

第二章 指紋辨識演算法與 VLSI 設計

第一節 指紋特性與辨識步驟

(一)指紋特性

指紋是人類與生俱來的身體特徵，大約在 14 歲以後，每個人的指紋就已經定型。指紋具有「固定性」，不會因人的繼續成長而改變，指紋同時也具有「唯一性」，不同的兩個人不會具有相同指紋。由於指紋容易取得，並具備「固定」與「唯一」的特性，早在西元約一百年前就被成功的用來作為犯罪偵察的工具。典型的指紋可以分為 Arch, Loop, Whorl 等三種基本類型，如圖 1 所示[4]。

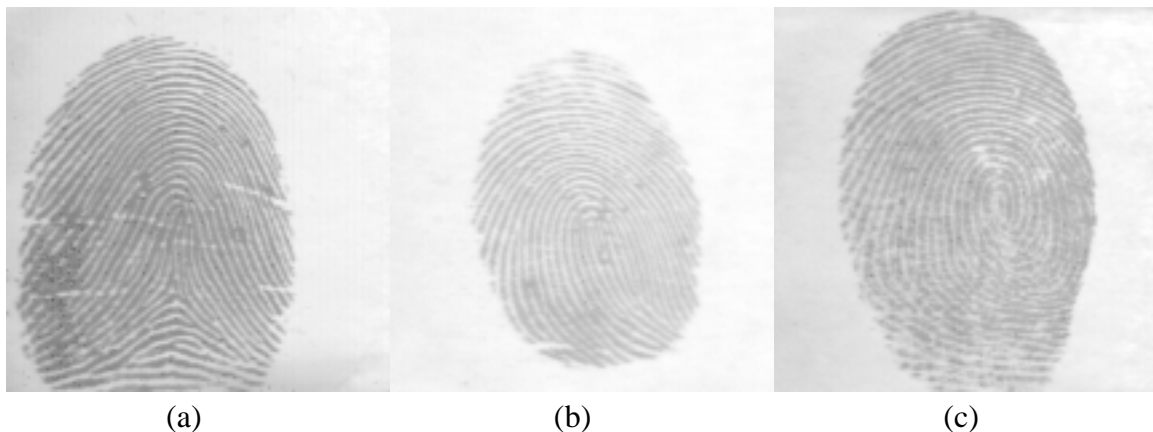


圖 1 典型指紋的三種基本類型

(a)為 Arch 帳型

(b)為 Loop 箕型

(c)為 Whorl 螺旋型

指紋的脊脈(ridge)與走向(orientation)構成指紋影像的特徵要素，在指紋辨識處理的過程中，利用複雜的影像處理方法，設法精確的檢測這些特徵並加以比對。目前指紋辨識的方法大多使用指紋脊脈的端點(ending)與分叉點(bifurcation)作為特徵點依據，並以特徵點對特徵點的方式進行比對。

(二)指紋辨識步驟

指紋辨識的步驟可以分成「指紋註冊(Fingerprint Registration)」與「指紋辨識

(Fingerprint Recognition)」兩個階段。在指紋註冊階段所登錄註冊的指紋圖像若愈清楚，對於後續指紋辨識階段之辨識率提高，將會有明顯幫助；反之，則會有明顯不利的影響，嚴重者甚至無法進行辨識。

(1)指紋註冊階段

個人將指紋資料向辨識單位註冊登錄後，以供往後身份識別的依據。註冊過程中，先記錄個的身份人基本資料後，依下列「指紋辨識」階段的步驟 1)、 2)與 3)將指紋特徵資料存檔，用以提供下述步驟 4)使用。

(2)指紋辨識階段

1)指紋影像取樣(Acquisition)

指紋的獲取必須強調是以活體指紋即時(real time)取像，這樣才能避免有心人士偽造，而且指紋影像的取像設備，必須價格便宜以及讓使用者方便使用。

2)指紋影像前處理(Preprocessing)

前處理的內容包括數位轉換(transformation)、影像切割(segmentation)、雜訊消除(noise removal)、影像調整(adjustment)、影像增強(enhancement)、影像平滑(smoothing)、正規化(normalization)、二值化(binarization)等，讓指紋影像進行辨識前，能去除一些不利於辨識的因素，並將欲辨識之指紋圖像分離出來。

3)指紋影像特徵編碼(Feature encoding)

一般特徵編碼是將欲辨識之指紋影像圖案或已具有分類之特徵，透過特定演算法，將這些具有相同特徵的資訊，編輯成可代表原先影像的符號、字元、數字或位元等。例如文字辨認中，利用線段的分佈當作文字辨認的特徵。特徵編碼的方法有：線萃取(line extraction)、區域萃取(region extraction)、斜轉換(slant transform)、紋理(texture)等。

通常灰階指紋影像需經過二值化(binarization) 細線化(thinning)等步驟才可進行特徵萃取(feature extraction),而指紋影像的原始資料量約從 128x128 至 512x512 個像素(pixels)不等，本步驟由於計算量大，故較耗費運算時間，若採用 586 等級的 CPU(或以上)或專用晶片(ASIC 或 FPGA)，才有機會達到即時計算的要求。

(4)指紋特徵比對(Feature matching)

特徵編碼後的結果，必須與影像資料庫中的資料進行比對，找出最接近之特徵影像，比對方法有利用相合(match)、距離(distance)、識別函數(identify function)、特徵值比對等方法。而指紋比對又可分為兩種方式：(1)指紋辨別(fingerprint verification)如果使用者被要求鍵入使用者代號(ID)，則可根據使用者代號自資料庫中調出預存的特徵檔直接進行一對一比對即可；(2)指紋識別(fingerprint identification)若使用者不須鍵入 ID 號碼或其它密碼，則資料庫中的特徵檔必須逐一被調出來比對，即所謂一對多比對。

第二節 VLSI 晶片設計之重要考慮因素

數位電路設計是 VLSI 電路設計上不可缺少的一部份。由於數位電路具有高性能、高穩定度、抗雜訊能力強、精確度高、易於模組化設計與易於集成的特性。使得數位電子技術在各個應用領域上，都擔任著重要的角色。在 VLSI 設計中必須考慮的重要因素，分為兩個部份的說明[5][6][7]。

(一)設計工具(design tools)的選用

現代科技日新月異，使得產品週期的更迭速度愈來愈快，加之以電腦科技的快速發展，使得藉助電腦的快速計算與資料處理能力來做為設計上的輔助工具已成為工程設計上的潮流。而在電子電路的設計上的輔助工具稱之為電子電路設計自動化(Electronics Design Automation, EDA)。也有將自動測試也加入在這一門學科中，稱之為電子電路設計自動化與自動測試(Electronics Design Automation and Test, EDA&T)。

所謂「工欲善其事，必先利其器」，所以在設計工具的選用上可是非常的重要。可以將其分為四個較為重要的因素來予以說明。

(1)可流通性(interchangeable)

設計工具的普遍性與可流通性，可以說是設計工具在選用上的第一考慮因素。由於電子科技上下游產業之分工已愈趨明顯，以往電子科技從設計包到製造

完成、包裝及測試等情形已不復見。所以必須考慮到所設計的產品，能否順利轉移到晶圓廠進行 ASIC 晶片實作。其中又需特別注意在設計階段所使用的設計工具，是否與其它公司所使用的設計工具相容。若設計工具不相容，對於所完成的結果，將與本先預期有極大的落差，甚至可能因此導致所設計出來的晶片功能錯誤而必須重來。

(2)功能性(functionality)

設計工具的功能性為選擇設計工具時的第二選擇因素。設計工具之功能性會影響到電路設計之性能與正確性，及產品之設計週期。功能性強的設計工具，可以使開發者在很短的時間內，設計出符合性能規格的電路，進而使設計產能增加。

(3)可擴充性(scalable)

微電子技術的快速進步及創新使得設計工具必須相對地予以擴充。例如積體電路的製程技術，從 $1.2\ \mu\text{m}$ 到 $0.8\ \mu\text{m}$ 、 $0.6\ \mu\text{m}$ 、 $0.5\ \mu\text{m}$ 、 $0.35\ \mu\text{m}$ 、 $0.25\ \mu\text{m}$ 、 $0.18\ \mu\text{m}$ 等，設計工具也必須同時來擴充以便整合技術得以最佳化。由此技術進步的技術看來，設計工具的可擴充性亦是不可或缺的項目。

(4)操作之方便與簡易性(convenience and easy to use)

操作簡單的設計工具可以減少開發者在熟悉工具的時間，更可以減低在設計時犯錯的機率。因為複雜的工具往往讓開發者無法在短時間內操控自如，而在開發時因工具不熟造成設計的錯誤。

(二)電路特性(characteristics)之考慮因素

電路特性的考慮因素有很多，在此針對需特別注意的要項加以說明如下：

(1)工作電壓(operation voltage)

以 CMOS 製程為製造技術的工作電壓一般有 5V、3.3V、2.5V 等，積體電路操作在不同的供應電壓的狀況下，會有功率消耗的不同與時脈延遲的差異。在一般情況下，電子電路工作電壓可容許的變動範圍，約為 5~10% 不等。另有一些製程技術，可以支援寬域(wide range)的操作電壓，例如 3V~5.5V 的寬域工作電壓。所以開發者必需先決定設計的電路，可以配合的工作電壓為何？再來選定可

支援的晶片。

(2)工作時脈(operation frequency)

工作時脈或操作頻率範圍是電路設計所要考慮的重要因素之一，數位電路的操作頻率範圍主要由所設計的電路其時脈延遲情況來決定。設計數位電路時，就所要求的操作頻率規格，反覆依據所設計的電路時脈延遲情況，加以模擬、驗證及測試。以保證能在規格之要求的操作頻率下能夠正常的工作。好的設計應有足夠的時序餘裕(timing margin)來保持電路操作的穩定性及可靠性。

(3)功率消耗(power consumption)

電路的功率消耗情形，在注重環保及省電的要求下，已是日漸重要。在不同的操作電壓會有不同的功率消耗情形，愈低的操作電壓會使得電路的時脈延遲加長，卻也能擁有較低的功率損耗之優點，如果電子電路是應用在可攜式的設備上，如行動電話、PDA 或筆記型電腦 等，省電的要求就更為重要。開發者應在進行電路設計時，就該對電路的功率消耗情形十分瞭解，並且詳細的分析。

(4)設計所佔資源與操作速度(die area and operation speed)

在晶片設計中，所佔資源指的是設計所耗用的面積，通常面積與速度是兩難的抉擇。因為要使操作速度加快，必須加一些邏輯元件，因此增加了電子電路設計的數量(gate count)，這也就增加晶圓的面積(die area)，例如 Carry Adder 與 Look Ahead Adder 即是明顯的例子。另外所佔面積增加了，除了增加成本外，亦會使得生產上的良率(yield rate)大打折扣。但如果操作速度不夠快，以致無法符合性能規格時，所設計的電路就無法使用。面積與速度的權重決定，可以在邏輯合成前加以評估。當然，最好的情況就是在符合性能規格的前提下，使用最小的面積以求得最佳的成本條件。

(5)臨界路徑(critical path)

在一個 VLSI 電路設計中，通常決定電路的工作性能，是由電路中的臨界路徑來決定。所謂臨界路徑通常是時間延遲最多的一條路徑，它決定了電路可操作的最高頻率。一個較好的設計，最好是將時間延遲因素平均分配於電路中的每一

個路徑。

(6)扇出與扇入(fan-in and fan-out)

每一個邏輯 Cell 都有一定的扇入標準與扇出能力。如果電路的設計超過了電路所能做到的扇出或扇入能力，將使電路工作情形變成不穩定，而造成無法正常工作的情形。

第三節 指紋前處理與編碼演算法

影像前處理的動作包括影像切割、雜訊消除、影像增強、二值化、細線化等[1][8]。本研究在指紋前處理的硬體化部份，選擇二值化與細線化之演算法，進行硬體化設計，因為這兩項工作在指紋辨識前處理上佔據了多數的時間；在指紋編碼演算法上，使用 3x3 遮罩進行找尋端點與分叉點的工作。演算法內容逐一說明如下：

(一)指紋前處理演算法

指紋影像經由前處理後可獲得品質較好的指紋影像，有助於往後指紋的特徵萃取或指紋編碼的工作。。

影像切割[8]通常為影像分析的第一步工作。切割就是將所要處理的目的物(object)從影像中分離出來，也就是將前景(目的物)與背景分離開來。當目的物已被完全分離出來時，切割的動作就該停止。圖 2 為一影像切割的例子，原始影像中包括有正方形()、圓形()及三角形()三個物體，若在原始影像中，要找尋的物體為正方形()，那就是所要處理的目的物，透過影像切割的動作，就可將 物體切割分離出來。

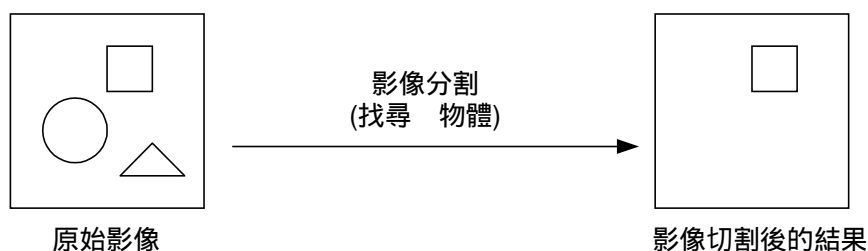


圖 2 影像分割圖例說明

指紋影像若要進行前處理的動作，通常應先分析所輸入的指紋影像的特性，例如：是否為灰階指紋影像、指紋影像背景是否為單一色調、指紋影像的大小、指紋影像完整度 等，針對不同的情況，選擇指紋前處理的方法，必需有效率的獲得良好指紋影像品質。綜合而之，指紋影像通常最少都需經由兩個較基本的前處理步驟處理後，才會再進行指紋特徵萃取或指紋編碼，而這兩個前處理的步驟分別是影像切割與二值化。

若以指紋影像而言，指紋本身就是屬於前景，我們必須從整個指紋影像中將前景與背景分離開來。進行影像切割可獲得指紋在整個指紋影像的位置，有助於加快往後處理的速度，因為背景影像是无需予以理會的，且影像切割同時具有濾除雜訊的功能。

進行影像切割時，通常會使用遮罩(mask)來檢測影像，圖 3 為一個典型的 3x3 遮罩，檢測的過程是把影像內被遮罩所包涵的每一點灰階值 GV(Grey Value)與遮罩中運算的係數 W(Weight)相乘，再一連串相加後的結果稱之為響應 R(Response)，公式(1)為遮罩在影像的響應計算方式。將所獲得不同的響應，進而透過特定的判斷式，如梯度運算子、Laplacian 運算子 等，便可對影像進行切割的動作。

| | | |
|-------|-------|-------|
| W_1 | W_2 | W_3 |
| W_4 | W_5 | W_6 |
| W_7 | W_8 | W_9 |

圖 3 典型的 3x3 遮罩， W_i 表示遮罩運算係數

$$R = W_1GV_1 + W_2GV_2 + \dots + W_9GV_9 = \sum_{i=1}^9 W_iGV_i \dots\dots\dots (1)$$

(1)二值化

由於二值化影像擁有較容易儲存、處理與辨認的特性，因此二值影像處理在影像辨識處理中佔著重要的地位。二值化，也就是將灰階影像轉換成 0 與 1 的表示方式，其基本原理亦即設定一個門檻值(threshold value)，凡是影像本身灰階值大於所設定的門檻值，便令該像素為 1，而當灰階值低於門檻值時便令其像素為

0，如此便可獲得二值化的影像。

假設有一灰階影像 f ，灰階影像中的水平座標及垂直座標分別以 x 與 y 表示，則 $f(x,y)$ 代表該像素的灰階值，設定門檻值為 m ，則二值化的方式如公式(2)所示。

$$\begin{cases} \text{if } f(x,y) \geq m & , \text{then } f(x,y) = 1 \\ \text{if } f(x,y) < m & , \text{then } f(x,y) = 0 \end{cases} \dots\dots\dots(2)$$

(2) 細線化

由於細線化的結果對於後續指紋特徵點的抽取效果影響很大，並且佔據掉大量的運算時間，所以本研究在比較不同細線化的方法與種類後，提出效果較好且適用於硬體化的細線化方法。

細線化在執行時有幾個需特別注意的條件有以下幾點[1][9][10]：

1) 端點不可誤認為雜訊而被刪除，如圖 4 所示(以 3×3 遮罩說明)。

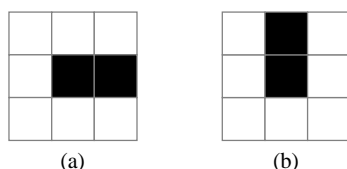


圖 4 端點不可刪除的情況範例

(a) 表示水平端點不可刪除

(b) 表示垂直端點不可刪除

2) 影像中原先就已連接的部份，不可因進行細線化後而被斷開，也就是連接點不可刪除，否則將造成影像不連續，如圖 5 所示(以 3×3 遮罩說明)。

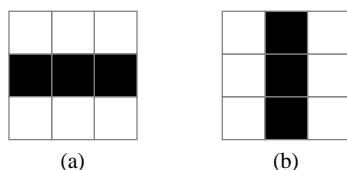


圖 5 連接點不可刪除的情況範例

(a) 表示水平連接點不可刪除

(b) 表示垂直連接點不可刪除

3) 細線化後的結果，必需可以表示原圖形。

在一般細線化演算法中，包括下列三步驟：

(a) 找出哪些像素需要被刪除

細線化便是要將圖像中多餘的資訊去除，因此第一步驟的目的便是決定影像

中的哪些資訊，需要被保留或刪除。決定哪些像素必須刪除的方法，也就是細線化演算法。

(b)進行刪除

處理步驟一的過程中，一旦決定像素需要被移除時，還不能直接將之刪除，必須等待掃描過全部影像後才能刪除。因為在細線化處理過程尚未結束前就刪除像素，相當於刪除了下一像素判斷的資訊，造成錯誤的細線化結果。所以一般在處理時，會先把需要刪除的像素先記錄起來，等到全部資訊掃描完一次後，再將需刪除的像素一次除去。

(c)重覆操作直至完成細線化

重覆步驟一及步驟二，直到沒有像素需要再被刪除為止即完成細線化的動作。因為一般影像沒有辦法經一次操作就完成細線化，所以必需再重覆執行第一步驟，判斷並記錄刪除點，於第二步驟進行刪除，同時觀察並判斷是否尚有像素需要刪除，如果沒有，就停止操作；反之，就得重覆操作。

細線化的方法可區分為下列四種基本型式

- 1) 3x3 型遮罩法(Zhang & Suen)[9][10][11][12]
- 2) Rectangular array[13]
- 3) Hexagonal array[14]
- 4) Loop Up Table

除了這四種基本型式外，另外還有利用歐幾里德距離(Euclidian distance)測量的方式達到骨架化(skeleton)的目的，但是使用此方式在於硬體化設計時，不但會造成複雜度的提高，並需花費更久的計算時間。

(二)指紋編碼演算法

指紋編碼泛指對原始指紋影像進行特定資訊編輯的動作。舉例而言，影像經過細線化所得的結果為一個像素寬(1-pixel width)影像，所以細線化處理即為編碼

方式的其中一種。此外，指紋影像的特徵點，如端點(ending point)或分叉點(bifurcation point)的萃取也都屬於指紋編碼的後一種資訊。以下針對二種編碼方式，即細線化、端點找尋及分叉點找尋，逐一說明：

(1)端點(Ending point)找尋

端點的物理意義表示一連續線段的開始點或結束點。端點為指紋細微特徵(minutiae)的其中一種，可利用遮罩找尋的方法，將端點從細線化後指紋影像中找尋出來。使用大小不同的遮罩(如 3x3、5x5、7x7...等)，將會得到不同找尋端點的結果。若使用的遮罩運算愈大，端點找尋的結果將愈準確，但所需耗費的時間將愈多；反之，若使用的遮罩運算愈小，端點找尋所花費的時間將愈少，但準確度相對會降低，所以必需視指紋影像品質的狀況選擇適合的遮罩進行找尋端點的動作。圖 6 為使用 3x3 遮罩找尋端點的例子，圖 7 則是使用 5x5 遮罩找尋端點的例子。

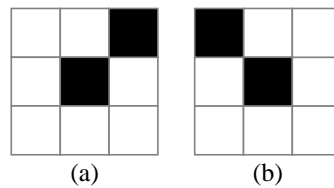


圖 6 兩個使用 3x3 遮罩找尋端點範例

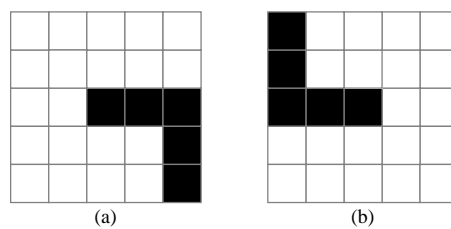


圖 7 兩個使用 5x5 遮罩找尋端點範例

(3)分叉點(Bifurcation point)找尋

分叉點的物理意義表示三條線段以上所交會的接點。分叉點也是為指紋細微特徵的其中一種，同樣可利用遮罩找尋的方法，將分叉點從細線化後指紋影像中找尋出來。使用大小不同的遮罩(如 3x3、5x5、7x7...等)，將會得到不同找尋端點的結果。若使用的遮罩運算愈大，分叉點找尋的結果將愈準確，但所需耗費的時間將愈多；反之，若使用的遮罩運算愈小，分叉點找尋所花費的時間將愈少，

但準確度相對會降低，所以必需視指紋影像品質的狀況選擇適合的遮罩進行找尋分叉點的動作。圖 8 為使用 3x3 遮罩找尋分叉點的例子，圖 9 則是使用 5x5 遮罩找尋分叉點的例子。

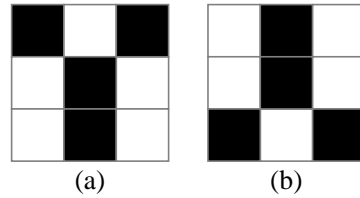


圖 8 兩個使用 3x3 遮罩找尋分叉點範例

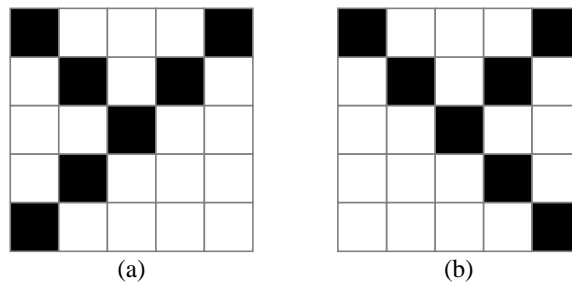


圖 9 兩個使用 5x5 遮罩找尋分叉點範例

第四節 可撓性與可擴充性硬體化的設計原則

可撓性與可擴充性的硬體化設計將使本架構擁有適應不同演算法與易於擴充影像遮罩尺寸的彈性空間，如此，將更符合系統單一晶片(System-On-Chip)的設計要求。為達成上述功能，將可撓性與可擴充性硬體化的設計原則歸納如下：

(一)可撓性硬體化設計原則

若要將單一演算法硬體化，可針對其演算法的特性及功能分析後，著手進行硬體化設計[15][16][17]；但假使要將不同的幾個演算法有效地設計到同一顆晶片內，要考慮的因素就必需更多。可撓性代表的就是針對不同的演算法，如影像切割、二值化、細線化 等，想出一可行的電路，同時涵蓋這幾種演算法的功能，經深入分析後，可歸納出可撓性硬體電路設計方法的設計原則如下：

(1)深入瞭解各演算法

對於不同演算法的特性及功能必需非常充份的瞭解，設計晶片之前必需對於各種演算法，透過撰寫軟體程式實際模擬驗證無誤。

(2)資源共用可行性分析

晶片資源分配，將左右該晶片的效能(performance)及成本(gate count)，在確實瞭解各演算法後，將相同的運算或可共用的資源列舉出來，做為可撓性電路設計原則的參考。

(3)可撓性硬體電路設計最佳化的考量

簡化演算法中計算的複雜度，如演算法內若使用到乘法，考慮是否可使用移位或加法 等方法來取代，亦可使得所設計的可撓性電路進一步得到最佳化(optimization)。

(二)可擴充性硬體化設計原則

在本章第二節曾經提及，若在同一種演算法內，選擇使用不同的尺寸大小的遮罩，對影像進行運算處理，將會獲得不同的結果。而一般影像處理硬體化的設計中[18][19][20]，往往只使用單一尺寸大小的遮罩進行運算，可擴充性將相對降低，本論文提出的可擴充性硬體電路設計架構，將可易於擴充影像中遮罩的尺寸。可擴充性的硬體電路設計原則如下：

(1)資料流控制(Data flow control)方式

影像輸出與輸入的控制是達成可擴充硬體電路設計的重要因素之一，將不同尺寸的遮罩影像資訊使用同一資料流控制的方式將是最佳的處理方式，不僅可節省硬體資源，並且可更為突顯可擴充性的功能的價值。

(2)處理單元(Processing Element, PE)的設計

處理單元是整個影像運算的重要核心，所以處理單元的設計必需同時兼具不同尺寸的遮罩運算能力。[21][22]