PROGRAMMABLE AND RECONFIGURABLE STRAIN-POWERED MICRO-DATA-LOGGERS BASED ON LINEAR PIEZO-FLOATING-GATE INJECTORS

By

Pikul Sarkar

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Electrical Engineering

2012

ABSTRACT

PROGRAMMABLE AND RECONFIGURABLE STRAIN-POWERED MICRO-DATA-LOGGERS BASED ON LINEAR PIEZO-FLOATING-GATE INJECTORS

By

Pikul Sarkar

This thesis describes the design, implementation and testing of self-powered, large dynamic-range, micro-data-logger that can be used for sensing, computing and non-volatile data storage of mechanical-strain statistics. At the core of the proposed design is a linear floating-gate injector that can achieve more than 13 bits of precision and are self-powered by the piezoelectric transducers that convert mechanical energy from strain-variations into electrical energy. The first fundamental contribution of this thesis is a novel differential injector topology that is used to measure static-strain by integrating the difference between the L1 measure of the piezoelectric signal generated during the positive and negative strain-cycles. The second fundamental contribution of this thesis is a novel compressive self-powering technique that overcomes the input threshold effect of most self-powered sensors. By using a non-linear impedance circuit at the output of the piezoelectric transducer and by using programmable level-crossing circuit and offset cancelation circuits, the thesis demonstrates an extended self-powering range greater than 40dB. A system-on-chip solution has been designed that integrates the linear floating-gate injectors with high-voltage charge-pumps, digital calibration and digital programming circuits. Extensive experiments with the system-on-chip prototypes fabricated in a 0.5μm standard CMOS process and piezoelectric material (PZT) have been performed using a bench-top mechanical test setup. An automated programming and calibration of the sensor has been developed and the results have been calibrated against standard strain-gauge measurements.

ACKNOWLEDGMENTS

I would like to take this opportunity to acknowledge several people who have helped me in some way during my master work.

First of all, I thank Dr. Shantanu Chakrabartty for giving me the opportunity to work in his group with financial support. He was extremely supportive throughout my studies with the right mixture of freedom and discipline. I specially thank him for making me familiar with the chip testing procedure and debug problems with patience. I am also very thankful to my other committee members, Dr. Donnie Reinhard and Dr. Subir Biswas for their valuable suggestions.

To my great labmates specially-Tao Feng, Ming Gu and Chenling Huang who made working so much fun and also gave great inputs to my work. Extra thanks to Chenling Huang as I continued the work done by him before and he had been a great help in the initial days.

And to all my friends, thank you for making me feel at home and providing the motivation for research work.

Finally, my most gratitude goes to my parents, sister and my brother-in-law. They have been a constant source of inspiration and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

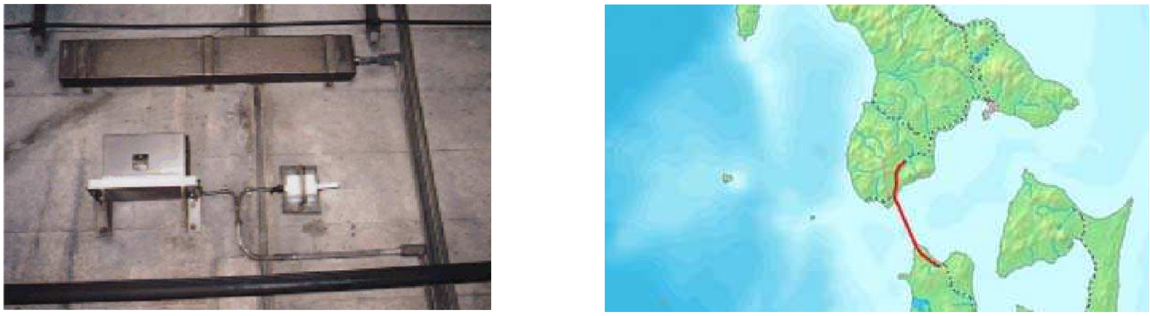
This chapter begins with the applications of strain-gauges and a brief survey of existing strain-gauge technologies. The motivation of self-powered strain-gauges has been discussed next followed by the problems of using piezoelectric self-powered strain-gauges by conventional methods and also by a sensor previously proposed by us. This is where the work in the thesis comes in and the chapter finishes with a list of the contributions of this thesis.

1.1 Application of Strain Gauge

Measurement of mechanical strain is extremely important in both civil structures and also inside the human body as it indicates the degradation and health of the structure.

1.1.1 In Structural Health Monitoring

A good example of the usage of strain-gauge in Structural Health Monitoring(SHM) is the Seikan tunnel, which is a 33.46 mi railway tunnel in Japan, with a 14.5 mi long portion under the seabed and it is 460 ft below seabed and 790 ft below sea level. It connects the islands of Honshu and



(a) (b)

Figure 1.1: The Seikan Tunnel:(a)Strain gauges on tunnel lining; (b)Location and Depth. (Fig Sources:(a)http://www.jsce_int.org/civil_engineering/1997/Longest_Tunnel/FOREWORD.htm, (b)http://letzwave.blogspot.com/2010/03/seikan-tunnel-worlds-longest-under-sea.html) (For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this thesis.)

Hokkaido, carrying a total of about 100 scheduled commuter and freight trains daily. At four locations, selected from among tunnel sections which proved difficult during construction or where bad geological conditions were noted, strains at seven points in the longitudinal direction and at 14 points in the circumferential direction are measured automatically using high sensitivity strain gauges as shown in Fig. 1.1(a). There are over 600,000 bridges in the United States and almost 13% have some sort of structural damage. Thus it is important to use strain-gauge to monitor the health of the bridges. Tragedies can be avoided by its usage. For instance, the six-lane, 2.9 km (2 mile) Charilaos Trikoupis Bridge (Rion-Antirion Bridge) in Greece has 100 sensors (300 channels) that monitor its condition. Soon after opening in 2004, the sensors detected abnormal vibrations in the cables holding the bridge, which led engineers to install additional weight to dampen the cables. The Tsing Ma Bridge in Hong Kong, the world's seventh longest suspension bridge, is equipped with more than 350 sensor channels which includes accelerometers, strain gauges, anemometers, weigh-in-motion devices, and temperature sensors. [29] Applications of strain-gauge can be also be seen in the field of aerospace. Fig. 1.3 shows one such case where strain gages are attached to the propeller blades to find out the optimal direction to place them.

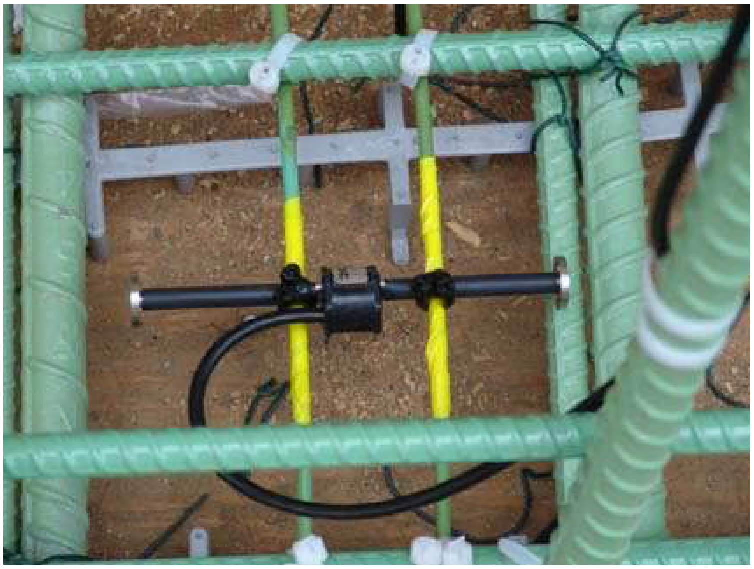


Figure 1.2: St. Anthony Falls Bridge: Structural deformations are measured by 195 vibrating wire strain gauges(VWSGs), 24 resistive strain gauges, and 12 fiber optic displacement sensors.

(Fig source:http://buildipedia.com/operations/public-infrastructure/innovative-infrastructure-smart-bridges)

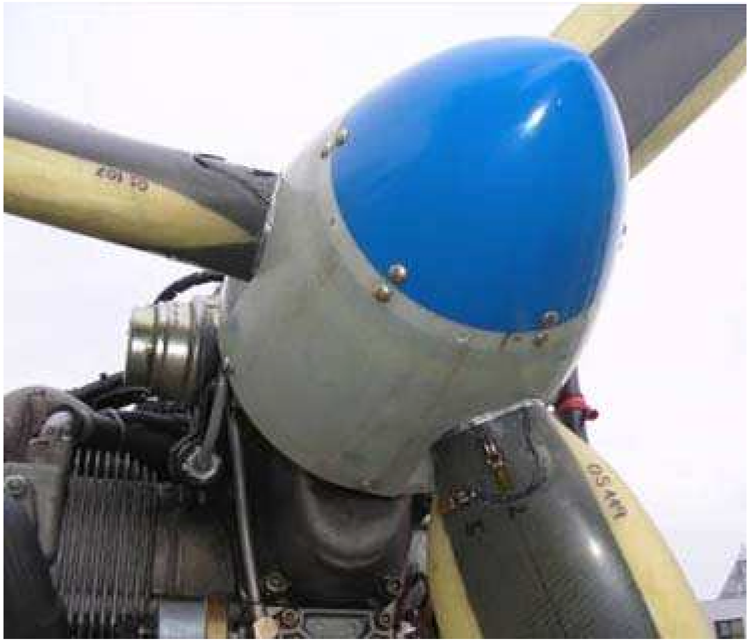


Figure 1.3: Fiber-composite propeller blades with LY41-6/350 strain gages from HBM, with telemetric data transmission.

(Fig Source:http://www.hbm.com/)

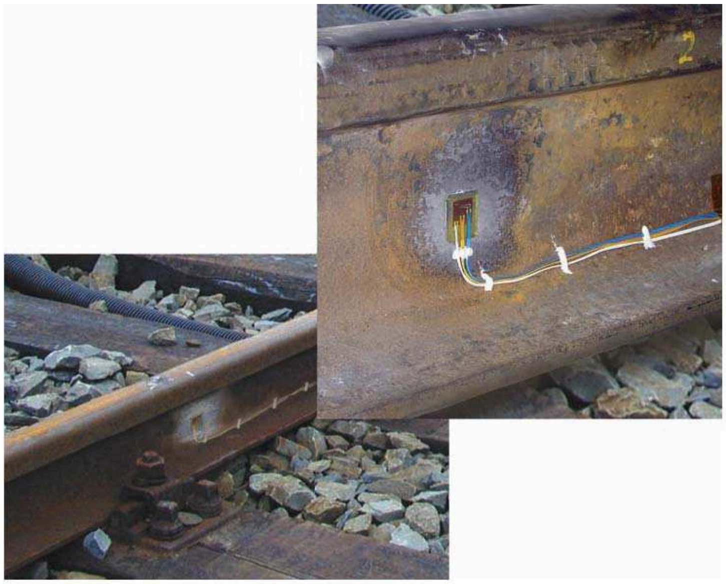


Figure 1.4: Experimental investigations on a railway track using encapsulated SG series HBM strain gauge.
(Fig Source:www.hbm.com)

Figure 1.5: Tensile force measurement using C series HBM strain gauge at low temperatures.
(Fig Source:www.hbm.com)

1.1.2 Biomedical Applications

In-vivo monitoring of mechanical strain is important in biomedical experiments related to the study of osteoporosis or muscular dystrophy where the objective is to understand the progressive failure and degradation mechanisms of biomechanical structures like bones, muscles or ligaments [1, 2, 3, 4]. For example, repetitive strains greater than approximately 1500$\mu\varepsilon$ can lead to fatigue damage and possible stress-fractures in bone [5]. Conversely, strains lower than 500$\mu\varepsilon$ are thought to increase the risk of bone absorption and osteoporosis [6]. Strain measurement is also important for rehabilitative studies in ACL injuries. Thus some kind of strain sensor is needed to monitor the strain levels.

Typical strain levels experienced in biomechanical structures are summarized in Table 1.1.

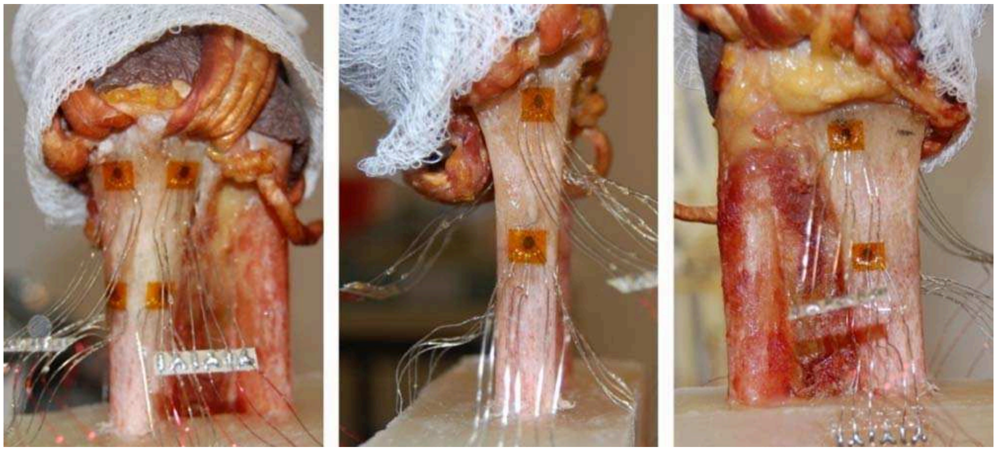
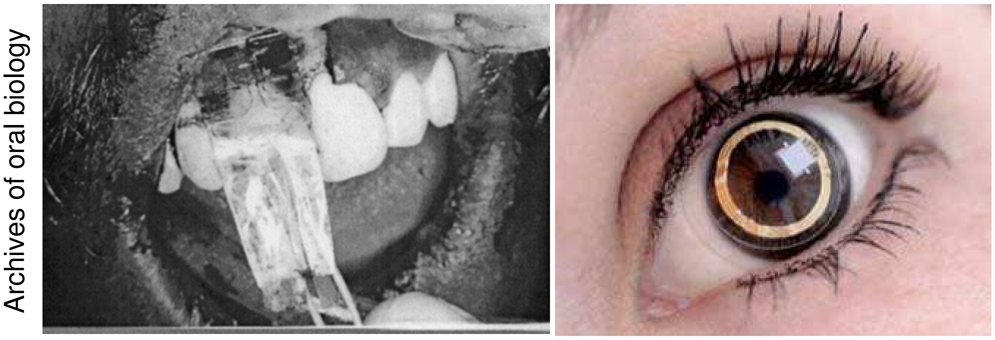


Figure 1.6: Gauges on Bones.
(Fig source:W. Brent Edwards et.al)



Archives of oral biology

Figure 1.7: Gauges on Teeth and Eyes.

Table 1.1: Strain Levels in Biomechanical Structures.

Structures	Strain Levels
Nerves	$1000-200,000\mu\varepsilon$ [3]
Bones	$400-1,600\mu\varepsilon$ [1]
Ligaments	$1000-40,000\mu\varepsilon$ [2]
Muscles	$1000-50,000\mu\varepsilon$ [4]

1.2 Brief Survey of existing Strain gauge technologies

There are multiple existing technologies to measure strain. The most popular among them is the resistive strain-gauge. Resistive strain gauge can be made of three types of material-metal, semiconductor and vapor deposited(thin film). Strain gauges can also be capacitive where strain is measured by the change in the capacitance due to the change in plate separation distance. Another type of strain measurement is Fibre Bragg grating(FBG) based biomedical pressure sensors which measures spectroscopic changes due to mechanical displacement of an implanted optical fiber. Piezoelectric strain gauges are also popular in which the piezo material generates voltage from a strain input and this voltage is measured to estimate the strain. Selection of a particular type of strain gauge depends on the application.

1.2.1 Resistive

This type of strain-gauge depends on the change in resistance due to the change in length(l) and width(cross sectional area A) when it is deformed due to strain. As we know the relation between resistance(R) of a metal wire and its dimensions is given by,

$$R = \rho\frac{l}{A} \tag{1.1}$$

where ρ is the resistivity. To get the maximum resistance change it is usually laid out in a concertina pattern. The gauge is placed in such a way that the strain is in its longitudinal direction. They are generally made of thin metal-foil grids that can be adhesively bonded to a surface. Different metals and alloys such as constantan, advance, karma, nichrome, and germanium are used. The resistance change is measured using a wheatstone bridge as shown in Fig. 1.8(a). To cancel out temperature effects another static strain gauge is used as another branch of the bridge. The change in voltage is often too small and thus it has to be amplified using a differential amplifier. One important parameter is the Gauge Factor(GF) which is given by the following expression,

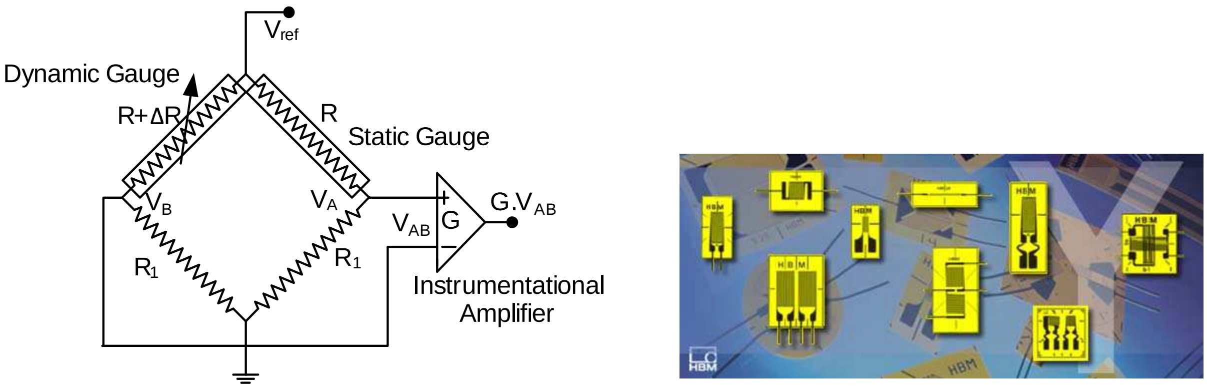
$$GF = \frac{\frac{\Delta R}{R}}{\varepsilon} = \frac{\beta}{\varepsilon}, \tag{1.2}$$

where ε is the strain, ΔR is the change in resistance and $\beta = \frac{\Delta R}{R}$. GF relates the change in resistance to the strain. From Fig. 1.8(a) it can be seen that

$$V_{AB} = V_{ref} \cdot \left(\frac{R}{R_1} - \frac{R + \Delta R}{R_1}\right), \tag{1.3}$$

When there is a positive or negative ΔR present due to strain V_{AB} would be non zero too and it would be amplified by the amplifier gain G. From this reading ΔR can be calculated and then strain can be calculated from eq. 1.2.

Metal-foil strain gauges are the most popular as they have multiple advantages. First of all, they are of very small size(size of a postage stamp or even smaller, yet quite rugged and bonds excellently to most surfaces. They can be also wrapped around curved surfaces. As they are metal they readily dissipate heat. Other good qualities are high linearity, minimal sensitivity to transverse strain (perpendicular to intended direction) and good spatial resolution (measure strain at a point).



(a) (b)

Figure 1.8: (a) Balanced bridge circuit to measure strain; (b)metal-foil strain gauges.

But there are some disadvantages too. Firstly, the resistance changes with temperature and thus the gage factor changes with temperature too. Also strain gage grid expands and contracts at a different rate than the surface it is attached. It has typically very low gage factor(in the order of 2) and thus has lower sensitivity to strain. So very small strain variation cannot be measured by it.

1.2.2 Piezoresistive(Silicon)

The resistance of silicon changes with strain due to the change in its resistivity(ρ). Resistivity is related to mobility which depends on the mean free time between collisions and that depends on the interatomic distances which change due to the strain. This could be either silicon crystal moderately doped or polycrystalline in MEMS piezoresistors [27]. They have gauge factors more than that of metal gauges.

1.2.3 Capacitive

Capacitive strain sensors have advantages in temperature drift, sensitivity, noise, and dynamic range, as compared to metal foil and piezoresistive sensors. One such capacitive mems based strain gauge was reported in [28].

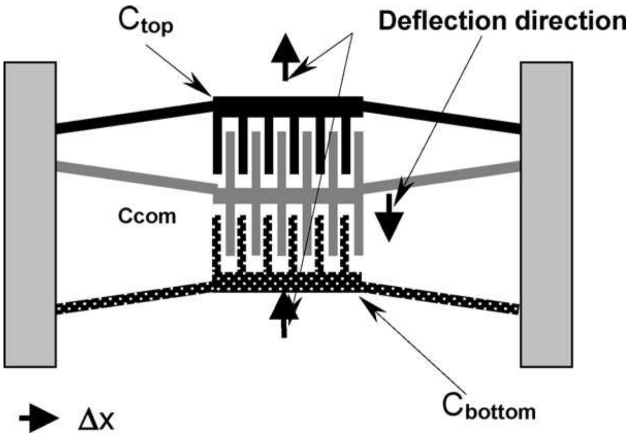


Figure 1.9: Mems capacitive sensor.
(Fig source:[28])

The sensor is a linear differential comb capacitor. The capacitive comb based structure is shown in figure 1.9. The strain applied to the structure along the X axis, causes the arms to move in the Y direction. The net displacement of the arms can be enhanced by appropriately designing the length and angle of the arms. The two outer electrodes move in the positive Y direction, while the central electrode moves in the opposite direction. This increases the overlap area and hence capacitance, between the lower pair of combs and reduces the overlap area and capacitance between the upper pair. This double differential mechanism further increases the sensitivity of the sensor. The maximum measurable strain is limited by the length of the fingers and the required finger overlap area. At the limiting strain, the fingers of adjacent comb will touch the support beam or there will be no overlap area between adjacent combs. This is an intrinsic limitation of this design. For this kind of sensor, gauge factor(GF) is given by

$$GF = \frac{\frac{\Delta C}{C}}{\varepsilon}, \tag{1.4}$$

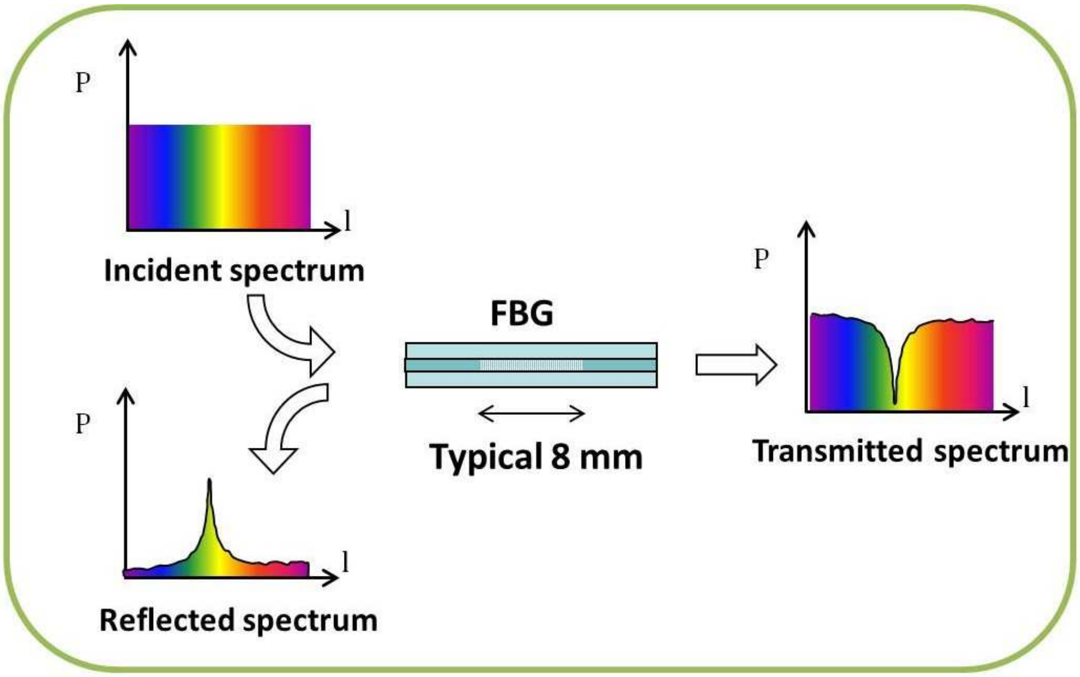


Figure 1.10: Working principle of FBG
(Fig Source:http://www.fbgs.com/technology/fbg-principle/)

1.2.4 Fibre Optic

This kind of sensor measures the change in wavelength or phase of light and them corresponding strain in calculated accordingly.([9, 23, 24, 25])

Fiber Bragg Gratings are made by laterally exposing the core of a single-mode fiber to a periodic pattern of intense ultraviolet light. The exposure produces a permanent increase in the refractive index of the fiber's core, creating a fixed index modulation according to the exposure pattern. This fixed index modulation is called a grating.

At each periodic refraction change a small amount of light is reflected. All the reflected light signals combine coherently to one large reflection at a particular wavelength when the grating period is approximately half the input light's wavelength. This is referred to as the Bragg condition, and the wavelength at which this reflection occurs is called the Bragg wavelength. This principle

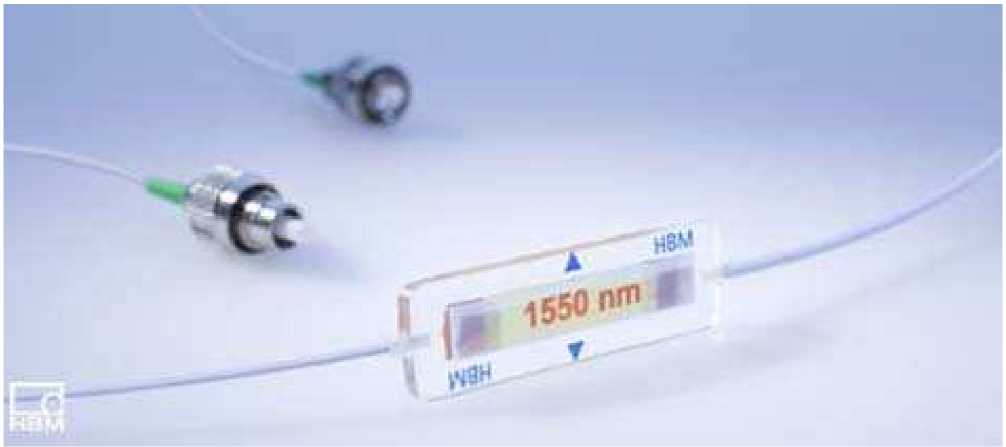


Figure 1.11: A HBM made FBG sensor.
(Fig source:www.hbm.com)

is shown in Fig. 1.10.

The central wavelength of the reflected component satisfies the Bragg relation: $\lambda_{refl} = 2n\nu$, with n the index of refraction and ν the period of the index of refraction variation of the FBG. Due to strain dependence of the parameters n and ν, λ_{refl} will also change as function of strain. This dependency allows determining the strain from the reflected FBG wavelength.

Though FBG cannot fully compete with metal strain gauges regarding price and precision it has certain advantages: 1. They match very well with composite materials and thus can be integrated on the surface of test objects like airplanes or power plants; 2.Can measure very high strain($> 10,000\mu\varepsilon$); 3. Lightweight; 4. Immune to EMI or lighting interference; 5. FBG signals not distance dependent; 6. High long term stability.

Its weaknesses are: 1. High temperature dependence(effect of 1° Celsius temperature is of $8\mu\varepsilon$); 2. Gage factor is quite low(0.77-0.81); 3. Stiffness is higher than that of foil strain gages causing higher parallel force to specimen; 4. Sensing fibre core is positioned at a greater distance from specimen causing errors; 5. Radius$> 10mm$. Therefore quite large; 6. Bulky optics for conditioning and controlling the light beam.

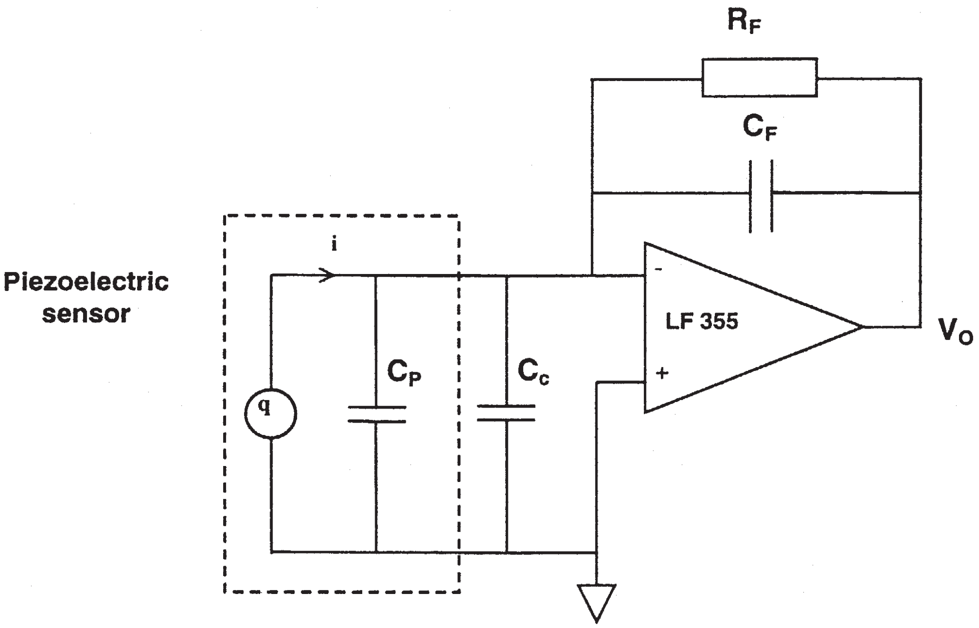


Figure 1.12: Charge Amplifier with piezo.
[26]

1.2.5 Piezoelectric

Piezoelectric elements can be used as strain sensors [26]. Strain is measured in terms of the charge generated by the element. Piezo materials cannot be isotropic as then it would not generate any electrical polarization when force is applied.

Piezo materials are of mainly two types-piezoceramics and polymers. The most commonly used piezoceramics, PZT(Lead Zirconate Titanate) are solid solutions of lead Zirconate and lead Titanate with other dopings. They exhibit properties like high elastic modulus, brittleness and low tensile strength. PVDF is a polymer(Polyvinylidene Fluoride). The Young's modulus of PVDF is approximately 1/12th of PZT. So it is more suited to sensing applications as it is more likely to adjust in the host structure. But the problem is the piezoelectric coefficient is 1/10th of PZT and also highly temperature dependent.

It is important to pass the output of the piezoelectric sensor through some signal conditioning as the piezo output impedance is very high. One way is to connect the outputs to a high vauled resistor and measure the current through it using a current amplifier. Another way is to use a charge amplifier to measure the charge generated by the sensor, which is equivalent to the strain as shown in Fig. 1.12.

1.3 Motivation for Self-Powered Sensor

Even though all these techniques can precisely measure instantaneous strain-levels down to a few $\mu\varepsilon$, they are passive in nature and do not provide any historical information about the strain signal which could be used for understanding progression of mechanical degradation. An example of the historical information could be some measure of the strain-energy dissipated through the bio-mechanical structure or could be the running average of the strain signal during the entire observation period. Without historical information, the strain-measurement could be prone to ageing artifacts of the gauge and also prone to the degradation in the interface between the gauge and the bio-mechanical structure being monitored. In principle, passive strain-gauges could be complemented with additional circuitry that continuously read, process and store the desired information. Continuous operation of the add-on circuitry could be achieved through powering by implanted energy storage devices (batteries or super-capacitors) that could be periodically replenished remotely or using energy-scavenging, in-vivo. However, small volume requirements of the sensor severely limits the capacity of energy storage and in-vivo energy harvesting devices. An ideal solution would be an asynchronous self-powering approach where the strain-gauge harvests its operating power directly from strain variations, compute the parameters of interest and store the parameters till they are retrieved or read-out.

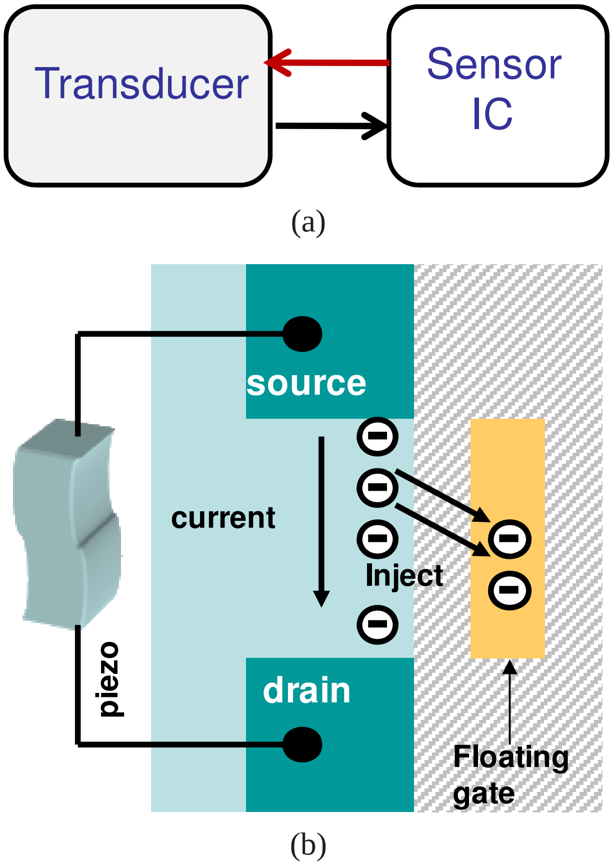


Figure 1.13: Asynchronous Self-Powered Sensor(a)Principle; (b)An example using the interface physics between piezoelectric material and floating-gate MOS.

1.4 Challenges in Piezo-based self-powered strain gauges

Piezoelectric strain gauges described before can be potentially made self-powered if the sensor is powered by the electrical power generated. But that would mean the load seen by the piezo is not infinite now, it will draw current which would lead to problems in strain measurement. This problem is discussed below.

1.4.1 Piezoelectric Model

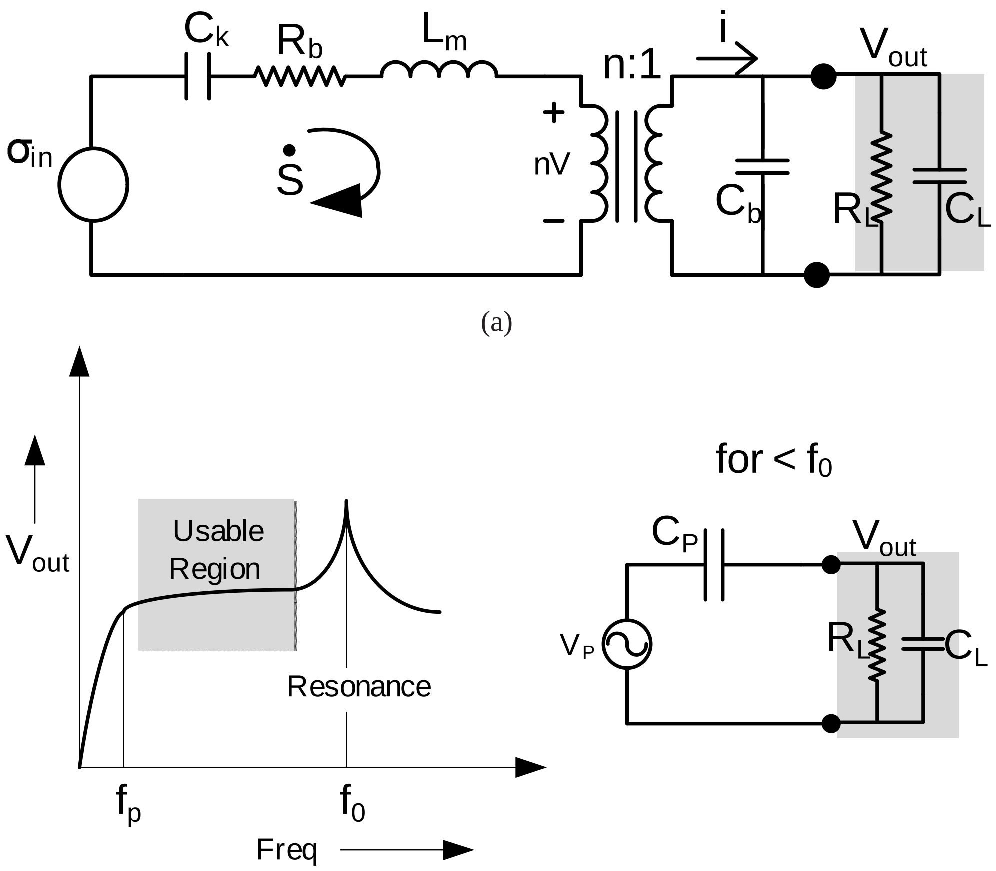
In literature, piezoelectricity based mechanical-to-electrical conversion has been extensively modeled and Fig. 1.14(a) [15, 17, 16, 18] shows a widely accepted electrical model that is valid over a wide frequency range. The model maps all the mechanical parameters into its equivalent counterpart which simplifies the analysis of any electrical circuit that is driven by a piezoelectric transducer. For instance, the inductor L_m models the effect of the mass of the transducer and the effective mass of the substrate to which the transducer is attached. The capacitance C_k models the mechanical stiffness of the transducer and the R_b models the mechanical damping of the transducer. The voltage source models the mechanical stress σ_{in} induced by strain variations and the mechanical-to-electrical transformer (with transformation ratio n) reflects the transformation of the mechanical variables into electrical variables (voltages and currents). On the electrical side of the transformer, the electrical load connected to the transducer is modeled by the load capacitance C_L of the material and the load resistance R_L.

Based on the electrical model in Fig. 1.14(a), the frequency response of a piezoelectric transducer is shown in Fig. 1.14(b). The series inductance and capacitance leads to the resonance at a frequency $f_0 = 1/2\pi\sqrt{L_m C_k}$. Below the resonant frequency, the response of the transducer is captured by simple capacitively coupled voltage source which forms a high-pass R-C circuit, as shown in Fig. 1.14(b). This simple R-C model is sufficient to capture the dynamics of the transducer under quasi-static operating conditions or frequency of operation below 20Hz.

1.4.2 Problem in quasi-static Strain Measurement

For a piezoelectric cantilever with dimensions $L \times b \times h$, the open-load voltage($v_{in}(t)$) generated across the transducer as a function of a perpendicular mechanical force $F(t)$ is given by [12]:

$$v_{in}(t) = \frac{F(t)g_{31}}{b} = S(t)Y^E hg_{31} = \frac{S(t)Y^E d_{31}h}{\varepsilon} \tag{1.5}$$



(a)

(b)

Figure 1.14: Model of a piezoelectric transducer: (a) Complete electrical model; (b)Frequency response and model for low frequencies.

where g_{31} and d_{31} are piezoelectric constants, $S(t)$ is the time-varying mechanical strain, Y^E is the short circuit elastic modulus and ε is the electrical permittivity. Based on the simple R-C model, the cut-off frequency f_p will be equal to $1/2\pi R_L.(C_P + C_L)$, where R_L is the load resistance, C_L is the load capacitance and C_P is the output capacitance of the transducer given by

$$C_P = \varepsilon \frac{b \times L}{h}. \tag{1.6}$$

According to the high-pass R-C model, the output voltage generated by the piezoelectric transducer at a particular frequency f is given by

$$V_{out}(f) = V_{in}(f).\frac{f/f_p}{\sqrt{1+(f/f_p)^2}} \tag{1.7}$$

which based on equation 1.5 leads to

$$V_{out}(f) = \frac{S(f)Y^E d_{31}h}{\varepsilon}.\frac{f/f_p}{\sqrt{1+(f/f_p)^2}} \tag{1.8}$$

where $S(f)$ denotes the Fourier transform of the strain-signal. It can be seen from equation 1.8 that the piezoelectric output voltage $V_{out}(f)$ is negligible at frequencies $f << f_p$ and is only proportional to $S(f)$ when $f > f_p$. This highlights the difficulty of measuring static strain (low-frequency strain) using piezoelectric transducers. However, quasi-static strain (with frequency less than 20 Hz) can be measured if f_p can be made lower than 20 Hz, which would imply increasing C_L, C_P and R_L.

1.4.3 Previous Work & Problem

Earlier in [10] we had proposed and successfully demonstrated an asynchronous self-powered sensor based on the integration of piezoelectric transducers and floating-gate injectors. The piezoelectric transducer was used for sensing strain-variations and the same sensing signal was used for powering the computation and non-volatile storage functions implemented by the floating-gate injectors. In [11, 12] we used the self-powering approach to design mechanical usage monitors and event detectors.

There were two major disadvantages which obviates the use of the method in [13, 14], for

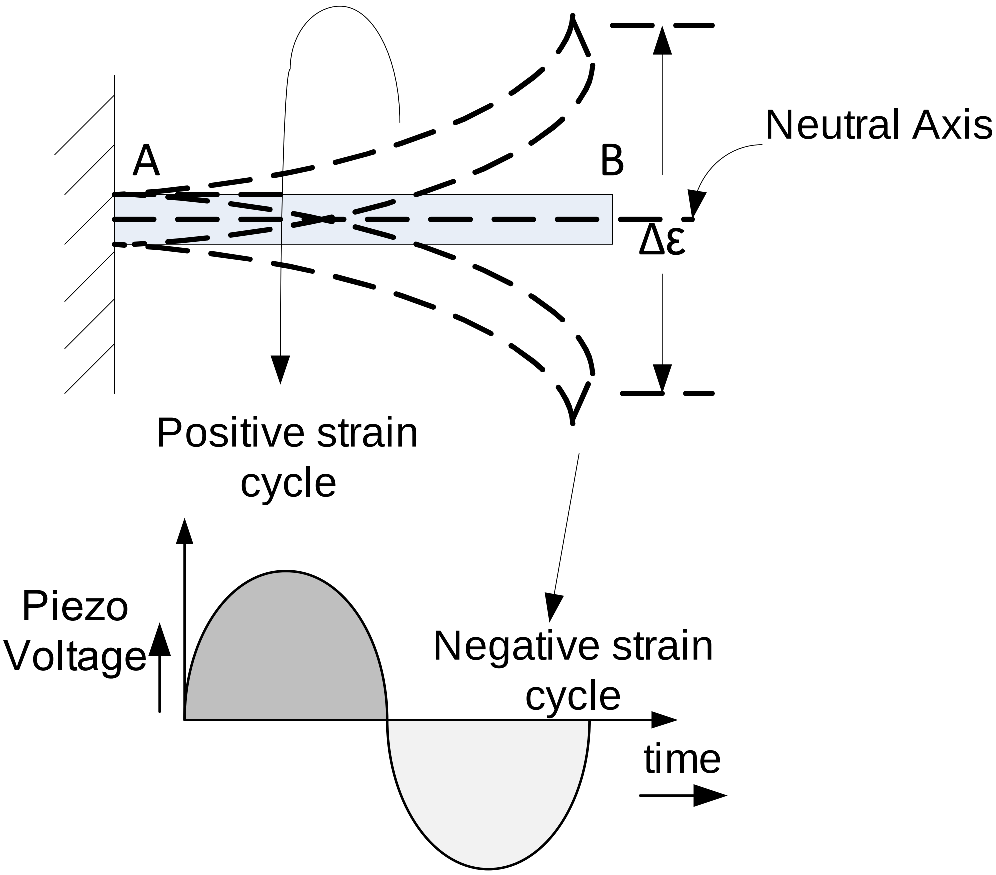


Figure 1.15: Mechanical deformation of a cantilever beam and the corresponding positive and negative voltage cycles generated by a piezoelectric transducer attached to the cantilever.

designing self-powered strain-gauges: (a) the response of the floating-gate injector in [10] is log-linear with a dynamic range less than 100mV; and (b) the resolution of the sensor was measured to be less than 5 bits. At such low-resolutions and low dynamic-range, measurement of quasi-static strain variations using a piezoelectric transducer is difficult. Another challenge in measuring quasi-static-strain using piezoelectric transducer is that at ultra-low-frequency (less than 20Hz) the output of the transducer is capacitively coupled, as a result of which the signals of interest get filtered out. To address both the challenges while ensuring self-powered operation, in this thesis, we show how a constant current loading in conjunction with a differential operation could surmount these challenges.

1.5 List of Contributions of the thesis

The contributions of this thesis are the following:

- The use of an ultra-linear floating-gate injector for quasi-static strain measurements. The injector which was first reported in [13, 14] and was shown to achieve a resolution greater than 13 bits and a dynamic range greater than 4V.

- The use of a differential configuration of the linear injector circuit which measures quasi-static-strain by integrating the difference between the energy content of the piezoelectric signal generated during the positive and negative strain-cycles. The principle of operation is shown in Fig. 1.15 for a piezoelectric cantilever beam which is subjected to mechanical strain variations. The positive and negative cycles of the electrical signal correspond to the deformation of the cantilever in each direction about the neutral-axis or the resting state. If the cantilever returns back to its resting state (no stored potential energy), the energy transduced during the positive strain-cycle should be equal to the energy transduced during negative strain-cycle. However, if the transducer is subjected to deformation (or subjected to static-strain), there will be a difference between the positive and negative signal cycles which if integrated should measure the magnitude of static-strain.

- A system-on-chip solution has been designed that integrates the linear floating-gate injectors with high-voltage charge-pumps, analog references, ring oscillator and digital programming circuits.

- Extensive experiments with the system-on-chip prototypes fabricated in a 0.5μm standard CMOS process and piezoelectric material (PZT) have been performed using a bench-top mechanical test setup and the results have been calibrated against standard strain-gauge measurements.

- Another fundamental contribution of this thesis is a novel compressive self-powering technique that overcomes the input threshold effect of most self-powered sensors. By using a non-linear impedance circuit at the output of the piezoelectric transducer and by using programmable level-crossing circuit and offset cancellation circuits, the thesis demonstrates an extended self-powering range greater than 40dB.

- An automated programming and calibration of the sensor has been developed comprising of an FPGA and MATLAB interface. This test platform is also useful for long-term, automated reliability testing of the self-powered piezo-floating-gate sensors.

The thesis is organized as follows: Chapter 2 describes the architecture of the piezo-floating-gate linear injector and the usage of it as a differential strain-gauge with its underlying mathematical model. Also a simulation model to simulate the linear injector in cadence has been provided. Chapter 3 presents a complete system-on-chip which integrates the linear injectors with high-voltage charge-pumps and digital command and control circuitry. In Chapter 4 the test results of the self-powered strain-gauge with piezoelectric material and commercial strain-gauge have been discussed. Chapter 5 describes the non-linear impedance circuit, programmable level-crossing circuit and offset cancellation circuits. Calibration and floating gate programming algorithms have been discussed in Chapter 6. Finally, chapter 7 summarizes the thesis and discusses about future research directions.

Chapter 2

Linear Floating gate Injector based strain gauge system

As mentioned in Chapter 1, here the operational principle of the differential strain-gauge has been discussed in details. In the next section the linear floating-gate injector, which is the key element of the strain-gauge, has been described. Then a simulation model to simulate the floating-gate injectors has been provided.

2.1 Differential strain-gauge Architecture

The first-order analysis presented here is based on the Fig. 2.1, using the quasi-static model given by the equation 1.5. The voltage generated by the piezoelectric voltage $V(t)$ can be written in its differential form as

$$V(t) = V^{+}(t) - V^{-}(t) \tag{2.1}$$

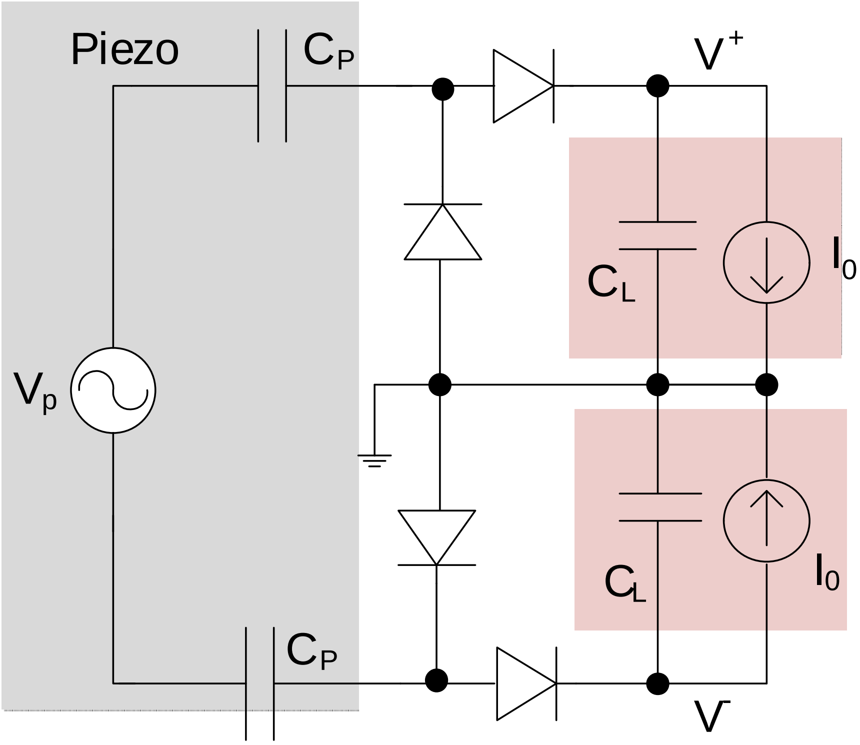


Figure 2.1: Differential current load connected to a full-wave rectifier and a piezoelectric transducer.

where $V^+(t)$ ($V^-(t)$) denote the source voltages(assuming zero diode drops) during the positive (negative) strain-cycles (see Fig. 1.15). Thus, $V^+(t), V^-(t) \geq 0$ and $V^+(t)V^-(t) = 0, \forall t$. Based on equation 1.8, the differential voltages are related to the differential strains $S^+(t), S^-(t) \geq 0$ according to

$$S^+(t) = \frac{\varepsilon V^+(t)}{Y^E d_{31} h} \text{ and } S^-(t) = \frac{\varepsilon V^-(t)}{Y^E d_{31} h} \tag{2.2}$$

where the instantaneous strain $S(t)$ is given by

$$S(t) = S^+(t) - S^-(t). \tag{2.3}$$

The quasi-static strain (average strain) computed over a measurement period T is then given by

$$S(T) = \frac{1}{T}\int_0^T S(t)dt \tag{2.4}$$

which using equation 2.2 leads to

$$S(T) = \frac{\varepsilon}{Y^E d_{31} h T}\left[\int_0^T V^+(t)dt - \int_0^T V^-(t)dt\right]. \tag{2.5}$$

Equation 2.5 can be reformulated using the piezoelectric capacitance C_P as

$$S(T) = \frac{\varepsilon}{Y^E d_{31} h C_P}\left[\frac{1}{T}\int_0^T C_P V^+(t)dt - \frac{1}{T}\int_0^T C_P V^-(t)dt\right]. \tag{2.6}$$

Thus, measurement of the quasi-static strain could be achieved if the differential charges

$$\begin{aligned} Q^+(T) &= \frac{1}{T}\int_0^T C_P V^+(t)\mathrm{d}t && (2.7)\\ Q^-(T) &= \frac{1}{T}\int_0^T C_P V^-(t)\mathrm{d}t && (2.8)\end{aligned}$$

could be directly measured. Note that $Q^{+(-)}(T)$ represents the average charge transferred by the piezoelectric transducer to the load, and because $V^+(t) \geq 0$ with $V^+(t)V^-(t) = 0, \forall t$, $Q^{+(-)}(T)$ can be used to estimate the L_1 norm of the strain-signal $||S(T)||_1$ given by

$$||S(T)||_1 = \frac{1}{T}\int_0^T |S(t)|\mathrm{d}t = \frac{1}{T}\int_0^T \left(S^+(t) + S^-(t)\right)\mathrm{d}t. \tag{2.9}$$

The measure of L_1 norm of the strain-signal is important because it is an indicator of the strain-energy dissipated through the structure, and hence could be used for understanding progression of mechanical damage. The quantities $Q^{+(-)}(T)$ could be measured in a self-powered manner using the circuit shown in Fig. 2.1. It consists of a half-wave rectifier for producing $V^{+(-)}(t)$ and a constant current-sink I_0 for measuring $Q^{+(-)}(T)$. For the sake of simplicity, we will neglect the threshold voltage of the diode and the minimum voltage required for the current-sink to be operational. This is a valid assumption as many piezoelectric transducers can generate open-load voltages much larger than the threshold voltage of the diode. Therefore

$$\mathrm{I}_0.\tau^{+}(T) = \frac{1}{T}\int_0^T C_P V^{+}(t)\mathrm{d}t \tag{2.10}$$

$$\mathrm{I}_0.\tau^{-}(T) = \frac{1}{T}\int_0^T C_P V^{-}(t)\mathrm{d}t \tag{2.11}$$

where $\tau^{+(-)}(T)$ is the total discharge time. Combining equations 2.11 and 2.6 the following relationship is obtained:

$$S(T) = \frac{\varepsilon I_0}{Y^E d_{31} h C_P}\left[\tau^{+}(T) - \tau^{-}(T)\right] \tag{2.12}$$

The self-powered differential strain-gauge continuously computes the parameters $\tau^{+}(T)$ and $\tau^{-}(T)$ and continuously stores them in a non-volatile memory formed by a floating-gate transistor. In the next section we briefly describe the principle of operation of an ultra-linear floating-gate injector which was first reported in [13].

2.2 Linear Floating-gate Injector

The circuit level schematic and the principle of operation of the linear floating-gate injector is shown in Fig. 2.2(a). The circuit consists of a floating-gate pMOS transistor M_{fg} whose source is driven by a constant current source I_{ref} which is powered by either a piezoelectric transducer or by some other energy source V_{dda}. Note that both the energy sources are isolated by a diode which

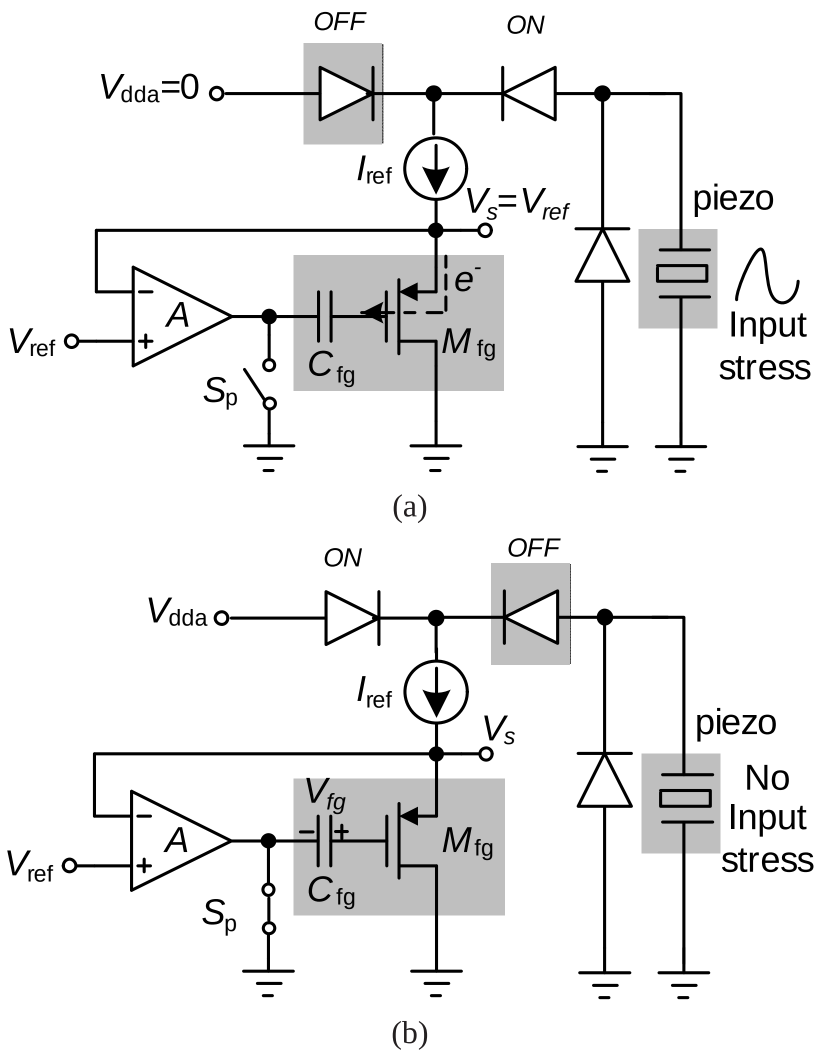


Figure 2.2: Linear Injector circuit : (a) Sensor mode when piezo power is available; (b)Read-out mode using different power source.

allows V_{dda} to supersede the signal generated by the piezoelectric transducer. The polysilicon gate of the pMOS transistor is electrically insulated by silicon-dioxide (hence the name “floating-gate”), therefore, any electron injected onto the gate is retained for a long period of time (8 bits precision for 8 years) [19, 20]. Electrons are injected onto the floating-gate using an impact-ionized hot-electron injection (IHEI) process which involves applying $V_{\text{sd}} > 4.2$V (in 0.5-μm CMOS process) across the source and the drain terminal. The large electric field near the drain of the pMOS transistor creates impact-ionized hot-electrons whose energy when exceeds the gate-oxide potential barrier ($\approx$3.2eV) can get injected onto the floating-gate. IHEI current, I_{inj}, in a pMOS transistor is dependent on the transistor source current I_{s}, the source-to-drain voltage V_{sd} and the gate-to-drain voltage V_{gd} across the transistor. This dependence can be expressed in functional form as

$$I_{\text{inj}} = f\left(I_{\text{s}}, V_{\text{sd}}, V_{\text{gd}}\right), \tag{2.13}$$

where $f(\cdot)$ is an arbitrary function that could be empirically determined [20]. However, the circuit in Fig. 2.2 achieves stable and ultra-linear injection using a negative feedback loop formed by the opamp A and the floating-gate transistor M_{fg}. The source current is held constant at I_{ref} which ensures that the source-to-gate voltage V_{sg} remains constant during injection. When switch S_P is open,the feedback is enabled and opamp A ensures that the source-to-drain voltage V_{sd} is held constant to V_{ref}. Thus, according to equation 2.13 the injection current I_{inj} remains constant. The amount of charge injected onto the floating-gate and hence the decrease in floating-gate voltage V_{fg} is proportional to the duration for which the source current I_{s} is activated and S_P is open. This can be expressed as

$$\Delta V_{fg} = \frac{1}{C_T}\int_0^T I_{inj}dt = \frac{I_{inj}}{C_T}\tau(T) \tag{2.14}$$

where τ is the duration of injection and C_T is the total floating-gate capacitance. The change in floating-gate voltage ΔV_{fg} could be measured by closing the switch S_{P} which breaks the feedback loop by shorting the other terminal of C_{fg} to ground. Because the source current I_{ref} is constant, $\Delta V_s = \Delta V_{fg}$ which is read-out through a unity-gain buffer. Figure. 2.3 shows the measured response of a linear injector where the source voltage V_{s} is first initialized to 4.3V (using FN tunneling), $V_{\text{ref}} = 4.8V$ and $I_{\text{ref}} = 30nA$. The piezoelectric transducer is emulated by applying a 50ms long pulse signal (amplitude $V_{\text{dd}} = $6.5V) after which the switch S_{P} is turned ON and the source voltage

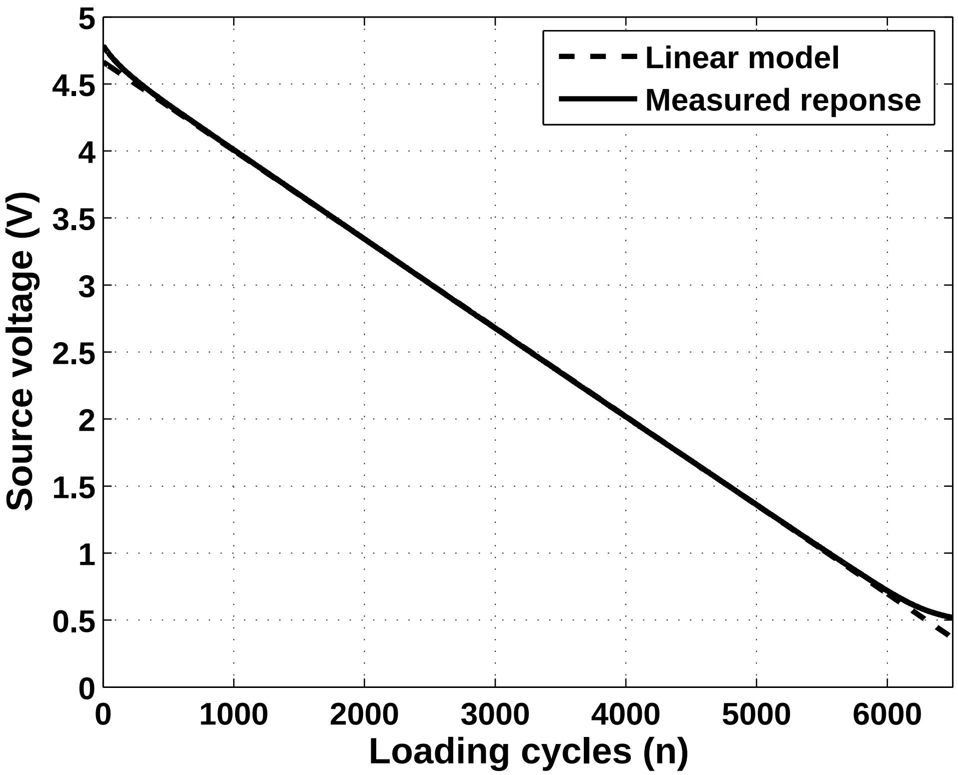


Figure 2.3: Measured output range for linear injection circuit.

V_s is measured. Figure. 2.3 shows that the change in V_s is linear with respect to the number of applied pulses. The deviation from the linear injection model occurs at the end points of the operating voltage and is due to the finite operating range of the amplifier A. This shows that the linear injector has a linear range of almost 4V. Also, in [13] the resolution of the linear injector was measured to be greater than 13 bits.

2.3 Linear Injector Based Differential Strain-Gauge

Fig. 2.4 shows the architecture of the differential strain-gauge using the linear floating-gate injectors. Half-wave rectifiers formed by diodes extract the positive and negative voltage cycles V^+ and V^- and the differential injectors estimate the quasi-static-strain according to equations 2.12

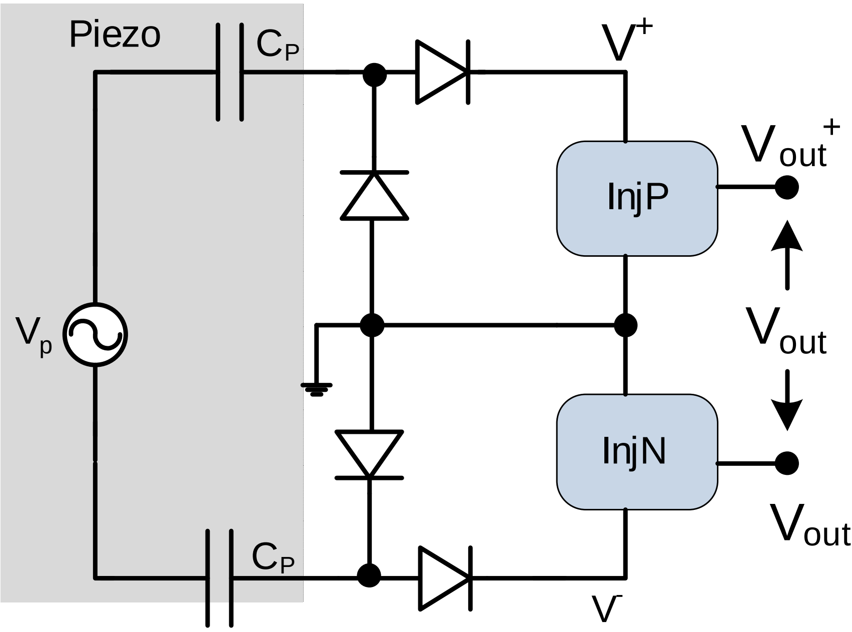


Figure 2.4: Differential Sensor Architecture.

and 2.14. The differential output V_{out} in Fig. 2.4 can be expressed as

$$V_{out}(T) = V_{out}^+(T) - V_{out}^-(T) \tag{2.15}$$

$$V_{out}^+(T) = \alpha S^+(T) \tag{2.16}$$

$$V_{out}^-(T) = \alpha S^-(T). \tag{2.17}$$

where the gain of the strain-gauge α is given by

$$\alpha = \frac{Y^E d_{31} h C_P}{\varepsilon} \frac{I_{inj}}{I_0 C_T}. \tag{2.18}$$

Note that the single ended outputs V_{out}^+ and V_{out}^- are historical indicators of strain-energy which are stored in a non-volatile fashion on the floating-gate injectors. The gain α in 2.18 is a product of two terms; the first term is determined by the property of the piezoelectric transducer; the second

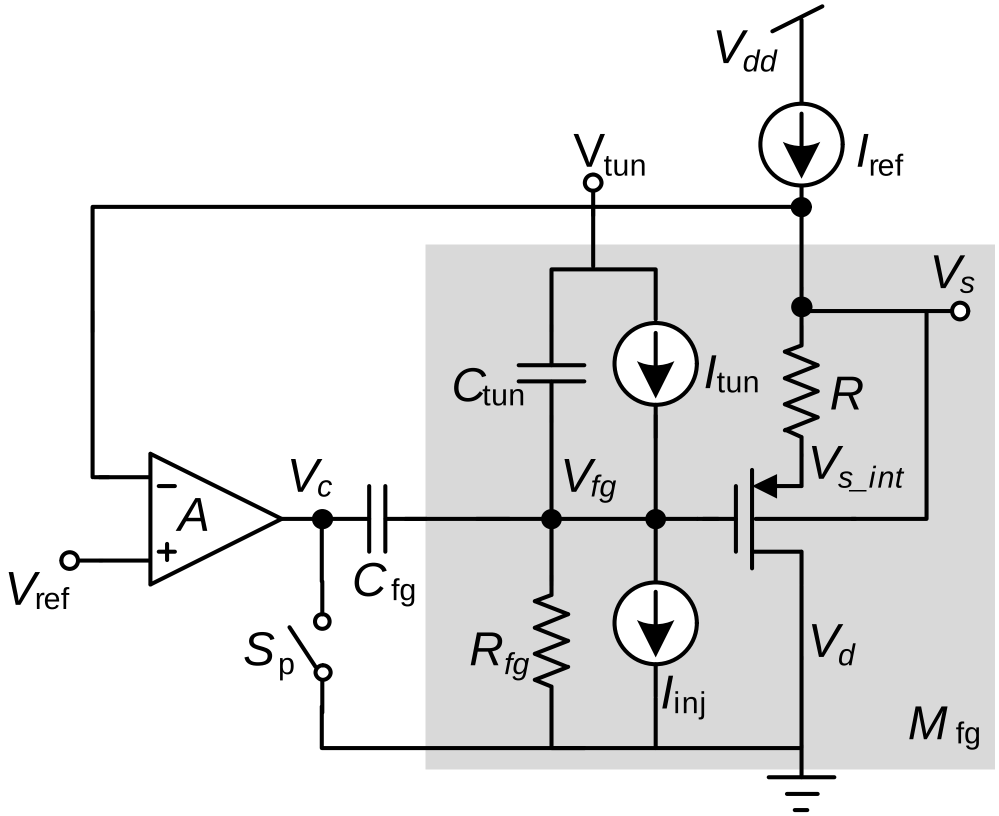


Figure 2.5: Linear Injector Simulation model.

term represents the electrical parameters that can be effectively controlled.

2.4 Linear Injector Simulation Model

Normal BSIM MOS models do not have injection or tunneling models. To simulate the linear injector in cadence, they have been modeled in Verilog-A using empirical relations as voltage dependent current sources. Fig. 2.5 shows the cadence model for the linear injector. I_{inj} is the injection current source modeled in verilog-A according to the equation

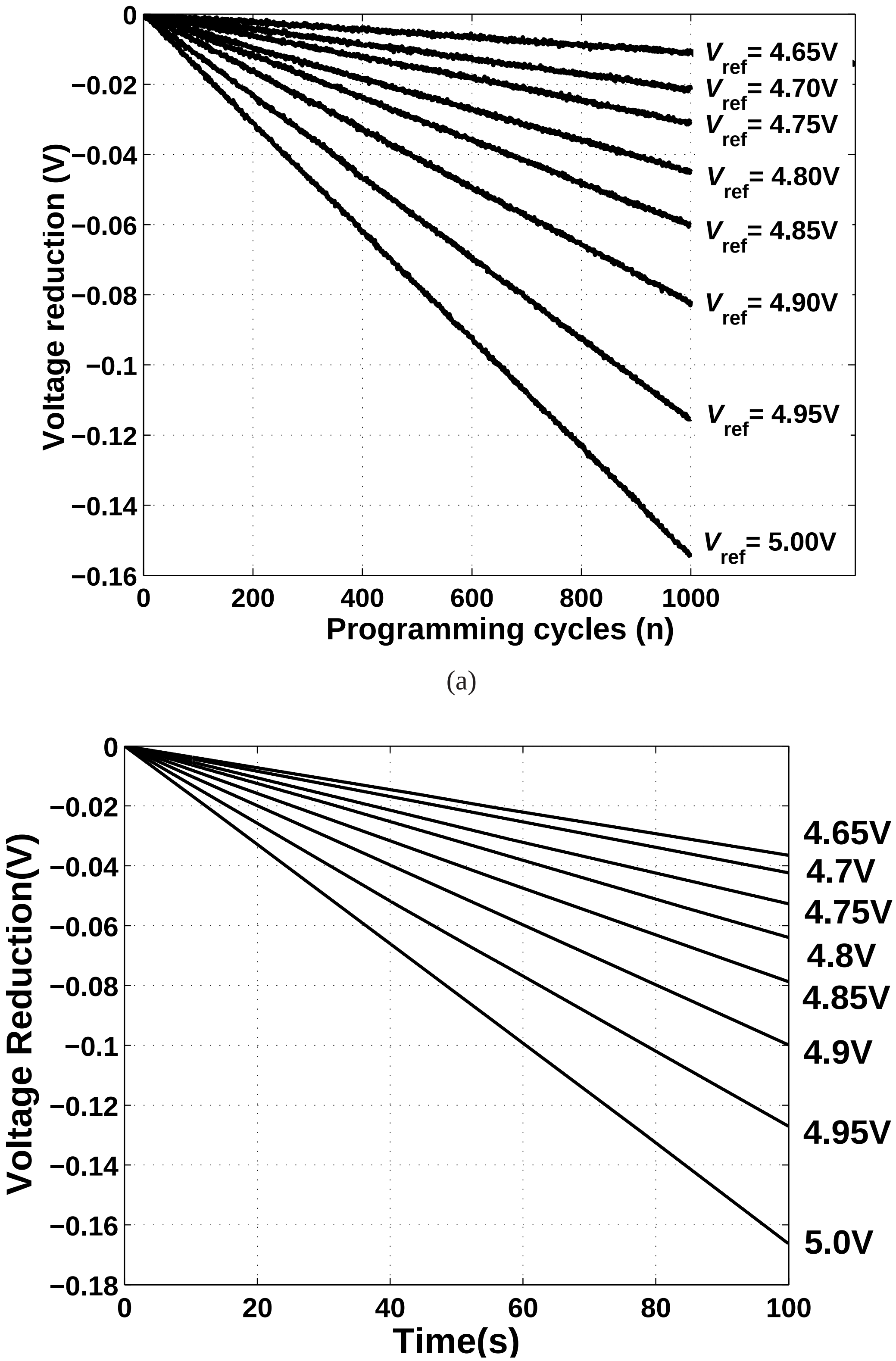
$$I_{inj} = a.I_{ref}.(1 + b.I_{ref}).exp(k.V_{sd}) \tag{2.19}$$

where a, b and k are empirical parameters obtained from measurement results. In our case the values have been taken a=3×10^{-23}, b=5×10^{6} and k=6.55. V_{sd} is the source to drain voltage and I_s is the source current of the PMOS. There would be injection current only when I_s is non-zero and when $V_{sd} > 4.3V$. At all other conditions $I_{inj} = 0$. Similarly tunneling has been modeled too according to the following relation

$$I_{tun} = c.exp(-d.(V_{tun} - V_{fg})) \tag{2.20}$$

where c=9.35×10^{8} and d=800 in our model, V_{tun} is the tunneling voltage and V_{fg} the floating gate voltage. The verilog-A models have been provided in Appendix. C. C_{tun} is the tunneling capacitor which is a small overlap capacitance($\sim$1fF). R_{fg} is a very high valued resistance($10^{30}\Omega$) to provide a dc path to ground from the floating node. This ensures that the simulation converges and yet the charge remains there for a long time. The initial floating gate voltage is specified as the initial condition(.ic) of the capacitor. Resistance R is a very small resistor(1 Ω) to create a branch between V_s and V_{s_int} to measure I_{ref} for I_{inj}. To verify the model, the simulation results were compared with measured results. First, I_{ref} was kept fixed at 50nA and V_{ref} was varied from 4.65V to 5V. The results are shown in Fig. 2.6. Then, V_{ref} was kept fixed at 4.9V and I_{ref} was varied from 30nA to 100nA. The results are shown in Fig. 2.7.

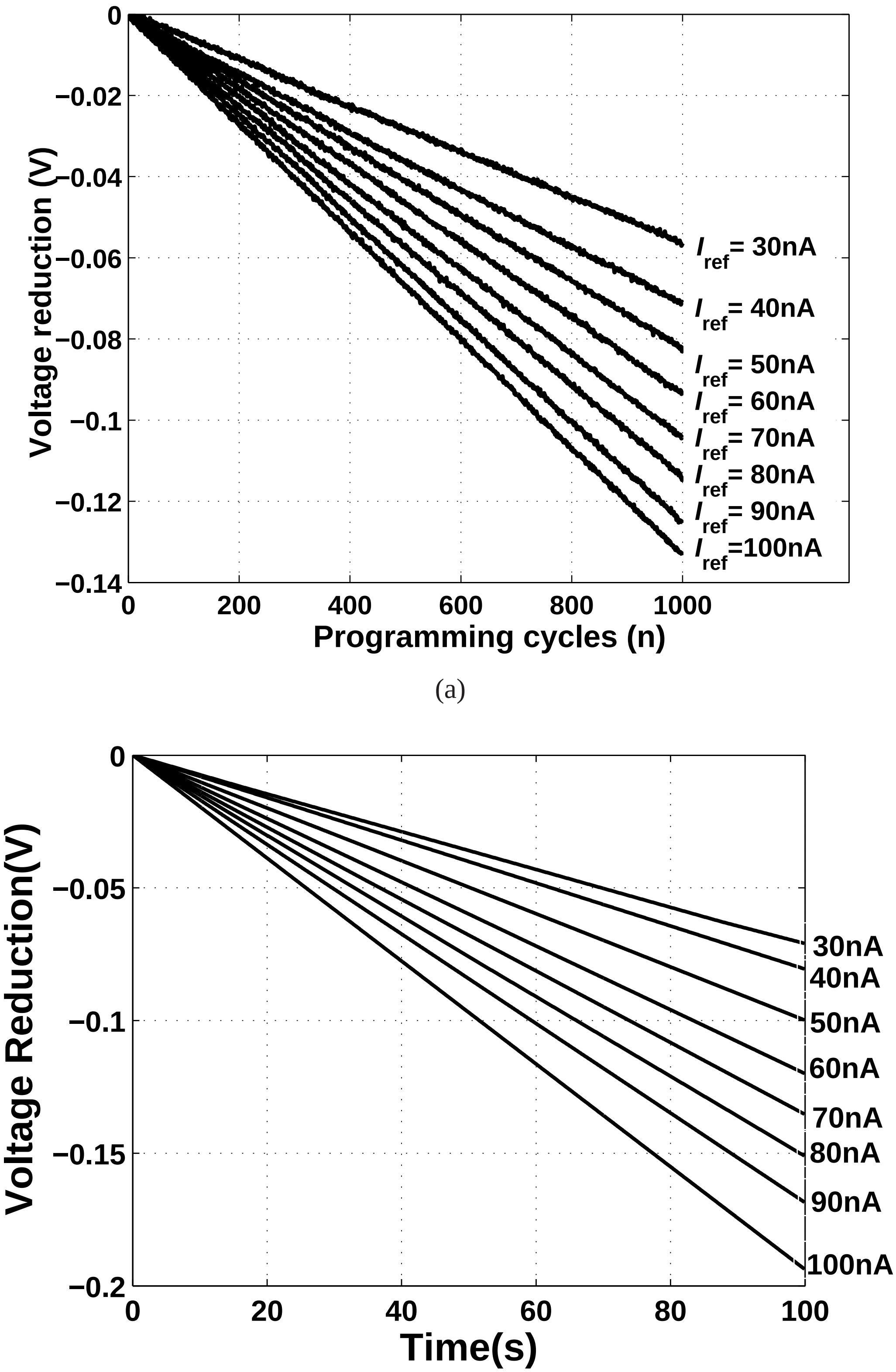
This model has been proved useful specially for complex circuits involving floating-gates. In each case we have got a close match with the measured results.



(a)

(b)

Figure 2.6: (a)Measured injector output varying V_{ref} keeping I_{ref}=50nA; (b)Simulation result with same condition.



(a)

(b)

Figure 2.7: (a)Measured injector output varying I_{ref} keeping V_{ref}=4.9V; (b)Simulation result with same condition.

Chapter 3

System Architecture

This chapter describes a complete strain-gauge system implemented in standard CMOS 0.5μ technology. Earlier version of the sensor system was with a RFID based wireless read-out. The contribution of the thesis work from circuit design point of view was to separate the sensor system from the RFID system and make it an independent system. This makes the sensor much more dynamic from application point of view as it can be interfaced with a commercial RFID system. A major improvement has been done on the reliability point of view by not making the tunneling node an external pin and controlling the tunneling voltage via a control loop. This has increased the life-time of the sensor dramatically.

3.1 Top View

Fig. 3.1 shows the architecture of the self-powered strain-gauge where the injector is powered in two ways: (a) when the strain-gauge sensor is in self-powering mode, it is powered directly using the signal generated by the piezoelectric transducer; and (b) when the sensor is being interrogated or programmed, the sensor module is powered by an external DC supply via a high-voltage charge-

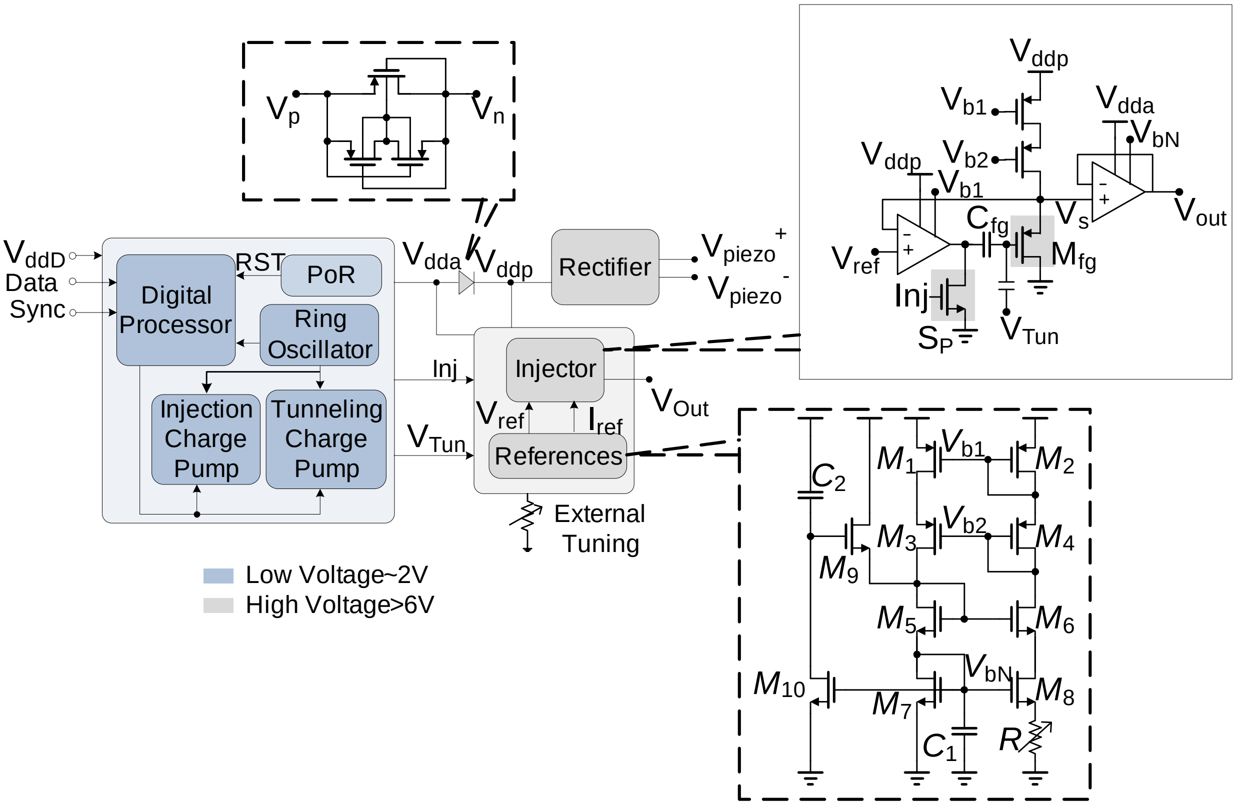


Figure 3.1: Full system architecture.

pump. Note that this demarcation in powering is necessary because self-powering can only support continuous monitoring and data-logging. However, the power is not enough to energize digital control circuits. The external DC power used for programming, digital command and control could be supplied by a telemetry interface. For example in [10], we reported an RFID based telemetry interface that could provide a 2V regulated supply.

3.2 Power On Reset

The system-on-chip architecture includes a POR(Power On Reset) module as shown in Fig. 3.2 that generates a reset (initialization) signal (RST) that is used for zeroing all the internal register states of the digital modules, once the input supply has reached a sufficient value. The POR module

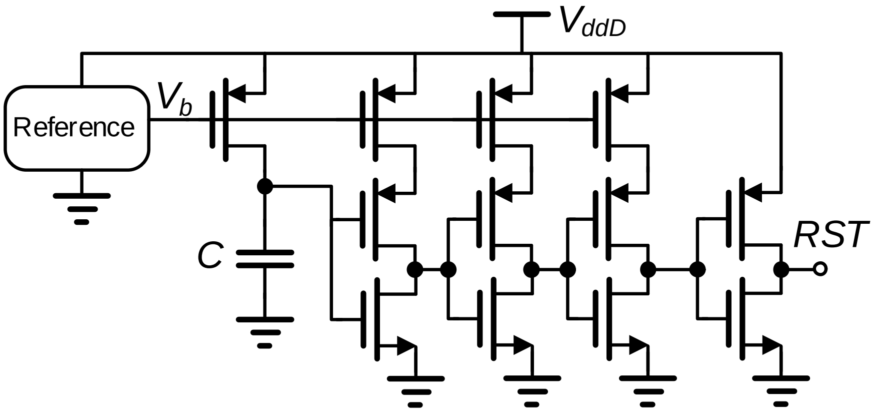


Figure 3.2: Power On Reset.

also ensures that the strain-gauge sensor does not load the external supply unless enough energy can be scavenged to generate the DC voltage. The POR architecture used is based on the charging of a capacitor C by a constant current source. The current reference starts only when the supply reaches above 1.5V. When the voltage across the capacitor crosses a threshold, POR becomes high. Typical on time is $30\mu s$.

The ON time depends on the current that is charging the capacitor, the capacitor size and the threshold of the inverter following it. The current is generated by a supply-independent reference like the one shown in Fig. 3.1. The resistance R is kept tunable externally in case we want to change the ON time.

3.3 Digital Processor

The sensor IC has been designed to process commands encoded using commands. There have been two versions of the digital processor. For version-1 the commands are 8-bit words. For version-2, they are 24 bit words. Decoding of the commands is achieved using an externally supplied clock signal (*Sync*/ENVP) and a data signal (*Data*) which encodes the PIE encoded commands.

Table 3.1: Commands supported by the strain-gauge sensor IC(Version-9).

Command	Opcode(Ver-1)(Bin)	Opcode(Ver-2)(Hex)	Function
Cmd1-RST	1100-001-0	C95-222	Resets the shift register to ch1
Cmd2-SHIFT	1100-010-0	C95-481	Shifts to the next channel
Cmd3-TUN	1100-011-1	C95-6A3	Controls Tunneling
Cmd4-INJ	1100-100-0	C95-814	Controls injection

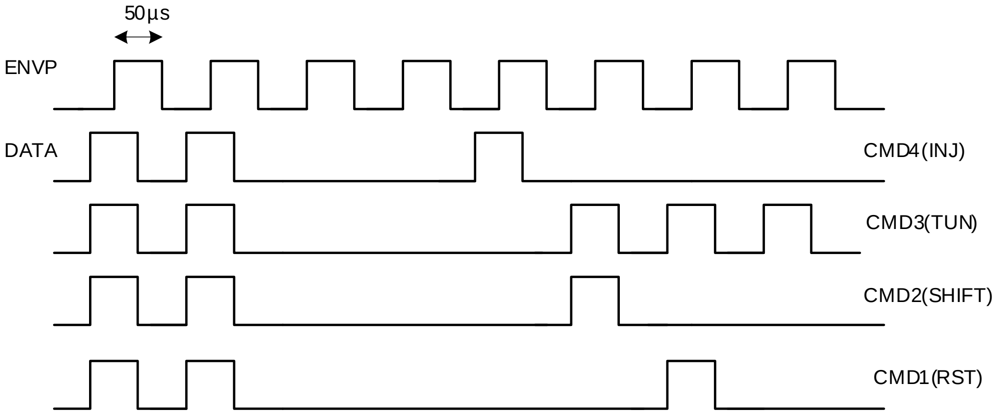


Figure 3.3: Timing diagram for the DATA and ENVP(Sync) for version-1.

Table 3.1 summarizes the commands used for programming and interrogating the sensor IC and the table also provides a brief functional description of the commands. In Fig. 3.3 the timing diagram for all the 4 commands in version-1 with the synchronization between the *Data* signal and the *Sync* signal and their frequency requirements are shown. Once *Data* is latched into a command register, a digital state-machine determines the validity of the command and synchronizes different digital control signals as shown in Fig. 3.4. The internal synchronization clock operates at 200 KHz and is generated using an on-chip three-stage current-starved ring-oscillator.

The change from version-1 to version-2 was made to make the command interface more robust. In version-1 each command has 3 parts-4 bits preamble(1100), 3 bits command(001/010/011/100) and an odd parity bit. In version-2 the command is of 24 bits with the first 12 bits preamble and the rest 12 bits command. Each 12 bits have 3 parts in which a 4 bit word have been sent thrice

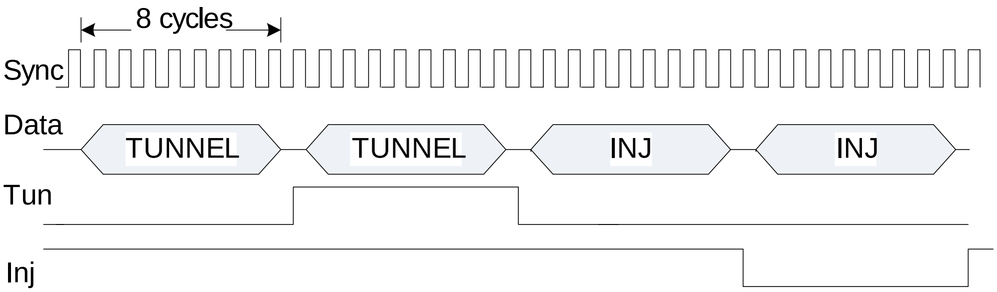


Figure 3.4: Timing diagram for the commands.

changing the order of the bits(3210 2013 0312). The commands for which at least 2 out of 3 is received correctly is accepted as a valid preamble. For example, the preamble 1100 is now sent as 1100 1001 0101(C95). The same goes for the commands 1-4.

3.4 Ring Oscillator

A 3 stage current starved ring oscillator generates a 1 MHz clock(ClkCP) needed for the charge pumps. This clock is also divided by 8 to get the low frequency clock(ClkDig) for running the digital processor. The circuit of the ring oscillator is shown in Fig. 3.5. The sizing of the components are given in the Table. 3.2. The oscillator is a 3 stage current starved one with 3 buffer stages following it to bring the signal to the full swing. The oscillator works for a wide supply range of 1.5V-3V and consumes low current. The simulation and measurement results are shown in Table. 3.3. The design target was deliberately done for a higher frequency as the parasitic capacitances would bring the frequency down which has happened in this case. One example simulation result has been shown in Fig. 3.6 where the oscillator starting time is about 50μs.

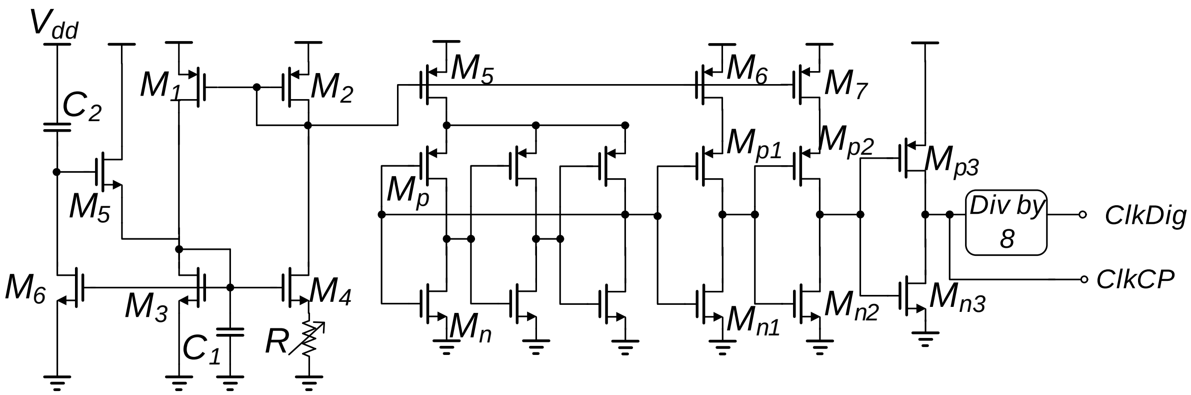


Figure 3.5: Circuit Diagram for Ring Oscillator.

Table 3.2: Sizing for Ring Oscillator.

Component	Size
M_1	$40\mu/5\mu$
M_2, M_3, M_4	$20\mu/5\mu$
M_5	$40\mu/5\mu$
M_p, M_{p3}	$3\mu/0.6\mu$
M_n, M_{n3}	$1.5\mu/0.6\mu$
M_{p1}, M_{p2}	$10\mu/0.6\mu$
M_{n1}, M_{n2}	$5\mu/0.6\mu$
M_6, M_7	$200\mu/5\mu$
R	$450k$

Table 3.3: Specifications for Ring Oscillator with R=450k.

Supply	ClkCP Freq(Sim)	ClkCP Freq(Measurement)	Current(Sim)
2V	1.83 MHz	1.11MHz	2.6 μA
3V	2.07 MHz	1.12MHz	12.25 μA
1.5V	1.75 MHz	1.16MHz	1.64 μA

3.5 Charge Pumps

The sensor IC also integrates two high-voltage charge-pumps. The *Injection Charge Pump* generates supply voltage $V_{dda} > 6V$ and is used for powering the linear-injectors in the interrogation and

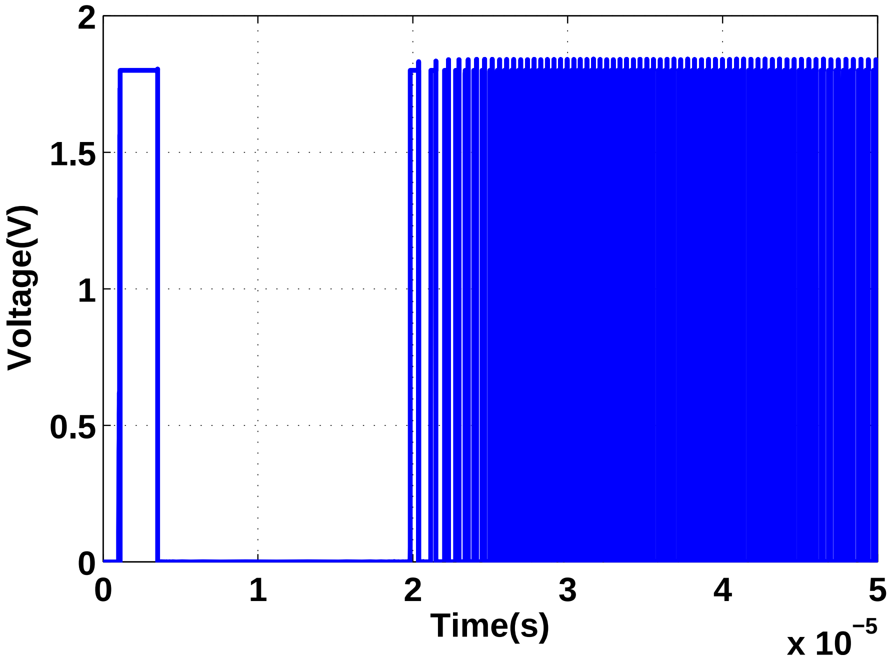


Figure 3.6: Simulation of Ring Oscillator showing ClkCP with V_{dd}=1.8V.

programming mode. The architecture of the charge pump which was first reported in [30] and was also implemented in [10], is shown in Fig. 3.7 and is activated when the RST signal is logic high. The second charge pump referred to as the Tunneling Charge Pump is activated when Tun goes high and generates tunneling-voltage $V_{Tun} > 16V$. The architecture and functioning of the charge pumps are briefly described here.

For each stage of charge pump, four pMOS transistors with a capacitor form the equivalent diode to replace the diode in Dickson's charge pump. The switch bulk technique is used to make sure the bulk potential is always the higher one between source and drain, which eliminates the latch up from the parasitic pnp transistor. The function of charge pump requires four clock signals shown in Fig. 3.7. Clk1 and Clk2 are non-overlapping clocks for charging capacitor C_{p} and Clk3 and Clk4 are auxiliary clocks to reduce forward voltage drop for each stage. The function of the second stage is described which is similar to the other stages. For the second stage, the bulk voltage is connected to V_2 when Clk1 is zero and Clk2 is V_{dd}. Since V_2 is higher than V_1, the gate of conducting pMOS transistor is also connected to V_2 which turns off the transistor and thus no current is flowing from V_2 to V_1. When Clk1 is V_{dd} and Clk2 is zero, the conducting transistor is turned on and starts to charge C_{p} from V_1 to V_2. Without the top capacitor, the charging procedure will stop when $V_2 = V_1 - V_{\mathrm{th}}$ which gives a V_{th} voltage drop for each stage. In the 4-phase charge pump case, Clk3 goes to zero after Clk1 stays at V_{dd}. Due to the capacitive coupling, the gate voltage of conducting transistor becomes $V_2 - kV_{\mathrm{dd}}$ where k depends on the ratio of capacitor C_{g} with parasitic capacitor. Therefore, the voltage at gate has an additional drop from the original value which compensates V_{th} and guarantees V_2 to be charged to V_1 eventually. For an N-stage charge pump, the output voltage can be expressed as

$$V_{\mathrm{out}} = V_{\mathrm{in}} + N \cdot \left(\frac{C_{\mathrm{p}}}{C_{\mathrm{par}} + C_{\mathrm{p}}} \cdot \eta \cdot V_{\mathrm{dd}} - \frac{I_{\mathrm{L}}}{C_{\mathrm{p}} \cdot f} \right), \tag{3.1}$$

where C_{par} is parasitic capacitance, I_{L} is loading current, f is the frequency of clock signals and factor η comes from the energy loss when conducting transistor is turned on. So it can be seen if we want more load current I_{L} from the charge pump with the same voltage, either C_{p} or the frequency f or both have to be increased. Because of the much higher output voltage than the injection charge pump, the tunneling charge pump uses a feedback control loop, shown in Fig. 3.7, to externally set the output voltage which also optimizes power dissipation of the charge pump. A reference voltage V_{ref} is set to 5% of the desired tunneling voltage V_{Tun}. The comparator in the feedback loop ensures that whenever value of the tunneling voltage V_{Tun} is higher than the target value $20 \times V_{ref}$, the charge-pump clock is disabled causing V_{Tun} to decrease. Similarly, the feedback loop ensures V_{Tun} increases when it is lower than $20 \times V_{ref}$. Fig. 3.7 illustrates this in a simulation result, where V_{Tun} oscillates about the target value. We have verified that an 18-stage charge pump can be used to reliably generate up to 25V. Fig. 3.8 shows measured results where V_{Tun} is applied at the tunneling-node of the linear injectors. The output voltage increases as more electrons are removed from the floating-gate and hence could be used to erase the contents of the sensor.

Both the charge pumps have same architecture and sizing. But the injection charge pump does not have a control loop as we did not want to add any extra ripple on the sensor supply. It had two taps, one of which was connected to V_{dda}. A 10pF capacitance was connected to this node to remove the ripple and give a stable supply to the sensor. Also a 12 diodes chain was connected for high voltage protection. So this node would be clipped off approximately at $0.7 \times 12 = 8.4V$.

3.6 Sensor

The sensor block contains an array of linear injectors, references to generate I_{ref} and V_{ref} and level crossing block which generates control signals to turn ON or OFF an injector depending on the supply value.

3.6.1 Reference

The references used are resistor based supply independent current reference. They are well cascoded to tolerate the high supply voltage. There are two separate references to generate I_{ref} and V_{ref}. The reference to generate I_{ref} is shown in the inset of Fig. 3.1. The current is in the order of 40nA which is further divided by half to be used as I_{ref}. The expression for current I_d in the reference is given by,

$$I_d = \frac{U_T}{R} ln(k) \tag{3.2}$$

Table 3.4: Reference Voltage for Sensor at 6V Supply.

Reference	Simulation	Measurement
V_{ref1}	4.68V	4.45V
V_{ref2}	5.02V	4.8V
V_{ref3}	5.48V	5.24V
V_{ref4}	5.83V	5.57V

where, k is the ratio of sizes of the MOS in two branches, R is the resistance and U_T is the thermal voltage=.026V.

V_{ref} is also generated using another similar reference shown in Fig. 3.9. But this reference have 5 stages($M_5 - M_{14}$) of NMOS cascode instead of two. The two highest NMOS gate voltages V_{hi} and V_{low} are then level shifted using pmos level shifters M_{l1} to M_{l4} to generate 4 reference voltages V_{ref1} to V_{ref4}. One of these 4 taps is connected externally to the opamp input V_{ref}. But in this case multiple taps from 4.7-5.5V have been kept in order to take care of process variations and also to control the injector gain. Each tap is obtained by using a pmos source follower as a level-shifter of the previous tap. In the level-shifters there are pmos diodes to protect them from injection due to high supply. Also the current in the level shifters is only 1/3rd of the main reference. The table below(3.4) provide results from simulation and measurement. There are differences between them as the circuits are in sub-threshold. But the resistance R(1.2M) is tunable externally adding a pot with it in series. The total current consumption for this block in simulation is about 122nA.

3.6.2 Injector Array

The injector array shares the same V_{ref}. Their outputs are multiplexed to get one single output V_{out}. One injector circuit is shown in Fig. 3.1. M_{p1} and M_{p2} forms the current source I_{ref}. M_{N1} is switch S_P. The shifting and selection of channels is controlled by the digital processor via a level shifter. When the external supply V_{ddD} is available, only one channel can be selected and injected at a time. But when V_{ddD} becomes zero and piezo power is available, all the channels are capable of injecting provided the level crossing switches are ON. The level-crossing switches(not shown in Fig) are between I_{ref} and M_{fg}. They are pmos switches which get turned ON when the control from the level-crossing block goes low. This is described in the next section.

3.6.3 Level Crossing

One of the specific architectures that is amenable to asynchronous self-powered signal processing is for computing level-crossing statistics. Level-crossing statistics refers to the number of occurrence of the event when an attribute of the signal exceeds a pre-defined threshold.

When the sensor supply 'Vdd' crosses certain thresholds, we want different injectors to be turned ON so that they can be used for event detection. The circuit for one such threshold detector is shown in Fig. 5.9(a). $M_{c1} - M_{c5}$ are cascoded current sources. There is a chain of N pmos diodes connected to one current source. When the supply rises, the diode chain works like a resistance until all the diodes are turned ON fully. So the voltage V_l rises with the supply(with a different slope according to the resistance of the chain) and become stable at one diode drop. Depending on the slope, V_l crosses the threshold of the MOS M_a at a certain supply voltage and turns it ON making V_{sw} goes to zero. V_{sw} is the level crossing switch and it can be controlled by changing N, the number of diodes in the diode chain.

The simulation result for 7 such level-crossing thresholds V_{sw1} to V_{sw7} has been shown in Fig. 5.9(b) when 'VddSensor' is a sinewave of 10Hz and amplitude of 10V.

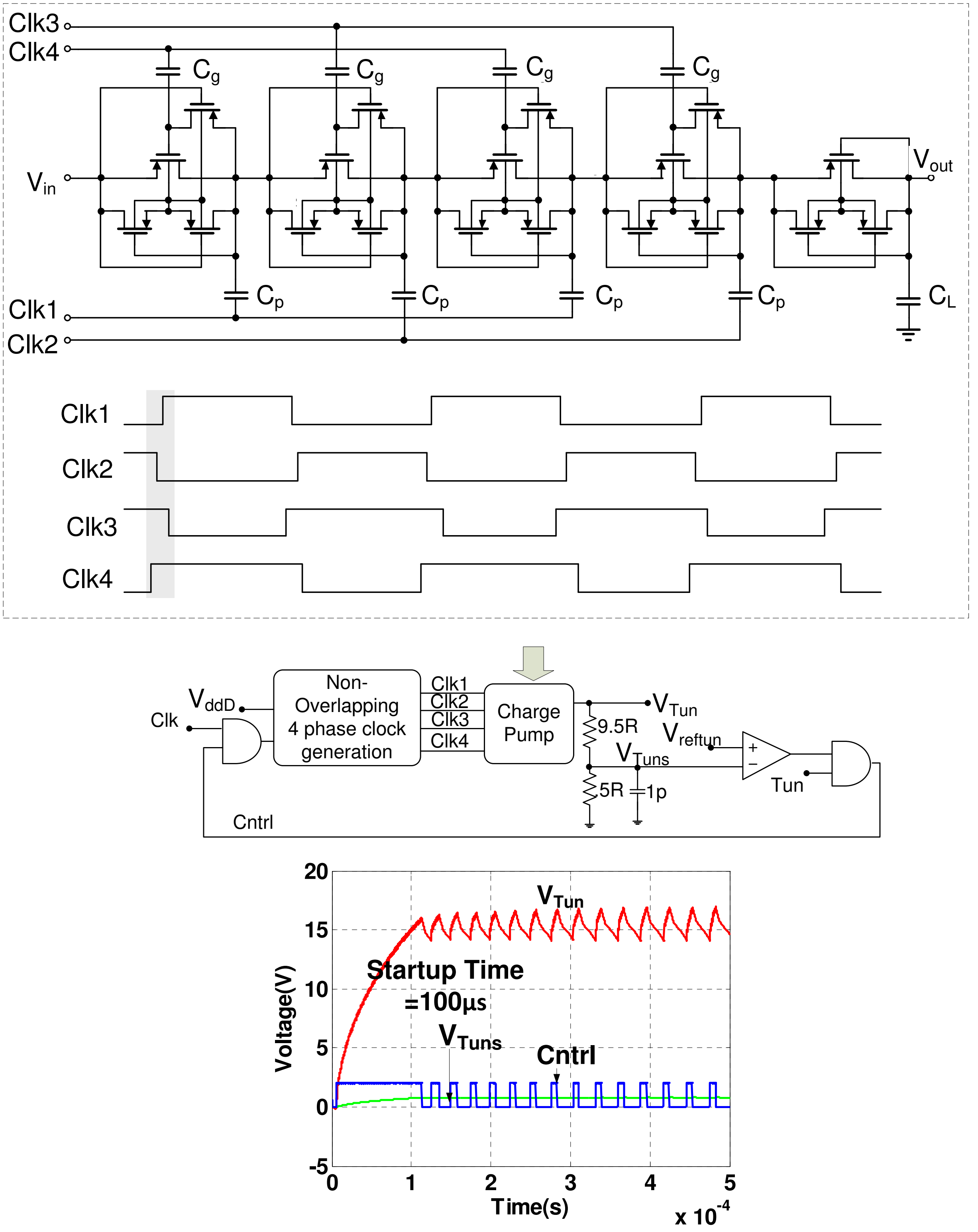


Figure 3.7: Charge Pump control loop and simulation result with 1p load capacitance and 100nA load current.

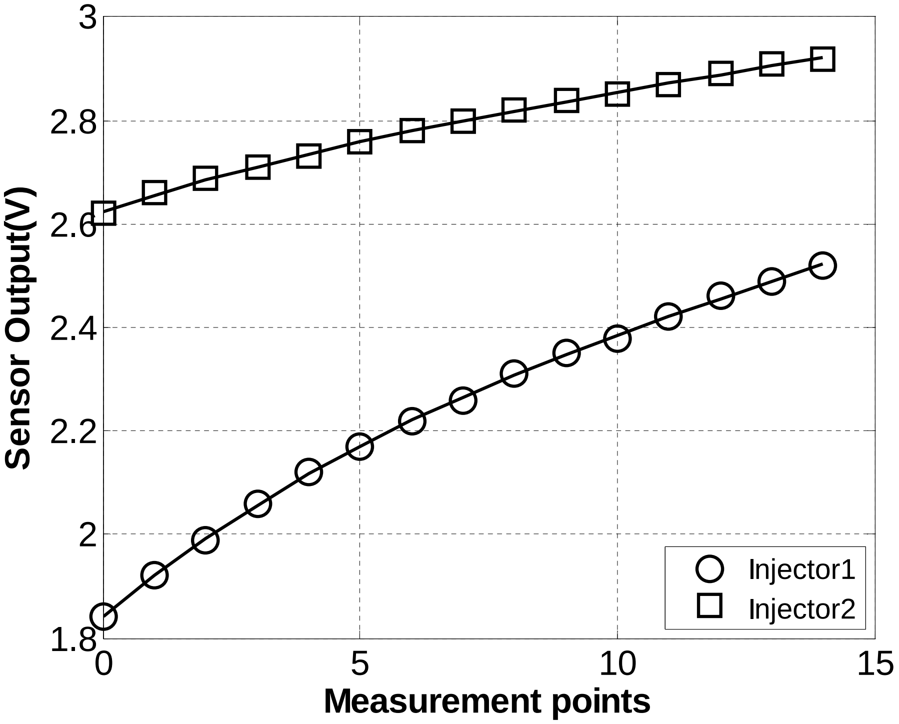


Figure 3.8: Measured result of injector voltage increase when tunneling is ON.

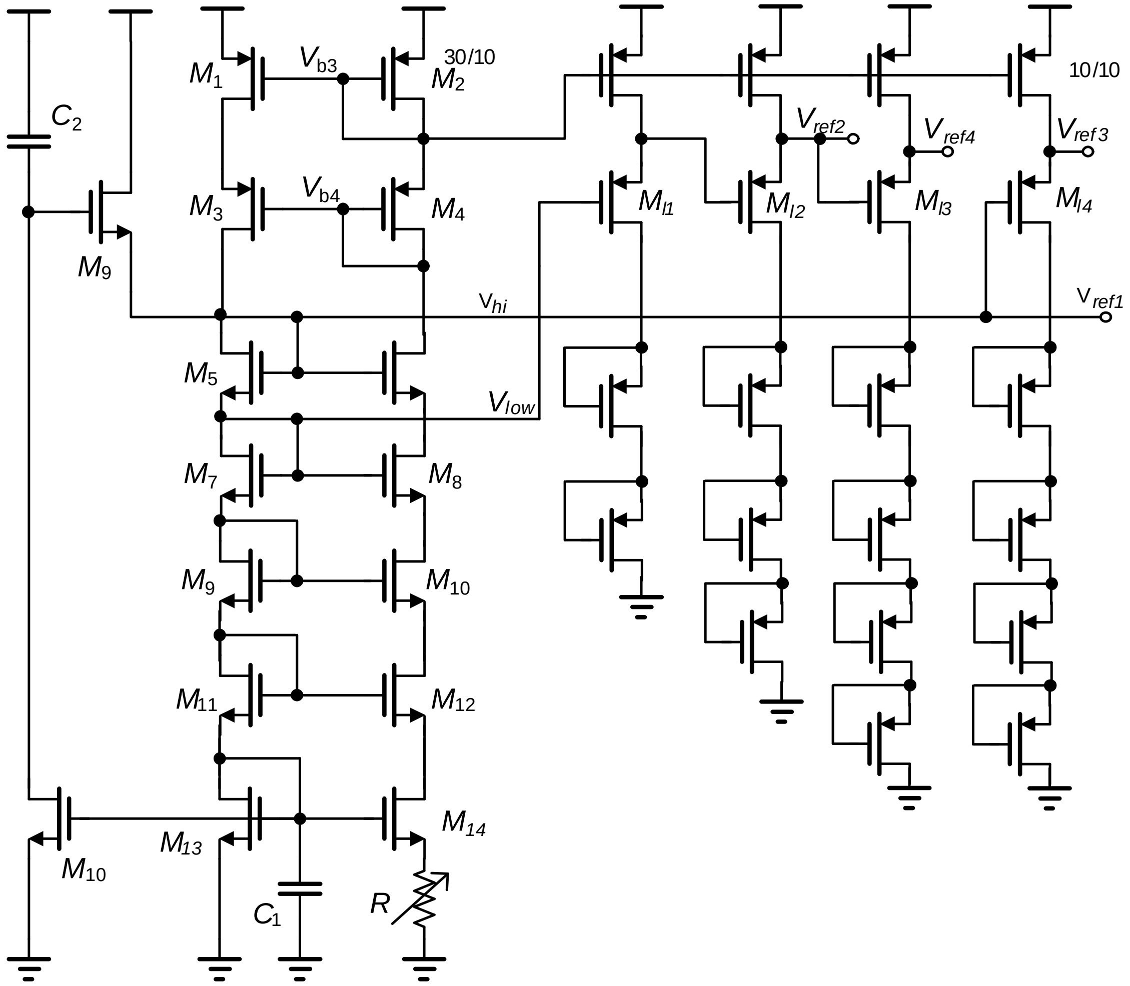
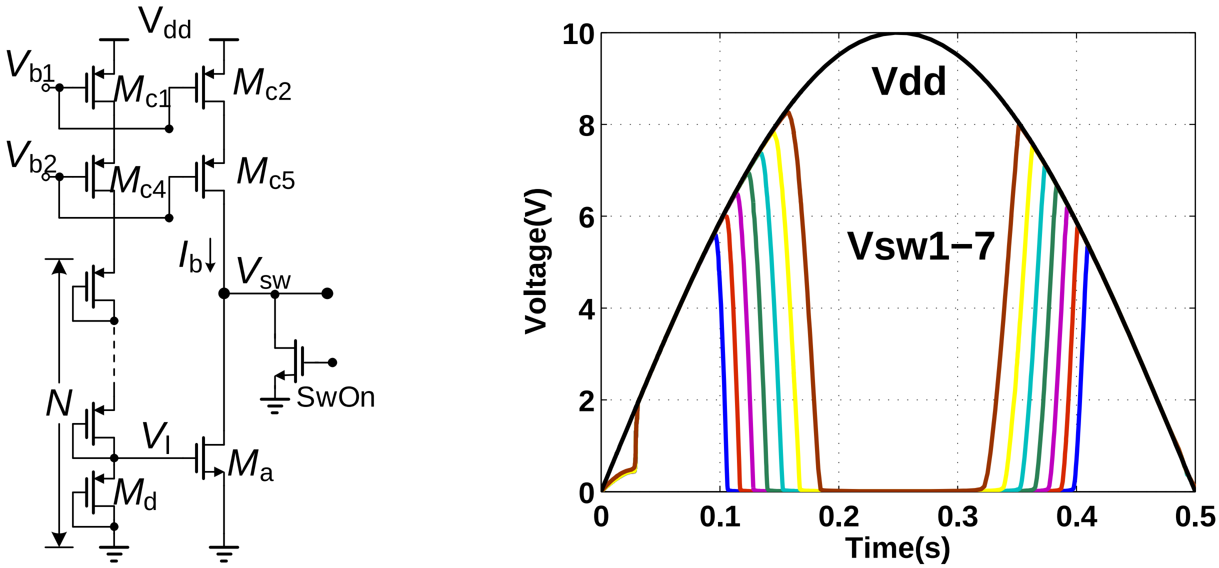


Figure 3.9: V_{ref} generator circuit with 4 taps.



(a) (b)

Figure 3.10: Linear Injector circuit : (a) Level crossing Detector; (b)Level crossing Detector Simulation Result for 7 level-crossing thresholds V_{sw1} to V_{sw7}.

Chapter 4

Testing of the strain-gauge

4.1 Introduction

The self-powered strain-gauge IC has been prototyped in a 0.5-μm standard CMOS process and Fig. 4.1 shows the micrograph of the prototype which occupies an area of 1400μm$\times$1800μm. Table 4.1 summarizes some of the high-level specifications that have been measured using the fabricated prototypes. The measurement results presented in this section are sub-divided into four categories: (a) DC characteristics which validates that the gain of the proposed strain-sensor can be controlled; (b) Dynamic response which validates that the strain-sensor can measure the charge generated by a piezoelectric transducer during positive and negative strain-cycles; (c) Response of the differential strain-sensor using an emulated model of a piezoelectric transducer; and (d) Response of the differential strain-sensor using a PZT transducer and calibrated using a metallic strain-gauge. For the differential measurements linear injectors from two sensor prototypes have been used.

4.2 Strain gauge gain control

The first set of experiments measured the gain of the strain-gauge under different user-defined settings. As shown in equation 2.18, the gain of the proposed gauge $\alpha = \frac{I_{inj}}{I_0 C_T}\frac{Y^E d_{31} h C_P}{\varepsilon}$ is directly proportional to the injection current I_{inj} which in turn is controlled by I_{ref} and V_{ref}. Fig. 4.2 shows the response of a single injector when the parameters I_{ref} and V_{ref} are varied. In Fig. 4.2(a), V_{ref} was fixed at 4.4V and I_{ref} was varied from 20nA to 50nA. For each configuration, the source voltage

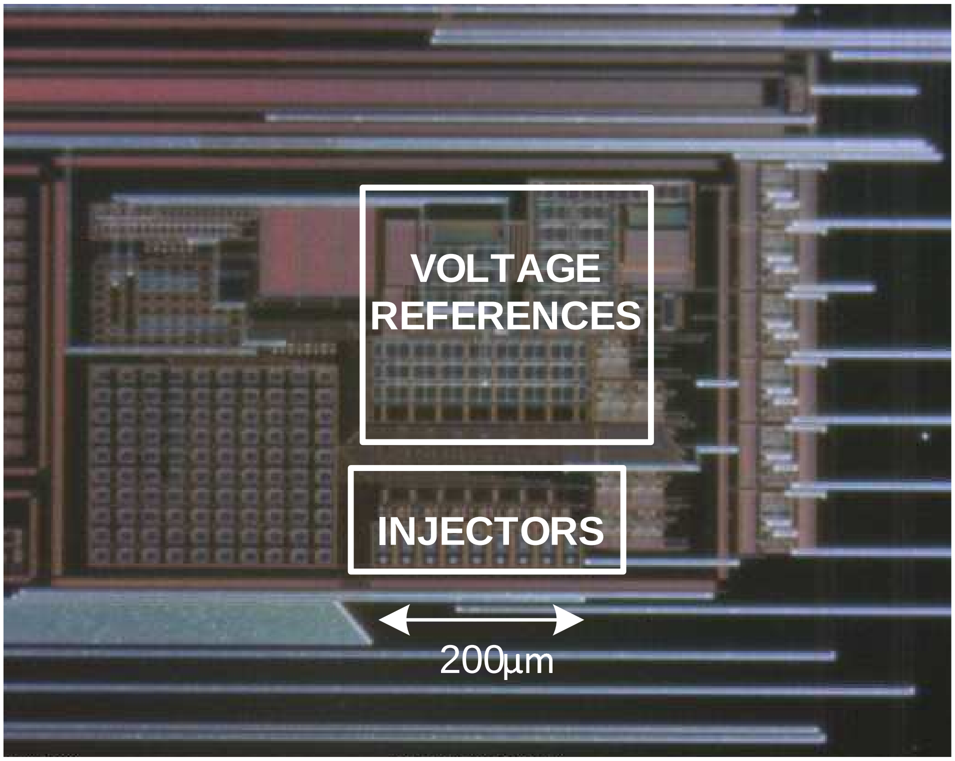
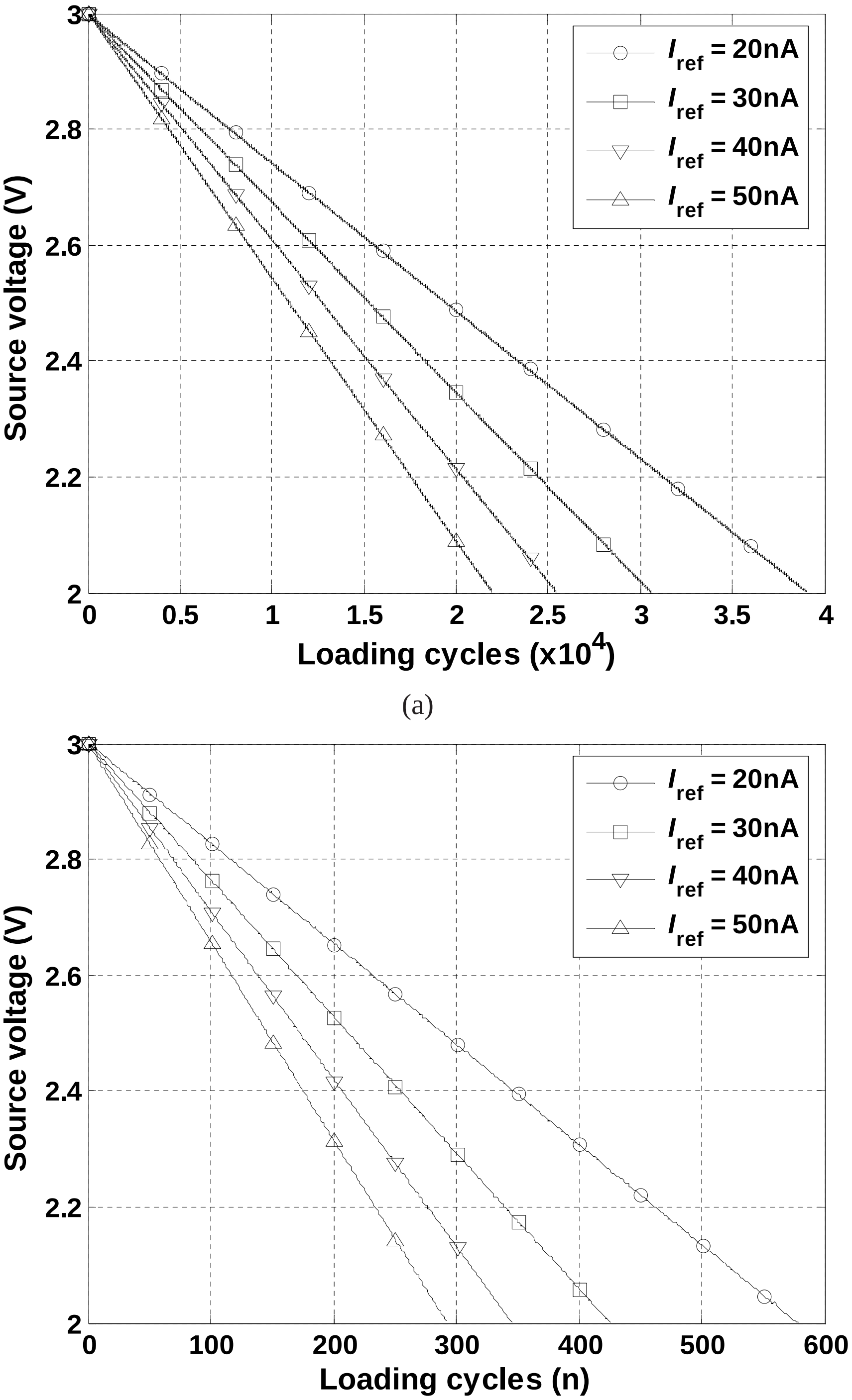


Figure 4.1: The micrograph of the chip.

Table 4.1: Sensor specifications.

Fabrication process	0.5-μm standard CMOS
Die size	1.4mm×1.8mm
Current Consumption (Piezo)	$<$ 400nA
Current consumption (Programming)	$<$ 100μA

was first initialized back to 3V and a 1-Hz pulse signal was applied at V_{dd} until the sensor output voltage decreased by 1-V. The results show that the gain α reduces as the current I_{ref} reduces with



(a)

(b)

Figure 4.2: Measured injector gain for different V_{ref}: (a) 4.4V; (b) 5.0V.

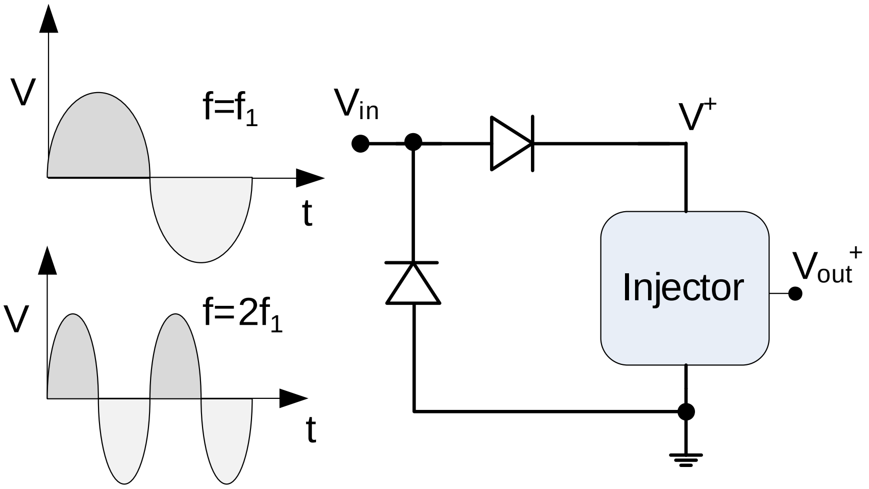


Figure 4.3: Experiment to measure charge for different frequency and amplitude inputs.

a minimum gain was calculated to be 25.6μV/cycle. Similarly, Fig. 4.2(b) shows results when V_{ref} was fixed at 5.0V and I_{ref} was varied by the same level as the previous experiment. The result shows that compared to Fig. 4.2(a), the gain of the injector increases for all currents in Fig. 4.2(b).

The ability to control the gain of the sensor is important because it allows the user to trade-off sensitivity of of strain-measurements with the operational life of the sensor before it has to be erased and re-calibrated.

For example, for long-term in-vivo applications the sensor might have to operational for over 10^5 mechanical activity cycles. In such cases, smaller values of V_{ref} and I_{ref} are preferred. Considering the operation range shown in Fig. 2.3, the sensor IC can process over 1.4×10^5 loading cycles for the minimum injection rate. On the other hand, the strain-sensor could also be used for short-term measurements, where larger values of V_{ref} and I_{ref} (larger gain) might be desirable.

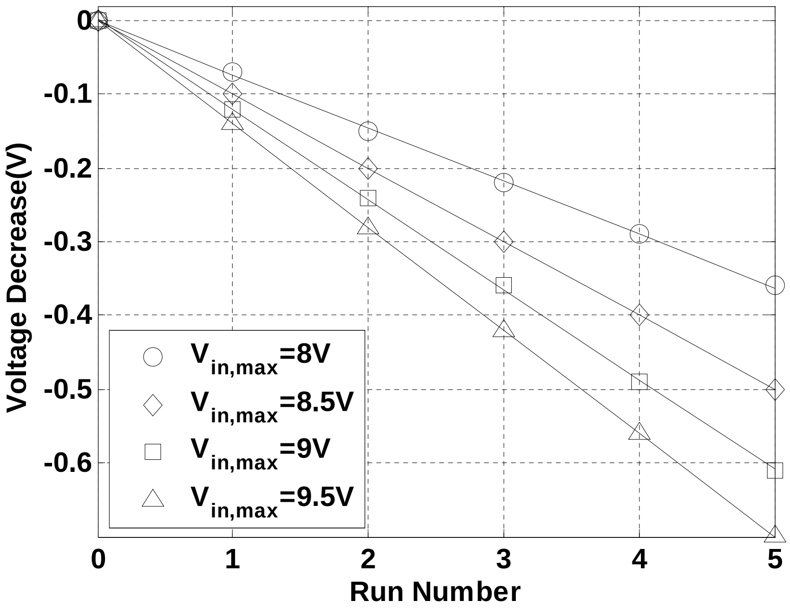


Figure 4.4: Measured response when the amplitude of the input is varied.

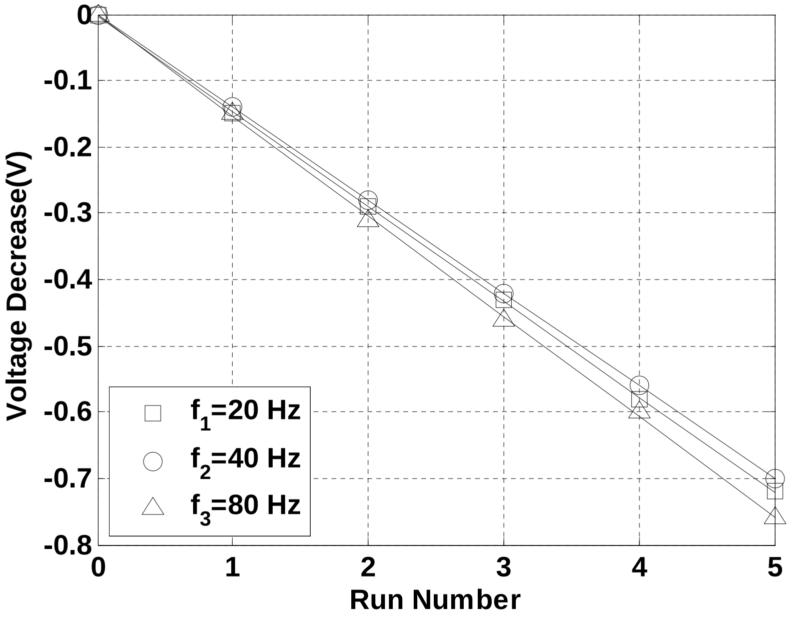


Figure 4.5: Measured response when the frequency of the input is varied.

4.3 Calculating energy difference between inputs

The next set of experiments to validate the ability of the linear injector to measure the charge generated during the positive cycles. We emulate the output voltage of the piezoelectric transducer as an AC source V_{in} shown in Fig. 4.3 which drives a half-wave rectifier and a single linear injector. The amplitude of the AC source was set to 8V, 8.5V, 9V and 9.5V and the frequency of the source was set to 40 Hz. For each amplitude, the sensor output was recorded at five intervals which are plotted in Fig. 4.4. The measured result shows that for each amplitude level, the sensor voltage decreases linearly with each measurement interval. It can also be noted that the gain of the injector increases with the increase in the amplitude, as expected from the results in section 2.2.

Next, the frequency of the source is varied while keeping the amplitude fixed. The energy content of the signal was kept constant by doubling the frequency. Thus it is expected that the output of the injectors will be equal. The measured result is shown in Fig. 4.5 showing a small difference between the response of the injectors. This is attributed to the mismatch between each of the injectors which can be calibrated post-measurement.

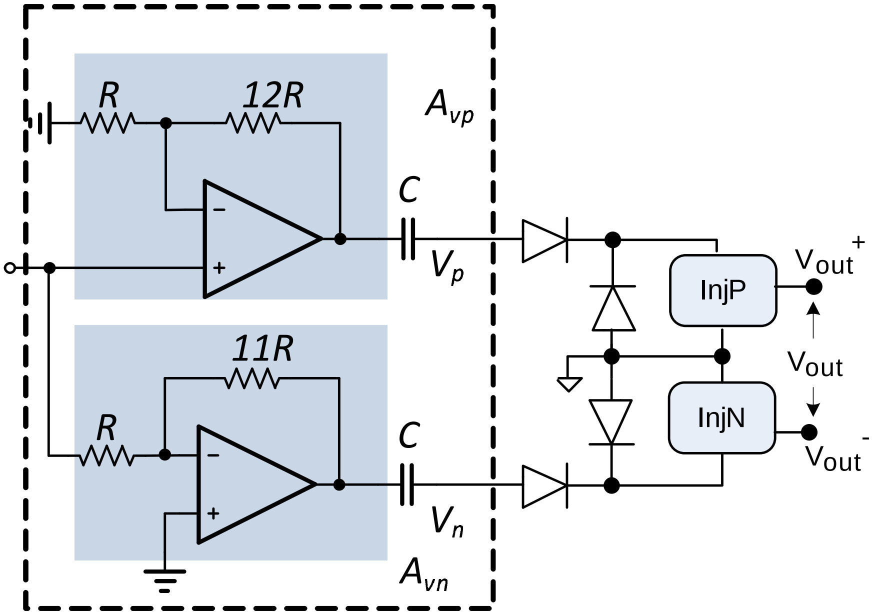


Figure 4.6: The test setup with piezo emulator.

4.4 Differential Measurements using a Piezoelectric Emulator

In the next set of experiments, the response of the differential injector circuit is validated using a circuit that emulates a low-frequency piezoelectric transducer model as shown in Fig. 1.14. The emulator circuit has been designed using amplifiers configured in an inverting and a non-inverting mode, as shown in Fig. 4.6. The output of the amplifier is capacitively coupled to the V_{piezo}^{+} and V_{piezo}^{-} inputs of the sensor IC (see Fig. 3.1). The circuit configuration thus generates a floating, differential and capacitively coupled output, similar to a piezoelectric transducer operating at sub-resonance frequencies. The advantage of using an emulator as opposed to a real piezoelectric transducer is that the electrical parameters of the circuit (emulating piezoelectric parameters) can be varied at ease by changing the resistance and capacitance values.

The emulator was configured to generate AC signals with programmable positive and negative cycle amplitudes. A control experiment consisted of applying an AC signal with equal positive and negative cycle amplitudes. The sensor response is shown in Fig. 4.7 where the output of each injector is measured after a certain number of voltage cycles (denoted by a run) have been applied. The results show a linear monotonic response for both the injectors, however, the difference between the output voltages is attributed to the mismatch between the two injectors. This mismatch is measured and used for calibration.

In the next experiment, the emulator was configured to generate a large amplitude during the positive cycle as opposed to the negative cycle. The measured response is shown in Fig. 4.8 which clearly shows a larger difference in the injector output voltages when compared to the control experiment. This experimental result verifies that the sensor can self-power, record and measure the difference in energy between the positive and negative voltage cycles.

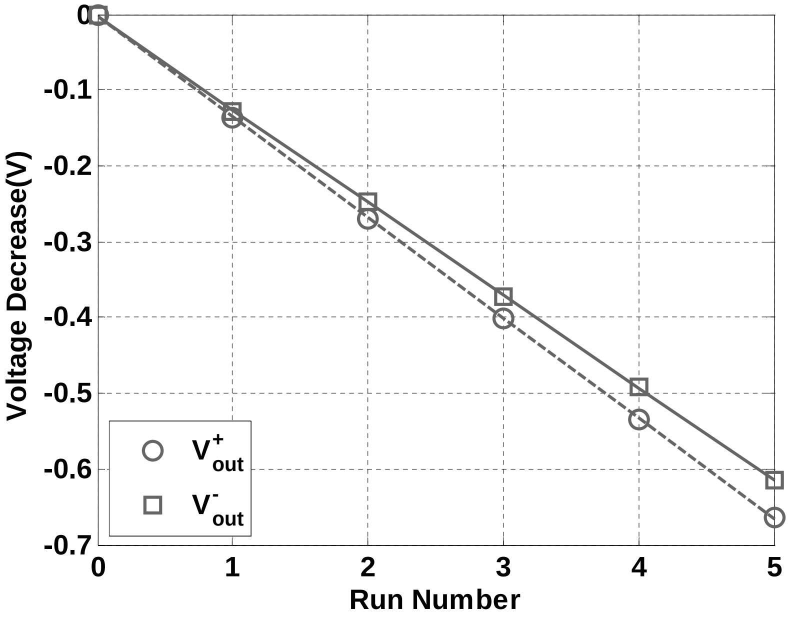


Figure 4.7: Sensor output with symmetrical positive and negative cycles

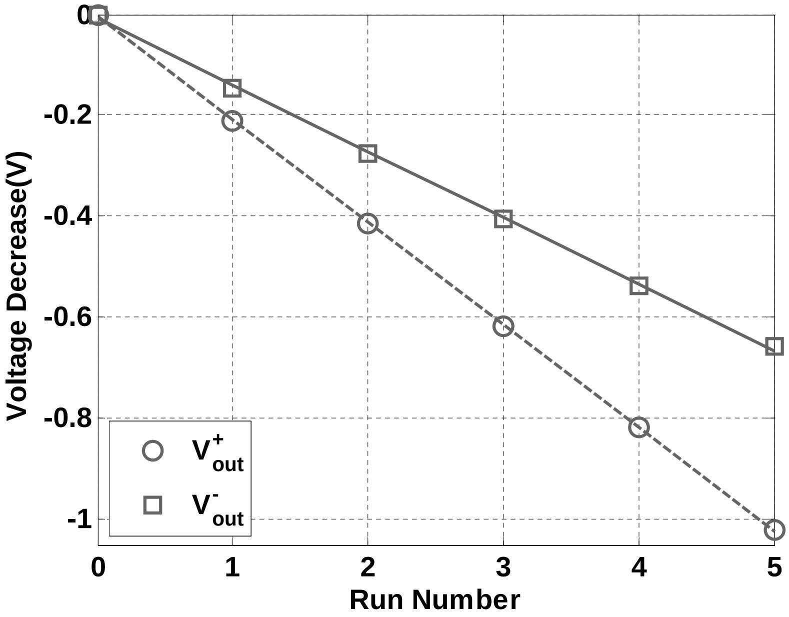


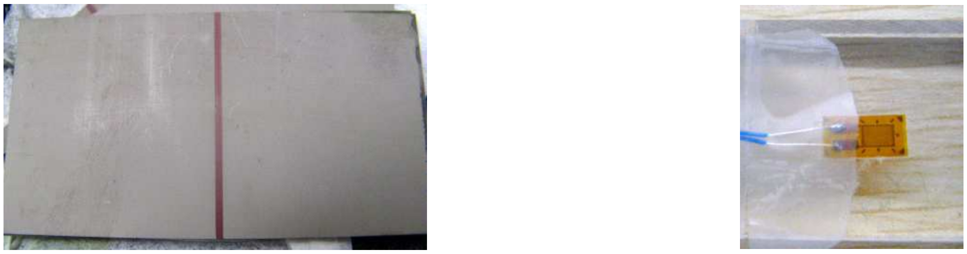
Figure 4.8: Sensor output with asymmetrical positive and negative cycles

4.5 Measurements using a Piezoelectric Transducer

In the last set of experiments, the sensor IC was interfaced with a piezoelectric transducer and was attached to a mechanical phantom that was designed to act as a biomechanical structure. The piezoelectric transducer chosen for this experiment was a commercial PZT ceramic available from Piezo Systems Inc. as shown in Fig. 4.9(a). Note that due to lead (Pb) content, PZT transducers are generally considered bioincompatible. However, the transducer is used in this work to demonstrate the proposed proof-of-concept, even though the design could be easily translated to a polymer based piezoelectric transducer. Also, appropriate packaging and shielding ensures that PZT transducers can be used in biomechanical studies as has been reported before [21]. Table 4.2 summarizes the mechanical and electrical specification of the PZT transducer using which the parameters of the low-frequency model in Fig. 1.14(b) can be readily derived.

As a biomechanical phantom, we used a plexi-glass beam to which the PZT transducer was attached as shown in Fig. 4.10. A programmable servo-motor was used to apply stress on the beam and the induced strain was measured/calibrated using a metallic strain-gauge as shown in the figure. The dynamics of the servo-motor was controlled using a PWM (Pulse Width Modulated) signal generated by a field-programmable gate array (FPGA). By adjusting the number of PWM cycles, the forward and backward movement of the motor shaft can be accurately controlled. The output of the PZT transducer is directly connected to the sensor IC with no additional power-sources.

The mechanical calibration of the set up was performed using a 350 Ohm(R) general purpose resistance strain-gauge from ‘Micro-Measurements’ (Fig. 4.9(b)). The strain-gauge was connected to a wheatstone bridge as shown in Fig. 4.10 and the circuit component values are shown in Ta-


(a)

(b)

Figure 4.9: (a)Piezo Material Used, (b)Static Strain-gauge mounted on a surface.

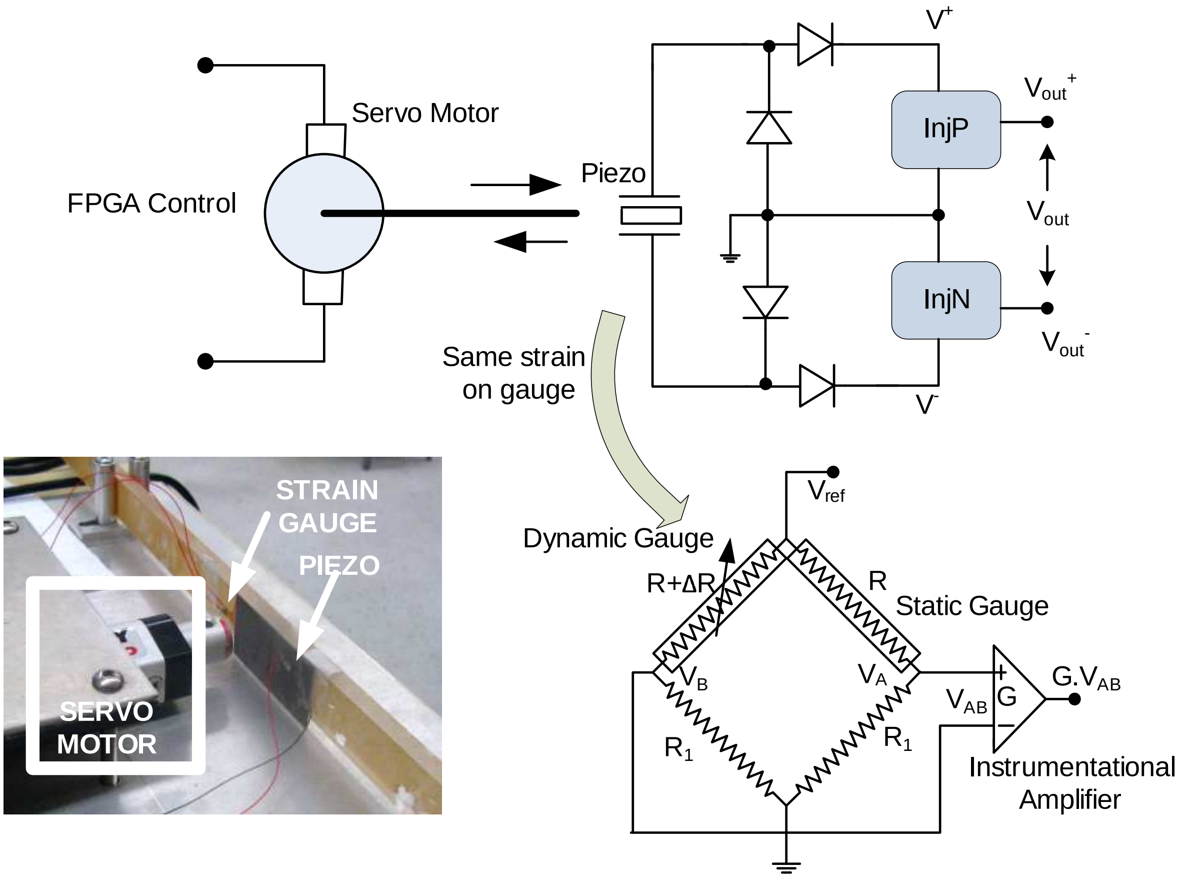


Figure 4.10: The schematic and photograph of the experiment setup with piezo material and strain-gauge attached to a plexi-glass beam which is put under mechanical strain using a servomotor controlled shaft. Also shown is the balanced bridge structure to measure the strain using the strain-gauge.

ble 4.3. Fig. 4.10 also shows a calibration strain-gauge used in the Wheatstone bridge, which was used to compensate for temperature variations in strain measurements. A commercial, fully differential instrumentation amplifier was used to amplify the difference in voltages across the Wheatstone bridge.

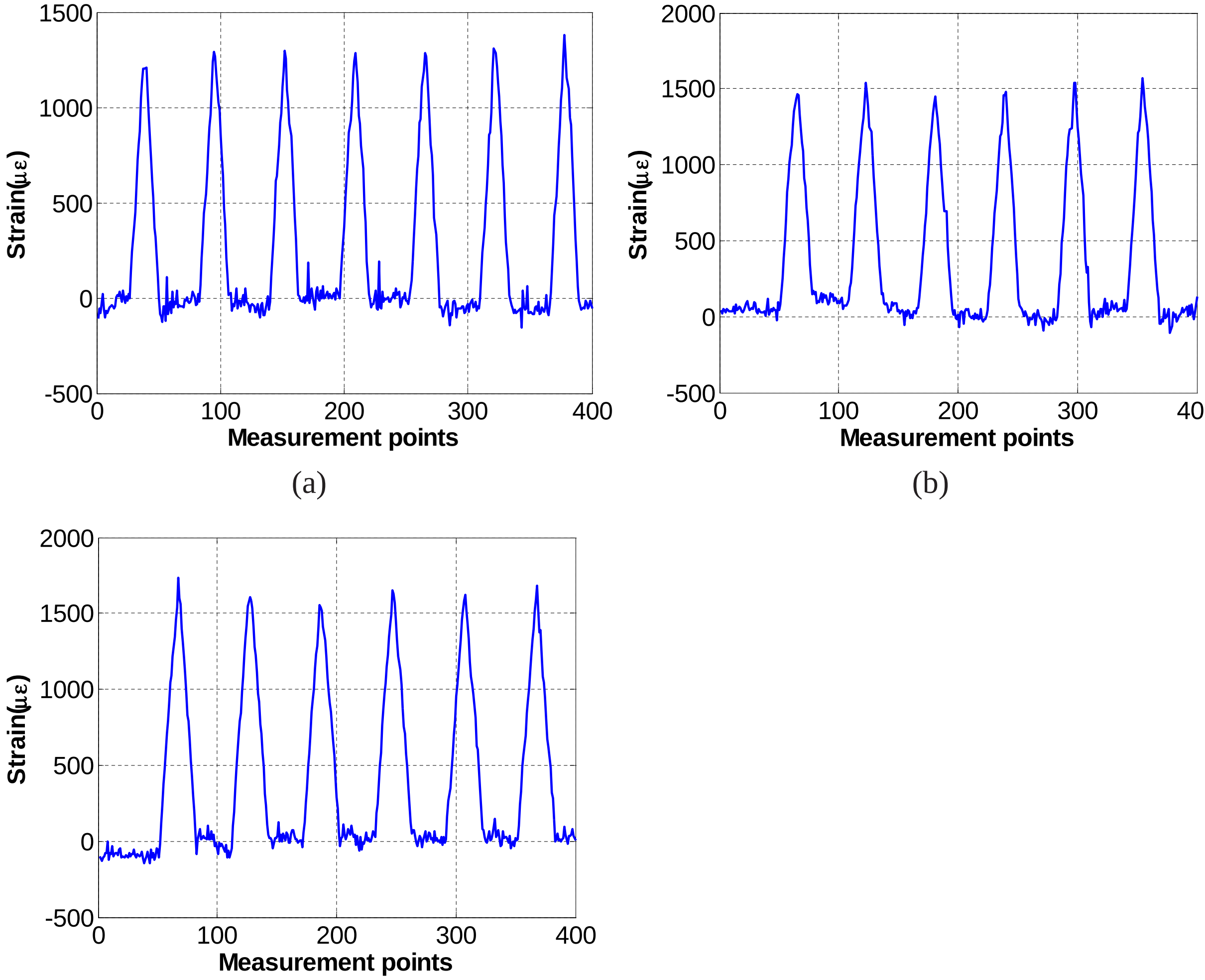
Fig. 4.11 shows the plots of time-varying strain (of different amplitudes) that is induced in the plexi-glass beam and Fig. 4.13 shows the output across the PZT transducer for one specific case. It can be clearly seen in Fig. 4.13 that the output voltage of the transducer shows positive and negative voltage cycles that correspond to the direction of the deformation of the beam. For all the experiments, the nominal strain-levels were chosen to be consistent with the strain levels summarized in Table 1.1. The output of the sensor was measured after every 10 loading cycles. The measured results are shown in Fig. 4.12 and Fig. 4.14 corresponding to both the injectors. The result shows that the response of the both the injectors are similar, indicating that the energy in the positive and negative strain cycles are similar. This is expected since the plexi-glass beam is allowed to return to its resting state (zero static strain state). However, unlike conventional strain-gauges, the self-powered gauge provides a historical indicator of the L_1 norm that is proportional to the energy dissipated through the PZT and hence the plexi-glass beam. The energy is proportional to the average of the output generated by both the injectors. This historical information could be more useful for determining progression of damage in a structure than just a passive static-strain measurement.

Table 4.2: Piezo Material specifications [22].

Length(l)	2.5 inch
Width(w)	1.25 inch
Thickness(h)	.02 inch
Material	Lead Zirconate Titanate(PZT)
Electrodes	Nickel
Capacitance	73nF
Strain/Field(d_{31})	-190×10^{-12} m/V
Field/stress(g_{31})	-11.6×10^{-3} V-m/N
coupling(k_{31})	0.35
Elastic Modulus(Y^E)	$5.2 \times 10^{10} N/m^2$

Table 4.3: Strain-gauge specifications.

R	350Ω
$\mathbf{R}_1$	$1k\Omega$
Gauge Factor(GF)	2.11
$\mathbf{V}_{ref}$	5V



(a)

(b)

(c)

Figure 4.11: Strain levels from strain gauge measurements: (a) Maximum strain value of 1300$\mu\varepsilon$; (b) 1500$\mu\varepsilon$; (c) 1600$\mu\varepsilon$.

To demonstrate that the proposed sensor IC can indeed measure quasi-static strain using a real piezoelectric transducer, the following experiment was designed. The servo-motor was programmed to generate strain-levels according to the waveform as shown in Fig. 4.15. The waveform shows that the plexi-glass beam is not allowed to return to its resting-state, implying that the measured strain at the start of the experimental run is not equal to the measured strain at the end of the experimental run. After each run, the output of the injectors are measured before the beam is subjected to a similar strain-cycle. Fig. 4.16 shows difference between the output of the injectors

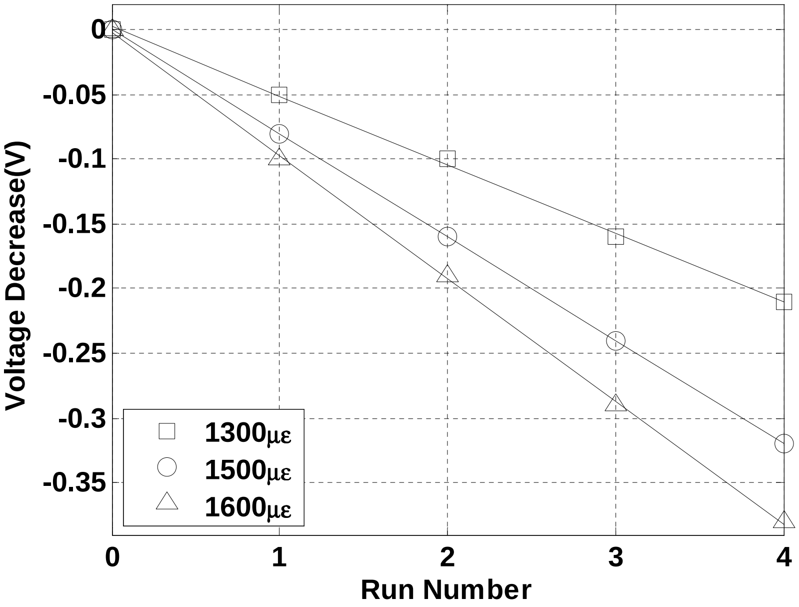


Figure 4.12: Response of the positive cycle injector for the 3 strain levels.

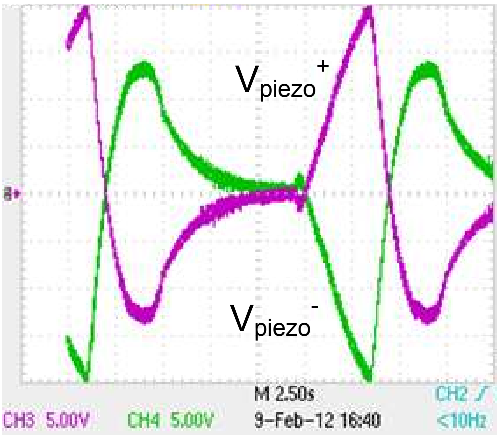


Figure 4.13: Oscilloscope capture of the piezo outputs before the rectifier for one of the strain values.

decreases with each run, which is proportional to the level of the quasi-static strain. Also the first injector measures a larger decrease in the output voltage as compared to the second injector. When the same experiment is repeated, however, after swapping the input terminals of the sensor, the second injector measures a larger decrease in the output voltage (after calibrating for mismatch effects) as shown in Fig. 4.17 , which is consistent with the expected results.

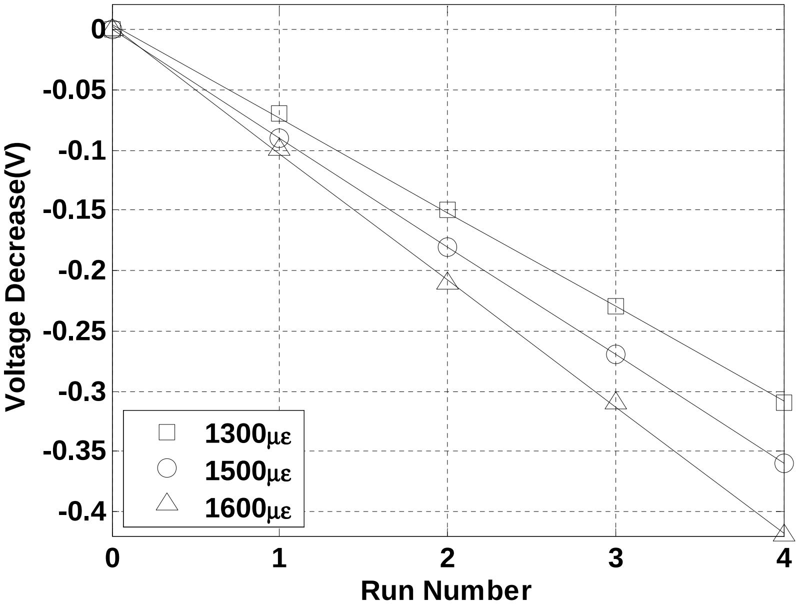


Figure 4.14: Response of the negative cycle injector for the 3 strain levels.

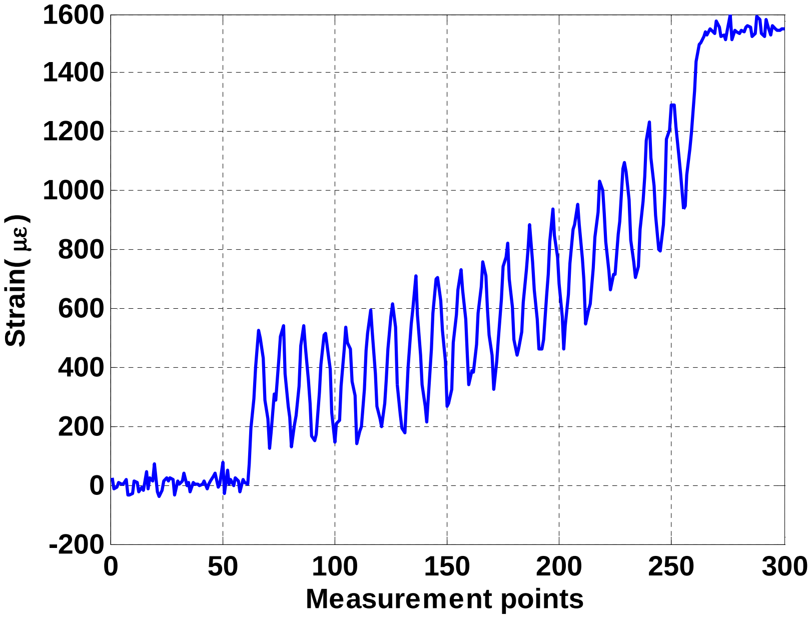


Figure 4.15: Strain Gauge measurement for one cycle of static strain.

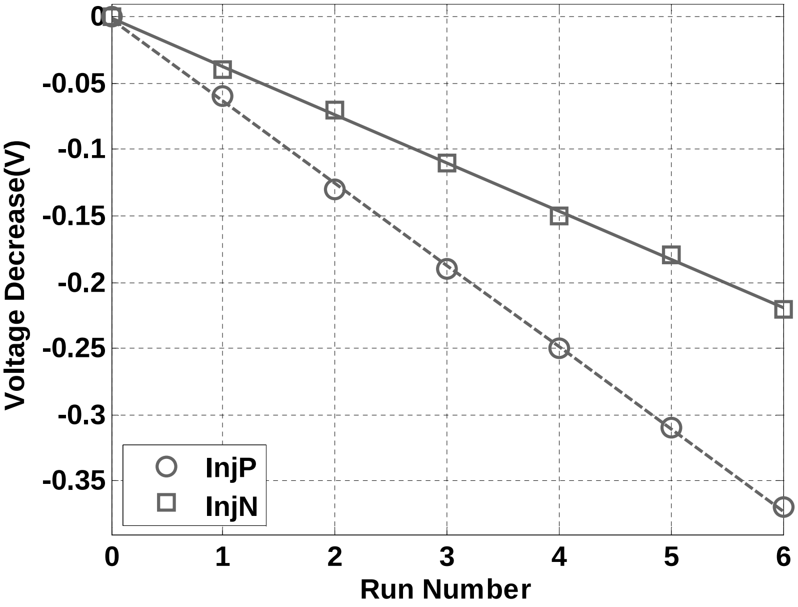


Figure 4.16: Sensor Output when InjP has more strain than InjN.

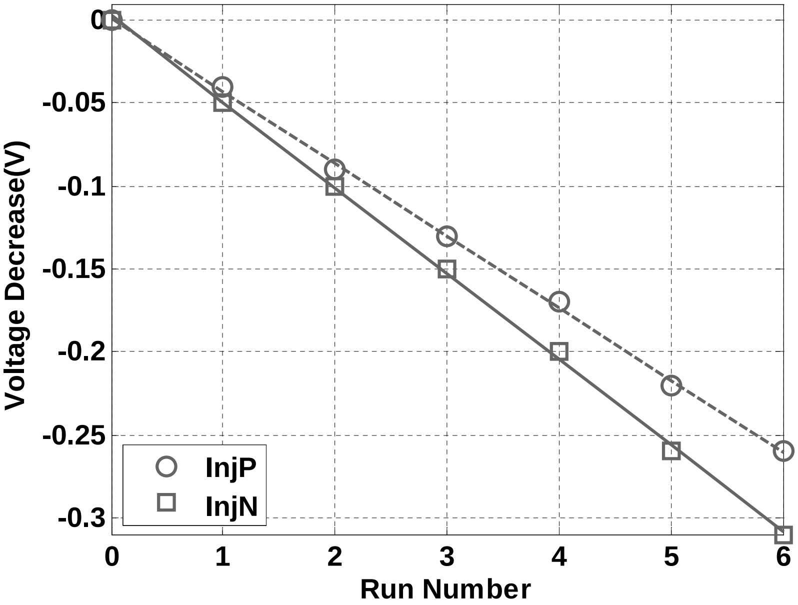
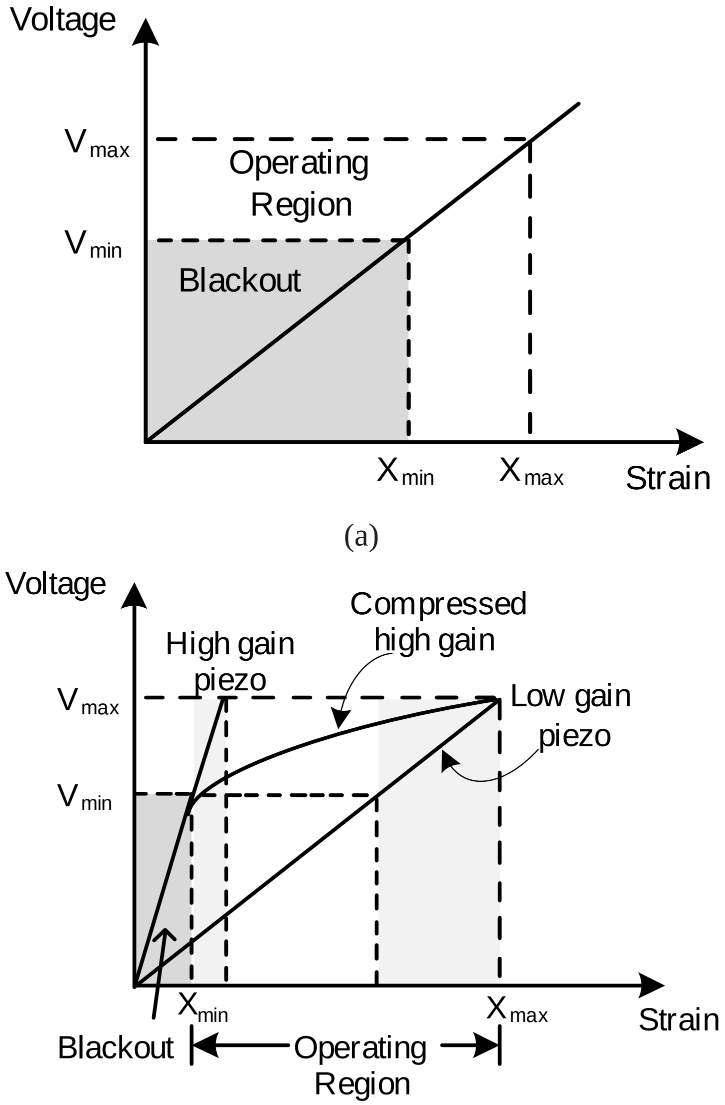


Figure 4.17: Sensor Output when InjN has more strain than InjP.

Chapter 5

Compressive Piezo-Floating-Gate Event-Counters

A significant disadvantage of the asynchronous self-powered sensor presented is the *threshold effect* as illustrated in Fig. 5.1(a). The electronics being powered by the transducer typically requires a minimum threshold voltage(5V at a load of 10MΩ) to activate the non-volatile storage functions. On the other hand, the electronics also determines a maximum limit on the magnitude of the sensor signal and this limit is determined by the breakdown voltages of active components on-chip. For example, the maximum voltage of the strain-powered sensor reported in [10] is 9V which is determined by the reverse break-down voltage of the protection diodes. The lower and upper threshold in the sensor signal severely limits the operating and sensing range of the sensor as is illustrated in Fig. 5.1(a). The lower voltage threshold determines the minimum strain that needs to be applied before the sensor becomes operational, and the higher voltage threshold determines the maximum strain-level can be sensed. The effect remains unchanged if a high-gain piezoelectric transducer (as shown in Fig. 5.1(b)) is used, which even though lowers the minimum strain-level required for



(a)

(b)

Figure 5.1: Sensor Operating Region : (a) Without any Compressive Gain; (b)With the Non-Linear Compressive Gain.

powering the sensor but significantly reduces the maximum level of strain that can be sensed.

In this chapter we present the design of a compressive-self-powered strain-sensor that alleviates the threshold effect for self-powering and in the process significantly enhances the powering/sensing range. The approach is illustrated in Fig. 5.1(b), where a high-gain piezoelectric transducer is used to reach the powering-threshold at low-levels of strain, where as the gain is progressively reduced such that the sensing-threshold is reached at large-levels of strain. This compressive response can be achieved by using a non-linear resistive load and will be discussed in section 5.1. The use of a compressive response, however, requires that the circuits computing the statistics of the strain signal can be tuned or calibrated to process the compressed input signal. Also, the compressive input requires precision calibration of any mismatch due to the analog components. In sections 5.2 and 5.4, floating-gate circuits that can be used to precisely program our previously reported strain-level crossing circuits [11] and our previous reported linear floating-gate injectors [31] have been presented. Also to reduce power consumption a daisy chain level crossing scheme has been implemented in sec 5.3.

5.1 Non-linear Compressive Circuit for Protection and Range Mapping

So, the requirement on the piezo would be that it should generate a voltage higher than the threshold for the lowest strain and a voltage lower than the breakdown voltage for the maximum strain we need to monitor. Thus, we need some kind of compressive gain before the piezo outputs are fed to the sensor. The structure described below and shown in Fig. 5.2 performs this task.

R_1, R_2 and R_3 with multiple diode chains in parallel make a voltage divider. The main working principle is that for low voltage levels, the diode chain resistance is quite high and most of the voltage drop happens across R_3 and the sensor. But when voltage increases, the diode chain resistance decreases non-linearly and the major voltage drops happen at R_1 and R_2. We have used R_1=R_3=1M and R_3=99M. D_{R1}, D_{R2}, D_{R3} and D_{R4} form a full-wave rectifier. The diode chains consist of three conduction paths formed by diodes D_1-D_{10}, D_{11}-D_{18} and D_{19}-D_{20}. Note that the

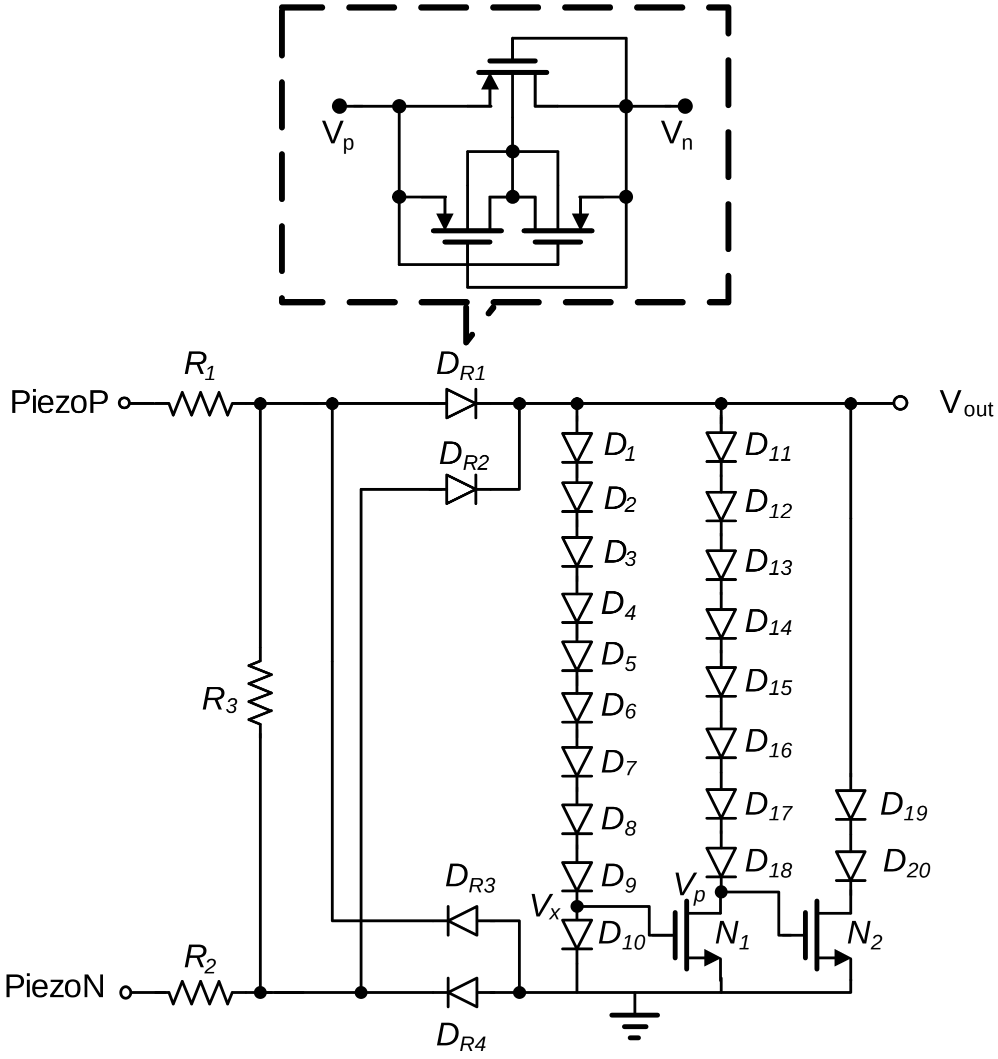


Figure 5.2: Non-linear compressive circuit.

voltage $V_x \approx V_{out}/10$. When $V_x < V_{th}$, with V_{th} being the threshold voltage of the nMOS transistor N_1, the diode chain D_{11}-D_{18} is OFF. V_p is floating and it follows V_x roughly. So the diode chains D_{11}-D_{18} and D_{19}-D_{20} turn ON almost together. When D_{11}-D_{18} starts conducting, it will load more current and try to turn ON all the 8 diodes. When they are fully turned ON, $(V_{out} - V_p)$ would remain almost constant(increase logarithmically to be precise). When V_{out} increases, V_p would increase too and this would make transistor N_2 conduct more current which would bring down V_{out}. Thus V_{out} would be clipped off or increase with a very small slope.

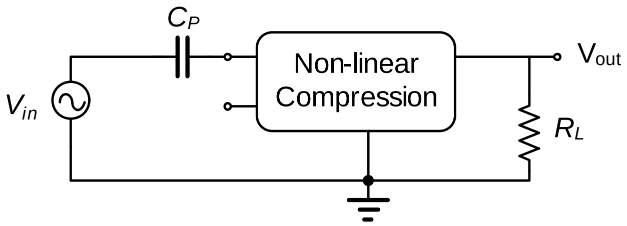


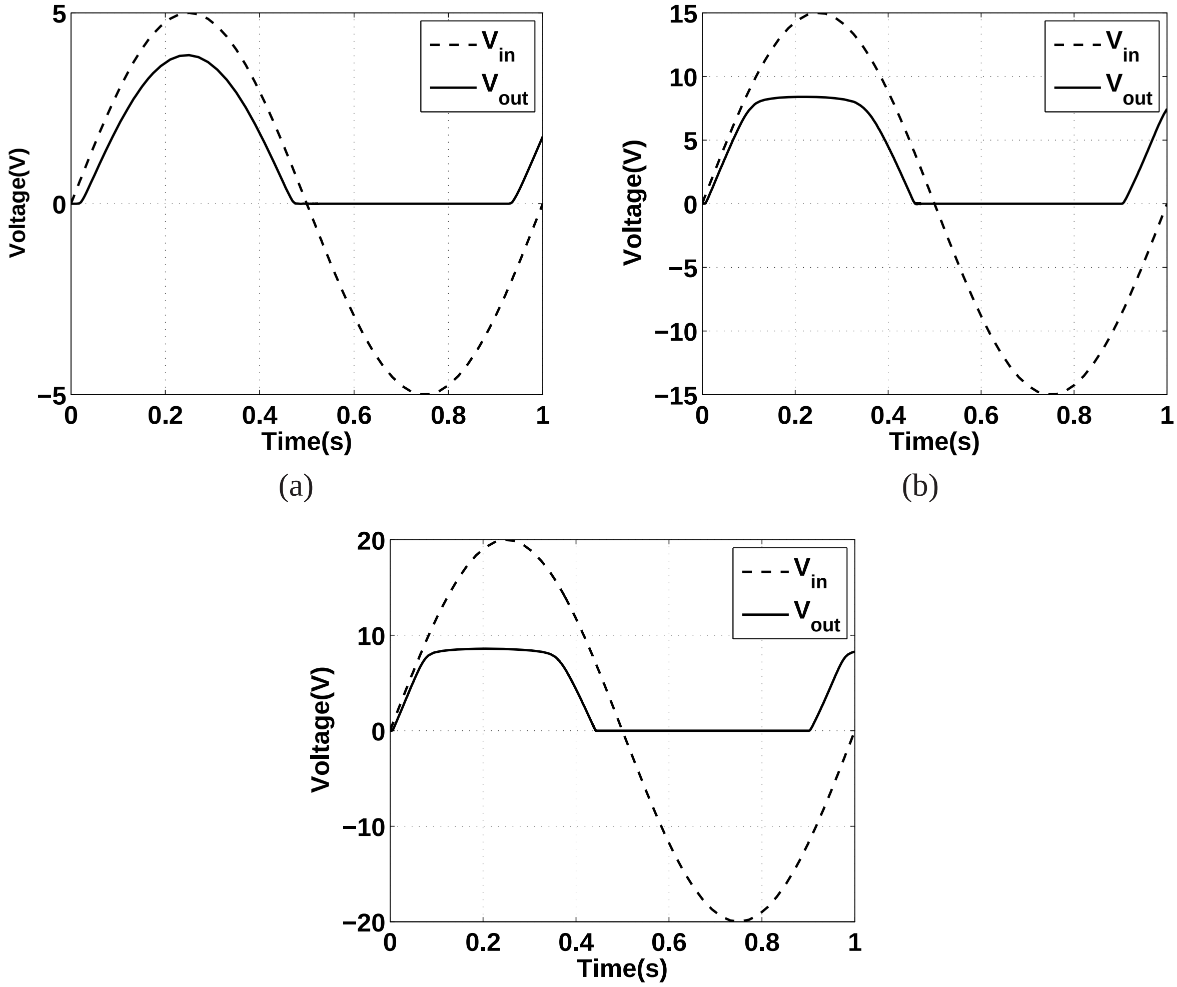
Figure 5.3: Simulation setup for Non-linear compressive circuit.

This circuit was simulated using the test setup shown in Fig. 5.3. Here it is used as a half-wave rectifier and the low frequency piezo model is made of voltage source V_{in} and capacitance C_P. V_{in} is a sine voltage source of frequency 1 Hz whose amplitude is varied in 4 runs. C_P has been taken as 200nF. R_L is the load resistance equal to 25MΩ. V_{out} was observed with V_{in}. The results have been shown in Fig. 5.6. It shows that for V_{in} amplitude of 5V, there have been only 1V compression, but for 20V the compression is almost 11V. It has been seen for even a V_{in} of 100V, V_{out} remains below 10V.

Fig. 5.5 shows the measurement result from silicon. This result is without the capacitor and the load is a multimeter. The input V_{in} is through a keithley from 0 to 21V in steps of 0.5V. For each step V_{out} was observed and has been plotted. It shows that there are two distinct slopes in the response. From about 9V the V_{out} rises with a different slope which is due to turning ON of the first diode chain.

As it is difficult to generate very high voltages for testing, the circuit has been simulated in cadence. In this simulation, V_{in} has been applied as a linear function of strain(S). For a piezoelectric cantilever with dimensions $L \times b \times h$, the open-load voltage($v_{in}(t)$) generated across the transducer as a function of a perpendicular mechanical force $F(t)$ is given by [12]:

$$V_{in}(t) = \frac{F(t)g_{31}}{b} = S(t)Y^E hg_{31} = \frac{S(t)Y^E d_{31}h}{\varepsilon} \tag{5.1}$$



(a)

(b)

(c)

Figure 5.4: Non-linear compression simulation result : (a) For a max amplitude of 5V; (b)For a max amplitude of 15V; (c)For a max amplitude of 20V.

where g_{31} and d_{31} are piezoelectric constants, $S(t)$ is the time-varying mechanical strain, Y^E is the short circuit elastic modulus and ε is the electrical permittivity. In this example our minimum strain level of interest was 100$\mu\varepsilon$ and the maximum was $10^4\mu\varepsilon$. Thus if the piezo properties are such that, $Y^E hg_{31}$=.058 in eq. 5.1, then V_{in}=5.8V and V_{out}=5V after one diode drop(across D_{R1})for 100$\mu\varepsilon$. A dc simulation was performed varying S and the result has been shown in Fig. 5.6.

It can be seen that V_{out}=13V for $10^4\mu\varepsilon$ strain. If the figure is zoomed in till $10^3\mu\varepsilon$ strain, then the response is very similar to the measurement result shown in Fig. 5.5. This figure also shows

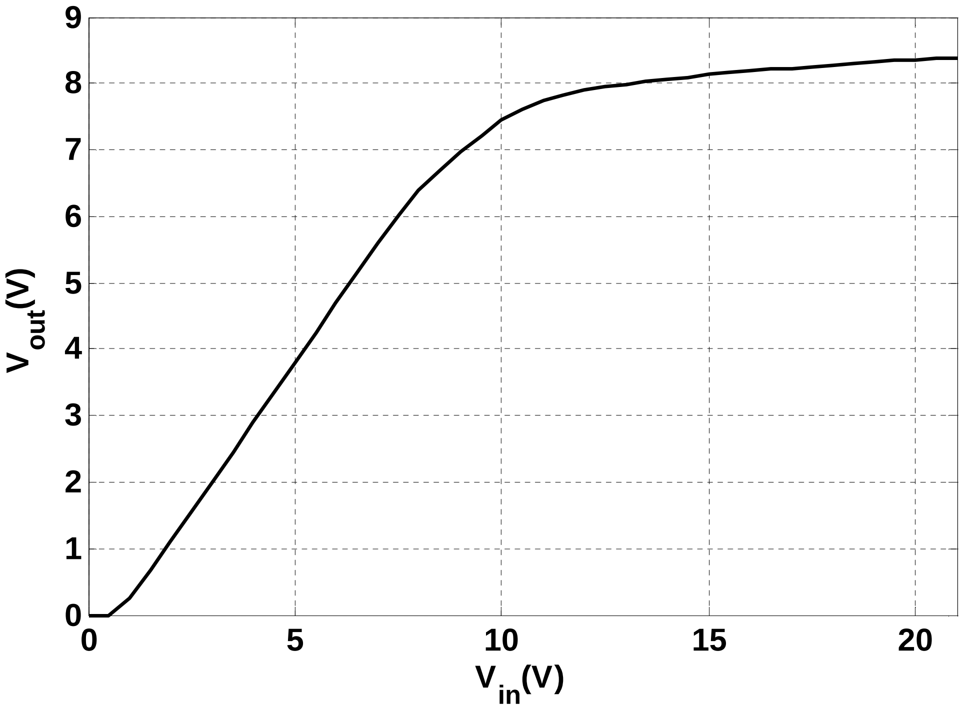


Figure 5.5: Measured Non-linear response when V_{in} has been swept from 0 to 21V.

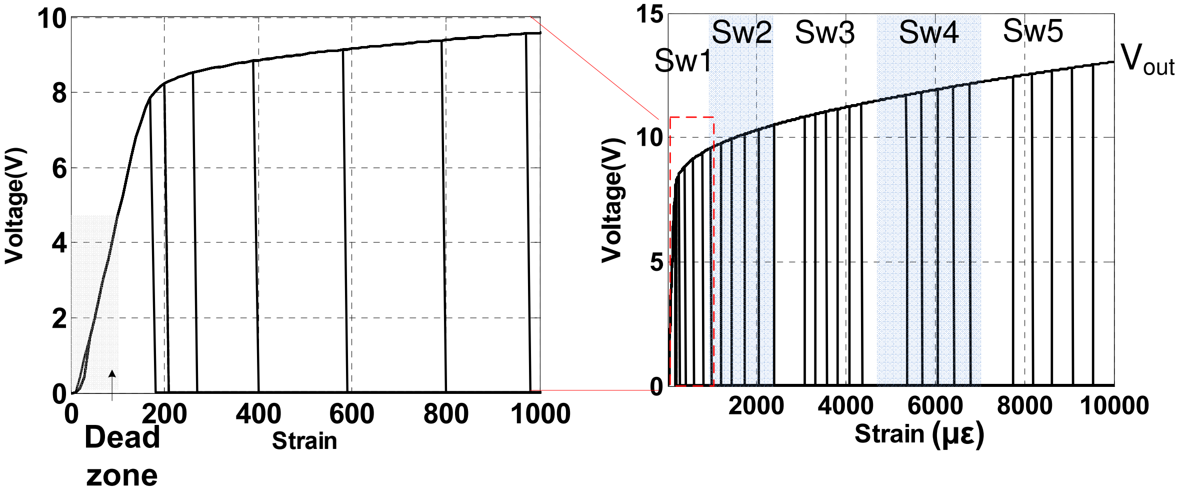


Figure 5.6: Simulation result showing the compressive circuit allowing the sensor to operate from strain values of 100 to $10^4\mu\varepsilon$. Also 5 level crossing detectors covering the whole region by programming the charge in the floating-gate.

that the whole strain-voltage range can be covered by level-crossing detectors to use the sensor as event counter. This is explained in the next section.

5.2 Programmable Level-Crossing Detector

Level-crossing detectors are needed to determine the supply range. There are multiple detectors that gets turned ON for different supply levels. Each detector triggers a separate injector. In chapter 3 a ordinary level crossing detector was described. But due to the compressive response we need more level-crossing detectors whose thresholds are very close to each other. This cannot be achieved by just changing the number of diodes in the chain. Thus floating-gate MOS was used to tune the thresholds. This can also take care of the mismatch in thresholds between different detectors.

Fig. 5.7 shows a programmable level-crossing detector. M_{c1} and M_{c3} form a cascoded current source that feeds current to a diode chain of N diode-connected PMOS. The source of the bottom-most diode(M_d), V_l is used as the control signal. When V_{dd} rises, V_l rises too and ultimately settles to a voltage required for fully turning on M_d. The slope of the voltage rise depends on the resistance of the diode chain which depends on the number of diodes. When V_l is high enough to make the current through M_a higher than the current source M_{c2} and M_{c4}, the switch V_{sw} goes low. Thus, V_{sw} goes low at a particular supply voltage. Similar kinds of diode chains are kept changing the number of diodes to achieve coarse programming of level-crossing thresholds.

For fine tuning, M_d has been made a floating-gate MOS instead of a regular MOS. It can be programmed using a linear injector M_{fg}, amplifier A and switches S_p and S_{pb}. In normal mode, S_p will be closed and S_{pb} open making M_d a MOS whose gate it connected to ground via a capacitor which has some charge in it. Depending on the floating gate charge, the threshold of M_d will be

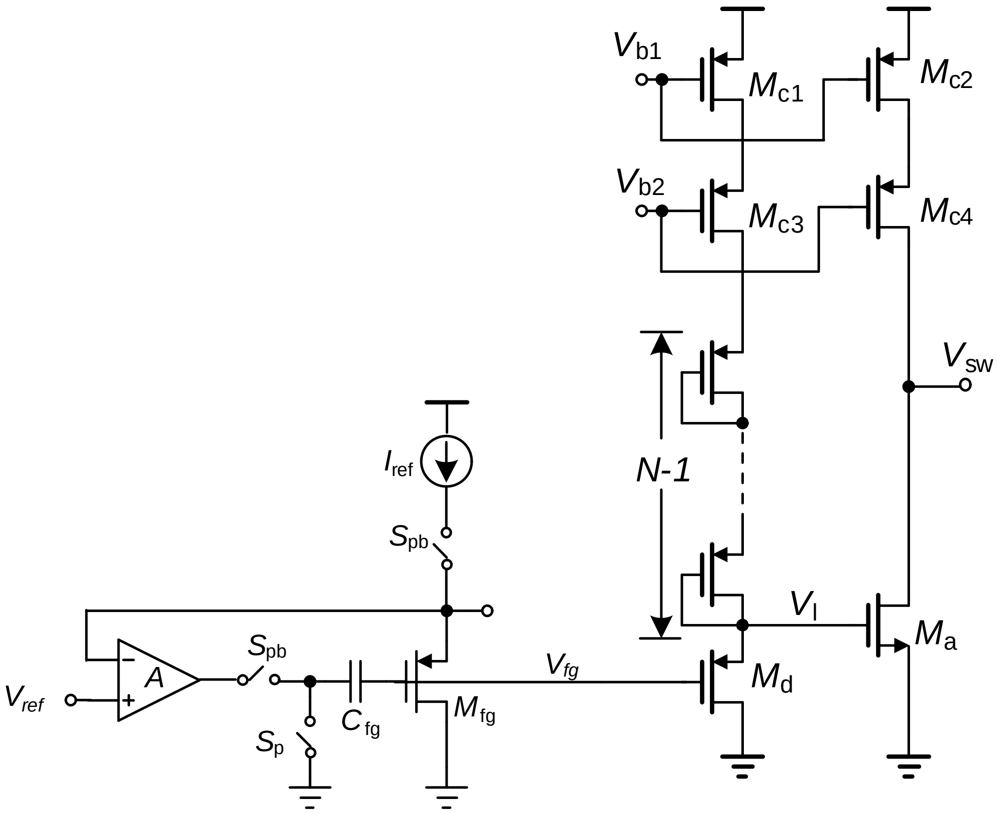


Figure 5.7: Level Crossing Detector with tuning.

changed. Thus M_a can also be turned ON at a different V_{dd}. To program the floating gate V_{fg}, S_p has to be opened and S_{pb} closed. There is also a tunneling capacitor(not shown in Fig) to program. The amplifier A can be easily shared between multiple detectors. It is to be noted that during normal operation only the amplifier would consume a minimal extra power.

The simulation results can be seen in Fig. 5.6. This simulation was done with the non-linear gain and level-crossing detector circuits. There were 5 level-crossing detectors with different number of diodes. Multiple runs were done with different amount of charge in the floating-gates which shows that the entire voltage range can be covered.

The programmability was tested in silicon with one detector. The result is shown in Fig. 5.8. V_{dd} is ramped from 0 to 9V, V_l and V_{sw} have been observed for four runs with different amount of charge in V_{fg}. It can be seen that V_{sw} turns ON at different points of V_{dd}. The programmability

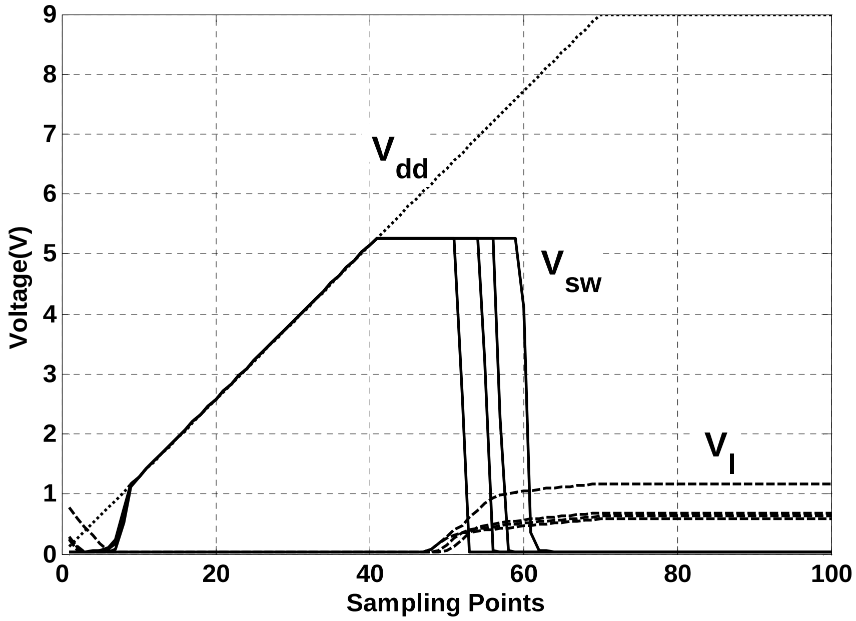


Figure 5.8: Measurement result for the level-crossing detector.

is more efficient when V_l is less than the normal MOS threshold voltage, which means V_{fg} has a negative voltage.

5.3 Daisy Chain based Level Crossing Detector

When multiple injector channels are used for detecting voltage levels, the power consumption adds up. Daisy chain based injector ensures that only one channel is ON at a time. For example, in Fig. 5.9 the three levels are V_1, V_2, V_3. In normal level crossing detector, the first channel would be ON for time $(t_6 - t_1)$ when voltage$> V_1$, 2nd channel for $(t_5 - t_2)$ when voltage$> V_2$ and third channel for $(t_4 - t_3)$ when voltage$> V_3$. So there is a overlap between the channels. All the three channels are ON between $(t_4 - t_3)$. In daisy chain system, the first channel would be ON when the voltage is between V_1 and V_2, 2nd channel between V_2 and V_3 and the third channel when voltage is above V_3. So there will be no repetition of information.

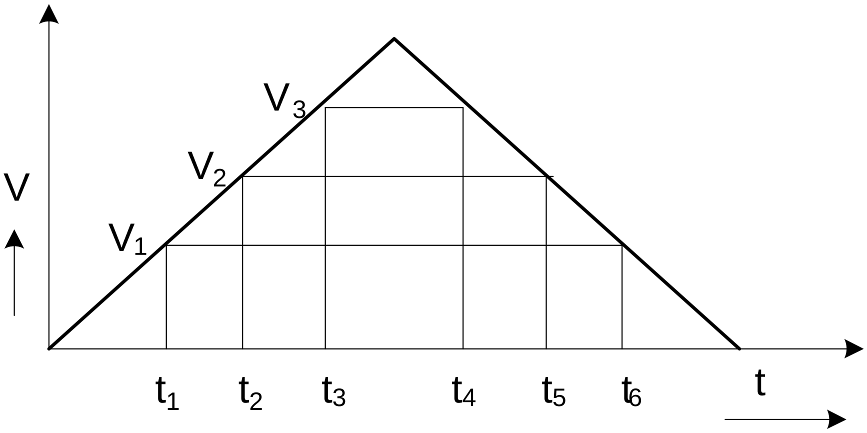


Figure 5.9: Level crossing.

For implementing this function the normal level crossing detector described earlier was followed by an extra logic block. The logic is the channels should turn ON one by one and when the next channel is ON the previous channel should turn OFF. This is shown in Fig. 5.10 using two consecutive level crossing switches $Sw < i >$ and $Sw < i+1 >$. It can be easily seen that

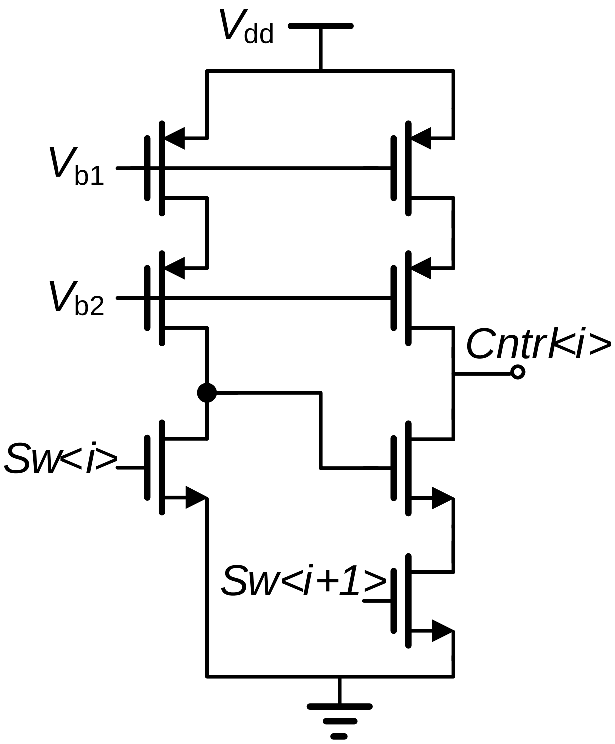
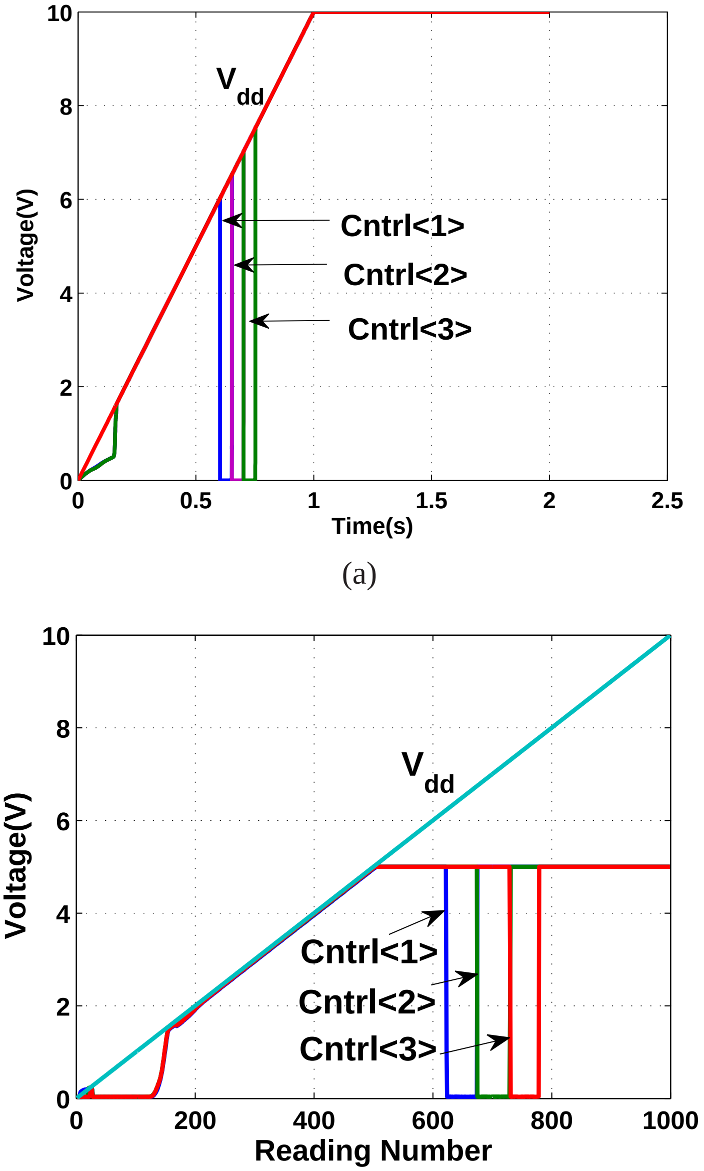


Figure 5.10: Daisy Chain Level crossing.



(a)

(b)

Figure 5.11: Results of Daisy Chain Level Crossing for 3 Cntrls : (a) Simulation; (b)From the chip.

$Cntrl < i >$=0(ON) only when $Sw < i >= 0$(ON) but $Sw < i+1 >$=1(OFF). This logic was added after each $Sw < i >$ to ensure that no two channels are ON at the same time. The entire logic is current starved with few nanoampere so that the loading on piezo does not increase too much.

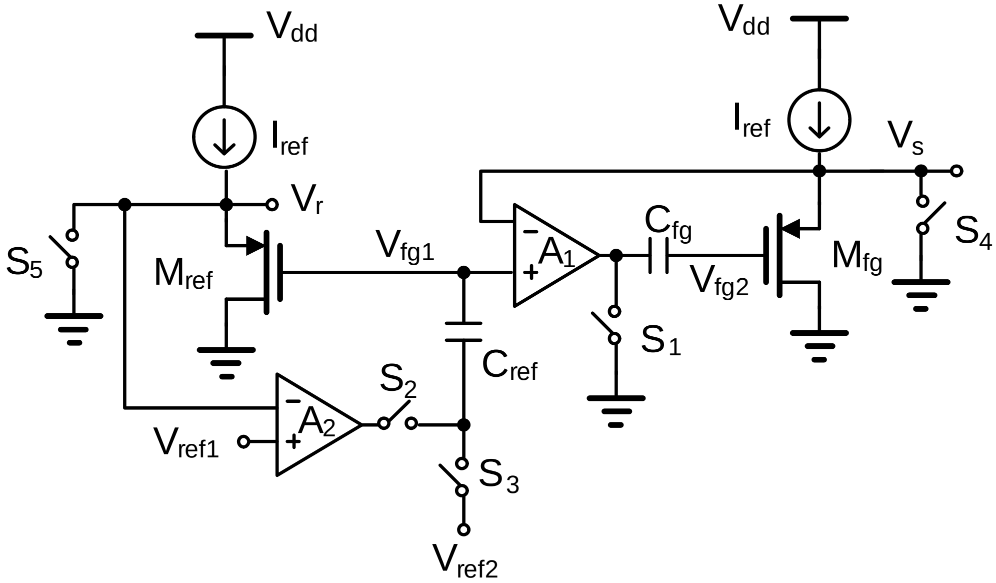


Figure 5.12: Offset cancelation Circuit.

5.4 FG based Opamp Offset cancellation in Linear injectors.

Multiple injector channels are used in the sensor that get turned ON for different supply levels. It is important that the injector slopes are similar. But there is always a slope difference between them due to mismatches in I_{ref} and opamp offset. This changes the injection rate and as a result the slope. A circuit block was designed to overcome this problem using floating gate MOS to tune V_{ref} and make the slopes equal. This concept was applied to two channels in a testchip.

In Fig. 5.12 one such channel has been shown. M_{fg} is the primary linear injector with amplifier A_1, control capacitor C_{fg} and current source I_{ref}. This injector injects when switches S_1 and S_4 are open. S_1 is there to switch between injection mode and read-out mode. S_4 is to select or deselect a particular channel from injection. M_{ref} is another linear injector with amplifier A_2, control capacitor C_{ref} and current source I_{ref} to program floating node V_{fg1}, which acts like a reference voltage V_{ref} to M_{fg}. Switches S_2 and S_3 are complimentary which connects one plate of C_{ref} to either amplifier A_2 output(injection mode for M_{ref}) or to another reference V_{ref2}, which can be either 0V(For M_{ref}) or 7V(For M_{fg}) according to requirement. V_{fg1} is given by the following relation,

$$V_{fg1} = V_r . \frac{C_{gs}}{C_T} + V_{ref2} . \frac{C_{ref}}{C_T} + \frac{Q}{C_T} \tag{5.2}$$

where, C_{gs} is the gate to source capacitance of M_{ref}, C_T is the total capacitance connected to node V_{fg1} and Q is the charge stored in the floating-gate. First V_{fg1} is programmed to change Q. But to inject in M_{fg}, V_{fg1} has to be around 5V. If M_{ref} is kept turned ON, it would also inject and change V_{fg1}. Thus it has to be made injection-safe. The way that has been done is to close switch S_5. Thus V_{sd} of M_{ref}=0 and there would be no injection. But when V_r is made zero, V_{fg1} also decreases. So V_{ref2} has to be high enough to compensate for that. Also, the amplifier A_1 has to be well cascoded so that the input stage MOS V_{sd} does not cross 4.3V and starts to inject.

First V_{fg1} is programmed. To read-out from M_{ref}, S_2 is open, S_3 is closed and $V_{ref2} = 0$. To start injection, S_2 is closed and S_3 is opened. Also, S_5=0 in the channel we want to program and '1' in the channel we don't. During injection cycle, $V_r = V_{ref1}$ and during read-out phase V_r would give the floating gate voltage. There is also a tunneling node(not shown in fig) to increase V_{fg1}. V_r is programmed to a voltage close to 4V in both the channels. We aim to have $V_{fg1} > 5V$ so that it can enable injection in M_{fg}. But we cannot program V_{fg1} to that value directly as M_{ref} would start to inject beyond 4.3V. So we need a second reference V_{ref2} in addition to the floating-gate voltage.

Now to use V_{fg1} as a constant reference we open S_2 and close S_5 in both the channels and make V_{ref2}=7V. First S_1 is closed and V_{fg2} is read through V_s. Then S_1 is opened and S_4 is also kept opened to inject in M_{fg}. It was seen from both simulation and measurements that when V_r was programmed to 3.9V, we would get V_{fg1} to be around a 5.2V reference. The requirement of V_{ref2} can be brought down by increasing C_{ref} as that would enable more of V_{ref2} to be transferred to V_{fg1}.

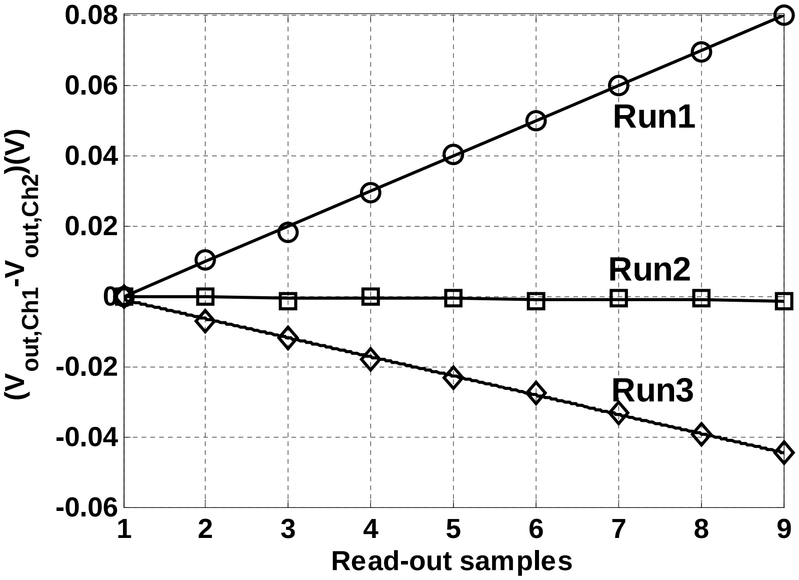


Figure 5.13: Measured result of Offset cancelation Circuit showing the difference in response between the two channels for 3 runs.

V_r of both channels were programmed to a value close to each other and then injection was performed in M_{fg}. It was seen that channel2 had a higher reference and thus higher injection rate as shown in Fig. 5.13 as 'Run1' . Then V_r was programmed again so that both the channel injection rate were exactly same. This is called 'Run2'. To show consistency in 'Run3' the reference of channel1 was made higher and it showed higher injection rate too. The three conditions are shown in table 5.1.

Thus this circuit could be used to remove mismatch between injectors. It can not only take care of the opamp offset, but also the mismatch in I_{ref} as by programming the floating gate reference

Table 5.1: Values of V_r and V_{fg1} of the 2 channels in the 3 runs.

Run Number	V_r-Ch1(V)	V_{fg1}-Ch1(V)	V_r-Ch2(V)	V_{fg1}-Ch2(V)
1	3.77	5.2	3.83	5.25
2	3.67	5.12	3.67	5.14
3	3.63	5.1	3.58	5.08

we can get any injection slope. As can be seen in table 5.1 for Run2, V_{fg1} are not same but the injection rate is exactly same. This is due to the compensation of I_{ref} mismatch.

Chapter 6

Programming, Calibration and Reliability of Injectors

The floating gate injectors need to be initialized to proper values so that they can utilize their full linear range. Also there would be slope differences between different injector channels due to mismatch and offset of analog components. To solve these two problems an automated initialization and calibration methodology was implemented. This chapter discusses the algorithms, programs and test setup in details.

6.1 Initialization

The injector voltages have to be programmed by repeated use of injection and tunneling. As discussed before they are activated in the sensor through digital control commands. The control commands are send through an Xilinx spartan-3 FPGA. The basic idea is to read the sensor voltage from the chip and then send necessary trigger to the FPGA to activate either injection or tunneling as required. This procedure has to be repeated for all the channels until the voltages are between the specified limits. The algorithm used has been shown in Fig. 6.1.

The algorithm takes care of few important points:

- Tunneling is global. So it is activated for very small time and all the channel voltages are checked repeatedly so that they do not go too high. Whenever a channel voltage goes beyond 3.5V it is first brought down using injection.

- If initially all the channels are above 3.5V, all of them are injected to 3V. If one or more of them are below 2.5V and some are more than 3.5V, the higher ones are first injected to 3V and then tunneled.

The MATLAB program to implement the algorithm shown in Fig. 6.1 is provided in Appendix. A.

6.2 Calibration

The injection slopes varies from channel to channel due to mismatch in I_{ref} and due to opamp offset. Thus it is important to calculate the slope for each of them and then use them later to calibrate the sensor response. There would be variation also due to temperature but this has been ignored for this work. A test algorithm was devised to calculate the slopes which is shown in Fig 6.2.The MATLAB program to implement the algorithm shown is provided in Appendix. A.

6.3 Test setup

For testing purpose, we need to provide the sensor prototype with the supplies and inputs. The sensor needs two supplies-V_{ddD} which is 1.8-2V and a supply V_{dd} which is used to power the test buffers internal to the chip. These buffers are used to read out the sensor output and also other debug analog signals. The buffers are basically opamps in unity gain loop strong enough to drive the pin capacitances. The bias voltage for the opamp which is 1V is also provided externally. V_{dd} can be anything more than 5.8V(V_{ref4}) as that is the highest voltage we wanted to read. The other inputs are the ’Sync’(ENVP) and ’Data’ signals through which the commands are send.

All the inputs were controlled through MATLAB. A low-cost and portable test environment developed in our lab was used to provide the biasing to the chip. The test board DACs were accessed through National Instrument DAQCard-6024E. ’Sync’(ENVP) and ’Data’ signals were sent from a Spartan-III FPGA board. A state machine was implemented in the FPGA that takes in the triggers and send out the commands to the chip. The FPGA code is provided in Appendix. B. The triggers to control the FPGA was controlled by MATLAB and the test-station send them to the FPGA. A schematic diagram of the test setup is shown in Fig. 6.3 and a photo of the setup is shown in Fig. 6.4.

6.4 Measurement Results

To show the initialization and calibration algorithms together a MATLAB program was written to combine both of them. Repeated injection was performed on all the channels and their outputs stored. The result of one channel is shown in Fig. 6.5(a). There were 10 runs and in each of them there were 6 cycles of injection. After each run the sensor voltage was initialized back to the reset range. Each injection cycle was of 10s and V_{ref} used for this measurement was 5.57V. In Fig. 6.5(b) the slopes of each run has been plotted. As can be seen the error in slopes between each run is within a small band. The input DACs of the test station through which the sensor output is read has a low resolution which has added up to the slope error. This also shows the reliability of the linear injector and the sensor as the gain does not change from cycle to cycle. In Fig. 6.5(c), the difference between the real response with a linear regression curve has been shown which is mostly within a ±3% error band.

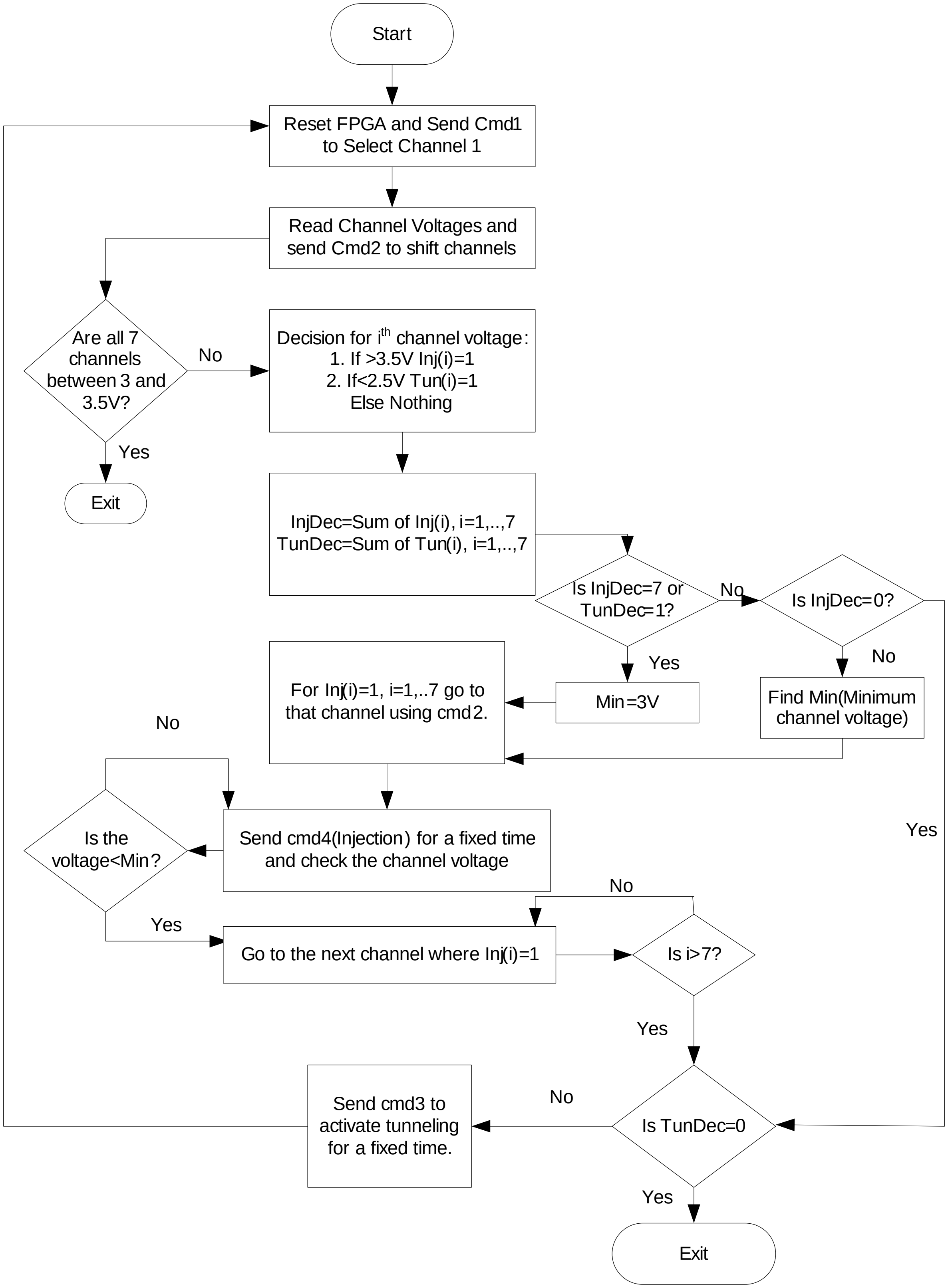

Start
Reset FPGA and Send Cmd1 to Select Channel 1
Read Channel Voltages and send Cmd2 to shift channels
Are all 7 channels between 3 and 3.5V?
No
Yes
Exit
Decision for i th channel voltage:
1. If >3.5V Inj(i)=1
2. If<2.5V Tun(i)=1
Else Nothing
InjDec=Sum of Inj(i), i=1,..,7
TunDec=Sum of Tun(i), i=1,..,7
Is InjDec=7 or TunDec=1?
No
Is InjDec=0?
Yes
Min=3V
No
Find Min(Minimum channel voltage)
For Inj(i)=1, i=1,..7 go to that channel using cmd2.
No
Is the voltage<Min?
Send cmd4(Injection) for a fixed time and check the channel voltage
Yes
No
Yes
Go to the next channel where Inj(i)=1
Is i>7?
Yes
Send cmd3 to activate tunneling for a fixed time.
No
Is TunDec=0
Yes
Exit

Figure 6.1: Flow Chart of Initialization Algorithm.

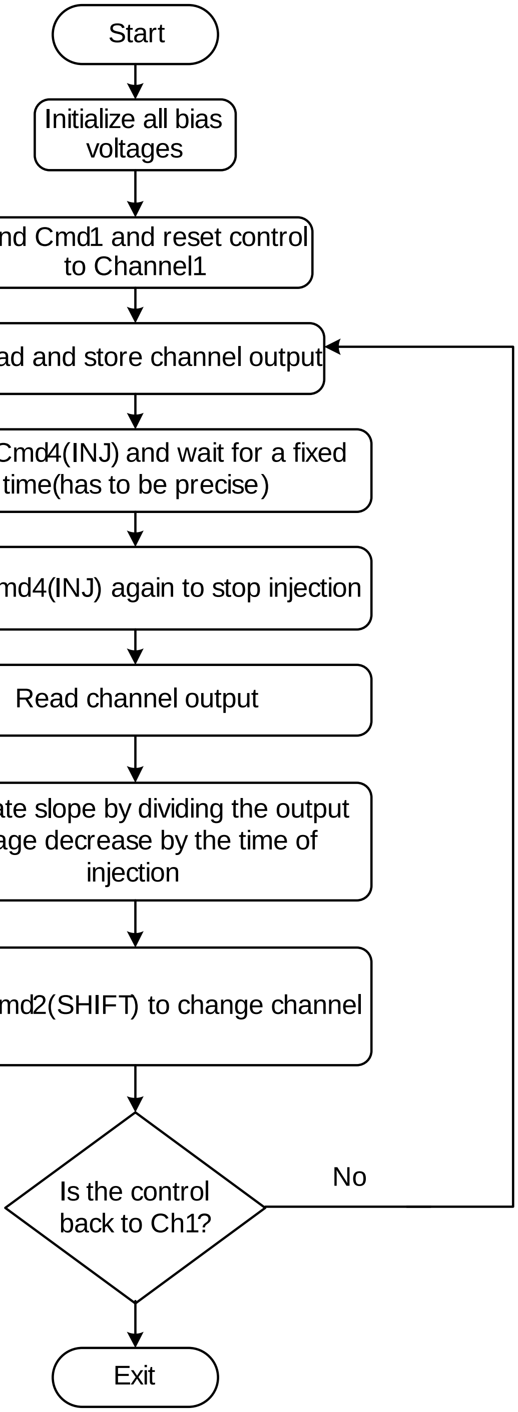

Start
Initialize all bias voltages
Send Cmd1 and reset control to Channel1
Read and store channel output
Send Cmd4(INJ) and wait for a fixed time(has to be precise)
Send Cmd4(INJ) again to stop injection
Read channel output
Calculate slope by dividing the output voltage decrease by the time of injection
Send Cmd2(SHIFT) to change channel
Is the control back to Ch1?
No
Exit

Figure 6.2: Flow Chart of Calibration Algorithm.

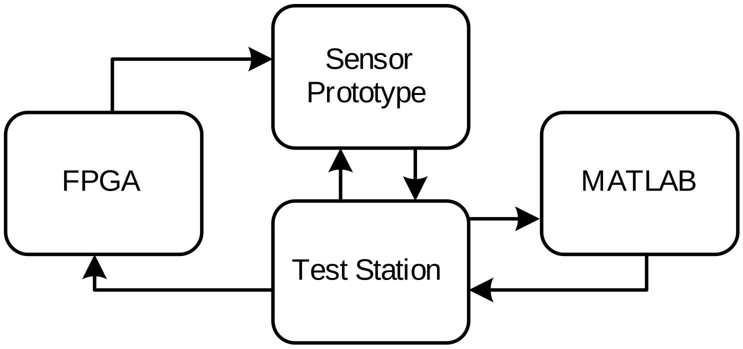


Figure 6.3: Schematic of the test setup.

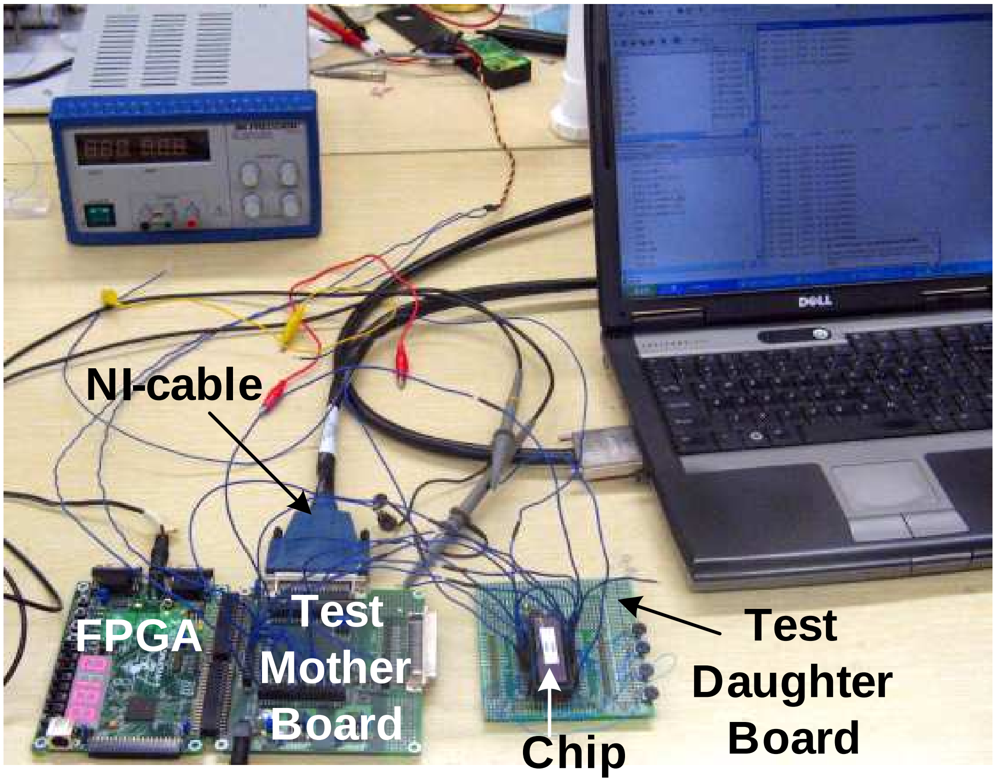
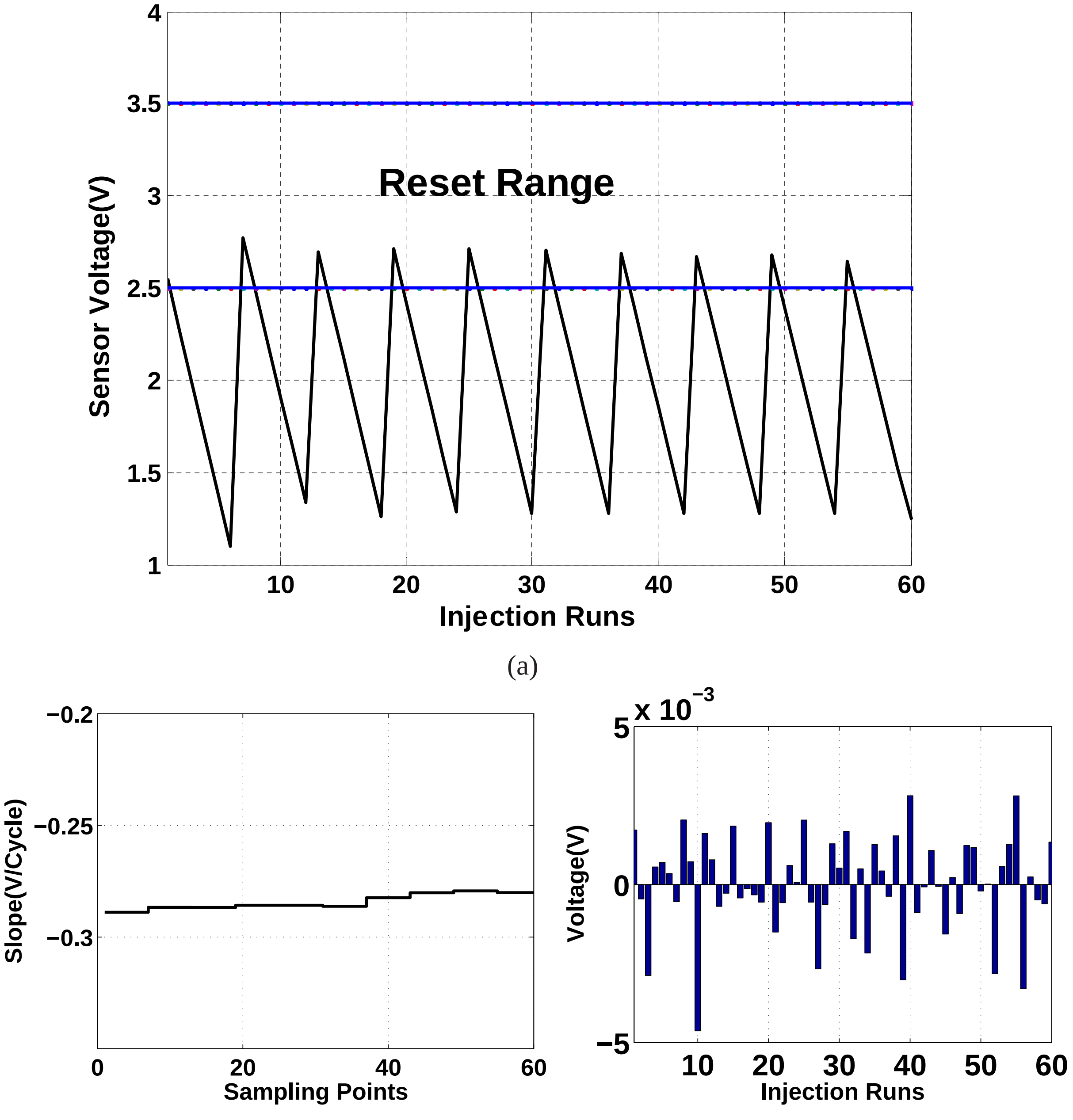


Figure 6.4: Photo of the test setup.



(a)

(b)

(c)

Figure 6.5: (a)Measurement result from one injector under-going repeated cycles of Injection and then initialized to a reset range; (b)Variation of slopes in the 10 injection cycles; (c)Residue of the injector response with a linear regression curve for each region.

Chapter 7

Conclusion and Discussions

The contributions of the thesis are summarised below.

First a novel self-powered static-strain sensor using a differential configuration of a linear p-IHEI(piezo-IHEI) circuit was proposed. The sensor is powered directly from ambient strain-variations and therefore does not require any batteries. For this reason, the proposed sensor could be miniaturized and used in in-vivo and embedded monitoring application. However, the sensor has a drawback, that it requires a minimum voltage-level (or strain-level) for operation which introduces dead-zones in its response which is unlike traditional strain-gauges that can be used to measure ultra-low levels of strain. There are also mismatches between the injectors which results in slope errors.

The second part of the thesis describes a novel compressive self-powering technique that overcomes the limited powering and sensing range of the sensor. The proposed technique uses a non-linear impedance circuit to dynamically load the output of a piezoelectric transducer and in the process reduce the magnitude of the output voltage at large levels of mechanical strain. The approach, however requires precise programming of level-crossing thresholds and precision mismatch compensation, both of which are achieved using different variants of a linear piezo-floating-gate injection circuit.

Measured results obtained from prototypes fabricated in a 0.5-μm standard CMOS process validate the proposed techniques.

The third part is the reliability testing of the piezo-floating-gate injection circuit that demonstrates repeatability of the results across multiple experimental runs. This involves in automated programming and calibration of the senor.

The thesis also presents a verilog-A and SPICE based model of the floating-gate transistor that has been used for simulating the long-term dynamics of the PFG sensor circuits.

7.1 Future Research Directions

There are multiple research directions available based on this sensor:

- *Temperature Compensation*

The current version of the p-IHEI self-powered sensor is sensitive to temperature. The voltage and current references depend on temperature, as a result I_{inj} and the injector slope too. Also the piezoelectric materials are also temperature dependent. To make the sensor more robust it needs to be made temperature independent.

- *Post-processing of sensor output*

The output of the sensor could be further processed to implement a industry standard strain measuring algorithm like the ’Rainflow Counting’([32]). This would make the sensor more useful and would be accepted more easily.

- *Ultrasonic Powering*

To power the sensor in programming mode piezoelectric materials in resonant mode(ultrasonic frequencies) can be used. This would make the whole sensor size much smaller than with the RFID interface.

- *Piece-wise linear compressive gain*

The compressive response described would be more useful if it could be made piecewise linear with two/three slopes without the non-linear part. This would make the strain calibration of the sensor much simpler.

APPENDICES

Appendix A

MATLAB Codes

A.1 Initialization:Decision.m

```
%Programs 7 channels to a voltage between 3.5V-2.5V
%Files ’Rst.m’, ’state.m’ and ’cmd.m’ needed.
%%DAC Pin numbers from test station%%%
DeleteNiCard;
InitializeNiCard;

r1=3;  r2=8;  %Reset
st1=2;  st2=2;  %State
c1=2;   c2=6;   %Command
se1=2;  se2=8;   %Send
v1=5;   v2=5;   %VddDig
b1=5;   b2=6;   %BufBias
va=1;     %Vdd
c=1;    %AnalogInput from chip

%%%Parameter Values%%%
vdddig=2;
bufbias=1;
vdd=7;
SetAnalogOutput(va,vdd);          %Vdd
SetSpecificBias(b1,b2,bufbias);     %BufBias
SetSpecificBias(v1,v2,vdddig);     %VddDig

Ch=[];
%%%%%%%%Check Channel Voltages%%%%%

while(1)
    flag=0;
    Rst(r1,r2); %Reset
```

```
state(st1,st2); %state
cmd(c1,c2);  %cmd1(Rst to ch1)
cmd(se1,se2);   %cmd1send
cmd(c1,c2); %cmd2(shift channel)
for i=1:1:7
    temp=0;
    for j=1:1:40
        data=GetAnalogInput(c);
        if(j>30)
            temp=temp+data;
        end
    end
    Ch(i)=temp/10;
    if(Ch(i)<3.5 && Ch(i)>3.0 )
      flag=flag+1;
    end
    cmd(se1,se2);   %cmd2send
    pause(.01);
end

Ch
%%%%If all channels are within range:exit
if(flag==7)
    break;end;

%%%Decision%%%
TunDec=0;
InjDec=0;
for j=1:1:7
    if(Ch(j)>3.5)
        Inj(j)=1;
        Tun(j)=0;
    elseif(Ch(j)<2.5)
        Tun(j)=1;
        Inj(j)=0;
    else
        Inj(j)=0;
        Tun(j)=0;
    end
TunDec=TunDec+Tun(j);
InjDec=InjDec+Inj(j);
end

TunDec
```

```
InjDec
%%%%%%Finding Minimum%%%%%
if(InjDec==7)
    Min=3.3;
else
    Min=Ch(1);
    for j=1:1:6
        if(Ch(j)<Min)
            Min=Ch(j);
        else
            Min=Min;
        end
    end
end

if(Min<2.5)
    Min=2.7;
end
Min

if(InjDec>0)
      for j=1:1:7
          j
        if(Inj(j)==1)
            Rst(r1,r2); %Reset
            state(st1,st2); %state
            cmd(c1,c2);  %cmd1
            cmd(se1,se2);   %cmd1send
            cmd(c1,c2); %cmd2
            for k=1:1:(j-1)
                cmd(se1,se2);   %cmd2send
                k=k+1;
            end
            cmd(c1,c2);
            cmd(c1,c2); %Cmd4(Inj)


        while(Ch(j)>Min)
            cmd(se1,se2);   %InjEnB
            pause(10);
            cmd(se1,se2);   %InjEnB
            temp=0;
            for k=1:1:20
                data=GetAnalogInput(c);
```

```
                     if(k>10)
                     temp=temp+data;
                     end
                end
                Ch(j)=temp/10
            end
            end
          end
    end

    if(TunDec>0)
        Rst(r1,r2); %Reset
        state(st1,st2); %state
        cmd(c1,c2);  %cmd1(Rst to ch1)
        cmd(c1,c2); %cmd2(shift channel)
        cmd(c1,c2);     %TunEn
        cmd(se1,se2);   %cmd3send
        pause(.001);   %Adjust Tunneling time
        cmd(se1,se2);   %cmd3send
    else
       break;
    end;
end
```

A.2 Associate files

A.2.1 Rst.m

```
function status = Rst (a,b)

SetSpecificBias(a,b,3.3);
 SetSpecificBias(a,b,0);
```

A.2.2 Cmd.m

```
function status = cmd (a,b)

SetSpecificBias(a,b,3.3);
SetSpecificBias(a,b,0);
```

A.2.3 State.m

```
function status = state (a,b)

i=1;
while(i<3)
    SetSpecificBias(a,b,3.3);
    SetSpecificBias(a,b,0);
    i=i+1;
end
```

A.3 Calibration

```
%%%%Calculates the slope of 7 channels of the sensor%%%%
%%%Files ’Rst.m’, ’state.m’ and ’cmd.m’ needed.%%
%%DAC Pin numbers from test station%%%
DeleteNiCard;
InitializeNiCard;
r1=3;  r2=8;  %Reset
st1=2;  st2=2;  %State
c1=2;   c2=6;   %Command
se1=2;  se2=8;   %Send
v1=5;   v2=5;   %VddDig
b1=5;   b2=6;   %BufBias
cal1=3; cal2=6;
va=1;     %Vdd
c=1;    %AnalogInput from chip

%%%Parameter Values%%%
vdddig=2;
bufbias=1;
vdd=7;


 SetAnalogOutput(va,vdd);          %Vdd
 SetSpecificBias(b1,b2,bufbias);     %BufBias
 SetSpecificBias(v1,v2,vdddig);     %VddDig
slope=[];
    Rst(r1,r2); %Reset
    state(st1,st2); %state
    cmd(c1,c2);  %cmd1(Rst to ch1)
    cmd(se1,se2);   %cmd1send
    cmd(c1,c2);
cmd(c1,c2);
cmd(c1,c2); %Cmd4
for i=1:1:7         %For 7 channels
    slope_sum=0;
    Ch_fin=[];
    Ch_ini=[];
    islope=[];
    for k=1:1:2        %For 2 slope calculations
        temp=0;
        for j=1:1:40    %For 10 Readings to get an avg
            data=GetAnalogInput(c);
            if(j>30)
                temp=temp+data;
            end
        end
        Ch_ini(k)=temp/10
        cmd(cal1,cal2); %Sends cmd4 twice in 10s
        temp=0;
        pause(11);
        for j=1:1:40    %For 10 Readings to get an avg
            data=GetAnalogInput(c);
            if(j>30)
                temp=temp+data;
            end
        end
        Ch_fin(k)=temp/10
        islope(k)=(Ch_fin(k)-Ch_ini(k))/9.5
        slope_sum=slope_sum+islope(k);
    end
    slope(i)=slope_sum/2
    Rst(r1,r2); %Reset
    state(st1,st2); %state
    cmd(c1,c2);
    cmd(c1,c2);
    cmd(se1,se2);   %cmd2send
    cmd(c1,c2);
    cmd(c1,c2);     %cmd4
end
slope
```

Appendix B

FPGA Code

```
‘timescale 1ns / 1ps
////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    11:11:21 07/17/2009
// Design Name:
// Module Name:    Reader
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 – File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////
module Reader(test, Calib, ClkCP, MODB, Clkwave, Servocont, Clk, Rst,
Mode, ComSel, ComTx, MOSI, MISO, SS, DClk, MOD, ASK, DataRx, SegDis,
An3, An2, An1, An0, DataIn, DataOut, DataEvp);
input  Clk, Rst, Mode, ComSel, ComTx, MISO;
input  [7:0] DataOut;
input Calib;
//input servoRst;
output MOSI, SS, DClk, MOD, ASK, DataRx, An3, An2, An1, An0,  MODB, DataEvp;
output [6:0] SegDis;
output [7:0] DataIn;
output test;


output ClkCP;
output Clkwave;
output Servocont;
reg Servocont;
reg [15:0] ClkCnt1;
reg [15:0] ClkCnt2;
reg [15:0] ClkCnt4;
reg [15:0] servocnt;
reg Clkwave;
reg ClkCP;
reg [4:0] wavebak;
reg [4:0] wavefor;
reg servoRst;

reg [7:0] DataIn;


reg [15:0] ClkCnt;


reg [7:0] BounceClkCnt;
reg ClkDiv,BounceClk;
reg ClkDiv4;

reg SS,ModeB,ComSelB,ComTxB;
reg CalibB;

reg [63:0] TxData;
reg [63:0] TxData1;
reg [3:0] SubState,Data1,Data2,Data3,Data4;
reg [7:0] Cnt;
reg [1:0] ModeState;
reg [19:0] iCnt;
reg flag;
reg iBTN0;
reg [3:0] Clk4;
//reg temp;

assign test=Clk4[2];

DigitDisplay D1 (Data1,Data2,Data3,Data4
,SegDis ,An3,An2,An1,An0,Clk ,Rst );



Buffer B1 (ClkDiv ,Rst ,ModeB,BTN2);
Buffer B2 (ClkDiv ,Rst ,ComSelB ,BTN1);
Buffer B3 (ClkDiv ,Rst ,ComTxB,BTN0);
Buffer B4 (ClkDiv ,Rst ,CalibB ,iCalibB );
Buffer B5 (ClkDiv ,Rst ,iBTN0 ,bBTN0);


assign DClk = ~ClkDiv & (~SubState[1]&SubState[0]);
assign MOSI = !(ModeState[1]&~ModeState[0]) & TxData[63];
//assign MOD = !(SubState[1]&SubState[0]) | TxData[63];
assign MOD = (SubState[1]&SubState[0]) & Clk4[3];
assign MODB= !MOD;
assign DataEvp= (SubState[1]&SubState[0]) & TxData1[63] & ClkDiv;
//assign MOD = ~MODn;
assign DataRx = MISO;
assign sBTN0=bBTN0 | BTN0;

assign ASK = 1'b0;


always @(posedge ClkDiv4 or posedge Rst)
begin
  if (Rst == 1'b1)
  begin
        Clk4[3:0]<=4'b1100;
  end
else
begin
        Clk4[3:1]<=Clk4[2:0];
        Clk4[0]<=Clk4[3];
end
end

//Command for calibration

always @(posedge ClkDiv or posedge Rst)
begin
  if (Rst == 1'b1)
  begin
         iBTN0<=1'b0;
         iCnt <=20'd0;
         flag <=1'b0;
  end
else
begin
  if(iCalibB==1'b1)
        begin
          flag <=1'b1;
          servoRst=~servoRst;
        end
        if(flag==1'b1)
        begin
                if(iCnt===20'd1)
                begin
                        iBTN0<=1'b1;
                        iCnt<=iCnt+1;
                end
                else if(iCnt===20'd10)
                begin
                        iBTN0<=1'b0;
                        iCnt<=iCnt+1;
                end
                else if(iCnt==20'd50010)
                begin
                        iBTN0<=1'b1;
                        iCnt<=iCnt+1;
                end
                else if(iCnt==20'd50020)
                begin
                        iBTN0<=1'b0;
                        iCnt <=20'd0;
                        flag <=1'b0;

                end
                else
                   iCnt<=iCnt+1;
        end
end
end

//Generating Bounce Clk for latching input data

always @(posedge ClkDiv or posedge Rst)
begin
```

```
  if (Rst == 1'b1)
  begin
    BounceClkCnt <= 8'd1;
    BounceClk <= 1'b0;
  end
  else
  begin
    if (BounceClkCnt == 8'd200)
    begin
      BounceClkCnt <= 8'd1;
             BounceClk <= !BounceClk;
    end
    else
    begin
      BounceClkCnt <= BounceClkCnt + 8'd1;
             BounceClk <= BounceClk;
    end
  end
end

// Latching input data
always @(posedge BounceClk or posedge Rst)
begin
  if (Rst == 1'b1)
  begin
    ModeB    <= 1'b0;
    ComSelB <= 1'b0;
    ComTxB   <= 1'b0;
         CalibB <= 1'b0;
  end
  else
  begin
    ModeB    <= Mode;
    ComSelB <= ComSel;
    ComTxB   <= ComTx;
         CalibB <= Calib;
  end
end

always @(negedge DClk or posedge Rst)
begin
  if (Rst == 1'b1)
    DataIn <= 8'b0;
  else
  begin
    if (ModeState == 2’b0)
    begin
      DataIn[7:1] <= DataIn[6:0];
      DataIn[0] <= MISO;
    end
  end
end

//Generating clock for internal state machine
always @(posedge Clk or posedge Rst)
begin
  if (Rst == 1’b1)
  begin
    ClkCnt <= 16’b0;
    ClkDiv <= 1’b0;
  end
  else
  begin
    if (ClkCnt == 16’d4800)//1500
    begin
      ClkCnt <= 16’d1;
      ClkDiv <= ~ClkDiv;
    end
    else
    begin
      ClkCnt <= ClkCnt + 16’d1;
      ClkDiv <= ClkDiv;
    end
  end
end

always @(posedge Clk or posedge Rst)
begin
  if (Rst == 1’b1)
  begin
    ClkCnt4 <= 16’b0;
    ClkDiv4 <= 1’b0;
  end
  else
  begin
    if (ClkCnt4 == 16’d1200)//1500
    begin
      ClkCnt4 <= 16’d1;
      ClkDiv4 <= ~ClkDiv4;
    end
    else
    begin
      ClkCnt4 <= ClkCnt4 + 16’d1;
      ClkDiv4 <= ClkDiv4;
    end
  end
end


//Clock for running charge pump at 1MHz
always @(posedge Clk or posedge Rst)
begin
  if (Rst == 1’b1)
  begin
    ClkCnt2 <= 16’b0;
    ClkCP <= 1’b0;
  end
  else
  begin
    if (ClkCnt2 == 16’d20)
    begin
      ClkCnt2 <= 16’d1;
      ClkCP <= ~ClkCP;
    end
    else
    begin
      ClkCnt2 <= ClkCnt2 + 16’d1;
      ClkCP <= ClkCP;
    end
  end
end

//Clock for Servomotor control
always @(posedge Clk or posedge Rst)
begin
  if (Rst == 1’b1)
  begin
    ClkCnt1 <= 16’b0;
    Clkwave <= 1’b0;
  end
  else
  begin
```

```
    if (ClkCnt1 == 16'd25000) //f=50MHz/(2*25k)=1kHz, T=1ms
    begin
      ClkCnt1 <= 16'd1;
      Clkwave <= ~Clkwave;
    end
    else
    begin
      ClkCnt1 <= ClkCnt1 + 16'd1;
      Clkwave <= Clkwave;
    end
  end
end



//Servomotor control


initial
begin
        wavebak = 4'b11_00;   //2ms pulse width
        wavefor = 4'b01_00;      //1ms pulse width
        servocnt=16'd1;
        servoRst=1'b1;
end



always @(posedge Clkwave or posedge servoRst)
begin
        if (servoRst == 1'b1)
        begin
                        Servocont <=1'b0;
                        servocnt <=16'd1;

        end
        else
                  begin
                if(servocnt <=16'd8000)   //stream of back pulses
                begin
                        Servocont <=wavebak[4];
                        wavebak[4:2]<=wavebak[3:1];
                        wavebak[1]<=Servocont;
                        servocnt <=servocnt+16'd1;
                end
                else
                begin
                        Servocont<=wavefor[4]; //forward pulses
                        wavefor[4:2]<=wavefor[3:1];
                        wavefor[1]<=Servocont;
                        if(servocnt==16’d15000)
                                servocnt <=16’d1;
                        else
                                servocnt<=servocnt+16’d1;
                end

                end
end


//Main state Machine


always @(posedge ClkDiv or posedge Rst)
begin
  if (Rst == 1’b1)
  begin
    ModeState <= 2’b0;
    TxData <= 64’b0;
         TxData1 <= 64’b0;
    Data1 <= 4’b0;
    Data2 <= 4’b0;
    Data3 <= 4’b1;
    Data4 <= 4’b0;
    SubState <= 4’b0;
    Cnt <= 8’b0;
    SS <= 1’b1;

  end
  else

  begin
 Data2[0]<=˜servoRst;
 Data4[0]<=flag;

    if (BTN2 == 1’b1)
    begin
      if (ModeState == 2’b0)
      begin
        ModeState <= 2'd1;
        Data3 <= 4'd2;
      end
      else if (ModeState == 2'd1)
      begin
        ModeState <= 2'd2;
        Data3 <= 4'd3;
      end
      else if (ModeState == 2'd2)
      begin
        ModeState <= 2'd0;
        Data3 <= 4'd1;
      end
    end
//======Mode Shift 1. Read 2. Write 3. PIE
    if (BTN1 == 1'b1)
      if (Data1 == 4'd15)
        Data1 <= 4'd0;
      else
        Data1 <= Data1 + 4'd1;
//======Command Select
    case (ModeState)
    2'd0:
    begin
      case (SubState)
      4'd0:
      begin
        if (sBTN0 == 1'b1)
        begin
          SS <= 1'b0;
          TxData[63:60] <= 4'b0100;
          TxData[59:56] <= Data1;
          TxData[55:0]  <= 56'b0;
          SubState <= 4'd1;
        end
      end
      4'd1:
      begin
        if (Cnt == 8'd15)
          SubState <= 4'd2;
        else
        begin
          Cnt <= Cnt + 8'd1;
          TxData[63:57] <= TxData[62:56];
          TxData[56] <= 1’b0;
        end
      end
      4’d2:
      begin
        SS <= 1’b1;
        Cnt <= 8’d0;
        TxData <= 64’b0;
        SubState <= 4’d0;
      end
      default: SubState <= 4’d0;
      endcase
    end
//======Read
    2’d1:
    begin
      case (SubState)
      4’d0:
      begin
        if (sBTN0 == 1’b1)
        begin
          SS <= 1’b0;
          TxData[63:60] <= 4’b0000;
          TxData[59:56] <= Data1;
          TxData[55:48] <= DataOut;
          TxData[47:0]  <= 48’b0;
          SubState <= 4’d1;
        end
      end
      4’d1:
      begin
        if (Cnt == 8’d15)
          SubState <= 4’d2;
        else
        begin
          Cnt <= Cnt + 8’d1;
          TxData[63:49] <= TxData[62:48];
          TxData[48] <= 1’b0;
        end
      end
      4’d2:
      begin
        SS <= 1’b1;
        Cnt <= 8'd0;
        TxData <= 64'b0;
        SubState <= 4'd0;
      end
      default: SubState <= 4'd0;
      endcase
    end
//======Write
    2'd2:
    begin
      case (SubState)
      4'd0:
      begin
        if (sBTN0 == 1'b1)
          begin
          SubState <= 4'd3;
          if (Data1 == 4'd1)
                                begin
            TxData[63:32]  <= 24'hFFFFFF;
                                TxData1[63:32] <= 24'hC95222;
                                end
          else if (Data1 == 4'd2)
                                begin
            TxData[63:32]  <= 24'hFFFFFF;
                                TxData1[63:32] <= 24'hC95481;
                                end
          else if (Data1 == 4'd3)
                                begin
             TxData[63:32]  <= 24'hFFFFFF;
                                TxData1[63:32] <= 24'hC956A3;
                                end
          else if (Data1 == 4'd4)
                                begin
             TxData[63:32]  <= 24'hFFFFFF;
                                TxData1[63:32] <= 24'hC95814;
                                end
          else if (Data1 == 4'd5)
                                begin
            TxData[63:32] <= 24'b0;
                                TxData1[63:32] <= 24'b0;
                                end
          else if (Data1 == 4'd6)
                                begin
             TxData[63:32] <= 24'b0;
```

```
                                TxData1[63:32] <= 24'b0;
        //                      servoRst=~servoRst;
                                end
                          else
                                begin
              TxData[63:32] <= 24'b0;
                                TxData1[63:32] <= 24'b0;
                                end

                         end

      end
      4'd3:
      begin
        if (Cnt == 8'd40)
          SubState <= 4'd4;
        else
        begin
          Cnt <= Cnt + 8'd1;
          TxData[63:32] <= TxData[62:31];
                         TxData1[63:32] <= TxData1[62:31];
          TxData[31] <= 1'b1;
                         TxData1[31] <= 1'b0;
        end
      end
      4'd4:
      begin
        Cnt <= 8'd0;
        TxData <= 64'h11111111_11111111;
        SubState <= 4'd0;
      end
      default: SubState <= 4'd0;
      endcase
    end
//======PIE
    default: ModeState <= 2'd0;
    endcase
  end
end
endmodule
```

Appendix C

Verilog-A Model

C.1 Injection Model

```
‘include ”constants.vams”
‘include ”disciplines.vams”

module Inj_model2(Vs, Vs1, Vd, plus, minus);
input Vs, Vs1, Vd;
inout plus, minus;
electrical Vs,Vs1, Vd, plus,minus;
branch (plus,minus) cond;
branch (Vs1,Vs) cond1;

parameter real a1=3e−23;
parameter real k2=6.55;
parameter real k3=5e6;


analog begin

if ( I(cond1)>0 & V(Vs)>4.3) begin
I(cond) <+ a1*I(cond1)*(1+k3*I(cond1))*exp(V(Vs,Vd)*k2);
end
else I(cond) <+ 0;
end
endmodule
```

C.2 Tunneling Model

```
‘include ”constants.vams”
```

```
‘include ”disciplines.vams”

module Tunn_model2(Vt,Vfg,plus,minus);
input Vt,Vfg;
inout plus, minus;
electrical Vt,Vfg,plus,minus;
branch (plus,minus) cond;

parameter real a=9.35*1e8;
parameter real b=800;

analog begin
if ( V(Vt,Vfg)>0) begin
        I(cond) <+ a*exp(-b/V(Vt,Vfg));
end
else
        I(cond) <+ 0;
end
endmodule
```

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] J. Borodkin, N. Caldwell, and S. Hollister,“The influence of mechanical strain magnitude on bone adaptation to porous coated implants,” *Proc. IEEE Eng. Med. Biol. Soc. Conf.*, 1999, p. 768.

[2] B. D. Beynnon and B. C. Fleming, “Anterior cruciate ligament strain in-vivo: A review of previous work,” *J. Biomech.*, vol. 31, pp. 519525, 1998.

[3] T. Wright, F. Glowczewski, D. Cowin, and D. Wheeler, “Ulnar nerve excursion and strain at the elbow and wrist associated with upper extremity motion,” *J. Hand Surg. Amer.*, vol. 26, no. 4, pp. 655662, July 2001.

[4] R. L. Pradhan, E. Itoi, Y. Hatakeyama, M. Urayama, and K. Sato, “Superior labral strain during the throwing motion: A cadaveric study,” *Amer. J. Sports Med.*, vol. 29, pp. 488492, 2001.

[5] B. Burr, R.B Martin, M.B Schaffler and E.L Radin, “Bone remodeling in response to in vivo fatigue microdamage,”*Journal of biomechanics*, 1985.

[6] P.J Ehrlich and L.E Lanyon, “Mechanical strain and bone cell function: a review,” *Osteoporosis International*, 2002; 13(9):688-700

[7] W. K. Schomburg, Z. Rummler, P. Shao, K. Wulff and L. Xie, “ The design of metal strain gauges on diaphragms,” *Journal of Micromechanics and Microengineering*, Volume 14 Number 7 June 2004.

[8] Yongdae Kim; Youngdeok Kim; Chulsub Lee; Sejin Kwon, “ Thin Polysilicon Gauge for Strain Measurement of Structural Elements,” *IEEE Sensors Journal*, vol.10, no.8, pp.1320-1327, Aug. 2010.

[9] Kuo-Chih Chuang, Chien-Ching Ma and Chao-Hsiang Wang, “ Performance analysis of a fiber Bragg grating filter-based strain/temperature sensing system based on a modified Gaussian function approximation method,” *Appl. Opt.* , 50, 5243-5255 (2011).

[10] C. Huang, S. Chakrabartty, “An Asynchronous Analog Self-powered Sensor-Data-Logger with a 13.56MHz RF Programming Interface,” *IEEE Journal of Solid-State Circuits*, Feb 2012.

[11] C. Huang, N. Lajnef and S. Chakrabartty, “Calibration and characterization of self-powered floating-gate usage monitor with single electron per second operational limit,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 3, pp. 556-567, Mar 2010.

[12] S. Chakrabartty and N. Lajnef, “Infrasonic Power-Harvesting and Nanowatt Selfpowered Sensors,” *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp.157-160, May 2009

[13] C. Huang, P. Sarkar and S. Chakrabartty, “ Rail-to-Rail, Linear Hot-Electron Injection Programming of Floating-Gate Voltage Bias Generators at 13-Bit Resolution,” *IEEE Journal Of Solid-State Circuits*, vol.46, no:11, 2011.

[14] P. Sarkar, C. Huang and S. Chakrabartty, “ A Self-powered Static-strain Sensor based on Differential Linear Piezo-floating gate Injectors,” *ISCAS*, to appear May 2012.

[15] S. Roundy and P K Wright, “ A piezoelectric vibration based generator for wireless electronics,” *Smart Materials and Structures*, vol.13, 2004, pp.1131-1142.

[16] Geffrey K. Ottman, Heath F. Hofmann, Archin C. Bhatt and George A. Lesieutre “Adaptive Piezoelectric Energy Harvesting Circuit for Wireless Remote Power Supply,” *IEEE Transactions on Power Electronics*, VOL. 17, NO. 5, September 2002.

[17] Sodano, H.A., Park, G. and Inman, D.J., “A Review of Power Harvesting Using Piezoelectric Materials,” *Shock and Vibration Digest*, Vol. 36, No. 3, pp. 197-206.

[18] N.G Elvin and A.A Elvin, “A General Equivalent Circuit Model for Piezoelectric Generators,” *Journal of Intelligent Systems and Structures*, May 2008.

[19] P. Hasler, “Foundations of learning in analog VLSI,” *Ph.D. dissertation*, Dept. Comput. Neural Syst., California Inst. Technol., Pasadena, CA, 1997.

[20] K. Rahimi, C. Diorio, C. Hernandez, and M. D. Brockhausen, “ A simulation model for floating-gate MOS synapse transistors,” *Proc. IEEE Int. Symp. Circuits Syst., Phoenix, AZ* , May 2002, pp. 532535.

[21] S. R. Platt, S. Farritor, K. Garvin, and H. Haider, “The use of piezoelectric ceramics for electric power generation within orthopedic implants,” *IEEE/ASME Trans. Mech.*, vol. 10, no. 4, pp. 455-461, Dec. 2005.

[22] “PSI-5A4E Piezoceramic Sheets and their properties,” Piezo Systems, Inc.

[23] G. Meltz, W. Morey and W. Glenn, “Formation of Bragg gratings in optical fibres by a transverse holographic method,” *Opt. Lett.* vol. 14, pp. 823-825, 1989.

[24] B. Lee, “Review of the present status of optical fiber sensors,” *Optical Fiber Technology.* vol. 229, no. 2, pp. 57-79, Apr. 2003.

[25] M. Kreuzer, “Strain Measurement with Fibre Bragg Grating Sensors,”

[26] J. Sirohi and I. Chopra, “Fundamental understanding of Piezoelectric Strain Sensors,” *Journal of Intelligent Material Systems and Structures* vol. 11-April 2000.

[27] French P.F. and A.G.R. Evans, “Piezoresistance in Polysilicon and Its Applications to Strain Gauges,” *Solid-State Electronics,* 1989. 32: p. 110.

[28] Wen H. Ko, Darrin J. Young, Jun Guo, Michael Suster, Hung-I Kuo, N. Chaimanonart, “A high-performance MEMS capacitive strain sensing system,” *Sensors and Actuators A*, 133 (2007) 272277.

[29] Laurel Sheppard, “Innovative Infrastructure: Smart Bridges,” *Buildipedia.com*, Wed Nov 10 2010.

[30] Na Yan and Hao Min, “A High Efficiency ALL-PMOS Charge Pump for Low-Voltage Operations,” *Asian Solid-State Circuits Conference*, 2005 , vol., no., pp.361-364, 1-3 Nov. 2005.

[31] P. Sarkar, C. Huang and S. Chakrabartty, “An Ultra-linear Piezo-Floating-Gate Strain-Gauge for Self-powered Measurement of Quasi-static-strain”, *IEEE Transactions on Biomedical Circuits and Systems(TBioCAS)*, minor revision 2012.

[32] C. Amzallag, J.P. Gerey, J.L. Robert and J. Bahuaudl, “Standardization of the rainflow counting method for fatigue analysis”, *International Journal of Fatigue*, Volume 16, Issue 4, June 1994, Pages 287293.