Summer 2014

Improved Random Demodulator for Compressed Sensing Applications

Sathya Narayanan Hariharan
Purdue University

Graduate School ETD Form 9
(Revised 01/14)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Sathya Narayanan Hariharan

Entitled

Improved Random Demodulator for Compressed Sensing Applications

For the degree of Master of Science

Is approved by the final examining committee:

Walter D. Leon-Salas

Helen McNally

Suranjan Panigrahi

To the best of my knowledge and as understood by the student in the *Thesis/Dissertation Agreement. Publication Delay, and Certification/Disclaimer (Graduate School Form 32)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Walter D. Leon-Salas

Approved by Major Professor(s): _____

Approved by: Duane Dunlap 07/01/2014

Head of the Department Graduate Program Date

IMPROVED RANDOM DEMODULATOR FOR COMPRESSED SENSING APPLICATIONS

A Thesis

Submitted to the Faculty

of

Purdue University

by

Sathya N. Hariharan

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

August 2014

Purdue University

West Lafayette, Indiana

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

ABSTRACT

Hariharan, Sathya N. M.S., Purdue University, August 2014. Improved Random Demodulator for Compressed Sensing Applications . Major Professor: Walter D. Leon-Salas.

The advances in the field of signal processing have led to the continuous increase in the bandwidth of signals. Sampling these signals becomes harder and harder due to the increased bandwidth. This brings in need for a complex high rate ADCs to meet the Nyquist rate which is the minimum rate to avoid aliasing. For a given increase in bandwidth, there has to be a corresponding increase in the sampling rate of ADC. This might not be possible in the near future at the current rate of increase in bandwidth. Hence, there is a need to replace the current Nyquist rate sampling method by a process that relaxes the requirements but still keeps the quality of signal reconstruction good .

Compressed sensing is a new technique in the field of signal acquisiton. Compressed sensing allows a signal to be acquired below Nyquist rate if the signal is sparse in a given domain. Compressed sensing makes possible to acquire sparse signals at rates below Nyquist rate. Signals like audio and images are sparse and can be sampled at a rate below the Nyquist rate. The random demodulator (RD) is a hardware architecture that is used to implement compressed sensing. A direct implementation of compressed sensing in hardware requires several copies of the RD. To reduce the complexity fewer RDs can be used. Usage of fewer RDs comes at the cost of decreased signal reconstruction performance. The contribution of this thesis is about improving the efficiency of RD. First contribution of this thesis involves proposing a new RD architecture that improves signal reconstruction quality using a post-acquisition randomization step. The second contribution of this

thesis is to develop a hardware platform for compressed sensing using field programmable analog arrays (FPAAs) and field programmable gate arrays (FPGAs). This platform can be used to test new architectures of RD in hardware.

CHAPTER 1. INTRODUCTION

This chapter of the thesis provides the necessity for such a research. It gives an idea about the importance, significance and possible applications with compressed sensing and random demodulator.

1.1 Scope

The sampling rate of analog-to-digital converters (ADC) keeps increasing due to the continuous increase in bandwidth of the signal. There is increase in sampling rate of the ADC because the signals have to be sampled at the Nyquist rate. This increases the complexity of ADCs. Better and efficient methods have to be found to replace the conventional methods to relax the increasing need for more complex ADCs. Compressed sensing is a signal acquisition technique for efficient reconstruction of signals sparse in the frequency domain. Compressed sensing works with only signals that are sparse in the frequency domain (Davenport, 2012). Sparse in frequency domain implies that the discrete cosine transform (DCT) of the signal has very less distinct non-zero values . A signal with only 1 to 5 percent non-zero values in its frequency domain is considered sparse according to this study. A wide variety of signals are sparse in the frequency domain. For example: radar, ultraband communication, video, music and images are some of the signals to which compressed sensing can be applied.

The primary question to this research is how to increase the efficiency of an architecture called single RD. The main objective of this project is to accomplish reconstruction of signals effectively through RD. This research aims to create a hardware platform of the RD for demonstration and future research in the field of compressed sensing. Reconstruction results are better when the resemblance of the

reconstructed signal to the original signal is greater. This is measured by parameters such as mean square error (MSE) and signal to noise ratio (SNR). Smaller the MSE, the better the reconstruction. The greater the SNR, the better the reconstruction. The present day reconstruction method is based on to the Nyquists sampling theorem.

1.2 Significance

Most real world signals are sparse in the frequency domain. Certain other signals which are not sparse, can be transformed to a sparse representation by a function known as the sparsifying transform. This makes compressed sensing applicable to a wide variety of signals. Since this study can be applied to a variety of signals, implementing a hardware platform to test different architectures will be of great significance. This could help in achieving better levels of reconstruction using compressed sensing(CS) concepts.

There are certain other types of signals called compressible signals. Compressible signals are those which are well approximated by sparse signals (Foucart & Rauhut, 2013). These encompass a wide variety of signals like music, speech, radio waves, magnetic resonance imagining (MRI) data (Lusting, Donoho, Santos & Pauly, 2008). Most communications involves high bandwidth with images and video being transmitted. Compressed sensing implementation will be a breakthrough for complex ADCs and a very good tool in the field of signal processing and communications.

Compressed sensing is a new theory for signal acquisition. A hardware implementation in the field of compressed sensing could be a real motivation to develop much efficient alternatives. A printed circuit board is developed at the end of this thesis to test various hardware architecutures which use compressed sensing concepts.

1.3 Research Objective

To design an efficient and flexible hardware platform to implement compressed sensing signal acquisition. This work is divided into two parts:

- Improving single RD through post acquisition randomization step.
- Development of a programmable hardware platform to test different architectures of RD.

1.4 Assumptions

The assumptions for this study include:

- The hardware platform does not introduce significant distortion or noise.
- The quantization error of ADC is small enough to make reconstructions possible.
- The sensing random matrices used in this study are pseudo-random and are assumed to be random enough to apply compressed sensing.
- Measurement noise is assumed to be smaller than the quantization noise.

1.5 Limitations

The limitations for this study include:

- This study will only focus on signals sparse in the frequency domain.
- This study uses LFSR (linear feedback shift register) based pseudo-random numbers. These random numbers repeat themselves after a period. The repetition period is assumed to be larger than the signal length.

1.6 Delimitations

The delimitations for this study include:

- This study is not applicable to non-sparse signals in the frequency domain.
- The results from this study must be compared to the same arrangement in software and not to previous studies which do not quantize measurements or use ADCs with better resolution than 8-bits in the random demodulator arrangement.

1.7 Definitions

Analog-to-Digital Convereter (ADC): It is a device that converts a continuous physical quantity like voltage to a digital binary sequence of numbers that represent the quantity (Lathi (1998)).

Compressed sensing: It is a signal acquisition technique for efficient acquisition and reconstruction of a sparse signal by finding solutions to underdetermined linear systems (Donoho & David, 2006).

Compression ratio: It is the ratio between uncompressed size to that of the compressed size of the signal (Poynton, 2012).

Field Programmable Analog Array (FPAA): It is an integrated circuit which can be reconfigured to implement different analog circuits (Anadigm Inc (2003)).

Field Programmable Gate Array (FPGA): It is an integrated circuit designed to be configured by a designer (Wisniewski & Remigiusz (2009)).

Nyquist rate: Minimum sampling frequency required to avoid aliasing (Shannon, 1949).

Nyquist-Shannon sampling theorem: The sampling frequency should be at least twice the highest frequency in the signal (Shannon, 1949).

Quantization Error: In analog-to-digital conversion the difference between actual analog value and quantized digital value is the quantization error (Gersho & Gray (1991)).

Signal to Noise Ratio (SNR): Signal-to-noise ratio is defined as the power ratio between a signal (meaningful information) and the background noise (Johnson, 2006).

Sparse signal: Signal whose frequency domain has very less unique non-zero values. The frequency domain of a signal can be viewed by taking its Fourier transform.

Underdetermined linear system: A system where the number of unknowns are greater than the number of equations (Lai, 2009)

1.8 Summary

This chapter provided an overview of the research, its scope and significance. It also gave an outline of the focus for this research and the considerations to be made. The next chapter provides a review of relevant literature which helps in understanding the motivation for the methodology used in this research.

CHAPTER 2. LITERATURE REVIEW

Digital signal acquisition follows the well known Nyquist-Shannon's theorem that states the sampling rate must be twice the maximum frequency present in the signal. The sampling theorem forms the basis for all of the existing reconstructions (Candes & Wakin, 2008). In case of images, the sampling rate is determined by the spatial frequency. For signal acquisition and data conversion from the analog to digital world, ADCs are being used. ADCs work on par with the Nyquist's sampling rate. Usage of compressed sensing (CS) concepts makes reconstruction possible even if signals are acquired less than Nyquist rate. The basic concept of compressed sensing and architectures making use of these constructs are been explained below.

2.1 Compressed Sensing

The theory of compressed sensing states that a sparse signal could be recovered from a very few number of samples (Candes & Wakin, 2008). Compressed sensing is bounded by two principles

- Sparsity : The rate of information in the signal which could be less, than the actual bandwidth of the signal. Sparse signal implies that the number of non-zero components in the frequency domain is very less. Compressed sensing exploits the fact that a wide variety of signals are actually sparse or could be represented as sparse in nature. Sparsity is represented by K in this study.
- Incoherence: Incoherence extends the duality between the domains of time and frequency. This gives the idea that signals sparse and spread out in one domain are dense in the other.

Using these principles it is possible to construct very efficient protocols that capture only the necessary information in the signal. Candes and Wakin (2008) explained that there are numerical optimization techniques from which most of the signal can be reconstructed. Compressed sensing, in other words, is a very efficient method of signal acquisition. Sparsity and incoherence the principles behind compressed sensing are explained in the following sections.

2.1.1 Sparsity

A wide variety of signals have sparse representations when represented through a proper basis (Candes & Wakin, 2008). Mathematically speaking a signal can be viewed as a *x(t)* in time domain that belongs to the set of real numbers which can be expanded in a basis $\Psi = [\Psi_1 \Psi_2 ... \Psi_n]$ as shown in equation 2.1.

$$x(t) = \sum_{i=1}^{n} \Psi_i(t) a_i \tag{2.1}$$

where Ψ is an n *x* n matrix (with columns $\Psi_1 ... \Psi_n$) and *x(t)* is the signal. Sparsity implies that when a signal has sparse expansion, small coefficients can be neglected without a great loss(Candes & Wakin, 2008). A signal, is called *K-sparse* if K of its coefficients are non-zero. Sparsity has a lot of implications on the acquisition process itself. Sparsity determines if efficient acquisition can happen nonadaptively (Candes & Wakin, 2008). This tranformation basis $\mathbf{\Psi}$ is a transformation which transforms a signal from one domain to another.

2.1.2 Incoherent sampling

The equations of compressed sensing are given below

$$x = \Psi a \tag{2.2}$$

$$y = \Phi x \tag{2.3}$$

where **x** is the mathematical representation of time domain signal *x(t)*, **a** is the mathematical representation of frequency spectrum of the signal *x(t)*, **Φ** and **Ψ** are orthonormal basis used to acquire the signal. **y** is the measurements of the acquired signal. **Ψ** is a transformation basis which transforms the signal from frequency domain to time domain (also shown in equation 2.1). **Φ** is called the sensing matrix which is random in nature. These two matrices play a significant role in acquiring the signal. These two matrices **Φ** and **Ψ** are orthonormal to one another. (Candes & Wakin,2008) The coherence between two matrices in this case **Φ** and **Ψ** is shown in equation 2.4.

$$\mu(\Phi, \Psi) = \sqrt{n}. \max_{1 \leq k,j \leq n} | < \Phi_k, \Psi_j > | \tag{2.4}$$

Coherence is the maximum of the correlation between the rows of **Φ** and the columns of **Ψ**. This factor $\mu(\mathbf{\Phi}, \mathbf{\Psi})$ must be as low as possible thus making the incoherence maximum. Compressed sensing (CS) needs large incoherence between the **Φ** and **Ψ** matrices to work more effectively. Spikes (dirac deltas) and sinusoids have been experimentally observed to give maximum incoherence (Goyal, Fletcher, Rangan, 2008). Therefore, dirac deltas in **Φ** and Fourier basis as **Ψ** will make measurements easier and more effective(Candes,2008). Mathematically **Φ** is a random matrix and **Ψ** is a discrete cosine transform (DCT). **Ψ** is taken as a DCT and not as an FFT(fast foutier transform). This is because DCT being cosines yields real valued signals whereas the FFT leads to imaginary parts of the signal which makes calculations more complex.

2.1.3 Minimum measurements criteria for CS

Φ, the sensing matrix is made random so that incoherence is large enough between **Φ** and **Ψ**. But this gives a chance that reconstruction will fail (Goyal, Fletcher, Rangan, 2008).

Let M be the minimum number of measurements required. Let x be the time domain signal that has a sparsity factor K in the frequency domain. Let the length of the signal x be MN. The length of the spectrum of the signal (x) is N. Mathematical results show that $M \approx 2K\log(N/K)$. This has been proved to be a sharp threshold to make the reconstruction process successful. The above result is derived from a convex optimization problem which is later discussed in equation 2.13 (Foucart & Rauhut, 2013).

2.2 Random Demodulator

Random demodulator (RD) is an architecture that is used to acquire sparse signals in compressed sensing (Tropp, Laska, Duarte, Romberg & Baranuik, 2010). Referring equation 2.3, $\mathbf{y}$ is the measurements of the signal $\mathbf{x}$. $\mathbf{y}$ is obtained by matrix multiplication of $\mathbf{\Phi}$ and $\mathbf{x}$. This operation needs a mutiplier and an integrator to accumulate the product. The signal is multiplied with a high rate pseudo-random sequence, which spreads throughout the entire spectrum of the signal. This pseudo-random sequence is represented as $\mathbf{\Phi}$ in matrix form. $\mathbf{\Phi}$ is called the sensing matrix as discussed previously in equation 2.3. The multiplied signal is then summed up using an integrator. These integrated samples are digitized by the use of low rate ADC. This mutiplication and accumulation operation is done by the random demodulator. Given the fact that the signal is sparse, lesser number of tones are present in the signal. So the tones and respective amplitudes are present in these digitized measurements in a different form. In the Figure 2.1 (Hariharan & Leon-Salas, 2014), F_s is the Nyquist sampling rate. $T_s = 1/F_s$ and NT_s is the integration time and $\mathbf{y}$ is the measurement samples.

The measurements from the RD are recovered by using a compressed sensing solver. Convex optimization is one way by which a compressed sensing solver could be implemented. The greatest advantage of the RD is that the need for high ADC sampling rate is no more (Tropp, Laska, Duarte, Romberg & Baranuik, 2010). But

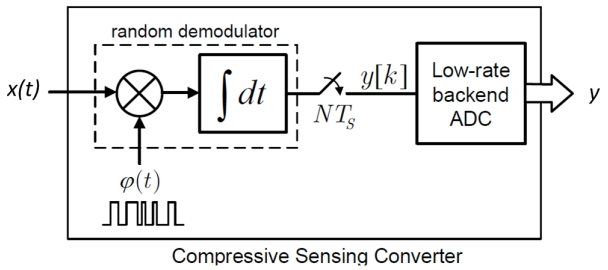

Figure 2.1. Single RD architecture.

there is a problem with using a low rate ADC - the original signal $\mathbf{x}$ is no longer a linear representation of the samples in time (Oppenheim, Schafer & Buck, 1999). Rather $\mathbf{x}$ is actually smeared up evenly in the measurements. The reconstruction is much more complex and highly non-linear. This makes the reconstruction much more computationally intense to perform although it lessens the complexity of ADCs (Tropp, Laska, Duarte, Romberg & Baranuik, 2010). This seems acceptable because compressed sensing makes reconstruction possible at low sampling rates. Eventhough the single RD samples below Nyquist rate, the quality of reconstruction is not good enough. This leads to different versions of RD later being developed. Two of those architectures are discussed in sections 2.3.1 and 2.3.2.

2.2.1 Signal model

Tropp et al (2010) primarily focused on a class of signals characterizing them as follows:

- Band limited: The maximum or peak frequency of the signal is bounded
- Sparse: The active number of frequencies is small compared to the bandwidth
- Periodic: Tones are integral frequency in hertz

In this thesis, the focus is on signals with properties as stated above. Tropp et al, 2010 developed a mathematical model of a signal as described above. The

signal is shown in the equation 2.5. K is the number of active tones in the signal. So *x(t)* is K-sparse in the frequency domain(represented by a).

$$x(t) = \sum_{\omega \in \Omega} a_\omega e^{-2\pi\omega t} \tag{2.5}$$

The above equation is just an example of a signal model. This is not used in this thesis. Let W/2 be a positive integer assumed to exceed highest frequency present in the signal. In equation 2.5, Ω is a set of K-integer valued frequencies that satisfies

$\Omega \subset [0, \pm 1, \pm 2, ... \pm (W/2 - 1), W/2]$

$a_\omega : \omega \in \Omega$

Real world examples of the modeled signals are (Tropp et al, 2010):

- Communication signals
- Acoustic signals
- Slowly varying chirps

2.2.2 Sensing matrix of the RD

Let *x(t)* be the signal. A simple RD action can be described by the sensing matrix given below: (Harms, Bajwa & Calderbank, 2013)

$$\begin{pmatrix} y[1] \\ y[2] \\ \vdots \\ y[M] \end{pmatrix} = \begin{pmatrix} \Phi_{1,1} & \Phi_{1,2} & \cdots & \cdots & \Phi_{1,MN} \\ \Phi_{2,1} & \Phi_{2,2} & \cdots & \cdots & \Phi_{2,MN} \\ \vdots & \cdots & \cdots & \ddots & \vdots \\ \Phi_{M,1} & \Phi_{M,2} & \cdots & \cdots & \Phi_{M,MN} \end{pmatrix} \begin{pmatrix} x(1) \\ x(2) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ x(MN) \end{pmatrix}$$

The Nyquist rate of the signal represented by F_s earlier in Figure 2.1. A RD collects totally M measurements and so these measurements can be collected

together to represent them as $\mathbf{y}$. The primary objective of the RD is to make sure that recovery of the signal is possible even when M is very low compared to MN, the length of the signal in time domain (Harms, Bajwa & Calderbank, 2013). In the represented matrices, $\mathbf{x}$ is the time domain representation of the signal to be reconstructed. From equations of compressed sensing, equations 2.2 and 2.3 it is known that $\mathbf{y}$ is a product of $\mathbf{x}$ and the sensing matrix $\mathbf{\Phi}$, which is a random matrix. The above represented $\mathbf{\Phi}$ is an example showing its structure.

2.2.3 Restricted isometric property (RIP)

For stable compressed sensing, the measuring matrix needs to satisfy the restricted isometric property(RIP). Measuring matrix is the matrix which is used to acquire the sparse spectrum $\mathbf{a}$ which is given by $\mathbf{\Theta} = \mathbf{\Phi}^*\mathbf{\Psi}$. The RIP is given by equation 2.7 where $\delta_k \in (0,1)$ is the restricted isometry constant (RIC) with $\mathbf{\Theta}$ satisfying the equation 2.7 below:

$$(1 - \delta_k)\|a\|_2^2 \leq \|\Theta a\|_2^2 \leq (1 + \delta_k)\|a\|_2^2 \tag{2.6}$$

or

$$|\frac{\|\Theta a\|_2^2 - \|a\|_2^2}{\|a\|_2^2}| \leq \delta_k \tag{2.7}$$

where the number of non-zero components in $\mathbf{a}$ is K. Mathematically $\|a\|_0 = K$. The equation 2.7 is the RIP of order K or for sampling a *K-sparse* signal (Harms, Bajwa & Calderbank, 2013). It is noted that RIP is the actual phenomenon that needs to be checked to select a measuring matrix $\mathbf{\Theta}$. Checking for the incoherence between $\mathbf{\Psi}$ and $\mathbf{\Phi}$ is easier. A large incoherence implies that RIP is satisfied. Thus the incoherence property is used for simplicity. It is also to be noted that RIP provides or ensures stable recovery in the below cases (Harms, Bajwa & Calderbank, 2013):

- If the signal is compressible
- Measurements are contaminated with noise

- Number of measurements M is much smaller than the length of the spectrum N

Compressible signals are signals whose frequency components, when arranged in a decreasing order rapidly decay to zero. These signals could be reduced to a *K-term* approximation denoted by $\mathbf{a}$ in the frequency domain. The *K-term* approximation can be recovered from M measurement samples. The lower bound of M is given by equation 2.8.

$$M = 2K \log\left(\frac{N}{K}\right) \tag{2.8}$$

This lower bound M will increase as the compressibility of the signal decreases (Laska et al., 2007).

2.2.3.1. Relation between RIP and signal to noise ratio

Let $\mathbf{a}$ be a *K-sparse* signal. Let $\mathbf{\Theta}$ satisfy RIP of the order K with constant δ_k. Given that, $\delta_{3k} + 3\delta_{4k} < 2$, then the SNR of the system obeys the lower bound given by equations 2.9 and 2.10 (Laska, Kirolos, Duarte, Ragheb, Baranuik & Massoud, 2007) where $\hat{\mathbf{a}}$ is the reconstructed spectrum:

$$SNR = 20 \log\left(\frac{\|a\|_2}{\|\hat{a} - a\|_2}\right) \tag{2.9}$$

$$SNR \geq 20 \log((1 + \delta_k) C_{1,K}) \tag{2.10}$$

Where $\mathbf{a}$ is the *K-sparse* signal in frequency domain, SNR is the signal to noise ratio of the random demodulator and $C_{1,K}$ is a constant that depends only on K the sparsity of $\mathbf{a}$.

2.2.4 Signal recovery algorithms

The samples taken according to the RD lead to a mathematical problem (Tropp et al, 2010). Let $\mathbf{a}$ be the sparse spectrum to be recovered (representation of $\mathbf{x}$ in the frequency domain):

$$\hat{a} = \min \|\hat{a}\|_0 \quad \text{subject to} \quad \Theta a = y \tag{2.11}$$

where $\mathbf{\Phi\Psi} = \mathbf{\Theta}$ and $\hat{\mathbf{a}}$ is the recovered sparse spectrum. The above equation gives $\mathbf{\Phi\Psi a} = \mathbf{\Phi y}$ where $\mathbf{y}$ is the measurement samples produced by the random demodulator and $\mathbf{\Phi}$ is the sensing matrix. The above problem is a l_0 minimization. l_0 minimization is much difficult to solve computationally since it is NP-hard. A NP-hard problem is a computationally complex problem that is at least as hard to solve as any problem in non-deterministic polynomial time. Recovery using l_1 norm is much better. Recovery using l_1 norm is called convex relaxation and is given below by the equation 2.12

$$\hat{a} = \min \|\hat{a}\|_1 \quad \text{subject to} \quad \|\Theta a - y\|_2 = 0 \tag{2.12}$$

The above equation is called basis pursuit. This equation does not take any measurement noise into consideration. Usually the measurements are contaminated with noise given by $\mathbf{y} = \mathbf{\Phi x} + \mathbf{n}$, where n is measuremement noise. The equation 2.12 is modified to take noise as a factor. The modified equation with an upper bound for noise ε is called basis pursuit denoise (BPDN). This equation is used in the recovery process in this thesis. It is given by the equation below:

$$\hat{a} = \min \|\hat{a}\|_1 \quad \text{subject to} \quad \|\Theta a - y\|_2 \leq \varepsilon \tag{2.13}$$

Recovery using l_1 norm is not NP-hard. The l_1 norm optimization problem can be solved efficiently in polynomial time. Convex relaxation has been found to promote signal recovery as a function of sparsity. Convex relaxation being very powerful is capable of achieving very good reconstruction results. In actual or real

applications, the signals are compressible rather than being sparse. Compressible signals are signals that can be approximated to sparse signals. The hardware measurements are quantized. So the chances of noise increases. Convex relation is very robust to noise. So the problem with CS is not the noise by itself but the method of compressing these signals into smaller number of samples (Tropp et al, 2010). Thus this study aims in developing an architecture by designing a hardware platform which produces measurement samples that will yield better signal recovery.

2.3 RD Hardware Architectures

RD has been implemented on hardware using different kinds of architectures. The digital and the analog parallel architectures are hardware architectures which yield good reconstruction results. Various authors (Chen, Chandrakasan & Stojanovic 2012., Laska et al., 2008) have claimed the advantages from these architecture which includes decreased power consumption to increased efficiency in spite of handling much lesser data in the whole process. The parallel architectures are discussed in the following sections:

2.3.1 Analog parallel RD architecture

The analog implementation of such a RD involves the use of the following components (Chen, Chandrakasan & Stojanovic, 2012):

- Multiplier - Which can be realized by a passive mixer
- Accumulator - An integrator circuit that sums up the input samples
- Sample and hold- To hold the summed up value for the ADC
- An ADC to convert the analog measurements into digital measurements

In the work of Chen, Yu, Hoyos, Sadler and Martinez, M copies of RD are used. The multiplier operation is done with a passive mixer. The power consumed

at each point is analyzed. The ADC works at a frequency F_s/N where $F_s = 1/T_s$ and NT_s is the integration time and F_s is the Nyquist rate. The analog parallel architecture gives much better signal reconstruction with drawback that it requires M copies of the RD. This increases hardware cost and power consumption. The below shown Figure 2.2 is the analog parallel RD architecture designed by Chen, Yu, Hoyos, Sadler and Martinez.

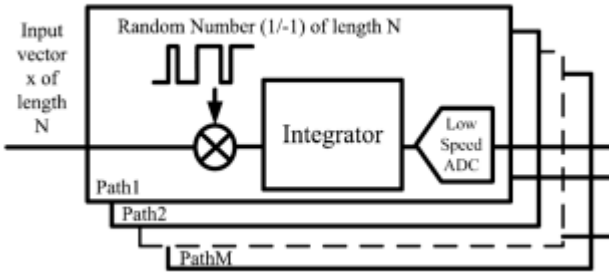

Figure 2.2. Analog parallel RD architecture.

2.3.2 Digital parallel RD architecture

In the digital parallel RD architecture the input is amplified and digitized with an ADC which works at the Nyquist rate (Chen, Chandrakasan & Stojanovic, 2012). The ADC output passes M parallel arrangement which involves multiplication with a pseudo-random sequence and accumulators which accumulate the product of signal and random sequence. The pseudo-random matrix in this architecture is a Bernoulli random matrix consisting of ± 1s. Thus, the multiplication can be done just with XOR operation and carry input of the accumulator. The digital implementation proves to be beneficial due to its lower power consumption as proved by the authors (Chen et al., 2012). Also the generation of random numbers for these have to be taken a note of. A pseudo-random binary sequence (PRBS) generator loaded with a seed is been used. This makes more memory available for CS and so better compression is possible

(Chen, Chandrakasan & Stojanovic, 2012). But the digital parallel RD architecture needs an ADC which works at the Nyquist rate. Figure 2.3 shows the work of Chen, Chandrakasan & Stojanovic on digital paralled RD architecture.

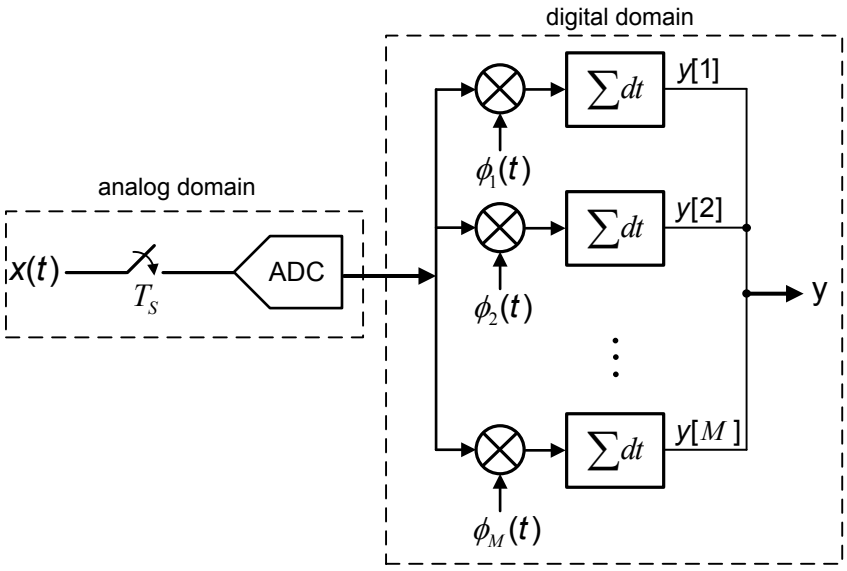

Figure 2.3. Digital parallel RD architecture.

2.3.3 Hardware implementation issues

The authors have observed the following issues while practically implementing RD architecture to perform compressed sensing (Kirolos, Ragheb, Laska, Duarte, Massoud, Baraniuk, 2006):

- In case of streaming signal, the multiplication process with a packed measurement matrix is not feasible
- There are non-idealities in the circuit level implementations
- The pseudo-random number generator might have a clock jitter. This implies that clock may arrive at a earlier or later time or might be a phase shifted to

the ideal clock. Uncertainty in the clock degrades the overall performance as given in equation 2.14

$$SNR = 20\log\left(\frac{1}{2\pi f t_j}\right) \tag{2.14}$$

where f is the signal frequency and t_j is the clock jitter.

- ADC conversion has quantization error. This is because every value is approximated to the nearest values that it could be digitized. This affects the performance as given by the equation given below

$$SNR = 6.02b + 1.76 \tag{2.15}$$

where b is the resolution of the ADC .

- The ADC non-linearity has been modeled by the equation below

$$Z = t + c_3 x^3 \tag{2.16}$$

where x is the input and Z is the output and c_3 is the corresponding constant for non-linear distortion (Yu, Zhou, Ramirez, Hoyos, Sadler, 2012).

- Integrator is not ideal. Due to the presence of filter gain, the integrator circuit can be viewed as a low pass filter. Performance of the system degrades with more non-idealities in the integrator (Kirolos,Ragheb, Laska, Duarte, Massoud, Baraniuk, 2008).

- In case of a mutiplier a mixer is widely used. The most important problem with a mixer is its non-linearity. (Kirolos,Ragheb, Laska, Duarte, Massoud, Baraniuk, 2006).

- There could be gain variation or mismatch in parallel paths leading to incorrect measurements at the integrator (Yu, Chen, Hoyos, Sadler, Gong & Qian, 2010).

- Analog mixing with spectrally rich waveforms and constructing periodic waveforms with the necessary speed of alternation is complex (Mishali, Eldar, Dounaesky & Shoshan, 2010).

2.4 Summary

This chapter provided an overview of compressed sensing and random demodulator architectures. Similar architecture overcoming the disadvantages discussed above is used in the methodology. It combines concepts of analog and digital parallel architectures. The new architecture is proposed by introducing a post acquisition randomization step. This is explained in detail in chapter 4. The literature also gave conditions for minimum number of measurements using CS for robust recovery and the convex optimization problem to be solved. These constraints are kept in mind to take the required number of measurements depending on the signal. The principle behind compressed sensing and various signal recovery techniques are discussed. The factors that lead to lower performance and the practical issues have been elaborated. Reduction of these factors could be possible by using a single integrated circuit. The design of the sensing matrix is based on RIP and incoherence factors described earlier. The above literature is very important on building up the case for the methodology used.

CHAPTER 3. FRAMEWORK AND METHODOLOGY

This chapter provides the framework and methodology to be used in the research study.

3.1 Introduction

This chapter explains the methodology used to perform experiments. It describes the idea behind using these experiments to test reconstruction using compressed sensing. The purpose of this study is to improve the RD architecture by building an efficient signal acquisition system and to design a hardware plaform to test architectures in real time.

3.2 Research Approach

A new RD architecture is proposed in this study. This is done by introducing a post-acquisition randomization step. The methodology used in this study is experiments by which the proposed random demodulator architecture is tested. The proposed architecture of RD is a combination of concepts from analog parallel architecture and digital parallel architecture which were discussed in sections 2.3.1 and 2.3.2. The post-acquisition randomization is introduced in the digital domain. The analog part of the architecture involves single RD. This study emulates proposed RD architecture in software using Matlab. The efficiency of the post-acquisition randomization step is tested by numerical simulations. The proposed RD is tested with signals of various levels of sparsity and different sensing matrices. The following statements relating to the research question are:

- Modifications that can be made to existing architecture that will improve reconstruction quality.
- An efficent trade off between between single RD and M RDs can be made to improve reconstructions.

3.3 Instrumentation

A different RD architecture is proposed by introducing a post acquisition randomization step. The proposed RD combines concepts of analog parallel architecture and digital parallel architecture. The proposed architecture is shown in the Figure 3.1 (Hariharan, Leon-Salas (2014)). The proposed architecture has a single RD followed by a post-acquisition randomization step.

The proposed architecture is tested with different sensing matrices by simulations in Matlab. The simulations use numerical optimization packages such as l1-magic (Candes, Romberg & Tao (2005)), SPG-L1 (Van Den Berg & Friedlander (2011)) and CVX (CVX Research Inc (2011)). These packages are not from Mathworks. These are functions written by researchers in the field of CS. The proposed RD architecture along with the structure of $\omega_{\mathbf{1}}...\omega_{\mathbf{M}}$, and the overall structure of the modified sensing matrix $\mathbf{\Phi}$ are explained in the next chapter along with the simulation results. Continuous testing of each of these matrices for better reconstruction results is done in the initial part of this thesis.

To test various RD architectures on hardware a printed circuit board is designed using FPAAs (Field Programmable Analog Array) from Anadigm and a FPGAs (Field Programmable Gate Array) from Xilinx. This helps in analog data flow into the system and get digitized measurements, which are used in the signal recovery. Four FPAAs are used to have copies of the single RD. Hence there are four RDs in the hardware platform (PCB) fabricated. Digital architecture has very less power consumption. Where as the analog architecture yields better reconstructions. The post-acquisition randomization step (proposed architecture)

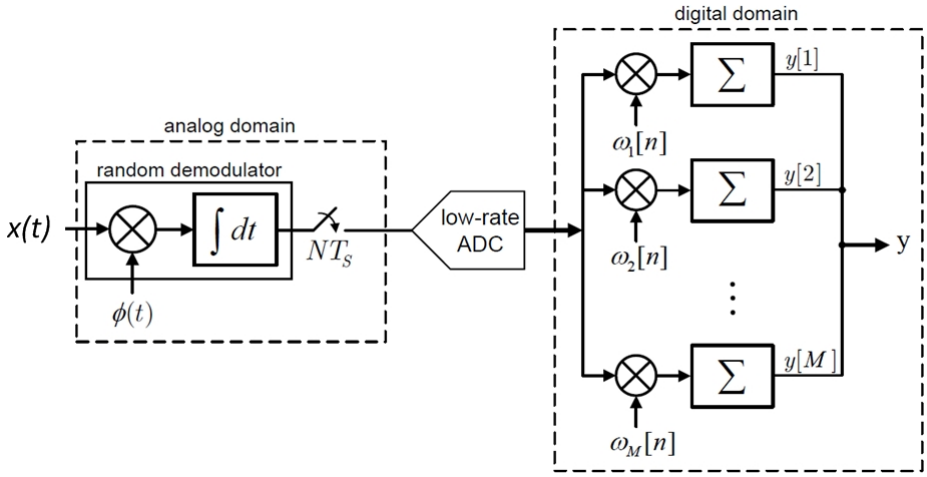

Figure 3.1. Post acquisition randomization step.

will overcome the shortcoming such as circuit complexity or the need for high rate ADC. Every FPAA can act as a single RD in the analog domain. Thus the hardware implementation has four RDs and the digital domain operation are executed through an FPGA. The recovery is done using a mathematical package called SPG L1-magic in Matlab 2013.

3.4 Data Collection

Several kinds of experiments are done using the software simulations. The post acquisition randomization step is tested for different kinds of independent variables of the system which includes

- Percentage of sparsity of the signal given by K
- Sensing matrix $\mathbf{\Phi}$ and its incoherence with basis $\mathbf{\Psi}$
- Number of measurements taken, which is given by M
- Number of RDs in the architecture RD(1,2,3 and 4)

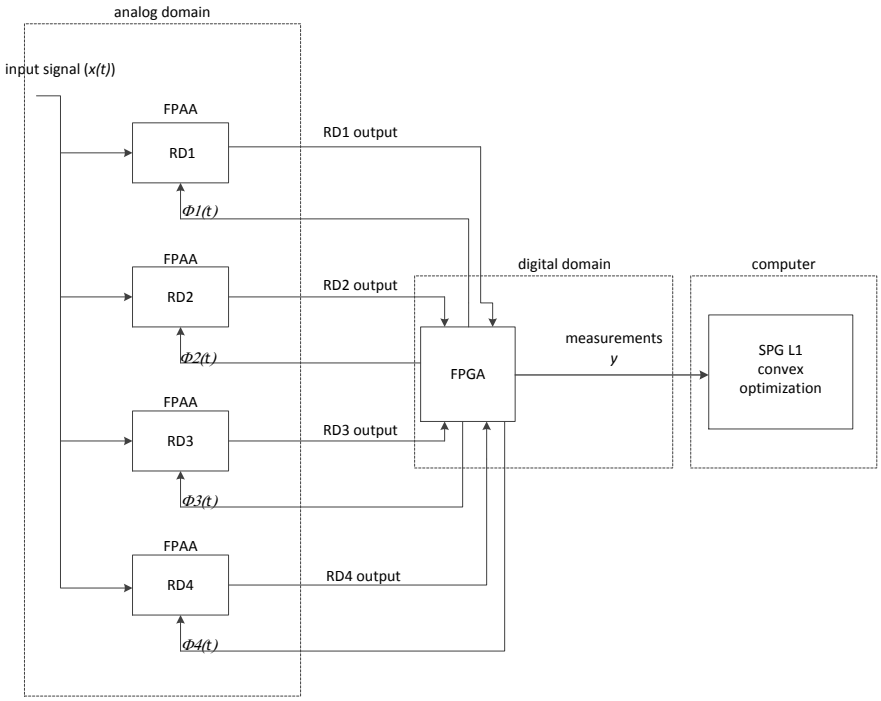

Figure 3.2. Hardware data flow chart.

Every one of these independent variables are varied to improve dependent variable, the quality of reconstruction. Compression ratio determines how much the signal is compressed or the amount of data that is used to reconstruct the signal. A higher compression ratio means greater data compression. A high compression ratio is one of the desirable factors in these experiments. Compression ratio is an indirect dependent variable which is obtained by adjusting the number of measurements taken by the system M, number of RDs and the integration time NT_s. Improving compression ratio is one of the future aims of this work. There are mathematical lemmas which proves the necessary and sufficient conditions for recovery with respect to sparsity (K) of the signal. Those lemmas are kept in mind when performing a new simulation experiment. An extensive amount of software testing in Matlab 2013 is made to find the right parameters for better reconstruction. The hardware platform designed to test new architectures performs the function as

shown in figure 3.2. This is implemented by a printed circuit board designed by the author.

3.5 Data Analysis

Graphical analysis and reconstruction plots are the tool chosen to evaluate experiments for this thesis. A list of different analysis is given below

- Taking N as the length of the spectrum and M as the number of measurements and assuming K the sparsity of the spectrum to be contant, a plot of quality of reconstruction (SNR) versus M/N is drawn.
- Keeping M and N constant, varying K the spectrum sparsity, an analysis of quality of reconstruction is done.
- The above analysis is done for different values of N.
- The simulations are tested for 100 different sensing matrices $\mathbf{\Phi}$.
- The following analysis needs done with measurements from four RDs in future, since the hardware designed can have upto four random demodulators.

3.6 Summary

This chapter provides an overview of the hardware and the software used. This describes what kind of experiments are used to test and the plots by which they are analyzed. The findings of these experiments are presented in the next chapter. Finding right parameters to give better reconstruction requires an extensive amount of experiments and challenges besides the ones mentioned in this chapter. Exact findings and respective experiments are described in detail in the next chapter.

CHAPTER 4. DESIGN, EXPERIMENTS AND RESULTS

This chapter elaborates the system-level experiments performed and the results obtained.

4.1 Introduction

This research work aims to find methods to improve the efficiency and quality of reconstruction using RD. This research work consists of two main parts:

- System-level experiments which simulate the RD arrangement with a post-acquisition randomization step.
- Development of a programmable hardware platform to test different RD architectures.

4.2 System-level Experiments

The single RD is shown in the Figure 4.1 (Hariharan, Leon-Salas (2014)).

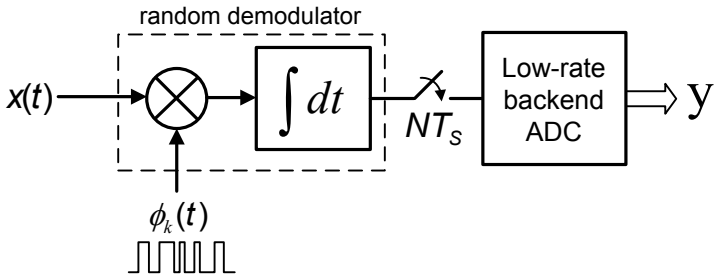

Figure 4.1. Single random demodulator.

The matrix representation of the sensing matrix $\mathbf{\Phi}$ of the single random demodulator is shown in equation 4.1 (Hariharan, Leon-Salas (2014)).

$$\mathbf{\Phi_{RD}} = \begin{bmatrix} \vec{\phi}_1 & \vec{0} & \vec{0} & \cdots & \vec{0} \\ \vec{0} & \vec{\phi}_2 & \vec{0} & \cdots & \vec{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vec{0} & \vec{0} & \vec{0} & \cdots & \vec{\phi}_M \end{bmatrix} \tag{4.1}$$

Considering the signal in time domain to be *x(t)*, the measurements of the single RD can be expressed by the equation below

$$y = \Phi_{RD}(t)x(t) \tag{4.2}$$

The above equation is equivalent of equation 2.3 with $\mathbf{\Phi} = \mathbf{\Phi_{RD}}$. The theory of compressed sensing states that the incoherence between the sensing matrix $\mathbf{\Phi_{RD}}$ and $\mathbf{\Psi}$ should be as high as possible for the reconstructions to be of better quality. The problem with the single RD architecture is that, due to the special structure of the sensing matrix $\mathbf{\Phi_{RD}}$, the incoherence between $\mathbf{\Phi_{RD}}$ and $\mathbf{\Psi}$ is low resulting in low signal reconstruction performance. The proposed solution uses a post-acquisition randomization step that improves the incoherence between the sensing matrix $\mathbf{\Phi}$ and $\mathbf{\Psi}$.

4.2.1 Proposed architecture

The proposed architecture for compressed sensing signal acquisition involves introducing a randomization step in the digital domain. By introducing a randomization step, this architecture increases the randomness of the sensing matrix $\mathbf{\Phi}$. Performing the randomization introduced in the digital domain has the advantage of benefitting from deep-sub-micron scalability. Moreover, this randomization process increases the incoherence between the new sensing matrix $\mathbf{\Phi}$ and the $\mathbf{\Psi}$ which is a discrete cosine transform in this thesis. Figure 4.2 (Hariharan, Leon-Salas (2014)) shows a diagram of the proposed architecture. This proposed

architecture employes a single RD. A single RD reduces the complexity and its power consumption. The randomization step involves the multiplication and accumulation of the RD output with the pseudo-random sequences $\vec{\omega}_k = \left[\omega_k[1], \omega_k[2] \cdots \omega_k[L]\right]$ where, $k = 1, 2 \cdots M$. In this manner, the proposed architecture minimizes the analog complexity and makes use of digital randomization (Hariharan, Leon-Salas (2014)).

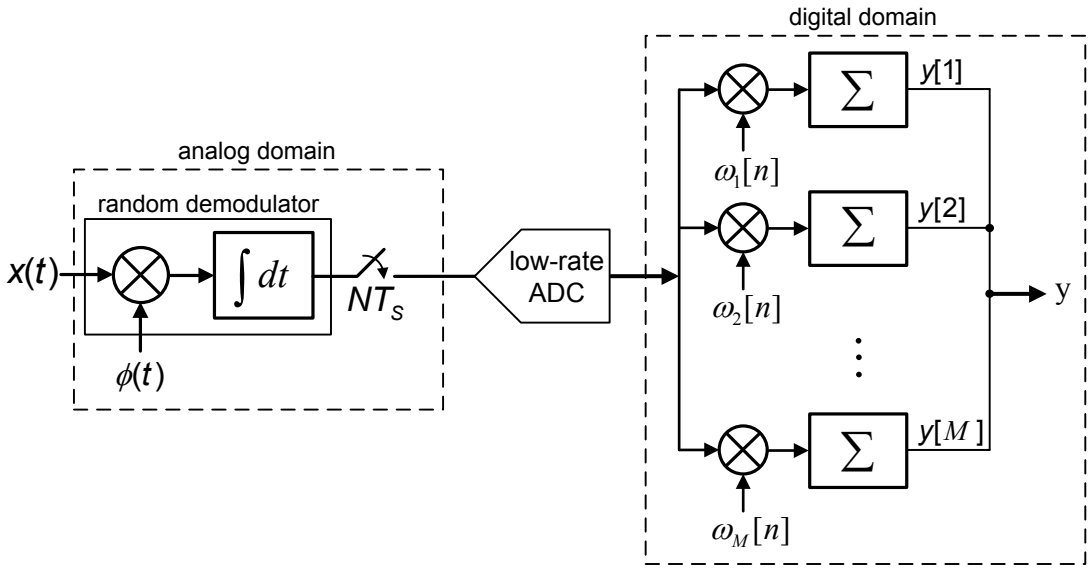

Figure 4.2. Proposed architecture with post-acquisition randomization step.

Compared to the digital parallel architecture (shown in Figure 2.3), the proposed architecture requires an ADC that runs at a rate N times lower. Thus, the ADC employed in this architecture consumes less power than the ADC employed in the digital parallel architecture. The digital circuit of the proposed architecture also runs at a lower rate than the digital circuit in the parallel architecture.

From Figure 4.2 it follows that, in the proposed architecture, the measurements calculation can be expressed as:

$$\mathbf{y} = \mathbf{\Omega}(\mathbf{\Phi_{RD}} \cdot \mathbf{x}) = \mathbf{\Phi} \cdot \mathbf{x} \tag{4.3}$$

where, $\mathbf{\Omega}$ is a matrix whose columns are the pseudo-random sequences $\vec{\omega}_k$. $\mathbf{\Omega}$ is the additional randomization step introduced as shown in the Figure 4.2. Hence, the sensing matrix becomes: $\mathbf{\Phi} = \mathbf{\Omega} \cdot \mathbf{\Phi_{RD}}$.

$$\mathbf{\Omega} = \begin{bmatrix} \omega_1[1] & \omega_2[1] & \omega_3[1] & \cdots & \omega_M[1] \\ \omega_1[2] & \omega_2[2] & \omega_3[2] & \cdots & \omega_M[2] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega_M[L] & \omega_2[L] & \omega_3[L] & \cdots & \omega_M[L] \end{bmatrix} \tag{4.4}$$

The performance of the above architecture is gauged in terms of the SNR of the reconstructed signal and it is compared to the performance of the analog parallel architecture and the single RD case.

Matrices $\mathbf{\Omega}$ that result in a high incoherence between the sensing matrix $\mathbf{\Phi} = \mathbf{\Phi_{RD}\Omega}$ and the matrix $\mathbf{\Psi}$ have been dealt with more interest. This is because high incoherence between $\mathbf{\Psi}$ and $\mathbf{\Phi}$ yields better reconstruction results. Moreover, the matrix $\mathbf{\Omega}$ should require a simple digital circuit to generate it (Hariharan, Leon-Salas (2014)). Another consideration is the dynamic range of the elements of $\mathbf{\Omega}$. The range of elements in $\mathbf{\Omega}$ determines the size of the digital multipliers and accumulators. Limiting the range of these elements will result in reducing the complexity of the digital circuit. Based on these considerations, the elements of $\mathbf{\Omega}$ are limited to uniformly-distributed integer numbers in the range $[-8, 8]$.

Aiming at simplifying the digital circuit even further, a second $\mathbf{\Omega}$ matrix is considered. In the second case, the elements of $\mathbf{\Omega}$ are uniformly-distributed integer numbers in the range $[-8, 8]$ but with the constraint that the absolute value of each element must be a power of 2 including 0, i.e. in the set $\{0, \pm1, \pm2, \pm4, \pm8\}$ (Hariharan, Leon-Salas (2014)). Using power of two integers simplifies the multiplication operation to a simple bit shifting operation. With a 4-bit shift register and just one bit with logic '1', powers of 2 (1,2,4 and 8) can be generated by shifting bits.

4.2.2 Testing the post-acquisition randomization step

Several numerical simulations have been performed to test the performance of the post-acquisition randomization stage. In the simulations, sparse signals were generated using the signal model in equation 2.1. Let K be the number of non-zero components in the spectrum of the sparse signal $\mathbf{a}$, with N being the length of the spectrum or the maximum possible frequencies in its spectrum. It is to be noted that N is also the length of random vectors in $\mathbf{\Phi}$. The location of the K non-zero coefficients in the sparse vector $\mathbf{a}$, is randomly selected. This has been performed using a random permutation function. The placing of these K non-zero components are uniformly distributed in the range $[0, N)$. The orthonormal basis $\{\mathbf{\Psi_n}\}$ was chosen based on the discrete cosine transform (DCT) as follows (Hariharan, Leon-Salas (2014)):

$$\Psi_n[k] = \begin{cases} \sqrt{\frac{1}{MN}} & \text{if} \quad n = 1 \\ \sqrt{\frac{2}{MN}} \cos\left(2\pi(n-1)k\right) & \text{if} \quad n = 2 \ldots N \end{cases} \tag{4.5}$$

From equation 4.5 the number of different frequency components (including DC) in the input signal is N and k varies from 0 to 0.5. With the generated sparse vector $\mathbf{a}$ and the matrix $\mathbf{\Psi}$, the input signal is generated using: $\mathbf{x} = \mathbf{\Psi a}$ (applying equation 2.2), where $\mathbf{x}$ is the signal representation in time domain. The $\mathbf{\Psi}$ matrix being a discrete cosine transform acts as a transformation from frequency domain $\mathbf{a}$ to the time domain signal $\mathbf{x}$.

The generated signal $\mathbf{x}$ was then encoded using the sensing matrix $\mathbf{\Phi}$ and quantized using 8 bits to generate the measurement vector $\mathbf{y}$. Thus, $\mathbf{y} = Q[\mathbf{\Phi s}]$ where $Q[\cdot]$ is the transfer function of the low-rate backend ADC which was modeled as an 8-bit quantizer (Hariharan, Leon-Salas (2014)). The input signals were reconstructed from $\mathbf{y}$ using the BPDN (basis pursuit denoising) reconstruction method which is given by equation 2.13. The equation 2.13 is mentioned again below:

$$\hat{a} = \min \|\hat{a}\|_1 \quad \text{subject to} \quad \|\Theta a - y\|_2 \leq \varepsilon \tag{4.6}$$

where $\hat{\mathbf{a}}$ is the recovered spectrum and $\mathbf{\Theta} = \mathbf{\Phi\Psi}$

4.3 Simulation Results

The following cases are considered in the simulations:

- Analog parallel architecture.
- Single RD.
- Proposed architecture with different $\mathbf{\Omega}$ matrices.

For each one of these cases the average SNR of the reconstructed signal $\tilde{\mathbf{x}}$ was computed as M, N and K were varied. N had two different values fixed to 512 and 1024. K, the sparsity of the spectrum varied from 1 to 8 percent. The average SNR was calculated across 100 signal reconstruction trials for each M and N combination. The SNR in each trial is computed using the equation below:

$$\text{SNR} = 10\log_{10}\frac{\|\mathbf{x}\|_2^2}{\|\mathbf{x}-\tilde{\mathbf{x}}\|_2^2} \tag{4.7}$$

where, $\tilde{\mathbf{x}}$ denotes the recovered signal.

Figures 4.3 to 4.10 show the simulation results (Hariharan, Leon-Salas (2014)). Figure 4.3 and Figure 4.4 shows the average SNR in dB for the analog parallel architecture case for N=1024 and N=512. The standard deviation of the SNR is shown as vertical bars. In case 1 the elements of $\mathbf{\Phi}$ are constrained to +1 and -1 values as it simplifies the analog multiplier used in each random demodulator.

Figure 4.5 and Figure 4.6 shows the average and standard deviation (as vertical bars) of the SNR for the single random demodulator case. In this case, the elements of the matrix $\mathbf{\Phi_{RD}}$ are constrained to +1 and -1 values as it simplifies the analog multiplier used in each random demodulator (Chen, Yu, Hoyos & Martizez (2010)).

Figure 4.7 and Figure 4.8 shows the average and standard deviation of the SNR for the proposed architecture for the case when $\omega_k[n] \in [-8, 8]$. Figure 4.9 and

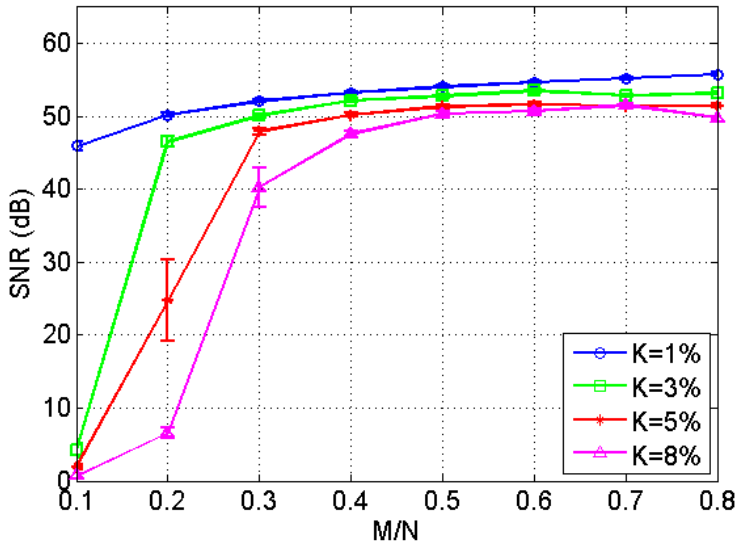

Figure 4.3. SNR of reconstructed signal for the parallel compressive sensing hardware architecture for N=1024.

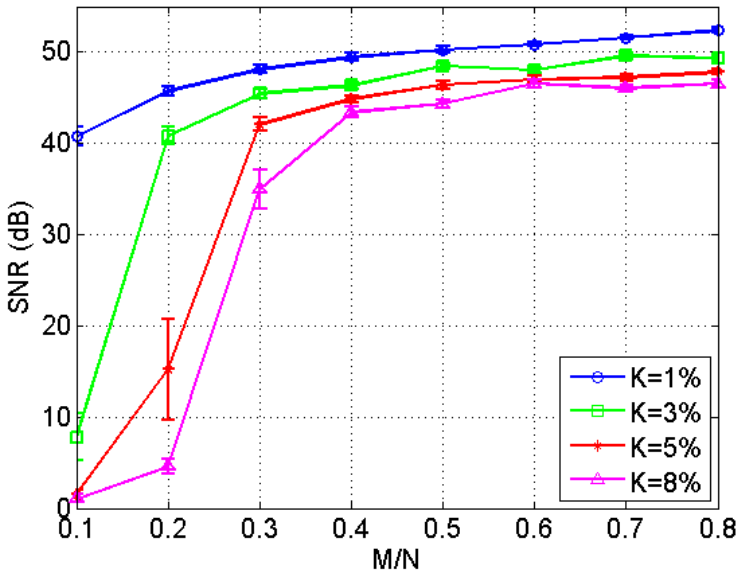

Figure 4.4. SNR of reconstructed signal for the parallel compressive sensing hardware architecture for N=512.

Figure 4.10 shows the average and standard deviation of the SNR for the proposed architecture for the case when $\omega_k[n] \in \{0, \pm 1, \pm 2, \pm 4, \pm 8\}$.

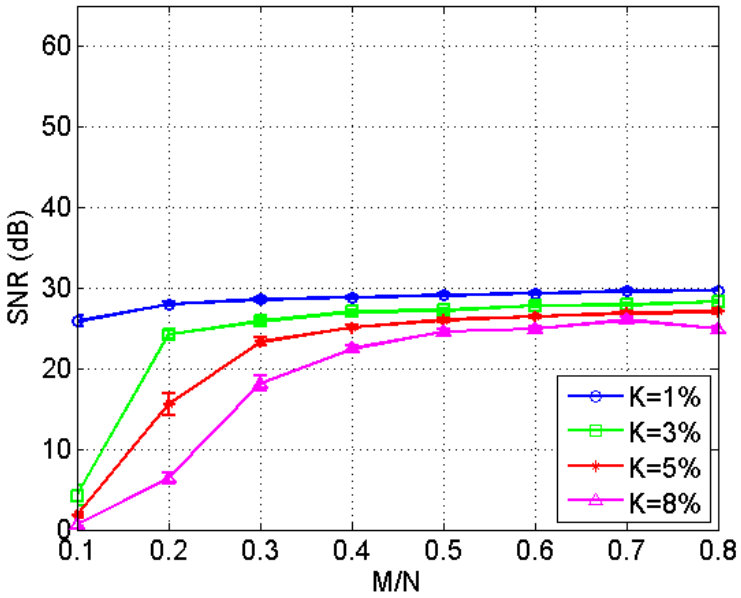

Figure 4.5. SNR of reconstructed signal for the single random demodulator hardware architecture for N=1024.

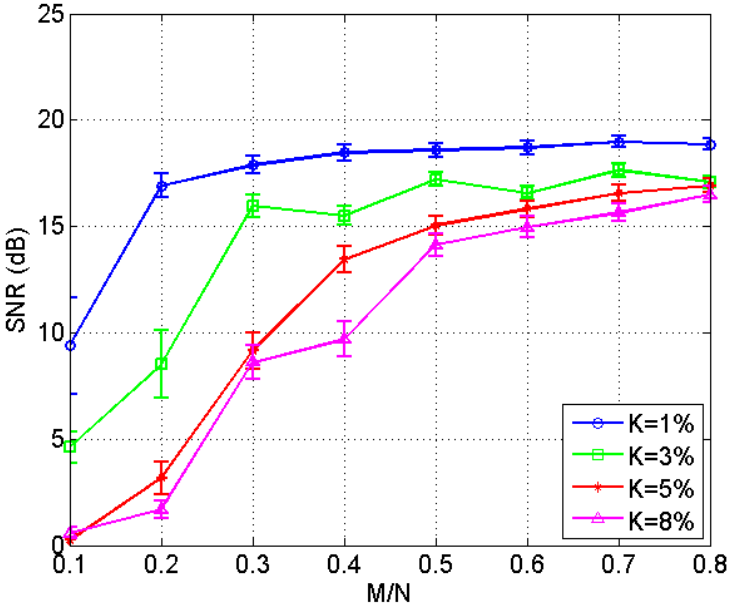

Figure 4.6. SNR of reconstructed signal for the single random demodulator hardware architecture for N=512.

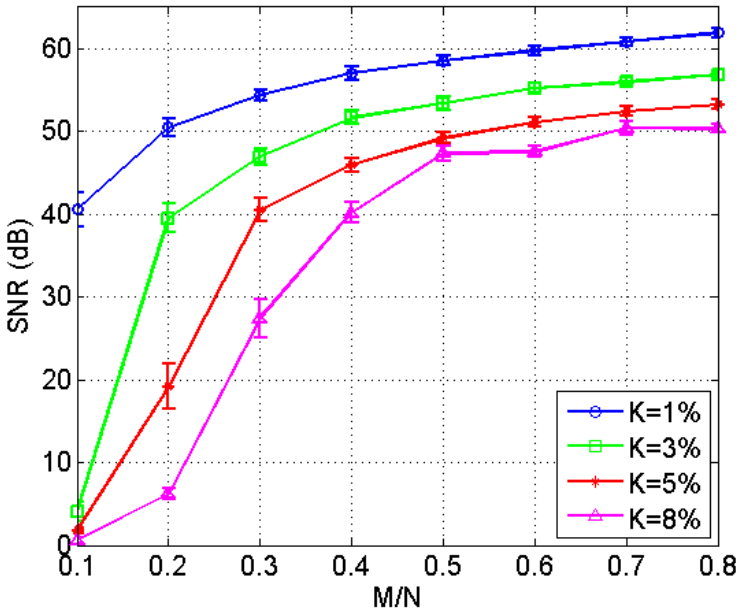

Figure 4.7. SNR of reconstructed signal for the proposed architecture when $\omega_k[n] \in [-8, 8]$ for N=1024.

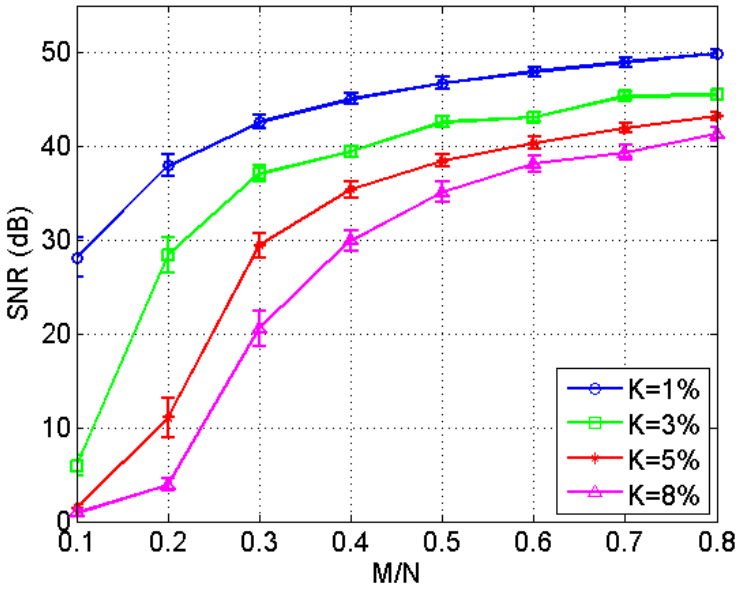

Figure 4.8. SNR of reconstructed signal for the proposed architecture when $\omega_k[n] \in [-8, 8]$ for N=512.

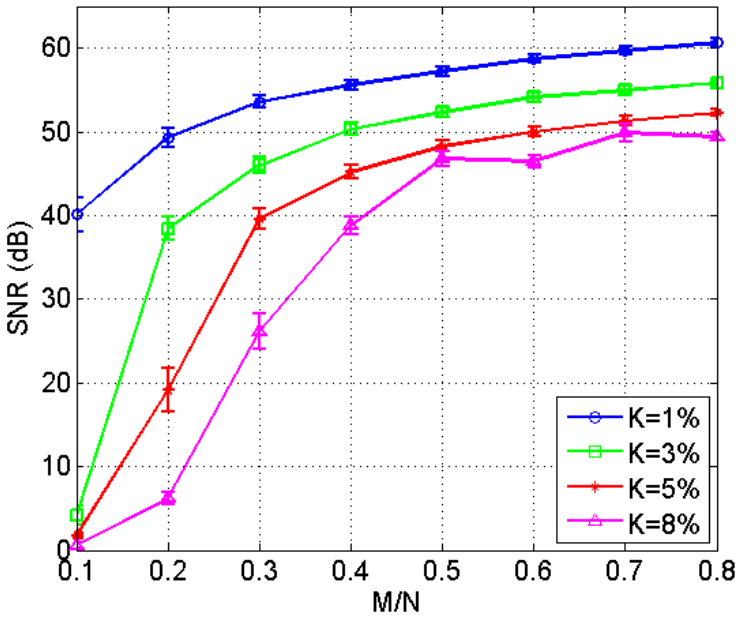

Figure 4.9. SNR of reconstructed signal for the proposed architecture when $\omega_k[n] \in \{0, \pm 1, \pm 2, \pm 4, \pm 8\}$ for N=1024.

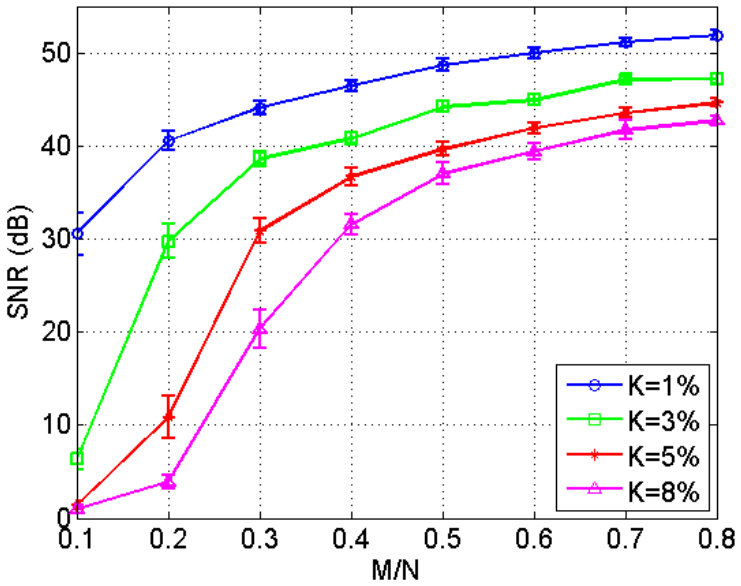

Figure 4.10. SNR of reconstructed signal for the proposed architecture when $\omega_k[n] \in \{0, \pm 1, \pm 2, \pm 4, \pm 8\}$ for N=512.

Notably, the performance of the single RD is very poor compared to the analog parallel architecture due to the low incoherence between $\mathbf{\Phi_{RD}}$ and $\mathbf{\Psi}$. Although the proposed architecture uses a single RD, its performance is improved by the post-acquisition randomization step performed in the digital domain. The simulation results also show that restricting the elements of $\mathbf{\Omega}$ to be powers of two has a very small impact on reconstruction performance. Hence, the complexity of the digital circuit can be reduced by replacing the multipliers with bit shifting operations (Hariharan, Leon-Salas (2014)).

4.4 Board Design and Hardware Architecture

The design of a hardware for RD architecture involves analog components like multipliers and integrators. There is need for analog multipliers and integrators to perform the matrix multiplication operation. A low-rate ADC is also required to digitize these measurements. A sample and hold circuit is needed to hold the input of the ADC which digitizes the measurements produced by the RD.

The hardware design of the RD architecture has been done with a printed circuit board (PCB) which uses field programmable gate arrays (FPGAs) and field programmable analog arrays (FPAAs). Figure 4.11 shows the image of the designed board. It has four FPAAs, each of which can be programmed to act as a RD. The FPGA on the board is used to generate random numbers for the sensing matrix $\mathbf{\Phi}$, collect and transmit measurements to the computer. The compressed sensing convex optimization solver in Matlab is used to recover the original signal from the measurements taken using the designed board. There are a number of compressed sensing solvers which work with Matlab. L1-Magic , SPG L1 and CVX are some of the numerical optimization packages which minimize l1-norm using convex optimization technique to arrive at a solution. The CS solvers implement the optimization problem given by 2.13. SPG-L1 is used to recover hardware measurements because it is more robust to noise.

4.4.1 Components on the board

A photograph of the fabricated board is shown in Figure 4.11. Power connection is present at the bottom right corner of the board. The LDOs (Low Dropout Regulator) supplying power to the components are placed in the right end of the board. The FPGA on the PCB is programmed using the JTAG programmer. JTAG programming socket can be found right next to the power supply. The board has a 3-pin serial port which can be used to transmit and receive information onto the serial port in the computer via UART. There four LEDs wired to the FPGA which can be used for indication. There is also an FPGA-done and a FPAA-done LED that are dedicated to indicate the successful programming of the FPGA and FPAA respectively. There are 8 DIP switches wired to the FPGA. These DIP switches can be used as FPGA inputs. There are a number of FPGA I/O pins connected to the headers. A header is present on the left center of the board. The pins connected to the header can be used for troubleshooting.

There are 3 push buttons on the board. They are FPGA reset, FPAA reset and a general purpose pushbutton. FPGA reset is used to erase the programmed data in FPGA and FPAA reset is used to erase programmed data in FPAA. The third push button is wired to an I/O pin in FPGA and can be used as a reset to the executing program. The board has number of level shifters to convert voltages from 3.3V to 5V and 5V to 3.3V. This is because the FPAA I/O pins require and respond with 5V, whereas the I/O pins in the FPGA has a maximum limit of 3.3V. The level shifters are not indicated in the image of the board.

The following sections will explain the design and working of the components of the board in detail. Explained first, is the designing of RD using FPAA. Figure 4.11 shows an image of the board designed to program and test random demodulator hardware.

There is an 8-pin header on the top center of the board. These pins are called daisy chaining pins. Daisy chaining pins are use to program the FPAAs on

this board from another FPAA. A brief description of the function of each of these pins are given below.

- ERRb is an open drain bidirectional pin. It indicates the proper programming of the FPAAs.
- PORb is power on reset pin. This is connected to the FPAA reset push button. It is used to erase programmed data in FPAA.
- D_IN or data in pin of the FPAA. Data to be programmed in the FPAA is sent through this pin.
- DCLK is the digital clock which synchronizes along with the data in pin to program the FPAA.
- ACLK is the analog clock source of the FPAA. ACLK can have a maximum value of 40 MHz.
- CFGFLGb is the configuration flag. In a multi FPAA system all CFGFLGb are tied together. It goes low during configuration.
- ACT or the activate is an open drain bidirectional pin. In a multi FPAA system all ACT are tied together. This pin does not allow the FPAA go active until all the FPAAs in the system are programmed.
- EXE or execute pin is grounded. In multi FPAA system all EXE pins are tied together.

4.4.2 Design of RD using FPAA

Anadigm Inc. manufactures field programmable analog arrays (FPAA). These are integrated circuits which have four configurable analog blocks (CAB). CABs in an FPAA can be programmed as specific analog modules using a graphical user interface software called Anadigm Designer. The availability of analog circuitry

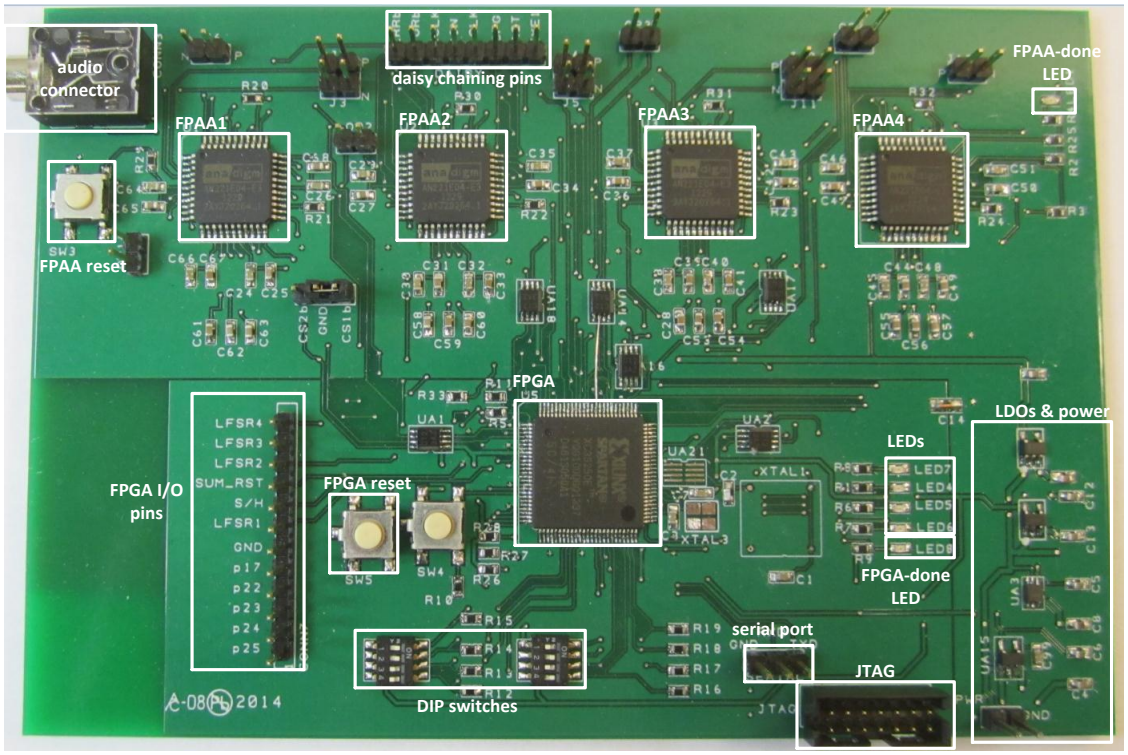

Figure 4.11. Image of the printed circuit board designed.

depends on the capability of a particular FPAA. The AN221E04 FPAA, was used to design RD arrangement in hardware in this thesis. Anadigm Designer 2.7.0 is the version of the software used to program AN221E04 FPAA in this work. Figure 4.12 shows the RD circuit designed using the Anadigm Designer 2.7.0. Figure 4.13 shows the operation of the circuits as a flow diagram.

4.4.3 Working of circuit components in RD

The random number sequence for the sensing matrix $\mathbf{\Phi}$ is generated using the FPGA on the board. Spartan-3E (XCS250E) FPGA from Xilinx is used in the board. Linear feedback shift register (LFSR) is a method by which pseudo-random sequences of 0's and 1's can be generated digitally. A LFSR can be created by a

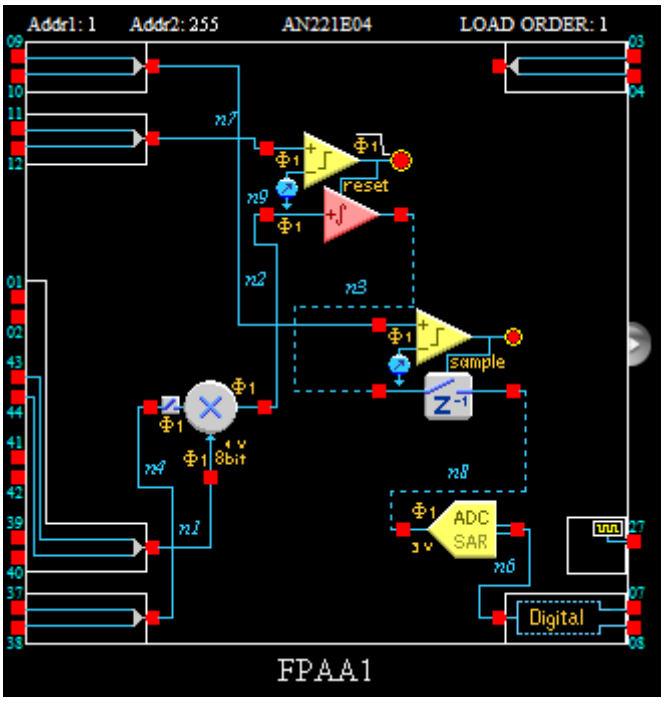

Figure 4.12. RD in Anadigm Designer software.

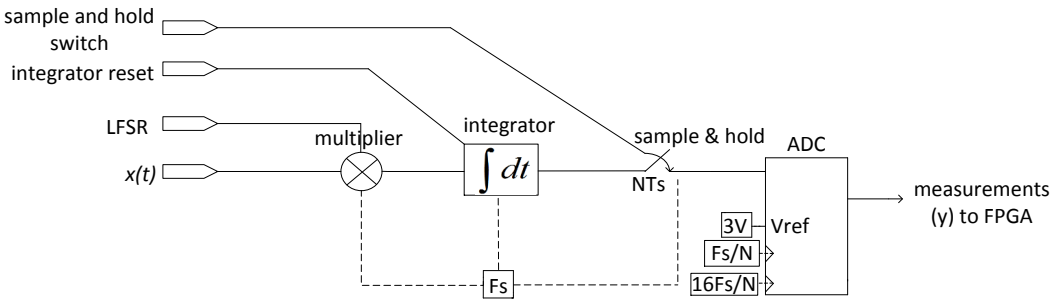

Figure 4.13. RD flow diagram.

shift register and EXOR operation in a feedback fashion. This method is employed in this circuit. FPGA is programmed using VHDL (VLSI hardware description language) to produce LFSR random sequences and hence the sensing matrix $\mathbf{\Phi}$ for the RD. Four different LFSRs are used in this study. These four LFSRs generate the sensing matrix $\mathbf{\Phi}$ for the hardware. LFSRs are represented by the length of the shift register used to create them and the positioning of the bits from which EXORs for a

particular LFSR are tapped. LFSR with n-bit shift register can produce a maximum sequence of length of $2^n - 1$. The bits from which EXOR operation is performed is called taps. Below mentioned are the exact details of the LFSR used in this study

- LFSR 1 : 16-bit LFSR, with taps positioned at bits 4, 5, 6, 8, 10, 11, 14 and 16.
- LFSR 2 : 16-bit LFSR, with taps positioned at bits 11, 13, 14 and 16.
- LFSR 3 : 17-bit LFSR, with taps positioned at bits 14 and 17.
- LFSR 4 : 18-bit LFSR, with taps positioned at bits 11 and 18.

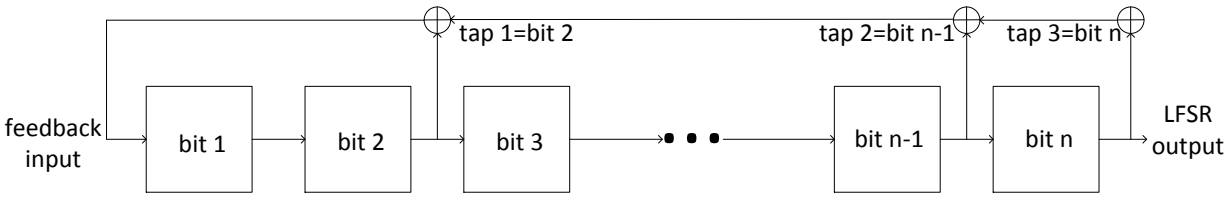

Figure 4.14. Flow diagram of a n-bit LFSR

Figure 4.14 shows the working of a LFSR with the feedback fashion (Clive (1957)). The position of EXOR taps shown are just a depiction. The exact LFSRs with the position of taps has been mentioned earlier. The Figure 4.14 shows the position of input and output bits in a LFSR.

Figure 4.15 shows the timing diagram, depicting the operations in RD. The random sequence generated by the LFSR gets multiplied with the input signal $x(t)$. The product then gets accumulated by an integrator which sums upto N number of samples (depending on the structure of the sensing matrix $\mathbf{\Phi}$) to generate a measurement. Then these integrated samples are held by a sample and hold circuit. This holding operation stabilizes the input to the ADC. Finally, these voltages are

digitized by the ADC. These digitized measurements are then collected by the FPGA. FPGA transmits these measurements to the computer via UART (universal asynchronous receiver transmitter).

Figure 4.15 shows the integrator reset pulse and sample and hold switch. Let N be the size of each vector in $\mathbf{\Phi_{RD}}$. Then N samples of the input signal is multiplied with the random sequence and then gets integrated. It can be observed from Figure 4.15 that, the sample and hold switch goes high at every Nth cycle and the integrator gets reset at $(N+1)$ cycle of the clock. This corresponds to the samples being mutiplied and integrated for every N samples. Thus, this operation makes matrix multiplication possible with a multiplier and integrator. All of the operations which include mutiplication, integration, sample and hold gets executed at the sampling rate F_s which is user defined in the FPAA. A sample LFSR, integrator output and sample and hold output are shown in Figure 4.15.

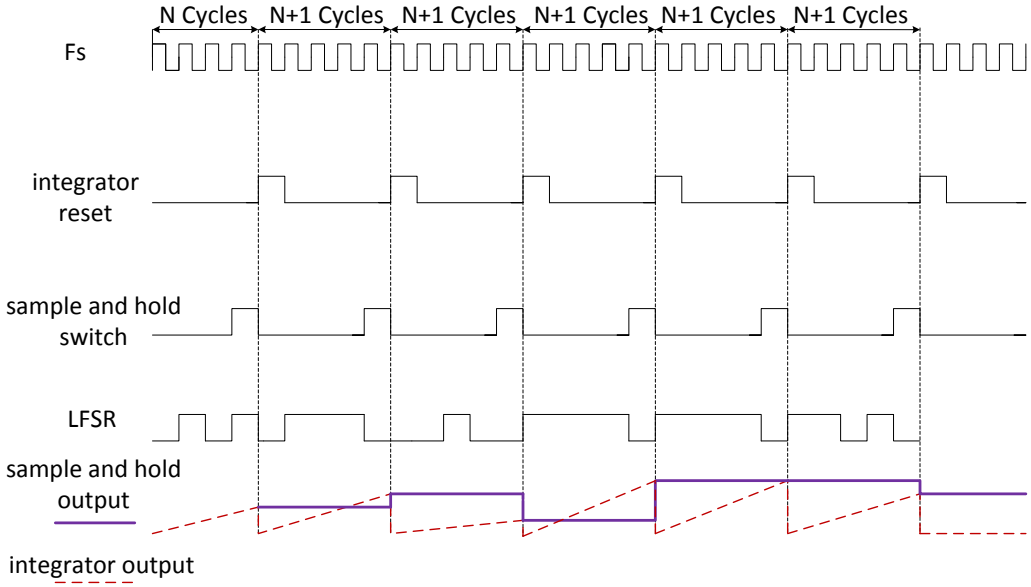

Figure 4.15. Integrator, sample and hold timing diagram

4.4.3.1. ADC operation and timing diagram

The ADC in the FPAA is a successive approximation (SAR) 8-bit ADC. The ADC's MSB (most significant bit) acts as a sign bit in the FPAA. The SAR ADC in the FPAA converts at the rate of F_s/N that is N times lesser than the rate at which multiplication or integration happens in the random demodulator. Figure 4.16 shows the timing diagram and the exact sequence of events in the conversion process. There is a **sync** output to the ADC which identifies the 8-output bits of the ADC. During the positive half cycle of sync, each output bit of the ADC appears at the rate of $16F_s/N$. During the negative half cycle of the sync, the output of the ADC is not valid. While the multiplier and integrator are producing measurements, the ADC digitizes the previous measurement. In other words when ADC is converting one measurement, the multiplier and the integrator keeps accumulating the next sample of input data.

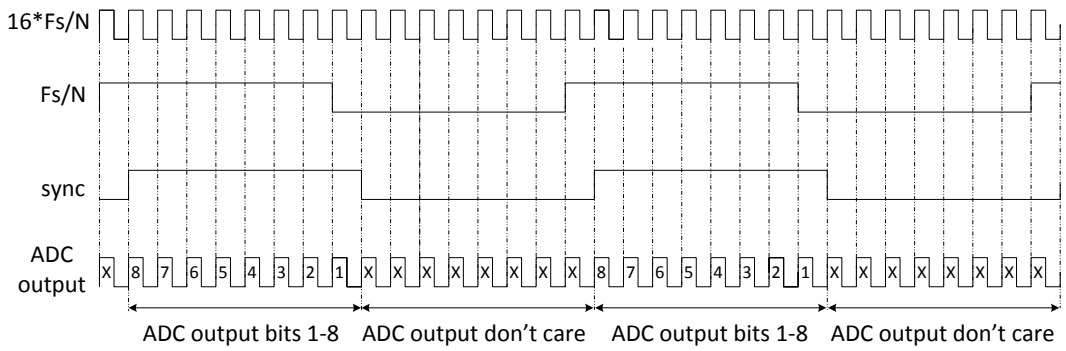

Figure 4.16. ADC output timing diagram.

4.4.4 Function of FPGA on the board

The Spartan-3E (XCS250E) FPGA, from Xilinx is used in this board. This is a 100-pin VQG packaged chip from Xilinx. This FPGA has internal logic and

memory to support operations of 4 RDs. These operations performed by the FPGA include:

- Generation of a random sequence (0s and 1s) to be used as the sensing matrix $\mathbf{\Phi}$ or the LFSR output in digital terms.
- Generation of an integrator reset pulse and the sample and hold switch at the rate of F_s/N.
- Obtain the measurements from the RD. The measurements are the output of the ADC explained in the previous section.
- Transmit the RD measurements to the computer via UART.
- It is to be noted that all these operation take place in a different speed or clock.
- Pseudo-random sequences are generated at F_s.
- ADC output bits are received using a shift register at $16F_s/N$.
- The measurements obtained are transmitted at a baud rate which is user defined in the VHDL code.
- FPGA is also used to program the FPAA with the random demodulator circuit. This is explained in the next section.

4.4.5 Modified architecture and hardware wiring

The design of RD using FPAA has been fairly simplified given the complexity of the circuit. For parallel architectures it is a known fact that the SNR of the recovered signal increases with increasing the number of RDs. Hence, this board was designed with four FPAAs which will indeed increase the number of RDs to four. Hence the modified architecture will look like the one in Figure 4.17. Thus

the board has four FPAAs each capable of acting as a RD and an FPGA which acts a controller by generating random sequences and transmitting measurements to be recovered in the computer.

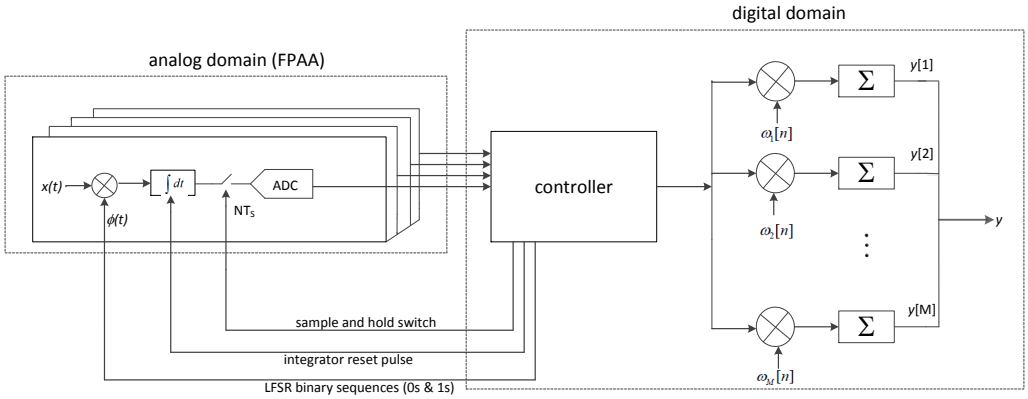

Figure 4.17. Hardware architecture designed.

The wiring of certain important connections is shown in Figure 4.18. FPGA in this board also acts as a programmer to the FPAA. The Anadigm Designer software provided by Anadigm generates a bit stream of data according to the graphical circuit to program the FPAA. This bitstream is created in the format .ahf. .ahf files are a collection of 8 bit data in hexadecimal format. The bit stream of data to program the FPAA can be converted onto a file format that is compatible with the FPGA compilers. Files of format .mif is used by Altera FPGAs, while .coe is a format used by Xilinx FPGAs. Thus the FPGA on this board can also be used as a programmer to program the FPAAs. In the Figure 4.18, D_IN is wired from the FPGA to FPAA and is used to program the FPAA. ACLK is a clock output from the FPGA and provides clock for the analog blocks in the FPAA. All internal executions in the FPAA are based on this clock. This clock is further divided by the FPAA for various operations to be used in the circuit programmed within the FPAA.

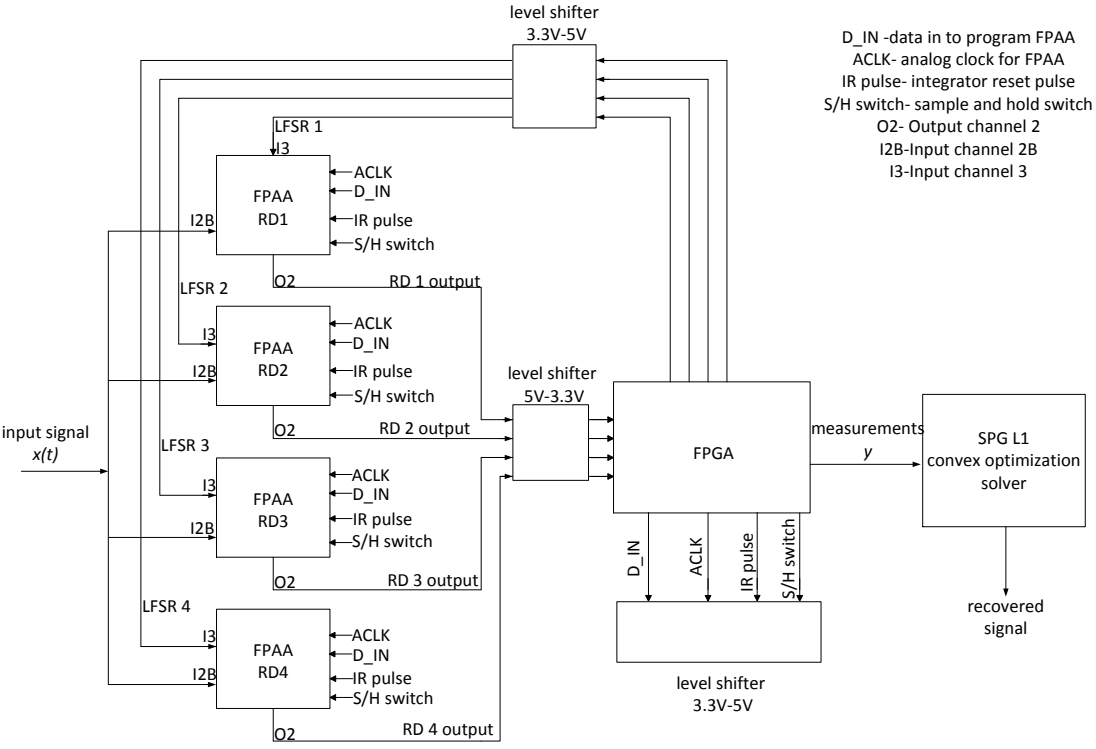

Figure 4.18. Connections in the PCB.

4.4.6 Structure of sensing matrix $\mathbf{\Phi}$ for multiple random demodulators

The fabricated board has four FPAAs which work as RDs. The structure of the sensing matrix for this compressed sensing signal acquisition system is different from that of single RD. To generate the sensing matrix for recovery, the same logic that is used to generate the LFSR, is used in Matlab. The structure of this matrix needs to be analyzed for correct reproduction and hence the recovery. The sensing matrix has the same structure like the one in equation 4.8. The length of each vector in $\mathbf{\Phi}$ is N. N is directly proportional to the compression ratio of the signal acquisition system. Compression is the rate at which data is compressed to form measurements. A higher compression ratio with the same reconstruction quality means more data is compressed more effectively. Equation 4.8 is the modified structure of sensing matrix $\mathbf{\Phi}$ for four RDs.

$$\mathbf{\Phi} = \begin{bmatrix} \vec{\phi}_{1,1} & \vec{0} & \vec{0} & \cdots & \vec{0} \\ \vec{\phi}_{2,1} & \vec{0} & \vec{0} & \cdots & \vec{0} \\ \vec{\phi}_{3,1} & \vec{0} & \vec{0} & \cdots & \vec{0} \\ \vec{\phi}_{4,1} & \vec{0} & \vec{0} & \cdots & \vec{0} \\ \vec{0} & \vec{\phi}_{1,2} & \vec{0} & \cdots & \vec{0} \\ \vec{0} & \vec{\phi}_{2,2} & \vec{0} & \cdots & \vec{0} \\ \vec{0} & \vec{\phi}_{3,2} & \vec{0} & \cdots & \vec{0} \\ \vec{0} & \vec{\phi}_{4,2} & \vec{0} & \cdots & \vec{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vec{0} & \vec{0} & \vec{0} & \cdots & \vec{\phi}_{1,M} \\ \vec{0} & \vec{0} & \vec{0} & \cdots & \vec{\phi}_{2,M} \\ \vec{0} & \vec{0} & \vec{0} & \cdots & \vec{\phi}_{3,M} \\ \vec{0} & \vec{0} & \vec{0} & \cdots & \vec{\phi}_{4,M} \end{bmatrix} \tag{4.8}$$

4.5 Hardware Results

This section shows the reconstructed signal measurements acquired by the hardware. The Figure 4.19 shows the actual and ideal measurements for a constant voltage input. It is observed from Figure 4.19 that the actual and ideal measurements coincide with each other. This proves the correct operation of the RD in hardware. Measurement error can also be observed from the Figure 4.19. Measurment errors affect the reconstruction quality. Figures 4.20 and 4.21 show the orginal and reconstructed waveform for a single sinewave of frequency 15 Hz. There is a glitch in the reconstructed waveform. This is due to the discontinuity when playing back the original waveform. This hardware reconstruction shows that there can be further improvements made. The length of the random vector N in the sensing matrix $\mathbf{\Phi}$ was kept 16 in the following recontruction shown in Figure 4.21. In order to improve the quality of measurements variation in different parameters has to be made. N, the lengh of the random vectors in $\mathbf{\Phi}$ plays a vital role in

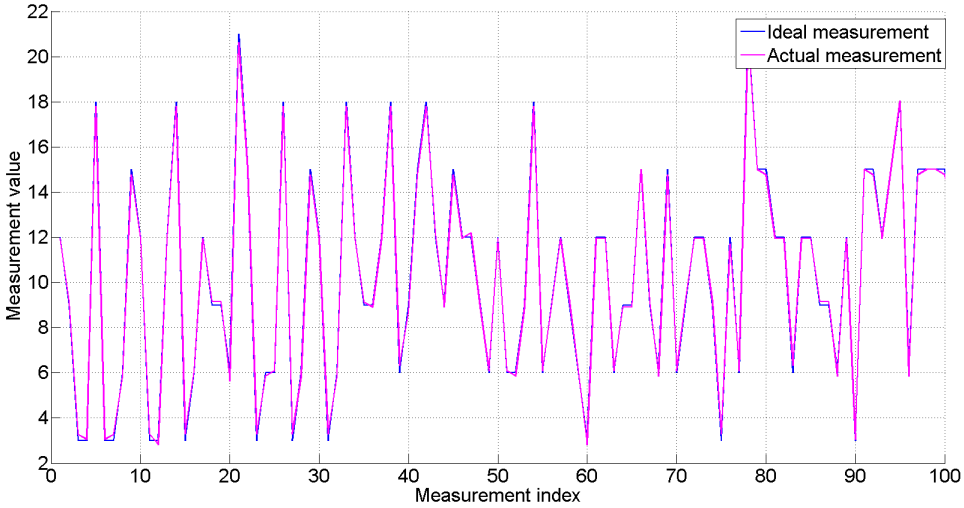

Figure 4.19. Ideal and actual measurements for a contant input of 3V.

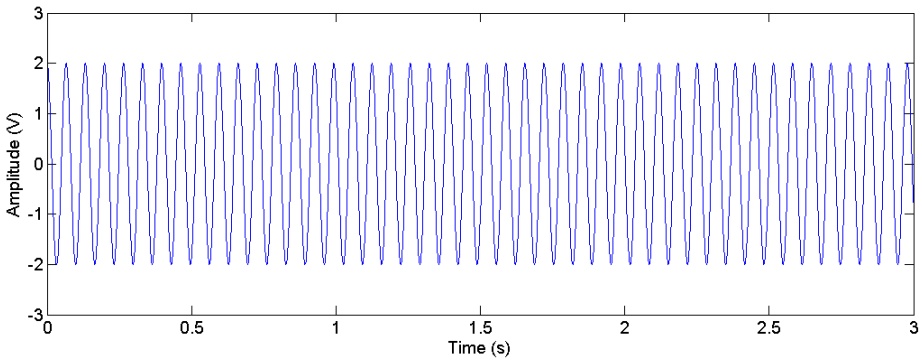

Figure 4.20. Original waveform for a single sinewave with N=16 and M=256.

reconstruction quality. Higher values of N results in saturation of the integrator. This is explained in detail in the section 4.6.

4.6 RD using Sigma-Delta Modulator

The hardware reconstructions using the RD arrangement can be further improved. The issue with the current implementation is that the integrator output in the hardware saturates for large values of N. To prevent saturation of the

integrator, an attenuated signal has been fed to the input. As the signal is been attenuated, the SNR at the input decreases resulting in poor signal reconstruction. Hence, the reconstructions need to be improved by, neither letting the integrator saturate nor allowing SNR to decrease along with signal attenuation. To overcome these limitations, a RD arrangement with a sigma-delta modulator can be used. The Figure 4.22 shows the integrator operation of a RD using a sigma-delta modulator. The inverting input to the integrator is fed with a voltage $Vref$ when the integrator output reaches a voltage $Vref$. A counter is used to count the number of times the integrator output reaches $Vref$. When the integrator is fed with $Vref$ to the inverting input, its output is decreased by $Vref$. Thus saturation of the integrator is avoided and hence larger values of N can be used to acquire the signal. Figures 4.23 and 4.24 show the design of a sigma-delta modulator as a block diagram and in Anadigm Designer software which can be implemented in FPAA. Figure 4.24 does not implement the counter shown in Figure 4.23. This is because counter being a digital circuit is programmed in the FPGA on the board.

In Figure 4.24, gain with switchable inputs CAM acts as a multiplier. The G1 and G2 of this CAM is 1 and -1. The multiplier output is fed to the integrator. As the integrator output reaches or overshoots above $Vref$ it is fed with a voltage $Vref$ to its inverting input. Another CAM programmed as an amplifier with switchable inputs is used to feed a $Vref$ to the inverting input of the integrator. When the integrator output is below $Vref$ this CAM feeds 0V. The values of gain in the CAM which feeds $Vref$ to the integrator in the circuit shown are G1=1, G2=0.01 and G=0.01 where G is the gain of another amplifier as shown in Figure 4.24. G2 and G are made 0.01 to make the voltage fed to the inverting input of the integrator close to zero when its output is less than $Vref$. As the integrator output reaches $Vref$ this CAM feeds $Vref$ to the inverting input of the integrator and thus reduces the output by $Vref$.

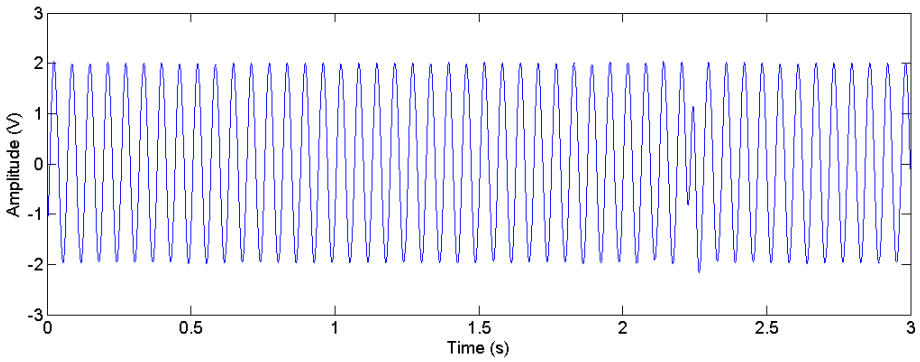

Figure 4.21. Typical reconstruction for a single sinewave with N=16 and M=256.

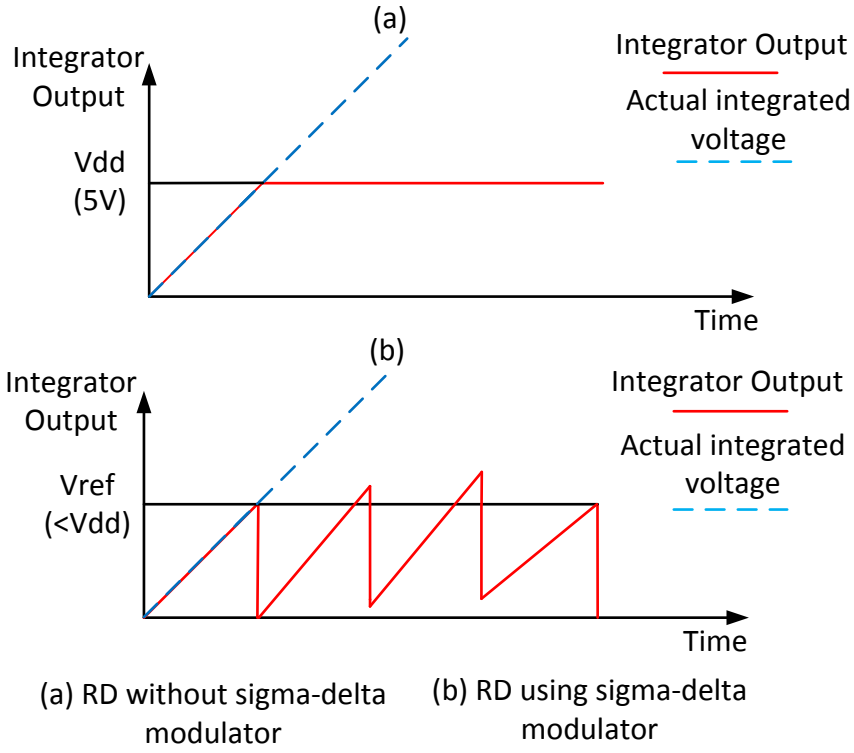

Figure 4.22. Integrator operation with and without sigma-delta modulator.

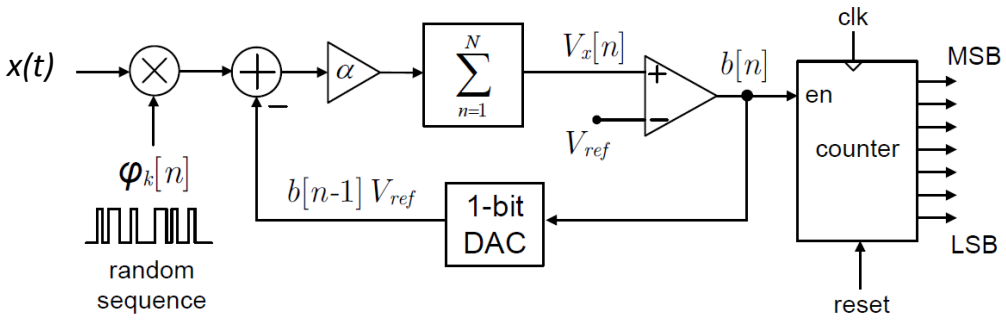

Figure 4.23. Block diagram of a RD using sigma-delta modulator.

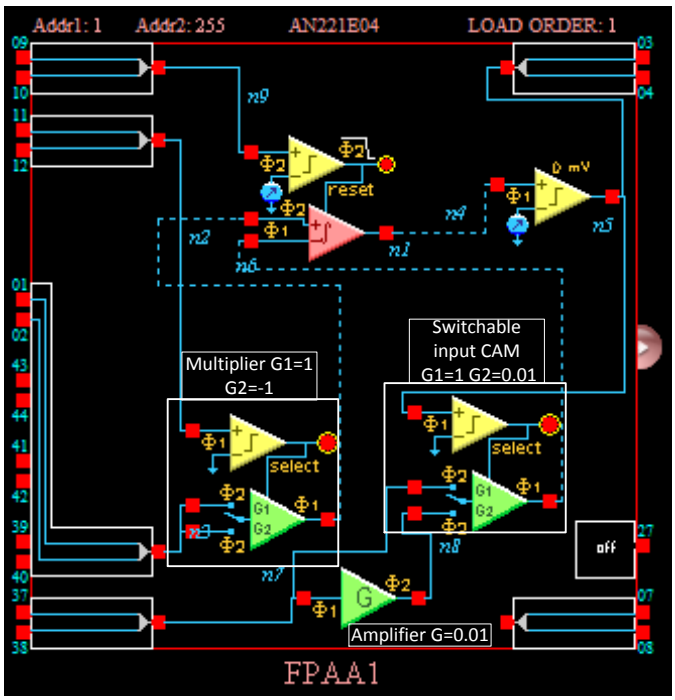

Figure 4.24. RD using sigma-delta modulator in Anadigm designer 2.7.0.

CHAPTER 5. SUMMARY, CONCLUSION AND FUTURE ENHANCEMENT

This chapter summarizes the important conclusions that can be made from this work. The aim of this work is to find a better and efficient RD architecture. A programmable and flexible hardware platform has also been developed. The first part of this research involved system-level simulations which evaluated a new architecture that included a post acquisition randomization step. The second part of this research included the development of a hardware implementation of a RD by designing a printed circuit board. Both parts of this project examines the working of the RD architecture to reconstruct signals using compressed sensing technique.

5.1 Summary and Conclusion from System-level Simulations

The system-level simulation tests RD by introducing a post-acquisition randomization step. Different cases which include parallel architectures, single RD and RD with post-acquisition randomization step have been evaluated. Single RD with additional randomization stage has been found to work better than just a single RD arrangement. The post-acquisition randomization step, with elements restricted to powers of 2 has been selected since it can be implemented using a simple digital circuit. It has been observed that with increase in the sparsity of the signal K the quality of reconstruction (SNR) decreased. This simulation analysis also leads to an architecture with four RDs. The architecture with four RDs is designed in hardware.

5.2 Summary and Conclusion from Hardware Reconstructions

Hardware design involves design of a printed circuit board with FPAAs and FPGAs. Each FPAA in the board is programmed as a RD. There are four FPAAs,

hence there are four RDs in the hardware designed. Measurements have been taken using the developed board and they are recovered by compressed sensing solver in the computer. The flexibility of the board is the important point to be noted here. This board can be used to program, analog and digital circuitry with ease. Measurements from the board prove the correct operation of the RD in hardware. These measurements also lead to analysis on factors affecting reconstructions.

5.3 Future Enhancement

This project on RD can be further experimented in the following ways:

The additional randomization stage tested in the software simulations can be implemented on the FPGA. Acquisition of signals using RD with sigma-delta modulator can be implemented.

Look up table (LUT) in the FPAA controls the output of the ADC. Since the LUT can be reprogrammed, four 8-bit LUTs can be programmed to acts as a 10 bit ADC which will result in a much lesser quantization error. However implementing a 10-bit ADC will result in having one less RD in the hardware arrangement.

The hardware results can be evaluated in terms of SNR. The original and reconstructed waveform need to be matched point by point to find the exact quality of reconstruction. Real time signals such as audio can be reconstructed by this hardware.

LIST OF REFERENCES

LIST OF REFERENCES

Candes,J.E., Romberg, J., Terence, T. (2005). Robust Uncertainty Principles: exact signal reconstruction from highly incomplete frequency information. *IEEE transactions on information theory, 52(2)*, 489-509.

Candes,J.E., Wakin, M.B. (2008). An introduction to compressive sampling. *IEEE signal processing magazine, 25(2)*, 21-29.

Chen, Xi., Yu, Z., Hoyos, S., Sadler, B.M., Martinez, J.S. (2010). A sub-nyquist rate sampling receiver exploiting compressive sensing. *IEEE transactions on circuits and systems, (1)* 507-519.

Chen, F., Chandrakasan, A.P., Stojanovic, V.M. (2012). Design and analysis of a hardware effiencient compressed sensing architecture for data compression in wireless sensors. *IEEE journal of solid-state circuits, 47(3)*, 744-756.

Clive, M. (1957). *Bebop to the boolean boogie : an unconventional guide to electronics*. Burlington, MA: Elsevier Press.

CVX Research, Inc. (2012). CVX: Matlab software for disciplined convex programming, version 2.0. *http://cvxr.com/cvx*.

Foucart, S., Rauhut, H. (2013). *A mathematical introduction to compressive sensing*. Newyork, NY: Springer science & business media.

Goyal,V.K., Fletcher, A.K., Rangan, S. (2008). Compressive sampling and lossy compression. *IEEE signal processing magazine, 25(2)*, 48-56.

Hariharan, S., Leon-Salas, W.D. (2014). Single random demodulator architecture for compressive sensing signal acquisition (under print). *INTERCON International conference on electrical, electronics engineering and computer science*.

Harms, A., Bajwa, W.U., Calderbank, R. (2013). A constrained random demodulator for sub-nyquist sampling. *IEEE transactions on signal processing, 61(3)*, 707-723.

Kirolos, S., Ragheb, T., Laska, J.N., Massoud, Y., Baranuik, R.G. (2006). Practical issues in implementing analog to information converters. VI^{th} *international workshop on system on chip for real time applications*.

Laska, J.N., Kiriolos, S., Duarte, M.F., Ragheb, T., Baranuik, R.G., Massoud, Y. (2007). Theory and implementation of an analog to information converter using random demodulation. *IEEE international symposium on circuits and systems(ISCAS)*.

Mishali, M., Eldar, Y.C., Dounaevsky, O., Shoshan, E. (2010). Xampling : analog to digital at sub-nyquist rates. *IET journal for cicuits devices and systems, 5(1)*, 8-20.

Oppenheim, A.V., Schafer, R.W., Buck, J.R. (1999). *Discrete time signal processing.* Eaglewood cliffs,NJ: Prentice-Hall.

Ragheb,Kirolos, S., T., Laska, J.N., Nejati, H., Baranuik, R.G., Massoud, Y. (2006). *IEEE Dallas circuit and systems workshop.*

Ragheb, T., Laska, J.N., Nejati, H., Kirolos, S., Baranuik, R.G., Massoud, Y. (2008). A prototype hardware for random demodulation based compressive analog to digital conversion. *IEEE conference on circuit and systems*, 37-40.

Tropp, J.A., Laska, J.N., Duarte, M.F., Romberg, J.K., Baraniuk, R.G. (2010). Beyond nyquist: efficient sampling of sparse bandlimited signals. *IEEE transaction on information theory, 56(1)*, 520-544.

Van Den Berg, E., Friendlander, M, P. (2011). Sparse optimization with least-squares constraints. *Society for industrial and applied mathematics, 21(4)*, 1201-1229.

Yu, Z., Chen, Xi., Hoyos, S., Sadler, B.M., Gong, J., Qian, C. (2010). Mixed-signal compressive spectrum sensing for cognetive radios. *International journal of digital mutimedia broadcasting, ID 730509*, 1-10.

Yu, Z., Zhou, J., Ramirez, M., Hoyos, S., Sadler, B.M. (2012). The impact of ADC non-linearity in a mixed signal compressive sensing system for frequency domain sparse signals. *Science direct physical communications journal, 1(5)*, 196-207.