NONUNIFORMLY SAMPLED DIGITAL SIGNAL PROCESSING FOR LOW-POWER BIOMEDICAL APPLICATIONS

HONG YIBIN

NATIONAL UNIVERSITY OF SINGAPORE

2014

NONUNIFORMLY SAMPLED DIGITAL SIGNAL PROCESSING FOR LOW-POWER BIOMEDICAL APPLICATIONS

HONG YIBIN
(B.Eng. (Hons.), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2013

DECLARATION

I hereby declare that this thesis is my original work and it has been written by me in its entirety.
I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Hong Yibin

ACKNOWLEDGEMENT

I would like to express my deepest gratitude towards my supervisor Professor Lian Yong, who gave me the opportunity to conduct cutting-edge research in this challenging and exciting new field. His patient guidance, insightful advice and continuous encouragement played a vital part in my progress and accomplishments. Being his student, I also benefited tremendously from his vast knowledge, clear vision and profound wisdom.

My sincere appreciation also goes to Dr. Xu Yong Ping and Dr. Yang Zhi for their valuable comments during my qualifying exams and inspiring discussions throughout the course of my study. Helps and suggestions from them have ensured the road to my destination was far less bumpy than it could have been.

I also feel truly grateful to all my lab-mates and team members from both the Signal Processing and VLSI lab as well as the Bioelectronics Lab. To name but a few, Dr. Liew Wen-Sin, Dr. Tan Jun, Mr. Xu Xiaoyuan, Dr. Zou Xiaodan, Mr. Chacko John Deepu, Dr. Zhang Jinghua, Dr. Yang Zhenglin, Mr. Zhang Xiaoyang, Mr. Wang Lei, Mr. Zhang Zhe, Mr. Li Yong Fu, Dr. Mahmood Khayatzadeh, Mr. Xue Chao, Mr. Li Ti and Mr. Zhang Daren, their kind help and cheerful support over the years have certainly been indispensable to me.

Last but most importantly, I am deeply indebted to my parents, Mr. Hong Chengrong and Mrs. Xu Deping, as well as my girlfriend Xue Fei, whose unconditional love and selfless care have always been my source of courage and strength to conquer the challenges and stand the pressure during those hard times along the journey. I dedicate this thesis as a humble gift to them.

TABLE OF CONTENTS

SUMMARY

"*It always seems impossible until it's done.*" – Nelson Mandela

The concept of preventive healthcare systems has attracted increasing attention in recent years, wherein the health condition of each individual is to be monitored in real time by wearable or implantable devices. In cases when a symptom is detected in its early stage, immediate treatments can be made before the situation worsens, increasing the chances of healing with reduced cost.

The biggest challenge in designing such monitoring devices is their battery life, as frequent replacement of the batteries would be highly intolerable considering the wearable or implantable nature of these devices. Energy harvesting techniques can be a solution to achieving uninterrupted monitoring in continuous time, wherein the devices would be powered primarily using energy harvested from human bodies, such as heat and body movements. However, the energy can be gained from these sources are very limited in amount, which poses a very stringent requirement on the power consumption of such devices.

Conventional digital signal processing (DSP) systems take samples from analog inputs periodically without considering their statistical properties. For biomedical signals, fast changes only occur in brief moments while most of the time the signal varies slowly. Processing such sporadically-varying signals using the conventional approach gives rise to a large number of samples that carry redundant information, wasting power not only in the Analog to Digital Converter

(ADC), but also in the subsequent digital signal processor (DSP). New signal processing approaches with better energy efficiency for biomedical applications were thus investigated in this work.

With its event-driven nature, level-crossing sampling is well suited for digitizing biomedical signals with burst-type waveforms. However, conventional DSP techniques cannot be directly applied to level-crossing sampled data, as they are not uniformly spaced in time. Based on a study conducted to compare different interpolation methods, a new approach to convert signals sampled using level-crossing analog to digital conversion (ADC) back into uniform format using linear interpolation is proposed. The choices of various design parameters including level-crossing sampling resolution, interpolation frequency and interpolation resolution are examined to achieve a good balance between signal quality and power consumption. Significant energy saving is achievable from the reduction in sampling rate and filter order as compared to a conventional uniform DSP system.

To further improve the energy efficiency of this processing scheme by making fully use of the statistical properties of biomedical signals, a new class of systems called continuous-time (CT) DSP systems have been investigated. Operating in continuous time and discrete amplitude, CT DSP combines the best attributes of both conventional analog signal processing (ASP) and conventional DSP. On one hand, quantized amplitude allows signal processing that involves only ‘0’s and ‘1’s, giving rise to the benefit of noise immunity and programmability. On the other hand, it does not suffer from aliasing by operating in continuous time. More

importantly, its signal-dependent sampling rate also enables power saving both in the ADC and in the DSP for slowly-varying portions of the input, making it a perfect match for biomedical signals.

One major drawback of CT DSP however, is that the processed results cannot be directly saved due to the fact that they are defined in continuous time, as it would require infinite memory to save an infinite number of points. This limits CT DSP to real-time processing tasks only. To solve this problem, a novel timing storage circuit based on memristors is proposed. Its ability to store and reproduce timing information in an analog manner without performing quantization allows CT digital signals to be recorded for later use, extending the benefits of CT DSP to applications that require signal storage.

The implementation of delay blocks in CT digital filters is another big challenge. Since the incoming data are not synchronized to a global clock, registers used to delay signals in conventional digital filters are not applicable in CT DSP systems. As a result, analog delay blocks such as inverter chains and their variants are usually employed to delay CT digital signals with no information loss. Depending on the amount of delay needed, these delay blocks can be very costly in terms of power consumption and chip area. For biomedical signals with very low bandwidth, the required delay is usually in the millisecond range or even longer. It would hence be very difficult if not impossible to implement such long delay using inverter chains. The proposed timing storage circuit is then adopted to delay CT digital signals in a more efficient way.

Lastly, the amounts of delay of the proposed memristor-based delay blocks can be easily adjusted by changing the frequency of an external square wave, allowing the delay blocks to be effectively duplicated without consuming much additional power and area. This makes it possible to apply frequency response masking (FRM) techniques in CT DSP, enabling the design of sharp-transition CT FIR filters with reduced orders and thereby saving power and area. A delta modulator is also proposed to allow for the first time the cascading of CT FIR filters that operate on delta-modulated signals.

Based on circuit simulations with real electrocardiogram (ECG) signals, the memristor-based CT DSP systems proposed in this work was proven to be a good fit for biomedical applications. Significant power saving was achieved compared to the current state-of-the-art CT digital filters.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

A/D	Analog to Digital
ADC	Analog to Digital Conversion / Analog to Digital Converter
ASP	Analog Signal Processing
CT	Continuous-time
CTCA	Continuous-Time and Continuous-Amplitude
CTDA	Continuous-Time and Discrete-Amplitude
D/A	Digital to Analog
DAC	Digital to Analog Converter
DFF	Delay Flip-Flop
DSP	Digital Signal Processing / Digital Signal Processor
DTCA	Discrete-Time and Continuous-Amplitude
DTDA	Discrete-Time and Discrete-Amplitude
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FR	Fixed Resolution
FRM	Frequency Response Masking
GPS	Global Positioning System
IIR	Infinite Impulse Response
JKFF	J-K Flip-Flop
LSB	Least Significant Bit
MOS	Metal Oxide Semiconductor
MSB	Most Significant Bit

S/H	Sample-and-Hold
SDR	Signal to Distortion Ratio
S-G	Savitzky-Golay
SNDR	Signal to Noise and Distortion Ratio
TFF	Toggle Flip-Flop
VR	Variable Resolution

CHAPTER 1

INTRODUCTION

1.1 Motivation

With an unprecedentedly fast pace of our society, people today work and live under tremendous pressure. Factors including unhealthy diet, lack of exercise and environmental pollutions have all contributed to an ever increasing number of patient records in hospitals and clinics every day. In fact, this situation is expected to become even worse with the aging of our population. The conventional healthcare system is therefore facing sever challenges in meeting the needs of everyone due to the limited availability of doctors and medical facilities. Such high demand also leads to continuous increase in healthcare costs, making it less affordable to the general public. This motivates a new prevention oriented healthcare model, in which the health condition of each individual is continuously monitored to allow potential problems to be discovered and treated at the earliest possible stage, maximizing the chances of cure with a minimum need of medical resources [1-3].

In order to achieve the large-scale continuous monitoring required by this prevention oriented healthcare system, wearable or implantable devices that are capable of recording and processing biomedical signals in real time will be needed [4-10]. The acquired signals can be stored locally in the built-in memory and some preliminary processing may be performed to determine the user's health condition. If anything goes wrong, the corresponding part of the recorded signal will be sent to the user's smart phone via a short-distance wireless communication link. The smart phone will then process the received signal more thoroughly to examine whether the problems indeed exist, and if necessary, the results will be sent via the cellular network to medical centers for doctors to decide if any treatments are required. In the case of an emergency, the device will send an alert signal to the smart phone, which will then notify medical centers for immediate response. With the location of the patient provided by the Global Positioning System (GPS) of the smart phone and the biomedical signals recorded by the device, the right treatment can be arranged and delivered to the patient within the shortest amount of time.

Such devices should come in small size and light weight, so that minimal interruption will be caused to our daily activities when they are carried every day. However, battery life is perhaps the greatest challenge in designing such devices, because for implantable devices, we can't afford to open up the chest of a patient every few days just to replace the battery. Of course the situation for wearable devices will be better, but the need of battery-changing will still be a big trouble. Therefore, the ultimate solution is to power such devices using energy harvested from our human bodies, such as heat and body movements [11-15]. This however, poses a very stringent requirement on the power consumption of such devices, as the energy can be gained from these sources are very limited in amount.

Conventional DSP samples and processes an analog signal periodically without considering the statistical properties of it. For biomedical signals, fast changes only occur in short periods while most of the time the signal does not vary significantly. Processing such sporadically varying signals using the conventional approach would give rise to a large number of samples that carry redundant information, wasting power both in the ADC and in the DSP. This motivates my research in new signal processing approaches that use power more efficiently.

1.2 Background

It is widely accepted that signal processing systems can be classified into four categories, depending on whether the signal they process is continuous or discrete

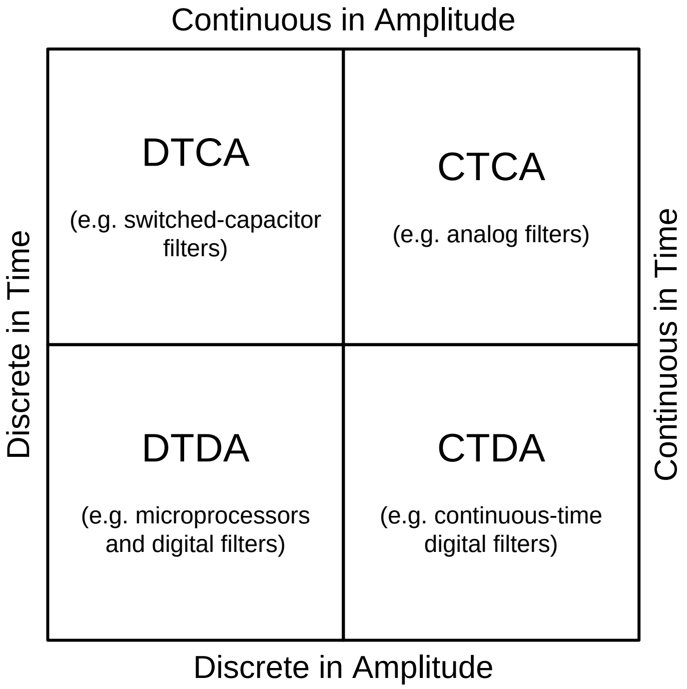

Fig. 1.1 Four categories of signal processing systems: both time and amplitude can be either continuous or discrete, leading to a total of four combinations.

in time and amplitude [16-18], as illustrated in Fig. 1.1.

Traditionally, signal processing systems were implemented using analog circuits that operate in continuous-time and continuous-amplitude (CTCA). Examples like passive RC filters and active op-amp-based Butterworth filters belong to this category. These ASP systems correspond to the first quadrant in Fig. 1.1.

The block diagram of an ASP system is shown in Fig. 1.2 below. Since such systems do not involve sampling or quantizing, aliasing does not occur. Yet due to the nature of analog signals, ASP systems are very sensitive to noise, matching and component tolerances, which not only limits the attainable accuracy, but also leads to high power consumption if large dynamic ranges are required.

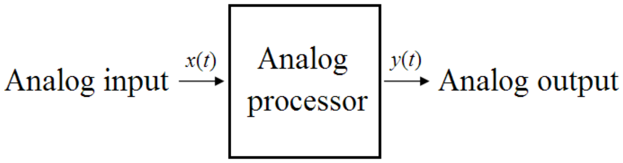

Fig. 1.2 Block diagram of an ASP system.

To overcome the drawbacks of ASP systems, DSP systems that operate in discrete-time and discrete-amplitude (DTDA) were developed with the advent of digital computers. Examples including microprocessors and dedicated DSP chips belong to this category corresponding to the third quadrant in Fig. 1.1. Since all signal sources in the real world are analog in nature, a DSP system needs to first convert the analog input into its digital form before processing in the digital domain. Unless the digital output is to be saved for a later use, digital-to-analog conversion is also needed after processing.

Fig. 1.3 shows the block diagram of a conventional uniform DSP system. Since the analog input varies with time, a sample-and-hold (S/H) circuit is used first to sample the analog input at periodic intervals and hold the sampled value constant at the input of the analog to digital (A/D) converter to allow accurate conversion. The output of the S/H circuit is a staircase-shaped analog signal, which after fed into the ADC is translated into a binary data stream containing only ‘0’s and ‘1’s. It is then possible to process this binary stream using digital circuitry. The resulting output of the digital processor, which is another binary stream, can either be stored in digital media or converted back to its analog form using a digital-to-analog (D/A) converter (DAC). An analog low-pass filter is added to remove any undesirable high-frequency distortions beyond the band of interest.

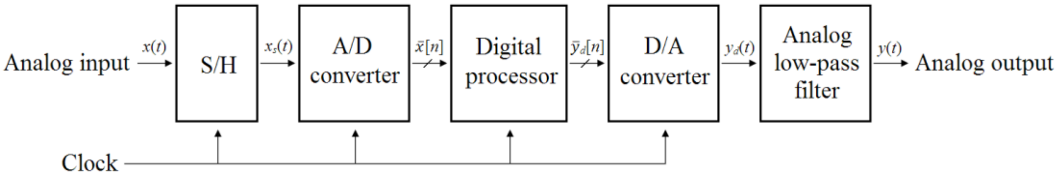

Fig. 1.3 Block diagram of a conventional uniform DSP system.

Since digital signals are binary data streams containing only ‘0’s and ‘1’s, they can be encoded using the lowest and highest voltages in real circuits. Even though component tolerances, temperature changes and other various disturbances will still cause the exact voltage level to fluctuate, there will be no ambiguity in interpreting the signal as long as the fluctuations are within the noise margin – the amount by which the signal exceeds the threshold for a proper ‘0’ or ‘1’. Therefore unlike ASP systems, the accuracy of DSP systems does not depend on the precise values of voltage levels, resulting in better noise immunity.

In terms of signal storage, DSP systems are also much more reliable than ASP systems. Digital signals can be stored almost indefinitely without any loss of information on various storage media such as optical discs and flash memory. These stored binary data can also be easily transferred and duplicated with absolute fidelity. In contrast, stored analog signals deteriorate rapidly as time progresses and the information lost can never be recovered.

In addition, arithmetic operations like addition and multiplication of digital signals can be more easily implemented using binary logic, making it possible to realize highly sophisticated and complex processing algorithms that are much harder if not impossible to realize in the analog domain.

Moreover, since both the signals and the coefficients involved in the processing operations of a DSP system are represented as binary words, the accuracy can be easily adjusted by changing the word length. If very high accuracy is needed, even floating-point arithmetic can be used.

Last but not least, there are no big problems for DSP systems to deal with signals of very low frequencies, while for ASP systems it will need inductors and capacitors that are physically very large in size.

Despite all these advantages mentioned above, DSP systems also have their disadvantages as compared to ASP systems. First and foremost, DSP systems suffer from aliasing because of its use of sampling and quantizing. More details about aliasing will be discussed in Chapter 2. Second, according to the Nyquist-Shannon sampling theorem, the sampling frequency has to be at least twice the highest frequency component present in the analog input. However, due to the finite operating speed of digital circuitry, the sampling frequency cannot increase indefinitely. This limits the applicable frequency range of DSP systems. For example, radio-frequency (RF) signals in the GHz frequency range cannot be processed in the digital domain with the current technology. Lastly, DSP systems usually consume lots of power, especially when compared to analog passive filters that consume almost zero power. This is because of its more complex structure consisting of multiple blocks.

The advantages and disadvantages of DSP systems as compared to ASP systems are summarized in Table 1.1 below.

Table 1.1 Advantages and disadvantages of DSP systems.

Advantages	**Disadvantages**
Better noise immunity	Aliasing
More reliable signal storage	Inability to process high-frequency signals
Possible complex processing	High power consumption
Easier control of accuracy	
Ability to process low-frequency signals	

To overcome the drawbacks of aliasing and high power consumption while preserving the benefits listed above, the relatively unexplored CT DSP systems which operate in continuous-time and discrete-amplitude (CTDA) were chosen to be the main focus of my research. Nevertheless, other nonuniformly sampled digital signal processing approaches were also investigated.

1.3 Research Contributions

In this work, new nonuniformly sampled digital signal processing approaches were developed, and improvements were also made on existing ones to make them more suitable for low-power biomedical applications.

To take advantage of the statistical properties of biomedical signals, a new signal processing scheme combining level-crossing sampling and conventional uniform DSP with the aid of linear interpolation was proposed. It was show by an example that a system designed using this processing scheme was able to achieve 88.8% reduction in the sampling rate and 92.6% reduction in the order of the filter. Designed using a 0.35 μm technology, the linear interpolator for this system consumed an average power of 12.1 μW under a 3.3-V supply.

With signal dependent power consumption not only in the digitization part, but also in the processing part, CT DSP is believed to be an ideal choice for biomedical signals. However, the inability of signal storage and power consuming delay implementation used to be the two main obstacles to its adoption in biomedical applications.

By making use of the memory effect of memristors, a timing storage circuit was proposed to allow the recording and reproducing of CT digital signals, which extended the benefits of CT DSP to applications that require signal storage. Various design considerations and practical challenges were analyzed in details. Circuit simulation verified the feasibility of this approach.

More importantly, it was proven that the delay blocks in CT DSP systems could also be replaced by the proposed timing storage circuits, enabling significant power and area saving for low-frequency biomedical applications. An ECG signal processing example using the proposed method achieved more than 20% power saving compared to the current state-of-the-art CD DSP system implementations, without even considering the much older process and higher supply voltage used. With a 0.35 µm process, a 15-tap CT FIR filter designed using this method consumed an average power of 6.196 µW under a 3.3-V supply.

Lastly, the tunability of the proposed memristor-based delay implementation also enabled the use of FRM techniques in designing sharp-transition CT FIR filters with reduced filter orders. A delta modulator was proposed to allow for the first time the cascading of CT FIR filters that operate on delta-modulated signals. As an example, a CT FRM high-pass filter with a combined order of 40 was designed using the same 0.35-µm process. This filter consumed a total power of 28.0 µW under the same 3.3-V supply, which was about 75.2% lower than the power consumed by a 168th-order filter capable of achieving the same frequency response specifications.

Shown below is a list of publications achieved in this work:

1. Y. Hong, I. Rajendran, and Y. Lian, "A new ECG signal processing scheme for low-power wearable ECG devices," in *Proc. 2011 Asia Pacific Conf. Postgraduate Research in Microelectronics and Electron. (PrimeAsia'11)*, 2011, pp. 74-77.
2. Y. Hong, Z. Xie, and Y. Lian, "Wireless wearable ECG sensor design based on level-crossing sampling and linear interpolation," in *Proc. 2013 IEEE Int. Symp. Circuits Syst. (ISCAS'13)*, 2013, pp. 1300-1303.
3. Y. Hong and Y. Lian, “A memristor-based continuous-time digital FIR filter for biomedical signal processing,” *IEEE Trans. Circuits Syst. I* (Resubmitted after revision).

1.4 Organization of the Thesis

This thesis is organized as follows:

In Chapter 2, the drawbacks of conventional uniform DSP is reviewed to show why it is not an energy-efficient choice for biomedical signals with long periods of inactivity. Some projects based on uniform DSP is also briefly mentioned.

In Chapter 3, a new signal processing scheme combining level-crossing sampling and conventional uniform DSP with the aid of linear interpolation is presented.

In Chapter 4, a literature review of CT DSP is conducted to show why it can be a perfect fit for biomedical signals. Two major problems of CT DSP that make it hard to be applied in biomedical applications are discussed: the first is its inability of signal storage, and the second is its power and area costly delay implementation for low-frequency signals.

In Chapter 5, a memristor-based timing storage circuit is proposed to allow the recording and reproducing of CT digital signals. Various design considerations and practical challenges are analyzed in details.

In Chapter 6, the proposed timing storage circuit is used to replace the delay blocks in CT DSP systems. Simulation on ECG signals show significant power saving compared to the current state of the art.

In Chapter 7, the use of FRM techniques in designing sharp-transition CT FIR filters with reduced orders is discussed. A delta modulator is proposed to enable for the first time the cascading of CT FIR filters.

In Chapter 8, a summary about my research in this work is made to conclude this thesis.

CHAPTER 2

CONVENTIONAL UNIFORM DSP

2.1 Theories

In a conventional uniform DSP system, the analog input is first sampled periodically before being converted into its digital form using an ADC. The resulting outputs are therefore uniformly spaced in time, i.e. the time intervals between any two consecutive samples are equal to the same sampling interval T.

Mathematically, uniform sampling is equivalent to multiplying the analog input $x(t)$ with a unit impulse train (also known as the Dirac comb function) that has a period of T [19, 20]. The sampled signal $x_s(t)$ can then be expressed as

$$x_s(t) = x(t)\sum_{n=-\infty}^{\infty}\delta(t-nT) = \sum_{n=-\infty}^{\infty}x(nT)\delta(t-nT), \qquad (2\text{-}1)$$

where $\delta(t\text{-}nT)$ is a unit impulse (also known as the Dirac delta function) at nT, and integer n is the sample index. From this expression it is obvious that $x_s(t)$ is another impulse train scaled to the values of $x(t)$ at the corresponding time instants. In other words, the sampled signal is zero everywhere except at instants t=nT. Thus for a signal of finite duration, a finite number of samples will be generated for subsequent processing and storage. Information contained in the original analog input $x(t)$ is then transferred to the sequence

$$x[n] = \{\dots, x(-t), x(0), x(t), x(2t), x(3t), \dots\} \qquad (2\text{-}2)$$

It should be remembered that all the elements in the sequence $x[n]$ are still continuous in value, since quantization has not been performed yet.

A graphical illustration of uniform sampling is shown in Fig. 2.1. Even though the sampled signal $x_s(t)$ are considered to be discrete in time, electrically it is still analog in nature: the signal values between consecutive samples are just zero, rather than undefined.

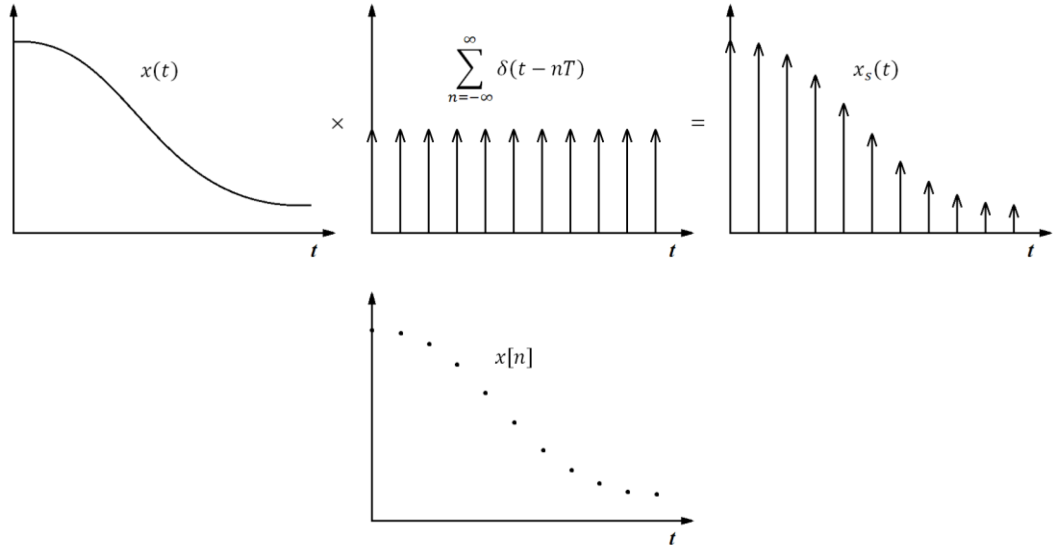

Fig. 2.1 Uniform sampling of an analog signal $x(t)$ in the time domain.

All the above discussions are based on signals in their time-domain representations. It is sometimes more efficient however, to investigate signals in the frequency domain, as it provides information about the frequency content present in a signal, which are not easily visible from the time-domain waveforms. The frequency-domain representation (also referred to as spectrum) of a signal can be obtained by taking the Fourier transform of its time-domain representation. The Fourier transform $X(f)$ of a continuous-time signal $x(t)$ is given by

$$\mathrm{F}\{x(t)\} = X(f) = \int_{-\infty}^{\infty} x(t)\, e^{-j2\pi f t} dt, \tag{2-3}$$

where F{} denotes the Fourier transform operation.

To investigate uniform sampling in the frequency domain, Fourier transform is performed on the sampled signal $x_s(t)$. From (2-1) $x_s(t)$ is equal to the product of $x(t)$ and the Dirac comb function $\sum_{n=-\infty}^{\infty} \delta(t - nT)$. Since multiplication in the time domain is equivalent to convolution in the frequency domain, the frequency transform $X_s(f)$ of $x_s(t)$ is equal to the convolution of $X(f)$ and the Fourier transform of the Dirac comb function:

$$\begin{aligned}
\mathrm{F}\{x_s(t)\} &= \mathrm{F}\{x(t) \times \textstyle\sum_{n=-\infty}^{\infty} \delta(t - nT)\} \\
&= \mathrm{F}\{x(t)\} * \mathrm{F}\{\textstyle\sum_{n=-\infty}^{\infty} \delta(t - nT)\} \\
&= X(f) * \tfrac{1}{T} \textstyle\sum_{n=-\infty}^{\infty} \delta\left(f - \tfrac{n}{T}\right) \\
&= X(f) * f_s \textstyle\sum_{n=-\infty}^{\infty} \delta(f - nf_s) \\
&= \ f_s \textstyle\sum_{n=-\infty}^{\infty} X(f - nf_s).
\end{aligned} \tag{2-4}$$

It is obvious from (2-4) that the spectrum of $x_s(t)$ is the periodic version of the spectrum of $x(t)$ with a period equal to the sampling frequency f_s. A frequency-domain illustration of uniform sampling is shown in Fig. 2.2.

A signal $x(t)$ is said to be frequency band-limited to B Hertz if its Fourier transform $X(f)$ is zero beyond that frequency [19]. B is also called the bandwidth of *x*(*t*) and is equal to the highest frequency component present in *x*(*t*), as shown by *X*(*f*) in Fig. 2.2(a).

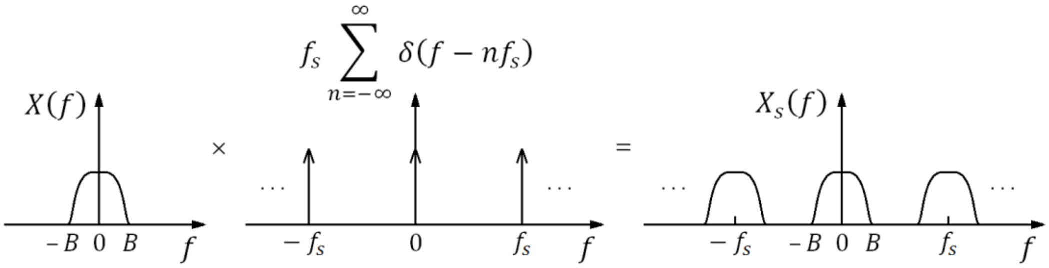
$X(f)$
$-B$ 0 B f
×
$f_s \sum_{n=-\infty}^{\infty} \delta(f - nf_s)$
...
$-f_s$ 0 f_s f
=
$X_s(f)$
...
$-f_s$ $-B$ 0 B f_s f

Fig. 2.2 Uniform sampling of an analog signal *X*(*f*) in the frequency domain.

A band-limited signal *x*(*t*) can be perfectly reconstructed from its discrete-time samples *x*[*n*] if the sampling frequency f_s (the reciprocal of the sampling interval *T*) is at least twice the signal bandwidth B. This is commonly referred to as the Nyquist-Shannon sampling theorem [19]. The minimum sampling frequency f_N = 2B for perfect reconstruction is called the Nyquist sampling rate. Signals sampled at a rate lower than the Nyquist rate are said to be under-sampled. Under-sampling will cause frequency content beyond $f_s/2$ in the original signal to overlap after sampling, as illustrated in Fig. 2.3. Such overlapping makes it impossible to reconstruct the original signal from the sampled output, giving rise to an undesirable phenomenon called aliasing.

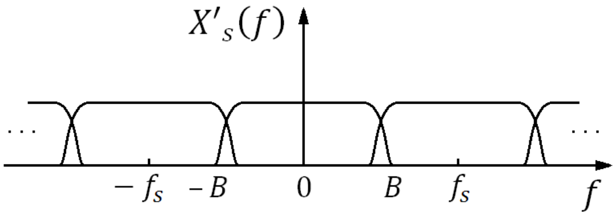

Fig. 2.3 Illustration of aliasing in the frequency domain.

At this point, it may seem like there will be no problems of aliasing as long as a sampling frequency higher than the Nyquist rate is used. However, this is not true for DSP systems, which operate not only in discrete-time, but also in discrete-amplitude. After sampling is performed on the analog input, the value of each sample will next be rounded to a nearest discrete level among a predefined set – a process called quantizing. This finite number of quantization levels will then be encoded using binary words, which can be processed using digital circuitry. Even though the sampled signal $x_s(t)$ does not see overlapping in the frequency domain (as shown in Fig. 2.2), quantizing as a nonlinear operation will introduce a lot more frequency components than are present in $X_s(f)$. These new frequency components will be related to both the frequency content already present in $X_s(f)$ and the sampling frequency f_s, making it difficult to analyze the overall effect. This calls for a new equivalent system model as shown in Fig. 2.4.

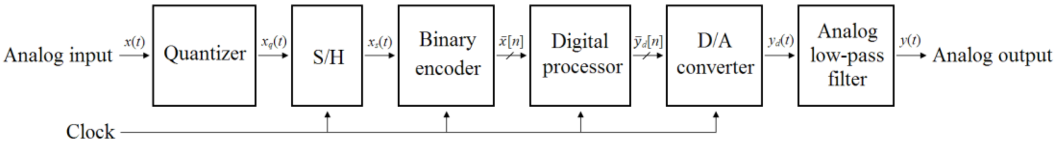

Fig. 2.4 Illustration of aliasing in the frequency domain.

The new model is exactly the same as the block diagram of a uniform DSP system as shown in Fig. 1.3, except that the sequence of sampling and quantizing are switched (In Fig. 1.3, the quantizer and binary encoder are represented as a single ADC block). The two models are equivalent because the input-output relationship of a S/H block followed by a quantizer is identical to that of a quantizer followed by a S/H block [21].

When quantizing is performed directly on the input, harmonic distortions will arise due to the nonlinearity of the quantizer. In the frequency domain, these distortions are located at integer multiple frequencies of the original signal, but with decaying magnitude as frequency increases. The quantized spectrum $X''_q(f)$ of a single-tone sinusoidal signal $x''(t) = \sin(2\pi f_0 t)$ is sketched in Fig. 2.5(a) for illustration. As can be seen, apart from the fundamental component at f_0, $X''_q(f)$ also contains harmonic components at mf_0 (m being any integer), occupying an infinite bandwidth.

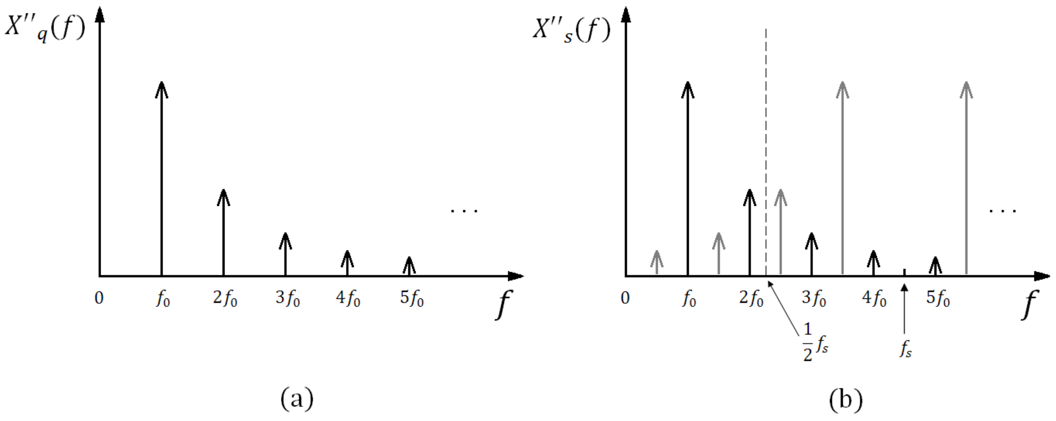

Fig. 2.5 Fourier transforms of (a) the quantized signal $x_q''(t)$ and (b) the sampled signal $x_s''(t)$ (only the first side-lobe at f_s is included for better clarity) from the new equivalent system which performs quantizing before sampling.

The quantized signal $x_q''(t)$ is then sampled at $f_s = 4.5f_0$. The Nyquist rate for this input $x''(t) = \sin(2\pi f_0 t)$ is $f_N = 2f_0$, since f_0 is the highest frequency component in the original signal. Therefore the sampling frequency used is much higher than the Nyquist rate, satisfying the Nyquist-Shannon sampling theorem. However, due to the harmonics introduced during quantizing, frequency overlapping still occurs in the sampled signal spectrum $X''_s(f)$, as shown in Fig. 2.5(b). From this sketch it is noticed that even though shifted versions of the fundamental component are all outside the baseband (indicated by the vertical dashed line at $2.25f_0$, which is half the sampling frequency), shifted harmonic components at pf_s+mf_0 (p being any integer) are aliased into the baseband, causing distortion to the signal of interest. In this sketch, only the first periodic side-lobe is included (for better clarity), but in fact all further side-lobes will have harmonics falling into the baseband, giving rise to a noise floor that covers the entire frequency range.

In summary, conventional uniform DSP systems will always suffer from aliasing, even if the Nyquist-Shannon sampling theorem is satisfied. This is due to its discretization in both time and amplitude.

Discrete-time Continuous-amplitude (DTCA) signal processing systems (second quadrant of Fig. 1.1) which only sample the analog input signal without quantizing it, will not suffer from aliasing if only the sampling theorem is satisfied. This is obvious from the previous discussion. However, DTCA systems share most of the disadvantages of analog systems, like its sensitivity to noise and component tolerances. Since the target of my research is mostly on biomedical applications, such as ECG and electroencephalogram (EEG) systems that usually involve a lot of noise resulted from patient body movement and ambient interference, the ability to reject noise will be vital to the overall performance of a system. DTCA systems that are sensitive to noise are therefore not considered.

The final class of signal processing systems – CTDA systems (fourth quadrant of Fig. 1.1) that only quantize the analog input without sampling it do not have the problem of aliasing either. Even though harmonic distortion will still be present, most of them will be beyond the frequency band of interest and therefore does not affect the system performance, leading to a better signal quality in terms of in-band signal to distortion ratio (SDR). On the other hand, discretization in amplitude gives CTDA systems most of the advantages of digital systems, such as noise immunity and programmability. These benefits make CTDA systems an ideal choice for processing biomedical signals. In fact even more attractive advantages like reduced power consumption will be gained by using level-crossing sampling. More details about level-crossing sampling and CTDA systems will be discussed in Chapter 3 and Chapter 4 respectively.

CHAPTER 3

NONUNIFORM SAMPLING AND UNIFORM CONVERSION

3.1 Uniform DSP vs. Nonuniform DSP

A conventional uniform DSP system takes samples from its analog input, digitizes and processes them periodically according to a clock. Such a system does not take advantage of the statistical properties of the analog input, but instead it samples the analog input at a constant rate that is at least twice the signal bandwidth, no matter how fast or how slow the signal changes.

For biomedical signals, such as ECG signals, fast changes only occur in brief moments while most of time the signal varies slowly. Sampling such sporadically varying signals using the conventional uniform approach would give rise to a large number of samples that carry redundant information, wasting power not only in digitization, but also in subsequent processing [22, 23].

Nonuniform DSP was therefore developed to process signals with a combination of fast changes and long periods of inactivity. In [24-26], level-crossing sampling is first used to digitize the analog input, before Second-order polynomial interpolation is used to convert the resulting nonuniform samples back into uniform format. The uniform digital signals can then be processed in the conventional way. In [27], nearest neighbor interpolation is used for uniform conversion due to its lower error in variance estimation. In [28], A local timer is adopted for timing quantization, and asynchronous filtering is performed by processing the convolution product between the interpolated impulse response and the interpolated input signal. In [29-32], an activity selection algorithm is employed to select and window the active part of non-uniformly sampled signals. Nearest neighbor interpolation is used to resample the active parts of the signal before adaptive rate filtering is performed.

3.2 Level-Crossing Sampling

3.2.1 Concept

Unlike conventional uniform DSP systems, which sample the input at a fixed clock frequency, level-crossing sampling takes the statistical properties of the inputs into consideration [24, 33-36]. Only when a significant change occurs in the input will a new sample be generated. For low-frequency or inactive inputs, the constant-frequency sampling in conventional uniform DSP systems simply wastes power. For level-crossing sampling however, slow inputs naturally result in sparse samples, which lead to lower dynamic power dissipation. During silent periods of the input, the system waits for a change in the signal while dissipating no dynamic power.

This is achieved by employing a level-crossing ADC that generates samples only when the input crosses a predefined set of regularly-spaced amplitude

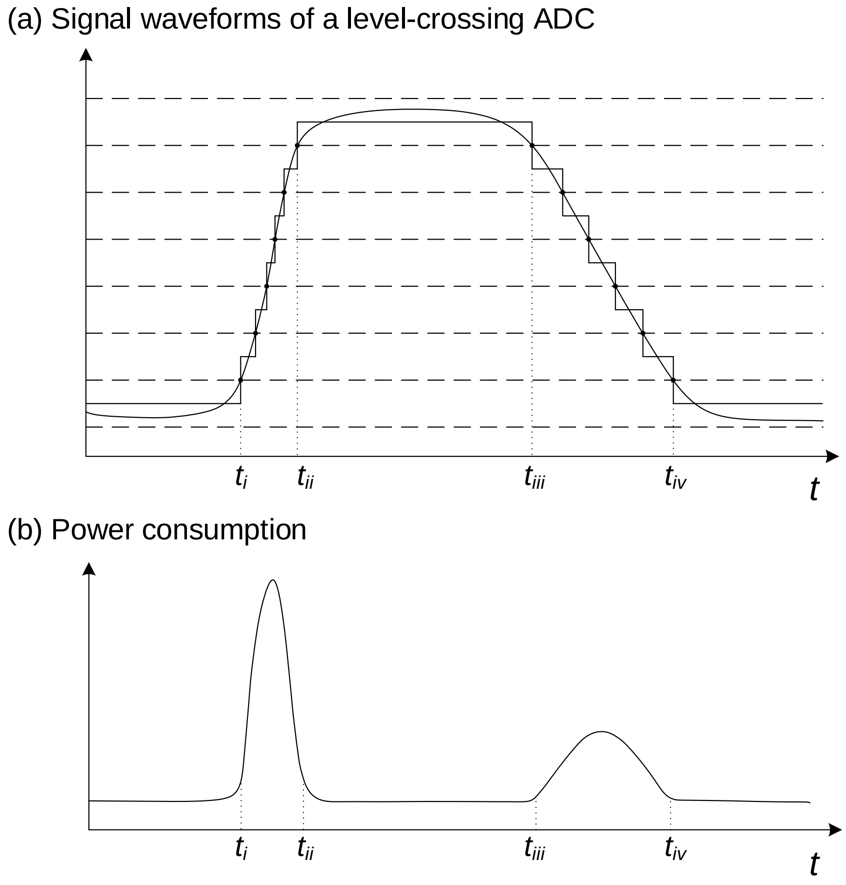

Fig. 3.1 (a) Level-crossing ADC: the thick solid curve represents the analog input, the thin staircase waveform represents the digital output. The set of quantization boundaries are represented by the dashed lines. (b) The corresponding power dissipation of the level-crossing ADC.

boundaries, as illustrated by the dashed lines in Fig. 3.1(a). Each solid line resting in the middle of two dashed lines is used to represent the quantized value for signals falling within the two boundaries. This finite number of solid lines can then be encoded using binary words, yielding the digital representation of a level-crossing ADC.

As can be seen from the power dissipation plot shown in Fig. 3.1(b), significant power is consumed only during the two time intervals t_i-t_{ii} and t_{iii}-t_{iv}, when substantial changes occur in the analog input. The first interval also has a higher peak than the second interval since during t_i-t_{ii} the input changes more drastically and therefore samples are being generated more frequently. During the rest of the time when the input remains more or less constant, the system consumes minimum static power mainly due to biasing current and transistor leakage. Such adaptive-rate sampling of level-crossing ADC makes it well suited for biomedical signals that exhibit burst-type waveforms.

This however creates another problem since the samples obtained from a level-crossing ADC are not uniformly spaced in time, as is the case for a conventional uniform-sampling ADC. Most of the available digital signal processing theories and techniques were developed based on uniform sampling, which means they cannot be directly applied to the digital output of a level-crossing ADC. Although efforts have been made in developing the corresponding theories and techniques for nonuniformly sampled digital signals, no systematic approaches have been formed yet. Techniques like general discrete Fourier transform and Lomb's algorithm face problems like noise in spectra [37, 38].

With these constraints, a new system that combines level-crossing sampling and conventional synchronous processing with the aid of linear interpolation is proposed. The overall computation complexity and power consumption of this new processing scheme is expected to be significantly lower than those of the conventional uniform approach. Matlab simulation on real ECG signals showed that the signal quality is preserved with an average error of less than 2%, while the average sampling rate of such a new system is only 11.2% of the sampling rate of a uniform DSP system.

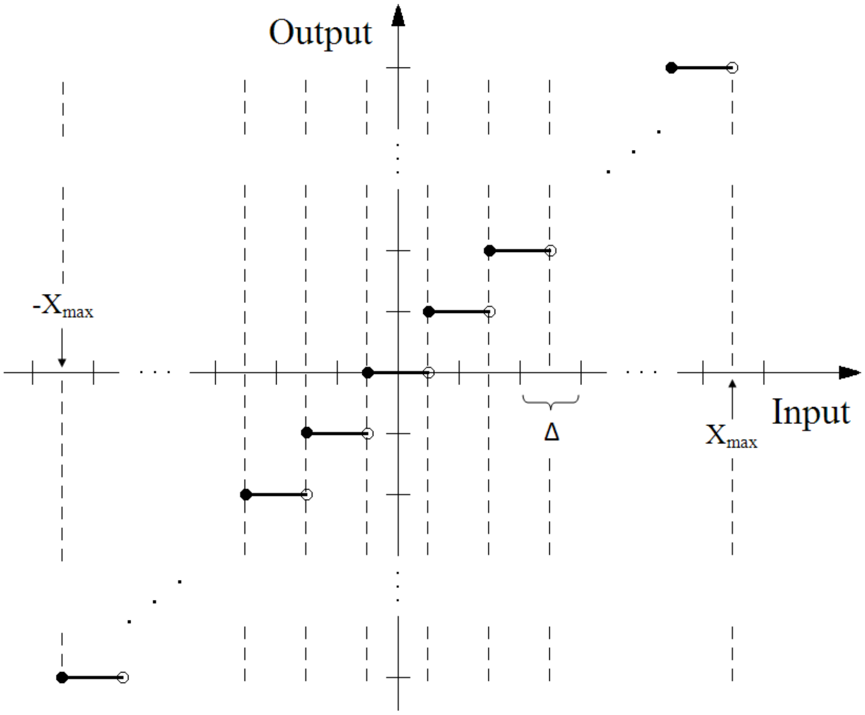

Fig. 3.2 Input-output transfer characteristic of a middle-tread quantizer. *Δ* denotes the quantizer resolution and X_{max} denotes the maximum allowable input range.

3.2.2 Quantization Scheme

The input-output transfer characteristic of an *N*-bit middle-tread quantizer used for level-crossing sampling is shown in Fig. 3.2. The horizontal axis corresponds to the analog input, which is continuous, while the vertical axis corresponds to the digital output, which only takes discrete values. The regularly-spaced dashed lines again represent the quantization boundaries. *Δ* denotes the quantizer resolution and X_{max} is the maximum allowable input range. They can be calculated from:

$$\Delta = \frac{D}{2^N}, \tag{3-1}$$

$$X_{max} = D(1 - \frac{1}{2^N}), \tag{3-2}$$

where *D* denotes the full dynamic range. The reason why the maximum allowable input range is less than the full dynamic range is that one quantization level is purposely left unused, in order to preserve odd symmetry in the transfer characteristic.

3.2.3 Hysteresis

Biomedical signals such as ECG signals are usually very noisy. Such noises

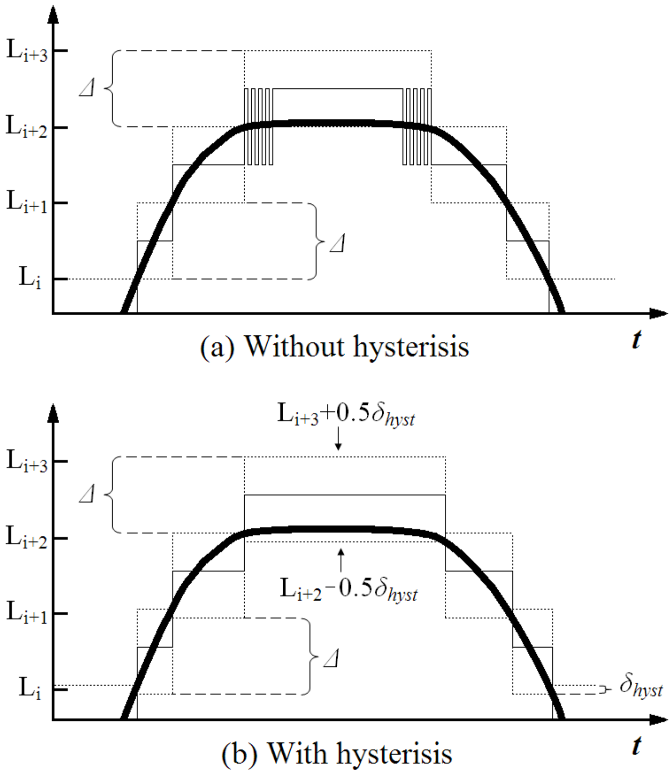

Fig. 3.3 Level-crossing ADC in the presence of noise for (a) quantizers without hysteresis and (b) quantizers with hysteresis. In both plots, the thick solid curve represents the analog input, the thin solid staircase waveform represents the digital output. The quantization boundaries are represented by the dotted lines. Δ denotes the quantizer resolution and δ_{hyst} denotes the amount of hysteresis introduced.

may result from patient body movement, ambient electromagnetic interference as well as other various types of disturbances. This has a severe impact on level crossing sampling. Every time the analog input approaches a quantization boundary, the added noises make the signal crossing that boundary back and forth even though it actually is not, as shown in Fig. 3.3(a). This causes the level-crossing ADC to generate extra unnecessary samples. Such noise-induced samples provide no additional information about the signal of interest, yet they consume significant power in both the ADC as well as the subsequent DSP.

To avoid such undesirable noise-induced toggling, hysteresis is purposely introduced in the quantizer, as illustrated in Fig. 3.3(b). Normally, when the analog input is situated between the two boundaries L_i and L_{i+1}, a new sample will be generated once the following condition is violated:

$$L_i \leq x(t) < L_{i+1}. \tag{3-3}$$

To introduce hysteresis, the above condition is modified as

$$L_i - \frac{1}{2}\delta_{hyst} \leq x(t) < L_{i+1} + \frac{1}{2}\delta_{hyst}. \tag{3-4}$$

In other words, each quantization interval is now widened by an amount equal to δ_{hyst}. For any two neighboring intervals, the upper boundary of the lower interval is no longer aligned with the lower boundary of the higher interval, creating an overlapping between the two. This way, when the analog input increases over a certain boundary causing the ADC to produce a new sample, it must decrease by at least δ_{hyst} before another sample to be generated. This prevents the input from going back and forth between neighboring intervals in the presence of noise, as long as the amount of fluctuation is less than δ_{hyst}.

3.2.4 Simulation with Real ECG Signals

The choice of sampling resolution N is a tradeoff between signal quality and power consumption. A higher sampling resolution which corresponds to denser quantization intervals allows the analog input to be tracked more accurately. On one hand, a finer set of quantization intervals leads to lower quantization errors and therefore improves the signal to noise and distortion ratio (SNDR). On the other hand, due to the event-driven nature of level-crossing sampling, small variations present in the input can only be captured if the quantization interval is narrow enough. However, unlike conventional Nyquist-based DSP, wherein the sampling frequency and quantization resolution are independent, the average sampling rate in level-crossing sampling has an exponential dependence on the quantization resolution used. This implies as N increases, the average sampling rate and the resulting dynamic power consumption also rises very rapidly. Therefore, a resolution just enough to capture the required degree of accuracy will be most desirable.

To find the sampling resolution suitable for ECG signals, Matlab simulations were performed on real ECG signals retrieved from the MIT-BIH database [39]. These signals in the database have already been digitized at 360 Hz and 11 bits using conventional uniform sampling. To make them suitable for level-crossing sampling, these signals were first up-sampled to 36 kHz so that they behave nearly the same as analog signals, which are supposed to be seen by level-crossing ADCs in real scenarios. Extensive level-crossing sampling simulations with different values of *N* were then conducted on different ECG records. With a 3-bit quantizer used in the level-crossing ADC, the ECG waveforms are barely captured: only the QRS waves are captured with all other details lost. Even for the QRS waves the peak heights are not accurately reflected due to the coarse quantization intervals. Better results are achieved when *N*=4: the QRS peaks are more accurate, although the P and T waves are still missed occasionally. The positions of these small waves are however of vital importance for diagnosis, which calls for an even higher sampling resolution. The results become

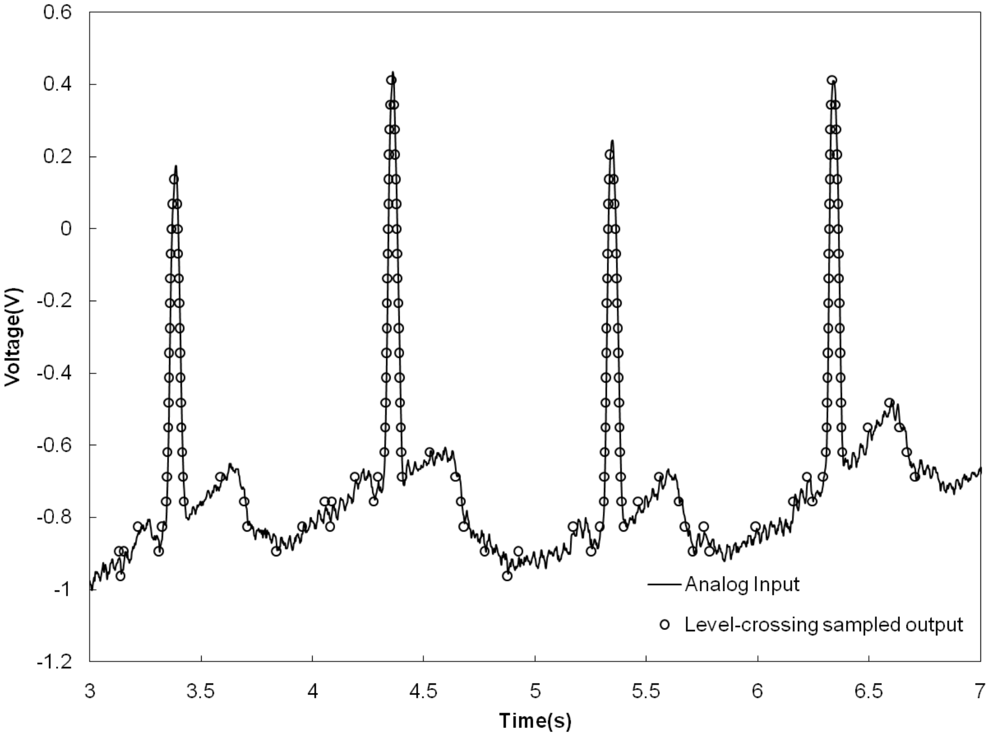

Fig. 3.4 Level-crossing sampling of ECG signal #151 retrieved from the MIT-BIH arrhythmia database. The solid curve represents the original analog input and the series of bubbles represent the level-crossing sampled output.

satisfactory when *N* is increased to 5 bits. Most of the details are well preserved in the digitization process. Fig. 3.4 shows a portion of the level-crossing sampling result of record 151 using a 5-bit quantizer. The resulting average sampling rate is around 40 Hz, with slight deviations among different ECG records. An 88.8% reduction is achieved in the sampling frequency as compared to the original signal, yet negligible sacrifice is incurred in signal quality thanks to the signal-dependent adaptive-rate sampling of level-crossing ADC.

The high-frequency noise present in the original signal also needs to be suppressed by introducing hysteresis. Despite the gain of noise immunity and better robustness, the introduction of hysteresis also causes a certain degree of distortion to the digitized output due to the nonlinearities involved in this operation [21]. A proper choice for the value of δ_{hyst} is therefore needed to achieve a good balance between the pros and cons. The best amount of hysteresis should be one that is just enough to prevent noise-induced toggling so that the amount of distortion introduced is minimized. By observing the ECG records from the MIT-BIH database, it was noticed that most of the fluctuations caused by noise have peak-to-peak amplitudes of no more than a quarter of one quantization interval corresponding to a 5-bit quantizer. Therefore, δ_{hyst} was chosen to be 25%. This parameter of a level-crossing ADC can also be made tunable to accommodate different noise levels under different circumstances.

3.3 Interpolation

3.3.1 Comparison of Interpolation Methods

Due to the many reasons mentioned in Section 3.2.1, the nonuniform samples obtained from the level-crossing ADC must first be converted to uniform format before any traditional DSP techniques can be applied. Of course uniform oversampling would serve the purpose but that again makes the sampling rate high, canceling the benefit of reduced power consumption achieved by level-crossing sampling. Therefore, various interpolation techniques were investigated to achieve the uniform conversion while keeping the total number of samples the same.

Predicting signal values from neighboring samples may sound risky, but considering the event-driven nature of level-crossing sampling it is realized that the signal value between any two consecutive samples is in fact guaranteed to be bounded between those two samples, since any further crossing of quantization boundaries will trigger the ADC to produce additional samples. Therefore, it is fair to conclude that extracting information through interpolation is safe for level-crossing sampled data.

Theoretically, if the time and amplitude values of the nonuniform samples are known to infinite precision, it is possible to compute the amplitude at any time

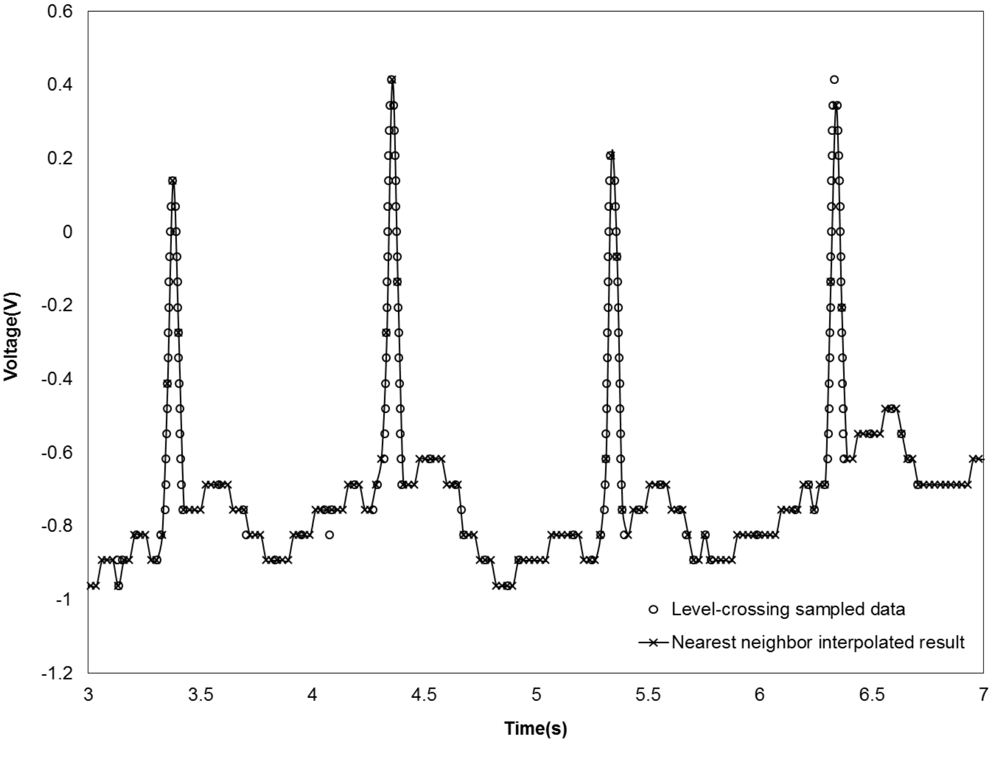

Fig. 3.5 Nearest neighbor interpolation result of the level-crossing sampled data obtained in Fig. 3.4. The series of bubbles represent the level-crossing sampled data, and the solid curve with crosses on it represents the nearest neighbor interpolated result.

instant to infinite precision using an infinite degree interpolation polynomial. This is obviously not the case in practical scenarios, since the sample values are only known to a finite precision (5 bit in this case), and any interpolation scheme must also have a finite order for it to be implementable. Therefore, classical techniques including nearest neighbor interpolation, linear interpolation and cubic interpolation are explored and compared.

Nearest neighbor interpolation is the simplest interpolation wherein the value at the interpolated point is assigned with the value of the closest point [40]. In terms of computational efficiency this is the best since it requires no multiplication at all. However, the signal quality of nearest neighbor interpolation is poor. A staircase-like waveform is obtained after the interpolation as shown in Fig. 3.5.

As shown in Fig. 3.6, for linear interpolation, the weighted average of the two closest nonuniform samples at t_1 and $t_2$is taken to be the value of the interpolated

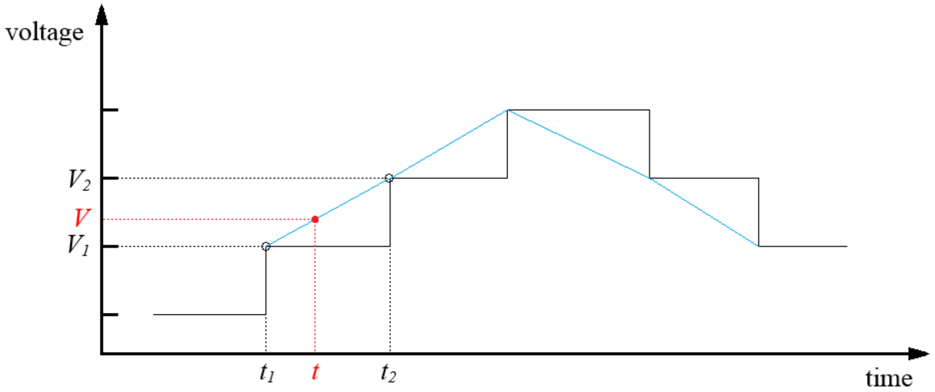

Fig. 3.6 Illustration of linear interpolation

point at *t* [40], given by

$$V(t) = V_1 + (V_2 - V_1)\frac{t-t_1}{t_2-t_1}. \tag{3-5}$$

The computational complexity of linear interpolation is higher than that of the nearest neighbor interpolation, but the signal quality is significantly better. Fig. 3.7 shows the linearly interpolated result from the nonuniform samples obtained in Fig. 3.4. The pointwise average error between the linearly interpolated result and the original input is only 1.53%.

Cubic interpolation forms a third order polynomial that passes through the set

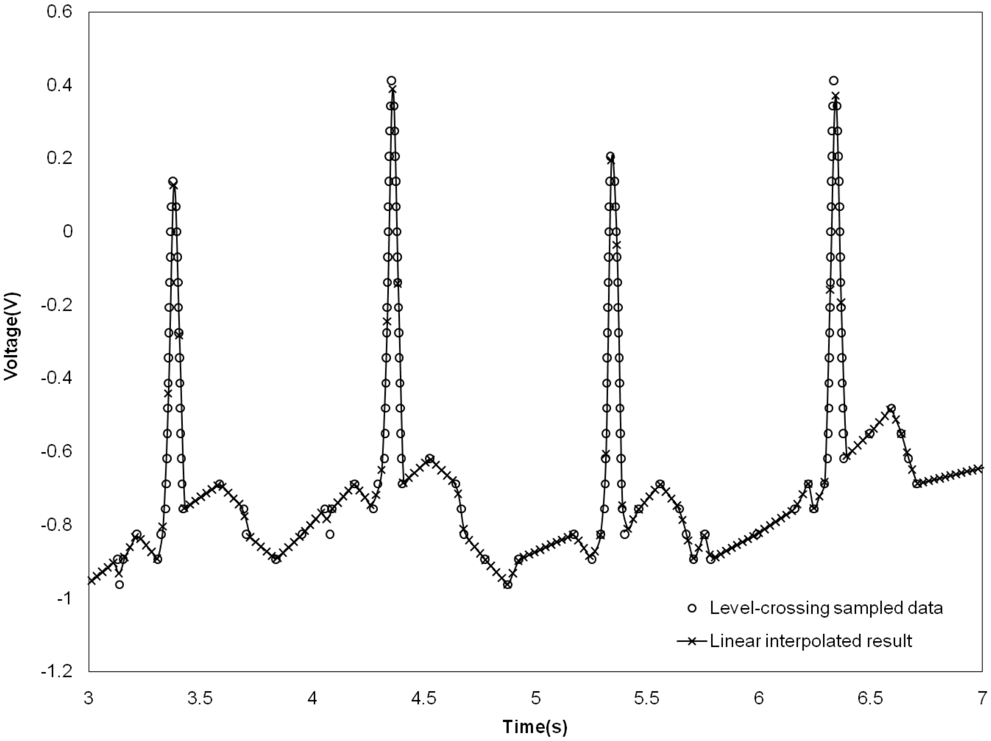

Fig. 3.7 Linear interpolation result of the level-crossing sampled data obtained in Fig. 3.4. The series of bubbles again represent the level-crossing sampled data, and the solid curve with crosses on it represents the linearly interpolated result.

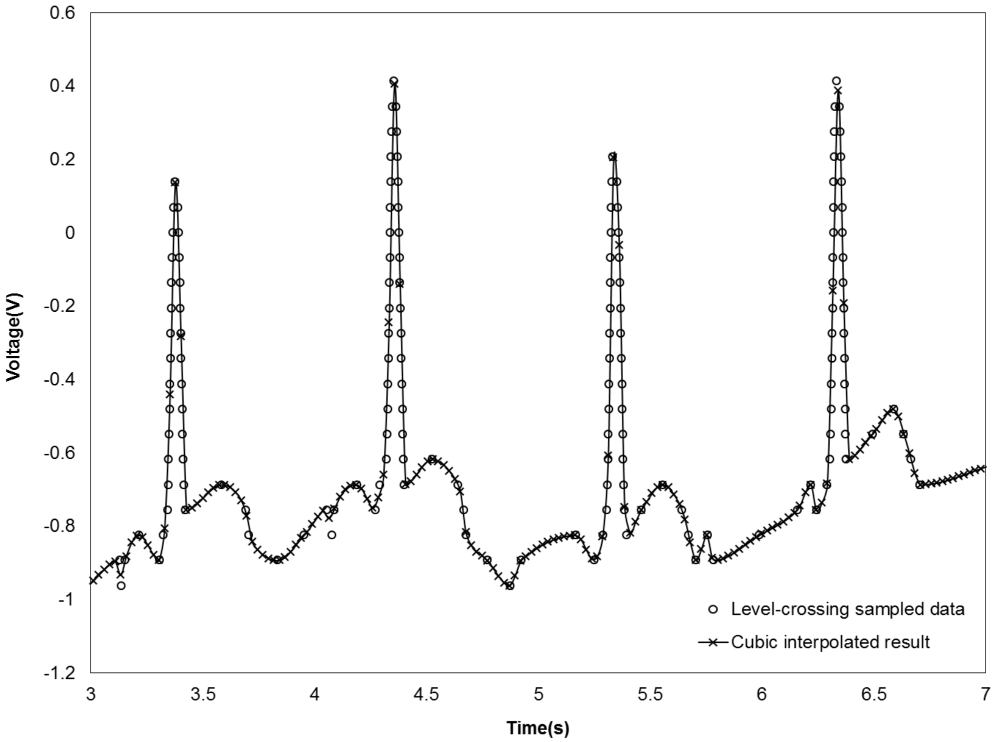

Fig. 3.8 Cubic interpolation result of the level-crossing sampled data obtained in Fig. 3.4. The series of bubbles again represent the level-crossing sampled data, and the solid curve with crosses on it represents the cubic interpolated result.

of interpolated points between any two nonuniform samples [40]. The detailed algorithm is very complex and is not presented here. Such complexity translates into increased circuit area and power consumption, but the signal quality for this case is in fact not any better: The resulting waveform is plotted in Fig. 3.8. The pointwise average error of the cubic interpolated signal is 1.59%, which is slightly higher than that of linear interpolation.

With the above comparison, linear interpolation is chosen for the proposed processing scheme due to its nice balance between signal quality and computational efficiency.

3.3.2 Implementation of Linear Interpolator

Having decided to adopt linear interpolation, the interpolation frequency f_i and interpolation resolution M are two parameters need to be determined. The spacing between neighboring interpolated points is determined by f_i. Similar to the choice of level-crossing sampling resolution N, the choice of f_i is again a tradeoff between signal quality and power consumption. The parts of ECG signals that are most sensitive to changes in the interpolation frequency are the QRS waves, where the fastest changes take place. If the frequency is not high enough, these fast changes are more likely to be distorted than the slowly varying P and T waves. Matlab simulation shows that when f_i =30Hz, the interpolation results suffer severe information loss at the QRS portions. Even when f_i is increased to 40Hz, the peak heights are not always accurately reflected. Satisfactory interpolation cannot be achieved until f_i is increased to 50Hz, when all the QRS waves are well preserved in the linear interpolation process.

When linear interpolation is performed, there is also a need for increase in the quantization resolution. If the interpolation results keep the same quantization resolution of the level-crossing ADC, which is 5 bits in this case, then each interpolated point will have the exact same value as one of the two samples V_1 and V_2, depending on which one is closer. This effectively becomes nearest neighbor interpolation. To get more accurate results through linear interpolation, each quantization interval must be further divided, which means the interpolation resolution M needs to be higher than the sampling resolution N. Extensive simulation is again performed for different values of M. The interpolated signal is

Table 3.1 Linear interpolation error for different resolutions

ECG record index	**Average Error (%)**			
	M=9	*M*=10	*M*=11	*M*=12
100	1.64	0.94	0.64	0.55
101	1.67	0.94	0.65	0.56
102	1.68	0.92	0.61	0.52
103	1.81	1.07	0.75	0.65
104	1.77	1.04	0.74	0.64
105	1.65	0.93	0.65	0.58
106	1.76	0.99	0.69	0.60
111	1.74	0.93	0.61	0.52
221	1.73	0.97	0.68	0.61

compared to the original ECG signal by computing the point-wise average error between the 36kHz-upsampled versions of the two. The results are summarized in Table 3.1. As can be seen from this table, the average error drops significantly as *M* is increased from 9 bits to 11 bits, but the improvements brought by further increase from 11 bits onwards are marginal. The interpolation resolution is therefore chosen to be 11 bits.

Fig. 3.10 shows the block diagram of the linear interpolator implemented. Although interpolations can be carried out in real time, there has to be a small amount of delay between the level-crossing sampled signal and the interpolated signal. This is due to the non-causality of a linear interpolation system: as can be

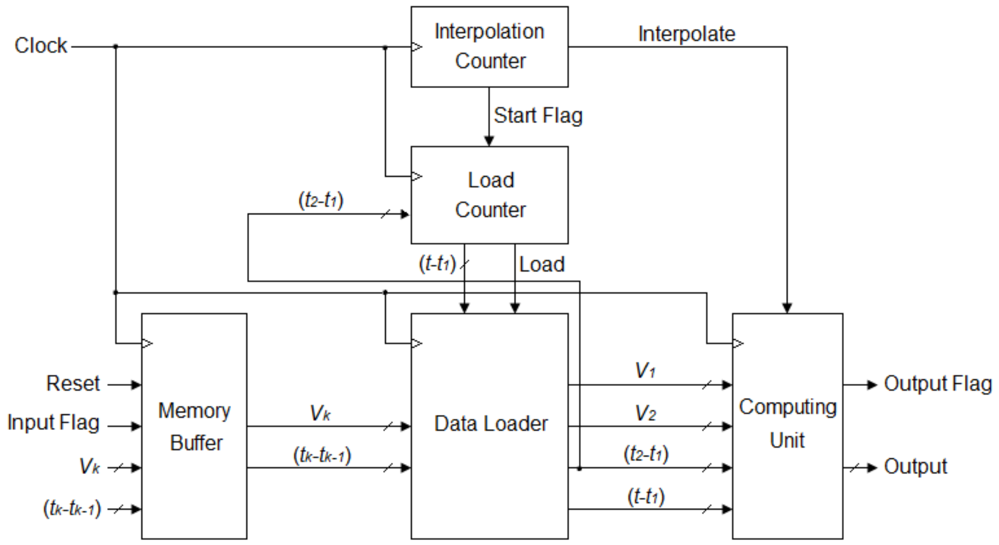

Fig. 3.10 Implementation of the linear interpolator. The main blocks include a memory buffer, a data loader, a computing unit, an interpolation counter and a load counter.

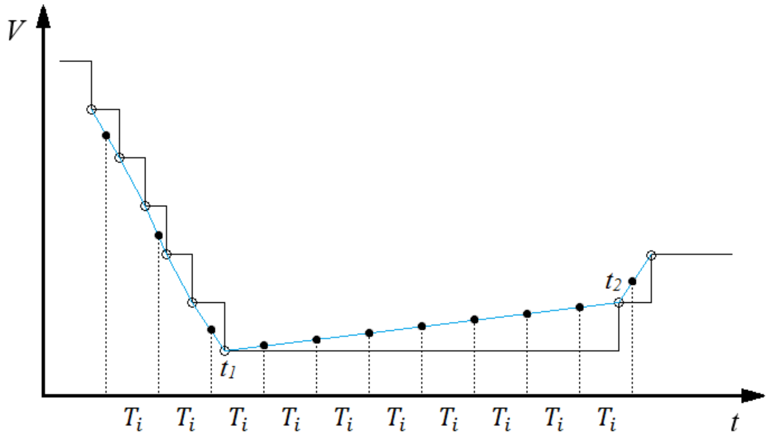

Fig. 3.9 Illustration for the need of a memory buffer in the linear interpolator

seen from Fig. 3.6, the interpolated value at t not only depends on a past sample at t_1, but also depends on a future sample at t_2. As illustrated in Fig. 3.9, in cases when the level-crossing samples, as represented by the bubbles, are much sparser than the interpolation interval T_i, then all the interpolated points sandwiched between t_1 and t_2 will depend on the sample at t_2 that comes much later. A delay that equals the maximum possible time difference T_{max} between t_1 and t_2 must therefore be waited before any interpolation can be carried out. For my case, this value is 0.25 seconds. During this period however, there could also be a large number of samples arriving, depending on the rate of change in the analog input. These samples need to be saved temporarily for the computations performed 0.25s later. A buffer is therefore employed for this purpose. Its capacity is determined in such a way that the maximum number of samples generated by the 5-bit level-crossing ADC within 0.25 seconds can be stored. Every time a new sample is received, both its sample value and its quantized time spacing from the earlier sample will be saved into the buffer with the oldest set of data discarded. However, it is always guaranteed that any sample points generated within the past 0.25s are ready to be retrieved from the buffer.

The interpolation counter serves two purposes. The first is to hold the rest of the circuits idling for 0.25s upon start-up, as samples are being accumulated in the buffer, and then wake them up 0.25s later by asserting the “Start Flag”. The second is to request an interpolation result from the computing unit every 0.02s, according to the 200Hz interpolation frequency.

The load counter is adopted to control data loading from the memory buffer to the data loader, the outputs of which will be fed into the computing unit for computations of interpolated results upon requests from the interpolation counter. Every time a new sample is loaded into the data loader, the load counter will be reset to zero. Once this counter reaches the time spacing of t_2-t_1 for the current pair of samples loaded in the data loader, a new sample will be read into the register for V_2 from the buffer, and the old V_2 value is passed to the register for V_1. The same updates happen for the timing data.

This linear interpolator has been coded using Verilog and synthesized by Synopsis Design Compiler using a 0.35μm CMOS technology. Simulation results showed this block function just as expected. The total power consumption is 12.1μW under 3.3-V supply voltage.

3.4 Digital Filtering

Having the samples converted to uniform format, signal processing can be easily carried out using the available DSP tools. As an example, an FIR high-pass filter is designed to remove the low frequency baseline wandering noise in ECG signal #151.

As before, the Remez exchange algorithm in Matlab is used to design a high-pass FIR filter with parameters summarized in Table 3.2. The interpolated ECG signal is filtered by the designed high-pass filter and the output is shown in Fig. 3.11. It is clear that the baseline wandering noise is successfully suppressed. The

Table 3.2 Design parameters of the high-pass filter.

Stop-band edge	0.5 Hz
Pass-band edge	1.5 Hz
Pass-band ripple	0.01
Stop-band attenuation	17 dB
Filter order	52

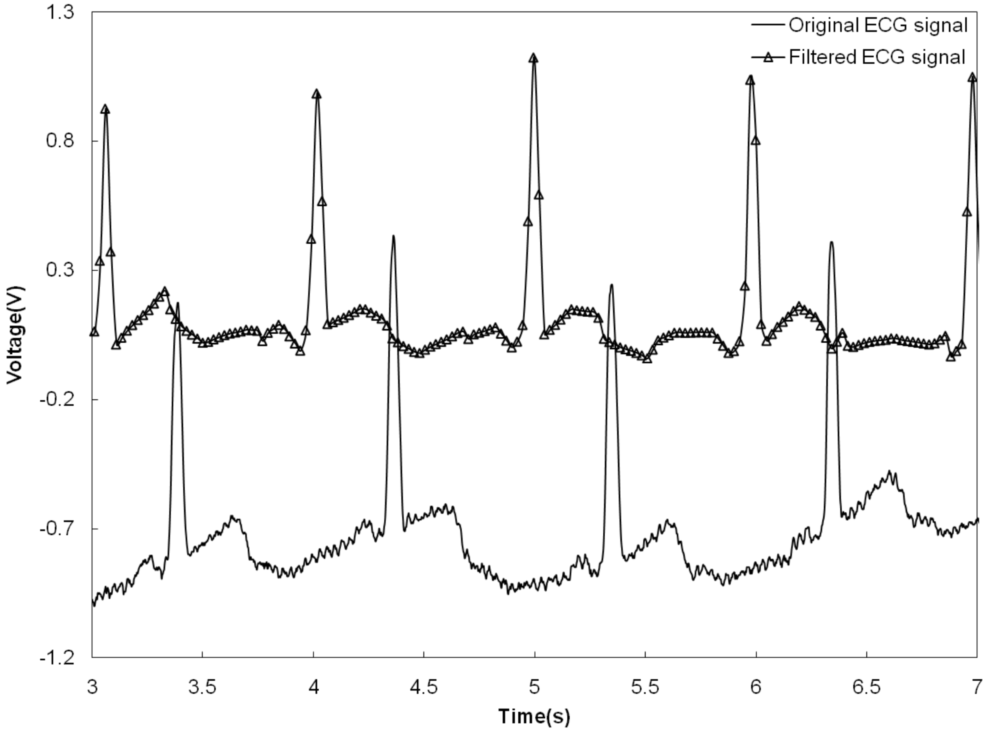

Fig. 3.11 High-pass filtered result of the linearly interpolated data. The solid curve represents the original ECG signal, while the curve with triangles represents the filtered output.

entire waveform also shifts upward due to the removal of the DC component, which carries no useful information for the analysis of ECG signals. The designed FIR filter has an order of 52. By comparison, filters designed for the uniform DSP system has an order of 701 due to the higher sampling rate. This implies further power saving in the DSP by the proposed scheme.

In summary, a new ECG signal processing approach that combines level-crossing sampling with conventional uniform DSP through linear interpolation is presented in this chapter. Significant power and area reduction is achievable from Matlab and circuit simulations. Although ECG signals are used for demonstration, it is believed that other burst-type signals could also derive the same benefits when using the proposed processing scheme.

CHAPTER 4

CONTINUOUS-TIME DSP

4.1 Introduction

In Chapter 3, a new DSP approach based on level-crossing sampling and linear interpolation is proposed. Even though significant power reduction is achievable, this new system still has not made optimal use of the statistical properties of the analog input. As can be seen from Fig. 3.7, after linear interpolation, the slowly-varying portions of the signal again become densely covered with samples. This causes the subsequent DSP block to spend most of its processing power on signals carrying redundant information, just like conventional uniform DSP. The advantages gained through adaptive-rate sampling according to signal activity are therefore lost after the uniform conversion. To overcome this drawback and pass on the benefit of signal-dependent power consumption to the DSP block as well, a new type of system called CT DSP systems was investigated.

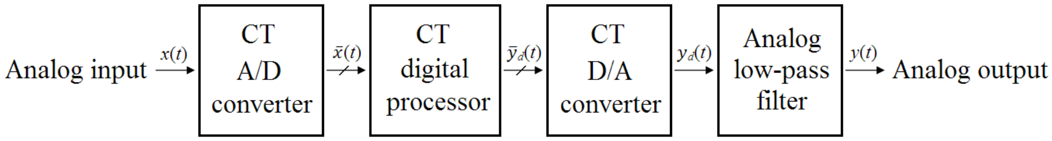

Fig. 4.1 Block diagram of a CT DSP system.

A CT DSP system or a CTDA system (fourth quadrant of Fig. 1.1) is a unique type of system that operates in continuous time and discrete amplitude. The block diagram of a CT DSP system is shown in Fig. 4.1. Symbols with bars on top of them are used to represent digital signals in binary form. As compared to the block diagram of a conventional uniform DSP system shown in Fig. 1.3, two main differences are noticed: the first is the removal of the S/H circuit, which is easy to understand since no discretization of time is performed in CT DSP systems. The second difference is the absence of a global clock. In a conventional uniform DSP system, only periodic samples generated at the clock ticks need to be processed, no other information in between will concern the DSP. Therefore operations of all the blocks can be synchronized to the same clock. In a CT DSP system however, any changes occurring in the analog input is tracked and processed in real time. Besides, the time spacing between consecutive samples is also random and unpredictable. Therefore, all the blocks will have to operate in continuous time without the aid of a global clock.

In the data acquisition part, the same level-crossing sampling discussed in Chapter 3 is used to convert the analog input into digital format, as illustrated in Fig. 4.2. However, since no discretization in time is performed, the word "sample" in CT DSP has a slightly different meaning as it is used in conventional uniform DSP. Instead of extracting information from the analog input in the time domain, a level-crossing ADC in a CT DSP system keeps track of any changes in the input continuously, and updates its digital output every time an amplitude boundary is

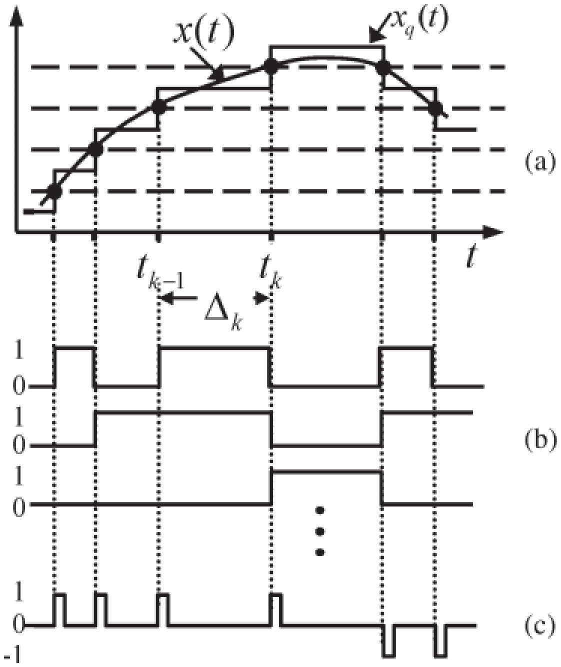

Fig. 4.2 (a) Input signal, level-crossing samples, and quantized signal. (b) Digital representation of quantized signal. (c) Delta-modulation of (b). [47]

crossed. In other words, the level-crossing ADC only quantizes its input and encodes the result using binary words without discarding any signal information between consecutive samples.

Even though samples generated from level-crossing ADC are not uniformly spaced in time, it is possible to process the CT digital signal $\bar{x}(t)$ using conventional uniform approaches. No interpolation as discussed in Section 3.3 is needed any more given the fact that the input of the digital processer is now defined at any instant of time.

Fig. 4.3 shows the architecture of a CT FIR filter to illustrate the concept of CT DSP. Although this structure looks very similar to a conventional FIR filter, in terms of implementation it actually differs in many details. First of all, all the tap delay blocks shown here have to be realized using CT delay elements, like current-starved inverter chains for example, instead of conventional registers.

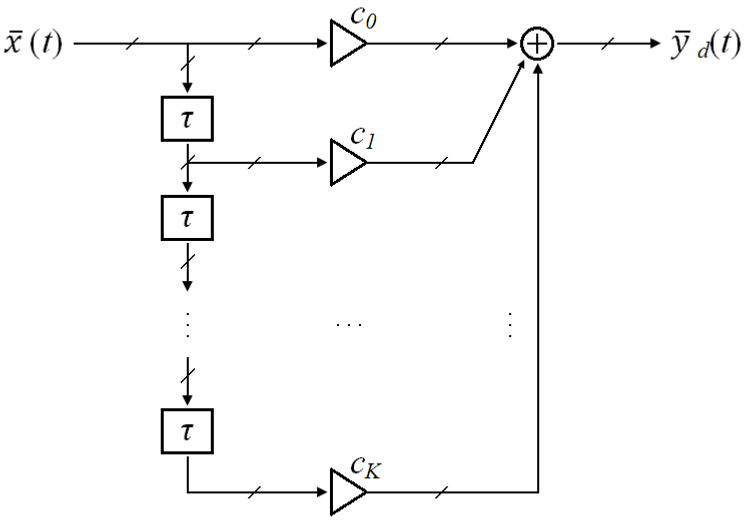

Fig. 4.3 Architecture of a K-th order CT FIR filter.

Second, all the multipliers and adders are free-running combinational circuits not controlled by a clock. Asynchronous handshaking is therefore needed at each interface to prevent glitches from giving rise to momentary errors.

The input-output relationship defined by the CT FIR filter in Fig. 4.3 can be written as

$$\bar{y}_d(t) = \sum_{k=0}^{K} c_k \bar{x}(t - k\tau). \tag{4-1}$$

Taking Laplace transform of (4-1) one obtains

$$\bar{Y}_d(s) = H(s)\bar{X}(s), \tag{4-2}$$

whrere *H*(*s*) is the transfer function given by

$$H(s) = \sum_{k=0}^{K} c_k e^{-sk\tau}. \tag{4-3}$$

To get the frequency response of this filer, set $s=j\omega$ in (4-3):

$$H(j\omega) = \sum_{k=0}^{K} c_k e^{-j\omega k\tau}. \tag{4-4}$$

From (4-4) it is obvious that the frequency response of this CT FIR filter is periodic in ω with period $2\pi/\tau$. In fact it has the exact same form as that of a conventional FIR filter with transfer function

$$H(z) = \sum_{k=0}^{K} c_k z^{-k}, \tag{4-5}$$

where *z* is the complex frequency variable. This implies conventional DSP techniques based on uniform sampling can be applied in the design of CT DSP systems as well [41, 42].

4.2 Matlab Simulation

As an example, Matlab simulations with sinusoidal signals were carried out to verify the performance of CT DSP systems. The time-domain waveforms and the frequency-domain spectra of various signals are plotted in Fig. 4.4 and Fig. 4.5 respectively. The original analog input (shown by the black curve) was a two-tone signal comprising two sinusoids at 13Hz and 41Hz. The purpose of processing was to remove the higher tone in the input and uncover the lower tone. Both conventional uniform DSP and CT DSP were performed to compare their outcomes.

Conventional uniform sampling was first used to obtain the digital signal (shown by the green curve) at a sampling frequency of 100Hz, which is slightly higher than the Nyquist rate. This uniformly sampled signal was then passed through a 20th-order FIR low-pass filter to obtain the digital output (shown by the brown curve). As can be seen from both the time and frequency plots, the resulting signal was a clean sinusoid at 13 Hz with the 41Hz tone removed. The design parameters of the FIR low-pass filter were summarized in Table 4.1 below.

Table 4.1 Design parameters of the low-pass filter.

Pass-band edge	21 Hz

Stop-band edge	33 Hz
Pass-band ripple	0.0041
Stop-band attenuation	47.7 dB
Filter order	20

By comparison, level-crossing ADC was performed to obtain the CT digital signal (shown by the red curve) using a 5-bit quantizer. This CTDA signal was then passed through the same 20th-order FIR low-pass filter but in the CT DSP manner to get the CT digital output (shown by the blue curve). As can be seen, the resulting waveform after CT filtering to a great extent resembles the shape of the conventional DSP output, just with high-frequency noises out of the frequency band of interest, which will be removed by the analog low-pass filter after the CT

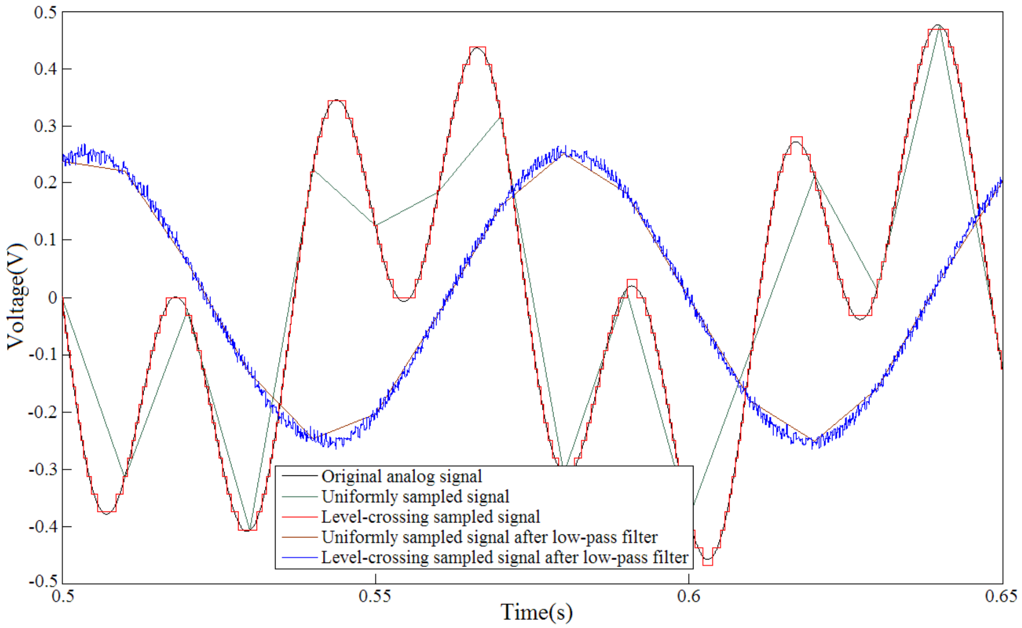

Fig. 4.4 Time-domain waveforms of a CT FIR low-pass filter: the black curve represents the original analog input; the green curve represents the digital signal obtained from conventional uniform sampling; the red curve represents the CT digital signal obtained from level-crossing ADC; the brown curve represents the digital signal processed using the conventional uniform approach; the blue curve represents the CT digital signal processed using CT DSP.

DAC. The same consistent result was also observed from the spectra in the frequency domain: both the brown and blue curves have the second peak at 41Hz removed while preserving the first peak at 13Hz.

Since Matlab, which is a computer software based on computations of arrays and matrices, can only deal with discrete signals, all CT signals mentioned above are actually discrete-time with a sampling frequency as high as 1MHz, so they effectively behave like CT signals without much influence on the results.

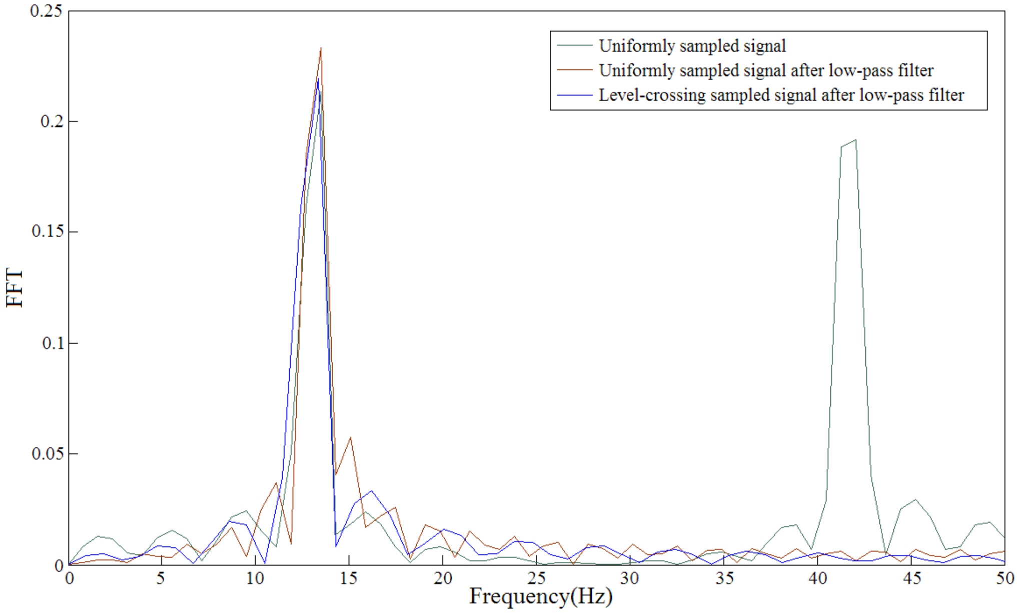

Fig. 4.5 Frequency-domain spectra of a CT FIR low-pass filter: the green curve represents the digital signal obtained from conventional uniform sampling; the brown curve represents the digital signal processed using the conventional uniform approach; the blue curve represents the CT digital signal processed using CT DSP.

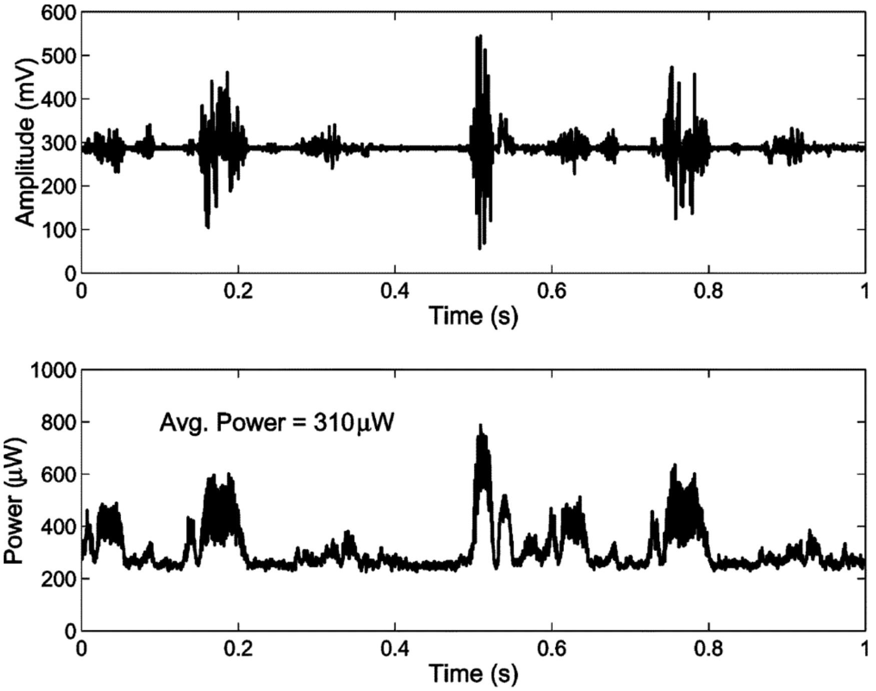

Fig. 4.6 Plot of input speech signal (top) and the instantaneous power consumption (bottom) of the CT ADC/DSP/DAC system [43].

At this stage it is still hard to quantify the amount of power reduction achievable using CT DSP, due to its different implementation as compared to conventional uniform DSP systems. It may appear that the output of a CT DSP system gets updated quite frequently, but the amount of computation required by each update is actually much smaller than that for a conventional DSP system. In a conventional FIR filter, each update in the output requires multiplication at all the filter taps, while in a CT FIR filter, every update is triggered by only one multiplier which encounters new samples. In fact, the processing power of a CT DSP system can even be further reduced by employing the technique of asynchronous delta modulation, which is able to eliminate all the multipliers by using simple adders. More details about asynchronous delta modulation will be discussed in Section 4.5. Moreover, the power consumption of a CT DSP system is strongly dependent on its input activity. For sporadically varying signals which contain long periods of silence, it is believed that CT DSP will lead to much lower power consumption. To illustrate this, the power measurement results of a real CT speech signal processing system [43] is cited in Fig. 4.6. As can be seen, the instantaneous power corresponding to the slowly-varying portions of the input is only about one quarter of the amount corresponding to the rapidly-varying portions.

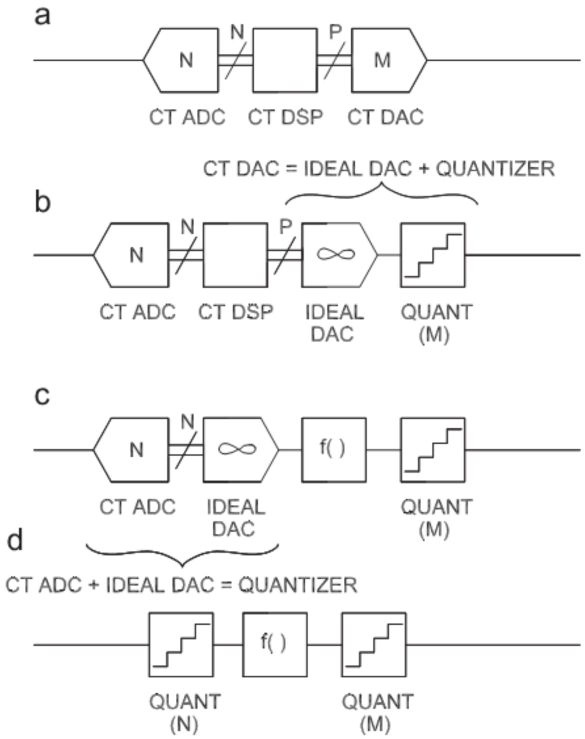

Fig. 4.7 Four input-output equivalent representations of a CT ADC-DSP-DAC. The top row shows an actual implementation of a CT ADC-DSP-DAC while the others are mathematical simplifications, resulting in a simple but equivalent system at the bottom. [44]

4.3 Quantitative Analysis of Quantization Distortion

In section 2.1, the quantization noise floor due to aliasing in conventional uniform DSP systems was explained. The immunity of CT DSP systems to aliasing and the resulting better signal quality was also briefly touched. It is obvious that without aliasing, the in-band quantization distortion of a CT DSP system is much less severe as compared to a conventional DSP system. However, it is unclear how much the signal quality improves quantitatively and how such improvement may vary as the input signal varies. In [44], an equivalent model of the CT DSP system was proposed. This model enables quantitative analysis of the in-band SDR, so that comparison of signal quality with conventional uniform DSP systems can be made.

The process of deriving the equivalent system model is illustrated in Fig. 4.7. The basic idea was to capture any nonidealities associated with both the CT ADC and the CT DAC using quantizers, the impact of which could then be analyzed mathematically. First, the M-bit DAC is decomposed into an ideal DAC with infinite resolution in series with an M-bit quantizer. Next, the ideal DAC is swapped in sequence with the CT DSP block originally preceding it. This swapping step, however, creates a problem since the DSP block which operates on digital input and digital output is now forced to operate on analog input and analog output. Therefore, an analog operator f(), which processes its input signals in the same manner, is added to replace the original DSP block. Finally, the N-bit ADC and the ideal DAC is combined into a simple N-bit quantizer. The resulted equivalent model shown in Fig. 4.7(d) then reduces the comparison with conventional DSP systems to the study of quantizers and their output spectra within the frequency band of interest.

Although the process of going through a CT quantizer is nonlinear, certain categories of inputs could still be analyzed mathematically. Using Fourier series expansion, closed-form expressions that describe the quantizer output given a symmetric sinusoidal input of arbitrary amplitude can be derived. Quantitative analysis of signal quality could then be carried out using these expressions.

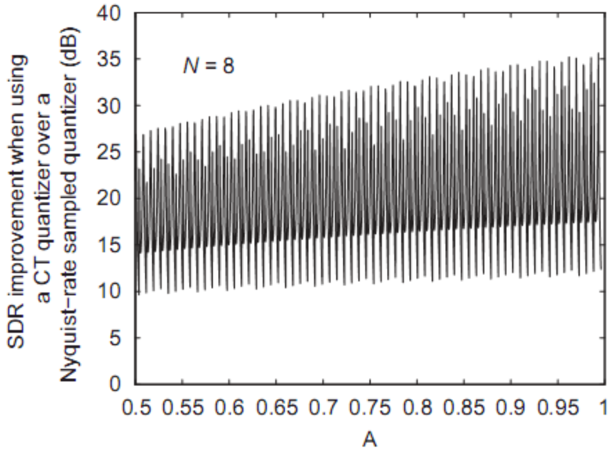

Fig. 4.8 The improvement in SDR of an 8-bit CT quantizer over a Nyquist-rate sampled quantizer acting on a 1kHz sinusoid with a bandwidth of 26kHz as a function of amplitude [44].

Fig. 4.8 shows the improvement in in-band SDR of a CT quantizer over a conventional uniformly-sampled quantizer when changing the amplitude of the input sinusoid. Rapid variations are observed in the improvement in SDR, indicating a strong dependence on the input amplitude. This can be explained by partitioning the quantization error waveform into “tip” portions and “sawtooth” portions, as illustrated in Fig. 4.9. The tip portions correspond to slowly-varying portions of the input, which, for the case of a sinusoid, are at its peaks. The sawtooth portions results from steeper portions of the input, when quantization levels are crossed more quickly. These two portions of the error waveform contribute differently to the spectrum: the tip portions contribute to low-frequency distortions that usually fall within the baseband, while the sawtooth portions result in high-frequency harmonics that usually lie beyond the baseband. Since the tip portions are very sensitive to the input amplitude, the in-band SDR of a CT quantizer experiences tremendous fluctuation as the input amplitude varies. However, an improvement of at least 10dB is still achievable for a CT quantizer over a conventional quantizer.

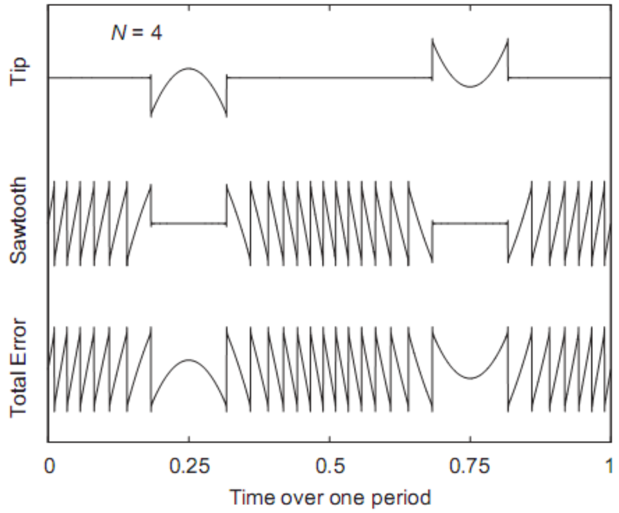

Fig. 4.9 The total error waveform is the sum of the ‘tip’ portion and the ‘sawtooth’ portion [44].

The same consistent result was also verified by computer simulation. Fig. 4.10 compares the quantization error of a CT DSP system with that of a conventional DSP system. As seen, the quantization error of a CT DSP system increases with the bandwidth to input frequency ratio, while that of a conventional DSP system stays constant. This is because a higher bandwidth will include more harmonic components in a CT DSP system, leading to a larger quantization error. In a conventional DSP system however, all the harmonic components will be aliased into the baseband, thus any increase in the bandwidth will not introduce more distortion. Nonetheless, even when the bandwidth is 1000 times higher than the input frequency (in which case 1000 harmonic components will fall within the baseband for a CT DSP system), the total in-band quantization error of a CT DSP system is still much lower than the value for a conventional DSP system.

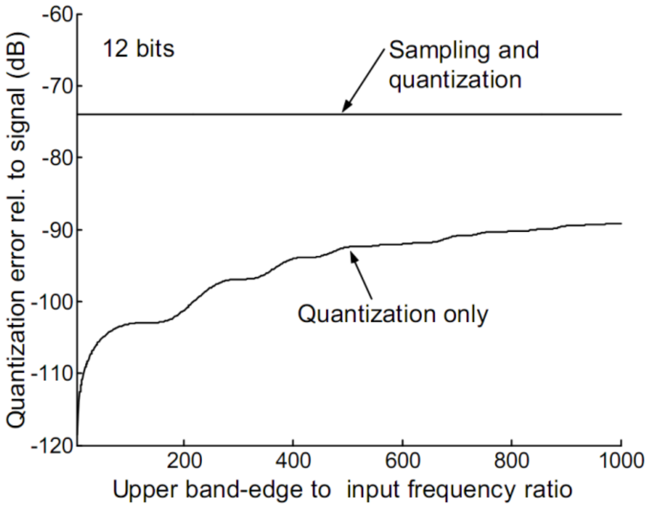

Fig. 4.10 Quantization power relative to signal power for a sinusoidal input, as a function of the ratio of bandwidth to input frequency. No oversampling is assumed in the case of conventional DSP [41].

4.4 Variable-Resolution Quantizing Scheme

Even though level-crossing ADCs consume less power for slow and inactive inputs as compared to conventional uniformly-sampled ADCs, they may actually generate more samples for fast inputs and therefore give rise to higher dynamic power dissipation. Take a full-scale maximum-frequency sinusoidal input as an example, a uniformly-sampled ADC only needs to take minimally two samples per cycle, which corresponds to the Nyquist rate, while a level-crossing ADC has to take 2^{N+1}-4 samples per cycle, where N is the quantizer resolution in bits [21]. This number is usually much larger than 2, especially when a high resolution is used for the level-crossing ADC. Therefore, ways to reduce the high sampling rate of a level-crossing ADC for fast inputs have always been sought for. The technique of variable-resolution (VR) quantizing was then proposed to alleviate this problem [45]. By adapting the quantizer resolution according to signal activity, significant dynamic power saving is achievable without in-band performance degradation.

In addition to reducing the number of samples generated for fast inputs, the VR quantizing scheme also yields a number of indirect advantages. In level-crossing sampling, the minimum time-spacing between consecutive samples is defined as the granular time T_{GRAN} [21]. Obviously T_{GRAN} corresponds to the steepest portion of the maximum-frequency input, where quantization levels are crossed most frequently. Now that the steep portions of fast inputs are quantized with larger steps, T_{GRAN} will become longer, relaxing the speed requirements on the hardware.

In CT FIR filters, each tap delay block is implemented with a chain of delay cells instead of a simple register as in conventional FIR filters. Each delay cell must have a delay of less than or equal to T_{GRAN}, and the number of such delay cells needed in each tap delay block is equal to τ/T_{GRAN}, where τ is the tap delay value used in designing the desired frequency response. Since T_{GRAN} has increased, the number of delay cells can be reduced, leading to a reduction in chip area, as well as further reduction of both static and dynamic power dissipation due to the reduced number of delay cells. Considering the total number of delay cells used to build a CT FIR filter, which usually goes to tens of thousands, the area and power reductions may indeed be substantial.

The basic idea of VR quantizing is to adjust the resolution according to the input activity: slowly varying portions of the input are quantized with maximum precision to best track any small changes, whereas fast deviations are tracked more efficiently with larger steps, reducing the number of samples generated. However, it is of vital importance to ensure that such lowering of the resolution does not degrade the quality of the system performance.

In section 4.3, the quantization distortion was studied by partitioning the error waveform into “tip” portions and “sawtooth” portions. These two portions affect different frequency ranges of the signal. Having realized this difference, it is safe to conclude that fast portions of the input, which contribute out-of-band harmonics, can be quantized with a lower resolution without degrading the in-band performance. This is valid since although the total mean square error of the signal would increase due to a larger high-frequency error, the in-band SDR would still remain unaffected.

An example of VR quantizing was cited in Fig. 4.11. Among other choices, a symmetric reduced-resolution transfer characteristic achieved by skipping an even number of steps was chosen. This is a choice with good reasons: first, only by

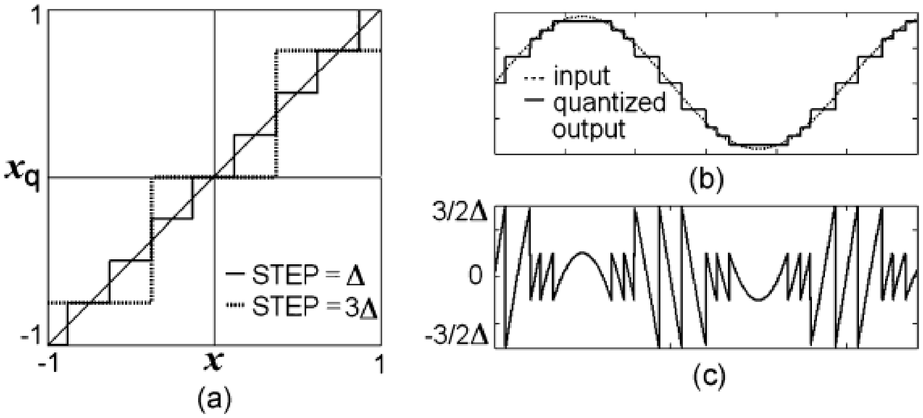

Fig. 4.11 VR quantizing achieved by skipping two steps. (a) VR transfer characteristic. (b) Example output. (c) Resulting symmetric quantization error. [45]

skipping an even number of steps is it possible to preserve the symmetry of the transfer characteristic, which ensures a zero-mean quantization error that does not increase the in-band distortion; second, no additional reference thresholds or output levels are needed in this scheme when the ADC operates in low-resolution mode, adding little to the hardware overhead.

Formulas for determining the slope thresholds could be derived as

$$S_{threshold} = (10^m f_{BW})(k\Delta), \qquad (4\text{-}6)$$

where f_{BW} is the bandwidth of the input, k is the ratio of the step size in low-resolution mode to the minimum step size in maximum-resolution mode, which can only take odd integer values like 3, 5, 7…, and Δ is the minimum step size. The $S_{threshold}$ obtained using this formula allows the minimum sawtooth frequency to be kept m decades away from the band of interest.

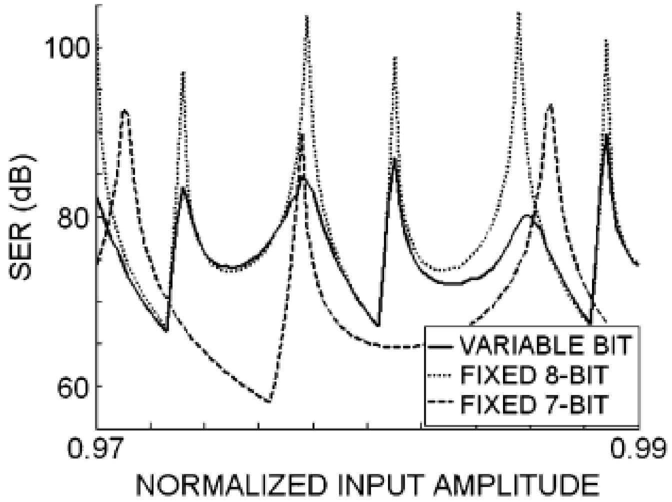

Fig. 4.12 SDR for a small range of amplitudes of a 400-Hz sinusoid for a maximum 8-b VR, an 8-b FR quantizer, and a 7-b FR quantizer. SER is calculated in a 3.6-kHz voice bandwidth, with quantizer slope threshold selected to keep the sawtooth frequency a decade away from the band of interest. [45]

With the above slope threshold selection criterion and the VR quantizing scheme shown in Fig. 4.11, Matlab simulations were performed to evaluate the performance of a CT DSP system with VR quantizing. It is shown in Fig. 4.12 that, for a maximum 8-bit VR quantizer, although the peak SDR values are not as high as those of an 8-bit fixed-resolution (FR) quantizer, the worst-case SDR values corresponding to the valleys are practically the same. Since the performance is always characterized by the worst-case scenarios, the lowering of the peak SDR values is not considered degradation of the system performance.

4.5 Asynchronous Delta Modulation

One distinct feature of level-crossing sampling is that every time a new sample is generated, its value always differs from the previous sample by 1 least significant bit (LSB). Specifically, there could only be two cases: either the new sample is one LSB larger than the previous sample, or it is one LSB smaller. Therefore, instead of transmitting the complete binary word for each sample, one could simply transmit two bits indicating whether the signal has increased, decreased, or remained unchanged. This leads to the technique of asynchronous delta modulation [46], as illustrated in Fig. 4.2(c). For subsequent DSP, a simple up/down counter can be used to reconstruct the complete binary word. The architecture of a CT FIR filter implemented using asynchronous delta modulation is shown in Fig. 4.13. As compared to the original design shown in Fig. 4.3, the number of delay lines has reduced from *N* to only two, where *N* is the quantizer resolution. Considering the huge number of delay cells used to build each delay block, the use of asynchronous delta modulation indeed provides tremendous power and area savings.

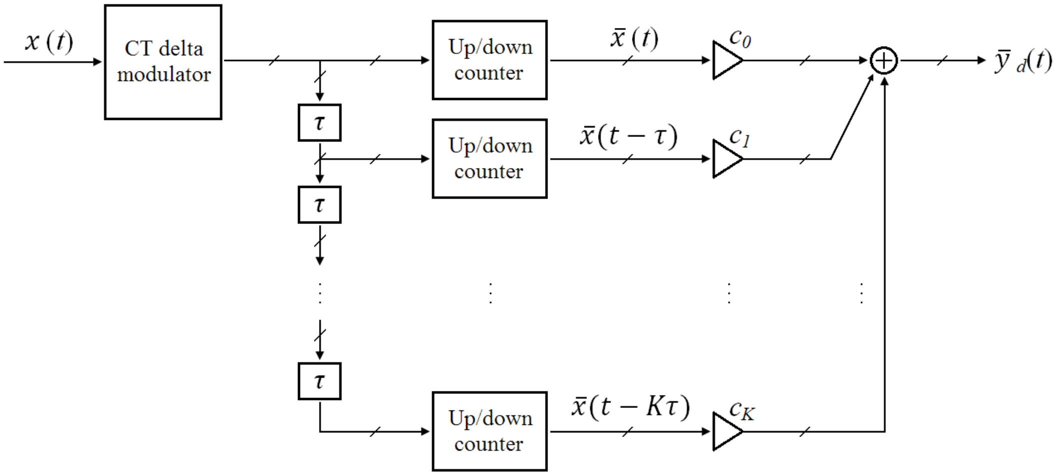

Fig. 4.13 Architecture of a CT FIR filter implemented using asynchronous delta modulation.

In fact, the above design may even be further simplified by combining the up/down counters with the coefficient multipliers: instead of counting up or down every time by 1 LSB, the counter can be modified to count up or down by a step equal to the filter coefficient. This way, all multipliers can be eliminated, further reducing power and area. The architecture of the improved design is shown in Fig. 4.14.

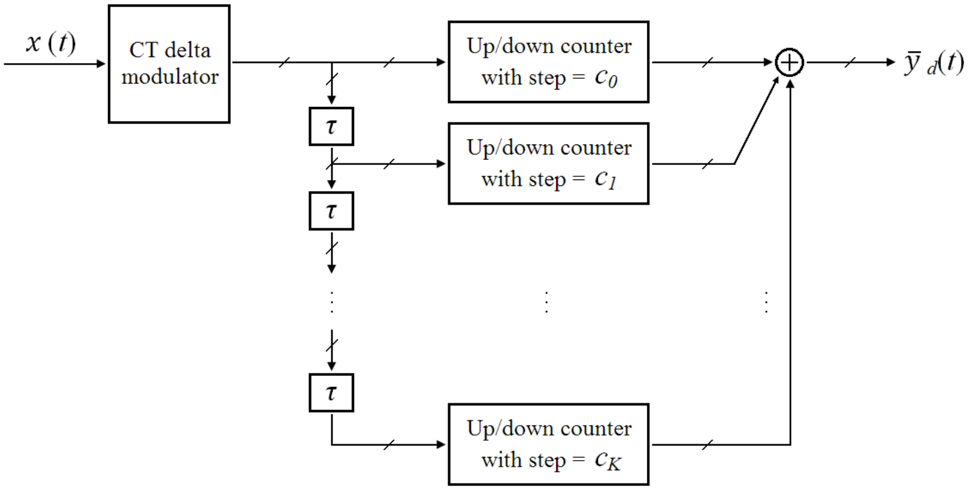
$x(t)$
CT delta modulator
τ
τ
τ
Up/down counter with step = c_0
Up/down counter with step = c_1
Up/down counter with step = c_K
$\bar{y}_d(t)$

Fig. 4.14 Improved architecture of a CT FIR filter implemented using asynchronous delta modulation.

4.6 Benefits and Drawbacks of CT DSP Systems

Event-based sampling, with level-crossing sampling being one typical example, originated a long time ago in designing control systems, where it took a long time to mature due to the lack of systematic design approaches. Its application in DSP systems is much more recent [47]. Having its own benefits and drawbacks, it is believed that CT DSP employing level-crossing sampling will also need time and effort to become mature for industrial and commercial use.

In summary, by being continuous in time and discrete in amplitude, CT DSP systems combine both advantages of conventional digital systems and conventional analog systems. On one hand, quantized amplitude allows signal processing that involves only ‘0’s and ‘1’s, giving rise to the benefit of noise immunity and programmability. On the other hand, it does not suffer from aliasing by being continuous in time. Signal dependent sampling rate results in power saving for slowly varying and inactive portions of the input. The nature of level-crossing sampling also allows immediate response to sudden changes in the input signal, and enables the use of asynchronous delta modulation, which leads to even further power and area savings.

Even though CT DSP systems sound very attractive with all the benefits described above, they also suffer from many drawbacks. First, due to the absence of a clock, asynchronous design techniques need to be used for the design of CT DSP systems. However, unlike conventional asynchronous digital systems, which only seek to preserve the relative ordering of samples, CT DSP systems require the exact time spacing between consecutive samples to be preserved as well. This is a very challenging task as it implies every part of the entire system must have signal-independent delay.

In addition, the huge number of delay cells needed to realize FIR filters is one serious drawback of CT DSP systems, for it is too costly in terms of power and chip area. Besides, process variation makes the exact amount of time delay caused by each delay cell very hard to control, unless special techniques like delay-locked-loops are used, but that will inevitably make the system even more complicated. Therefore, ways to improve the design of tap delay blocks are still in bad need.

Moreover, unlike uniformly-sampled ADCs used in conventional DSP systems, level-crossing ADCs need to track the analog input in real time, and respond immediately once a quantization level is crossed. Therefore, the static power consumption of a level-crossing ADC is expected to be much higher than that of a conventional ADC.

Despite all these drawbacks, the fact that CT DSP systems work in continuous time makes it impossible for the output to be directly stored, as it would require infinite memory to store an infinite number of data points, unless they are resampled and digitized in the conventional way. Timing quantization can be an alternative: by quantizing the time spacing between consecutive samples, the timing information can be stored together with the amplitude of the signals. But not only will this approach require huge amount of additional data storage if high accuracies are needed, but more importantly, any attempt to quantize time will destroy the benefits of aliasing-free processing and alter CT DSP systems into conventional sampled system [21]. This serious drawback limits CT DSP to real-time processing applications only. To overcome this problem, a novel timing storage circuit will be introduced and discussed in CHAPTER 5.

CHAPTER 5

MEMRISTOR-BASED TIMING STORAGE CIRCUIT

5.1 The Memory Effect of Memristors

The resistor, the capacitor and the inductor have been known as the three fundamental passive circuit elements for a long time. In 1971, Professor Leon Chua from the University of California at Berkeley reasoned from symmetry that there should be a missing fourth fundamental circuit element, which he named it "memristor" [48].

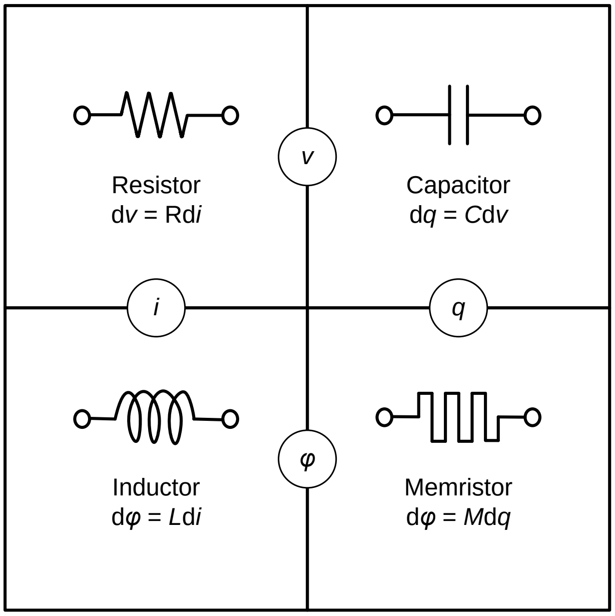

Fig. 5.1 The four fundamental passive circuit elements and four fundamental circuit variables

It was noted that there are six different mathematical relations connecting pairs of the four fundamental circuit variables: current i, voltage v, charge q and magnetic flux φ, as shown in Fig. 5.1. Among these six relations, two are determined by fundamental physics laws or definitions. First, charge is defined as the time integral of current, which if written in differential form is

$$i = \frac{\mathrm{d}q}{\mathrm{d}t}. \tag{5-1}$$

Faraday's law of induction states that voltage is the time rate of change of magnetic flux, which gives

$$v = \frac{\mathrm{d}\varphi}{\mathrm{d}t}. \tag{5-2}$$

The remaining four relations describe characteristics of the four fundamental passive circuit elements. Resistance is the rate of change in voltage with respect to current:

$$R = \frac{\mathrm{d}v}{\mathrm{d}i}. \tag{5-3}$$

Capacitance is the rate of change in charge with respect to voltage:

$$C = \frac{\mathrm{d}q}{\mathrm{d}v}. \tag{5-4}$$

Inductance is the rate of change in magnetic flux with respect to current:

$$L = \frac{\mathrm{d}\varphi}{\mathrm{d}i}. \tag{5-5}$$

And similarly, memristance is the rate of change in magnetic flux with respect to charge:

$$M = \frac{\mathrm{d}\varphi}{\mathrm{d}q}. \tag{5-6}$$

Therefore, memristance can be interpreted as the slope at an operating point on the *φ* – *q* curve [48], as illustrated in Fig. 5.2.

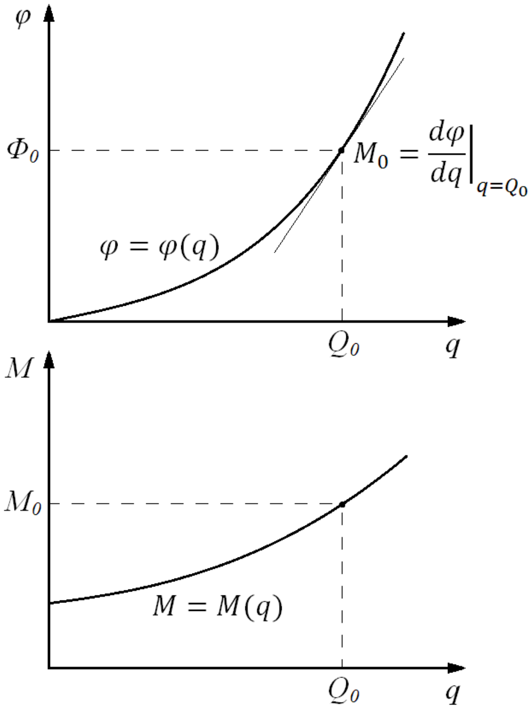

Fig. 5.2 Plots of a *φ* – *q* curve and its corresponding *M* – *q* characteristic.

If both the numerator and denominator of the right hand side of (5-6) are divided by d*t*, and substitute (5-1) and (5-2) into the result, the following relation can be obtained:

$$M = \frac{\mathrm{d}\varphi}{\mathrm{d}q} = \frac{\frac{\mathrm{d}\varphi}{\mathrm{d}t}}{\frac{\mathrm{d}q}{\mathrm{d}t}} = \frac{v}{i}. \tag{5-7}$$

From (5-7) it is realized that a memristor relates its voltage and current in the same way as a resistor does, but only at fixed operating points. This is because memristance is a function of charge rather than current. In the case of linear elements, which mean a memristor with a linear $\varphi - q$ characteristic and a resistor with a linear $v - i$ characteristic, the memristance and resistance will be constant regardless of the operating points. In that case, the memristor is identical to the resistor. However, if the $\varphi - q$ characteristic of a memristor is nonlinear, like what is shown in Fig. 5.2, then M itself will be a function of q, leading to a more interesting scenario.

With memristance M being a function of charge q, the instantaneous resistance (resistance at a particular operating point) of a memristor seen by external circuit components will be depending on the net amount of charge that has flown through it. As shown in Fig. 5.3, when current flows through the memristor in one direction, the memristance starts to decrease, and the larger is the current magnitude, the faster will the memristance be dropping. When current flows in the opposite direction, the memristance starts to increase; when current is stopped, the memristor retains the memristance value it last had until current starts to flow again. This nonvolatile nature of memristors makes them an attractive candidate for the next generation memory technology, in fact not only for two-state digital memories, but also for multi-state or even continuous-value memories [49-53]. This is also the reason why Chua named this device "memristor", which stands for memory resistor, as the instantaneous resistance of a memristor not only depends on the present state, but also depends on the history of charge movements through the device.

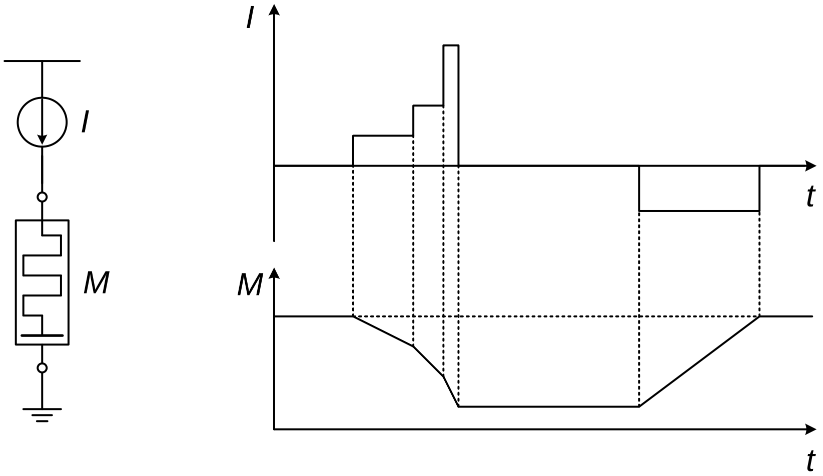

Fig. 5.3 Change in memristance under different current levels and directions.

Although the concept of memristor proposed by Chua sounds to be an interesting circuit element with attractive features for many different circuit applications, no real examples of memristors have been found in nearly forty years until in 2008, a team at Hewlett-Packard Laboratory announced their development of the world's first real memristor based on thin-film devices [54].

The structure of the memristor implemented by HP is shown in Fig. 5.4. The device is essentially a thin film of titanium dioxide (TiO_2) of thickness D sandwiched between two metal contacts. The film is not uniformly doped along its thickness: only a region at the left side with thickness w is doped while the rest is intrinsic. The doped region has a certain percentage of its oxygen missing ($TiO_{2\text{-}x}$), with x depending on the doping concentration. These oxygen vacancies make the doped region more conductive than the intrinsic region. The total resistance of the film is then equal to the resistance of the doped region in series with the resistance of the intrinsic region. If an external bias is applied across the device, the positively-charged oxygen vacancies will drift in the direction of the electric field, causing the boundary between the two regions to move [25]. The total resistance of the device will then be altered.

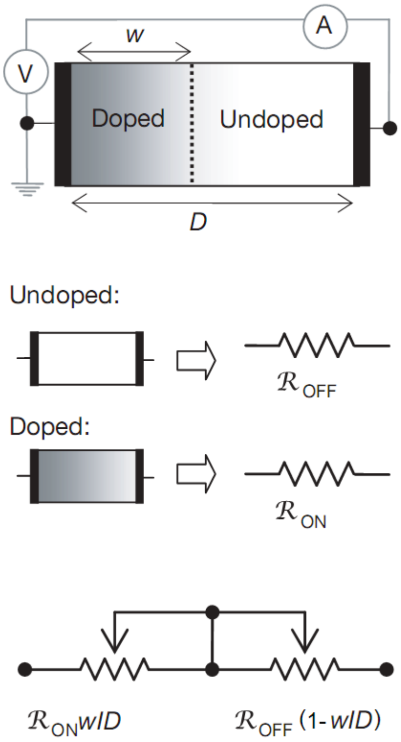

Fig. 5.4 Structure of the HP memristor and the coupled variable resistor model [22].

Based on the understanding of the device mechanism, HP proposed a coupled variable resistor model that allows analysis of changes in memristance as a result of charge movements quantitatively. For the simplest case of ohmic conduction, the following equation is obtained:

$$v(t) = \left(R_{ON}\frac{w(t)}{D} + R_{OFF}\left(1 - \frac{w(t)}{D}\right)\right)i(t), \tag{5-8}$$

where *v*(*t*) is the instantaneous voltage across the device and *i*(*t*) is the instantaneous current through it; *w*(*t*) is the instantaneous width of the doped region; R_{ON} and R_{OFF} are the resistance values corresponding to *w* being *D* and 0 respectively. If linear ionic drift in a uniform field with average ion mobility μ_v is assumed, the following simple differential equation can be derived:

$$\frac{dw(t)}{dt} = \mu_v \frac{R_{ON}}{D} i(t). \tag{5-9}$$

Substitute (5-1) into (5-9), the following is obtained:

$$w(t) = \mu_v \frac{R_{ON}}{D} q(t). \tag{5-10}$$

By substituting (*5-10*) into (5-8), the memristance of the device can be obtained as a function of charge *q*:

$$M(q) = R_{OFF} - (R_{OFF} - R_{ON})\frac{\mu_v R_{ON}}{D^2} q. \tag{5-11}$$

The charge dependence in the above expression leads to the memristive behavior of the device.

5.2 Sandglass Analogy

By taking advantage of the ability of memristors to store continuous values, a novel timing storage approach is proposed. The basic idea is to track the amount of time elapsed using an electronic implementation of sandglasses. How a real sandglass can be used to track time is first illustrated in Fig. 5.5(a) to facilitate understanding of the proposed approach using memristors. Initially, all the sand is in chamber β at the bottom. At instant t_1, event X occurs; the sandglass is flipped over and the sand starts to trickle down from chamber β to chamber α. At instant t_2, event Y occurs; the sandglass is suspended horizontally to prevent any further movement of the sand between the two chambers. The time duration T_1 between event X and event Y is then stored in this sandglass. Upon reproduction, one can simply turn the sandglass in the opposite direction to allow the sand in chamber α to trickle back to chamber β. The same amount of time T_2 will be needed for all the sand to return to its original chamber. This is a quite accurate method to

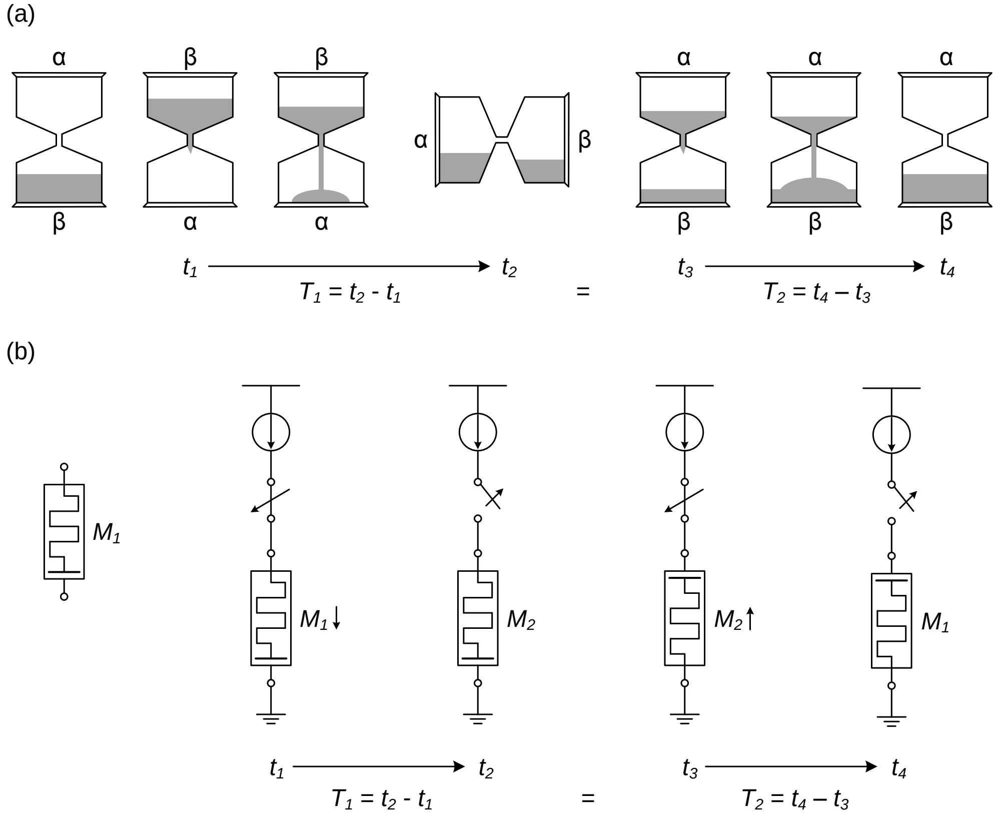

Fig. 5.5 (a) Illustration of time tracking using a sandglass. (b) Illustration of time tracking using a memristor.

record timing, due to the careful design of the sandglass: the opening between the two chambers is so small that regardless of the depth of sand in the upper chamber, the rate of sand flow is always constant (like velocity saturation of electrons in a metal-oxide-semiconductor (MOS) transistor). This same constant flow rate when positioned in either direction guarantees it will take the same amount of time for the same amount of sand flow, whether from chamber α to chamber β or from chamber β to chamber α.

Similarly, a memristor can be employed to achieve the same function electronically. As shown in Fig. 5.5(b), the initial memristance of a memristor is M_1. At instant t_1, event X occurs; a constant current source with current I is connected to the memristor in series, causing the memristance to decrease. At instant t_2, event Y occurs; the constant current source is disconnected. The memristance then stays constant at M_2. The difference between M_1 and M_2 reflects the time duration between event X and event Y. When reproduction is required, the memristor can simply be reconnected to the same constant current source, but in the opposite orientation. The memristance then starts to increase, since the net amount of charge that has passed through the memristor is now decreasing due to the reversed current direction. The same amount of time between events X and Y will then be needed for the memristance to increase to its original value M_1. This is because for the original memristance value to be restored, the net amount of charge that has passed through the memristor must be zero. In other words, the same amount of charge passed through the device in the recording phase must be passed through the device in the reproducing phase as well. Since the current is the same, the time should also be the same.

5.3 Memristor Models

In order to design memristor-based circuits, a compact memristor model that is able to simulate its charge-dependent memristance characteristic is needed. In my design, a SPICE behavioral model proposed by Mahvash et al [55] was used. This model tracks the amount of charge flow using an ideal capacitor, and uses the voltage drop across this capacitor together with the branch current to control the output of a dependent voltage source, which eventually defines the current-voltage relation of this two-terminal device. The *M-q* characteristic of this model was developed based on the coupled variable resistor model shown in Fig. 5.4.

This behavioral model simulates the memory effect of a memristor elegantly using a simple ideal capacitor. It allows me to simulate large-scale memristor circuits with much better time-efficiency compared to other more complicated models. However, most of the available memristor models including this one I’ve chosen have convergence problems due to the non-Markovian nature of memristors, which create a lot of troubles when performing transient simulations. This problem has to be dealt with by carefully introducing all kinds of small non-idealities and mismatches in the circuits, but they have to be small enough so that circuit behaviors will not be altered significantly.

5.4 Circuit Implementation

All the circuits shown in this paper were designed using a 0.35-µm CMOS technology, and they work under a 3.3-V DC power supply. As mentioned in Section 4.1, CT DSP systems operate without clock synchronization. As a result, asynchronous design techniques have to be used for the entire system, which means every circuit block are to be custom designed with little help available from digital synthesis tools. In addition, unlike conventional asynchronous digital systems, which only seek to preserve the relative ordering of samples, CT DSP systems require the exact time spacing between consecutive samples to be preserved as well. Therefore, extra caution needs to be exercised to ensure constant delay along signal paths.

In Fig. 5.5(b), a constant current source is used to change the memristance value of a memristor. Although a constant voltage source will also generate current causing the memristance value to change, a constant current source is still a better option for the following reason: when a constant current I is passing through the memristor, the accumulated amount of charge that has passed through the device can be expressed as

$$q(t) = It. \qquad (5\text{-}12)$$

When (5-12) is substituted into (5-11), the memristance as a function of time becomes

$$M(t) = R_{OFF} - (R_{OFF} - R_{ON})\frac{\mu_V R_{ON}}{D^2} It. \tag{5-13}$$

A linear relation between time and memristance is established. In the case of a constant voltage source, during the recording phase the memristance will decrease at an increasing rate, since with the same voltage, lower memristance will lead to larger current, causing the memristance to decrease even faster. This is undesirable because it will degrade the accuracy for the recording of short time intervals. For this consideration, a constant current source is preferred for the proposed timing storage circuit.

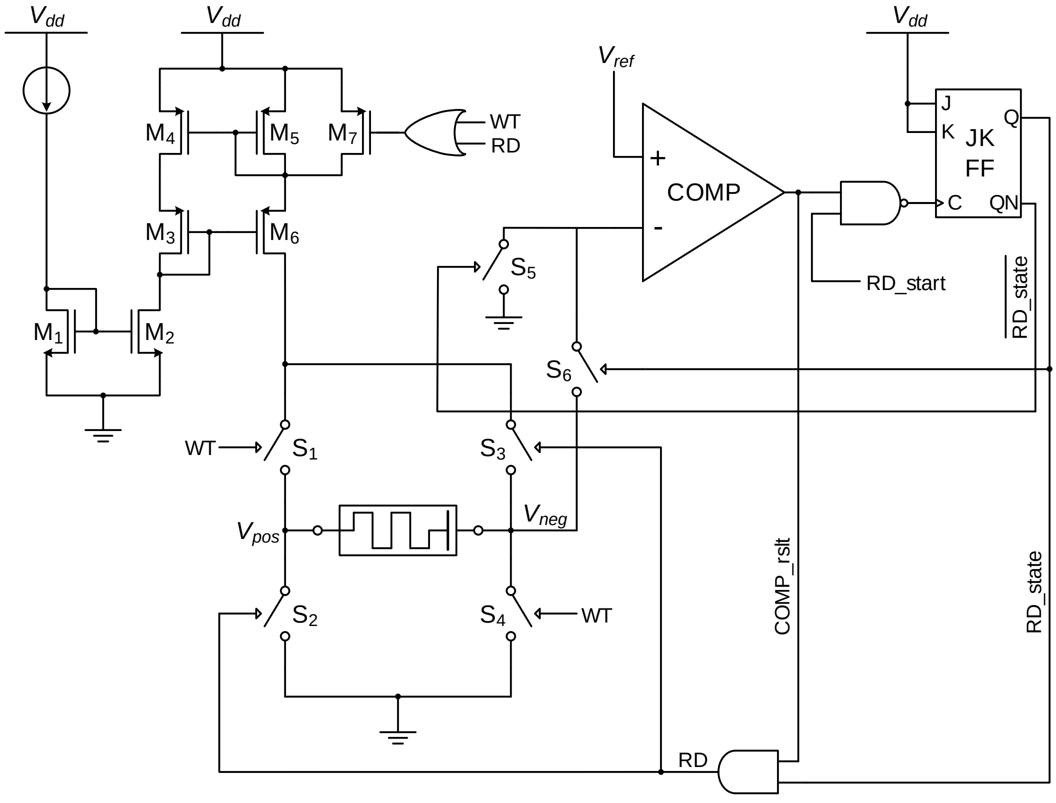

Fig. 5.6 Schematic of a single timing storage cell.

The schematic of a single timing storage cell based on a single memristor is shown in Fig. 5.6. Transistors M_1-M_6 form a Wilson current mirror [56].

Compared to conventional two-transistor simple current mirrors, this current mirror exhibits a much higher output resistance with only a small sacrifice of one threshold voltage V_{th} plus one saturation voltage V_{dsat} in the output voltage swing. This is achieved by forming a feedback loop with transistors M_3-M_6. The higher output resistance of this current mirror leads to smaller variations in the output current when the output voltage changes, which is exactly what is needed to keep a constant rate of change in the memristance value during the recording (WT) and reproducing (RD) phases. Transistor M_7 serves as a switch to turn on and off the current through the memristor. When neither "RD" nor "WT" is asserted, the output of the OR gate will be zero, tuning on transistor M_7, which pulls the drain of M_5 to V_{dd}, forcing the branch current to be zero.

The J-K flip-flop (JKFF) on the top right corner is configured as a toggle flip-flop (TFF) by keeping both its J and K inputs tied up to V_{dd}. It is used to trigger the start and finish of a reproducing event. Before use, this TFF should first be cleared, so that the "RD_state" and "$\overline{\text{RD\_state}}$" outputs become '0' and '1' respectively. This keeps switch S_5 on but switch S_6 off. The negative input of the comparator is therefore connected to ground. Since the positive input is connected to V_{ref}, a reference voltage level above zero, the comparator output "COMP_rslt" stays at '1'. "RD_start" is an external signal which informs the circuit to reproduce a previously recorded time interval by generating a very short pulse of '0', and stays at '1' during the rest of the time. This makes the output of the NAND gate '0' initially.

Fig. 5.7 shows the simulation waveforms of the circuit to record and reproduce a 30-ms time interval. At 0.03s, the recording event began, as indicated by the rise of “WT” from ‘0’ to ‘1’. This turned on switches S_1 and S_4, and at the same time, the output of the OR gate became ‘1’, turning off M_7 and a constant current was built up instantly from M_5 and M_6. This current flowed through the memristor from the anode, causing its memristance to drop at a constant rate, as reflected in the constant decrease in V_{pos}. At 0.06s, “WT” went back to ‘0’ and the recording event ended. At 0.13s, a short pulse of ‘0’ at “RD_start” created another short pulse of ‘1’ at the clock input of the TFF, causing the state to toggle. The output “RD” of the AND gate became ‘1’, turning on switches S_2 and S_3. The same constant current flowed through the memristor, only the direction was reversed. The memristance value started to rise, as can be seen from the constant increase in V_{neg}, which was then connected to the negative input of the comparator to be compared against V_{ref}. At 0.16s, “COMP_rslt” dipped down to ‘0’ as V_{neg} reached the same level of V_{ref}, and a series of changes was triggered in a very short period of time: First, the output “RD” of the AND gate became ‘0’, which stopped the constant current and disconnected the memristor by turning of S_2 and S_3. Second, the drop of “COMP_rslt” from ‘1’ to ‘0’ propagated through the

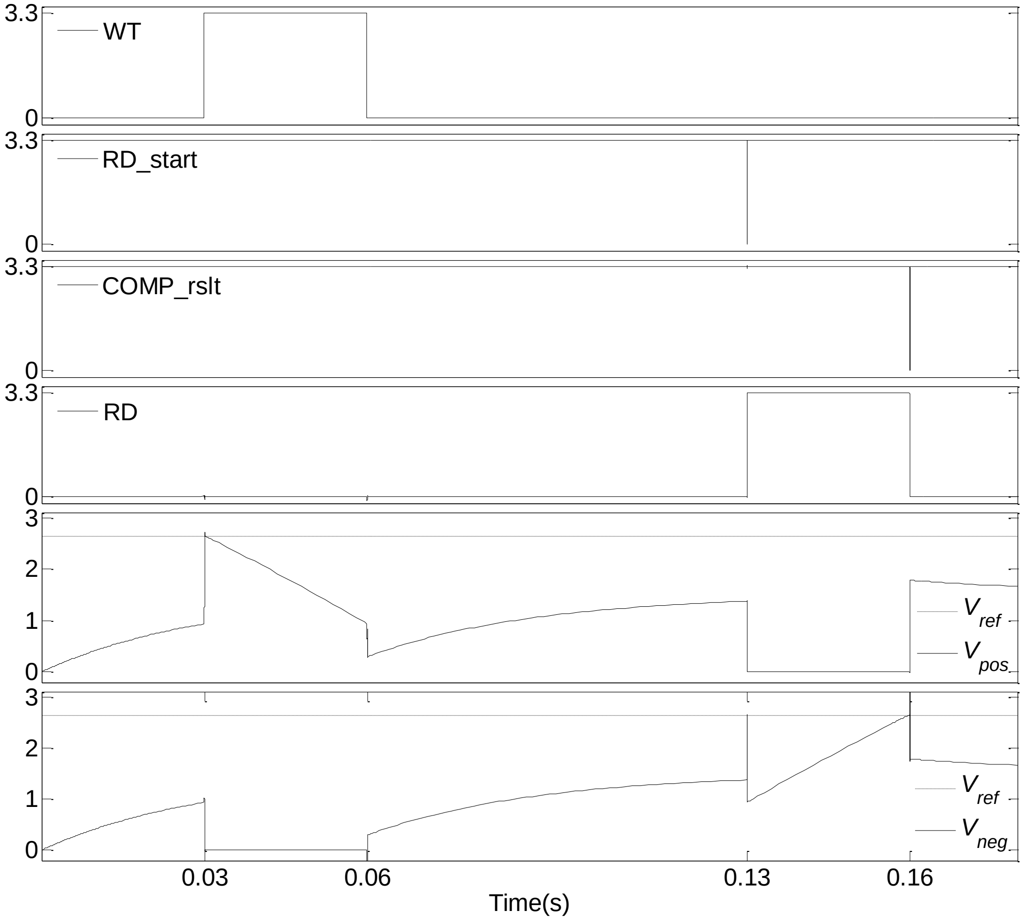

Fig. 5.7 Simulation waveforms of the single timing storage cell.

NAND gate to present another rising-edge at the clock input of the TFF, causing the state to toggle again. Although this would connect the negative input of the comparator back to ground, making “COMP_rslt” reasserted, “RD” would remain at ‘0’ given the cleared “RD_state”. At this point in time, everything was reverted to the original state, and the circuit would be ready for the next round of recording and reproducing events.

5.5 Practical Considerations

Having explained the working mechanism of the timing storage circuit, there are a few practical issues which need to be properly addressed to guarantee the robustness and accuracy of timing storage.

Table 5.1 Parameter values for memristor model

Parameter	Quantity	Value
R_{ON}	Minimum memristance	400 kΩ
R_{OFF}	Maximum memristance	98 MΩ
μ_V	Ion drift mobility	5×10^{-14} m^2/(V·s)
D	Device thickness	5 nm

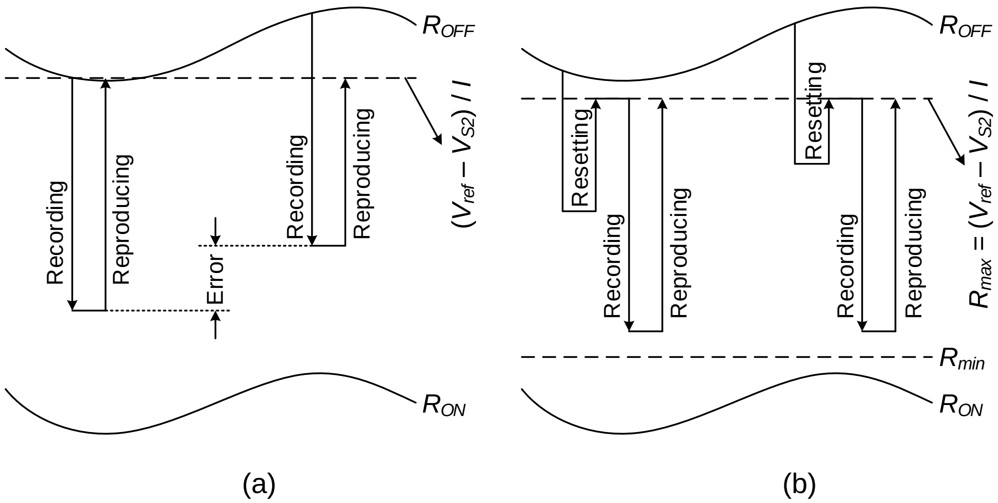

Fig. 5.8 Range of memristance values used considering process variation.

First of all, although the memristor model used for simulation has fixed parameter values summarized in Table 5.1, physical implementation would inevitably see memristors with different values due to process variation. As can be seen from (5-13), μ_V and D only control the rate of change in memristance with no influence on the absolute range. For each individual memristor, no matter how fast or how slow the memristance changes with a constant current, this change rate remains the same during the recording and reproducing phases. Therefore, variations in μ_V and D would not affect the accuracy of timing storage, although it would affect the maximum length of time intervals that can be stored in a single memristor. In my design, the Wilson current mirror produces a constant current I of 27 nA when turned on. The choice of this current magnitude was determined in such a way that the resulting maximum storable interval length (44.6 ms in our design) is just sufficient to cover the maximum expected spacing between consecutive samples (43.3 ms in our design), so that the accuracy of short interval storage can be maximized. On the other hand, variations in R_{ON} and R_{OFF} would directly affect the accuracy since the same V_{ref} is used for all the cells. As illustrated in Fig. 5.8(a), when two memristors with different R_{OFF} values are used to record the same interval, the reproduced results differ quite significantly. Had the R_{OFF} value corresponded to a V_{neg} even lower than V_{ref}, reproducing of a stored interval would take forever to complete. To resolve these problems, a usable range of memristance values between R_{min} and R_{max} needs to be chosen based on my estimations of the maximum possible variations in R_{ON} and R_{OFF}. As illustrated in Fig. 5.8(b), this usable range of memristance values must be bounded by the minimum value of R_{ON} and the maximum value of R_{OFF}. Before being used, each memristor would first be reset to the same value of R_{max}, which determines the value of V_{ref} as

$$V_{ref} = R_{max}I + V_{S2}, \tag{5-14}$$

where V_{S2} is the voltage drop across switch S_2 during the reproducing phase. The same circuit shown in Fig. 5.6 can be used to perform reset by recording an interval of T_{rst} for each memristor. After reproducing, the memristance values would stay right at R_{max} as V_{neg} reaches the level of V_{ref} defined in (5-14). If it is assumed that $R_{OFF} \gg R_{ON}$, from (5-13) it can be deduced that the reset interval length T_{rst} must satisfy

$$T_{rst} \geq \frac{{D_{max}}^2 (1 - \frac{R_{max}}{R_{OFF,max}})}{I \mu_{V,min} R_{ON,min}} \tag{5-15}$$

for every memristor to be properly reset, where D_{max} denotes the maximum device thickness and $\mu_{V,min}$ denotes the minimum ion drift mobility, while $R_{ON,min}$ and

$R_{OFF,max}$ denotes the minimum value of R_{ON} and the maximum value of R_{OFF} respectively.

Secondly, all the five switches connected to the memristor should be implemented as CMOS transmission gates instead of simple NMOS or PMOS switches to minimize charge injection. From (5-11) it is obvious that the memristance value depends on the net amount of charge that has flown through a memristor. Since a constant current that is as small as 27 nA has been chosen to save power, the memristor model was parameterized to be very sensitive to any small amount of charge movements. Therefore, any charge injection through transistor parasitic coupling during ON/OFF switching can significantly affect the memristance value and hence alter the stored timing information. To alleviate this problem, transmission gates with the same minimum-size NMOS and PMOS transistors were employed. The complementary control signal was carefully generated to have the same slope as the original control signal to fully cancel out the charge injected from the NMOS with the charge injected from the PMOS. Although not used, half-size dummy switches with both terminals shorted and controlled by an inverted switching signal can also be added to further reduce charge injection at the expense of nearly doubled switching power and area. The use of transmission gates also yields another benefit: the total equivalent on-state resistance of a transmission gate is almost independent of the output voltage, while a single transistor switch exhibits varying on-state resistance. Although the on-state resistance of a MOS transistor is very small compared to the memristance of the memristor, any variations in it would still alter the level of V_{neg} and therefore interfere with the decision making of the comparator.

Last but not least, any transistor leakage current passed through the memristor, no matter how small, will get accumulated and change the memristance over time. Therefore, instead of trying to minimize leakage using techniques such as power gating, this problem is approached from a different perspective. By arranging the four switches S_1-S_4 in a perfectly symmetric topology with respect to the memristor as shown in Fig. 5.6, it managed to cancel out the leakage through the memristor. Although non-zero off-state current does exist through these switches when the circuit is idling, no observable current flows through the memristor, keeping the stored timing information intact. The key point is to keep all four switches off during the idling state, as any switch that remained on will disturb the symmetry and cause the leakage cancelation to fail.

5.6 Comparator Design

The comparator is an important block in this timing storage circuit for two reasons: first, it is responsible for detecting the instant when V_{neg} crosses over V_{ref} during the reproducing phase and hence directly affects the accuracy. However, this does not mean an infinite resolution would be needed to achieve perfect reproduction. For example, if the minimum resolvable input difference of the comparator is δ, then by the time the comparator detects the crossing V_{neg} should have already reached the level of $V_{ref} + \delta$. This will not necessarily cause any error if the amount of overshoot is consistent in every recording/reproducing cycle. To be specific, if the memristor always starts and stops at the same memristance value of $R_{max} + \delta/I$, it should also take the circuit the same amount of time to traverse this extra amount during the recording and reproducing phases. Second, due to the static biasing current of the analog circuits, the comparator is expected to be a major power consumer within the entire system. Therefore, an ultra-low-power design with reasonably high resolution is needed to realize the crossing detection functionality while keeping the total power consumption low.

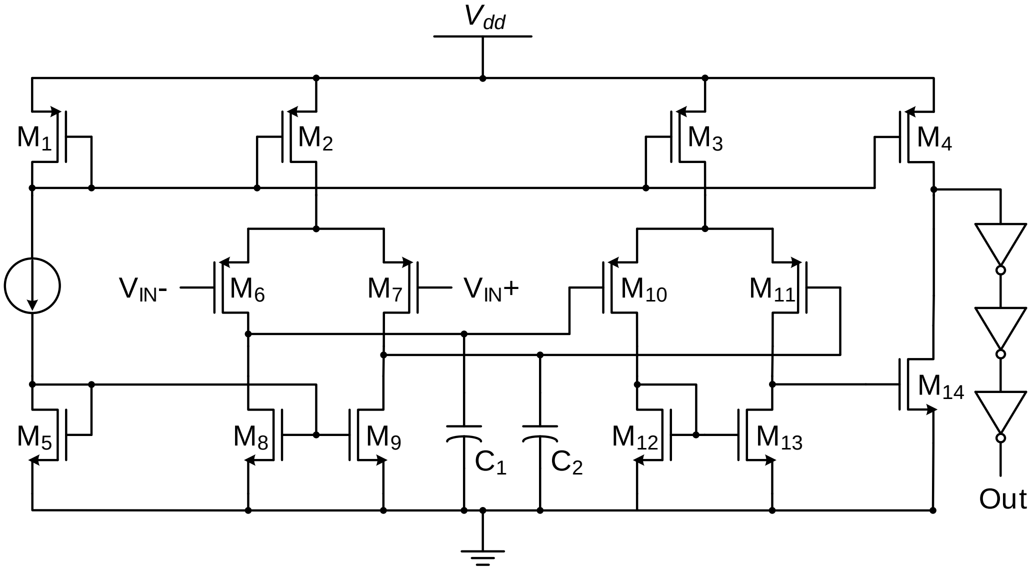

Fig. 5.9 Schematic of the comparator.

The schematic of the designed comparator is shown in Fig. 5.9. No regenerative latch is used since the comparator needs to compare varying inputs in continuous time, which makes it impossible to predict the right time for latch reset. As an alternative, a three-stage amplifier is designed to boost the voltage gain. Stability is not a concern as this amplifier will never operate in closed-loop, which avoids the need of phase margin compensation. Transistors M_1-M_4 form a simple current mirror to bias the three stages: The first stage is a fully differential amplifier consisting of transistors M_6-M_9. The second stage is a single-ended differential pair made up of transistors M_{10}-M_{13}. M_{14} is connected as a common-source amplifier and forms the third stage. All transistors operate in sub-threshold region to save power. The open-loop gain is 91dB and the static power consumption is only 112 nW. The high voltage gain not only ensures the comparator's capability of resolving very small differences in the inputs, but also suppresses the kick-back noise from the inverters at the output. This is very important as any spikes appearing at the comparator negative input will be fed into the memristor through switch S_6 and alter the stored timing information. To avoid this undesirable effect, two 100-fF capacitors are inserted between the first and second stages to block the kick-back noise by shunting them to ground. This will lower the bandwidth of the amplifier but for my application the input varies at a constant slope of only 57.18 V/s, which is the same as the maximum slope of a 9.1-Hz sinusoid. Bandwidth is thus not a problem in handling such low frequency inputs. At the output of the comparator, a short inverter chain is added. On one hand, this helps to pull the amplifier output to full V_{dd} or ground; on the other hand, they are sized properly to serve as a driving buffer for the following stage. When multiple memristors are used to build a large timing storage circuit, the shared comparator will need to drive multiple gates and therefore a large fan-out will be necessary. Lastly, this comparator has an input offset of 170 µV, but as long as this value is consistent, it will not affect the accuracy for the same reason why a finite resolution is acceptable.

CHAPTER 6

MEMRISTOR-BASED CT DIGITAL FILTERS

6.1 Recording and Reproducing CT Digital Signals

6.1.1 Timing Storage Cell Integration

To record and reproduce a CT digital signal over a certain period of time, the same number of memristors as the number of level-crossing samples generated within this period will be required. Having designed the timing storage cell based on a single memristor, the integration of multiple cells and their selection control are discussed in this section.

Fig. 6.1 shows an example of a four-cell timing storage circuit. To save power and chip area, the current mirror and comparator are shared among the four cells. Upon startup, the two JKFFs are cleared to make both “WT_state” and “RD_state” ‘0’ so the current mirror remains off. The inverse “$\overline{\text{RD\_state}}$” keeps “Vin-” grounded so the comparator output “COMP_rslt” stays at ‘1’. A two-bit ring counter “CNT_WT” is used to select the memristor cell for recording. At the beginning, all four outputs of “CNT_WT” are ‘0’ as the counter is disabled, although internally it has an initial count of ‘11’. Another same counter “CNT_RD” with initial count ‘00’ is used to select the memristor cell for reproducing. It is very important to disable the two counters when the system is idling because otherwise

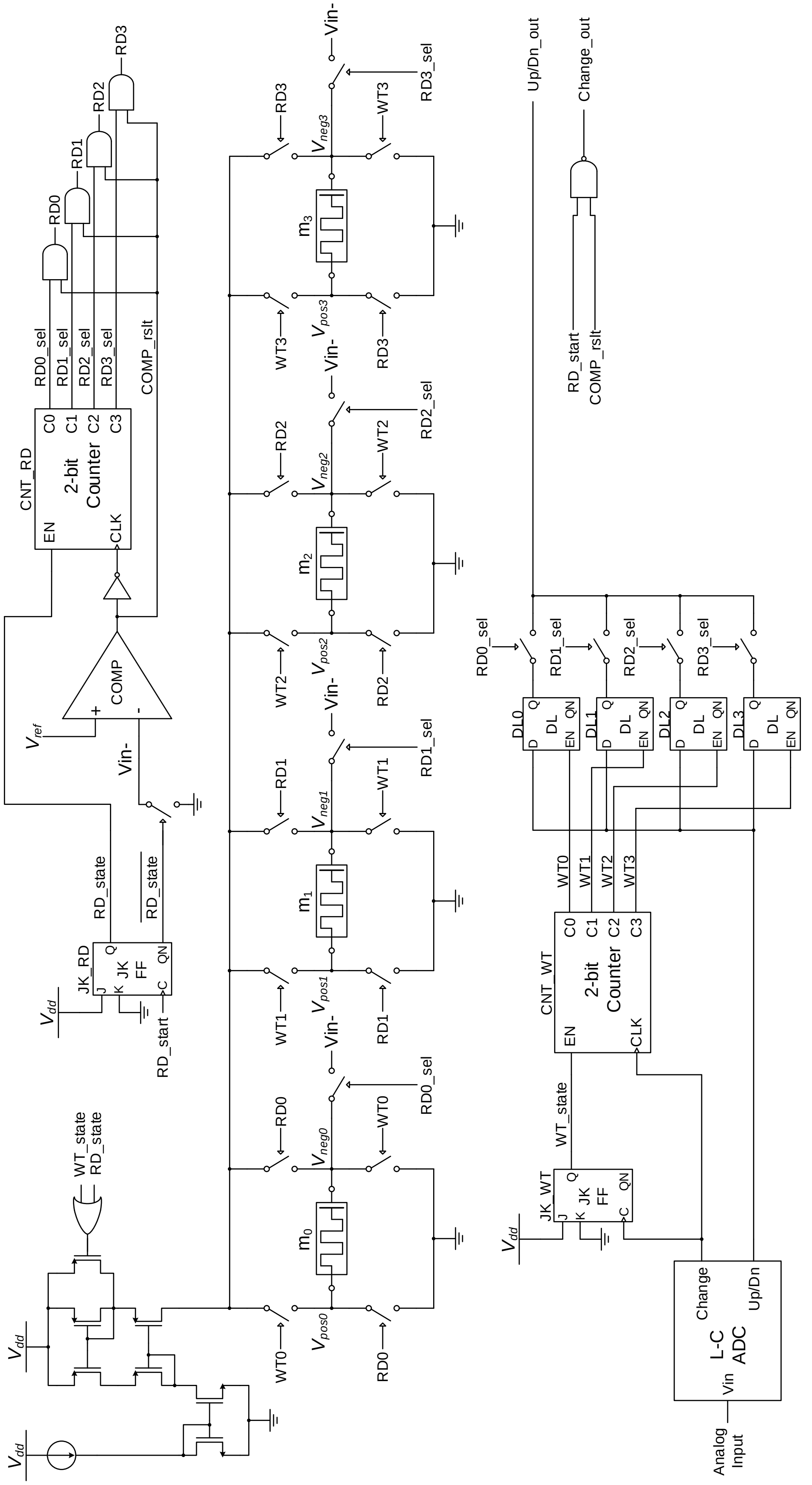

Fig. 6.1 Schematic of a four-cell timing storage circuit to record and reproduce CT digital signals.

the leakage current will alter the memristance values for reasons mentioned previously in Section 5.5.

The two output bits “Change” and “Up/Dn” of the level-crossing ADC are the signals to be delayed in CT digital filters. The circuit starts recording when the first pulse is received from the “Change” bit of the level-crossing ADC. This pulse sets “JK_WT” and the asserted “WT_state” turns on the current mirror. At the same time, this pulse also triggers “CNT_WT” to count to ‘00’, and being enabled by “WT_state”, “WT0” becomes ‘1’ to allow the constant current to pass through memristor m_0. When the second pulse arrives, the counter counts up to ‘01’, causing “WT0” to become ‘0’ and “WT1” to become ‘1’. As a result, memristor m_1 is connected to the current mirror while m_0 is disconnected. The time interval between the first two pulses has then been stored in m_0 while m_1 proceeds to record the second interval. This process continues until a reset pulse arrives to clear “JK_WT”, making “WT_state” back to ‘0’, disabling “CNT_WT” and turning off the current mirror, which marks the end of a recording session.

Reproducing of the stored CT digital signal will start once a pulse of ‘0’ is received from the “RD_start” input. This pulse propagates through the NAND gate to give the first pulse of ‘1’ at “Change_out” – an output that gives the reproduced “Change” bit of the level-crossing ADC. On the other hand, the “RD_start” pulse also sets “JK_RD” and the asserted “RD_state” turns the current mirror back on. Meanwhile, counter “CNT_RD” is enabled so “RD0_sel” becomes ‘1’. The output “RD0” of the first AND gate then also becomes ‘1’ since “COMP_rslt” has been ‘1’ since the beginning. The constant current is therefore directed to pass through memristor m_0 from its cathode, and V_{neg0} is connected to the negative input "Vin-" of the comparator to be compared against V_{ref}. When V_{neg0} crosses over V_{ref}, "COMP_rslt" dips down and counter "CNT_RD" counts up to make "RD0_sel" '0' and "RD1_sel" '1'. "RD0" then also turns '0' to disconnect memristor m_0. As V_{neg1} is connected to "Vin-", "COMP_rslt" goes back to '1', causing "RD1" to become '1' which directs the constant current to flow through memristor m_1. At the same time, the short pulse of '0' at "COMP_rslt" during this transition propagates through the NAND gate to give the second pulse of '1' at "Change_out". This process continues until counter CNT1 reaches the same count as counter CNT2, which corresponds to the index of the memristor that stores the last time interval. A reset pulse will then be generated to clear "JK_RD", "CNT_RD" and "CNT_WT" and bring the circuit back to its original state.

The same method can be used to record and reproduce changes in the "Up/Dn" bit, but a better way is to save only the binary states and reconstruct the signal by making use of the same timing information of the "Change" bit. This is possible because any toggles at the "Up/Dn" bit always occur at the same instants when pulses are generated at the "Change" bit, since the level-crossing ADC always update the two bits simultaneously. Not only does this method allow me to save another set of recording/reproducing circuits, but it also helps to avoid errors caused by misalignments in the two reproduced signals. The implementation is shown in Fig. 6.1: the same control signals "WT0" – "WT3" used to select memristors are used to direct the "Up/Dn" states to their corresponding D latches, which upon reproduction, will update the states of “Up/Dn_out” at the right times according to “RD0_sel” – “RD3_sel”.

The schematic shown in Fig. 6.1 is a simplified example for illustration purpose only. Some auxiliary control signals for initialization and reset are purposely left out to keep the diagram clear and easy to read. The real circuit designed to record and reproduce ECG signals also included a much larger number of memristors and D latches for storage. Multiplexed addressing techniques were therefore employed to make the selection more efficient.

6.1.2 Sinusoidal Signals

The same two-tone sinusoidal signal used in Section 4.2 was used to test the performance of the designed timing storage circuit when integrated in a large scale to achieve recording and reproducing of CT digital signals. The original analog input was a two-tone signal comprising two sinusoids at 13Hz and 41Hz. Level-crossing sampling was performed to convert the signal to its CT digital counterpart using a 5-bit quantizer.

The reproduced signal is plotted as a red curve in Fig. 6.2. Compared to the original signal which is plotted as a blue curve, the timing difference is marginal throughout the period of recording. The circuit consumes an average power of 388.1 nW during recording and 873.6 nW during reproducing.

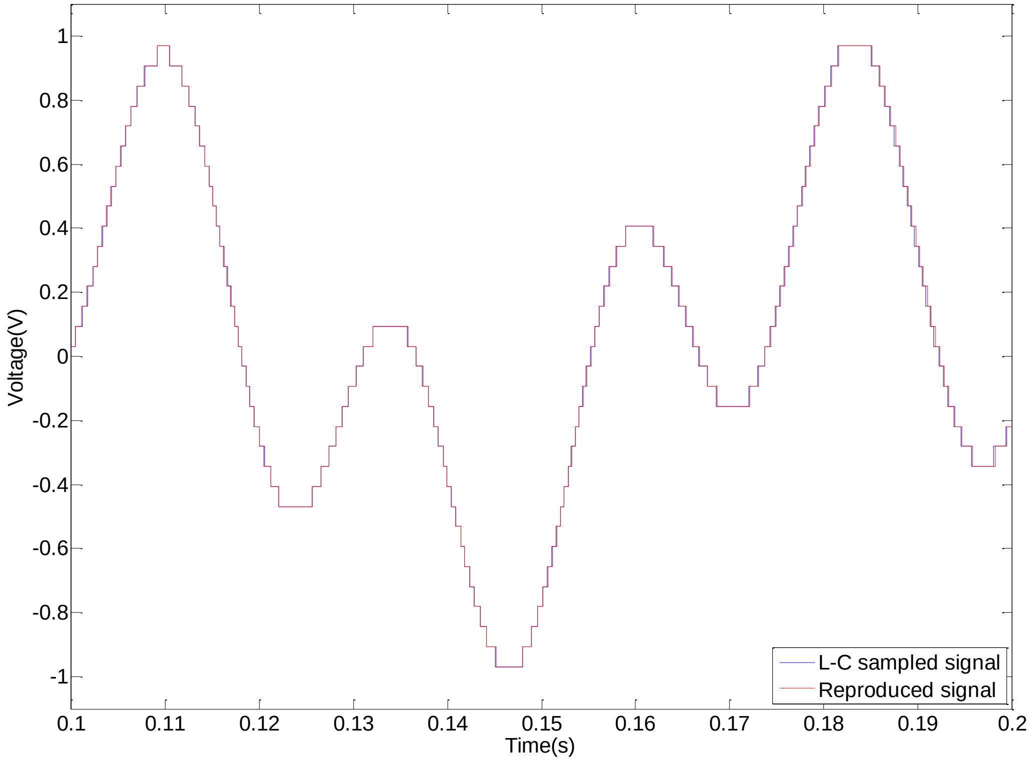

Fig. 6.2 Reproduction of the level-crossing sampled sinusoidal signal using the proposed timing storage circuit. The blue curve represents the original signal and the red curve represents the reproduced signal.

6.1.3 Biomedical Signals

The same ECG signal shown in Fig. 3.4 was used again to test the proposed timing storage circuit in biomedical applications. However, unlike the 5-bit level-crossing ADC with 25% hysteresis demonstrated in Section 3.2.4, this time a 6-bit quantizer is used to record not only the ECG waveforms, but also the small fluctuations caused by noise and interference. A different approach will be shown in Section 6.2.3 to remove such fluctuations by using CT DSP.

Fig. 6.3 shows a portion of the level-crossing sampling result of record 151 from the database. As represented by the series of bubbles, the level-crossing sampled output tracked the analog input (the solid curve) very well: all the P, Q, R, S, and T waveforms were accurately recorded. From the inset which zooms into a slowly-varying segment, it is fair to conclude that most of the details were well preserved during the digitization process. A total of 1595 samples were produced for this 10-second portion, leading to an average sampling rate of 159.5 Hz. Compared to the 360 Hz sampling frequency used in the original data, a reduction of 55.7% was achieved.

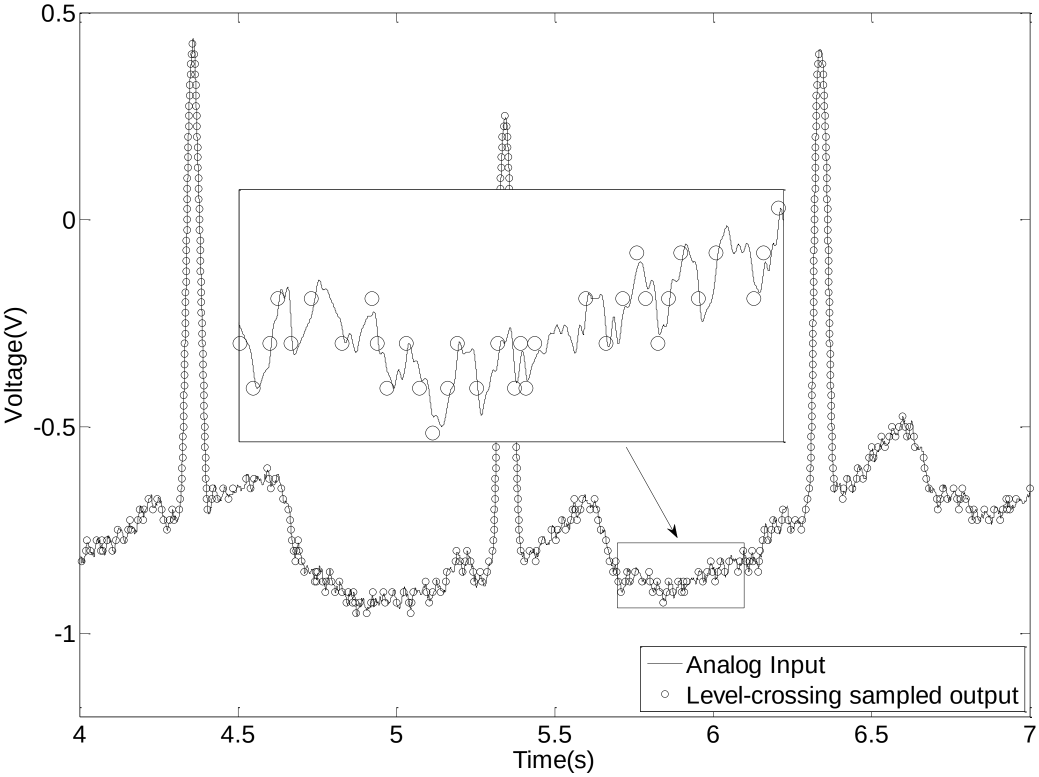

Fig. 6.3 Level-crossing sampling of ECG signal #151 retrieved from the MIT-BIH arrhythmia database. The solid curve represents the original analog input and the series of bubbles represent the level-crossing sampled output.

This level-crossing sampling result shown in Fig. 6.3 was then fed into the memristor-based timing storage circuit to test its performance. In Fig. 6.4, the solid curve represents the original level-crossing sampled output and the dotted

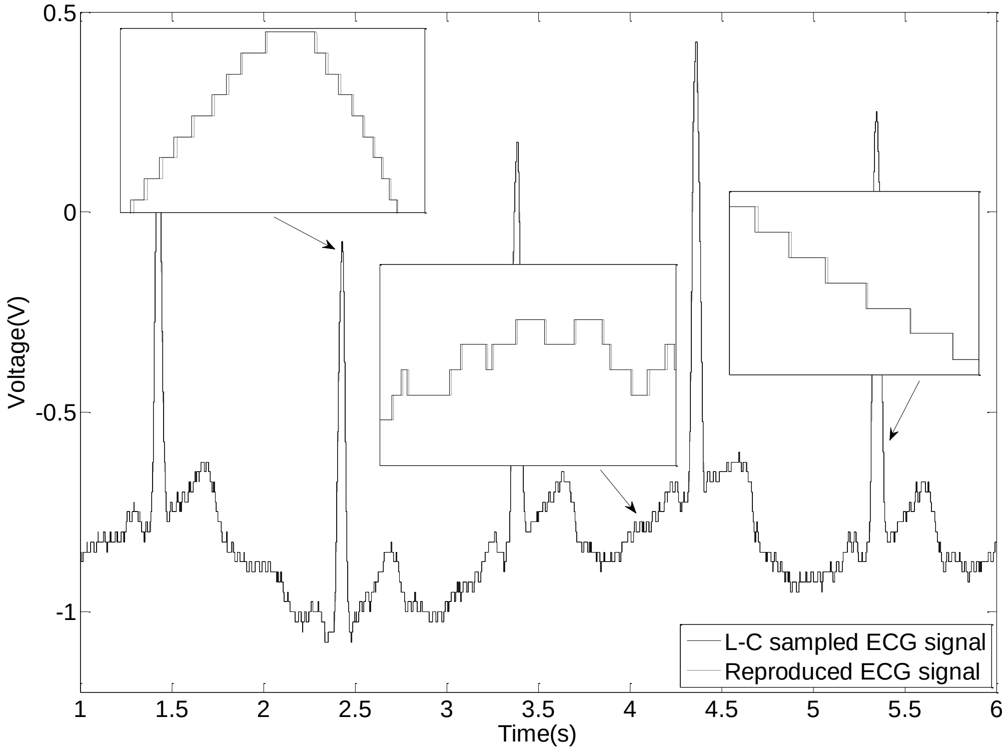

Fig. 6.4 Reproduction of the level-crossing sampled ECG signal using the proposed timing storage circuit. The solid curve represents the original signal and the dashed curve represents the reproduced signal.

curve represents the reproduced result. The fact these two curves almost coincide with each other demonstrates the accuracy of timing storage. Even from the insets which zoom in to reveal the details, negligible timing difference is observed throughout the period of recording.

The circuit consumes an average power of 148 nW during recording and 310.53 nW during reproducing. The instantaneous power consumption of the circuit during recording actually varies depending on signal activity: over the QRS peaks when the signal changes more rapidly, the power consumption gets close to 1 µW, while over the slowly varying segments, the power consumption drops to around 280 nW. Such difference arise from the increase in current through the comparator during switching, and it is also in line with the event-driven nature of CT DSP systems.

The longest time spacing between consecutive level-crossing samples for this signal is 43.3 ms. Since the memristor model used has a maximum storable interval length of 44.6 ms from the parameter values chosen, no overflow was observed during recording, where overflow is defined as the recording of an interval with a length exceeding the limit. In general, the longest time spacing is predictable in biomedical applications, so overflow can be avoided with carefully selected current and memristor parameters. On the other hand, overflow could happen in applications involving signals with much longer periods of inactivity. In such cases, less sensitive memristors need to be used to cater for longer interval storage. Although this would solve the problem but the accuracy for storing short intervals will inevitably drop. A better solution is to use a signal-dependent variable current source to record and reproduce time intervals in different ranges. The “Up/Dn” bit can simply be expanded to multiple bits to store this additional information about the level of current used.

6.2 CT Digital Filters

6.2.1 Memristor-Based Delay Blocks

Delay blocks in CT digital filters need to be implemented in an analog way so that signals defined in continuous time can be delayed with no information loss. This means simple registers used in conventional digital filters are not applicable in CT DSP systems. This small change translates into significant increase in both power consumption and chip area, so significant that these delay blocks even become the dominant part of the entire system both power-wise and area-wise [43, 57, 58].

The problem of delay implementation becomes more serious when processing biomedical signals. With very low signal bandwidth, the tap delay in biomedical signal processing filters are usually in the milliseconds range. Such long analog delay are usually very difficult and costly to realize. In [59], a series of five analog Butterworth filters were cascaded to produce a 1.65-ns tap delay with each filter tap consuming 10 mW of power. The power efficiency of such purely analog implementation is very low due to the static biasing current of the analog filters. Simple inverter chains can also be used to delay CT digital signals. Although the switching power consumed by each inverter is negligible, a huge number of inverters are usually needed to achieve the required tap delay due to the short delay of each inverter, making the total power consumption very high. In [57], simple inverter chains were employed to build a 16-tap CT FIR filter with 6.4-μs tap delay. Fabricated using a 0.25-μm technology, the delay blocks consume 9.84 mW of power and occupy 6.6-mm^2 chip area. To increase the unit delay of each inverter and thereby reducing the total number of inverters needed, current-starved inverter chains have also been used [60]. However, due to the long transition times as a result of limited switching current, there is a long period of time over which both the NMOS and the PMOS transistors in the same brunch are conducting at the same time, causing significant energy waste. To solve this problem, a CMOS thyristor was added to each current-starved inverter cell to speed up the switching using regenerative positive feedback once the inverter output gets close to the mid-level [61]. Improvements had been made in this approach by adding self-resetting and charge recycling capabilities to further reduce power and area [62, 63]. Research in this type of inverter chain-based delay blocks is still ongoing, but it is believed that it may never be feasible to implement tap delay in milliseconds or even longer using this approach. In [43], the 1-μs tap delay blocks designed from [62] occupy 65.6% of the total active area of a 16-tap CT FIR filter chip fabricated using a 90-nm technology, and to achieve a tap delay of 1ms, these blocks would have to be duplicated 1000 times, giving rise to power and area that are hardly acceptable. To solve this long tap delay implementation problem, the memristor-based timing storage circuit proposed in Section 5.4 is made use of to implement CT delay blocks in a more efficient way.

With the ability to record and reproduce CT digital signals, a similar circuit as the one shown in Fig. 6.1 can be used to delay the signal by reproducing it after a time period of nτ, where n is the tap index and τ denotes the unit tap delay. As shown in Fig. 6.5, the only modification needed is to separate the “RD” branches of all memristors and connect them to another current mirror controlled by “RD_state” while having the original current mirror controlled by “WT_state” alone. This allows the circuit to record the incoming signal into one memristor cell and at the same time reproduce the delayed signal from another previously recorded cell. Since every cell can be reused after a recording/reproducing cycle is completed, the total number of memristor cells required can be greatly reduced as the delay block only needs to buffer the signal for a period of nτ. In terms of addressing, the

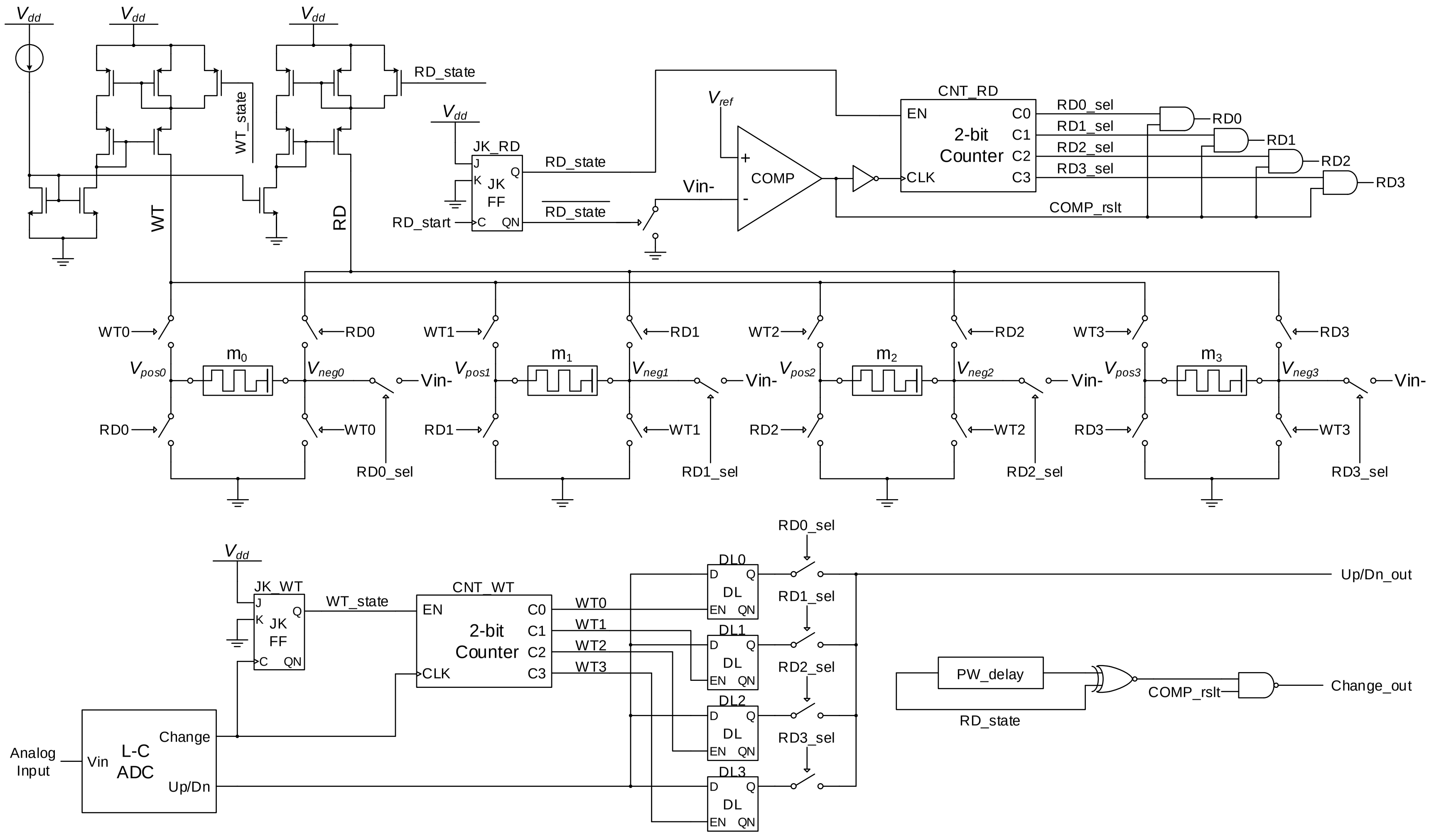

Fig. 6.5 Schematic of a delay block for CT digital signals consisting of four timing storage cells.

ring counters select the cells in circular order, which achieves cell reuse exactly the way needed. To ensure proper functioning, the number of cells needed to build each delay block should be no less than twice the maximum number of samples that can be generated within a period of nτ. If this condition is violated, errors could arise when the circuit tries to record a new interval into a cell with timing information that has not yet been reproduced.

On the other hand, when an interval longer than the delay period of nτ needs to be recorded, the circuit will reproduce from a cell that has not yet finished its recording. This seemingly intolerable scenario in fact will not cause any problems at all, thanks to the symmetrical topology adopted in the design of the memristor cells. When the “RD” switches are turned on while the “WT” switches of the same cell has not yet been turned off, the current from both current mirrors will be passed directly to ground without going through the memristor. The memristance will therefore remain constant during this period of recording/reproducing overlap. Once the “WT” switches are turned off, it will take the same amount of time for the reproducing current to recover the drop in memristance caused previously by the recording current. The total length of the reproduced interval including the overlap period should thus also be equal to the total length of the recorded interval.

It should be noted that the schematic shown in Fig. 6.5 is again a simplified example illustration for clarity. The real circuit designed for CT digital filters included a larger number of memristors and D latches as well as some auxiliary control signals.

6.2.2 CT FIR Low-Pass Filter

Using the proposed memristor-based delay blocks discussed in Section 6.2.1, a 15-tap CT FIR low-pass filter was designed to filter the two-tone sinusoidal signals previously digitized in continuous time, as illustrated in Section 4.2.

The schematic of the design is shown in Fig. 6.7. To adopt asynchronous delta modulation, the output of the level-crossing ADC is encoded using only two bits: a “Change” bit indicates the occurrence of every level-crossing event by generating a short pulse of ‘1’ and stays at ‘0’ during the rest of the time, while another “Up/Dn” bit records the direction of crossing with ‘0’ and ‘1’ representing downward crossings and upward crossings respectively. These two bits combined allow me to know at any instant if the signal has increased, decreased, or remained unchanged. To facilitate initialization, the level-crossing ADC was designed in such a way that the first pulse at the “Change” bit indicates a mid-level crossing. With this common knowledge, the DAC or DSP block will always start counting with the MSB initialized as ‘1’ and all other bits initialized as ‘0’s.

By using delta modulation, each filter tap can be simplified to count up or down at each “Change$_n$” pulse by the same step equal to the tap coefficient C$_n$ depending on the state of the “Up/Dn$_n$” bit. Such simplification avoids the need of multipliers in conventional FIR filters, saving both power and chip area for the processing task. This low-pass filter is designed using Matlab based on a uniform sampling frequency of 100 Hz. The coefficients of the filter is summarized in Table 6.1 and the frequency response is shown in Fig. 6.6.

Table 6.1 Coefficients of the FIR low-pass filter

c_0	-0.0156	c_1	-0.0391	c_2	0.0391	c_3	0.0391
c_4	-0.0859	c_5	-0.0469	c_6	0.3125	c_7	0.5469
c_8	0.3125	c_9	-0.0469	c_{10}	-0.0859	c_{11}	0.0391
c_{12}	0.0391	c_{13}	-0.0391	c_{14}	-0.0156		

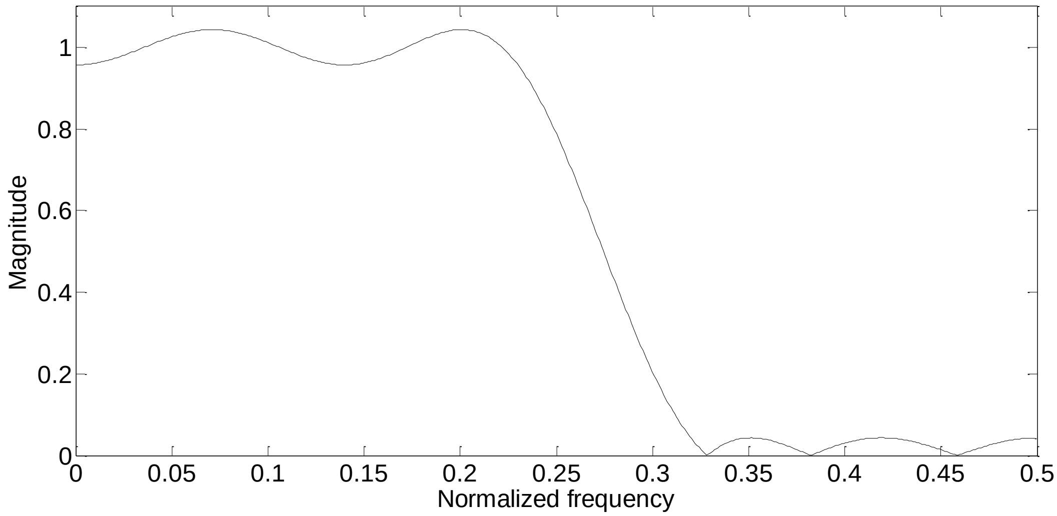

Fig. 6.6 Frequency response of the low-pass filter

Upon startup, “JK_Delay” is cleared to disable the 4-bit counter, causing all 15 delay blocks and their corresponding taps to be idling. When the level-crossing ADC generates the first pulse at its “Change” output, all 15 delay blocks are turned on to start recording the incoming signals. At the same time, “JK_Delay” is set to enable the counter, which has an initial count of ‘1111’, making “C15” the only output to become ‘1’. The counter is clocked by an external square wave with period equal to the tap delay τ, which in this case is 10 ms. The rising edges in this square wave following the enabling of the counter will then assert “C0” to “C14” in sequential order, causing each delay block to start reproducing the delayed signal

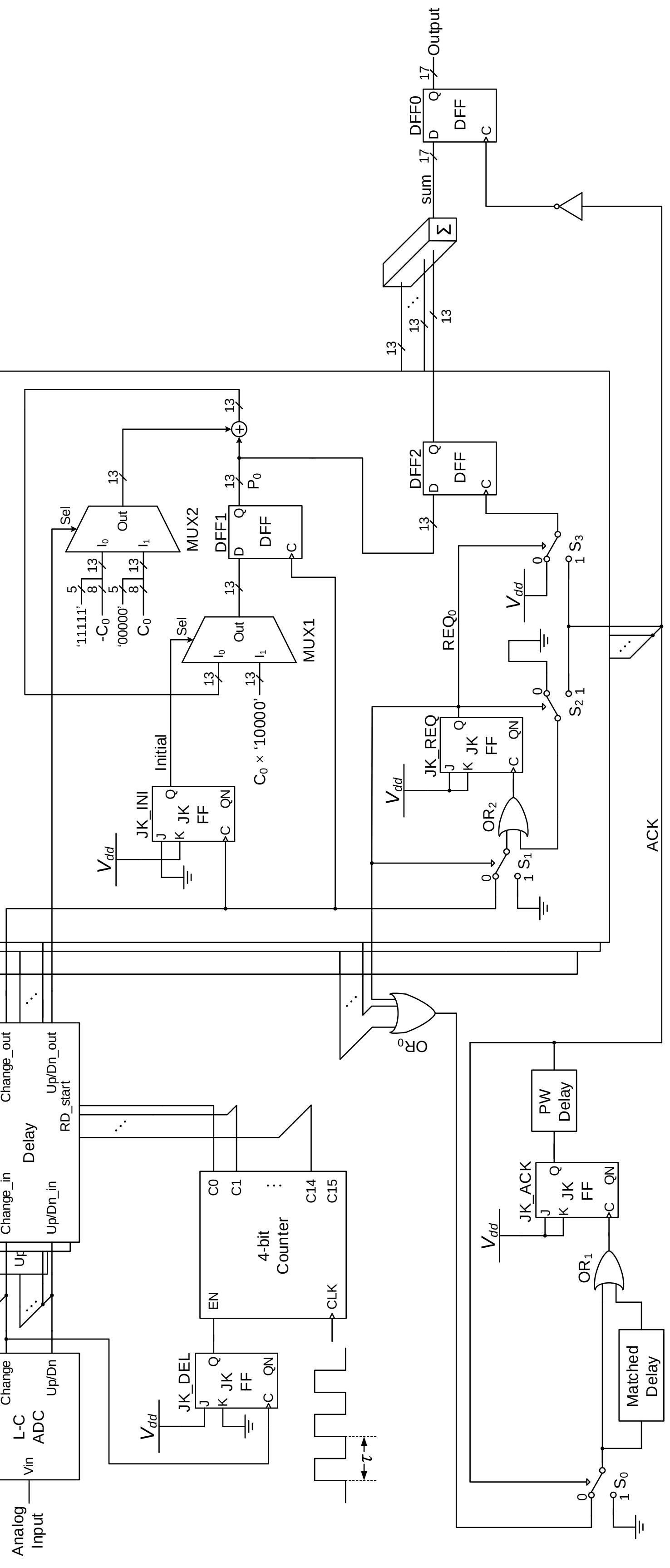

Fig. 6.7 Schematic of a 15-tap CT FIR low-pass filter.

separated by τ. Due to the difficulty in synchronizing the external clock with the first “Change” pulse, the first block can start reproducing at any time within the duration of 0 to τ after receiving the first “Change” pulse. For this reason, the inputs of the first tap are also generated by a delay block instead of taken directly from the ADC so that the same spacing of τ between all the neighboring taps can be guaranteed. This solution also allows easy tuning of the tap delay by changing the clock frequency of the square wave. Once all the delay blocks start reproducing, the external clock can be turned off to save power.

Due to the fact that different delay blocks need to buffer the signals for different durations, the number of memristor cells needed in each block in fact increases in sequential order from the first one controlled by “C0” to the last one controlled by “C14”. As discussed earlier in Section IV, the number of memristor cells required in each block should be no less than twice the maximum number of samples that can be generated within a period of $n\tau$. In our design, the memristor count for the first block is 44 while that for the last block is 414. Statistical analysis of the input signal was made to determine these numbers. It is worth mentioning that while a larger memristor count does correspond to a larger area, the impact on power consumption is in fact quite negligible. At any instant of time, there is only one memristor cell selected for recording and one other memristor cell selected for reproducing for any delay block. The leakage power consumed by additional memristor cells while they are idling is hardly noticeable compared to the power consumed by the current mirrors and comparators.

Within each filter tap, “JK_INI” is preset to make “Initial” asserted, causing “MUX1” to select the initial value equal to the product of C_n and ‘10000’ – the initial output of the level-crossing ADC. Since we quantize the coefficients to 8 bits, the bit-width of tap product P_n is set to 13 bits. When the first pulse arrives from the delayed “Change_n” input, this initial value is latched by “DFF1” to its output P_n. At the same time, “JK_INI” is cleared to make “Initial” ‘0’, which causes the sum of P_n and the output of “MUX2” to be selected by “MUX1” and passed to the input of “DFF1”. From the second pulse onwards, “DFF1” will latch the result of $P_n + C_n$ or $P_n - C_n$, depending on the state of the “Up/Dn” bit before the rising-edge. A point to take note here is that when the second pulse arrives, the sum being latched actually corresponds to the “Up/Dn” state selected by “RD0_sel” instead of “RD1_sel” due to the delay of “CNT_RD”, “MUX2” and the adder. This explains why in Fig. 6.5 “DL0” is enabled by “WT0”: when the second pulse arrives at the timing storage circuit, the updated “Up/Dn” bit is latched by “DL0” instead of “DL1” due to the delay of “CNT_WT”. As a result, the reproduced “Up/Dn” bit does not bare the same shape as the original “Up/Dn” bit although the states latched by the filter are still correct.

Due to the asynchronous nature of the signals being processed, two or more taps may have their P_n results updated arbitrarily close in time, posing a potential conflict at the summation block denoted by “Σ” that is responsible for summing up all 15 P_n results. A handshaking mechanism was therefore implemented for arbitration in such scenarios. Every time a new “Change_n” pulse arrives to pass through “S_1” and “OR_2”, “JK_REQ” will toggle to make “REQ_n” asserted, signaling a new summation request. If the summation block is in the idling state as indicated by “ACK” resting at ‘0’, the rise of “REQ_n” will then propagate through “OR_0”, “S_0” and “OR_1” to cause a new toggling at “JK_ACK”. After a very short delay caused by “PW Delay”, “ACK” will be asserted, causing the updated P_n to be latched by “DFF2”, and a new summation begins immediately having the inputs changed. At the same time, the rise of “ACK” will propagate through “S_2” and “OR_2” to toggle “JK_REQ” and bring “REQ_n” back to ‘0’. On the other hand, “S_0” changes its connection from “OR_0” to ground, and while the output of “OR_1” drops to ‘0’, a short pulse of ‘1’ with a pulse width equal to the delay of “PW Delay” is being propagated through a delay block whose delay matches that of the summation block. When this pulse emerges at the output of this delay block, the output of “OR_1” will be reasserted to toggle “JK_ACK” and eventually bring “ACK” back to ‘0’. As this indicates the completion of a summation task, the inverted “ACK” will trigger “DFF0” to latch the updated sum to the output. In cases when other taps also assert their “REQ_n” during this process, these requests will be put on hold as the current summation continues without being disturbed. However, as soon as “ACK” becomes ‘0’, all these newly updated P_n’s will be latched together as a new summation kicks off immediately. It is noted that there is a bit growth of 4 bits in the final sum caused by the addition of 15 binary numbers to prevent overflow. Since the summation block is purely combinational, “DFF0” is needed to shield any glitches from appearing at the output.

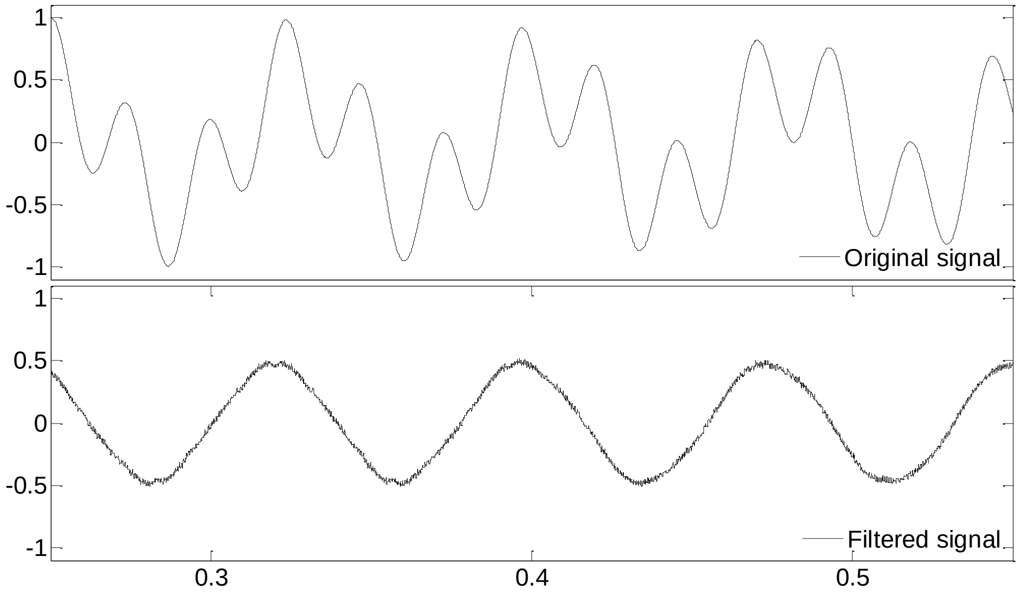

Fig. 6.8 Sinusoidal signal before and after low-pass filtering.

The filtered signal is plotted in Fig. 6.8 for comparison with the original signal. As can be seen, the filter successfully removes the higher tone at 41 Hz while preserving the lower tone at 13 Hz. With a relatively low quantizer resolution chosen for the level-crossing ADC, some high-frequency noise are observed in the filtered signal. This is caused by the asynchronous design of CT digital filters, wherein the output of each filter tap is updated independently. Such high frequency noise will not cause much trouble in real applications as they usually lie beyond the frequency band of interest and will be automatically removed after D/A conversion.

The total power consumption of the filter is 13.65 µW, with each delay block consuming an average of 868.9 nW and the computing part consuming 618.1 nW. Table 6.2 shows a comparison of the memristor-based delay block in this CT FIR low-pass filter to some of the previously published delay circuits. Our proposed

Table 6.2 Comparison of Analog Delay Circuits

	Li [57]	Schell [62]	Kurchuk [63]	This work
V_{DD}	2.5V	1 V	1.2 V	3.3 V
Technology	0.25 μm	90 nm	65 nm	0.35 μm
Power consumption	385.3 μW	10.28 μW	13.70 μW	868.9 nW
Tuning range	N/A	750 ns – 150 ms	150 ns – 150 ms	1.55 ns – 160 ms

delay block has a much wider tuning range that can be easily adjusted by changing the external clock frequency, enabling the use of CT DSP in low-frequency applications. The lower bound of our tap delay is limited by the speed of the counter shown in Fig. 6.7, which can be improved using more advanced CMOS technology. The upper bound on the other hand, depends on the storage capacity, which is determined by the number of memristors available as well as the sampling rate of the level-crossing ADC. For our design, this upper bound is 160 ms, and it can be easily increased by integrating more memristors.

The power consumption of our delay block is also significantly lower than the other delay circuits if they are linearly normalized to achieve the same 62.5-ms delay implemented in our CT FIR filter. However, it needs to be noted that such normalization is made only for the purpose of fair comparison, when it comes to real circuit implementation, area and routing constraints may make it impractical to duplicate these inverter chain-based delay circuits thousands of times to produce the millisecond-delay achieved in this work. Considering the much older technology and higher supply voltage used in our design, the power reduction achievable using the proposed memristor-based approach may even be a lot higher than the results shown in this table. In fact area saving is expected to be a greater advantage but due to the unavailability of commercial memristor fabrication technologies, this remains yet to be verified experimentally.

6.2.3 CT FIR S-G Filter

To study the performance of the proposed memristor-based CT digital filters when processing biomedical signals, the same ECG signal digitized previously as shown in Fig. 6.4 was used again for circuit simulation. A 15-tap CT FIR Savitzky-Golay (S-G) filter was designed to smoothen this ECG signal. Unlike conventional low-pass filters, S-G filters suppress fluctuations with minimal influence on the high-frequency content of the original signals [64]. The schematic of the design is the same as the one shown in Fig. 6.7, except the data widths need to be changed. Since a 6-bit level crossing ADC had been used to digitize the ECG signal, the product of each filter tap need to be expanded to 14 bits, given the same 8-bit quantization used for the filter coefficients. The final output of the filter summing up the result of each individual tap should then be changed to 18 bits, considering the 4-bit growth resulted from the summation of 15 14-bit values.

Unlike the low-pass filter designed in Section 6.2.2 to process sinusoidal signals, where the number of memristor cells needed in each delay block is nearly

Table 6.3 Coefficients of the FIR S-G filter

c_0	-0.0391	c_1	0.0137	c_2	0.0566	c_3	0.0918
c_4	0.1191	c_5	0.1367	c_6	0.1465	c_7	0.1465
c_8	0.1387	c_9	0.1230	c_{10}	0.0977	c_{11}	0.0645
c_{12}	0.0215	c_{13}	-0.0293	c_{14}	-0.0879		

proportional to the duration of its delay, the relation between memristor counts and delay durations for the delay blocks in this S-G filter is highly nonlinear. In fact, the memristor count for the first block is 20 while that for the last block is 156. Statistical analysis of ECG signals were made to determine these numbers.

This S-G filter is designed using Matlab based on a uniform sampling frequency of 240 Hz. The coefficients of the filter is summarized in Table 6.3 and the frequency response is shown in Fig. 6.9.

The filtered ECG signal is plotted in Fig. 6.10. As compared to the original signal, the high-frequency noise has been smoothed out to make the small P waves and T waves much easier to identify, while the fast-changing QRS peaks have not been significantly affected. The total power consumption of the filter is 6.196 µW, with each delay block consuming an average of 406.2 nW and the computing part consuming 103 nW.

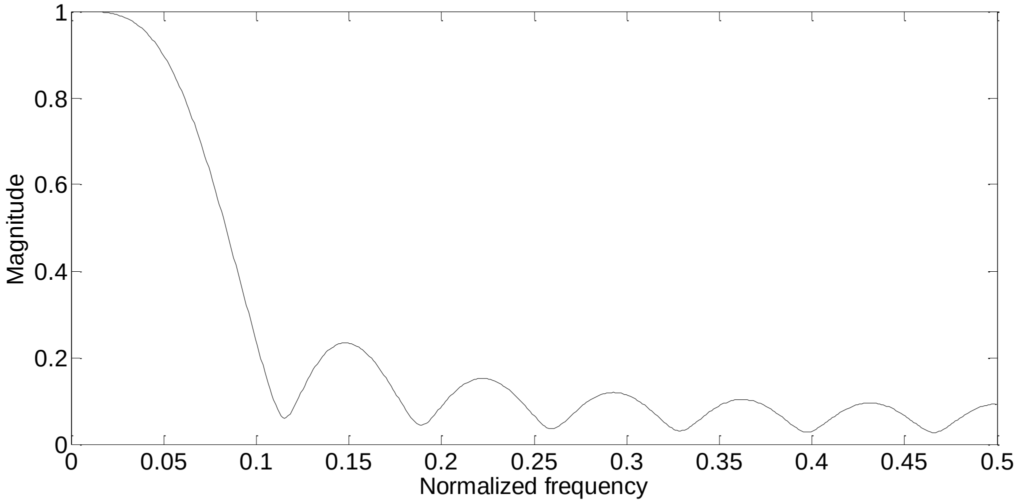

Fig. 6.9 Frequency response of the S-G filter.

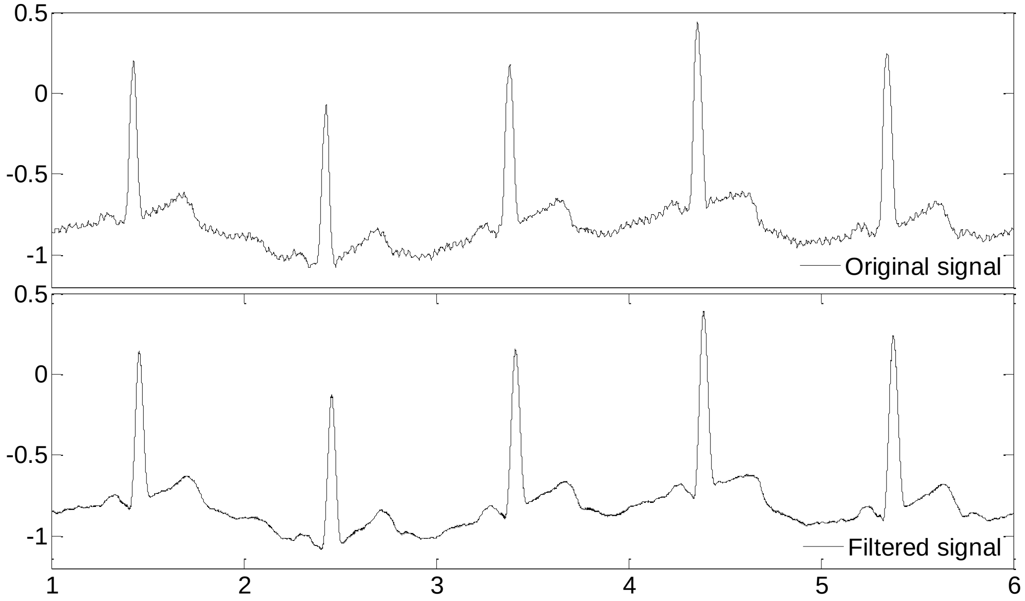

Fig. 6.10 ECG signal before and after filtering.

Table 6.4 Comparison of Analog Delay Circuits

	Li [57]	Schell [62]	Kurchuk [63]	This work
V_{DD}	2.5V	1 V	1.2 V	3.3 V
Technology	0.25 µm	90 nm	65 nm	0.35 µm
Power consumption	18.69 µW	498.4 nW	664.6 nW	406.2 nW
Tuning range	N/A	312.5 ns – 62.5 ms	62.5 ns – 62.5 ms	1.55 ns – 66.7 ms

Table 6.4 compares the performance of the memristor-based delay block in this CT FIR S-G filter to some of the previously published delay circuits. The power consumption of these cited circuits are linearly scaled to achieve the same 62.5-ms delay implemented in my filter. As can be seen from the results summarized in this table, the memristor-based delay blocks again outperformed

other state-of-the-art delay circuits in terms of power consumption and tuning range, in spite of the older technology and higher supply voltage used in my design.

CHAPTER 7

CT FIR FILTERS WITH FREQUENCY RESPONSE MASKING

7.1 Frequency Response Masking in Conventional FIR Filters

The linear phase response of FIR filters gives rise to the benefit of constant group delay, which is an important feature for many applications as it ensures there will be no phase distortion among different frequency components of the signals being processed [19, 20]. However, FIR filters usually require higher orders and therefore more hardware to achieve the same specifications compared to IIR filters. This may become a problem when very sharp transition is needed in the magnitude response of a filter, as the resulting filter order for FIR implementations can be prohibitively high, leading to unrealistic requirements in power and area [65].

To solve this problem, the technique of frequency response masking (FRM) was developed to enable the design of sharp-transition FIR filters with reduced filter orders [66-76]. The basic concept is straightforward: Fig. 7.1(a) shows the frequency response $H_a(e^{j\omega})$ of a moderate-transition low-pass filter with a transition band of width Δ_a. When each delay block of this filter is replaced by a

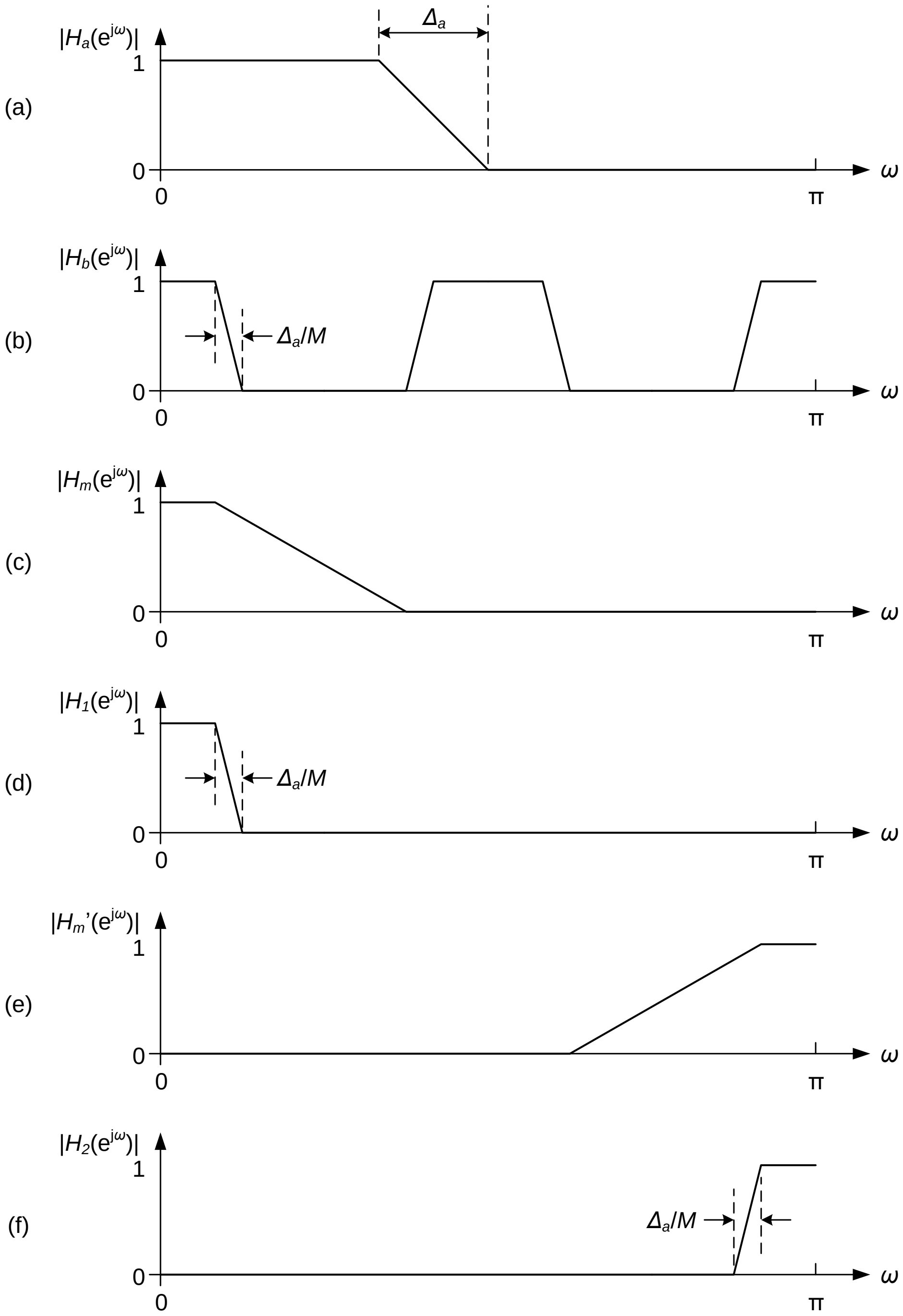

Fig. 7.1 Simple FRM for designing narrow-band sharp-transition FIR filters.

series of M (in this example, M = 4) identical delay blocks, the resulting frequency response $H_b(e^{j\omega})$ is scaled down horizontally by a factor of M with replicas of the pass band falling into the baseband, creating a multiple pass band response as shown in Fig. 7.1(b). When another low-pass masking filter $H_m(\mathrm{e}^{\mathrm{j}\omega})$ with properly selected transition band edges is applied on top of $H_b(\mathrm{e}^{\mathrm{j}\omega})$, the replicas will be removed to create a single pass band low-pass filter $H_1(\mathrm{e}^{\mathrm{j}\omega})$ with M-time steeper roll-off compared to $H_a(\mathrm{e}^{\mathrm{j}\omega})$, as illustrated in Fig. 7.1(c) and (d). Similarly, a high-pass filter with sharper transition $H_2(\mathrm{e}^{\mathrm{j}\omega})$ can be obtained by applying a high-pass masking filter $H_m'(\mathrm{e}^{\mathrm{j}\omega})$ on top of $H_b(\mathrm{e}^{\mathrm{j}\omega})$, as illustrated in Fig. 7.1(e) and (f).

This simple FRM technique enables the realization of sharp-transition frequency response with relaxed roll-off requirements in the filter design, and thereby achieving hardware savings through reduced filter orders. However, it is only useful when the signals to be processed has a much narrower bandwidth compared to the sampling frequency adopted. As is obvious from Fig. 7.1, although the transition bands in $H_c(\mathrm{e}^{\mathrm{j}\omega})$ and $H_d(\mathrm{e}^{\mathrm{j}\omega})$ are narrowed down by the scaling factor of *M*, their pass bands also shrink by the same factor.

To overcome this limitation and extend the benefits of FRM to wide-band applications, the complementary FRM technique was proposed in [66]. The same multiple pass band response $H_b(\mathrm{e}^{\mathrm{j}\omega})$ is first obtained from the original response $H_a(\mathrm{e}^{\mathrm{j}\omega})$ by replacing each delay block with a series of M=4 identical delay blocks, as shown in Fig. 7.2(a). Based on that, a complementary response $H_c(\mathrm{e}^{\mathrm{j}\omega})$ which satisfies

$$\left|H_b\left(\mathrm{e}^{\mathrm{j}\omega}\right)+H_c\left(\mathrm{e}^{\mathrm{j}\omega}\right)\right|=1, \tag{7-1}$$

is derived by subtracting the output of $H_b(e^{j\omega})$ from the input. However, before taking the difference, the input needs to be delayed by the same amount caused by

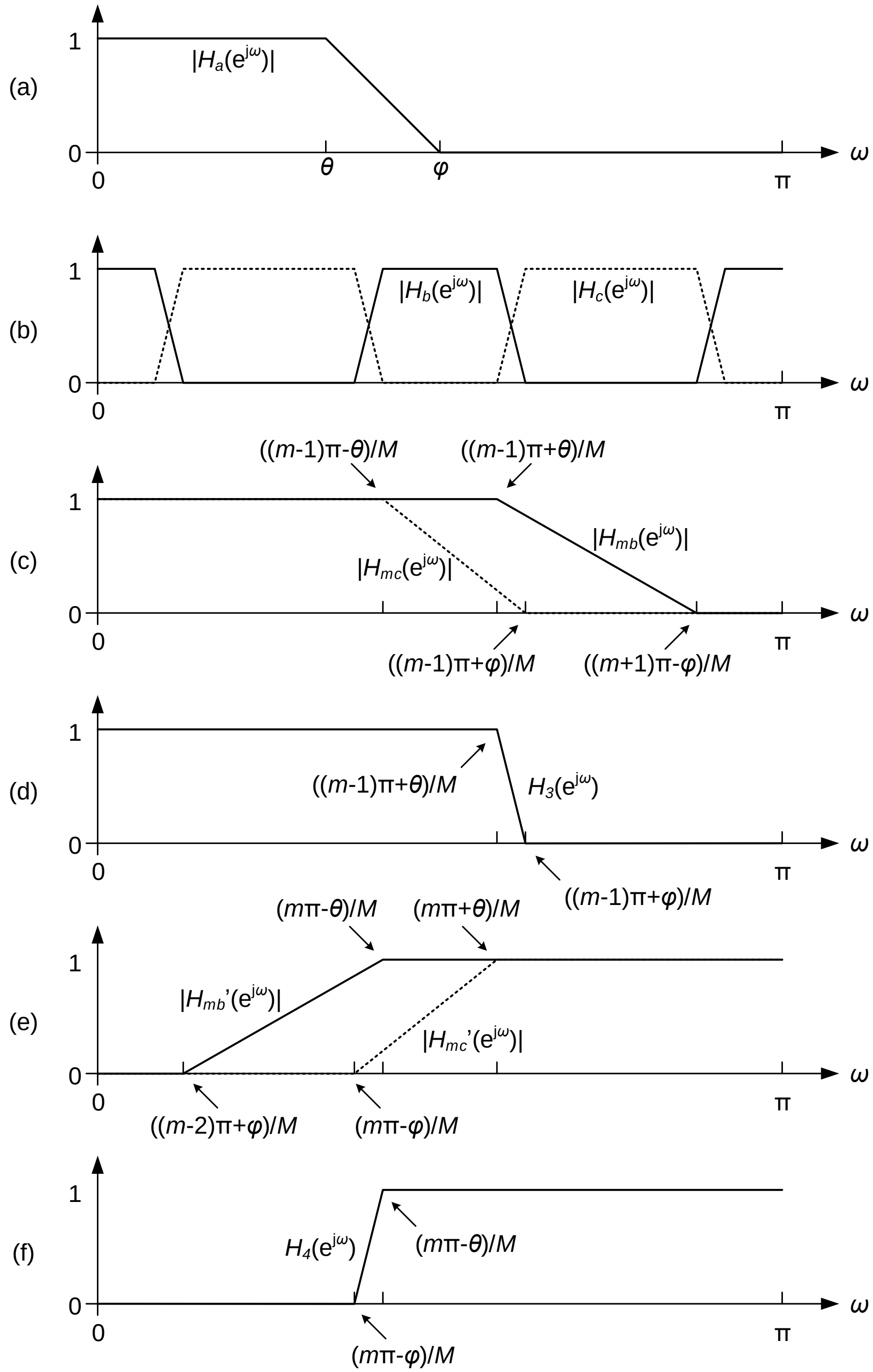

Fig. 7.2 Complementary FRM for designing wide-band sharp-transition FIR filters.

$H_b(e^{j\omega})$ so that it will be in proper synchronization with the output. From Fig. 7.2(b) it is obvious that a unity gain all-pass filter can be formed by summing the outputs of $H_b(e^{j\omega})$ and $H_c(e^{j\omega})$. This property can be used to generate wide-band sharp-transition frequency responses by first masking $H_b(e^{j\omega})$ and $H_c(e^{j\omega})$ appropriately and then summing their outputs.

Fig. 7.2(c) shows an example pair of masking filters $H_{mb}(e^{j\omega})$ and $H_{mc}(e^{j\omega})$ used to preserve only the third transition in $H_b(e^{j\omega})$, and for this reason, parameter m takes the value of 3. $H_{mb}(e^{j\omega})$ is constructed as a low-pass filter that preserves anything up to the end of the third pass band but stops anything beyond the end of the fourth stop band (the frequency range from $((m\text{-}1)\pi\text{-}\theta)/M$ to $((m\text{-}1)\pi+\theta)/M$ is considered as two pass bands because they belong to different replicas of the original response $H_a(e^{j\omega})$, and similarly the frequency range from $((m\text{-}1)\pi+\varphi)/M$ to $((m+1)\pi\text{-}\varphi)/M$ is considered as two stop bands). $H_{mc}(e^{j\omega})$ is constructed as another low-pass filter that preserves anything up to the beginning of the second stop band but stops anything beyond the beginning of the third pass band. When these two masking filters are applied on top of $H_b(e^{j\omega})$ and $H_c(e^{j\omega})$ respectively, and sum up their results, The two stop bands sandwiched between the first two pass bands of $H_b(e^{j\omega})$ will be perfectly compensated by the first two pass bands of $H_c(e^{j\omega})$, creating a single connected wide pass band that spans across the frequency range from 0 to $((m\text{-}1)\pi+\theta)/M$. More importantly, the transition band in the resulting frequency response $H_3(e^{j\omega})$ is reduced by a factor of M, as shown in Fig. 7.2(d). Similarly, a wide-band sharp-transition high-pass filter $H_4(e^{j\omega})$ can be obtained by using the pair of masking filters $H_{mb}'(e^{j\omega})$ and $H_{mc}'(e^{j\omega})$, as illustrated in Fig. 7.2(e) and (f).

The block diagram of a complementary FRM filter is shown in Fig. 7.3. Since the group delay of a symmetric FIR filter is $z^{-(N-1)/2}$, where N is the filter length, with each delay block replaced by a series of M delay blocks, the input signal $X(z)$ must be delayed by $z^{-(N-1)M/2}$ before the output of $H_b(z)$ is subtracted from it to generate the output of $H_c(z)$. The outputs of $H_b(z)$ and $H_c(z)$ are then passed through masking filters $H_{mb}(z)$ and $H_{mc}(z)$ respectively before summed up to form the final output.

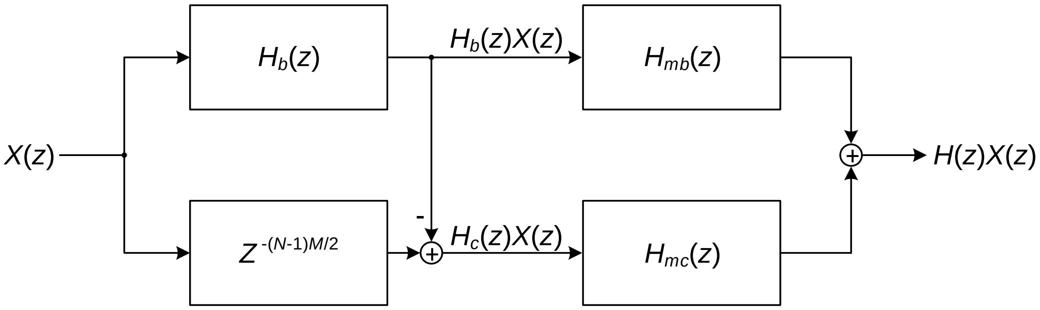

Fig. 7.3 Block diagram of a complementary FRM filter.

There are two points which need to be pointed out. First, in order to avoid the implementation of half delay blocks, either (N-1) or M must be even so that the result of (N-1)M/2 will be an integer. Second, the two masking filters $H_{mb}(z)$ and $H_{mc}(z)$ may not necessarily have the same order, so the difference in their group delay must be offset before the two outputs are summed up. This can be done by inserting additional delay blocks at the end of the filter with a lower order, and again to avoid half delay issues, $H_{mb}(z)$ and $H_{mc}(z)$ must be both odd or both even.

The same filter can be designed with different values of the scaling factor M, which will result in different designs in all three filters $H_b(z)$, $H_{mb}(z)$ and $H_{mc}(z)$. In general, as M increases, the required order of $H_b(z)$ will drop while the required orders of $H_{mb}(z)$ and $H_{mc}(z)$ combined will rise. The optimum value of M is one that corresponds to the lowest combined order of $H_b(z)$, $H_{mb}(z)$ and $H_{mc}(z)$, which can be significantly lower than the order of the filter designed to fulfill the same specifications without using the technique of FRM.

7.2 Frequency Response Masking in CT FIR Filters

7.2.1 Overall structure

The ability to implement sharp-transition frequency responses with reduced filter orders using FRM techniques is very attractive to the design of CT FIR filters as well. However, the need to upscale tap delay by a factor of M presents a big challenge to the existing inverter-chain-based delay block implementations, as it would require the physical duplications of these already very costly circuit blocks.

The memristor-based delay block implementation proposed in CHAPTER 6 provides an ideal solution to this problem. With the ability of delay tuning by changing the frequency of the external square wave, the tap delay can be easily adjusted. To achieve upscaling by an integer factor of M as required in the design of FRM filters, a simple clock divider will suffice. Although this would also require the capacity (the number of memristor cells) of the delay blocks to be scaled up by the same factor to buffer the signals for extended durations, the resulting increase in power consumption is quite negligible. At any instant of time, there is only one memristor cell selected for recording and one other memristor cell selected for reproducing for each delay block. The leakage power consumed by additional memristor cells while they are idling is hardly noticeable compared to the power consumed by the current mirrors and comparators.

Fig. 7.4 shows the proposed block diagram of a CT complementary FRM filter. Compared to the block diagram of a conventional complementary FRM filter shown in Fig. 7.3, the main difference is the addition of one accumulator block and two delta modulator blocks. This is because the design of CT FIR filters can be greatly simplified to achieve power and area savings when their inputs are delta-modulated.

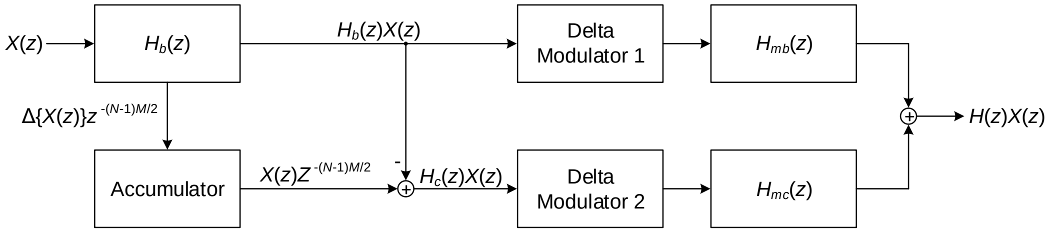

Fig. 7.4 Block diagram of a CT complementary FRM filter.

The delayed version of the input $X(z)Z^{-(N-1)M/2}$ does not need to be generated separately, but instead, it can be obtained directly from $H_b(z)$. Specifically, when the filter length is an odd number, the output of the middle delay block provides the input delayed exactly by the group delay of $H_b(z)$. However, since the delayed signal is delta-modulated, an accumulator is needed to demodulate it before taking difference with the output of $H_b(z)$.

7.2.2 Frequency response

To illustrate the proposed design, a CT FRM high-pass filter is demonstrated to remove the low-frequency baseline wandering noise commonly observed in

ECG signals. The design specifications of this high-pass filter is summarized in Table 7.1. Without using FRM techniques, a 168th-order filter would be required to implement this frequency response with a narrow transition band of only 1Hz.

Table 7.1 Design parameters of the high-pass filter.

Stop-band edge	0.5 Hz
Pass-band edge	1.5 Hz
Pass-band ripple	0.01
Stop-band attenuation	60 dB
Unity tap delay	16.67 ms

Table 7.2 Filter orders with different scaling factor *M*.

M	n_a	n_{mb}	n_{mc}	n_{total}	**Saving**
3	44	0	8	52	58.73%
4	32	0	10	42	66.67%
5	26	0	14	40	68.25%
6	22	0	20	42	66.67%
7	20	0	24	44	65.08%
8	16	0	30	46	63.49%

As discussed in Section 7.1, different values of the scaling factor M will lead to different combined orders of the FRM filter. To find the optimum design that is able to achieve maximum saving, different values of *M* were explored and compared as summarized in Table 7.2. As expected, the order n_a of $H_a(z)$ decreases monotonically as *M* goes up. This is because with a given target transition band width of 1 Hz, a larger scaling factor will result in a wider transition band in $H_a(\mathrm{e}^{\mathrm{j}\omega})$, reducing the roll-off requirements further. The reverse trend is observed in the orders n_{mb} and n_{mc} of the two masking filters $H_{mb}(z)$ and

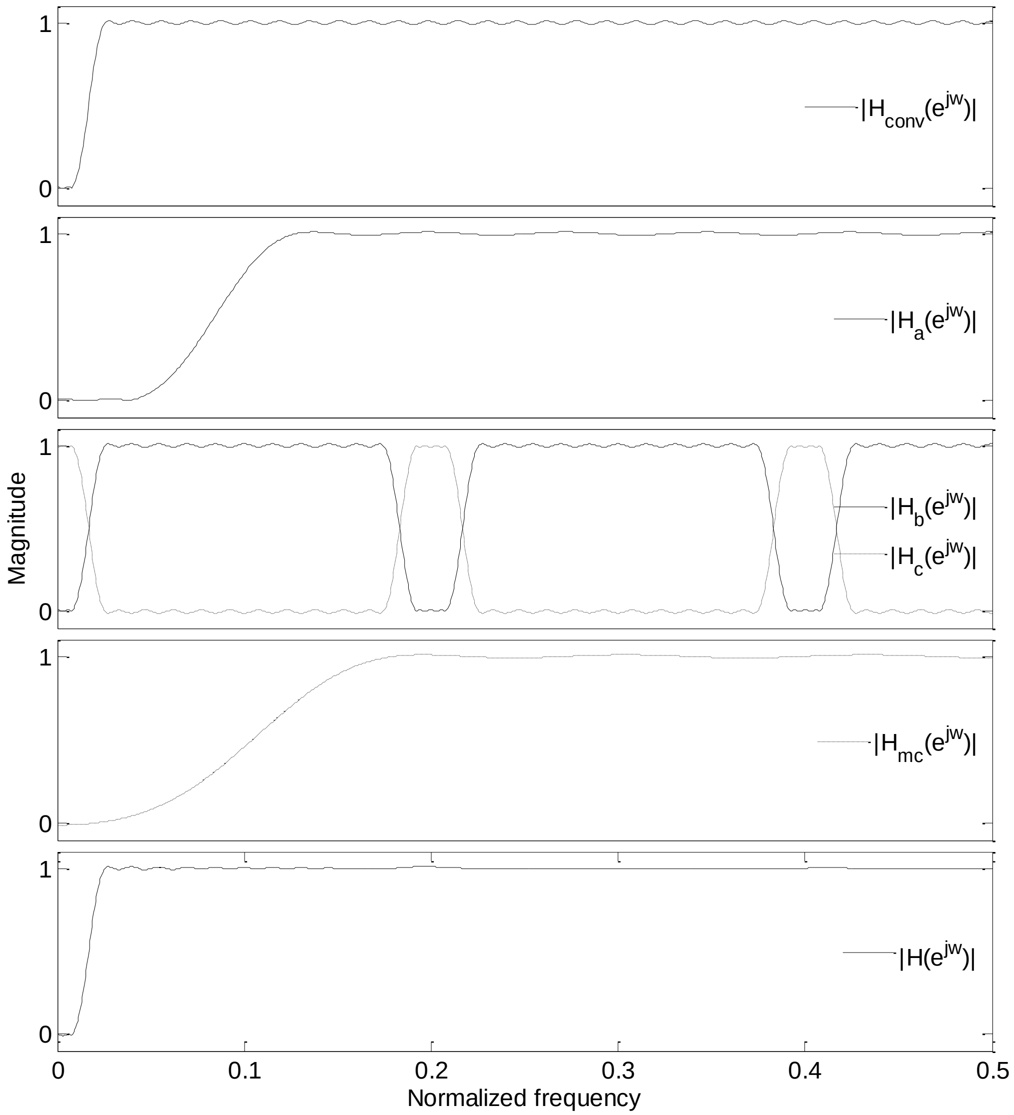

Fig. 7.5 A high-pass filter designed using complementary FRM. The five subplots shows the frequency responses of (a) the conventional filter designed without using FRM, (b) the conventional filter with the transition band widened by a factor of 5, (c) the multiple pass band filter obtained by replacing each delay block in (b) by a series of 5 identical delay blocks (solid line) and its complementary filter (dotted line), (d) the masking filter, (e) the resulting complementary FRM filter.

$H_{mc}(z)$. This is caused by the narrowing of the stop bands in $H_b(e^{j\omega})$ and $H_c(e^{j\omega})$, which then requires sharper transition in the frequency response of the two masking filters. However, since the target design H $(e^{j\omega})$ has a very narrow stop band, it is usually the first transition in $H_b(\mathrm{e}^{\mathrm{j}\omega})$ that will eventually emerge as the only transition in $H\ (\mathrm{e}^{\mathrm{j}\omega})$, which means m usually takes the value of 1 unless the scaling factor M takes a very large value. Since no masking will be needed in $H_b(\mathrm{e}^{\mathrm{j}\omega})$ when m is equal to 1, the order n_{mb} is simply 0 for the relatively low range of values for M. As a result, the optimum value of M is one that gives the lowest sum of n_a and n_{mc}. In this case, the optimum value is 5, which results in a combined order of 40 in the design of the FRM filter. Compared to the 168th-order conventional design without using FRM techniques, a saving of 68.25% is achieved.

The respective frequency responses are plotted in Fig. 7.5. By blocking the first pass band in $H_c(\mathrm{e}^{\mathrm{j}\omega})$ using the masking filter $H_{mc}(\mathrm{e}^{\mathrm{j}\omega})$, the sharpened first transition in $H_b(\mathrm{e}^{\mathrm{j}\omega})$ is uncovered in the final response $H(\mathrm{e}^{\mathrm{j}\omega})$. The missing parts of the pass band in $H_b(\mathrm{e}^{\mathrm{j}\omega})$ is also perfectly filled by the masked response of $H_c(\mathrm{e}^{\mathrm{j}\omega})$. The final frequency response of this FRM filter resembles the shape of $H_{conv}(\mathrm{e}^{\mathrm{j}\omega})$ – the 168th-order conventional design without using FRM techniques. The target design specifications are fully met with a pass-band ripple of 0.0096 and a stop-band attenuation of 63dB.

7.2.3 Accumulator Design

In Section 7.2.2, the order n_a has been determined to be 26, which means $H_a(z)$ will have 27 taps. When this filter is implemented using the CT FIR filter structure discussed in CHAPTER 6, a total of 27 memristor-based delay blocks will be needed to produce the delayed versions of the input for all the taps. To implement $H_b(z)$ which exhibits the frequency response of $H_a(z)$ scaled down horizontally by a factor of $M = 5$, one can simply divide the external clock by a factor of 5 (from 60 Hz to 12 Hz), so that the delay of each block will be extended by 4 times (from 16.67 ms to 83.33 ms). According to the analysis made in Section 7.2.1, since the filter length 27 is an odd number, the 14th delay block provides the input delayed exactly by the group delay of $H_b(z)$. Therefore, the signal $\Delta\{X(z)\}z^{-(N-1)M/2}$ shown in Fig. 7.4 in this case would just be the output of the 14th delay block in $H_b(z)$.

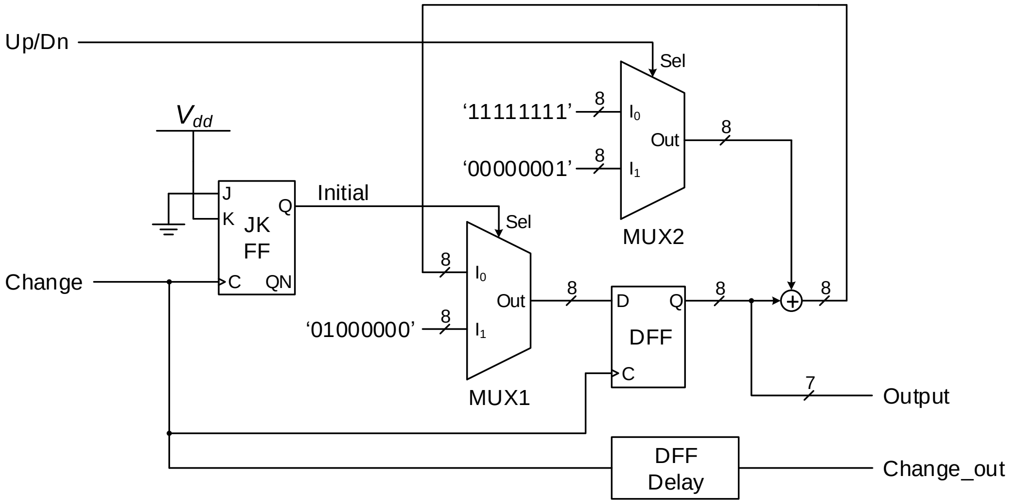

Fig. 7.6 Schematic of a 7-bit accumulator.

An accumulator is then needed to demodulate the delta-modulated signal $\Delta\{X(z)\}z^{-(N-1)M/2}$. The schematic of the accumulator design is shown in Fig. 7.6. The input is digitized in continuous time using a 7-bit level-crossing ADC. To enable the addition of signed numbers, the data are expanded to 8 bits with the MSB reserved as a signed bit.

Upon startup, the JKFF is preset to make “Initial” asserted, causing “MUX1” to select ‘01000000’ – the initial output of the level-crossing ADC. When the first pulse is received from the “Change” input, this initial value is latched by the DFF to the output. At the same time, this pulse also resets the JKFF to bring “Initial” down to ‘0’, which causes the output of the adder to be selected by “MUX1” and passed to the input of the DFF. The two inputs of “MUX2” are connected to the 8-bit two’s complement of +1 and -1 respectively, so from the second pulse onwards, the DFF will latch the result of the previous “Output” value incremented or decremented by 1, depending on the state of the “Up/Dn” bit before the rising-edge of the pulse. The “Change_out” is just the “Change” signal delayed by the Clock-to-Q delay of the DFF, and it can be used to indicate updates in the “Output” value.

Since only the lower 7 bits of the DFF output carry the accumulated value, the MSB, which presumably will always be ‘0’, can simply be ignored if the following processing does not deal with signed numbers. As a result, “Output” is taken from the lower 7 bits of the DFF output.

In fact the realization of $H_c(z)$ is not the only place where this accumulator is needed in the entire design of this CT FRM high-pass filter. As explained in Section 7.2.2, no masking is needed in $H_b(e^{j\omega})$, so no further processing is required for the output of $H_b(z)$. However, the final adder in Fig. 7.4 requires its two input signals to be synchronized in time. A matched delay of $H_{mc}(z)$ should then be added to the output of $H_b(z)$. This can be done by replacing the $H_{mb}(z)$

block shown in Fig. 7.4 with a memristor-based delay block, followed by an accumulator to convert the delta-modulated signal back into binary format.

7.2.4 Adder/Subtractor Design

As can be seen from Fig. 7.4, this CT FRM filter requires a subtractor to generate the output of $H_c(z)$ and an adder to sum up the results of $H_{mb}(z)$ and

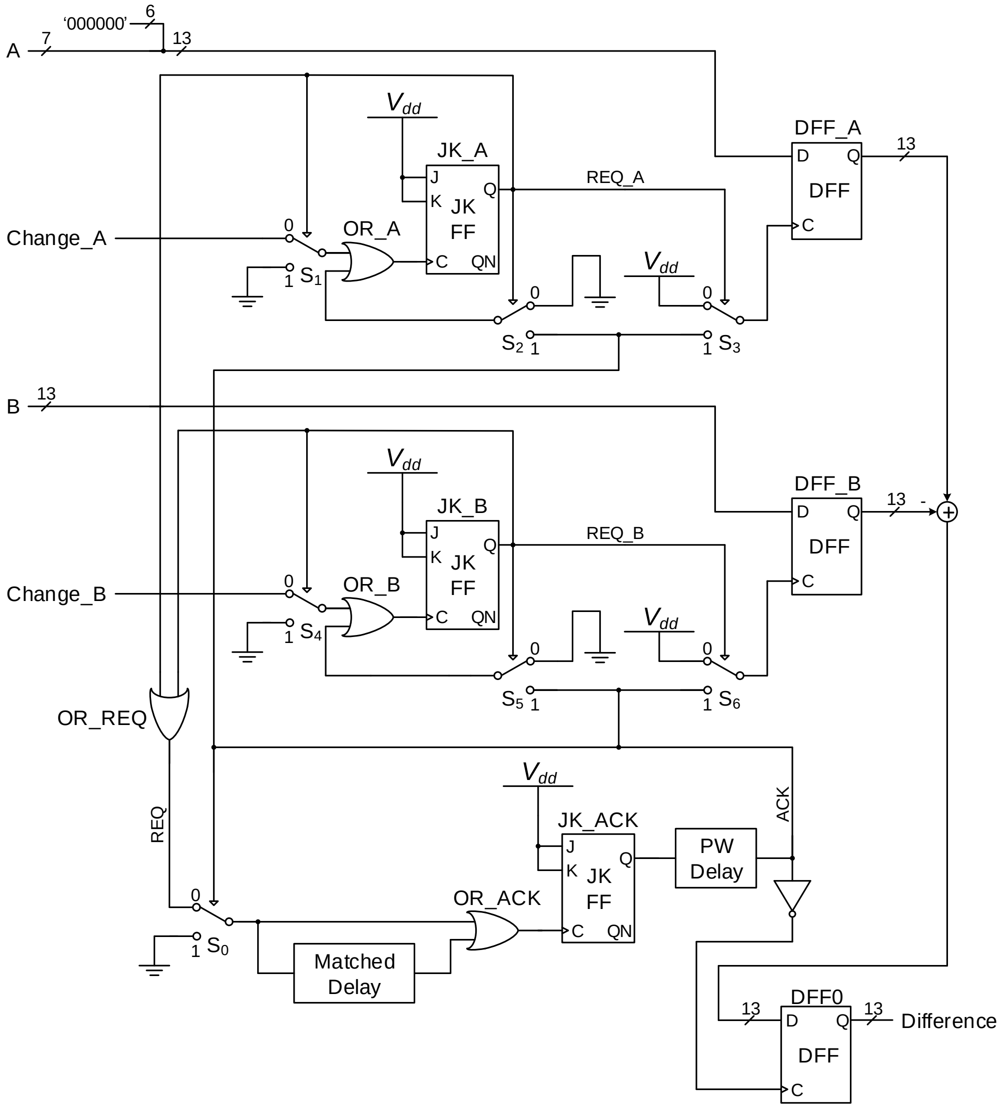

Fig. 7.7 Schematic of a 13-bit CT subtractor.

$H_{mc}(z)$. These two blocks may seem trivial and straightforward, but considering the randomness in their input changes, some special precautions need to be taken in the design to avoid glitches and metastability problems.

The schematic of the subtractor is shown in Fig. 7.7. The two input signals A and B are taken from the accumulator and $H_b(z)$ respectively. Due to the difference in bit-width, the output of the accumulator will be extended by 6 bits while the output of $H_b(z)$ will have its lower 7 bits truncated. Such truncation was made for considerations related to the design of delta modulators, which will be explained in the next subsection. In the end, both inputs as well as the output of the subtractor have the same bit width of 13.

Both A and B come with signals to indicate changes in them in the form of positive transitions: for the accumulator this is simply the “Change_out” output, while for $H_b(z)$ this is the inverted “ACK” signal passed through a delay block having the matched Clock-to-Q delay of the DFF. A handshaking mechanism was therefore incorporated in the design. All three JKFFs in the schematic are connected as TFFs and have all been cleared upon startup. Every time a change occurs in A, a positive transition in “Change_A” will propagate through “S_1” and “OR_A” to toggle “JK_A”. The asserted “REQ_A” signals a new subtraction request from input A. If the arithmetic unit (represented by the “+” symbol) of the subtractor is in idle state as indicated by “ACK” resting at ‘0’, the rise of “REQ_A” will then propagate through “OR_REQ”, “S_0” and “OR_ACK” to cause a new toggling at “JK_ACK”. After a very short delay caused by “PW Delay”, “ACK” will be asserted, causing the updated input to be latched by

"DFF_A", and a new subtraction begins immediately at the arithmetic unit having its input changed. At the same time, the rise of "ACK" will propagate through "S_2" and "OR_A" to toggle "JK_A" and bring "REQ_A" back to '0'. On the other hand, "S_0" changes its connection from "OR_REQ" to ground, and while the output of "OR_ACK" drops to '0', a short pulse of '1' with a pulse width equal to the delay of "PW Delay" is being propagated through a delay block whose delay matches that of the arithmetic unit. When this pulse emerges at the output of this delay block, the output of "OR_ACK" will be reasserted to toggle "JK_ACK" and eventually bring "ACK" back to '0'. As this indicates the completion of a subtraction task, the inverted "ACK" will trigger "DFF0" to latch the updated difference to the output. In case "Change_B" also asserts "REQ_B" during this process, this request will be put on hold as the current subtraction continues without being disturbed. However, as soon as "ACK" becomes '0', the newly updated B input will be latched as a new subtraction kicks off immediately. Such handshaking avoids potential conflicts at the arithmetic unit when inputs A and B update their values very close in time.

The same circuit structure is used to design the adder that sums up the results of $H_{mb}(z)$ and $H_{mc}(z)$, so the details will be omitted.

7.2.5 Delta Modulator Design

The memristor-based delay blocks and CT digital filters proposed in this work require their inputs to be delta-modulated. For this reason, two delta modulators are needed to modulate the outputs of $H_b(z)$ and $H_c(z)$ before they are fed into

$H_{mb}(z)$ and $H_{mc}(z)$ respectively, as can be seen from the system block diagram shown in Fig. 7.4. The schematic of a delta modulator is shown in Fig. 7.8. Just like the way a level-crossing ADC converts an analog signal into a “Change” bit indicating the occurrence of level-crossing events and an “Up/Dn” bit indicating the direction of crossing, a delta modulator is required to do the same for a CT

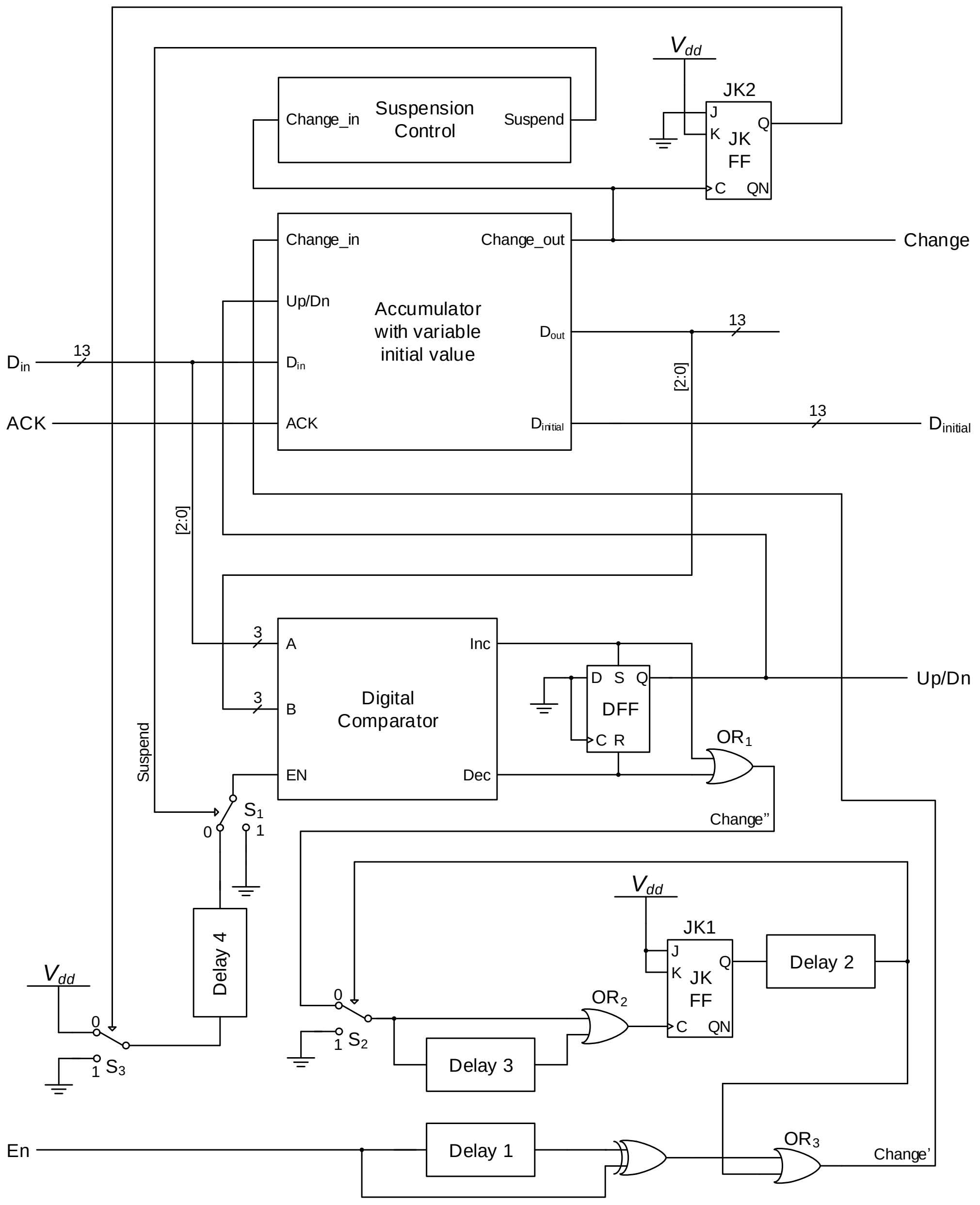

Fig. 7.8 Schematic of a 13-bit delta modulator.

digital signal. In addition to that, a delta modulator also needs to record the initial value it starts at and pass it to the following blocks, as such information is necessary for the demodulation of the CT digital signal but it is not contained in either the “Change” or the “Up/Dn” bit.

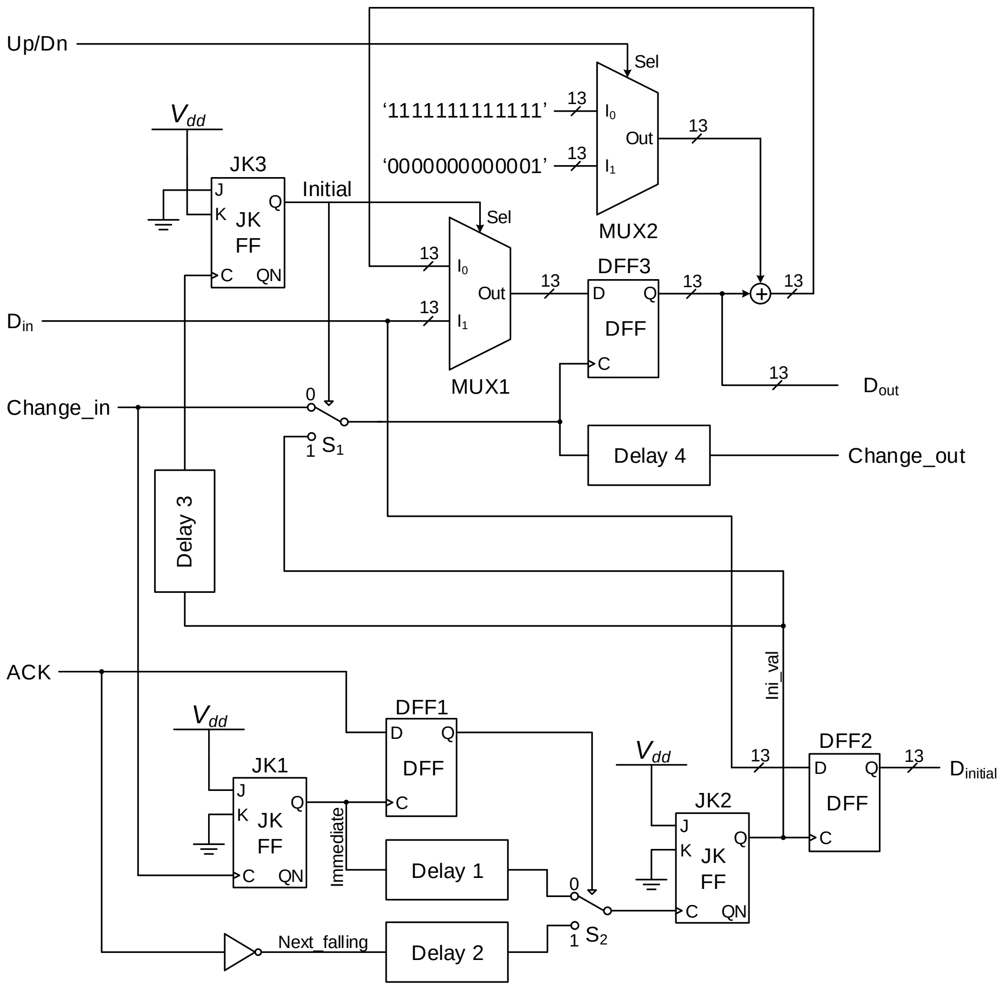

Fig. 7.9 Schematic of a 13-bit accumulator with variable initial value.

The delta modulator starts its operation once the input "En" is asserted. Due to the delay caused by "Delay 1", there will be a short period when the two inputs of the XOR gate are not the same, which will then give rise to a short pulse of '1' at its output. This pulse will then propagate through "OR$_3$" and be fed into the "Change_in" input of the accumulator with variable initial value block.

The schematic of the accumulator with variable initial value is shown in Fig. 7.9. "Initial" takes the state of '1' since "JK1" has been preset upon startup, which

Table 7.3 Time to take the initial value under different conditions.

REQ	ACK	Soonest moment when "D_{in}" is in a steady state
'0'	'0'	Immediate
'0'	'1'	After the next falling edge of "ACK"
'1'	'0'	Immediate
'1'	'1'	After the next falling edge of "ACK"

prevents the first "Change_in" pulse from passing through "S_1" and "Delay 4" to appear directly at "Change_out". This is done to make sure the accumulator takes its initial value at an appropriate time. Since the input "D_{in}" is taken from a CT digital signal that is constantly changing, there is a chance that "D_{in}" is in a transition state at the instant when the first "Change_in" pulse arrives. Errors could arise in the initial value if it is taken from "D_{in}" straight away. As a result, the accumulator needs to wait and find the soonest moment when "D_{in}" is in a steady state. There are four different scenarios depending on the state of "REQ" and "ACK" of the preceding stage (for "Delta Modulator 1" this is just the "ACK" signal of $H_b(z)$ while for "Delta Modulator 2" this is the "ACK" signal of the subtractor), as summarized in Table 7.3. When both "REQ" and "ACK" are '0', the preceding block is in idling and the initial value can be taken immediately. When "REQ" is '0' and "ACK" is '1', the preceding block is busy computing and updating its output. The accumulator then needs to wait for "ACK" to drop to '0', as this would indicate the completion of an output update in the preceding block. When "REQ" is '1' and "ACK" is '0', an update request has just been made but the preceding block has not yet started its computation, which means the accumulator will have enough time to take its initial value before the next transition. When both “REQ” and “ACK” are ‘1’, the preceding block is busy handling its current update request, while a new request is in the waiting. In this case, the accumulator needs to take its initial value as soon as “ACK” drops to ‘0’.

Based on the previous discussion, it is enough to find the soonest moment when “D_{in}” is in a steady state by observing the “ACK” signal from the preceding stage. This is done by sampling the state of “ACK” at the time of the first “Change_in” pulse. Both “JK1” and “JK2” have been cleared upon startup. The arrival of the first “Change_in” pulse sets “JK1”, and the asserted “Immediate” signal triggers “DFF1” to latch the state of “ACK”. If this latched state is ‘0’, the positive transition in “Immediate” will propagate through “Delay 1” and “S_2” to set “JK2”, and the asserted “Ini_val” will trigger “DFF2” to latch the initial value immediately. “Delay 1” has the matched Clock-to-Q delay of “DFF1” and the switching delay of “S_2” combined, and it is inserted to make sure the connection at “S_2” has settled in accordance with the sampled “ACK” state at the time of the first “Change_in” pulse. If the state latched by “DFF1” is ‘1’ however, then the accumulator will wait for “ACK” to drop to ‘0’. When this happens, the positive transition at the output of the inverter will propagate through “Delay 2” and “S_2” to set “JK2”, which will then cause “DFF2” to latch the initial value. “Delay 2” has the matched delay of “JK1”, “DFF1” and “S_2” combined to serve the same purpose of “Delay 1” in case “ACK” drops to ‘0’ right after the first “Change_in” pulse. In both cases regardless of the sampled “ACK” state, the positive transition at “Ini_val” will pass through “Delay 3” to reset “JK3”, and the cleared “Initial” will cause “S_1” to change its connection from “Ini_val” to “Change_in”. A positive pulse will then be generated to pass through “Delay 4” and appear at “Change_out”. The width of this pulse is determined by the delay of “Delay 3”. After that, the accumulator will resume its operation in the same way a simple accumulator shown in Fig. 7.6 does. The “Change_out” and “$D_{initial}$” outputs of this accumulator are connected to the “Change” and “$D_{initial}$” outputs of the delta modulator.

The output “D_{out}” of the accumulator is fed into a digital comparator to be compared with the input “D_{in}” to determine if there is a change in the signal. Due to the finite speed of digital circuitry, the output from the preceding block needs to be truncated, so that the “Change” pulses will be generated at a reasonable rate that is manageable to the following blocks. In my design, only the higher 13 bits of the 20-bit output are used while the lower 7 bits are discarded.

Due to the fact that each tap within a CT FIR filter operates independently, it is common to see spikes in the filter output. As a result, the change in “D_{in}” may not always be continuous. In my design, it is assumed that any change will not exceed 3 LSBs, where the LSB is defined in the context of 13-bit signals. With this assumption, the design of the digital comparator can be greatly simplified. This is because for the detection of changes within 3 LSBs, only the lower 3 bits of the signal need to be observed while the higher bits can simply be ignored. The truth tables of “Inc” and “Dec” of the digital comparator is shown in Table 7.4 and Table 7.5 respectively. These two truth tables can be described using hardware description languages like Verilog and then synthesized and optimized

Table 7.4 Truth table of “Inc” of the digital comparator.

A B	000	001	010	011	100	101	110	111
000	0	1	1	1	0	0	0	0
001	0	0	1	1	1	0	0	0
010	0	0	0	1	1	1	0	0
011	0	0	0	0	1	1	1	0
100	0	0	0	0	0	1	1	1
101	1	0	0	0	0	0	1	1
110	1	1	0	0	0	0	0	1
111	1	1	1	0	0	0	0	0

Table 7.5 Truth table of “Dec” of the digital comparator.

A B	000	001	010	011	100	101	110	111
000	0	0	0	0	0	1	1	1
001	1	0	0	0	0	0	1	1
010	1	1	0	0	0	0	0	1
011	1	1	1	0	0	0	0	0
100	0	1	1	1	0	0	0	0
101	0	0	1	1	1	0	0	0
110	0	0	0	1	1	1	0	0
111	0	0	0	0	1	1	1	0

using synthesis tools like Synopsis Design Compiler. An enable signal “EN” is added so that both “Inc” and “Dec” will always be ‘0’s unless “EN” is asserted.

The “Inc” output of the digital comparator is connected to the asynchronous “Set” input of a DFF while the “Dec” output is connected to the asynchronous “Reset” input. This DFF then implements the “Up/Dn” output: the Q output becomes ‘1’ as soon as “Inc” is asserted, and becomes ‘0’ as soon as “Dec” is asserted. “OR_1” combines the two outputs “Inc” and “Dec” so that “Change’”” takes the state of ‘1’ when either one is asserted. A similar circuit like the one used in Fig. 7.7 is employed to generate a positive pulse every time “Change’”” is asserted, indicating a change in “D_in”. While “Up/Dn” is updated immediately after the assertion of “Inc” or “Dec”, it takes time for “Change’” to rise to ‘1’ due to the combined delay of “OR_1”, “S_2”, “OR_2”, “JK_1”, “Delay $_2$” and “OR_3”. The accumulator is therefore given enough time to update the output of its internal adder, so that when the pulse arrives at its “Change_in” input, there is a stable result ready to be latched to “D_out”.

Unlike the way an analog signal is digitized using a level-crossing ADC, where the shortest interval between consecutive samples is limited by the bandwidth of the analog signal, the CT digital signal to be delta-modulated is not only discontinuous but also very spiky, and as a result, the time spacing between consecutive change events can be infinitely small. However, due to the finite resolution of the memristor-based timing storage circuit discussed in CHAPTER 5, the storable interval length has a lower limit of 0.1 ms, beyond which error will increase significantly. Therefore, if nothing is done about the narrow spacing between consecutive pulses generated by the delta modulator, significant distortions will be caused in the following processing blocks due to their inabilities to handle change events very closely spaced in time.

A suspension control unit is therefore designed to address this issue. The basic idea is to suspend the operation of the digital comparator for a period of 0.1 ms every time a new change event occurs. This ensures the spacing between consecutive pulses generated will be no shorter than 0.1 ms, as no change events will be detected unless the digital comparator is enabled. Unlike those inverter-chain-based delay blocks commonly used in asynchronous digital circuits for delay matching, it is more efficient to adopt the memristor-based delay implementation in the design of the suspension control unit since the 0.1-ms suspension interval is too much longer than the delay of an inverter.

The schematic of the suspension control unit is shown in Fig. 7.10. Both "JK1" and "JK2" are connected as TFFs and both have been cleared upon startup. Therefore, when this suspension control unit is not in operation, the cleared "Suspend" will keep the current mirror off through M_7, while the asserted "$\overline{\text{Suspend}}$" will power down the comparator, reducing the standby power consumption to a minimum level. Both "Up" and "Down" will remain '0' since "Suspend" takes the state of '0', which ensures all four switches around the memristor are properly turned off so that any leakage current will not affect the memristance value during the idling state.

When the first pulse is received from the "Change" input, the positive transition will pass through the OR gate and toggle "JK1". The asserted "Suspend"

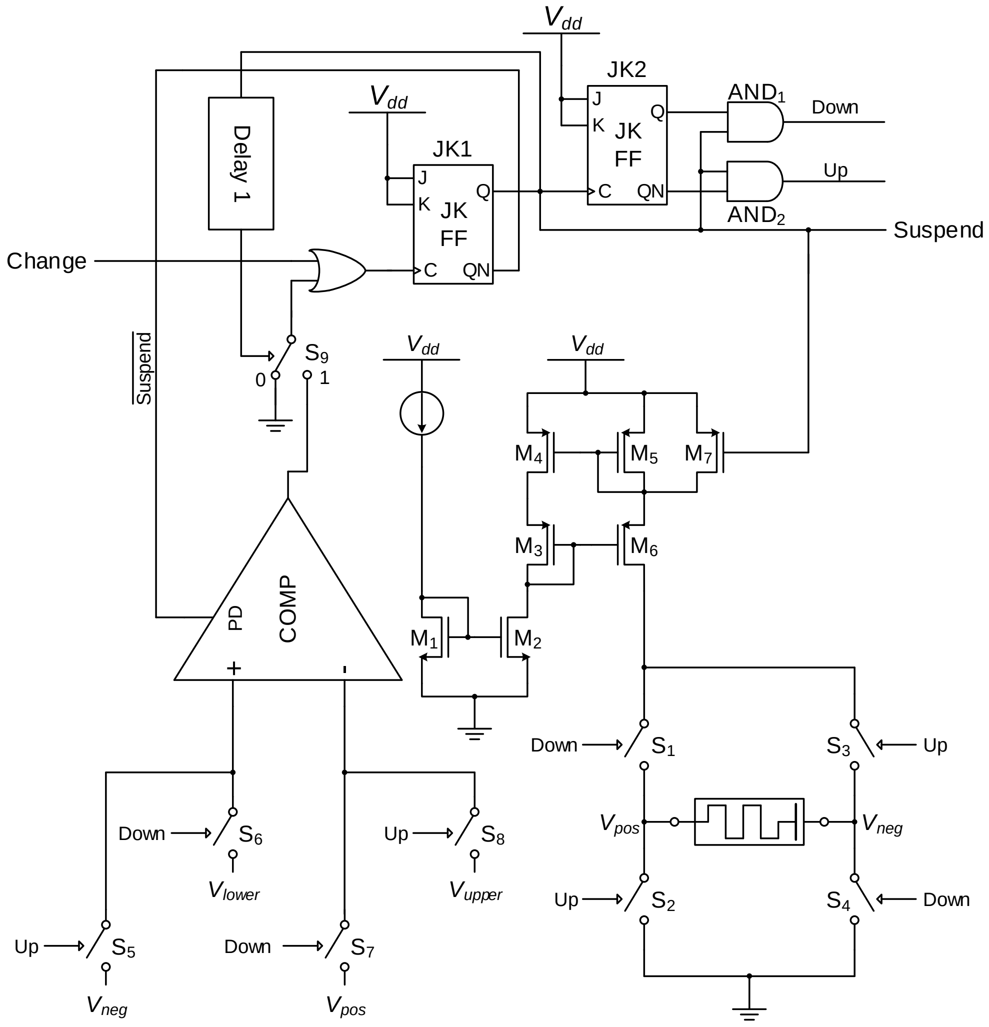

Fig. 7.10 Schematic of a memristor-based suspension control unit.

will turn on the current mirror and the cleared “$\overline{\text{Suspend}}$” will allow the comparator to wake up. At the same time, the positive transition in “Suspend” will toggle JK2 to make its “Q” output ‘1’ and “QN” output ‘0’. As a result, “Down” will settle to the state of ‘1’ while “Up” will settle to the state of ‘0’, which will then turn on switches “S_1” and “S_4” while “S_2” and “S_3” remains off. This will allow the current from the current mirror to pass through the memristor through its anode and causing its memristance to start decreasing. Meanwhile, the asserted “Down” turns on switches “S_6” and “S_7” while the cleared “Up” keeps “S_5” and “S_8” off. The voltage V_{pos} at the anode of the memristor will then be connected to the negative input of the comparator to be compared against a reference voltage V_{lower}, which is connected to the positive input of the comparator. As soon as V_{pos} drops below V_{lower}, the comparator output will become ‘1’. This positive transition will then propagate through “S_9” and the OR gate to toggle “JK1” again. The current mirror will be turned off as “Suspend” becomes ‘0’ and the comparator will also be powered down as “$\overline{\text{Suspend}}$” becomes ‘1’, which marks the end of a suspension procedure. A small delay block “Delay 1” is inserted to give time for the comparator to wake up, so that its output will have settled to the right state of ‘0’ by the time it is connected to the input of the OR gate through switch “S_9”. When the next “Change” pulse arrives to start a new suspension procedure, “Up” will be ‘1’ and “Down” will be ‘0’, which causes the current to pass through the memristor from its cathode, causing its memristance to increase. The increasing voltage V_{neg} at the cathode of the memristor will then be compared against a different reference voltage V_{upper}. The process will reach its end when V_{neg} rises above V_{upper}.

A more sensitive memristor is used in this suspension control unit as compared to the one used in the timing storage circuit discussed in CHAPTER 5, which gives rise to a higher change rate in the memristance value. This magnifies the change in V_{pos} or V_{neg} between the start and the end of a suspension process, which helps to reduce variations in the suspension interval given a finite comparator resolution. The parameters values of the memristor model used in this suspension control unit is summarized in Table 7.6.

Table 7.6 Parameter values for memristor model

Parameter	Quantity	Value
R_{ON}	Minimum memristance	90.4 kΩ
R_{OFF}	Maximum memristance	22.1 MΩ
μ_V	Ion drift mobility	5×10^{-14} $\mathrm{m}^2/(\mathrm{V}\cdot\mathrm{s})$
D	Device thickness	1.13 nm

When this suspension control unit is integrated into the delta modulator shown in Fig. 7.8, it will be used to control the switch “S_1” at the “EN” input of the digital comparator. When the first “Change” pulse is generated, “Suspend” will be asserted immediately and cause Switch “S_1” to change its connection from the output of “Delay 4” to ground. Meanwhile, “JK2” which has been preset upon startup will be cleared, which will then cause switch “S_3” to change its connection from ground to V_{dd}. “Delay 4” has the matched response delay of the suspension control unit, and is inserted to ensure the “EN” input of the digital comparator never sees the state of ‘1’ during these transitions at “S_1” and “S_3”. Upon completion of the first suspension, “Suspend” will be brought back to ‘0’ and the digital comparator will be enabled for the first time, which will then start detecting changes in “D_{in}” by comparing it with the accumulator output. Every time a change is detected, a pulse will be sent to the accumulator to have its output updated based on the state of “Up/Dn”. At the same time, “Suspend” will be asserted to disable the digital comparator for a period of about 0.1 ms. This process will then be repeated for every new change detected in “D_{in}”.

7.2.6 Cascading CT FIR Filters with Delta Modulators

As mentioned in Section 7.2.5, it is very important to properly initialize those blocks following delta modulators, as the “Change” pulses and “Up/Dn” states only tell about changes in the signal relative to its previous value without carrying information about the absolute levels. The delta modulator has been carefully designed to record the initial value it starts at. However, some precautions still need to be taken in the time when the delta modulator is to be enabled and the way the initial value is to be used in the following blocks.

Fig. 7.11 shows the schematic of the 15-tap masking filter $H_{mc}(z)$. As mentioned in Section 7.2.5, the lower 7 bits of the output of the preceding stage will be truncated, so “D_{in}” of “Delta Modulator 2” will only take the higher 13 bits. “Delta Modulator 2” should not start its operation until the output of $H_c(z)$ has settled, and since the output of $H_c(z)$ is derived from the difference between the delayed input and the output of $H_b(z)$, “Delta Modulator 2” should simply wait for the output of $H_b(z)$ to settle. With a total of 27 taps, the output of $H_b(z)$ settles only when the 27th delay block starts reproducing, and this is signaled by the assertion of “C26” from the 5-bit counter that controls the starting sequence of all 27 delay blocks. For this reason, “C26” from $H_b(z)$ is used to set “JK_DM” (cleared upon startup), which then enables “Delta Modulator 2”. A small delay block “Delay 1” is inserted to match the response delay of the memristor-based delay block, the delay of the summation block at the end of $H_b(z)$, and the delay of the subtractor used to generate the output of $H_c(z)$, so that it is ensured the output of $H_c(z)$ will have settled by the time “Delta Modulator 2” is enabled.

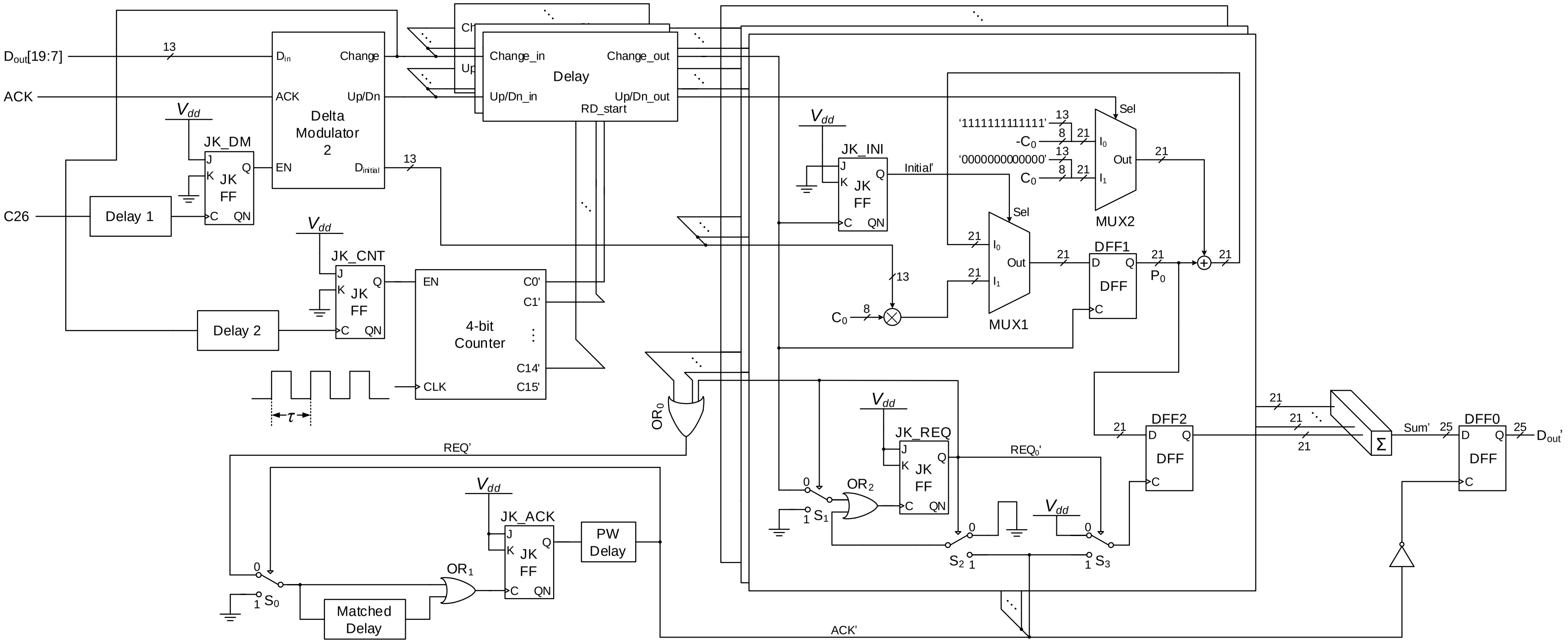

Fig. 7.11 Schematic of the 15-tap CT FIR masking filter $H_{mc}(z)$.

The “$D_{initial}$” output of “Delta Modulator 2” is fed into each of the 15 taps to be multiplied with the corresponding tap coefficients C_n to get the right initial product P_n that will be latched by “DFF1” at the arrival of the first “Change” pulse. The first “Change” pulse generated from “Delta Modulator 2” is used to set “JK_CNT” (cleared upon startup), which then enables the 4-bit counter that controls the starting sequence of the 15 delay blocks. A small delay block “Delay 2” is inserted to match the combined delay of the multiplier and “MUX1”, so that it is ensured the initial product P_0 of the first tap will have been settled to the right value by the time the first delay block starts reproducing. The remaining part of the filter works in the same way as described in CHAPTER 6.

As explained in Section 7.2.3, $H_{mb}(z)$ is simply implemented as a memristor-based delay block and an accumulator since no masking is required for the frequency response of $H_b(z)$. In order to match the group delay of $H_{mb}(z)$ with that of $H_{mc}(z)$, the delay block in $H_{mb}(z)$ should start reproducing at the same time when the middle delay block (the 8th among a total of 15) in $H_{mc}(z)$ starts reproducing. For this reason, “C7’” of the 4-bit counter in Fig. 7.11 is used to trigger “RD_start” of the delay block in $H_{mb}(z)$. A 13-bit accumulator with the same structure shown in Fig. 7.6 will be used to demodulate the delayed signals to generate the output of $H_{mb}(z)$. The “$D_{initial}$” output from “Delta Modulator 1” will be fed into the “I_1” input of “MUX1” within the accumulator for proper initialization.

7.2.7 Simulation Results

The smoothed ECG signal obtained in Section 6.2.3 was used to test the performance of this CT FRM high-pass filter. The original signal was first passed through a 7-bit delta modulator before fed into this filter. The filtered waveform is plotted in Fig. 7.12. As compared to the original signal, the baseline wandering noise is successfully suppressed. The entire waveform also shifts upward due to the removal of the DC component, which carries no useful information for analyzing ECG signals.

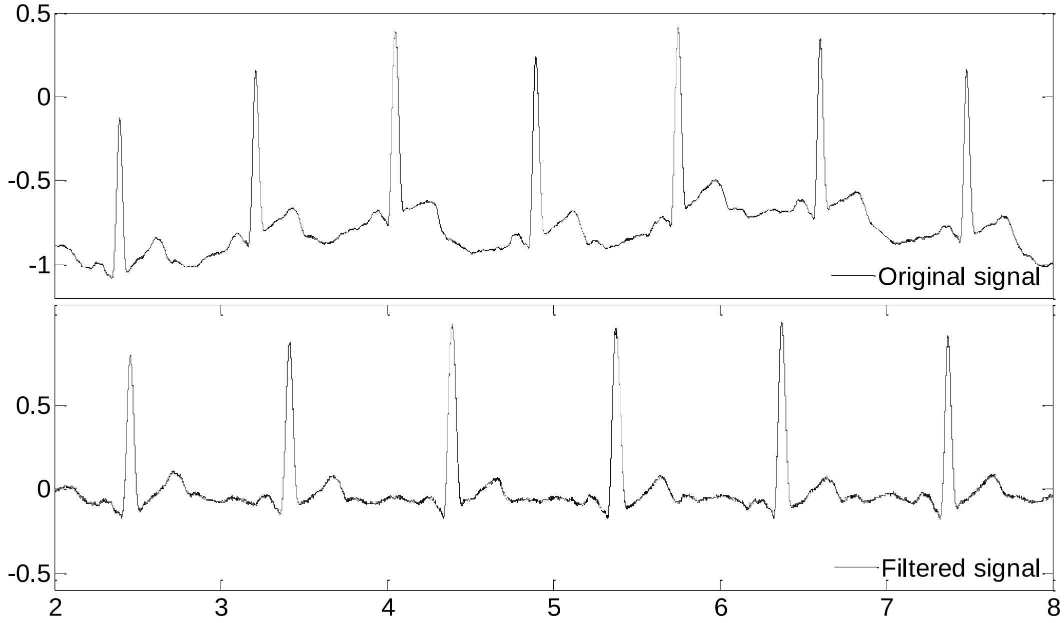

Fig. 7.12 ECG signal before and after high-pass filtering.

This filter consumes a total of 28.0 µW of power. Had the FRM technique not been used, the 168th-order CT high-pass filter capable of achieving the same frequency response would have consumed approximately 113 µW of power. A reduction of 75.2% is therefore achieved. Table 7.7 shows a breakdown of the power consumed by each block within the filter.

Table 7.7 Power consumption of each block within the filter

Block	**Power**
$H_b(z)$	18.1 μW
Accumulator	4.89 nW
Subtractor	149 nW
Delta Modulator 1	185 nW
Delta Modulator 2	147 nW
$H_{mb}(z)$	580 nW
$H_{mc}(z)$	8.75 μW
Adder	68.7 nW
Total	**28.0 μW**

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

In this work, new nonuniformly sampled digital signal processing approaches have been developed, and improvements have also been made on existing ones to make them more suitable for low-power biomedical applications.

A study on the four main categories of signal processing systems has first been made to show why conventional uniform DSP is not an energy-efficient choice for biomedical signals with long periods of inactivity.

To take advantage of such statistical properties of biomedical signals, a new signal processing scheme combining level-crossing sampling and conventional uniform DSP with the aid of linear interpolation has been presented. An example has been shown that a system designed using this processing scheme was able to achieve 88.8% reduction in the sampling rate and 92.6% reduction in the order of the filter. Designed using a 0.35-μm technology, the linear interpolator for this system consumed an average power of 12.1 μW under a 3.3-V supply.

A literature review of CT DSP has then been conducted. With signal dependent power consumption not only in the digitization part, but also in the processing part, CT DSP was believed to be an ideal choice for biomedical signals. However, the inability of signal storage and power consuming delay implementation were the two main obstacles to its adoption in biomedical applications.

By making use of the memory effect of memristors, a timing storage circuit has been proposed to allow the recording and reproducing of CT digital signals, which extends the benefits of CT DSP to applications that require signal storage. Various design considerations and practical challenges have been analyzed in details. Circuit simulation verifies the feasibility of this approach.

More importantly, it has been proven that the delay blocks in CT DSP systems can also be replaced by the proposed timing storage circuits, enabling significant power and area saving for low-frequency biomedical applications. An ECG signal processing example using the proposed method achieved more than 20% power saving compared to the current state-of-the-art CD DSP system implementations, without even considering the much older process and higher supply voltage used. With a 0.35-µm process, a 15-tap CT FIR filter designed using this method consumed an average power of 6.196 µW under a 3.3-V supply.

Lastly, the tunability of the proposed memristor-based delay implementation also enables the use of FRM techniques in designing sharp-transition CT FIR filters with reduced filter orders. A delta modulator was proposed to allow for the first time the cascading of CT FIR filters that operate on delta-modulated signals. As an example, a CT FRM high-pass filter with a combined order of 40 was designed using the same 0.35-µm process. This filter consumed a total power of 28.0 µW under the same 3.3-V supply, which is about 75.2% lower than the power that would be consumed by a 168th-order filter capable of achieving the same frequency response specifications.

8.2 Future Work

Due to the immaturity of memristor fabrication technologies, the designs proposed in this work remain yet to be verified through real circuit implementations. Although various nonidealities and practical issues haven been taken into account in my simulations, it will be interesting to see how well a memristor-based CT DSP system would perform as a real physical implementation.

In addition, it will be even more exciting to see how such a system is to be integrated with other blocks, such as the sensor frontend amplifier, the radio-frequency transmitter and receiver to form a complete ultra-low-power device for the preventive healthcare system mentioned at the beginning of this dissertation.

The rising demand in healthcare resources and facilities have attracted increasing investment and research in biomedical sciences and technologies, which paves the path to an entirely new healthcare concept. With more and more innovations being made, we can expect the preventive healthcare system may soon become reality, bringing better health and quality of life in the years to come.

REFERENCES

[1] A. Lymberis and D. D. Rossi, *Studies in health technology and informatics*: IOS Press, 2004.

[2] R. F. Yazicioglu, *et al.*, "Ultra-low-power biopotential interfaces and their applications in wearable and implantable systems," *Microelectronics Journal*, vol. 40, pp. 1313-1321, 2009.

[3] Y. Lian and X. Zou, "Towards self-powered wireless biomedical sensor devices," in *Proc. 2008 IEEE Int. Conf. Solid-State and Integrated-Circuit Technology (ICSICT'08)*, 2008, pp. 1556-1559.

[4] I. Korhonen, J. Parkka, and M. van Gils, "Health monitoring in the home of the future," *IEEE Engineering in Medcine and Biology Magzine*, vol. 22, no. 3, pp. 66-73, 2003.

[5] U. Anliker, *et al.*, "AMON: a wearable multiparameter medical monitoring and alert system," *IEEE Trans. Information Technology in Biomedicine*, vol. 8, no. 4, pp. 415-427, 2004.

[6] K. Lorincz, *et al.*, "Sensor networks for emergency response: challenges and opportunities," *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 16-23, 2004.

[7] E. Jovanov, A. Milenkovic, C. Otto, and P. C. d. Groen, "A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation," *Journal of NeuroEngineering and rehabilitation*, 2005.

[8] R. Paradiso, G. Loriga, and N. Taccini, "A wearable health care system based on knitted integrated sensors," *IEEE Trans. Information Technology in Biomedicine*, vol. 9, no. 3, pp. 337-344, 2005.

[9] J. Yao, R. Schmitz, and S. Warren, "A wearable point-of-care system for home use that incorporates plug-and-play and wireless standards," *IEEE Trans. Information Technology in Biomedicine*, vol. 9, no. 3, pp. 363-371, 2005.

[10] A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Review*, vol. 40, no. 1, pp. 1-12, 2010.

[11] M. J. Ramsay and W. W. Clark, "Piezoelectric energy harvesting for bio MEMS applications," in *Proc. of The International Society for Optical Engineering (SPIE'01)*, 2001, pp. 429-438.

[12] N. Ben Amor, *et al.*, "Energy harvesting from human body for biomedical autonomous systems," in *Proc. IEEE Sensors*, 2008, pp. 678-680.

[13] M. Koplow, A. Chen, D. Steingart, P. K. Wright, and J. W. Evans, "Thick film thermoelectric energy harvesting systems for biomedical applications," in *Proc. 5th Int. Summer School and Symp. Medical Devices and Biosensors*, 2008, pp. 322-325.

[14] Q. A. Khan and S. J. Bang. (2009). *Energy harvesting for self-powered wearable health monitoring system*.

[15] M. R. Mhetre, N. S. Nagdeo, and H. K. Abhyankar, "Micro energy harvesting for biomedical applications: a review," in *Proc. 3rd Int. Conf. Electronics Computer Technology (ICECT'11)*, 2011, pp. 1-5.

[16] R. Sarpeshakr, "Analog versus digital: extrapolating from electronics to neurobiology," *Neural Computation*, vol. 10, no. 7, pp. 1601-1638, 1998.

[17] Y. W. Li, K. L. Shepard, and Y. P. Tsividis, "Continuous-time digital signal processors," in *Proc. 2005 IEEE Int. Symp. Async. Circuits Syst. (ASYNC'05)*, 2005, pp. 138-143.

[18] P.-C. Huang, D. Macii, and J. M. Rabaey, "An information-theoretic framework for joint architectural and circuit level optimization for olfactory recognition processing," in *Proc. 2011 IEEE Workshop on Signal Processing Syst. (SiPS'11)*, 2011, pp. 19-24.

[19] S. K. Mitra, *Digital signal processing: a computer-based approach*, 4th ed.: McGraw-Hill Higher Education, 2011.

[20] J. Lies, *Digital signal processing – a Matlab based tutorial approach*: Hertfordshire Research Studies Press, 1996.

[21] B. Schell, "Continuous-time digital signal processors: Analysis and implementation," Ph.D. dissertation, Grad. School Arts Sci., Columbia Univ., New York, 2008.

[22] Y. Hong, I. Rajendran, and Y. Lian, "A new ECG signal processing scheme for low-power wearable ECG devices," in *Proc. 2011 Asia Pacific Conf. Postgraduate Research in Microelectronics and Electron. (PrimeAsia'11)*, 2011, pp. 74-77.

[23] Y. Hong, Z. Xie, and Y. Lian, "Wireless wearable ECG sensor design based on level-crossing sampling and linear interpolation," in *Proc. 2013 IEEE Int. Symp. Circuits Syst. (ISCAS'13)*, 2013, pp. 1300-1303.

[24] N. Sayiner, H. N. Sorensen, and T. R. Viswanathan, "A level-crossing sampling scheme for A/D conversion," *IEEE Trans. Circuits Syst. II: Analog and Digital Signal Process.*, vol. 43, no. 4, pp. 335-339, 1996.

[25] E. Allier, G. Sicard, L. Fesquet, and M. Renaudin, "A new class of asynchronous A/D converters based on time quantization," in *Proc. 9th IEEE Int. Symp. Async. Circuits Syst. (ASYNC'03)*, 2003, pp. 196-205.

[26] E. Allier, G. Sicard, L. Fesquet, and M. Renaudin, "Asynchronous level crossing analog to digital converters," *Measurement*, vol. 37, no. 4, pp. 296-309, 2005.

[27] S. d. Waele and P. M. T. Broersen, "A time domain error measure for resampled irregular data," in *Proc. 16th IEEE Instrumentation and Measurement Technology Conf.*, 1999, pp. 1172-1177.

[28] F. Aeschlimann, E. Allier, L. Fesquet, and M. Renaudin, "Asynchronous FIR filters: towards a new digital processing chain," in *Proc. 2004 IEEE Int. Symp. Asynchronous Circuits and Systems (ASYNC'04)*, 2004, pp. 198-206.

[29] S. M. Qaisar, L. Fesquet, and M. Renaudin, "Adaptive rate filtering for a signal driven sampling scheme " in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP'07)*, 2007, pp. 1465-1468.

[30] S. M. Qaisar, L. Fesquet, and M. Renaudin, "An improved quality filtering technique for time varying signals based on the level crossing sampling," in

Proc. Int. Conf. Signals and Electronic Systems (ICSES'08), 2008, pp. 355-358.

[31] S. M. Qaisar, L. Fesquet, and M. Renaudin, "Adaptive rate sampling and filtering based on level crossing sampling," *EURASIP J. Advances in Signal Processing,* 2009.

[32] L. Fesquet, G. Sicard, and B. Bidegaray-Fesquet, "Targeting ultra-low power consumption with non-uniform sampling and filtering," in *Proc. 2010 IEEE Int. Symp. Circuits Syst. (ISCAS'10),* 2010, pp. 3585-3588.

[33] J. W. Mark and T. D. Todd, "A nonuniform sampling approach to data compression," *IEEE Trans. Commun.,* vol. 29, no. 1, pp. 24-32, 1981.

[34] J. Foster and T.-K. Wang, "Speech coding using time code modulation," in *Proc. 1991 IEEE Southeastcon*, 1991, pp. 861-863.

[35] F. Akopyan, R. Manohar, and A. B. Apsel, "A level-crossing flash asynchronous analog-to-digital converter," in *Proc. 2006 IEEE Int. Symp. Async. Circuits Syst. (ASYNC'06),* 2006, pp. 11-22.

[36] A. Baums, U. Grunde, and M. Greitans, "Level-crossing sampling using micropocessor based system," in *Proc. 2008 IEEE Int. Conf. Signals and Electronic Systems (ICSES'08)*, 2008, pp. 19-22.

[37] P. C. Bagshaw and M. Sarhadi, "Analysis of samples of wideband signals taken at irregular, sub-Nyquist, intervals," *Electronics Letters,* vol. 27, no. 14, pp. 1228-1230, 1991.

[38] S. M. Qaisar, L. Fesquet, and M. Renaudin, "Spectral analysis of a signal driven sampling scheme," in *Proc. 14th European Signal Processing Conference (EUSIPCO'06)*, Florence, Italy, 2006.

[39] A. L. Goldberger, *et al.*, "PhysioBank, Physio Toolkit, and PhysioNet: components of a new research resource for complex physiologic signals," *Circulation,* vol. 101, no. 23, pp. e215-e220, 2000.

[40] E. Kreyszig, *Advanced engineering mathematics*: Wiley Eastern, 2006.

[41] Y. Tsividis, "Digital signal processing in continuous time: a possibility for avoiding aliasing and reducing quantization error," in *Proc. 2004 IEEE Int. Conf. Acoust., Speech, and Signal Process*, 2004, pp. 589-592.

[42] B. Schell and Y. Tsividis, "Analysis of continuous-time digital signal processors," in *Proc. 2007 IEEE Int. Symp. Circuits Syst. (ISCAS'07),* 2007, pp. 2232-2235.

[43] B. Schell and Y. Tsividis, "A continuous-time ADC/DSP/DAC system with no clock and with activity-dependent power dissipation," *IEEE J. Solid-State Circuits,* vol. 43, no. 11, pp. 2472-2481, 2008.

[44] B. Schell and Y. Tsividis, "Analysis and simulation of continuous-time digital signal processors," *Signal Process.,* vol. 89, no. 10, pp. 2013-2026, 2009.

[45] M. Kurchuk and Y. Tsividis, "Signal-dependent variable-resolution clockless A/D conversion with application to continuous-time digital signal processing," *IEEE Trans. Circuits Syst. I, Reg. Papers,* vol. 57, no. 5, pp. 982-991, 2010.

[46] H. Inose, T. Aoki, and K. Watanabe, "Asynchronous delta-modulation system," *Electronics Letters,* vol. 2, no. 3, pp. 95-96, 1966.

[47] Y. Tsividis, "Event-driven data acquisition and digital signal processing - a tutorial," *IEEE Trans. Circuits Syst. II, Exp. Briefs,* vol. 57, no. 8, pp. 577-581, 2010.

[48] L. O. Chua, "Memristor – the missing circuit element," *IEEE Trans. Circuit Theory,* vol. 18, no. 5, pp. 507-519, 1971.

[49] Y. Ho, G. M. Huang, and P. Li, "Nonvolatile memristor memory: device characteristics and design implications," in *Proc. 2009 IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD'09)*, 2009, pp. 485-490.

[50] Y. Ho, G. M. Huang, and P. Li, "Dynamic properties and design analysis for nonvolatile memristor memories," *IEEE Trans. Circuits Syst. I, Reg. Papers,* vol. 58, no. 4, pp. 724-736, 2011.

[51] H. Kim, M. P. Sah, C. Yang, and L. O. Chua, "Memristor-based multilevel memory," in *Proc. 2010 Int. Workshop Cellular Nanoscale Networks and Their Applications (CNNA'10)*, 2010, pp. 1-6.

[52] M. Laiho and E. Lehtonen, "Arithmetic opeartions within memristor-based analog memory," in *Proc. 2010 Int. Workshop Cellular Nanoscale Networks and Their Applications (CNNA'10)*, 2010, pp. 1-4.

[53] C. E. Merkel, N. Nagpal, S. Mandalapu, and D. Kudithipudi, "Reconfigurable N-level memristor memory design," in *Proc. 2011 Int. Joint Conf. Neural Networks (IJCNN'11)*, 2011, pp. 3042-3048.

[54] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature,* vol. 453, pp. 80-83, 2008.

[55] M. Mahvash and A. C. Parker, "A memristor SPICE model for designing memristor circuits," in *Proc. 2010 IEEE Int. Midwest Symp. Circuits Syst. (MWSCAS)*, 2010, pp. 989-992.

[56] G. R. Wilson, "A monolithic junction FET—n-p-n operational amplifier," *IEEE J. Solid-State Circuits,* vol. SC-3, no. 4, pp. 341-348, 1968.

[57] Y. W. Li, K. L. Shepard, and Y. P. Tsividis, "A continuous-time programmable digital FIR filter," *IEEE J. Solid-State Circuits,* vol. 41, no. 11, pp. 2512-2520, 2006.

[58] B. Schell and Y. Tsividis, "A clockless ADC/DSP/DAC system with activity-dependent power dissipation and no aliasing," in *Proc. 2008 IEEE Int. Solid-State Circuits Conf. (ISSCC'08),* 2008, pp. 550-635.

[59] E. Burlingame and R. Spencer, "An analog CMOS high-speed continuous-time FIR filter," in *Proc. 26th European Solid-State Circuits Conf. (ESSCIRC'00)*, 2000, pp. 288-291.

[60] R. Richter and H.-J. Jentschel, "An analogue delay line for virtual clock enhancement in DDS," in *Proc. 26th European Solid-State Circuits Conf. (ESSCIRC'00)*, 2000, pp. 476-479.

[61] G. Kim, M.-K. Kim, B.-S. Chang, and W. Kim, "A low-voltage, low-power CMOS delay element " *IEEE J. Solid-State Circuits,* vol. 31, no. 7, pp. 966-971, 1996.

[62] B. Schell and Y. Tsividis, "A low power tunable delay element suitable for asynchronous delays of burst information," *IEEE J. Solid-State Circuits,* vol. 43, no. 5, pp. 1227-1234, 2008.

[63] M. Kurchuk and Y. Tsividis, "Energy-efficient asynchronous delay element with wide controllability " in *Proc. 2010 IEEE Int. Symp. Circuits Syst. (ISCAS'10)*, 2010, pp. 3837-3840.

[64] R. W. Schafer, "What is a Savitzky-Golay filter," *IEEE Signal Processing Magazine,* vol. 28, no. 4, pp. 111-117, 2011.

[65] L. Rabiner, J. H. McClellan, and T. W. Parks, "FIR digital filter design techniques using weighted Chebyshev approximation," *Proc. IEEE,* vol. 63, no. 4, pp. 595-610, 1975.

[66] Y. C. Lim, "Frequency-response masking approach for the synthesis of sharp linear phase digital filters," *IEEE Trans. Circuits Syst.,* vol. 33, no. 4, pp. 357-364, 1986.

[67] R. Yang, B. Liu, and Y. C. Lim, "A new structure of sharp transition FIR filters using frequency-response masking," *IEEE TRans. Circuits Syst.,* vol. 35, no. 8, pp. 955-966, 1988.

[68] Y. C. Lim and Y. Lian, "The optimum design of one- and two-dimensional FIR filters using the frequency response masking technique," *IEEE Trans. Circuits Syst. II: Analog and Digital Signal Process.,* vol. 40, no. 2, pp. 88-95, 1993.

[69] Y. C. Lim and Y. Lian, "Frequency-response masking approach for digital filter design: complexity reduction via masking filter factorization," *IEEE Trans. Circuits Syst. II: Analog and Digital Signal Process.,* vol. 41, no. 8, pp. 518-525, 1994.

[70] Y. Lian, "Design of discrete valued coefficient FIR filters using frequency response masking," in *Proc. 6th IEEE Int. Conf. Electronics, Circuits and Systems (ICECS'99)*, 1999, pp. 253-255.

[71] T. Saramaki and Y. C. Lim, "Use of the Remez algorithm for designing FIR filters utilizing the frequency-response masking approach " in *Proc. 1999 IEEE Int. Symp. Circuits Syst. (ISCAS'99)*, 1999, pp. 449-455.

[72] Y. Lian, "FPGA implementation of high speed multiplierless frequency response masking FIR filters " in *Proc. 2000 IEEE Workshop on Signal Processing Syst. (SiPS'00)*, 2000, pp. 317-325.

[73] Y. Lian, "A new frequency-response masking structure with reduced complexity for FIR filter design," in *Proc. 2001 IEEE Int. Symp. Circuits Syst. (ISCAS'01),* 2001, pp. 609-612.

[74] Y. Lian and J. H. Yu, "The reduction of noises in ECG signal using a frequency response masking based FIR filter," in *Proc. 2004 IEEE Int. Biomedical Circuits. Syst.*, 2004, pp. 17-20.

[75] Y. C. Lim and R. Yang, "On the synthesis of very sharp decimators and interpolators using the frequency-response masking technique," *IEEE Trans. Signal Processing,* vol. 53, no. 4, pp. 1387-1397, 2005.

[76] W. R. Lee, L. Caccetta, K. L. Teo, and V. Rehbock, "A unified approach to multistage frequency-response masking filter design using the WLS technique," *IEEE Trans. Signal Processing,* vol. 54, no. 9, pp. 3459-3467, 2006.