

國立臺灣大學電機資訊學院生醫電子與資訊學研究所

碩士論文

Graduate Institute of Biomedical Electronics and Bioinformatics

College of Electrical Engineering and Computer Science

National Taiwan University


Master Thesis

基於 FPGA 的部份可重置技術之

心律不整診斷支援系統

An Arrhythmia Diagnosis Support System Base on

Partial Reconfiguration of FPGA



許宏瑞

Hung-Jui Hsu

指導教授：張璞曾 教授 林伯星 教授

Advisor : Fok-Ching Chong Bor-Shing Lin

中華民國 101 年 7 月

July, 2012

國立臺灣大學碩士學位論文
口試委員會審定書

基於 FPGA 的部份可重置技術之心律不整診斷支援系統
An Arrhythmia Diagnosis Support System Base on Partial
Reconfiguration of FPGA

本論文係許宏瑞 (R99945033) 在國立臺灣大學生醫電子與資訊學研究所完成之碩士學位論文，於民國 101 年 7 月 28 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

洪謹常

(指導教授)

余松年

林作星

詹曉龍

所長：

賴中龍

摘要

心電圖(Electrocardiography)是診斷心律不整最重要的生理訊號，目前相關研究所提出的心電圖之系統架構，大致可以分為兩類，分別是診斷型和記錄型；診斷型系統是本身具有判斷病徵的功能(例如心律不整)，當診斷到疑似異常的心電圖訊號時，就發出警示，並將心電圖資料傳送給醫護人員；記錄型系統則是直接儲存或是傳送原始的心電圖訊號，但是這將有資料量過大的問題，所以系統通常會將資料做壓縮的處理。

本研究提出一個整合診斷與壓縮模組之架構，其同時具備上述系統之優點；我們將此架構實現在 FPGA 平台上，由於 FPGA 能夠高效率的處理複雜的運算，並維持良好的處理速度，因此系統能夠做到即時(Real-Time)的訊號處理；另外，我們更利用 FPGA 的部份可重置(Partial Reconfiguration)技術，進一步降低系統的使用資源、成本、以及功率消耗；FPGA 的系統架構亦適合被轉換成 ASIC，將更適合作為可攜式的生理監控裝置。

關鍵詞：FPGA、部份可重置、心電圖、心律不整、壓縮

ABSTRACT

Electrocardiography is an important method used in physiological signals to detect arrhythmia. At present, related researches in the architecture of Electrocardiography system can be classed as two kinds: diagnosis support system and recording system. Diagnosis support system has the ability to detect suspected arrhythmia. When the system detects abnormal signals, it will send alarm or begin to transmit the detected signals to the hospital. However, the recording system will directly store the original signal into the memory or sent all the data to hospital. But the amount of data may be too huge. Generally, the system will transmit the data after compression.

Our research proposes an integrated diagnosis support system with compression. It has the advantages of the previous two types of systems. We implement the architecture base on FPGA. Therefore, the system can achieve real-time work since the FPGA has high performance computing ability and speed. Furthermore, partial reconfiguration technology of FPGA is used to save internal logic, power consumption, and cost. When the system is converted to application-specific integrated circuit (ASIC), it will be appropriate to used as a portable device.

Key words:, Field Programmable Gate Array (FPGA), Partial Reconfiguration,

Electrocardiography, Arrhythmia, Compression

總目錄

口試委員會審定書.....	i
摘要.....	ii
ABSTRACT.....	iii
總目錄.....	iv
圖目錄.....	vii
表目錄.....	x
第一章 緒論.....	1
1.1 心電圖簡介.....	1
1.1.1 心電圖原理與功能.....	1
1.1.2 心律不整.....	2
1.2 研究動機.....	4
1.3 研究目的.....	6
1.4 論文架構.....	7
第二章 背景知識.....	8
2.1 FPGA 基本概念.....	8
2.1.1 FPGA 簡介.....	8
2.1.2 SOPC 系統.....	11
2.1.3 MicroBlaze 處理器.....	11
2.2 Xilinx Virtex-6 ML605.....	13
2.3 設計軟體與流程.....	14
2.3.1 System Generator.....	14
2.3.2 Xilinx ISE.....	17
2.3.3 EDK.....	18

2.3.4	PlanAhead.....	21
2.4	配置介面.....	22
2.4.1	JTAG 與 SelectMAP.....	22
2.4.2	ICAP.....	22
2.5	部份可重置技術.....	24
2.5.1	基本概念.....	24
2.5.2	相關研究.....	25
第三章	系統設計.....	28
3.1	系統概述.....	28
3.2	監測模組.....	31
3.2.1	心電圖 R 波偵測.....	31
3.2.2	心律不整判斷.....	35
3.2.3	模組設計.....	40
3.2.3.1	斜率與初始閾值計算.....	41
3.2.3.2	R 波偵測與閾值更新.....	43
3.2.3.3	心律不整判斷.....	46
3.3	壓縮模組.....	50
3.3.1	離散小波轉換.....	51
3.3.2	Run-Length 編碼.....	55
3.3.3	模組設計.....	56
3.3.3.1	資料擷取與位移.....	57
3.3.3.2	5/3 整數對整數小波轉換.....	59
3.3.3.3	閾值篩選.....	62
3.3.3.4	Run-Length 編碼.....	64
3.4	MicroBlaze 處理器系統模組.....	66

第四章	實驗結果與討論	69
4.1	實驗資料與平台	69
4.1.1	MIT-BIH 心律不整資料庫.....	69
4.2	實驗結果.....	71
4.2.1	R 波偵測	71
4.2.2	心律不整判斷	75
4.2.3	心電圖訊號壓縮.....	78
4.2.4	系統效能量測	83
第五章	結論.....	88
參考文獻	90



圖目錄

圖 1.1.1 正常心電圖圖解.....	1
圖 1.1.2 心臟圖	3
圖 2.1.1 FPGA 內部結構.....	9
圖 2.1.2 LUT 示意圖	10
圖 2.1.3 FPGA 設計示意圖	10
圖 2.1.4 MicroBlaze 內部架構[12].....	12
圖 2.3.1 System Generator 的作用	14
圖 2.3.2 System Generator 設計實例示意圖[13].....	15
圖 2.3.3 System Generator 的 FPGA 開發流程[13].....	16
圖 2.3.4 軟體 ISE 開發流程	17
圖 2.3.5 EDK 設計流程.....	20
圖 2.4.1 配置架構[20].....	23
圖 2.5.1 部份可重置示意圖[21].....	24
圖 2.5.2 部份可重置技術分解指紋辨識程序[11].....	26
圖 3.1.1 系統概括圖.....	28
圖 3.1.2 系統流程圖.....	29
圖 3.2.1 MATLAB 模擬 So and Chan 演算法	32
圖 3.2.2 So and Chan 演算法流程圖	34
圖 3.2.3 Tsipouras 演算法簡單示意圖	35
圖 3.2.4 Tsipouras 演算法的流程.....	39
圖 3.2.5 監測模組設計架構	40
圖 3.2.6 計算斜率設計方塊圖	41
圖 3.2.7 心電圖原始訊號與系統計算之斜率	42

圖 3.2.8 斜率與初始閾值計算模組設計圖	42
圖 3.2.9 R 波偵測與閾值更新模組簡要示意圖	43
圖 3.2.10 R 波偵測與閾值更新 ASM 圖	44
圖 3.2.11 R 波偵測與閾值更新模組設計圖	45
圖 3.2.12 心電圖 R-R 區間訊號	45
圖 3.2.13 規則 Category2_Rule 邏輯電路圖	46
圖 3.2.14(a)規則 Category3_Initial 邏輯電路圖；(b) 規則 Category3_Episode 邏輯電路圖	47
圖 3.2.15 規則 Category4_Rule 邏輯電路圖	47
圖 3.2.16 心律不整判斷 ASM 圖	48
圖 3.2.17 心律不整模組設計圖	49
圖 3.3.1 離散小波轉換壓縮架構[27]	50
圖 3.3.2 一階離散小波轉換示意圖	51
圖 3.3.3 圖片經過一階二維小波轉換的處理結果	52
圖 3.3.4 (a)心電圖原始訊號；(b)心電圖經小波轉換後的低頻訊號	52
圖 3.3.5 5/3 濾波器上提式小波轉換架構	53
圖 3.3.6 5/3 濾波器上提式整數對整數小波轉換架構	54
圖 3.3.7 Run-Length 編碼實例	55
圖 3.3.8 心電圖訊號的小波轉換係數	56
圖 3.3.9 壓縮模組設計架構	57
圖 3.3.10 資料擷取與位移模組設計圖	58
圖 3.3.11(a)原始心電圖訊號；(b)平移後的心電圖訊號	58
圖 3.3.12 一階 5/3 整數對整數小波轉換硬體架構	59
圖 3.3.13 一階 5/3 整數對整數小波轉換設計圖	60
圖 3.3.14 一階 5/3 小波轉換模組接腳圖	60

圖 3.3.15 四階 5/3 整數對整數小波轉換設計圖	61
圖 3.3.16 四階 5/3 整數對整數小波轉換係數	62
圖 3.3.17(a)原始的小波係數；(b)經過閾值篩選的小波係數	63
圖 3.3.18 閾值篩選模組設計圖	63
圖 3.3.19 Run-Length 編碼 ASM 圖	65
圖 3.3.20 Run-Length 編碼模組設計圖	65
圖 3.4.1 MicroBlaze 處理器系統架構	66
圖 4.1.1 MIT-BIH 資料庫心電圖訊號	69
圖 4.1.2 MIT-BIH 心電圖訊號心跳類型註解	70
圖 4.2.1 MIT-BIH R 波偵測(a)檔案 100；(b)檔案 102；(c)檔案 109；(d)檔案 115； (e)檔案 123；(f)檔案 205	72
圖 4.2.2 MIT-BIH R 波偵測(a)檔案 107；(b)檔案 111；(c)檔案 112；(d)檔案 113； (e)檔案 217；(f)檔案 228	73
圖 4.2.3 MIT-BIH 心電圖壓縮相關訊號(a)檔案 100；(b)檔案 205	80
圖 4.2.4 MIT-BIH 心電圖壓縮相關訊號(a)檔案 215；(b)檔案 222	81

表目錄

表 2.2.1 Virtex-6 系列資源列表.....	13
表 3.2.1 Tsiouras 演算法與 MIT-BIH 資料庫心律不整種類之對應[26].....	36
表 3.2.2 Tsiouras 演算法判斷條件列表.....	37
表 3.2.3 心律不整規則列表.....	37
表 4.2.1 MIT-BIH 心電圖檔案 R 波偵測實驗結果.....	75
表 4.2.2 實驗結果與 MIT-BIH 檢定狀況對照定義.....	77
表 4.2.3 心律不整判別效能評估.....	77
表 4.2.4 心律不整判別效能評估.....	78
表 4.2.5 壓縮效能評估.....	82
表 4.2.6 硬體架構列表.....	83
表 4.2.7 監測模組與壓縮模組硬體使用資源.....	84
表 4.2.8 系統效能量測結果.....	86

第一章 緒論

1.1 心電圖簡介

1.1.1 心電圖原理與功能

心電圖(Electrocardiography, ECG) 如圖 1.1.1 所示，是一種經胸腔以時間為單位記錄心臟的電生理活動，並通過皮膚上的電極捕捉並記錄下來的診療技術，提供了了解心肌電生理活動的重要依據。

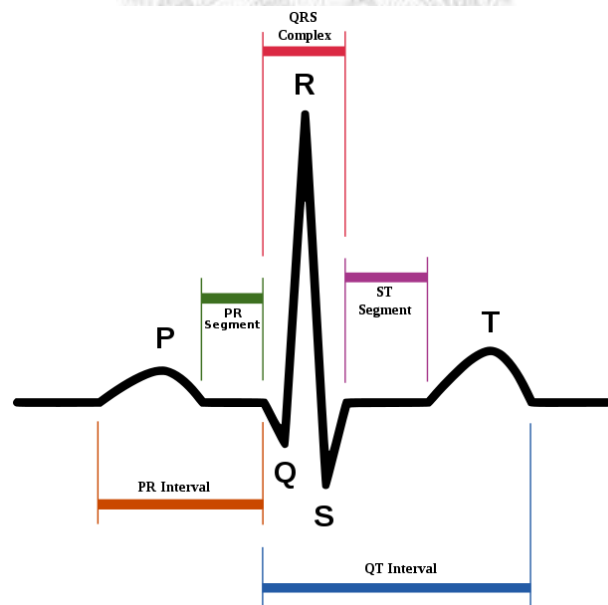


圖 1.1.1 正常心電圖圖解

ECG 的工作原理簡單的來說是這樣的：在每次心跳心肌細胞去極化的時候，會在皮膚表面引起很小的電學改變，這個小變化被心電圖記錄裝置捕捉並放大即可描繪心電圖。在心肌細胞處於靜息狀態時，心肌細胞膜兩側存在由正負離子濃度差形成的電勢差，去極化即是心肌細胞電勢差迅速向零變化，並引起心肌細胞收縮的過程。在健康心臟的一個心動周期中，由竇房結細胞產生的去極化波有序的依次在心臟中傳播，先傳播到整個心房，經過「內在傳到通路」傳播至心室。如果在心臟的任意兩面放置 2 個電極，那麼在這個過程中就可以記錄到兩個電極間微小的電壓變化，並可以在心電圖紙或者監視器上顯示出來。心電圖可以反應整個心臟跳動的節律，以及心肌薄弱的部分。

ECG 是測量和診斷異常心臟節律的最好的方法，其是診斷心電傳導組織受損時心臟的節律異常以及由於電解質平衡失調引起的心臟節律的改變。

1.1.2 心律不整

正常人心跳之律動，是由右心房的竇房結來負責發電而產生一個節律，經由房室結、希氏束的傳導，把電刺激經由心房傳到心室，最後引發心臟肌肉收縮，以維持正常的血壓，足以供給身體所需之血液，相關位置可參考圖 1.1.2。

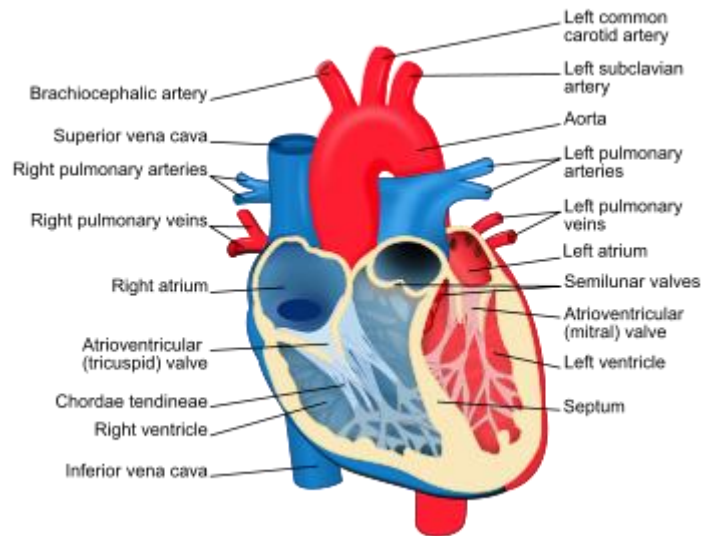


圖 1.1.2 心臟圖

當這個心臟電氣傳導系統出現問題，可能有各種不正常、或不規則的心跳出現，進而引起心悸、心律不整等問題。而正常人的心臟的跳動是有規律的，是以一定的間隔並以每分鐘約六十到八十次的頻率在跳動，心律不整(Arrhythmia)的症狀，病人可以是全無症狀，或是感覺心悸、心跳加快，或是感覺不規則的心跳。嚴重之心律不整時，可引發病人休克、暈厥昏倒、甚至猝死。由心電圖檢查我們可以診斷心律的不整。

1.2 研究動機

心電圖是診斷心律不整最重要的生理訊號，所以目前相關研究已經提出了很多心電圖的系統架構，也被實現在不同的平台上；然而為了進一步提高系統的效能，最近很多研究開始選擇FPGA(Field-Programmable Gate Array)作為平台[1]-[9]，原因是由於FPGA可以完成複雜的運算且具備高速的處理能力，特別是有即時處理的需求時，這樣的特性是非常重要的；另外FPGA的設計架構也適合被轉換成ASIC(Application-Specific Integrated Circuit)，在量產與微小化上提供進一步的優勢。

目前以FPGA為平台的心電圖系統大致可以分為兩類，分別是診斷型和記錄型；診斷型系統[1]-[4]是本身具有判斷心律不整的功能，但由於目前檢測心律不整的演算法其準確性與可靠度尚無法達到醫療規格的要求，所以目前仍以輔助診斷的角色為主，此類系統的設計概念是，當系統偵測到疑似異常的心電圖訊號時，就會發出警示，並將心電圖資料傳送給醫護人員做進一步的診斷，雖然系統只有在偵測到疑似異常的訊號時才会有傳送資料的需求，但是傳送資料仍不免耗費功率、且必須考慮傳送當下環境的頻寬條件，也就是說，如果系統需要傳送的資料量可以越小，它的可靠度就越高。

記錄型系統[5]-[9]則是直接儲存或是傳送原始的心電圖訊號，但是在持續監控的情況下資料量就顯得很大，在儲存空間上將造成隱憂，在傳送上也有可靠度與功率消耗的問題，所以此類系統通常會先將資料做壓縮的處理，再進行儲存或傳送；不過記錄型系統在即時監控的應用上，雖然能夠有效的監控病人的狀況，但

是也需要醫護人員去持續的觀察心電圖訊號，這將造成醫護人員的負擔與增加醫療成本。

由以上敘述可知，兩種類型的系統架構各有其缺點，但卻也同時存在互補關係，基於此理由驅使本研究提出一個更完善的架構，簡單的說，我們將兩類型系統的功能模組建構在同一個系統上，並加以降低硬體使用資源與成本。



1.3 研究目的

本研究同樣以 FPGA 作為平台，去整合診斷型與記錄型兩種心律不整系統的模組功能，並建立一個有效率的使用機制，完整的利用兩個不同功能模組的效能。在使用機制上，本系統能夠主動偵測心律不整，當檢測到異常的心電圖訊號時，以往的診斷型系統會開始把原始的心電圖訊號開始傳送回醫護端，但我們建立的系統則會把接下來所接收到的心電圖資料，先經過壓縮處理後再傳送出去，這將提高傳送可靠度及降低系統的功耗；另外，由於我們持續利用系統作初步的心律不整的診斷，使得醫護端不必從頭至尾監督患者的狀況，這降低了醫護人員的負擔與醫療成本，也就是說修正了原本記錄型系統的缺點。

但是當我們在同一個系統，同時建立兩個不一樣的功能模組時，以硬體設計的考慮必須思考的是，在功能擴增的同時也會增加使用資源、成本、以及功率的消耗。所以為了解決這些問題，我們進一步的使用 FPGA 近年來被廣泛討論與應用的技術一部份可重置(Partial Reconfiguration)技術[10][11]，它能夠有效的降低因為功能性擴增所帶來的硬體資源與成本問題；簡單的說，動態部份可重置技術適用於當一個系統擁有多個功能模組，而這些模組又有分時工作的相互關係時，動態部份可重組技術可以有效率的進行使用資源的分配，同時降低硬體成本與功率消耗等問題；由於我們的系統運行機制，恰符合部份可重置技術要求的使用時機，所以我們將這樣的技術建立在我們的系統上，進一步增加系統的使用效率。

1.4 論文架構

本論文共分成五章。首章為緒論，介紹本研究的研究動機與目的；第二章是背景知識，介紹本研究系統設計所使用的工具與其流程；第三章為系統設計，介紹本研究系統設計所使用的演算法及實作方法與過程；第四章為實驗結果與討論，呈現本研究功能驗證與效能評估的實驗結果，並對其加以討論；第五章是結論，對本研究論文的內容與作概述與總結。



第二章 背景知識

2.1 FPGA 基本概念

2.1.1 FPGA 簡介

FPGA 即是現場可程式化閘陣列(Field-Programmable Gate Arrays)，為可重複程式設計的晶片。FPGA 具備硬體時脈的速度與穩定性，且不需大量晶片即可進行作業，進而降低客制化 ASIC 設計的費用。FPGA 本身為平行架構，因此不同的處理作業並不會佔用相同資源，各個獨立處理作業均將指派至專屬的晶片區塊，不會影響其他邏輯區塊即可自動運作。因此，當要新增其他處理作業時，亦不致影響其他部分的效能。

FPGA 內部包含可配置邏輯區塊(Configurable Logic Block, CLB)、輸入輸出模塊(Input Output Block, IOB)、和內部連結(Interconnection)三部份，如圖 2.1.1 所示。

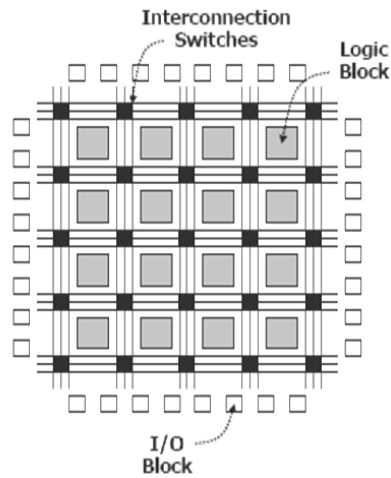
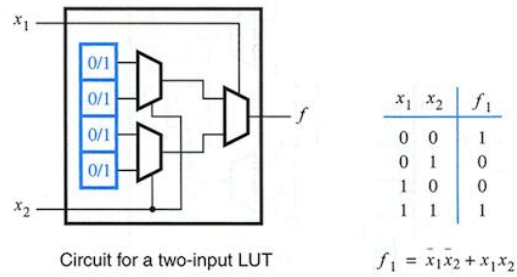


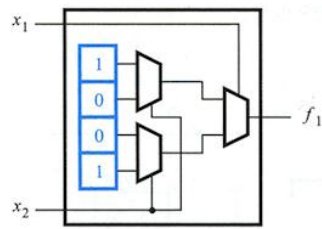
圖 2.1.1 FPGA 內部結構

CLB 是 FPGA 實現大部分邏輯的地方，其中主要包含了查找表(Look Up Table, LUT)；IOB 是晶片與外部信號溝通的界面；內部連結顧名思義，就是負責邏輯間的連線，一開始邏輯區塊間預先就會有通路存在，之後再依據設計的需要決定是否連結。

FPGA 的工作原理是基於 LUT，查找表其實就是一個 RAM。例如很多 FPGA 是使用 4 輸入的查找表，所以每一個 LUT 可以看成一個有 4 位元地址線、大小為 16 乘 1 的 RAM。當用戶通過 HDL 語言等設計方式描述一個邏輯電路以後，FPGA 的開發軟體會計算邏輯電路的所有可能的結果，並事先把計算好的結果存放在 RAM 中。之後每當有訊號輸入需要進行邏輯運算時，不必再用邏輯閘去搭建電路，只要把輸入作為一個地址進行查表，找出相對應地址所儲存的內容，然後輸出即可。如圖 2.1.2 所示。



Circuit for a two-input LUT

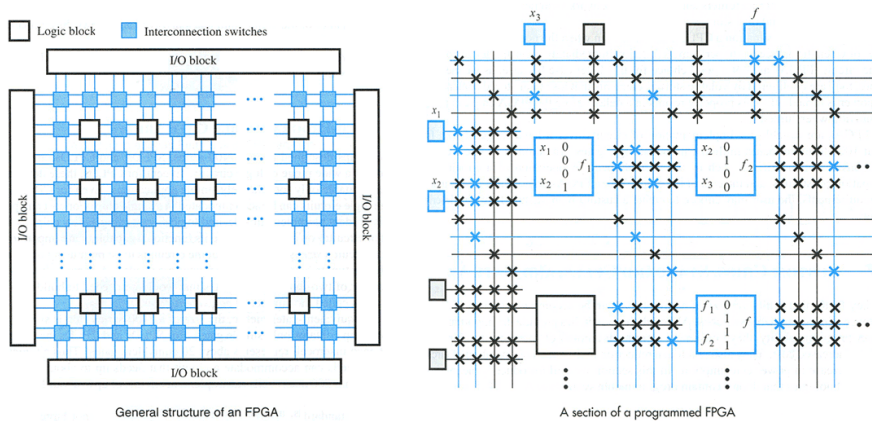


Storage cell contents in the LUT

A two-input lookup table (LUT)

圖 2.1.2 LUT 示意圖

總結上述 FPGA 內部結構及工作原理，FPGA 設計結構如圖 2.1.3 所示，右邊顯示的是 FPGA 內部的結構，左邊是 FPGA 設計的簡易示意圖，邏輯區塊由查找表構成。



General structure of an FPGA

A section of a programmed FPGA

圖 2.1.3 FPGA 設計示意圖

2.1.2 SOPC 系統

SOPC 為可程式化系統晶片(System on Programmable Chip)的縮寫，顧名思義它是將一顆可規劃系統的晶片(即 FPGA)規劃成一個系統單晶片。Xilinx 的 Virtex 系列中就植入了 IBM PowerPC405 硬核，這樣就能使得 FPGA 的硬體設計可以與處理器的強大軟體功能相結合，高效地實現 SOPC 系統。但由於硬核是預先植入的，設計者無法裁減處理器之硬體資源以降低 FPGA 成本，也無法根據實際設計需求在同一 FPGA 中使用多個處理器核，所以 Xilinx 的 Spartan 2 及 Virtex 系列以上的 FPGA 皆有嵌入軟核 MicroBlaze，軟核能夠填補硬核的不足之處，使得設計更為彈性。由於半導體製程的進步，FPGA 性能已漸漸趕上 ASIC，且價格與成本亦日趨下降，因此直接以 FPGA 晶片量產新產品已越來越普遍，居於這種趨勢 SOPC 已經是設計者最佳的選擇之一。

2.1.3 MicroBlaze 處理器

MicroBlaze 是 Xilinx 公司 FPGA 的微處理器[12]，運行時脈為 150MHz，可提供 125 D-MIPS 的性能，能與其它外部 IP 核一起完成 SOPC 的設計。MicroBlaze 處理器採用 RISC 架構和哈佛結構的 32 位元指令和資料匯流排，可以全速執行儲存在晶片上儲存器和外部儲存器中的程序，並存取其數據。

MicroBlaze 是所謂的軟核，也就是當有使用需求時，才需要被合成並出現在系統上；一個系統在資源許可的情況下，也可以同時加入數個 MicroBlaze。

MicroBlaze 能夠彈性地被修改部分的硬體組織，如快速緩衝儲存體(cache)的大小、外部 I/O 週邊設備的支援等，如此使用者能根據需求的不同而修改部分的處理器功能，因而快速的建構出客製化及良好效能的系統。MicroBlaze 的內部結構如圖 2.1.4 所示。

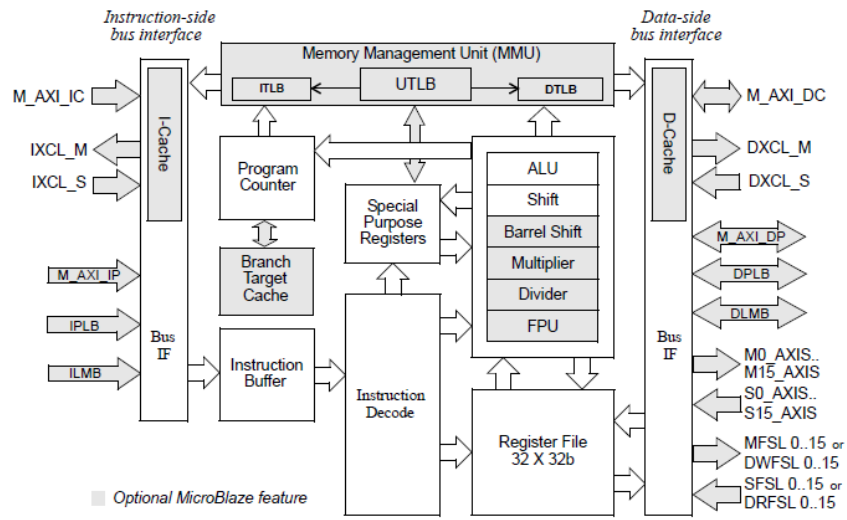


圖 2.1.4 MicroBlaze 內部架構[12]

2.2 Xilinx Virtex-6 ML605

本研究所使用的系統開發平台，是 Xilinx 公司所推出的 Virtex-6 ML605 開發板[31]，板上了建立許多週邊裝置與元件供開發人員選擇使用，例如我們系統所選用的 UART、System ACE 控制器等。表 2.2.1 為 Virtex-6 系列的資源列表，Virtex-6 FPGA 的 Slice 包括 4 個 LUT 和 8 個正反器，每個 LUT 有 6 個輸入；每個 DSP 48E1 Slice 包括一個 25 乘 18 的乘法器、一個加法器與一個累加器；BRAM(Block RAM) 大小基本為 36Kb，每個模塊還可以作成兩個獨立的 18Kb 模塊。紅框標選的地方是我們開發板使用的 Virtex-6 晶片的資源。

表 2.2.1 Virtex-6 系列資源列表

Virtex-6 Family Product Table									
Part Number	LX75T	LX130T	LX195T	LX240T	LX365T	LX550T	LX760	SX315T	SX475T
Logic Cells	74.5K	128K	200K	241K	364K	530K	759K	315K	476K
Maximum Distributed RAM (Kbits)	1,045	1,740	3,040	3,650	4,190	6,200	8,280	5,090	7,640
Block RAM/FIFO (36Kbits each)	156	264	344	416	416	632	720	704	1,064
Total Block RAM (Kbits)	5,616	9,504	12,384	14,976	14,976	22,752	25,920	25,344	38,304
Mixed Mode Clock Manager (MMCM)	6	10	10	12	12	18	18	12	18
DSP48E1 Slices	288	480	640	768	576	864	864	1,344	2,016
PCI Express® Interface Blocks	1	2	2	2	2	2	0	2	2
10/100/1000 Ethernet MAC Blocks	4	4	4	4	4	4	0	4	4
GTX Low-Power Transceivers	12	20	20	24	24	36	0	24	36
Package	Size (Pitch)	Maximum User I/O: Select I/O* Interface Pins (GTX Transceivers)							
FF484	29 x 29 mm (1.0 mm)	240 (8)	240 (8)						
FF784	29 x 29 mm (1.0 mm)	360 (12)	400 (12)	400 (12)	400 (12)				
FF1156	35 x 35 mm (1.0 mm)		600 (20)	600 (20)	600 (20)	600 (20)			
FF1759	42.5 x 42.5mm (1.0 mm)			720 (24)	720 (24)	720 (24)	840 (36)	720 (24)	840 (36)
FF1760	42.5 x 42.5mm (1.0 mm)						1,200 (0)	1,200 (0)	

2.3 設計軟體與流程

2.3.1 System Generator

傳統的 FPGA 開發流程需要編寫大量的程式碼(Verilog 或 VHDL)，然後將程式碼轉換成具體的硬體來實現功能，工作量大。但 FPGA 晶片的資源是有限的，要想設計出高速、可靠、且佔用較少資源的系統，就必須要求開發人員對具體的 FPGA 晶片有一定的了解。雖然可以通過 Core Generator 使用現成的 IP 來提高性能並且降低開發難度，但在做系統設計時並不一定能獲得滿足功能要求的 IP，同時系統工程師難以面面俱到，這部份的工作通常需要專業的硬體工程師來完成。System Generator[13]-[15]的出現旨在打破這樣的局面，在 MATLAB/Simulink 上搭建能否 Xilinx FPGA 設計軟體 ISE 溝通的平台(如圖 2.3.1)，讓對硬體不甚了解的系統工程師也可以進行 FPGA 的設計開發。

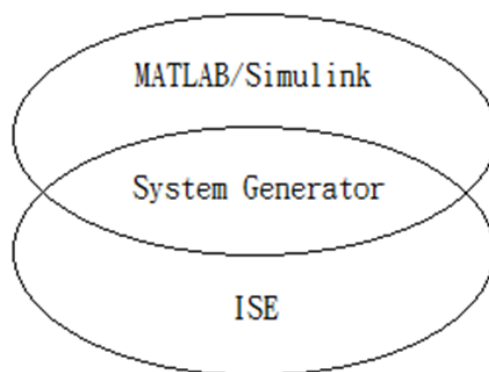


圖 2.3.1 System Generator 的作用

System Generator 有以下的優點及特性：

- 模組的參數化和設計的圖形化使得修改設計變得很方便，明確將控制流與數據流分開，使得設計結構清晰。
- 設計方法多樣性；除了可以使用 System Generator 內建的功能模組設計系統外，System Generator 也可以導入硬體描述語言、MATLAB 等編寫的程式代碼，包攬成模組共同作為設計的一部份。

提供方便的模擬/驗證平台；由於 System Generator 是建構在 Simulink 平台上，所以很容易的能夠透過 MATLAB/Simulink 導入驗證資料，並觀察輸出結果以驗證系統功能性；System Generator 亦支援 ChipScope 和硬體協同模擬，可以實時的將 FPGA 內的信號變化回覆給開發人員。

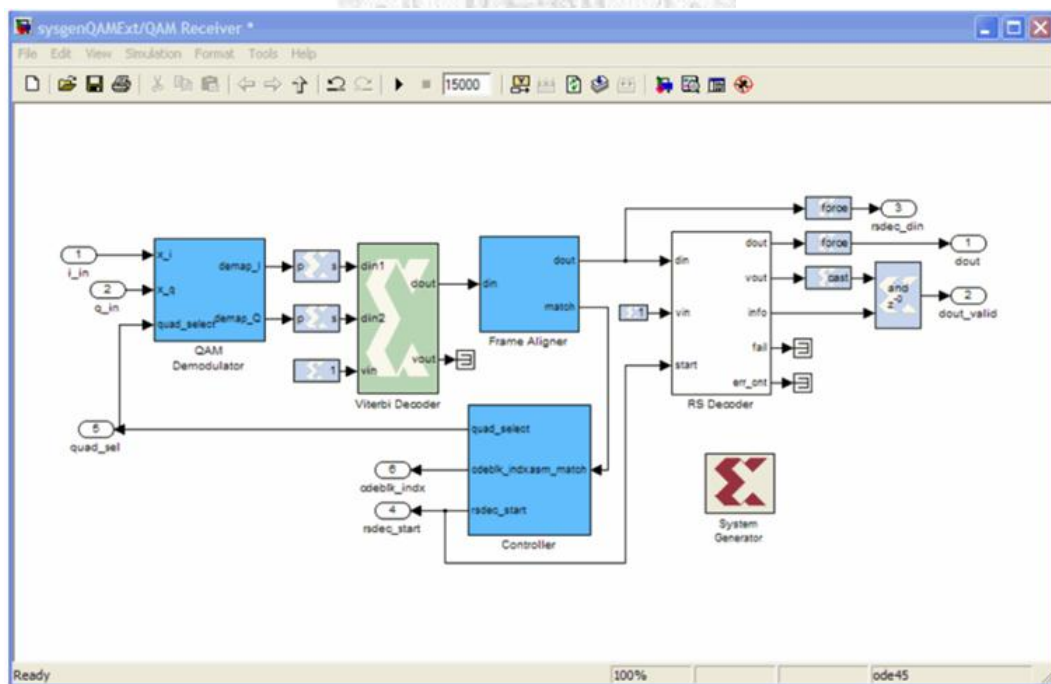


圖 2.3.2 System Generator 設計實例示意圖[13]

使用 System Generator 進行 FPGA 的設計流程如圖 2.3.3。首先使用 System Generator 進行系統設計，產生副檔名為 mdl 的 Simulink 檔案；再來利用 System Generator 自動產生在 ISE 進一步處理的 RTL(Register Transfer Level)和 IP 程式碼，同時產生在模擬時需要的 testbench 測試檔案；最後在模擬程式(ISE XST 或 ModelSim)驗證系統是否滿足設計需求後，就算是完成整個設計流程了；由上述流程可以得知，使用 System Generator 去開發系統並沒有增加設計程序，幾乎和傳統的設計流程相同。

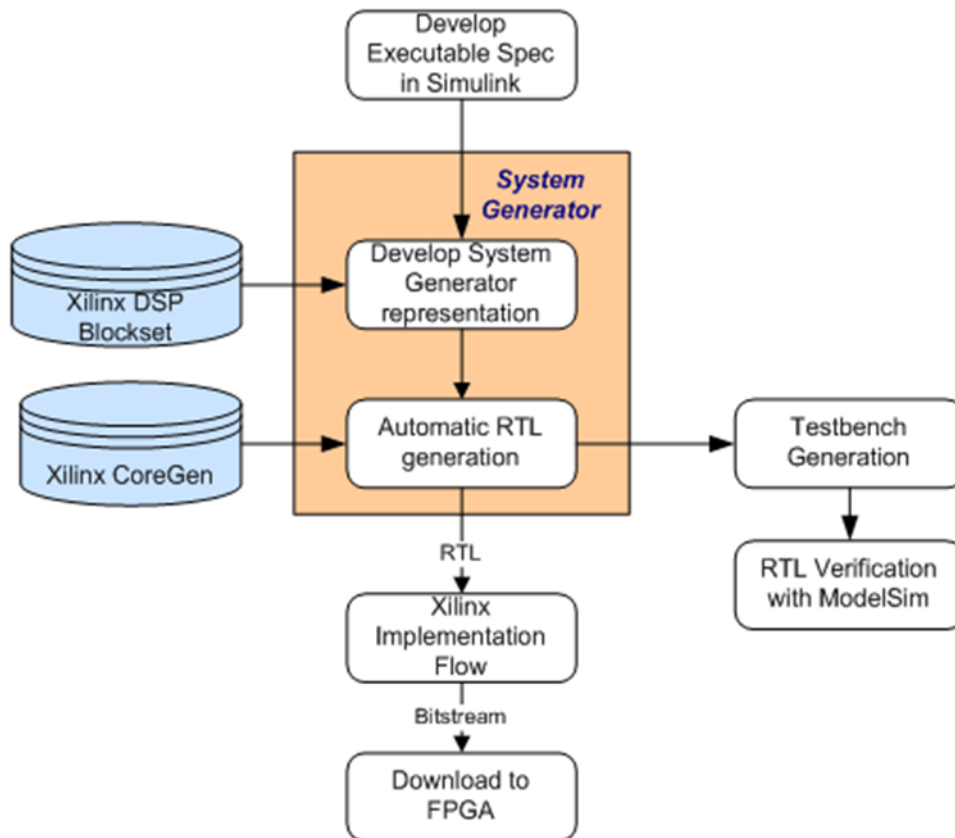


圖 2.3.3 System Generator 的 FPGA 開發流程[13]

2.3.2 Xilinx ISE

ISE 是使用 Xilinx 公司的 FPGA 作為設計平台之開發人員必需熟練的一套軟體 [16]。ISE 主要的功能包括設計輸入、合成(Synthesize)、模擬(Simulation)、實現(Implement)和下載，涵蓋了 FPGA 一般開發的所有流程(圖 2.3.4)。從功能上來看，其工作流程無需借助任何第三方軟體的支援。

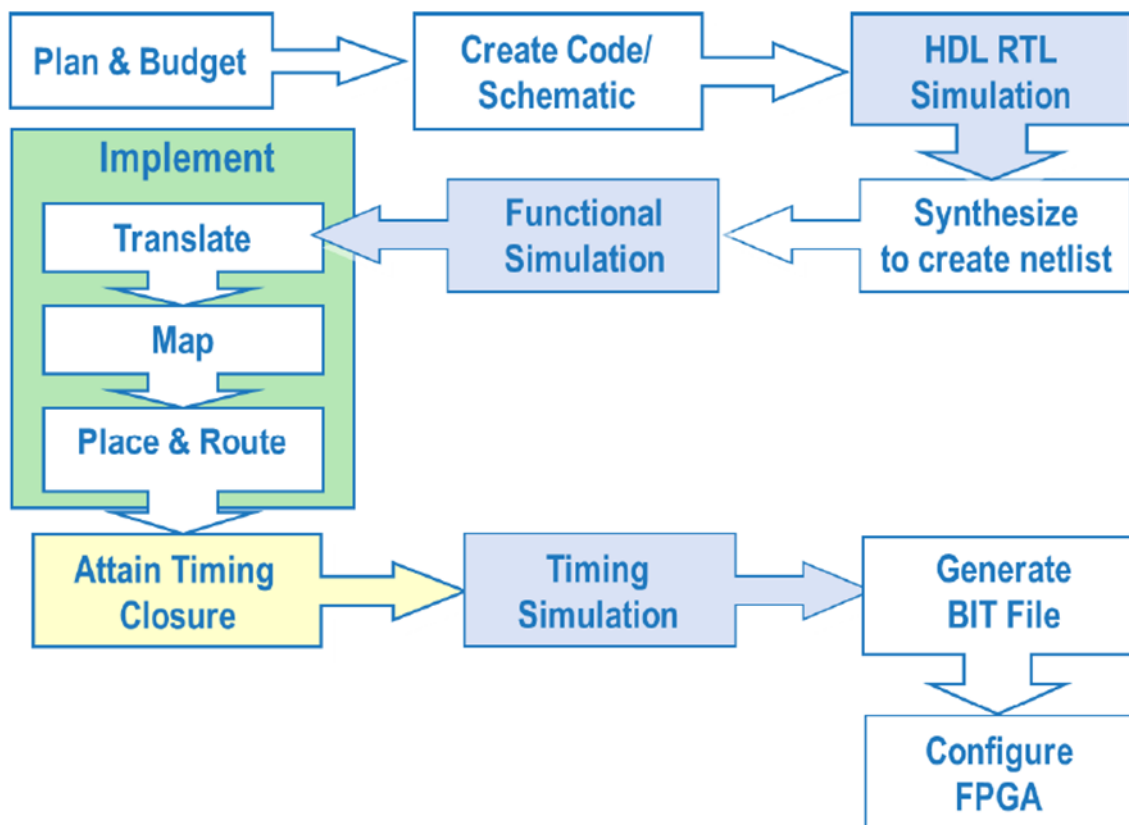


圖 2.3.4 軟體 ISE 開發流程

以下針對幾個比較重要的步驟簡述：

- **模擬**：一般而言，在電路設計的模擬上可分為 Pre-Simulation 跟 Post-Simulation。Pre-Simulation 是針對電路的 function 做模擬，此時只在意由 HDL 所撰寫的程式之功能是否正確，而 Post-Simulation 則是針對合成過且做完 APR(Auto Place and Route)的電路做模擬，以確保所設計的電路實現在 FPGA 上時，與 Pre-Simulation 的功能一樣。ISE 本身內建一個具有圖形化波形編輯功能的模擬工具 HDL Bencher，同時提供使用 Mentor Graphic 公司的 ModelSim。
- **合成**：合成工具的功用是將 HDL 轉換成由電路所組成的 Netlist，Netlist 為一種描述邏輯電路的檔案格式。ISE 的合成工具不但包含 Xilinx 自己提供的 XST，還可以內嵌 Mentor Graphic 公司的 LeonardoSpectrum 和 Synplicity 公司的 Synplify。
- **實現**：此功能包括轉譯(Translate)、映射(Map)、布局與繞線(Place and Route)等。還具備時序分析、接腳指定等高級功能。其中轉譯是將 Netlist 轉換成 FPGA 內部的 Netlist；映射會應用各種邏輯電路最佳化的方法，目的是用來增加電路的速度與減少邏輯閘的數目；布局與繞線則將 Netlist 裡的邏輯閘分配到 FPGA 內部特定的位置，且所有邏輯閘之間的繞線也會做最佳化的處理。

2.3.3 EDK

近年來所生產的 FPGA 晶片多數都有內嵌微處理器，加上設計軟體的配合與輔助，目前要使用 FPGA 開發一個 SOPC 系統變得相對容易。要利用 FPGA 設計一套嵌入式系統，必須完成兩個流程：

1. 選用硬體元件；包括微處理器(Power PC、MicroBlaze)、記憶體、輸出與輸入週邊等，並且依照它們之間的關係連結成一個完整的硬體架構。
2. 以 C 語言外加一些 I/O 週邊控制指令編寫控制程式，並利用翻譯器將程式碼翻譯成所使用處理器相對應的機器語言，再載入至適當的記憶體中執行。

Xilinx 公司為了幫助開發人員完成上述流程，提供了嵌入式開發軟體 EDK(Embedded Development Kit)[17]，它以友善的使用者圖形界面，導引開發人員簡單地整合系統設計、偵錯、模擬等繁瑣流程。為了完成這些工作，在硬體方面，EDK 建立了不同的處理器、記憶體、週邊等硬體元件，同時提供各種控制匯流排將這些元件連結在一起；在軟體方面，EDK 提供驅動程式、程式庫(Library)、翻譯器(Compiler)、偵錯工具、作業系統等應用程式，配合硬體以完成一個完整的嵌入式系統。

如上所述，一個完整的 FPGA 嵌入式系統設計，必須完成硬體與軟體兩個部份的工作，而這些工作都可以在 EDK 提供的 EST(Embedded Software Tool)內完成，其流程如圖 2.3.5 所示。

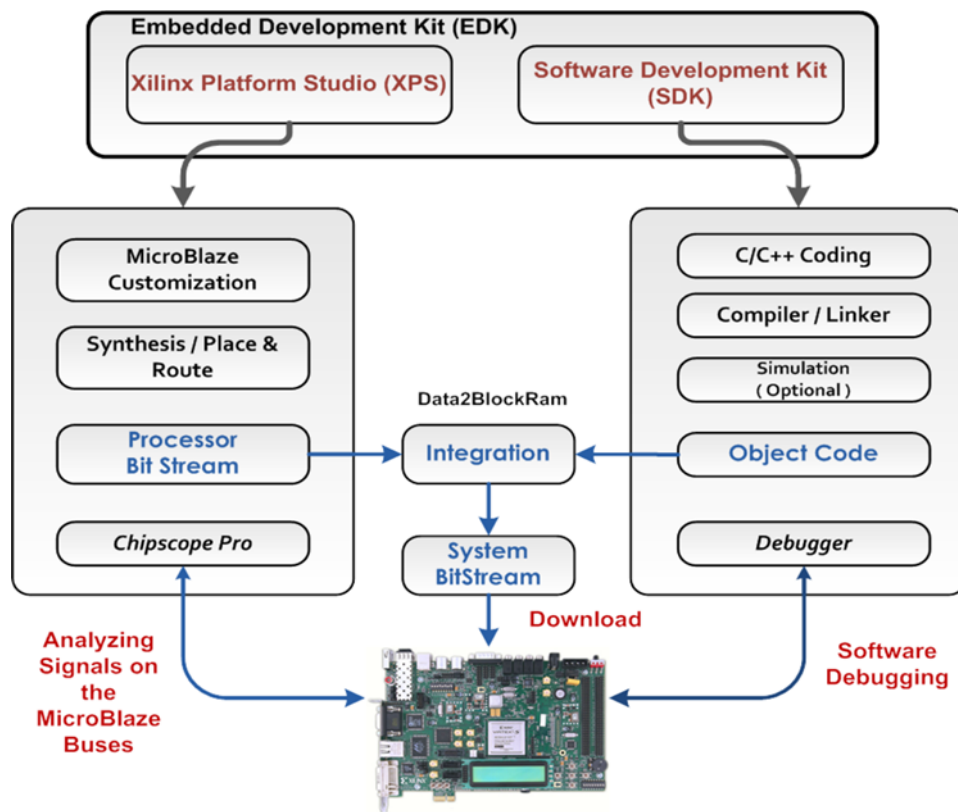


圖 2.3.5 EDK 設計流程

概要的說明，圖 2.3.5 左方為 EDK 處理硬體的流程，右方為處理控制軟體的流程，中間為最後將硬體與軟體整合起來，把設計載入至 FPGA 內。上述流程在 EST 內的 XPS(Xilinx Platform Software)平台上，藉由系統建立精靈 BSB(Base System Builder Wizard)的引導，可以去選擇硬體元件以搭建欲開發的硬體架構，再由開發人員完成硬體元件之間的連結關係等工作後，此時系統的完成度已經很高了。再來利用 SDK(Software Development Kit)處理軟體的部份，主要是將開發人員的控制程式進行編譯、組譯、連結工作，以便產生可執行檔。最後將控制程式的執行檔載入至硬體控制電路對應的記憶體內，並產生 FPGA 需要的 bitstream 檔，下載至 FPGA 晶片內就大功告成了。

值得一提的是，若我們想要在微處理器系統上加入自己設計的 IP，由於沒辦法直接掛在匯流排上直接與處理器互聯的緣故，所以我們必須額外提供媒介。這樣的媒介大致有兩種，一個就是將自行設計的 IP 掛在週邊元件後面，透過週邊元件與處理器溝通；另一種方式就是在 IP 前方掛上所謂的 IPIF(IP InterFace)，其功能是用來產生 OPB 與 PLB 匯流排與使用者邏輯電路中間的溝通與控制訊號，簡單的說，這種方法是以更直接的方式與處理器互聯。

2.3.4 PlanAhead

PlanAhead 是 Xilinx 提供的一種圖形化的 FPGA 設計工具[18]，它可以被應用於 FPGA 開發過程中的不同階段，常見的應用包括用 PlanAhead 進行 RTL 源代碼的開發、I/O 接腳規劃、RTL 的 Netlist 分析、布局與繞線結果的分析、布局規劃等。PlanAhead 還可以將 Chipscope 加入設計以輔助測試，進而提高性能。PlanAhead 也可以做各種實現屬性(Implement Property)的不同設置，利用不同的時序約束(Timing Constraints)、實體約束(Physical Constraints)和布局規劃來提高設計性能。另外還可以將 ISE 的布局與繞線結果導入 PlanAhead 進行分析，定位關鍵路徑，找到影響設計性能的真正原因。PlanAhead 已經被整合進 ISE 11.X 套件以後的版本，包括合成前/後的接腳規劃(I/O pin planning Post/Pre-Synthesis)、合成後面積/接腳/邏輯規劃(Floorplan Area/IO/Logic Post-Synthesis)和實現後時序分析/設計規劃(Analyze Timing/Floorplan Design Post-Implementation)。它替代了以前常用的 PACE 和 Floorplanner 工具。

2.4 配置介面

2.4.1 JTAG 與 SelectMAP

JTAG(Joint Test Action Group)是一種專門用於配置(Configuration)或測試電路板的介面，JTAG 介面技術是由 IEEE-1149.1 邊界掃描架構(IEEE-1149.1 Boundary Scan Architecture) 標準規定所規範的一項技術。JTAG 可以讓一個控制器同時連接多個裝置進行測試，另外也可以內置一些預先定義好的功能模式，例如 FPGA 配置動作即是其中一種功能，其採用序列傳輸的方式完成配置的工作。通過邊界掃描通道來使晶片處於某個特定的功能模式，以提高系統控制的靈活性和方便的系統設計。

除了可以使用 JTAG 完成 FPGA 晶片的配置工作以外，Xilinx 亦提供另一項選擇—SelectMAP 技術；其採用平行傳輸概念，一次傳輸 8 位元資料至 FPGA 有別於 JTAG 一次傳遞一個位元的資料，因此其做重組時速度較其它方式快。該介面依時脈傳遞的方向分為 Master SelectMAP 與 Slave SelectMAP 兩種，Master 即是其 CCLK 是由 FPGA 傳向 PROM，而 Slave 則剛好相反。

2.4.2 ICAP

ICAP(Internal Configuration Access Port)[19]是一種簡化的 SelectMAP 介面，不同的是 ICAP 位於 FPGA 的內部。動態部份可重置技術最關鍵的步驟之一，就是

bitstream 檔案(即配置檔案)的下載，通過 ICAP 介面可以讓開發人員輕易完成這個動作。通過 ICAP 介面對 FPGA 做部份可重置的基本流程是，首先把配置檔案存放在儲存裝置上，在重置時處理器(例如 MicroBlaze)從儲存裝置上讀取配置資料，然後通過匯流排將配置資料傳送給封裝 ICAP 模組的 HWICAP，最後通過 HWICAP 的控制模組將配置資料寫入 ICAP 介面，完成部份可重置的功能。利用不同配置模式的配置架構，可參考圖 2.4.1。

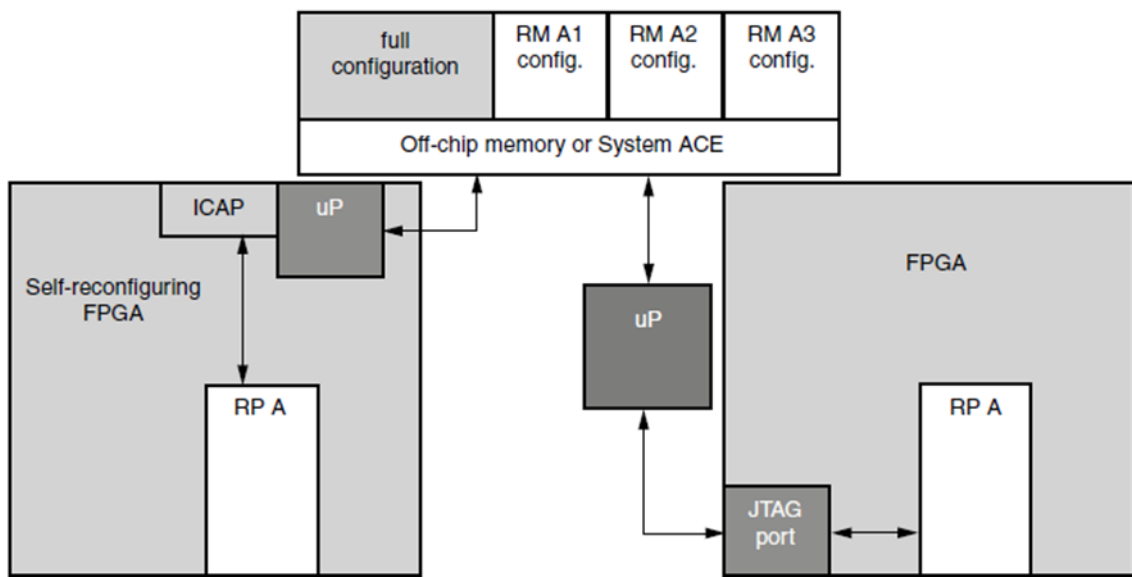


圖 2.4.1 配置架構[20]

2.5 部份可重置技術

2.5.1 基本概念

眾所皆知的 FPGA 最大的特點，就是可以隨時並重覆的修改晶片內的設計，所以一直是 IC 驗證的技術主流。但傳統上重新燒錄這個動作，必須在 FPGA 處於關機(Shutdown)的狀態下才能被執行。近年來 FPGA 突破這樣的限制，讓 FPGA 即使是在運行中也能修改其內部設計，這樣的革新使得 FPGA 的應用更加寬廣、彈性。這個技術被稱為部份可重置(Partial Reconfiguration)[20][21]。

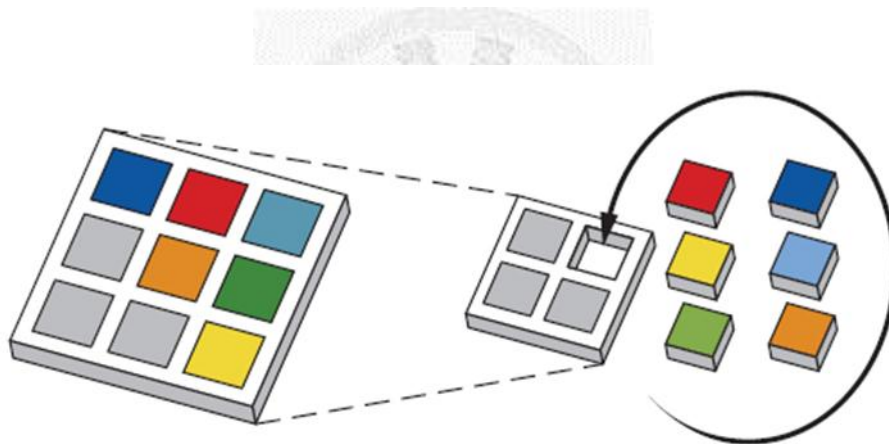


圖 2.5.1 部份可重置示意圖[21]

當一個系統愈加複雜，開發人員必須考慮的問題，就是如何簡化設計及減少使用資源。部份可重置技術提供了一個解決方法，能夠降低使用資源、功率損耗、以及開發成本。舉個例子說明部份可重置技術，見圖 2.5.1[21]，灰色區塊是所謂的靜態邏輯(Static Logic)，系統運行的過程中，靜態邏輯持續被使用到；彩色區塊是可重置邏輯(Reconfigurable Logic)，系統運行的過程中，不同顏色的區塊在不同時間才會被使用到。圖 2.5.1 左方是一般 FPGA 設計的規劃，所有邏輯都預先被安

置在 FPGA 晶片上，當系統運行時，即使沒有被使用到的彩色區塊也會佔用面積與消耗功率；圖 2.5.1 右方是使用部份可重置技術的 FPGA 設計，除了靜態邏輯被預先放置在 FPGA 晶片上以外，其他彩色區塊只有在需要被使用到時，才需要被加入晶片中，若沒有任何彩色區塊被使用到，甚至可以切換成黑盒子模組，以節省功耗；從以上敘述可知，部份可重置技術適用於當系統的模組擁有分時(Time Sharing)工作的特性，可以減小使用面積、節省功耗、以及降低成本等。

下面對部份可重置技術後面會使用到的重要名詞做解釋 -

- 可重置模組(Reconfigurable Module, RM)：擁有分時工作特性的模組，視使用時機可靈活置換。
- 可重置區(Reconfigurable Partition, RP)：規劃給可重置模組置換的區域，一個 FPGA 晶片可規劃多塊可重置區，一個可重置區則可置換多個可重置模組。
- 配置(Configuration)：一個配置就是一個完整的設計架構，其中包含整個靜態邏輯與可重置區內的一個可重置模組，意謂一個設計最多會有所有可重置模組所有組合這麼多的配置。

2.5.2 相關研究

Francisco Fons 和 Mariano Fons 於 2010 年提出了應用 FPGA 部份可重置技術的研究[11][12]，他們建立了一個指紋辨識系統(Automatic Fingerprint Authentication System, AFAS)。他們評估一個好的指紋辨識系統需要符合下列條件：

1. 在短時間內正確的比對與辨識指紋。
2. 系統可以做升級或修改，以延長使用年限。

3. 成本必須盡可能降低，以符合市場競爭力。

由於 FPGA 具備良好的運算能力與可重覆修改的特性，所以符合上述前兩點的要求；另外雖然指紋辨識系統的比對程序繁多，但這些程序擁有分時處理的特性，恰好符合使用 FPGA 部份可重置技術的要求；基於以上理由，作者決定利用 FPGA 作為平台，並運用部份可重置技術降低系統成本。

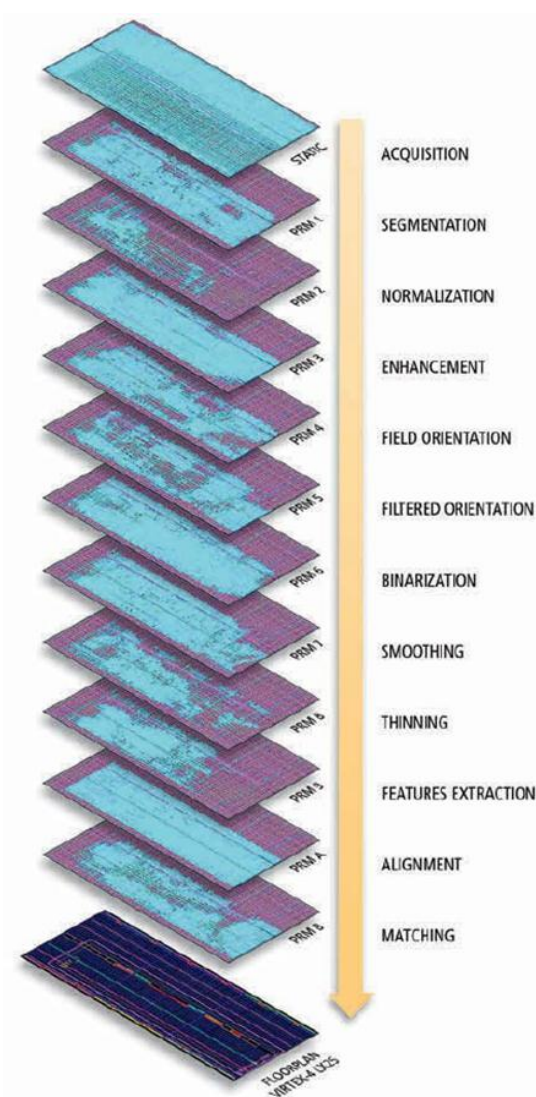


圖 2.5.2 部份可重置技術分解指紋辨識程序[11]

如前所述，指紋辨識程序繁瑣，若依照傳統 FPGA 的設計方式，必然會使用到大量資源而產生成本問題；於是作者將速度要求低的擷取資料部份，交予 FPGA 內嵌的處理器 MicroBlaze 處理，而其餘程序則切割成 11 個模組(如圖 2.5.2)，在唯一一個可重置區內運行，所以當目前的程序處理完畢，則會將輸出提供給下一道程序的輸入，以此類推，直到所有的程序都執行結束為止。根據作者的實驗結果顯示，系統在可接受的執行時間下，節省了可觀的使用資源，當然同時的也降低了系統成本。



第三章 系統設計

3.1 系統概述

圖 3.1.1 為本研究的系統簡易示意圖。如圖所示，我們使用 Xilinx 公司 FPGA 中的 Virtex-6 作為系統平台，訊號來源則是 MIT-BIH 提供的 ECG 紀錄。系統的規劃大致可分成三個部份：

1. 正常模式時，可重置區的預設模組 – 監測模組(Monitoring Module)。
2. 進入緊急模式時，可重置區置換的模組 – 壓縮模組(Compression Module)。
3. 控制可重置區置換模組時機的靜態邏輯；包括微處理器(MicroBlaze)與其週邊。

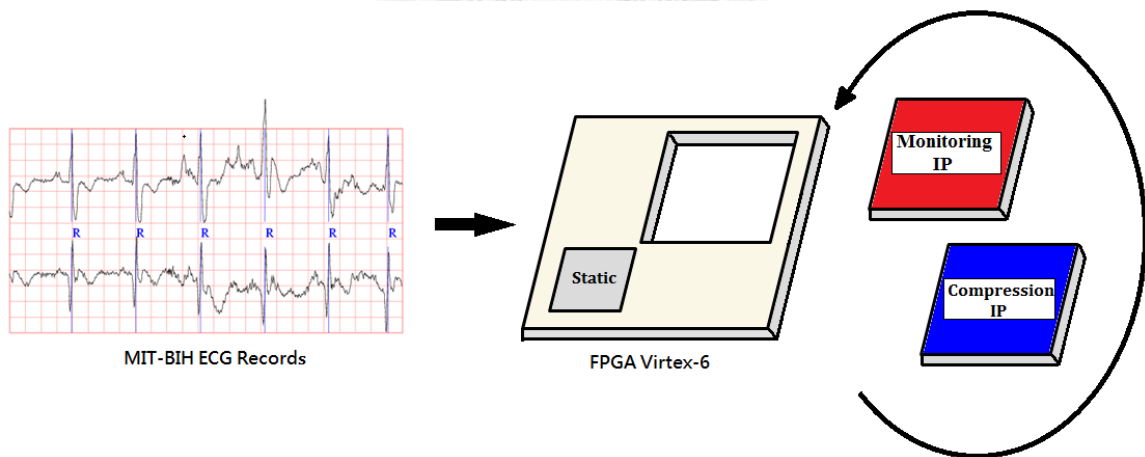


圖 3.1.1 系統概括圖

所以系統規劃了一個可重置區(Reconfigurable Partition)，可以視情境置換不同的模組 – 監測模組(Monitoring Module)與壓縮模組(Compression Module)；而負責置換模組工作的就是規劃成靜態模組的 MicroBlaze 微處理與其必要週邊。

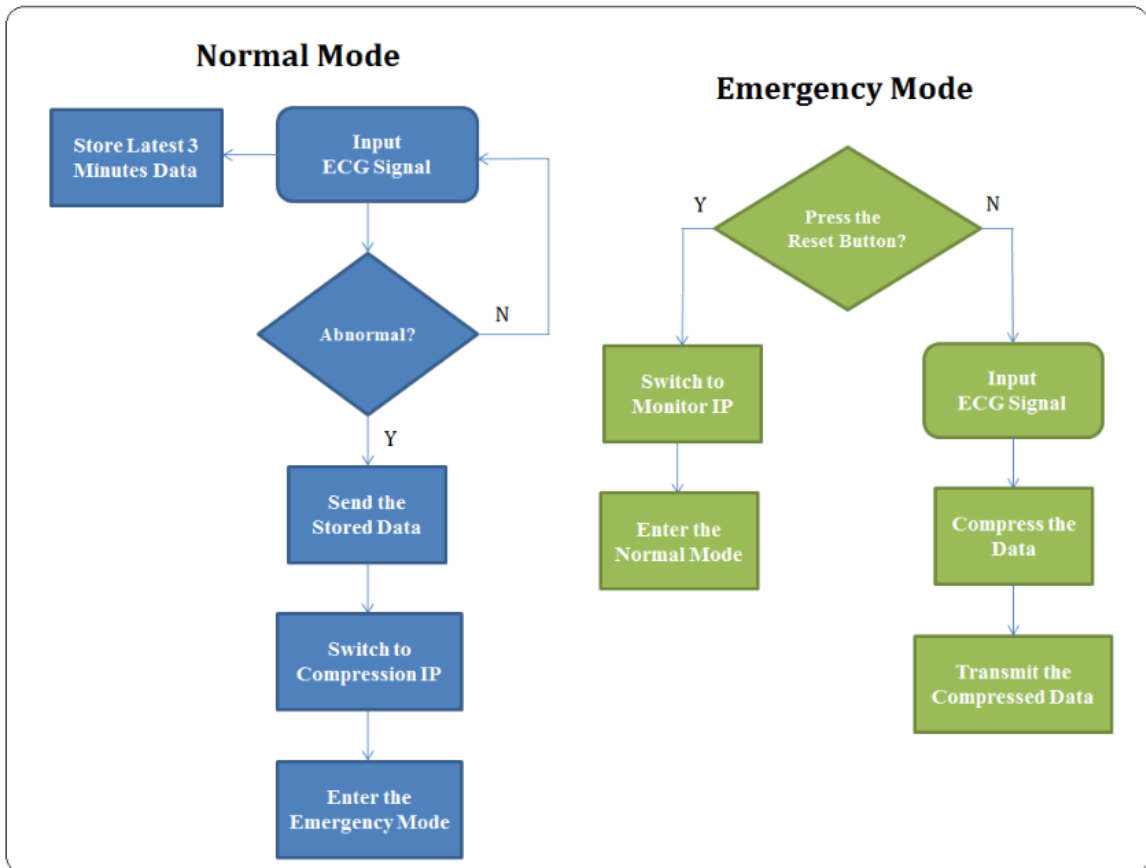


圖 3.1.2 系統流程圖

系統的機制與流程如圖 3.1.2 所示，系統分為正常模式(Normal Mode)與緊急模式(Emergency Mode)。系統的初始狀態為正常模式，此時可重置區預設的工作模組為監測模組，負責持續監控病人是否有疑似心律不整的心電圖訊號出現，同時儲存最近三分鐘的心電圖資料；當檢測到異常的心電圖訊號時，則先傳送出儲存的資料，此舉是為了供醫護人員得到系統判定為心律不整的心電圖資料，以做進

一步的判讀與檢測；傳送完最近三分鐘的資料後，系統則將可重置區上的監測模組置換成壓縮模組，也就是從正常模式進入緊急模式，此後進入系統的心電圖訊號由壓縮模組進行資料壓縮的處理後，直接傳送給醫護人員做持續的監控，由於我們是以小波係數的形式去傳送壓縮資料，要重建出心電圖訊號相當容易，所以醫護端不需要做繁複的後處理；另外，當處於緊急模式時，病人若自身體察狀況並無異常，也可以經由重置鍵返回正常模式，停止持續傳送心電圖資料至醫護端，以減輕系統功率、成本與醫護人員的負荷。

系統所使用的功能模組，其演算法與設計將利用此章接下來的篇幅詳細介紹與說明。



3.2 監測模組

監測模組的功能為檢測心電圖訊號是否疑似為心律不整，其中又可略分成兩個步驟；首先是擷取能判斷心律不整的心電圖特徵，不同的演算法需要的特徵會有所差異，而本系統需要的特徵是心電圖的 R 波；其次就是判斷心律不整。以下子節將就兩步驟的演算法作介紹，最後詳細說明實際設計的過程與結果。

3.2.1 心電圖 R 波偵測

我們偵測心電圖 R 波的方法是使用 So and Chan 演算法，此演算法是由 H. H. So 和 K. L. Chan 於 1997 年首度提出[22]，並於 2000 年[23]進一步與著名的 Pan and Tompkins[24]演算法進行比較，根據實驗數據的顯示 So and Chan 在準確率上有明顯的提升，由於其演算法簡單且具備良好的準確性，所以常被選擇實現在很多的硬體電路上。演算法首先需要計算每一點原始心電圖訊號 $x(n)$ 的斜率 $slope(n)$ ，如公式(3.2.1)所式：

$$slope(n) = -2x(n-2) - x(n-1) + x(n+1) + 2x(n+2) \quad (3.2.1)$$

然後計算斜率的閾值($slope_thresh$)，公式如(3.2.2)所示：

$$slope_thresh = \frac{thresh_param}{16} \times \max i \quad (3.2.2)$$

(3.2.2)式中的 $thresh_param$ 可以設定為 2、4、8、或 16，我們實際測試選擇 8 的效果較好，所以在本系統是選用 8。另外初始的 $max\ i$ 是取訊號前 300 點的斜率最大值，取 300 點是以 MIT-BIH 心電圖訊號的取樣頻率為考量，以確保可以取到初始 R 波的值。

當閾值越小時，表示對於 R 波的偵測有較高的靈敏度，但抗雜訊的能力則較差；反之當閾值越大時，R 波的偵測有較低的靈敏度，但抗雜訊的能力較好。一般來說，雜訊干擾多為瞬間的突波，所以單點的斜率高於閾值時不予理會，但當連續兩點心電圖訊號的斜率 $slope(n)$ 大於閾值 $slope_thresh$ 時，即表示 QRS 複合波的起始點被偵測出來了，於是我們搜尋往後 50 點訊號中斜率的最大值，並判定此位置為 R 波的所在，如圖 3.2.1 所示。

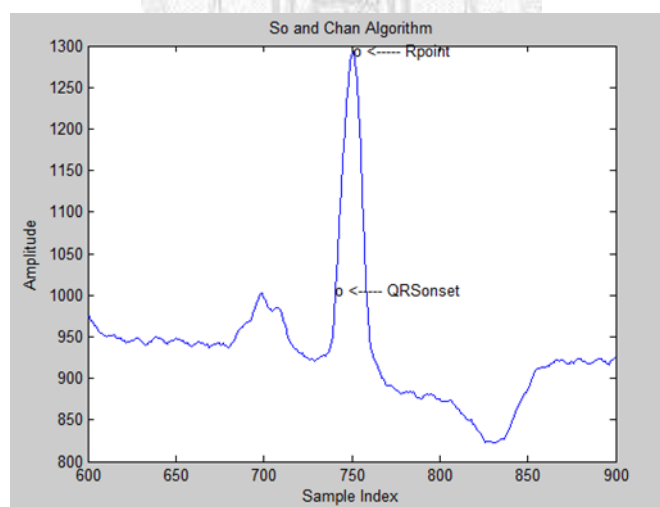


圖 3.2.1 MATLAB 模擬 So and Chan 演算法

每次完成 R 波的偵測程序後，(3.2.2)式的 $max\ i$ 就需要被更新一次，以此保持較佳的閾值，更新的公式如(3.2.3)與(3.2.4)所示：

$$\max i = \frac{\text{first_max} - \max i}{\text{filter_param}} + \max i \quad (3.2.3)$$

$$\text{first_max} = \text{height of R point} - \text{height of QRS onset} \quad (3.2.4)$$

filter_param 可以設定為 2、4、8、或 16，參數 *filter_param* 值越低則表示對相鄰的 R-R 波有較高的偵測靈敏度；反之參數越高則表示有較低的偵測靈敏度。我們實際測試選擇 8 的效果較好，所以在本系統裡是選用 8。

綜合上述，整個演算法的流程如圖 3.2.2 所示。其中取前 300 點的最大斜率步驟，只會執行一次，所以特別以卡片形的流程方塊標記。



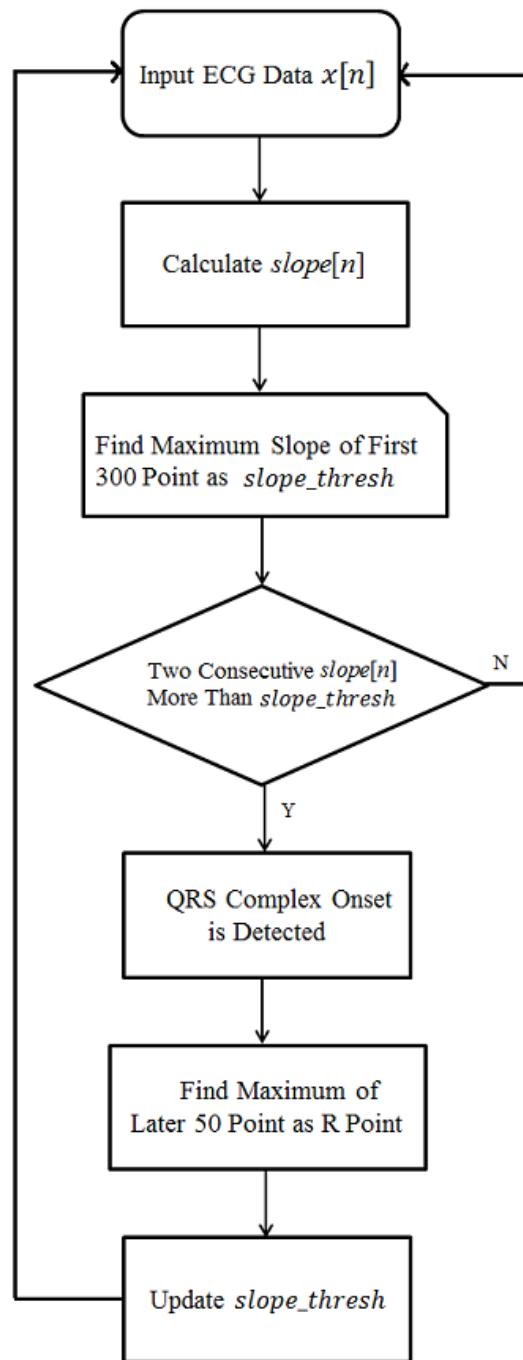


圖 3.2.2 So and Chan 演算法流程圖

3.2.2 心律不整判斷

在上節討論如何偵測出心電圖 R 波的位置後，我們將利用 R-R 區間(R-R Interval)去判斷心電圖訊號是否有心律不整的症狀。Tsipouras 等人於 2002 年與 2005 年皆有提出利用 R-R 區間去判別心律不整的演算法[25][26]，我們選擇經改良過的 2005 年版本的演算法為系統實現對象。此演算法的程序經醫學專家認可、且程序簡易，所以被廣為引用與實現。

圖 3.2.3 為演算法的簡易示意圖。首先有一個移動窗(Sliding Window)執行一次擷取三筆 R-R 區間資料的動作，雖然如此，但每次只對中間的 RR-區間進行心律不整的判定，當判定完成後，移動窗則平移一個資料的區間，擷取下三筆的 R-R 區間進行判斷。

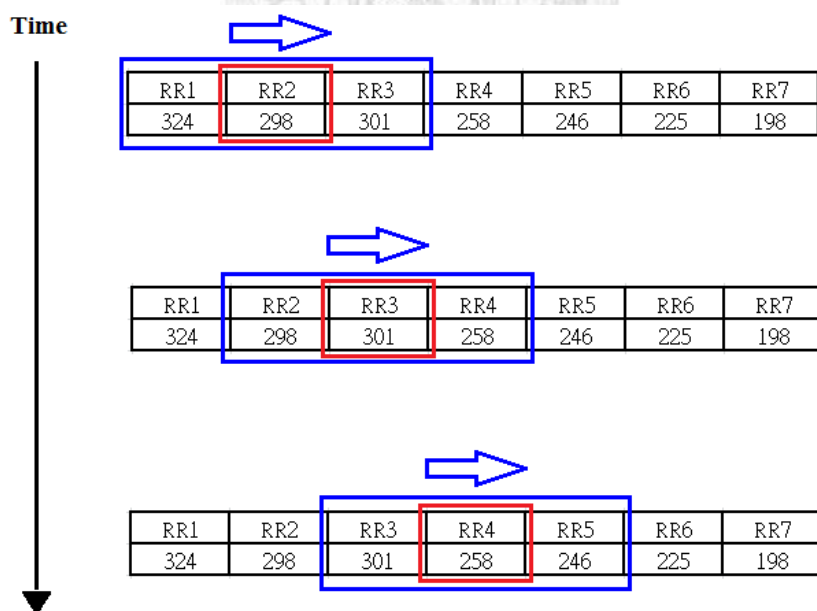


圖 3.2.3 Tsipouras 演算法簡單示意圖

以圖 3.2.3 的 R-R 區間為例，移動窗一開始會先擷取 RR1、RR2、RR3 三筆資料進行條件判斷，但同一時間只會判定 RR2 是否為心律不整；之後移動窗會移動一筆資料的距離，擷取 RR2、RR3、RR4 三筆資料，並對 RR3 進行判定，以此類推。

Tsipouras 演算法可以判斷以下心律的種類：

- ◆ 類型一：正常心跳(Normal)
- ◆ 類型二：心室早期收縮(Premature Ventricular Contraction, PVC)
- ◆ 類型三：心室顫動(Ventricular Flutter/Fibrillation, VF)
- ◆ 類型四：心臟傳導阻礙(2° Heart Block)

表 3.2.1 為 Tsipouras 演算法與 MIT-BIH 心電圖資料庫的心律不整種類對應關係。

表 3.2.1 Tsipouras 演算法與 MIT-BIH 資料庫心律不整種類之對應[26]

MIT-BIH annotation symbol	Type of arrhythmia	Classification
N	Normal beat	Normal (N) (category 1)
P	Paced beat	
f	Fusion of paced and normal beat	
P	Non-conducted P-wave (blocked APB)	
L	Left bundle branch block beat	
R	Right bundle branch block beat	
Q	Unclassifiable beat (beat annotations)	
V	Premature ventricular contraction (beat annotation)	Premature ventricular contraction (PVC) (category 2)
[Start of ventricular flutter/fibrillation	Ventricular flutter/fibrillation (VF) (category 3)
!	Ventricular flutter wave	
]	End of ventricular flutter/fibrillation (beat annotations)	
(BII	2° heart block (rhythm annotation)	2° heart block (BII) (category 4)

表 3.2.2 Tsipouras 演算法判斷條件列表

C1	$(RR2i < 0.6s) \text{ and } (1.8 * RR2i < RR1i)$
C2	$((RR1i < 0.7s) \text{ and } (RR2i < 0.7s) \text{ and } (RR3i < 0.7s)) \text{ or } (RR1i + RR2i + RR3i < 1.7s)$
C3	$((1.15 * RR2i < RR1i) \text{ and } (1.15 * RR2i < RR3i))$
C4	$((RR1i - RR2i < 0.3s) \text{ and } ((RR1i < 0.8s) \text{ or } ((RR1i < 0.8s)) \text{ and } (RR3i > 1.2 * \text{mean}(RR1i, RR2i))))$
C5	$((RR2i - RR3i < 0.3s) \text{ and } ((RR2i < 0.8s) \text{ or } ((RR3i < 0.8s)) \text{ and } (RR1i > 1.2 * \text{mean}(RR2i, RR3i))))$
C6	$(2.2s < RR2i < 3.0s) \text{ and } ((RR1i - RR2i < 0.2s) \text{ or } (RR2i - RR3i < 0.2s))$

接著介紹條件判斷的細節，表 3.2.2 為所有條件的列表。從上表條件中，很難看出明顯且具體的意義，只有將條件作子集組合才會顯現出來，所以為了理解與後續設計上的便利，我們進一步將上述條件分類成幾個有意義的規則，如表 3.2.3 所示。

表 3.2.3 心律不整規則列表

Category2_Rule	C3 or C4 or C5
Category3_Initial	C1
Category3_Episode	C2
Category4_Rule	C6

每一個規則成立所代表的意義就是，若 Category2_Rule 成立(即條件 C3、C4、C5 任一個成立)，表示 RR2i 被判定為類型二；若 Category3_Initial 成立(即 C1 成立)，代表 RR2i 被判定為類型三波段的起始點；在此特別說明類型三(心室早期收縮)，由於心室早期收縮是一整個波段的現象，所以 Category3_Initial 的成立並無法直接判定 RR2i 為類型三的心律不整，而只是暫時將其歸屬於類型三，並持續以規則 Category3_Episode 進行判斷；當 Category3_Episode(即 C2)成立，且成立連續 2 次以上，即正式將 RR2i 歸類為類型三，反之，則必須將所有暫時歸類為類型三的 R-R 區間訊號重新進行其他類型的判斷；若 Category4_Rule 成立(即 C6 成立)，RR2i 被判成為類型四的心律不整；當以上規則都不成立時，表示 RR2i 是正常的，也就是類型一(正常心跳)。上述規則依循程序即為 Tsipouras 演算法的流程，如圖 3.2.4 所示。



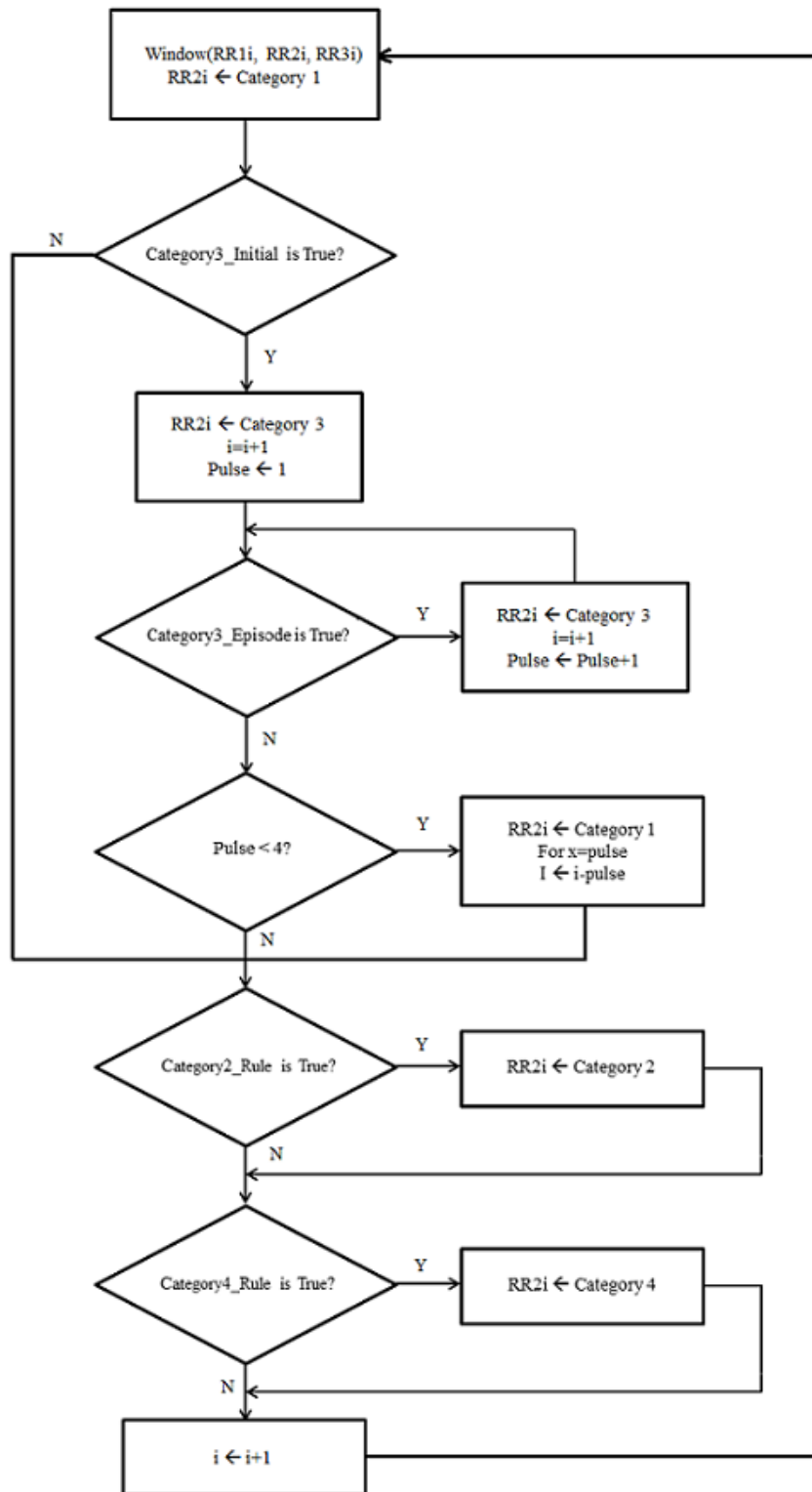


圖 3.2.4 Tsipouras 演算法的流程

3.2.3 模組設計

圖 3.2.5 為監測模組的設計架構，主要由三個子模組組成，分別為斜率與初始閾值計算、R 波偵測與閾值更新、心律不整判斷，每個模組負責不同功能的工作。監測模組的功用主要是判斷心電圖是否異常，若異常則先透過 ECG_Out 送出暫存的 3 分鐘心電圖資料，然後致能 Switch_En 通知系統進行緊急模式，使可重置區配置成壓縮模組。

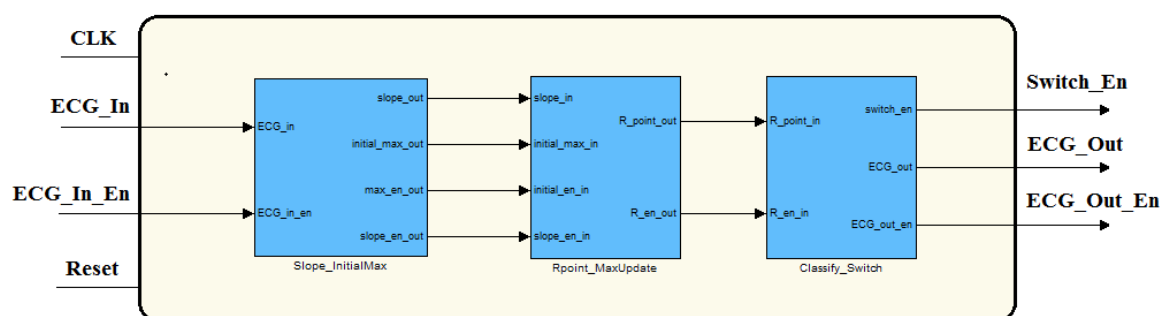


圖 3.2.5 監測模組設計架構

以下子節將對各個子模組做進一步的介紹，由於我們是以 System Generator 作為開發系統的主要工具，並以 Verilog 與 MATLAB 為輔，所以在設計的呈現方面，會以 System Generator 的方塊模組結構來展示設計。

3.2.3.1 斜率與初始閾值計算

在 So and Chan 演算法中，最仰賴的參數就是心電圖訊號的斜率，並以此為基礎偵測出 QRS 波群的確切位置，所以首先我們必須找出訊號的斜率。如公式(3.2.1)式所示，我們只要以延遲的方式擷取訊號前後兩點的值，並加以計算即可得到斜率，實際設計的結果如圖 3.2.6 計算斜率設計方塊圖所示。

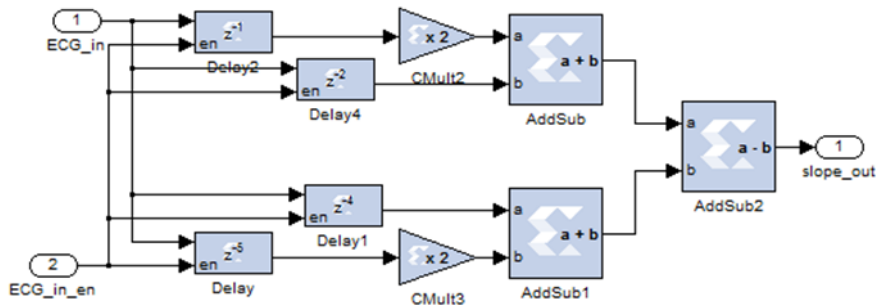


圖 3.2.6 計算斜率設計方塊圖

圖 3.2.7 為 MIT-BIT 提供的紀錄編號 100 心電圖，前 300 點的原始訊號與系統計算出來的斜率，可以看出 R 波的所在位置斜率最大。

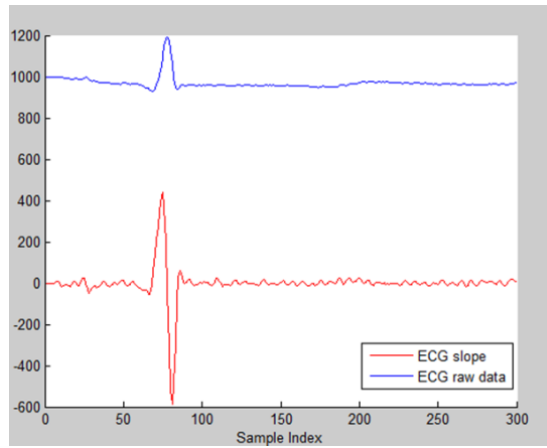


圖 3.2.7 心電圖原始訊號與系統計算之斜率

計算完斜率後，由於一開始尚無法以 R 波為基準訂定閾值，所以我們取前 300 點訊號的斜率最大值作為初始閾值。在這方面的設計，我們主要利用計數器做計算點數的工作，當比對完 300 點訊號的最大值，也就是找出初始閾值後，則接下來的訊號點將被導入正常程序。圖 3.2.8 為子模組完整的設計，在框線內的為計算初始閾值的功能結構，框線以外則為求斜率的部份。

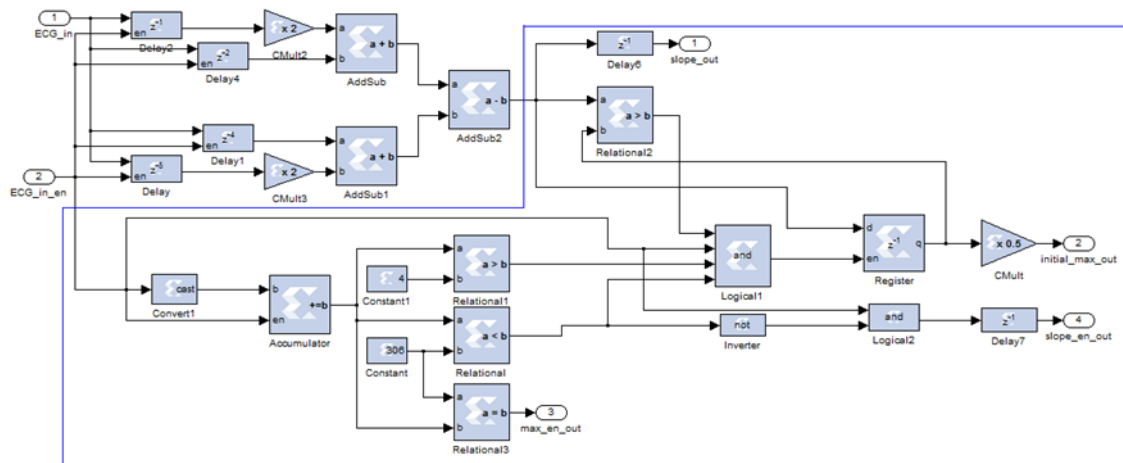


圖 3.2.8 斜率與初始閾值計算模組設計圖

3.2.3.2 R 波偵測與閾值更新

找到斜率與初始閾值後，我們可以開始偵測 R 波的位置，以此找到 R-R 區間，並且不斷更新閾值以維持良好的 R 波偵測靈敏度。雖然此模組的功能不難，但實際設計有頗多細節需要處理，電路因此有些繁瑣。我們以圖 3.2.9 概要說明模組的設計。

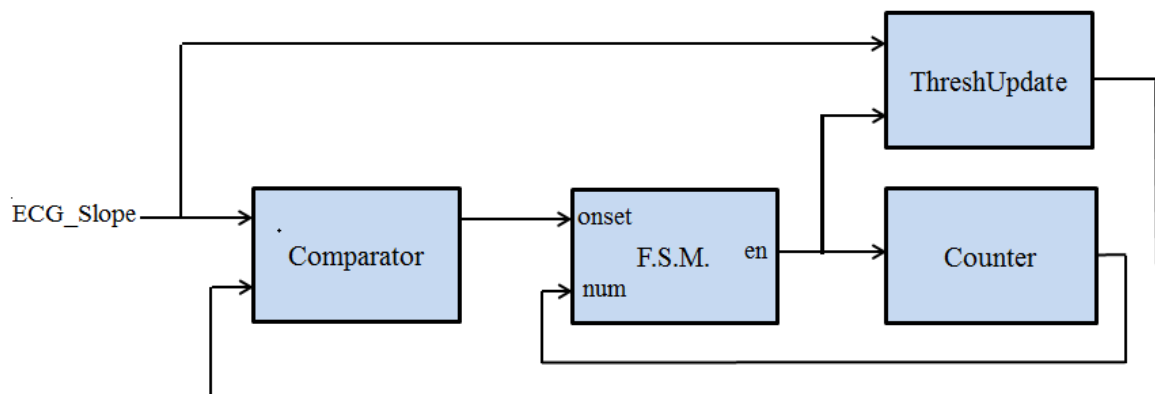


圖 3.2.9 R 波偵測與閾值更新模組簡要示意圖

圖 3.2.9 的核心為有限狀態機(Finite State Machine, FSM)，以此延伸至整個設計，圖 3.2.10 為有限狀態機的 ASM(Algorithmic State Machine)圖。

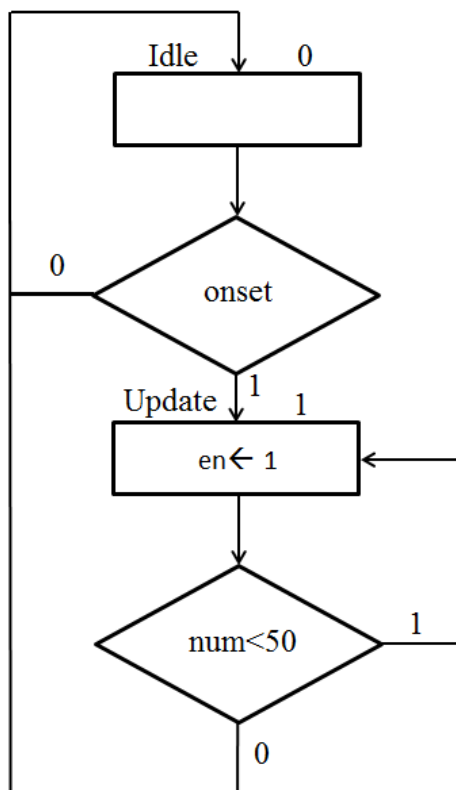


圖 3.2.10 R 波偵測與閾值更新 ASM 圖

圖 3.2.10 中，輸入為心電圖訊號的斜率，它首先進入比較器(Comparator)與目前的閾值比較大小，當連續兩筆心電圖訊號的斜率大於閾值，則會驅動(onset 為真)有限狀態機進入下一個狀態；此時有限狀態機會從狀態 Idle 進入狀態 Update，狀態 Update 會使 en 為真，以此驅動計數器(Counter)進行計數，同時致能閾值更新(ThreshUpdate)功能方塊，找出接下來 50 點心電圖訊號的斜率最大值，並以先前介紹的公式(3.2.2)、(3.2.3)、(3.2.4)進行閾值更新；更新完成後，有限狀態機回到 Idle 狀態，重覆進行上述的步驟。如同我們一開始的說明，以上描述的機制尚有細微之處需要處理，但我們已經傳達了這個模組的重點。詳細的設計可以參考完整的電路設計圖。圖 3.2.11 是此模組完整的 System Generator 設計電路圖。

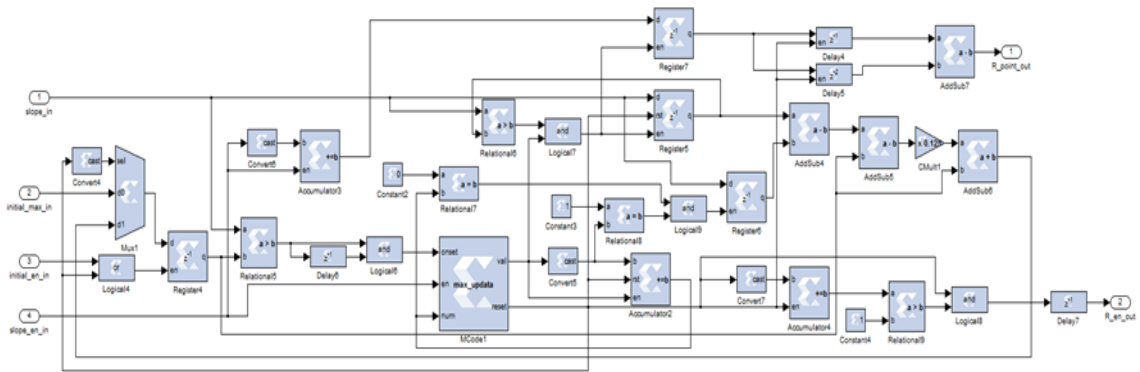


圖 3.2.11 R 波偵測與閾值更新模組設計圖

圖 3.2.12 是系統計算 MIT-BIH 紀錄編號 100 的心電圖訊號其前 300 點的 R-R 區間訊號。

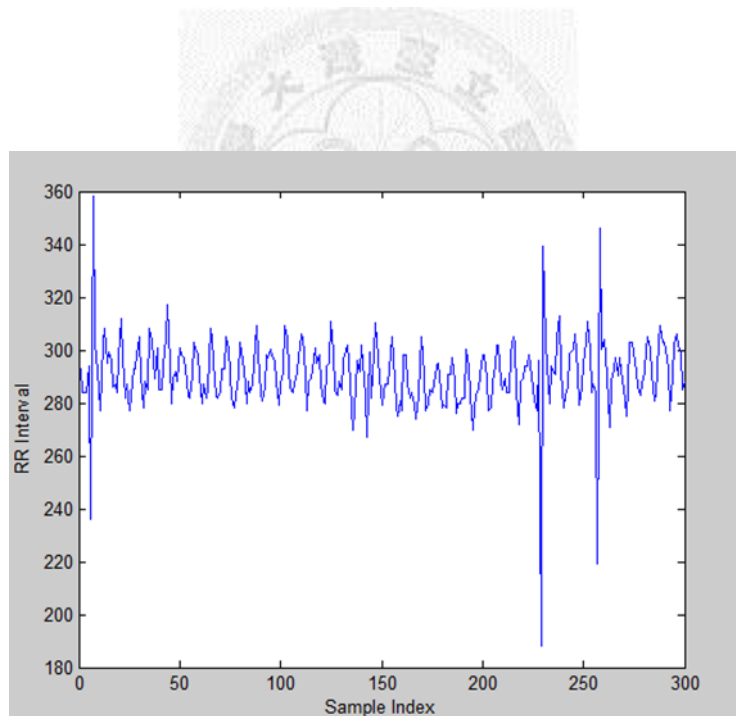


圖 3.2.12 心電圖 R-R 區間訊號

3.2.3.3 心律不整判斷

在上節中我們已經偵測出 R 波的所在位置，接下來我們可以利用 R-R 區間的距離判別心律不整。在 3.2.1 節中我們曾定義判斷心律不整的規則(表 3.2.3)，分別是 Category2_Rule、Category3_Initial、Category3_Episode、Category4_Rule，它們又各自包涵判定的條件(表 3.2.2)，為了方便設計，我們將 4 個規則的邏輯電路設計成更底層的子模組，使得我們心律不整判定模組的設計可以更有組織、簡潔。圖 3.2.13、圖 3.2.14(a)及(b)、及圖 3.2.15 分別為 4 個規則的邏輯電路設計圖。

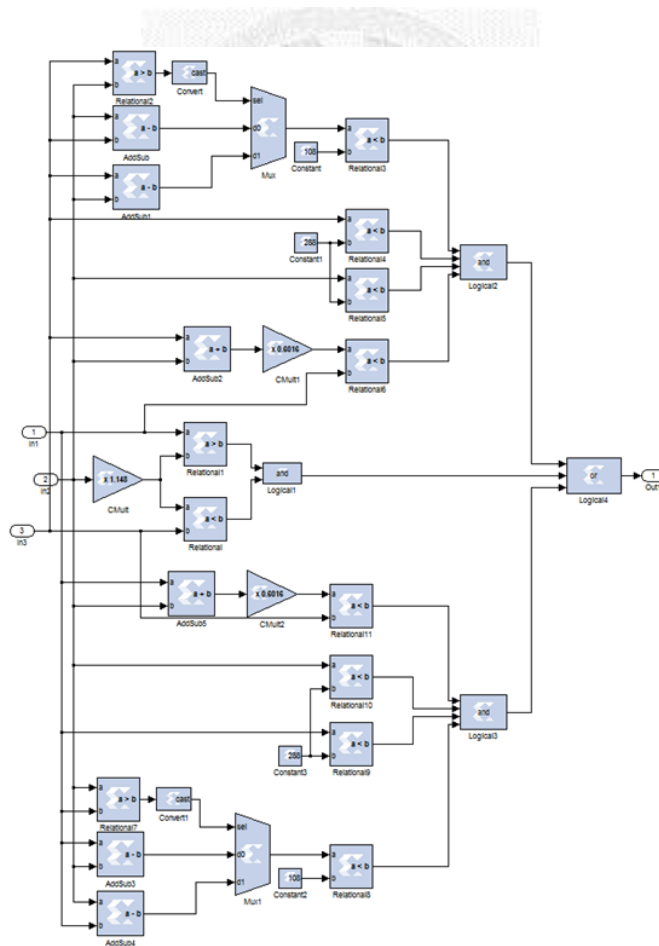


圖 3.2.13 規則 Category2_Rule 邏輯電路圖

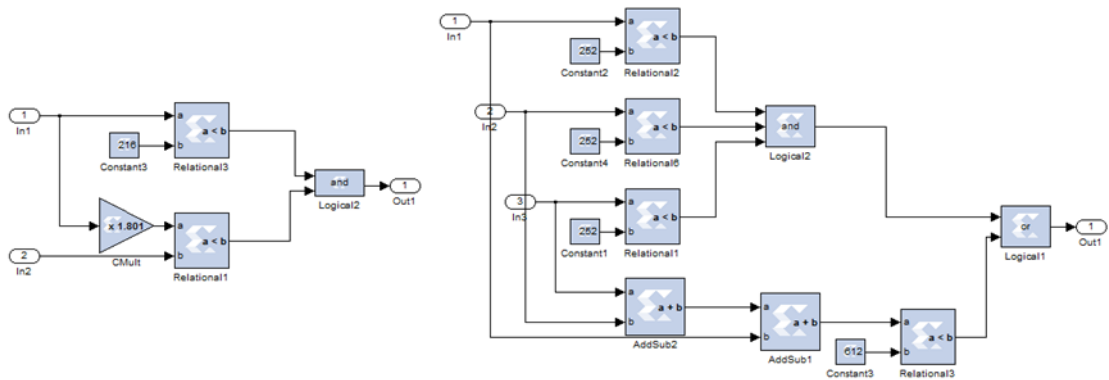


圖 3.2.14(a)規則 Category3_Initial 邏輯電路圖；(b) 規則 Category3_Episode 邏輯電路圖

路圖

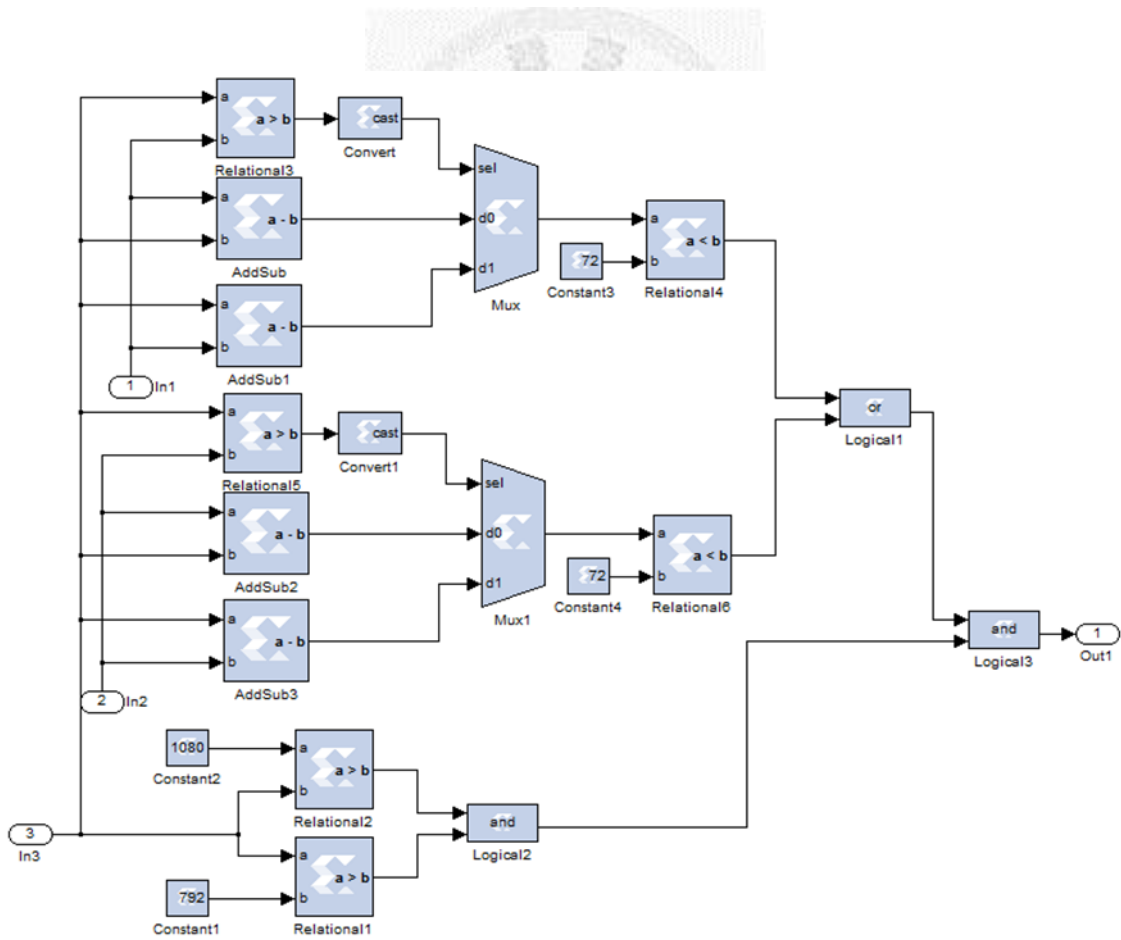


圖 3.2.15 規則 Category4_Rule 邏輯電路圖

在心律不整的判別上，類型二與類型四屬於可以直接被判定的心律不整，但比較麻煩的是，類型三需要數筆 R-R 區間的資料才能作判讀；當有連續三筆以上的 R-R 區間訊號被暫時歸類為類型三時，表示這些訊號真正屬於類型三的心律不整；反之，若僅有三筆以下的 R-R 區間訊號被暫時歸類為類型三時，則所有暫時被歸類為類型三的 R-R 區間訊號，需要重新做其他類型的判別。我們使用一個有限狀態機去解決上述工作，圖 3.2.16 所示為有限狀態機的 ASM 圖。

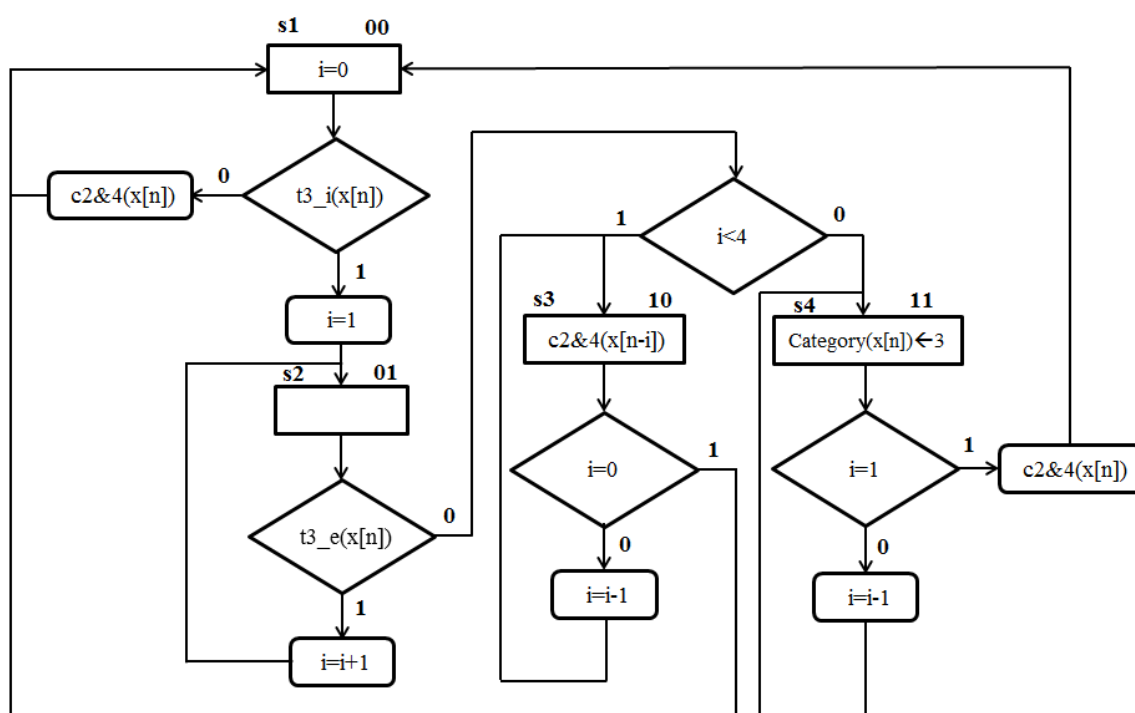


圖 3.2.16 心律不整判斷 ASM 圖

圖 3.2.16 中我們定義了一些符號，在此作說明：

- $x[n]$: R-R 區間訊號。
- $t3_i()$: 規則 Category3_Initial 的邏輯函數；輸入為 R-R 區間訊號，輸出為訊號經 Category3_Initial 邏輯運算的輸出。
- $t3_e()$: 規則 Category3_Episode 的邏輯函數。輸入為 R-R 區間訊號，輸出為

訊號經 Category3_Episode 邏輯運算的輸出。

- c2&4()：規則 Category2_Rule 與 Category4_Rule 的分類函數。輸入為 R-R 區間訊號，輸出為訊號經 Category2_Rule 與 Category4_Rule 邏輯運算的輸出；當 Category2_Rule 為真則將訊號分類為類型二，當 Category4_Rule 為真則將訊號分類為類型四，若都不成立則歸類為類型一，即正常心跳。
- Category()：輸入為 R-R 區間訊號，輸出為訊號的類型。

圖 3.2.17 為心律不整判斷模組的完整的設計，其中黃色方塊為每個規則的邏輯電路，紅色接線為有限狀態機連結往其他功能方塊的接線。

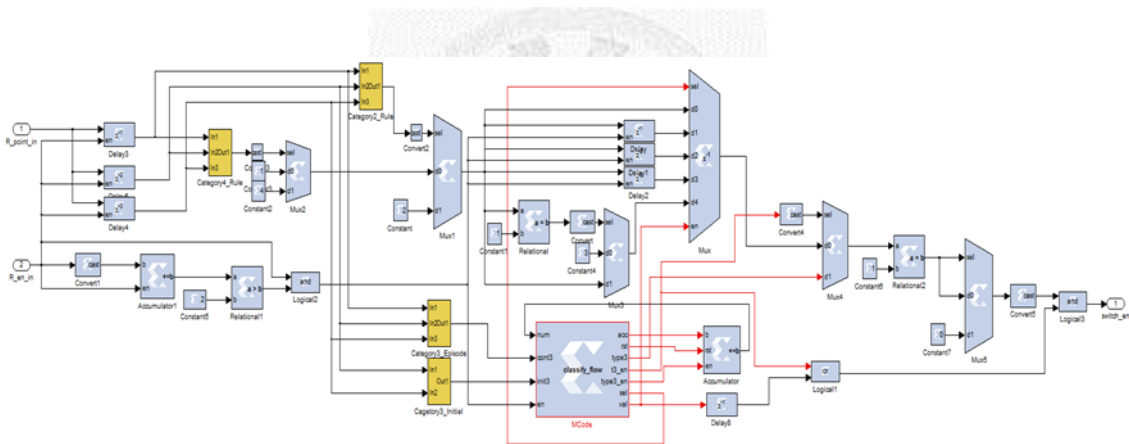


圖 3.2.17 心律不整模組設計圖

3.3 壓縮模組

當進入緊急模式時，可重置區將把監測模組置換成壓縮模組；壓縮模組的功能就是將心電圖訊號作壓縮的動作，以此減小對儲存裝置的需求或是傳送的可靠性與接收端的負荷。我們選用以離散小波轉換(Wavelet Transform Discrete)為基礎的訊號壓縮架構，小波轉換可以用於原始數據(例如圖像)，然後將轉換後的數據編碼，得到有效的壓縮，如圖 3.3.1 所示。此架構也被廣泛使用在一些生理訊號的處理上[27]。

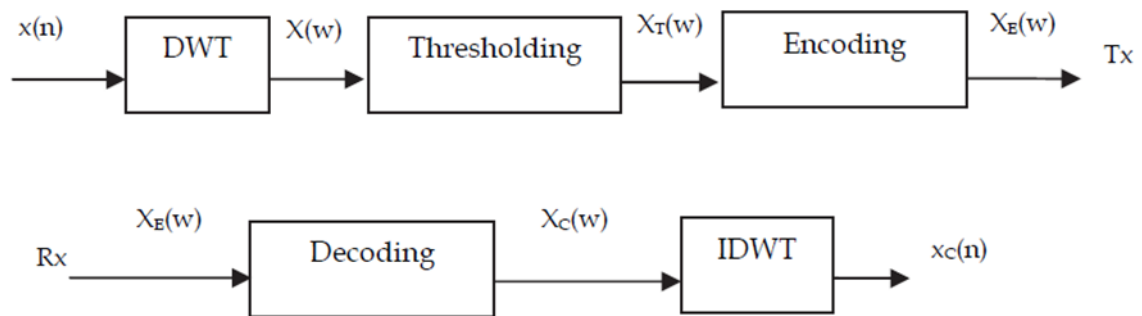


圖 3.3.1 離散小波轉換壓縮架構[27]

壓縮模組的主要功能有兩個：小波轉換與 Run-Length 編碼，以下子節將先說明這兩個方法，然後再介紹模組的設計。

3.3.1 離散小波轉換

離散小波轉換（Discrete Wavelet Transform, DWT）是一種具多重解析度的演算法，它能將資料或訊號分解成不同的頻帶，並針對資料的重要性進行分解。離散小波轉換可視為把信號經一組濾波器之後的輸出，如圖 3.3.2 所示。

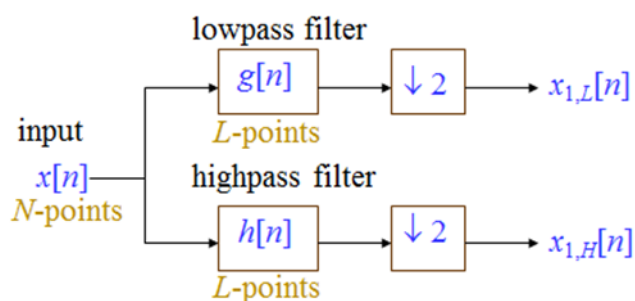


圖 3.3.2 一階離散小波轉換示意圖

$g[n]$ 和 $h[n]$ 分別代表離散小波轉換在分解時的一組高通與低通濾波器， $\downarrow 2$ 代表的是降階取樣，經過降階取樣之後，信號長度變為原來的一半。訊號經過小波轉換後，訊號會被分解成兩個頻帶，分別是高頻 $x_{1,H}[n]$ 與低頻 $x_{1,L}[n]$ 兩個部分。以影像為例，高頻部分通常是影像中較為細微的部份，而低頻成份則是屬於影像中比較平滑、沒有太大的像素變化的部份，但對於人類的視覺而言，低頻反而是辨識影像的重點，例如圖 3.3.3 是影像經一階二維小波轉換後的原圖與結果，可以發現小波轉換後的圖片其最低頻的部份與原圖幾乎相同。

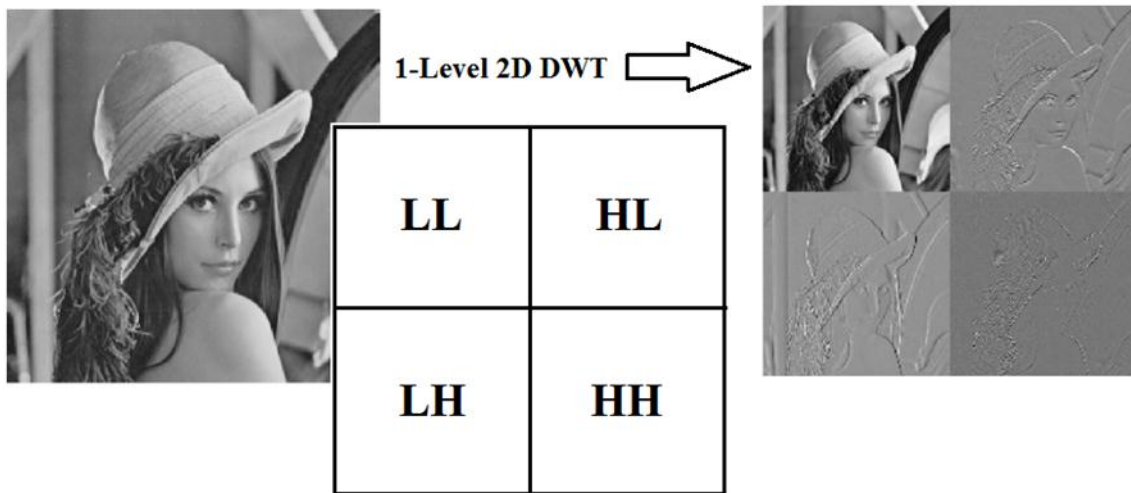


圖 3.3.3 圖片經過一階二維小波轉換的處理結果

同樣的，我們將小波轉換運用在心電圖訊號上，如圖 3.3.4 所示，圖 3.3.4 (a) 為原始心電圖訊號，圖 3.3.4 (b) 是經過二階一維小波轉換後最低頻部份的資訊，可以從圖上觀察到，雖然細節的小突波不見了，但波形的趨勢起伏依舊非常清楚，並不會影響到心律不整的判別。

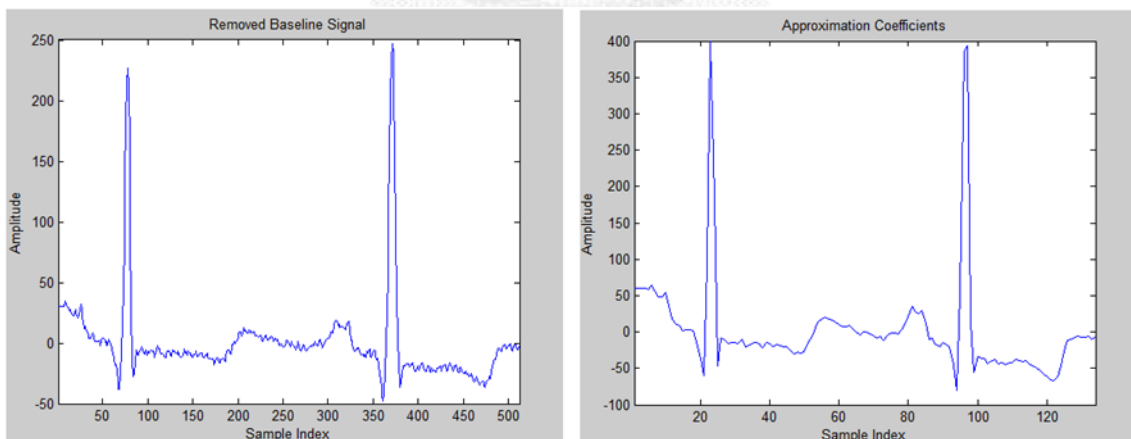


圖 3.3.4 (a)心電圖原始訊號；(b)心電圖經小波轉換後的低頻訊號

傳統離散小波轉換的運算量大，且需要使用不少的硬體資源去實現，為了解決這樣的問題，1996 年 Sweldens 提出上提式離散小波轉換[28]，其主要優點是使用較少的硬體；以 5/3 濾波器為例，乘法器只需要 2 個，加法器為 4 個，均是傳統 DWT 硬體的一半，且大幅降低計算複雜度。5/3 濾波器上提式小波轉換架構如圖 3.3.5，主要包含三個部分，分別為 Splitting 模組、Lifting 模組、以及 Scaling 模組。

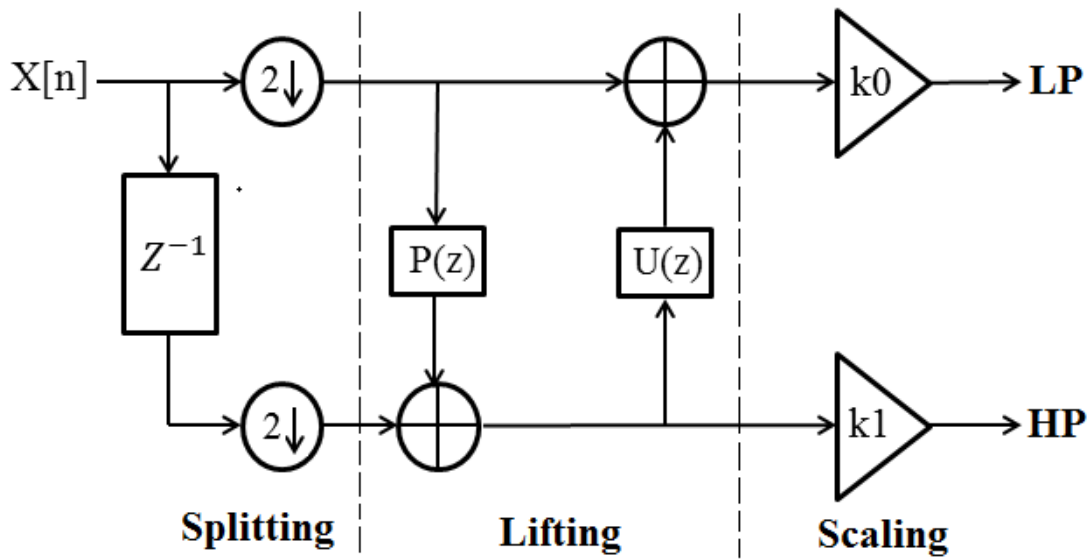


圖 3.3.5 5/3 濾波器上提式小波轉換架構

其中 $P(z)$ 與 $U(z)$ 由公式 (3.3.1) 與 (3.3.2) 表示：

$$P(z) = \frac{1}{4}(1 + z^{-1}) \quad (3.3.1)$$

$$U(z) = -\frac{1}{2}(1 + z) \quad (3.3.2)$$

上提式離散小波的動作步驟，主要是先將輸入 Splitting 模組分解成奇數項及偶數項，然後將偶數項送入 Lifting 模組的運算單元 $P(z)$ ，並與奇數項相加得到高頻係數，之後把得到的高頻係數送入運算單元 $U(z)$ ，並與偶數項相加得到低頻係數；最後則是高低頻係數通過 Scaling 模組乘上常數後輸出。

由於一般離散小波轉換是以迴旋積的方式進行計算，所以結果會以實數的資料型態出現；然而浮點數有一定的精準度，對硬體的實現上也會有其困擾，基於此 Sweldens 於 1998 年提出了所謂的上提式整數對整數小波轉換[29]，在原本的架構上加入高斯函數(取整函數)，將浮點數轉換成整數；如圖 3.3.6 所示，為 5/3 濾波器上提式整數對整數小波轉換的架構。

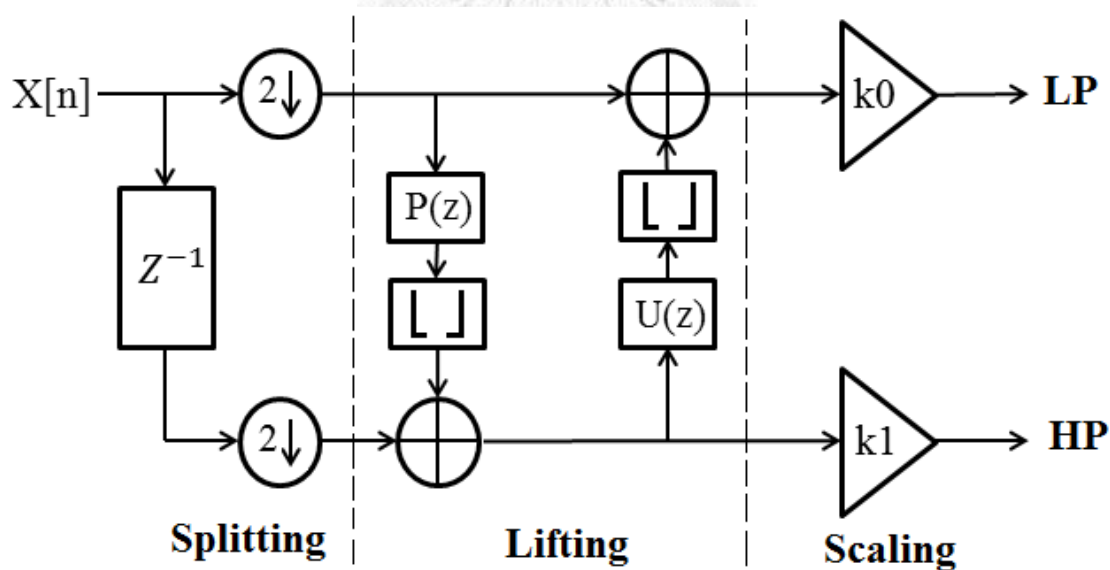


圖 3.3.6 5/3 濾波器上提式整數對整數小波轉換架構

其轉換公式如公式 (3.3.3)、(3.3.4)：

$$d[n] = X[2n+1] - \left\lfloor \frac{X[2n] + X[2n+2]}{2} \right\rfloor \quad (3.3.3)$$

$$s[n] = X[2n] - \left\lfloor \frac{d[n-1] + d[n] + 2}{4} \right\rfloor \quad (3.3.4)$$

3.3.2 Run-Length 編碼

Run-Length 編碼(Run Length Encoding, RLE) 是一種簡單的壓縮方法。由於其方法簡易，因此在壓縮以及解壓縮方面速度都非常快，所以被使用在很多的實際應用上。Run-Length 編碼的基本方法是把一串相同的資料簡縮成兩筆資料表示，第一筆表示的是該資料串的重複次數，第二筆則是用來記錄該重複的資料。這個壓縮方式，在該資料的重複次數越多時，壓縮的效果越好。圖 3.3.7 展示了一個利用 Run-Length 編碼的例子。

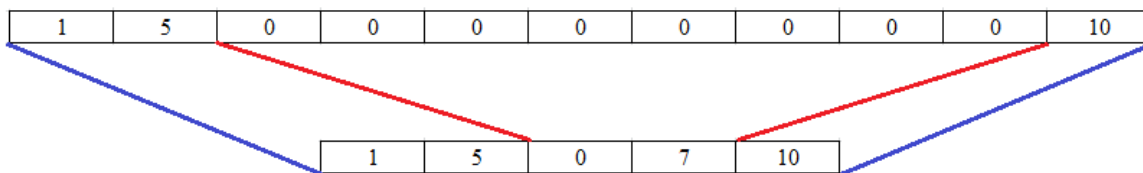


圖 3.3.7 Run-Length 編碼實例

心電圖經小波轉換後能量將會集中在低頻，如圖 3.3.8 所示；若再通過閾值的篩選後，大多數的小波係數都會是零，這恰符合 Run-Length 編碼的要求。

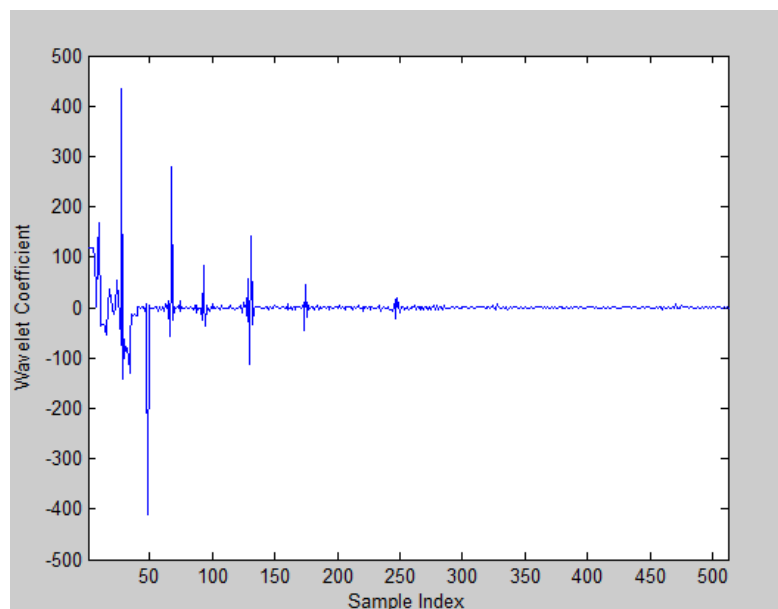


圖 3.3.8 心電圖訊號的小波轉換係數

3.3.3 模組設計

圖 3.3.9 為壓縮模組的設計架構，主要由四個子模組組成，每個子模組負責不同的功能，由左而右分別是：資料擷取與位移、 $5/3$ 整數對整數小波轉換、閾值篩選、Run-Length 編碼。

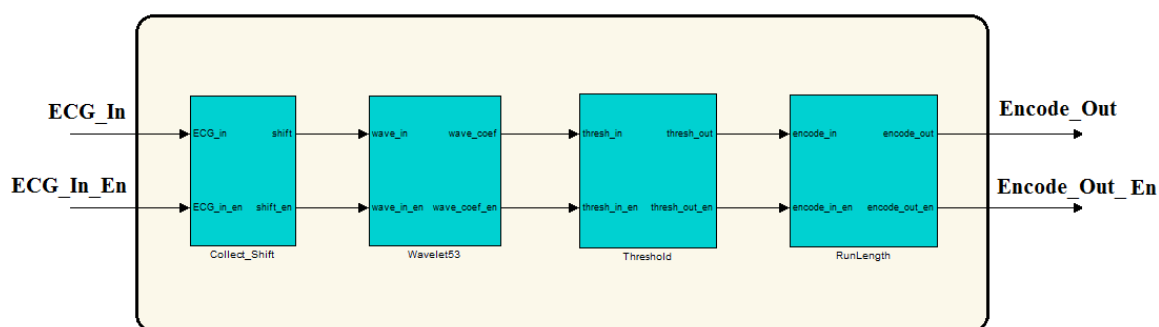


圖 3.3.9 壓縮模組設計架構

壓縮模組的功能顧名思義即是把資料作壓縮的動作；它的啟用時機是當監測模組偵測到異常的心電圖訊號時，系統會從正常模式進入緊急模式，並同時將可重置區置換成壓縮模組，此後進入系統的心電圖訊號都會經過壓縮模組的處理，然後將壓縮好的資料傳送出去。以下子節將對各個子模組的設計加以說明。

3.3.3.1 資料擷取與位移

訊號在做小波轉換之前，需要做一些前置的處理：

1. 小波轉換的資料長度會影響壓縮的效率，基於 MIT-BIH 心電圖訊號的取樣率和系統即時(Real-time)處理的設計考量，我們的小波轉換長度取 512 點為一個區段；我們使用深度為 1K 容量的 FIFO(先進先出資料儲存器)儲存輸入訊號，當它存滿一半即 512 點資料時，控制電路會至 FIFO 依序取走 512 點資料進行後續處理。
2. MIT-BIH 提供的心電圖一筆資料是以 12 位元表示，但實際用到的位元只有 11 位元，所以可以表示的範圍是[0, 2047]；但這個範圍的值在計算方根誤差百分比(Percent root-mean-square Difference, PRD)時，可能會獲得過低且不夠客觀的

評估，為了解決這樣的疑慮，而我們必須將所有值都減去 1024，將數值範圍平移至[-1024, 1023]。另外，經過平移後的數值範圍在處理上也可以減少硬體的資源。

圖 3.3.10 所示為實際設計的電路圖。圖 3.3.11(a)為原始的心電圖訊號，圖 3.3.11(b)為經過平移後的心電圖訊號。

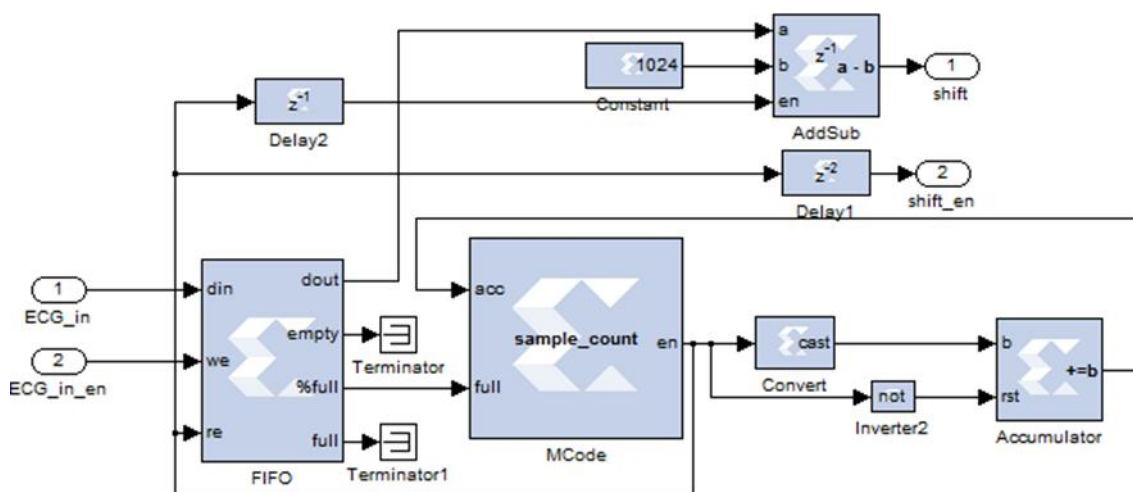


圖 3.3.10 資料擷取與位移模組設計圖

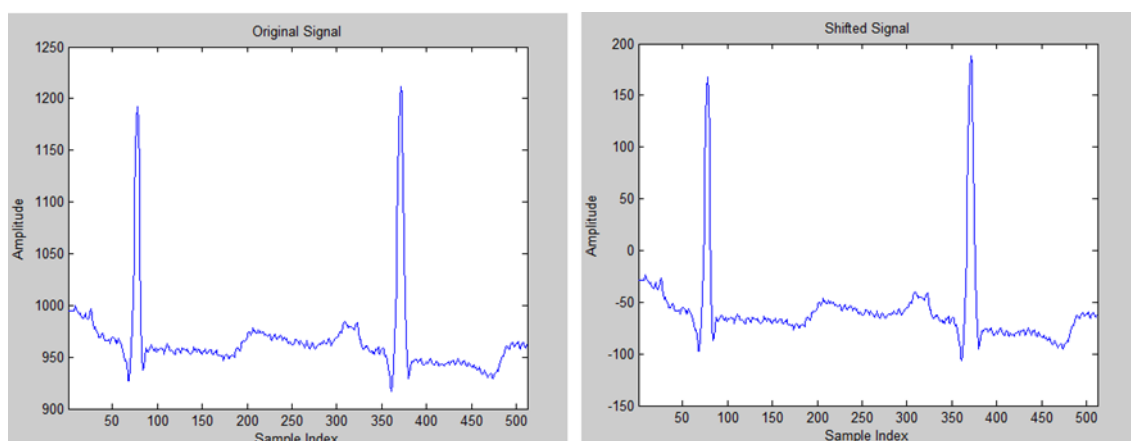


圖 3.3.11(a)原始心電圖訊號；(b)平移後的心電圖訊號

3.3.3.2 5/3 整數對整數小波轉換

我們壓縮模組的核心是四階的 5/3 整數對整數小波轉換，它具備一般離散小波轉換的優點，且適合被實現在硬體架構上；選擇四階是配合我們選取的資料長度 512 點而定，經過模擬的結果顯示，四階的小波轉換擁有較高的處理效率。我們先說明一階的 5/3 小波轉換的設計，如圖 3.3.12 所示為一階的硬體架構。

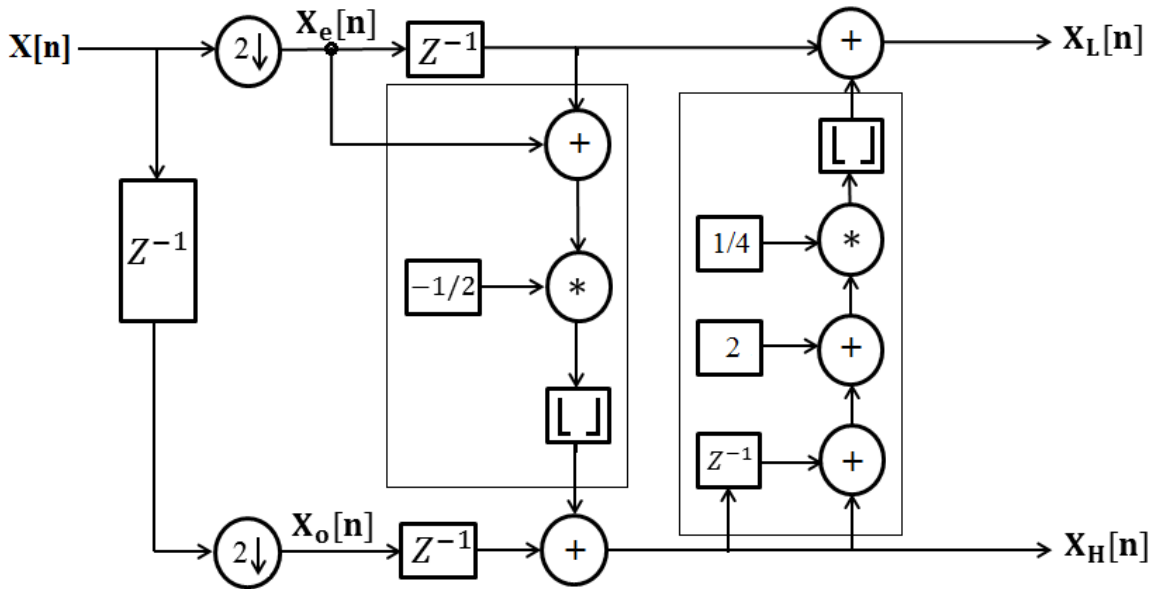


圖 3.3.12 一階 5/3 整數對整數小波轉換硬體架構

輸入訊號 $X[n]$ 經降低取樣後取得偶數點 $X_e[n]$ 與奇數點 $X_o[n]$ ，為了對準訊號點的位置，兩訊號同時經過一個單位的延遲；之後 $X_e[n]$ 與延遲一個單位的 $X_e[n]$ 相加(即 $X[2n]$ 與 $X[2n+2]$ 相加)，將結果乘上常數 $(-1/2)$ 並通過高斯函數取整數，再把輸出加上經過一個單位延遲的 $X_o[n]$ (即 $X[2n+1]$)，相加的結果即為公式 (3.3.3) 描述的高頻係數 $X_H[n]$ 。接下來 $X_H[n]$ 與延遲一個單位的 $X_H[n]$ 相加，然後加上常

數(2)並乘上常數(1/4)後通過高斯函數取整數，其輸出與 $X[2n]$ 相加即是公式 (3.3.4) 所描述的低頻係數 $X_L[n]$ 。圖 3.3.13 為一階 5/3 整數對整數小波轉換的實際設計圖。

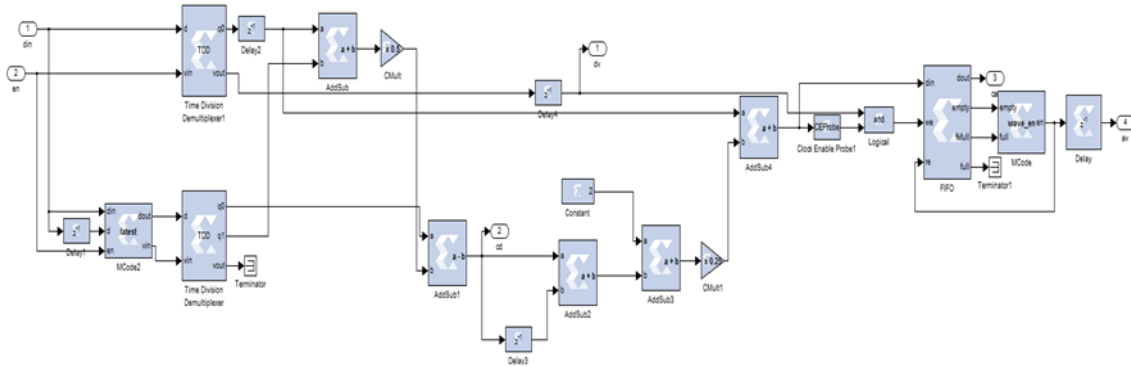


圖 3.3.13 一階 5/3 整數對整數小波轉換設計圖

在完成一階 5/3 小波轉換的設計後，我們將其規劃成子模組，如圖 3.3.14 所示為子模組的接腳圖； $X[n]$ 為上一級子模組的輸出低頻係數， En 為其有效信號， $X_L[n]$ 與 $X_H[n]$ 是此模組的低頻係數與高頻係數， En_L 與 En_H 分別為其有效訊號。接下來我們只需將 4 個一階 5/3 小波轉換做適當的連結，即可完成四階 5/3 小波轉換的設計。

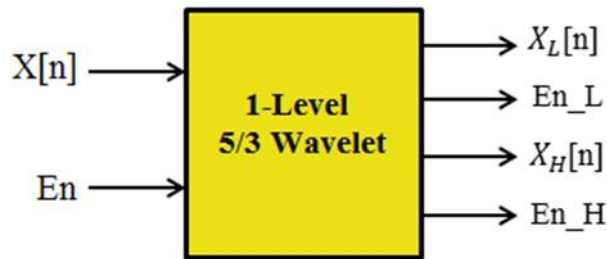


圖 3.3.14 一階 5/3 小波轉換模組接腳圖

圖 3.3.15 所示為四階 5/3 整數對整數小波轉換的設計圖，每個一階 5/3 小波轉換(黃色方塊)輸出低頻係數與其有效信號，作為下一階的輸入；另外為了整併所有子模組輸出的小波係數，我們將所有有效的小波係數的輸出連結到同一個多工器 (Mux)(橘色方框)，並將小波係數輸出的有效信號依子模組的階序進行編號 (紅色方框)，以此作為多工器的選擇線。整併完小波係數，由於子模組的小波係數輸出在時間上並不會重疊產生，所以我們將子模組所有有效信號連結到同一個或邏輯模組上，以此作為完整的小波係數的有效信號。

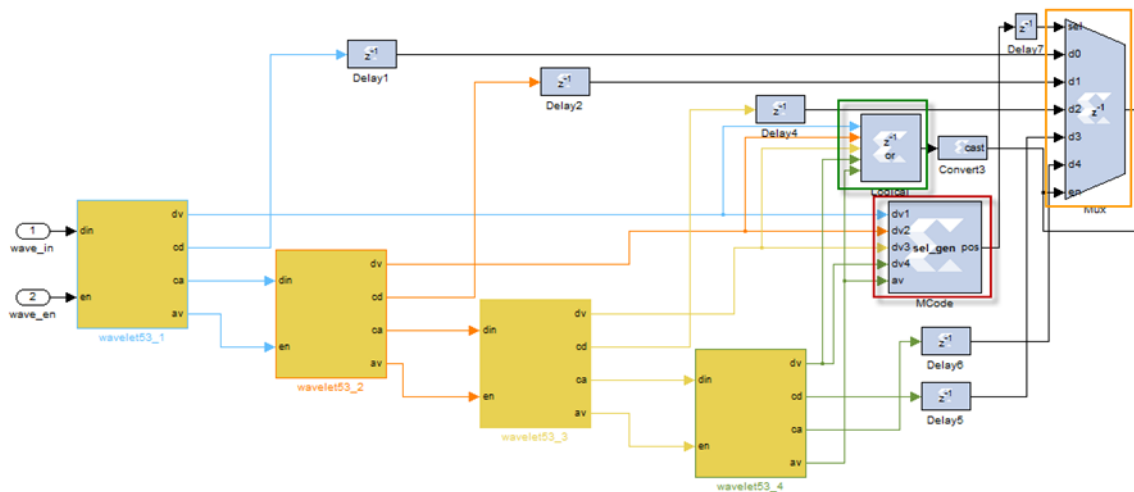


圖 3.3.15 四階 5/3 整數對整數小波轉換設計圖

圖 3.3.16 為小波係數的結果，由上而下分別是 256 點的 $X_{H1}[n]$ 、128 點的 $X_{H2}[n]$ 、64 點 $X_{H3}[n]$ 、及 32 點的 $X_{H4}[n]$ 和 $X_{L4}[n]$ 。

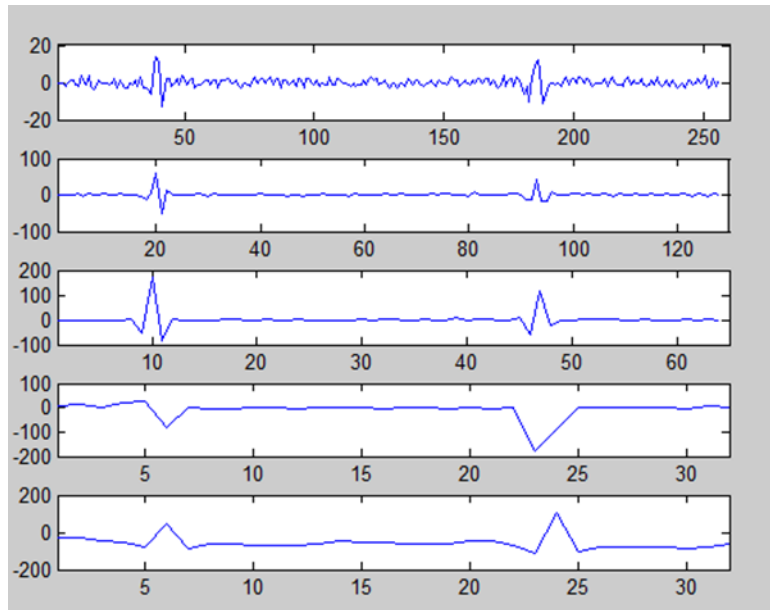


圖 3.3.16 四階 5/3 整數對整數小波轉換係數

3.3.3.3 閾值篩選

為了減少小波係數傳送的個數，以此提高壓縮比率，所以我們要藉由訂定閾值的方式，去篩選出能量高的小波係數進行傳送，並兼顧高的訊號重建率。經過模擬不斷的測試，我們訂定了符合上述需求的閾值公式，如公式 (3.3.5) 所示。

$$2^{\lfloor \log_2(\max(X) - (L+1)) \rfloor} \quad (3.3.5)$$

其中 X 為小波係數， L 為小波轉換的階數， $\lfloor \ \rfloor$ 為取整函數。原本在公式 (3.3.5) 中存在的對數運算(log)，對硬體的設計上會稍嫌複雜，但由於其最終經過一個取整運算，也就是說我們並不需求出對數運算(log)的精確值，這在設計上將精簡

許多。所以我們僅需設計一個查找表，然後視取整函數內的數值範圍，輸出一個預先定義的值即可。圖 3.3.17(a)為未經閾值篩選的小波係數，圖 3.3.17(b)為經過閾值篩選的小波係數，可以發現一些能量比較小的係數都已經被過濾掉了。圖 3.3.18 為閾值篩選模組設計電路圖。

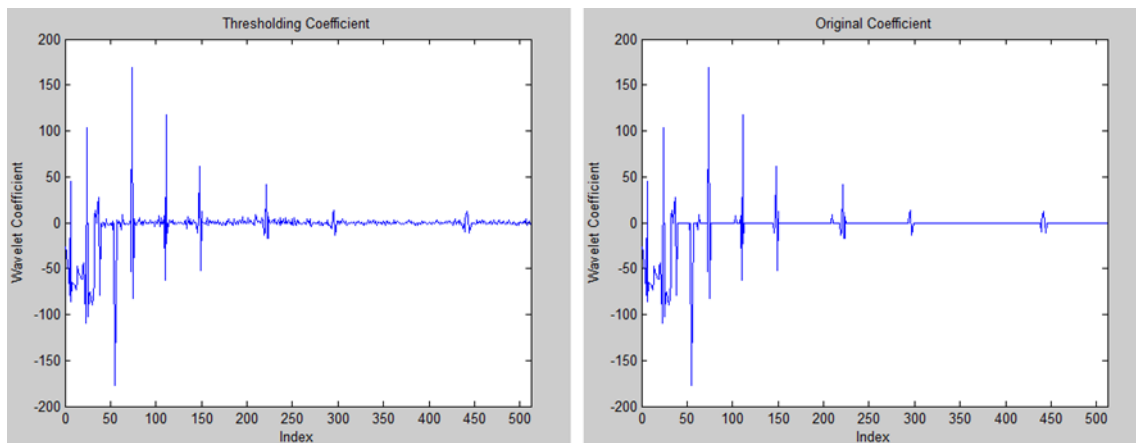


圖 3.3.17(a)原始的小波係數；(b)經過閾值篩選的小波係數

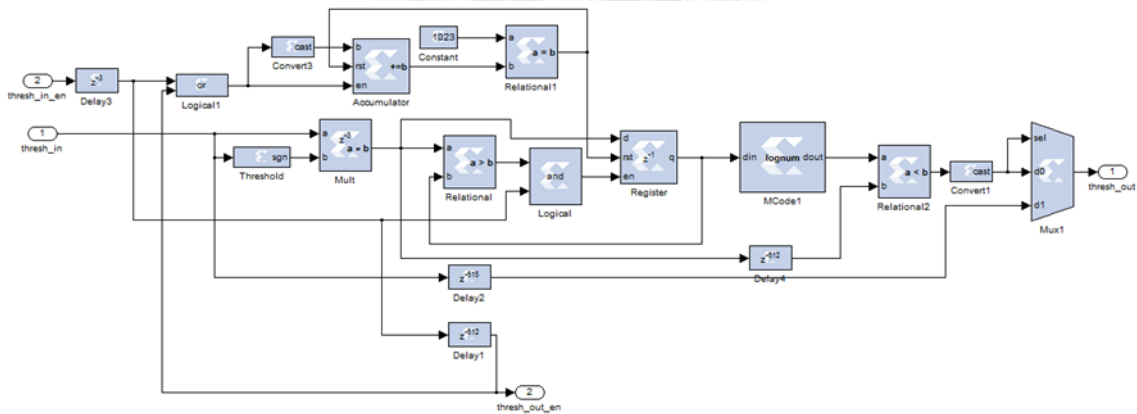
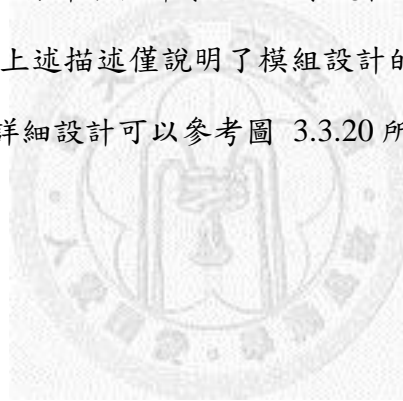


圖 3.3.18 閾值篩選模組設計圖

3.3.3.4 Run-Length 編碼

在小波係數經過篩值後，我們要進入最後的階段 Run-Length 編碼，此步驟是真正的降低需要傳送的小波係數個數。由於我們在篩值階段將能量低的小波係數都設定成零，這將讓我們的編碼效果更好。圖 3.3.19 為 Run-Length 編碼的 ASM 圖，概念其實很簡單，就是當輸入為非零時，由於無須編碼所以將值直接輸出；而當輸入為零時，必須啟動計數器的功能，只要接下來的輸入仍為零，則計數器持續進行計數，直接接收到非零的值時，模組依序輸出計數器的值、零、與當下的值，最後回到初始狀態並將計數器歸零。由於系統在設計上是以即時(Real-Time)處理為考量，所以事實上上述描述僅說明了模組設計的概要，實際上還要考量一些關於時間的細節問題。詳細設計可以參考圖 3.3.20 所示的 Run-Length 編碼完整的設計圖。



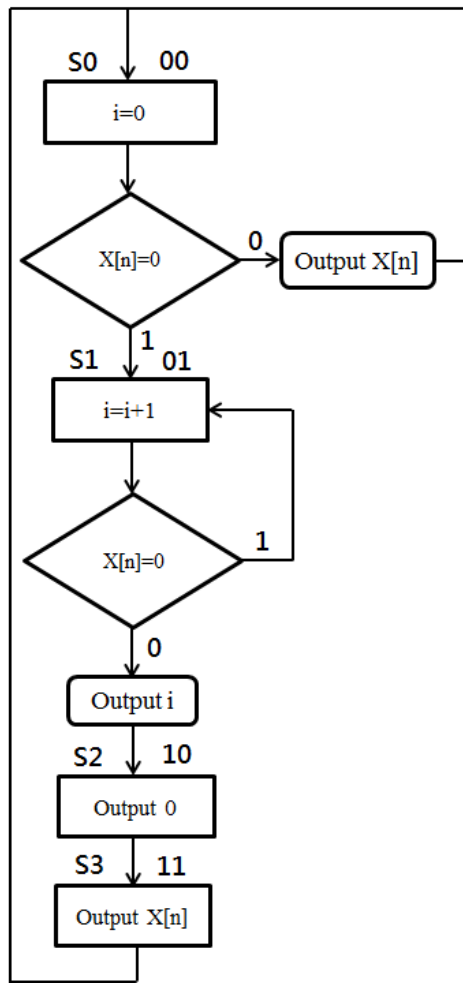


圖 3.3.19 Run-Length 編碼 ASM 圖

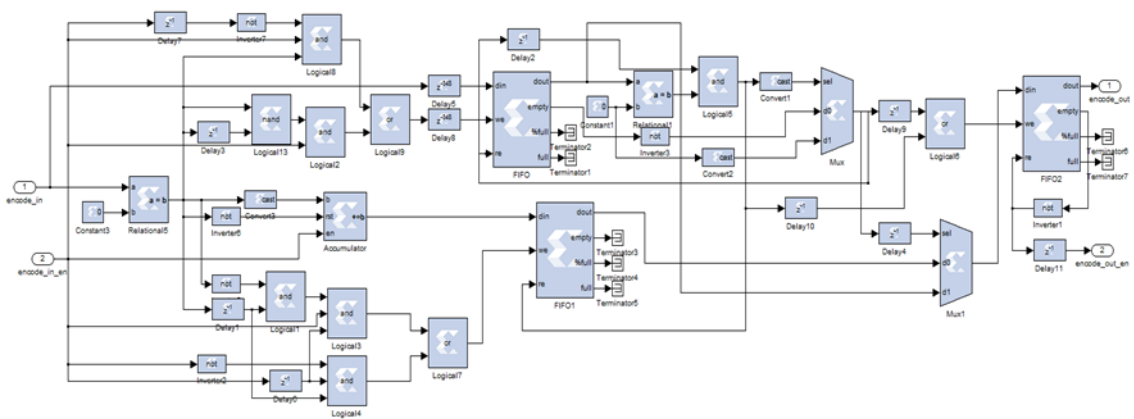


圖 3.3.20 Run-Length 編碼模組設計圖

3.4 MicroBlaze 處理器系統模組

MicroBlaze 處理器系統主要功能是控制週邊元件、及控制可重置區的切換工作。由於 MicroBlaze 處理器系統的設計多半是在軟體程序，圖 3.4.1 為 MicroBlaze 處理器系統的架構，接下來我們將對每個裝置與運作機制做說明。

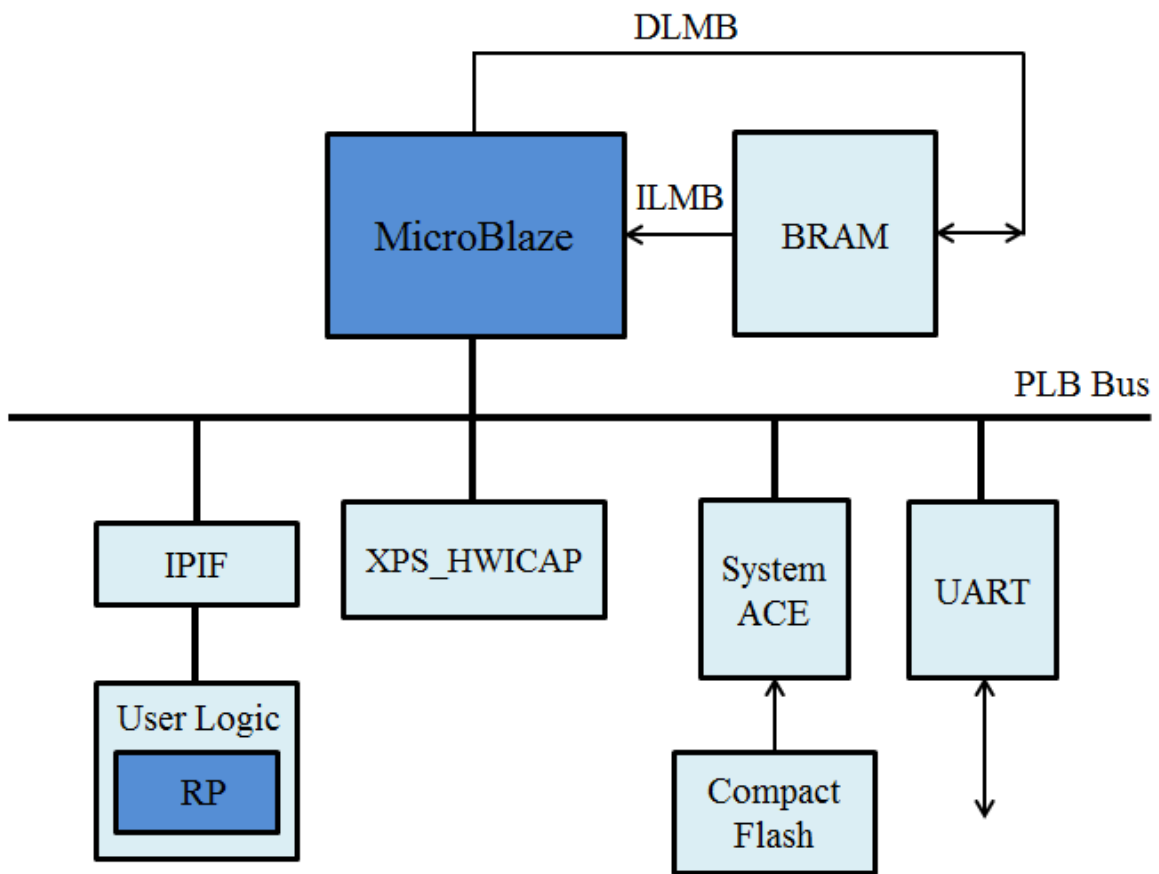


圖 3.4.1 MicroBlaze 處理器系統架構

1. 本地記憶體匯流排(Local Memory Bus, LMB)最主要是用來連接 BRAM(Block RAM)，又可以分成指令匯流排 ILMB 與資料匯流排 DLMB，通常 MicroBlaze 處理器對於 LMB 的存取速度大約為 1 個時序週期。

2. 處理器本地匯流排(Processor Local Bus, PLB)最主要用來連結 MicroBlaze 與各個週邊裝置及外加的記憶體，PLB 亦是 IBM CoreConnect 的架構，提供了 64 位元的特殊區域性處理器之間的傳輸。
3. 於 PLB 上我們可以掛上系統所提供的週邊裝置，或是使用者自己設計的 IP。
4. IPIF(IP Interface)其主要功能是用來產生與 PLB 匯流排與使用者邏輯電路中間的溝通與控制訊號；使用者邏輯電路包括使用者自行設計的 IP 與 IPIC(IP Interconnect)訊號產生器，有了控制訊號我們才能夠透過 IPIF 直接與處理器溝通。
5. Bitstream 檔案存放在 Compact Flash 上，SystemACE 則為控制 CompactFlash 的微控制器。在需要作部份可重置時 MicroBlaze 從 Compact Flash 上讀取 Bitstream 檔案，然後通過匯流排將 Bitstream 檔案傳送給封裝 ICAP 模組的 HWICAP，最後通過 HWICAP 的控制模組將配置資料寫入 ICAP 介面。
6. 通用非同步收發器(Universal Asynchronous Receiver/Transmitter, UART)負責以串聯的方式建立兩個裝置之間的連線。UART 在從週邊裝置接收到資料字元時，執行序列到平行的轉換；從處理器接收到資料字元時，執行平行到序列的轉換。在運作過程中，處理器可以讀取 UART 所有狀態。狀態資訊包含傳送操作的種類及條件，以及與 UART 相關的錯誤情況(同位元、過載、訊框組成、或中斷)。

由於 MicroBlaze 處理器系統的設計屬於軟體程序的應用，所以以下就我們所建立具備部份可重置技術 MicroBlaze 處理器系統的軟體流程作概要的介紹：

1. 利用硬體描述語言、System Generator 等設計方式建立階層式設計(Hierarchical Design)；其中頂層(Top-Level)設計定義所有系統子模組，其中包括靜態模組

與動態模組；靜態模組即不隨時間更動的使用邏輯，動態模組為系統視需求可以置換的設計，也就是實現部份可重置功能。

2. 使用 EDK 建立微處理器(MicroBlaze)系統，並加入需要的週邊(如 ICAP)與撰寫 C 語言控制程式，微處理器系統是靜態邏輯的一部份，同樣需在頂層設計內被定義。
3. 使用 PlanAhead 詳細規劃部份可重置設計；包括定義可重置區 (Reconfigurable Partition)範圍、加入可重置模組(Reconfigurable Module)、定義所需的不同的配置(Configuration)等。
4. 產生 bitstream 檔案(包括 full.bit 及 partial.bit)，燒錄至 FPGA 晶片內做測試與驗證。



第四章 實驗結果與討論

4.1 實驗資料與平台

4.1.1 MIT-BIH 心律不整資料庫

本研究所使用的實驗與驗證資料，是 MIT-BIH 心律不整資料庫的心電圖數據 [32]，MIT-BIH 是由美國麻省理工學院提供研究的心律不整資料庫，是目前國際上公認可作為標準的心電圖資料庫之一；資料庫共提供 48 筆心電圖紀錄，每筆大約 30 分鐘長，其中資料所採用的取樣頻率是 360Hz，電壓範圍為 $\pm 5mV$ 並以 11 位元的解析度表示。另外為了節省儲存空間 MIT-BIH 使用自定義的格式儲存資料，利用三個位元組去儲存兩筆心電圖訊號，也就是每筆心電圖訊號有 12 位元的儲存空間，但事實上只有用到 11 位元。所以我們在使用上必須先做前處理以得到我們所需要的輸入格式。圖 4.1.1 為從 MIT-BIH 讀取出來的心電圖訊號。

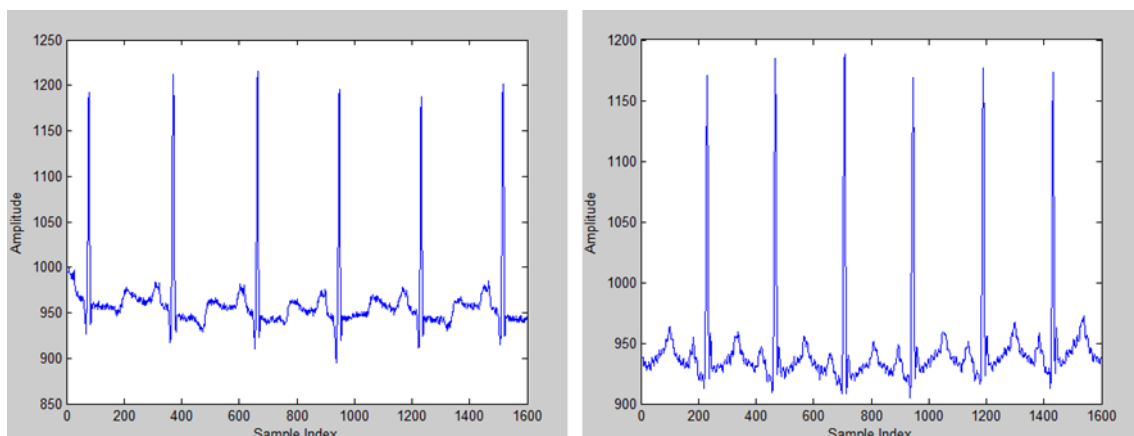


圖 4.1.1 MIT-BIH 資料庫心電圖訊號

另外 MIT-BIH 資料庫還提供說明檔(Annotation File)，檔案提供了 R-R 間距的距離與心跳所屬的類型，每一個心跳分類的註解都至少經過兩名心臟科醫生互相比對，這些註解對於我們系統的驗證有很直接的幫助。如圖 4.1.2 所示，每個心跳上方都標示了醫生的註解。

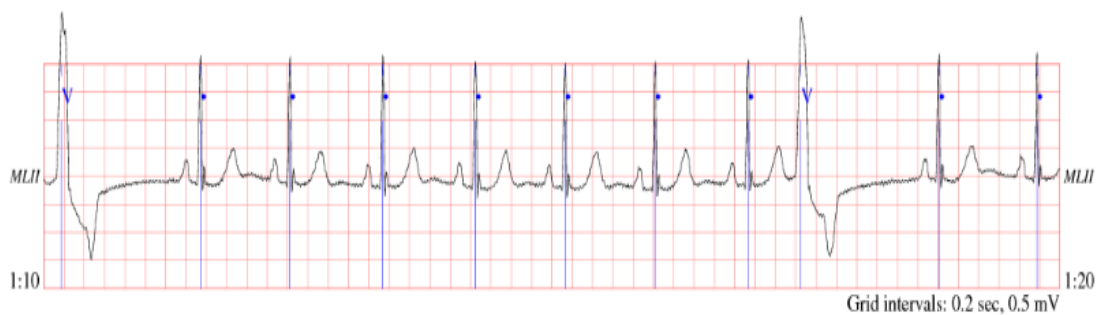


圖 4.1.2 MIT-BIH 心電圖訊號心跳類型註解



4.2 實驗結果

本節主要是呈現系統中的幾個重要功能的實驗結果，其中功能包括 R 波偵測、心律不整判斷、心電圖訊號壓縮。為了方便呈現實驗結果，我們將系統的輸出結果導入 MATLAB，利用其優越的計算與顯示功能，以圖表的方式加以說明成果。

4.2.1 R 波偵測

由於本研究心律不整演算法最重要的參數之一就是 R-R 區間的距離，所以使得 R 波位置的偵測就顯得相當重要，其準確性完全影響接下來判定心律不整的工作。圖 4.2.1 及圖 4.2.2 為部份 MIT-BIH 心電圖記錄的 R 波偵測實驗結果，在 R 波的位置上特別以圓圈標記。



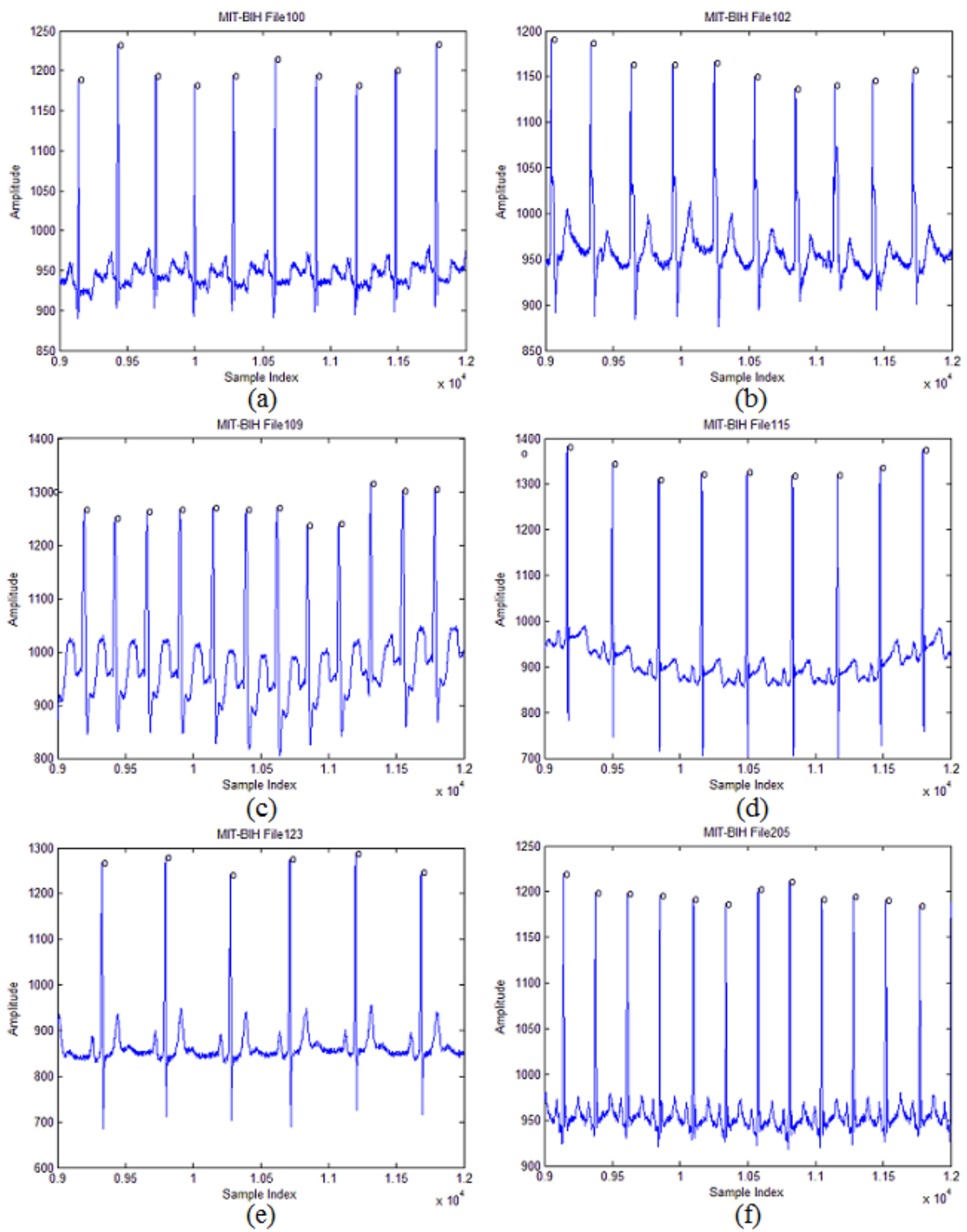


圖 4.2.1 MIT-BIH R 波偵測(a)檔案 100；(b)檔案 102；(c)檔案 109；(d)檔案 115；
(e)檔案 123；(f)檔案 205

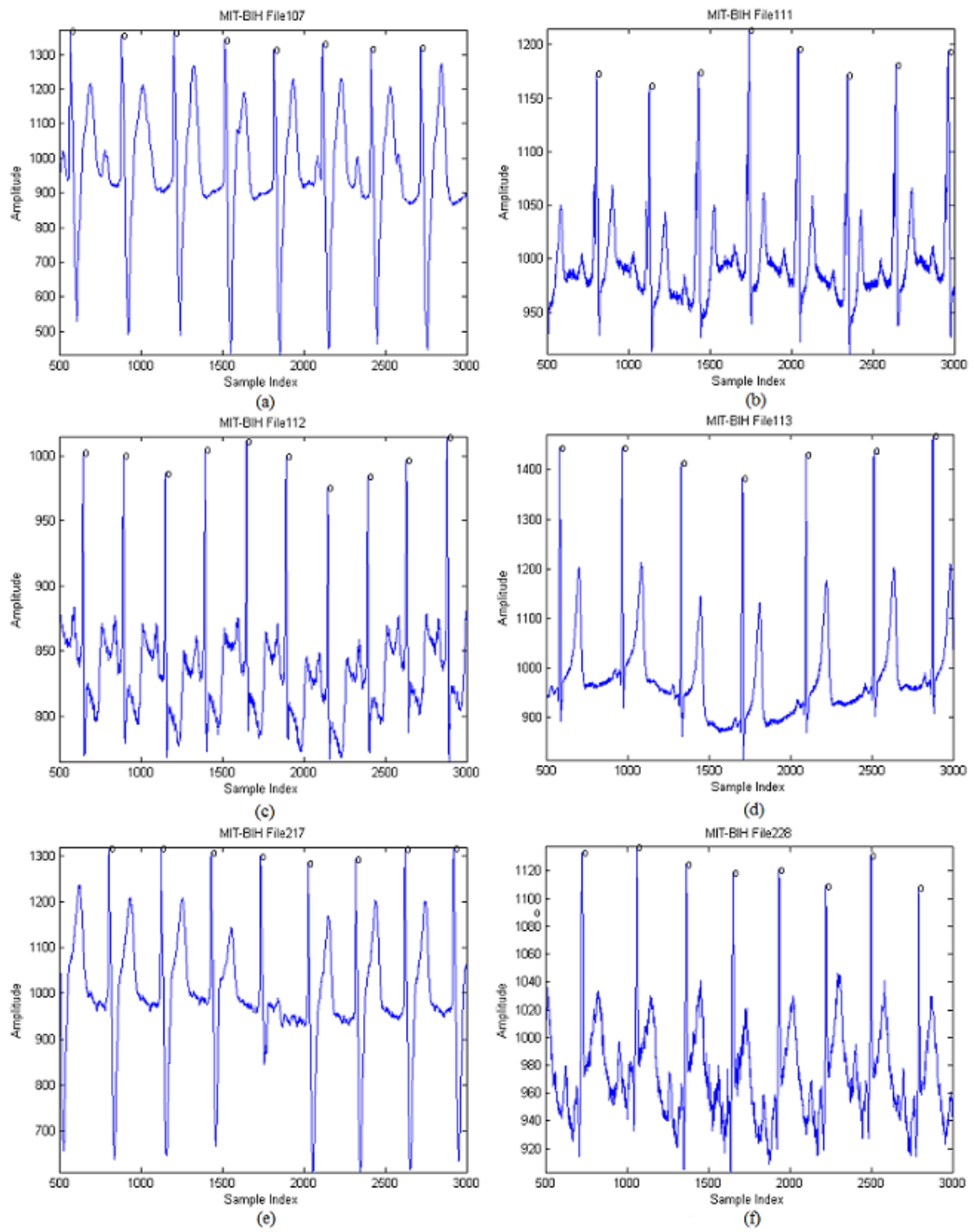


圖 4.2.2 MIT-BIH R 波偵測(a)檔案 107；(b)檔案 111；(c)檔案 112；(d)檔案 113；
(e)檔案 217；(f)檔案 228

在評估心電圖 QRS 波群偵測的能力時，通常以靈敏度(Sensitivity)作為衡量的指標，其公式如 (4.2.1)所示。

$$Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (4.2.1)$$

其中 *True Positive* 為系統判斷為 R 波，心電圖也確實為 R 波；*False Negative* 為系統沒判斷出 R 波，但心電圖為 R 波。

我們總共對系統進行了 20 個完整的 MIT-BIH 心電圖檔案的 R 波偵測實驗，並透過 MIT-BIH 資料庫提供的說明檔檢驗結果以驗證功能性。實驗結果的數據如表 4.2.1，從表中可以看出所有檔案皆達到 98% 以上的靈敏度，以此我們可以準確計算出 R-R 區間的距離，進而做心律不整的判斷與分類。



表 4.2.1 MIT-BIH 心電圖檔案 R 波偵測實驗結果

File	True Positive	False Positive	False Negative	Sensitivity
119	1987	1	2	99.90%
207	2223	0	9	99.60%
100	2270	0	2	99.91%
102	2186	0	0	100.00%
109	2526	1	6	99.76%
123	1517	2	0	100.00%
115	1951	0	1	99.95%
117	1519	15	16	98.96%
205	2641	0	14	99.47%
233	3090	16	5	99.84%
105	2536	0	36	98.60%
107	2099	12	38	98.22%
112	2532	0	7	99.72%
101	1845	0	20	98.93%
215	3363	71	6	99.82%
222	2483	2	0	100.00%
217	2203	14	5	99.77%
103	2047	0	37	98.22%
113	1773	28	22	99.33%
228	2040	0	13	99.37%

4.2.2 心律不整判斷

在心律不整的判斷上，本研究使用的 Tsipouras 演算法可以診斷心律不整的類型。通常使用幾個統計量去評估診斷工具檢驗病症的效能，以下對這幾個統計量進行說明：

- 靈敏度(Sensitivity)：受驗者有心律不整，系統亦診斷出心律不整。(公式 (4.2.2))
- 特異度(Specificity)：受驗者沒有心律不整，系統亦診斷出沒有心律不整。(公式 (4.2.3))
- 陽性預測值(Positive Predictive Value, PPV)：系統診斷為心律不整，受驗者確實有心律不整。(公式 (4.2.4))
- 陰性預測值(Negative Predictive Value, NPV)：系統診斷沒有心律不整，受驗者確實沒有心律不整。(公式 (4.2.5))
- 精確度(Accuracy)：在所有受驗者的受驗結果中，系統診斷正確的比率。(公式 (4.2.6))

$$Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (4.2.2)$$

$$Specificity = \frac{True\ Negative}{True\ Negative + False\ Positive} \quad (4.2.3)$$

$$PPV = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (4.2.4)$$

$$NPV = \frac{True\ Negative}{True\ Negative + False\ Negative} \quad (4.2.5)$$

$$Accuracy = \frac{True\ Positive + True\ Negative}{All\ Samples} \quad (4.2.6)$$

表 4.2.2 說明公式 (4.2.2)~(4.2.6)內參數的意義。

表 4.2.2 實驗結果與 MIT-BIH 檢定狀況對照定義

		Results of MIT-BIH Database	
		Present Disease	Absent Disease
Result of Experience	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

由於 MIT-BIH 資料庫的 200 系列心電圖檔案有較多樣性的心律不整心跳類型 (100 系列缺乏類型三與類型四的心律不整)，為了充份評估系統的心律不整判定功能的效能，所以本研究進行了所有 200 系列心電圖檔案(共計 25 個)的實驗，並將實驗結果的數據進行上述統計公式的分析，其結果如表 4.2.3 所示。

表 4.2.3 心律不整判別效能評估

	MIT-BIH Database Annotation				Positive Predictive Value(%)
	N	PVC	VF	BII	
N	39124	576	2	1	99.52
PVC	835	4160	8	36	82.56
VF	26	74	470	1	82.31
BII	257	30	4	382	38.19
Sensitivity(%)	97.22	85.95	97.11	90.95	Accuracy
Specificity(%)	89.92	97.86	99.78	99.36	96.17%

可以觀察到雖然幾乎所有統計量皆呈現一定的準確度，但在類型四心律不整(BII)的陽性預測值僅有 38.19%的準確率；另外本系統的定位是以輔助診斷並降低醫護端的負擔為目的，所以如上一章所展示的系統機制所示，我們只需要輔助判別是否為心律不整，對於心律不整的類型仍必須交予醫護人員作進一步的診斷。所以我們將心律不整的效能評估重新再作整理如表 4.2.4。

表 4.2.4 心律不整判別效能評估

	Database Annotation		Positive Predictive Value(%)
	Normal	Arrhythmia	
Normal	39124	579	98.54
Arrhythmia	1118	5165	82.21
Sensitivity(%)	89.92		Accuracy
Specificity(%)	97.22		96.17%

從表中可以觀察到各個統計量皆有一定的準確度，由於心律不整的判定與系統的模式轉換有關，更進一步的考慮也會影響到醫護人員的負擔程度，所以心律不整的準確性相當重要。

4.2.3 心電圖訊號壓縮

本節將呈現系統壓縮功能的效率評估。通常在評估資料壓縮的效率時，以兩個指標作為參考標準，分別是方根誤差百分比(Percent Root-mean-square Difference, PRD)與壓縮位元率(Compressed Rate, CR)。PRD 用來評估壓縮後經重建訊號與原始訊號的誤差，CR 則是用來評估壓縮的倍率。兩個評估指標的公式如(4.2.7)及(4.2.8)所示。

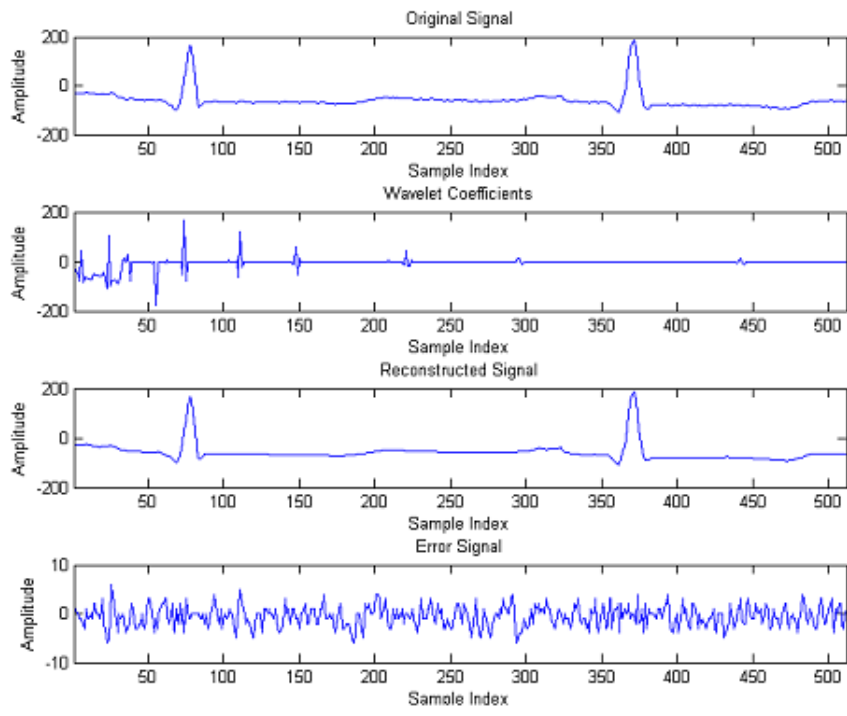
$$PRD = \sqrt{\frac{\sum_1^{n_x} (x_i - \bar{x}_i)^2}{\sum_1^{n_x} (x_i)^2}} \quad (4.2.7)$$

$$CR = n_x b_x / n_s b_s \quad (4.2.8)$$

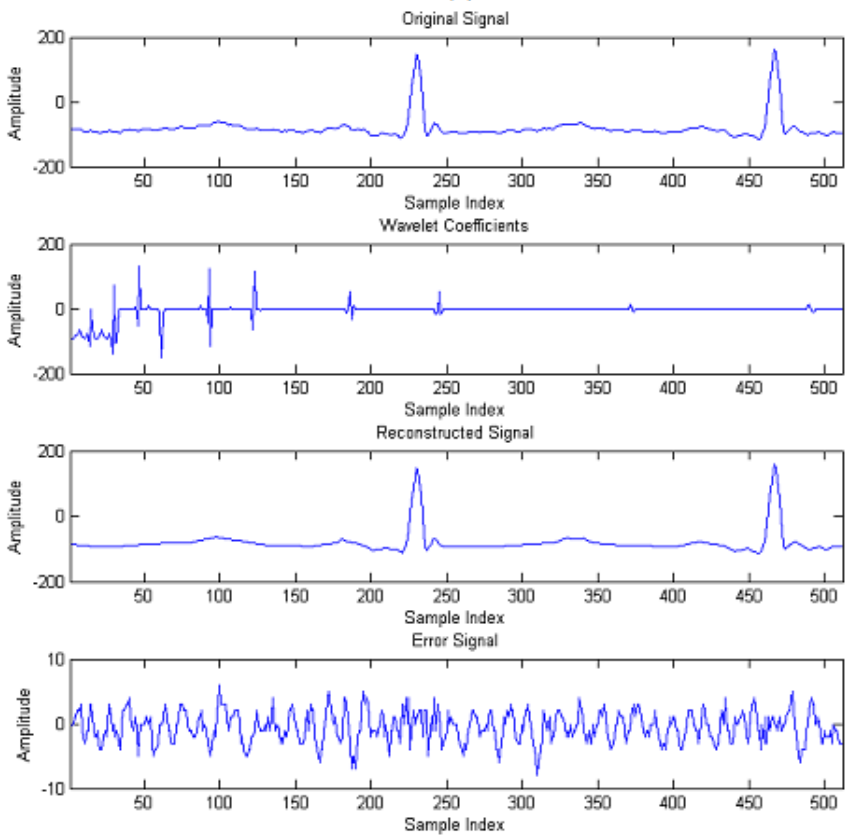
其中 x_i 和 \bar{x}_i 代表原始訊號與重建訊號， n_x 和 n_s 代表原始訊號長度與壓縮訊號長度， b_x 和 b_s 代表原始訊號的表示位元長度與壓縮訊號的表示位元長度。

實際上，本系統在表示壓縮訊號時僅需 9 位元的長度，但為了評估實際傳送狀況時的效能，我們仍將每筆訊號用 12 位元的長度表示(n_s)，即每三個位元組可以表示兩筆訊號，與 MIT-BIT 資料庫的資料表示格式相同。所以在本系統中原始訊號表示位元長度(b_x)與壓縮訊號表示位元長度(b_s)皆為 12 位元。

圖 4.2.3 及圖 4.2.4 所示由上而下分別是：原始訊號、原始訊號經小波轉換並通過閾值篩選後的小波係數、重建訊號與誤差訊號。



(a)



(b)

圖 4.2.3 MIT-BIH 心電圖壓縮相關訊號(a)檔案 100；(b)檔案 205

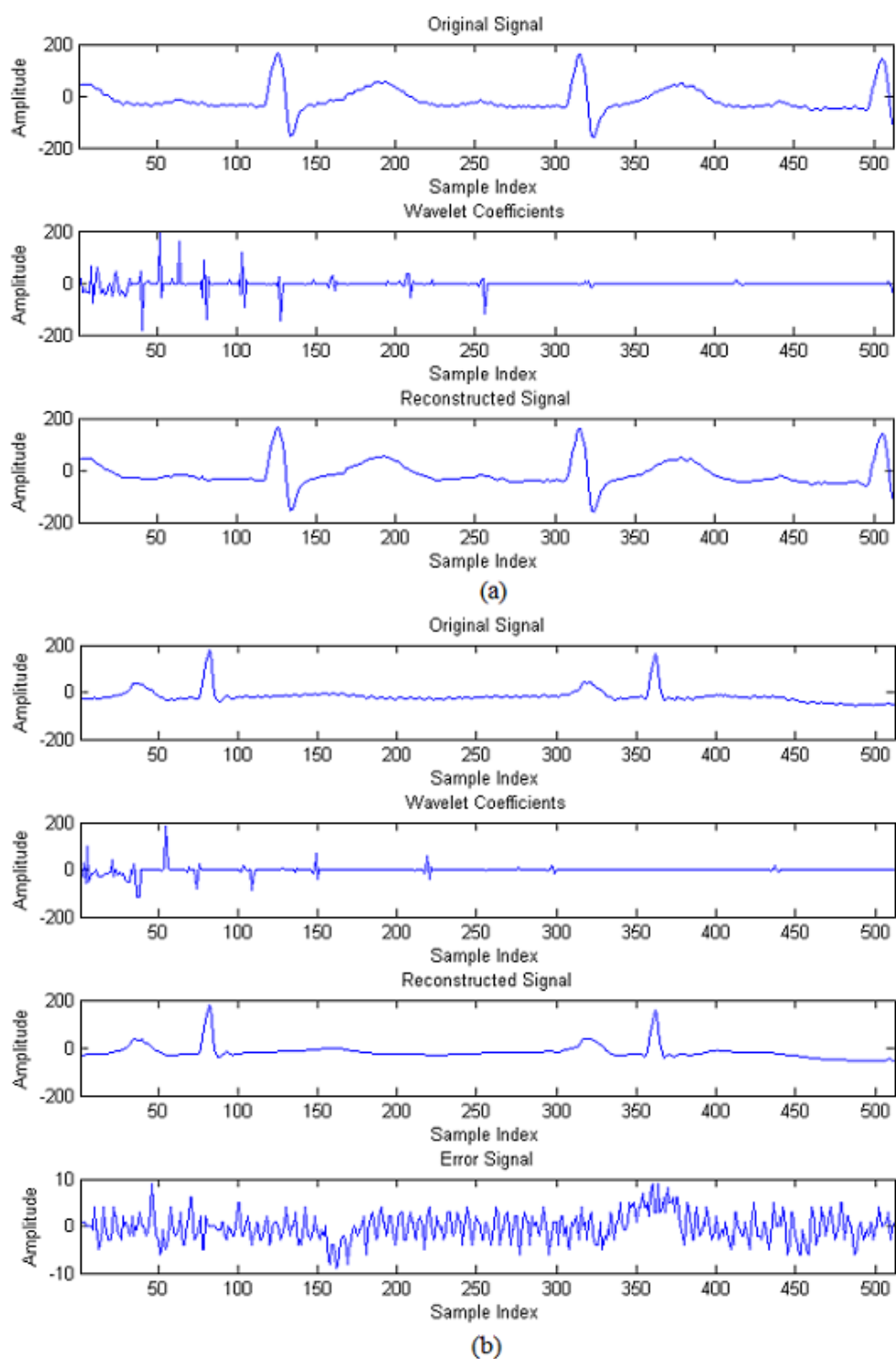


圖 4.2.4 MIT-BIH 心電圖壓縮相關訊號(a)檔案 215；(b)檔案 222

我們進行了和 R 波偵測相同的二十筆 MIT-BIH 心電圖檔案的實驗，結果如表 4.2.5 所示。所有檔案的實驗結果顯示，平均 PRD 與 CR 分別為 3.955 與 7.865，且各別單一檔案的 PRD 與 CR 也都維持一定的比例關係，也就是說壓縮品質是呈現很穩定的狀態。

表 4.2.5 壓縮效能評估

	PRD(%)	CR
100	3.04	7.31
102	4.48	6.74
109	2.81	5.07
115	3.26	10.89
117	4.91	3.74
119	2.19	9.14
123	2.02	10.45
205	2.65	8
207	2.44	7.42
233	4.06	9.31
105	2.47	8.24
107	3.11	9.47
112	3.89	9.14
101	3.25	10.33
215	4.12	8.76
222	2.87	9.24
217	2.02	8.30
103	2.57	7.57
113	3.65	10.34
228	4.87	8.42
Average	3.955	7.865

4.2.4 系統效能量測

在相同的設計架構下，本研究為了比較傳統設計與我們提出的設計的差異性，我們利用 Xilinx 公司所開發的 Virtex-6 晶片作為實驗平台，實際量測了兩組設計幾個較為重要的參數，分別為使用資源、處理時間與功率消耗，以下先對我們所要量測的參數進行說明：

- 使用資源(Resource Usage)

由於兩種設計的硬體架構並不相同，如表 4.2.6，所以在資源的使用程度上當然也就有所不同。在傳統設計上，除了直接在晶片規劃監測模組與壓縮模組外，我們尚需加入的功能模組只剩下 I/O 介面 UART，也就是說我們只需自行建立出 UART(Hardware UART)功能模組於系統內，就可以避免使用到較為複雜的微處理器 MicroBlaze 嵌入式系統的設計，並完成整個系統的設計。

表 4.2.6 硬體架構列表

Traditional Design	PR Design
Monitoring IP	Monitoring IP & Compression IP (PR Design)
Compression IP	
Hardware UART	UART Lite(v1.01a)
	MicroBlaze(100MHZ)
	HWICAP
Virtex6 XC6VLX240T	

而本研究所提出的以部份可重置技術為基礎的設計架構，在監測模組與壓縮模組的實體規劃上可以共同放在同一區域(可重置區)，以此節省使用資源。而部份可重置技術的運用規則規定，可重置區必須被規劃成和所有可重置模組中最大使用資源面積之模組的大小相同，以本系統為例，我們必須採用監測模組與壓縮模組中較耗費硬體資源的模組大小，作為可置區面積規劃的依據。兩模組的使用資源如表 4.2.7，其中 LUTs 為查找表(Look Up Table)的使用個數，Flip-Flops 為正反器的使用個數，DSP48E1 為 Virtex-6 特有的數位訊號處理單位。為從表中可以觀察到壓縮模組使用了較多的硬體資源，所以根據部份可重置技術的運用規則，我們必須將可重置區的尺寸規劃成和壓縮模組的大小相同。

表 4.2.7 監測模組與壓縮模組硬體使用資源

	LUTs	Flip-Flops	DSP48E1
Monitoring Module	3438	3213	1
Compression Module	4755	4286	4
Virtex-6 (XC6VLX240T-1FFG1156)	150720	301440	768

由於我們已經必須使用 MicroBlaze 進行模組切換的工作，為了方便設計我們亦選用 Xilinx 的 UART Lite(v1.01a)作為系統的 I/O 介面。然而 MicroBlaze 只負責下模式切換的命令，實際作重新燒錄動作的是 ICAP 介面，這些元件都要被包含進以可重置技術為基礎的設計架構內。

➤ 處理時間(Processing Time)

在速度的評估上，本研究量測了執行時間(Execution Time)與重置時間(Reconfiguration Time)。執行時間又分為正常模式(Normal Mode)與緊急模式(Emergency Mode)兩種模式；在正常模式下，我們量測的是每筆心電圖資料處理所花費的時間，以 MIT-BIT 的取樣頻率 360Hz 為基準，若要達到所謂的即時(Real Time)處理的效率，每筆資料處理的時間必須在 2.78ms(1/360)內完成；在緊急模式下，由於系統一次是處理一個資料區段(512 筆資料)，所以我們量測的是一個資料區段處理的時間，同樣的若要達到所謂的即時處理，每個資料區段必須在 1423.36ms(512*2.78ms)內完成處理。

另外本研究的系統架構係以部份可重置技術為基礎，所以我們尚需獨立量測模組重置時間，以證明在模組重置的過程中不會造成問題，例如模式切換耗費過多時間，以致丟掉過多的資料對使用者的監測可靠度造成影響。

➤ 功率損耗(Power Consumption)

在功率損耗的分析上，我們設定一個使用情境作為功率評估的基礎。首先系統一開機初始狀態為正常模式，經過五分鐘後我們給予一個心律不整的心跳，讓系統進入緊急模式，此時在傳統設計上需將資料流導向壓縮模組，而我們所提出的設計則需先經過模組置換(監測模組至壓縮模組)後，才真正進入緊急模式。我們同樣讓系統在緊急模式運行五分鐘的時間後，終止量測。

利用上述情境我們可以量測到兩組不同設計的功率損耗，並進行比較。

我們進行了上述實驗項目得到結果如表 4.2.8。首先在使用資源上，從表中可以看到我們所提出的設計在每個類型的資源元件使用皆少於傳統設計，且節省了可觀的數量。在系統的處理速度量測上，由於兩組設計運行在不同的模式下其資料處理架構幾乎相同，所以不論是在正常模式或緊急模式的執行時間量測上，數據都呈現非常接近的結果。

表 4.2.8 系統效能量測結果

		Traditional Design	Our Design
Resource Usage	LUTs	8420	6226
	Flip-Flops	7632	5720
	DSP48E	6	4
Execution Time(ms)	Normal Mode	0.472	0.498
	Emergency Mode	327.68	304.06
Reconfiguration Time(ms)	Normal to Emergency	0.944	X
	Emergency to Normal	1.13	
Power Consumption(mW)		63.04	56.93

另外本研究系統之實測數據分別為 0.498ms 及 304.06ms，確實也達到之前所述即時處理的要求 2.78ms 和 1423.36ms。

我們也獨立量測了本研究所提出的以部份可重置技術為基礎的設計，其模組切換的時間，以耗費較多時間的壓縮模組切換至監測模組為例，表中所示切換時間共耗費了 1.13ms，以 MIT-BIT 的取樣頻率 360Hz 為基準，在模組重置期間會有 0.41 筆(0.0013×360)資料沒有被處理到，也就是說大約僅會有一筆的資料沒有聽過系統做心律不整的檢驗，從此結果顯示切換過程對系統的影響甚小。在功率損耗的量測上，本研究提出的系統架構亦優於傳統的設計，主因是本研究利用部份可重置技術分時工作的特性，避免空閒的功能模組持續運行於系統內。



第五章 結論

雖然近年來有很多研究在 FPGA 上實現不同算法的心電圖訊號處理系統，但相較於其他的實現平台仍顯得功能性相對的單調。本研究旨在硬體上(FPGA)實現複合功能的心律不整診斷支援系統，即在 FPGA 平台上同時建立了心律不整的診斷與壓縮的功能模組，並建立一個有效率的使用機制，完整的利用兩個不同功能模組的效能。

本研究利用 MIT-BIH 資料庫所提供的心電圖訊號去驗證系統的功能性，在心律不整的判別上，靈敏度(Sensitivity)與準確度(Accuracy)分別有 89.92%和 96.17%，顯示本研究所建立之系統有足夠的效能，作為輔助醫護端先行診斷的角色；在壓縮效率的評估上，所有檔案平均的方根誤差百分比 PRD 與壓縮位元率 CR 分別是 3.19%與 7.81 bits/sample，且大部份檔案皆維持一定的比例關係，顯示本系統在壓縮訊號的效能上呈現穩定的狀態。綜合上述，本研究所建立的系統具備良好的功能性。

本研究利用部份可重置技術進一步優化系統的設計。我們於 Virtex-6 平台上進行傳統設計與本研究提出之設計的比較，實際量測結果顯示，我們確實有效的降低了使用資源與功率消耗，換句話說也降低了系統的開發成本與延長使用週期，對於醫療系統來說這些改善都是必要且重要的。另外，近年來部份可重置技術雖然被廣泛的討論與研究，但相關的應用卻仍是少數；本研究將部份可重置技術應

用於生理訊號處理系統上，並且證明了其可行性，另外亦可作為後續相關研究之參考。



參考文獻

- [1] W. Jatmiko, P. Mursanto, A. Febrian, M. Fajar, W.T. Anggoro, R.S. Rambe, M.1. Tawakal, Fauzi, F. Jovan, and M. Eka S., “Arrhythmia Classification from Wavelet Feature on FGPA,” 2011 International Symposium on Micro-NanoMechatronics and Human Science (MHS), Nagoya, pp.349-354, Nov. 2011.
- [2] Chia-Hung Lin and Guo-Wei Lin, “FPGA implementation of Fractal Patterns Classifier for Multiple Cardiac Arrhythmias Detection,” Journal of Biomedical Science and Engineering, vol.5, pp.120-132, Mar. 2012.
- [3] M. Cvikl and A. Zemva, “FPGA-oriented HW/SW Implementation of ECG Beat Detection and Classification Algorithm,” Digital Signal Process., vol.20, no.1, pp.238–248, Jan. 2010.
- [4] Nambakhsh, M.S., Tavakoli, V. and Sahba, N., “FPGA-core Defibrillator Using Wavelet-Fuzzy ECG Arrhythmia Classification,” In Proceeding of 30th Annual International IEEE EMBS Conference, Vancouver, pp.2673-2676, Aug. 2008.
- [5] Rigler, S., Bishop, W., and Kennings, A., “FPGA-based Lossless Data Compression Using Huffman and LZ77 Algorithms,” In Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering, pp. 1235–1238, 2007.

- [6] M. Meira, J. Lima, and L. Batista, "An FPGA Impementation of A Lossless Electrocardiogram Compressor Based on Prediction and Golomb-Rice Coding," In Proceedings of the V Workshop de Informatica Medica, 2005.
- [7] Dora Mara Ballesteros Larrotta, Diana Marcela Moreno Enciso, and Andres Eduardo Gaona Barrera, "Compression of Biomedical Signals on FPGA by DWT and Run-Length," In Proceeding of IEEE ANDESCON 2010, Bogota, Colombia, pp. 1-5, 2010.
- [8] Y. Yongming, L. Jungang and W. Jianmin, "LADAT Arithmetic Improved and Hardware Implemented for FPGA-based ECG Data Compression," 2nd IEEE Conference on Industrial Electronics and Applications, pp.2230-2234, 2007.
- [9] Yongming Yang, Xiaobo Huang, and Xinghuo Yu, "Real-Time ECG Monitoring System Based on FPGA," 33rd Annual Conference of the IEEE Industrial Electronics Society 2007.
- [10] M. Fons, F. Fons, and E. Canto, "Fingerprint Image Processing Acceleration Through Run-Time Reconfigurable Hardware," IEEE Transaction on Circuits and Systems—II: Express Briefs, vol. 57, no. 12, pp.991-995, Dec. 2010.
- [11] M. Fons, and F. Fons, "Run-time Reconfigurable Hardware Technology Brings Key Advantages in the Design of Automatic Personal Recognition Systems," Xcell Journal ,pp.24-31, 2010.
- [12] Xilinx, Inc., MicroBlaze Processor Reference Guide, Ver.11.1, Mar. 1, 2011.

- [13] Xilinx, Inc., System Generator for DSP Getting Started Guide, Ver.10.1, Mar., 2008.
- [14] Xilinx, Inc., System Generator for DSP Reference Guide, Ver.12.3, Sep. 21, 2010.
- [15] Xilinx, Inc., System Generator for DSP User Guide, Ver.12.3, Sep. 21, 2010.
- [16] Xilinx, Inc., ISE Simulator(ISim) In-Depth Tutorial, Ver.12.3, Sep. 21, 2010.
- [17] Xilinx, Inc., EDK Concepts, Tools, and Techniques -- A Hands-On Guide to Effective Embedded System Design, Sep. 21, 2010.
- [18] Xilinx, Inc., PlanAhead User Guide, Ver12.3, Jul. 6, 2011.
- [19] Xilinx, Inc., LogiCORE IP XPS HWICAP (v5.00a), Jul. 23, 2010.
- [20] Xilinx, Inc., Partial Reconfiguration User Guide, Ver.12.3, May 3, 2010.
- [21] Xilinx, Inc., Partial Reconfiguration of Xilinx FPGAs Using ISE Design Suite, Ver.1.1, Jul. 6, 2011.
- [22] H.H.So and K.L.Chan, "Development of QRS Detection Method for Real Time Ambulatory Cardiac Monitor," Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Chicago, USA, pp. 289-292,1997.
- [23] K. F. Tan, K. L. Chan, and K. Choi, "Detection of the QRS-complex, P Wave and T Wave in Electrocardiogram," Advances in Medical Signal and Information Processing, pp.41-47, 2000.

- [24] J. Pan and W.J. Tompkins, "A Real-Time QRS Detection Algorithm," IEEE Transaction Biomedical Engineering, vol. 32, pp. 230-236, 1985.
- [25] M. G. Tsipouras, D. I. Fotiadis, and D. Sideris, "Arrhythmia Classification Using the RR-interval Duration Signal," in Proceedings Computers in Cardiology, 2002, vol. 29, pp. 485-488.
- [26] M. G. Tsipouras, D. I. Fotiadis, and D. Sideris, "An Arrhythmia Classification System based on the RR-interval Signal," Artificial Intelligence Medicine, vol. 33, pp. 237-250, 2005.
- [27] Hannu Olkkonen, "Discrete Wavelet Transform-Biomedical Applications," 1st ed., Croatia : InTech, Aug. 2011, pp.17-32.
- [28] W. Sweldens, "The Lifting Scheme: A Custom-Design Construction of Biorthogonal Wavelets," Applied and Computational Hardware Analysis, vol.3 , no. 3, pp. 186-200, 1996.
- [29] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet Transforms that Map Integers to Integers," Applied and Computational Hardware Analysis, vol. 5, no. 3, pp. 332-369, Jul. 1998.
- [30] SW Smith. "Digital Signal Processing: A Practical Guide ofr Engineers and Scientistics," Elsevier Science, Newnes, 2003.
- [31] Xilinx, Inc., ML605 Hardware User Guide, Ver.1.7, Jun. 19, 2012.

[32] MIT-BIH arrhythmia database CD-ROM. 3rd ed. Harvard-MIT Division of Health Sciences and Technology, 1997.

