

第三章 聰明型攝影機系統

本章將詳細說明聰明型攝影機系統各部分。以 SOPC Nios Stratix 嵌入式系統為平台，我們發展聰明型攝影機系統，此系統可以擷取影像或視訊，也可藉由 VGA controller 透過螢幕顯示灰階影像。在我們的攝影機系統裡，我們加入硬體化影像處理演算法 Morphology。Morphology 演算法通常用來消除雜訊。若要作物件偵測時，Morphology 演算法用來做影像的前處理。由於我們發展硬體化 Morphology 電路，因此可以對擷取的影像作即時的處理。攝影機系統包括硬體化 Morphology 演算法，讓此攝影機更加聰明，同時也滿足物件偵測的需求。本論文中，聰明型攝影機系統包括 SOPC、CMOS 攝影機、VGA controller、及物件偵測器。物件偵測器即是硬體化 Morphology 電路。CMOS 攝影機及 VGA controller 如何與 SOPC Nios Stratix 發展電路板溝通，以及硬體化 Morphology 架構將在本章作詳細的說明。

3.1 聰明型攝影機系統概觀

聰明型攝影機系統包括 SOPC Nios Stratix 發展電路板、CMOS 攝影機、VGA controller、及硬體化 Morphology 電路。此系統架構圖如圖 3.1。聰明型攝影機資料流架構圖如圖 3.2。

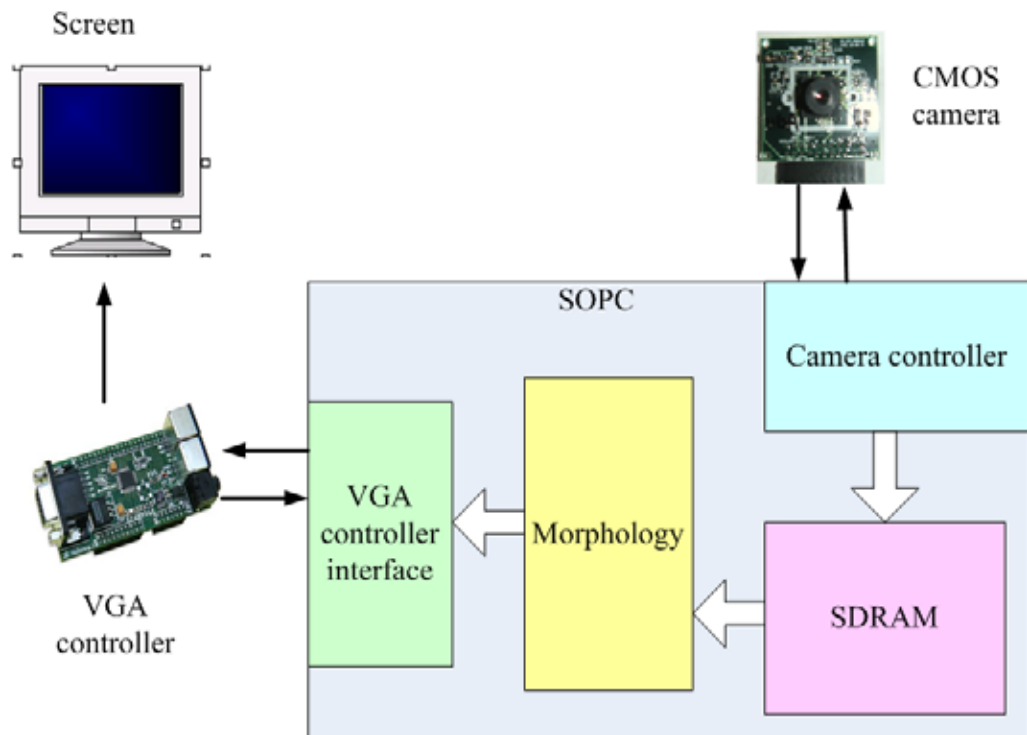


圖 3.1 聰明型攝影機系統架構圖

camera controller 是介於 SOPC Nios Stratix 發展電路板及 CMOS 攝影機溝通的橋樑。它控制 CMOS 攝影機，並傳送 CMOS 攝影機的輸出(RGB raw data)至 Avalon 匯流排。CMOS 攝影機擷取 640×480 解析度的影像，每秒可達 30 張畫面。此 30 張影像存在 16 Mbytes 的 SDRAM 裡。當系統要處理這些影像時，會從 SDRAM 傳送選擇的影像至 Morphology 電路。結果可藉由 VGA controller 顯示在螢幕上。VGA controller interface 是介於 SOPC Nios Stratix 發展電路板及 VGA controller 的溝通橋樑。在 VGA controller interface 上有調色盤(palette)，我們採用灰階調色盤。

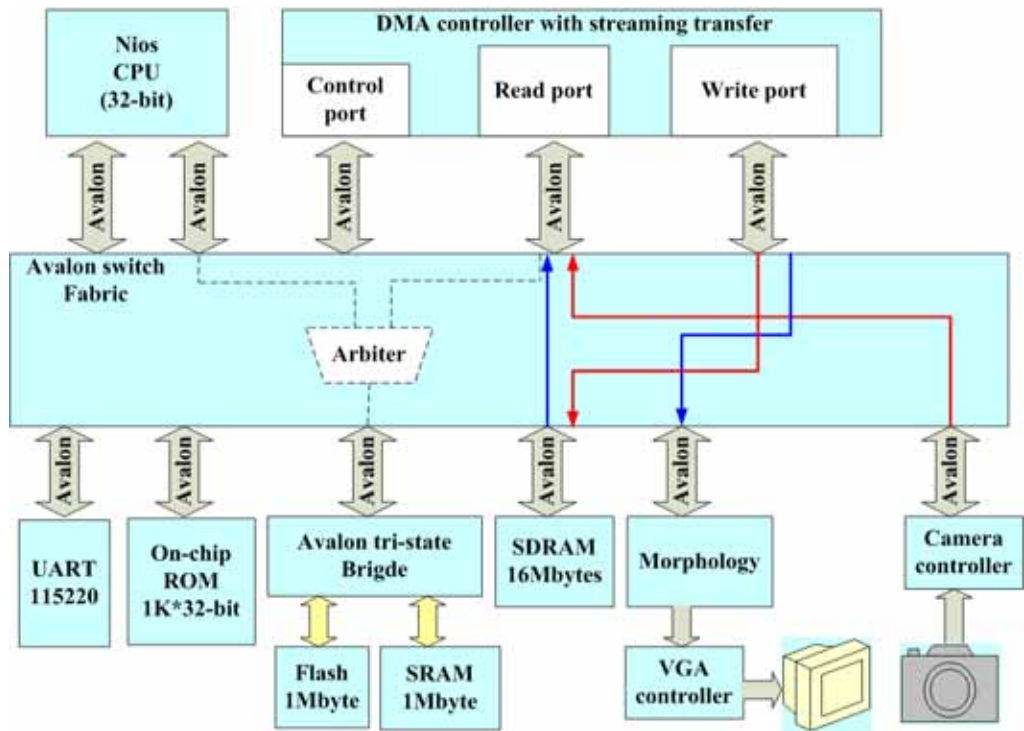


圖 3.2 聰明型攝影機資料流架構圖

SOPC Nios Stratix 嵌入式系統支援 DMA(Direct Memory Access)。藉由 DMA，SOPC 可以從週邊傳送資料至 SDRAM 或從 SDRAM 傳送至週邊而不需經由 Nios CPU。在 SOPC 系統裡，週邊和週邊的溝通必須經過 Avalon 匯流排。系統運作首先是 camera controller 從 CMOS 攝影機擷取影像，DMA 透過 Avalon 匯流排將 camera controller 擷取到的影像傳送至 SDRAM。同樣的，DMA 經由 Avalon 匯流排將 SDRAM 的影像傳送至 Morphology 電路。藉由 DMA，本論文之聰明型攝影機可更快速地處理影像。

3.2 Camera Controller

camera controller 是介於 SOPC Nios Stratix 發展電路板及 CMOS 攝影機的溝通橋樑。camera controller 介面如圖 3.3。

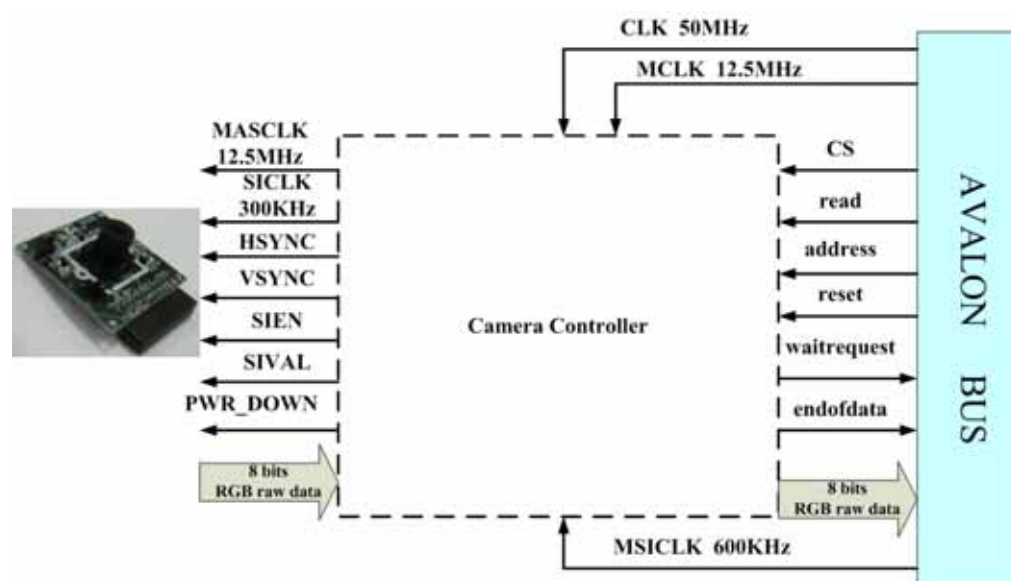


圖 3.3 camera controller 介面

由於 CMOS 攝影機使用 12.5 MHz 時脈，而 SOPC 系統為 50 MHz，CMOS 攝影機處理速度不如 SOPC 系統，因此 camera controller 必須具備溝通協調使其同步的能力。camera controller 之訊號主要分兩部分，一部分是：Dpixel、PWR_DOWN、MASCLK、SICLK、HSYNC、VSYNC、SIEN 及 SIVAL 用來控制 CMOS 攝影機。另一部分是：CLK、CS、read、address、reset、waitrequest、endofdata、及 Mpixel 用來和 Avalon 匯流排溝通協調。詳細的腳位說明見附錄 B。

camera controller 架構圖如圖 3.4。

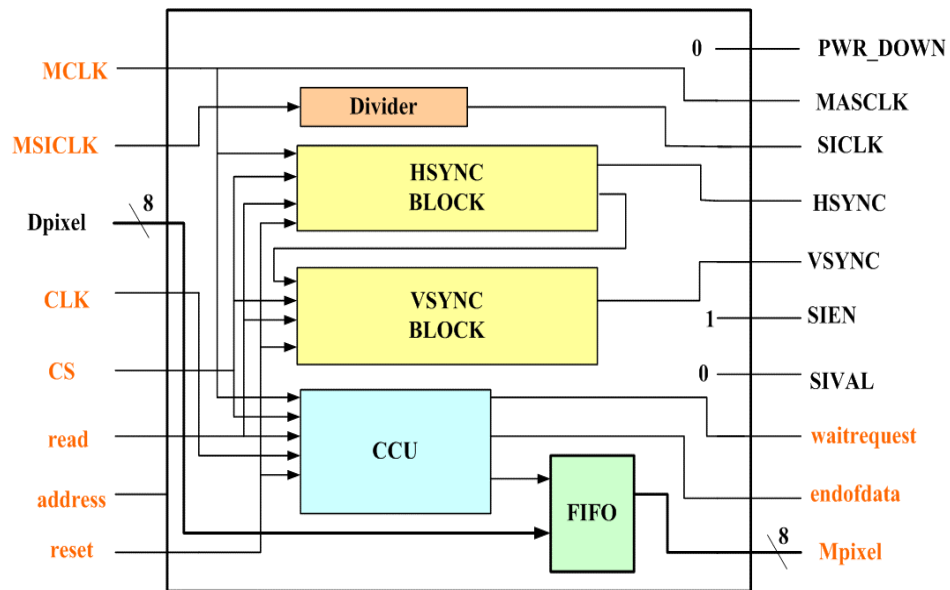


圖 3.4 camera controller 架構圖

圖 3.4 中，左邊是輸入訊號，右邊是輸出訊號。Camera controller 必須處理 CMOS 攝影機及 Avalon 匯流排的訊號。在 Nios 嵌入式系統裡，Avalon 匯流排主要用來作資料的傳輸。Avalon 匯流排的系統時脈為主時脈(master clock)。在 Nios 系統裡，主時脈為 50 MHz，而 CMOS 攝影機為 12.5 MHz。在資料讀取比資料輸出慢之情況下，我們在 camera controller 架構裡，設計 CCU 電路及 FIFO 使其同步。

3.2.1 camera controller 功能描述

(1) Divider:

Divider 為一個除頻器，在 CMOS 攝影機的規格需求上，若 CMOS 攝影機

每秒擷取 30 張畫面，則 SICLK 的時脈須為 300 KHz。然而 SOPC Nios Stratix 發展電路板可提供之最低時脈為 600 KHz。因此我們需要除頻器將 600 KHz 降至 300 KHz。

(2) HSYNC BLOCK 與 VSYNC BLOCK:

HSYNC BLOCK 是由除頻器及計數器所組成，用來控制 HSYNC 訊號。HSYNC 為水平同步訊號，用來表示換新列。VSYNC BLOCK 也是由除頻器及計數器所組成，主要用來控制 VSYNC 訊號。VSYNC 是用來表示換新的影像畫面的訊號。VSYNC BLOCK 的時脈來自 HSYNC BLOCK。MASCLK 之時脈可由設計者自訂，然而最高可支援時脈為 24 MHz。MASCLK 和 VSYNC 及 HSYNC 有以下的關係：

$$(\text{MASCLK rate}) = (\text{frame rate}) \times (N) \times (M)。$$

N 代表列的數目，由 VSYNC 控制。M 代表每個列裡有多少個像素，由 HSYNC 所控制。只要 MASCLK 之時脈在 24 MHz 之內，N 及 M 可以由設計者彈性決定。在 CMOS 攝影機的規格中，N 必須大於 501，M 必須介於 758~1023 之間。我們要擷取 640 (行) × 480 (列) 的影像，因此我們設計當 CMOS 攝影機產生 801 個像素(即 M=801)時，HSYNC BLOCK 會將 HSYNC 設成低電位，並告知 VSYNC BLOCK。當 CMOS 攝影機產生 520 個列(即 N=520)時，

VSYNC BLOCK 會將 VSYNC 設成低電位。圖 3.5 為有效像素關係圖。

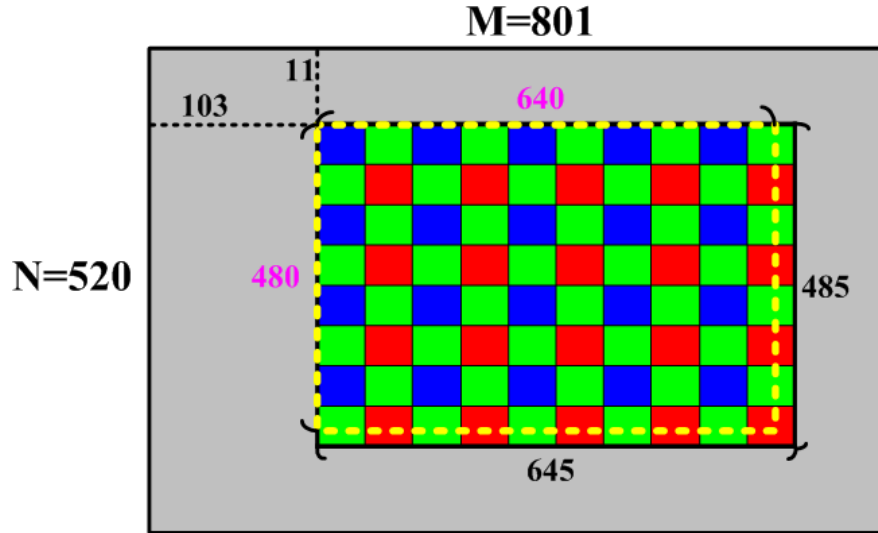


圖 3.5 有效像素關係圖

每列自第 103 個像素(從第 0 像素開始)開始為有效像素。每列 801 個像素中，有 645 有效像素，156 (801-645)個 optical black 像素(無效像素)。此 645 有效像素可以以我們的需求擷取，在此我們取前 640 個像素輸出。每行自第 11 個像素(從第 0 像素開始)開始為有效像素。在 520 個列中，有 485 個有效列，35 (520-485)個無效列。我們取前 480 個有效列輸出。圖 3.6 為 MASCLK 和 HSYNC 關係範例。在每列 801 個像素中，第 103 個像素為第一個有效像素。圖中 CMOS 攝影機已擷取至第二列的第四個有效像素。第二列的第四個有效像素即為 801 像素中，第 106 像素。

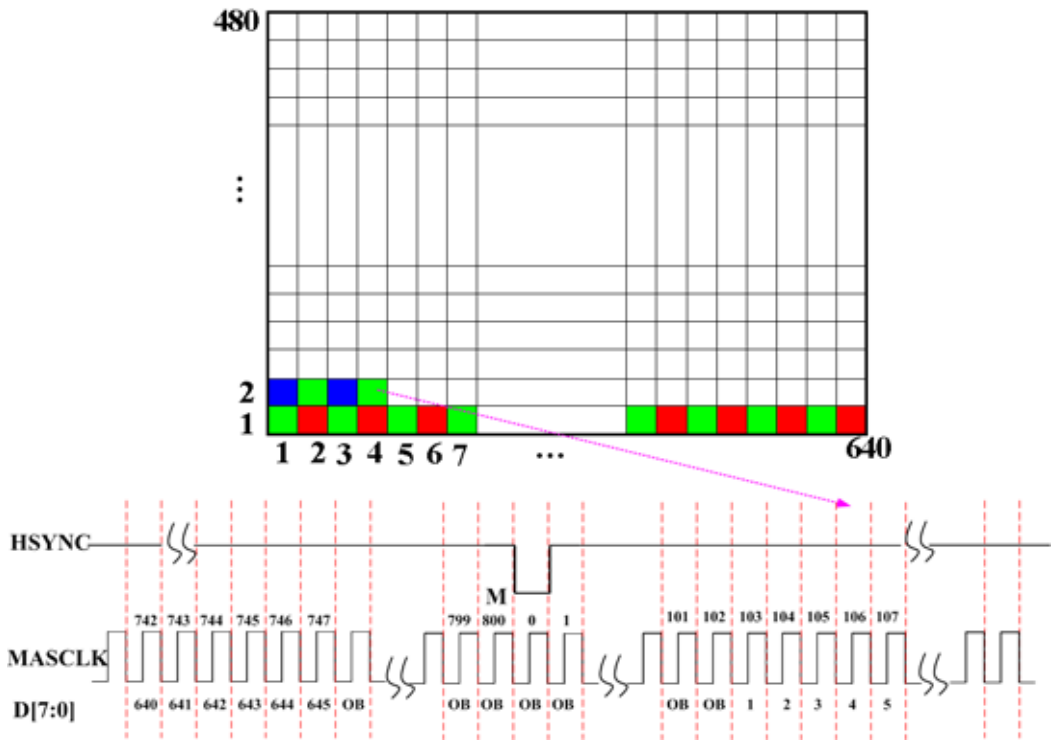


圖 3.6 MASCLK 與 HSYNC 關係範例

圖 3.6 為 VSYNC 和 HSYNC 關係範例。

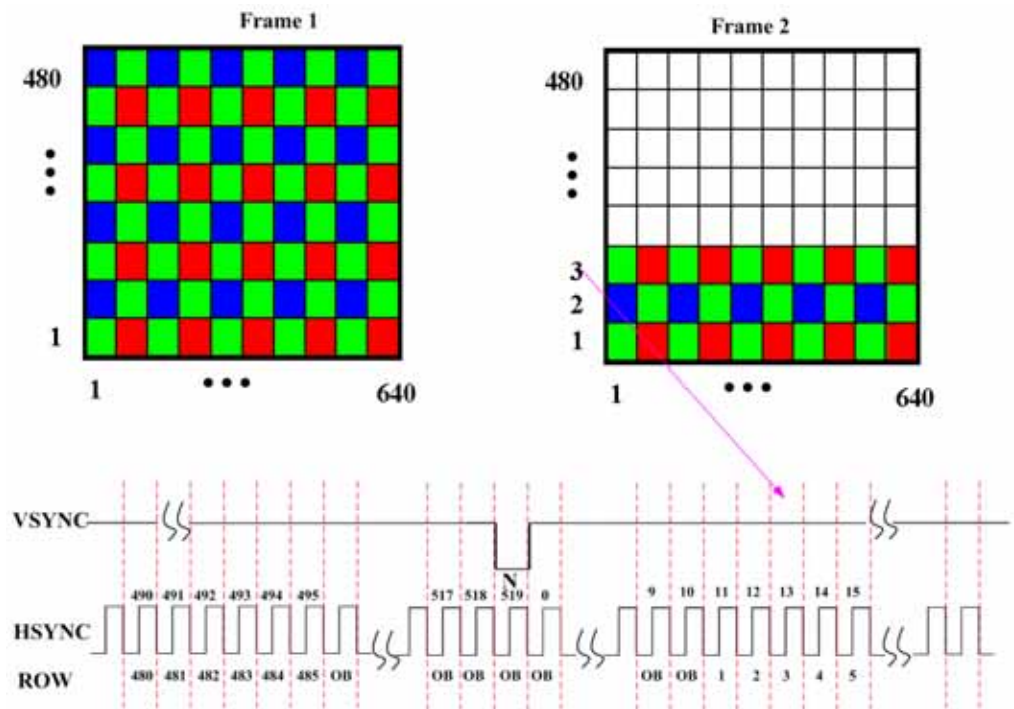


圖 3.7 VSYNC 和 HSYNC 關係範例

圖 3.7 中，CMOS 攝影機已擷取至第二張影像之第三列。此第三列為 520 個列中第 13 個列。由於 MASCLK 和 VSYNC 及 HSYNC 有以下的關係：

$$(\text{MASCLK rate}) = (\text{frame rate}) \times (N) \times (M)$$

的關係。

因此在 N=520，M=801，frame rate 為 30 的條件下，MASCLK 為 12495600 Hz。我們取 MASCLK 為 12.5 MHz 當 CMOS 攝影機的主時脈。SOPC Nios Stratix 發展電路板中，可由 SDRAM PLL 模組產生 600 KHz 至 200 MHz 的頻率。因此我們用它來產生 CMOS 攝影機所需的時脈。

(3) CCU:

CCU 是”Camera Control Unit”的縮寫。CCU 電路主要用來和 Avalon 匯流排溝通，同時決定 HSYNC 及 VSYNC 所產生的有效像素中，有哪些像素是需要的，也就是決定影像之行和列的大小。由於 CMOS 攝影機所擷取的影像像素不斷地進來，CCU 電路利用 waitrequest 訊號通知 SOPC Nios 發展電路板作像素的擷取。當 waitrequest 為低電位時，SOPC Nios 系統擷取像素，當 waitrequest 為高電位時，SOPC Nios 系統暫停擷取像素。當 CCU 和 FIFO 電路用來協調 SOPC Nios 發展電路板及 CMOS 攝影機，使其同步。當 FIFO 從 Dpixel 訊號接收新像素時，CCU 通知 Nios 系統開始從 FIFO 傳送資料至 Avalon 匯流排。

(4) FIFO:

FIFO 是"First In First Out"的縮寫。FIFO 是個資料緩衝器(buffer)，負責暫存從 CMOS 攝影機擷取的像素。由於 CMOS 攝影機比 SOPC Nios 發展電路板慢，因此 FIFO 的寫入速度不如 FIFO 的讀取速度，所以 FIFO 只需 1 byte 的容量即可。當 waitrequest 訊號為低電位時，FIFO 無法輸出資料至 Avalon 匯流排。在 FIFO 未完成從 Dpixel 接收新的像素前，Nios 系統必須等待。

3.2.2 CMOS 攝影機與 Nios 系統之 DMA 傳輸

無論 CMOS 攝影機擷取影像或視訊，其資料皆非常大。因此我們使用 DMA 作資料的串流傳輸控制來縮短傳輸的時間。在聰明型攝影機系統中，利用 DMA 從 CMOS 攝影機傳送資料至 SDRAM。DMA 有 master port，如：read 及 write。master port 可以在 Avalon 匯流排上初始資料的傳輸。本論文之 camera controller 為擁有 master port 之主週邊(master peripheral)。圖 3.8 為 CMOS 攝影機與 Nios 系統之 DMA 傳輸。

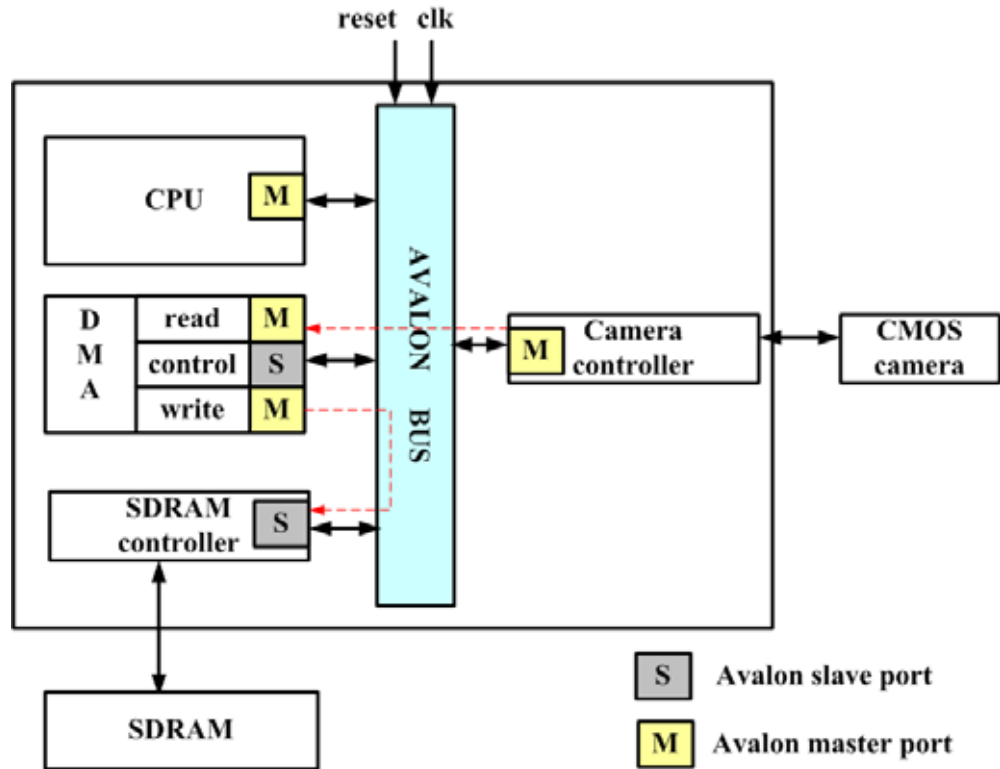


圖 3.8 CMOS 攝影機與 Nios 系統之 DMA 傳輸

當 DMA 想要從 camera controller 讀取資料時，Nios CPU 對 address 訊號作解碼。address 及 read 訊號有意義時，DMA 直接進行資料的傳輸。若 camera controller 將 waitrequest 設為高電位，則 DMA 暫停 Avalon 匯流排資料的傳送。當 DMA 完成 30 張影像(30x640x480)的資料傳送，endofdata 會被設為高電位，結束 DMA 的傳送。

圖 3.9 為 camera controller 資料串流時序描述。

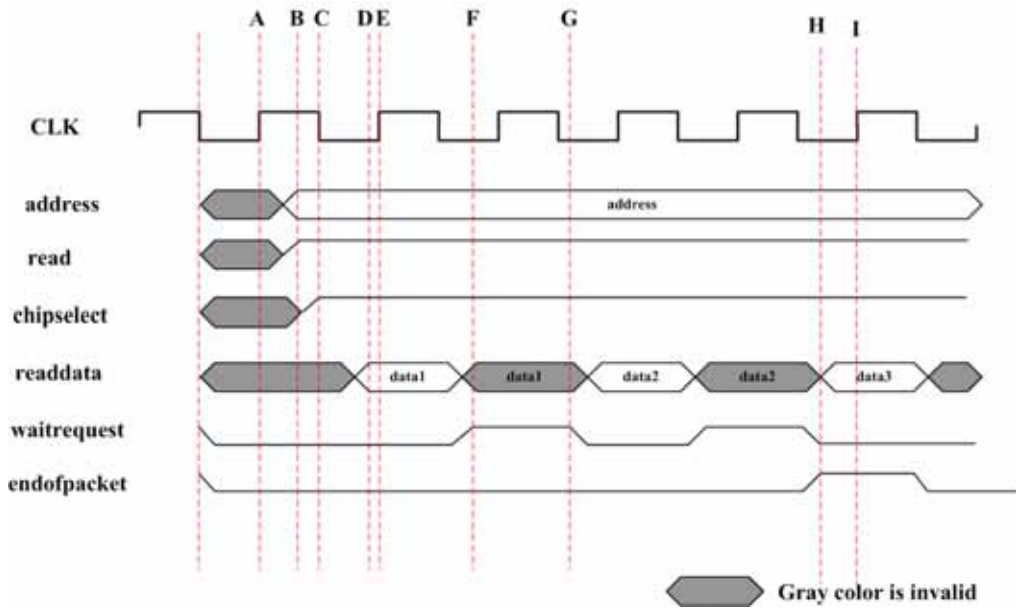


圖 3.9 camera controller 資料串流時序描述

- (A) 第一個匯流排 cycle 在 clk 的上緣觸發。
- (B) address 及 read 訊號有意義。
- (C) Avalon 匯流排模組對 address 進行解碼，然後將 chipselect 設為高電位。
- (D) 在下個 clk 上緣觸發前，readdata 訊號值變得有意義。Avalon 匯流排在
下個 clk 上緣觸發時擷取資料。
- (E) chipselect 及 read 在每個匯流排 cycle 持續高電位，camera controller 輸
出有意義的 readdata。
- (F) waitrequest 被設為高電位，代表在下個 clk 上緣觸發時，camera controller
無法輸出有意義的 readdata。

(G) waitrequest 被設為低電位，代表在下個 clk 上緣觸發時，camera controller

輸出有意義的 readdata。

(H) 當完成 30 張影像(30×640×480)的資料傳送，endofdata 會被設為高電位。

(I) camera controller 輸出最後之有意義的 readdata。

3.2.3 camera controller 之合成結果

本論文之 camera controller 為由 VHDL 發展而成之電路，它定義了與 CMOS 攝影機及 Avalon 匯流排之訊號連接。我們使用 Altera QuartusII 3.0 去編譯及合成此 camera controller 並得到表 3.1 的結果。

Family	LEs	Pins	DSP block	Total memory bits
Stratix	361(3%)	34 (7%)	0(0%)	0 (0%)

表 3.1 camera controller 之合成結果

3.3 物件偵測器

本論文中，我們發展聰明型攝影機，使攝影機可以自己分析及偵測物件。

物件偵測過程如圖 3.10。

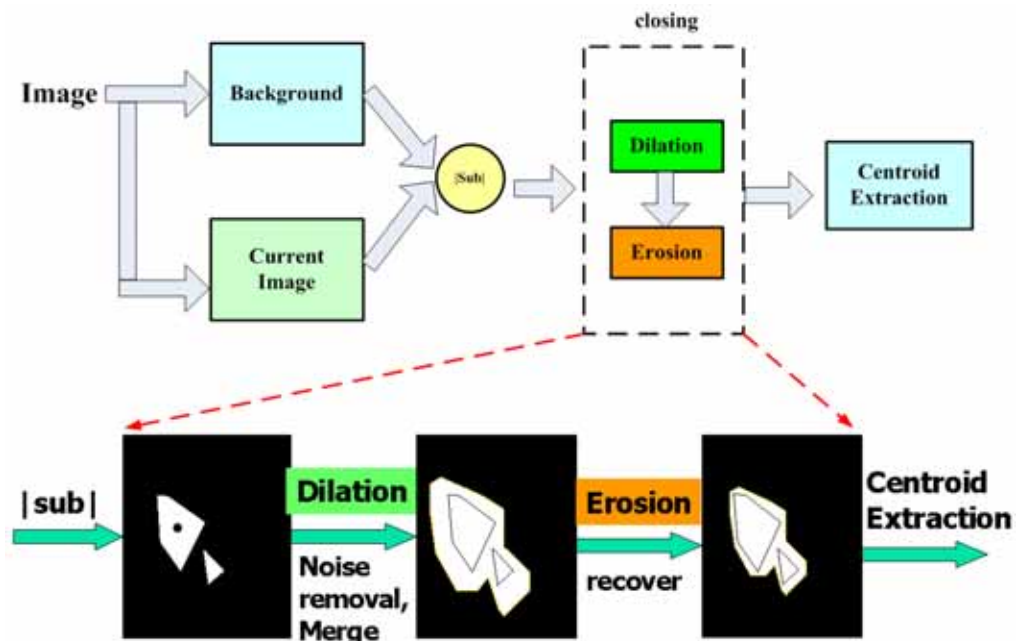


圖 3.10 物件偵測過程

CMOS 攝影機所擷取的影像有背景圖(background)及目前圖(current image)。目前圖有我們想要偵測的物件。當聰明型攝影機系統要偵測物件時，會將目前圖減背景圖然後取絕對值。得到的結果其目標物可能會破碎同時有一些黑點的雜訊。因此使用 Morphology 之 dilation 將物件合併同時去除雜訊。為了增加物件之重心擷取的準確性，使用 Morphology 之 erosion 將 dilation 後膨脹的體積縮小和原圖差不多。最後再擷取重心。本論文只實作至 Morphology 演算法。

3.3.1 Morphology 演算法

morphology 字義代表形狀及構造。在電腦視覺裡，它可表示區域的形狀。

數學形態學 (Mathematical Morphology) 在分析二元影像(binary image)及灰

階圖時非常有用。它在影像處理上有下列的應用：

- (1) 邊緣偵測(edge detection)。
- (2) 去除雜訊(Noise removal)。
- (3) 影像增益(Image enhancement)。
- (4) 影像分割(Image segmentation)。

Morphology 演算法有兩個基本的運算：擴張(Dilation)及侵蝕(erosion)。這

兩個運算需要兩部分：

- (1) Active image：進行 Dilation 或 erosion 的影像。
- (2) Kernel image: 同時也被稱為 structuring element，代表著形狀。

本論文的系統中，我們發展硬體化 Morphology 電路來處理影像。此硬體化 Morphology 之 structuring element 為可調的。雖然 DMA 透過 Avalon 匯流排從 SDRAM 傳送像素至 VGA controller interface 必須經過硬體化 Morphology，然而我們可以調整 structuring element 的大小及形狀以顯示原圖。本論文之 Morphology 電路具備 dilation 及 erosion 兩個基本運算。

3.3.2 二元形態運算--dilation

二元擴張(binary dilation)運算可用下列數學式表示：

A (代表前景像素), S (代表 structuring element) 為 Z^2 (two dimensional Euclidean space) 之子集合。

$A \oplus S = \{c \in Z^2 \mid c = a + s, a \in A \text{ and } s \in S\}$
 + 為向量加法(vector addition)。圖 3.11 為 dilation 運算範例

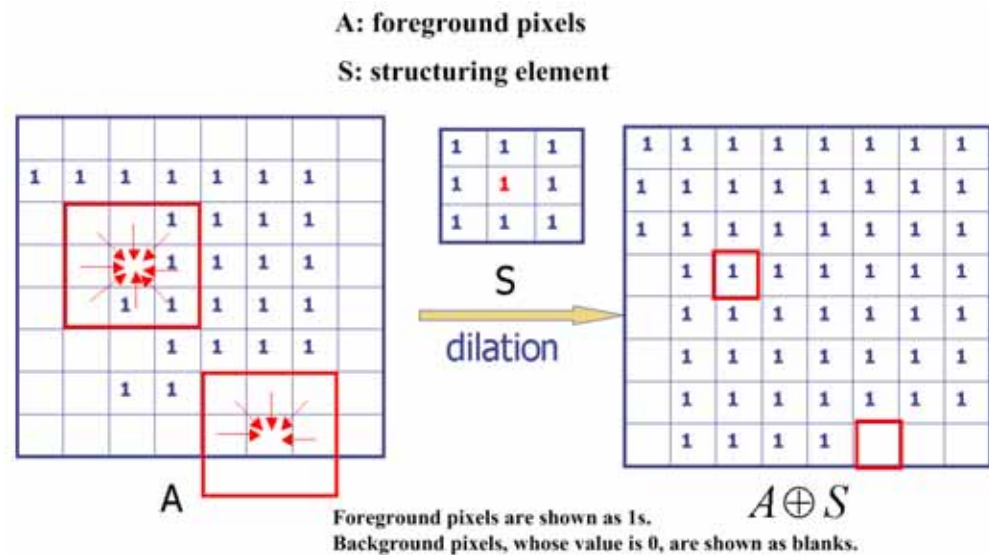


圖 3.11 dilation 運算範例

圖 3.11 之 structuring element 為 3x3 矩形。structuring element 可為任意大小及形狀，例如：環狀及十字形。若 structuring element 為 1x1，處理的結果

為原圖。本論文處理之灰階圖可以下列數學式表示：

$$(f \oplus S)(i, j) = \max\{f(i-x, j-y) + s(x, y) \mid (i-x, j-y) \in D_f \text{ and } (x, y) \in D_s\}$$

D_f = domain of f (picture).

D_s = domain of S (structuring element).

3.3.3 二元形態運算--dilation

侵蝕(erosion)為 dilation 相反運算。可用下列的數學式表示：

$$A \ominus S = \{c \in Z^2 \mid c + s \in A \text{ for every } s \in S\}$$

圖 3.12 為 erosion 運算範例。

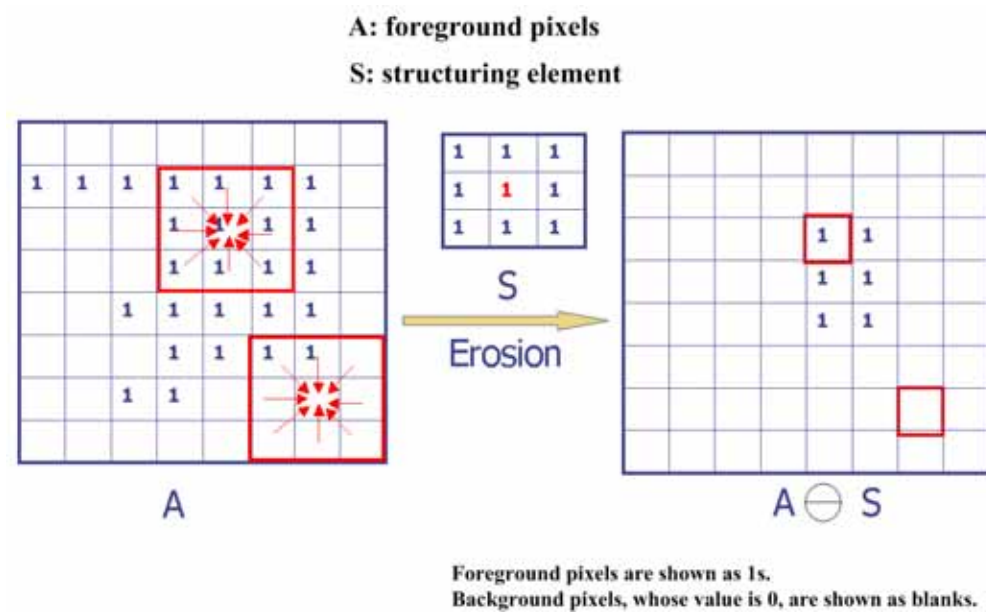


圖 3.12 erosion 運算範例

本論文處理之灰階圖可以下列數學式表示：

$$(f \ominus S)(i, j) = \min \{ f(i + x, j + y) - s(x, y) \mid (i + x, j + y) \in D_f \text{ and } (x, y) \in D_s \}$$

D_f = domain of f (picture).

D_s = domain of S (structuring element).

在本論文之硬體化 Morphology 架構裡， $s(x, y)$ 皆為 0。因此以 dilaton 處理灰階圖時，取周圍之最大值。以 erosion 處理灰階圖時，取周圍之最小值。

3.3.4 硬體化 Morphology 電路

硬體化 Morphology 電路架構如圖 3.13。

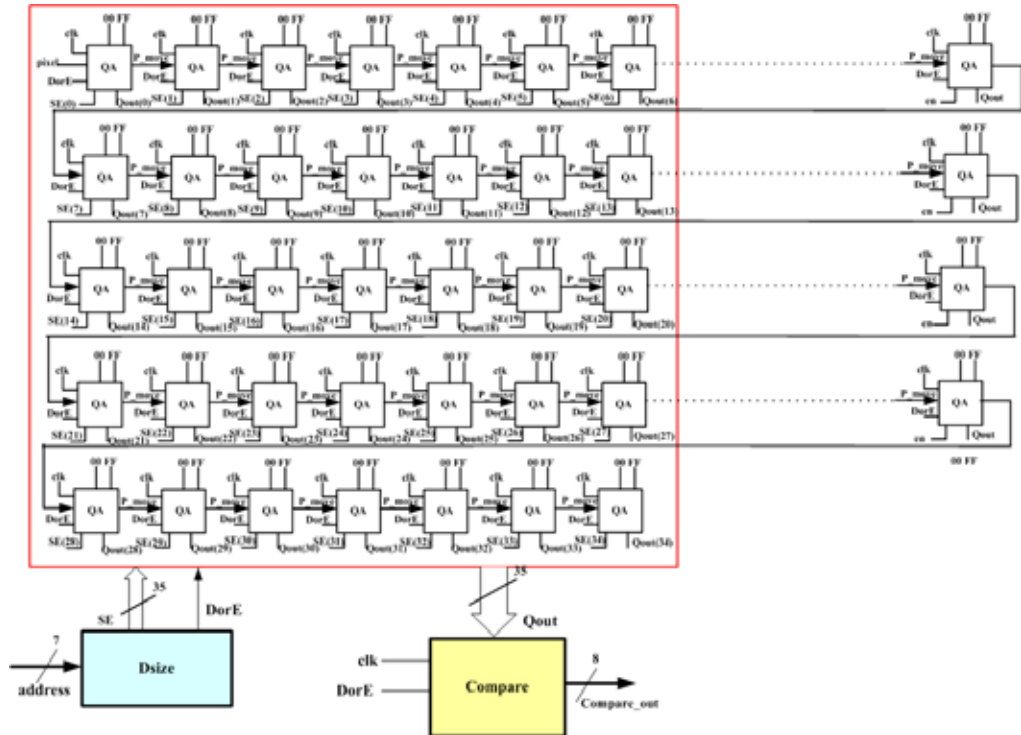


圖 3.13 硬體化 Morphology 電路架構

硬體化 Morphology 電路主要由蛇狀(snake-like) QA、Dsize、Compare 電路所組成。它們的說明如下：

(1) 蛇狀(snake-like) QA:

QA 是移位暫存器，當 clk 上緣觸發時會將資料傳給下個 QA。我們將 2567 個 QA 串起來，使其第一列、第二列、第三列、第四列皆有 640 個 QA，由於像蛇的形狀，因此稱蛇狀(snake-like) QA。本論文設計可調之 structuring element，在 5x7 之矩形範圍內，例如：1x1、3x3、5x5、5x6、5x7 等等，可

以任意改變。因此如圖 3.13，我們選擇每列之前 7 個 QA，共 35 個 QA。比較此 35 個 QA 之 Qout 值，取其最大值或最小值。圖中，Qout(0)為第一個 QA 之輸出訊號，Qout(7)為第 641 個 QA 之輸出訊號。

表 3.2 為 QA 所有訊號之關係。

SE & DorE	Qout
0 0	00
0 1	FF
1 0	P_move
1 1	P_move

表 3.2 QA 訊號之關係

QA 暫存的像素值利用 P_move 訊號來傳送至下個 QA。當 DorE 為低電位時，Morphology 電路作 dilation 的運算，同時我們取 35 個 Qout (Qout(0)~Qout(34))值中之最大值當輸出。當 DorE 為高電位時，Morphology 電路作 erosion 的運算，我們取 35 個 Qout (Qout(0)~Qout(34))值中之最小值當輸出。SE 訊號用來決定 structuring element 之大小及形狀。在此舉兩個例子來說明：

(I) Dilation 運算且 structuring element 為 3×3 矩形:

(a) 所有 QA 之 DorE 皆為低電位。

(b) SE(0), SE(1), SE(2), SE(7), SE(8), SE(9), SE(14), SE(15), SE(16) 為高電

位，其它為低電位。

(II) Erosion 運算且 structuring element 為 4x5 矩形:

(a) 所有 QA 之 DorE 皆為高電位。

(b) The SE(0), SE(1), SE(2), SE(3), SE(4), SE(7), SE(8), SE(9), SE(10),

SE(11), SE(14), SE(15), SE(16), SE(17), SE(18), SE(21), SE(22),

SE(23), SE(24), SE(25) 為高電位，其它為低電位。

(2) Dsize :

Dsize 為一解碼器。Dsize 是由 look-up table 所發展而成的電路。我們用 Dsize 來決定 structuring element 之大小及形狀，同時它也決定 Morphology 電路進行 dilation 或 erosion 運算。Address 是 Avalon 匯流排 7-bit 的訊號，address(0)至 address(5)用來決定 structuring element 之大小及形狀，address(6)用來決定 Morphology 電路進行 dilation 或 erosion 運算。SE 是 Dsize 電路之輸出，用來決定 structuring element 之大小及形狀。

(3) Compare :

Compare 電路比較 35 個 8-bit 訊號。當 DorE 為低電位時，Morphology 電路進行 dilation 的運算，同時我們取 35 個 Qout 值中之最大值當輸出。當 DorE 為高電位時，Morphology 電路作 erosion 的運算，我們取 35 個 Qout(0)至 Qout(34)值中之最小值當輸出。最後，當得到最大或最小值時，將此結果傳送到 VGA controller interface 電路。Compare 電路架構如圖 3.14。

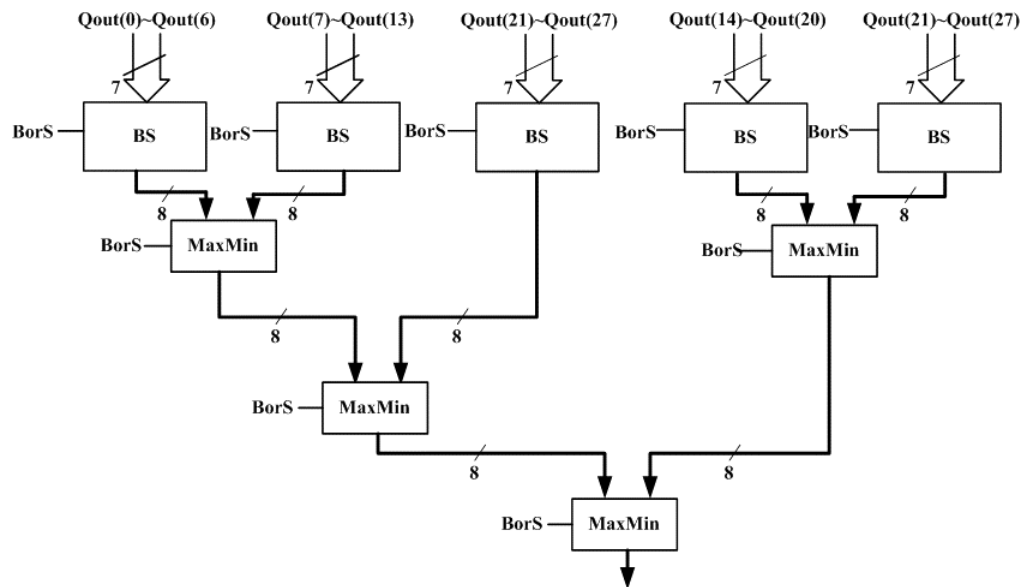


圖 3.14 Compare 電路架構

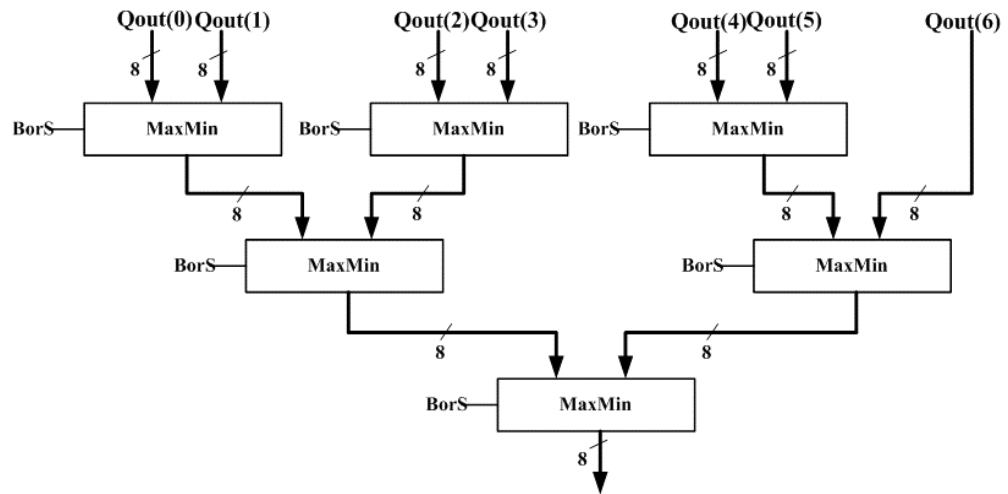


圖 3.15 BS 電路架構

BorS 訊號和 DorE 訊號相同，BorS 的訊號來源為 DorE。當 BorS 為低電位時，Morphology 電路進行 dilation 的運算，因此取 35 個 Qout 值中之最大值當輸出。當 BorS 為高電位時，取 35 個 Qout 值中之最小值當輸出。圖 3.14 及圖 3.15 中，MaxMin 電路為比較器，當 BorS 為低電位時，取最大值。BorS 為高電位時，取最小值。

3.3.5 硬體化 Morphology 電路之合成結果

本論文之 Morphology 電路由 VHDL 發展而成。我們使用 Altera QuartusII 3.0 去編譯及合成此 Morphology 電路並得到表 3.3 的結果。

Family	LEs	Pins	DSP block	Total memory bits
Stratix	1211(11%)	27 (6%)	0(0%)	20224 (2%)

表 3.3 Morphology 電路之合成結果

3.4 VGA Controller Interface

VGA Controller Interface 是介於 SOPC Nios Stratix 發展電路板和 Lancelot VGA controller 的溝通橋樑。圖 3.16 為 VGA Controller Interface 之架構圖。

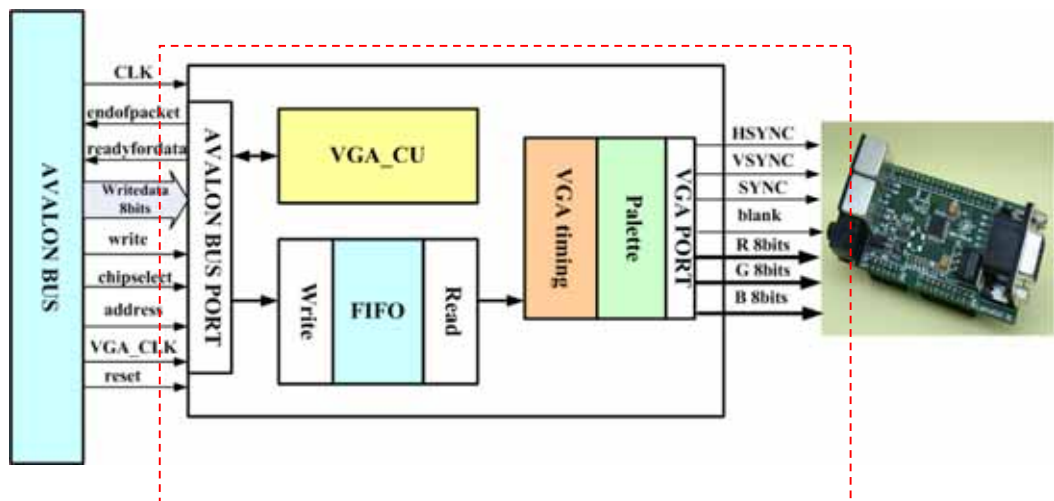


圖 3.16 VGA Controller Interface 之架構

VGA controller interface 的訊號主要分兩部分，一部分用來控制 Lancelot VGA controller，例如：HSYNC、VSYNC、SYNC、blank、R (Red)、G (Green)、B (Blue)。另一部分用來和 Avalon 匯流排溝通，例如：CLK、endofpacket、readyfordata、writedata、write、chipselect、address。VGA controller interface 的功能類似 camera controller。它們皆要和 Avalon 匯流排溝通，同時傳送 HSYNC 及 VSYNC 訊號至週邊。然而，VGA controller interface 傳送影像至週邊。camera controller 從週邊接收影像。

Nios 嵌入式系統使用的主時脈為 50 MHz，VGA controller 之時脈為 25

MHz，在 FIFO 的寫入比 FIFO 的讀取快情況下，VGA controller interface 必須有溝通協調使其同步的能力。詳細的 VGA controller interface 腳位說明見附錄 C。

3.4.1 VGA controller interface 功能描述

(1) VGA_CU :

VGA_CU 為”VGA Control Unit”之縮寫。VGA_CU 主要工作在於和 Avalon 匯流排溝通協調。當 FIFO 滿時，VGA_CU 利用 readyfordata 告知 Nios 系統 Avalon 匯流排目前的情況，並且 DMA 停止傳送資料。當 VGA controller interface 完成傳送一張影像(640×480)，VGA_CU 利用 endofpacket 訊號通知 Nios 系統結束 Avalon 匯流排資料傳輸。

(2) FIFO :

FIFO 是資料緩衝器。由於資料輸入和資料輸出速度不同，因此需要 FIFO 使 SOPC Nios Stratix 系統和 VGA controller 同步。此 FIFO 為 Altera 的 dual-clock FIFO megafunction，利用 Altera FPGA 之 dual-port 記憶體架構提供各自獨立的資料讀取與寫入之時脈。此 FIFO 之容量大小為 1024-byte。當 FIFO 從 SDRAM 接收 1000 個像素(1000 bytes)時，我們認為 FIFO 已滿，並且設定 readyfordata 為低電位，停止 Avalon 匯流排資料傳送。

(3) VGA timing :

VGA timing 電路控制 HSYNC 及 VSYNC 訊號。此電路也使用 look-up table 將 8-bit 像素值轉換成 8-bit R、G 及 B 值。在 VGA controller interface 中，此 look-up table 為調色盤(palette)。

(4) Palette :

此調色盤的顏色為灰階色。VGA timing 電路使用此調色盤將 8-bit 像素值轉換成 8-bit R、G 及 B 值。此 8-bit R、G 及 B 值皆相同。

3.4.2 VGA controller 與 Nios 系統之 DMA 傳輸

儲存在 SDRAM 中的影像資料龐大，因此我們利用 DMA 將影像從 SDRAM 傳送到 Lancelot VGA controller。圖 3.17 為 VGA controller 與 Nios 系統之 DMA 傳輸。

DMA 為一主週邊(master peripheral)。DMA 首先從 SDRAM 讀取資料，然後寫入資料至 VGA controller interface。在 DMA 架構中有 master port，例如：read 及 write。master port 可以主動初始 Avalon 匯流排的傳輸。VGA controller 為只擁有 slave port 的從週邊(slave peripheral)，slave port 為週邊上數個 port 之集合，只能被動接受 Avalon 匯流排之訊號。

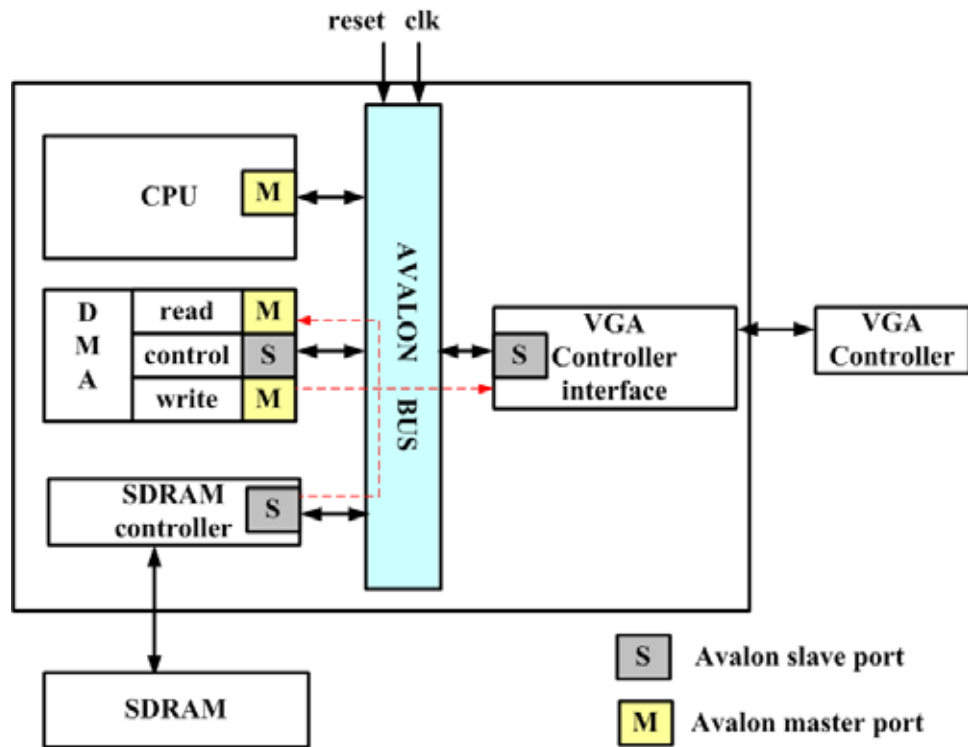


圖 3.17 VGA controller 與 Nios 系統之 DMA 傳輸

當 DMA 想要寫入資料至 VGA controller interface 時，Nios CPU 對 address 作解碼。address 及 write 變有意義時，DMA 直接從 SDRAM 傳送資料至 VGA controller interface。當 FIFO 滿了，readyfordata 訊號會設為低電位，停止資料的寫入，當 FIFO 清空後，readyfordata 訊號會設為高電位，繼續作資料的寫入。當 DMA 透過 Avalon 匯流排完成一張影像的傳輸後，endofpacket 會被設為高電位，結束資料傳輸。圖 3.18 為 VGA controller interface 資料串流時脈描述。

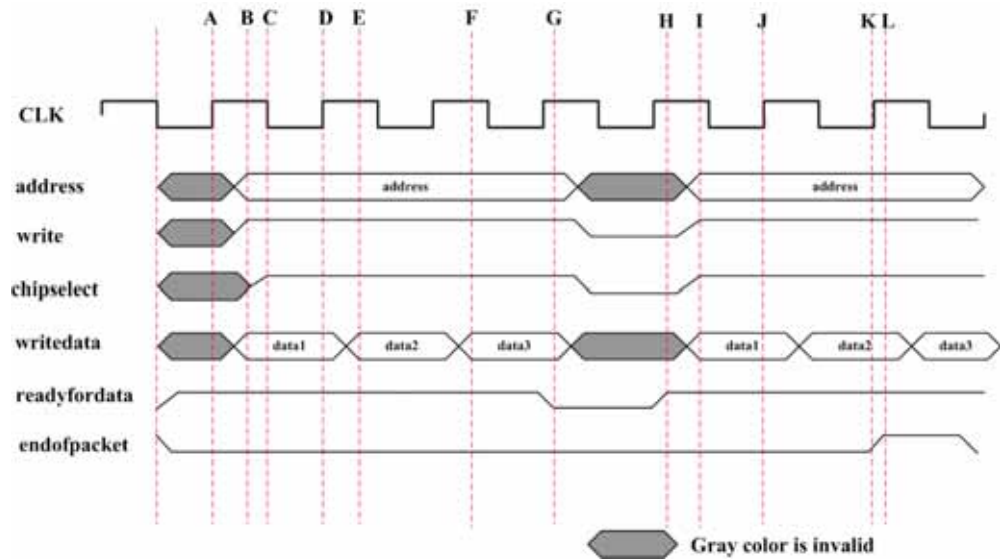


圖 3.18 VGA controller interface 資料串流時脈描述

- (A) 第一個匯流排 cycle 在 clk 的上緣觸發。
- (B) address、write、writedata 訊號有意義。
- (C) Avalon 匯流排模組對 address 進行解碼，然後將 chipselect 設為高電位。
- (D) VGA controller interface 在 clk 上緣觸發時從 Avalon 匯流排擷取資料。
- (E-F) chipselect 及 write 在每個匯流排 cycle 持續高電位，Avalon 匯流排輸出有意義的 writedata。VGA controller interface 必須在下個 clk 上緣觸發時擷取資料。
- (G) 當 FIFO 滿時，VGA controller interface 將 readyfordata 設為低電位，強迫 Avalon 匯流排延緩資料串流的寫入。需注意的是 write 及 chipselect 仍為高電位，writedata 仍然有意義，代表資料串流之 master post 仍在等

待傳送的結束。接著 write 及 chipselect 被設為低電位，address 及 writedata 為無意義之值。

(H) FIFO 清空後，VGA controller interface 再度將 readyfordata 設為高電位。

(I) 由於 readyfordata 為高電位，Avalon 匯流排將 write 及 chipselect 設為高電位，address 及 writedata 變成有意義。

(J-K) VGA controller interface 在 clk 上緣觸發時擷取 writedata。chipselect 及 write 在每個匯流排 cycle 持續高電位，Avalon 匯流排輸出有意義的 writedata。

(L) 當要結束一張影像的傳送時，VGA controller interface 在最後一個 clk 上緣觸發前將 endofpacket 設為高電位。

3.4.3 VGA controller interface 之合成結果

本論文之 VGA controller interface 為由 verilog 發展而成的電路。它定義了與 VGA controller 及 Avalon 匯流排之訊號連接。我們使用 Altera QuartusII 3.0 去編譯及合成此 VGA controller interface 並得到表 3.4 的結果。

Family	LEs	Pins	DSP block	Total memory bits
Stratix	226(2%)	44 (10%)	0(0%)	8192 (<1%)

表 3.4 VGA controller interface 之合成結果