DSP Chip of Compressed Sensing Algorithm for Bio-Sensor Application

Xinping Zhang

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2014-88
http://www.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-88.html

May 16, 2014

University of California, Berkeley College of Engineering

MASTER OF ENGINEERING - SPRING 2014

Electrical Engineering and Computer Sciences

Integrated Circuits

DSP Chip of Compressed Sensing Algorithm for Bio-Sensor Application

Xinping Zhang

This **Masters Project Paper** fulfills the Master of Engineering degree requirement.

Approved by:

1. Capstone Project Advisor:

Signature: _____ Date _____

Print Name/Department: David J. Allstot / EECS

2. Faculty Committee Member #2:

Signature: _____ Date _____

Print Name/Department: Jan M. Rabaey / EECS

Master of Engineering Capstone Project Final Report

DSP Chip of Compressed Sensing Algorithm for Bio-Sensor Application

Xinping Zhang

Group Member: Jun Kwang Oh, Kevin Park, Brian Wang

Faculty Advisor: David J. Allstot

Industry Advisor: Daibashish Gangopadhyay

Abstract

To reduce the power consumption of bio-sensors which will be potentially implanted into human body, compressed sensing is introduced by taking advantage of the sparsity of bio-signals. Using 32nm CMOS technology, the prototype is able to achieve 10-bit resolution under power consumption of 320 nW. The clock frequency is 20 kHz. The circuit described below is designed to compress bio-signal in time domain. [1]

I. Introduction

As the smart phone market has begun to saturate, the whole Integrated Circuit industry is seeking for the next growth point of the personal consumption microprocessor market. One possible answer is bio-sensor implanted into human body. According to the World Health Organization (WHO), almost 20 million people will die from heart disease in 2015. [2] With proper health care and patient monitoring system, this number would be significantly decreased. With such a significant demand remains unmet, implanted bio-sensor collecting bio-signals is a promising growth point in the near future.

Nowadays, the most serious limitation for implanted biosensor is its battery life time. Due to the critical space limitation and difficulty for charging the battery, the total power consumption of the bio-sensor need to be small compared with the energy stored in battery. However, the transmitting power of the implanted antenna needs to be large enough to overcome the high losses caused by human body absorption. [3] So using compressed sensing algorithm, it can transmit less sample while representing the same signal, and by doing that it is able to save great amount of power.

By taking advantage of the sparsity of bio-signals, it is possible to correlate the signal into a smaller data set. [4] Instead of transmitting the whole signal which consumes too much power, it is more power-economics to transmit only the small data set. The goal of this paper is to verify that the microprocessor adopting compressed sensing algorithm consumes less power when processing and transmitting sparse signals.

This paper proceeds as following. In section II, the processor architecture and circuit topology used in our design will be described. In section III, the measured performance such as error to signal ratio and power consumption will be discussed. Final section concludes.

II. Method and Material

1. Overview

The transmitter microprocessor has three sub blocks: ADC, Spike sorting and random matrix compressing. The input for the transmitter is neural signal which has sparsity in time domain. The neural signal is first converted into digital signal that has 10 bit resolution. Then the spike sorting block will filter out all the noise, so that only the spikes are passed to the random matrix compressing block. In the random matrix compressing block, Hybrid Linear Feedback Shift Register (HLFSR) is used to generate required random matrix. The overview block diagram is shown below.

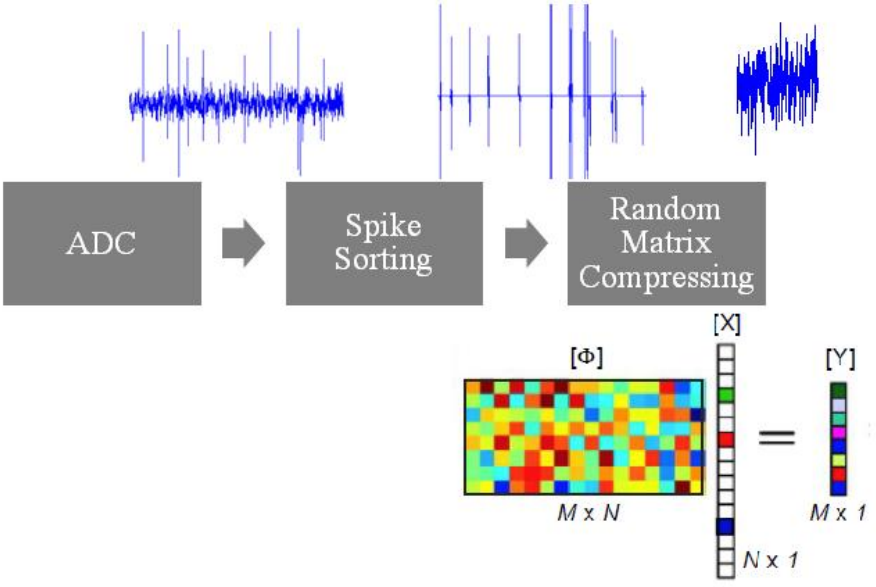

Figure 1. Overview Architecture (**Quoted from [1]**)

In this project, I am involved in the design of spike sorting and random matrix compressing blocks. I will focus more on the design of these two blocks and report about ADC can be found in Brain and Kevin's paper.

2. Spike Sorting

The spike sorting block is used to reduce data rate, so that within a given amount of power budget, more channels can be supported and longer battery life is achieved. The idea of spike sorting is that a threshold voltage is determined at the beginning and signal that is under this threshold will be rejected while the others are passed. More detailed algorithm will be presented in the following paragraph.

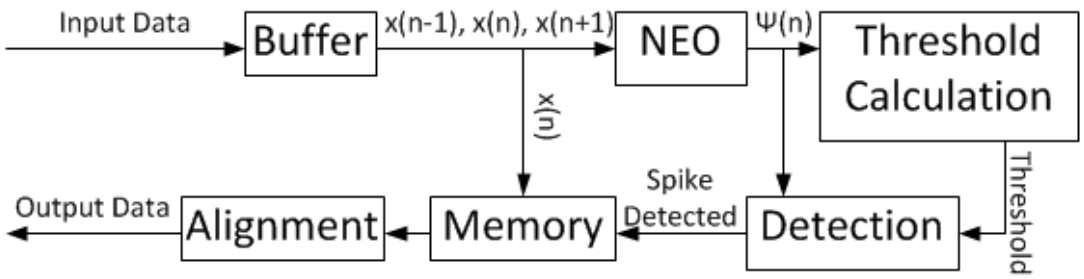

Figure 2. Spike Sorting Block Diagram (**Quoted from** [5])

In order to balance the error rate and complexity [5], nonlinear energy operator (NEO) is chosen to implement. In the above block diagram, NEO block is used to calculate $\psi(n)$ from the following equation [6]:

$$\psi(n) = x^2(n) - x(n-1) \times x(n+1)$$

The threshold value is defined as [6]:

$$\text{Threshold} = C \times \frac{1}{N}\sum_{n=1}^{N} \psi(n)$$

For the sample input file that contains 10,000 samples. The threshold value is determined by the

first 1024 samples. And C in the above equation is determined as 8. [5]

If the signal is above the threshold value, a spike is detected. 22 signal samples before this threshold crossing point and 50 signals sample after this threshold crossing point would be send to memory block. The final output will contains 48 samples. Within the memory block signals will be aligned so that the maximum derivative happens at the 11^{th} sample of the final output. The maximum derivative is defined as the difference between the current original signal and one before that. [5]

$$\text{Derivative}(n) = s(n) - s(n-1)$$

3. Random Matrix Compressing

A. Compressing

In the transmitter side, compressed sensing is described by the following equation. [Φ] is a matrix with size of [N*M], where N>>M. Therefore, the input vector with size N is converted to an output vector with size M, that is to say compressed by a factor of N/M [1].

$$[Y] = [\Phi][X]$$

B. Reconstruction

On the receiver side, accurate reconstruction can be realized when these two conditions are met: 1. [X] is sparse in time domain; 2. the sparsifying matrix [ψ] (in time domain, [ψ] is identity matrix) and random matrix [Φ] are incoherent.

If the above two conditions are met, accurate reconstruction can be realized by using l1-norm algorithm. In this project l1-MAGIC program [7] is adopted that can find [X] to minimized ||[Y]-[Φ][X]|| [8].

Since the receiver is usually less power constrained, in this project more effort is focused on the

transmitter side. But MATLAB simulation for reconstruction is done in order to ensure accurate recovery.

III. Result

1. Spike Sorting

A. Starting Point

Without any optimization, the power dissipation for the Spike sorting block is summarized in the following table. With the functionality correct Verilog code, the total power dissipation is 1.74 mW and it is dominated by the leakage.

Table 1. Power Consumption for Spike Sorting Block without Optimization

	Switch Power (uW)	**Int Power (uW)**	**Leak Power (mW)**	**Total Power (mW)**	**%**
Buffer	**2.23e-3**	**7.07e-3**	**0.312**	**0.312**	**17.9**
NEO/Threshold	**10.9e-3**	**29.5e-3**	**0.296**	**0.296**	**17**
Memory	**4.57e-3**	**12.1e-3**	**0.821**	**0.821**	**47.1**
Clock Divider	**103e-3**	**1.160**	**0.0155**	**0.0155**	**0.9**
Alignment	**31.3e-3**	**71.3e-3**	**0.296**	**0.296**	**17.1**
Total	**0.152**	**1.280**	**1.74**	**1.74**	**100**

In order to reduce the leakage power which dominated the total power dissipation, Power-gating, Clock gating, Muti-Vdd and High-threshold devices are used. I will report about the power gating,

other optimization methods can be found in Kevin and Jun Kwang's report.

B. Power Gating

Power gating technology is used to selectively shunt off current where the block is not used. In this design we use PMOS header in the real implement.

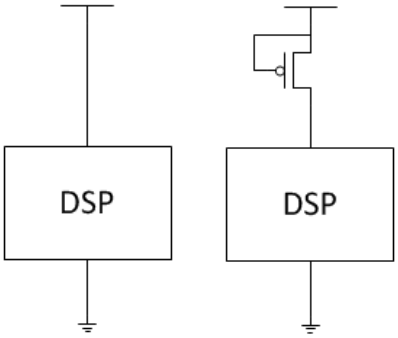

Figure 3. Power Gating Sample

The idea of power gating is that the Memory and Alignment blocks are turned off, when there is no spike detected signal, while the NEO, including spike detection and threshold calculation blocks, and buffer blocks are cut off when there is spike that is detected. As expected, the impact of power gating to the total power dissipation is heavily depends on the input vector. The equation used to calculate the power dissipation is shown as follows.

$$\begin{aligned}\text{Power gating estimation} &= (\text{Original Power}) \cdot \left(\frac{\text{gated power}}{\text{ungated power}}\right) \cdot (\%\ \text{of gated time}) \\ &+ (\text{Original Power}) \cdot\ (1 - \%\ \text{of gated time})\end{aligned}$$

In order to determine the efficiency of the power gating, the leakage current of a simple inverter under different supply voltage is compared.

Table 2. Efficiency of Power Gating

Supply Voltage (V)	Leakage Power Without PMOS header (pW)	Leakage Power With PMOS header (pW)	% Power Savings
0.5	**200.3**	**38.3**	**81**
0.6	**335.5**	**49.4**	**85.3**
0.7	**541.4**	**61.7**	**89**
0.8	**849.2**	**75.3**	**91**
0.9	**1303**	**90.4**	**93**
1.0	**1964**	**107.1**	**94.5**

Based on the current simulation result and the equation shown before, the efficiency of power gating can be summarized in the following table. With power-gating technology, the total power dissipation is reduced by 59%.

Table 3. Power Consumption of Spike Sorting Block with Power Gating

	Power with Power Gating (mW)	% of performance	% Saving
Buffer	**0.288**	**40**	**7.7**
NEO/Threshold	**0.274**	**37.9**	**7.4**
Memory	**0.106**	**14.8**	**87**
Clock Divider	**0.0142**	**2.0**	**0**
Alignment	**0.0384**	**5.3**	**87**
Total	**0.721**	**100**	**58.6**

C. Optimized Result

With all possible optimization techniques, including high-threshold device, clock-gating, power gating, VDD lowing and multi VDD, adopted, the final result is shown below. The power dissipation for spike sorting block is 240 nW.

Table 4. Optimized Power Consumption of Spike Sorting Block

	Switch Power (W)	**Int Power (W)**	**Leak Power (W)**	**Total Power (W)**
HVT library at VDD = 0.78V	**6.54e-08**	**5.95E-07**	**1.30E-05**	**1.36E-05**
With scaled down to VDDH = 400 mV and VDDL = 300 mV	**2.5e-9**	**2.28e-8**	**5.1e-7**	**5.2e-7**
With power gating (delay=703 ns)	**1.15e-9**	**1.05e-8**	**2.3e-7**	**2.4e-7**

2. Random Matrix Compressing

In order to ensure accurate recovery, two explorations are done in order to find the optimized solution.

(1) Is the spike sorting block is necessary for accurate reconstruction? Can it be removed?

In order to verify whether the spike sorting block is necessary or not, two architectures are compared. One with spike sorting block, while the other architecture without spike sorting block. Using ideal Bernoulli random matrix generated by MATLAB, compressing factor of 4 and input vector of neural signal with sparsity in time domain, the difference between uncompressed signal and reconstructed signal is presented in the following picture.

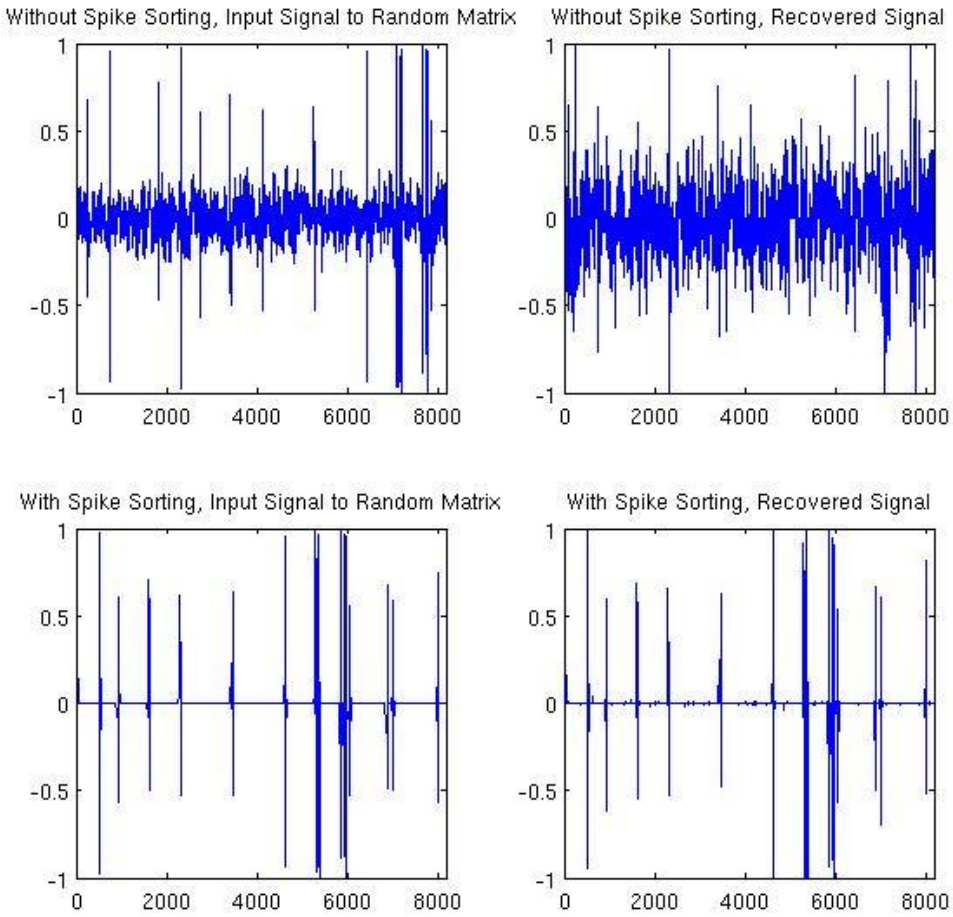

Figure 4. (a) Input signal to random matrix without spike sorting block (b) Recovered signal without spike sorting block (c) Input signal to random matrix with spike sorting block (d) Recovered signal with spike sorting block

To quantify the result, the error to signal ratio is defined in the following equation:

$$\mathrm{ESR} = \frac{\sum_{n=1}^{N}\left(x(n) - x'(n)\right)^2}{\sum_{n=1}^{N} x(n)^2}$$

The result is summarized in the following table:

Table 5. ESR Comparison between with and without Spike Sorting Block

	Without Spike Sorting Block	**With Spike Sorting Block**
ESR	**0.84**	**0.02**

From the table with spike sorting block, the reconstructed signal has less error. Since accurate

reconstruction requires that the signal is sparse with respect to sparsifying matrix [ψ], with noise removed, the sparsity of the signal is improved. Therefore, more accurate reconstruction can be achieved. So, spike sorting block indeed helps to improve the sparsity of the signal by removing the noise.

(2) How many bits of coefficient are necessary for the random matrix?

The criteria to compare different random matrix is to check the Restricted Isometry Property (RIP). For a random matrix [Φ] with size M*N, there exist a isometry constant δ_s. So that for any sub-matrix [Φ_s] with size M*S, (1<S<N), and any vector [X], the following equation holds [1]:

$$1 - \delta_s \leq \frac{\left|\left|[\Phi_s][X]\right|\right|_2^2}{\left|\left|[X]\right|\right|^2} \leq 1 + \delta_s$$

But due to the complexity of checking RIP, coherence is used as a substitution. It is defined as the following equation. The lower the coherence value implies a more accurate reconstruction [1].

$$\mu(\phi, \psi) = \sqrt{N} \max_{1 \leq k,j \leq N} |< \phi_k, \psi_j >|$$

Linear feedback shift register (LFSR) [7] is used to generate the random matrix with size of 64*16. Adopting the idea of Monta Carlo method, in simulation 1000 seed vectors are randomly generated to better compare the coherence for random matrix with different number of coefficient.

Table 6. Statistic Number for Coherence Distribution

Number of Bits	**1**	**2**	**3**	**4**	**5**	**6**
Mean	**1.18**	**1.57**	**2.14**	**2.14**	**2.16**	**2.16**
Standard Diviation	**0.09**	**0.12**	**0.19**	**0.18**	**0.18**	**0.17**

The detailed distribution is plotted in the following picture:

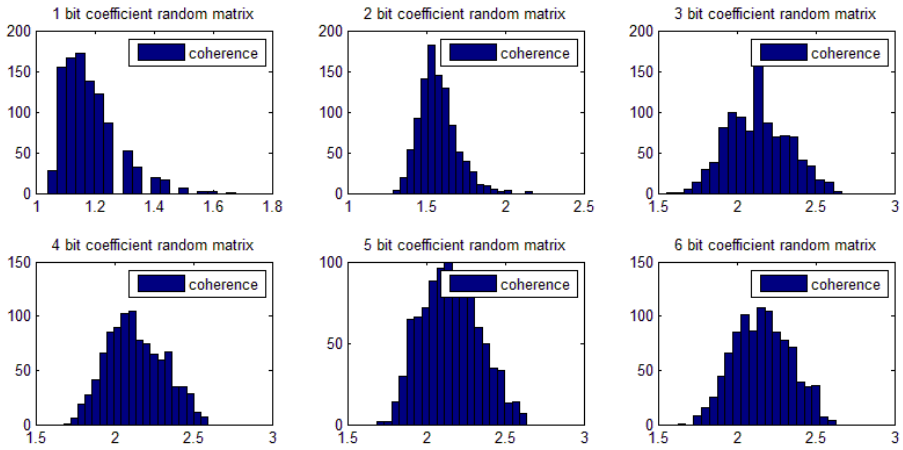

Figure 5. Coherence Distribution Histogram

From the plot and table shown above, it can be shown that 1 bit coefficient random matrix is able to generate a lower coherence compared with others. In the next section, it is also shown that 1-bit coefficient will have smaller power dissipation.

From the above analysis, 1-bit coefficient random matrix is used in the final design. More detailed information about how the 1-bit coefficient random matrix can be found in Jun Kwang's report. We also made a 6-bits coefficient random matrix for rough comparison.

The architecture is adopted from [1]. A 6 bits Fibonacci LFSR is constructed to generate one 6 bits element in matrix, using the tap table in [10]. With such connection, a feedback primitive polynomial of x^6+x^5+1 is formed and it is able to achieve a maximized length of 63. 8 such cells are connected to generate a 48 bits Fibonacci/Galois hybrid linear feedback shift register; finally, eight such blocks are cascade to generate the total 384 random bits that is required. The plot is shown below and from Verilog simulation, the power result is shown in the following table.

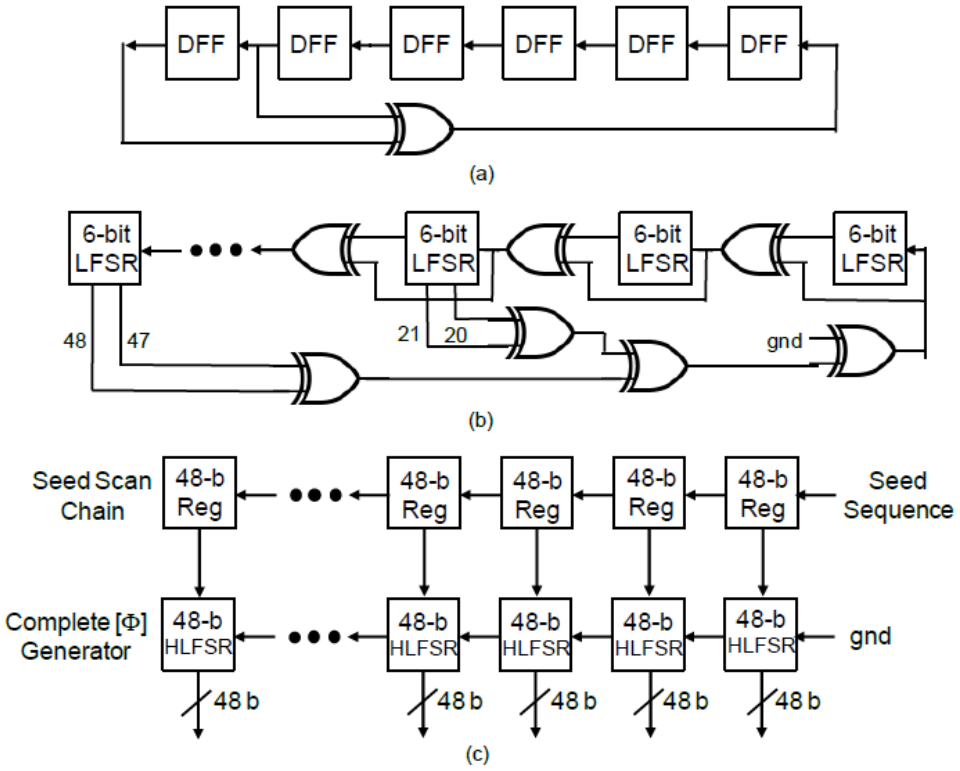

Figure 6. Hybrid Linear Feedback Shift Register Architecture (**Quoted from [1]**)

Table 6. Power Consumption for 6-bits Coefficient Random Matrix

Block	**Power Dissipation for 6-bits Random Matrix (nW)**
6 bits LFSR	**3**
48 bits LFSR	**52**
324 bits LFSR	**421**

To compare the power dissipation between 1-bit coefficient and 6-bits coefficient random matrix, a rough analysis is to compare the number of registers that is needed to generate the random matrix. For an M*N random matrix, the number of registers that is needed is 6*M*N for 6-bits random matrix, while for Bernoulli random matrix this number can be divided by 6. Roughly speaking, the Bernoulli random matrix would have a power consumption of 70nW. More detailed information about the 1-bit random matrix is included in Daniel's report.

IV. Conclusion

Wearable implanted microchip has potentially large market ahead. But the technology barrier is high due to the severe ultra-low power environment. The system presented is designed to compress neural signal that has sparsity in time domain. Compressed sensing technology in this paper enables a large reduction of data rate needed to be transmitted. Moreover, the total power dissipation for its own is low.

V. Reference

[1] E. G. A. A. M. R. D. K. N. S. G. a. D. J. A. Daibashish Gangopadhyay, "Compressed Sensing Analog Front-End for Bio-Sensor Applications," *IEEE Journal of Solid-State Circuits,* pp. 1-2, 2013.

[2] World Health Organization, [Online]. Available: http://www.who.int/mediacentre/factsheets/fs317/en/index.html..

[3] K. Y. Y. a. R. Kohno, "Wireless Communications for Body Implanted," in *Asia-Pacific Microwave Conference*, 2007.

[4] S. S. D. B. Marco F. Duarte, "Distributed Compressed Sensing of Jointly Sparse Signal".*Department of Electrical and Computer Engineering, Rice University.*

[5] S. G. a. D. M. Vaibhav Karkare, "A 130-uW, 64-Channel Neural Spike-Sorting DSP Chip," *IEEE JOURNAL OF SOLID-STATE CIRCUIT,* MAY 2011.

[6] S. M. a. G. C. Ray, "A New Interpretation of Nonlinear Energy Operator and Its Efficiency in Spike Detection," *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, 2 1998.

[7] E. C. a. J. Romberg, "L1-magic: Recovery of Sparse Signals," 2005.

[8] D. L. Donoho, “For Most Large Underdetermined Systems of Equations,the Minimal L1-norm Near-Solution Approximates the Sparsest Near-Solution,” 8 2004.

[9] N. BAJAJ, "LFSR," http://www.mathworks.com/matlabcentral/fileexchange/29119-lfsr, 2010.

[10] "LFSR TAPS TABLE," [Online]. Available: http://home1.gte.net/res0658s/electronics/LFSRtaps.html. [Accessed 13 4 2014].