

# 國立交通大學

電子工程學系 電子研究所

碩士論文

用於行動照護應用之低能量同步非同步混合式  
心電訊號特徵擷取器設計

An Energy-Efficient  
Mixed Sync-Async Cardiac Delineator  
for Mobile Healthcare Applications

研究生：張博堯

指導教授：李鎮宜 教授

中華民國 102 年 01 月

用於行動照護應用之低能量同步非同步混合式  
心電訊號特徵擷取器設計

An Energy-Efficient  
Mixed Sync-Async Cardiac Delineator  
for Mobile Healthcare Applications

研究生：張博堯 Student: Po-Yao Chang

指導教授：李鎮宜 Advisor: Chen-Yi Lee

國立交通大學

電子工程學系 電子研究所

碩士論文

A Thesis

Submitted to Department of Electronics Engineering and  
Institute of Electronics

College of Electrical and Computer Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Electronics Engineering

January 2013

Hsinchu, Taiwan, Republic of China

中華民國 102 年 01 月

# 用於行動照護應用之 低能量同步非同步混合式心電訊號特徵擷 取器設計

學生：張博堯

指導教授：李鎮宜 博士

國立交通大學

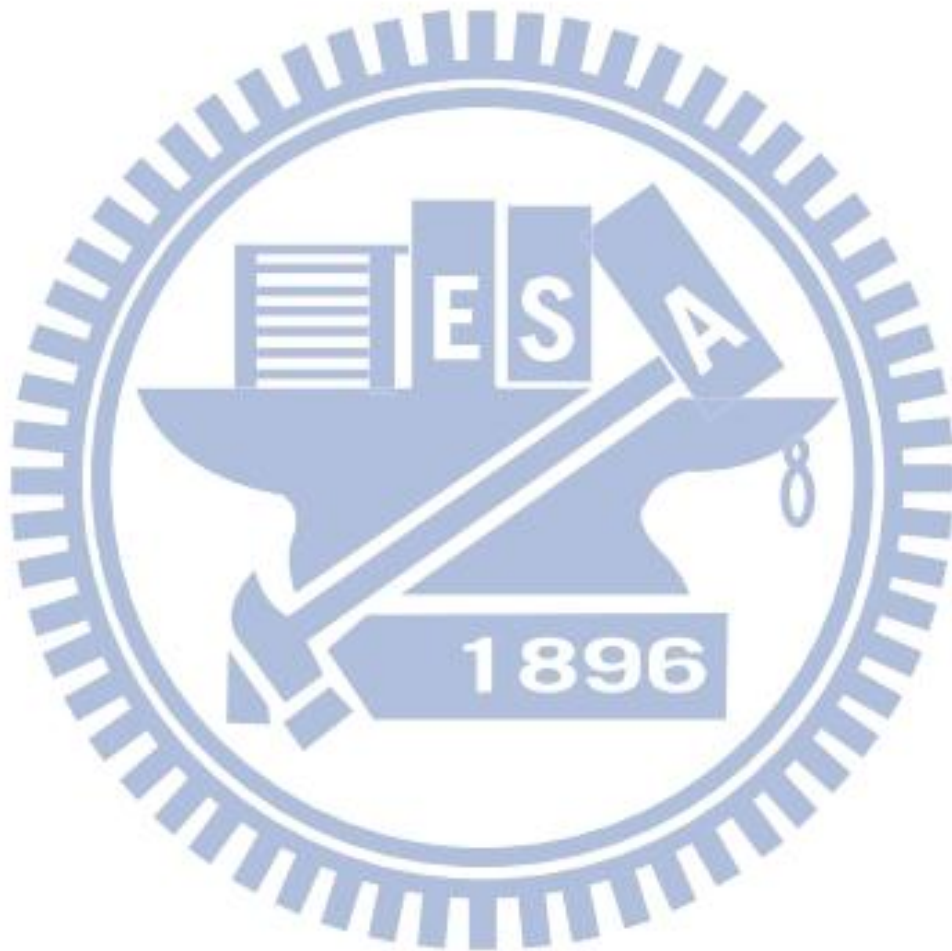
電子工程學系 電子研究所

## 摘要

行動照護應用使用了具備無線傳輸能力的感測器來達到生理狀況監控，能夠長時間的監控就成為了這類應用的主要需求。我們藉由在感測器上做訊號處理以減少無線傳輸的資料量。在感測器上做事先的特徵擷取，傳輸的資料將可以被大大地減少，同時擷取出的生理特徵也可被用於即時的疾病診斷，減少感測器到醫院的診斷延遲時間，給予病患更大的保護。

以心血管疾病為例，心臟疾病的診斷是藉由心電圖的 P、Q、R、S、T 等特徵的大小及區間來判斷。在這份論文中，我們提出了一個基於小波轉換的心電訊號特徵擷取演算法及硬體，並且使用兩個標準的心電圖資料庫來驗證特徵擷取的結果。我們提出的演算法對所有提供的特徵都達到了 99.4% 及 96.1% 以上的靈敏度及準確率。在硬體架構上，擷取器被設計在極低的操作頻率（250Hz），並藉由共用搜尋核心、記憶體最佳化、以及觸發式的獨立電源管理非同步搜尋核心，來達到低能量消耗的需求。極低的操作頻率加上非同步電路更提供了降低工作電壓來降低功率消耗的可能性。我們提出的心電訊號擷取器採用 90 奈米標準 CMOS 製成晶片實現。在 0.5V 的電源供應之下，擷取器功率消耗為 2.56 $\mu$ W。最後，我們

也利用了市售晶片以及嵌入式處理器建構了一個極小的無線傳輸感測器模組來驗證我們的特徵擷取演算法。藉由這個感測器模組，我們驗證了演算法的行動環境中的準確率以及藉由特徵擷取減少資料傳輸能量的想法。



# **An Energy-Efficient Mixed Sync-Async Cardiac Delineator for Mobile Healthcare Applications**

Student: Po-Yao Chang

Advisor: Chen-Yi Lee

Department of electronics engineering and Institute of electronics,  
National Chiao Tung University

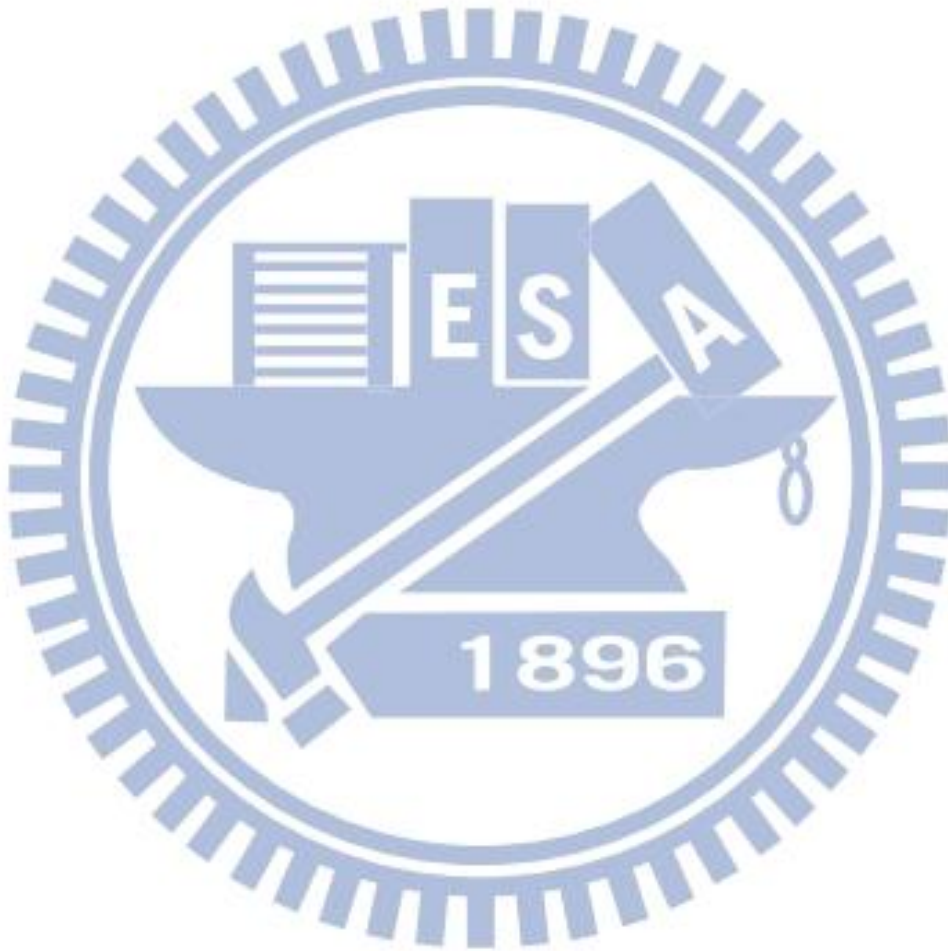
## ***Abstract***

Long-term monitoring is the key requirements for mobile healthcare applications, where the wireless sensor nodes are worn to record the human's vital signals. On-sensor signal analysis is proposed for these applications to enable timely detection of risky syndromes and extend the monitoring time. Instead of raw data transmission, the wireless transmission energy is reduced by only transmitting the vital features.

In case of cardiac diseases, the syndrome analysis is performed based on different extracted features of ECG signals like P, QRS<sub>on</sub>, R, QRS<sub>end</sub>, and T wave. In this work, an energy-efficient cardiac delineation algorithm based on multi-scale wavelet transform is designed together with its hardware implementation. The detection result is evaluated on two annotated databases including MIT-BIH arrhythmia database and QT database. The obtained sensitivity and positive predictivity are over 99.4% and 96.1% for the five ECG features, respectively. With shared search kernels, storage optimization and event-driven asynchronous search kernel with individual power management, the delineator can operate at 250Hz without the needs for additional high speed clock. The slow operating speed and asynchronous search kernel also enables further voltage scaling to reduce power. Implemented using UMC 90nm technology

and operating at 250Hz with 0.5V supply voltage, the overall power is 2.56 $\mu$ W for real-time ECG monitoring.

Besides, a miniaturized prototype wireless sensor is constructed using commercial products with on-sensor delineation. The prototype provides evidence to the delineation robustness for mobile monitoring and power reduction for on-sensor feature extraction.



# 誌謝

很高興能夠在 Si2 實驗室完成我的碩士學歷。

感謝李鎮宜老師從我專題到研究所以來，給予我在研究以及人生上的指導。實驗室完整的資源以及實力堅強的研究陣容讓我們在研究上能夠自由的發展。也感謝口試委員黃威教授及莊景德教授，能撥空來指導並且給予我一些研究方面的建議，使得本篇論文可以更加完整。

感謝在我的研究中一路給我指導的書餘學長，一路給我指導方和拉拔，讓我能夠順利完成碩士學位。也感謝所有 Si2 實驗室的夥伴們以及所有電工系及熱舞社的同學朋友，豐富了我在交大的生活。

最後感謝我的父母，謝謝你們從小的拉拔，沒有你們就沒有今天的我。謝謝姊姊，從小到大一直包容我，我很想你，我愛你們。

# Table of Contents

---

---

	Pages
Chapter 1: Introduction and Motivation.....	1
1-1 Introduction to Mobile Healthcare Application.....	1
1-2 Motivation .....	2
1-3 Introduction to Cardiac Signal.....	4
1-4 Organizations.....	9
Chapter 2: ECG Delineation Algorithm .....	10
2-1 Background.....	10
2-2 Dyadic Wavelet Transform (DWT) .....	12
2-2.1 Wavelet Theory.....	12
2-2.2 Quadratic Spline Wavelet Transform (QSWT).....	14
2-3 Detection Algorithm .....	16
2-3.1 Wave Characteristic and Detection Flow .....	16
2-3.2 R Peak Detection .....	19
2-3.3 QRS <sub>on/end</sub> Detection.....	20
2-3.4 P/T Wave Detection .....	21
2-3.5 Adaptive Threshold and Window Update .....	22
2-4 Simulation Result and Performance Evaluation.....	24
2-5 Summary.....	28
Chapter 3: Asynchronous Design.....	29
3-1 Motivation .....	29
3-2 Introduction to Asynchronous Circuit .....	30
3-2.1 Moving from Synchronous to Asynchronous Approach .....	30
3-2.2 Handshake Protocol.....	32



3-2.3	Muller C Element .....	34
3-2.4	Asynchronous Pipelines .....	35
3-3	Design Flow Using Commercial CAD Tool.....	37
3-3.1	Design flow .....	37
3-3.2	Delay Margin Tuning.....	40
3-4	Design Example: An 16-tap FIR Filter.....	44
3-4.1	Iterative FIR Architecture and Implementation.....	44
3-4.2	Performance Comparison .....	47
3-5	Summary.....	51
Chapter 4:	Architecture Design and Hardware Implementation.....	52
4-1	Architecture .....	52
4-2	Implementation Result.....	57
4-3	Comparison with State-of-The-Art.....	59
Chapter 5:	Prototype Construction .....	60
5-1	Experiment Platform and System .....	60
5-2	On-Sensor ECG Delineator .....	61
5-3	Emulation Result .....	64
Chapter 6:	Conclusion and Future Work.....	66
6-1	Conclusion .....	66
6-2	Future Work .....	67
Reference	68	

# List of Figures

---

---

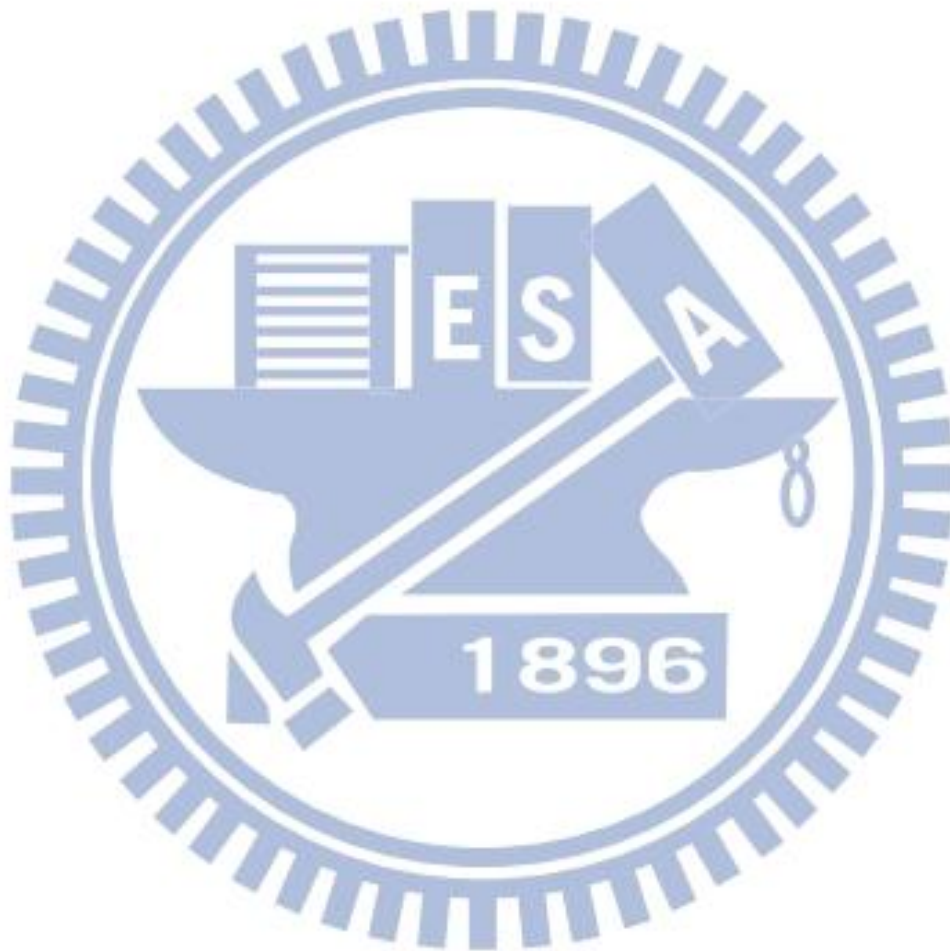
Fig. 1-1 Scenario for mobile healthcare application.....	1
Fig. 1-2 On-sensor delineation reduces Tx energy and provides real-time alarm.....	2
Fig. 1-3 The basic heart model and feautre waves inside a cardiac cycle.....	5
Fig. 1-4 P wave generation. ....	6
Fig. 1-5 AV node – the conduction blockage between atrial and ventricular. ....	7
Fig. 1-6 Ventricular contractionresulting in QRS complex .....	8
Fig. 1-7 The repolarization of ventricular results in T wave .....	8
Fig. 2-1 Morphological changes ECG signals extracted from QT database. ....	11
Fig. 2-2 (a) Mallat’s Algorithm, (b) <i>algorithm á trous</i> (SWT).....	14
Fig. 2-3 A general ECG signal and the corresponded frequency response.....	15
Fig. 2-4 The first 5 wavelet decomposition of ECG signal with noise coupling .....	17
Fig. 2-5 Flow graph for the proposed detection algorithm.....	18
Fig. 2-6 R peak detection detail.....	19
Fig. 2-7 $QRS_{on/end}$ detection detail.....	20
Fig. 2-8 Search window defined for P/T detection.....	21
Fig. 2-9 Detection result of ECGs with morphological changes and noise.....	24
Fig. 3-1 Delay variance under 1.0V and 0.5V supply voltage .....	29
Fig. 3-2 The difference between synchronous design and asynchronous design.....	31
Fig. 3-3 (a) 4-phase handshake protocol (b) 2-phase handshake protocol .....	33
Fig. 3-4 Muller C element .....	34
Fig. 3-5 Fork and join structures .....	35
Fig. 3-6 Muller pipeline.....	35
Fig. 3-7 MOUSETRAP pipeline.....	36
Fig. 3-8 A Verilog template for the Muller pipeline template .....	38

Fig. 3-9 Break timing loops for un-constraint path .....	39
Fig. 3-10 Time constraints .....	40
Fig. 3-11 The delay distribution of datapath and delay line at different corner case ...	41
Fig. 3-12 Tuning Circuit including the tunable delay line and a lead-lag detector .....	42
Fig. 3-13 (a) The 8 tuning steps at 3 corners (b) Reduced margin .....	43
Fig. 3-14 Block diagram for the 16-tap FIR filter .....	44
Fig. 3-15 Ring structure for 4-phase Muller pipeline .....	45
Fig. 3-16 The modified MOUSETRAP ring structure and time diagram .....	46
Fig. 3-17 Layout photo for the sync/async 16-tap FIR filter .....	47
Fig. 3-18 Energy distribution of the three asynchronous implementation .....	48
Fig. 3-19 Operation time/energy of the 3 async designs at corner and temperature ....	48
Fig. 3-20 Operation time/energy of sync/async design at different corners/temp .....	50
Fig. 4-1 Delineator architecture .....	52
Fig. 4-2 The state transition graph of the QRS FSM .....	53
Fig. 4-3 The architecture and pre-search process of $QRS_{on}$ .....	54
Fig. 4-4 The adaptive THR/WIN update engine .....	55
Fig. 4-5 Time diagram of the shared P/T search kernel .....	56
Fig. 4-6 The input and output interface for the asynchronous P/T kernel .....	57
Fig. 4-7 Power reduction with strategy at different design level .....	58
Fig. 4-8 Layout photo for the proposed cardiac delineator .....	58
Fig. 5-1 Experiment environment setup .....	60
Fig. 5-2 The prototype wireless sensor .....	61
Fig. 5-3 5 different Power mode supported by MPS430 micro-controller .....	62
Fig. 5-4 Delineation flow inside MSP430 .....	63
Fig. 5-5 ECG packet format .....	63

Fig. 5-6 SPI handshake protocol between MSP430 and G2 WIFI module..... 64

Fig. 5-7 User interface for real time display of ECG and extracted fiducial points..... 64

Fig. 5-8 Delineation Under mobile environment (with baseline drift)..... 65

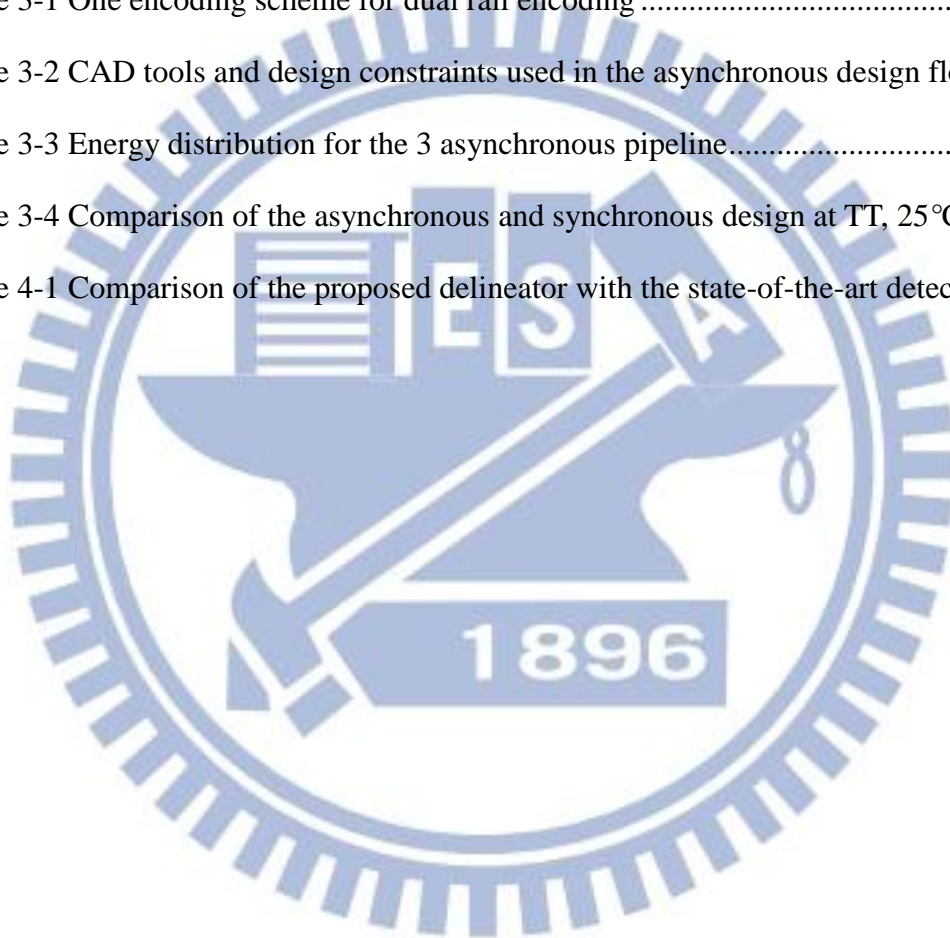


# List of Tables

---

---

Table 1-1 Syndromes supported by the provided P, QRS <sub>on</sub> , R, QRS <sub>end</sub> , P features.....	4
Table 2-1 R peak detection comparison with state-of-the-art detector using MITDB .	26
Table 2-2 Fiducial points delineation result comparison using QTDB .....	26
Table 2-3 R peak detection result within MITDB .....	27
Table 3-1 One encoding scheme for dual rail encoding .....	32
Table 3-2 CAD tools and design constraints used in the asynchronous design flow ...	37
Table 3-3 Energy distribution for the 3 asynchronous pipeline.....	48
Table 3-4 Comparison of the asynchronous and synchronous design at TT, 25°C .....	49
Table 4-1 Comparison of the proposed delineator with the state-of-the-art detector ...	59



# Chapter 1:

## Introduction and Motivation

---

### 1-1 Introduction to Mobile Healthcare Application

Mobile healthcare is defined broadly as the use of any mobile telecommunication technologies for the use of wireless health care delivery systems. Thanks to the developing of technology, healthcare services is no longer limited to patients in the hospital, but also extends to public anywhere and anytime using portable devices. These kinds of mobile healthcare devices target not only on high risk patients but also to general public, providing functions like long-term bio-signal recording, early syndrome detection (reducing time delay to hospital).

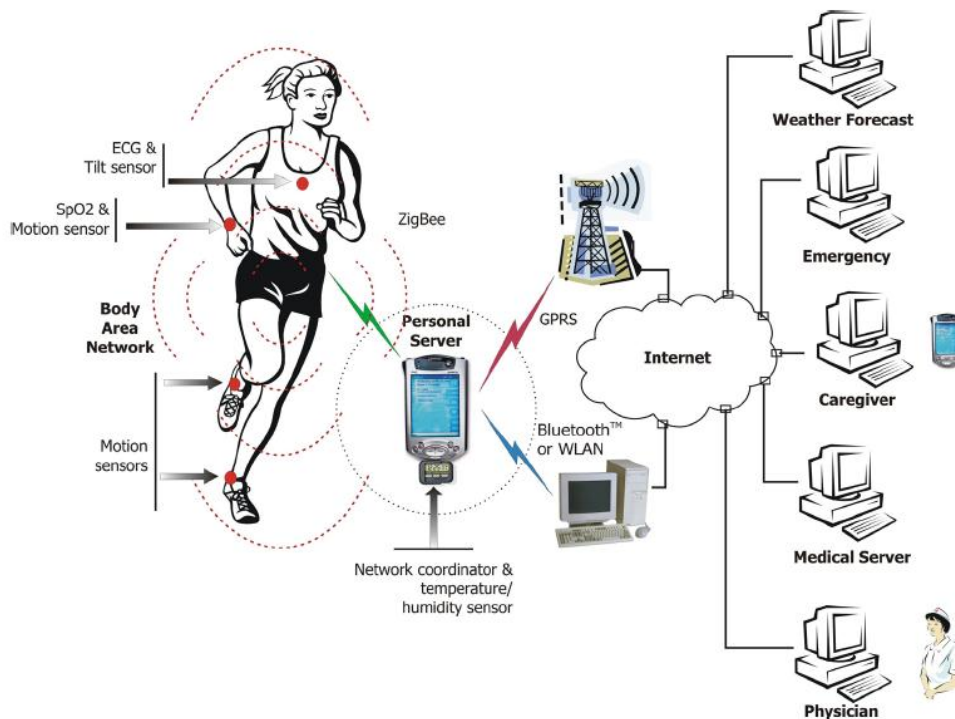


Fig. 1-1 Scenario for mobile healthcare application

Fig. 1-1 shows the scenario of mobile healthcare. With different kinds of wireless sensor attached, information about body conditions such as ECG, EEG signals, blood pressures, and motion acceleration can be collected and transmitted through existed wireless connection to the hospital server for further data analysis.

## 1-2 Motivation

The challenge for these kinds of applications is the limited battery power for wireless sensor nodes. To extend monitoring time, many different kinds of technologies are proposed. Some aiming for more efficient wireless transmission, some emphasized on signal pre-processing on sensor nodes to reduce the transmission data by compression or feature extraction. Taking cardiac signal as example, feature extraction extracts the vital features that are required for syndrome diagnosis. With only the vital features transmitted, large transmission energy can be reduced. Besides, with the collected features, syndrome analysis can also be performed on the sensor for early alarm. Fig. 1-2 shows the possible implementation of such ECG processor.

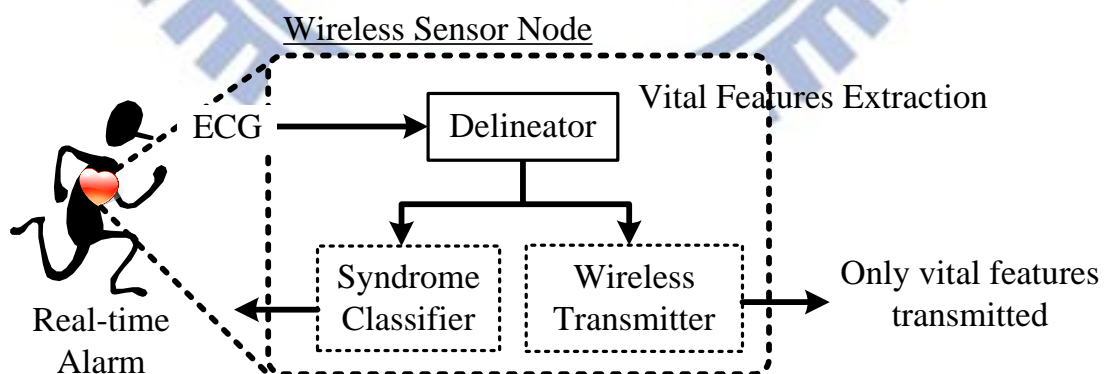


Fig. 1-2 By introducing on-sensor delineation, vital features can be extracted thus reducing transmission energy and providing real-time alarm to patients

To satisfy the application requirement, low power and high accuracy feature extractor is required. There have been many algorithms proposed. However, most of them are designed for off-line detection, which are inappropriate for low power implementation [9] [10]. A real-time QRS detector has been implemented [14], nevertheless, the power is too large for mobile healthcare application ( $>100\mu\text{W}$ ). Some low power hardware are proposed, but the detection accuracy is limited [16] or only the R peak is detected [16] [17] [18]. This limits the detection syndrome to only abnormal heart rate and confines the use for mobile healthcare applications. Therefore, we proposed a delineation algorithm with hardware implementation, which is able to detect the 5 most significant ECG features including P, QRS<sub>on</sub>, R, QRS<sub>end</sub>, and T wave. Table 1-1 lists the syndromes that can be detected with all 5 features. Various syndromes are supported, including high risk syndromes such as Myocardial Infarction, which cannot be detected by exist detectors.

To achieve further power reduction, the supply voltage for the delineator is scaled down. To combat the severe PVT variation under low supply voltage, we adopt asynchronous circuits to track the large delay variance for the critical design part. The most computation cost search P/T wave search kernel of the delineator is designed using a handshake template modified from the asynchronous 2-phase MOUSETRAP pipeline [21], while the rest part of the design can operate at low speed reducing the switching power. Because of the event-triggered property of P/T wave search, the search kernel can be power gated when idle. The use of asynchronous technique reduces the requirement for additional high speed clock source and exhibit fast power ON/OFF property suitable for such kind of design.



Table 1-1 Syndromes that can detected with the provided P, QRS<sub>on</sub>, R, QRS<sub>end</sub>, P features.

<b>P, QRS<sub>on</sub>, R, QRS<sub>end</sub>, T related syndrome</b>	
<i>Syndrome</i>	<i>Recognizable symptoms</i>
<b>Heart-rate</b> (Tachycardia, Bradycardia)	HR > 120bpm (Tachycardia) HR < 30bpm (Bradycardia)
<b>Ventricular Hypertrophy</b>	R amplitudes in different leads
<b>Supraventricular Arrhythmias</b> (PSVT, Atrial Flutter, Atrial Fibrillation, Multifocal/Paroxysmal atrial tachycardia)	Opposite P, regular/irregular HR, P morphologies
<b>Ventricular Arrhythmias (PVCs)</b>	QRS morphologies
<b>AV blocks</b>	PRI, PRI changes
<b>Bundle Branch Blocks</b>	QRS <sub>on</sub> -QRS <sub>end</sub> > 0.12sec, RSR', ST down, opposite T
<b>Preexcitation</b> (WPW, LGL)	PR < 0.12sec, wide QRS PR < 0.12sec, Normal QRS
<b>Myocardial Infarction</b>	ST rise, high T (maybe opposite)

### 1-3 Introduction to Cardiac Signal

This part introduces the basics of ECG signals. ECG is actually the voltage changes across the heart which can be measured by the sensor node attached to our skin. The basic waves in a cardiac cycle consist of the P, Q, R, S, and T waves. As the signals transmitted through the conducting cells triggering the myocardial cells according to all kinds of physical events, the voltage difference can be recorded and

analyzed. Fig. 1-3 (a) shows the basic waves of a general ECG in a cardiac cycle. Fig. 1-3 (b) shows a basic heart model including the conducting system. The mechanism of a heart beat is like this: the pacemaker cells (sinus node) perform the action of depolarizing and repolarizing at a certain frequency based on the status of sympathetic and the required amount of cardiac output. At every depolarizing and repolarizing, an action potential is generated, and this voltage change transmits to the myocardial cells through the help of the electrical conducting cells. Among receiving the depolarizing signal,  $Ca^{+}$  will be liberated into the myocardial cells resulting in contraction.

A general cardiac cycle consists of several wave characteristic as previously shown in Fig. 1-3 (a). . They are generated by electrical events such as:

- Atrial depolarization
- A pause separated the atria from the ventricles
- Ventricular depolarization
- Repolarization

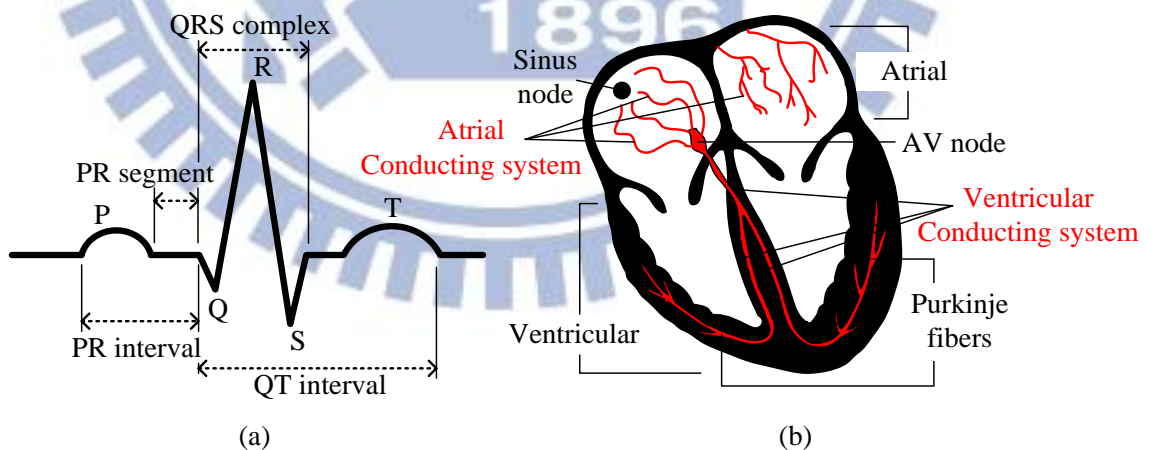


Fig. 1-3 (a) The basic waves inside a cardiac cycle. (b) A basic heart model including the conducting path

To begin, the activation of the sinus node generates a depolarization wave propagating to the myocardial cells of atrial, causing the atrial to contract. This event can be detected and is regarded as the P wave (Fig. 1-4). Because the sinus node is located at the right atrial, the right atrial will contract first. The first half of P wave is represented by the depolarization of the right atrial and the later part of P wave is represented the depolarization of the left atrial.

In general, an electrical gate called atrioventricular node (AV node) appears between the atrial and ventricular. This gate slows down the propagation of the depolarization signal causing a pause between the P wave and QRS complex as shown in Fig. 1-5. This physical delay exists to prevent ventricular contract before the bloods comes in.

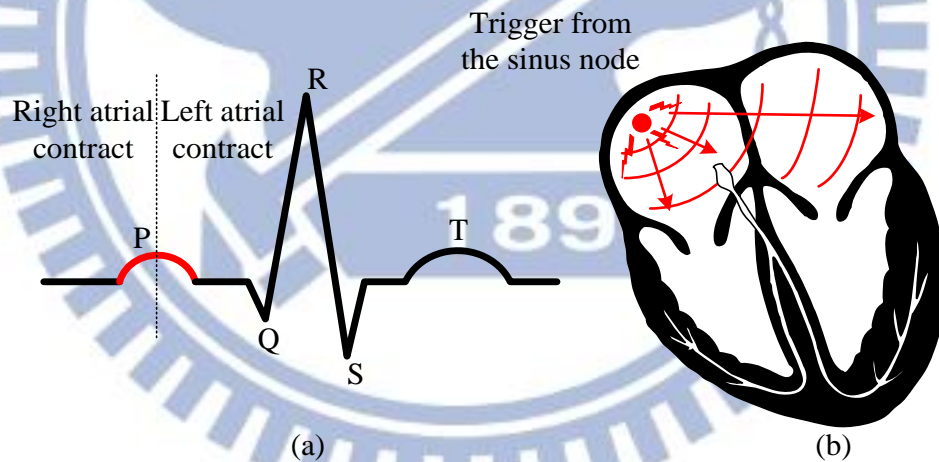


Fig. 1-4 The trigger from sinus node signals atrial to contraction result in P wave.

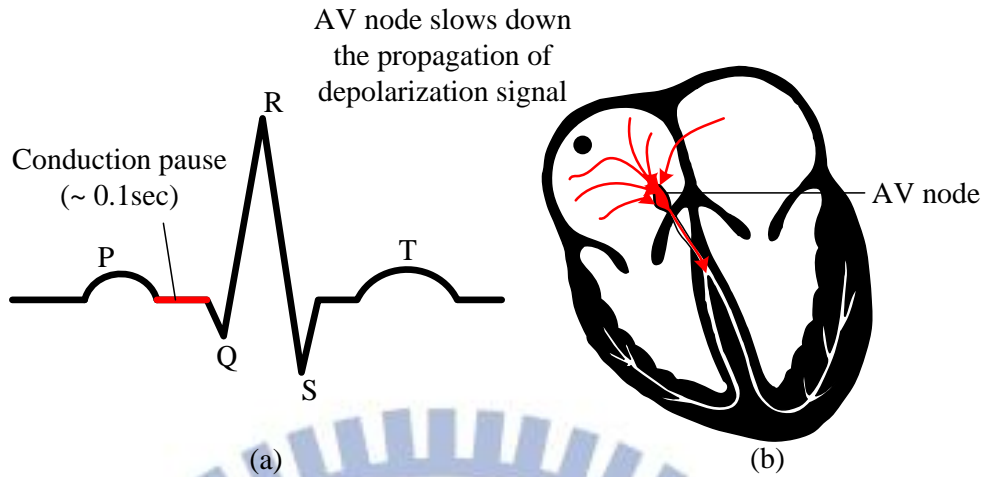


Fig. 1-5 The AV node slows down the depolarization signal result in a small pause between the contraction of atrial and ventricular.

After about 0.1 seconds of delay, the depolarization wave propagates through the AV node along the Purkinje fibers causing the ventricular to contract. This physical event results in a new transition on ECG signal and is regarded as the QRS complex. Because ventricular is usually larger and the network of Purkinje fibers is more complex than atrial, the amplitude of QRS complex is larger and shape varying (Fig. 1-6).

After depolarization of all the cells, there will be a time interval where no more contraction can be made, called the refractory period. During this period, cells repolarized in order for the trigger of the next depolarization. This repolarization process of ventricular also results in a wave called T wave (Fig. 1-7). Note that the repolarization of atrial also generates a wave. But because it occurs at the same time as ventricular depolarized, the wave is covered by the QRS complex.

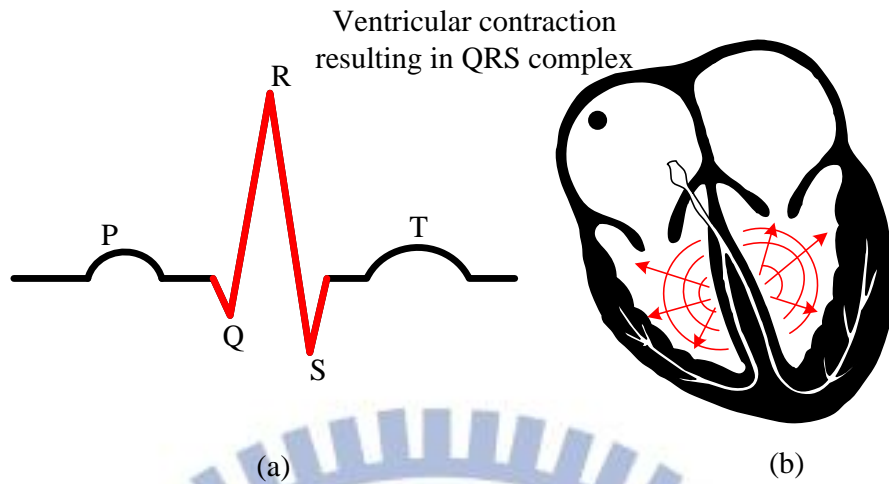


Fig. 1-6 Depolarization transmits through the Purkinje fibers causing ventricular to contract. The signal event showing on ECG is known as the QRS complex

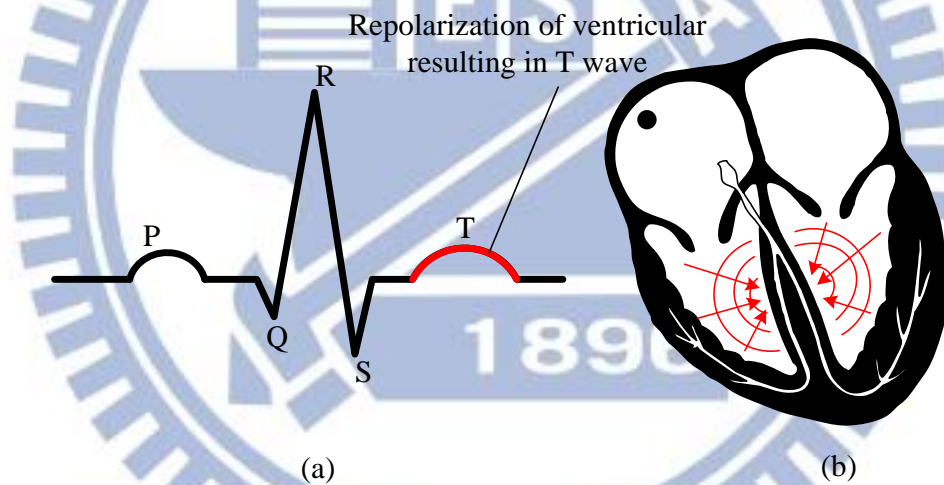


Fig. 1-7 The repolarization of ventricular results in T wave

## 1-4 Organizations

The thesis is organized as follows. Chapter 2 explains the proposed multi-scale wavelet-based delineation algorithm and the performance comparison with other existing algorithms verified using standard ECG databases. Chapter 3 presents the motivation for moving from synchronous to asynchronous circuit design. We first introduce the basics for asynchronous design. A 2-phase handshake protocol targeting especially for iterative computation is proposed. A design flow for such asynchronous design using commercial CAD tools, together with an example design of 16-tap FIR filter is shown in this chapter. Chapter 4 describes the architecture and hardware implementation of the proposed mixed sync-async cardiac delineator with low power techniques at different design level. A prototype wireless sensor is constructed using embedded micro-controller with on-sensor delineation to verify the delineation for real mobile environment in chapter 5. Finally, chapter 6 gives the conclusion and future work.

# *Chapter 2:*

## *ECG Delineation Algorithm*

---

### 2-1 Background

ECG is the transthoracic interpretation of the electrical activity of heart over a period of time. A typical ECG tracing of the cardiac cycle (heartbeat) consists of a P wave, a QRS complex, and a T wave. The most commonly used features, which delineates the ECG waveform, are the signal amplitude and intervals within P, QRS<sub>on</sub>, R, QRS<sub>end</sub>, T wave boundaries.

The detection of these features is challenging for several reasons:

- Because of the small amplitude of ECG signal (<5mV), it is usually coupled with noise and artifacts, such as power line interference, electrode contact noise, patient-electrode motion artifacts, Electromyography (EMG), baseline wandering, data collecting device noise, quantization noise and aliasing, etc.
- The wide variation of QRS morphologies and rhythms, from abnormal ECGs and interpersonal variations.

Fig. 2-1 shows some ECG signals extracted form QT database including a broad range of QRS and ST-T variety to show the morphological changes. Some non-ideal effect for mobile measurement including muscular noise, motion artifact, amplitudes changes are also shown in these examples.

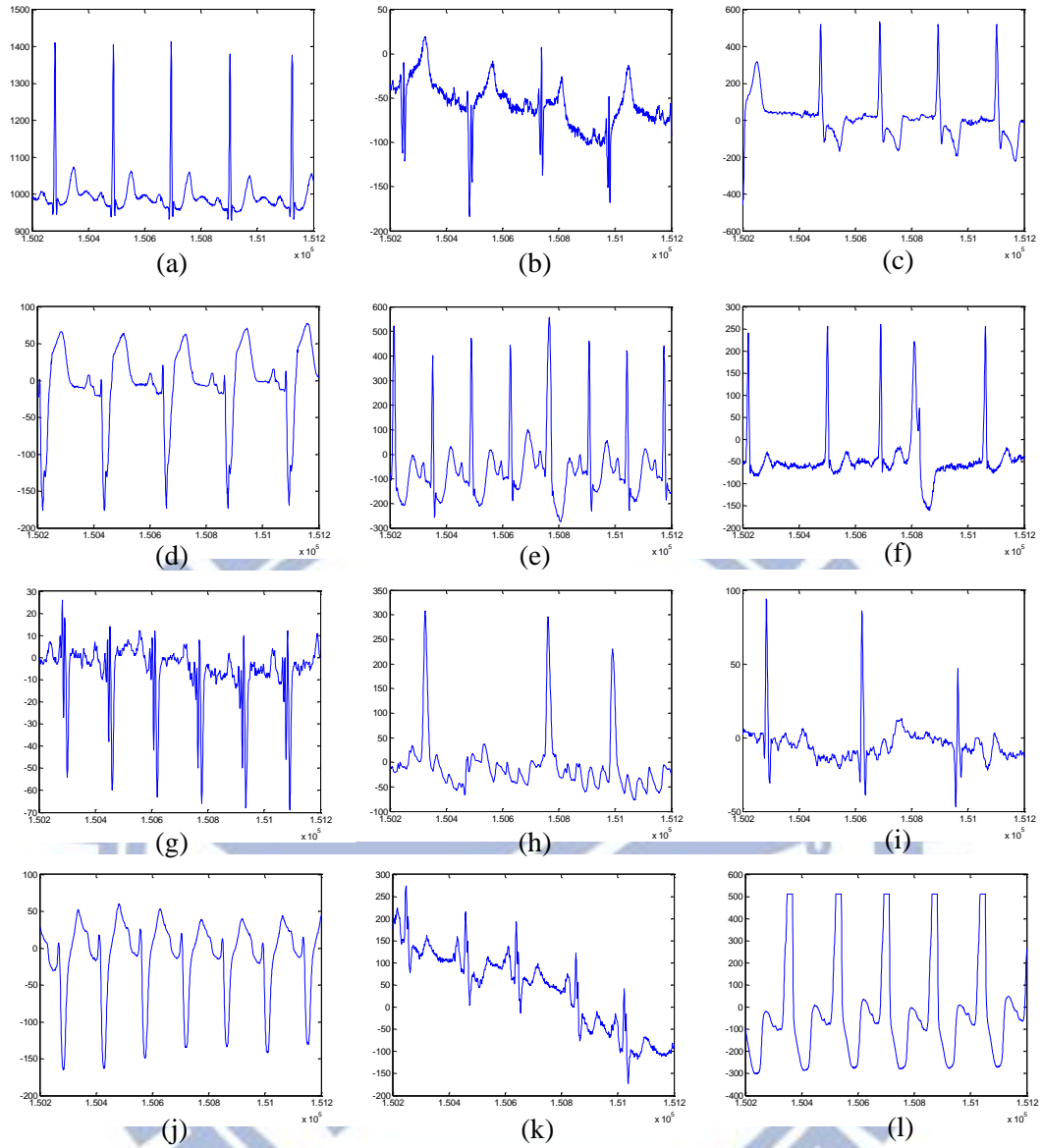


Fig. 2-1 Morphological changes of some example ECG signals extracted from QT database.

Accordingly, most ECG delineation algorithm usually consists of a preprocessing stage and a decision stage. The preprocessing stage usually includes filtering of high frequency noise and baseline drift, or transforming the data into different patterns to make the features more conspicuous. Previous works of ECG QRS detection algorithm utilize methods like wavelet transform [9], band pass filtering [6], genetic algorithm [7], mathematical morphology [8], and phasor transform [10]. Among them, the multi-scale wavelet-based methods are proven to provide effective noise removal and



exists fast transform method which are implementation-friendly for digital implementation. Therefore, wavelet-based method is selected as the basis of the proposed delineation algorithm.

The proposed ECG delineation comprises the multi-scale dyadic wavelet transform and the feature extractor. The DWT decomposes the ECG signal and noise to different wavelet scales. And the feature extractor with search rules and adaptive threshold are applied for the ECG fiducial point decision.

## 2-2 Dyadic Wavelet Transform (DWT)

### 2-2.1 Wavelet Theory

Wavelet transform is widely used in applications such as noise reduction and edge detection and is usually implemented in the form of FIR filter banks with little hardware requirement. Wavelet transform decompose signal by a set of basis function obtained by dilation ( $a$ ) and translation ( $b$ ) of a single prototype wavelet  $\psi(t)$  and is defined as

$$W_ax(b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi\left(\frac{t-b}{a}\right) dt, a > 0. \quad (2-1)$$

where  $W_ax(b)$  is the wavelet coefficient at scale  $a$ , and  $x(t)$  is the original signal. The greater the scale factor ( $a$ ), the wider is the basis function. And the corresponding coefficients give information about lower frequency components of the signal.

If the prototype wavelet is defined as the derivative of a smoothing function  $\theta(t)$ . (2-1) can be rewritten as

$$W_ax(b) = -a \left(\frac{d}{db}\right) \int_{-\infty}^{\infty} x(t) \theta_a(t-b) dt. \quad (2-2)$$

$$\theta_a(t) = \frac{1}{\sqrt{a}} \theta\left(\frac{t}{a}\right). \quad (2-3)$$

Then the wavelet transform at scale  $a$  can be interpreted as the derivative of the filtered version of original signal with impulse response equal to  $\theta_a(t)$ . Therefore, every local maximum/minimum in the time domain will be represented by a zero crossing points surrounded by a positive and a negative peaks, with the amplitude of the peaks corresponded to the maximum/minimum slope. Regarding the application of detecting various ECG features occurring at different time instant coupled with different kinds of noise, the flexibility of scales and the corresponded frequency response give convenience for such application.

For discrete time signal, the dilation ( $a$ ) and translation ( $b$ ) can be chosen to be in dyadic form (2-4) on the time scale plane. Such kind of wavelet transform is then called dyadic wavelet transform, with basis function equal to (2-5).

$$a = 2^k, b = 2^k l. \quad (2-4)$$

$$\psi_{k,l}(t) = 2^{-\frac{k}{2}} \psi(2^{-k}t - l). \quad (2-5)$$

According to [11], the dyadic wavelet transform can be implemented using filter banks with cascaded identical high-pass and low-pass filters as shown in Fig. 2-2 (a). To achieve the same sampling frequency and provide approximate translation invariance, *algorithm á trous* [12] is used. The filter response is interpolated with zero and the down sampler is removed to overcome the translation-invariance (Fig. 2-2 (b)). This is also known as the stationary wavelet transform.

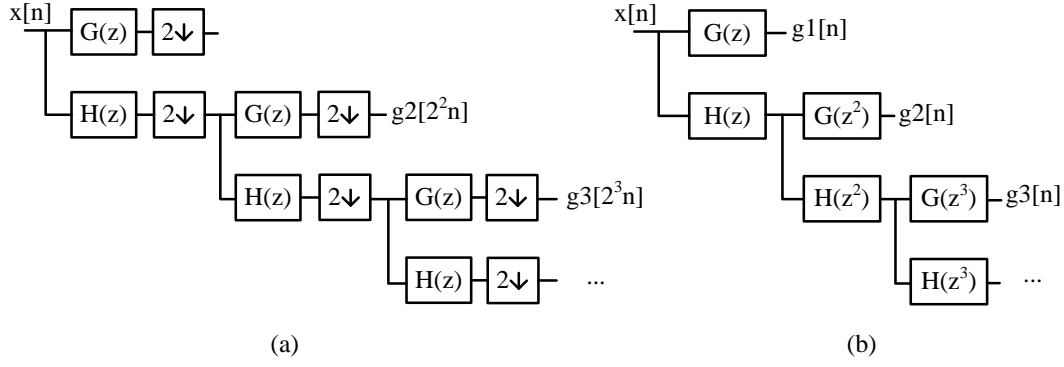


Fig. 2-2 (a) Mallat's Algorithm, (b) algorithm á trous (SWT)

## 2-2.2 Quadratic Spline Wavelet Transform (QSWT)

A quadratic spline originally proposed in [13] is selected as the prototype waveform for the detection algorithm. The Fourier transform of this quadratic spline is depicted as

$$\psi(\Omega) = j\Omega \left( \frac{\sin\left(\frac{\Omega}{4}\right)}{\frac{\Omega}{4}} \right)^4. \quad (2-6)$$

The high-pass  $H(z)$  and the low-pass filter  $G(z)$  implemented in the DWT filter bank as in Fig. 2-2 are

$$H(e^{j\omega}) = e^{\frac{j\omega}{2}} \left( \cos \frac{\omega}{2} \right)^3, \quad G(e^{j\omega}) = 4je^{\frac{j\omega}{2}} \left( \sin \frac{\omega}{2} \right). \quad (2-7)$$

which are FIR filters with impulse response as

$$h_i[n] = \frac{1}{8} \times \{\delta[n + 2^i] + 3\delta[n + 2^{i-1}] + 3\delta[n] + \delta[n - 2^{i-1}]\}. \quad (2-8)$$

$$g_i[n] = 2 \times \{\delta[n + 2^{i-1}] - \delta[n]\}. \quad (2-9)$$

To decide the number of scales to be used, the frequency components of some ECG signals are analyzed together with the filter bank frequency response. Fig. 2-3 shows the frequency response of the ECG signal extracted from MIT-BIH Arrhythmia Database (data 103) together with its QSWT up to 5 scales with 250Hz sampling frequency. From the figure we can see that most energy concentrate in frequency band 0 Hz to 50 Hz. Scale-1 is discarded considering the high frequency noise. Considering hardware cost and filtering performance, the proposed delineation algorithm used scale 2, 3, and 4 for detection of the 5 fiducial points (P, QRSON, R, QRSEND, T).

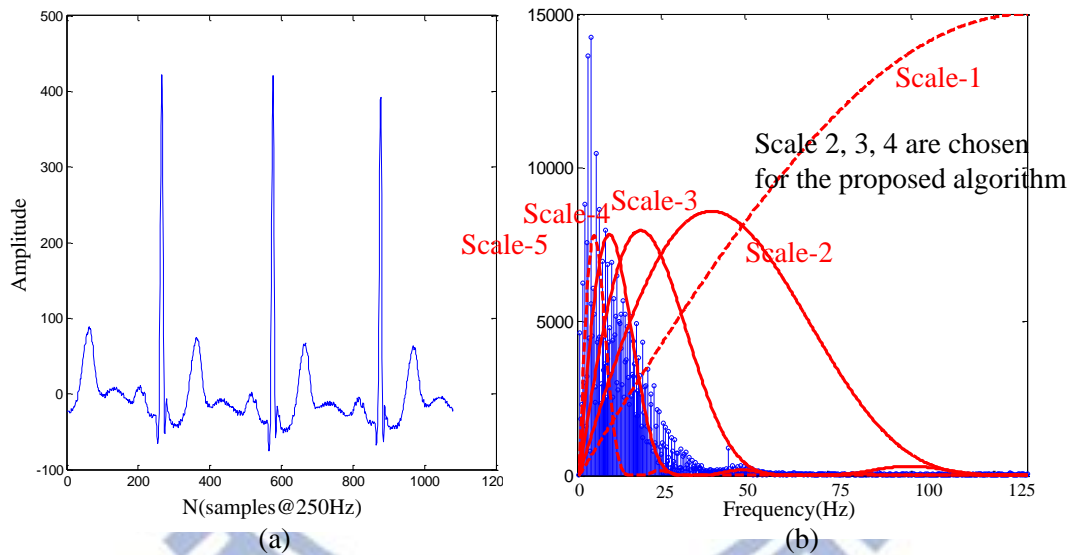


Fig. 2-3 (a) Data #103 from MIT\_BIH Arrhythmia Database at 360Hz sampling frequency and (b) the corresponded frequency response.

## 2-3 Detection Algorithm

The detection algorithm presented in this section targets for the 5 most significant ECG fiducial points (P, QRS<sub>on</sub>, R, QRS<sub>end</sub>, and T) based on the quadratic spline wavelet transform described in the previous section. The dyadic wavelet transform filter out the interference of high frequency noise and baseline drift and decompose the ECG signal into different scales. Detection is then performed based on the cross examination among these scales of coefficients. Comparing with existing off-line detection methods with costly computation, the proposed algorithm is designed suitable for hardware implementation providing comparable detection result. The detection rules for each feature and the adaptive generation for threshold and search window will be described in the following paragraphs.

### 2-3.1 Wave Characteristic and Detection Flow

Fig. 2-4 shows the decomposition of some example ECG waveform using the QSWT. From the figure we can see that a peak in the time domain will be represented by a zero crossing point surrounded by a local maximum and minimum point in the wavelet domain, each representing the deepest rising and falling slope. The reason for discarding scale-1 becomes clear in this figure (high frequency noise). For the 5 desired fiducial points with different wave characteristics, detections are done using different scales of wavelet coefficients. For the most important R peak, we use scale 2, 3, 4 for detection. Because of reduced resolution in higher scales, sharp edges such as the boundary for QRS complex (i.e. QRS<sub>on</sub>, QRS<sub>end</sub>) use coefficients of scale-2 for detection. Wide wave such as P and T wave use higher scales (scale-4) for detection. Considering hardware cost, increased latency in higher scales, and the interference of baseline drift, the scale of wavelet decomposition is limited to four.

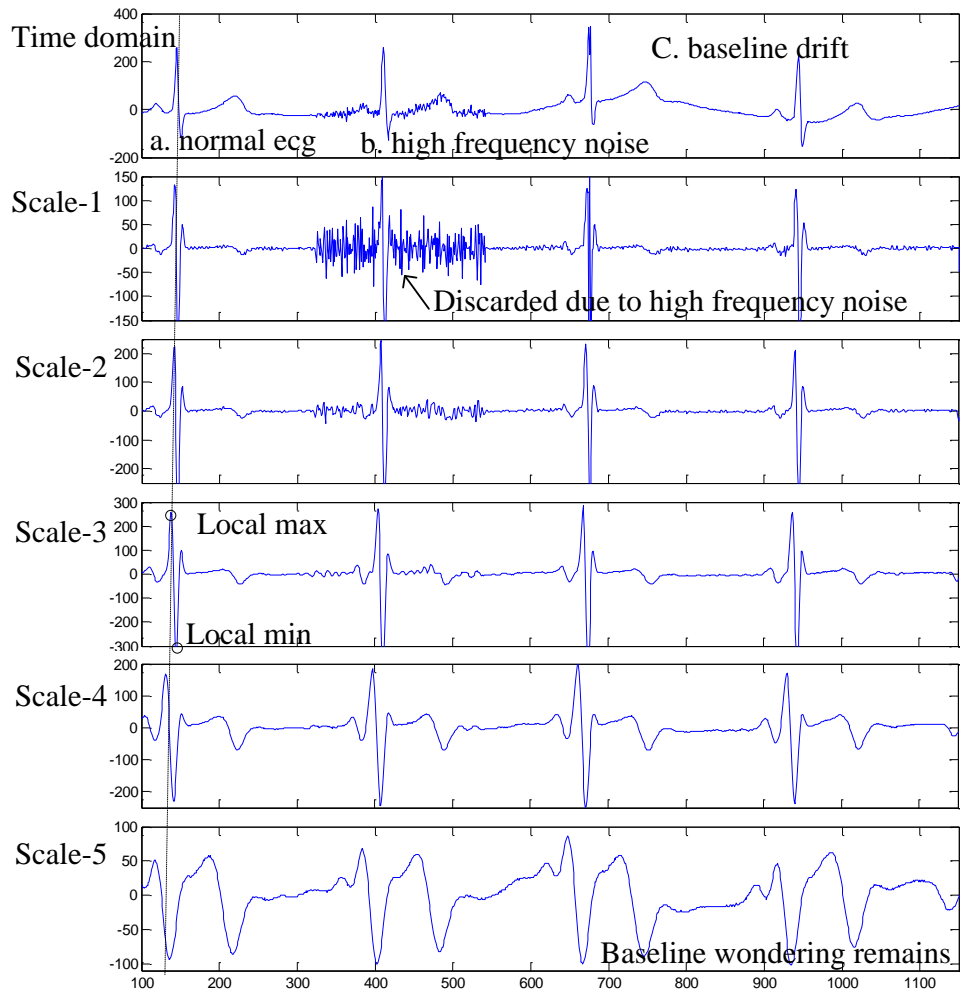


Fig. 2-4 The first 5 wavelet decomposition of ECG signal with noise coupling

Using the zero crossings and local maximum/minimum at each scale, the proposed algorithm detects the five fiducial points within a cardiac cycle by:

- a) Detection for R peak.
- b) Search-back for  $QRS_{on}$  and P wave.
- c) Moves on for  $QRS_{end}$  and T wave.
- d) Update detection threshold and search window.

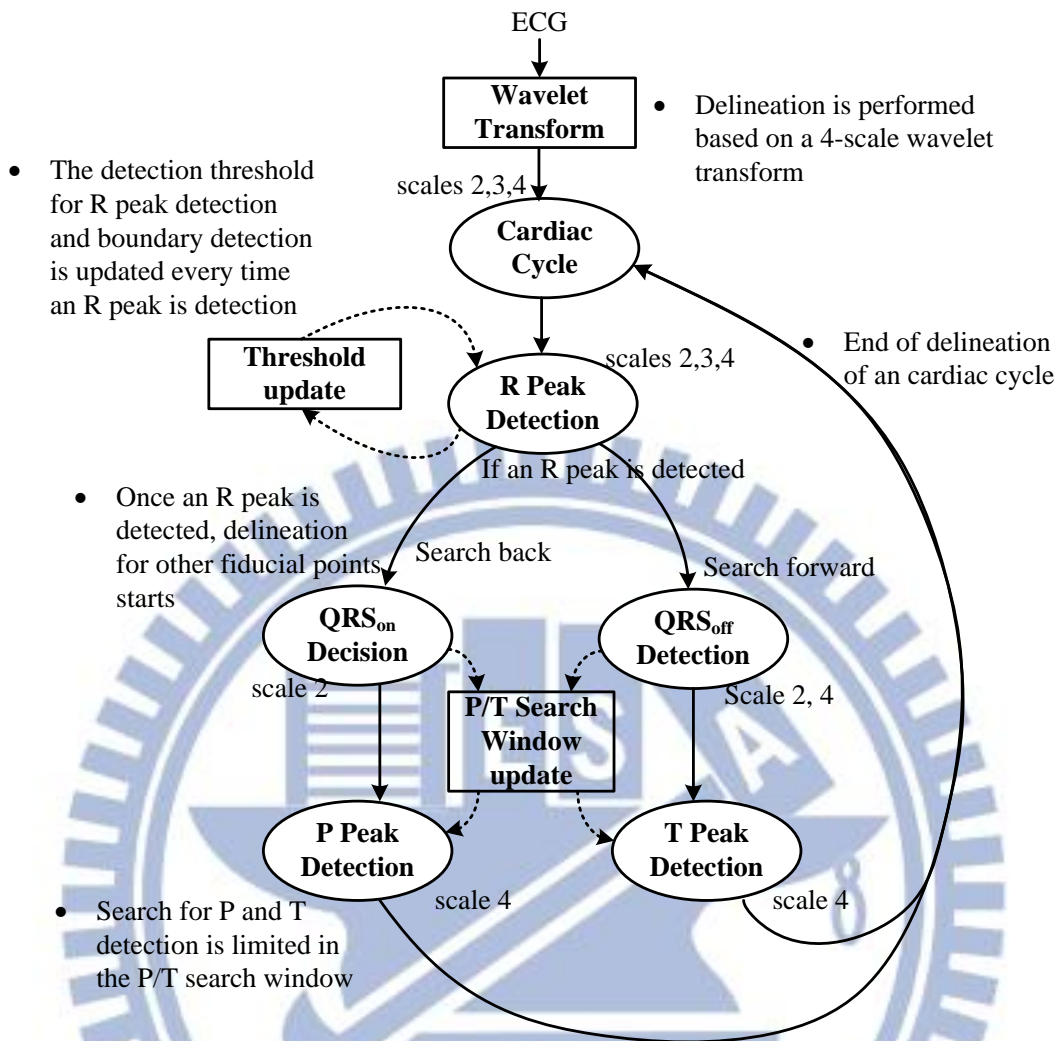


Fig. 2-5 Flow graph for the proposed detection algorithm

Fig. 2-5 shows the flow graph of the detection algorithm. For the best extraction performance, the feature extraction process starts with the most obvious R peak. Based on the detected R peak, the detector searches back for the starting boundary of the QRS complex (i.e.  $QRS_{on}$ ) and P wave. After successful search of these wave points, the detection moves forward for  $QRS_{end}$  and T wave. This completes the detection of the 5 wave within a cardiac cycle. To reduce unnecessary search time and power, the detection for P and T wave is limited in a search window. The search window and the detection thresholds for locating the peaks are updated every cardiac cycle.

Considering hardware cost, the design rules for  $QRS_{on/end}$  and P/T waves are designed to be similar so the hardware can be shared. The detection detail will be explained as follows.

### 2-3.2 R Peak Detection

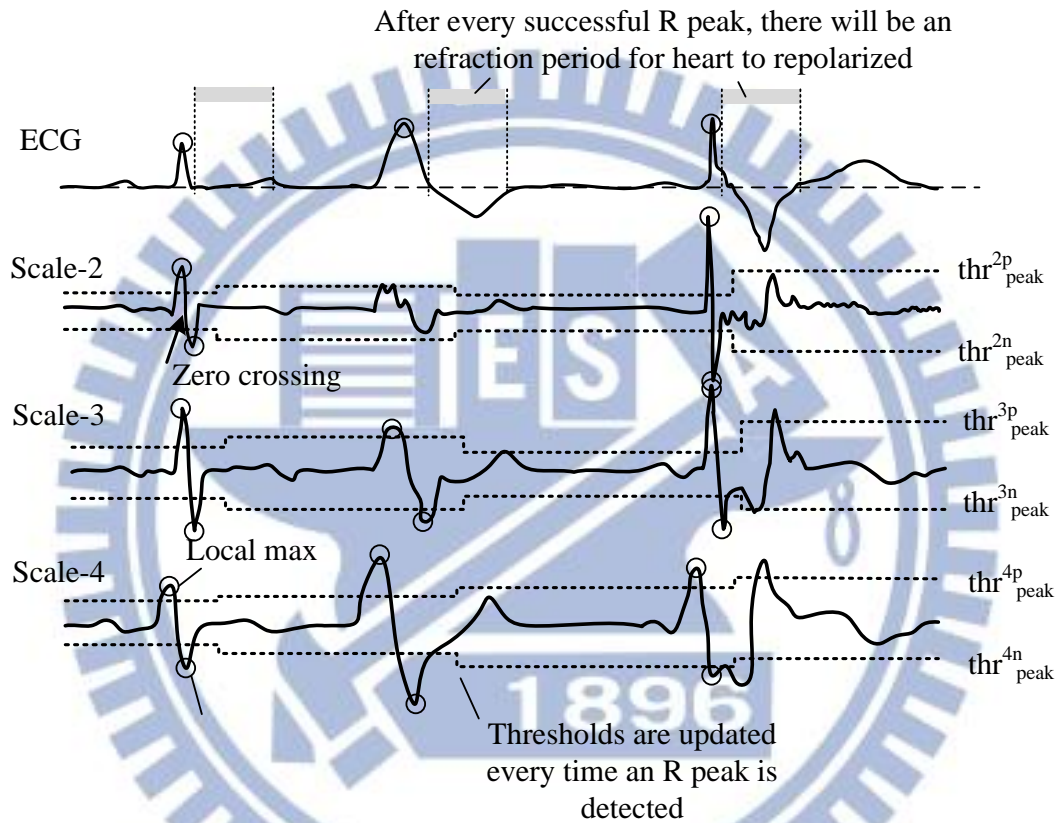


Fig. 2-6 R peak detection is performed by searching for the min-max pair exceeding the peak threshold in scale 2, 3, and 4

A peak is indicated by the temporal relationship of local minimum and maximum peak pair defined as the point exceeding the peak threshold ( $thr^{2p}_{peak}$ ,  $thr^{2n}_{peak}$ , etc.) together with the zero crossing between them. The detection of R peak is performed with cross examination in the 3 scales (scale-2, 3, and 4) because of its high importance. To prevent large data storage, the detection is done sequentially using 3 parallel state machines. According to the 3 scales of wavelet coefficients, the state



machines change state when finding a positive or negative peak, a zero crossing point, and the proceeding peaks opposite to the previous detected peak, and output the marking for possible candidate for QRS complex. Using the rule of majority, if candidate markings are found in 2 or more scales, the zero crossing point in scale-2 will be considered as the detected R peak. Besides locating the R peak location, the information of this min-max pair (amplitude, location) is recorded for successive  $QRS_{end}$  detection and threshold update.

After successful locating an R peak, the conducting cell needs to repolarize in order contract again. This period is called the refraction period. During this period, no peak will be considered as R peak.

### 2-3.3 $QRS_{on/end}$ Detection

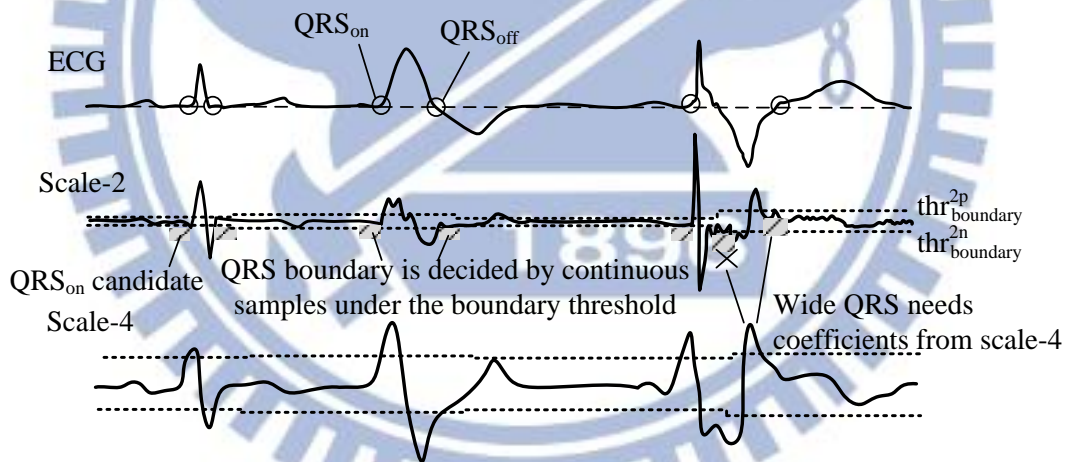


Fig. 2-7  $QRS_{on/end}$  detection search for continuous samples under the edge threshold

Considering reduced resolution in higher scales, coefficients of scale-2 is used of detection of QRS edges ( $QRS_{on/end}$ ). Because of similar wave characteristic, the detection rules for  $QRS_{on/end}$  detection are designed to be the same so the hardware can be shared (with only comparators). The detection of  $QRS_{on/end}$  is performed by searching for continuous points under the boundary threshold as illustrated in Fig. 2-7.

The detection of  $QRS_{end}$  can be performed following successful detection of R peak. To avoid search back and additional storage for  $QRS_{on}$  detection, possible points satisfying the detection rules are saved as  $QRS_{on}$  candidates. Finally the candidate that supplies the most nearest sample to the R peak will be confirmed as the real  $QRS_{on}$  point.

To avoid morphological changes of wide QRS complex (syndrome: PVC-premature ventricular contraction) which cannot be characterized by scale-2, coefficients of scale-4 is used to distinguish the wide morphology and avoid detecting false boundary (shown in the right part of Fig. 2-7).

### 2-3.4 P/T Wave Detection

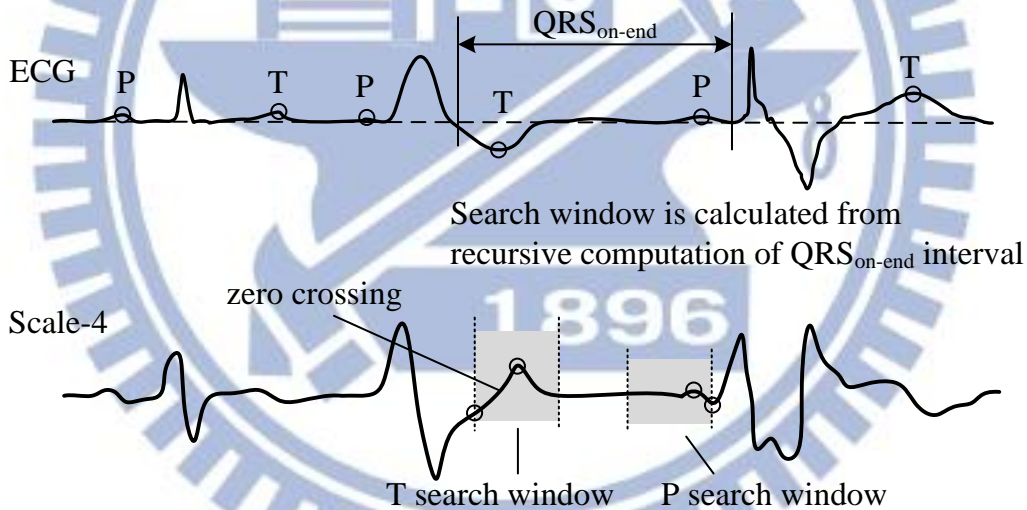


Fig. 2-8 Search window defined for P/T detection

Because of the wider wave characteristic, the detection process of P and T wave detection use scale-4 for detection. The process is as follows: First, the search is limited in a search window defined relatively according to the recursive computing of  $QRS_{on}$  to  $QRS_{end}$  interval. This reduces extra time and power for unnecessary search according to the physical phenomenon for a normal cardiac cycle. Instead of using RR interval as reference,  $QRS_{on}$  to  $QRS_{end}$  interval eliminates the influence of

morphological changes of QRS complex. The size of the search window is carefully designed because it results in extra storage for P wave search back after R peak detection. Finally we define the search window boundary for P and T wave detection to be:

$$SW_{pr} = 10. \quad (2-10)$$

$$SW_{pl} = \begin{cases} 100 & \text{if}(SW_{pl} > 100) \\ 10 + QRS_{on-end} \times 0.375. & \text{o.w.} \end{cases}$$

$$SW_{tl} = 15.$$

$$SW_{tr} = \begin{cases} 100 & \text{if}(SW_{tr} > 100) \\ 15 + QRS_{on-end} \times 0.4. & \text{o.w.} \end{cases} \quad (2-11)$$

with a maximum search range of 100 samples under 250Hz sampling frequency. The  $QRS_{on-end}$  is the value that updated every time a new QRS complex is detected. These values are designed according to the physical nature of our heart. For example, the value of  $SW_{pr}$  is chosen to be 10 because the delay caused by AV node between atrial and ventricular is approximately 0.1 second.

Within this window, we search for the global maximum/minimum points. If one of them exceeds the P/T threshold, a wave is considered to exist and is indicated by the zero crossing point between them.

### 2-3.5 Adaptive Threshold and Window Update

As mentioned previously, the robustness of the proposed algorithm lies from the adaptively update of detection parameters including the peak threshold for R peak detection and the boundary threshold for  $QRS_{on/end}$  detection and the search window for P and T wave detection.

For R peak detection, separate thresholds ( $thr_{peak}^{xp}, thr_{peak}^{xn}$ ) are used for positive and negative peaks to avoid failed detection for asymmetric rise and fall peaks shown in Fig. 2-6. Avoiding costly computation such as division, square roots [14], or root-mean-square [9], thresholds are computed based on the information of the recorded value of local min-max pair (signal peak) and the recorded noise level (noise peak). The equation of the peak threshold for R peak detection and boundary threshold for QRS<sub>on/end</sub> detection are depicted as follows:

$$\begin{cases} \text{if } (local\_max_x \geq thr_{peak}^{xp}) \rightarrow SP_{peak}^{xp} \\ \text{else if } (local\_max_x < thr_{peak}^{xp}) \rightarrow NP_{peak}^{xp} \end{cases} \quad \text{(positive threshold)} \\ \begin{cases} \text{if } (local\_min_x \geq thr_{peak}^{xn}) \rightarrow SP_{peak}^{xn} \\ \text{else if } (local\_min_x < thr_{peak}^{xn}) \rightarrow NP_{peak}^{xn} \end{cases} \quad \text{(negative threshold)} \end{cases} \quad (2-12)$$

where x is the scale.

$$thr_{peak}^{xp'} = thr_{peak}^{xp} \times 3 + NP_{peak}^{xp} + (SP_{peak}^{xp} - NP_{peak}^{xp}) \gg 1 \quad (2-13)$$

$$thr_{peak}^{xn'} = thr_{peak}^{xn} \times 3 + NP_{peak}^{xn} + (SP_{peak}^{xn} - NP_{peak}^{xn}) \gg 1$$

$$thr_{bdry}^p = thr_{peak}^{2p} \gg 4 \quad (2-14)$$

$$thr_{bdry}^n = thr_{peak}^{2n} \gg 4$$

Equation (2-12) describes the rule to classify the noise peak and signal peak. The new threshold is computed by the weighted average of the current threshold and the new threshold based on the detected noise peak and signal peak (2-13). Equation (2-14) shows that the threshold for QRS complex boundary detection is based on a simple shift of the peak threshold.

## 2-4 Simulation Result and Performance Evaluation

Fig. 2-9 shows the detection result of the proposed algorithm with different wave morphologies and noise coupling.

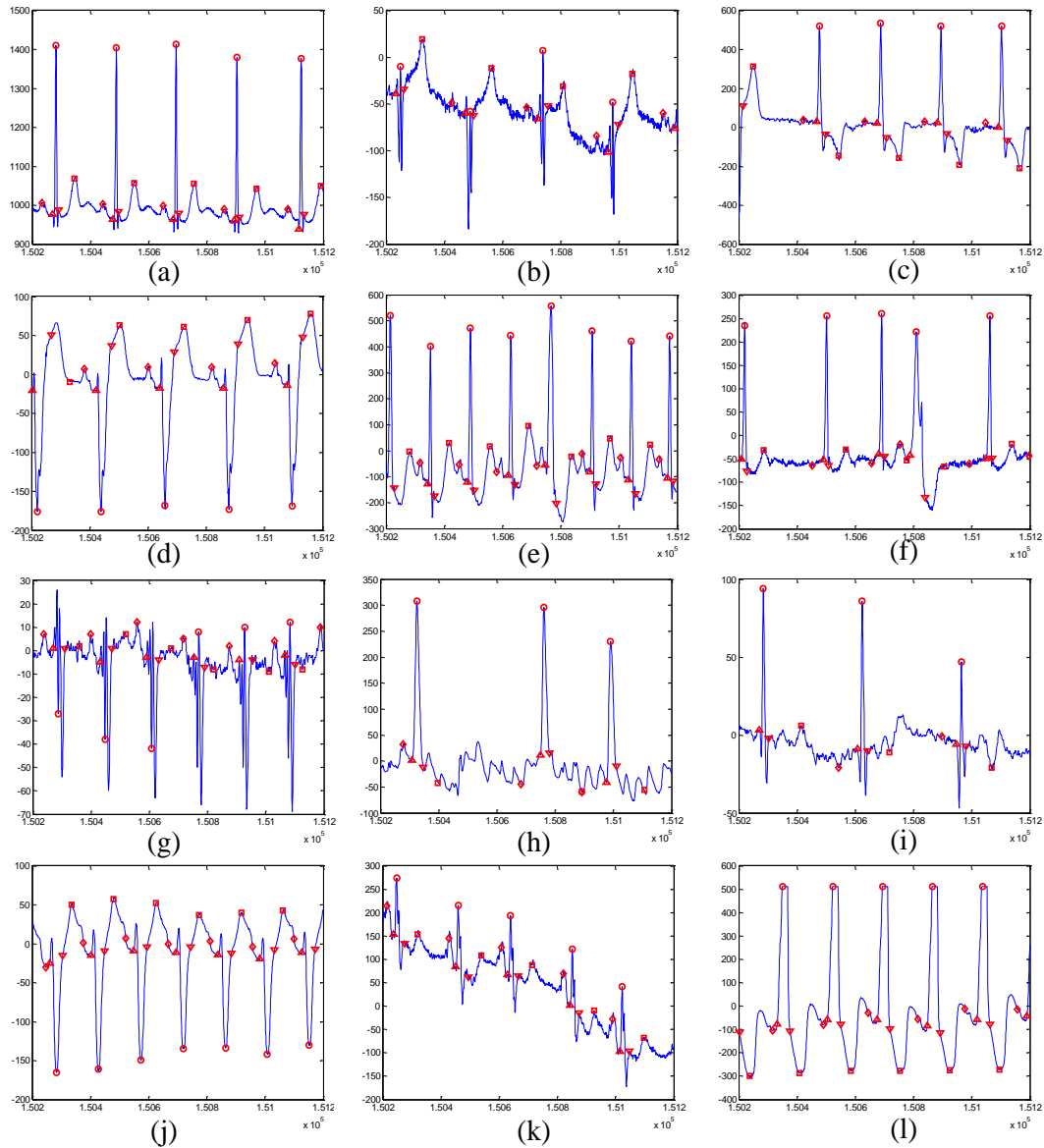


Fig. 2-9 Detection result of the proposed with ECGs with morphological changes and noise

As there is no golden rule for the decision for peaks, onset, and endset, the validation for the detection result needs to be performed by doctors. Thanks to Physionet [3], lots of standard databases are provided and sorted with detail information about the ECG including the corresponded syndrome and either manually

or automatically annotated fiducial points. In this report, we choose the two common databases for the validation of our proposed algorithm, namely the MIT-BIH Arrhythmia Database (MITDB) [4] and QT Database (QTDB) [5]. Here we first make a brief introduction about the databases and provide the validation result.

- MIT-BIH Arrhythmia Database (MITDB) [4]

The MITDB includes 48 specially selected Holter recordings with anomalous but clinically important phenomena at 360Hz sampling frequency, 11-bits resolution and 10-mV amplitude range with automatically determined R peak annotations. We use this database for the validation for R peak detection.

- QT Database (QTDB) [5]

The original goal for QTDB is to make a database with sufficient ECGs coverage for variety of QRS and ST-T morphologies in order to challenge existing algorithms with real-world variability. The 105 records were chosen primarily from among existing ECG databases, including the MITDB, the European Society of Cardiology ST-T Database [4], and several other ECG databases collected at Boston's Beth Israel Deaconess Medical Center. All records all resample to 250Hz in QTDB. Different annotations are provided including the automatically annotation for QRS complex (.man) and manually determined waveform boundaries by two experts (.q1c .q2c). We validate the detection result of all the fiducial points using this database.

The two parameters to qualify the detection result are sensitivity ( $Se$ ) and positive predictivity ( $Pr$ ) and are depicted as

$$Se = \frac{TP}{TP + FN}, \quad Pr = \frac{TP}{TP + FP}. \quad (2-15)$$

where TP stands for true positive detection, FN stands for false negative detection, and FP stands for false positive detection. The sensitivity  $Se$  reports the percentage of true

beats that were correctly detected. The positive predictivity *Pr* reports the percentage of beat detections which were in real true beats (accuracy).

Comparison of the detection result with the state-of-the art detection algorithm (including software algorithm and hardware detector) are also listed in Table 2-1 and Table 2-2. Table 2-3 lists the detail detection result of R peak detection verified using MITDB.

Table 2-1 R peak detection comparison with state-of-the-art detector using MITDB

Detector		# Annotation	TP	FP	FN	Se (%)	Pr (%)
<b>This work</b>							
<b>Wavelet Transform</b>	hardware	109980	109632	317	348	99.71	99.68
[9] 2004 TBE	software	109428	109208	153	220	99.80	99.86
Wavelet Transform							
[10] 2010 PMea	software	109428	109111	35	317	99.71	99.97
Phasor Transform							
[17] 2012 TBCAS	hardware	109134	108381	330	753	99.31	99.70
Wavelet Transform							
[16] 2010 ISCAS	hardware	N/A	N/A	N/A	N/A	95.65	99.36
Filtering, Diff, Square							
[15] 2009 TBCAS	hardware	N/A	109510	213	214	99.81	99.80
Mathematical							
Morphology							
[14] 2009 ASSCC	hardware	109492	108892	117	500	99.63	99.89
Wavelet Transform							

Table 2-2 Fiducial points delineation result comparison using QTDB

Detector	P		QRS <sub>on</sub>		QRS <sub>end</sub>		T	
	Se (%)	Pr (%)	Se (%)	Pr (%)	Se (%)	Pr (%)	Se (%)	Pr (%)
<b>This work</b>								
<b>Wavelet Transform</b>	99.59	96.11	99.97	100.00	99.97	100.00	99.38	99.41
[9] 2004 TBE								
Wavelet Transform	98.87	91.03	99.97	N/A	99.97	N/A	99.77	97.79
[10] 2010 PMea								
Phasor Transform	99.27	98.75	99.91	99.94	99.91	99.94	99.77	99.66

Table 2-3 R peak detection result within MITDB

Records	Total (beats)	TP	FP	FN	Se (%)	Pr (%)
100	2272	2272	1	0	100.00	99.96
101	1864	1863	2	1	99.95	99.89
102	2186	2186	1	0	100.00	99.95
103	2084	2082	0	2	99.90	100.00
104	2228	2225	32	3	99.87	98.58
105	2585	2562	64	23	99.11	97.56
106	2018	2009	5	9	99.55	99.75
107	2135	2131	1	4	99.81	99.95
108	1809	1796	8	13	99.28	99.56
109	2530	2524	3	6	99.76	99.88
111	2123	2123	17	0	100.00	99.21
112	2537	2537	5	0	100.00	99.80
113	1794	1794	0	0	100.00	100.00
114	1878	1878	10	0	100.00	99.47
115	1953	1953	0	0	100.00	100.00
116	2396	2388	1	8	99.67	99.96
117	1534	1534	4	0	100.00	99.74
118	2277	2277	2	0	100.00	99.91
119	1986	1986	1	0	100.00	99.95
121	1862	1860	1	2	99.89	99.95
122	2475	2474	2	1	99.96	99.92
123	1518	1518	1	0	100.00	99.93
124	1619	1618	1	1	99.94	99.94
200	2598	2595	7	3	99.88	99.73
201	1962	1949	3	13	99.34	99.85
202	2136	2130	0	6	99.72	100.00
203	2986	2936	25	50	98.33	99.16
205	2655	2651	1	4	99.85	99.96
207	2323	2238	25	85	96.34	98.90
208	2954	2939	3	15	99.49	99.90
209	3005	3004	5	1	99.97	99.83
210	2651	2626	4	25	99.06	99.85
212	2747	2747	2	0	100.00	99.93
213	3251	3242	1	9	99.72	99.97
214	2261	2255	3	6	99.73	99.87
215	3364	3363	1	1	99.97	99.97
217	2207	2203	2	4	99.82	99.91
219	2157	2153	1	4	99.81	99.96
220	2047	2047	0	0	100.00	100.00
221	2426	2419	2	7	99.71	99.92
222	2484	2479	5	5	99.80	99.80
223	2604	2596	1	8	99.69	99.96
228	2057	2039	35	18	99.12	98.31
230	2257	2256	1	1	99.96	99.96
231	1571	1571	0	0	100.00	100.00
232	1783	1780	25	3	99.83	98.61
233	3078	3072	1	6	99.81	99.97
234	2753	2752	2	1	99.96	99.93
<b>Total</b>	<b>109980</b>	<b>109632</b>	<b>317</b>	<b>348</b>	<b>99.71</b>	<b>99.68</b>



From Table 2-1, the proposed algorithm achieves 99.71% sensitivity and 99.68% positive predictivity for R peak detection. Although with reduced computation complexity, the performance of the proposed algorithm is still compatible with the published off-line detection algorithms. Our algorithm achieves similar detection accuracy comparing with the on-line detection ASICs.

Table 2-2 shows the detection result of other targeted fiducial points (P, QRS<sub>on</sub>, QRS<sub>end</sub>, T) comparing with the 2 off-line detector verified using QTDB q1c annotation. The proposed algorithm achieves better detection result at QRS<sub>on</sub> and QRS<sub>end</sub> detection and similar result for P and T wave detection.

## 2-5 Summary

In this chapter we proposed an ECG delineation algorithm especially for abnormal alarm based on 4-scale quadratic spline wavelet transform. The delineation algorithm can extract P, QRS<sub>on</sub>, R, QRS<sub>end</sub>, and T wave with accuracy over 99%. The wavelet transform removes noise interference and decompose ECG signal into different frequency bands. With cross examination among the decomposed scales and adaptive threshold update considering noise level, the algorithm is suitable for mobile ECG monitoring. Designed using only simple operations, the algorithm can be implemented using low power ASICs. Chapter 4 describes the architecture for the implemented ASIC delineator, using mixed synchronous and asynchronous design style with low power techniques.

# Chapter 3:

## Asynchronous Design

---

In this chapter, we first provide the motivation for going from synchronous design to asynchronous design, the advantages and disadvantages. Then we introduce the basic theory for asynchronous design. A 2-phase handshake protocol modified from the MOUSETRAP [21] pipeline for iterative computation is proposed and tested. Design flow using commercial CAD tool is also provided, making asynchronous design an option for synchronous designers with standard cells. In the end, an example design of an energy-efficient 16-tap FIR filter is built to make comparison between asynchronous and synchronous design.

### 3-1 Motivation

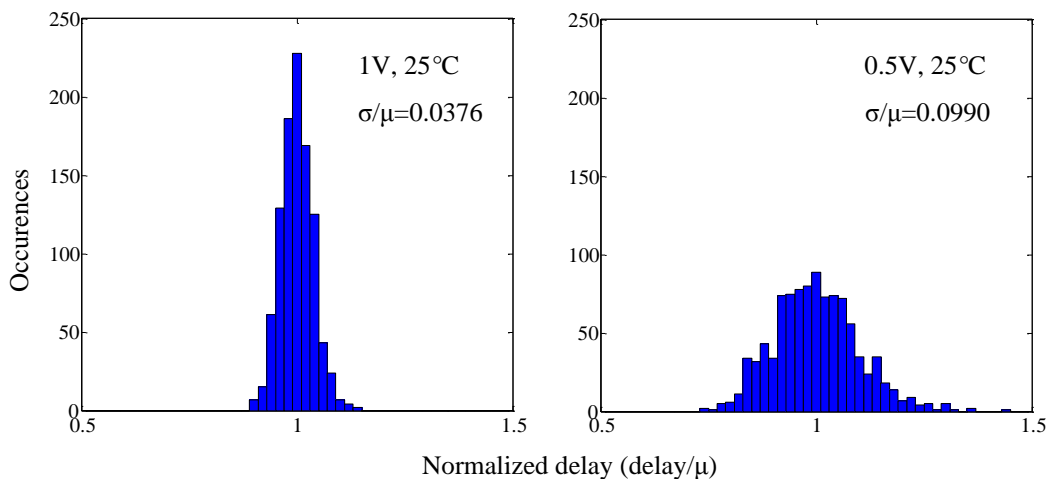


Fig. 3-1 The Monte Carlo simulation with intra-die variation of 3 sigma using UMC 90nm process under 1.0V and 0.5V supply voltage respectively

Voltage scaling is a common technique to reduce the power consumption in circuit design. However, as the supply voltage is scaling down, circuit propagation delay becomes extremely sensitive to PVT variation. Hence, large delay margin is required

for successful operation for synchronous design. However, for asynchronous circuit, the operation speed is decided by the handshakes between registers instead of the global worst case clock cycle. This average case design can therefore result in faster operation speed or lower power consumption.

Fig. 3-1 shows the Monte Carlo simulation for the delay of the critical path in a multiplied accumulator (MAC) under 1.0V and 0.5V supply voltage. The figure clearly shows that the delay variance at 0.5V is approximately three times larger than the variance at 1.0V because synchronous design uses the worst case as its operation condition. Large safety timing margin is wasted for synchronous design.

Besides combating PVT variation under low supply voltage, asynchronous circuit is also an attractive technique for low-speed systems with event-driven asynchronous functions that are activated only when certain event occurred. Asynchronous design provides power management with low latency and removes the requirement for additional high speed clock. Simple handshake interface also makes it easy for system integrations between asynchronous and synchronous portion.

## 3-2 Introduction to Asynchronous Circuit

### 3-2.1 Moving from Synchronous to Asynchronous Approach

Synchronous designs use a global clock to synchronize the whole design. Besides dealing with severe PVT variations under low supply voltage, uncertainty of clock source is another important issue. Clock skew and jitter together with the logic propagation variation under low supply voltage makes synchronous approach an inefficient design for low supply systems. Although current CAD tools provide powerful algorithm to generate clock trees, the well-spread buffers and delay cells still results in large power overhead.

Unlike synchronous design, asynchronous design replaces the synchronous global clock into locally handshake circuit. Fig. 3-2 shows a common pipeline structure for both synchronous design and asynchronous design. The red part shows the replacement from global clock to locally handshake circuit. There are numerous approaches to designing without clocks, each with various pros and cons depending on the design style. Some of the major potential benefits include the follows.

- Robust operation against PVT variation due to elimination of fixed clock.
- Modular composition and delay insensitive interfacing.
- Power management with very low latency.

However, some significant potential drawbacks still exists for clock-less designs includes:

- Complicated design approaches unfamiliar for synchronous designer
- Lack of support from existing EDA tools
- Area and performance overhead for handshake circuits

Therefore, one must be clear about the properties of asynchronous designs when applying it to the system, giving its advantages and reduce the overhead as much as possible to deliver an efficient design. The next part will be some introduction to the basic asynchronous pipeline style and handshake protocol.

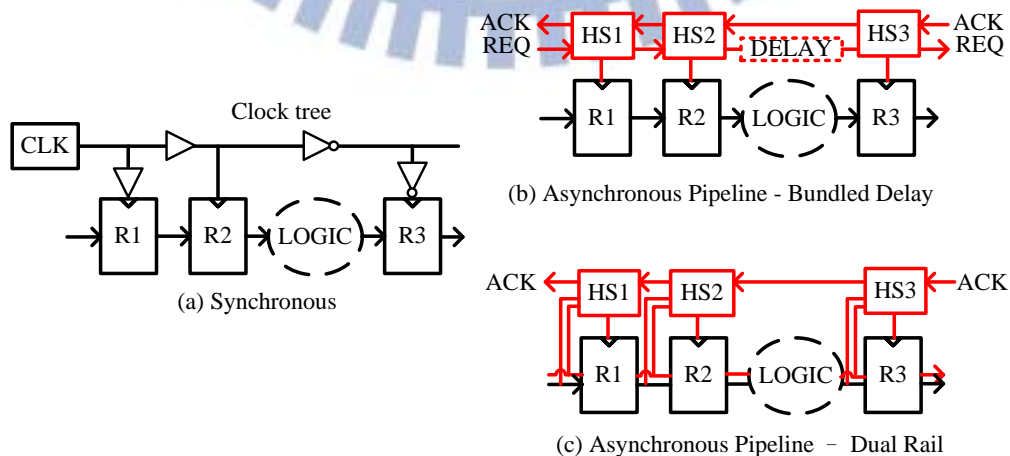


Fig. 3-2 The difference between synchronous design and asynchronous design (red part).

### 3-2.2 Handshake Protocol

The two most common asynchronous design styles are the bundled-delay asynchronous pipeline (Fig. 3-2 (b)) and dual rail asynchronous pipeline (Fig. 3-2 (c)).

A brief introduction is made here above the two asynchronous

- Dual rail asynchronous pipeline

One common type of the quasi delay insensitive circuits is the dual rail asynchronous design. Dual rail design encodes the arrival information into the data bit itself. Therefore, two bits are required to represent the arrival of one-bit information which makes the design “dual rail”. Table 3-1 shows a dual rail encoding scheme. With this kind of encoding, two parties can communicate reliably regardless of the delay variations. However, additional dual rail datapath and storage results in large area and power penalties. Although customized gates can be designed to reduce the overhead, the design process is too much effort and time costly for synchronous designer.

Table 3-1 One encoding scheme for dual rail encoding

State	True rail	False rail
Null	0	0
Logic 0	0	1
Logic 1	1	0
N/A	1	1

- Bundled-delay asynchronous pipeline

The bundled-data asynchronous designs are conceptually closest to synchronous design. Each datapath through a combination block is matched with a delay line with the same propagation delay. Because of this “bundled” delay line, this type of asynchronous circuit is called “bundled delay” or “matched delay”. Thanks to the similarity with synchronous design, we can use standard cells and commercial CAD tool with customized design constraints to implement the design.

Handshake circuit can also be categorized using the number of phase that is required for one data transmission. The most common protocols are the 4-phase and 2-phase handshake protocol. The 4-phase handshake protocol, also named “Return-to-Zero (RTZ)” or level triggered protocol, uses logic “1” to signal data valid, therefore simple transparent latches can be used as pipeline registers. However, additional return-to-zero operation is required for the start of the next handshake, which results in additional switching and time. To reduce the extra RTZ time, AND delay line configured as Fig. 3-3 (c) [19] can be used to force the output to zero once the input goes to zero.

The 2-phase handshake protocols, also named non return-to-zero (NRTZ) or transition signaling protocol, use signal transitions as signal event, no addition RTZ is required, but more complex handshake circuit or latched may be needed for recognition of both positive and negative edges.

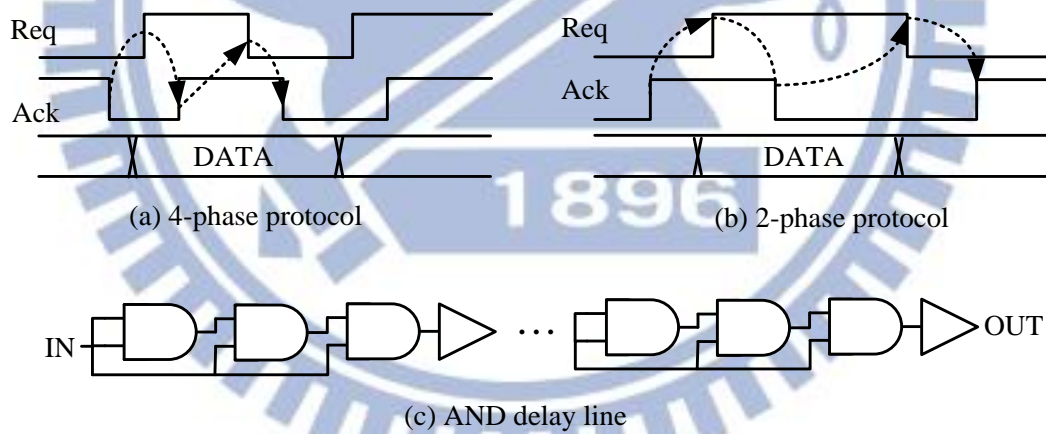


Fig. 3-3 (a) 4-phase handshake protocol (b) 2-phase handshake protocol

### 3-2.3 Muller C Element

One of the basic components that are generally used in asynchronous design is the Muller C-element cell [20]. The output reflects the input when all the input matches. Fig. 3-4(a) shows the symbol and truth table of a 2-input C element. Such kind of function can be implemented using either by standard NAND gates (Fig. 3-4(a)) or by customized transistors level construction (Fig. 3-4 (b)).

C element is the basic gate for the famous Muller pipeline [20]. Because of its unique function, it can also be used in fork and join structures (Fig. 3-5). Fork structure is used when one request is sent to more than two recipients. The corresponded acknowledge signal will consist of a join structure using C element (Fig. 3-5(a)) and vice versa for conditions when two requests are being combined to one recipient (Fig. 3-5(b)).

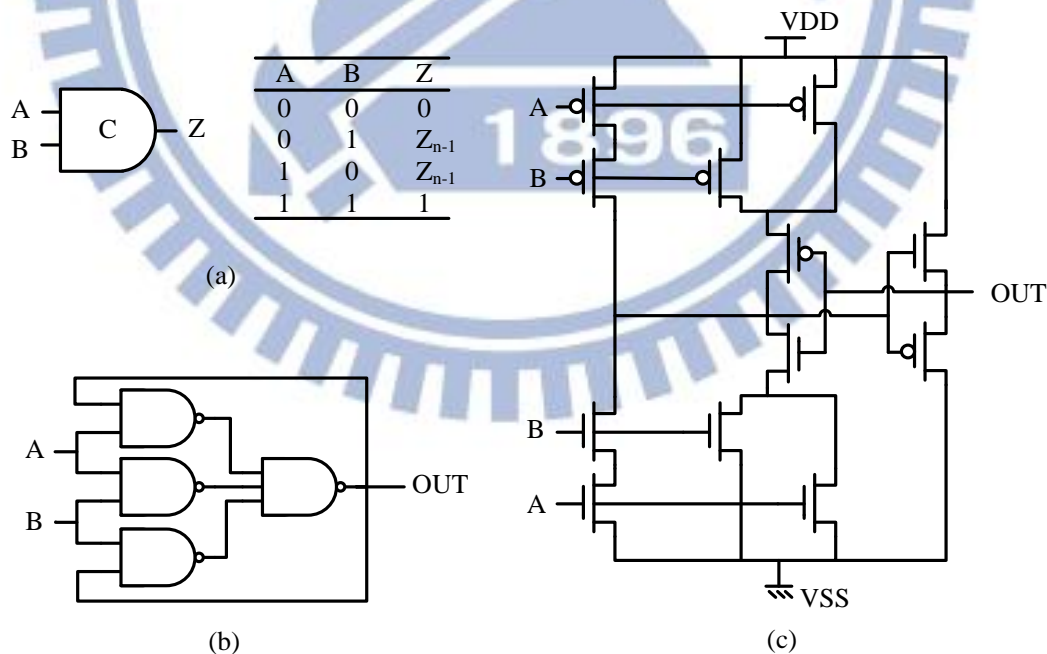


Fig. 3-4 (a) Symbol and truth table of a 2 input C element (b) Standard cell based C element (c) Transistor level implementation

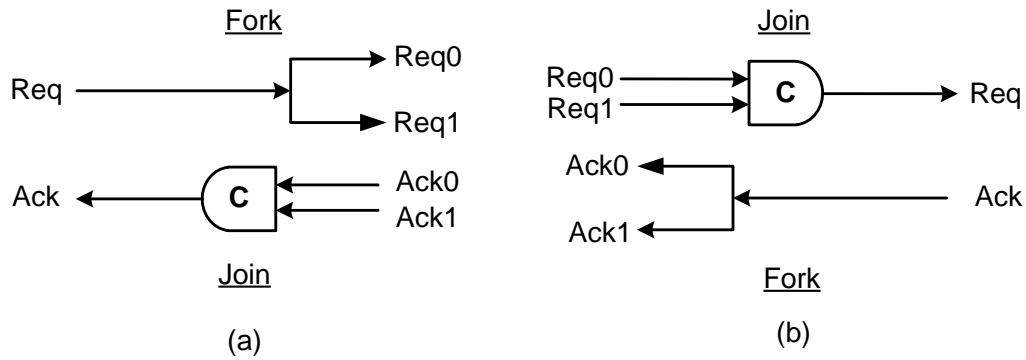


Fig. 3-5 Fork and join structures

### 3-2.4 Asynchronous Pipelines

Here a brief introduction about 2 existing asynchronous pipelines is presented: The 4-phase Muller pipeline [20] and 2-phase minimal-overhead ultra-high-speed transition-signaling asynchronous pipeline (MOUSETRAP) [21], including the structure and working mechanism. Both of the pipelines are bundled-delay pipelines.

- 4-phase Muller pipeline [20]

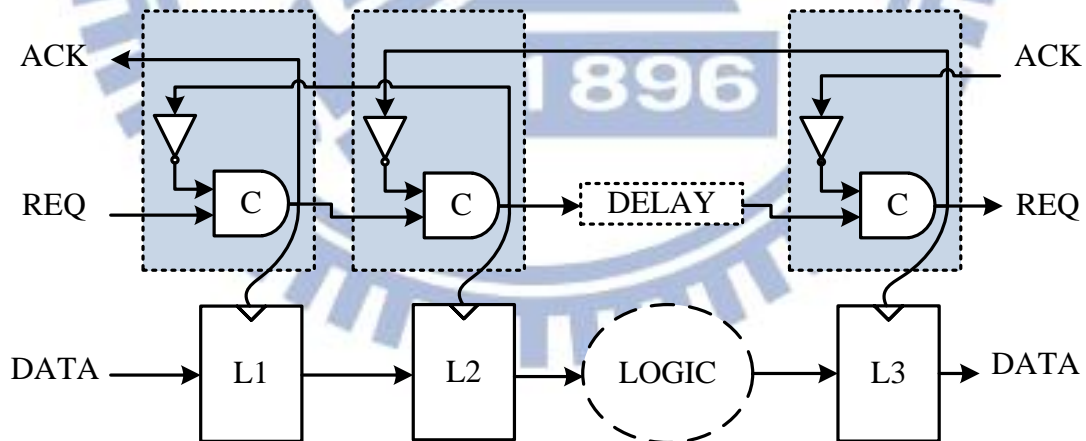


Fig. 3-6 Muller pipeline

Muller pipeline is the backbone for many other variations and extensions of asynchronous pipeline. Simple handshake circuit is built using only inverters and C element. To understand the handshake working mechanism, we first assume all the



outputs of C element are reset to zero. A firing of 0 to 1 from the left REQ will make the output of the first C element to 1 and triggered the first latch R1. This request signal will propagate through the pipeline stages together with the data. When the successor receives the request, an acknowledge signal will be sent back to the predecessor, and the predecessor will be able to receive the new data. With no customized cell and simple timing constraint, this pipeline style is suitable to build using commercial CAD tools.

- 2-phase MOUSETRAP pipeline [21]

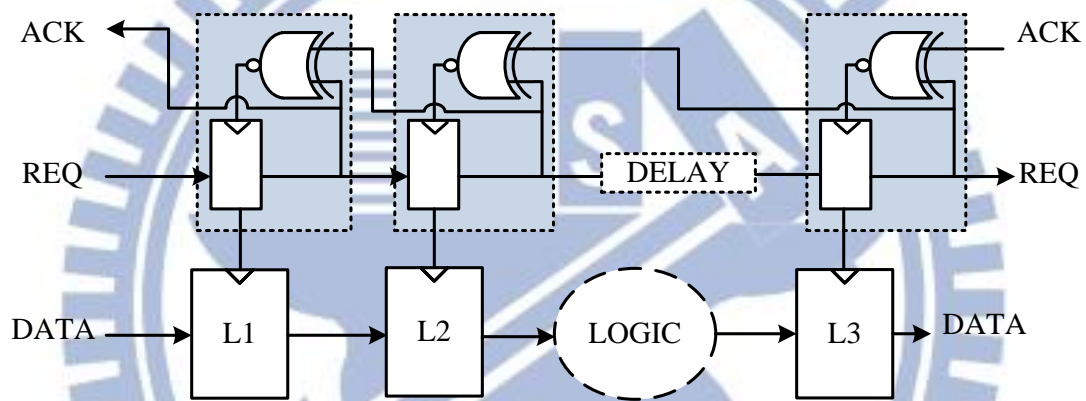


Fig. 3-7 MOUSETRAP pipeline

The MOSETRAP pipeline is a 2-phase pipeline style that can be constructed using simple transparent latch while other 2-phase pipeline style may require special designed latched to capture the transition of 2-phase protocols. The handshake circuit consists of a latch and a XNOR gate. With all handshake latches reset to zero, each latch is transparent before the data arrives. Upon receiving the arrival of a transition on the request signal, the latch captures the data and becomes opaque. The REQ transition and the data propagate continuously to the next stage, and are again captured. An acknowledge signal is sent back to the predecessor, enabling the predecessor and makes it transparent again in order to capture new data.

### 3-3 Design Flow Using Commercial CAD Tool

While asynchronous designs can prove substantial benefits, it is still largely limited by the incompatibility with existing CAD tool. Here we attempt a design flow based on relative timing and post-layout SPICE verification, to create and prove correct constraints for the bundled-delay asynchronous pipeline using the mature timing engine from existing CAD tool. These constraints are supported by most CAD tools and support timing driven synthesis and auto place and route.

#### 3-3.1 Design flow

The signal in bundled-delay designs can be separated into two parts, the signal part and the control part. In this flow, we use the following CAD tools together with some .sdc constraint to complete the design for the 4-phase Muller pipeline and 2-phase MOUSETRAP pipeline. Table 3-2 shows the CAD tool and the constraints used in this asynchronous design flow.

Table 3-2 CAD tools and design constraints used in the asynchronous design flow

CAD Tool	Design Constraints
Tech: UMC 90nm	set_dont_touch
RTL: Verilog	set_size_only
Synthesis: Synopsys Design Compiler	set_max_delay
P&R: Cadence SOC Encounter	set_min_delay
Verification: Ultrasim (SPICE Model)	set_disable_timing

To start, we first construct a template model using standard cells at gate level for the handshake circuit (control part). For the Muller pipeline, it would be like the one shown in Fig. 3-8.

Structure modification may occur during synthesis when optimization, such as removing buffer or back-to-back inverters, breaking complex gate into simple gates, which may result in hazard or substantially modify necessary delay properties. Constraints “set\_dont\_touch” and “set\_size\_only” are used to prevent this from happening. The “set\_dont\_touch” command disallow the tools to change the cells in any kinds of ways while “set\_size\_only” command prevents change of cells but still allow the tool to change cell size of driving strength, delay, and power optimization.

```

1  `include "l90sprvt.v"
2
3  module HS(
4      output L_ACK,
5      input  L_REQ,
6      input  R_ACK,
7      output R_REQ,
8      output EN,
9      input  RSTn
10 );
11
12 wire w0, w1, w2, w3, ;
13
14 //  GATE
15 ND3MON  c_ele0_nand0 (.A(inv_out), .B(w3), .C(RSTn), .Z(w0) );
16 ND3MON  c_ele0_nand1 (.A(inv_out), .B(L_REQ), .C(RSTn), .Z(w1) );
17 ND3MON  c_ele0_nand2 (.A(L_REQ), .B(w3), .C(RSTn), .Z(w2) );
18 ND3MON  c_ele0_nand3 (.A(w0), .B(w1), .C(w2), .Z(w3) );
19
20 INVMON   hs0_inv (.A(R_ACK), .Z(inv_out));
21
22 assign R_REQ = w3;
23 assign L_ACK = w3;
24 assign EN   = w3;
25
26 endmodule

```

Fig. 3-8 A Verilog template for the Muller pipeline template

```

set_size_only -all_instance {*/c_ele0_nand0}
set_size_only -all_instance {*/c_ele0_nand1}
set_size_only -all_instance {*/c_ele0_nand2}
set_size_only -all_instance {*/c_ele0_nand3}
set_size_only -all_instance {*/hs0_inv}

```

Clock CAD tools use clock domain to optimize power and performance based on the defined frequency. These tools operate based on directed acyclic graph (DAG). If the timing got loops, algorithms in the tool are called to break it. In asynchronous design there are plenty of feedback loops either in the standard cell based C element or in the protocol cycles. Cutting these loops at the right place is necessary for correct and efficient timing analysis and optimization. The command “set\_disable\_timing” does the job. For example, we can use this command to break the acknowledge path that has no timing requirement but ends up in timing loops.

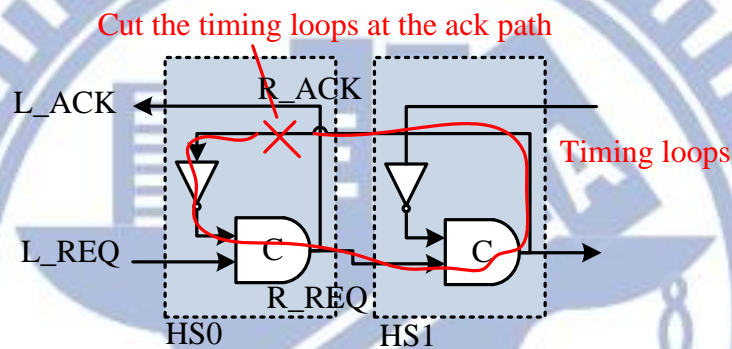


Fig. 3-9 Break timing loops for un-constraint path

`set_disable_timing -from HS1/L_ACK -to HS0/R_ACK`

Beside the control part, the registers and datapath can be generated using the standard flow used in synchronous design expect that in synchronous design, the maximum delay is decided by the required clock cycle. In bundled-delay asynchronous design, there's no clock defined. The maximum delay for the datapath and matched delay is set manually by the “set\_max\_delay” and “set\_min\_delay”.

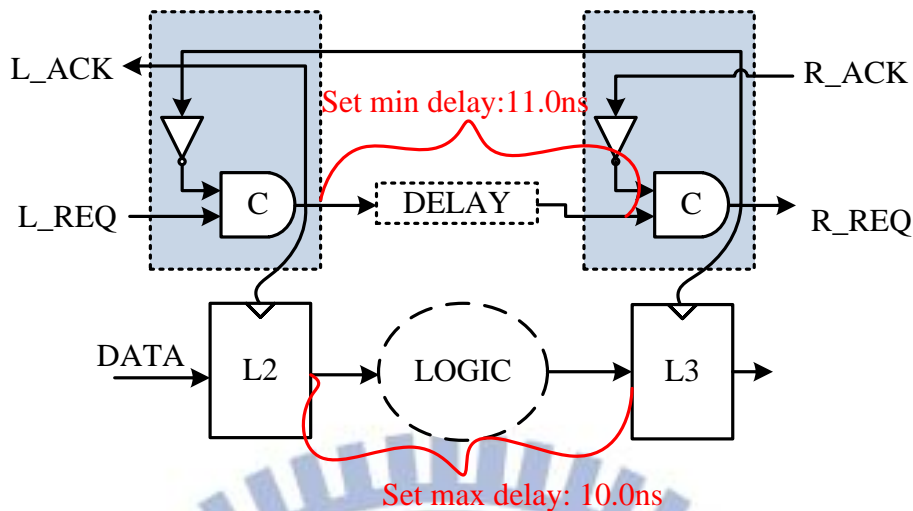


Fig. 3-10 Time constraints should be set manually at the desired start and end points

```

set_max_delay 10.0 -from L2_reg[*] -to L3_reg[*]
set_min_delay 10.5 -rise_from DELAY/IN -rise_to DELAY/OUT
set_max_delay 11.0 -rise_from DELAY/IN -rise_to DELAY/OUT

```

In this example the maximum delay from register L2 to register L3 is set to 10.0ns and the minimum delay is set to 10.5ns with a 5% safety margin. The next section will discuss the additional margin in detail based on Monte-Carlo simulation using UMC 90nm technology. After applying the above .sdc constraint, the design is ready for synthesis. The function is verified at gate-level. The same constraints can be applied again for backend place and route.

### 3-3.2 Delay Margin Tuning

Bundled-delay asynchronous relies on correct handshake and relative timing between delay line and datapath to compute the correct function. In the previous section, we mentioned that additional margin is required between the matched delay and datapath. Large margins results in wasted computation time but small margin may generate false output which cannot be solved. Here some Monte-Carlo simulation is

made to estimate the margin range and a lead-lag detector is proposed for margin tuning.

In the example, we extract the critical path in an  $8 \times 8 + 16$  bit multiplied accumulator (MAC) and produced a matched delay line using worst-case standard cell library. 5000 times Monte Carlo simulation is done with intra-die variance of 3 sigma at 3 different corner case ( $\{0.45V, -40^\circ C\}$ ,  $\{0.5V, 25^\circ C\}$ ,  $\{0.55V, 120^\circ C\}$ ). In the simulation, we assume the global voltage and temperature are the same because of the small design size. Fig. 3-11 shows the simulation result.

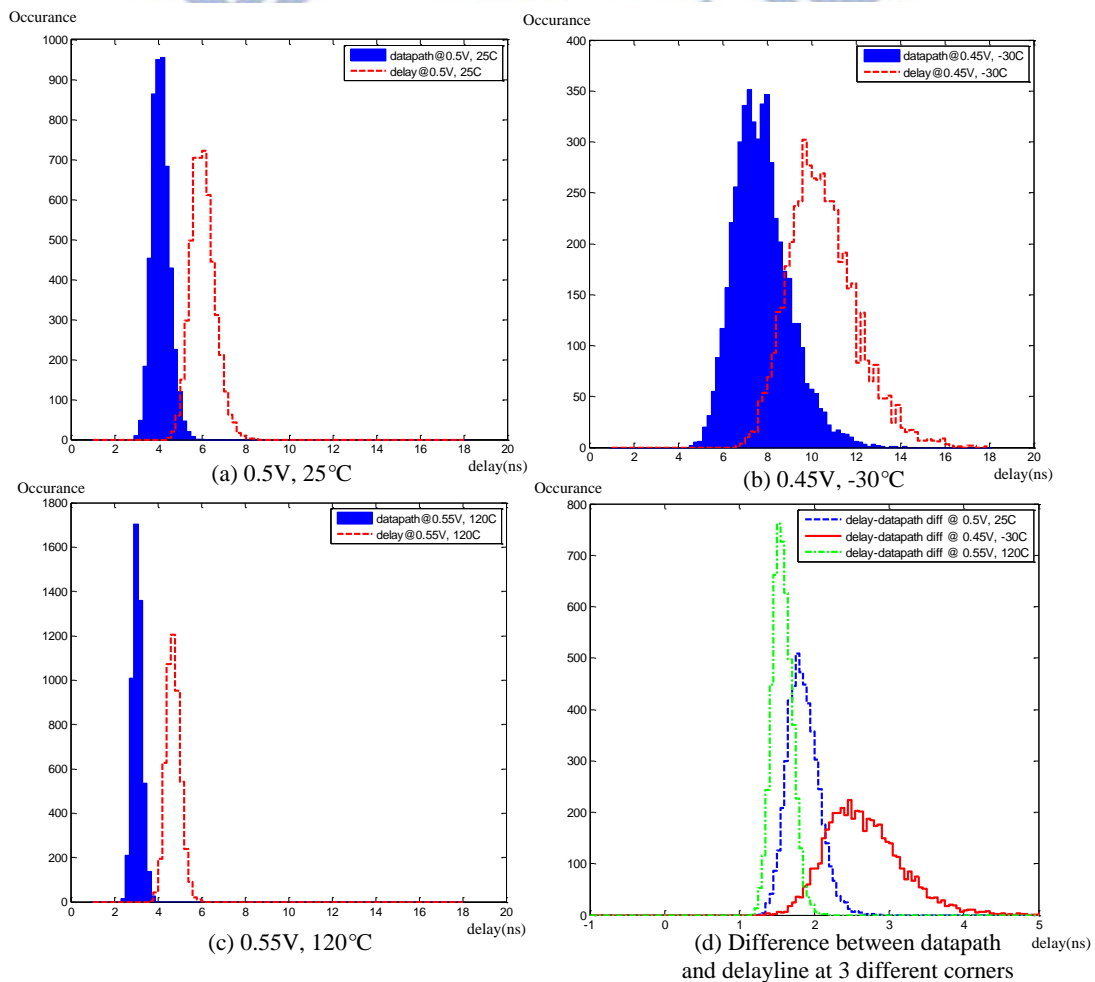


Fig. 3-11 The delay distribution of datapath and delay line at 3 different corner case (a) 0.5V, 25°C (b) 0.45V, -30°C (c) 0.55V, 120°C (d) Difference between datapath and delay line

From this figure we again verified that the delay variation increased in critical case (Fig. 3-11 (a)(b)(c)). Fig. 3-11 (d) shows the difference between matched delay line and the datapath. The difference is always positive, which ensures the function correctness. The delay line tracks the process variation within a margin from 1 to 4 ns, which is significantly smaller than synchronous design which always assumes worst case design condition.

To further minimize this margin, a lead-lag detector and a delay-tunable delay line is proposed. The lead-lag detector checks the relationship between the delay of the critical datapath and the delay line and chooses the best delay time. Fig. 3-12 shows the architecture of the lead-lag detector together with the tunable delay line.

The lead-lag relationship of the delay line and the datapath is detected using a simple D flip-flop. The output of the delay line is connected to the clock pin of the flip-flop and the output of the critical path is connected to the D pin. As shown in Fig. 3-12.

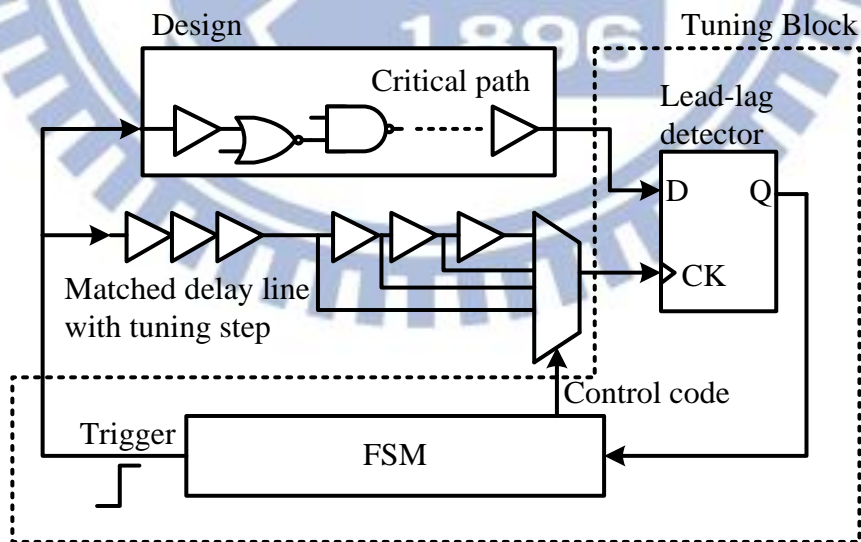


Fig. 3-12 Tuning Circuit including the tunable delay line and a lead-lag detector

The tuning process starts with the FSM setting the control code to the delay line's minimum delay. To start, a triggering rising edge is sent to the input of both the critical path and delay line at the same time. If the output of the delay line rises before the rising of critical path, the output of the D flip-flop will capture a zero indicating that the matched delay is shorter than the critical delay. The FSM will then change the control code for another iteration of testing until the sampled value of the flip-flop is one. Fig. 3-13(a) shows the tuning resolution at 3 corner cases for the same example of an 8×8+16bit MAC with critical path of 8 ns. Fig. 3-13 (b) shows the Monte-Carlo simulation after tuning. Comparing with Fig. 3-11(b), the difference between delay line and datapath is greatly reduced.

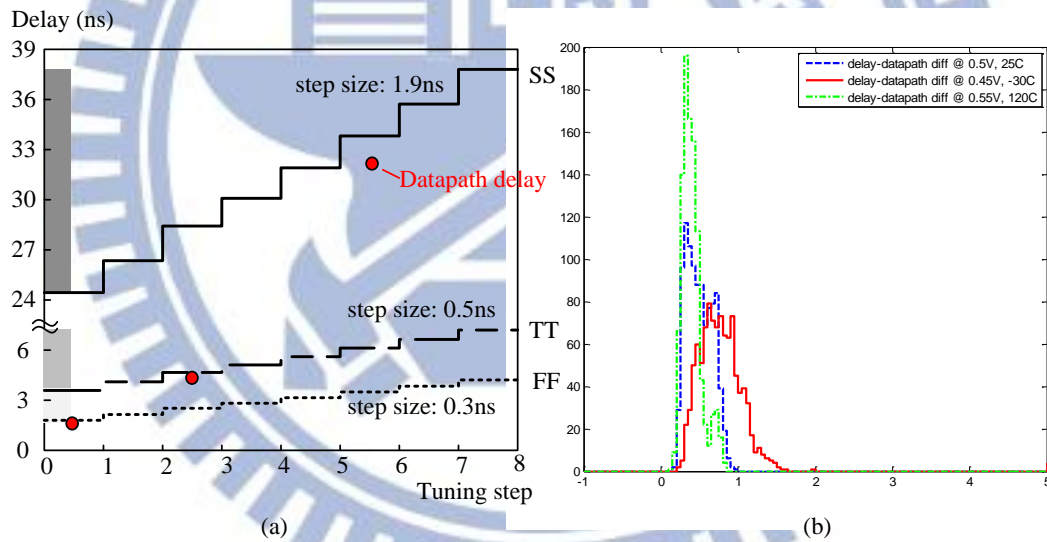


Fig. 3-13 (a) The 8 tuning steps at 3 corners (b) Reduced difference between datapath delay and delay line delay after tuning





handshake element are free for the data to flow inside the ring. Therefore, two data tokens and only one bubble token circulate inside the ring. The speed is limited by the bubble number. Two different Muller pipeline designs are implemented using different kinds of delay line cell (BUF delay and AND delay line), the performance comparison in respect of speed and energy are summarized in the next section.

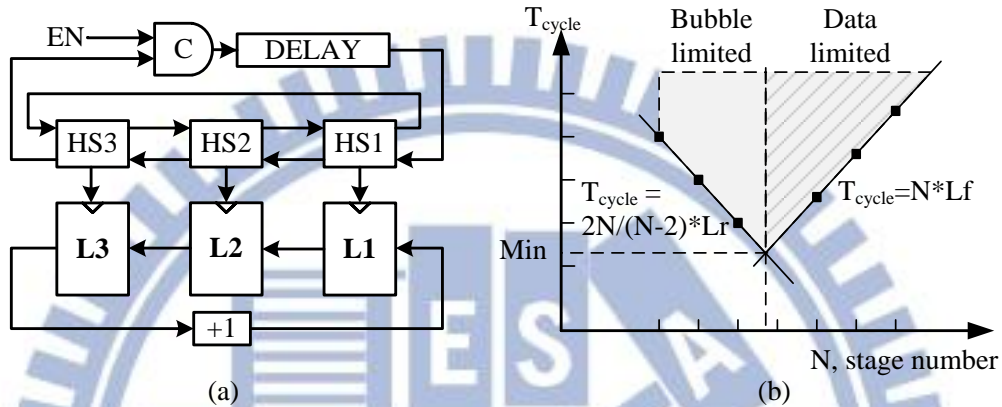
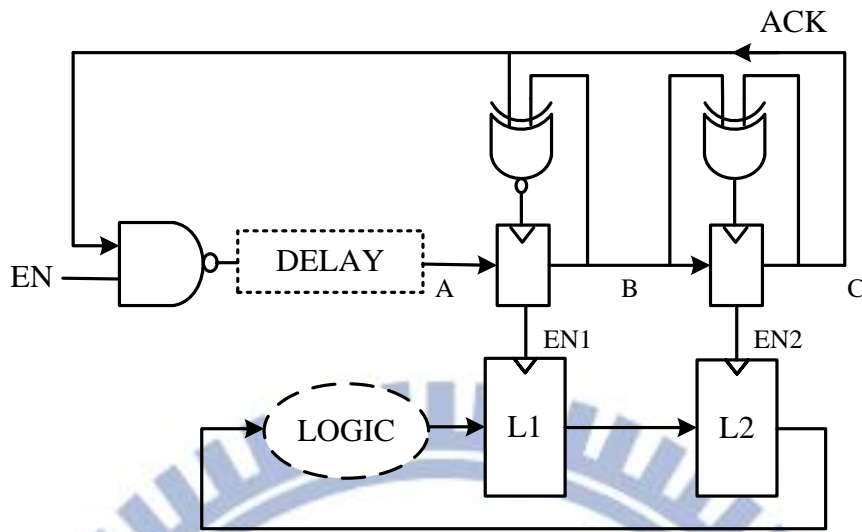
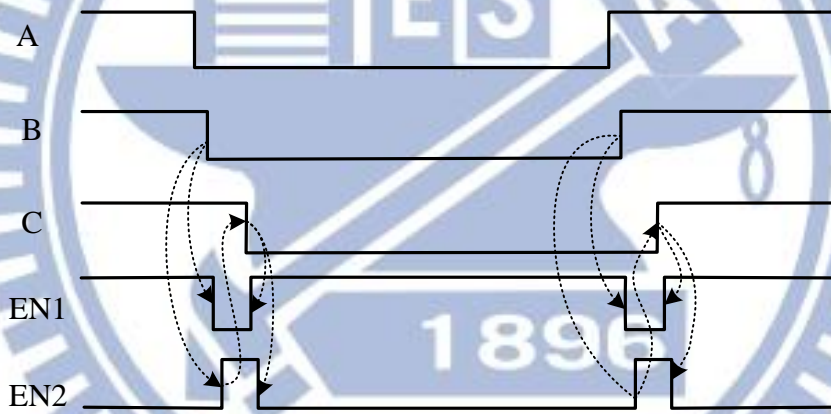


Fig. 3-15 (a) Ring structure for 4-phase Muller pipeline (b) The operating cycle time of the ring structure is decided by the available bubble and data number in the ring

The second structure (Fig. 3-16 (a)) utilizes the proposed ring structure modified based on the MOUSETRAP pipeline. In this structure, the XNOR gate in the second stage is replaced by an XOR gate. The XOR gate in the second stage distinguishes the difference of REQ signal before and after itself. If the REQ signals are different, it triggers the latch and propagates the REQ signal to the other side. After the propagation, the REQ signal becomes the same and latch becomes opaque. After the second stage receives the data, an ACK signal is sent back to start the next iteration. Fig. 3-16 (b) shows the time diagram of the handshake protocol. No extra return-to-zero operation is needed for this structure which saves the extra switching and time comparing with 4-phase protocol.



(a)



(b)

Fig. 3-16 The modified MOUSETRAP ring structure and time diagram

### 3-4.2 Performance Comparison

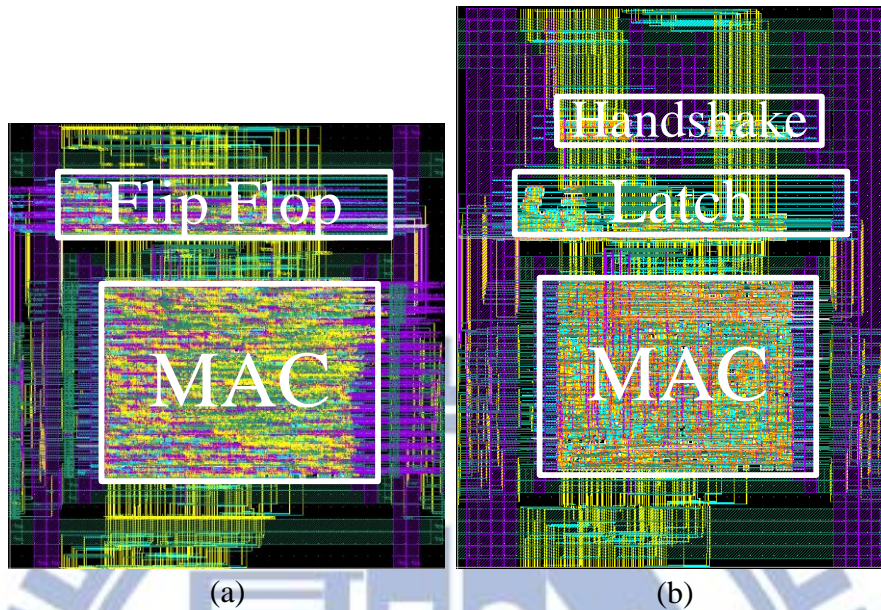


Fig. 3-17 Layout photo for the (a) synchronous and (b) asynchronous 16-tap FIR filter

The above mentioned three different implementations (Muller pipeline with BUF delay line, Muller pipeline with AND delay line, 2-phase modified MOUSETRAP) together with the synchronous design are realized using the design flow described in the previous section. The results are verified using SPICE model after P&R at 3 different corners. Fig. 3-17 shows the layout of the two designs. To make a fair comparison, the combinational part of the design is made into a HardMacro to eliminate the possible difference during APR. The three designs share the same combination HardMacro and floor plan. The only differences are that flip-flops in the synchronous design are replaced by latches in the asynchronous design. Additional handshake circuits are separated into another power domain for detail power analysis.

Table 3-3 and Fig. 3-18 first shows the energy distribution of the three different asynchronous implementations. Then Fig. 3-19 shows some operation detail of the three designs at different corners and temperature.

Table 3-3 Energy distribution for the 3 asynchronous pipeline

		Muller-BUF	Muller-AND	MMOUSE
	Operation time	275 ns	210 ns	141 ns
Energy (pJ)	Handshake	2.1 (9.9%)	2.0 (8.4%)	0.9 (4.9%)
	Delay	1.5 (7.1%)	3.25 (13.7%)	0.5 (2.7%)
	Datapath	17.6 (83%)	18.4 (77%)	17.0 (92%)
	Total (pJ)	21.2	23.7	18.4

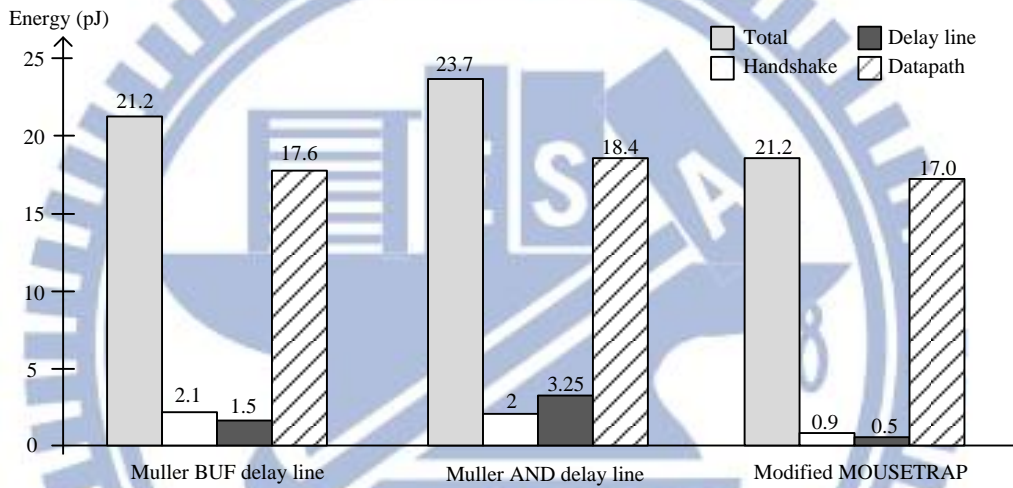


Fig. 3-18 Energy distribution of the three asynchronous implementation

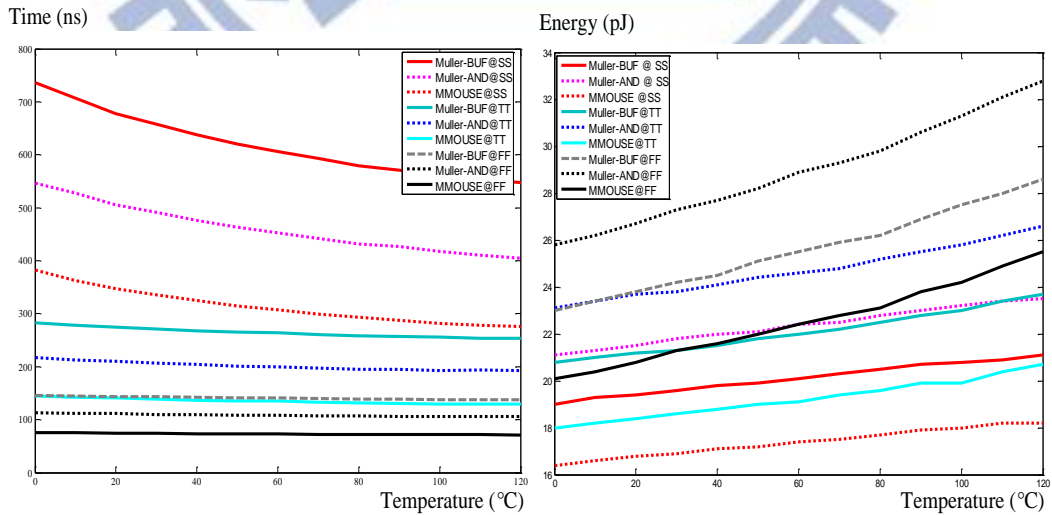


Fig. 3-19 Operation time and energy of the three asynchronous implementation at corner and temperature

From Table 3-3, the 4-phase Muller pipeline with BUF delay line requires the longest time to finish the computation time due to the extra return-to-zero time. The faster return-to-zero time for AND delay line speeds up the iterations. Among all, 2-phase modified MOUSETRAP design operates the fastest. Although a faster operating speed can be achieved by applying AND delay line, larger switching power of AND gates results in larger energy consumption than the BUF gates. Among the three implementations, the proposed 2-phase ring structure has less handshake and delay line overhead, resulting in faster operating speed and consumes less power than the 4-phase design.

Table 3-4 Comparison of the asynchronous design and synchronous design at TT, 25°C

		2-phased modified MOUSETRAP		Synchronous	
Operation time (ns)		141		288	
Energy (pJ)	Handshake	0.9	(4.9%)		
	Delay line	0.5	(2.7%)		
	Datapath	13.4	(72.8%)	14.6	(75%)
	Register	3.6	(19.5%)	4.9	(25%)
Total (pJ)		18.4		19.5	

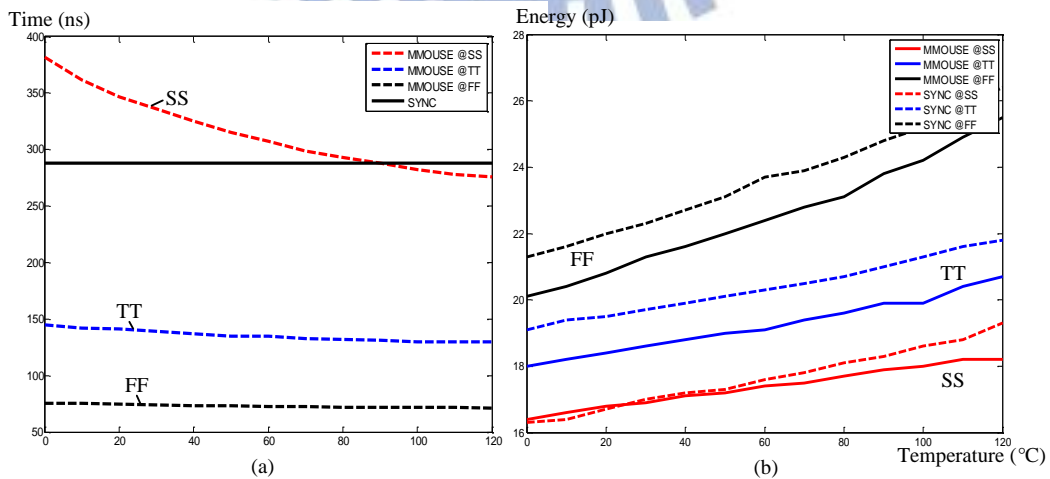
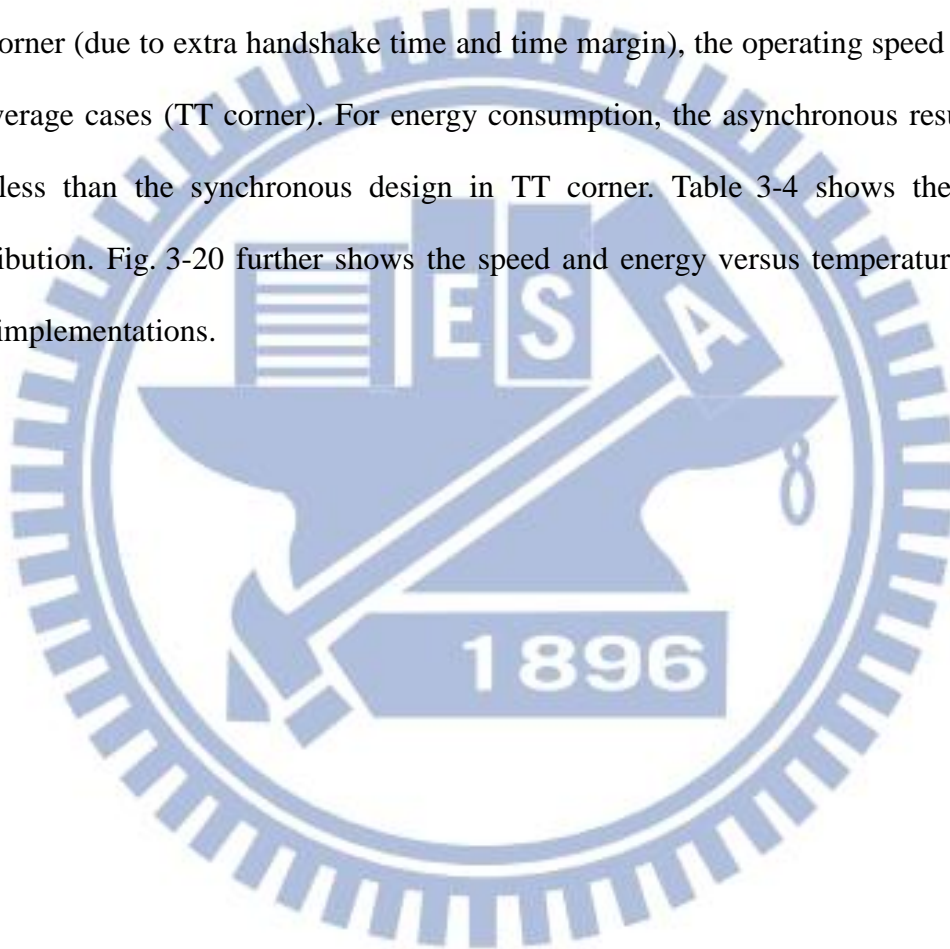


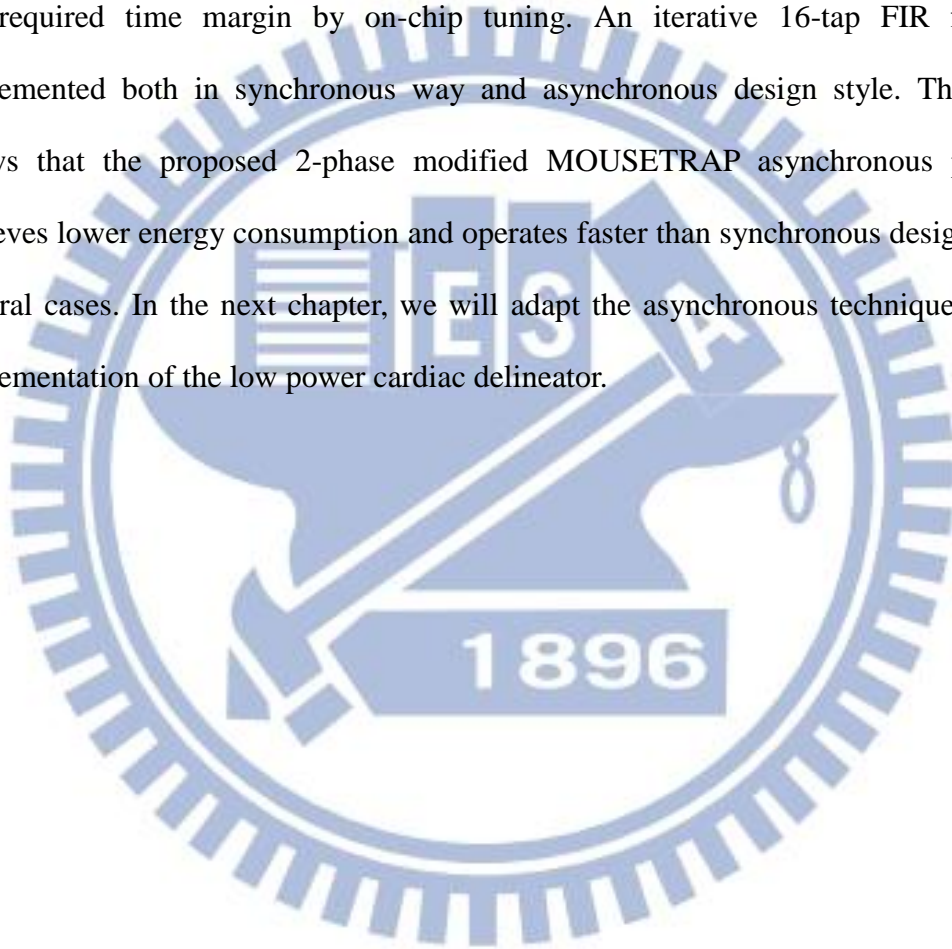
Fig. 3-20 Operation time and energy of the sync and async implementation at different corners/temperatures

Table 3-4 shows the comparison result (energy and operation time) of the modified MOUSETRAP asynchronous pipeline and synchronous at TT corner. The operation time for synchronous design is fixed to the worst case cycle time. From the figure we can see that although the asynchronous design operates slower than the synchronous at SS corner (due to extra handshake time and time margin), the operating speed is faster in average cases (TT corner). For energy consumption, the asynchronous results in a 5% less than the synchronous design in TT corner. Table 3-4 shows the energy distribution. Fig. 3-20 further shows the speed and energy versus temperature of the two implementations.



### 3-5 Summary

As the supply voltage is scaling down to reduce the system power consumption, asynchronous design is proven to be a good solution to combat the severe PVT variation. In this chapter we provide a design flow using commercial CAD tool to generate asynchronous design. Additional tuning circuit can be used to further reduce the required time margin by on-chip tuning. An iterative 16-tap FIR filter is implemented both in synchronous way and asynchronous design style. The result shows that the proposed 2-phase modified MOUSETRAP asynchronous pipeline achieves lower energy consumption and operates faster than synchronous design under general cases. In the next chapter, we will adapt the asynchronous techniques to the implementation of the low power cardiac delineator.





# Chapter 4: Architecture Design and Hardware Implementation

In this chapter we introduce the corresponded hardware for the algorithm described in Chapter 2. First we describe the architecture of the whole system, then the function block individually. Some low power techniques are used to further decrease the power consumption. Finally we compare our work with the state-of-the-art detection ASIC.

## 4-1 Architecture

With the algorithm optimized for hardware including reduced storage size and shared search rules, Fig. 4-1 shows the architecture for the proposed ECG delineator. This includes the DWT block and the feature extractor. The feature extractor can be further divided into QRS FSMs, boundary detector, register-based memory, adaptive threshold/window update engine, and an event-triggered asynchronous P/T wave search kernel.

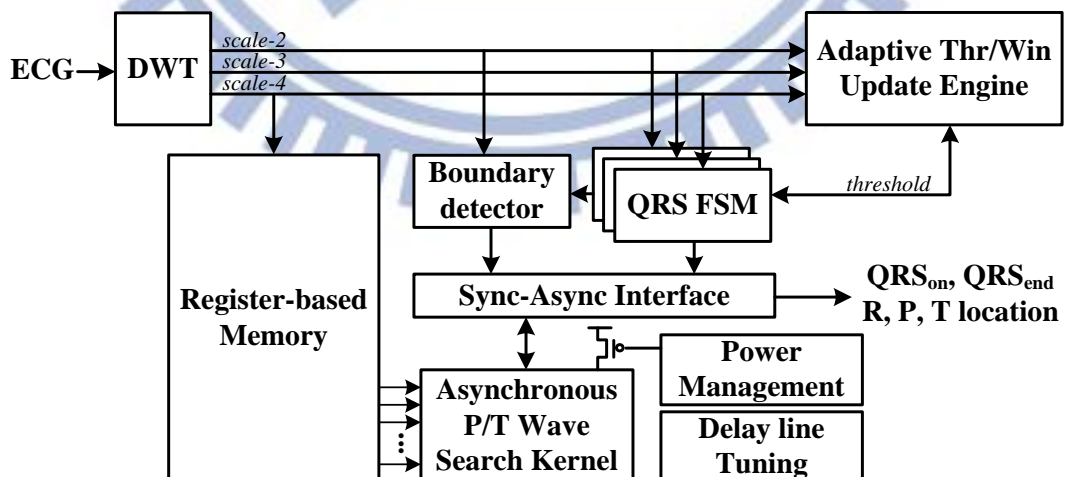


Fig. 4-1 Delineator architecture

The delineator operates at 250Hz to reduce switching power. To start, the input ECG goes through the 4-scale DWT block and is decomposed into 3 scales (scale-2, 3, 4). As the impulse response shown in (2-8) and (2-9), the DWT is implemented as a multiplier-free shift-and-add filter banks. The generated wavelet coefficients are used by the QRS FSMs, boundary detector, and adaptive threshold/window update engine. The coefficients of scales-4 are saved in the memory for future P wave search back.

Coefficients of scale 2, 3, 4 are used for the most important QRS detection. The detection of the QRS complex is performed by using FSMs to capture the temporal relationship of local min-max pair and the zero crossing points in different scales. Fig. 4-2 shows the state graph of the FSMs among receiving different ECG feature event.

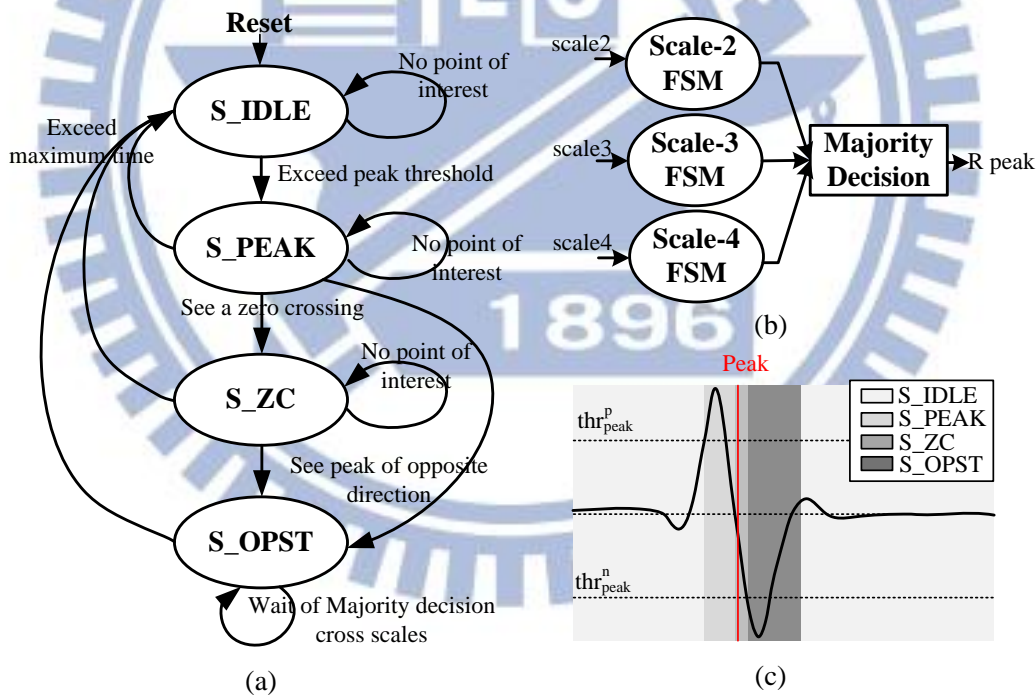


Fig. 4-2 (a) The state transition graph of the QRS FSM. (b) Majority decision is made according to the 3 FSMs (c) An example ECG waveform showing the state transition.

Besides the QRS FSMs, the boundary detector also works at full time searching for possible candidates for QRS start boundary ( $QRS_{on}$ ). According to the proposed algorithm, this can be done using several comparators searching for continuous samples under the boundary threshold. Once a QRS complex is confirmed, the nearest candidate will be confirmed as the starting boundary. This pre-detection reduces the storage for additional search back for  $QRS_{on}$ . After the detection of R peak, the boundary detector can then be reused to detect the  $QRS_{end}$ . Fig. 4-3 (a) shows the pre-detection process of  $QRS_{on}$  and following search for  $QRS_{end}$  after R peak. Fig. 4-3 (b) shows the architecture of the boundary detector consisting of only comparators and state machines.

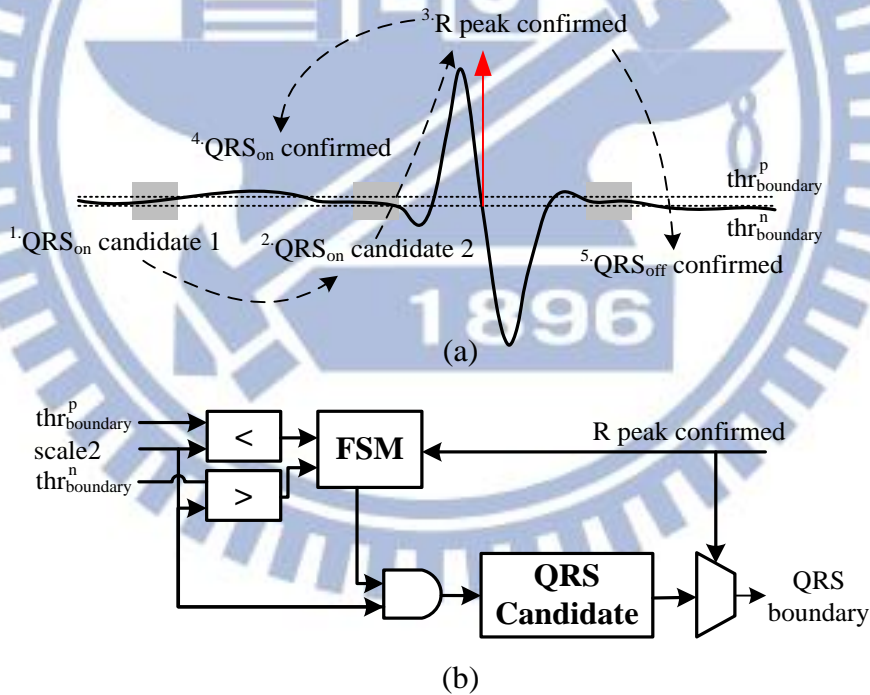


Fig. 4-3 (a) The pre-search process of  $QRS_{on}$  detection and the followed detection of  $QRS_{end}$  after R peak. (b) Architecture of the boundary detector

The adaptive THR/WIN update engine generates the peak /boundary threshold for the QRS FSMs and boundary detector and the search window for P/T detection. Upon every successful detection of QRS complex, the thresholds are updated based on the

signal and noise level. Designed avoiding complex mathematical operations, the multiplier-free engine consists of only adders and comparators and some registers to hold the current value. Fig. 4-4 shows the architecture of the adaptive THR/WIN update engine.

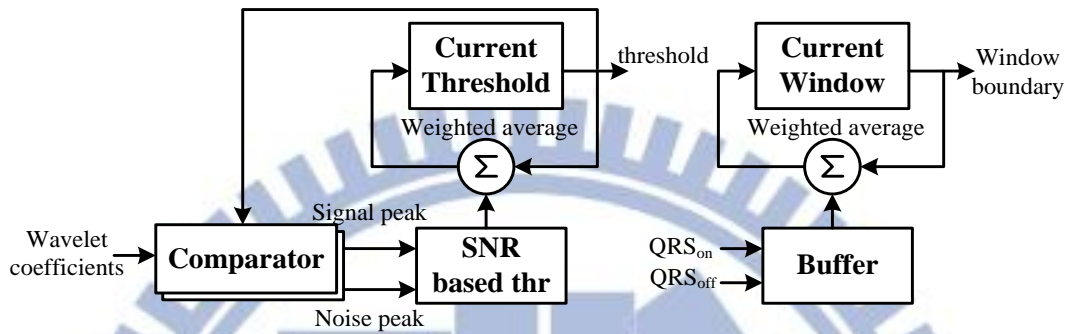


Fig. 4-4 The adaptive THR/WIN update engine

The search kernel for P and T wave is shared for the similar detection rules and is only activated after successful detection of QRS complex. For the detection of P and T wave, the scale-4 coefficients located in the search window must be first examined to locate the minimum and maximum point in order to check the wave existence then find the zero crossing point between them. Searching in an iterative approach, large cycles are needed to complete the detection. Optionally, two different implementations can be made. The first using doubled memory storage for saving the coefficients for P and T wave individually. The search is performed at system operating frequency (250Hz). The second implementation uses only half of the memory but operates in a higher operating speed in order to finish the P wave search before saving the T window coefficients.

Because of the low operating frequency, the total power is dominated by leakage. And among them, the storage element takes the largest percentage of the total power. The second implementation with less memory is clearly the better solution. With the

help of asynchronous techniques in Chapter 3, the asynchronous design consumes similar energy as synchronous design but eliminates the additional requirement for high frequency clock. The asynchronous search operates in an event-triggered way and can be shut down when not needed. The asynchronous search kernel also exhibits low latency for power up and down suitable for this application. Fig. 4-5 shows the time diagram showing the shared search kernel for P/T detection and the shot active duty cycle using asynchronous circuit. Additional on-chip tuning circuit proposed in Chapter 3 is also included in the delineator to acquire the best delay line margin to ensure computation efficiency and function correctness.

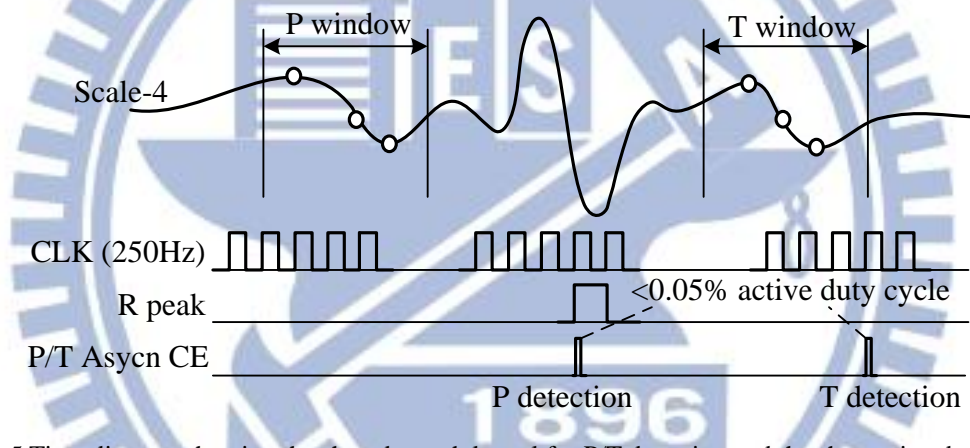


Fig. 4-5 Time diagram showing the shared search kernel for P/T detection and the shot active duty cycle using asynchronous circuit

A register-based memory of size 100x12bit is designed for this architecture. The decoder in the memory is moved into the asynchronous power domain to reduce the leakage power. By doing so, 4% of memory leakage power can be reduced.

A synchronous-asynchronous interface connects the data links between the synchronous and asynchronous portion. Fig. 4-6 shows the detail of the interface. Because all the input parameters are ready before the activation of the asynchronous kernel, no complex handshake is required at the input interface. As EN goes high, the loop formed inside the proposed asynchronous handshake circuits, starting the

iterative search. The search kernel searches the P/T wave by locating the min-max pair and zero crossing points. Every computation that requires iterative computations (e.g. counter, min-max searching) utilizes the two latches (slave and master) for correct computation. As soon as the wave is found, the VALID wire goes high and triggers the output register. The asynchronous kernel can then be shut down with very little latency reducing the leakage power.

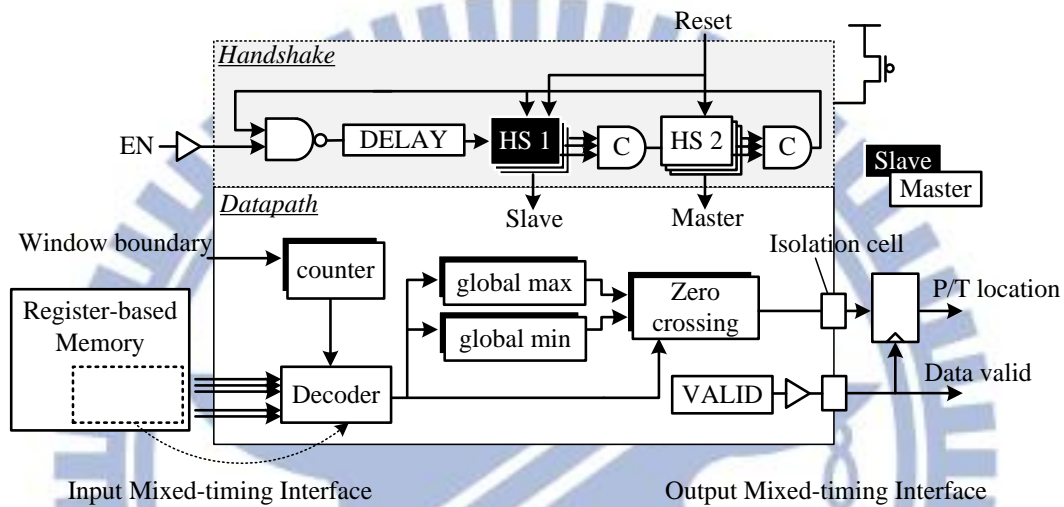


Fig. 4-6 The input and output interface for the asynchronous P/T kernel

## 4-2 Implementation Result

The proposed delineator is implemented using UMC 90nm CMOS technology and operates at 250Hz with input ECG of 12-bit resolution. Fig. 4-7 shows the power dissipation with the proposed low power strategies. By reduced wavelet scales and storage optimization in algorithm level, 45% power is saved. In addition, the introducing of event-triggered asynchronous search kernel further reduces half of the memory size and can be power gated reducing leakage power. This results in another 30.3% power reduction. Since the delineator power is dominated by the leakage, the usage of high  $V_T$  devices with 0.5V voltage scaling suppresses the static power. With optimization on algorithm, architecture, and circuit level, the average power of the

proposed ECG delineator is  $2.56\mu\text{W}$ . Fig. 4-7 also shows the power distribution, memory is still occupies the most power consumption.

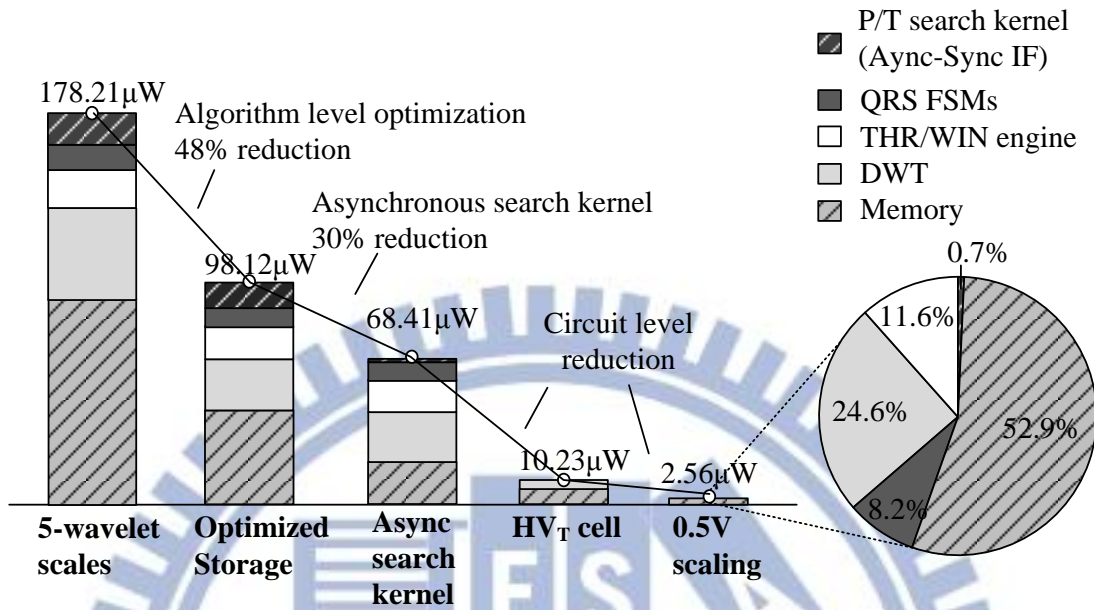


Fig. 4-7 Power reduction with strategy at different design level

Fig. 4-8 shows the layout photo of the proposed delineator. The area is  $250\mu\text{m}\times 250\mu\text{m}$ . The asynchronous P/T wave search kernel and the corresponded tuning circuit are separated in different power domains for individual power management.

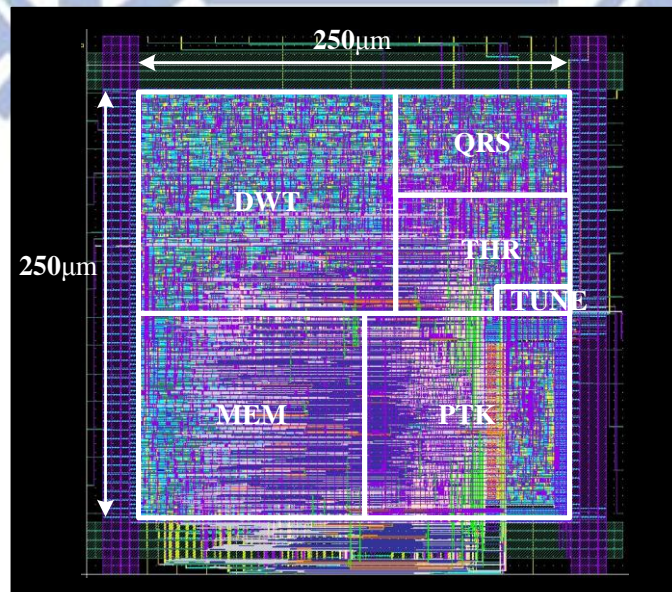


Fig. 4-8 Layout photo for the proposed cardiac delineator

### 4-3 Comparison with State-of-The-Art

Table 4-1 compares the proposed delineator with the state-of-the-art hardware cardiac feature extractor. All the existed designs only provide feature detection of R peaks [14]-[17], while the proposed design provides the detection of all 5 fiducial points (P, QRS<sub>on</sub>, R, QRS<sub>end</sub>, T). With the provided features, the syndromes number which can be detected is much more than the syndromes which can be classified using only R peaks (Table 1-1). This includes syndromes with high risk such as Myocardial Infarction. Operating at 0.5V, the average power is 2.56μW, which is fairly small comparing with the previous designs with only R peak detected.

Table 4-1 Comparison of the proposed delineator with the state-of-the-art detector

	<b>[14] 2009</b> <b>ASSCC</b>	<b>[15] 2009</b> <b>TBCAS</b>	<b>[16] 2010</b> <b>ISCAS</b>	<b>[17] 2012</b> <b>TBCAS</b>	<b>This</b> <b>Work</b>
Technology	0.18μm	0.35μm	0.18μm	0.35μm	90nm
Area (mm <sup>2</sup> )	1.1	N/A	0.68	1.11	0.06
Normalized Area (mm <sup>2</sup> )	0.28	N/A	0.17	0.07	0.06
Supply (V)	1.8	3.3	N/A	1.8	0.5
Frequency	1MHz	N/A	500Hz	300Hz	250Hz
Power	176μW	2.7μW	2.21μW	0.83μW	2.56μW
Features	R	R	R	R	P, R, T, QRSon, QRSend
R peak Se (%)	99.63%	99.81%	95.65%	99.31%	99.71%
R peak Pr (%)	99.89%	99.80%	99.36%	99.70%	99.68%



# Chapter 5:

## Prototype Construction

---

In this chapter, a wireless sensor node prototype is constructed using embedded micro-controller (MSP430) and commercial front-end, and a WIFI chip. The delineation result including the P, QRS<sub>on</sub>, R, QRS<sub>end</sub>, and T wave and can be shown on the computer with the recorded ECG. The prototype is tested both with real human measurement under mobile environment and MPS-450 ECG Simulator. The MPS-450 ECG Simulator can generate ECG signal with various syndrome and noise.

### 5-1 Experiment Platform and System

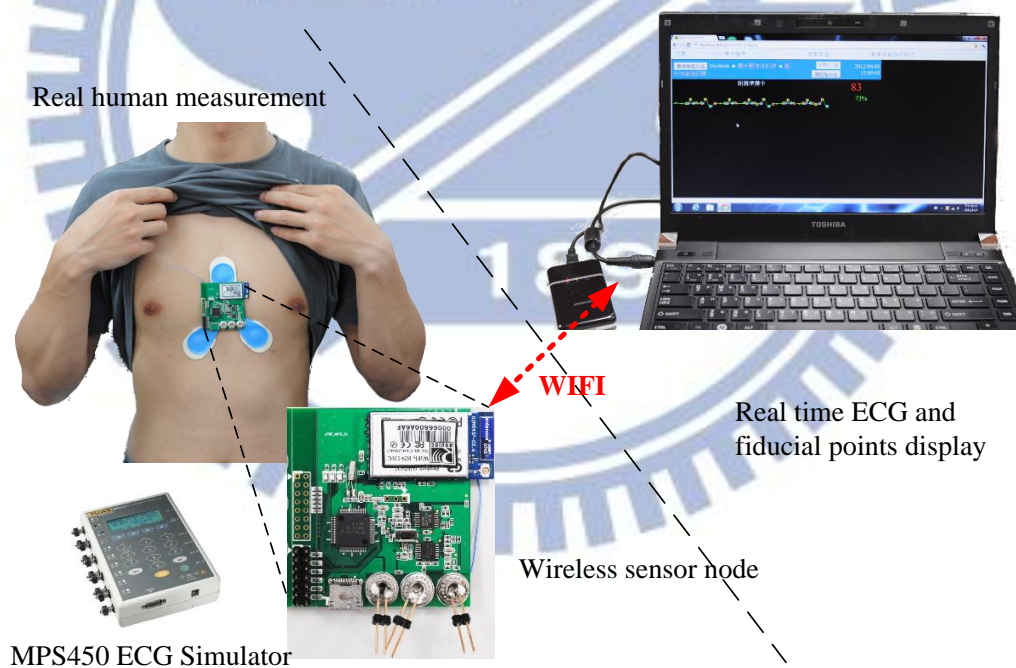


Fig. 5-1 Experiment environment setup

The experiment platform includes wireless sensor with WIFI function, which is able for real-time one-lead ECG processing and transmission. Fig. 5-1 shows the experiment platform. Three electrodes are required for one lead measurement. Two nodes are used to measure the voltage difference, and the third one is used for voltage reference. In order to verify the delineation result, we use the Fluke ECG simulator MPS450 to simulate the different kinds of arrhythmia including respiration.

## 5-2 On-Sensor ECG Delineator

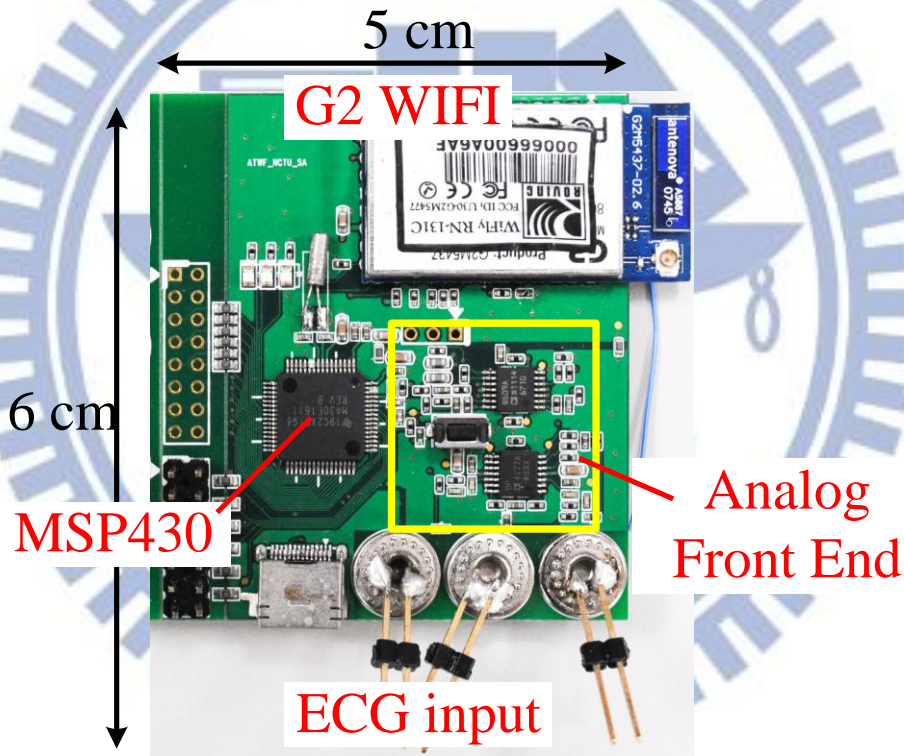


Fig. 5-2 The prototype wireless sensor

The sensor node consists of three main parts built using commercial products:

- Power IC: LTI1761IS5-3.0, LTC1046IS8, ADP121-AUJZ33R7
- Analog front end(AFE): AD8221ARMZ, AD8609ARUZ
- MSP430 microcontroller: TI-MSP430-F1611
- WIFI module: G2-M5437 Wi-Fi SoC

The amplitude of human ECG is approximately 1~5mV. Because of this small amplitude, we need the analog front end for noise filtering and signal amplification. After noise removal and amplification, ECG is sampled at 250Hz using the ADC in MSP430 microcontroller. The MSP430-F1611 is a microcontroller with 16-bit RISC architecture and maximum 8MHz operating clock built with 48KB+256B Flash memory and 10KB RAM. Five power saving modes is provided, each with different modules and internal clock turned OFF. Fig. 5-3 shows the average current of the five modes at two different supply voltages respectively. MSP430 also integrates features like a three-channel internal DMA, eight-channel 12-bit Analog-to-Digital converter (ADC), and standard serial communication interface (SPI, USART).

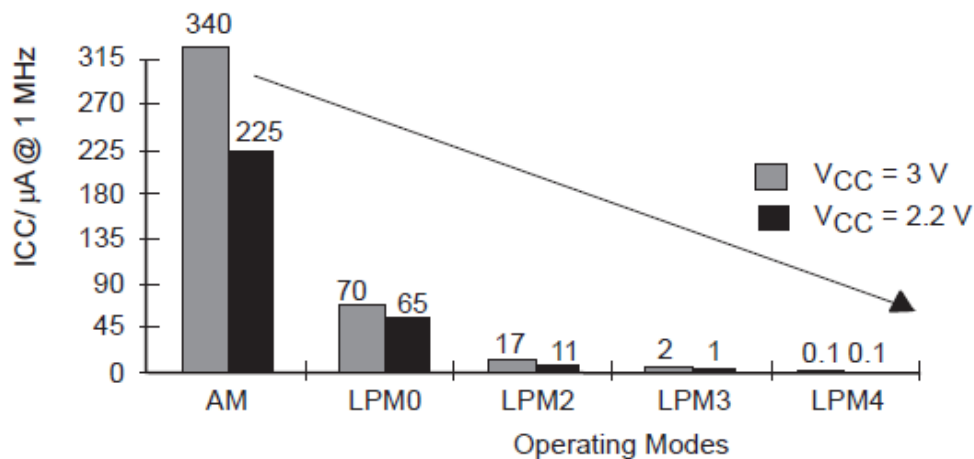


Fig. 5-3 Current of the 5 different Power mode supported by MPS430 microcontroller

The operation flow for real-time ECG delineation is like this. After startup, the MSP430 operates at the low power mode 0 (LPM0), with the master clock (MCLK) turned OFF. The ADC module starts to sample the ECG at 250Hz using an internal counter with 12 bits resolution. At each sample, the DMA module continuously moves the sampled data to ECG buffer (seg.1). After moving 4 seconds of ECG data (1000 samples) to the RAM, an interrupt is sent by the DMA module to wake the CPU. CPU then performs the wavelet-based ECG delineation algorithm proposed in Chapter 2:

The delineation result is written back to the RAM and tagged after the ECG raw data. After completion of the delineation, CPU goes to LPM0 again. A request signal is sent to G2-WIFI module using the SPI. Using the handshake protocol depicted in Fig. 5-6, DMA is again used to move the ECG raw data together with the delineation result to the SPI, and are transmitted to the WIFI module. Fig. 5-5 shows the packet format. The memory is separated into 2 segments and is written and read in a Ping-Pong buffering way for this real time sampling, delineation, and transmission process. Fig. 5-4 shows the module used during the whole process.

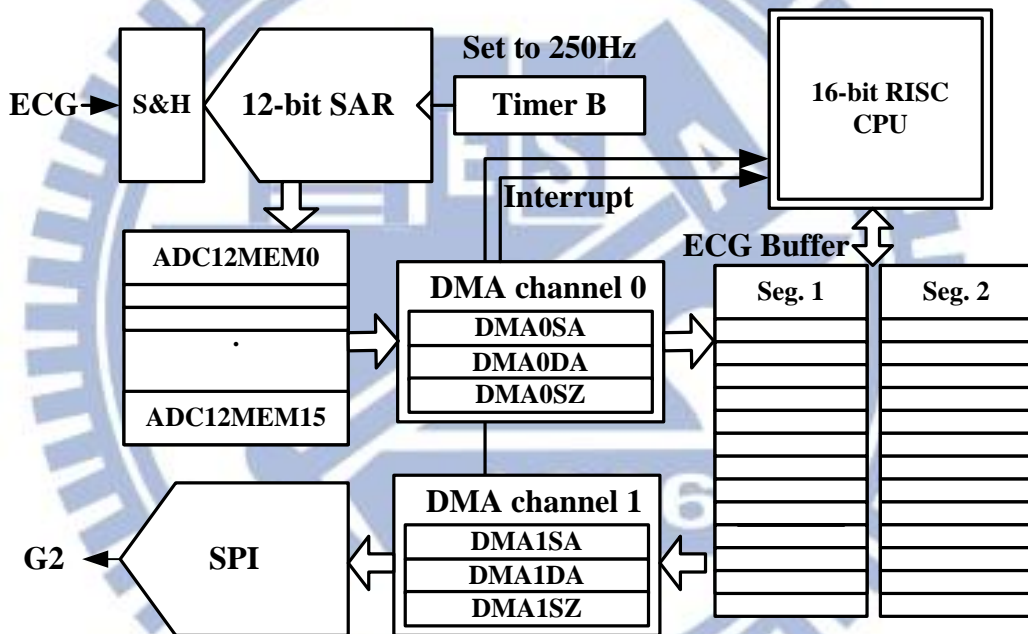


Fig. 5-4 The used module and delineation flow inside MSP430

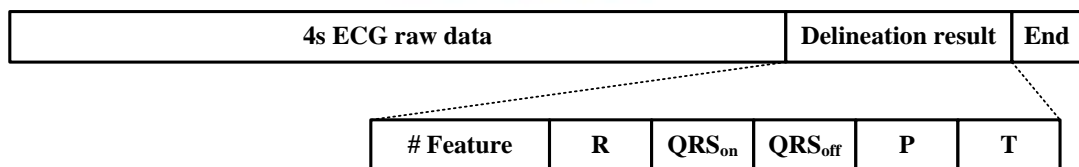


Fig. 5-5 The transmitted packet format

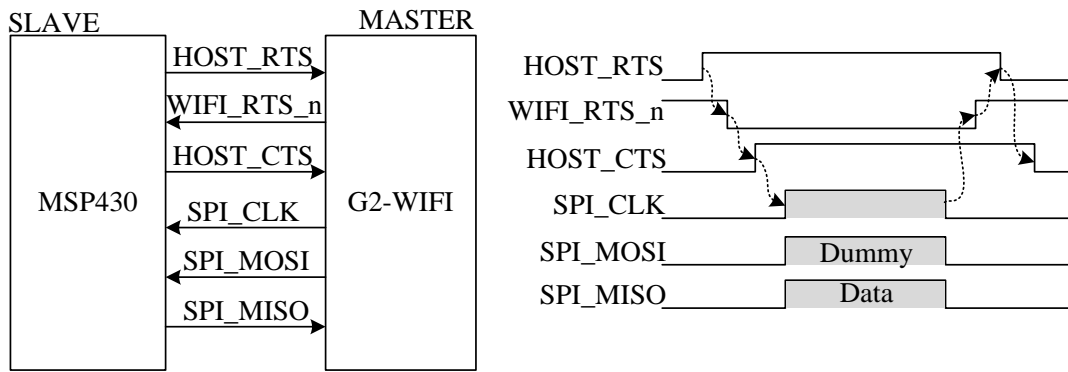


Fig. 5-6 SPI handshake protocol between MSP430 and G2 WIFI module

Fig. 5-6 shows the connection and handshake protocol between MSP430 and the WIFI module. The G2-WIFI remains in idle mode and is awaked every 4 seconds for data transmission.

### 5-3 Emulation Result

The prototype including the wireless link and the delineation part is tested. The delineation algorithm is tested both with real human measurement and the MPS-450 ECG Simulator. Fig. 5-7 shows the user interface of real time ECG and extracted fiducial points display.

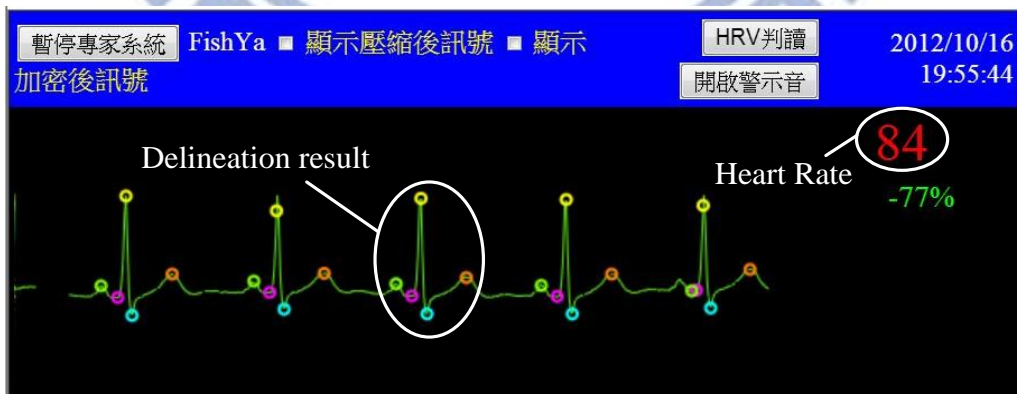


Fig. 5-7 User interface for real time display of ECG and extracted fiducial points

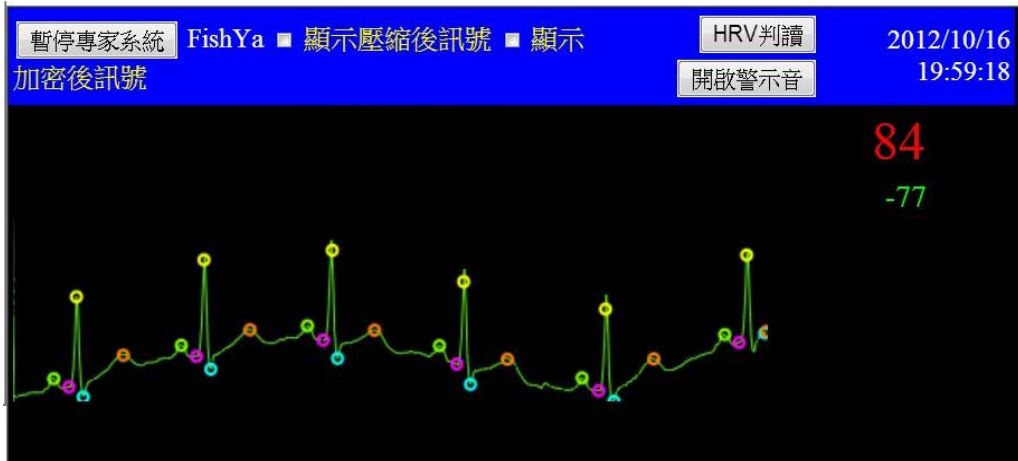


Fig. 5-8 Delineation Under mobile environment (with baseline drift)

Fig. 5-8 shows that the proposed delineator provides the correct delineation result under mobile environment. With the proposed delineation, 95% of the data can be reduced, which may result in a large reduction on wireless transmitting energy.

# ***Chapter 6:***

## ***Conclusion and Future Work***

---

### 6-1 Conclusion

In this work, a low power wavelet-based ECG delineator is proposed for mobile healthcare applications. The transmission energy of the wireless sensor node can be greatly reduced by sending the extracted features only, thus increasing the monitoring time. The proposed delineator detects the ECG fiducial points including the P, QRS<sub>on</sub>, R, QRS<sub>end</sub>, and T waves. With all important features extracted, 8 different categories of heart syndromes can be detected providing instant alarm or feedback to users.

The wavelet preprocess eliminates the noise interference for mobile monitoring. With adaptive threshold and search window generation considering noise level, the proposed algorithm achieves the detection accuracy over 99% for all provided feature. The algorithm is designed with storage optimization and avoiding complex operations, making it suitable for hardware implementation.

Because of the different occurrence of features in time axis, the search kernel is shared and event-triggered only when needed. Operating using the proposed 2-phase asynchronous ring structure, the search kernel can be turned ON and OFF with little latency without the needs of additional high frequency clocks, making it suitable for system integration. A design flow is built and tested using commercial CAD tool for the construction of asynchronous search kernel. The asynchronous design also enables the possibility for the delineator to operate under low supply voltage with the ability to combat the severe PVT variation. Implemented in UMC 90nm technology with voltage scaling to 0.5V, the overall power is 2.56 $\mu$ W for real time ECG transmission.

Finally, a wireless sensor node prototype is constructed using microcontroller MSP430 with WIFI transmission to verify the concept and algorithm for real time mobile on-sensor ECG delineation.

## 6-2 Future Work

In the future, based on the extracted ECG feature, on-sensor syndrome classification can be added to the system to make the healthcare application more complete. Instant result can be feedback to the user including health status information and advices. The techniques and concepts used in this thesis such as the wavelet noise removal, adaptive threshold generation and event-triggered asynchronous search kernel can also be applied to different kinds of bio-signals such as electromyography (EMG) or blood pressure for robust and efficient human body monitoring system.

Asynchronous circuit is clearly one of the techniques to combat PVT variation in future designs when going for low supply voltage. We are looking forward to exploit more efficient asynchronous design, and analysis the performance in sub-threshold region. New design flow for more convenient asynchronous designs is always an interesting research field. With improved and reliable design flow, we will be able to apply asynchronous circuit to larger design, making sub-threshold design an easy option for circuit designer to achieve lowest possible power consumption.



# Reference

---

- [1] R. J. Prineas, R. S. Crow, Z. M. Zhang, “The Minnesota Code Manual of Electrocardiographic Findings,” Wright, 2nd ed., 2010, XIV.
- [2] Malcolm S. Thaler, “The Only EKG Book You'll Ever Need 5<sup>th</sup>, ed.”, Lippincott Williams & Wilkins, 2009.
- [3] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, et al., PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* 101(23):e215-e220 [Circulation Electronic Pages; <http://circ.ahajournals.org/cgi/content/full/101/23/e215>]; 2000 (June 13).
- [4] Moody GB, Mark RG. “The impact of the MIT-BIH Arrhythmia Database,” *IEEE Eng in Med and Biol* 20(3):45-50 (May-June 2001).
- [5] Laguna P, Mark RG, Goldberger AL, Moody GB., “A Database for Evaluation of Algorithms for Measurement of QT and Other Waveform Intervals in the ECG,” *Computers in Cardiology* 24:673-676 (1997).
- [6] V. X. Afonso, W. J. Tompkins, T. Q. Nguyen, and L. Shen, “ECG beat detection using filter banks,” *IEEE Trans. Biomed. Eng.*, vol. 46, pp. 192–202, 1999.
- [7] R. Poli, S. Cagnoni, and G. Valli, “Genetic design of optimum linear and nonlinear QRS detectors,” *IEEE Trans. Biomed. Eng.*, vol. 42, pp. 1137–1141, 1995.
- [8] P. E. Trahanias, “An approach to QRS complex detection using mathematical morphology,” *IEEE Trans. Biomed. Eng.*, vol. 40, pp. 201–205, 1993.
- [9] J. P. Martinez, R. Almeida, S. Olmos, A. P. Rocha, and P. Laguna, “A wavelet-based ECG delineator: Evaluation on standard databases,” *IEEE Trans.*

- Biomed. Eng.*, vol. 51, pp. 570–581, 2004.
- [10] A. Martínez, R. Alcaraz, and J.J. Rieta, "Application of the phasor transform for automatic delineation of single-lead ECG fiducial points," *Physiological Measurement*, vol.31, no.11, pp. 1467–1485, 2010.
- [11] S. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE Trans. Acoust. Signal Processing*, vol. 37, pp. 2091–2110, Dec. 1989.
- [12] A. Cohen and J. Kovačević, "Wavelets: The mathematical background," *Proc. IEEE*, vol. 84, pp. 514–522, Apr. 1996.
- [13] S. Mallat and S. Zhong, "Characterization of signals from multiscale edge," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 710–732, July 1992.
- [14] M. W. Phyu, Y. Zheng, B. Zhao, L. Xin, and Y. S. Wang, "A real-time ECG QRS detection ASIC based on wavelet multiscale analysis," *IEEE Asian Solid-State Circuits Conf.*, pp. 293–296, 2009.
- [15] F. Zhang and Y. Lian, "QRS detection based on multiscale mathematical morphology for wearable ECG devices in body area networks," *IEEE Trans. Biomed. Circuits Syst.*, vol. 3, pp. 220–228, 2009.
- [16] H. M. Wang, Y. L. Lai, M. C. Hou, S. H. Lin, B. S. Yen, Y. C. Huang, et al., "A  $\pm 6$ ms-accuracy,  $0.68 \text{ mm}^2$  and  $2.21 \mu\text{W}$  QRS detection ASIC," *IEEE Int. Symp. Circuits and Systems*, pp. 1372–1375, 2010.
- [17] C. I. Jeong, P. I. Mak, C. P. Lam, C. Dong, M. I. Vai, P. U. Mak, S. H. Pun, F. Wan, and R. P. Martins, "A  $0.83\text{-}\mu\text{W}$  QRS Detection Processor Using Quadratic Spline Wavelet Transform for Wireless ECG Acquisition in  $0.35\text{-}\mu\text{m}$  CMOS", " *IEEE Transactions on Biomedical Circuits and Systems*, pp.1, 0, no.99., vol.PP.
- [18] M. Singh, J. A. Tierno, A. Rylyakov, S. Rylov, and S. M. Nowick, "An

Adaptively Pipelined Mixed Synchronous-Asynchronous Digital FIR Filter Chip Operating at 1.3 Gigahertz," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol.18, no.7, pp.1043-1056, July 2010

- [19] I. J. Chang, S. P. Park, and K. Roy, "Exploring Asynchronous Design Techniques for Process-Tolerant and Energy-Efficient Subthreshold Operation," *IEEE Journal of Solid-State Circuits*, vol.45, no.2, pp.401-410, Feb. 2010
- [20] D.E. Muller, W.S. Bartky, "A theory of asynchronous circuits," *Proceedings of an International Symposium on the Theory of Switching*, Cambridge, April 1957, Part I, pages 204–243. Harvard University Press, 1959. The annals of the computation laboratory of Harvard University, Volume XXIX
- [21] M. Singh, S. M. Nowick, "MOUSETRAP: High-Speed Transition-Signaling Asynchronous Pipelines," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol.15, no.6, pp.684-698, June 2007
- [22] K. S. Stevens, Y. Xu, and V. Vij, "Characterization of Asynchronous Templates for Integration into Clocked CAD Flows," *IEEE Symposium on Asynchronous Circuits and Systems*, pp.151-161, 17-20 May 2009
- [23] J. Sparsø, S. Furber. "Principles of Asynchronous Circuit Design – A System Perspective," Norwell, MA: Kluwer, 2001
- [24] T. W. Chen, J. Y. Yu, C. Y. Yu, and C. Y. Lee, "A 0.5 V 4.85 Mbps Dual-Mode Baseband Transceiver With Extended Frequency Calibration for Biotelemetry Applications," *IEEE Journal of Solid-State Circuits*, vol. 44, pp. 2966-2976, Nov. 2009.