 617-644.
2. Petrellis N, Birbas A, Kikidis J, Birbas M (2010) Asynchronous Analog-to-Digital Conversion Techniques, (1st edn.), INTECH Open Access Publisher.
3. Allier E, Sicard G, Festquet L, Renaudin M (2003) A New Class of Asynchronous A/D Converters Based on Time Quantization. Proceedings of the 9th Int Symp on Asynch Circuits and Systems, pp: 196-205.
4. Hauser MW (1991) Principles of Oversampling A/D Conversion. J Audio Engin Soc 39: 3-26.
5. Agrawal R, Trakimas M, Sonkusale S (2009) Adaptive Asynchronous Analog to Digital Conversion for Compressed Biomedical Sensing. IEEE Biomedcial Circuits and Systems Conference, pp: 69-72.
6. Mark JW, Todd TD (1981) A Nonuniform Sampling Approach to Data Compression. IEEE Trans. on Comm 29: 24-32.
7. Inose H, Aoki T, Watanabe K (1966) Asynchronous delta-modulation system. Electron Lett 2: 95-96.
8. Sayiner N, Sorensen HV, Viswanathan TR (1996) A Level-Crossing Sampmling Scheme for A/D Conversion. IEEE Trans Circ Syst–II: Anal Dig Sign Proc 43: 335-339.
9. Kózmin K, Johansson J, Delsing J (2009) Level-Crossing ADC Performance Evaluation Toward Ultrasound Application. IEEE Trans Circ Syst-Reg Pap 56: 1708-1719.
10. Trakimas M and Sonkusale SR (2011) An Adaptive Resolution Asynchronous ADC Architecture for Data Compresssion in Energy Constrained Sensing Applications. IEEE Trans Circ Syst- Reg Pap 58: 921-934.
11. Silva VML, Catunda SYC (2014) Flexible A/D Converter Architecture Targeting Sparse Signals. Proceedings of the IEEE International Instrumentation and Measurement Technology Conference.
12. Ravanshad N, Rezaee-Dehsorkh H, Lotfi R, Lian Y (2014) A Level-Crossing Based QRS-Detection Algorithm for Wearable ECG Sensors. IEEE J Biomed Health Informatics 18: 183-192.
13. Tang W, Osman A, Kim D, Goldstein B, Huang C, et al. (2013) Contiunuous Time Level Crossing Sampling ADC for Bio-potential Recording System. IEEE Trans on Circ Syst- Reg Pap 60: 1407-1418.
14. Weltin-Wu C, Tsividis Y (2013) An Event-driven Clockless Level-Crossing ADC with Signal-Dependent Adaptive Resolution. IEEE J Sol State Circ 48: 2180-2190.
15. Baums A, Grunde U, Greitans M (2008) Level-Crossing sampling using microprocessor based system. ICSES 2008 International Conference on Signals and Electronic Systems.
16. Microchip Tech Inc (2015) PIC16(L)F1764/5/8/9-14/20-Pin, 8-Bit Flash Microcontrollers. Datasheet DS40001775B.
17. Future Technology Devices International (2016) TTL-RS232R to USB Serial Converter Range of Cables. Datasheet FT_000054.
18. Ashley EA, Niebauer J (2004) Cardiology Explained. London: Remedica.
19. Merri M, Farden DC, Mottley JG, Titlebaum EL (1990) Sampling Frequency of the Electrocardiogram for Spectral Analysis of the Heart Rate Variability. IEEE Trans Biomed Engin 37: 99-106.
20. Texas Instruments (2016) ECG and EEG applications. Quick reference guide.
21. Ghate AH, Khanchandani KB (2013) Difference between Reconstruction from Uniform and Non-Uniform Samples using Sinc Interpolation. Int J Research Rev Eng Sci Techn 2: 17-21.
22. Tertinek S, Vogel C (2008) Reconsruction of nonuniformly sampled bandlimited signals using a differentiator-multiplier cascade. IEEE Trans Circ Syst 55: 2273-2286.
23. Feichtinger HG, Gröchenig K, Strohmer T (1995) Efficient numerical methods in non-uniform sampling theory. Num Math 69: 423-440.

Citation: Bengtsson L (2017) Single-Chip Implementation of Level-Crossing ADC for ECG Sampling. J Electr Electron Syst 6: 219. doi:10.4172/2332-0796.1000219