國立交通大學

電機與控制工程學系

碩 士 論 文

以FPGA為基底的即時車道偵測系統

An FPGA-Based Real Time Lane Detection System

研 究 生：林洋霆

指導教授：李祖添　教授

中 華 民 國 九 十 四 年 七 月

誌　　謝

在此由衷的感謝恩師李祖添教授在研究所的悉心指導和鼓勵,從老師身上學得不僅是專業領域的知識，更學習到做學問的方法，使我受益良多。同時感謝瞿忠正在論文研究上所提供寶貴的意見和指導，使我論文更加完善，也感謝有著共同革命情感的實驗室的同學們在學業上的討論和生活上的照顧。

最後僅以本論文獻給我的父母、家人及所有關心我的師長與朋友們，和他們共享這份成果與榮耀。

以 FPGA 為基底的即時車道偵測系統

研究生：林洋霆　　指導教授：李祖添 教授

國立交通大學電機與控制工程學系 碩士班

摘要

在本論文中，我們提出了一個簡單、快速又可靠的方法來偵測即時車道資訊，以及演算法之積體電路實現方式，並提出完整的硬體系統架構。我們以電腦視覺技術，根據道路標線影像特徵來偵測車道資訊。並以 FPGA 晶片作為演算法的實現方式，如此可縮短在晶片設計上的開發時程及費用。我們選用目前最普遍的 PC 系統作為前端影像擷取及應用平台，因此對於用戶端在影像擷取設備選用上或是車道資訊應用上有非常大的彈性。此外藉由 PCI 介面將 FPGA 晶片與 PC 系統連接，對於所開發的 FPGA 晶片將有高度的系統整合能力。總合以上特點，我們所提出的即時車道偵測系統不但在設備價格上相當便宜，在整合上也非常容易，因此相當具有實用性。

An FPGA-Based Real Time Lane Detection System

Student：Yang-Ting Lin　Advisor：Tsu-Tian Lee

Department of Electrical and Control Engineering

National Chiao Tung University

Abstract

In this thesis, we propose a simple, fast, and reliable method to detect a lane information in real time.In addition ,we propose a complete hardware architecture to implement our algorithm.We use computer vision to detect a lane information according to the image of the lane’s mark. Using FPGA for implementation, we can reduce the time for development and the expenses in IC-Design. We choose the most common PC system to be a front platform for image processing and applications, so it is very flexile for various applications. By using PCI interface to link FPGA and PC system, our developed FPGA possesses a high ability of system integration. For here reasons, we proposed lane detection system is not only inexpensive but also very easy to implement, and thus show the applicability of the proposed method.

目錄

圖目錄

表目錄

第一章　緒論

1.1 前言

即時車道偵測是智慧型運輸系統(ITS)裡的一個重要研究領域，它提供重要的道路與標線資訊給後端的車輛自走系統，使車輛自走時能保持在車道中央，並提前讓車輛自走系統獲得道路變化趨勢，以應付前方道路彎曲所需反應時間。亦可提供車道偏移資訊，作為車輛警報系統使用。此外在智慧型運輸系統中另一個重要課題”障礙物偵測”，車道擷取亦是重要的前端處理程序。

現今在道路偵測上存在許多不同的技術，主要都是基於電腦視覺(computer vision)的應用。依據其特徵大致可分成幾類，例如：道路模型(2D、3D)，影像擷取方式(單鏡頭、左右雙鏡頭、上下雙鏡頭)，影像格式(灰階、彩色)，偵測特徵(如：邊緣偵測、亮度特徵、連續特徵)。

Broggi 在[1]中，將道路假設為 2D 平面(Z=0)，選用單鏡頭灰階影像，使用反透視法(Inverse Perspective Mapping)[2]，先將影像轉換到鳥瞰圖方式移除透視效應，再應用標線寬度固定特徵，找出車道。但是反透視法(IPM)需要龐大的運算量，對於即時系統的晶片設計上是非常困難的。

在[5]中，作者將路面模擬成一 B 曲線(B-spline)，將車道方程式以三階多項式表示，再將全部影像作邊緣偵測後找出道路標線。此法因複雜的道路模型引出相當龐大的運算量。在分析靜態影像上或許可行，但在即時系統的硬體設計上，則需要龐大資源或是運算時間，

這對於晶片化的過程是非常不利的。在[9]中，作者率先提出藉由估測興趣區間(ROI)，以縮小搜尋範圍，降低偵測時間的概念。且在有限的範圍內搜尋，亦可降低擷取到錯誤影像特徵之風險。此外在[10]中，作者進一步的將路面模擬成一斜平面。將車道視為二次曲線。利用標線表面亮度較高、與標線寬度固定等特徵偵測車道，並藉由預估標線範圍決定偵測區間，藉以求出車道二次曲線參數。

本文主要架構在[10]的基礎上來發展道路偵測演算法，在影像擷取方面使用單鏡頭 644 X 493 像素、256 灰階影像。在道路模型方面將路面視為三度空間中之一平面(2D＋頃斜角 θ)。並將車道視為斜平面上之三次多項式，參數化為 $x = b_3 \cdot \hat{y}^3 + b_2 \cdot \hat{y}^2 + b_1 \cdot \hat{y} + b_0$ 、 $\hat{y} \equiv (z + y) = (\tan q + 1) \cdot y$ 。在偵測特徵方面加入許多道路標線特徵，如：線寬變化不大、片段連續、表面光滑、相對環境有較高亮度、前後標線亮度變化不大、左右標線引入路寬條件判斷等等，來偵測擷取影像。亦藉由精準的標線落點預估決定偵測興趣區間(ROI)，以縮小偵測範圍增加偵測可靠度。另外在首張影像偵測方面，使用六種預估樣板(泛用、直線，左彎、右彎、小幅偏左、小幅偏右)，使估測曲線更貼近實際曲線，在估測樣版與實際曲線誤差過大造成偵測失敗的問題上獲得大幅改善。

演算法經軟體模擬，驗證其可靠度、再將其最佳化後，晶片化實現於 FPGA 晶片上，滿足處理速度每秒 30 張影像之實際應用要求。軟體模擬結果亦作為晶片運算結果之對照組。

硬體系統採用 ALTERA 公司生產的 PCI Development Board – Stratix EP1S25F1020C5 Device 來執行車道偵測演

算法運算。基於價格與相容性的考量，前端影像擷取採用 PC 準系統架構，影像經 CCD 攝影機擷取後透過影像擷取卡傳入 PC，再藉由 PCI 介面傳入 FPGA 晶片(ALTERA Stratix EP1S25F1020C5 Device)作運算，運算結果在透過 PCI 介面傳回到 PC 準系統，以端供後端 ITS 系統(如警報系統、方向盤煞車控制等)使用。

在晶片上採用 PCI 33MHz-32bit Target 設計 PCI 介面，傳輸最大頻寬為每秒 133M-byte，在使用 Pipeline 方式傳輸下，即時車道偵測系統每秒可處理 697 張影像(每張影像約 1.4ms 處理時間)。若假設 PCI BUS 壅塞率 50%(即頻寬為 66M-byte/sec)，且不使用 Pipeline 方式傳輸下，每秒仍可處理 289 張影像。

1.2 論文架構

本文章節架構如下，首先在第二章中將針對車道偵測系統的理論基礎與偵測流程進行說明，藉由 2.1 節道路模型說明偵測系統如何看待車道問題，2.2 節座標轉換公式明確定義偵測目標，而 2.3 節車道偵測流程則說明偵測特徵及偵測方法。在第三章中主要介紹車道偵測處理器的實現方法，在 3.1 節中說明量化問題，3.2 節中介紹基本運算單元及暫存器規劃，而 3.3 節中介紹對應演算法的運算元件。於第四章中介紹硬體系統架構，從前端影像擷取系統、PC 系統、至執行車道偵測的 FPGA 裝置，完整的系統整合及軟硬體架構之說明。第五章說明實驗環境與結果分析，針對架構完成的車道偵測系統進行實測，分析運算效能與偵測誤差，以驗證本文所提系統架構確為有效且具實用性。最後於第六章中提出整體結論，並探討未來有那些精進和改善空間。

第二章　車道偵測系統

2.1　道路模型

2.1.1　路面模型

本文主要將路面視為三度空間中之一平面(*xyz* domain)，再透過映射公式將其投影到影像平面(*u*–*v* domain)。圖(2.1)、(2.2)為道路模型示意圖。

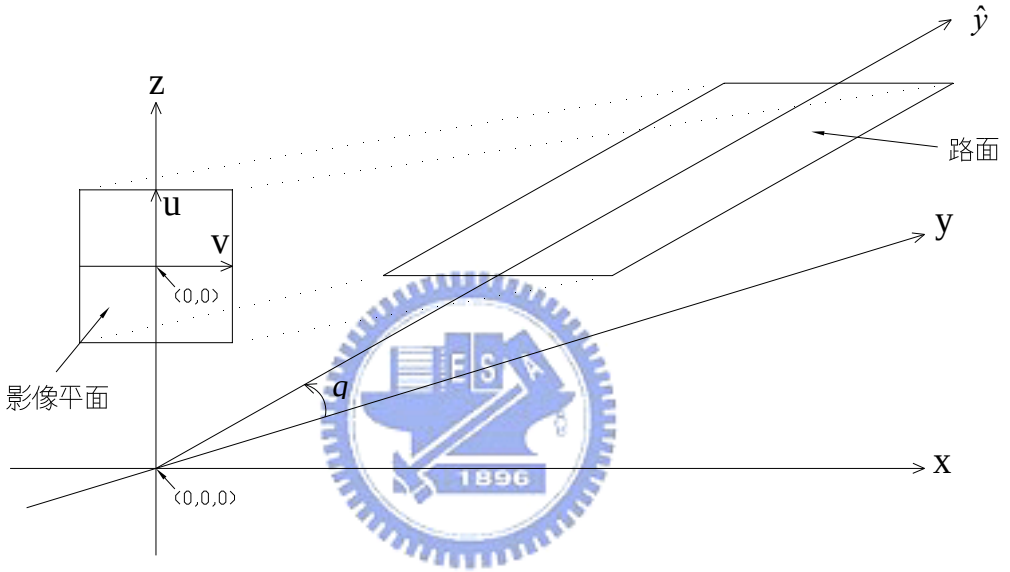


圖 2.1 路面模型圖

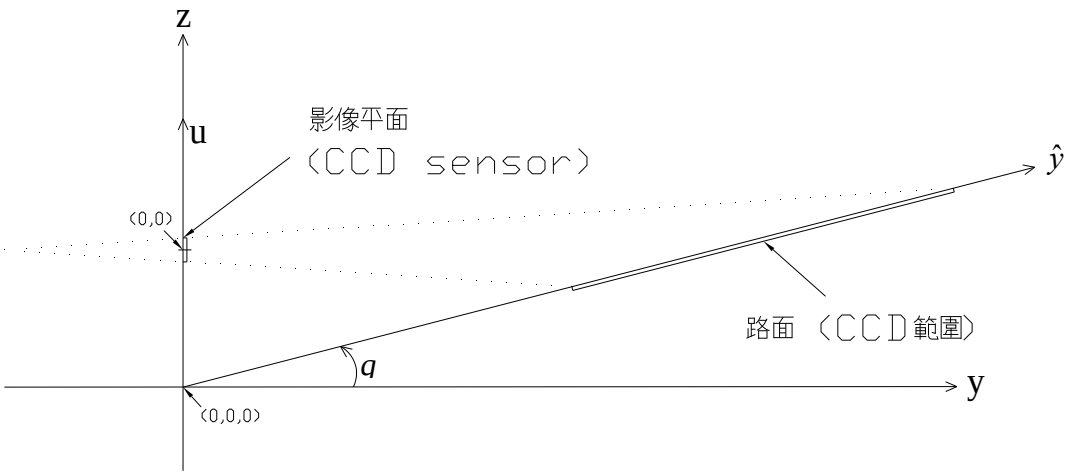


圖 2.2 路面模型剖面圖

2.1.2　車道模型

一般而言，將車道模擬為二次曲線已足夠描述道路九成

以上的特徵，但道路發展常受制於地形地物影響，在複雜地形中車道並非簡單的直線或是二次曲線所能表示。如連續彎路、上下坡彎路，或是環山道路等，二次曲線已無法精確模擬之。故本文更精確的將車道模擬為斜平面上之三次多項式，並參數化為 $x = b_3 \cdot \hat{y}^3 + b_2 \cdot \hat{y}^2 + b_1 \cdot \hat{y} + b_0$、$\hat{y} = (z + y) = (\tan q + 1) \cdot y$，藉由道路偵測演算法偵測標線位置來決定 b_0、b_1、b_2、b_3 參數以獲得道路變化趨勢。在運算複雜度方面，應用最佳化技巧，巧妙的將矩陣運算中運算量與階數 n 的關係，從原本的指數成長降為線性成長，使得在硬體實現上所需資源(硬體空間、計算時間)大幅降低。

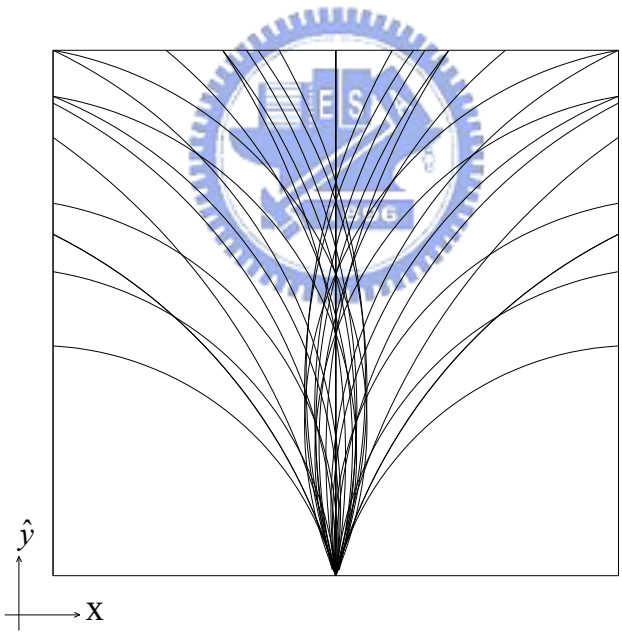


圖 2.3 二次曲線模擬車道中心線圖

2.2 車道偵測演算法

2.2.1 座標轉換關係式

本文將路面模擬為三度空間中之一平面，其中x定義為平行$u-v$平面之方向、y定義為垂直$u-v$平面之方向、z為自地面垂直向上之高度座標、且路面傾斜角為q(如圖 2.1、2.2 所示)。故x、y、z、u、v具有以下轉換關係式：

1.三度空間中道路平面與傾斜角關係式

$$\hat{y} = (z + y) = (\tan q + 1) \cdot y \qquad (2.1)$$

此關係式可將三度空間<x、y、z>與影像空間<u、v>關係式轉變成道路空間<x、$\hat{y}$>與影像空間<u、v>關係(2D 對 2D 映射)，其中傾角q為常數參數。

2.道路空間<x、y>至影像空間<u、v>映射關係式

$$u = e_u \frac{z - H}{y} = e_u \frac{\tan q \cdot y - H}{y} = T_u(y) \qquad (2.2)$$

$$v = e_v \frac{x}{y} \qquad = T_v(x, y) \qquad (2.3)$$

其中e_u、e_v為鏡頭參數，e_u表示像距與一個像素高的比值，e_v表示像距與一個像素寬的比值,正常的光學鏡頭水平與垂直放大倍率為相同($e_u = e_v$)。H表示鏡頭中心(影像空間原點)到地平面($z = 0$)之高度。

3.影像空間<u、v>至三度空間<x、y、z>映射關係式

$$x=\frac{v\cdot H}{e_u\cdot\tan q-u}\frac{e_u}{e_v}=T_X(u,v) \qquad (2.4)$$

$$y=\frac{e_u\cdot H}{e_u\cdot\tan q-u}=T_y(u) \qquad (2.5)$$

$$z=\frac{e_u\cdot\tan q\cdot H}{e_u\cdot\tan q-u}=T_z(u) \qquad (2.6)$$

2.2.2 三次多項式變換關係式

本文中主要將車道模擬為斜平面上之三次多項式 $x=b_3\cdot\hat{y}^3+b_2\cdot\hat{y}^2+b_1\cdot\hat{y}+b_0$、$\hat{y}=(z+y)=(\tan q+1)\cdot y$，透過座標轉換關係式(2.4)、(2.5)，將多項式中x、y變數換成u、v變數，以描述三次多項式在$u-v$平面的關係式如下：

$$v=\frac{b_3\cdot(\tan q+1)^3\cdot e_u{}^2\cdot H^2\cdot e_v}{e_u\cdot\tan q-u}+\frac{b_2\cdot(\tan q+1)^2\cdot e_u\cdot H\cdot e_v}{e_u\cdot\tan q-u}$$
$$+b_1\cdot(\tan q+1)\cdot e_v+b_0\cdot\frac{(e_u\cdot\tan q-u)}{H}\cdot\frac{e_u}{e_v}=f_v(u) \qquad (2.7)$$

若鏡頭水平垂直放大倍率一致($e_u=e_v$)，則上式可簡化如下：

$$v=\frac{b_3\cdot(\tan q+1)^3\cdot e_u{}^3\cdot H^2}{e_u\cdot\tan q-u}+\frac{b_2\cdot(\tan q+1)^2\cdot e_u{}^2\cdot H}{e_u\cdot\tan q-u}$$
$$+b_1\cdot(\tan q+1)\cdot e_u+b_0\cdot\frac{(e_u\cdot\tan q-u)}{H}$$
$$=\frac{B_3}{U^2}+\frac{B_2}{U}+B_1+B_0\cdot U=f_v(U) \qquad (2.8)$$

其中 $B_3 \equiv b_3 \cdot (\tan q + 1)^3 \cdot e_u^{\,3} \cdot H^2$

$B_2 \equiv b_2 \cdot (\tan q + 1)^2 \cdot e_u^{\,2} \cdot H$

$B_1 \equiv b_1 \cdot (\tan q + 1) \cdot e_u$

$B_0 \equiv \dfrac{b_0}{H}$

$U \equiv e_u \cdot \tan q - u$

e_u、H、q、b_0、b_1、b_2、b_3 皆為常數。

2.2.3 寬度轉換關係式

在道路偵測演算法中常需要引入標線寬度 W_M、路面寬度 W 作為特徵判斷條件。其實際寬度(單位：公分)與影像寬度(單位：像素)之間比例關係如下：

$$(v_2 - v_1) = (x_2 - x_1) \cdot \frac{e_u \cdot \tan q - u}{H} \cdot \frac{e_u}{e_v} \qquad (2.9a)$$

若 $e_u = e_v$，且令 $U \equiv e_u \cdot \tan q - u$，則寬度轉換為一比例關係：

$$\Delta v = \Delta x \cdot \frac{U}{H} = T_w(\Delta x) \qquad (2.9b)$$

$$\Delta x = \Delta v \cdot \frac{H}{U} = T_{Iw}(\Delta v) \qquad (2.9c)$$

2.2.4 最小平方近似法估測多項式參數

若已知部分左右標線座標與道路寬度資訊時，將資訊引入座標轉換關係式中，令 W 為道路寬度，u_L、v_L、u_M、v_M、u_R、v_R 分別代表左右道路標線與車道中心線在 u-v 平面之座標，代入(2.1)、(2.2)式可得：

$$u_M = e_u \frac{z_M - H}{y_M} \qquad (2.10)$$

$$v_M = e_v \frac{x_M}{y_M} = \frac{1}{2}(v_R + v_L) \qquad (2.11)$$

$$(v_R - v_L) = e_v \frac{(x_R - x_L)}{y_M} = e_v \frac{W}{y_M} \qquad (2.12)$$

因為左右標線與道路中線在同一橫切面上，故$u_M = u_L = u_R = u$、$y_M = y_R = y_L = y$、$z_M = z_L = z_R = z$。令$e_u = e_v$帶入(2.10)、(2.11)、(2.12)式整理得：

$$x_M = \frac{v_M W}{v_R - v_L} \qquad (2.13)$$

$$y_M = \frac{e_v W}{v_R - v_L} \qquad (2.14)$$

$$z_M = H + \frac{u \cdot W}{v_R - v_L} \qquad (2.15)$$

再將(2.13)、(2.14)、(2.15)關係式代入三次多項式$x = b_3 \cdot \hat{y}^3 + b_2 \cdot \hat{y}^2 + b_1 \cdot \hat{y} + b_0$、$\hat{y} \equiv (\tan q + 1) \cdot y$中，整理可得以下關係式：

$$v_M (v_R - v_L)^2 = b_3 \cdot (\tan q + 1)^3 \cdot e_v^{\,3} \cdot W^2 + b_2 \cdot (\tan q + 1)^2 \cdot e_v^{\,2} \cdot W \cdot (v_R - v_L)$$
$$+ b_1 \cdot (\tan q + 1) \cdot e_v \cdot (v_R - v_L)^2 + \frac{b_0}{W}(v_R - v_L)^3 \qquad (2.16)$$

令 $Coef_3 \equiv b_3 \cdot (\tan q + 1)^3 \cdot e_v^{\,3} \cdot W^2$

$Coef_2 \equiv b_2 \cdot (\tan q + 1)^2 \cdot e_v^{\,2} \cdot W$

$Coef_1 \equiv b_1 \cdot (\tan q + 1) \cdot e_v$

$Coef_0 \equiv \frac{b_0}{W}$

整理(2.16)如下：

$$v_M (v_R - v_L)^2 = Coef_3 + Coef_2 \cdot (v_R - v_L) + Coef_1 \cdot (v_R - v_L)^2 + Coef_0 \cdot (v_R - v_L)^3 \quad (2.17)$$

將得到部分左右標線座標資訊代入(2.17)式，使用最小平方近似法即可求得 $Coef_0$、$Coef_1$、$Coef_2$、$Coef_3$ 參數後，可更進一步求得多項式參數 b_0、b_1、b_2、b_3：

$$b_3 = \frac{Coef_3}{(\tan q + 1)^3 \cdot e_v^{\,3} \cdot W^2} \quad (2.18)$$

$$b_2 = \frac{Coef_2}{(\tan q + 1)^2 \cdot e_v^{\,2} \cdot W} \quad (2.19)$$

$$b_1 = \frac{Coef_1}{(\tan q + 1) \cdot e_v} \quad (2.20)$$

$$b_0 = Coef_0 \cdot W \quad (2.21)$$

2.2.5 最小平方近似法修正道路模型

若已知左右標線座標資訊時，可利用(2.1)、(2.14)、(2.15)修正先前預估的道路模型參數 W、$\tan q$。將(2.1)、(2.14)、(2.15)重新整理如下：

$$(v_R - v_L) = \tan q \cdot \frac{W \cdot (e_v - u)}{H} \quad (2.22a)$$

令 $C_{wm1} \equiv \tan q \cdot e_v \cdot \frac{W}{H}$、$C_{wm0} \equiv -\frac{W}{H}$ 則(2.22a)改寫成：

$$(v_R - v_L) = C_{wm1} + C_{wm0} \cdot u_M \qquad (2.22b)$$

將(2.22)代入最小平方近似法即可求得參數 C_{wm1}、C_{wm0}。之後可更進一步求得道路模型參數 W 及 $\tan q$：

$$W = -C_{wm0} \cdot H \qquad (2.23)$$

$$\tan q = -\frac{C_{wm1}}{C_{wm0} \cdot e_v} \qquad (2.24)$$

2.2.6 預估道路變化趨勢

在獲得足夠左右標線座標資訊後，可藉由三次多項式預估未偵測區域之左右標線落點範圍，縮小偵測區域。首先利用(2.7)式計算出左右標線座標為中心，加上2~4倍標線寬度與斜率補償值作為搜尋範圍(ROI)，若估測點與參考點非常接近時(2 pix以內)，忽略斜率補償值簡化算式。(2.7)式為道路中心線多項式公式，因此在計算左右標線時須先將參數 b_0 替換成 $b_0 \pm \frac{W}{2}$ 將(2.7)式偏移成左右標線多項式公式。

$$\begin{aligned} v_R &= \frac{b_3 \cdot (\tan q + 1)^3 \cdot e_u^{\,3} \cdot H^2}{e_u \cdot \tan q - u} + \frac{b_2 \cdot (\tan q + 1)^2 \cdot e_u^{\,2} \cdot H}{e_u \cdot \tan q - u} \\ &\quad + b_1 \cdot (\tan q + 1) \cdot e_u + (b_0 + \frac{W}{2}) \cdot \frac{(e_u \cdot \tan q - u)}{H} \\ &= \frac{B_3}{U^2} + \frac{B_2}{U} + B_1 + B_{0_R} \cdot U = f_{v_R}(U) \end{aligned} \qquad (2.25)$$

$$v_L = \frac{b_3 \cdot (\tan q + 1)^3 \cdot e_u{}^3 \cdot H^2}{e_u \cdot \tan q - u} + \frac{b_2 \cdot (\tan q + 1)^2 \cdot e_u{}^2 \cdot H}{e_u \cdot \tan q - u}$$
$$+ b_1 \cdot (\tan q + 1) \cdot e_u + (b_0 - \frac{W}{2}) \cdot \frac{(e_u \cdot \tan q - u)}{H}$$
$$= \frac{B_3}{U^2} + \frac{B_2}{U} + B_1 + B_{0_L} \cdot U = f_{v_L}(U) \qquad (2.26)$$

其中 $B_3 \equiv b_3 \cdot (\tan q + 1)^3 \cdot e_u{}^3 \cdot H^2$

$B_2 \equiv b_2 \cdot (\tan q + 1)^2 \cdot e_u{}^2 \cdot H$

$B_1 \equiv b_1 \cdot (\tan q + 1) \cdot e_u$

$B_{0_R} \equiv \frac{b_0 + \frac{W}{2}}{H}$

$B_{0_L} \equiv \frac{b_0 - \frac{W}{2}}{H}$

$U \equiv e_u \cdot \tan q - u$

而 e_u 、 H 、 $\tan q$ 、 b_0 、 b_1 、 b_2 、 b_3 皆為常數。

將(2.25)、(2.26)式微分一次可得車道斜率公式，代入參考點後乘上差值即為斜率補償值。而標線寬度可由(2.9b)獲得。

$$f_{v_R}'(u) = \frac{dv}{du} = \frac{2B_3}{U^3} + \frac{B_2}{U^2} - B_{0_R} \qquad (2.27)$$

$$ROI_R = f_{v_R}(u) \pm [w_{eight} * T_w(\Delta x) + 2 * f_{v_R}'(u_e) * (u - u_e)] \qquad (2.28)$$

$$f_{v_L}'(u) = \frac{dv}{du} = \frac{2B_3}{U^3} + \frac{B_2}{U^2} - B_{0_L} \qquad (2.29)$$

$$ROI_L = f_{v_L}(u) \pm [w_{eight} * T_w(\Delta x) + 2 * f_{v_L}'(u_e) * (u - u_e)] \qquad (2.30)$$

$$ROI = f_v(u) \pm [w_{eight} * T_w(\Delta x)] \qquad (2.31)$$

其中權值 w_{eight} 介於 2~4 之間、u 為估測點、u_e 為參考點。

2.3 車道偵測流程

車道偵測模式分為初始偵測模式與連續偵測模式，若無前張影像資訊時使用初始偵測模式偵測影像，反之則使用連續偵測模式偵測。

2.3.1 初始偵測模式

初始偵測模式流程如圖 2.4 所示，首先執行初始資訊程序，依據影像 u 分割成幾個區塊(zone)，如圖 2.5 所示。由下至上逐區塊搜尋標線資訊(初始模式)，每區塊搜索結束後累計偵測資訊，直至累計超過 15 列標線資訊為止。在初始資訊程序中使用初始參數(首先使用 A 樣板)計算搜尋區間 ROI，程序成功結束後即更新道路模型參數W、W_M、tan$q$及多項式參數b_0、b_1、b_2、b_3後進入連續資訊程序。若初始程序失敗則更換初始 ROI 樣板重新搜尋(共有六種初始 ROI 樣板，A、M、P、Q、R、L 依序更換。如圖 2.6 所示)。

在連續資訊程序中依據初始資訊程序更新後的參數進行 ROI 的計算，搜索方式使用連續模式，在結束每個區間搜尋後馬上更新道路模型參數及多項式參數供下一區間使用。連續資訊程序主要分為兩段進行，第一段自初始資訊程序結束列開始向前搜尋(index 遞減搜索)，第二段主要搜索初始資訊程序未搜索的區間為主(index 遞增搜索)。程序結束後若累計標線資訊已達標準(滿足 45 列單邊或雙邊搜尋成功、且最少 30 列雙邊搜尋成功)則判斷為偵測成功。若判斷為偵測失敗則更換初始 ROI 樣板重新偵測，直至六種初始 ROI 樣板全部偵測結束。

2.3.2 連續偵測模式

連續偵測模式將前張影像偵測結果當成初始參數，跳過初始資訊程序，直接進入連續資訊程序。判斷偵測成功、失敗標準同初始偵測模式。若偵測失敗則重新執行初始偵測模式，重新偵測一次。連續偵測模式影像區塊分割同圖 2.5 所示。連續偵測模式流程如圖 2.7 所示。

圖 2.5 道路偵測區間劃分圖

◆參數初始化程序：

依據初始參數設定道路模型參數：路面寬度 W、標線寬度 W_M、傾角參數 $\tan q$，多項式參數：b_0、b_1、b_2、b_3，以及平均亮度參數 $pre\_brightness$。在初始偵測模式中初始參數為初始 ROI 樣板，在連續偵測模式中初始參數為前一影像偵測結束更新後之參數結果。另外 e_u、H 為固定常數，由影像擷取環境決定之。

(A) 泛用型 ROI

(M) 直線型 ROI

(P) 偏左型 ROI

(Q) 偏右型 ROI

(L) 左彎型 ROI

(R) 右彎型 ROI

圖 2.6 六種初始 ROI 樣板

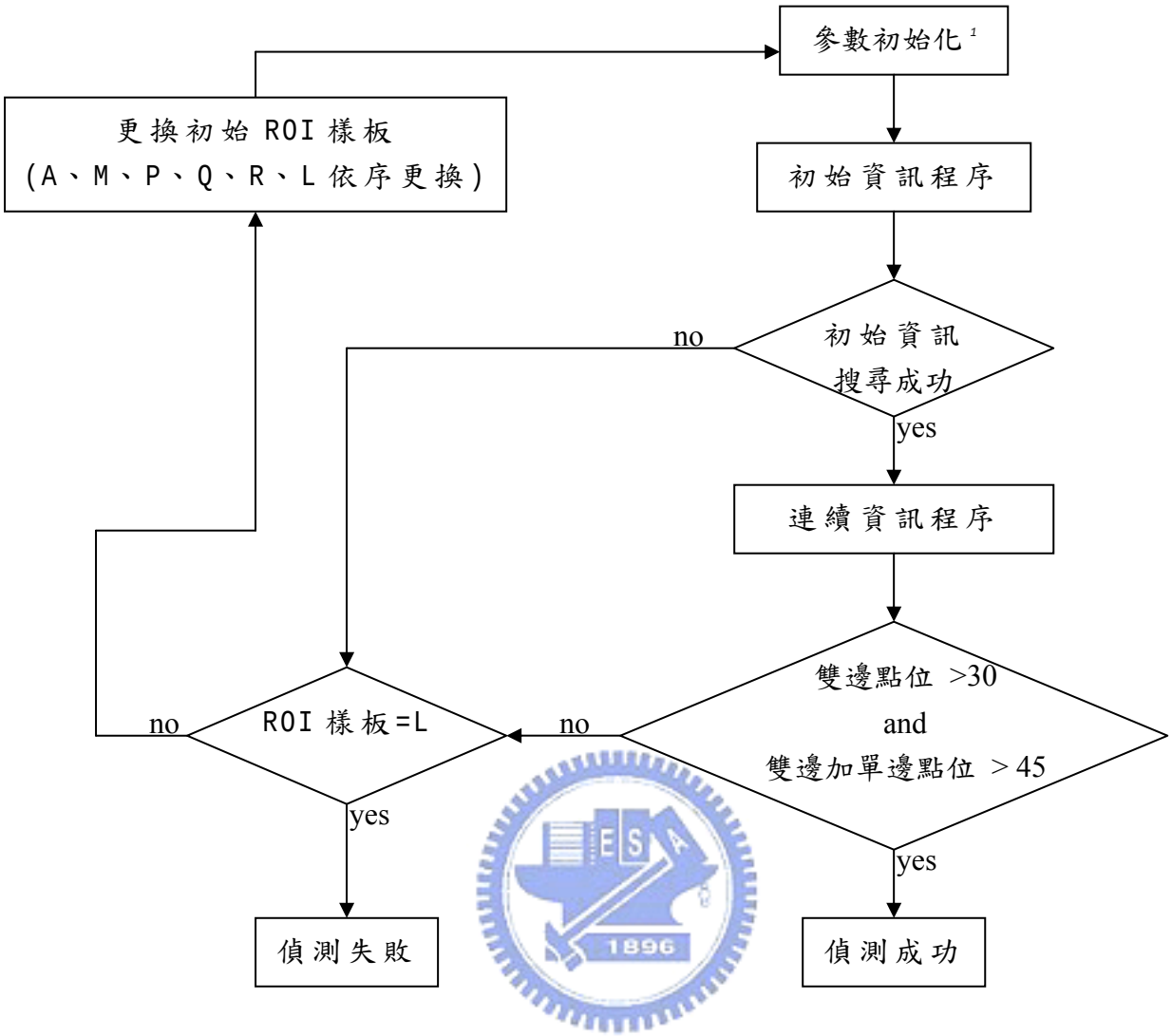


圖 2.4 初始偵測模式流程圖

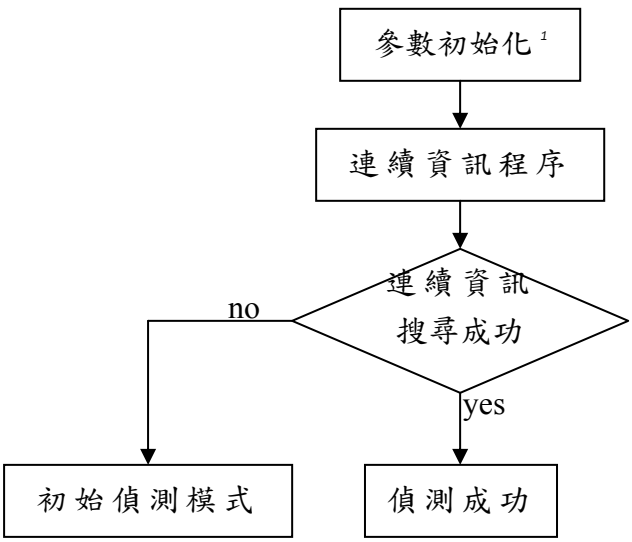


圖 2.7 連續偵測模式流程圖

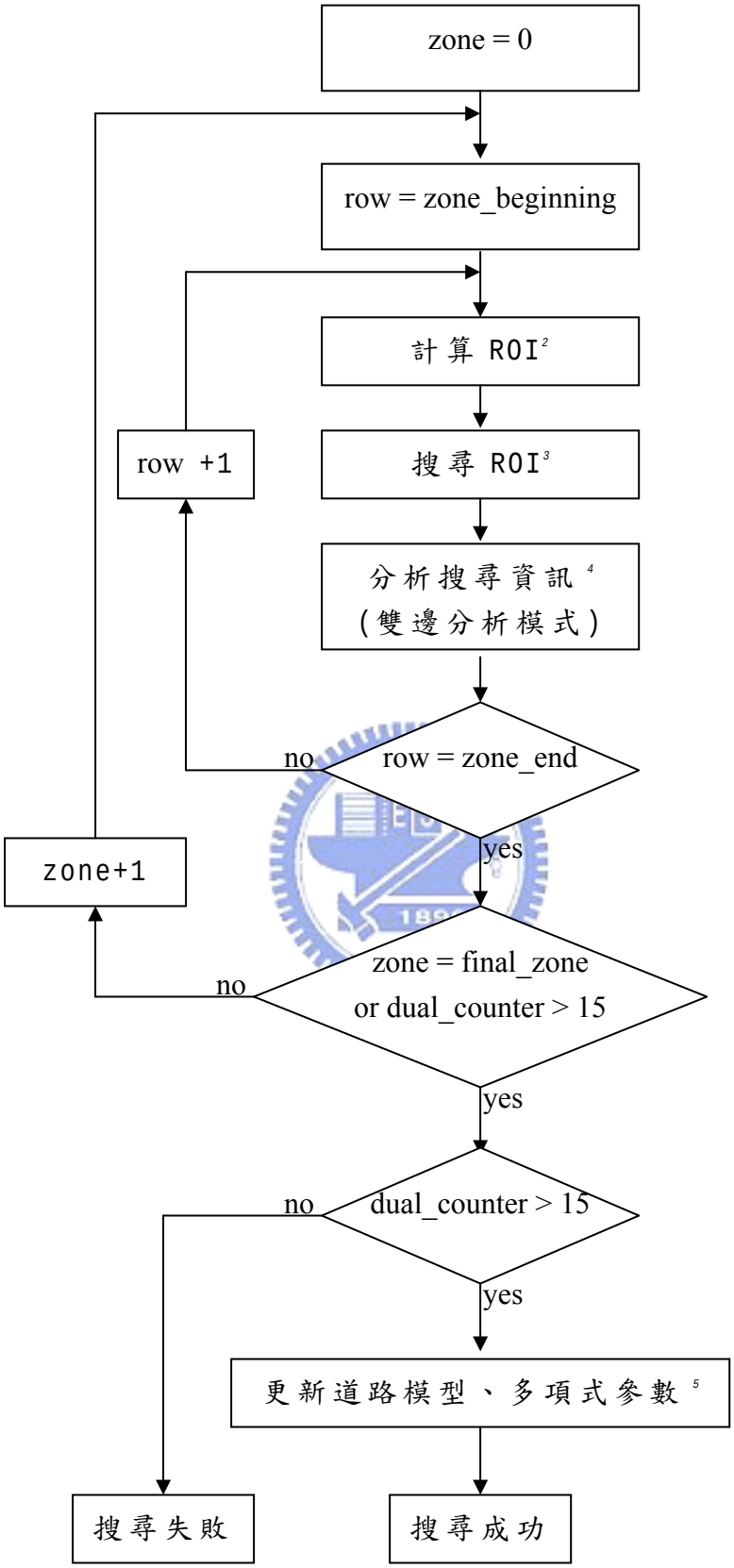
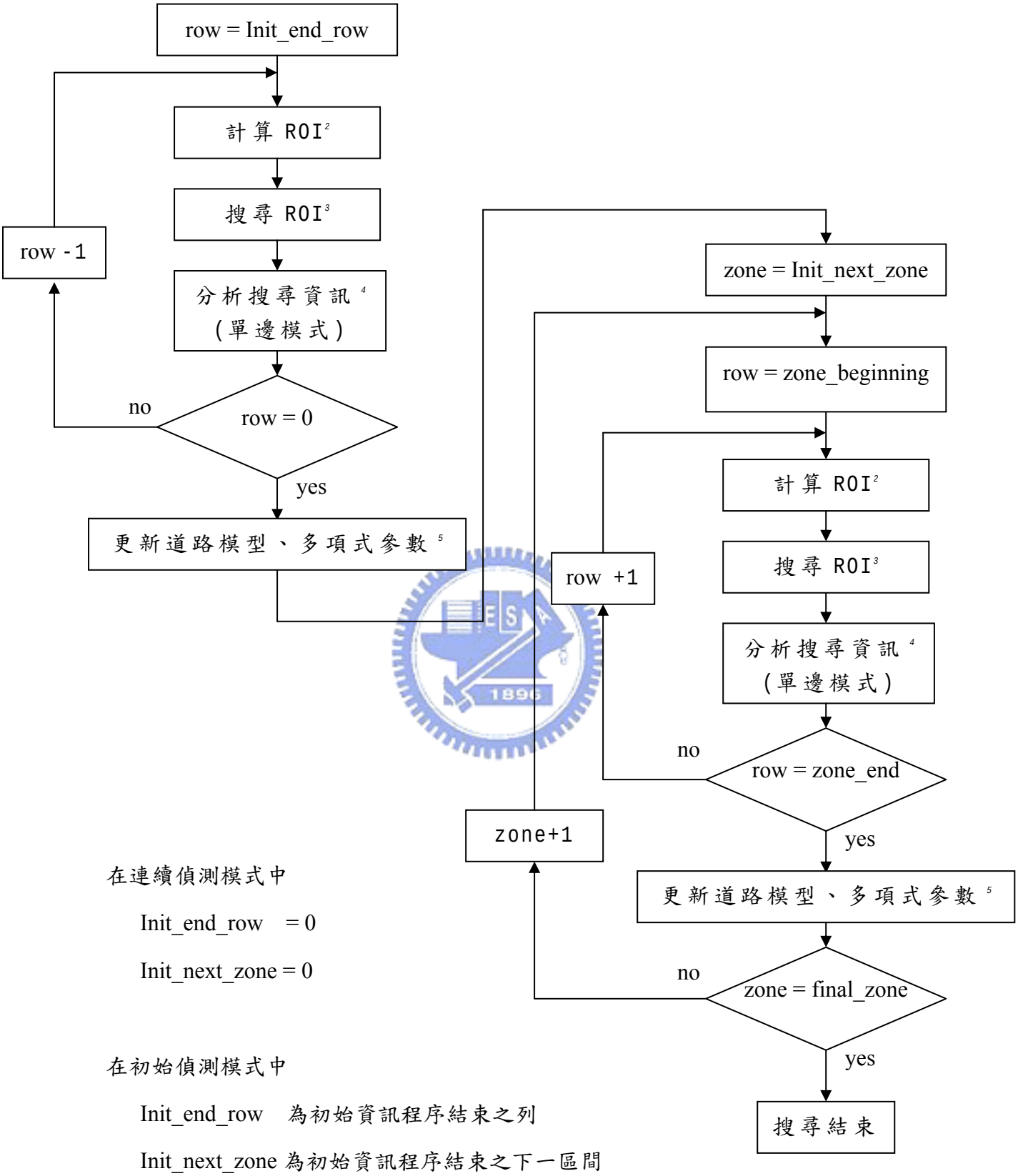


圖 2.8 初始資訊程序流程圖



在連續偵測模式中

Init_end_row = 0

Init_next_zone = 0

在初始偵測模式中

Init_end_row 為初始資訊程序結束之列

Init_next_zone 為初始資訊程序結束之下一區間

圖 2.9 連續資訊程序流程圖

◆ROI 計算程序：

ROI 計算分為三種模式：初始模式、單一模式、連續模式。在初始資訊程序下使用初始模式。在連續資訊程序中前列搜尋成功使用連續模式，反之則為單一模式。

初始模式：使用初始 ROI 樣板相對應之參數。

單一模式：使用(2.28)、(2.30)計算。權值 w_{eight}=2。

連續模式：使用(2.31)式計算。權值 w_{eight}=4。

◆搜尋 ROI 程序：

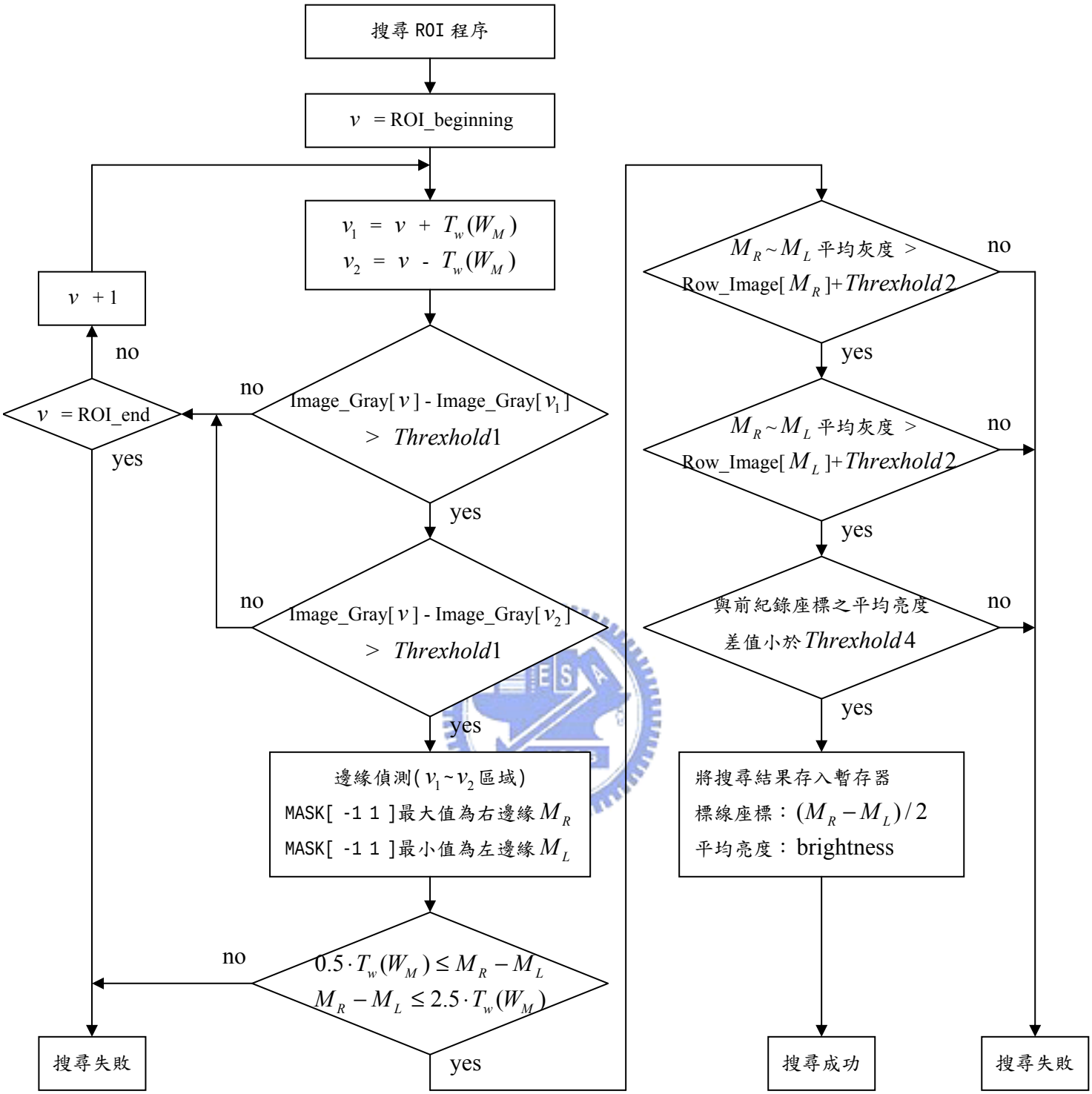
在目前的興趣區間(ROI)內使用邊緣偵測方式(mask[1 －1])搜尋道路標線，由於道路標線與週遭環境亮度具有高反差特徵、標線上亮度相差不大、片段連續、且標線線寬固定。藉由以上特徵搜尋道路標線。流程如圖 2.10 所示。搜尋結果存入暫存區，結束分析程序後，結果為真再則丟入資料堆疊區供更新參數程序使用。

◆分析搜尋資訊程序：

此程序主要功能為判斷在 ROI 內所搜尋到的標線資訊是否為真。搜尋 ROI 分成兩種模式：雙邊模式、單邊模式。在初始資訊程序使用下雙邊模式。在連續資訊程序中使用單邊模式。流程分別如圖 2.11、2.12 所示。

◆更新道路模型、多項式參數程序：

使用(2.22)~(2.24)計算更新道路模型參數 W 、 $\tan q$ ，使用(2.16)~(2.21)計算更新多項式參數 b_0、b_1、b_2、b_3，而 W_M 只有在初始資訊程序完成時更新，使用方式為直接將搜尋到線寬資訊取平均值運算後更新，參數變化如圖 2.13 所示。

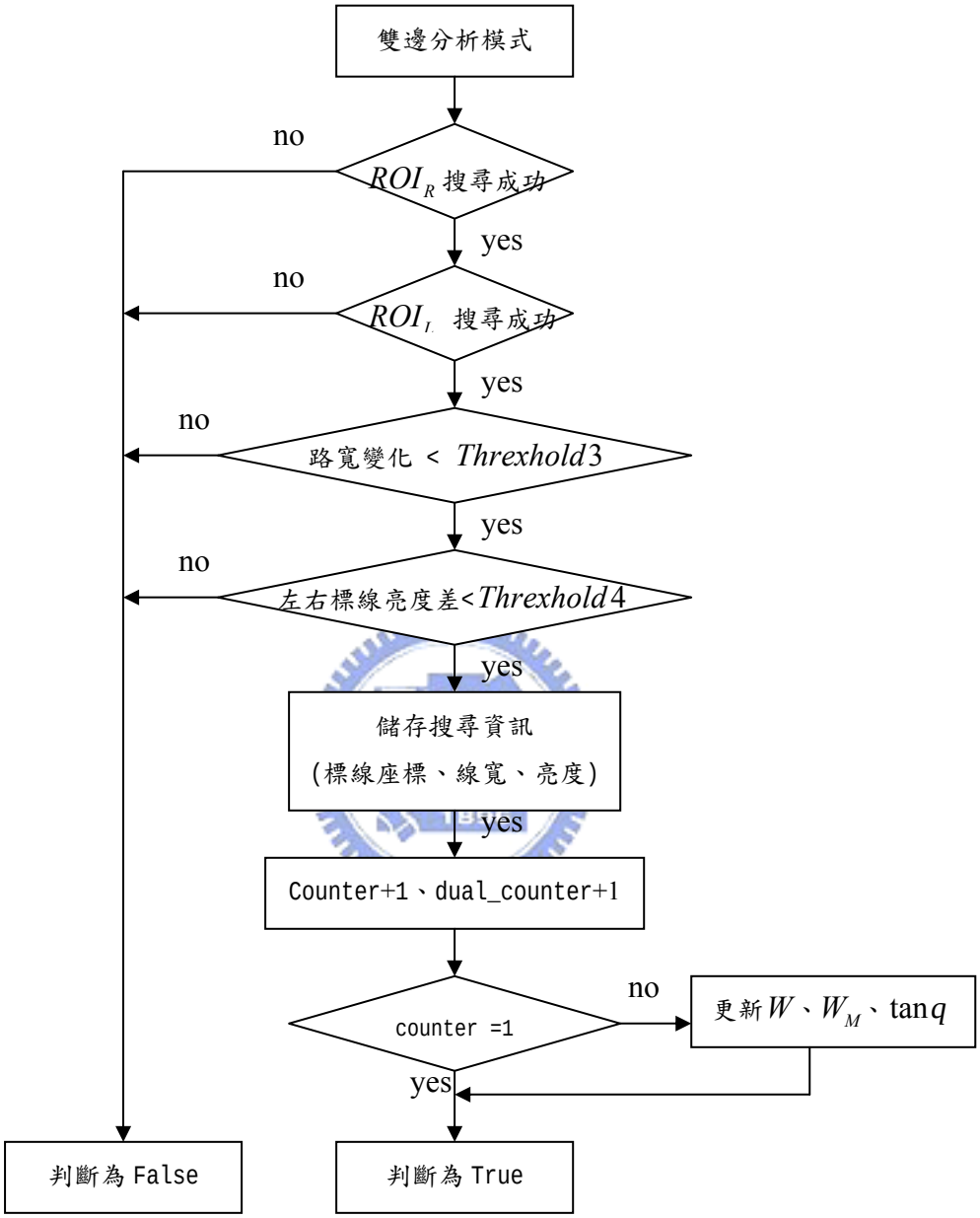


*Threxhold*1 = 20 grey_level (路面與標線灰階差距臨界值)

*Threxhold*2 = 10 grey_level (標線表面灰度變化容許值)

*Threxhold*4 = 40 grey_level (標線平均亮度變化容許值)

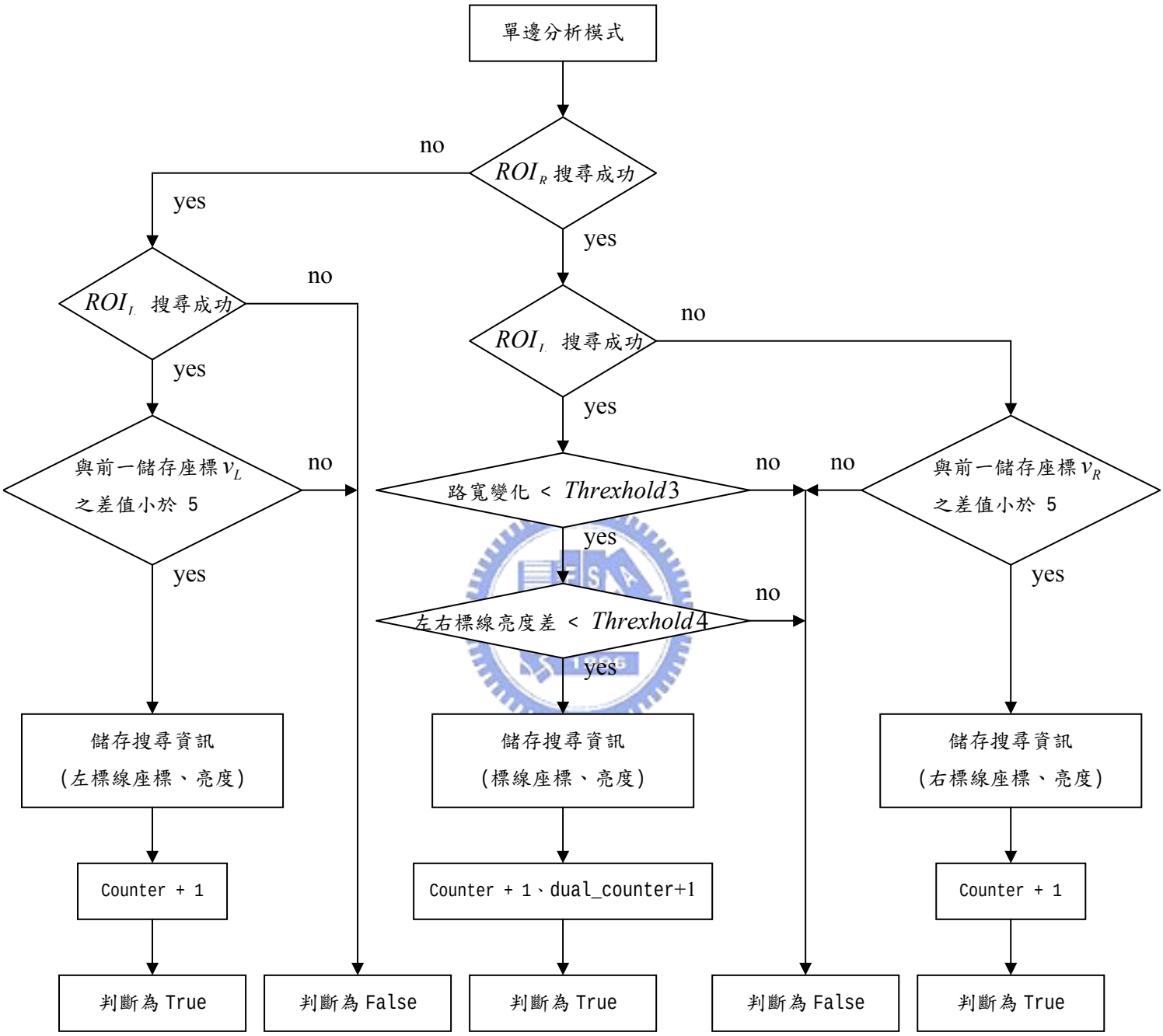
圖 2.10 搜尋 ROI 標線程序流程圖



*Threxhold*3 = 75 cm (路寬變化容許值)

*Threxhold*4 = 40 grey_level (標線平均灰度變化容許值)

圖 2.11 雙邊分析模式流程圖



$Threxhold3 = 75$ cm (路寬變化容許值)

$Threxhold4 = 40$ grey_level (標線平均灰度變化容許值)

圖 2.12 單邊分析模式流程圖

(a) 初始資訊程序結果

(b) 連續資訊程序第一段結果

(c) 連續資訊程序第二段結果

(d) 初始偵測模式結果

(e) 搜尋 ROI 程序示意圖

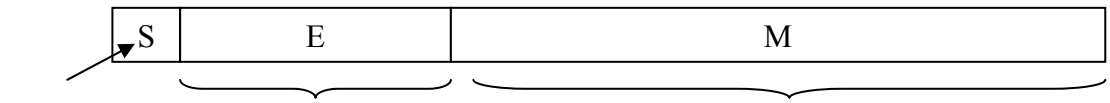
圖 2.13 車道偵測流程示意圖

第三章　車道偵測處理器

3.1　量化與量化誤差

3.1.1　tan θ 量化

路面模型中之傾斜角q，代表車輛前方路面傾斜角度，對於 y-z 平面的剖面圖來說即為路面坡度曲線之切線斜率。在真實的道路上坡度變化皆為漸進式的，切線斜率並不大，一般來說q介於$\pm 15^{o}$之間。在車道偵測演算法中，對於坡度的敏感度並不高，在q的量化中 scale 取0.01^{o}即可。在晶片化的過程中，三角函數($\tan q$)的計算一般使用查表法來實現，每筆資料使用 24 位元浮點數格式儲存，格式如圖 3.1 所示。一般浮點數格式單精度為 32 位元(exponent 8-bit、mantissa 23-bit、sign 1-bit，雙精度為 64 位元(exponent 11-bit、mantissa 52-bit、sign 1-bit)，但道路偵測演算法並不需如此高之精度，以 24 位元浮點數格式即可以滿足。由於$\tan q$為奇函數($\tan(-q) = -(\tan q)$)，所以只需儲存0^{o}~15^{o}間資料即可。儲存資料總共 1500 筆、每筆 24 位元，共需花費 4.4K-Byte 的 ROM 來實現。



Sign bit
1:負數
0:正數

8-bit
指數部分(exponent)

15-bit
數值部分(mantissa)

$$\text{Value} = \begin{cases} = 0\ (underflow)\ for\ E = 0 \\ = \infty\ (overflow)\ for\ E = 255 \\ = \pm 1.M \times 2^{E-127}\ for\ 1 \le E \le 254 \end{cases}$$

圖 3.1 24-bit 浮點數格式

3.1.2 三次多項式

本文中主要將車道模擬為斜平面上之三次多項式 $x = b_3 \cdot \hat{y}^3 + b_2 \cdot \hat{y}^2 + b_1 \cdot \hat{y} + b_0$、$\hat{y} \equiv (z + y) = (\tan q + 1) \cdot y$，其中 b_0、b_1、b_2、b_3、$\tan q$ 為回傳 PC 端之偵測資訊。全部皆以 24 位元浮點數格式儲存之。若以量化後數值代入三次多項式計算，於前方 100 公尺處（$y = 10^4 cm$），在 $b_3 = 10^{-8}$ 情形下（實際彎曲車道 b_3 遠小於 10^{-8}），量化誤差為 $2^{-15} \times 10^{-8} \times (10^4)^3 \approx 3(cm)$。

(2.8)式中 u、v 為整數型態影像座標，以 16-bit 二補數整數格式儲存之。e_u、H 為固定參數，以 24 位元浮點數格式儲存 e_u，以 16-bit 二補數整數格式儲存 H。若以量化後數值 b_0、b_1、b_2、b_3、e_u、H、$\tan q$、u 代入(2.8)式中計算結果 v 之誤差量小於 0.5 pix。因此量化誤差並不影響運算結果，滿足演算法要求。

3.1.3 寬度比例關係式

(2.9)式中 Δv（單位 pix）、Δx（單位 cm）皆為整數型態資料，以 16-bit 二補數整數格式儲存之。以量化後數值 e_u、H、$\tan q$、u、Δx、Δv 代入(2.9b)、(2.9c)中計算結果 Δv、Δx 之誤差量皆小於 0.5 (cm)。因此量化誤差並不影響運算結果，滿足演算法要求。

3.1.4 多項式參數

(2.17)式中 v_R、v_L 為標線搜尋程序中所搜尋到的左右標線中心影像座標，為整數型態(16-bit 二補數整數格式)。而最小平方近似法輸出結果 P_0、P_1、P_2、P_3、Δ，若以 24 位元浮點

數格式儲存，將 $Coef_0 = P_0 / \Delta$、$Coef_1 = P_1 / \Delta$、$Coef_2 = P_2 / \Delta$、$Coef_3 = P_3 / \Delta$ 代入(2.18)、(2.19)、(2.20)中計算 b_3、b_2、b_1、b_0 參數時所產生的誤差量約為實際 b_3 之 1.53×10^{-4} 倍。於前方 100 公尺處($y = 10^4 cm$)，在 $b_3 = 10^{-8}$ 情形下(實際彎曲車道 b_3 遠小於 10^{-8})，量化誤差為 $2^{-15}\times(1+1.53\times10^{-4})\times10^{-8}\times(10^4)^3 \approx 3.05(cm)$。道路偵測演算法運算只多出約 $0.05(cm)$ 的累進誤差。

3.1.5 道路模型

(2.22)式中(v_R , u_M)、(v_L , u_M)為標線搜尋程序中所搜尋到的左右標線中心影像座標，為整數型態(16-bit 二補數整數格式)。而 2 階最小平方近似法輸出結果 C_{wm1}、C_{wm2} 以 24 位元浮點數格式儲存，由(2.23)、(2.24)式計算儲存 W、tan$q$ 參數。W 以整數型態儲存(16-bit 二補數整數格式)、tan$q$ 以 24 位元浮點數格式儲存。運算結果累進誤差皆遠小於 1，因此量化誤差並不影響運算結果。

3.1.6 興趣區間(ROI)

(2.27)~(2.31)中輸出結果 ROI 為標線影像座標範圍預估值，整數型態儲存(16-bit 二補數整數格式)。以量化後數值代入(2.27)~(2.31)式中計算，運算結果之誤差量皆遠小於 1，因此量化誤差並不影響運算結果。

3.2 運算單元與運算暫存器

本文中使用 16 位元二補數加(減)法器、shift-adder 乘法器(常數乘法)、16x16 乘法器(變數乘法)、16÷16 除法器，為運算基本元件，以及衍生出的 24 位元浮點數加(減)法器、24 位元浮點數乘法器、24 位元浮點數除法器，來實現道路偵測演算法中四則運算部分。

(一) 24 位元浮點數乘法器

浮點數格式本身非常容易設計乘(除)法器，但是加(減)法器設計上就比較麻煩(跟二補數格式剛好相反)。24 位元浮點數乘法器結構如圖 3.2 所示，其中 A 為被乘數、B 為乘數。首先 Mantissa 部分先由〝16x16 無號乘法器運算〞(運算前必須先在 Mantissa 最高位元前面補上〝1〞還原 16-bit 原始資訊)，所得乘積部份再經由〝Leading Zeros Detector〞計算乘積部份有多少高位元為〝0〞，之後丟入〝Left Shifter〞作左移運算(浮點數格式：16-bit Mantissa 最高位元必須為〝1〞)。

指數(Exponent)部分直接執行 $E_A + E_B - 127$ 運算後，再減去〝Leading Zeros Detector〞計數的位移值後輸出即為 Exponent，而〝Exponent Corrector〞即是執行此運算之元件。另外 Sign bit 部分直接作 XOR 運算輸出即可。

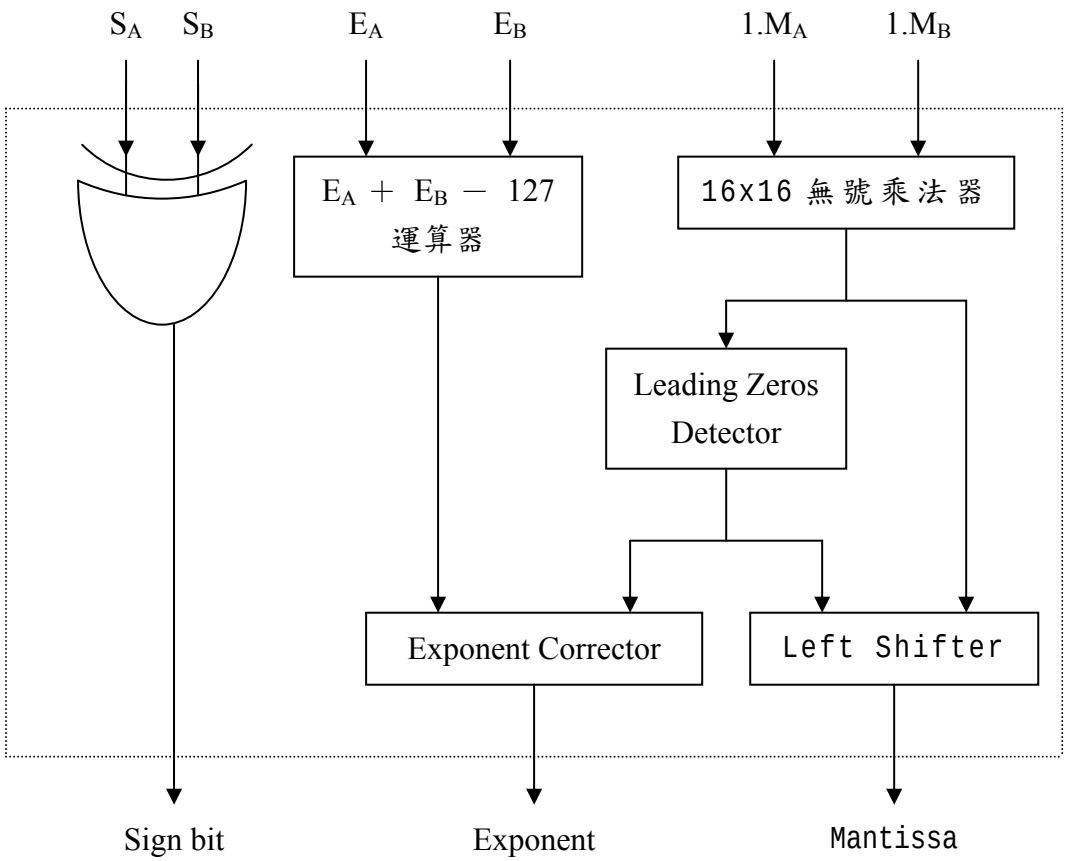


圖 3.2 24 位元浮點數乘法器

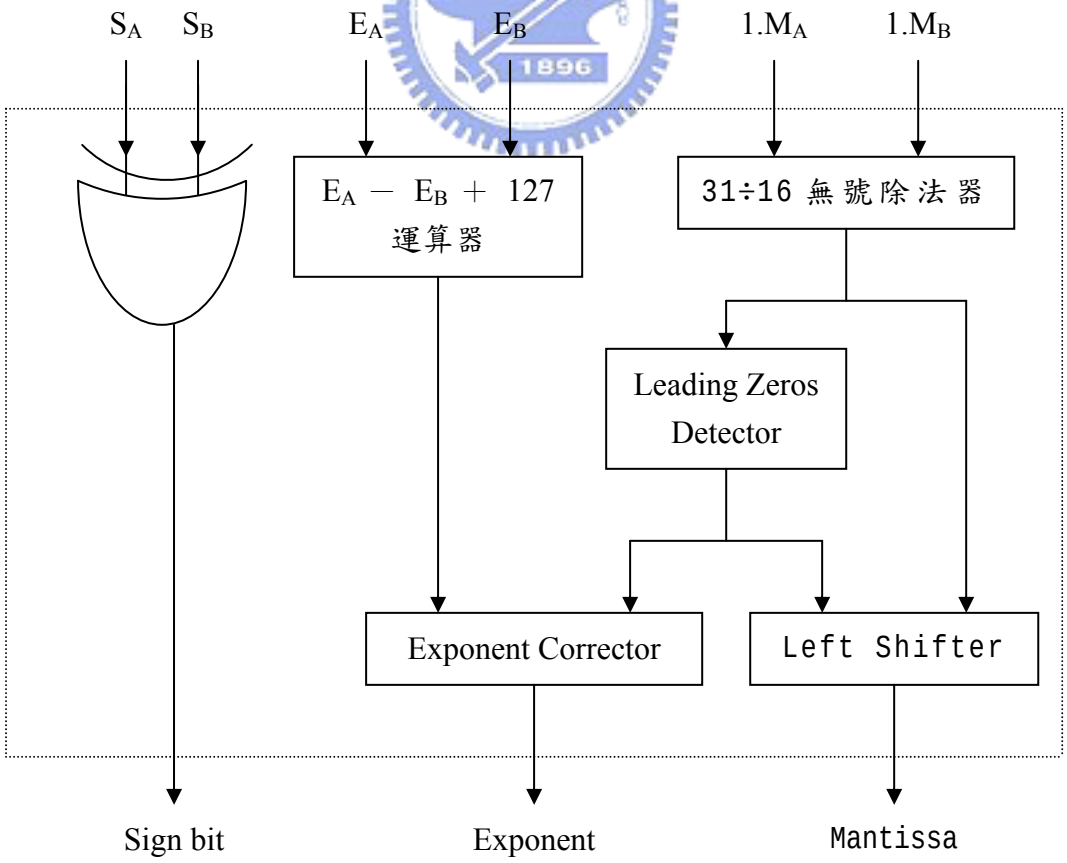


圖 3.3 24 位元浮點數除法器

(二) 24 位元浮點數除法器

24 位元浮點數除法器除了將〝E_A+E_B-127 運算器〞換成〝E_A-E_B+127 運算器〞，〝16x16 無號乘法器〞換成〝31÷16 無號除法器(餘數直接丟棄)〞外，其餘結構同 24 位元浮點數乘法器，如圖 3.3 所示。

(三) 24 位元浮點數加(減)法器

24 位元浮點數加(減)法器結構如圖 3.4 所示。A 為被加(減)數、B 為加(減)數，〝Shift Amount Generator〞執行 8-bit 二補數減法，若 A≦B 則將 $-(E_A-E_B)$結果傳至〝Right Shifter A〞，M_A執行 Shift 運算。若 A>B 則將 E_A-E_B結果傳至〝Right Shifter B〞，M_B執行 Shift 運算。〝Magnitude Generator〞為一比較器，產生的邏輯訊號驅動多工器執行 A±B 或是 B±A 的運算選擇，亦作為〝Sign Generator〞產生〝+/-select〞訊號與 Sign bit 訊號之參考輸入。

執行 16-bit 無號加(減)法運算後經由〝Leading Zeros Detector〞計算和(差)的部份有多少高位元為〝0〞，之後丟入〝Left Shifter〞作左移運算(浮點數格式：16-bit Mantissa 最高位元必須為〝1〞)。而〝Exponent Selector〞為一多工器，若 $E_A>E_B$(即 $E_A-E_B>0$)則將 E_A指定給〝Exponent Corrector〞，反之則指定 E_B。〝Exponent Corrector〞為一個 8-bit 二補數加(減)法器，功能是將〝Leading Zeros Detector〞所計數結果加(減)回 Exponent 後輸出。

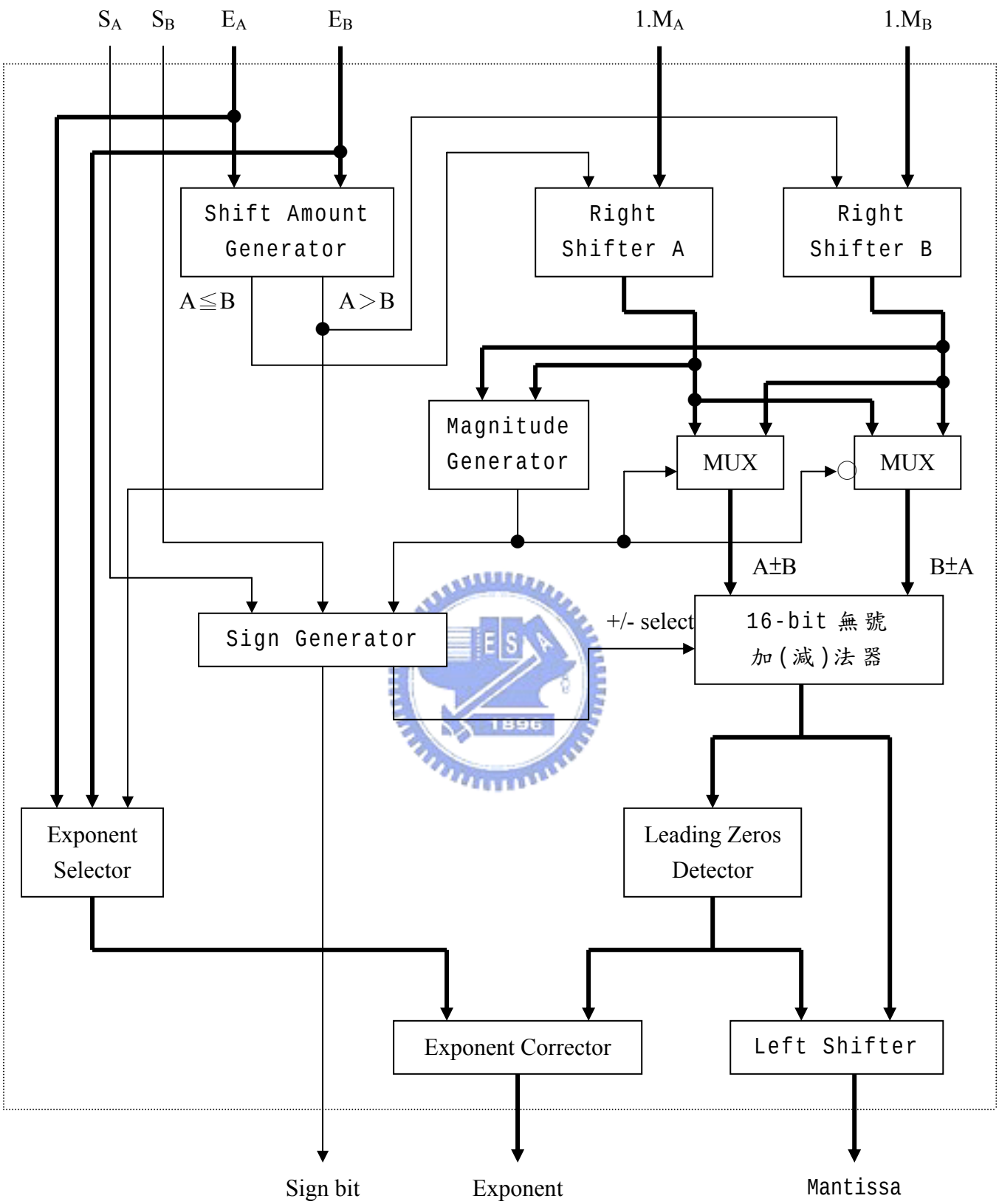


圖 3.4 24 位元浮點數加(減)法器

在本文中工作時脈為 33MHz，此工作頻率對於 24 位元浮點數加(減)法器、乘法器、除法器來說頻率並不高，並不需要特別 retiming 技巧即可在一個 CLOCK 時脈週期中完成運算。若工作時脈需提高到 133MHz，由於浮點數加(減)法器、乘法器、除法器皆屬 free forward cut-set 結構內並無 loop，可以任意加 pipeline 達到頻率要求。

(四) 暫存器規劃

道路偵測演算法中，運算過程中有許多共同項。使用暫存器儲存這些中間項避免發生重複運算，浪費硬體資源。另外固定常數項間的運算，在設計時先行計算，待系統開機後填入暫存器中，以避免浪費硬體時間計算。暫存器更新時機主要分為三種情況分別為系統初始化指定、程序要求時、以及被動式更新(依據指定的暫存器更新而更新)。

1. $\mathrm{Re}g_1 = e_u = e_v = 1081.08$

   e_u、e_v 為固定常數，在系統初始化時指定給 $\mathrm{Re}g_1$(24 位元浮點數格式)。

2. $\mathrm{Re}g_2 = \tan q$

   $\mathrm{Re}g_2$ 在系統初始化時指定(由 q 查表得之)或是在執行〝修正道路模型參數程序〞中產生更新。$\mathrm{Re}g_2$ 使用 24 位元浮點數格式。

3. $\mathrm{Re}g_3 = \mathrm{Re}g_1 \times \mathrm{Re}g_2 = e_u \cdot \tan q$

   $\mathrm{Re}g_3$ 為 24 位元浮點數格式，依據指定之暫存器被動更新。

4. $\mathrm{Re}g_4 = \mathrm{Re}g_2 + 1 = \tan q + 1$

$\mathrm{Re}g_5 = \mathrm{Re}g_4 \times \mathrm{Re}g_4 = (\tan q + 1)^2$

$\mathrm{Re}g_6 = \mathrm{Re}g_5 \times \mathrm{Re}g_4 = (\tan q + 1)^3$

$\mathrm{Re}g_4$、$\mathrm{Re}g_5$、$\mathrm{Re}g_6$皆為 24 位元浮點數格式，且依據指定之暫存器被動更新。

5. $\mathrm{Re}g_7 = \mathrm{Re}g_4 \times \mathrm{Re}g_1 = (\tan q + 1) \cdot e_u$

$\mathrm{Re}g_8 = \mathrm{Re}g_7 \times \mathrm{Re}g_7 = \{(\tan q + 1) \cdot e_u\}^2$

$\mathrm{Re}g_9 = \mathrm{Re}g_8 \times \mathrm{Re}g_7 = \{(\tan q + 1) \cdot e_u\}^3$

$\mathrm{Re}g_7$、$\mathrm{Re}g_8$、$\mathrm{Re}g_9$皆為 24 位元浮點數格式，且依據指定之暫存器被動更新。

6. $\mathrm{Re}g_{10} = H = 125$、 $\mathrm{Re}g_{11} = \frac{1}{H} = 0.008$

H為固定常數，在系統初始化時指定給$\mathrm{Re}g_{10}$、$\mathrm{Re}g_{11}$。$\mathrm{Re}g_{10}$為 16-bit 二補數整數格式、$\mathrm{Re}g_{11}$為 24 位元浮點數格式。

7. $\mathrm{Re}g_{12} = W$、 $\mathrm{Re}g_{13} = 1 \div \mathrm{Re}g_{12} = \frac{1}{W}$、 $\mathrm{Re}g_{14} = \mathrm{Re}g_{12} \div 2 = \frac{W}{2}$

$\mathrm{Re}g_{12}$資訊可由初始系統指定或是在執行〝修正道路模型參數程序〞中產生更新。$\mathrm{Re}g_{13}$、$\mathrm{Re}g_{14}$依據指定之暫存器$\mathrm{Re}g_{12}$改變而更新。$\mathrm{Re}g_{14}$由$\mathrm{Re}g_{12}$右移一位元即可，無需使用除法器。$\mathrm{Re}g_{12}$、$\mathrm{Re}g_{14}$為 16-bit 二補數整數格式、$\mathrm{Re}g_{13}$為 24 位元浮點數格式。

8. $\mathrm{Re}g_{16} = \mathrm{Re}g_8 \times \mathrm{Re}g_{10} = \{(\tan q + 1) \cdot e_u\}^2 \cdot H$

$\mathrm{Re}g_{17} = \mathrm{Re}g_9 \times \mathrm{Re}g_{10} = \{(\tan q + 1) \cdot e_u\}^3 \cdot H^2$

$\mathrm{Re}g_{16}$、$\mathrm{Re}g_{17}$皆為 24 位元浮點數格式，且依據指定之暫存

器被動更新。

9. $\mathrm{Re}g_{19} = \mathrm{Re}g_8 \times \mathrm{Re}g_{12} = \{(\tan q + 1)\cdot e_u\}^2 \cdot W$
$\mathrm{Re}g_{20} = \mathrm{Re}g_9 \times \mathrm{Re}g_{12} = \{(\tan q + 1)\cdot e_u\}^3 \cdot W^2$

$\mathrm{Re}g_{19}$、$\mathrm{Re}g_{20}$皆為 24 位元浮點數格式，且依據指定之暫存器被動更新。

10. $\mathrm{Re}g_{21} = u$ 、 $\mathrm{Re}g_{22} = v$

u、v為座標變數，由程序中 index 決定。$\mathrm{Re}g_{21}$、$\mathrm{Re}g_{22}$皆為 16-bit 二補數整數格式。

11. $\mathrm{Re}g_{23} = \mathrm{Re}g_3 - \mathrm{Re}g_{21} = e_u \cdot \tan q - u \equiv U$
$\mathrm{Re}g_{24} = \mathrm{Re}g_{23} \times \mathrm{Re}g_{23} = U^2$
$\mathrm{Re}g_{25} = \mathrm{Re}g_{23} \times \mathrm{Re}g_{11} = \dfrac{U}{H}$

$\mathrm{Re}g_{23}$、$\mathrm{Re}g_{24}$、$\mathrm{Re}g_{25}$皆為 24 位元浮點數格式，且依據指定之暫存器被動更新。

12. $\mathrm{Re}g_{26} = b_0$ 、 $\mathrm{Re}g_{27} = b_1$ 、 $\mathrm{Re}g_{28} = b_2$ 、 $\mathrm{Re}g_{29} = b_3$

$\mathrm{Re}g_{26}$、$\mathrm{Re}g_{27}$、$\mathrm{Re}g_{28}$、$\mathrm{Re}g_{29}$由初始系統化時指定或在〝更新多項式參數程序〞中產生更新。$\mathrm{Re}g_{26}$~$\mathrm{Re}g_{29}$皆為 24 位元浮點數格式。

13. $\mathrm{Re}g_{30} = \mathrm{Re}g_{26} + \mathrm{Re}g_{14} = b_0 + \dfrac{W}{2}$ 、 $\mathrm{Re}g_{31} = \mathrm{Re}g_{26} - \mathrm{Re}g_{14} = b_0 - \dfrac{W}{2}$

$\mathrm{Re}g_{30}$、$\mathrm{Re}g_{31}$ 皆為 24 位元浮點數格式，且依據指定之暫存器被動更新。

14. $\mathrm{Re}g_{32} = \mathrm{Re}g_{26} \times \mathrm{Re}g_{11} = \frac{b_0}{H} \equiv B_0$

$\mathrm{Re}g_{33} = \mathrm{Re}g_{27} \times \mathrm{Re}g_{7} = b_1 \cdot (\tan q + 1) \cdot e_u \equiv B_1$

$\mathrm{Re}g_{34} = \mathrm{Re}g_{28} \times \mathrm{Re}g_{16} = b_2 \cdot \{(\tan q + 1) \cdot e_u\}^2 \cdot H \equiv B_2$

$\mathrm{Re}g_{35} = \mathrm{Re}g_{29} \times \mathrm{Re}g_{17} = b_3 \cdot \{(\tan q + 1) \cdot e_u\}^3 \cdot H^2 \equiv B_3$

$\mathrm{Re}g_{36} = \mathrm{Re}g_{26} \times \mathrm{Re}g_{11} = \frac{b_0 + \frac{W}{2}}{H} \equiv B_{0_R}$

$\mathrm{Re}g_{37} = \mathrm{Re}g_{26} \times \mathrm{Re}g_{11} = \frac{b_0 - \frac{W}{2}}{H} \equiv B_{0_L}$

$\mathrm{Re}g_{32}$、$\mathrm{Re}g_{33}$、$\mathrm{Re}g_{34}$、$\mathrm{Re}g_{35}$、$\mathrm{Re}g_{36}$、$\mathrm{Re}g_{37}$皆為 24 位元浮點數格式，且依據指定之暫存器被動更新。

15. $\mathrm{Re}g_{38} = \mathrm{Re}g_{24} \times \mathrm{Re}g_{36} = U \cdot B_{0_R}$

$\mathrm{Re}g_{39} = \mathrm{Re}g_{24} \times \mathrm{Re}g_{37} = U \cdot B_{0L}$

$\mathrm{Re}g_{40} = \mathrm{Re}g_{34} \div \mathrm{Re}g_{24} = \frac{B_2}{U}$

$\mathrm{Re}g_{41} = \mathrm{Re}g_{35} \div \mathrm{Re}g_{24} = \frac{B_3}{U^2}$

$\mathrm{Re}g_{38}$、$\mathrm{Re}g_{39}$、$\mathrm{Re}g_{40}$、$\mathrm{Re}g_{41}$皆為 24 位元浮點數格式，且依據指定之暫存器被動更新。

本系統工作頻率為 33MHz，處理需求為每秒 33MHz，此規格對於道路偵測元件而言並不難達成，故晶片上所有設計皆以面積(Chip Area)為優先考量。

上列所有暫存器更新運算所使用之運算單元如加法器、乘法器、除法器皆由 finite state machine 配置，使用完後釋放給其他運算元件使用，以減少運算單元的總量，但此舉必須設置運算單元與暫存器佈線切換之多工器，整體來說還

是有達到節省面積之功用。

另外雖然可透過時程排序法(schedule)達到節省暫存器數量的目的，但此舉將使多工器的設計上更為複雜，但是暫存器本身並不會花費太多資源或晶片面積。因此以 schedule 方法使暫存器數量最佳化後，在節省晶片面積方面並非真正得到的好處。

3.3 道路偵測運算元件

以下所有道路偵測運算元件，如同暫存器更新電路一樣，所使用之運算單元如加法器、乘法器、除法器皆由 finite state machine 配置，使用完後釋放給其他運算元件使用。

(一) 三次多項式 u-v 平面關係式運算元件

將運算暫存器代入(2.8)式中得：

$$v = \frac{B_3}{U^2} + \frac{B_2}{U} + B_1 + B_0 \cdot U = f_v(U)$$
$$= \mathrm{Re}g_{35} \div \mathrm{Re}g_{24} + \mathrm{Re}g_{34} \div \mathrm{Re}g_{23} + \mathrm{Re}g_{33} + \mathrm{Re}g_{32} \times \mathrm{Re}g_{23}$$
$$= [\mathrm{Re}g_{35} \div \mathrm{Re}g_{24} + \mathrm{Re}g_{32} \times \mathrm{Re}g_{23}] + [\mathrm{Re}g_{34} \div \mathrm{Re}g_{23} + \mathrm{Re}g_{33}] \quad (3.1)$$

(3.1)式運算結構如圖 3.5 所示，總共需要 2 次除法、1 次乘法、3 次加法，所有〝＋〞、〝×〞、〝÷〞運算皆在一個 clock 內計算完成。執行 $u-v$ 平面關係式運算元件時最少需使用 1 個浮點數除法器、1 個浮點數乘法器、1 個浮點數加法器，花費 5 個 clock 週期完成運算。

(二) 寬度轉換關係式運算元件

將運算暫存器代入公式 2.9b、2.9c 中得：

$$\Delta v = \Delta x \cdot \frac{U}{H} = T_w(\Delta x)$$
$$= \Delta x \times \mathrm{Re}g_{25} \quad (3.2)$$

$$\Delta x = \Delta v \cdot \frac{H}{U} = T_{hw}(\Delta v)$$
$$= \Delta v \div \mathrm{Re}\, g_{25} \qquad (3.3)$$

(3.2)、(3.3)式運算結構如圖 3.6 所示。執行寬度轉換關係式運算元件時需使用 1 個浮點數除法器(3.3)或是 1 個浮點數乘法器(3.2)，需花費 1 個 clock 週期完成運算。

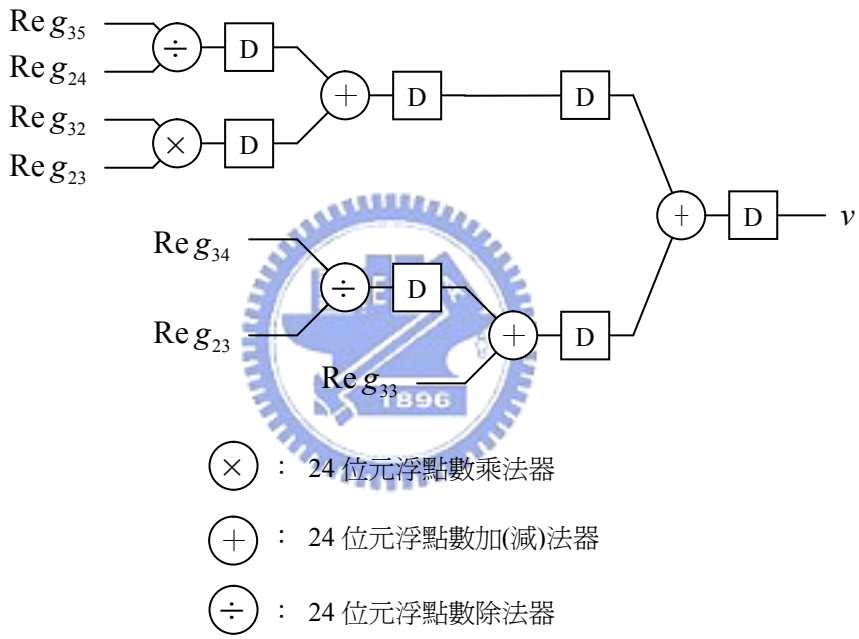


圖 3.5 三次多項式 $u-v$ 平面關係式運算元件

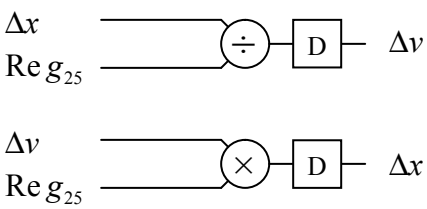


圖 3.6 寬度轉換關係式運算元件

(三) 多項式參數估測元件

執行多項式參數估測元件，首先需將偵測到的標線座標資訊代入(2.17)式中，利用 4 階多項式最小平方近似法運算單元計算 P_0、P_1、P_2、P_3、Δ後，將 $Coef_0=\frac{P_0}{\Delta}$、$Coef_1=\frac{P_1}{\Delta}$、$Coef_2=\frac{P_2}{\Delta}$、$Coef_3=\frac{P_3}{\Delta}$ 連同運算暫存器代入(2.18)~(2.21)中得：

$$\mathrm{Re}\,g_{26}=\frac{P_0\times \mathrm{Re}\,g_{12}}{\Delta}=\frac{P_0\cdot W}{\Delta}=b_0 \qquad (3.4)$$

$$\mathrm{Re}\,g_{27}=P_1\div(\mathrm{Re}\,g_7\times\Delta)=\frac{P_1}{(\tan q+1)\cdot e_u\cdot\Delta}=b_1 \qquad (3.5)$$

$$\mathrm{Re}\,g_{28}=P_2\div(\mathrm{Re}\,g_{19}\times\Delta)=\frac{P_2}{\{(\tan q+1)\cdot e_u\}^2\cdot W\cdot\Delta}=b_2 \qquad (3.6)$$

$$\mathrm{Re}\,g_{29}=P_3\div(\mathrm{Re}\,g_{20}\times\Delta)=\frac{P_3}{\{(\tan q+1)\cdot e_u\}^3\cdot W^2\cdot\Delta}=b_3 \qquad (3.7)$$

多項式參數估測元件運算結構如圖 3.7 所示，需使用一個三次多項式最小平方近似法運算單元、1 個浮點數除法器、1 個浮點數乘法器，不計最小平方近似法運算單元部分，總計需花費 6 個 clock 週期完成運算。(三次多項式最小平方近似法運算單元處理時間依據階數 n 的不同而不同，最慢在 2n+37 個 clock 週期內可以處理完成。)

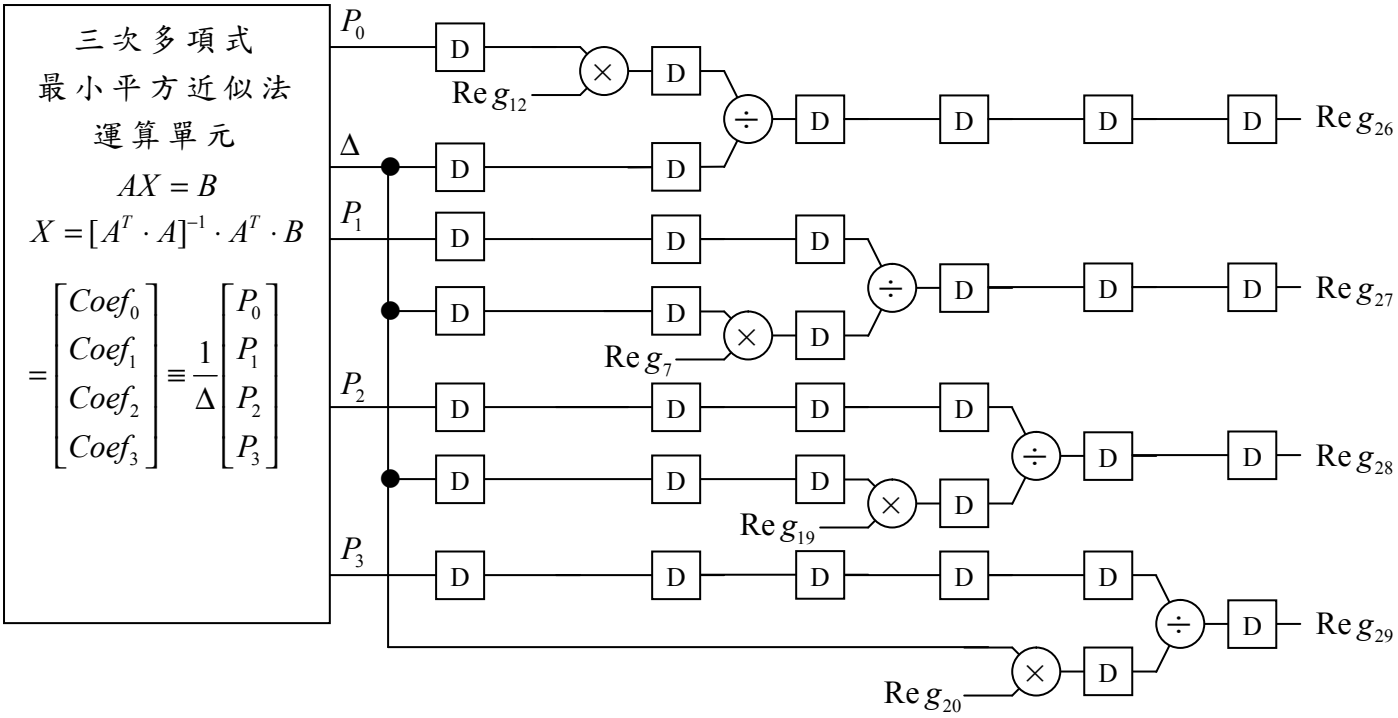


圖 3.7 多項式參數估測元件

(四) 道路模型參數估測元件

執行道路模型參數估測元件，如同多項式參數估測元件一樣，首先需將偵測到的標線座標資訊 $\Delta v = (v_R - v_L)$、u_M 代入公式 2.22 中，利用最小平方近似法(一次)計算 C_{wm1}、C_{wm0} 後連同將運算暫存器代入公式 2.23、2.24 中得：

$$\mathrm{Re}\,g_{12} = \frac{P_1 \times \mathrm{Re}\,g_{10}}{\Delta} = \frac{P_1 \cdot H}{\Delta} = -C_{wm0} \cdot H = W \qquad (3.8)$$

$$\mathrm{Re}\,g_2 = P_0 \div (P_1 \times \mathrm{Re}\,g_1) = \frac{P_0}{P_1 \cdot e_u} = \frac{C_{wm1}}{C_{wm0} \cdot e_v} = \tan q \qquad (3.9)$$

多項式參數估測元件運算結構如圖 3.8 所示，需使用一個一次多項式最小平方近似法運算單元、1 個浮點數除法器、1 個浮點數乘法器，不計最小平方近似法運算單元部分，總計需花費 4 個 clock 週期完成運算。(一次多項式最小平方近似法運算單元，結構類似三次多項式最小平方近似法運算單元，處理時間依據階數 n 的不同而不同，在 n+6 個 clock 週期內可以處理完成。)

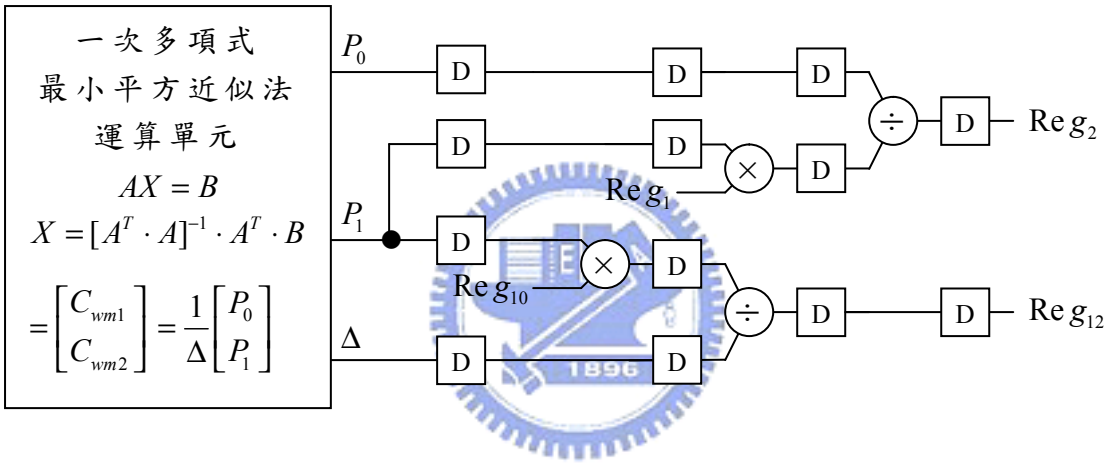


圖 3.8 道路模型參數估測元件

(五) 興趣區間(ROI)運算元件

首先將運算暫存器代入(2.25)~(2.30)中得：

$$f_{v_R}(U) = \frac{B_3}{U^2} + \frac{B_2}{U} + B_1 + B_{0_R} \cdot U = \mathrm{Re}\,g_{41} + \mathrm{Re}\,g_{40} + \mathrm{Re}\,g_{33} + \mathrm{Re}\,g_{38} \quad (3.10)$$

$$f_{v_L}(U) = \frac{B_3}{U^2} + \frac{B_2}{U} + B_1 + B_{0_L} \cdot U = \mathrm{Re}\,g_{41} + \mathrm{Re}\,g_{40} + \mathrm{Re}\,g_{33} + \mathrm{Re}\,g_{39} \quad (3.11)$$

$$f_{v_R}'(u) = (\frac{2B_3}{U^2} + \frac{B_2}{U}) \cdot \frac{1}{U} - B_{0_R} = \frac{2B_3}{U^3} + \frac{B_2}{U^2} - B_{0_R}$$

$$= (\mathrm{Re}\, g_{41} \times 2 + \mathrm{Re}\, g_{40}) \div \mathrm{Re}\, g_{24} - \mathrm{Re}\, g_{36} \qquad (3.12)$$

$$f_{v_L}'(u) = (\frac{2B_3}{U^2} + \frac{B_2}{U}) \cdot \frac{1}{U} - B_{0_L} = \frac{2B_3}{U^3} + \frac{B_2}{U^2} - B_{0_L}$$

$$= (\mathrm{Re}\, g_{41} \times 2 + \mathrm{Re}\, g_{40}) \div \mathrm{Re}\, g_{24} - \mathrm{Re}\, g_{37} \qquad (3.13)$$

$$\Delta_{ROI_R} = [w_{eight} \cdot T_w(\Delta x) + 2 \cdot f_{v_R}'(u_e) \cdot (u - u_e)]$$
$$= w_{eight} \times T_w(\Delta x)$$
$$+\{[(\mathrm{Re}\, g_{41} \times 2 + \mathrm{Re}\, g_{40}) \div \mathrm{Re}\, g_{24} - \mathrm{Re}\, g_{36}] \times [u - u_e]\} \times 2 \qquad (3.14)$$

$$\Delta_{ROI_L} = [w_{eight} \cdot T_w(\Delta x) + 2 \cdot f_{v_L}'(u_e) \cdot (u - u_e)]$$
$$= w_{eight} \times T_w(\Delta x)$$
$$+\{[(\mathrm{Re}\, g_{41} \times 2 + \mathrm{Re}\, g_{40}) \div \mathrm{Re}\, g_{24} - \mathrm{Re}\, g_{37}] \times [u - u_e]\} \times 2 \qquad (3.15)$$

$$ROI_{R_1} = f_{v_R}(u) + \Delta_{ROI_R} \qquad (3.16)$$

$$ROI_{R_2} = f_{v_R}(u) - \Delta_{ROI_R} \qquad (3.17)$$

$$ROI_{L_1} = f_{v_L}(u) + \Delta_{ROI_L} \qquad (3.18)$$

$$ROI_{L_2} = f_{v_L}(u) - \Delta_{ROI_L} \qquad (3.19)$$

其中 ROI_{R_1}、ROI_{R_2}、ROI_{L_1}、ROI_{L_2} 分別為左右興趣區間邊界座標。ROI 運算結構如圖 3.9 所示，需使用 1 個浮點數除法器、1 個浮點數乘法器、1 個浮點數加法器、1 個浮點數減法器，不計最小平方近似法運算單元部分，總計需花費 6 個 clock 週期完成運算。計算左右興趣區間為相同結構，使用一個 ROI 運算元件以摺疊(folding)方式實現左右 ROI 運算，全部運算在 12 個 clock 週期內完成。

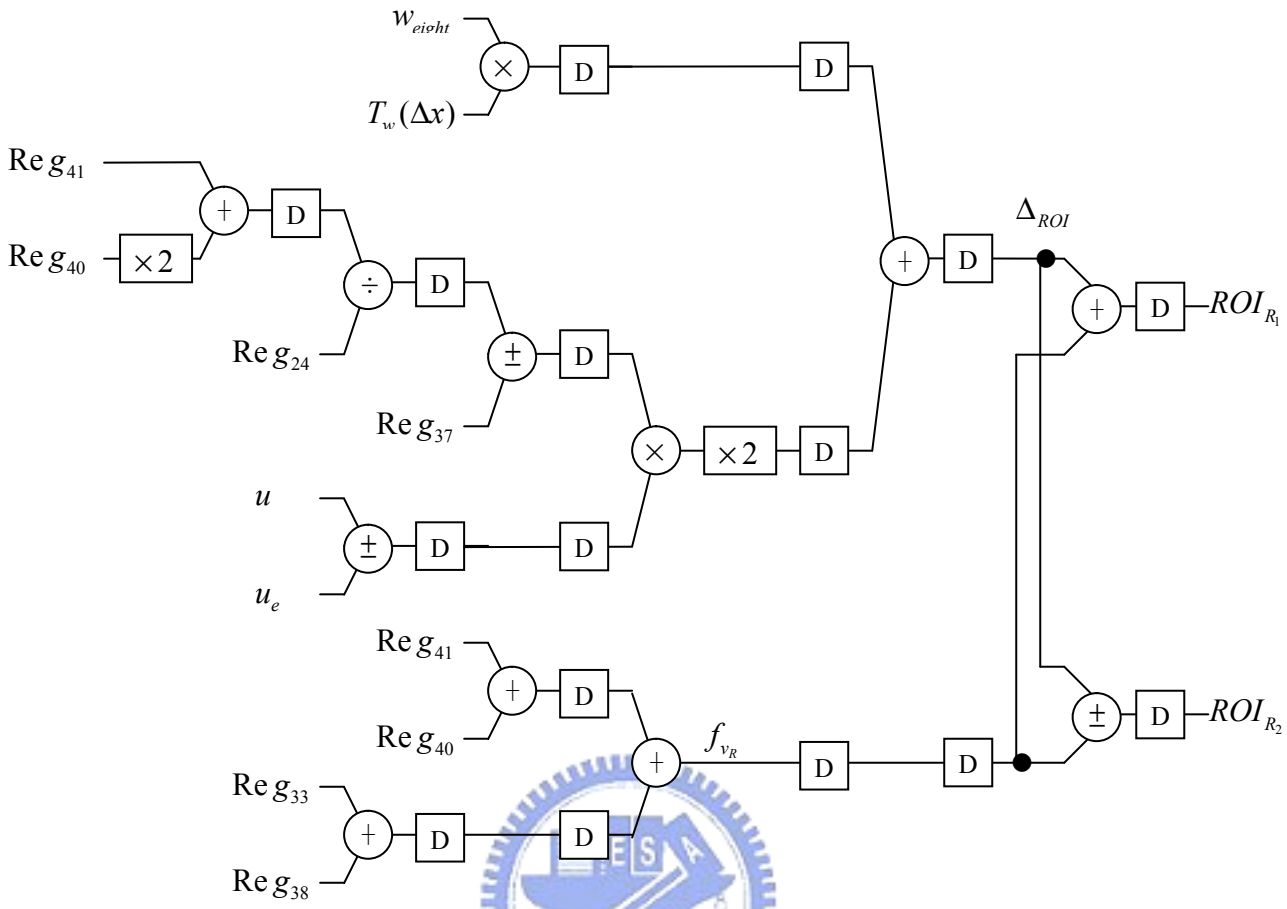


圖 3.9 興趣區間(ROI)運算元件

在浮點數運算中〝×2〞及〝÷2〞的運算並不需要用到真正的浮點數乘法器或除法器，只需將其指數(Exponent)部份加一或減一即可，此無號加(減)法運算連同浮點數運算單元仍在 1 個時脈週期內完成。

3.4　三次多項式最小平方近似法運算元件

(一)　最小平方近似法運算最佳化

實現三次多項式最小平方近似法運算元件，首先將所偵測到的標線座標資訊代入(2.17)式中寫成矩陣格式 $A_{nx4}\cdot X_{4x1}=B_{nx1}$，其中階數 n 等於全部搜尋到的點數，介於 15~210 之正整數，n=210 階表示 A 為 210x4 矩陣，因此執行最小平方近似法 $X=[A^T\cdot A]^{-1}\cdot A^T\cdot B$ 矩陣運算時，產生非常龐大的運算量，需要許多的硬體資源去實現。因此藉由分析 $X=[A^T\cdot A]^{-1}\cdot A^T\cdot B$ 運算結構，找尋共用結構以減少實現硬體所需資源。將所偵測到的標線座標資訊 $v_M=\frac{1}{2}(v_R+v_L)$ 、$\Delta v=(v_R-v_L)$ 代入(2.17)式中寫成矩陣格式如(3.20)式所示。

$$Coef_0\cdot\Delta v^3+Coef_1\cdot\Delta v^2+Coef_2\cdot\Delta v+Coef_3=v_M\cdot\Delta v^2$$

$AX=B$ 、 $X=[A^TA]^{-1}A^TB$

$$\begin{bmatrix}\Delta v_1^3 & \Delta v_1^2 & \Delta v_1 & 1\\ \Delta v_2^3 & \Delta v_2^2 & \Delta v_2 & 1\\ \Delta v_3^3 & \Delta v_3^2 & \Delta v_3 & 1\\ \text{M} & \text{M} & \text{M} & \text{M}\\ \Delta v_n^3 & \Delta v_n^2 & \Delta v_n & 1\end{bmatrix}\begin{bmatrix}Coef_0\\ Coef_1\\ Coef_2\\ Coef_3\end{bmatrix}=\begin{bmatrix}v_{M_1}\cdot\Delta v_1^2\\ v_{M_2}\cdot\Delta v_2^2\\ v_{M_3}\cdot\Delta v_3^2\\ \text{M}\\ v_{M_n}\cdot\Delta v_n^2\end{bmatrix}$$

$$A=\begin{bmatrix}\Delta v_1^3 & \Delta v_1^2 & \Delta v_1 & 1\\ \Delta v_2^3 & \Delta v_2^2 & \Delta v_2 & 1\\ \Delta v_3^3 & \Delta v_3^2 & \Delta v_3 & 1\\ \text{M} & \text{M} & \text{M} & \text{M}\\ \Delta v_n^3 & \Delta v_n^2 & \Delta v_n & 1\end{bmatrix}、X=\begin{bmatrix}Coef_0\\ Coef_1\\ Coef_2\\ Coef_3\end{bmatrix}、B=\begin{bmatrix}v_{M_1}\cdot\Delta v_1^2\\ v_{M_2}\cdot\Delta v_2^2\\ v_{M_3}\cdot\Delta v_3^2\\ \text{M}\\ v_{M_n}\cdot\Delta v_n^2\end{bmatrix}\qquad(3.20)$$

首先針對 $X=[A^T\cdot A]^{-1}\cdot A^T\cdot B$ 矩陣運算逐步展開分析如下：

$$A^TA=\begin{bmatrix}\Delta v_1^{\,6}+\Delta v_2^{\,6}+\mathrm{L}\ +\Delta v_n^{\,6} & \Delta v_1^{\,5}+\Delta v_2^{\,5}+\mathrm{L}\ +\Delta v_n^{\,5} & \Delta v_1^{\,4}+\Delta v_2^{\,4}+\mathrm{L}\ +\Delta v_n^{\,4} & \Delta v_1^{\,3}+\Delta v_2^{\,3}+\mathrm{L}\ +\Delta v_n^{\,3}\\ \Delta v_1^{\,5}+\Delta v_2^{\,5}+\mathrm{L}\ +\Delta v_n^{\,5} & \Delta v_1^{\,4}+\Delta v_2^{\,4}+\mathrm{L}\ +\Delta v_n^{\,4} & \Delta v_1^{\,3}+\Delta v_2^{\,3}+\mathrm{L}\ +\Delta v_n^{\,3} & \Delta v_1^{\,2}+\Delta v_2^{\,2}+\mathrm{L}\ +\Delta v_n^{\,2}\\ \Delta v_1^{\,4}+\Delta v_2^{\,4}+\mathrm{L}\ +\Delta v_n^{\,4} & \Delta v_1^{\,3}+\Delta v_2^{\,3}+\mathrm{L}\ +\Delta v_n^{\,3} & \Delta v_1^{\,2}+\Delta v_2^{\,2}+\mathrm{L}\ +\Delta v_n^{\,2} & \Delta v_1+\Delta v_2+\mathrm{L}\ +\Delta v_n\\ \Delta v_1^{\,3}+\Delta v_2^{\,3}+\mathrm{L}\ +\Delta v_n^{\,3} & \Delta v_1^{\,2}+\Delta v_2^{\,2}+\mathrm{L}\ +\Delta v_n^{\,2} & \Delta v_1+\Delta v_2+\mathrm{L}\ +\Delta v_n & n\end{bmatrix} \quad (3.21)$$

令 $S_n^m\equiv\sum_{i=1}^{n}\Delta v_i^{\,m}=\Delta v_1^{\,m}+\Delta v_2^{\,m}+\mathrm{L}\ +\Delta v_n^{\,m}$ 、$(S_n^0=n)$ 代入(3.20)中得：

$$A^T\cdot A=\begin{bmatrix}S_n^6 & S_n^5 & S_n^4 & S_n^3\\ S_n^5 & S_n^4 & S_n^3 & S_n^2\\ S_n^4 & S_n^3 & S_n^2 & S_n^1\\ S_n^3 & S_n^2 & S_n^1 & S_n^0\end{bmatrix} \quad (3.22)$$

在 $X=[A^TA]^{-1}A^TB$ 矩陣運算中、$[A^TA]^{-1}$ 運算會產生共同分母項

$\Delta=del(A^TA)$，因此將此項獨立提出，展開結果最佳化如下：

$$X=[A^TA]^{-1}A^TB=\begin{bmatrix}Coef_0\\ Coef_1\\ Coef_2\\ Coef_3\end{bmatrix}\equiv\frac{1}{\Delta}\begin{bmatrix}P_0\\ P_1\\ P_2\\ P_3\end{bmatrix} \quad (3.23)$$

此式

$$\begin{aligned}\Delta=del(A^TA)&=\left(S_n^0\right)\left(S_n^2\right)\left(S_n^4\right)\left(S_n^6\right)-\left(S_n^0\right)\left(S_n^2\right)\left(S_n^5\right)^2-\left(S_n^0\right)\left(S_n^3\right)^2\left(S_n^6\right)\\ &+2\left(S_n^0\right)\left(S_n^3\right)\left(S_n^4\right)\left(S_n^5\right)-\left(S_n^0\right)\left(S_n^4\right)^3-\left(S_n^1\right)^2\left(S_n^4\right)\left(S_n^6\right)+\left(S_n^1\right)^2\left(S_n^5\right)^2\\ &+2\left(S_n^1\right)\left(S_n^2\right)\left(S_n^3\right)\left(S_n^6\right)-2\left(S_n^1\right)\left(S_n^2\right)\left(S_n^4\right)\left(S_n^5\right)-2\left(S_n^1\right)\left(S_n^3\right)^2\left(S_n^5\right)\\ &+2\left(S_n^1\right)\left(S_n^3\right)\left(S_n^4\right)^2-\left(S_n^2\right)^3\left(S_n^6\right)+2\left(S_n^2\right)^2\left(S_n^3\right)\left(S_n^5\right)+\left(S_n^2\right)^2\left(S_n^4\right)^2\\ &-3\left(S_n^2\right)\left(S_n^3\right)^2\left(S_n^4\right)+\left(S_n^3\right)^4\end{aligned} \quad (3.24)$$

$$P_0 = v_{M_1} \cdot \Delta v_1^{\,5} \cdot R_1 + v_{M_2} \cdot \Delta v_2^{\,5} \cdot R_1 + \mathrm{L} + v_{M_n} \cdot \Delta v_n^{\,5} \cdot R_1$$
$$+ v_{M_1} \cdot \Delta v_1^{\,4} \cdot R_5 + v_{M_2} \cdot \Delta v_2^{\,4} \cdot R_5 + \mathrm{L} + v_{M_n} \cdot \Delta v_n^{\,4} \cdot R_5$$
$$+ v_{M_1} \cdot \Delta v_1^{\,3} \cdot R_7 + v_{M_2} \cdot \Delta v_2^{\,3} \cdot R_7 + \mathrm{L} + v_{M_n} \cdot \Delta v_n^{\,3} \cdot R_7$$
$$+ v_{M_1} \cdot \Delta v_1^{\,2} \cdot R_0 + v_{M_2} \cdot \Delta v_2^{\,2} \cdot R_0 + \mathrm{L} + v_{M_n} \cdot \Delta v_n^{\,2} \cdot R_0$$
$$= \sum_{i=1}^{n} v_{M_i} \times \left[\left(\Delta v_i^{\,5} \times R_1 + \Delta v_i^{\,4} \times R_5 \right) + \left(\Delta v_i^{\,3} \times R_7 + \Delta v_i^{\,2} \times R_0 \right) \right] \qquad (3.25a)$$
$$= \sum_{i=1}^{n} \left\{ v_{M_i} \times \Delta v_i^{\,2} \right\} \times \left\{ \Delta v_i \times [\Delta v_i \times (\Delta v_i \times R_1 + R_5) + R_7] + R_0 \right\} \qquad (3.25b)$$

$$P_1 = v_{M_1} \cdot \Delta v_1^{\,5} \cdot R_5 + v_{M_2} \cdot \Delta v_2^{\,5} \cdot R_5 + \mathrm{L} + v_{M_n} \cdot \Delta v_n^{\,5} \cdot R_5$$
$$+ v_{M_1} \cdot \Delta v_1^{\,4} \cdot R_3 + v_{M_2} \cdot \Delta v_2^{\,4} \cdot R_3 + \mathrm{L} + v_{M_n} \cdot \Delta v_n^{\,4} \cdot R_3$$
$$+ v_{M_1} \cdot \Delta v_1^{\,3} \cdot R_9 + v_{M_2} \cdot \Delta v_2^{\,3} \cdot R_9 + \mathrm{L} + v_{M_n} \cdot \Delta v_n^{\,3} \cdot R_9$$
$$+ v_{M_1} \cdot \Delta v_1^{\,2} \cdot R_8 + v_{M_2} \cdot \Delta v_2^{\,2} \cdot R_8 + \mathrm{L} + v_{M_n} \cdot \Delta v_n^{\,2} \cdot R_8$$
$$= \sum_{i=1}^{n} v_{M_i} \times \left[\left(\Delta v_i^{\,5} \times R_5 + \Delta v_i^{\,4} \times R_3 \right) + \left(\Delta v_i^{\,3} \times R_9 + \Delta v_i^{\,2} \times R_8 \right) \right] \qquad (3.26a)$$
$$= \sum_{i=1}^{n} \left\{ v_{M_i} \times \Delta v_i^{\,2} \right\} \times \left\{ \Delta v_i \times [\Delta v_i \times (\Delta v_i \times R_5 + R_3) + R_9] + R_8 \right\} \qquad (3.26b)$$

$$P_2 = v_{M_1} \cdot \Delta v_1^{\,5} \cdot R_7 + v_{M_2} \cdot \Delta v_2^{\,5} \cdot R_7 + \mathrm{L} + v_{M_n} \cdot \Delta v_n^{\,5} \cdot R_7$$
$$+ v_{M_1} \cdot \Delta v_1^{\,4} \cdot R_9 + v_{M_2} \cdot \Delta v_2^{\,4} \cdot R_9 + \mathrm{L} + v_{M_n} \cdot \Delta v_n^{\,4} \cdot R_9$$
$$+ v_{M_1} \cdot \Delta v_1^{\,3} \cdot R_4 + v_{M_2} \cdot \Delta v_2^{\,3} \cdot R_4 + \mathrm{L} + v_{M_n} \cdot \Delta v_n^{\,3} \cdot R_4$$
$$+ v_{M_1} \cdot \Delta v_1^{\,2} \cdot R_6 + v_{M_2} \cdot \Delta v_2^{\,2} \cdot R_6 + \mathrm{L} + v_{M_n} \cdot \Delta v_n^{\,2} \cdot R_6$$
$$= \sum_{i=1}^{n} v_{M_i} \times \left[\left(\Delta v_i^{\,5} \times R_1 + \Delta v_i^{\,4} \times R_9 \right) + \left(\Delta v_i^{\,3} \times R_4 + \Delta v_i^{\,2} \times R_6 \right) \right] \qquad (3.27a)$$
$$= \sum_{i=1}^{n} \left\{ v_{M_i} \times \Delta v_i^{\,2} \right\} \times \left\{ \Delta v_i \times [\Delta v_i \times (\Delta v_i \times R_1 + R_9) + R_4] + R_6 \right\} \qquad (3.27b)$$

$$P_3 = v_{M_1} \cdot \Delta v_1^{\,5} \cdot R_0 + v_{M_2} \cdot \Delta v_2^{\,5} \cdot R_0 + \mathrm{L} + v_{M_n} \cdot \Delta v_n^{\,5} \cdot R_0$$
$$+ v_{M_1} \cdot \Delta v_1^{\,4} \cdot R_8 + v_{M_2} \cdot \Delta v_2^{\,4} \cdot R_8 + \mathrm{L} + v_{M_n} \cdot \Delta v_n^{\,4} \cdot R_8$$
$$+ v_{M_1} \cdot \Delta v_1^{\,3} \cdot R_6 + v_{M_2} \cdot \Delta v_2^{\,3} \cdot R_6 + \mathrm{L} + v_{M_n} \cdot \Delta v_n^{\,3} \cdot R_6$$
$$+ v_{M_1} \cdot \Delta v_1^{\,2} \cdot R_2 + v_{M_2} \cdot \Delta v_2^{\,2} \cdot R_2 + \mathrm{L} + v_{M_n} \cdot \Delta v_n^{\,2} \cdot R_2$$
$$= \sum_{i=1}^{n} v_{M_i} \times \left[\left(\Delta v_i^{\,5} \times R_1 + \Delta v_i^{\,4} \times R_8 \right) + \left(\Delta v_i^{\,3} \times R_6 + \Delta v_i^{\,2} \times R_2 \right) \right] \qquad (3.28a)$$
$$= \sum_{i=1}^{n} \left\{ v_{M_i} \times \Delta v_i^{\,2} \right\} \times \left\{ \Delta v_i \times [\Delta v_i \times (\Delta v_i \times R_1 + R_8) + R_6] + R_2 \} \right\} \qquad (3.28b)$$

其中 $v_{M_i}=\frac{1}{2}\left(v_{R_i}+v_{L_i}\right)$ 、 $\Delta v_i=\left(v_{R_i}-v_{L_i}\right)$

$$R_0\equiv\left[+\left(S_n^3\right)^3+\left(S_n^1\right)\left(S_n^4\right)^2+\left(S_n^2\right)^2\left(S_n^5\right)-\left(S_n^1\right)\left(S_n^3\right)\left(S_n^5\right)-2\left(S_n^2\right)\left(S_n^3\right)\left(S_n^4\right)\right]$$

$$R_1\equiv\left[-\left(S_n^2\right)^3-\left(S_n^0\right)\left(S_n^3\right)^2-\left(S_n^1\right)^2\left(S_n^4\right)+\left(S_n^0\right)\left(S_n^2\right)\left(S_n^4\right)+2\left(S_n^1\right)\left(S_n^2\right)\left(S_n^3\right)\right]$$

$$R_2\equiv\left[-\left(S_n^4\right)^3-\left(S_n^2\right)\left(S_n^5\right)^2-\left(S_n^3\right)^2\left(S_n^6\right)+\left(S_n^2\right)\left(S_n^4\right)\left(S_n^6\right)+2\left(S_n^3\right)\left(S_n^4\right)\left(S_n^5\right)\right]$$

$$R_3\equiv\left[-\left(S_n^0\right)\left(S_n^4\right)^2-\left(S_n^2\right)\left(S_n^3\right)^2-\left(S_n^1\right)^2\left(S_n^6\right)+\left(S_n^0\right)\left(S_n^2\right)\left(S_n^6\right)+2\left(S_n^1\right)\left(S_n^3\right)\left(S_n^4\right)\right]$$

$$R_4\equiv\left[-\left(S_n^0\right)\left(S_n^5\right)^2-\left(S_n^2\right)^2\left(S_n^6\right)-\left(S_n^3\right)^2\left(S_n^4\right)+\left(S_n^0\right)\left(S_n^4\right)\left(S_n^6\right)+2\left(S_n^2\right)\left(S_n^3\right)\left(S_n^5\right)\right]$$

$$R_5\equiv\left[-\left(S_n^1\right)\left(S_n^3\right)^2+\left(S_n^1\right)^2\left(S_n^5\right)+\left(S_n^2\right)^2\left(S_n^3\right)-\left(S_n^0\right)\left(S_n^2\right)\left(S_n^5\right)+\left(S_n^0\right)\left(S_n^3\right)\left(S_n^4\right)-\left(S_n^1\right)\left(S_n^2\right)\left(S_n^4\right)\right]$$

$$R_6\equiv\left[+\left(S_n^1\right)\left(S_n^5\right)^2+\left(S_n^3\right)\left(S_n^4\right)^2-\left(S_n^3\right)^2\left(S_n^5\right)-\left(S_n^1\right)\left(S_n^4\right)\left(S_n^6\right)+\left(S_n^2\right)\left(S_n^3\right)\left(S_n^6\right)-\left(S_n^2\right)\left(S_n^4\right)\left(S_n^5\right)\right]$$

$$R_7\equiv\left[-\left(S_n^0\right)\left(S_n^4\right)^2-\left(S_n^2\right)\left(S_n^3\right)^2+\left(S_n^2\right)^2\left(S_n^4\right)+\left(S_n^0\right)\left(S_n^3\right)\left(S_n^5\right)-\left(S_n^1\right)\left(S_n^2\right)\left(S_n^5\right)+\left(S_n^1\right)\left(S_n^3\right)\left(S_n^4\right)\right]$$

$$R_8\equiv\left[+\left(S_n^2\right)\left(S_n^4\right)^2-\left(S_n^2\right)^2\left(S_n^6\right)-\left(S_n^3\right)^2\left(S_n^4\right)+\left(S_n^1\right)\left(S_n^3\right)\left(S_n^6\right)-\left(S_n^1\right)\left(S_n^4\right)\left(S_n^5\right)+\left(S_n^2\right)\left(S_n^3\right)\left(S_n^5\right)\right]$$

$$R_9\equiv\left[\left(S_n^3\right)^3-\left(S_n^0\right)\left(S_n^3\right)\left(S_n^6\right)+\left(S_n^0\right)\left(S_n^4\right)\left(S_n^5\right)+\left(S_n^1\right)\left(S_n^2\right)\left(S_n^6\right)-\left(S_n^1\right)\left(S_n^3\right)\left(S_n^5\right)-\left(S_n^2\right)\left(S_n^3\right)\left(S_n^4\right)\right]$$

(3.29)

(二) 運算暫存器

在最佳化運算式中有許多中間共同項。使用暫存器儲存這些中間項的運算結果，以減少運算量。另外 S_n^m 暫存器需累加 n 項，可打散於每次結束搜索程序後，儲存座標時更新，以減少最小平方近似法運算元件之 latency-time，其餘暫存器則在最小平方近似法運算元件執行前才更新。最佳化運算式中暫存器更新電路皆使用 24 位元浮點數格式儲存，全部更新程序總共需要 10 個 24 位元浮點數乘法器、9 個 24 位元浮點數加(減)法器，由 finite state machine 配置給各個更新電路所使用。

1. S_n^m 暫存器

$$Lq\_\mathrm{Re}g_0 = S_n^0 = n$$
$$Lq\_\mathrm{Re}g_1 = Lq\_\mathrm{Re}g_1 + \Delta v_i^1 = S_n^1$$
$$Lq\_\mathrm{Re}g_2 = Lq\_\mathrm{Re}g_2 + \Delta v_i^2 = S_n^2$$
$$Lq\_\mathrm{Re}g_3 = Lq\_\mathrm{Re}g_3 + \Delta v_i^3 = S_n^3$$
$$Lq\_\mathrm{Re}g_4 = Lq\_\mathrm{Re}g_4 + \Delta v_i^4 = S_n^4$$
$$Lq\_\mathrm{Re}g_5 = Lq\_\mathrm{Re}g_5 + \Delta v_i^5 = S_n^5$$
$$Lq\_\mathrm{Re}g_6 = Lq\_\mathrm{Re}g_6 + \Delta v_i^6 = S_n^6$$

S_n^m 暫存器在每次 n 變化時累加一次，在分析程序後加法器有空閒時進行，故不算入最小平方近似法運算元件處理時間。暫存器更新電路為單層結構，使用一個 24 位元浮點數加(減)法器實現即可，在一個 clock 週期內更新完畢。全數 S_n^m 暫存器在 6 個 clock 週期更新完畢。

2.一般中間項暫存器

中間項暫存器共 59 項分成 6 Level 執行，每個 Level 使用 10 個 24 位元浮點數乘法器執行更新程序，一個 clock 時脈週期內更新完畢，全數中間項暫存器更新程序於 6 個 clock 週期更新完畢。

Level 1 中間項暫存器：

$$Lq\_\mathrm{Re}g_{11} = Lq\_\mathrm{Re}g_1 \times Lq\_\mathrm{Re}g_1 = \left(S_n^1\right)^2$$
$$Lq\_\mathrm{Re}g_{22} = Lq\_\mathrm{Re}g_2 \times Lq\_\mathrm{Re}g_2 = \left(S_n^2\right)^2$$
$$Lq\_\mathrm{Re}g_{33} = Lq\_\mathrm{Re}g_3 \times Lq\_\mathrm{Re}g_3 = \left(S_n^3\right)^2$$
$$Lq\_\mathrm{Re}g_{44} = Lq\_\mathrm{Re}g_4 \times Lq\_\mathrm{Re}g_4 = \left(S_n^4\right)^2$$
$$Lq\_\mathrm{Re}g_{55} = Lq\_\mathrm{Re}g_5 \times Lq\_\mathrm{Re}g_5 = \left(S_n^5\right)^2$$

$$Lq\_\mathrm{Re}g_{02} = Lq\_\mathrm{Re}g_0 \times Lq\_\mathrm{Re}g_2 = \left(S_n^0\right)\left(S_n^2\right)$$
$$Lq\_\mathrm{Re}g_{03} = Lq\_\mathrm{Re}g_0 \times Lq\_\mathrm{Re}g_3 = \left(S_n^0\right)\left(S_n^3\right)$$
$$Lq\_\mathrm{Re}g_{12} = Lq\_\mathrm{Re}g_1 \times Lq\_\mathrm{Re}g_2 = \left(S_n^1\right)\left(S_n^2\right)$$
$$Lq\_\mathrm{Re}g_{13} = Lq\_\mathrm{Re}g_1 \times Lq\_\mathrm{Re}g_3 = \left(S_n^1\right)\left(S_n^3\right)$$
$$Lq\_\mathrm{Re}g_{23} = Lq\_\mathrm{Re}g_2 \times Lq\_\mathrm{Re}g_3 = \left(S_n^2\right)\left(S_n^3\right)$$

Level 2 中間項暫存器：

$$Lq\_\mathrm{Re}g_{45} = Lq\_\mathrm{Re}g_4 \times Lq\_\mathrm{Re}g_5 = \left(S_n^4\right)\left(S_n^5\right)$$
$$Lq\_\mathrm{Re}g_{46} = Lq\_\mathrm{Re}g_4 \times Lq\_\mathrm{Re}g_6 = \left(S_n^4\right)\left(S_n^6\right)$$
$$Lq\_\mathrm{Re}g_{222} = Lq\_\mathrm{Re}g_{22} \times Lq\_\mathrm{Re}g_2 = \left(S_n^2\right)^3$$
$$Lq\_\mathrm{Re}g_{333} = Lq\_\mathrm{Re}g_{33} \times Lq\_\mathrm{Re}g_3 = \left(S_n^3\right)^3$$
$$Lq\_\mathrm{Re}g_{444} = Lq\_\mathrm{Re}g_{44} \times Lq\_\mathrm{Re}g_4 = \left(S_n^4\right)^3$$

$$Lq\_\mathrm{Re}g_{033} = Lq\_\mathrm{Re}g_0 \times Lq\_\mathrm{Re}g_{33} = \left(S_n^0\right)\left(S_n^3\right)^2$$
$$Lq\_\mathrm{Re}g_{044} = Lq\_\mathrm{Re}g_0 \times Lq\_\mathrm{Re}g_{44} = \left(S_n^0\right)\left(S_n^4\right)^2$$
$$Lq\_\mathrm{Re}g_{055} = Lq\_\mathrm{Re}g_0 \times Lq\_\mathrm{Re}g_{55} = \left(S_n^0\right)\left(S_n^5\right)^2$$
$$Lq\_\mathrm{Re}g_{133} = Lq\_\mathrm{Re}g_1 \times Lq\_\mathrm{Re}g_{33} = \left(S_n^1\right)\left(S_n^3\right)^2$$
$$Lq\_\mathrm{Re}g_{144} = Lq\_\mathrm{Re}g_1 \times Lq\_\mathrm{Re}g_{44} = \left(S_n^1\right)\left(S_n^4\right)^2$$

Level 3 中間項暫存器：

$$Lq\_\mathrm{Re}g_{155} = Lq\_\mathrm{Re}g_1 \times Lq\_\mathrm{Re}g_{55} = \left(S_n^1\right)\left(S_n^5\right)^2$$
$$Lq\_\mathrm{Re}g_{233} = Lq\_\mathrm{Re}g_2 \times Lq\_\mathrm{Re}g_{33} = \left(S_n^2\right)\left(S_n^3\right)^2$$
$$Lq\_\mathrm{Re}g_{244} = Lq\_\mathrm{Re}g_2 \times Lq\_\mathrm{Re}g_{44} = \left(S_n^2\right)\left(S_n^4\right)^2$$
$$Lq\_\mathrm{Re}g_{255} = Lq\_\mathrm{Re}g_2 \times Lq\_\mathrm{Re}g_{55} = \left(S_n^2\right)\left(S_n^5\right)^2$$
$$Lq\_\mathrm{Re}g_{344} = Lq\_\mathrm{Re}g_3 \times Lq\_\mathrm{Re}g_{44} = \left(S_n^3\right)\left(S_n^4\right)^2$$
$$Lq\_\mathrm{Re}g_{114} = Lq\_\mathrm{Re}g_{11} \times Lq\_\mathrm{Re}g_4 = \left(S_n^1\right)^2\left(S_n^4\right)$$
$$Lq\_\mathrm{Re}g_{115} = Lq\_\mathrm{Re}g_{11} \times Lq\_\mathrm{Re}g_5 = \left(S_n^1\right)^2\left(S_n^5\right)$$
$$Lq\_\mathrm{Re}g_{116} = Lq\_\mathrm{Re}g_{11} \times Lq\_\mathrm{Re}g_6 = \left(S_n^1\right)^2\left(S_n^6\right)$$
$$Lq\_\mathrm{Re}g_{223} = Lq\_\mathrm{Re}g_{22} \times Lq\_\mathrm{Re}g_3 = \left(S_n^2\right)^2\left(S_n^3\right)$$

$$Lq\_\mathrm{Re}g_{224} = Lq\_\mathrm{Re}g_{22} \times Lq\_\mathrm{Re}g_{4} = \left(S_n^2\right)^2\left(S_n^4\right)$$

Level 4 中間項暫存器：

$$Lq\_\mathrm{Re}g_{225} = Lq\_\mathrm{Re}g_{22} \times Lq\_\mathrm{Re}g_{5} = \left(S_n^2\right)^2\left(S_n^5\right)$$
$$Lq\_\mathrm{Re}g_{226} = Lq\_\mathrm{Re}g_{22} \times Lq\_\mathrm{Re}g_{6} = \left(S_n^2\right)^2\left(S_n^6\right)$$
$$Lq\_\mathrm{Re}g_{334} = Lq\_\mathrm{Re}g_{33} \times Lq\_\mathrm{Re}g_{4} = \left(S_n^3\right)^2\left(S_n^4\right)$$
$$Lq\_\mathrm{Re}g_{335} = Lq\_\mathrm{Re}g_{33} \times Lq\_\mathrm{Re}g_{5} = \left(S_n^3\right)^2\left(S_n^5\right)$$
$$Lq\_\mathrm{Re}g_{336} = Lq\_\mathrm{Re}g_{33} \times Lq\_\mathrm{Re}g_{6} = \left(S_n^3\right)^2\left(S_n^6\right)$$

$$Lq\_\mathrm{Re}g_{024} = Lq\_\mathrm{Re}g_{02} \times Lq\_\mathrm{Re}g_{4} = \left(S_n^0\right)\left(S_n^2\right)\left(S_n^4\right)$$
$$Lq\_\mathrm{Re}g_{025} = Lq\_\mathrm{Re}g_{02} \times Lq\_\mathrm{Re}g_{5} = \left(S_n^0\right)\left(S_n^2\right)\left(S_n^5\right)$$
$$Lq\_\mathrm{Re}g_{026} = Lq\_\mathrm{Re}g_{02} \times Lq\_\mathrm{Re}g_{6} = \left(S_n^0\right)\left(S_n^2\right)\left(S_n^6\right)$$
$$Lq\_\mathrm{Re}g_{034} = Lq\_\mathrm{Re}g_{03} \times Lq\_\mathrm{Re}g_{4} = \left(S_n^0\right)\left(S_n^3\right)\left(S_n^4\right)$$
$$Lq\_\mathrm{Re}g_{035} = Lq\_\mathrm{Re}g_{03} \times Lq\_\mathrm{Re}g_{5} = \left(S_n^0\right)\left(S_n^3\right)\left(S_n^5\right)$$

Level 5 中間項暫存器：

$$Lq\_\mathrm{Re}g_{036} = Lq\_\mathrm{Re}g_{03} \times Lq\_\mathrm{Re}g_{6} = \left(S_n^0\right)\left(S_n^3\right)\left(S_n^6\right)$$
$$Lq\_\mathrm{Re}g_{123} = Lq\_\mathrm{Re}g_{12} \times Lq\_\mathrm{Re}g_{3} = \left(S_n^1\right)\left(S_n^2\right)\left(S_n^3\right)$$
$$Lq\_\mathrm{Re}g_{124} = Lq\_\mathrm{Re}g_{12} \times Lq\_\mathrm{Re}g_{4} = \left(S_n^1\right)\left(S_n^2\right)\left(S_n^4\right)$$
$$Lq\_\mathrm{Re}g_{125} = Lq\_\mathrm{Re}g_{12} \times Lq\_\mathrm{Re}g_{5} = \left(S_n^1\right)\left(S_n^2\right)\left(S_n^5\right)$$
$$Lq\_\mathrm{Re}g_{126} = Lq\_\mathrm{Re}g_{12} \times Lq\_\mathrm{Re}g_{6} = \left(S_n^1\right)\left(S_n^2\right)\left(S_n^6\right)$$
$$Lq\_\mathrm{Re}g_{134} = Lq\_\mathrm{Re}g_{13} \times Lq\_\mathrm{Re}g_{4} = \left(S_n^1\right)\left(S_n^3\right)\left(S_n^4\right)$$
$$Lq\_\mathrm{Re}g_{135} = Lq\_\mathrm{Re}g_{13} \times Lq\_\mathrm{Re}g_{5} = \left(S_n^1\right)\left(S_n^3\right)\left(S_n^5\right)$$
$$Lq\_\mathrm{Re}g_{136} = Lq\_\mathrm{Re}g_{13} \times Lq\_\mathrm{Re}g_{6} = \left(S_n^1\right)\left(S_n^3\right)\left(S_n^6\right)$$
$$Lq\_\mathrm{Re}g_{234} = Lq\_\mathrm{Re}g_{23} \times Lq\_\mathrm{Re}g_{4} = \left(S_n^2\right)\left(S_n^3\right)\left(S_n^4\right)$$
$$Lq\_\mathrm{Re}g_{235} = Lq\_\mathrm{Re}g_{23} \times Lq\_\mathrm{Re}g_{5} = \left(S_n^2\right)\left(S_n^3\right)\left(S_n^5\right)$$

Level 6 中間項暫存器：

$$Lq\_\mathrm{Re}g_{236} = Lq\_\mathrm{Re}g_{23} \times Lq\_\mathrm{Re}g_{6} = \left(S_n^2\right)\left(S_n^3\right)\left(S_n^6\right)$$
$$Lq\_\mathrm{Re}g_{045} = Lq\_\mathrm{Re}g_{0} \times Lq\_\mathrm{Re}g_{45} = \left(S_n^0\right)\left(S_n^4\right)\left(S_n^5\right)$$
$$Lq\_\mathrm{Re}g_{145} = Lq\_\mathrm{Re}g_{1} \times Lq\_\mathrm{Re}g_{45} = \left(S_n^1\right)\left(S_n^4\right)\left(S_n^5\right)$$
$$Lq\_\mathrm{Re}g_{245} = Lq\_\mathrm{Re}g_{2} \times Lq\_\mathrm{Re}g_{45} = \left(S_n^2\right)\left(S_n^4\right)\left(S_n^5\right)$$
$$Lq\_\mathrm{Re}g_{345} = Lq\_\mathrm{Re}g_{3} \times Lq\_\mathrm{Re}g_{45} = \left(S_n^3\right)\left(S_n^4\right)\left(S_n^5\right)$$
$$Lq\_\mathrm{Re}g_{046} = Lq\_\mathrm{Re}g_{0} \times Lq\_\mathrm{Re}g_{46} = \left(S_n^0\right)\left(S_n^4\right)\left(S_n^6\right)$$
$$Lq\_\mathrm{Re}g_{146} = Lq\_\mathrm{Re}g_{1} \times Lq\_\mathrm{Re}g_{46} = \left(S_n^1\right)\left(S_n^4\right)\left(S_n^6\right)$$
$$Lq\_\mathrm{Re}g_{246} = Lq\_\mathrm{Re}g_{2} \times Lq\_\mathrm{Re}g_{46} = \left(S_n^2\right)\left(S_n^4\right)\left(S_n^6\right)$$
$$Lq\_\mathrm{Re}g_{2334} = Lq\_\mathrm{Re}g_{233} \times Lq\_\mathrm{Re}g_{4} = \left(S_n^2\right)\left(S_n^3\right)^2\left(S_n^4\right)$$

3. R_0~R_9 暫存器

R_0~R_9 暫存器更新電路為 3 層結構，R_5~R_9 暫存器更新電路中使用 5 個 24 位元浮點數加(減)法器實現，R_0~R_4 暫存器更新電路中使用 4 個 24 位元浮點數加(減)法器，暫存器更新電路各在 3 個 clock 時脈週期內更新完畢，全數暫存器更新程序在 15 個 clock 週期更新完畢。

$$R_0 \equiv \left[\left(S_n^3\right)^3 + \left(S_n^1\right)\left(S_n^4\right)^2 + \left(S_n^2\right)^2\left(S_n^5\right) - \left(S_n^1\right)\left(S_n^3\right)\left(S_n^5\right) - 2\left(S_n^2\right)\left(S_n^3\right)\left(S_n^4\right)\right]$$
$$= Lq\_\mathrm{Re}g_{333} + Lq\_\mathrm{Re}g_{144} + Lq\_\mathrm{Re}g_{225} - Lq\_\mathrm{Re}g_{135} - Lq\_\mathrm{Re}g_{234} \times 2$$
$$= \left[\left(Lq\_\mathrm{Re}g_{225} - Lq\_\mathrm{Re}g_{135}\right) - \left(Lq\_\mathrm{Re}g_{234} \times 2\right)\right] + \left(Lq\_\mathrm{Re}g_{333} + Lq\_\mathrm{Re}g_{144}\right)$$

$$R_1 \equiv \left[-\left(S_n^2\right)^3 - \left(S_n^0\right)\left(S_n^3\right)^2 - \left(S_n^1\right)^2\left(S_n^4\right) + \left(S_n^0\right)\left(S_n^2\right)\left(S_n^4\right) + 2\left(S_n^1\right)\left(S_n^2\right)\left(S_n^3\right)\right]$$
$$= -Lq\_\mathrm{Re}g_{222} - Lq\_\mathrm{Re}g_{033} - Lq\_\mathrm{Re}g_{114} + Lq\_\mathrm{Re}g_{024} + Lq\_\mathrm{Re}g_{123} \times 2$$
$$= \left[\left(Lq\_\mathrm{Re}g_{024} - Lq\_\mathrm{Re}g_{114}\right) + \left(Lq\_\mathrm{Re}g_{123} \times 2\right)\right] - \left(Lq\_\mathrm{Re}g_{222} + Lq\_\mathrm{Re}g_{033}\right)$$

$$R_2 \equiv \left[-\left(S_n^4\right)^3 - \left(S_n^2\right)\left(S_n^5\right)^2 - \left(S_n^3\right)^2\left(S_n^6\right) + \left(S_n^2\right)\left(S_n^4\right)\left(S_n^6\right) + 2\left(S_n^3\right)\left(S_n^4\right)\left(S_n^5\right) \right]$$
$$= -Lq\_\mathrm{Re}\, g_{444} - Lq\_\mathrm{Re}\, g_{255} - Lq\_\mathrm{Re}\, g_{336} + Lq\_\mathrm{Re}\, g_{246} + Lq\_\mathrm{Re}\, g_{345} \times 2$$
$$= \left[\left(Lq\_\mathrm{Re}\, g_{246} - Lq\_\mathrm{Re}\, g_{336}\right) + \left(Lq\_\mathrm{Re}\, g_{345} \times 2\right) \right] - \left(Lq\_\mathrm{Re}\, g_{444} + Lq\_\mathrm{Re}\, g_{255}\right)$$

$$R_3 \equiv \left[-\left(S_n^0\right)\left(S_n^4\right)^2 - \left(S_n^2\right)\left(S_n^3\right)^2 - \left(S_n^1\right)^2\left(S_n^6\right) + \left(S_n^0\right)\left(S_n^2\right)\left(S_n^6\right) + 2\left(S_n^1\right)\left(S_n^3\right)\left(S_n^4\right) \right]$$
$$= -Lq\_\mathrm{Re}\, g_{044} - Lq\_\mathrm{Re}\, g_{233} - Lq\_\mathrm{Re}\, g_{116} + Lq\_\mathrm{Re}\, g_{026} + Lq\_\mathrm{Re}\, g_{134} \times 2$$
$$= \left[\left(Lq\_\mathrm{Re}\, g_{026} - Lq\_\mathrm{Re}\, g_{116}\right) + \left(Lq\_\mathrm{Re}\, g_{134} \times 2\right) \right] - \left(Lq\_\mathrm{Re}\, g_{044} + Lq\_\mathrm{Re}\, g_{233}\right)$$

$$R_4 \equiv \left[-\left(S_n^0\right)\left(S_n^5\right)^2 - \left(S_n^2\right)^2\left(S_n^6\right) - \left(S_n^3\right)^2\left(S_n^4\right) + \left(S_n^0\right)\left(S_n^4\right)\left(S_n^6\right) + 2\left(S_n^2\right)\left(S_n^3\right)\left(S_n^5\right) \right]$$
$$= -Lq\_\mathrm{Re}\, g_{055} - Lq\_\mathrm{Re}\, g_{226} - Lq\_\mathrm{Re}\, g_{334} + Lq\_\mathrm{Re}\, g_{046} + Lq\_\mathrm{Re}\, g_{235} \times 2$$
$$= \left[\left(Lq\_\mathrm{Re}\, g_{046} - Lq\_\mathrm{Re}\, g_{334}\right) + \left(Lq\_\mathrm{Re}\, g_{235} \times 2\right) \right] - \left(Lq\_\mathrm{Re}\, g_{055} + Lq\_\mathrm{Re}\, g_{226}\right)$$

$$R_5 \equiv \left[-\left(S_n^1\right)\left(S_n^3\right)^2 + \left(S_n^1\right)^2\left(S_n^5\right) + \left(S_n^2\right)^2\left(S_n^3\right) - \left(S_n^0\right)\left(S_n^2\right)\left(S_n^5\right) + \left(S_n^0\right)\left(S_n^3\right)\left(S_n^4\right) - \left(S_n^1\right)\left(S_n^2\right)\left(S_n^4\right) \right]$$
$$= -Lq\_\mathrm{Re}\, g_{133} + Lq\_\mathrm{Re}\, g_{115} + Lq\_\mathrm{Re}\, g_{223} - Lq\_\mathrm{Re}\, g_{025} + Lq\_\mathrm{Re}\, g_{034} - Lq\_\mathrm{Re}\, g_{124}$$
$$= \left(Lq\_\mathrm{Re}\, g_{115} - Lq\_\mathrm{Re}\, g_{133}\right) + \left[\left(Lq\_\mathrm{Re}\, g_{223} - Lq\_\mathrm{Re}\, g_{025}\right) + \left(Lq\_\mathrm{Re}\, g_{034} - Lq\_\mathrm{Re}\, g_{124}\right) \right]$$

$$R_6 \equiv \left[+\left(S_n^1\right)\left(S_n^5\right)^2 + \left(S_n^3\right)\left(S_n^4\right)^2 - \left(S_n^3\right)^2\left(S_n^5\right) - \left(S_n^1\right)\left(S_n^4\right)\left(S_n^6\right) + \left(S_n^2\right)\left(S_n^3\right)\left(S_n^6\right) - \left(S_n^2\right)\left(S_n^4\right)\left(S_n^5\right) \right]$$
$$= Lq\_\mathrm{Re}\, g_{155} + Lq\_\mathrm{Re}\, g_{344} - Lq\_\mathrm{Re}\, g_{335} - Lq\_\mathrm{Re}\, g_{146} + Lq\_\mathrm{Re}\, g_{236} - Lq\_\mathrm{Re}\, g_{245}$$
$$= \left(Lq\_\mathrm{Re}\, g_{155} + Lq\_\mathrm{Re}\, g_{344}\right) + \left[\left(Lq\_\mathrm{Re}\, g_{236} - Lq\_\mathrm{Re}\, g_{245}\right) - \left(Lq\_\mathrm{Re}\, g_{335} + Lq\_\mathrm{Re}\, g_{146}\right) \right]$$

$$R_7 \equiv \left[-\left(S_n^0\right)\left(S_n^4\right)^2 - \left(S_n^2\right)\left(S_n^3\right)^2 + \left(S_n^2\right)^2\left(S_n^4\right) + \left(S_n^0\right)\left(S_n^3\right)\left(S_n^5\right) - \left(S_n^1\right)\left(S_n^2\right)\left(S_n^5\right) + \left(S_n^1\right)\left(S_n^3\right)\left(S_n^4\right) \right]$$
$$= -Lq\_\mathrm{Re}\, g_{044} - Lq\_\mathrm{Re}\, g_{233} + Lq\_\mathrm{Re}\, g_{224} + Lq\_\mathrm{Re}\, g_{035} - Lq\_\mathrm{Re}\, g_{125} + Lq\_\mathrm{Re}\, g_{134}$$
$$= -\left(Lq\_\mathrm{Re}\, g_{044} + Lq\_\mathrm{Re}\, g_{233}\right) + \left[\left(Lq\_\mathrm{Re}\, g_{224} + Lq\_\mathrm{Re}\, g_{035}\right) + \left(Lq\_\mathrm{Re}\, g_{134} - Lq\_\mathrm{Re}\, g_{125}\right) \right]$$

$$R_8 \equiv \left[+\left(S_n^2\right)\left(S_n^4\right)^2 - \left(S_n^2\right)^2\left(S_n^6\right) - \left(S_n^3\right)^2\left(S_n^4\right) + \left(S_n^1\right)\left(S_n^3\right)\left(S_n^6\right) - \left(S_n^1\right)\left(S_n^4\right)\left(S_n^5\right) + \left(S_n^2\right)\left(S_n^3\right)\left(S_n^5\right) \right]$$
$$= Lq\_\mathrm{Re}\, g_{244} - Lq\_\mathrm{Re}\, g_{226} - Lq\_\mathrm{Re}\, g_{334} + Lq\_\mathrm{Re}\, g_{136} - Lq\_\mathrm{Re}\, g_{145} + Lq\_\mathrm{Re}\, g_{235}$$
$$= \left(Lq\_\mathrm{Re}\, g_{244} - Lq\_\mathrm{Re}\, g_{226}\right) + \left[\left(Lq\_\mathrm{Re}\, g_{136} + Lq\_\mathrm{Re}\, g_{235}\right) - \left(Lq\_\mathrm{Re}\, g_{334} + Lq\_\mathrm{Re}\, g_{145}\right) \right]$$

$$R_9 \equiv \left[\left(S_n^3\right)^3 - \left(S_n^0\right)\left(S_n^3\right)\left(S_n^6\right) + \left(S_n^0\right)\left(S_n^4\right)\left(S_n^5\right) + \left(S_n^1\right)\left(S_n^2\right)\left(S_n^6\right) - \left(S_n^1\right)\left(S_n^3\right)\left(S_n^5\right) - \left(S_n^2\right)\left(S_n^3\right)\left(S_n^4\right) \right]$$
$$= Lq\_\mathrm{Re}\, g_{333} - Lq\_\mathrm{Re}\, g_{036} + Lq\_\mathrm{Re}\, g_{045} + Lq\_\mathrm{Re}\, g_{126} - Lq\_\mathrm{Re}\, g_{135} - Lq\_\mathrm{Re}\, g_{234}$$
$$= \left(Lq\_\mathrm{Re}\, g_{333} - Lq\_\mathrm{Re}\, g_{036}\right) + \left[\left(Lq\_\mathrm{Re}\, g_{045} + Lq\_\mathrm{Re}\, g_{126}\right) - \left(Lq\_\mathrm{Re}\, g_{135} + Lq\_\mathrm{Re}\, g_{234}\right) \right]$$

4. P_0~P_3分子項暫存器

◆平行結構(parallel)

$$P_0 = \sum_{i=1}^{n} v_{M_i} \times \left\{ \left[\left(\Delta v_i^5 \times R_1 \right) + \left(\Delta v_i^4 \times R_5 \right) \right] + \left[\left(\Delta v_i^3 \times R_7 \right) + \left(\Delta v_i^2 \times R_0 \right) \right] \right\}$$

$$P_1 = \sum_{i=1}^{n} v_{M_i} \times \left\{ \left[\left(\Delta v_i^5 \times R_5 \right) + \left(\Delta v_i^4 \times R_3 \right) \right] + \left[\left(\Delta v_i^3 \times R_9 \right) + \left(\Delta v_i^2 \times R_8 \right) \right] \right\}$$

$$P_2 = \sum_{i=1}^{n} v_{M_i} \times \left\{ \left[\left(\Delta v_i^5 \times R_7 \right) + \left(\Delta v_i^4 \times R_9 \right) \right] + \left[\left(\Delta v_i^3 \times R_4 \right) + \left(\Delta v_i^2 \times R_6 \right) \right] \right\}$$

$$P_3 = \sum_{i=1}^{n} v_{M_i} \times \left\{ \left[\left(\Delta v_i^5 \times R_0 \right) + \left(\Delta v_i^4 \times R_8 \right) \right] + \left[\left(\Delta v_i^3 \times R_6 \right) + \left(\Delta v_i^2 \times R_2 \right) \right] \right\}$$

分子項平行結構為 5 級電路，使用 5 個 24 位元浮點數乘法器、4 個 24 位元浮點數加(減)法器執行更新程序，在 n+5 個 clock 時脈週期內執行完畢。為加速運算時間，同時使用 2 組更新電路作平行處理，P_0、P_1共用一組更新電路，P_2、P_3共用一組更新電路。總計使用 10 個 24 位元浮點數乘法器與 8 個 24 位元浮點數加(減)法器，在 2n+10 個 clock 時脈週期內完成運算，電路結構如圖 3.10 所示。

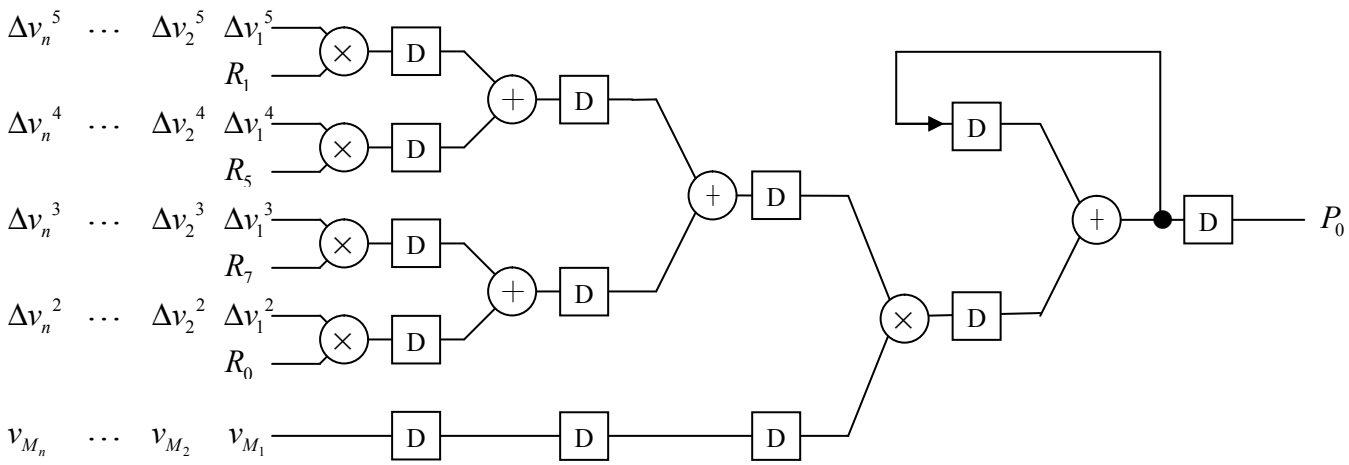


圖 3.10 最小平方近似法分子項平行結構

◆摺疊結構(folding)

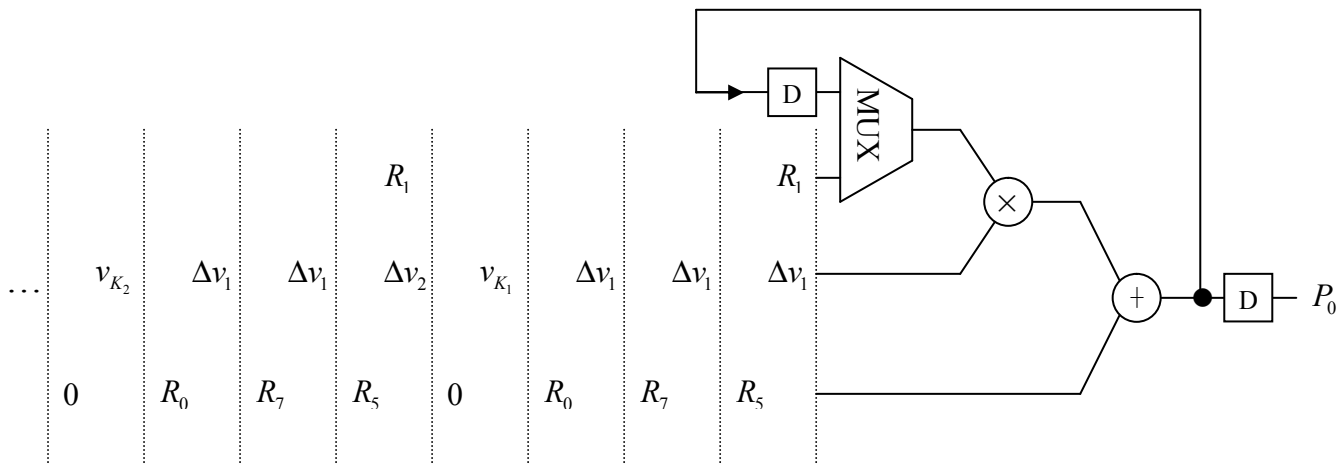
$$P_0 = \sum_{i=1}^{n} \left\{ v_{M_i} \times \Delta v_i^2 \right\} \times \left\{ \Delta v_i \times [\Delta v_i \times (\Delta v_i \times R_1 + R_5) + R_7] + R_0 \right\}$$

$$P_1 = \sum_{i=1}^{n} \left\{ v_{M_i} \times \Delta v_i^2 \right\} \times \left\{ \Delta v_i \times [\Delta v_i \times (\Delta v_i \times R_5 + R_3) + R_9] + R_8 \right\}$$

$$P_2 = \sum_{i=1}^{n} \left\{ v_{M_i} \times \Delta v_i^2 \right\} \times \left\{ \Delta v_i \times [\Delta v_i \times (\Delta v_i \times R_1 + R_9) + R_4] + R_6 \right\}$$

$$P_3 = \sum_{i=1}^{n} \left\{ v_{M_i} \times \Delta v_i^2 \right\} \times \left\{ \Delta v_i \times [\Delta v_i \times (\Delta v_i \times R_1 + R_8) + R_6] + R_2 \right\}$$

分子項摺疊結構為 9 級電路，使用 1 個 24 位元浮點數乘法器、1 個 24 位元浮點數加(減)法器執行更新程序，在 16n 個 clock 時脈週期內執行完畢。為加速運算時間，同時使用 4 組更新電路作平行處理，總計使用 4 個 24 位元浮點數乘法器與 4 個 24 位元浮點數加(減)法器，電路結構如圖 3.11 所示，其中 $v_{K_i} \equiv v_{M_i} \times \Delta v_i^2$，在搜尋&分析程序結束後執行運算，儲存於記憶體中，故不計入摺疊結構中。



(Iteration period = 2 system clock)

圖 3.11 最小平方近似法分子項摺疊結構

以平行結構實現分子項電路，總計使用 10 個乘法器、8 個加(減)法器，於 2n+10 個時脈週期完成運算，而摺疊結構使用 4 個乘法器、4 個加(減)法器，於 16n 個時脈週期完成運算，雖然平行結構使用較多硬體，但運算時間比摺疊結構快上 8 倍。最小平方近似法運算單元是決定車道偵測處理器處理時間的關鍵元件，因此選擇平行結構實現分子項電路比較適合，在非運算時間中，加(減)法器、乘法器等運算單元亦由 final state machine 配置給其餘運算元件共同使用，以避免資源浪費。

5.分母項暫存器

$$\Delta = del(A^T A) = \left(S_n^0\right)\left(S_n^2\right)\left(S_n^4\right)\left(S_n^6\right) - \left(S_n^0\right)\left(S_n^2\right)\left(S_n^5\right)^2 - \left(S_n^0\right)\left(S_n^3\right)^2\left(S_n^6\right) + 2\left(S_n^0\right)\left(S_n^3\right)\left(S_n^4\right)\left(S_n^5\right)$$
$$-\left(S_n^0\right)\left(S_n^4\right)^3 - \left(S_n^1\right)^2\left(S_n^4\right)\left(S_n^6\right) + \left(S_n^1\right)^2\left(S_n^5\right)^2 + 2\left(S_n^1\right)\left(S_n^2\right)\left(S_n^3\right)\left(S_n^6\right) - 2\left(S_n^1\right)\left(S_n^2\right)\left(S_n^4\right)\left(S_n^5\right)$$
$$-2\left(S_n^1\right)\left(S_n^3\right)^2\left(S_n^5\right) + 2\left(S_n^1\right)\left(S_n^3\right)\left(S_n^4\right)^2 - \left(S_n^2\right)^3\left(S_n^6\right) + 2\left(S_n^2\right)^2\left(S_n^3\right)\left(S_n^5\right) + \left(S_n^2\right)^2\left(S_n^4\right)^2$$
$$-3\left(S_n^2\right)\left(S_n^3\right)^2\left(S_n^4\right) + \left(S_n^3\right)^4$$

$$= \left\{Lq\_\mathrm{Re}g_6 \times \left\{\left[\left(Lq\_\mathrm{Re}g_{024} - Lq\_\mathrm{Re}g_{033}\right) - \left(Lq\_\mathrm{Re}g_{114} + Lq\_\mathrm{Re}g_{222}\right)\right] + Lq\_\mathrm{Re}g_{123} \times 2\right\}\right\}$$
$$+ \left\{Lq\_\mathrm{Re}g_5 \times \left\{Lq\_\mathrm{Re}g_{115} + \left[\left(Lq\_\mathrm{Re}g_{223} + Lq\_\mathrm{Re}g_{034}\right) - \left(Lq\_\mathrm{Re}g_{124} + Lq\_\mathrm{Re}g_{133}\right)\right] \times 2\right\}\right\}$$
$$+ \left\{\left[Lq\_\mathrm{Re}g_4 \times \left(Lq\_\mathrm{Re}g_{224} + Lq\_\mathrm{Re}g_{134} \times 2\right)\right] - \left[Lq\_\mathrm{Re}g_0 \times \left(Lq\_\mathrm{Re}g_{255} + Lq\_\mathrm{Re}g_{444}\right)\right]\right\}$$
$$+ \left\{\left[Lq\_\mathrm{Re}g_{333} \times Lq\_\mathrm{Re}g_3\right] + \left[Lq\_\mathrm{Re}g_{2334} + Lq\_\mathrm{Re}g_{2334} \times 2\right]\right\}$$

分母項暫存器為 6 層結構，使用 2 個 24 位元浮點數乘法器、4 位元浮點數加(減)法器，在 6 個 clock 時脈週期內執行完畢，結構圖如圖 3.12 所示。

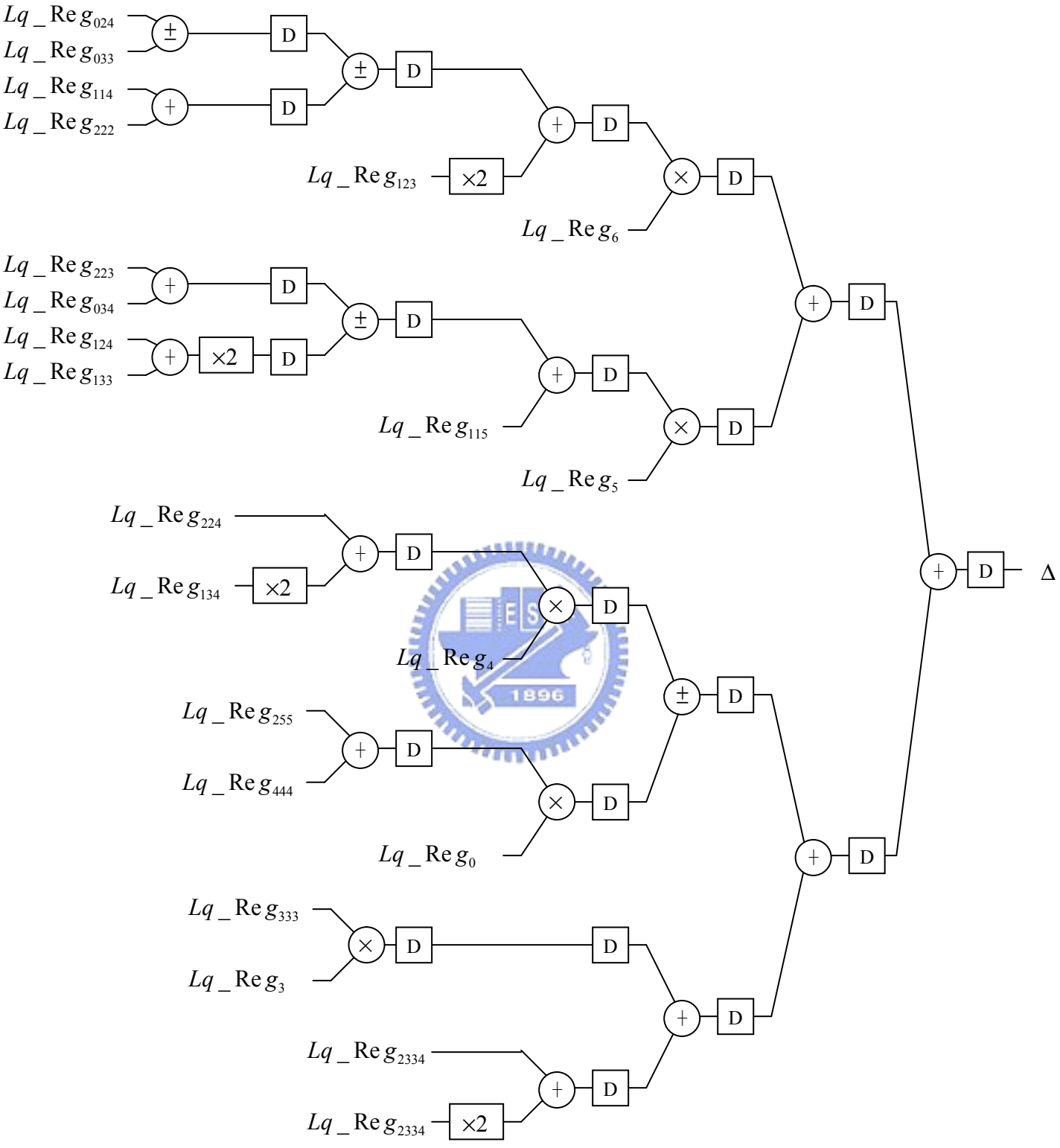
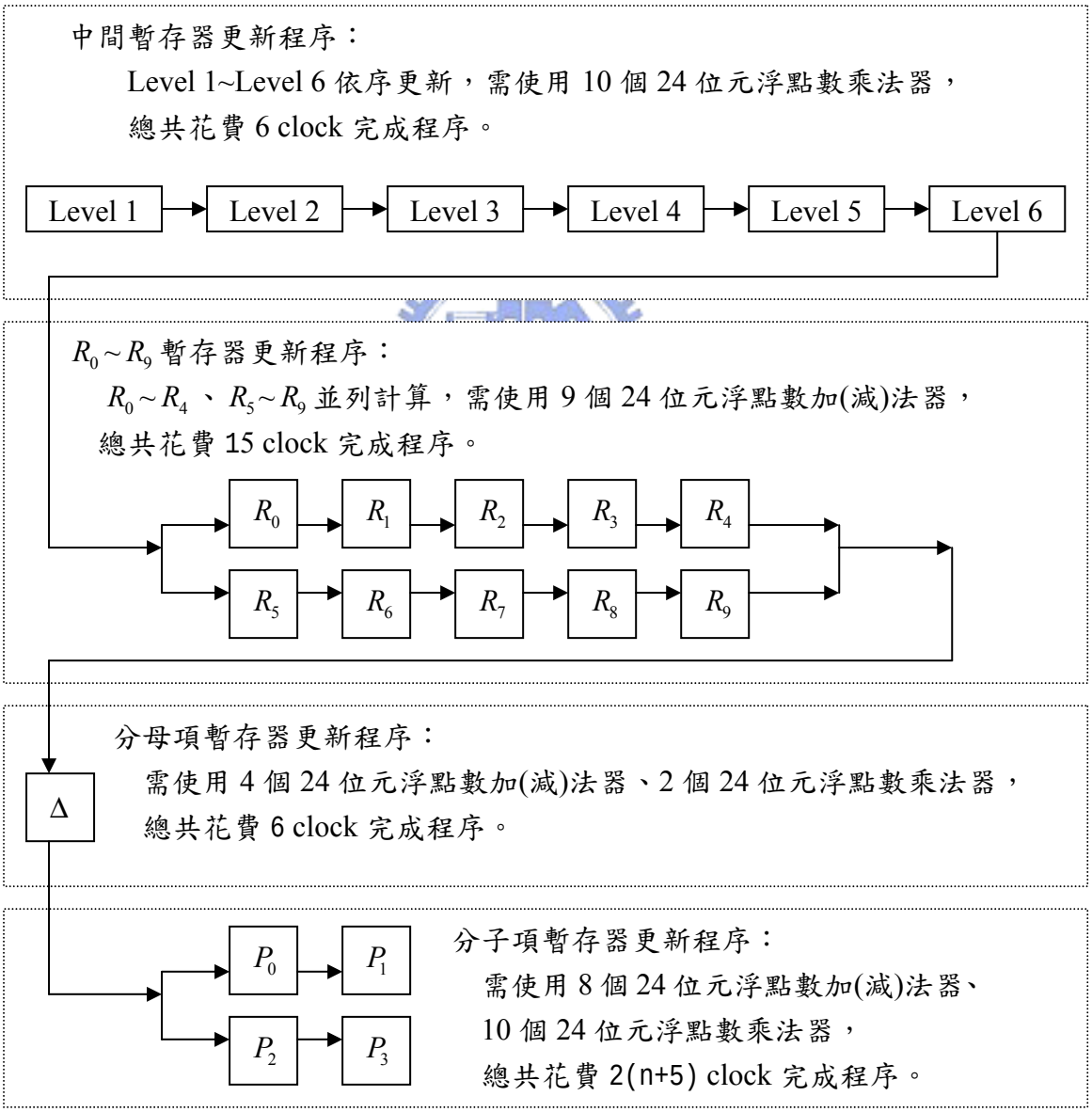


圖 3.12 最小平方近似法分母項更新電路

(三) 討論

最小平方近似法運算元件總計使用 10 個浮點數乘法器、9 個元浮點數加(減)法器、2 個浮點數位移器(shifter〝<<1〞)來實現，於 2n+27 個 clock 時脈週期內完成運算，其中 n 介於 15~210 間之正整數，為目前程序搜尋到的標線座標數。因此最小平方近似法運算元件最慢在 247 個 clock 週期內可以處理完成(n=210)。程序方塊圖如圖 3.13 所示。



(以上運算單元皆由 finite state machine 配置給各電路共同使用)

圖 3.13 最小平方近似法運算元件程序方塊圖

3.5 一次多項式最小平方近似法運算元件

一次多項式最小平方近似法運算元件，實現方式同三次多項式最小平方近似法運算元件將所偵測到的標線座標資訊 u 、 $\Delta v=(v_R-v_L)$ 代入(2.2b)式中寫成矩陣格式 $A_{nx2}\cdot X_{2x1}=B_{nx1}$ ，如(3.29)式所示。

$$Cwm_0\cdot u+Cwm_1=\Delta v^2$$

$$AX=B \text{ 、 } X=[A^TA]^{-1}A^TB$$

$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \text{M} & \text{M} \\ u_3 & 1 \end{bmatrix}\begin{bmatrix} Cwm_0 \\ Cwm_1 \end{bmatrix}=\begin{bmatrix} \Delta v_1 \\ \Delta v_2 \\ \text{M} \\ \Delta v_n \end{bmatrix}$$

$$A=\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \text{M} & \text{M} \\ u_3 & 1 \end{bmatrix} \text{ 、 } X=\begin{bmatrix} Cwm_0 \\ Cwm_1 \end{bmatrix} \text{ 、 } B=\begin{bmatrix} \Delta v_1 \\ \Delta v_2 \\ \text{M} \\ \Delta v_n \end{bmatrix} \qquad (3.29)$$

首先針對 $X=[A^T\cdot A]^{-1}\cdot A^T\cdot B$ 矩陣運算逐步展開分析如下：

$$A^TA=\begin{bmatrix} u_1^2+u_2^2+\text{L}\ u_n^2 & u_1+u_2+\text{L}\ u_n \\ u_1+u_2+\text{L}\ u_n & n \end{bmatrix} \qquad (3.30)$$

令 $U_n^m\equiv\sum_{i=1}^{n}u_i^m=u_1^m+u_2^m+\text{L}\ +u_n^m$ 、 $(U_n^0=n)$ 代入(3.30)中得：

$$A^T\cdot A=\begin{bmatrix} U_n^2 & U_n^1 \\ U_n^1 & U_n^0 \end{bmatrix} \qquad (3.31)$$

在 $X=[A^TA]^{-1}A^TB$ 矩陣運算中、 $[A^TA]^{-1}$ 運算會產生共同分母項

$\Delta = del(A^T A)$，因此將此項獨立提出，展開結果最佳化如下：

$$X = [A^T A]^{-1} A^T B = \begin{bmatrix} Cwm_0 \\ Cwm_1 \end{bmatrix} \equiv \frac{1}{\Delta} \begin{bmatrix} P_0 \\ P_1 \end{bmatrix} \tag{3.32}$$

此式

$$\Delta = del(A^T A) = \left(U_n^2\right)\left(U_n^0\right) - \left(U_n^1\right)^2 \tag{3.33}$$

$$P_0 = \sum_{i=1}^{n} \Delta v_i \times \left\{u_i \times U_n^0 - U_n^1\right\} \tag{3.34}$$

$$P_1 = \sum_{i=1}^{n} \Delta v_i \times \left\{-u_i \times U_n^1 + U_n^2\right\} \tag{3.35}$$

分子項為 4 級電路，使用 2 個 24 位元浮點數乘法器、2 個 24 位元浮點數加(減)法器執行更新程序，在 n+4 個 clock 時脈週期內執行完畢。為加速運算時間，同時使用 2 組更新電路作平行處理，總計使用 4 個 24 位元浮點數乘法器與 4 個 24 位元浮點數加(減)法器，電路結構如圖 3.14 所示。

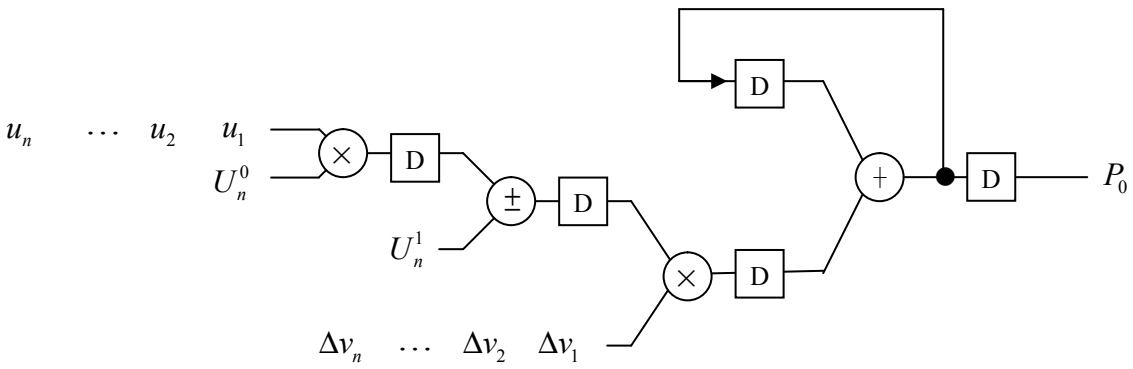


圖 3.14 一次多項式最小平方近似法分子項運算結構

分母項為 2 級乘加結構，使用 2 個 24 位元浮點數乘法器、1 個 24 位元浮點數加(減)法器執行更新程序，在 2 個 clock 時脈週期內執行完畢，電路結構如圖 3.15 所示。

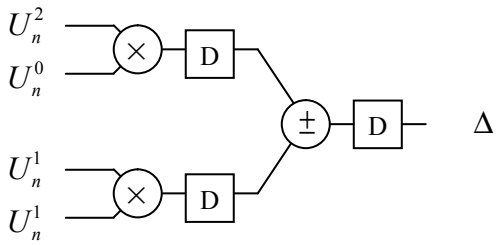


圖 3.15 一次多項式最小平方近似法分母項運算結構

U_n^m 暫存器在每次 n 變化時累加一次，在分析程序後加法器有空閒時進行，故不算入最小平方近似法運算元件處理時間。一次多項式最小平方近似法運算元件總計使用 4 個 24 位元浮點數乘法器與 4 個 24 位元浮點數加(減)法器，在 n+6 個 clock 時脈週期內執行完畢。

第四章　硬體系統架構

硬體系統採用 ALTERA 公司生產的”PCI Development Board – Stratix EP1S25F1020C5 Device”來執行車道偵測演算法運算。基於價格與相容性的考量，前端影像擷取採用 PC 準系統架構，影像經 CCD 攝影機擷取後，透過影像擷取卡傳入 PC(使用單鏡頭 644 X 493 像素、256 灰階影像)，再藉由 PCI 介面傳入 FPGA(ALTERA Stratix EP1S25F1020C5 Device)作運算，運算結果在透過 PCI 介面傳回到 PC 準系統，以供後端 ITS 系統(如警報系統、方向盤煞車控制等)使用。硬體系統方塊圖如圖 4.1 所示。

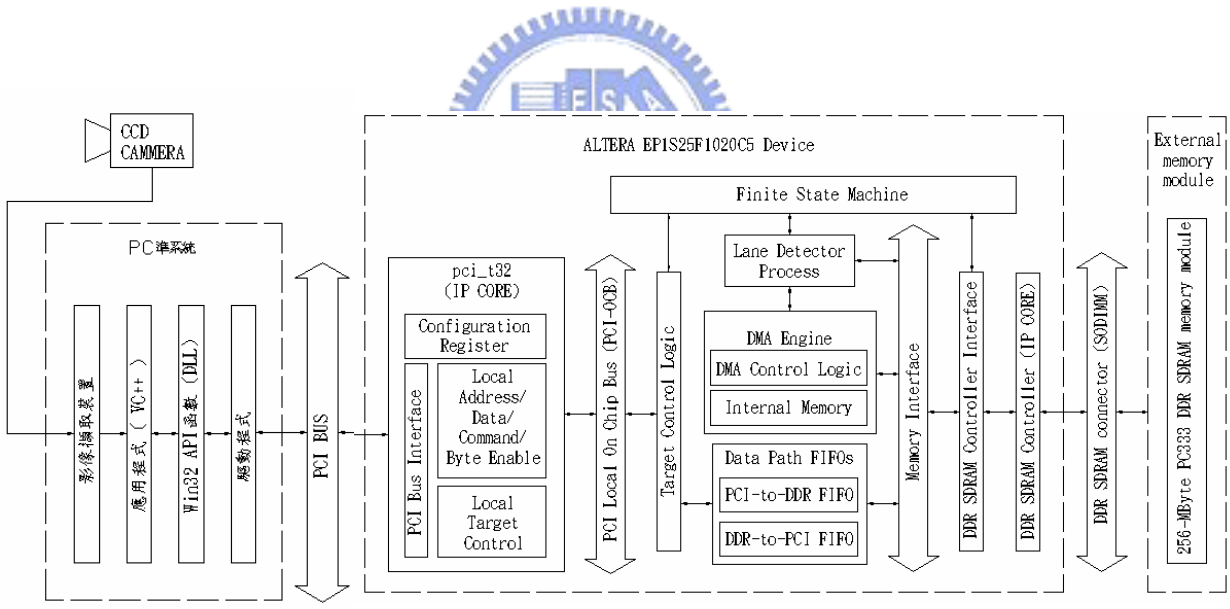


圖 4.1 硬體系統方塊圖

4.1 PC 準系統端軟硬體架構

選用 PC 準系統(搭配 Windows XP 作業系統)當作前端影像擷取並作為後端偵測資訊的應用平台，主要原因在於價格考量與相容性。PC 準系統不但非常容易取得且價格低廉，對於擴充性更是其他系統無法比擬的。用來發展前端影像擷取系統，對於鏡頭與擷取卡的選擇上有非常大的空間。在後端車道資訊應用上作為發展平台，可以更容易的達到客制化的要求。

PC 端對於本例主要只有兩項功能，首先將擷取影像透過 PCI 介面傳入 PCI 裝置(PCI Development Board - Stratix EP1S25F1020C5 Device)中，待車道偵測程序處理完後，在透過 PCI 介面將車道資訊從 PCI 裝置中讀出。但要完成以上功能必須透過一連串複雜的程序始能達成，尤其是自行開發的 PCI 裝置需要開發相對應的驅動程式。

Windows 2000/XP 作業系統的元件主要分為兩部分，一部分在用使用者模式(user model)，另一部分使用在核心模式(kernel model)中。對於工作在核心模式的設備驅動程式而言，較有關係的元件為：I/O 管理器(I/O Manager)、即插即用管理器(PnP Manager)、電源管理器(Power Manager)、硬體抽象層(Hardware Abstraction Layer)、配置管理器、記憶體管理器等等。

通常開發驅動程式有許多種模式，而 Windows 2000/XP 對驅動程式的編寫不再基于以往的 Win 9x 下的 VDD（虛擬設備驅動程式）架構，因為 Windows 2000/XP 並不允許使用者

對硬體直接操作、只能透過作業系統控制硬體。在此環境下開發驅動程式主要採用 WDM (Windows Driver Model)模式開發。

WDM 為 Windows 98/2000/XP 作業系統的設備驅動程式的設計提供了統一的框架。WDM 來源于 Windows NT 的分層 32 位元設備驅動程式模型（layered 32-bit device driver model)。它支持更多的特性，如即插即用（PnP)、電源管理、WMI 和 NT 事件。設備驅動程式是作業系統的一個組成部分，它由 I/O 管理器(I/O Manager)管理和調動。I/O 管理器每收到一個來自用戶應用程式的請求就建立一個 I/O 請求封包（IRP）的資料結構，並將其作為參數傳遞給驅動程式。驅動程式透過識別 IRP 中的實際設備對象（PDO）來區別是發送給哪一個設備。IRP 架構中存放請求的類型、用戶緩衝區的起始位址、用戶需求數據的長度等訊息。驅動程式處理完這個請求後，在該架構中填入處理結果的有關訊息，呼叫 IoCompleteRequest 程序，返回給 I/O 管理器，用戶應用程式的請求隨即結束。存取硬體時，驅動程式透過呼叫硬體抽象層的函數實現。

本文中 PC 端主要執行程序如下：首先由前端 CCD 攝影機擷取到前方車道影像後，藉由擷取裝置送至 PC 端記憶體暫存。此時應用程式(VC++)讀取暫存影像後，透過 Win32 API 函數(DLL)呼叫系統服務介面(I/O Manager)，再藉由系統服務介面呼叫核心模式中的服務常式(WDM 驅動程式)，此時服務常式即可存取硬體抽象層(Hardware Abstraction Layer)控制 PCI BUS 讀取，而作業系統(Windown 2000/XP)會負責維持

硬體抽象層與實際硬體間的映射關係。PC 準系統端架構如圖 4.2 所示。

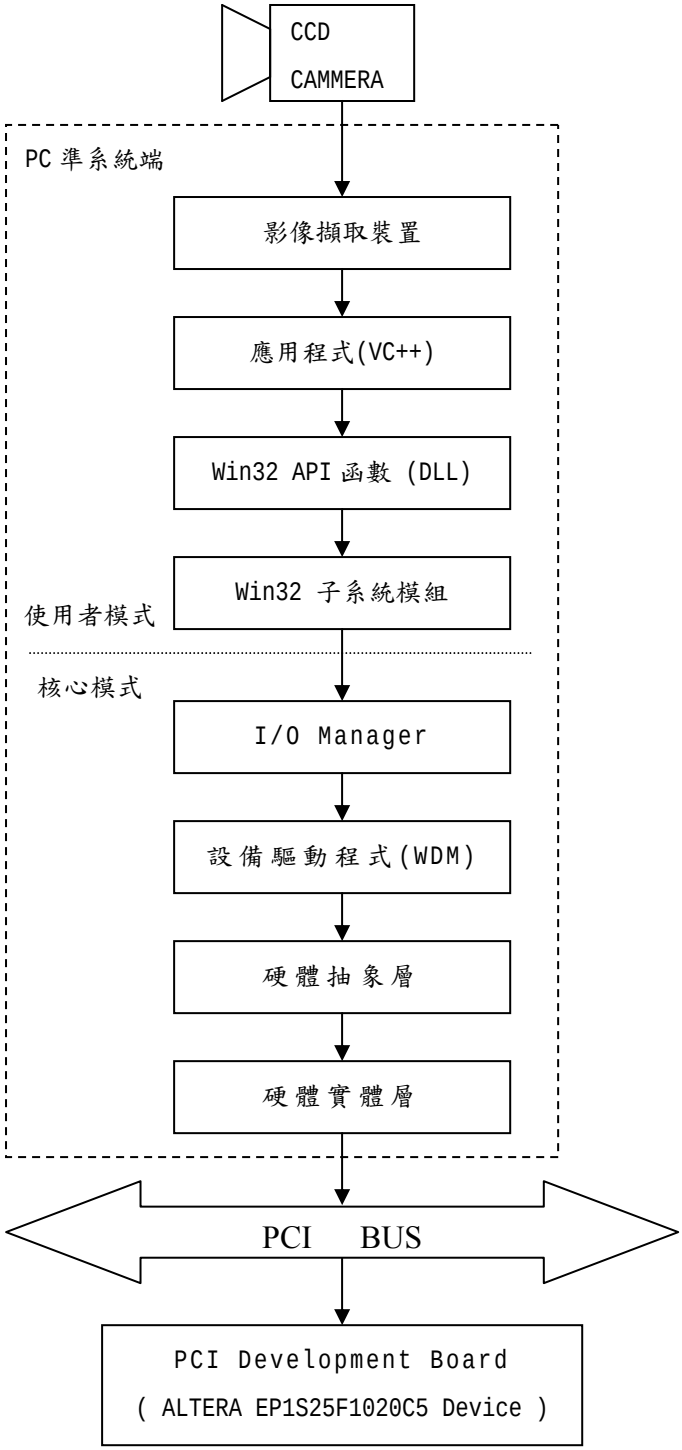


圖 4.2 PC 準系統端架構方塊圖

4.2 PCI 介面

(一) PCI 介面簡介

Intel 公司於 1992 年制定了 PCI(Peripheral Component Interface)匯流排規格，以確保市場上各種不同功能處理器區域匯流排架構上的相容性 (目前最新版本為 3.0 版)。PCI 匯流排上裝置可彼此間或者與系統記憶體間快速存取。所有匯流排讀寫的傳輸皆能用 burst 方式來執行，由匯流排主控端(Bus Master)來決定 burst 的長度。在資料交易開始時，被控端(Target)會被指定起始位址與交易型態，但不告知 Target 其傳輸長度，當 Master 準備傳輸資料時，會通知 Target 此資料是否為最後一筆(由 FRAMEn 訊號告知)，傳完最後一筆交易即完成。效率上明顯較 ISA 或是 VESA 等介面好，現今已成為 PC 準系統標準匯流排規格。而顯示介面(AGP)可視為 PCI 介面之延伸，主要提供高頻寬給予圖形介面所使用，而現今最新的顯示介面 PCI-Express 亦是以 PCI 匯流排為基礎之延伸介面。

PCI 裝置可分類為被控端(Target)、主控端(Master)、初始端(Initator)三種。一般 PC 準系統中，初始端與主控端可視為同一裝置(Initator 亦具備 Master 之權限)，都可以控制 PCI 匯流排，PC 準系統中之北橋晶片通常內定為初始端(Initator)，PCI 匯流排可支援多 Master 匯流排，權限判別由匯流排仲裁者(bus arbiter)亦即是 Initator 判定。若匯流排上無相關 PCI Master 功能晶片之匯流排，則系統內僅有之主控端(Master)即為初始端(Initator)。

本文中應用程式藉由驅動程式存取硬體抽象層，透過作

業系統控制北橋晶片作為初始端(Initator)及 主控端(Master)，而”PCI Development Board”規劃作為被控端(Target)。

(二) PCI 連接器與接腳

一般而言 PCI 接腳分為必要接腳與選用接腳，必要接腳包含資料流接腳(AD[31:0]、CBEn[3:0]、PAR)、介面控制訊號(FRAMEn、TRDYn、IRDYn、STOPn、DEVSELn、IDSEL 等)、以及 Error 訊號(PERRn、SERRn)與系統訊號(CLK、RSTn)等。另外若是主控端(Master)則還包括仲裁訊號(arbitation)。選用接腳則包含 64-bit 延伸接腳、LOCKn 訊號（介面控制接腳）、中斷訊號（INTAn、INTBn、INTCn、INTDn）、快取記憶體控制訊號(SBOn、SDONE)與 JTAG 接腳。PCI 訊號分類如圖 4.3 所示。

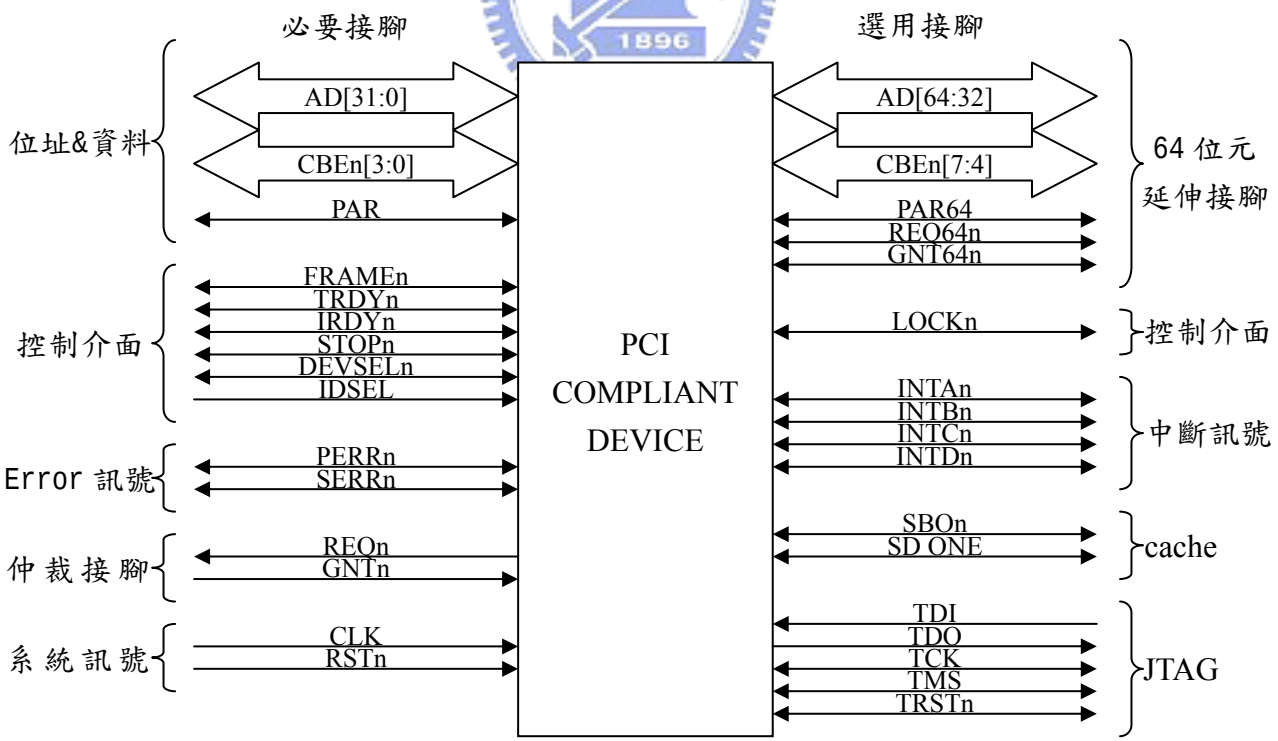


圖 4.3 PCI 訊號分類圖

PCI 接腳邏輯定義可分輸入接腳、輸出接腳、三態邏輯接腳(Tri-State)、保持式三態邏輯接腳(Sustain Tri-State)與開汲極(Open Drain)等五種型態。保持式三態邏輯接腳操作需特別注意，此接腳只能被一個裝置驅動，不驅動訊號時，必須拉高訊號一個時脈週期後浮接，此外如同開汲極一樣，還必須增加提升電阻使得再浮接時仍保持高電壓準位。由於晶片包裝接腳有限，故晶片外匯流排規格多採雙向多工型態共用接腳以節省 pin 腳使用。而此類接腳在換邊驅動時需要一段兩邊都不驅動時間呈現開汲極狀態，此時稱為反轉週期(因 Sustain Tri-State 型態所致)。許多 PCI 匯流排訊號亦屬此類，如位址與資料共用接腳(AD[32:0])、資料致能與命令共用接腳(CBEn [3:0])等。

本文中”PCI Development Board”所使用到的 PCI 訊號功能定義與驅動規則於 4.3 中再作詳細介紹。關於其他 PCI 匯流排的規格請參照 PCI Special Interest Group (PCI-SIG)制定的”PCI Local Bus Specifiation,Revision 3.0 ”文件[11]。

(三) PCI 驅動程序

對 PC 端而言每個 PCI 裝置都有三個識別編號，分為匯流排編號（bus number）、裝置編號（device number）、功能編號（function number）。在 PCI 的規格中，允許系統最多可接 256 組匯流排（PC 上只使用一組），每組匯流排最多可接 32 種裝置，而每種裝置可以有多達 8 項功能。每項功能的編號可以是一組 16-bit 的二進位碼數值，或是由兩個 8-bit 的二進位碼所組成的數值。

每個裝置的硬體電路必須要能回應關於三組定址空間（address space）的查詢，分別是：記憶體位置（memory location）、I/O 埠與組態暫存器（configuration register）。前兩組定址空間由同一 PCI 匯流排上的所有裝置共享，也就是說，當你在存取某記憶體位置（或某 I/O 埠）時，所有的裝置會同時看到該匯流排週期（bus cycle）；而組態空間（configuration space）是依據擴充槽（slot）的位置而來的，每擴充槽都各有一專用的啟用線路（enable wire），供規劃組態時使用。因此，PCI 控制器（PCI controller）可以在某個時間點存取特定的介面卡，而不會產生位址衝突的狀況。每個 PCI 裝置擁有 4 支中斷接腳，主機板的電路會將這些接腳連到處理器的 IRQ 線路。雖然處理器的 IRQ 線路有數量的限制，但由於裝置可共用相同的中斷訊號，因此 PCI 介面卡的數量可以不受 IRQ 線路數量的限制。

在 PCI 匯流排中，由於 I/O 空間採用 32-bit 的位址匯流排，所以可定址的空間高達 4 GB。在記憶體定址空間方面，則採用 32-bit 或 64-bit 方式位址。通常每個裝置都有其獨特的位址，但是也有可能發生兩個裝置有相同的位址，造成兩邊都無法順利運作。所以在規劃組態的時候設定為系統指定位址，就可以重新對應（remap）PCI 裝置所提供的 I/O 與記憶位址區。而在開機時就是用這個機制來對裝置初始化，以避免位址衝突。至於這些區域的位址到底被重新對應到何處，則可以在組態空間內讀到相關資訊，驅動程式只要依據從組態暫存器所取得的資訊，就能安全地存取週邊了。

在 PCI 組態空間中，每個裝置有 256 bytes 的空間來存

放其功能資訊，而且組態暫存器的佈局也有標準規範。每種功能的代碼是 4 bytes，彼此不會重覆，而驅動程式可以利用裝置獨特的代碼來識別裝置。總之，可以用實體的排列方式來找出裝置介面卡，然後再找出對應的組態暫存器，最後再用所取得的資訊來識別 PCI 裝置，這樣即可進行後續的操作。

在系統開機階段 PCI 裝置在此時就已完成組態的設定。當電源開始供應 PCI 裝置時，其實硬體仍處於關閉的狀態，也就是說，裝置只能對組態規劃（configuration transaction）作回應，因此當開啟電源時，裝置沒有任何記憶體或 I/O 埠會對應到電腦的定址空間；而每一種裝置專屬的特性，像是中斷線路，也都是失效的。但每片符合 PCI 規格的 PC 主機板，一定會配備內含 PCI 功能的韌體（firmware），例如 BIOS 等。所以就算處理器的指令集（instruction set）沒有提供存取裝置組態空間的能力時，韌體還是能存取裝置所屬的組態空間。

在系統啟動時，韌體會去規劃每個 PCI 週邊的組態，為裝置所需的任何位址空間配置一個安全的地方。在此之後，每當裝置驅動程式存取裝置，裝置的記憶體與 I/O 位址就已經對應到處理器的定址空間中。驅動程式也可修改預設安排方式。當作業系統(OS)載入 Driver 到記憶體後，驅動程式的初始碼(initial code)會藉由制造商識別碼(Vendor ID)、裝置識別碼(Device ID)來呼叫 PCI BIOS 以發出找尋相關裝置的請求。同時也利用類別碼(Class code)找尋該裝置是否存在(000000h 代表裝置不存在，本例中類別碼 FF0000h 為不符合已定義類別裝置)。若同樣的制造商識別碼(Vendor ID)、裝置識別碼(Device ID)有一個以上時，

PCI BIOS 會將這些匯流排編號（bus number）、裝置編號（device number）、功能編號（function number）傳回驅動程式，而後驅動程式藉由呼叫 PCI BIOS 的服務(service)及編號存取該裝置的組態空間以得知須配置的資源。

本例中 PCI 裝置使用 32 位元 PCI 介面，規劃為被動端(Target)，工作頻率為 33MHz。其開機初始組態暫存器（configuration register）如表 4.1 所示(位址空間全部 256Byte，只有前 64Byte 有被定義使用)。驅動程序為系統讀取裝置組態空間後配置兩塊記憶體空間與一個中斷向量。第一塊記憶體空間為 256Mbyte，主要作為記憶體映射區(存放擷取影像資料)。第二塊記憶體空間為 1Kbyte，規劃作為 PCI 裝置之控制命令暫存區。將配置的第一塊記憶體空間起始位址寫回 Base Address Register 0 (BAR0)組態暫存器(位址 13h~10h)，將第二塊記憶體空間起始位址寫回 Base Address Register 1 (BAR1)組態暫存器(位址 17h~14h)，如此即算完成記憶體映射。另外命令暫存器(Command Register)需填入 0142h (位址 05h~04h)，指定 PCI 裝置功能(only target memory access)，最後再將中斷向量填入 3Ch 位置，如此即算完成 PCI 裝置的驅動。完成驅動後可透過 BAR0 與 BAR1 的存取來控制 PCI 裝置的動作。應用程式將擷取影像寫入 BAR0 後，只需向 BAR1 寫入啟動命令，車道偵測程序即開始進行，之後應用程式監測(讀取)BAR1 回應狀態可知程序執行完成，之後再從 BAR0 中讀回偵測資訊偵測程序即告完成。

表 4.1 PCI 組態暫存器初始狀態列表

pci bus configuration register (開機初始狀態)				
address	Byte			
	3	2	1	0
00H	Device ID		Vendor ID	
	00	04	11	72
04H	Status Register		Command Register	
	04	20	00	00
08H	Class Code			Revision ID
	FF	00	00	01
0CH	BIST	Header Type	Latency Timer	Cache Line Size
	00	00	00	00
10H	Base Address Register 0			
	BB	00	00	00
14H	Base Address Register 1			
	BB	10	00	00
18H	Base Address Register 2			
	00	00	00	00
1CH	Base Address Register 3			
	00	00	00	00
20H	Base Address Register 4			
	00	00	00	00
24H	Base Address Register 5			
	00	00	00	00
28H	Card Bus CIS Pointer			
	00	00	00	00
2CH	Subsystem ID		Subsystem Vendor ID	
	00	00	00	00
30H	Expansion ROM Base Address Register			
	00	00	00	00
34H	Reserved			Capabilities Pointer
	00	00	00	00
38H	Reserved			
	00	00	00	00
3CH	Maximum Latency	Minimum Grant	Interrupt Pin	Interrupt Line
	00	00	01	00

4.3 PCI 裝置 (Altera PCI Development Board)

(一) PCI 裝置 (Altera PCI Development Board)

”PCI Development Board” (如圖 4.4 所示)為 ALTERA 公司所生產，作為開發 PCI 原型裝置的一個平台。支援 5V、3.3V 的雙電壓規格，也支援 32-bit 或 64-bit 的 PCI 插槽規格(一般 PC 準系統為 5V 32bit 插槽)。同時亦支援 33MHz、66MHz、133MHz(PCI-X 介面) 三種工作頻率。

使用 Stratix EP1S25F1020C5 Device 作為使用者規劃設計之 FPGA。同時亦提供 256M-Byte PC333 DDR SDRAM 外部體模組，透過 DDR SDRAM connector (SODIMM)連接到 Stratix EP1S25F1020C5 Device 中的 Bank3、Bank4 接腳。PCI 連接器(俗稱金手指)經過 DC-to-DC 電壓轉換器(Level Conveters)後直接與 Stratix EP1S25F1020C5 Device 的 Bank7、Bank8 接腳作連接，中間並無經過其他邏輯處理。板上 2 個震盪器提供 33MHz 與 100MHz 兩種時脈輸入、分別接到 A19、C19 接腳。另外板上還提供 EPM3256ATC144-7 Device 與 8 M-Byte Flash Memory 作為 FPGA(Stratix EP1S25F1020C5 Device)初始化裝置的 2 個主要元件(在電源供應瞬間 EPM3256ATC144-7 Device 會將存在 Flash Memory 上的 FPGA 程式載入 Stratix EP1S25F1020C5 Device 中)。另外還提供有 8-bit User LEDs 輸出顯示，亦連接至 Stratix EP1S25F1020C5 Device 中的 Bank8。(如圖 4.5 所示)

其他非與本文相關之元件在此不作贅述，請參考文件 PCI Development Kit, Stratix Edition Getting Started User Guide [12] 與 Stratix PCI Development Board Data Sheet [13]。

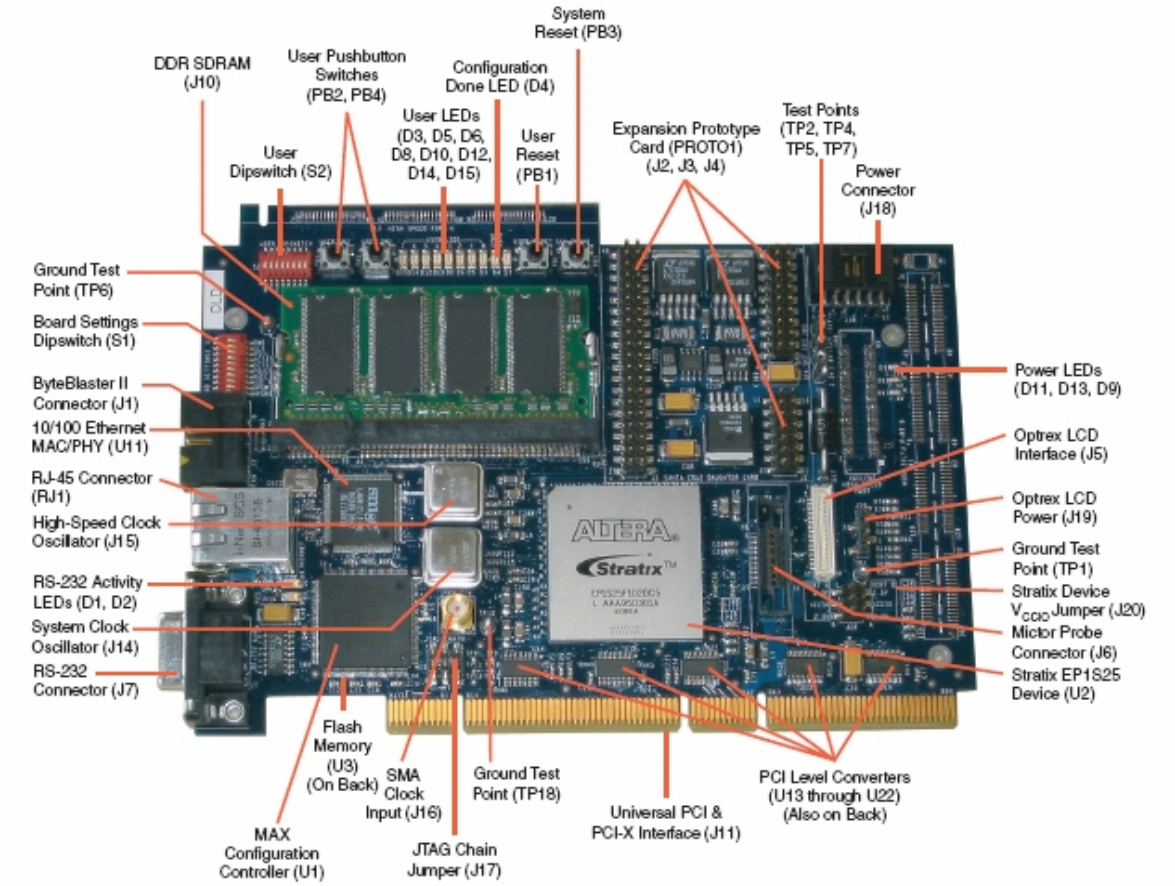


圖 4.4 PCI Development Board 外觀圖

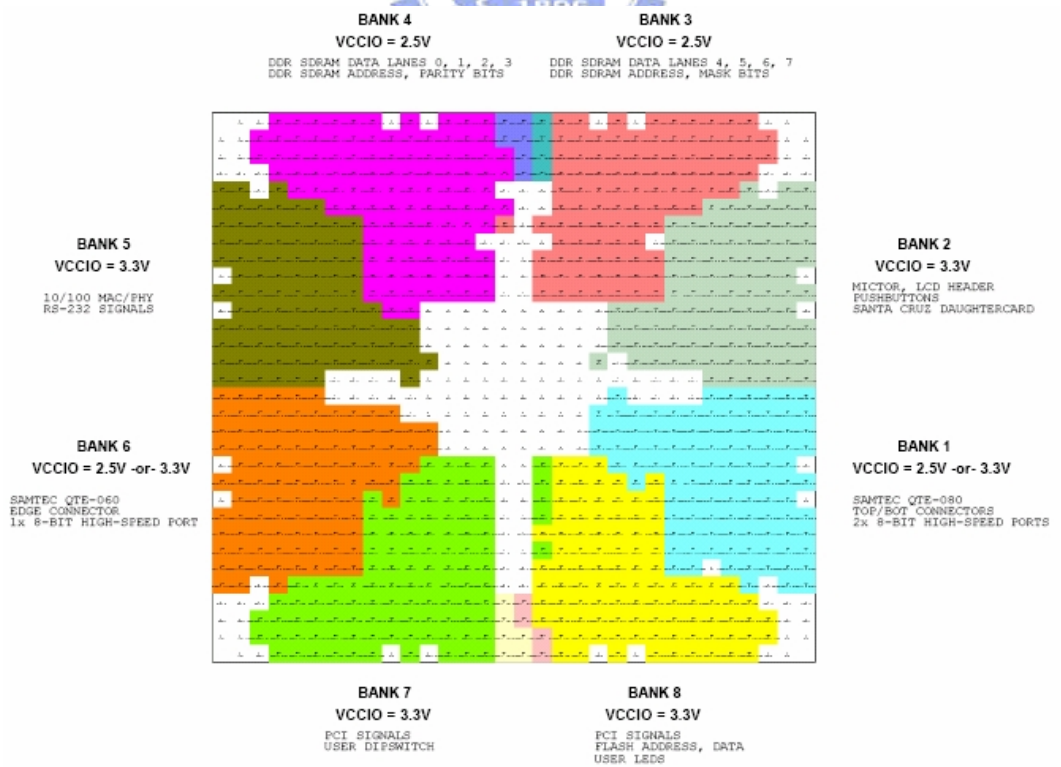


圖 4.5 Stratix Package TOP View

(二) Stratix EP1S25F1020C5 Device 構造

”PCI Development Board”使 Stratix EP1S25F1020C5 Device，Stratix Device 結構方塊圖如圖 4.6 所示。主要結構有 6 種分別為 M-RAM Blocks、M4K RAM Blocks、M512 RAM Blocks、DSP Blocks、I/O Elements、LAB Structure。

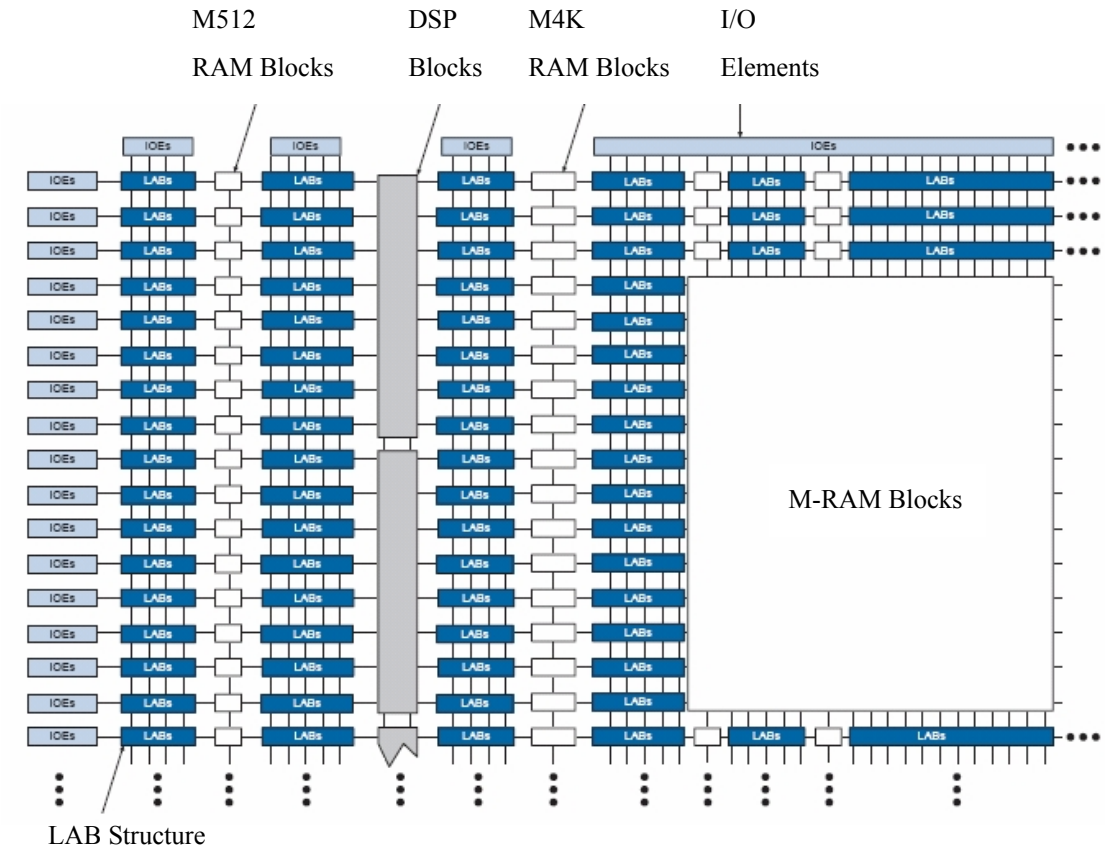


圖 4.6 Stratix Device 結構方塊圖

M512 RAM Blocks(32x18bit) 、M4K RAM Blocks(128x36bit)、M-RAM Blocks(4Kx144bit)為 3 種不同大小的內部記憶體區塊，以因應設計者不同的記憶體需求。DSP Blocks 可以快速執行乘法運算或是 FIR 濾波器運算。I/O Elements 支援 DDR、PCI、GTL+、SSTL-3、SSTL-2、HSTL、LVDS、LVPECL、PCML、HyperTransport 或是其他 I/O 標準的 pin 腳電氣規格與時間規格。

一個 LAB Structure 包含了 10 個邏輯元件(LE)與邏輯元件進位鏈(LE carry chains)，LAB 控制訊號，區域連接線(local interconnect)，LUT 鏈(LUT chain)，以及暫存器鏈連接線(register chain connection lines)。LUT(Look Up Table)為 FPGA 實現邏輯的基本元件，每個 LE 中擁有一個 4 輸入的 LUT。區域連接線連接同 LAB 的 LE 元件，暫存器鏈連接線連接暫存器到相鄰的 LE(在同 LAB 中)的暫存器，LUT 鏈連接 LUT 到相鄰的 LE(在同 LAB 中)的 LUT。LAB 藉由各種連接線即可組成較大的邏輯元件以應付複雜的邏輯實現。詳細的 LE 結構圖如 4.7 所示。

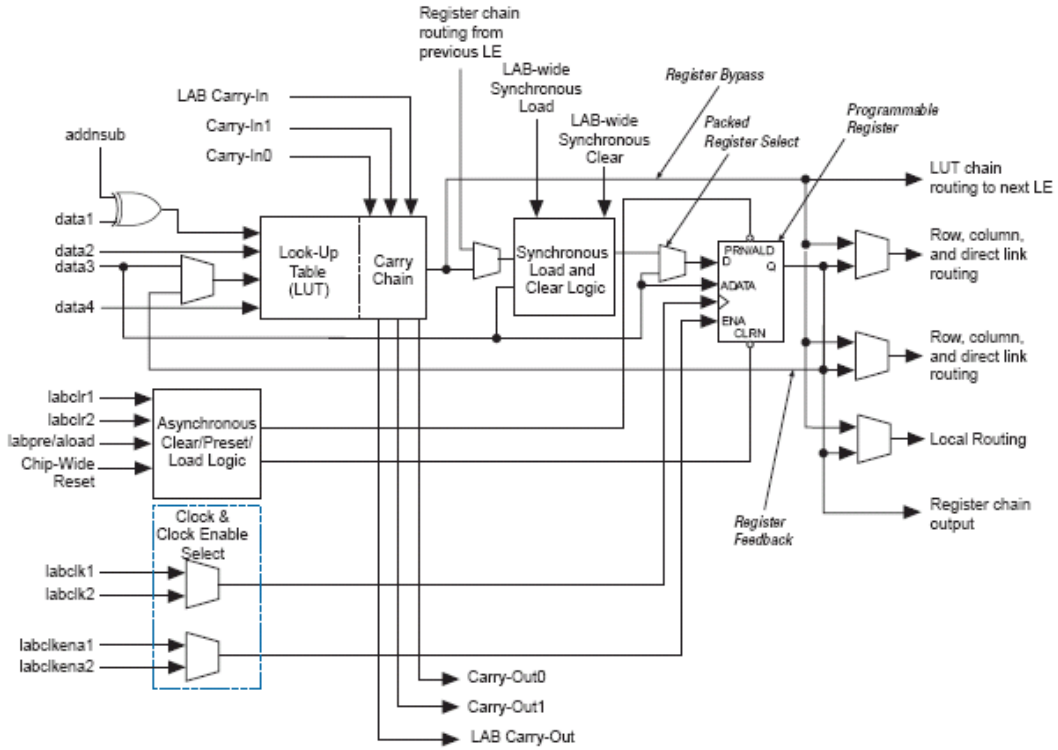


圖 4.7 邏輯元件(LE)結構圖

一個 Stratix Device 基本上就由以上六種結構所組成，不同 Device 之間基本六種組成結構為相同，只是每種結構數量以及密度不同，以及晶片封裝型態上的不同而已。

每個 Stratix EP1S25F1020C5 Device 具有 2566 個 LAB，即 25660 個(約 25K)邏輯元件(LE)、80 個 9-bit 高速乘法器(DSP Blocks)以及 10 個 FIR 濾波器(DSP Blocks)、2 個 M-RAM Blocks、138 個 M4K RAM Blocks、224 個 M512 RAM Blocks、內部記憶體總計為 1944576bit(237K-Byte)，如表 4.2 所示。另外附加 6 個相鎖迴路(PLL)，此包裝最大可使用的 I/O 接腳為 706 隻接腳。對於更詳盡的 Stratix EP1S25F1020C5 Device 資料，請參考”Stratix Device Handbook, Volume 1”[14]、”Stratix Device Handbook, Volume 2”[15]文件。

表 4.2 Stratix EP1S25F1020C5 Device 資源表

EP1S25F1020C5 Device	數量
邏輯元件(LE)	25660 個
DSP 9-bit 乘法器	80 個
DSP FIR 濾波器	10 個
M-RAM Blocks (4Kx144bit)	2 個
M4K RAM Blocks (128x36bit)	138 個
M512 RAM Blocks (32x18bit)	224 個
記憶體總計(M-RAM、M4K、M512K)	237K-Byte

(三) ALTERA PCI MegaCore Function (IP CORE)

ALTERA 公司提供了兩樣 MegaCore Function – PCI Compiler 與 DDR SDRAM Controller 供發展 PCI Development Board 原型裝置之設計者所使用，藉以縮短原型裝置的開發與驗證時間(MegaCore Function 為 ALTERA 公司針對旗下 FPGA 所開發的 IP CORE 產品)。本文中使用 PCI Compiler 4.0 版本與 DDR SDRAM Controller 1.2.0 版本。為配合 MegaCore Function 版本使用的發展工具環境為 Quartus II 5.0 與 ModelSim SE 5.7d。

ALTERA 公司所提供的 PCI Compiler MegaCore Function 完全符合 PCI Special Interest Group (PCI-SIG)制定的”PCI Local Bus Specifiation,Revision 3.0 ”[11]文件所定義的規格要求，且支援 66MHz 工作頻率，對於開發 PCI 介面提供很好的解決辦法，省去許多 PCI 介面 timing 與 functional 上的驗證。另外亦提供行為模型(Behavioral models) 與 Verilog、 VHDL 兩種版本的 testbench 供第三者模擬工具 (third-party simulation tools) 模擬與驗證所使用(本文模擬使用 ModelSim)。PCI MegaCore functions 包含了 32/64 bit、master/target 與 target only 的版本供使用者選擇。分別對應為 pci_mt64、pci_t64、pci_mt32、pci_t32 等 4 個獨立 IP CORE。每個 MegaCore functions 跟據 Device 不同提供三種版本 Full、Preliminary、No support。Full 版本保證 timing 與 functional 都能達到規格要求，Preliminary 版本僅保證 functional 達到規格要求，timing 上還需設計者作 re-timing 自行達到規格要求。對於 Stratix EP1S25F1020C5 Device 而言，PCI Compiler 3.0 以下版本僅提供 Preliminary 版本，於今年初推出的 PCI Compiler 4.0 才能支援 Full 版本。

1. PCI_T32 MegaCore functions

在本例的應用中主要使用 pci_t32 MegaCore function，其方塊圖如圖 4.8 所示。使用 pci_t32 需花費 659 個邏輯元件(LE)去實現，使用 PIN 腳數為 48 個(PCI Target 接腳)，且 FloorPlan 位置要求為 bank7、bank8 鄰近位置。pci_t32 MegaCore function 參數規劃如下：CLASS_CODE=X"FF0000"、DEVICE_ID=X"0004"、REVISION_ID =X"01"、SUBSYSTEM_ID=X"0000" 、 SUBSYSTEM_VENDOR_ID=X"0000" 、 TARGET_DEVICE=STRING、VENDOR_ID=X"1172"、MIN_GRANT=X"00"、MAX_LATENCY=X"00" 、 CAP_PTR=X"40" 、 CIS_PTR=X"00000000" 、 BAR0=X"FFFC0000" 、 BAR1=X"FFFFFC00" 、 NUMBER_OF_BARS=X"00000002" 、 EXP_ROM_BAR=X"FFF00000"、PCI_66MHZ_CAPABLE=STRING、INTERRUPT_PIN_REG=X"01"、ENABLE_BITS=X"00000003"。

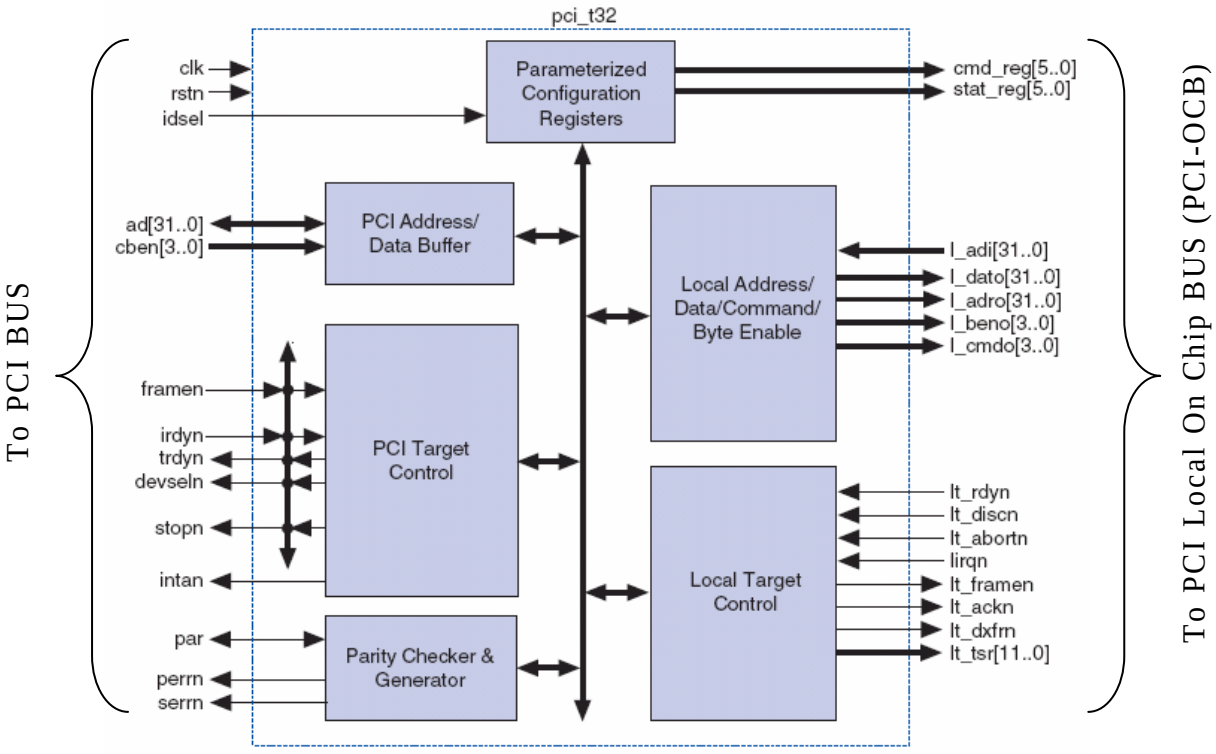


圖 4.8 pci_t32 MegaCore function 方塊圖

2. PCI 匯流排訊號功能定義

clk：PCI BUS 端的 clock 輸入訊號，亦作為全系統 Clock 來源。除非同步訊號外，系統所有訊號皆在 clk 正緣取樣。輸入端必須保證 clk 為乾淨無彈跳訊號。

rstn：PCI BUS 端的非同步 reset 訊號，亦作為全系統 reset 來源 (命名字尾小寫 n 代表負邏輯動作)。

idsel：PCI BUS 端的初始化裝置(device)選擇訊號，使用於組態暫存器（configuration register）的傳輸上。系統內每個 slot 都有其獨立 idsel 硬體接線。

AD[31:0]：PCI BUS 端的位址與資料雙向多工匯流排，在第一個偵測到 framen=〝0〞的 clk 正緣時，AD[31:0]代表位址線(此時 MegaCore function 會將其 Latch 住送到 l_adro 上)。其餘時間被則視為資料線，因為 AD[31:0]為雙向多工通道，所以在每次使用完後需一段反轉週期，以確保不會有兩邊同時驅動造成邏輯短路情況。

cben[3:0]：PCI BUS 端的命令與位元組致能雙向多工匯流排，如同 AD[31:0]在第一個偵測到 framen=〝0〞的 clk 正緣時代表匯流排命令，其餘時間則代表位元組致能，cben[0]代表最低位元組(LSB)， cben[3]代表最高位元組(MSB)，cben[x]=〝0〞對應的位元才是真正傳輸的資料(命名字尾小寫 n 代表負邏輯動作)。匯流排命令如表 4.3 所示。

par : PCI BUS 端的同位元檢查訊號。PCI BUS 使用〝偶同位〞方式檢查碼，檢查 AD[31:0]&cben[3:0]上傳送之資料。

framen : PCI BUS 端的傳輸(欄位)控制訊號。由 PCI BUS Master 控制，framen 由〝1〞轉變到〝0〞表示要求資料交易，維持在〝0〞狀態表示資料持續收送當中，由〝0〞轉變到〝1〞代表為最後一筆交易資料。(以上皆在 CLK 正緣取樣判斷)

irdyn : PCI BUS 主控端(Master/Initiator) ready 訊號，irdyn=〝0〞代表主控端準備好收送訊號。

itdyn : PCI BUS 被控端(Target) ready 訊號，itdyn=〝0〞代表被控端準備好收送訊號。

stopn : PCI BUS 被控端(Target) stop 訊號，stopn =〝0〞代表被控端要求停止目前傳輸週期。

intan : PCI BUS 端中斷訊號(Interrupt A)，intan =〝0〞代表裝置端請求中斷。

perrn : PCI BUS 端同位元檢查錯誤訊號，perrn =〝0〞代表裝置端上筆傳輸資料同位元檢查錯誤(上一個 CLK 正緣傳輸資料)。

serrn : PCI BUS 端系統錯誤訊號，serrn =〝0〞代表裝置端發生系統錯誤或是位址、同位元錯誤錯誤。

表 4.3 PCI 匯流排命令對照表

Cben[3:0]	命令型態	Cben[3:0]	命令型態
0000	Interrupt Acknowledge	1000	Reserved
0001	Special Cycle	1001	Reserved
0010	I/O Read	1010	Configuration Read
0011	I/O Write	1011	Configuration Write
0100	Reserved	1100	Memory Read Multiple
0101	Reserved	1101	Dual address Cycle
0110	Memory Read	1110	Memory Read Line
0111	Memory Write	1111	Memory Write and Invalidate

3. PCI Local On Chip BUS (PCI-OCB)端訊號功能定義

cmd_reg[6:0] : PCI-OCB 端命令暫存器，對應到 PCI 組態暫存器中的命令暫存器狀態。

stat_reg[6:0] : PCI-OCB 端狀態暫存器，對應到 PCI 組態暫存器中的狀態暫存器狀態。

l_adi[31:0] : PCI-OCB 端位址與資料多工輸入匯流排，PCI 讀取命令狀態下對應到 PCI BUS 端的 ad[31:0]。Target Controller 必須負責在位址週期內送入位址訊號，在資料週期中送入資料訊號。

l_adro[31:0] : PCI-OCB 端位址輸出訊號，PCI 寫入命令狀態下的位址週期中 MegaCore function 會將 ad[31:0] 訊號 Latch 到 l_adro 上。

l_dato[31:0] : PCI-OCB 端資料輸出訊號，PCI 寫入命令狀態下的資料週期中 MegaCore function 會將 ad[31:0] 位址 Latch 到 l_adro 上。

l_beno[3:0] : PCI-OCB 端位元組致能輸出訊號，如同 l_adro[31:0]，在 PCI 寫入命令狀態下的位址週期中 MegaCore function 會將位元組致能訊號(cben[3:0]) 對應到 l_beno 上。

l_cmdo[3:0] : PCI-OCB 端匯流排命令輸出訊號，如同 l_dato[31:0]，在 PCI 寫入命令狀態下的資料週期中 MegaCore function 會將匯流排命令訊號(cben[3:0]) 對應到 l_cmdo 上。

lt_rdyn：PCI-OCB 端 Target ready 訊號，對應到 PCI BUS 端 itdyn 訊號。lt_rdyn= 〝0〞 代表 Target Controller 準備好收送訊號。

lt_abort : PCI-OCB 端 Target abort reguest 訊號，lt_abort = 〝0〞 代表 Target Controller 宣告此次傳輸失敗。

lt_discn：PCI-OCB 端脫離(disconnect)或是重試(retry)訊號，lt_discn= 〝0〞、lt_rdyn = 〝0〞 代表 Target Controller 宣告脫離連線。lt_discn=〝0〞、lt_rdyn = 〝1〞 代表 Target Controller 忙碌中要求稍後連

線。依”PCI Local Bus Specifiation,Revision 3.0 ”[11]規格標準，當傳輸超過硬體位址空間時設備端要提出 disconnect 要求。

lirgn：PCI-OCB 端請求中斷訊號，對應到 PCI BUS 端 intan 訊號，lirgn=〝0〞代表 Target Controller 端請求中斷。

lt_framen: PCI-OCB 端傳輸(欄位)控制訊號，對應到 PCI BUS 端 framen 訊號。

lt_ackn: PCI-OCB 端 acknowledge 訊號，lt_ackn=〝0〞表示 MegaCore function 認可交易資料為有效。

lt_dxfm: PCI-OCB 端資料傳輸中訊號，lt_dxfm=〝0〞表示 MegaCore function 資料傳輸中。

lt_tsr[11:0]: PCI-OCB 端傳輸狀態暫存器訊號，lt_tsr[5:0] 分別對應 BAR5~0，若 lt_tsr[0]=〝0〞表示傳輸位址落在 BAR0 範圍中。lt_tsr[8]=〝0〞表示 Target 存取要求，lt_tsr[9]=〝0〞表示 burst 存取模式，lt_tsr[10]=〝0〞表示前筆資料傳輸成功結束。

4.PCI_T32 功能規格(Function Specification)

PCI_T32 功能簡單的來說就是一種橋接器(bridge),主要功能就是將雙向多工的 PCI 匯流排訊號，處理成單向(輸入或輸出)訊號，並提供雙向多工安全的反轉週期，以及控制 timing 上的保障。其 PCI Local On Chip BUS (PCI-OCB)端的訊號功能大致上與 PCI 規格一致，只是在時序有些延遲(delay 1~3 clock 週期)。以下就本文應用所需功能如記憶體讀寫、組態空間讀寫作簡單說明，詳細規格說明請參照”PCI Compiler Data Sheet ”[16]、”PCI MegaCore Function User Guide ”[17]。表 4.4 為寫入組態空間動作時的組態位址 ad[31:0]各訊號功能定義、圖 4.9、4.10 說明 pci_t32 在執行組態空間讀寫時的動作時序、圖 4.11、4.12 說明 pci_t32 在執行記憶體讀寫時的動作時序。

表 4.4 組態位址訊號功能定義

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	保留(全部為〝0〞)							匯流排號碼								裝置號碼					Fun_num			Dword						00	

AD[1:0]=〝00〞：表示使用 TYPE 0 類型主態交易。

(TYPE 1 交易型態在 PCI 2.2 板之後時已經廢除。)

AD[7:2]：空間位址共有 256 byte，由 AD[7:2]定址。

AD[10: 8]：PCI Target 裝置之功能號碼參數。

AD[15:11]：PCI Target 裝置之裝置號碼參數。

AD[23:16]：PCI Target 裝置之匯流排號碼參數。

在 PCI 的規格中，允許系統最多可接 256 組匯流排，每組匯流排最多可接 32 種裝置，每種裝置可以有多達 8 項功能。

AD[31]=〝0〞 表示為組態存取(AD[31]=〝1〞為 I/O 存取)。

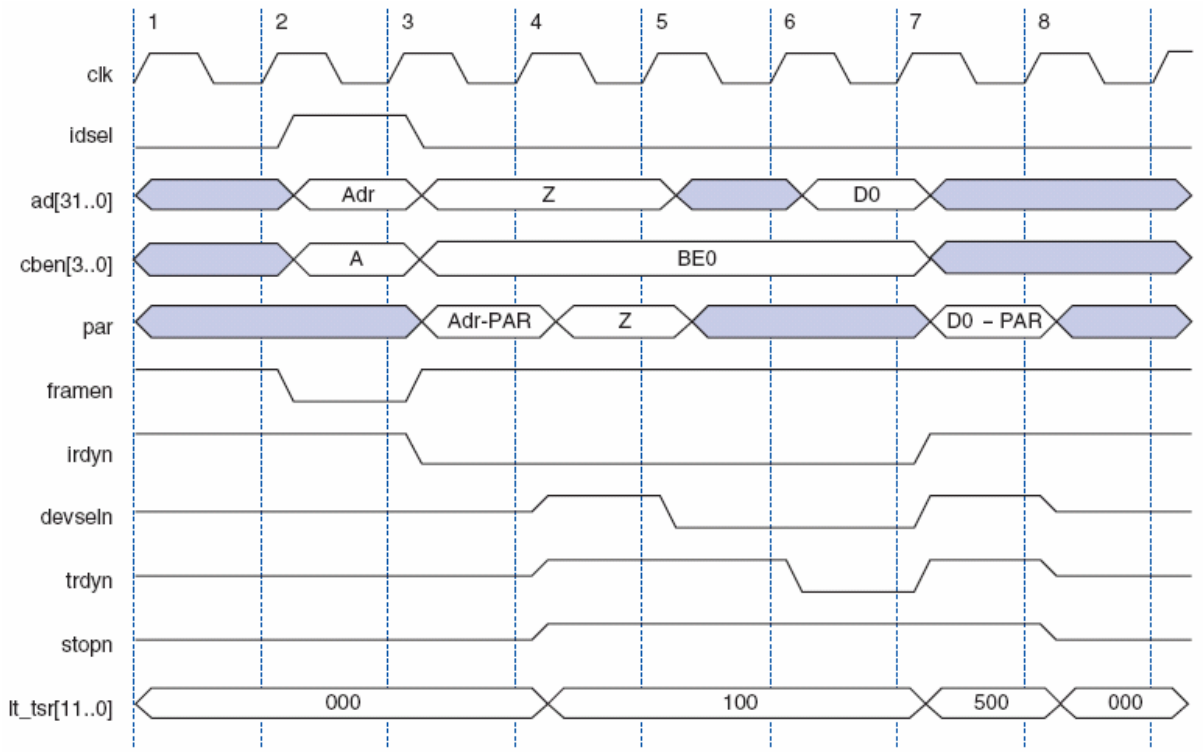


圖 4.9 pci_t32 組態讀取時序圖

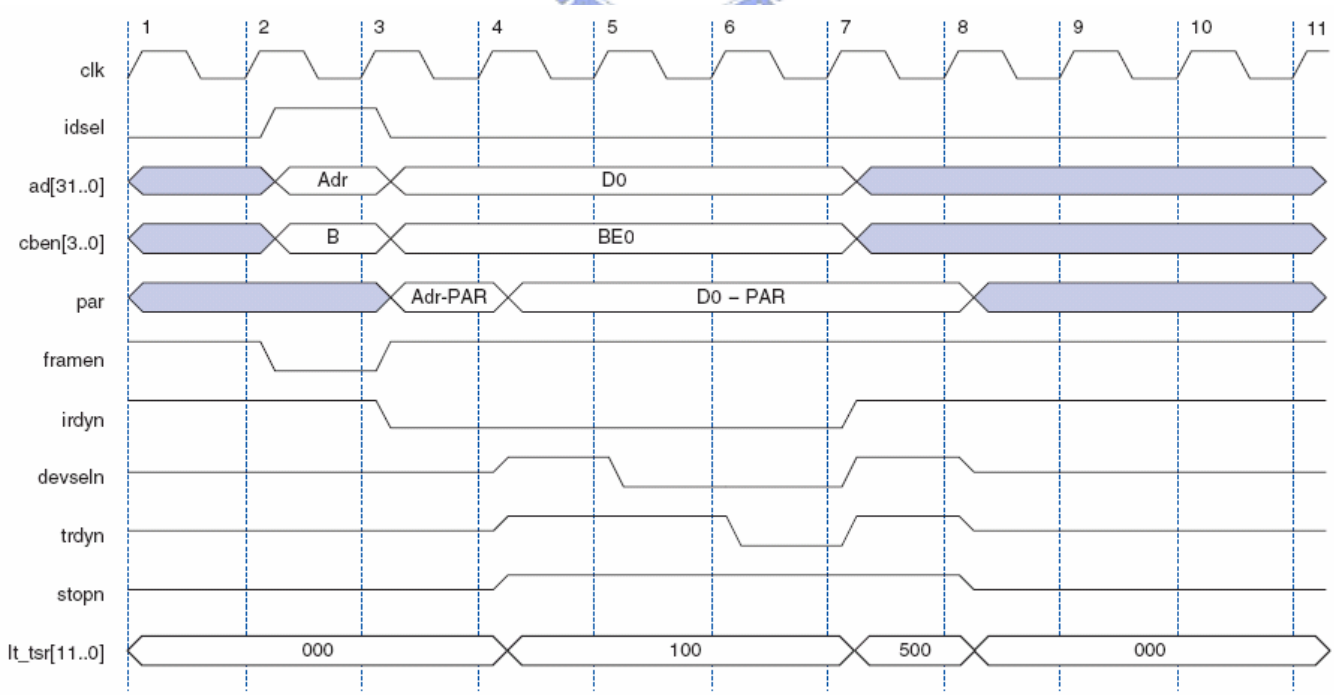


圖 4.10 pci_t32 組態寫入時序圖

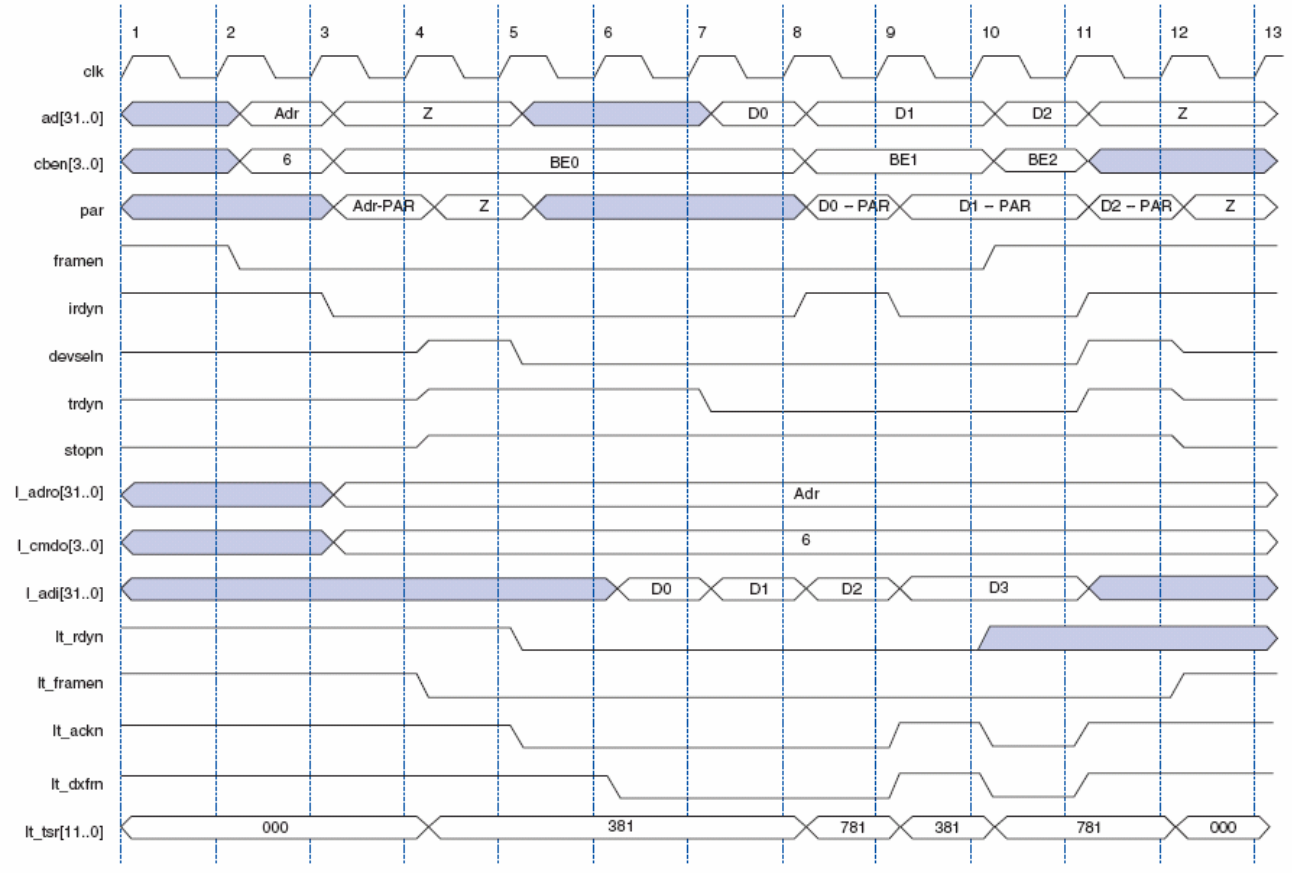


圖 4.11 pci_t32 記憶體讀取時序圖

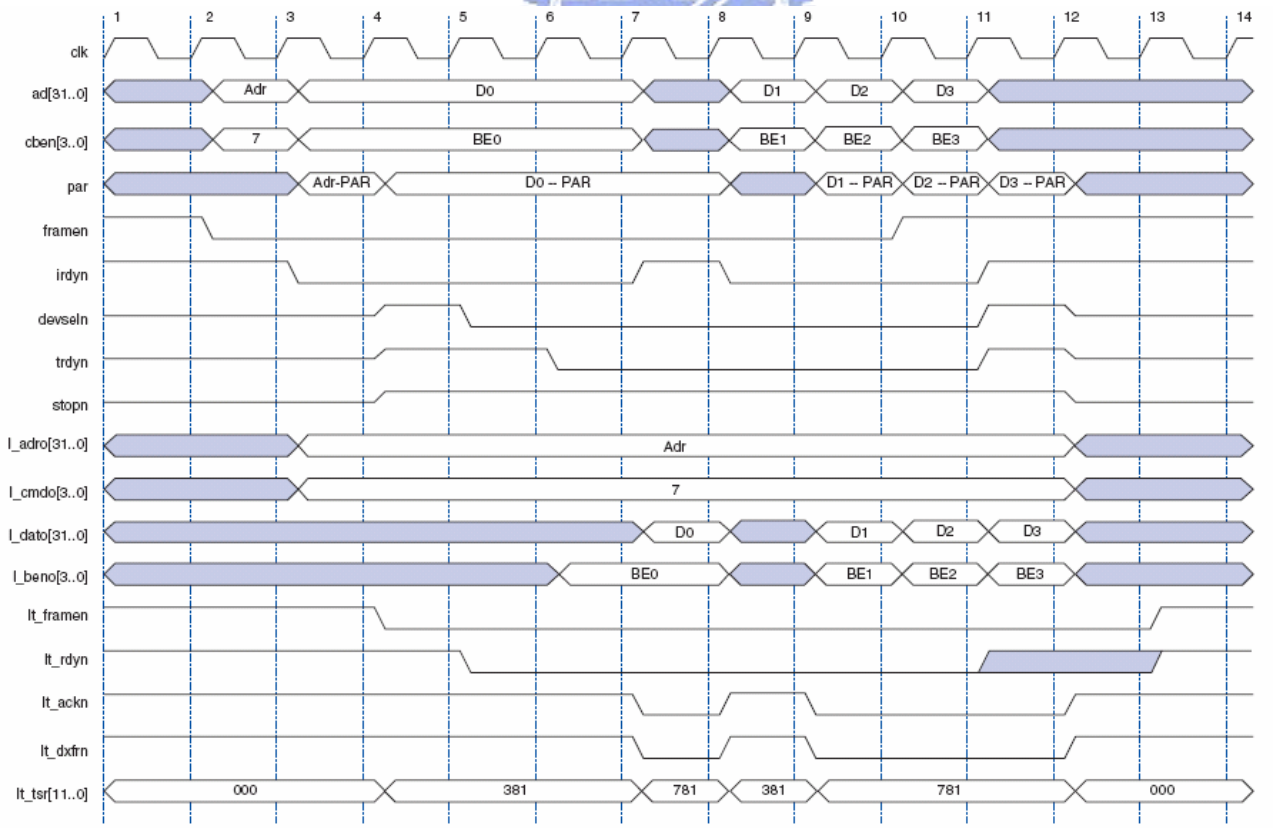


圖 4.12 pci_t32 記憶體寫入時序圖

（四） DDR SDRAM Controller IP CORE

ALTERA 公司所提供的 DDR SDRAM Controller MegaCore Function 管理著複雜的 DDR SDRAM 的控制如記憶體模組初始化、SDRAM banks 的管理、自動 refresh 功能等等，且提供了易整合的數位介面，可以幫助使用者避掉設計記憶體時序控制上非同步電路的困擾，使用這些 IP 可以讓設計者更專注在核心元件的設計上。MegaCore functions 跟據 Device 不同提供三種版本 Full、Preliminary、No support。對於 Stratix EP1S25F1020C5 Device 而言，DDR SDRAM Controller 1.2.0 版僅提供 Preliminary 版本(僅保證 functional 達到規格要求，timing 上還需設計者作 re-timing 自行達到規格要求)，目前尚無支援 Full 的版本。

在本文的應用中 DDR SDRAM Controller MegaCore Function 參數規劃如下：Data_width_bits=16、Row_address_bits=13、Column_address_bits=12、Number_of_bank=4、Number_of_chip_selects=1、Memory_Size=256MB、Burst_length=8、System_Clock=133MHz。實現 MegaCore Function 需要花費 1020 個邏輯元件(LE)，且 FloorPlan 位置要求為 bank3、bank4 鄰近位置。另外 PC333 DDR SDRAM 記憶體模組需要操作在 133MHz 的工作頻率上與系統的 33MHz 不同，所以必須再加上相鎖迴路電路(PLL)解決同步化之問題後 DDR SDRAM Controller MegaCore Function 才可以正常工作。實現後方塊圖如圖 4.13、4.14、4.15、4.16、4.17 所示。

DDR SDRAM Controller MegaCore Function 詳細規格說明請參照 ”DDR SDRAM Controller MegaCore Function User Guide ” [18]。

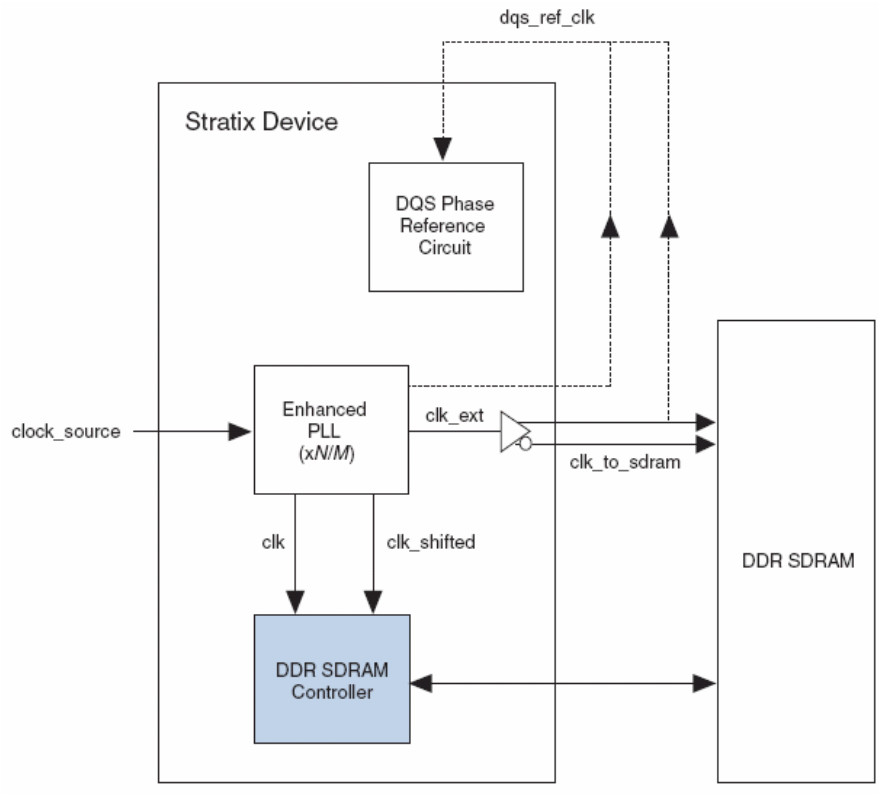


圖 4.13 相鎖迴路(PLL)方塊圖

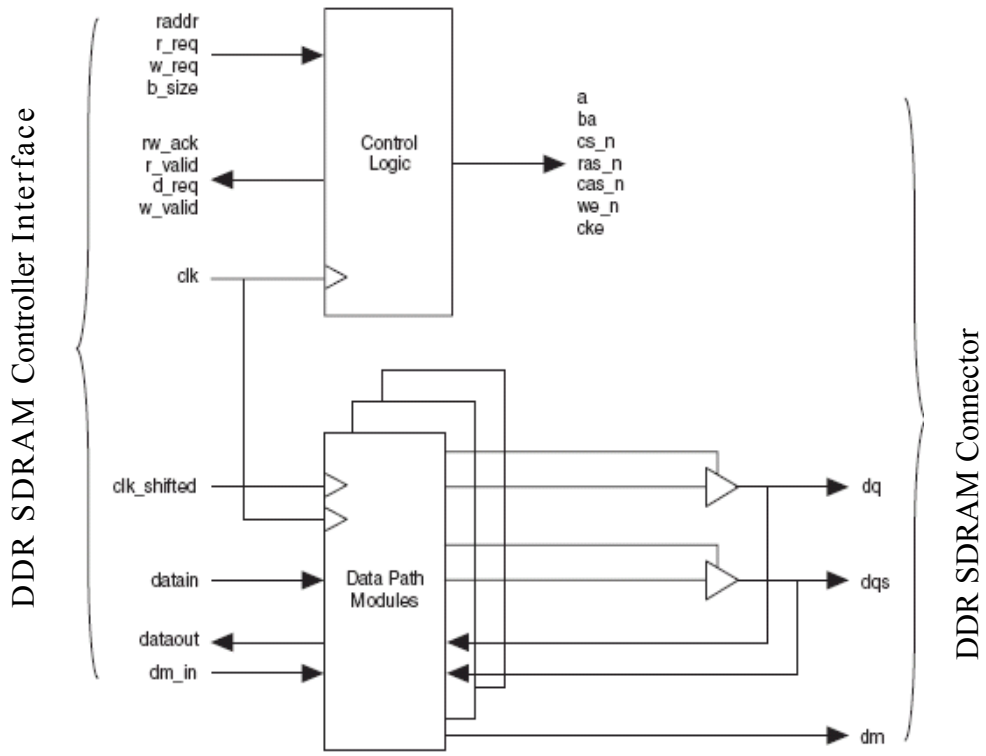


圖 4.14 DDR SDRAM Controller 方塊圖

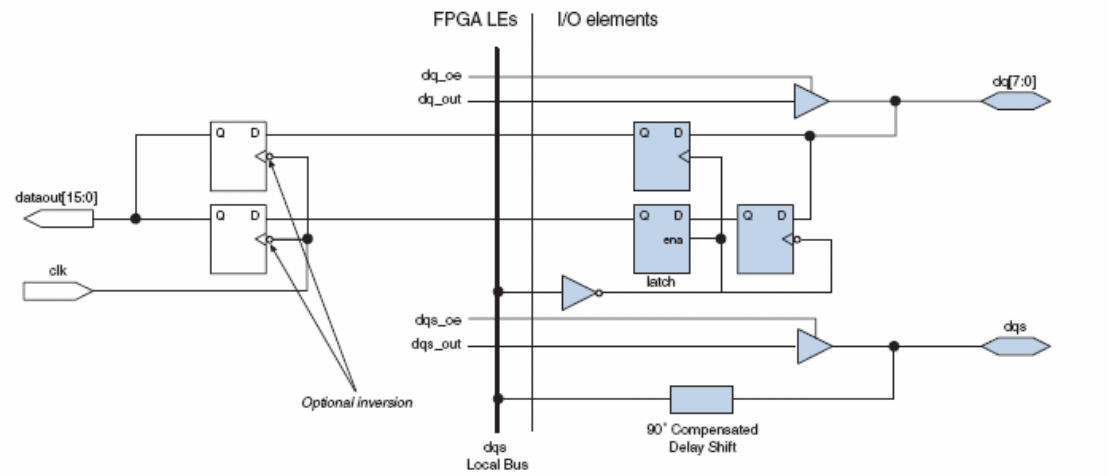


圖 4.15 Data Path Module (Read) 方塊圖

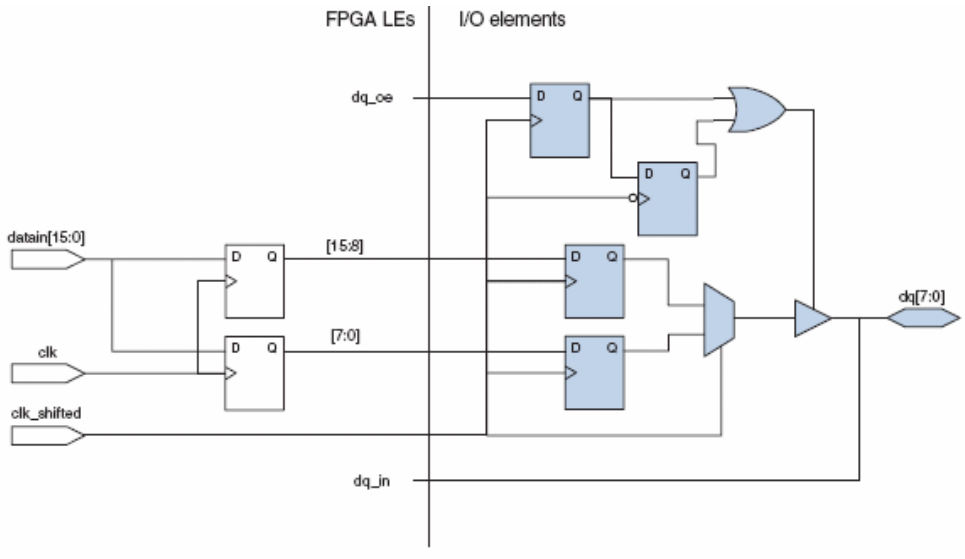


圖 4.16 Data Path Module (Write) 方塊圖

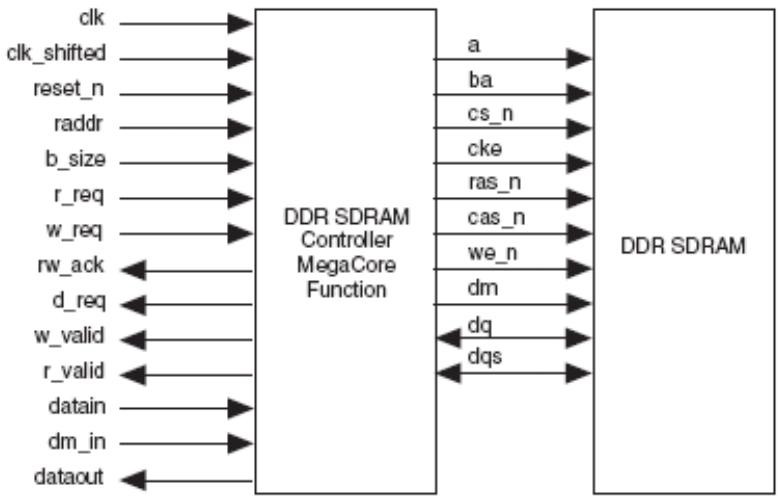


圖 4.17 DDR SDRAM Controller 系統方塊圖

1. DDR SDRAM Controller Interface 端訊號功能定義

本文中利用 DDR SDRAM Controller Interface 模組作為 DDR SDRAM Controller MegaCore Function 與 Memory Interface 間的橋接器，接面訊號功能定義如下：

clk ： 系統 clock(33MHz)。

clk_shifted ： 相位領前 clk 90°之時脈訊號。

reset_n ：系統 reset 訊號(字尾小寫 n 代表負邏輯動作)。

raddr[26:0] ： 位址匯流排。

b_size[2:0] ： burst 傳輸長度訊號，DDR SDRAM Controller 數位端介面仍以 burst 方式傳輸。

r_req ： 讀取要求訊號(Read request)。r_req=〝1〞表示資料已從 DDR SDRAD 讀取到 MegaCore Function 中，向 DDR SDRAM Controller Interface 端提出讀取要求。

w_req ： 寫入要求訊號(Write request)。w_req=〝1〞表示 DDR SDRAM Controller Interface 端，向 MegaCore Function 提出寫入 DDR SDRAD 的要求。

rw_ack： 傳輸允許訊號，rw_ack=〝1〞表示 MegaCore Function 接受讀寫要求中。

d_req： 請求資料訊號。d_req=〝1〞表示 MegaCore Function

要求 DDR SDRAM Controller Interface 端在下個 clk 正緣前將寫入資料準備好在 datain 上。

r_valid：讀取資料正確訊號(read data valid)。r_valid=〝1〞表示 MegaCore Function 保證目前 dataout 上之資料訊號為正確有根據的，DDR SDRAM Controller Interface 端可安心讀取。

w_valid：寫入資料正確訊號(write data valid)。w_valid=〝1〞表示 MegaCore Function 將目前 datain 上資料寫入 DDR SDRAM 中，DDR SDRAM Controller Interface 端此時必須保證 datain 上之資料訊號為正確有根據的。

datain[31:0]：資料匯流排(write)。

dataout[31:0]：資料匯流排(read)。

dm_in[3:0]：位元組致能，dm_in[0]代表最低位元組(LSB)，dm_in[3]代表最高位元組(MSB)。dm_in[x]=〝0〞對應的位元才是真正傳輸的資料。

2. DDR SDRAM Connector (SODIMM)端訊號功能定義

DDR SDRAM Connector (SODIMM)，其介面訊號功能定義如下：

a[12:0] : 位址匯流排。

ba[1:0] : Bank 位址訊號。

cs_n : chip 選擇訊號。

cke : clock enable 訊號。

[ras_n、cas_n、we_n] : SDRAM 控制訊號。

dq[15:0] : 資料匯流排。

dm[1:0] : 位元組致能。

dqs : data strobe 訊號。

3. DDR SDRAM Controller 功能規格

DDR SDRAM Controller 對於 DDR SDRAM Controller Interface 端而言是一項被動模組，只在接受讀取命令時才動作，寫入、讀取時序圖如圖 4.18、4.19 所示。詳細規格請參照 ”DDR SDRAM Controller MegaCore Function User Guide ” [18]文件說明。

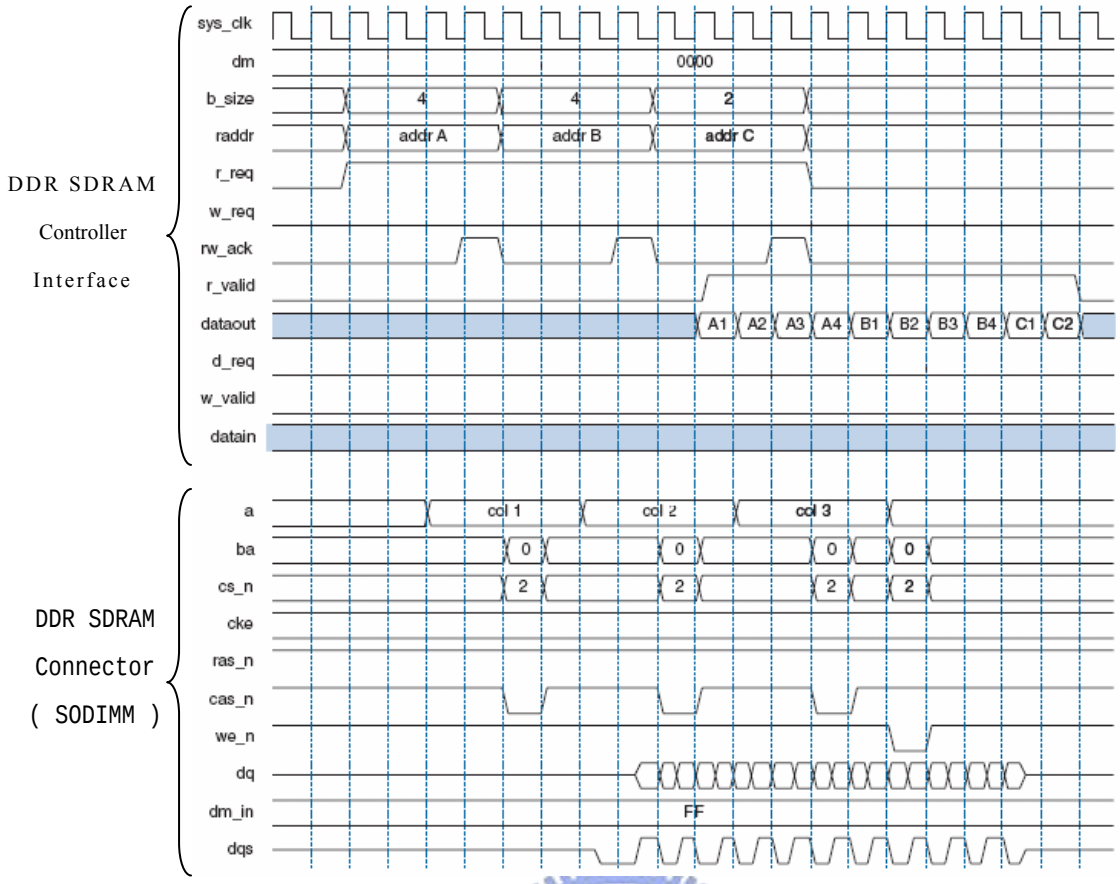


圖 4.18 DDR SDRAM Controller 寫入時序圖

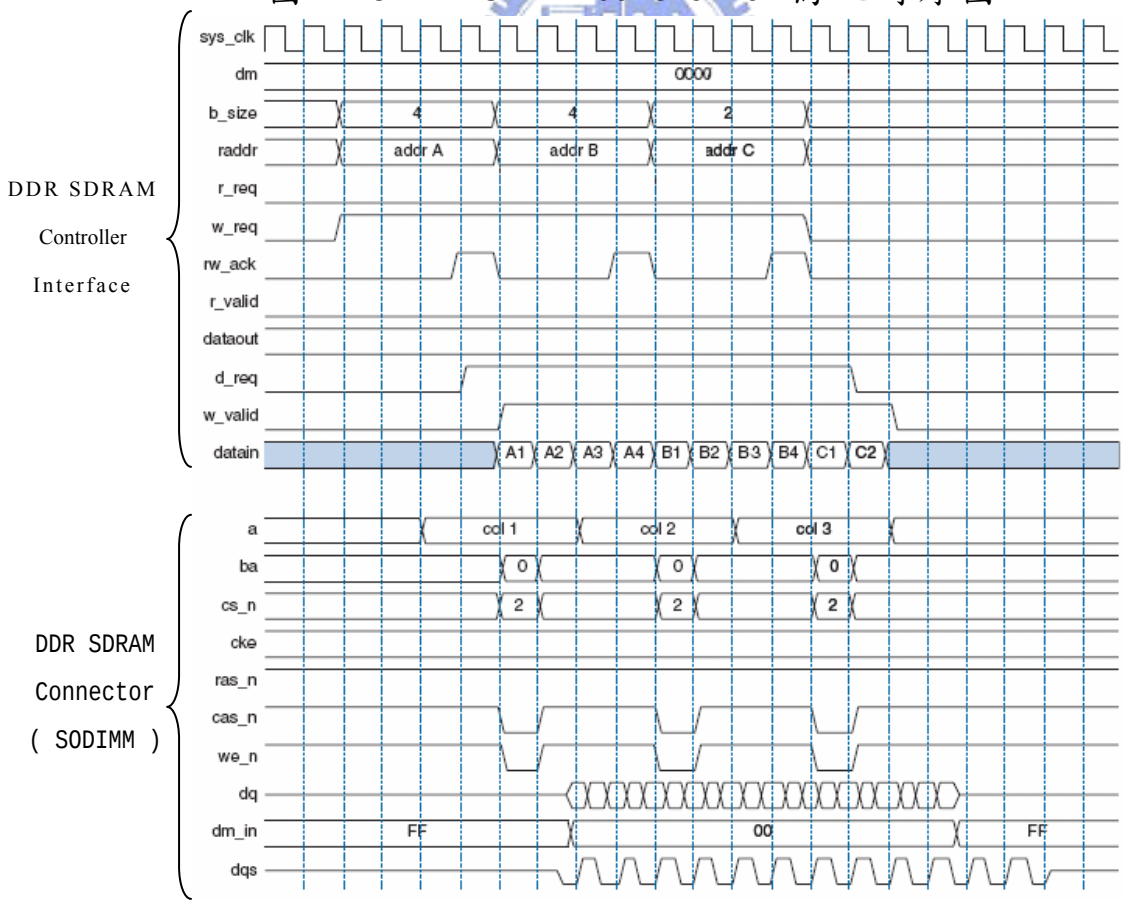


圖 4.19 DDR SDRAM Controller 讀取時序圖

（五） 記憶體計畫 (Memory Plan)

道路偵測演算法所需影像資訊為 644x210，所以只需將所需影像傳入 PCI Development Board 之中。儲存影像資訊所需記憶體為 644x210x8 bit 約 132k-Byte，即若不使用 pipeline 方式處理影像讀取與處理(pipeline 方式需要 2 塊影像儲存空間)，雖可直接將影像資訊全部使用內部記憶體儲存，但仍不建議如此為之。因 FPGA 一般只是作為發原型版所使用的臨時環境而已，最終仍須將 FPGA 已 ASIC 晶片方式產品化，此時過大的內部記憶體空間將會佔據 ASIC 晶片的大部分面積，導致 die 過大不利生產。若單純就 FPGA 設計上而言亦不建議這麼作，132k-Byte 佔據全部記憶體空間的 55.6%，再考慮佈線花費掉的空間後，實際所佔據的資源會膨脹很多。再加上還需考慮 wire delay 等問題，種種因素都可能讓 Synthesis、Place and Route(P&R)最後無法收斂。基於以上原因，規劃使用外部記憶體(DDR SDRAM)作為影像儲存之空間。在不讓外部記憶體存取延遲到道路偵測處理時間下，使用最小限度的內部記憶體暫存影像資料。

由於道路偵測處理器(Lane Detector processor)依影像 u 軸作切割，每次只需擷取一列影像資料，處理完後只會向前遞增或向後遞減。由此特性非常容易掌控道路偵測處理器所需影像資料範圍，設計專用的 DMA 控制器(DMA Control Logic)處理內外部記憶體間的映射。另外使用環型結構方式設計內部記憶體，使得在道路偵測處理器對記憶體讀取的定址上完全不用改變，仍以 132k-Byte 內部記憶體空間作定址(16-bit 定址)。定址訊號為 mem_adr[15:0]。

環型記憶體結構由 4 塊 2k-Byte(512x32 bit)內部記憶模組所構成，每塊由 9-bit 方式定址(000~1FF)。每塊記憶體模組直接由 mem_adr[8:0]定址，而記憶體區塊由 mem_adr[10:9]選擇，mem_adr[15:11]忽略不管即造成環型定址結構，環狀記憶體位址與區塊對照圖如圖 4.20 所示。

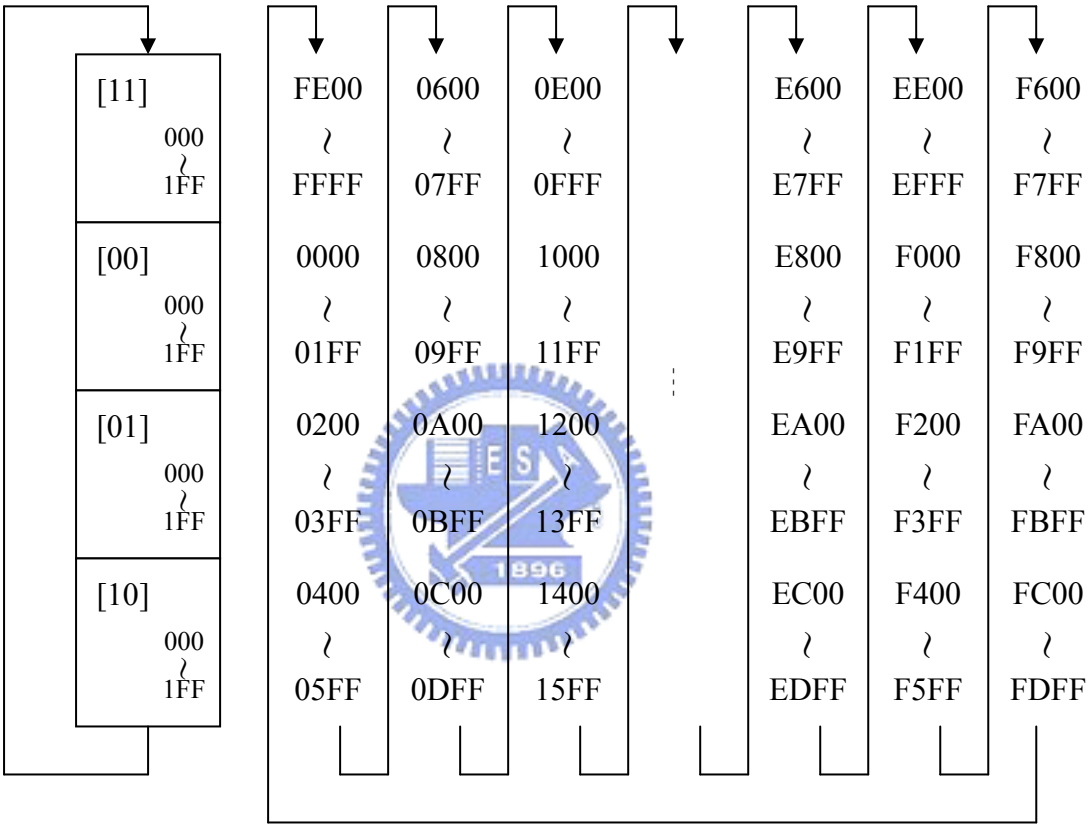


圖 4.20 環狀記憶體位址與區塊對照圖

DMA Control Logic 則隨時監控 mem_adr[15:9]與遞增/遞減旗標 index_dir 處理內外部記憶體間映射，如圖 4.21 所示。若使用 pipeline 方式處理影像讀取與處理，則需 2 塊影像儲存空間(外部記憶體)作運算替換，DMA Control Logic 亦會選擇正確的儲存空間作映射。

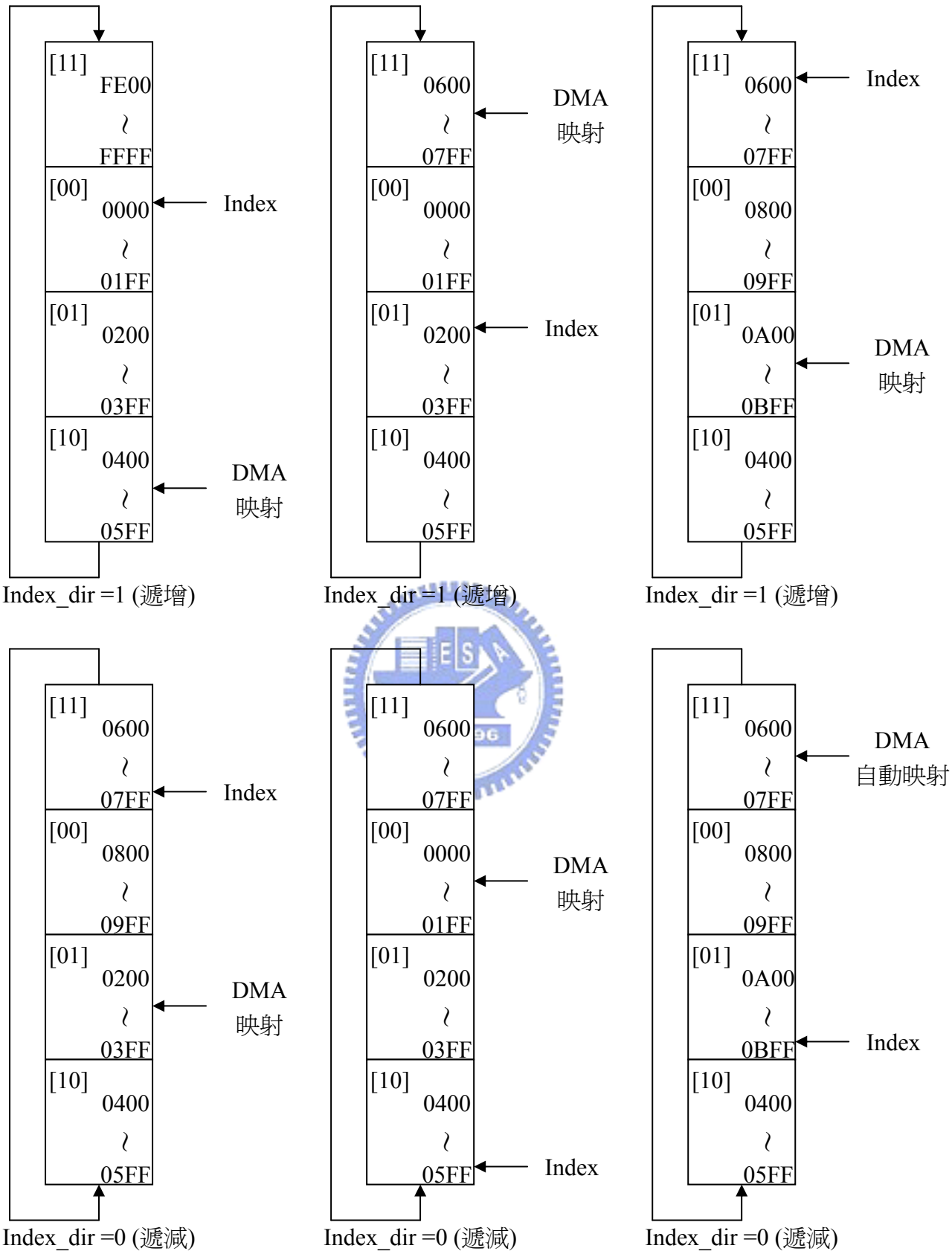


圖 4.21 DMA Control Logic 映射關係圖

（六） 有限狀態機 (Finite State Machine)

Top Level 的 Finite State Machine 控制 PCI 讀寫、道路偵測、等 Top Level 的執行流程，狀態流程圖如圖 4.21 所示。Target Control Logic 隨時監視 PCI-OCB bus 狀態，將 PCI 事件回傳給狀態機判斷。狀態機 Idle 狀態接受 BAR0、BAR1 的讀寫要求，若 PCI 位址錯誤提出 PCI disconnect。在 PCI-BAR1 寫入命令執行結束後，馬上執行道路偵測程序。道路偵測程序執行時只允許 PCI BAR1 讀取命令、與 BAR0 讀寫命令(會與目前使用影像儲存空間錯開、為因應 pipeline 而設計)、其他 PCI 命令回覆 retry，待程序結束後返回 idle 狀態。等待 PC 準系統端讀取偵測結果。

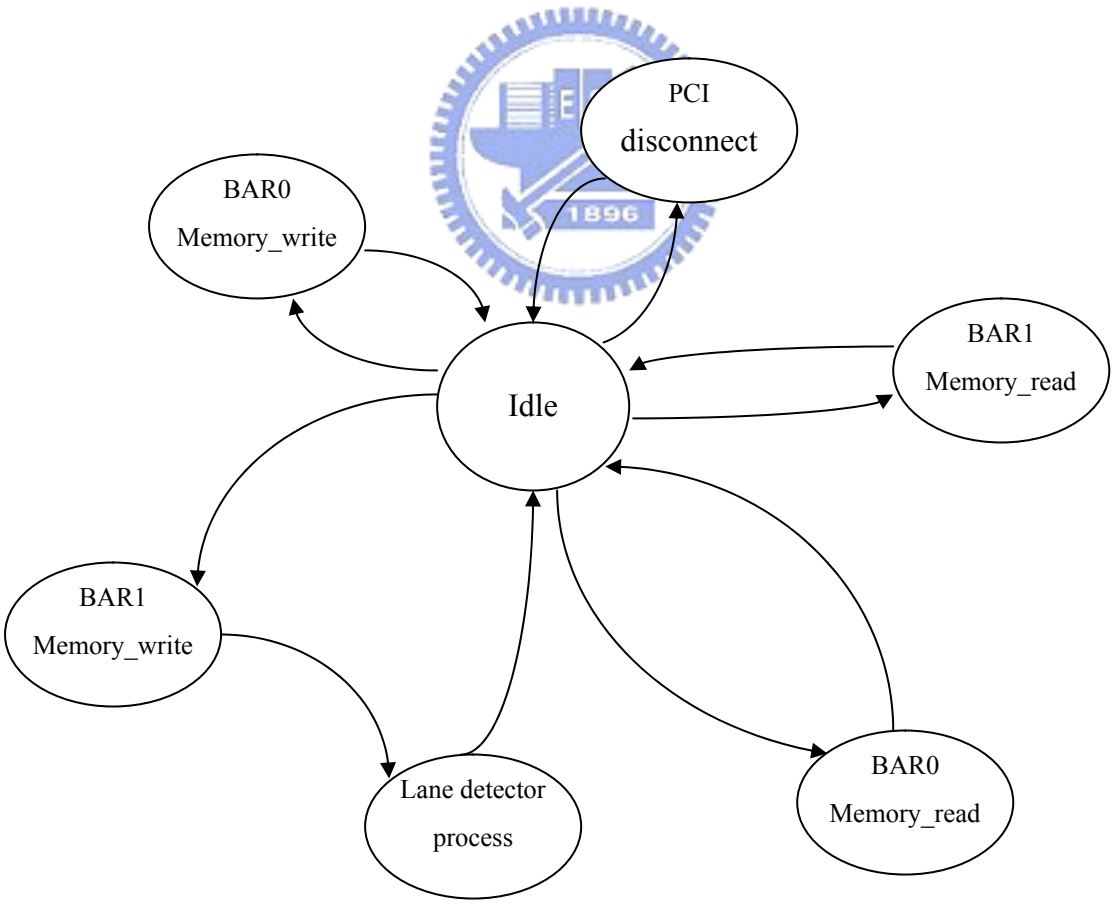


圖 4.22 Top Level Finite State Machine 狀態流程圖

圖 4.23 系統 pipeline 流程圖

第五章　實驗環境與結果分析

5.1　實驗環境介紹

為了驗證道路偵測系統的效能與可靠度，將所架構的硬體系統連上 PC 端進行實測。在實驗中使用儲存於 PC 硬碟中預先拍攝的道路影像作為輸入，此影像是固定於車輛前方之 CCD 攝影機所錄製的道路影像(Sample rate = 30 frame/sec)。由於前端影像擷取部分架構在 PC 平台上，若 PC 系統滿足 30 frame/sec 的影像輸入要求，對處理器而言 PC 端使用預先錄製的車道影像或是即時的車道影像並無差異。

而 PC 端設備在本系統中規劃為用戶端，對於道路偵測系統而言，PC 端只負責 PCI 裝置的驅動及讀取工作如道路影像寫入及讀出偵測結果等，全部的運算皆在 PCI 裝置上執行。因此 PC 端設備效能可依客制化需求而定，在本實驗中使用 Intel Pentium CPU 2.4GHz、256MB 的 RAM 搭配 Windowns XP 作業系統作為 PC 端平台。使用 Visual C++開發一個簡單的測試程式，執行 PCI 裝置驅動程序、寫入偵測影像程序、下達偵測命令、以及讀出偵測結果等程序，並保證每秒寫入 30 張偵測影像。

由於車道影像屬於連續影像，車道資訊亦屬連續，因此可容許少數幾張影像偵測失敗，藉由先前偵測成功的資訊來估測道路變化趨勢。以連續 10 張影像偵測失敗來計算(即遺失 0.3 秒車道資訊)，在時速 100km/hr 下移動了 9.25 公尺，10 公尺內的車道變化趨勢仍前次車道資訊的可信任的範圍內。因此以 10 張影像間格時間作為警告標準，若連續 10 張

影像偵測失敗，則實驗系統發出警告訊號。

車道偵測演算法經硬體系統運算輸出範例如圖 5.1 所示，對於各種路況如左彎道、右彎道、上下坡道、或是部份標線受前方障礙物(汽車、機車等)遮蔽等，皆可有效偵測出車道資訊。

圖 5.1 硬體系統偵測範例

5.2 實驗結果分析

車道偵測演算法經硬體系統運算結果與軟體模擬時幾乎相同。因為在設計初期已考慮量化問題，將量化誤差控制在合理範圍。由於車道偵測演算法主要依據車道標線來做預估道路變化趨勢，若前方障礙物遮蔽到過多車道標線，則會造成預估趨勢偏移(如圖 5.2 所示)甚至造成偵測失敗。

圖 5.2 前方障礙物遮蔽車道標線範例

以道路標線作為偵測依據的先天障礙，使得道路偵測系統無法使用於擁擠的路況中。而快速道路或高速公路車速較快的緣故，車與車間保持一段安全距離，去除了前方障礙物

的限制後，車道偵測成功率就大大的提高了。因此車道偵測系統用於發展高速、長途的自動駕駛上就顯的非常適合。

設計車道偵測處理器最大的挑戰來自於三次方程式的最小平方近似法實現上，經過運算最佳化設計後，使得原本與階數 n 呈現指數成長關係的運算複雜度，成功的降為線性關係，在 2n+37 個 clock 週期內計算完成，大大的縮短了處理器的處理時間及晶片面積。

目前車道偵測處理器的處理能力為每秒 697 張影像，平均每張影像處理時間為 45000 個 clock 週期(在 33MHz 下約等於 1.43ms)，為系統所要求的 23 倍，在實際使用上系統閒置時間太多，仍可在晶片設計的架構方面使用更多的共用結構，以節省面積。由於處理能力充足，將可再增加道路偵測初始 ROI 樣板，更多的 ROI 樣板可使估測曲線更貼近實際曲線，對於偵測的失敗率上可以更降低一些。此外亦可加入省電功能電路，如待機與喚醒電路，在系統閒置時將所有晶片功能關閉，只留下少許監控電路，在系統需求運算時再將晶片運算功能喚醒，以達到待機省電功能。

本例中道路偵測處理器部份只使用到 10 個浮點數乘法器、9 個元浮點數加(減)法器以及 1 個浮點數除法器，每個浮點數乘法器使用 276 個邏輯元件(LE)、3 個 DSP 乘法器(9-bit element)來實現，每個浮點數除法器使用 1252 個邏輯元件(LE)來實現，每個浮點數加(減)法器使用 712 個邏輯元件(LE)來實現。所有運算單元皆由 Finite State Machine 配置給各電路共同使用，以最少的運算單元滿足所有運算元件，大大節省了處

理器使用面積。但是過多的結構共用運算單元，所帶來的繞線問題仍是產生許多面積的浪費，亦造成 P&R 不易收斂等問題。要解決此問題最好使用 SISC (Special-Instruction-set Single Chip)的架構(SISC 為設計 DSP PROCESS 常用架構)。使用 Special Instruction Set(特殊指令集)代替 Finite State Machine 複雜的繞線，作為運算單元分配的控制元件。但此舉會增加額外的暫存器存取時間(存取暫存器需多花一個 clock 週期)，由於目前處理能力為系統所要求的 23 倍，在節省面積的考量下，使用 SISC 降低部分處理速度(2 倍以內)仍在可允許的範圍內的。實現車道偵測系統所需資源如表 5.1 所示。

表 5.1 車道偵測系統使用資源表

EP1S25F1020C5 Device	總數量	使用數量	使用百分比
邏輯元件(LE)	25660 個	17130 個	66%
DSP 9-bit 乘法器	80 個	80 個	100%
DSP FIR 濾波器	10 個	0 個	0%
M-RAM Blocks (4Kx144bit)	2 個	0 個	0%
M4K RAM Blocks (128x36bit)	138 個	7 個	5%
M512 RAM Blocks (32x18bit)	224 個	39 個	17%
記憶體總計(M-RAM、M4K、M512K)	237K-Byte	6.68K-Byte	2.82%

第六章　結論與未來展望

道路偵測演算法使用道路標線作為偵測依據，優點為特徵偵測簡單、運算量少，容易實現在晶片上，其缺點是容易受障礙物影響，造成資訊不足而偵測失敗。此一特點在應用面上可多加思索，以自動駕駛為例，在高速長途單調的駕駛車輛亦使人類發生疲憊、精神渙散，容易產生意外。但對於道路偵測處理系統而言，高速道路上標線規律整齊，且前後車有一段安全距離，不會有障礙物遮蔽標線情形，偵測率將近 99%，而且駕駛方向固定，非常適合此一應用。反觀市區道路路況複雜擁擠，容易發生標線遮蔽問題，不利道路偵測系統運作，但市區道路叉路眾多，旅行方向不固定，本身對於自動駕駛的需求不大。

對於道路偵測演算法上仍有許多改善空間，本文是基於道路標線作車道曲線的預估，若前方障礙物遮蔽過多標線可能會導致偵測失敗，在無標線之道路上(如十字路口或是鄉間小路)亦無法使用。可能還必須找出其他特徵判斷(使用較抽象的影像特徵、如單一色調區域)，再配合障礙物偵測資訊，作為判斷前方道路。此外亦可加入消失點資訊，作為影像搜尋範圍的依據。若能使用數種道路特徵判斷車道曲線加以多數決，將可得到更穩健的車道資訊。

在後端道路資訊的應用上，亦可加入條件判斷如多項式參數的變化率等，合理化取用偵測結果，以避免取用到錯誤的偵測資訊。由於擷取影像的 Sample Rate=30 frame/sec，在時速 100 km/hr 的狀態下，每張影像間格只前進約 1 公尺

範圍，道路趨勢變化不大，且 30 frame/sec 對於車輛控制而言是非常快速變動的訊號，因此偵測資訊可經過 Low pass filter 後再行輸出，避免輸出產生高頻跳動。

參考文獻

[1] Alberto Broggi, “Robust real-time lane and road detection in critical shadow conditions”, Proceedings of IEEE International Symposium, Computer Vision, Coral Gables, FL, 353-358, Nov. 21-23, 1995.

[2] Gang Yi Jiang, Tae Young Choi, Suk Kyo Hong, Jae Wook Bae and Byung Suk Song, “Lane and obstacle detection based on fast inverse perspective mapping algorithm”, IEEE International Conference, Systems, Man, and Cybernetics, Nashville, TN, vol.4, 2969 - 2974, 8-11 Oct. 2000.

[3] K. Kluge and S. Lakshmanan, “A Deformable-Template Approach to Lane Detection”, Proceedings of IEEE International Symposium,, Intelligent Vehicles, Detroit, MI, 54 - 59, 25-26 Sept. 1995

[4] C. Kreucher and S. Lakshmanan, “LANA: A Lane Extraction Algorithm that Uses Frequency Domain Features”, IEEE Transactions, Robotics and Automation, vol. 15, 343 - 350, April 1999.

[5] Y. Wang, E. K. Teoh, and D. Shen, “Lane Detection Using B-Snake”, Proceedings of International Conference, Information Intelligence and Systems, Bethesda, MD, 438 - 443, 31 Oct.-3 Nov. 1999

[6] J. Goldbeck and B. Huertgen, “Lane Detection and Tracking by Video Sensors”, Proceedings of IEEE/IEEJ/JSAI International Conference, Intelligent Transportation Systems, Tokyo, 74 - 79, 5-8 Oct. 1999.

[7] R. Chapuis, R. Aufrere, and F. Chausse, “Accurate Road following and Reconstruction by Computer Vision”, IEEE Transactions, Intelligent Transportation Systems, vol. 3, 261 - 270, Dec. 2002.

[8] A. Takahashi, Y. Ninomiya, M. Ohta, and K. Tange, “A Robust Lane Detection using Real-time Voting Processor”, Proceedings of IEEE/IEEJ/JSAI International Conference, Intelligent Transportation Systems, Tokyo, 577 - 580, 5-8 Oct. 1999.

[9] A. Broggi, M. Bertozzi and A. Fascioli, “ARGO and the MilleMiglia in Automatico Tour”, IEEE International Conference, Intelligent Systems and Their Applications, Honolulu, Hawaii, USA, 518-523, Jan.-Feb. 1999.

[10] Bing-Fei Wu, Chao-Jung Chen, Chung-Cheng Chiu, and Tze-Chiuan Lai, “A real-time robust lane detection approach for autonomous vehicle environment”, Proceedings of the IASTED Conference, Signal and Image Processing, 23-25 August 2004.

[11] PCI Special Interest Group,PCI Local Bus Specifiation, Revision 3.0, February 1999. http://www.pcisig.com

[12] Altera Corporation,PCI Development Kit, Stratix Edition Getting Started User Guide, First publication, May 2003. http://www.altera.com

[13] Altera Corporation ,Stratix PCI Development Board Data Sheet, ver. 2.0, September 2003. http://www.altera.com

[14] Altera Corporation,Stratix Device Handbook - Volume 1, v3.1, September 2004. http://www.altera.com

[15] Altera Corporation,Stratix Device Handbook - Volume 2, v3.1, September 2004. http://www.altera.com

[16] Altera Corporation,PCI Compiler Data Sheet, ver. 1.2, February 2003. http://www.altera.com

[17] Altera Corporation,PCI MegaCore Function User Guide, v4.0.0, April 2005. http://www.altera.com

[18] Altera Corporation,DDR SDRAM Controller MegaCore Function User Guide,v1.3 , March 2003. http://www.altera.com

[19] Altera Corporation,PCI Testbench User Guide,v.1.2 ,February 2003. http://www.altera.com

[20] Altera Corporation,AN 169 Simulating the PCI MegaCore Function Behavioral Models, v1.2, February 2003 http://www.altera.com

[21] Altera Corporation,FS 12 pci_mt32 MegaCore Function Reference Design,v1.1, February 2001 http://www.altera.com

[22] Altera Corporation,Introduction to Quartus II ,Version 5.0, April 2005 http://www.altera.com

[23] Altera Corporation,Quartus II Development Software Handbook v5.0 (Complete Four-Volume Set),V1.5.0,April 2005 http://www.altera.com