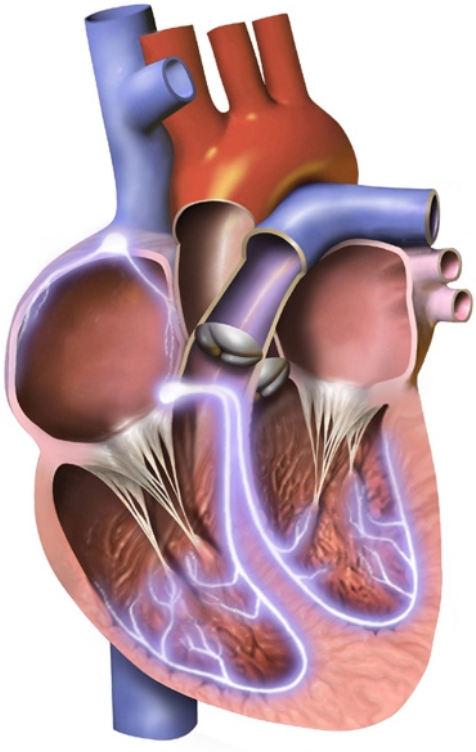
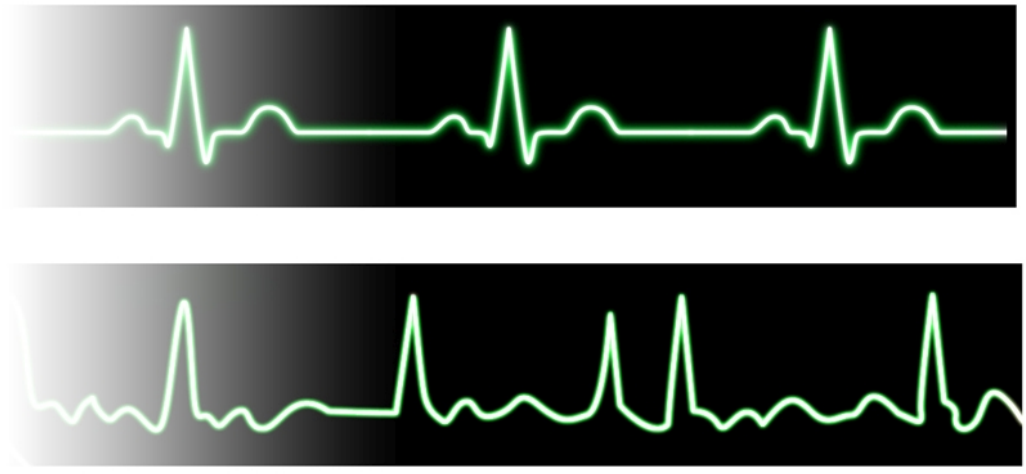
Multichannel LC ADC
to Record Atrial Electrograms





Aurojyoti Das

TU Delft

Multichannel LC ADC

to Record Atrial Electrograms

by

Aurojyoti Das

to obtain the degree of Master of Science in Electrical Engineering
at the Delft University of Technology,
to be defended publicly on Friday August 23, 2019 at 10:00 AM.

Student number:	4737652	
Thesis Project:	August, 2018 - August, 2019	
Thesis committee:	Prof. dr. ir. W. A. Serdijn,	TU Delft, supervisor
	Dr. Ir. C. J. M. Verhoeven,	TU Delft
	Dr. V. Valente,	University College London & TU Delft
	Ir. S. Rout,	TU Delft

ABSTRACT

Biosignals such as electoencephalogram (EEG), electrocorticogram (ECoG), atrial electrogram (AEG) etc. are being recorded from multiple channels simultaneously to improve the spatial resolution of the signals. Conventional multichannel synchronous Analog-to-Digital Converters (ADCs) are used to convert the analog continuous time signals into discrete digital values. Several biosignals have a sparsity in time domain as they have fast-rising peaks in between periods of low activity. Use of conventional synchronous ADCs for conversion of such signals is not an efficient approach as their operation is constant, irrespective of the activity of the input signals. Asynchronous ADCs such as level-crossing (LC) ADCs exploit the sparsity of biosignals and thus their operation is activity-dependent. However, multichannel configurations of LC ADCs do not yet exist. This problem is investigated in this work and a new ADC architecture is presented that can combine synchronous sampling with level-crossing quantisation method while converting input signals from several channels simultaneously. The synchronous LC ADC presented in this work achieves 3.37 times reduction in quantisation steps and 6 times reduction in number of output bits generated during conversion of AEG signals as compared to conventional synchronous ADCs. The problem in existing LC ADCs of data overhead in adaptive resolution technique is solved through a novel method named split resolution technique which is also presented in this work.

PREFACE

Aurojyoti Das,
Delft, August 2019

अनुगच्छति प्रवाह

(Anugacchati Pravahah)
Translation: Go with the flow

The sanskrit expression stated above is the guiding light for this work. As in life and in engineering, the input affects everything. Rather than trying to manipulate the input fruitlessly, it is better to exploit what the input has to offer; it is better to go with the flow. The following work is inspired by this wisdom, and implements the same in a biosignal recording system.

ACKNOWLEDGEMENTS

I am grateful to several people who have helped me, accompanied me and inspired me in this journey. Firstly I would like to thank Wouter for the freedom and support he gave me in exploring ideas on my own and also for inspiring me to look for different perspectives which are normally not considered. I would like to thank Alessandro and Sampi for guiding me throughout the project and supporting me whenever I needed their help.
I am grateful to everyone in Bioelectronics group for their engaging discussions and the fun times we spent together. I will miss the movie nights, borrel and the (often long) monthly meetings.
I thank my friends at 'ME house' for all the fun evenings, chai sessions and weekend parties. I would not have survived without their support. I am grateful to Shoubhik for helping me even with the smallest doubts.
I thank my friends and family back home whose love and support encouraged me and kept my spirits up. Lastly, I would like to thank my parents for always believing in me and for encouraging me to take up new challenges every day.

Aurojyoti Das
Delft, August 2019

Contents

1

INTRODUCTION

1.1. BACKGROUND

Developments in bio-signal recording systems have enabled deeper investigation of medical conditions. Down-scaling of transistor feature sizes and design of low-power circuits have enabled use of microchips in implants and wearable devices. Bio-signals from several locations can now be recorded simultaneously by using a single device. The increased density of acquired data points provides a better understanding of the underlying medical conditions. These benefits however come at a cost of increased design complexity of the device and higher power consumption owing to increased data transmission rates. Power consumption is particularly exacerbated for wireless modes of data transmission which is critical for implantable and wearable devices. This project explores possibilities of finding a solution to these problems through a novel data conversion approach.

1.2. PROBLEM STATEMENT

Atrial fibrillation (AF) is a medical condition which causes irregularities in the electrical impulses conducted across the heart and thus adversely affects its regular rhythm. While electrocardiograms (ECGs) can help in detection of AF, they cannot help in localising the origin of AF sources and identification of the substrate perpetuating AF. Electrical paths of conduction through the cardiac muscles need to be analyzed to determine the source of the condition and to formulate methods for therapy. Atrial electrograms (AEGs) are recorded from the atrial myocardium to accomplish this [1].

Recording of AEGs from multiple locations simultaneously is helpful in the analysis of the condition [2]. However, the current setup used for recording the AEGs from multiple locations involves transmitting the analog input signals across a long cable which is then conditioned and converted into digital values. This setup suffers from addition of interference and disturbance to the input signal captured by the electrodes.

This problem can be mitigated by using an IC positioned near the recording patch which consists of the analog front ends (AFEs) for all the recording channels. The AFE consists of a low noise amplifier (LNA), a filter and an analog to digital converter (ADC) which amplify, filter and convert the analog signal into the discrete digital values, respectively. Block diagram of an AFE is shown in Fig.1.1. The converted signal is then stored in a digital memory and can be used later for signal analysis. The challenge in creating an IC for a multichannel recording system is to fit the AFEs for all the channels together within the area and power constraints.

Different approaches could be adopted to design such a multichannel recording system. A separate AFE could be used for each channel and signal could be converted in each AFE independently. This approach is not very efficient as it would require a large area for the IC and power would be consumed in ADCs for each separate channel. Another approach could be to share some components of the AFE among several channels, as shown in Fig.1.2. In this approach the ADC is time-shared among several channels and at a given instant of time the signal from one of the channels undergoes conversion in the ADC [3]. This is repeated

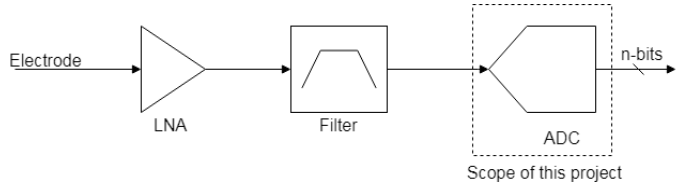


Figure 1.1: Conventional analog front end (AFE) with low noise amplifier (LNA), filter and analog to digital converter (ADC).

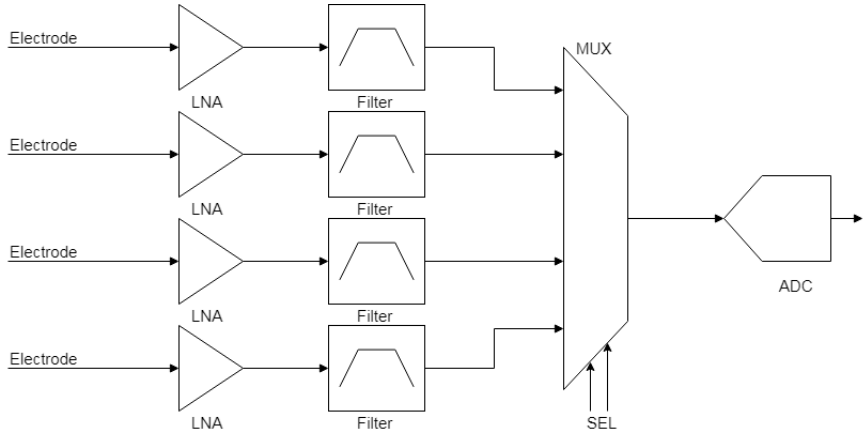


Figure 1.2: Multichannel analog front end with time-shared ADC.

for each of the channels in a round-robin fashion within a fixed number of clock cycles. Hence, a substantial amount of area can be saved through this approach. At the same time the time-multiplexed ADC needs to operate multiple times faster than a regular ADC to maintain the same sampling rate for each channel. The multiplicative factor depends on the number of channels being multiplexed to the ADC. Time-multiplexing is a good approach for bio-signal recording systems as the bandwidth of bio-signals is limited to a few KHz and hence an ADC with a sampling rate in the range of kS/s would be applicable.

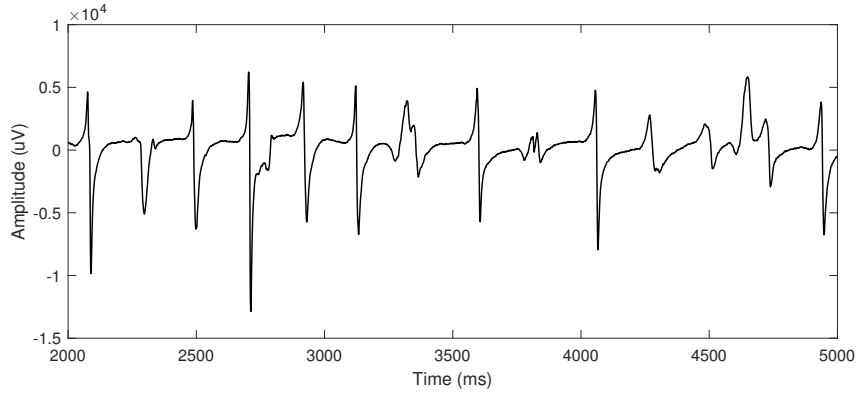


Figure 1.3: Example of a typical atrial electrogram.

Several types of ADCs such as Nyquist Rate ADCs and ΣΔ ADCs are used in such AFEs. In recent times the SAR ADC has gained popularity owing to its simple operating principle and low power consumption [4]. It follows the synchronous sampling principle, i.e. it generates a sample with every clock cycle as per the sampling rate. However, some signals have minimal activity for considerable duration of time, as shown in Fig. 1.3. This can be seen as sparsity in time domain. For these kind of signals, the synchronous recording method is not efficient as the signal is sampled even though there is no change in the input signal. In this

scenario, asynchronous recording methods are a better choice [5]. In the asynchronous recording method a constant sampling rate is not used, but rather the signal is sampled only when a property (such as amplitude) of the input signal changes.

Level-crossing (LC) sampling is one such asynchronous recording method. In LC sampling the signal is sampled when it crosses a preset reference level. Thus when the signal is stagnant at any amplitude, no samples are generated. The output of synchronous ADCs is an n-bit digital value whereas in LC ADCs the output is just 2 bits which denote the time and direction of level-crossing. Hence, the number of output wires are reduced to just 2. This can be further reduced to 1 by using an encoding scheme [6]. This reduces the amount of data generated by the ADC as well. These advantages make the LC ADC an attractive option in applications where wiring complexity and storage space are critical.

1.3. Research Questions

While synchronous ADCs such as SAR ADCs are used in multichannel configurations by using the time-multiplexing approach [3], multichannel asynchronous ADCs have not yet been developed. As explained above, they offer an advantage over synchronous ADCs in the conversion of biosignals. Moreover, the AEG signal, much like the ECG signal has its own unique characteristics. Exploiting these characteristics could also be help in making the recording system more efficient. Hence, this project aims to answer the following research questions -

1. Can the level-crossing sampling approach be implemented in a multichannel configuration?
2. Can prior knowledge of the input signal (in this case, AEG) be helpful in designing a more efficient ADC?

This project aims to develop a multi-channel LC ADC which can satisfy the requirements of the multi-channel atrial electrogram recording system.

1.4. Structure of Report

The operating principle of LC ADCs and its existing implementations reported in literature are discussed in Chapter 2. The architectures considered to design the multichannel LC ADC and the system level design considerations are discussed in Chapter 3. The model implementations of the proposed multichannel LC ADC (MLC ADC) architecture in VerilogA and MATLAB are discussed along with their results in Chapter 4. The circuit implementation of the proposed MLC ADC is discussed in Chapter 5. The results and conclusions from the models and circuit implementation are discussed in Chapter 6. Finally, the conclusions of the project and recommendations for future work are discussed in Chapter 7.

2

LEVEL CROSSING ADC

In this chapter the operating principle of asynchronous sampling methods such as LC sampling is explained in detail. Features of the existing LC ADCs reported in literature are discussed and some of their disadvantages are listed which would be addressed in this work.

2.1. SYNCHRONOUS RECORDING VS ASYNCHRONOUS RECORDING

Recording of continuous-time analog signals in discrete digital form can be performed through two different methods. The synchronous recording method creates samples of the input signal at regular, fixed time intervals and then converts them into discrete digital values. The sampling rate is required to adhere to the Nyquist criterion i.e. it should be at least twice that of the bandwidth of the input signal. Moreover, the amount of data generated is fixed at each interval. This form of sampling is shown in Fig.2.1(a). As can be observed, even for signals that remain inactive for significant periods of time, the sampling rate is constant and the same amount of data is generated at each sampling interval. This is an inefficient method of conversion.

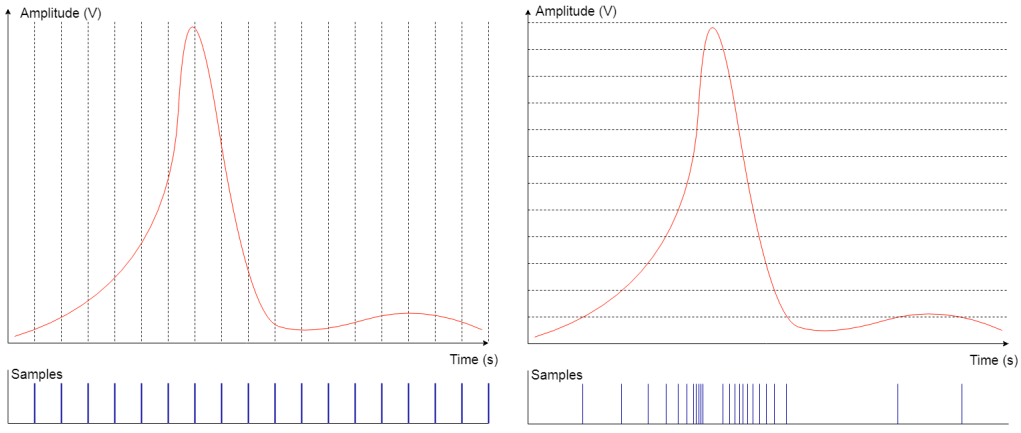


Figure 2.1: (a) Example of synchronous recording (b) example of asynchronous recording.

Asynchronous recording however is driven by the signal directly. The rate of sampling and thus the amount of data generated is dependent on the rate of change of the signal. For slow moving signals the number of samples generated is low and for peaks number of samples generated is high. Hence, there is an inherent compression of the data generated in this sampling paradigm. Example of LC sampling is shown in Fig.2.1(b). As can be observed in the figure the number and density of samples generated increases when the signal slope increases and vice versa.

2.2. Level Crossing ADC

The Level Crossing ADC is different from conventional Nyquist-rate ADCs as it implements an asynchronous recording paradigm rather than the synchronous recording paradigm. Use of this asynchronous recording approach obviates the requirement of a clock signal as well since the whole operation is asynchronous. The ADC tracks the input signal and generates output bits when the input signal crosses one of the reference voltage levels. The reference voltage levels are defined by the target quantiser resolution and the full scale voltage level of the input signal, whose relation is shown in Eq.2.1.

$$Vref = \frac{V_{in,p-p}}{2^M} \tag{2.1}$$

where M is the quantiser resolution and $V_{in,p-p}$ is the full scale input signal. Another voltage reference with same magnitude is also defined for sampling signals below the V_{cm} level (mean level of the input signal). These two reference levels together are referred to as the 'reference window'. If the input signal stays within the reference window, there is no output generated by the ADC. Hence, the conversion process is entirely signal-driven. The output of the LC ADC does not have any aliasing as the sampling is done in continuous time.

In synchronous recording, the amplitude is determined by the ADC and the time interval of sampling is fixed. However, in the LC ADC which uses asynchronous sampling the amplitude levels are fixed while the time interval of sampling is undetermined. This property of LC ADCs is exploited in continuous time DSPs as there the timing information of the samples is not required. However, for LC ADCs to be compatible with discrete-time/synchronous DSPs, the time of generation of samples needs to be quantized. Hence, the source of quantization error in asynchronous sampling is the oversampling ratio (ratio of timer speed/input signal bandwidth) rather than the LSB value of the quantizer. The resulting SNR due to time quantization in LC ADCs (Eq.2.2) is derived in [7].

$$SNR = 20.logOSR - 14.2\,dB \tag{2.2}$$

where OSR is the Over-Sampling Ratio of the ADC. There are broadly two ways of implementing level crossing sampling[8] - the 'Fixed Window' architecture and the 'Floating Window' architecture.

2.2.1. Fixed Window LC ADC

In the fixed window architecture the input signal is tracked continuously and as soon as it crosses the reference levels a certain voltage is subtracted (or added) to it so that it reaches the common mode / mean voltage (V_{cm}). The input signal is again tracked, and this operation is repeated. The block diagram of a Fixed Window Level Crossing ADC and its operation are shown in Fig.2.2.

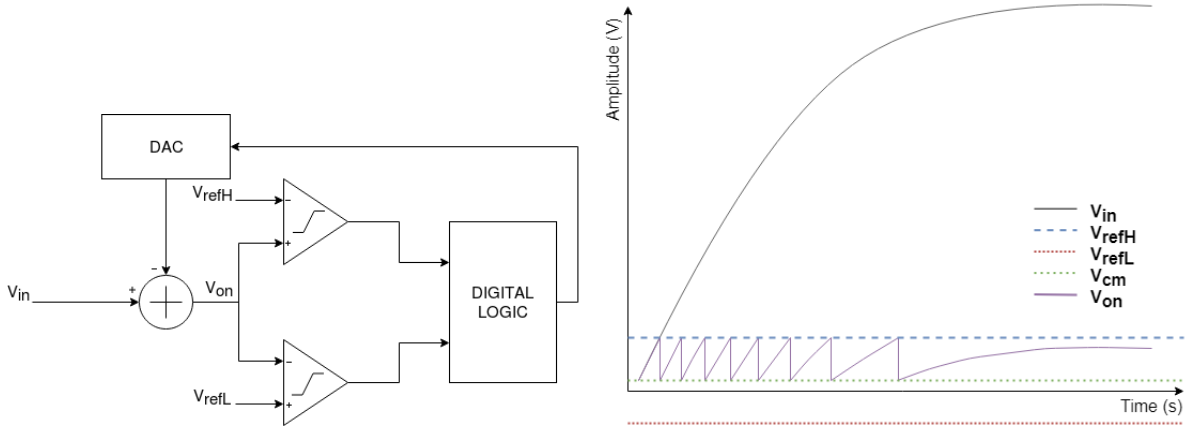


Figure 2.2: (a) Block diagram of fixed window LC ADC and (b) timing waveform of fixed window LC ADC.

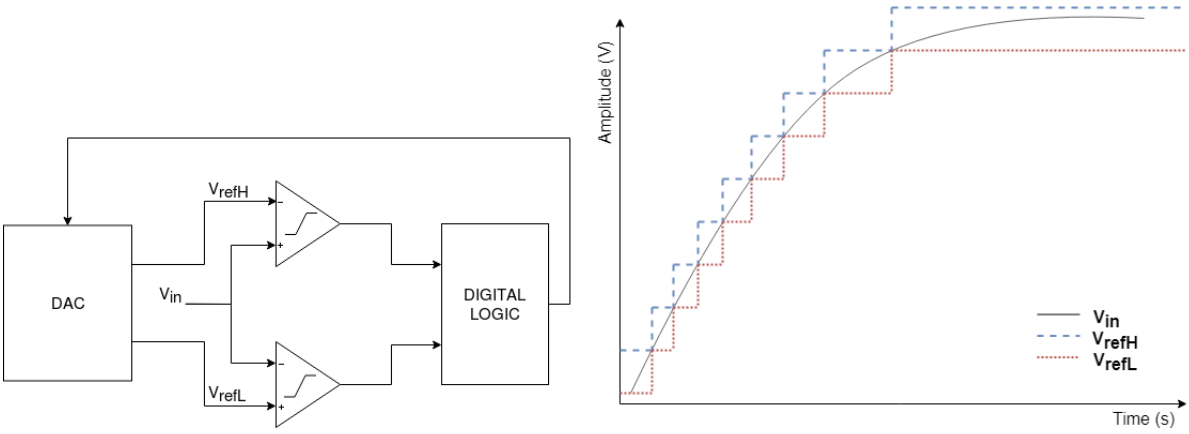


Figure 2.3: (a) Block diagram of floating window LC ADC and (b) timing waveform of floating window LC ADC.

2.2.2. Floating Window LC ADC

In the floating window architecture the reference window itself follows the signal as it crosses one of the reference levels of the window. The reference levels of the window are incremented (or decremented) by 1 LSB step (equivalent to the reference voltage level) depending on which direction the input signal is traversing. The block diagram of a Floating Window Level Crossing ADC and its operation is shown in Fig.2.3.

The type of DAC to be used in the ADC depends on the selected architecture. While the floating window architecture requires a multi-bit DAC, the fixed window architecture requires a 1-bit DAC. In [9], it was demonstrated that the fixed window LC ADC is smaller and more power-efficient owing to its 1-bit capacitive DAC.

2.3. Literature Review

Several LC ADCs are reported in literature. While most of the designs are fully asynchronous, some like [7] are pseudo-asynchronous to be better compatible with digital blocks that follow the ADC. However, the design still uses two comparators. Some designs have fully asynchronous frontend, DSP and DAC [10] which obviate the need for a clock signal in the whole signal processing chain. Such designs cannot be easily integrated into systems which require storage of the output in traditional digital memory. Different methods of storing and tracking the input signal are designed as well. An analog storage element is used in [7] to store and track the input signal instead of using a DAC. Similarly, a buffer with reset function is used in [8]. Using a buffer instead of a DAC simplifies the design of the LC ADC but at the same time it introduces an error in the sampling. The buffer takes some time to recover after being reset and to start tracking the input signal again. The input signal is not tracked in this time and is lost. In [11] a floating window LC ADC implementation is presented which is powered by a 300mV supply. A QRS detection circuit is also implemented along with the LC ADC which uses the output of the LC ADC and timing information of the samples to perform the QRS detection. In [12], the LC ADC is designed to be fully synchronous and uses a single path for level crossing operation. Instead of using two comparators as in conventional LC ADCs the design compares the signal with two reference voltage levels in different clock cycles using the same comparator. Similarly in [13] a single comparator and a threshold detection circuit is used to determine the level crossing of the signal. LC ADCs generally use asynchronous comparators which are affected by the slope of the input signal. In [14], the reference window size is changed according to the slope of the signal, thus ensuring similar response from the comparators for all signal slopes. This approach however requires an extra DAC and thus causes overhead in area and circuit design.

Operating principle of the LC ADC can be likened to that of the Δ Modulator used in $\Sigma\Delta$ ADCs. These ADCs can perform conversion with high resolution even though they use a very low resolution quantiser.

This is achieved by noise shaping of the quantisation noise. Noise shaping is added to an LC ADC in [15] to achieve better SNR performance. Moreover, no DACs are used in the design and the loop-delay is reduced considerably enabling the ADC to be used for applications in the Intermediate Frequency range (10MHz - 50MHz). Due to the increased bandwidth of the input signals the resolution of the resulting ADC is reduced to a lower range. The problem of realizing the required dynamic range in a low-voltage system is resolved in [16] by converting the input voltage into a current by using a Programmable Voltage-to-Current Converter (PVCC) which is then fed into the LC-ADC.

Chopping is generally used in AFEs to move the 1/f noise out of the band of the input signal bandwidth. This is accomplished by up-modulating the input signal to a frequency above the 1/f knee frequency and then down-modulating it at the same frequency. However, LC ADCs operate in the continuous time domain. A clockless method of chopping is introduced in [17] which uses the input signal frequency along with a pseudo-random number sequence to generate the chopping control signal.

The relation between quantiser resolution, timer resolution and the resulting SNR and dynamic range is derived in [18], which can be written as Eq.2.3.

$$BW = \frac{1}{\pi\sigma 2^M}, \quad DR <= \frac{1}{2^{M-1}} \tag{2.3}$$

where BW is the input signal bandwidth, σ is the loop delay of the quantiser, M is the quantiser resolution and DR is the dynamic range of the LC ADC. This shows that there is a limitation on the maximum dynamic range achievable in an LC ADC for a given signal bandwidth. The dynamic range of the ADC can be increased by increasing the quantiser resolution but at the cost of reducing the maximum input signal bandwidth. At the same time the number of samples generated in an LC ADC increases exponentially as its quantiser resolution increases[19]. The amount of data generated is increased two-fold for every extra bit of resolution increased in the quantiser. Hence, for higher resolutions the LC ADCs produce much higher amount of data as compared to conventional synchronous ADCs. This issue overshadows the benefits of activity-dependent operation found in LC ADCs.

These limitations can be overcome by using adaptive resolution techniques in which the quantiser resolution is changed according to the slope of the signal [18],[20]. While in [18], the slope detection is done off-chip in an FPGA, in [20] the slope detection is done on-chip based on the time elapsed between token generation. The operation of an adaptive resolution LC ADC is shown in Fig.2.4. Even though the adaptive resolution technique reduces the the number of samples generated, it increases the amount of data generated by the ADC. The event of change of resolution and the degree of change are not recorded in the pulses generated by the LC ADC. They are however needed so that the signal can be accurately reconstructed. Hence, additional bits need to be added to the output to specify where the resolution was changed and by how much. This increases the amount of data generated by the ADC as a whole and increases power consumption in the data transmission block, especially if it uses a wireless mode of transmission.

LC sampling is combined with a SAR-based quantiser in [21] to implement a fixed window LC ADC. Here, the SAR ADC is used as a residue quantiser while the level crossing sampling is implemented in the digital domain. The output of the residue quantiser is used to perform the subtraction of the input signal through a DAC. An AR algorithm is also implemented in the digital domain to reduce the amount of output data generated. In [22] an activity-dependent SAR ADC was reported in which the number of quantisation steps for each sample is dependent on the signal activity as compared to its previous sample. The quantised reference voltage level of the previous sample is used as the starting point of quantisation for the next sample. Thus, if the difference between two samples is less than one LSB step then the quantisation of the new sample is completed in just two quantisation steps. However, the output of the ADC is constant for each sample and thus it does not reduce the output like LC ADCs.

A multichannel implementation of the LC ADC has not yet been reported. Existing multichannel ADCs [3] sample the input signal synchronously and hence do not exploit the benefits of asynchronous sampling. Compressed Sensing is an important feature which is being integrated into biosignal recording systems

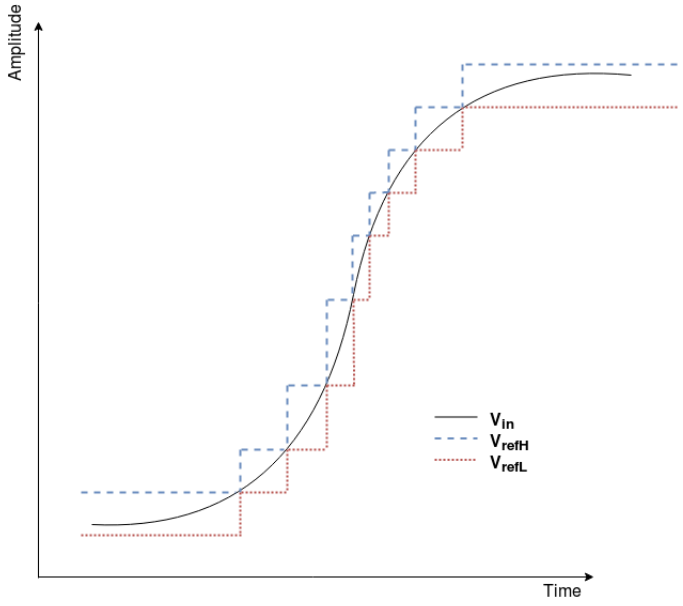


Figure 2.4: Operation of Adaptive Resolution technique in Floating Window LC ADC. The window size is changed according to the slope of the input signal.

Table 2.1: Advantages and disadvantages of an LC ADC

Advantages	Disadvantages
simple design	continuous-time comparators
small area	total number of output bits
signal dependent operation	unsuitable for high resolution
optimisation to reduce samples	resolution is bandwidth-limited

[23]. These features require complex algorithms or circuitry and put an overhead on the frontend in terms of area, power and design complexity. Several biosignal recording systems record signals from multiple channels simultaneously to improve the spatial resolution of the recorded signals. While some systems such as in [24] use one ADC for each channel, some like [25] use time-multiplexed ADCs. In [26], 1024 input channels are used to record neural signals and 64 time-multiplexed SAR ADCs are used to record from all the channels. The amount of data produced in such systems would be quite large. Thus, data compression would be required to minimise the data size transmitted outside the device. The number of input channels in biosignal recording systems will increase in the future and the data would need to be transmitted wirelessly. Therefore, reduction in the amount of data produced would become critical in such systems.

Thus, the following problems in existing LC ADCs need to be addressed-

1. Large amount of data generated in LC ADCs at high quantiser resolutions.

2. Extra data required to encode the change in quantiser resolution in adaptive resolution technique.

3. Absence of multichannel LC ADC architectures to record from several input channels simultaneously.

The advantages and disadvantages of LC ADCs as compared to conventional synchronous ADCs are summarised in Table. 2.1.

2.4. CONCLUSIONS

LC ADCs are a way to implement the asynchronous recording method. They can be designed using one of the two architectures - fixed window and floating window architectures. There are several LC ADCs reported in literature which address specific problems associated with conventional synchronous ADCs or improve performance over existing LC ADCs. However, there are still some issues which have not been addressed or can be addressed in a better way. Moreover, multichannel configurations of LC ADCs have not yet been designed. Proposed solutions to these problems are discussed in the next chapter.

3

System Level Design Considerations

The working principle and existing designs of LC ADCs are discussed in detail in the previous chapter along with some problems in these implementations. In this chapter the proposed solutions to the problems are discussed. Different solutions considered to address the problems and the procedure adopted to select the best solutions are also discussed.

3.1. System Requirements

The AEG signals being recorded have a maximum amplitude of $80mV_{p-p}$ and the minimum amplitude that needs to be recorded is $0.1mV_{p-p}$. Hence, the required dynamic range (DR) is calculated as per Eq.3.1 to be 800. As shown in the previous chapter, the relation between the target DR and the quantiser resolution required in an LC ADC is given by Eq.2.3. This gives the required quantiser resolution, M to be 11-bits. The target bandwidth for the input AEG signals is set to be 400Hz. The LC ADC should be able to quantise the signals in the given bandwidth reliably. The signal characteristics of the AEG signal are summarised in Table3.1.

$$DR = \frac{Vmax_{(p-p)}}{Vmin_{(p-p)}} \tag{3.1}$$

Table 3.1: Signal properties of atrial electrograms

Property	Value
Max amplitude (mV_{p-p})	80
Min amplitude (mV_{p-p})	0.1
Bandwidth (Hz)	0.5-400

Table 3.2: Target specifications of proposed LC ADC

Property	Value
Number of channels	192
Dynamic Range (dB)	58.06
Bandwidth (Hz)	400
Quantiser Resolution (bits)	11

The AEG signals are recorded simultaneously from 192 electrodes. Using an ADC for each channel is not a viable approach as a large area of the IC would be used by ADCs. The power consumption would also be much higher since each ADC would operate and draw power simultaneously. A single ADC cannot convert signals from all 192 channels either. Hence, a number of ADCs can be used with each ADC

converting input signals from several channels simultaneously. As mentioned in chapter 2 a multichannel LC ADC currently does not exist. The architecture for an LC ADC needs to be designed such that multiple channels can be serviced together. The different topologies considered that could fulfill the listed criteria are described in subsequent sections.

As discussed in the previous chapter the number of output samples generated by the LC ADC doubles with the addition of every bit of resolution. The target resolution of the LC ADC is 11 bits which is quite high for an LC ADC. This implies that the number of output samples generated would also be high. Hence, techniques such as adaptive resolution need to be applied to reduce the number of samples generated without compromising on the quality of the output. These techniques are also discussed in subsequent sections. The target specifications of the LC ADC are listed in Table.3.2.

3.2. Modelling Level Crossing ADC

VerilogA models are developed to analyse the behavioral operation of the LC ADC. The fixed window LC ADC design and the floating window LC ADC deisgn are used as reference for the models. The model for fixed window LC ADC as shown in Fig.3.1 consists of a DAC, a multiplexer (MUX) and two comparators. The logic block shown in Fig.2.2 is included inside the DAC in the verilogA model. The multiplexer is used to select the reference level which is supplied to the comparator. Moreover, the signals for each input polarity of the comparator are also selected through the MUX. The reference level is selected depending on the output of the other comparator which determines the direction of the input signal. This is accomplished by comparing the output signal of the DAC with the common mode voltage (Vcm, in this case 0V). If the signal is ramping up then the output of the DAC is compared with the upper reference level. Otherwise the lower reference level is compared with the output of the DAC as the reference (connected to negative polarity input of the comparator). Operation of the fixed window LC ADC model is shown in

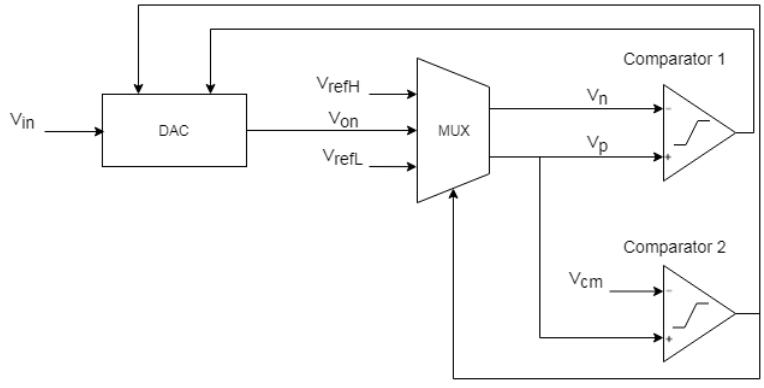


Figure 3.1: Model of Fixed Window Level Crossing ADC.

the flowchart in Fig.3.2. As soon as a level-crossing is detected by the comparators in either direction, the comparators' output signals are fed back to the DAC which then subtracts (or adds) a voltage equivalent to the LSB voltage to the input signal. Hence, the DAC keeps tracking the input signal while ensuring its output stays within the reference window. The simulation waveforms are shown in Fig.3.3. The input here is a 2Vp-p signal while the LSB voltage is 200mV. Hence, ten reference levels are crossed by the input signal during conversion. The output signals from the comparators are used to reconstruct the input signal, shown in Fig.3.4.

The block diagram of the model for floating window LC ADC is shown in Fig.3.5. The reference window in this model is updated every time the input signal crosses either of the reference levels. The reference window is thus shifted by 1 LSB step in the direction in which the signal crosses the reference window. The DAC receives feedback from both comparators and changes the reference levels accordingly. The simulation waveforms are shown in Fig.3.6. The input signal with 1 Vp is converted sampled with 10 reference levels.

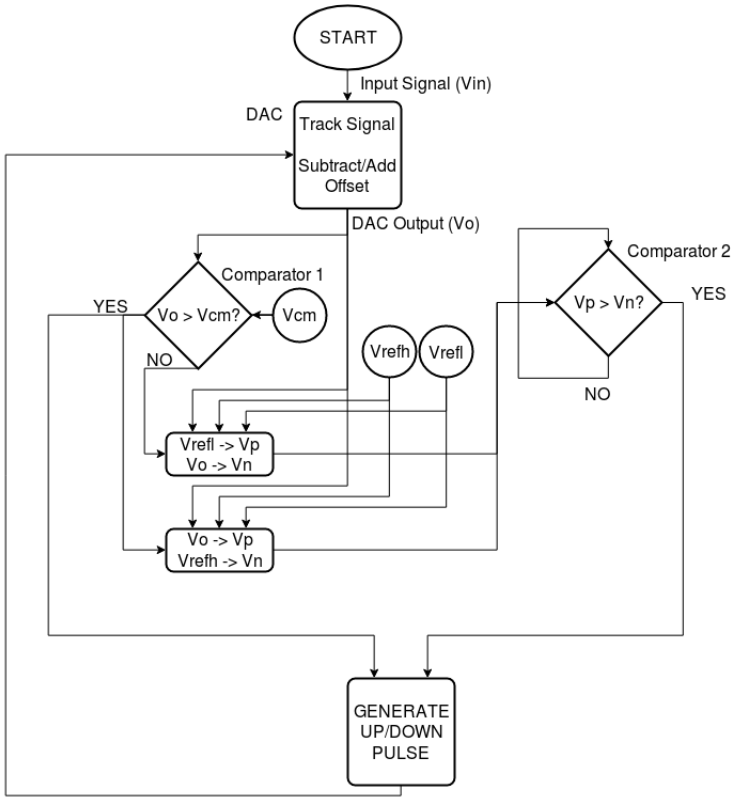


Figure 3.2: Flowchart for operation of VerilogA Model of LC ADC.

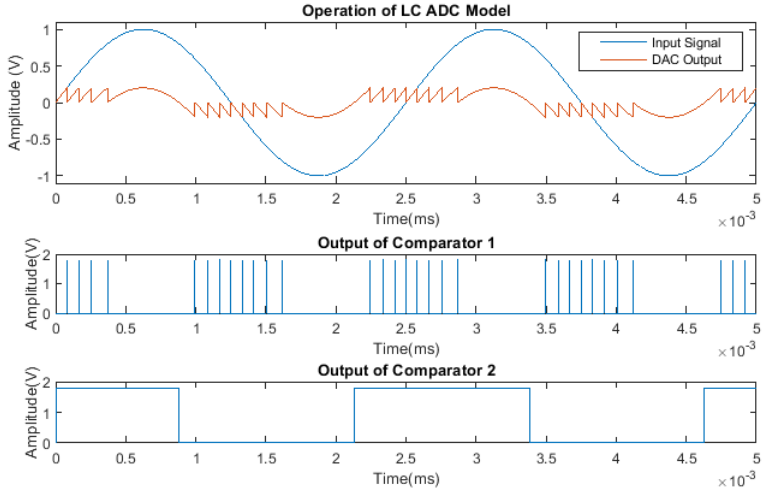


Figure 3.3: Operation of LC ADC model built with VerilogA. The top plot shows the input signal and the output of the DAC. The two plots below show the outputs of the comparators.

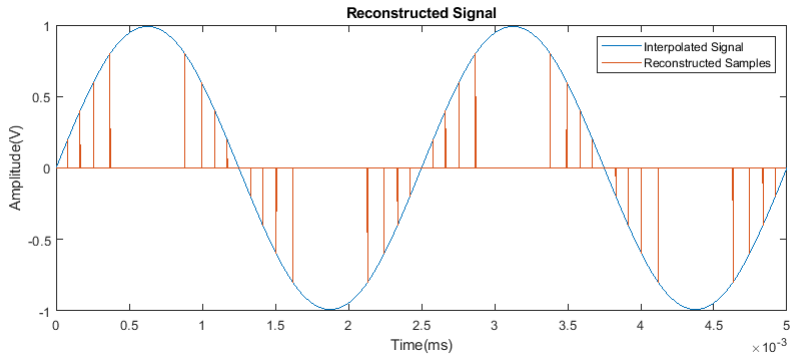


Figure 3.4: Reconstructed signal from samples by spline interpolation.

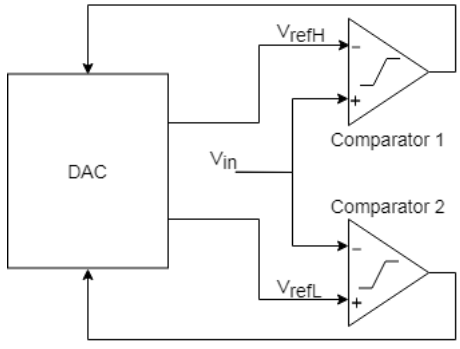


Figure 3.5: Block diagram of model of floating window LC ADC.

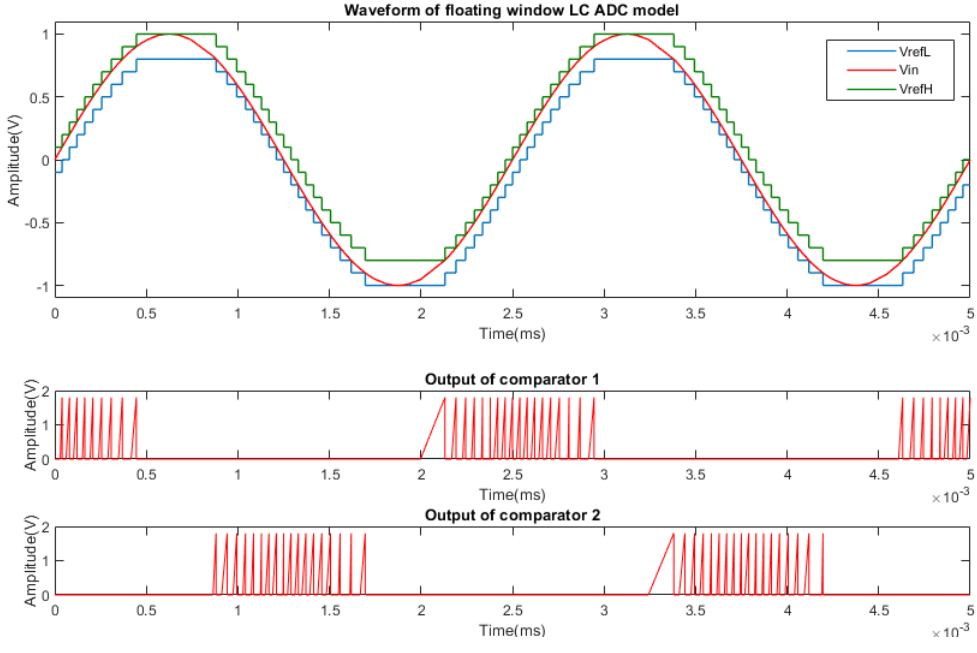


Figure 3.6: Waveforms illustrating operation of floating window LC ADC model.

3.3. Split Resolution: A Proposed Alternative to Adaptive Resolution

As explained in the previous chapter the adaptive resolution techniques reported in literature increase the total amount of data generated in the ADC. Hence, a novel method of data reduction, named Split Resolution (SR) technique is proposed which does not require additional bits to show change in resolution. AEG signals are similar to ECG signals as they have fast and high peaks in between periods with relatively low-amplitude and low-frequency signals. An example of AEG signal is shown in Fig.3.7. Most of the signal activity is confined within a narrow range of amplitude close to the common mode. The spikes are narrow and reach relatively higher amplitudes compared to the remaining parts of the signal. The spikes therefore contribute significantly to the total number of samples generated by the LC ADC. A threshold can be set beyond which when the signal increases, the resolution of the LC ADC is reduced. The threshold level and the degree of reduction of resolution can be programmed in the LC ADC. The advantage of this method is that no extra memory is required to store the moment and degree of change of resolution. The parameters are programmed in the LC ADC prior to recording. During reconstruction the number of levels can be counted from V_{cm} in either direction to determine the exact moment of application of split resolution and thus reconstruct the signal accurately. Moreover, the change in resolution can be made gradually in steps, such that if the spike is high enough then the resolution can be decreased gradually instead of abruptly all at once. Even then the number of LSB steps crossed for each resolution setting can be fixed and hence, extra memory is not required.

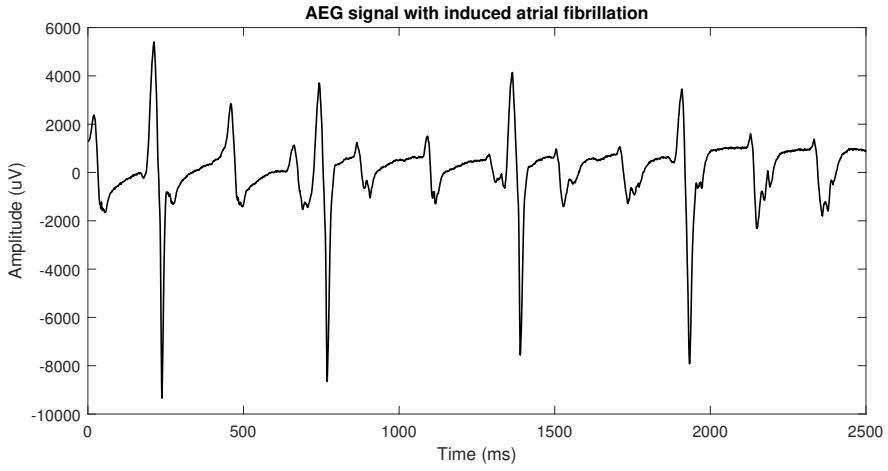


Figure 3.7: AEG signal with atrial fibrillation condition. Source: EMC Rotterdam.

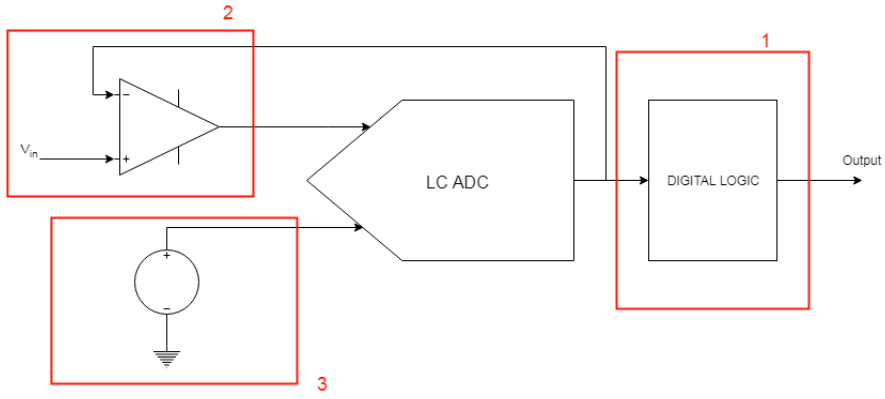


Figure 3.8: Methods for implementation of Split Resolution technique in LC ADC.

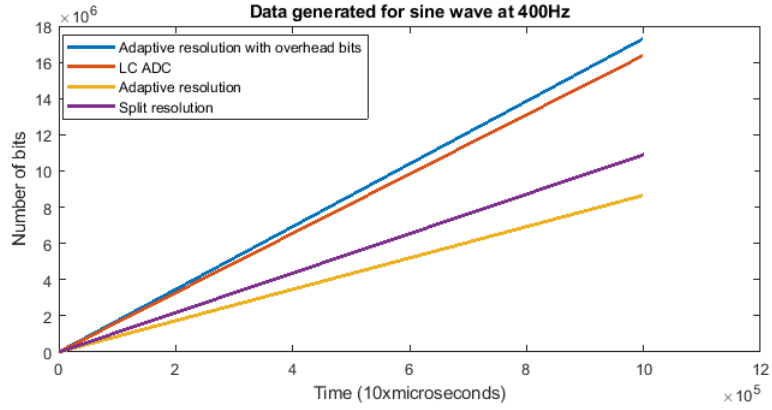


Figure 3.9: Plot for amount of data generated in different level crossing algorithms for a sinusoidal input signal at 400Hz.

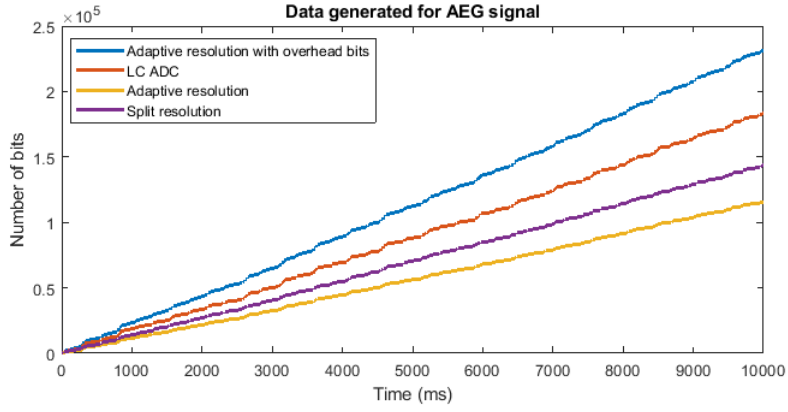


Figure 3.10: Plot for amount of data generated by different level crossing algorithms for an AEG signal.

The disadvantage of the proposed split resolution technique is that the accuracy of location of the peaks are reduced. However, since the LC ADC itself has a high resolution it would not degrade the resolution at the peaks substantially. Implementing the split resolution technique would help in reducing the number of samples generated by the LC ADC.

Three methods for implementation of the SR technique in an LC ADC are proposed. These are illustrated in Fig.3.8 and described below -

1. Dropping Samples in Output Logic - The output logic block combines the output pulses of the comparators and encodes the output accordingly. The Split Resolution technique can be applied here by removing pulses from the output stream. If a one-bit reduction is applied, every other pulse can be removed from the stream.

2. Changing gain of Variable Gain Amplifier (VGA) - The input signal coming from the electrodes need to be filtered and amplified so that the signal can be properly converted in the ADC. The amplifiers used for such applications have a constant gain. However, if the gain is changed according to the amplitude level of the signal then the effective number of quantisation levels required can be changed. Thus, as the signal crosses a threshold, the gain can be reduced so that the ADC would need less number of level crossings to sample the peak.

3. Varying reference window size in DAC - The DAC used in LC ADCs decides the reference window size. Hence, the window size can be varied in the DAC by using feedback from the comparators. In the Fixed Window architecture the offset injection can be varied and in the Floating Window architecture the reference levels generated can be varied.

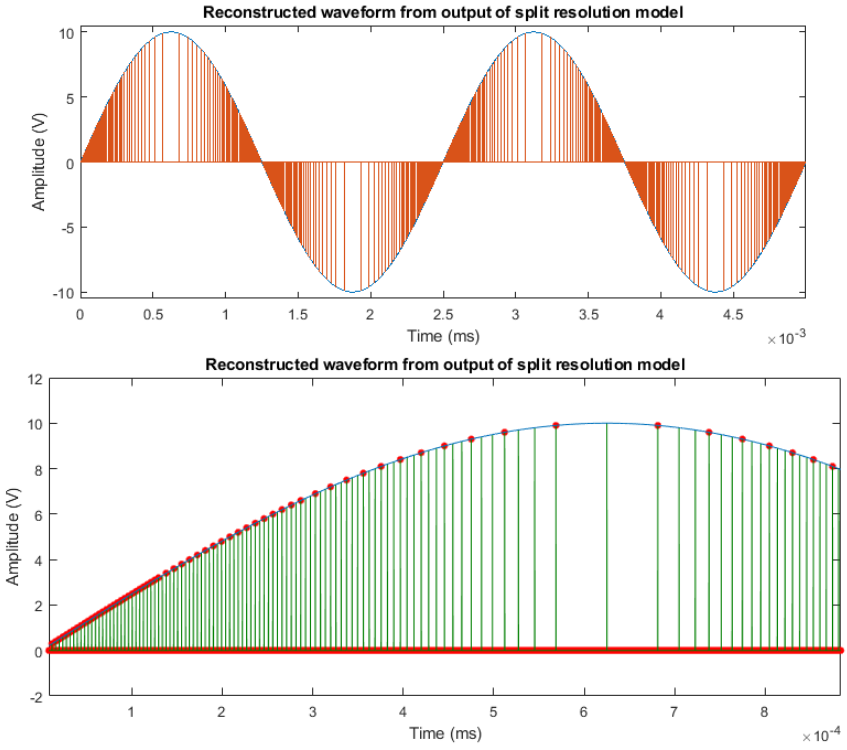


Figure 3.11: (a) signal reconstructed from output samples of LC ADC model with split resolution technique (b) reconstructed signal showing samples dropped (not marked with red) in the output after application of split resolution technique in the digital output block.

Effectiveness of the Split Resolution technique is checked by performing a comparative estimation study of the different LC ADC operation modes in MATLAB. In the modelling setup for the adaptive resolution technique the slope of the input signal is estimated and the resolution is decreased in two steps (by a bit in each step) according to certain thresholds in slope. Similarly for the split resolution technique thresholds are defined in amplitude and resolution of the quantiser is reduced in two steps (with a bit in each step) when the signal crosses the preset thresholds.

Fig.3.9 shows the number of output bits generated by the different ADCs for a sinusoidal input signal at 400 Hz over a duration of 10 seconds. Similarly, Fig.3.10 shows the amount of data generated by the ADCs for an example AEG signal over a duration of 10 seconds. The estimation illustrates that the adaptive resolution technique produces lower amount of data compared to the level crossing sampling and the split resolution techniques. However, considering the overhead due to the addition of slope information or change of resolution the amount of data increases considerably, even more than that of a normal LC ADC.

The three methods for implementation of the split resolution (SR) technique are implemented in the VerilogA model of the fixed wave LC ADC and their operation is verified. The reconstructed signal from the output sample of the model with SR technique applied in the digital output block is shown in Fig.3.11.(a). The output pulses are removed in steps, as shown in Fig.3.11.(b). Similarly the waveforms of other methods are shown in Fig.3.12.

The implementation of SR technique in the input amplifier introduces sudden changes in the output of the DAC. This also affects the output pulses in the comparators. The other two methods do not cause such disturbances as they do not affect the input signal directly and are easier to implement as well. Hence, they are opted to be implemented in the LC ADC design.

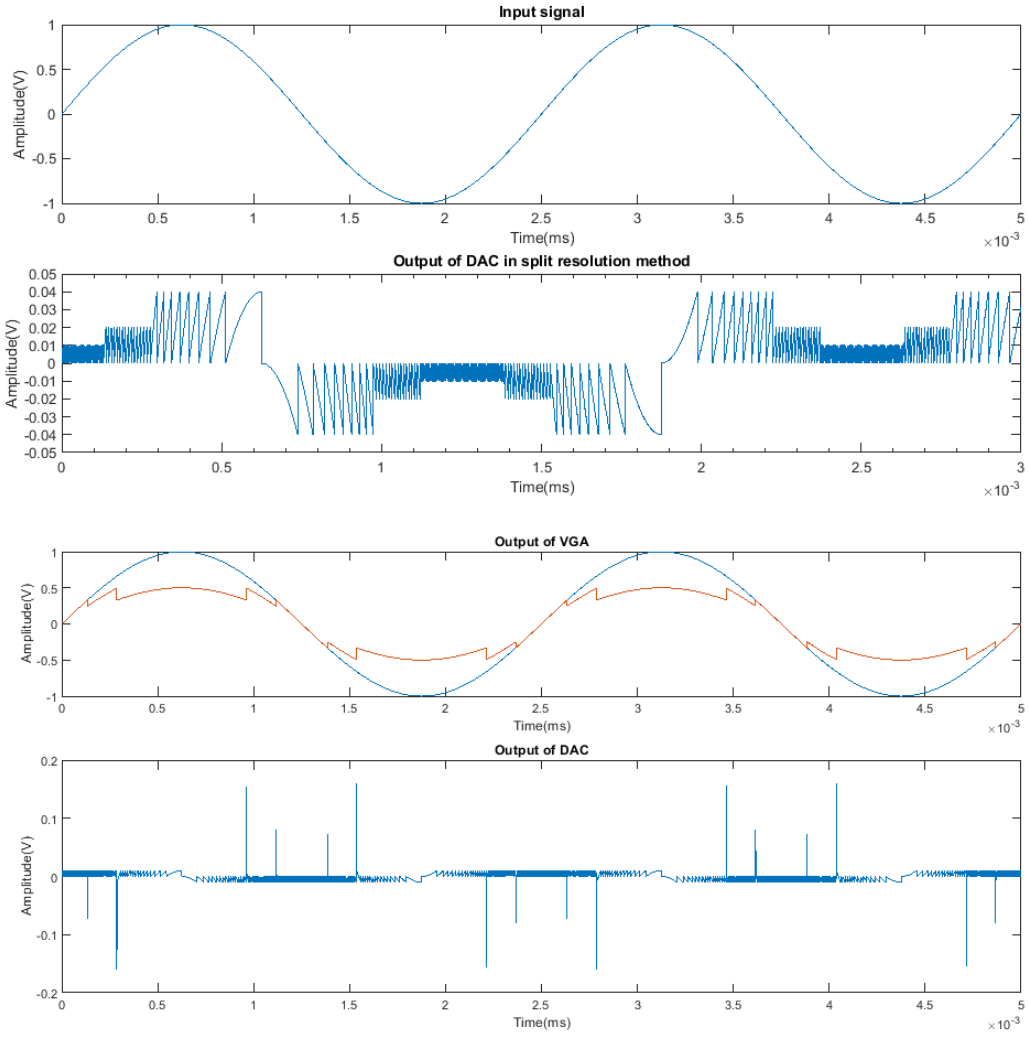


Figure 3.12: (a) waveform illustrating operation of SR technique implemented in the DAC (b) waveform illustrating SR technique applied in the variable gain amplifier.

3.4. Architectures Considered for Multichannel LC ADC

Several architectures and design ideas are considered to design the multichannel LC ADC. The system level design considerations of all the architectures considered are explained below -

3.4.1. Spatially Correlated Sampling

AEG signals obtained from EMC are studied in order to find signal properties which could be exploited in sampling and quantising the signal. Since the recording patch consists of electrodes at regular intervals, the electrical impulse propagating over the surface of the atrium would be observed as a wave across the electrodes. This property is observed in the signal analysis of the recorded AEG signals, as shown in Fig.3.13. Hence, this property is considered for designing a difference-based cluster of ADCs with a single high-resolution ADC and several low-resolution ADCs for the adjacent channels. The intention is to use the high-resolution ADC to obtain a high-resolution recording from one channel and use that as a reference for recording the difference in signals from the adjacent electrodes. However, the problem with this approach is that even though the signals from adjacent channels have a similar pattern they have different amplitudes which cannot be resolved through a direct correlation with the reference signal. Moreover, the correlation among the signals from adjacent channels are not always the same, even though a pattern could be observed. Hence, this approach can not be implemented without using overly complicated algorithms to estimate the difference of signals between adjacent channels and thus may not be practically viable.

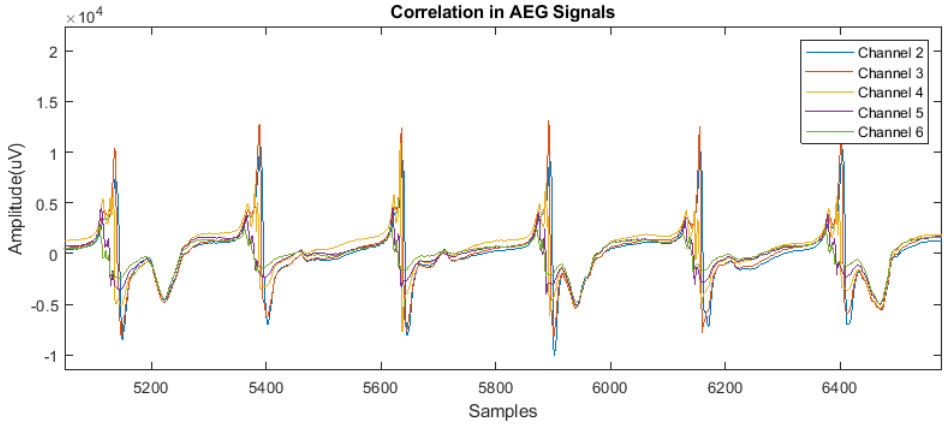


Figure 3.13: Illustration of correlation in AEG signals from adjacent channels.

3.4.2. Time-Multiplexed LC ADC

Another approach considered is to time-multiplex the input channels as is done in multichannel SAR ADCs [27]. A specific period of time is provided to each channel within which the level crossing has to be detected. The architecture is shown in Fig.3.14. In this architecture the feedback to each channel is also synchronised with the counter through a DEMUX block. Each channel has a DAC which tracks the signal and affects the input signal according to its corresponding feedback signal. This method forces the asynchronous LC ADC to behave synchronously, since the level crossing can now be detected at specific time points only. The clock thus needs to be fast enough so that it does not miss a level crossing in any of the channels. This means that it needs to be able to switch to each channel before two consecutive level crossings have occurred for that channel. The clock speed required to ensure that all level crossings are detected for every channel can be calculated by Eq.3.2.

$$\begin{aligned} Fs &= 1/((Vdd/2^M) * (1/(2 * \pi * Vdd * BW)) \\ &= 1/((1.8/2048) * (1/(2 * \pi * 1.8 * 400)) \\ &= 5.147MHz \end{aligned} \tag{3.2}$$

In this architecture the signal bandwidth(BW), peak-to-peak amplitude (Vdd) and number of quantisa-

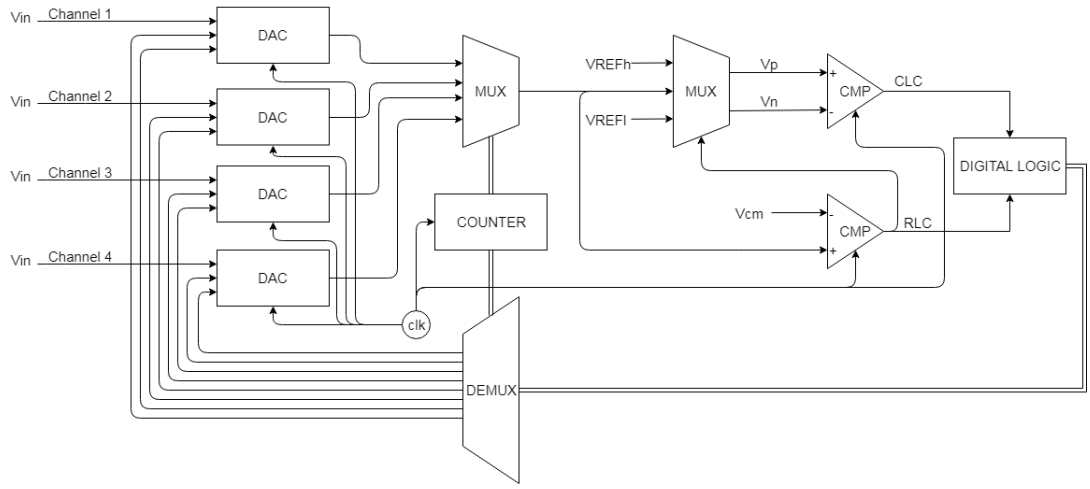


Figure 3.14: Architecture for time-multiplexed LC ADC for 4 channels.

tion levels(2^M) are considered to calculate the required clock speed. It should be noted here that the clock speed is dependent not only on the signal bandwidth but also on the slope of the signal. At the maximum slope of the signal the least amount of time would be taken for it to cross adjacent reference levels. Hence, the multiplicative factor 2π signifies the maximum slope of the signal. Using the same relation for the current scenario the minimum clock speed required would be greater than 5MHz. This calculated clock speed is required per channel. Hence, for an n-channel multiplexed ADC the clock speed would be $n * 5MHz$.

Even at this clock speed, the maximum error in timing resolution that can be encountered is almost equivalent to 1 LSB. This is due to the fact that the minimum time taken by the signal between two consecutive level crossings is considered to calculate the clock speed of the timer. If a level-crossing occurs in a channel immediately after it was checked for a level-crossing, it would be detected the next time this channel is checked, i.e. after the calculated minimum time. This problem is illustrated in //Fig. This problem can be mitigated by increasing the timer frequency.

3.4.3. Asynchronous Multiplexed LC ADC

In the approach considered above, the time-multiplexing approach is found to be unsuitable as then the timer frequency required is substantially high and thus defeats the continuous-time property of the LC ADC. Hence, a different approach is sought to design the multichannel LC ADC which does not involve synchronising the operation with a clock. The architecture in Fig.3.15 is considered to implement this. In this approach the same multiplexing technique is retained while removing the clocked counter for selection of the channels. An asynchronous triggering mechanism is used instead of the clocked counter. The trigger acts as an interrupt which retains access to the ADC until the Level Crossing event is fully registered as an output pulse by the ADC. The triggering mechanism is intended to be engaged when a level crossing event is about to happen.

A Winner Take All (WTA) circuit [28] shown in Fig.3.16 can be used in this approach for triggering the interrupt and for selecting the channel. A WTA circuit consists of several branches through which similar magnitudes of current flow and it selects the branch with the highest magnitude of current. In the circuit if the currents (I_{in1} and I_{in2}) are identical, then the output currents I_{out1} and I_{out2} are also identical as the biasing conditions for both cells are the same. If there is a difference in the magnitude of input currents such that $I_{in1} > I_{in2}$, the output voltage difference increases such that $V_{out1} > V_{out2}$. Thus, as V_{out2} decreases, M_2 shuts down and thus I_{c2} reduces to 0. This causes all the bias current to flow through M_{1b} and therefore $I_{c2} = 2.I_{bias}$. In this case cell 1 has won the control as it had more input current.

Designs of WTA circuits reported in the literature show that differences of upto nA can be differentiated in WTA circuits[29]. In this architecture the voltage in DACs can be tracked in the WTA circuit and the one

with the highest voltage is selected as its corresponding channel would have the level crossing first. When the level crossing occurs and it is detected in the comparators then offset injection occurs in the DAC and the next channel with the highest voltage is then selected. Thus, the selection of channels is done entirely asynchronously.

However, the WTA circuits consume a considerable amount of power, which even exceeds the power consumption of a single channel LC ADC reported in literature [9]. Hence, the benefits obtained in avoiding synchronisation of the multiplexer are far outweighed by the cost of power consumption of the WTA circuit. The speed of operation of WTA circuits is inversely proportional to the number of branches it tracks on. This implies that with higher number of channels the speed of the WTA circuit is also affected. Even though the multiplexing can be achieved asynchronously, a timer is still required that can resolve the timing of generation of samples accurately. Moreover, additional bits would be required to specify which channel generated the level-crossing sample. Hence, overhead in the amount of data generated would also occur.

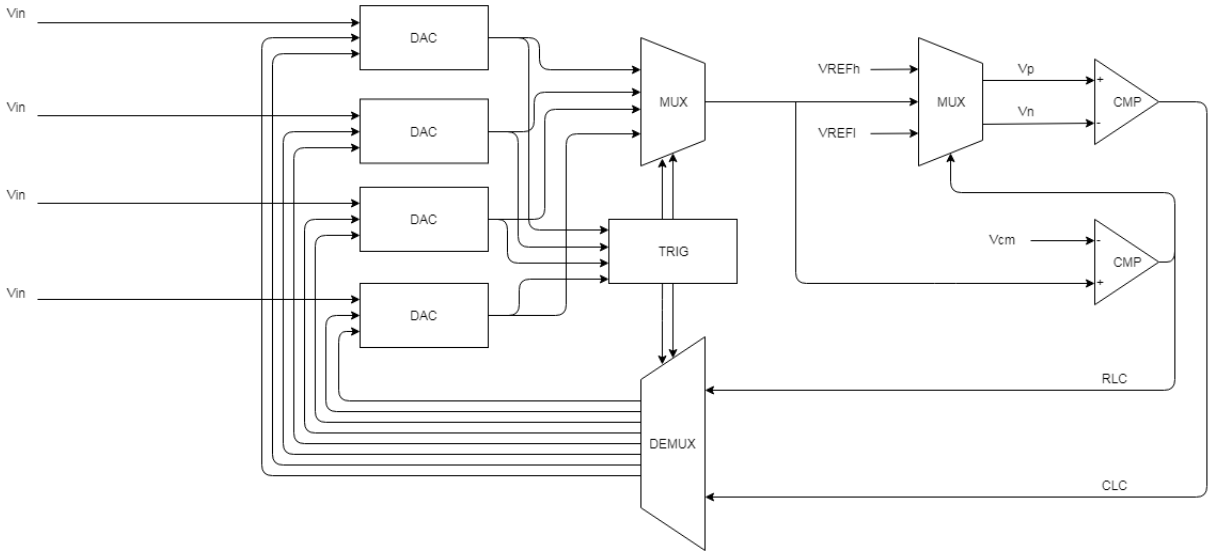


Figure 3.15: Architecture for asynchronously multiplexed LC ADC for 4 channels.

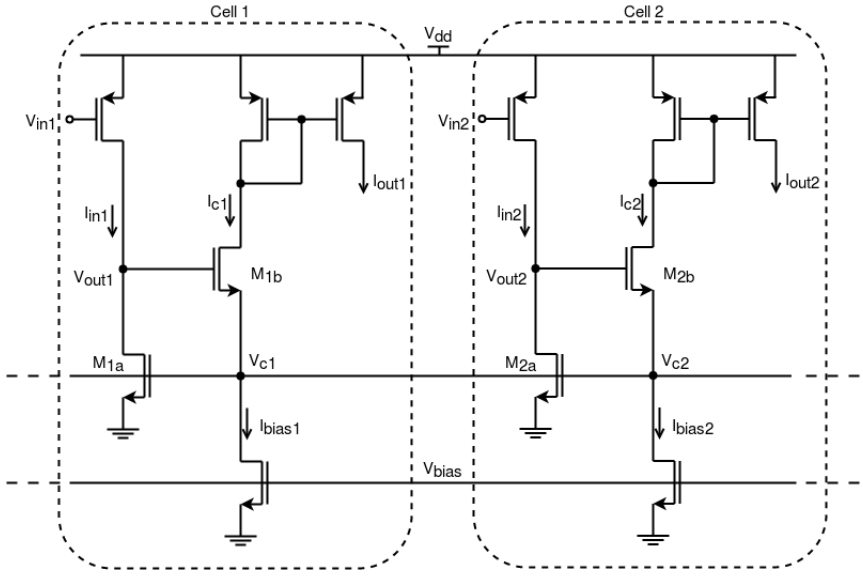


Figure 3.16: Schematic of Winner Take All (WTA) circuit implemented in CMOS.

3.4.4. Encoding Output from Multiple Channels

The signal acquisition device can be made with a single LC ADC for each channel. Since LC ADCs are quite small (0.045mm2) [9], this method is possible to be implemented in an IC. However, if 192 ADCs are put in the IC, it would lead to twice as many output wires as well. Considerable amount of area and logic would be required to route the wires to a digital storage/transmission system and to synchronise the data, respectively. This problem can be resolved by combining the output wires in a digital logic and streaming the output pulses from several channels together through a single wire/port as shown in Fig.3.17. This can be implemented through an asynchronous circuit.

The challenge in designing this block would be in handling the simultaneous output pulses from multiple channels. Memory buffers can be used to handle signals coming in several channels together. It is difficult to use an asynchronous digital block to generate an output stream of data as a clock is needed to synchronise the data rate. Otherwise the receiver block would not be able to process the data sent from the ADC reliably. Thus a clock signal is required to ensure a constant baud rate which can be synchronised with the receiver before storage in a memory device. The clock signal to be used here should be as fast as the output generation rate of the ADC. Since the output generation rate of the ADC is dependent on the slope of the input signal and quantiser resolution, it should be faster than the maximum number of samples generated per second. This calculation can be related to 3.2 which is equivalent to 5.14MHz for the given scenario. Even in this method extra bits would be required to specify the channel generating the sample, thus increasing the total amount of data generated.

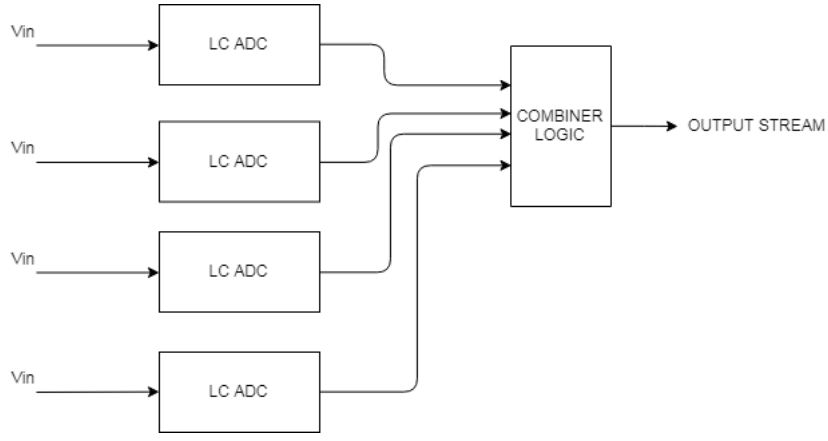


Figure 3.17: Architecture for combination of output wires in a Digital Logic block.

3.4.5. Coarse-Fine Architecture

The coarse-fine architecture is a popular architecture used in synchronous ADCs. In this architecture a high-resolution ADC is composed of two medium/low-resolution ADCs which give a combined output of the desired resolution. The residue of the signal quantised by the 'coarse' ADC is amplified and fed to the 'fine' ADC to quantise. Thus the MSB bits are derived from the coarse ADC while the LSB bits are derived from the fine ADC. This approach reduces the stringent requirements on the individual components of a high-resolution ADC.

The design constraints of the LC ADC become tighter for every bit added to the resolution of its quantiser. With every additional bit the required speed of the timer doubles and the amount of data produced doubles as well. Moreover, the halved LSB voltage requires more accuracy in matching of the DAC, comparator etc. Hence, a coarse-fine architecture can be a good solution for the LC ADC in this case. Two LC ADCs with lower resolution can be cascaded to make a higher resolution LC ADC.

However, a regular coarse-fine architecture also relies on the fact that the signal being quantised in both the ADCs is a sampled signal. Since the LC ADC operates on the continuous-time signal, the residue of the input signal from the coarse ADC would still be in continuous time. This can be sampled in the fine LC

ADC as is done in the coarse LC ADC. However, since the conversion in the fine LC ADC needs to be completed before the next conversion in the coarse LC ADC, the clock frequency for driving the fine LC ADC needs to be as fast as the minimum time between the level crossings of the signal. The data generated in total from both the ADCs would be the same as that of the single LC ADC with the same overall quantiser resolution. Hence, this approach helps in relaxing the design of the analog components but doesn't aid in reducing the required clock speed or amount of data generated.

3.4.6. SYNCHRONOUS LC ADC

As observed in the methods discussed above, the continuous time sampling approach is not suitable to implement in multichannel configurations. Hence, a different approach needs to be adopted in which the sampling is performed synchronously and the quantisation is performed based on the level-crossing approach. In this method the sample is quantised by counting the number of LSB steps required by the quantiser to reach the sample amplitude from mean (half of the common mode input range). However, this approach is rather inefficient as in the worst case the quantiser would need to count 2^{M-1} LSB steps to reach the sample amplitude. Moreover, if the signal is stagnant at a high amplitude for considerable period of time then a large number of quantisation steps would be required for each sample.

This problem can be mitigated by using the previous sample as the starting point for quantisation of each new sample. Instead of counting the difference of the sample amplitude from the mean in terms of number of LSB steps, the difference in amplitudes of the current sample and the previous sample could be counted instead. Since the level crossing sampling scheme allows the quantiser to remember the exact amplitude level of the input signal at each moment, this property can be used to implement the proposed scheme. The operation of the proposed method is illustrated in Fig. 3.18.

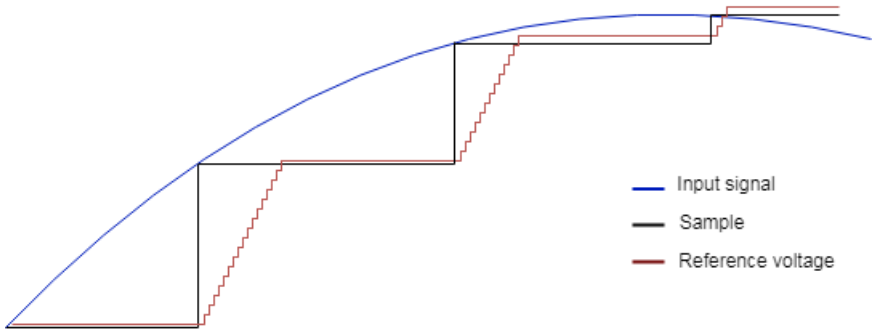


Figure 3.18: Operation of Synchronous LC ADC.

Conventional synchronous ADCs provide the quantised value of the sample at the output. Thus, an n-bit ADC provides an n-bit output for every sample. Since the proposed LC ADC is activity dependent, its output should also adhere to this property. Hence, the output of the proposed LC ADC denotes the difference in amplitude from the previous sample in terms of number of LSB steps counted by the quantiser. This quantity is encoded as a binary number along with a separate bit for direction. The maximum number of LSB steps that can be covered by the quantiser in the proposed ADC is 2^M. Hence, a maximum of $M+1$ bits would be given at the output for the worst case. It should be noted that for most samples this worst case condition would not apply, since the difference in amplitude between successive samples of biosignals is rarely as high as $Vref$.

The proposed method will improve the performance of the ADC considerably, especially for temporally sparse signals such as ECG, AEG etc. These biosignals don't change in amplitude for considerable periods of time. The number of LSB steps covered for each sample would therefore be quite less. While the SAR algorithm uses the same number of quantisation steps for each sample irrespective of the signal activity, this approach would make the quantisation process entirely dependent on the signal activity. If the signal does not change in between samples then the proposed level-crossing quantiser would be able to deter-

mine this in just one quantisation step. As can be observed in Fig. 3.19 most of the high-amplitude content of the AEG signal appears in the low frequency region. Hence, the average number of quantisation steps required for most samples would be low. The amount of data generated is related to the number of LSB steps covered. Thus the average amount of data produced would also be lower as compared to conventional synchronous ADCs. This is also lower than the output of the time-multiplexed LC ADC architectures proposed previously, in which every level crossing generates data.

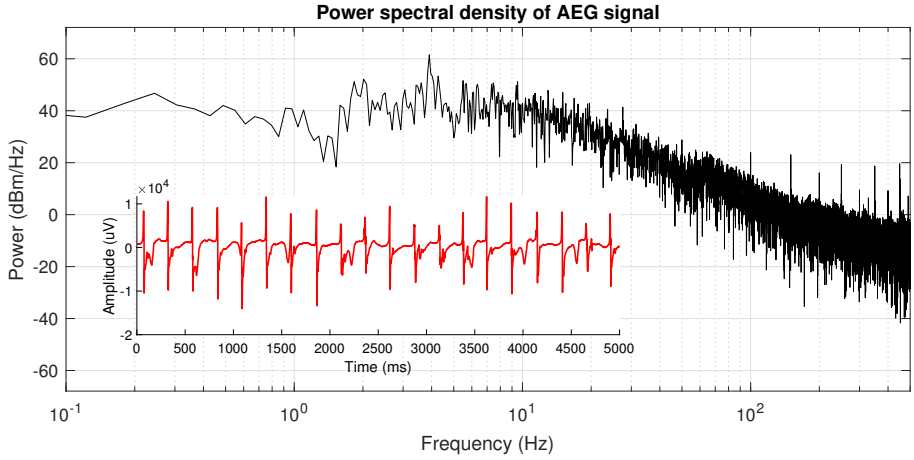


Figure 3.19: Power spectrum of an AEG signal (inlay: an AEG signal).

Conventional LC ADCs require two comparators to determine the level-crossing. Since the proposed ADC is based on the level-crossing sampling approach, two comparators were considered at first. However, this implies that two DACs are required to generate the two reference levels. A better approach is to use a single comparator and perform the quantisation with one changeable reference voltage. The reference voltages of the window can be loaded in subsequent steps and compared with the sample. The block diagram of the proposed synchronous LC ADC is shown in Fig. 3.20.

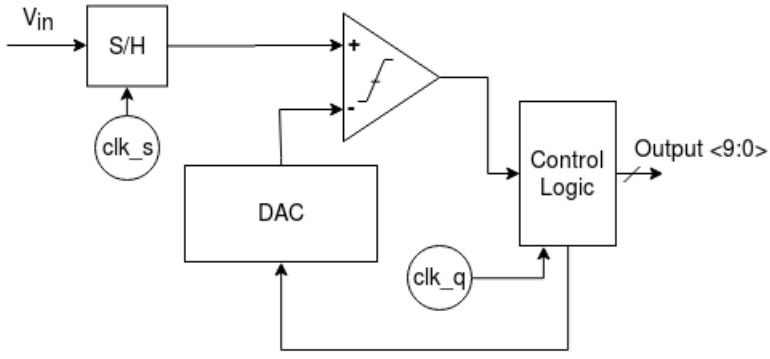


Figure 3.20: Block diagram of the proposed synchronous LC ADC.

The flowchart representing operation of this method is shown in Fig. 3.21. Every new sample is compared with the highest reference level reached for the previous sample. Based on the output of the comparator the direction of counting in the quantiser is determined. A new reference level is loaded and the compared with the sample until the output of the comparator is opposite to that of its output for the first comparison. E.g. Let the previous sample be at 22.5 mV and the highest reference level thus determined to be 23 mV (for 1 mV quantiser resolution) and the current sample be at 47.2 mV. Then the output of the first comparison will be high, showing that the amplitude of the sample is greater than the reference level. Hence, the quantiser will increase the reference level by 1 LSB step i.e., 1 mV and compare again. This process

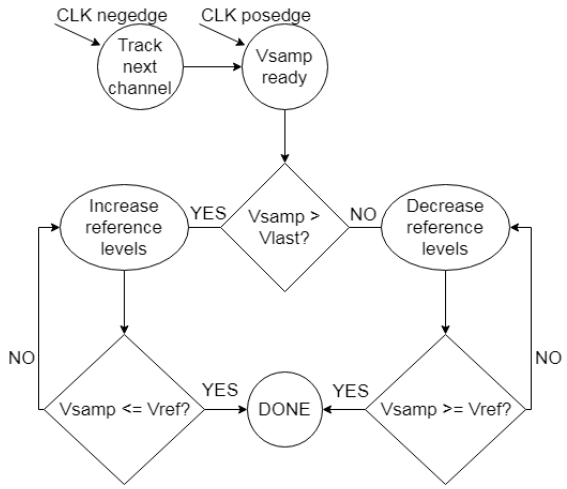


Figure 3.21: Flowchart of operation of proposed synchronous LC ADC.

is repeated until the comparator output is low, signifying that the reference level is now greater than the sample. The quantisation process is now complete and the difference in amplitude (i.e. +23) is given as output from the ADC. The format of output of the ADC is discussed in the next chapter.

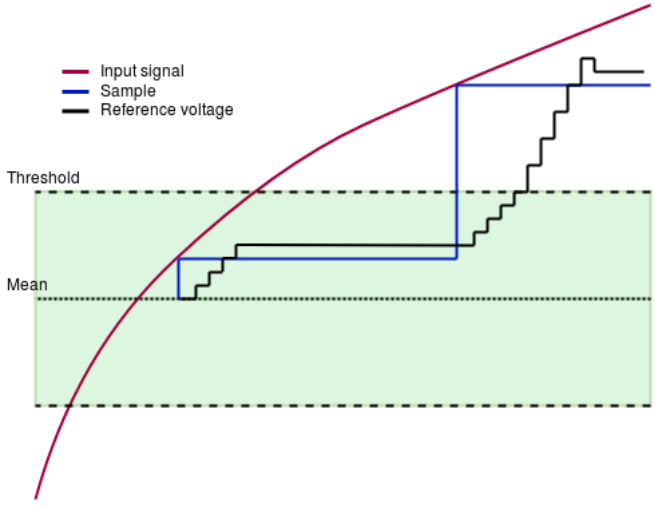


Figure 3.22: Illustration of operation of split resolution technique in the synchronous LC ADC.

The quantisation process in the proposed method is synchronised by a clock signal. Unlike the conventional SAR algorithm in which the quantisation process has a fixed number of steps for each sample, the proposed method has a variable number of quantisation steps depending on the input signal activity. Hence, the clock signal frequency must be able to accommodate the highest number of steps required for quantisation of the input signal. In the worst case the number of steps required could be 2^M, if the difference between two successive samples is equal to the maximum amplitude. However, as discussed above, this condition would not occur for sparse signals such as AEGs, ECGs etc. The maximum number of steps required for quantisation of the signal can be estimated through modelling of the synchronous LC ADC while using typical biosignals as input.

The number of quantisation steps required for each sample in the proposed method can be reduced by introducing the split resolution technique. As described previously the rapid changes in signals are related to peaks which reach high amplitudes. Hence, for samples with amplitudes outside a specific (programmed) amplitude range, the resolution for quantisation can be reduced while quantising beyond the preset am-

plitude range. This method introduces an error of +/- 1 LSB as sample amplitude may be within +/- 2 LSB of the final reference level. This error can be corrected by adding another step in which the quantiser reduces the reference voltage level by 1 LSB (instead of 2 LSBs as used in the split resolution technique) and compares again with the sample. The sample is then quantised accurately. This process is illustrated in Fig. 3.22.

3.5. Comparison of Architectures

Several architectures can be considered to implement multi-channel configuration of level-crossing sampling. However, each architecture exhibits some advantages and disadvantages which affect the performance and cost of operation or design. Hence, a qualitative analysis was performed to determine the best architecture for implementation of the multi-channel LC ADCs. The advantages and disadvantages of each architecture are listed in Table 3.3.

The asynchronous sampling paradigm is incompatible with the synchronicity inherent in time-multiplexed systems. Even if synchronicity is introduced into the asynchronous ADC, it degrades the performance while increasing the cost of operation (higher clock speed). This problem was observed in the time-multiplexed LC ADC architecture, the output combiner architecture and the coarse-fine architecture. While circuits such as WTA can resolve this problem, they too increase the cost of operation (higher power consumption). Hence, the asymchronous sampling approach needs to be replaced by the synchronous sampling approach.

Combining level crossing quantisation with synchronous sampling is a good approach to combine the best features of both synchronous and asynchronous recording methods. The synchronous LC ADC architecture as shown above implements this approach. This architecture has activity-dependent operation and activity-dependent output. It uses a single dynamic comparator instead of two continuous-time comparators and thus decreases the power consumption considerably. It can be scaled for higher sampling rates and higher number of channels easily. Moreover, it is not troubled by most of the problems faced by the other architectures. Hence, this architecture is found to be the optimum architecture that meets all the requirements of a multi-channel LC ADC. The performance of the synchronous LC ADC and its advantages over other ADC architectures are quantified through modelling. The method of modelling and the results are discussed in the next chapter.

3.6. Conclusions

The adaptive resolution technique as discussed in the previous chapter suffers from data overhead which increases the total amount of data generated in the LC ADC. Thus, Split Resolution technique was introduced that solves the data overhead problem in adaptive resolution techniques. The various methods for implementation of the split resolution technique are also discussed along with model simulations.

Several architectures can be considered to implement the multi-channel LC ADC. As the level-crossing sampling implements asynchronous sampling while time-multiplexed systems are inherently synchronous, they are not compatible. Thus, a novel approach is introduced in which sampling is performed synchronously while quantisation is performed according to the level-crossing quantisation method. This makes the operation of the quantiser activity-dependent. Moreover, the output data generated is also activity-dependent. The modelling and model simulations of the proposed syncrhonous LC ADC architecture are discussed in the next chapter.

Table 3.3: Comparison of Architectures.

Architecture	***Advantages***	***Disadvantages***
Spatially Correlated Sampling	• Exploits correlation of signal between adjacent channels	• Complex circuit design • Degraded performance of ADC due to large differences in multi-channel input signals
Time-multiplexed LC ADC	• Simple circuit design • Dynamic comparators are used instead of continuous-time comparators	• Very fast quantisation clock signal required • Operation is no longer activity dependent
Asynchronously Multiplexed LC ADC	• Activity-dependent operation	• High power consumption in WTA circuit
Multichannel Output Encoding	• Reduced number of output wires	• Very fast clock signal required to synchronise output • Minimal reuse of ADC components
Coarse-Fine Architecture	• Less constraints on individual LC ADC design	• Not activity dependent • Very fast quantisation clock signal required
Synchronous LC ADC	• Activity dependent operation • One dynamic comparator used • Output data is activity-dependent	• Fast quantisation clock signal required • Complex circuit logic required

4

PROPOSED MULTICHANNEL LC ADC

The proposed multichannel synchronous LC ADC was introduced in the previous chapter along with its major features. This chapter discusses the models used to test the functionality of the proposed ADC at the system level and to quantify the performance of the ADC as compared to conventional synchronous ADCs.

4.1. MODELLING OF SYNCHRONOUS LC ADC

The VerilogA model of the LC ADC as shown previously operates asynchronously and converts the analog input signal into digital form in the continuous time domain. However, the proposed synchronous LC ADC samples the input signal synchronously and then quantises the sample using level-crossing approach. The operation is different from that of the asynchronous LC ADC. Hence, the model is modified accordingly.

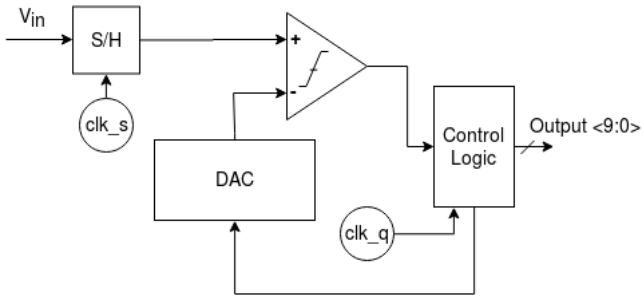


Figure 4.1: Block diagram of synchronous LC ADC.

The VerilogA model of a single channel LC ADC shown previously is modified to operate synchronously while a clock input is also provided to the ADC. The VerilogA code of the model can be found in Appendix-A.1. The block diagram of the model is shown in Fig.4.1. A Sample and Hold block is added to hold the sample while it is quantised. The sample is compared with the quantised reference voltage of the previous sample. Otherwise the reference voltage is changed in steps of 1 LSB each until the amplitude of the sample is crossed by the reference voltage. The number of steps moved in either direction are counted and given as output for that sample. The output of the comparator is checked in each iteration to determine the end of the quantisation process. This reference voltage is the starting point for quantisation of the next sample.

The model is simulated with a sinusoidal input signal at 400 Hz. The input signal has a peak-peak amplitude of 1.2 V. The quantiser is configured with LSB step size of 4.6875 mV. Thus, the quantiser quantises the sample at 8 bits of resolution. The quantisation process is synchronised with a 1 MHz quantisation clock signal. The sampling is performed at 1 kS/s. The quantisation clock frequency used here is much

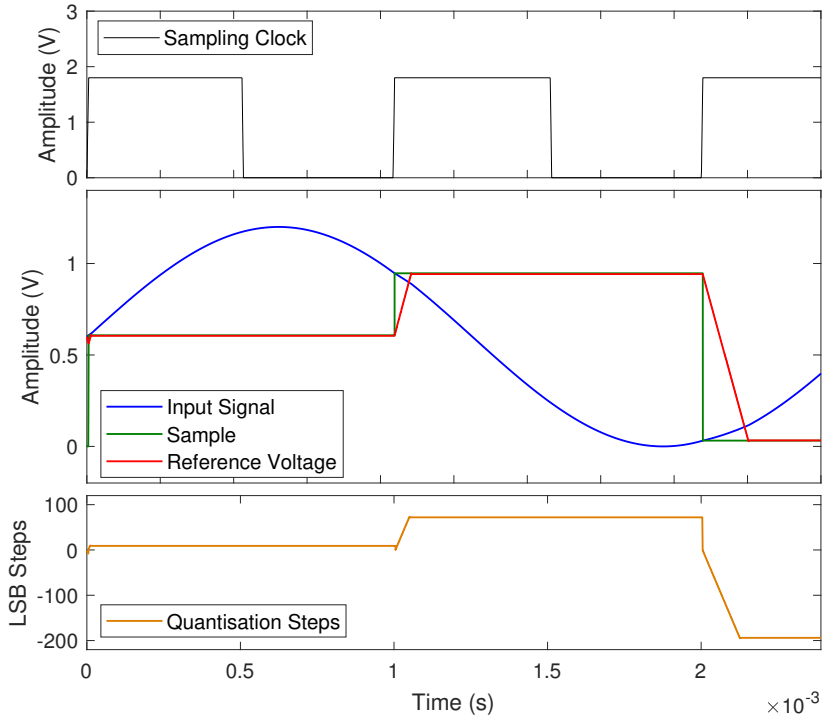


Figure 4.2: Operation of VerilogA Model of Synchronous LC ADC.

higher than required. The method of estimation for the required quantisation clock frequency for a particular input signal will be discussed later. The input signal, the sample captured in the S&H block, the quantised value and the ADC output are shown in Fig. 4.2.

4.2. Estimation of Quantisation Clock Frequency

In conventional synchronous ADCs the quantisation process is synchronised with a specific clock signal. The quantisation clock frequency is determined on the basis of the quantisation method. For instance, the SAR algorithm requires 'n' clock cycles to complete the quantisation for each sample, where 'n' is the quantiser resolution. Thus, the quantisation clock frequency for a SAR ADC would be -

$$F_{q,SAR} = n.Fs \tag{4.1}$$

where Fs is the sampling rate of the SAR ADC. However, in the proposed synchronous LC ADC architecture, the quantisation process is not fixed to a specific number of quantisation steps. The number of quantisation steps required is dependent on the signal activity. Hence, the quantisation clock frequency must be set according to the worst case, i.e. the most number of quantisation steps required for a sample of the input signal.

Ideally, the worst case would denote V_{max} or the common mode input range. The quantiser would then require $2^M + 1$ clock cycles to complete the quantisation. However, as discussed in the previous chapter, the AEG signals have higher amplitudes at lower frequencies. Hence, the worst case condition would never occur but the maximum number of quantisation steps required for a specific signal needs to be estimated. This can be performed in a MATLAB model in which the synchronous LC ADC's quantisation process is modelled and the input signal is sampled (Appendix-A.3). The number of quantisation steps required for each sample are then calculated, and the maximum value is used to calculate the frequency of the quantisation clock. A margin is added to the estimate to ensure error-free operation. The estimate of number of quantisation steps required for samples a typical AEG signal is shown in Fig. 4.3. As can be observed in the plot, the maximum number of quantisation steps required for the AEG signal sampled at 1 kS/s is 100. 2 clock cycles are required for each quantisation step. Hence, 200 clock cycles are required for the

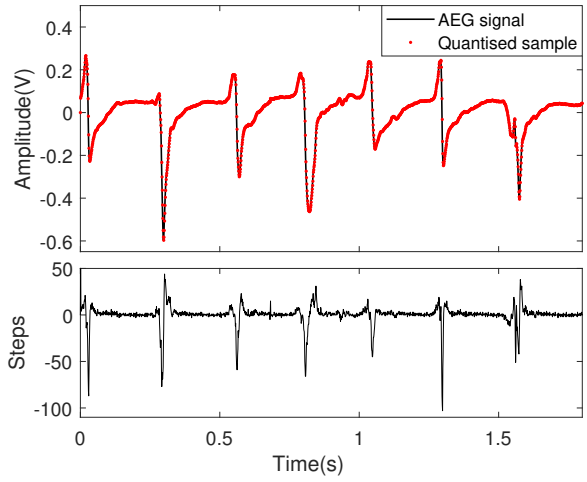


Figure 4.3: Estimation of number of quantisation steps required for an AEG signal. The plot above shows the input AEG signal and the quantised samples. The plot below shows the number of quantisation steps required to complete the quantisation for each sample.

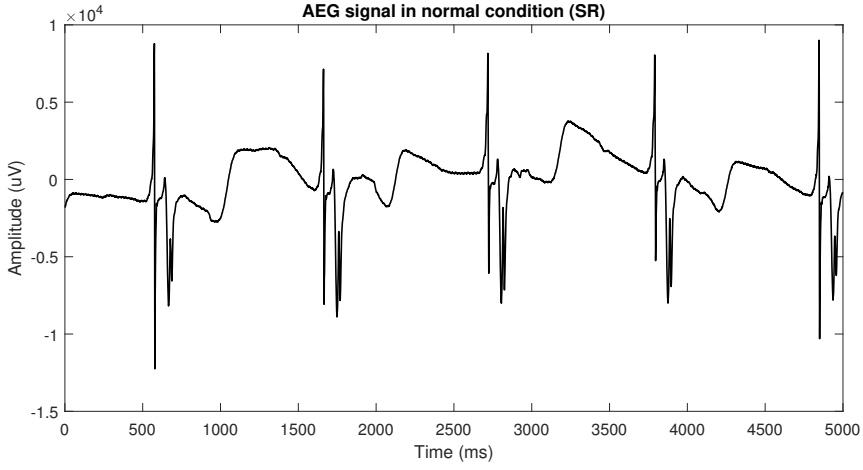
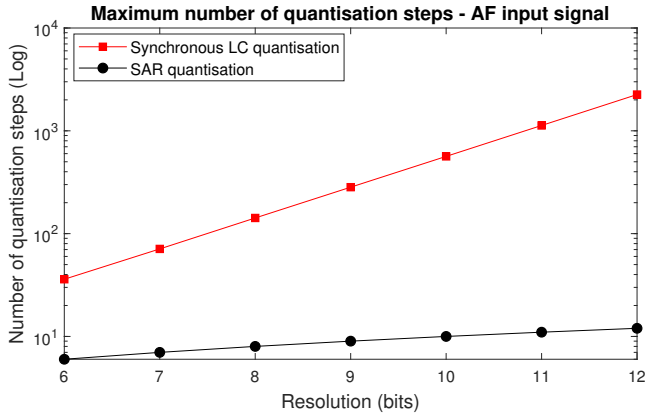


Figure 4.4: Example of atrial electrogram in sinus rhythm (SR) condition.

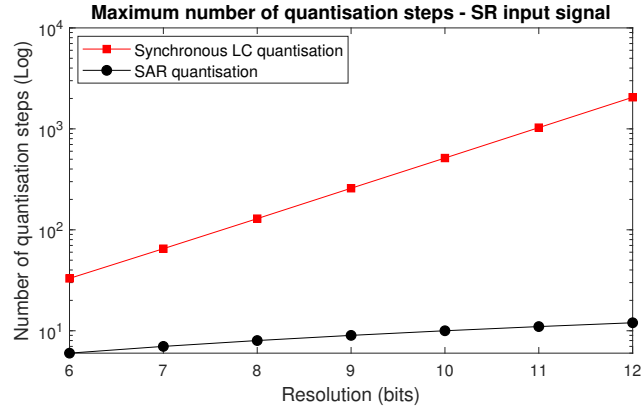
quantisation of the sample. The quantisation is completed in half the time period of the sampling clock signal. Hence, if the sampling rate is 1 kS/s then the quantisation clock frequency needs to be:

$$F_{q,SLC} = 1/(500e-6/(200)) = 400kHz \tag{4.2}$$

The maximum number of quantisation steps required for quantisation of AEG input signals with induced AF condition and with normal SR condition are determined using the model for different quantiser resolutions. The results are shown in Fig. 4.5. The example of an AEG signal with AF condition was shown in Fig. 1.3. The example of an AEG in sinus rhythm i.e. normal condition is shown in Fig. 4.4. As can be seen from the results the maximum number of quantisation steps required for quantisation of AEG signals is much higher in the proposed synchronous LC ADC than in the conventional SAR ADC. Moreover, the results are similar for both AEG signals (with AF condition and with SR condition). It should be noted however that the maximum number of quantisation steps is required in the synchronous LC ADC for very few samples and not for every sample. It can also be observed that the maximum number of quantisation steps required is doubled for every extra resolution bit as expected.



(a)



(b)

Figure 4.5: Maximum number of quantisation steps required for (a) atrial electrogram with induced AF (b) atrial electrogram in sinus rhythm, determined in MATLAB model simulations.

4.3. Activity-Dependent Quantisation

The quantisation process of the proposed synchronous LC ADC is dependent on the activity of the input signal, as illustrated in the model simulations above. While conventional synchronous ADCs use the same number of quantisation steps for each sample, the quantisation process in the synchronous LC ADC is variable.

Also, since the AEG signals have higher amplitudes at lower frequencies, the average change (in number of LSB steps) between samples is low. This property is observed by determining the average number of quantisation steps required per sample of the AEG signal. This is achieved through the MATLAB model added in Appendix-A.3. In this model the total number of quantisation steps required for a number of samples is determined. Then the average number of quantisation steps required per sample is calculated by dividing the total number of quantisation steps by the number of samples considered. The simulation results are shown in Fig. 4.6. The total number of quantisation steps required for 9999 samples of AEG signals is much lower in the synchronous LC ADCs than in the conventional SAR ADC, especially for the lower quantiser resolutions. The average number of quantisation steps required for 8-bit quantisation an AEG signal is 2.37 whereas in a SAR ADC with an 8-bit quantiser it is 8. Thus, the synchronous LC ADC achieves a 3.37x reduction in number of quantisation steps for 8-bit quantisation of AEG signals. The AEG signal in SR condition has lower amount of activity as compared to the AEG signal in AF condition. This difference can also be observed in the number of quantisation steps estimated in the model, especially at higher resolutions. The reduction in number of quantisation steps is different for different quantiser resolutions. At higher quantiser resolutions the number of quantisation steps required increases and would also be higher than that in a conventional SAR ADCs. However, for most biosignals lower quantisation

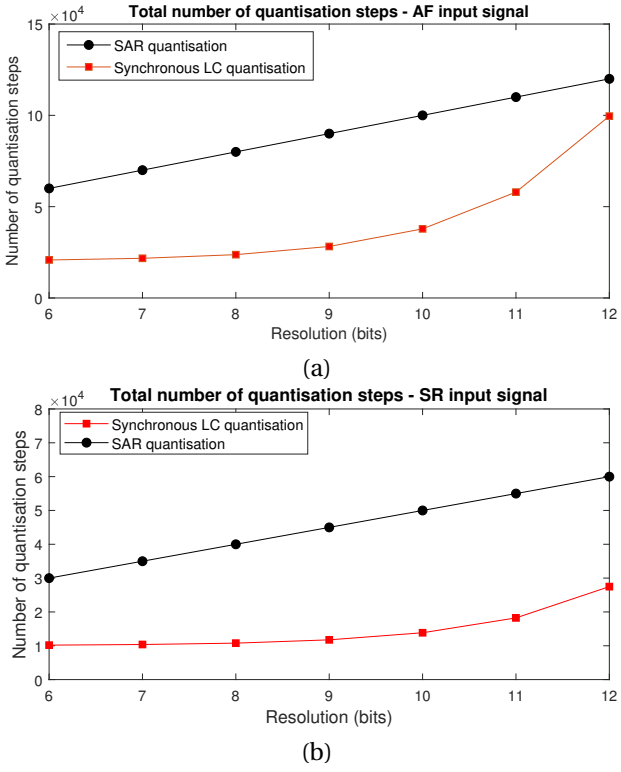


Figure 4.6: Total number of quantisation steps required for (a) atrial electrogram with induced AF (b) atrial electrogram in sinus rhythm, determined in MATLAB model simulations. The AEG signal with AF has 9999 samples and the AEG signal with SR has 4999 samples.

resolutions are sufficient and hence can exploit this advantage of synchronous LC ADCs.

Conventional LC ADCs use continuous time comparators that draw power continuously. Hence, the proposed synchronous LC ADC architecture has an advantage over conventional LC ADCs as it uses just one dynamic comparator instead of two continuous time comparators. Each comparison operation draws a certain amount of energy [30]. Moreover, each time the digital logic block performs a quantisation step, energy is consumed. Hence, by decreasing the number of quantisation steps required, the total energy consumption can also be reduced.

4.4. Activity-Dependence of Output

The output of the proposed synchronous LC ADC architecture is the difference in amplitude between successive samples. This is different from conventional synchronous ADCs which give the quantised value of the sample at the output for every sample. The output in conventional synchronous ADCs thus has the same size for every sample. On the other hand, in the proposed synchronous LC ADC architecture, the number of bits in the output varies according to the activity of the input signal. If the input signal is relatively constant across several samples, then the total output generated would be less for each sample.

The structure of the output generated for each sample is shown in Fig. 4.7. The output bits convey two pieces of information. The LSB bit indicates the direction in which the sampled signal has changed and the rest of the bits indicate the number of LSB steps covered by the quantiser to perform the quantiser for the sample. If the sample has changed by just 1 LSB step, then only one output bit is generated to indicate the same. Thus, the number of bits generated for each sample is different. As can be inferred from the previous sections, since the activity of AEG signals is low for considerable periods of time, the average number of output bits generated per sample in the synchronous LC ADC would also be low.

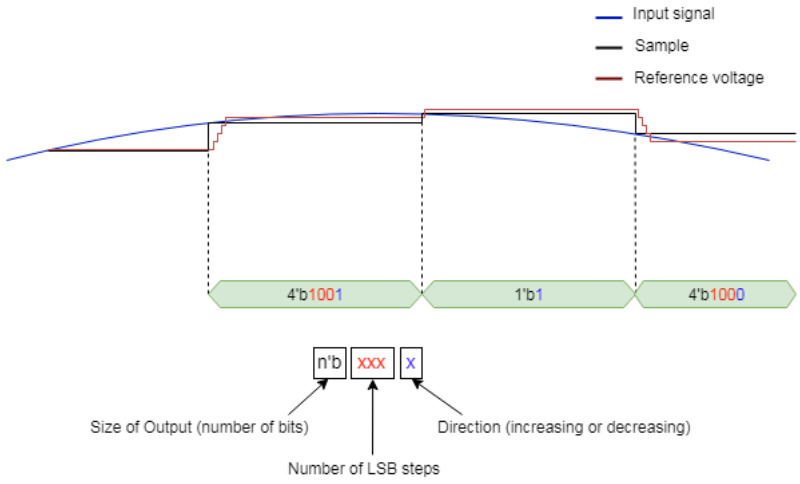


Figure 4.7: Quantisation process in synchronous LC ADC and structure of output bits from the ADC. The size of the ouput is shown for indication; it is not part of the output bits.

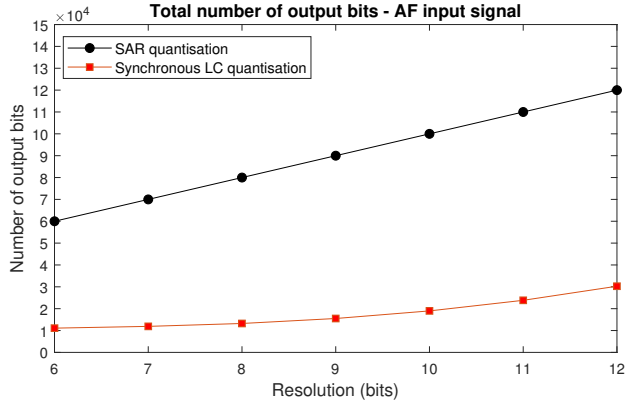


Figure 4.8: Total number of output bits generated for quantisation of atrial electrogram with induced AF condition, determined in MATLAB model simulations. The AEG signal with AF has 9999 samples

This assumption was tested by determining the number of output bits generated per sample for an AEG signal for different quantiser resolutions and the average number of bits generated per sample was also determined. The results are shown in Fig. 4.8 and Fig. 4.9. The results show that the average number of output bits generated per sample in a synchronous LC ADC with an 8-bit quantiser is 1.32 while the average number of output bits generated in a conventional synchronous ADC with an 8-bit quantiser is 8. Thus the average number of output bits is reduced by 6.04 times in the synchronous LC ADC as compared to a conventional SAR ADC for an AEG signal. Even at higher resolutions the total number of output bits generated does not increase substantially. Even for 12-bit resolution the number of output bits is reduced by about 4-times.

This is highly beneficial as then the total amount of data to be transmitted or stored is reduced substantially. Moreover, if the data is required to be transmitted wirelessly, this would also reduce the pressure on the transmission circuit and save power as well.

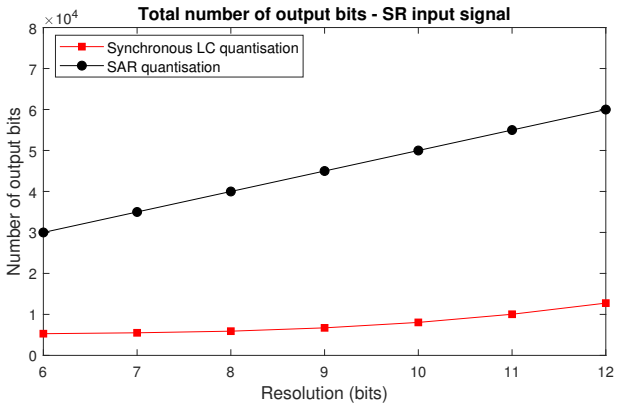


Figure 4.9: Total number of output bits generated for quantisation of trial electrogram in sinus rhythm, determined in MATLAB model simulations. The AEG signal with SR has 4999 samples.

4.5. Modelling Split Resolution technique in synchronous LC ADC

The number of quantisation steps required in the proposed synchronous LC ADC was determined in the MATLAB model. The number of quantisation steps increases for samples which have significant difference in amplitude from the previous sample. The number of quantisation steps required can be reduced by introducing the split resolution technique. A MATLAB model of the synchronous LC ADC (Appendix-A.4) was used to implement the split resolution technique. The threshold for the split resolution was set at +/- 15% of the mean level of the amplitude range. The operation of the split resolution technique in the synchronous LC ADC is shown in Fig. 4.10. The number of quantisation steps for each sample of the AEG signals were determined and plotted in Fig. 4.11. The number of quantisation steps required when the split resolution techique was not implemented is also plotted to show the difference.

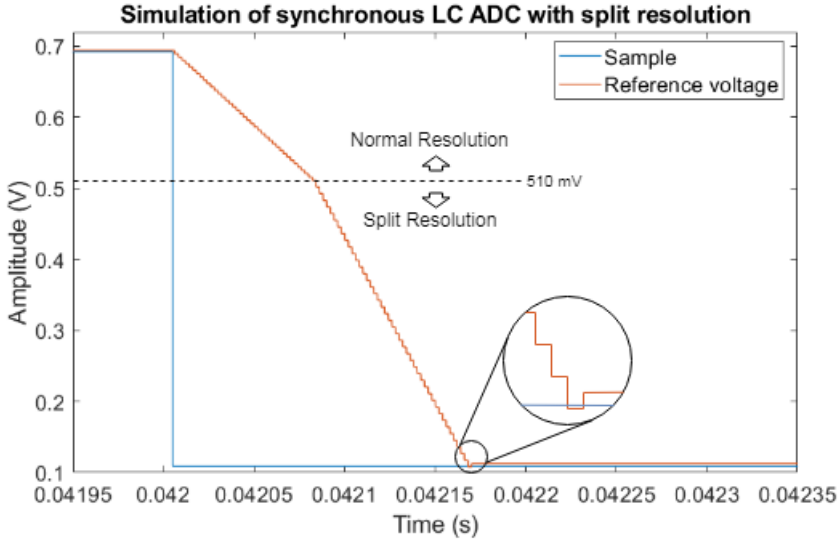


Figure 4.10: Simulated waveform of synchronous LC ADC with split resolution technique.

As can be observed in Fig. 4.11, the magnitude of decrease of the number of quantisation steps required due to implementation of the split resolution technique increases for higher resolutions. However, the reduction in number of steps required is about 34% on average for all quantiser resolutions. This ratio can be varied by changing the threshold used in the split resolution method or by introducing more thresholds to further reduce the quantiser resolution in steps.

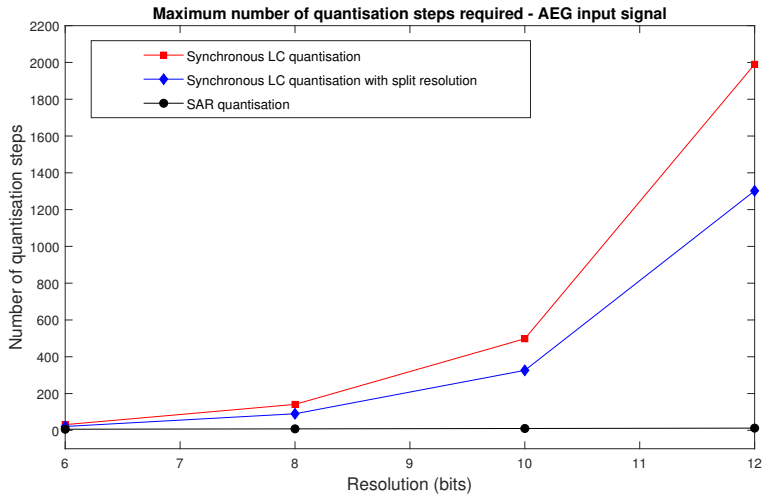


Figure 4.11: Maximum number of quantisation steps required for quantisation of AEG signals in the proposed synchronous LC ADC without split resolution and with split resolution. The estimation was performed in the VerilogA model with AEG input signal. The threshold for the split resolution was set at +/- 15% from the mean level of the signal.

4.6. MODELLING OF MLC ADC

Similarly, a multichannel model is designed to convert input signals from 4 channels. A multiplexer (MUX) is used to select one of the channel wires depending on the output of the counter. The clock signals driving the MUX and the SH block are synchronised such that a sample is taken every time a different channel is selected by the MUX. After a sample is taken the reference window from the last sample of that corresponding channel is loaded and then quantisation is performed. The operation of the VerilogA model (Appendix-A.2)of the multichannel synchronous LC ADC is shown in Fig. 4.13.

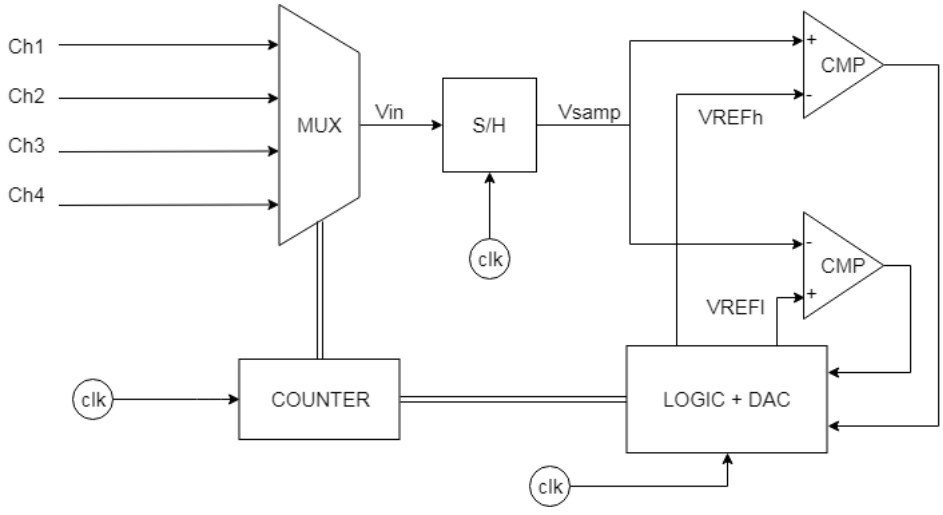


Figure 4.12: Model for multichannel synchronous LC ADC.

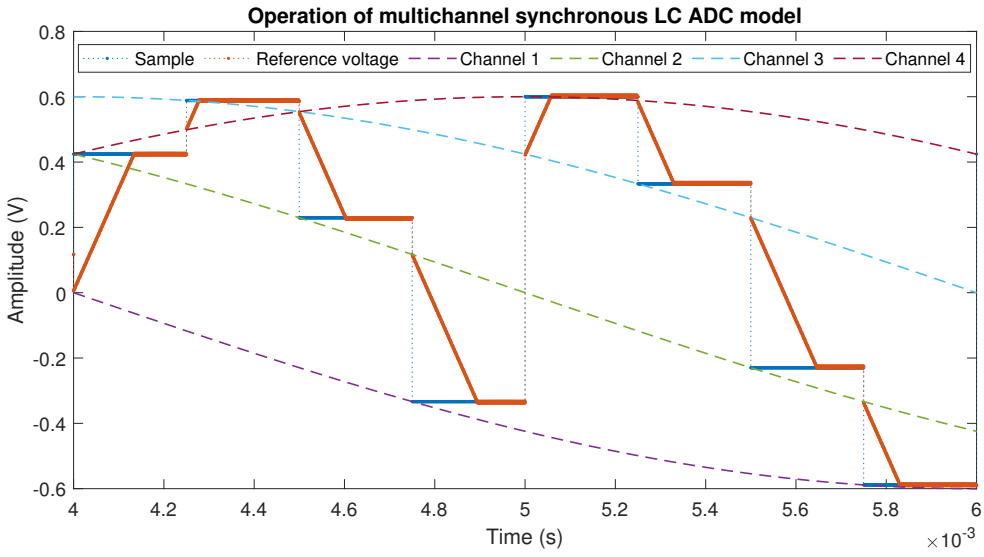


Figure 4.13: Simulated waveform of VerilogA model of multichannel synchronous LC ADC.

4.7. SHARED REFERENCE MEMORY FOR MULTICHANNEL SYNCHRONOUS LC ADC

The multichannel configuration of the synchronous LC ADC offers two possible methods of design. After quantisation of each sample the final value of the reference voltage level is stored in the memory of the control logic block. A separate memory could be created for each channel so that the memory for that corresponding channel is used as and when a sample from that specific channel is being quantised next. Another method is to use the same memory for all channels such that the stored reference voltage level for one channel will be used as the reference for the next channel and so on.

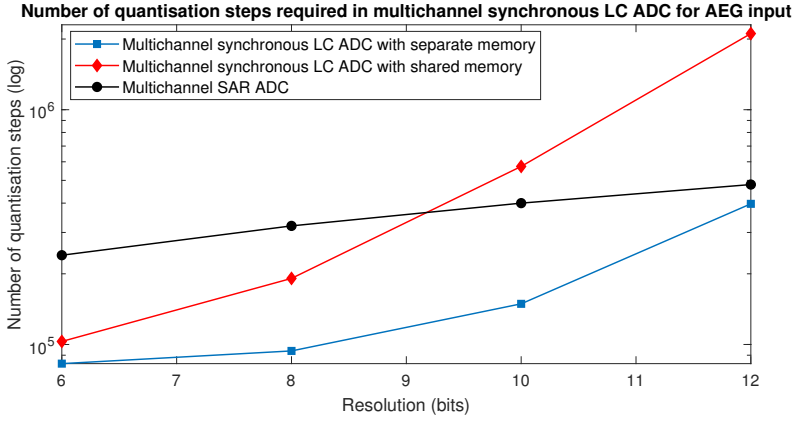


Figure 4.14: Estimation of number of quantisation steps required for quantisation of 39996 samples of the AEG signal in multichannel synchronous LC ADC while using the same memory across all channels and while using separate memory for each channel.

The advantage of using the same memory across all channels is that it reduces the additional circuitry required to store, switch between memory locations and fetch for every channel. It makes the design of the logic block less complex. Moreover, since the signals coming through several channels are roughly in the same voltage range, this should also reduce the number of quantisation steps required by the quantiser

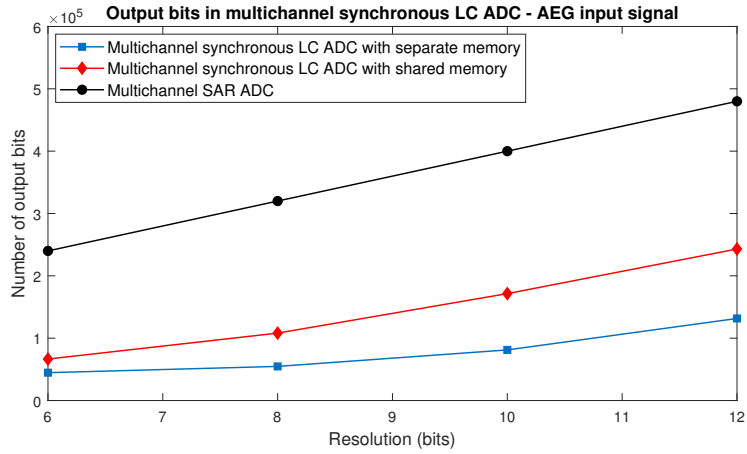


Figure 4.15: Estimation of number of output bits generated for quantisation of 39996 samples of the AEG signal in multichannel synchronous LC ADC while using the same memory across all channels and while using separate memory for each channel.

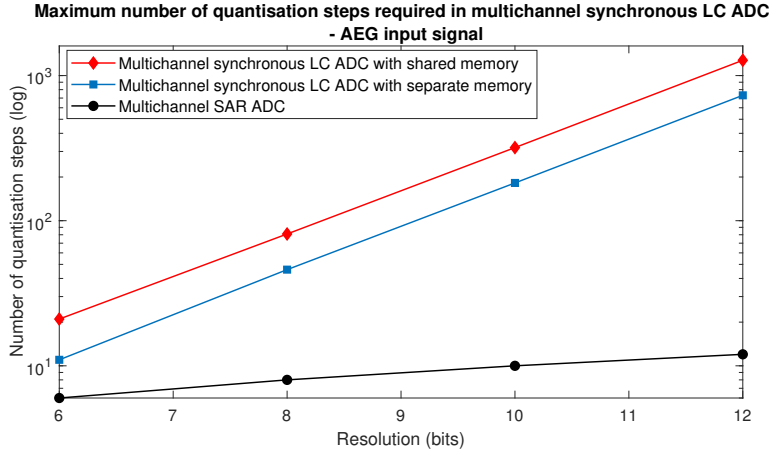


Figure 4.16: Estimation of maximum number of quantisation steps required for quantisation of 39996 samples of the AEG signal in multichannel synchronous LC ADC while using the same memory across all channels and while using separate memory for each channel.

to quantise of samples across several channels.
This assumption was tested in the MATLAB model and compared with the model with separate memory for each channel. The results are shown in Fig. 4.12. As can be observed in the results, the assumption that the number of quantisation steps would be reduced while using the same memory across several channels is true. However, it can also be observed that using separate memory for each channel results in even greater reduction in quantisation steps required. Moreover, in the case of the shared memory configuration more output bits were generated (Fig. 4.15) and the maximum number of quantisation steps required was also higher (Fig. 4.16). Hence, depending on the requirements of the application either of the two methods can be implemented to design the multichannel synchronous LC ADC.

4.8. Conclusions

The operation of the proposed synchronous LC ADC was verified through VerilogA and MATLAB models. The advantages of the proposed ADC over conventional synchronous ADCs was quantified as well. The overhead in the synchronous LC ADC is that it requires a faster quantisation clock signal as compared to conventional SAR ADCs. At the same time, the proposed ADC can achieve a reduction of 3.3-times in the average number of quantisation steps required per sample. It can also achieve a 6-times reduction in the average number of output bits generated per sample. This estimation was performed with AEG input sig-

nals and it was observed that the sparsity of the AEG signal (in both AF and SR conditions) was exploited successfully by the synchronous LC ADC.

The multichannel configuration of the synchronous LC ADC was also tested through VerilogA and MATLAB models. It was observed that while using a shared memory across all channels to store the quantised reference level reduces the design complexity, it increases the average number of quantisation steps required per sample and the average number of output bits generated per sample. However, its performance in these metrics is still better than a conventional SAR ADC. Hence, depending on the requirements, this approach can be considered as well.

5

CIRCUIT IMPLEMENTATION OF MULTICHANNEL SYNCHRONOUS LC ADC

The synchronous multichannel LC ADC (MLC ADC) introduced in the previous chapter is designed to be implemented on the TSMC 180 nm CMOS process. The block level diagram of the MLC ADC is shown in Fig. 5.1. The various circuit blocks in the proposed MLC ADC and their design considerations are discussed in subsequent sections.

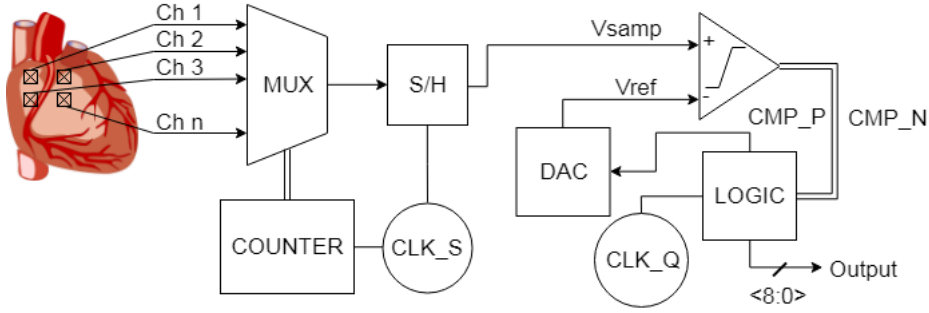


Figure 5.1: Block level diagram of proposed MLC ADC.

5.1. MUTIPLEXING BLOCK

The input channels are multiplexed to the Sample Hold (SH) block through a multiplexing block. The multiplexing block consists of a 2-bit ring counter and an analog multiplexer. The counter is defined in Verilog and synthesized in Synopsys Design Compiler. The counter is driven by a clock signal with a frequency of 4*Fs, where Fs is the sampling frequency. The counter has 4 output signals which are used to select one of the input channels. The counter is configured such that when a particular number is reached, the corresponding output signal is pulled high while the other signals are pulled low. These signals are used to drive the NMOS switches in the analog multiplexing block. When one of the signals is pulled high the corresponding NMOS switch is closed and the input signal is passed to the Sample and Hold block. The count is incremented at the next falling edge of the clock signal such that the SH block can enter the track phase for that corresponding channel at the same time. The analog multiplexing block is shown in Fig. 5.2.

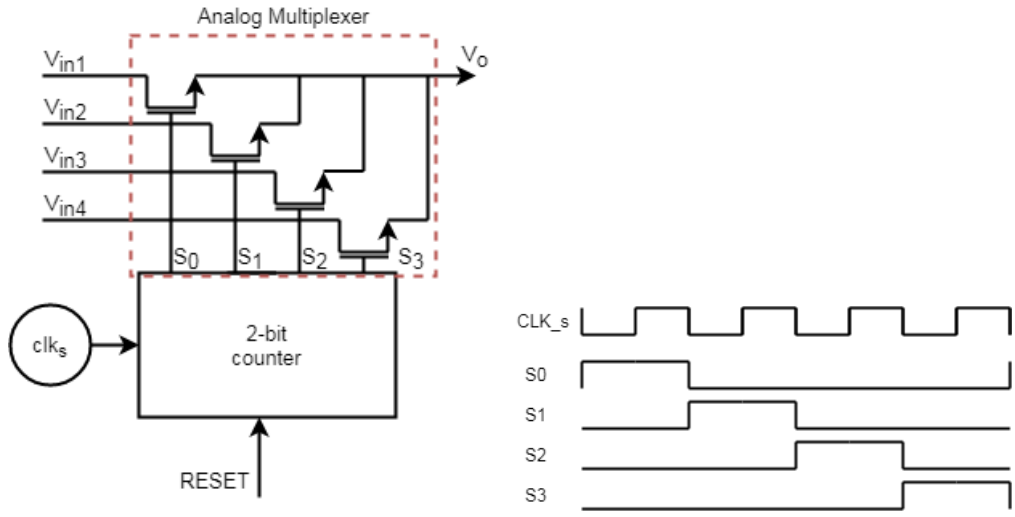


Figure 5.2: Circuit schematic of analog multiplexer block and timing diagram of operation of the multiplexer block.

5.2. Sample & Hold

The Sample & Hold block tracks the input signal when the sampling clock signal is low and holds it at the positive edge of the sampling clock signal. Two transmission gates are used to transfer the input signal to the hold capacitor and to the comparator, respectively. The circuit of the Sample and Hold block is shown in Fig. 5.3. A 1 pF hold capacitor is used to store the sample. A large capcitor is used to minimise the noise contribution from the capacitance. A delay is introduced to the sampling clock signal to generate the controlling signal for the second T gate in the SH block. This is done to ensure that the input signal does not affect the sample when the switches are opened.

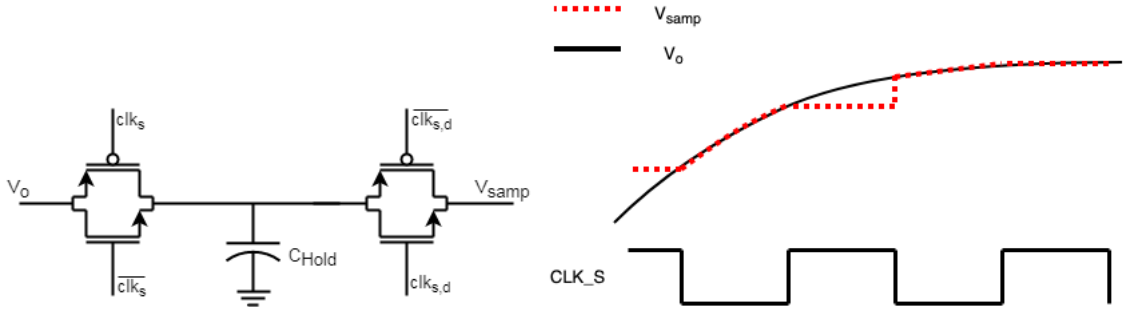


Figure 5.3: Circuit schematic of sample and hold block and timing diagram of operation of sample and hold block.

5.3. Comparator and Preamplifier

The proposed ADC follows the synchronous operation. Thus a dynamic latch comparator circuit was implemented, based on the Strong-Arm latch architecture [31]. The dynamic latch suffers from considerable kick-back noise which affects the input signals during comparison. Hence, a preamplifier was implemented to reduce the effect of the kickback noise from the comparator. The circuit of the comparator and the preamplifier are shown in Fig. 5.4.

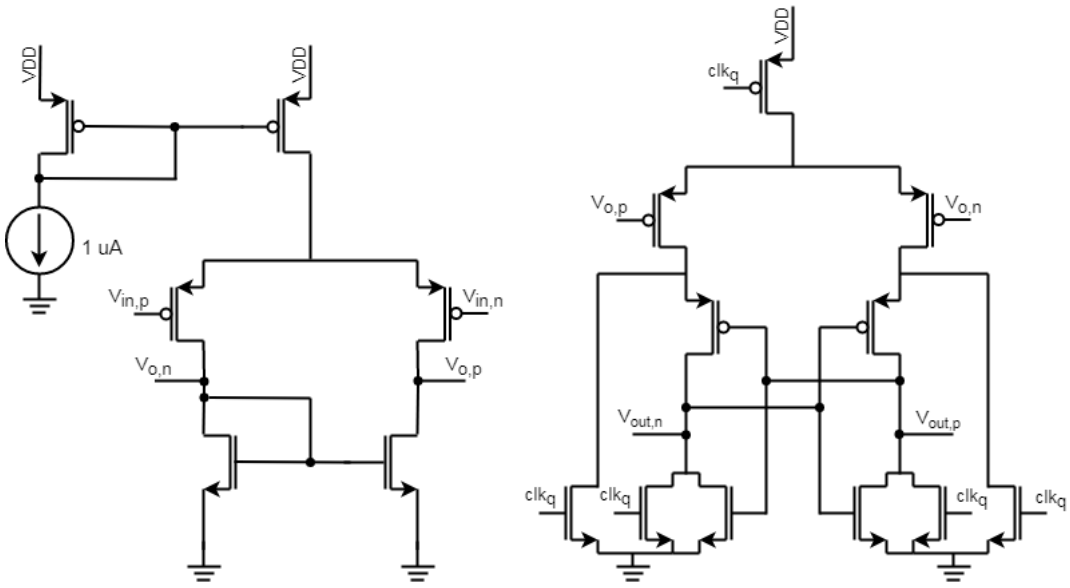


Figure 5.4: Circuit schematic of preamplifier (left) and dynamic comparator (right).

5.4. DAC

A multi-bit binary-weighted capacitive DAC was implemented to generate the reference levels for comparison with the sample. The unit capacitor of the DAC was designed to be 35.6 fF. A dummy capacitor with the same dimensions as that of the unit capacitor was added to the DAC to create ratios in the powers of 2. NMOS and PMOS switches are used to charge and discharge the capacitors to Vref and VSS, respectively. As the common mode input range of the comparator is 0-1.2 V, the Vref was taken to be 1.2V. A simplified schematic of the ADC is shown in Fig. 5.5.

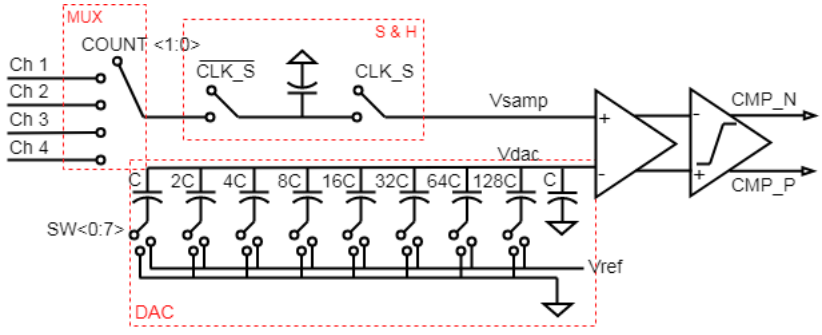


Figure 5.5: Simplified circuit schematic of proposed MLC ADC.

5.5. Control Logic Block

The operation of the ADC is controlled by the Control Logic Block. The operation of the Control Logic Block as described in the previous chapter is implemented in Verilog and synthesized using the TSMC .18 um CMOS library in Synopsys Design Compiler.

The Control Logic Block consists of memory for each channel in which the quantised signal (reference level) is stored. This memory is accessed when the respective channel is selected for conversion in the counter described previously. This functionality could also be modified to use a single memory for each channel.

There are two methods for implementing the control logic which was also described in the previous chapter:

1. A separate memory for each channel such that the quantised reference voltage level of the previous sample of each channel is used as the starting point for the quantisation of the next sample of the corresponding channel.

2. A single memory to save the quantised reference voltage level of the previous sample which is used as the starting point for each new sample, irrespective of the channel from which the sample is captured.

Using the same memory across all channels reduces the complexity in designing the control logic block and also reduces the amount of space required for the synthesized CMOS circuit. However, there is a tradeoff in the number of quantisation steps required for the quantisation of samples, which is higher in the case of the single shared memory option.
As the design of the control logic block is done in Verilog, the overhead in creating a separate memory for each channel is minimal. Moreover, the difference in the area of the synthesized circuit is small as compared to the area required for the whole ADC. Hence, it was decided to opt for the creation of a separate memory for each channel.

The Split Resolution technique is also implemented to reduce the number of quantisation steps required for samples with large differences from previous samples. The threshold level T_{split} is pre-programmed at +/- 15% from the mean reference level (mid-swing) in the Control Logic Block. The threshold level can be decided based on signal activity such that only the spikes in signal are above the threshold level T_{split}. The Verilog code for the Control Logic Block can be found in Appendix-B.1.

The timing diagram of the operation of the Control Logic Block is shown in Fig. 5.6. When the sampling clock signal (CLK_s goes high the SH block samples the input signal and the quantisation process is started. The final reference level from the previous quantisation of that particular channel is loaded in the DAC with the DAC_SET signal. Then the CMP_EN is then pulled high to start the comparison. This process is repeated until the output of the comparator is inverted, as described in the flowchart in Fig. 3.21. When the quantisation is completed the DONE signal is pulled high. The ADC needs to satisfy the require-

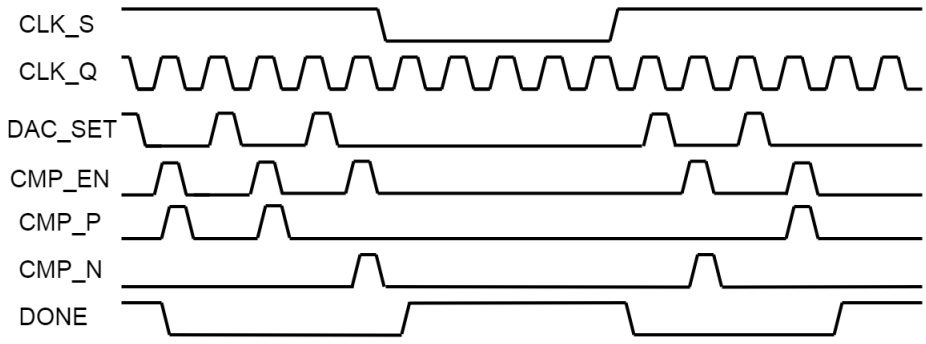


Figure 5.6: Timing diagram of operation of the control logic circuit.

ment of the dynamic range which is 58.06 dB. Hence, in order to satisfy the requirement the ADC needs to have SNDR performance better than the dynamic range requirement. Thus, a 10-bit ADC is required as its SNDR (theoretical) performance would be 61.98 dB. The architecture of the proposed synchronous LC ADC can be scaled for higher resolutions, higher sampling frequencies and higher number of channels. Hence, another version of the ADC was also designed with 10-bit quantiser resolution. The DAC and the control logic blocks were redesigned accordingly (Appendix-B.2).

5.6. Reference Multichannel SAR ADC Design for comparison of performance

The architecture of the proposed synchronous LC ADC is similar to that of a conventional SAR ADC as it also uses a comparator, preamplifier, multi-bit DAC and S&H block. The primary difference between the two ADCs is in the method of quantisation. Thus, the control logic is defined differently in the SAR ADC and in the synchronous LC ADC. Moreover, while the SAR ADC always uses a quantisation clock signal with a certain frequency, the frequency of the quantisation clock signal required by the synchronous LC ADC is set according to the activity of the input signal.

SAR ADCs can operate with ultra low power consumption and thus are able to achieve the best FoM. However, the design of all the circuit blocks in such SAR ADCs are optimised to operate with ultra low power consumption. Moreover, the SAR ADCs are designed at advanced process nodes which offer other advantages as well. Since the synchronous LC ADC is being designed in TSMC's 180 nm CMOS process and it is designed only as a proof of concept, the comparison between the state of the art SAR ADC and the proposed synchronous LC ADC would not be possible. Hence, a SAR ADC is also designed in a similar manner to compare its performance with that of the synchronous LC ADC (Appendix-B.3). Except for the control logic block used in the synchronous LC ADC, all other circuit blocks are reused in the SAR ADC.

The control logic for the SAR quantisation method is developed in Verilog and synthesized in Synopsys Design Compiler using the TSMC .18 um library in the same way as was done for the control logic block of the synchronous LC ADC.

5.7. Conclusions

The circuit blocks of the proposed synchronous LC ADC are designed at the circuit schematic level using the TSMC .18 um process library. The design considerations and specifications of the circuit blocks are discussed. Four ADCs are designed in a similar process:

1. Multichannel synchronous LC ADC with 8-bit quantiser resolution
2. Multichannel synchronous LC ADC with 8-bit quantiser resolution and Split Resolution technique
3. Multichannel synchronous LC ADC with 10-bit quantiser resolution
4. Multichannel SAR ADC with 8-bit quantiser resolution

Each ADC is designed to convert input signals from 4 channels simultaneously. The designed ADC circuits are simulated to check and compare their performance. The simulation results are discussed in the following chapters.

6

RESULTS AND DISCUSSION

The circuit schematic of the ADCs were simulated with different input signals to determine the performance of the ADCs. The simulation setup is described in Appendix-C.

6.1. RESULTS FOR SINUSOIDAL INPUT

The sinusoidal input signal in each channel was sampled at 1 kS/s and quantised using the level crossing quantisation method which was introduced in the previous chapter. The frequency of the quantisation clock signal was determined to be 500 kHz per channel by using the MATLAB model. Hence, the quantisation clock signal with frequency of 2 MHz was used for the quantisation. A sampling clock signal with frequency of 4 kHz was used for the sampling.

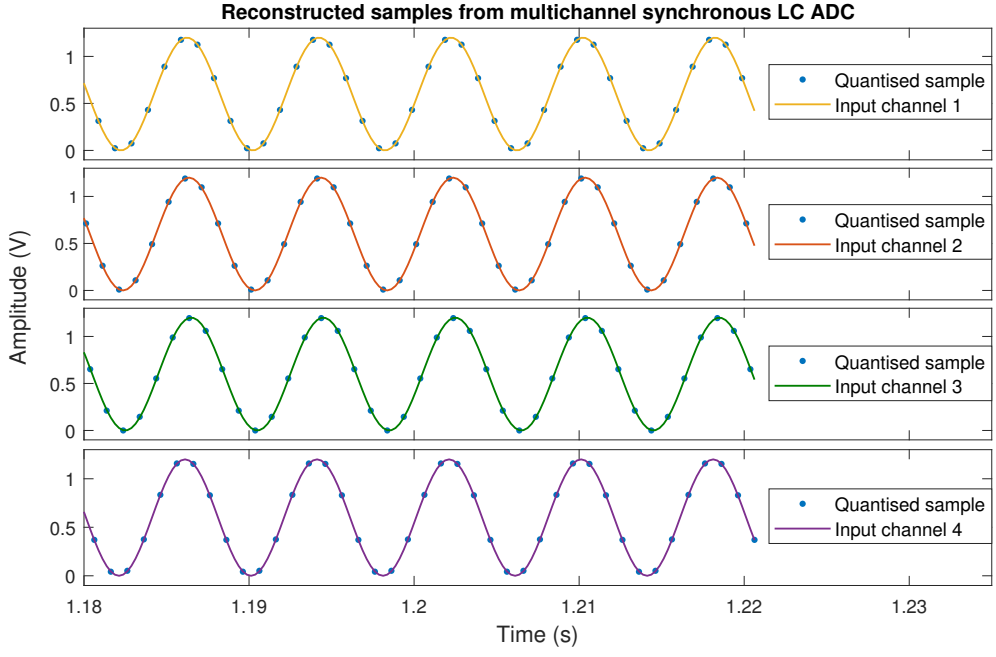


Figure 6.1: Input sinusoidal signal and quantised samples obtained by reconstruction of output derived from the circuit level implementation of the MLC ADC. The input signal is a sinusoidal signal at 125 Hz and 1.2 V_{p-p} input swing and is sampled in each channel at 1 kS/s.

After collection of the samples the original input signals were reconstructed in MATLAB. The output of the synchronous LC ADC consists of the difference between the quantised values of the current sample and the previous sample in terms of LSB steps. This data is used to reconstruct the quantised samples. The reconstructed samples are shown in Fig. 6.1. Welch's power spectrum estimate function is used to plot a 1024-point FFT of the reconstructed samples. The power spectrum of the reconstructed samples from channel 1 and channel 2 are shown in Fig. 6.2 and Fig. 6.3, respectively.

The power spectrum of each channel shows the main tone of the input signal at 125 Hz and its harmonic distortion components at 250 Hz, 375 Hz and 500 Hz as well. It also shows the DC component (caused by the offset of 600 mV) in the signal as well as the noise floor. Several performance metrics of the multichannel synchronous LC ADC are determined from this plot. The performance metrics are tabulated for channel 1 and channel 2 separately in Table. 6.1. The power spectrum and performance metrics of channel 3 and channel 4 are not shown as they were found to be erroneous. The power spectrum plot of both channels showed a much lower noise floor than expected. In the power spectrum plot of channel 3 the noise floor was observed to be flat while in that of channel 4 the power spectrum was observed to be at -350 dBW. Hence, the performance metrics derived from these plots were not considered.

The power spectrum plot of channel 1 and channel 2 show intermodulation artifacts around the main tone and harmonic distortion components. Upon investigation it was found that these artifacts matched with the sampling frequency. The intermodulation was caused by the clock feedthrough in the switches in S&H block. Transmission gates were used in the S&H block to prevent clock feedthrough but the sampled signal was still affected. The on-resistances of both the PMOS and NMOS switches should be matched in the transmission gates used in the S&H blocks which was not done in this case. Unfortunately, due to the long simulation times the performance could not be measured after rectification of the problem.

The SNDR performance of the ADC was found to be 50.42 dB from the reconstructed samples in channel 1. This is higher than the expected value as the theoretical maximum quantisation noise performance for an 8-bit ADC is estimated to be:

$$\begin{aligned} SQNR &= 6.022 * N + 1.76 \\ &= 6.02 * 8 + 1.76 \\ &= 49.93\, dB \end{aligned} \tag{6.1}$$

This results in a an effective number of bits (ENOB) of 8.08 bits which is higher than theoretically possible. It is suspected that the presence of the intermodulation components affect the accurate determination of the signal to noise and distortion (SNDR/SINAD) performance of the ADC. During calculation of SNDR, the first 5 harmonic distortion components in the power spectrum are considered. However, in the power spectrum being considered only 3 distortion components are observed. Although the higher distortion components are folded back into the existing frequency range i.e. 0 - $f_s/2$, the accuracy in calculation is reduced. In the power spectrum of output from channel 2 the SNDR is found to be 48.96 dB which results in an ENOB of 7.83 bits. While this result is more realistic, the effect of intermodulation effects is observed in the power spectrum in this case as well. Hence, more measurements are required to measure the performance on the ADC more accurately.

The multichannel synchronous ADC consumes 9.32 μW from the 1.8 V supply while converting from 4 input channels simultaneously. Therefore, the power consumption per channel is 2.33 μW. As discussed in the previous chapter, the power consumption in the synchronous LC ADC is dependent on the activity of the input signal. Therefore, the power consumption in the synchronous ADC would be the highest for a sinusoidal input signal. In comparison, the reference multichannel SAR ADC designed using the same components consumes 4.33 μW while sampling the sinusoidal input signal at 1 kS/s. The power consumption in the SAR ADC is the same irrespective of the input signal. Furthermore, each channel of the synchronous LC ADC produces 7674 output bits for recording 1024 samples of the sinusoidal input signal whereas the SAR ADC produces 8192 output bits for the same. The for each sample the synchronous LC ADC produced about 7.5 bits whereas the SAR ADC produced 8 bits. This metric is activity-dependent in the synchronous LC ADC whereas it is constant for the SAR ADC. The Figure of Merit (FoM) [32] is used

to compare the performance of ADCs as it shows the power consumption per conversion step in the ADC. Generally, in synchronous ADCs the number of quantisation steps is constant and thus the calculation of the FoM is straight-forward. However, the synchronous LC ADC uses different number of quantisation steps for quantisation of each sample. Hence, calculation of FoM for a synchronous LC ADC requires extensive modelling.

Similarly the multichannel synchronous LC ADC with 10-bit quantiser resolution was simulated with a sinusoidal input signal. The ADC consumed 36 μW on average, which is 4 times higher than the amount of power consumed in the synchronous LC ADC with 8-bit quantiser resolution. This is because of the 4 times higher quantiser clock frequency used in the ADC. The simulation results discussed until now were obtained with a supply voltage of 1.8 V. By reducing the power supply voltage the power consumption should be reduced as well. Hence, the synchronous LC ADC with 8-bit quantiser resolution was simulated with a supply voltage of 1 V and with sinusoidal input signals. The reference voltage used in the DAC was reduced to 500 mV as the common-mode input range of the comparator also reduced with the supply voltage. At the reduced supply voltage the power consumption in the ADC was reduced to 3.2 μW (800 nW per channel). The performance metrics could not be measured. However, the quantised samples were reconstructed and the input signal could be successfully reconstructed. The input signal and reconstructed samples of channel 1 are shown in Fig. 6.4.

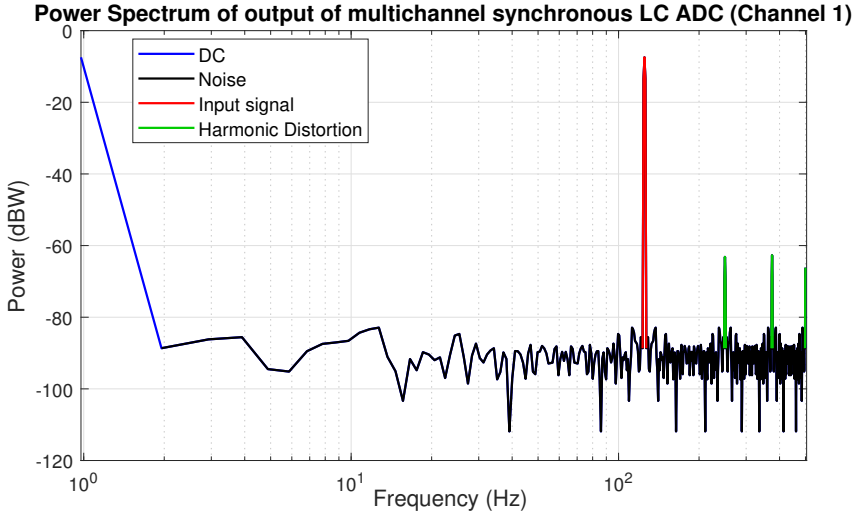


Figure 6.2: Power spectrum of output obtained from the circuit schematic level implementation of the MLC ADC (Channel 1). The power spectrum is plotted with 1024 points.

Table 6.1: performance metrics of multichannel synchronous LC ADC

Parameter	***Channel 1***	***Channel 2***
V_{supply} (V)	1.8	1.8
Sampling rate (kS/s)	1	1
SNDR (dB)	50.42	48.96
SNR (dB)	56.81	54.67
THD (dB)	-51.56	-50.36
ENOB (bits)	8.08	7.84
SFDR (dBc)	54.97	53.33
Power/channel (uW)	2.33	2.33

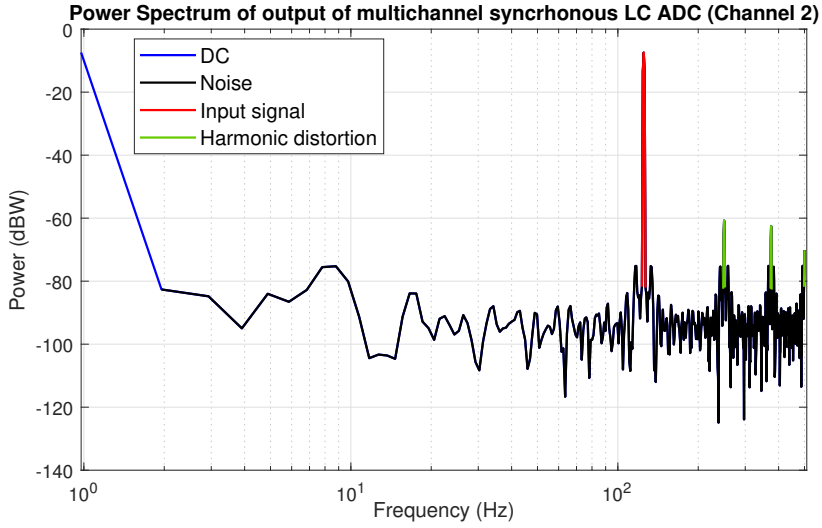


Figure 6.3: Power spectrum of output obtained from the circuit schematic level implementation of the MLC ADC (Channel 2). The power spectrum is plotted with 1024 points.

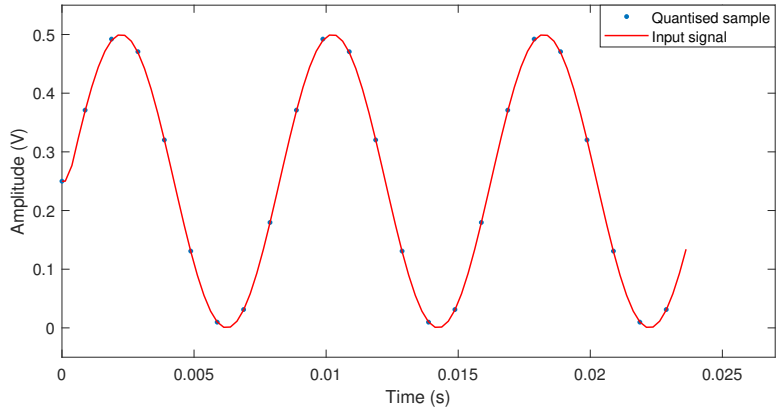


Figure 6.4: Reconstructed samples overlaid over input signal of synchronous LC ADC with 1 V supply.

6.2. Results for AEG input

AEG signals were also sampled and quantised in the synchronous LC ADC using the simulation setup described in Appendix-C. The reconstructed samples are overlaid over the original input signals in Fig. 6.5. As can be observed in the plot, the original AEG input signal could be reconstructed accurately from the quantised output of the synchronous LC ADC. The maximum number of quantisation steps required for the AEG signals was estimated in the MATLAB model to be 150. Hence, a quantisation clock signal with a frequency of 2.4 MHz (600 kHz per channel) was supplied to the synchronous LC ADC. The multichannel synchronous LC ADC consumed 7.58 μW (1.9 μW per channel) of power while converting the AEG input signals. Thus, due to reduced activity of the AEG input signals, the power consumption was reduced as well. Power consumption in the SAR ADC was unchanged at 4.33 μW. During conversion of 3920 samples (980 samples in each channel) of the AEG input signal, a total of 4728 bits of output were generated in the multichannel synchronous LC ADC whereas the multichannel SAR ADC produced 31360 output bits. This implies that for each sample of the AEG input signal the synchronous LC ADC produced about 1.2 bits of data while the SAR ADC produced 8 bits. Thus, the number of output bits generated is reduced by over 85% in the synchronous LC ADC.

The split resolution technique was introduced in the proposed synchronous LC ADC architecture to re-

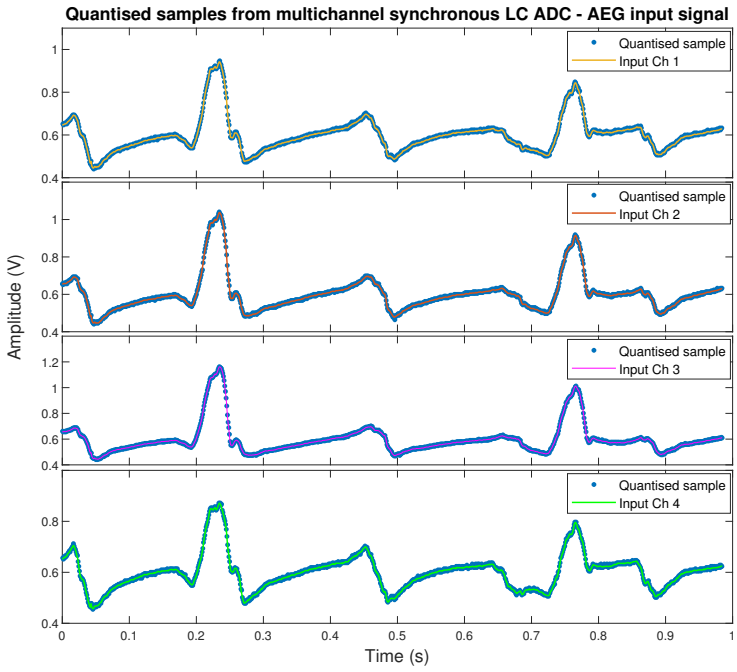


Figure 6.5: Reconstructed samples of AEG input signal recorded in the multichannel synchronous LC ADC. The input signal is scaled according to the common mode input range of the ADC.

duce the maximum number of quantisation steps required per sample. The schematic level implementation of the multichannel synchronous LC ADC with split resolution technique was simulated to convert AEG input signals. AEG input signals with sharp drops were converted in this ADC. While the synchronous LC ADC requires a maximum of 150 quantisation steps per sample, with the introduction of the split resolution technique the number could be reduced to 100. Thus, the quantisation clock frequency was reduced to 1.6 MHz (from 2.4 MHz). The reconstructed samples overlaid over the original input signals are shown in Fig. 6.6. With the reduction in the quantisation clock frequency, the power consumption was reduced as well to 7.4 μW (1.85 μW per channel). This simulation was however performed for a small segment of AEG input signal (100 ms) with the maximum activity. Thus, with further simulations the average measured power consumption is expected to be lower.

6.3. Discussion

The circuit simulation results prove that the synchronous LC ADC can be used to record input signals from multiple input channels simultaneously while exploiting the sparsity of the input signal in time domain. Moreover, the synchronous LC ADC reduces the amount of data generated considerably.

The overhead in the synchronous LC ADC is that it requires a high-frequency quantisation clock signal which also causes the power consumption in the ADC to increase. However, at the same time the reduced data output is advantageous as it will reduce the power consumption in the data transmission block. Moreover, introduction of split resolution technique helps in reduction of the required quantisation clock frequency. Thus, the tradeoffs can be weighed to design the synchronous LC ADC.

The target application for this ADC is recording of atrial electrograms from 192 channels simultaneously. While the 10-bit quantiser resolution satisfies the dynamic range requirement of the input signal, all 192 channels can not be recorded by using a single synchronous LC ADC as then the resulting sampling and quantisation clock frequencies would be much higher than the range considered in this work. Moreover, routing 192 channels to the ADC from the bond pads would prove to be difficult and introduce delay in between the signals. Hence, multiple synchronous LC ADCs can be used to sample from several channels in each ADC and thus record from all input channels.

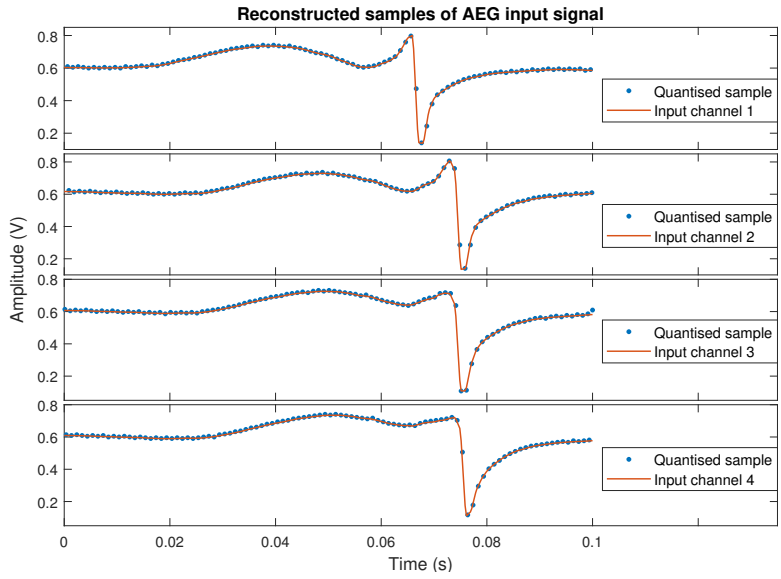


Figure 6.6: Reconstructed samples of AEG input signal recorded in the multichannel synchronous LC ADC. The input signal is scaled according to the common mode input range of the ADC.

As each synchronous LC ADC requires a multi-bit capacitive DAC, considerable amount of area would be required to implement several ADCs together on a single chip. However, creating bond pads for 192 input channels would require a large area as well and each input channel would need analog circuit blocks such as amplifiers, filters etc. Hence, the total area occupied by the ADCs would be much lower in comparison. An alternative approach could be to design several chips which can record from a certain number of channels.

6.4. Conclusions

The simulation results prove that the synchronous LC ADC is capable of recording from several input channels simultaneously while exploiting the sparsity property of the input signal and reducing the amount of output data generated as well. The synchronous LC ADC can be scaled to higher quantisation resolutions and can also be operated at a lower voltage as well. The split resolution technique can improve the performance of the synchronous LC ADC even further by reducing the required quantisation clock frequency and thus the power consumption.

7

Conclusions and Future Propositions

The conclusions from the previous chapters are summarised in this chapter and the research contributions are discussed. Some recommendations for future work are also discussed.

7.1. Conclusions

The main purpose of this work was to seek the design of an LC ADC which can convert input signals from several input channels simultaneously while ensuring that the operation of the ADC is activity-dependent. The operation of conventional LC ADCs were investigated in **Chapter 2** and problems in existing designs reported in literature were determined, which are -

1. Large amount of data generated in LC ADCs at high quantiser resolutions.
2. Extra data required to encode the change in quantiser resolution in adaptive resolution technique.
3. Absence of multichannel LC ADC architectures to record from several input channels simultaneously.

Several architectures were considered in **Chapter 3** to implement a multichannel LC ADC and a system-level qualitative analysis was conducted to determine the best option for the same. It was found that the level-crossing sampling approach is inherently asynchronous and hence it is not compatible with the synchronous property of the time-multiplexed approach used in conventional multichannel ADC. This observation was used to explore alternative architectures which could solve this problem. The synchronous LC ADC architecture was selected based on the system-level analysis as it could incorporate both a synchronous sampling method with a level-crossing based quantisation method. The split resolution technique was developed as an alternative to the adaptive resolution technique and the various methods of implementation of the the technique were discussed as well. Through modelling simulations it was found that the split resolution technique reduces the amount of data generated in LC ADCs and does not suffer from the data overhead observed in the adaptive resolution technique.

Models were developed and presented in **Chapter 4** which verified the operation of the synchronous LC ADC. The activity-dependence of the operation of the synchronous LC ADC was proved through models and so was the activity-dependence of output generation. It was observed that the number of quantisation steps could be reduced by 3.37 times and the number of output bits could be reduced by 6 times by using the synchronous LC ADC architecture while converting AEG input signals. The analysis was performed for a range of quantisation resolutions (6 - 10 bits). Models for multichannel configuration of the synchronous LC ADC were also presented and possible approaches for the design of the memory were also discussed along with modelling simulations.

The CMOS circuit implementation of the synchronous LC ADC was presented in **Chapter 5**. Design configurations with split resolution and 10-bit quantiser resolution were also presented. A reference SAR ADC implementation for comparison of performance was also presented.

The simulation results of the schematic level implementation of the ADCs were presented in **Chapter 6**. The simulation results proved the capability of the synchronous LC ADC to convert the input signal from multiple input channels simultaneously and the activity-dependence of the operation was observed in simulation and power measurements. While the synchronous LC ADC used a faster quantisation clock frequency as compared to the SAR ADC (2 MHz compared to 128 kHz), the power consumption was not scaled in the same way as compared to the SAR ADC (9.32 μW compared to 4.33 μW). Thus, the activity-dependence of synchronous LC ADC helped in achieving lower power consumption, although it was still higher than that in the SAR ADC. The split resolution technique can improve the performance of the synchronous LC ADC and this assumption was proved through simulations. Some system level design aspects were discussed in which the approaches to implement the synchronous LC ADC for 192 channels (for AEG recording) were considered.

The research questions formulated in **Chapter 1** can thus now be answered. The research questions and their answers are-

1. **Can the level-crossing sampling approach be implemented in a multichannel configuration?**

   Yes. The level-crossing sampling approach is incompatible with a time-multiplexed approach for implementing a multichannel LC ADC. Hence, a different approach can be considered in which the sampling is performed synchronously while the quantisation is performed according to principle of level-crossing quantisation.

2. **Can prior knowledge of the input signal (in this case, AEG) be helpful in designing a more efficient ADC?**

   Yes. The AEG signal is sparse in the time domain. Thus, the quantisation method can be optimised by selecting the maximum quantisation clock frequency required for recording of the AEG signals in the synchronous LC ADC. Moreover, the split resolution technique can be used in which a threshold is used to define the resolution of the quantiser based on the input signal level. Thus, for peaks in the input signal a lower quantiser resolution can be used and for smaller signals a higher quantiser resolution can be used.

7.2. SUMMARY OF CONTRIBUTIONS

This work resulted in a novel ADC architecture which samples synchronously while being activity-dependent during quantisation of the sample. Thus, a multichannel LC ADC could be designed which was not previously available in literature. The synchronous LC ADC could also achieve activity-dependent data generation which has not yet been reported in synchronous ADCs. The synchronous LC ADC is a better design for an LC ADC as it uses a single dynamic comparator compared to two continuous-time comparators used in conventional LC ADCs.

The adaptive resolution technique presented in several sources suffers from a data overhead which was not considered earlier. This work found the problem and presented an alternative solution, namely the split resolution technique which does not suffer from the same problem. The methods for implementation of the split resolution technique in a conventional LC ADC were also developed and presented.

Several new architectures were developed in this work which can be considered for design of a multichannel LC ADC. This analysis was not found previously in literature.

7.3. Recommendations for Future Work

1. Controlling the quantisation process with a quantisation clock signal puts a limit on the maximum bandwidth of the input signal and the maximum sampling rate. This limitation can be removed by making the quantisation method in the synchronous LC ADC fully asynchronous [33]. The quantisation process thus becomes self-timed and is completed much faster. Moreover, the requirement for estimating the quantisation clock frequency is no longer required.

2. The bandwidth of the input signal of the synchronous LC ADC is in the low frequency range as this ADC is targeted towards acquisition of biosignals. Flicker noise affects the conversion of the input signal in the low frequency range. This problem was not considered in this work and methods such as chopping or auto-zeroing could be considered in future to solve this problem.

3. The circuit level implementation in this work was aimed towards proving the system-level design and hence is not fully optimised. The circuit-level implementation can be optimised further to improve the performance metrics of the ADC. The synchronous LC ADC could be designed for lower supply voltages as then it would be able to achieve lower power consumption, as was also proved in the circuit simulations.

4. The split resolution technique used in the synchronous LC ADC was tested with a single threshold. More number of thresholds could be considered in future to reduce the maximum number of quantisation steps required even further.

5. The system-level model and the circuit implementation were tested only with AEG signals. Other biosignals such as EEG, ECoG, ECG etc. and other types of input signals which are sparse in the time domain could be used for testing the synchronous LC ADC and optimising the same for the specific application.

6. The split resolution technique developed in this work was tested only at the system level in VerilogA models. It could be integrated into an LC ADC (operating asynchronously) to test it at the circuit level.

Appendices

A

MODELS

The code for the models used to test the synchronous LC ADC at the system level are given below. Some models were developed in VerilogA and some were developed in MATLAB.

A.1. VERILOGA MODEL FOR SINGLE-CHANNEL SYNCHRONOUS LC ADC

This model takes inputs and quantises according to the level-crossing quantisation approach. The split resolution technique can be enabled or disabled in the model. Please check comments for the same.

A.1.1. SAMPLE AND HOLD

```
`include "constants.vams"
`include "disciplines.vams"

module snh_model(vin, clk, vsamp);
electrical vin, clk, vsamp;
real usamp;
analog begin
  @(cross(V(clk)-1.8,+1)) begin
    usamp = V(vin);
  end
V(vsamp) <+ usamp;
end
endmodule
```

A.1.2. COMPARATOR

```
`include "constants.vams"
`include "disciplines.vams"

module cmp(inp,inm,outp);
electrical out;
electrical inp,inm,outp;
parameter real out_h = 1.8;
parameter real out_l = 0;
real s_outp;

analog begin
  if(V(inp)>V(inm)) begin
    s_outp = out_h;
  end
  else if(V(inp)<V(inm)) begin
      s_outp = out_l;
  end

V(outp) <+ s_outp;

end
endmodule
```

A.1.3. CONTROL LOGIC BLOCK AND DAC

```
`include "constants.vams"
`include "disciplines.vams"

module logic_sync_model_1ch(en,clk,co,vref,lvl,done);
electrical en,co,clk,vref,lvl,done;
real r;      // DAC output (reference voltage)
real vr=0.0046875; //LSB step value
integer s=128;  //starting reference step (mid-swing)
integer steps;  //number of steps covered for quantisation
integer dir=0;  //direction of quantisation
integer d=0;    //to show that quantisation is done
analog begin
  @(cross(V(en)-1.8,+1)) begin
    d=0;
    dir=0;
    steps=0;
  end
  @(cross(V(clk)-1.8,+1)) begin
  if(V(co)==1.8 && d==0 && V(en)==1.8) begin
    s=s+1;
    steps=steps+1;
    if(s>=148 || s<=108) begin
// Comment the following line to disable split resolution
      s=s+1;
    end
    if(dir==0) begin
      dir=1;
    end
  end
  if(V(co)==0 && d==0 && V(en)==1.8) begin
    s=s-1;
    steps=steps-1;
    if(s<=108 || s>=148) begin
// Comment the following line to disable split resolution
      s=s-1;
    end
    if(dir==0) begin
      dir=-1;
    end
  end
  if(((V(co)==0 && dir==1)||(V(co)==1.8 && dir==-1)) && V(en)==1.8) begin
    if(dir==1 && d==0) begin
      s=s+1;
      steps=steps+1;
    end
    else if(dir==-1 && d==0) begin
      s=s-1;
      steps=steps-1;
    end
    d=1;
  end
  end
  r=(s)*vr;
    $strobe("V(r)=%f, dir=%d, done=%d, T=%e",r,dir,d,$realtime);
  V(vref)<+absdelay(r,100n);
  V(lvl)<+steps;
  V(done)<+d;
end
endmodule
```

A.2. VERILOGA MODEL FOR MULTI-CHANNEL SYNCHRONOUS LC ADC

The comparator and sample-and-hold block models given in previous section can be reused with this model. The other blocks are given here.

A.2.1. COUNTER FOR ANALOG MUX

```
`include "constants.vams"
`include "disciplines.vams"

module counter_4(clk,sel1,sel0,rst);
electrical clk,sel1,sel0,rst;
integer cnt=0;
real s1,s0;
analog begin
  @(cross(V(clk)-1.8,+1)) begin
    cnt=cnt+1;
    if(cnt==4) begin
      cnt=0;
    end
  end
  if(cnt==0) begin
    s1=0;
    s0=0;
  end
  if(cnt==1) begin
    s1=0;
    s0=1.8;
  end
  if(cnt==2) begin
    s1=1.8;
    s0=0;
  end
  if(cnt==3) begin
    s1=1.8;
    s0=1.8;
  end
V(sel1) <+ s1;
V(sel0) <+ s0;
end
endmodule
```

A.2.2. ANALOG MUX

```
`include "constants.vams"
`include "disciplines.vams"

module anamux_4(vin1,vin2,vin3,vin4,sel1,sel0,vsel);
electrical vin1,vin2,vin3,vin4,sel1,sel0,vsel;
real vo;
analog begin
  if(V(sel1)==1.8 && V(sel0)==1.8) begin
    vo = V(vin4);
  end
  if(V(sel1)==1.8 && V(sel0)==0) begin
    vo = V(vin3);
  end
  else if(V(sel1)==0 && V(sel0)==1.8) begin
    vo = V(vin2);
  end
  else if(V(sel1)==0 && V(sel0)==0) begin
    vo = V(vin1);
  end
V(vsel) <+ vo;
end
endmodule
```

A.2.3. LOGIC CONTROL BLOCK AND DAC

```
`include "constants.vams"
`include "disciplines.vams"

module logic_sync(en,co,clk,vref,lvl,done,change);
electrical en,co,clk,vref,lvl,done,change;
real r;
parameter real vr=0.0046875;
integer s=0;
integer dir=0;
integer d=0;
integer ch=0;
integer lv[0:3]={0,0,0,0};
integer ch_l,ch_n;
analog begin
  @(cross(V(en)-1.8,+1)) begin
    d=0;
    dir=0;
    ch=ch+1;
    if(ch==4) begin
      ch=0;
    end
    s=lv[ch];
    ch_l=lv[ch];
    $strobe("ref=%f, ch=%d, done=%d, T=%e",s,ch,d,$realtime);
  end
  @(cross(V(clk)-1.8,+1)) begin
  if(V(co)==1.8 && d==0) begin
    s=s+1;
    if(dir==0) begin
      dir=1;
    end
  end
    else if(V(co)==0 && d==0) begin
    s=s-1;
    if(dir==0) begin
      dir=-1;
    end
  end
  else if(((V(co)==0 && dir==1)||(V(co)==0 && dir==-1)) && d==0) begin
    if(dir==1) begin
      s=s+1;
    end
    else if(dir==-1) begin
      s=s-1;
    end
    d=1;
    dir=0;
    ch_n=s-ch_l;
  end
  lv[ch]=s;
  end
  r=(s)*vr;
  V(change)<+ch_n;
  V(vref)<+absdelay(r,10n);
  V(lvl)<+s;
  V(done)<+d;
end
endmodule
```

A.3. MATLAB MODEL OF SINGLE CHANNEL SYNCHRONOUS LC ADC

The following MATLAB model is used to estimate the maximum number of quantisation steps required for quantisation of a particular signal with a specific resolution. The input signal 'vin' is assumed to be a sampled signal without a DC offset. The outputs are -

- 'tot' - total number of quantisation steps performed for the entire simulation
- 'totbit' - total number of output bits generated
- 'outbit' - output bits generated for each sample
- 'stepsint' - difference in number of LSB steps from previous sample
- 'maxstep' - maximum number of quantisation steps required
- 'minstep' - minimum number of quantisation steps required

```
function [tot,totbit,outbit,stepsint,comps,maxstep,minstep] = samplelc(vin);
        l=length(vin);
        ref=0.0046875;   %reference voltage of 1 LSB step
        lvl=zeros(size(vin));
        diff=zeros(size(vin));
        outbit=zeros(size(vin));
        stepsint=zeros(size(vin));
        comps=zeros(size(vin));
        stepsint(1)=ceil(vin(1)/ref);
        [m,n]=size(de2bi(ceil(abs(stepsint(1)))));
        outbit(1)=n+1;
        totbit=outbit(1);
        tot=abs(stepsint(1));
        comps(1)=tot;
        diff(1)=0;
        lvl(1)=stepsint(1);
        maxstep=stepsint(1);
        minstep=stepsint(1);
        for(i=1:1:l-1)
            diff(i+1)=(vin(i+1)-lvl(i)*ref)/ref;
            if(diff(i+1)<=1 && diff(i+1)>0)
                stepsint(i+1)=1;
            elseif(diff(i+1)>=-1 && diff(i+1)<0)
                stepsint(i+1)=-1;
            elseif(diff(i+1)>1)
                stepsint(i+1)=ceil(diff(i+1));
            elseif(diff(i+1)<-1)
                stepsint(i+1)=floor(diff(i+1));
            end
            lvl(i+1)=lvl(i)+stepsint(i+1);
            [m,n]=size(de2bi(abs(stepsint(i+1))));
            if(n==1)
                outbit(i+1)=1;
            else
                outbit(i+1)=n+1;
            end
            totbit=totbit+outbit(i+1);
            if(abs(stepsint(i+1))<=1)
                tot=tot+2;
            else
                tot=tot+abs(stepsint(i+1))+1;
            end
            comps(i+1)=tot;
            if(stepsint(i+1)>maxstep)
                maxstep=stepsint(i+1);
            end
            if(stepsint(i+1)<minstep)
                minstep=stepsint(i+1);
            end
        end
    end
```

A

A.4. MATLAB MODEL OF SINGLE CHANNEL SYNCHRONOUS LC ADC WITH SPLIT RESOLUTION

The following MATLAB model is used to implement the split resolution technique in the synchronous LC ADC and a threshold is set for the same. The outputs are same as that in the previous model.

```
function [tot,totbit,outbit,stepsint,comps,maxstep,minstep] = samplelc_sr(vin);
        l=length(vin);
        ref=0.0046875;
        th=0.046875; //threshold set at +/- 10 LSB steps
        lvl=zeros(size(vin));
        diff=zeros(size(vin));
        outbit=zeros(size(vin));
        stepsint=zeros(size(vin));
        comps=zeros(size(vin));
        stepsint(1)=ceil(vin(1)/ref);
        [m,n]=size(de2bi(ceil(abs(stepsint(1)))));
        outbit(1)=n+1;
        totbit=outbit(1);
        tot=abs(stepsint(1));
        comps(1)=tot;
        diff(1)=0;
        lvl(1)=stepsint(1);
        maxstep=stepsint(1);
        minstep=stepsint(1);
        for(i=1:1:l-1)
            if(abs(lvl(i))<=th && vin(i+1)<=th)
                diff(i+1)=(vin(i+1)-lvl(i)*ref)/ref;
            elseif(abs(lvl(i))<=th && vin(i+1)>=th)
                diff(i+1)=(vin(i+1)-th)/(2*ref) + (th-lvl(i)*ref)/ref;
            elseif(abs(lvl(i))<=th && vin(i+1)<=-th)
                diff(i+1)=(vin(i+1)-(-th))/(2*ref) + (-th-lvl(i)*ref)/ref;
            elseif(lvl(i)>th && vin(i+1)>th)
                diff(i+1)=(vin(i+1)-lvl(i)*ref)/(2*ref);
            elseif(lvl(i)<-th && vin(i+1)<-th)
                diff(i+1)=(vin(i+1)-lvl(i)*ref)/(2*ref);
            elseif(lvl(i)>th && vin(i+1)<-th)
                diff(i+1)=(lvl(i)*ref-th)/(2*ref)+6e3/ref+(vin(i+1)-(-th))/(2*ref);
            elseif(lvl(i)<-th && vin(i+1)>th)
                diff(i+1)=(lvl(i)*ref-(-th))/(2*ref)+6e3/ref+(vin(i+1)-th)/(2*ref);
            end
            if(diff(i+1)<=1 && diff(i+1)>0)
                stepsint(i+1)=1;
            elseif(diff(i+1)>=-1 && diff(i+1)<0)
                stepsint(i+1)=-1;
            elseif(diff(i+1)>1)
                stepsint(i+1)=ceil(diff(i+1));
            elseif(diff(i+1)<-1)
                stepsint(i+1)=floor(diff(i+1));
            end
            lvl(i+1)=lvl(i)+stepsint(i+1);
            [m,n]=size(de2bi(abs(stepsint(i+1))));
            if(n==1)
                outbit(i+1)=1;
            else
                outbit(i+1)=n+1;
            end
            totbit=totbit+outbit(i+1);
            if(abs(stepsint(i+1))<=1)
                tot=tot+3;
            else
                tot=tot+abs(stepsint(i+1))+2;
            end
            comps(i+1)=tot;
            if(stepsint(i+1)>maxstep)
                maxstep=stepsint(i+1)+1;
            end
            if(stepsint(i+1)<minstep)
                minstep=stepsint(i+1)+1;
```

```
            end
        end
    end
```

A.5. MATLAB MODEL OF MULTICHANNEL SYNCHRONOUS LC ADC WITH A SHARED MEMORY ACROSS ALL CHANNELS

The following MATLAB model takes input signals from multiple channels and quantises them using a single memory. The input signal is a single stream consisting of samples from all channels.

```
function [tot,totbit,outbit,stepsint,comps,maxstep,minstep] = samplelc_onemem(vin)
        l=length(vin);
        ref=4.6875;
        lvl=zeros(size(vin));
        diff=zeros(size(vin));
        outbit=zeros(size(vin));
        stepsint=zeros(size(vin));
        comps=zeros(size(vin));
        stepsint(1)=ceil(vin(1)/ref);
        [m,n]=size(de2bi(ceil(abs(stepsint(1)))));
        outbit(1)=n+1;
        totbit=outbit(1);
        tot=abs(stepsint(1));
        comps(1)=abs(stepsint(1));
        diff(1)=0;
        lvl(1)=stepsint(1);
        maxstep=stepsint(1);
        minstep=stepsint(1);
        for(i=1:1:l-1)
            diff(i+1)=(vin(i+1)-lvl(i)*ref)/ref;
            if(diff(i+1)<=1 && diff(i+1)>0)
                stepsint(i+1)=1;
            elseif(diff(i+1)>=-1 && diff(i+1)<0)
                stepsint(i+1)=-1;
            elseif(diff(i+1)>1)
                stepsint(i+1)=ceil(diff(i+1));
            elseif(diff(i+1)<-1)
                stepsint(i+1)=floor(diff(i+1));
            end
            lvl(i+1)=lvl(i)+stepsint(i+1);
            [m,n]=size(de2bi(abs(stepsint(i+1))));
            if(n==1)
                outbit(i+1)=1;
            else
                outbit(i+1)=n+1;
            end
            totbit=totbit+outbit(i+1);
            if(abs(stepsint(i+1))<=1)
                tot=tot+2;
            else
                tot=tot+abs(stepsint(i+1))+1;
            end
            comps(i+1)=tot;
            if(stepsint(i+1)>maxstep)
                maxstep=stepsint(i+1);
            end
            if(stepsint(i+1)<minstep)
                minstep=stepsint(i+1);
            end
        end
    end
```

B

DIGITAL LOGIC

The Verilog code for the Control Logic Blocks are added below -

B.1. CONTROL LOGIC BLOCK OF SYNCHRONOUS LC ADC WITH 8-BIT QUANTISER RESOLUTION AND SPLIT RESOLUTION TECHNIQUE

```
//In this block the split resolution threshold is pre-set at +/- 15\% from the mid-swing level. This can be changed according to the requirements.

module sync_logic_8b_sr (input clk,rst,smp_rdy,co,cmp_done, output sw7,sw6,sw5,sw4,sw3,sw2,sw1,sw0, output reg cmp,done,dac_set, output [8:0] bout);
reg [7:0] switch [0:3];
reg first;
reg co1;
reg trig;
reg l;
reg [1:0] sw;
reg [7:0] sw_last;
reg [7:0] step;
reg [7:0] diff;
assign bout = {diff,co1};
assign sw7 = switch[sw][7];
assign sw6 = switch[sw][6];
assign sw5 = switch[sw][5];
assign sw4 = switch[sw][4];
assign sw3 = switch[sw][3];
assign sw2 = switch[sw][2];
assign sw1 = switch[sw][1];
assign sw0 = switch[sw][0];

always@(posedge clk) begin
  if(rst==1) begin
    cmp<=0;
    done<=1;
    first<=0;
    dac_set<=0;
    l<=0;
    switch[0]<=8'b10000000;
    switch[1]<=8'b10000000;
    switch[2]<=8'b10000000;
    switch[3]<=8'b10000000;
    co1<=0;
    trig<=0;
    sw<=2'b11;
    step<=0;
    diff<=0;
  end
  else begin
```

```
    if(smp_rdy==1 && trig==0) begin
      first<=1;
      cmp<=0;
      done<=0;
      co1<=0;
      dac_set<=1;
      trig<=1;
      sw<=sw+1;
      sw_last<=0;
      step<=0;
      diff<=0;
    end
    if(smp_rdy<=0) begin
      trig<=0;
    end
    if(dac_set==1) begin
      cmp<=1;
      dac_set<=0;
    end
    if(cmp_done==1) begin
      if(first==1) begin
        sw_last<=switch[sw];
        first<=0;
        if(co==0 && (switch[sw]<8'd108 || switch[sw]>8'd148) && switch[sw]>=8'd2)
    begin
          switch[sw]<=switch[sw]-2;
        end
        else if(co==0 && switch[sw]>=8'd108 && switch[sw]<=8'd148) begin
          switch[sw]<=switch[sw]-1;
        end
        if(co==1 && (switch[sw]<8'd108 || switch[sw]>8'd148) && switch[sw]<=8'd253)
    begin
          switch[sw]<=switch[sw]+2;
        end
        else if(co==1 && switch[sw]<=8'd148  && switch[sw]>=8'd108) begin
          switch[sw]<=switch[sw]+1;
        end
        dac_set<=1;
        co1<=co;
        step<=step+1;
      end
      else if(first==0) begin
        if(co!=co1 || l==1) begin
          if(l==0 && (switch[sw]<8'd108 || switch[sw]>8'd148)) begin
            l<=1;
            if(co1==1) begin
              switch[sw]<=switch[sw]-1;
              step<=step-1;
            end
            else begin
              switch[sw]<=switch[sw]+1;
              step<=step-1;
            end
            dac_set<=1;
          end
          else if(l==1) begin
            l<=0;
            if(co1==1 && co==1) begin
              step<=step+1;
              switch[sw]<=switch[sw]+1;
            end
            else if(co1==0 && co==0) begin
              step<=step+1;
              switch[sw]<=switch[sw]-1;
            end
            done<=1;
            dac_set<=0;
          end
          else begin
```

```
108             done<=1;
109             dac_set<=0;
110           end
111         end
112         else if(co==1) begin
113           if((switch[sw]<8'd108 || switch[sw]>8'd148) && switch[sw]<=8'd253) begin
114             switch[sw]<=switch[sw]+2;
115             dac_set<=1;
116             step<=step+2;
117           end
118           else if(switch[sw]>=8'd108 && switch[sw]<=8'd148) begin
119             switch[sw]<=switch[sw]+1;
120             dac_set<=1;
121             step<=step+1;
122           end
123           else begin
124             done<=1;
125             dac_set<=0;
126           end
127         end
128         else if(co==0) begin
129           if((switch[sw]<8'd108 || switch[sw]>8'd148) && switch[sw]>=8'd2) begin
130             switch[sw]<=switch[sw]-2;
131             dac_set<=1;
132             step<=step+2;
133           end
134           else if(switch[sw]>=8'd108 && switch[sw]<=8'd148) begin
135             switch[sw]<=switch[sw]-1;
136             dac_set<=1;
137             step<=step+1;
138           end
139           else begin
140             done<=1;
141             dac_set<=0;
142           end
143         end
144       end
145       cmp<=0;
146     end
147     if(done==1) begin
148       if(step==1) begin
149         diff<=0;
150       end
151       else begin
152         diff<=step;
153       end
154     end
155   end
156 end
157
158 endmodule
```

B.2. Control Logic Block of multichannel synchronous LC ADC with 10-bit resolution


```
module sync_logic_10bit (input clk,rst,smp_rdy,co,cmp_done, output sw9,sw8,sw7,sw6,
    sw5,sw4,sw3,sw2,sw1,sw0, output reg cmp,done,dac_set, output [10:0] bout);
reg [9:0] switch [0:3];
reg first;
reg co1;
reg trig;
reg [1:0] sw;
reg [9:0] sw_last;
reg [9:0] step;
reg [9:0] diff;
assign bout = {diff,co1};
assign sw9 = switch[sw][9];
assign sw8 = switch[sw][8];
assign sw7 = switch[sw][7];
assign sw6 = switch[sw][6];
assign sw5 = switch[sw][5];
assign sw4 = switch[sw][4];
assign sw3 = switch[sw][3];
assign sw2 = switch[sw][2];
assign sw1 = switch[sw][1];
assign sw0 = switch[sw][0];

always@(posedge clk) begin
  if(rst==1) begin
    cmp<=0;
    done<=1;
    first<=0;
    dac_set<=0;
    switch[0]<=10'b1000000000;
    switch[1]<=10'b1000000000;
    switch[2]<=10'b1000000000;
    switch[3]<=10'b1000000000;
    co1<=0;
    trig<=0;
    sw<=2'b11;
    step<=0;
    diff<=0;
  end
  else begin
    if(smp_rdy==1 && trig==0) begin
      first<=1;
      cmp<=0;
      done<=0;
      co1<=0;
      dac_set<=1;
      trig<=1;
      sw<=sw+1;
      sw_last<=0;
      step<=0;
      diff<=0;
    end
    if(smp_rdy<=0) begin
      trig<=0;
    end
    if(dac_set==1) begin
      cmp<=1;
      dac_set<=0;
    end
    if(cmp_done==1) begin
      if(first==1) begin
        sw_last<=switch[sw];
        first<=0;
        if(co==0 && switch[sw]>10'b0000000000) begin
          switch[sw]<=switch[sw]-1;
        end
```

```
65          else if(co==1 && switch[sw]<10'b1111111111) begin
66            switch[sw]<=switch[sw]+1;
67          end
68          dac_set<=1;
69          co1<=co;
70          step<=step+1;
71        end
72        else if (first==0) begin
73          if(co!=co1) begin
74            done<=1;
75            dac_set<=0;
76          end
77          else if(co==1) begin
78            if(switch[sw]<10'b1111111111) begin
79              switch[sw]<=switch[sw]+1;
80              dac_set<=1;
81              step<=step+1;
82            end
83            else begin
84              done<=1;
85              dac_set<=0;
86            end
87          end
88          else if(co==0) begin
89            if(switch[sw]>10'b0000000000) begin
90              switch[sw]<=switch[sw]-1;
91              dac_set<=1;
92              step<=step+1;
93            end
94            else begin
95              done<=1;
96              dac_set<=0;
97            end
98          end
99        end
100       cmp<=0;
101     end
102     if(done==1) begin
103       if(step==1) begin
104         diff<=0;
105       end
106       else begin
107         diff<=step;
108       end
109     end
110   end
111 end
112
113 endmodule
```

B.3. Control Logic Block of SAR ADC

```
module sar_logic_8b (input clk,rst,smp_rdy,co,cmp_done, output sw7,sw6,sw5,sw4,sw3,
    sw2,sw1,sw0, output reg cmp,dac_set, output reg [7:0] bout);

reg [7:0] switch;
reg [7:0] quant;
reg [2:0] qn;

assign sw7 = switch[7];
assign sw6 = switch[6];
assign sw5 = switch[5];
assign sw4 = switch[4];
assign sw3 = switch[3];
assign sw2 = switch[2];
assign sw1 = switch[1];
assign sw0 = switch[0];

always@(posedge clk) begin
  if(rst==1) begin
    cmp<=0;
    dac_set<=0;
    qn<=3'b111;
    switch<=8'b00000000;
    quant<=8'b00000000;
    bout<=8'b00000000;
  end
  else if(smp_rdy==0) begin
    cmp<=0;
    dac_set<=1;
    switch<=8'b10000000;
    bout<=quant;
    qn<=3'b111;
  end
  else if(smp_rdy==1 && qn>=0) begin
    if(dac_set==1) begin
      cmp<=1;
      dac_set<=0;
    end
    if(cmp_done==1) begin
      if(co==1) begin
        switch[qn]<=1;
        quant[qn]<=1;
      end
      else begin
        switch[qn]<=0;
        quant[qn]<=0;
      end
      if(qn!=0) begin
        switch[qn-1]<=1;
        qn<=qn-1;
      end
      dac_set<=1;
      cmp<=0;
    end
  end
end

endmodule
```

C

SIMULATION SETUP

The circuit schematic design of the synchronous LC ADC discussed in the previous section is simulated with a supply voltage of 1.8 V.

The transient simulations are performed with different input signals such as -

1. sinusoidal input signal,
2. AEG input signal with induced atrial fibrillation condition, and
3. AEG input signal in sinus rhythm condition

The sinusoidal input signal is connected to 4 input channels with a delay of 120 μs in between each successive channel. Each sinusoidal input signal has a frequency of 125 Hz and a peak-peak amplitude of 1.2 V as it is also the common mode input range of the comparators.

The sampling frequency is set at 1 kS/s as the signal bandwidth of AEG signals is 0.5 - 400 Hz. 1024 samples are quantised in each channel to determine the performance of the ADCs. Thus, the sinusoidal input signal is set at 125 Hz such that FFT can be calculated with 128 bins. The quantisation clock signal given to the ADC has a frequency of 500 kHz. This frequency is determined after estimating the maximum number of quantisation steps required for samples of a sinusoidal input signal in a synchronous LC ADC.

The AEG signal with induced atrial fibrillation condition is obtained from Erasmus Medical Center which was originally recorded at 1 kS/s with a resolution of 16 bits [2]. It was converted into ASCII format (with .p extension) in Matlab and used as an input file in Cadence design environment through the 'pwl' option in vsource input (found in 'analoglib'). During conversion into ASCII format in Matlab the signal is scaled so as to fit within the common mode input range of the ADC, i.e. 0 - 1.2 V. The cadence design environment allows signal input files with a limited size. Hence, the length of input file was limited to 1 s (therefore, 1000 samples). The AEG input signal with normal sinus rhythm is also converted to ASCII format and used as input to the circuit schematic of the synchronous LC ADC.

The simulation of the ADC creates a considerable amount of data that cannot be imported to MATLAB for reconstruction. Hence, strobing is used to sample the output signals from the simulation such that the output of the quantiser after the completion of quantisation of each sample is recorded. The settings for strobing the output can be found in ADE inside 'Options -> Output' of the analysis selection pane.

Generally analog mixed-signal (AMS) simulators are used to simulate the analog circuit blocks along with digital logic blocks defined in Verilog/VHDL. This method is preferred as the digital logic circuits are complex and at higher operating frequencies they cause the simulation time to increase. However, due to issues with the AMS simulator in the current setup, the digital logic block defined in Verilog could not be simulated. Hence, the synthesized CMOS circuit of the digital logic block was simulated along with

the analog circuit blocks. This impacted simulation time as well, and hence only limited results could be obtained to determine the overall performance of the ADCs.

D

PAPER ACCEPTED AT BIOCAS 2019

The following paper based on the current work is accepted for presentation at BioCAS 2019. After the presentation the paper will be published in the conference proceedings as well as in IEEE Xplore database.

Activity Dependent Multichannel ADC Architecture using Level Crossing Quantisation for Atrial Electrogram Recording

Aurojyoti Das, Samprajani Rout, Alessandro Urso, Wouter A. Serdijn
Section Bioelectronics, TU Delft
Delft, the Netherlands
aurojyoti@gmail.com, S.Rout@tudelft.nl, A.Urso@tudelft.nl, W.A.Serdijn@tudelft.nl

***Abstract*—This paper presents a novel multichannel level-crossing (MLC) ADC architecture aimed at recording atrial electrograms from multiple channels. The proposed architecture combines synchronous sampling with level-crossing (LC) quantisation to achieve activity dependent operation while recording from multiple channels simultaneously. In the proposed architecture the number of comparisons performed by the quantiser to reach a decision is dependent on the activity of the input signal and is 2-3.3 times lower than that in a conventional SAR ADC. The architecture uses one comparator and one reference level instead of two comparators and two reference levels as in conventional LC ADCs. The proposed architecture is modeled in VerilogA and is designed to be implemented in a standard 0.18 um CMOS process. The MLC ADC converts signals from 4 channels simultaneously and achieves an SFDR of 45.5 dB while consuming 8.05 μW of power from a 1.8 V power supply.**

***Index Terms*—event-driven, level crossing, asynchronous, biosignal acquisition, multichannel LC ADC, atrial electrogram**

I. Introduction

Atrial electrograms (AEGs) are recorded from the atrial myocardium to help in deeper diagnosis of the atrial fibrillation condition. The recording is performed by using a patch of 192 electrodes [1]. The current setup uses a 3 m long cable to transmit the acquired signals to an analog front end (AFE) and suffers from the fact that noise and interference can corrupt the signal. To mitigate this, an IC-based AFE is required that can be placed near the electrodes and can thus prevent corruption of the acquired signals.

The AEGs behave similar to regular (surface) electrocardiograms (ECGs) as they have high-amplitude peaks in between time intervals of low activity, making the signals sparse in the time domain. Conventional nyquist rate ADCs sample the input signals at a constant rate irrespective of signal activity and thus do not exploit the temporally sparse property of certain biosignals such as AEGs, ECGs, etc. Asynchronous ADCs such as LC ADCs (shown in Fig. 1(a)) exploit the signal sparsity by waiting for an event to happen (such as the signal crossing over a reference level) instead of sampling it at regular intervals [2]. However, LC ADCs are not compatible with discrete time signal processing (DSP) blocks [3]. Also, they produce higher amounts of data as compared to nyquist rate ADCs, especially at the higher resolutions that are required for signals with high dynamic range such as AEGs [4].

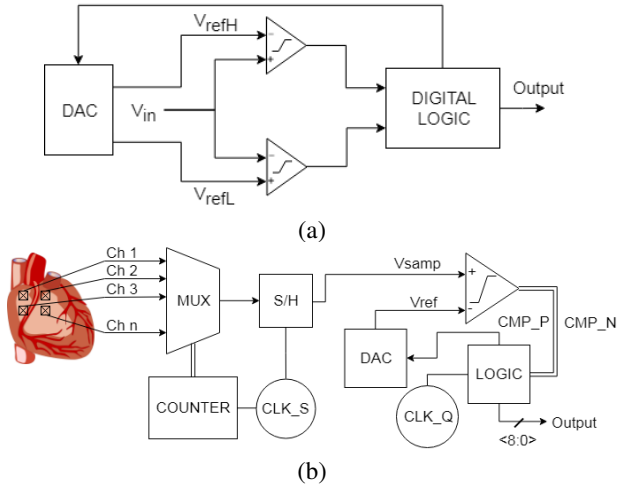


Fig. 1. Block diagram of (a) a conventional LC ADC architecture and (b) the proposed multichannel LC ADC architecture.

Multichannel ADC topologies have been reported in literature, which can be used in the AFE for recording AEGs [5]. The existing multichannel ADCs however are not activity-dependent. LC ADCs are a better choice here as their operation is activity dependent. Currently, multichannel configurations of LC ADCs do not yet exist.

In this paper we propose the novel MLC ADC architecture shown in Fig. 1(b), which combines features of both synchronous and asynchronous recording methods. The signal is acquired synchronously as is done in nyquist rate ADCs. Then the sample is quantised using the level-crossing sampling approach, which makes the quantisation process activity-dependent. Multiple channels are sampled simultaneously by time-multiplexing the ADC across the channels. The quantisation process is configured based on the signal activity. This approach reduces the average number of comparisons required per sample for quantisation of AEGs by up to 3.3 times.

The system design and modelling of the proposed MLC ADC architecture are explained in Section II. The circuit implementation is described in Section III. The simulation results are discussed in Section IV. Finally the conclusions are discussed in Section V.

II. Proposed Architecture

In this section the system-level design considerations and modelling of the proposed MLC ADC architecture are discussed.

A. System Design

In conventional LC ADCs (Fig. 1.(a)), the reference window consists of two reference levels to which the input signal is compared. The reference window follows the input signal as it tries to keep the current value of the input signal within the reference window. Conventional LC ADCs track the signal continuously and generate events when a level crossing is detected. Although this method is signal-driven, it is not power-efficient as the continuous-time comparators draw power all the time. Moreover, the LC ADCs operate in continuous time and hence cannot be configured to convert signals from multiple channels simultaneously, as is done in synchronous ADCs, without increasing the data rate and power consumption considerably. Sampling the input signal synchronously and quantising the sample using level-crossing quantisation allows the conversion to be signal-driven and reduces power consumption as well. Rather than counting the number of LSB steps crossed by the sample from the mean level (half of the common mode input range), the highest reference level crossed by the previous sample is used as the starting point of quantisation for each sample.

Conventional SAR ADCs and similar synchronous ADCs use a fixed number of steps to quantise the sample but in the proposed method the number of steps required is signal-dependent. If the current sample is at the same amplitude level as the previous sample, the quantisation is completed after just two comparisons. In the worst case, the number of levels counted would be equal to 2^N where N is the resolution of the quantiser. However most of the high-amplitude contents of biosignals occur at lower frequency ranges, as shown in the FFT plot of an AEG in Fig. 2.

Hence, the average number of steps required for quantisation of each successive sample in the proposed method would be less than that in conventional synchronous sampling methods. This assumption is verified with MATLAB models, which show that the proposed method can reduce the number of comparisons required by 2-3.3 times depending on the target resolution of the quantiser (10 bit - 6 bit, respectively). The number of comparisons required for conversion of 4 typical AEGs with 8-bit resolution is shown in Fig. 3. The model samples the AEGs at 1 kS/s from 4 channels for 10 s. Thus a total of 40000 samples are quantised. The plot also shows that the number of comparisons increase and even reach the same number of comparisons as that of the multichannel SAR algorithm at higher resolutions. Thus, for lower resolutions the proposed method has an advantage over the conventional SAR algorithm.

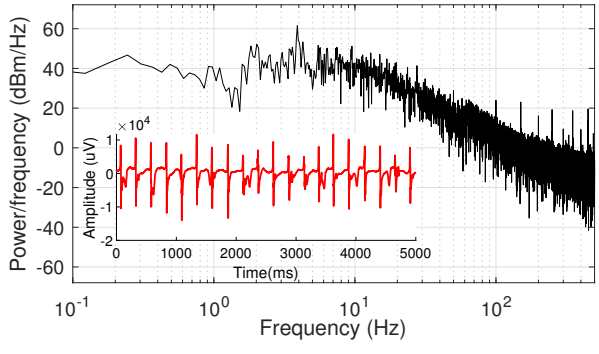


Fig. 2. Power spectral density of AEG (inlay: AEG). Data Courtesy: Erasmus MC, Rotterdam [1]

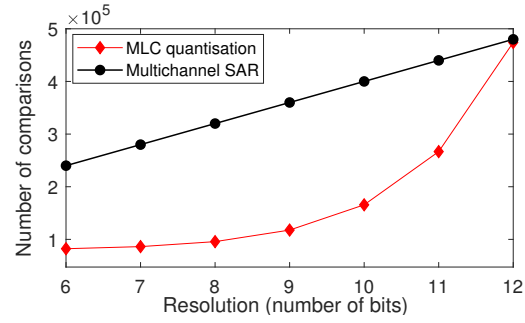


Fig. 3. Estimation of number of comparisons performed by the quantiser for quantisation of AEGs by using the SAR algorithm and by using the proposed method in a MATLAB model for 10 s of conversion from 4 channels.

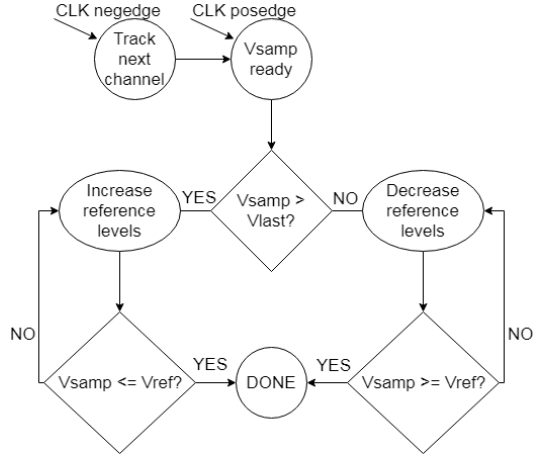


Fig. 4. Flowchart of operation of proposed multichannel LC ADC architecture.

The flowchart of the operation of the proposed MLC ADC architecture is shown in Fig. 4. At the positive edge of CLK, a sample is captured in the Sample & Hold (S/H) block and the highest reference level crossed by the previous sample is loaded in the DAC and compared with the sample. If the sample has a higher amplitude then the reference level is increased by 1 LSB and compared again. This process continues until the sample amplitude is under the reference level and the comparator output is changed. The highest

reference level reached is given as output. The final reference level is then used as the starting point for the quantisation of the next sample. Multiple channels are sampled with the same ADC by time-multiplexing the S/H. The digital logic stores the output of the first comparison of each sample and uses it to reach the end point of quantisation. Moreover, each channel has a set of registers to store the reference level after completion of the quantisation of its corresponding sample.
The clock speed required for quantisation is determined by the signal activity and time required for each conversion. The maximum number of steps covered by the quantiser for each sample is estimated through VerilogA models and thus the clock speed is set to cover the worst case. The requirement for a clock signal for quantisation can be obviated by implementing an asynchronous quantisation method [6].

B. VerilogA modelling

A model of the proposed MLC ADC architecture is developed in VerilogA to verify its functionality. The model is used to convert an AEG and the output is used to reconstruct the AEG in MATLAB. The LSB step size for the conversion is 1mV. The input signal and the quantised samples are shown in Fig. 5. The lower plot of Fig. 5 shows the number of LSB steps covered by the quantiser for each sample. For most of the samples, the number of steps covered is nearly zero since the signal does not change considerably for most periods of time. The maximum number of steps to be covered for a specific signal can be estimated through this model.

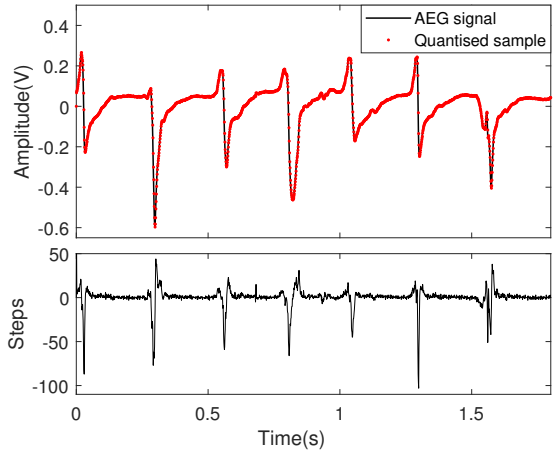


Fig. 5. Waveform showing input AEG waveform and quantized samples in VerilogA model of the proposed MLC ADC architecture. The lower plot shows the number of LSB steps covered by the quantiser for each sample.

III. Circuit Implementation

The proposed MLC ADC architecture is designed to be implemented in TSMC's 0.18 um CMOS process. The ADC supports conversion from 4 channels simultaneously at a resolution of 8 bits.
A S/H block is implemented with a 1 pF hold capacitor. Two transmission gates are used to switch between the 'track' and 'hold' phases. An 8-bit DAC is implemented in binary-weighted configuration with a unit capacitor of 35.6 fF.
The S/H is multiplexed with several channels by using a counter to select the channel number and NMOS transistors as switches. Since the sampling frequency is 1 kS/s, the counter increments at 4 kHz and thus selects the subsequent channel for sampling every 250 ms.
A strong-arm dynamic latch with a PMOS input pair is used for the comparison such that the common mode input range is within 0-1.2 V for a power supply of 1.8 V. A preamplifier is used to reduce kickback noise [5].
The control logic block is designed in Verilog and synthesized using the Synopsys Design Compiler. The control logic uses a 1.6 MHz clock signal to synchronize the quantisation. The speed of the clock signal is calculated after estimation of the maximum number of steps that need to be covered for quantisation of the AEG from the VerilogA model, as discussed in the previous section. The control logic stores the output of the first comparison for each sample and uses this value to determine the end point of quantisation of the sample. A separate memory is used to store the final reference level of each channel and thus the signal in each channel is tracked separately. The signal SW<0:7> is used to load the reference level in the DAC for each comparison. A simplified schematic of the whole ADC is shown in Fig. 6. The timing diagram of operation of the quantiser is shown in Fig. 7. CLK_Q is used to synchronise the quantiser while CLK_S is used to synchronise the sample and hold block. After quantisation is completed for a specific sample the DONE signal is set. When the next sample is ready, DONE is reset and quantisation starts again.

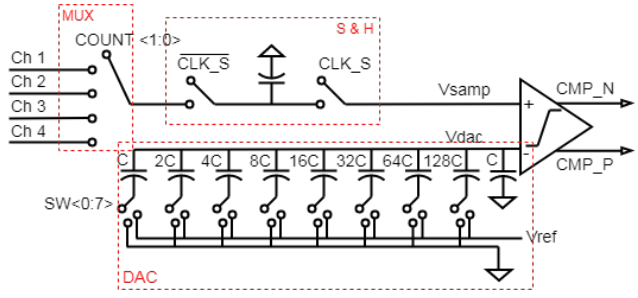


Fig. 6. Circuit schematic of the implemented design of the proposed MLC ADC architecture.

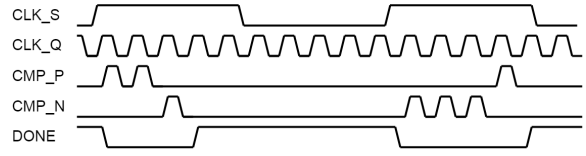


Fig. 7. Timing diagram of operation of the proposed MLC ADC.

IV. Results and Discussion

The transistor-level implementation of the ADC is tested with sinusoidal input signals with V_{p-p} = 1.2 V at 400 Hz on all input channels with a delay of 120 μs between each channel. The signals are sampled at 10 kS/s and the quantisation is performed at 8 MHz while considering the maximum number of steps required by the quantiser to be 100, which is much higher than actually required. The quantized output is used to reconstruct the signal in MATLAB through spline interpolation. The input signals and the quantised samples are shown in Fig. 8.

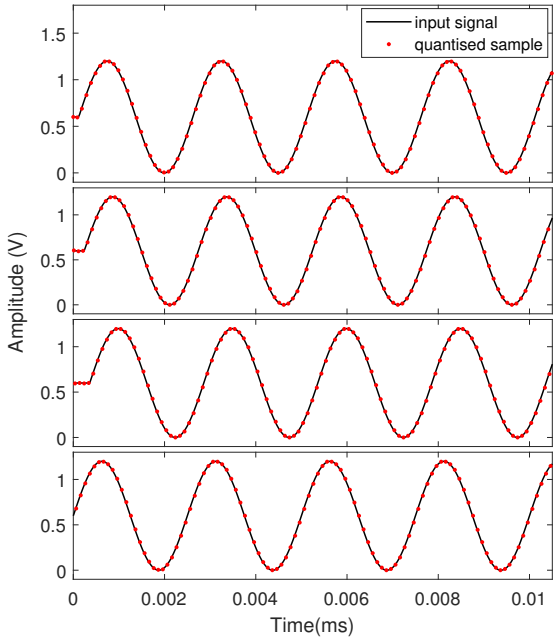


Fig. 8. Reconstructed signals from 4 multiplexed channels

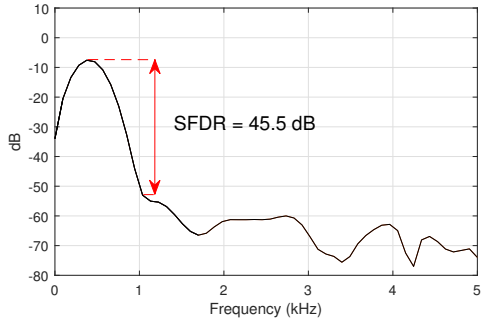


Fig. 9. Output spectrum of the proposed MLC ADC for a 400 Hz sinusoidal input at f_s = 10 kS/s

The implemented design converts signals from 4 channels simultaneously and achieves an SFDR of 45.5 dB (Fig. 9) while consuming 8.05 μW of power. Due to long simulation times, sufficient number of samples could not be obtained to calculate the SNDR and ENOB accurately. However, as shown in Fig. 8 the input signal is reconstructed successfully from the output of the ADC. The proposed MLC ADC architecture is scalable and hence the number of channels, resolution, sampling rate and quantisation time can be reconfigured according to the application requirements. Unlike the conventional LC ADCs in which the amount of data generated doubles with every extra bit of resolution, the proposed MLC ADC produces only n bits of data at the sampling frequency. The differences between conventional ADC architectures and the proposed ADC architecture are summarised in Table 1.

TABLE I
Comparison of Architectures

Feature	***LC ADC***	***Synchronous ADC***	***Proposed MLC ADC***
Activity dependent	Yes	No	Yes
Multichannel	No	Yes	Yes
Comparators	2	1	1
Operation	Asynchronous	Synchronous	Synchronous
Data rate (max)*	$\frac{2\pi f.Vmax}{Vref}$	F_s	F_s
Data volume (bits)#	2	n	n

* Here f refers to bandwidth of the input signal, Vmax refers to maximum input signal amplitude and F_s refers to sampling rate.
\# Here n refers to resolution of the ADC

V. Conclusion

The proposed MLC ADC architecture combines synchronous sampling with level-crossing quantisation and is demonstrated using a VerilogA model and transistor-level simulations. The motivation behind the design of the architecture and its benefits are discussed. The MLC ADC achieves lower data rate as compared to conventional LC ADCs and it is shown that for AEG signals the number of comparisons required by the MLC ADC is reduced by 2-3.3 times as compared to conventional multichannel SAR ADCs.

References

[1] A. Yaksh et al., A novel intra-operative, high-resolution atrial mapping approach, J. Interv. Card. Electrophysiol., vol. 44, no. 3, pp. 221225, 2015.

[2] S. Patil, A. Ratiu, D. Morche and Y. Tsividis, "A 3–10 fJ/conv-step Error-Shaping Alias-Free Continuous-Time ADC," in IEEE Journal of Solid-State Circuits, vol. 51, no. 4, pp. 908-918, April 2016.

[3] B. Schell and Y. Tsividis, A continuous-time ADC/DSP/DAC system with no clock and with activity-dependent power dissipation, IEEE J. Solid-State Circuits, vol. 43, no. 11, pp. 24722481, Nov. 2008.

[4] Y. Li, D. Zhao, and W. A. Serdijn, A sub-microwatt asynchronous level-crossing ADC for biomedical applications, IEEE Trans. Biomed. Circuits Syst., vol. 7, no. 2, pp. 149157, 2013.

[5] F. Shahrokhi, K. Abdelhalim, D. Serletis, P. L. Carlen, and R. Genov, The 128-channel fully differential digital integrated neural recording and stimulation interface, IEEE Trans. Biomed. Circuits Syst., vol. 4, no. 3, pp. 149161, 2010.

[6] P. Harpe, E. Cantatore, and A. Van Roermund, A 10b/12b 40 kS/s SAR ADC with data-driven noise reduction achieving up to 10.1b ENOB at 2.2 fJ/conversion-step, IEEE J. Solid-State Circuits, vol. 48, no. 12, pp. 30113018, 2013.

BIBLIOGRAPHY

[1] C Schilling. *Analysis of atrial electrogram*, volume 17. 2012. ISBN 978-3-86644-894-0.

[2] Ameeta Yaksh, Lisette J.M.E. van der Does, Charles Kik, Paul Knops, Frans B.S. Oei, Pieter C. van de Woestijne, Jos A. Bekkers, Ad J.J.C. Bogers, Maurits A. Allessie, and Natasja M.S. de Groot. A novel intra-operative, high-resolution atrial mapping approach. *Journal of Interventional Cardiac Electrophysiology*, 44(3):221–225, 2015. ISSN 15728595. doi: 10.1007/s10840-015-0061-x.

[3] Farzaneh Shahrokhi, Karim Abdelhalim, Demitre Serletis, Peter L. Carlen, and Roman Genov. The 128-channel fully differential digital integrated neural recording and stimulation interface. *IEEE Transactions on Biomedical Circuits and Systems*, 4(3):149–161, 2010. ISSN 19324545. doi: 10.1109/TBCAS.2010.2041350.

[4] Hung Yen Tai, Yao Sheng Hu, Hung Wei Chen, and Hsin Shu Chen. A 0.85fJ/conversion-step 10b 200kS/s subranging SAR ADC in 40nm CMOS. *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, 57:196–197, 2014. ISSN 01936530. doi: 10.1109/ISSCC.2014.6757397.

[5] B. Schell and Y. Tsividis. A continuous-time adc/dsp/dac system with no clock and with activity-dependent power dissipation. *IEEE Journal of Solid-State Circuits*, 43(11):2472–2481, Nov 2008. ISSN 0018-9200. doi: 10.1109/JSSC.2008.2005456.

[6] A. L. Mansano, Y. Li, S. Bagga, and W. A. Serdijn. An asynchronous event-driven data transmitter for wireless ecg sensor nodes. In *2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings*, pages 404–407, Oct 2014. doi: 10.1109/BioCAS.2014.6981748.

[7] Thomas Marisa, T. Niederhauser, A. Haeberlin, R. A. Wildhaber, R. Vogel, J. Goette, and M. Jacomet. Pseudo Asynchronous Level Crossing adc for ECG Signal Acquisition. *IEEE Transactions on Biomedical Circuits and Systems*, 11(2):267–278, 2017. ISSN 19324545. doi: 10.1109/TBCAS.2016.2619858.

[8] Wei Tang, Ahmad Osman, Dongsoo Kim, Brian Goldstein, Chenxi Huang, Berin Martini, Vincent A. Pieribone, and Eugenio Culurciello. Continuous time level crossing sampling ADC for bio-potential recording systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(6):1407–1418, 2013. ISSN 15498328. doi: 10.1109/TCSI.2012.2220464.

[9] Yongjia Li, Duan Zhao, and Wouter A. Serdijn. A sub-microwatt asynchronous level-crossing ADC for biomedical applications. *IEEE Transactions on Biomedical Circuits and Systems*, 7(2):149–157, 2013. ISSN 19324545. doi: 10.1109/TBCAS.2013.2254484.

[10] Bob Schell and Yannis Tsividis. A Continuous-Time ADC/DSP/DAC System With No Clock and With Activity-Dependent Power Dissipation. *IEEE Journal of Solid-State Circuits*, 43(11):2472–2481, nov 2008. ISSN 0018-9200. doi: 10.1109/JSSC.2008.2005456. URL http://ieeexplore.ieee.org/document/4685415/.

[11] Xiaoyang Zhang, Student Member, and Yong Lian. A 300-mV 220-nW Event-Driven ADC With. 8(6): 834–843, 2015.

[12] Nassim Ravanshad and Hamidreza Rezaee-dehsorkh. An Event-Based ECG-Monitoring and QRS-detection System Based on Level-Crossing Sampling. pages 302–307, 2017.

[13] Michal Maslik, Yan Liu, Tor Sverre Lande, and Timothy G. Constandinou. Continuous-Time Acquisition of Biosignals Using a Charge-Based ADC Topology. *IEEE Transactions on Biomedical Circuits and Systems*, 12(3):471–482, 2018. ISSN 19324545. doi: 10.1109/TBCAS.2018.2817180.

[14] Colin Weltin-Wu and Yannis Tsividis. An event-driven clockless level-crossing ADC with signal-dependent adaptive resolution. *IEEE Journal of Solid-State Circuits*, 48(9):2180–2190, 2013. ISSN 00189200. doi: 10.1109/JSSC.2013.2262738.

[15] A Error-shaping Alias-free and Continuous-time Adc. A 3–10 fJ/conv-step Error-Shaping Alias-Free Continuous-Time ADC. 51(4):908–918, 2016.

[16] A. L. Mansano, Y. Li, S. Bagga, and W. A. Serdijn. An autonomous wireless sensor node with asynchronous ecg monitoring in 0.18μm cmos. *IEEE Transactions on Biomedical Circuits and Systems*, 10 (3):602–611, June 2016. ISSN 1932-4545. doi: 10.1109/TBCAS.2015.2495272.

[17] Michal Maslik, Tor Sverre Lande, and Timothy G. Constandinou. A Clockless Method of Flicker Noise Suppression in Continuous-Time Acquisition of Biosignals. *2018 IEEE Biomedical Circuits and Systems Conference, BioCAS 2018 - Proceedings*, pages 1–4, 2018. doi: 10.1109/BIOCAS.2018.8584788.

[18] Michael Trakimas and Sameer R. Sonkusale. An Adaptive Resolution Asynchronous ADC Architecture for Data Compression in Energy Constrained Sensing Applications. *Tcasi*, 58(5):921–934, 2011. ISSN 1549-8328. doi: 10.1109/TCSI.2010.2092132. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5672382.

[19] Yongjia Li. *Level-Crossing ADCs and Their Applications in Biomedical Readout Systems*. 2015. ISBN 9789461865144.

[20] M. Kurchuk and Y. Tsividis. Signal-dependent variable-resolution clockless a/d conversion with application to continuous-time digital signal processing. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57(5):982–991, May 2010. ISSN 1549-8328. doi: 10.1109/TCSI.2010.2043987.

[21] Hongying Wang, Filippo Schembari, Marek Miskowicz, and Robert Bogdan Staszewski. An Adaptive-Resolution Quasi-Level-Crossing-Sampling ADC Based on Residue Quantization in 28-nm CMOS. *IEEE Solid-State Circuits Letters*, 1(8):178–181, 2018. ISSN 2573-9603. doi: 10.1109/LSSC.2019.2899723. URL https://ieeexplore.ieee.org/document/8642805/.

[22] Frank M. Yaul and Anantha P. Chandrakasan. A 10 bit SAR ADC with data-dependent energy reduction using LSB-first successive approximation. *IEEE Journal of Solid-State Circuits*, 49(12):2825–2834, 2014. ISSN 00189200. doi: 10.1109/JSSC.2014.2352304.

[23] D. Gangopadhyay, E. G. Allstot, A. M. R. Dixon, K. Natarajan, S. Gupta, and D. J. Allstot. Compressed sensing analog front-end for bio-sensor applications. *IEEE Journal of Solid-State Circuits*, 49(2):426–438, Feb 2014. ISSN 0018-9200. doi: 10.1109/JSSC.2013.2284673.

[24] H. Gao, R. M. Walker, P. Nuyujukian, K. A. A. Makinwa, K. V. Shenoy, B. Murmann, and T. H. Meng. Hermese: A 96-channel full data rate direct neural interface in 0.13μm cmos. *IEEE Journal of Solid-State Circuits*, 47(4):1043–1055, April 2012. ISSN 0018-9200. doi: 10.1109/JSSC.2012.2185338.

[25] C. Mora Lopez, J. Putzeys, B. C. Raducanu, M. Ballini, S. Wang, A. Andrei, V. Rochus, R. Vandebriel, S. Severi, C. Van Hoof, S. Musa, N. Van Helleputte, R. F. Yazicioglu, and S. Mitra. A neural probe with up to 966 electrodes and up to 384 configurable channels in 0.13 μm soi cmos. *IEEE Transactions on Biomedical Circuits and Systems*, 11(3):510–522, June 2017. ISSN 1932-4545. doi: 10.1109/TBCAS.2016.2646901.

[26] C. M. Lopez, H. S. Chun, L. Berti, S. Wang, J. Putzeys, C. Van Den Bulcke, J. Weijers, A. Firrincieli, V. Reumers, D. Braeken, and N. Van Helleputte. A 16384-electrode 1024-channel multimodal cmos mea for high-throughput intracellular action potential measurements and impedance spectroscopy in drug-screening applications. In *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, pages 464–466, Feb 2018. doi: 10.1109/ISSCC.2018.8310385.

[27] and T. W. Rondeau, J. H. Reed, and C. W. Bostian. Analog-to-digital converters. *IEEE Signal Processing Magazine*, 22(6):69–77, Nov 2005. ISSN 1053-5888. doi: 10.1109/MSP.2005.1550190.

[28] Rafał DŁugosz and Tomasz Talaśka. Low power current-mode binary-tree asynchronous min/-max circuit. *Microelectronics Journal*, 41(1):64 – 73, 2010. ISSN 0026-2692. doi: https://doi.org/10.1016/j.mejo.2009.12.009. URL http://www.sciencedirect.com/science/article/pii/S0026269209002304.

[29] Alexander Fish, Vadim Milrud, and Orly Yadid-Pecht. High-speed and high-precision current winner-take-all circuit. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 52:131 – 135, 04 2005. doi: 10.1109/TCSII.2004.842062.

[30] Samaneh Babayan-Mashhadi, Mojtaba Daliri, and Reza Lotfi. Analysis of power in dynamic comparators. *2013 21st Iranian Conference on Electrical Engineering, ICEE 2013*, pages 3–6, 2013. ISSN 2164-7054. doi: 10.1109/IranianCEE.2013.6599853.

[31] B. Razavi. The strongarm latch [a circuit for all seasons]. *IEEE Solid-State Circuits Magazine*, 7(2): 12–17, Spring 2015. ISSN 1943-0582. doi: 10.1109/MSSC.2015.2418155.

[32] Marcel Pelgrom. *Analog-to-Digital Conversion*. Springer International Publishing, 2017. ISBN 9783319449708.

[33] P. J A Harpe, Cui Zhou, Yu Bi, Nick P. Van Der Meijs, Xiaoyan Wang, Kathleen Philips, Guido Dolmans, and Harmke De Groot. A 26 μ W 8 bit 10 MS/s asynchronous SAR ADC for low energy radios. *IEEE Journal of Solid-State Circuits*, 46(7):1585–1595, 2011. ISSN 00189200. doi: 10.1109/JSSC.2011.2143870.