To the Graduate Council:

I am submitting herewith a dissertation written by Xiaoyan Yu entitled "A Novel Frequency Based Current-to-Digital Converter with Programmable Dynamic Range." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Discipline, with a major in Electrical Engineering.

Ethan Farquhar, Major Professor

We have read this dissertation
and recommend its acceptance:

Ben Blalock

Syed Islam

Suzanne Lenhart

Accepted for the Council:

Carolyn R. Hodges
Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

A Novel Frequency Based Current-to-Digital Converter with Programmable Dynamic Range

A Dissertation

Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Xiaoyan Yu

August 2009

Dedication

To my beloved family.

Acknowledgement

First, I would like to thank my advisor, Dr. Ethan Faquhar, for kindly advising me these years. It has been one of the best things that happened to me for having been worked in Dr. Farquhar's lab. I'd like to also thank Dr. Blalock for offering me another wonderful environment to continue my study, research and dissertation. Thanks his supporting and guidance all these years. I'd like to thank Dr. Islam for his most valuable and inspiring discussions. They are very precious to me. It is an honor for me to have Dr. Lenhart being in my committee. Thank her for the insightful comments and evaluation of my work.

Also, I'd like to thank all my lab mates: Dr. Neena Nambiar, Chandra Ulaganathan, Kevin Tham, Yu Long, Malathi Narnur, Deepak Jayaram, Adrianne Thrash, Richard Smiddy, Saeed Ghezawi, Reid Rankin. You are like family to me. It has been a long and rough road. And without your accompany and inspiration, it would be so much harder.

Last, but not least, thanks to all the faculty members in the E.E.C.S. department who have assisted in my education at the University of Tennessee.

Abstract

This work describes a novel frequency based Current to Digital converter, which would be fully realizable on a single chip.

Biological systems make use of delay line techniques to compute many things critical to the life of an animal. Seeking to build up such a system, we are adapting the auditory localization circuit found in barn owls to detect and compute the magnitude of an input current.

The increasing drive to produce ultra low-power circuits necessitates the use of very small currents. Frequently these currents need to accurately measured, but current solutions typically involve off-chip measurements. These are usually slow, and moving a current off chip increases noise to the system. Moving a system such as this completely on chip will allow for precise measurement and control of bias currents, and it will allow for better compensation of some common transistor mismatch issues.

This project affords an extremely low power (100s nW) converter technology that is also very space efficient. The converter is completely asynchronous which yields ultra-low power standby operation [1].

Table of Contents

Chapter 1 Introduction 1
1.1 Motivation 1
Chapter 2 Literature Review 3
2.1 Reviews on Analog Voltage to Digital Converter 3
2.2 Reviews on Analog Current to Digital Converter 10
Chapter 3 Previous Related Work: Sound Localization 20
3.1 Background 20
3.2 Why Localization is Important 22
3.3 Other methods of doing localization 24
3.4 System Architecture 25
Chapter 4 Delay Line Structure and operation 29
4.1 Biological Neurons 29
4.2 Integrate and Fire Neuron 31
4.3 Delay Line 34
Chapter 5 System Structure and Operational Principle 37
5.1 System Architecture 37
5.2 Modified integrate and fire neuron 37
5.3 Operational Principle 40
Chapter 6 Proof of Concept 46
6.1 Size and its Improvement 46
6.2 Power 50
6.3 Linearity 50
6.3.1 Various Capacitors Solution 54
6.3.2 Floating-Gate Solution 56
6.3.3 Translinear-Circuit Solution 59
6.4 Prototype layout and simulation result 62

Chapter 7 Test Results and Analysis 64
7.1 Test setup 64
7.2 Test results 66
7.3 Original contributions and figure of merit 75
7.3.1 Original contributions 75
7.3.2 Figure of merit 77
7.4 Future work 83
Bibliography 84
Vita 90

List of Figures

Figure 2-1 A 3-bit parallel ADC [2] 4
Figure 2-2 the successive-approximation process [2] 5
Figure 2-3 Block diagram of a delta-encoded ADC 6
Figure 2-4 Pipeline implementation of the algorithmic ADC [2] 8
Figure 2-5 diagram of a single-slope serial ADC [2] 8
Figure 2-6 Block diagram of a delta-sigma ADC [3] 9
Figure 2-7 Block diagram of a first order delta-sigma modulator [3] 10
Figure 2-8 A schematic of flash ADC [4] 11
Figure 2-9 Circuit of basic flash ADC block [4] 11
Figure 2-10 Modified current comparator from H. Traff [5] 12
Figure 2-11 A half-flash ADC schematic [6] 13
Figure 2-12 Block diagram of current-mode folding and interpolating ADC [7] 14
Figure 2-13 N-bit successive-approximation ADC [8] 15
Figure 2-14 Schematic of the elementary 1-bit cell of the ADC [8] 17
Figure 2-15 Schematic of a delta-encoded ADC [9] 17
Figure 2-16 Block diagram of a serial ADC circuit architecture [10] 19
Figure 2-17 Block diagram of the second-order incremental A/D converter [11] 19
Figure 3-1 Barn owl's locating its prey [12] 21
Figure 3-2 The Jeffress (1948) model [13] 21
Figure 3-3 5 nodes delay line and sound angle of 60^{o} 23
Figure 3-4 System architecture of arrays of integrated and fired neurons 26
Figure 3-5 Position map for sound localization 28
Figure 4-1 A view of a typical neuron [14] 30
Figure 4-2 Real action potential from the invertebrate snail (helisoma trivolvis) [15] 30
Figure 4-3 Integrate and fire Neuron [16]. (a) n-type (b) p-type 31
Figure 4-4 Simulation results of typical V_1 and V_{out} of n-type integrate and fire neuron. 33
Figure 4-5 Simulation results of typical V_1 and V_{out} of p-type integrate and fire neuron . 33
Figure 4-6 Delay line structure [16] 35
Figure 4-7 Experiment result of V_1 and V_{out} of one cell in the delay line 35
Figure 4-8 Experiment result of V_{out} of two adjunct cells in the delay line 36
Figure 5-1 Block diagram of the current ADC 38
Figure 5-2 Schematic of the modified integrate and fire neuron 38
Figure 5-3 System progress 42
Figure 5-4 Delay lines with different transport speeds 42
Figure 5-5 Input current 1: 1^{st} delay line spikes 43
Figure 5-6 Input current 1: 4^{th} delay line spikes 44
Figure 5-7 Input current 1: 8^{th} delay line spikes 45
Figure 6-1 Area optimizing design 47
Figure 6-2 Layout picture of 8-bit ADC (a) 256 cells, (b) 32 cells and a 3 -bit counter .. 49
Figure 6-3 System operation principle 49
Figure 6-4 Current consumption for (a) rest condition (b) conversion condition 51

Figure 6-5 Input/output characteristics [17] 51
Figure 6-6 I-array position curve: linear resolution 52
Figure 6-7 I-array position curve: this ADC 53
Figure 6-8 A part of the I-array position curve 53
Figure 6-9 I-array position curve 57
Figure 6-10 Floating gate architecture to get 8 different bias voltages [45]. 60
Figure 6-11 Translinear circuit [38] 60
Figure 6-12 micrograph of the chip A 62
Figure 6-13 Simulation result of the chip layout 63
Figure 7-1 Micrograph of the chip B 64
Figure 7-2 System structure for chip B 65
Figure 7-3 Test setup 65
Figure 7-4 Delay line outputs 66
Figure 7-5 Translinear output I_1 versus I_2 67
Figure 7-6 Translinear output I1 × I2 67
Figure 7-7 Frequency versus input current 68
Figure 7-8 Frequency versus input current 69
Figure 7-9 Duty cycle versus input current 69
Figure 7-10 Period (1/f) versus ADC output (8bits) 70
Figure 7-11 Current to digital conversion compared with absolute linear line 71
Figure 7-12 INL 71
Figure 7-13 DNL 72
Figure 7-14 Standby power 74
Figure 7-15 Conversion Power 74
Figure 7-16 Energy cost versus input Current 76
Figure 7-17 Area (mm^2) per output codes 79
Figure 7-18 Power (mW) per output codes 79
Figure 7-19 Minimum detectable current 80
Figure 7-20 Comparison between this work and other work by FOM1 82
Figure 7-21 Comparison between this work and other work by FOM2 82

Chapter 1
Introduction

1.1 Motivation

Analog-to-Digital converters (ADCs) are one of the most critical blocks in the electronics field. Traditionally ADCs are most often based on voltage-mode, however, due to voltage-mode ADCs' inherent disadvantages, such as voltage swing problems, current-mode ADCs have attracted great interest in recent years, especially, when the analog signal to be processed comes in the form of a current. Take for example, the output current of a photo detector. Voltage-mode ADCs have to convert current to voltage and then convert the voltage to a digital representation. Current-mode ADCs (also referred to as current to digital converters) can perform a direct conversion, which yields advantages [7] [8] [18] [19] [20].

The increasing drive to produce ultra-low power circuits necessitates the use of very small currents. Frequently these currents need to be accurately measured, but current solutions typically involve off-chip measurements. They are usually slow, and moving a current off chip increases noise in the system. Moving a system such as this completely on chip will allow for precise measurement and control of bias currents, and it will allow for better compensation of some common transistor mismatch issues.

This is the basic motivation for us: to build a small size current to digital converter, which is fully realizable on a single chip, and is remarkably power efficient, allowing it to be used in implantable bio-medical applications, and vision chips [1].

Chapter 2

Literature Review

2.1 Reviews on Analog Voltage to Digital Converter

Many ways have been conceived of to implement a voltage ADC. By speed, they can be classified to high-speed (Flash), medium-speed (Successive-approximation, Delta-encoded, Pipeline), or slow-speed (Serial). By sampling rate, they can be classified as Nyquist or oversampled (Delta-Sigma). For oversampled ADC, a trade off exists between speed and resolution. Generally speaking, the higher speed, the lower resolution one should expect to achieve.

The common ways to implement a voltage ADC are reviewed here [2] [21]:

Parallel or Flash ADCs

The basic structure of a parallel or flash ADC is shown in Figure 2-1. Eight equal resistances divides the V_{REF} into eight values as indicated on the figure. Each of these values is then compared with V_{in}, in "parallel". The results would be sent to an encoder for translation to a digital output.

The advantage of this type of ADC is that it achieves the highest conversion speed; thus the name "flash". The disadvantages are that it uses a large area, and the resistance and bias currents result in inefficient energy use. As a result of the large areas consumed, flash ADCs are usually limited to 8 bits or less resolution.

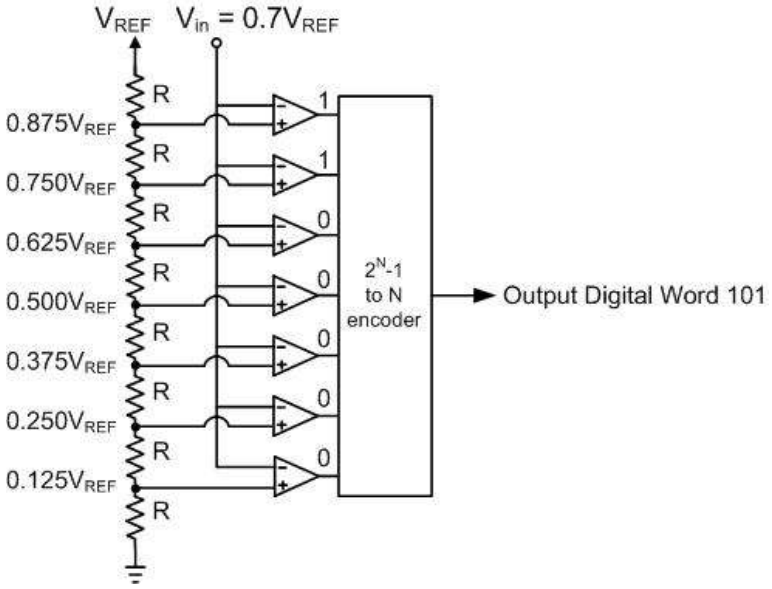


Figure 2-1 A 3-bit parallel ADC [2]

Some have sought to make improvements to standard parallel ADCs. This has led to the development of both interpolating ADCs and folding ADCs. Both of these designs seek to reduce the number of comparators used, resulting in a decrease in both area and power consumption [2].

Successive-approximation ADC

Successive-approximation ADCs trade off speed for area. They iteratively seek for the correct value over a number of clock cycles in a binary search pattern. This is illustrated in Figure 2-2.

Take $0.7V_{REF}$ as V_{in} for an example. In the first iteration, the voltage V_{in} is compared to half of V_{REF}. The output is "1", since V_{in} is bigger than V_{REF}. Then, the second iteration is to compare V_{in} with $0.75V_{REF}$. The output is "0", since V_{in} is smaller than $0.75V_{REF}$. The third

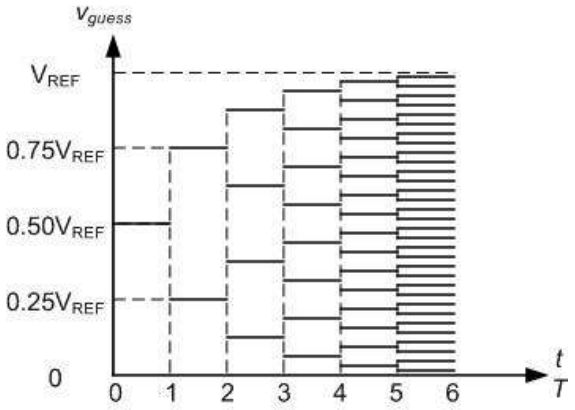


Figure 2-2 the successive-approximation process [2]

iteration compares V_{in} with $0.625V_{REF}$. The output is "1", since V_{in} is bigger than $0.625V_{REF}$. This process is repeated until the correct voltage is achieved.

From the process, we can see that, for N-bit resolution, N-iterations of comparison are required, which makes it a medium-speed ADC. It's more complicated than some of other ADCs, but it has a good resolution and a wide range.

Delta-encoded ADC

The block diagram of Delta-encoded ADC is shown in Figure 2-3. A digital reference voltage v_{ref} is fed into a digital to analog converter (DAC). Then the output of the DAC, and also the input voltage, are both fed into the comparator. The output voltage, after going through conditional gates, gives negative feedback to the digital reference voltage. That is to say, when the reference voltage is larger than the input voltage, v_{ref}^{*} will be decreased; when it's smaller than the input voltage, v_{ref}^{*} will be increased, until they match. Then the corrected v_{ref}^{*} will be read out as the output.

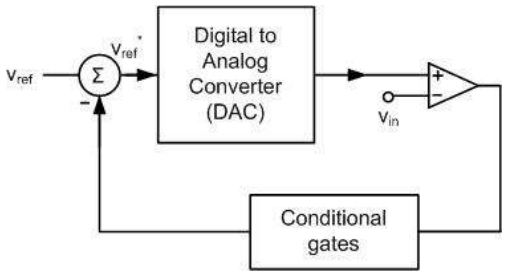


Figure 2-3 Block diagram of a delta-encoded ADC [2]

Delta-encoded ADCs have very wide ranges, and high resolution. The speed is related to the input signal level. This ADC is often be used to detect real world signals, since they frequently do not change abruptly.

Pipeline ADC

A pipeline ADC can be looked as a refinement of successive-approximation ADC. For successive-approximation ADC, the current stage only brings one piece of information, "1" or "0", to next stage. In a pipeline ADC, it brings much more information, which speeds up the conversion. From Figure 2-4, the mathematical description of the voltages of two stages next to each other is

$$V_i = 2V_{i-1} - b_{i-1}V_{REF} \tag{2-1}$$

Where bi_{-1} is given as

$$b_{i-1} = \begin{cases} 1, if V_{in,i-1} > V_R / 2 \\ 0, if V_{in,i-1} < V_R / 2 \end{cases} \tag{2-2}$$

Serial ADC

Single-slope ADCs are an example of a serial ADC. As shown in Figure 2-5, an input clock signal is sent to generate a ramp and one input of an AND gate at the same time. At the beginning, the output voltage of the comparator jumps to "1", since v_{in} would be larger than the initial ramp voltage. When t is equal to nT, the ramp voltage begins to catch v_{in}, and the output voltage of the comparator jumps back to "0". Thus by the output of the AND gate, we get a serial of clock signals, with a length of nT.

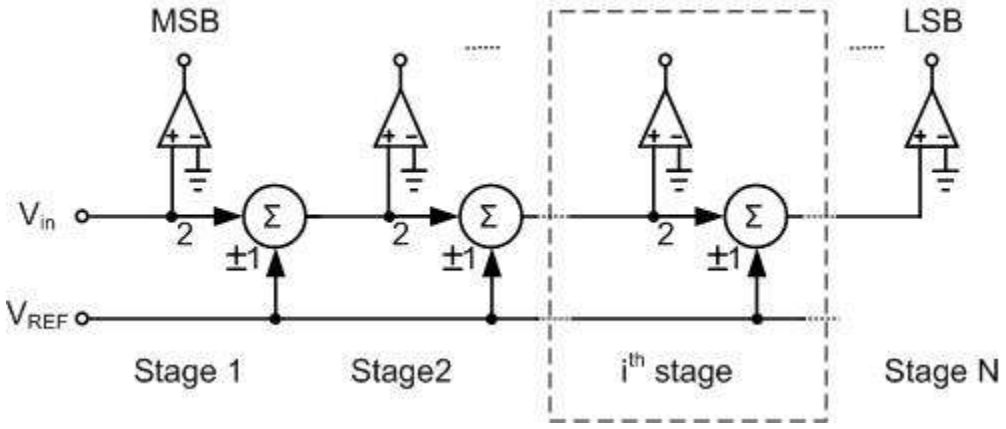


Figure 2-4 Pipeline implementation of the algorithmic ADC [2]

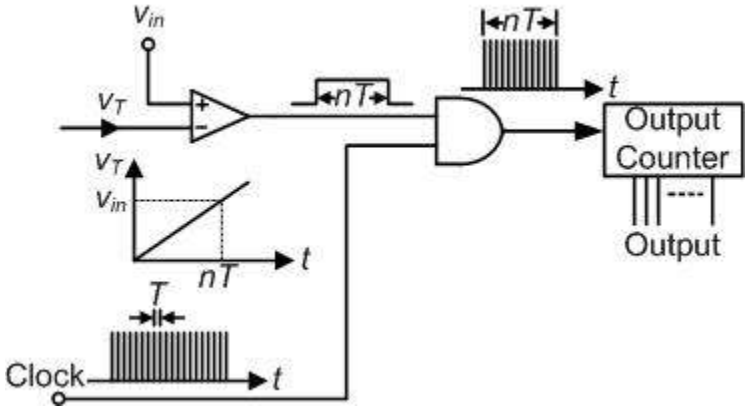


Figure 2-5 diagram of a single-slope serial ADC [2]

Serial ADC has high resolution but low speed. Under worst condition, for an N-bit ADC, a conversion time of $2^N T$ is required, while v_{in} is close to V_{REF}.

Delta-Sigma ADC [3]

All the above ADCs are sampled at the Nyquist rate, this results in relatively high speeds but relatively low resolution, when compared with Delta-Sigma ADC. Delta-Sigma ADCs are over-sampling converter, typically with a oversampling ratio between 8 and 256 [2].

The block diagram is shown in Figure 2-6. An analog signal first goes through a delta-sigma modulator, which will convert the analog signal to a digital signal, and then pass it through a low pass filter to filter out the high frequency components of the signal.

The key to a delta-sigma ADC is the delta-sigma modulator. A sample of first order analog delta-sigma modulator is shown in Figure 2-7.

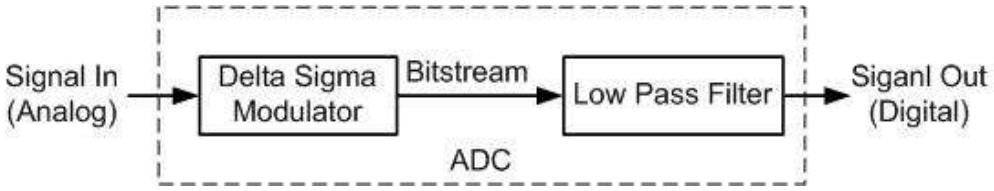


Figure 2-6 Block diagram of a delta-sigma ADC [3]

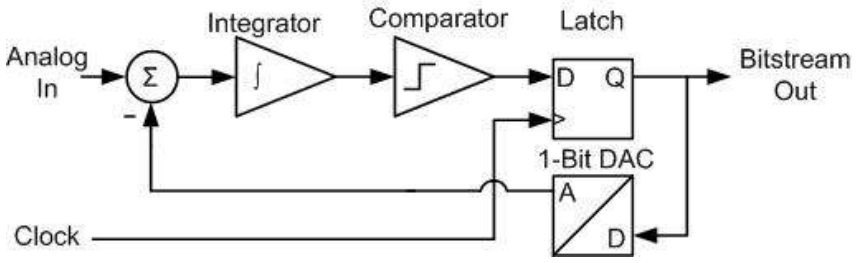


Figure 2-7 Block diagram of a first order delta-sigma modulator [3]

2.2 Reviews on Analog Current to Digital Converter

The existing ways to implement an analog *current* to digital converter are mainly parallel/folding and interpolating ADCs. Other ways, such as serial [10], delta-encoded [9] [22], and delta-sigma ADCs [11] are rarely used.

Parallel ADCs [4] [5]

Figure 2-8 is a typical schematic of flash ADCs. A current comparator is used to compare the reference current and the input current. The output of the comparator is boosted to become a digital signal. Then an encoder will transfer the 2^n digital number to n digital number. This design method is straightforward, and it is among the fastest. However, if the number of bits (n) increases, the area will increase by the level of 2^n, which results in this topology's limitation. A typical realization of this schematic is shown Figure 2-9.

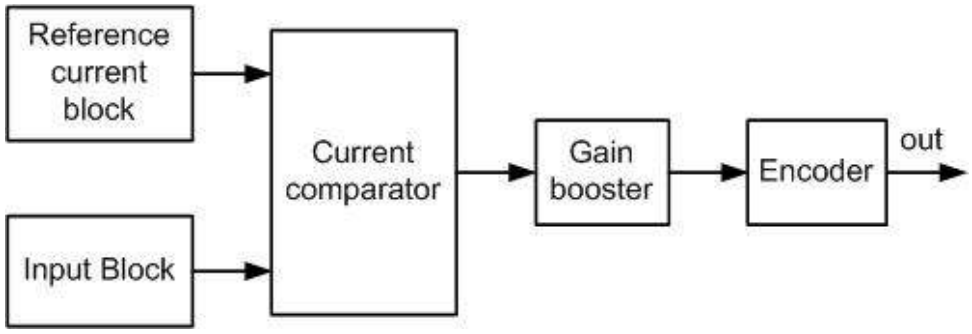


Figure 2-8 A schematic of flash ADC [4]

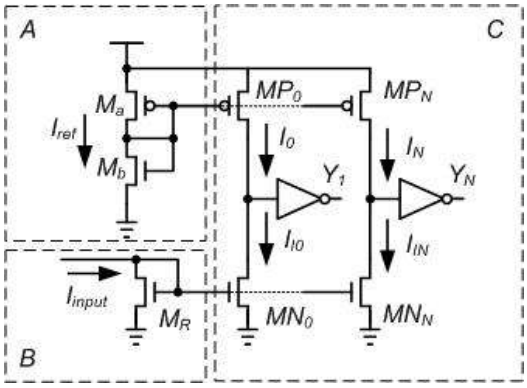


Figure 2-9 Circuit of basic flash ADC block [4]

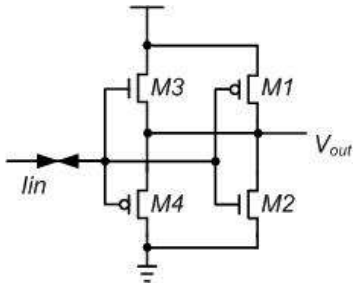


Figure 2-10 Modified current comparator from H. Traff [5]

In Figure 2-9, part A is a reference current generator. The reference current generated from this circuit depends on the parameters of M_a and M_b. By changing the width/length ratio of M_{P0}, M_{P1} ... M_{PN}, a group of various current references can be achieved.

Part B is the input block. A DC current offset may need to be added to the input current to adjust the MOSFET into the right bias range. Similarly to part A, a current mirror is used to get the input current I_{input} on the other side.

Part C is a basic comparator with a gain booster, which is implemented with an inverter. To achieve high speed performance, the basic comparator can be replaced by a modified comparator, shown in Figure 2-10.

The modified current comparator decreases the input impedance of the current comparator, thus increasing the high speed performance.

Half-flash ADCs [6] 23]

For a flash ADC, the resolution is hard to be increased, due to area and cost limitations. To take the advantage of the speed of flash ADCs, while not sacrificing much in area, half-flash ADCs are used in [6] [23].

As Shown in Figure 2-11, this half-flash ADC mainly includes *N* units of a flash ADC. Each of the parallel flash ADC converts its input current to 4-bits. The input current I_{in} is fed into the first flash ADC, generating 4-bit outputs and I_{in1}; then I_{in1} is fed into the second flash ADC for getting another 4-bit outputs and I_{in2}. This continues until the desired number of bits is achieved.

Folding and interpolating ADCs [7]

Figure 2-12 is a block diagram of a current-mode folding and interpolating ADC [7]. The input current is first fed into a coarse ADC to get the MSBs, and also to get the inputs to the folding amplifiers. Folding amplifiers then generate a folded current to feed the interpolating stages. The current generated from the interpolating stages compares with the threshold current to get the LSBs.

This process is similar to how voltage-mode folding and interpolating ADCs works. The main difference is: voltage mode folding amplifiers are based on differential-pairs, while current mode folding amplifiers are based on current mirrors. These have better linearity, no voltage swing problem, and are more suitable for low supply voltage application. The current mode interpolating circuit is based on current mirrors also.

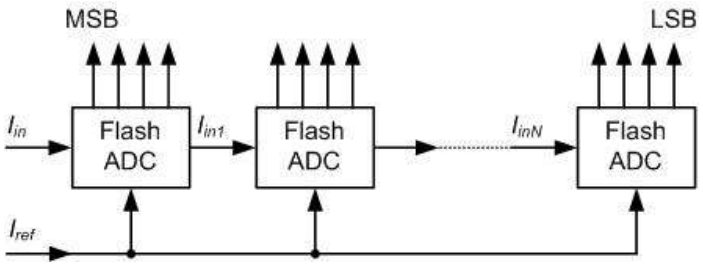


Figure 2-11 A half-flash ADC schematic [6]

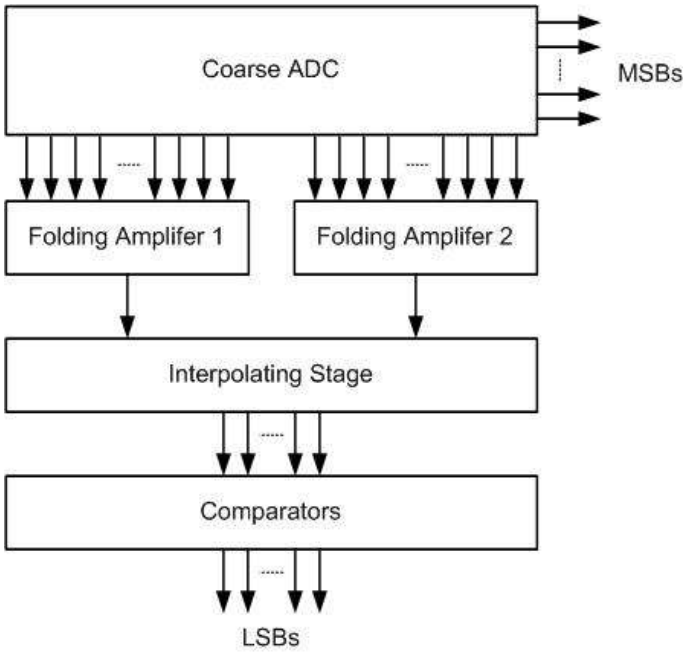


Figure 2-12 Block diagram of current-mode folding and interpolating ADC [7]

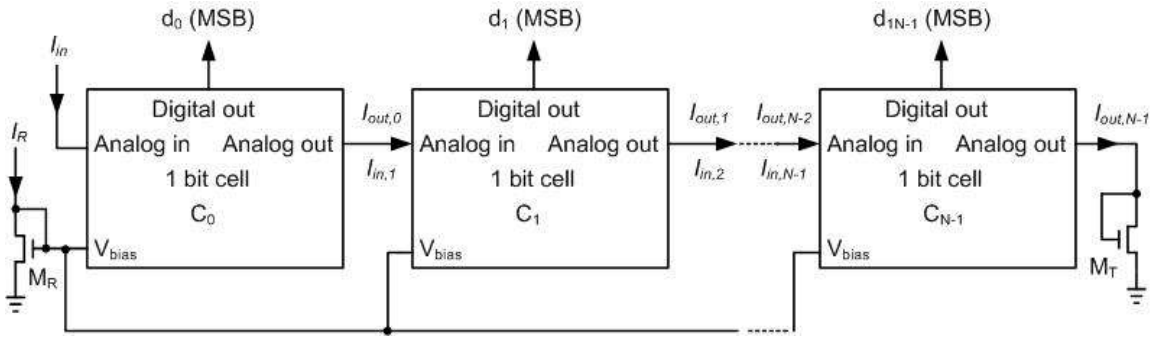


Figure 2-13 N-bit successive-approximation ADC [8]

Algorithmic/successive/pipeline/cyclic ADCs [8]

Figure 2-13 is a typical block diagram of a pipeline ADC [8]. An input current is first fed into the first cell to generate a digital number (d_0), which would be the MSB. Then, depending on the result of d_0, an output current ($I_{0ut,0}$), which is also $I_{in,1}$, is going to be sent to the second cell. The second cell then convert this input current ($I_{in,1}$) to another digital number (d_1) , and generate an output current ($I_{0ut,1}$), which is also $I_{in,2}$, depending on the result of d_1. This process continues until the lowest digital bit d_{N-1} is converted. Then the last MOSFET M_T gets $I_{0ut,N-1}$, and the conversion is done. By changing the reference current I_R, different conversion range can be achieved.

The relationship between d_i and $I_{in,i}$ is:

$$d_i = \begin{cases} 1, if I_{in,i} > I_R/2 \\ 0, if I_{in,i} < I_R/2 \end{cases} \quad 2\text{-}3$$

And the relationship between $I_{out,i}$, which is also $I_{in,i+1}$ is:

$$I_{out,i} = I_{in,i+1} = 2 * I_{in,i} - d_i * I_R \qquad 2\text{-}4$$

Figure 2-14 is one way of the circuit design of 1-bit ADC cell.

Assume M_1, M_2 and M_3 have the same width/length ratio, and M_4, M_5 and M_6 have the same width/length ratio. Then through two current mirrors, the comparator then compares $2I_{in,i}$ with I_R. If $I_{in,i} > I_R/2$, get d_i, which is the digital output, equal to *1*; or If $I_{in,i} < I_R/2$, get d_i equal to *0*. This digital output will control the gate of M_9. When $d_i = 1$, I_R is going to be subtracted from the output current $I_{out,i}$; when $d_i = 0$, $2I_{out,i}$ will just be equal to $2I_{in,i}$.

Delta-encoded ADCs [9]

Figure 2-15 is a typical delta-encoded ADC [9].

A 6-bit flash ADC is used as a coarse ADC for quick speed. The input current first goes through the 6-bit flash ADC, and then is converted back to an analog current. Subtracting this converted current from the original input current gets the residual current. This residual current will go through a current amplifier, which would amplify the input current by 2^{6-1} times under this condition, and then feed back to the 6-bit flash ADC. The result of this time coming out of the 6-bit flash ADC would be the LSBs.

The current residue amplifier determines the speed of whole system, since the flash ADC part is very fast. One possible way to solve this problem was discussed in [9]. An operational amplifier was assumed to deliver high output current.

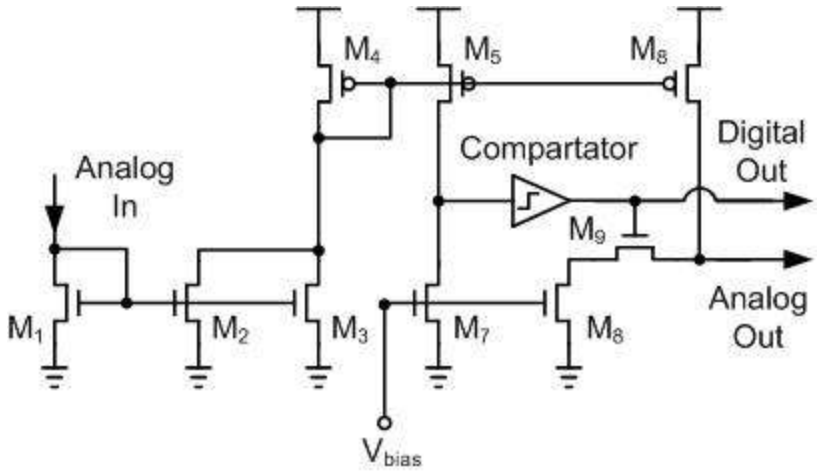


Figure 2-14 Schematic of the elementary 1-bit cell of the ADC [8]

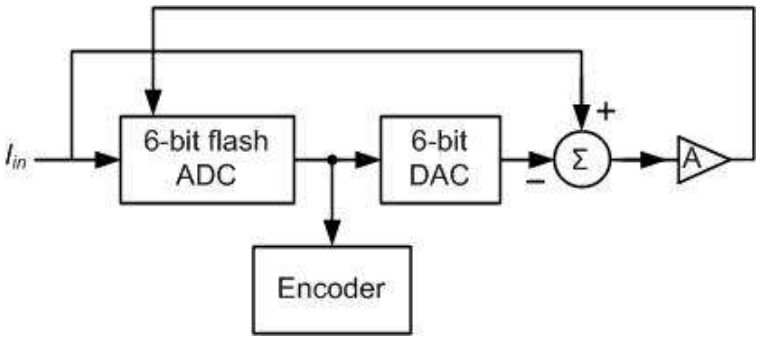


Figure 2-15 Schematic of a delta-encoded ADC [9]

Serial ADCs [10]

Figure 2-16 is a block diagram of a serial ADC circuit architecture [10]. An analog input current is converted to a ramp current signal by an integrator and an amplifier stage. The input current I_{in} is first converted to a voltage by capacitor C_1; then this voltage is converted to current information again by the MOSFET T_1. Biasing the gate voltage of T_1 in a preset small range can make the conversion linear. The output current is a ramp current, which feeds the window current comparator to generate a pulse voltage. The pulse width of this pulse voltage is then fed into a counter to determine the speed of the ramp current, which is linearly related to the input current I_{in}. Digital signal processing after that will get the output digital current.

Delta-sigma ADCs [11]

A second-order Delta-sigma ADC is described in [11]. Figure 2-17 is the block diagram. This second-order converter is based on a cascade of two first-order converters. For each first-order converter, it includes an integrator, a comparator, and a D/A converter. The output of a is the most significant bits, and the output of b is the least significant bits. a is converted back to analog and subtracted from Iin. The difference is integrated, feeding a comparator to get a, and at the same time, working as the input of the other first-order converter to get b. A decimation filter is used to remove the noise, and output the desired digital current.

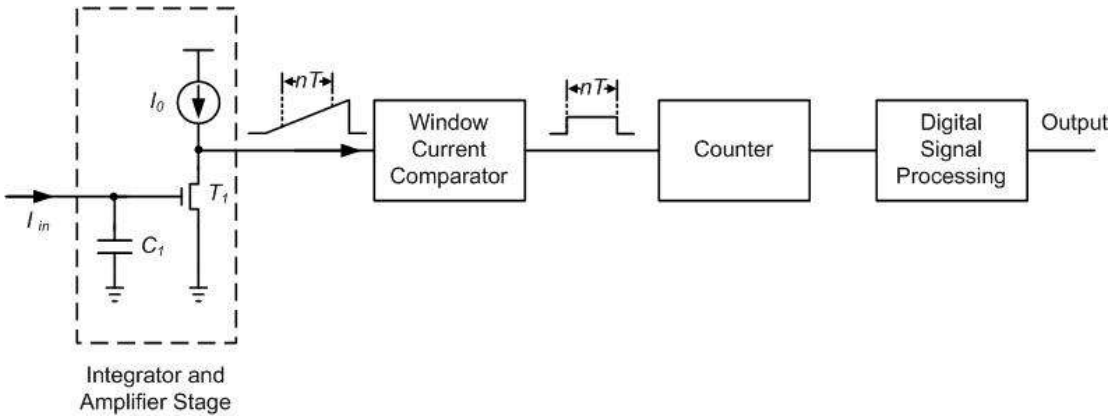


Figure 2-16 Block diagram of a serial ADC circuit architecture [10]

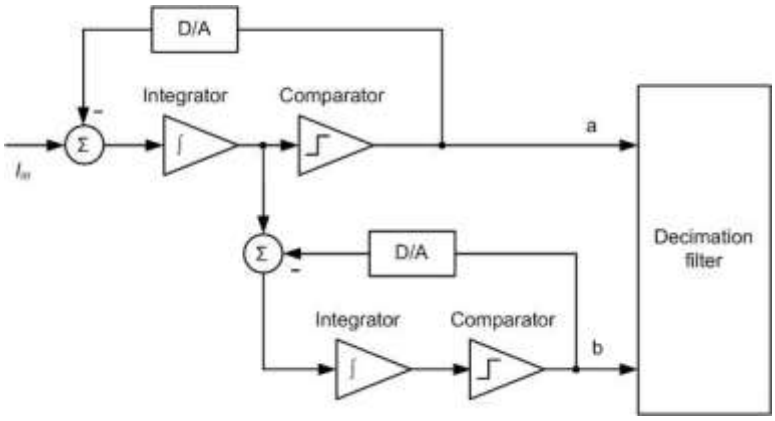


Figure 2-17 Block diagram of the second-order incremental A/D converter [11]

Chapter 3

Previous Related Work: Sound Localization

3.1 Background

The idea behind this work is inspired from a certain biological species. The barn owl has developed a unique method of locating prey, which it can accomplish to a high degree of accuracy even in the absence of visual feedback [13] [24] [25].

Unlike a bat, a barn owl uses a passive system to locate its prey [12]. To do this, it employs a time delay circuit. By simply correlating two time-delayed auditory signals (one from each ear), the owl is able to determine where its prey is at in the horizontal plane, as shown in Figure 3-1. Interestingly, the ears of the barn owl are not located in the same horizontal plane of its head, but are slightly offset to help with detection in the other plane [16].

The circuit established in the nucleus laminaris by the magnocellular afferents resembles a model for the detection of interaural time differences, which is proposed by Jeffress in 1948. This model is composed of two elements, delay lines and coincidence detectors, as shown in Figure 3-2 [13].

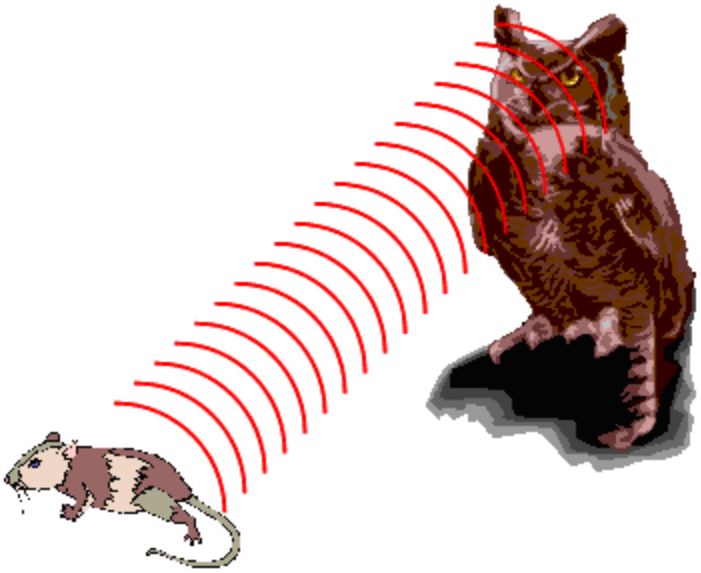


Figure 3-1 Barn owl's locating its prey [12]

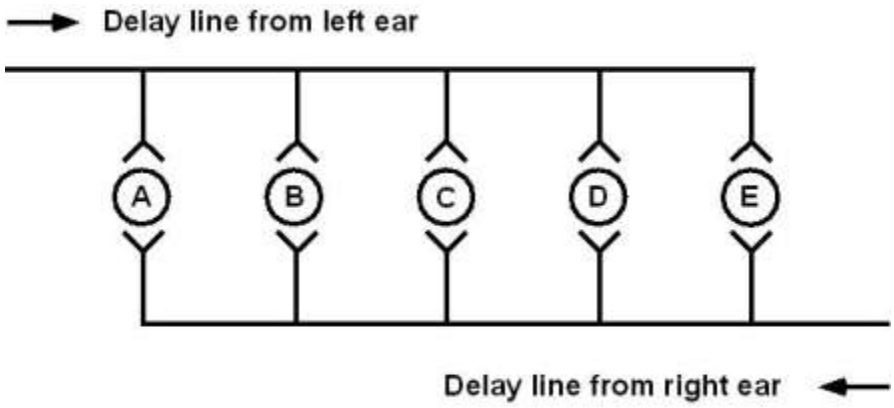


Figure 3-2 The Jeffress (1948) model [13]

Figure 3-3 is the simulation result of a simplified barn owl localization which has *5* delay lines starting from both left and right sides. Define the very left side as the 0^o, and 4^{th} delay line cell spikes when the sound coming from an angle of 60^o.

3.2 Why Localization is Important

Finding the exact position of a tag in a 3-D space is useful in numerous applications [26]:

A. Mobile robots application

Localization is a fundamental problem of mobile robots. It is useful in different tasks, such as office delivery, to have the knowledge about the position of a robot [27]. For the goal of localization of robots, a common way is to have a sensor and some landmarks or beacons (which can be artificial or natural) [28].

B. Medical application

There are many examples in the medical field where localization is important. For example, a prostate tumor needs to be located in three dimensions during external beam radiotherapy and irradiate the tumor as a moving target. The goal of radiotherapy is to reduce the dose to the surrounding tissue (so as to decrease complications) and escalate the tumor radiation dose to eliminate all cancer cells. Localization is the decisive factor to increased tumor control [26].

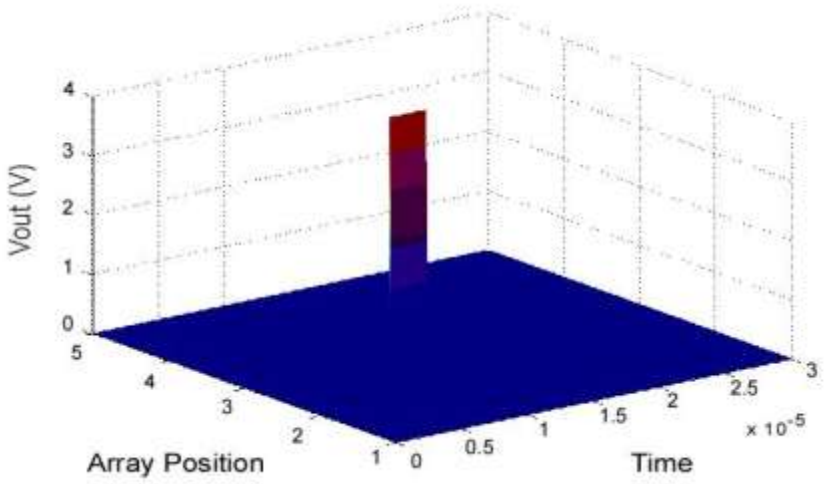


Figure 3-3 5 nodes delay line and sound angle of 60^{o}

C. Aquatic environment application

Echo sounding in continental waters has been worked on for some years, which has led to the development of portable instruments to process acoustic signals. The goal is to study the aquatic environment by measuring the fish population, to determine the nature of the lakebed and do bathymetric measurements. A precise and reliable system of localization is essential part of this project [29].

D. Wireless sensor networks application

Wireless sensor networks consist of a large number of inexpensive wireless sensors deployed in a geographical area, which has the ability to communicate with each other within a limited radio range. Wireless sensor networks are playing a more and more important role in a

range of monitoring applications in civil and military scenarios. The localization of the sensors is often very useful in the applications of sensor networks [30].

E. Magnetic field source application

Localization of a magnetic field source can be applied in motion capture system, magnetic tracking, crack detection, ubiquitous computing, and so on. Especially, in the last one, radio frequency identification (RFID) tag localization attracts great attention, because it realizes interfaces with location awareness such as haptic display and navigations for blind people/mobile robots [31].

3.3 Other methods of doing localization

A common method of localization is by sound source localization. Sound localization [32][33][34][35][36] is the process of determining a sound source spatial location with multiple observations of the emitted sound signal. Generally speaking, most current sound localization techniques are based on the idea of computing the delay-and-sum energy of an array of observations, and then selecting the position in space which results in a maximum energy. This can be done with just two or more microphones [37].

The drawback of the existing techniques is almost all of them require significant amount of calculation, which is generally time and power consuming.

3.4 System Architecture

Figure 3-4 is the system architecture of arrays of integrate and fired neurons, which includes an $n \times 3$ matrix of neurons.

A microphone is used to convert sound to voltage, and then an operational amplifier is used to adjust this voltage to needed range. An input voltage V_{right} will generate a pulse V_{out} in the n^{th} neuron in the top line, and then this V_{out} will become the input voltage of the $(n-1)^{th}$, and generate another pulse, so it looks like a pulse is passing through the top line from the right to the left. Also, an input voltage V_{left} will generate a pulse (V_{out}) in the 1^{th} neuron in the low line, and then this V_{out} will become the input voltage of the 2^{th}, and generate another pulse, so it looks like a pulse is passing through the low line from the left to the right. When the two pulses reach the same column k, the middle k^{th} neuron will generate a pulse.

The location of a object K determines its orientation. For example, if K is right in the middle of the n columns, then the orientation angle of the object must be around 90^o, since V_{left} and V_{right} are generated in the same time; if K is in the left side of the n columns, then the orientation angle of the object must be larger than 90^o, since V_{right} is generated before V_{left}; if K is in the right side of the n columns, then the orientation angle of the object must be less than 90^o, since V_{left} is generated before V_{right}.

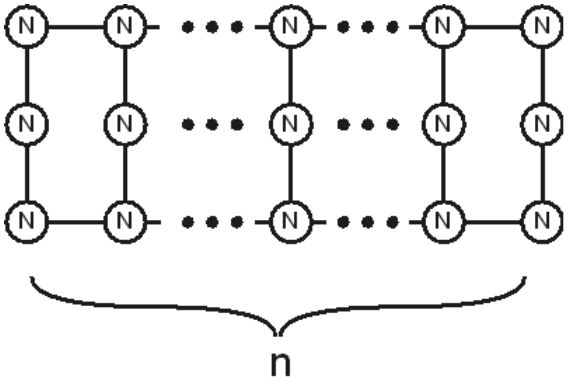


Figure 3-4 System architecture of arrays of integrated and fired neurons

Figure 3-5 shows the spatial description of this problem. Suppose a barn owl's ears are located at the points “B” and “C”, while the center is “O” (*BO=CO=h*). An object is located in “A”, while the orientation angle respected to “O” and “BC” is α and the distance is *d*.

Then the distance difference of the sound from that object to the two ears will be: $l_2 - l_1 = \sqrt{(h^2 + d^2 + 2hd\cos\alpha)} - \sqrt{(h^2 + d^2 - 2hd\cos\alpha)}$.

When *d*>>*h*, then we have $\frac{h}{d} \to 0$, so

$$\begin{aligned}
&\lim_{\frac{h}{d}\to 0}(l_2 - l_1) = \lim_{\frac{h}{d}\to 0}(\sqrt{(h^2 + d^2 + 2hd\cos\alpha)} - \sqrt{(h^2 + d^2 - 2hd\cos\alpha)}) \\
&= \lim_{\frac{h}{d}\to 0} \frac{4hd\cos(\alpha)}{\sqrt{(h^2 + d^2 + 2hd\cos\alpha)} + \sqrt{(h^2 + d^2 - 2hd\cos\alpha)}} \\
&= \lim_{\frac{h}{d}\to 0} \frac{4h\cos(\alpha)}{\sqrt{((\frac{h}{d})^2 + 1 + 2\frac{h}{d}\cos\alpha)} + \sqrt{((\frac{h}{d})^2 + 1 - 2\frac{h}{d}\cos\alpha)}} \qquad \text{3-1} \\
&= \frac{4h\cos\alpha}{\sqrt{(0+1+0)} + \sqrt{(0+1-0)}} \\
&= 2h\cos\alpha
\end{aligned}$$

So the distance difference from the object to the two ears is 2hcosα which is independent of *d*. So we can use the time difference of reaching the two ears to tell the orientation angle α. That is the starting point of this work.

The previous work about sound localization is related to this work (current to digital conversion) in: they both use delay line to digitalize frequency information. In this way, there is no algorithmic involved, and also the conversion is completely asynchronous.

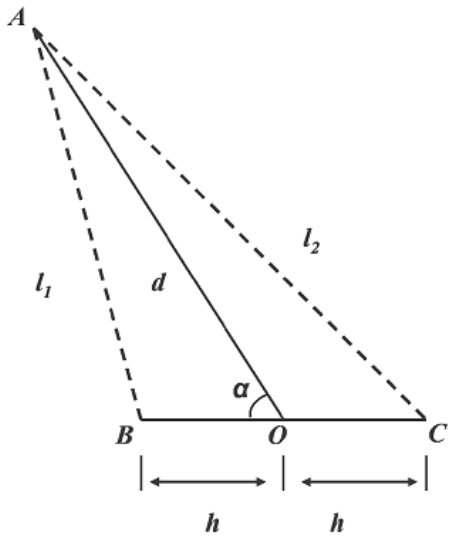


Figure 3-5 Position map for sound localization

Chapter 4

Delay Line Structure and operation

4.1 Biological Neurons

The basic anatomical unit in the nervous system is a specialized cell called the neuron. A view of a typical neuron is shown in Figure 4-1.

Many extensions of the single cell are long. The classical neuron has a tree of filamentary dendrites that aggregate synaptic inputs from other neurons. The input currents are integrated by the capacitance of the cell until a critical threshold potential is reached, and then an output is generated in the form of a nerve pulse, which is called an action potential. This output pulse propagates down the axon, which ends in a tree of synaptic contacts to the dendrites of other neurons [16].

An action potential is the product of many different types of current, which interact with each other. A typical action potential from the snail helisoma trivolvis is shown in Figure 4-2 [15]. Note that the rise from resting potential to threshold voltage, although threshold can be clearly seen, is omitted from this graph.



Figure 4-1 A view of a typical neuron [14]

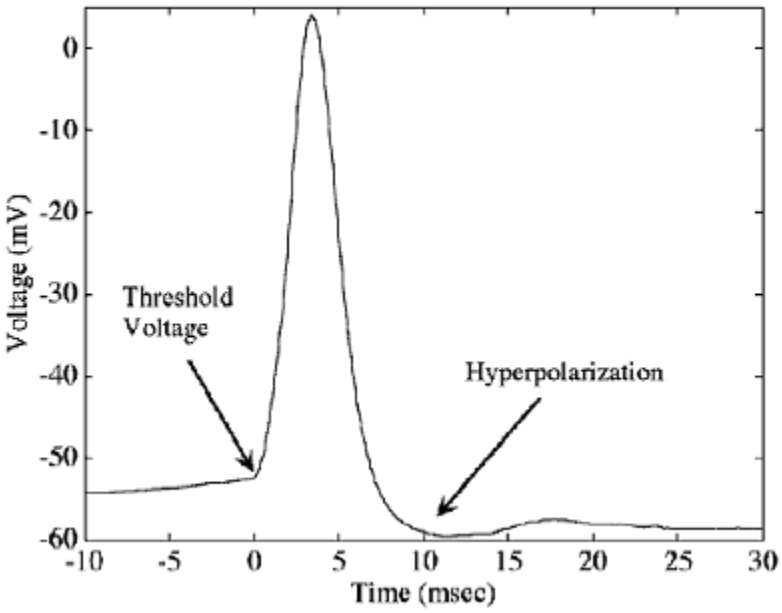


Figure 4-2 Real action potential from the invertebrate snail (helisoma trivolvis) [15]

4.2 Integrate and Fire Neuron

There are many circuits that model the behavior of neurons. One family simple model is called an integrate and fire neuron model. A schematic of a complete integrate and fire circuit for generating multiple action potentials (n-type) is shown in Figure 4-3 (a). The current-control transistor Q_7 limits the time at which the circuit spikes and, after that, the current-control transistor Q_1 limits the rate at which the circuit recovers when the output is high, and hence limits the pulse duration [16].

Figure 4-3 (b) is the p-type of this circuit. The difference is that, n-type neuron is used to generate a pulse jumping from ground to V_{dd} and then back from V_{dd} to ground, while the p-type neuron is used to generate a pulse jumping from V_{dd} to ground and then come back.

Here, an n-type integrated and fire neuron is discussed how it works.

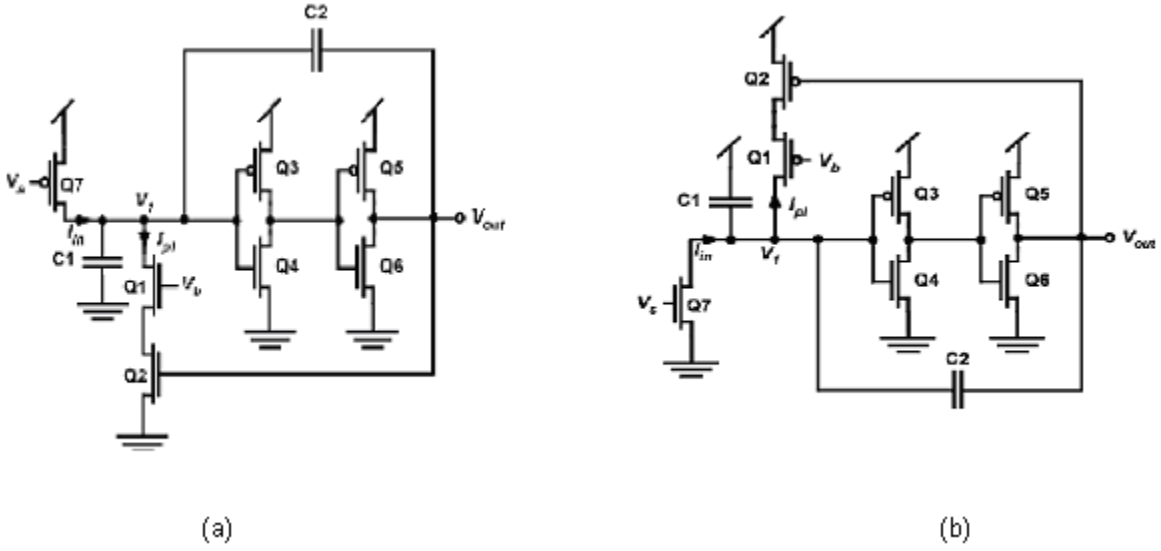


Figure 4-3 Integrate and fire Neuron [16]. (a) n-type (b) p-type

At the beginning, both V_1 and V_{out} are at ground and V_s is at V_{dd}. When V_s changes from V_{dd} to some value which is less than V_{dd}, the current begin to flow through Q_7. C_1 is charged by this current I_{in}, and V_1 steadily increases until some threshold voltage is reached. At the threshold voltage V_{out} jumps to V_{dd}. By leaking some charge away to ground through Q_1 and Q_2, the neuron can be turned off and V_{out} jump back to ground again. Q_1 is a current-control transistor, and used as a pulse-duration control to decide the time length of a pulse. Once the output is on, transistor Q_2 is on, and C_1 start to discharge and lower V_1 node through Q_1 and Q_2. The more Q_1 is based to be on, the faster leak charge away from C_1, and the circuit will turn itself off sooner. The positive - feedback sequence is initiated again once V_1 reaches V_h. A small decrease in V_1 results in a large decrease in V_{out}, which results in an even further decrease in V_1, so the output is driven to ground, and the nerve pulse is terminated [16].

Simulation result of a typical V_1 and V_{out} of an integrated and fire neuron (n-type), is shown in Figure 4-4, with C_1*= 20e-15F*, C_2*= 20e-15F*. When current I_{in} is fed, V_1 first increase steadily until the threshold voltage is reached, then V_{out} jumps to high, and V_1 jumps with it too. After that, V_1 decreases because of the leaking current I_{pl}, which is activated by V_{out}. When V_1 decreases to the threshold voltage, V_{out} jump to low, and V_1 jumps with it too. With I_{in} is continuously supplied, this process can repeat continuously. Figure 4-5 is a simulation result of a typical V_1 and V_{out} of a p-type integrate and fire neuron.

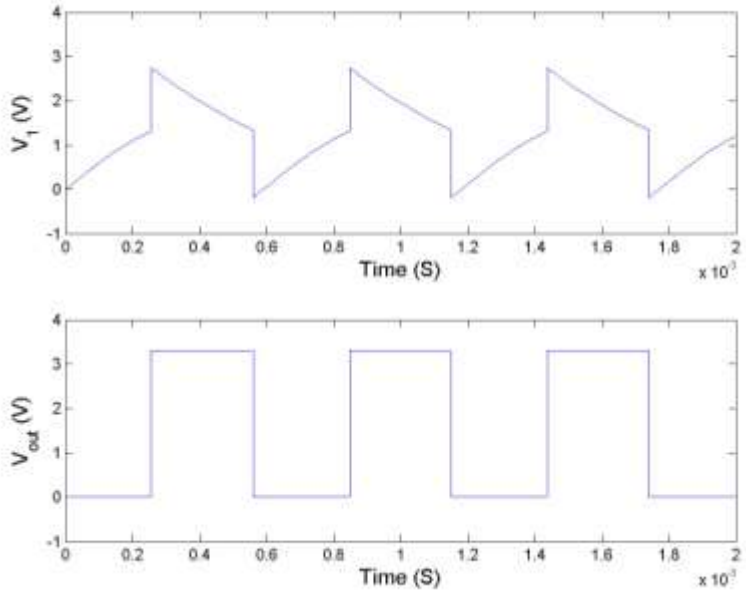


Figure 4-4 Simulation results of typical V_1 and V_{out} of n-type integrate and fire neuron.

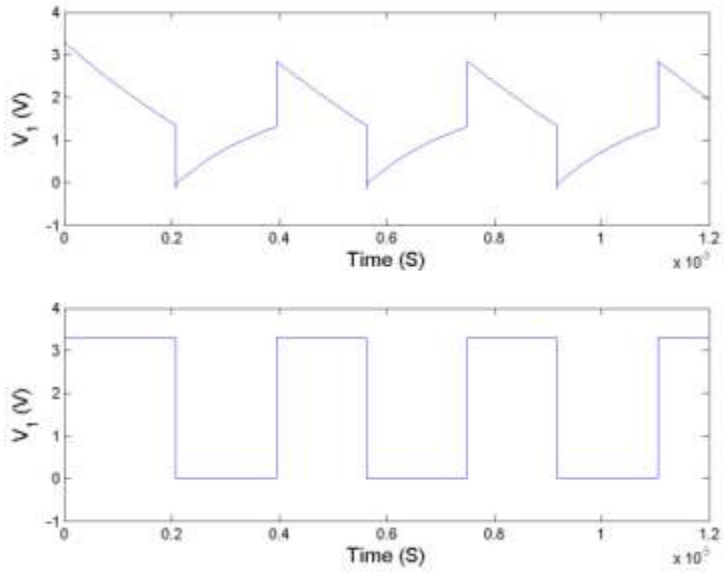


Figure 4-5 Simulation results of typical V_1 and V_{out} of p-type integrate and fire neuron

4.3 Delay Line

The delay line's structure is shown in Figure 4-6 [16]. It includes two adjacent delay line cells. N-type integrate and fire neurons are assumed in the delay line cells. Each delay line cell get inputs from its previous delay line cell. Each delay line will delay its input to some extent. Assume each delay line cell has the same bias voltage V_{bias}, then each delay line cell will delay the input for the same time amount. In this condition, the delay time of each V_{out} is equal to its spike time. Any given cell cannot reset until both it and its next cell have fired. This assures propagation.

Let's use experimental results to look closely at how the delay line works.

Figure 4-7 is the experiment result showing V_1 and V_{out} of one cell in the delay line. Once a delay line cell receives input from the previous cell, V_1 will start to increase, until it reaches the threshold voltage, then V_{out} jump to V_{dd}. V_{out} stays in V_{dd}, and will not be discharged until its next cell gets enough input and its V_{out} jumps to V_{dd}.

Figure 4-8 shows the experiment result of V_{out} of two adjunct cells in the delay line. It shows that once the next delay line cell spikes, then the current cell's V_{dd} will be dragged down. Because the same bias voltage is used, all the delay line cells need the same amount of time to spike. Once it spikes, it will hold until the next cell spikes.

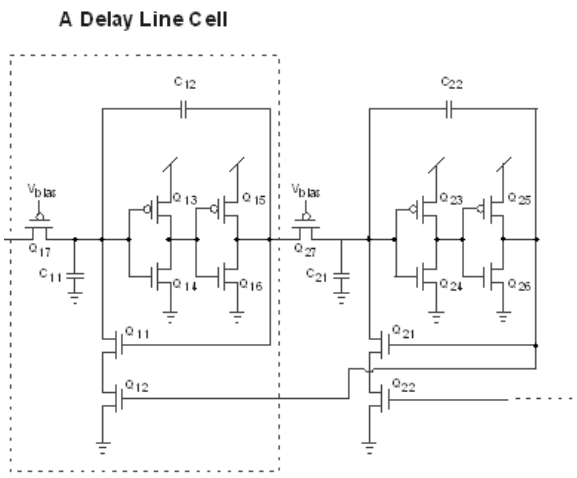


Figure 4-6 Delay line structure [16]

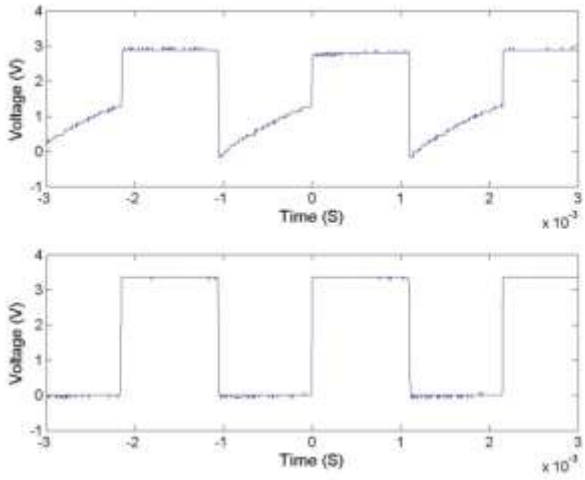


Figure 4-7 Experiment result of V_1 and V_{out} of one cell in the delay line

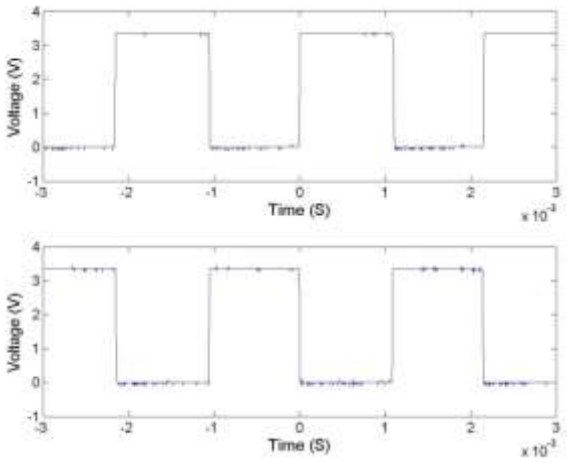


Figure 4-8 Experiment result of V_{out} of two adjunct cells in the delay line

Chapter 5

System Structure and Operational Principle

5.1 System Architecture

A block diagram of this circuit is shown in Figure 5-1, which includes an integrate and fire neuron to convert current to frequency (N), an inverter, a counter, delay lines cells (D), AND logic gates, encoder and programmable logic.

Modified integrate and fire neuron is used at the front end of the ADC for current to frequency conversion purpose.

5.2 Modified integrate and fire neuron

The modified integrate and fire neuron circuit diverges from the basic integrate and fire neuron circuit [19] in that it generates a pulse which has a fixed duty cycle over a wide range of input currents and the frequency is proportional to the input current. In this way, the current signal can be converted to a frequency signal. Then the generated pulse will come to the input of the first delay line cell and the inverter at the same time.

Figure 5-2 is schematic of the modified integrate and fire neuron.

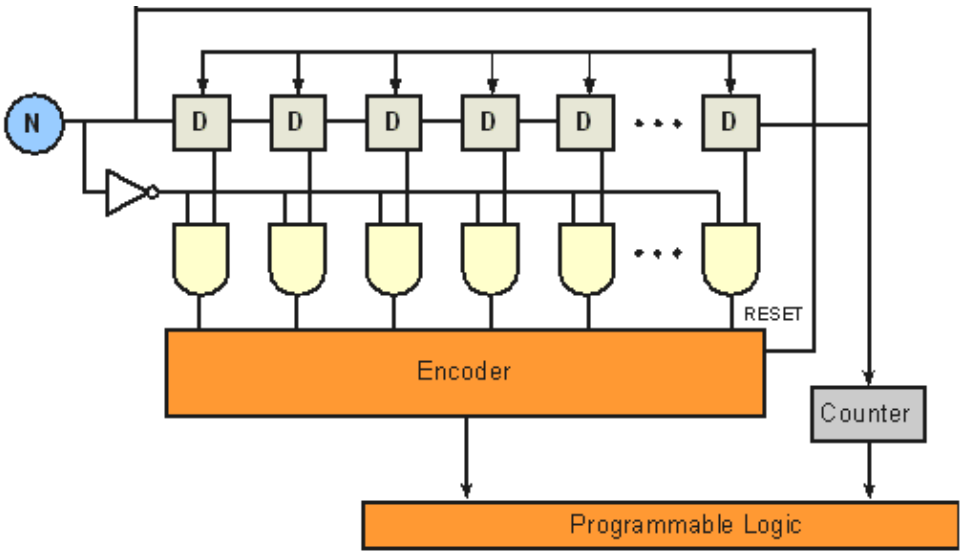


Figure 5-1 Block diagram of the current ADC

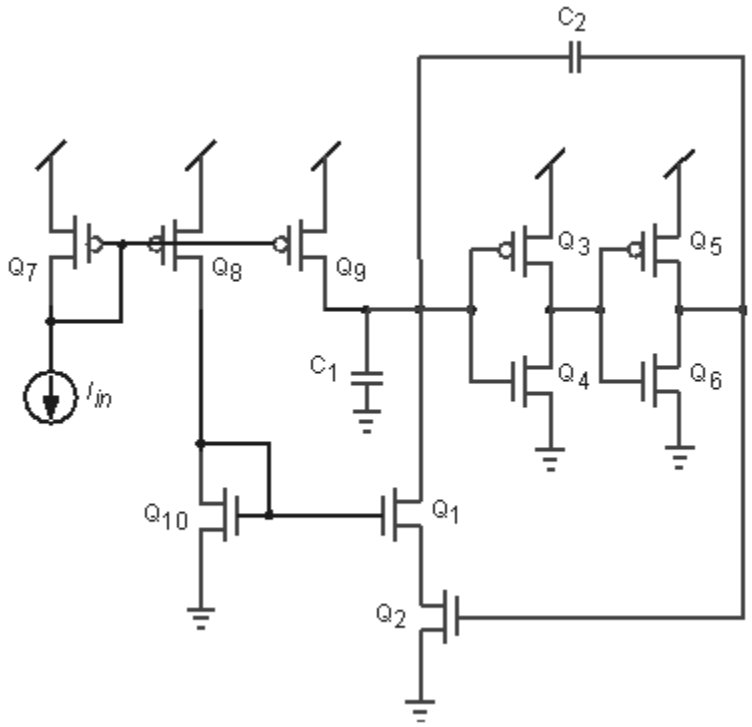


Figure 5-2 Schematic of the modified integrate and fire neuron

Current mirrors are used to control the duty cycle of integrate and fire neuron. For the basic n-type integrate and fire neuron, as shown in chapter 3, the frequency can be calculated by [16]:

$$t_{low} = \frac{C_2 V_{dd}}{I_{in}} \tag{5-1}$$

$$t_{hi} = \frac{C_2 V_{dd}}{I_{pl} - I_{in}} \tag{5-2}$$

$$f = \frac{1}{t_{low} + t_{hi}} \tag{5-3}$$

After setting

$$\left(\frac{W}{L}\right)_{Q_7} = \left(\frac{W}{L}\right)_{Q_8} = \left(\frac{W}{L}\right)_{Q_9} \tag{5-4}$$

$$\left(\frac{W}{L}\right)_{Q_1} = 2\left(\frac{W}{L}\right)_{Q_{10}} \tag{5-5}$$

Then we can get:

$$I_{pl} = 2I_{in} \tag{5-6}$$

This will result in:

$$f = \frac{I_{in}}{2C_2 V_{dd}} \tag{5-7}$$

After modification, the duty cycle is pretty much fixed and close to *50%*.

5.3 Operational Principle

Assume the generated pulse of the neuron circuit has a period of *T*, with a fixed duty cycle of *d*, and every delay line cell's delay time is *t*. *T* is inversely proportional to the input current, as a result of using the modified integrate and fire neuron circuit.

At time zero, the neuron circuit's output jumps from ground to V_{dd}. After a period of time *dT*, it jumps back to ground, and the inverter's output jumps from ground to V_{dd}. Assume $kt \leq Dt < (k+1)t$, then k^{th} delay line's output is V_{dd} at this time, and it's also the furthest delay line whose output is V_{dd} at this time, which results in the k^{th} AND logic gate's output going high. This V_{dd} will be detected by the encoder, and it will generate a reset signal to reset all the delay lines. Also it will send this information (the array position, *k*) to the programmable logic to determine the current's value.

If the pulse goes through all the delay lines and the output of the inverter has not jumped to V_{dd} yet, $dT > nt$. The last delay line's output will feed back to the first delay line cell (in a loop), and a counter will add "1". This loop can continue working and the counter's number will continue adding "1" for each loop, until the inverter's output jumped to V_{dd}, then the output of a AND logic gate jump to V_{dd}. The counter's output number and the AND logic gate array position are sent to the programmable logic to determine the current's value.

The resulting current and array position relationship of this circuit is non-linear with identical delay lines; however, by varying the parameter of the delay lines, a linear resolution can be achieved.

This circuit avoids complex computation, which uses time and energy; avoids complex hardware blocks such as operational amplifiers, conserving space; and also, it can be used to detect very low current. In addition, the whole circuit is clean and highly modular.

Figure 5-3 shows propagation of a signal down a delay line. When there is an input current, it will cause the modified integrate and fire neuron spike, which will then cause the delay line cell to spike. Array position 1 is the modified integrate and fire neuron. Array Position 2-10 is the delay line, while the last one is different. Array position 2-9, is also referred as "delay line array position 1-8" later.

Figure 5-4 shows a single delay line biased such that it produces different conduction velocities. Delay lines with different conduction velocities can be used to satisfy the requirement of different input current range.

With different input currents, different AND gates should output a high signal. Figure 5-5 - Figure 5-7 illustrates the propagation of a signal down the delay line (top plot) and the resulting output of the AND gate array (below plot). Input current 1 is larger than input current 2, and input current 2 is larger than input current 3.

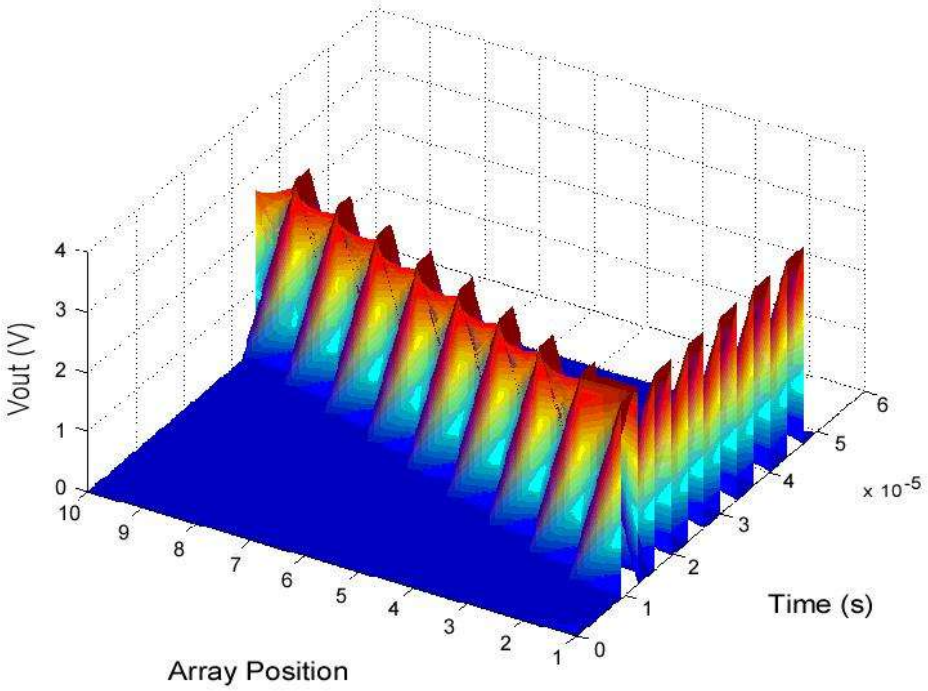


Figure 5-3 System progress

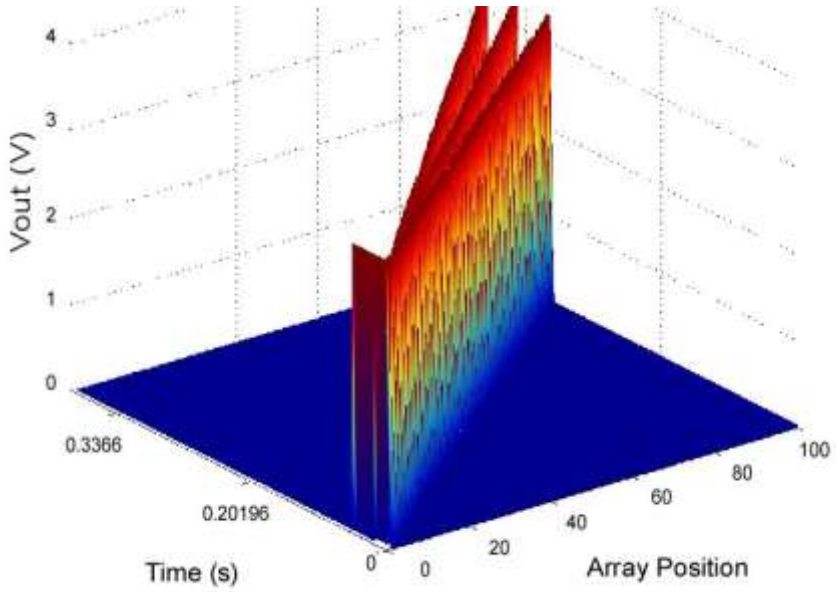
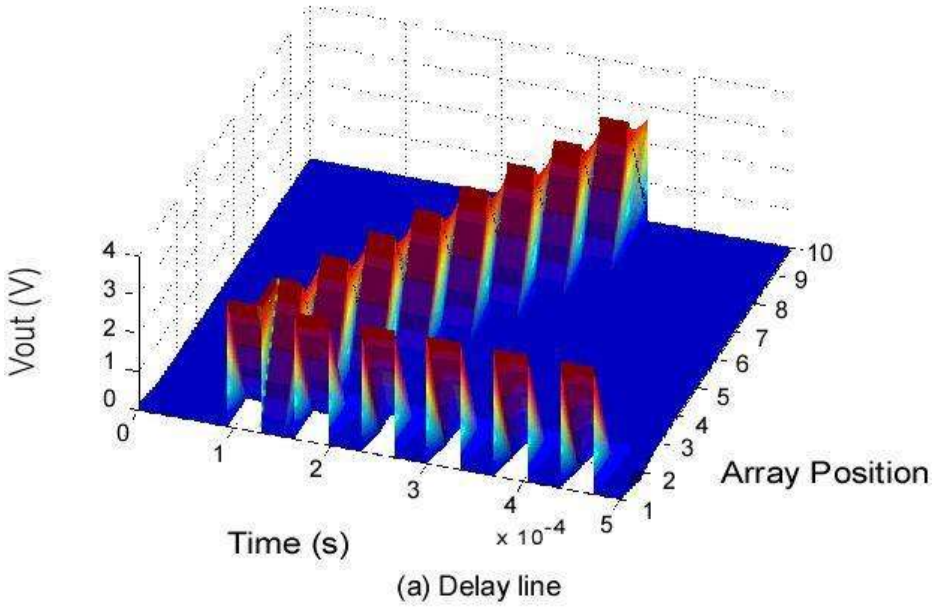
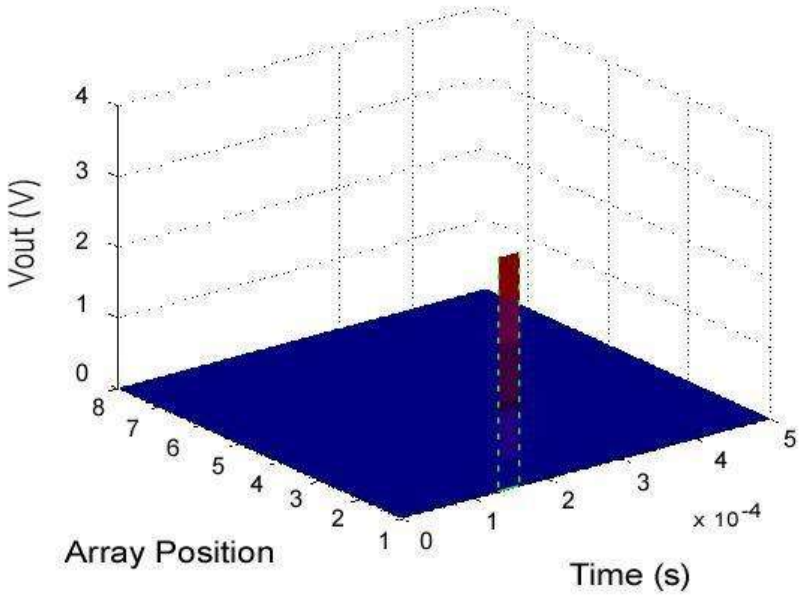


Figure 5-4 Delay lines with different transport speeds

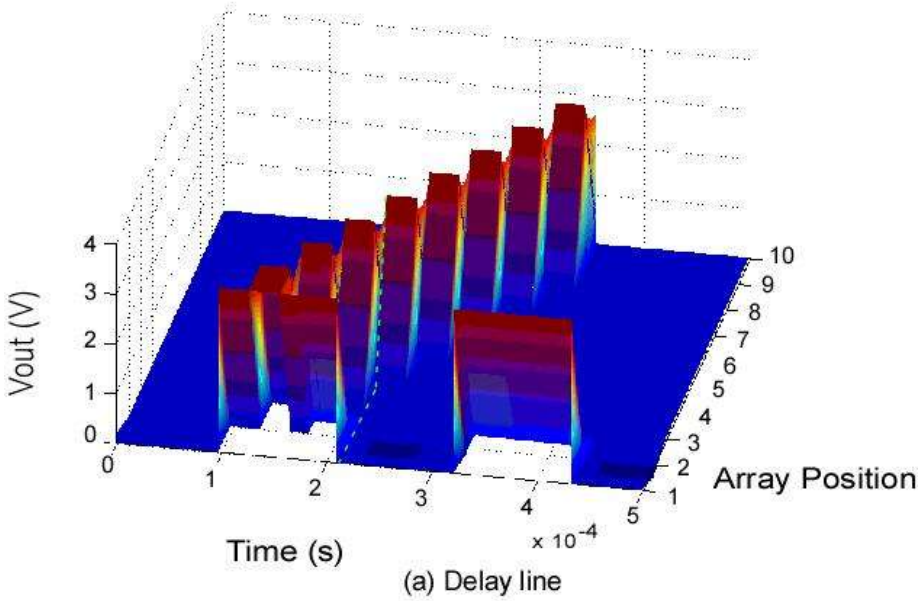


(a) Delay line



(b) AND output position

Figure 5-5 Input current 1: *1st* delay line spikes



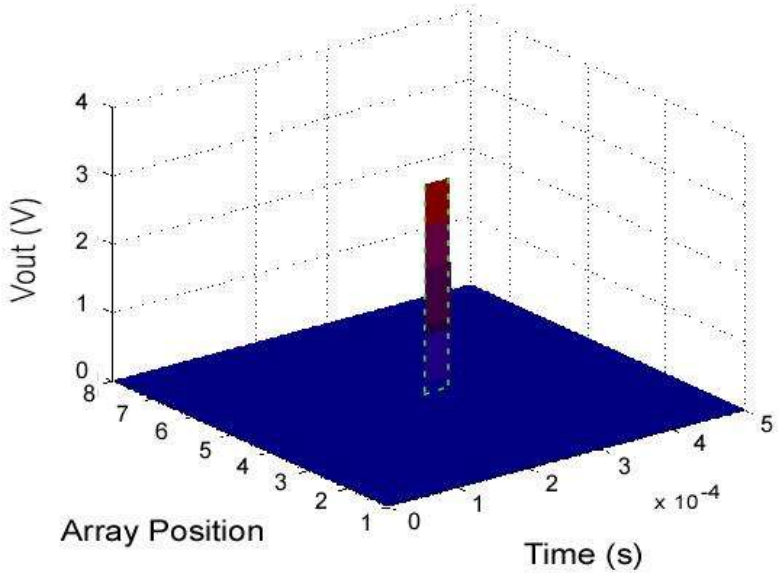
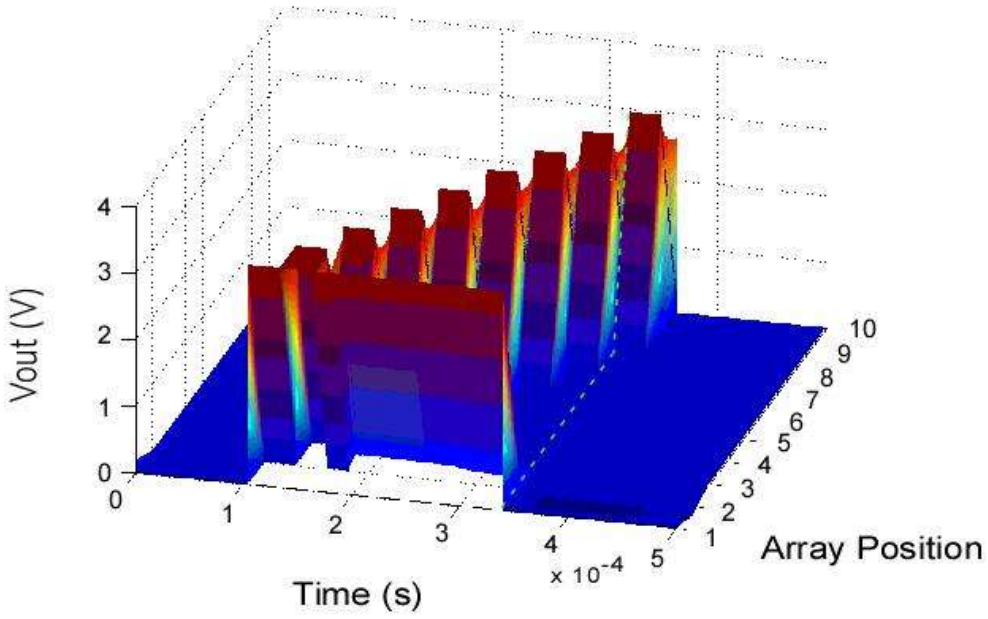
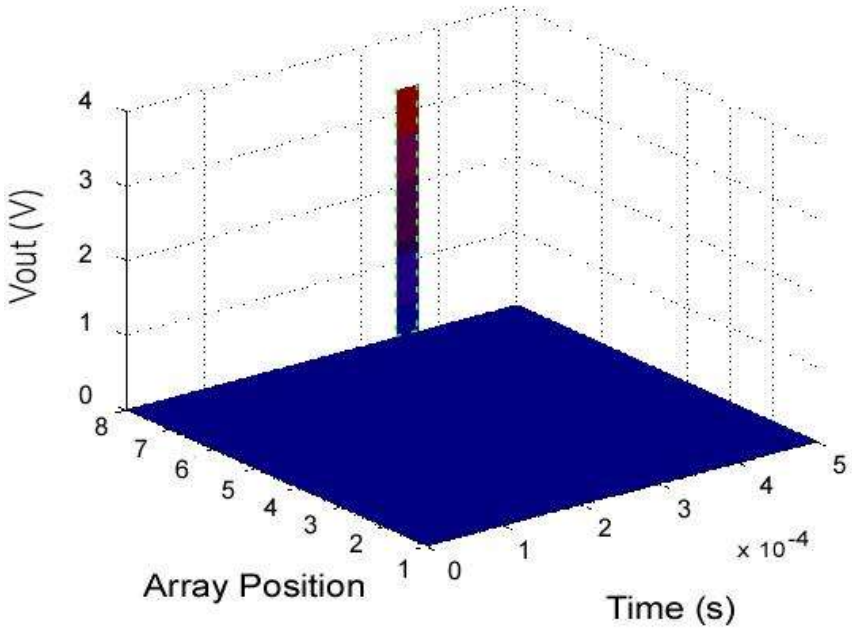


Figure 5-6 Input current 1: 4^{th} delay line spikes



(a) Delay line



(b) AND output position

Figure 5-7 Input current 1: 8^{th} delay line spikes

Chapter 6
Proof of Concept

6.1 Size and its Improvement

In the original conception of this design, the number of delay cells determines the number of bits of accuracy. However, this lead to significant scaling issues. Very large area usage is needed for higher resolution (e. g. 8 bit resolution needs 256 delay cells, 16 bit resolution requires 65,536 cells.)

The design is an attempt to alleviate the scaling issues. This new block diagram can be found in Figure 6-1. Essentially what this does is make use of the symmetry of the blocks and loop the delay line back on itself. One can now increase this converter's resolution, by adding a digital counter to count the number of times through the loop. To keep with an 8 bit example, the delay line is reduced to 32 nodes and now includes a 3 bit counter. This has the benefit of not only reducing the area used by the design, but it also reduces the static power used because fewer transistors results in smaller leakage at rest [1].

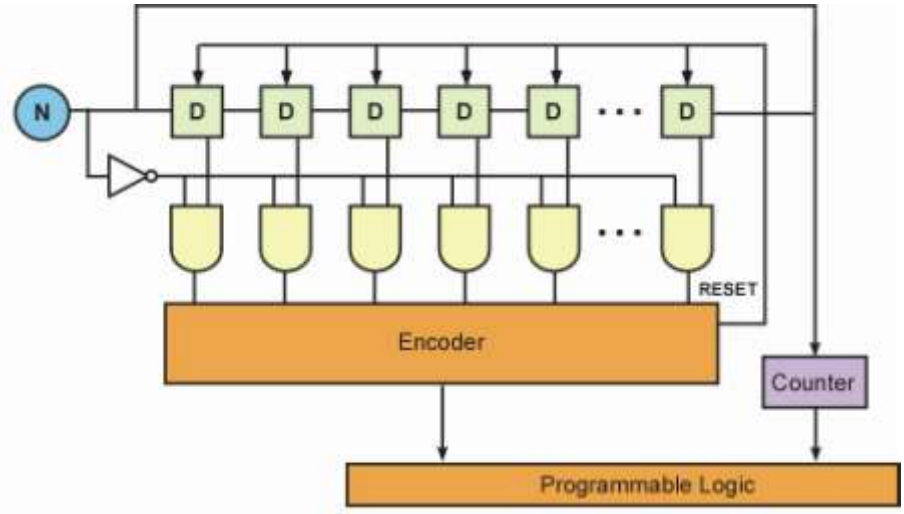


Figure 6-1 Area optimizing design

Figure 6-2 shows the layout of a flat 8-bit ADC (256 cells) and one with a 3-bit counter (32 cells, 3 bit counter).

With a flat 256 cells, the design uses an area of *250um* × *1000um* in a *0.5um* process available through MOSIS. With 32 nodes and a 3-bit counter, the design is reduced to an area of *40um* ×*1000um*. Significant area is saved through the use of a counter while sacrificing only a small amount of complexity. It was determined that this trade-off was worth the cost.

Figure 6-3 illustrates the system operation principle. A current is fed into the modified integrate and fire node (N). This node generates a pulse whose width is dependent on the magnitude of the input current. We define this width as T/2. Once N spikes, the first delay line cell will "see" an input voltage, which will result in a spike of width d. The width of this pulse is determined by the bias voltages on the delay line. It should be noted that these widths are designed to be significantly shorter than N.

The spike of the first delay line cell then feeds the second delay line cell, which will also result in a spike of width d. In this way, the remaining delay line cells will be fed one by one until the output of the modified integrate and fire neuron (N) falls to ground. Assuming that this occurs during the period of the kth delay cell's spike, then one can see T/2 = k d, and by the number of k, which is a number between 1 and n, the input current will be digitalized.

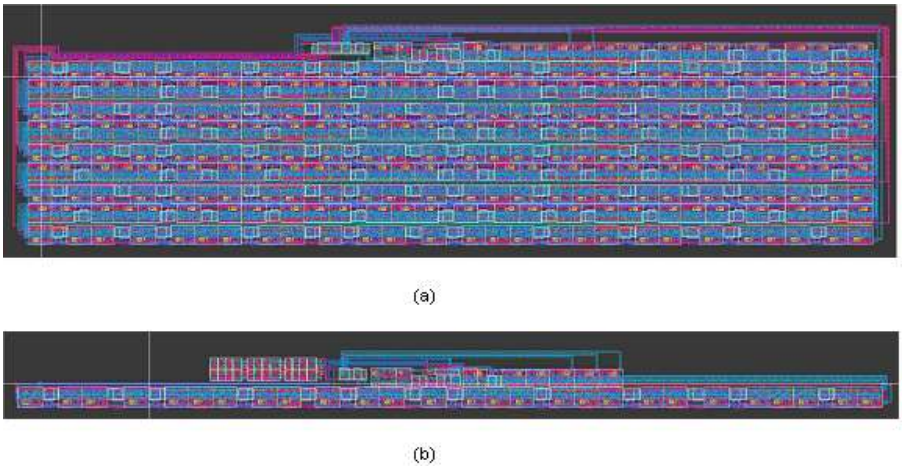


Figure 6-2 Layout picture of 8-bit ADC (a) 256 cells, (b) 32 cells and a 3 -bit counter

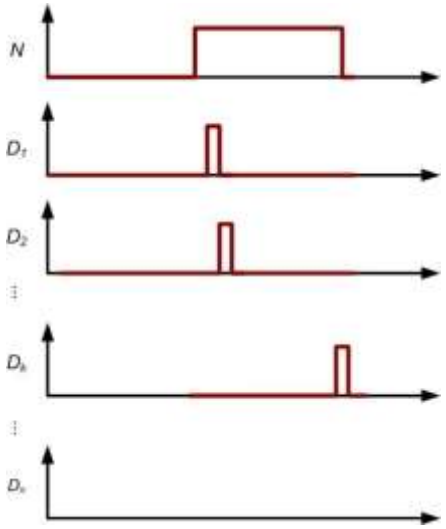


Figure 6-3 System operation principle

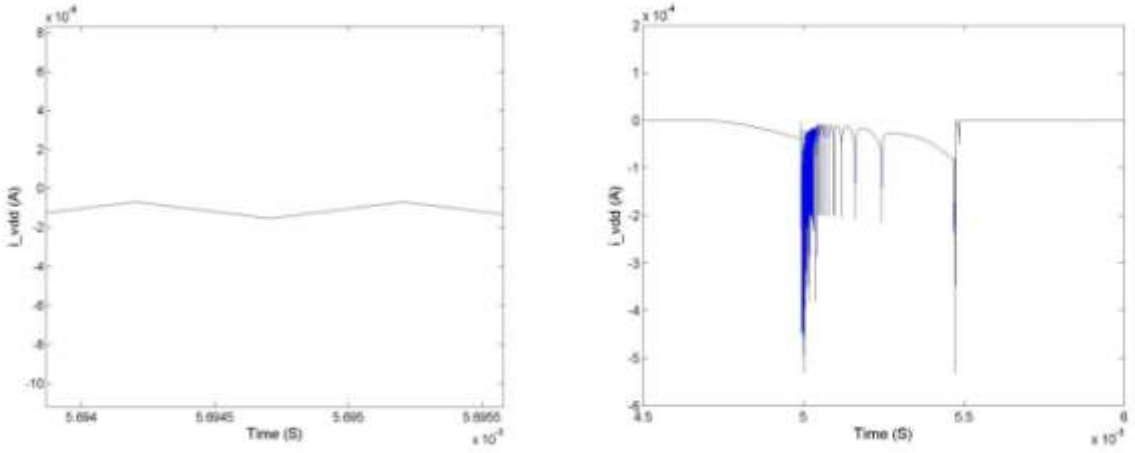
6.2 Power

This circuit uses an asynchronous design. That is to say that only the parts of the circuit that are actively involved in a conversion are using power at that instant. The majority of this circuit is the delay line. However, even in the delay line, only the current cell which is spiking is using the power, all the others are in a rest condition, which makes this circuit extremely power efficient. Figure 6-4 (a) is the current consumption for rest condition, and Figure 6-4 (b) is the current consumption for conversion condition.

6.3 Linearity

Figure 6-5 shows a typical input/output characteristic of ADC. Most of existing ADCs are designed to be linear, that is to say, those steps are designed to be equal to each other. This is not essential but typically easy to realize. According to a tutorial given by Dr. Keramat, the next generation of ADC will be nonlinear. For example when people speak, they sometimes speak in a high voice, and sometimes in a low voice. It is possible that one would not receive the same resolution for both conditions. One may prefer to a higher resolution when they speak in a soft voice, so that it can still be heard clearly [17].

In this ADC, the input current is proportional to the frequency of N, or, inversely proportional to the period of the output of N. This results in a *1/x* relationship between the input current and the delay line cell which determines the current.



(a) (b)

Figure 6-4 Current consumption for (a) rest condition (b) conversion condition

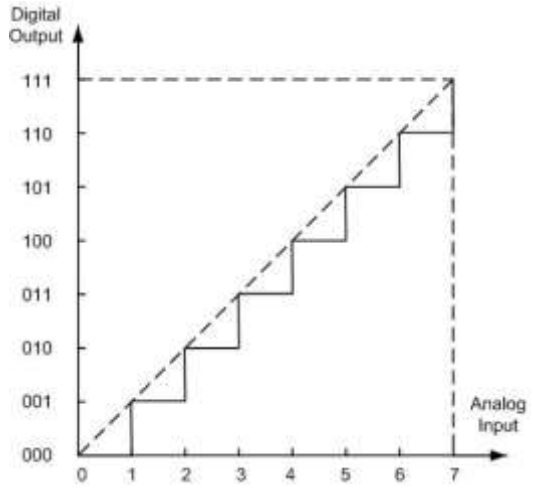


Figure 6-5 Input/output characteristics [17]

Figure 6-6 shows a linear ADC's desired current to array position curve.

Figure 6-7 shows the current to array position curve of this ADC would do under ideal conditions. It is nonlinear, and inherently could work as a next-generation ADC.

On the left of the graph one can see a region of high resolution which transitions to a region of low resolution.

Zoom in on Figure 6-7, one can see that this converter inherently has higher resolution when the current is high.

This could be a useful feature that could be exploited by a designer. For example, consider the left side of Figure 6-7, shown up in Figure 6-8. As the input current goes from *I* to *2I*, the converter goes through 2^7 array positions. This adds 7 bits of accuracy to this part so that it could be treated in this range as a 15-bit ADC.

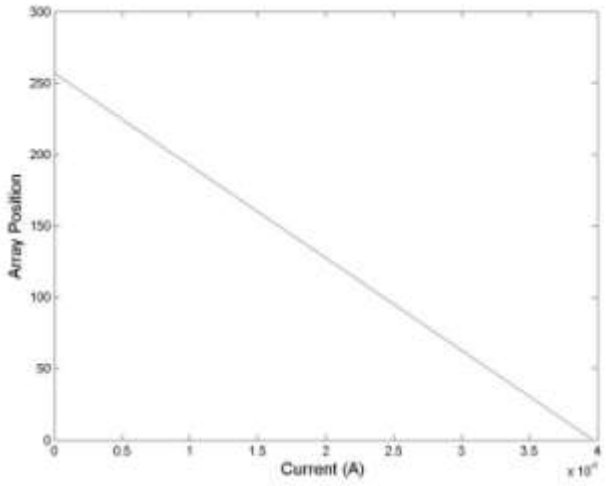


Figure 6-6 I-array position curve: linear resolution

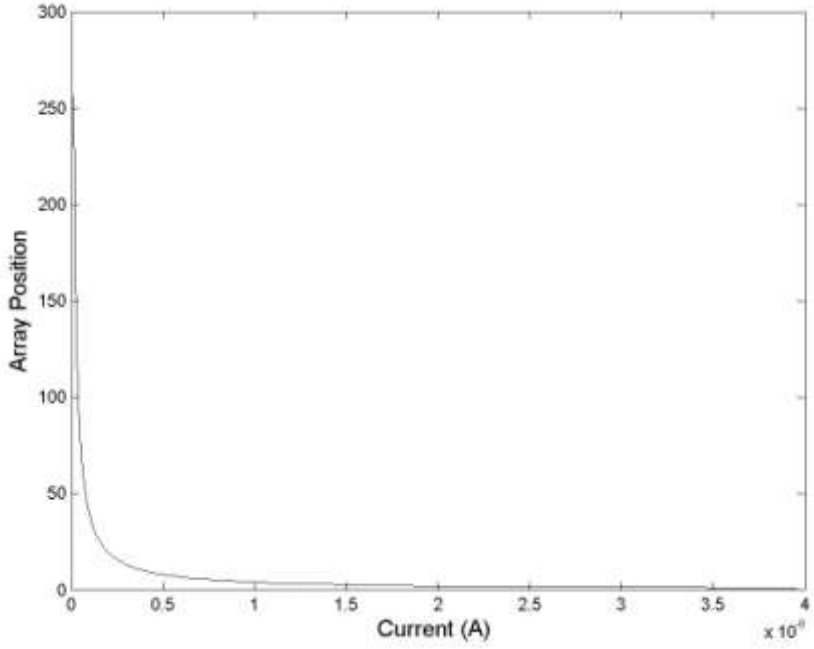


Figure 6-7 I-array position curve: this ADC

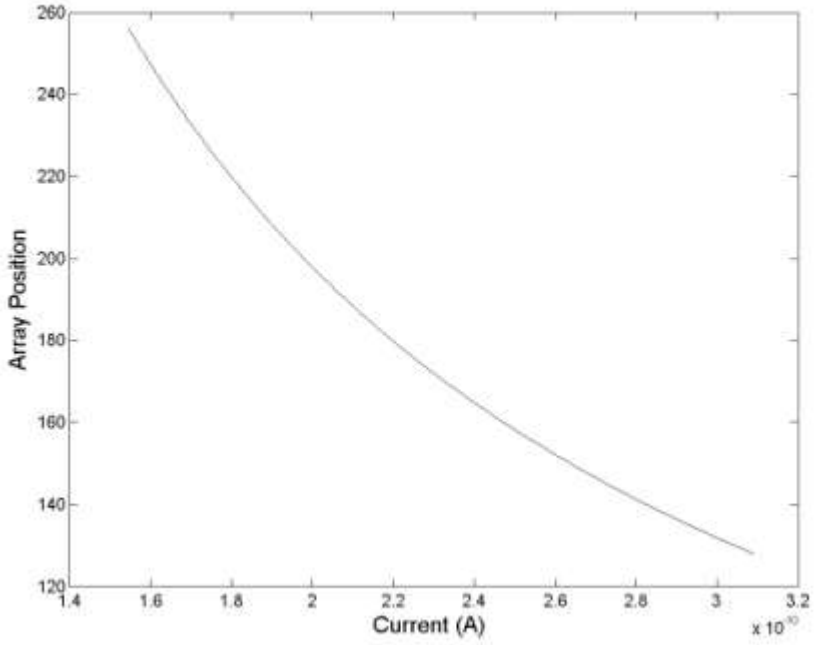


Figure 6-8 A part of the I-array position curve

6.3.1 Various Capacitors Solution

If one needs to make the ADC linear, there are several methods one might use to achieve this result. One such method is to use differently sized capacitors for each of the delay line cells. This has the effect of shifting the delay of each of the cells to some time value appropriate to linearize the system.

Assume that:

1^{st} array spikes, we have

$$I_1 = 256I \tag{6-1}$$

$$T_1 = \frac{T}{256} \tag{6-2}$$

where I_1 is the input current for 1^{st} array spike, and T_1 is the time period it will take to get the 1^{st} array to spike.

Then we get:

For m^{th} array spikes, we want:

$$I_m = (257 - m)I\,, \tag{6-3}$$

Therefore we need to have:

$$\begin{aligned} T_m &= \frac{T}{257-m} - \frac{T}{258-m} \\ &= \frac{T}{(257-m)(258-m)} \end{aligned} \tag{6-4}$$

where m=2, 3 ... 256.

So we can get:

$$\begin{cases} T_1 = \dfrac{T}{256} \\ T_m = \dfrac{T}{(257-m)(258-m)} \end{cases} \tag{6-5}$$

where m=2, 3 ... 256.

For each delay line cell, to make it spike, we have:

$$\frac{iT_m}{C} = V_{th} \tag{6-6}$$

where $C=C_1+C_2$ for the cell, V_{th} is the threshold voltage for V_1, and i is the I_{in}.

V_{th} is a constant, since the threshold voltage for V_1 is constant; i is a constant since we have the same bias voltage for each delay line to make equal I_{in}, so we can get:

$$\frac{T_m}{C_m} = CONSTANT\ . \tag{6-7}$$

And then

$$\begin{cases} C_1 = \dfrac{C}{256} \\ C_m = \dfrac{C}{(257-m)(258-m)} \end{cases} \tag{6-8}$$

where m=2, 3 ... 256.

Figure 6-9 shows comparison between nonlinear resolution, the desired ideal linear resolution, and post layout-extraction simulation results with various capacitors using Equation 6-7. One can see the linearization resolution is pretty good.

Table 1 shows the DNL and INL after capacitor-modification. Most of the DNL and INL are within ± 0.5 LSB range, which proves this way could effectively improve the linearity.

However, this method was not chosen as a viable method because it leads to non-standard cells in the delay line. This fact negates the ability to loop the delay line back on itself. This leads to a significant cost in terms of area used.

6.3.2 Floating-Gate Solution

A second method for linearization was to use floating gate circuits for the V_{bias} transistors of the delay line (Figure 4-6). Floating gates allow for different voltages to be stored at all the bias voltages without needing to provide numerous off-chip biases [45]. For equation 6-6, to get desired $t_{low} = \frac{C_2 V_{dd}}{I_{in}} T_m$, except making various capacitors, another way is to make various i, while keeping capacitor untouched. Then we will have

$$i_m T_m = CONSTANT\ . \tag{6-9}$$

Assume all in subthreshold region. Then:

$$i_m = \frac{W}{L} I_s \exp(\frac{kV_{gs}}{U_T}) \tag{6-10}$$

where K=0.6, U_T=0.026.

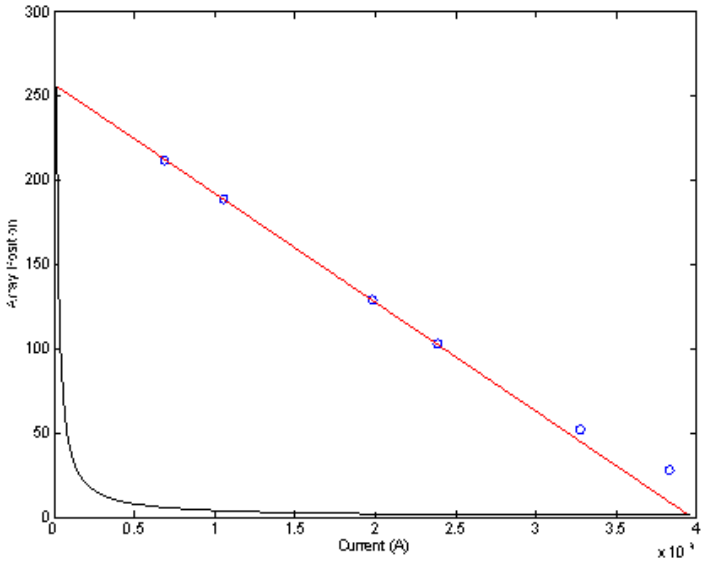


Figure 6-9 I-array position curve

Table 1 DNL and INL after capacitor-modification

INL	INL
	-0.6958
0.0412	0.2524
-0.0022	0.1230
-0.0132	0.4660
-0.1270	6.9417
0.4968	18.8641

From Equation 6-5,

We can have

$$\begin{aligned} \max(T_m) &= \frac{T}{2}, \\ \min(T_m) &= \frac{T}{255\times 256}. \end{aligned} \tag{6-11}$$

By Equation 6-10 and 6-11, we can get

$$|V_{gs}|_{\max} - |V_{gs}|_{\min} = \frac{U_T}{K}\ln(\frac{255\times 256}{2}) = 0.4508. \tag{6-12}$$

By that we can prove it's possible to bias all in subthreshold region.

And then by equation 6-5 and 6-9, we can get:

$$\frac{i_m}{i_1} = \frac{T_1}{T_m} = \frac{(256-m)(258-m)}{256}. \tag{6-13}$$

Bring equation 6-10 to 6-13, then get

$$\frac{\exp(\frac{K|V_{gs}|_m}{U_T})}{\exp(\frac{K|V_{gs}|_1}{U_T})} = \frac{(257-m)(258-m)}{256}. \tag{6-14}$$

This results in:

$$|V_{gs}|_m - |V_{gs}|_1 = \frac{U_T}{K}\ln(\frac{(257-m)(258-m)}{256}) \tag{6-15}$$

where m=2, 3 ...256.

Equation 6-15 can be used to determine the bias voltage from each delay line cell. Since it's almost impossible to have 256 bias voltages for one single chip, floating gate can be used to build this circuit easier. As mentioned before, a 3-bit counter can be added to decrease the area. So basically, the delay line will be gone through again and again, for up to 8 times, and every time it will use a different bias voltage V_{gs}.

Figure 6-10 shows an architecture using floating gates to get 8 different bias voltages. Multiple values are stored on multiple floating gate transistors. Each transistor is switched into the circuit on subsequent passes through the delay line, allowing for the delay of each cell to be different from each other and different on each pass through the delay line.

The problem with this solution is that it significantly increases the complexity of the design in order to allow for the use of a reduced delay line and counter. As a result, this method was also determined not to be optimum.

6.3.3 Translinear-Circuit Solution

In addition to the two ways already mentioned, a third way exists to linearize this circuit. It involves pre-distorting the input in a way that cancels out he non-linearities in the circuit.

Figure 6-11 illustrates a translinear circuit to realize $I_y = \frac{1}{I_x}$. T_1 and T_2 are identical translinear elements (TEs). An ideal TE has the infinite input impedance of a MOS transistor and also has a pure exponential current-voltage relationship. TE can be realized as an NPN bipolar transistor, or a subthreshold MOS transistor with its source and bulk connected together [38].

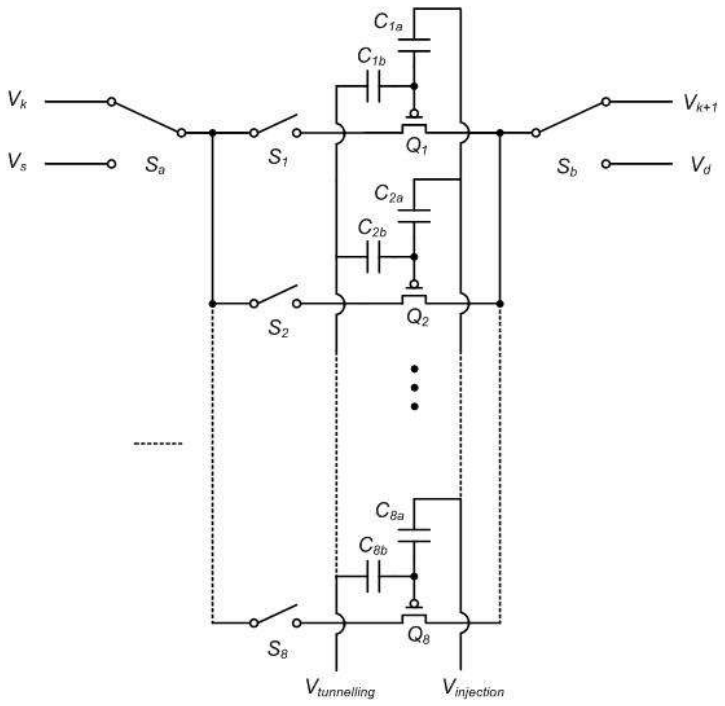


Figure 6-10 Floating gate architecture to get 8 different bias voltages [45].

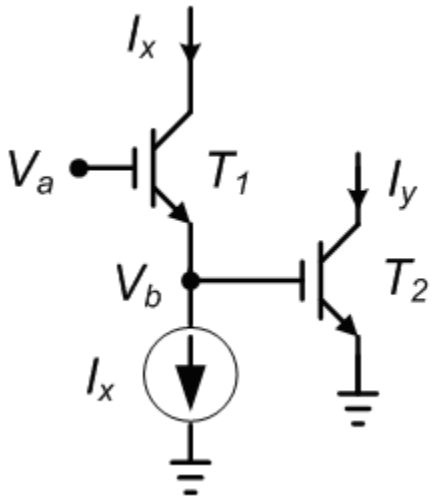


Figure 6-11 Translinear circuit [38]

Assume for T_1 and T_2, we have:

$$I_x = k_1 e^{k_2 (V_a - V_b)} \quad \text{6-16}$$

$$I_y = k_1 e^{k_2 (V_b)} . \quad \text{6-17}$$

So,

$$I_x I_y = k_1^{\,2} e^{k_2 (V_a)} . \quad \text{6-18}$$

Then $I_x I_y$ = *CONSTANT* for a given V_a.

$$I_x I_y = C_1 \quad \text{6-19}$$

where, C_1 is a constant value.

Since we already have

$$TI_y = C_2 \quad \text{6-20}$$

where T is the period of the modified integrate and fire neuron's output voltage, C_2 is also a constant value.

By the last two equations, we have

$$I_x = \frac{C_1}{C_2} T . \quad \text{6-21}$$

This indicates the input current Ix would be proportional to the period of the output voltage of the N circuit (Figure 6-1), so the circuit will be linearized from input to output by adding a nonlinearity to the front end that cancels the nonlinearity of the back end.

6.4 Prototype layout and simulation result

Figure 6-12 shows a micrograph of an early implementation of this design. Figure 6-13 shows some simulation results of the chip layout. Once the output of the *i2f* goes down to zero, it will be detected and the output of oa_1, oa_2, oa_3, oa_4, oa_5 will indicate which delay line cell spike, and the magnitude of the input current.

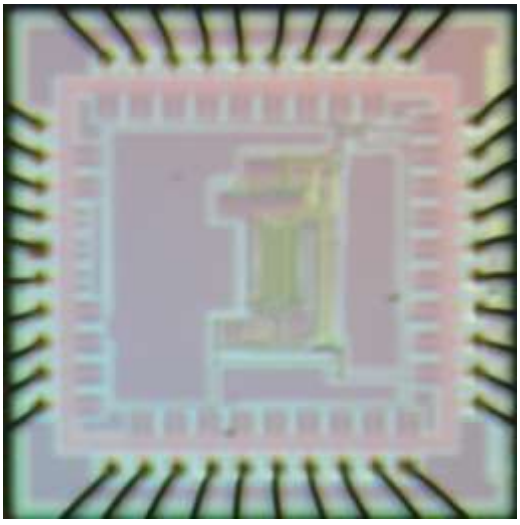


Figure 6-12 micrograph of the chip A

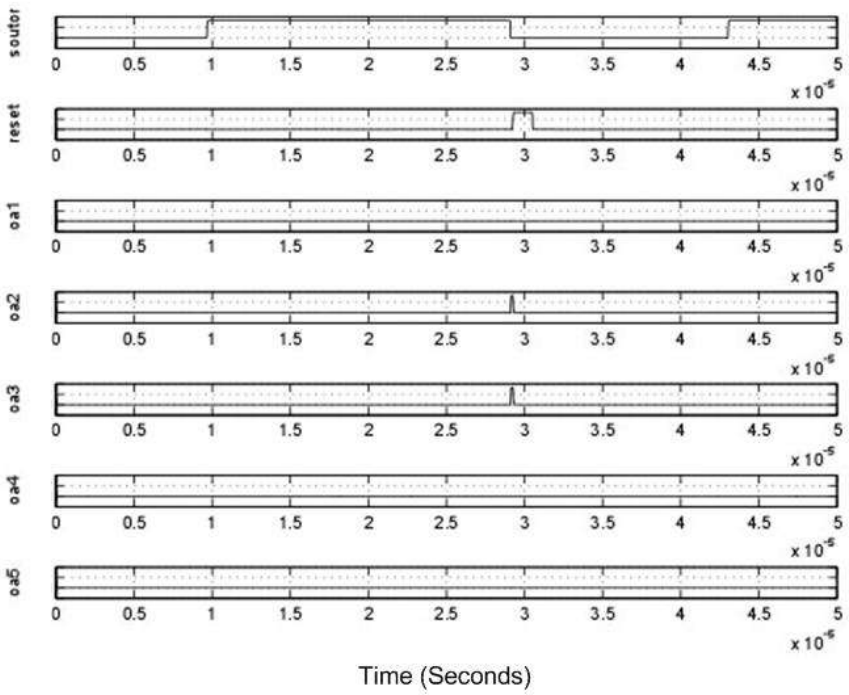


Figure 6-13 Simulation result of the chip layout

Chapter 7

Test Results and Analysis

7.1 Test setup

Figure 7-1 is micrograph of a later implementation this complete design. It was fabricated in a commercially available 0.5u process available through MOSIS. The significant difference between this and the earlier implementation is the addition of a translinear circuit, which enables the whole system to be tested. A transistor-level schematic is shown in Figure 7-2.

Since the ADC is fully realized in one single chip, a PCB was not needed to support the chip. A PCI 6259 is used to generate the input signal and a PCI 6713 to collect output voltage signal. For current meauses, a Keithley 6285 Picomameter was used. Also, an Agilent MSO6034A mixed signal oscilloscope is used to collect the frequency and duty cycle for voltage signal. A test setup is shown in Figure 7-3.

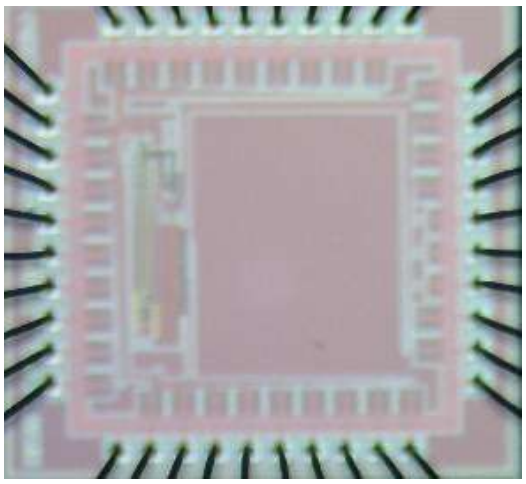


Figure 7-1 Micrograph of the chip B

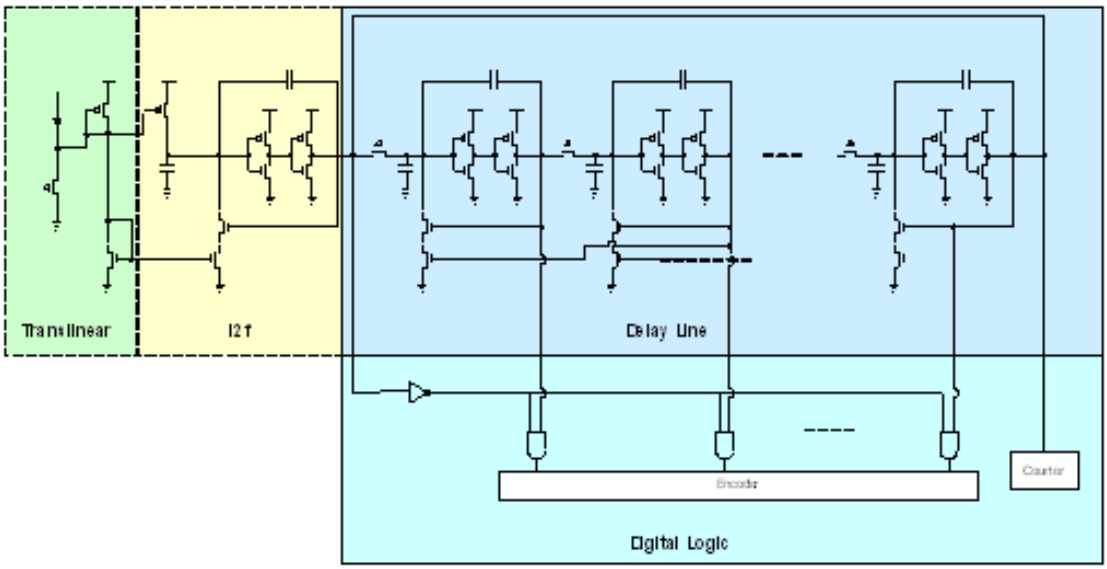


Figure 7-2 System structure for chip B

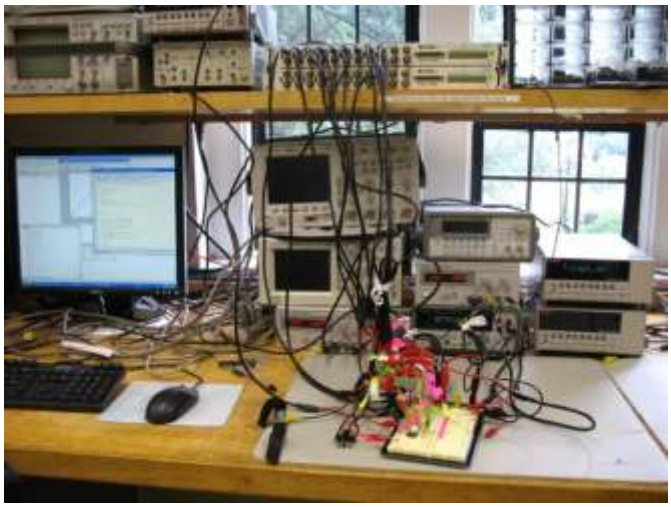


Figure 7-3 Test setup

7.2 Test results

Figure 7-4 shows the three sequential outputs of the delay line. The X-axis is time, and the Y-axis is the output of the delay line cells.

The output's pulses are right next to each other; when the first falls, the second one rises. This illustrates explicitly that there is only one delay line cell is spiking at a time yielding tremendous energy savings.

Figure 7-5 gives test results for the translinear circuit portionof the system. The X-axis is input current 1, and Y-axis is input current 2.

They should to be inversely related to each other. By multiply input current 1 with input current 2, as shown in Figure 7-6, one can determine how the inverse function is working. The output of the multiplication should be constant.

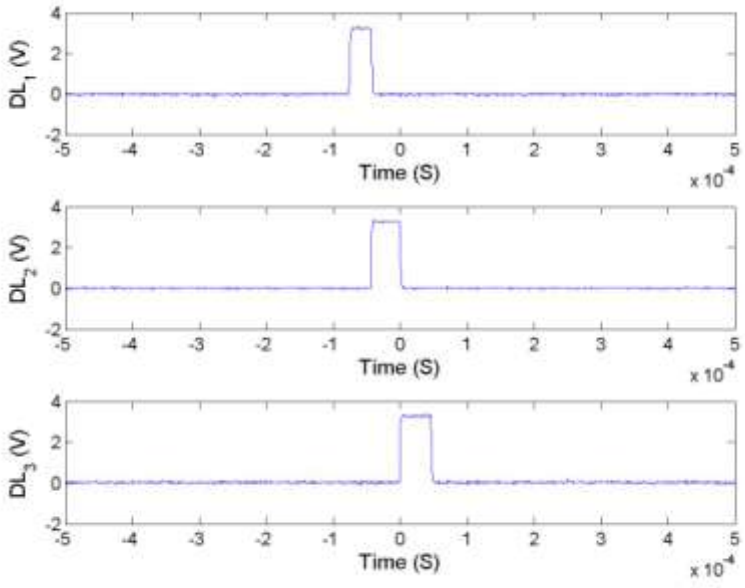


Figure 7-4 Delay line outputs

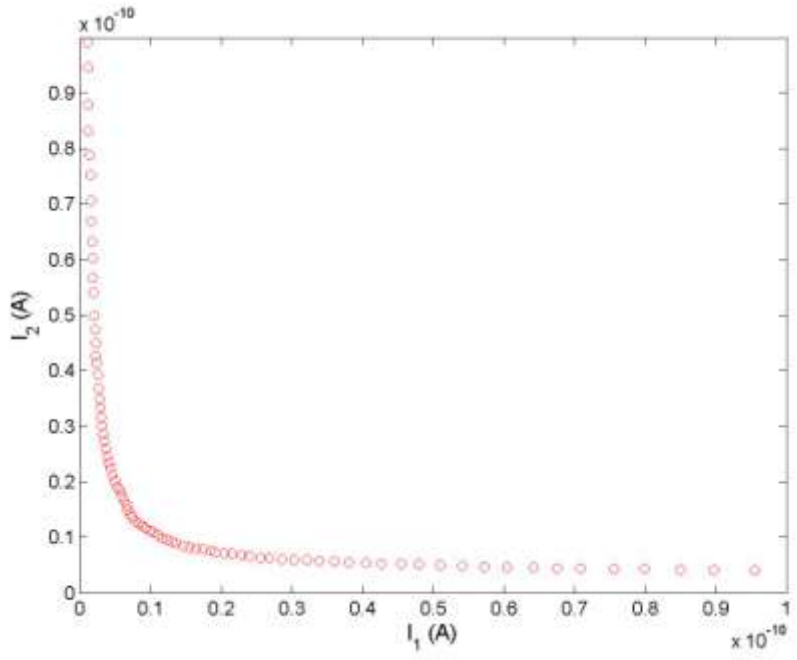


Figure 7-5 Translinear output I_1 versus I_2

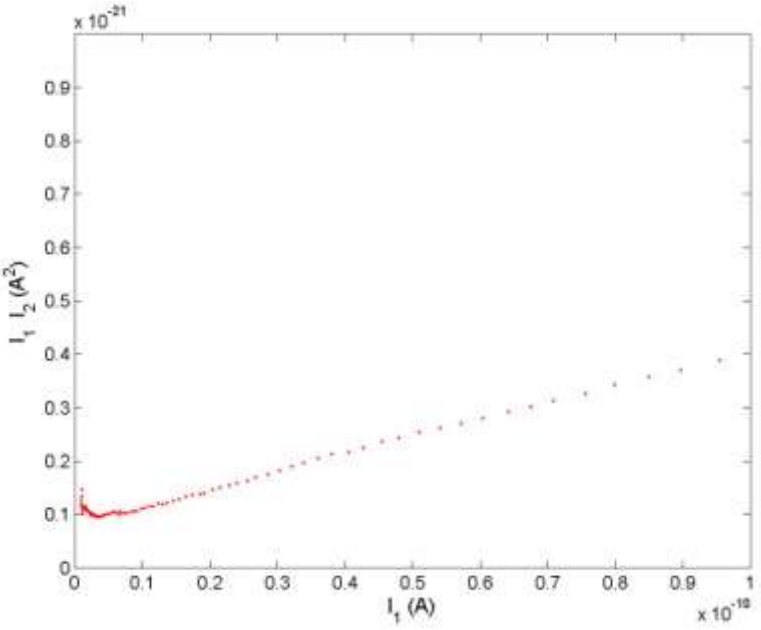


Figure 7-6 Translinear output I1 × I2

Figure 7-7 gives frequency versus input current of the modified integrate and fire neuron (N). The X- axis is the input current, and Y-axis is the frequency.

One would expect this result to be linear It can be clearly seen (in Figure 7-8) is very close to absolute linear.

Figure 7-9 shows the duty cycle again for the modified integrate and fire neuron for the same data shown in Figure 7-7. The X-axis is input current; Y-axis is the duty cycle.

While any duty cycle could be designed for, a duty cycle of 50% was chosen for this implement.

The key to this design is for the duty cycle to be constant. With a constant duty cycle, spike frequency will be proportionally related to the amplitude of the input current, thus to get he current to frequency conversion.

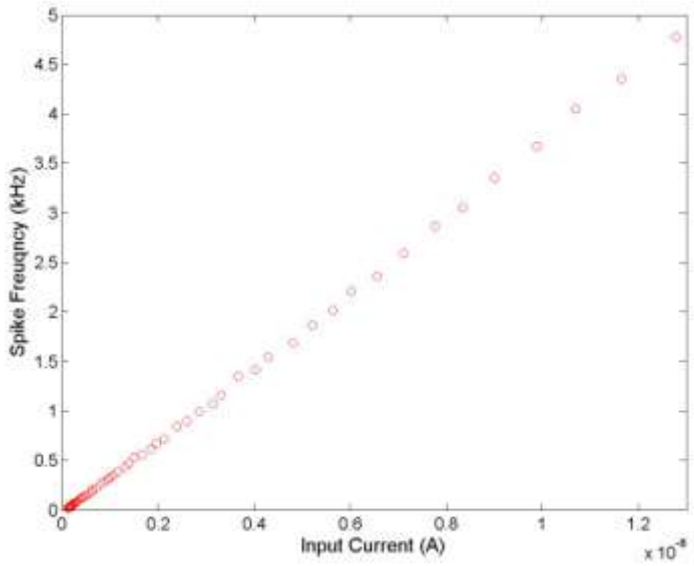


Figure 7-7 Frequency versus input current

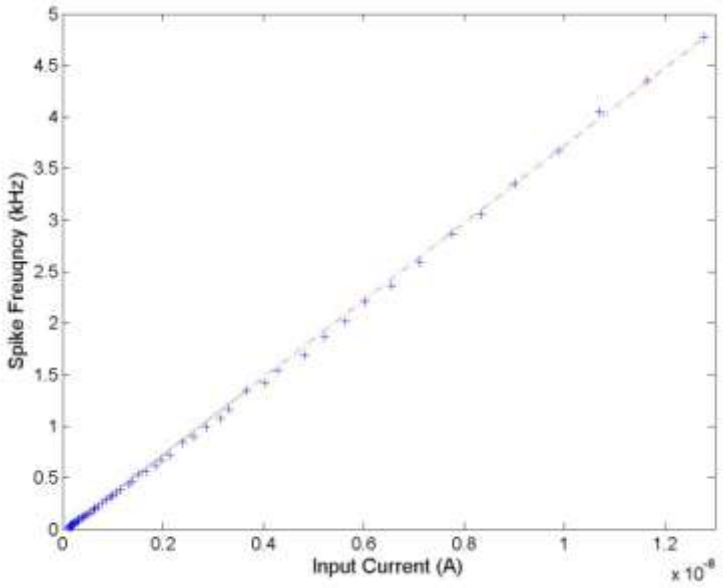


Figure 7-8 Frequency versus input current

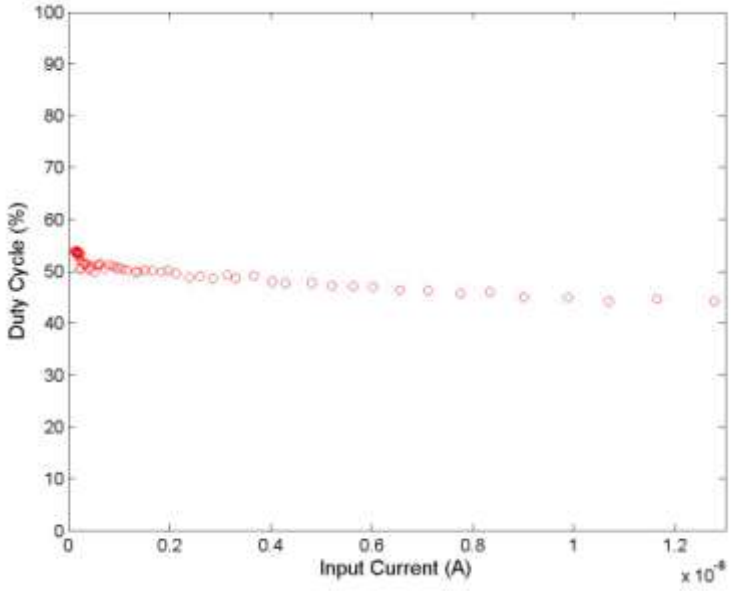


Figure 7-9 Duty cycle versus input current

Figure 7-10 gives period (1/f) versus ADC outputs. The X-axis is the pulse-up period, and Y-axis is the ADC outputs. They are also semi –linear to each other.

The nonlinearity was brought to this part since some of the output of delay cells were connected to the pins without buffering, which limits the uniformity of the delay line and thus limits the linearity of the system. This part could be improved by buffering any nodes of the delay line which will be brought out to pins.

Figure 7-11 is the current versus digital curve, which is the main goal of this project (converting current to digital). The X-axis is the input current and Y-axis is the output digital code.

Note that the conversion approaches the desired linear result. Measures of INL and DNL for this data can be found in Figure 7-12 and Figure 7-13.

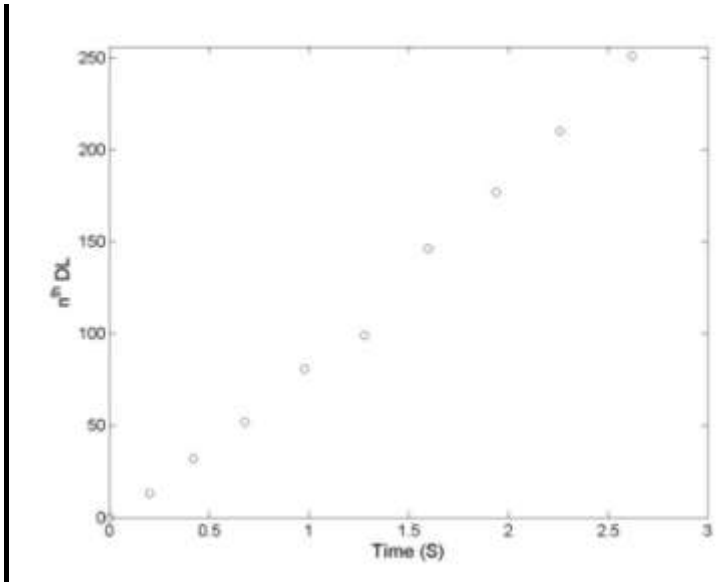


Figure 7-10 Period (1/f) versus ADC output (8bits)

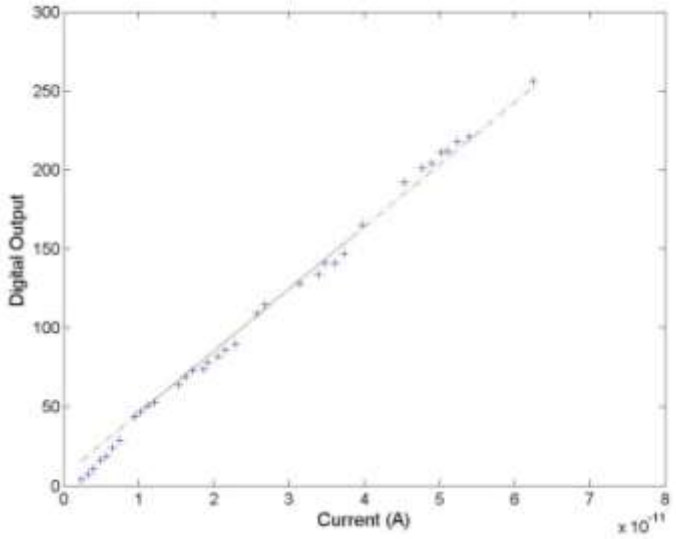


Figure 7-11 Current to digital conversion compared with absolute linear line

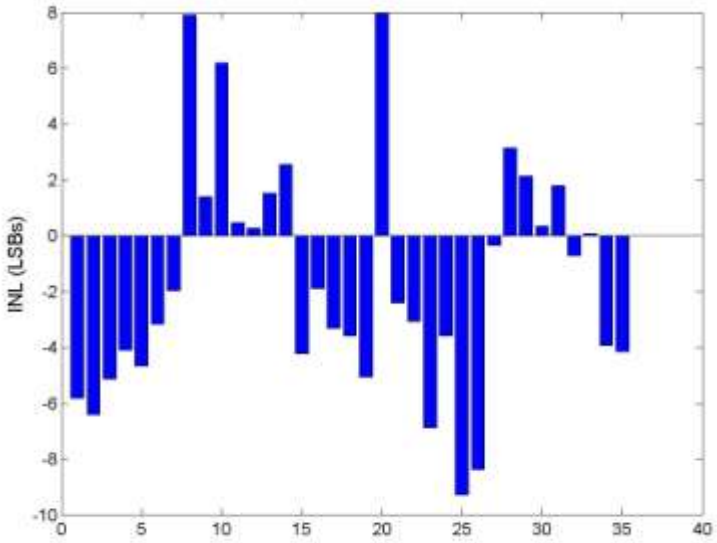


Figure 7-12 INL

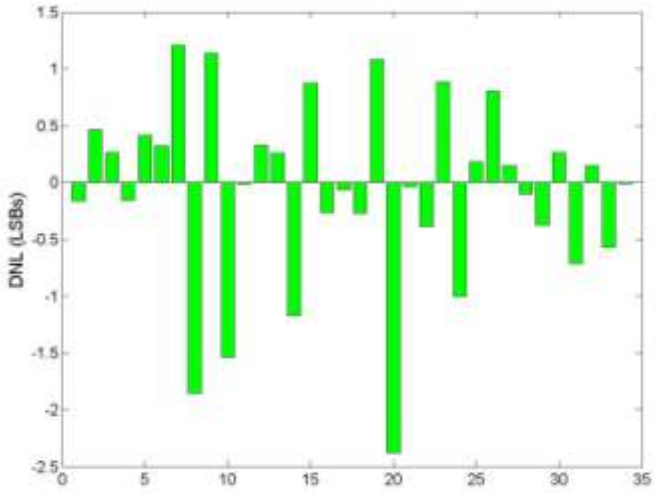


Figure 7-13 DNL

Figure 7-12 shows the INL of the I2D. The X-axis is the number of the group, and the Y-axis is the INL.

Figure 7-13 is the DNL of the I2D. The X-axis is the number of the group, and the Y-axis is the DNL.

While these results are not as good as could be hoped for, two more options could yield significantly better results. The first involves the implementation of the translinear circuit with Bipolar transistors. This necessitates a BICMOS process which was unavailable at this time. The second involves buffering any nodes of the delay line which will be brought out to pins. This was not done in this implementation which limits the uniformity of the delay line and thus limits the linearity of the system.

Figure 7-14 shows typical standby power. The X-axis is the time, and the Y-axis is the standby power. The standby power is around 5 n W, which is extremely small, and is due entirely to leakage power of the transistors.

Figure 7-15 shows the power curve when it's in conversion mode. The X-axis is the time, and the Y-axis is the supply power. When it's in conversion mode, the supply power is around 240uW. As the detecting current increases, the supply current will slightly decrease, but still in the same general magnitude.

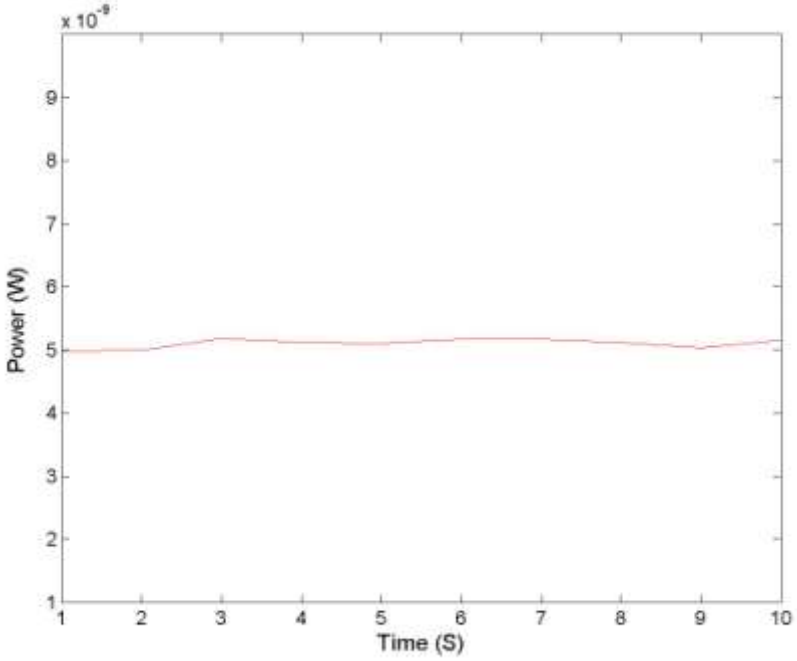


Figure 7-14 Standby power

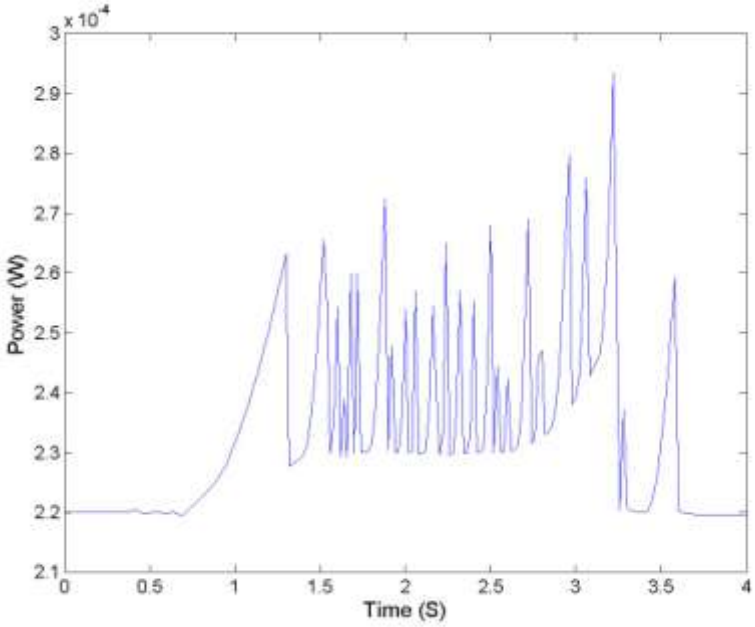


Figure 7-15 Conversion Power

Figure 7-16 is the energy cost versus input current. The X-axis is the input current, and the Y-axis is the energy cost. The smaller current it is, the less energy will cost.

7.3 Original contributions and figure of merit

7.3.1 Original contributions

This project has following original contributions:

- The whole system is a novel design, that is to say, it's a brand new way to do current to digital conversion;
- It is fully realizable in one single chip;
- Because of its unique conversion method, it uses minimal power;
- The detected current range is programmable;
- Ability to do both linear conversion and non-linear conversion
- No need to supply external current for companion purposes; while almost all the other ways are based on current comparator
- Adjustable power supply voltage VDD.

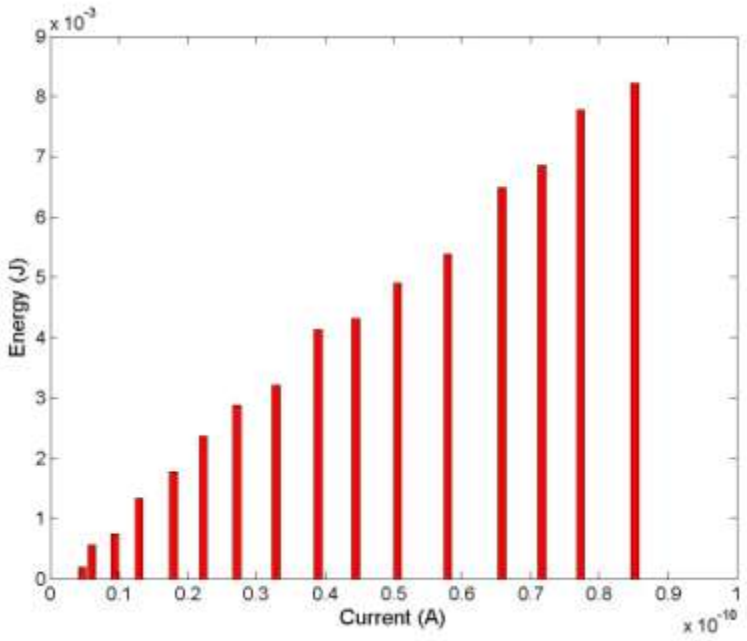


Figure 7-16 Energy cost versus input Current

7.3.2 Figure of merit

Table 2 is a comparison between this work and other work. Among them, notice that some data was not available for some of the different designs.

Figure 7-17 illustrates the area (mm^2) used per output codes comparison between this work and theirs. The X-axis complicating of the number of bits (N), and the Y-axis is the area (mm^2) per 2^N. The less area (mm^2) per 2^N, the better it is. This work is actually better than the average position of other work.

Figure 7-18 again for designs in Table 2 illustrates the power (mW) used per output code, the X-axis is the number of bits (N), and the Y-axis is the power (mW) per 2^N. The lower power (mW) per 2^N, the better it is. Note that there are two points for this work. The upper one stands for the worst condition, and the lower one stands for the best condition. Actual power used depends on usage, but would be something between them. The best case of this work is better than all the other work, and even the worst case is still better than most of the other work.

Figure 7-19 is the minimum detectable current. The X-axis is the number of bits, and the Y-axis is the minimum detectable current. From the figure, this work is much better than other work. This design can reach current in Pico range. In theory, this system should be able to detect fA or less, but instrumentation limits us to the pico range at the current time

Table 2 Comparison between this work and other work

Ref.	Bits	Process (um)	VDD (V)	Power (mW)	Area (mm^2)	Input Current Range(A)	Minimum Input Current	Type
This work	8	0.6	3.3	0.06 – 0.24	0.12	1p-	1p	Asynchronous
[6]	4	0.6	3	0.127	N/A	0-100u	6.25u	Algorithmic
[23]	4	0.6	2.5	1	N/A	0-60u	3.15u	Algorithmic (half flash)
[39]	7	0.7	5	78	N/A	32u	0.25u	Flash
[22]	10	0.8	5	110	N/A	16-528u	15.6n	Multistep flash
[10]	N/A	1	3.3	N/A	0.097	N/A	1f	Integrating
[9]	10	0.7	5	<300	N/A	512u	0.5u	Two step flash
[18]	8	1.2	5	5	0.1	0-40u	0.1563u	Successive-approximation
[40]	12	2	3.3	1.9	2.13	500u	0.1221u	Algorithmic
[41]	8	0.18	N/A	62-135u	7e-4	2.5u	9.8n	Successive-approximation
[20]	6	3	5	4.62	0.371	10-50u	0.1563u	Algorithmic
[42]	10	0.6	3.3	150	N/A	N/A	N/A	Folding and interpolating
[43]	12	0.6	3	280	N/A	N/A	N/A	Folding and interpolating
[44]	8	1.5	5	30	3.52	N/A	N/A	Folding and interpolating

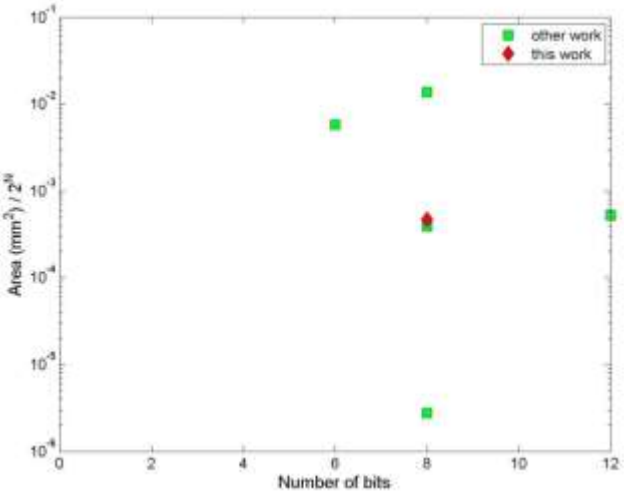


Figure 7-17 Area (mm^2) per output codes

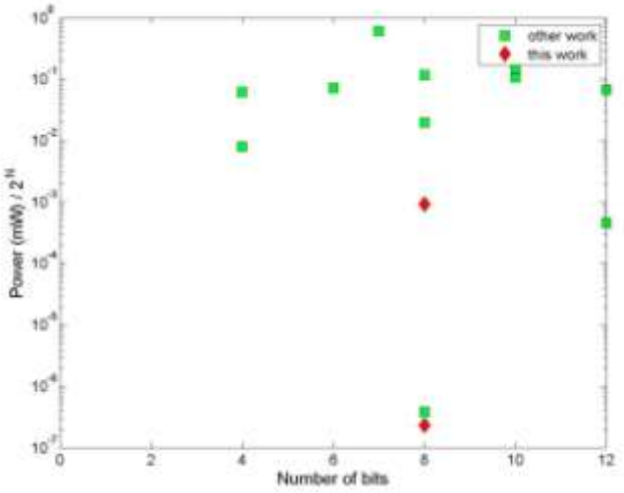


Figure 7-18 Power (mW) per output codes

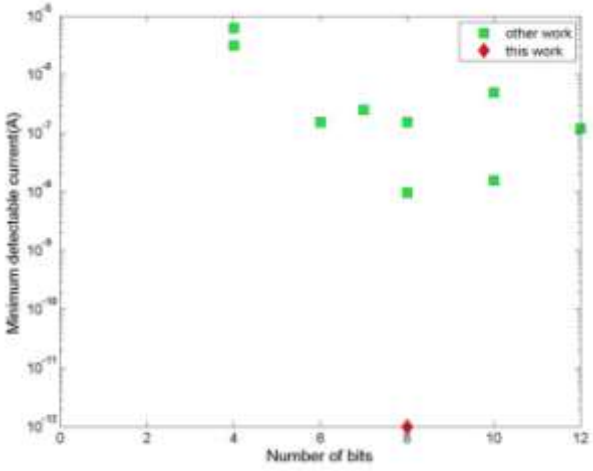


Figure 7-19 Minimum detectable current

To compare this work with other work in one figure, the following Figure of Merit could be used for evaluation:

$$FOM1 = \frac{2^{\mathrm{ENOB}} \cdot \mathrm{L}_{\mathrm{min}}{}^{2}}{\mathrm{Imin} \cdot \mathrm{A} \cdot \mathrm{P}}\ , \qquad 7\text{-}1$$

where *ENOB* stands for effective number of bits, L_{min} stands for minimum length of the process, *Imin* stands for the minimum detectable current, *A* stands for area, *P* stands for average power rating. This result is shown in Figure 7-20.

Since part of the reference work does not have information about area, so another formula excluding area information could be generated to include comparison with these parts of work:

$$FOM2 = \frac{2^{\mathrm{ENOB}} \cdot \mathrm{L}_{\mathrm{min}}{}^{2}}{\mathrm{Imin} \cdot \mathrm{P}}\ . \qquad 7\text{-}2$$

The result is shown in Figure 7-21

By both formulas, this work is far exceeds the very best among all the previous work.

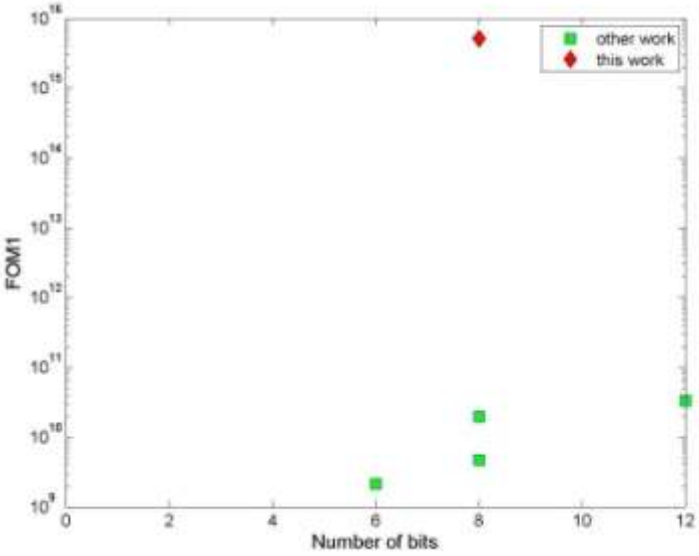


Figure 7-20 Comparison between this work and other work by FOM1

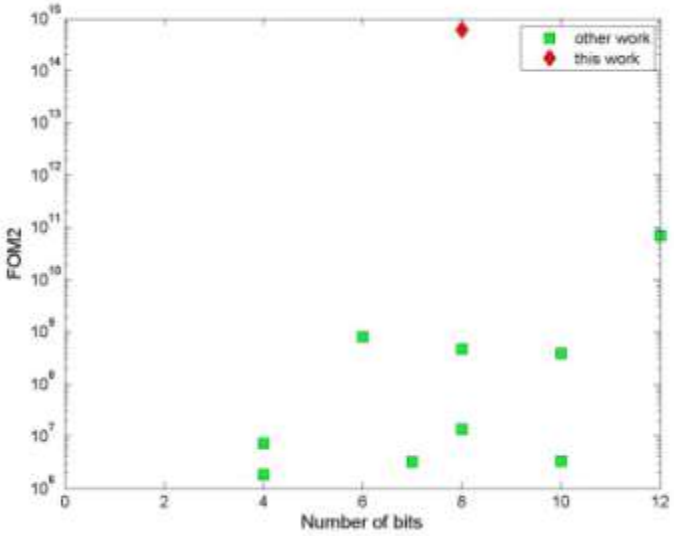


Figure 7-21 Comparison between this work and other work by FOM2

7.4 Future work

In the future, the following work should be undertaken to improve this work.

First, a sample and hold circuit needs to be developed and added to this circuit. This will allow for the circuit to continue to operate in the presence of changing input currents (AC mode). Presently, the input current must be kept at a continuous DC level until the output is available. This is not optimum for commercial designs.

Secondly, the linearity needs to be improved. As was stated before, there were two issues that significantly affected the linearity of the system. The most significant was the fact that certain delay line cells were tapped, and brought out to be viewed by external equipment. However, these taps were unfortunately un-buffered. This leads to variable delays in the delay line due to the additional capacitance of the proto-board and various pieces of test equipment that were connected to the chip. Uniformity in the delay is important to this design.

Lastly, one should seek to increase the performance over a wider current range. A major factor in this is implementing the trans-linear circuit portion of the system with bipolar transistors. Sub-threshold MOSFETs have an exponential relationship as long as they remain in their sub-threshold region. Increasing the size of the current will force the MOSFET into moderate inversion, and eventually into its quadratic region of operation. Bipolar transistors do not have this limitation. They remain in an exponential regime through all reasonable current levels. So the implementation of this design in a BiCMOS process would greatly benefit this design.

Bibliography

[1] E. Farquhar, B. Blalock, and X. Yu, “Investigation of a fully integrated frequency based current to digital converter,” CDADIC Summer Meeting, 2008.

[2] P. E. Allen and D. R. Holberg, CMOS Analog Circuit Design. Oxford University Press, 2002.

[3] http://www.beis.de/Elektronik/DeltaSigma/DeltaSigma.html.

[4] M. Bhat and H. Jamadagni, “Power optimization in current mode circuits,” 18th International Conference on VLSI Design, pp. 175–180, January 2005.

[5] H. Traff, “Novel approach to high speed CMOS current comparators,” Electronics Letters, vol. 28, no. 3, pp. 310 – 312, January 1992.

[6] V. Tipsuwanporn, A. Numsomran, Chuchotsakunleot, W., S. Chuenarom, and S. Maitreechit, “Algorithmic ADC using current mode without DAC,” Asia-Pacific Conference on Circuits and Systems, vol. 1, pp. 452 – 456, October 2002.

[7] R.-M. Weng and C.-C. Chao, “A 1.5V high folding rate current-mode folding amplifier for folding and interpolating ADC,” pp. 3942 – 3945, 2006.

[8] G. Giustolisi, G. Palumbo, and S. Pennisi, “Resolution of a current-mode algorithmic analogto- digital converter,” IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 49, no. 10, pp. 1480 – 1486, October 2002.

[9] J. Oliveira, J. Vital, and J. Franca, “A digitally calibrated current-mode two-step flash A/D converter,” IEEE International Symposium on Circuits and Systems, vol. 1, pp. 199 – 202, May 1996.

[10] M. Uster, T. Loeliger,W. Guggenbuhl, and H. Jackel, “Integrating ADC using a single transistor as integrator and amplifier for very low (1 fA minimum) input currents,” Third International Conference on Advanced A/D and D/A Conversion Techniques and Their Applications, pp. 86 – 89, July 1999.

[11] J. Nakamura, B. Pain, T. Nomoto, T. Nakamura, and E. Fossum, “On-focal-plane signal processing for current-mode active pixel sensors,” IEEE Transactions on Electron Devices, vol. 44, pp. 1747 – 1758, October 1997.

[12] E. Farquhar, X. Yu, and B. Blalock, “A frequency based current-to-digital converter with programmable dynamic range,” Poster for CDADIC Summer Meeting, 2007.

[13] C. E. Carr, “Delay line models of sound localization in the barn owl,” American Zoologist, vol. 33 (1), pp. 79 – 85, 1993.

[14] http://www.tbi.univie.ac.at/ pks/Presentation/leipzig-04.pdf.

[15] E. Farquhar and P. Hasler, “a bio-physically inspired silicon neuron,” IEEE Transactions on Circuits and Systems, vol. 52, no. 3, pp. 477 – 488, March 2005.

[16] C. Mead, Analog VLSI and Neural Systems. Reading, MA: Addison-Wesley, 1989.

[17] M. Keramat, “Design & test of low-power/high-speed data converters,” The Midwest Symposium on Circuits and Systems (MWSCAS), 2008.

[18] L. Ravezzi, D. Stoppa, and G.-F. D. Betta, “Current-mode A/D converter,” Electronics Letters, vol. 34, no. 7, pp. 615 – 616, April 1998.

[19] T. Kaya and A. Zeki, “A modified active current mirror suitable for current-mode algorithmic analog-to-digital converters,” 9th International Conference on Electronics, Circuits and Systems, vol. 1, pp. 15 – 18, September 2002.

[20] D. Nairn and C. Salama, “A current mode algorithmic analog-to-digital converter,” IEEE International Symposium on Circuits and Systems, vol. 3, pp. 2573 – 2576, June 1988.

[21] http://en.wikipedia.org/wiki/Analog_ to_ digital_ converter.

[22] S. H. Shim and K. S. Yoon, “A 10-bit current-mode low-power CMOS A/D converter with a current predictor and a modular current reference,” the 40th Midwest Symposium on Circuits and Systems, vol. 1, pp. 342 – 345, August 1997.

[23] S. Chuenarom, S. Maitreechit, P. Roengruen, and V. Tipsuwarnpron, “Low power currentmode algorithmic ADC in half flash (BCD),” IEEE Asia Pacific Conference on Circuits and Systems, pp. 175 – 178, December 2006.

[24] M. Konishi, “Study of sound localization by owls and its relevance to humans,” Comparative Biochemistry and Physiology, vol. 126 (4), pp. 459 – 469, 2000.

[25] E. I. Knudsen, "Neural derivation of sound source location in the barn owl. an example of a computational map," Annals of the NY Academy of Science, vol. 510 (11), pp. 33 – 38, 1987.

[26] M. Beigel and J. McGary, "Accurate, real-time localization of subminiature inductive transponders: Tumor localization as an example," Proceedings of the 2005 European Conference on Circuit Theory and Design, vol. 2, pp. II/173 – II/176, September 2005.

[27] J. Wolf, W. Burgard, and H. Burkhardt, "Robust vision-based localization by combining an image-retrieval system with monte carlo localization," IEEE Transactions on Robotics, vol. 21, no. 2, pp. 208 – 216, April 2005.

[28] C. Pradalier and S. Sekhavat, " "localization space": a framework for localization and planning, for systems using a sensor /landmarks module," IEEE International Conference on Robotics and Automation, vol. 1, pp. 708–713, May 2002.

[29] C. Deyres-Pescay, J. L. Boizard, P. Leman, and B. Aragon, "A precise localization system dedicated to hydrobiological studies of lakes. fish localization, topography, tomography and lakes bed types mapping," Conference Proceedings, Prospects for the 21st Century, vol. 3, pp. 1551 – 1556, 1996.

[30] P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," IEEE Transactions on Automation Science and Engineering, vol. 3, no. 4, pp. 360 – 371, October 2006.

[31] T. Nara, S. Suzuki, and S. Ando, "A closed-form formula for magnetic dipole localization by measurement of its magnetic field and spatial gradients," IEEE Transactions on Magnetics, vol. 42, no. 10, pp. 3291 – 3293, October 2006.

[32] P. Aarabi, "The application of spatial likelihood functions to multi-camera object localization," in Proc. 5th SPIE Conf. Sensor Fusion, April 2001.

[33] M. S. Brandstein, "A framework for speech source localization using sensor arrays," Ph.D. thesis, Dept. Elec., Eng., Brown University, Providence, RI, May 1995.

[34] M. S. Brandstein and H. Silverman, "A robust method for speech signal time-delay estimation in reverberant rooms," in Proc. ICASSP, May 1997.

[35] J. D. H. Silverman and M. Brandstein, "Robust localization in reverberant rooms," in Microphone Arrays: Signal Processing Techniques and Applications, M. S. Brandstein and D. B. Ward, Eds. New York: Springer-Verlag, 2001.

[36] R. Z. J. Flanagan, J. Johnston and G. Elko, "Computer-steered microphone arrays for sound transduction in large rooms," J. Acoust. Soc. Amer., pp. 1508 – 1518, November 1985

[37] B. Mungamuru and P. Aarabi, "Enhanced sound localization," IEEE Transactions on Systems, Man and Cybernetics, Part B, vol. 34, no. 3, pp. 1526 – 1540, June 2004.

[38] B. A. Minch, "Analysis and synthesis of static translinear circuits," Technical Report No. CSL-TR-2000-1002, March 2000.

[39] M. S. Bhat, Rekha S, and H. S. Jamadagni, "Design of a lower current-mode flash ADC", IEEE Region 10 Conference TENCON, vol. 4, no. 21-24, pp. 241 – 244, November 2004.

[40] Jin-Sheng Wang, Chin-Long Wey, "A 12-bit 100-ns/bit 1.9-mW CMOS switched-current cyclic A/D converter", IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 46, no. 5, pp.507 – 516, May 1999.

[41] L. Vesalainen, J. Poikonen, A. Paasio, "A gray-code current-mode ADC for mixed-mode cellular computer", Proceedings of the 2004 International Symposium on Circuits and Systems, vol. 3, pp. 81 – 84, May 2004.

[42] Jin Won Chung; Kwang Sub Yoon, "Design of 3.3 V 10 bit current-mode folding/interpolating CMOS A/D converter with an arithmetic functionality", Proceedings of the Second IEEE Asia Pacific Conference on ASICs, pp.45 – 48, August 2000.

[43] Hyung Hoon Kim, Kwang Sub Yoon, "A 12 bit current-mode folding/interpolation CMOS A/D converter with 2 step architecture", The First IEEE Asia Pacific Conference on ASICs, pp. 174 – 177, August 1999.

[44] Kyung Myun Kim, Kwang Sub Yoon, “An 8-bit 42 Msamples/s current-mode folding and interpolation CMOS analog-to-digital converter with three-level folding amplifiers**”,** IEEE 39th Midwest symposium on Circuits and Systems, vol. 1, pp. 201 – 204, August 1996

[45] C. Diorio, P. Hasler, A. Minch, C. A. Mead, “A single-transistor silicon synapse”, IEEE Transactions on Electron Devices, vol. 43, no. 11, pp. 1972 – 1980, November 1996

Vita

Xiaoyan (Lucy) Yu was born in Hubei, China on October 26, 1978. She entered Huazhong University of Science and Technology in 1996. She majored in Electrical Engineering, and obtained a bachelor degree of Electrical Engineering in 2000; at the same time, she minored in Financial Management, and also obtained a bachelor degree of Management. As a No. 1 student in the class, she was directly admitted into graduate program and obtained a master degree of Engineering in 2003.

In 2004, Xiaoyan started her graduate study in Department of Electrical and Computer Engineering (Now is Min H. Kao Department of Electrical Engineering and Computer Science), University of Tennessee, Knoxville. She obtained a Master degree of Electrical Engineering in December 2006. Her master project title is Object Localization Using Arrayed Integrated and Fire Neuron Circuits.

Xiaoyan's Ph. D study has been in the NEUT (Neuro-engineering at UT) lab under the direction of Dr. Ethan Farquhar. Her research area is focused on Analog & Mixed Signal Circuit design.