An Efficient A/D Converter using Electronic Neurons

Ke Wei Zheng

A Thesis

In

The Department

Of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

January 2006

Your file Votre référence
ISBN: 0-494-14292-8
Our file Notre référence
ISBN: 0-494-14292-8

Abstract

An Efficient A/D Converter using Electronic Neurons

Ke Wei Zheng

Analog to digital converter (ADC) is an important building block for modern electronic design. There exist different type of ADC, such as Integrating ADC; Successive approximation ADC; Flash ADC and so on. Each of them usually focuses on one or more design consideration. For example Flash ADC has high working frequency but it consume more power than other type of ADCs.

We are also aware of that human brain works by receiving minute electrical information signals produced by nerve cells known as neurons. A neuron 'fires' (i.e., turns on) when it receives a stimulus of sufficient strength, and emits a number of narrow pulses known as 'action potentials'. The number of such pulses is proportional to the strength of the stimulus. Inspired by this physiological nature of functioning of a neuron, i.e., production of a bundle of actions potentials proportional to the level of a stimulus, an ADC with the architecture of 'electronic neuron cell plus counter' is implemented using transistors in a standard CMOS integrated circuit technology. The approach is simple and the result is a low voltage low power unipolar A/D converter working in middle frequency range. The converter is very competitive with other known converters in terms of 'energy per sample'. Furthermore, by using a new calibration method, our ADC affords to very good linearity performance.

Acknowledgements

The research was supported by a grant awarded to Dr. Raut by the Natural Science and Engineering Research Council (NSERC), Canada.

I would like to express my sincerest gratitude to my supervisor Dr. R. Raut, who originally suggested the idea for this thesis, for his never failing support and encouragement, for all the insightful comments he gave me at different stages of my research, as well as for challenging and criticizing my ideas and for making me think harder than I thought I could.

I am also thankful to Sourendra Raut, MDCM, for providing very helpful basic information about neurons.

Finally, I would like to express my gratitude to Dr. J. X. Zhang, Dr. V. Ramachandran and Dr. M. Packirisamy, for volunteering their valuable times to review this thesis and contributing several constructive comments.

List of Contents

List of Contents v

List of Figures vii

List of Symbols and Abbreviations xiii

Chapter 1 Introduction and Motivation 1

Chapter 2 Brief Background of Natural Neural System 6

2.1 Physiological structure of neuron cells 6
2.2 Electrical Property of the Neurons 7
2.3 Propagation of Signal in a Neuron Cell 8
2.4 Mechanism of the generation of AP pulse 11
2.5 Hodgkin-Huxley Model 12
2.6 Network behavior of neural system 18
2.7 System diagram of our ADC system 19
2.8 Summary 22

Chapter 3 Neuron cell implementation 23

3.1 Mechanism of operation 23
3.2 Transfer Characteristic 26
3.3 Calculation of the components of the Neuron Cell Part 31
3. 4 Improving the shape of the AP pulse 33
3.5 Performance of the Neuron Cell part 37
3.6 Summary 40

Chapter 4 The Counter Implementation 41

4.1 Typical Asynchronous Counter 41
4.2 New Counter structure 45
4.3 Simulation results 46
4.4 Summary 47

Chapter 5 Performance and Improvements 49

5.1 The Performance of the ADC 49
5.1.1 Simulation results 49
5.1.2 Post analysis by Matlab 52
5.2 Error analysis 55
5.3 Calibration method 56
5.4 Design of the linear calibration circuit 58
5.6 Simulation result 64
5.7 Summary 67
Chapter 6 Layout implementation and results 68
6.1 Layout Design 68
6.2 Simulation Results with a Sinusoid Input Signal 69
6.3 Power consumption 70
6.4 Robustness of our system 71
6.5 Summary 75
Chapter 7 Conclusion and Future Work 76
7.1 Conclusion 76
7.2 Future work 77
7.3 Contributions from this thesis 78
Appendix 79
A-1: Matlab Code for Hodgkin-Huxley model 79
A-2: Verilog_A code of the IDEAL 6 BITS DAC 81
A-3: Matlab code for the post analysis of ADC 82
A-4: Definition of the metrics for the performance of ADC 87

List of Figures

Figure 1-1 Integrating (dual slop) ADC 2

Figure 1-2 A 3-bit Flash ADC 3

Figure 2-1 The structure of a neuron cell and the connection with other cells 7

Figure 2-2 Microcosmic environment of a living neuron cell to show the electrical property. The ions may flow in or out through biological 'tunnel' [7] 8

Figure 2-3 Summation of two signals in trigger zone. (a) Two signals from different dendrites reaching the trigger zone at different time will not be summed up; (b) simultaneous signals are summed up by magnitude in the trigger zone [7]. 9

Figure 2-4 Signal propagates in a neuron cell [7] 11

Figure 2-5 The process of an action potential generation [7] 12

Figure 2-6 Hodgkin-Huxley Model 13

Figure 2-7 Membrane potential vs. time for Hodgkin-Huxley Model 16

Figure 2-8 Channel conductance parameter vs. time for Hodgkin-Huxley Model 16

Figure 2-9 Potassium and sodium conductance of Hodgkin-Huxley Model 17

Figure 2-10 A three layer network coupled by weighted connections 18

Figure 2-11 Basic structure of our ADC 20

Figure 2-12 Controlling clock CLK1 (top) and CLK2 (bottom). The period of these two clock are same and it is the sampling period for our ADC 21

Figure 3-1 Neuron Cell Part 24

Figure 3-2 (a) The output of Neuron Cell Part; (b) voltage curve of Membrane 25

Figure 3-3 Comparison of two sampling method. (a) Point-sampling method (b) Area-sampling method 28

Figure 3-4 (a) An ideal AP pulse, (b) A distorted AP pulse due to practical performance of the non-ideal inverters 34

Figure 3-5 The inverter and its DC sweep result 34

Figure 3-6 complete schematic of Neuron Cell Part 36

Figure 3-7 Transient simulation results to show the enhancement of 4 pairs of inverter. (a) Schematic of the testing circuit in which V1 is a linearly increasing voltage source with increasing rate of 1V/us. (b) Transient simulation result of the output of 1st pair inverter whose slewing rate is 400V/ μs, and the output of 4th pair inverter whose slewing rate is 18KV/ μs. 36

Figure 3-8 Simulation result of Neuron Cell Part, when the input current is 0.5 μA. (a) Waveform of AP pulses. There are three pulses in each Sampling period (b) Waveform of CLK2. The time interval between two pulses constructs a sampling period. 38

Figure 3-9 The waveform of one AP pulse when the input current is 0.5 μA. 38

Figure 3-10 Simulation result of Neuron Cell Part, when the input current is 5 μA. (a) Waveform of AP pulses. There are 36 pulses in each Sampling period (b) Waveform of CLK2. The time interval between two pulses constructs a sampling period. 39

Figure 3-11 The waveform of one AP pulse when the input current is 5 μA. 39

Figure 4-1 A typical 6-bits asynchronous counter composed by T flip-flops...................... 41

Figure 4-2 A one-bit counter composed only by a T-latch.. 42

Figure 4-3 (a) The counter without the pulse detector mal-functions when the input pulse width is over the triggering limit time; (b) the counter without the pulse detector works well when the input pulse width is shorter the triggering limit time.. 42

Figure 4-4 A positive edge pulse detector.. 43

Figure 4-5 Simulation result for the Pulse Detector. (a) Output of the Pulse Detector with one inverter; (b) Output of the Pulse Detector with three inverters.. 44

Figure 4-6 One bit structure of the new counter.. 46

Figure 4-7 Complete structure of the new 6-bits counter 46

Figure 4-8 Simulation result of the Non-Pulse-Detector Counter. The input pulse is AP pulses from Neuron Cell Part 47

Figure 5-1 Test bench for the ADC.. 50

Figure 5-2 Simulation result of the non-calibration ADC, where the input is a linearly increasing current in the range from 0 to 9.1 μA 52

Figure 5-3 Result of post analysis by Matlab. Upper left: transfer response of our non-calibration ADC Vs. ideal 6bits ADC. Upper right: transfer response of our non-calibration ADC after Offset error and Gain error are removed Vs. ideal 6bits ADC. Lower left: INL of our our non-calibration ADC. Lower right: DNL of our non-calibration ADC. 54

Figure 5-4 Non-ideal AP pulses, where the pulse width is labeled as t_{pulse}, and the time interval between two pulses is $t_{\text{int}\,egral}$ 55

Figure 5-5 The best compensation charge and its first order fitting line. 58

Figure 5-6 Schematic of our ADC with calibration 59

Figure 5-7 (a) the output of the charge pump, which is used to control M22; (b) one AP pulse, which is fed back to control M1. 60

Figure 5-8 The comparison of the best compensation and the actual compensation achieved 62

Figure 5-9 Simulation result of the ADC with calibration, where the input is a linear increasing current in the range from 0 to 8 μA 65

Figure 5-10 Result of post analysis by Matlab. Upper left: transfer response of our ADC with calibration Vs. ideal 6bits ADC. Upper right: transfer response of our ADC with calibration after Offset error and Gain error are removed Vs. ideal 6bits ADC. Lower left: INL of our ADC with calibration. Lower right: DNL of our ADC with calibration. 66

Figure 6-1 Layout of the core circuit of our ADC. 69

Figure 6-2 The simulation result with a sinusoid input signal 69

Figure 6-3 Matlab post analysis result of our ADC with Calibration under 80°C. Upper left: transfer response of our ADC Vs. ideal 6bits ADC. Upper right: transfer response of our ADC after Offset error and Gain error are removed Vs. ideal 6bits ADC. Lower left: INL of our ADC. Lower right: DNL of our ADC. 73

Figure 6-4 Matlab post analysis result of our ADC with Calibration under -20°C. Upper left: transfer response of our ADC Vs. ideal 6bits ADC. Upper right: transfer response of our ADC after Offset error and Gain error are removed Vs. ideal 6bits ADC. Lower left: INL of our ADC. Lower right: DNL of our ADC. 74

List of Tables

Table 1-1 Comparison between Integrating ADC and Flash ADC 3

Table 3-1 Design parameters of our ADC 33

Table 5-1 Comparison of the performance of ADC with and without Calibration 67

Table 6-1 Comparison between our ADC and other low power ADC 71

Table 6-2 Comparison of our ADC under different test condition. 72

List of Symbols and Abbreviations

ADC	Analog to Digital Conversion
ANN	Artificial Neural Network
AP	Action Potential
CLK1	Counter reset clock
CLK2	Neuron reset clock
C_1	The value of the membrane capacitor
$C_{1\_\max}$	The maximum possible value of the membrane capacitor C_1
DNL	Differential Nonlinearity error
E_{off}	Offset error
FSR	Full Scale Range of the input signal
i_{input}	The input current of ADC
i_x	The compensation current
i_{input_FSR}	The FSR value of the input current
I_B	The input analog current at which the digital output transition from B-1 to B occurs.
I_{LSB}	The input current corresponding to the Least Significant Bit (LSB)
INL	Integral Nonlinearity error
$Q_{injected}$	The charge injected into the Neuron Cell Part during one sampling period
Q_x	The compensation charge needed in each sampling period

Q_{input}	The charge injected into the Neuron Cell Part during each sampling period
SOC	System on a chip
$t_{\mathrm{int}egral}$	Time interval for the integral step
t_{pulse}	Time interval for the pulse step
$t_{\mathrm{integral\_min}}$	The minimum value of the time interval for the integral step. It happens when the amplitude of the input current is equal to FSR.
$T_{sampling}$	Sampling period
$v_{Membrane}$	The voltage of the membrane in the Neuron Cell Part
$V_{threshold\_of\_U1}$	The threshold voltage of the first inverter in the Neuron Cell Part

Chapter 1

Introduction and Motivation

Since last century, extensive studies show that the brain of human being possesses extraordinary power in processing information and learning knowledge with great fault-tolerance and tremendously low power consumption, compared with the most advanced processors and equipments, such as computers and artificial intelligent machines of the modern time. These appealing features have drawn attention of many scientists and engineers since the middle of last century [1] - [3] towards understanding and implementing the functionalities of human brain using electronic devices.

In reality human brain is a very complex machine. During the last seventy years the intricacies behind some of these complexities have been resolved, but many of them are yet to be known. Whatever has become known so far has brought in great developments in both biological and engineering domain. Thus scientists have introduced the concept of artificial neurons and the theory of artificial neuron network (ANN) [1], which has led to many valuable applications realized either in software packages or in hardware systems. For example, based on the study of bat echolocation system, Cheely and Horiuchi proposed a 'VLSI Range-Tuned Neuron model' that is used in real time sonar system [4]. Another example is Nielsen and Lund's 'Spiking Neural Building Block', which is used for the application of robot [5]. It allows the mobile robot to adapt to changing environmental conditions.

In understanding the functionalities of human brain, we need to dig into the fundamental elements that facilitate the various functions of brain. One class of elements that have interesting electrical properties is nerve cells known as neurons. By studying the behavior of the natural neuron cell, we learn that an Analog to Digital Conversion (ADC) mechanism naturally exists [1], [6] in the functioning of neurons. This gives us a clue that using electronic neuron might be a possible approach to build an electronic ADC.

In modern electronic engineering domain, ADC is an important building block. Many type of ADCs have been developed and widely used, such as Integrating ADC (Figure 1-1); Successive approximation ADC; Flash ADC (Figure 1-2) and so on. Each of them has its strength and weakness in different aspects, such as circuit complexity; power consumption; linearity performance; resolution and so on. The choice of what type of ADC to be used is usually based on specific design consideration. For example, if one uses an ADC in a high frequency design, and the power consumption and high linearity are not a major consideration, Flash ADC would be a preferred choice.

To see the difference of different type of ADC, a brief comparison between typical Integrating ADC and Flash ADC is shown in table 1-1

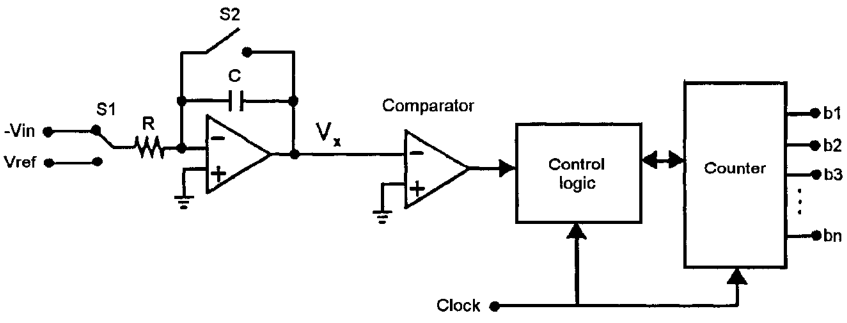


Figure 1-1 Integrating (dual slope) ADC

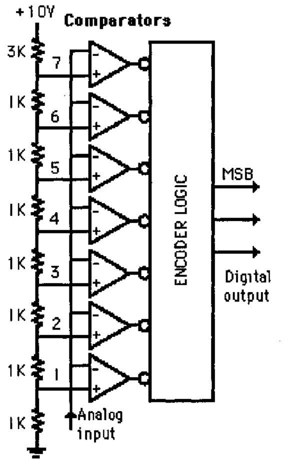


Figure 1-2 A 3-bit Flash ADC

Table 1-1 Comparison between Integrating ADC and Flash ADC

	INTEGRATING	FLASH
Working Frequency	Lowest in all type of ADCs	Highest in all type of ADCs
Power Consumption	Moderate	High. Power consumption increases exponentially with resolution.
Resolution	High	Typically within 8 bits.
Circuit Complexity	Low. Only one OPAM, one comparator and one counter needed. Typically just about one hundred transistors needed	High. For N bits resolution, 2^{N-1} comparators needed. Typically, hundreds or more than thousand transistors needed.
Die Size	Small. Core die size usually does not change with increase in resolution.	Large. Die size increases exponentially with resolution.
Cost	Cheap	Expensive

We embarked on building an ADC using electronic neurons because we wanted to exploit the natural A/D conversion behavior inherent in a neuron cell. Further, to the best of our knowledge, since nobody has yet produced such work, we felt encouraged to explore the potentials of a neuron cell as an ADC system. The knowledge that biological neural system is extremely power efficient and highly reliable, we expected that our research may lead to an ADC that offers good performance in power consumption and robustness.

Hodgkin-Huxley Model [8] first mathematically described the behavior how a neuron generates pulses (namely action potential pulses or simply AP pulses) corresponding to an input stimulus. Their work embodies the fundamental theory about implementing an electronic neuron. Based on this model and Elisabetta Chicca et al's earlier work [20], we proposed our electronic neuron model. It takes current as input and generates a bunch of voltage pulses as output. The density of these pulses is proportional to the amplitude of input current. This electronic neuron is one of the fundamental building blocks of our ADC.

A follow-up counter counts the number of AP pulses from electronic neuron in a constant time interval. This transfers the density information carried the pulses to binary voltage streams.

Finally, by employing a new calibration method, the simulation results show that our ADC achieves satisfactory performance in power consumption, linearity, simplicity of circuit structure and robustness against temperature variation.

The thesis is organized as follows. In chapter 2 we present a brief background of some fundamental knowledge about the physiological neurons and Hodgkin-Huxley Model. Based on the knowledge introduced in chapter 2, we present the design work of a major part of the electronic ADC, namely Neuron cell part in chapter 3. In this chapter, the complete schematic of electronic neuron cell and its transfer function are introduced. Also, simulation results of the electronic neuron are shown at the end of this chapter. Another major part, namely counter is presented in chapter 4. In this chapter, we first introduce the typical asynchronous counter, and then we proposed a compact counter structure based on our specific design condition. Simulation results of the new counter are also included in the end of this chapter. In chapter 5, we discuss the performance of the ADC and some deficiencies are analyzed. Based on this analysis, a linear calibration circuit is constructed and used. The final simulation result shows substantial improvement our ADC's linearity performance. The layout design, post layout analysis about the performance of power consumption and robustness against variation of temperature and power supply are introduced in chapter 6. This is followed by conclusion and suggestions about future work.

Chapter 2

Brief Background of Natural Neural System

In this chapter, we first introduce some knowledge pertaining to neurons in sections 2.1, 2.2 and 2.3; next, the mechanism of generating an action potential (AP) pulse is discussed in section 2.4. In section 2.5, we introduce the Hodgkin-Huxley model. This is followed by a brief discussion of the electrical behavior of the neurons and some important characteristics of neural network system in section 2.6.

2.1 Physiological structure of neuron cells

The structure of a typical neuron cell and three neurons to which it connects is shown in Figure 2-1. From this figure, we see that a neuron cell mainly consists of a cell body, a bunch of dendrites, and one or more axons. The cleft between two neuron cells is called synapse. It couples the signal from the axons of the previous cell to the dendrites of the next cell, and the coupling efficacy is often called synapse weight. The base of the axon is called hillock or trigger zone where the natural ADC process occurs. More details about this process will be discussed in the following section.

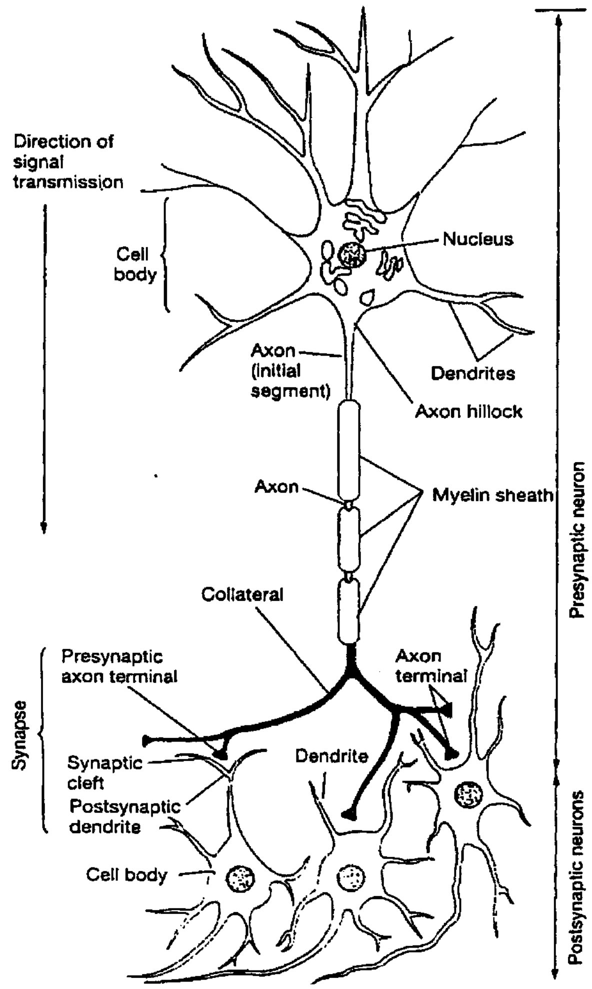


Figure 2-1 The structure of a neuron cell and the connection with other cells

2.2 Electrical Property of the Neurons

Research in the past revealed that the human brain consists of about 10 billion of neurons, massively interconnected, and the signal propagated in a natural neuron network is mainly electrical [1], [7]. The electrical property is originated from the electric ions inside and outside a neuron cell, namely positive Sodium ion (Na^+) and positive Potassium ion (K^+). Figure 2-2 shows this point, where the bold circle refers to the neuron cell membrane and the word 'ATP' is the abbreviation of 'Adenosine Triphosphate'. At the rest state the concentration of the ions inside and outside of the membrane are not equal; this generates a so-called rest potential difference around –70mV between the two sides of a living neuron cell's membrane. The minus sign means that the potential inside the cell is lower. The ions may flow in or out through some biological 'tunnel', which means that the potential between the two sides of the membrane (it is referred as membrane potential thereafter) may change.

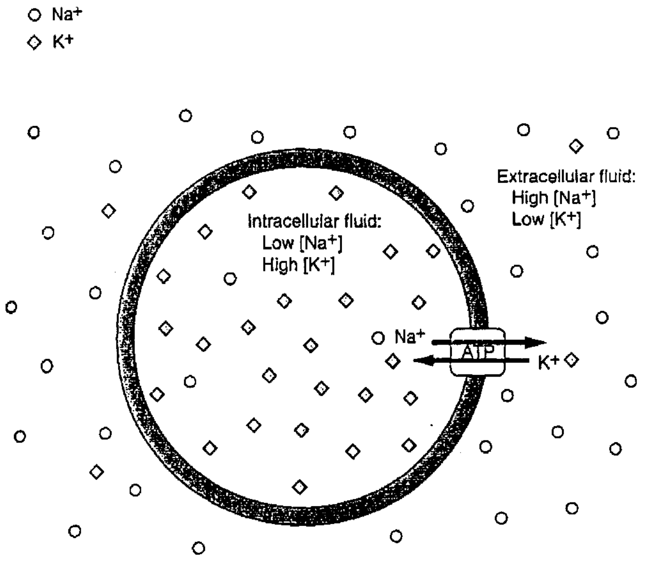


Figure 2-2 Microcosmic environment of a living neuron cell to show the electrical property. The ions may flow in or out through biological 'tunnel' [7]

2.3 Propagation of Signal in a Neuron Cell

According to the direction of a signal transmission, we first define the previous neuron cell and the rear cell as pre-synaptic neuron and post-synaptic neuron respectively. The process of signal propagation in a particular neuron cell can be divided into three steps to clearly understand the operation.

First, in the case that an outside stimulus or the signal from a pre-synaptic neuron appears, the tips of dendrites close to this pre-synaptic neuron generate an electrical signal. It is called graded potential because its magnitude depends on the strength of the stimulus. This signal then propagates towards the center of the cell of the neuron we are considering, and its amplitude gradually decreases in this propagation process. Similar electrical signals picked up by other dendrites close to other neighboring pre-synaptic cells, also propagate towards the same zone of the neuron in question. In the trigger zone near the center of cell body, all these signals are summed up. Figure 2-3 shows the summation process. So far, the process is completely analog.

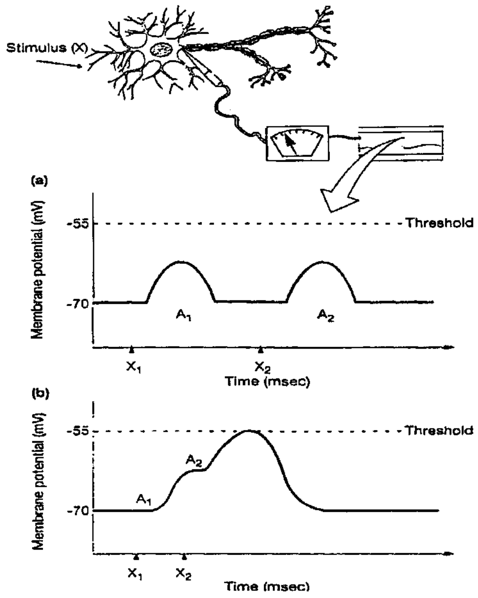


Figure 2-3 Summation of two signals in trigger zone. (a) Two signals from different dendrites reaching the trigger zone at different time will not be summed up; (b) simultaneous signals are summed up by magnitude in the trigger zone [7].

In the trigger zone, the summed signal is compared with a threshold. If it is lower than the threshold, the input signal is simply ignored, otherwise a bunch of pulses are generated. When the stimulus is stonge, it is very likely that the summed signal will exceed the threshold. The density (i.e., the number in some specific time) of the pulses mentioned above is proportional to the magnitude of the summed signals. Conventionally, this pulse is called as action potential or simply AP. All these pulses have the same amplitude and pulse width; furthermore, in the following propagation process their shape will not be modified. So, the AP pulse is much similar to a digital pulse (even though it is not a binary word). This step is similar to an ADC process.

The last step is the coupling step. The AP pulses generated at the trigger zone propagate along the axon and finally reach the axon terminal. There, it is coupled to the post-synaptic neuron through the synapse. In most cases, the coupling process is chemical, that is, the axon terminal of pre-synaptic neuron releases chemical transmitters into the synaptic cleft. The number of the transmitters depends on the density of AP pulses. The transmitters then stimulate the dendrites of the post-synaptic cell and an electrical signal is regenerated in the post-synaptic neuron cell. The magnitude of this electrical signal is determined by the concentration of the chemical transmitters in the cleft. By repeating above steps, the signal propagates forward from one cell to the next until finally it reaches the destination (i.e., the control center of the brain).

Figure 2-4 illustrates, in a signal transmission process, the various forms of the signal in different portions of a neuron cell. It clearly shows that

i) The graded potential decrease in magnitude along the propagation;

ii) If the amplitude of signal is beyond a certain threshold, action potential pulses are fired;

iii) The density of the action potential is dependant on the amplitude of the summed graded potential;

iv) The number of the transmitters released by pre-synaptic neuron in the synapse depends on the density of the action potential that reaches the axon terminal.

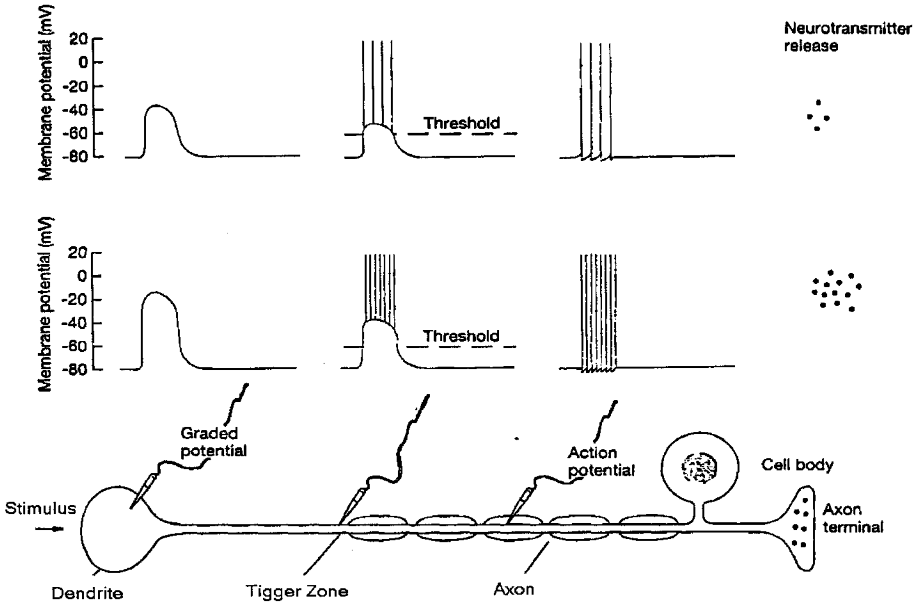
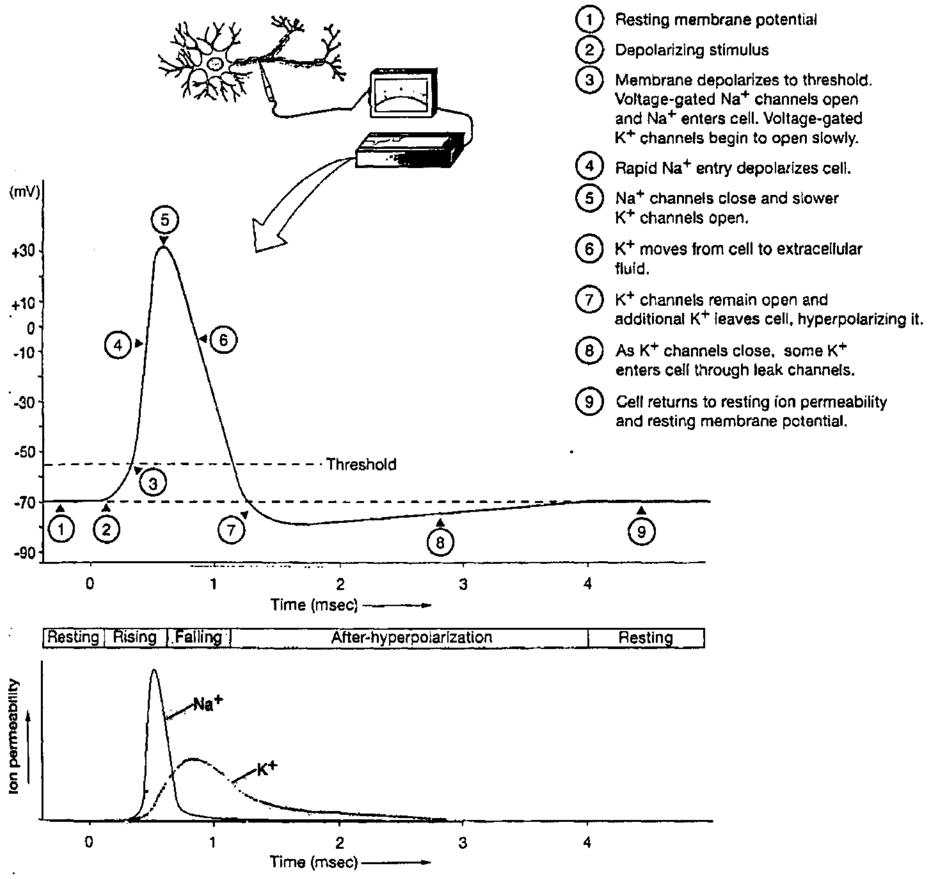


Figure 2-4 Signal propagates in a neuron cell [7]

2.4 Mechanism of the generation of AP pulse

Figure 2-5 illustrates the mechanism how an action potential pulse is generated in a natural neuron cell. As mentioned above, the uneven ionic concentration causes the potential difference cross the neuron membrane. To generate an action potential pulse, the Na^+ ions flow into the cell at a fast rate, which depolarize the membrane to around +30mV, almost simultaneously, the K^+ ions flow out the cell but at a slower rate, which repolarize the membrane to its resting membrane potential. The combination of ions flowing in and flowing out the neuron cell generates an electrical pulse (i.e. the action potential pulse). The flowing in and flowing out behavior is controlled by some biological channel, which is composed by some specific type of protein. One should note that in Figure 2-5, the potential curve is about the inside layer of the membrane while the outside layer is thought as grounded.



(1) Resting membrane potential

(2) Depolarizing stimulus

(3) Membrane depolarizes to threshold. Voltage-gated Na^+ channels open and Na^+ enters cell. Voltage-gated K^+ channels begin to open slowly.

(4) Rapid Na^+ entry depolarizes cell.

(5) Na^+ channels close and slower K^+ channels open.

(6) K^+ moves from cell to extracellular fluid.

(7) K^+ channels remain open and additional K^+ leaves cell, hyperpolarizing it.

(8) As K^+ channels close, some K^+ enters cell through leak channels.

(9) Cell returns to resting ion permeability and resting membrane potential.

Figure 2-5 The process of an action potential generation [7]

2.5 Hodgkin-Huxley Model

Based on vast observations, in 1949, Hodgkin and Huxley, who are recognized for their Nobel Prize winning work on the ionic theory of nervous conduction, proposed a mathematical model to describe the behavior how an AP pulse is generated. This model name is known as Hodgkin-Huxley Model [8]

In this model, a patch of neural membrane is considered as a novel RC circuit (see

Figure 2-6) [8]. A capacitor C_M is used to duplicate the neural membrane; the resistors R_K, R_{Na} and R_L represent the permeability of potassium, sodium, and leakage channel respectively; the parameters V_K, V_{Na} and V_L are the equilibrium potentials of potassium, sodium, and leakage channel respectively, which is artificially adopted to make the model precisely imitate the behavior of biological neuron cell.

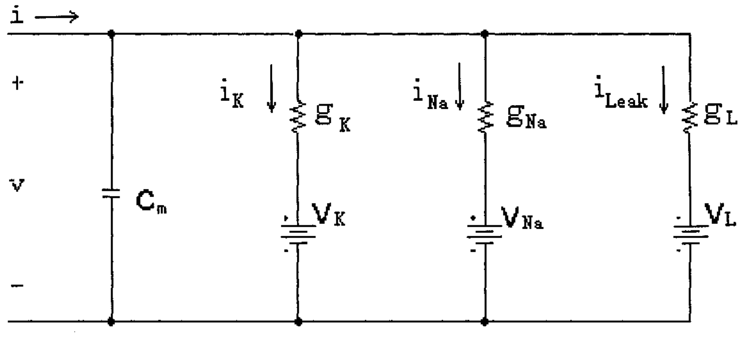


Figure 2-6 Hodgkin-Huxley Model

The parameters in this model are defined by the set of equations [1]:

$$i = C_m \frac{dv}{dt} + i_K + i_{Na} + i_{Leak} \tag{2.1}$$

$$i_{Leak} = (v - V_L)\bar{g}_L \tag{2.2}$$

$$i_K = (v - V_K)\, g_K \tag{2.3}$$

$$i_{Na} = (v - V_{Na})\, g_{Na} \tag{2.4}$$

$$g_K = \bar{g}_K n^4 \tag{2.5}$$

$$g_{Na} = \bar{g}_{Na} m^3 h \tag{2.6}$$

$$\frac{dm}{dt} = \alpha_m (1 - m) - \beta_m m \tag{2.7}$$

$$\frac{dn}{dt} = \alpha_n(1-n) - \beta_n n \tag{2.8}$$

$$\frac{dh}{dt} = \alpha_h(1-h) - \beta_h h \tag{2.9}$$

$$\alpha_m = \frac{-0.1(v+35)}{e^{-0.1(v+35)} - 1} \tag{2.10}$$

$$\alpha_n = \frac{-0.01(v+50)}{e^{-0.1(v+50)} - 1} \tag{2.11}$$

$$\alpha_h = 0.07e^{-0.05(v+60)} \tag{2.12}$$

$$\beta_m = 4e^{-(v+60)/18} \tag{2.13}$$

$$\beta_n = 0.125e^{-(v+60)/80} \tag{2.14}$$

$$\beta_h = \frac{1}{e^{0.1(v+30)} + 1} \tag{2.15}$$

Where,

C_m : Membrane capacitor of unit area (μF/cm^2)

i : Input stimulus current density (μA/cm^2)

v : Membrane potential difference (mV)

i_K : Potassium ions current density (μA/cm^2)

i_{Na} : Sodium ions current density (μA /cm^2)

i_{Leak} : Leakage current density (μA /cm^2)

V_K : Equilibrium potentials of potassium channels (mV)

V_{Na} : Equilibrium potentials of sodium channels (mV)

V_L : Equilibrium potentials of leakage channels (mV)

g_K : Potassium channel conductance of unit membrane area, or permeability of potassium channel (mmho/cm^2)

g_{Na} : Sodium channel conductance of unit membrane area, or permeability of sodium channel (mmho/cm^2)

g_L : Leakage conductance of unit membrane area (mmho/cm^2)

$\bar{g}_K$: Maximum potassium channel conductance of unit membrane area or permeability of potassium channel (mmho/cm^2)

$\bar{g}_{Na}$: Maximum sodium channel conductance of unit membrane area or permeability of sodium channel (mmho/cm^2)

$\bar{g}_L$: Leakage conductance of unit membrane area (mmho/cm^2)

m : Sodium activation factor

n : Sodium inactivation factor

h : Potassium activation factor

By using Matlab, we can plot the membrane potential, channel conductance parameters (i.e. m, n, h) and channel conductance of the Hodgkin-Huxley Model in Figures 2-7, 2-8 and 2-9 respectively. The Matlab code is listed in Appendix A-1.

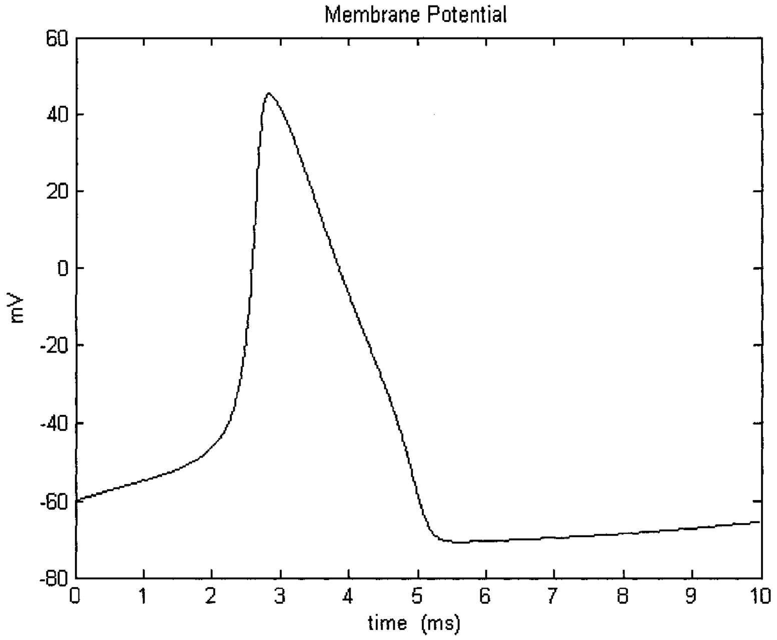


Figure 2-7 Membrane potential vs. time for Hodgkin-Huxley Model

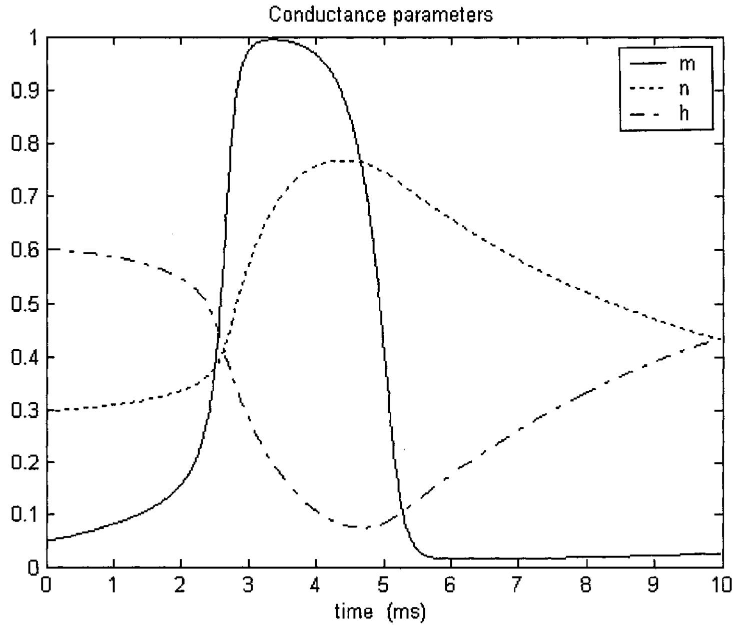


Figure 2-8 Channel conductance parameter vs. time for Hodgkin-Huxley Model

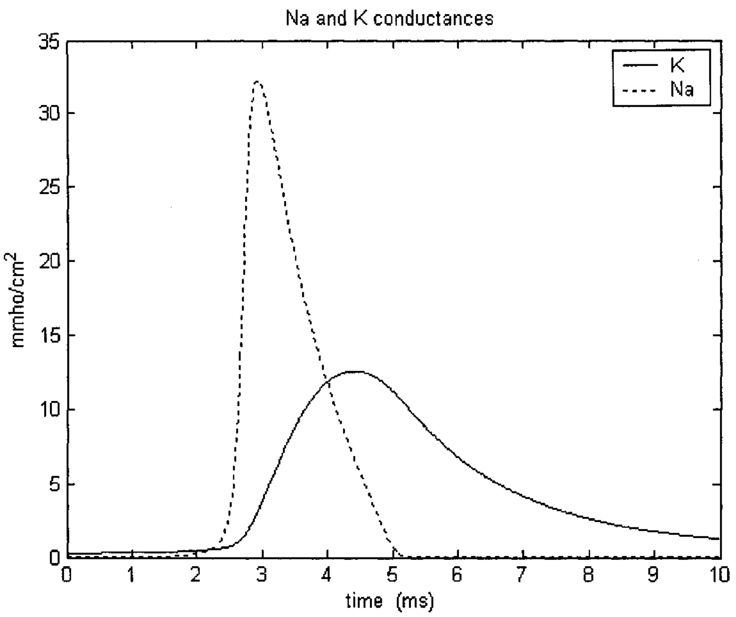


Figure 2-9 Potassium and sodium conductance of Hodgkin-Huxley Model

In the plots above, the time is in the unit of ms; the input current density is set to $5\mu A/cm^2$, and the following numerical values are used, [10]

$$C_m = 1uF/cm^2 \quad V_K = -72mV \quad V_{Na} = 55mV \quad V_L = 49.387mV$$
$$\bar{g}_K\text{=36mmho/cm}^2 \quad \bar{g}_{Na}\text{=120mmho/cm}^2 \quad \bar{g}_L\text{=0.3mmho/cm}^2$$

The initial condition is,

$$v\text{=-60 mV,} \quad \text{m=0.05,} \quad \text{n=0.3,} \quad \text{h=0.6}$$

No matter how complex the Hodgkin-Huxley Model is, we may draw some simple conclusion by carefully examining those equations and the plots above.

- The membrane voltage is simply controlled by three channel conductance namely, g_K, g_{Na}, g_L and the input current, when all other parameter keep constant in a given model.
- The channel conductance of Na^+ and K^+ vary with time, and they are strongly

related to the membrane potential.

- The variation of Na^+ and K^+ channel conductance follow different curve, the combinational effect is to generate an AP pulse.

By comparison, we see that the Hodgkin-Huxley model perfectly matches the natural behavior of a neuron cell

2.6 Network behavior of neural system

Even though the network behavior of a neural system is not related to our design, for completeness, it is briefly introduced in this section.

As mentioned in previous section, every neuron cell is connected to other cells by synapse, and synapse weight determines the coupling efficacy. Donald Hebb postulated that the learning behavior of creature principally involves altering the synapse weight [11] [12].

To make this point clear, let us assume that we have a two ports three layers network, and each layer is composed of a couple of nodes (see Figure 2-10).

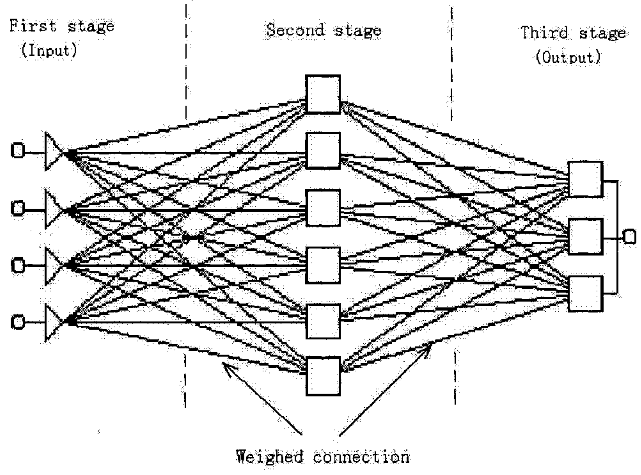


Figure 2-10 A three layer network coupled by weighted connections

The nodes in different layers are coupled together, and the coupling efficacy is adjustable. It is easy to see, by adjusting the coupling efficacy of each connection, there will be almost infinite mapping relation from the input port to the output port. Similarly, in the natural neural system, its structure may be considered as a multi-layer multi-nodes network, and the nodes in different layer are coupled by synapse. By changing the synapse weight, the mapping relation of the network is modified so that learning behavior is achieved.

One example is the famous Pavlovian conditioned reflex experiment. In this experiment the sound of the bell is the input and the salivating response of the dog is the output. The initial mapping relation of these two things in the dog's brain might be something else. After proper training, the synapse weight between the appropriate part of the auditory cortex and the salivation glands are strengthened, and the mapping relation is finally modified to 'the sound of the bell --- salivate'. So, anytime the bell rings, the dog starts to salivate even though there is no food presented.

2.7 System diagram of our ADC system

Based on the study of the natural neurons, many theories and techniques about reproducing the behavior of neurons emerged since 1970s. These neurons are called artificial neurons (AN). Some of them imitate the behavior of a single neuron to implement systems with superior performance compared with other methods of implementation. Many others focus on the network behavior of a neural system, which leads to aritifical neural network (ANN) theory.

In Section 2.4, we came to the understanding that there exists an A/D conversion operation in a natural neuron cell. Furthermore, from Section 2.5, the Hodgkin-Huxley Model tells us that the time variant and membrane voltage dependent K^+ and Na^+ channel conductance are responsible for generation of an AP pulse. Inspired by this knowledge, we propose an electronic A/D converter (ADC) structure.

In a natural neuron cell, the stimulus (i.e., input) signal is transferred into a bunch of AP pulses, and the density of the pulses carries the information of the input signal (see Section 2.4). Based on this process, we propose a 6-bits unipolar ADC whose basic structure is shown in Figure 2-11. It mainly consists of two parts, namely Neuron Cell Part, Counter Part, and two controlling clocks, say CLK_1, CLK_2.

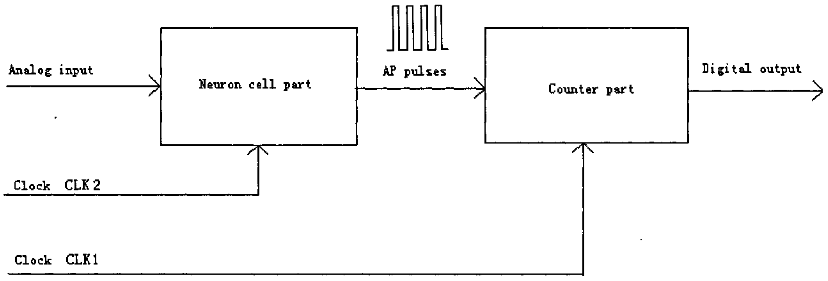


Figure 2-11 Basic structure of our ADC

Before we discuss the details of the circuit, we shall introduce the two clocks first. Unlike other ADC, our design does not have explicit Sampling and Holding (S/H) circuit. However, the sampling concept does exist implicitly in our ADC through the two clocks. From Figure 2-12, we see that these clocks are opposite in phase to each other but have the same period, which is just the sampling period for our ADC (from now on, we call it $T_{sampling}$). CLK_1 is used to reset the Counter Part, while CLK_2 is used to reset the Neuron Cell Part at the beginning of each sampling period. More detail is given in subsequent chapter.

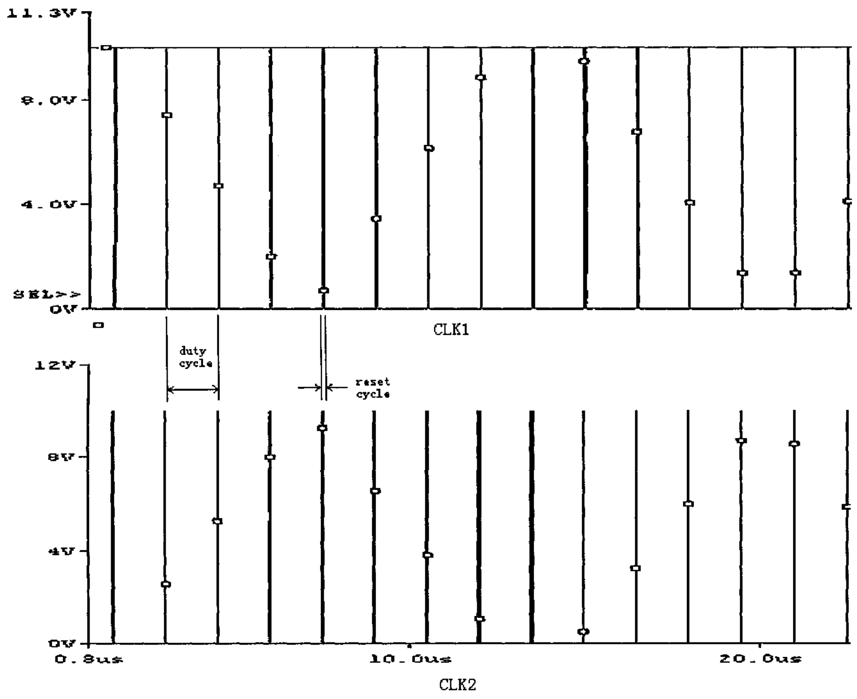


Figure 2-12 Controlling clock CLK_1 (top) and CLK_2 (bottom). The period of these two clock are same and it is the sampling period for our ADC

In the Neuron Cell Part, an analog input signal is transferred to a bunch of pulses (we call it as AP pulses following the convention of natural neuron) whose density is directly proportional to the amplitude of input signal. The process is similar to what a natural neuron does in its trigger zone. So, at the output of Neuron Cell Part, the information is carried by the density of AP pulses. That is,

$$density_{\_of\_AP\_pulse} \propto ampliaude_{\_of\_input}$$

In the Counter Part, the information carried in the density of AP pulses is extracted and transferred to a binary digital word at its output. In other words, the counter counts the number of AP pulses in each duty cycle (see Figure 2-12) of CLK$_1$. It is clear that the number that the counter counts is proportional to the density of AP pulse. So, we have

$$number_{\_of\_AP\_pulses\_in\_cons\tan t\_\mathrm{int}\,erval} \propto density_{\_of\_AP\_pulse} \propto ampliaude_{\_of\_input}$$

The function of ADC is achieved by virtue of the above relation.

2.8 Summary

In this chapter, we learnt that the signal transmitted in the natural neural system is mainly electrical. When the stimulus is applied, an analog signal is generated at the dendrite. After a biological A/D conversion, a bunch of AP pulses is created whose density is proportional to the magnitude of the stimulus. Then, the AP pulses propagate along the axon and their shapes do not change any more. Hodgkin-Huxley model mathmatically described this genernation process of an AP pulse. When many neurons are vastly connected, by modifying the synapse weight, some network behavior such as learning can be achieved. Based on the study of the natural neurons, a basic unipolar ADC structure has been introduced. This consists of a Neuron Cell part, a Counter Part, and two controlling clocks.

Chapter 3

Neuron cell implementation

In this chapter, we first introduce the basic structure and the mechanism of operation of our ADC in section 3.1. In section 3.2, we derive its transfer characteristic and introduce the area-sampling method that is adopted in our design. Having calculated the value of the each component in section 3.3, we present the complete schematic of Neuron Cell part in section 3.4. Finally, we analyze the performance of this part by simulation result in section 3.5

3.1 Mechanism of operation

In section 2.7, we have mentioned that the goal of the Neuron part is to generate AP pulses and make the density of AP pulses directly proportional to the amplitude of the input (stimulus) signal.

To construct a neuron cell, based on the Hodgkin-Huxley model, many circuits have been proposed; some of them try to exactly mimic the natural neuron cell's behavior by individually simulating the Na^+, K^+ channel [17] [18]. This approach is more useful in biomedicine application, but the circuit is usually too complex and is not attractive for low power and compact design. The other approach is to functionally resemble the natural neuron cell's behavior, based on their specific goal. In our design, we follow the latter approach.

Figure 3-1 depicts our proposed Neuron Cell which follows an earlier work [20], but with slight modification. Here, the input is analog current and the output is voltage pulses. C_1 is the membrane capacitor; U_1 and U_2 are two inverters; M_1 and M_2 transistors function as switches.

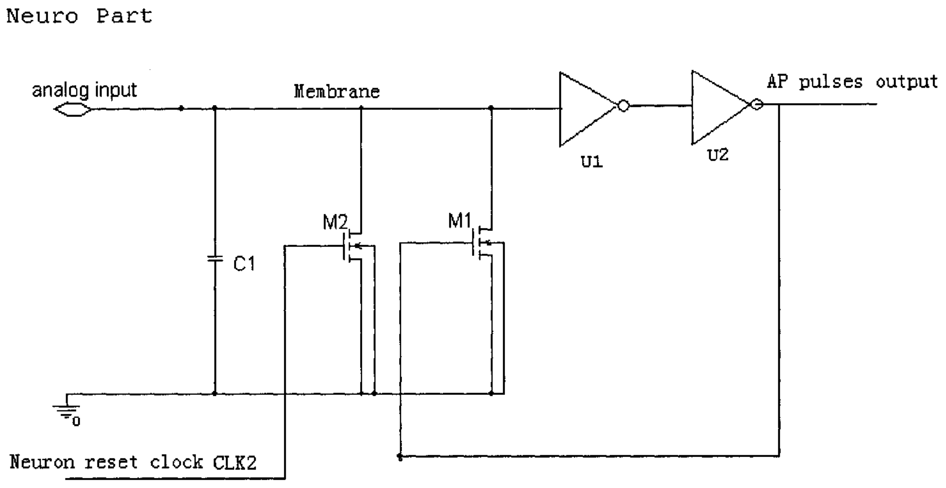


Figure 3-1 Neuron Cell Part

The operation of this circuit is simple. At the end of every sampling period (see section 2.7), the circuit is set to its initial state, that is, M_1 and M_2 remain off. C_1 is charge free and the output of this part (output of U_2) is zero. So, at the beginning of next sampling period, the input current charges C_1, causing the voltage of the membrane to increase from its initial point (see Figure 3-1). As this voltage reach the threshold of U_1, the output of U_1 flips down, and the output of U_2 flips up. The high output voltage of U_2 is then fed back to turn switch M_1 on so that M_1 completely discharges C_1 and bypass all the input current. This causes the input of U_1 dropping down and the output of U_1 flipping up. Consequently, the output of U_2 drops down. Then, M_1 is shut off, and the circuit is in its initial state again. In this process, the output of U_2 flips up first and then flips down, so, a pulse (AP pulse) is generated. The circuit repeats this process to generate more AP pulses until this sampling period is over. At the end of each sampling period, the reset clock CLK_2 resets the Neuron Cell Part. It turns on switch M_2, which completely discharge C_1 and bypass all the input current so that the Neuron Cell part is in its initial state and be ready for next sampling period. At the same time, CLK_1 (see Figure 2-11) reset the Counter Part to start a new counting progress.

For convenience, we name the interval for the membrane voltage to increase from initial point to the threshold of U_1 as 'Integral step' (referred as t_{integral}), and the interval for the membrane voltage to go from the threshold of U_1 to next initial point as 'Pulse step' (referred as t_{pulse}, See Figure 3-2 for detail).

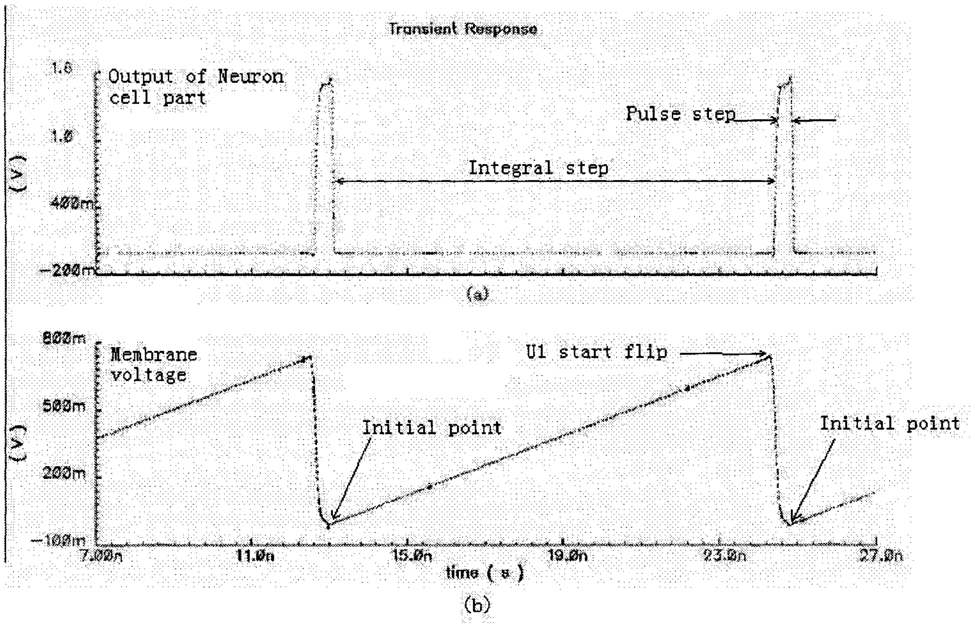


Figure 3-2 (a) The output of Neuron Cell Part; (b) voltage curve of Membrane

If we compare our Neuron Cell part with the Hodgkin-Huxley Model (see Section 2.5), we see that in Hodgkin-Huxley Model, an AP is generated mainly by two time variant and membrane voltage dependent channel conductance, similarly, the conductance of switch M_1 in our circuit is also time variant and membrane voltage dependent. So, the switch M_1 is equivalent to the combination of Na^+ and K^+ channel conductance. Even though it does not exactly match the equation of Hodgkin-Huxley Model, it functionally accomplishes this job, namely to generate AP pulses depending the input current. This is considered enough for design of the simple ADC.

3.2 Transfer Characteristic

In this section, we will analyze the transfer relation of this circuit, namely, the relation between the density of output AP pulses and the amplitude of input current. Let us start from a simple case in which the input is a DC current. At the initial point, the two switches M_1 and M_2 are both off, that is all the input current will charge capacitor C1.

$$i_{input} = C_1 \frac{dv_{Membrane}}{dt} \tag{3.1}$$

By rearrangement,

$$\frac{dv_{Membrane}}{dt} = \frac{i_{input}}{C_1} \tag{3.2}$$

Where,

i_{input} is the input current.

$v_{Membrane}$ is the voltage of the membrane, which is also the input voltage of inverter U1.

Eq. (3.2) means that the rate that the membrane voltage increases is proportional to the input current. Since we assume that the input is a DC current, $\frac{dv_{Membrane}}{dt}$ is constant.

So, the time interval for each integral step is,

$$t_{\text{int}egral} = \frac{V_{threshold\_of\_U1} - 0}{\frac{dv_{Membrane}}{dt}} \tag{3.3}$$

Plugging Eq. (3.2) into Eq. (3.3), and after rearrangement, we get

$$t_{\text{int}egral} = C_1 \frac{V_{threshold\_of\_U1} - 0}{i_{input}} \tag{3.4}$$

Since for a given circuit, $V_{threshold\_of\_U1}$ is constant, so, we get a reverse proportional relation between $t_{\text{int}egral}$ and i_{input} from Eq. (3.4).

If we ideally assume the width of each AP pulse, namely t_{pulse} is zero, the density of AP pulses is equal to the number of AP pulses divided by $T_{sampling}$, that is

$$D_{ensity\_of\_AP\_pulses} = \frac{N_{umber\_of\_pulses\_in\_one\_sampling\_period}}{T_{sampling}} \tag{3.5}$$

And clearly,

$$N_{umber\_of\_pulses\_in\_one\_sampling\_period} = \frac{T_{sampling}}{t_{\text{int}egral}} \tag{3.6}$$

Plugging Eq. (3.4) and Eq. (3.6) into Eq. (3.5), we get,

$$D_{ensity\_of\_AP\_pulses} = \frac{1}{t_{\text{int}egral}} = \frac{i_{input}}{C_1(V_{threshold\_of\_U1} - 0)} \tag{3.7}$$

From Eq. (3.7), we see that the density of AP pulses is directly proportional to the amplitude of input current, when the input is DC current. Next, let us consider a more general situation.

One should note that in above analysis, we assumed t_{pulse} is zero. We will continue this assumption until chapter 5, in which we analyze the effect of the non-zero t_{pulse}.

As what we mentioned in Section 2.7, our ADC does not have an explicit Sampling and Holding (S/H) circuit, but the sampling concept does exist in our design. Figure 3-3 shows two sampling methods. Figure 3-3 (a) is the ordinary sampling method. It samples the input signal value at every sampling point, namely point (i), point (ii), point (iii) etc. then the sampled value is held for subsequent processing. So we call this method as Point-sampling. Unlike point sampling, we use Area-sampling method in our ADC.

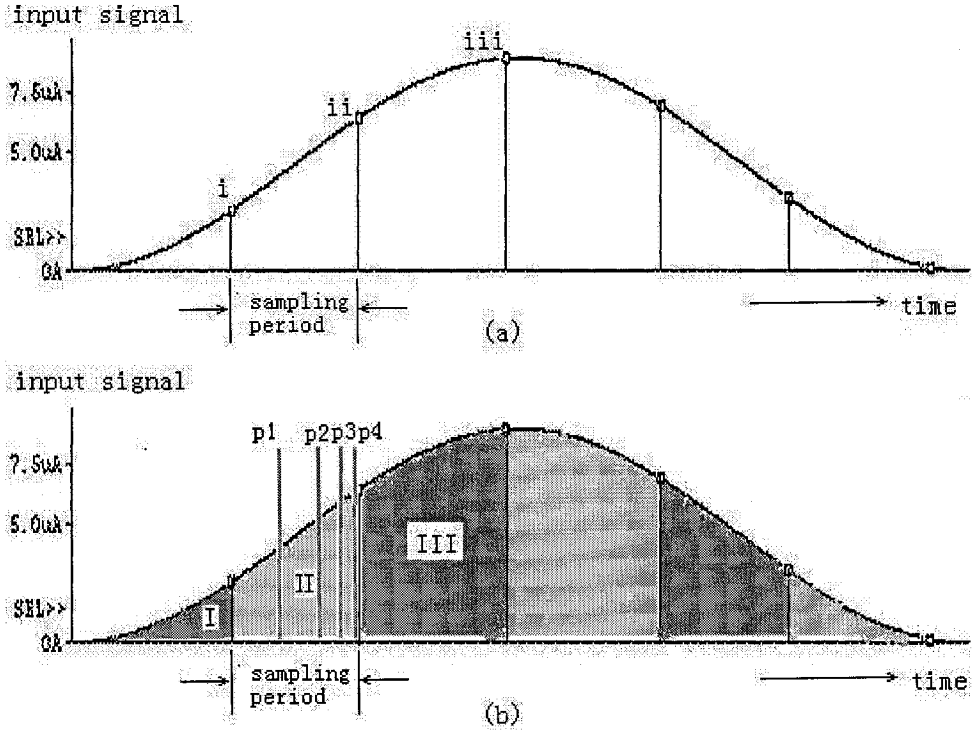


Figure 3-3 Comparison of two sampling method. (a) Point-sampling method (b) Area-sampling method

We know that the Neuron Cell Part generates a couple of AP pulses depending on the amplitude of input current in each sampling period. For example, in the sampling period II of Figure 3-3 (b), four AP pulses are generated. Furthermore, we know that for each AP pulse, the Membrane voltage will increase

$$\Delta V = V_{threshold\_of\_U1} - 0 = V_{threshold\_of\_U1} \tag{3.8}$$

So, in one sampling period the total increment of Membrane voltage is

$$\Delta V_{total\_due\_to\_input\_current} = N \cdot V_{threshold\_of\_U1} \tag{3.9}$$

Where, N is the number of AP pulses generated in one sampling period.

One should note that Eq. (3.9) gives the total increment of the membrane voltage merely due to input current. Actual membrane voltage will not be that high since at each Pulse step, the switch M1 will discharge C1 to drop membrane voltage down to zero (see Figure 3-2).

Here, since we continue assuming $t_{pulse} = 0$, so, from Eq. (3.9), we get,

$$\frac{1}{C_1} \int_0^{T_{smpling}} i_{input} dt = \Delta V_{total\_due\_to\_input\_current} = N \cdot V_{threshold\_of\_U1} \tag{3.10}$$

By rearrangement,

$$N = \frac{1}{C_1 \cdot V_{threshold\_of\_U1}} \int_0^{T_{smpling}} i_{input} dt \tag{3.11}$$

The density of AP pulses in each sampling period is,

$$D_{ensity\_of\_pulse} = \frac{N}{T_{sampling}} = \frac{1}{C_1 \cdot V_{threshold\_of\_U1} \cdot T_{sampling}} \cdot \int_0^{T_{smpling}} i_{input} dt \tag{3.12}$$

Where, $\int_0^{T_{smpling}} i_{input} dt$ is actually the area of under the input current within each sampling period in Figure 3-3, and Eq. (3.12) says that the density of AP pulses is proportional to this area. This is the source that the name of Area-sampling comes from. This area is also equal to the charge injected into the Neuron Cell Part during each sampling period, which is named as

$$Q_{input} = \int_0^{T_{smpling}} i_{input} dt$$

Then, Eq. (3.12) may be rewritten as,

$$D_{ensity\_of\_pulse} = \frac{1}{C_1 \cdot V_{threshold\_of\_U1} \cdot T_{sampling}} \cdot Q_{input} \tag{3.13}$$

By Nyquist principle, the sampling rate (reciprocal of sampling period) must be at least two times of the maximum frequency component of the input signal. To reduce the quantization noise, usually 3 to 20 times is recommended [28]. With a high (ten times or more of the Nyquist rate) sampling rate it is reasonable to assume that the input signal varies linearly over each sampling period. Then we can write:

$$i_{input\_i} = kt + Z_i \tag{3.14}$$

Where,

i_{input_i} is the input current in ith sampling period,

Z$_i$ is a constant, which is the value of input current at the beginning of ith sampling periods.

k is the slope, and is treated as constant.

t is the time measured from the beginning of the current sampling period.

Plugging Eq. (3.14) into Eq. (3.12), we get

$$D_{ensity\_of\_pulse} = \frac{N}{T_{sampling}} = \frac{1}{C_1 \cdot V_{threshold\_of\_U1}} \cdot (\frac{1}{2} kT_{sampling} + Z_i) \tag{3.15}$$

In Eq. (3.15), $(\frac{1}{2} kT_{sampling} + Z_i)$ is Q_{input}, and is equal to the middle point value of the input current in the ith sampling period. So, unlike the ordinary point-sampling method that samples the input current value at each sampling time point, in our design, the area sampling method samples the input charge within each sampling period, which is proportional to the middle point value of the input current in each sampling period.

In conclusion, the period of CLK_1 and CLK_2 is the sampling period of our ADC (as explained in section 2.7 of the thesis) Without S/H circuit, our ADC continually samples the input signal in each sampling period and transfers the amplitude information of the input current to the density of AP pulses. Essentially, our ADC samples the charge injected into the Neuron Cell part in each sampling period.

3.3 Calculation of the components of the Neuron Cell Part

From the description above, it is clear that for an ADC with resolution of B bits, the number of the pulses that the Neuron Cell Part generates in each sampling period should be in the range from 0 to $2^B - 1$. So, in case that the value of the input current is equal to its Full Scale Range (abbreviated as FSR, it is the maximum value of the input signal before overload), the Neuron Cell Part should generate the maximum number, namely $2^B - 1$. There will be ($2^B - 1$) AP pulses in one sampling period. Hence, the time interval of the integral step for each pulse will reach the minimum value,

$$t_{\text{integral\_min}} = \frac{T_{sampling}}{2^B - 1} \tag{3.16}$$

If this ADC is designed to convert an input signal with frequency up to F Hz, and we set the sampling rate as M times higher than F, then we get,

$$T_{sampling} = \frac{1}{F \cdot M} \tag{3.17}$$

Plugging Eq. (3.17) into Eq. (3.16)

$$t_{\text{integral\_min}} = \frac{1}{(2^B - 1) \cdot F \cdot M} \tag{3.18}$$

Then, plugging Eq. (3.4) into Eq. (3.18), we get

$$C_{1\_\max}\frac{V_{threshold\_of\_U1}-0}{i_{input\_FSR}}=\frac{1}{(2^B-1)\cdot F\cdot M} \tag{3.19}$$

By re-arrangement,

$$C_{1\_\max}=\frac{i_{input\_FSR}}{(2^B-1)\cdot F\cdot M\cdot V_{threshold\_of\_U1}} \tag{3.20}$$

Where,

i_{input_FSR} is the FSR value of the input current

$C_{1\_\max}$ is the maximum value of the membrane capacitor C_1 in the given situation.

Eq. (3.20) offers the formula to determine the value of C_1 when FSR of the input current, maximum input frequency, and the resolution of the ADC are given. In this design, our goal is to make a 6 bits ADC, which may convert an input signal with the frequency up to 20kHz, and the input range is from 0 to 10 μA. The power supply is 1.5V the value of M is no less than 30.

Plugging this value into Eq. (3.20), we get

$$C_{1\_\max}=\frac{10\mu}{(2^6-1)\times 20k\times 30\times 0.75}=353f\text{F} \tag{3.21}$$

In our design, we use 200 fF as the capacitance of C1. This value makes M greater than 30, namely 52, which may lower the quantization noise [28].

From Eq. (3.17), we get

$$T_{sampling}=\frac{1}{F\cdot M}=\frac{1}{20k\times 52.5}=0.95\mu\text{s} \tag{3.22}$$

The design parameter is summarized in table 3-1

Table 3-1 Design parameters of our ADC

Power supply voltage	1.5V
Input current range	0 to 10 μA
Maximum frequency of Input current	20kHz
Membrane capacitor C_1	200 *f*F
Sampling period	0.95μs

3. 4 Improving the shape of the AP pulse

In previous discussion, we get a linear relation between the density of AP pulses and the amplitude of input current based on the ideal assumption of $t_{pulse} = 0$. The non-ideal situation will be analyzed in chapter 5. So far, we just keep in mind that shorter is the pulse width (the smaller t_{pulse}), better is linearity performance.

However, too short pulse width might cause problem due to the non-ideal performance of the inverter. Unlike ideal inverter, the parasitic components of the practical inverter will produce a distortion in the shape of the output when the input varies across its threshold (i.e. the output cannot change abruptly but has a finite slewing rate (see Figure 3-4). So, if the pulse width is too short, the peak value of the distorted (i.e., triangular) pulse might fail to reach its '1' amplitude, say positive rail voltage, and thus shall not exceed the necessary threshold value. The distorted pulse will bring two major problems. First, it will cause the next stage, namely the Counter part to malfunction; next, as mentioned before, during each pulse step, the AP pulse is fed back to turn on the switch M_1 so that it completely discharge C_1 and bypass all the input current (see Section 3.1). If a distorted pulse is fed back, M_1 may not accomplish this job properly and cause the Neuron Cell part malfunction (C_1 is not completely discharged, and the next AP pulse does not start from the initial point. This will destroy the linear relation).

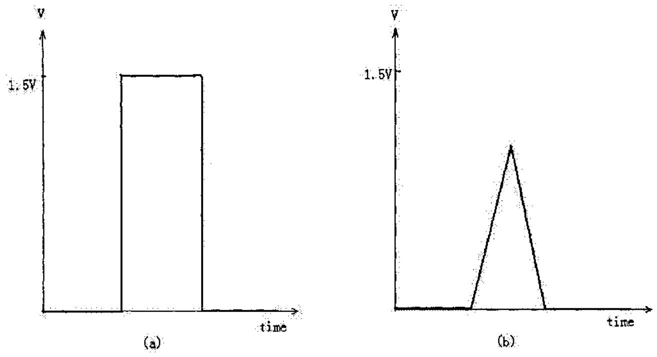


Figure 3-4 (a) An ideal AP pulse, (b) A distorted AP pulse due to practical performance of the non-ideal inverters

From the above analysis, we see that to achieve good linearity and guarantee reliable operation of the whole system, we need AP pulses with sharp rising and falling edges (or large slewing rate).

DC sweep result shows that a practical inverter is like a voltage amplifier in the transition region (see Figure 3-5), namely

$$v_{out} = A \cdot v_{in} \tag{3.23}$$

Where, A is the gain

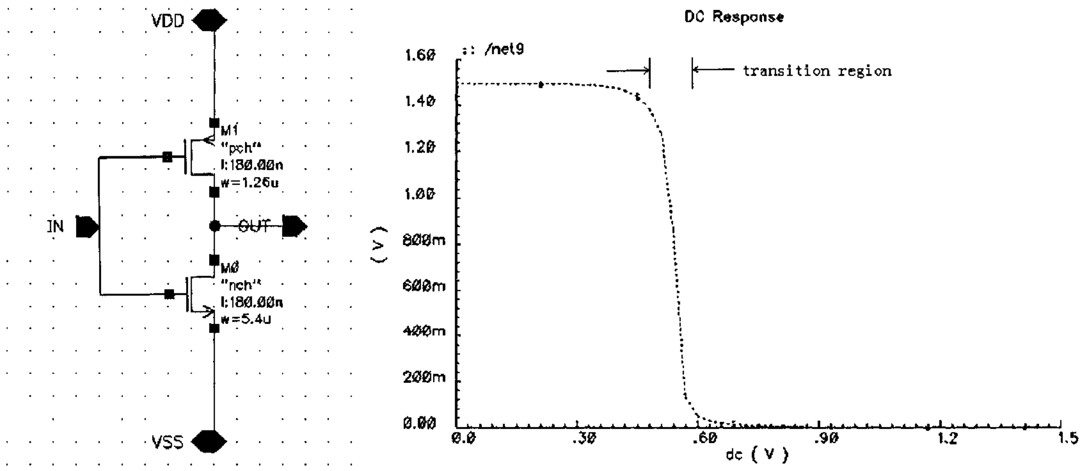


Figure 3-5 The inverter and its DC sweep result

We also know from previous analysis,

$$\frac{dv_{Membrane}}{dt} = \frac{1}{C_1} \cdot i_{input} \tag{3.24}$$

So, the slewing rate of the AP pulse in Figure 3-1 is

$$\frac{dv_{AP}}{dt} = A^2 \cdot \frac{dv_{Membrane}}{dt} = A^2 \cdot \frac{1}{C_1} \cdot i_{input} \tag{3.25}$$

Where, the square sign of A^2 is because there are two inverters in cascade (see Figure 3-1).

From Eq. (3.25), the slewing rate of the AP pulse is dependent on the gain of the inverter in its transition region and also on the number of inverters in cascade. We could increase the gain by enlarging its size, however, this effect usually is limited and will cause large parasitic capacitor problem. So, we put more inverters to improve the slewing rate (to get sharper rising and falling edge). One thing should be mentioned here is that even though more inverters may achieve better slewing rate, it also increase the width of the AP pulse since the width is equal to the delay of the inverter chain. Thus linearity will be impaired. We experimented with several inverters in cascade and found that eight (i.e., four pairs) of inverters produced a satisfactory result. The complete schematic of the Neuron Cell Part is shown in Figure 3-6. Figure 3-7 shows the enhancement of the performance on slewing rate when more inverters are used.

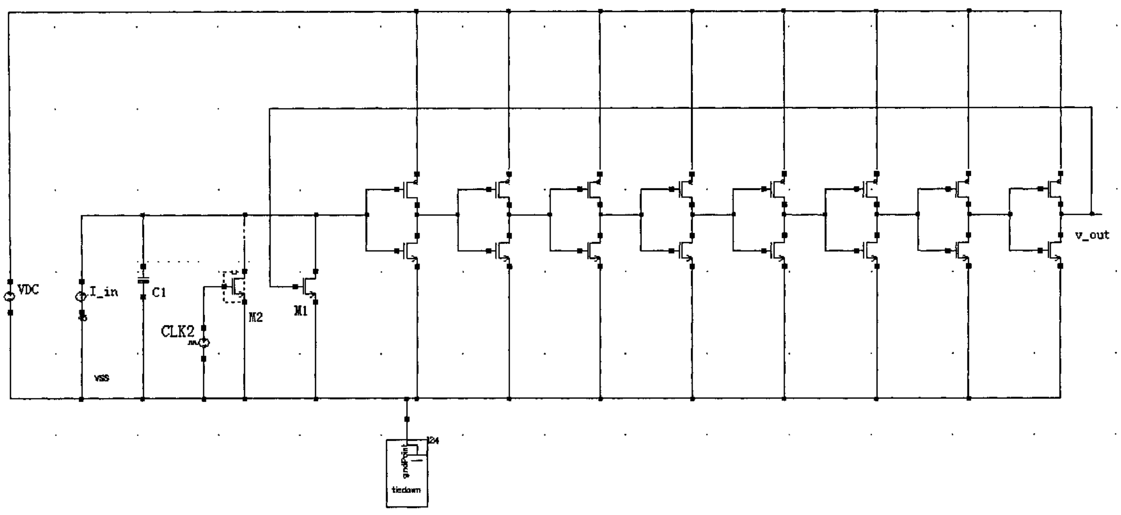


Figure 3-6 complete schematic of Neuron Cell Part

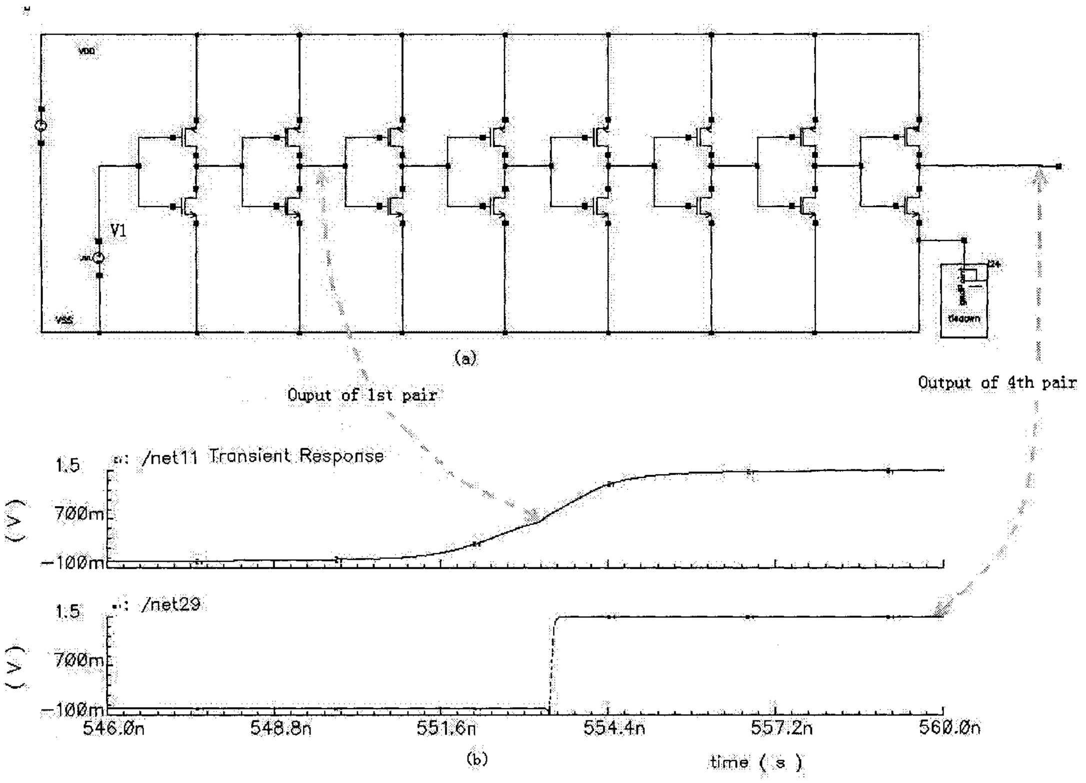


Figure 3-7 Transient simulation results to show the enhancement of 4 pairs of inverter. (a) Schematic of the testing circuit in which V_1 is a linearly increasing voltage source with increasing rate of 1V/μs. (b) Transient simulation result of the output of 1st pair inverter whose slewing rate is 400V/ μs, and the output of 4th pair inverter whose slewing rate is 18KV/ μs.

3.5 Performance of the Neuron Cell part

To see the performance of the Neuron Cell Part, DC currents with two different amplitudes are applied at the input of the Neuron Cell Part, and the simulation results are shown in Figures 3-8 to 3-11.

From these simulations, we can verify the followings:

- According to Eq. (3.22) the sampling period is to be around 0.95us. Figure 8 (b) and Figure 10 (b) confirm this point. This is the sampling period of our ADC.
- From both Figure 3-8 (a) and Figure 3-10 (a), we see that the AP pulses cease to occur at the time when CLK_2 pulse exists. The reason is that the clock CLK_2 resets the Neuron Cell Part through switch M_2 (see Figure 3-6 for reference).
- From Figure 3-8 (a), we see three pulses in each sampling period with an input current level of 0.5 μA. Figure 3-10 (a) shows 36 pulses existing in one sampling period. This happens with an input current level of 5 μA.The ratio of the numbers of these pulses is very close to the ratio of the amplitude of the two input current. This roughly proof the linearity of our ADC
- Figures 3-9 and 3-11 show a zoomed view of one AP pulse when the input current is 0.5 μA and 5 μA respectively. From this we see that both of them have pulse width around 0.37ns, and the shapes are good, namely the pulse has sharp rising and falling edge and its peak reach the positive rail voltage. This substantiates the improvement that we discussed in section 3.4

mentioned before.

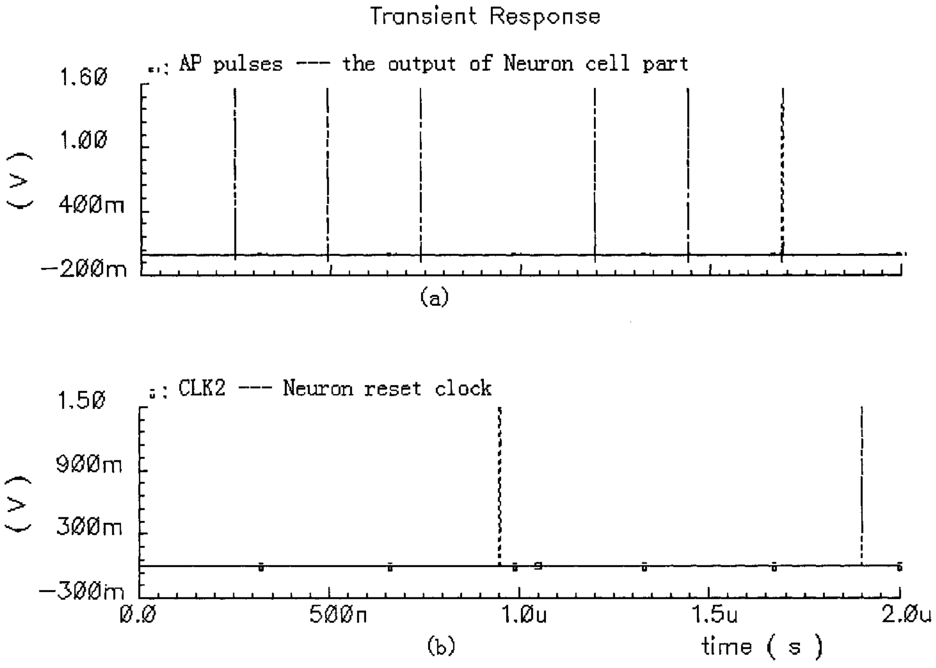


Figure 3-8 Simulation result of Neuron Cell Part, when the input current is 0.5 μA. (a) Waveform of AP pulses. There are three pulses in each Sampling period (b) Waveform of CLK2. The time interval between two pulses constructs a sampling period.

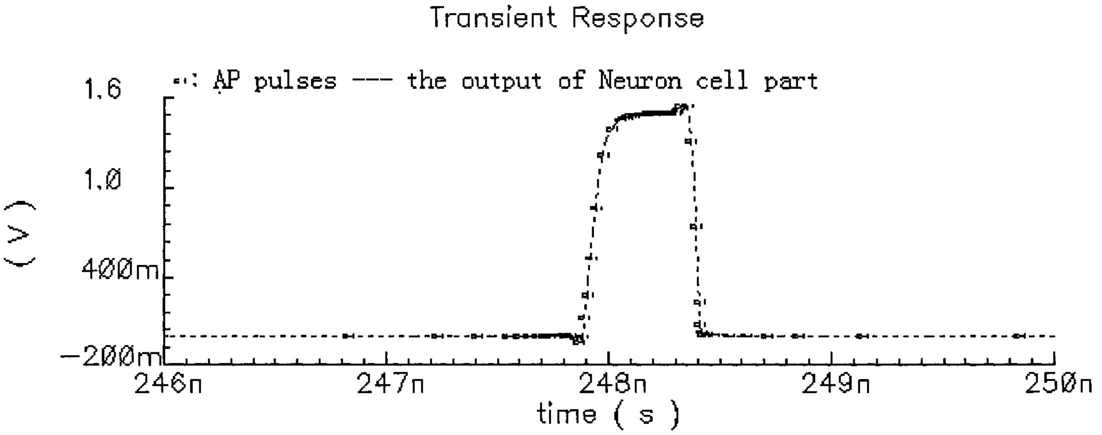


Figure 3-9 The waveform of one AP pulse when the input current is 0.5 μA.

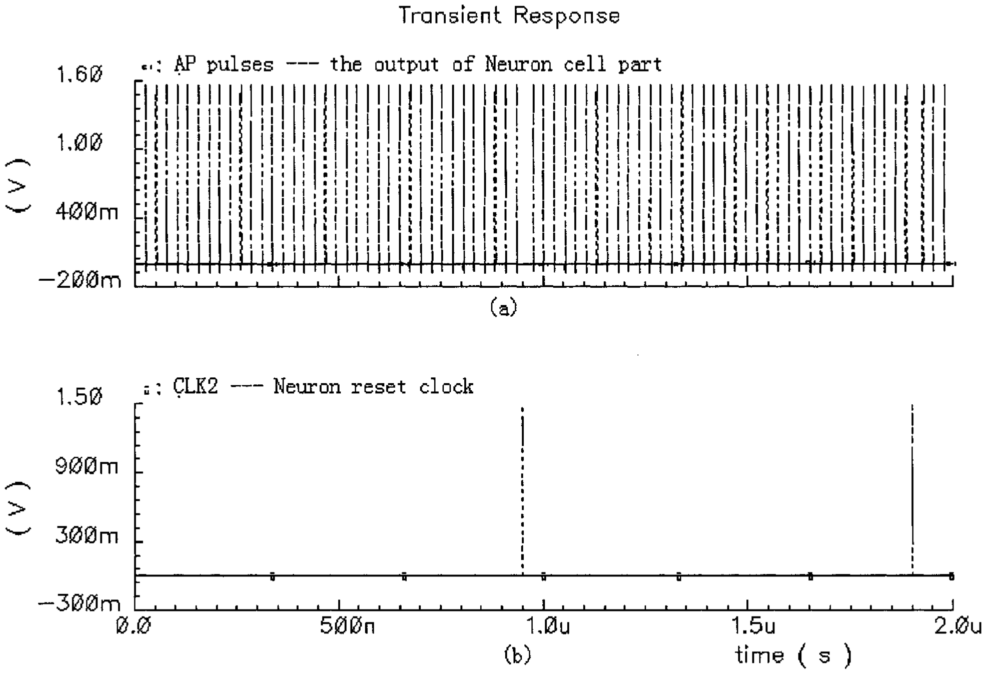


Figure 3-10 Simulation result of Neuron Cell Part, when the input current is 5 μA. (a) Waveform of AP pulses. There are 36 pulses in each Sampling period (b) Waveform of CLK2. The time interval between two pulses constructs a sampling period.

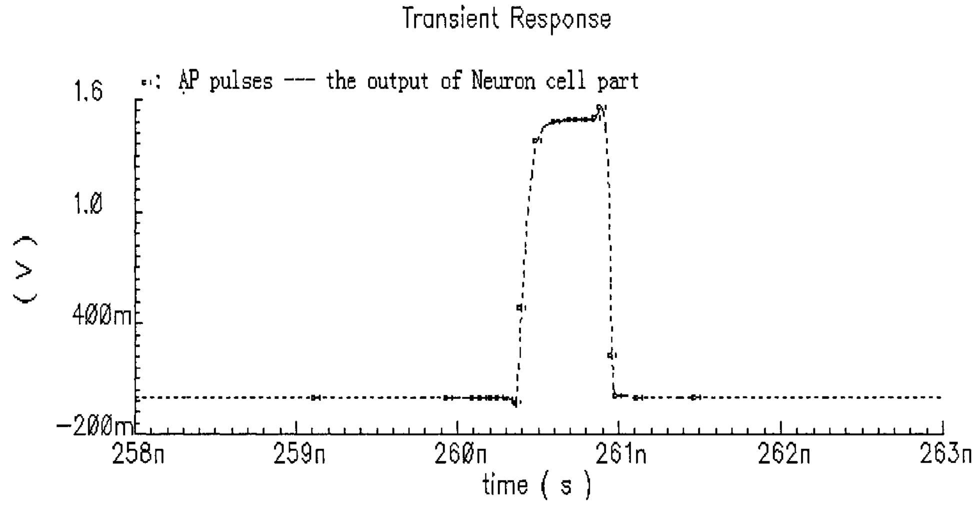


Figure 3-11 The waveform of one AP pulse when the input current is 5 μA.

3.6 Summary

An electronic neuron functionally resembling the Hodgkin-Huxley neuron model, has been presented. It is featured by the compact structure and the simple operation. Instead of point-sampling method, area-sampling method is adopted in this design (some important advantage of this method will be clear in chapter 5). The validity of this sampling method is proved by the analysis that the output is linearly related to the input. Furthermore, the influence of shape of the AP pulse and a method to improve it are also introduced. Finally, the simulation result show satisfactory performance.

Chapter 4

The Counter Implementation

In this chapter, we first introduce a typical asynchronous counter structure in section 4.1. Next, based on our specific design situation, we propose a new counter structure in section 4.2. It uses fewer transistors than the general asynchronous counter.

4.1 Typical Asynchronous Counter

For the counter part, we have many handy choices, such as, asynchronous counter, synchronous counter etc. From the point of view of less power consumption, asynchronous counter would be a better choice since it uses less digital gates and consumes less power [30]. A typical 6-bits asynchronous counter is shown in Figure 4-1, in which each bit is constructed by a T-latch and a pulse detector. The input of the rear bit is connected to the node Q of previous one [34].

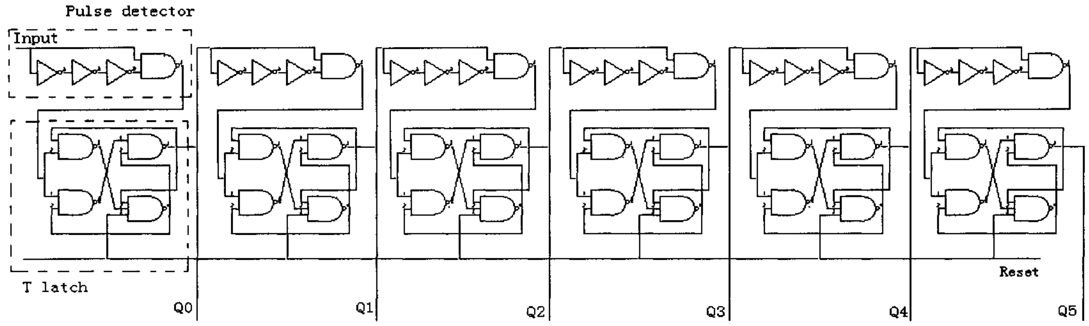


Figure 4-1 A typical 6-bits asynchronous counter composed by T flip-flops

If the counter is composed by T-Latch alone without the pulse detector (see Figure

4-2), it will work well only when the pulse width is narrow (see Figure 4-3). It will malfunction in case that the input pulse width is beyond certain amount (it is called trigger-limit time, and is equal to 0.22ns for our circuit by simulation).

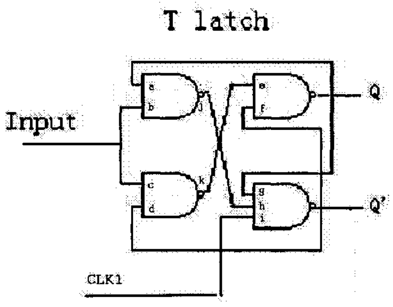


Figure 4-2 A one-bit counter composed only by a T-latch

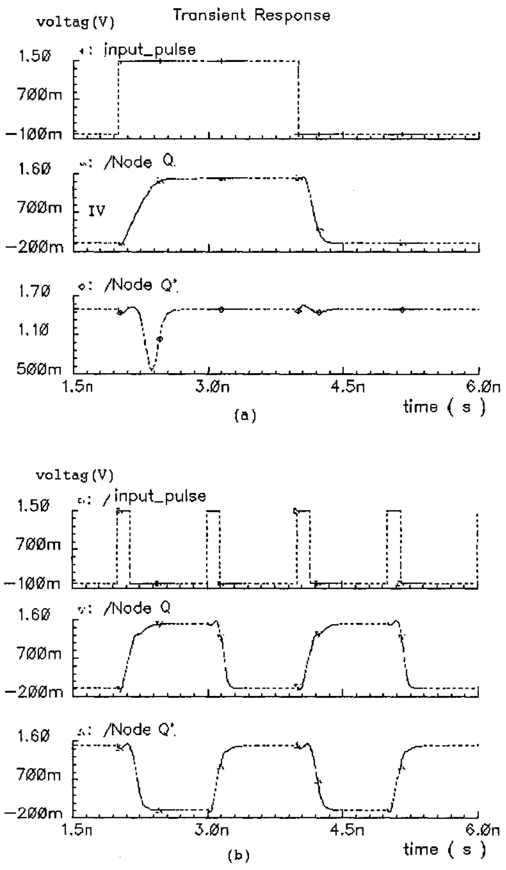


Figure 4-3 (a) The counter without the pulse detector mal-functions when the input pulse width is over the triggering limit time; (b) the counter without the pulse detector works well when the input pulse width is shorter the triggering limit time.

Figure 4-3 (a) shows the wrong output of the counter without the pulse detector when the input pulse width is over the trigger-limit time; Figure 4-3 (b) shows the correct output of the counter without the pulse detector when the input pulse width is shorter than the triggering limit time.

So, the pulse detector is necessary in common case. It detects either rising or falling edge of the input pulse (depending on it is positive or negative edge pulse detector), and generates narrow pulses correspondingly (Figure 4-4). In other word, the pulse detector ensures that the width of input pulse is less than the trigger-limit time.

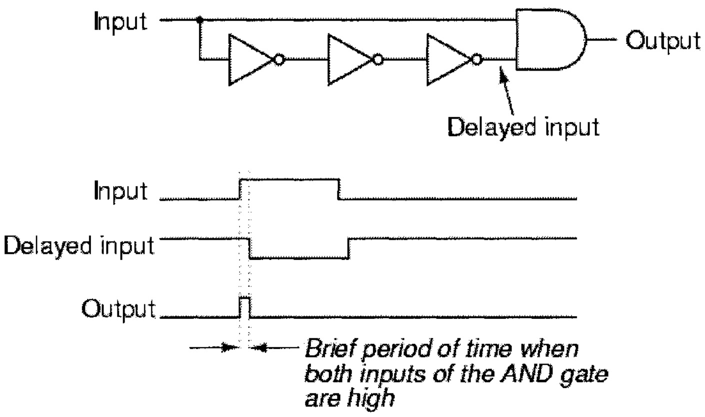

Input
Output
Delayed input
Input
Delayed input
Output
Brief period of time when both inputs of the AND gate are high

Figure 4-4 A positive edge pulse detector

Figure 4-4 shows a positive edge pulse detector, in which the number of the inverters must be odd. The total propagation delay of the inverter chain determines the width of the output pulse. Simulation result shows that one inverter cannot give good output pulse since it does not offer enough propagation delay, while three inverters may achieve acceptable performance. From Figure 4-5, we see that the output of a Pulse detector with only one inverter does not even reach the positive rail voltage.

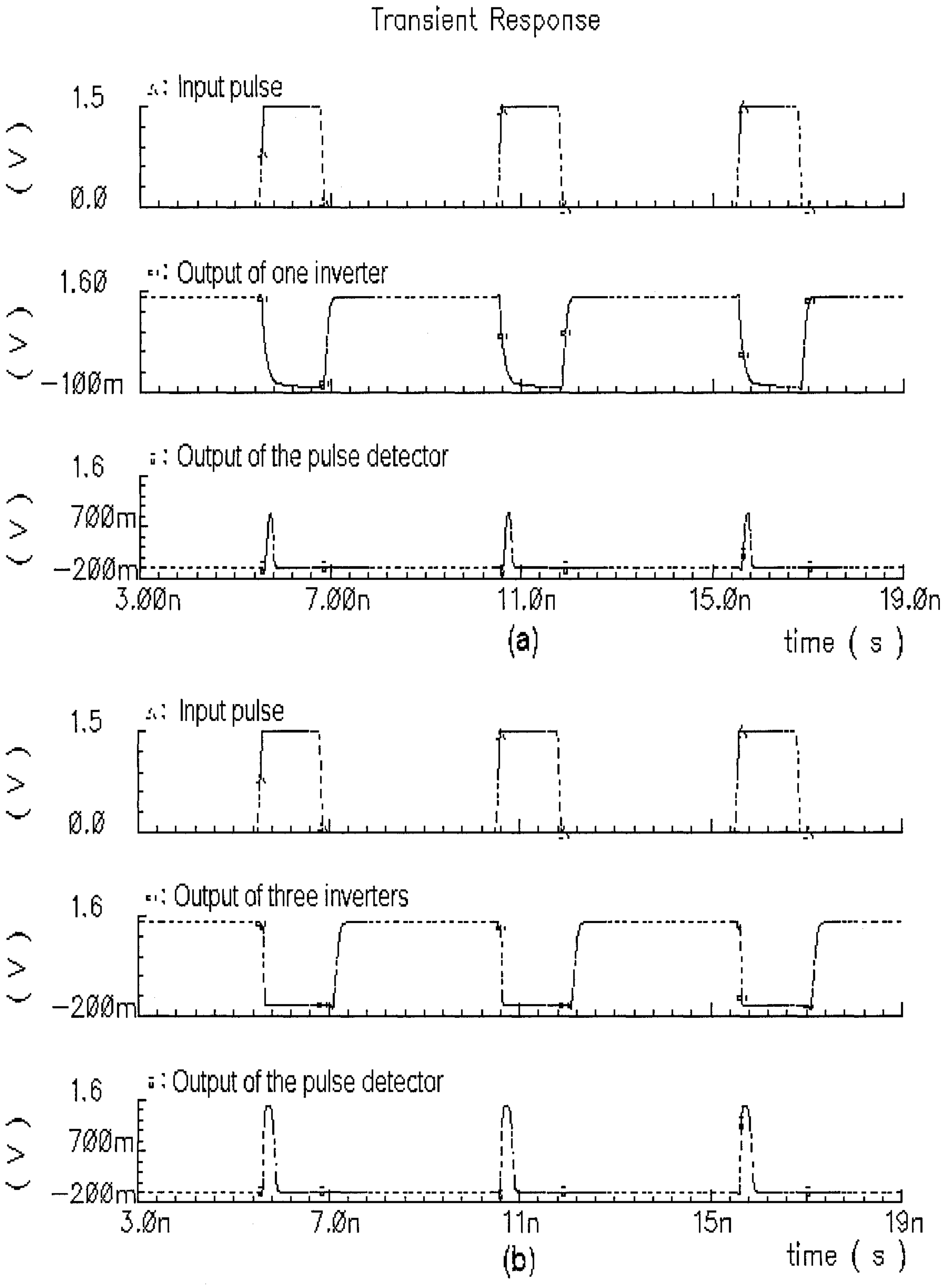


Figure 4-5 Simulation result for the Pulse Detector. (a) Output of the Pulse Detector with one inverter; (b) Output of the Pulse Detector with three inverters.

Based on the analysis above, we use the 6-bits asynchronous counter shown in Figure 4-1. Its performance is satisfactory. However, by carefully examining our circuit, we proposed a new counter structure based on our specific situation. This will save one AND gate or 6 transistors for each bit by eliminating the pulse detector section and yet ensure reliable operation.

4.2 New Counter structure

We found that the width of the input pulse (0.37 ns, it is the AP pulse from the Neuron Cell part) is wider than the trigger-time limit (0.22 ns) by only 0.15ns. This time is shorter than the propagation delay of one inverter (0.17ns- 0.2ns from simulation). So, it is preferable to maintain the trigger-time limit wider than the width of the input pulse. This will eliminate the necessity of the pulse detector in the asynchronous counter structure.

We achieved this by adding a pair of inverter in the feedback paths (from node Q to node 'a', and from node Q' to node 'd' in Figure 4-2). This makes the trigger-time limit about 0.17ns longer. The new counter structure based on this idea is shown in Figures 4-6,7.

One thing should be noted here is that since the two inverters reversely change the feedback signal, a modification is needed on the original T-latch to make the circuit work correctly. Another modification is that, unlike regular asynchronous counter (Figure 4-1), in the new counter, the input of the next bit is connected to node 'k' of previous bit through an inverter, namely U3. Both these modifications are included in the new proposed structure (Figure 4-7).

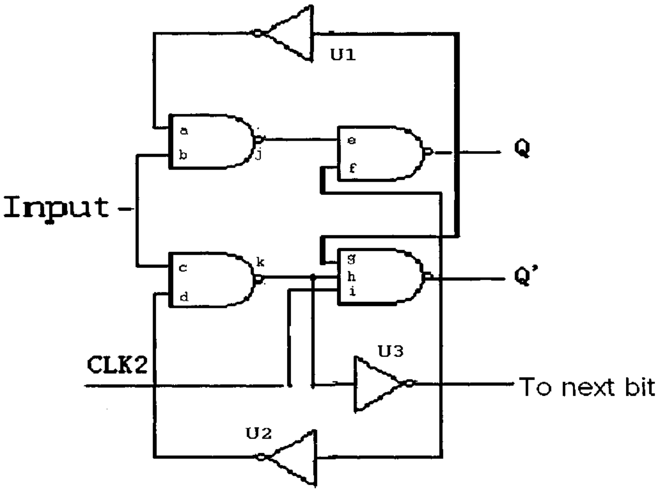


Figure 4-6 One bit structure of the new counter

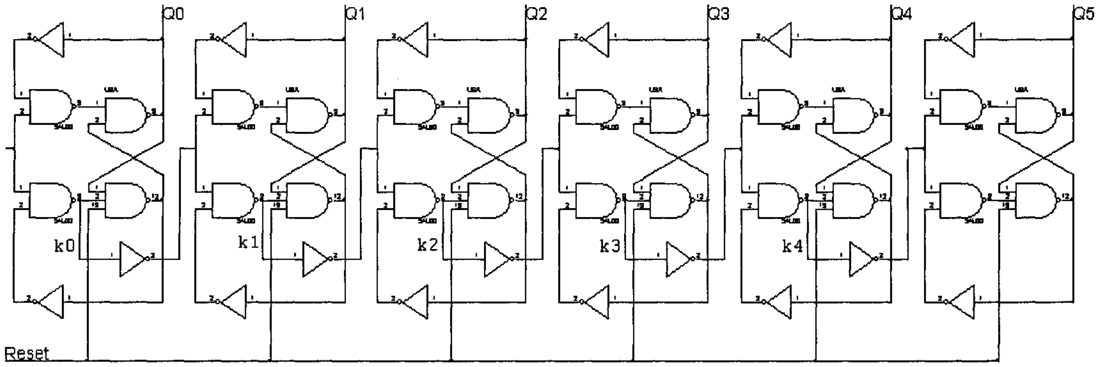


Figure 4-7 Complete structure of the new 6-bits counter

4.3 Simulation results

Figure 4-8 gives complete simulation result, from which we see the new counter works properly. Also, we see that, for each bit, two input pulses generate one pulse on node ‘k’ (see Figure 4-6 for reference). This is just what the next bit needs. Since the pulse from node ‘k’ is opposite in phase, the inverter U3 is used.

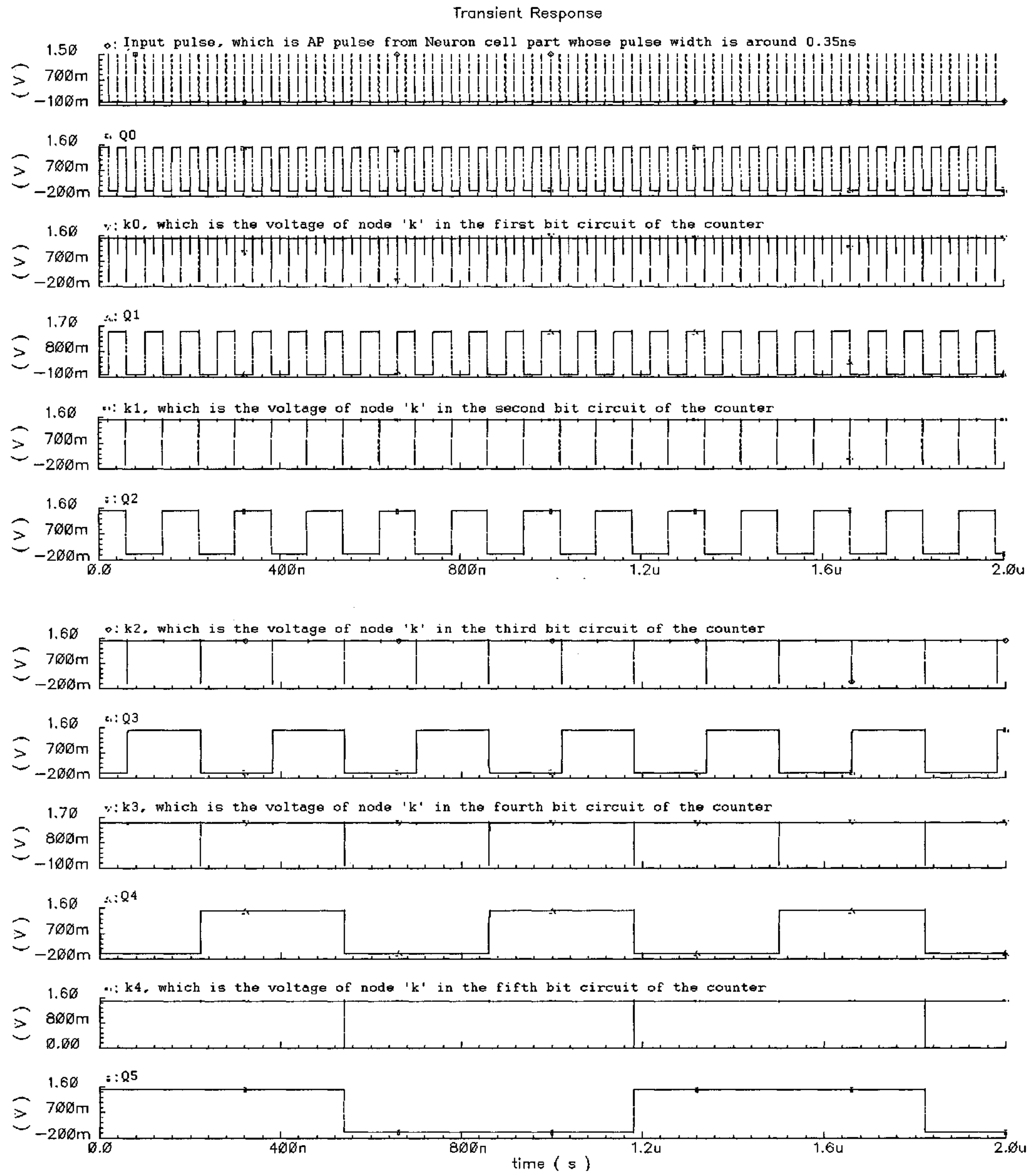


Figure 4-8 Simulation result of the Non-Pulse-Detector Counter. The input pulse is AP pulses from Neuron Cell Part

4.4 Summary

From the point of view of power consumption, asynchronous counter is a good choice for our design. The pulse detector is necessary for a common asynchronous counter. Based on our specific situation, where the input pulse comes from the Neuron

Cell, we proposed a new counter. This stretches the triggering time limit and eliminates the pulse detector. It saves thirty-six transistors for our six-bit counter and achieves satisfactory results.

Chapter 5

Performance and Improvements

In the previous chapters, we have reported on two major parts of our ADC, namely Neuron Cell part and Counter part. The performance of the overall ADC is now reported in section 5.1. From the simulation result, we see that the linearity of the ADC is not very good. So, we analyze the error in section 5.2. Then, in section 5.3, we mathematically discuss a calibration method. Based on the analysis we construct the calibration circuit and present it in section 5.4. The final simulation results are shown in section 5.5.

5.1 The Performance of the ADC

5.1.1 Simulation results

To evaluate the performance of the system, a test bench is setup (Figure 5-1).

In Figure 5-1, there are three blocks, which are the Neuron Cell part, the Counter part and the DAC part. We use the DAC to transfer the digital output of our ADC back to analog signal so that one may visually judge our ADC's performance by comparing the analog input of the ADC and the analog output of the DAC. Here, the DAC is composed of Verilog_A code. So, it is ideal and does not consume any power. Furthermore, all error appearing in this test is due to the ADC. The Verilog_A code of the DAC is attached in Appendix A-2.

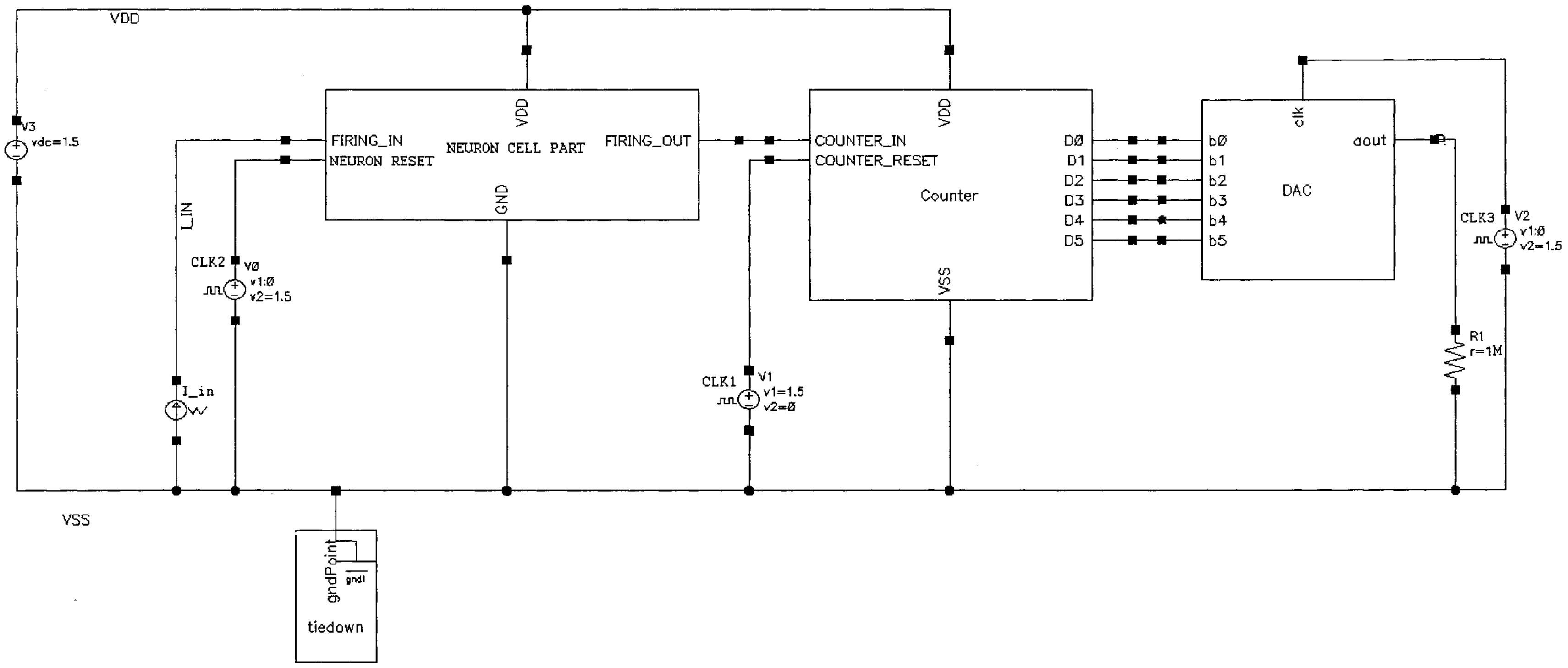


Figure 5-1 Test bench for the ADC

To evaluate the typical characteristics of the ADC, such as DNL, INL etc, theoretically, we need to do a transient analysis for a specific value of the input DC signal (current) over the time interval of the sampling period, and repeat this for other values of the input signal. This may not be practical because:

i) It would take too much time (may be more than thousands hours), and

ii) It would need too much disk space, which is far beyond the disk quota allocation.

So, we adopted the following technique for the simulation.

If the amplitude of the input current linearly increase from zero to its full scale range (FSR) over a time interval very large compared with the sampling period, it is reasonable to think that the amplitude of the input current is constant in each sampling period, namely 0.95us, and the change of amplitude between two joint sampling period is continuous. Based on this idea and our specific situation (the disk quota limitation), we use a current source whose amplitude linearly increases from zero to FSR, which is experimentally found as 9.1 μA (a little bit smaller than the theoretical analysis) in a time interval of 650us as the input of the test. This means that we totally sample about 684 ($\frac{650us}{0.95us} = 684.2$) points of the input signal. By easy calculation, we know that the change of input current in every sampling period is 0.13 μA (i.e. $\frac{9.1\mu A}{684}$), which is 0.15% of the FSR. So, it is acceptable to be treated as constant.

The simulation result is shown in Figure 5-2. The X-axis is time. Since the input current linearly increases with time, the X-axis is linearly related to the input analog current. The Y-axis is the output of DAC. Since the DAC is ideal, each step of the staircase like curve is related to a digital output word of our ADC.

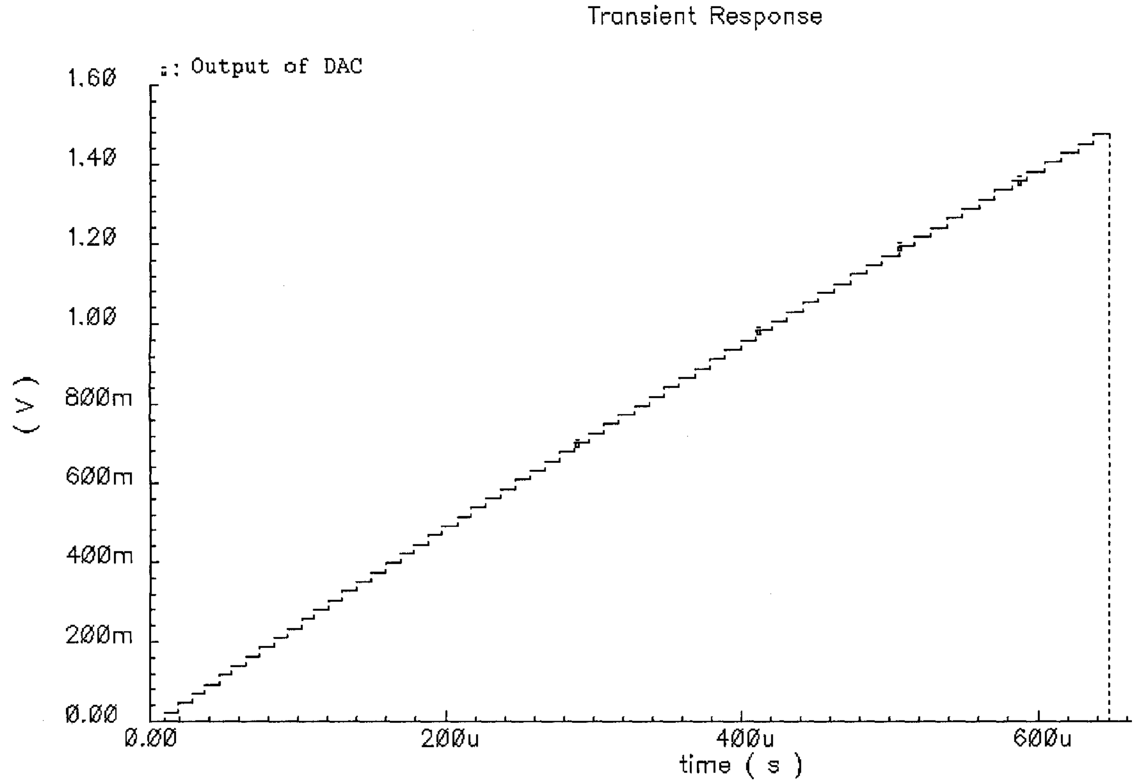


Figure 5-2 Simulation result of the non-calibration ADC, where the input is a linearly increasing current in the range from 0 to 9.1 μA

From Figure 5-2, one can easily see that there is no monotonicity error and no missing codes in our ADC. For Offset error, Gain error, INL error and DNL error etc., we need to extract the data from Cadence simulation result and put them into Matlab to do the post analysis. The Matlab code is included in Appendix A-3.

5.1.2 Post analysis by Matlab

There exist more than one definition for performance factors such as INL and DNL error; we included the definitions adopted in our design in Appendix A-4.

After post analysis, for the ADC, Matlab gives the following.

Offset error = 0.54 LSB
Gain error = -0.7154 LSB
Maximum DNL = 0.1621 LSB
Maximum INL = 1.68 LSB
Absolute accuracy = 5.4 bit
Relative accuracy = 5.3 bit

Some plots of the post analysis from Matlab are given in Figure 5-3.

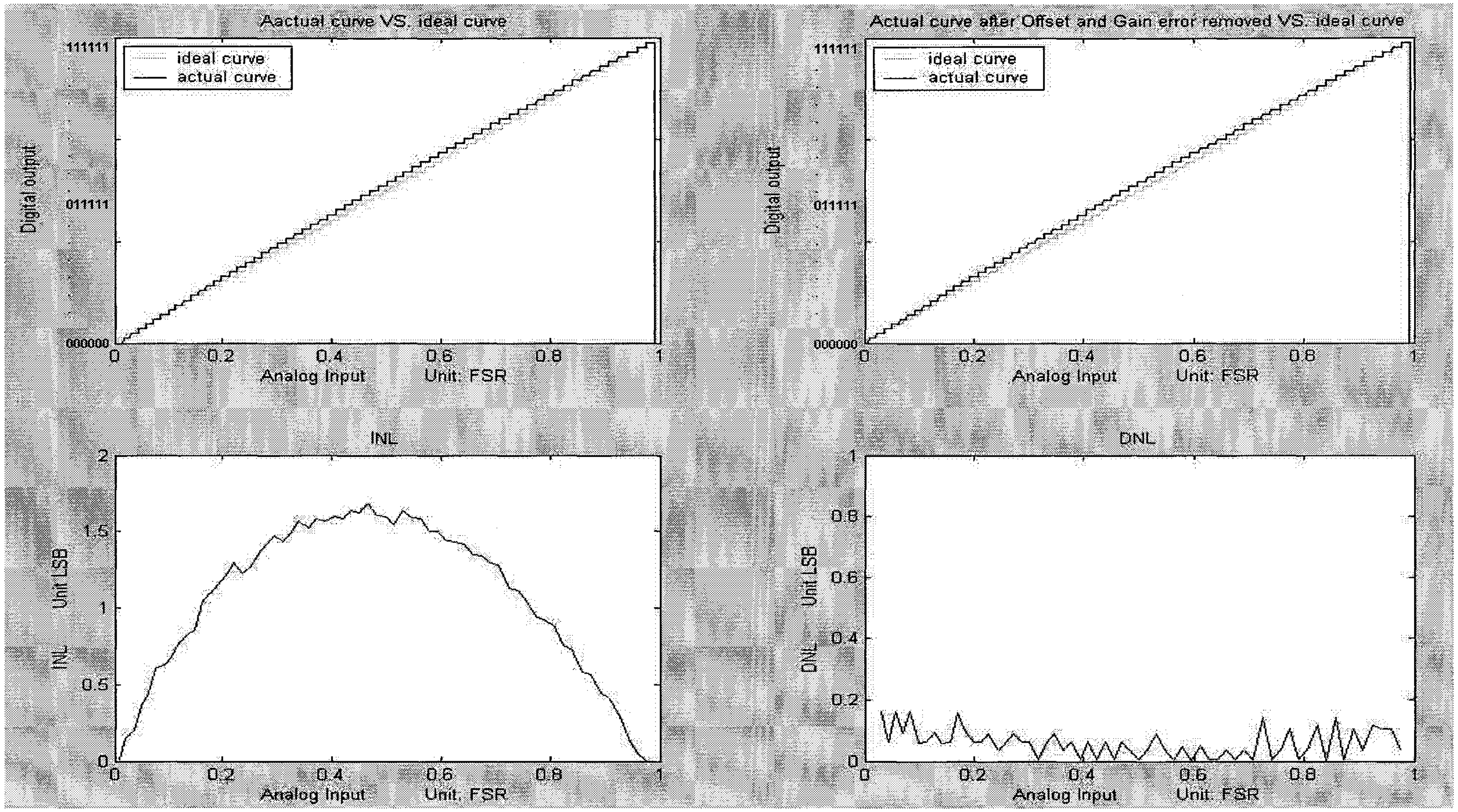


Figure 5-3 Result of post analysis by Matlab. Upper left: transfer response of our non-calibration ADC Vs. ideal 6bits ADC. Upper right: transfer response of our non-calibration ADC after Offset error and Gain error are removed Vs. ideal 6bits ADC. Lower left: INL of our our non-calibration ADC. Lower right: DNL of our non-calibration ADC.

5.2 Error analysis

From the post analysis by Matlab, we see that the most significant error is the INL error, which can be visually seen from Figure 5-3 that the transfer curve bends down (the curve's slope getting smaller) at high input current side compared with the ideal 6-bits ADC transition response. It causes the accuracy of our ADC is only 5.4 bit, which is lower than the resolution of the ADC, namely 6 bits.

In the previous development, we assumed that the width of the AP pulse is ignorable, namely $t_{pulse} = 0$. However, this is not true in practical situation. In this section, we will analyze how the nonzero t_{pulse} affect the relation between the input current and AP pulse density.

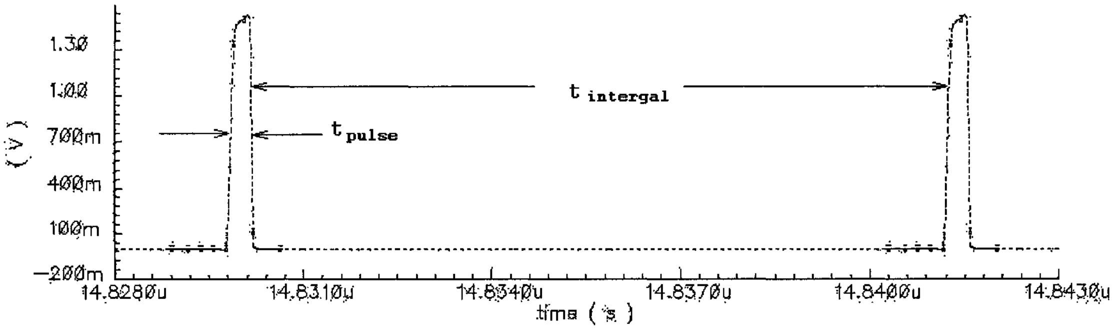


Figure 5-4 Non-ideal AP pulses, where the pulse width is labeled as t_{pulse}, and the time interval between two pulses is $t_{\text{int}egral}$.

If the width of AP pulse is ideally assumed as zero, from Eq. (3.7), we know,

$$D_{ensity\_of\_AP\_pulses} = \frac{1}{t_{\text{int}egral}} = \frac{i_{input}}{C_1 \cdot V_{threshold\_of\_U1}} \qquad (5.1)$$

Here, we rewrite Eq. (3.7) as Eq. (5.1) for convenience. After taking the nonzero pulse width in account, Eq. (5.1) should be modified as,

$$D_{ensity\_of\_AP\_pulses} = \frac{N}{T_{sampling}} = \frac{N}{N \cdot (t_{\text{int}egral} + t_{pulse})} = \frac{1}{t_{\text{int}egral} + t_{pulse}} \qquad (5.2)$$

From Eq. (3.4), we know,

$$t_{\text{int}egral} = \frac{C_1 \cdot V_{threshold\_of\_U1}}{i_{input}} \qquad (5.3)$$

Plugging Eq. (5.3) into Eq. (5.2) and after arrangement,

$$D_{ensity\_of\_AP\_pulses} = \frac{i_{input}}{t_{pulse} \cdot i_{input} + C_1 \cdot V_{threshold\_of\_U1}} \qquad (5.4)$$

From Eq. (5.4), we see that the directly proportional relation between $D_{ensity\_of\_AP\_pulses}$ and i_{input} does not hold any more. Furthermore, the denominator is $t_{pulse} \cdot i_{input} + C_1 \cdot V_{threshold\_of\_U1}$, which increases as the input current goes high. This matches our visual observation from Figure 5-2, that is the curve bend down at higher input current side.

5.3 Calibration method

To find a suitable calibration method, the first idea appearing in our mind is compensation. From previous analysis, the AP pulse density is supposed to be proportional to the amplitude of the input current, or more precisely, to the charge injected into the Neuron Cell part during each sampling period (see section 3.2). The effect of nonzero pulse width may be equivalently considered as more current (or charge) leaking as the input current is getting higher.

To find how much compensation is needed, we assume that the amount of the compensation current is i_x, and our goal is to make the AP pulse density proportional to the input current, say

$$D_{ensity\_of\_AP\_pulses} = k \cdot i_{input} \tag{5.5}$$

Where,

k is a constant and is in the unit of $A^{-1}s^{-1}$.

Based on Eq. (5.4), we get,

$$k \cdot i_{input} = \frac{i_{input} + i_x}{t_{pulse} \cdot (i_{input} + i_x) + C_1 \cdot V_{threshold\_of\_U1}} \tag{5.6}$$

By rearrangement, we get,

$$i_x = \frac{t_{pulse} \cdot k \cdot i_{input}^2 - (1 - k \cdot C_1 \cdot V_{threshold\_of\_U1}) \cdot i_{input}}{1 - t_{pulse} \cdot k \cdot i_{input}} \tag{5.7}$$

Multiply both sides by $T_{sampling}$, we get,

$$i_x \cdot T_{sampling} = T_{sampling} \cdot \frac{t_{pulse} \cdot k \cdot i_{input}^2 - (1 - k \cdot C_1 \cdot V_{threshold\_of\_U1}) \cdot i_{input}}{1 - t_{pulse} \cdot k \cdot i_{input}} \tag{5.8}$$

In Eq. (5.8), $i_x \cdot T_{sampling}$ is the compensation charge needed in each sampling period. We name it as Q_x. So,

$$Q_x = T_{sampling} \cdot \frac{t_{pulse} \cdot k \cdot i_{input}^2 - (1 - k \cdot C_1 \cdot V_{threshold\_of\_U1}) \cdot i_{input}}{1 - t_{pulse} \cdot k \cdot i_{input}} \tag{5.9}$$

Equation Eq. (5.7) gives us the exact amount of compensation current needed to keep the AP pulse density proportional to the input current, or to make the transfer response of ADC linear. So, we call it as best compensation current. Correspondingly, Eq. (5.9) gives us the best compensation charge needed in each sampling period. Since we will use a charge compensation method, we pay more attention on Eq. (5.9).

Plugging specific design parameters of our design into Eq. (5.9), we get

$$Q_x = \frac{2.11 \times 10^{-3} \times i_{input}^2 + 9.5 \times 10^{-8} \times i_{input}}{1 - 2.11 \times 10^{-3} \times i_{input}} \tag{5.10}$$

We plot the equation Eq. (5.10) and its first order fitting line in Figure 5-5. From this we get that the range of the compensation charge needed in each sampling period is from 0 to $1.05*10^{-12}$C corresponding to the input current varying from 0 to its FSR.

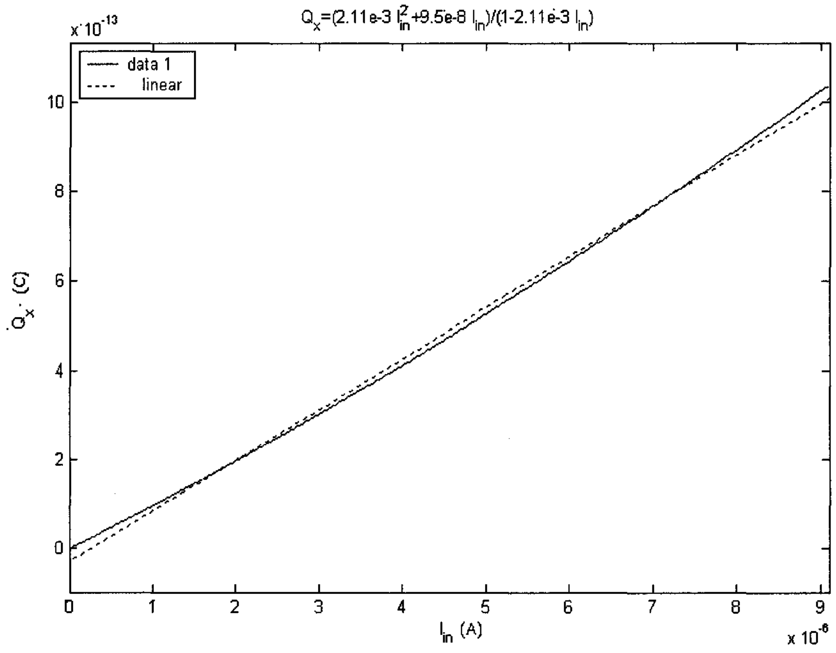


Figure 5-5 The best compensation charge and its first order fitting line.

From Eq. (5.10), we see that it is not easy to achieve the best compensation, in other words it is not easy to make a compensation circuit whose transfer function exactly match Eq. (5.10). However, from Figure 5-5, we see that the curve of Q_x is quite close to a straight line as the input current varies within its full-scale range. This gives us a clue that a linear compensation might be possible.

5.4 Design of the linear calibration circuit

Based on above analysis, we propose a simple compensation circuit in Figure 5-6, in which the compensation charge is linearly related to the input current.

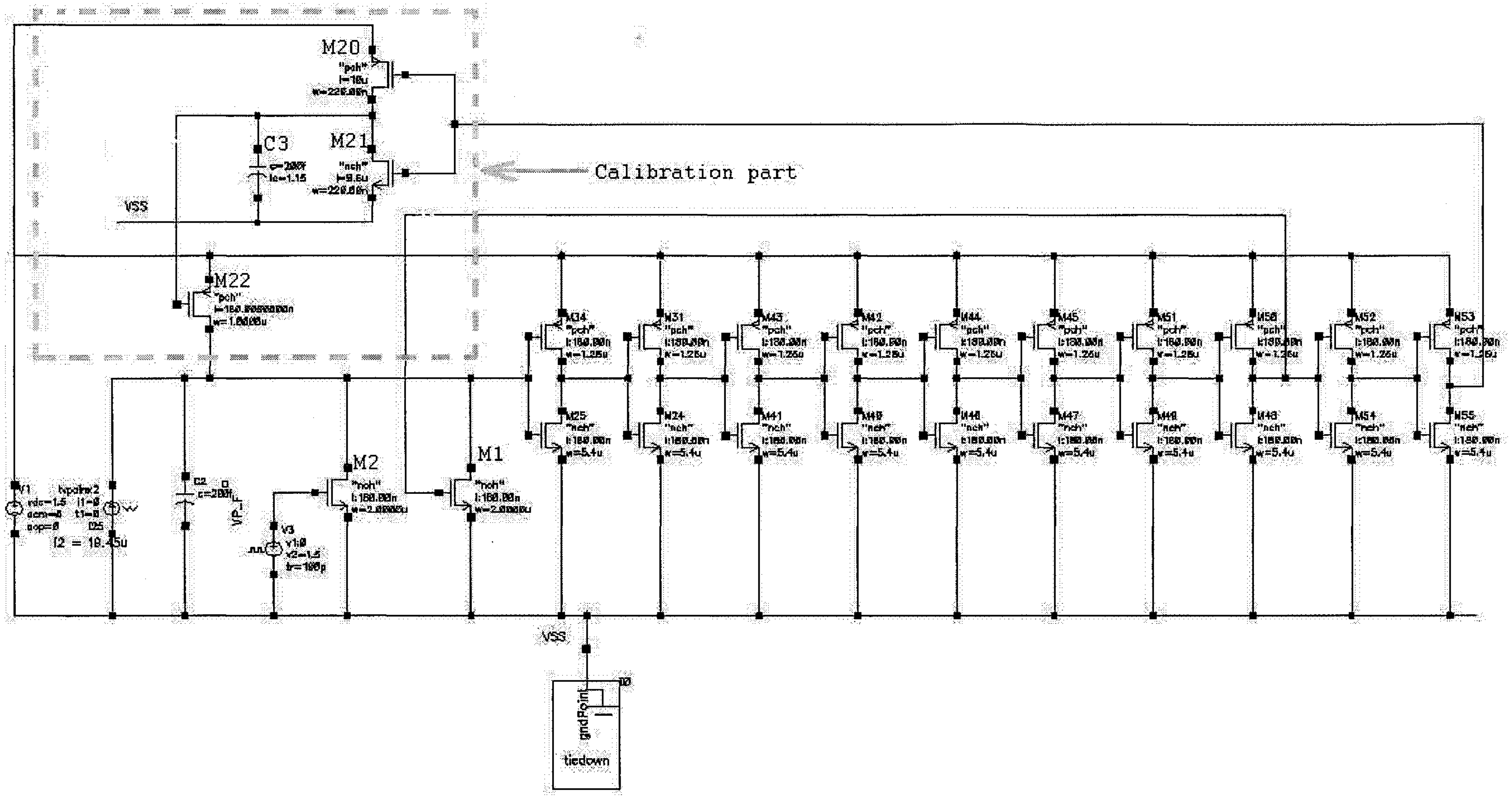


Figure 5-6 Schematic of our ADC with calibration

In Figure 5-6, the compensation circuit is composed by a charge-pump and a PMOS transistor, namely M_{22}. The input of the charge-pump is the AP pulses. As the Neuron Cell part is in its rest state, namely no AP pulse is generated, the output of the charge-pump is the positive rail voltage, which cuts off M_{22}, and no compensation charge is injected into the membrane capacitor. As an AP pulse arrives, the gate of both M_{20} and M_{21} go high, which cuts off M_{20} and turn on M_{21}. So, the capacitor C_3 is discharged and the output of the charge pump drops down. After an interval of t_{pulse}, the gate of both M_{20} and M_{21} drop down. Opposite to the previous process, M_{20} is turned on and M_{21} is cutoff; C_3 is charged and the output of the charge pump goes up to positive rail voltage quickly. Figure 5-7 illustrates this scenario.

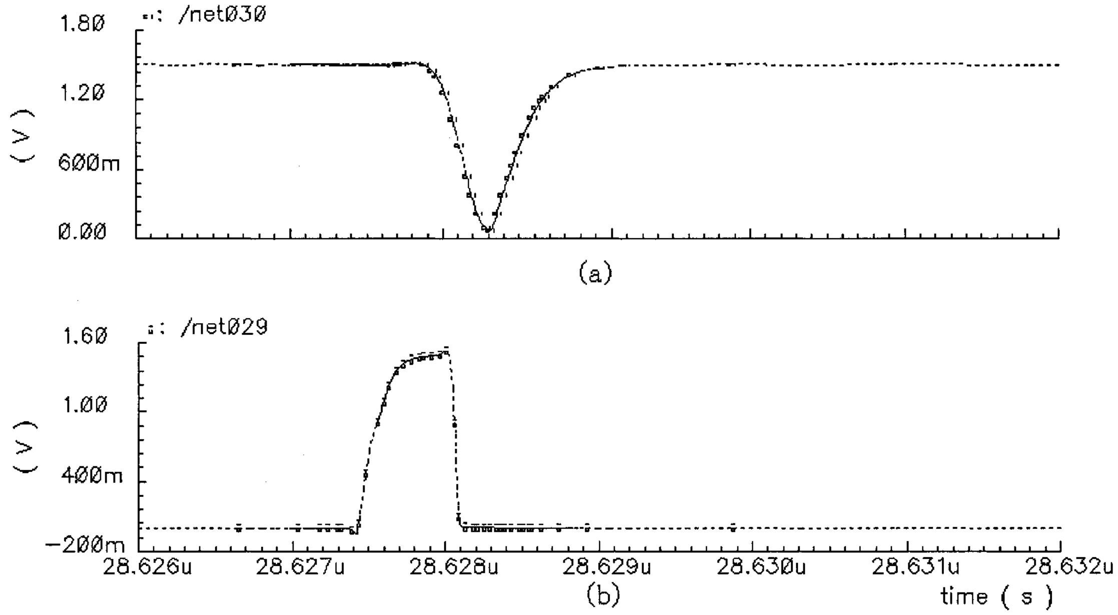


Figure 5-7 (a) the output of the charge pump, which is used to control M_{22}; (b) one AP pulse, which is fed back to control M_1.

From Figure 5-7, we see that corresponding to each AP pulse, there is a negative pulse generated at the output of the charge-pump. Its density is the same as that of the AP pulse. This negative pulse in turn generates a current pulse through M_{22}, in other words, it controls M_{22} to inject a little charge into the membrane capacitor. Furthermore, since the shape of every AP pulse is almost the same, the amount of charges injected corresponding to each pulse is also constant. So, we get,

$$Q_x = N \cdot Q_0 = D_{ensity\_of\_AP\_pulse} \cdot T_{sampling} \cdot Q_0 \qquad (5.11)$$

Where,

Q_x is the compensation charges injected into the membrane capacitor in each sampling period.

N is the number of AP pulses in each sampling period.

Q_0 is the amount of compensation charges injected into the membrane capacitor during one t_{pulse}.

From previous analysis, we know,

$$D_{ensity\_of\_AP\_pulse} \propto i_{input} \qquad (5.12)$$

By combining Eq. (5.11) and Eq. (5.12), we get a linear relation,

$$Q_x \propto i_{input} \cdot Q_0 \qquad (5.13)$$

Now, we see from Eq. (5.13) that the compensation charges are proportional to the input current, which just meets our goal, and the slope is controlled by Q_0.

The careful reader may find that Eq. (5.12) does not hold strictly since the density of the AP pulse is not perfectly proportional to the input current, instead, from Eq. (5.4), it is,

$$D_{ensity\_of\_AP\_pulses} = \frac{i_{input}}{t_{pulse} \cdot i_{input} + C_1 \cdot V_{threshold\_of\_U1}}$$

Plugging Eq. (5.4) into Eq. (5.11), we get

$$Q_x = D_{ensity\_of\_AP\_pulse} \cdot T_{sampling} \cdot Q_0 = \frac{i_{input}}{t_{pulse} \cdot i_{input} + C_1 \cdot V_{threshold\_of\_U1}} \cdot T_{sampling} \cdot Q_0 \qquad (5.14)$$

Plugging specific design parameters of our design into Eq. (5.14), we get

$$Q_x = \frac{i_{input}}{3.5\times10^{-10} \times i_{input} + 1.5\times10^{-13}} \times 9.5\times10^{-7} \times Q_0 \qquad (5.15)$$

We plot Eq. (5.15) and Eq. (5.10) in Figure 5-8, here we set $Q_0 = 1.772\times10^{-14}$ C

From Figure 5-8, we see that the actual compensation curve from Eq. (5.15) is quite straight and very close to the best compensation curve from Eq. (5.10). So, Eq. (5.12) may still be considered as valid in our analysis, namely the actual compensation charge is proportional to the input current.

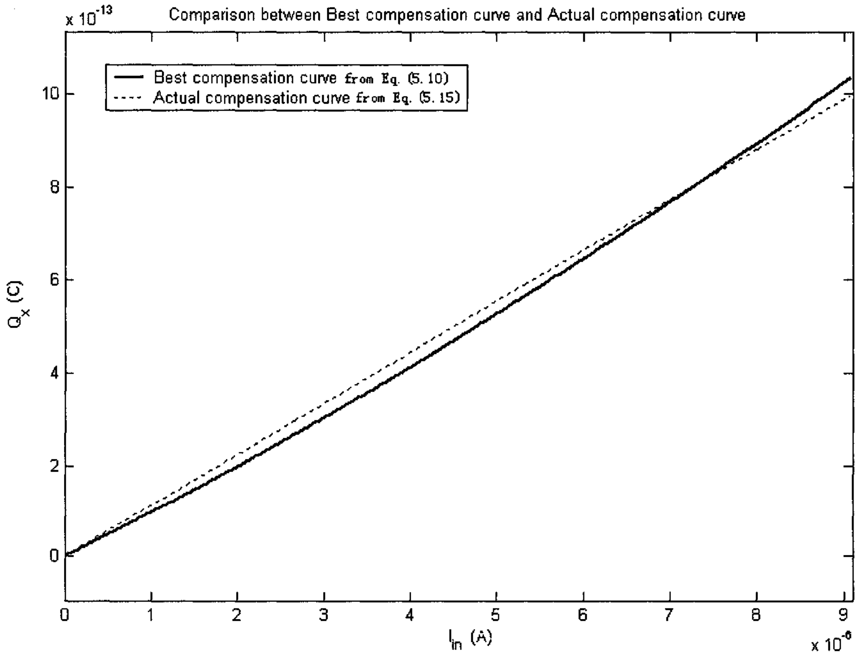


Figure 5-8 The comparison of the best compensation and the actual compensation achieved

By modifying the size of M_{20}, M_{21} and C_3, we may adjust the shape of the output pulse from the charge-pump. Furthermore, by adjusting the size of M_{22}, we may control the amplitude of the compensation current for each pulse. By either of these methods, we can control the value of Q_0, or the slope of the actual compensation curve to achieve desired compensation effect.

Besides the analysis above, three more points about the calibration circuit are mentioned below,

i) In this design, we do not have to precisely care in which region (Saturation, Triode, or Subthreshold etc.) the transistor M22 is working, since as long as the amount of injected compensation charges keep constant for each AP pulse, the linear relation between Q_x and i_{input} will hold (see Eq. (5.13)). Thus, the compensating circuit will work well. This makes the design much easy, and this is one merit brought from the area sampling method. In other words, in this method, the linear relation is between the integral of current (or the charge) and pulse density. So, in the calibration circuit, we compensate charges instead of current, and, we do not have to depend upon the nonlinear relation between V_{gs} and I_D of the MOS transistor. We may get satisfactory result by simply adjusting the sizes of M_{20}, M_{21}, and M_{22} (but mostly M_{22}).

ii) Having compared Figure 5-6 with Figure 3-6, one finds that one extra pair of inverter is added in the inverter series of the Neuron Cell part. This pair of inverters is important for the calibration circuit. From previous section, we know that each AP pulse is fed back to turn on the switch M_1 (see Figure 5-6) so that it discharges the membrane capacitor and bypass the input current. We also know that each AP pulse generates a compensation current pulse through M_{22}. So, if the M_1 and M_{22} are turned on at the same time, all the compensation current will be bypassed through M_1, and the calibration circuit will be useless. So, we add one pair of inverter to introduce some delay so that the output pulse of the charge-pump turns on M_{22} after M_1 cuts off. Figure 5-7 clearly shows this point.

iii) Having read the description of how the calibration circuit works, people may think that the capacitor, namely C_3 in the charge-pump may be eliminated. However, the answer is no. If we get rid of C_3, the charge-pump will become just an inverter. So, the low value of the negative pulse of the output of the charge-pump will be zero, and its width will be close to that of an AP pulse, namely 0.37ns in our design. From Eq. (5.17), we know that the amount of the compensation charges is quite small. This means that the transistor M_{22} that is controlled by such a negative pulse must have very small W/L value. That is, the value of L for M22 has to be very large. Since the parasitic capacitance is proportional to the product of W and L, the large L will generate a large parasitic capacitance connecting to the membrane capacitor, namely C_1. Furthermore, because the value of the parasitic capacitance is difficult to control, large L is undesired. This will dramatically reduce the precision of the ADC. In conclusion, C_1 controls the shape of negative pulse of the output of the charge-pump so that both L and W of M_{22} may be kept small.

5.6 Simulation result

A simulation result and the plots of the post analysis of the ADC with calibration are shown in Figure 5-9 and Fig 5-10 respectively.

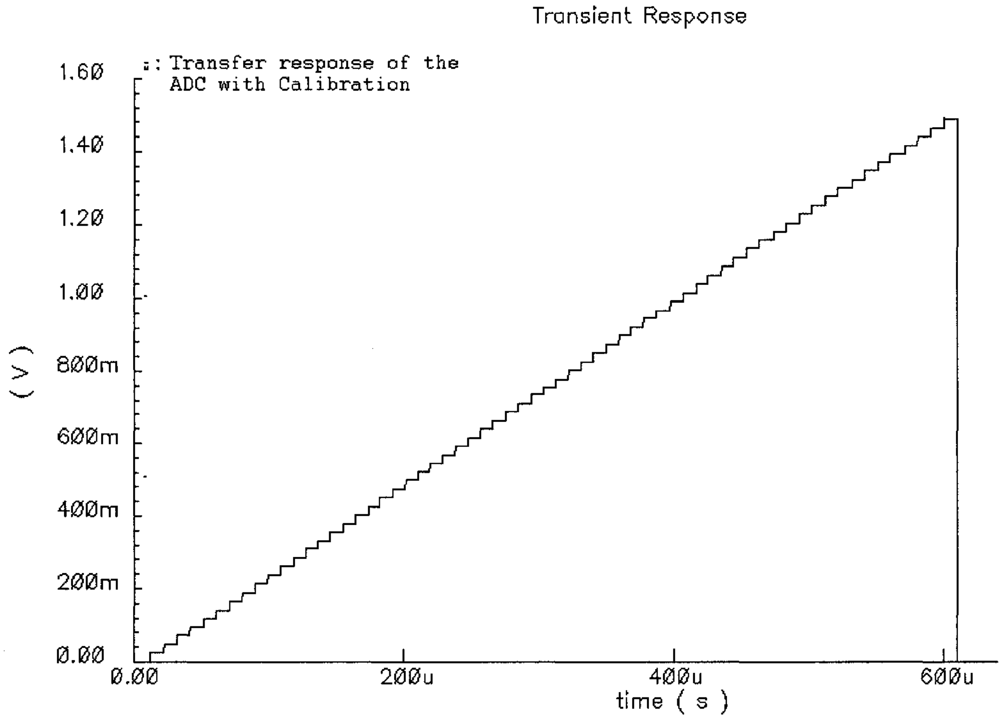


Figure 5-9 Simulation result of the ADC with calibration, where the input is a linear increasing current in the range from 0 to 8 μA

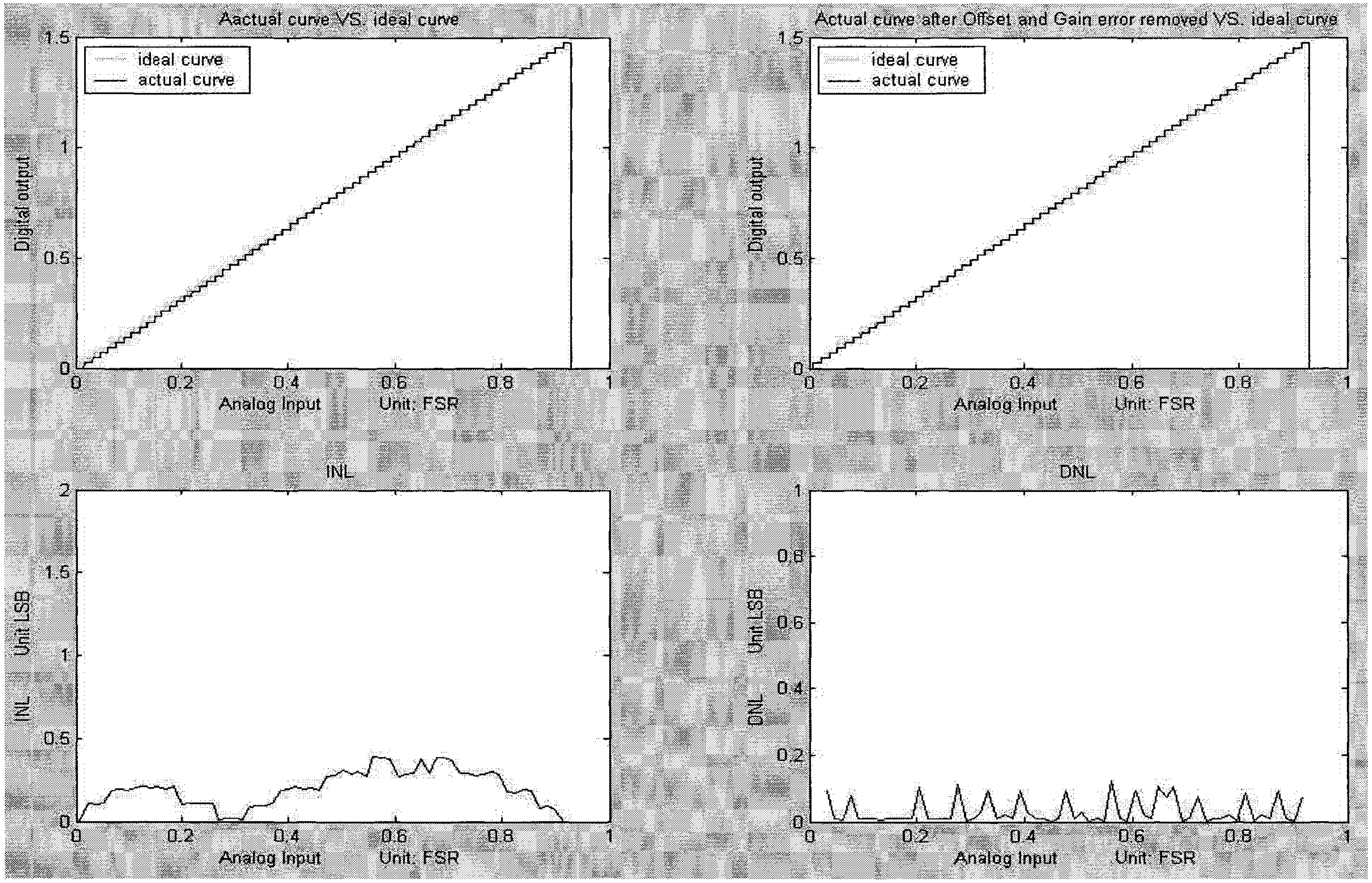


Figure 5-10 Result of post analysis by Matlab. Upper left: transfer response of our ADC with calibration Vs. ideal 6bits ADC. Upper right: transfer response of our ADC with calibration after Offset error and Gain error are removed Vs. ideal 6bits ADC. Lower left: INL of our ADC with calibration. Lower right: DNL of our ADC with calibration.

A comparison of the performance factors between the ADC with and without calibration is shown in table 5-1. From this table, we see that the relative accuracy of our ADC is improved by more than 2 bits. This means that the linearity is greatly improved. Another thing should be mentioned here is that the FSR is modified from 9.1 μA to 8 μA for the ADC with calibration, this is reasonable since instead that all current being injected from the input source, some current gets injected from the compensation circuit, and this change does not reduce the performance of our ADC.

Table 5-1 Comparison of the performance of ADC with and without Calibration

	ADC without Calibration	ADC with Calibration
Temperature	27°C	27°C
Offset error	0.54 LSB	0. 59 LSB
Gain error	-0.7154 LSB	-0.59 LSB
Maximum DNL	0.1621 LSB	0.12 LSB
Maximum INL	1.68 LSB	0.39 LSB
FSR	9.1 μA	8 μA
Absolute accuracy	**5.4 bit**	**6.44 bit**
Relative accuracy	**5.3 bit**	**7.35 bit**

5.7 Summary

In this chapter, we first examined the performance of our ADC without calibration. The simulation result shows a large INL error. From analysis, we see that it is principally caused by the non-zero pulse width. Based on this, we proposed a simple calibration method, namely charge compensation. Finally we achieve it by circuit. The simulation shows that the final ADC provides improved performance.

Charter 6

Layout implementation and results

The whole design of our ADC is based on TSMC's COMS 0.18um technology. The layout design is briefly introduced in section 6.1. In section 6.2, a post layout simulation result with a sinusoidal input is shown. Next, we present the power consumption of our ADC in section 6.3, which is followed by the robustness analysis in section 6.4. We would like to mention that all the simulation results presented in the previous chapters (Chapter 3, 4, 5) are for the system after extracting electrical equivalent of the physical layout, i.e., the post layout simulation.

6.1 Layout Design

The layout of the core section of our ADC occupies an area of approximately 0.0189 mm^2, and is shown in Figure 6-1. In this figure, the top part is the Neuron Cell, and the bottom part is the six-bit Counter. The layout design is simple, and all components are selected from standard TSMC's COMS 0.18um library. Since this ADC works with low frequency and very small amplitude signal, there is no special requirement (such as rigorous symmetry requirement) needed in the layout design.

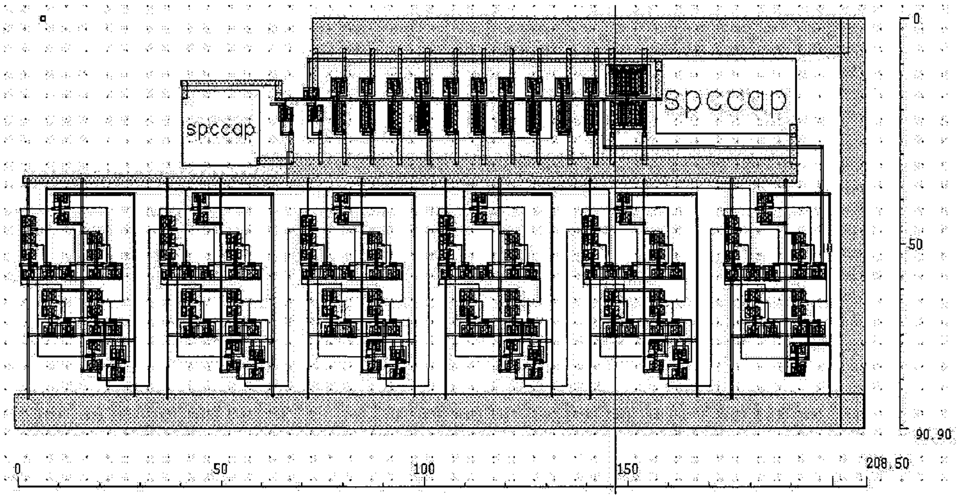


Figure 6-1 Layout of the core circuit of our ADC.

6.2 Simulation Results with a Sinusoid Input Signal

In this experiment, a sinusoid current is applied at the input of the test bench shown in Figure 5-1. Its frequency is 20kHz and peak-to-peak amplitude is 7.95 μA.The input signal, the output of the Neuron Cell Part, the analog output of the DAC and the output of the Counter part (D0 to D5) are shown in Figure 6-2.

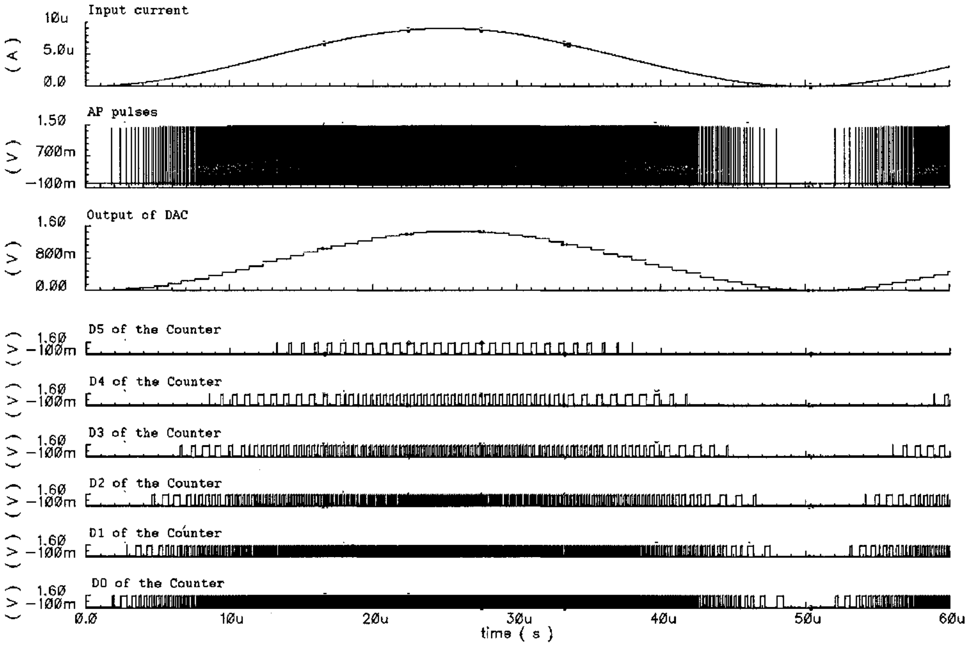


Figure 6-2 The simulation result with a sinusoid input signal

From Figure 6-2, we see that the density of the AP pulses varies corresponding to the amplitude of the input current; also the output of the DAC visually matches the input signal well.

6.3 Power consumption

In the simulation above, the average power consumed by our ADC is less than 76uW when a sinusoid signal of 20 kHz is applied. A widely accepted metric that is used to judge the power consumption characteristic between different types of ADC is 'Energy per Sample'. It defined as the average power of an ADC times the sampling period. Lower this value is, better is the power consumption performance. This is analogous to 'delay power product' for a digital gate, or 'power per bandwidth' in an analog system. The following table is a comparison between our ADC and some recently reported low power ADC, from which we see that our ADC has superior performance regarding 'Energy per Sample'.

Table 6-1 Comparison between our ADC and other low power ADC

Name	Type	Technology	Power	Power Supply Voltage	Energy per Sample	Reference
Our design	**Neuron based counter**	**.18um**	**75.9uW**	**1.5V**	**72.3pJ**	
Switched capacitor multi-bit ADC	Delta-sigma	.35um	12mW	2.7V	368pJ	[38]
A Low-Power 6-b Integrating- Pipeline Hybrid Analog-to-Digital Converter	Pipeline Hybrid	.25um	14.7mW	2.5V	294pJ	[39]
An 8-bit folding/interpolating ADC		.15um	35mW	1.5V	344pJ	[40]
Very low-power, high-speed flash A/D converter	Flash	.35um	8mW	1V	78pJ	[41]
An Ultra-Low Power ADC	Successive Approximation	.25μm	3.1uW	1V	31pJ	[42]
6-bit 250 MS/s analog-to-digital converter (ADC)	Successive Approximation	.35um	30 mW	3.3 V	120pJ	[43]

In this comparison, it may appear that the ADC reported in [42] has better 'Energy per Sample' performance than ours'. However, on careful study we found that this ADC is successive approximation type and in reporting the power consumption the authors have not taken into account the power consumed by the logic control circuitry sub-system of their ADC.

6.4 Robustness of our system

We examined the robustness of our ADC system with temperature variations and power supply variations. The item of concern was the linearity performance.

First, we tested our ADC over a range of one hundred degrees between –20°C and 80°C. The simulation results have been extracted and analyzed by Matlab. The outcome is shown in Table 6-2. From this we see that, under these two extreme temperature conditions (i) the degradation of Maximum DNL is less than 0.02 LSB; (ii) the degradation of Maximum INL is less than 0.04 LSB; (iii) the degradation of absolute accuracy is less than 0.05 bit, and (iv) the degradation of relative accuracy is less than 0.04 bit. So, our ADC appears very robust in performance as temperature varies over a wide range. Figures 6-3,4 incorporated from the Matlab analysis make these claims evident.

Table 6-2 Comparison of our ADC under different test condition.

	ADC without Calibration	ADC with Calibration	ADC with Calibration	ADC with Calibration
Temperature	27°C	27°C	-20°C	80°C
Offset error	0.54 LSB	0. 59 LSB	0.58 LSB	0.63 LSB
Gain error	-0.7154 LSB	-0.59 LSB	-0.49 LSB	-0.80 LSB
Maximum DNL	0.1621 LSB	0.12 LSB	0.12 LSB	0.10 LSB
Maximum INL	1.68 LSB	0.39 LSB	0.40 LSB	0.35 LSB
Absolute accuracy	**5.4 bit**	**6.44 bit**	**6.39 bit**	**6.5 bit**
Relative accuracy	**5.3 bit**	**7.35 bit**	**7.31 bit**	**7.5 bit**

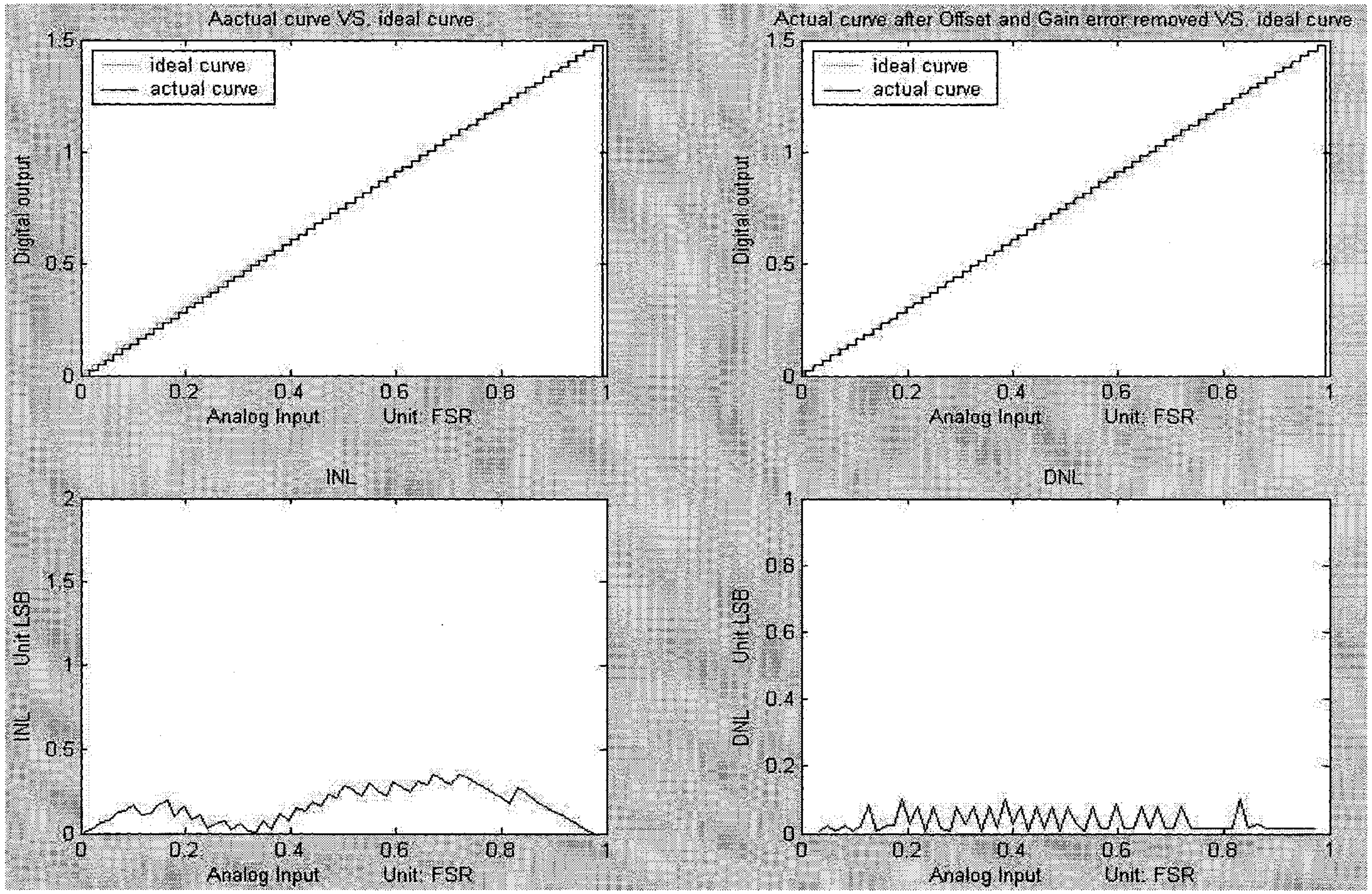


Figure 6-3 Matlab post analysis result of our ADC with Calibration under 80°C. Upper left: transfer response of our ADC Vs. ideal 6bits ADC. Upper right: transfer response of our ADC after Offset error and Gain error are removed Vs. ideal 6bits ADC. Lower left: INL of our ADC. Lower right: DNL of our ADC.

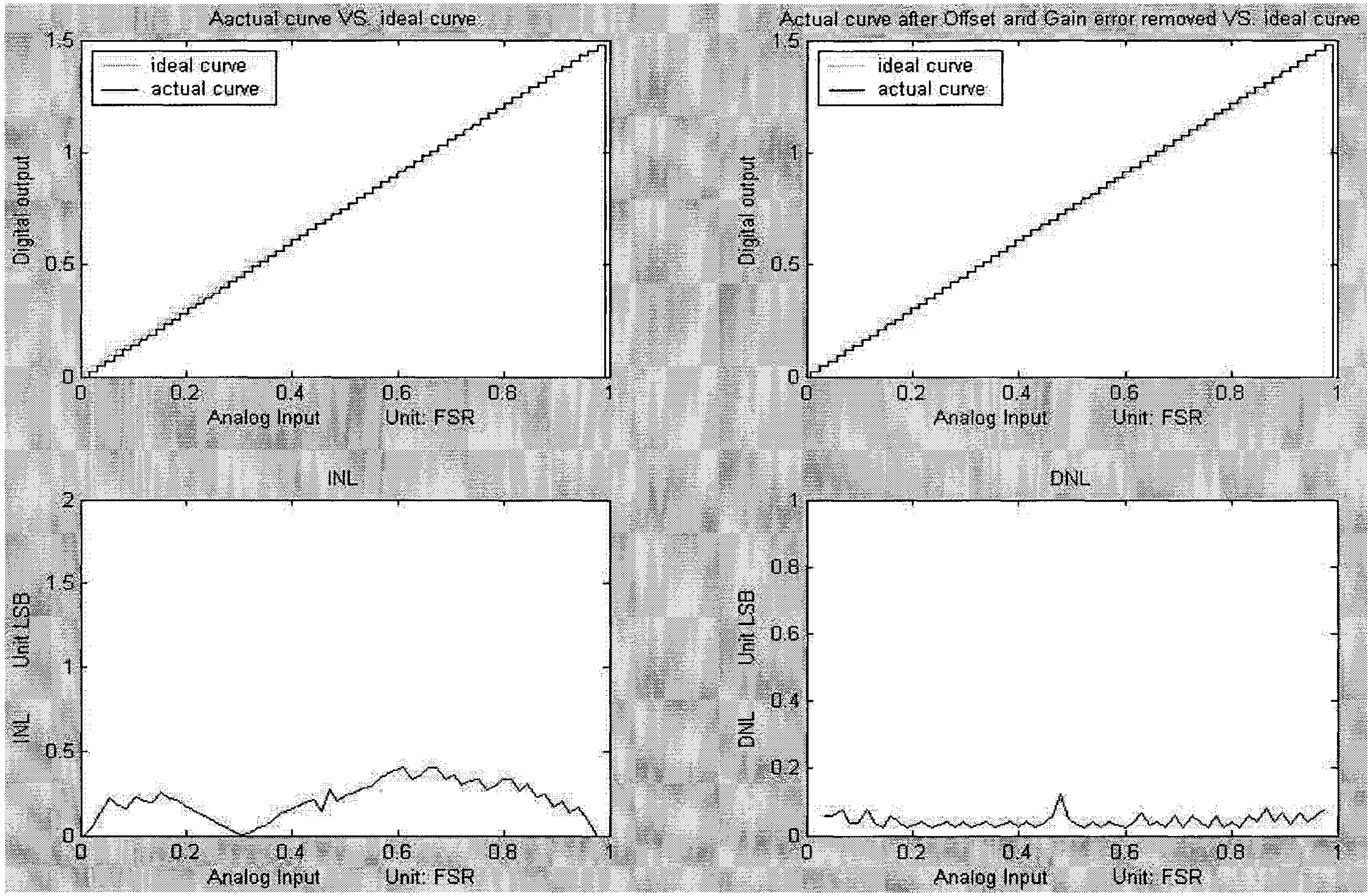


Figure 6-4 Matlab post analysis result of our ADC with Calibration under -20°C. Upper left: transfer response of our ADC Vs. ideal 6bits ADC. Upper right: transfer response of our ADC after Offset error and Gain error are removed Vs. ideal 6bits ADC. Lower left: INL of our ADC. Lower right: DNL of our ADC.

However, regarding robustness with respect to power supply changes within $\pm 20\%$, our ADC does not exhibits satisfactory performance. From Eq. (3.12), we see that our ADC is sensitive to $V_{threshold\_of\_U1}$, which is about half of the power supply. So, the robustness against the variation of power supply of our ADC is not good. That means a stabilized voltage supply may be required for faithful operation of the ADC system in an environment with large temperature variations.

6.5 Summary

The layout design of our ADC is simple and easily achieved. The post-layout simulation results show that it works well with a sinusoid input of the maximum nominal input frequency, namely 20 kHz. Moreover, our ADC has superior power consumption performance and good robustness against temperature variation.

Chapter 7
Conclusion and Future Work

7.1 Conclusion

A brief background of physiological Neuron system and a neuron-based unipolar 6-bit ADC design have been presented in this thesis. The ADC is designed to work with a voltage supply of 1.5V and has the ability to operate with excellent linearity (INL is not greater than 0.4 LSB) for signals in the low frequency range (i.e., 20 kHz). Schematic and layout designs have been done. From the analysis and the simulation results, this design exhibits some special features. Thus,

- It uses 'Area sampling' method, and there is no S/H circuit needed in this design.
- It has a compact structure. It may be roughly categorized into 'Integral type ADC'. Unlike other 'Integral type ADC', no OPAM is needed in this design. Instead, it uses a small on-chip capacitor to accomplish the 'Integral' function, and by a simple calibration circuit, satisfactory performance is achieved.
- The basic idea of this design is the so-called 'Pulse plus Counter' method. By properly adjusting the width of the pulse, a simpler counter structure is required. This saves thirty six transistors compared with those in a six bit regular asynchronous counter.
- Because it uses 'Charge Sampling' method, a very simple charge compensation calibration circuit (total four components used, see Chapter 5 for reference) could

achieve satisfactory performance.

- Because the system uses fewer devices and components, leading to low power consumption, the converter becomes very competitive with other known converters in terms of ‘energy per sample’.
- System on chip (SOC) design usually requires that the building block take as small area as possible. The layout of the core circuitry of our ADC measures approximately 0.0189 mm^2. It is compact and suitable for (SOC) design.

Generally speaking, this ADC is very favorable for low power consumption, compact circuit structure and low frequency response. It will be attractive in low power low-resolution demands, such as massive, dense, intelligent, adaptive sensor networks and some portable electronic device.

7.2 Future work

The ADC using electronic neurons has six-bit resolution and its maximum working frequency is up to 20kHz. To find wider applications, increasing the resolution and making it work at higher frequencies without costing much power consumption would be the two major improvements required. Furthermore, improvement about the robustness against power supply voltage variation is needed. To achieve the goal of high speed of operation, we may employ two or more electronic neurons connected in a parallel structure, and sum the outputs of all the neurons in certain way. If all of these electronic neurons work at the same time and they collaborate properly, then the operating speed might be increased. Furthermore, a weighted sum and multilayer combination of several neurons could be used to combat effectively the sensitivity of performance of a single neuron to power supply variations.

7.3 Contributions from this thesis

R Raut and Ke Wei Zheng, “An efficient A/D Converter using Electronic Neurons”, The 9th World Multi-Conference on Systemics, Cybernetics and Informatics, Volume VI, 2005, pp.70-74.

Appendix

A-1: Matlab Code for Hodgkin-Huxley model

```
% Matlab Code for Hodgkin-Huxley model
% Code by Zheng Kewei
% Oct 6th,2005

clear all;
% Maximum condunctances
Gna=120;           %mmho/cm2
Gk=36;             %mmho/cm2
Gl=0.3;            %mmho/cm2

% Numerical Values
Vk=-72;            %mV/cm2
Vna=55;            %mV/cm2
Vl=-49.387;          %mV/cm2

% Arrais to store each varibles
double T=[];
double V=[];
double M=[];
double N=[];
double H=[];
double K=[];
double Na=[];

% Initial Conditions
T(1)=0;
V(1)=-60;               %mV/cm2
M(1)=0.05;
N(1)=0.3;
H(1)=0.6;
K(1)=Gk*N(1)^4;        %mmho/cm2
Na(1)=Gna*M(1)^3*H(1);   %mmho/cm2
```

```
dt=0.01;            %time incremental unit:mS
tmax=10;            %ending time unit:mS
u=1;                %counter
i=5;

for t=dt:dt:tmax
    t
    u=u+1;

dn=(0.01*(-50-V(u-1))/(exp((-50-V(u-1))/10.)-1.0)*(1-N(u-1))-0.125*exp((-60-V(u-1))/80.)*N(u-1))*dt;

dm=((0.1*(-35-V(u-1)))/(exp((-35-V(u-1))/10.)-1.0)*(1-M(u-1))-4.0*exp((-60-V(u-1))/18.)*M(u-1))*dt;
    dh=(0.07*exp((-60-V(u-1))/20.)*(1-H(u-1))-H(u-1)/(exp((-30-V(u-1))/10.)+1.))*dt;
    dv=((-1)*Gna*M(u-1)^3*H(u-1)*(V(u-1)-Vna)-Gk*N(u-1)^4*(V(u-1)-Vk)-Gl*(V(u-1)-Vl)+i)*dt;

    T(u)=T(u-1)+dt;
    M(u)=M(u-1)+dm;
    N(u)=N(u-1)+dn;
    H(u)=H(u-1)+dh;
    V(u)=V(u-1)+dv;
    Na(u)=Gna*M(u)^3*H(u);
    K(u)=Gk*N(u)^4;
end

plot (T,M,T,N,':',T,H,'-.');
xlabel 'time    (ms)';
title 'Conductance parameters';
legend ('m','n','h');
pause;

plot(T,V);
xlabel 'time    (ms)';
ylabel 'mV';
title 'Membrane Potential';
pause;
```

```
plot (T,K,T,Na,':');
xlabel 'time    (ms)';
ylabel 'mmho/cm^2';
title 'Na and K conductances';
legend ('K','Na');
```

A-2: Verilog_A code of the IDEAL 6 BITS DAC

```
// IDEAL 6 BITS DAC FOR THE TESTING PURPOSE OF NEURO BASED ADC
// CREATED BY KEWEI ZHENG
// MAY 30 2005


`include "constants.h"
`include "discipline.h"

module dac (clk, b5, b4, b3, b2, b1, b0, aout);
electrical clk, b5, b4, b3, b2, b1, b0, aout;

parameter real vmax=1.5;

real result;

analog begin
    @(cross (V(clk)-0.75 , +1)) begin
        result = 0;
        result = result + (( V(b5) > 0.75) ? (vmax/2) : 0 );
        result = result + (( V(b4) > 0.75) ? (vmax/4) : 0 );
        result = result + (( V(b3) > 0.75) ? (vmax/8) : 0 );
        result = result + (( V(b2) > 0.75) ? (vmax/16) : 0 );
        result = result + (( V(b1) > 0.75) ? (vmax/32) : 0 );
        result = result + (( V(b0) > 0.75) ? (vmax/64) : 0 );
    end
    V(aout) <+ result;
end
endmodule
```

A-3: Matlab code for the post analysis of ADC

```
%Matlab code for the post analysis of ADC
%Coded by Zheng Kewei
%09-25-2005

clear all
load d:\thesis\extracted\Final_extracted_data.txt;
m= Final_extracted_data;
x=m(:,1);% time
y=m(:,2);%output voltage of DAC
plot(x,y)
%remove the data in first Sampling period
for c=1:length(x)
    if x(c)<=0.95e-6
        y(c)=[];
        y(length(x))=0;
    end
end

% The following For loop find the last transition point of actual curven and the one before last transition
point of actual curve

l_y=0;% last transition point of actual curve (the output value become zero)
l_y_1=0;% the one before last transition point of actual curve

for iii=2:length(m)
    if y(iii-1)>0&y(iii)==0
        l_y=iii-1;
        for nn=iii:length(m) % set all the data after the last point to zero
            y(nn)=0;
        end
    end
    if y(iii)>y(iii-1)
        l_y_1=iii-1;
    end
```

```
end

ref=x(l_y)+0.5*(x(l_y)-x(l_y_1)); % here I use the time as analog input since the time is proportional to
input current.
%from the plot, ref=time of 64th transition + 0.5*((64th transition -63rd transition)

lsb=ref/64;
s=[];% for ideal curve
s(1)=0;
step=.5*lsb;

% the following For block construct a ideal transfer curve, based on the
% actual extracted X-axis data.
q=0;
for n=2:length(m)



    if (x(n)>step) & (x(n-1)<step) %find the ideal transition point
        step=step+lsb;
        s(n)=s(n-1)+1.5/64;

        if s(n)>1.48 % for ideal DAC, V11...11=Vref-VLSB=1.4766
            l_s=n-1;% last point of ideal curve
            s(n)=0;% overloaded in ideal curve
        end

        q=q+1;
        if q==63
            last_transition=n;% the last transition point in ideal curve
        end
    else
        s(n,1)=s(n-1,1);
    end
end

% the following block is to find the Offset error, Gain error, and DNL
dnl=[];
```

```
inl=[];
x_r=[];% for the Offset error and Gain error free curve
x_r(1)=0;
j=-1;
k=1;
l=0;
dnl_max=0;

for n=2:l_y
    if y(n)>y(n-1)% find the transition point in actual curve
        u=n;
        j=j+1;

        if j==0
            E_off=(x(n)-0.5*lsb)/lsb% Offset error
        else
            dnl(j,1)=x(n);
            dnl(j,2)=abs((x(n)-x(k))/lsb-1); % x(n) is current transition point, while x(k) is the previous transition point
            if dnl(j,2)>dnl_max
                dnl_max=dnl(j,2); % maximum dnl error
            end
            if j==62
                E_gain=(x(n)-x(last_transition))/lsb-E_off% Gain error (must remove the E_off first)
            end
        end
        k=n;
    end
end

for w=2:length(m)
    if y(w)>y(w-1)% find the transition point in actual curve
        l=l+1;
    end

    if y(w)==0 & y(w-1)>0
        x_r(w)=x(w);
        else
            x_r(w)=x(w)-E_off*lsb-l*E_gain*lsb/63; % creat the Offset error and Gain error free curve
        end
    end

    plot(x,s, x_r,y,'r');


    o=0;
    inl_max=0;
    d_max=0; %for absolute accruacy
    for v=2:l_y
        if y(v)>y(v-1)% find the transition point in actual curve
            o=o+1;
            for i=2:l_s
                if abs(s(i)-y(v))<0.01 & abs(s(i-1)-y(v))>0.01
                    inl(o,1)=x_r(v);
                    inl(o,2)=abs((x_r(v)-x(i))/lsb);
                    if inl(o,2)>inl_max
                        inl_max=inl(o,2);
                    end
                    if abs(x(v)-x(i))/lsb>d_max
                        d_max=abs(x(v)-x(i))/lsb;
                    end
                end
            end
        end
    end
    % plot(inl(:,1),inl(:,2));
    % pause
    % plot(dnl(:,1),dnl(:,2));

    a_a=6-log(d_max)/log(2)
    a_r=6-log(inl_max)/log(2)


    %following code to plot the fig
    xp=x(:)/ref;
```

```
x_rp=x_r(:)/ref;
inl(:,1)=inl(:,1)/ref;
dnl(:,1)=dnl(:,1)/ref;

subplot (2,2,1);
plot(xp,s,'c', xp,y,'k');
xlabel('Analog Input                    Unit: FSR');
ylabel('Digital output');
title ('Aactual curve VS. ideal curve');
legend ('ideal curve','actual curve',2);

subplot (2,2,2);
plot(xp,s,'c', x_rp,y,'k');
xlabel('Analog Input                    Unit: FSR');
ylabel('Digital output');
title ('Actual curve after Offset and Gain error removed VS. ideal curve');
legend ('ideal curve','actual curve',2);
axis([0,1,0,1.5]);

subplot (2,2,3);
plot(inl(:,1),inl(:,2));
xlabel('Analog Input                    Unit: FSR');
ylabel('INL                Unit LSB');
title ('INL');
axis([0,1,0,2]);

subplot (2,2,4);
plot(dnl(:,1),dnl(:,2));
xlabel('Analog Input                    Unit: FSR');
ylabel('DNL                Unit LSB');
title ('DNL');
axis([0,1,0,1]);

E_off
E_gain
dnl_max
inl_max
```

a_a

a_r

A-4: Definition of the metrics for the performance of ADC

Based on the reference [28], the definitions adopted in our analysis to examine the performance of our ADC are presented below.

First we define I_B as the current for the digital output word B, which is the sampled value of the input current, at the transition point between two adjacent digital words representing the sampled analog signal. The adjacency is defined to be the beginning of a transition from the (j-1)st and jth the word. Here B is a 6-bit digital word of the output of our ADC.

- Offset error: it reflects the deviation of the fist transition between the actual and ideal transfer response, and it is mathematically defined as

$$E_{off} = \frac{I_{000001} - \frac{1}{2} I_{LSB}}{I_{LSB}} \tag{A.1}$$

  Where,

  I_{000001} is the input current while the digital output 000000 to 000001 transition occurs.

  I_{LSB} is given by $\frac{I_{FSR}}{2^R}$, here R is the number of resolution of the ADC, which is equal to 6 in our design.

- Gain error: it reflects the deviation of the last transition between the actual and ideal transfer response after the Offset error is removed, and it is mathematically defined as

$$E_{gian} = \frac{I_{111111} - I_{000001}}{I_{LSB}} \text{-}(2^{R} \text{-}2) \qquad \text{(A.2)}$$

Where,

I_{111111} is the input current while the digital output 111110 to 111111 transition occurs.

- INL (Integral Nonlinearity) error: it reflects how closely the actual transfer response curve resembles a straight line. It is mathematically defined as,

$$INL = \left| \frac{I_{B\_after\_Offset\_error\_and\_Gain\_error\_removed}}{I_{LSB}} - B \right| \qquad \text{(A.3)}$$

Where,

$I_{B\_after\_Offset\_error\_and\_Gain\_error\_removed}$ is I_B after the Offset error and Gain error are removed, and it is given by,

$$I_{B\_after\_Offset\_error\_and\_Gain\_error\_removed} = I_B - E_{Off} \cdot I_{LSB} - \frac{B}{2^{R}-1} \cdot E_{Gain} \cdot I_{LSB} \qquad \text{(A.4)}$$

When we use Eq. (A.3) and Eq. (A.4), we need to transfer B from binary format to its corresponding decimal value first.

Two points about INL error should be mentioned here:

i) From Eq. (A.3), one may get a set INL values corresponding to each transition current. However, it is customary to consider only the maximum value, say INL_{max}.

ii) Depending on how we choose the straight-line transfer characteristic, there are best-fitting-line INL and end-point INL. We used end-point INL since it is easier for calculation.

- DNL (Differential Nonlinearity) error: It reflects the deviation between an actual step width (the difference between two adjacent transient current) and the ideal value, namely one I_{LSB}. It is mathematically defined as,

$$DNL = \frac{I_B - I_{B-1}}{I_{LSB}} - 1 \tag{A.5}$$

Like INL, people usually are interested in the maximum DNL, say DNL_{max}.

- Absolute accuracy: it reflects the maximum difference between the actual transfer response and ideal one. It is expressed by number of effective bits. The mathematical definition is,

$$R_{absolute\_accuracy} = R - \frac{\ln(D_{max})}{\ln 2} \tag{A.6}$$

Where,

R is the resolution of the ADC.

D_{max} is the maximum value of the deviation in the unit of LSB between the actual transition point and it corresponding ideal transition point.

- Relative accuracy: it reflects the maximum difference between the actual transfer response that the Offset error and Gain error are removed, and the ideal transfer response. It is expressed by number of effective bits. The mathematical definition is,

$$R_{relative\_accuracy} = R - \frac{\ln(INL_{max})}{\ln 2} \tag{A.7}$$

Where, R is the resolution of the ADC.

Reference

[1] Sid Deutsch and Alice Deutsch, “Understanding the Nervous System—An Engineering Perspective”, Published by the Institute of Electrical and Electronics Engineers, Inc. ISBN 0-87942-296-3, ©1993

[2] D. H. Hubel, “The Brain”, Sci. Am., vol. 241, Sept. 1979, pp.44-53

[3] Scott Alwyn, “Neuroscience: a Mathematical Primer”, published by Springer, ISBN: 0387954031, ©2002

[4] M. Cheely and T. Horiuchi, “Analog VLSI Models of Range-Tuned Neurons in the Bat Echolocation System”, EURASIP Journal, Special Issue on Neuromorphic Signal Processing and Implementations, 2003, pp. 649-658.

[5] J. Nielson and H. H. Lund, "Spiking Neural Building Block Robot with Hebbian Learning", Proceeding s of the 2003 IEEE/RSJ intl. conference on Intelligent Robots and Systems Las Vegas, Nevada, October 2003, pp. 1363-1369.

[6] Cole, Kenneth Stewart, “Membranes, Ions, and Impulses; a Chapter of Classical Biophysics”, published by Berkeley, University of California Press, 1968.

[7] Dee. U. Silverthorn with William. C. Ober, Claire W. Garrison, and Andrew C. Silverthorn, “Human Physiology – An Intergrated Approach”, Prentice Hall, ISBN 0-13-262528-8, ©1997

[8] L. Hodgkin and A. F. Huxley, “A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerve”, J. Physiol., vol. 117, 1952, pp.500-544

[9] J. Palm William, "Introduction to MATLAB 7 for Engineers", published by McGraw-Hill, ISBN: 0072548185, ©2005

[10] "Communication, Computation, and Memory in Biological Systems", <http://monitor.admin.musc.edu/~cfs/cps266/>

[11] D. O. Hebb, "The Organization of Behavior: a Neuropsychological Theory", published by Mahwah, N. J. ; London : L. Erlbaum Associates, ISBN 0805843000, ©2002

[12] Rodney A. Rhoades, George A. Tanner, "Medical Physiology", published by Lippincott Williams &Wilkins, ©1995

[13] G. Peter Zhang, "Neural Networks in Business Forecasting", published by Hershey, PA ; London : Idea Group Publishing, ISBN 1591401763, ©2004

[14] Lipo Wang, "Soft Computing in Communications", published by Springer, ISBN 3540405755, ©2004

[15] Shu-Heng Chen, Paul P. Wang, "Computational Intelligence in Economics and Finance", published by Springer, ISBN 3540440984, ©2004

[16] Parodi Mauro and Storace Marco, "On a Circuit Representation of the Hodgkin Huxley Nerve Axon Membrane Equations", International J. of Circuit Theory and Applications, vol.25, 1997, pp.115-124

[17] Rodney Douglas and Misha Mahowald, "A Construction Set for Silicon Neurons", from "An Introduction to Neural and Electronic Networks", published by Academic Press, 1995, pp. 277-296

[18] Jonghan Shin and Christof Koch, "Dynamic Range and Sensitivity Adaptation in a Silicon Spiking Neuron", IEEE transactions on neural networks, Vol. 10,

No. 5, Sept. 1999, pp. 1232-1238

[19] Federico Faggin and Carver Mead, "VLSI Implementation of Neural Networks", from "An Introduction to Neural and Electronic Networks", 2nd Edition, published by Academic Press, 1995, pp. 297-314

[20] Elisabetta Chicca, Davide Badoni,Vittorio Dante, Massimo D'Andreagiovanni, Gaetano Salina, Luciana Carota, Stefano Fusi, and Paolo Del Giudice, "A VLSI Recurrent Network of Integrate and Fire Neurons connected by Plastic Synapses With Long-Term Memory", IEEE Transactions on Neural Networks, Vol. 14, No.5, September 2003, pp. 1297-1036

[21] Bo Liu, James F. Frenzel, "A CMOS Neuron for VLSI Circuit Implementation of Pulsed Neural Networks", IEEE 0-7803-7474-6-02, 2002

[22] D. O'Neill, J. C. Lin, and Y.-C. Ma, "Estimation and Verification of a Stochastic Neuron Model", IEEE Trans. Biomed. Eng., vol. BME-33, July 1986, pp. 654-666

[23] Rabin Raut , "A novel VCT for analog IC applications", IEEE, Circuits and Systems II: Analog and Digital Signal Processing, December, 1992

[24] Hkan Traff, Torbjorn Holmberg and Sven Eriksson, "Application of Swithched-Current Technique to Algorithmic DA and AD Converter", IEEE, CH 3006-4/91m – 1549

[25] Kyoya Kawamura, Takashi Matsubara and Yoshiaki Koga, "Technologies for Designing Dependable A/D Converters", IEEE, 1996, 0731-3071/96

[26] D.W. Tank and J. J. Hopfield, "Simple 'Neural' Optimization Networks: An A/D Converter, signal decision Circuit, and a Linear Programming circuit",

IEEE Transactions on Circuits and Systems. Vol.CAS-33, No.5, May 1986, pp. 533-541,

[27] J. van de Plassche Rudy, H. Huijsing Johan, and Sansen Willy, “Analog Circuit Design : High-speed Analog-to-digital Converters, Mixed-signal Design, PLL’s and Synthesizers”, published by Kluwer Academic Publishers, ISBN 079237956X, ©2000

[28] David Johns, Ken Martin, “Analog Integrated Circuit Design” published by New York : John Wiley & Sons, ISBN 0471144487, 1997, pp. 445-461

[29] “Digital Integrated Circuit Design - Digital Designer's Guide”, <http://www.ami.ac.uk/courses/ami4407_dicdes/dicdes-guide/index.asp>

[30] “All about circuit, Volume IV”, <http://www.allaboutcircuits.com/vol_4/>

[31] “Counters - Starting With a Flip-Flop, Using The 74193 Counter”, <http://www.facstaff.bucknell.edu/mastascu/eLessonsHTML/Logic/Logic5.html>

[32] “Lycoscanada, Flip-Flops”, <http://telnet7.tripod.com/articles/flip_flops.htm>

[33] “Digital Sequential Circuits”, <http://doppler.unl.edu/~bcorner/307/exp4.pdf>

[34] Geoffrey J. Coram, “How to (and how not to) Write a Compact Model in Verilog-A”, Proc. 2004 IEEE International Behavioral Modeling and Simulation Conference (BMAS), 2004, pp. 97-104,

[35] Sumit Ghosh, “Hardware Description Languages : Concepts and Principles”, published by IEEE Press, ISBN 0780347447, 1999

[36] Dan FitzPatrick and Ira Miller, “Analog Behavioral Modeling with the Verilog-A Language” published by Kluwer Academic Publishers, ISBN

0792380444, ©1998

[37] Gordon W. Roberts, Albert K. Lu, “Analog Signal Generation for Built-in-self-test of Mixed-signal Integrated Circuits” published by Kluwer Academic Publishers, ISBN 0792395646, ©1995

[38] Jorge Grilo, Ian Galton, Kevin Wang, Raymond Montemayor, “A 12 mW ADC Delta-Sigma Modulator with 80dB of Dynamic Range Integrated in A Single-Chip Bluetooth Transceiver”, IEEE 2001 Custom Integrated Circuits Conference, 2001, pp.900-909

[39] Quentin Diduck, Martin Margala, “A Low-Power 6-b Integrating-Pipeline Hybrid Analog-to-Digital Converter”, The Proceedings of the 16th Annual IEEE International SOC Conference, September, 2003, pp. 337-340

[40] Hamid Movahedian, Meysam Azin, Mehrdad Sharif Bakhtiar, “A Low Voltage Low Power 8-bit Folding/interpolating ADC With Rail-to-rail Input Range”, IEEE Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on, Vol.1, 2004, pp. I-77- I-80

[41] J. Terada, Y. Matsuya, F. Morisawa and Y. Kado, “8-mW, 1-V, 100-msps, 6-bit A/D Converter Using a Transconductance Latched Comparator”, IEEE ASICs, 2000. AP-ASIC 2000, ISBN 0-7803-6470-8/00, 2000, pp. 53-56

[42] Michael D. Scott, Bernhard E. Boser and Kristofer S. J. Pister “An Ultra-Low Power ADC for Distributed Sensor Networks”, ESSCIRC 2002, pp. 255-258

[43] Chi-Sheng Lin and Bin-Da Liu, “A New Successive Approximation Architecture for Low-Power Low-Cost CMOS A/D Converter”, IEEE Journal of Solid-State Circuits, Vol.38, NO. 1, 2003, pp. 54-62

[44] Holland, “High Resolution, High Linearity Interpolating A/D Converter” in IEEE Int. Test Conf. Proc., 1984, pp. 96-104

[45] Mahmoud Fawzy Wagdy, “Diagnosing ADC Nonlinearity at the Bit Level” IEEE Transactions on Instrumentation and Measurement. VOL. 38, NO. 6. Dec. 1989, pp.1139-1141

[46] K. Kim, “Analog-to-digital Conversion and Harmonic Noises due to the Integral Nonlinearity,” IEEE Trans. Instrum. Meas., vol. 43, Apr. 1994, pp. 151–156

[47] D. Asta and F. H. Irons, “Dynamic Error Compensation of Analog-to-digital Converters,” Lincoln Lab. J., vol. 2, no. 2, 1989, pp. 161–182

[48] M. V. Bossche, J. Schoukens, and J. Renneboog, “Dynamic Testing and Diagnostics of A/D Converters,” IEEE Trans. Circuits Syst., vol. CAS-33, Aug. 1986, pp. 773–785

[49] “Dynamic Testing of High-Speed ADCs”, <http://www.maxim-ic.com/appnotes.cfm/appnote_number/729>