THESIS

LOW POWER BIOSENSOR AND DECIMATOR DESIGN

Submitted by

Kristin Scholfield

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Summer 2013

Master's Committee:

Advisor: Tom Chen

George Collins
Stuart Tobet

ABSTRACT

LOW POWER BIOSENSOR AND DECIMATOR DESIGN

This paper examines the use of low power circuits applied to biosensors used to observe neurotransmission. The term “biosensors” in the broadest sense describes many devices which are used to measure a biological state e.g. neural signal acquisition. The methods for developing biosensors are just as diverse, but one common thread is that many biomedical devices are battery operated and require low power for mobility. As biosensors become more complex they also require more functions such as data storage, digital signal processing, RF transmission etc. The more functions a sensor needs, the tighter the constraint for power consumption on a battery operated device becomes. In order to solve this problem, biosensors are increasingly being designed for low power consumption while weighing tradeoffs for performance and noise. Designers accomplish this by lowering the supply voltage, which reduces the overall size, and thus the load, of the devices. The amount of individual components will also be reduced, allowing for a smaller, faster device.

Biosensors are important because they grant the ability for scientists to better understand complex biological systems. While many other methods exist for observing biological systems, electrochemistry is a practical method for measuring redox reaction because it senses chemical reactions on the surface of an electrode. The reaction will create a current, which can be interpreted via electronics. With the use of electrochemistry, scientist can cheaply and practically observe changes occurring between cells. On the engineering side, modern silicon processes provide small, tightly packed microelectrodes for high spatial resolution. This allows scientists to detect minute changes over a small spatial range. With an array of electrodes on the scale of

1000s, electrochemistry can be used to record data from a sizable cellular sample. Such an array could be used to identify several biological functions such as communication between cells.

By combining known electrochemistry methods with low power circuit designs, we can create a biosensor that can further advance the understanding of the operation of cells, such as neurotransmission. The goal of our project is to create a device that uses electrochemistry to detect a redox reaction between a chemical, such as nitric oxide, and an electrode. The device needs to be battery operated for mobility and it must contain all needed electronics on chip, including amplification, digital signal processing, data transmission etc. This requires a surface of electrodes on chip that can handle the environment needed for a living tissue such as: specific temperature, pH and humidity. In addition, it requires a chip that is low power and which produces little heat.

This thesis describes two separate designs, both of which are part of a final biosensor design that will be used for the detection of nitric oxide. The first design is a biosensor microelectrode array. The array will be used along with electrochemistry to detect the release of nitric oxide from a living tissue sample. The electrodes are connected to a chain of electronics for on chip signal processing. The design runs at a voltage of 3V in a 0.6µm CMOS process. The final layout for the microelectrodes measured approximately 4.84mm^2 with a total of 8,192 electrodes and consumed 0.310mW/channel.

The second design is a low power decimator for a sigma-delta analog to digital converter designed for biomedical applications. The ADC will be used along with a chain of amplifying electronics to interpret the signals received from the microelectrode array. The design runs at a voltage of 0.9V in a 0.18µm CMOS process. Its final layout measured approximately 0.0158mm^2

and consumed 3.3uW of power. The ADC and microelectrode array were designed and fabricated separately to ensure their validity as standalone designs.

ACKNOWLEDGMENTS

I would like to thank Dr. Tom Chen for his advice and experience over the past years as well as the National Science Foundation for their grant DGE-0841259. I would also like to thank Ryan Selby, William Wilson, Tucker Kern and William Tedjo for their design work on the modulator, TIA and potentiostat. In addition, I would like to thank Avago Technologies Inc. for their collaboration with the design and manufacturing of the biosensor chip and Texas Instruments (National Semiconductor) for their collaboration with the design and manufacturing of the decimator and its corresponding components. Finally, I would like to thank my husband, Matthew for all of his help and support.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

The world of biosensors is as diverse as it is complex, providing many different sensors for the detection of numerous biological states. Glucose monitoring, DNA detection, and analysis of living tissue are a few examples of the range of biosensing technologies [1] [2] [3]. The analysis of living tissue has special significance, having varied applications in both the scientific and medical communities. One such application involves observing the release of nitric oxide (NO) from cells, which is known to have an important role in biological systems such as signal transduction in neurotransmitters, regulation of blood pressure and cystostatic agents, such that NO can suppress cell growth and division [4]. Though NO is known to play a part in these systems, questions remain as to what impact NO has on them as well as how NO could be used to affect them.

NO is a reactive oxygen species (ROS) meaning that it contains one or more unpaired electrons, causing it to be chemically unstable [5]. NO is electrochemically active and reacts quickly in oxygen, having a half-life of two to six seconds [4]. As a result, NO is difficult to measure, requiring expensive equipment, costly reagents and time consuming techniques or indirect methods of detection. In addition, to effectively measure NO as it affects biological systems an *in situ* method is required, which is not always feasible.

As a signal transducer, NO release has been observed to play a role in the release of neurotransmitters [6]. In the classic picture of neurotransmission via chemicals, information is transferred between neurons in one direction via discrete synapses [7]. With NO as a neurotransmitter signal transducer, rapid three-dimensional diffusion occurs. In addition, the concentration of NO reduces slowly. When observing NO 20μm from the source, the concentration will be reduced by 10%, assuming a 5 second half-life. Even with a half-life of

fewer than 5 seconds, NO can theoretically influence function within a diameter of approximately 300μm. This makes microelectrodes ideal for studying NO and its effect on neurotransmission as they can provide spatial resolution of 30μm and they can cover millimeters of area which in turn permits an observer to monitor NO on an appropriate scale as it diffuses through the tissue.

One standard method of NO detection is chemiluminescence [8]. This method relies on the detection of photons created when NO reacts with ozone (O_3) to create NO_2. Chemiluminescence has the advantages of being a well established method with high selectivity and sensitivity, but it is also expensive and time consuming. In addition, the method involves mixing a stream of NO gas with ozone in front of a photomultiplier which increases noise in the signal and is thus not applicable for *in situ* measurements. Another method, fluorometry, uses a fluorescent indicator that reacts with NO to signal its presence. As with chemiluminescence, fluorometry is time consuming, and has a high dependence on pH and pure commercial reagents, all of which makes the method expensive.

To skirt the issue of time consumption, one could use electron paramagnetic resonance (EPR) spectroscopy which can detect a molecule with unpaired electrons, such as NO [9]. EPR uses the interaction of an external magnetic field with the unpaired electron to detect the presence of a molecule. This method has the advantage of allowing *in situ* measurements. However, EPR is insensitive at detecting NO in its liquid phase which is essential for studying its affects in living tissues. Thus, EPR can only be used for indirect detection of liquid NO, requiring NO traps to form an EPR-detectable product.

The above problems can be solved using modern silicon technologies and electrochemistry. Electrochemistry, which can detect a reaction between NO and an electrode, is the most practical method for detection of NO *in situ* [8]. Silicon technology offers the ability to construct a large array of microelectrodes that allows great spatial resolution as well as fast detection. The spatial resolution provided by such an array is especially important to being able to detect how NO moves in all directions and how it contributes to communication between cells. Electrochemistry allows the detection of a NO reaction based on electron exchange between the chemical and the electrode [4]. This makes electrochemistry detection of NO both simple and low cost with minimal to no reagents required for detection. It does have some drawbacks such as tedious calibration, high temperature dependence and the need for frequent cleaning to prevent fouling.

For the reasons outlined above, we chose to develop a biosensor array design. This design incorporates the use of electrodes and electrochemistry to detect a redox reaction from a chemical such as NO. Not only does this make the detection rather simple, but it also offers spatial resolution that is not often seen even in current electrode array designs. Many of the designs only include a few electrodes, with the larger array containing 100 electrodes [10]. Even with this high of density, the chip only has a spatial resolution of 400μm. Diffusion of NO could be detected just 20μm from the source, so a higher resolution, such as the 30μm offered by this design, is highly desirable.

All biosensors, regardless of their detection methods, are being developed to be more diverse, effective and power efficient. Power efficiency is the goal of many biosensors as they are often required to be battery operated for mobility and versatility of monitoring. Glucose monitors, for example, must be attached to a person, making battery operation paramount. These

types of devices require excellent battery life so that a person can freely go about his or her business. In addition, these devices require added functionality such ADCs and RF transmission so that the information can be transmitted to an external source. Thus, power efficiency is one of the greatest goals of many biosensor devices.

There are many approaches to reducing the power consumption of any given device. Given that power equals voltage times current, reducing the size and number of elements in a design is the easiest way to reduce power consumption. This reduces the load of the circuit, thus reducing the amount of current drawn. In addition, lowering the supply voltage, and having circuits that can function at lower voltages will also reduce the overall power consumption.

A decrease in power consumption does, however come at a cost. The performance of the design can decrease when the power is lowered since an increase in frequency will increase the power consumption [10]. Leakage power also becomes a bigger factor as the supply voltage, and thus the threshold voltage of the transistors, is scaled down. Additionally, the lowered supply voltage can increase noise observed in the measurement.

This thesis presents two separate designs to achieve the end goal of detecting a reaction between NO and an electrode *in situ* with low power consumption. While the end goal is to detect NO specifically, the current setup allows us only to see if there is a redox reaction at the electrode. The first step in creating this device is to create an electrode array with adequate spatial resolution. A microelectrode array was created consisting of 128x64 electrodes in a 0.6µm CMOS process which provides a spatial resolution of 30µm. The design also requires several electronic components. The first of these is a potentiostat. The potentiostat's function is to hold the electrodes at a specific potential, creating a known amount of current. Thus, when a

change on the electrodes such as redox reaction occurs, the potential between electrodes changes, thus changing the amount of current produced. The potentiostat created for this chip provides up to 10,000 V/s cyclic voltammetry. This means that the potentiostat can detect a change of up 10,000V in a single second allowing the circuit to measure rapid potential changes.

This circuit design also includes a transimpedance amplifier (TIA) and main amplifier for amplification of the electrode signal. The TIA is designed to receive a current input and translate it to a voltage output. The TIA input is directly connected to an electrode so that it can receive the appropriate current. The main amplifier is a simple amplification device that is necessary due to the small amount of voltage produced by the TIA. Finally, the design includes a non-overlapping clock generator to control the switches in both the TIA and the main amp. The clocks must not overlap so that no two switches are closed at the same time when they should not be. The final design has 128 on-chip read channels, operates on a 3V power supply and consumes 39.65mW of power.

The second design is part of a larger back-end electronics design for signal processing. These electronics include the potentiostat, TIA, main amplifier, an ADC and digital signal processing. For this design, a sigma-delta modulator combined with a decimator is used for the analog to digital conversion. The modulator design will receive an analog signal from the main amp. It will then translate the signal into a digital stream of ones and zeros. 0V in produces all zeros and 0.9V in produces all ones, with varying amounts of ones and zeros depending on how close the signal is to the extreme. The decimator side receives this string of numbers and filters it, creating a digital decimal value to represent the analog input. The decimator was designed with a bit-serial architecture, sacrificing the speed that can be gained with a bit-parallel architecture in favor of the low power consumption and small size offered by a bit-serial design.

The decimator design, which is currently separate from the electrode array design, operates on a 0.9V power supply in a 0.18µm CMOS process and consumes 3.3µW of power.

Both of these designs are part of the larger goal of creating a biosensor to observe NO *in situ*. The electrodes have been designed to have one section of 128 electrodes active at any given time, allowing the user to switch between active electrode regions freely. The active region measures approximately 230x255µm so that the user can easily observe certain areas of a given tissue sample. Each of the electrodes is connected internally to a TIA and main amp created in the 0.6µm processes. Additionally, the electrodes have been connected directly to external pins so that they can be tested with new TIA designs as the designs are modified. This gives the designers freedom to change and adapt their individual circuit design without having to recreate the electrodes, which are costly to manufacture. Once the TIA, main and ADC components are verified and perfected, they can be added to the electrodes to create a final biosensor product for the observation of a redox reaction at the electrodes. Once the device is created, it will allow scientists to more accurately, cheaply and quickly observe this reaction.

CHAPTER 2: EXISTING RESEARCH

There are many designs available for both sigma delta ADCs and integrated biosensor arrays. Each design has its trade-offs in power, size, and speed. Some of the designs require high performance and will often result in higher power consumption. However, since low power consumption is desired, speed can be sacrificed and the size of the design reduced to achieve the power consumption goal. This design runs at a clock speed of 1MHz or 1,000,000 samples per second. Given that the biological components decay at a half-life of seconds, a 1MHz clock speed is more than adequate.

2.1: POTENTIOSTATS, TIAs AND EXISTING BIOSENSORS DESIGNS

Biosensors designs have useful applications for signal recording and tissue stimulation. Often this is accomplished through the use of electrochemistry and electrode arrays on chip for the ability to cheaply and quickly detect and record changes. For signal recording, an array of electrodes can be fabricated and covered by a tissue sample. The electrodes are designed to produce a constant current when a reaction occurs between them and the tissue. This is done with the use of an external bias and a potentiostat. The potentiostat is connected between a reference electrode (RE) and counter electrode (CE) on either side of an array of working electrodes (WE), as seen in Figure 1. The potentiostat maintains a constant current via the external bias voltage, Vbias. Release of a chemical, such as NO, from the tissue sample creates an oxidation reaction on the electrodes which will cause a change in current. A TIA connected to the electrodes can detect this current change and translate it to a voltage which can later be amplified and read out.

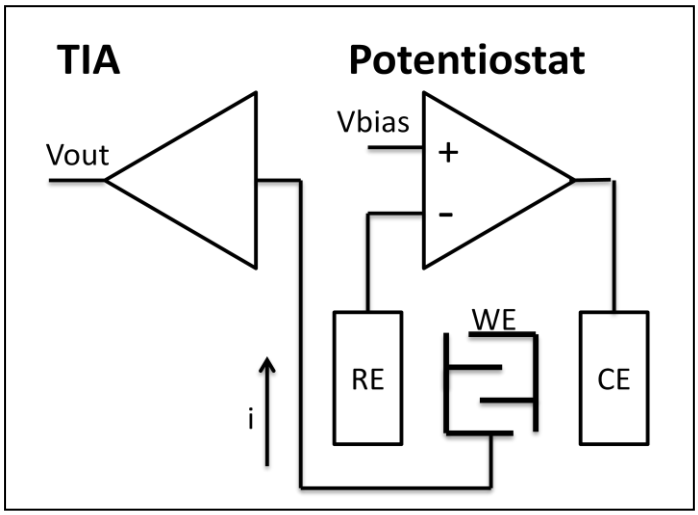

Figure 1: A top level model of the TIA and potentiostat interacting with the electrodes

The TIA is an amplifier that has been configured specifically to translate a current to a voltage. The easiest way to do this is to connect a resistor in a feedback loop as shown in Figure 2. All of the input current will flow through the resistor and thus the voltage will follow Ohm's law such that

$$V_{out} = I_{in} * R. \qquad (5)$$

However, for this application the input will be in the range of hundreds of pA with an output that should have a magnitude in mV. According to equation (5) we would then need a resistor in the MΩ range to accomplish this which is impractical to manufacture in a CMOS process. Since this chip is designed to have no external electronic components, a different method is required to accomplish the conversion from current to voltage.

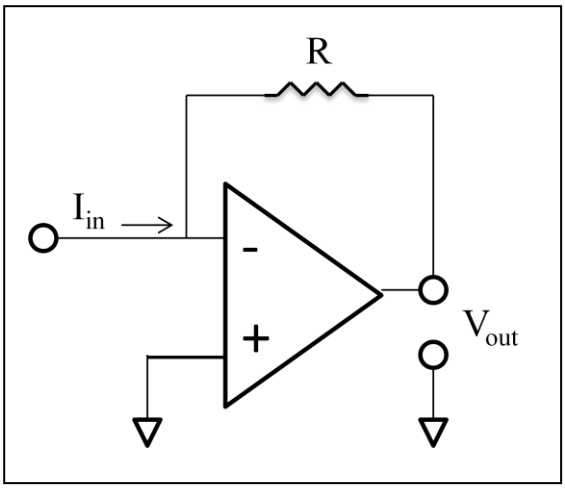

Figure 2: The simplest form of TIA made of a single amplifier combined with a resistor.

To circumvent the resistor problem, a switched capacitor can be used in place of the resistor, as seen in the first part of Figure 3 [11]. The equivalent resistance is modeled as

$$R = \frac{1}{C_s f}. \qquad (6)$$

In this case, f is the frequency at which the switches are closed with non-overlapping clocks so that neither is closed at the same time. According to equation (6) if we wanted a 1MΩ resistor and had a switching frequency of 1MHz, we would need a capacitance value of 1pF, which is practical for CMOS manufacturing. The switched capacitor principle can easily be applied to the simple TIA op amp to create a feasible circuit, as seen in the second part of Figure 3.

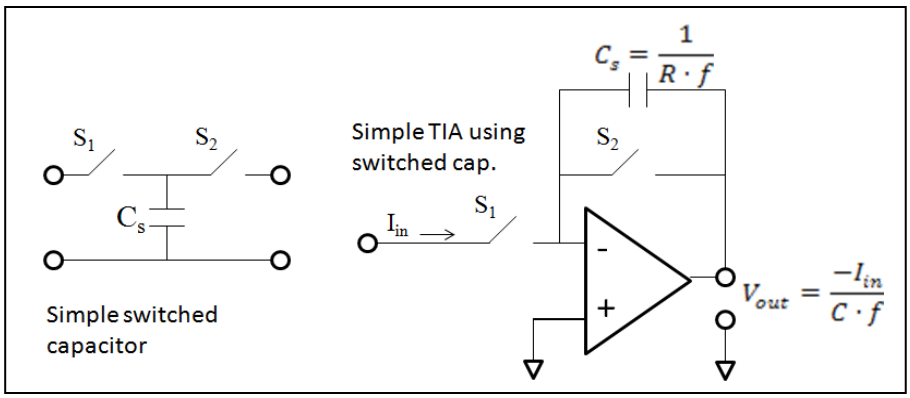

Figure 3: A simple switched capacitor that will act as resistor both by itself and in a TIA.

The TIA and potentiostat are some of the fundamental building blocks for interpreting and controlling the signals from the electrodes and are thus necessary for the design as a whole. The main emphasis for this thesis, however, is the design of the electrode array. Many electrode arrays currently in use for biology include only a few electrodes allowing detection of substances like NO but not permitting much precision as to how fast and where a chemical is diffusing through a sample. The larger scale arrays will number between 60 and 100 electrodes [12] [13]. In addition, the electrodes can be fairly large, measuring as much as 80µm in diameter with up to a 400µm pitch. These sizes can be sufficient for measuring certain reactions from a biological sample. Greater spatial resolution is required, however, for detecting NO as it moves through a system because in just 20µm its concentration will have already reduced by 10%.

Many other groups have created integrated sensor arrays, including microelectrodes and some electronics. One such group created a device for cortical monitoring with the goal of integrating up to 1000 multiplexed channels [14]. The power budget for the device given 6 channels was 40mW [15]. The power consumption of 6.67mW per channel is high and an expansion to 1000 channels is not practical. With 1000 channels active concurrently, the chip would burn over 6W of power. Although not all channels would actually be on concurrently, the overall power consumption is high for a battery operated device. In the end this amount of power consumption would result in a reduced battery life, and could produce undesired amounts of heat, which would be detrimental to the biological tissue that is being monitored.

Another group designed a system for neural signal acquisition from the central/peripheral nervous system and included 32 read channels and on chip ADCs [16]. The chip also included a multiplexer to connect the 32 channels to two ADCs. Overall the chip consumed 22mW, 10mW of which was consumed by the ADC. Since each of the channels is multiplexed through one of

the ADCs, this gives a power consumption of just over 10mW per channel for this design. Furthermore, this design did not include on-chip electronics such as a potentiostat and on-chip electrodes which would be required for a standalone device.

A final design is meant to observe neuron activity in the brain. This design had 100 available read channels, which is much more than any of the other designs and much closer to the number available in our design [13]. Since this design was made for an implant, it had an acceptable amount of power consumption, 0.135mW/channel, which is several orders of magnitude smaller than the other designs. However, because of the constraints for the application, it does not include on-chip electrodes or the electronics that would be used to govern the electrodes such as the potentiostat. This makes a direct comparison between their work and the design presented in this thesis difficult. Each of these designs is compared in Table 1.

Table 1: A comparison of power consumption for previous biosensor designs

Number of read Channels	**Total Power**	**Power per Channel**	**Process Used**
6 [15]	40mW	6.67mW/channel	0.18µm CMOS
32 [16]	22mW	10mW/channel	0.6µm CMOS
100 [13]	13.5mW	0.135mW/channel	0.5µm CMOS

2.2: ANALOG TO DIGITAL CIRCUITS AND EXISTING DECIMATOR DESIGNS

Analog to digital conversion is useful for many types of devices and is especially useful for music recording and digital signal processing (DSP). For music, analog recording is often used, but most music is stored in a digital format thus requiring a conversion. In DSP, ADCs are useful when an analog signal needs to be stored or transmitted digitally. Because of this, ADCs can have various designs each with its advantages depending on its application.

The simplest design is flash ADC, which employs resistive ladder networks and comparators to convert from analog to digital [17]. A 2-bit resolution flash ADC can be seen in Figure 4. The resistive ladder acts as a voltage divider, giving each comparator a unique reference voltage. When the input analog voltage exceeds the reference, the comparator will output a one. The output from the comparators is called thermometer or unary code. The numbers of ones in a given sequence will represent the decimal number so that 100 represents 1, 110 represents 2, 1110 represent 3 etc. A priority encoder will generate a binary number based on the given unary code input. This type of encoder will take the input 110, for example, and output 10 for a binary 2.

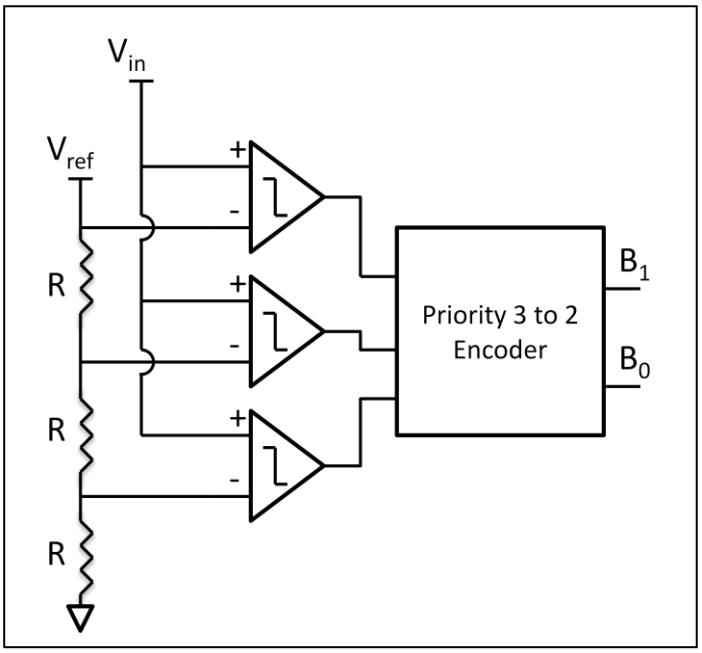

Figure 4: A 2-bit resolution flash ADC. V_{in} represents the analog input and B_1 and B_0 are the 2-bit digital binary outputs.

The advantage of flash ADCs is that they are extremely fast and easy to design. However, for accurate conversion, the flash ADC requires precise resistors with little variation. Finding resistors that match closely can be difficult and variation will add error to the final output. In addition, as the resolution is increased, so is the number of comparators and resistors required.

For n-bit conversion, the flash ADC requires 2^n-1 comparators and resistors. This means that greater resolution also results in more power consumption. Because of this, flash ADCs are generally not used for higher than 8-bit resolution or for low power designs. For our application we require a 10-bit resolution, low power ADC to be manufactured on chip, causing the flash ADC design to be impractical.

Another option for conversion is the successive approximation register (SAR) ADC [18]. The basic structure of this ADC is fairly simple, consisting of a sample and hold circuit, an analog comparator, a SAR with an N-bit register and a digital to analog converter (DAC) as seen in Figure 5. The track and hold simple samples the incoming analog signal and holds its value constant for a specified amount of time. The SAR starts with the MSB and initializes it to 1, which sets the initial $V_{DAC}=V_{REF}/2$. The comparator evaluates the two values and if Vin is greater than V_{DAC}, the bit remains at 1. Otherwise, the bit is set to 0. This same pattern continues for all N bits.

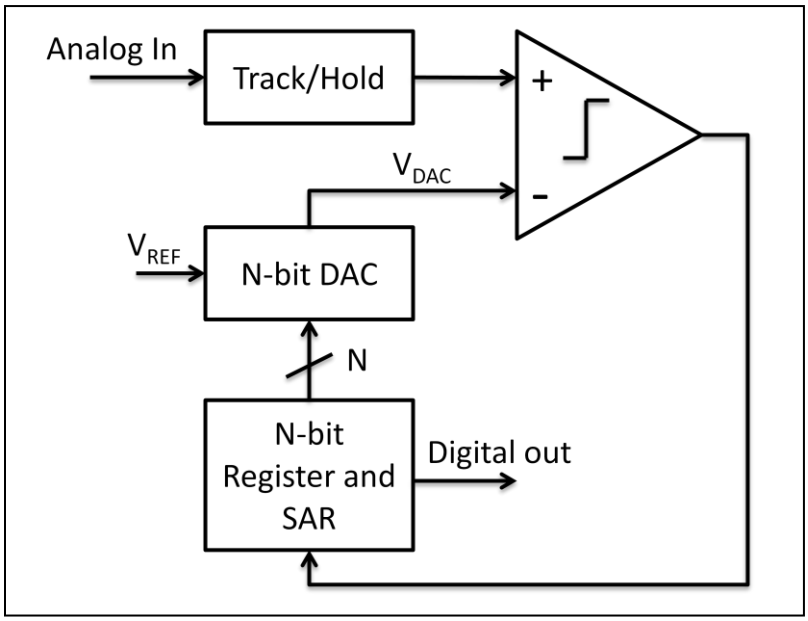

Figure 5: The basic structure of the SAR ADC

The SAR ADC has the advantages of lower power consumption, high resolution and high accuracy, especially when compared to the flash ADC. While the low power consumption is highly desirable for our application, the SAR ADC does have some distinct disadvantages. The most pressing disadvantage is that they require an extremely accurate DAC and comparator. These two devices must be extremely precise to ensure that the bits are recorded correctly. Noise in the system could affect the final outcome. In addition, the SAR ADC requires trimming or calibration to reach higher resolution.

The sigma-delta ADC is another common converter used for low power applications. The main advantages of the sigma-delta ADC are its high resolution and accurate conversions. The sigma-delta ADC has an advantage over the flash ADC because it has no resistor chain, instead taking incremental samples to do its conversion, which makes it more accurate. In addition, it does not use comparators so its size does not grow as quickly as a flash ADC when an increase in resolution is desired. They also have advantage over SAR ADCs as they include filters to make anti-aliasing and noise less of a problem without the need for high accuracy components. Sigma-delta ADCs do, however, lack in performance. The converter relies on oversampling the input to do its conversion, so the operation takes several clock cycles to complete. Furthermore, the back end of the ADC includes decimation which lowers the sampling rate. This makes the sigma-delta impractical for high frequency applications. Because the half-life of the biological samples is so long, however, low frequency is acceptable and 1MHz is more than fast enough to detect changes. For these reasons, the sigma-delta ADC was chosen for the final design.

The sigma-delta ADC consists of two parts, a modulator and decimator [19] [20]. The modulator takes in an analog signal and converts it into a digital bit stream that represents an average of the input signal. It uses two techniques to accomplish this goal, oversampling and

quantization [21]. Oversampling involves sampling a signal with a frequency that is much higher than the highest frequency of the incoming signal. This process helps prevent aliasing, which occurs when two different signals become indistinguishable. In addition, oversampling will improve resolution and reduce noise.

Quantization is the process of mapping the continuous input signal to a finite set of discrete levels. This introduces error and has an inherent loss of resolution compared to the analog signal, but is necessary to make the conversion to digital. The error added to the system from quantization is called quantization error, and because it affects the processing system in a similar manner as white noise, it can be treated as such. The error is often called quantization noise and will be filtered through noise shaping. A noise transfer function can be included that is typically high-pass or band-stop, filtering the noise so that most of its power lies at high frequency, outside the signal band.

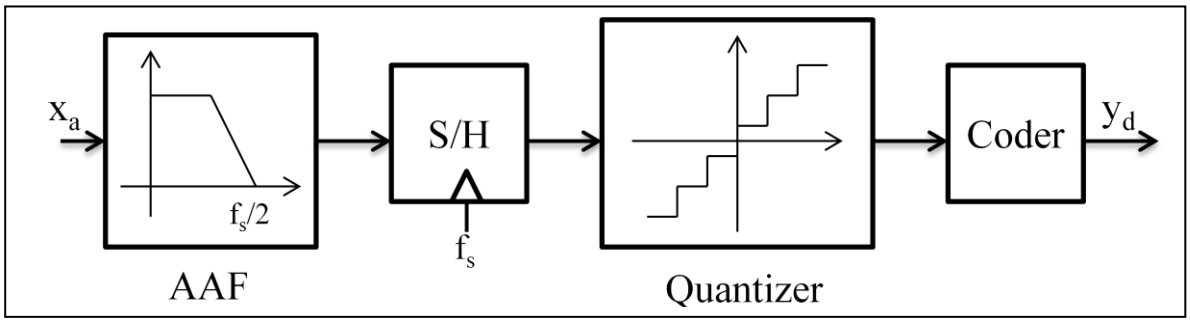

Figure 6: A basic block-level diagram explaining the workings of the sigma-delta modulator

A simplistic model of a sigma-delta modulator can be seen in Figure 6. The analog signal, x_a, is first passed through an anti-aliasing filter (AAF), which will prevent the high frequency components of the signal from being aliased into the signal bandwidth. The cutoff frequency is $f_s/2$, where f_s is the sampling frequency. The resulting signal is sampled by the sampling-and-hold (S/H) circuit, which will yield a discrete-time signal. The quantizer will then

map the continuous range of amplitudes into discrete levels. The final signal, y_d, is produced by the coder which assigns a unique binary number to each level, thus providing a bit-stream of ones and zeros to represent the incoming analog signal. An example analog sine wave passing through the modulator is modeled in Figure 7. This shows the anti-aliased signal, x_{aaf}, the signal sampled by the S/H, x_s, the quantized signal, x_q, and the final output, y_d. The coder will output all ones at the positive peak of the sine wave, and all zeroes at the negative peak. In between there will be varying amounts to represent the slope of the sine wave.

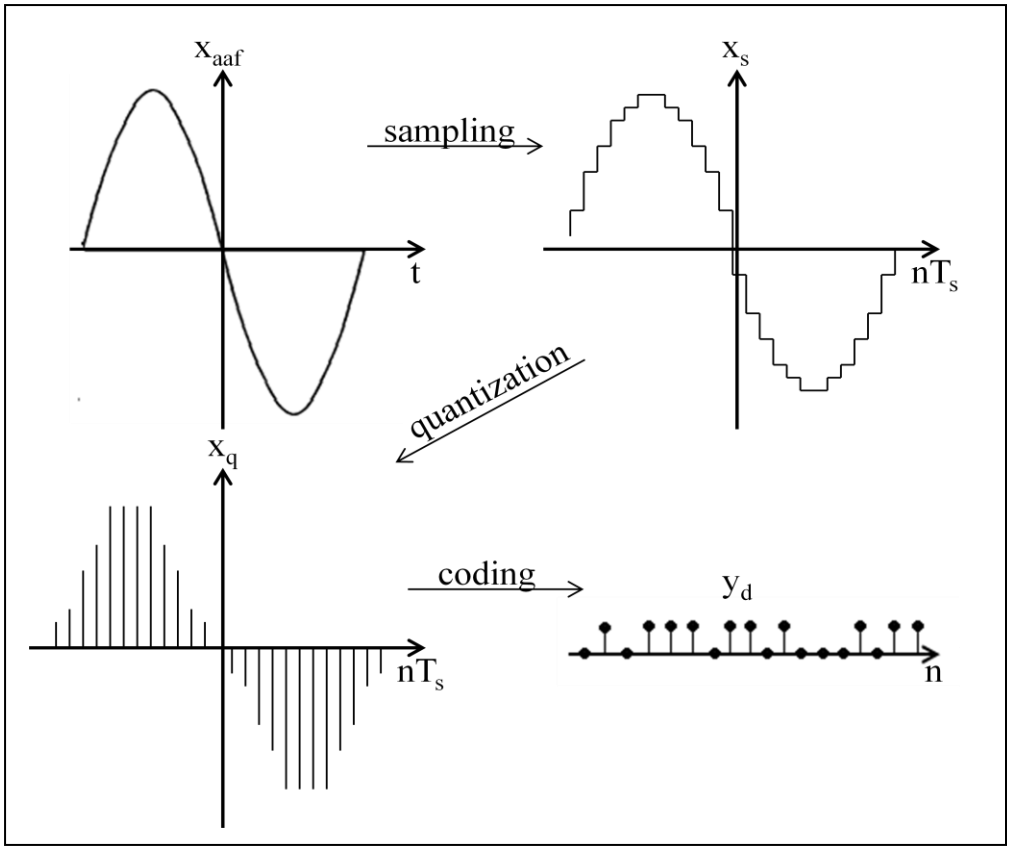

Figure 7: An example analog signal passing through the different stages of a sigma-delta modulator. The highest amount of ones will be coded at the positive peak of analog sine wave and the highest amount of zeroes will be coded at the negative peak.

After the signal passes through the modulator, it will be passed to the decimator to be filtered and decimated for a final digital output with a word length N. Part of the decimator's job is to act as a low pass filter, removing the quantization noise that is contained in the higher

frequencies. The cascaded integrator comb (CIC) filter is often used for decimation because of its simplicity. In order to create a basic CIC filter, one needs only an adder, a subtractor and a few registers to act as storage elements. The CIC's most basic form consists of two stages, one integrator and one comb stage as seen in Figure 8. Higher order filters can be achieved by increasing the number of integrator and comb stages. For example, a second order filter would have two integrator stages and two comb stages.

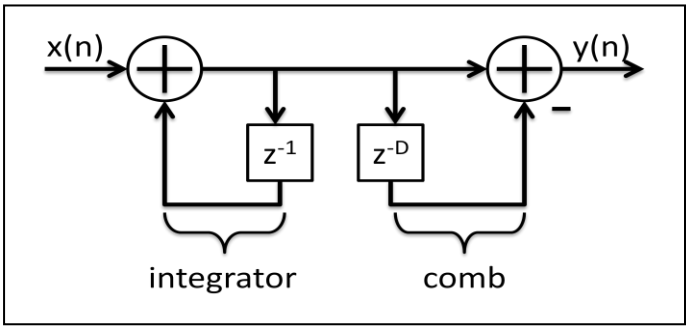

Figure 8: A simple first order CIC filter with an integrator stage followed by a comb stage

The CIC input x(n) is the bit serial stream created by the modulator. This signal is delayed and recursively added to the incoming signal each clock cycle. The comb stage subtracts the differentially delayed output of the integrator from the original output. The integrator functions as a single pole filter with unity feedback [22]. The transfer function for a single integrator is:

$$H_I(z) = \frac{1}{1-z^{-1}} \qquad (1)$$

The comb section includes a differential delay, D, which will control the filter's frequency response. The transfer function for a single comb is:

$$H_C(z) = 1 - z^{-M} \qquad (2)$$

where M equals D, divided by decimation rate R. For a multi-order CIC filter, each of the transfer functions will be raised to the order, N, power and multiplied together for a final CIC transfer function of

$$H(z) = \left(\frac{1-z^{-M}}{1-z^{-1}}\right)^N \qquad (3)$$

Equation (3) can be used to find the frequency response given $z = e^{j2\pi f}$. After some derivation, the final response can be seen as

$$H(z) = \left[e^{-j2\pi f(M-1)/2}\,\frac{\sin(\pi fM)}{\sin(\pi f)}\right]^N \qquad (4)$$

Ignoring the phase factor in this equation, the frequency response can be approximated by a sin(x)/x or sinc function, as illustrated in Figure 9. Different values of M will adjust the position of the first lobe. The higher the value of M, the better the low pass filter.

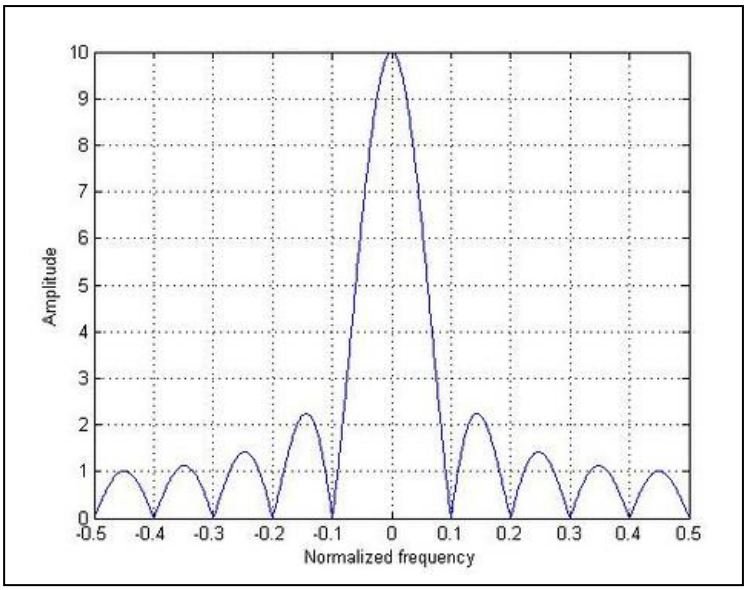

Figure 9: The shape of a CIC frequency response, similar to that of a sinc function

There are several options for CIC decimator designs and two commonly discussed variations are a bit-parallel and bit-serial design. The bit-parallel design is typically used in

applications where power consumption is a secondary concern and high speed is required. Conversely, the bit-serial design is slower, usually with decreased sampling speed, but it can greatly reduce the power consumption. For biomedical applications, sacrificing speed for power is desirable since the speed reduction, possibly down to the 100kHz range, does not adversely affect sampling of biological reactions which take place on the scale of seconds.

The bit-parallel design is most often used in high performance applications. In a bit-parallel design, a 10-bit word can be processed in one clock cycle. The same can be said for a 15-bit word. The main drawback of this design is that it requires large busses and large components to process data. For example, a 10-bit decimator requires the use of an adder and for bit-parallel it would require a 10-bit adder. Increasing the bits to 15 would then increase the adder to a 15-bit adder, adding to the power consumption and the complexity of the routing. A bit-serial decimator, on the other hand, will require a 1-bit adder regardless of the number of bits being processed. A visual comparison of two adders, one bit-serial and one bit-parallel, can be seen in Figure 10.

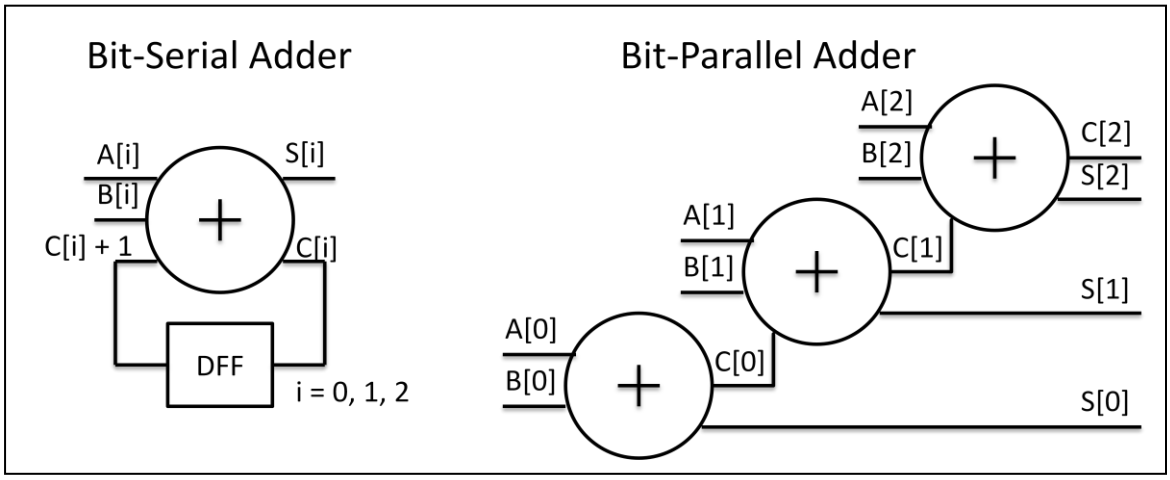

Figure 10: A comparison between a bit-serial and bit-parallel adder

Choosing between the two CIC designs will depend on the application. The bit-parallel architecture will often be chosen for high performance and bit-serial for low power and smaller area. There are some cases where a change to bit-serial will actually increase the power consumption [23]. In this design, the bit-parallel consumed 315μW and the bit-serial consumed more than 20 times this at 6399μW. However, this particular design was used in a high performance application so the bit-serial CIC filter had an increased operating frequency that was 25 times the frequency of the bit-parallel version, causing the significant increase in power consumption. In addition, the bit-serial design can sometimes require large registers for storing data which could potentially increase the size. This increase can be rather small, however, when compared to an increase in adder size.

While this increase in power consumption is possible, typically a switch to bit-serial will reduce the power consumption. For example, if the operating frequency is not increased and the sampling frequency is subsequently decreased, there will be a significant drop in power consumption [24]. In this paper, a bit-serial decimator was directly compared to a bit-parallel decimator with no changes in operating frequency. The bit-parallel design consumed 33.28mW compared to 15.12mW consumed by the bit-serial design. Another way to reduce the size and power consumption is to switch to a smaller process [25]. A direct comparison between these designs can be seen in Table 2.

Table 2: Power consumption comparison for previous designs

Serial/Parallel	Total Power	Total Area or Gates per Bit	Clock Frequency	Process Used
Parallel [23]	315µW	0.0018 mm^2/bit	6.144MHz	0.18µm CMOS
Serial [23]	6399 µW	0.0013 mm^2/bit	153.6MHz	0.18µm CMOS
Parallel [26]	2.67mW	0.0055 mm^2/bit	16.8GHz	0.18µm CMOS
Parallel [24]	33.28mW	312 gates/bit	200MHz	0.6µm CMOS
Serial [24]	15.12mW	267 gates/bit	200MHz	0.6µm CMOS
Parallel [25]	85mW	0.2 mm^2/bit	10MHz	0.8µm CMOS

As is shown, a direct comparison across papers is difficult since each uses different clock speeds and processes. However, [23] shows that bit-serial is not well suited for high performance applications. On the other hand [24] shows that it is possible to reduce power consumption by switching to bit-serial and reducing the overall area of the design. The last entry in the table shows simply how a larger process size will also lead to larger power consumption as well as area.

CHAPTER 3: INTEGRATION OF BIOSENSOR CHIP AND ELECTRODE ARRAY

The biosensor chip consists of an array of 128x64 electrodes, a potentiostat, TIA, main amplifier, differential to single ended converter and 128 read channels. The electrodes are designed in pairs with an "F" shape as shown in Figure 11. The goal of the shape is to increase the perimeter of the electrode because a majority of the current produced comes from mass transport at the edges of the electrode. The current density increases as the ratio of the perimeter length to electrode area increases [27]. The actual electrode area is comprised only of metal layer 4, which is outlined in Figure 11 in red. Each square and each gap is 2.5µm long. The total area for one electrode is 68.75µm^2 and the perimeter is 60µm.

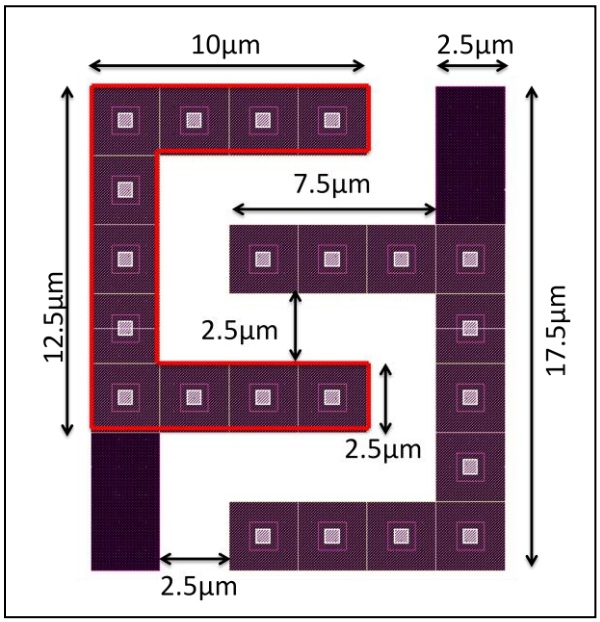

Figure 11: A pair of working electrodes designed with an "F" shape.

The 128x64 array measures approximately 2200x2200µm for an area of 4.84mm^2. This amounts to 8,192 electrodes total. However, due to limitations in space and routing only one small sub-array will be active at any given time. This smaller sub-array consists of 128 working electrodes as well as one reference and counter electrode. The sub-array measures approximately 194x230µm, or 0.0446mm^2, and can be seen in Figure 12. The 128 working electrodes are

represented in the middle as 64 pairs in the shape outlined in Figure 11. To each side of the working electrodes are two large rectangles that represent CE and the RE which will be connected to the potentiostat.

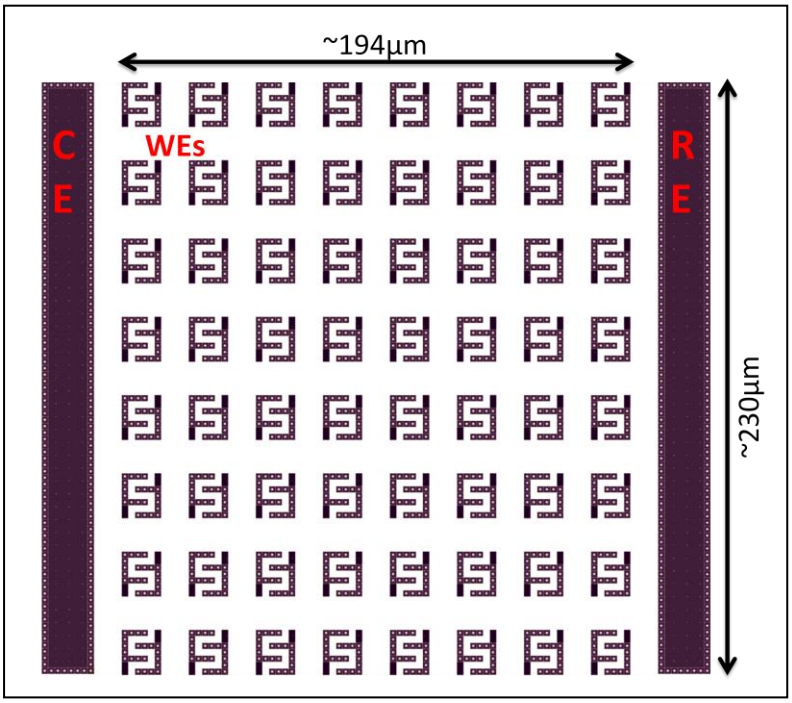

Figure 12: A single sub-array of 128 working electrodes, 1 reference and 1 counter electrode

As only 128 electrodes will be on at any given time, there are a total of 128 read channels corresponding to 128 TIAs, main amps and differential to single ended converters. This has advantages not only in routing and spacing, but also in reducing the amount of electronics needed on chip. In addition, since the electronics can be switched at high frequencies, the operator can set up the circuit to switch through every sub-array before the half-life of NO has expired. A good sample time for a single sub-array will be about 500µs and with this constraint it will take 32ms to switch through all 64 sub-arrays. Therefore, each sub-array can observe the tissue 6 to 18 times, depending on how long it takes the NO to dissipate. A basic breakdown of the electronics chain as it connects to the array is shown in Figure 13.

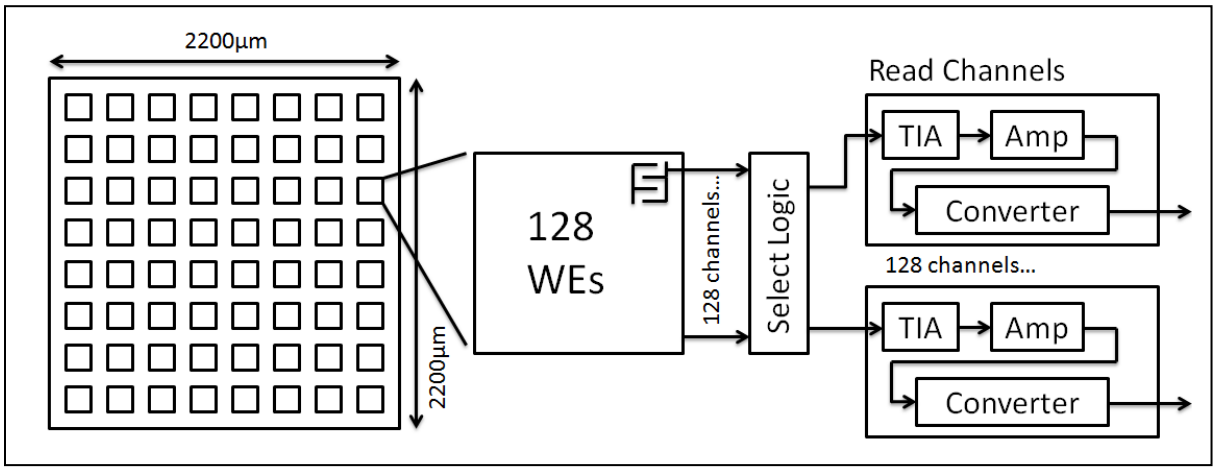

Figure 13: The entire 128x64 and its dimensions and how each sub-array of 128 Wes connects through select logic to 128 TIAs, amplifiers, and differential to single-ended converters.

Each electrode in each sub-array is connected to a switch that will be turned on or off by a 3:8 decoder. The switches are then connected out to one of the 128 TIAs, as outlined in Figure 14. Each of the electrode pairs in the figure represents one 128 electrode sub-array, of which there are 64 total. Although there are separate switches for each electrode in a pair, both the vertical and horizontal switches are controlled by the same decoder so that all 128 electrodes of a single sub-array will be on at the same time.

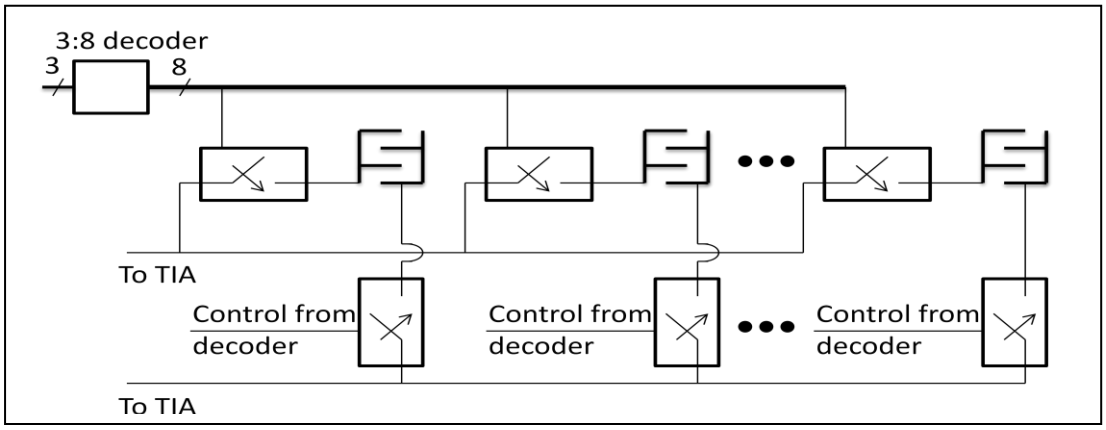

Figure 14: An illustration of the control block for the electrode sub-arrays.

The TIAs, main amps and converters are arranged 32 apiece on each side of the 128x64 electrode array, as seen in Figure 15. The final outputs from the converters are each routed to external pins. In addition to this, the electrodes, as they are connected through the switches, are

routed out to their own external pins. This is done so that the electrodes can be used with future iterations of the electronics without having to remanufacture the electrodes themselves.

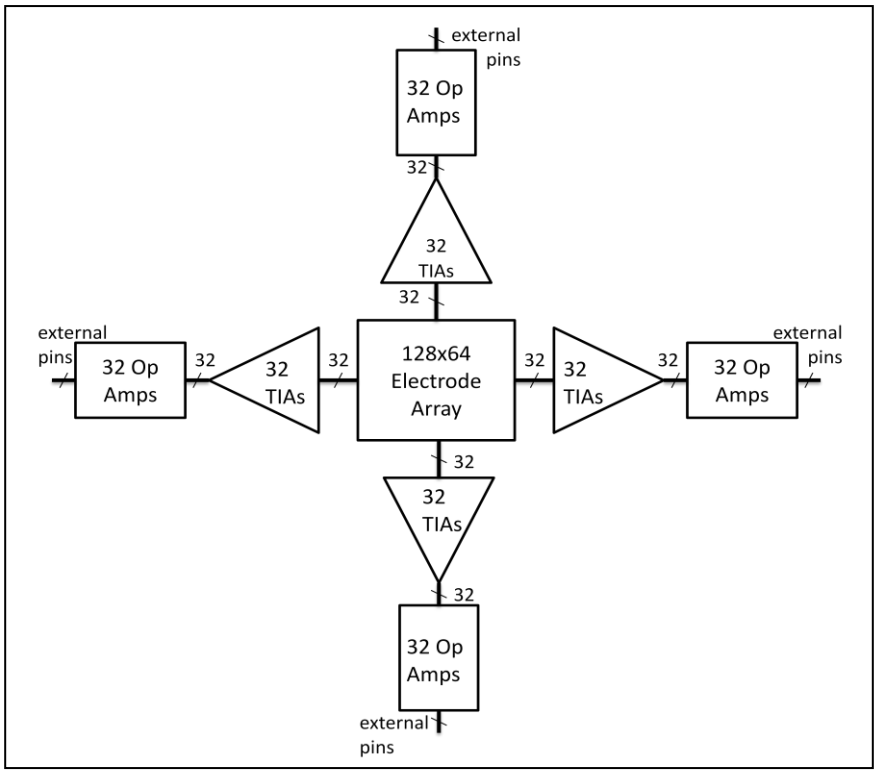

Figure 15: A top level diagram of the entire array

Although each sub-array has its own reference and counter electrode, only one potentiostat was needed for the entire design. The main goal of the potentiostat is to maintain a constant potential between the working electrodes and a RE while supplying current to a CE. A bias voltage is applied to the RE and the corresponding induced current will describe the concentration of the chemical present at the working electrode. Each of the counter and reference electrodes found in the sub-arrays is connected together so that only one potentiostat is required to control the entire array. The potentiostat is designed to operate up to 10,000V/s for cyclic voltammetry and in the range of -1V to +1V for amperometry. The single op-amp structure used for the potentiostat is shown in Figure 16 [28].

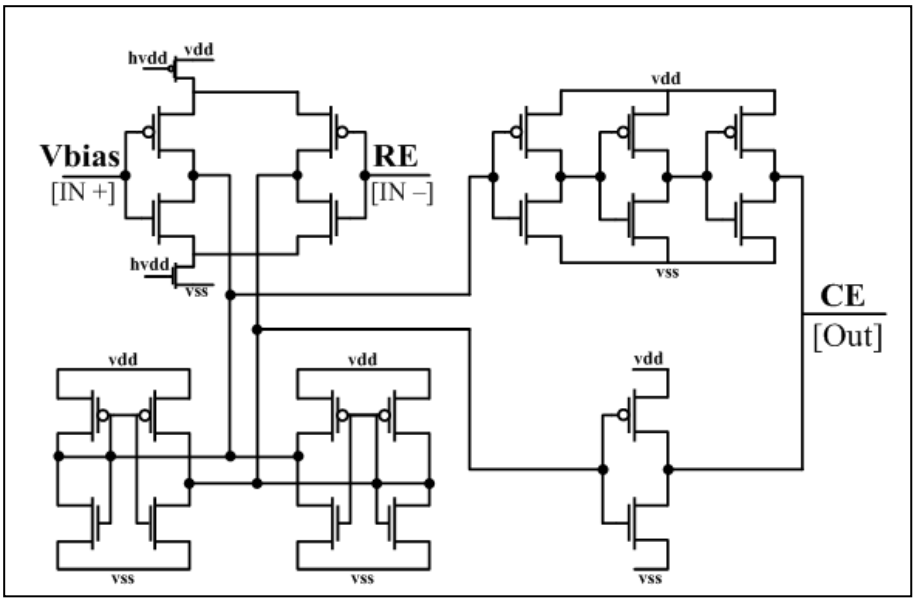

Figure 16: A self-biased inverter based potentiostat circuit

The potentiostat, along with the TIA, main amp and converter, was designed and implemented by other members of the design team and their basic structure is outlined as follows. The potentiostat requires that there be low offset between the two inputs, Vbias and RE, and high open loop gain. The input offset is defined as the difference in DC voltage between IN+ and IN-. The closer the offset is to zero, the more precise the potential held at the electrodes. The high open loop gain is required because we want a wide sweeping range, allowing the operator to set a Vbias between ±1V. In order to accomplish these goals, the potentiostat was designed as a two stage amplification circuit. The first stage was made with large transistors to minimize the offset seen at the input. The second stage increases the open loop gain.

The second electronic component in the chain, the TIA, was designed with novel features to reduce power consumption and noise while increasing signal sensitivity. These features include the use of negative load impedance to improve DC gain and reduce systematic error, and inverter-based amplifiers designed to improve common-mode noise rejection. The final TIA design is a more complex version of the switched-capacitor model, which was outlined in

Chapter 2, section 2, and seen in Figure 3, including a differential op-amp and sophisticated signal sampling to minimize signal and sampling noise. The final design can be seen in Figure 17 [29] [30]. Each of the switches in the form of a single or pair of transistors is controlled by a signal from the non-overlapping clock generator. The signals ϕ1 and ϕ2 are non-overlapping as are ϕ1_d and ϕ2_d. The ϕ1_d and ϕ2_d signals are equivalent to the ϕ1 and ϕ2 signals, but are delayed by approximately 3ns. A current mirror is included to change the current received from the WE to a differential current.

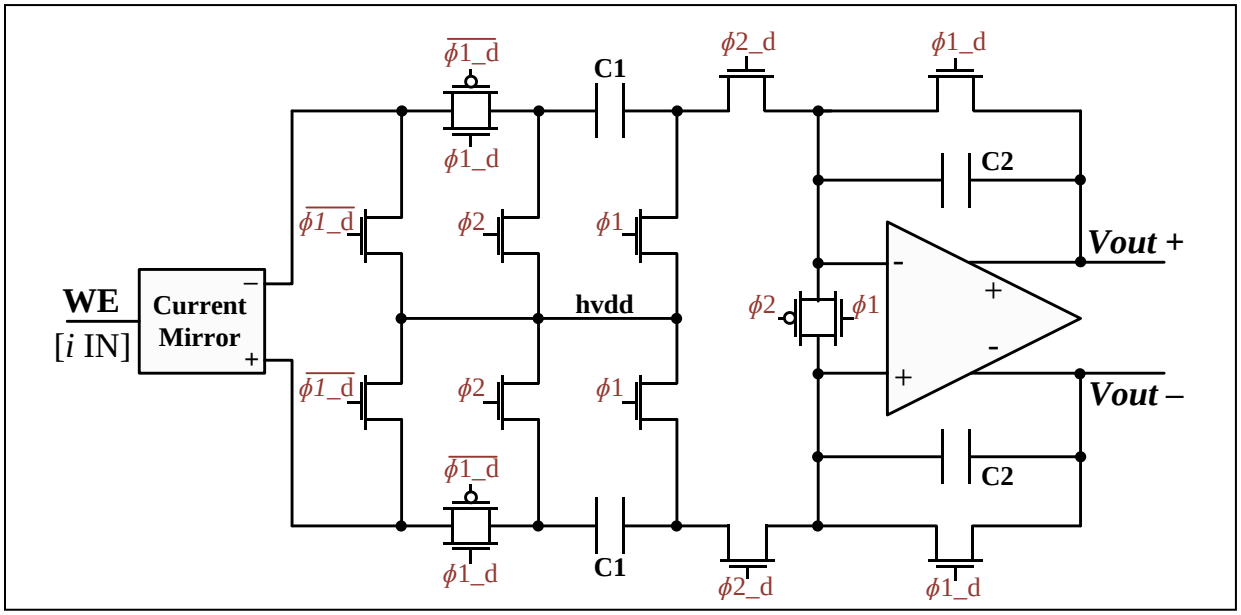

Figure 17: The final differential TIA design with switched capacitor and current mirror

At first it is difficult to see the equivalent switches and what happens when a switch is opened or closed. To better illustrate what happens in the circuit, one can observe the circuit for each phase, as seen in Figure 18. Seeing the circuit in this way, with the different switches closed or open at the appropriate time, it is easy to see that C2 is a switched capacitor equivalent to that seen in Figure 3. In the ϕ1 phase, C1 is charged by the incoming current. In the ϕ2 phase, C1 discharges onto C2.

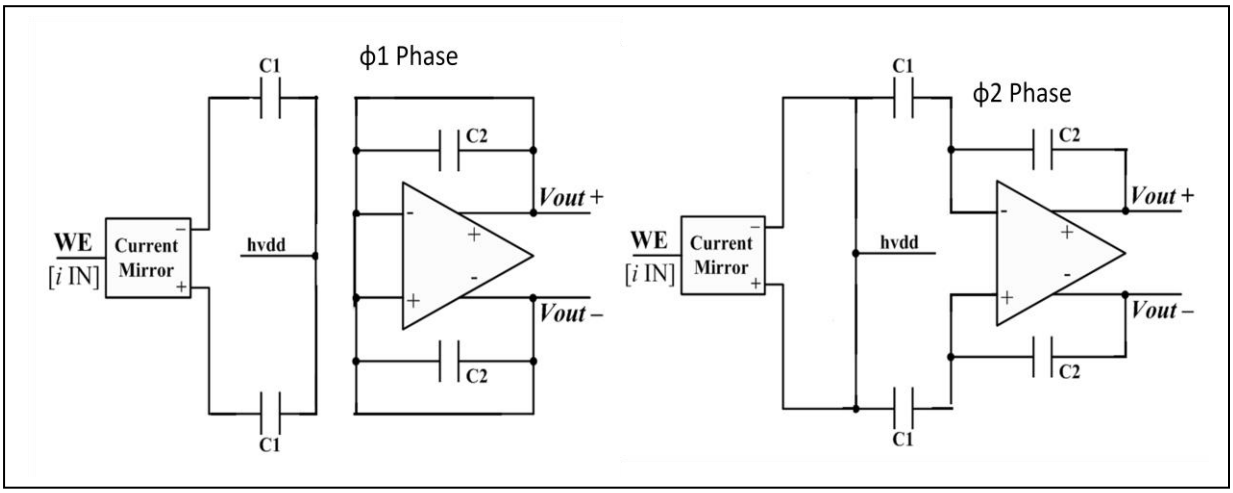

Figure 18: The TIA during its two phases, illustrating how C2 acts as the switched capacitor.

Once the signal has passed through the TIA, it will be sent to the main amp for amplification. The outline for the basic main amp structure can be seen in Figure 19. For this figure I have replaced the transmission gates with switch symbols since it is a larger design. Again, this design uses switched capacitors to implement the amplification, this time receiving a voltage input from the TIA. Not pictured is the common feedback, represented in the figure as Vcmfb. The common mode feedback is made of a switched capacitor bank that is connected to the ideal common mode, ground. The voltage in a differential circuit can have a tendency to drift to the rail because of variations and offsets [31]. The feedback loop allows for correction of the drift.

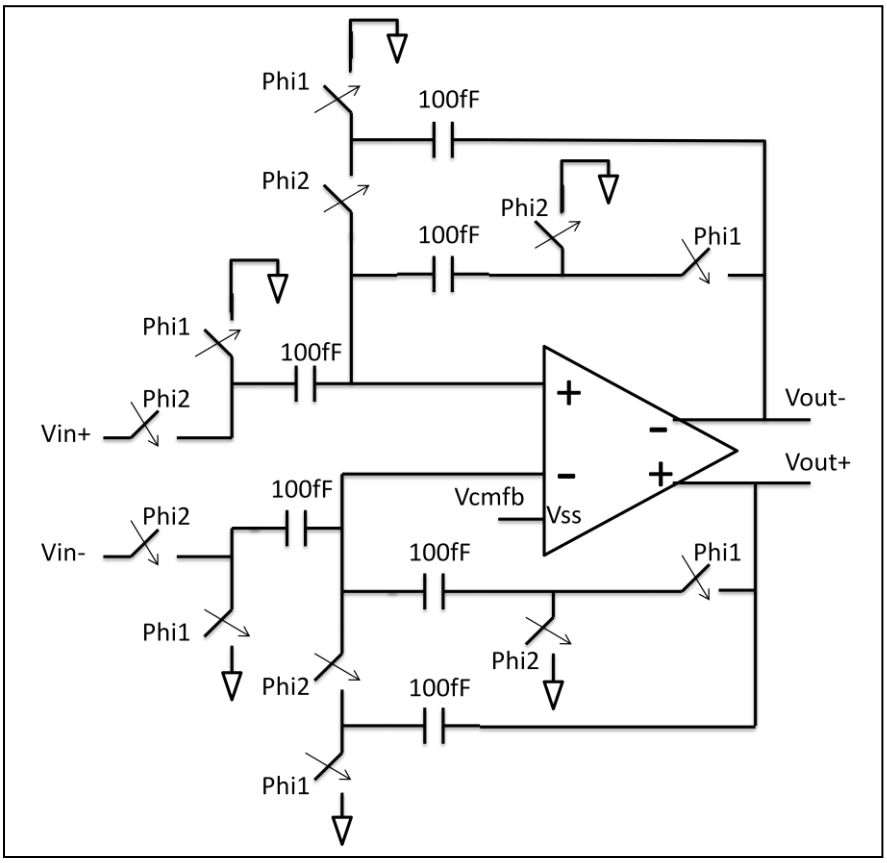

Figure 19: The main amp, which employs switched capacitors and common mode feedback to amplify the incoming TIA signal.

Finally, the signal will pass through a differential to single ended converter. The differential signals are used for better rejection of noise and power supply variations. However, due to a limited number of pins, we needed single ended signals to be routed out. To accomplish this, the two differential signals are fed through switches and a buffer op amp to create a single ended signal as shown in Figure 20. The switches are set up in such a way that that Vin+ will be connected to the output during the phi2 phase and Vin- will be connected to the output during the phi1 phase. This final output of Vout is routed to the pins and is the only signal in the electronics chain that can be observed besides the input to the TIA.

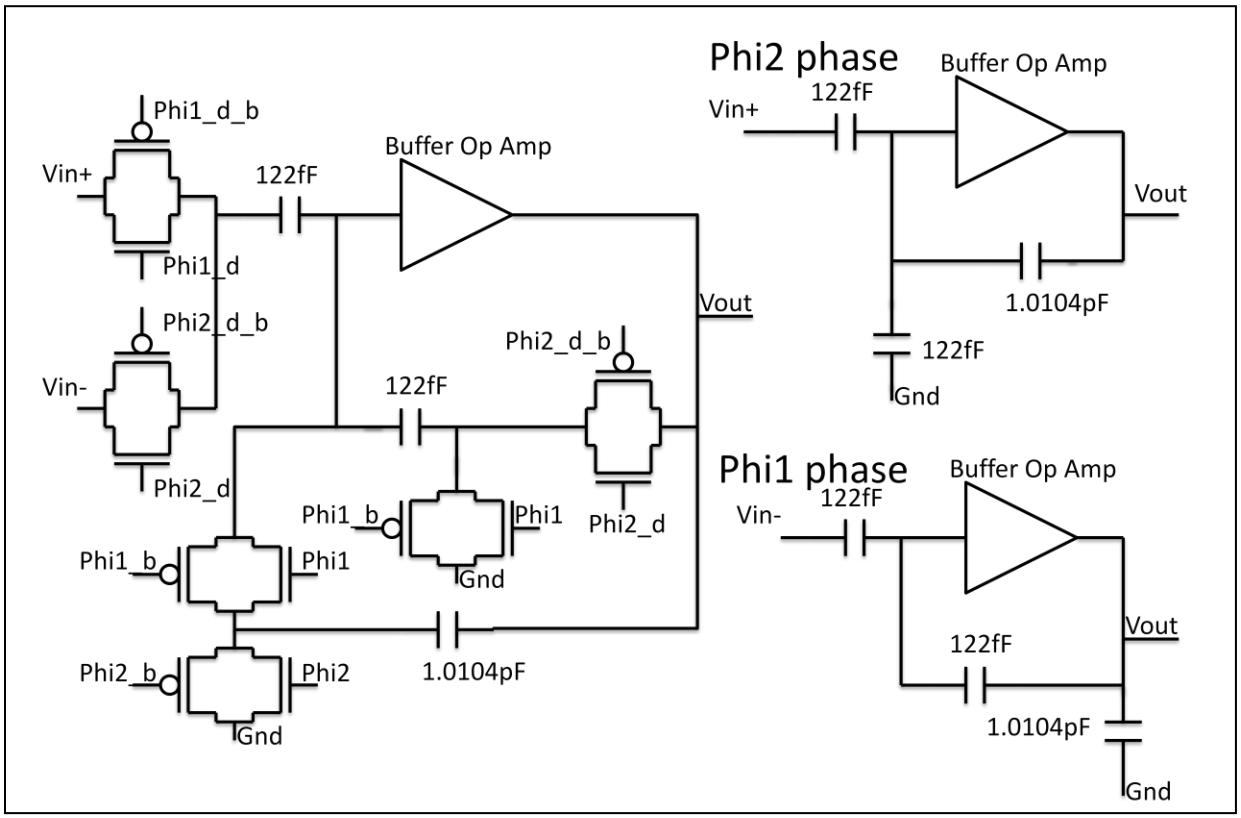

Figure 20: The converter circuit shown with switches and in different phases.

The clock speed for this design is 1MHz and the TIA and main amplifier pair takes approximately 20μs to settle. For each sub-array, we will allow 500 clock cycles for measurement to ensure that the amplifiers have settled and that we have an adequate amount of data. Thus each sub-array will be on for 500μs each with an effective frame rate of 30 frames/sec.

CHAPTER 4: DECIMATOR DESIGN USING BIT-SERIAL ARCHITECTURE

The overall decimator design is one part of a larger integrated sensor chip. Since this chip is intended for biomedical applications, one of the main goals is to achieve low power. In addition to this, bandwidth of biomedical signals is low, allowing us to use a 0.18µm process with no loss in performance. In order to reduce the overall size and power consumption of the decimator, we chose a bit-serial architecture as opposed to the usual bit-parallel since, in our case, reduced performance is not an issue.

4.1 TOP LEVEL DESIGN: THE CIC

For this decimator design, we chose the CIC filter for its simplicity and known behavior. In a typical CIC decimator filter one finds one integrator and one comb for each order of the filter. In order to improve the anti-aliasing of the decimator, we used a third order CIC. If we made the order of the filter any higher than this, it would begin to get expensive to implement with less return. The decimator receives its output from the modulator output which produces a word length of 10 bits. When fed to the decimator, this word length is sign extended to 15 bits in order to avoid overflow. Some overflow still occurs, but it only affects the output for two or three output cycles before the circuit corrects itself.

A basic outline of how the modulator and decimator work together is shown in Figure 21. The final output of the decimator is a 15 bit binary word that can be translated into decimal to form the digital representation of the original analog sine wave input. Eventually this final signal will be sent through addition digital signal processing if needed, and then transmitted wirelessly to a computer for data analysis. The final binary output from the decimator and its translation into decimal can be observed in Appendix A.

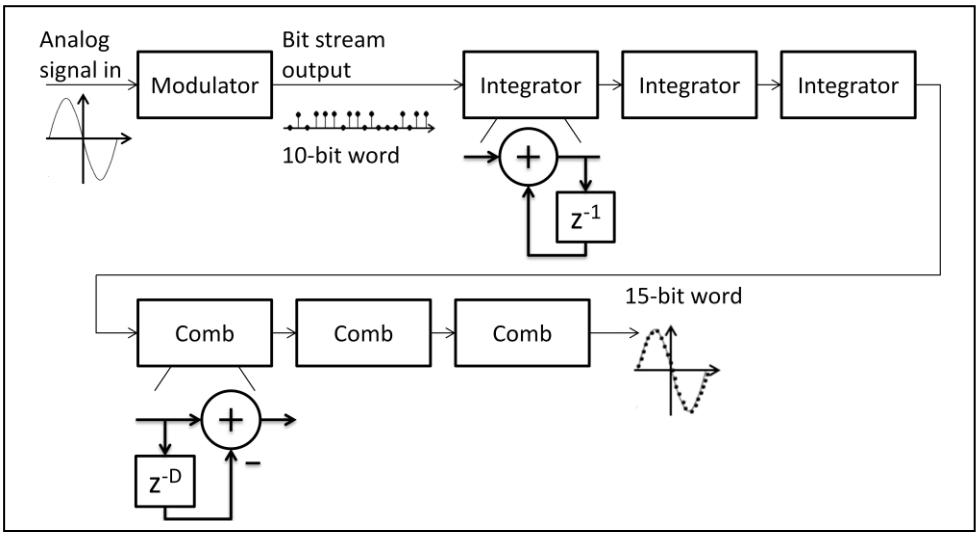

Figure 21: The modulator creates a bit stream output based on a given analog input. The CIC is then made of three integrators and three combs that produce a final digital output.

The amount of differential delay for the comb section, D, was determined through MATLAB simulation. The first simulation run was of the frequency response of the circuit using equation 3. As was mentioned in Chapter 2, this response will resemble a sinc function. To determine approximately how wide or narrow the low-pass filter would be for different delays, the frequency response was run with D values of 15, 30 and 45. As Figure 22 shows, as the delay increases, the frequency response narrows, allowing for better filtering.

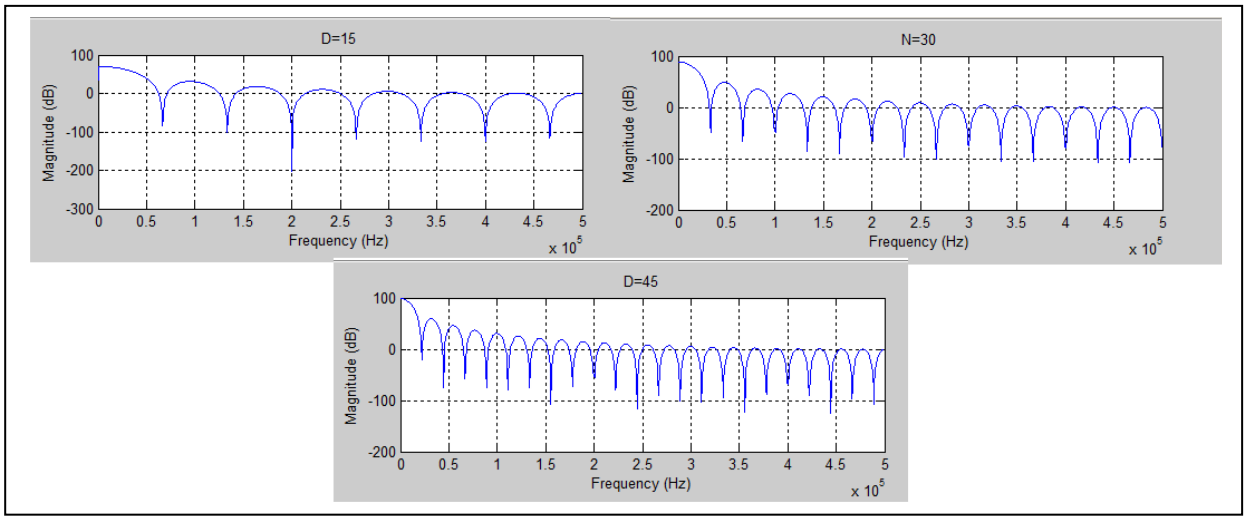

Figure 22: The frequency response of the decimator given different differential delays

In order to see how the filtering affects the final output, an ideal third-order CIC filter was implemented in MATLAB with several different differential delays. We fed an analog sine wave into the modulator and saved the digitized output stream. This output was then fed into the MATLAB code, which can be found in Appendix B. Equations 1-3 outline the basic structure for creating the code to simulate the design in MATLAB. Given delays of 15 and 30, there was observed distortion in the peaks of the final digital sine wave, as can be seen in Figure 23. To completely remove the distortion seen at the output, we needed to narrow the low pass range of the filter by increasing the differential delay to 45. The final sine wave given D = 45 can be seen in Figure 24. Though a delay 45 increases the number of needed components, we wanted to ensure there was no distortion on the output and D = 45 was used. For our 15-bit word count we had an R = 15, so our final comb delay was 3 and required a 45-bit shift register for each comb section.

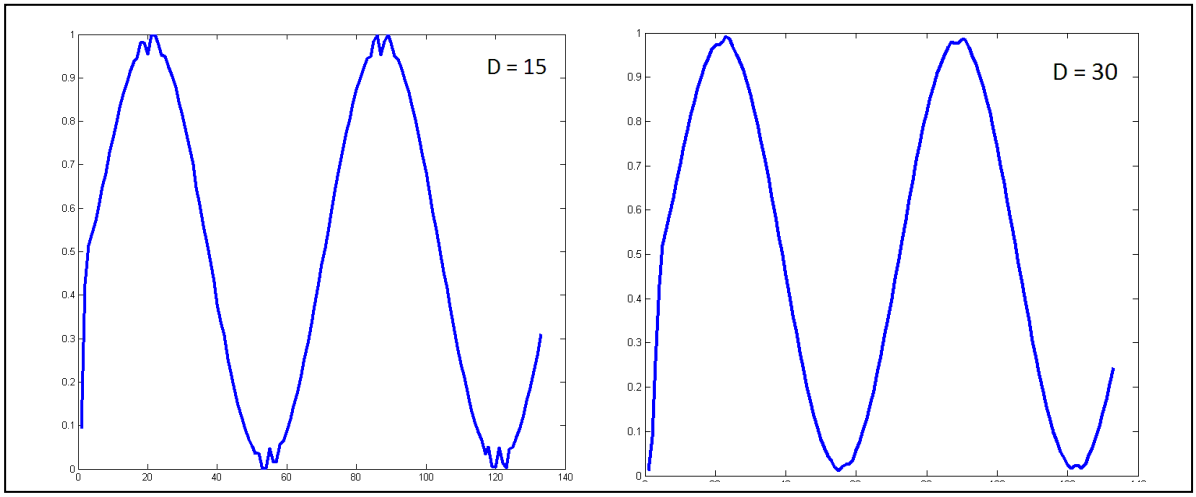

Figure 23: A comparison of two differential delays that can be used for the CIC filter showing extreme distortion at D=15 and slight distortion at D=30

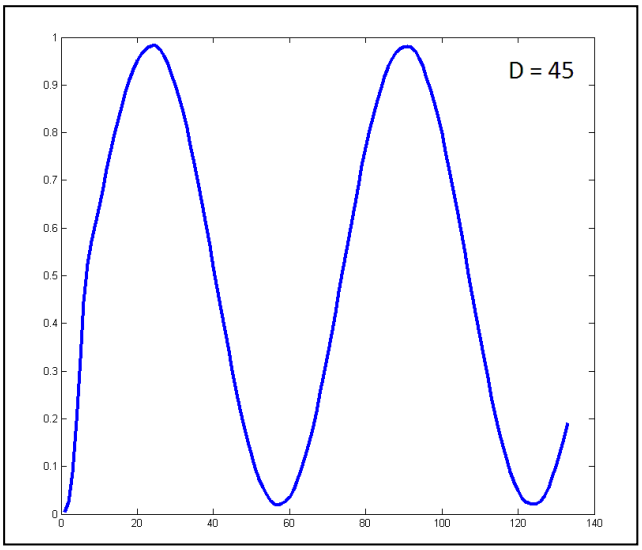

Figure 24: The final output sine wave for the chosen differential delay of 45

The final setup for each section of the CIC decimator can be seen in Figure 25. Since the decimator is bit-serial, a 15-bit shift register is used to delay the entire 15-bit word once. In a bit-parallel system, each bit would pass through at the same time and be delayed by one clock cycle, which would require wide buses as well as larger adders. In the bit-serial system, each bit gets stored in the 15-bit register and added one bit at a time to the input from the modulator. This results in a slower sampling speed for the decimator as the data will be valid every 15 clock cycles as opposed to every clock cycle.

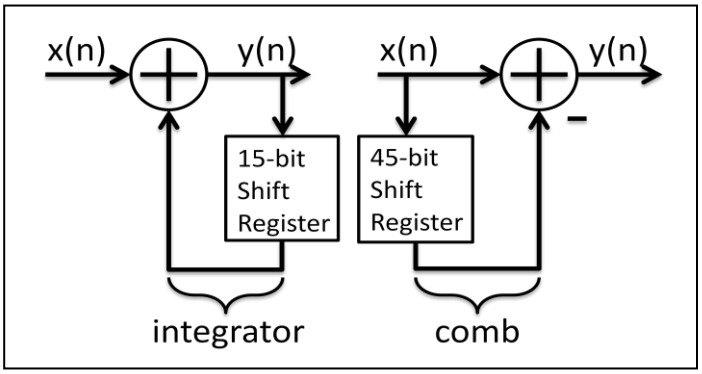

Figure 25: Single integrator and comb used for a CIC filter. A third order filter will have three of each connected in series.

The comb section seen in the second part of Figure 25 has a 45-bit shift register. This will delay the 15-bit word three times before it is subtracted from the incoming signal. Although the data is only valid every 15 clock cycles, the design will run at the same clock speed as the modulator. Since the clock speed has not been increased, we will not see an increase in power consumption as was observed in other works [23]. The entire top level block diagram for the third order CIC filter can be seen in Figure 26. The final block in the chain is a serial in, parallel out (SIPO) converter. This piece is included to change the serial output to a parallel output for the digital signal processing (DSP) block which was originally designed for parallel information. In the future the additional DSP may be changed to receive serial information so that the SIPO converter may not be necessary.

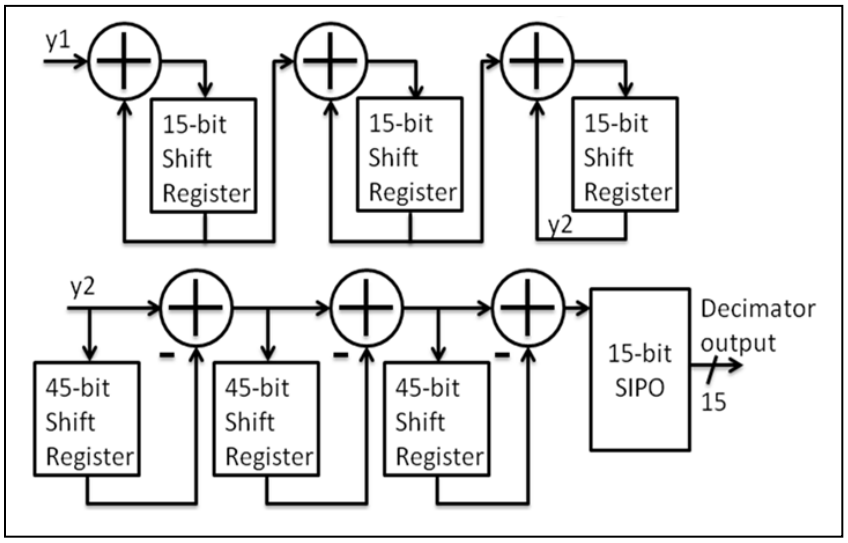

Figure 26: Top level diagram for a third order CIC decimation filter

4.2: FINITE STATE MACHINE AND CONTROL LOGIC

Since the original word length from the modulator is 10 bits, and the decimator works with a word length of 15 bits, the entire design required a few extra components to ensure that we were recording and filtering the right data from the modulator. The first of these elements

was a counter. The modulator creates a digitized version of an incoming signal. Thus a modulated sine wave will have all ones at its positive peak and all zeros at its negative peak with variable ones and zeros in between. The counter will record the number of ones it sees at each 10 bit interval. Then the counter will wait 5 clock cycles, reset and count the next 10 bits. The counter is forced to wait so that it doesn't get ahead of the rest of the circuit, which is operating in 15-bit increments. Because the counter must wait, some data from the modulator will be lost. However, since the data is being filtered by the CIC, enough precision will remain to create the final digital output. Finally, the output from the counter must go through a parallel in, serial out (PISO) shift register in order to feed a serial input to the main CIC decimator. A finite state machine (FSM) is included to create a signal that will force the counter to wait for 5 clock cycles while the rest of the circuit operates normally. The FSM accomplishes this by creating a control signal that is high for 10 clock cycles and low for 5 clock cycles. The clock for the counter is subsequently a combination of the regular 1MHz clock and the FSM control signal, as seen in Figure 27. All other devices besides the counter will operate at the 1MHz clock.

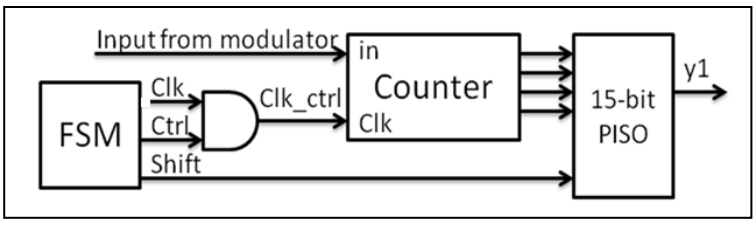

Figure 27: The FSM, counter and PISO setup used to interpret input from the modulator.

The FSM is made of a counter and simple combinational logic as seen in Figure 28. The Boolean equation for the control signal, which is high for 10 clock cycles and low for 5 clock cycles, is

$$Ctrl = [!Q4 + (!Q3 * !Q2)] * "shift" \qquad (5)$$

In this equation Q4 represents the most significant bit (MSB) of the FSM counter and Q1 is the least significant bit (LSB). This equation was derived using Table 3.

Table 3: Truth table for the Ctrl logic. Q4 through Q1 are the output from a counter.

Q4	Q3	Q2	Q1	Ctrl
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
Reset				0

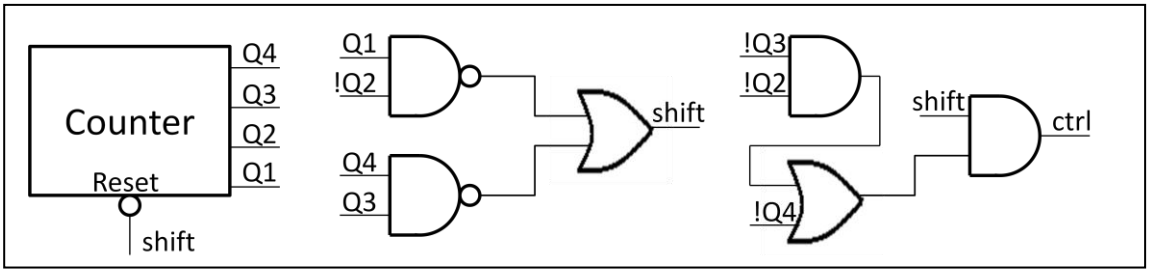

Figure 28: The inner workings of the FSM including counter and combinational logic for creation of the control and shift signals

Equation (5) also includes a shift signal which, as Figure 28 shows, is used as a reset for the counter. The shift signal is high for 14 clock cycles and low for 1 cycle ensuring that the counter will only count from zero to thirteen. Normally, the counter would count from zero to fifteen. However, the control and shift signals are based on a 15-bit cycle. Since the shift holds the counter in reset for one clock cycle, it is necessary to make the counter only count for 14 cycles, or from zero to thirteen, in order to produce the desired outputs. The Boolean equation for the shift signal is found using

$$"shift" = !(Q1 * !Q2) + !(Q3 * Q4) \quad (6)$$

Equation (6) makes a pre-shift signal low at count 13. The truth table for this equation can be found in Table 4. Therefore, the signal must pass through a DFF before it is sent out as the shift signal. This is done because the shift signal acts as a reset for the counter, which we only want to count to 13. The delay will allow shift to be high for 14 clock cycles while simultaneously resetting the counter at the 15th clock cycle, or what would be count 14. Because the shift signal is 1/15th the frequency of the regular clock it can also be used as the clock signal for sampling the decimator output. To achieve this goal, the shift signal can be inverted and its rising edge used as a clock.

Table 4: Truth table for the pre-shift and shift logic.

Q4	Q3	Q2	Q1	Pre-shift	Shift
0	0	0	0	1	1
0	0	0	1	1	1
0	0	1	0	1	1
0	0	1	1	1	1
0	1	0	0	1	1
0	1	0	1	1	1
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	0	1
Reset				1	0

In addition to acting as reset for the counter, the shift signal also changes the PISO from a parallel input to a serial input. This is done so that data is received by the PISO only when the counter output is valid. To accomplish this, the PISO's D-type flip flop (DFF) shift registers are connected to multiplexers (MUX) as is shown in Figure 29. When shift is low, the bottom input

to the MUX is active, and the data will be fed through serially. Shift goes high at clock cycle fifteen, switching the MUX input to the top. Thus, the incoming counter data will be stored in the DFFs to be passed through serially when shift goes low again. The PISO is a 10 bit shift register, with the 6 most significant bits (MSB) held at ground. When shift is low and the input is set to serial, the MSB of PISO is the only bit receiving outside input, while the rest of the PISO shift registers are passing along the information received in the parallel input stage. The PISO input S10 is connected to Q1, the LSB of the counter, so that the data is fed through LSB first.

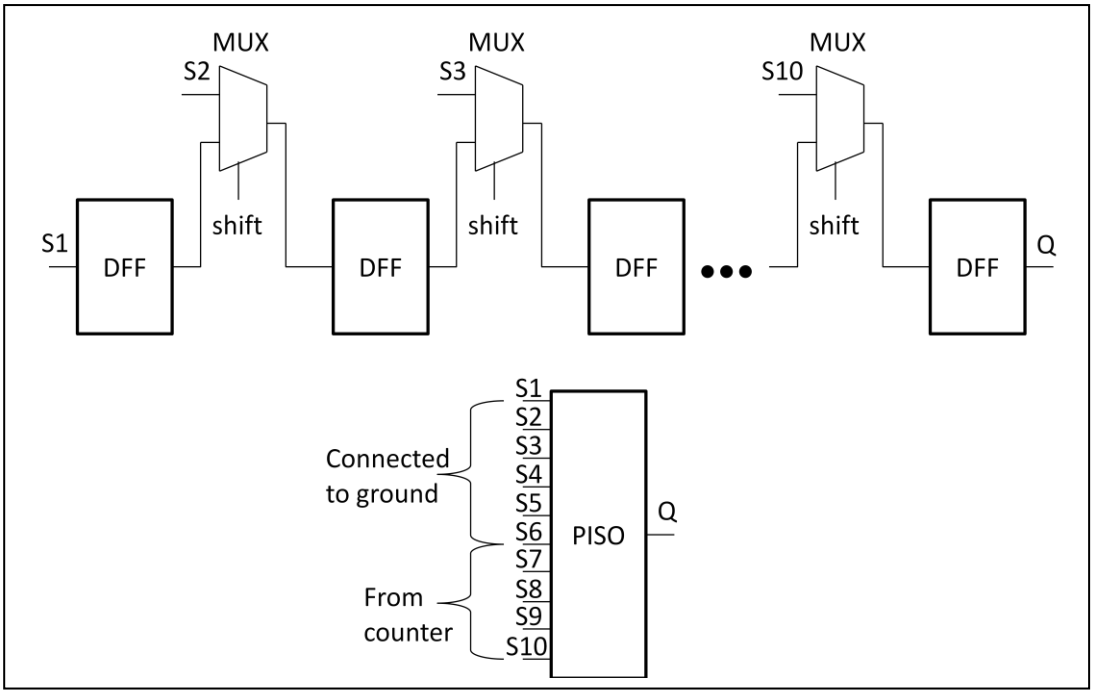

Figure 29: The inner workings of the PISO consisting of a chain of DFFs for storing the incoming data and MUXes to change the input from serial to parallel.

4.3: FLIP FLOP DESIGN

The key component in the design of the decimator is the DFF, as all of the storage and delay elements consist of this component. Originally, the design used a standard cell master-slave DFF that was supplied by National Semiconductor. The master section of the DFF stores the incoming value while the clock is low. During this period, the slave section does not change.

When the clock signal goes high, the slave section responds to the input from the master and the master's state will not change, making the DFF a positive edge flip flop. This design worked well and has a response that is well known for a variety of inputs. However, its layout was fairly large with an area of approximately 84.8μm^2. With a total of 250 DFFs taking up a majority of the design, a smaller DFF design would greatly reduce the overall decimator size. Originally, the entire decimator had an area 0.029mm^2, and 0.0212mm^2 of the area was from the DFFs. The size reduction is important not only for power consumption but also to conserve space on chip. Since there are potentially up to 128 channels being used and 128 corresponding decimators, the smaller it can be the better. Redesigning the DFF is simple and feasible since the inputs and loads for the DFFs are known. The original, standard DFF schematic can be seen in Figure 30. D represents the input, Q is the output, Qn is an inverted output, and Rn is the reset for the circuit.

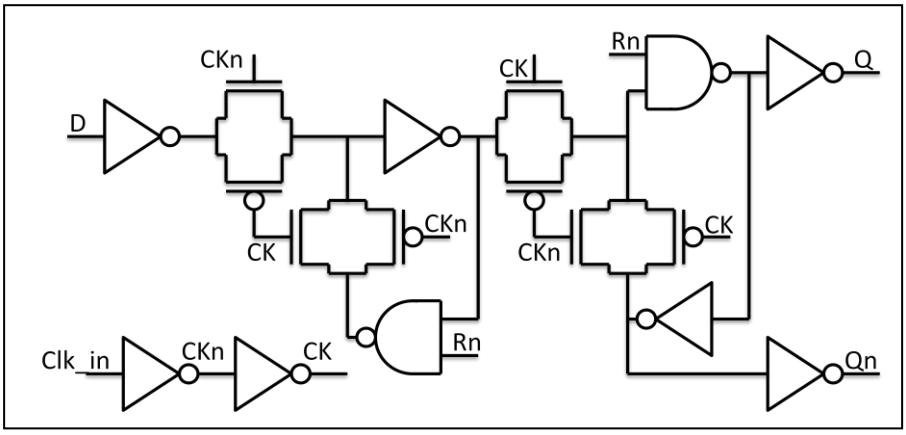

Figure 30: The original DFF design consisting of 28 transistors total

Each of the paired transistors in the circuit is a transmission gate. The first two transmission gates are the master section and the last two are the slave section. Inverter pairs are used as buffers throughout the design to ensure proper rise/fall times. NANDs are used in both the master and slave section to implement the reset. The NANDs are the component taking up the most area in this circuit as each NAND consists of 4 transistors, while the switches have only

2 as do the inverters. Because of this, the NANDs were the first part eliminated from the design. To implement the reset, a single transistor was included that pulls Qn to VDD, and subsequently Q to ground, when reset is low. The buffers are another component that takes up a fair amount of room in layout. They are useful precautions to ensure setup and hold times are met, but were unnecessary given a clock speed of 1MHz and a desired rise/fall time of less than 200ps. In addition, the feedback switches are helpful to ensure that the signal remains at the desired value, but were again unnecessary.

The original design, including two transistors per inverter and four per NAND, consisted of 28 transistors total. With the reductions outlined above, the number of transistors was reduced down to 11 as can be seen in Figure 31. The final rise/fall time for the new circuit measured 161.4ps, which was only 20.2ps slower than the original design and well within the required specification. This newer DFF design replaced all of the standard cell DFFs in the design except for the counter receiving input from the modulator. This counter required more precision to ensure that it recorded the correct output from the modulator. By passing through the counter, the signal has a regular and expected rise/fall time which the newer DFFs can handle without any problems. Using the standard cells for this single counter is not a problem, however, as the counter contains only 4 DFFs out of a total of 250, so the space lost to the larger DFFs is only a small portion of the overall decimator area.

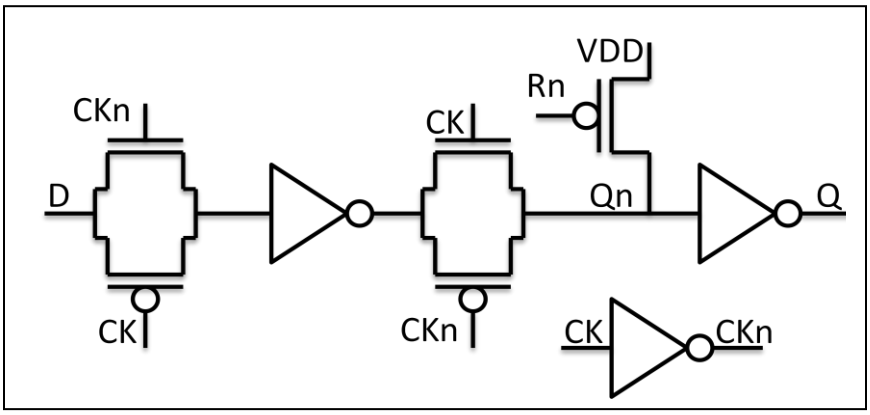

Figure 31: A reduced size DFF useful for low power applications

Despite the reduction in number of transistors by 60%, the final layout measured 38.544μm^2, a reduction in size of just over 50%. The size reduction was smaller than expected partly because the actual sizes of the individual transistors were slightly larger in the smaller design in order to ensure proper rise/fall times. In addition, the height and width of the cell had strict limitations so that it would still function as a standard cell. Overall, the decimator was reduced from 0.029mm^2 to 0.0158mm^2 through the use of the new DFF for a total reduction of 46%. The new DFF layout can be seen in Figure 32.

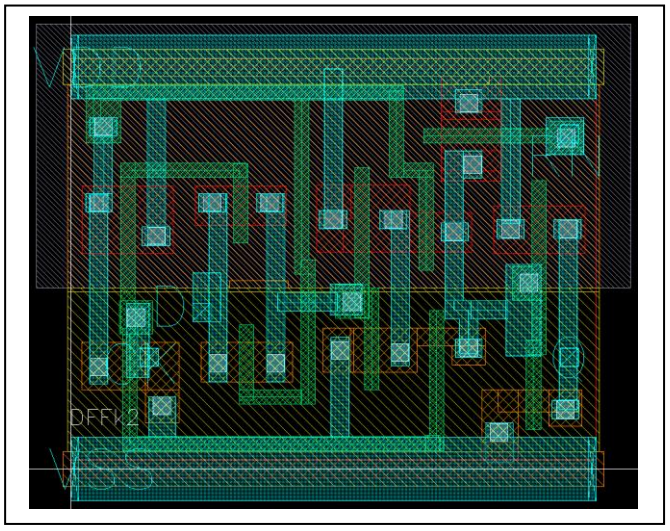
Figure 32: The layout for the smaller, redesigned DFF

4.4 TOP LEVEL CRITICAL PATH ANALYSIS

To ensure that the new DFF works, we need to measure both its setup and hold time and ensure that all signals in the decimator using the DFFs adhere to these times. The final setup time for the DFF was 300ps and the hold time was -100ps. The setup time means that any incoming signal, to be recorded as a 1 for example, must have made the transition from 0 to 1 300ps before the clock signal's rising edge. For the hold time, the signal must hold its value of 1 for at least -100ps to be recorded as a 1. Once these values were known, each critical path in the decimator was observed to see that the signals followed the appropriate times. Where they did not, buffers were added as needed to force the signals to meet both setup and hold times.

The path where the setup and hold times were the most critical was in the first integration stage. This stage is receiving data from the counter that is run on a different clock, the clk_ctrl, than the rest of the circuit. This was seen most pointedly at the first adder and the DFF for the carry out. Without any delays the carry out would go low a full 1ns before the DFF clock had a rising edge, as seen in Figure 33. The carry should have been counted, however, so this results in inaccurate addition. To correct for this timing issue, a buffer was added to delay the carry out so that it would meet the setup and hold times for the DFF.

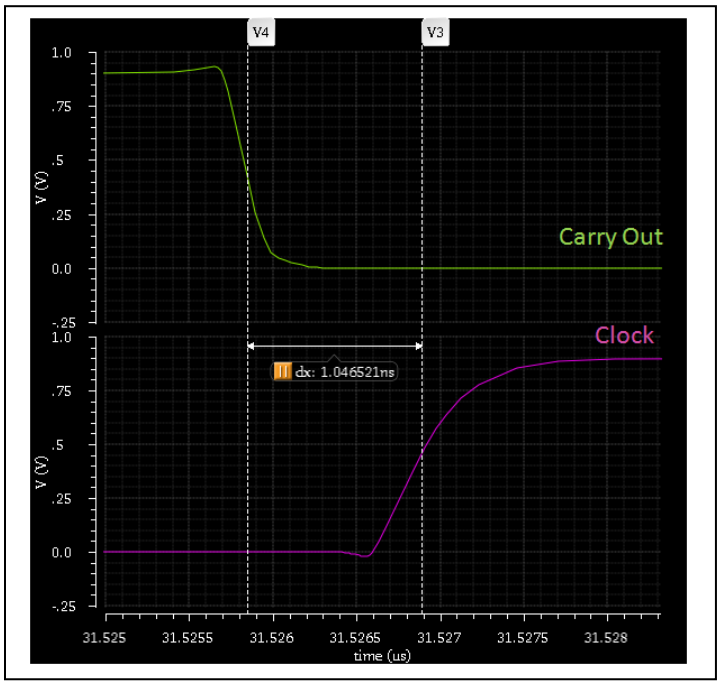

Figure 33: One example where the carry out does not meet hold time and is subsequently not recorded by the DFF

This was the only problem originally found in simulation when running the design in schematic. However, when implemented in layout, the added parasitics introduced additional timing problems. When the extracted version was simulated on a slow corner timing issues were discovered at the input to the first integrator as well as at the output of the combs. This was largely due to long clock lines and the inadequate buffers used for the clock at the 45-bit shift registers. Additional buffering was required in the clock tree to ensure accurately recorded results. Once these buffers were added, the timing requirements were adequately met and the decimator worked as expected. The final compensated carry out can be seen in Figure 34. Although the carry out signal is somewhat overcompensated with a hold time of 4.2ns, this is simply a result of the delay associated with the buffer and the extra hold time has no ill effects on the final outcome.

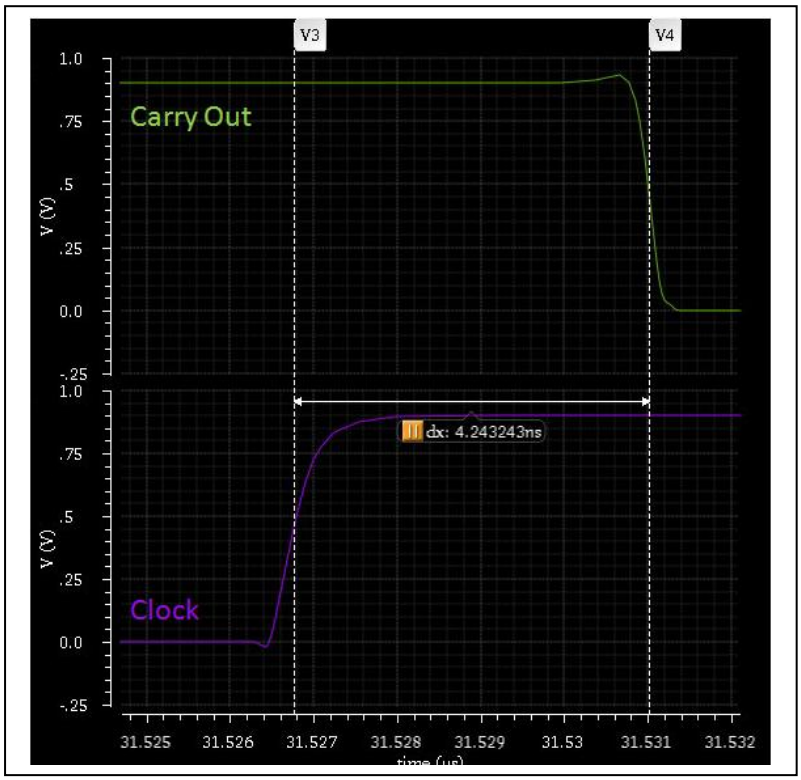

Figure 34: The same instance of the carry out shown in Figure 33, this time with an added buffer so that it meets hold time.

CHAPTER 5: SIMULATION AND SILICON RESULTS

In order to test the functionality of the two designs presented here, several simulations were run. For the decimator ideal simulations were run using MATLAB. The final design schematic and layout were extracted and simulated using Cadence. These results were then compared to the ideal. The decimator was then further tested after the design was fabricated in silicon. The biosensor array, on the other hand, has only schematic and layout simulation results, as the chip has not yet been fabricated.

5.1: BIOSENSOR SIMULATION RESULTS

The biosensor chip was simulated at the extracted layout level. With the routing included on the layout, the potentiostat had a load of approximately 25pF. The reference and counter electrodes were connected together for testing and the voltage at the working electrode was swept to test that the potentiostat could respond to a rapid change of 10,000V/s. The results can be seen in Figure 35. The bandwidth was also measured and the output was able to follow the 10,000V/s sweep with a maximum error of no more than 2mV in a ±1V range.

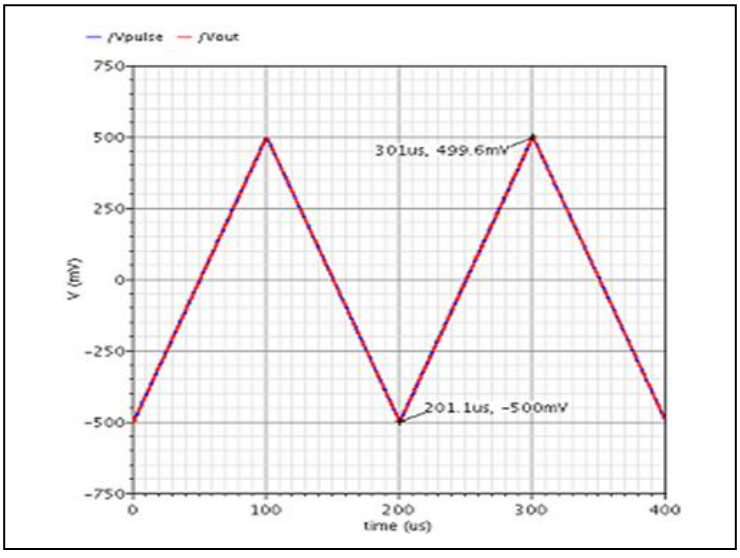

Figure 35: The input to the potentiostat compared to the output

To test the arrays as well as the TIAs, main amps and differential to single ended converters, 20 electrodes were randomly selected and connected to different current sources from 2nA to 100pA. If working correctly, when sampled, the output voltage from each of the TIA electronic chains should be linear. The settling time for the device is 15-20µs so each channel was sampled for 175µs to ensure sufficient time to settle and sample. Figure 36 shows the final simulation output for two different channels.

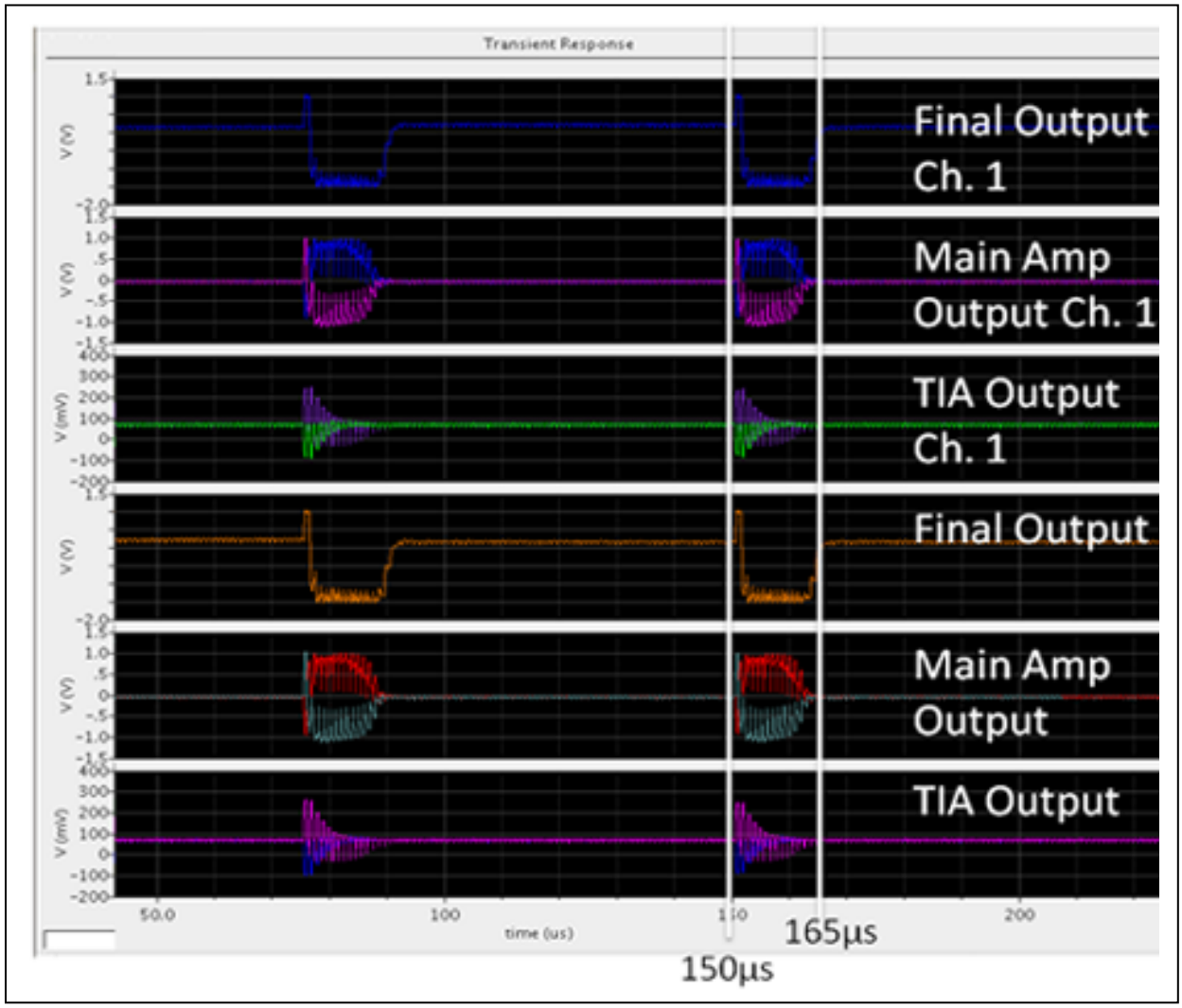

Figure 36: The biosensor simulation marking a read channel setting time of 15µs

The results for all 20 randomly selected electrodes were recorded after sufficient settling time had passed. The final result of current v. voltage can be seen in Figure 37. The current v.

voltage is fairly linear for the given currents. A second simulation was run for smaller currents. The output voltage could not be distinguished, however, from the noise at currents lower than 2nA.

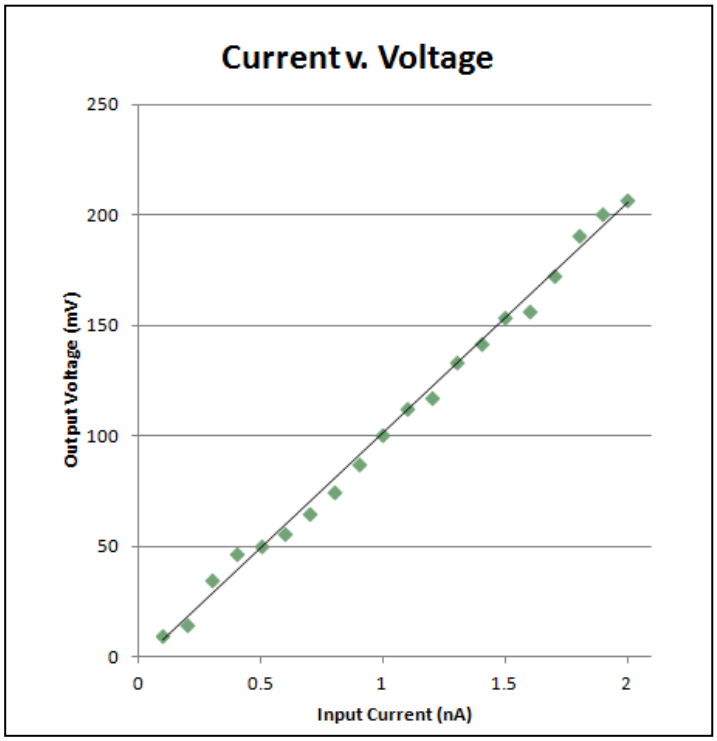

Figure 37: Sampled output voltages compared to a given current

Since the biosensor was also designed so that the electronics could be circumvented, we were able to easily test the electrodes and their switches independently from the TIA chain. To do this, we attached voltages from +1.5V to -1.5V to single electrodes in several sub-arrays. The outputs were then observed as the switches cascade through the sub-arrays. The final output for four electrode lines can be seen in Figure 38. The output looks like a staircase, stepping through each of the voltages. The only variance is near the middle when the voltages were purposefully stepped from 450mV to 400mV and then from 450mV to 400mV again. This was done because even with 50mV steps, there were not enough voltages for each electrode.

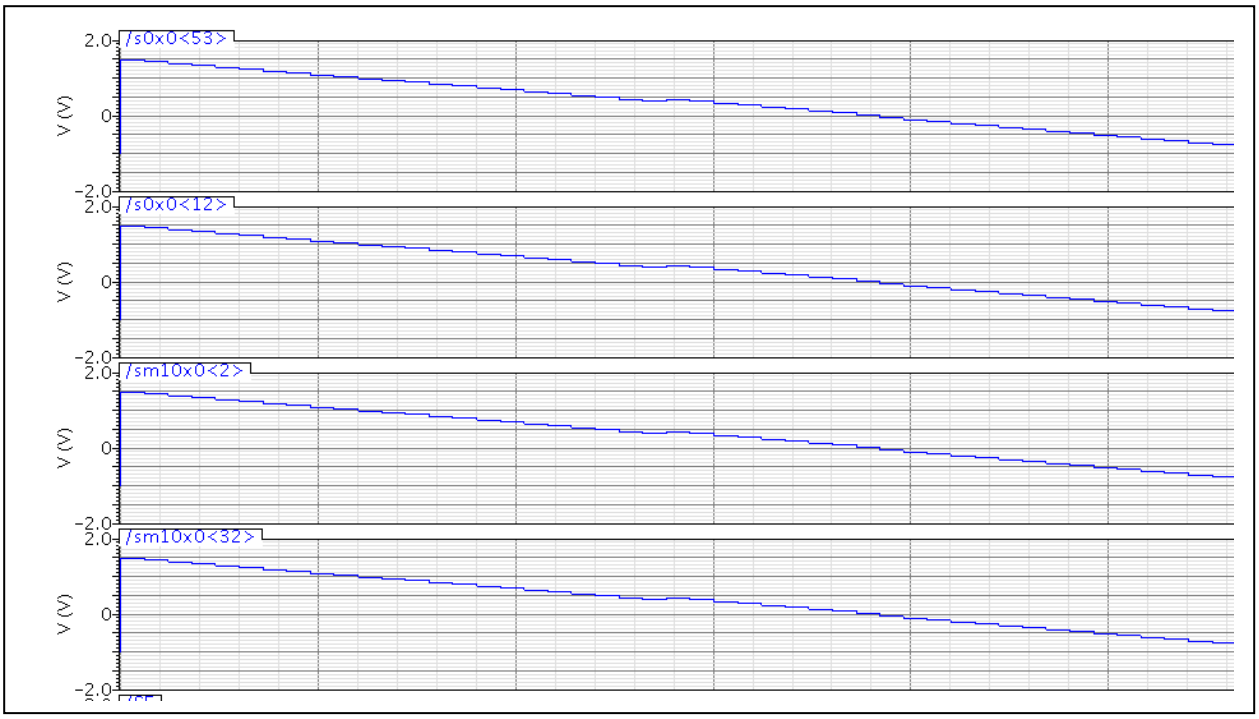

Figure 38: The output of electrode lines switching through several sub-arrays

The final simulation we ran was with the TIA electronics chain by itself, to make sure that it was working in layout. To do this we fed it a 1kHz sine wave current with an amplitude of 1nA and looked at the output of the differential to single ended converter. The final output from the differential to single ended converter was then observed. Figure 39 shows the final response to the input sine wave for one electronic chain not connected to the electrodes and switches, out101, and one that is connected, out100. In the simulation, the signals for switching through sub-arrays were still attached, so out100 had to settle each time it was connected to a new electrode, even though the input sine wave was directly connected to the TIA input.

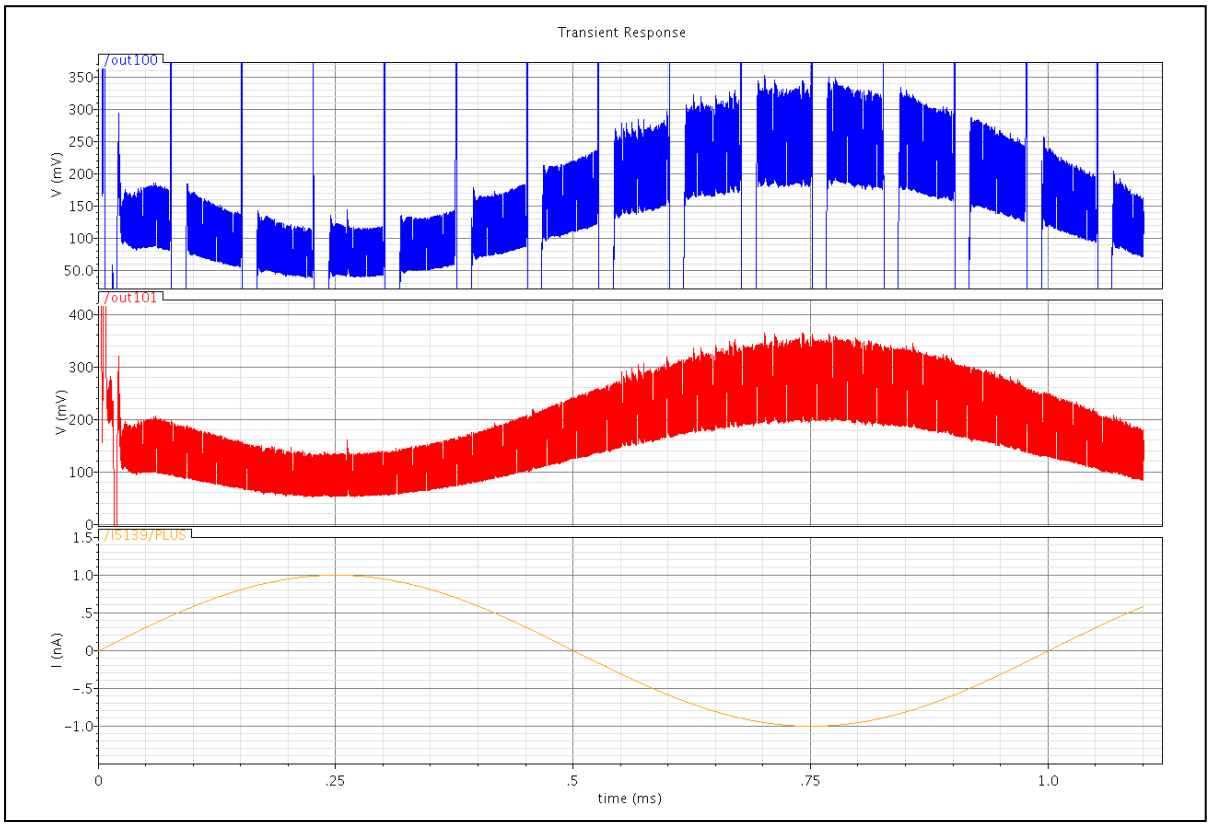

Figure 39: The final output with sine wave input with and without switching.

The output signals out100 and out101 appear thick because the TIA and main amp both rely on switched capacitors. For one half of a clock cycle, the circuits will be in a sampling phase. For the next half of the clock cycle the circuits will be in a hold phase. The output will be valid on the hold phase, as illustrated in Figure 40. The black vertical line marks the tail end of one of the hold phases.

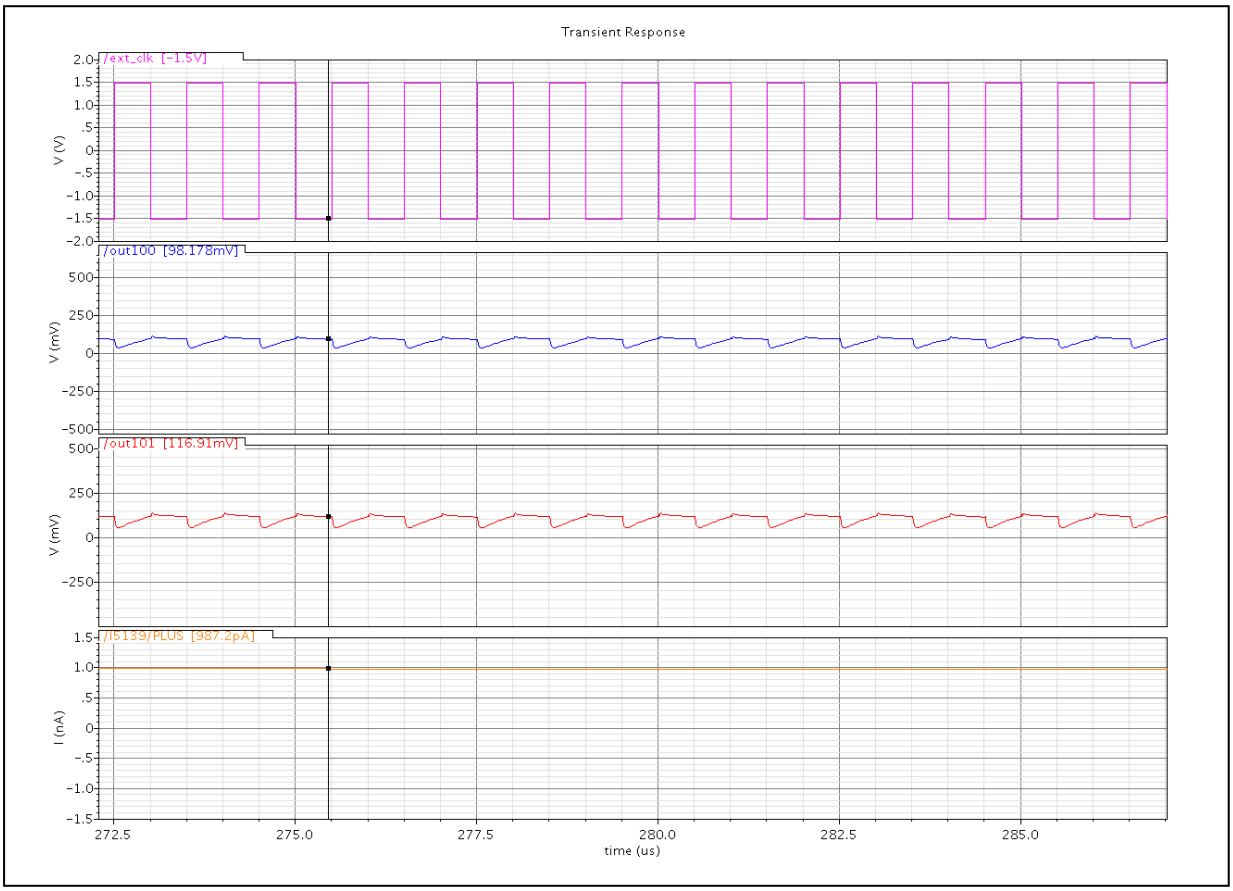

Figure 40: A zoomed in look at the final output showing several sample and hold phases.

5.2: PROPOSED BIOSENSOR SILICON STRATEGIES

Verification through simulation is sufficient prior to fabrication to be reasonably sure that the chip will work correctly. Once the chip is made, additional steps must be taken to ensure that there are no shorts between electrodes and that all electronics are working as expected. Thus, although the silicon fabrication of the biosensor chip has not yet finished, I have planned an outline on how to verify that it is working correctly.

The first step for verification is to test the electrodes themselves. The original test for the electrodes in simulation was shown in Figure 38. However, this test cannot be repeated in silicon. The test was originally done by connecting voltage sources directly to the electrodes and observing the output on the other side of the switches. In silicon, however, the electrodes are not

directly connected to the pads, but are connected via the switches to conserve both space and pads. Instead of this test, we can run a test to ensure that the electrodes are not shorted. Though the distance between electrodes is more than minimum spacing, it is still possible that one or more may be shorted. To test this we can apply a small voltage or current to an electrode pad and check pads for adjacent electrodes to ensure that they don't change at the same time. A basic flow chart for this process is displayed in Figure 41.

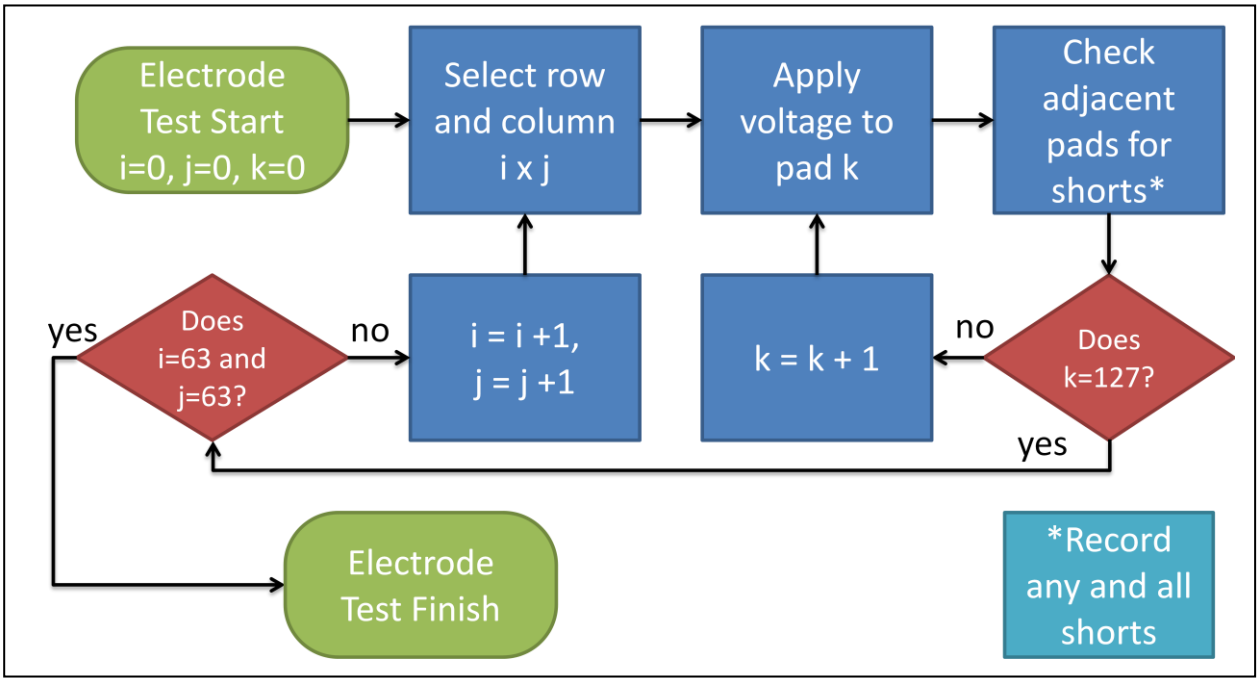

Figure 41: The flow chart for checking electrodes for shorts. If there is a short, the signal will show up on adjacent electrodes.

Once we have tested for electrode shorts, we can proceed with testing the electronics chain. The pads connected to the electrodes via the switch are also connected to the input of the TIA. This means we can use these pads to test the electronics chain without having to actually put a live sample on the electrodes. The one problem with this is that none of the equipment in the lab is capable of producing the small amount of current usually used to test the TIA in simulation. To get around this, we can apply Ohm's law to the situation. By using a very large

external resistor, such as 40MΩ, we can apply 40mV to the input and get a 1nA output. The only problem with this set up is that the resistor must be high precision so that we can get an accurate current out with little variation. Resistors used for previous TIA measurements were accurate to ±0.001%. The final results should be similar to the simulation results seen in Figure 37. The output will be read from the differential to single ended converter and should be linear.

Once it has been confirmed that all of the electronics are working as desired, the chip will then be ready for use. Our lab works in conjunction with a chemistry lab which will be able to test the chip further. They will test the electrodes and how they react with physical samples. The full versatility of the 8000+ electrode array can be tested and hopefully prove helpful in the study of NO. If, for some reason, some of the electronics appear to be working incorrectly, the electrodes can be wired externally to previously fabricated electronics.

5.3: DECIMATOR SIMULATION RESULTS

The decimator design was simulated at the extracted layout level. For a full sigma-delta ADC the modulator would provide the input for the decimator. A sample string of ones and zeros from the modulator was used to simulate the decimator. This string was created by feeding a 1kHz sine wave in to the modulator. The final output from decimator reads as a 15-bit word representing the input string in digital format. The final results can be seen in Figure 42. Each of the signals represents one stage from the CIC filter. S_out, S_out2 and S_out3 are the outputs from the first three integrator stages. S_out4, S_out5 and y15 are the last three outputs from the comb stages. Thus, y15 will be the final output from the decimator, valid every fifteenth clock cycle with a clock cycle of 1MHz. The decimator also includes a serial in, parallel out (SIPO) converter that will provide a parallel output as needed.

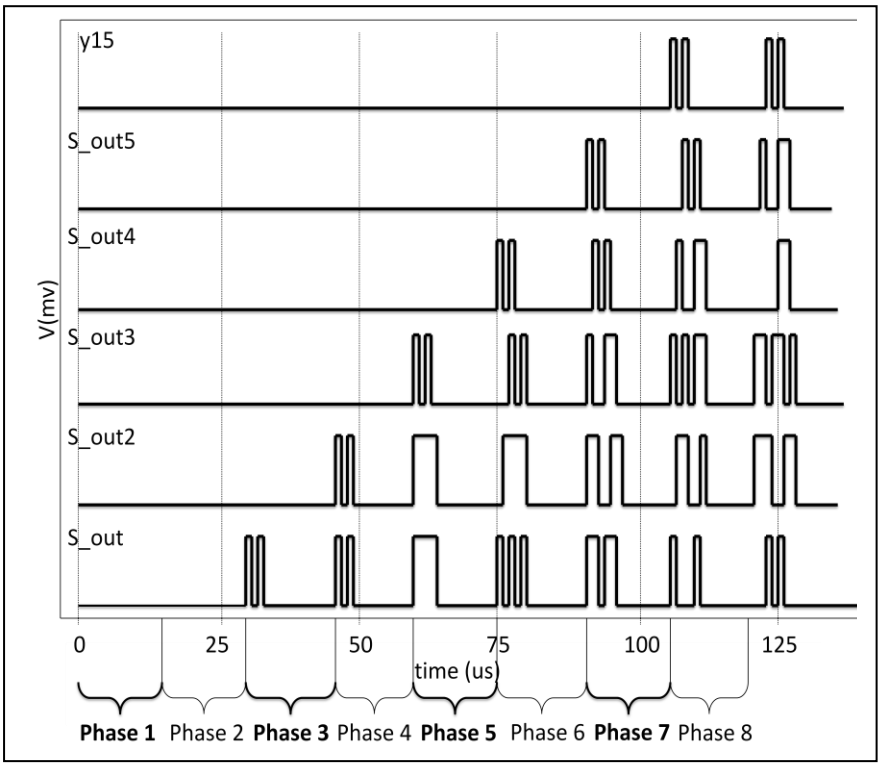

Figure 42: The digital output from the decimator with 15μs phases marked at the bottom

During each phase in Figure 42 one 15-bit word will be processed. The first phase is simply a reset period. In phase 2, the data has been stored in the first 15-bit register, but has not yet appeared at the integrator output. At phase 3, the first word has reached the output of the first integrator. By phases 4 and 5, the second and third integrators have processed data. The first two combs have processed data at phases 6 and 7. Finally, a valid output can be seen by phase 8. Since the data must pass through so many stages before reaching the output, it takes 90 clock cycles, or six phases, before data will begin to appear at the output.

The digital output y15 can be taken in 15-bit words and translated from binary to decimal to see the final representation of the sine wave. Figure 43 shows this normalized sine wave compared to the ideal MATLAB output with D = 45. Since the 1kHz sine wave is decimated by 15 clock cycles, the final output is 66Hz with a normalized 1V peak.

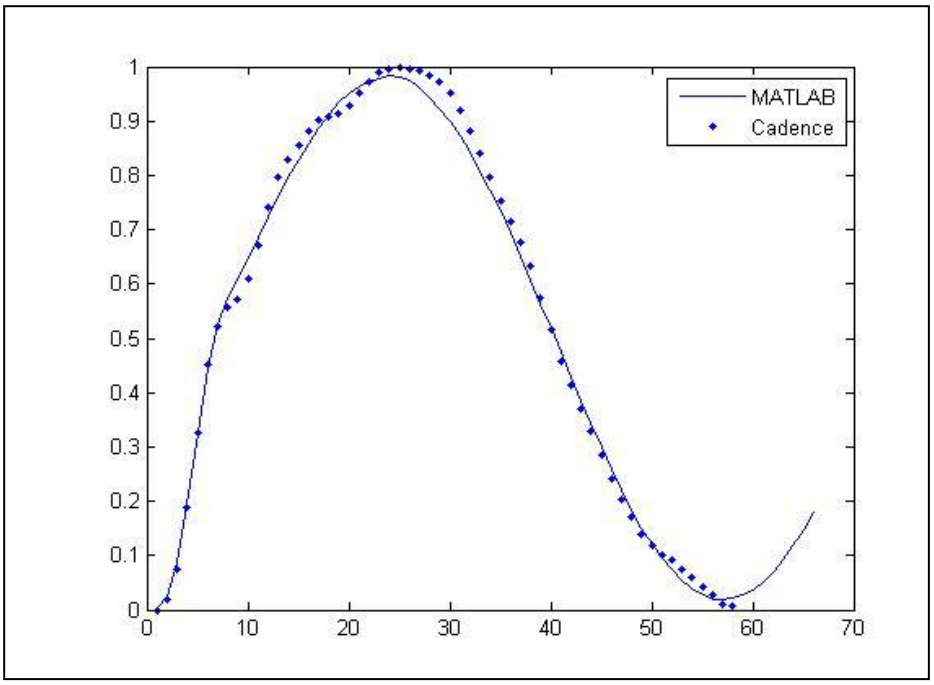

Figure 43: Final simulated output compared to the ideal MATLAB

There is a small amount of distortion at the final output. In order to define this distortion, a Gaussian distribution was taken between the simulated output and the ideal MATLAB output. This curve can be seen in Figure 44. The ADC was designed to have a 400mV peak-to-peak signal at the input with 10-bit resolution. The equation for the LSB voltage is

$$Q = \frac{V_{p-p}}{2^M - 1} \qquad (7)$$

In this equation M is the resolution; therefore the LSB voltage, Q, will equal 400µV. In order for the amount of error observed to be acceptable, it must be significantly below the LSB voltage value. The error distribution has a sigma value of 16.8µV and a mean error of 10µV. The worst case error is the mean plus 3sigma, or 60.4µV which is well below the LSB voltage.

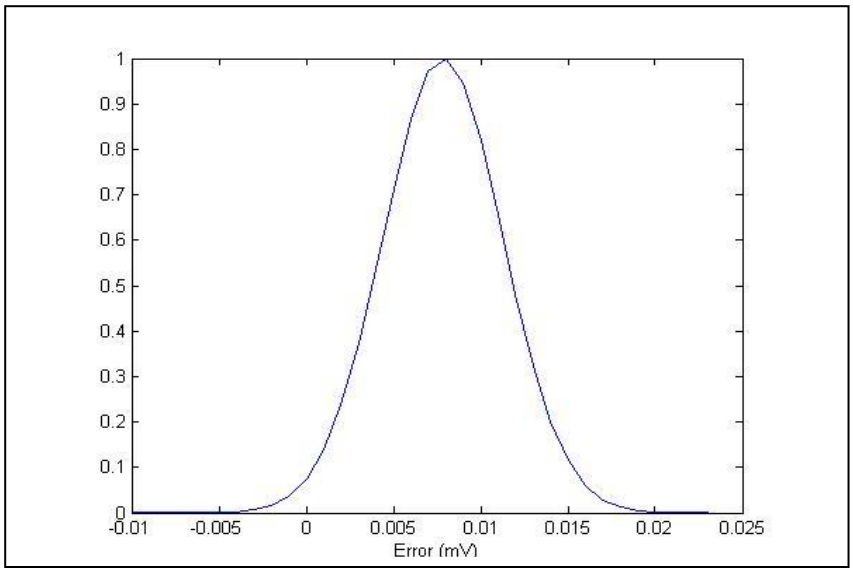

Figure 44: The Gaussian distribution of error between the simulation and the ideal

5.4: DECIMATOR SILICON RESULTS

The decimator, along with several other electronics, was fabricated in silicon by National Semiconductors (now Texas Instruments.) The test bench for testing the chip included voltage sources, function generators and an oscilloscope to replicate the signals that would be seen by the decimator. This setup can be seen in Figure 45. The left picture shows the voltage sources on the left, several function generators in the middle and an oscilloscope on the right. The right picture shows an up close picture of the chip package, which has been wired to external pins.

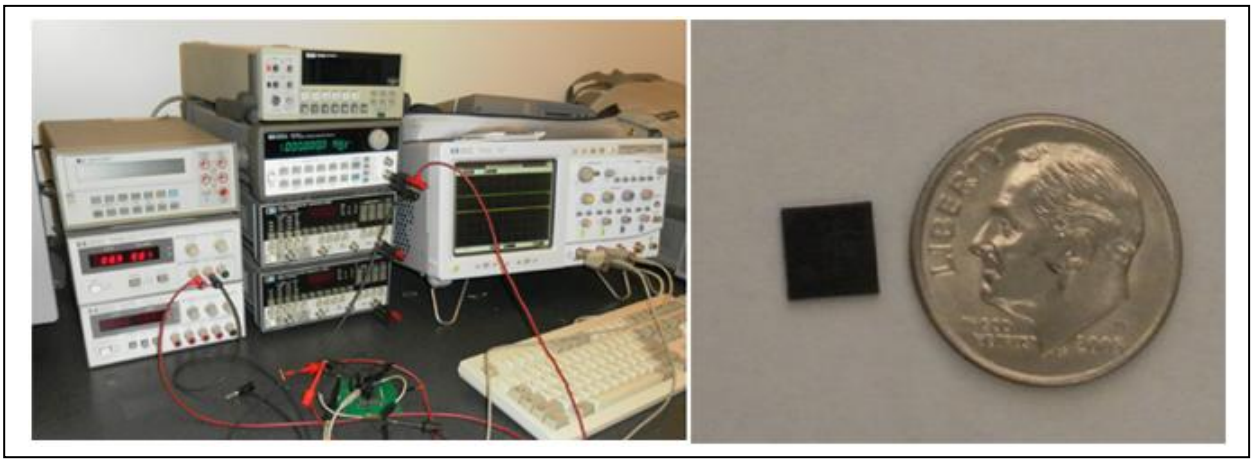

Figure 45: Pictures of the test bench on the left, and a size comparison of the chip itself.

The simplest test to run on the decimator is to give it a solid, non-varying amount of ones, which can be accomplished with a 900mV DC input. Given a *constant* input with number of ones counted n and a word delay of D, the final output will be a decimal number as follows:

$$output = n * D^3 \qquad (8)$$

In this case, D is cubed because it is a third order system. The 15-bit word is delayed by 3 at the comb. Given a 900mV input, 10 ones will be counted and the output will equal $10*3^3$ for a decimal output of 270 or a 15-bit word of 000000100001110. This output was measured on an oscilloscope and can be seen in Figure 46. The LSB of the decimator output is located when the shift signal is high, as indicated in the figure. The decimator output will eventually overflow. However, the decimator quickly recovers, only displaying the incorrect output for 3 phases, or 45 clock cycles.

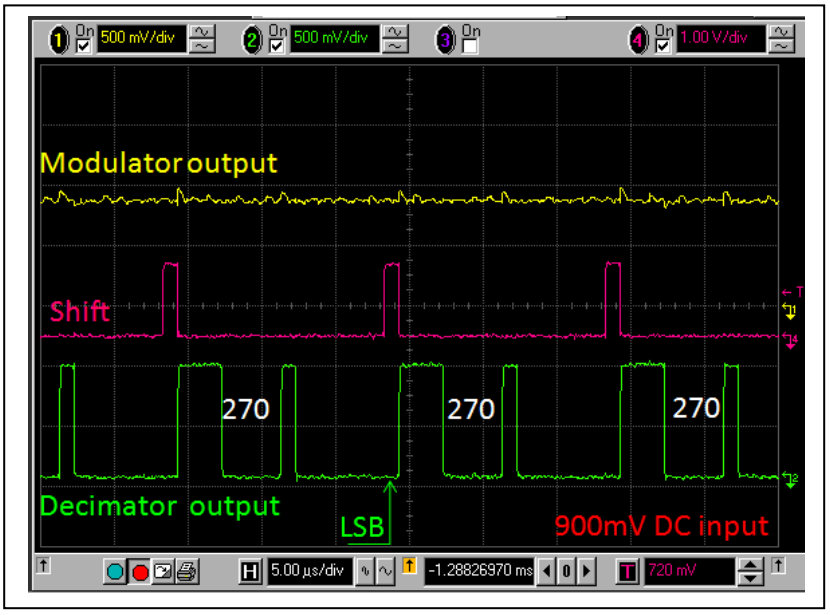

Figure 46: Modulator and decimator outputs given a 900mV DC input

Several additional tests were run by setting the input of the decimator to be square waves of varying frequencies and duty cycles. This allowed the decimator to receive varying amounts

of ones and zeros at the input without having to rely on a working modulator. The results from these inputs can be seen below in Figure 47-Figure 49. The approximate number of ones recorded at the input takes into account that in a 15 clock cycle period, 10 bits will be counted and 5 will be excluded.

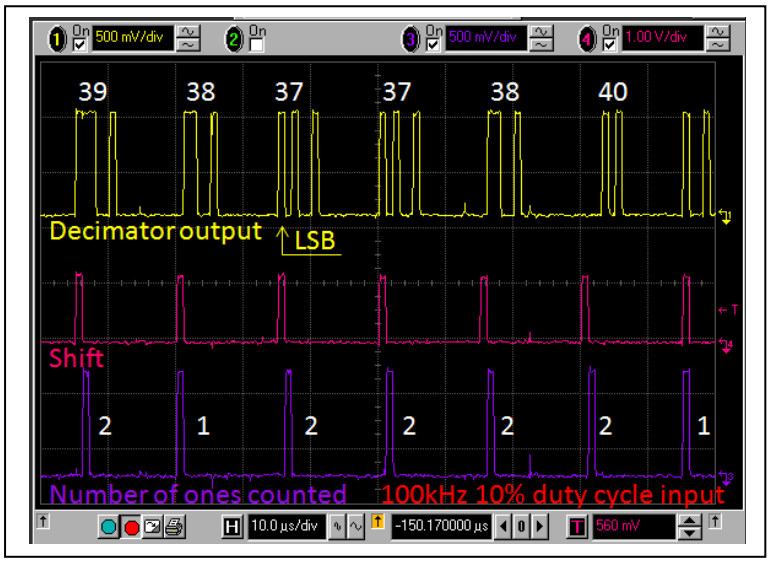

Figure 47: Decimator output given a 100kHz square wave input with 10% duty cycle

Figure 47 shows the first iterations of the decimator output given a 100kHz square wave with a 10% duty cycle. For this input, the decimator recorded a one count between 1 and 2. This should give us an output between 27 and 54. We see an output range of 37-40 which indicates that the decimator is receiving roughly equal counts between 1 and 2. For the graph seen in Figure 48, the decimator recorded a one count between 2 and 3. This should give an output between 54 and 81. The actual output is between 74 and 77, which indicates that the decimator is likely receiving more counts of 3 than of 2.

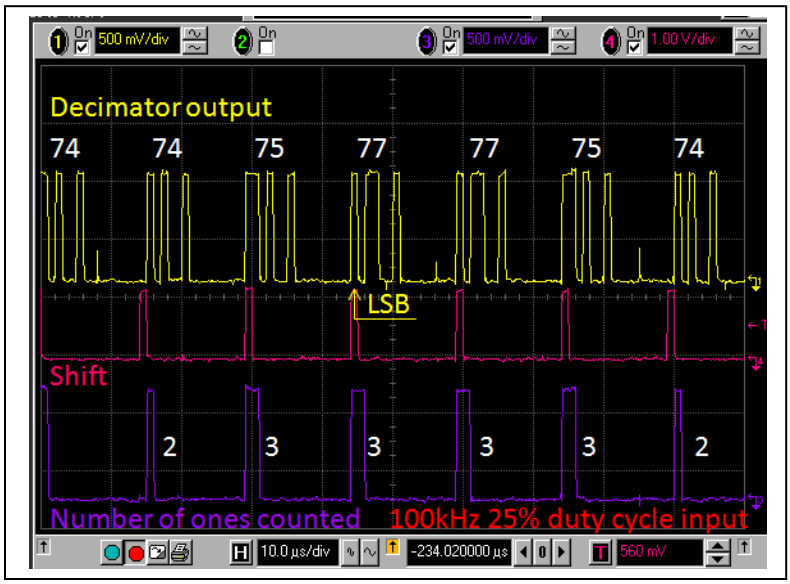

Figure 48: Decimator output given a 100kHz square wave input with 25% duty cycle

The last square wave iteration can be seen in Figure 49. For this run the decimator was given a 300kHz square wave with a 50% duty cycle. The decimator recorded a one count between 5 and 7, which should produce an output between 135 and 189. The actual output varies from 157 to 174, indicating that more 6s and 7s than 5s were recorded. This experiment, as well as the others, was repeated for several different chips to ensure the validity of the results.

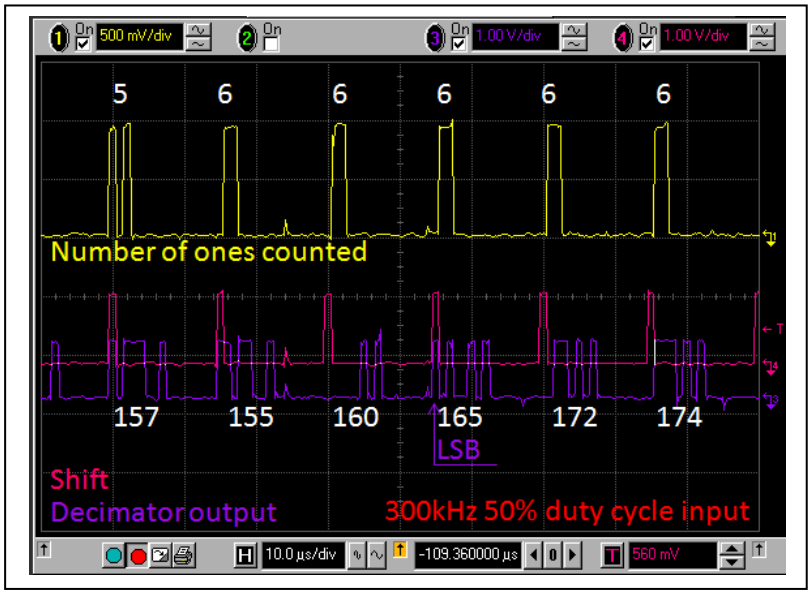

Figure 49: Decimator output given a 300kHz 50% duty cycle square wave input

After this was completed, we connected a working modulator to the input of the decimator to test the ADC as a whole. First, the experiment shown in Figure 47-Figure 49 was repeated, this time by feeding the modulator various DC inputs from 900mV down to 100mV. The results are displayed in Figure 50. This time the modulator output is displayed instead of the number of ones recorded. Recall that only the first 10 ones in a 15-bit cycle will be recorded.

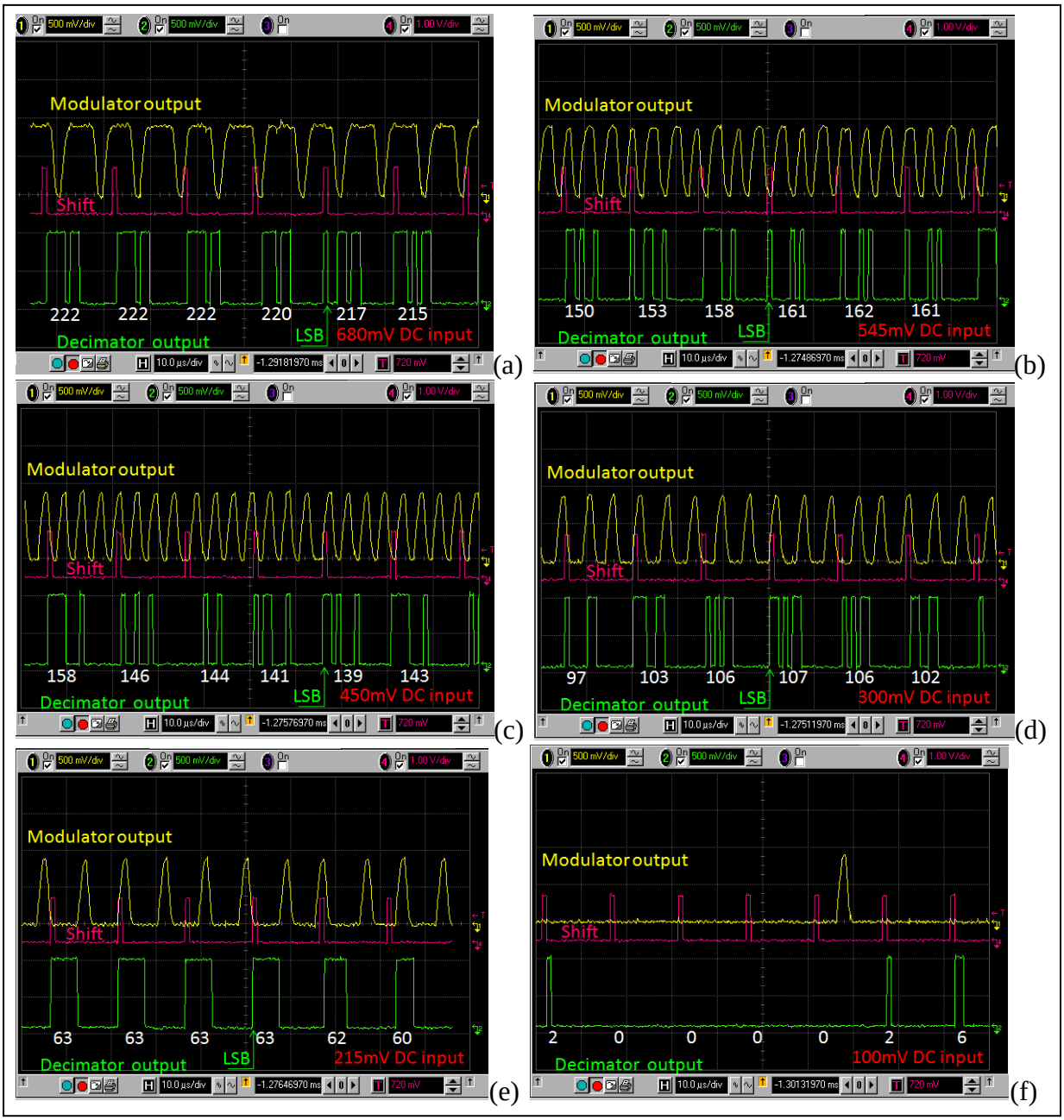

Figure 50: The decimator and modulator outputs given various DC inputs, recorded in red.

The outputs shown in Figure 50 behaved as expected. A direct comparison between the actual results and the expected results can be seen in Table 5. The expected range was calculated by giving the decimator the extreme of the approximate number of ones recorded. So, for example, if the decimator were given 8 ones continuously with no variation, the output would remain steady at 216. For an input of 9 ones only, the output would stay at 243. Thus, we would expect that the decimator output, given a variation of 8-9 ones, would output numbers within the range 216-243. Alternatively, this range could also be found using equation 8. The actual range recorded will depend on how often 8 ones were recorded as opposed to 9 ones. A higher occurrence of 8 will result in a range closer to the lower end of the spectrum and vice versa for a higher occurrence of 9.

Table 5: Decimator results compared to the expected for a given number of ones counted from the modulator

DC Input	Approximate number ones recorded	Maximum expected decimal output range	Actual decimal output	Figure Reference
900mV	10	270	270	Figure 46
680 mV	8-9	216-243	215-222	Figure 50 (a)
545 mV	5-6 (more 6s than 5s)	135-162	150-162	Figure 50 (b)
450 mV	5-6 (more 5s than 6s)	135-162	139-158	Figure 50 (c)
300 mV	3-4	81-108	97-107	Figure 50 (d)
215 mV	2-3	54-81	60-63	Figure 50 (e)
100 mV	0-1 (more 0s than 1s)	0-27	0-6	Figure 50 (f)

The final test for the decimator involved applying a sine wave to the input of modulator to see if both the modulator and decimator worked as expected given such an input. The overall response from the modulator and decimator can be seen in Figure 51. At the top level view, it is easy to see that the modulator is working as expected. When given all zeros, it is obvious that the

decimator is working, as it will also output zeros. The final response to the ones is not immediately apparent from a top level view.

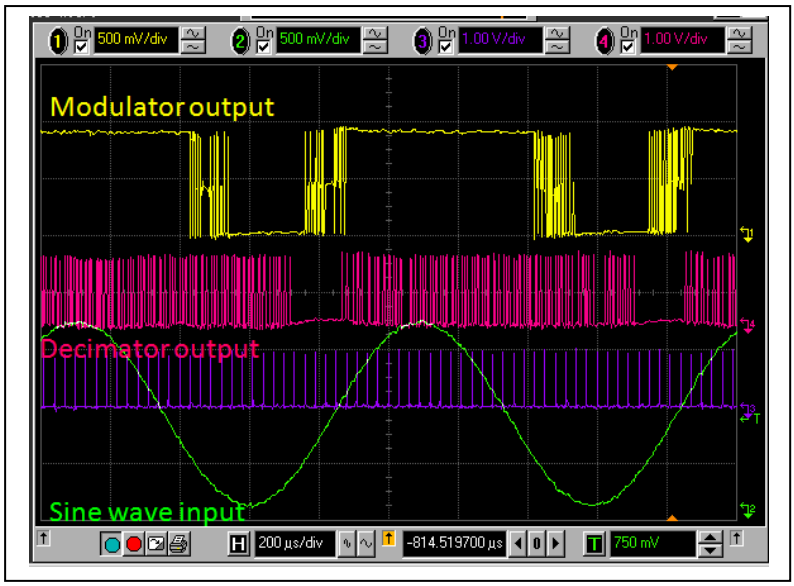

Figure 51: A top level view of the modulator and decimator response to a sine wave input

In order to see if the decimator works correctly when the modulator is producing all ones, a zoomed in response was observed. At this level we could easily see that the output was staying at the expected output of 270 or 000000100001110. The response can be seen in Figure 52. Thus, we can tell that the decimator is working as expected at the extremes.

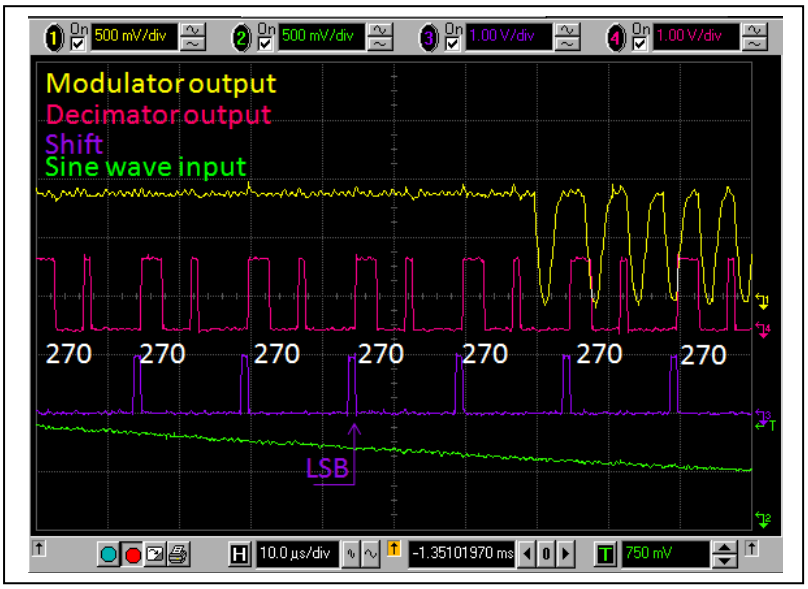

Figure 52: A close up look at the decimator output given all ones from the modulator

To check if the decimator is working correctly in-between the peaks, the decimator responses to the rising and falling slopes of the sine wave were recorded. Figure 53 shows the ascending sequence and Figure 54 shows descending sequence seen at the decimator output. The specific number of incoming ones is not known. Due to the limitations of the oscilloscope there was a limited view of the output waveforms. Therefore, we relied on observing several iterations of the rising and falling slopes to ensure that they behave as expected.

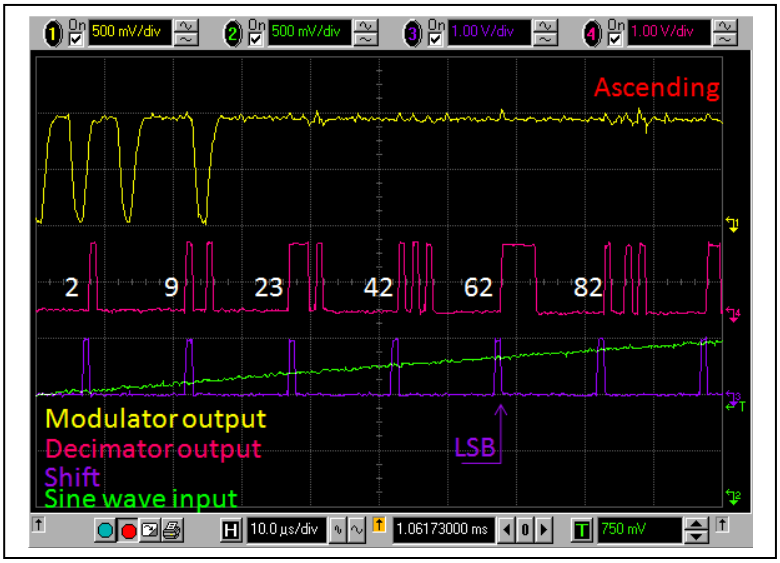

Figure 53: The decimator output as it received a modulator input rising from all zeros

For the rising slope we recorded the following sequences: 2, 8, 22, 42, 62, 82; 2, 9, 23, 42, 62, 82 and 2, 8, 22, 42, 67, 90. Similarly, for the falling slope we recorded the sequences: 94, 73, 48, 25, 9, 2; 92, 65, 40, 21, 8, 2 and 83, 65, 52, 26, 9, 2. The values were observed to increase/decrease at a similar and expected rate. The exact numbers in the sequences vary slightly depending on how many ones were observed.

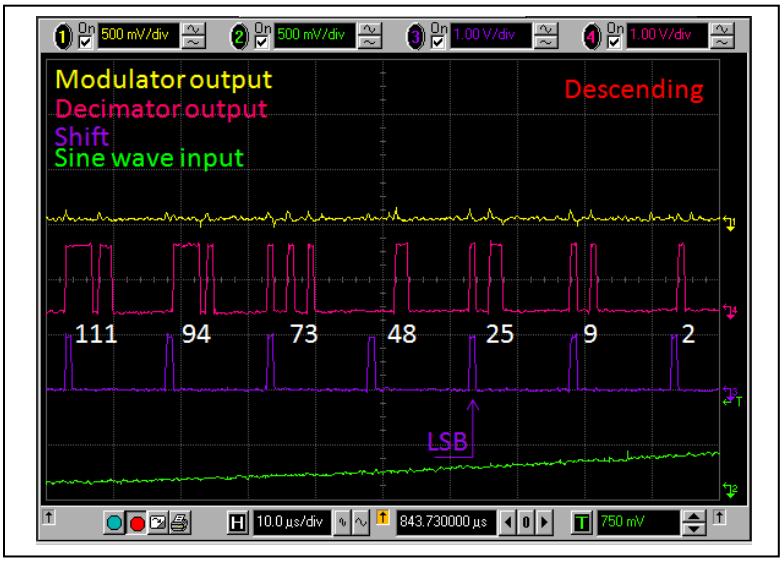

Figure 54: The decimator output as the modulator began descending from all ones

The final check to ensure that the decimator was working correctly was to see if the latency from the modulator to the decimator correlated with the expected 90 clock cycle delay. Figure 55 illustrates the phases for the decimator as it receives input from the modulator. In phase 0, no ones were recorded due to the timing of the input from the modulator. In phase 1, the first ones are recorded. The data will be processed through the integrators in phases 2-4 and through the combs in phases 5-7. Thus, we see the output at phase 7, after a 90 clock cycle delay.

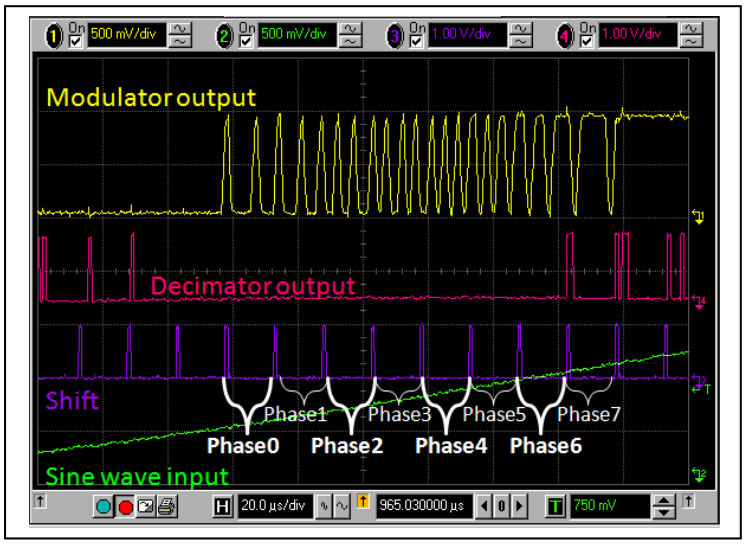

Figure 55: An illustration of the latency between the modulator output and the decimator output

CHAPTER 6: DISCUSSIONS AND CONCLUDING REMARKS

The goal for the decimator was to create a new design that reduced the size of the decimator while keeping it low power. The final layout, shown in Figure 56, was implemented in a 0.18µm CMOS process and runs on 900mV power supply instead of a nominal 1.8V supply. The final layout includes both vdd and vss rings around the outside. In addition, all of the inputs and outputs are routed out to the edges for easy integration with the modulator and any subsequent digital signal processing. The final dimensions of the decimator measured 130.7µm by 121.04µm. This design included the smaller DFF design which reduced the overall area from $0.029mm^2$ to $0.0158mm^2$ for a total area reduction of approximately 46%. The total chip area was $1mm^2$ and the decimator takes up only 1.6% of the available area.

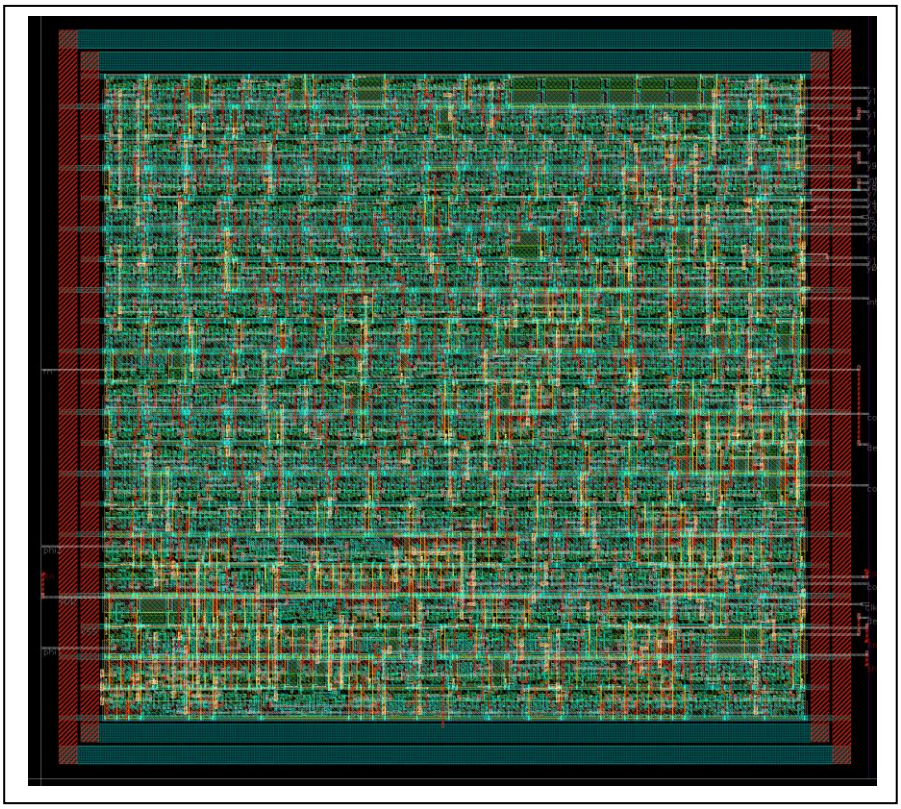

Figure 56: The final layout for the decimator including vss and vdd rings that enclose the overall layout.

This configuration also offers a reduction in power consumption compared to similar decimator designs, consuming 3.3µW of power. Its only drawback is that its data will be valid at a lower frequency than the rest of the chip. However, since this decimator is being used for biomedical applications, a reduced frequency is not an issue. A final comparison between previous work and this work can be seen in Table 6.

Table 6: Final power consumption comparison between decimators

Serial/Parallel	**Total Power**	**Total Area or Gates per Bit**	**Clock Frequency**	**Process Used**
Serial (this work)	3.3µW	0.00105mm^2/bit	1MHz	0.18µm CMOS
Parallel [23]	315µW	0.0013 mm^2/bit	6.144MHz	0.18µm CMOS
Serial [23]	6399 µW	0.0013 mm^2/bit	153.6MHz	0.18µm CMOS
Parallel [26]	2.67mW	0.0055 mm^2/bit	16.8GHz	0.18µm CMOS
Parallel [24]	33.28mW	312 gates/bit	200MHz	0.6µm CMOS
Serial [24]	15.12mW	267 gates/bit	200MHz	0.6µm CMOS
Parallel [25]	85mW	0.2 mm^2/bit	10MHz	0.8µm CMOS

A direct comparison with [24] is difficult as they did not record the final layout area. However, the reduction in power is evident and desirable. This work did show a comparable reduction in size compared to [23] as well as a significant drop in power consumption. Our data was calculated using the final layout which includes the parasitic capacitance and resistance. The final layout as it was pictured in silicon can be seen Figure 57. This design was fabricated by National Semiconductor along with layout for the modulator as well as experimental designs for a potentiostat, TIA and main amp.

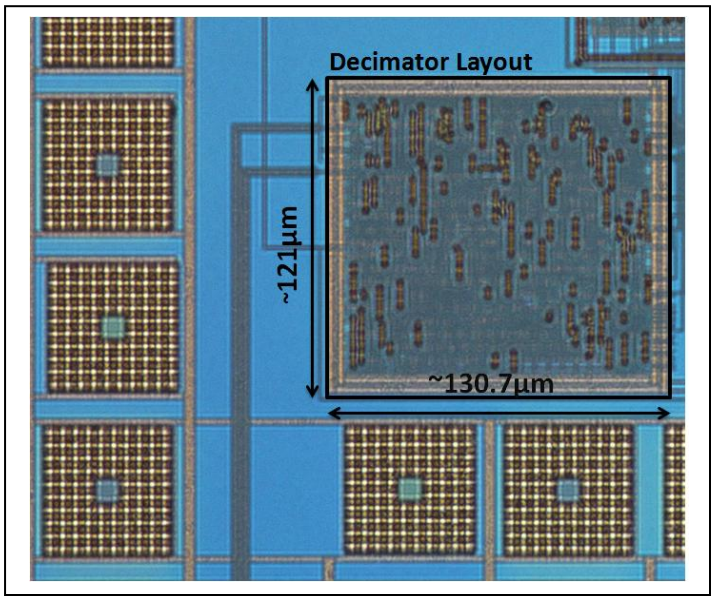

Figure 57: The decimator as it is laid out in silicon

For the biosensor chip, a low power design was also desired, though size was not as much of an issue. The final layout, seen in Figure 58, was fabricated in a 0.6µm CMOS process. The entire 128x64 electrode array measured approximately 2200x2200µm which is large enough to cover most bio-tissue samples used in neurobiology. The chip operates on a 3V supply instead of the nominal 5V and consumes 39.65mW of power. A majority of this power is being consumed by the potentiostat which has a hefty load of 25pF. The 128 read channels, excluding the potentiostat and clock generator, consume a total of 26.16µW. A final comparison between this work and previous work can be seen in Table 7.

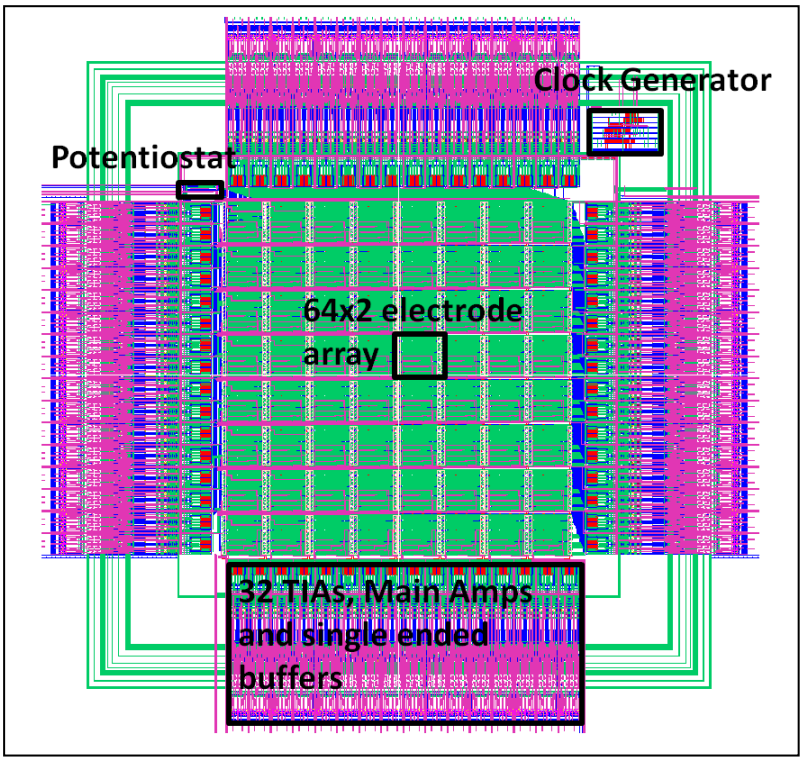

Figure 58: The final layout for the biosensor

Table 7: Final power consumption comparison between biosensor arrays

Number of Read Channels	Total Power	Power per Channel	Process Used
6 [15]	40mW	6.67mW/channel	0.18µm CMOS
32 [16]	22mW	10mW/channel	0.6µm CMOS
100 [13]	13.5mW	0.135mW/channel	0.5µm CMOS
128 (this work)	39.65mW	0.310mW/channel	0.6µm CMOS

There is a significant gain in both number of read channels and power consumed per channel between this work and previous work. While the power consumption per channel in [13] is slightly less, the design does not include an on-chip potentiostat. For our design a potentiostat is including on-chip as well as the entire electrode array. Additionally, though the total power for this work is only slightly below the power consumption in [15], this work also includes more than 20 times the amount of channels, making the power consumed per channel much better.

The final goal for each of the designs was low power and low area. A summary of the performance of each piece is outline in Table 8. The vital components, a single sub-array and the decimator, are emphasized in italics. A summary for the TIA, main amp and potentiostat are also included to show how their performances and area affect the final design.

Table 8: The final performance summary for array, decimator, and other electronics

	Single Electrode Array	**Decimator**	**TIA**	**Main Amp**	**Potentiostat**
Area	*0.0446mm^2*	*0.0158mm^2*	0.0114mm^2	0.0131mm^2	0.0107mm^2
Power	*0.310mW/channel*	*3.3µW*	112.5µW	84.15µW	645µW
Input Range	*100pA-2nA*	*10-bit word bit-stream*	100pA-2nA	±500mV	±800mV with <1mV offset
Output Range	*100pA-2nA*	*15-bit word*	10-200mV	±1V	

Both the decimator and electrode array designs will play a part in the larger goal of designing a biosensor for the detection of NO. The decimator advances the backend signal processing electronics. Its reduced size and power allows for several ADC to be included on a single chip so that several read channels can be used at once. The electrode array design is the first step in creating a viable design to test the electronics directly in a real world environment without the need of external electronics.

BIBLIOGRAPHY

[1] J. Wang, "Electrochemical glucose biosensors," *Chemical Reviews,* vol. 108, pp. 814-825, 2008.

[2] A. Sassolas, B. D. Leca-Bouvier and L. J. Blum, "DNA biosensors and microarrays," *Chemical Reviews,* vol. 108, pp. 109-139, 2008.

[3] J. Wang, C. Wu, N. Hu, J. Zhou, L. Du and P. Wang, "Microfabricated electrochemical cell-based biosensors for analysis of living cells in vitro," *Biosensors,* vol. 2, no. 2, pp. 127-170, 2012.

[4] C. M. Li, J. Zang, D. Zhan, W. Chen, C. Q. Sun, A. L. Teo, Y. T. Chua, V. S. Lee and S. M. Moochhala, "Electrochemical detection of nitric oxide on a SWCNT/RTIL composite gel microelectrode," *Electroanalysis,* vol. 18, no. 7, pp. 713-718, 2006.

[5] H. Bayir, "Reactive oxygen species," *Critical Care Medicine,* vol. 33, no. 12, pp. S498-S501, 2005.

[6] K. Kuriyama and S. Ohkuma, "Role of Nitric Oxide in central synaptic trnasmission: effects on neurotransmitter release," *The Japanses Journal of Pharmacology,* vol. 69, no. 1, pp. 1-8, 1995.

[7] J. Garthwaite and C. L. Boulton, "Nitric oxide signaling in the central nervous system," *Physiology,* vol. 57, pp. 683-706, 1995.

[8] Z. H. Taha, "Nitric oxide measurements in biological samples," *Talanta,* vol. 61, no. 1, pp. 3-10, 2003.

[9] N. Hogg, "Detection of nitric oxide by electron paramagnetic resonance spectroscopy," *Free Radical Biology and Medicine,* vol. 49, pp. 122-129, 2010.

[10] V. Tiwari, D. Singh, S. Rajgopal, G. Mehta, R. Patel and F. Baez, "Reducing power in high-performance microprocessors," in *Design Automation Conference,* New York, 1998.

[11] B. J. Hosticka, R. W. Brodersen and P. R. Gray, "MOS sampled data recursive filters using switched capacitor integrators," *IEEE Journal of Solid-State Circuits,* Vols. SC-12, no. 6, pp. 600-608, 1977.

[12] T. Gabay, M. Ben-David, I. Kalifa, R. Sorkin, Z. R. Abrams, E. Ben_jacob and Y. Hanein, "Electro-chemical and biological properties of carbon nanotube based multi-electrode arrays," *Nanotechnology,* vol. 18, no. 3, 2007.

[13] R. Harrison, P. Watkins, R. Kier, R. Lovejoy, D. Black, B. Greger and F. Solzbacher, "A low-power integrated circuit for a wireless 100-electrode neural recording system," *Solid-State Circuits,* vol. 42, no. 1, pp. 123-133, 2007.

[14] M. Sawan, "Microsystems dedicated to wireless multichannel monitoring and microstimulation: design, test and packaging," in *Solid-State and Integrated Circuits Technology,* 2004.

[15] J. Coulombe, Y. Hu, J.-F. Gervais and M. Sawan, "An implant for a visual cortical stimulator," in *Canadian Design Engineering Network Conference*, Montreal, 2004.

[16] X. Yun, D. Kim, M. Stanacevic and Z. Mainen, "Low-power high-resolution 32-channel neural recording system," in *Engineering in Medicine and Biology Society*, Lyon, 2007.

[17] K. Uyttenhove and M. S. J. Steyaert, "A 1.8V 6-bit 1.3-GHz flash ADC in 0.25um CMOS," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 7, pp. 1115-1122, 2003.

[18] M. O. Shaker and M. A. Bayoumi, "A 90nm low-power successive approximation register for A/D conversions," in *Quality Electronic Design, 12th International Symposium on*, Santa Clara, 2011.

[19] Y. Park, S. Karthikeyan, W. M. Koe, Z. Jiang and T. Tan, "A 16-bit, 5MHz multi-bit sigma-delta ADCusing adaptively randomized DWA," in *Custom Integrated Circuits Conference*, Dallas, 2003.

[20] S. Lee and C. Cheng, "A low-voltage and low-power adaptive switched-current sigma-delta ADC for bio-acquisition microsystems," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 53, no. 12, pp. 2628-2636, 2006.

[21] J. M. de la Rosa, "Sigma-delta modulators: tutorial overview, design guide and state-of-the-art survey," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 58, no. 1, pp. 1-21, 2011.

[22] E. B. Hogenauer, "An economical class of digital filters for decimation and interpolation," in *Acoustics, speech and signal processing*, 1981.

[23] S. M. M. Zanjani, S. R. Omam, S. M. Fakhraie and O. Shoaei, "Experimental evaluation of different realizations of recursive CIC filters," in *Canadian Conference on Electrical and Computer Engineering*, Ottawa, 2006.

[24] C. Lei, Z. Yuanfu, G. Deyuan, W. Wu, W. Zongmin, Z. Xiaofei and P. Heping, "A decimation filter design and implementation for oversampled sigma delta A/D converters," in *IEEE International Workshop on VLSI Design and Video Technology*, Suzhou, 2005.

[25] K. Nakamura, M. Hotta, R. Carley and D. Allstot, "A 85-mW, 10-bit 40-Ms/s ADC with decimated parallel architecture," in *Custom Integrated Circuits Conference*, San Diego, 1994.

[26] B. Babu, K. N. Shesharaman and H. M. Kittur, "Power optimized digital decimation filter for medical applications," *Advances in Computing and Communications*, pp. 90-93, 2012.

[27] W. Pettine, M. Jibson, T. Chen, S. Tobet, P. Nikkel and C. S. Henry, "Characterization of novel microelectrode geometries for detction of neurotransmitters," *Sensors Journal, IEEE*, pp. 1187-1192, 2012.

[28] M. Duwe and T. Chen, "Low power integrated potentiostat design for uelectrodes with improved accuracy," in *Circuits and Systems (MWSCAS)*, Seoul, 2011.

[29] W. Wilson, R. Selby and T. Chen, "A current-starved inverter-based differential amplifier design for ultra-.ow power applications," in *IEEE LASCAS*, Cusco, 2013.

[30] R. Selby, T. Kern, W. Wilson and T. Chen, "A 0.18um CMOS switched-capacitor amplifier using current-starving inverter based op-amp for low-power biosensor applications," in *IEEE LASCAS*, Cusco, 2013.

[31] O. Choksi and R. Carley, "Analysis of Switched-Capacitor Common-Mode Feedback Circuit," *Circuits and Systems II: Analog and Digital Signal Processing*, vol. 50, no. 12, pp. 906-917, 2003.

APPENDIX A: RAW CADENCE DATA AND ITS RESULTS

Table 9: The following table describes how the final output from the decimator is derived into decimal form. The raw data is represented in the decimator output column, selected at 1μs intervals.

Time (μs)	Decimator output (V)	One or zero detected	Binary word	Decimal output
1	2.68e-7	0		
2	2.68e-7	0		
3	2.68e-7	0		
4	2.68e-7	0		
5	2.68e-7	0		
6	2.68e-7	0		
7	2.68e-7	0		
8	2.68e-7	0		
9	2.68e-7	0		
10	2.68e-7	0		
11	2.68e-7	0		
12	2.68e-7	0		
13	2.68e-7	0		
14	2.68e-7	0		
15	2.68e-7	0	000000000000000	0
…	…	…	…	…
106	0.9	1		
107	2.82e-7	0		
108	0.9	1		
109	2.4e-7	0		
110	2.42e-7	0		
111	2.44e-7	0		
112	2.46e-7	0		
113	2.48e-7	0		
114	2.50e-7	0		
115	2.52e-7	0		
116	2.54e-7	0		
117	2.57e-7	0		
118	2.59e-7	0		
119	2.61e-7	0		
120	2.63e-7	0	000000000000101	5
…	…	…	…	…
301	2.68e-7	0		
302	0.9	1		
303	2.66e-7	0		
304	2.67e-7	0		
305	2.68e-7	0		
306	0.9	1		

307	0.9	1		
308	0.9	1		
309	1.1e-6	0		
310	9.62e-7	0		
311	8.24e-7	0		
312	6.85e-7	0		
313	5.47e-7	0		
314	4.09e-7	0		
315	2.7e-7	0	000000011100010	226

First the raw decimator output is determined to be a one or zero. In this case the supply voltage is 900mV, so a value of 0.9V is considered a one. Values in the μV range and lower are small enough to be considered zeros. The outputs are collected into 15 bit binary words with the LSB first. These values are then translated to decimal. The final string of decimal numbers will recreate the analog sine wave input.

A final decimal output string will typically look as follows:

0, 0, 0, 0, 0, 0, 0, 5, 20, 50, 86, 119, 138, 147, 151, 161, 177, 196, 210, 219, 226, 233, 238, 240, 241, 245, 251, 257, 261, 263, 264, 263, 262, 260, 257, 251, 243, 233, 222, 210, 199, 189, 179, 167, 152, 136, 121, 109, 98, 87, 75, 64, 54, 45, 37, 31, 27, 24, 20, 16, 11, 7, 3, 2, 4, 10…

The beginning is all zeros as the decimator starts its iterations through the three integrators and three combs. Then the output starts rising steadily before reaching a value of about 264, the peak of the sine wave, and then descending back towards zero. This string represents half of the digitized sine wave, as was shown in Figure 43.

APPENDIX B: MATLAB CODE FOR IDEAL DECIMATION

```
% desired differential delay:

D = 30; % D = 15; and D = 45 were also used
% desired decimation:
R = 15;
% resulting comb delay:
N = D/R;

% bit stream from modulator given an analog sine wave input:
xn = [0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 1 0 1 0 1 0 1 0 1 1 0 1 0
1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1
0 1 1 0 1 1 1 0 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1 1 1 0 1
1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1
1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1
0 1 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 0 1 1 0 1 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0
1 0 1 0 1 1 0 1 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 1 0 1 0 1 0 1 0 0 1
0 1 0 0 1 0 1 0 0 1 0 0 1 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0
1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0
0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0
0 1 0 0 1 0 1 0 0 1 0 0 1 0 0 1 0 1 0 0 1 0 1 0 0 1 0 1 0 0 1 0 1 0 1 0 1 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 1 0 0 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 1 1 0
1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1
1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1 0 1
1 1 0 1 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 0 1 1 0 1 0
1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 1 1 0 0 1 0 1 0 1
0 1 0 0 1 0 1 0 1 0 0 1 0 1 0 1 0 0 1 0 0 1 0 1 0 0 1 0 0 1 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 1 0
0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0
0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0
```

```
0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0
1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 1 0];

% Third order CIC filter
% Below are the buffers to be used for the comb section delay. These buffers will eventually be
filled in with the integrator output, similar to a shift register made of flip flops. They are
initialized to be length N and filled with zeros.
delayBufferOne = zeros(1,N);
delayBufferTwo = zeros(1,N);
delayBufferThree = zeros(1,N);

% These are the placeholders for the output of the integrator and comb sections. They can be
treated as wires, storing one bit of data. For each iteration through the integrators and combs, the
final values will be stored in an array. The values are initialized to zero.
intOneOut = 0;
intTwoOut = 0;
intThreeOut = 0;
combOutOne = 0;
combOutTwo = 0;
combOutThree = 0;

% R_Value is set up in order to simulate the decimation by 15 seen in the physical circuit. It has
the value of R (which equals 15), R*2, R*3 ... R*length(xn), so we can discard all but every Rth
value. The actual circuit version will do the decimation differently, but in theory the two will do
the same thing.
R_Value = R:R:length(xn);

j=1; % index for R_Value, initialized to 1

for i = 1:length(xn)
   %integrators:
   intOneOut = intOneOut + xn(i);
   intTwoOut = intTwoOut + intOneOut;
   intThreeOut = intThreeOut + intTwoOut;

   % storing the integrator outputs:
   intOneOutNode(i) = intOneOut;
   intTwoOutNode(i) = intTwoOut;
   intThreeOutNode(i) = intThreeOut;

   % decimation and comb:
   if(i == R_Value(j)) % checks to see if we are at the Rth value, the 2*Rth value, etc.
      temp(j)=intOneOutNode(i); % for double checking the value if necessary
      output(j) = intThreeOutNode(i); % final integrator output will only use every Rth
intThreeOutNode point
      % Comb 1:
```

```
        combOutOne = output(j) - delayBufferOne(end);

        % Updating comb 1's buffers:
        delayBufferOne(2:end) = delayBufferOne(1:end-1);
        delayBufferOne(1) = output(j);

        % Comb 2:
        combOutTwo = combOutOne - delayBufferTwo(end);

        % Updating comb 2's buffers:
        delayBufferTwo(2:end) = delayBufferTwo(1:end-1);
        delayBufferTwo(1) = combOutOne;

        % Comb 3:
        combOutThree = combOutTwo - delayBufferThree(end);

        % Updating comb 3's buffers:
        delayBufferThree(2:end) = delayBufferThree(1:end-1);
        delayBufferThree(1) = combOutTwo;

        % storing the comb outputs:
        combOneNode(j) = combOutOne;
        combTwoNode(j) = combOutTwo;
        combThreeNode(j) = combOutThree;

        if(j < length(R_Value))     % a check to make sure the index j doesn't get larger than
R_Value can handle
            j = j + 1;
        end
        % implied else j = j;
    end
end

% plot the normalized output from the third comb filter. This is the plot that will be considered
the ideal when it is compared the Cadence simulated output.
plot(combThreeNode/D^3,'b','LineWidth',3)
```