Integrated Circuits and Systems for Sparse Signal Acquisition based on Asynchronous Sampling and Compressed Sensing

A dissertation

submitted by
Michael Trakimas

In partial fulfillment of the requirements
for the degree of

Doctor of Philosophy
in
Electrical Engineering

Tufts University

May, 2011

Adviser: Professor Sameer Sonkusale

Abstract

This dissertation builds on the recent theoretical and experimental work on asynchronous sampling and compressed sensing. Our goal is to exploit the advances in the theory to design practical data acquisition systems capable of directly acquiring sparse signals at sub-Nyquist rates. We focus specifically on increasing the power efficiency and decreasing the complexity of the signal acquisition process compared to existing conventional Nyquist rate solutions for biomedical sensor and wideband spectrum sensing applications.

The first half of the dissertation presents the design and implementation of an adaptive resolution asynchronous ADC which achieves data compression for sparse and burst like signals by the inherent signal dependent sampling rate of the asynchronous architecture. The main contribution of this work is the implementation of an adaptive resolution (AR) algorithm which varies the quantizer resolution of the ADC with the slope of the input signal, in order to overcome the tradeoff between dynamic range and input bandwidth typically seen in asynchronous ADCs. This allows the maximum possible input bandwidth to be achieved regardless of the dynamic range requirement. By reducing the quantizer resolution during periods of high input slope, further data compression is also achieved. A prototype ADC was fabricated in a 0.18μm CMOS technology and optimized for subthreshold operation in order to increase the power efficiency for

low-frequency biomedical sensor applications. The prototype ADC achieves an equivalent maximum sampling rate of 50kS/s, an SNDR of 43.2dB, and consumes 25µW from a 0.7V supply. The ADC is also shown to provide data compression for accelerometer and ECG applications as a proof of concept demonstration.

The second half of this dissertation presents the design and implementation of a compressed sensing based analog-to-information converter (AIC) for wideband spectrum sensing applications. The core of the design is an ultra low power moderate rate ADC that randomly samples the received signal at sub-Nyquist rates. In order to ensure proper functionality with the random clock signal and to maximize power efficiency, a prototype edge-triggered charge-sharing SAR ADC core was implemented in 90nm CMOS technology. The prototype SAR ADC core achieves a maximum sample rate of 9.5MS/s, an ENOB of 9.3 bits, and consumes 550µW from a 1.2V supply. Measurement results of the compressed sensing AIC demonstrate effective sub-Nyquist random sampling and reconstruction of signals with sparse frequency support suitable for wideband spectrum sensing applications. When accounting for the increased input bandwidth compared to Nyquist, the AIC achieves an effective figure of merit (FOM) of 10.2fJ/conversion-step.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would first like to thank my adviser, Professor Sameer Sonkusale, for all the support he has given me throughout my time at Tufts University. I am especially grateful for the freedom he has allowed me in defining and pursuing research which interests me. It has been a great experience working in his research group, the NanoLab, and seeing how it has grown from its formation a year before I joined.

I would also like to thank the members of my dissertation defense committee, Professor Hwa Chang, Professor Valencia Joyner, and Dr. Timothy Hancock for their interest, time, and feedback given to my dissertation. I would like to thank Professor Chang, who was my undergraduate adviser, for always showing interest in my development as both a student and a person throughout both my undergraduate and graduate years at Tufts University.

I have enjoyed working with all the other members of the NanoLab over the past several years, especially the group of students who all joined around the same time as I did, which includes Sungkil Hwang, Jon Chow, Krenar Komoni, Ashwin Duggal, Vinay Agarwal, Ritika Agarwal, Jian Guo, Kyoungchul Park, Sam MacNaughton, Saroj Rout, and Wangren Xu. I am thankful for the help they have provided me with my research as well as the friendship and common experiences we have shared over the past several years.

This work would not have been possible without the support I received from MIT Lincoln Laboratory over the past several years. I am grateful for not only the financial support they have provided me to pursue my research, but also for providing me with an office at the laboratory so that I could take advantage of their many resources as well as interact with the staff. I would especially like to thank Tim Hancock, who served as my supervisor at the laboratory, for always taking the time to help me with any problem I have had with my research or to just listen to me complain for the last year about wanting to graduate. His help has been invaluable and greatly improved the quality of my work as well as forwarded my development as a professional engineer. I would also like to thank Mark Gouker and Leonard Johnson for bringing me in as a member of Group 86, as well as all the other members of our group for their help and support over the past several years. While there are too many people at the laboratory to mention them all, I would especially like to thank my officemates Jeff Myer, Chris Galbraith, and Mike Hamilton for their support and technical insight as I have been finishing up my dissertation this past year, as well as Milan Raj and Brian Tyrrell for their help with chip fabrication.

Finally, I would like to thank my parents and sisters, Nicole and Danielle, for all the support and love they have given me not only during my graduate years but throughout my life. This work would have not been possible without all the opportunities they have provided me in life. Last, but of course not least, I want to thank my girlfriend Christine. I'm sure when we met a week before I entered graduate school she had no idea what she was getting into. Thank you for all of

the constant support, patience, and love you have given me throughout these past six years. I look forward to our future together, which I promise will include no more practice runs of my defense talk.

Chapter 1

Introduction

1.1 Research Motivation

Throughout human history, we have always strived to acquire as much information as possible from the world around us. Arguably the greatest impact on our ability to acquire and interpret information from the world we live in has been the advent of modern electronics. This has allowed us to use specially designed sensors to acquire real world signals in electronic form, and then use electronic signal processing to extract, store, and transmit the information embedded in them.

Traditionally, most signal processing was done in the analog domain, which limited the processing capabilities available to us. A shift began with the pioneering work done by Whittaker, Nyquist, Kotel'nikov, and Shannon [1]-[4], which demonstrated that bandlimited, continuous time signals, can be exactly recovered from a set of uniformly-spaced samples taken at the Nyquist rate of twice the bandlimit. This allowed signal processing to move from the analog to the digital domain. When combined with the ever increasing processing power provided by Moore's law [5], this has allowed for the creation of more complex

sensing and processing systems which are more robust, flexible, and cheaper than their analog counterparts.

While sensing technology greatly benefited from the move to digital signal processing, it was not until recent advances in wireless communication technology when its full potential would be realized. This has become evident in our everyday life where wireless sensors are now embedded in everything from the phones we carry to the cars we drive. Other examples of the diverse tasks now performed by wireless sensing systems are the monitoring of our health, providing wide area constant surveillance in both civilian and military applications, and acquiring environmental data in environments which are inhospitable to humans, among countless others tasks.

While wireless communication technology has expanded the range of potential applications for sensing technology, the fact that each sensor must now operate from a battery or other energy limited supply has its own set of challenges. This is exacerbated in emerging technologies such as biomedical implants where the battery size is severely constrained and replacing the battery is inconvenient. Therefore, it is desirable to design each sensor with as low complexity and power consumption as possible, in order to increase their operational lifetime as well as decrease their size, weight, and cost. In many applications this is accomplished by offloading as much of the signal processing and analysis as possible to a central processing node or other external device. Each sensor is then tasked with acquiring the signals of interest, storing them, and

eventually transmitting them back to the central processing node, where ample power and processing resources are available.

In order to reduce the transmission and storage requirements of each sensor, digital compression algorithms are often employed to compress the data once it has been acquired. Conventional compression techniques operate on the full Nyquist rate signal by first transforming the signal into a domain where it has a sparse representation and then encoding only a certain number of the largest transform coefficients. For signals which are sparse or compressible, the majority of the coefficients can be discarded with minimal loss of information. While this "acquire-then-compress" technique improves the transmission power efficiency of the sensor and reduces the memory requirements, it actually increases the complexity and power consumption of the signal acquisition process due to the additional compression algorithms needed.

For emerging technologies such as wireless body area networks (WBANs) [6]-[9] and cognitive radio [10]-[13], a more efficient signal acquisition process is desirable. Recent interest in WBANs has been driven by the desire for low-cost continuous ambulatory monitoring of patients with chronic diseases, those recovering from surgery, etc. For example, continuous monitoring of a patient's blood pressure in their natural environment allows for a more accurate diagnosis compared to relying on a one time measurement at the doctor's office. Long term electrocardiogram (ECG) monitoring could also be used to detect an impending heart attack for at risk patients. Figure 1.1 illustrates a simple representation of an example WBAN, where wireless sensors are used to sense and acquire various

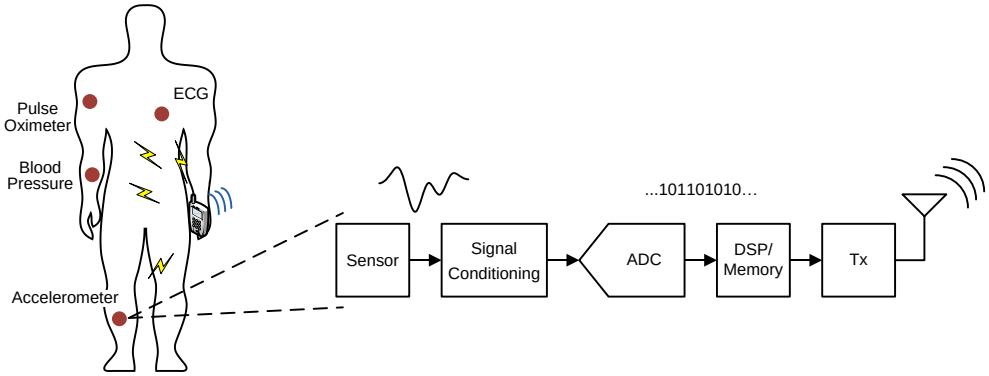


Figure 1.1: Wireless body area network with typical wireless sensor block diagram.

vital signs and transmit them to a central processing node, which could be a smart phone or PDA. The acquired information could then be displayed to the patient or transmitted to a physician to allow timely and more accurate diagnosis and response to emergency situations. Typical sensors may include ECG and electroencephalography (EEG) electrodes, accelerometers, blood pressure and glucose monitors, pulse oximeters, etc. A block diagram of a typical sensor node is shown in Fig. 1.1, and will be described in more detail in the next chapter. Since each sensor will be worn on the patient's body or implanted inside of them, it is critical that they be as unobtrusive and low maintenance as possible. This is especially true for implantable devices, for which replacement of the power source is particularly concerning. If the power consumption of each sensor node can be reduced to 10s of μWs or less, then energy scavenging could be used to allow autonomous operation for the required lifetime of the implant [14]-[17].

In light of the extreme energy and size constraints placed on each sensor in WBAN applications, the increased complexity and power consumption of the "acquire-then-compress" signal acquisition process described previously is

undesirable. A more efficient signal acquisition process could be designed if the signal could be directly acquired in compressed form without the need for additional compression algorithms. This has been the focus of recent work on asynchronous sampling [18]-[26] and compressed sensing [27]-[35]. Asynchronous sampling is an event driven signal acquisition approach which exploits the sparsity in time of certain types of signals such as those often seen in ECG and EEG monitoring applications, hearing aids, implantable biomedical diagnostic devices, and certain environmental sensors. By only sampling the input signal when it changes by a preset amount, burst-like or sparsely active signals can be directly acquired at sub-Nyquist rates and later reconstructed with close to full fidelity. Similarly, compressed sensing exploits a priori knowledge that a signal is sparse or compressible in some basis to design a measurement process which directly acquires the signal at sub-Nyquist rates. Both of these techniques effectively trade reconstruction efficiency for a reduction in sample rate. This is advantageous for energy constrained wireless sensor applications where signal analysis is performed remotely at a central processing node or other device with ample power and processing resources.

Compressed sensing also has the potential to improve the signal acquisition process in the emerging field of cognitive radio. First presented by Mitola in [13], cognitive radio is a paradigm for wireless communication in which each wireless node adapts its transmission and reception frequency depending on the spectrum occupancy in its area. This allows for a more efficient usage of increasingly scarce spectrum real estate by allowing unlicensed secondary users to access a

band licensed to primary users when it is unoccupied. Wideband spectrum sensing [36]-[38] plays an important role in cognitive radio since the secondary user must continuously monitor the full spectrum to avoid transmitting in a band occupied by a primary user. This is also an essential function in signal intelligence applications where the RF spectrum in different areas is monitored to determine the frequency of emitters, among other tasks.

Conventional Nyquist rate solutions for wideband spectrum sensing often employ either a bank of tunable narrowband band-pass filters which scan across the frequency band of interest, or a high speed ADC to directly acquire the full spectrum [39]. The increased complexity of the filter bank approach and very high sampling rates required by the direct sampling approach make it difficult to design low cost and power efficient wideband spectrum sensing systems. This is especially troubling for autonomous wireless sensor applications where replacing the battery is difficult or may not be possible at all. Since the RF spectrum to be monitored is often sparse in the frequency domain, compressed sensing can instead be used to directly acquire the entire spectrum in compressed form at sub-Nyquist rates. This would reduce the complexity and power consumption of the signal acquisition process and relax the requirements of the ADC, leading to increased operational lifetime and decreased cost for each sensor.

1.2 Summary

This dissertation builds on the recent theoretical and experimental work on asynchronous sampling and compressed sensing. Our goal is to exploit the

advances in the theory to design practical data acquisition systems capable of directly acquiring sparse signals at sub-Nyquist rates. We focus specifically on increasing the power efficiency and decreasing the complexity of the signal acquisition process compared to existing conventional Nyquist rate solutions for biomedical sensor and wideband spectrum sensing applications. We assume that in our applications of interest, the signal will be reconstructed and analyzed remotely where ample power and processing resources are available.

In this spirit, the first half of this dissertation presents the design and implementation of an asynchronous analog-to-digital converter (ADC) targeted at biomedical signal acquisition applications such as ECG and EEG monitoring. The ADC was fabricated in a 0.18μm CMOS process and optimized for subthreshold operation in order to increase the power efficiency for low-frequency biomedical sensor applications. The main contribution of this work was the implementation of an adaptive resolution algorithm which varied the quantizer resolution of the ADC with the slope of the input signal, in order to overcome the tradeoff between dynamic range and input bandwidth typically seen in asynchronous ADCs. This allowed the maximum possible input bandwidth to be achieved regardless of the dynamic range requirement. By reducing the quantizer resolution during periods of high input slope, further data compression was also achieved.

The second half of this dissertation presents the design and implementation of a compressed sensing based analog-to-information converter (AIC) for wideband spectrum sensing applications. The core of the design is an ultra low power moderate rate ADC that randomly samples the received signal at sub-Nyquist

rates. In order to ensure proper functionality with the random clock signal and to maximize power efficiency, a prototype edge-triggered charge-sharing SAR ADC was implemented in 90nm CMOS technology. Measurement results for wideband spectrum sensing demonstrate that our proposed compressed sensing AIC improves power efficiency and reduces complexity compared to the conventional direct sampling architecture, while still maintaining high performance for signals with sparse frequency support. We also analyze and propose solutions to issues which have not been fully addressed in previous works but arise in practice, such as mismatch between the chosen basis functions and the actual received signal. The main contribution of this work was the implementation of a physical compressed sensing acquisition system which moved beyond the proof-of-concept stage and improved power efficiency and reduced system complexity compared to existing solutions for wireless sparse spectrum sensing applications.

This chapter has presented a brief overview of the evolution of electronic data acquisition systems and why it is important in modern wireless sensor applications such as WBANs and cognitive radio to improve the power efficiency and reduce the complexity of the signal acquisition process. We have also introduced asynchronous sampling and compressed sensing as two sub-Nyquist signal acquisition methods capable of achieving these goals. In the following chapter we give a more detailed overview of traditional Nyquist rate digital signal acquisition and give some examples of traditional data compression techniques used to reduce storage and transmission requirements. This serves to provide a

good foundation to compare our proposed asynchronous sampling and compressed sensing based signal acquisition systems to.

In Chapter 3, we introduce the theory behind asynchronous sampling and discuss how it can be used to improve the power efficiency and reduce the complexity of the signal acquisition process for signals with sparse activity in time. In Chapter 4, we present the design and implementation of our proposed asynchronous ADC as well as our proposed adaptive resolution algorithm which overcomes the tradeoff between input bandwidth and dynamic range typically seen in asynchronous ADC designs. Measurement results comparing the performance of our proposed adaptive resolution ADC to asynchronous ADCs with fixed quantizer resolution are then presented followed by measured results demonstrating direct data compression for accelerometer and ECG applications.

In Chapter 5, we give a brief overview of the theory behind compressed sensing and discuss how it can be used to improve the power efficiency and reduce the complexity of the signal acquisition process for signals which are sparse in some basis. In Chapter 6 we present the design and implementation of our proposed compressed sensing based AIC. The core of the design is an ultra low power edge-triggered charge-sharing SAR ADC that randomly samples the input signal at sub-Nyquist rates. Measurement results of the AIC demonstrate sub-Nyquist sampling and reconstruction of signals with sparse frequency support suitable for wideband spectrum sensing applications. The power efficiency and complexity of the proposed compressed sensing AIC based acquisition system is then compared with the conventional Nyquist rate sampling approach for

wideband spectrum sensing applications. We finally conclude the dissertation in Chapter 7, and overview opportunities for future research.

Chapter 2

Nyquist Rate Data Acquisition

The previous chapter served as motivation for why we are interested in increasing the power efficiency and decreasing the complexity of the signal acquisition process for wireless sensor applications. In this chapter we overview traditional Nyquist rate digital signal acquisition and give some examples of traditional data compression techniques used to reduce the memory and transmission requirements of wireless sensors. This serves to provide a good foundation to compare our proposed asynchronous sampling and compressed sensing based signal acquisition systems to in later chapters.

As shown in Fig. 2.1, a typical digital signal acquisition system consists of three main parts [40]: sensors, which convert physical parameters or other real world signals into electrical signals; signal conditioning circuitry, which converts the output of the sensor into a form suitable for conversion to digital values and bandlimits the signal; and an analog-to-digital converter (ADC), which converts the conditioned sensor output to digital values. Once the data is in digital form, digital signal processing can be used to extract information from the acquired signals as well as compress, store, and transmit the acquired data.

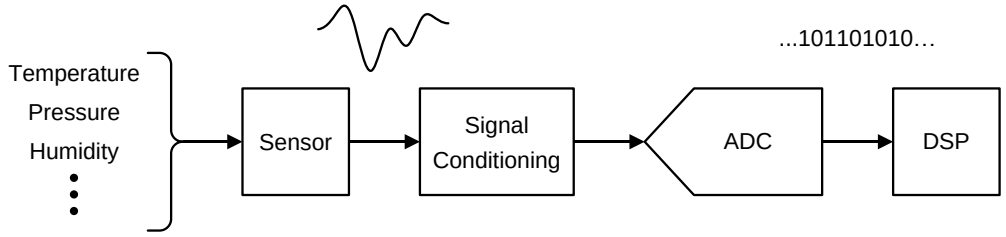


Figure 2.1: Digital signal acquisition system.

As the link between the analog and digital domains, the analog-to-digital conversion is the most critical part of the signal acquisition process. In the next section we overview the basic Nyquist rate analog-to-digital conversion theory. We follow this with an overview of the different ADC architectures used for different applications. We focus specifically on the successive approximation register (SAR) architecture [41], which is often employed in wireless sensor applications due to its good tradeoff between speed and accuracy, and its power efficient and low complexity design. We conclude the chapter with a look at typical data compression techniques, with an emphasis on wavelet based techniques [42] due to their recent popularity and applicability to a wide range of applications.

2.1 Analog-to-Digital Conversion

An analog-to-digital converter is a device which converts a continuous time analog signal, typically a voltage or current, into a sequence of digital values which are proportional to the magnitude of the input signal [43]. As shown in Fig. 2.2, a typical ADC consists of a sample-and-hold stage, which samples the input signal at uniform time intervals, followed by a quantizer, which quantizes each

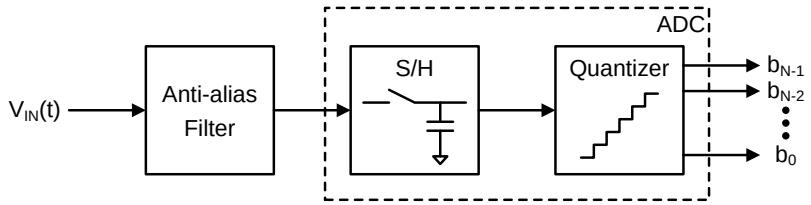


Figure 2.2: Simplified ADC block diagram.

sample to a finite number of digital values. The resulting length N digital output for each sample is a linear combination of the N binary values (b_i) equal to

$$D_{OUT} = \sum_{i=0}^{N-1} b_i 2^i \,, \tag{2.1}$$

and is discrete in both time and amplitude. The relationship between the digital output and the analog input can then be expressed as

$$V_{IN} = D_{OUT} V_{LSB} + V_Q \,, \tag{2.2}$$

where V_{LSB} is the voltage equivalent to a change in the least significant bit (LSB) of the digital output, and V_Q is the error voltage introduced by the amplitude quantization. As shown in Fig. 2.2, the ADC is normally preceded by an anti-aliasing filter to bandlimit the input, as will be discussed later.

2.1.1 Resolution

The bit resolution of the quantizer, which is equal to the length of the digital output word, determines the number of discrete values it can produce to approximate the full range of possible analog sample values. An N bit quantizer divides the full input range of the ADC into 2^N equally spaced levels. The resulting input-output transfer curve for a 3-bit quantizer is shown in Fig. 2.3(a). Compared to the ideal transfer curve of an infinite resolution ADC, which is shown as the dashed line in Fig. 2.3(a), the finite resolution of the quantizer

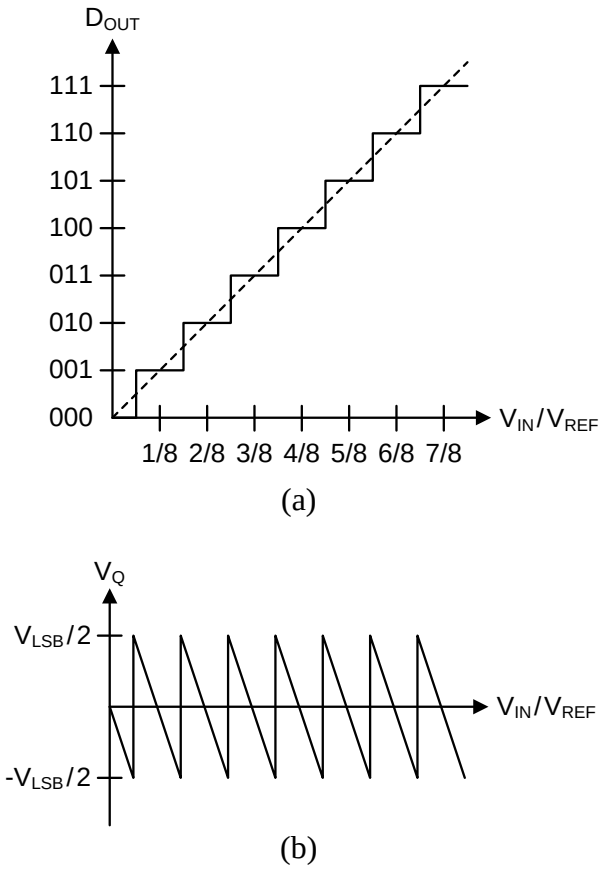


Figure 2.3: (a) Input-output transfer curve for 3-bit quantizer. (b) Quantization error versus normalized input amplitude.

introduces an input dependent error in the output. As shown in Fig. 2.3(b), this quantization error (V_Q) will vary between $\pm V_{LSB}/2$ across the full-scale input range (V_{FS}), where V_{LSB} is the quantizer step size

$$V_{LSB} = \frac{V_{FS}}{2^N}. \tag{2.3}$$

If we assume that the quantization error is a random variable uniformly distributed between $\pm V_{LSB}/2$ with a mean of zero, as shown in Fig. 2.4, then the rms value of this error will be equal to

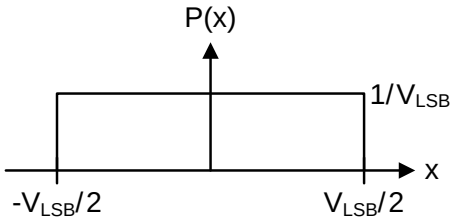


Figure 2.4: Probability density function of quantization noise.

$$V_{Q(RMS)} = \left(\frac{1}{V_{LSB}} \int_{-V_{LSB}/2}^{V_{LSB}/2} x^2 dx \right)^{1/2} = \frac{V_{LSB}}{\sqrt{12}}. \tag{2.4}$$

We can then calculate the resulting maximum achievable signal-to-noise ratio (SNR) for a full-scale sinusoidal input as

$$SNR = 20\log\left(\frac{V_{IN(RMS)}}{V_{Q(RMS)}} \right) = 6.02N + 1.76dB. \tag{2.5}$$

We see that every 1-bit increase in the ADC resolution increases the theoretical maximum SNR by 6dB. In a physical implementation there will be other sources of noise from the individual components of the ADC and the rest of the system which may limit the SNR.

2.1.2 Sampling Rate

Before amplitude quantization is performed, the ADC samples the continuous time input signal at uniformly spaced time intervals. This holds the amplitude steady during the quantization process and accurately sets the sample times. The sampling operation for an analog signal which has been bandlimited to a maximum frequency of f_0 is demonstrated in Fig. 2.5. As shown in Fig. 2.5(b), sampling this signal with an infinite resolution ADC at a frequency of f_s leads to the original spectrum being repeated at intervals equal to f_s. Looking at the resulting spectrum, we see that we should be able to reconstruct the original signal

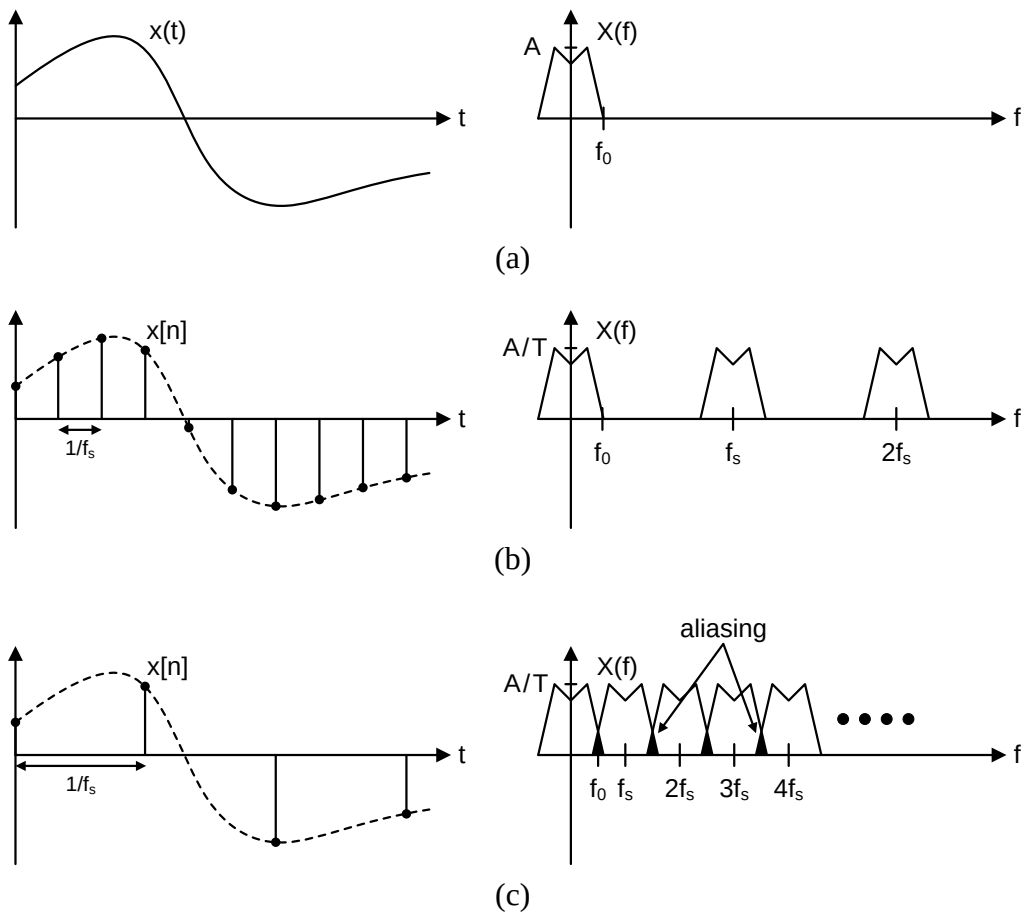


Figure 2.5: Example sampling operation of an analog signal bandlimited to a maximum frequency of f_0. (a) Input signal and spectrum. (b) Sampled signal and spectrum when $fs > 2f_0$. (c) Sampled signal and spectrum when Nyquist criterion is not satisfied.

by low-pass filtering the sampled output. This is normally accomplished using a digital-to-analog converter (DAC) followed by a low-pass filter with cut-off frequency equal to $f_s/2$. In order to ensure that the original signal is exactly reconstructed, the sampling frequency must satisfy the Nyquist sampling criterion [43], which is

$$f_S > 2 f_0 . \tag{2.6}$$

This ensures that the repeated spectra do not overlap each other, which would cause aliasing and unrecoverable distortion in the sampled signal, as is shown in

Fig. 2.5(c). In order to avoid aliasing, the input signal is normally bandlimited by an anti-aliasing filter to a maximum frequency less than $f_s/2$ before being processed by the ADC.

2.2 ADC Architectures

There are many different types of ADC architectures, each best suited for different types of applications. It is common to divide the different architectures into three broad categories based on their relative achievable sample rate and resolution. As shown in Table 2.1, there is typically a tradeoff between speed and resolution which often dictates the architecture used for a particular application. The different architectures can also be broken up into two different classes of converters: Nyquist rate and oversampling [41]. Most conventional ADCs are Nyquist rate converters in that they generate a series of digital output values with a one-to-one correspondence to a single input value. Nyquist-rate converters typically operate at sample rates between 1.5 and 10 times the Nyquist rate in order to make the design of the anti-aliasing and reconstruction filters easier. Oversampling converters are a specialized type of ADC which sample the input at a rate much higher than the Nyquist rate in order to spread the quantization noise

Table 2.1: Overview of different ADC architectures.

Low Speed High Accuracy	Medium Speed Medium Accuracy	High Speed Low/Medium Accuracy
Integrating	SAR	Flash
Oversampling	Algorithmic	Pipelined
		Time-interleaved

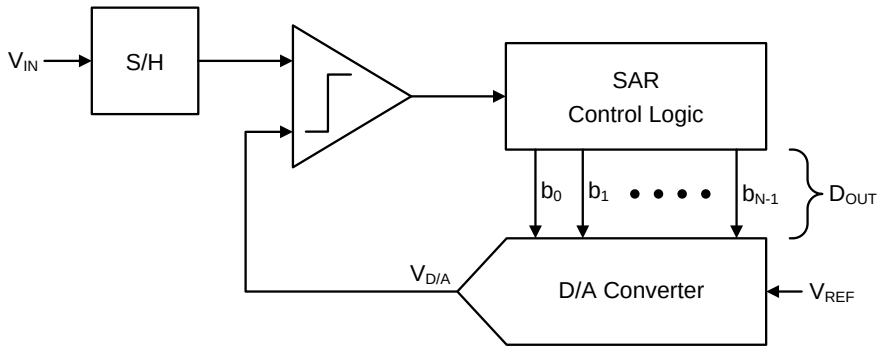


Figure 2.6: Successive approximation register ADC architecture.

over a wider bandwidth. The noise that is not in the signal's bandwidth can then be filtered out to increase the SNR. Often, as in the case of Sigma Delta converters [41], noise shaping is used to push the quantization noise to higher frequencies outside the signal's bandwidth, allowing the SNR to be increased further.

The successive approximation register (SAR) ADC [41] is a type of Nyquist rate converter that has become very popular in wireless sensor applications due to its good tradeoff between speed and accuracy, as well as its power efficient and low complexity architecture. Some recently reported SAR ADCs have achieved sample rates in the 10s of MHz, with resolutions around 10-bits, while only consuming microwatts of power [44],[45]. The basic architecture of a SAR ADC is shown in Fig. 2.6. The ADC consists of a sample-and-hold circuit, a comparator, a DAC, and control logic. The sample-and-hold circuit is used to define the sample time and hold the sampled value steady during the conversion, while the comparator, DAC, and control logic implement the amplitude quantization.

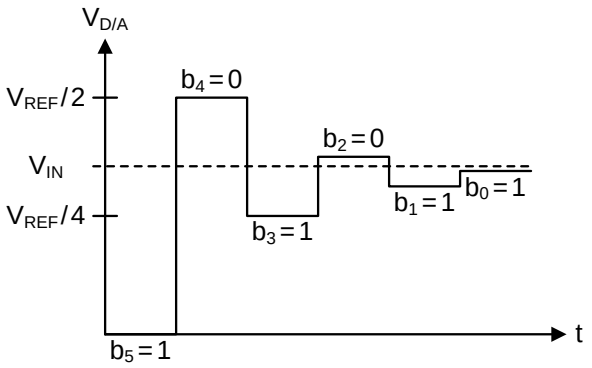


Figure 2.7: Successive approximation conversion.

Once the input has been sampled, the control logic runs a binary search algorithm to find the closest digital value that represents the sampled input. The most straightforward implementation is to use a DAC to successively approximate the input amplitude, as is shown in Fig. 2.7 for a signed input. The resulting successive approximation algorithm is shown in Fig. 2.8. During each period, the algorithm determines each successive output bit of the ADC and divides the possible range of values to represent the input sample by half. At the end of the conversion all *N* bits are resolved and the DAC voltage will be within ½ LSB of the sampled value. The total conversion takes *N* clock cycles to complete, which is what limits the speed of SAR ADCs. The resolution of the ADC is determined by the linearity of the DAC as well as the noise and offset of the comparator and sample-and-hold circuits.

2.3 Data Compression

As previously discussed, in wireless sensor applications, it is often desirable to compress the data once it has been acquired in order to reduce the storage and

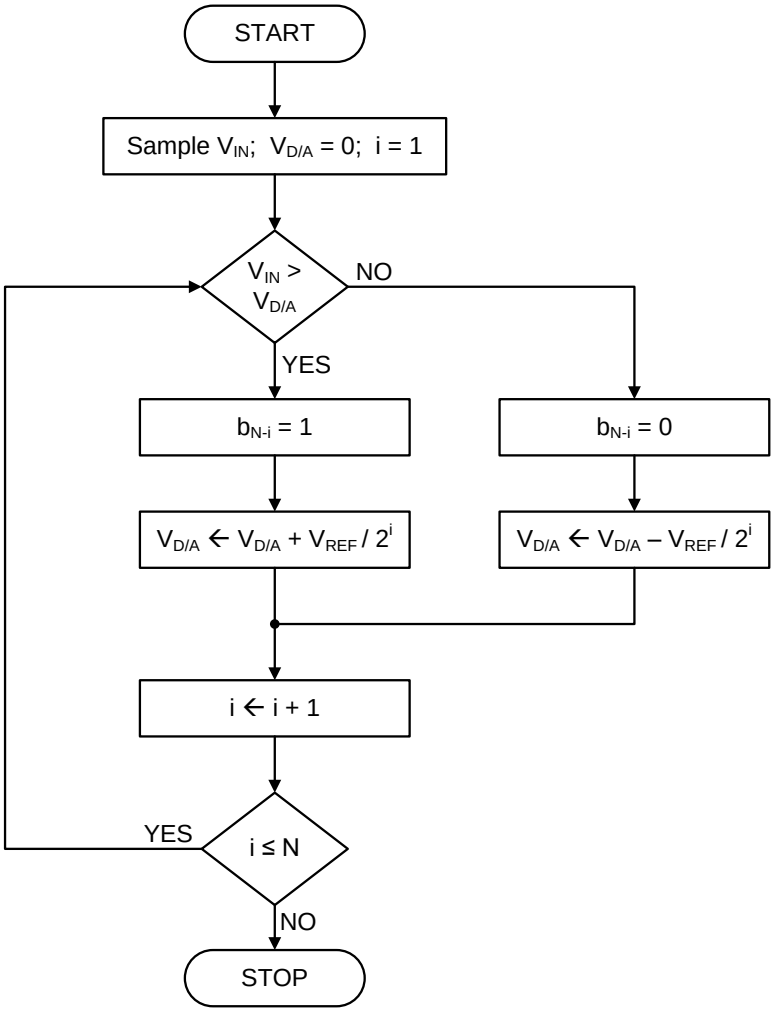


Figure 2.8: Successive approximation algorithm.

transmission requirements of each sensor. Conventional compression techniques [46] operate on the full Nyquist rate signal by first transforming the signal into a domain where it has a sparse representation and then encoding only a certain number of the largest transform coefficients. For signals which are sparse or compressible in some domain, the majority of the coefficients can be discarded with minimal loss of information.

Typically, compression methods are described as being either lossless or lossy. Lossless compression algorithms exploit statistical redundancy in data to

represent it more concisely while still allowing the original signal to be perfectly reconstructed. For example, the well known ZIP file compression method is based on a Lempel-Ziv compression method [47] that utilizes a table based compression model where table entries are substituted for repeated strings of data. Lossy compression methods trade a loss in fidelity for increased compression. This is acceptable for many applications, especially audio and visual where the limitations of the human sensory system can be exploited. For example, the human eye is more sensitive to subtle variations in luminance than it is to variations in color. This allows JPEG [48] image compression to "round off" some of the less important information with limited loss in perceived fidelity.

The JPEG image compression method works by taking the discrete cosine transform (DCT) [49] of an image and then quantizing and encoding the transform coefficients. The DCT is similar to the discrete Fourier transform except that the image is represented as a sum of cosines instead of a sum of complex sinusoids. It is also used in the MJPEG and MPEG video compression methods [50]. The use of smooth cosine waveforms to represent the image makes the JPEG compression method best suited for images of realistic scenes with smooth variations of tone and color. The more modern JPEG-2000 [48] compression method uses a discrete wavelet transform, which will be discussed in more detail later.

Most modern lossy audio compression formats, including MP3, WMA, and AAC, are based on a modified discrete cosine transform [51]. These compression methods use the study of psychoacoustics [52] to determine which information in

an audio stream cannot be perceived by the human auditory system and can therefore be discarded. Linear predictive coding [53] is another popular technique for audio compression, which uses a model of the sound source, such as the human vocal tract, to compress the data. More specialized compression methods can be used for speech applications [54]-[55], which have a limited frequency range and less complex signals than general audio signals. In traditional PCM digital telephony, A-law and μ-law companding algorithms are used to optimize the dynamic range of the system [56]. Mobile phone standards, such as GSM and CDMA, make use of linear predictive coding methods such as Code Excited Linear Prediction [57] to further compress the audio data.

There are many more compression methods other than the ones mentioned above. Additional examples include vector quantization [58], Wyner-Ziv coding [59], and fractal compression [60]. Recently, wavelet based compression [42] has become popular due to its suitability for a wide range of signals which have transient like characteristics. A few examples of these types of signals include most natural images, audio signals, and many biomedical signals such as the output of ECG and EEG monitors. Like all conventional compression techniques, the first step in wavelet based compression is to acquire the full length N signal at or above the Nyquist rate. The acquired signal is then transformed into a particular wavelet domain by applying a discrete wavelet transform. Compared to the discrete Fourier transform, this has the advantage of capturing both frequency and location (time/spatial) information and can be tailored to a particular signal by choosing a particular wavelet. For example, the scaling and wavelet functions of

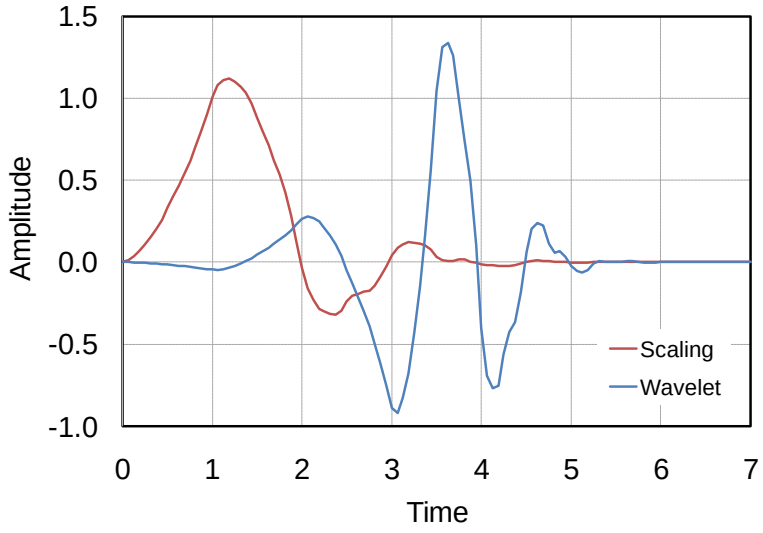


Figure 2.9: Scaling and wavelet functions for Daubechies-4 wavelet.

the Daubechies-4 wavelet are shown in Fig. 2.9 with arbitrary time and amplitude scales.

A compact implementation of the wavelet transform first passes the acquired length N signal $x[n]$ through a low-pass filter with impulse response $g[n]$ equal to the scaling function of the chosen wavelet, resulting in a convolution of the two [61]:

$$y[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[k]g[n-k]. \tag{2.7}$$

The resulting output of the filter is called the approximation coefficients. The detail coefficients are then obtained by simultaneously passing the input through a high-pass filter with impulse response $h[n]$ equal to the wavelet function. Since each set of resulting coefficients only has half the original bandwidth, they can both be subsampled by a factor of two so that the total number of coefficients is still equal to the length of the original signal. In order to further increase the

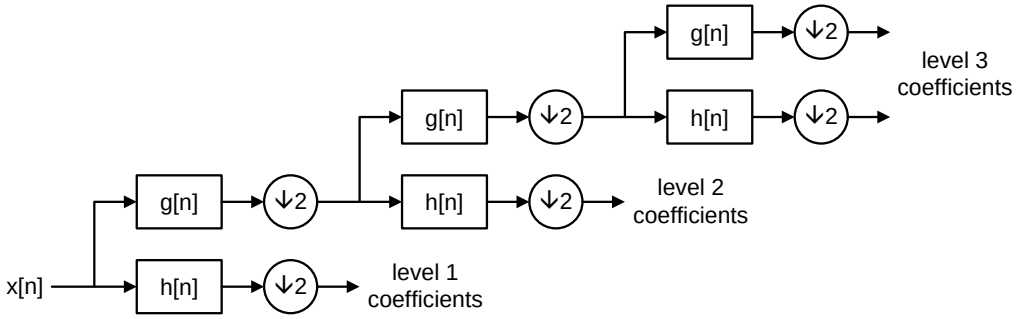
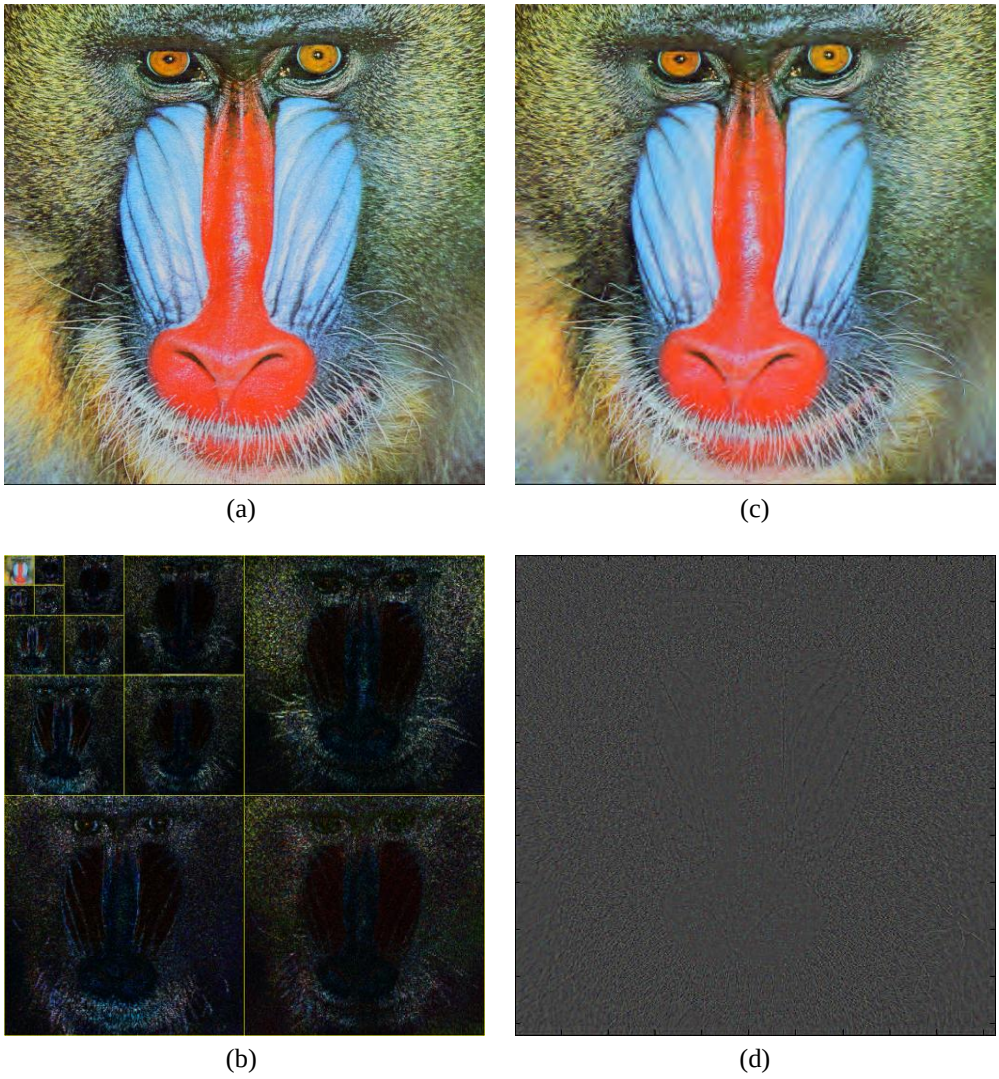


Figure 2.10: Filter bank implementation of wavelet decomposition.

frequency resolution, the pervious operations can be repeated on the resulting approximation coefficients. As shown in Fig. 2.10, this results in a filter bank architecture which decomposes the signal into low and high frequency components at each level. Once the desired number of levels is traversed, data compression is achieved by encoding the largest coefficients. The original signal can then be reconstructed by repeating the filter bank operation in reverse. Since most of the signal energy will be concentrated in a relatively small number of large coefficients for sparse and compressible signals, the reconstructed signal will closely approximate the original signal.

In Figs 2.11-2.13 we demonstrate wavelet compression on several images and signals. In Fig. 2.11(a), an example of a popular test image used in the signal processing field is shown along with its 4-level Daubechies-4 wavelet decomposition in Fig. 2.11(b). It can be seen that most of the transform coefficients are close to zero, as there are large portions of black in the decomposed image which correspond to the smooth parts of the original image. The reconstructed image using only 10% of the coefficients is shown in Fig. 2.11(c) along with the difference between the reconstructed and original images in

Fig. 2.11(d). The reconstructed image contains 99% of the energy in the original signal despite only using 10% of the wavelet coefficients. There is also very little discernable difference between the two images other than some blurring on the lower corners. The same process is repeated for the “Barbara” image in Fig. 2.12 with similar results. In Fig. 2.13 we repeat the same process for a 1-dimensional ECG signal obtained from the MIT-BIH PhysioBank arrhythmia database [62]. The reconstructed signal contains 99.5% of the energy from the original signal using only 10% of the transform coefficients. It is interesting to note that the wavelet compression has effectively denoised the input signal. This is often exploited to design wavelet based filters.



(a) (c)

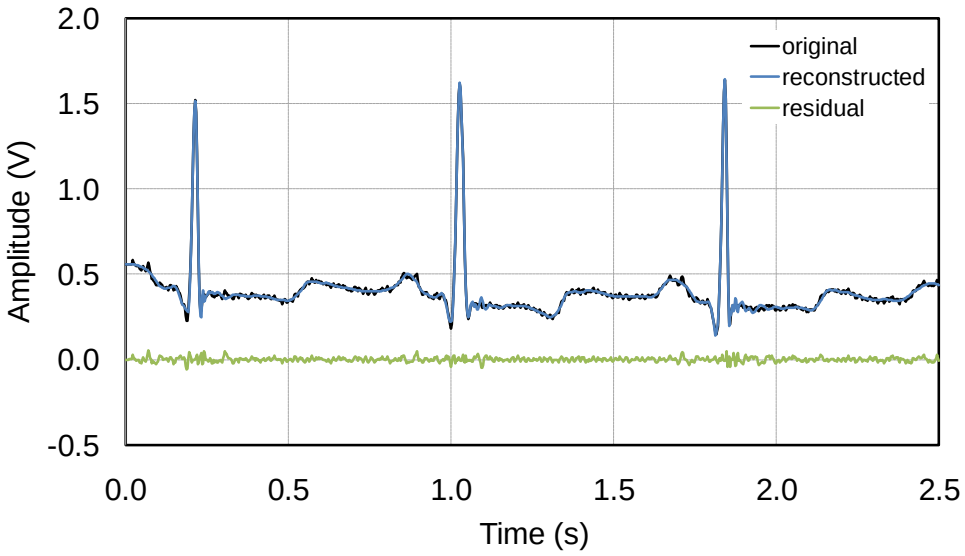
(b) (d)

Figure 2.11: Example of wavelet based image compression. (a) Original image. (b) 4-level Daubechies-4 wavelet decomposition. (c) Reconstructed image from 10% of decomposition coefficients. (d) Residual image.

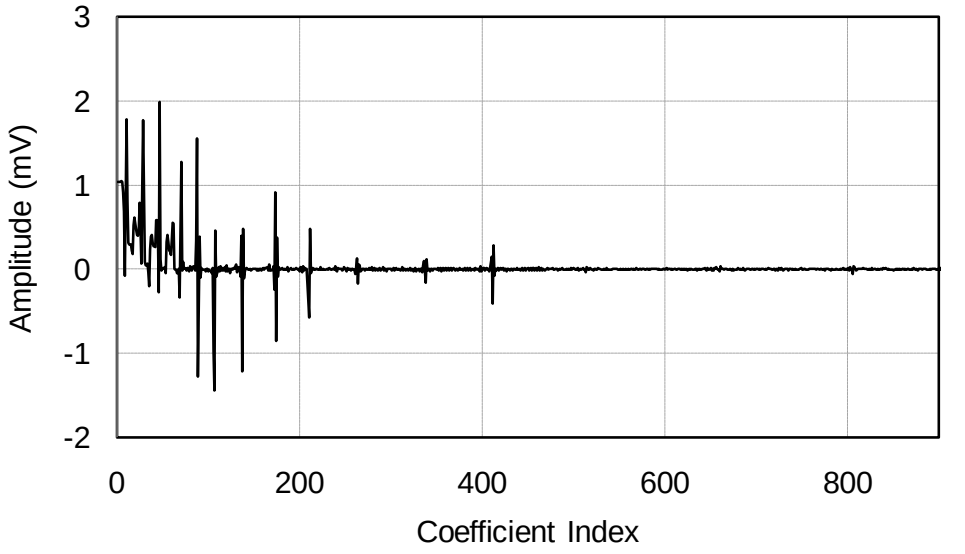
(a) (c)

(b) (d)

Figure 2.12: Example of wavelet based image compression. (a) Original image. (b) 4-level Daubechies-4 wavelet decomposition. (c) Reconstructed image from 10% of decomposition coefficients. (d) Residual image.



(a)



(b)

Figure 2.13: Example of wavelet based compression of an ECG signal. (a) Original signal, reconstructed signal from 10% of wavelet decomposition coefficients, and residual signal. (b) 4-level Daubechies-4 wavelet decomposition.

Chapter 3

Asynchronous Sampling

3.1 Introduction

In the previous chapter, we overviewed how data compression can be an effective way to reduce the energy, processing, storage, and transmission requirements in many sensor devices, especially those where signal analysis is offloaded to a central processing node or other device. Traditional methods of compression use conventional Nyquist rate ADCs followed by computationally intensive digital algorithms for data compression [30]. As previously discussed, this "acquire-then-compress" approach is not optimal in terms of system complexity and power efficiency due to the overhead of acquiring the full Nyquist rate signal, even though most of the acquired information will be discarded. In this section we overview an alternative approach to data compression, which takes advantage of the fact that the outputs of many types of sensors are sparse in the time domain. Data compression can then be achieved by performing signal activity dependent deterministic sampling by way of an asynchronous ADC. By directly acquiring the data in compressed form, the system complexity and power

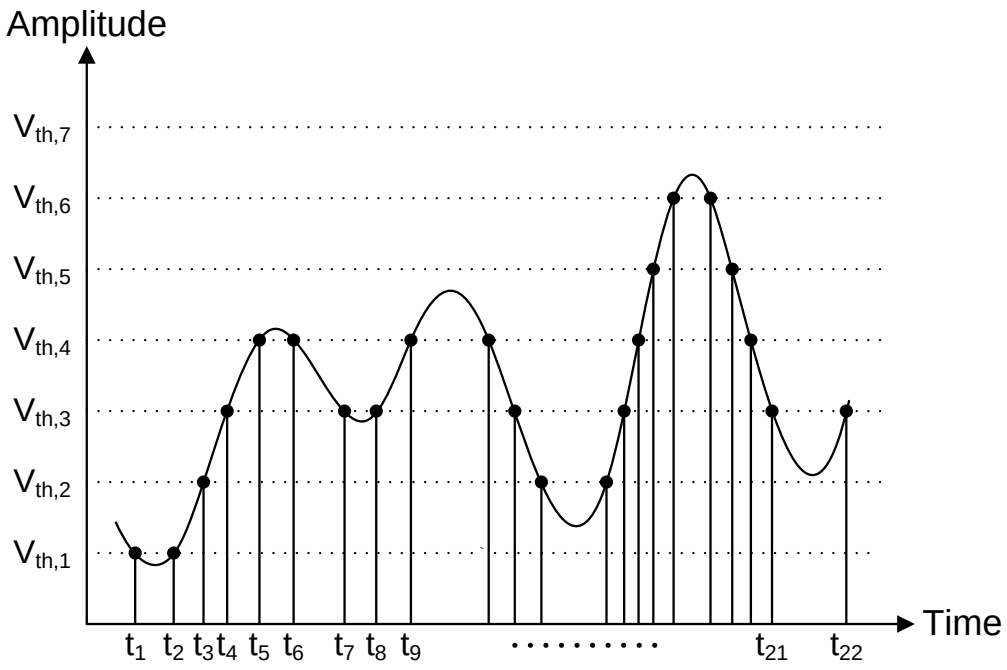


Figure 3.1: Non-uniform sampling of a signal by an asynchronous ADC. The samples are shown as dots.

consumption are reduced and the need for additional compression algorithms following the ADC is avoided.

As shown in Fig. 3.1, an asynchronous ADC only samples the input signal when it crosses predefined threshold levels (quantization levels) [20]. This leads to non-uniformly spaced samples in time and a sample rate that adapts to the rate of change of the input signal. As a result, the average sample rate can be much lower than the sample rate of a conventional Nyquist rate ADC. This creates the potential for large compression of data, especially in applications with burst-like or sparse signals, such as ECG/EEG monitoring, hearing aids, implantable biomedical diagnostic devices, and certain environmental sensors.

Asynchronous analog-to-digital conversion was first proposed in [18] and [19]. These initial systems were based on an asynchronous delta-modulation

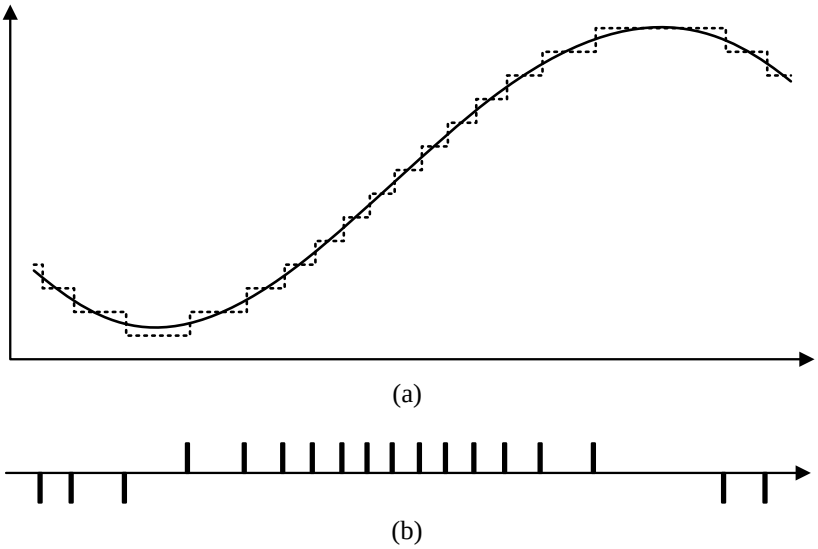


Figure 3.2: (a) Example input and demodulated output waveforms for an asynchronous delta-modulation system. (b) Corresponding delta modulated waveform.

architecture where pulses were generated at the moment when the input signal amplitude changed by more than a predetermined amount. The output pulses were then fed to a demodulator to generate the typical digital staircase approximation waveform, as shown in Fig. 3.2. The goal of these initial asynchronous sampling systems was to reduce the number of pulses generated during pauses in speech and television signals, thus conserving signal power and reducing the average channel bandwidth required to transmit the signals. Most recent asynchronous ADC designs are based on the level crossing sampling scheme initially proposed in [20] and [22]. The level crossing scheme generates samples of the input signal that are nonuniformly spaced in time by recording the time instants at which the signal crosses fixed quantization levels. Further work presented in [24]-[26], [63], and [64] implemented integrated level crossing ADCs which employed comparators to determine when the input signal crossed predetermined

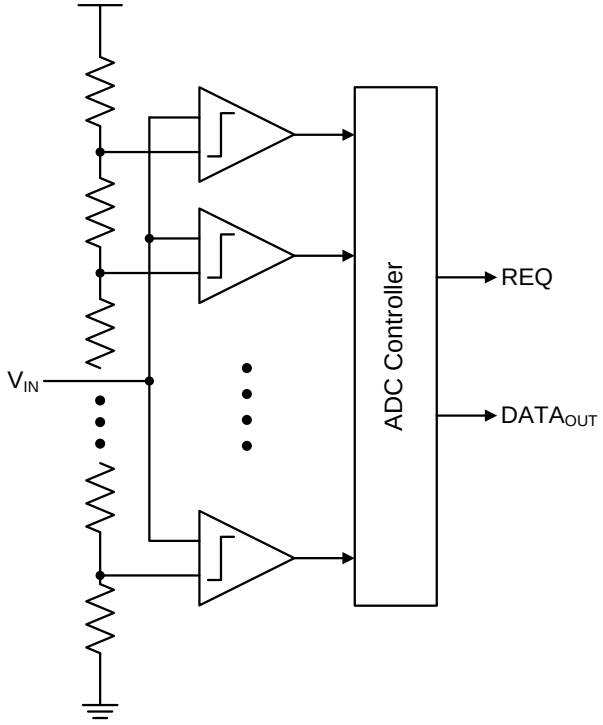


Figure 3.3: Example architecture of a flash based level crossing ADC.

quantization levels. An example flash like architecture which employs a resistor string DAC to create the quantization levels is shown in Fig. 3.3 [24].

An asynchronous sigma-delta sampling scheme which processes the input signal in the time domain instead of the amplitude domain has also been proposed in [65] and [66]. Moving the signal processing to the time domain has the potential to improve the dynamic range compared to conventional synchronous sigma delta ADCs due to the improved time resolution compared to voltage resolution in modern submicron CMOS processes. As shown in Fig. 3.4, an asynchronous sigma-delta ADC pulse-width modulates the input signal using an integrator and Schmitt trigger in a feedback loop. The resulting width of the pulses is proportional to the amplitude of the input signal. A time-to-digital converter (TDC) is then used to quantize and digitally encode the width of each

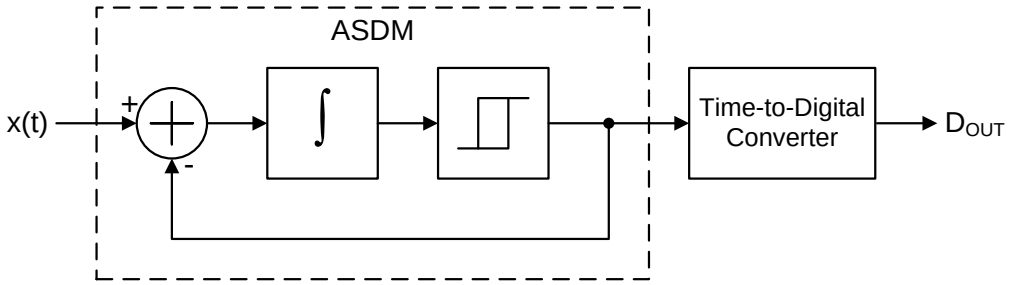


Figure 3.4: Block diagram of an asynchronous sigma-delta ADC architecture.

pulse. The dynamic range of the ADC is directly proportional to the resolution of the time-to-digital converter.

A simple implementation of a TDC for measuring the time difference between two clock edges is shown in Fig. 3.5(a). The first clock edge propagates through the chain of inverters and its position is saved by the D flip-flops on the rising edge of the second clock edge. The number of inverters traversed corresponds to the time difference between the two clock edges. The resolution of the TDC is limited by the calibrated inverter delay. The time resolution can be improved by using the Vernier delay line technique shown in Fig. 3.5(b). This TDC operates by sending each clock edge through separate delay lines. By making the delay of the bottom half buffers slightly less than the top half buffers, the second clock edge will eventually catch up to the first, at which point the position of the top clock edge is stored in the D flip-flops. The timing resolution can be improved since the difference in delay between two buffers can be made much smaller than the total delay through the buffer.

Several methods have been proposed for processing the non-uniformly spaced data output from asynchronous ADCs. To allow for digital transmission and storage of the sampled data, and interfacing with conventional synchronous

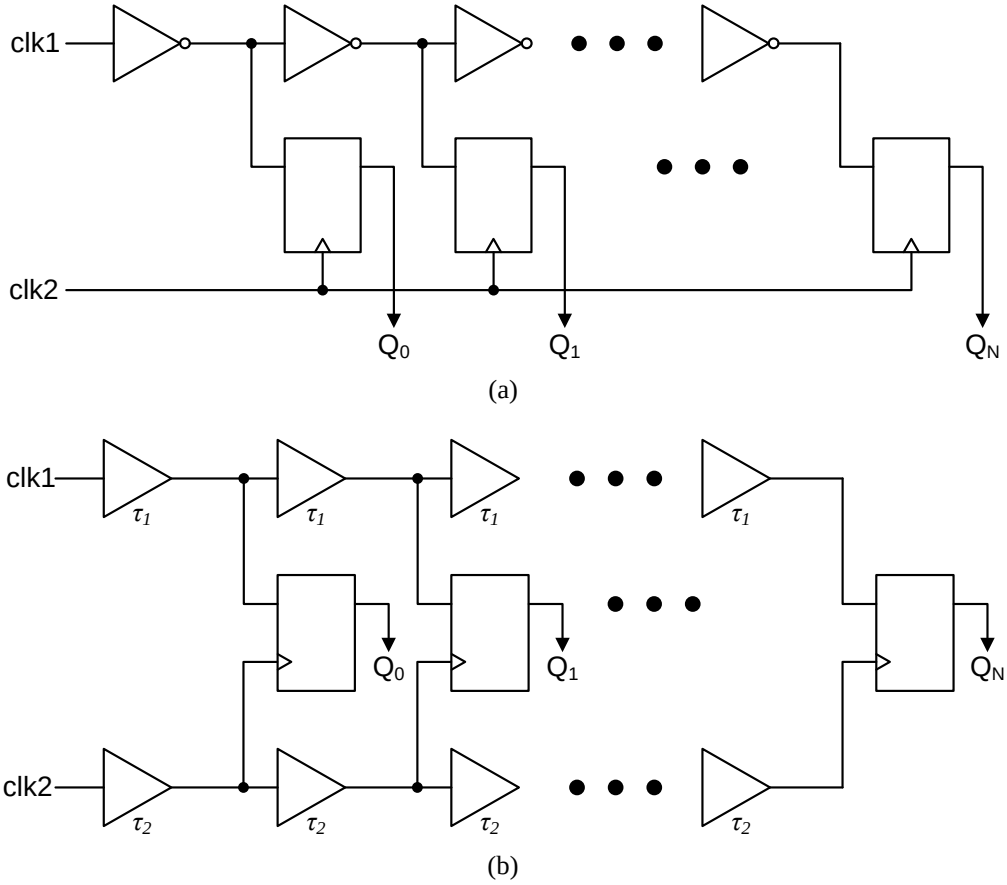


Figure 3.5: (a) Delay line time-to-digital converter. (b) Vernier delay line time-to-digital converter.

systems, the time between samples can be quantized and digitally encoded. The amplitude and time data can then be transmitted and stored using conventional techniques [22], and processed with either asynchronous signal processing techniques [23], or a conventional digital signal processor (DSP), preceded by an interpolator which reconstructs a synchronous signal [21]. For optimal performance, the interpolation can be done after the asynchronous data has been transmitted to a central processing node or other device with ample power and processing resources. To obviate the need for transmission of time information, Tsividis et al. have presented a continuous-time DSP which processes the non-

uniform samples without the need for a clock [25],[26]. This is made possible by implementing the asynchronous ADC as an asynchronous delta modulator, where the time information is inherently encoded in the time between pulses. Apart from energy savings provided by the activity dependent power dissipation, the real-time processing provided by the continuous time DSP has been shown to decrease the response time and voltage deviations in DC-DC converters [67].

In the remainder of this and the following chapter, we focus solely on a novel method for asynchronous data acquisition and its practical implementation. The data thus acquired may be processed by any of the techniques mentioned above. As will be seen in the following chapter, for testing purposes, we use an interpolator to reconstruct a synchronous signal so that the performance of our prototype asynchronous ADC can be measured with standard ADC post processing techniques and compared to that of conventional ADCs.

3.2 Background Theory

The theory behind asynchronous ADCs, also referred to as level-crossing ADCs, or asynchronous delta modulators in the literature, was initially reported in [18] and [19], and further developed in [21], [24], and [63]. An overview of some of the important properties is given in this section and later used to highlight some of the improvements offered by our proposed architecture, which is presented in Chapter 4. The given analysis assumes that the quantizer resolution is fixed and the time between samples is approximated by a timer. This allows for digital

transmission and storage of the sampled data and the ability to interface with conventional DSPs.

3.2.1 Signal-to-Noise Ratio

In conventional sampling, a signal is sampled at set time intervals and the amplitude is approximated by a quantizer. Conversely, in asynchronous sampling, the signal is sampled at known amplitude levels while the time at each sample is approximated by a timer. Therefore, asynchronous sampling can be viewed as quantizing a signal in time instead of amplitude.

The signal-to-noise ratio (SNR) can be calculated by transforming the quantization error in time to an error in amplitude. This can be visualized by considering a signal $y(t)$ crossing an amplitude threshold $V_{th,j}$, as shown in Fig. 3.6. Quantization of the time axis with resolution T_{clk} results in a time error equal to δt in the time-amplitude sample (t_i,y_i). The rms value of the amplitude quantization error due to δt has been calculated in [21], for a sinusoidal input of amplitude A_{in} and frequency f_{in}, as

$$\sqrt{E[\delta_v^2]} = \sqrt{2/3}\,\pi A_{in}\, f_{in}\, T_{clk} = \sqrt{2/3}\,\pi A_{in}\,\frac{1}{R}, \tag{3.1}$$

where $E[\delta_v^2]$ is the variance of the equivalent amplitude quantization noise and R is the timer resolution ratio, which is equal to the ratio of the equivalent timer frequency $(1/T_{clk})$ to the input frequency. This calculation assumes that T_{clk} is sufficiently small, so that the slope of $y(t)$ is approximately constant during one period of the timer, δt is uniformly distributed between 0 and T_{clk}, and the threshold levels are ideal. The SNR can then be calculated as

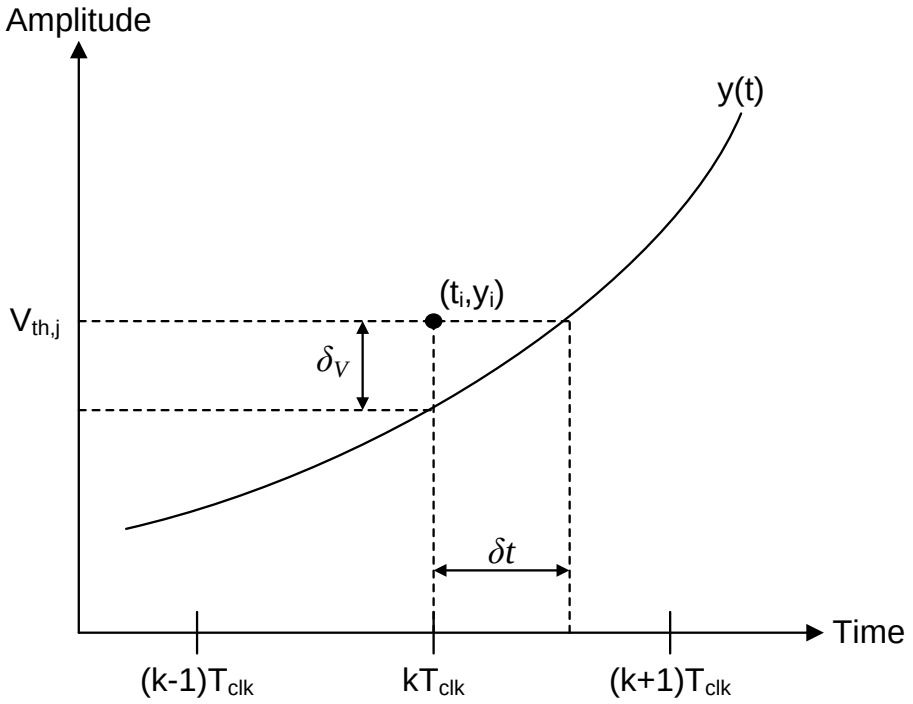


Figure 3.6: Error due to quantization of time in asynchronous sampling.

$$SNR = 10\log\left(\frac{P_{signal}}{P_{noise}}\right) = 10\log\left(\frac{\left(A_{in}/\sqrt{2}\right)^2}{\left(\sqrt{\frac{2}{3}}\pi A_{in}\frac{1}{R}\right)^2}\right) dB, \tag{3.2}$$

and simplified to

$$SNR = 20\log R - 11.2\, dB. \tag{3.3}$$

Therefore, the SNR only depends on the timer resolution ratio, unlike a conventional ADC where the SNR is set by the quantizer bit resolution N and given by

$$SNR = 6.02N + 1.76\, dB. \tag{3.4}$$

The SNR given by (3.3) is the theoretical maximum for a given timer resolution and input frequency. Similar to conventional ADCs, in an actual implementation, the maximum SNR which can be achieved by increasing the

timer resolution will be limited by non-idealities, such as error in the threshold levels and timer jitter.

3.2.2 Maximum Sampling Rate

Similar to a flash ADC, an asynchronous ADC can process a sample in a single step by simply determining which threshold level was crossed. As explained in Chapter 4, our design requires new threshold levels to be set after each sample. Therefore, the minimum time between samples, which is called the loop-delay (σ), is equal to the settling time of the threshold levels plus any additional digital control logic which is required. The maximum sampling rate is then

$$f_{s,\max} = \frac{1}{\sigma}. \tag{3.5}$$

3.2.3 Input Bandwidth

If the input to an asynchronous ADC crosses two threshold levels in less than the loop-delay of the ADC, slope overload distortion will occur due to the resulting time error in the second sample. Therefore, the slope of the input is limited to

$$\left|\frac{dV_{in}}{dt}\right| < \frac{\Delta q}{\sigma} = \frac{V_{FS}}{\sigma 2^{M}}, \tag{3.6}$$

where Δq is the difference between threshold levels, and V_{FS} and M are the quantizer full-scale input swing and resolution in bits, respectively. Assuming a bandlimited input signal with maximum frequency $f_{in,max}$, and maximum amplitude $A_{in,max}$, (3.6) can be used to solve for the maximum distortion free input frequency of

$$f_{in,max} = \frac{\Delta q}{2\pi\sigma A_{in,\max}} = \frac{V_{FS}}{2\pi\sigma A_{in,\max} 2^M}. \tag{3.7}$$

The maximum input frequency is therefore inversely proportional to the input amplitude. For a fixed quantizer resolution, the input bandwidth must be set to the worst case value of (3.7), which occurs when $A_{in,max}$ is at its full-scale value of $V_{FS}/2$ and equals

$$BW = \frac{1}{\pi\sigma 2^M}. \tag{3.8}$$

3.2.4 Dynamic Range

The dynamic range of an asynchronous ADC, defined as the ratio of the maximum to minimum amplitude signal that can be resolved, is calculated the same way as for a conventional ADC. If we assume that in order to process the input it must cross at least two threshold levels, and therefore, have a peak-to-peak voltage of at least $2\Delta q$, the dynamic range will be

$$Dynamic\ Range = 2^{M-1}. \tag{3.9}$$

Chapter 4

Adaptive Resolution Asynchronous ADC

4.1 Introduction

The background theory presented in the previous chapter shows that, unlike conventional ADCs, the effective output resolution of an asynchronous ADC is independent of the quantizer resolution. This has allowed several recently reported asynchronous ADCs to achieve effective output resolutions greater than 8 bits using only 3 to 6-bit quantizers [24],[64],[68],[69]. The low resolution quantizer allows for simple and compact designs, as well as increased data compression and power efficiency due to the reduced number of threshold levels that a given signal will cross. However, the relatively large distance between threshold levels will limit the dynamic range of these ADCs. One drawback to increasing the quantizer resolution, in order to obtain a higher dynamic range, is the resulting increase in inherent sample rate and decrease in data compression and power efficiency for a given signal, due to the increased number of threshold levels that the signal will be required to cross. This will also lead to a “slew-rate limiting” like behavior as the sample rate increases beyond a certain maximum

value, which is determined by the time required to process a sample (the loop-delay). As a result, the maximum input frequency will be severely constrained as the quantizer resolution is increased.

To address this problem, we proposed an adaptive resolution (AR) algorithm in [70] and [71], which varies the quantizer resolution of an asynchronous ADC based on the slope of the input signal. High dynamic range is achieved by processing slowly changing signals with the maximum resolution. As the slope of the input increases, the quantizer resolution is decreased in order to increase the input bandwidth as needed. The decreased resolution also leads to increased data compression.

The proposed algorithm is similar to the approach used in Adaptive Differential PCM and Adaptive Delta Modulation, where the quantizer step size is varied to reduce the bit rate and slope overload [72]. A similar idea is proposed in [73], for a continuous-time DSP system. The focus of this work was to reduce the number of tokens (samples) processed by the continuous-time DSP in order to lower the power dissipation. The quantizer resolution is varied by using digital logic to post process the output of a fixed resolution continuous time ADC and a slope detector, which detects the slope of the input signal every other token. Our proposed algorithm directly varies the threshold levels of the ADC and detects the input slope at every sample.

The ability to provide increased data compression by varying the quantizer resolution is also studied theoretically in [74] for ECG signals. In [75], performance is improved by accumulating statistics of the input signal and using

them to periodically adapt the threshold levels according to predictions made about the input. Other proposed solutions to increase the dynamic range include using logarithmically spaced threshold levels to take advantage of the characteristics of ultrasound signals [68], and adding a dither signal to the input which is later digitally subtracted [69].

In the remainder of this chapter, we present the development of our proposed AR algorithm and its implementation in a prototype asynchronous ADC. The ADC was fabricated in a 0.18μm CMOS process and optimized for subthreshold operation in order to increase the power efficiency for low-frequency sensor applications. Measured results verify that the AR algorithm allowed the ADC's dynamic range to be increased without limiting the input bandwidth or potential for data compression.

In the following section we describe in more detail the tradeoff between input bandwidth and dynamic range in conventional asynchronous ADCs. We then develop our proposed AR algorithm in Section 4.3. In Section 4.4, we present the architecture of our proposed ADC, including the implementation of the AR algorithm. This is followed by the circuit implementation in Section 4.5. Measurement results comparing the performance of the proposed ADC to asynchronous ADCs with fixed quantizer resolution are given in Section 4.6. Results demonstrating data compression for accelerometer and ECG applications are also presented, followed by conclusions.

4.2 Input Bandwidth/Dynamic Range Tradeoff

As discussed in the previous section, in conventional asynchronous ADCs, there exists a tradeoff between the input bandwidth and dynamic range, which is determined by the quantizer resolution. From (3.8) and (3.9), it is seen that every 1 bit increase in the quantizer resolution doubles the dynamic range but cuts the input bandwidth in half. This will lead to an unacceptable limit on the input bandwidth for many applications with even moderate dynamic range. This is demonstrated by the example shown in Fig. 4.1, which shows three cases of an arbitrary input signal sampled by a hypothetical asynchronous ADC. The quantizer resolution of the ADC was set differently for each case, and the loop-delay was set to emphasize the effects of the tradeoff between dynamic range and input bandwidth. The sample rate for each case can be calculated from the tick marks on the time axis, which indicate the sample times.

The sampled output in Fig. 4.1(a) was obtained with the quantizer resolution set just low enough to meet the input bandwidth requirement of the given input and to avoid slope overload distortion. The sampled output accurately tracks the input during the high amplitude pulse, but the limited dynamic range of the ADC prevents it from resolving the two small amplitude peaks at the beginning and end. For the second case, shown in Fig. 4.1(b), the quantizer resolution was increased in order to meet the dynamic range requirement of the input. The sampled output is seen to initially track the input, but begins to fall behind as the input slope increases, leading to slope overload distortion. This is due to the time between level crossings becoming less than the loop-delay of the ADC. The

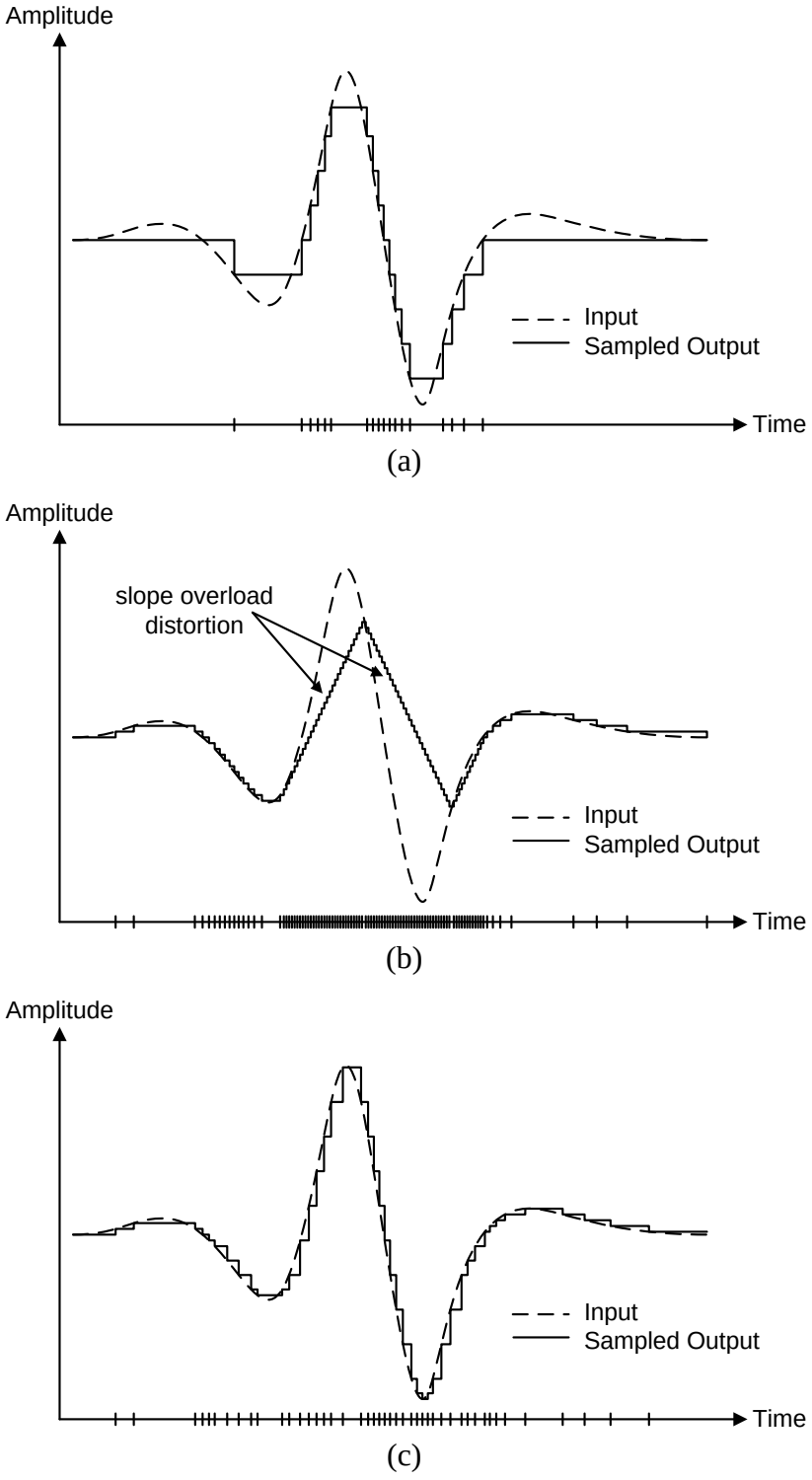


Figure 4.1: Example waveforms of a signal asynchronously sampled with the quantizer resolution: (a) set to the maximum value that meets the input bandwidth requirement of the signal; (b) set to the minimum value that meets the dynamic range requirement of the signal; (c) varied with the input slope, allowing both the input bandwidth and dynamic range requirements of the signal to be met.

increased quantizer resolution is also seen to lead to an excessive sample rate, which prevents any data compression from being achieved.

Figs. 4.1(a) and 4.1(b) illustrate how, in many applications, the tradeoff between dynamic range and input bandwidth will place an unacceptable limit on one of these parameters. Our proposed ADC overcomes this tradeoff by taking advantage of the fact that the effective output resolution of an asynchronous ADC is independent of the quantizer resolution, as shown by (3.3). The quantizer resolution is therefore free to vary with the characteristics of the input signal, as long as the input amplitude remains within its dynamic range. This key observation is critical to the success of our proposed AR asynchronous ADC.

To demonstrate how the tradeoff between dynamic range and input bandwidth can be overcome, the sampled output is obtained again in Fig. 4.1(c), but with the quantizer resolution now dependent on the slope of the input signal. Initially, the quantizer resolution is set to its maximum value, allowing the ADC to resolve the initial small amplitude peak. As the input slope increases, the output step size is increased, corresponding to a decrease in quantizer resolution. This prevents slope overload distortion and allows the ADC to accurately track the input during regions of high input slope. As the input slope decreases again, the quantizer resolution is increased, allowing the last peak to be resolved. Therefore, both the dynamic range and input bandwidth requirements of the given input have been met by allowing the quantizer resolution to vary with the input slope. Since the input signal crosses fewer threshold levels when the quantizer resolution is decreased, the resulting sample rate is also much lower than in Fig. 4.1(b), when

the quantizer resolution was fixed to the minimum value needed to meet the dynamic range requirement.

4.3 Adaptive Resolution Algorithm

The example in the previous section illustrates how the tradeoff between dynamic range and input bandwidth can be overcome by allowing the quantizer resolution to vary with the slope of the input signal. In this section, we develop an AR algorithm for controlling the quantizer resolution, in order to overcome this tradeoff with minimal increase to the complexity and power dissipation of the ADC.

In order to meet the dynamic range requirement of a given application, the maximum quantizer resolution is set by (3.9). This creates a minimum threshold level step size of Δq, and sets the minimum input amplitude which can be resolved to $2\Delta q$, assuming the input must cross at least two threshold levels. According to (3.6), when the quantizer resolution is set to its maximum value, the maximum input slope which can be processed without slope overload distortion is $\Delta q/\sigma$. For input slopes between $\Delta q/\sigma$ and $2\Delta q/\sigma$, slope overload distortion can be avoided by decreasing the quantizer resolution by 1 bit. The quantizer resolution can be decreased by another bit for input slopes between $2\Delta q/\sigma$ and $4\Delta q/\sigma$, and so on until the resolution is decreased to its minimum value of 2 bits, which is required to create two threshold levels for the input to cross. According to (3.8), by allowing the quantizer resolution to decrease to 2 bits, the maximum bandwidth of the ADC will be

$$BW_{MAX} = \frac{1}{4\pi\sigma}. \tag{4.1}$$

The above arguments show that by varying the quantizer resolution with the slope of the input signal, the maximum possible bandwidth of BW_{MAX} can be achieved regardless of the dynamic range requirement. This leads to an algorithm which compares the input slope to thresholds of $\Delta q/\sigma$, $2\Delta q/\sigma$, $4\Delta q/\sigma$, and so on. The quantizer resolution is then set to the maximum value calculated from (3.9) if the slope is below $\Delta q/\sigma$, reduced by 1 bit if the slope is between $\Delta q/\sigma$ and $2\Delta q/\sigma$, and so on until the minimum resolution is reached. The actual implementation of the algorithm in the prototype ADC will be described in the next section.

4.4 ADC Architecture

The fully-differential architecture of the proposed asynchronous ADC is shown in Fig. 4.2(a). The ADC consists of: two digital-to-analog converters (DACs), which create threshold levels bounding the input; two comparators, which detect when the input crosses one of the threshold levels; a resolution controller, which implements the proposed AR algorithm and sets the quantizer resolution; control logic; and a timer, which was used to quantize the time between samples and calculate the slope of the input signal. The input signal is subtracted from the threshold levels within the DACs. This allows the threshold level crossings to be detected by the comparators connected to the output of the DACs.

4.4.1 ADC Operation

The operation of the proposed ADC can be described by referring to Fig.

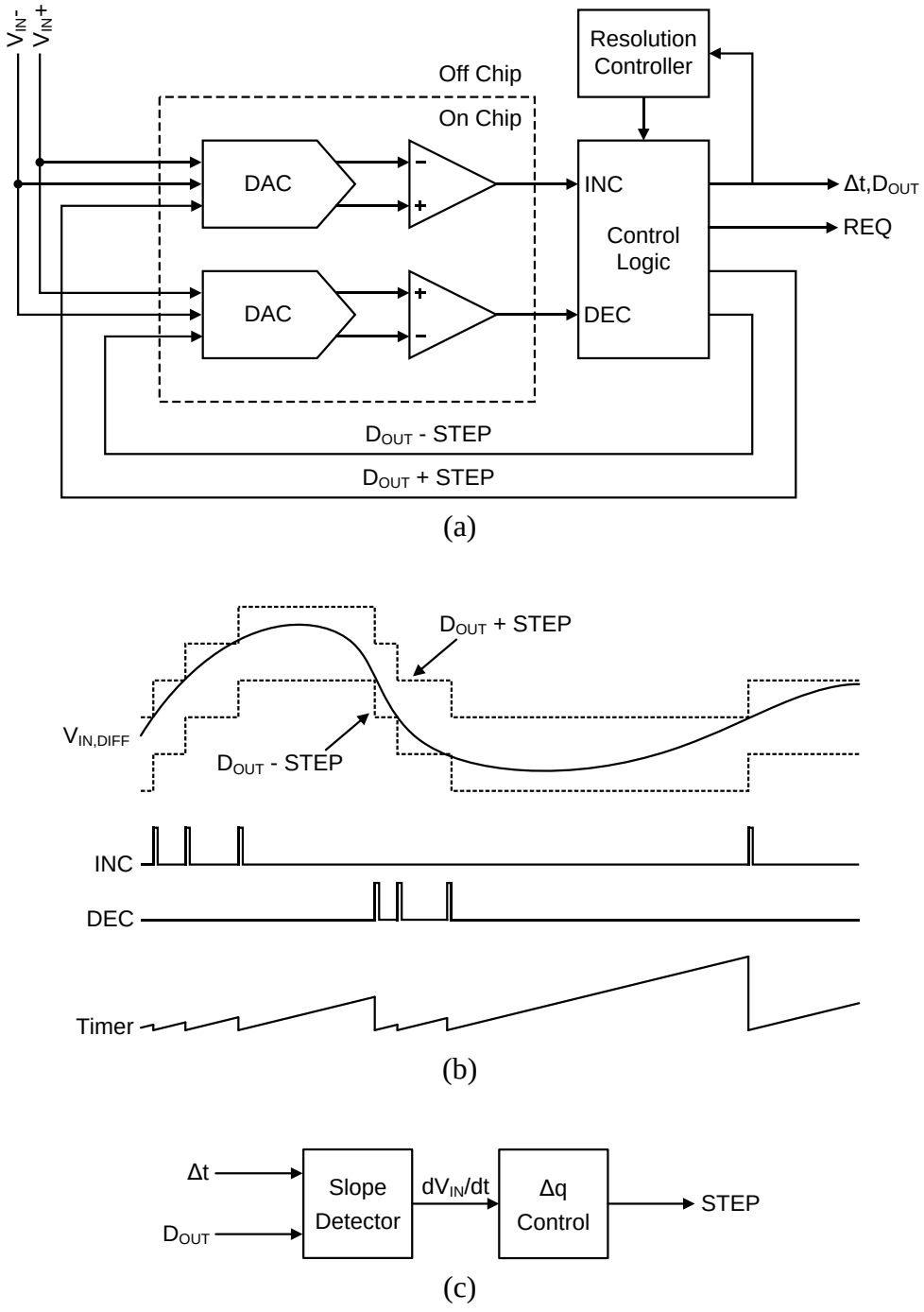


Figure 4.2: (a) Architecture of the proposed AR asynchronous ADC. (b) Example waveforms demonstrating operation of proposed ADC. (c) Simplified block diagram of the resolution controller.

4.2(a) and the example waveforms in Fig. 4.2(b). D_{OUT} is the digital output of the ADC and equals the value of the last threshold level crossed by the input signal.

STEP is the threshold level step size, which is set by the resolution controller and corresponds to the quantizer resolution.

Assume, as is true in the example shown, that the two DACs have initially been set to create threshold levels (the dashed lines in the top waveform) which bound the differential input signal ($V_{IN,DIFF}$) from above and below. When the input crosses one of these threshold levels, the output of the corresponding comparator (*INC* or *DEC*) will toggle high. This triggers the control logic to output the value of the threshold level crossed (D_{out}), the value of the timer (Δt), which represents the time elapsed since the previous sample, and a 1-bit request signal (*REQ*), for synchronization with subsequent stages. The timer is then reset and the resolution controller updates the quantizer resolution by setting the value of *STEP*. The actual operation of the resolution controller will be described in the next section. The control logic then updates the threshold levels so that they bound the last threshold which was crossed by setting the value of the upper DAC to $D_{OUT} + STEP$ and the lower DAC to $D_{OUT} - STEP$. As long as the input slope meets the requirement given by (3.6), these threshold levels will bound the input from above and below as shown in the example waveform. The ADC then waits for the input to cross one of the new threshold levels before repeating the previously described steps.

The resulting output of the ADC consists of non-uniform time-amplitude samples ($\Delta t, D_{OUT}$) which track the input signal and can be processed using the methods described in Chapter 3. As shown in the example waveforms, the sample rate will depend on the rate of change of the input signal. The minimum sample

time (loop-delay) of the ADC will be equal to the time from when the input crosses a threshold level to when the updated threshold levels settle to the required accuracy. If the input crosses two threshold levels in less than this time, slope overload distortion will occur.

4.4.2 Resolution Controller

The addition of the resolution controller differentiates the proposed architecture from a conventional asynchronous ADC. In a conventional architecture, the quantizer resolution would be fixed. As previously mentioned, this leads to a tradeoff between dynamic range and input bandwidth. The proposed architecture overcomes this tradeoff by having the resolution controller vary the quantizer resolution between 8 and 4 bits, depending on the slope of the input signal. According to (3.8), setting the minimum resolution to 4 bits instead of 2 bits will limit the input bandwidth to one quarter of the maximum possible value given by (4.1).

In the actual implementation, the resolution controller sets the quantizer resolution by setting the value of *STEP*, which sets the distance between threshold levels. To set the quantizer resolution to 8 bits, the value of *STEP* is set to 1 LSB, where LSB is the least significant bit of the DACs when their quantizer resolution is 8 bits. In order to decrease the quantizer resolution by one bit, the value of *STEP* is doubled. In the rest of this section, only the quantizer resolution is discussed in order to simplify the description of the resolution controller and AR algorithm. The corresponding *STEP* values can be determined based on the previous discussion.

A simplified block diagram of the resolution controller is shown in Fig. 4.2(c). After each new sample, the slope of the input signal is determined from the output samples and used to set the quantizer resolution according to the proposed AR algorithm. As described in Section 4.3, the AR algorithm compares the slope of the input signal to several slope thresholds derived from (3.6), and then sets the quantizer resolution to the maximum value which avoids slope overload distortion. This algorithm can be simplified since, at each sample, the change in amplitude since the last sample (ΔV_{in}) is equal to the current threshold level step size (Δq). Therefore, the condition for avoiding slope overload distortion can be reduced to the time between samples being greater than the loop-delay, which was measured to be 20μs for the prototype chip. The algorithm can be further simplified by limiting the quantizer resolution to being incremented or decremented by at most 1 bit after each sample. A lower Δt threshold can then be set to determine when to decrement the quantizer resolution, and an upper Δt threshold set to determine when to increment the quantizer resolution. In this design these thresholds were set to 40μs and 80μs, respectively. The choice of these values will be explained later.

A simple flowchart of the resulting AR algorithm is shown in Fig. 4.3. A few details have been left out in order to make the algorithm easier to understand. These details do not change the basic operation of the algorithm and will be discussed at the end of this section. Initially, the quantizer resolution is set to 8 bits. The choice of starting resolution is arbitrary since the AR algorithm will automatically adjust the quantizer resolution to the correct value. After each

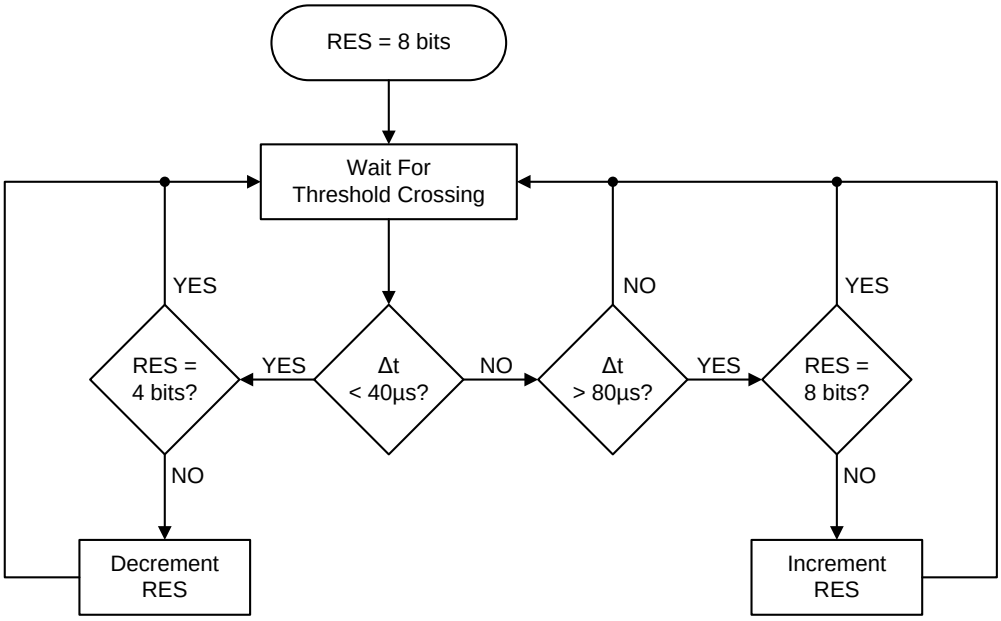


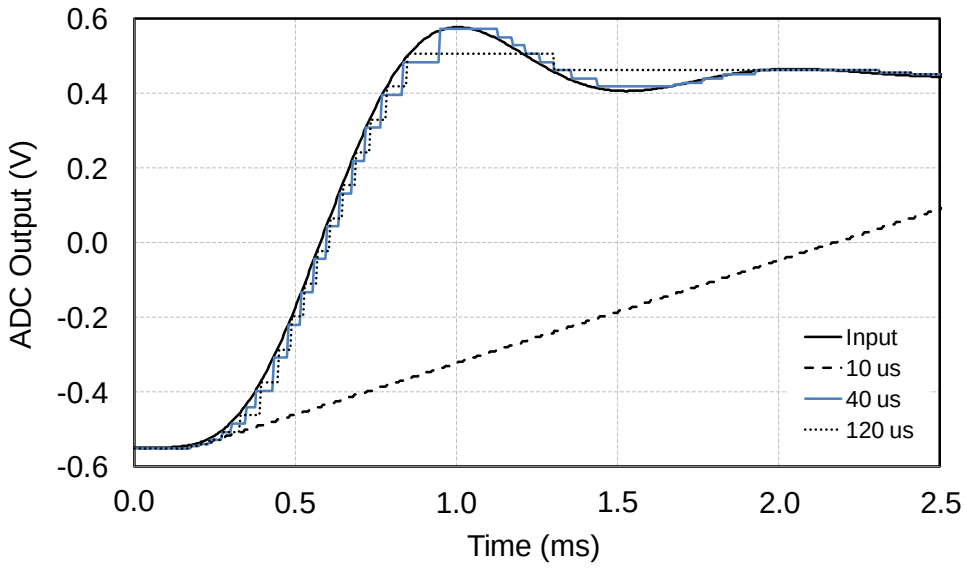
Figure 4.3: Simplified flowchart of the AR algorithm. *RES* is the quantizer resolution and *Δt* is the time since the previous threshold crossing.

sample, the value of Δ*t* is compared to the two Δ*t* thresholds. If Δ*t* is less than 40μs, the quantizer resolution is decreased by 1 bit, unless it is already at the minimum resolution of 4 bits. If Δ*t* is greater than 80μs, the quantizer resolution is increased by 1 bit, unless it is already at the maximum resolution of 8 bits. Otherwise, the quantizer resolution is left unchanged.

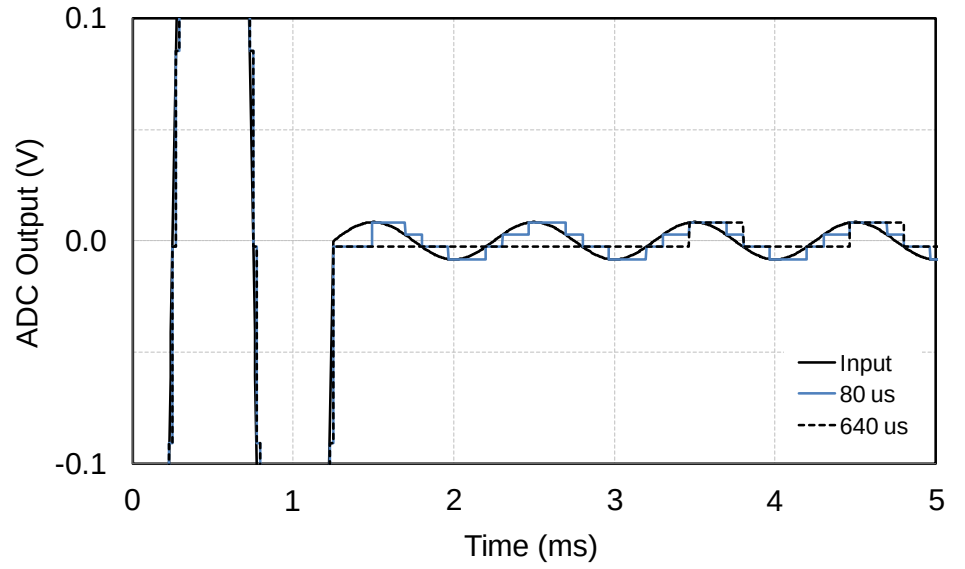
When determining how to set the Δ*t* thresholds, two main requirements for the AR algorithm were taken into consideration. First, when the quantizer resolution is initially set to its maximum value, the AR algorithm must respond fast enough to an increase in input slope to avoid slope overload distortion for all possible inputs. The input signal with the worst case increase in slope would be a full-scale step function, which would be filtered at the 1kHz bandwidth of the ADC in a practical system. Fig. 4.4(a) shows the response of the ADC to this particular input signal, simulated in MATLAB, for different values of the lower Δ*t* threshold

and a loop-delay of 20μs (equal to the measured value for the prototype chip). For each simulation, the upper Δt threshold was set to twice the value of the lower Δt threshold. When the lower Δt threshold was set to 10μs, the AR algorithm failed to decrease the quantizer resolution, which led to slope overload distortion. Increasing the lower Δt threshold to 40μs allowed the AR algorithm to adjust the quantizer resolution fast enough for the ADC to accurately follow the input signal. Increasing the lower Δt threshold beyond 40μs decreases the response time of the AR algorithm to increases in input slope, but also increases the response time to decreases in input slope. This prevents the AR algorithm from increasing the quantizer resolution fast enough to track the ringing at the top of the input signal, when the lower Δt threshold was set to 120μs.

The second main requirement is that when the quantizer resolution is initially set to its minimum value, the AR algorithm must respond fast enough to a drop in input amplitude to avoid missing any small amplitude details in the input signal. The worst case input would be a sinusoid with frequency equal to the bandwidth of the ADC, and amplitude that drops instantly from full-scale to the minimum value within the dynamic range of the ADC. Fig. 4.4(b) shows the simulated response of the ADC to this particular input signal, for different values of the upper Δt threshold and a lower Δt threshold of 40μs and loop delay of 20μs. The y-axis has been zoomed in to clearly show the response when the input amplitude is minimum. When the upper Δt threshold is set to 80μs, the AR algorithm increases the quantizer resolution fast enough for the ADC to accurately follow the input signal. In order to prevent the time between samples from dropping



(a)



(b)

Figure 4.4: Simulated response of the ADC for different values of the Δt thresholds used to control the AR algorithm. (a) Response to a step function filtered at the bandwidth of the ADC, for different values of the lower Δt threshold. (b) Response to a sinusoidal input with frequency equal to the bandwidth of the ADC, and amplitude which drops from full-scale to the minimum value within the dynamic range of the ADC, for different values of the upper Δt threshold.

below the lower Δt threshold when the quantizer resolution is increased, the upper Δt threshold should not be decreased below 80µs. Increasing the upper Δt

threshold beyond 80μs increases the response time of the AR algorithm to decreases in input slope, or in this case decreases in input amplitude. This causes the ADC to stop tracking the input signal for two cycles after the amplitude drops, when the upper Δt threshold is set to 640μs.

As mentioned earlier, several details of the AR algorithm were not discussed in order to simplify the explanation. It was previously stated that the quantizer resolution is increased only after a sample occurs. A problem will occur if the input does not cross a threshold level due to the peak-to-peak voltage variation decreasing to less than the distance between threshold levels. Since no threshold is crossed, the quantizer resolution will not be increased and the ADC will not follow the input. In order to overcome this problem, instead of only increasing the quantizer resolution after a sample occurs, the resolution controller monitors the value of the timer and increases the quantizer resolution by 1 bit every 80μs that a threshold level is not crossed. This guarantees that the quantizer resolution will still be increased even when no threshold is crossed. One problem with changing the threshold levels at a time other than when a sample occurs is that the value of the input signal is unknown. If the input has already crossed one of the new threshold levels, distortion will occur due to the amplitude of the next sample being wrong. MATLAB simulations have shown that this distortion is prevented by not outputting any samples which occur within 50μs after the quantizer resolution is increased. The oversampling nature of the ADC prevents this from degrading the output.

4.5 Circuit Design

This section describes the implementation of each of the blocks in Fig. 4.2(a). In order to increase the power efficiency for low-frequency sensor applications, the analog blocks were optimized for subthreshold operation and a supply voltage of 0.7V in a 0.18μm CMOS process. The control logic and resolution controller were implemented in an FPGA to allow testing of various algorithms for controlling the quantizer resolution.

4.5.1 DAC

The DACs were implemented with the 10-bit fully-differential hybrid switched-capacitor/resistor-string architecture shown in Fig. 4.5 [76], where one side has been omitted for clarity. The resolution was increased by 2 bits beyond the maximum quantizer resolution of the ADC in order to allow the comparator offset to be calibrated to less than 1 LSB. The offset calibration scheme will be discussed in more detail in Section 4.5.3.

One drawback to switched-capacitor DACs is the decreased accuracy of the output over time due to leakage currents [43], which is referred to as output drift. This becomes more of a concern in asynchronous designs where the time between when the DAC is reset will increase as the sample rate decreases. The architecture in Fig. 4.5 was chosen for the low level of output drift provided by the fully-differential design and active feedback. Transient simulations were run to verify that the leakage currents were low enough to avoid degrading the output resolution of the ADC when the sample rate reached its minimum value, which is calculated later. In order to eliminate output drift altogether, the DAC could be

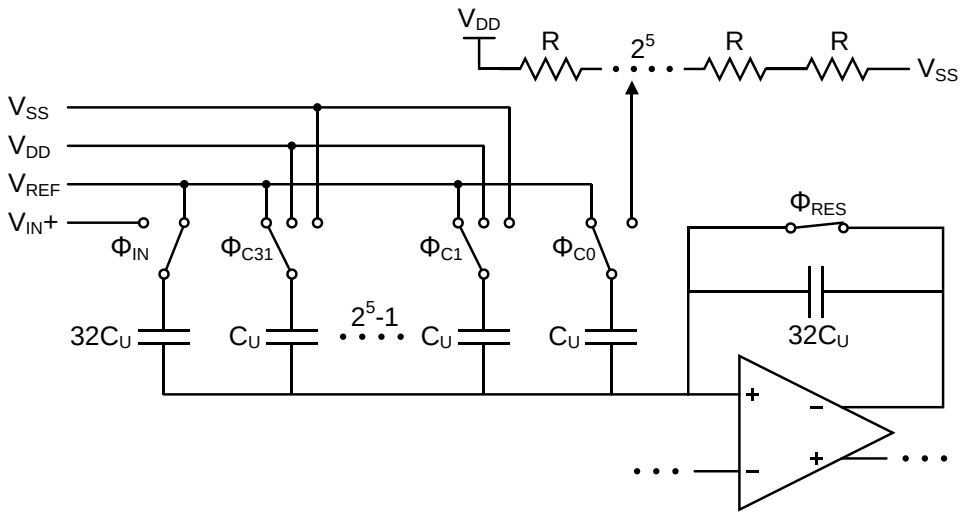


Figure 4.5: 10-bit fully-differential hybrid switch-capacitor/resistor-string DAC and example timing waveform. For clarity, only one side is shown.

implemented as a resistor string. A dynamic switched-capacitor DAC without a feedback amplifier could also be used to decrease static power consumption [64]. These topologies were not chosen for the current design in order to avoid the increased wiring complexity of the resistor string DAC and the increased output drift of the dynamic switched-capacitor DAC.

The DAC operates similarly to standard charge-redistribution architectures [76]. The DAC is reset by closing the switch controlled by Φ_{RES} and connecting the switches controlled by Φ_{C31-0} and Φ_{IN} to V_{REF}, which was set to $V_{DD}/2$. This discharges the feedback capacitor around the amplifier and connects the bottom plate of the remaining capacitors to V_{REF}. The feedback switch is then opened and the output is set according to the DAC input word. The most significant bit (MSB) of the digital input sets the polarity of the differential output voltage. The next five MSBs configure the thermometer encoded capacitor array. If the input is positive, N capacitors, where N is equal to the corresponding 5-bits of the digital

input, in the top half array are connected to V_{DD} and N capacitors in the bottom half array are connected to V_{SS}. The connections are switched if the polarity is negative. The LSB capacitor is then set to one of the taps of the resistor ladder based on the four least significant bits (LSBs). If the input polarity if positive, a tap above $V_{DD}/2$ is chosen, while a tap below $V_{DD}/2$ is chosen if the input polarity is negative. The differential output voltage of the DAC will then equal

$$V_{OUT,DIFF} = \pm \frac{V_{DD}}{2^{N-1}} \sum_{i=0}^{N-2} D_i \cdot 2^i \,, \tag{4.2}$$

where the polarity is determined by the MSB of the digital input, and the D_i*'s* are the remaining 9 bits of the binary encoded digital input, which determine how the switches $\Phi_{C31\text{-}0}$ are configured.

An additional capacitor, with value equal to that of the feedback capacitor, is placed in parallel with the main capacitor array. This capacitor is connected to V_{REF} when the DAC is reset and to the input signal when the DAC output is set. This effectively subtracts the input signal from the threshold levels (D_{OUT} ± *STEP*) within the DAC. This allows the output of the DAC to be directly connected to a comparator, in order to detect when the input crosses one of the threshold levels. An additional benefit is that the DAC differential output is kept less than the threshold level step size. This allows for: rail-to-rail input swing, which is important due to the low supply voltage; decreased settling time; and decreased leakage through the feedback switch, which reduces output drift.

The linearity of the DAC is determined mainly by the matching in the capacitor array and to a lesser extent the resistor ladder. Any non-linearity will limit the effective resolution of the DAC and the maximum achievable SNDR of

the asynchronous ADC, as was mentioned in Section 3.2.1. The input-referred noise of the amplifier will also reduce the resolution of the DAC. The input-referred offset of the amplifier will lead to an offset in the threshold levels, which is discussed in more detail Section 4.5.3.

The size of the unit capacitors, C_U, was set to 150fF to allow for 10-bit matching. The unit resistor size, R, was set to 50kΩ in order to minimize power dissipation without reducing the settling time of the DAC. The switches were implemented with nMOS transistors with a width and length of 0.4µm and 0.26µm, respectively. The simulated reset and settling times of the DAC were 2.2µs and 5.7µs, respectively. In a fully integrated design, these times would approximately set the loop-delay of the ADC, which would give a maximum sample rate of approximately 100kS/s.

4.5.2 Operational Transconductance Amplifier

The amplifier in the DAC was implemented as a two-stage Miller-compensated operational transconductance amplifier (OTA) [77]. This architecture, which is shown in Fig. 4.6, was chosen for its ability to provide reasonable gain without restricting the output swing, which was important due to the low supply voltage. The common-mode feedback (CMFB) circuit is essentially a copy of the first stage of the OTA and sets the output common-mode voltage of the OTA to V_{CM} by adjusting the bias voltage V_{CMFB}. The output common-mode voltage and the bias nodes (V_B) were set by an external voltage source to $V_{DD}/2$. In order to make the amplifier independent of supply voltage, temperature, and process (PVT) variations, V_B could be set with a constant-g_m bias

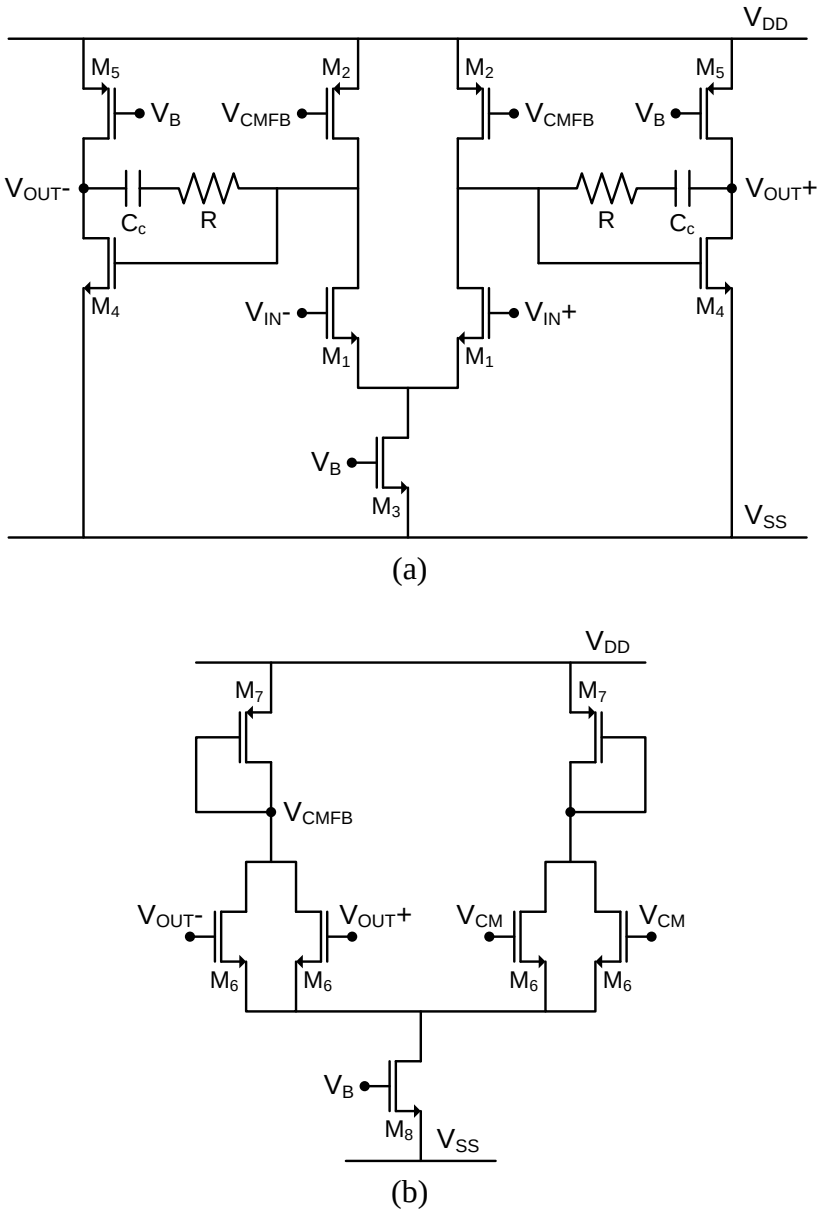


Figure 4.6: (a) Two-stage Miller-compensated OTA. (b) Common-mode feedback circuit.

circuit in the future [77].

The low supply voltage and input common mode voltage of $V_{DD}/2$ left around only 250mV to bias the gate-source voltage of the input transistors. This would bias the input transistors in the deep subthreshold region due to the nominal transistor threshold voltage (V_T) of 400mV. This would limit the performance of

Table 4.1: OTA component sizes.

Device	Size	Device	Size
M_1	9µm/1µm	M_6	4.5µm/1µm
M_2	18µm/1µm	M_7	18µm/1µm
M_3	6µm/1µm	M_8	6µm/1µm
M_4	6µm/1µm	R	100kΩ
M_5	24µm/1µm	C	1.35pF

the amplifier due to the constrained current drive capabilities and g_m, and large mismatch due to V_T variations of transistors biased in this region. In order to improve the amplifier performance, the V_T was decreased so that the input transistors were biased closer to the moderate inversion region. This was accomplished by biasing the body potentials of all the transistors at $V_{DD}/2$ and increasing the gate length of the transistors to 1µm. By slightly forward biasing the body of the transistors, the threshold voltage was decreased through the body effect [77]. In order to allow the body of the nMOS transistors to be biased separately from the substrate, triple-well transistors, which have an isolated body terminal, were used. Increasing the gate length to 1µm overcame the reverse short channel effect [78], which causes an increase in V_T at short channel lengths.

The size of each of the components of the OTA is given in Table 4.1. The DC gain of the OTA is given by

$$A_v = g_{m1} g_{m4} (r_{o1} \parallel r_{o2})(r_{o4} \parallel r_{o5}), \tag{4.3}$$

where g_m and r_o are the transconductance and output resistance, respectively, of the corresponding transistors. The frequency response is determined by the output poles of the first and second stages, which are given by

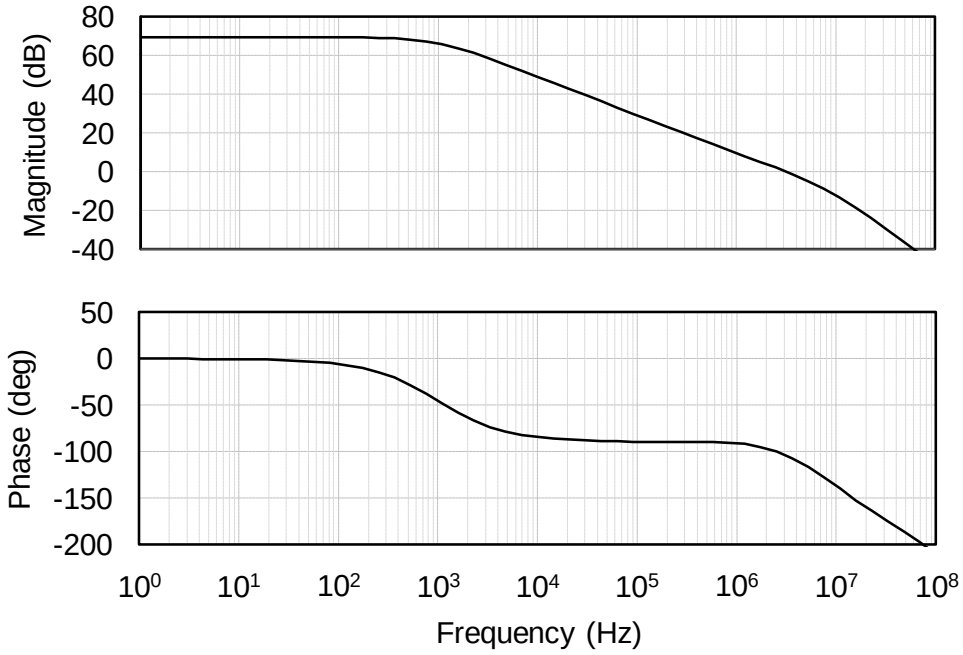


Figure 4.7: OTA simulated open-loop frequency response.

$$p_1 = -\frac{g_{m1}}{g_{m4}(r_{o4} \parallel r_{o5})C_c} \tag{4.4}$$

and

$$p_2 = -\frac{g_{m4}}{C_L}, \tag{4.5}$$

where C_L is the load capacitance. Simulations show that for a 5pF load on each output, the OTA achieves a DC gain of 69.5dB, a unity-gain bandwidth of 3.5MHz, a phase margin of 75^0, and has a static power consumption of 7.5µW at a supply voltage of 0.7V. The simulated open-loop frequency response is shown in Fig. 4.7.

4.5.3 Comparator and Offset Calibration

A continuous-time topology was chosen for the comparator in order to avoid the need for a clock and make it possible for the ADC to be integrated into a

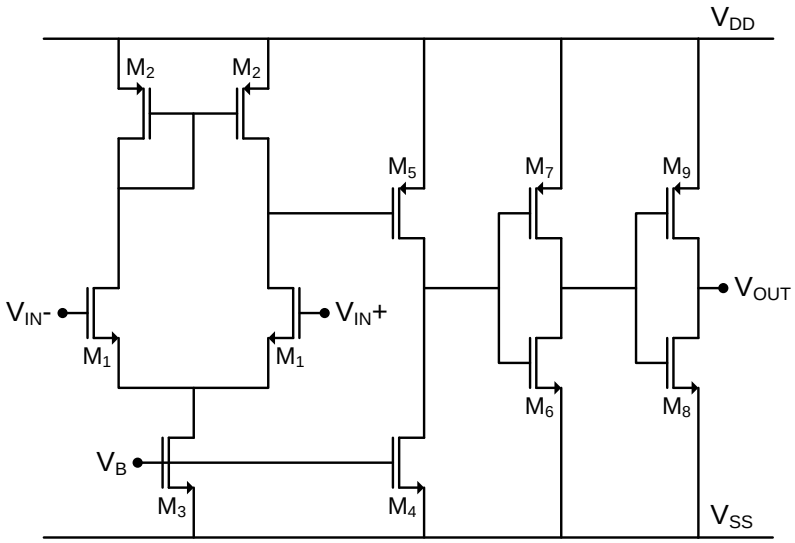


Figure 4.8: Continuous-time Comparator.

clockless continuous-time DSP system in the future [25]. The Class-A amplifier structure shown in Fig. 4.8 was chosen for its ability to operate at low supply voltages. Two inverters were added at the output to allow for rail-to-rail (logic level) output swing and to prevent an increase in the decision time due to the output being brought off chip to interface with the control logic. The size of each of the components of the comparator is given in Table 4.2. The static power consumption from simulations was 1.4µW at a supply voltage of 0.7V.

The decision time of the comparator will add to the time error in the sample

Table 4.2: Comparator component sizes.

Device	Size	Device	Size
M_1	6µm/1µm	M_6	6µm/1µm
M_2	5µm/1µm	M_7	24µm/1µm
M_3	3µm/1µm	M_8	18µm/1µm
M_4	3µm/1µm	M_9	72µm/1µm
M_5	12µm/1µm		

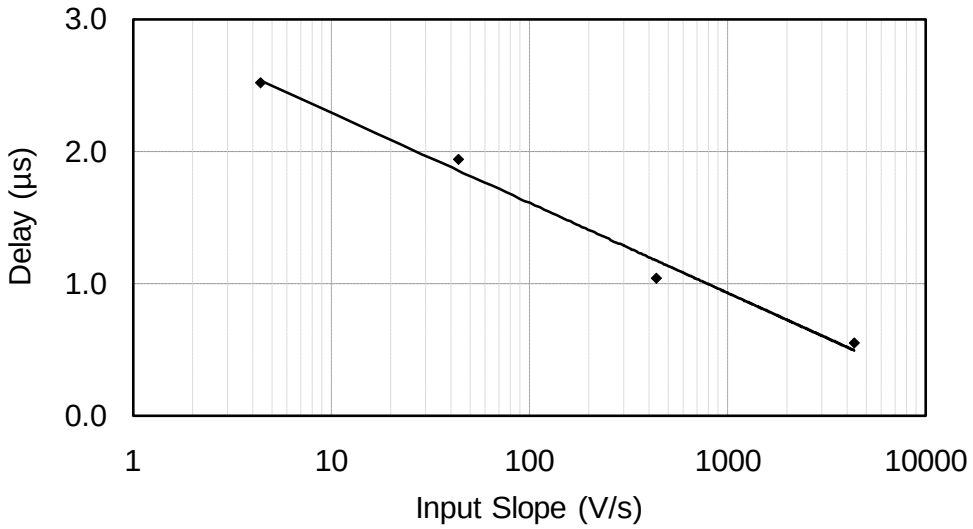


Figure 4.9: Simulated input dependent decision delay of the continuous-time comparator.

points. As described in Chapter 3, the time error will create an error in amplitude which is dependent on the slope of the input signal. Simulations show a decision time of 0.55µs for the worst case input slope of 4375V/s and a supply voltage of 0.7V. This value is less than the resolution of the timer (1µs) and therefore will have minimal effect on the output resolution. As shown in Fig. 4.9, for every order of magnitude decrease in the input slope, simulations show only a two-fold increase in the decision time. Therefore, the error in amplitude will decrease as the input slope decreases and will not affect the output resolution of the ADC.

Any difference in offset between the two comparators will also affect the output resolution of the ADC. The difference in offset can be viewed as a shift in the lower threshold levels in Fig. 4.2 relative to the upper threshold levels. This creates an error in amplitude equal to the comparator offset for samples which occur when the input crosses a lower threshold level compared to samples which occur when the input crosses an upper threshold level. MATLAB simulations

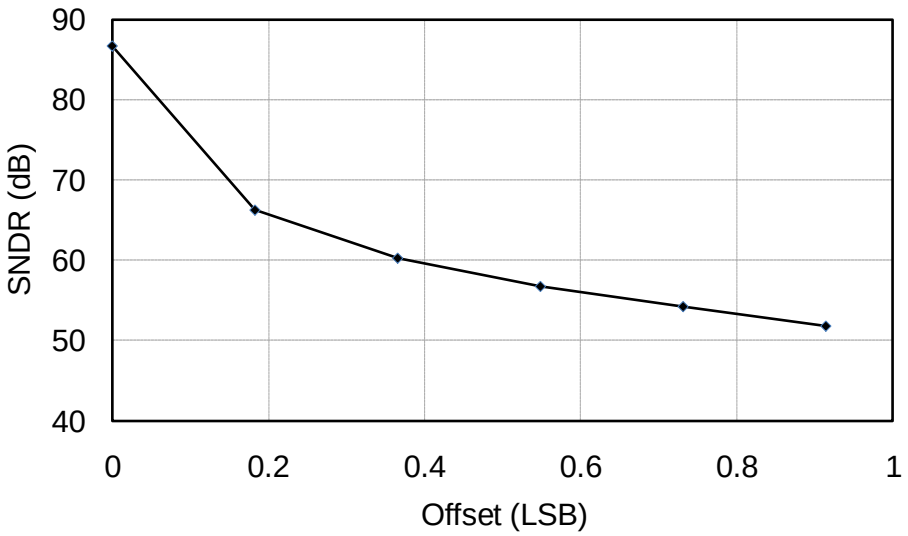


Figure 4.10: Simulated SNDR versus the difference in comparator offset for a 34Hz sinusoidal input with amplitude of 535mV, a timer resolution of 1MHz, a loop-delay of 20μs, and a quantizer resolution of 8 bits.

show that this leads to an increase in harmonic distortion and a decrease in the SNDR of the ADC. The simulated results for a 34Hz sinusoidal input with amplitude of 535mV, a quantizer resolution of 8-bits, a timer frequency of 1MHz, and a loop-delay of 20μs are shown in Fig. 4.10. The reconstructed SNDR is seen to quickly decrease as the difference in comparator offset is increased from 0 to 1 LSB, where LSB is the step size at a quantizer resolution of 8 bits.

The difference in offset may also cause the ADC output to toggle between two adjacent levels if the lower threshold is moved to less than 1 LSB below the upper threshold. This can be explained by assuming a slowly changing input has just increased past the upper threshold level. The ADC will then output the level crossed and increment both threshold levels by 1 LSB. Since the lower threshold was less than 1 LSB below the threshold just crossed, it will now be above the input signal. Therefore, the ADC will immediately register a decrement. The

ADC will continue to alternately increment and decrement the output as long as the input signal remains just above the initial threshold level which was crossed. An illustration of this effect is given in [79].

To prevent a decrease in output resolution and an increase in activity due to output toggling, the difference in comparator offset can be cancelled by adding an offset to one of the DACs. For chip testing, the offset was manually calibrated. The quantizer resolution was set to 8 bits and a low frequency sinusoid was input to the ADC. A negative offset of a few LSB was initially added to the lower DAC to ensure that the upper and lower thresholds were more than 2 LSB apart. The added offset was then made more positive in increments of 0.5 LSB until the ADC output began toggling between two values at the top of the input sinusoid. This indicated that the lower threshold level was now between 0.5 and 1 LSB below the upper threshold. An additional 1 LSB was then subtracted from the offset added to the lower DAC to bring the lower threshold level approximately 2 LSB below the upper threshold, which is the ideal value. This final offset was then added to the lower DAC for all measurements. According to simulations, the 0.5 LSB resolution of the calibration scheme is sufficient to prevent the SNDR from being degraded by more than 3dB due to any remaining difference in offset.

4.5.4 Control Logic and Timer

As previously mentioned, the control logic and resolution controller were implemented in an FPGA. This allowed the ADC to function as a test-bed where various algorithms for controlling the quantizer resolution could be tested and characterized. The off-chip control logic also allowed the prototype ADC to be

configured with a fixed quantizer resolution. This allowed the tradeoff between dynamic range and input bandwidth to be measured and used to demonstrate the improved performance offered by the proposed AR algorithm.

The timer was implemented as a counter, which was incremented by a clock on the demonstration board used for the FPGA, and reset by the control logic. The size of the counter was set to 16 bits. For testing purposes, the counter clock frequency was set to 1MHz. This frequency was chosen so that the measured SNDR for a full-scale sinusoidal input would not be limited by the timer resolution ratio for frequencies below the bandwidth of the ADC. In order to prevent the timer from overflowing, the controller was configured to output a sample and reset the timer and threshold levels when the counter reached its maximum value before a threshold level was crossed. This set the minimum sample rate of the ADC to 15S/s.

4.6 Measured Results

A prototype of the proposed AR asynchronous ADC was fabricated in a 1.8V 0.18μm CMOS process and packaged in a 68-pin leadless chip carrier ceramic package. The nominal threshold voltages of the nMOS and pMOS transistors were approximately 400mV and -400mV, respectively. The digital logic was implemented off-chip in an Altera Cyclone II 2C35 FPGA embedded on an Altera DE2 development board. The die micrograph is shown in Fig. 4.11. The core area of the ADC, not including off-chip logic, was 0.96mm^2. The analog supply, including the reference voltage for the DACs, was set to 0.7V. The supply for the

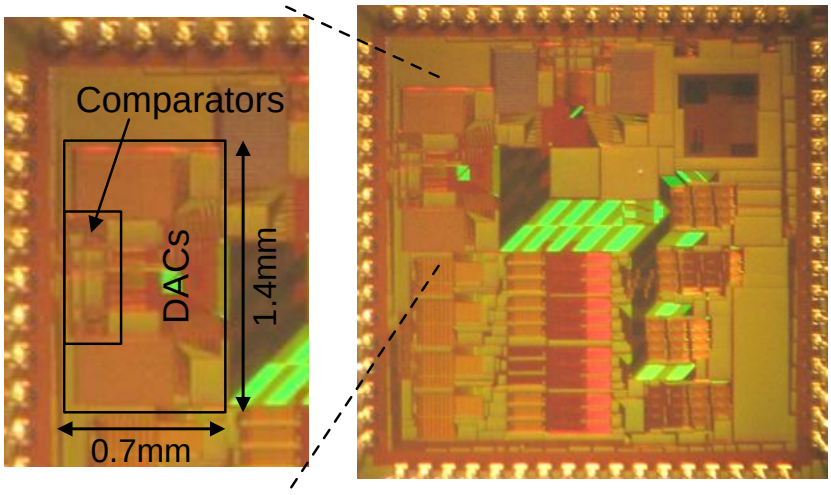


Figure 4.11: Die micrograph of the prototype ADC.

digital I/O, which drove the DAC switches, was set to 1.4V in order to reduce the switch on-resistance and allow for interfacing with the external FPGA. The full test setup is presented in Appendix A.

In order to allow standard FFT techniques to be used to calculate the signal-to-noise and distortion ratio (SNDR) and output spectrum, a 6^{th}-order polynomial interpolator in MATLAB was used to reconstruct a synchronous signal from the asynchronous samples. Reconstruction of synchronous signals from the outputs of level-crossing ADCs, by way of an interpolator, has been analyzed in [21], [68], and [69]. The interpolator output was used to calculate the SNDR values presented in Figs. 4.12-4.14 and Table 4.3. As discussed in Chapter 3, to save power in an actual system, the interpolation could be performed after the asynchronous data has been transmitted to a central processing node or other device with ample power and processing resources. If real time processing was needed, a DAC could be used instead, to reconstruct a continuous time signal. Similar results could be expected, based on the analysis of a continuous-time DSP

system in [26] and [80].

4.6.1 ADC Performance with AR Algorithm

The performance of the prototype ADC was measured with its quantizer resolution varied between 4 and 8 bits by the AR algorithm and a timer resolution of 1µs. The comparator offset was manually calibrated by adding a constant offset to each of the DACs. For all tests, the sample rate of the interpolator output was adjusted to equal 19.3 times the input frequency. This allowed the SNDR of the ADC to be measured accurately, and prevented the reconstruction algorithm from filtering out any significant harmonics as the input frequency increased, as is discussed in [68].

The loop-delay of the ADC was measured to be 20µs, which set the equivalent maximum sampling rate to 50kS/s. The maximum sampling rate would increase to 100kS/s in a fully integrated design by eliminating the 10µs required to load the DAC control words from the FPGA. The long load time was due to the control words being loaded over a shared bus and the 1MHz clock used to control the FPGA logic. The peak SNDR and input bandwidth of the ADC were measured to be 52.2dB (8.4 effective number of bits) and 1kHz, respectively, for a sinusoidal full-scale differential input with peak-to-peak voltage of 1.4V. The SNDR was measured by removing even order distortion caused by some of the ESD diodes in the prototype chip being connected to the wrong supply. This will be explained in more detail at the end of this section. The maximum possible bandwidth of 4kHz, given by (4.1), could be achieved by reducing the minimum quantizer resolution to 2 bits.

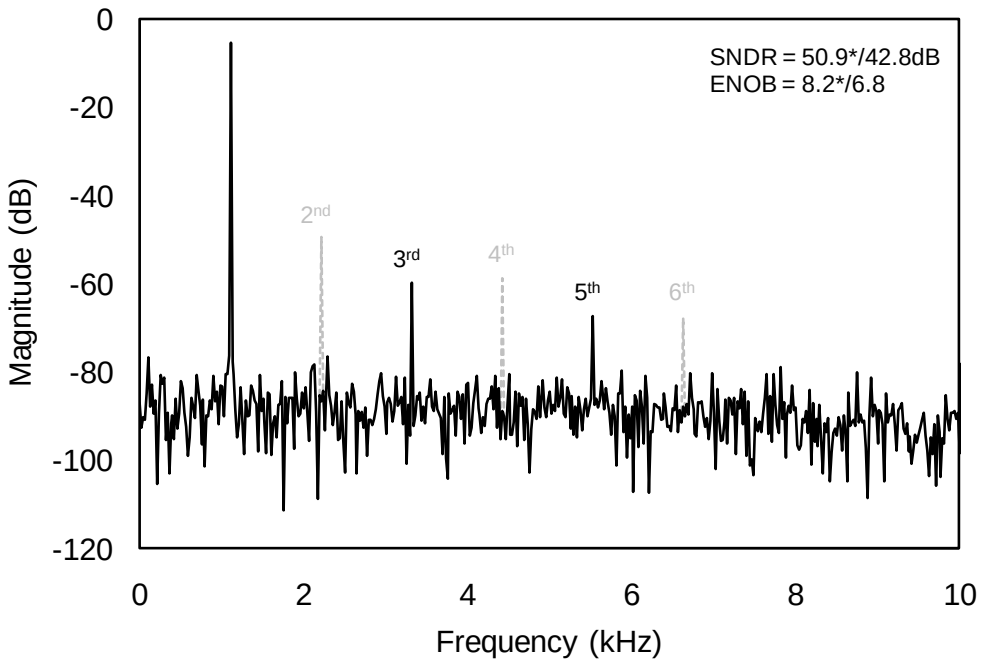


Figure 4.12: ADC output spectrum for a 1.1kHz sinusoidal differential input with amplitude of 570mV. A 6th-order polynomial interpolator was used to reconstruct a synchronous signal at 20kS/s from the sampled output, so that standard FFT techniques could be used to compute the output spectrum. Even order distortion is due to some of the ESD diodes in the prototype chip being connected to the wrong supply. SNDR and ENOB values are given both not including* and including this distortion.

An example output spectrum for a 1.1kHz sinusoidal differential input, with amplitude of 570mV, is shown in Fig. 4.12. The even order distortion caused by the ESD diodes has been plotted with a broken line to differentiate it from the spectrum of the core ADC. This figure demonstrates how the input frequency can increase beyond the bandwidth of the ADC without introducing distortion, when the input amplitude is below its full-scale value. The measured power dissipation, not including the off-chip control logic, was 25µW, due mostly to the static power dissipation of the OTA in the DACs. This value could be reduced by implementing the DACs as resistor ladders or dynamic switched-capacitor DACs

without feedback amplifiers, although threshold level drift would become a problem at low sample rates for the latter case. If the control logic and resolution controller are integrated, their power consumption can be estimated based on the reported power consumption of an asynchronous controller used in a recently published medium resolution SAR ADC [81]. Scaling the sample rate of this ADC to the worst case average sample rate of the prototype ADC (16kS/s) gives an estimated power consumption of approximately 100nW. This is a conservative value since the complexity of the control logic for the prototype ADC is less than that of a SAR ADC. The resolution controller will not add much complexity since it only has to compare the output of the timer to two thresholds, and then increment or decrement the quantizer resolution when one of the thresholds are crossed. Therefore, integrating the control logic and resolution controller will not noticeably increase the power consumption of the ADC. An overview of the performance results for the prototype ADC is given in Table 4.3. The SNDR and ENOB values are given with both the even order distortion from the ESD diodes removed from the output spectrum and with it included.

The even order distortion in the output spectrum of the ADC was caused by the ESD diodes for the digital I/O pads in the prototype chip being mistakenly connected to the analog supply. The 700mV difference between the supplies led to the diodes that were connected to the positive supply turning on whenever the corresponding digital I/O pins toggled high. It was verified that when the supplies were brought within 300mV of each other, by setting the analog supply to 0.9V and the digital supply to 1.2V, the diodes no longer turned on and the distortion

Table 4.3: ADC performance summary.

Parameter	Value
Technology	0.18µm CMOS
Package	68-pin LLC
Supply Voltage	0.7 V
Quantizer Resolution	Variable (4 to 8 bits)
Timer Resolution	1 µs
Maximum Sampling Rate	50 kS/s
Input Bandwidth	1 kHz (for full-scale input)
Peak SNDR*	52.2**/43.2 dB
ENOB*	8.4**/6.9 bits
Dynamic Range	40.2 dB
Differential Input Swing	1.4 V_{p-p}
Power Consumption	25 µW (w/o off-chip logic)
Area	0.96mm^2 (w/o off-chip logic)
*calculated from output of interpolator **does not include even order distortion caused by incorrect connection of ESD diodes in the prototype chip	

disappeared. These supply levels were not used to measure the ADC performance because the CMFB circuit in the OTAs, which was originally designed to operate at a supply level of 0.5V, was not fully stable at a supply voltage of 0.9V. This led to an increase in the odd order harmonics and degradation of the overall SNDR. It was decided that the best way to measure the actual performance of the core ADC was to allow the ESD diodes to turn on and then subtract the even order distortion from the output spectrum since it was not due to the core ADC design. If included, the only change in the results would be a decrease in peak SNDR, as shown in Table 4.3. This would not affect the main thesis of this work, which is that the proposed AR algorithm overcomes the tradeoff between dynamic range

and input bandwidth in asynchronous ADCs. In future designs the distortion will be removed by connecting the ESD diodes to the proper supply.

4.6.2 Comparison to Fixed Quantizer Resolution

In order to measure the tradeoff between input bandwidth and dynamic range when the quantizer resolution is fixed, and to evaluate the improved performance offered by the AR algorithm, the dynamic range and input bandwidth of the prototype ADC were measured with its quantizer resolution fixed at 4, 6, and 8 bits, and varied by the AR algorithm.

In order to calculate the dynamic range at each quantizer resolution setting, the SNDR was measured versus input amplitude for a sinusoidal differential input with frequency of 40Hz. The results are shown in Fig. 4.13 for each fixed quantizer resolution, and for when the quantizer resolution was varied according to the AR algorithm. The dynamic range was calculated as the difference between the amplitude that gave the peak SNDR and the amplitude where the SNDR dropped below zero. The results show that for a fixed quantizer resolution, increasing the resolution from 4 to 8 bits increases the dynamic range from 19.6dB to 40.2dB. When the AR algorithm was used to vary the quantizer resolution, the maximum dynamic range of 40.2dB was achieved. These values match the theoretical values given by (3.9) within a couple dB.

In order to calculate the input bandwidth at each quantizer resolution setting, the SNDR was measured versus input frequency for a sinusoidal differential input with amplitude of 570mV. The results are shown in Fig. 4.14 for each fixed quantizer resolution, and for when the quantizer resolution was varied according

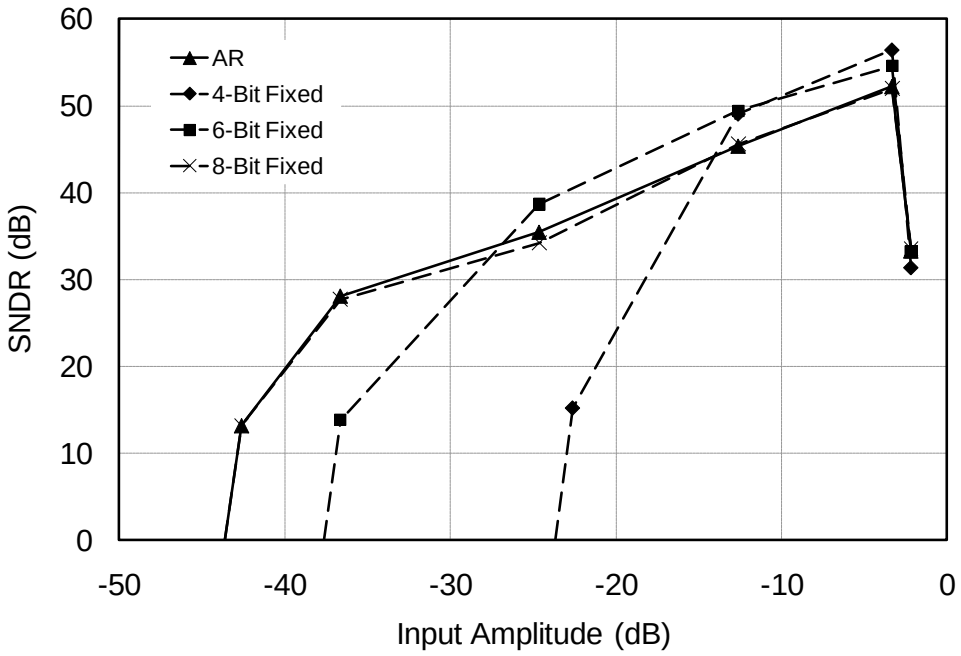


Figure 4.13: Comparison of SNDR versus input amplitude for when the quantizer resolution was set by the AR algorithm and fixed at 4, 6, and 8 bits. The SNDR was calculated for a 40Hz sinusoidal differential input.

to the AR algorithm. For a conventional ADC, the input bandwidth is calculated as the input frequency where the SNDR drops 3dB below its peak value. This definition of bandwidth is not relevant for asynchronous ADCs, since, according to (3.3), the SNDR will decrease at a rate of 20dB/decade as the input frequency increases. The bandwidth is instead defined, by (3.7), as the maximum input frequency which does not violate (3.6). The input frequency where this occurs is easily calculated from the plots in Fig. 4.14 by determining when the SNDR begins to decrease at a rate greater than 20dB/decade.

The results in Fig. 4.14 show that when the quantizer resolution is fixed at 4, 6, and 8 bits, the input bandwidth is 1.2kHz, 300Hz, and 78Hz, respectively. These values match the theoretical maximum frequencies given by (3.7). When the quantizer resolution is fixed to either 6 or 8 bits, the SNDR does not drop

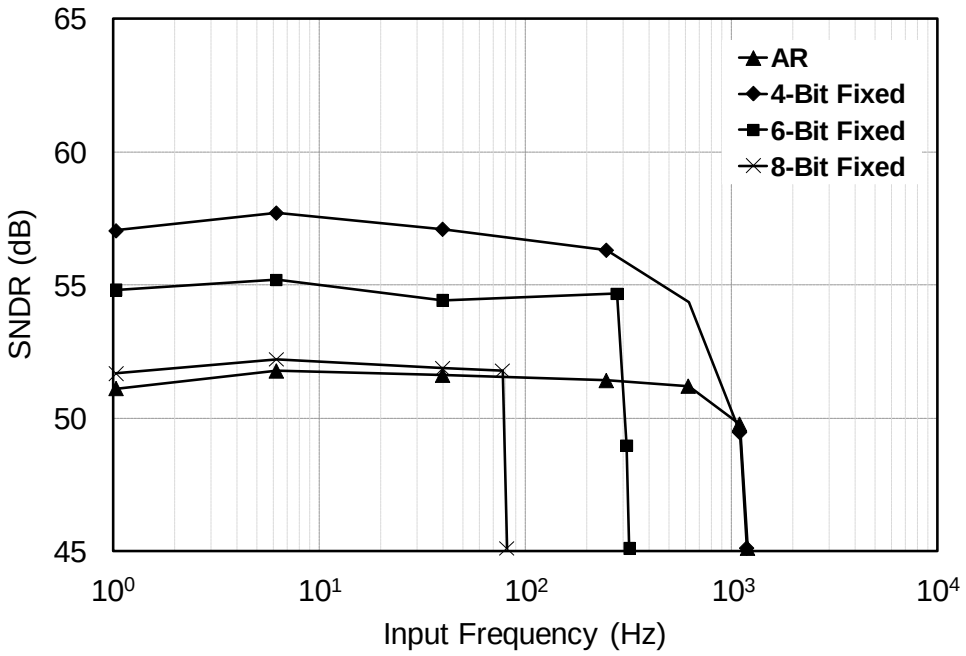


Figure 4.14: Comparison of SNDR versus input frequency for when the quantizer resolution was set by the AR algorithm and fixed at 4, 6, and 8 bits. The SNDR was calculated for a sinusoidal differential input with amplitude of 570mV.

below its peak value for frequencies up to the bandwidth of the ADC. This is due to the fact that at low frequencies, the SNDR is limited by the accuracy of the analog components rather than by the resolution ratio. When the quantizer resolution is set to 4 bits, the SNDR drops to 8dB below its peak value at the bandwidth of the ADC. When the AR algorithm is used to vary the quantizer resolution, the maximum bandwidth of 1.2kHz is achieved. At this frequency, the SNDR drops 2dB below its peak value.

The dynamic range and input bandwidth calculated from Figs. 4.13 and 4.14 are plotted in Fig. 4.15 for each fixed quantizer resolution and for when the quantizer resolution was varied by the AR algorithm. A linear regression line, drawn between the three fixed resolution data points, illustrates the tradeoff between dynamic range and input bandwidth when the quantizer resolution is

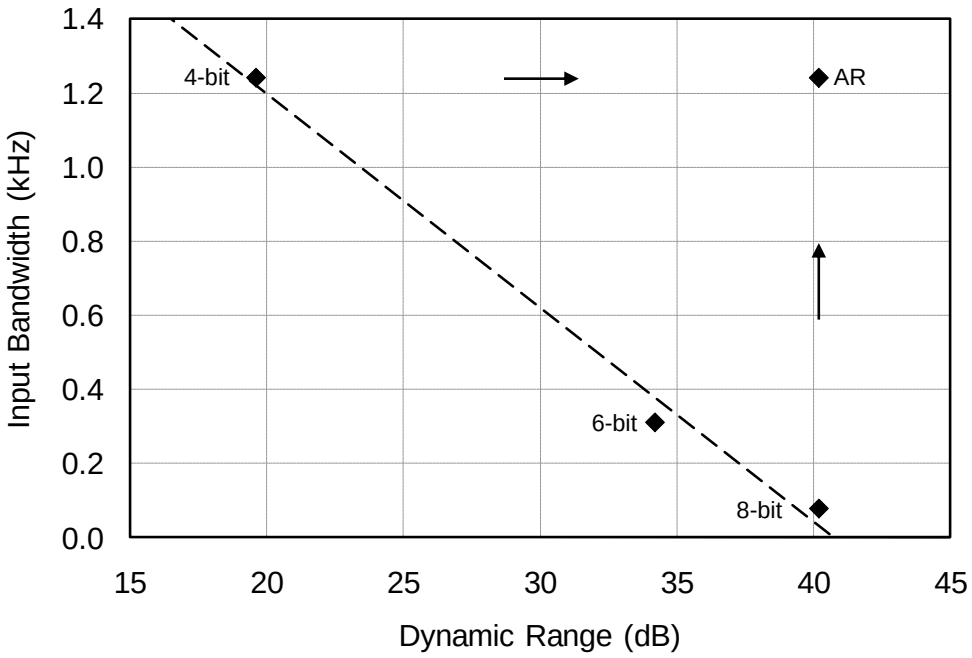


Figure 4.15: Comparison of input bandwidth versus dynamic range, calculated from Figs. 4.13 and 4.14, for when the quantizer resolution was set by the AR algorithm and fixed at 4, 6, and 8 bits.

fixed. The data point for when the AR algorithm is used to set the quantizer resolution is seen to move off of this line and achieve both the highest dynamic range and input bandwidth. This proves the main thesis of this work, which is that the AR algorithm overcomes the tradeoff between dynamic range and input bandwidth typically seen in asynchronous ADCs.

Several properties of the plots in Figs. 4.13 and 4.14 should be mentioned to clarify the results. In Fig. 4.13, the plots for when the quantizer resolution was varied according to the AR algorithm and fixed at 8 bits are seen to closely match. This is because, at an input frequency of 40Hz, the AR algorithm kept the quantizer resolution set to its maximum value of 8 bits for the full range of input amplitudes which were tested. If the input frequency was increased, the AR algorithm would eventually start to reduce the quantizer resolution at higher input

amplitudes in order to prevent (3.6) from being violated. This would not decrease the dynamic range since, for small amplitude inputs, the quantizer resolution could remain set at its maximum value of 8 bits without violating (3.6), for frequencies up to the bandwidth of the ADC.

In Fig. 4.14, the input bandwidth which was achieved at each resolution was greater than the worst case value from (3.8). This was due to the input amplitude used for this test being less than the full-scale value of 700mV. Also, the SNDR is seen to be constant at low frequencies, instead of decreasing at a rate of 20dB/decade. This is due to the SNDR being limited by the accuracy of the analog components rather than by the resolution ratio. Lastly, the peak SNDR is seen to drop by approximately 5dB when the quantizer resolution is increased from 4 to 8 bits. Measurements show that across all input amplitudes, the average difference in peak SNDR is 2.8dB. This difference is due to the different effect that nonlinearity in the threshold levels, which is caused by nonlinearity in the DAC transfer function, has at each quantizer resolution. A common cause of non-linearity in the type of DAC designed for the ADC is mismatch between the MSB capacitor array and the LSB resistor array. When the quantizer resolution is set to 8 bits, this mismatch increases the overall distortion and reduces the overall SNDR of the ADC. When the quantizer resolution is set to 4 bits, only the MSB array of the DAC is utilized. Therefore, nonlinearity due to mismatch between the two arrays is irrelevant, and does not affect the SNDR. In order to verify this behavior, MATLAB simulations were run. The DAC transfer function was modeled with nonlinearity that would typically be seen when a mismatch was

present between the MSBs and LSBs. The nonlinearity was sized to give a maximum integral nonlinearity (INL) and differential nonlinearity (DNL) of 1 LSB, with respect to a quantizer resolution of 8 bits. The SNDR was then simulated with the quantizer resolution set to 4, 6, and 8 bits. The simulations gave an average decrease in SNDR of 3.6dB when the quantizer resolution was increased from 4 to 8 bits. This corresponds closely with the measured decrease in SNDR. The large increase in dynamic range at higher resolution outweighs this relatively small decrease in peak SNDR.

4.6.3 Accelerometer Test

The low bandwidth and, for many applications, burst like nature of the output from an accelerometer makes it a good application to demonstrate the potential for data compression provided by the prototype asynchronous ADC. The prototype ADC was therefore used to digitize the output of an ADXL335 3-axis accelerometer mounted on an EVAL-ADXL335Z evaluation board from Analog Devices. The accelerometer output bandwidth was matched to the ADC bandwidth of 1kHz using a 4^{th}-order low-pass RC filter. The filtered x-axis output of the accelerometer was then sampled by the prototype ADC as the evaluation board was struck with varying force, resulting in accelerations within the ±3g dynamic range of the accelerometer. The full test setup is presented in Appendix A. The sampled output for a particular 14s period is shown in Fig. 4.16, along with the time varying sample rate. The sample rate scales from the minimum sample rate of the ADC (15S/s) during periods of inactivity, up to a maximum of 20kS/s as the signal activity increases. The ADC thereby provides large

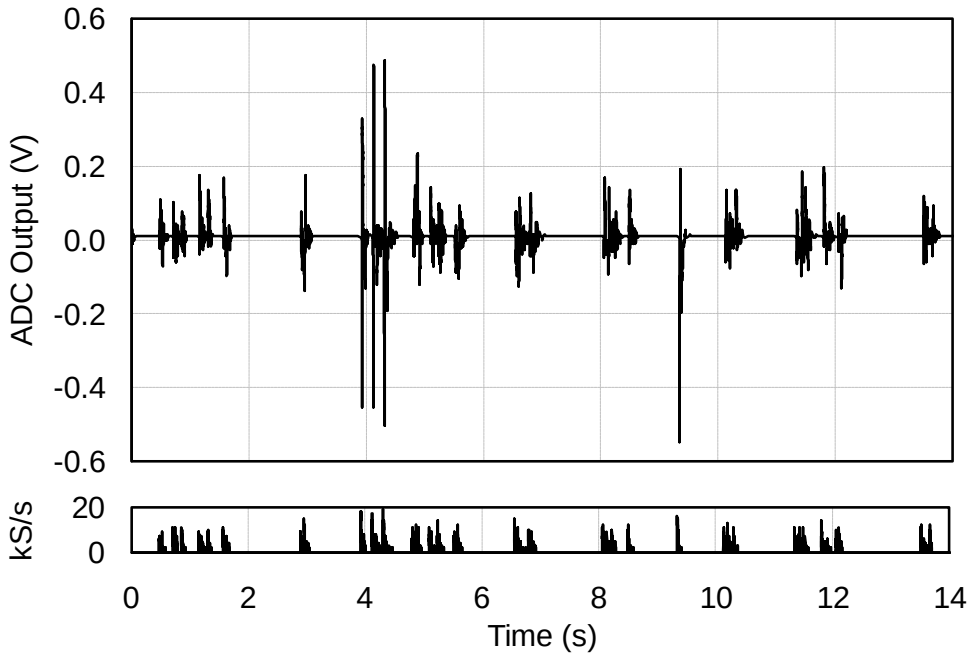


Figure 4.16: Sampled output of the ADXL335 accelerometer and corresponding time varying sample rate.

oversampling when needed while having an average sample rate of only 530S/s, which is lower than the Nyquist rate of 2kS/s.

Compared to an 8-bit conventional ADC with no oversampling, the prototype ADC achieved a data compression ratio of about 4, on a sample rate basis, for this particular example. If the sample rate of the conventional ADC is set to 20kS/s, in order to achieve the same maximum oversampling ratio as the prototype ADC, the compression ratio will increase to 40. As the signal activity decreases, the compression ratio will increase further. The previous compression ratios are based only on the sample rate of the ADCs. If the total bit rate is considered, the compression ratio will decrease by a factor of three, due to the extra 16 bits required to represent the time information. In future designs, the compression ratio can be increased by optimizing the number of time bits for a given

application or by only transmitting the change in amplitude at each sample.

The effective output resolution of the sampled data was determined by also sampling the output of the accelerometer with a NI USB-6009 14-bit 48kS/s data acquisition device (DAQ) from National Instruments. The mean squared error (MSE) between the two sampled outputs was calculated after the output of the prototype ADC was interpolated to match the sampling rate of the DAQ and any difference in gain and offset between the two outputs was removed. For the example in Fig. 4.16, the calculated MSE corresponded to an effective number of bits (ENOB) of 8.2. This value was achieved over multiple runs and was only 0.2 bits lower than the measured peak value reported in Table 4.3. The 8-bit resolution and 25μW power dissipation make the ADC suitable for low-cost accelerometer applications [82].

4.6.4 ECG Test

In order to demonstrate the feasibility of the prototype asynchronous ADC for ECG monitoring applications, the prototype ADC was also used to digitize the output of an ECG electrode connected to a real test subject. A three electrode arrangement was used to measure the ECG signal, where one signal electrode was placed on the left and right shoulders of the test subject and the reference electrode was placed on the right ankle. The electrodes were then connected to the input of a front-end amplifier, which is described in detail in [83]. A buffer was added between the front-end amplifier and the ADC to bring the common mode level within the range required by the ADC. Fig. 4.17 shows the output of the ADC and the time varying sample rate for a period spanning two ECG pulses.

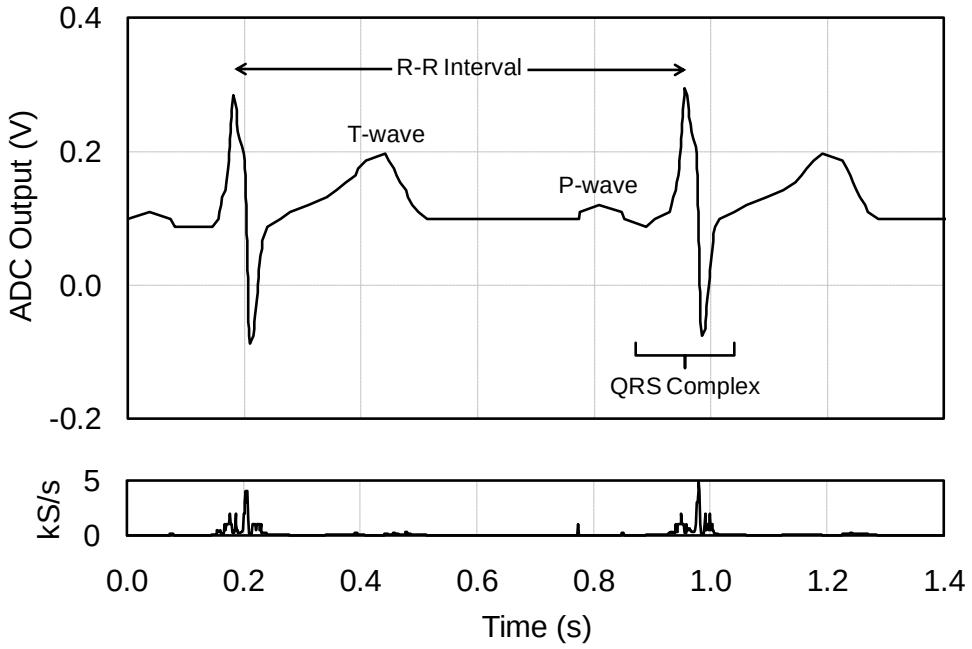


Figure 4.17: Measured ECG signal at ADC output and corresponding time varying sample rate.

The important ECG components such as the QRS complex, the R-R interval, and T and P-waves are clearly visible. The sample rate is seen to vary from the minimum sample rate of the ADC (15S/s) during period of inactivity, up to a maximum level of 5kS/s as the signal activity increases. The ADC thereby provides large oversampling when needed while having an average sample rate of 150S/s.

4.7 Chapter Summary

In this chapter we have presented an asynchronous ADC which achieves data compression for sparse and burst like signals by the inherent signal dependent sampling rate of the asynchronous architecture. The main contribution of this work was the implementation of an AR algorithm which varied the quantizer

resolution of the ADC with the slope of the input signal, in order to overcome the tradeoff between dynamic range and input bandwidth typically seen in asynchronous ADCs. This allowed the maximum possible input bandwidth to be achieved regardless of the dynamic range requirement. By reducing the quantizer resolution during periods of high input slope, further data compression was also achieved.

The presented AR algorithm was developed to improve performance for general low frequency sensor applications. Future work will optimize the algorithm for particular applications, in order to achieve maximum data compression. For example, in a simple heart rate monitor, where only relatively large amplitude pulses need to be resolved, the AR algorithm could be programmed to decrease the quantizer resolution more quickly than in the current version when the input slope increases. This would reduce the number of samples required to resolve each pulse and would prevent samples from being wasted on irrelevant small amplitude pulses. These optimized algorithms could easily be tested on the current prototype ADC since the control logic was implemented off-chip in an FPGA.

The focus of the current work was to develop a platform to test the AR algorithm. In the next implementation, several improvements could be made to the architecture to improve the performance of the ADC. The power dissipation could be reduced by an order of magnitude by removing the static power dissipation of the OTAs in the DACs. This could be accomplished by implementing the DACs as a single resistor ladder, which would only require

enough static current to drive the parasitic capacitance of the switches connecting the DACs to the comparators as well as the input capacitance of the comparators. The DAC levels could also be calibrated to prevent their accuracy from limiting the peak SNDR. In the ideal case, the SNDR would only be limited by the resolution of the timer, which can be made very precise in modern CMOS processes. The resolution of the calibration used to remove the comparator offset could also be increased to prevent any difference in offset between the comparators from limiting the peak SNDR. An automatic calibration procedure could also be designed. A simple method could set the differential output of the DACs to zero and then use a calibration DAC to determine the trip points of the comparators. This would also allow the offset of the amplifiers in the DACs to be compensated. A background calibration method could also be developed to allow the offset to be compensated across temperature and voltage variations during operation.

A final note should be made about the power dissipation of the ADC. While achieving a very low value is important, in a complete system, the data compression offered by the asynchronous architecture will provide additional power savings beyond that of the proposed ADC. In applications where transmission power dominates, this could lead to a reduction in transmission power much higher than the power dissipation of the ADC. Also, by directly acquiring the data in compressed form, the system complexity and power consumption for data acquisition applications are reduced and the need for additional compression algorithms following the ADC is avoided. In order to

directly compare the efficiency of asynchronous data acquisition systems to conventional Nyquist rate systems, a figure of merit similar to that used for conventional ADCs [76] should be developed which takes into account the final data rate as well as the additional power consumption of any post processing required to compress the data. This would allow the total power consumption of the data acquisition process to be taken into account as well as the required transmission data rate. Since the asynchronous acquisition process is very signal dependent, more analysis should be done in the future to optimize the acquisition efficiency for particular applications and allow a fair comparison of the figure of merit with conventional solutions.

Chapter 5

Compressed Sensing

5.1 Introduction

As discussed in Chapter 1, wideband spectrum sensing is an essential function in cognitive radio and signal intelligence applications, where it may be used to monitor spectrum occupancy or determine the frequency of emitters, among other tasks. When integrated into wireless sensor networks, the sensing operation should be realized with as low complexity and power consumption as possible, in order to increase the operational lifetime and decrease the size, weight, average power, and cost of each sensor. Conventional Nyquist rate solutions often employ either a bank of tunable narrowband band-pass filters which scan across the frequency band of interest, or, as shown in Fig. 5.1, a high speed ADC to directly acquire the full spectrum [39]. The resulting data is then often compressed to reduce storage and transmission requirements. This “acquire-then-compress” acquisition process is inefficient due to the high sample rate required of the ADC and the additional post processing required to compress the sampled data.

The previous two chapters presented asynchronous sampling as a more efficient data acquisition technique for signals with sparse activity in time. By

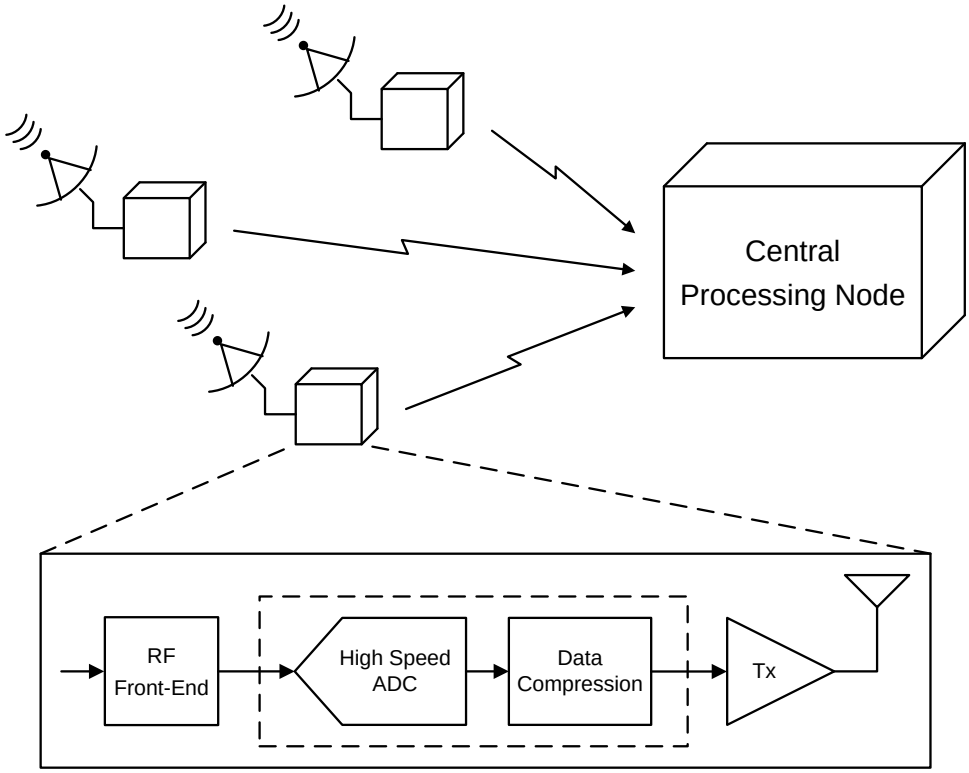


Figure 5.1: Conventional direct sampling architecture for distributed wideband spectrum sensing application.

directly acquiring the data in compressed form, the system complexity and power consumption were reduced compared to conventional Nyquist rate acquisition techniques and the need for additional compression algorithms following the ADC was avoided. In this chapter, we introduce the field of compressed sensing as another sub-Nyquist data acquisition technique better suited for wideband spectrum sensing applications where the signals of interest are often sparse in the time-frequency domain. Compressed sensing builds off the recent work of Candes, Romberg, Tao, and Donoho [27]-[31], which revealed that a signal having a sparse representation in some basis can be recovered via nonlinear optimization from a small number of linear non-adaptive measurements taken in a second basis that is incoherent with the first. In so doing, reconstruction

complexity is once again traded for more efficient data acquisition. This is advantageous for the energy constrained wireless sensor applications we are interested in, where signal analysis is performed remotely at a central processing node or other device with ample power and processing resources.

Compared to asynchronous sampling, compressed sensing has the added benefit of not being restricted to signals which are sparse in time. This allows compressed sensing to potentially benefit a broader range of applications such as software defined radio [84], wideband spectrum sensing and communications [85], and imaging applications where increasing sample and data rates make it increasingly difficult to design simple low power acquisition systems. This will also allow compressed sensing to exploit the recent advances in sparse representation of images and other signals in various wavelet domains. One area where compressed sensing has already had a tangible impact is magnetic resonance imaging [86], where the reduced number of measurements required has enabled a reduction in the scan times required to produce an image. This has obvious benefits for the patient in addition to helping reduce costs. Recent work has also investigated the use of compressed sensing to benefit such applications as: the design of a compressive single-pixel camera [87]; biomedical signal acquisition [88],[89], where many signals such as ECG and EEG have sparse wavelet representations; and compressed synthetic aperture radar [90], where the radar pulses also have sparse wavelet representations. In the rest of this and the following chapter we focus on applying compressed sensing to wideband spectrum sensing applications in order to reduce the complexity and power

consumption of the signal acquisition process and relax the requirements of the ADC.

5.2 Background Theory

In this section we give a brief overview of the theory behind compressed sensing so that the reader may gain an appreciation for how it can be exploited to design more efficient and less complex data acquisition systems. Interested readers can find more in depth analysis in [27]-[31] and a more tutorial like overview in [33],[35], and [87]. Since this dissertation is concerned with digital signal acquisition, we restrict our discussion to the discrete time theory.

5.2.1 Sparse Signals

Compressed sensing theory tells us that a length N discrete time signal $x[n]$, which is sparse or compressible in some basis Ψ, can be acquired and reconstructed by taking only $M << N$ measurements, where M is proportional to the information content in the signal. This suggests that under the condition of sparsity, one should be able to acquire data at rates much lower than what is allowed by the Nyquist-Shannon sampling theory. Signal $x \in R^N$ is said to be K-sparse in $\Psi = [\psi_1,\psi_2,\ldots,\psi_N]$ if x can be represented by a linear combination of K vectors from Ψ, where $K << N$. As shown in Fig. 5.2, by arranging the elements of $x[n]$ as an N x 1 column vector and the basis Ψ as an N x N matrix, with columns equal to the basis functions $\{\psi_i\}$, x can then be expressed as

$$x[n] = \sum_{i=1}^{N} \alpha_i \psi_i[n], \tag{5.1}$$

where α is the N x 1 vector of basis weighting coefficients. In practice few real-

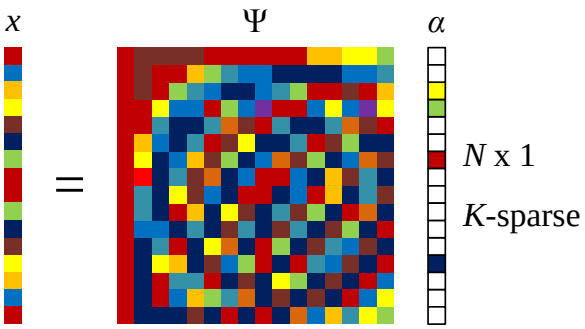


Figure 5.2: Sparse representation of input signal x with $K = 4$.

world signals are strictly sparse, but instead are compressible, in that most of the coefficients $\{\alpha_i\}$ are non-zero but small compared to relatively few large coefficients. Since most of the signal energy will be located in a small number of large coefficients, compressible signals can be well approximated by a K-sparse signal with only K non-zero coefficients. This was demonstrated in Chapter 2, where, for the examples shown, there was little loss of information when the particular images and signals used were approximated by only 10% of the coefficients in their wavelet representations.

5.2.2 Compressed Measurements

In Chapter 2, we discussed how conventional digital compression algorithms are able to compress a K-sparse or compressible signal by first acquiring the full N-sample signal, then computing the coefficients $\{\alpha_i\}$, and finally encoding the K largest coefficients. Compressed sensing provides a more efficient method of compression by directly acquiring x in compressed form by taking M linear measurements of x projected onto a set of measurement functions $\varphi_1,\varphi_2\ldots\varphi_m$, where $M < N$. This is equivalent to taking M inner products between x and a set of length N measurement vectors $\{\varphi_i\}$. As shown in Fig. 5.3, by arranging the

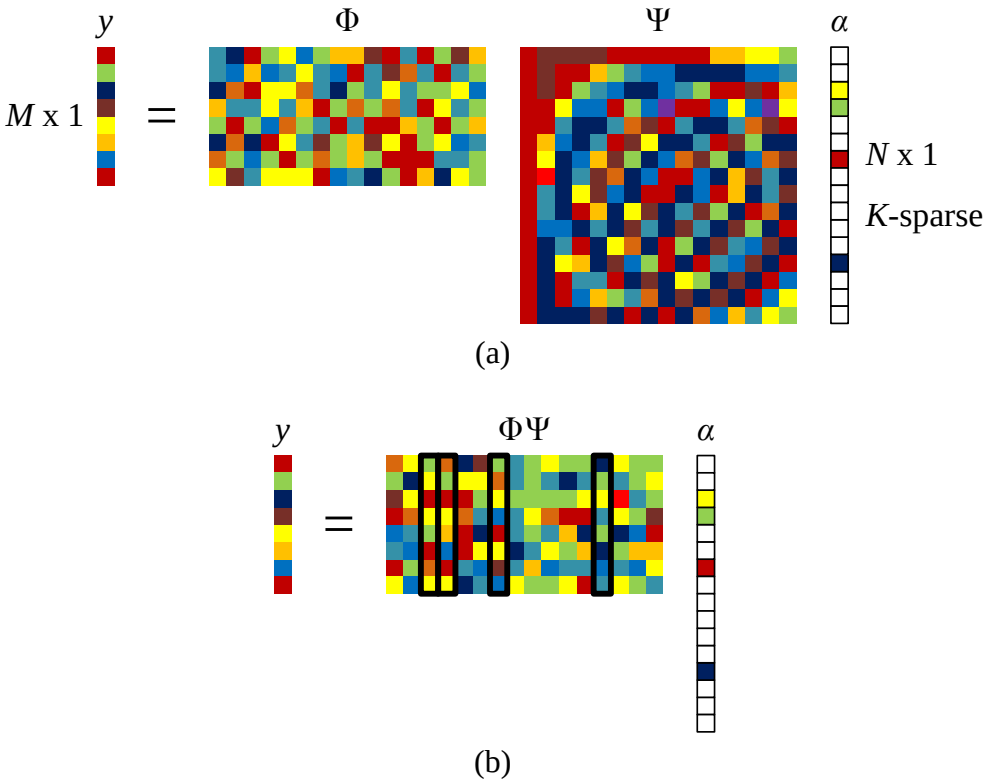


Figure 5.3: (a) Compressed sensing measurement process. The length N vector of coefficients α is sparse with $K = 4$. (b) The measurement vector y is a linear combination of K columns from matrix $\Phi\Psi$.

resulting measurements into an M x 1 vector y and the measurements vectors as rows in an M x N matrix Φ, we can write

$$y = \Phi x = \Phi \Psi \alpha \, . \tag{5.2}$$

For a K-sparse signal x the measurement vector y will be a linear combination of K columns from matrix $\Phi\Psi$.

In order to prevent information from being lost in the dimensionality reduction from $x \in R^N$ to $y \in R^M$ and ensure that x can be stably recovered from the compressed measurements, the measurement matrix Φ must satisfy certain conditions. The first is that the measurement matrix Φ and basis matrix Ψ must be incoherent with each other. This ensures that the rows of Φ cannot sparsely

represent the columns of Ψ, and vice versa. A quantitative measure of the coherence between Φ and Ψ is

$$\mu(\Phi,\Psi)=\sqrt{N}\cdot\max_{1\le i,j\le N}\left|\left\langle\phi_i,\psi_j\right\rangle\right|. \tag{5.3}$$

The level of coherence between Φ and Ψ determines the minimum number of measurements required for a given *K*, which is equal to

$$M\ge C\mu^2(\Phi,\Psi)K\log_2(N), \tag{5.4}$$

where *C* is a small constant. For maximally incoherent Φ and Ψ, this reduces to approximately

$$M\ge K\log_2(N). \tag{5.5}$$

An alternative approach to ensuring stable recovery of a *K*-sparse signal *x*, which is based on the so-called restricted isometry property (RIP), has been developed in [27]-[31]. For each $K = 1,2,\ldots,N$, the isometry constant δ_k of a matrix Φ is defined as the smallest number such that

$$(1-\delta_k)\|x\|_{\ell_2}^2\le\|\Phi x\|_{\ell_2}^2\le(1+\delta_k)\|x\|_{\ell_2}^2 \tag{5.6}$$

holds for all *K*-sparse signals *x*. Measurement matrix Φ is then said to obey the RIP of order *K* if δ_k is significantly smaller than 1. This in effect says that all subsets of *K* columns taken from Φ are nearly orthogonal, which is required to guarantee that a unique solution is found for each possible *K*-sparse signal *x*.

The previously stated conditions can be satisfied with high probability for any fixed basis Ψ by choosing Φ as a random matrix, where, for example, each of its elements are drawn from an independent and identically distributed (i.i.d.) random distribution. Most of the reported work on compressed sensing exploits

this fact to avoid the need to verify that a given Φ satisfies the RIP, which is combinatorial complex [33],[87]. This has also been shown to allow the design of universal measurement matrices capable of acquiring signals which are sparse in arbitrary basis [91].

5.2.3 Signal Recovery

Once we have acquired the M measurements $y_1, y_2 \ldots y_m$, we are left with the problem of recovering the original length N signal x, or equivalently its sparse coefficient vector α, from the compressed measurements. Since $M < N$, this problem is ill-posed in general and typical reconstruction algorithms will fail to return a unique solution. Compressed sensing overcomes this problem by taking advantage of the fact that the solution is sparse and instead solves the ℓ_1-minimization problem

$$\hat{\alpha} = \arg\min \|\alpha'\|_1 \text{ subject to } \Phi\Psi\alpha' = \mathrm{y}, \tag{5.7}$$

which solves for the sparsest set of coefficients $\{\alpha_i\}$ that match the M measurements $\{y_j\}$. This is a convex optimization problem that conveniently reduces to a linear program known as Basis Pursuit [92]. As long as the number of measurements is large enough compared to the level of sparsity in the signal, the reconstruction will succeed with high probability.

For real world signals, the measurements will contain noise and can be modeled by

$$y_n = \Phi\Psi\alpha + n, \tag{5.8}$$

where n is a zero-mean white Gaussian noise vector. To account for this, an alternative recovery algorithm called Basis Pursuit Denoising (BPDN) is used to

solve

$$\hat{\alpha} = \text{argmin} \left\| \alpha' \right\|_1 \text{ subject to } \left\| y - \Phi \Psi \alpha' \right\|_2 < \varepsilon, \tag{5.9}$$

where ε is an upper bound on the noise magnitude.

Chapter 6

Random Sampling Analog-to-Information Converter

6.1 Introduction

The previous chapter demonstrated how compressed sensing has the potential to benefit a broad range of applications such as software defined radio, wideband spectrum sensing and communications, and imaging applications where increasing sample and data rates make it increasingly difficult to design simple low power acquisition systems. In the case of wideband spectrum sensing, compressed sensing techniques can be used to directly acquire the entire spectrum in compressed form without the need for additional compression algorithms following the ADC. In practice this is done by way of an analog-to-information converter (AIC) [32],[33],[93],[94]. The resulting data can then be directly transmitted at a low rate and reconstructed remotely with close to full fidelity using ℓ_1-minimization based reconstruction algorithms [31],[95]. Compared to the conventional Nyquist rate approach shown in Fig. 5.1, the complexity and power consumption of the signal acquisition process are reduced, and the requirements of the ADC are relaxed.

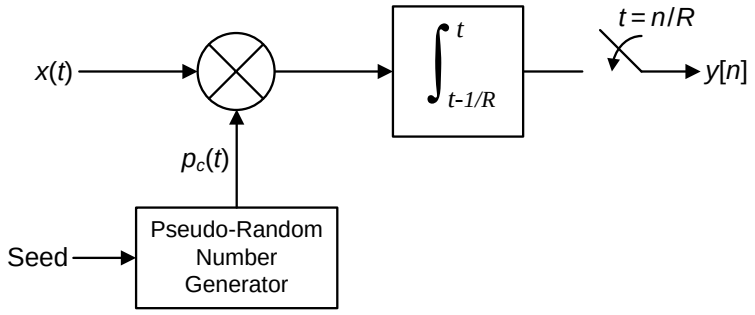


Figure 6.1: Block diagram of the random demodulator.

As the field of compressed sensing is still relatively young, physical systems which offer better performance than existing solutions have been slow to emerge. One solution that has shown promising results for sparse signals composed of multiple discrete frequency tones is the random demodulator [34],[93], which is shown in Fig. 6.1. The input signal $x(t)$ is first mixed by a pseudo-random chipping sequence of ±1s at a rate equal to at least twice the input bandwidth. The mixed output is then integrated and dumped at a constant rate R. The integrated output is sampled by a low-rate ADC resulting in the measurement sequence $y[n]$. Referring to the theory presented in the previous chapter, the corresponding measurement matrix consists of M rows of pseudo-random ±1's. The original input spectrum can then be reconstructed using the ℓ_1-minimization algorithms discussed in the previous chapter. A low-frequency proof-of-concept prototype system has been developed with off-the-shelf parts in [96]. The same authors have also proposed a simpler architecture based on randomly sampling the received signal [94], and have implemented a low-frequency proof-of-concept prototype system developed with off-the-shelf parts in [97]. We will discuss this architecture in more detail later.

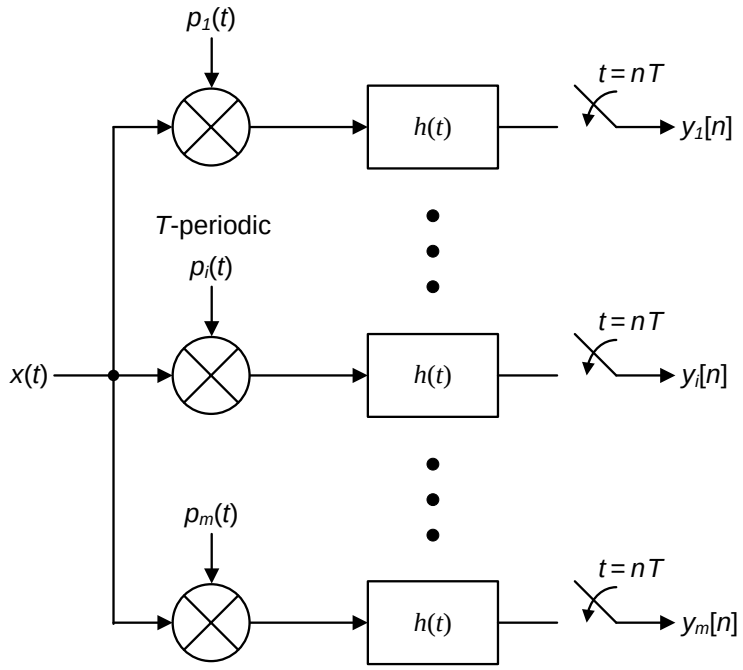


Figure 6.2: Block diagram of the modulated wideband converter.

A system similar to the random demodulator, referred to as the modulated wideband converter, has been developed in [98]-[100]. The goal of this work was to enable sub-Nyquist sampling of multiband signals, whose unknown frequency support occupies only a small portion of a wide spectrum. As shown in Fig. 6.2, the analog input signal $x(t)$ is first multiplied by a bank of periodic waveforms. This scrambles the spectrum of $x(t)$ such that the baseband frequencies that reside below the cutoff frequency of the low-pass filters, which follow the mixing operation, contain a mixture of the spectral contents from the entire Nyquist range. Each product is then low-pass filtered and sampled uniformly at a low rate, which is orders of magnitude smaller than Nyquist. A reconstruction algorithm similar to that used in standard compressed sensing systems is then used to recover the original analog input signal under certain necessary and sufficient

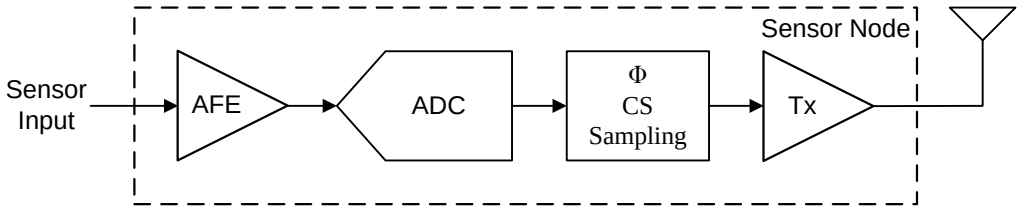


Figure 6.3: Block diagram of compressed sensing (CS) wireless bio-sensor.

conditions. The interested reader can refer to [98] for a detailed explanation of the recovery algorithm used. In [100] a proof-of-concept prototype system was implemented with off-the-shelf parts and used to demonstrate sub-Nyquist acquisition of a multi-band signal with a Nyquist rate of 2GHz and spectrum occupancy up to 120MHz.

In the previously discussed systems, compressed sensing was used to directly reduce the sample rate of the signal acquisition system. In [88], the authors propose using compressed sensing in biomedical sensor applications to compress the data after it has first been acquired at the Nyquist rate. This makes sense since the sampling frequency is rarely a limitation in these applications, but data transmission is expensive. As shown in Fig. 6.3, data compression is obtained by multiplying the acquired signal by a compressed sensing measurement matrix composed of pseudo-random sequences of ±1s. The compressed data is then transmitted and reconstructed at the receiver using ℓ_1-minimization techniques.

Several compressed sensing based imaging systems have also recently been reported. In [101], the current output of each pixel is multiplied by a pseudo-random sequence of ±1s. The resulting current of each column is then summed and sampled by an ADC. Reducing the total required samples for each image frame relaxes the requirements of the ADC. In [87], a digital micromirror device

(DMD) is used to focus the received light onto a single pixel in a pseudo-random pattern. Each compressed measurement is then just the output of the single pixel.

In the rest of this chapter, we present the design and implementation of a compressed sensing AIC with the goal of moving beyond the proof-of-concept stage and improving power efficiency and reducing complexity compared to existing solutions for wireless sparse spectrum sensing applications. We also analyze and propose solutions to issues which have not been fully addressed in previous works but arise in practice, such as mismatch between the chosen basis functions and the actual received signal. As we have throughout this dissertation, we restrict our focus to the signal acquisition process and assume that signal reconstruction will occur remotely, where power and computational requirements are of less concern. The core of the design is an ultra low power moderate rate ADC that randomly samples the received signal at sub-Nyquist rates. In order to ensure proper functionality with the random clock signal and to maximize power efficiency, a prototype edge-triggered charge-sharing SAR ADC was implemented in 90nm CMOS technology. The requirements of the ADC were relaxed by restricting the minimum spacing between the pseudo random sample times. Measurement results for wideband spectrum sensing demonstrate that our proposed compressed sensing AIC improves power efficiency and reduces complexity compared to the conventional direct sampling architecture shown in Fig. 5.1, while still maintaining high performance for signals with sparse frequency support.

In the following section we present the architecture of our proposed

compressed sensing AIC, and discuss how the original spectrum is reconstructed from the compressed data. In Section 6.3, we present the design of the prototype edge-triggered SAR ADC, followed by measurement results in Section 6.4. We then demonstrate the functionality and measure the performance of our proposed compressed sensing AIC. Before concluding the chapter, we compare the power efficiency and complexity of our proposed AIC based acquisition system with the conventional direct sampling approach for wideband spectrum sensing applications.

6.2 Random Sampling AIC Architecture

The architecture of our proposed AIC based acquisition system is shown in Fig. 6.4. The AIC repeatedly takes M random samples of the input signal which are digitized with an ultra low power ADC. The input signal is then reconstructed using a decoder based on an ℓ_1-minimization algorithm [95]. Compared to the direct sampling approach in Fig. 5.1, we effectively replace the high speed ADC with a moderate rate edge-triggered ADC and PN clock generator to provide random sampling of the input signal. This reduces the power consumption of the signal acquisition process and relaxes the requirements of the ADC. We also no longer need to compress the data from the ADC since it will be directly compressed by the AIC at a rate of N/M compared to Nyquist.

If we view the input $x(t)$ as a discrete time signal $x[n]$ with sample period $T_S = 1/(2BW)$, where BW is the input bandwidth of the system, the proposed random sampling operation can be viewed as choosing M samples at random from each

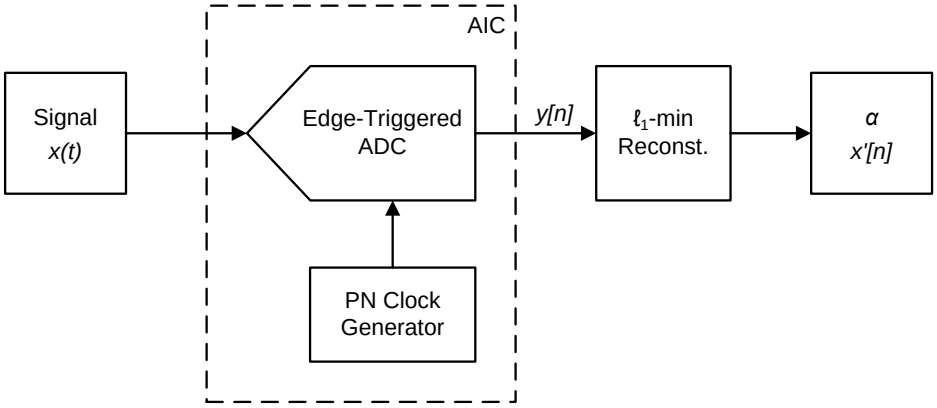


Figure 6.4: Block diagram of proposed compressed sensing AIC based acquisition system.

successive length N window of $x[n]$. The pseudo random clock signal can be generated by gating a synchronous clock signal of period T_S with a PN sequence of length N, as shown in Fig. 6.5. The PN sequence can be programmed into a shift register prior to operation. If the PN sequence is generated completely randomly, then the maximum sample rate of the ADC must be equal to at least the Nyquist rate. We can relax the ADC requirements by restricting the minimum sample spacing, as shown in Fig. 6.5. We can still achieve a reasonable degree of randomness as long as the average sample rate, which will be equal to

$$F_{s,avg} = \frac{M}{N \cdot T_s} = \frac{2BW \cdot M}{N}, \tag{6.1}$$

is at most half the maximum sample rate of the ADC.

After each set of M samples is taken, the input spectrum during the corresponding window is reconstructed by solving the ℓ_1-minimization optimization problem given by (5.9). Although one could envision implementing an FPPGA or custom digital ASIC to perform the ℓ_1-minimization, for the wireless sensor applications we are interested in, the reconstruction will occur

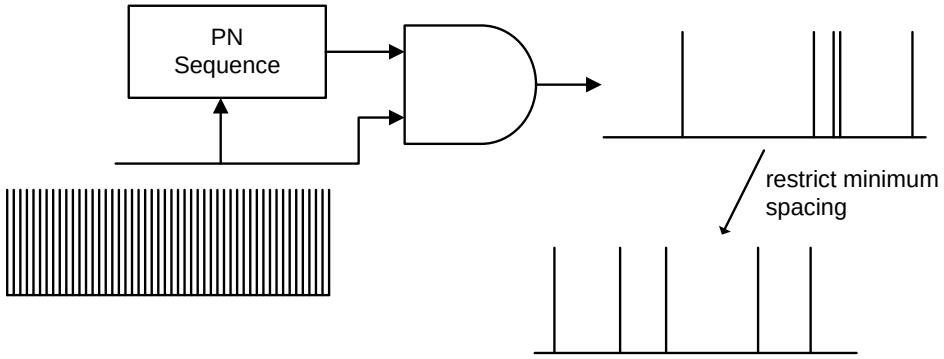


Figure 6.5: Schematic of pseudo-random clock generator. Restricting the minimum sample spacing relaxes the requirements of the ADC.

remotely where ample power and computational resources are available. Since the received signal will be sparse in the frequency domain, the basis matrix Ψ used for reconstruction is the Fourier basis

$$\Psi = \sum_{j=0}^{N-1} e^{i2\pi jn/N} \ . \tag{6.2}$$

The measurement matrix Φ is the canonical basis (delta functions)

$$\Phi = \sum_{j=0}^{M-1} \delta\left(n - k_j\right), \tag{6.3}$$

where each delta function corresponds to one of the M sample points. Since the Fourier basis is maximally incoherent with the canonical basis, the reconstruction will succeed with high probability as long as $M > K\log_2 N$ samples are taken during each input window.

6.3 Edge-Triggered Charge-Sharing SAR ADC

6.3.1 Architecture

In essence, the proposed compressed sensing AIC consists of an ADC core

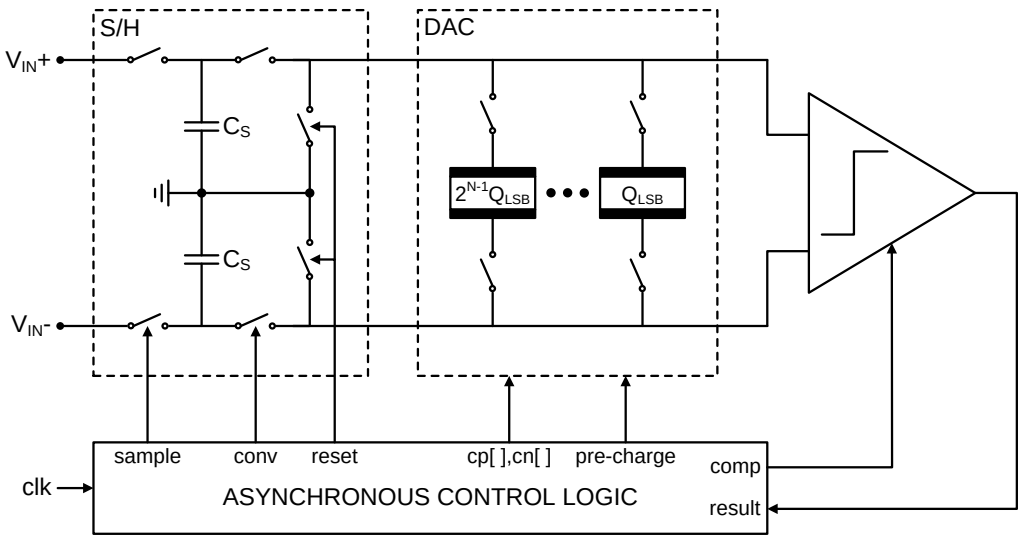


Figure 6.6: Architecture of the proposed edge-triggered charge-sharing SAR ADC.

with random sampling (and conversion) operation. In order to ensure that the ADC would function properly with the random clock signal, an edge-triggered architecture was required. To meet these requirements, we implemented the ADC with the 10-bit edge-triggered charge-sharing SAR architecture shown in Fig. 6.6 [44]. Its ultra-low power consumption that scales linearly with the sample rate allowed the power efficiency of the AIC to be maximized by taking advantage of the variable time between samples. The proposed SAR ADC uses passive charge sharing to sample the input signal and perform the successive approximation (SA) algorithm used to determine the digital output. An asynchronous controller was implemented to allow for edge-triggered operation and to eliminate the need for a high speed clock.

The operation of the ADC can be described by referring to Fig. 6.6. On the rising edge of the clock, the differential input is connected to capacitors C_S and any charge stored on the parasitic capacitance of the comparator inputs and DAC

switches is reset. A binary scaled array of capacitors is also precharged to the power supply to create a binary scaled array of charge, which is used as the reference for the SA algorithm. Since the precharging finishes before the conversion begins, the settling requirements of the reference voltage are relaxed compared to traditional charge-redistribution SAR architectures [76]. After ~30ns, the input is disconnected from the sampling capacitors, leaving a differential charge equal to $C_S V_{IN}/2$ on them. The sampling capacitors are then connected to the comparator and the SA algorithm begins.

In order to determine the most-significant bit (MSB), the comparator is initiated and the decision stored once it is valid. Based on the comparator decision, either signal $C_P[N\text{-}1]$ or $C_N[N\text{-}1]$ is set high, which connects the MSB capacitor in the DAC array across the sampling capacitors in the orientation required to converge the differential charge on them towards zero. This will add/subtract an amount of charge equal to half the input range to the sampling capacitors, resulting in a differential charge equal to $C_S V_{IN}/2 \pm C_{MSB} V_{DD}$ on them. This operation can be visualized using Fig. 6.7(a), which shows the MSB capacitor connected across the sampling capacitors, as well as the next capacitor in the DAC and its switch matrix. The remaining bits are determined by repeating the previous operations and adding or subtracting each successive binary scaled amount of charge to the sampling capacitors. An example of the control signal waveforms and the resulting voltages on the sampling capacitors is shown in Fig. 6.7(b) during the first two bits of the conversion.

In order to allow the ADC to be edge-triggered, and to avoid the need for a

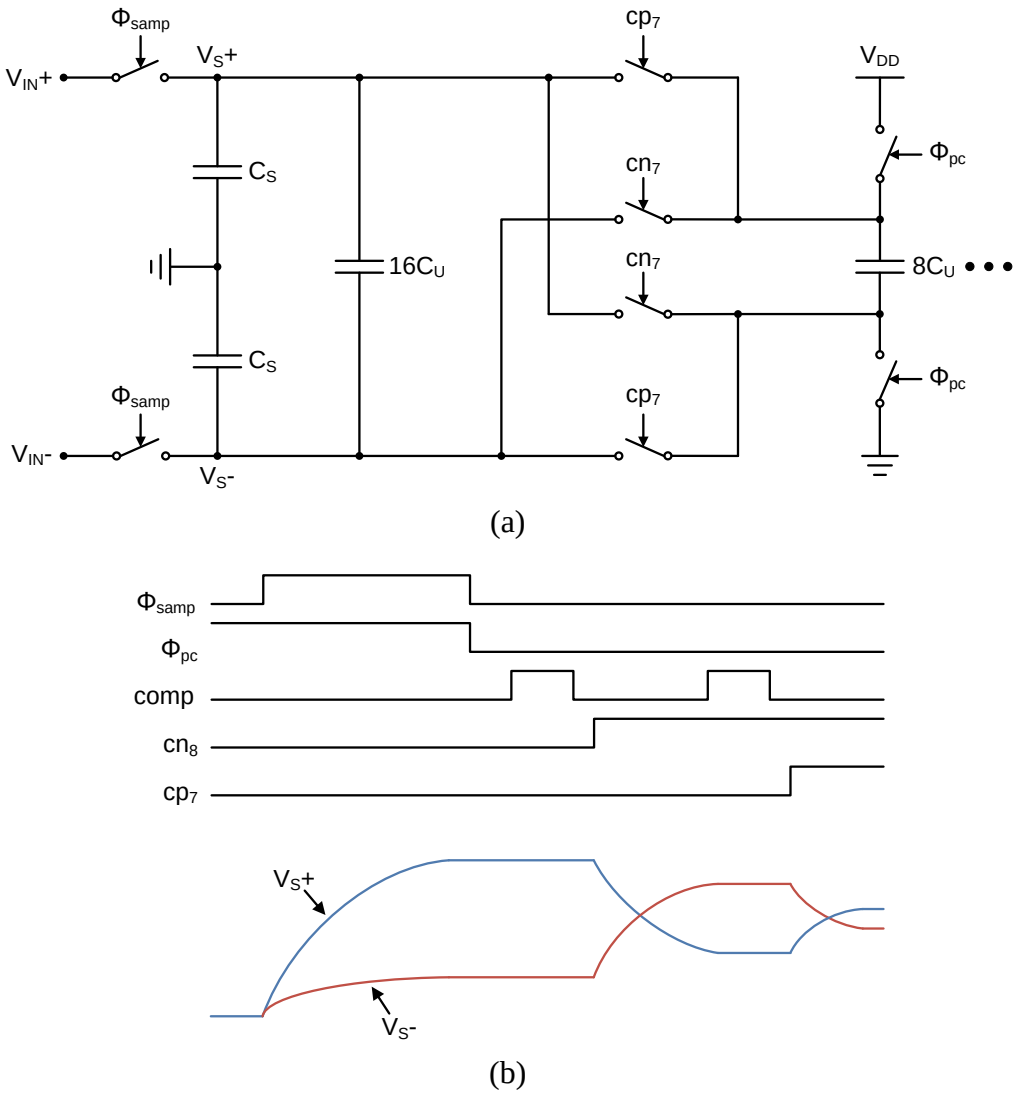


Figure 6.7: (a) Simplified schematic of the sample-and-hold capacitors and DAC during the conversion. (b) Example control signal waveforms for the first 2 bits of the conversion.

high speed clock to control the conversion process, an asynchronous controller was implemented for the control logic. On the rising edge of the clock signal, a reset pulse is generated in the controller which initiates the conversion and connects the differential input to the sampling capacitors. After waiting ~30ns, the controller then repeatedly activates the comparator, waits for the comparator output to be valid, and then adds or subtracts charge to the sampling capacitors,

until all 10 bits have been resolved.

The total conversion process takes ~100ns, after which the ADC goes idle until the next rising clock edge. The resulting maximum sample rate of 10MS/s is sufficient to allow the proposed compressed sensing AIC to acquire signals with bandwidths up to tens of MHz. This will be suitable for many spectrum sensing applications, such as monitoring the entire 26MHz wide ISM 915MHz frequency band. By keeping the conversion time independent of the sample rate, the ADC performance remains constant at low sample rates, where leakage normally degrades the performance of high rate ADCs. The power consumption of the ADC will also scale linearly with the sample rate, which allows the power efficiency to be maximized when used with a random clock, where the time between samples will vary.

6.3.2 Circuit Design

6.3.2.1 DAC

The total size of the DAC capacitor array was set to 2.56pF to allow for 10-bit matching accuracy. This, combined with the 6.4pF size of the sampling capacitors C_S, set the full-scale differential input swing to $1.6V_{DD}$. In order to avoid a unit capacitance of only 5fF, the four least significant bit (LSB) capacitors in the DAC were all designed to be the same size. This increased the size of the unit capacitor to 80fF. During precharging, the binary scaled array of charge was formed by charging each of the four LSB capacitors to only a fraction of the full supply voltage. A schematic of the DAC along with the control waveforms which perform the precharging operation is shown in Fig. 6.8.

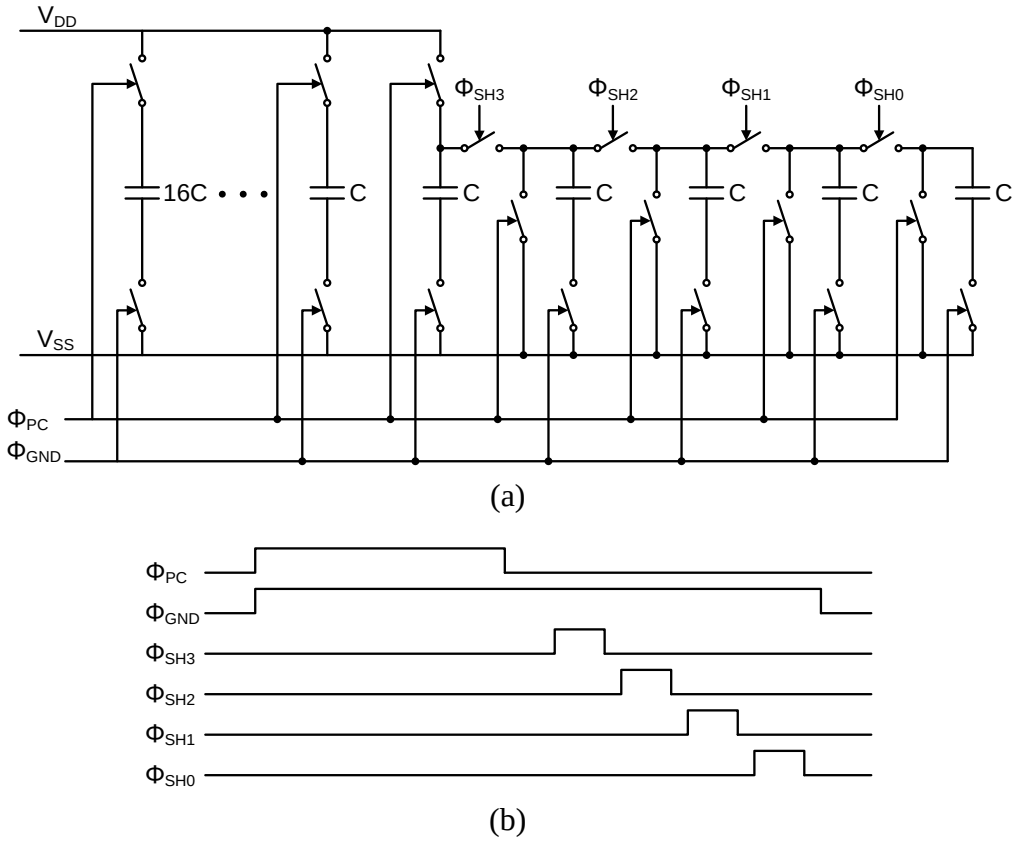


Figure 6.8: (a) Schematic of the DAC. (b) Example control signal waveform during precharging.

6.3.2.2 Comparator

The comparator, shown in Fig. 6.9, is composed of a regenerative latch preceded by a preamplifier, which reduces the input referred noise and offset of the latch stage [102]. During the reset phase, when *CLK* is low, nodes F_P and F_N are precharged to the negative supply and nodes S_P and S_N are precharged to the positive supply. The tail current source of the preamplifier is also disabled to prevent any quiescent current from flowing. On the rising clock edge, the precharge transistors are turned off and current begins to flow through the preamplifier tail current source. In order to maximize the voltage gain of the preamplifier input differential pair and the signal to noise ratio (SNR) at it output,

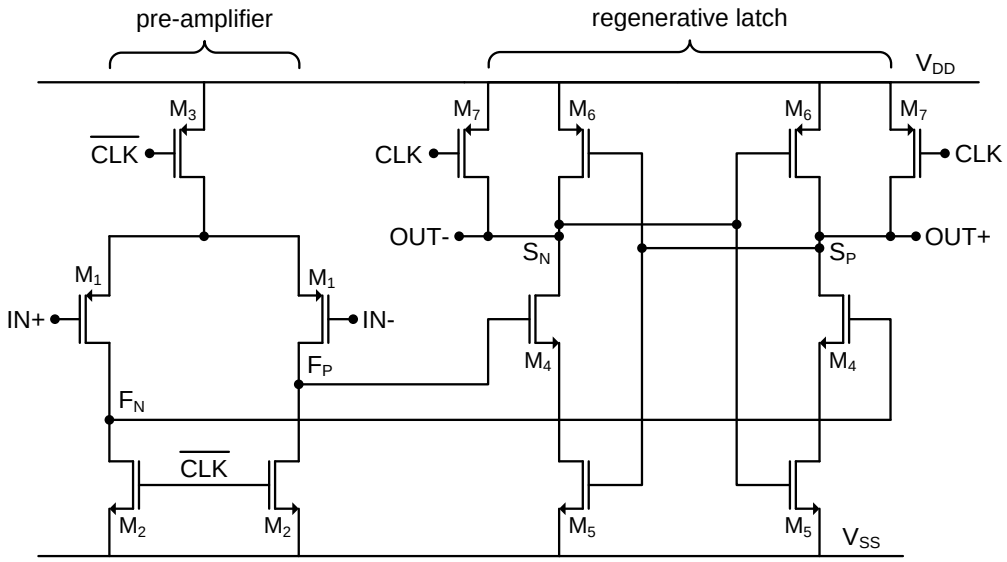


Figure 6.9: Two-stage dynamic latched comparator.

the input transistors are biased in the subthreshold region. As the common mode level of nodes F_P and F_N rises, the differential input signal is integrated onto them. When the common mode output of the preamplifier rises above the threshold voltage of the second stage input transistors, they begin to turn on, and the differential input is amplified onto nodes S_P and S_N. When the common mode level of nodes S_P and S_N decreases below a certain value, the second stage begins to regenerate. The comparison ends when the differential outputs of the second stage latch to opposite supply rails. The total comparison takes ~2ns. An XOR operation is performed on the differential output to create a *valid* signal for the asynchronous controller, which indicates that the comparison has finished. On reception of this signal, the controller immediately resets the comparator to stop any current flow in the preamplifier and then continues the conversion process.

Unlike in traditional SAR ADCs, the input referred offset voltage of the comparator directly affects the linearity of charge-sharing SAR ADCs. This is due

to the non-linear relationship between the signal, which is in the charge domain, and the offset, which is a voltage, as the total capacitance connected across the sampling capacitors varies during the conversion. The preamplifier helped reduce the input referred offset of the dynamic latch. To ensure the overall offset of the comparator was low, the preamplifier input transistors were designed with a width of 250µm. The size of the rest of the components of the comparator is given in Table 6.1. Monte Carlo simulations of the transistor level comparator design gave a 3σ variance for the input-referred offset of less than 3mV. The simulated input-referred noise and the comparison time are given in Fig. 6.10 for different common-mode input levels at a supply voltage of 1V. The input-referred noise and offset values were verified through system level MATLAB simulations to be sufficient for a resolution of 10 bits.

6.3.2.3 Sample-and-Hold

The design of the sample-and-hold circuit is shown in Fig. 6.11(a). The sampling switches were design as bootstrapped nMOS switches to provide a low signal-independent on-resistance [103]. The size of the sampling capacitors C_S was 6.4pF. As shown in Fig. 6.11(a), the sampling capacitors were split into two capacitors C_{S1} and C_{S2}. This allowed the common-mode output of the sample-and-

Table 6.1: Comparator component sizes.

Device	Size	Device	Size
M_1	250µm/0.14µm	M_5	5µm/0.14µm
M_2	2.5µm/0.14µm	M_6	15µm/0.14µm
M_3	2µm/0.14µm	M_7	7.5µm/0.14µm
M_4	5µm/0.14µm		

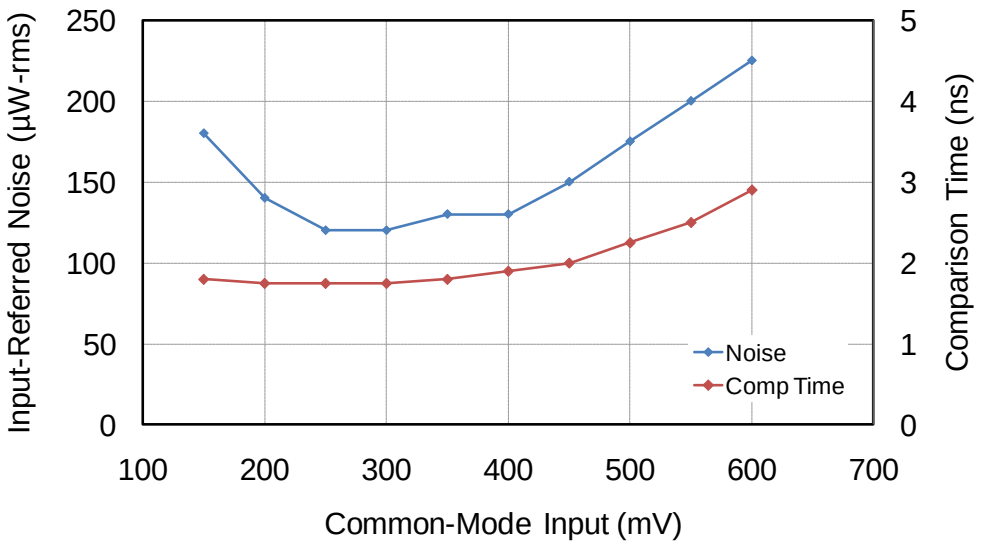
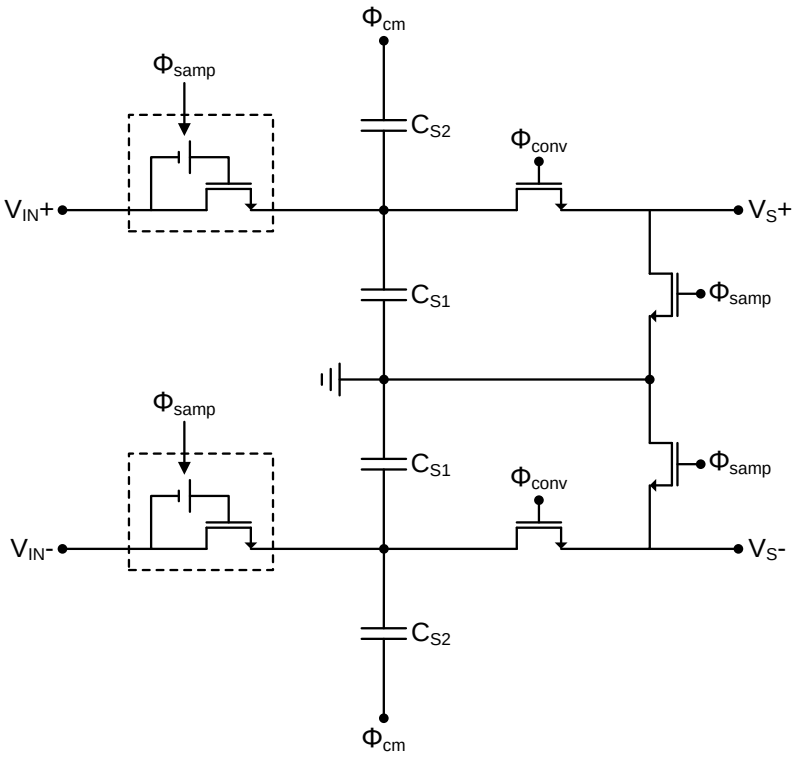


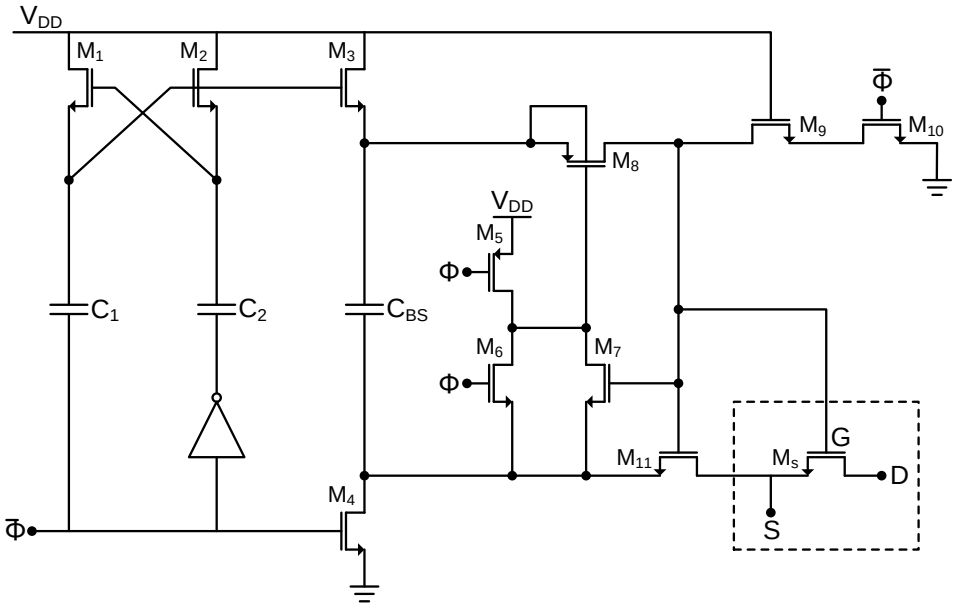
Figure 6.10: Comparator input-referred noise and comparison time versus common-mode input voltage.

hold circuit to be reduced after the first bit was resolved. This decreases the conversion time by reducing the on-resistance of the DAC switches and decreasing the comparator comparison time. By sizing C_{S1} three times the size of C_{S2}, and switching Φ_{cm} from V_{DD} to ground, the common-mode is reduced by half.

The design of the bootstrap nMOS sampling switch is shown in Fig. 6.11(b) [104]. When Φ is low, the gate of the sampling switches is grounded, and the voltage doubling circuit composed of capacitors C_1 and C_2, charges the bootstrap capacitor C_{BS} to the power supply. When Φ goes high, capacitor C_{BS} is connected between the input and the gate of the sampling switch. This creates a signal independent gate-source voltage and on-resistance and improves the linearity of the sample-and-hold circuit. The size of each of the components of the bootstrap switch is given in Table 6.2.



(a)



(b)

Figure 6.11: (a) Sample-hold circuit. (b) Bootstrap nMOS sampling switch.

Table 6.2: Bootstrap switch component sizes.

Device	Size	Device	Size
M_1	0.14µm/0.1µm	M_9	0.14µm/0.12µm
M_2	0.14µm/0.1µm	M_{10}	0.14µm/0.08µm
M_3	0.14µm/0.1µm	M_{11}	0.14µm/0.08µm
M_4	0.14µm/0.08µm	M_S	3µm/0.08µm
M_5	0.42µm/0.08µm	C_1	100fF
M_6	0.14µm/0.08µm	C_2	100fF
M_7	0.14µm/0.08µm	C_{BS}	100fF
M_8	0.42µm/0.12µm		

6.3.2.4 Asynchronous Controller

In order to allow the ADC to be edge-triggered and to avoid the need for a high speed clock to control the conversion process, an asynchronous controller was implemented for the control logic. On the rising edge of the clock, a pulse is sent through an asynchronous delay line. The length of the delay between different sections of logic sets the timing of the control signals. During the SAR conversion process, D flip-flops stall the delay line and wait for the valid signal from the comparator before continuing the conversion. By carefully designing the delay, the total conversion time can be optimized compared to when a high speed

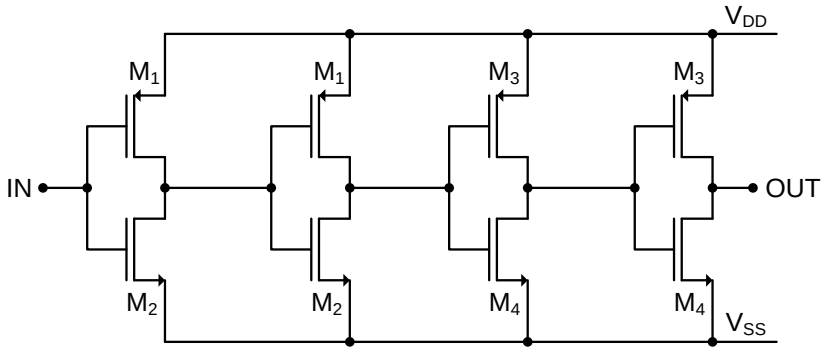


Figure 6.12: 500ps unit delay cell.

Table 6.3: Delay line unit cell component sizes.

Device	Size	Device	Size
M_1	0.4µm/0.35µm	M_3	0.56µm/0.08µm
M_2	0.2µm/0.7µm	M_4	0.14µm/0.08µm

clock is used for timing. The delay line was built with the 500ps unit cells shown in Fig. 6.12. The size of the transistors is shown in Table 6.3. The first two inverters set the delay, while the second two sharpen the output transition. The ADC was simulated across process, temperature, and voltage variations to ensure that enough delay was given under all operating conditions.

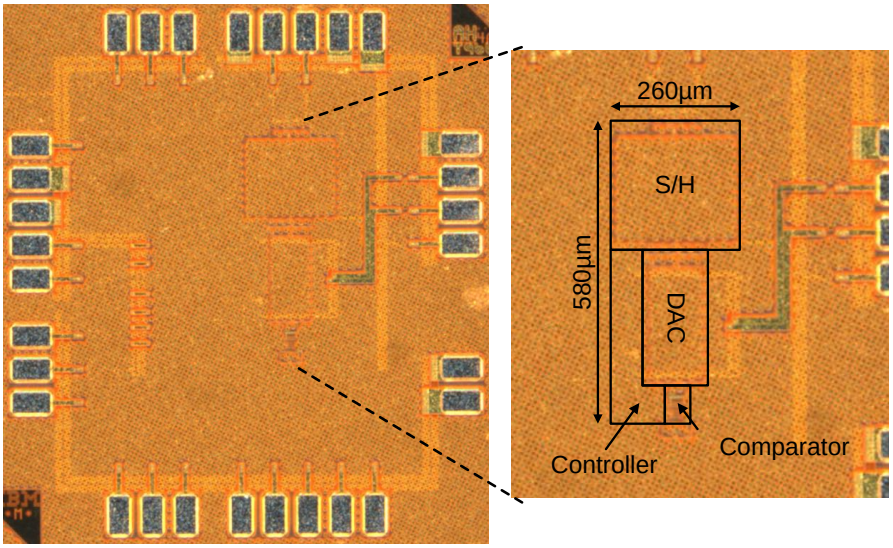


Figure 6.13: Die micrograph of the SAR ADC.

6.4 Measured Results

6.4.1 SAR ADC

A prototype of the core SAR ADC was fabricated in a 1.2V 90nm CMOS process with a single poly layer, eight metal layers, and a vertical natural capacitor (VNCAP) option. The nominal threshold voltages of the nMOS and pMOS transistors were 0.38V and -0.43V, respectively. The die micrograph is shown in Fig. 6.13. The total area of the chip is 1.3mm x 1.2mm, and the ADC core is 580µm x 260µm. The full test setup is presented in Appendix A. At a supply voltage of 1.2V, the ADC achieves a maximum sample rate of 9.5MS/s while consuming 550µW. These values decrease to 5.5MS/s and 175µW when the supply voltage is lowered to 0.9V. The power consumption scales linearly with the sample rate until it is limited by leakage currents. The figure of merit (FOM), calculated as

Table 6.4: ADC performance summary.

Parameter	Value
Technology	90nm CMOS
Package	32-pin QFN
Supply Voltage	1.2 – 0.9 V
Resolution	10 bits
Maximum Sample Rate	9.5 MS/s
Peak SNDR	57.6 dBFS
ENOB	9.3 bits
Differential Input Swing	$1.6 \bullet V_{DD}$
Full-Power Bandwidth	100 MHz
Power Consumption	550 µW @ 1.2V - 9.5MS/s
Figure of Merit	55 fJ/conversion-step @ 0.9V
Die Area	$0.15mm^2$

$$FOM = \frac{Power}{2^{ENOB} \cdot F_S}, \tag{6.4}$$

is 92fJ/conversion-step at a supply voltage of 1.2V, and improves to 55fJ/conversion step at a supply voltage of 0.9V. An overview of the performance results of the prototype ADC core for Nyquist operation is given in Table 6.4.

The static performance of the ADC core was characterized through integral nonlinearity (INL) and differential nonlinearity (DNL) measurements. The measured INL and DNL, at a supply voltage of 1.2V and sample rate of 9.5MS/s, are shown in Fig. 6.14. The output spectrum for a full-scale 100kHz sinusoidal input, at a supply voltage of 1.2V and sample rate of 9.5MS/s, is shown in Fig. 6.15(a). The signal-to-noise and distortion ratio (SNDR) and effective number of bits (ENOB) are 57.5dB and 9.3 bits, respectively. When the input frequency is increased close to the Nyquist frequency, as shown in Fig. 6.15(b), the spectrum

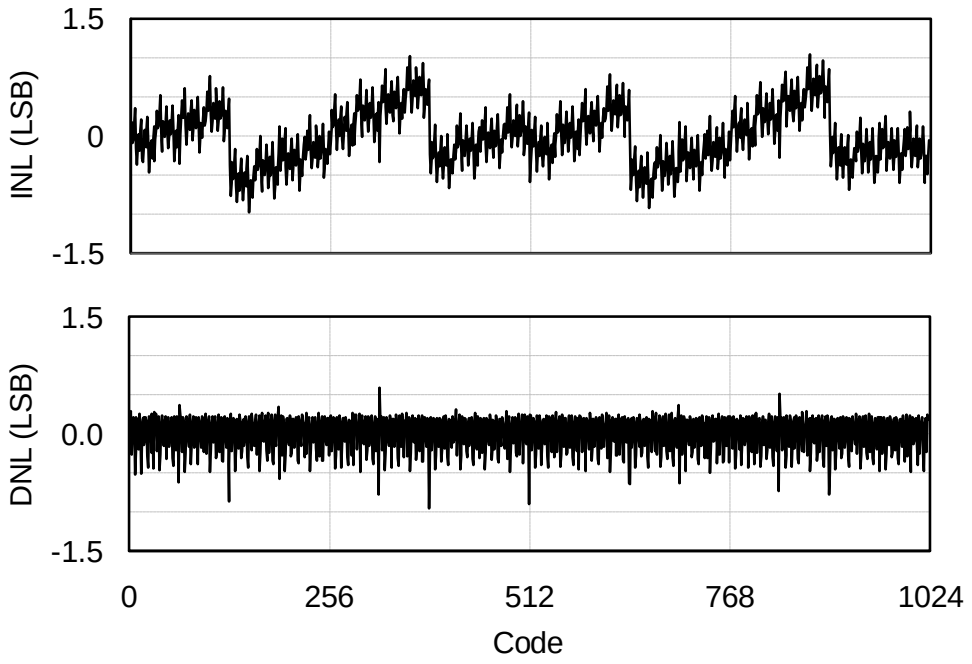
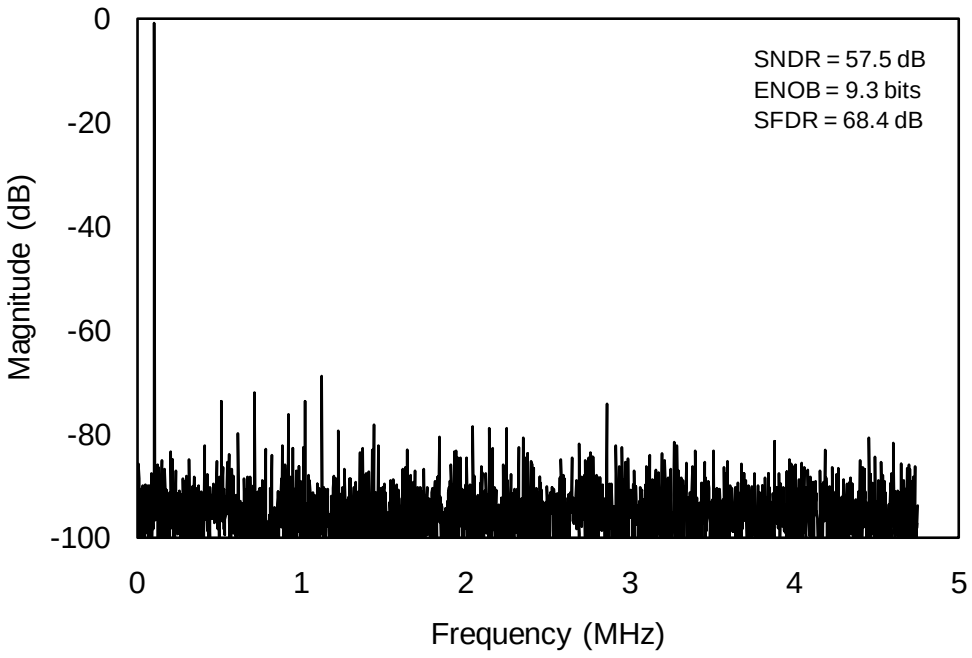


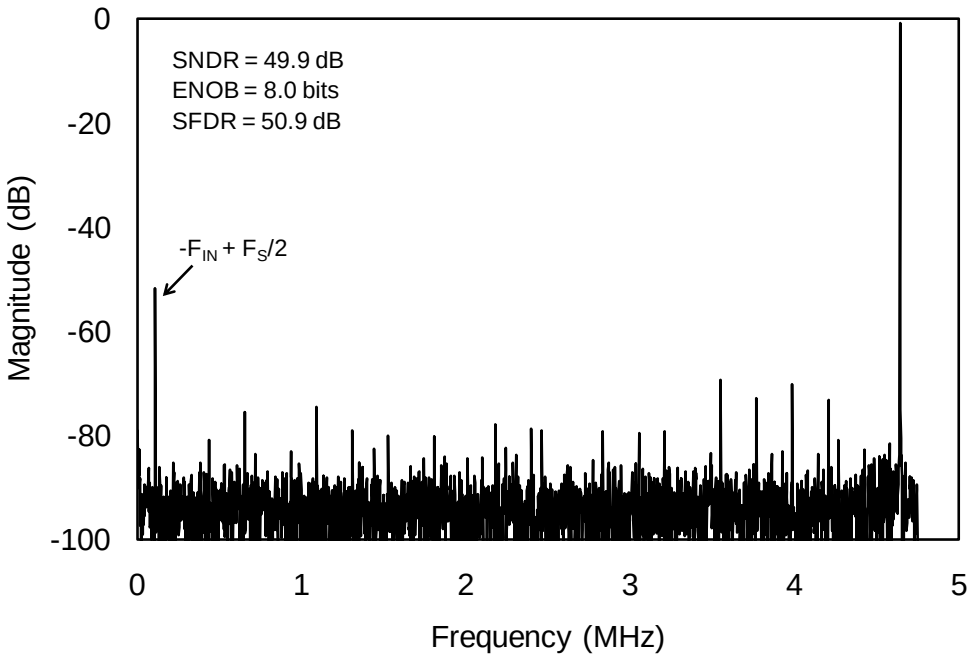
Figure 6.14: Measured INL and DNL at V_{DD} = 1.2V and F_S = 9.5MS/s.

is similar to Fig. 6.15(a), except for a spurious tone at $-F_{IN} + F_S/2$, which degrades the SNDR by 7.6 dB. This spurious tone is due to deterministic jitter on the sample time caused by an improperly designed delay line in the controller. On successive conversions, a digital '1' and then a '0' is passed through part of the delay line, creating a mismatch in the sample times similar to that seen in time-interleaved ADCs. This can easily be fixed in a redesign. For now, the deterministic jitter is removed by discarding every other sample. As shown in Fig. 6.16, this allows the ADC to achieve an ENOB greater than 9 bits for input frequencies up to the Nyquist frequency.

The performance of the prototype ADC was also measured for input frequencies above the Nyquist frequency, in order to demonstrate its suitability for the proposed compressed sensing AIC. The full-power bandwidth of the ADC was determined by measuring the output power in the fundamental as the



(a)



(b)

Figure 6.15: Measured ADC output spectrum at 1.2V and FS = 9.5MS/s for a full-scale (a) 100kHz and (b) near Nyquist input.

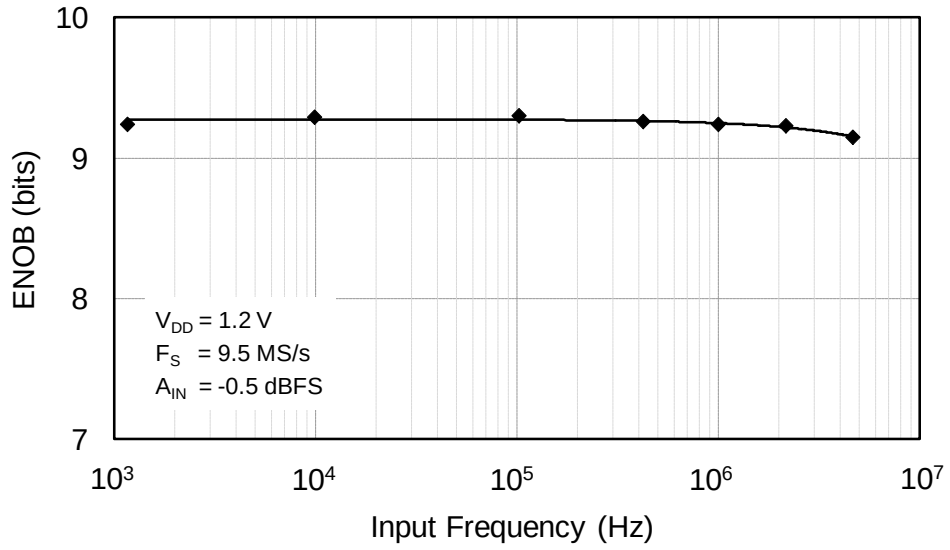


Figure 6.16: Measured ENOB versus input frequency at V_{DD} = 1.2V, F_S = 9.5MS/s, and A_{IN} = -0.5dBFS.

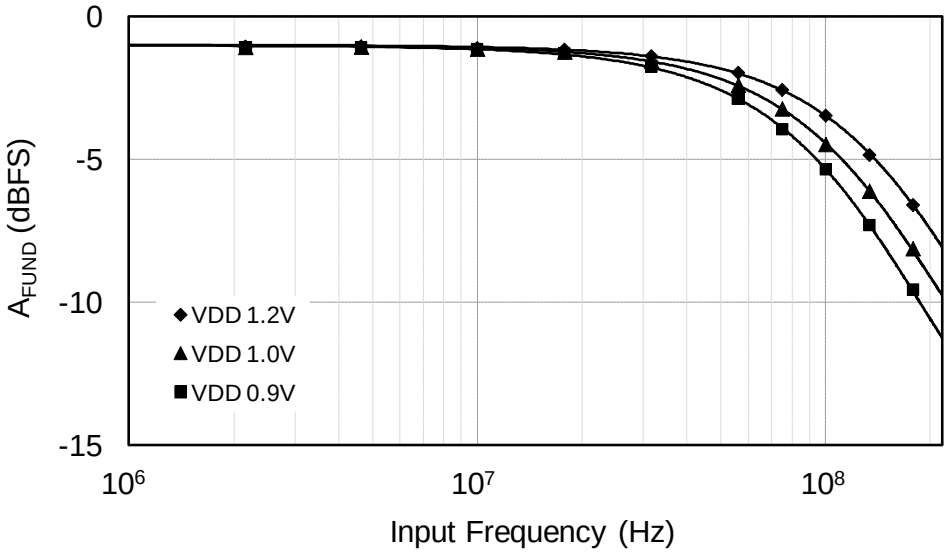


Figure 6.17: Full-power bandwidth measurements.

frequency of a sinusoidal input, with amplitude equal to -1dBFS, was increased. The results, shown in Fig. 6.17, indicate a full-power bandwidth greater than 100MHz at a supply voltage of 1.2V. Fig. 6.18 shows the measured ENOB, normalized to a full-scale input, for input frequencies up to 100MHz. For a full-scale input, the ENOB remains greater than 8 bits for input frequencies up to

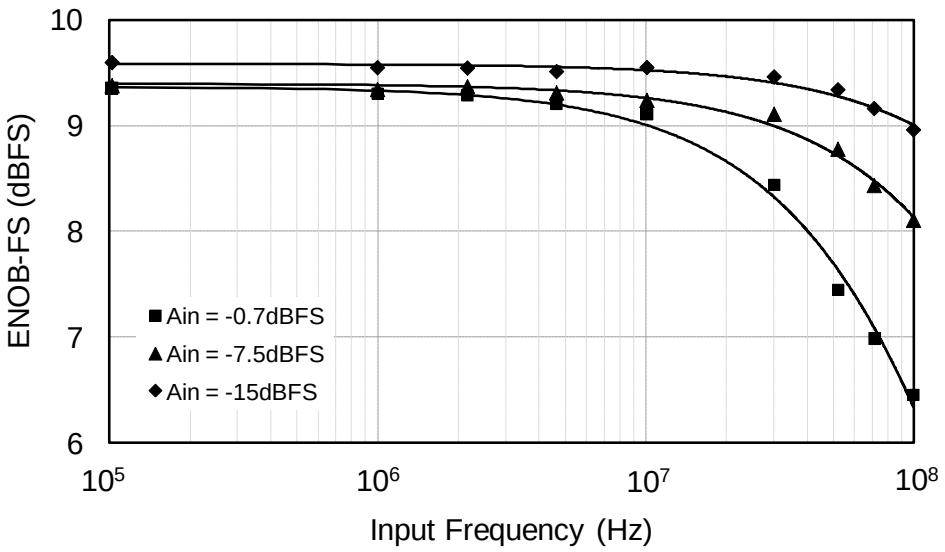


Figure 6.18: Measured ENOB, normalized to a full-scale input, versus input frequency at V_{DD} = 1.2V and F_S = 9.5MS/s.

40MHz. When the input amplitude is backed off by 7.5 and 15 dB, the normalized ENOB remains above 8 and 9 bits, respectively, for input frequencies up to 100MHz.

6.4.2 Compressed Sensing AIC

The compressed sensing AIC was realized by connecting the SAR ADC core to a random clock generator with an average sample rate of 5MS/s, as explained in Section 6.2 and shown in Fig. 6.4. To demonstrate the functionality of the compressed sensing AIC for wideband spectrum sensing applications, we provided an input signal consisting of five discrete frequency tones spread across a bandwidth of 25MHz. This represents an order of magnitude increase above the maximum input bandwidth that a Nyquist rate sampling system could handle at an equivalent overall sample rate. The resulting digital output of the ADC core was then fed to a reconstruction algorithm to recover the input spectrum from the compressed data. Since we have assumed that signal reconstruction will occur

remotely for the wireless sensor applications we are interested in, we performed the reconstruction in MATLAB using the ℓ_1-minimization convex optimization program from [95]. For testing purposes, we used two Agilent 33250A arbitrary waveform generators to generate the input and pseudo random clock signals. The full test setup is presented in Appendix A. Setting N to 1024 resulted in 102 samples per input window, which represents a compression ratio of 10 compared to Nyquist rate sampling. As shown in Fig. 6.19, the reconstructed signal and spectrum from both MATLAB simulations and measurements match the original input, except for a phase shift from our test setup.

We quantify the performance of the acquisition system by measuring the average SNDR of the reconstructed spectrum for varying signal sparsity and input SNR. Since the input signal was composed of K discrete frequency tones, the reconstructed SNDR was calculated as the ratio of the total power at these K frequencies, to the total power in the rest of the reconstructed spectrum, excluding DC. The results were averaged over 250 simulated iterations and 10 measured iterations, with a new input signal and pseudo random clock sequence generated for each iteration. As shown in Fig. 6.20(a), the performance remains high until the number of input tones reaches about 8, which is close to the theoretical limit of 10. The reconstructed SNDR even exceeds the input SNR for low values of K, which is due to the reconstruction algorithm rejecting some of the noise in the input spectrum.

Since the reconstruction algorithm has a finite probability of failure, which is dependent on the number of measurements and signal sparsity, we also measured

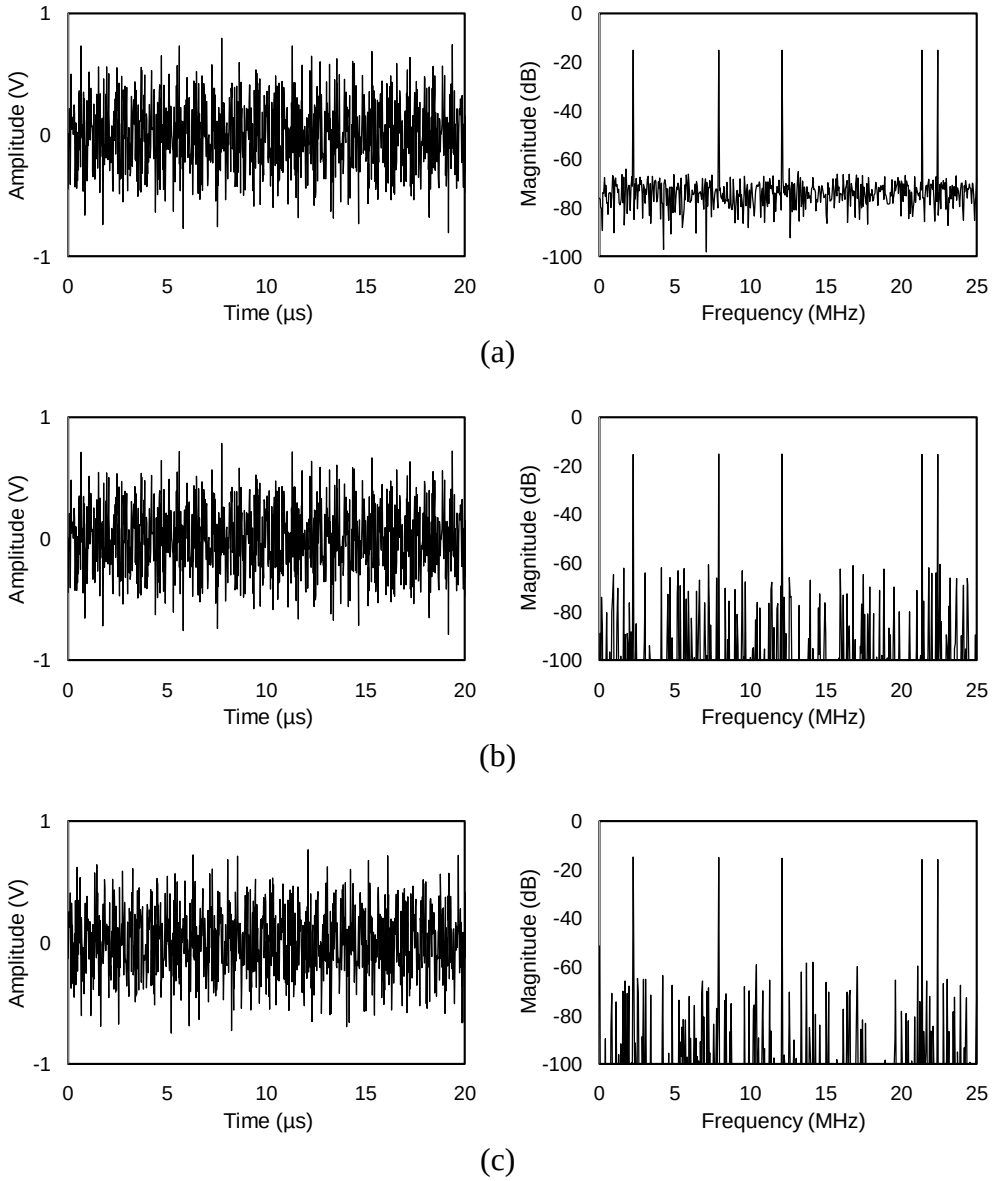
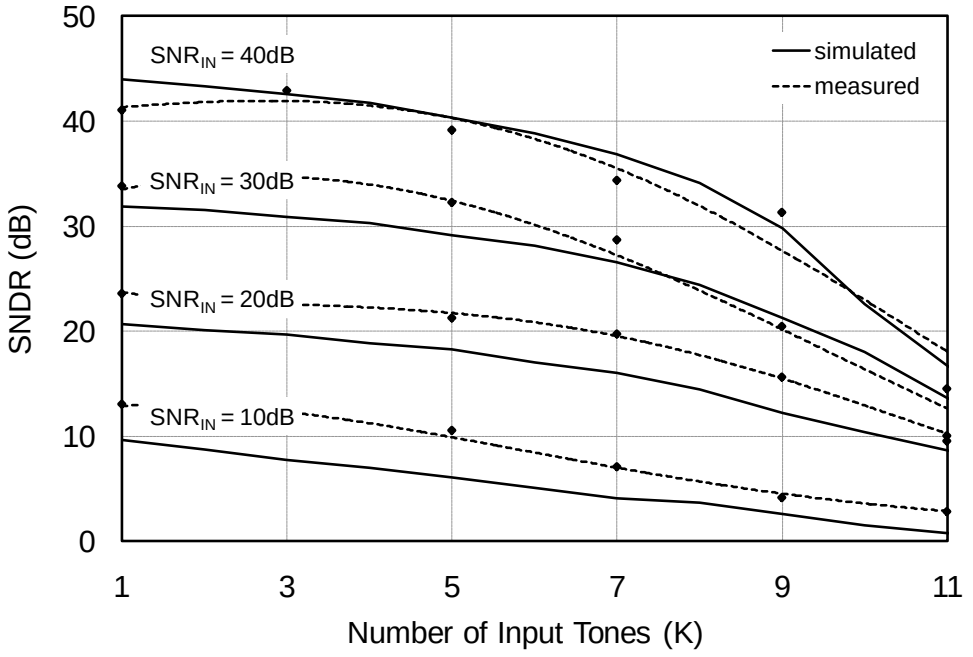
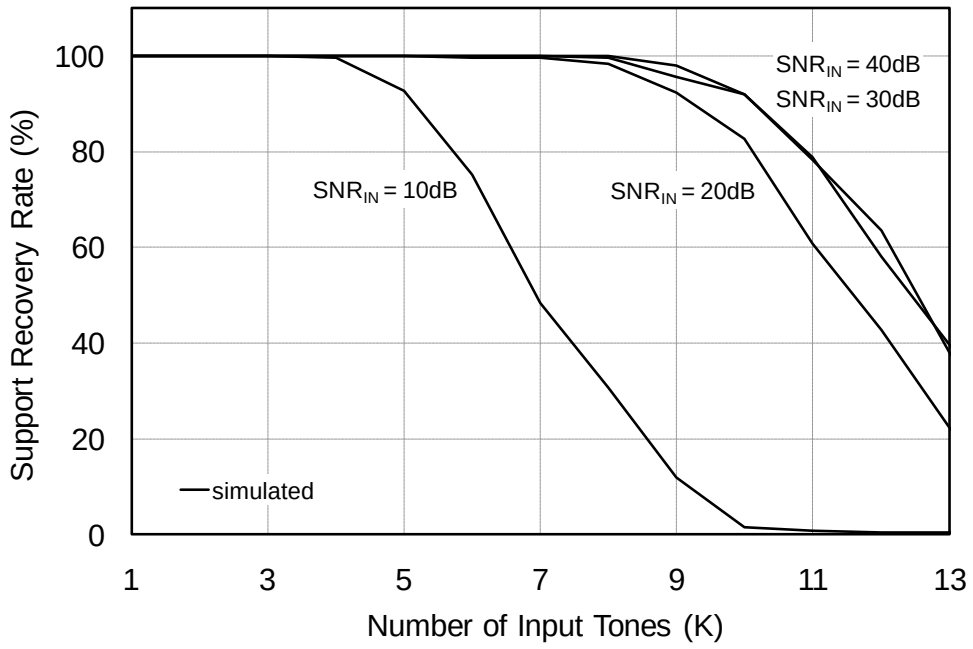


Figure 6.19: Example signals from demonstration of proposed compressed sensing ADC based acquisition system. (a) Input signal and spectrum. (b) Reconstructed signal and spectrum from MATLAB simulations. (c) Reconstructed signal and spectrum from measurements.

the probability of recovering the correct frequency support. For each test iteration, the frequency support was counted as being successfully recovered if the *K* largest frequencies in the reconstructed spectrum matched the *K* input tones. As shown in Fig. 20(b), for high input SNR, the correct frequency support was recovered every



(a)



(b)

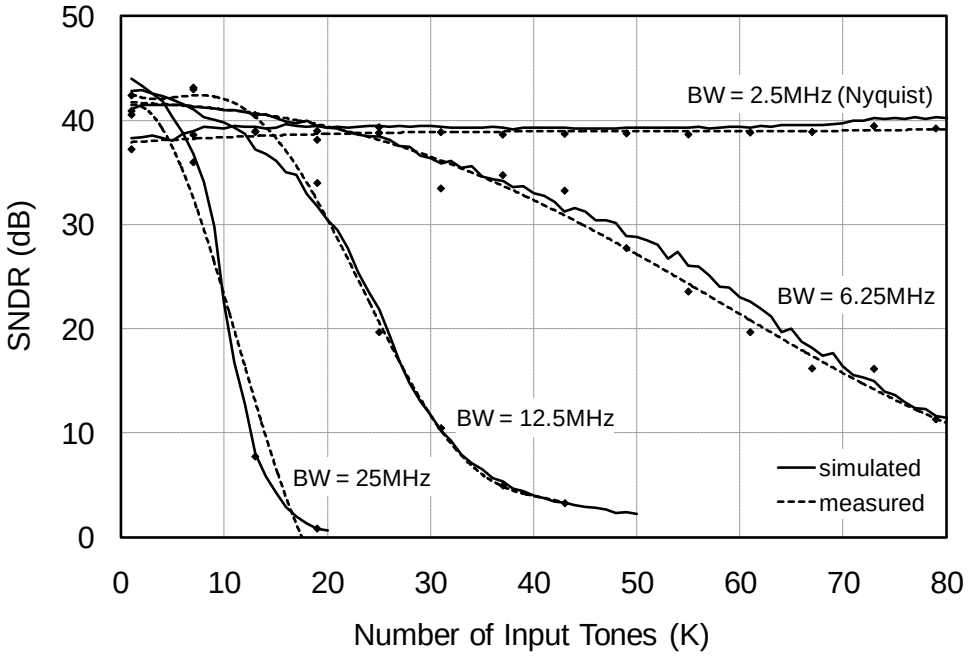
Figure 6.20: (a) Average reconstructed SNDR and (b) frequency support recovery rate versus signal sparsity (K), for varying input SNR.

time for values of K up to 8. As the input SNR is decreased, the performance remains fairly constant until the input SNR decreases to around 10dB, at which point the increased noise level makes the input signal less sparse.

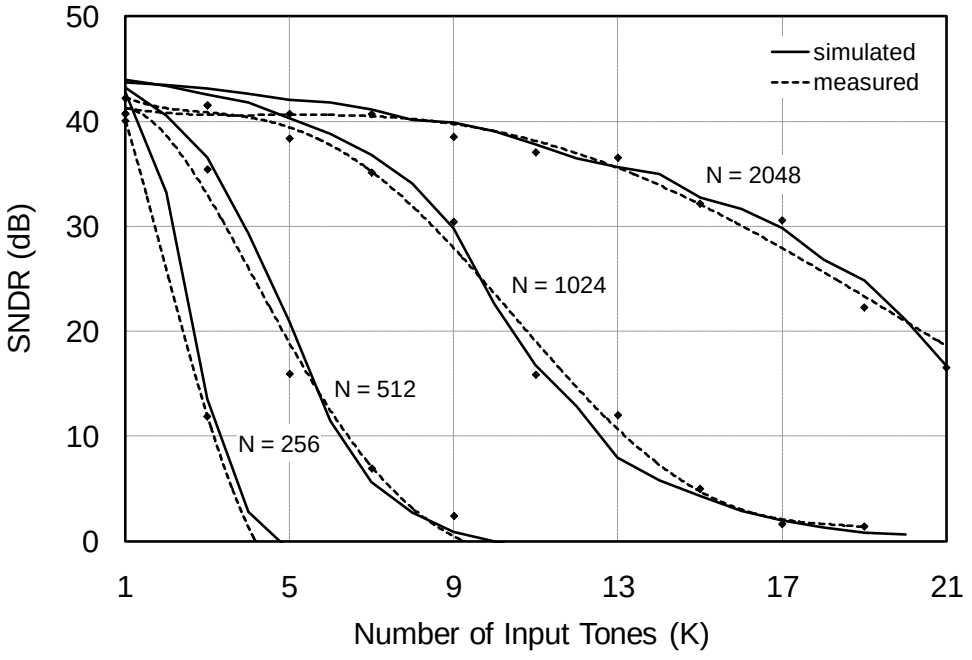
According to compressed sensing theory, we can increase the reconstruction performance by taking more samples during each input window. If we fix the average sample rate to 5MS/s, then, according to (6.1), we can increase the number of samples (M) by either decreasing the input bandwidth or increasing N. We can then repeat the previous tests for different input bandwidths and values of N. The results shown in Figs. 6.21, for an input SNR of 40dB, confirm that by decreasing the input bandwidth or increasing N, the reconstruction performance is improved and input signals consisting of more tones are able to be acquired and faithfully reconstructed. The results in Fig. 6.21(a) also show that as long as the signal is sparse enough, the performance of the AIC based acquisition system will be comparable to Nyquist sampling, despite the fewer number of samples taken.

6.4.3 Reconstruction Performance with Basis Mismatch

As the previous results demonstrate, the ℓ_1-minimization based reconstruction algorithm relies on the input signal being sparse in the discrete Fourier domain. In the previous section, we ensured that this was the case by restricting the frequency of each input tone to the $N/2$ frequencies of our Fourier basis functions. If the frequency of each tone was instead allowed to take on any value between the basis functions, spectral leakage would occur in the discrete Fourier representation of the input signal. This would decrease the sparsity of the input signal and degrade the reconstruction performance [105]-[107]. One way to



(a)



(b)

Figure 6.21: Average reconstructed SNDR versus signal sparsity, for varying (a) input bandwidth and (b) N.

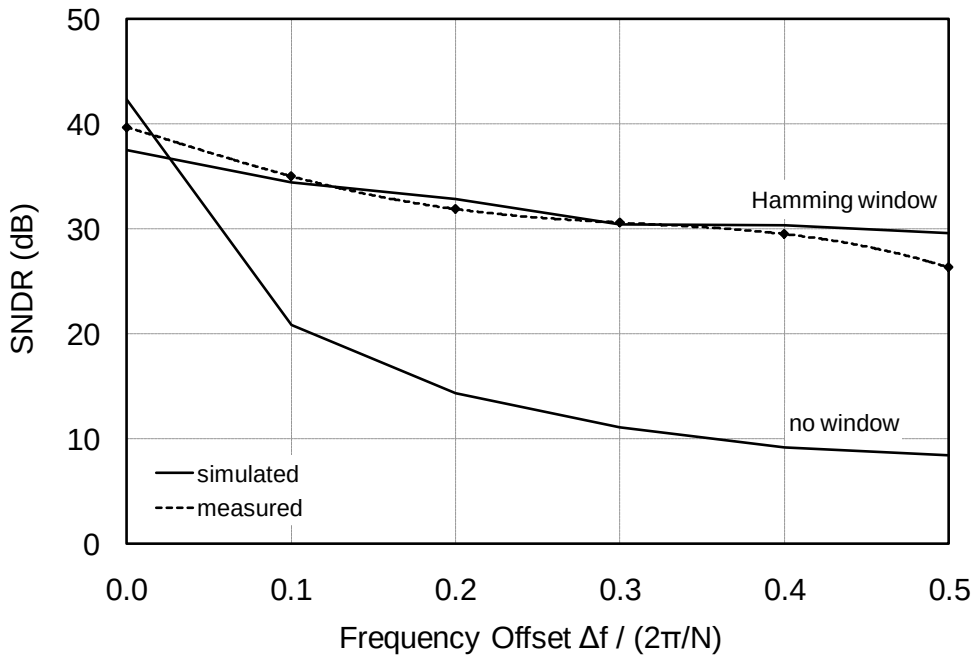


Figure 6.22: Average reconstructed SNDR versus normalized frequency mismatch, with and without sampled data multiplied by a window function.

combat the spectral leakage is to apply a tapered window function to the sampled data before reconstructing the spectrum.

If we remove the frequency restriction on the input tones, then the frequency mismatch (Δf) of each tone, measured as the distance in frequency to the closest basis function, will be between 0 and $2BW/N$. We measure how this will affect the reconstruction performance by repeating the tests from the last section for an input consisting of four discrete frequency tones, all with equal frequency mismatch. The results, for an input bandwidth of 12.5MHz, N equal to 1024, and an input SNR of 40dB, are shown in Fig. 6.22 for varying frequency mismatch. The reconstruction performance is seen to quickly degrade as the frequency mismatch is increased. We next show how using a window function can help avoid this decrease in performance by repeating the same test but now multiplying

the sampled data by a Hamming window before reconstructing the spectrum. As shown in Fig. 6.22, the reconstruction performance is significantly improved.

6.4.4 Figure of Merit

We can compare the power efficiency of our proposed compressed sensing AIC based acquisition system to the conventional direct sampling method shown in Fig. 5.1 by using an effective FOM of

$$FOM = \frac{Power}{2^{ENOB} \cdot 2BW}, \tag{6.5}$$

which takes into account that the input bandwidth may be greater than the Nyquist frequency. For an input bandwidth of 25MHz and an average sample rate of 5MS/s, our proposed AIC achieves an effective FOM of 10.2fJ/conversion-step. Compared to the Nyquist rate charge-sharing SAR ADC in [44], which achieves a FOM of 65fJ/conversion-step when sampling at the Nyquist rate of 50MHz, we improve the power efficiency of the signal acquisition process by over 6 times. This demonstrates the significant advantage of using the compressed sensing paradigm for signal acquisition compared to traditional Nyquist rate sampling systems.

6.5 Chapter Summary

This chapter has presented the design of a compressed sensing based analog-to-information converter (AIC). The core of the AIC is an ultra low-power edge-triggered SAR ADC which was implemented in 90nm CMOS technology. Measured results of the AIC demonstrated its ability to acquire and reconstruct wideband signals with sparse frequency support at sub-Nyquist rates. We also

analyzed and proposed solutions to issues which have not been fully addressed in previous works but arise in practice, such as mismatch between the chosen basis functions and the actual received signal. The ultimate goal of our work was to improve the power efficiency and reduce the complexity of the signal acquisition process for wireless spectrum sensing applications. By using an effective FOM which takes into account the increased input bandwidth of the AIC compared to Nyquist, we demonstrated a 6 times increase in power efficiency compared to conventional Nyquist rate direct sampling methods. Since the proposed AIC directly acquires the received signal in compressed form, additional compression following the ADC is also no longer needed. This, along with the relaxed requirements of the ADC, reduces the complexity of the signal acquisition process.

Chapter 7

Conclusions and Future Research

This dissertation has built on the recent theoretical and experimental work on asynchronous sampling and compressed sensing. Our goal was to exploit the advances in the theory to design practical data acquisition systems capable of directly acquiring sparse signals at sub-Nyquist rates. We focused specifically on increasing the power efficiency and decreasing the complexity of the signal acquisition process compared to existing conventional Nyquist rate solutions for biomedical sensor and wideband spectrum sensing applications.

The first half of this dissertation presented the design and implementation of an asynchronous ADC which achieves data compression for sparse and burst like signals by the inherent signal dependent sampling rate of the asynchronous architecture. The ADC was fabricated in a 0.18μm CMOS process and optimized for subthreshold operation in order to increase the power efficiency for low-frequency biomedical sensor applications. The main contribution of this work was the implementation of an AR algorithm which varied the quantizer resolution of the ADC with the slope of the input signal, in order to overcome the tradeoff between dynamic range and input bandwidth typically seen in asynchronous ADCs. This allowed the maximum possible input bandwidth to be achieved

regardless of the dynamic range requirement. By reducing the quantizer resolution during periods of high input slope, further data compression was also achieved.

The presented AR algorithm was developed to improve performance for general low frequency sensor applications. Future work will optimize the algorithm for particular applications, in order to achieve maximum data compression. For example, in a simple heart rate monitor, where only relatively large amplitude pulses need to be resolved, the AR algorithm could be programmed to decrease the quantizer resolution more quickly than in the current version when the input slope increases. This would reduce the number of samples required to resolve each pulse and would prevent samples from being wasted on irrelevant small amplitude pulses. These optimized algorithms could easily be tested on the current prototype ADC since the control logic was implemented off-chip in an FPGA.

The focus of the current work was to develop a platform to test the AR algorithm. In the next implementation, several improvements could be made to the architecture to improve the performance of the ADC. The power dissipation could be reduced by an order of magnitude by removing the static power dissipation of the OTAs in the DACs. This could be accomplished by implementing the DACs as a single resistor ladder, which would only require enough static current to drive the parasitic capacitance of the switches connecting the DACs to the comparators as well as the input capacitance of the comparators. The DAC levels could also be calibrated to prevent their accuracy from limiting the peak SNDR. In the ideal case, the SNDR would only be limited by the

resolution of the timer, which can be made very precise in modern CMOS processes. The resolution of the calibration used to remove the comparator offset could also be increased to prevent any difference in offset between the comparators from limiting the peak SNDR.

The second half of this dissertation presented the design and implementation of a compressed sensing based analog-to-information converter (AIC) for wideband spectrum sensing applications. The core of the design was an ultra low power moderate rate ADC that randomly samples the received signal at sub-Nyquist rates. In order to ensure proper functionality with the random clock signal and to maximize power efficiency, a prototype edge-triggered charge-sharing SAR ADC was implemented in 90nm CMOS technology. Measured results of the AIC demonstrated its ability to acquire and reconstruct wideband signals with sparse frequency support at sub-Nyquist rates. We also analyzed and proposed solutions to issues which have not been fully addressed in previous works but arise in practice, such as mismatch between the chosen basis functions and the actual received signal.

The ultimate goal of our work was to improve the power efficiency and reduce the complexity of the signal acquisition process for wireless spectrum sensing applications. By using an effective FOM which takes into account the increased input bandwidth of the AIC compared to Nyquist, we demonstrated a 6 times increase in power efficiency compared to conventional Nyquist rate direct sampling methods for spectrum sensing applications. Since the proposed AIC directly acquires the received signal in compressed form, additional compression

following the ADC is also no longer needed. This, along with the relaxed requirements of the ADC, reduces the complexity of the signal acquisition process.

The main contribution of this work was the implementation of a physical compressed sensing acquisition system which moved beyond the proof-of-concept stage and improved power efficiency and reduced complexity compared to existing solutions for wireless sparse spectrum sensing applications. Future work will investigate the performance of the system for more complex communication signals typically seen in practice. It will also be necessary to investigate how to integrate compressed sensing based acquisition systems as parts of larger systems. This would include investigating whether the probabilistic nature of the reconstruction algorithm will affect overall system performance. One solution to overcoming the finite probability of reconstruction failure is to cycle the sampling sequence through several sequences to reduce the probability of the reconstruction failing all the time for certain signals. Multiple sensors could also be used to add diversity and ensure that at least one sensor successfully reconstructs the signal at all times.

Appendix A

Test Setups

A.1 Asynchronous ADC

The test setup used for the FFT tests presented in Section 4.6 is shown in Fig. A.1. The sinusoidal test signals were generated by an Agilent 33250A function generator. An Analog Devices AD8138 differential ADC driver converted the single-ended input to a differential signal for the ADC. The ADC digital control logic was implemented off-chip in an Altera Cyclone II 2C35 FPGA embedded on an Altera DE2 development board. For each test, the output of the ADC (the time-amplitude samples output from the controller) was stored in memory on the FPGA development board. The output data was then readout off-line into MATLAB for further processing. A 6^{th}-order polynomial interpolator was then used to reconstruct a synchronous signal from the asynchronous samples. The sample rate of the interpolator was set to 19.3 times the input frequency. Standard FFT analysis was then performed on the resulting synchronous data to calculate the SNDR and ENOB of the sampled data.

The test setup for the accelerometer test presented in Section 4.6.3 is shown in Fig. A.2. The function generator from the previous setup was effectively replaced

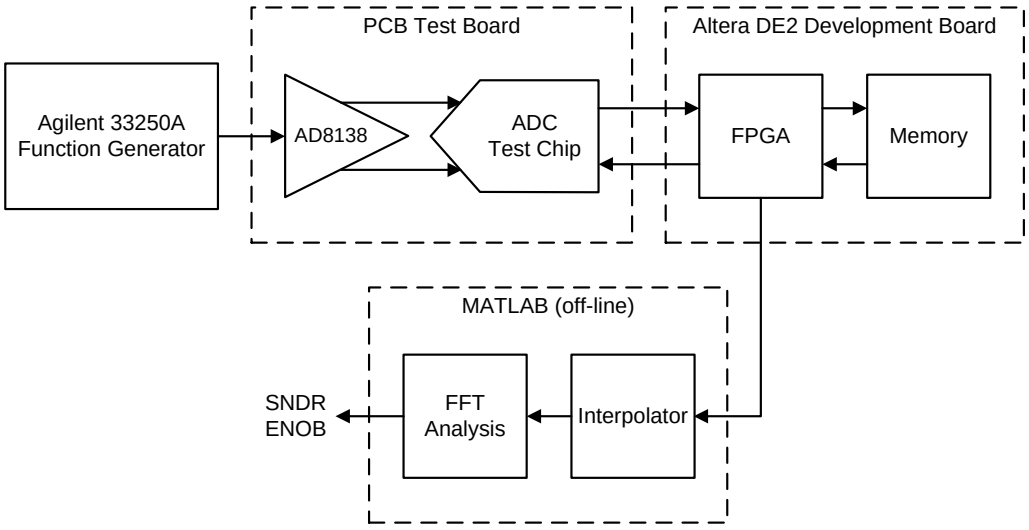


Figure A.1: Test setup for asynchronous ADC FFT tests.

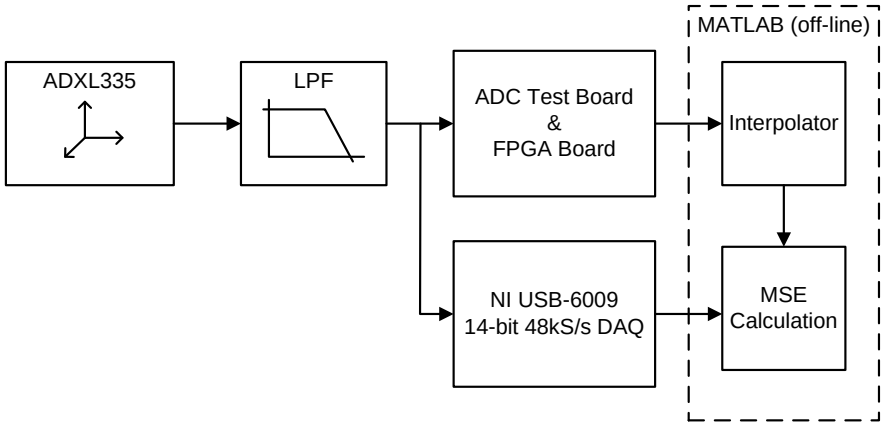


Figure A.2: Test setup for asynchronous ADC accelerometer test.

with an Analog Devices ADXL335 3-axis accelerometer mounted on an EVAL-ADXL335Z evaluation board. The accelerometer output bandwidth was matched to the ADC input bandwidth of 1kHz using a 4^{th}-order low-pass RC filter built on a breadboard. The ADC digitized the filtered x-axis output of the accelerometer as the evaluation board was struck with varying force within the ±3g dynamic range of the accelerometer. In order to allow the effective output resolution of the sampled data to be determined, the output of the accelerometer was also sampled

with a National Instruments USB-6009 14-bit 48kS/s data acquisition device (DAQ). The mean squared error between the two sampled outputs was calculated after the output of the prototype ADC was interpolated to match the sampling rate of the DAQ and any gain and offset between the two outputs was removed.

For the ECG test presented in Section 4.6.4, the accelerometer in Fig. A.2 was replaced with an ECG electrode followed by a front-end amplifier. A three electrode arrangement was used to measure the ECG signal, where one signal electrode was placed on the left and right shoulders of the test subject and the reference electrode was placed on the right ankle. A buffer was added between the front-end amplifier and the ADC to bring the common mode level within the range required by the ADC.

A.2 SAR ADC

The test setup used to measure the performance of the SAR ADC presented in Chapter 6 is shown in Fig. A.3. The sinusoidal input and clock signals were generated by two Agilent E8257D PSG analog signal generators. For frequencies below 100kHz, the input signal was generated by an Agilent 33250A function generator. Before the input and clock signals were applied to the ADC, they were filtered with different filters depending on their frequency. The 9.5MHz clock signal was filtered with a Mini-Circuits SBP-10.7+ band-pass filter in series with two CirQtel FLT low-pass filters. For input frequencies below 1MHz, low-pass filters from Kiwa Electronics were used. For input frequencies between 1MHz and 5MHz, TTE LE1182-T low-pass filters were used. For input frequencies

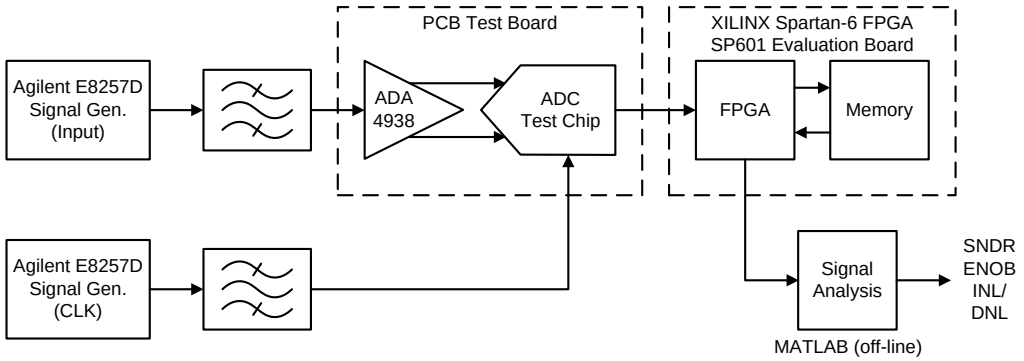


Figure A.3: Test setup for SAR ADC performance measurements.

above 10MHz, CirQtel FBT band-pass filters were used. An Analog Devices ADA4938 differential ADC driver was used to convert the single-ended input to a differential signal for the ADC. A XILINX Spartan-6 FPGA SP601 evaluation board was used to acquire and store the output of the ADC. After each test was completed, the resulting digital output was readout from memory to MATLAB. Standard FFT analysis was then performed to calculate the SNDR and ENOB of the sampled data. The INL and DNL of the ADC were calculated using the histogram testing method described in [108]. For the histogram test, 2^{20} (~1 million) points were collected for a 100kHz input and a sample rate of 5MS/s.

A.3 Compressed Sensing AIC

The test setup used to measure the performance of the compressed sensing AIC presented in Chapter 6 is shown in Fig. A.4. Before each test, the sparse input and pseudo-random clock signals were generated in MATLAB and programmed into two Agilent 33250A arbitrary waveform generators (AWGs) through their GPIB interfaces. The trigger points of each AWG were synchronized so that the two waveforms overlapped. The resulting pseudo-

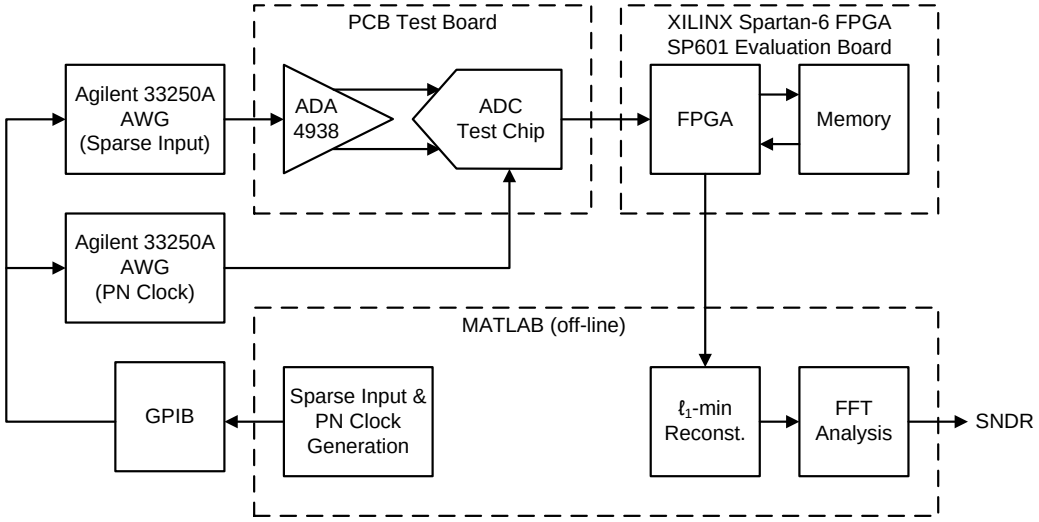


Figure A.4: Test setup for compressed sensing AIC performance measurements.

random samples were then acquired with the FPGA evaluation board and readout to MATLAB after each test completed. The spectrum of the input signal was then reconstructed using the ℓ_1-minimization convex optimization program from [95]. The resulting SNDR was then calculated from the reconstructed spectrum.

Bibliography

[1] E. Whittaker, "On the functions which are represented by the expansions of the interpolation theory," *Proc. Royal Society Edinburgh*, Sec. A, vol. 35, pp. 181-194, 1915.

[2] H. Nyquist, "Certain topics in telegraph transmission theory," *Trans. AIEE*, vol. 47, pp. 617-644, Apr. 1928.

[3] V. Kotelnikov, "On the carrying capacity of the ether and wire in telecommunications," *Izd. Red. Upr. Svyazi RKKA*, Moscow, Russia, 1933.

[4] C. Shannon, "Communication in the presence of noise," *Proc. Institute of Radio Engineers*, vol. 37, no. 1, pp. 10-21, Jan. 1949.

[5] G. Moore, "Cramming more components onto integrated circuits," *Electronics Magazine*, vol. 38, no. 8, Apr. 1965.

[6] IEEE 802.15 WPAN Task Group 6 Body Area Networks [Online]. Available: http://www.ieee802.org/15/pub/TG6.html

[7] S. Ullah, H. Higgins, B. Braem, et al., "A comprehensive survey of wireless body area networks on PHY, MAC, and network layers solutions," *Journal of Medical Systems*, 2010, DOI: 10.1007/s10916-010-9571-3.

[8] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V. Leung, "Body area networks: a survey," *Mobile Networks and Applicatoins*, DOI: 10.1007/s11036-010-0260-8.

[9] M. Yuce, "Implementation of wireless body area networks for healthcare systems," *Sensors and Actuators A: Physical*, vol. 162, no. 1, pp. 116-129, Jul. 2010.

[10] J. Mitola and G. Maguire, "Cognitive radio: making software radios more personal," *IEEE Perfonal Communications*, vol. 6, no. 4, pp. 13-18, Aug. 1999.

[11] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Comm.*, vol. 23, no. 2, pp. 201-220, Feb. 2005.

[12] C. Stevenson, G. Chouinard, L. Zhongding, et al., "IEEE 802.22: the first cognitive radio wireless regional area network standard," *IEEE Communications Magazine*, vol. 47, no. 1, pp. 130-138, Jan. 2009.

[13] J. Mitola, "Cognitive radio an integrated agent architecture for software defined radio," Ph.D. Dissertation, Royal Institute of Technology (KTH), Sweden, May 2000.

[14] L. Huang, M. Ashouei, F. Yazicioglu, et al., "Ultra-low power sensor design for wireless body area networks: challenges, potential solutions, and applications," *International Journal of Digital Content Technology and its Applications*, vol. 3, no. 3, Sep. 2009.

[15] J. Penders, B. Grundlehner, R. Vuller, and B. Gyselinckx, "Potential and challenges of body area networks for affective human computer interaction," *Proc. 5th International Conference on Foundations of Augmented Cognition*, 2009.

[16] R. Vullers, R. van Schaijk, I. Doms, C. van hoof, and R. Mertens, "Micropower energy harvesting," *Solid-State Electronics*, vol. 53, pp. 684-693, 2009.

[17] M. Hanson, H. Powell, A. Barth, et al., "Body area sensor networks: challenges and opportunities," *Computer*, vol. 42, no. 1, pp. 58-65, Jan. 2009.

[18] H. Inose, T. Aoki, and K. Watanabe, "Asynchronous delta-modulation system," *Electronics Letters*, vol. 2, no. 3, pp. 95-96, Mar. 1966.

[19] P. D. Sharma, "Characteristics of asynchronous delta modulation and binary slope quantized PCM systems," *Electronic Eng.*, vol. 40, pp. 32-37, Jan. 1968.

[20] N. Sayiner, H. V. Sorensen, and T. R. Viswanathan, "A new signal acquisition technique," *Proc. 35th Midwest Symp. Circuits Syst.*, Aug. 1992, vol. 2, pp. 1140-1142.

[21] N. Sayiner, H. V. Sorensen, and T. R. Viswanathan, "A level-crossing sampling scheme for A/D conversion," *IEEE Trans. Circuits Syst. II*, Analog Digital Signal Process., vol. 43, no. 4, pp. 335-339, April 1996.

[22] J. Mark and T. Todd, "A nonuniform sampling approach to data compression," *IEEE Trans. Communications*, vol. 29, no. 1, pp. 24-32, Jan. 1981.

[23] F. Aeschlimann, E. Allier, L. Fresquet, and M. Renaudin, "Asynchronous FIR filters: towards a new digital processing chain," *Proc. 10th Int. Symp. Asynchronous Circuits Syst.*, Apr. 2004, pp. 198-206.

[24] Y. W. Li, K. L. Shepard, and Y. P. Tsividis, "Continuous-time digital signal processors," *Proc. 11th IEEE Int. Symp. Asynchronous Circuits Syst.*, Mar. 2005, pp. 138-143.

[25] B. Schell and Y. P. 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. 2472-2481, Nov. 2008.

[26] Y. W. Li, K. L. Shepard, and Y. P. Tsividis, "A continuous-time programmable digital FIR filter," *IEEE J. Solid-State Circuits*, vol. 41, no. 11, pp. 2512-2520, Nov. 2006.

[27] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489-509, Feb. 2006.

[28] E. Candes and T. Tao, "Near-optimal signal recovery from random projections: universal encoding strategies," *IEEE Trans. Inform. Theory*, vol. 52, no. 12, pp. 5406-5425, Dec. 2006.

[29] D. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289-1306, Apr. 2006.

[30] E. Candes, "Compressive sampling," in *Proc. Int. Cong. Mathematicians*, Madrid, Spain, vol. 3, Aug. 2006, pp. 1433-1452.

[31] E. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inform. Theory*, vol. 51, no. 12, pp. 4203-4215, Dec. 2005.

[32] D. Healy, D. Brady, "Compression at the physical interface," *IEEE Signal Proc. Magazine*, vol. 25, no. 2, pp. 67-71, Mar. 2008.

[33] E. Candes and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Proc. Magazine*, vol. 25, no. 2, pp. 21-30, Mar. 2008.

[34] J. Tropp, J. Laska, M. Duarte, J. Romberg, and R. Baraniuk, "Beyond Nyquist: efficient sampling of sparse bandlimited signals," *IEEE Trans. Inform. Theory*, vol. 56, no. 1, pp. 520-544, Jan. 2010.

[35] R. Baraniuk, "Compressive sensing," *IEEE Signal Proc. Magazin*, vol. 24, no. 4, pp. 118-124, Jul. 2007.

[36] D. Cabric, S. Mishra, and R. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," *38th Asilomar Conference on Signal, Systems and Computers*, pp. 772-776, Nov. 2004.

[37] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 116-130, 2009.

[38] A. Ghasemi and E. Sousa, "Spectrum sensing in cognitive radio networks: requirements, challenges and design trade-offs," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 32-39, Apr. 2008.

[39] M. Mishali and Y. Eldar, "Wideband spectrum sensing at sub-nyquist rates," [Online] arXiv 1009.1305, Sep. 2010.

[40] Analog Devices Staff and D. Sheingold, *Analog-Digital Conversion Handbook*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[41] D. Johns and K. Martin, *Analog Integrated Circuit Design*. John Wiley & Sons, 1997.

[42] C. Chui, *An Introduction to Wavelets*. San Diego, CA: Academic Press, 1992.

[43] W. Kester, *The Data Conversion Handbook*. Analog Devices, Inc., 2005.

[44] J. Craninckx and G. Van der Plas, "A 65fJ/conversion-step 0-to-50MS/s 0-to-0.7mW 9b charge-sharing SAR ADC in 90nm digital CMOS," *IEEE Int. Solid-State Circ. Conf.*, Feb. 2007, pp. 246-247.

[45] M. Yoshioka, K. Ishikawa, T. Takayama, and S. Tsukamoto, "A 10b 50MS/s 820µW SAR ADC with on-chip digital calibration," *IEEE International Solid-State Circuit Conference*, pp. 384-385, Feb. 2010.

[46] K. Sayood, *Introduction to Data Compression*. San Diego, CA: Academic Press, 2000.

[47] T. Welch, "A technique for high-perofrmance data compression," *Computer*, vol. 17, no. 6, pp. 8-19, Jun. 1984.

[48] JPEG Homepage [Online]. Available: http://www.jpeg.org/jpeg/index.html

[49] N. Ahmed, T. Natarajan, and K. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-23, no. 1, pp. 90-93, Jan. 1974.

[50] The MPEG Homepage, [Online]. Available: http://mpeg.chiariglione.org/

[51] J. Princen, A. Johnson, and A. Bradley, "Subband/transform coding using filter bank designs based on time domain aliasing cancellation," *IEEE International Conf. on Acoustics, Speech, and Signal Procissing*, pp. 2161-2164, 1987.

[52] C. Plack, *The Sense of Hearing*. Lawrence Erlbaum Associates, 2005.

[53] B. Atal, "The history of linear prediction," *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 154-161, Mar. 2006.

[54] R. Shafer and L. Rabiner, "Digital representations of speech signals," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 662-677, Apr. 1975.

[55] J. Haskew, J. Kelly, R. Kelley, and T. Mckinney, "Results of a study of the linear prediction Vocoder," *IEEE Transactions on Communications*, vol. 21, no. 9, pp. 1008-1015, Sep. 1973.

[56] "Waveform coding techniques," Cisco Systems, Document ID: 8123 [Online]. Available: http://www.cisco.com/application/pdf/paws/8123/waveform_coding.pdf

[57] M. Schroeder and B. Atal, "Code-excited linear prediction (CELP): high-quality speech at very low bit rates," *IEEE Conf. on Acoustics, Speech, and Signal Processing*, vol. 10, pp. 937-940, 1985.

[58] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*. Springer, 1991.

[59] A. Wyner and J. Ziv, "The rate-distortion function for sourcee coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. IT-22, no. 1, pp. 1-10, Jan. 1976.

[60] Y. Fisher, *Fractal Image Compression: Theory and Application*. Springer, 1994.

[61] S. Mallat, *A Wavelet Tour of Signal Processing*. San Diego, CA: 1998.

[62] MIT-BIH Arrhythmia Database [Online]. Available: http://www.physionet.org/physiobank/database/mitdb/

[63] E. Allier, G. Sicard, L. Fesquet, and M. Renaudin, "A new class of asynchronous A/D converters based on time quantization," *Proc. 9th Int. Symp. Asynchronous Circuits Syst.*, May 2003, pp. 196-205.

[64] E. Allier, J. Goulier, G. Sicard, A. Dezzani, E. Andre, and M. Renaudin, "A 120nm low power asynchronous ADC," *Proc. 2005 Int. Symp. Low Power Electronics Design*, Aug. 2005, pp. 60-65.

[65] D. Koscielnik and M. Miskowicz, "Asynchronous sigma-delta analog-to-digital converter based on the charge pump integrator," *Analog Integrated Circuits and Signal Processing*, vol. 55, pp. 223-238, 2008.

[66] S. Yan Ng, "A continuous-time asynchronous sigma-delta analog to digital converter for broadband wireless receiver with adaptive digital calibration technique," Ph.D. Dissertation, The Ohio State University, Columbus, Ohio, 2009.

[67] Z. Zhao and A. Prodic, "Continuous-time digital controller for high-frequency DC-DC converters," *IEEE Trans. Power Electronics*, vol. 23, no. 2, pp. 564-573, Mar. 2008.

[68] K. Kozmin, J. Johansson, and J. Delsing, "Level-crossing ADC performance evaluation toward ultrasound application," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 8, pp. 1708-1719, Aug. 2009.

[69] T. Wang, D. Wang, P. J. Hurst, B. C. Levy, and S. H. Lewis, "A level-crossing analog-to-digital converter with triangular dither," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 9, pp. 2089-2099, Sept. 2009.

[70] M. Trakimas and S. Sonkusale, "A 0.8 V asynchronous ADC for energy constrained sensing applications," *IEEE Custom Integrated Circuits Conf.*, Sept. 2008, pp. 173-176.

[71] M. Trakimas and S. Sonkusale, "An adaptive resolution asynchronous ADC architecture for data compression in energy constrained sensing applicatoins," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 7, Jul. 2011.

[72] S. Haykin, *Communication Systems*. New York, NY: John Wiley & Sons, 2001, pp. 229-234.

[73] M. Kurchuk and Y. Tsividis, "Signal-dependent variable-resolution clockless A/D conversion with application to continuous-time digital signal processing," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 5, pp. 982-991, May 2010.

[74] R. Agarwal, M. Trakimas, and S. Sonkusale, "Adaptive asynchronous analog to digital conversion for compressed biomedical sensing," *IEEE Biomedical Circuits Syst. Conf.*, Nov. 2009, pp. 69-72.

[75] K. Guan, S. Koxat, and C. Singer, "Adaptive reference levels in a level-crossing analog-to-digital converter," *EURASIP J. Advances Sig.. Processing*, vol. 2008, Jan. 2008.

[76] R. van de Plassche, *CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters*. Kluwer Academic Publishers, 2003.

[77] B. Razavi, *Design of Analog CMOS Integrated Circuits*. New York, NY: McGraw-Hill, 2000, pp. 23-25,392-393.

[78] T. H. Kim, J. Keane, H. Eom, and C. H. Kim, "Utilizing reverse short-channel effect for optimal subthreshold circuit design," *IEEE Trans. VLSI Syst.*, vol. 15, no. 7, pp. 821-829, Jul. 2007.

[79] B. Schell, "Continuous-Time Digital Signal Processors: Analysis and Implementation," Ph.D. dissertation, Columbia University, New York, NY, 2008.

[80] B. Schell and Y. Tsividis, "Analysis of continuous-time digital signal processors," *IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 2232-2235.

[81] V. Giannini, P. Nuzzo, G. Van der Plas, J. Craninckx, et al., "An 820μW 9b 40MS/s noise-tolerant dynamic-SAR ADC in 90nm digital CMOS," *IEEE ISSCC Dig. Tech. Papers*, Feb. 2008, pp. 238-239.

[82] "±2g/±4g/±8g three axis low-g digital output accelerometer," Freescale Semiconductor Inc., Austin, TX, Doc. MMA7455L, Rev. 10, Dec. 2009.

[83] S. Hwang, M. Trakimas, and S. Sonkusale, "A low-power asynchronous ECG acquisition system in CMOS technology," *International Conf. of the IEEE Engineering in Medicine and Biology Society*, pp. 5262-5265, Sep. 2010.

[84] A. Abidi, "The path to the software-defined radio receiver," *IEEE Journal of solid-state circuits*, vol. 42, no. 5, pp. 954-966, May 2007.

[85] Y. Liuqing and G. Giannakis, "Ultra-wideband communications: an idea whose time has come," *IEEE Signal Processing Magazine*, vol. 21, no. 6, pp. 26-54, Nov. 2004.

[86] M. Lustig, D. Donoho, J. Santos, and J. Pauly, "Compressed sensing MRI," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 72-82, Mar. 2008.

[87] M. Duarte, M. Davenport, D. Takhar, J. Laska, R. Sun, K. Kelly, and R. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Proc. Magazine*, vol. 25, no. 2, pp. 83-91, Mar. 2008.

[88] F. Chen, A. Chandrakasan, and V. Stojanovic, "A signal-agnostic compressed sensing acquisition system for wireless and implantable sensors," *IEEE Custom Integrated Circuits Conference*, Sep. 2010.

[89] P. Baheti and H. Garudadri, "An ultra low power pulse oximeter sensor based on compressed sensing," *6th International Workshop on Wearable and Implantable Body Sensor Networks*, pp. 144-148, Jun. 2009.

[90] V. Patel, G. Easley, D. Healy, and R. Chellappa, "Compressed synthetic aperture radar," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 244-254, Apr. 2010.

[91] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Transactions in Information Theory*, vol. 56, pp. 1982-2001, Apr. 2010.

[92] S. Chen, D. Donoho, and M. Saunders, "Atomic decompositoin by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33-61, 1998.

[93] S. Kirolos, J. Laska, M. Wakin, R. Baraniuk, et al., "Analog to information conversion via random demodulation," in *Proc. 2006 IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software*, Dallas, TX, Oct. 2006, pp. 71-74.

[94] J. Laska, S. Kirolos, Y. Massoud, R. Baraniuk, et al., "Random sampling for analog-to-information conversion of wideband signals," in *Proc. 2006 IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software*, Dallas, TX, Oct. 2006, pp. 119-122.

[95] E. Candes and J. Romberg, "ℓ1-MAGIC : Recovery of sparse signals via convex programming," [Online]. Available: http://www.acm.caltech.edu/l1magic/

[96] T. Ragheb, J. Laska, H. Nejati, et al., "A prototype hardware for random demodulatoin based compressive analog-to-digital conversion," *51st Midwest Symposium on Circuits and Systems*, pp. 37-40, Aug. 2008.

[97] S. Pfetsch, T. Ragheb, J. Laska, et al., "On the feasibility of hardware implementation of sub-Nyquist random-sampling based analog-to-information conversion," *IEEE International Symposium on Circuits and Systems*, pp. 1480-1483, May 2008.

[98] M. Mishali and Y. Eldar, "From theory to practice: sub-Nyquist sampling of sparse wideband analog signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 375-391, Apr. 2010.

[99] M. Mishali and Y. Eldar, "Xampling: analog data compressoin," *Data Compression Conference*, pp. 366-375, Mar. 2010.

[100] M. Mishali, Y. Eldar, O. Dounaevsky, and E. Shoshan, "Xampling: analog to digital at sub-Nyquist rates," *IET Circuits, Devices & Systems*, vol. 5, no. 1, pp. 8-20, Jan. 2011.

[101] R. Robucci, J. Gray, K. Leung, J. Romberg, and P. Hasler, "Compressive sensing on a CMOS separable-transform image sensor," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1089-1101, Jun. 2010.

[102] M. van Elzakker, E. van Tuijl, P. Geraedts, D. Schinkel, E. Klumperink, and B. Nauta, "A 1.9μW 4.4fJ/conversion-step 10b 1MS/s charge-

redistribution ADC," *IEEE Int. Solid-State Circ. Conf.*, Feb. 2008, pp. 244-245.

[103] M. Gustavsson, J. Wikner, and N. Tan, *CMOS Data Converters for Communications*. Kluwer Academic Publishers, 2000.

[104] A. Abo and P. Gray, "A 1.5-V, 10-bit, 14.3-MS/s CMOS pipeline analog-to-digital converter," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 5, May 1999.

[105] T. Chi, A. Pezeshki, L. Scharf, and R. Calderbank, "Sensitivity to basis mismatch in compressed sensing," *IEEE Int. Conf. Acoustics Speech and Signal Proc.*, Mar. 2010, pp. 3930-3933.

[106] M. Duarte and R. Baraniuk, "Spectral compressive sensing," *IEEE Trans. Signal Proc.*, submitted for publication.

[107] D. Chae, P. Sadeghi, and R. Kennedy, "Effects of basis-mismatch in compressive sampling of continuous sinusoidal signals," *2nd Int. Conf. Future Comp. Comm.*, vol. 2, May. 2010, pp. 739-743.

[108] "Histogram testing determines DNL and INL errors," Maxim Inc., Application Note 2085 [Online]. Available: http://www.maxim-ic.com/app-notes/index.mvp/id/2085