RAiO

RA8889

TFT LCD 文字圖形控制器

規格書

March 24, 2025

RAiO Technology Inc.

Revise History		
Version	**Date**	**Description**
1.0	April 6, 2020	Preliminary Version
1.1	May 13, 2020	1. modify page0/page1 REG [B7h] bit4 2. modify page0/page1 REG [B7h] bit[3:0] 3. remove Figure 16-8 dual-mode1 4. modify Table 16-1 5. Add Chapter 22: Application Circuit
1.2	June 17, 2020	Update Table 5-2 : DC characteristic table
1.3	July 22, 2022	Modify Section 2-5 : Support Various Panel Resolution
1.4	March 24, 2025	Modify Section 6.1.1

CONTENTS

1. 簡介

RA8889 支援 CMOS 準位的介面，規格書內包含: 系統方塊圖、腳位圖、AC/DC 電氣特性、各個功能子方塊、暫存器、省電模式的詳細描述。

1.1 概觀

RA8889 是一款低功耗及顯示功能強大的彩色 TFT 控制器，內部具有記憶體 SDRAM，為了可以快速為顯示記憶體進行螢幕更新，RA8889 支持 MCU 端 8080/6800 8/16-bit 非同步並列介面與 3/4 線 SPI 及 IIC 串列介面，提供多段的顯示記憶體緩衝區段，並提供畫中畫 (PIP)、透明度控制與顯示旋轉鏡像及內建 JPEG Decoder 等功能。

1.2 系統與晶片意示圖

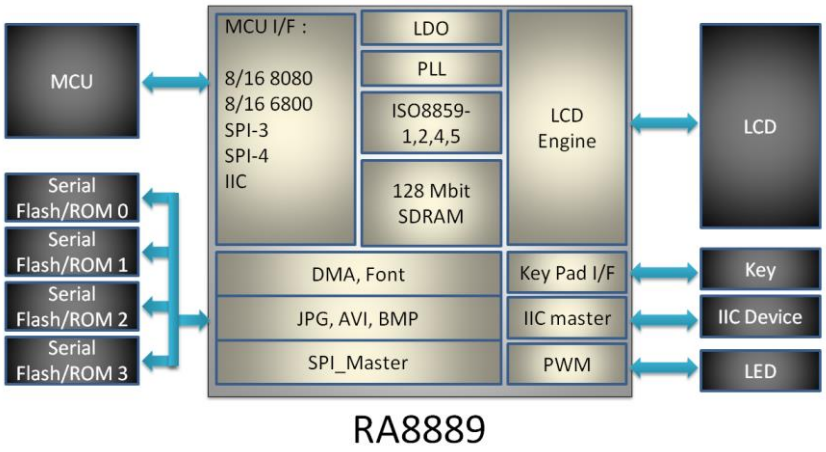


Figure 1-1 : System Diagram

2.

2. 特性

2.1 圖框緩衝區

- 內建 128Mb SDRAM

2.2 主控端介面

- 支援 8080/6800 8/16-bit 非同步並列介面
  - 對於擴展的 MPU 週期提供 Xnwait 的訊號以供交握
- 支援串列主控端介面，例如. IIC, 3/4-wire SPI
- 對於圖像資料寫入支援鏡射與旋轉的功能

2.3 輸入顯示資料格式

- 1bpp: 單色 (1-bit/像素)
- 8bpp: RGB 3:3:2 (1-byte/像素)
- 16bpp: RGB 5:6:5 (2-byte/像素)
- 24bpp: RGB 8:8:8 (3-byte/像素或 4-byte/像素)
  - Index 2:6 (64 索引色/像素並帶透明度屬性)
  - αRGB 4:4:4:4 (4096 索引色/像素並帶透明度屬性)
  - αRGB 8:8:8:8 (8 bit alpha，24bpp 色深)

2.4 顯示模式

- 使用者可以設定 24/18/16-bit TFT 顯示輸出方式

2.5 支援多種螢幕解析度

- 支援 16/18/24-bit CMOS 介面螢幕
- 支持面板解析度，寬度最大支持 1366 像素點，高度為 2048 像素點以下 (註 : 實際的面板解析度是取決於 pixel clock 與色深)

  當 RA8889 TFT Output 支持 24bpp 時

  CCLK Max. = 120MHz

  SCLK Max. = 60MHz

  Required LCD clock ≒ LCD vertical Pixel * LCD horizontal Pixel * 60(Hz) * 1.1

  如果 Required LCD Clock > SCLK，此應用下 LCD 刷新率 Refresh Rate (VSYNC rate) 將會低於 60Hz。

可支持屏幕有:

- QVGA: 320 x 240 x 16/18/24-bit LCD 螢幕
- WQVGA: 480 x 272 x 16/18/24-bit LCD 螢幕
- VGA: 640 x 480 x 16/18/24-bit LCD 螢幕
- WVGA: 800 x 480 x 16/18/24-bit LCD 螢幕
- SVGA: 800 x 600 x 16/18/24-bit LCD 螢幕
- QHD: 960 x 540 x 16/18/24-bit LCD 螢幕
- WSVGA: 1024 x 600 x 16/18/24-bit LCD 螢幕
- XGA: 1024 x 768 x 16/18/24-bit LCD 螢幕
- WXGA: 1280 x 768 x 16/18/24-bit LCD 螢幕
- WXGA: 1280 x 800 x 16/18/24-bit LCD 螢幕
- WXGA: 1366 x 768 x 16/18/24-bit LCD 螢幕

2.6 顯示功能

- 使用者可自行定義 4 個 32X32 圖形游標
- 顯示視窗

  顯示視窗大小是經由定義 LCD 暫存器得到，而透過底圖 (canvas) 暫存器設定可以對顯示視窗進行全部或部分更新。工作視窗的大小與起始位置的解析度在水平上必須是以 8 個像素的倍數，以垂直而言則是 1 個掃描線的倍數。視窗的座標參考原點為左上角(即使在翻轉圖像或旋轉文字時，亦不需要主控端處理)。
- 虛擬顯示

  虛擬顯示可用於顯示大於 LCD 面板尺寸的圖像。 圖像可以在任何方向上輕鬆滾動。
- 畫中畫 (PIP)

  支援兩個畫中畫視窗，當致能畫中畫視窗時則畫中畫視窗會永遠顯示在主視窗中。畫中畫視窗的大小與起始位置水平上是 4 個像素的倍數，垂直上則是一條掃描線。透過設定畫中畫視窗的起始位置可以達成圖像的滾動。 畫中畫 1 的視窗永遠顯示在畫中畫 2 上面。
- 多重緩衝區

  多重緩衝允許在緩衝區之間切換主顯示窗口。 緩衝區的數量取決於欲寫入緩衝區的影像大小。 多重緩衝允許通過切換緩衝區來執行簡單的動畫顯示。
- 喚醒顯示

  喚醒顯示效果如果被致能時，那喚醒時可以快速顯示預先儲存在 SDRAM 中的顯示資料。這個功能是在 Standby 與 Suspend 模式喚醒時使用。
- 水平/垂直翻轉顯示

  水平/垂直翻轉顯示功能只適用在顯示上，對於其他功能子方塊的讀寫是不影響的，在垂直翻轉顯示致能時 PIP 是被禁能的。
- 彩條顯示 (Color Bar Display)

  在沒有 SDRAM 的情況下仍然可以以彩條的方式顯示，預設解析度為 640x480 像素。

2.7 媒體解碼單元 (MDU)

- 自動分辨 JPEG, BMP 和 AVI 格式。
- 支援 JPEG baseline profile,YUV444, YUV422, YUV420, YUV400，不支援 restart interval 格式。
- 支援帶原始數據 (未壓縮) 的標準 BMP 格式。
- 支援 AVI (motion JPEG) 視頻顯示。
- 提供 AVI 顯示的自動播放、暫停和停止功能。

2.8 區塊傳輸引擎 (BTE)

- 2D BitBLT 引擎
- 具有光柵操作與顏色擴展的複製資料
- 方型填滿與圖樣填滿
  - 提供使用者定義的 8x8/16x16 像素的圖樣
- 混合透明 (Opacity)

  使用混合透明模式可以將兩個圖檔混和成新的圖形，然後再用畫中畫的方式顯示出來。在處理的速度上而言混合透明與待處理圖檔大小有關，此外，亦可處理單張圖檔。
  - 關鍵彩度 (Chroma-keying) 功能: 經由指定的 RGB 顏色來做為透明的參考並進行混和影像的處理。
  - 圖形混合透明 (Alpha-blending): 根據暫存器設定透明的比率來進行兩張圖像的混成 (淡入與淡出功能必須被致能)。
  - 像素混合透明 (Alpha-blending): 根據 RGB 格式來混合影像，例如 8bit RGB，則 MSB 2bit 為α值。

2.9 幾何繪圖引擎

- 支援畫點、線、曲線、圓、橢圓、三角形、矩形、圓角矩形

2.10 SPI 主介面

2.10.1 文字功能

- 內建 ISO/IEC 8859-1/2/4/5. 12x24
- 支援集通 16X16/24X24/32X32 串列字型 ROM 例如 Uni-code/BIG5/GB 等等，支援的集通型號有 GT21L16T1W、GT30L16U2W、GT30L24T3Y、GT30L24M1Z、GT30L32S4W、GT20L24F6Y、GT21L24S1W
- 支援使用者自定義字型半形 (8x16/12x24/16x32) 與全型
- 對於寫入文字支援可程式文字游標
- 支援垂直水平放大字型 X1, X2, X3, X4 倍數
- 支援文字 90 度旋轉

2.10.2 DMA 功能

- 支援外部串列快閃記憶體 (serial flash) 資料複製至圖框緩衝區
- 支援外部快閃記憶體 Signle/Dual/Quad mode

2.10.3 一般主 SPI 功能

- 相容 Motorola SPI 規格
- 16 bytes 讀取深度的 FIFO
- 16 bytes 寫入深度的 FIFO
- 在 Tx FIFO 完全清空並且 SPI Tx/Rx 引擎閒置時會發出中斷

2.10.4 IDEC 功能

- 支援外部串列快閃記憶體 (serial flash) 資料透過 MDU 至圖框緩衝區
- 支援外部快閃記憶體 Quad mode

2.11 IIC 介面

- IIC master interface
  - 可以使用在擴充 I/O device，例如在螢幕控制的觸控螢幕
  - 支援標準模式 (100kbps) 與快速模式 (400kbps)

2.12 脈寬調變與計時器

- 內建兩個 16-bit 計數器
- 一個 8-bit pre-scalars 與一個 4-bit 除頻
- 輸出波形的工作週期是可程式化的
- 自動重載入模式或單擊模式
- Dead-Zone 保護

2.13 按鍵介面

- 支援 5x5 鍵盤 (必須使用與 GPIO 的共用腳)
- 可程式化的掃描周期
- 支援長按鍵與重覆鍵
- 支援同時按兩鍵

  註: 在限制條件下可以支援同時按 3 鍵 (3 個鍵線段組成角度必須不是 90°)
- 支援鍵盤喚醒功能

2.14 省電模式

- 支援 3 種省電模式
  - 待機 (Standby)、休眠 (Suspend) 與睡眠 (Sleep) 模式
- 可以使用主控端、按鍵、外部事件喚醒

2.15 時脈來源

- 內建可程式鎖相回路 PLL 以提供系統時脈、LCD 掃描時脈與 SDRAM 時脈使用
- 單一石英晶體震盪輸入: (XI/XO: 10MHz)
- 內部核心最大系統時脈 (最大值 120MHz)
- SDRAM 時脈 (最大值 166MHz)
- LCD 螢幕掃描時脈 (最大值 100MHz)

2.16 重置

- 接受外部硬體重置
- 軟體命令重置

2.17 電源

- I/O 電壓: 3.3V +/- 0.3V
- 內建 1.2V LDO for core power

2.18 封裝

- LQFP-100
- 操作溫度: -40℃ ~ 85℃

3. 產品封裝

3.1 RA8889 封裝腳位圖

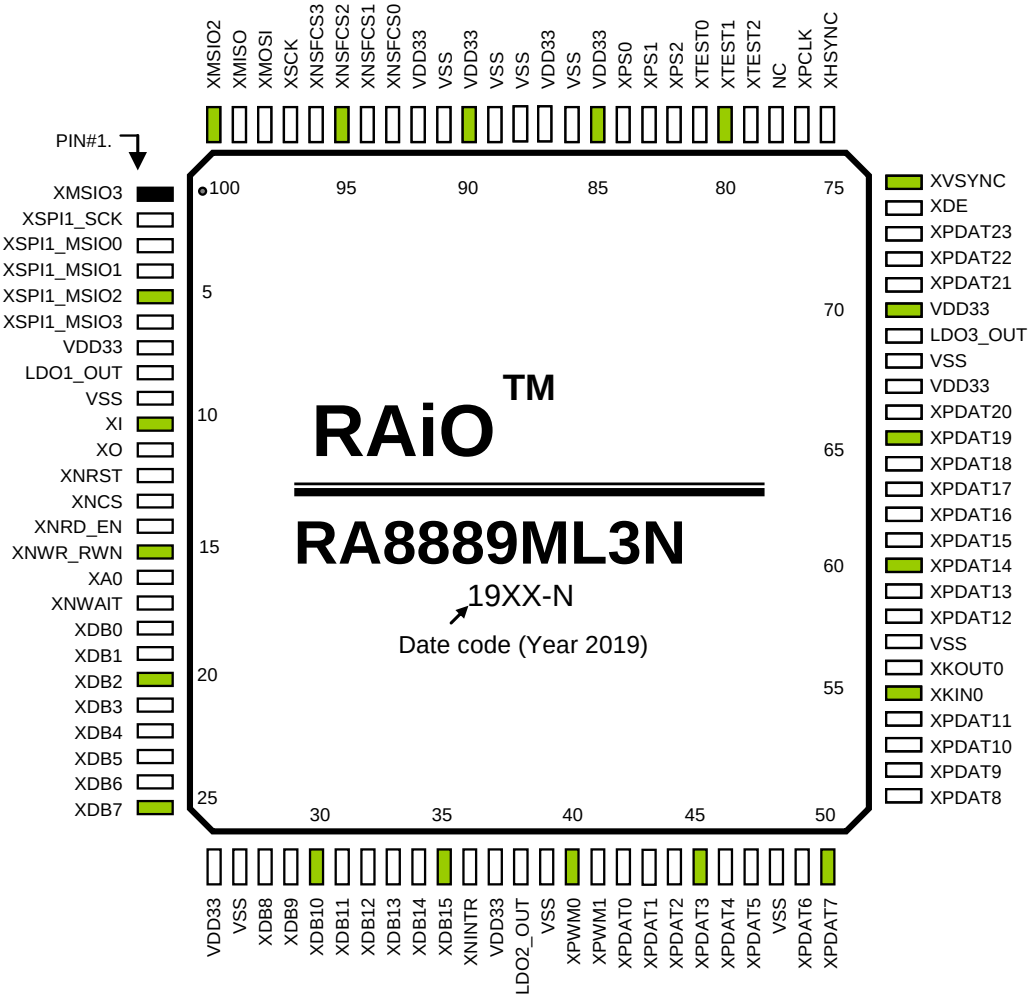
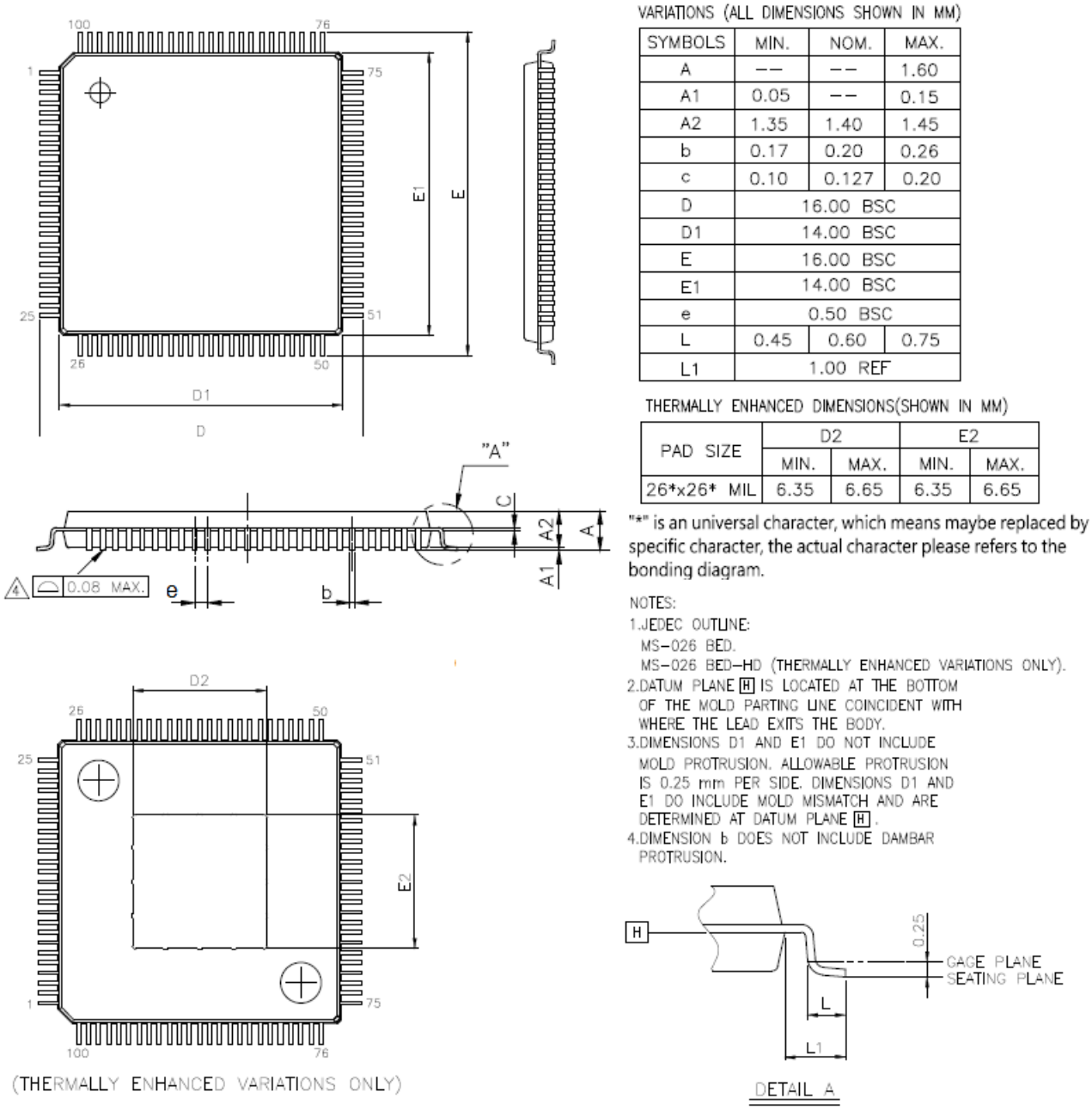


Figure 3-1

3.2 封裝尺寸



VARIATIONS (ALL DIMENSIONS SHOWN IN MM)

SYMBOLS	MIN.	NOM.	MAX.
A	--	--	1.60
A1	0.05	--	0.15
A2	1.35	1.40	1.45
b	0.17	0.20	0.26
c	0.10	0.127	0.20
D	16.00 BSC		
D1	14.00 BSC		
E	16.00 BSC		
E1	14.00 BSC		
e	0.50 BSC		
L	0.45	0.60	0.75
L1	1.00 REF		

THERMALLY ENHANCED DIMENSIONS(SHOWN IN MM)

PAD SIZE	D2		E2	
	MIN.	MAX.	MIN.	MAX.
26*x26* MIL	6.35	6.65	6.35	6.65

"*" is an universal character, which means maybe replaced by specific character, the actual character please refers to the bonding diagram.

NOTES:
1.JEDEC OUTLINE:
MS-026 BED.
MS-026 BED-HD (THERMALLY ENHANCED VARIATIONS ONLY).
2.DATUM PLANE [H] IS LOCATED AT THE BOTTOM OF THE MOLD PARTING LINE COINCIDENT WITH WHERE THE LEAD EXITS THE BODY.
3.DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 mm PER SIDE. DIMENSIONS D1 AND E1 DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE [H].
4.DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

Figure 3-2 : RA8889 Package Outline Dimensions

4. 腳位定義

4.1 並列主控端介面 (25 腳位)

接 腳 名 稱	I/O	腳 位 說 明
XDB[15:0]	IO (8mA)	**資料匯流排** 資料匯流排提供主控端與RA8889 的並列介面資料傳送。 XDB[15:8] 可以設定GPIO (GPIO-A[7:0])，前提是沒有設定成 8080/6800 16-bits並列介面資料匯流排。 XDB[7:0] 如果在串列主控端模式下，此訊號也提供為串列的主控端訊號使用，請參考串列主控端介面章節。
XA0	I	**命令/資料 選擇** 此腳位被使用在選擇命令還是資料的周期。 XA0 = 0，狀態讀取/命令寫入。 XA0 = 1，資料讀取/資料寫入。
XNCS	I	**晶片致能** 低電位致能，如果主控端設定 RA8889 為串列主控端模式，則此腳位設定為 GPI-B0 並且讀取腳位的值，腳位內部有提升電阻。
XNRD_EN (XEN)	I	**致能/讀取致能** 當微處理器是 8080 系列，此腳位是當作 XnRD 使用 (讀取資料)，低電位動作。 當微處理器是 6800 系列，此腳位是當作 XEN 使用 (致能信號)，高電位動作。 如果主控端介面設定成串列主控模式，那麼此腳位則為 GPI-B1，並且可讀取腳位上的電壓值。 內建提升電阻。
XNWR_RWN (XRnW)	I	**寫入/讀寫** 當微處理器介面是 8080 系列，此腳位會成為 XnWR (資料寫入)，低電位動作。 當微處理器介面是 6800 系列，此腳位會成為 XRnW (資料 讀取/寫入)，讀取時是高電位動作，寫入是低電位動作。 如果主控端介面是設定成串列主控模式，那麼此腳位將會成為 GPI-B2。 內建提升電阻。
XNINTR	O (8mA)	**中斷信號輸出** 告知主控端目前內部狀態的中斷輸出。
XNWAIT	O (8mA)	**等待信號輸出** 當 XnWAIT 為 high，表示 RA8889 已經準備好傳輸資料，當 XnWAIT 為 low，微處理器應該進入等待周期。
XPS[2:0]	I	**並列/串列 主控端介面選擇** 00X: (並列主控端) 8080 8/16-bits 資料匯流排介面。 01X: (並列主控端) 6800 8/16-bits 資料匯流排介面。 100: (串列主控端) 3-wire SPI。 101: (串列主控端) 4-wire SPI。 11x: (串列主控端) IIC。 ***註*:** 如果主控端介面設定成並列主控端模式，那麼 XPS[0] 為外部中斷接腳。

4.2 串列主控端介面 (與並列主控端介面)

接 腳 名 稱	I/O	腳 位 說 明
XSSCL (XDB[7])	I	**SPI 與 IIC 時脈** XSSCL、3-wire、4-wire 串列或 IIC 介面時脈。
XSSDI XSSDA (XDB[6])	I	**IIC 資料/4-wireSPI 資料輸入** 3-wire SPI 介面: NC，請連接到 GND。 4-wire SPI 介面: XSSDI 串列介面資料輸入。 IIC 介面: XSSDA 串列介面輸入輸出雙向。
XSSD XSSDO (XDB[5])	IO	**3-wire SPI 資料/4-wireSPI 資料輸出/IIC Slave 位置選擇** 3-wire SPI I/F: XSSD，串列介面輸入輸出雙向資料傳輸。 4-wire SPI I/F: XSSDO，串列介面資料輸出。 IIC 介面: XIICA[5]，IIC 裝置位址 bit [5]。
XnSCS (XDB[4])	I	**SPI 致能/IIC Slave 位址選擇** XnSCS, 在 3-wire 與 4-wireSPI 串列介面中，此腳位為致能訊號。 IIC 介面: XIICA[4]，IIC 裝置位址 bit [4]。
XIICA[3:0] (XDB[3:0])	I	**IIC 介面: IIC Slave 位址選擇** XIICA[3:0]，在 3-wire 與 4-wire SPI 介面: NC，請連接到 GND。 IIC 介面: IIC 裝置位址 bit [3:0]。

4.3 Serial Flash 或 SPI master 介面 (14 腳位)

接 腳 名 稱	I/O	腳 位 說 明
XNSFCS0	IO (8mA)	**外部 Serial Flash/ROM SPI 晶片選擇 0** SPI 晶片選擇接腳#0 使用在 Serial Flash/ROM 或 SPI 裝置選擇上。 *如果 SPI master 被禁能，那麼此腳位可以被程式規劃成 GPIO (GPIO-C3)，預設 GPIO-C3 為輸入功能。
XNSFCS1	IO (8mA)	**外部 Serial Flash/ROM SPI 晶片選擇 1** SPI 晶片選擇腳#1 使用在 Serial Flash/ROM 或 SPI 裝置選擇上。 * 如果 SPI master 被禁能，那麼此腳位可以被程式規劃成 GPIO (GPIO-C4)，預設 GPIO-C4 為輸入功能。 *如果 xtest[2:1]不等於 01b 那麼在 reset 週期時會自動 pull-high。
XNSFCS2	IO (8mA)	**外部 Serial Flash/ROM SPI 晶片選擇 2** SPI 晶片選擇腳#2 使用在 Serial Flash/ROM 或 SPI 裝置選擇上。
XNSFCS3	IO (8mA)	**外部 Serial Flash/ROM SPI 晶片選擇 3** SPI 晶片選擇腳#3 使用在 Serial Flash/ROM 或 SPI 裝置選擇上。
XSCK	IO (8mA)	**SPI 串列時脈** 此腳位是串列時脈輸出，主要是給 Serial Flash/ROM 或 SPI 裝置使用。 * 如果 SPI master 介面被禁能，那麼此腳位可以被程式規劃為 GPIO (GPIO-C0); 預設 GPIO-C0 為輸入功能。

接 腳 名 稱	I/O	腳 位 說 明
XMOSI (XSIO0)	IO (8mA)	**主輸出從輸入** Single 模式: Serial Flash/ROM 或 SPI 裝置輸入資料用。對 RA8889 而言此腳為輸出。 Dual 模式: 此腳位為雙向資料傳送#0 (SIO0)，此功能只能在 Serial flash DMA 使用。 *如果 SPI master 介面被禁能，那麼此腳位可以被程式規劃為 GPIO (GPIO-C1); 預設 GPIO-C1 為輸入功能。
XMISO (XSIO1)	IO (8mA)	**主輸入從輸出** Single 模式: Serial Flash/ROM 或 SPI 裝置輸出資料用。對 RA8889 而言此腳為輸入。 Dual 模式: 此腳位為雙向資料傳送#1 (SIO1)。此功能只能在 Serial flash DMA 使用。 *如果 SPI master 介面被禁能，那麼此腳位可以被程式規劃為 GPIO (GPIO-C2)，預設 GPIO-C2 為輸入功能。
XSIO2	IO (8mA)	**從輸入 IO2** Qaud mode: Serial Flash/ROM 或 SPI 裝置輸出資料用. 對 RA8889 而言此腳為輸入。
XSIO3	IO (8mA)	**從輸入 IO3** Qaud mode: Serial Flash/ROM 或 SPI 裝置輸出資料用. 對 RA8889 而言此腳為輸入。
XSPI1_SCK	IO (8mA)	**SPI 串列時脈 (SPI 1)** 此腳位是串列時脈輸出，主要是給 Serial Flash/ROM 或 SPI 裝置使用。 * 如果 SPI master 介面被禁能，那麼此腳位可以被程式規劃為 GPIO (GPIO-C0); 預設 GPIO-C0 為輸入功能。
XSPI1_MSIO0	IO (8mA)	**主輸出從輸入 (SPI 1)** Single 模式: Serial Flash/ROM 或 SPI 裝置輸入資料用。對 RA8889 而言此腳為輸出。 Dual 模式: 此腳位為雙向資料傳送#0 (SIO0)，此功能只能在 Serial flash DMA 使用。 *如果 SPI master 介面被禁能，那麼此腳位可以被程式規劃為 GPIO (GPIO-C1); 預設 GPIO-C1 為輸入功能。
XSPI1_MSIO1	IO (8mA)	**主輸入從輸出** Single 模式: Serial Flash/ROM 或 SPI 裝置輸出資料用。對 RA8889 而言此腳為輸入。 Dual 模式: 此腳位為雙向資料傳送#1 (SIO1)。此功能只能在 Serial flash DMA 使用。 *如果 SPI master 介面被禁能，那麼此腳位可以被程式規劃為 GPIO (GPIO-C2)，預設 GPIO-C2 為輸入功能。

接 腳 名 稱	I/O	腳 位 說 明
XSPI1_MSIO2	IO (8mA)	**從輸出 IO 2 (SPI 1)** Qaud mode: Serial Flash/ROM 或 SPI 裝置輸出資料用. 對 RA8889 而言此腳為輸入。
XSPI1_MSIO3	IO (8mA)	**從輸出 IO 3 (SPI 1)** Qaud mode: Serial Flash/ROM 或 SPI 裝置輸出資料用. 對 RA8889 而言此腳為輸入。

4.4 PWM 介面 (2 腳位)

接 腳 名 稱	I/O	腳 位 說 明
XPWM0	IO (8mA)	**PWM 訊號輸出 1** XPWM 0 的輸出模式可以在暫存器中指定。 如果 PWM 被禁能，那麼此腳位可以被程式規劃為 GPIO (GPIO-C7)，預設 GPIO-C7 是輸入功能或是輸出核心時脈。
XPWM1 (XCLK3)	IO (8mA)	**PWM 訊號輸出 2 / 時脈 3 輸入(螢幕掃描時脈)** 當 XTEST[0]為低電位時: XPWM1 可以被設定為輸出其輸出模式可經由暫存器設定來完成。那麼其輸出可以指定為標準的 XPWM1 功能，oscillator 時脈輸出或是 SCAN 頻寬不足與超過記憶體位址的錯誤旗標。 當 XTEST[0] 為高電位時: XPWM1 腳位就是外部螢幕掃描時脈 3 輸入。

4.5 鍵盤掃描 (10 腳位)

接 腳 名 稱	I/O	腳 位 說 明
XKIN[4:0]	I	**按鍵資料或 GPIs (General Purpose Input)** 按鍵資料輸入 (預設)，內建 pull-up 電阻。 XKIN[0] 具有 IIC master 的 XSCL 功能。 **In RA8889，XKIN [4:1] 與 XPDAT、GPIO-D 共用腳位。**
XKOUT[4:0]	O (2mA)	**Keypad 閃控或 GPOs (General Purpose Output)** Keypad 矩陣使用閃控掃描鍵盤，腳位上為 open-drain 輸出 (預設)。 XKOUT[0] 具有 IIC master 的 XSDA 功能。 **In RA8889，XKOUT [4:1] 與 XPDAT、GPIO-D 共用腳位。**

4.6 LCD 螢幕數位介面 (28 腳位)

<table>
<tr><th>接 腳 名 稱</th><th>I/O</th><th>腳 位 說 明</th></tr>
<tr><td>XPCLK</td><td>O
(8mA)</td><td>螢幕掃描時脈
螢幕掃描時脈相容於通用的 TFT 介面訊號。
此訊號為 SPLL 驅動產生。</td></tr>
<tr><td>XVSYNC</td><td>O
(4mA)</td><td>VSYNC Pulse
垂直同步訊號 VSYNC 相容於通用的 TFT 介面訊號。</td></tr>
<tr><td>XHSYNC</td><td>O
(4mA)</td><td>HSYNC Pulse
水平同步訊號 HSYNC 相容於通用的 TFT 介面訊號。</td></tr>
<tr><td>XDE</td><td>O
(4mA)</td><td>Data 致能
通用 TFT 介面的 data 有效或 data 致能信號。</td></tr>
<tr><td>XPDAT
[23:0]</td><td>IO
(4mA)</td><td>LCD 螢幕資料匯流排
輸出資料至 TFT LCD 資料匯流排，RA8889 可經由暫存器設定以支援 64K/262K/16.7M 色深，使用者可以經由不同的設定連結相對應的 RGB 匯流排。
<table>
<tr><td>Pin Name</td><td colspan="4">Digital TFT Interface</td></tr>
<tr><td>TFT output Setting</td><td>11b
(GPIO)</td><td>10b
(16-bits)</td><td>01b
(18-bits)</td><td>00b
(24-bits)</td></tr>
<tr><td>XPDAT[0]</td><td colspan="3">GPIO-D0/ XKIN[1]</td><td>B0</td></tr>
<tr><td>XPDAT[1]</td><td colspan="3">GPIO-D1/ XKIN[2]</td><td>B1</td></tr>
<tr><td>XPDAT[2]</td><td colspan="2">GPIO-D6/ XKIN[4]</td><td>B0</td><td>B2</td></tr>
<tr><td>XPDAT[3]</td><td>GPIO-E0</td><td>B0</td><td>B1</td><td>B3</td></tr>
<tr><td>XPDAT[4]</td><td>GPIO-E1</td><td>B1</td><td>B2</td><td>B4</td></tr>
<tr><td>XPDAT[5]</td><td>GPIO-E2</td><td>B2</td><td>B3</td><td>B5</td></tr>
<tr><td>XPDAT[6]</td><td>GPIO-E3</td><td>B3</td><td>B4</td><td>B6</td></tr>
<tr><td>XPDAT[7]</td><td>GPIO-E4</td><td>B4</td><td>B5</td><td>B7</td></tr>
<tr><td>XPDAT[8]</td><td colspan="3">GPIO-D2/ XKIN[3]</td><td>G0</td></tr>
<tr><td>XPDAT[9]</td><td colspan="3">GPIO-D3/ XKOUT[3]</td><td>G1</td></tr>
<tr><td>XPDAT[10]</td><td>GPIO-E5</td><td>G0</td><td>G0</td><td>G2</td></tr>
<tr><td>XPDAT[11]</td><td>GPIO-E6</td><td>G1</td><td>G1</td><td>G3</td></tr>
<tr><td>XPDAT[12]</td><td>GPIO-E7</td><td>G2</td><td>G2</td><td>G4</td></tr>
<tr><td>XPDAT[13]</td><td>GPIO-F0</td><td>G3</td><td>G3</td><td>G5</td></tr>
<tr><td>XPDAT[14]</td><td>GPIO-F1</td><td>G4</td><td>G4</td><td>G6</td></tr>
<tr><td>XPDAT[15]</td><td>GPIO-F2</td><td>G5</td><td>G5</td><td>G7</td></tr>
<tr><td>XPDAT[16]</td><td colspan="3">GPIO-D4/ XKOUT[1]</td><td>R0</td></tr>
<tr><td>XPDAT[17]</td><td colspan="3">GPIO-D5/ XKOUT[2]</td><td>R1</td></tr>
<tr><td>XPDAT[18]</td><td colspan="2">GPIO-D7/ XKOUT[4]</td><td>R0</td><td>R2</td></tr>
<tr><td>XPDAT[19]</td><td>GPIO-F3</td><td>R0</td><td>R1</td><td>R3</td></tr>
<tr><td>XPDAT[20]</td><td>GPIO-F4</td><td>R1</td><td>R2</td><td>R4</td></tr>
<tr><td>XPDAT[21]</td><td>GPIO-F5</td><td>R2</td><td>R3</td><td>R5</td></tr>
<tr><td>XPDAT[22]</td><td>GPIO-F6</td><td>R3</td><td>R4</td><td>R6</td></tr>
<tr><td>XPDAT[23]</td><td>GPIO-F7</td><td>R4</td><td>R5</td><td>R7</td></tr>
</table>
*未使用的腳位可以被程式規劃成 GPIO-D/E/F(預設) 或 XKIN/XOUT，
預設是 18bpp 色深模式，因此 XPDAT[17:16/8:9/1:0] 預設是 GPI 模式。</td></tr>
</table>

4.7 時脈與重置與測試模式 (6 腳位)

接 腳 名 稱	I/O	腳 位 說 明
XI (XCLK1)	I	**Crystal 輸入/Clock 1 輸入(核心時脈-core clock)** Crystal Oscillator 必須是在 10MHz。 當 XTEST[0]設為低電位時，此腳位是給內部的 crystal 電路使用，而此腳位應該連接外部 crystal 電路，這將可以產生 RA8889 的時脈訊號。 當 XTEST[0]設為高電位時，此腳位被拿來當作外部時脈 1 輸入。
XO	O	**Crystal 輸出** 此腳位為內部 crystal 電路輸出，而此腳位應該連接至外部 crystal 電路。
XNRST	I/OC	**重置輸入訊號** 為了避免雜訊產生錯誤的重置訊號，外部重置訊號的準位必須最少要有 256 OSC 的時脈周期。
XTEST[0]	I	**時脈測試模式** 內建 pull down 電阻 此腳位是提供給晶片測試使用的，在標準操作上此腳位應該要連接至 GND。 0: 標準模式，使用內部 PLL 時脈。 1: 忽略 PLL，晶片時脈改使用外部 CLK1I、CLK2I、CLK3I 輸入。
XTEST[2:1]	I	**晶片測試模式** 00: 標準模式。 01: 令 SPI master 腳位浮接 (使用在 in-system-programming)。 1X: 保留。

4.8 電源與接地

接 腳 名 稱	I/O	腳 位 說 明
LDO1_OUT **LDO2_OUT** **LDO3_OUT**	P	**LDO 外接電容** 外接 1uF 電容到地。
VDD33	P	**IO VDD** 3.3V IO 電源輸入。
VSS	P	**GND** IO Cell/Core 接地。

5. AC/DC 特性

5.1 最大範圍限制

Table 5-1 :最大額定值

Parameter	Symbol	Value	Unit
Supply Voltage Range	V_{DD33}	-0.3 ~ 4.0	V
Input Voltage Range	V_{IN}	-0.3 ~ V_{DD33}+0.3	V
Operation Temperature Range	T_{OPR}	-40 ~ 85	℃
Power Dissipation	P_D	≦300	mW
Storage Temperature	$T_{Storage}$	-45 ~ 125	℃
Soldering Temperature	T_{Solder}	260	℃

註：

1. 假如該封裝被焊料侵入，平薄式封裝的濕度抵抗性是會減少的。當進行焊接作業時，勿過度施壓於封裝上。
2. 當供應電源端為高阻抗時，供應電源/輸入電源可能存在著一個很大壓差，須適度考量RA8889的電源端及其電源接線及佈局。

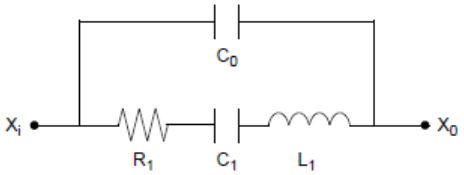
5.2 DC 特性

Table 5-2 : DC 電性特性

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
System Voltage	V_{DD33}	3.0	3.3	3.6	V	
Loading capacitor	Cap_{Vdd}	1	-	10	uF	
Operation Current	I_{OPR}		126		mA	Note 3, Note 4
Standby mode	I_{Stdby}		43		mA	Note 3
Suspend mode	I_{Susp}		14.5		mA	Note 3
Sleep Mode	I_{SLP}		6.3		mA	Note 3
Power ramp up time	T_{ramp}	3.5		35	ms	V_{DD33} Ramp up to 3.3 V
OSC/PLL						
Oscillator Clock	F_{OSC}		10		MHz	V_{DD33} = 3.3 V, Note 1
Clock Period Jitter	T_{jit_period}	-2.5		2.5	%	
Lock Time	T_{Lock}		1024		*OSC*	Note 2
MPLL Output Clock (MCLK)	$Freq_{mclk}$			166	MHz	V_{DD33}= 3.3 V
CPLL Output Clock (CCLK)	$Freq_{cclk}$			133	MHz	V_{DD33} = 3.3 V Without enable internal ISO-8859 font feature
CPLL Output Clock (CCLK)	$Freq_{cclk}$			120	MHz	V_{DD33} = 3.3 V When enable internal ISO-8859 font feature
SPLL Output Clock (PCLK)	$Freq_{pclk}$			100	MHz	V_{DD33}= 3.3 V
Serial MPU I/F						
SPI Input clock	$Freq_{xssck}$			50	MHz	
Input/Output (CMOS 3-state Output pad with Schmitt Trigger Input, Pull-Up/Down)						
Input High Voltage	V_{IH}	2		3.6	V	
Input Low Voltage	V_{IL}	-0.3		0.8	V	
Output High Voltage	V_{OH}	2.4			V	

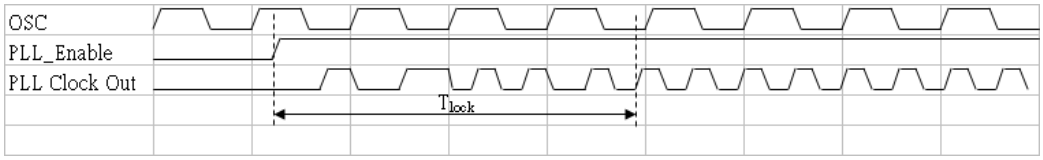
Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Output Low Voltage	V_{OL}			0.4	V	
Pull up resistance	R_{PU}	20		80	Kohm	
Pull down resistance	R_{PD}	20		80	Kohm	
Schmitt trigger low to high threshold	V_t+	1.5		2.1	V	
Schmitt trigger high to low threshold	V_t-	0.8		1.3	V	
Hysterisis	V_{hys}	200			mV	
Input Leakage Current	I_{leak}	-10		+10	µA	
Rise/fall slew rate	Slew		1.5		V/ns	

Note 1: 使用 Crystal Oscillator 時的寄生電容效應



Typical: R1 = 50ohm(25-100ohm), L1 =3.4mH, C1 =13fF, C0 =2.8pF

Note 2:



Note 3: Measured on tester with 8 bit MPU interface and without extra load.

Note 4: Measured on tester with Resolution 160x120, MCLK=151Mhz, CCLK=107Mhz, SCLK=60Mhz.

6. 時脈重置

6.1 時脈

RA8889 針對不同的功能方塊內建 3 PLL，舉例 CPLL 產生 CCLK 提供 MPU 介面、媒體解碼單元 (MDU)、BTE 引擎、繪圖引擎、文字 DMA 引擎使用；MPLL 則產生內部 MCLK 以提供給 SDRAM 使用；SPLL 則產生 SCLK 提供 LCD 螢幕掃描工作時脈。

6.1.1 時脈架構

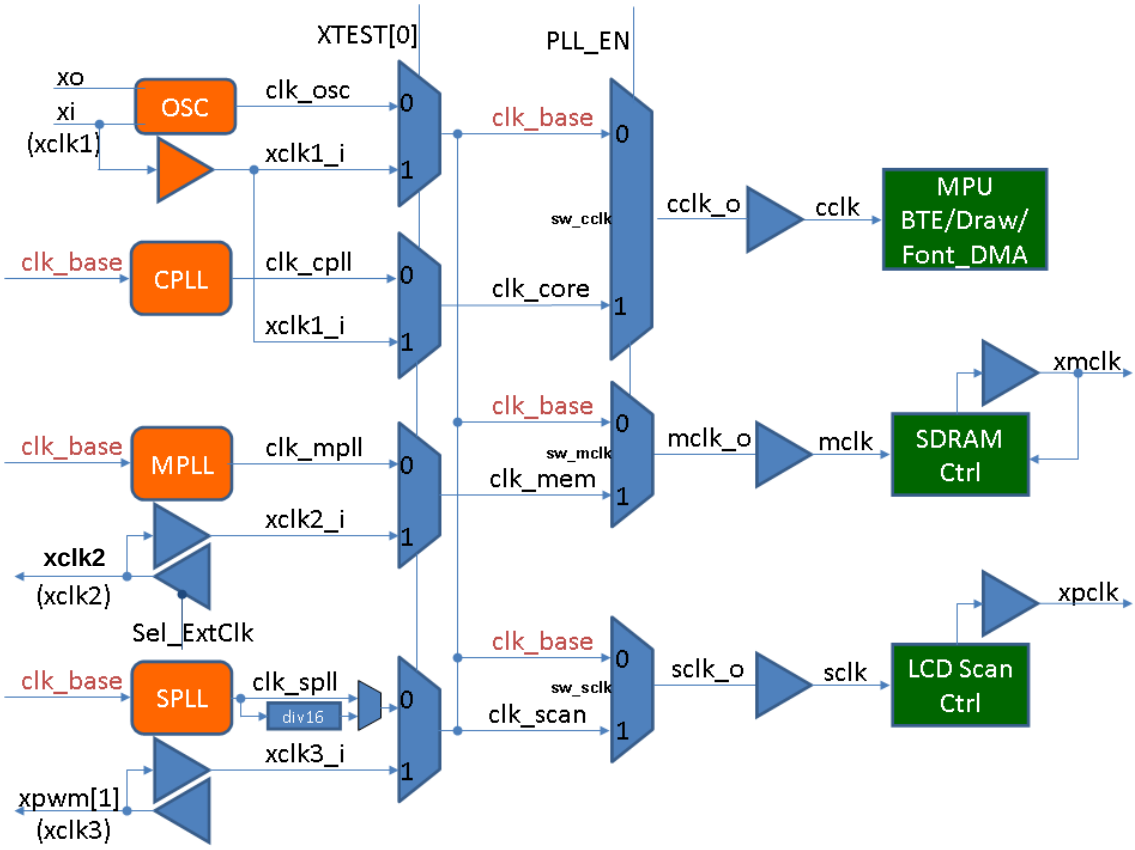


Figure 6-1

XTEST[0] 控制內部的 PLL 或是外部的時脈來產生主要的時脈，設定 XTEST[0] 為低電位將可以選擇 CPLL、MPLL、SPLL 為核心時脈、記憶體時脈、螢幕掃描時脈的來源。設定 XTEST[0] 為高電位則選擇 xi (xclk1), xclk2 and xpwm[1] (xclk3) IO 腳位為核心時脈、記憶體時脈、螢幕掃描時脈的來源。

時脈必須滿足以下條件

- Color depth = 16bpp:

  $Freq_{MCLK} \geq Freq_{CCLK} \geq Freq_{SCLK} * 1.5$

- Color depth = 24bpp:

  $Freq_{MCLK} \geq Freq_{CCLK} \geq Freq_{SCLK} * 2$

6.1.2 PLL 設定

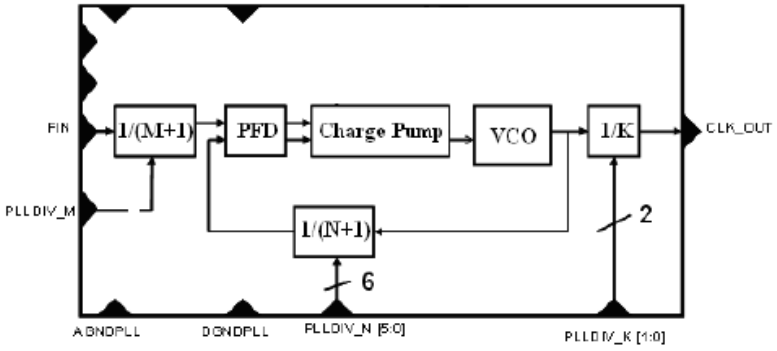


Figure 6-2

對於 PLL 鎖相迴路，輸出頻率可以透過以下暫存器 PLLDIVM、PLLDIVN、PLLDIVK。輸出頻率公式如下:

$$xCLK = \frac{\left(Fin / 2^{(xPLLDIVM)}\right) \times (xPLLDIVN + 1)}{2^{xPLLDIVK}}$$

除頻範圍：

i. PLLDIVM : 0 or 1

ii. PLLDIVN[5:0] : 1~63 (minimum≧1)

iii. PLLDIVK[1:0] : 0~3

註：

1. 只有在 PLL 禁能時才能修改 PLL 參數。
2. 在 REG[05h] ~ REG[0Ah] 被修改後，PLL 需要有 30us 的時間來穩定頻率輸出。
3. 輸入 OSC 頻率 F_{IN} 與 PLLDIVM 必須符合以下條件：

$$Fin = 10MHz$$
$$\&$$
$$10MHz \le \frac{Fin}{2^{PLLDIVM}} \le 40MHz$$

4. 內部倍頻 $Fvco = \frac{Fin}{2^{PLLDIVM}} \times (PLLDIVN + 1)$ 必須是大於等於 250 MHz 並且小於 500MHz。

   i.e,

$$250MHz \le Fvco \le 500MHz$$

Example: MCLK = 140Mhz,CCLK=120Mhz,SCLK=35Mhz,相關暫存器設定如下

Registers	MCLK (DRAM clock) = 140MHz
	CCLK (Core clock) = 120MHz
	SCLK (LCD scan clock) = 35MHz for TFT LCD resolution 800x480
PAGE0 REG[05h]	0x06
PAGE0 REG[06h]	0x1B
PAGE0 REG[07h]	0x02
PAGE0 REG[08h]	0x1B
PAGE0 REG[09h]	0x04
PAGE0 REG[0Ah]	0x2F
Note : OSC = 10MHz	

6.2 重置

6.2.1 外部重置訊號

RA8889 為了同步外部系統因此可以接收外部重置訊號 (低電位動作) 以達成同步化，外部重置訊號必須最少有 256 OSC 時脈週期，才會被認可為合法的重置訊號。

在使用 RA8889 時，MPU 應該要先確認 status 暫存器 bit [1]-operation mode status bit，這樣可以確定 RA8889 是否在標準操作狀態。

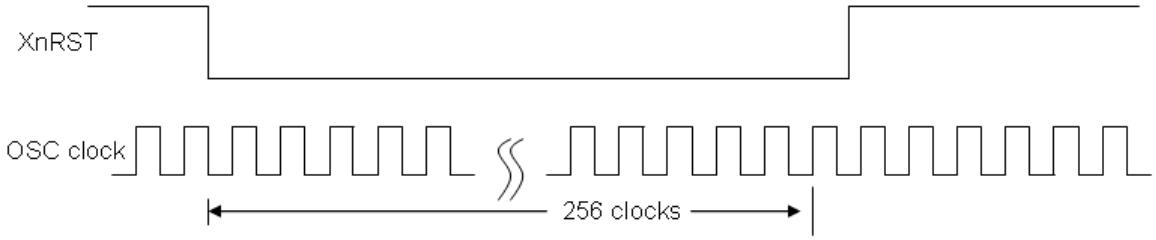


Figure 6-3：外部重置訊號需大於 256 OSC 週期

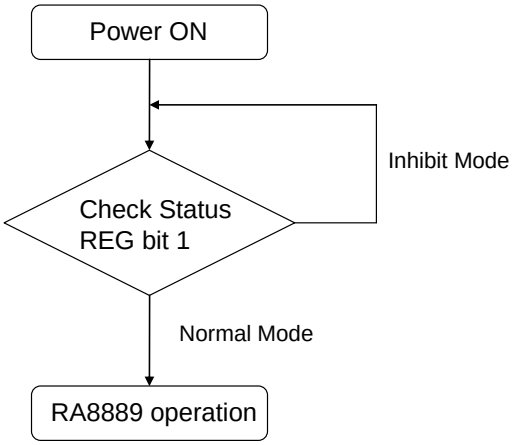


Figure 6-4：開機後使用 RA8889 必須先檢查 Status REG bit 1 狀態

7. 主控端介面

7.1 間接介面

RA8889 提供並列主控端介面(ex. 8080/6800 系列的 MPU 介面)與串列主控端介面(ex. IIC, 3-wire/4-wire SPI)。MPU 介面型態由 XPS[2:0] 指定，經由數個步驟可以達成以非同步的方式使用 RA8889。而暫存器與記憶體存取可以由暫存器空間得到。

註 **--**

除了記憶體存取外，所有暫存器的存取資料寬度皆為 8-bit。即使 MPU 寬度為 16bit 亦是如此，如果 MPU 寬度是 16bit，那麼暫存器的資料寬度仍然是 LSB (XDB[7:0]) 8-bit，但是 Memory Data Port (REG[04h]) 暫存器除外。當使用 Memory Data Port (REG[04h]) 暫存器時必須同時參考 host interface type bit (REG[01h], bit[0]) 暫存器，當 host interface type bit (REG[01h], bit[0]) 暫存器設定成 16-bits 寬度，那麼記憶體資料寬度就為 16bit 寬度，若是暫存器 (REG[01h], bit[0]) 設成 8bit 寬度則記憶體資料寬度就為 8-bit。

Table 7-1 : Parallel /Serial Host I/F Select Pin

XPS[2:0]	Host Mode
00X	(parallel host) 8080 interface with 8/16-bits data bus
01X	(parallel host) 6800 interface with 8/16-bits data bus
100	(serial host) 3-Wire SPI
101	(serial host) 4-Wire SPI
11X	(serial host) IIC

存取並設定暫存器的方法，第一步傳送 “Address Write” 來設定暫存器位址。下一步再處理 “Data Read/Write” 的部分，這樣就可以將指定的資料經由暫存器或記憶體作寫入或讀取，如果做連續資料讀寫的動作而沒去更改暫存的位址，那麼暫存器位址是不會自動增加的；如果連續資料讀寫的動作是作用在 Memory Data Port (REG[04h])，暫存位址不會自動增加，但是內部記憶體電路位址會自動增加。如果要顯示一個視窗，可以指定視窗的座標並且針對記憶體用連續資料的寫入，那麼內部的記憶體位址將會自動的遞增。

7.1.1 暫存器寫入

1. 寫入要處理的暫存器位址bits 7-0。
2. 寫入暫存器資料。

7.1.2 暫存器讀取

1. 寫入要處理的暫存器位址bits 7-0。
2. 讀取暫存器資料。

7.1.3 記憶體寫入

RA8889 有最簡單的方法可以達成圖像資料寫入記憶體中。

1. 寫入圖像資料前先設定工作視窗。
2. 設定圖像的讀取寫入座標 REG[5Fh]~REG[62h]。
3. 設定 Memory Data Port Register (REG[04h]) 完成位址設定。
4. 對工作視窗寫入資料，每個資料寫入 Memory Data Port 都將會自動累加記憶體位址。

7.2 並列主控端

7.2.1 並列主控端介面

8080 與 6800 系列的 MPU 介面接線圖，請參考 Figure 7-1 與 Figure 7-2。

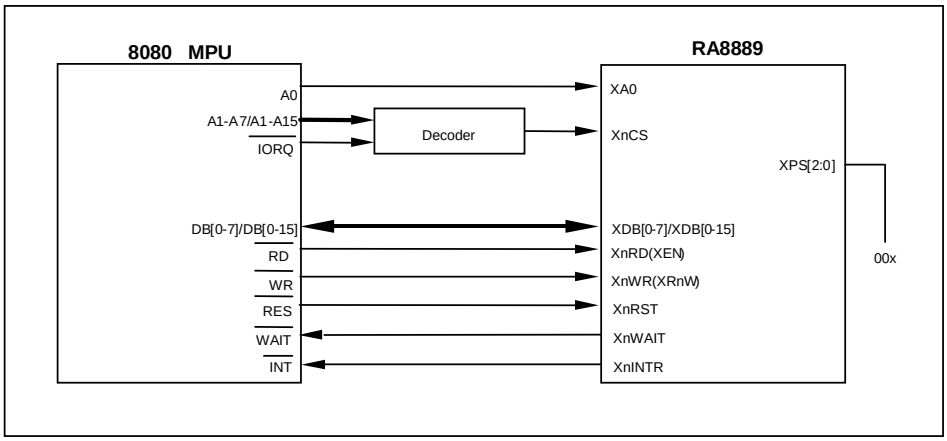


Figure 7-1 : 8080 MPU Interface

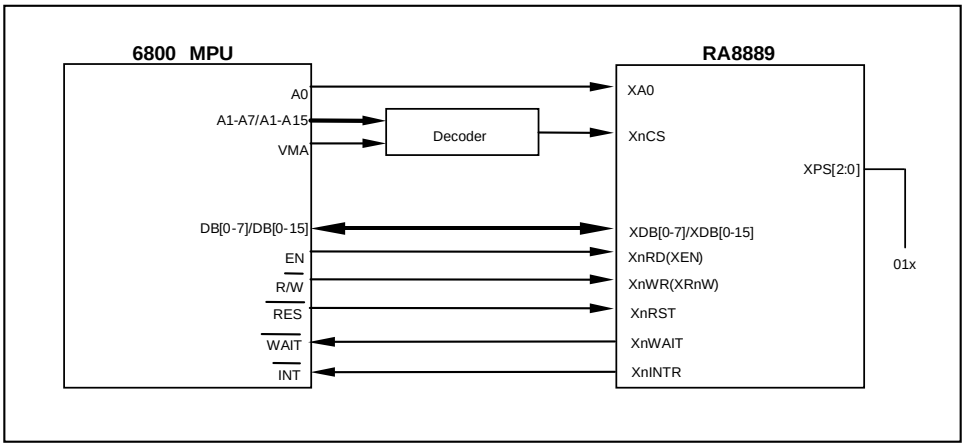


Figure 7-2 : 6800 MPU Interface

7.2.2 並列主控端介面協定

下面的時序圖是標準的 8080 與 6800 介面

8080 – 8/16-bit Interface

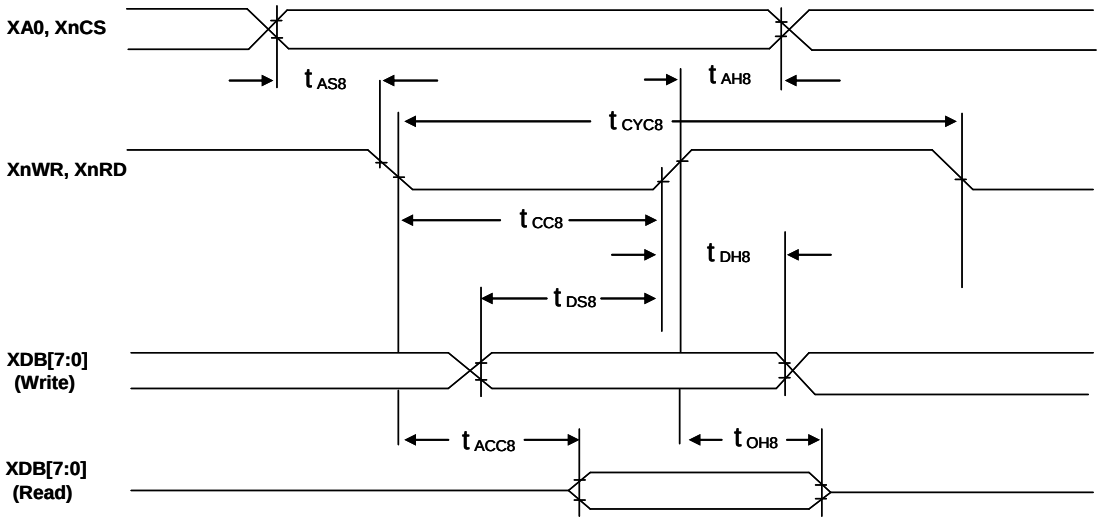


Figure 7-3 : 8080 Waveform

Table 7-2 : 8080 MPU I/F Timing

Symbol	Parameter	Rating		Unit	Symbol
		Min.	Max.		
t_{CYC8}	Cycle time	50	--	ns	tc is one system clock period: tc = 1/SYS_CLK
t_{CC8}	Strobe Pulse width	20	--	ns	
t_{AS8}	Address setup time	0	--	ns	
t_{AH8}	Address hold time	10	--	ns	
t_{DS8}	Data setup time	20	--	ns	
t_{DH8}	Data hold time	10	--	ns	
t_{ACC8}	Data output access time	0	20	ns	
t_{OH8}	Data output hold time	0	20	ns	

6800 – 8/16-bit Interface

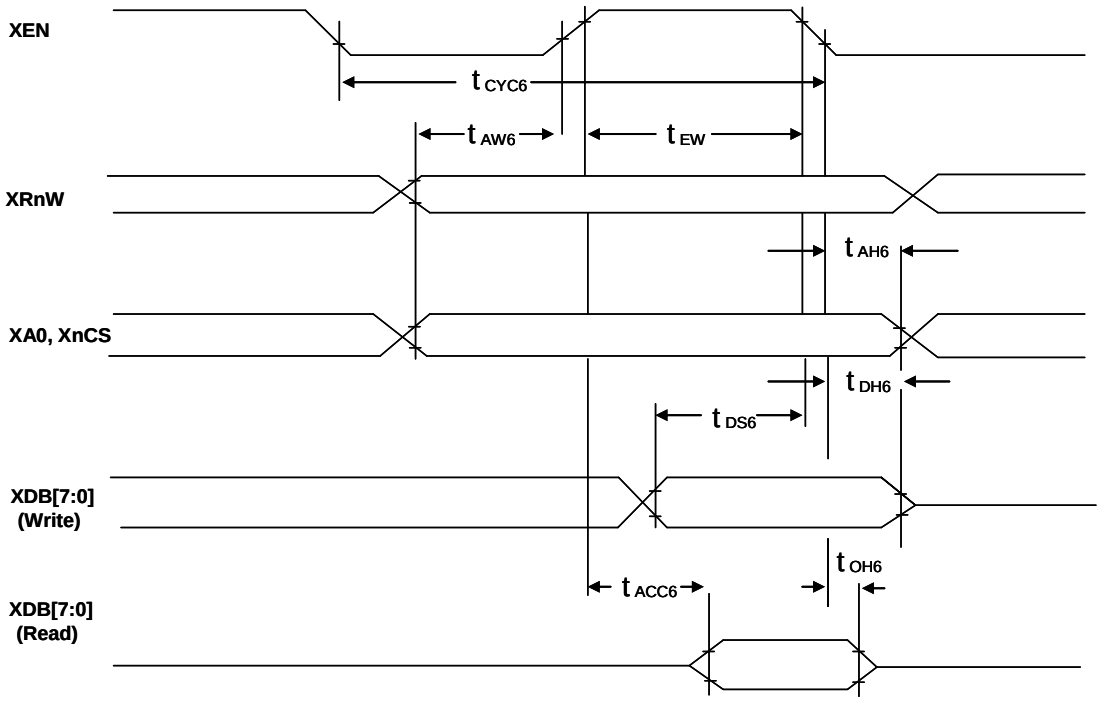


Figure 7-4 : 6800 MPU Waveform

Table 7-3 : 6800 MPU I/F Timing

Symbol	Parameter	Rating		Unit	Symbol
		Min.	Max.		
t_{CYC6}	Cycle time	50	--	ns	tc is one system clock period: tc = 1/SYS_CLK
t_{EW}	Strobe Pulse width	20	--	ns	
t_{AW6}	Address setup time	0	--	ns	
t_{AH6}	Address hold time	10	--	ns	
t_{DS6}	Data setup time	20	--	ns	
t_{DH6}	Data hold time	10	--	ns	
t_{ACC6}	Data output access time	0	20	ns	
t_{OH6}	Data output hold time	0	20	ns	

連續資料的寫入會決定螢幕更新速度，如果在沒有 XnWait 產生等待週期的話，各週期間的時間必須大於 5 個系統時脈週期。如果沒有使用 XnWait 的機制，並且各週期時間超過規範的話，那麼將會有資料漏失與功能錯誤的情形產生。詳細的波型請參考 Figure 7-5 與 Figure 7-6。

建議在 XnCS, XnRD_EN, XnWR_RnW 加上小電容，這樣可以減少 MPU 與 RA8889 傳輸上的干擾。如果使用連接線連接 MPU 與 RA8889，連接線的長度請小於 20cm。另外還建議在 XnCS,XnRD_EN,XnWR_RnW,XA0 上加上 1~10Kohm 提升電阻。

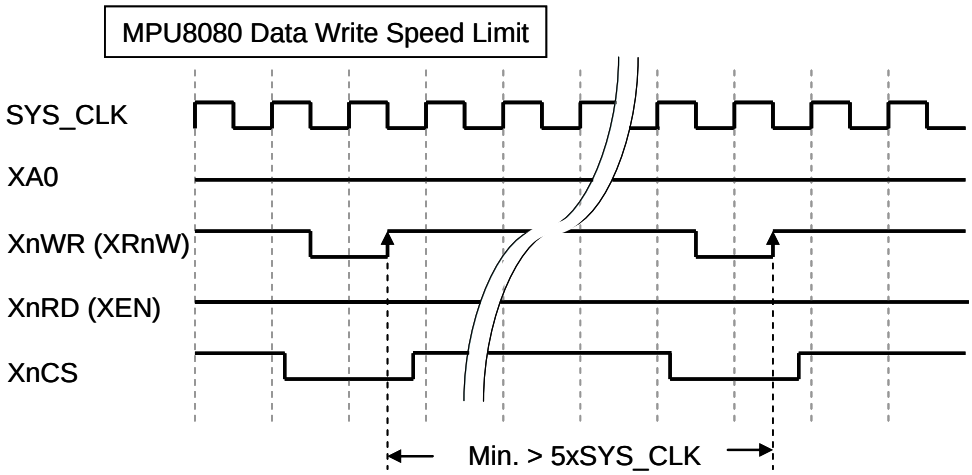


Figure 7-5 : 8080 I/F Continuous Data Write Cycle Waveform

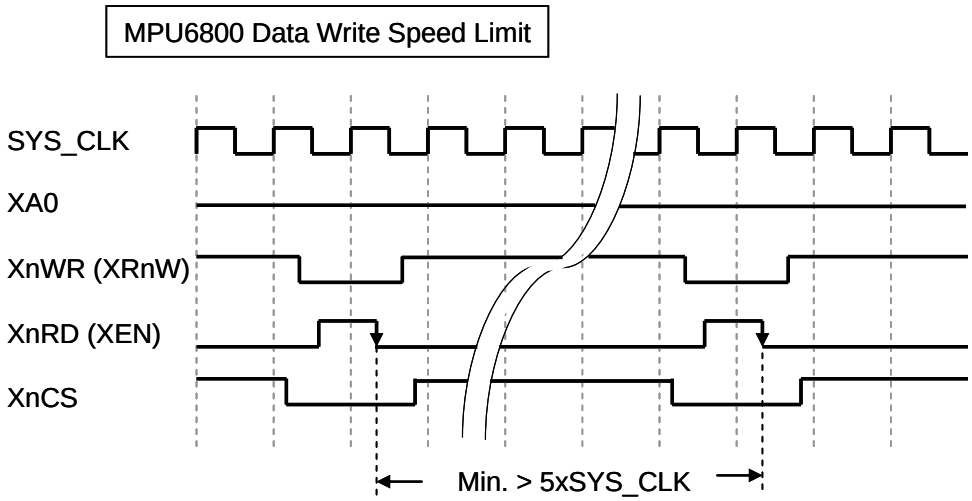


Figure 7-6 : 6800 I/F Continuous Data Write Cycle Waveform

7.3 串列主控端

7.3.1 3-Wire SPI

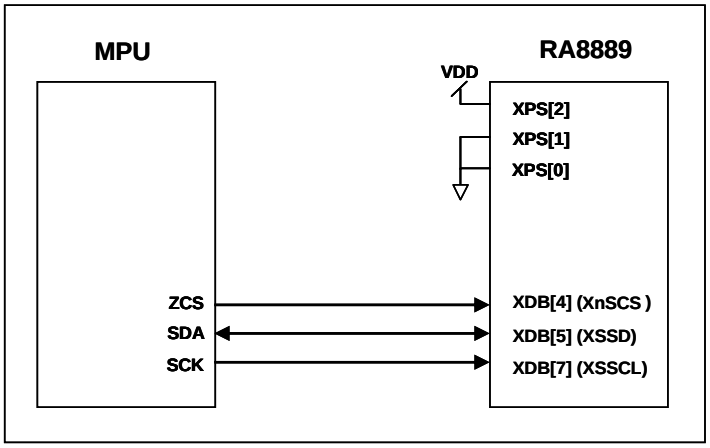


Figure 7-7 : The MPU Interface Diagram of 3-Wire SPI

RA8889提供一個SPI從屬 (Slave) 控制器，SPI 是由晶片選擇線 (XnSCS)、串列傳輸時脈線 (XSSCL) 以及串列資料輸入/輸出線 (XSSD) 所組成的。當 XnSCS 是動作時，XSSCL 是由主要控制器 (Master) 所驅動的，用來閂鎖 XSSD 的信號。使用 SPI 進行通訊時，通過對資料的第一個位元組的 MSB 2 Bits 可以設定目前的週期為指令/資料寫入模式，或是狀態位元/資料讀出的模式。在通訊的過程中，XnSCS 必須要一直保持在低電位狀態，直到通訊結束。

當SPI 在指令/資料寫入模式時 (Figure 7-9、Figure 7-11)，此時傳輸的第2位元組為透過 SPI 的 XSSD 腳位，由主要 (Master) 控制器端提供寫入資料。當SPI 在狀態位元/資料讀取模式時，第2位元組的資料讀取則是由RA8889 的SPI 從屬 (Slave) 控制器根據XSSCL 的動作透過SDA 傳送至主要 (Master) 控制器端。請參考 Figure 7-8、 Figure 7-10 的說明。XSSCL 最大工作頻率為 50Mhz。

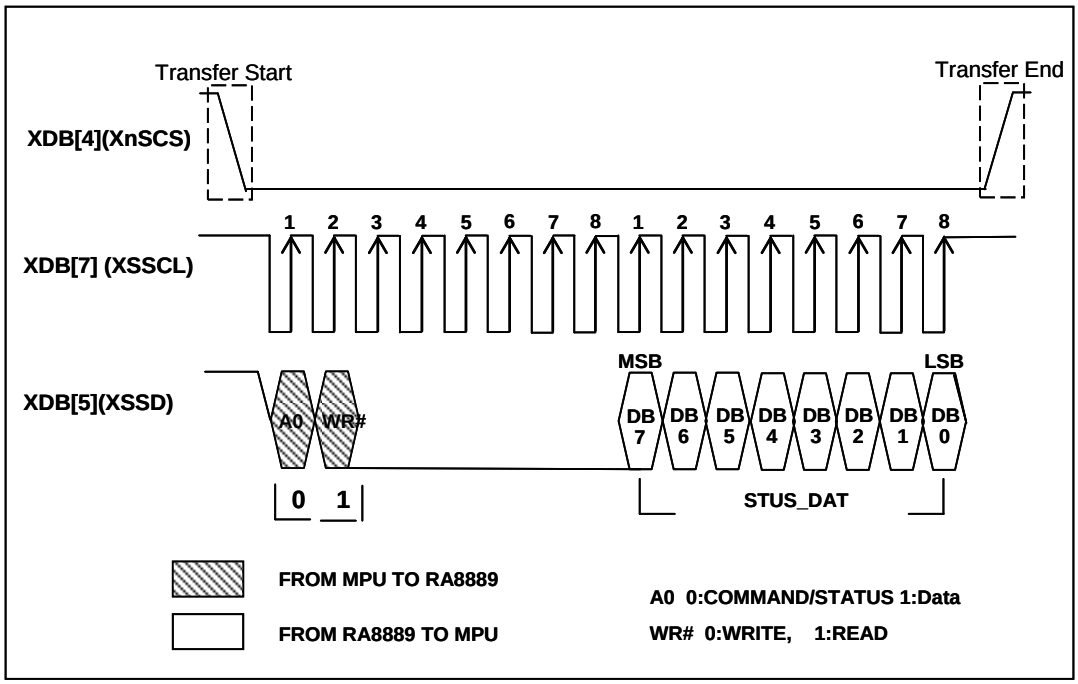


Figure 7-8 : Status Read on 3-Wire SPI-Bus

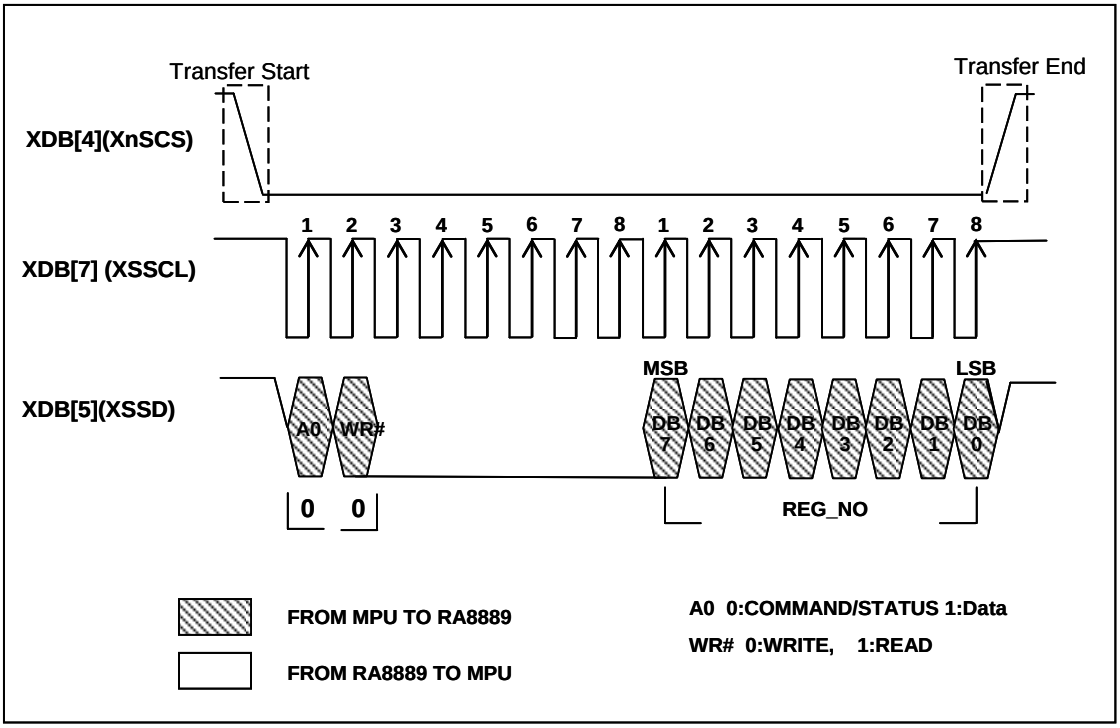


Figure 7-9 : CMD Write on 3-Wire SPI-Bus

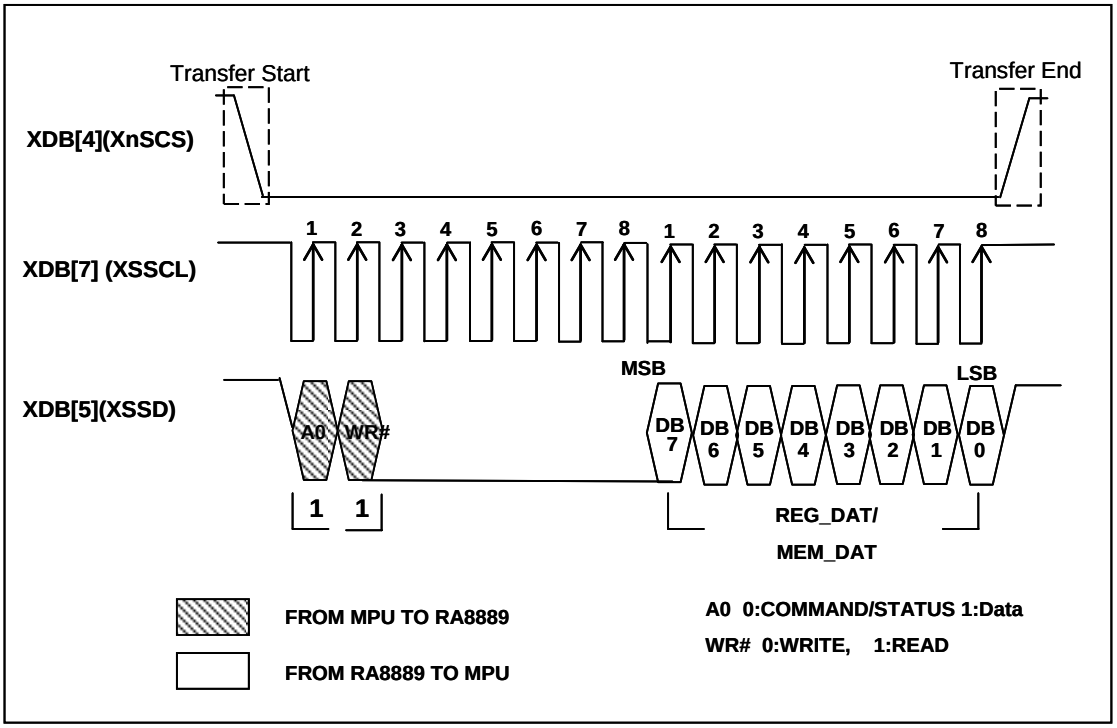


Figure 7-10 : Data Read on 3-Wire SPI-Bus

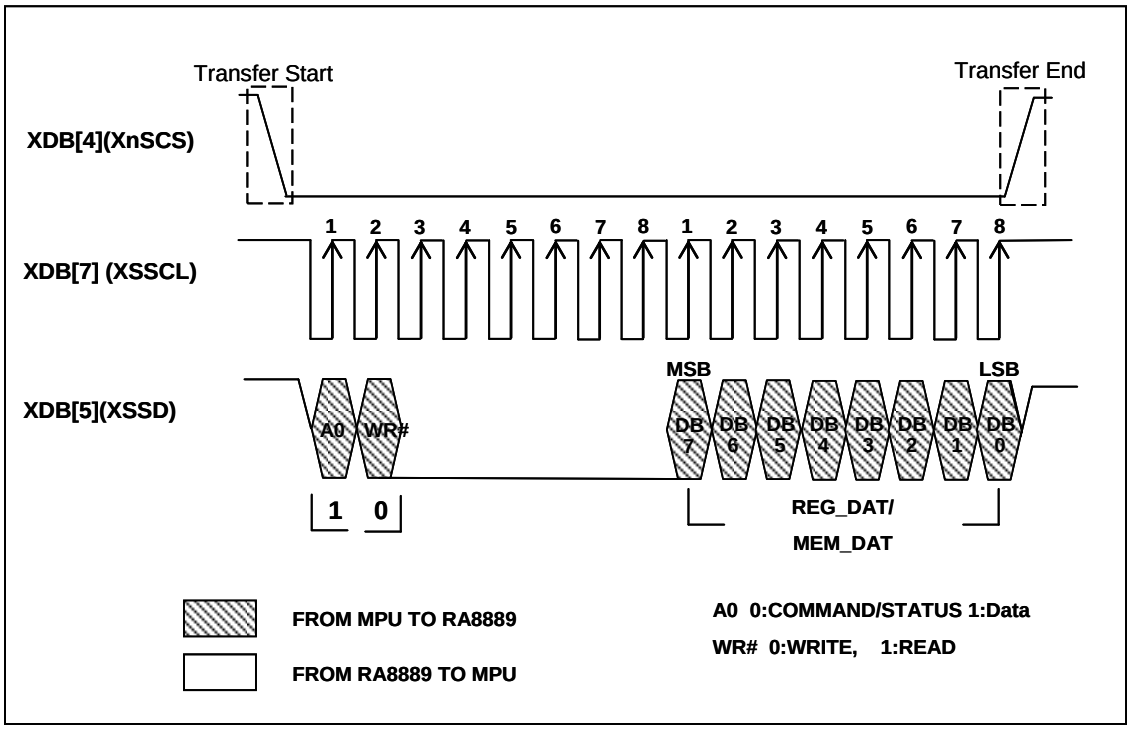


Figure 7-11 : Date Write on 3-Wire SPI-Bus

以下時序圖用於描述 3-Wire SPI 介面的時序規範。

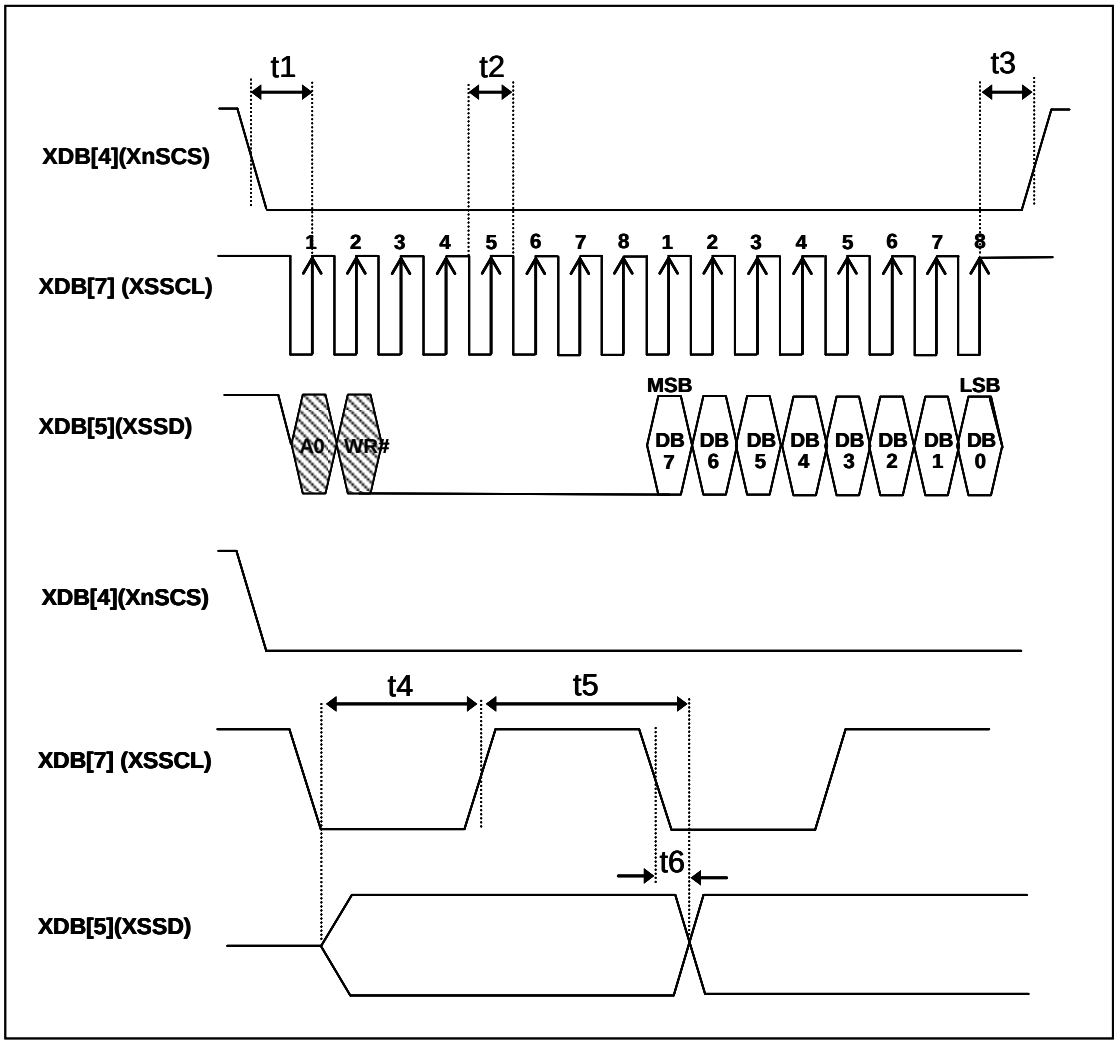


Figure 7-12 : 3-Wire SPI I/F Waveform

Table 7-4 : 3-wire SPI I/F Timing

Symbol	Parameter	Rating		Unit	Symbol
		Min.	Max.		
t_2	Cycle time	20	10000	ns	
t_1	CS setup time to rising edge of SCL	1/2 t_2	--	ns	
t_3	CS hold time from rising edge of SCL	1/2 t_2	--	ns	
t_4	Data setup time to rising edge of SCL	5	--	ns	
t_5	Data hold time from rising edge of SCL	5	--	ns	
t_6	Data output valid from falling edge of SCL	5	20	ns	

7.3.2 4-Wire SPI

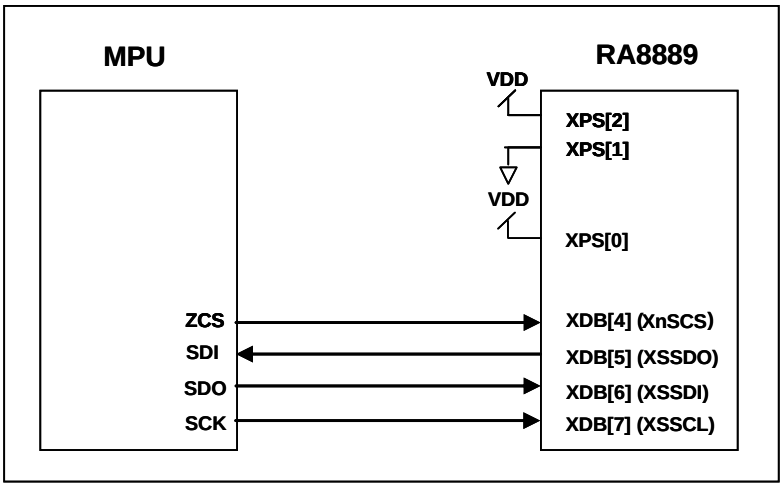


Figure 7-13 : The MPU Interface Diagram of 4-Wire SPI

4-wire SPI 介面與 3-wire SPI 介面類似，唯一不同的是資料信號。在 3-wire SPI 介面中，雙向的 XSSD 信號用來當作資料信號且從屬(Slave) / 主要(Master) 皆可驅動。在 4-wire SPI 介面中，XSSD 信號功能被區分為 XSSDI 與 XSDO 信號。SDI 是由 SPI master 驅動的資料腳位；SDO 則是來自 SPI 從屬 (Slave) 端的資料輸出。關於詳細的資料協定，請參考 Figure 7-14 ~Figure 7-17。

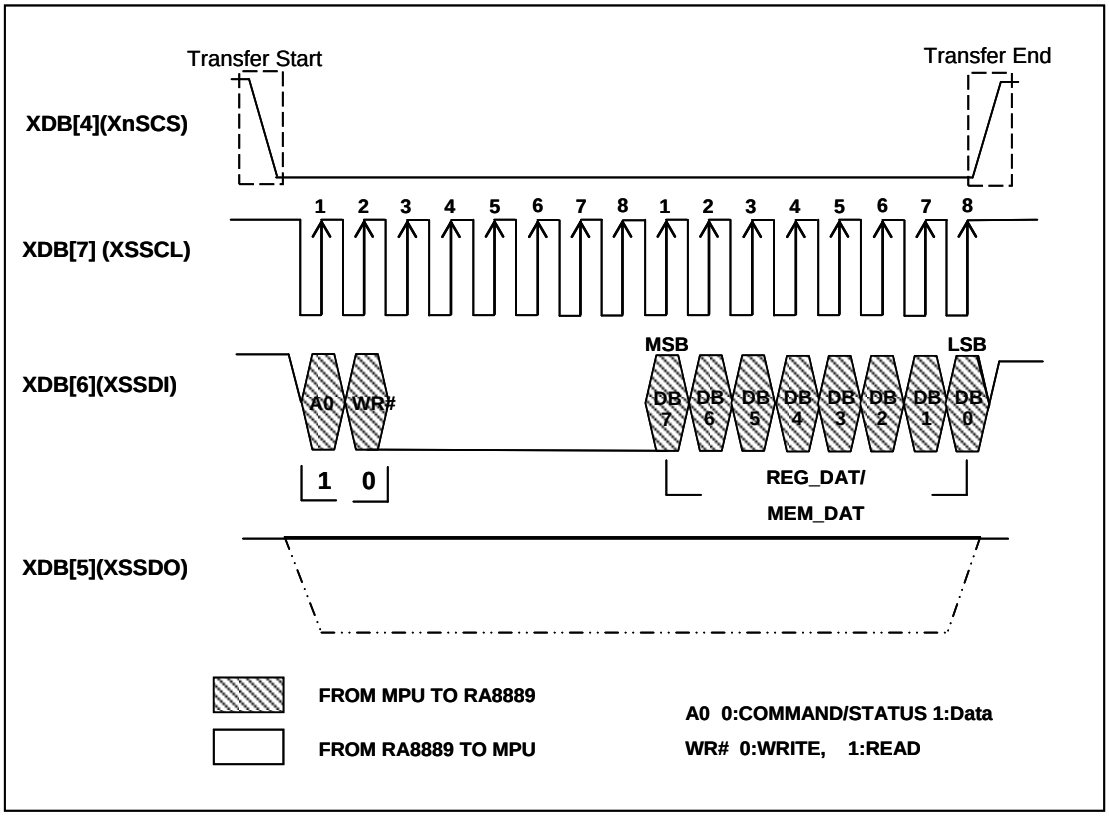


Figure 7-14 : Date Write on 4-Wire SPI-Bus

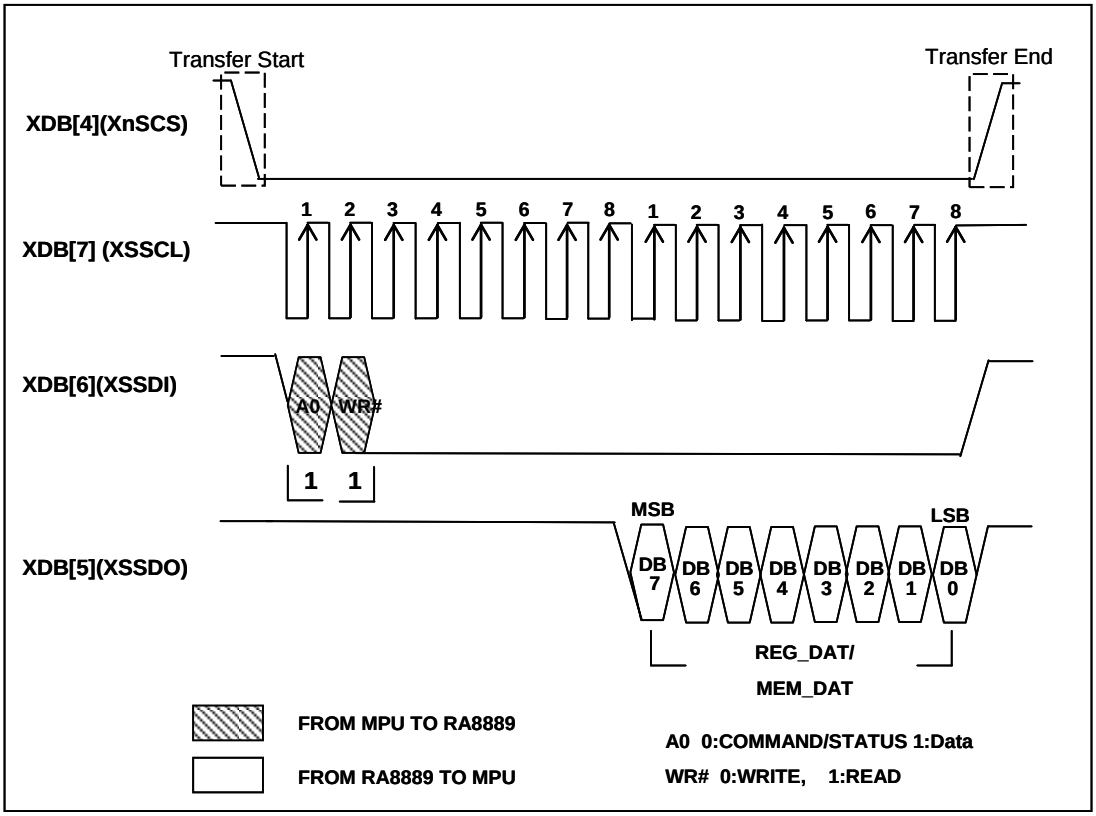


Figure 7-15 : Data Read on 4-Wire SPI-Bus

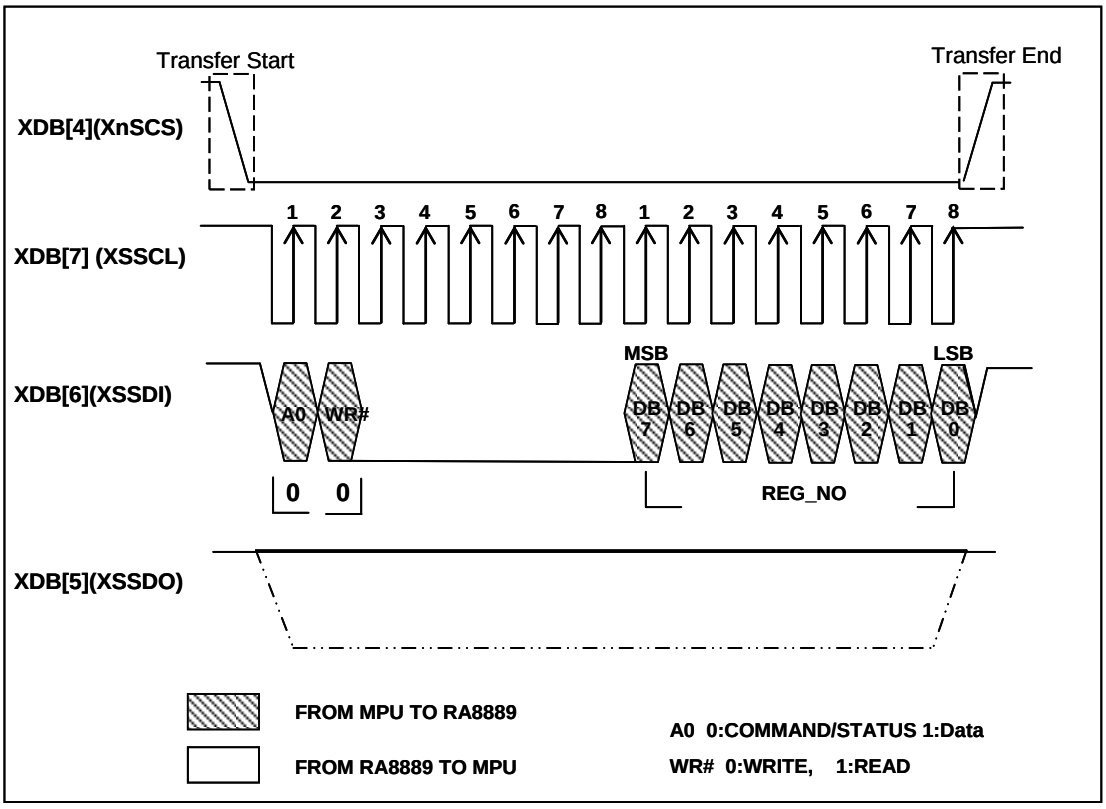


Figure 7-16 : CMD Write on 4-Wire SPI-Bus

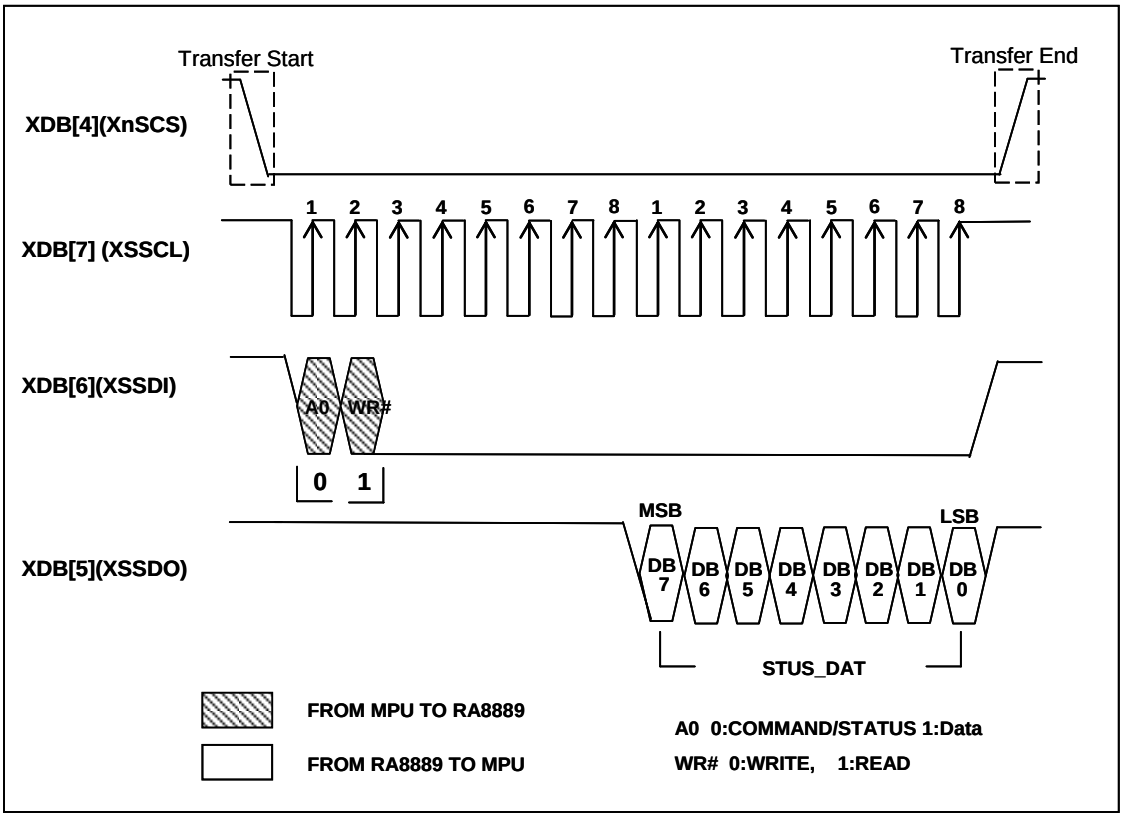


Figure 7-17 : Status Read on 4-Wire SPI-Bus

以下時序圖用於描述 4-Wire SPI 介面的時序規範。

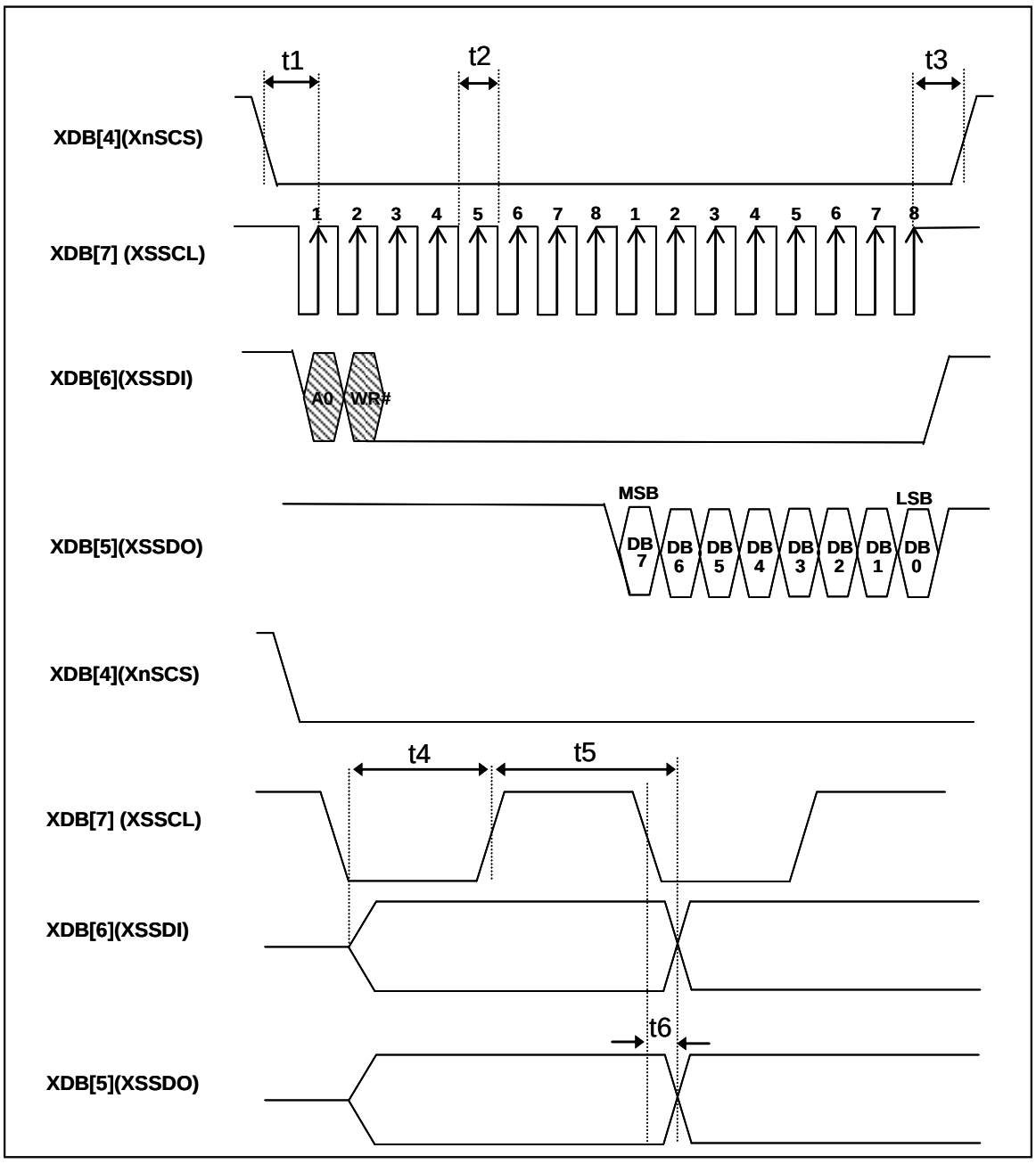
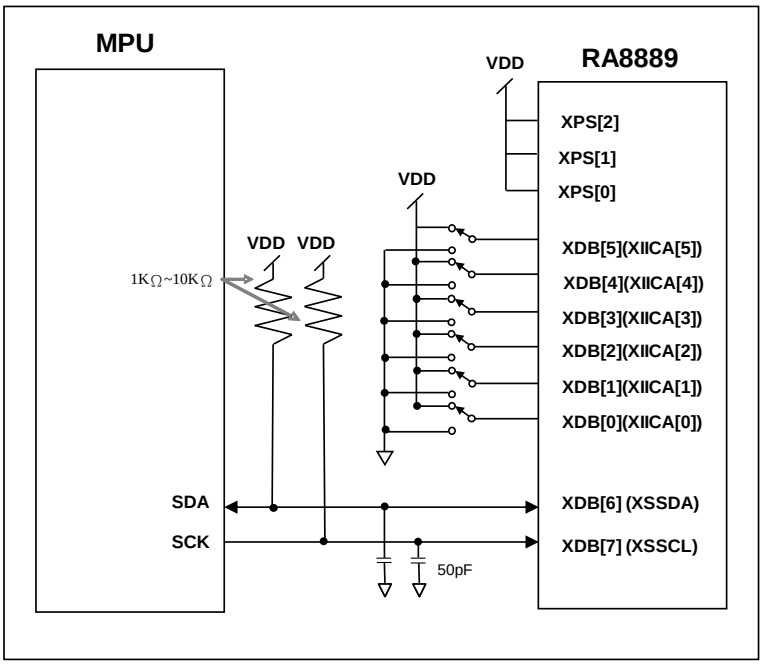


Figure 7-18 : 4-Wire SPI I/F Waveform

Table 7-5 : 4-wire SPI I/F Timing

Symbol	Parameter	Rating		Unit	Symbol
		Min.	Max.		
t_2	Cycle time	20	10000	ns	
t_1	CS setup time to rising edge of SCL	$1/2t_2$	--	ns	
t_3	CS hold time from rising edge of SCL	$1/2t_2$	--	ns	
t_4	Data setup time to rising edge of SCL	5	--	ns	
t_5	Data hold time from rising edge of SCL	5	--	ns	
t_6	Data output valid from falling edge of SCL	5	20	ns	

7.3.3 IIC I/F



IICA[5:0]					
BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
XIICA[5]	XIICA[4]	XIICA[3]	XIICA[2]	XIICA[1]	XIICA[0]

Figure 7-19 : The MPU Interface Diagram of IIC

IIC 介面由 XSSCL 與 XSSDA 兩條資料匯流排線所組成，相容於標準的 IIC 介面。IIC 傳輸的前 7 個位元，是指 IIC 的 Spec 中定義的從屬 (Slave) 端位址。前 6 個位元代表 RA8889 的 IIC device ID。接下來1 個位元是 A0，代表周期類型。當 A0= 1，代表接下來的週期為資料週期；當 A0 = 0，為命令/狀態週期。若 IIC 匯流排上的週期的 MSB 6 位元 (共有7bit) 與RA8889 的device ID 相同，RA8889 的 IIC 從屬 (Slave) 就會動作。

RA8889 的配置位置 (Device ID) 是可程式化的，設定上可以由 XIICA[5:0]/XDB[5:0] 來完成。 RA8889 有4 種週期類型，分別為:「指令寫入」、「狀態讀取」、「資料寫入」與「資料讀取」週期。週期型態是由 A0 及 WR 位元所設定。詳細協定說明，請參考Figure 7-20 ~ Figure 7-23。

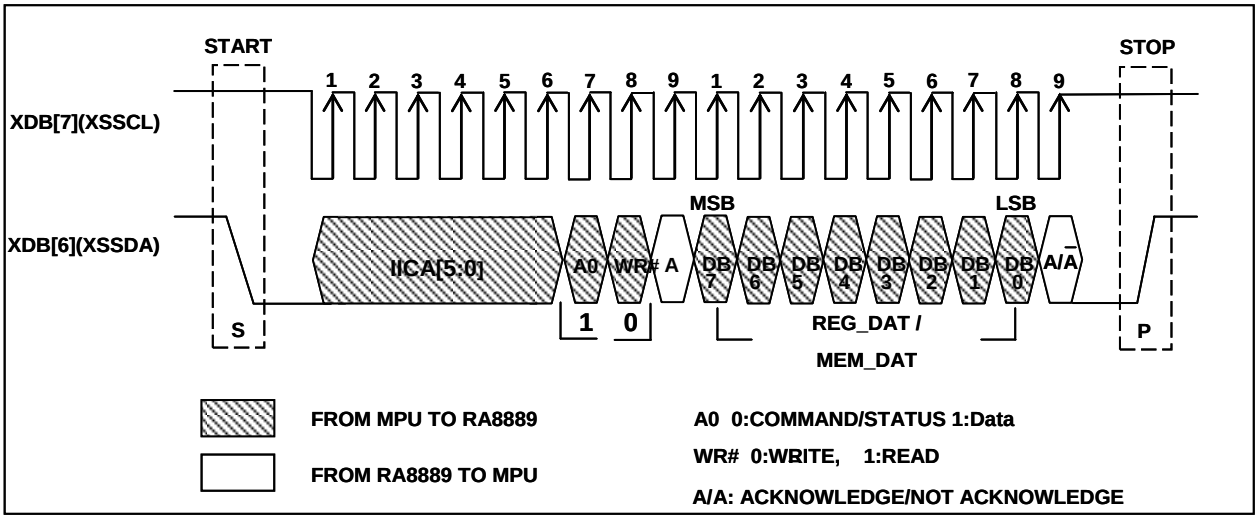


Figure 7-20 : Data Write on IIC-Bus

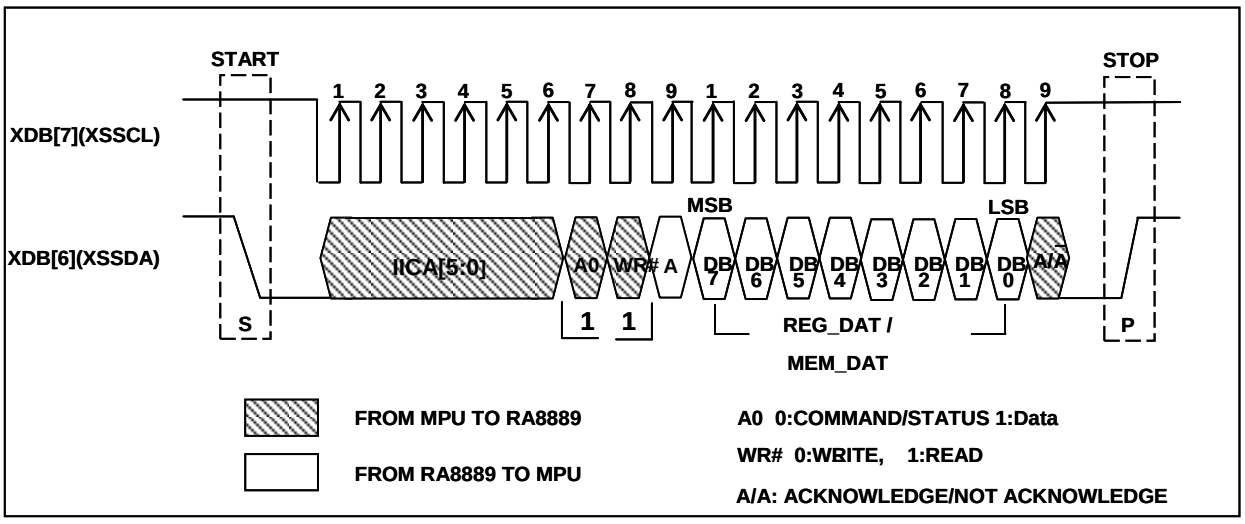


Figure 7-21 : Data Read on IIC-Bus

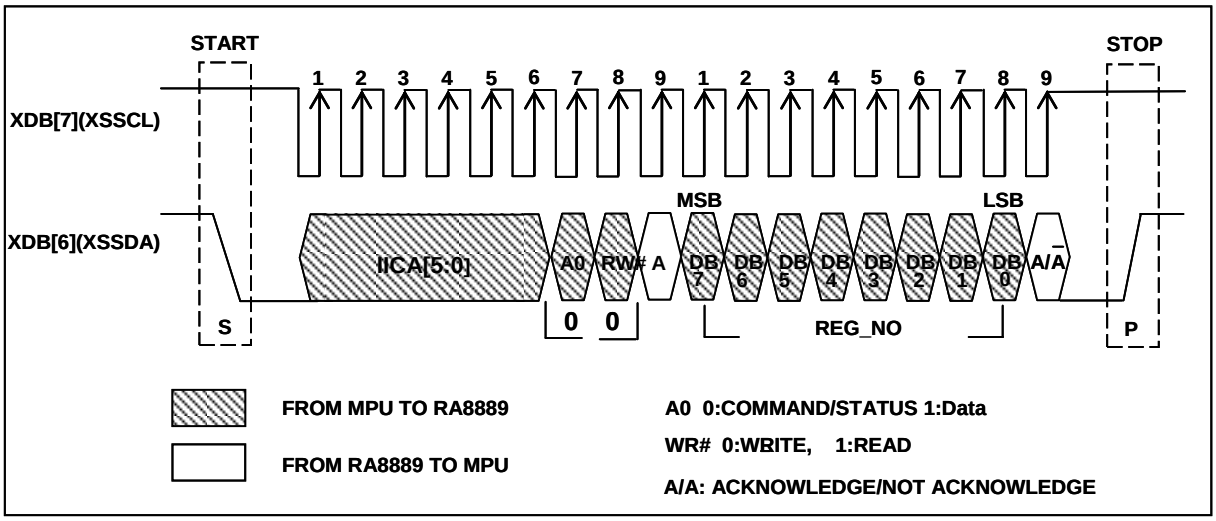


Figure 7-22 : CMD Write on IIC-Bus

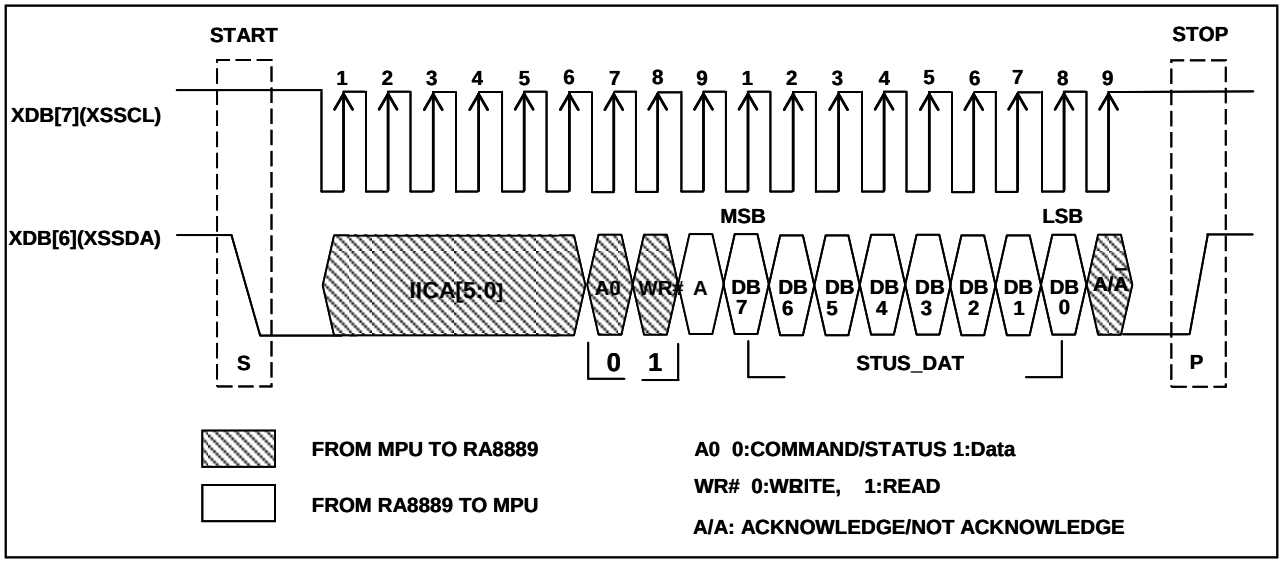


Figure 7-23 : Status Read on IIC-Bus

以下時序圖用於描述 IIC 介面的時序規範。

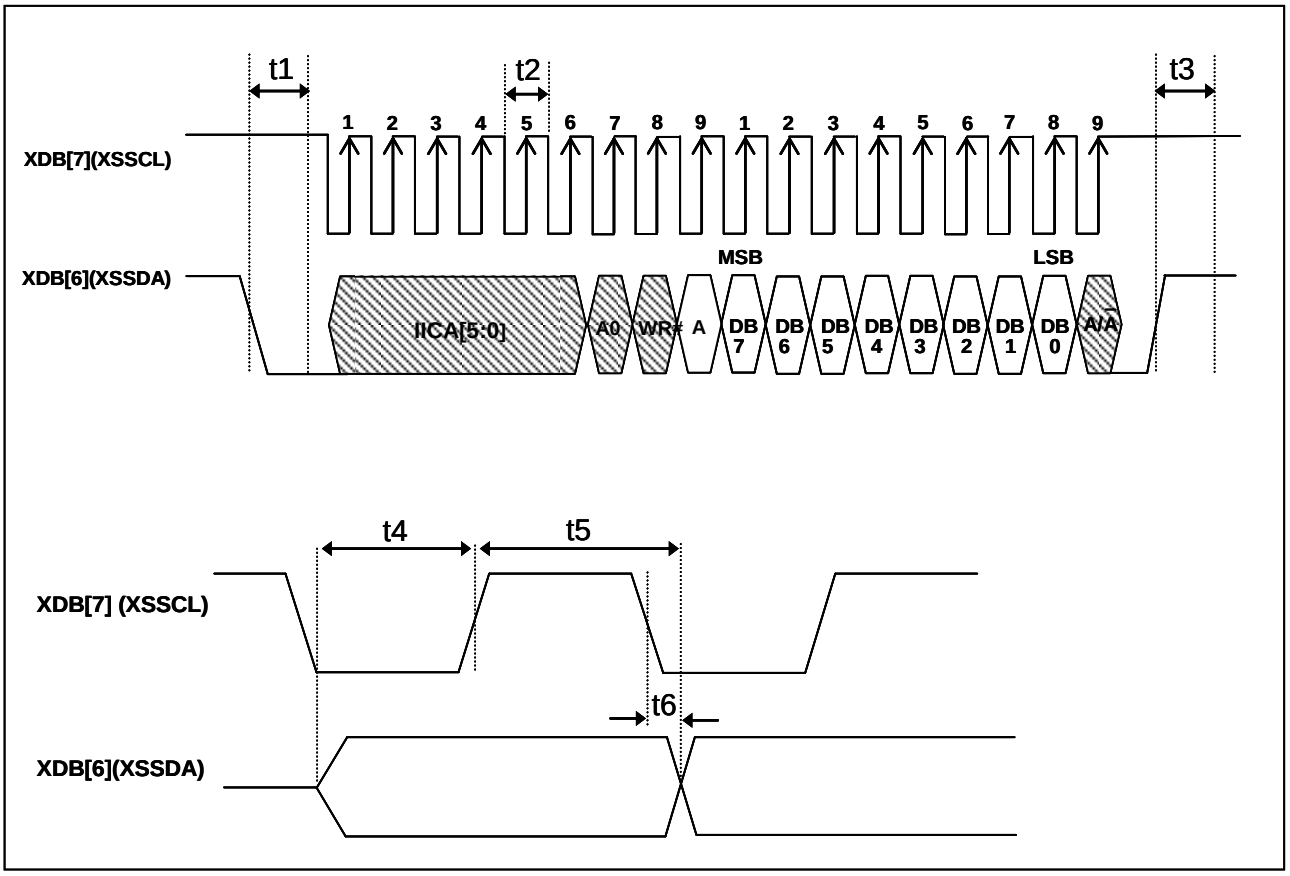


Figure 7-24 : IIC I/F Waveform

Table 7-6 : IIC I/F Timing

Symbol	Parameter	Rating		Unit	Symbol
		Min.	Max.		
t_2	Cycle time	10000	2500	ns	
t_1	Start Strobe Pulse width	180	--	ns	
t_3	Stop Strobe Pulse width	180	--	ns	
t_4	Data setup time to rising edge of SCL	5	--	ns	
t_5	Data hold time from rising edge of SCL	5	--	ns	
t_6	Data output valid from falling edge of SCL	5	20	ns	

7.4 顯示資料輸入格式

7.4.1 不包含混合位元 (Opacity) 的輸入資料 (RGB)

8-bit MPU, 1bpp mode (單色資料)

Order	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0
2	P_{15}	P_{14}	P_{13}	P_{12}	P_{11}	P_{10}	P_9	P_8
3	P_{23}	P_{22}	P_{21}	P_{20}	P_{19}	P_{18}	P_{17}	P_{16}
4	P_{31}	P_{30}	P_{29}	P_{28}	P_{27}	P_{26}	P_{25}	P_{24}
5	P_{39}	P_{38}	P_{37}	P_{36}	P_{35}	P_{34}	P_{33}	P_{32}
6	P_{47}	P_{46}	P_{45}	P_{44}	P_{43}	P_{42}	P_{41}	P_{40}

*** *註*: 這個只提供 BTE 的色彩擴展功能使用。使用此功能時底圖 (Canvas) 必須設定成 8bpp 色深，並且只能從 MPU 接收 8bits 資料。在寫入單色資料完成後，再使用 BTE 色彩擴展功能擴展成想要的顯示圖像。

8-bit MPU, 8bpp mode (RGB 3:3:2)

Order	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	R_0^7	R_0^6	R_0^5	G_0^7	G_0^6	G_0^5	B_0^7	B_0^6
2	R_1^7	R_1^6	R_1^5	G_1^7	G_1^6	G_1^5	B_1^7	B_1^6
3	R_2^7	R_2^6	R_2^5	G_2^7	G_2^6	G_2^5	B_2^7	B_2^6
4	R_3^7	R_3^6	R_3^5	G_3^7	G_3^6	G_3^5	B_3^7	B_3^6
5	R_4^7	R_4^6	R_4^5	G_4^7	G_4^6	G_4^5	B_4^7	B_4^6
6	R_5^7	R_5^6	R_5^5	G_5^7	G_5^6	G_5^5	B_5^7	B_5^6

8-bit MPU, 16bpp mode (RGB 5:6:5)

Order	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	G_0^4	G_0^3	G_0^2	B_0^7	B_0^6	B_0^5	B_0^4	B_0^3
2	R_0^7	R_0^6	R_0^5	R_0^4	R_0^3	G_0^7	G_0^6	G_0^5
3	G_1^4	G_1^3	G_1^2	B_1^7	B_1^6	B_1^5	B_1^4	B_1^3
4	R_1^7	R_1^6	R_1^5	R_1^4	R_1^3	G_1^7	G_1^6	G_1^5
5	G_2^4	G_2^3	G_2^2	B_2^7	B_2^6	B_2^5	B_2^4	B_2^3
6	R_2^7	R_2^6	R_2^5	R_2^4	R_2^3	G_2^7	G_2^6	G_2^5

8-bit MPU, 24bpp mode (RGB 8:8:8)

Order	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	B_0^7	B_0^6	B_0^5	B_0^4	B_0^3	B_0^2	B_0^1	B_0^0
2	G_0^7	G_0^6	G_0^5	G_0^4	G_0^3	G_0^2	G_0^1	G_0^0
3	R_0^7	R_0^6	R_0^5	R_0^4	R_0^3	R_0^2	R_0^1	R_0^0
4	B_1^7	B_1^6	B_1^5	B_1^4	B_1^3	B_1^2	B_1^1	B_1^0
5	G_1^7	G_1^6	G_1^5	G_1^4	G_1^3	G_1^2	G_1^1	G_1^0
6	R_1^7	R_1^6	R_1^5	R_1^4	R_1^3	R_1^2	R_1^1	R_1^0

16-bit MPU, 1bpp mode 1 (單色資料)

Order	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0
2	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	P_{15}	P_{14}	P_{13}	P_{12}	P_{11}	P_{10}	P_9	P_8
3	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	P_{23}	P_{22}	P_{21}	P_{20}	P_{19}	P_{18}	P_{17}	P_{16}
4	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	P_{31}	P_{30}	P_{29}	P_{28}	P_{27}	P_{26}	P_{25}	P_{24}
5	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	P_{39}	P_{38}	P_{37}	P_{36}	P_{35}	P_{34}	P_{33}	P_{32}
6	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	P_{47}	P_{46}	P_{45}	P_{44}	P_{43}	P_{42}	P_{41}	P_{40}

*** *註*: 這個功能只提供給 BTE 的色彩擴展功能使用，使用時必須設定底圖為 8bpp 色深，而在這個模式下只能接受 8bits 資料。底圖的工作視窗其寫入單色寬度必須設定是實際單色像素資料除以 8，以此寫入記憶體，在單色寫入記憶體後，致能色彩擴展功能並且設定想要的顯示色深。

16-bit MPU, 1bpp mode 2 (單色資料)

Order	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	P_{15}	P_{14}	P_{13}	P_{12}	P_{11}	P_{10}	P_9	P_8	P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0
2	P_{31}	P_{30}	P_{29}	P_{28}	P_{27}	P_{26}	P_{25}	P_{24}	P_{23}	P_{22}	P_{21}	P_{20}	P_{19}	P_{18}	P_{17}	P_{16}
3	P_{47}	P_{46}	P_{45}	P_{44}	P_{43}	P_{42}	P_{41}	P_{40}	P_{39}	P_{38}	P_{37}	P_{36}	P_{35}	P_{34}	P_{33}	P_{32}
4	P_{63}	P_{62}	P_{61}	P_{60}	P_{59}	P_{58}	P_{57}	P_{56}	P_{55}	P_{54}	P_{53}	P_{52}	P_{51}	P_{50}	P_{49}	P_{48}
5	P_{79}	P_{78}	P_{77}	P_{76}	P_{75}	P_{74}	P_{73}	P_{72}	P_{71}	P_{70}	P_{69}	P_{68}	P_{67}	P_{66}	P_{65}	P_{64}
6	P_{95}	P_{94}	P_{93}	P_{92}	P_{91}	P_{90}	P_{89}	P_{88}	P_{87}	P_{86}	P_{85}	P_{84}	P_{83}	P_{82}	P_{81}	P_{80}

*** *註*: 這個功能只提供給 BTE 的色彩擴展功能使用，使用上與 16bpp 類似。但是除了設定底圖色深為 16bit 外，其設定的底圖與工作視窗其寬度必須為單色寬度除以 16，以此設定將圖像資料寫入記憶體。在單色寫入記憶體後，致能色彩擴展功能並且設定想要的主顯示畫面或畫中畫色深。

16-bit MPU, 8bpp mode 1 (RGB 3:3:2)

Order	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	R_0^7	R_0^6	R_0^5	G_0^7	G_0^6	G_0^5	B_0^7	B_0^6
2	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	R_1^7	R_1^6	R_1^5	G_1^7	G_1^6	G_1^5	B_1^7	B_1^6
3	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	R_2^7	R_2^6	R_2^5	G_2^7	G_2^6	G_2^5	B_2^7	B_2^6
4	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	R_3^7	R_3^6	R_3^5	G_3^7	G_3^6	G_3^5	B_3^7	B_3^6
5	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	R_4^7	R_4^6	R_4^5	G_4^7	G_4^6	G_4^5	B_4^7	B_4^6
6	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	R_5^7	R_5^6	R_5^5	G_5^7	G_5^6	G_5^5	B_5^7	B_5^6

16-bit MPU, 8bpp mode 2 (RGB 3:3:2)

Order	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	R_1^7	R_1^6	R_1^5	G_1^7	G_1^6	G_1^5	B_1^7	B_1^6	R_0^7	R_0^6	R_0^5	G_0^7	G_0^6	G_0^5	B_0^7	B_0^6
2	R_3^7	R_3^6	R_3^5	G_3^7	G_3^6	G_3^5	B_3^7	B_3^6	R_2^7	R_2^6	R_2^5	G_2^7	G_2^6	G_2^5	B_2^7	B_2^6
3	R_5^7	R_5^6	R_5^5	G_5^7	G_5^6	G_5^5	B_5^7	B_5^6	R_4^7	R_4^6	R_4^5	G_4^7	G_4^6	G_4^5	B_4^7	B_4^6
4	R_7^7	R_7^6	R_7^5	G_7^7	G_7^6	G_7^5	B_7^7	B_7^6	R_6^7	R_6^6	R_6^5	G_6^7	G_6^6	G_6^5	B_6^7	B_6^6
5	R_9^7	R_9^6	R_9^5	G_9^7	G_9^6	G_9^5	B_9^7	B_9^6	R_8^7	R_8^6	R_8^5	G_8^7	G_8^6	G_8^5	B_8^7	B_8^6
6	R_{11}^7	R_{11}^6	R_{11}^5	G_{11}^7	G_{11}^6	G_{11}^5	B_{11}^7	B_{11}^6	R_{10}^7	R_{10}^6	R_{10}^5	G_{10}^7	G_{10}^6	G_{10}^5	B_{10}^7	B_{10}^6

*** *註*: 使用上與 16bpp 圖像資料類似，除了設定底圖圖像為 16bpp 色深外，底圖寬度與工作窗寬度為圖像資料除以 2，以此設定為基礎寫入記憶體。主圖像與畫中畫圖像色深需要設定為 8bpp。

16-bit MPU, 16bpp mode (RGB 5:6:5)

Order	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	R_0^7	R_0^6	R_0^5	R_0^4	R_0^3	G_0^7	G_0^6	G_0^5	G_0^4	G_0^3	G_0^2	B_0^7	B_0^6	B_0^5	B_0^4	B_0^3
2	R_1^7	R_1^6	R_1^5	R_1^4	R_1^3	G_1^7	G_1^6	G_1^5	G_1^4	G_1^3	G_1^2	B_1^7	B_1^6	B_1^5	B_1^4	B_1^3
3	R_2^7	R_2^6	R_2^5	R_2^4	R_2^3	G_2^7	G_2^6	G_2^5	G_2^4	G_2^3	G_2^2	B_2^7	B_2^6	B_2^5	B_2^4	B_2^3
4	R_3^7	R_3^6	R_3^5	R_3^4	R_3^3	G_3^7	G_3^6	G_3^5	G_3^4	G_3^3	G_3^2	B_3^7	B_3^6	B_3^5	B_3^4	B_3^3
5	R_4^7	R_4^6	R_4^5	R_4^4	R_4^3	G_4^7	G_4^6	G_4^5	G_4^4	G_4^3	G_4^2	B_4^7	B_4^6	B_4^5	B_4^4	B_4^3
6	R_5^7	R_5^6	R_5^5	R_5^4	R_5^3	G_5^7	G_5^6	G_5^5	G_5^4	G_5^3	G_5^2	B_5^7	B_5^6	B_5^5	B_5^4	B_5^3

16-bit MPU, 24bpp mode 1 (RGB 8:8:8)

Order	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	G_0^7	G_0^6	G_0^5	G_0^4	G_0^3	G_0^2	G_0^1	G_0^0	B_0^7	B_0^6	B_0^5	B_0^4	B_0^3	B_0^2	B_0^1	B_0^0
2	B_1^7	B_1^6	B_1^5	B_1^4	B_1^3	B_1^2	B_1^1	B_1^0	R_0^7	R_0^6	R_0^5	R_0^4	R_0^3	R_0^2	R_0^1	R_0^0
3	R_1^7	R_1^6	R_1^5	R_1^4	R_1^3	R_1^2	R_1^1	R_1^0	G_1^7	G_1^6	G_1^5	G_1^4	G_1^3	G_1^2	G_1^1	G_1^0
4	G_2^7	G_2^6	G_2^5	G_2^4	G_2^3	G_2^2	G_2^1	G_2^0	B_2^7	B_2^6	B_2^5	B_2^4	B_2^3	B_2^2	B_2^1	B_2^0
5	B_3^7	B_3^6	B_3^5	B_3^4	B_3^3	B_3^2	B_3^1	B_3^0	R_2^7	R_2^6	R_2^5	R_2^4	R_2^3	R_2^2	R_2^1	R_2^0
6	R_3^7	R_3^6	R_3^5	R_3^4	R_3^3	R_3^2	R_3^1	R_3^0	G_3^7	G_3^6	G_3^5	G_3^4	G_3^3	G_3^2	G_3^1	G_3^0

16-bit MPU, 24bpp mode 2 (RGB 8:8:8)

Order	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	G_0^7	G_0^6	G_0^5	G_0^4	G_0^3	G_0^2	G_0^1	G_0^0	B_0^7	B_0^6	B_0^5	B_0^4	B_0^3	B_0^2	B_0^1	B_0^0
2	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	R_0^7	R_0^6	R_0^5	R_0^4	R_0^3	R_0^2	R_0^1	R_0^0
3	G_1^7	G_1^6	G_1^5	G_1^4	G_1^3	G_1^2	G_1^1	G_1^0	B_1^7	B_1^6	B_1^5	B_1^4	B_1^3	B_1^2	B_1^1	B_1^0
4	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	R_1^7	R_1^6	R_1^5	R_1^4	R_1^3	R_1^2	R_1^1	R_1^0
5	G_2^7	G_2^6	G_2^5	G_2^4	G_2^3	G_2^2	G_2^1	G_2^0	B_2^7	B_2^6	B_2^5	B_2^4	B_2^3	B_2^2	B_2^1	B_2^0
6	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	R_2^7	R_2^6	R_2^5	R_2^4	R_2^3	R_2^2	R_2^1	R_2^0

7.4.2 包含混位元 (Opacity)的輸入資料 (αRGB)

8-bit MPU, 8bpp mode (αIndex 2:6)

RA8889 為了提供 OSD 應用的功能，因此內建從 4096 色中可選擇的 64 色調色盤。使用者可以內建調色盤為希望顯示的顏色，並且使用索引的方式使用。 α 值表示的是對比值。

Order	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	α_1^3	α_1^2	Index color of pixel 0					
2	α_3^3	α_3^2	Index color of pixel 1					
3	α_5^3	α_5^2	Index color of pixel 2					
4	α_7^3	α_7^2	Index color of pixel 3					
5	α_9^3	α_9^2	Index color of pixel 4					
6	α_{11}^3	α_{11}^2	Index color of pixel 5					

$\alpha_x^3\alpha_x^2$: 0 – 100%, 1 – 20/32, 2 – 11/32, 3 – 0

8-bit MPU, 16bpp mode (αRGB 4:4:4:4)

Order	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	G_0^7	G_0^6	G_0^5	G_0^4	B_0^7	B_0^6	B_0^5	B_0^4
2	α_0^3	α_0^2	α_0^1	α_0^0	R_0^7	R_0^6	R_0^5	R_0^4
3	G_1^7	G_1^6	G_1^5	G_1^4	B_1^7	B_1^6	B_1^5	B_1^4
4	α_1^3	α_1^2	α_1^1	α_1^0	R_1^7	R_1^6	R_1^5	R_1^4
5	G_2^7	G_2^6	G_2^5	G_2^4	B_2^7	B_2^6	B_2^5	B_2^4
6	α_2^3	α_2^2	α_2^1	α_2^0	R_2^7	R_2^6	R_2^5	R_2^4

$\alpha_x^3\alpha_x^2\alpha_x^1\alpha_x^0$: 0 –100%, 1 –30/32, 2 –28/32, 3 –26/32, 4 – 24/32, … …, 12 – 8/32, 13 – 6/32, 14 – 4/32, 15 – 0.

8-bit MPU, 32bpp mode (αRGB 8:8:8:8)

Order	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	B_0^7	B_0^6	B_0^5	B_0^4	B_0^3	B_0^2	B_0^1	B_0^0
2	G_0^7	G_0^6	G_0^5	G_0^4	G_0^3	G_0^2	G_0^1	G_0^0
3	R_0^7	R_0^6	R_0^5	R_0^4	R_0^3	R_0^2	R_0^1	R_0^0
4	α_0^7	α_0^6	α_0^5	α_0^4	α_0^3	α_0^2	α_0^1	α_0^0
5	B_1^7	B_1^6	B_1^5	B_1^4	B_1^3	B_1^2	B_1^1	B_1^0
6	G_1^7	G_1^6	G_1^5	G_1^4	G_1^3	G_1^2	G_1^1	G_1^0
7	R_1^7	R_1^6	R_1^5	R_1^4	R_1^3	R_1^2	R_1^1	R_1^0
8	α_1^7	α_1^6	α_1^5	α_1^4	α_1^3	α_1^2	α_1^1	α_1^0

16-bit MPU, 8bpp mode (αIndex 2:6)

Order	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	α_0^3	α_0^2	Index color of pixel 0					
2	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	α_1^3	α_1^2	Index color of pixel 1					
3	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	α_2^3	α_2^2	Index color of pixel 2					
4	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	α_3^3	α_3^2	Index color of pixel 3					
5	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	α_4^3	α_4^2	Index color of pixel 4					
6	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	α_5^3	α_5^2	Index color of pixel 5					

$\alpha_x^3\alpha_x^2$: 0 – 0, 1 – 11/32, 2 – 20/32, 3 – 100%

16-bit MPU, 16bpp mode (αRGB 4:4:4:4)

Order	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	α_0^3	α_0^2	α_0^1	α_0^0	R_0^7	R_0^6	R_0^5	R_0^4	G_0^7	G_0^6	G_0^5	G_0^4	B_0^7	B_0^6	B_0^5	B_0^4
2	α_1^3	α_1^2	α_1^1	α_1^0	R_1^7	R_1^6	R_1^5	R_1^4	G_1^7	G_1^6	G_1^5	G_1^4	B_1^7	B_1^6	B_1^5	B_1^4
3	α_2^3	α_2^2	α_2^1	α_2^0	R_2^7	R_2^6	R_2^5	R_2^4	G_2^7	G_2^6	G_2^5	G_2^4	B_2^7	B_2^6	B_2^5	B_2^4
4	α_3^2	α_3^3	α_3^1	α_3^0	R_3^7	R_3^6	R_3^5	R_3^4	G_3^7	G_3^6	G_3^5	G_3^4	B_3^7	B_3^6	B_3^5	B_3^4
5	α_4^2	α_4^3	α_4^1	α_4^0	R_4^7	R_4^6	R_4^5	R_4^4	G_4^7	G_4^6	G_4^5	G_4^4	B_4^7	B_4^6	B_4^5	B_4^4
6	α_5^2	α_5^3	α_5^1	α_5^0	R_5^7	R_5^6	R_5^5	R_5^4	G_5^7	G_5^6	G_5^5	G_5^4	B_5^7	B_5^6	B_5^5	B_5^4

$\alpha_x^3\alpha_x^2\alpha_x^1\alpha_x^0$: 0, 1 –2/32, 2 –4/32, 3 –6/32, 4 – 8/32, … …, 12 – 24/32, 13 – 26/32, 14 – 28/32, 15 –100%.

16-bit MPU, 32bpp mode (αRGB 8:8:8:8)

Order	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	G_0^7	G_0^6	G_0^5	G_0^4	G_0^3	G_0^2	G_0^1	G_0^0	B_0^7	B_0^6	B_0^5	B_0^4	B_0^3	B_0^2	B_0^1	B_0^0
2	α_0^7	α_0^6	α_0^5	α_0^4	α_0^3	α_0^2	α_0^1	α_0^0	R_0^7	R_0^6	R_0^5	R_0^4	R_0^3	R_0^2	R_0^1	R_0^0
3	G_1^7	G_1^6	G_1^5	G_1^4	G_1^3	G_1^2	G_1^1	G_1^0	B_1^7	B_1^6	B_1^5	B_1^4	B_1^3	B_1^2	B_1^1	B_1^0
4	α_1^7	α_1^6	α_1^5	α_1^4	α_1^3	α_1^2	α_1^1	α_1^0	R_1^7	R_1^6	R_1^5	R_1^4	R_1^3	R_1^2	R_1^1	R_1^0
5	G_2^7	G_2^6	G_2^5	G_2^4	G_2^3	G_2^2	G_2^1	G_2^0	B_2^7	B_2^6	B_2^5	B_2^4	B_2^3	B_2^2	B_2^1	B_2^0
6	α_2^7	α_2^6	α_2^5	α_2^4	α_2^3	α_2^2	α_2^1	α_2^0	R_2^7	R_2^6	R_2^5	R_2^4	R_2^3	R_2^2	R_2^1	R_2^0

8. 記憶體

8.1 SDRAM 控制器

SDRAM 控制器使用 bank interleave 方法有效的存取 SDRAM。硬體會自動執行初始化與自動更新的週期，RA8889 提供 128Mbit 容量給使用者使用。

8.1.1 SDRAM 初始化

SDRAM 在硬體被重置後與存取記憶體前必須被初始化，在硬體被重置後初始命令只會被執行一次。而這個命令在初始化後會被忽略，初始化步驟如下:

1. 設定 SDRAM 的屬性，透過寫入暫存器 REG[E0h]，根據暫存器定義可以定義 bank number (bit 5)、row addressing (bit 4-3) 與 column addressing (bit 2-1) 等等。
2. 設定 SDRAM 模式暫存器參數: 透過寫入暫存器為 REG[E1h] 可以設定 CAS 延遲。
3. 設定 SDRAM 暫存器 REG[E2h]、REG[E3h]刷新間隔，標準的刷新間隔時間為 15.6us。
4. 設定暫存器 REG[E4h] bit-0 為 1，開始 SDRAM 初始化處理。
5. 檢查 REG[E4h] bit0 的值直到它變為 1，如果變成 1 即可跳出初始化。Example:

Registers	MCLK = 140MHz SDRAM controller setting
PAGE0 REG[E0h]	0x29
PAGE0 REG[E1h]	0x03
PAGE0 REG[E2h]	0x89
PAGE0 REG[E3h]	0x08
PAGE0 REG[E4h]	0x01

8.2 SDRAM 資料結構

輸入圖像資料會被儲存在記憶體中如 1bpp、8bpp、16bpp、24bpp 或是具有對比度的圖像資料。

8.2.1 8bpp Display (RGB 3:3:2 Input Data)

Addr	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0000h	R_1^7	R_1^6	R_1^5	G_1^7	G_1^6	G_1^5	B_1^7	B_1^6	R_0^7	R_0^6	R_0^5	G_0^7	G_0^6	G_0^5	B_0^7	B_0^6
0002h	R_3^7	R_3^6	R_3^5	G_3^7	G_3^6	G_3^5	B_3^7	B_3^6	R_2^7	R_2^6	R_2^5	G_2^7	G_2^6	G_2^5	B_2^7	B_2^6
0004h	R_5^7	R_5^6	R_5^5	G_5^7	G_5^6	G_5^5	B_5^7	B_5^6	R_4^7	R_4^6	R_4^5	G_4^7	G_4^6	G_4^5	B_4^7	B_4^6
0006h	R_7^7	R_7^6	R_7^5	G_7^7	G_7^6	G_7^5	B_7^7	B_7^6	R_6^7	R_6^6	R_6^5	G_6^7	G_6^6	G_6^5	B_6^7	B_6^6
0008h	R_9^7	R_9^6	R_9^5	G_9^7	G_9^6	G_9^5	B_9^7	B_9^6	R_8^7	R_8^6	R_8^5	G_8^7	G_8^6	G_8^5	B_8^7	B_8^6
000Ah	R_{11}^7	R_{11}^6	R_{11}^5	G_{11}^7	G_{11}^6	G_{11}^5	B_{10}^7	B_{10}^6	R_{10}^7	R_{10}^6	R_{10}^5	G_{10}^7	G_{10}^6	G_{10}^5	B_{10}^7	B_{10}^6

8.2.2 16bpp Display (RGB 5:6:5 Input Data)

Addr	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0000h	R_0^7	R_0^6	R_0^5	R_0^4	R_0^3	G_0^7	G_0^6	G_0^5	G_0^4	G_0^3	G_0^2	B_0^7	B_0^6	B_0^5	B_0^4	B_0^3
0002h	R_1^7	R_1^6	R_1^5	R_1^4	R_1^3	G_1^7	G_1^6	G_1^5	G_1^4	G_1^3	G_1^2	B_1^7	B_1^6	B_1^5	B_1^4	B_1^3
0004h	R_2^7	R_2^6	R_2^5	R_2^4	R_2^3	G_2^7	G_2^6	G_2^5	G_2^4	G_2^3	G_2^2	B_2^7	B_2^6	B_2^5	B_2^4	B_2^3
0006h	R_3^7	R_3^6	R_3^5	R_3^4	R_3^3	G_3^7	G_3^6	G_3^5	G_3^4	G_3^3	G_3^2	B_3^7	B_3^6	B_3^5	B_3^4	B_3^3
0008h	R_4^7	R_4^6	R_4^5	R_4^4	R_4^3	G_4^7	G_4^6	G_4^5	G_4^4	G_4^3	G_4^2	B_4^7	B_4^6	B_4^5	B_4^4	B_4^3
000Ah	R_5^7	R_5^6	R_5^5	R_5^4	R_5^3	G_5^7	G_5^6	G_5^5	G_5^4	G_5^3	G_5^2	B_5^7	B_5^6	B_5^5	B_5^4	B_5^3

8.2.3 24bpp Display (RGB 8:8:8 Input Data)

Addr	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0000h	G_0^7	G_0^6	G_0^5	G_0^4	G_0^3	G_0^2	G_0^1	G_0^0	B_0^7	B_0^6	B_0^5	B_0^4	B_0^3	B_0^2	B_0^1	B_0^0
0002h	B_1^7	B_1^6	B_1^5	B_1^4	B_1^3	B_1^2	B_1^1	B_1^0	R_0^7	R_0^6	R_0^5	R_0^4	R_0^3	R_0^2	R_0^1	R_0^0
0004h	R_1^7	R_1^6	R_1^5	R_1^4	R_1^3	R_1^2	R_1^1	R_1^0	G_1^7	G_1^6	G_1^5	G_1^4	G_1^3	G_1^2	G_1^1	G_1^0
0006h	G_2^7	G_2^6	G_2^5	G_2^4	G_2^3	G_2^2	G_2^1	G_2^0	B_2^7	B_2^6	B_2^5	B_2^4	B_2^3	B_2^2	B_2^1	B_2^0
0008h	B_3^7	B_3^6	B_3^5	B_3^4	B_3^3	B_3^2	B_3^1	B_3^0	R_2^7	R_2^6	R_2^5	R_2^4	R_2^3	R_2^2	R_2^1	R_2^0
000Ah	R_3^7	R_3^6	R_3^5	R_3^4	R_3^3	R_3^2	R_3^1	R_3^0	G_3^7	G_3^6	G_3^5	G_3^4	G_3^3	G_3^2	G_3^1	G_3^0

8.2.4 6 bit index colors/pixel Index with opacity (αRGB 2:2:2:2)

Addr	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0000h	α_1^3	α_1^2	Index color of pixel 1						α_0^3	α_0^2	Index color of pixel 0					
0002h	α_3^3	α_3^2	Index color of pixel 3						α_2^3	α_2^2	Index color of pixel 2					
0004h	α_5^3	α_5^2	Index color of pixel 5						α_4^3	α_4^2	Index color of pixel 4					
0006h	α_7^3	α_7^2	Index color of pixel 7						α_6^3	α_6^2	Index color of pixel 6					
0008h	α_9^3	α_9^2	Index color of pixel 9						α_8^3	α_8^2	Index color of pixel 8					
000Ah	α_{11}^3	α_{11}^2	Index color of pixel 11						α_{10}^3	α_{10}^2	Index color of pixel 10					

$\alpha_x^3\alpha_x^2$: 0, 1 – 11/32, 2 – 20/32, 3 – 100%

8.2.5 12 bit RGB data with opacity (αRGB 4:4:4:4)

Addr	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0000h	α_0^3	α_0^2	α_0^1	α_0^0	R_0^7	R_0^6	R_0^5	R_0^4	G_0^7	G_0^6	G_0^5	G_0^4	B_0^7	B_0^6	B_0^5	B_0^4
0002h	α_1^3	α_1^2	α_1^1	α_1^0	R_1^7	R_1^6	R_1^5	R_1^4	G_1^7	G_1^6	G_1^5	G_1^4	B_1^7	B_1^6	B_1^5	B_1^4
0004h	α_2^3	α_2^2	α_2^1	α_2^0	R_2^7	R_2^6	R_2^5	R_2^4	G_2^7	G_2^6	G_2^5	G_2^4	B_2^7	B_2^6	B_2^5	B_2^4
0006h	α_3^2	α_3^3	α_3^1	α_3^0	R_3^7	R_3^6	R_3^5	R_3^4	G_3^7	G_3^6	G_3^5	G_3^4	B_3^7	B_3^6	B_3^5	B_3^4
0008h	α_4^2	α_4^3	α_4^1	α_4^0	R_4^7	R_4^6	R_4^5	R_4^4	G_4^7	G_4^6	G_4^5	G_4^4	B_4^7	B_4^6	B_4^5	B_4^4
000Ah	α_5^2	α_5^3	α_5^1	α_5^0	R_5^7	R_5^6	R_5^5	R_5^4	G_5^7	G_5^6	G_5^5	G_5^4	B_5^7	B_5^6	B_5^5	B_5^4

$\alpha_x^3\alpha_x^2\alpha_x^1\alpha_x^0$: 0, 1 –2/32, 2 –4/32, 3 –6/32, 4 – 8/32, … …, 12 – 24/32, 13 – 26/32, 14 – 28/32, 15 –100%.

8.2.6 24bits RGB data with opacity (αRGB 8:8:8:8)

Addr	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0000h	G_0^7	G_0^6	G_0^5	G_0^4	G_0^3	G_0^2	G_0^1	G_0^0	B_0^7	B_0^6	B_0^5	B_0^4	B_0^3	B_0^2	B_0^1	B_0^0
0002h	α_0^7	α_0^6	α_0^5	α_0^4	α_0^3	α_0^2	α_0^1	α_0^0	R_0^7	R_0^6	R_0^5	R_0^4	R_0^3	R_0^2	R_0^1	R_0^0
0004h	G_1^7	G_1^6	G_1^5	G_1^4	G_1^3	G_1^2	G_1^1	G_1^0	B_1^7	B_1^6	B_1^5	B_1^4	B_1^3	B_1^2	B_1^1	B_1^0
0006h	α_1^7	α_1^6	α_1^5	α_1^4	α_1^3	α_1^2	α_1^1	α_1^0	R_1^7	R_1^6	R_1^5	R_1^4	R_1^3	R_1^2	R_1^1	R_1^0
0008h	G_2^7	G_2^6	G_2^5	G_2^4	G_2^3	G_2^2	G_2^1	G_2^0	B_2^7	B_2^6	B_2^5	B_2^4	B_2^3	B_2^2	B_2^1	B_2^0
000Ah	α_2^7	α_2^6	α_2^5	α_2^4	α_2^3	α_2^2	α_2^1	α_2^0	R_2^7	R_2^6	R_2^5	R_2^4	R_2^3	R_2^2	R_2^1	R_2^0

8.3 Color Palette RAM

Addr	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0000h	R_0^7	R_0^6	R_0^5	R_0^4	G_0^7	G_0^6	G_0^5	G_0^4	B_0^7	B_0^6	B_0^5	B_0^4
0002h	R_1^7	R_1^6	R_1^5	R_1^4	G_1^7	G_1^6	G_1^5	G_1^4	B_1^7	B_1^6	B_1^5	B_1^4
0004h	R_2^7	R_2^6	R_2^5	R_2^4	G_2^7	G_2^6	G_2^5	G_2^4	B_2^7	B_2^6	B_2^5	B_2^4
0006h	R_3^7	R_3^6	R_3^5	R_3^4	G_3^7	G_3^6	G_3^5	G_3^4	B_3^7	B_3^6	B_3^5	B_3^4
0008h	R_4^7	R_4^6	R_4^5	R_4^4	G_4^7	G_4^6	G_4^5	G_4^4	B_4^7	B_4^6	B_4^5	B_4^4
000Ah	R_5^7	R_5^6	R_5^5	R_5^4	G_5^7	G_5^6	G_5^5	G_5^4	B_5^7	B_5^6	B_5^5	B_5^4

*若 BTE 致能且 BTE 的目的影像為 8bpp，則 Bit[1:0], Bit[4] & Bit[8]為無效。

9. 顯示資料路徑

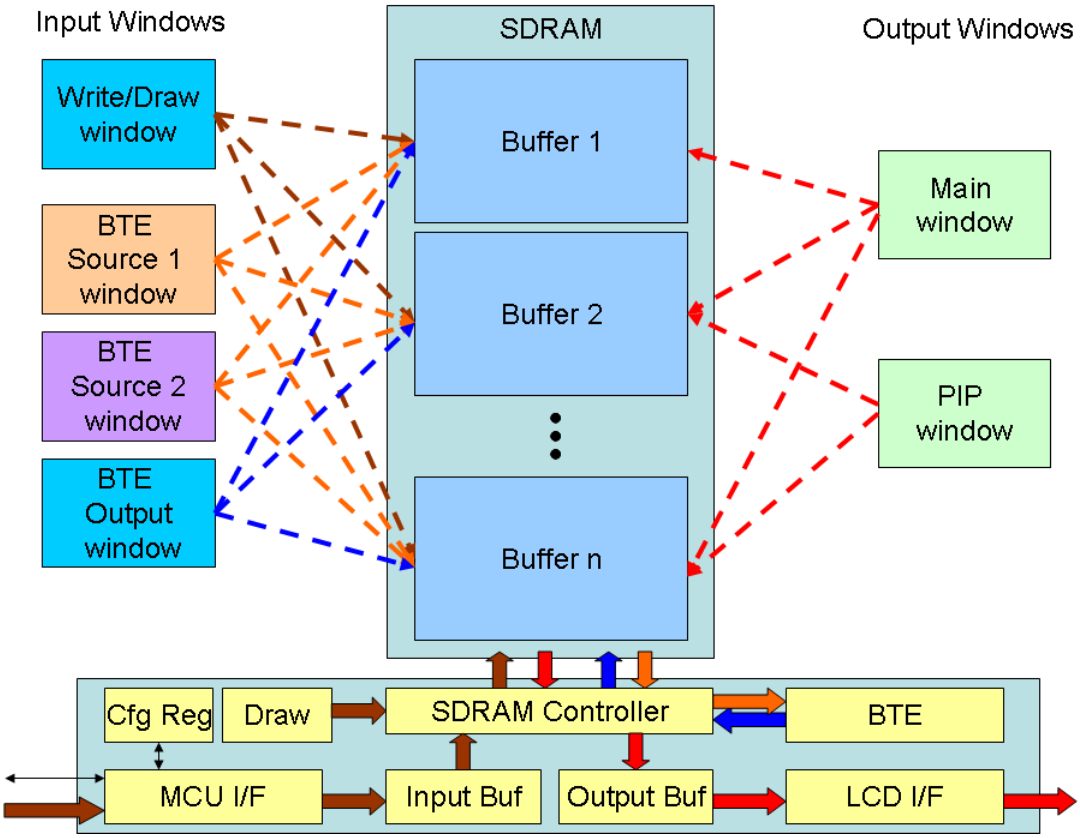


Figure 9-1 : Display Data path

10. LCD 介面

10.1 LCD 腳位對應

此功能不同的色深是由 REG[01h] bit 4-3 來控制的，如下表所示。

腳位名稱	TFT 介面		
色深	數位介面		
	16bpp	18bpp	24bpp
XVSYNC	XVSYNC	XVSYNC	XVSYNC
XHSYNC	XHSYNC	XHSYNC	XHSYNC
XPCLK	XPCLK	XPCLK	XPCLK
XDE	XDE	XDE	XDE
XPDAT[0]	GPIO-D0 / XKIN[1]		B0
XPDAT[1]	GPIO-D1 / XKIN[2]		B1
XPDAT[2]	GPIO-D6 / XKIN[4]	B0	B2
XPDAT[3]	B0	B1	B3
XPDAT[4]	B1	B2	B4
XPDAT[5]	B2	B3	B5
XPDAT[6]	B3	B4	B6
XPDAT[7]	B4	B5	B7
XPDAT[8]	GPIO-D2 / XKIN[3]		G0
XPDAT[9]	GPIO-D3 / XKOUT[3]		G1
XPDAT[10]	G0	G0	G2
XPDAT[11]	G1	G1	G3
XPDAT[12]	G2	G2	G4
XPDAT[13]	G3	G3	G5
XPDAT[14]	G4	G4	G6
XPDAT[15]	G5	G5	G7
XPDAT[16]	GPIO-D4 / XKOUT[1]		R0
XPDAT[17]	GPIO-D5 / XKOUT[2]		R1
XPDAT[18]	GPIO-D7 / XKOUT[4]	R0	R2
XPDAT[19]	R0	R1	R3
XPDAT[20]	R1	R2	R4
XPDAT[21]	R2	R3	R5
XPDAT[22]	R3	R4	R6
XPDAT[23]	R4	R5	R7

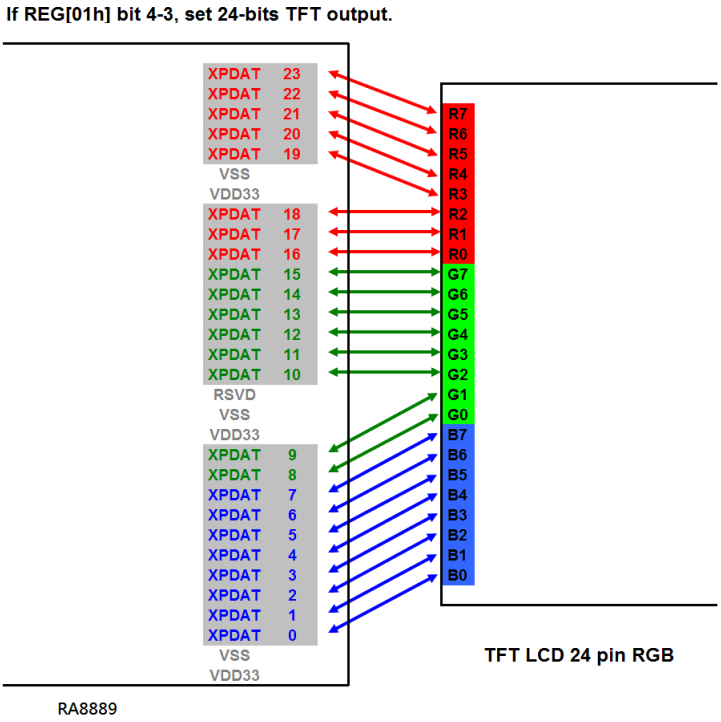


Figure 10-1 : RA8889 color depth 24bit with LCD panel pin connect

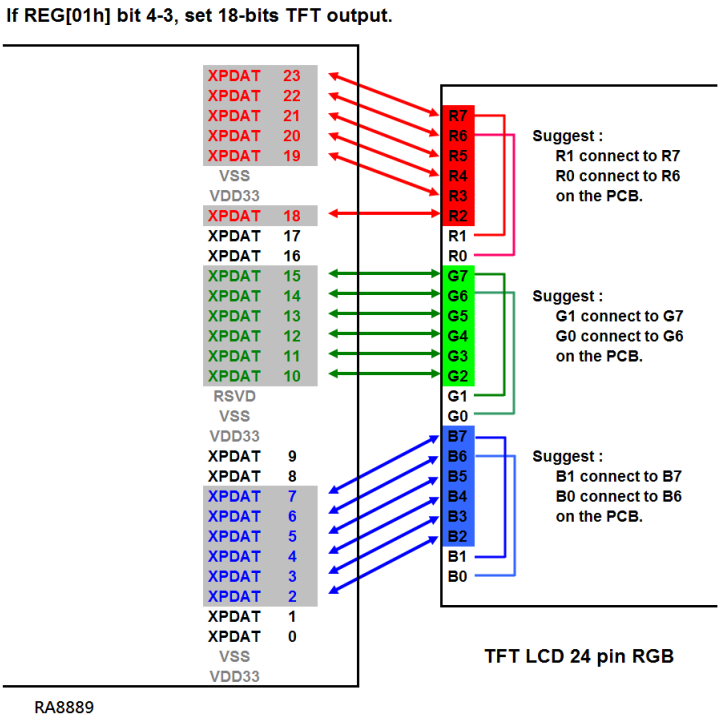


Figure 10-2 : RA8889 color depth 18bit with LCD panel pin connect

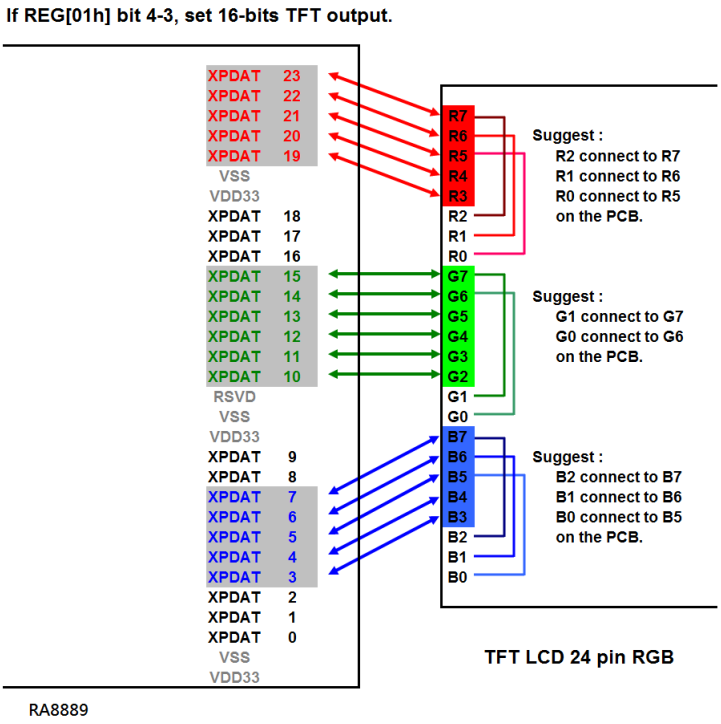


Figure 10-3 : RA8889 color depth 16bit with LCD panel pin connect

10.2 LCD 並列介面時序圖

平板顯示的時序參數如下圖所示:

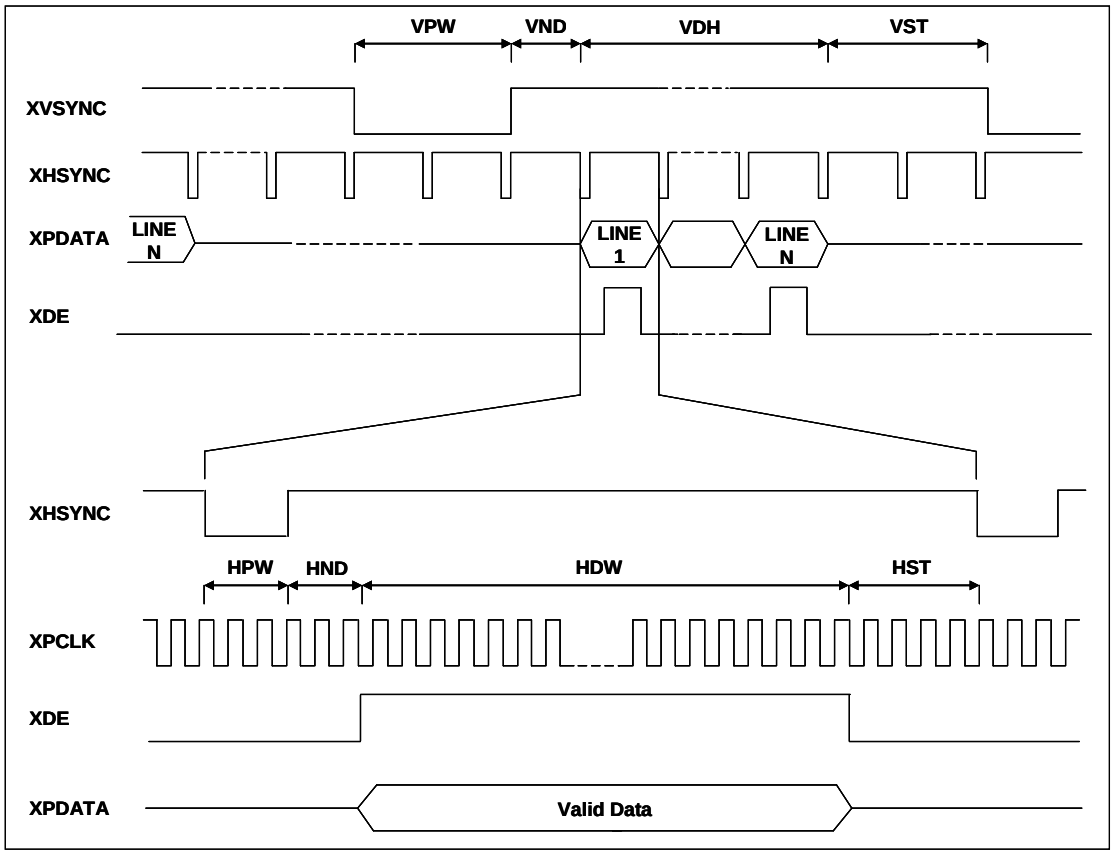


Figure 10-4 : Digital TFT Panel Timing

11. Display 功能

11.1 彩條 (Color Bar) 顯示測試

彩條 (Color Bar) 顯示測試並不需要搭配 SDRAM 使用。設定 REG[12h] bit5=1，可以進行彩條測試。

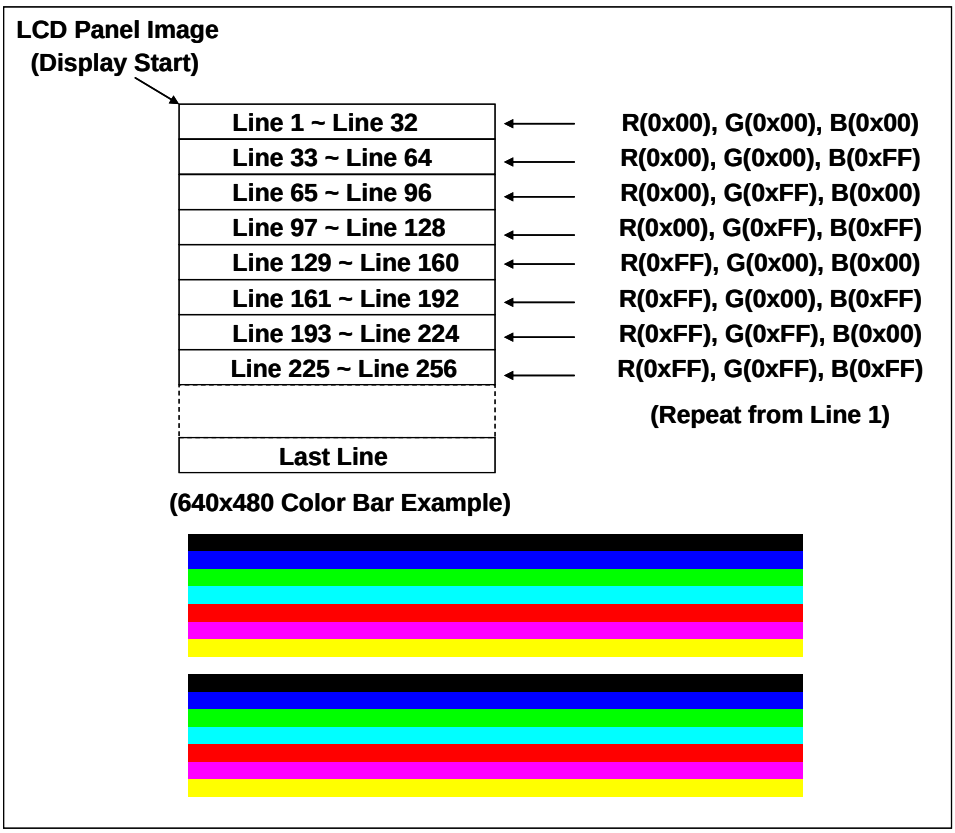


Figure 11-1: Color Bar Display Test

11.2 主視窗

經由定義 LCD 螢幕解析度，可定義主螢幕(參考 REG[14h] ~REG[1Fh])。使用上可先設定好不同的顯示緩衝區，再經由主視窗相關的暫存器 (參考 REG[20h] ~REG[29h]) 致能並選擇不同的緩衝區，來顯示不同的圖像。

11.2.1 設定不同的圖像緩衝區

SDRAM 可以被分成數個圖像緩衝區的空間，其最大緩衝區的數量受記憶體容量限制。舉例說明:圖像大小為 800x600 256 color 在 128 Mbits SDRAM 上可以有 34 個圖像緩衝區 (圖像寬度 x 圖像高度 x 圖像色深 bpp x 圖像數 < 128 Mbits)。為了定義圖像大小，寫圖像之前必須設定底圖起始位置、底圖寬度與工作視窗範圍 (參考 REG[50h] ~REG[5Eh])。

11.2.2 寫入圖像至圖像緩衝區

底圖 (canvas) 是對應於讀寫圖像資料的記憶體空間。使用者必須設定底圖起始位置、底圖寬度 (參考 REG[50h] ~REG[55h]) 來指明圖像大小，並且設定工作視窗範圍 (參考 REG[56h] ~REG[5Eh]) 來寫入緩衝區的圖像資料。

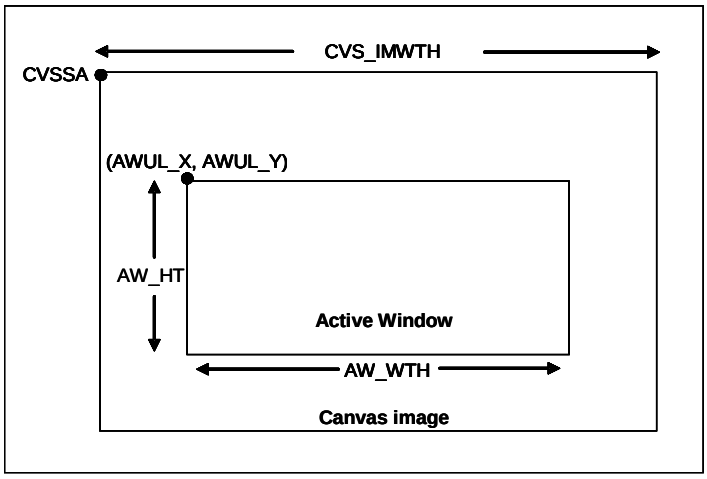


Figure 11-2

11.2.3 顯示主視窗圖像

主圖像是 LCD 螢幕上的顯示圖像，下面的圖是主視窗顯示的流程圖，使用者必須按照步驟來。

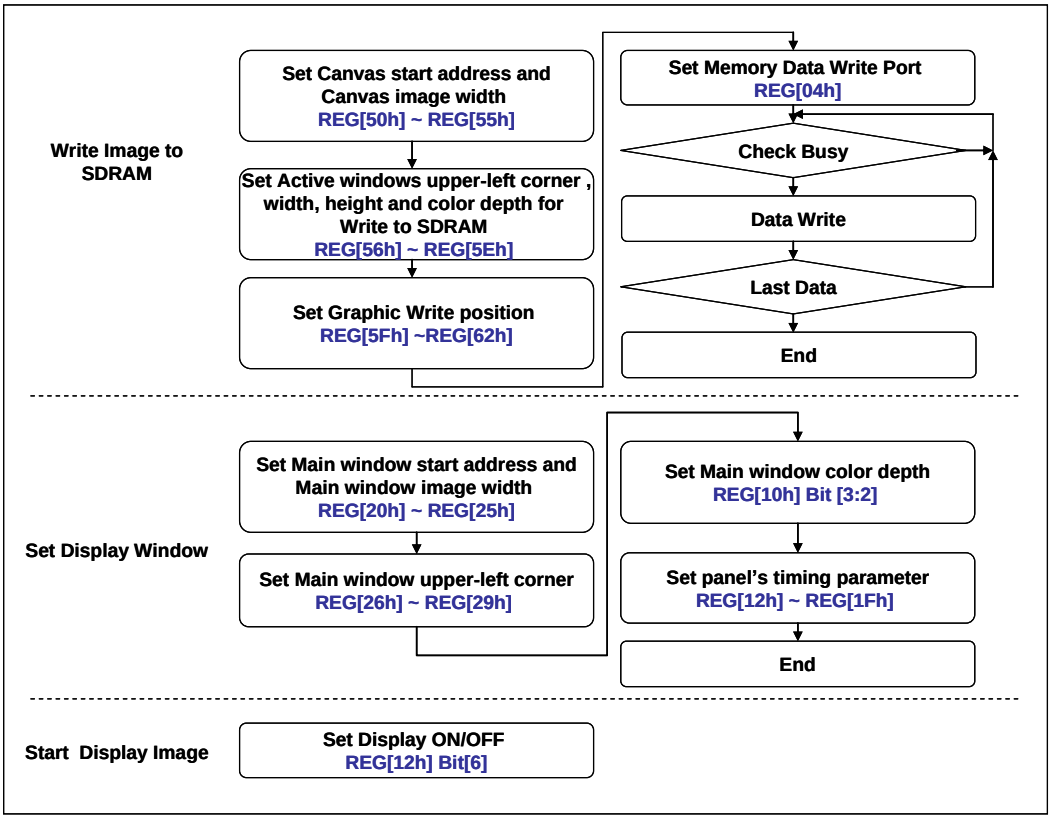


Figure 11-3

11.2.4 切換主視窗圖像

主視窗圖像可以由致能的圖像緩衝區來。使用者可以透過設定主視窗相關暫存器（參考 REG[20h] ~REG[29h]）來切換圖像緩衝區。

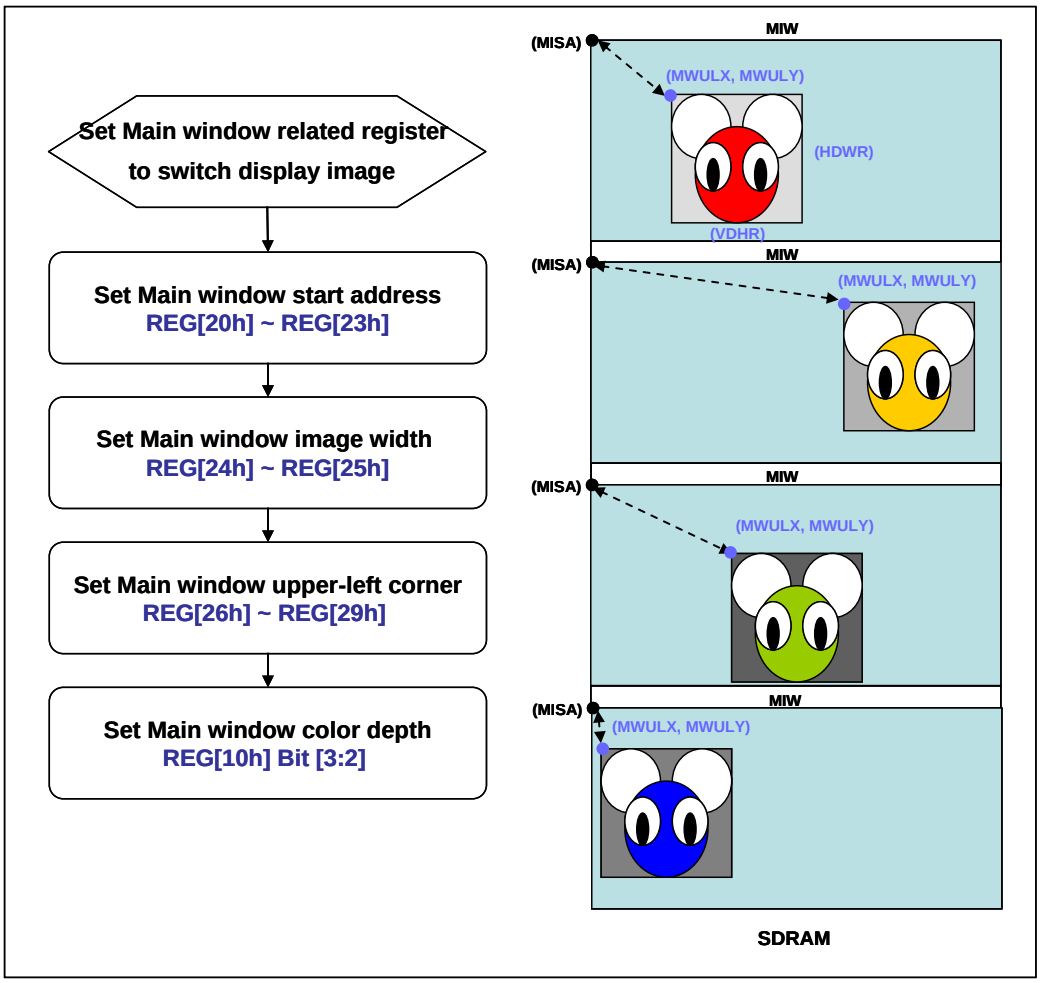


Figure 11-4

11.3 畫中畫 (PIP) 視窗

RA8889 在主視窗下可以支援兩個畫中畫視窗。 畫中畫視窗並不支援重疊透明顯示，畫中畫功能提供使用者致能或禁能顯示而不需要去覆寫主顯示視窗的圖像資料。如果畫中畫 1 與畫中畫 2 是重疊的，那麼畫中畫 1 視窗一定顯示在畫中畫 2 視窗上。

畫中畫視窗的大小與位置是被暫存器 REG[2Ah] ~ REG[3Bh] 與 REG[11h] 指定的。畫中畫 1 與畫中畫 2 視窗使用相同的暫存器，根據 REG[10h] Bit[4] 來選擇 REG [2Ah ~ 3Bh] 是畫中畫 1 或畫中畫 2 視窗的參數，而在使用這個功能上必須先設定畫中畫視窗的相關參數。畫中畫視窗大小與啟始位置解析度在水平上是 4 像素，垂直解析度則為 1 條掃描線。

註：當 REG[12h] Bit3 VDIR = 1，PIP 視窗、圖形游標、文字游標都將會被自動禁能。

11.3.1 畫中畫 (PIP) 視窗的設定

一個畫中畫視窗的位置與大小必須設定畫中畫圖像起始位置、畫中畫圖像寬度、畫中畫顯示 X/Y 座標、畫中畫圖像 X/Y 座標、畫中畫視窗色深、畫中畫視窗寬度與畫中畫視窗高度暫存器。畫中畫 1 與畫中畫 2 視窗共用相同的暫存器，並且根據 REG[10h] Bit[4] 來選擇 REG [2Ah ~ 3Bh] 是畫中畫 1 還是畫中畫 2 視窗的參數。

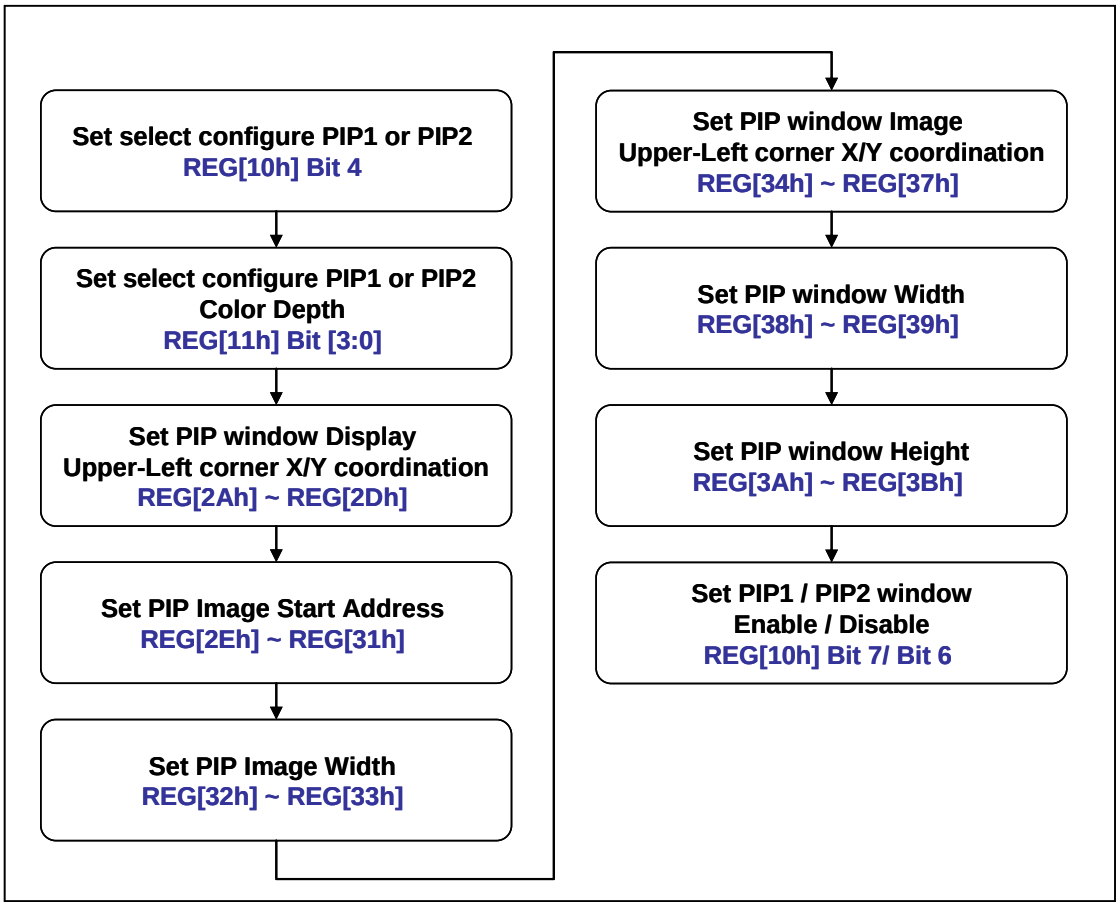


Figure 11-5

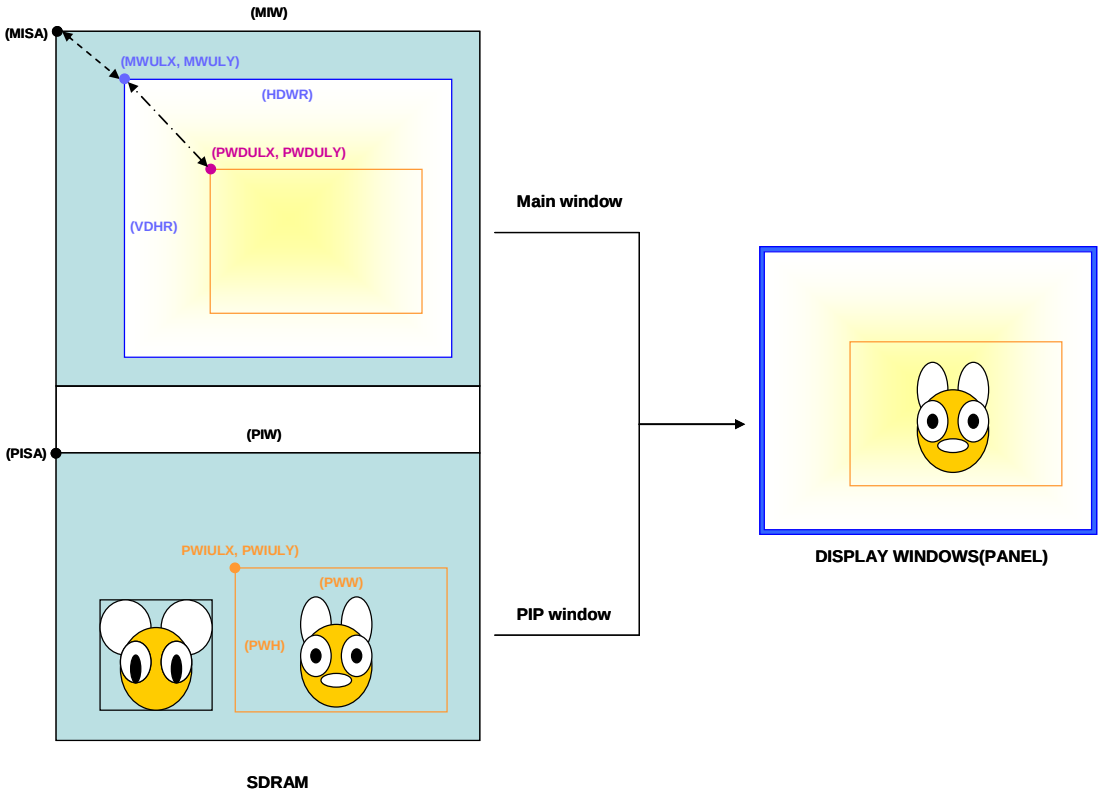

(MIW)
(MISA)
(MWULX, MWULY)
(HDWR)
(PWDULX, PWDULY)
(VDHR)
Main window
(PIW)
(PISA)
PWIULX, PWIULY)
(PWW)
(PWH)
PIP window
DISPLAY WINDOWS(PANEL)
SDRAM

Figure 11-6

11.3.2 畫中畫 (PIP) 視窗顯示位置與畫中畫 (PIP) 圖像位置

畫中畫視窗經由設定 PWDULX 與 PWDULY 來更改不同的顯示位置，而經由設定 PISA、PIW、PWIULX、PWIULY 可以更改畫中畫圖像位置，這個方法不會改變在記憶體中的圖像資料，但是可以很簡單的更改被顯示在畫中畫中的圖像。

下面的例子顯示一個主視窗與一個畫中畫視窗，畫中畫視窗可以經由更改畫中畫圖像位置來顯示不同的畫中畫圖像。

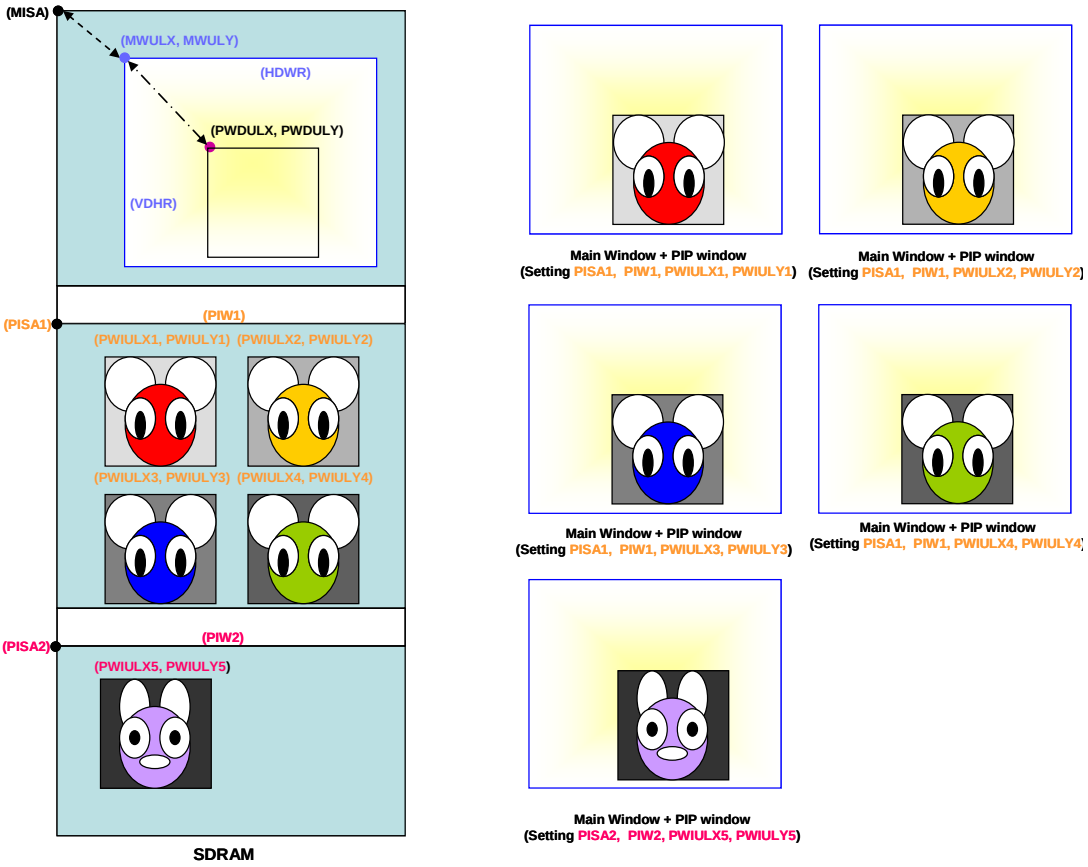


Figure 11-7

11.4 旋轉與鏡射

大部分顯示器更新方式都是橫向-由左至右由上而下，而儲存在記憶體中的圖像也是相同的方法。旋轉功能是設計成逆時針 90° 或 180° 旋轉圖像，對使用者來說是無負擔的，因為旋轉主要靠硬體就可完成的。旋轉功能主要是靠寫入記憶體方向旋轉來達成 (參考 REG[02h] bit 2-1)，在效率方面使用硬體完成旋轉功能較軟體完成旋轉更好。

鏡射功能指的是左右鏡射，鏡射是使用硬體來達成功能，因此對使用者是無負擔的；鏡射功能在記憶體寫入時需要設定暫存器(參考 REG[02h] bit 2-1)。在效率方面使用硬體完成旋轉功能較軟體完成旋轉更好。

註：當 REG[12h] Bit3 VDIR = 1，PIP 視窗、圖形游標、文字游標都將會被自動禁能。

REG[02h] bit 2-1 提供主控端寫入的記憶體方向控制 (只提供給繪圖模式使用)

00b: 左→右 然後 上→下 (初始值)

01b: 右→左 然後 上→下 (水平翻轉)

10b: 上→下 然後 左→右 (向右旋轉 90°並且水平翻轉)

11b: 下→上 然後 左→右 (向左旋轉 90°)

根據 REG[12h] bit 3 (VDIR) 可能會產生其他的效果。

舉例，如果原圖如下:

Figure 11-8

➔ **If HDIR (REG[12h] bit 4) == 0/1 and VDIR (REG[12h] bit 3) == 0**

設定 REG[02h]bit 2-1 為 00b，其意義是寫入圖像資料從左到右然後再上到下，這可以顯示原始圖像。

Figure 11-9

HDIR (REG[12h] bit 4) == 1 and VDIR (REG[12h] bit 3) == 0

Figure 11-10

設定 REG[02h]bit 2-1 為 01b，表示寫入圖像資料從右到左然後從上到下，因此顯示的將是水平鏡射的圖像。

HDIR (REG[12h] bit 4) == 0 and VDIR (REG[12h] bit 3) == 0

Figure 11-11

HDIR (REG[12h] bit 4) == 1 and VDIR (REG[12h] bit 3) == 0

Figure 11-12

設定 REG[02h]bit 2-1 為 10b 表示寫入圖像資料從上到下然後從左到右，因此顯示的將是向右旋轉 90°且水平鏡射的圖像。

HDIR (REG[12h] bit 4) == 0 and VDIR (REG[12h] bit 3) == 0

Figure 11-13

HDIR (REG[12h] bit 4) == 1 and VDIR (REG[12h] bit 3) == 0

Figure 11-14

設定 REG[02h]bit 2-1 為 11b 表示寫入圖像資料從下到上然後從左到右，因此顯示的將是向左旋轉 90°的圖像。

HDIR (REG[12h] bit 4) == 0 and VDIR (REG[12h] bit 3) == 0

Figure 11-15

HDIR (REG[12h] bit 4) == 1 and VDIR (REG[12h] bit 3) == 0

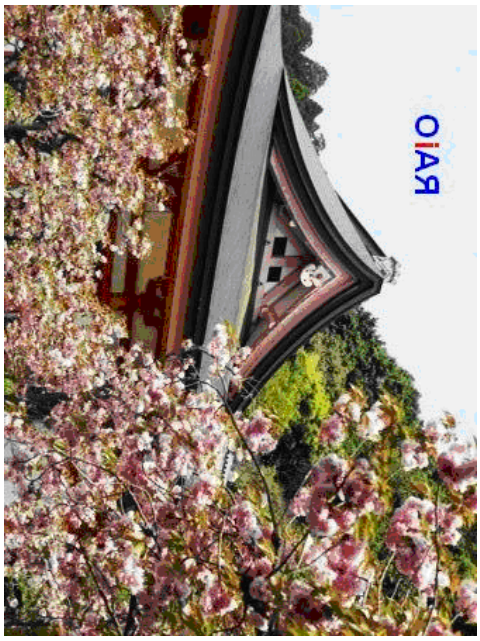


Figure 11-16

➔ **If VDIR (REG[12h] bit 3) == 1**

設定 REG[02h]bit 2-1 為 00b，因此顯示的將是因此顯示的將是旋轉 180˚的圖像。

HDIR (REG[12h] bit 4) == 0 and VDIR (REG[12h] bit 3) == 1

Figure 11-17

HDIR (REG[12h] bit 4) == 1 and VDIR (REG[12h] bit 3) == 1

Figure 11-18

設定 REG[02h]bit 2-1 為 01b，因此顯示的將是旋轉 180°的圖像。

HDIR (REG[12h] bit 4) == 0 and VDIR (REG[12h] bit 3) == 1

Figure 11-19

HDIR (REG[12h] bit 4) == 1 and VDIR (REG[12h] bit 3) == 1

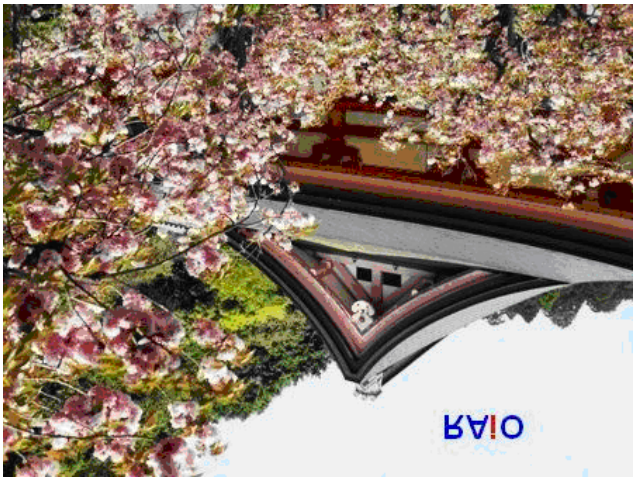


Figure 11-20

設定 REG[02h]bit 2-1 為 10b，因此顯示的將是向左旋轉 90°的圖像。

HDIR (REG[12h] bit 4) == 0 and VDIR (REG[12h] bit 3) == 1

Figure 11-21

HDIR (REG[12h] bit 4) == 1 and VDIR (REG[12h] bit 3) == 1

Figure 11-22

設定 REG[02h]bit 2-1 為 11b，因此顯示的將是向右旋轉 90°的圖像。

HDIR (REG[12h] bit 4) == 0 and VDIR (REG[12h] bit 3) == 1

Figure 11-23

HDIR (REG[12h] bit 4) == 1 and VDIR (REG[12h] bit 3) == 1

Figure 11-24

12. 幾何繪圖引擎

12.1 橢圓/圓

RA8889 提供 2D 繪圖引擎，可以讓使用者不增加 MPU 負擔的情形下即可在底圖上畫圓與橢圓。經由設定圓與橢圓的圓心 REG[7Bh~7Eh]，橢圓/圓長短軸半徑 REG[77h~7Ah]，橢圓/圓顏色 REG[D2h~D4h]，指定繪橢圓/圓 REG[76h] Bit5~4 為 00h，最後在致能開始畫圓功能 REG[76h]Bit7=1，這樣 RA8889 就會在底圖上畫橢圓與圓，更進一步的，使用者可以透過設定 REG[76h]Bit6=1 對橢圓/圓做填滿的動作。

註：圓心必須在工作視窗 (Active Window) 內。

畫橢圓/圓的流程圖如下:

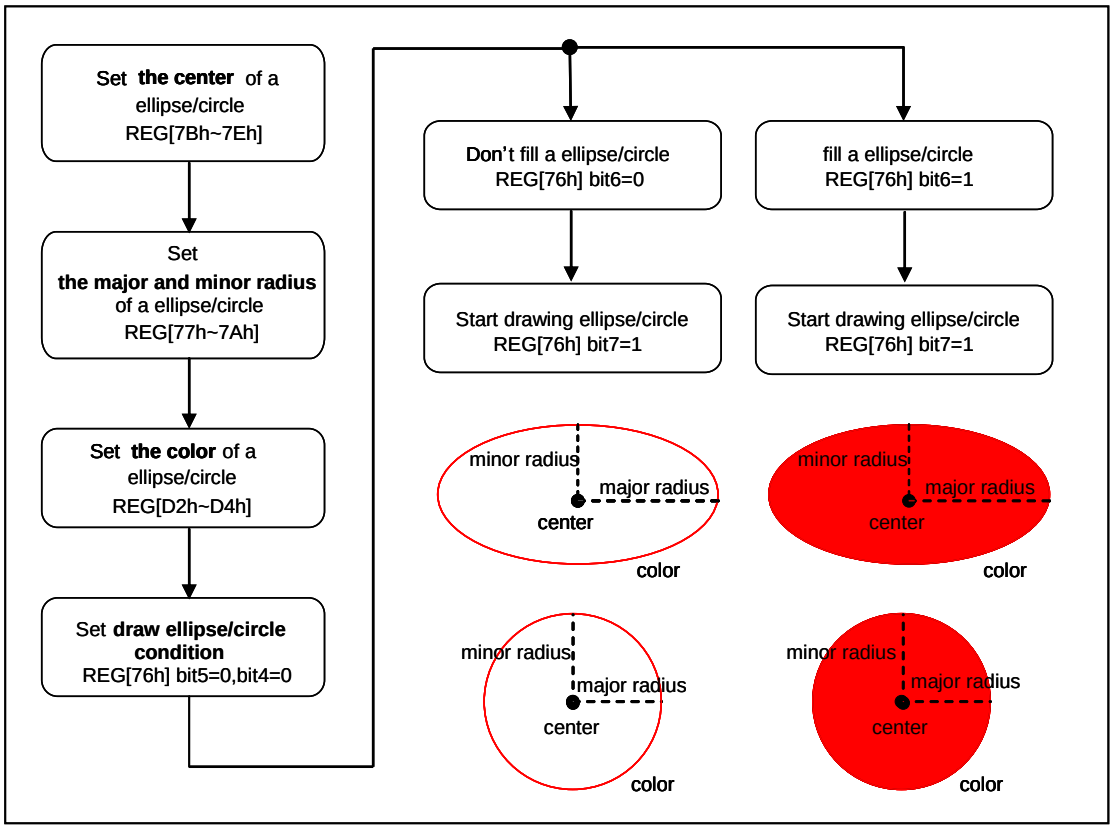


Figure 12-1

12.2 曲線

RA8889 支援畫曲線功能，使用者可以在不增加 MPU 負擔的情形下達到畫曲線功能。畫曲線功能必須設定曲線的圓心 REG[7Bh~7Eh] ，曲線的長短軸半徑 REG[77h~7Ah]，曲線的顏色 REG[D2h~D4h]，設定 REG[76h] Bit5~4 為 01b 以選定曲線，橢圓的曲線線段的選擇 REG[76h] Bit[1:0]，最後在致能繪圖功能 REG[76h] Bit7 = 1，RA8889 將會在底圖上繪出對應的曲線，更進一步的，使用者可以做填滿的動作。

註: 曲線的圓心必須在工作視窗 (Active Windows) 內

下面曲線繪製的流程圖:

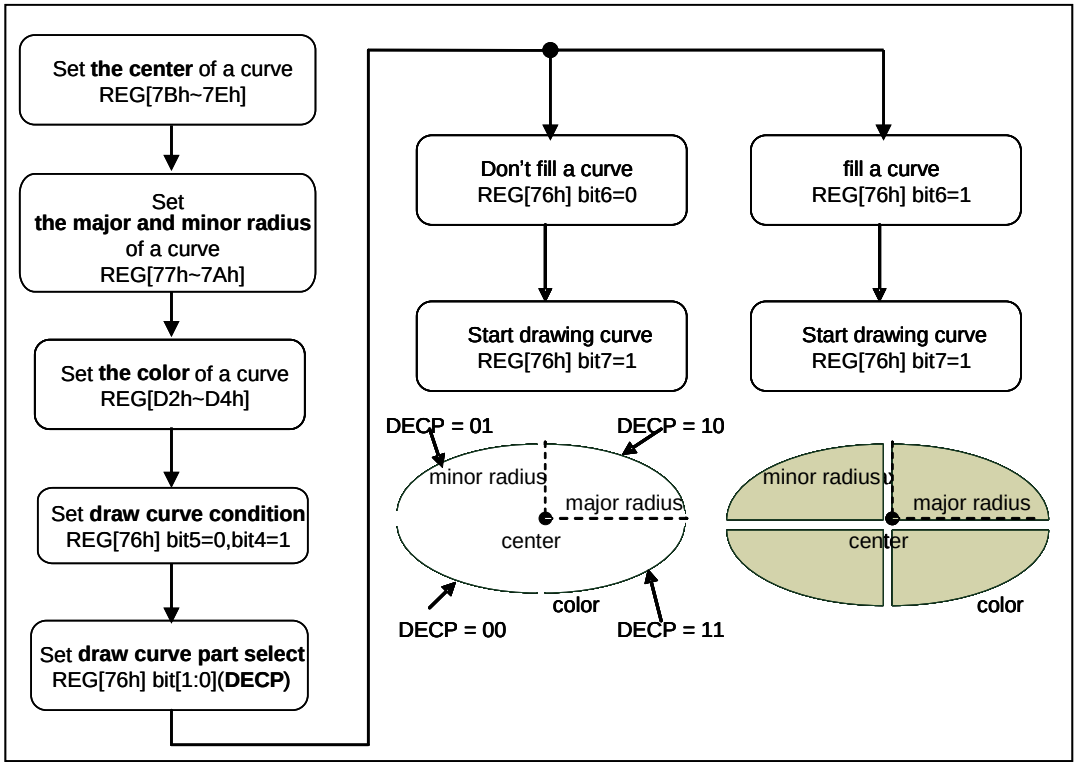


Figure 12-2

12.3 矩形

RA8889 支援矩形繪製功能，使用者可以在不增加 MPU 負擔的情形下達成繪製需求。經由設定矩形起始位置 REG[68h~6Bh]，與矩形結束位置 REG[6Ch~6Fh]，矩形顏色設定 REG[D2h~D4h]，最後在指定要繪製的圖形是矩形 REG[76h] Bit4=1，Bit0=0，並且致能 REG[76h] Bit7 = 1，那麼 RA8889 將會在底圖上繪出對應的矩形。更進一步的，使用者可以對矩形做填滿的動作 REG[76h] Bit6 = 1。

註 :矩形的起始位置與結束位置必須在工作視窗內。

下面為矩形繪製的流程圖:

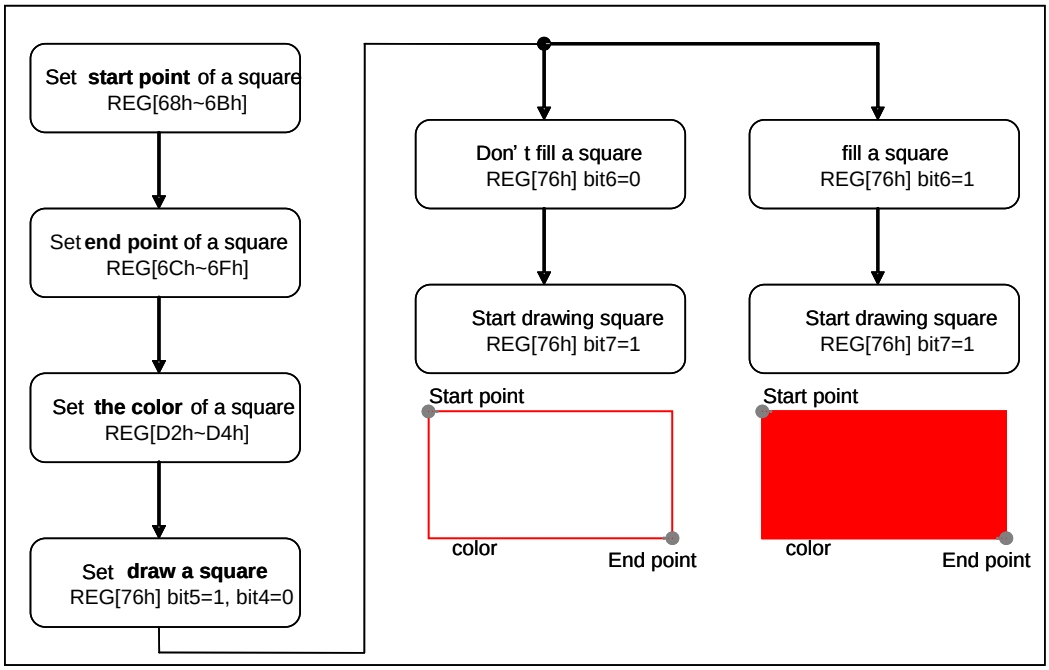


Figure 12-3 : Geometric Pattern Drawing- Draw Rectangle

12.4 線

RA8889 支援線段繪製，使用者可以使用少量的 MPU 週期達成線段繪製的功能。經由設定線段起始點 REG[68h~6Bh]，與線段結束點 REG[6Ch~6Fh]，線段顏色 REG[D2h~D4h]，最後設定 REG[67h] Bit1 = 0 指定為繪製線段，並且致能 REG[67h] Bit7 = 1，那麼 RA8889 將會在底圖 (canvas) 上繪製線段。

註：線段的起始點與結束點必須是在工作視窗 (Active windows) 內。下面是繪製線段的流程圖:

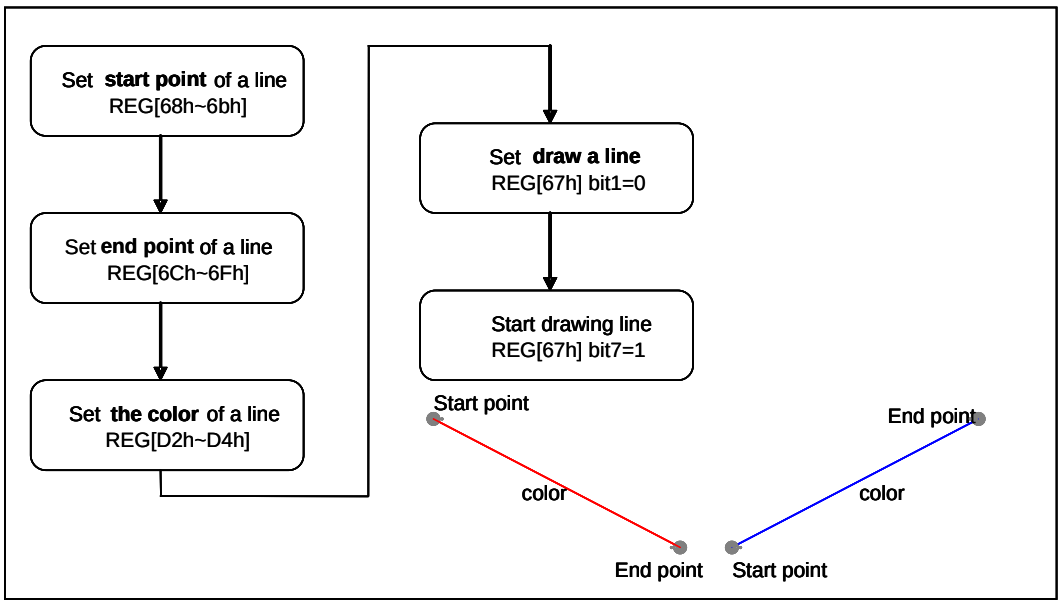


Figure 12-4 : Geometric Pattern Drawing- Draw Line

12.5 三角形

RA8889 支援繪製三角形，使用者可以使用少量 MPU 週期達成繪製三角形。經由設定三角形的點 1 REG[68h~6Bh]，點 2 REG[6Ch~6Fh]，點 3 REG[70h~73h] 與顏色 REG[D2h~D4h]，最後在設定繪製圖形為三角形 REG[67h] Bit1 = 1 並且致能 REG[67h] Bit7 = 1，那麼 RA8889 將會在底圖 (canvas) 上繪製三角形。更進一步的，使用者可對三角形做填滿的動作 REG[67h] Bit5 = 1。

註: 三角形點 1 點 2 點 3 必須在工作視窗 (Active windows)。

下面是繪製三角形的流程圖:

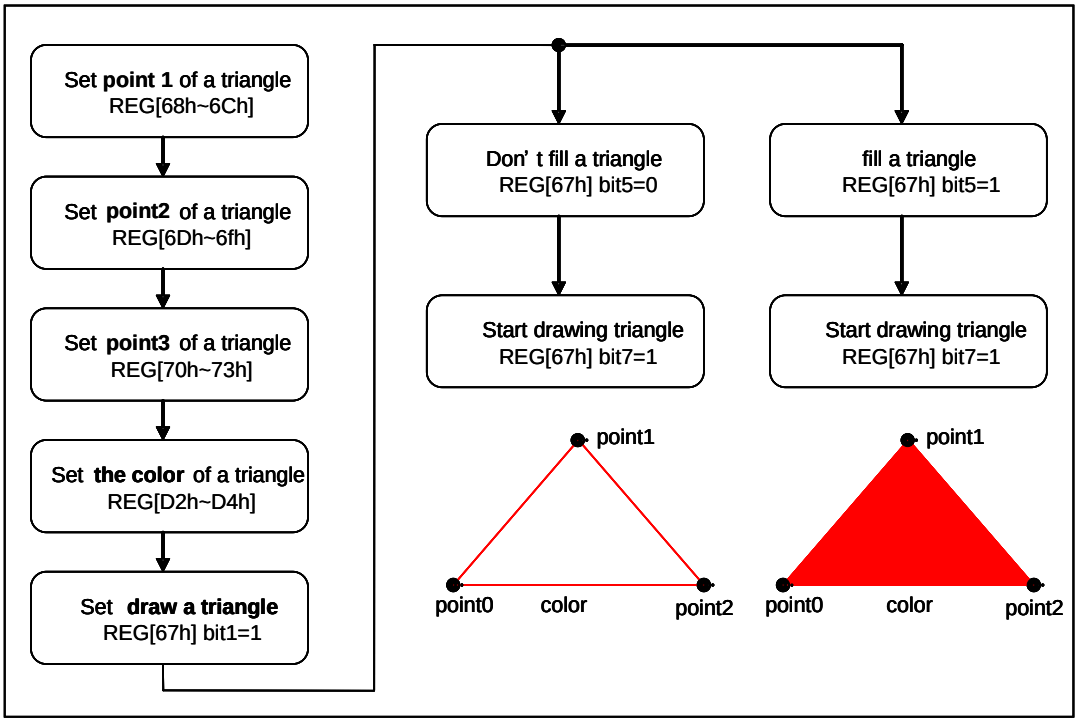


Figure 12-5 : Geometric Pattern Drawing- Draw Triangle

12.6 圓角矩形

RA8889 支援繪製圓角矩形，使用者可以使用少量的 MPU 週期達成繪製圓角矩形。經由設定圓角矩形起始點 REG[68h~6Bh] ，結束點 REG[6Ch~6Fh]，圓角矩形長短軸半徑 REG[77h~7Ah]，顏色 REG[D2h~D4h]，最後設定繪製圖形為圓角矩形 REG[76h] Bit5~4 為 11b，並且致能 REG[76h] Bit7 = 1，那麼 RA8889 將會在底圖上繪製圓角矩形，更進一步的，使用者可以設定填滿功能 REG[76h] Bit6 = 1。

註 1: (結束點 X – 起始點 X) 必須大於 (2*長軸半徑(major radius)+ 1)

(結束點 Y – 起始點 Y) 必須大於 (2*短軸(minor radius) + 1)

註 2: 起始點與結束點必須要在工作視窗 (Active windows)。

下面是繪製圓角矩形的流程圖:

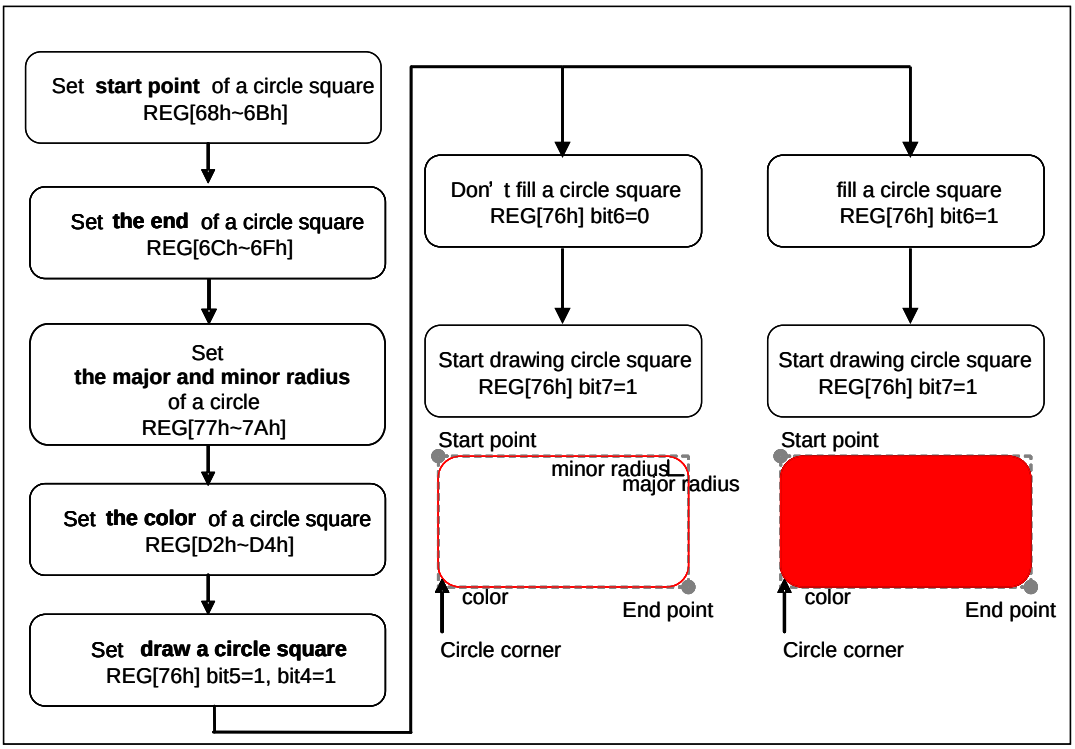


Figure 12-6 : Geometric Pattern Drawing- Draw Circle-Square

13. 區塊傳輸引擎 (BTE)

RA8889 內建 2D 區塊傳輸引擎(BTE)，可以增加區塊傳輸的效率。在指定區塊資料結合某些邏輯傳輸操作中，RA8889 內的 BTE 硬體可以提升傳輸速度，這可以簡化 MPU 的程序。因此這個章節主要是討論 BTE 的操作與功能。

在使用 BTE 功能之前，使用者必須選擇指定的 BTE 操作模式。關於操作上的描述，請參考 Table 13-1，對於 ROP 的 BTE 操作，因應用於不同的應用，因此支援 16 種光柵操作 (ROP)，這樣對於來源端與目的端可以提供多樣的 ROP 組合。經由組合 BTE 功能的光柵操作，使用者可以達到不同的應用。請參考後續章節的描述。

使用者可以使用檢查 BTE 忙碌訊號與硬體中斷來確認 BTE 執行狀況。如果使用者要讀取 BTE 狀態可以由 BTE_CTRL0 (REG[90h]) Bit4 或是狀態暫存器 (STSR) Bit3 得到。另一種方法，使用者可以檢查硬體中斷，當有硬體中斷 INT#產生時去中斷旗標暫存器 REG[0Ch] 確認中斷來源，而硬體線路上 INT# 必須要連接 MPU。

Table 13-1 : BTE Operation Function

BTE Operation REG[91h] Bits [3:0]	BTE Operation
0000b	MPU Write with ROP.
0010b	Memory Copy (move) with ROP.
0100b	MPU Write with chroma keying (w/o ROP)
0101b	Memory Copy (move) with chroma keying (w/o ROP)
0110b	Pattern Fill with ROP
0111b	Pattern Fill with chroma keying (w/o ROP)
1000b	MPU Write with Color Expansion (w/o ROP)
1001b	MPU Write with Color Expansion and chroma keying (w/o ROP)
1010b	Memory Copy with opacity (w/o ROP)
1011b	MPU Write with opacity (w/o ROP)
1100b	Solid Fill (w/o ROP)
1110b	Memory Copy with Color Expansion (w/o ROP)
1111b	Memory Copy with Color Expansion and chroma keying (w/o ROP)
Other combinations	Reserved

Table 13-2 : ROP Function

ROP Bits REG[91h] Bit[7:4]	Boolean Function Operation
0000b	0 (Blackness)
0001b	~S0・~S1 or ~ (S0+S1)
0010b	~S0・S1
0011b	~S0
0100b	S0・~S1
0101b	~S1
0110b	S0^S1
0111b	~S0+~S1 or ~ (S0・S1)
1000b	S0・S1
1001b	~ (S0^S1)
1010b	S1
1011b	~S0+S1
1100b	S0
1101b	S0+~S1
1110b	S0+S1
1111b	1 (Whiteness)

註:

1. 在 ROP 功能，S0: 來源 0 的資料，S1: 來源 1 的資料，D: 目的端的資料。
2. 對於圖案填充功能而言，來源的資料就是圖案的資料。

例:

如果 ROP 功能設定為 Ch，那麼目的端資料 D=來源 0 的資料 (D=S0)

如果 ROP 功能設定為 Eh，那麼目的端資料 D=S0 + S1

如果 ROP 功能設定為 2h，那麼目的端資料 D= ~S0・S1

如果 ROP 功能設定為 Ah，那麼目的端資料 D= 來源 1 的資料(D=S1)

Table 13-3 : Color Expansion Function

ROP Bits REG[91h] Bit[7:4]	Start Bit Position for Color Expansion BTE operation code = 1000/1001/1110/1111	
	16-bit MPU Interface	8-bit MPU Interface
0000b	Bit0	Bit0
0001b	Bit1	Bit1
0010b	Bit2	Bit2
0011b	Bit3	Bit3
0100b	Bit4	Bit4
0101b	Bit5	Bit5
0110b	Bit6	Bit6
0111b	Bit7	Bit7
1000b	Bit8	Invalid
1001b	Bit9	Invalid
1010b	Bit10	Invalid

ROP Bits REG[91h] Bit[7:4]	Start Bit Position for Color Expansion BTE operation code = 1000/1001/1110/1111	
	16-bit MPU Interface	8-bit MPU Interface
1011b	Bit11	Invalid
1100b	Bit12	Invalid
1101b	Bit13	Invalid
1110b	Bit14	Invalid
1111b	Bit15	Invalid

13.1 選擇 BTE 起始位置與層

ROP S0/S1/D 可以被設定成任意的記憶體位址，再處理 ROP 功能前，必須先指定要處理的水平與垂直起始位置。

1. S0 的位址暫存器是 REG [93h], REG[94h], REG[95h], REG[96h], REG[97h], REG[98h], REG[99h], REG[9Ah], REG[9Bh], REG[9Ch]
2. S1 的位址暫存器是[9Dh], REG[9Eh], REG[9Fh], REG[A0h], REG[A1h] , REG[A2h], REG[A3h], REG[A4h], REG[A5h], REG[A6h]
3. D 的位址暫存器是 REG [A7h], REG[A8h], REG[A9h], REG[AAh], REG [ABh], REG[ACh], REG[ADh], REG[AEh], REG[AFh], REG[B0h]

13.2 色彩調配色盤記憶體 (Color Palette RAM)

RA8889 具有彩色調色盤記憶體，主要是提供給 8 位元的透明混合 (alpha blend) 功能。經由索引色彩調色盤記憶體可以得到真實色彩 (real color) (參考 Figure 13-1)，而此功能的方塊圖請參考 Figure 13-2。使用者在初始化設定色彩調色盤記憶體上可以參考 Figure 13-3 流程圖。

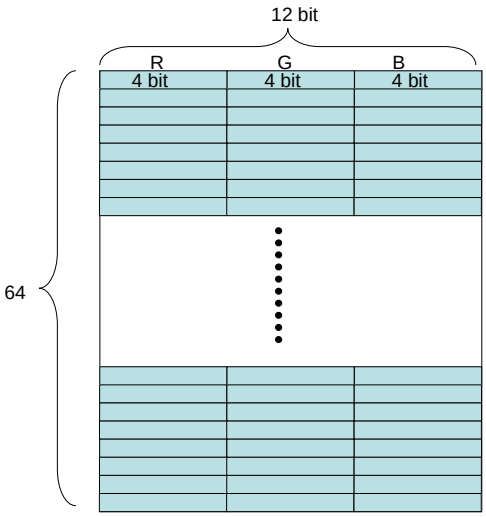


Figure 13-1 : Palette Ram Diagram

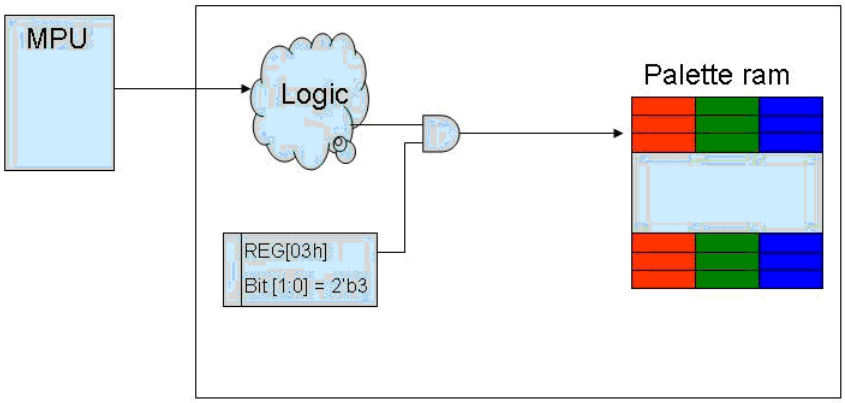


Figure 13-2 : Palette Ram Initial Data Path

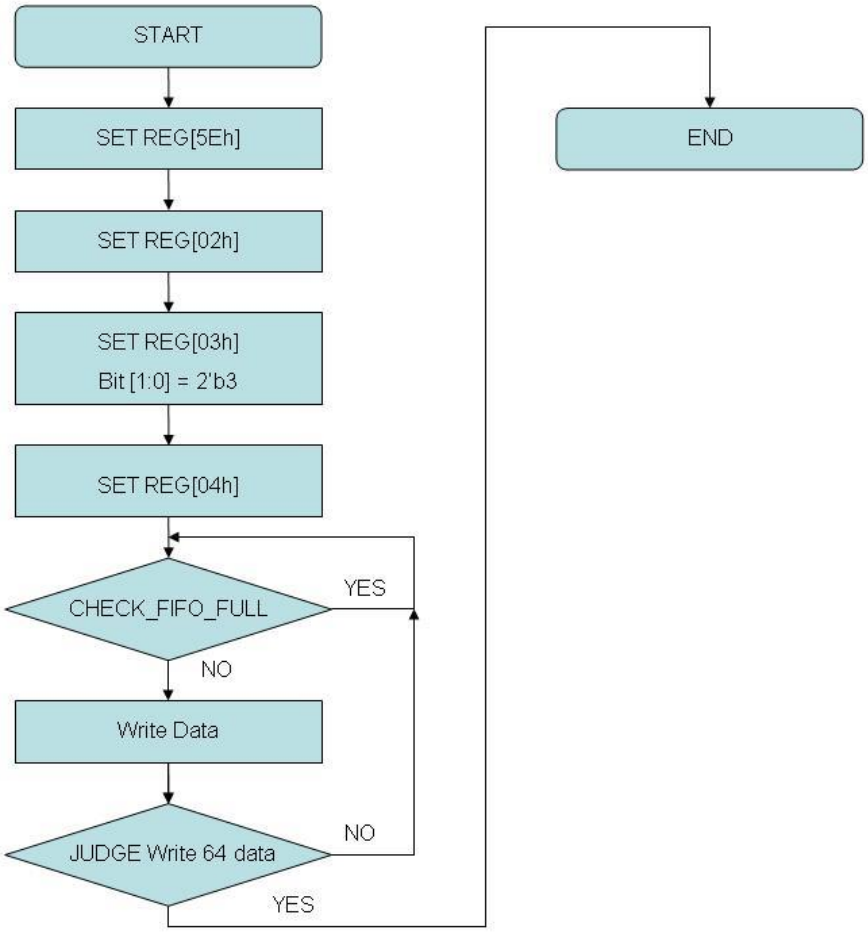


Figure 13-3 : Palette Ram Initial Flow

13.3 BTE 操作

13.3.1 結合光柵操作的 MPU 寫入

這個 MPU 寫入資料致記憶體中的功能可以結合光柵 (ROP) 的操作，BTE 提供 16 種 ROP 的操作，當透過 BTE 引擎做寫入資料至目的記憶體時，會自動處理光柵運算。

13.3.2 結合光柵操作的記憶體複製

記憶體搬移複製的功能可以結合 16 種光柵 (ROP) 操作，而其只支援正向處理資料。

13.3.3 矩形填滿

矩形填滿是 BTE 針對指定的目的記憶體進行前景色填滿。

13.3.4 圖樣填滿

這個操作是在指定的 BTE 區域填上 8X8/16X16 像素的圖樣 (pattern)。

13.3.5 結合 Chroma Key 的圖樣填滿

這個操作是在指定的 BTE 區域填上 8X8/16X16 像素的圖樣。但是若是圖樣 (pattern) 的顏色等於所設定的關鍵色 (key color)，那麼底圖的資料將不會被更新，而關鍵色被設定在 BTE 背景色暫存器，這個功能沒有光柵 (ROP) 操作。

13.3.6 結合 Chomra Key 的 MPU 寫入

這個操作支援傳輸資料由主控端到 SDRAM 區域，當由主控端的來源 0 (S0) 資料顏色等於關鍵色 (key color)，那麼目的端的記憶體資料並不會被更改，而關鍵色 (Key color) 被設定在 BTE 背景色暫存器。本功能不支援光柵 (ROP) 操作。

13.3.7 結合 Chroma Key 的記憶體複製

這個資料的傳輸方向僅支援正向傳輸，而來源與目的資料是在 SDRAM 上不同的區域。當來源資料 0 (S0) 等於關鍵色 (key color)，則目的端記憶體資料不會被更新，而關鍵色定義在 BTE 背景色暫存器。本功能不支援光柵 (ROP)操作。

13.3.8 擴展色彩

這個操作是將主控端輸入的單色資料擴展為 8/16/24 bpp 彩色資料格式。

來源資料如果為“1”，則 BTE 將會轉為前景色，前景色的設定在前景色暫存器中。

來源資料如果為“0”，則 BTE 將會轉成背景色，背景色的設定在背景色暫存器中。

如果背景透明被致能，當來源資料為”0”時，那麼目的記憶體上的的顏色將不會被改變。

註: 無論是否致能背景透明功能，前景(D2h~D4h)與背景(D5h~D7h) 暫存器不可設相同的值。

13.3.9 結合擴展色彩的記憶體複製

這個功能是將記憶體中的單色資料轉變成 8/16/24 bpp 彩色資料。來源資料如果是“1”則會轉為前景色並寫入記憶體中，前景色的設定在前景色暫存器中。來源資料如果是“0”則會轉為背景色並寫入記憶體中，背景色的設定在背景色暫存器中。如果背景透明被致能，那麼當來源資料是”0”時，目的記憶體資料不會有任何更改。

註: 無論是否致能背景透明功能，前景(D2h~D4h)與背景暫存器 (D5h~D7h) 不可設相同的值。

13.3.10 結合透明度的記憶體複製

這個操作是處理來源 0 (S0) 與來源 1 (S1) 資料並且將其混合後寫入目的記憶體。這個透明度處理具有兩個模式可供使用- Picture 模式與 Pixel 模式。

Picture 模式是指說 BTE 處理區域都是具有相同的 alpha 透明參數值，這個值透過暫存器讀取可得到。

Pixel 模式是指說 BTE 處理區域內每個像素具有不同的 alpha 透明參數值，這個透明的參數值紀錄在每個像素本身的高位元中。

來源 0(S0) : SDRAM
來源 1(S1) : SDRAM
目的(D) : SDRAM

13.3.11 結合透明度 MPU 的寫入

這個操作是處理來源 0 (S0) 與來源 1 (S1) 資料並且將其混合後寫入目的記憶體。這個 Alpha Blending 具有兩個模式可供使用- Picture 與 Pixel 模式。

Picture 模式是指說 BTE 處理區域都是具有相同的 alpha 透明參數值，這個值透過暫存器讀取可得到。

Pixel 模式是指說 BTE 處理區域內每個像素具有不同的 alpha 透明參數值，這個透明的參數值紀錄在每個像素本身的高位元中。

來源 0(S0) : MPU
來源 1(S1) : SDRAM
目的(D) : SDRAM

13.4 BTE 存取記憶體方法

在設定後，BTE 可以使用區塊的方法對來源與目的端的記憶體作存取。下面的圖檔就是說明存取的方法:

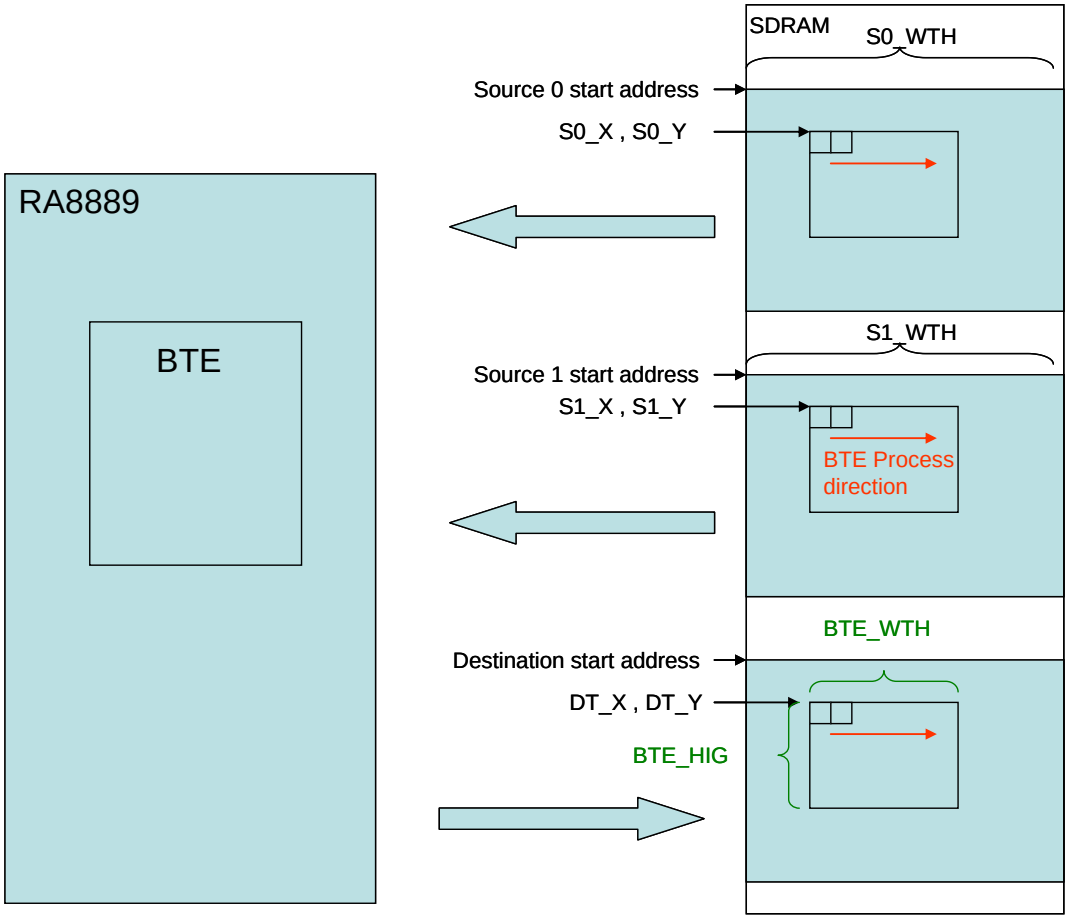


Figure 13-4 : Memory Access of BTE Function

13.5 BTE 透明關鍵色 (Chorma Key) 比較

在 BTE 中透明的關鍵色 (Chroma Key) 如果被致能的話，BTE 將會比較來源 0 (S0) 與背景色暫存器的值。 如果兩者資料相同那麼就不會修改目的端記憶體的資料，以達成透明的效果。

在來源色深為 256 色時，
Source 0 red 比較 REG[D5h] Bit [7:5],
Source 0 green 比較 REG [D6h] Bit [7:5],
Source 0 blue 比較 REG [D7h] Bit [7:6]

在來源色深為 65k 色時，
Source 0 red 比較 REG [D5h] Bit [7:3],
Source 0 green 比較 REG [D6h] Bit [7:2],
Source 0 blue 比較 REG [D7h] Bit [7:3]

在來源色深為 16.7M 色時，
Source 0 red 比較 REG[D5h] Bit [7:0],
Source 0 green 比較 REG [D6h] Bit [7:0],
Source 0 blue 比較 REG [D7h] Bit [7:0]

13.6 BTE 功能詳述

13.6.1 結合光柵操作的 MPU 寫入

此功能可以增加 MPU 寫入 SDRAM 的速度，寫入的資料可以結合光柵 (ROP) 操作填入目的記憶體中。BTE 本身提供 16 種 ROP，由下圖可以得知來源 0 (S0) 必須由 MCU (MPU) 提供。

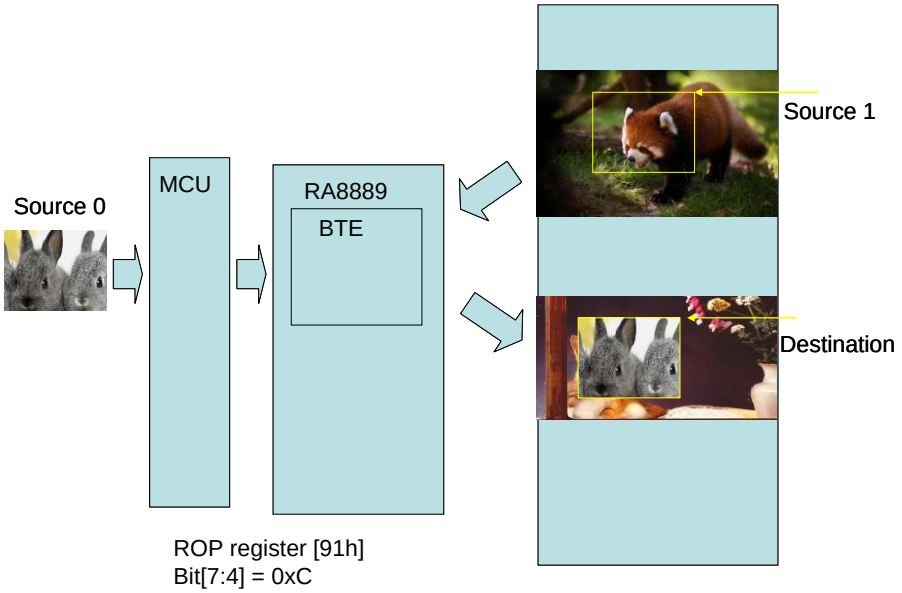


Figure 13-5 : Hardware Data Flow

完成這個功能的程式流程圖如下:

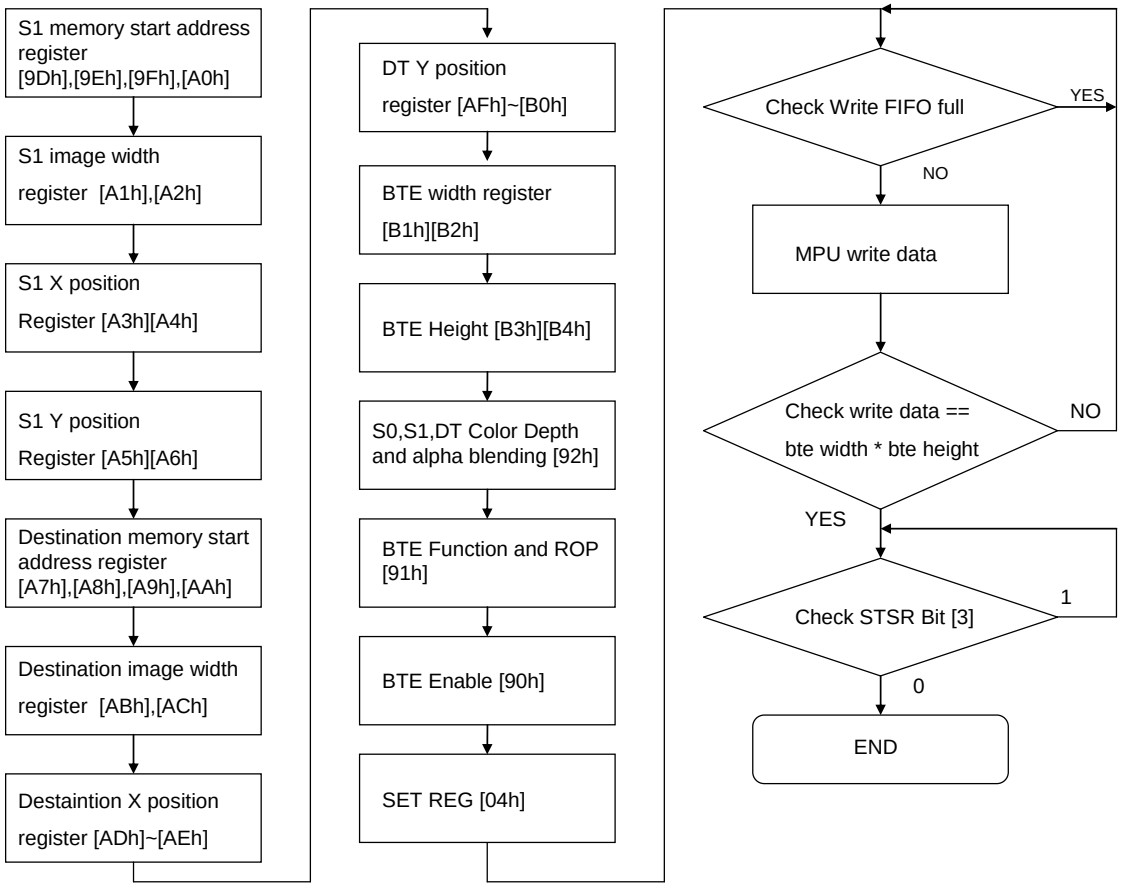


Figure 13-6 : Flow Chart

13.6.2 結合光柵操作的記憶體複製

這個功能將會從指定的記憶體來源區域複製搬移至指定的記憶體目的區域。這個操作可以減少 MPU 處理時間，進而提升記憶體資料複製搬移的速度。

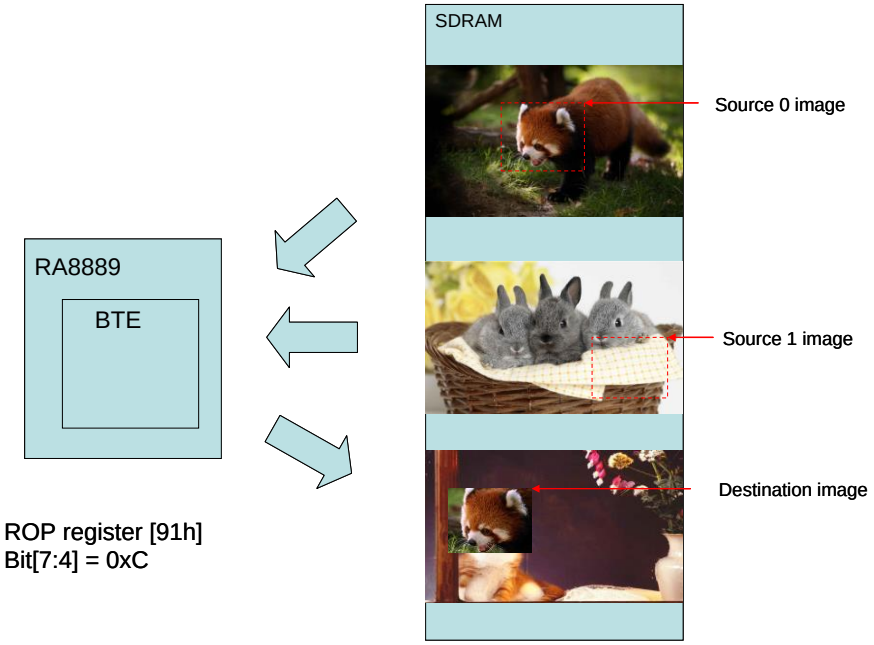


Figure 13-7 : Hardware Data Flow

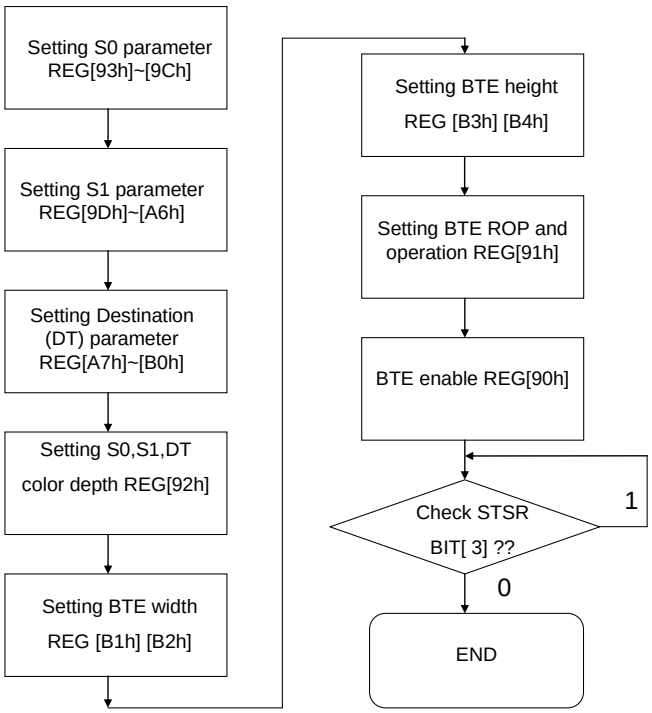


Figure 13-8 : Flow Chart

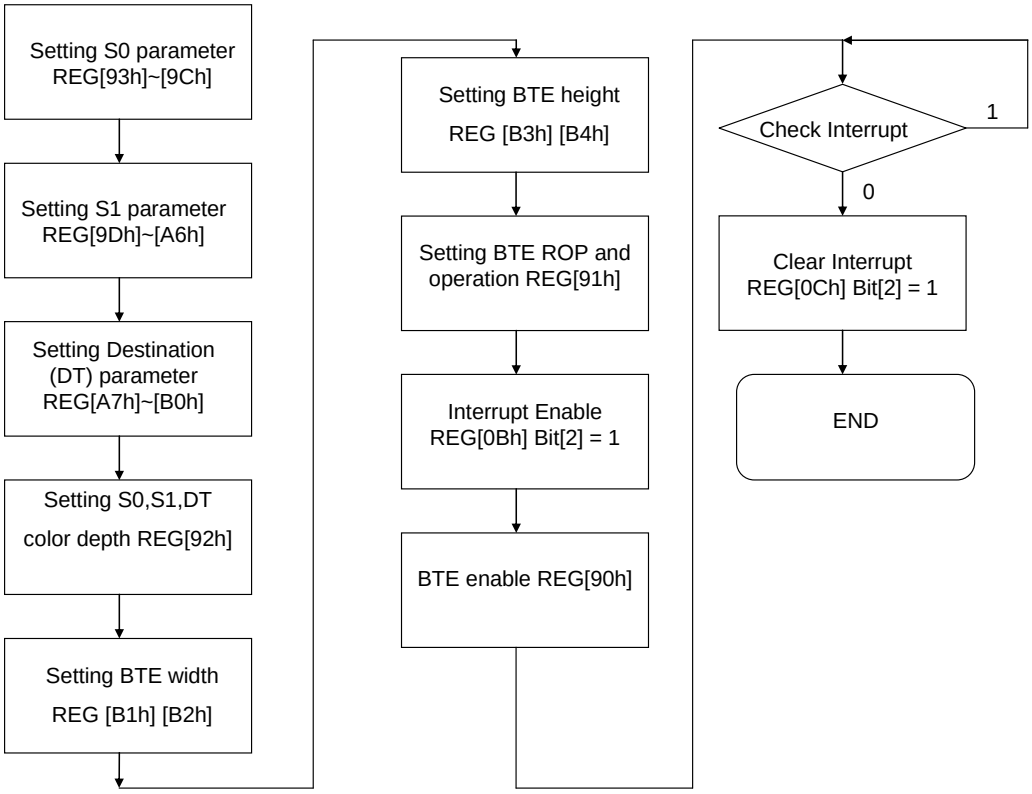


Figure 13-9 : Flow Chart – Check Int

13.6.3 結合 Chroma Key 的 MPU 寫入

此功能為 MPU 具有關鍵色的寫入資料功能。此功能可以提升 MPU 寫入 SDRAM 的速度。一旦這個功能被致能後，BTE 引擎會維持忙碌狀態直到所有資料被寫入為止。

與" BTE 寫入"功能不同的是"結合 Chroma Key 的 MPU 寫入"功能在處理資料時，如果 MPU 寫入資料與關鍵色 (Chroma key) 相同，則寫入 S0 資料會忽略掉。而關鍵色是被設定在 "BTE background Color" 暫存器中。舉例說明如果來源端是紅色背景上有一黃色的圓，經由選擇紅色為透明色的話，那麼透過此功能寫出來的圖就是一個黃色的圓，紅色則不會被寫入記憶體中。此功能的程式流程圖如下:

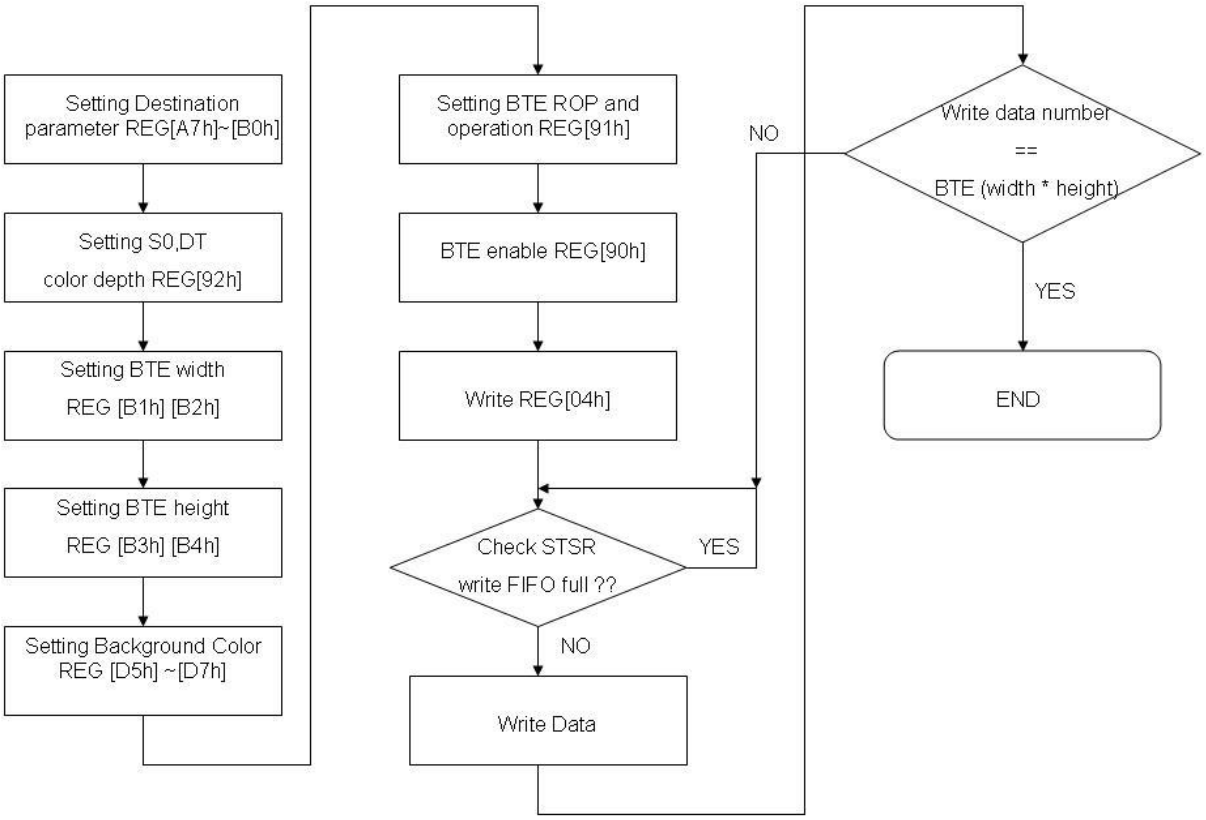


Figure 13-10 : Flow Chart

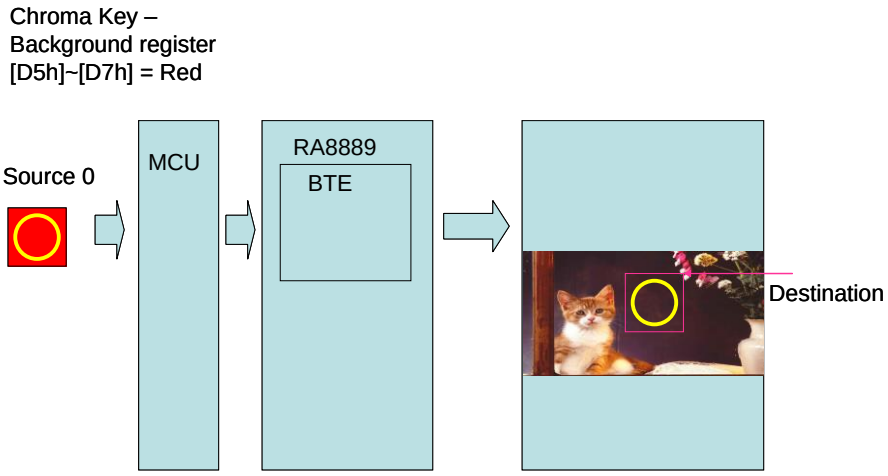


Figure 13-11 : Hardware Data Flow

13.6.4 結合 Chroma Key 的記憶體複製

此功能可以複製搬移一指定的記憶體來源區域到記憶體目的區域，並且在複製搬移的過程中會比較來源端資料與 (Chroma Key) 的顏色，當兩者相同時，不去更改記憶體目的端的資料，表現出來就是與關鍵色相同的會被透明處理。而關鍵色的設定在 “BTE background color” 暫存器中。來源端與目的端皆是記憶體為來源。此功能的程式流程圖如下:

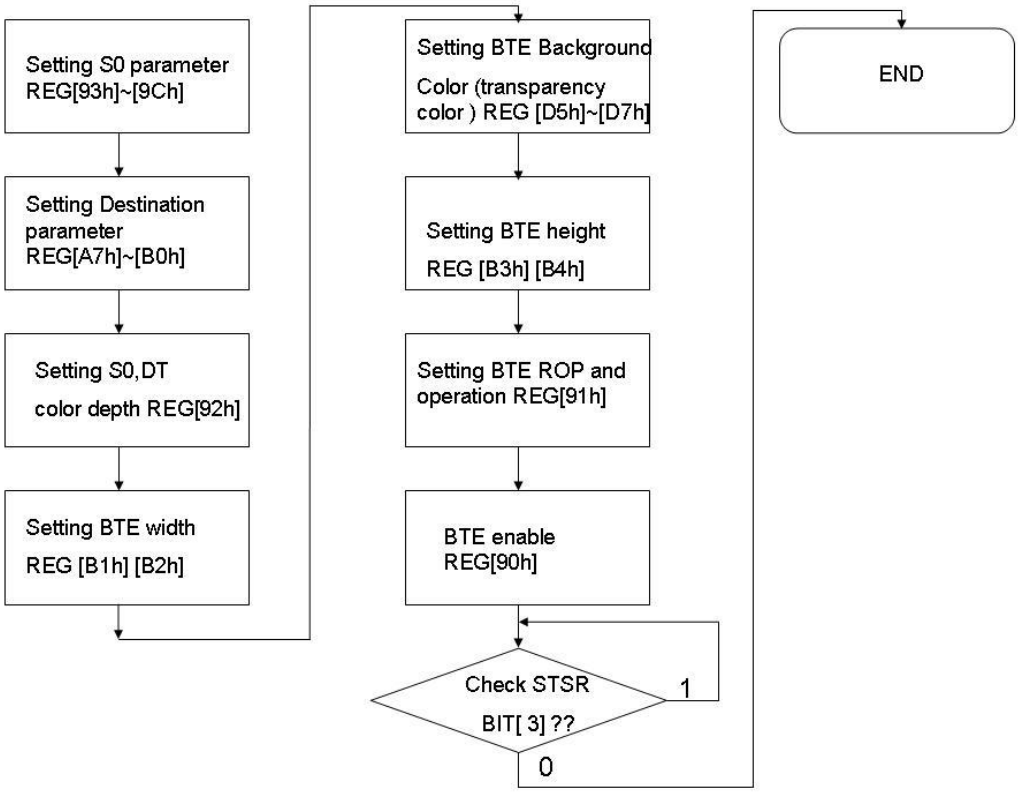


Figure 13-12 : Flow Chart

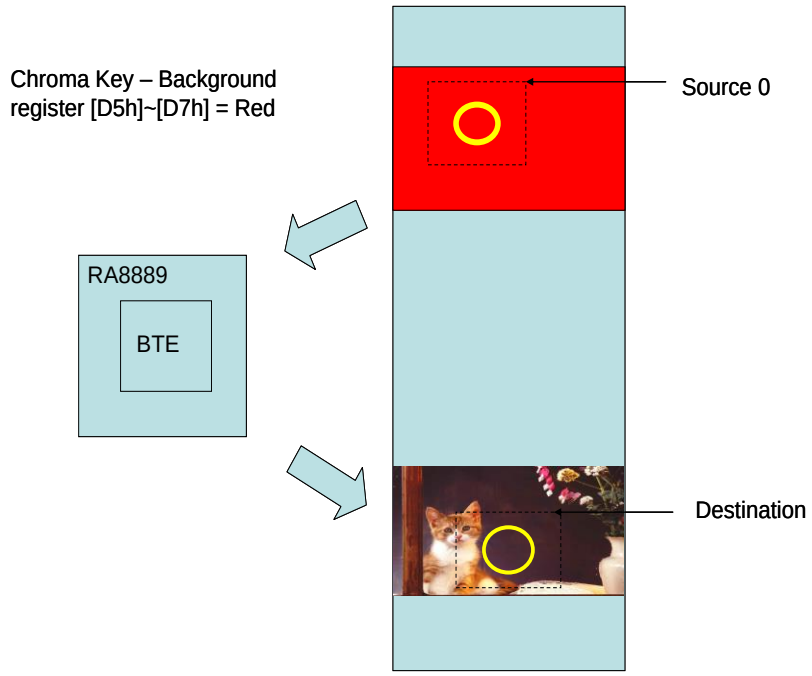


Figure 13-13 : Hardware Data Flow

13.6.5 結合光柵操作的圖樣填滿

功能將一指定區域重覆填滿指定的 8X8/16X16 圖案，而 8x8/16x16 像素的圖案是使用此功能前已經預先儲存在記憶體中。這個功能也可以結合 16 種光柵 (ROP) 操作。這個功能可以在一指定的區域加速複製圖樣。如快速背景張貼上。

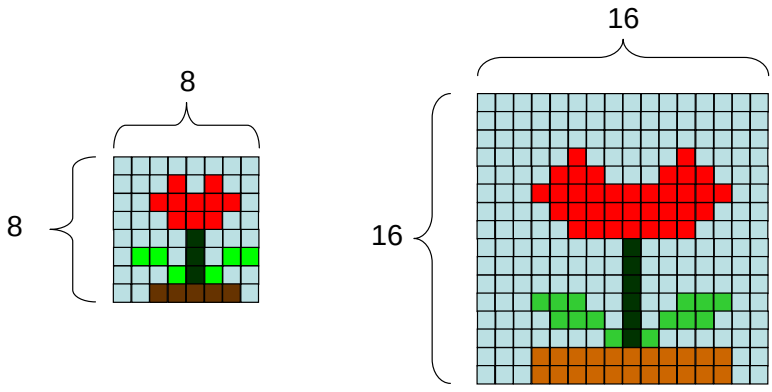


Figure 13-14 : Pattern Format

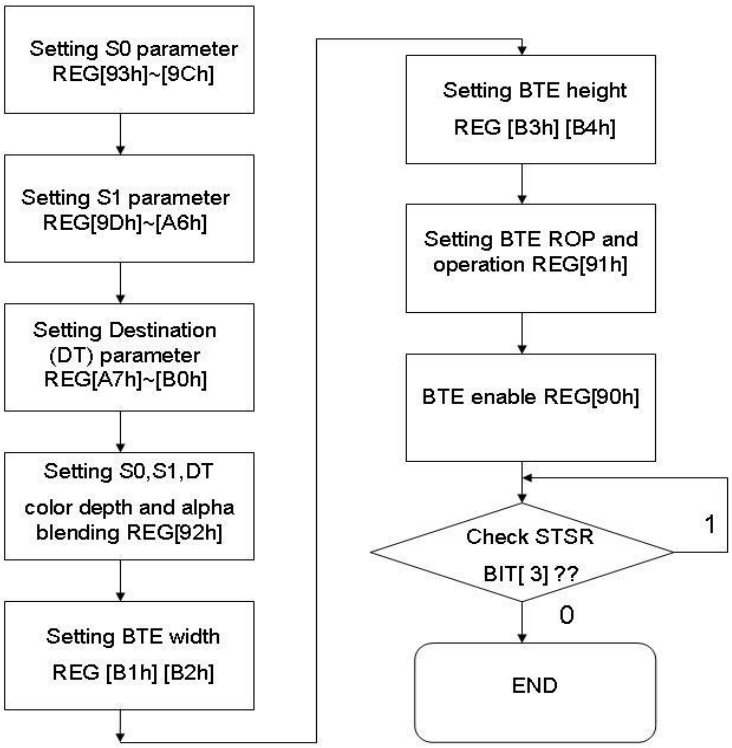


Figure 13-15 : Flow Chart

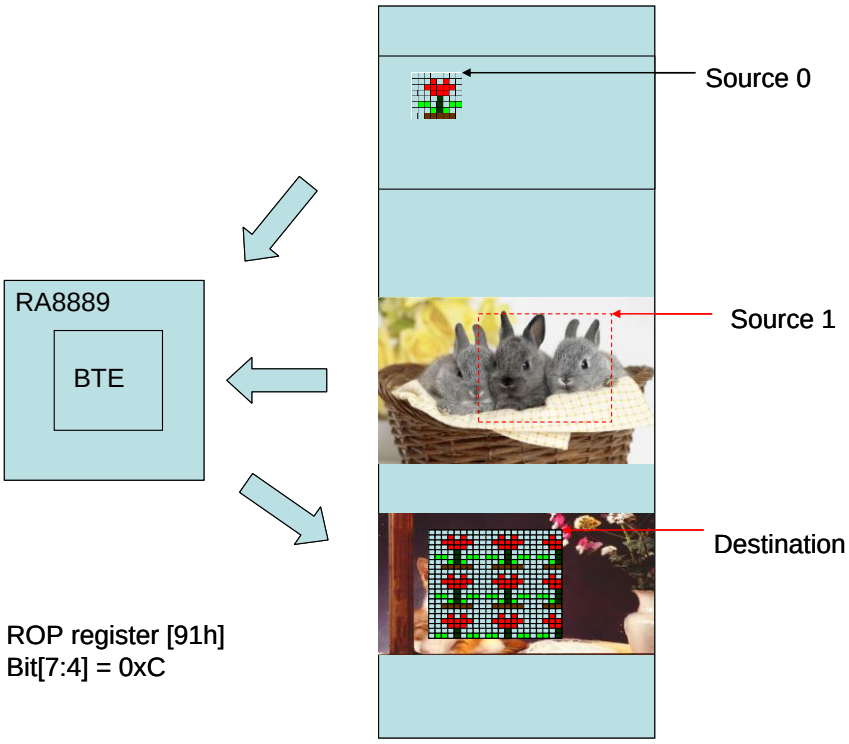


Figure 13-16 : Hardware Data Flow

13.6.6 結合 Chroma Key 的圖樣填滿

此功能將一指定記憶體區域重覆填滿指定的 8X8/16X16 圖案，但是在處理的過程中，如果來源端顏色與關鍵色 (Chroma key) 相同，那麼對於目的端就不做寫入，因此看到的效果將會是透明的。而關鍵色 (Chroma key) 被設定 REG[D5h]~[D7h] 暫存器中。

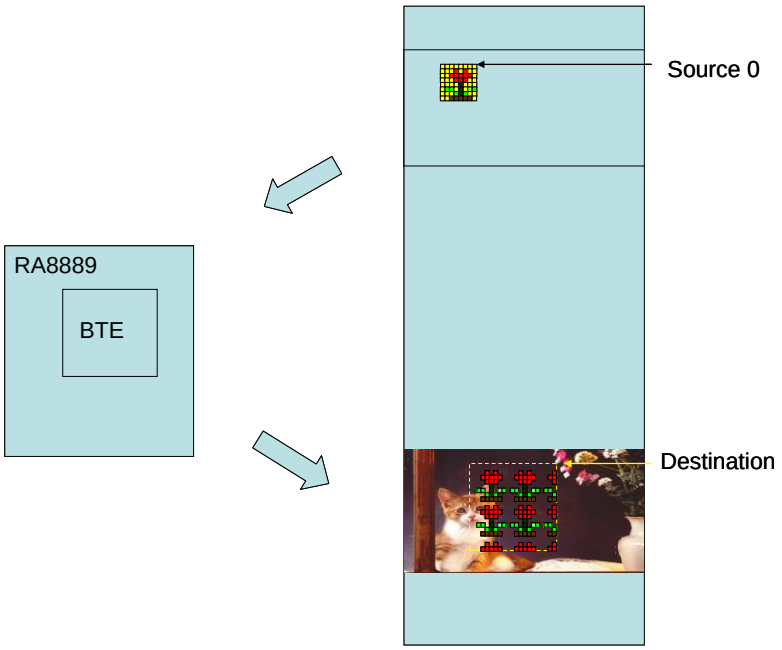


Figure 13-17 : Hardware Flow

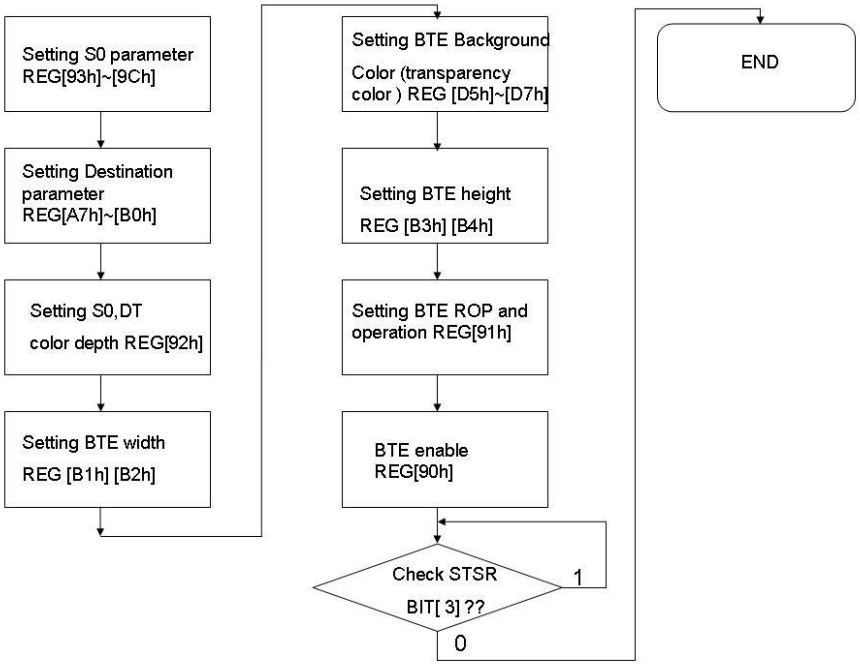


Figure 13-18 : Flow Chart

13.6.7 結合擴展色彩的 MPU 寫入

此功能為 MPU 將單色資料寫入記憶體中，在這個操作中來源圖檔是單色 (monochrome) 的資料，經過 BTE 功能可以轉成彩色的圖檔資料。如果單色圖檔的 bit 為 ”1” 則轉為前景色，如果單色圖檔的 bit 為 ”0” 則轉成背景色。這個功能讓使用者方便由單色系統轉成彩色系統。單色圖在 BTE 內部是每個掃描線分開處理的，當一條掃描線處理完時，沒有被處理的單色掃描線資料就被捨棄。下一行的資料則由下一筆數據包產生，每一筆寫目的記憶體的資料做顏色擴展時都是由 MSB 處理到 LSB。如果 MPU 介面被設定為 16bit 時，那麼 ROP 的起始位元可以被設為 15 到 0 的任一位元，MPU 介面被設定為 8bit，那麼 ROP 起始位元可以被設為 7 到 0 的任一位元。來源 0 顏色深度 REG [92h] Bit[7:6] 在此功能不被參考。

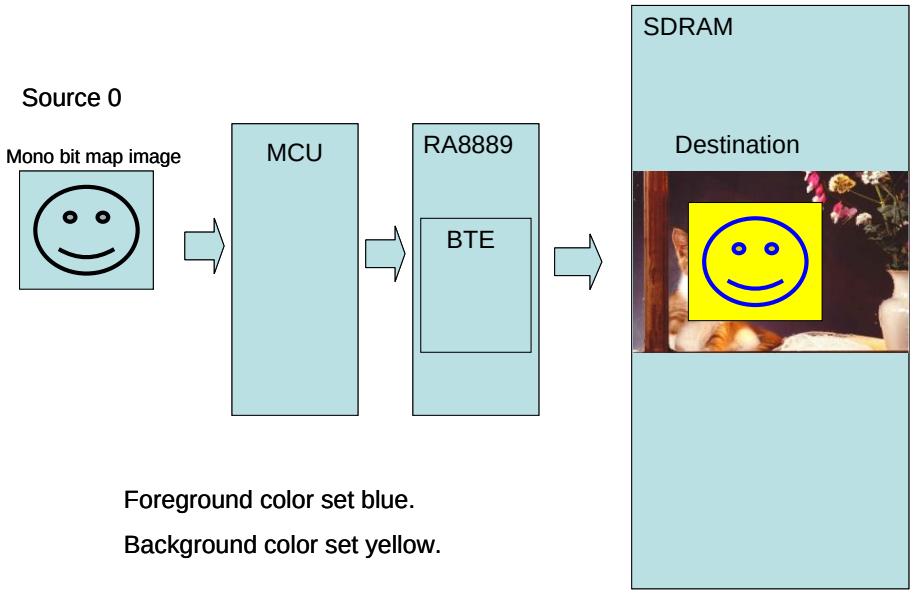


Figure 13-19 : Hardware Data Flow

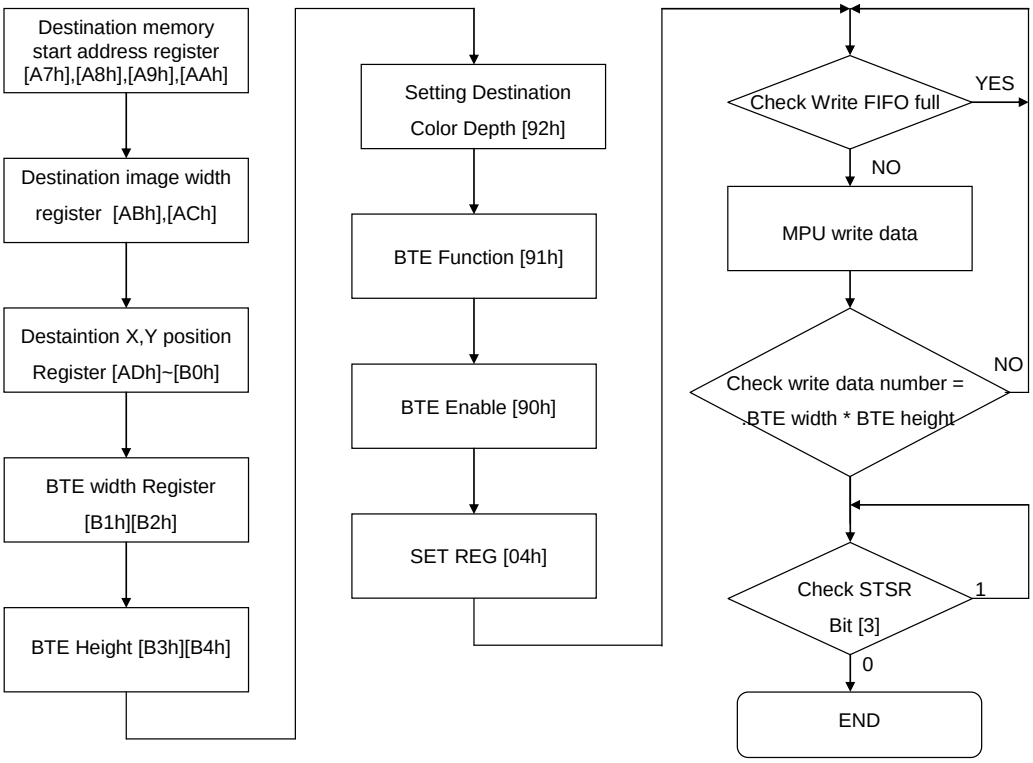


Figure 13-20 : Flow Chart

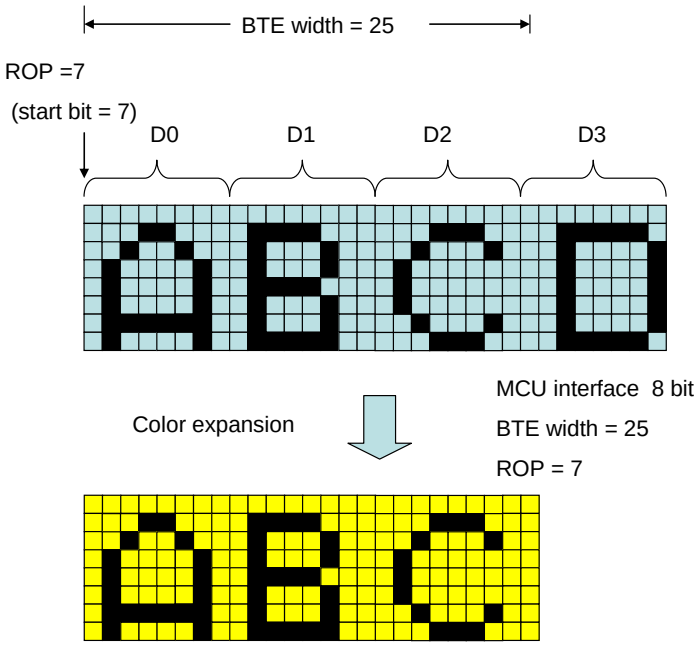


Figure 13-21 :Start Bit Example 1

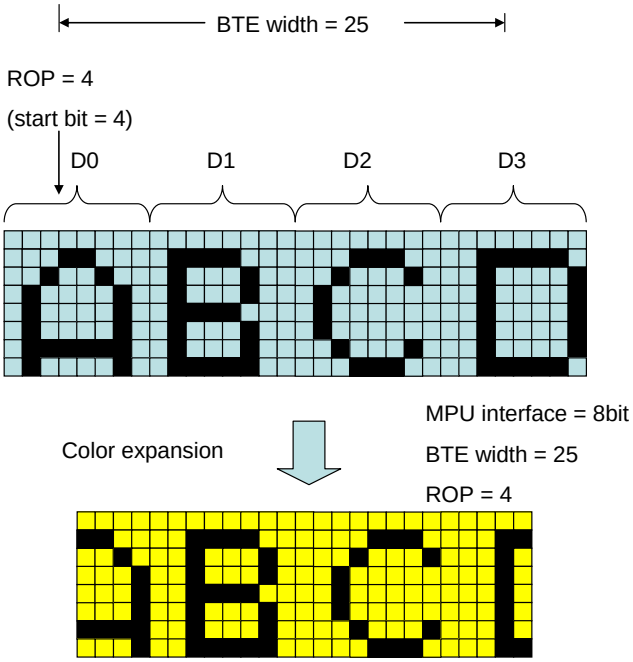


Figure 13-22 : Start bit Exapmle 2

註:

1. 計算每一列的資料數= ((BTE Width size REG – (MPU interface bits – (start bit + 1))) / MPU interface bits) + ((start bit + 1) / (MPU interface))
2. 總資料數= (sent data numbers per row) x BTE Vertical REG setting

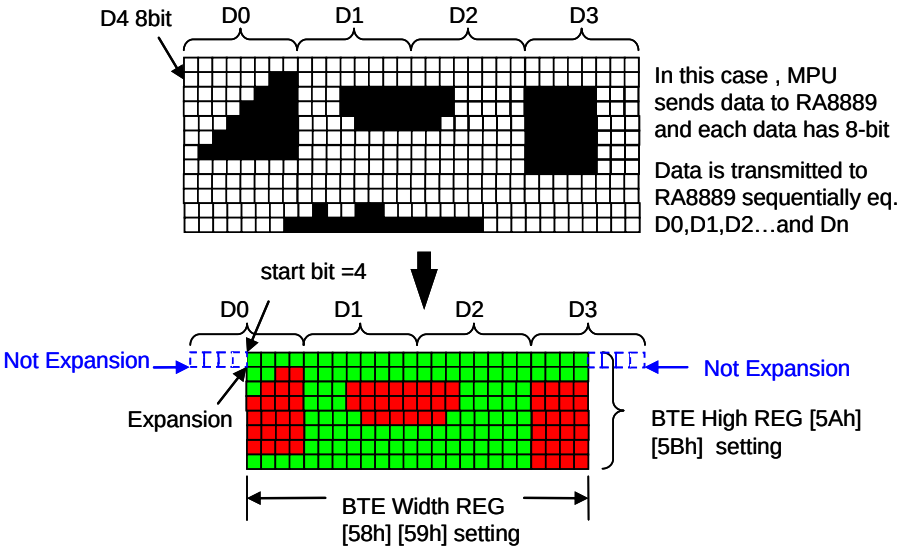


Figure 13-23 : Color Expansion Data Diagram

13.6.8 結合擴展色彩與 Chroma key 的 MPU 寫入

這個 BTE 操作實際上是等效於擴展色彩 BTE 的功能，除了來源端的背景色被設為可忽略的。在單色圖 bit 資料為"1"可轉為前景色，單色圖中 bit 資料為 "0" 不處理。

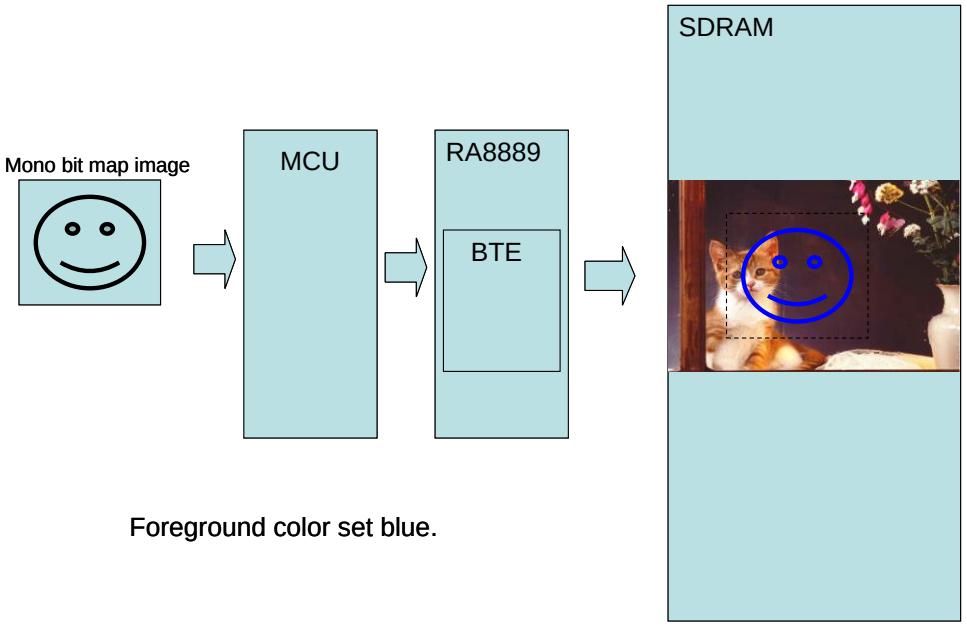


Figure 13-24 : Hardware Data Flow

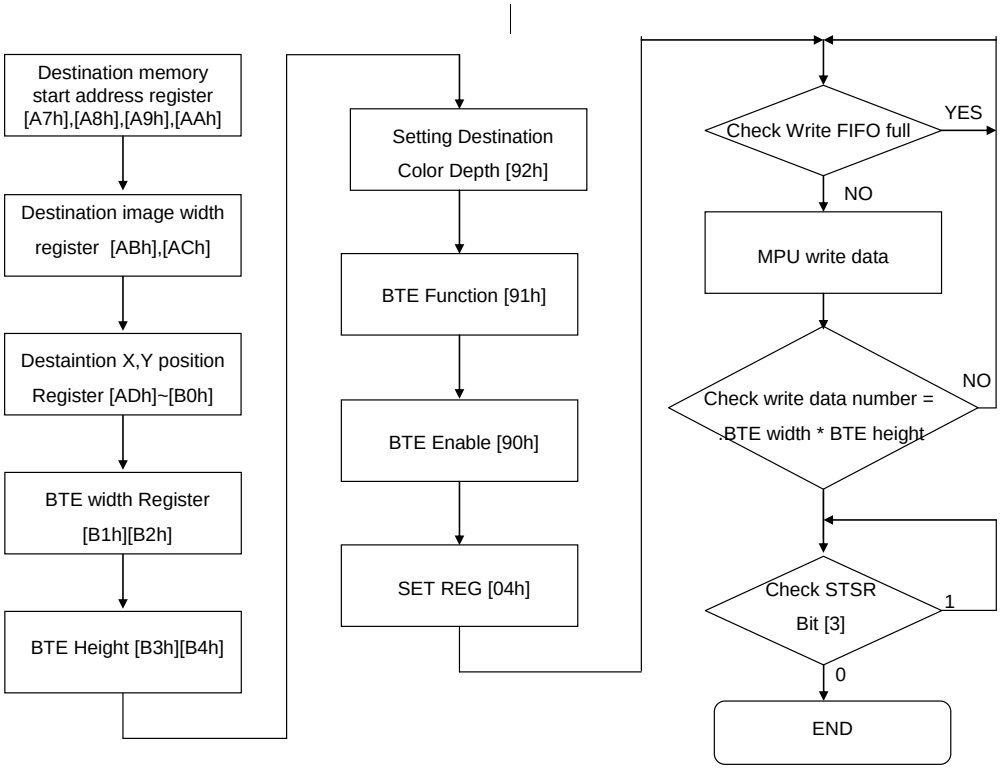


Figure 13-25 : Flow Chart

13.6.9 結合透明度的記憶體複製

“Memory Copy with opacity” 可以混合來源 0 資料與來源 1 資料然後再寫入目的記憶體。這個功能有兩個模式– **Picture 模式與 Pixel 模式**。Picture 模式可以被操作在 8 bpp/16bpp/32bpp 色深下並且對於全圖只具有一種混合透明度 (alpha level)，混合度被定義在 REG[B5h]。Pixel 模式只能被操作在來源 1 端是 8bpp/16bpp 模式，而各個 Pixel 具有其各自的混合度，在來源 1 為 16bpp 色深下像素的 bit [15:12] 是透明度 (alpha level)，剩餘的 bit 則為色彩資料；而來源 1 為 8bpp 色深情形下像素 bit [7:6] 是透明度 (alpha level)， Bit [5:0] 則是被使用在索引調色盤 (palette color) 的顏色。根據 32bpp 色深下的像素圖像，必須將 S1 顏色深度設置為 16bpp，並且必須將 S1 寬度設置為與原始圖像相同的寬度（寬度）。 在 32-bit 像素模式下，S1 圖像的 bit [31:24]代表其 alpha 值，bit [23：0]代表像素數據。Figure 13-31 顯示了有關如何通過 MPU 接口將αRGB 圖像數據寫入 SDRAM 的流程。

Picture mode - Destination data = (Source 0 * (1 - alpha Level)) + (Source 1 * alpha Level);
Pixel mode 16bpp - Destination data = (Source 0 * (1- alpha Level)) + (Source 1 [11:0] * alpha Level)
Pixel mode 8bpp – Destination data = (Source 0 * (1- alpha Level)) + (Index palette (Source 1[5:0]) * alpha Level)

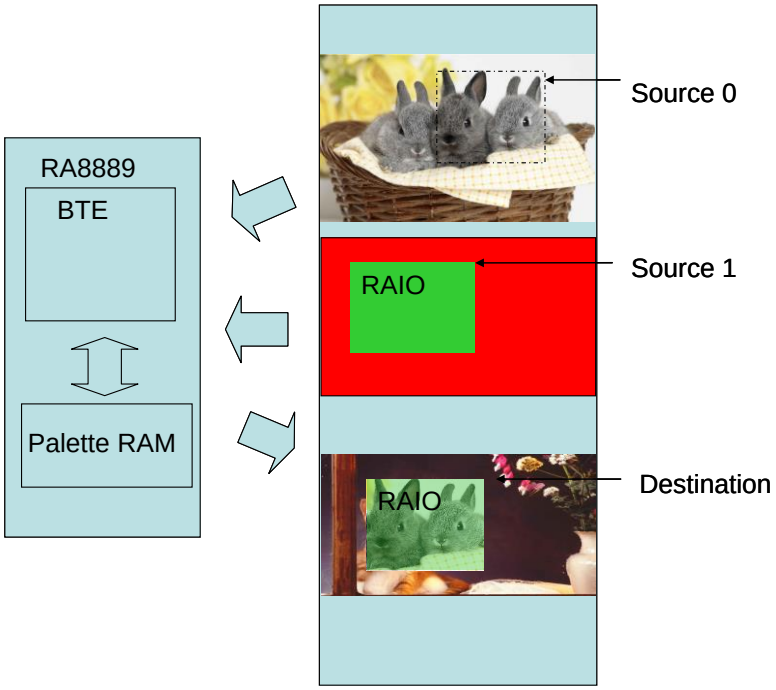


Figure 13-26 : 8bpp Pixel mode Hardware Data Flow

Table 13-4 : Alpha Blending Pixel Mode -- 8bpp

Bit [7:6]	Alpha Level
0h	0
1h	10/32
2h	21/32
3h	1

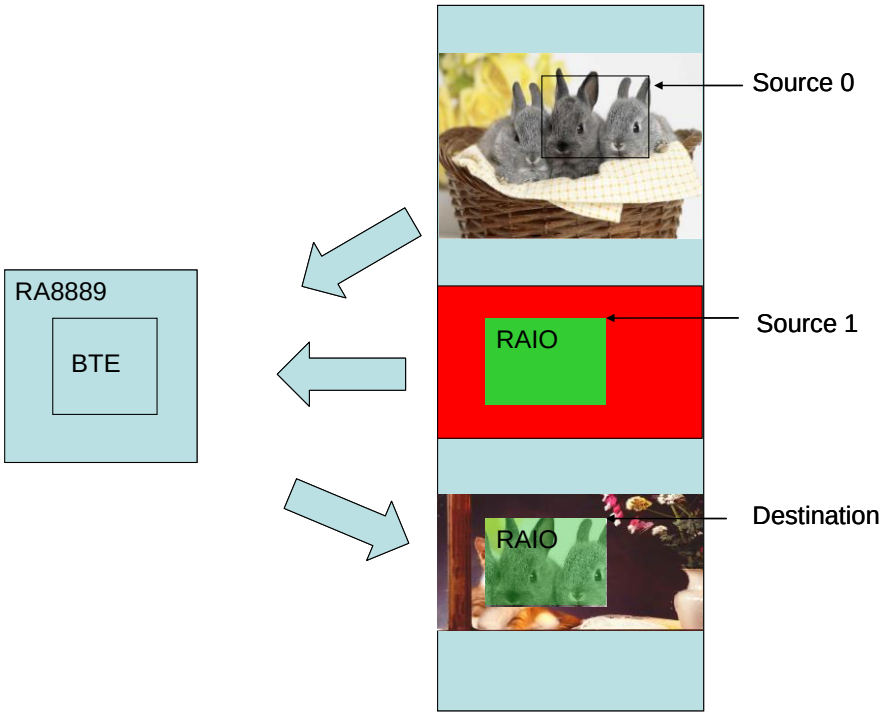


Figure 13-27 : 16bpp Pixel Mode Hardware Data Flow

Table 13-5 : Alpha Blending Pixel Mode -- 16bpp

Bit [15:12]	Alpha Level
0h	0
1h	2/32
2h	4/32
3h	6/32
4h	8/32
5h	10/32
6h	12/32
7h	14/32
8h	16/32
9h	18/32
Ah	20/32
Bh	22/32
Ch	24/32
Dh	26/32
Eh	28/32
Fh	1

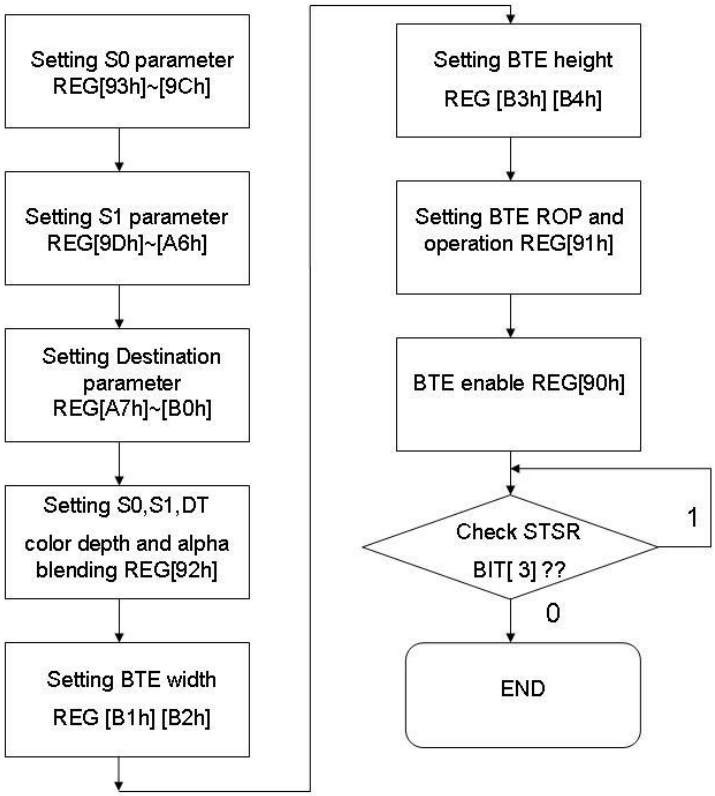


Figure 13-28 : Pixel Mode Flow Chart

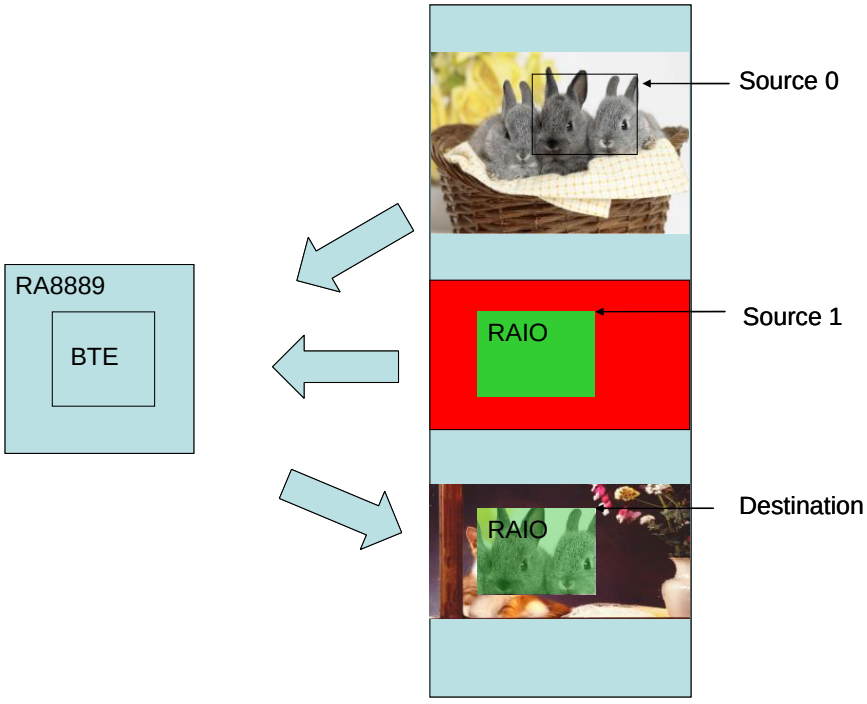


Figure 13-29 : Picture Mode Hardware Data Flow

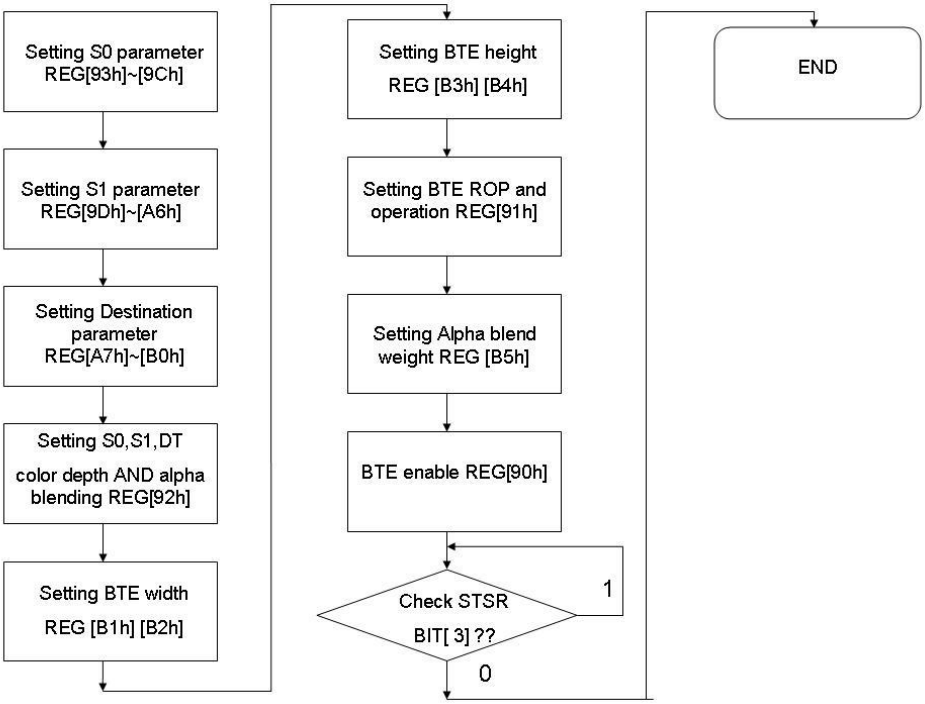


Figure 13-30 : Picture Mode Flow Chart

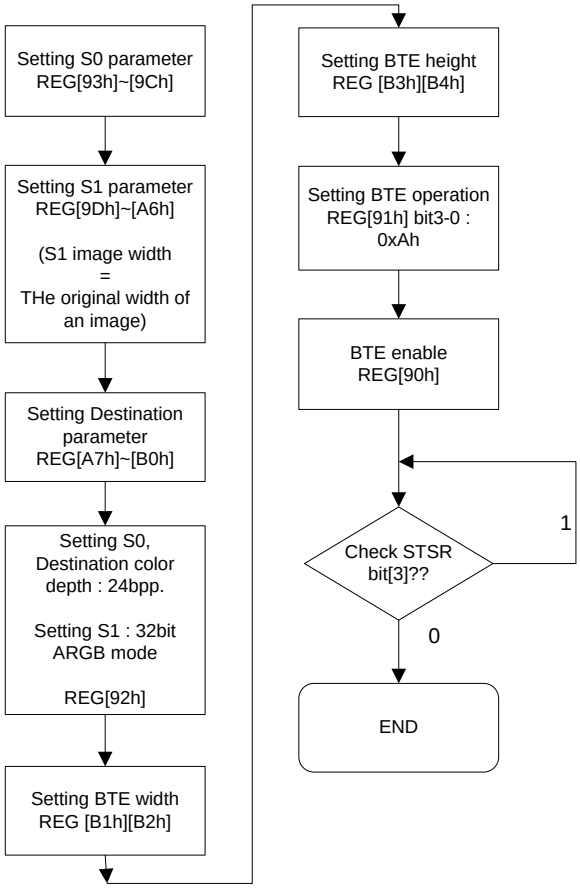


Figure 13-31

13.6.10 結合透明度的 MPU 寫入

"MPU Write with opacity" 功能混合了來源 0 與來源 1 的資料並寫入目的記憶體，而來源 0 的資料是從 MPU 來的 MPU (MCU)，來源 1 資料則由 SDRAM，其他有關於 Alpha blending 的模式 Picture 與 Pixel 與 "Memory Copy with opacity" 相同。

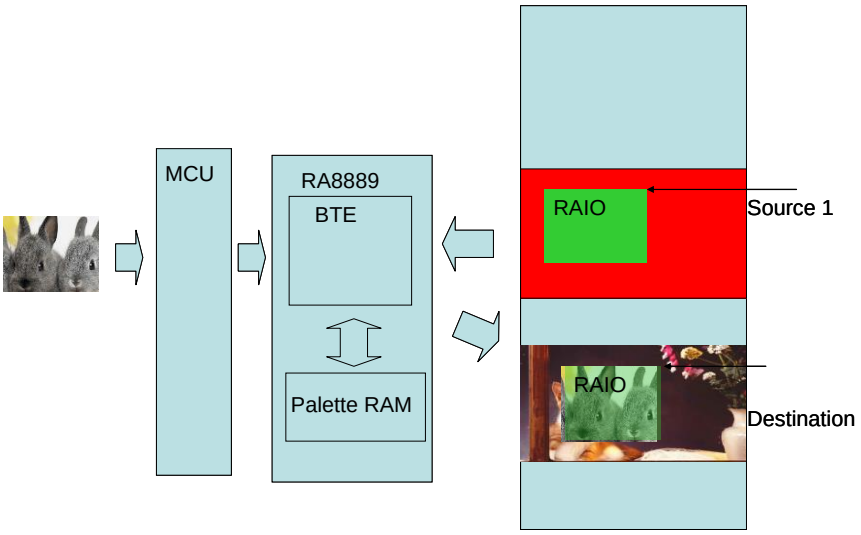


Figure 13-32 : Hardware Data Flow

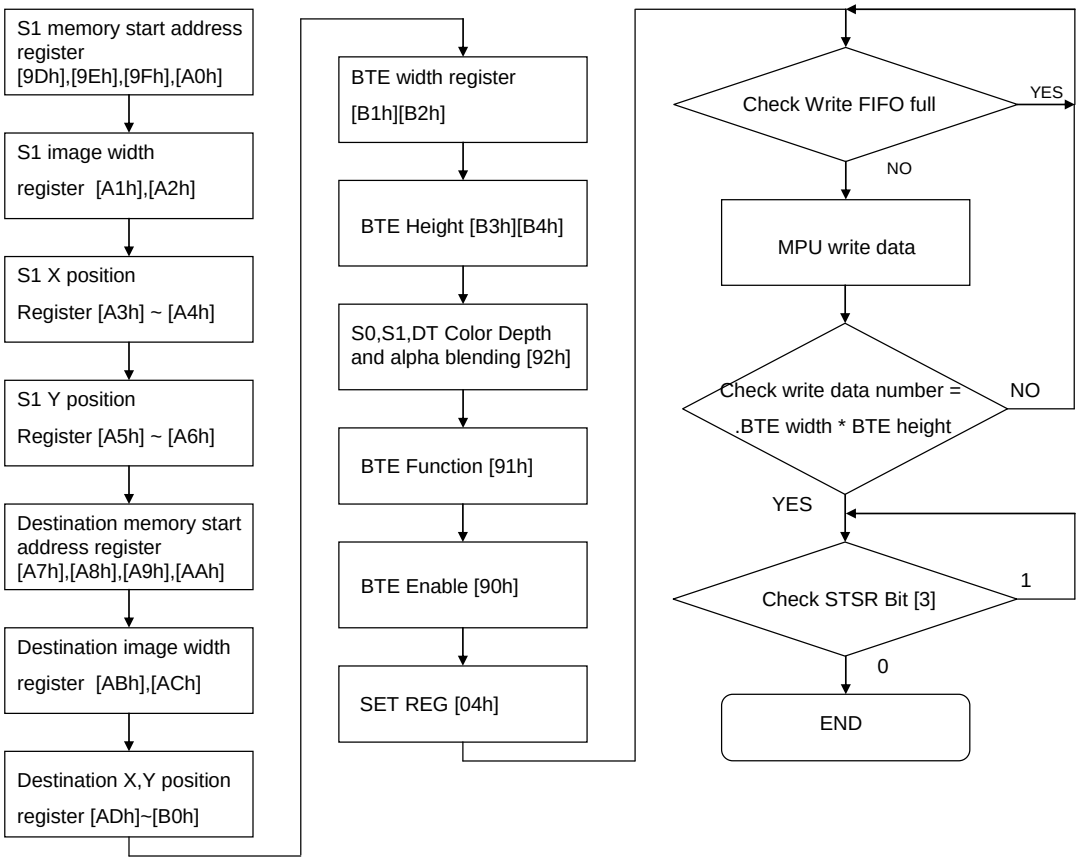


Figure 13-33 : Flow Chart

13.6.11 結合擴展色彩的記憶體複製

“Memory Copy w/ Color Expansion” 會將從 Buffer RAM 讀取的來源 0 (S0) 單色影像資料 (mono) 轉成彩色影像資料，並且寫入 SDRAM 目的記憶體中。如果單色資料 bit 為“1”，那麼將會轉換成前景色暫存器設定的顏色。如果單色資料 bit 為“0”，那麼將會轉換成背景色暫存器設定的顏色。單色資料寬度則是由 REG[92h] 來定義，來源 0 單色資料寬度可以定義為 8bit/16bit。如果單色資料寬度定義為 8bit，那麼 ROP (start bit) 可設定值可由 bit7~bit0 來當起始位元；如果單色資料寬度定義為 16bit，那麼 ROP (start bit) 可設定值可由 bit15~bit0 來當起始位元。

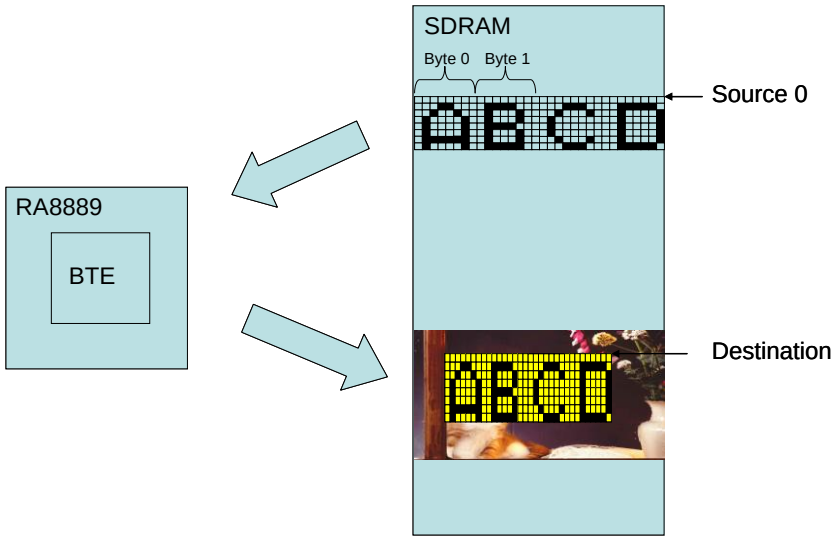


Figure 13-34 : Hardware Data Flow

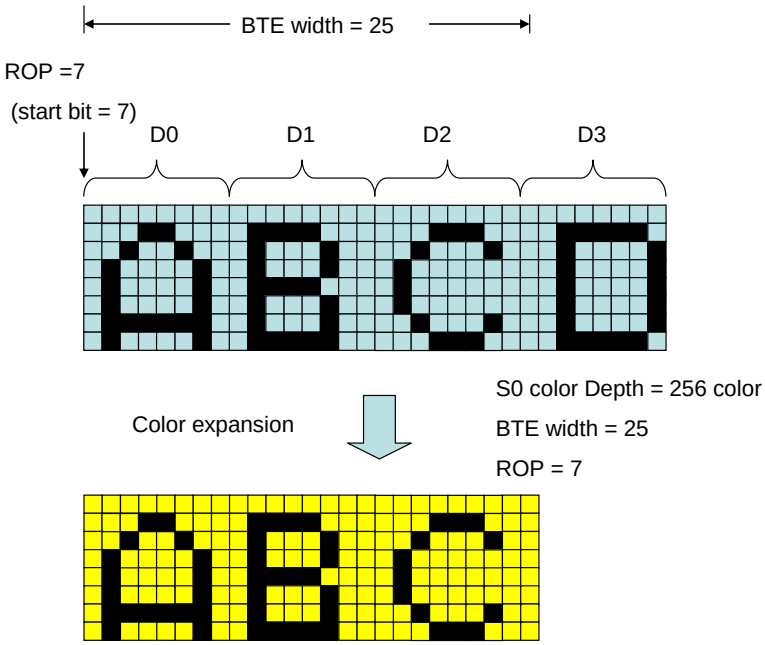


Figure 13-35 : Start Bit Example 1

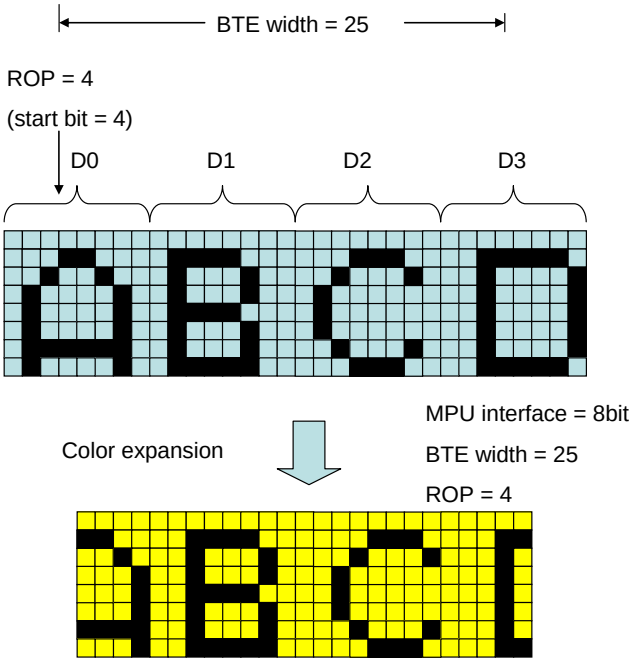


Figure 13-36 : Start Bit Example 2

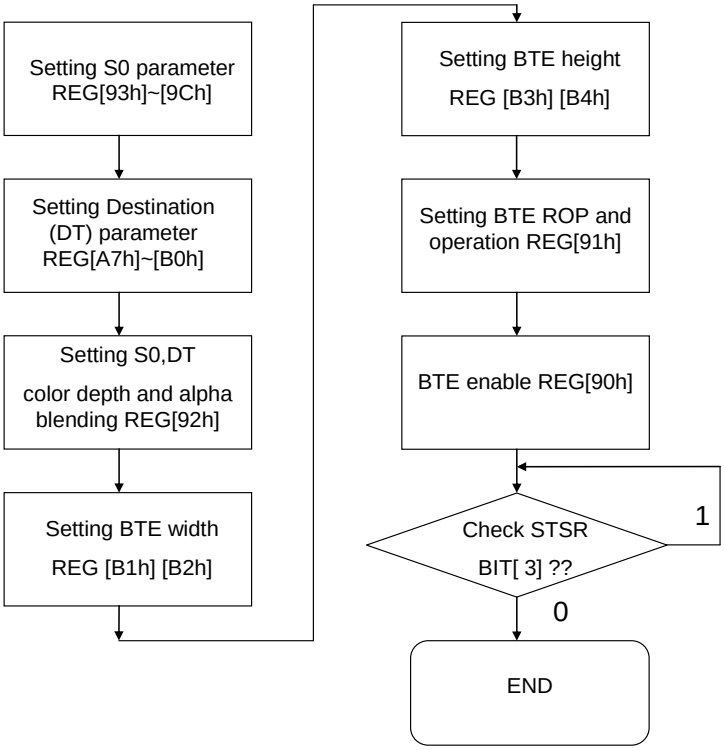


Figure 13-37 : Flow Chart

13.6.12 結合擴展色彩與 Chroma Keying 的記憶體複製

“Memory Copy w/ Color Expansion and chroma key” 會將從 SDRAM 讀取的來源 0 (S0) 單色影像資料 (mono) 轉成彩色影像資料，並且寫入 SDRAM 目的記憶體中。如果單色資料 bit 為“1”，那麼將會轉換成前景色暫存器設定的顏色。如果單色資料 bit 為 “0”，那麼將不會對目的記憶體做任何的更動，以達成透明的效果。

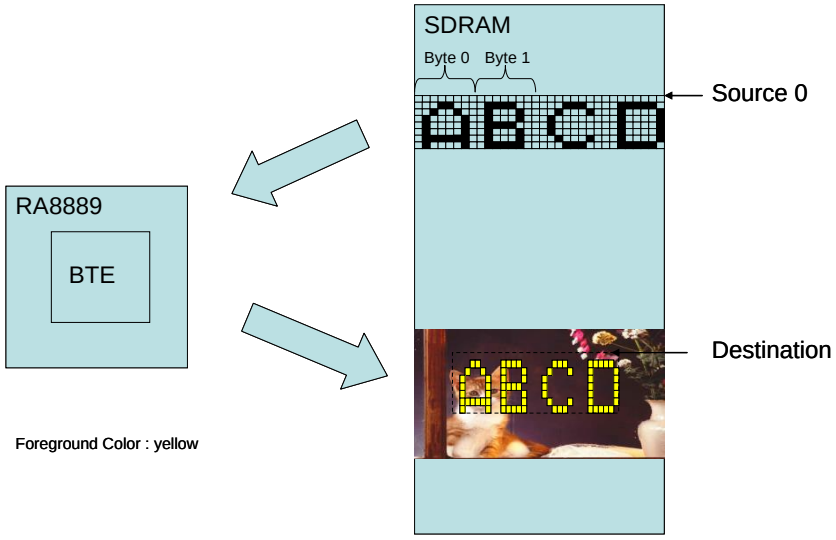


Figure 13-38 : Hardware Data Flow

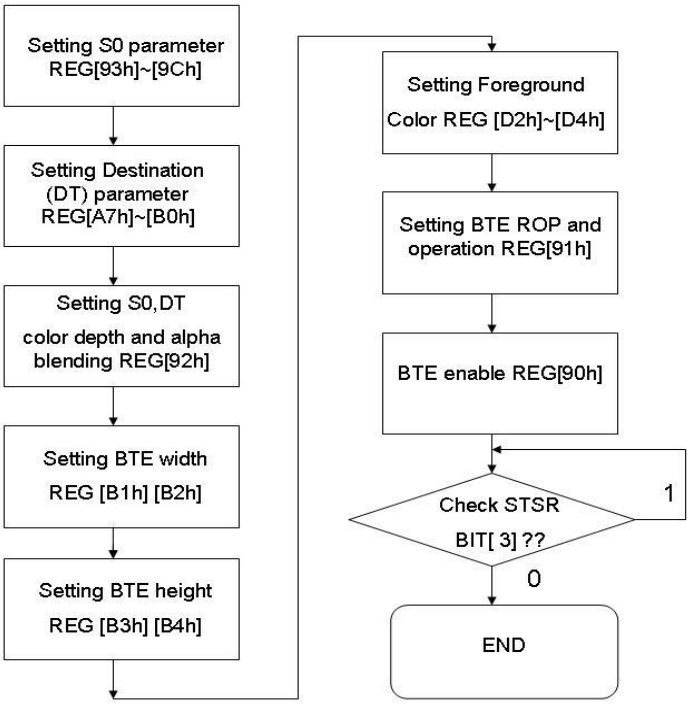


Figure 13-39 : Flow Chart

13.6.13 區域填滿

“Solid Fill BTE” 會針對 BTE 指定的矩形範圍做指定顏色的填滿。這個功能是被使用在填滿一個大範圍區域。而填滿的顏色被設定在 BTE 的前景色暫存器中。

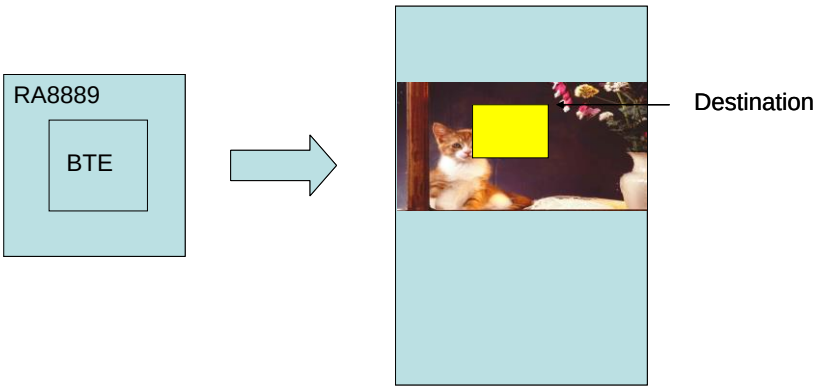


Figure 13-40 : Hardware Data Flow

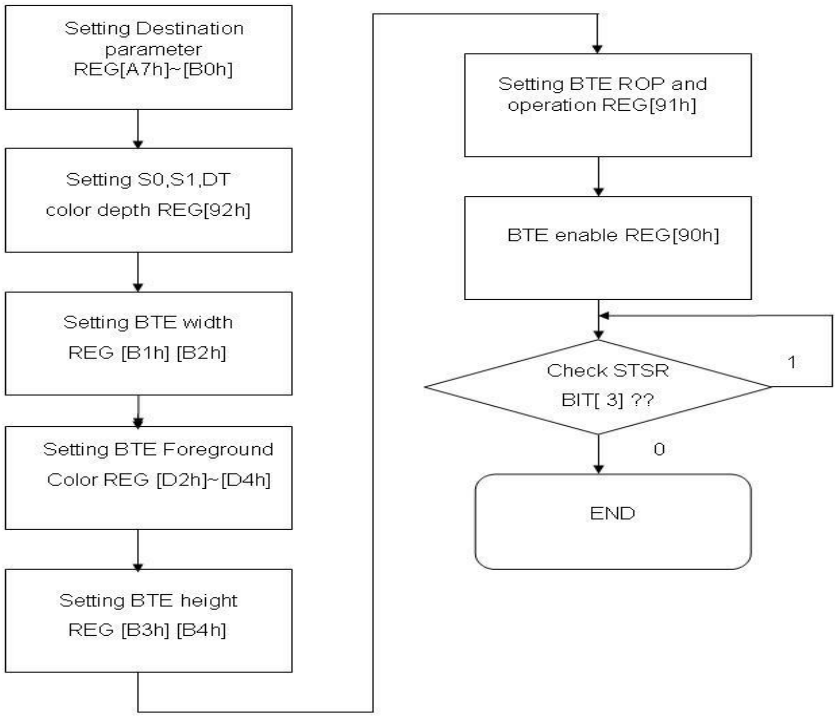


Figure 13-41

14. 文字輸入

RA8889 有三種文字圖形來源:

1. 內建字型，請參考章節 14.1。
2. 外部字型 ROM，請參考章節 14.2。
3. 使用者定義字型 (CGRAM)，請參考章節 14.3。

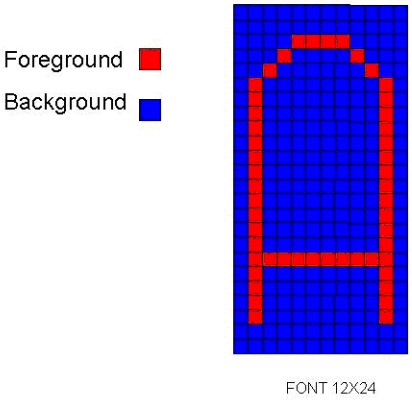


Figure 14-1 : Font Example

當使用者需要更改字型暫存器以顯示不同文字時 (字型參數暫存器是 REG[CCh]~REG[DEh])，使用者可以參考下面的流程圖。而文字顏色可以在前景色與背景色暫存中被設定 (REG[D2h]~REG[D7h])。

例: 以字型 1 寫入 64 個字，再以字型 2 寫入 64 個字。

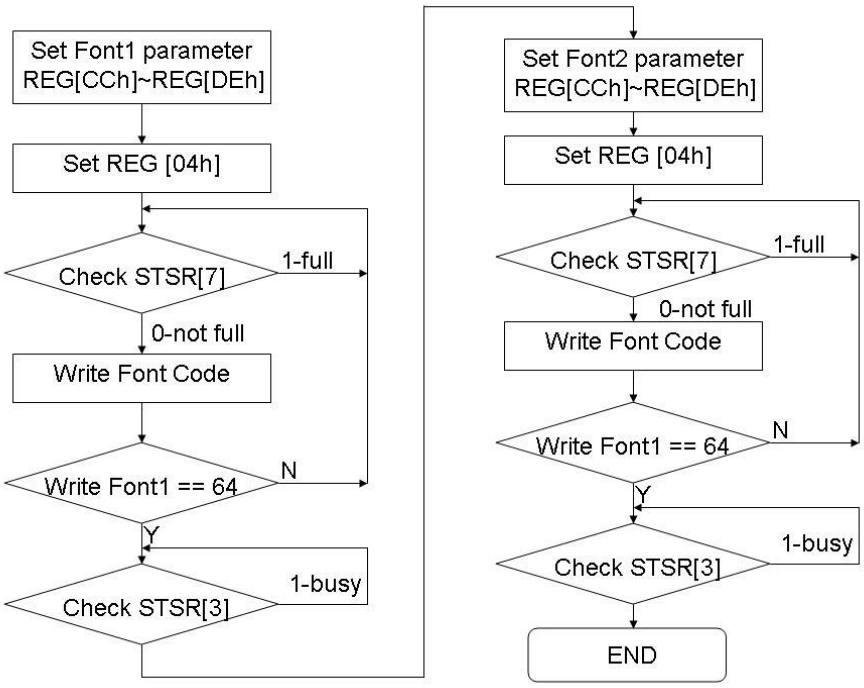


Figure 14-2

14.1 內建字體

RA8889 內建 12x24 ASCII 字體的 ROM，這可以讓使用者很方便的經由輸入 ASCII 以顯示文字。內建字型支援 ISO/IEC 8859-1/2/4/5 編碼標準，此外使用者可以透過前景色 REG[D2h~D4h] 與背景色 REG [D5h~D7h] 設定來選擇文字的顏色。對於程式的流程可以參考下圖:

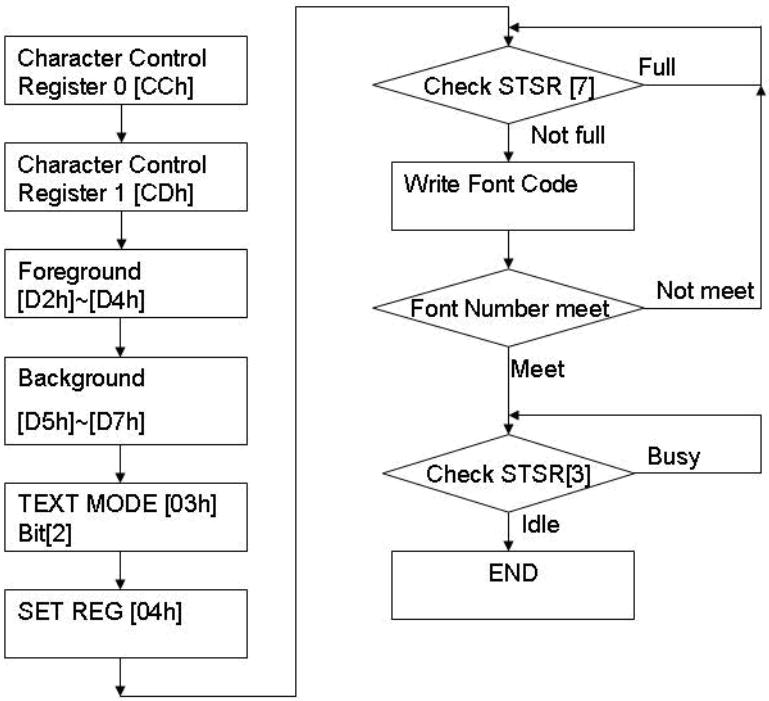


Figure 14-3 : ASCII Character ROM Programming Procedure

Table 14-1 顯示 ISO/IEC 8859-1 字元的編碼方式，ISO 的意思是"International Organization for Standardization"。ISO/IEC 8859-1 一般被稱為"Latin-1"，這是被 ISO 發展出來的 8-bit 字元集的第一部分。拉丁字母的部分要是由 0xA0-0xFF 組成。該字元集用於整個西歐，包括阿爾巴尼亞語、南非語、布列塔尼語、丹麥、法羅群島、弗里斯蘭、加利西亞語、德語、格陵蘭、冰島、愛爾蘭、意大利、拉丁、盧森堡、挪威、葡萄牙、羅曼拉丁語、蘇格蘭蓋爾語、西班牙語、瑞典。英文字母，沒有重音符號也可以使用 ISO / IEC8859-1。此外，它也常用於歐洲以外的許多語言，如斯瓦希里語、印尼、馬來西亞和他加祿語。

下面的表格中，字元碼 0x80-0x9F 是被 Microsoft windows 定義的，被稱為 CP1252 (WinLatin1)。

Table 14-1 : ASCII Block 1(ISO/IEC 8859-1)

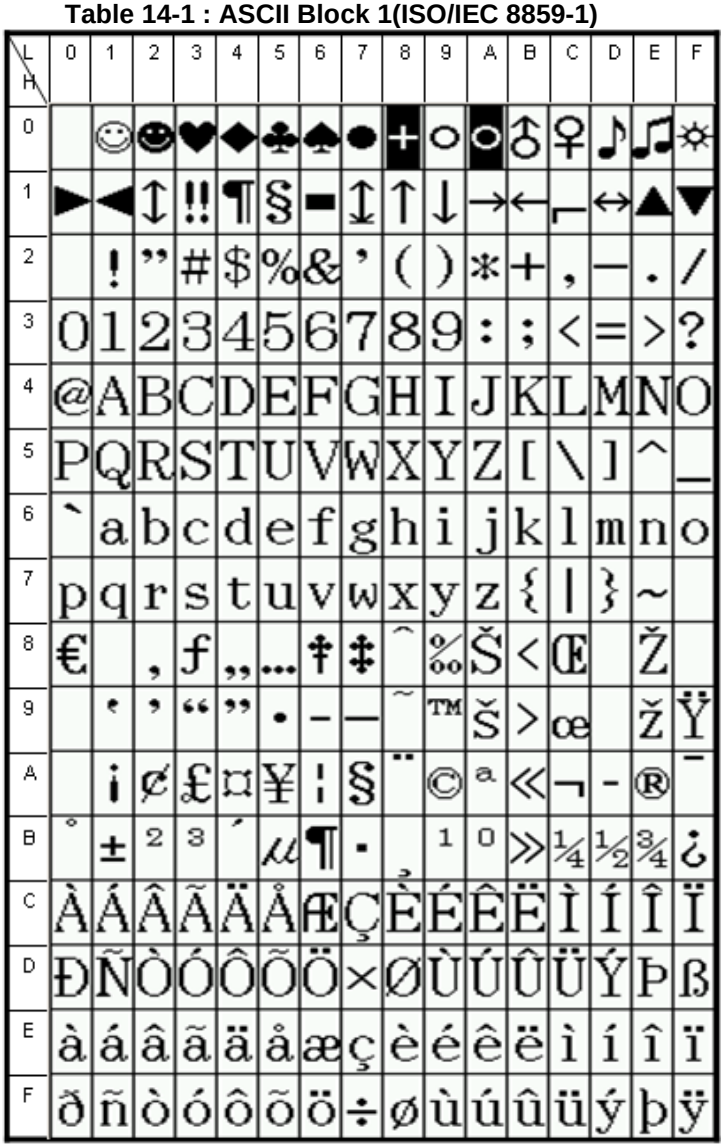


Table 14-2　是 ISO/IEC 8859-2 標準字元，ISO/IEC 8859-2 也被稱為 Latin-2 ，這是 ISO/IEC 8859 8 位元編碼字元的第二部分。這些編碼值幾乎可以用於下列歐洲的通訊交換系統，如克羅地亞語、捷克語、匈牙利語、波蘭語、斯洛伐克語、斯洛文尼亞語和上索布語。塞爾維亞、英語、德語、拉丁語也可以使用 ISO/IEC 8859-2。此外，它也可適用於一些西歐語言，如芬蘭 (除了瑞典和芬蘭使用之外)。

Table 14-2 : ASCII Block 2 (ISO/IEC 8859-2)

L / H	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		☺	☻	♥	♦	♣	♠	●	+	○	◙	♂	♀	♪	♫	☼
1	►	◄	↕	‼	¶	§	▬	↨	↑	↓	→	←	⌐	↔	▲	▼
2		!	”	#	$	%	&	’	(	)	*	+	,	—	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8																
9																
A		Ą	˘	Ł	¤	Ľ	Ś	§	¨	Š	Ş	Ť	Ź	-	Ž	Ż
B	°	ą	˛	ł	´	ľ	ś	ˇ	¸	š	ş	ť	ź	˝	ž	ż
C	Ŕ	Á	Â	Ă	Ä	Ĺ	Ć	Ç	Č	É	Ę	Ë	Ě	Í	Î	Ď
D	Đ	Ń	Ň	Ó	Ô	Ő	Ö	×	Ř	Ů	Ú	Ű	Ü	Ý	Ţ	ß
E	ŕ	á	â	ă	ä	ĺ	ć	ç	č	é	ę	ë	ě	í	î	ď
F	đ	ń	ň	ó	ô	ő	ö	÷	ř	ů	ú	ű	ü	ý	ţ	˙

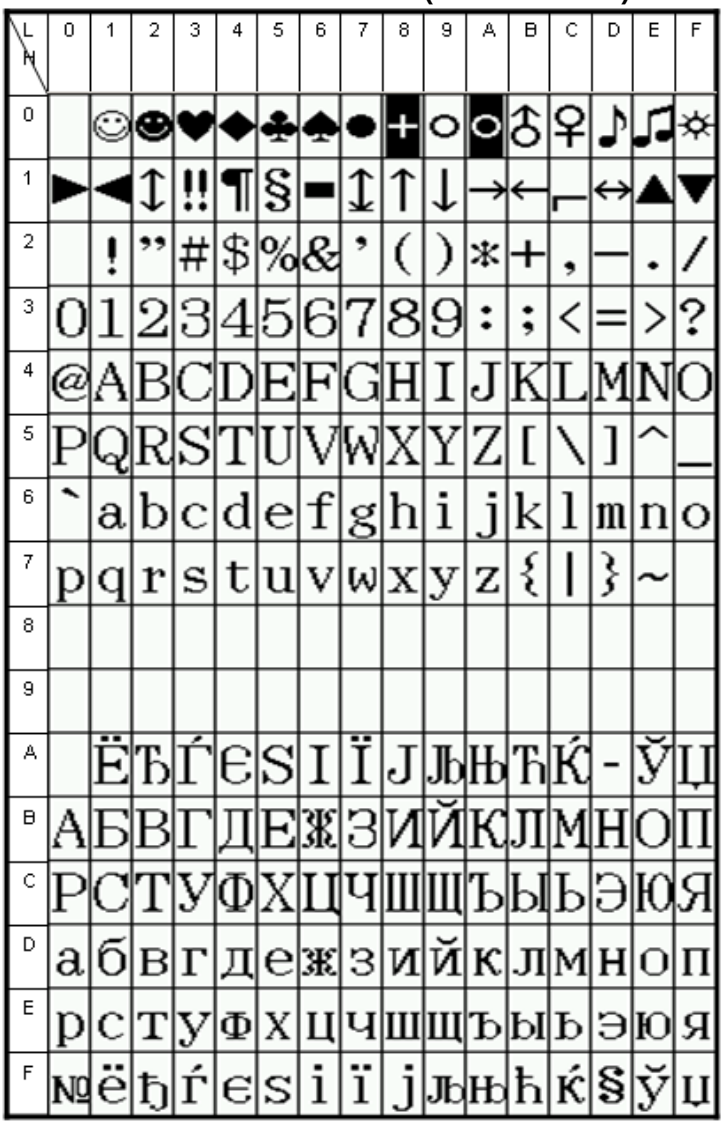
Table 14-3 是 ISO/IEC 8859-4。 ISO/IEC 8859-4 被稱為 Latin-4 或是“North European”，它是 ISO/IEC 8859 8-bit 字元編碼的第四部分。這個主要被使用在愛沙尼亞語、格陵蘭語、拉脫維亞語、立陶宛語和薩米語。而此字元也支援丹麥語、英語、芬蘭語、德語、拉丁語、挪威語、斯洛文尼亞語和瑞典語。

Table 14-3 : ASCII Block 3 (ISO/IEC 8859-4)

L/H	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		☺	☻	♥	♦	♣	♠	•	◘	○	◙	♂	♀	♪	♫	☼
1	►	◄	↕	‼	¶	§	▬	↨	↑	↓	→	←	∟	↔	▲	▼
2		!	”	#	$	%	&	’	(	)	*	+	,	—	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8																
9																
A		Ą	ĸ	Ŗ	¤	Ĩ	Ļ	§	¨	Š	Ē	Ģ	Ŧ	-	Ž	¯
B	°	ą	˛	ŗ	´	ĩ	ļ	ˇ	¸	š	ē	ģ	ŧ	Ŋ	ž	ŋ
C	Ā	Á	Â	Ã	Ä	Å	Æ	Į	Č	É	Ę	Ë	Ė	Í	Î	Ī
D	Đ	Ņ	Ō	Ķ	Ô	Õ	Ö	×	Ø	Ų	Ú	Û	Ü	Ũ	Ū	ß
E	ā	á	â	ã	ä	å	æ	į	č	é	ę	ë	ė	í	î	ī
F	đ	ņ	ō	ķ	ô	õ	ö	÷	ø	ų	ú	û	ü	ũ	ū	˙

Table 14-4 是 ISO/IEC 8859-5， ISO/IEC 8859-5 是 ISO/IEC 8859 8-bit 字元集的第五部。這個字元集主要是支援保加利亞、白俄羅斯、俄羅斯、塞爾維亞和馬其頓。

Table 14-4 : ASCII Block 4 (ISO/IEC 8859-5)



14.2 外部字體 ROM

RA8889 使用外部串列傳輸 ROM 介面以針對不同的應用提供更多的字元選擇。這個功能適用集通字元 ROM，集通公司是專業的字元廠商。RA8889 支援的型號有 GT21L16T1W, GT30L16U2W, GT30L24T3Y, GT30L24M1Z, GT30L32S4W, GT20L24F6Y, GT21L24S1W。集通公司提供的不同產品型號可以支援不同的字型如 16x16, 24x24, 32x32 與不等寬大小以供使用者選擇。詳細的功能描述請參考 Figure 16-12。

14.2.1 GT21L16T1W

- Reg[CEh][7:5]: 000b
- 字高: x16

可用的字組與字寬:

	GB12345 GB18030	BIG5	ASCII	UNI-jpn	JIS0208	Latin	Greek	Cyrillic	Arabic
Normal	V	V	V	V	V	V	V	V	
Arial			V			V	V	V	V
Roman									V
Bold			V						

*Arial & Roman 是可變寬度的。

14.2.2 GT30L16U2W

- Reg[CEh][7:5]: 001b
- 字高: x16

可用的字組與字寬:

	UNICODE	ASCII	Latin	Greek	Cyrillic	Arabic	GB2312 Special
Normal	V	V	V	V	V		V
Arial		V	V	V	V	V	
Roman		V				V	
Bold							

*Arial & Roman 是可變寬度的。

14.2.3 GT30L24T3Y

- Reg[CEh][7:5]: 010b
- 字高: x16

可用的字組與字寬:

	GB2312	GB12345/GB18030	BIG5	UNICODE	ASCII
Normal	V	V	V	V	V
Arial					V
Roman					
Bold					

*Arial & Roman 是可變寬度的。

- 字高: x24

可用的字組與字寬:

	GB2312	GB12345/GB18030	BIG5	UNICODE	ASCII
Normal	V	V	V	V	
Arial					V
Roman					
Bold					

*Arial & Roman 是可變寬度的。

14.2.4 GT30L24M1Z

- Reg[CEh][7:5]: 011b
- 字高: x24

可用的字組與字寬:

	GB2312 Extension	GB12345/ GB18030	ASCII
Normal	V	V	V
Arial			V
Roman			V
Bold			

*Arial & Roman 是可變寬度的。

14.2.5 GT30L32S4W

- Reg[CEh][7:5]: 100b
- 字高: x16

可用的字組與字寬:

	GB2312	GB2312 Extension	ASCII	GB2312 Special
Normal	V	V	V	V
Arial			V	
Roman			V	
Bold				

*Arial & Roman 是可變寬度的。

- 字高: x24

可用的字組與字寬:

	GB2312	GB2312 Extension	ASCII
Normal	V	V	V
Arial			V
Roman			V
Bold			

*Arial & Roman 是可變寬度的。

- 字高: x32

可用的字組與字寬:

	GB2312	GB2312 Extension	ASCII
Normal	V	V	V
Arial			V
Roman			V
Bold			

*Arial & Roman 是可變寬度的。

14.2.6 GT20L24F6Y

- Reg[CEh][7:5]: 101b
- 字高: x16

可用的字組與字寬:

	ASCII	Latin	Greek	Cyrillic	Arabic	Hebrew	Thai	ISO-8859
Normal	V	V	V	V		V	V	V
Arial	V	V	V	V	V			
Roman	V							
Bold	V							

*Arial & Roman 是可變寬度的。

- 字高: x24

可用的字組與字寬:

	ASCII	Latin	Greek	Cyrillic	Arabic
Normal		V	V	V	
Arial	V				V
Roman					
Bold					

*Arial & Roman 是可變寬度的。

14.2.7 GT21L24S1W

- Reg[CEh][7:5]: 110b
- 字高: x24

可用的字組與字寬:

	GB2312	GB2312 Extension	ASCII
Normal	V	V	V
Arial			V
Roman			
Bold			

*Arial & Roman 是可變寬度的。

14.3 使用者定義字體

使用者可以使用" User-defined Characters" 創建字元或符號,此功能可以支援半形 (8x16/12x24/16x32 dots) 與全形 (16X16/24X24/32X32 dots)，此功能支援 32,768 半形字或 32,768 全形字，半形字元編碼範圍是在 0000h~7FFFh，而全形字編碼範圍則是 8000h~FFFFh。當使用者輸入字元碼，則 RA8889 將會將其索引至 SDRAM 字元空間 (CGRAM)，並且將字元或符號寫字顯示記憶體區間。而字元的顏色可以由前景色 REG[D2h~D4h] 與背景色 REG[D5h~D7h] 暫存器定義。

14.3.1 CGRAM 中 8x16 字體的格式

CGRAM ADDRE CALCULATE = (CGRAM_START_ADDR) + ((FONT CODE) * 16)

EXAMPLE :

CGRAM_START_ADDR = 1000h

CHARACTER_CODE = 0001h

THEN FONT ADDR = 1010h

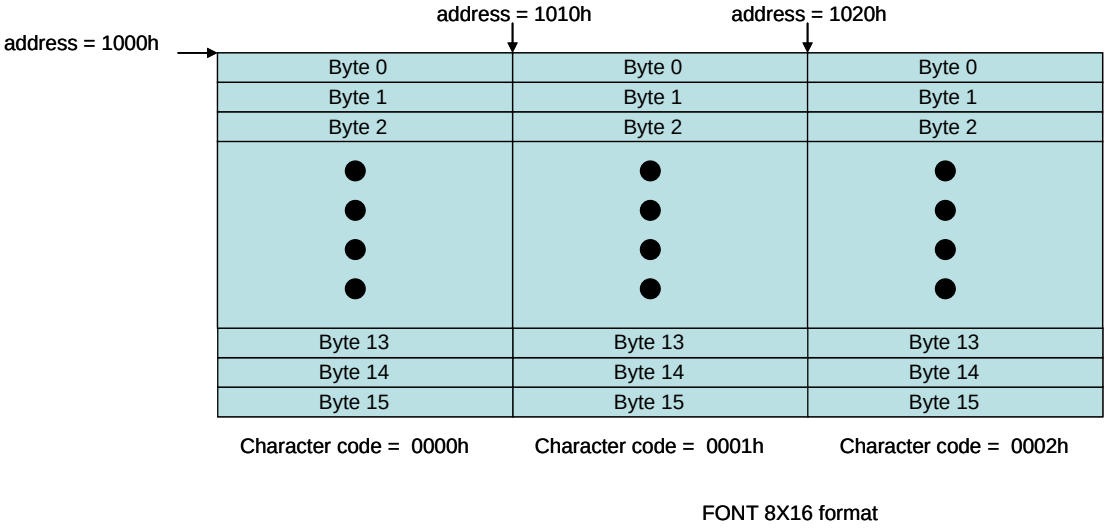


Figure 14-4 : Font 8X16 Array in SDRAM

14.3.2 CGRAM 中 16x16 字體的格式

CGRAM ADDRE CALCULATE = (CGRAM_START_ADDR) + ((FONT CODE - 8000h) * 32)

EXAMPLE :

CGRAM_START_ADDR = 1000h

CHARACTER_CODE = 8001h

THEN FONT ADDR = 1020h

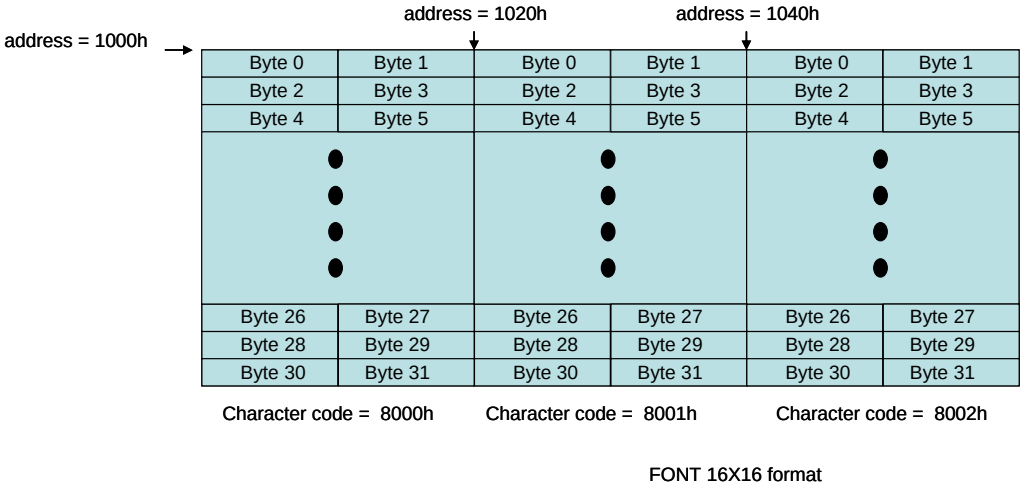


Figure 14-5 : Font Array 16x16 in SDRAM

14.3.3 CGRAM 中 12x24 字體的格式

CGRAM ADDRE CALCULATE = (CGRAM_START_ADDR) +
((FONT CODE) * 48)

EXAMPLE :

CGRAM_START_ADDR = 1000h

CHARACTER_CODE = 0001h

THEN FONT ADDR = 1030h

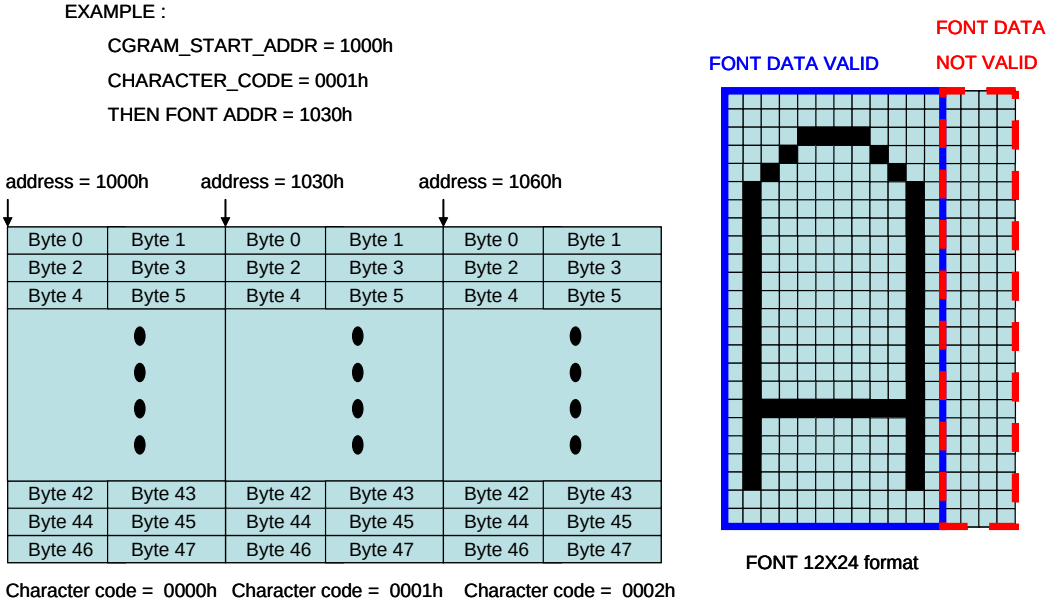


Figure 14-6 : Font Array 12x24 in SDRAM

14.3.4 CGRAM 中 24x24 字體的格式

CGRAM ADDRE CALCULATE = (CGRAM_START_ADDR) + ((FONT CODE – 8000h) * 72)

EXAMPLE :

CGRAM_START_ADDR = 1000h

CHARACTER_CODE = 8001h

THEN FONT ADDR = 1048h

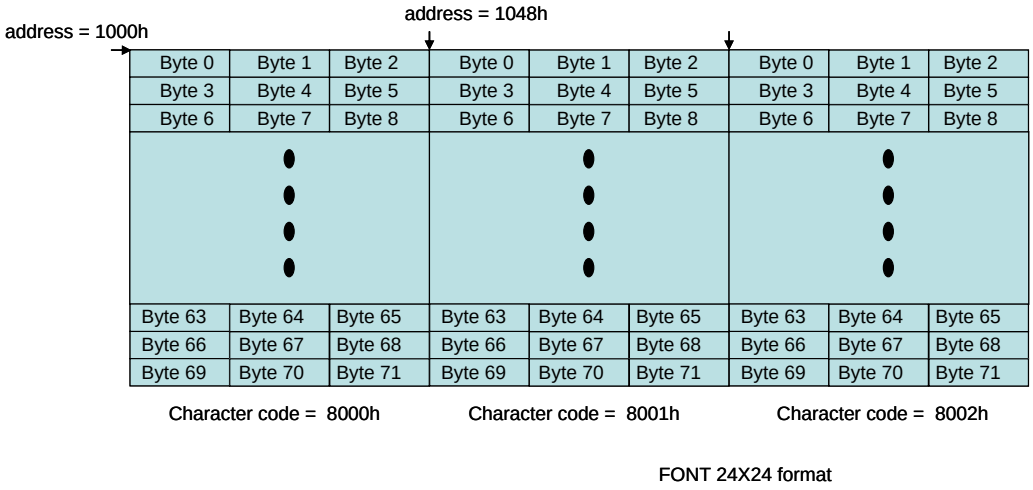


Figure 14-7 : Font Array 24x24 in SDRAM

14.3.5 CGRAM 中 16x32 字體的格式

CGRAM ADDRE CALCULATE = (CGRAM_START_ADDR) + ((FONT CODE) * 64)

EXAMPLE :

CGRAM_START_ADDR = 1000h

CHARACTER_CODE = 0001h

THEN FONT ADDR = 1040h

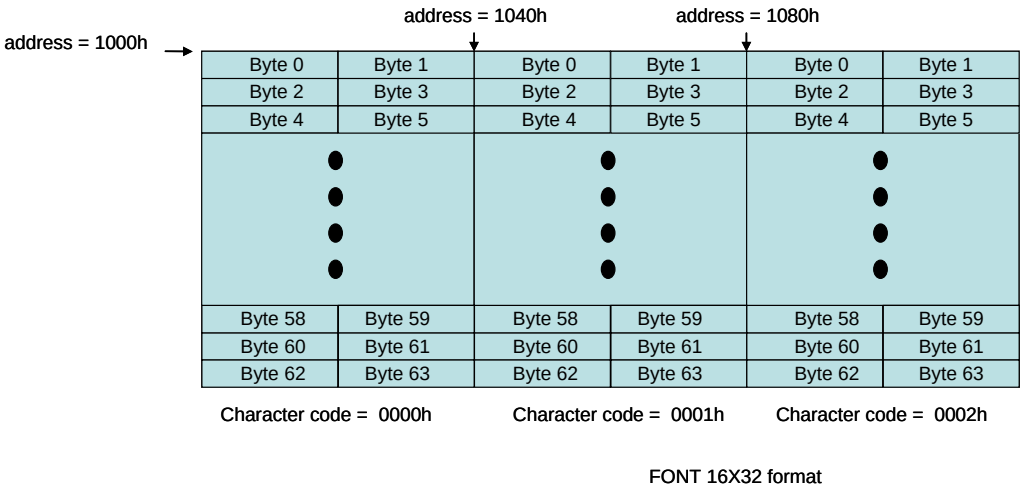


Figure 14-8 : Font 16x32 Array in SDRAM

14.3.6 CGRAM 中 32x32 字體的格式

CGRAM ADDRE CALCULATE = (CGRAM_START_ADDR) + ((FONT CODE – 8000h) * 128)

EXAMPLE :

CGRAM_START_ADDR = 1000h

CHARACTER_CODE = 8001h

THEN FONT ADDR = 1080h

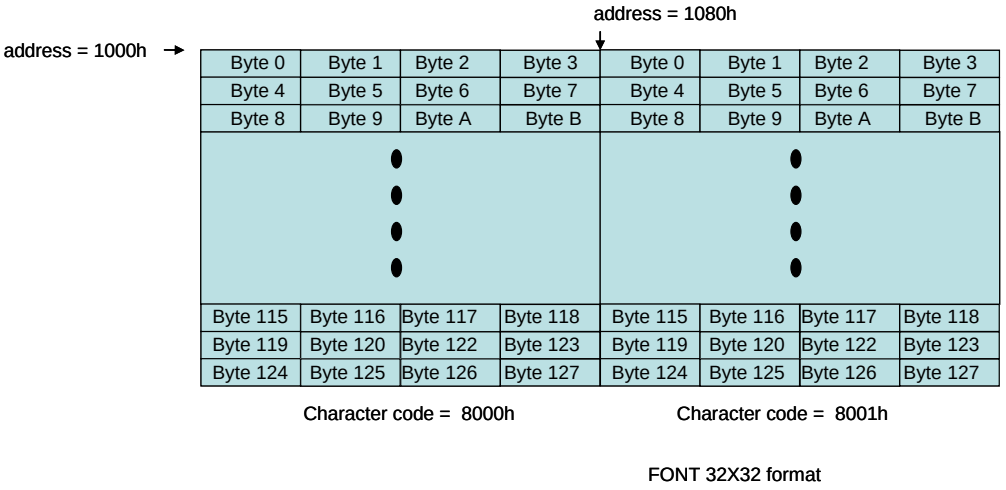


Figure 14-9 : Font 32x32 Array in SDRAM

14.3.7 關於 MPU 初始化 CGRAM 的流程

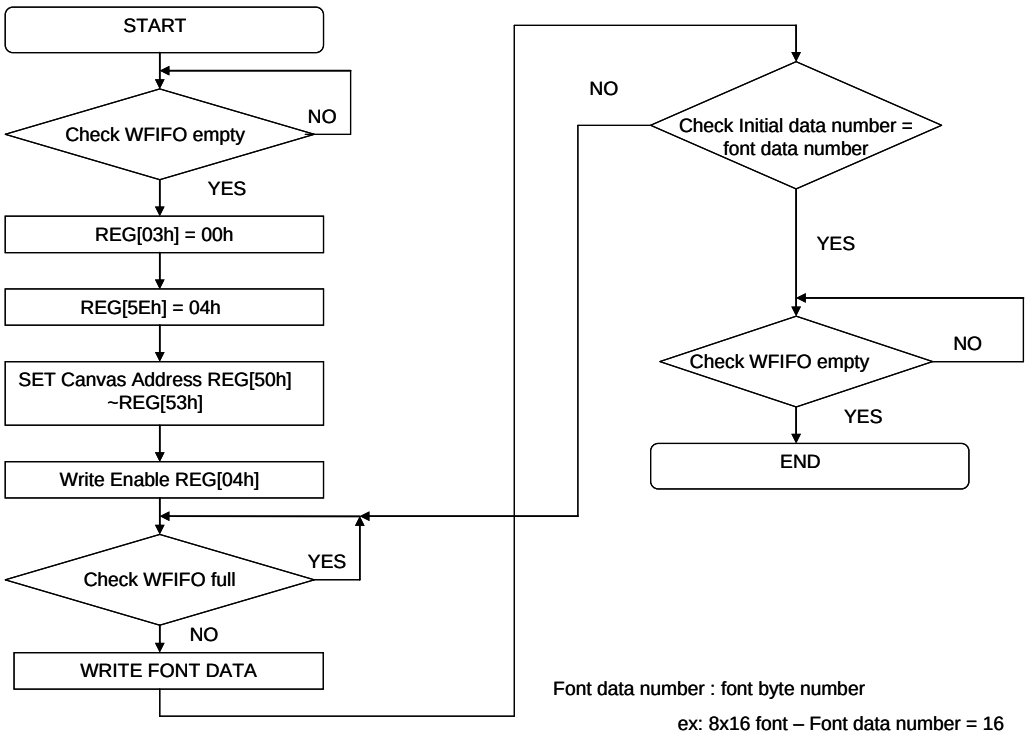


Figure 14-10 : Initial CGRAM from MPU

14.3.8 關於利用 Serial Flash 初始化 CGRAM 的流程

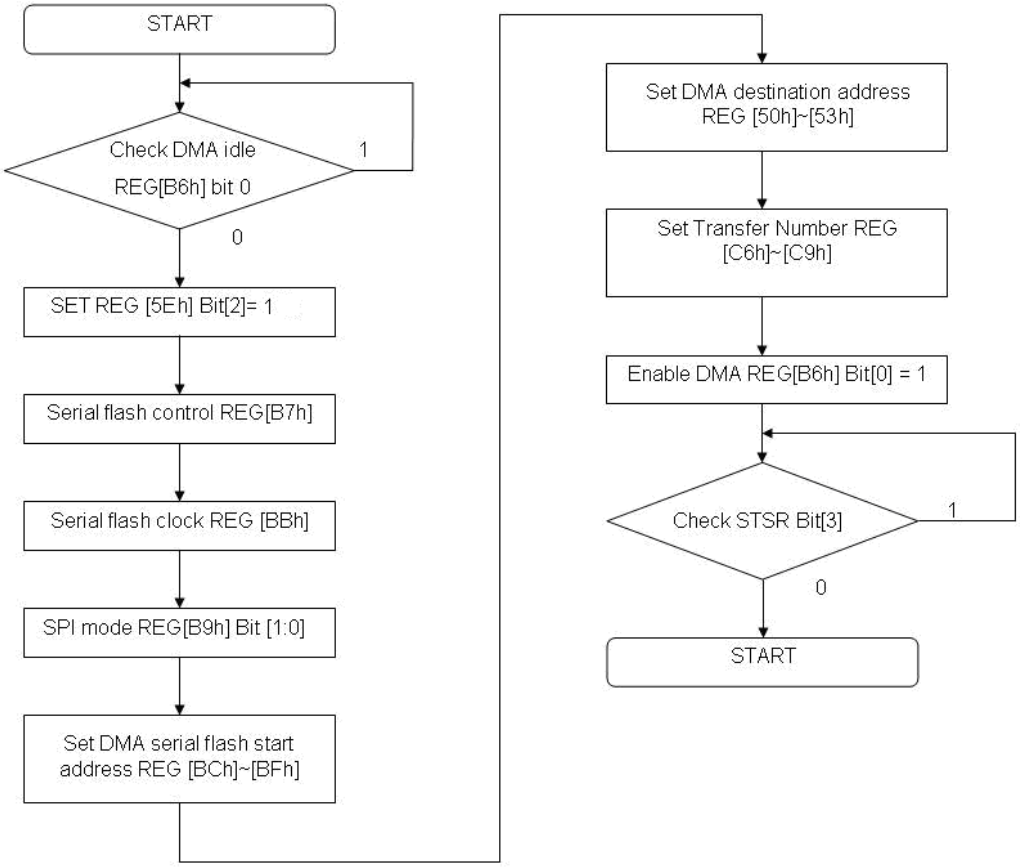


Figure 14-11 : Initial CGRAM from Serial Flash

14.4 文字旋轉 90 度

標準文字的輸入是由左到右然後再由上到下。而 RA8889 支援文字旋轉功能，字元可以逆時針旋轉 90 度，此功能的達成是需要設定暫存器 REG[CDh] Bit4 = 1，另外還需設定正確的 VDIR (REG[12h] Bit3)，這樣 LCD 模組可以顯示旋轉 90 的字元。在文字旋轉模式，達成這個功能主要在寫入時是先上到下然後再左到右。

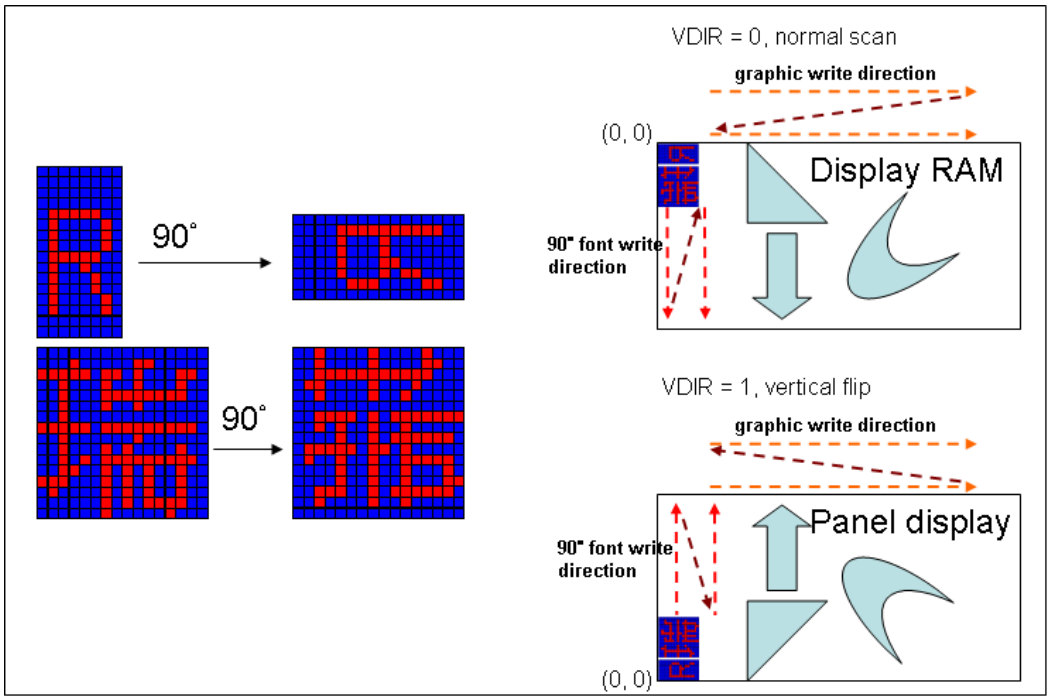


Figure 14-12: Rotation 90∘Characters

當使用者旋轉螢幕為順時針 90 度，使用者將會看到螢幕如下。

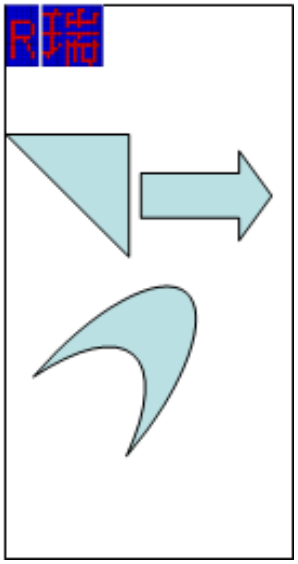


Figure 14-13

14.5 字體放大與透明

RA8889 支援字型放大(REG[CDh] Bit[3:0])，與透明功能(REG[CDh] Bit6)。而且這些功能可以同時被使用。

下圖為放大及透明字型的範例:

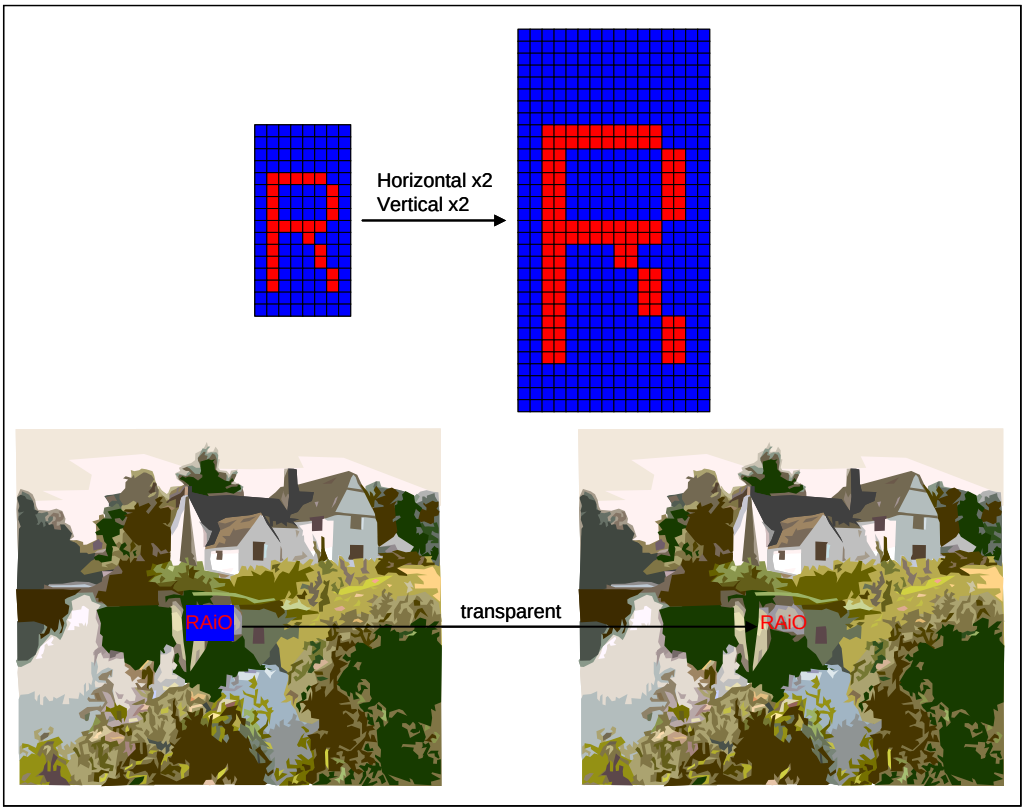


Figure 14-14 : Enlargement and Transparent Characters

14.6 自動換行

RA8889 支援在文字寫入時，文字游標位置自動累加，並且在寫入文字時遇到工作視窗邊緣會自動換行。在文字模式時，當寫入文字在垂直與水平超過工作視窗範圍時，會自動換到下一行。關於自動換行請參下圖:

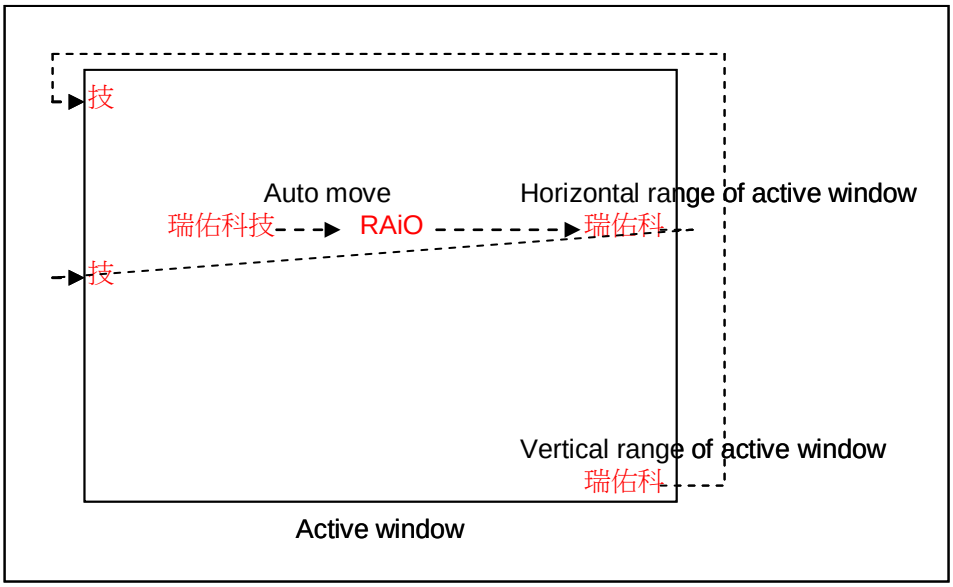


Figure 14-15 : Auto Line feed in Text Mode

14.7 字元對齊

RA8889 支援字元對齊功能，這個功能可以讓使用者再寫入全半形字可以很方便的對齊。經由設定 REG[CDh] Bit7 = 1，寫入的全半形文字會是如下面的圖所顯示的:

註: 當使用集通字型內的不等寬字型，字元對齊功能是被禁能的。

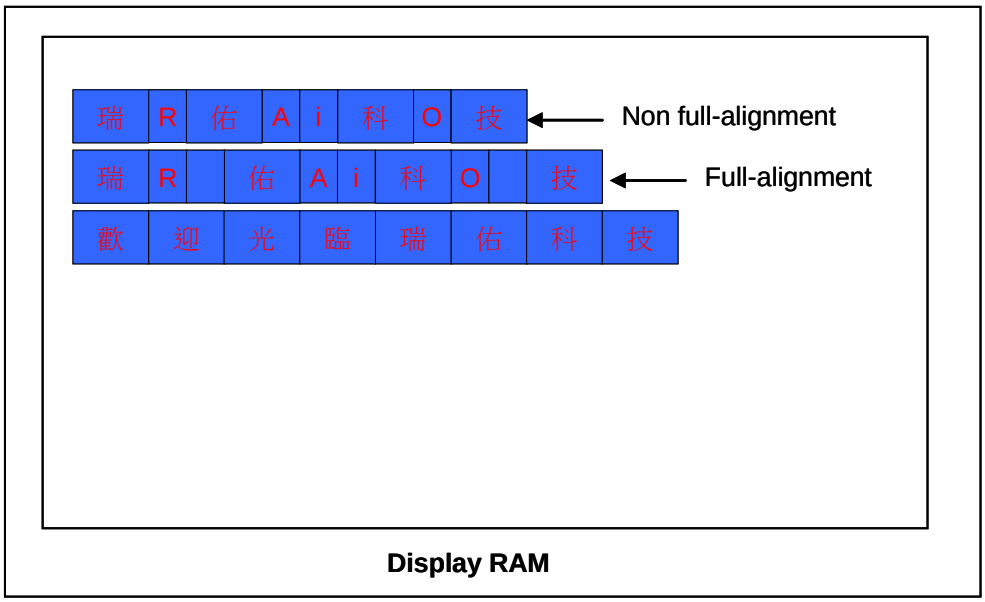


Figure 14-16: Full-Alignment Function

14.8 游標

RA8889 提供兩種游標。一個是圖形游標一個是文字游標。圖形游標使用 32X32 像素的圖形來表示，而圖形游標可以被顯示在使用者定義的位置，當設定位置改變時，圖形游標就會被移動。文字游標是提供文字寫入時的相關游標。文字游標的寬度與高度外觀是可以被程式化的。文字游標顯示的是文字可以寫入的位置。

註：游標只能顯示在主視窗的座標內。

14.8.1 文字游標

文字游標位置可以被設成閃爍/不閃爍。游標自動移動功能必須是在工作視窗內。當文字寫入時，文字游標會自動累加到下一個文字輸入的位置，而每次移動的距離與文字大小與方向有關。當符合工作視窗的邊緣時，游標將會移動到下一行。行高的大小可以以像素為單位來設定。Table 14-5 列出相關的暫存器描述。

Table 14-5 : Text Write Cursor Related Register Table

Register Name	Bit Num	Function Description	Address
FLDR	4-0	Character Line Gap Setting Register	D0h
F_CURX0/1	7-0 / 4-0	Text Cursor Horizontal Location	63h, 64h
F_CURY0/1	7-0 / 4-0	Text Cursor Vertical Location	65h, 66h
ICR	2	Text Mode Enable 0 : Graphic mode. 1 : Text mode.	03h
GTCCR	1	Text Cursor Enable 0 : Text cursor is not visible. 1 : Text cursor is visible.	3Ch
	0	Text Cursor Blink Enable 0 : Normal display. 1 : Blink display.	

Cursor Attribute – Cursor Blinking

文字游標可以設定成固定頻率的的閃爍或不閃爍。控制暫存器為 GTCCR(REG[3Ch])，閃爍的行為為 on (可見) 與 off (不可見)，閃爍時間可以被程式化其計算公式如下:

Blink Time (sec) = BTCR[3Dh]x(1/Frame_Rate).

Figure 14-17 為游標閃爍的例子，游標的位置將會是在最後一個寫入字的後面。

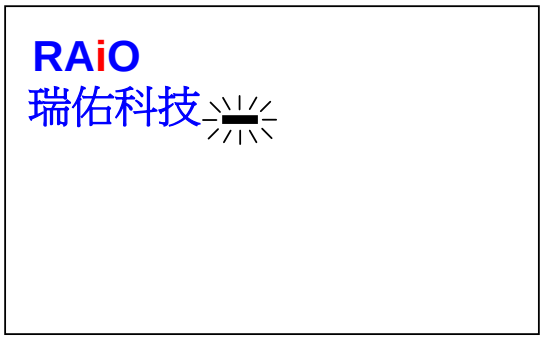
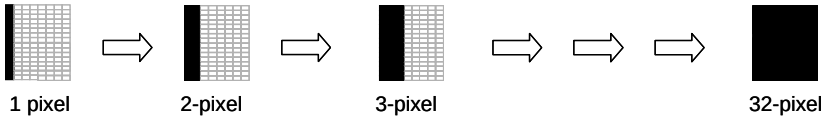


Figure 14-17: Cursor Blinking

游標屬性– 游標的寬跟高

文字游標的外觀是可以被程式化的，主要是指游標的高度與寬度部分。控制暫存器是 CURHS (REG[3Eh])與 CURVS (REG[3Fh])。文字游標在圖形模式中的寬度是可程式化，而高度則故固定為 1 像素高，請參考 Figure 14-18。文字游標的高度與寬度也與文字是否被放大有關 (REG[CDh] Bit3~0)，當放大功能被設為 1 時，游標寬度可以被設為 CURHS/CURVS 1~32 像素；當放大功能不是設成 1 時，游標的寬度與高度將會是原來的倍數相關。 Figure 14-18 是一個水平垂直設為 1 的例子。請注意文字游標外觀不會被字元旋轉影響。關於顯示部分請參考下面的 Figure 14-19。

REG[3Eh] Text Cursor Horizontal Size Register (CURHR)	
Bit4-0 Text cursor horizontal size setting[4:0]	Width (Unit : Pixel)
00000 ~ 11111	1 ~ 32



REG[3Fh] Text Cursor Vertical Size Register (CURVR)	
Bit4-0 Text cursor Vertical size setting[4:0]	Height (Unit : Pixel)
00000 ~ 11111	1 ~32

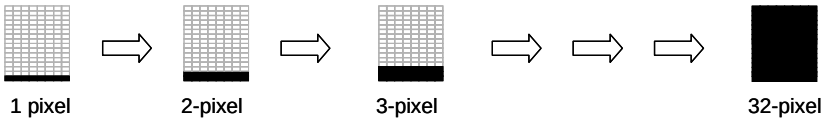


Figure 14-18 : Text Cursor Height and Width Setting

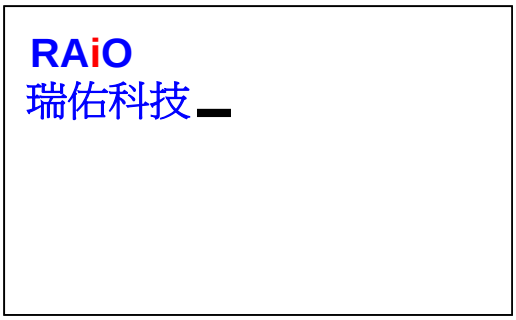


Figure 14-19 : Text Cursor Movement (without rotate)

14.8.2 圖形游標

圖形游標大小為 32x32 像素，每個像素由 2-bit 組成，指向四種顏色設定(color 0、 color 1、背景色、背景色反向)，這表示圖形游標需要 256 bytes (32x32x2/8) 大小。RA8889 提供 4 種圖形游標可供選擇，使用者可以經由設定相關的暫存起來選擇游標。另外，圖形游標位置可由透過 GCHP0 (REG[40h]), GCHP1 (REG[41h]), GCVP0 (REG[42h]) 與 GCVP1 (REG[43h]) 設定得到。而透過暫存器的顏色的設定可以得到顏色 0 (REG[44h])、顏色 1 (REG[45h])/背景色/背景色反向，關於詳細的說明請參考範例。

註：圖形游標的儲存方式只支援 8-bi t 資料。使用者初始化圖形游標需要在圖形模式下針對圖形游標的記憶空間寫入 256 個 8bit 的資料；如果在寫入過程中不檢查 Busy 並且 XnWait 機制沒被使用的話，那每筆資料的寫入必須相隔 5 個以上的系統時脈。

2 Bits Represent 1 Pixel	Pixel Color
2'b00	GCC0 (REG[44h])
2'b01	GCC1 (REG[45h])
2'b10	Background Color
2'b11	The Inversion of Background Color

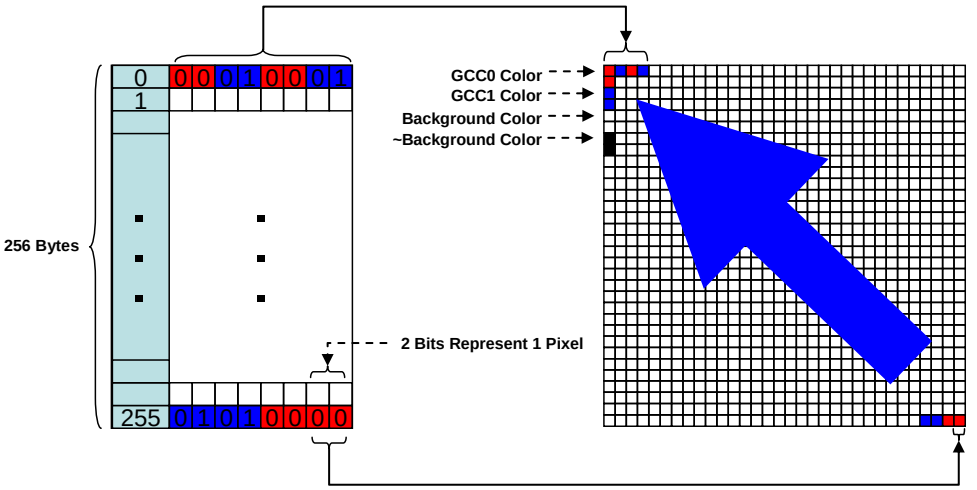


Figure 14-20 : Relation of Memory Mapping for Graphic Cursor

程序:

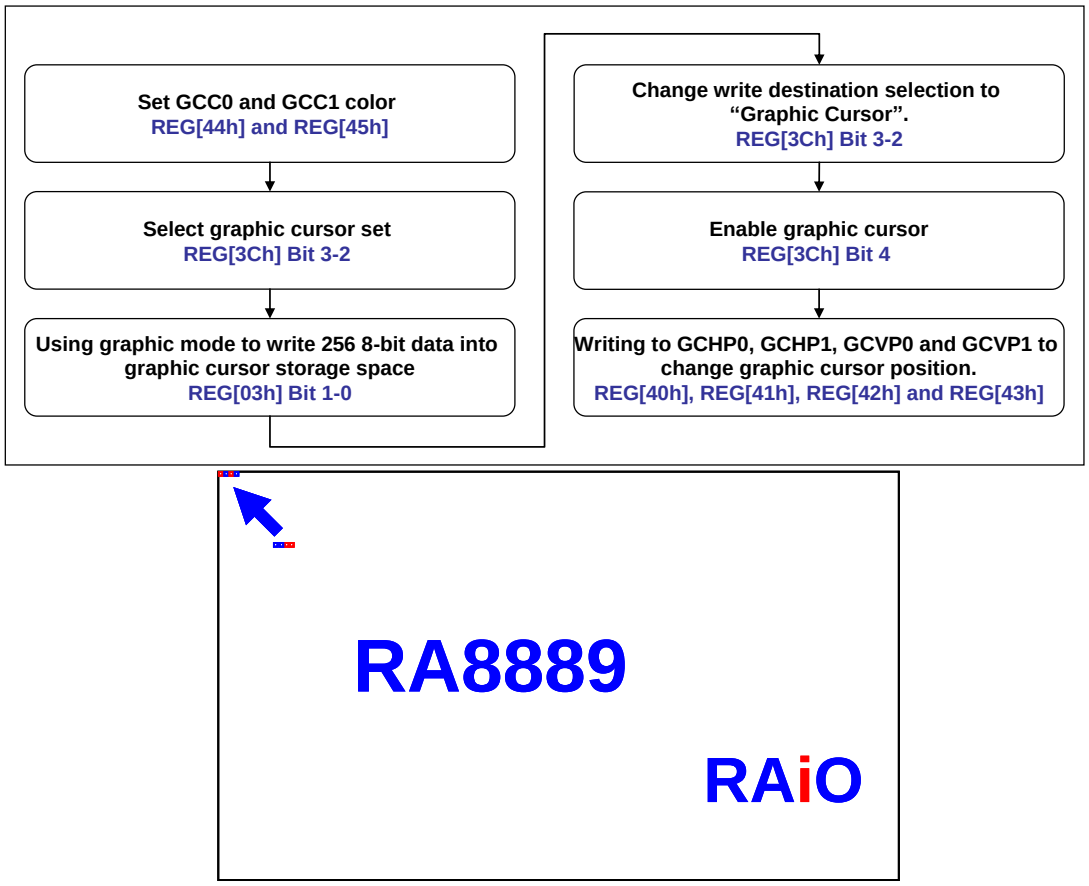


Figure 14-21 : The Display with Graphic Cursor

15. 脈寬調變計數器 (PWM Timer)

RA8889具有兩個16-bit 計數器，計數器0 與1 具有脈寬調變功能 (PWM)。計數器0具有Dead-Zone產生功能，被使用控制在大電流裝置的應用上。

計數器0與1分享同一個8-bit 預先倍數器。每個計數器的除頻器可以產生4種不同的除頻訊號 (1, 1/2, 1/4 & 1/8)。每個計數區塊由各自的除頻器產生各自的時脈訊號，除頻器的時脈則是由相應的8-bit預先倍數器而來。8-bit預先倍數器可被程式化，也可以根據載入值來使用CCLK除頻，這個相關的暫存器是 PSCLR 與PMUXR。計數器的緩衝暫存器 (TCNTBn) 在計數器致能並且下數完成時會載入初始值。計數器在下數時會與緩衝暫存器 (TCMPBn) 比較，當比較相等時會自動載入初始值。TCNTBn 與TCMPBn 雙緩衝的功能可以讓輸出頻率與工作週期發生改變時，有一個穩定的輸出。

每個計數器有各自的下數計數器。當下數達到0時，那麼計數器的中斷會產生以告知CPU 計數操作完畢。當計數器達到0時，TCNTBn 值會自動被載入下數計數器並且繼續下一次的計數。然而，如果計數器停止，假設在計數器執行中清掉 PCFGR 的計數器致能位元，則TCNTBn 將不會被載入計數器中。

TCMPBn 被使用在 PWM 上，主要是可以透過計數器控制邏輯讓輸出的準位與TCMPBn 比較。因此表現出來的行為就是透過 TCMPBn 可以控制PWM 輸出的開關時間 (turn-on,turn-off time)。

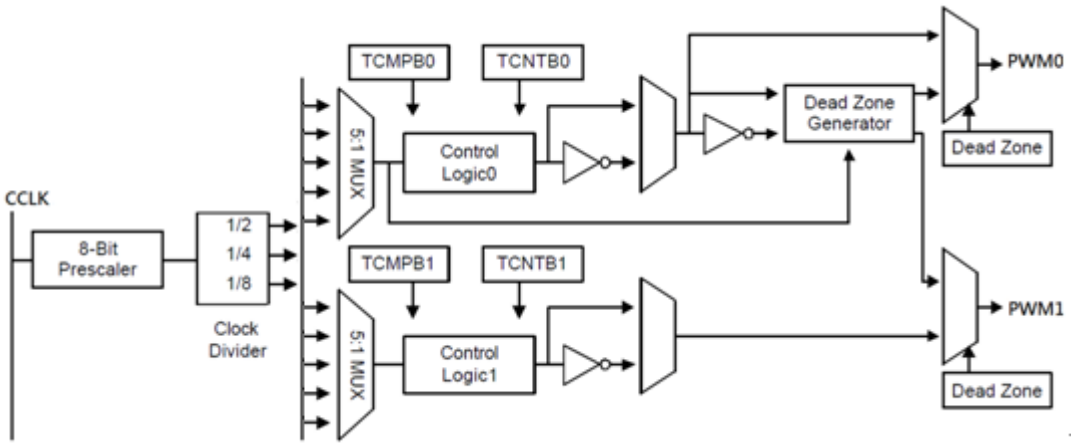


Figure 15-1 : 16-bit PWM Timer Block Diagram

15.1 計數器的基本運作

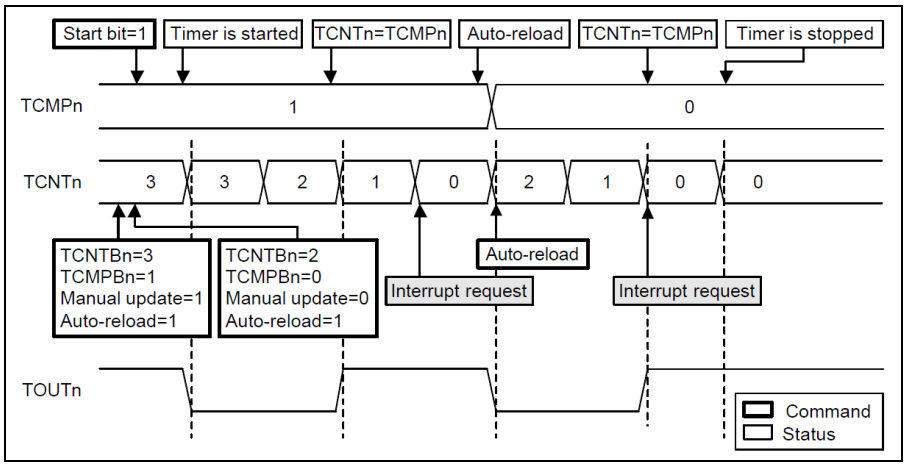


Figure 15-2 : Timer Operations

計數器具有TCNTBn、TCNTn、TCMPBn 與TCMPn 暫存器。(TCNTn 與 TCMPn 是內部暫存器，TCNTn 可以經由讀取TCNTOn 得到)，當下數到0時，TCNTBn 與 TCMPBn 會被載入 TCNTn 與 TCMPn 中。若是中斷被致能，則當TCNTn下數達到0時中斷會產生。

15.2 自動重載與雙緩衝

PWM計數器具有雙緩衝功能，對於下一次操作計數器必須要設定一個新的重載數值進入暫存器中，這個設定的過程不需停掉目前計數器的運作。因此雖然有新的計數器重載參數被設定，但是目前的PWM的行為仍能完整執行結束。

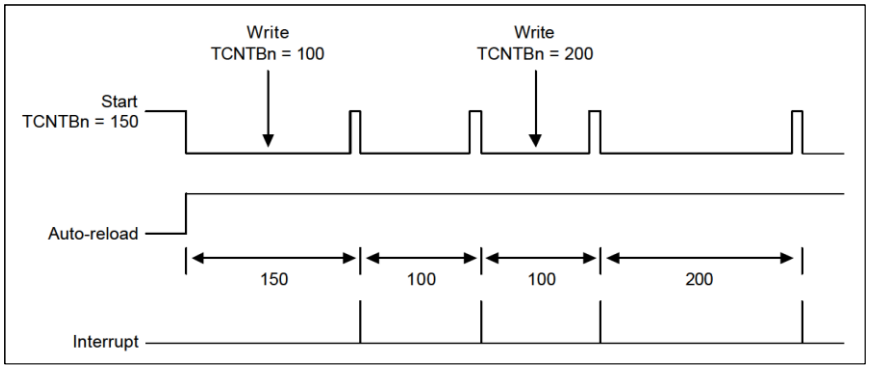


Figure 15-3 : Example of Double Buffering Function

15.3 初始化計數器與反向位元

自動重載的功能是發生在計數器下數到0的時候，因此在開始使用計數器之前必須先將 TCNTn 設定完成。下面的步驟說明如何使用計數器:

1) 寫入 TCNTBn 與 TCMPBn 的初始值。
2) 建議依照需求設定反相輸出位元(不論是否使用反相器)。
3) 設定對應的計數器致能起始位元。

如果計數器被強制停止，TCNTn 將會繼續計數到0再停止，如果有新的值被設定，那麼在下次開始計數之前 TCNTBn 的值會被載入。

註:
每當 PWMn 的反向位元被on/off 時，PWMn 的輸出值會馬上變化，因此使用者應該是在開始計數前就先設定好反相位元。

15.4 計數器的運作

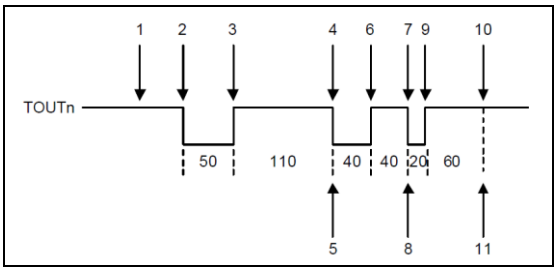


Figure 15-4 : Example of a Timer Operation

Figure 15-4 說明以下的步驟:

1. 致能自動重載功能，設定 TCNTBn 為160 (50+110)與 TCMPBn 為110。設定反相位元 (on/off)。然後再設定 TCNTBn 為80與 TCMPBn 為40，以決定下一個重載值。
2. 設定起始位元、反相關閉、自動重載開啟，計數器在等待一段時間後會開始下數。
3. 當 TCNTn 與 TCMPn 值相同時，PWMn 輸出將由 Low 到 high。
4. 當 TCNTn 下數到0時，中斷會被產生並且 TCNTBn 被載入到暫時的暫存器中。在下一個 clock 來到時，TCNTn 將會從暫時的暫存器中重新載入 (TCNTBn)。
5. 在中斷向量副程式 (ISR)，TCNTBn 與 TCMPBn 被設定80 (20+60)與60，為了下一次的計數使用。
6. 當 TCNTn 具有與 TCMPn 相同的值，PWMn 將會由 Low 到High。
7. 當 TCNTn 下數達到0，TCNTn 將會使用 TCNTBn 的做重載，並且會產生中斷。
8. 在中斷向量副程式 (ISR)，自動重載與中斷被禁能以停止計數器。.
9. 當 TCNTn 與 TCMPn 值相同時，PWMn 會由 low 到 high。
10. 即使 TCNTn 下數到0，TCNTn 也不會重載並且計數器會停止，因為自動重載被禁能了。
11. 沒有更多的中斷請求被產生。

15.5 脈寬調變 (PWM)

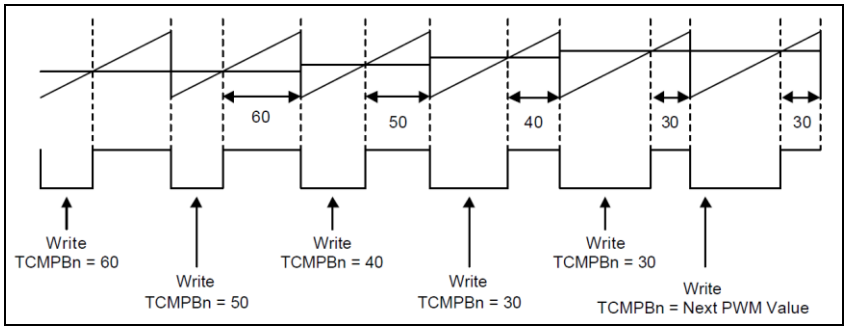


Figure 15-5 : Example of PWM

PWM 功能可以使用 TCMPBn 完成。PWM 頻率被 TCNTBn 決定，上圖說明 PWM 的值由 TCMPBn 決定。要得到高的 PWM 值，需要減少 TCMPBn 值。要得到低的 PWM 值，要增加 TCMPBn 值。對於以上描述而言，如果輸出反相被致能，則增加/減少則是相反。雙緩衝功能允許在任意時間點去改變數值，不論是由 ISR 或是其他的程式來的。

15.6 控制輸出準位

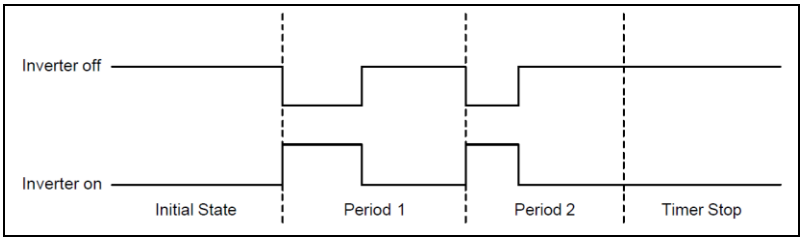


Figure 15-6 : Inverter On/Off

下面的步驟描述如何使 PWM 在高電位或低電位(假設反相位元被關閉):

1. 關閉自動重載，然後PWM 會輸出高電位，並且計數器在下數到0時計數器會停止。
2. 清除起始停止位元以便停止計數器，如果 TCNTn < TCMPn 則輸出值為高電位，如果 TCNTn > TCMPn，則輸出值為低電位。
3. PWMn 經由 PCFGR 反相位元可以設定輸出值反相，這樣可以移除外加的反相器電路。

15.7 Dead Zone 產生器

Dead-Zone 產生器是 PWM 用在控制大電力裝置，這個功能讓開啟的裝置與關閉的裝置間有一個時間差。這個時間差可以避免兩個裝置同時被開啟，即使是很少的時間。 PWM0 是原始的 PWM 訊號，nPWM0 則是 PWM 訊號的反相。如果Dead-Zone被致能，則 PWM0 與 nPWM0 是相當於PWM0_DZ 與 nPWM0_DZ 。而且 nPWM0 / nPWM0_DZ 是由 PWM1 輸出的。在內部電路的 Dead-Zone處理上，PWM0_DZ 與 nPWM0_DZ 不會同時被開啟。

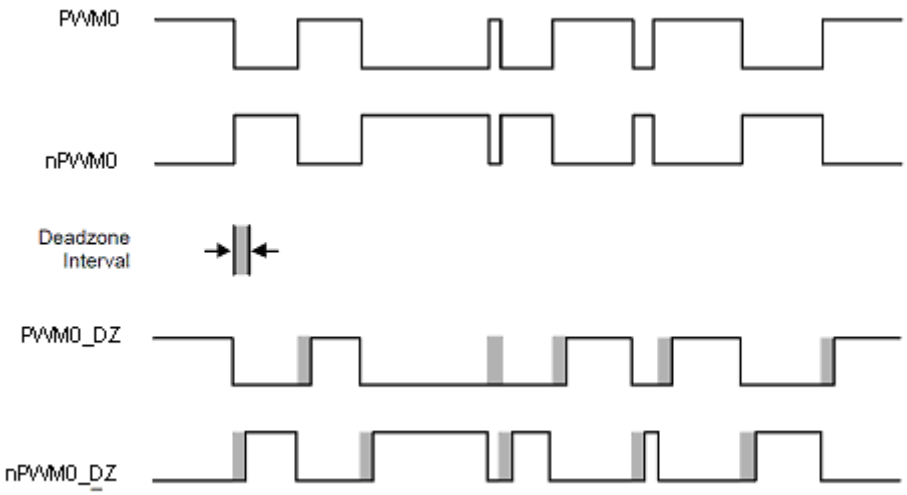


Figure 15-7 : The Wave Form When a Dead Zone Feature is Enabled

15.8 Dead Zone 應用

PWM Dead-Zone功能大部分被使用在開關式電源驅動上，表示如下圖:

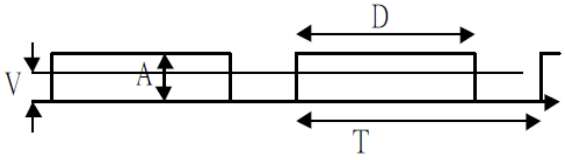


Figure 15-8

1. PWM 輸出只具有 ON/OFF 兩種狀態，電壓(A)是最大電壓在ON 狀態。電壓在OFF 狀態是0。
2. 如果週期為T，ON 的時間為D，那麼 PWM 平均電壓V = (D/T)*A。換句話說，我們可以產生任何電壓範圍 0 ~ A。
3. 在切換頻率為低速時，它會引起馬達震動或線路的高頻噪聲。一般而言，合理開關頻率為4KHz~8KHz。
4. 馬達的特性為低通濾波器，太高的頻率馬達不會產生動作。所以如果開關頻率在合理範圍，那麼工作起來就像線性放大器。

一個很簡單的負載，PWM 切換是電路的方塊如下列所示:

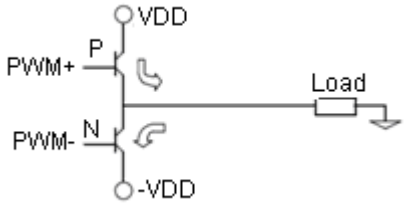


Figure 15-9

1. 使用兩個電晶體控制正電源與負電源，根據使用者需求可以選擇適合的電晶體工作組態。
2. PWM 訊號具有正與負兩種狀態。PWM+ 控制正電源導通或不導通，PWM- 控制負電源導通或不導通。
3. 切換式的各種可能狀態如下:

   P=OFF / N=OFF：separate Load & power.
   P=ON / N=OFF：Load connects to positive power.
   P=OFF / N=ON：Load connects to negative power.
   P=ON / N=ON：short power & burn out power driver.

基本上，PWM+ 與 PWM- 是互為反相的，是不可能同時被開啟的。但是考慮到power MOS 的傳輸反應，與電晶體關閉響應時間大於電晶體開啟的時間，因此有可能會有同時導通的狀況，這會產生以下結果:

1) 長時間導通導致驅動晶體燒毀
2) 短時間導通也許驅動晶體不會立即燒毀，但是經過長時間重覆的執行後會產生累積性的熱燒毀。

所以PWM控制電路必須避免以上情況。

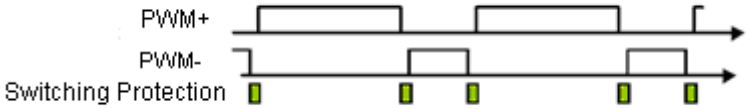


Figure 15-10

4. 基本上，PWM- 是 PWM+ 的反相。
5. 在切換的時間，必須要兩個晶體同時為OFF 的狀態，這根據不同的驅動晶體特性，可能需要1us ~ 4us。

16. 串列匯流排單元

16.1 SPI Master 單元

RA8889 在 SPI 傳輸資料中，資料是可以以同時傳送與接收。串列時脈 [SCK] 針對兩條串列資料線會同步移位與取樣，master 會在 clock edge 前半個週期放置相關資訊以供 slave 裝置參考抓取資料。在 SPI 的控制暫存器上，CPOL 與 CPHA 有 4 種可能的協定方式可供選擇，而 master 與 slave 裝置必須操作在同一個時脈之下。

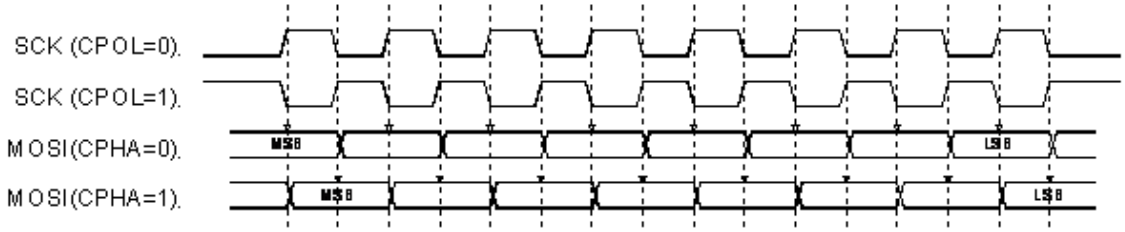


Figure 16-1

Transmitting data bytes

在程式化控制暫存器後，SPI 傳輸可以被初始化。初始化傳送資料的方式是將資料寫入 [SPIDR] 暫存器中，寫入 SPIDR 的資料實際上是寫入具有 16 個深度的 FIFO 被稱為 Write FIFO。每個寫入的資料會增加 Write FIFO 的 data byte。當 SS_ACTIVE 被設為 1 並且 FIFO 不是空的情況下，RA8889 會將最先寫入 Write FIFO 的資料開始傳輸給 Slave。

Receiving data bytes

接收資料與傳送資料是同時產生的。每當傳送一筆資料就會一筆資料被接收到。因此對每筆要接收的資料，都必須寫下空週期到 Write FIFO 中，這會產生 SPI 傳輸的動作，也就是傳輸空週期的同時也會接收資料。 每當傳輸結束時，接收到的資料會被放在 Read FIFO 中。Read FIFO 與 Write FIFO 是相對應的，也是一個獨立具有 16 個深度的 FIFO。FIFO 內容可以從 [SPDR] 暫存器中讀到。

FIFO Overrun

無論是 Write FIFO 還是 Read FIFO 都是使用 circular memories 去模擬一個無限制的大記憶體。當在 FIFO 已經滿的情況下，再寫入 FIFO 的資料將會覆蓋掉最舊的資料。經由[SPDR]暫存器寫入 Write FIFO 如果造成 Overflow 的話，就會造成資料的錯誤，那麼使用 SPI 介面傳輸的就不會是最早輸入的資料，而是最後進入 FIFO 的資料。

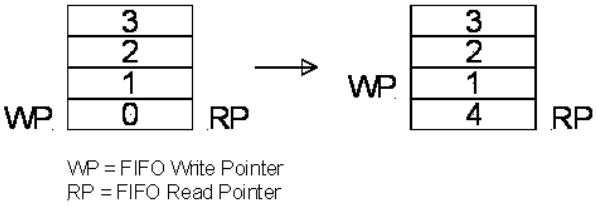


Figure 16-2

只有一種方法可以重置 Write 緩衝區。若是將 [SS_ACTIVE] 清為 0，無論是 Read FIFO 還是 Write FIFO 都會被重置。Read FIFO overruns 可能會有微小的傷害，特別是 SPI bus 只被使用在傳輸資料上，如傳輸資料給 DAC。那麼同時接收到資料可以被忽略，事實上 Read FIFO overruns 是無關緊要的。如果 SPI 被使用在要傳送與接收資料，那麼 Read FIFO 對齊就是很重要的，計算目前 RFIFO 的資料深度的方法是 dummy read 的數目等於已傳送 transmitted 除以 16 取餘數。

註：如果在 Read FIFO 沒有空的情況下，儲存 16 筆資料必定會造成 overwritten，因此在每接收 16 筆資料之前必須要確認 Read FIFO 是不是空的。

RA8889 支援兩種串列快閃位元 bus0 (xsck, xmosi, xmiso, xmsio2, xmsio3)以及 bus1 (xspi1_sck, xspi1_msio0, xspi1_msio1, xspi1_msio2, xpsi1_msio3). 在接收或是傳輸資料之前，使用者應該先設定暫存器 [C5h] Bit7(SPI master bus select) 來選擇串列快閃位元 bus0(xsck, xmosi, xmiso, xmsio2, xmsio3) 或 bus1 (xspi1_sck, xspi1_msio0, xspi1_msio1, xspi1_msio2, xpsi1_msio3)

```
REG_WR ('hC5, 8'h00); //Select bus0, rx register rising edge latch data
REG_WR ('hBB, 8'h1f); //Divisor, configure SPI clock frequency
REG_WR ('hB9, 8'b0001_1111); // {1'b0, mask, nSS_sel, ss_active, ovfirqen, emtirqen, cpol, cpha}, nSS low
REG_WR ('hB8, 8'h55); // TX
REG_WR ('hB8, 8'haa); // TX
REG_WR ('hB8, 8'h87); // TX
REG_WR ('hB8, 8'h78); // TX
wait (xintr);
REG_RD ('hBA, acc);
while (acc != 8'h84) begin
      $display ("wait for FIFO empty ...");
      REG_RD ('hBA, acc);
end
REG_WR ('hBA, 8'h04); // clear interrupt flag
REG_RD ('hB8, 8'h55); // RX
REG_RD ('hB8, 8'haa); // RX
REG_RD ('hB8, 8'h87); // RX
REG_RD ('hB8, 8'h78); // RX
REG_WR ('hB9, 8'b0000_1111); // {1'b0, mask, nSS_sel, ss_active, ovfirqen, emtirqen, cpol, cpha}, nSS high.
```

16.2 串列快閃記憶體控制單元

RA8889 內建 SPI master 介面，此功能主要是為了使用外部快閃記憶體/ROM，支援的協定有 4-BUS (Normal Read)、5-BUS (FAST Read)、Dual mode 0、Dual mode 1 與 Mode 0/Mode 3。快閃記憶體/ROM 功能可以被文字模式與 DMA 模式使用。文字模式表示外部的快閃記憶體/ROM 儲存的是文字的 bitmap 圖檔。RA8889 支援上海的專業字元廠商-集通公司通用的字元。DMA 模式表示外部的快閃記憶體儲存的是 DMA (Direct Memory Access) 的資料，通常是儲存圖檔，使用者可以使用 DMA 加快顯示資料由快閃記憶體傳送至顯示記憶體中，而這個處理過程不需要 MPU 的處理。

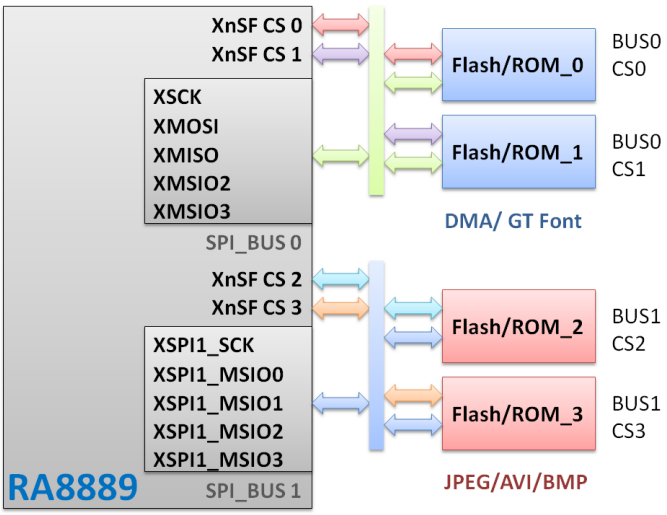


Figure 16-3 : RA8889 Serial Flash/ROM System

關於快閃記憶體/ROM 的讀取命令協定，請參考下表:

Table 16-1 : Read Command Code & Behavior Selection

REG [B7h] BIT[3:0]	Read Command code
000xb	1x **讀取命令碼**– 03h 預設讀取速度，快閃記憶體至 RA8889 的資料輸入為 xmiso 腳位。 在位址與資料間不需要空週期。
010xb	1x **讀取命令碼**– 0Bh 為快速讀取速度(fast read)，快閃記憶體至 RA8889 的資料輸入為 xmiso 腳位。 在位址與資料間會有 8 個空週期。
1x0xb	**1x 讀取命令碼 – 1Bh** 為高速讀取速度，快閃記憶體至 RA8889 的資料輸入為 xmiso 腳位。 再位址與資料間會有 16 個空週期。
xx10b	2x **讀取命令碼**– 3Bh 快閃記憶體至 RA8889 資料輸入方式為交錯輸入，其輸入腳位為 xmiso 與 xmosi。 在位址與資料間會有 8 個空週期(mode 0)。

REG [B6h] BIT[7:6]	Read Command code
01b	4x 讀取命令碼 – 6Bh. RA8889 的位址輸出與資料輸入皆為交錯式，其使用的輸出輸入腳位為 xmiso 與 xmosi 與 xsio2 與 xsio3。
10b	4x 讀取命令碼 – EBh. RA8889 的位址輸出與資料輸入皆為交錯式，其使用的輸出輸入腳位為 xmiso 與 xmosi 與 xsio2 與 xsio3。

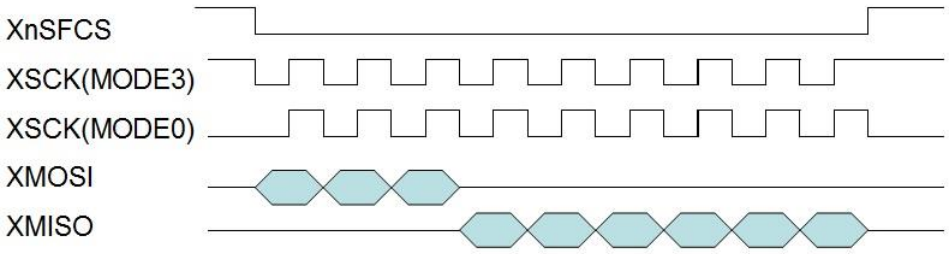
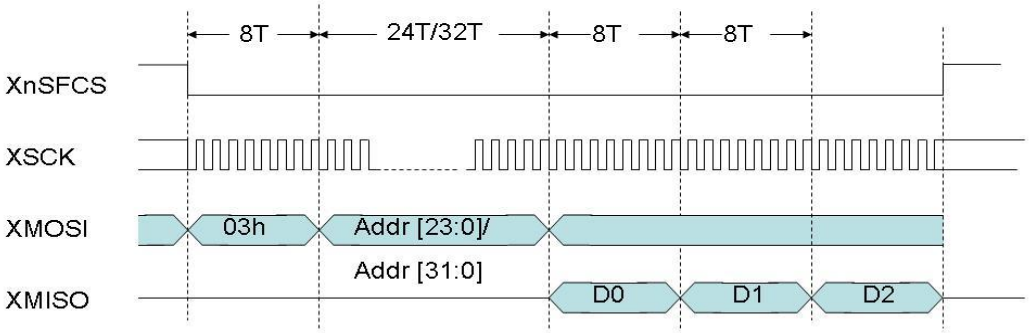


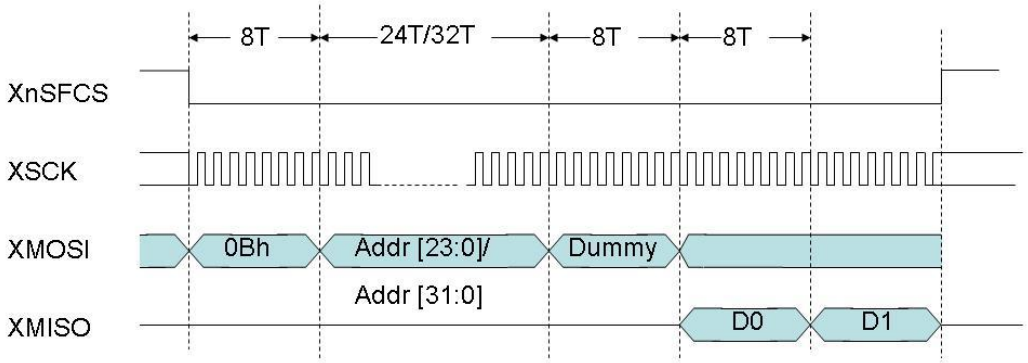
Figure 16-4 : Mode 0 and Mode 3 Protocol



If REG[B7h] Bit 5 set to 0, Then Addr state will be 24T

If REG[B7h] Bit 5 set to 1, Then Addr state will be 32T

Figure 16-5 : Normal Read Command



If REG[B7h] Bit 5 set to 0, Then Addr state will be 24T

If REG[B7h] Bit 5 set to 1, Then Addr state will be 32T

Figure 16-6 : Fast Read Command

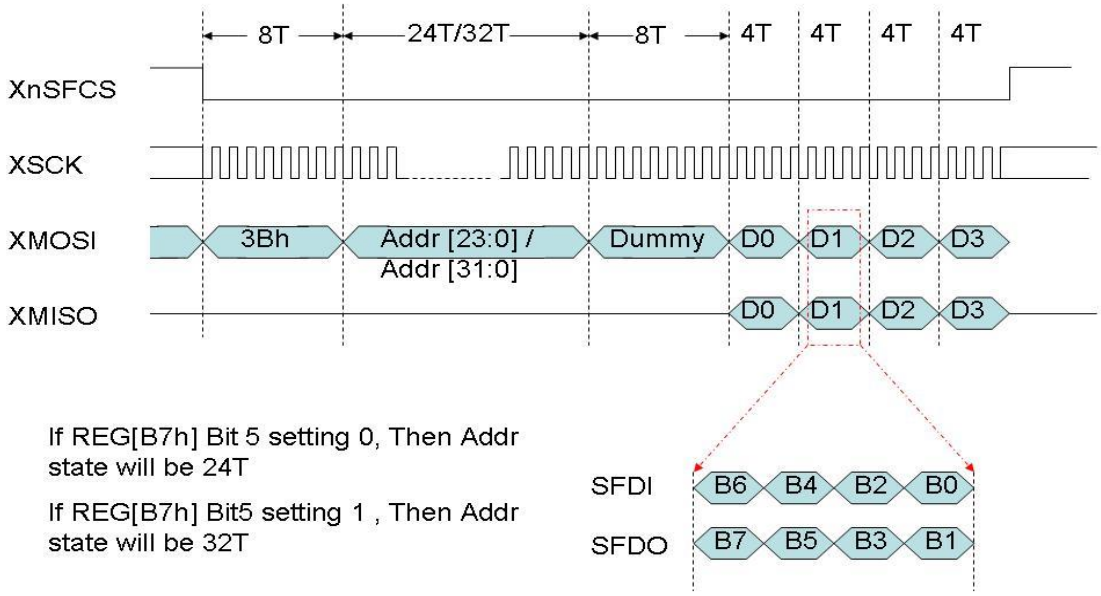
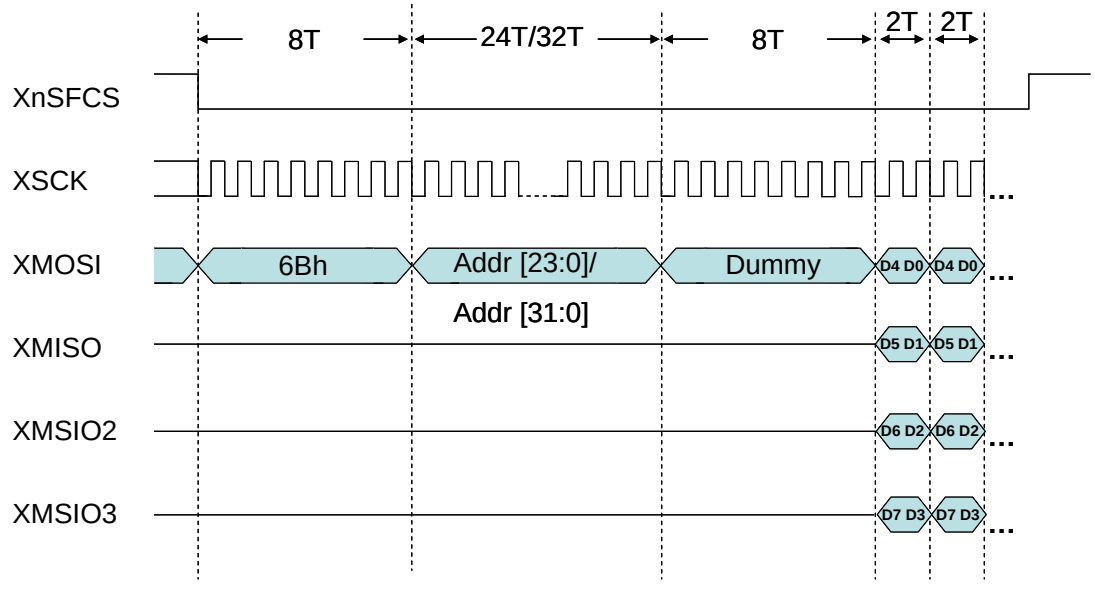


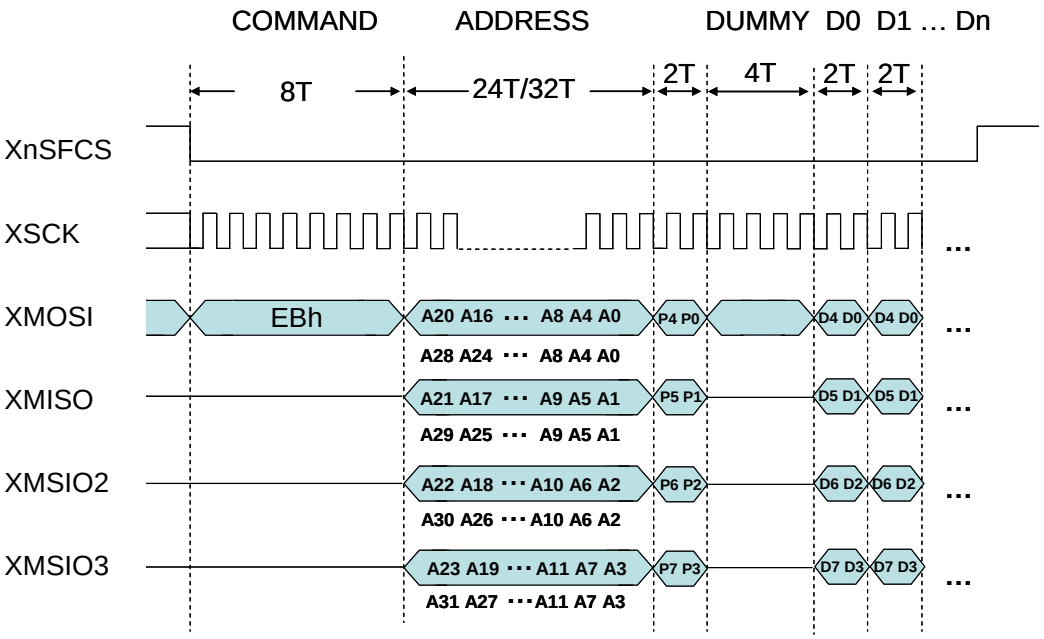
Figure 16-7 : Dual Output Read Command Mode 0



If REG[B7h] Bit 5 set to 0, Then Addr state will be 24T

If REG[B7h] Bit 5 set to 1, Then Addr state will be 32T

Figure 16-8 : Quad Mode Read (6B)



If REG[B7h] Bit 5 set to 0, Then Addr state will be 24T

If REG[B7h] Bit 5 set to 1, Then Addr state will be 32T

Figure 16-9 : Quad Mode Read (EB)

16.2.1 SPI Master 初始化

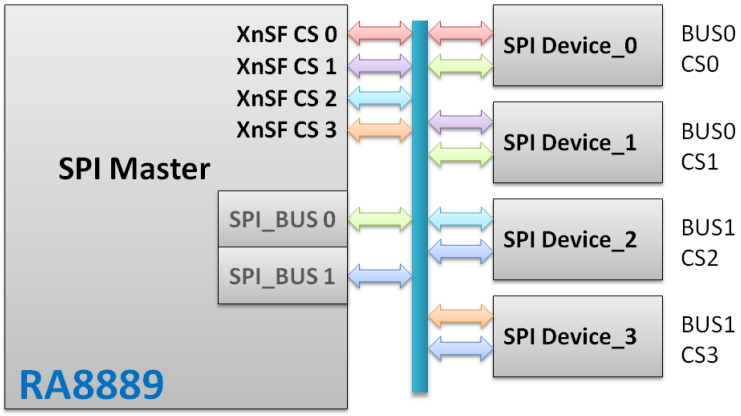


Figure 16-10

16.2.2 外部串列字元 ROM

RA8889 經由支援集通外部字元 ROM，可以將多種字元寫入顯示記憶體。而 RA8889 相容集通字元 ROM 的型號如下 GT21L16T1W/GT21H16T1W, GT30L16U2W, GT30L24T3Y/GT30H24T3Y, GT30L24M1Z, 與 GT30L32S4W/GT30H32S4W。這些字型包含了 16X16、24X24、32X32 與不等寬的字元大小。

外部字元的字元碼包含了 3 種不同的編碼方式，1 byte/2bytes/4bytes 的編碼方式，說明如下:

1. 1byte 字元碼– ASCII code for all Character ROMs。
2. 2~4bytes 元碼–如 GT30L24M1Z 的 GB18030 的編碼方式。
3. 2bytes 字元碼+2bytes 索引碼– 只有被 GT30L16U2W 的 Uni-code 使用。
4. 其他字元碼的長度皆為 2bytes。

在使用外部字元 ROM 時，使用者首先必須要了解編碼規則。而關於詳細的字元碼與字型對應方式，請詢問集通公司。

請注意 GT30L16U2W 規格書，uni-code 字元碼需要另外參考 “ZFindex Table” 來計算 ROM 的字元位址。如果使用者輸入 UNI-CODE 的字元碼範圍為 00A1h~33D5h 或 E76Ch~FFE5h，這是一個特殊的編碼範圍，需要額外的 2bytes 字元碼 (high byte first) 來參考到 “ZFindex table” 並計算字元位址。其他 UNICODE 編碼範圍只需要兩個字元碼。關於更詳細的說明，請參考 GT30L16U2W 規格書。

例子: 如果使用者使用 GT30L16U2W 並輸入 UNI-CODE 字元碼 (00A2)，因為其範圍為在 00A1h~33D5h 之間，然後 MPU 必須寫入額外的 2 bytes 以供索引 ZFindex table 給 R8889 計算確實的字元位址。

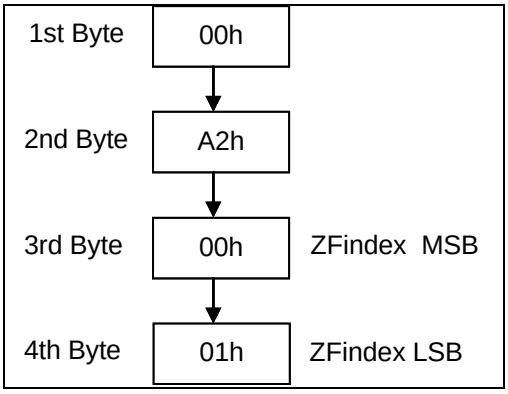


Figure 16-11 : Uni-Code Zfindex

RA8889 具外部字元 ROM 的時脈速度選擇暫存器，這可以提供使用者可以調整速度來符合外部 ROM 的時序。寫入文字的程式流程圖如下圖:

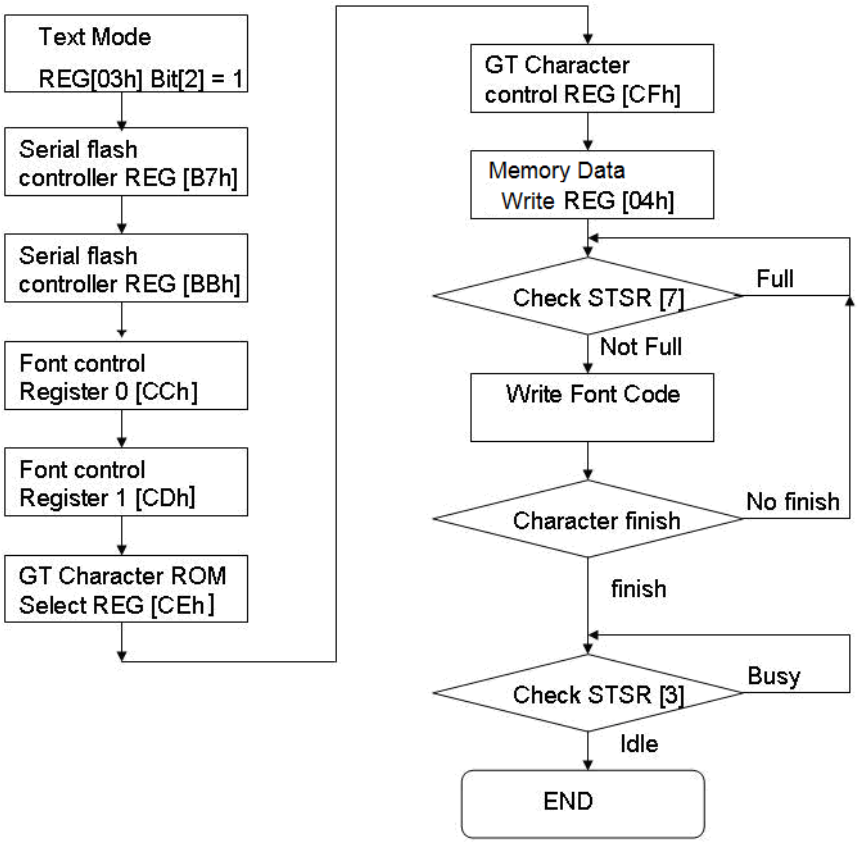


Figure 16-12 : External Character ROM Programming Procedure

16.2.3 外部串列資料 ROM

外部快閃記憶體/ROM 可以被視為圖檔來源，那麼就可以在圖形模式下使用 DMA (Direct Memory Access) 存取。串列快閃記憶體/ROM 可以當作是 DMA 功能的來源端，而快閃記憶體/ROM 可被視為大量儲存媒體。串列快閃記憶體/ROM 的內容格式必須跟 SDRAM 的格式一致。快閃記憶體/ROM 圖形格式如下:

8bpp data

Addr	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0001h	R_1^7	R_1^6	R_1^5	G_1^7	G_1^6	G_1^5	B_1^7	B_1^6	0000h	R_0^7	R_0^6	R_0^5	G_0^7	G_0^6	G_0^5	B_0^7	B_0^6
0003h	R_3^7	R_3^6	R_3^5	G_3^7	G_3^6	G_3^5	B_3^7	B_3^6	0002h	R_2^7	R_2^6	R_2^5	G_2^7	G_2^6	G_2^5	B_2^7	B_2^6
0005h	R_5^7	R_5^6	R_5^5	G_5^7	G_5^6	G_5^5	B_5^7	B_5^6	0004h	R_4^7	R_4^6	R_4^5	G_4^7	G_4^6	G_4^5	B_4^7	B_4^6
0007h	R_7^7	R_7^6	R_7^5	G_7^7	G_7^6	G_7^5	B_7^7	B_7^6	0006h	R_6^7	R_6^6	R_6^5	G_6^7	G_6^6	G_6^5	B_6^7	B_6^6
0009h	R_9^7	R_9^6	R_9^5	G_9^7	G_9^6	G_9^5	B_9^7	B_9^6	0008h	R_8^7	R_8^6	R_8^5	G_8^7	G_8^6	G_8^5	B_8^7	B_8^6
000Bh	R_{11}^7	R_{11}^6	R_{11}^5	G_{11}^7	G_{11}^6	G_{11}^5	B_{10}^7	B_{10}^6	000Ah	R_{10}^7	R_{10}^6	R_{10}^5	G_{10}^7	G_{10}^6	G_{10}^5	B_{10}^7	B_{10}^6

16bpp data

Addr	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0001h	R_0^7	R_0^6	R_0^5	R_0^4	R_0^3	G_0^7	G_0^6	G_0^5	0000h	G_0^4	G_0^3	G_0^2	B_0^7	B_0^6	B_0^5	B_0^4	B_0^3
0003h	R_1^7	R_1^6	R_1^5	R_1^4	R_1^3	G_1^7	G_1^6	G_1^5	0002h	G_1^4	G_1^3	G_1^2	B_1^7	B_1^6	B_1^5	B_1^4	B_1^3
0005h	R_2^7	R_2^6	R_2^5	R_2^4	R_2^3	G_2^7	G_2^6	G_2^5	0004h	G_2^4	G_2^3	G_2^2	B_2^7	B_2^6	B_2^5	B_2^4	B_2^3
0007h	R_3^7	R_3^6	R_3^5	R_3^4	R_3^3	G_3^7	G_3^6	G_3^5	0006h	G_3^4	G_3^3	G_3^2	B_3^7	B_3^6	B_3^5	B_3^4	B_3^3
0009h	R_4^7	R_4^6	R_4^5	R_4^4	R_4^3	G_4^7	G_4^6	G_4^5	0008h	G_4^4	G_4^3	G_4^2	B_4^7	B_4^6	B_4^5	B_4^4	B_4^3
000Bh	R_5^7	R_5^6	R_5^5	R_5^4	R_5^3	G_5^7	G_5^6	G_5^5	000Ah	G_5^4	G_5^3	G_5^2	B_5^7	B_5^6	B_5^5	B_5^4	B_5^3

24bpp data

Addr	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0001h	G_0^7	G_0^6	G_0^5	G_0^4	G_0^3	G_0^2	G_0^1	G_0^0	0000h	B_0^7	B_0^6	B_0^5	B_0^4	B_0^3	B_0^2	B_0^1	B_0^0
0003h	B_1^7	B_1^6	B_1^5	B_1^4	B_1^3	B_1^2	B_1^1	B_1^0	0002h	R_0^7	R_0^6	R_0^5	R_0^4	R_0^3	R_0^2	R_0^1	R_0^0
0005h	R_1^7	R_1^6	R_1^5	R_1^4	R_1^3	R_1^2	R_1^1	R_1^0	0004h	G_1^7	G_1^6	G_1^5	G_1^4	G_1^3	G_1^2	G_1^1	G_1^0
0007h	G_2^7	G_2^6	G_2^5	G_2^4	G_2^3	G_2^2	G_2^1	G_2^0	0006h	B_2^7	B_2^6	B_2^5	B_2^4	B_2^3	B_2^2	B_2^1	B_2^0
0009h	B_3^7	B_3^6	B_3^5	B_3^4	B_3^3	B_3^2	B_3^1	B_3^0	0008h	R_2^7	R_2^6	R_2^5	R_2^4	R_2^3	R_2^2	R_2^1	R_2^0
000Bh	R_3^7	R_3^6	R_3^5	R_3^4	R_3^3	R_3^2	R_3^1	R_3^0	000Ah	G_3^7	G_3^6	G_3^5	G_3^4	G_3^3	G_3^2	G_3^1	G_3^0

DMA 提供使用者可以快速的更新與傳送大量資料致顯示記憶體中。在 RA8889 中 DMA 的唯一來源就是外部的快閃記憶體/ROM。對於 DMA 有兩種傳送資料的方式，linear 模式與 block 模式。這可以提供使用者根據應用彈性選擇。而 DMA 傳送目的是在顯示記憶體的工作視窗內，傳送的方法為一個 byte 一個 byte 的將資料由外部快閃記憶體/ROM 傳送至顯示記憶體中。在 DMA 完成傳送後，RA8889 會發出一個中斷以提醒主控端。關於詳細的操作，請參考下列章節。

16.2.3.1 線性模式下的直接記憶體存取外部串列資料 ROM

DMA linear 模式被使用在將串列快閃記憶體的 CGRAM 傳送給 SDRAM，而此時工作視窗的色深必須設定是 8bpp，請參考 Figure 16-13。

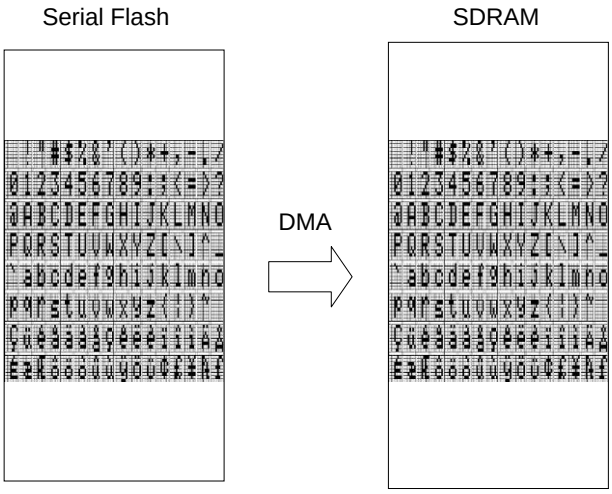


Figure 16-13

16.2.3.2 區塊模式下的直接記憶體存取外部串列資料 ROM

區塊模式的直接記憶體存取主要是被使用再傳送圖形資料，這個處理的基本單位是以 Pixel 為基本單位，請參考下面的程式流程圖:

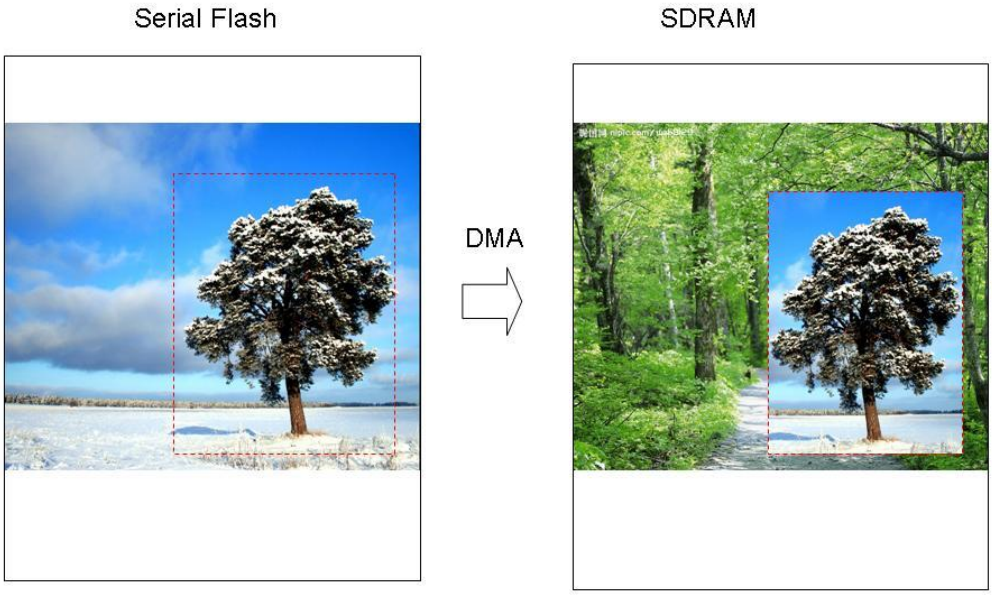


Figure 16-14 : DMA Function

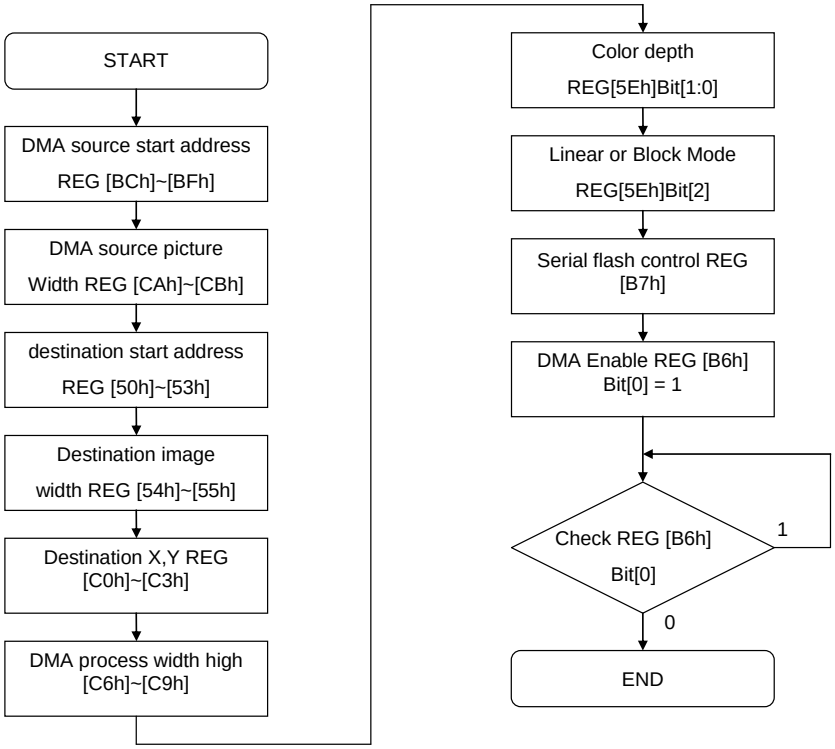


Figure 16-15 : Enable DMA Procedure – Check Flag

16.2.3.3 IDEC 功能

IDEC 功能主要是為了支援媒體解碼 MDU，使用者若要使用 MDU 必須設定 IDEC 相關暫存器，其 IDEC 串列快閃記憶體控制僅支援 Quad mode。相關設定請參考 Chatper 18 媒體解碼單元設定 flow-chart。

REG[03h] Bit[7] = 0

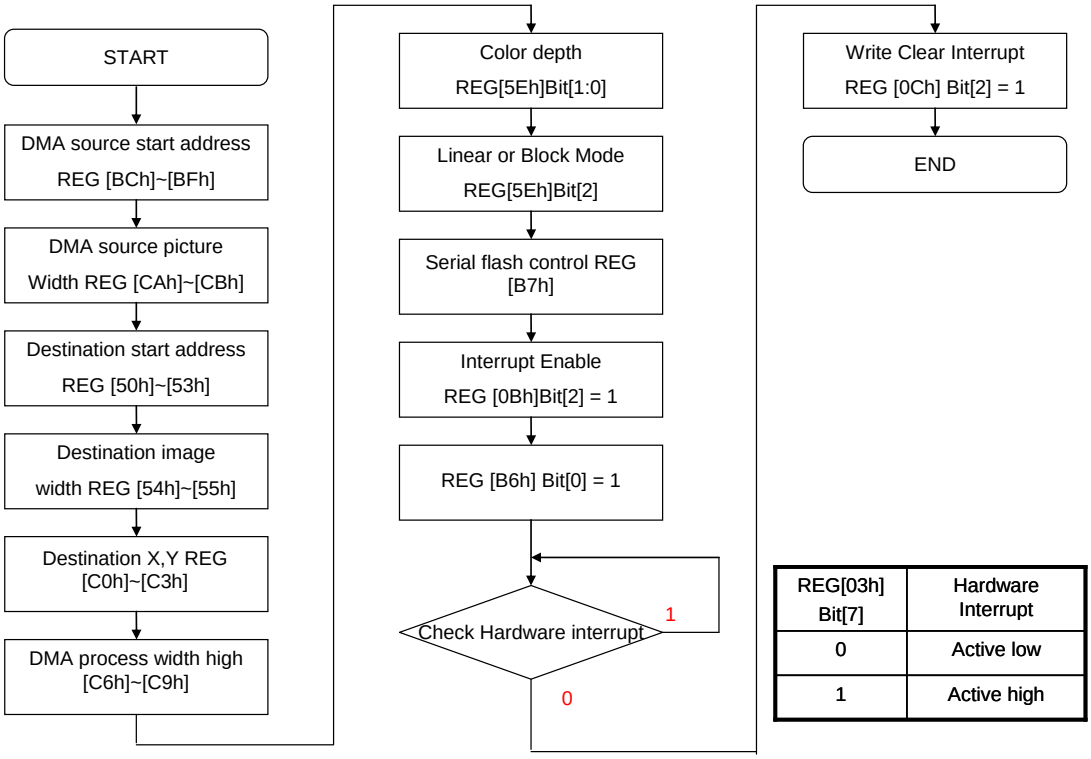
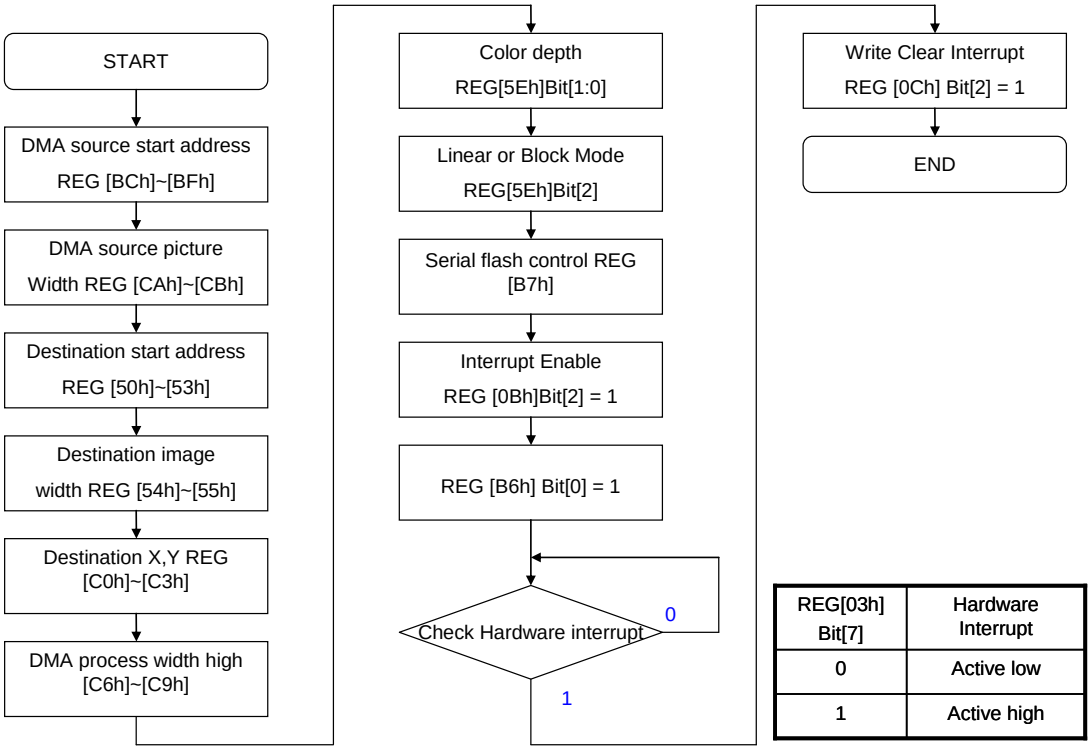


Figure 16-16 : DMA Enable Procedure – Check Hardware Interrupt pin -1

REG[03h] Bit[7] = 1



REG[03h] Bit[7]	Hardware Interrupt
0	Active low
1	Active high

Figure 16-17 : DMA Enable Procedure –Check Hardware Interrupt pin -2

16.3 IIC Master 單元

IIC Master 是雙線雙向串列介面，這提供簡單而有效的方法與裝置交換資料。只支援 100K bps 與 400K bps 模式。下面是 IIC Master XSCL 速度公式:

$$XSCL = CCLK / (5*(Pre\text{-}scale + 2))$$

舉例: 如果 XSCL 是 100 KHz 並且 CCLK 是 100 MHz，那麼 pre-scalar (REG[E5h] & REG[E6h]) 就必須設為 200。在 Master 與 Slave 間的資料傳輸是經由 XSCL 達成同步的，以 Bytes 為單位。每個資料 byte 是 8-bit，對每個 XSDA bit 都有相對應的一個 XSCL，並且傳輸時由 MSB 開始傳輸，在每個 byte 後面會有一個 acknowledge bit 傳送。每個 bit 都是在 XSCL 為高電位實被處理，因此 XSDA 只能在 XSCL 低電位時變化，並且 XSDA 必須在 XSCL 為高電位時是穩定不變化的。

一個標準化的 IIC 通訊協定具有 4 個部份的組成:

1. Start signal
2. Slave address transfer
3. Data transfer
4. STOP signal

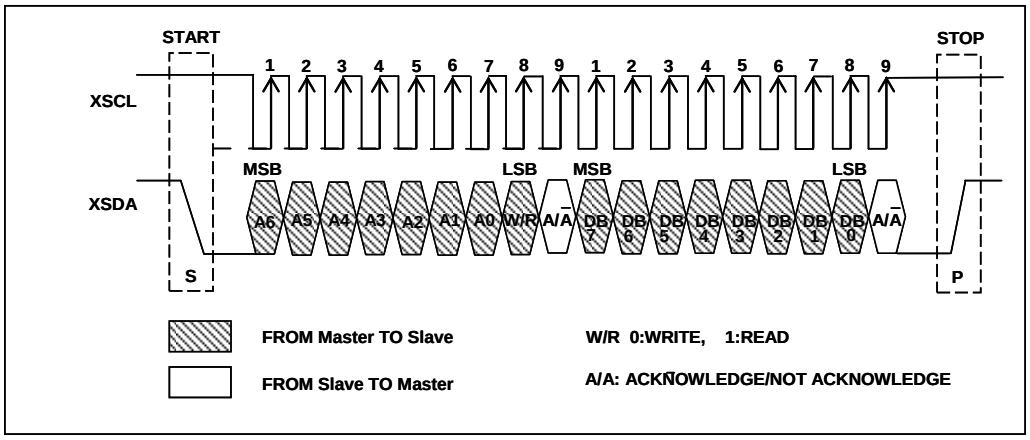


Figure 16-18

例 1. 寫 1 Byte 資料到裝置上

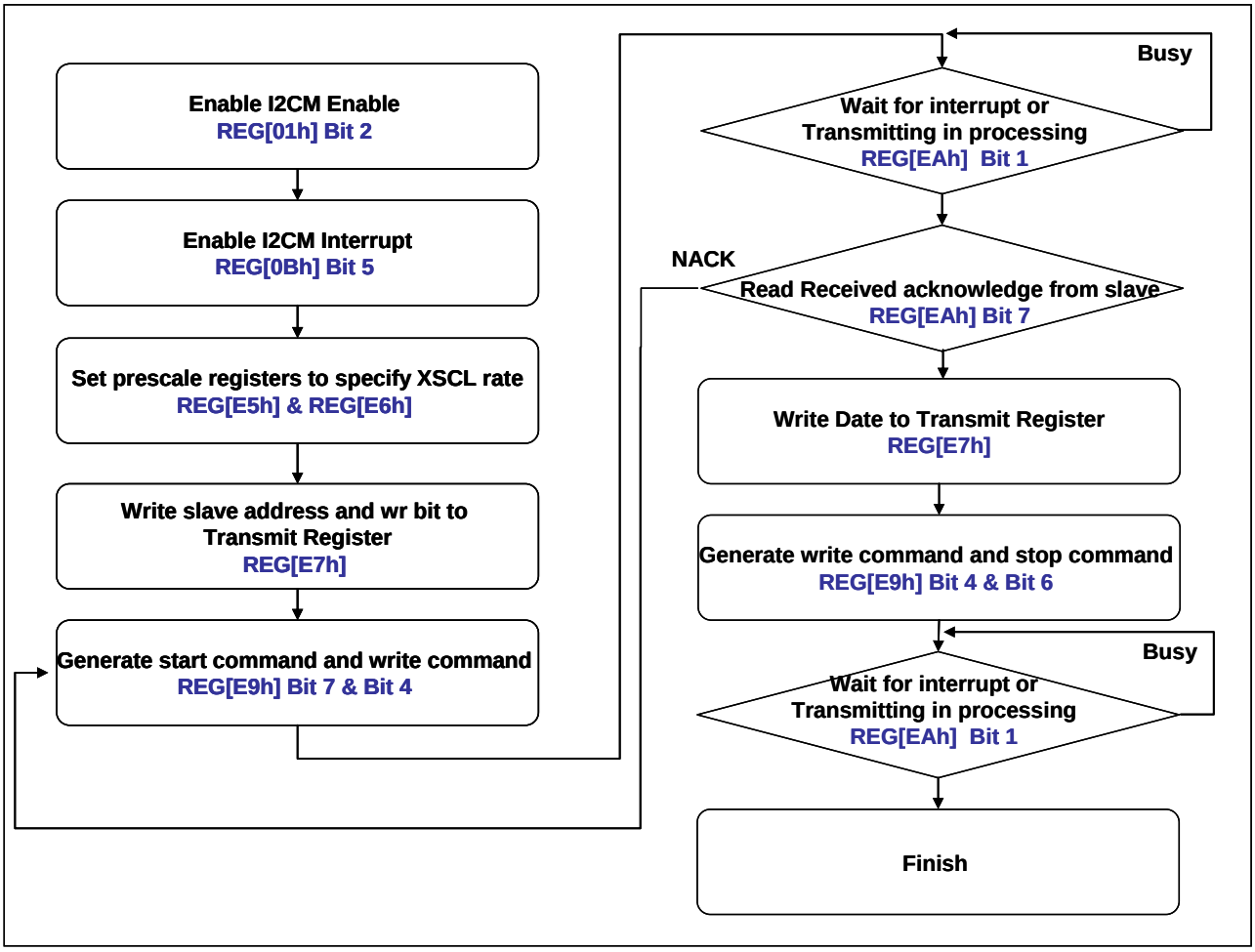


Figure 16-19 : Flow for Write 1 Byte Data to Slave

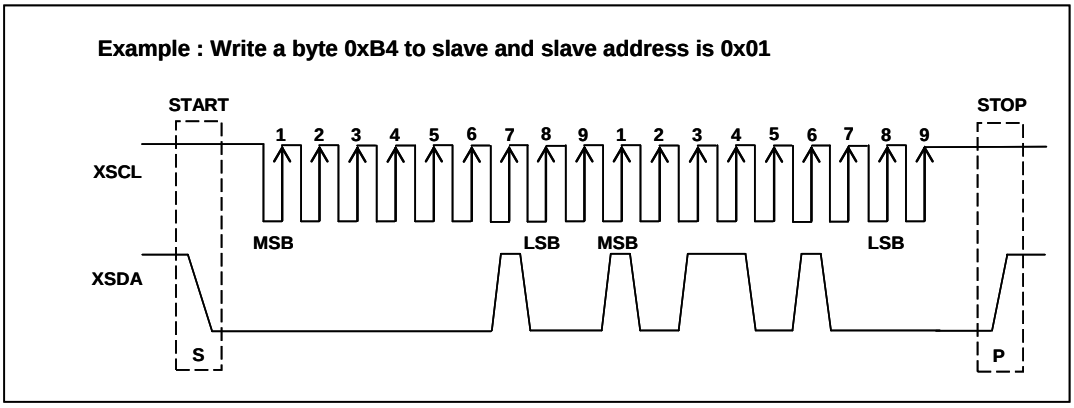


Figure 16-20 : Waveform for Write 1 Byte Data to Slave

例 2. 從裝置上讀 1 Byte 資料

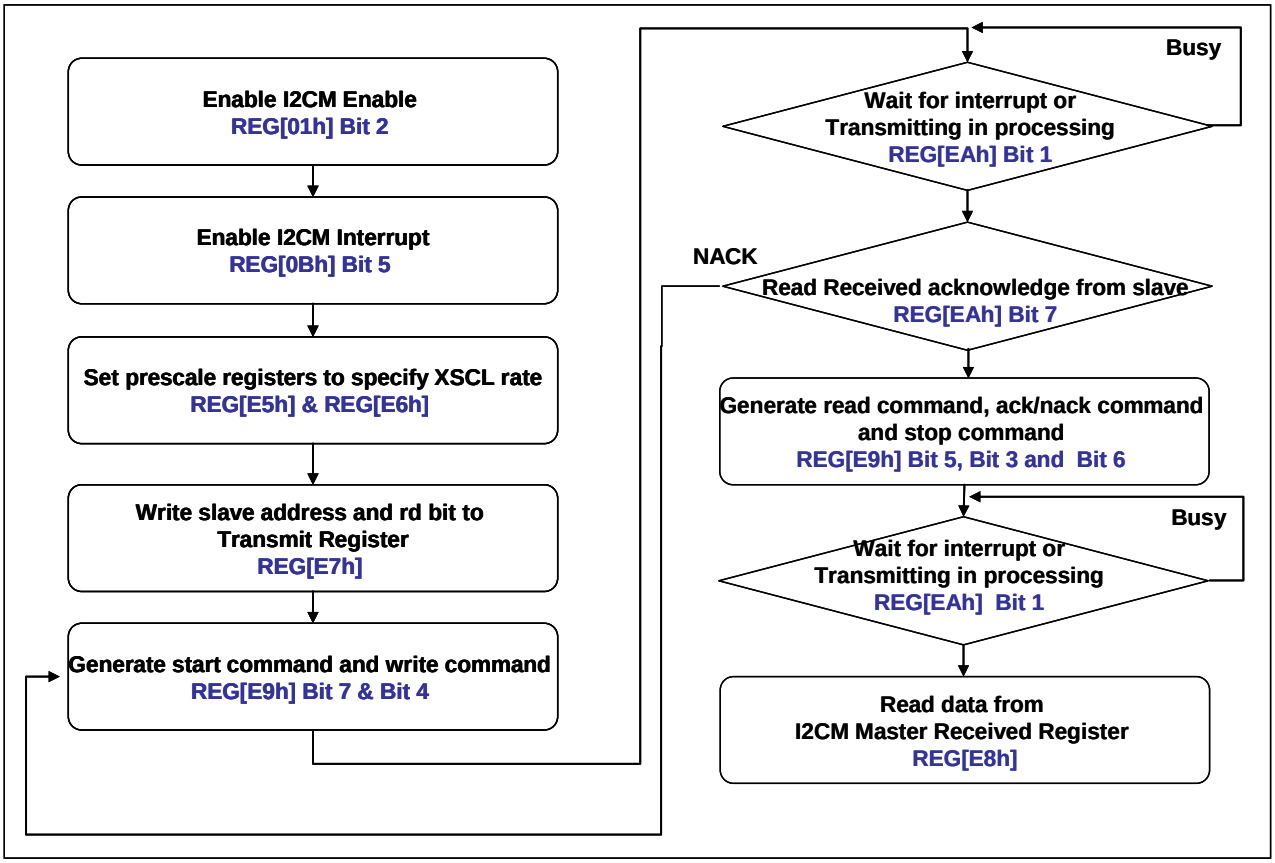


Figure 16-21 : Flow for Read 1 Byte Data from Slave

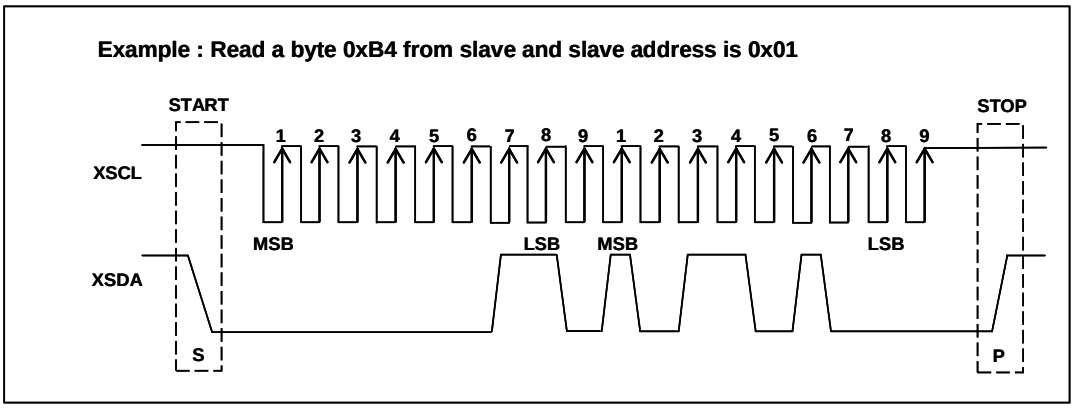


Figure 16-22 : Waveform for Read 1 Byte Data from Slave

17. 鍵盤掃描

鍵盤掃描會掃描並讀取鍵盤狀態，而鍵盤矩陣會由硬體來切換掃描線。這個功能可以提供鍵盤應用，Figure 17-1 顯示的是基本的鍵盤應用電路。RA8889 為因應外部電路需求，已經在 KIN[4:0] 內建提升電阻。

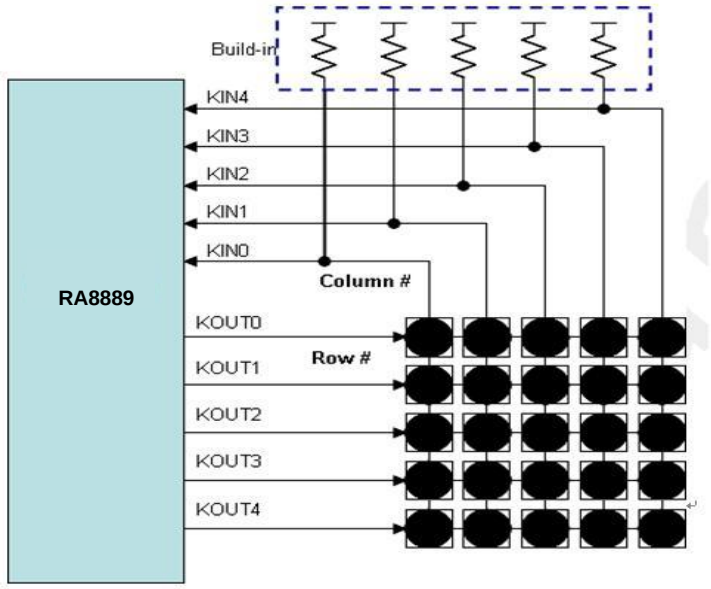


Figure 17-1 : Key-Pad Application

17.1 鍵盤掃描操作方式

RA8889 鍵盤掃描控制器的特點如下:

1. 最多支援 5x5 鍵盤矩陣
2. Key-Scan 具有可程式化的掃描頻率與取樣時間
3. 可調整的長按鍵時間
4. 支持多鍵同時按。注意: 可同時按 2 個按鍵或是要按 3 個按鍵 (但是 3 按鍵組成不能是 90°排列)
5. 可使用按鍵來喚醒系統

KSCR 是鍵盤掃描的狀態暫存器，這個暫存器被使用在了解鍵盤掃描的操作狀態，如取樣時間、取樣頻率、致能長按鍵。而若是有按鍵按下，使用者也可以透過中斷得知。在 KSCR2 bit1~0 紀錄目前按下的按鍵數目。然後使用者可以透過讀取 KSDR 得到按鍵碼。

註： “Normal key” 是在以取樣時間為基礎上有被認知為合格的按下按鍵行為。“Long Key” 則是在長按鍵取樣週期下有被認知為合格的按下按鍵行為。先產生 "Normal Key" 才會產生 “Long Key”，有時在某些應用上需要分開使用。

Table 17-1 是在 ”Nnomal Key” 下鍵碼與鍵盤矩陣的對應，按下的按鍵鍵碼會被存在 KSDR0~2。如果是長時間按下按鍵，則表現出的會是 ”Long Key”，而相關鍵碼在 Table 17-2。

Table 17-1: Key Code Mapping Table (Normal Key)

	Kin0	Kin1	Kin2	Kin3	Kin4
Kout0	00h	01h	02h	03h	04h
Kout1	10h	11h	12h	13h	14h
Kout2	20h	21h	22h	23h	24h
Kout3	30h	31h	32h	33h	34h
Kout4	40h	41h	42h	43h	44h

Table 17-2: Key Code Mapping Table (Long Key)

	Kin0	Kin1	Kin2	Kin3	Kin4
Kout0	80h	81h	82h	83h	84h
Kout1	90h	91h	92h	93h	94h
Kout2	A0h	A1h	A2h	A3h	A4h
Kout3	B0h	B1h	B2h	B3h	B4h
Kout4	C0h	C1h	C2h	C3h	C4h

當按下多鍵時，最多有三個按鍵會被存在 KSDR0，KSDR1 與 KSDR2 三個暫存器中。注意鍵碼儲存的方式與按鍵位置有關或者說與鍵碼有關，而與按鍵順序無關，請參下列例子:

在相同時間按下鍵碼 0x34, 0x00 and 0x22，在 KSDR0~2 儲存方式如下:

KSDR0 = 0x00
KSDR1 = 0x22
KSDR2 = 0x34

以上所提的鍵盤掃描設定介紹如下:

Table 17-3 : Key-Scan Relative Registers

Reg.	Bit_Num	Description	Reference
KSCR1	Bit 6	Long Key Enable bit	REG[FBh]
	Bit [5:4]	Key-Scan sampling times setting	
	Bit [2:0]	Key-Scan scan frequency setting	
KSCR2	Bit [7]	Key-Scan Wakeup Function Enable Bit	REG[FCh]
	Bit [3:2]	long key timing adjustment	
	Bit [1:0]	The number of key hit	
KSDR0 KSDR1 KSDR2	Bit [7:0]	Key code for pressed key	REG[FDh ~ FFh]
CCR	Bit 5	Key-Scan enable bit	REG[01h]
INTR	Bit 4	Key-Scan interrupt enable	REG[0Bh]
INTC2	Bit 4	Key-Scan Interrupt Status bit	REG[0Ch]

致能鍵盤掃描功能(Key-Scan)，使用者可以使用下列方法檢查按鍵狀態:

1) Software check method: 檢查 Key-scan 的狀態值(status)，來得知是否被按下。

2) Hardware check method: 由中斷來得知是否有按鍵被按下。

若是設定中斷致能 (INTEN bit[3]) 為 1，那麼有鍵盤有被按下時就會產生中斷。而當中斷產生時，Key-scan 中斷狀態旗標 (bit[3] of INTF) 將永遠為 1，無論使用何種方法，使用者在讀取鍵碼後必須清除中斷狀態旗標，否則以後就不會再產生中斷。

此外，RA8889 在省電模式下支援“Key-stroke wakeup ”，經由設定完成後，任何按鍵觸發都可以將 RA8889 由睡眠模式中喚醒。為了檢知喚醒事件，MPU 可以透過軟體程式去輪詢 RA8889 的中斷是否產生。

以上應用的暫存器程式設定流程圖如下:

1. 軟體方法

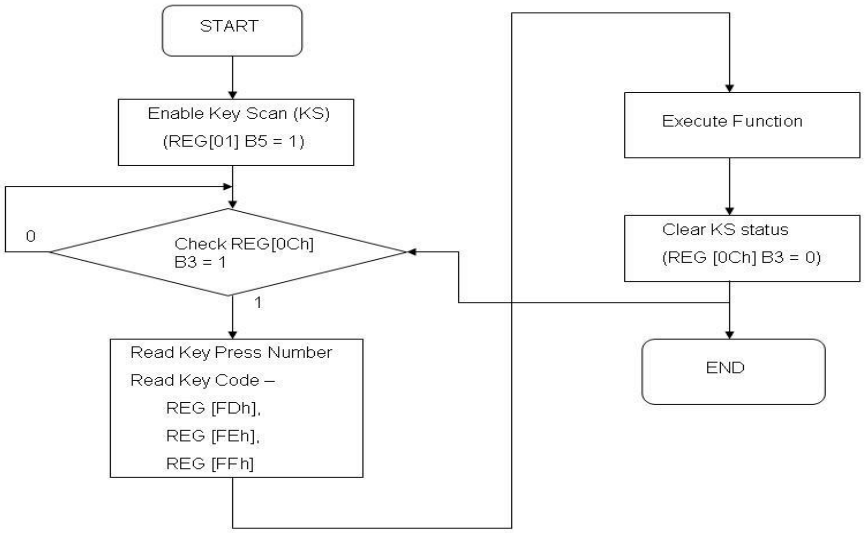


Figure 17-2 : Key-Scan Flowchart for Software Polling

2. 硬體方法

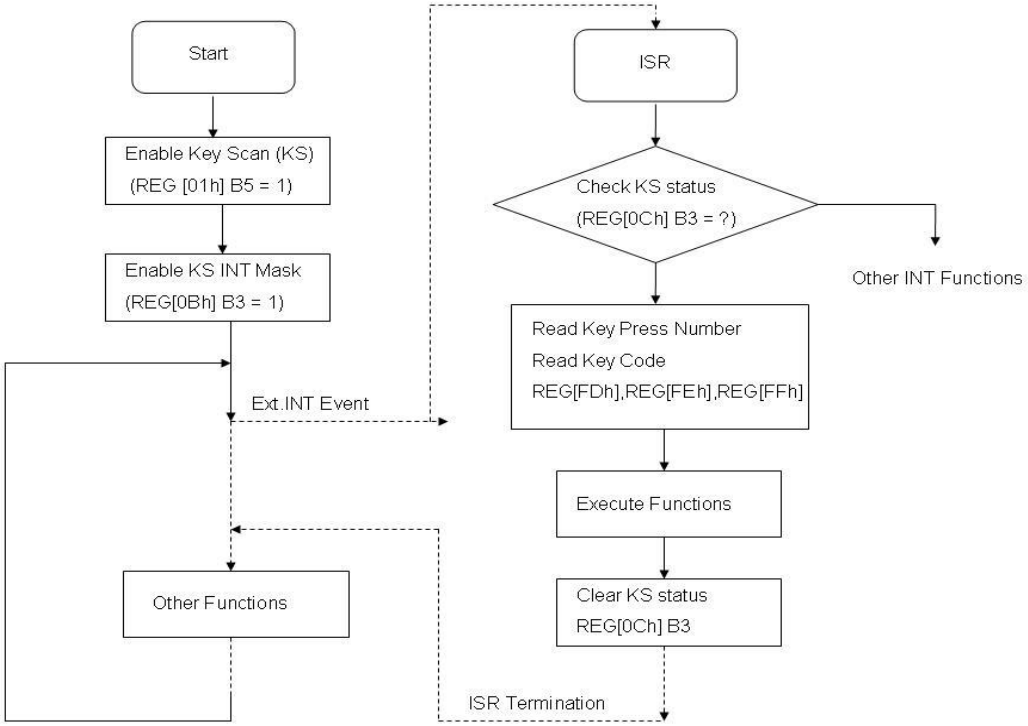


Figure 17-3 : Key-Scan for Hardware Interrupt

17.2 限制

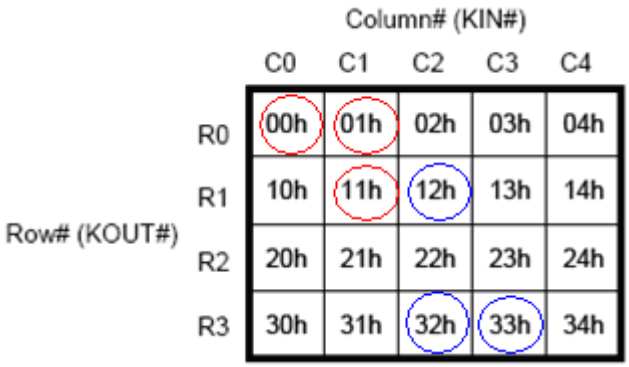


Figure 17-4

如果 3 個按鍵以 90° 的方式壓下，類似上圖的紅色或藍色圓圈，它將會造成錯誤的行為。

18. 媒體解碼單元(MDU)

RA8889 支援媒體解碼單元，包含 JPEG（ISO/IEC 10918-1 Baseline profile，YUV444，YUV422，YUV420，YUV400，並不支援重啟間隔格式），BMP（raw data）和 AVI（motion jpeg）格式。 RA8889 可以自動區分上述三種格式，並將其自動解析為相對的解碼器。 在視頻功能中，RA8889 提供自動播放，暫停和停止功能。使用者應事先將圖像或視頻下載到串列快閃記憶體中，並通過設定 IDEC、CANVAS 和 PIP 相對的暫存器讓他們顯示在 LCD 螢幕上。

圖像寫入的位址，請參考 CANVAS 相關暫存器。 由於視頻顯示在 PIP1 或 PIP2 窗口中，因此使用者應在播放視頻之前設定 PIP 相關暫存器。 此外，RA8889 還提供中斷和忙碌標誌進行檢查。

注意，

1.關於串列快閃存介面，**請使用支援 Quad mode 串列快閃記憶體**，建議核心時脈頻率高於 100MHz。

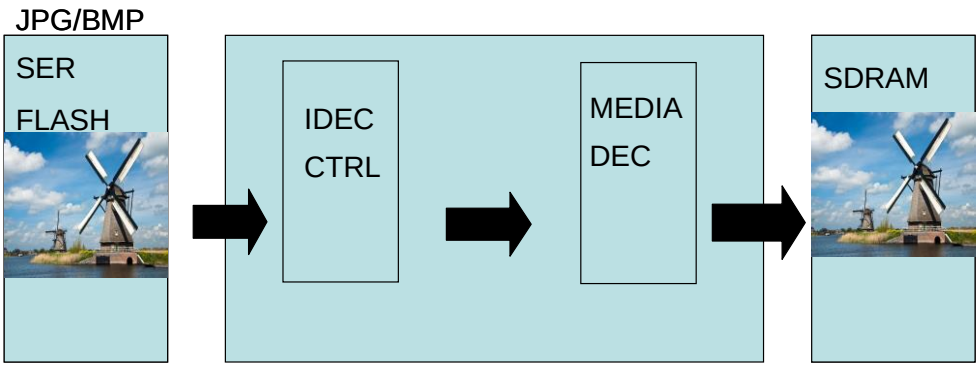
2. AVI frame rate 可以是 30、29.97、25、24、23.97、20 和 15。

3. PIP 的色深應與主要視窗的色深一致。

4. AVI / JPG 的寬度和高度必須是 8 的倍數。

5.串列快閃記憶體的 IDEC 長度應等於圖像或視頻的檔案容量。

18.1 硬體圖像的解碼流程



Reference CANVAS REG for write SDRAM data

Figure 18-1

18.2 圖像解碼流程圖

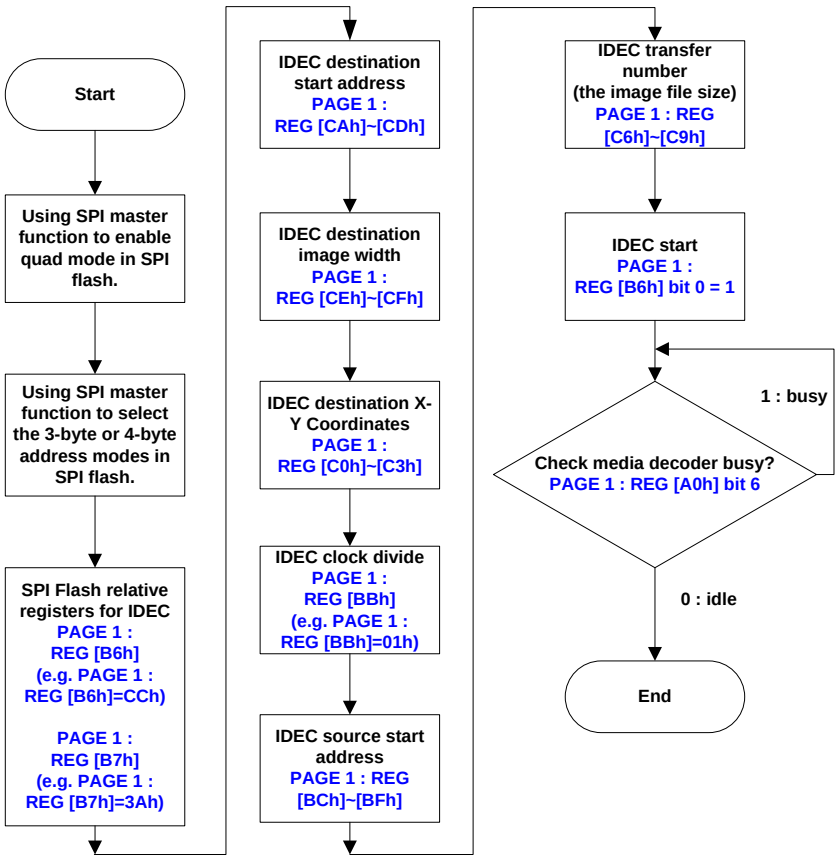
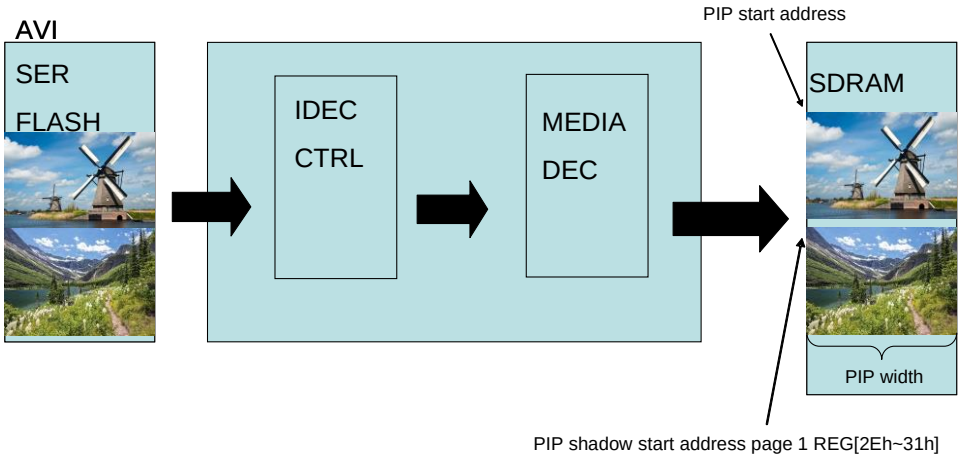


Figure 18-2

註：IDEC 為支援 Media Decoder Unit 的 serial flash 控制介面

18.3 AVI 的解碼流程



Reference PIP REG for write SDRAM data

Figure 18-3

18.4 AVI 解碼流程圖

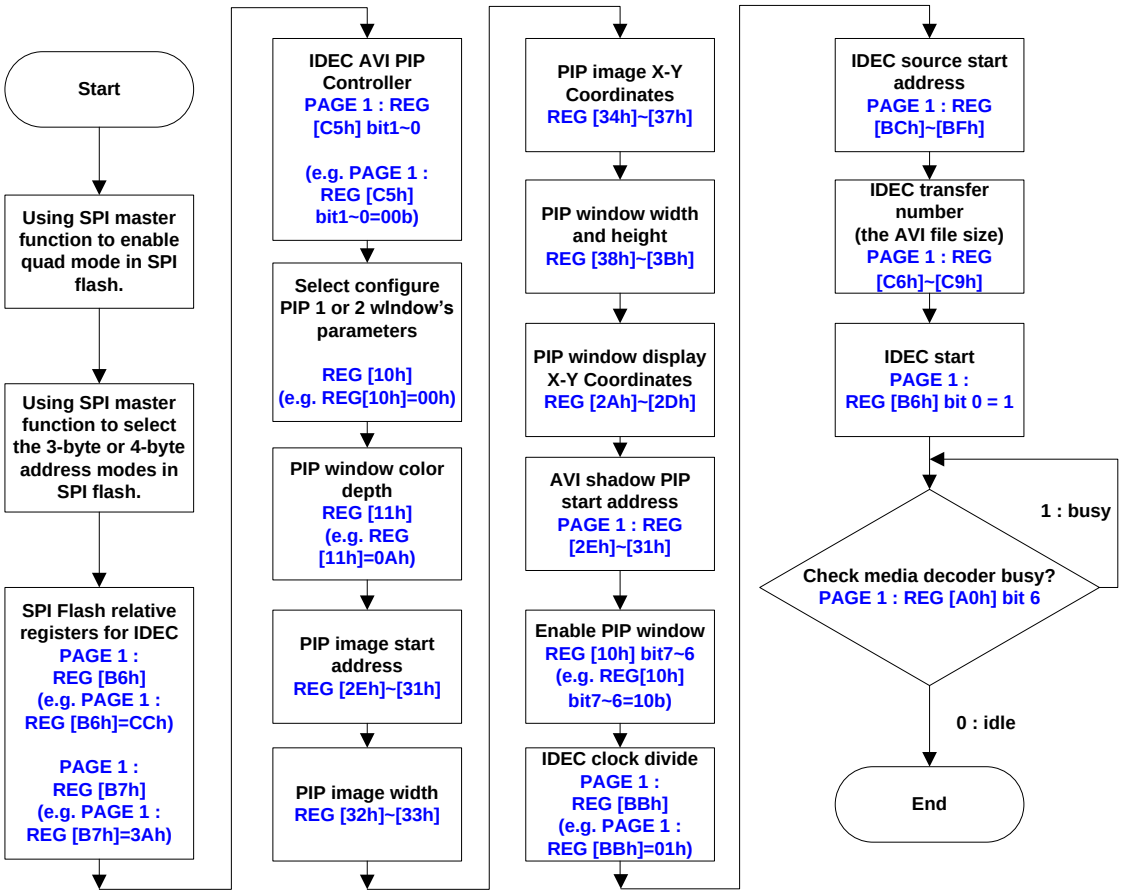


Figure 18-4

註: IDEC 為支援 Media Decoder Unit 的 serial flash 控制介面

19. 省電模式

RA8889 有兩種操作狀態，一個是一般狀態，另一個是省電狀態。因此操作模式總共有四種耗電模式，依照消耗電量容量由高至低為 Normal、Suspend、Standby、Sleep。在下面的黑體字表示使用者輸入的相關命令。

註: RA8889 進入省電模式時，RA8889 的 LCD 介面將不輸出訊號，因此進入省電模式前，需先在硬體系統上對 LCD 模組做 display off 或 power off 的動作，以避免 LCD 極化損壞。

19.1 一般狀態

19.1.1 標準模式

使用者必須針對 CPLL、MPLL、SPLL 設定合適的暫存器值。而使用者也必須等待 PLL 時脈穩定，這個以透過暫存器 01h bit[7]得知 PLL 時脈是否穩定。

19.2 省電狀態

19.2.1 睡眠模式

下面是睡眠模式，所有的時脈(系統時脈、記憶體時脈、掃描時脈) 最後都將會停止。

進入睡眠模式的步驟如下:

i. 設定省電模式為睡眠模式。
ii. 進入省電模式狀態(設定暫存器 DFh bit[7]為 1)。
iii. SDRAM 會自動進入 power down 模式或自我刷新模式，而這是根據**暫存器 E0h bit7 的設定** (若 E0h bit [7] 為 0，則在 RA8889 進入省電模式時，SDRAM 會 power down；若是設定 E0h bit 7 為 1，在 RA8889 進入省電模式時，SDRAM 會進入自我刷新模式。)
iv. 內部電路進入睡眠模式 (sleep state)。
v. 禁能記憶體時脈與掃描時脈。
vi. 切換系統時脈由 CPLL 時脈改為 OSC。
vii. 如果 MPU 介面是並列介面，那麼 RA8889 會停掉 OSC，如果 MPU I/F 是串列介面，那麼 RA8889 不會停止 OSC。
viii. 關閉所有 PLL 的電源(CPLL/SPLL/MPLL)。
ix. 使用者檢查狀態暫存器的 power saving 位元，並且等待位元變成 1，這樣可以確保 RA8889 已經進入了省電模式。

***註*：進入省電模式這些週期的過程中，任何喚醒功能都是不被接受的。**

回到標準模式的步驟如下:

i. 離開 power saving state (設定 DFh bit[7] as 0)。
ii. 如果在睡眠模式 OSC 被停止了，則必須要致能 OSC。
iii. 切換系統時脈為 OSC。
iv. 回復所有的 PLL (CPLL/SPLL/MPLL)。
v. 切換所有的時脈(系統時脈、記憶體時脈、掃描時脈) 為 PLL 時脈。
vi. 使用者檢查狀態暫存器的 power saving bit 並且等待 bit 變為 0。

19.2.2 休眠模式

在休眠模式 (suspend mode) 之下，系統時脈、記憶體時脈、掃描時脈將會停止，並且記憶體時脈將會被切換到 OSC 時脈。

進入休眠模式的步驟如下:

i. 根據 OSC 頻率設定合適的 SDRAM 刷新率。
ii. 設定省電模式為 suspend mode。
iii. 進入省電模式 (設定 DFh bit[7]為 1)。
iv. 內部電路進入休眠模式 (suspend state)。
v. 自動禁能掃描時脈。
vi. 自動切換系統時脈與記憶體時脈由 PLL 變為 OSC。
vii. 自動禁能系統時脈。
viii. 保持 OSC 執行。
ix. 關閉所有的 PLL 電源 (CPLL/SPLL/MPLL)。
x. 使用者檢查狀態暫存器的 power saving bit 並且等待變為 1，這個以確保 RA8889 已經進入省電模式。

***註*：進入省電模式這些週期的過程中，任何喚醒功能都是不被接受的。**

回到標準模式的步驟如下:

i. 離開 power saving state (設定 DFh bit[7] as 0)。
ii. 如果在休眠模式 OSC 被停止了，則必須要致能 OSC。
iii. 切換系統時脈為 OSC。
iv. 回復所有的 PLL (CPLL/SPLL/MPLL)。
v. 切換所有的時脈 (系統時脈、記憶體時脈、掃描時脈) 為 PLL 時脈。
vi. 使用者檢查狀態暫存器的 power saving bit 並且等待 bit 變為 0。

19.2.3 Standby 模式

進入 standby 模式後，系統時脈與掃描時脈將會被停止，記憶體時脈則會繼續由 MPLL clock 提供。

進入 standby 模式的步驟如下:

i. 設定省電模式為 standby 模式。
ii. 進入省電模式 (設定 DFh bit[7] as 1)。
iii. 內部電路進入 standby 模式。
iv. 禁能掃描時脈。
v. 切換系統時脈為 OSC，並且維持記憶體時脈是由 MPLL clock 提供。
vi. 保持 OSC 執行。
vii. 維持所有的 PLL 在動作狀態以便快速回復。
viii. 使用者檢查狀態暫存器的 power saving bit 並且等待變為 1，這個以確保 RA8889 已經進入省電模式。

***註*: 進入省電模式這些週期的過程中，任何喚醒功能都是不被接受的。**

回到標準模式的步驟如下:

i. 離開 power saving state (設定 DFh bit[7] as 0)。
ii. 切換系統時脈與掃描時脈為 PLL 時脈。
iii. 使用者檢查狀態暫存器的 power saving bit 並且等待 bit 變為 0。

19.3 電源模式比較表

Item	Normal State	Power Saving State					
	Normal mode	**Standby** mode		**Suspend** mode		**Sleep** mode	
	PLL enable	Parallel MPU	Serial MPU	Parallel MPU	Serial MPU	Parallel MPU	Serial MPU
MCLK	MPLL clock	**MPLL clock**	**MPLL clock**	**OSC**	**OSC**	stop	stop
CCLK	CPLL clock	**OSC**	**OSC**	stop	**OSC**	stop	**OSC**
PCLK	SPLL clock	stop	stop	stop	stop	stop	stop
CPLL	**On**	**On**	**On**	Off	Off	Off	Off
MPLL	**On**	**On**	**On**	Off	Off	Off	Off
SPLL	**On**	**On**	**On**	Off	Off	Off	Off

20. 暫存器說明

在 RA8889 的主控端介面提供 4 種形式的週期，而 RA8889 暫存器的讀寫就是透過這些週期組成的。RA8889 包含一個狀態暫存器與許多的指令暫存器。狀態暫存器可以透過狀態讀取週期來讀取資料，其本身是唯讀的。而指令暫存器可以透過 “Command Write” 週期與 “Data Write” 週期去控制絕大部分的功能。

“Command Write” 指定暫存器的位址 (register number)，接著 “Data Write” 週期就可以將資料寫入暫存器中。當要讀取指定的暫存器資料時，主控端須要先 送 “Command Write” 週期，然後再使用 “Data read” 週期來讀取資料。“Command Write” 是設定暫存器位址，“Data Read” 則是讀取暫存資料。

Table 20-1 : Host Cycle Type

Cycle Type	XnCS	XA0	MPU_8080		MPU_6800		Description
			XnRD_EN	XnWR_RnW	XnRD_EN	XnWR_RnW	
Command Write	0	0	1	0	1	0	Register number write cycle
Status Read	0	0	0	1	1	1	Status read cycle
Data Write	0	1	1	0	1	0	Corresponding Register data/Memory data write cycle following the Command Write cycle.
Data Read	0	1	0	1	1	1	Corresponding Register data/Memory data read cycle following the Command Write cycle.

下面列出的是暫存器功能描述，每個暫存器表格上方都是暫存器名稱與位址。每個暫存器最多皆為 8-bit，這部分在暫存器功能表格中會詳細的描述預設值與屬性。

(RO: Read only, WO: Write only, RW: Read-able and Write-able)

20.1 狀態暫存器

Status Register (STSR)

Bit	Description	Default	Access
7	**主控端記憶體 Write FIFO full** 0: 記憶體 Write FIFO 沒有 full。 1: 記憶體 Write FIFO 有 full。 只有在記憶體 Write FIFO 沒有 full 的情況下，MPU 才可以寫下一個像素。	0	RO
6	**主控端記憶體 Write FIFO empty** 0: 記憶體 Write FIFO 沒有 empty。 1: 記憶體 Write FIFO 有 empty。 當記憶體 Write FIFO 是 empty 時，MPU 可以寫入 8bpp 資料 64 個像素或 16bpp 資料 32 個像素或 24bpp 資料 16 個像素。	1	RO

Bit	Description	Default	Access
5	**主控端記憶體 Read FIFO full** 0: 記憶體 Read FIFO 沒有 full。 1: 記憶體 Read FIFO 有 full。 當記憶體 Read FIFO 是 full 時，MPU 可以讀取 8bpp 資料 64 個像素或 16bpp 資料 32 個像素或 24bpp 資料 16 個像素。	0	RO
4	**主控端記憶體 Read FIFO empty** 0: 記憶體 Read FIFO 沒有 empty。 1: 記憶體 Read FIFO 有 empty。	1	RO
3	**Core task is busy (fontwr_busy)** 此旗標為下面幾種核心忙碌旗標: BTE、幾何引擎、DMA、文字寫入或圖形寫入。 0: 任務完成或閒置。 1: 任務忙碌。 當使用者切換文字與圖形模式或是更改底圖相關設定時，必須要先確認 RA8889 是否閒置。 註 ： BTE、幾何引擎、DMA 也可以檢查其功能本身的起始位元。 而在文字模式下，如果使用者再更改文字旋轉、行間距、字元間隔、前景色、背景色與文字/圖形設定前，都必須確認 core_busy (fontwr_busy) 這個 bit 為 0。	0	RO
2	**SDRAM ready for access** 0: SDRAM 還沒準備好被存取。 1: SDRAM 已經可以被存取。 在使用者檢查這個位元的狀態之前，使用者必須先設定 “sdr_initdone” 位元為 1。	0	RO
1	**Operation mode status** 0: Normal 操作。 1: Inhibit 操作。 Inhibit 操作表示 RA8889 內部正在進行內部重置或是開機顯示或是進入了省電模式當中。 在省電模式，此位元會維持在 1 直到 PLL 時脈被停止。所以這個 bit 與 REG([DFh]bit[7]) 會有一點點的時間差。	0	RO
0	**Interrupt pin state** 0: 沒有中斷產生。 1: 有中斷產生。	0	RO

Note : “RO” means read only.

20.2 IC 組態暫存器

RA8889 有兩個 page-page0/page1 的暫存器，使用者可以在 page0/page1 的 REG[46h]bit0，切換 page1/page0。

PAGE0 REG[00h] Software Reset Register (SRR)

Bit	Description	Default	Access
7-5	NA	06h	RO
4-2	NA	05h	RO
1	NA	1	RO
0	**Software Reset** 0: Normal 操作。 1: Software Reset。 Software Reset 只會重置內部的狀態機，至於暫存器值是不會清除的。所以所有唯讀的暫存器可以回傳本身的初始值。使用者應該有適當的設定以確定旗標是期望的狀態。 *註*：這個 bit 在 reset 完成後會自動被清掉。	0	WO
0	**Warning condition flag** 0: 沒有警告產生。 1: 警告產生。 更詳細的資訊請檢查 REG[E4h] bit 3。	0	RO

PAGE0 REG[01h] Chip Configuration Register (CCR)

Bit	Description	Default	Access
7	**Reconfigure PLL frequency** 對這個 bit 寫“1”可以重新設定 PLL 頻率。 *註* a. 當使用者更改 PLL 相關參數，PLL 時脈不會馬上改變，使用者還必須再次將這個 bit 設定為 1，PLL 時脈才會改變。 b. 使用者可以讀取(檢查)這個 bit 以知道系統是否已經切換到 PLL 時脈，”1”表示 PLL 時脈已經就緒並且切換成功。	1	RW
6	**Mask XnWAIT on XnCS deassert** 0: No mask XnWAIT 不論在 XnCS assert /deassert 的情形下，只要內部是忙碌的 XnWAIT 會維持 assert，並且此時無法接受下一個讀寫週期。如果 MPU 本身的週期無法在 XnWAIT 為低電位時，去擴展週期以等待 RA8876 完成的話，那麼使用者應該輪詢 XnWAIT 的準位，並且在 XnWAIT 為高電位時才能進行下一次的存取。 1: Mask 當 XnCS 收掉時強制 XnWAIT 也會收掉，因此 MPU 使用上必須透過 XnWAIT 來自動的延長週期。	0	RW

Bit	Description	Default	Access
5	**Key-Scan Enable/Disable** 0: 禁能。 1: 致能。	0	RW
4-3	**For RA8889 TFT Panel I/F Output pin Setting** 00b: 24-bit TFT output。 01b: 18-bit TFT output。 10b: 16-bit TFT output。 11b: w/o TFT output。 其他未使用的 TFT 輸出腳位被設定為 GPIO 與按鍵功能 Key。	01b	RW
2	**IIC master Interface Enable/Disable** 0: 禁能 (GPIO function)。 1: 致能 (IIC master function)。 IIC master 與 XKIN[0] & XKOUT[0] 腳位共用。 這個 bit 較 Key-Scan 致能 bit 具有更高的優先權，換句話說如果 IIC master 與 Key-Scan 同時致能的話，那麼 XKIN[0] /XKOUT[0] 將會是 IIC 的功能，至於其他的 XKIN /XKOUT 腳位則會維持 Key-scan 功能。	0	RW
1	**Serial Flash or SPI Interface Enable/Disable** 0: 禁能 (GPIO function)。 1: 致能 (SPI master function)。	0	RW
0	**Host Data Bus Width Selection** 0: 8-bit 主控端資料匯流排。 1: 16-bit 主控端資料流排。 *** 如果 Serial host I/F 被選擇或是在開機顯示的操作週期，RA8889 將會將這個 bit 設為 0，並且只允許 8-bit 寬度的存取。	0	RW

PAGE0 REG[02h] Memory Access Control Register (MACR)

Bit	Description	Default	Access
7-6	**Host Read/Write image Data Format** MPU 針對記憶體的讀寫資料格式。 **0xb**:直接寫入，可以使用格式如下: 1. 8 bits MPU I/F 2. 16 bits MPU I/F with 8bpp data mode 1 & 2 3. 16 bits MPU I/F with 16/24-bpp data mode 1 4. serial host interface **10b**: 對每筆資料皆遮罩 high byte(如 16 bit MPU I/F 使用的是 8-bpp data mode 1 資料格式)。 **11b**: 對偶數資料遮罩 high byte(如 16 bit MPU I/F 使用 24-bpp data mode 2)。	0	RW

Bit	Description	Default	Access
5-4	**Host Read Memory Direction (Only for Graphic Mode)** 00b: 左→右 然後 上→下。 01b: 右→左 然後 上→下。 10b: 上→下 然後 左→右。 11b: 下→上 然後 左→右。 如果底圖設定是 linear 定址模式則此兩 bit 可忽略。	0	RW
3	**NA**	0	RO
2-1	**Host Write Memory Direction (Only for Graphic Mode)** 00b: 左→右 然後 上→下. (Original)。 01b: 右→左 然後 上→下. (Horizontal flip)。 10b: 上→下 然後 左→右. (Rotate right 90°& Horizontal flip)。 11b: 下→上 然後 左→右. (Rotate left 90°)。 如果底圖設定是線性定址模式，則此兩 bit 可忽略。	0	RW
0	**NA (must keep it as 0)**	0	RO

PAGE0 REG[03h] Input Control Register (ICR)

Bit	Description	Default	Access
7	**Output to MPU Interrupt pin's active level** 0 : active low. 1 : active high.	0	RW
6	**External interrupt input (XPS[0] pin) de-bounce** 0 : 不需要 de-bounce。 1 : 致能 de-bounce (1024 OSC clock)。	0	RW
5-4	**External interrupt input (XPS[0] pin) trigger type** 00 :低準位觸發。 01 :下降邊緣觸發。 10 :高準位觸發。 11 :上升邊緣觸發。	00b	RW
3	NA	0	RW
2	**Text Mode Enable** 0 : 圖形模式。 1 : 文字模式。 在設定這個 bit 之前，必須先確定 core task busy 是否正在忙碌或閒置中，而 core task busy 是狀態暫存器。 如果在 linear 定址模式中，這個 bit 始終為 0。	0	RW
1-0	**Memory port Read/Write Destination Selection** **00b**: 選擇 SDRAM 為 image/pattern/使用者自訂字型的資料寫入目的，支援 Read-modify-Write。	0	RW

Bit	Description	Default	Access
	01b: 選擇RGB色的Gamma table為寫入目的。每個顏色的table都是256 bytes。使用者需要指定需要寫入的gamma table然後再連續寫入256 bytes。 **10b**：圖形游標的記憶體 (只能接受low 8-bits MPU資料，類似一般暫存器的資料讀寫)，不支援Graphic Cursor 記憶體讀取功能。圖形游標記憶體包含 4 種圖形游標的顏色設定。每一個設定都具有 128x16 bits。使用者使用的時候需要指定寫入目標為 graphic cursor ，然後再連續寫256 bytes。 **11b**：調色盤記憶體，這是64x12 bits的SRAM。因為MPU每次寫入8bit，因此在偶數次數寫入時，只有low 4 bit被當顏色寫入RAM。不支援調色盤記憶體被讀取。使用者需要連續寫128 bytes。		

PAGE0 REG[04h] Memory Data Read/Write Port (MRWDP)

Bit	Description	Default	Access
7-0	**Write Function : Memory Write Data** Data to write in memory corresponding to the setting of REG[03h][1:0]. 在大量資料的條件下，可以使用連續資料寫入。 *註*: **a. Image data in SDRAM**: 參考MPU I/F寬度設定為8/16-bits，可以設定主控端R/W image 資料格式。並設定區塊模式的底圖色深與底圖相關設定。 **b. Pattern data for BTE operation in SDRAM**: 參考MPU I/F 寬度設定為8/16-bits，可以設定主控端R/W image 資料格式。並設定區塊模式的底圖色深與底圖相關設定。工作視窗依照使用者需求應該是被設定為8x8 或 16x16像素。 **c. User-characters in SDRAM**: 參考 MPU I/F 寬度設定為8/16-bits，可以設定主控端R/W image 資料格式。並且設定底圖為linear 模式。 **d. Character code**: 只能接受MPU 資料的low 8-bits，使用上類似於暫存器讀寫方式。若是字元碼為2byes ，則先輸入high bytes。若是以自建字型，字碼<8000h 為半形字；字碼>=8000h 為全形字。 **e. Gamma table data: 只能接受MPU資料的** low 8-bits。使用者另須設定“Select Gamma table ([3Ch] Bit6-5)” 來清除內部的Gamma table’s 位址計數器，然後才能開始進行寫入的動作。使用者應該寫入256 bytes資料到記憶體中。 **f. Graphic Cursor RAM data**: 只能接受MPU的low 8-bits 資料。還必須設定 “Select Graphic Cursor sets” 暫存器以清除Graphic Cursor RAM 位址計數器，然後再進行寫入的動作。	--	RW

Bit	Description	Default	Access
	g. Color palette RAM data: 只能接受 MPU 寫入的 low 8-bits 資料。使用者還必須針對 Color palette RAM (64x12) 連續寫下 128 byte 的資料，並且在寫入過程中不能改暫存器位址。 **Read Function : Memory Read Data** 使用讀取記憶體資料功能，必須設定 REG[03h][1:0]，若要使用連續資料讀取功能，則必須在大量資料讀取的設定條件下。 ***註 1*** **：** 如果在 read 要讀取不同的位址資料，那必須要發出空週期，因為空週期是第一個讀取資料的週期，而讀到的資料應該是要被捨棄的。圖形游標記憶體與調色盤記憶體並不支援讀取功能。 ***註 2*** **：** 不論色深的設定，讀取資料是以 4 bytes 做為基準的。 ***註 3*** **：** 如果使用者要更改寫入暫存器的位址，但若之前已經先寫資料到 SDRAM 中，那麼使用者應該先確認 RA8889 的 core task busy 旗標是否顯示為閒置狀態，若為閒置才可更改暫存器位址。		

20.3 PLL 組態暫存器

PAGE0 REG[05h] SCLK PLL Control Register 1 (PPLLC1)

Bit	Description	Default	Access
7	保留	0	RW
6	NA	0	RO
5-3	**SCLK extra divider** xx1b: 除 16。 000b: 除 1。 010b: 除 2。 100b: 除 4。 110b: 除 8。	0	RW
2-1	**SCLK PLLDIVK[1:0]** SCLK PLL 輸出除頻 00b: 除 1。 01b: 除 2。 10b: 除 4。 11b: 除 8。	2	RW
0	**SCLK PLLDIVM** PCLK PLL Pre-driver parameter. 0b: 除 1。 1b: 除 2。	0	RW

PAGE0 REG[06h] SCLK PLL Control Register 2 (PPLLC2)

Bit	Description	Default	Access
7-6	**NA**	0	RO
5-0	**SCLK PLLDIVN[5:0]** SCLK PLL 輸入參數，數值應該在 1~63。 (數值 0 是禁止的)。	17h	RW

*PCLK is used by panel's scan clock and derived from SCLK.

PAGE0 REG[07h] MCLK PLL Control Register 1 (MPLLC1)

Bit	Description	Default	Access
7-3	**NA**	0	RO
2-1	MCLK PLLDIVK[1:0] PCLK PLL Output divider 00b: 除 1。 01b: 除 2。 10b: 除 4。 11b: 除 8。	1	RW
0	**MCLK PLLDIVM** MCLK PLL Pre-driver parameter. 0b: 除 1。 1b: 除 2。	0	RW

PAGE0 REG[08h] MCLK PLL Control Register 2 (MPLLC2)

Bit	Description	Default	Access
7-6	**NA**	0	RO
5-0	**MCLK PLLDIVN[5:0]** MCLK PLL輸入參數，數值應該在 1~63。 (數值 0 是禁止的)。	1Dh	RW

*MCLK is used by SDRAM's clock

PAGE0 REG[09h] CCLK PLL Control Register 1 (SPLLC1)

Bit	Description	Default	Access
7-3	**NA**	0	RO
2-1	**CCLK PLLDIVK[1:0]** CCLK PLL 輸出除頻 00b: 除 1。 01b: 除 2。 10b: 除 4。 11b: 除 8。	2	RW
0	**CCLK PLLDIVM** CCLK PLL Pre-driver parameter. 0b: 除 1。 1b: 除 2。	0	RW

PAGE0 REG[0Ah] CCLK PLL Control Register 2 (SPLLC2)

Bit	Description	Default	Access
7-6	**NA**	0	RO
5-0	**CCLK PLLDIVN[5:0]** CCLK PLL輸入參數，數值應該在 1~63。 (數值 0 是禁止的)。	2Ah	RW

*CCLK is used by core's clock

RA8889 的時脈是由 OSC 及內部的 xCLK PLL 電路產生的。以下公式用於時脈計算。

$$xCLK = \frac{\left(Fin / 2^{(xPLLDIVM)}\right) \times (xPLLDIVN + 1)}{2^{xPLLDIVK}}$$

註：

1. 只有當 PLL 關閉時，PLL 的參數才能改變。
2. 如果 REG[05h]~REG[0Ah]被設定，那麼使用者應該要等待 PLL 輸出穩定，這個時間 lock time (< 30us)。
3. 輸入的 OSC 頻率(F_{IN})必須符合下面描述的範圍，並且 PLLDIVM 與 F_{IN} 的計算如下:

$$10MHz \le Fin \le 15MHz$$
$$\&$$
$$10MHz \le \frac{Fin}{2^{PLLDIVM}} \le 40MHz$$

4. 內部倍頻的頻率 $Fvco = \frac{Fin}{2^{PLLDIVM}} \times (PLLDIVN + 1)$ 必須要等於或大於 250 MHz，但是必須小於 500MHz。換句話說: $250MHz \le Fvco \le 500MHz$

20.4 中斷控制暫存器

中斷相關的暫存器為“Interrupt Enable”、”Interrupt Event Flag”與 ”Mask Interrupt Flag”暫存器。

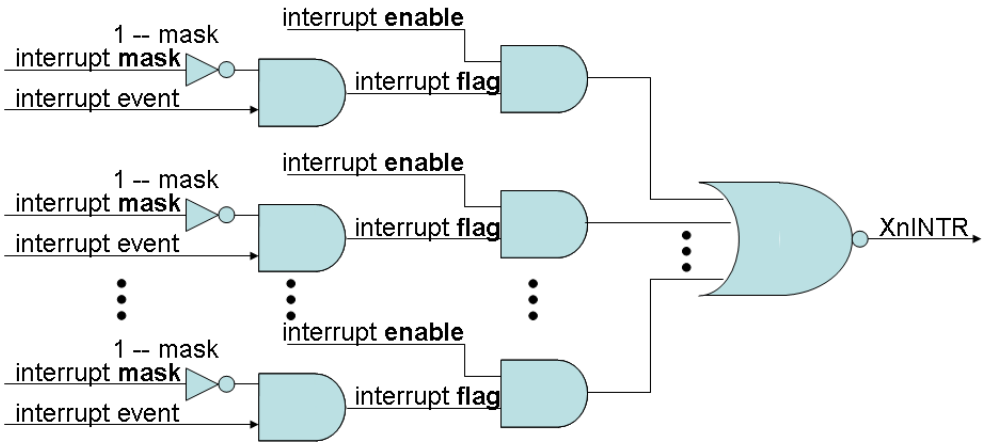


Figure 20-1

PAGE0 REG[0Bh] Interrupt Enable Register (INTEN)

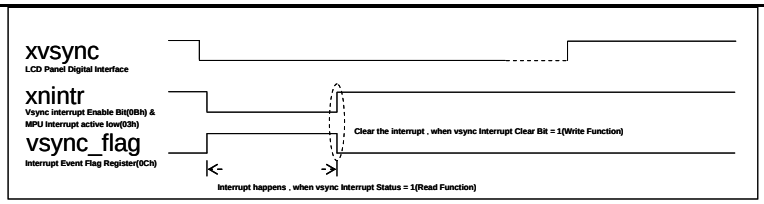
Bit	Description	Default	Access
7	**Wakeup/resume Interrupt Enable** 0: 禁能。 1: 致能。	0	RW
6	**External Interrupt input (XPS[0] pin) Enable** 0: 禁能。 1: 致能。	0	RW
5	**IIC Master Interrupt Enable** 0: 禁能。 1: 致能。	0	RW
4	**Vsync time base interrupt Enable Bit** 0: 禁能中斷。 1: 致能中斷。 This interrupt event may provide the host processor with Vsync signal information for tearing effect.	0	RW
3	**Key Scan Interrupt Enable Bit** 0: 禁能中斷。 1: 致能中斷。	0	RW
2	**Serial flash DMA Complete | Draw task finished | BTE Process Complete etc. Interrupt Enable** 0: 禁能中斷。 1: 致能中斷。	0	RW

Bit	Description	Default	Access
1	**PWM timer 1 Interrupt Enable Bit** 0: 禁能中斷。 1: 致能中斷。	0	RW
0	**PWM timer 0 Interrupt Enable Bit** 0: 禁能中斷。 1: 致能中斷。	0	RW

PAGE0 REG[0Ch] Interrupt Event Flag Register (INTF)

* 如果使用者收到中斷，但是透過這個暫存器卻沒有中斷，那麼使用者應該要去確認 SPI master 狀態暫存器的中斷旗標 REG[BAh]。

Bit	Description	Default	Access
7	**Wakeup/resume Interrupt flag** **Write Function ➔ Wakeup/resume Interrupt Clear Bit** 0: 無動作。 1: 清除 Wakeup/resume 中斷旗標。 **Read Function ➔Wakeup/resume Interrupt Status** 0: 沒有 Wakeup/resume 中斷產生。 1: Wakeup/resume 中斷產生。	0	RW
6	**External Interrupt input (XPS[0] pin) flag** **Write Function ➔ XPS[0] pin edge Interrupt Clear Bit** 0: 無動作。 1: 清除 XPS[0] 中斷旗標。 **Read Function ➔ XPS[0] pin Interrupt Status** 0: 沒有 XPS[0] 中斷產生。 1: XPS[0] 中斷產生。	0	RW
5	**IIC master Interrupt flag** **Write Function➔ IIC master Interrupt Clear Bit** 0: 無動作。 1: 清除 IIC master 中斷旗標。 **Read Function ➔ IIC master Interrupt Status** 0: 沒有 IIC master 中斷產生。 1: IIC master 中斷產生。	0	RW
4	**Vsync Time base interrupt flag** **Write Function ➔Vsync Interrupt Clear Bit** 0: 無動作。 1: 清除 vsync 中斷旗標。 **Read Function ➔ Vsync Interrupt Status** 0: 沒有 vsync 中斷產生。 1: 有 vsync 中斷產生。	0	RW

Bit	Description	Default	Access
	 *if xvsync is low active.		
3	**Key Scan Interrupt flag** **Write Function➔Key Scan Interrupt Clear Bit** 0: 無動作。 1: 清除 Key Scan 中斷。 **Read Function ➔Key Scan Interrupt Status** 0: 沒有 Key Scan 中斷產生。 1: 有 Key Scan 中斷產生。	0	RW
2	**Serial flash DMA Complete | Draw task finished | BTE Process Complete etc. Interrupt flag** **Write Function➔ Interrupt Clear Bit** 0: 無動作。 1: 清除中斷旗標。 **Read Function➔Interrupt Status** 0: 沒有中斷產生。 1: 有中斷產生。	0	RW
1	**PWM 1 timer Interrupt flag** **Write Function➔Interrupt Clear Bit** 0: 無動作。 1: 清除 PWM1 中斷旗標。 **Read Function➔Interrupt Status** 0: 沒有 PWM1 中斷產生。 1: 有 PWM1 中斷產生。	0	RW
0	**PWM 0 timer Interrupt flag** **Write Function➔Interrupt Clear Bit** 0: 無動作。 1: 清除 PWM0 中斷旗標。 **Read Function➔Interrupt Status** 0: 沒有 PWM0 中斷產生。 1: 有 PWM0 中斷產生。	0	RW

PAGE0 REG[0Dh] Mask Interrupt Flag Register (MINTFR)

*** 如果使用者遮罩中斷旗標，那麼 RA8889 不會發出中斷給 MPU，而 MPU 也不需要去檢查中斷旗標(Interrupt Flag)。但是如果使用只有某些中斷旗標沒有被遮罩掉，那麼 MPU 不會收到中斷，但是 MPU 可以透過檢查中斷旗標以得知中斷產生。

Bit	Description	Default	Access
7	**Mask Wakeup/resume Interrupt Flag** 0: 不遮罩。 1: 遮罩。	0	RW
6	**Mask External Interrupt (XPS[0] pin) Flag** 0: 不遮罩。 1: 遮罩。	0	RW
5	**Mask IIC Master Interrupt Flag** 0: 不遮罩。 1: 遮罩。	0	RW
4	**Mask Vsync time base interrupt Flag** 0: 不遮罩。 1: 遮罩。	0	RW
3	**Mask Key Scan Interrupt Flag** 0: 不遮罩。 1: 遮罩。	0	RW
2	**Mask Serial flash DMA Complete | Draw task finished | BTE Process Complete etc. Interrupt Flag** 0: 不遮罩。 1: 遮罩。	0	RW
1	**Mask PWM timer 1 Interrupt Flag** 0: 不遮罩。 1: 遮罩。	0	RW
0	**Mask PWM timer 0 Interrupt Flag** 0: 不遮罩。 1: 遮罩。	0	RW

PAGE0 REG[0Eh] Pull- high control Register (PUENR)

Bit	Description	Default	Access
7:6	**NA**	0	RO
5	**GPIO-F[7:0] Pull-high Enable (XPDAT[23:19, 15:13])** 0: 提升電阻禁能。 1: 提升電阻致能。 *只有在 XPDAT 被設成 GPIO 功能，此位元才有意義。	0	RW

Bit	Description	Default	Access
4	**GPIO-E[7:0] Pull- high Enable (XPDAT[12:10, 7:3])** 0: 提升電阻禁能。 1: 提升電阻致能。 * 只有在 XPDAT 被設成 GPIO 功能，此位元才有意義。	0	RW
3	**GPIO-D[7:0] Pull- high Enable (XPDAT[18, 2, 17, 16, 9, 8, 1,0])** 0: 提升電阻禁能。 1: 提升電阻致能。 *只有在 XPDAT 被設成 GPIO 功能，此位元才有意義。	0	RW
2	**GPIO-C[4:0] Pull- high Enable (XnSFCS1, XnSFCS0, XMISO, XMOSI , XSCK)** 0: 提升電阻禁能。 1: 提升電阻致能。	0	RW
1	**XDB[15:8] Pull- high Enable** 0: 提升電阻禁能。 1: 提升電阻致能。	0	RW
0	**XDB[7:0] Pull- high Enable** 0: 提升電阻禁能。 1: 提升電阻致能。	0	RW

PAGE0 REG[0Fh] PDAT for PIO/Key Function Select Register (PSFSR)

Bit	Description	Default	Access
7	**XPDAT[18] – GPIO or Key function select** 0: GPIO-D7 1: XKOUT[4]	0	RW
6	**XPDAT[17] –GPIO or Key function select** 0: GPIO-D5 1: XKOUT[2]	0	RW
5	**XPDAT[16] –GPIO or Key function select** 0: GPIO-D4 1: XKOUT[1]	0	RW
4	**XPDAT[9] –GPIO or Key function select** 0: GPIO-D3 1: XKOUT[3]	0	RW
3	**XPDAT[8] –GPIO or Key function select** 0: GPIO-D2 1: XKIN[3]	0	RW
2	**XPDAT[2] –GPIO or Key function select** 0: GPIO-D6 1: XKIN[4]	0	RW
1	**XPDAT[1] –GPIO or Key function select** 0: GPIO-D1 1: XKIN[2]	0	RW
0	**XPDAT[0] –GPIO or Key function select** 0: GPIO-D0 1: XKIN[1]	0	RW

20.5 LCD 顯示控制暫存器

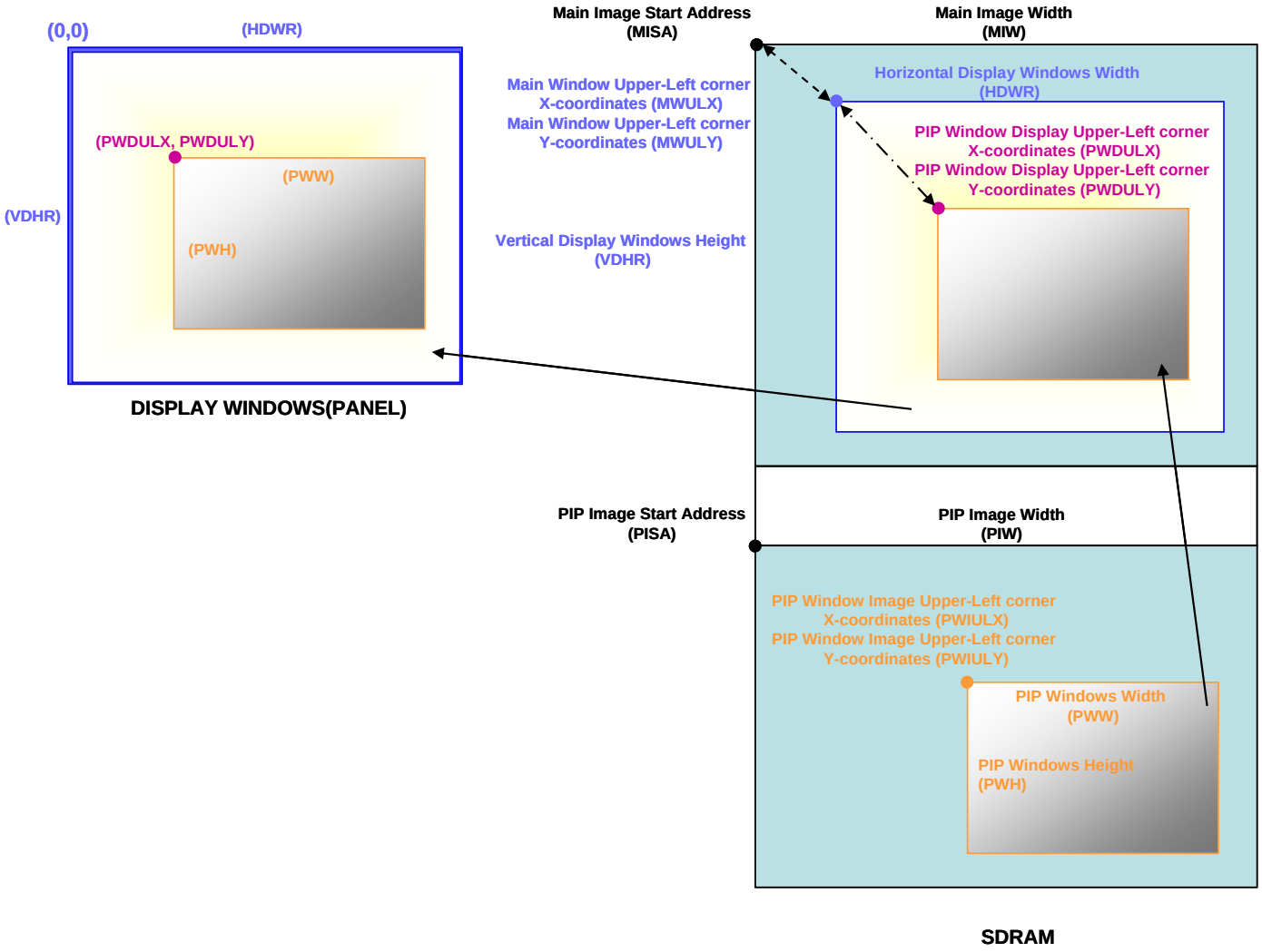


Figure 20-2 : LCD Display

PAGE0 REG[10h] Main/PIP Window Control Register (MPWCTR)

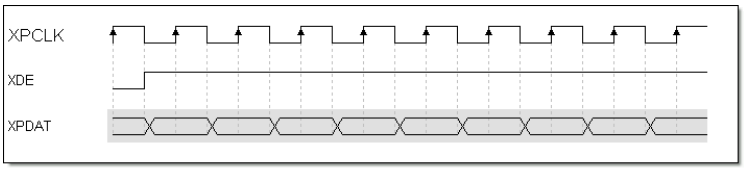
Bit	Description	Default	Access
7	**PIP 1 window Enable/Disable** 0 : PIP 1 禁能。 1 : PIP 1 致能。 PIP 1 視窗永遠在 PIP 2 視窗之上。	0	RW
6	**PIP 2 window Enable/Disable** 0 : PIP 2 禁能。 1 : PIP 2 致能。 PIP 1 視窗永遠在 PIP 2 視窗之上。	0	RW
5	NA	0	RO

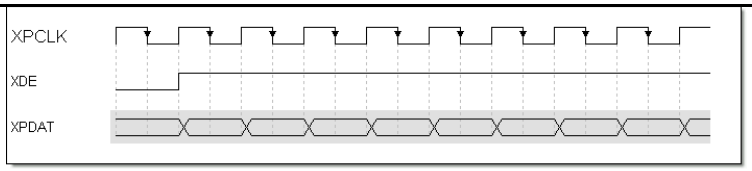
Bit	Description	Default	Access
4	**Select Configure PIP 1 or 2 Window's parameters** PIP 視窗的參數包含 ： 色深、起始位址、圖像寬度、顯示座標、視窗座標、視窗寬度、視窗高度。 0: 可以設定 PIP 1 的參數。 1: 可以設定 PIP 2 的參數。	0	RW
3-2	**Main Image Color Depth Setting** 00b: 8-bpp generic TFT, i.e. 256 色。 01b: 16-bpp generic TFT, i.e. 65K 色。 1xb: 18-bpp generic TFT, i.e. 16.7M 色。	1	RW
1	**NA**	0	RW
0	**To Control panel's synchronous signals** 0: Sync Mode: 致能 XVSYNC, XHSYNC, XDE。 1: DE Mode: 只有 XDE 致能，XVSYNC、XHSYNC 為閒置狀態。	0	RW

PAGE0 REG[11h] PIP Window Color Depth Setting (PIPCDEP)

Bit	Description	Default	Access
7-4	**NA**	0	RO
3-2	**PIP 1 Window Color Depth Setting** 00b: 8-bpp generic TFT, i.e. 256 色。 01b: 16-bpp generic TFT, i.e. 65K 色。 1xb: 24-bpp generic TFT, i.e. 1.67M 色。	1	RW
1-0	**PIP 2 Window Color Depth Setting** 00b: 8-bpp generic TFT, i.e. 256 色。 01b: 16-bpp generic TFT, i.e. 65K 色。 1xb: 24-bpp generic TFT, i.e. 1.67M 色。	1	RW

PAGE0 REG[12h] Display Configuration Register (DPCR)

Bit	Description	Default	Access
7	**PCLK Inversion** 0: XPDAT, XDE, XHSYNC ，Panel 抓取 XPDAT 是在 XPCLK 上升緣。  1: XPDAT, XDE, XHSYNC，Panel 抓取 XPDAT 在 PCLK 下降緣。	0	RW

Bit	Description	Default	Access
			
6	**Display ON/OFF** 0b: 顯示關閉。 1b: 顯示開啟。	0	RW
5	**Display Test Color Bar** 0b: 禁能。 1b: 致能。	0	RW
4	**HDIR** 水平掃描方向 0 : 從左到右 1 : 從右到左	0	RW
3	**VDIR** 垂直掃描方向 0 : 從上到下 1 : 從下到上	0	RW
2-0	**Parallel XPDAT[23:0] Output Sequence** 000b : RGB。 001b : RBG。 010b : GRB。 011b : GBR。 100b : BRG。 101b : BGR。 110b : 灰階。 111b : 送出閒置狀態 (全螢幕資料皆為 0 (黑色) 或 1 (白色)，另須設 REG[13h] bit2 XPDAT IDLE STATE	0	RW

PAGE0 REG[13h] Panel Scan Clock and Data Setting Register (PCSR)

Bit	Description	Default	Access
7	**XHSYNC Polarity** 0 : Low 動作。 1 : High 動作。	0	RW
6	**XVSYNC Polarity** 0 : Low 動作。 1 : High 動作。	0	RW
5	**XDE Polarity** 0 : High 動作。 1 : Low 動作。	0	RW

Bit	Description	Default	Access
4	**XDE IDLE STATE** **(in power saving mode or DISPLAY OFF)** 0 : Pin “DE” 輸出為 low。 1 : Pin “DE” 輸出為 high。	0	RW
3	**XPCLK IDLE STATE** **(in power saving mode or DISPLAY OFF)** 0 : Pin “PCLK” 輸出為 low。 1 : Pin “PCLK” 輸出為 high。	0	RW
2	**XPDAT IDLE STATE** **(Must use with reg[12h] bit2-0 Parallel XPDAT[23:0] Output Sequence to send out idle state)** 0 : Pins “PDAT[23:0]” 輸出為 low。 1 : Pins “PDAT[23:0]” 輸出為 high。	0	RW
1	**XHSYNC IDLE STATE** **(in power saving mode or DISPLAY OFF)** 0 : Pin “HSYNC” 輸出為 low。 1 : Pin “HSYNC” 輸出為 high。	1	RW
0	**XVSYNC IDLE STATE** **(in power saving mode or DISPLAY OFF)** 0 : Pin “VSYNC”輸出為 low。 1 : Pin “VSYNC”輸出為 high。	1	RW

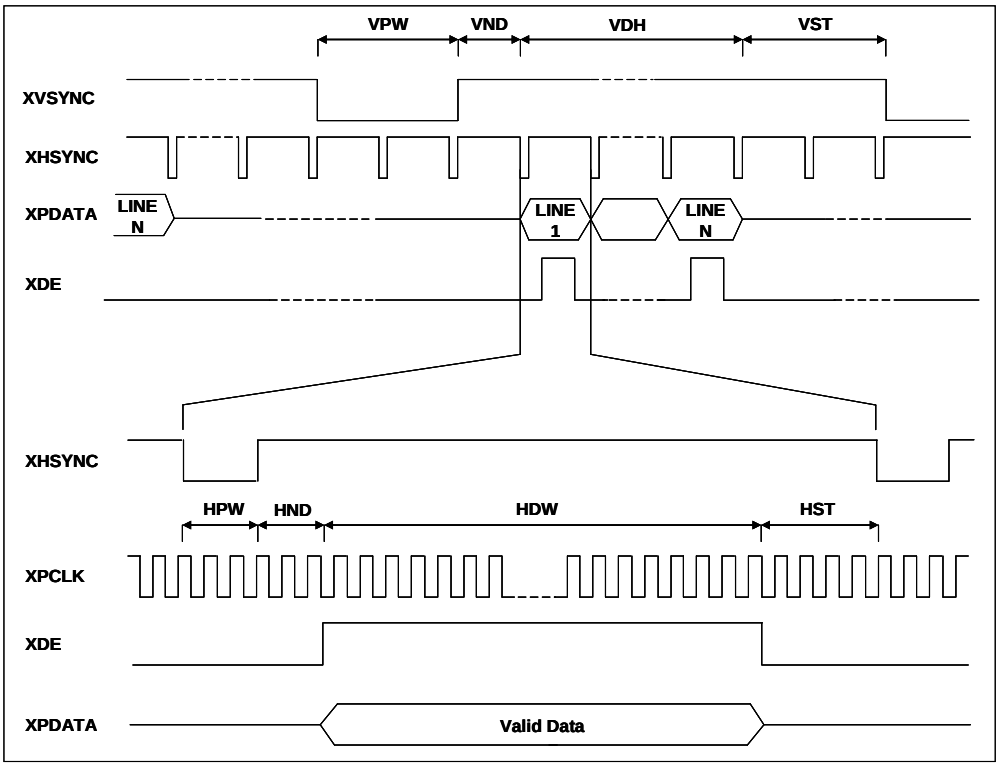


Figure 20-3 : Digital TFT Panel Timing

PAGE0 REG[14h] Horizontal Display Width Register (HDWR)

Bit	Description	Default	Access
7-0	**Horizontal Display Width Setting Bit[7:0]** 此暫存器為水平顯示寬度設定，其指定的的 LCD 螢幕解析度為 8 像素為一單元解析度。 Horizontal display width(pixels) = (HDWR + 1) * 8 + HDWFTR 水平寬度最大不可以超過 2048 像素。	4Fh	RW

PAGE0 REG[15h] Horizontal Display Width Fine Tune Register (HDWFTR)

Bit	Description	Default	Access
7-4	**NA**	0	RO
3-0	**Horizontal Display Width Fine Tuning (HDWFT) [3:0]** 此暫存器為水平顯示寬度的微調項，使用在螢幕的水平寬度並非為 8 的倍數上，每個細調的解析度為 1 個像素。 Horizontal display width(pixels) = (HDWR + 1) * 8 + HDWFTR 水平寬度最大不可以超過 2048 像素。	0	RW

PAGE0 REG[16h] Horizontal Non-Display Period Register (HNDR)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Horizontal Non-Display Period(HNDR) Bit[4:0]** 此暫存器為水平非顯示區域週期，這個暫存器指定了 horizontal non-display 的週期。因此又被稱為 **back porch**。 Horizontal non-display period or Back porch (pixels) = (HNDR + 1) * 8 + HNDFTR	03h	RW

PAGE0 REG[17h] Horizontal Non-Display Period Fine Tune Register (HNDFTR)

Bit	Description	Default	Access
7-4	**NA**	0	RO
3-0	**Horizontal Non-Display Period Fine Tuning(HNDFT) [3:0]** 此暫存器為水平非顯示區域週期(back porch)的微調項。通常被使用在具有 SYNC 模式的螢幕上，每個設定的基本單位是以 1-pixel 為單位。 Horizontal non-display period or Back porch (pixels) = (HNDR + 1) * 8 + HNDFTR	06h	RW

PAGE0 REG[18h] HSYNC Start Position Register (HSTR)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**HSYNC Start Position[4:0]** 此暫存器指定 HSYNC 的起始位址，其計算的起始點是顯示區域的結束時間點到開始產生 HSYNC 的時間點。每個調整的基本單位為 8-pixel，又被稱為 **front porch**。 HSYNC Start Position or Front porch (pixels) = (HSTR + 1)x8	1Fh	RW

PAGE0 REG[19h] HSYNC Pulse Width Register (HPWR)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**HSYNC Pulse Width(HPW) [4:0]** HSYNC 的週期寬度。 HSYNC Pulse Width (pixels) = (HPW + 1)x8	0	RW

PAGE0 REG[1Ah] Vertical Display Height Register 0(VDHR0)

Bit	Description	Default	Access
7-0	**Vertical Display Height Bit[7:0]** 此暫存器為垂直顯示高度(以 Line 為高度)，其計算式如下: Vertical Display Height (Line) = VDHR + 1	DFh	RW

PAGE0 REG[1Bh] Vertical Display Height Register 1 (VDHR1)

Bit	Description	Default	Access
7-3	**NA**	0	RO
2-0	**Vertical Display Height Bit[10:8]** 此暫存器為垂直顯示高度(以 Line 為高度)，其計算式如下: Vertical Display Height (Line) = VDHR + 1	01h	RW

PAGE0 REG[1Ch] Vertical Non-Display Period Register 0(VNDR0)

Bit	Description	Default	Access
7-0	**Vertical Non-Display Period Bit[7:0]** 此暫存器為垂直非顯示週期，其計算式如下: Vertical Non-Display Period (Line) = (VNDR + 1)	15h	RW

PAGE0 REG[1Dh] Vertical Non-Display Period Register 1(VNDR1)

Bit	Description	Default	Access
7-2	**NA**	0	RO
1-0	**Vertical Non-Display Period Bit[9:8]** 此暫存器為垂直非顯示週期，其計算式如下: Vertical Non-Display Period (Line) = (VNDR + 1)	0	RW

PAGE0 REG[1Eh] VSYNC Start Position Register (VSTR)

Bit	Description	Default	Access
7-0	**VSYNC Start Position[7:0]** VSYNC 的起始位址是由顯示區域結束時間點到開始有 VSYNC 的時間點。 VSYNC Start Position(Line) = (VSTR + 1)	0Bh	RW

PAGE0 REG[1Fh] VSYNC Pulse Width Register (VPWR)

Bit	Description	Default	Access
7-6	**NA**	0	RO
5-0	**VSYNC Pulse Width[5:0]** VSYNC 脈衝的寬度: VSYNC Pulse Width(Line) = (VPWR + 1)	0	RW

註：下面的暫存器 20h~3Bh 需要依次由 LSB 寫到 MSB。假設我們需要設定 Main Image Start Address，此暫存器為地址 20h 到 23h，必須依次由 LSB[20h] 寫到 MSB[23h]，當 REG[23h] 被寫入時，RA8889 才將會 REG[20h]~REG[23h] 的值寫到內部暫存器中。

PAGE0 REG[20h] Main Image Start Address 0 (MISA0)

Bit	Description	Default	Access
7-2	**Main Image Start Address[7:2]** 必須能被 4 整除，其中 Bit[1:0] 在 RA8889 中固定為 0。	0	RW
1-0	**NA**	0	RO

PAGE0 REG[21h] Main Image Start Address 1 (MISA1)

Bit	Description	Default	Access
7-0	**Main Image Start Address[15:8]**	0	RW

PAGE0 REG[22h] Main Image Start Address 2 (MISA2)

Bit	Description	Default	Access
7-0	**Main Image Start Address [23:16]**	0	RW

PAGE0 REG[23h] Main Image Start Address 3 (MISA3)

Bit	Description	Default	Access
7-0	**Main Image Start Address [31:24]**	0	RW

PAGE0 REG[24h] Main Image Width 0 (MIW0)

Bit	Description	Default	Access
7-0	**Main Image Width [7:0]** 單位: 像素。 必須能被 4 整除，MIW Bit [1:0] 在內部固定為 0。 這個值是真實的像素值。設定最大值為 8188 像素。	0	RW

PAGE0 REG[25h] Main Image Width 1 (MIW1)

Bit	Description	Default	Access
7-5	**NA**	NA	NA
4-0	**Main Image Width [12:8]** 單位: 像素。 必須能被 4 整除。 這個值是真實的像素值。設定最大值為 8188 像素。	0	RW

PAGE0 REG[26h] Main Window Upper-Left corner X-coordinates 0 (MWULX0)

Bit	Description	Default	Access
7-2	**Main Window Upper-Left corner X-coordinates [7:2]** 請參考 Main Image 座標 單位: 像素。 必須要能被 4 整除，MWULX Bit [1:0]在內部固定為“0”。 X-軸座標+Horizontal display width 不能大於 8188。	0	RW
1-0	**NA**	0	RO

PAGE0 REG[27h] Main Window Upper-Left corner X-coordinates 1 (MWULX1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Main Window Upper-Left corner X-coordinates [12:8]** 請參考 Main Image 座標。 單位: 像素。 必須要能被 4 整除。 X-軸座標+Horizontal display width 不能大於 8188。	0	RW

PAGE0 REG[28h] Main Window Upper-Left corner Y-coordinates 0 (MWULY0)

Bit	Description	Default	Access
7-0	**Main Window Upper-Left corner Y-coordinates [7:0]** 請參考 Main Image 座標。 單位: 像素。 此數值範圍為 0 到 8191。	0	RW

PAGE0 REG[29h] Main Window Upper-Left corner Y-coordinates 1 (MWULY1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Main Window Upper-Left corner Y-coordinates [12:8]** 請參考 Main Image 座標。 單位: 像素。 設定值範圍在 0 到 8191。	0	RW

PAGE0 REG[2Ah] PIP 1 or 2 Window Display Upper-Left corner X-coordinates 0 (PWDULX0)

Bit	Description	Default	Access
7-2	**PIP Window Display Upper-Left corner X-coordinates [7:2]** 請參考 Main Window 座標。 單位: 像素。 必須要能被 4 整除。 PWDULX Bit [1:0] 內部固定為 0。 X-軸座標應該要小於 horizontal display width。 根據 REG[10h] (Select Configure PIP 1 or 2 Window) 參數，這個設定值將為相關 PIP 的參數值。	0	RW
1-0	**NA**	0	RO

PAGE0 REG[2Bh] PIP 1 or 2 Window Display Upper-Left corner X-coordinates 1 (PWDULX1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**PIP Window Display Upper-Left corner X-coordinates [12:8]** 請參考 Main Window 座標。 單位: 像素。 必須要能被 4 整除。PWDULX Bit [1:0] 內部固定為 0。 X-軸座標應該要小於 horizontal display width。 根據 REG[10h] (Select Configure PIP 1 or 2 Window) 參數，這個設定值將為相關 PIP 的參數值。	0	RW

PAGE0 REG[2Ch] PIP 1 or 2 Window Display Upper-Left corner Y-coordinates 0 (PWDULY0)

Bit	Description	Default	Access
7-0	**PIP Window Display Upper-Left corner Y-coordinates [7:0]** 請參考 Main Window 座標。 單位: 像素。 Y-軸座標應該要小於 vertical display width。 根據 REG[10h] (Select Configure PIP 1 or 2 Window)參數，這個設定值將為相關 PIP 的參數值。	0	RW

PAGE0 REG[2Dh] PIP 1 or 2 Window Display Upper-Left corner Y-coordinates 1 (PWDULY1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**PIP Window Display Upper-Left corner Y-coordinates [12:8]** 請參考 Main Window 座標。 單位: 像素。 Y-軸座標應該要小於 vertical display width。 根據 REG[10h] (Select Configure PIP 1 or 2 Window) 參數，這個設定值將為相關 PIP 的參數值。	0	RW

PAGE0 REG[2Eh] PIP 1 or 2 Image Start Address 0 (PISA0)

Bit	Description	Default	Access
7-2	**PIP Image Start Address[7:2]** 根據 REG[10h] (Select Configure PIP 1 or 2 Window) 參數，這個設定值將為相關 PIP 的參數值。 必須要能被 4 整除。Bit [1:0] 內部固定為 0。	0	RW
1-0	**NA**	0	RO

PAGE0 REG[2Fh] PIP 1 or 2 Image Start Address 1 (PISA1)

Bit	Description	Default	Access
7-0	**PIP Image Start Address[15:8]** 根據 REG[10h] (Select Configure PIP 1 or 2 Window) 參數，這個設定值將為相關 PIP 的參數值。	0	RW

PAGE0 REG[30h] PIP 1 or 2 Image Start Address 2 (PISA2)

Bit	Description	Default	Access
7-0	**PIP Image Start Address [23:16]** 根據 REG[10h] (Select Configure PIP 1 or 2 Window) 參數，這個設定值將為相關 PIP 的參數值。	0	RW

PAGE0 REG[31h] PIP 1 or 2 Image Start Address 3 (PISA3)

Bit	Description	Default	Access
7-0	**PIP Image Start Address [31:24]** 根據 REG[10h] (Select Configure PIP 1 or 2 Window) 參數，這個設定值將為相關 PIP 的參數值。	0	RW

PAGE0 REG[32h] PIP 1 or 2 Image Width 0 (PIW0)

Bit	Description	Default	Access
7-2	**PIP Image Width [7:2]** 單位: 像素。 必須要能被 4 整除。PIW Bit [1:0] 內部固定為 0。 這個數值是物理上的像素值。 這個寬度應該要小於 horizontal display width。 根據 REG[10h] (Select Configure PIP 1 or 2 Window) 參數，這個設定值將為相關 PIP 的參數值。	0	RW
1-0	**NA**	0	RO

PAGE0 REG[33h] PIP 1 or 2 Image Width 1 (PIW1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**PIP Image Width [12:8]** 單位: 像素。 必須要能被 4 整除。PIW Bit [1:0] 內部固定為 0。 這個數值是物理上的像素值。 這個寬度應該要小於 horizontal display width。 根據 REG[10h] (Select Configure PIP 1 or 2 Window) 參數，這個設定值將為相關 PIP 的參數值。	0	RW

PAGE0 REG[34h] PIP 1 or 2 Window Image Upper-Left corner X-coordinates 0 (PWIULX0)

Bit	Description	Default	Access
7-2	**PIP 1 or 2 Window Image Upper-Left corner X-coordinates [7:0]** 請參考 PIP Image 座標。 單位: 像素。 必須要能被 4 整除。PWIULX Bit [1:0] 內部固定為 0。 X-軸 座標+ PIP image width 必須要小於或等於 8188。 根據 REG[10h] (Select Configure PIP 1 or 2 Window) 參數，這個設定值將為相關 PIP 的參數值。	0	RW
1-0	**NA**	0	RO

PAGE0 REG[35h] PIP 1 or 2 Window Image Upper-Left corner X-coordinates 1 (PWIULX1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**PIP Window Image Upper-Left corner X-coordinates [12:8]** 請參考 PIP Image 座標。 單位: 像素。 必須要能被 4 整除。PWIULX Bit [1:0] 內部固定為 0。 X-軸 座標+ PIP image width 必須要小於或等於 8188。 根據 REG[10h] (Select Configure PIP 1 or 2 Window) 參數，這個設定值將為相關 PIP 的參數值。	0	RW

PAGE0 REG[36h] PIP 1 or 2 Window Image Upper-Left corner Y-coordinates (PWIULY0)

Bit	Description	Default	Access
7-0	**PIP Windows Display Upper-Left corner Y-coordinates [7:0]** 請參考 PIP Image 座標。 單位: 像素。 Y-軸 座標+ PIP window height 必須要小於或等於 8191。 根據 REG[10h] (Select Configure PIP 1 or 2 Window) 參數，這個設定值將為相關 PIP 的參數值。	0	RW

PAGE0 REG[37h] PIP 1 or 2 Window Image Upper-Left corner Y-coordinates 1 (PWIULY1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**PIP Windows Image Upper-Left corner Y-coordinates [12:8]** 請參考 PIP Image 座標。 單位: 像素。 Y-軸 座標+ PIP window height 必須要小於或等於 8191。 根據 REG[10h] (Select Configure PIP 1 or 2 Window) 參數，這個設定值將為相關 PIP 的參數值。	0	RW

PAGE0 REG[38h] PIP 1 or 2 Window Width 0 (PWW0)

Bit	Description	Default	Access
7-2	**PIP Window Width [7:2]** 單位: 像素。 必須要能被 4 整除。PWW Bit [1:0] 內部固定為 0。 這個數值是物理上的像素值。最大值是 2044 像素。 根據 REG[10h] (Select Configure PIP 1 or 2 Window) 參數，這個設定值將為相關 PIP 的參數值。	0	RW
1-0	**NA**	0	RO

PAGE0 REG[39h] PIP 1 or 2 Window Width 1 (PWW1)

Bit	Description	Default	Access
7-3	**NA**	0	RO
2-0	**PIP Window Width [10:8]** 單位: 像素。 必須要能被 4 整除。這個數值是物理上的像素值。 最大值是 2044 像素。 根據 REG[10h] (Select Configure PIP 1 or 2 Window) 參數，這個設定值將為相關 PIP 的參數值。	0	RW

PAGE0 REG[3Ah] PIP 1 or 2 Window Height 0 (PWH0)

Bit	Description	Default	Access
7-0	**PIP Window Height [7:0]** 單位: 像素。 這個數值是物理上的像素值。最大值是 2047 像素。 根據 REG[10h] (Select Configure PIP 1 or 2 Window) 參數，這個設定值將為相關 PIP 的參數值。	0	RW

PAGE0 REG[3Bh] PIP 1 or 2 Windows Height 1 (PWH1)

Bit	Description	Default	Access
7-3	**NA**	0	RO
2-0	**PIP Window Height [10:8]** 單位: 像素。 這個數值是物理上的像素值。最大值是 2047 像素。 根據 REG[10h] (Select Configure PIP 1 or 2 Window) 參數，這個設定值將為相關 PIP 的參數值。	0	RW

註： PIP 視窗容量與起始位置在水平方向是以 8 個像素微解析度，垂直方向的解析度則是 1 個 line。

註： 上面的暫存器 20h~3Bh 需要依次由 LSB 寫到 MSB 才會生效。
假設我們需要設定 Main Image Start Address，此暫存器為地址 20h 到 23h，必須依次由 LSB[20h] 寫到 MSB[23h]，當 REG[23h] 被寫入時，RA8889 才會將 REG[20h]~REG[23h] 的值真正寫到內部暫存器中。

PAGE0 REG[3Ch] Graphic / Text Cursor Control Register (GTCCR)

Bit	Description	Default	Access
7	**Gamma correction Enable** 0: 禁能。 1: 致能。 **Gamma correction is the last output stage.**	0	RW
6-5	**Gamma table select for MPU write gamma data** 00b: 藍色的 Gamma table。 01b: 綠色的 Gamma table。 10b: 紅色的 Gamma table。 11b: NA。	0	RW
4	**Graphic Cursor Enable** 0 : Graphic Cursor 禁能。 1 : Graphic Cursor 致能。	0	RW
3-2	**Graphic Cursor Selection Bit** 從 4 種圖形游標中選擇 1 種。 00b : Graphic Cursor Set 1。 01b : Graphic Cursor Set 2。 10b : Graphic Cursor Set 3。 11b : Graphic Cursor Set 4。	0	RW
1	**Text Cursor Enable** 0 : 禁能。 1 : 致能。 文字游標與圖形游標無法同時被致能，若是同時被致能則圖形游標的優先權高於文字游標	0	RW
0	**Text Cursor Blinking Enable** 0 : 禁能。 1 : 致能。	0	RW

PAGE0 REG[3Dh] Blink Time Control Register (BTCR)

Bit	Description	Default	Access
7-0	**Text Cursor Blink Time Setting (Unit: Frame)** 00h : 1 frame 時間。 01h : 2 frames 時間。 02h : 3 frames 時間。 : FFh : 256 frames 時間。	0	RW

PAGE0 REG[3Eh] Text Cursor Horizontal Size Register (CURHS)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Text Cursor Horizontal Size Setting[4:0]** 單位: 像素。 Zero-based 的數字，數值“0” 表示 1 個像素。 ***註：*** 當字元被放大時，文字游標也會同時被放大。	07h	RW

PAGE0 REG[3Fh] Text Cursor Vertical Size Register (CURVS)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Text Cursor Vertical Size Setting[4:0]** 單位: 像素。 Zero-based 的數字，數值“0” 表示 1 個像素。 ***註：*** 當字元被放大時，文字游標也會同時被放大。	0	RW

PAGE0 REG[40h] Graphic Cursor Horizontal Position Register 0 (GCHP0)

Bit	Description	Default	Access
7-0	**Graphic Cursor Horizontal Location[7:0]** 請參考 Main Window 座標。	0	RW

PAGE0 REG[41h] Graphic Cursor Horizontal Position Register 1 (GCHP1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Graphic Cursor Horizontal Location[12:8]** 請參考 Main Window 座標。	0	RW

PAGE0 REG[42h] Graphic Cursor Vertical Position Register 0 (GCVP0)

Bit	Description	Default	Access
7-0	**Graphic Cursor Vertical Location[7:0]** 請參考 Main Window 座標。	0	RW

PAGE0 REG[43h] Graphic Cursor Vertical Position Register 1 (GCVP1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Graphic Cursor Vertical Location[12:8]** 請參考 Main Window 座標。	0	RW

PAGE0 REG[44h] Graphic Cursor Color 0 (GCC0)

Bit	Description	Default	Access
7-0	**Graphic Cursor Color 0 with 256 Colors** RGB Format [7:0] = RRRGGGBB.	0	RW

PAGE0 REG[45h] Graphic Cursor Color 1 (GCC1)

Bit	Description	Default	Access
7-0	**Graphic Cursor Color 1 with 256 Colors** RGB Format [7:0] = RRRGGGBB.	0	RW

PAGE0 REG[46h] Page Switch

Bit	Description	Default	Access
7-2	N/A	0	RW
1	**Must be 0**	0	RW
0	**Page switch** 0: page 0, 少於 256 個暫存器設定 1: page 1, 媒體解碼暫存器設定	0	RW

20.6 幾何引擎控制暫存器

PAGE0 REG[50h] Canvas Start address 0 (CVSSA0)

Bit	Description	Default	Access
7-2	**Start address of Canvas [7:2]** 如果底圖(canvas) 是 linear 模式，則可被忽略。	0	RW
1-0	固定為 0	0	RO

PAGE0 REG[51h] Canvas Start address 1 (CVSSA1)

Bit	Description	Default	Access
7-0	**Start address of Canvas [15:8]** 如果底圖(canvas) 是 linear 模式，則可被忽略。	0	RW

PAGE0 REG[52h] Canvas Start address 2 (CVSSA2)

Bit	Description	Default	Access
7-0	**Start address of Canvas [23:16]** 如果底圖(canvas) 是 linear 模式，則可被忽略。	0	RW

PAGE0 REG[53h] Canvas Start address 3 (CVSSA3)

Bit	Description	Default	Access
7-0	**Start address of Canvas [31:24]** 如果底圖(canvas) 是 linear 模式，則可被忽略。	0	RW

PAGE0 REG[54h] Canvas image width 0 (CVS_IMWTH0)

Bit	Description	Default	Access
7-2	**Canvas image width [7:2]** 這些 bits 是底圖(Canvas)的寬度。 單位: 像素，是以 4 個像素為解析度。 寬度=設定值。 如果底圖(canvas) 是 linear 模式，則可被忽略。	0	RW
1-0	固定為 0	0	RO

PAGE0 REG[55h] Canvas image width 1 (CVS_IMWTH1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Canvas image width [12:8]** **The bits are Canvas image width** 如果底圖(canvas) 是 linear 模式，則可被忽略。	0	RW

PAGE0 REG[56h] Active Window Upper-Left corner X-coordinates 0 (AWUL_X0)

Bit	Description	Default	Access
7-0	**Active Window Upper-Left corner X-coordinates [7:0]** 請參考 Canvas image 座標。 單位: 像素。 如果底圖 (canvas) 是 linear 模式，則可被忽略。	0	RW

PAGE0 REG[57h] Active Window Upper-Left corner X-coordinates 1 (AWUL_X1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Active Window Upper-Left corner X-coordinates [12:8]** 請參考 Canvas image 座標。 單位: 像素。 如果底圖 (canvas) 是 linear 模式，則可被忽略。	0	RW

PAGE0 REG[58h] Active Window Upper-Left corner Y-coordinates 0 (AWUL_Y0)

Bit	Description	Default	Access
7-0	**Active Window Upper-Left corner Y-coordinates [7:0]** 請參考 Canvas image 座標。 單位: 像素。 如果底圖 (canvas) 是 linear 模式，則可被忽略。	0	RW

PAGE0 REG[59h] Active Window Upper-Left corner Y-coordinates 1 (AWUL_Y1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Active Window Upper-Left corner Y-coordinates [12:8]** 請參考 Canvas image 座標。 單位: 像素。 如果底圖 (canvas) 是 linear 模式，則可被忽略。	0	RW

PAGE0 REG[5Ah] Active Window Width 0 (AW_WTH0)

Bit	Description	Default	Access
7-0	**Width of Active Window [7:0]** 單位: 像素。 這個數值是物理上的像素值。 如果底圖 (canvas) 是 linear 模式，則可被忽略。	0	RW

PAGE0 REG[5Bh] Active Window Width 1 (AW_WTH1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Width of Active Window [12:8]** 單位: 像素。 這個數值是物理上的像素值。 如果底圖 (canvas) 是 linear 模式，則可被忽略。	0	RW

PAGE0 REG[5Ch] Active Window Height 0 (AW_HT0)

Bit	Description	Default	Access
7-0	**Height of Active Window [7:0]** 單位: 像素。 這個數值是物理上的像素值。 如果底圖 (canvas) 是 linear 模式，則可被忽略。	0	RW

PAGE0 REG[5Dh] Active Window Height 1 (AW_HT1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Height of Active Window [12:8]** 單位: 像素。 這個數值是物理上的像素值。 如果底圖 (canvas) 是 linear 模式，則可被忽略。	0	RW

PAGE0 REG[5Eh] Color Depth of Canvas & Active Window (AW_COLOR)

Bit	Description	Default	Access
7-4	**NA**	0	RO
3	**NA**	0	RO
2	**Canvas addressing mode** 0: Block 模式 (X-Y 座標定址方法)。 1: Linear 模式。	0	RW
1-0	**Canvas image's color depth & memory R/W data width** **In Block Mode:** 00: 8bpp。 01: 16bpp。 1x: 24bpp。 *註*: 單色資料的輸入方法，可以使用任何一個色深，並搭配適合的圖像寬度，即可正確輸入。 **In Linear Mode:** X0: 8-bit 記憶體資料讀寫。 X1: 16-bit 記憶體資料讀寫。	0	RW

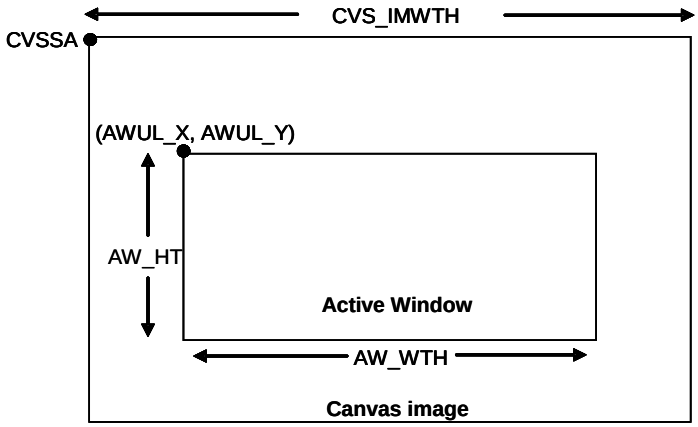


Figure 20-4 : Active Window

PAGE0 REG[5Fh] Graphic Read/Write position Horizontal Position Register 0 (CURH0)

Bit	Description	Default	Access
7-0	**Write: Set Graphic Read/Write position** **When Canvas In Linear mode:** 記憶體的讀寫位址 [7:0]。 單位: Byte。 **When Canvas In Block mode:** 圖形讀寫水平位置 0 [7:0]。 請參考 Canvas image 座標。 單位: 像素。	0	RW

註: 在設定此寄存器之前，用戶應設定好活動視窗(active window)相關參數

PAGE0 REG[60h] Graphic Read/Write position Horizontal Position Register 1 (CURH1)

Bit	Description	Default	Access
7-5	**Write: Set Graphic Read/Write position** **When Canvas In Linear mode:** 記憶體的讀寫位址 [15:13]。 單位: Byte。 **When Canvas In Block mode:** NA 請參考 Canvas image 座標。 單位: 像素。	0	RW
4-0	**Write: Set Graphic Read/Write position** **When Canvas In Linear mode:** 記憶體的讀寫位址 [12:8]。 單位: Byte。 **When Canvas In Block mode:** 圖形讀寫水平位置 1 [12:8] 請參考 Canvas image 座標。 單位: 像素。	0	RW

註: 在設定此寄存器之前，用戶應設定好活動視窗(active window)相關參數

PAGE0 REG[61h] Graphic Read/Write position Vertical Position Register 0 (CURV0)

Bit	Description	Default	Access
7-0	**Write: Set Graphic Read/Write position** **When Canvas In Linear mode:** 記憶體的讀寫位址 [23:16] 單位: Byte **When Canvas In Block mode:** 圖形讀寫垂直位置 0 [7:0] 請參考 Canvas image 座標。 單位: 像素。	0	RW

註: 在設定此寄存器之前，用戶應設定好活動視窗(active window)相關參數

PAGE0 REG[62h] Graphic Read/Write position Vertical Position Register 1 (CURV1)

Bit	Description	Default	Access
7-5	**Write: Set Graphic Read/Write position** **When Canvas In Linear mode:** 記憶體的讀寫位址 [31:29]。 單位: Byte。 **When Canvas In Block mode:**NA 請參考 Canvas image 座標。 單位: 像素。	0	RW
4-0	**Write: Set Graphic Read/Write position** **When Canvas In Linear mode:** 記憶體的讀寫位址 [28:24]。 單位: Byte。 **When Canvas In Block mode:** 圖形讀寫垂直位置 1 [12:8]。 請參考 Canvas image 座標。 單位: 像素。	0	RW

註: 在設定此寄存器之前，用戶應設定好活動視窗(active window)相關參數

PAGE0 REG[63h] Text Write X-coordinates Register 0 (F_CURX0)

Bit	Description	Default	Access
7-0	**Write: Set Text Write position** **Read: Current Text Write position** **Text Write X-coordinates [7:0]** 這是設定文字寫入的水平游標位置。 請參考 Canvas image 座標。 單位: 像素。	0	RW

PAGE0 REG[64h] Text Write X-coordinates Register 1 (F_CURX1)

Bit	Description	Default	Access
7-5	NA	0	RO
4-0	**Write: Set Text Write position** **Read: Current Text Write position** **Text Write X-coordinates [12:8]** 這是設定文字寫入的水平游標位置。 請參考 Canvas image 座標。 單位: 像素。	0	RW

PAGE0 REG[65h] Text Write Y-coordinates Register 0 (F_CURY0)

Bit	Description	Default	Access
7-0	**Write: Set Text Write position** **Read: Current Text Write position** **Text Write Y-coordinates [7:0]** 這是設定文字寫入的垂直游標位置。 請參考 Canvas image 座標。 單位: 像素。	0	RW

PAGE0 REG[66h] Text Write Y-coordinates Register 1 (F_CURY1)

Bit	Description	Default	Access
7-5	NA	0	RO
4-0	**Write: Set Text Write position** **Read: Current Text Write position** **Text Write Y-coordinates [12:8]** 這是設定文字寫入的垂直游標位置。 請參考 Canvas image 座標。 單位: 像素。	0	RW

PAGE0 REG[67h] Draw Line / Triangle Control Register 0 (DCR0)

Bit	Description	Default	Access
7	**Draw Line / Triangle Start Signal** **Write Function** 0：停止繪圖。 1：開始繪圖。 **Read Function** 0：繪圖完成。 1：繪圖進行中。	0	RW
6	**NA**	0	RO
5	**Fill function for Triangle Signal** 0：無填滿。 1：填滿。	0	RW

Bit	Description	Default	Access
4-2	**NA**	0	RO
1	**Draw Triangle or Line Select Signal** 0：畫線。 1：畫三角。	0	RW
0	**Must set 0**	0	RO

PAGE0 REG[68h] Draw Line/Square/Triangle Point 1 X-coordinates Register0 (DLHSR0)

Bit	Description	Default	Access
7-0	**Draw Line/Triangle Point 1 X-coordinates [7:0]** **Square diagonal Point 1 X-coordinates [7:0]** 請參考 Canvas image 座標。 單位: 像素。 ***註: 當繪製矩形時，起始點與結束點不可在圖一位置，起始點與結束點也不可同時在 X-軸或 Y-軸。	0	RW

PAGE0 REG[69h] Draw Line/Square/Triangle Point 1 X-coordinates Register1 (DLHSR1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Draw Line/Triangle Point 1 X-coordinates [12:8]** **Square diagonal Point 1 X-coordinates [12:8]** 請參考 Canvas image 座標。 單位: 像素。 ***註: 當繪製矩形時，起始點與結束點不可在圖一位置，起始點與結束點也不可同時在 X-軸或 Y-軸。	0	RW

PAGE0 REG[6Ah] Draw Line/Square/Triangle Point 1 Y-coordinates Register0 (DLVSR0)

Bit	Description	Default	Access
7-0	**Draw Line/Triangle Point 1 Y-coordinates [7:0]** **Square diagonal Point 1 Y-coordinates [7:0]** 請參考 Canvas image 座標。 單位: 像素。 ***註: 當繪製矩形時，起始點與結束點不可在圖一位置，起始點與結束點也不可同時在 X-軸或 Y-軸。	0	RW

PAGE0 REG[6Bh] Draw Line/Square/Triangle Point 1 Y-coordinates Register1 (DLVSR1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Draw Line/Triangle Point 1 Y-coordinates [12:8]** **Square diagonal Point 1 Y-coordinates [12:8]** 請參考 Canvas image 座標。 單位: 像素。 ***註: 當繪製矩形時，起始點與結束點不可在圖一位置，起始點與結束點也不可同時在 X-軸或 Y-軸。	0	RW

PAGE0 REG[6Ch] Draw Line/Square/Triangle Point 2 X-coordinates Register0 (DLHER0)

Bit	Description	Default	Access
7-0	**Draw Line/Triangle Point 2 X-coordinates [7:0]** **Square diagonal Point 2 X-coordinates [7:0]** 請參考 Canvas image 座標。 單位: 像素。 ***註: 當繪製矩形時，起始點與結束點不可在圖一位置，起始點與結束點也不可同時在 X-軸或 Y-軸。	0	RW

PAGE0 REG[6Dh] Draw Line/Square/Triangle Point 2 X-coordinates Register1 (DLHER1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Draw Line/Triangle Point 2 X-coordinates [12:8]** **Square diagonal Point 2 X-coordinates [12:8]** 請參考 Canvas image 座標。 單位: 像素。 ***註: 當繪製矩形時，起始點與結束點不可在圖一位置，起始點與結束點也不可同時在 X-軸或 Y-軸。	0	RW

PAGE0 REG[6Eh] Draw Line/Square/Triangle Point 2 Y-coordinates Register0 (DLVER0)

Bit	Description	Default	Access
7-0	**Draw Line/Triangle Point 2 Y-coordinates [7:0]** **Square diagonal Point 2 Y-coordinates [7:0]** 請參考 Canvas image 座標。 單位: 像素。 ***註: 當繪製矩形時，起始點與結束點不可在圖一位置，起始點與結束點也不可同時在 X-軸或 Y-軸。	0	RW

PAGE0 REG[6Fh] Draw Line/Square/Triangle Point 2 Y-coordinates Register1 (DLVER1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Draw Line/Triangle Point 2 Y-coordinates [12:8]** **Square diagonal Point 2 Y-coordinates [12:8]** 請參考 Canvas image 座標。 單位: 像素。 ***註*: 當繪製矩形時，起始點與結束點不可在圖一位置，起始點與結束點也不可同時在 X-軸或 Y-軸。	0	RW

PAGE0 REG[70h] Draw Triangle Point 3 X-coordinates Register 0 (DTPH0)

Bit	Description	Default	Access
7-0	**Draw Triangle Point 3 X-coordinates [7:0]** 請參考 Canvas image 座標。 單位: 像素。	0	RW

PAGE0 REG[71h] Draw Triangle Point 3 X-coordinates Register 1 (DTPH1)

Bit	Description	Default	Access
7-5	NA	0	RO
4-0	**Draw Triangle Point 3 X-coordinates [12:8]** 請參考 Canvas image 座標。 單位: 像素。	0	RW

PAGE0 REG[72h] Draw Triangle Point 3 Y-coordinates Register 0 (DTPV0)

Bit	Description	Default	Access
7-0	**Draw Triangle Point 3 Y-coordinates [7:0]** 請參考 Canvas image 座標。 單位: 像素。	0	RW

PAGE0 REG[73h] Draw Triangle Point 3 Y-coordinates Register 1 (DTPV1)

Bit	Description	Default	Access
7-5	NA	0	RO
4-0	**Draw Triangle Point 3 Y-coordinates [12:8]** 請參考 Canvas image 座標。 單位: 像素。	0	RW

***註: 關於三角形的三點設定:

1. 任兩點重疊會畫出直線。
2. 三點重疊會畫出一個點。

PAGE0 REG[74h – 75h] RESERVED

Bit	Description	Default	Access
7-0	**NA**	0	RO

PAGE0 REG[76h] Draw Circle/Ellipse/Ellipse Curve/Circle Square Control Register 1 (DCR1)

Bit	Description	Default	Access
7	**Draw Circle / Ellipse / Square /Circle Square Start Signal** **Write Function** 0：停止繪圖。 1：開始繪圖。 **Read Function** 0：繪圖完成。 1：繪圖進行中。	0	RW
6	**Fill the Circle / Ellipse / Square / Circle Square Signal** 0：無填滿。 1：填滿。	0	RW
5-4	**Draw Circle / Ellipse / Square / Ellipse Curve / Circle Square Select** 00：畫圓/橢圓(Circle / Ellipse)。 01：畫圓/曲線(Circle / Ellipse Curve)。 10：畫矩形 (Square)。 11：畫圓角矩形(Circle Square)。	0	RW
3-2	NA	0	RO
1-0	**Draw Circle / Ellipse Curve Part Select(DECP)** 00: 左下方曲線(Ellipse Curve)。 01: 左上方曲線(Ellipse Curve)。 10: 右上方曲線(Ellipse Curve)。 11: 右下方曲線(Ellipse Curve)。	0	RW

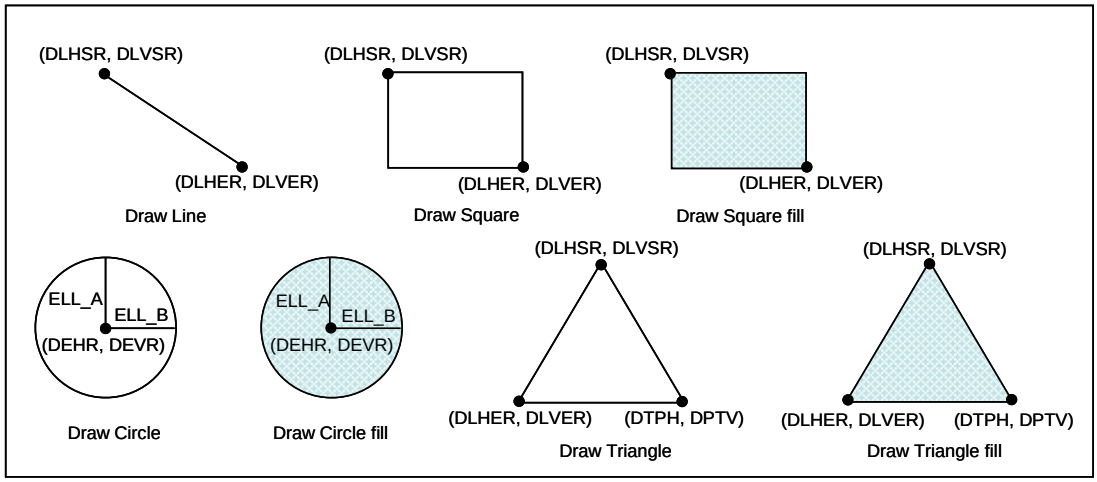


Figure 20-5 : Drawing Function Parameter

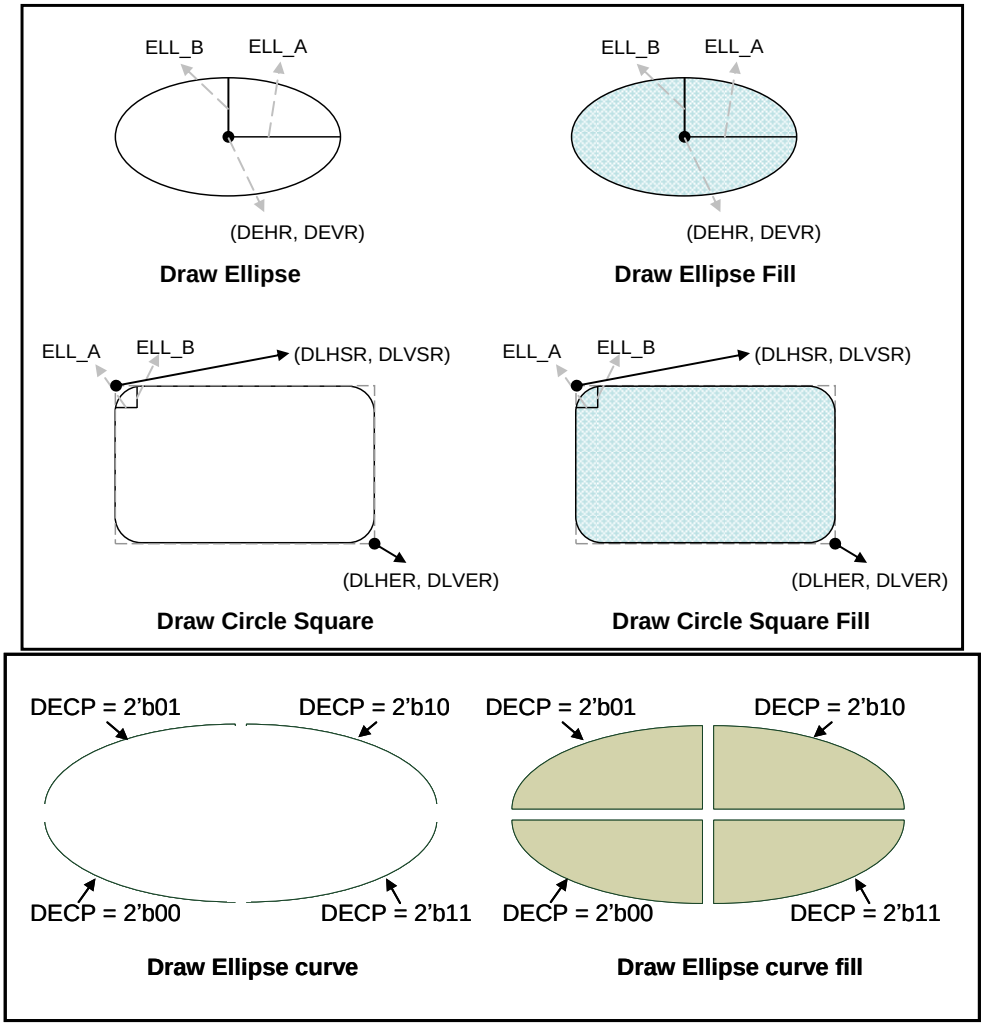


Figure 20-6 : The Drawing Function

PAGE0 REG[77h] Draw Circle/Ellipse/Circle Square Major radius Setting Register (ELL_A0)

Bit	Description	Default	Access
7-0	**Draw Circle/Ellipse/Circle Square Major radius [7:0]** 單位: 像素。 畫圓需要設定長(major) 短(minor) 軸相等	0	RW

PAGE0 REG[78h] Draw Circle/Ellipse/Circle Square Major radius Setting Register (ELL_A1)

Bit	Description	Default	Access
7-5	NA	0	RO
4-0	**Draw Circle/Ellipse/Circle Square Major radius [12:8]** 單位: 像素。 畫圓需要設定長(major) 短(minor) 軸相等。	0	RW

PAGE0 REG[79h] Draw Circle/Ellipse/Circle Square Minor radius Setting Register (ELL_B0)

Bit	Description	Default	Access
7-0	**Draw Circle/Ellipse/Circle Square Minor radius [7:0]** 單位: 像素。 畫圓需要設定長(major) 短(minor) 軸相等。	0	RW

PAGE0 REG[7Ah] Draw Circle/Ellipse/Circle Square Minor radius Setting Register (ELL_B1)

Bit	Description	Default	Access
7-5	NA	0	RO
4-0	**Draw Circle/Ellipse/Circle Square Minor radius [12:8]** 單位: 像素。 畫圓需要設定長(major) 短(minor) 軸相等。	0	RW

PAGE0 REG[7Bh] Draw Circle/Ellipse/Circle Square Center X-coordinates Register0 (DEHR0)

Bit	Description	Default	Access
7-0	**Draw Circle/Ellipse/Circle Square Center X-coordinates [7:0]** 請參考 Canvas image 座標。 單位: 像素。	0	RW

PAGE0 REG[7Ch] Draw Circle/Ellipse/Circle Square Center X-coordinates Register1 (DEHR1)

Bit	Description	Default	Access
7-5	NA	0	RO
4-0	**Draw Circle/Ellipse/Circle Square Center X-coordinates [12:8]** 請參考 Canvas image 座標。 單位: 像素。	0	RW

PAGE0 REG[7Dh] Draw Circle/Ellipse/Circle Square Center Y-coordinates Register0 (DEVR0)

Bit	Description	Default	Access
7-0	**Draw Circle/Ellipse/Circle Square Center Y-coordinates [7:0]** 請參考 Canvas image 座標。 單位: 像素。	0	RW

PAGE0 REG[7Eh] Draw Circle/Ellipse/Circle Square Center Y-coordinates Register1 (DEVR1)

Bit	Description	Default	Access
7-5	NA	0	RO
4-0	**Draw Circle/Ellipse/Circle Square Center Y-coordinates [12:8]** 請參考 Canvas image 座標。 單位: 像素。	0	RW

20.7 脈寬調變控制暫存器

PAGE0 REG[84h] PWM Prescaler Register (PSCLR)

Bit	Description	Default	Access
7-0	**PWM Prescaler Register** 此暫存器為 Timer 0 及 Timer 1 的 prescaler 值。 基頻是 “Core_Freq / (Prescaler + 1)”	0	RW

PAGE0 REG[85h] PWM clock Mux Register (PMUXR)

Bit	Description	Default	Access
7-6	**Select 2nd clock divider’s MUX input for PWM Timer 1** 00 = 1 01 = 1/2 10 = 1/4 11 = 1/8	0	RW
5-4	**Select 2nd clock divider’s MUX input for PWM Timer 0** 00 = 1 01 = 1/2 10 = 1/4 11 = 1/8	0	RW
3-2	**XPWM[1] pin function control** 0X: XPWM[1] 輸出系統錯誤旗標 (Scan FIFO pop 錯誤或是記憶體存取超過範圍)。 10: XPWM[1] 輸出PWM 計數器1的波形或是PWM 計數器0 的反相波形 (dead zone 致能)。 11: XPWM[1] 輸出oscillator 時脈。 如果XTEST[0] 為high，則XPWM[1] 將會是螢幕掃描時脈的輸入。	0	RW
1-0	**XPWM[0] pin function control** 0X: XPWM[0] 為GPIO-C[7]。 10: XPWM[0] 輸出PWM 計數器0。 11: XPWM[0] 輸出系統時脈。	0	RW

PAGE0 REG[86h] PWM Configuration Register (PCFGR)

Bit	Description	Default	Access
7	NA	0	RO
6	**PWM Timer 1 output inverter on/off** 計數器1的輸出是否反相。 0 = 反相關閉。 1 = PWM1反相開啟。	0	RW
5	**PWM Timer 1 auto reload on/off** 計數器1是否自動重載。 0 = 單擊 (one-shot)。 1 = 內部模式 (自動重載)。	1	RW

Bit	Description	Default	Access
4	**PWM Timer 1 start/stop** 計數器1開始/停止。 0 = 停止。 1 = 開始。 -- 在內部模式 (自動重載)，使用者若要停止PWM 計數器，則必須寫0。 --在單擊 (One-shot) 功能中，這個bit 會自動被清除。 使用者可以讀取這個bit，以便得知PWMx 是執行中還是停止中。	0	RW
3	**PWM Timer 0 Dead-Zone enable** Determine the Dead-Zone operation。 0 = 禁能。 1 = 致能。	0	RW
2	**PWM Timer 0 output inverter on/off** 計數器0 的輸出反相。 0 = 反相關閉。 1 = PWM0 的反相。	0	RW
1	**PWM Timer 0 auto reload on/off** 計數器0 的自動重載開啟與關閉。 0 = 單擊 (One-shot)。 1 = 內部模式 (自動重載)。	1	RW
0	**PWM Timer 0 start/stop** 計數器0 的開始與停止。 0 = 停止。 1 = 開始。 -- 在內部模式 (自動重載)，使用者若是要停止PWM 計數器，則需設定這個bit 為0。 -- 在單擊 (One-shot) 模式，這個bit會自動被清除。 使用者可以讀取這個 bit，以便得知 PWMx 是執行中還是停止中。	0	RW

PAGE0 REG[87h] Timer 0 Dead zone length register [DZ_LENGTH]

Bit	Description	Default	Access
7-0	**Timer 0 Dead-Zone length register** 此 8bits 為 Dead-Zone 的長度，以計數器 0 的計數完整的一個週期為 Dead-Zone 的一個單位時間長度。	0	RW

PAGE0 REG[88h] Timer 0 compare buffer register [TCMPB0L]

Bit	Description	Default	Access
7-0	**Timer 0 compare buffer register --- Low Byte** 比較緩衝 0 暫存器總共是 16bits，當計數器等於或小於比較緩衝 0 暫存器的值，並且在 PWM 計數器 0 反相關閉情況下，PWM0 輸出為 high。	0	RW

PAGE0 REG[89h] Timer 0 compare buffer register [TCMPB0H]

Bit	Description	Default	Access
7-0	**Timer 0 compare buffer register --- High Byte** 比較緩衝 0 暫存器總共是 16bits，當計數器等於或小於比較緩衝 0 暫存器的值，並且在 PWM 計數器 0 反相關閉情況下，PWM0 輸出為 high。	0	RW

PAGE0 REG[8Ah] Timer 0 count buffer register [TCNTB0L]

Bit	Description	Default	Access
7-0	**Timer 0 count buffer register --- Low Byte** 計數緩衝 0 暫存器總共有 16bit。當計數器等於 0 時，並且 reload_en 是致能的情況下，PWM 會重載計數緩衝 0 暫存器的值到計數器中。當 PWM 開始計數後，可以透過這個暫存器讀回目前的計數值。	0	RW

PAGE0 REG[8Bh] Timer 0 count buffer register [TCNTB0H]

Bit	Description	Default	Access
7-0	**Timer 0 count buffer register --- High Byte** 計數緩衝 0 暫存器總共有 16bit。當計數器等於 0 時，且 reload_en 是致能的情況下，PWM 會重載計數緩衝 0 暫存器的值到計數器中。當 PWM 開始計數後，可以透過這個暫存器讀回目前的計數值。	0	RW

PAGE0 REG[8Ch] Timer 1 compare buffer register [TCMPB1L]

Bit	Description	Default	Access
7-0	**Timer 1 compare buffer register --- Low Byte** 比較緩衝 1 暫存器總共是 16bits，當計數器等於或小於比較緩衝 1 暫存器的值，並且在 PWM 計數器 1 反相關閉情況下，PWM0 輸出為 high。	0	RW

PAGE0 REG[8Dh] Timer 1 compare buffer register [TCMPB1H]

Bit	Description	Default	Access
7-0	**Timer 1 compare buffer register --- High Byte** 比較緩衝 1 暫存器總共是 16bits，當計數器等於或小於比較緩衝 1 暫存器的值，並且在 PWM 計數器 1 反相關閉情況下，PWM0 輸出為 high。	0	RW

PAGE0 REG[8Eh] Timer 1 count buffer register [TCNTB1L]

Bit	Description	Default	Access
7-0	**Timer 1 count buffer register --- Low Byte** 計數緩衝 1 暫存器總共有 16bit。當計數器等於 0 時，並且 reload_en 是致能的情況下，PWM 會重載計數緩衝 1 暫存器的值到計數器中。當 PWM 開始計數後，可以透過這個暫存器讀回目前的計數值。	0	RW

PAGE0 REG[8Fh] Timer 1 count buffer register [TCNTB1H]

Bit	Description	Default	Access
7-0	**Timer 1 count buffer register --- High Byte** 計數緩衝 1 暫存器總共有 16bit。當計數器等於 0 時，並且 reload_en 是致能的情況下，PWM 會重載計數緩衝 1 暫存器的值到計數器中。當 PWM 開始計數後，可以透過這個暫存器讀回目前的計數值。	0	RW

20.8 區塊傳輸引擎 (BTE) 控制暫存器

PAGE0 REG[90h] BTE Function Control Register 0 (BTE_CTRL0)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4	**BTE Function Enable / Status** **Write** 0 : 無動作。 1 : BTE 致能。 **Read** 0 : BTE 閒置。 1 : BTE 忙碌。 *** 當 BTE 致能時，MPU 對底圖 (Canvas[工作視窗]) 記憶體的存取將不被允許。	0	RW
3-1	**NA**	0	RO
0	**PATTERN Format** 0: 8X8 1: 16X16	0	RW

PAGE0 REG[91h] BTE Function Control Register1 (BTE_CTRL1)

Bit	Description	Default	Access
7-4	**BTE ROP Code Bit[3:0] or Color expansion starting bit** **a.** ROP 是光柵操作的縮寫，某些 BTE 操作可以結合 ROP 的操作。(請參考章節 2.7)	0	RW

Code	Description
0000b	0 (Blackness)
0001b	~S0・~S1 or ~ (S0+S1)
0010b	~S0・S1
0011b	~S0
0100b	S0・~S1
0101b	~S1
0110b	S0^S1
0111b	~S0+~S1 or ~ (S0・S1)
1000b	S0・S1
1001b	~ (S0^S1)
1010b	S1
1011b	~S0+S1
1100b	S0
1101b	S0+~S1
1110b	S0+S1
1111b	1 (Whiteness)

b. 如果 BTE 操作在 color expansion (08h / 09h / Eh / Fh)。那麼這些 bits 指定每行第一筆 MPU 寫入單色資料的起始 bit，而這每行第一筆資料是 BTE 視窗左側邊緣的資料。並且其容量與 MPU 介面設定有關，因此若是在 8-bits MPU 介面上，其數值應該是 0 到 7，若是在 16-bits MPU 介面上，則數值為 0 到 15。

<table>
<tr><th>Bit</th><th>Description</th><th>Default</th><th>Access</th></tr>
<tr><td>3-0</td><td>
BTE Operation Code Bit[3:0]
RA8889 內建 2D BTE 引擎。此功能可以提供 13 BTE 操作。有些操作可以結合 ROP 功能。
<table>
<tr><th>Code</th><th>Description</th></tr>
<tr><td>0000b</td><td>MPU Write with ROP
S0: 由 MPU 輸入資料。
S1: 由記憶體提供資料。
D: 參考 ROP 功能並寫入目的記憶體中。</td></tr>
<tr><td>0001b</td><td>Reserved</td></tr>
<tr><td>0010b</td><td>Memory Copy with ROP
S0: 由記憶體提供資料。
S1: 由記憶體提供資料。
D: 參考 ROP 功能並寫入目的記憶體中。</td></tr>
<tr><td>0011b</td><td>Reserved</td></tr>
<tr><td>0100b</td><td>MPU Write w/ chroma keying (w/o ROP)
S0: 由 MPU 輸入資料。
如果 MPU 資料與 chroma key(background color 暫存器)
顏色不相同，那麼資料將會被寫入目的記憶體中。</td></tr>
<tr><td>0101b</td><td>Memory Copy (move) w/ chroma keying (w/o ROP)
S0 資料由記憶體來，並且不需要 S1 。
If S0 data doesn’t match with chroma key color (specified by background color) then S0 data will write to destination.</td></tr>
<tr><td>0110b</td><td>Pattern Fill with ROP
S0 資料來源為 Pattern。</td></tr>
<tr><td>0111b</td><td>Pattern Fill with chroma keying
S0 資料來源為 Pattern。
如果 S0 的 data 與 chroma key (background color) 顏色不同時，則將資料寫入目的記憶體中。</td></tr>
<tr><td>1000b</td><td>MPU Write w/ Color Expansion
S0 的需要的單色資料由 MPU 寫入，BTE 將其轉為指定的顏色與色深，並且寫入目的記憶體中。</td></tr>
<tr><td>1001b</td><td>MPU Write w/ Color Expansion and chroma keying
S0 的需要的單色資料由 MPU 寫入，如果單色資料的 bit 為 1，處理完的資料是前景色，如果單色資料為 0，那麼就不寫入。資料寫入目的記憶體中也會參考色深設定。</td></tr>
<tr><td>1010b</td><td>Memory Copy with opacity
S0, S1 & D: 來源與目的皆是記憶體。</td></tr>
<tr><td>1011b</td><td>MPU Write with opacity
S0: 由 MPU 輸入資料。
S1: 由記憶體提供資料。
D: 參考 Alpha blending 操作並寫入目的記憶體中。</td></tr>
<tr><td>1100b</td><td>Solid Fill
填滿矩形。寫入的值為暫存器設定值，寫入的目標為目的記憶體。</td></tr>
<tr><td>1101b</td><td>Reserved</td></tr>
<tr><td>1110b</td><td>Memory Copy w/ Color Expansion
S0 & D 位於記憶體，S1 未使用。
S0 的單色圖資須透過 MPU 或 DMA 以 8bpp 或 16bpp 的色深方式預載入記憶體。</td></tr>
</table>
</td><td>0</td><td>RW</td></tr>
</table>

Bit	Description		Default	Access
	1111b	**Memory Copy w/ Color Expansion and chroma keying** S0 & D 位於記憶體，S1 未使用。 S0 的單色圖資須透過 MPU 或 DMA 以 8bpp 或 16bpp 的色深方式預載入記憶體。 如果 S0 的位元資料=0 則 D 不會寫入任何資料. 如果 S0 的位元資料=1 則會將前景色寫入 D。		

PAGE0 REG[92h] Source 0/1 & Destination Color Depth (BTE_COLR)

Bit	Description	Default	Access
7	N/A	0	RO
6-5	**S0 Color Depth** 00: 256 色 (8bpp)。 01: 65k 色 (16bpp)。 1x: 16.7M 色 (24bpp)。	0	RW
4-2	**S1 Color Depth** 000: 256 色 (8bpp)。 001: 65k 色 (16bpp)。 010: 16.7M 色 (24bpp)。 011: Constant color (S1 memory start address' setting definition change as S1 constant color definition)。 100: 8 bit pixel alpha blending。 101: 16 bit pixel alpha blending。 110: 32bit ARGB mode	0	RW
1-0	**Destination Color Depth** 00: 256 色 (8bpp)。 01: 65k 色 (16bpp)。 1x: 16.7M 色 (24bpp)。	0	RW

PAGE0 REG[93h] Source 0 memory start address 0 (S0_STR0)

Bit	Description	Default	Access
7-2	**Source 0 memory start address [7:2]**	0	RW
1-0	**Fix at 0**	0	RO

PAGE0 REG[94h] Source 0 memory start address 1 (S0_STR1)

Bit	Description	Default	Access
7-0	**Source 0 memory start address [15:8]**	0	RW

PAGE0 REG[95h] Source 0 memory start address 2 (S0_STR2)

Bit	Description	Default	Access
7-0	**Source 0 memory start address [23:16]**	0	RW

PAGE0 REG[96h] Source 0 memory start address 3 (S0_STR3)

Bit	Description	Default	Access
7-0	**Source 0 memory start address [31:24]**	0	RW

PAGE0 REG[97h] Source 0 image width 0 (S0_WTH0)

Bit	Description	Default	Access
7-2	**Source 0 image width [7:2]** 單位: 像素。 必須要能被 4 整除。 S0_WTH Bit [1:0] 內部固定為 0。 這個數值是物理上的像素值。	0	RW
1-0	**Fix at 0**	0	RO

PAGE0 REG[98h] Source 0 image width 1 (S0_WTH1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Source 0 image width [12:8]** 單位: 像素。 必須要能被 4 整除。 S0_WTH Bit [1:0] 內部固定為 0。 這個數值是物理上的像素值。	0	RW

PAGE0 REG[99h] Source 0 Window Upper-Left corner X-coordinates 0 (S0_X0)

Bit	Description	Default	Access
7-0	**Source 0 Window Upper-Left corner X-coordinates [7:0]** 此暫存器是 Source 0 視窗左上角的 X 座標。	0	RW

PAGE0 REG[9Ah] Source 0 Window Upper-Left corner X-coordinates 1 (S0_X1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Source 0 Window Upper-Left corner X-coordinates [12:8]** 此暫存器是 Source 0 視窗左上角的 X 座標。	0	RW

PAGE0 REG[9Bh] Source 0 Window Upper-Left corner Y-coordinates 0 (S0_Y0)

Bit	Description	Default	Access
7-0	**Source 0 Window Upper-Left corner Y-coordinates [7:0]** 此暫存器是 Source 0 視窗左上角的 Y 座標。	0	RW

PAGE0 REG[9Ch] Source 0 Window Upper-Left corner Y-coordinates 1 (S0_Y1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Source 0 Window Upper-Left corner Y-coordinates [12:8]** 此暫存器是 Source 0 視窗左上角的 Y 座標。	0	RW

PAGE0 REG[9Dh] Source 1 memory start address 0 (S1_STR0) / S1 constant color – Red element (S1_Red)

Bit	Description	Default	Access
7-0	**Source 1 memory start address [7:2]** 如果 source 1 被設定為常數顏色，那麼此暫存器將會被定義為 S1 的常數顏色，此暫存器為紅色成分。當此暫存器為 **source 1 記憶體起始位置時，bit [1:0] 應該被設為 0**。	0	RW

PAGE0 REG[9Eh] Source 1 memory start address 1 (S1_STR1) / S1 constant color – Green element (S1_GREEN)

Bit	Description	Default	Access
7-0	**Source 1 memory start address [15:8]** 如果 source 1 被設定為常數顏色，那麼此暫存器將會被定義為 S1 的常數顏色，此暫存器將為綠色成分。	0	RW

PAGE0 REG[9Fh] Source 1 memory start address 2 (S1_STR2) / S1 constant color – Blue element (S1_BLUE)

Bit	Description	Default	Access
7-0	**Source 1 memory start address [23:16]** 如果 source 1 被設定為常數顏色，那麼此暫存器將會被定義為 S1 的常數顏色，此暫存器將為藍色成分。	0	RW

PAGE0 REG[A0h] Source 1 memory start address 3 (S1_STR3)

Bit	Description	Default	Access
7-0	**Source 1 memory start address [31:24]** 如果 source 1 被設定為常數顏色，那麼此暫存器將會被定義為 S1 的常數顏色。此暫存器將為不為顏色成分。	0	RW

REG[A1h] Source 1 image width 0 (S1_WTH0)

Bit	Description	Default	Access
7-2	**Source 1 image width [7:2]** 單位: 像素。 必須要能被 4 整除。 S1_WTH Bit [1:0] 內部固定為 0。 這個數值是物理上的像素值。	0	RW
1-0	**Fix at 0.**	0	RO

PAGE0 REG[A2h] Source 1 image width 1 (S1_WTH1)

Bit	Description	Default	Access
7-5	**N/A**	0	RO
4-0	**Source 1 image width [12:8]** 單位: 像素。 必須要能被 4 整除。 S1_WTH Bit [1:0] 內部固定為 0。 這個數值是物理上的像素值。	0	RW

PAGE0 REG[A3h] Source 1 Window Upper-Left corner X-coordinates 0 (S1_X0)

Bit	Description	Default	Access
7-0	**Source 1 Window Upper-Left corner X-coordinates [7:0]** 此暫存器是 Source 1 視窗左上角的 X 座標。	0	RW

PAGE0 REG[A4h] Source 1 Window Upper-Left corner X-coordinates 1 (S1_X1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Source 1 Window Upper-Left corner X-coordinates [12:8]** 此暫存器是 Source 1 視窗左上角的 X 座標。	0	RW

PAGE0 REG[A5h] Source 1 Window Upper-Left corner Y-coordinates 0 (S1_Y0)

Bit	Description	Default	Access
7-0	**Source 1 Window Upper-Left corner Y-coordinates [7:0]** 此暫存器是 Source 1 視窗左上角的 Y 座標。	0	RW

PAGE0 REG[A6h] Source 1 Window Upper-Left corner Y-coordinates 1 (S1_Y1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Source 1 Window Upper-Left corner Y-coordinates [12:8]** 此暫存器是 Source 1 視窗左上角的 Y 座標。	0	RW

PAGE0 REG[A7h] Destination memory start address 0 (DT_STR0)

Bit	Description	Default	Access
7-2	**Destination memory start address [7:2]**	0	RW
1-0	**Fix at 0**	0	RO

PAGE0 REG[A8h] Destination memory start address 1 (DT_STR1)

Bit	Description	Default	Access
7-0	**Destination memory start address [15:8]**	0	RW

PAGE0 REG[A9h] Destination memory start address 2 (DT_STR2)

Bit	Description	Default	Access
7-0	**Destination memory start address [23:16]**	0	RW

PAGE0 REG[AAh] Destination memory start address 3 (DT_STR3)

Bit	Description	Default	Access
7-0	**Destination memory start address [31:24]**	0	RW

註：目的記憶體起始位址不能在來源 0 來源 1 處理區塊內 ((image_width)*(image_height)*([1|2|3]color depth))，不然會有錯誤的結果輸出。

PAGE0 REG[ABh] Destination image width 0 (DT_WTH0)

Bit	Description	Default	Access
7-2	**Destination image width [7:2]** 單位: 像素。 必須要能被 4 整除. DT_WTH Bit [1:0] 內部固定為 0。 這個數值是物理上的像素值。	0	RW
1-0	**Fix at 0**	0	RO

PAGE0 REG[ACh] Destination image width 1 (DT_WTH1)

Bit	Description	Default	Access
7-5	**N/A**	0	RO
4-0	**Destination image width [12:8]** 單位: 像素。 必須要能被 4 整除。 DT_WTH Bit [1:0] 內部固定為 0。 這個數值是物理上的像素值。	0	RW

PAGE0 REG[ADh] Destination Window Upper-Left corner X-coordinates 0 (DT_X0)

Bit	Description	Default	Access
7-0	**Destination Window Upper-Left corner X-coordinates [7:0]**	0	RW

PAGE0 REG[AEh] Destination Window Upper-Left corner X-coordinates 1 (DT_X1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Destination Window Upper-Left corner X-coordinates [12:8]**	0	RW

PAGE0 REG[AFh] Destination Window Upper-Left corner Y-coordinates 0 (DT_Y0)

Bit	Description	Default	Access
7-0	**Destination Window Upper-Left corner Y-coordinates [7:0]**	0	RW

PAGE0 REG[B0h] Destination Window Upper-Left corner Y-coordinates 1 (DT_Y1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Destination Window Upper-Left corner Y-coordinates [12:8]**	0	RW

PAGE0 REG[B1h] BTE Window Width 0 (BTE_WTH0)

Bit	Description	Default	Access
7-0	**BTE Window Width Setting[7:0]** 單位: 像素。 這個數值是物理上的像素值。	0	RW

PAGE0 REG[B2h] BTE Window Width 1 (BTE_WTH1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**BTE Window Width Setting [12:8]** 單位: 像素。 這個數值是物理上的像素值。	0	RW

PAGE0 REG[B3h] BTE Window Height 0 (BTE_HIG0)

Bit	Description	Default	Access
7-0	**BTE Window Height Setting[7:0]** 單位: 像素。 這個數值是物理上的像素值。	0	RW

PAGE0 REG[B4h] BTE Window Height 1 (BTE_HIG1)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**BTE Window Height Setting [12:8]** 單位: 像素。 這個數值是物理上的像素值。	0	RW

PAGE0 REG[B5h] Alpha Blending (APB_CTRL)

Bit	Description	Default	Access
7-4	**N/A**	0	RO
5-0	**Window Alpha Blending effect for S0 & S1** 透明參數 alpha 值的範圍在 0.0~1.0 中，而 1.0 表示的是完全不透明，並且 0.0 表示的是全透明。 00h: 0 01h: 1/32 02h: 2/32 : 1Eh: 30/32 1Fh: 31/32 2Xh: 1 Output Effect = (S0 image x (1 - alpha setting value)) + (S1 image x alpha setting value)	0	RW

20.9 串列閃存與主 SPI 控制暫存器

PAGE0 REG[B6h] Serial flash DMA Controller REG (DMA_CTRL)

Bit	Description	Default	Access
7-6	**00b: [B7h] B3-0** **01b:** 4x read command code – 6Bh. Address output & data input interleaved on xmiso & xmosi & xsio2 & xsio3. **10b:** 4x read command code – EBh. Address output & data input interleaved on xmiso & xmosi & xsio2 & xsio3 **11b:NA**	00	RO
5-1	**NA**	0	RO
0	**Write Function: DMA Start Bit** 可經由 MPU 寫入為 1，並且馬上電路會自動清除為 0。 此位元無法與字元寫入同時使用，所以如果 DMA 被致能的話就無法設定設定為文字模式並且輸入字元碼。 **Read Function: DMA Busy Check Bit** 0: 閒置。 1: 忙碌。 *** 關於串列快閃記憶體的 DMA 傳輸方面，必須操作在圖形模式，並且須設定 SDRAM 中的 Canvas 目的起址位置、目的寬度、色深、定址模式。	0	RW

PAGE0 REG[B7h] Serial Flash/ROM Controller Register (SFL_CTRL)

Bit	Description	Default	Access
7	**Serial Flash/ROM I/F # Select** 0: 串列快閃記憶體/ROM 0 被選擇。 1: 串列快閃記憶體/ROM 1 被選擇。 *注*：當 page1 B7h bit 7 = 1 , serial flash chip 選擇 2,3。	0	RW
6	**Serial Flash /ROM Access Mode** 0: 字元模式– 使用在 CGROM。 1: DMA 模式– 使用在 CGRAM、pattern、boot start image 或 OSD 功能上。	0	RW
5	**Serial Flash/ROM Address Mode** 0: 24 bits 定址模式。 1: 32 bits 定址模式。 如果使用者希望使用 32 bits 定址模式，使用者必須自行輸入 EX4B 命令(B7h) 給串列快閃記憶體，並且設定此 bit 為 1。 使用者也可以檢查這個位元來知道是否在開機顯示中已經進入 32bit 位址模式。	0	RW

Bit	Description	Default	Access
4	**Must set to 1**	0	WO
3-0	**Read Command code & behavior selection** **000xb:** 1x 讀取命令 03h。讀取速度為 Normal read 速度。資料是由 xmiso 輸入。在位址與資料間不需要空週期。 **010xb:** 1x **讀取**命令 0Bh。為 faster read 速度。資料是由 xmiso 輸入，RA6807M 在位址與資料間會塞入 8 個空週期。 **1x0xb:** 1x 讀取命令 1Bh。為 fastest read 速度，資料是由 xmiso 輸入。RA6807M 在位址與資料間會塞入 16 個空週期 **xx10b:** 2x 讀取命令 3Bh。在 xmiso 與 xmosi 具有交錯資料輸入，在位址與資料間會塞入 8 個空週期 (Dual mode 0，請參考 Figure 16-7)。 ***註***：不是所有的 serial flash 都支援以上命令，請根據使用的 serial flash 來選擇正確的讀取命令。	0	R/W

PAGE0 REG[B8h] SPI master Tx /Rx FIFO Data Register (SPIDR)

Bit	Description	Default	Access
7-0	**SPI master Tx /Rx FIFO Data Register** 在程式化 core 控制暫存器後，SPI 可以進行傳送資料或命令。一個傳送要完成必須透過[SPIDR]暫存器。當 MPU 對 SPIDR 做寫入時，就必須透過 Write FIFO 來達成。每個寫入 Write FIFO 都會增加資料的位元組。使用上先將 core 致能 SS_ACTIVE，在 Write FIFO 在未滿的情形下寫入資料，就可做連續資料的寫入，此時最早寫入的資料將傳送出去。 在傳輸資料的同時也會接收資料，一個資料傳送就有一筆資料被接收。而讀取到的每筆資料都是由裝置提供的。而一個空週期必須被寫入 Write FIFO 中，這會導致開始做 SPI 傳輸，在傳輸的同時也會接收到資料。每當傳輸結束時，接收到的資料會存在 Read FIFO 中。Read FIFO 與 Write FIFO 是相對的，是具有 16 深度的 FIFO，Read FIFO 的內容可以經由 SPIDR 暫存器讀取。	NA	RW

PAGE0 REG[B9h] SPI master Control Register (SPIMCR2)

Bit	Description	Default	Access
7	B7 and B5 = 10b: nSS drive on xnsfcs[2] B7 and B5 = 11b: nSS drive on xnsfcs[3]	0	RW
6	**SPI Master Interrupt enable** 0: 禁能中斷。 1: 致能中斷。 *** 如果使用者禁能 SPIM 中斷旗標，那麼 RA8889 不會發出中斷給 MPU，所以使用者只能透過檢查 SPIMSR 暫存器的旗標來確認傳輸是否完成。	0	RW

<table>
<tr><th>Bit</th><th>Description</th><th>Default</th><th>Access</th></tr>
<tr><td>5</td><td>Control Slave Select drive on which xnsfcs
B7 and B5 = 00b: nSS 由 xnsfcs[0] 驅動。
B7 and B5 = 00b: nSS 由 xnsfcs[1] 驅動。</td><td>0</td><td>RW</td></tr>
<tr><td>4</td><td>Slave Select signal active [SS_ACTIVE]
0: 不動作 (nSS 將會輸出 high)。
1: 動作 (nSS 將會輸出 low)。
在 SS_ACTIVE 設為不動作時，FIFO 將會清除並且引擎將會維持在閒置狀態。
註: 建議在 SS_ACTIVE 動作時，不要更改 CPOL/CPHA 設定。</td><td>0</td><td>RW</td></tr>
<tr><td>3</td><td>Mask interrupt for FIFO overflow error [OVFIRQMSK]
0: 不遮罩。
1: 遮罩。</td><td>1</td><td>RW</td></tr>
<tr><td>2</td><td>Mask interrupt for while Tx FIFO empty & SPI engine/FSM idle [EMTIRQMSK]
0: 不遮罩。
1: 遮罩。</td><td>1</td><td>RW</td></tr>
<tr><td>1-0</td><td>SPI operation mode
當致能 DMA 或外部 CGROM 時，SPI 只支援 mode 0 與 mode 3。
<table>
<tr><th>mode</th><th>CPOL:
Clock Polarity bit</th><th>CPHA:
Clock Phase bit</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>2</td><td>1</td><td>0</td></tr>
<tr><td>3</td><td>1</td><td>1</td></tr>
</table></td><td>0</td><td>RW</td></tr>
</table>

- At CPOL=0，SCK 時脈在未動作時為 0。
  - For CPHA=0，資料是在時脈的上升緣讀取(low->high)，並且資料是在下降緣(high->low)變化。
  - For CPHA=1，資料是在時脈的下降緣讀取(high->low)，並且資料是在上升緣變化(low->high)。
- At CPOL=1 ，SCK 時脈在未動作時為 1(與 CPOL=0 反相)。
  - For CPHA=0, 資料是在時脈的下降緣讀取(high->low)，並且資料是在上升緣變化(low->high)。
  - For CPHA=1, 資料是在時脈的上升緣讀取(low->high)，並且資料是在下降緣(high->low)變化。

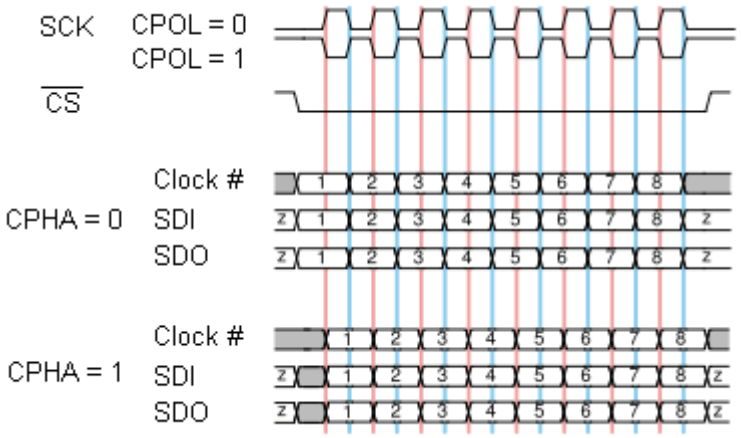


Figure 20-7

Table 20-2 : SPI MODES

SPI MODE	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

PAGE0 REG[BAh] SPI master Status Register (SPIMSR)

Bit	Description	Default	Access
7	**Tx FIFO empty flag** 0: 未空 (not empty)。 1: 已空 (empty)。	1	RO
6	**Tx FIFO full flag** 0: 未滿 (not full)。 1: 已滿 (full)。	0	RO
5	**Rx FIFO empty flag** 0: 未空 (not empty)。 1: 已空 (empty)。	1	RO
4	**Rx FIFO full flag** 0: 未滿 (not full)。 1: 已滿 (full)。	0	RO
3	1: Overflow interrupt flag 寫 1 將會清除此旗標。	0	RW
2	1: Tx FIFO empty & SPI engine/FSM idle interrupt flag 寫 1 將會清除此旗標。	0	RW
1-0	**NA**	0	RO

PAGE0 REG[BBh] SPI Clock period (SPI_DIVSOR)

Bit	Description	Default	Access
7-0	**SPI Clock Period** 參考系統時脈及 SPI 裝置需要的時脈以設定正確週期。 **SPI Master:** $F_{sck} = F_{core} / (divisor) \times 2$ **Serial Flash:** $F_{sck} = F_{core} / (divisor) \times 2$ When SPI_DIVSOR = 0, $F_{sck} = F_{core}$	3	RW

PAGE0 REG[BCh] Serial flash DMA Source Starting Address 0 (DMA_SSTR0)

Bit	Description	Default	Access
7-0	**Serial flash DMA Source START ADDRESS [7:0]** 此暫存器設定串列快閃記憶體的位址 address [7:0]。 直接指定來源圖檔的起始位址。	0	RW

PAGE0 REG[BDh] Serial flash DMA Source Starting Address 1 (DMA_SSTR1)

Bit	Description	Default	Access
7-0	**Serial flash DMA Source START ADDRESS [15:8]** 此暫存器設定串列快閃記憶體的位址 address [15:8]。 直接指定來源圖檔的起始位址。	0	RW

PAGE0 REG[BEh] Serial flash DMA Source Starting Address 2 (DMA_SSTR2)

Bit	Description	Default	Access
7-0	**Serial flash DMA Source START ADDRESS [23:16]** 此暫存器設定串列快閃記憶體的位址 address[23:16]。 直接指定來源圖檔的起始位址。	0	RW

PAGE0 REG[BFh] Serial flash DMA Source Starting Address 3 (DMA_SSTR3)

Bit	Description	Default	Access
7-0	**Serial flash DMA Source START ADDRESS [31:24]** 此暫存器設定串列快閃記憶體的位址 address[31:24]。 直接指定來源圖檔的起始位址。	0	RW

PAGE0 REG[C0h] DMA Destination Window Upper-Left corner X-coordinates 0 (DMA_DX0)

Bit	Description	Default	Access
7-0	**When REG 5Eh (AW_COLOR) bit 2 = 0 (Block Mode)** 此暫存器定義 DMA 的底圖 (Canvas) 上目的視窗左上角 X[7:0]。 **When REG 5Eh (AW_COLOR) bit 2 = 1 (Linear Mode)** 此暫存器定義 SDRAM 的目的記憶體位址[7:2]。	0	RW

PAGE0 REG[C1h] DMA Destination Window Upper-Left corner X-coordinates 1 (DMA_DX1)

Bit	Description	Default	Access
7-0	**When REG 5Eh (AW_COLOR) bit 2 = 0 (Block Mode)** 此暫存器定義 DMA 的底圖 (Canvas) 上目的視窗左上角 X[12:8]。 **When REG 5Eh (AW_COLOR) bit 2 = 1 (Linear Mode)** 此暫存器定義 SDRAM 的目的記憶體位址 [15:8]。	0	RW

PAGE0 REG[C2h] DMA Destination Window Upper-Left corner Y-coordinates 0 (DMA_DY0)

Bit	Description	Default	Access
7-0	**When REG 5Eh (AW_COLOR) bit 2 = 0 (Block Mode)** 此暫存器定義 DMA 的底圖 (Canvas) 上目的視窗左上角 Y[7:0]。 **When REG 5Eh (AW_COLOR) bit 2 = 1 (Linear Mode)** 此暫存器定義 SDRAM 的目的記憶體位址[23:16]。	0	RW

PAGE0 REG[C3h] DMA Destination Window Upper-Left corner Y-coordinates 1 (DMA_DY1)

Bit	Description	Default	Access
7-0	**When REG 5Eh (AW_COLOR) bit 2 = 0 (Block Mode)** 此暫存器定義 DMA 的底圖 (Canvas) 上目的視窗左上角 Y[12:8]。 **When REG 5Eh (AW_COLOR) bit 2 = 1 (Linear Mode)** 此暫存器定義 SDRAM 的目的記憶體位址[31:24]。	0	RW

PAGE0 REG[C4h] : RESERVED

Bit	Description	Default	Access
7-0	**N/A**	0	RO

PAGE0 REG[C5h] SPI Master Bus Select(SPIMBS)

Bit	Description	Default	Access
7	**SPI master bus select** 0: Bus 0(xsck, xmosi, xmiso) 1: Bus 1(xspi1_sck, xspi1_msio0, xspi1_msio1)	0	RW
6	**N/A**	0	RO
5	**SPI master rx register latch edge** 0: cclk rising edge 1: cclk falling edge	0	RW
4-0	**N/A**	0	RO

PAGE0 REG[C6h] DMA Block Width 0 (DMAW_WTH0)

Bit	Description	Default	Access
7-0	**When REG 5Eh (AW_COLOR) bit 2 = 0 (Block Mode)** DMA 區塊寬度[7:0]。 **When REG 5Eh (AW_COLOR) bit 2 = 1 (Linear Mode)** DMA 傳輸數目[7:0]。	0	RW

PAGE0 REG[C7h] DMA Block Width 1 (DMAW_WTH1)

Bit	Description	Default	Access
7-0	**When REG 5Eh (AW_COLOR) bit 2 = 0 (Block Mode)** DMA 區塊寬度[15:8]。 **When REG 5Eh (AW_COLOR) bit 2 = 1 (Linear Mode)** DMA 傳輸數目[15:8]。	0	RW

PAGE0 REG[C8h] DMA Block Height 0 (DMAW_HIGH0)

Bit	Description	Default	Access
7-0	**When REG 5Eh (AW_COLOR) bit 2 = 0 (Block Mode)** DMA 區塊高度[7:0]。 **When REG 5Eh (AW_COLOR) bit 2 = 1 (Linear Mode)** DMA 傳輸數目[23:16]。	0	RW

PAGE0 REG[C9h] DMA Block Height 1 (DMAW_HIGH1)

Bit	Description	Default	Access
7-0	**When REG 5Eh (AW_COLOR) bit 2 = 0 (Block Mode)** DMA 區塊高度[15:8]。 **When REG 5Eh (AW_COLOR) bit 2 = 1 (Linear Mode)** DMA 傳輸數目[31:24]。	0	RW

PAGE0 REG[CAh] DMA Source Picture Width 0 (DMA_SWTH0)

Bit	Description	Default	Access
7-0	**DMA Source Picture Width [7:0]** 單位: 像素。	0	RW

PAGE0 REG[CBh] DMA Source Picture Width 0 (DMA_SWTH1)

Bit	Description	Default	Access
7-5	**N/A**	0	RO
4-0	**DMA Source Picture Width [12:8]**	0	RW

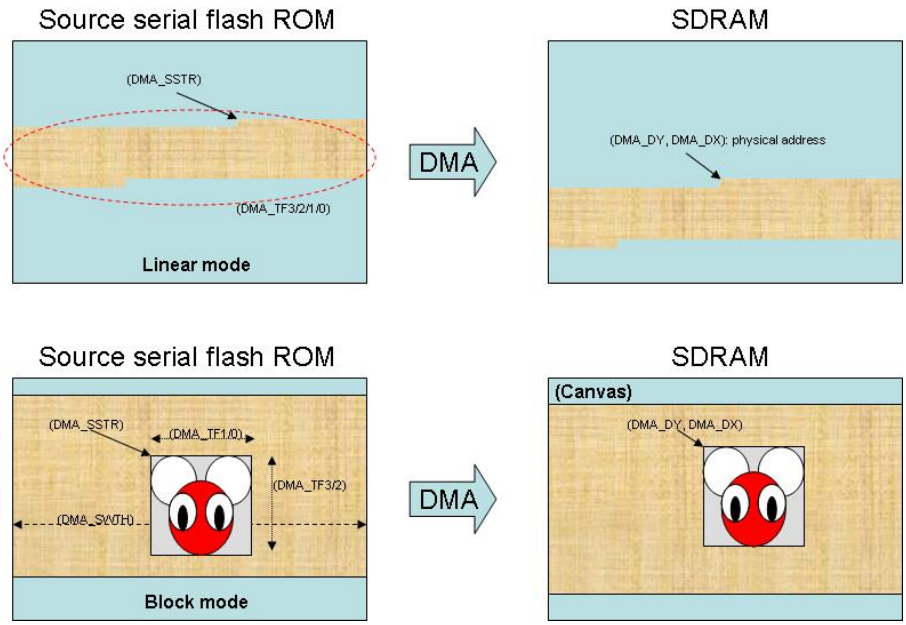


Figure 20-8 : DMA Linear and Block Mode

20.10 文字引擎

PAGE0 REG[CCh] Character Control Register 0 (CCR0)

Bit	Description	Default	Access
7-6	**Character source selection** 00: 內部 CGROM 為字元來源。 01: 外部 CGROM 為字元來源 (集通快閃記憶體)。 10: 使用者定義字元。 11: NA。	0	RW
5-4	**Character Height Setting** for external CGROM & user-defined Character 00b : 16; ex. 8x16 / 16x16 /不等寬 x 16。 01b : 24; ex. 12x24 / 24x24 /不等寬 x 24。 10b : 32; ex. 16x32 / 32x32 /不等寬 x 32。 *註* : 1. 使用者自定義字元的寬度另須參考字元碼，當字碼< 8000h 時為半形字，寬度為 8/12/16。當字碼>=8000h 為全形字，寬度為 16/24/32。 2. 集通快閃記憶體字元寬度必須參考字元記憶體規格書，並且設定 GT Font ROM (CEh, CFh) 相關暫存器。 3. 內部 CGROM 支援 12x24。	01b	RW
3-2	**NA**	0	RO
1-0	**Character Selection for internal CGROM** 當 FNCR0 B7 = 0 與 B6 = 0，將是選擇內部 CGROM 的字元組，並且內部 CGROM 包含了 ISO/IEC 8859-1,2,4,5，可以支援英文及大部份歐洲國家的語言。 00b : ISO/IEC 8859-1。 01b : ISO/IEC 8859-2。 10b : ISO/IEC 8859-4。 11b : ISO/IEC 8859-5。	0	RW

PAGE0 REG[CDh] Character Control Register 1 (CCR1)

Bit	Description	Default	Access
7	**Full Alignment Selection Bit** 0： 全對齊禁能。 1： 全對齊致能。 當全對齊致能時，顯示字元的寬度會是字元高度的 1/2。此條件為如果字元寬度是小於或等於字元高度 1/2 那麼就會顯示其寬度為 1/2 高度，否則就會顯示高度相同的寬度。	0	RW
6	**Chroma keying enable on Text input** 0： 字元的資料 0 會顯示為指定的顏色。 1： 字元的資料 0 會顯示為底圖 (Canvas)	0	RW

Bit	Description	Default	Access
5	**NA**	0	RO
4	**Character Rotation** 0 : Normal 文字方向從左到右然後從上到下。 1： 逆時針 90 度，並且垂直翻轉。 文字方向從上到下然後從左到右。 (這應該設定 VDIR 為 1)。 之前寫入文字必須被處理完，才可更改屬性，使用者可以去檢查狀態暫存器的 core_busy 來確定是否可以進行更改。	0	RW
3-2	**Character width enlargement factor** 00b: X1. 01b: X2. 10b: X3. 11b: X4.	0	RW
1-0	**Character height enlargement factor** 00b: X1. 01b: X2. 10b: X3. 11b: X4.	0	RW

PAGE0 REG[CEh] GT Character ROM Select (GTFNT_SEL)

Bit	Description	Default	Access
7-5	**GT Serial Character ROM Select** 000b: GT21L16T1W 001b: GT30L16U2W 010b: GT30L24T3Y 011b: GT30L24M1Z 100b: GT30L32S4W 101b: GT20L24F6Y 110b: GT21L24S1W	0	RW
4-0	**N/A**	0	RO

PAGE0 REG[CFh] GT Character ROM Control register (GTFNT_CR)

Bit	Description	Default	Access
7-3	**Character sets** 對於指定的集通 CGROM，編碼方式與解碼方式必須是對應的。 **a. Single byte character code for following character sets:** 00100b: ASCII only (00h-1Fh, 80-FFh will send “blank space”) 10001b: ISO-8859-1 + ASCII code 10010b: ISO-8859-2 + ASCII code 10011b: ISO-8859-3 + ASCII code 10100b: ISO-8859-4 + ASCII code 10101b: ISO-8859-5 + ASCII code	0	RW

Bit	Description	Default	Access
	10110b: ISO-8859-7 + ASCII code 10111b: ISO-8859-8 + ASCII code 11000b: ISO-8859-9 + ASCII code 11001b: ISO-8859-10 + ASCII code 11010b: ISO-8859-11 + ASCII code 11011b: ISO-8859-13 + ASCII code 11100b: ISO-8859-14 + ASCII code 11101b: ISO-8859-15 + ASCII code 11110b: ISO-8859-16 + ASCII code **b. Two byte character code for following character sets:** 00000b: GB2312 00001b: GB12345/GB18030 00010b: BIG5 00011b: UNICODE 00101b: UNI-Japanese 00110b: JIS0208 00111b: Latin / Greek / Cyrillic / Arabic / Thai / Hebrew *註*: 此 bits 設定不是 00011b, 00101b, 00110b, 00111b (UNICODE, UNI-Japanese, JIS0208, Latin / Greek / Cyrillic / Arabic / Thai / Hebrew) 那麼第一個字碼如果在 80h 以下，將會被視為 ASCII 來處理。		
2	**N/A**	0	RO
1-0	**GT Character width setting** **00b**: 對於固定寬度的字元組，字元的寬度是高度的一半。Ex. ISO-8859, GB2312, GB12345/GB18030, BIG5, UNI-Japanese, JIS0208, Thai. **Others**: 以下字元組具有不等寬字元: ASCII, Latin, Greek, Cyrillic & Arabic.	0	RW

Relationship of Character sets & GT Character width as following:

Char. set / Width	**ASCII Code/ ISO-8859-x (00100b /1xxxxb)**	**Latin / Greek / Cyrillic (00111b)**	**Arabic (00111b)**	**Others**
00b	固定寬度	固定寬度	NA	固定寬度 (auto set by chip)
01b	Arial 不等寬	不等寬	格式 A 不等寬	NA
10b	Roman 不等寬	NA	格式 B 不等寬	NA
11b	Bold	NA	NA	NA

PAGE0 REG[D0h] Character Line gap Setting Register (FLDR)

Bit	Description	Default	Access
7-5	**NA**	0	RO
4-0	**Character Line gap Setting** 設定字元的行距，當輸入字元達到是窗邊緣時會跳下一行。 (單位: 像素) 行距的顏色以背景色暫存器設定為主。 ***此行距不會與字符放大功能連動。	0	RW

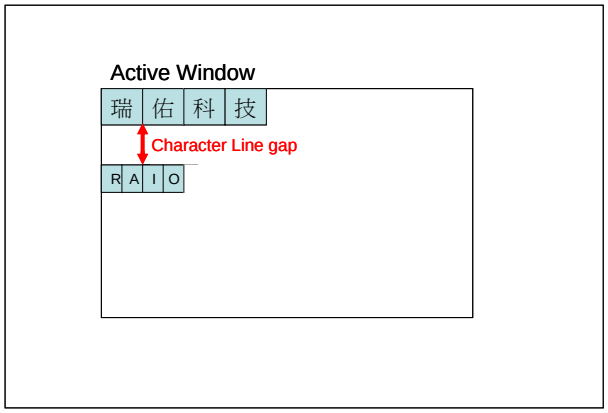


Figure 20-9 : Character Line Gap

PAGE0 REG[D1h] Character to Character Space Setting Register (F2FSSR)

Bit	Description	Default	Access
7-6	**NA**	0	RW
5-0	**Character to Character Space Setting** 00h : 0 pixel 01h : 1 pixel 02h : 2 pixels : 3Fh : 63 pixels 字元間距會填前景色。 ***此功能不會與字元放大連動。	0	RW

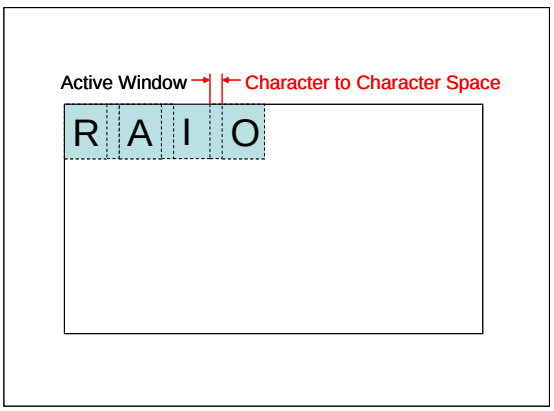


Figure 20-10 : Character to Character Space

PAGE0 REG[D2h] Foreground Color Register - Red (FGCR)

Bit	Description	Default	Access
7-0	**Foreground Color - Red; for draw, text or color expansion** 256 色，為此暫存器的 Bit[7:5]。 65K 色，為此暫存器的 Bit[7:3]。 16.7M 色，為此暫存器的 Bit[7:0]。	FFh	RW

PAGE0 REG[D3h] Foreground Color Register - Green (FGCG)

Bit	Description	Default	Access
7-0	**Foreground Color - Green; for draw, text or color expansion** 256 色，為此暫存器的 Bit[7:5]。 65K 色，為此暫存器的 Bit[7:2]。 16.7M 色，為此暫存器的 Bit[7:0]。	FFh	RW

PAGE0 REG[D4h] Foreground Color Register - Blue (FGCB)

Bit	Description	Default	Access
7-0	**Foreground Color - Blue; for draw, text or color expansion** 256 色，為此暫存器的 Bit[7:6]。 65K 色，為此暫存器的 Bit[7:3]。 262K 色，為此暫存器的 Bit[7:0]。	FFh	RW

PAGE0 REG[D5h] Background Color Register - Red (BGCR)

Bit	Description	Default	Access
7-0	**Background Color - Red; for Text or color expansion** 256 色，為此暫存器的 Bit[7:5]。 65K 色，為此暫存器的 Bit[7:3]。 16.7M 色，為此暫存器的 Bit[7:0]。 ***註：無論背景色透明是否被啟用，不要設定與前景色相同的值，否則圖像或文字將會是以前景色方形的方式顯示，在 BTE 功能中亦同，不可設相同值。	00h	RW

PAGE0 REG[D6h] Background Color Register - Green (BGCG)

Bit	Description	Default	Access
7-0	**Background Color - Green; for Text or color expansion** 256 色，為此暫存器的 Bit[7:5]。 65K 色，為此暫存器的 Bit[7:2]。 16.7M 色，為此暫存器的 Bit[7:0]。 ***註：無論背景色透明是否被啟用，不要設定與前景色相同的值，否則圖像或文字將會是以前景色方形的方式顯示，在 BTE 功能中亦同，不可設相同值。	00h	RW

PAGE0 REG[D7h] Background Color Register - Blue (BGCB)

Bit	Description	Default	Access
7-0	**Background Color - Blue; for Text or color expansion** 256 色，為此暫存器的 Bit[7:6]。 65K 色，為此暫存器的 Bit[7:3]。 16.7M 色，為此暫存器的 Bit[7:0]。 ***註：無論背景色透明是否被啟用，不要設定與前景色相同的值，否則圖像或文字將會是以前景色方形的方式顯示，在 BTE 功能中亦同，不可設相同值。	00h	RW

PAGE0 REG[D8h] – REG[DAh] : RESERVED

Bit	Description	Default	Access
7-0	**NA**	0	RO

PAGE0 REG[DBh] CGRAM Start Address 0 (CGRAM_STR0)

Bit	Description	Default	Access
7-0	**CGRAM START ADDRESS [7:0]** 使用者定義字元空間的位址。 使用者必須使用底圖 (Canvas) 的設定來輸入 CGRAM 的資料，並且使用 CGRAM 的位址暫存器來抓取 CGRAM 的資料。	0	RW

PAGE0 REG[DCh] CGRAM Start Address 1 (CGRAM_STR1)

Bit	Description	Default	Access
7-0	**CGRAM START ADDRESS [15:8]** 使用者定義字元空間的位址。 使用者必須使用底圖 (Canvas) 的設定來輸入 CGRAM 的資料，並且使用 CGRAM 的位址暫存器來抓取 CGRAM 的資料。	0	RW

PAGE0 REG[DDh] CGRAM Start Address 2 (CGRAM_STR2)

Bit	Description	Default	Access
7-0	**CGRAM START ADDRESS [23:16]** 使用者定義字元空間的位址。 使用者必須使用底圖 (Canvas) 的設定來輸入 CGRAM 的資料，並且使用 CGRAM 的位址暫存器來抓取 CGRAM 的資料。	0	RW

PAGE0 REG[DEh] CGRAM Start Address 3 (CGRAM_STR3)

Bit	Description	Default	Access
7-0	**CGRAM START ADDRESS [31:24]** 使用者定義字元空間的位址。 使用者必須使用底圖 (Canvas) 的設定來輸入 CGRAM 的資料，並且使用 CGRAM 的位址暫存器來抓取 CGRAM 的資料。	0	RW

*** ***註:*** 如果使用者需要更改屬性的話，如旋轉、行距、間距、前景色、背景色、文字圖形模式設定，使用者必須確定 core_busy (fontwr_busy) 狀態是在 low。

20.11 能源管理控制暫存器

PAGE0 REG[DFh] : Power Management register (PMU)

Bit	Description	Default	Access
7	**Enter Power saving state** 0: 標準模式或從省電模式中喚醒。 1: 進入省電模式。 ***註 :*** 有三種方法可以從省電模式中喚醒: 外部中斷喚醒、鍵盤掃描喚醒、軟體喚醒。 對這個 bit 寫 0 可以產生軟體喚醒，在系統喚醒後此 bit 才會被清為 0，在系統未完全甦醒時，讀取此 bit 仍為 1。MPU 必須等待系統跳出省電模式才能允許寫暫存器。使用者可以檢查此位元或是檢查狀態暫存器位元 bit [1] (power saving) 來得知系統是否已經回到標準操作模式了。	0	RW
6-2	**NA**	0	RO
1-0	**Power saving Mode definition** 00: NA 01: 待機模式 CCLK & PCLK 會停止，MCLK 將維持由 MPLL 提供。 10: 休眠模式 CCLK & PCLK 會停止，MCLK 則由 OSC 時脈提供。 11: 睡眠模式 所有時脈與 PLL 都會停止。	3	RW

20.12 SDRAM 控制暫存器

PAGE0 REG[E0h] SDRAM attribute register (SDRAR)

Bit	Description	Default	Access
7	**SDRAM Power Saving type** 0: 執行 power down 命令以進入省電模式。 1: 執行 self refresh 命令以進入省電模式。	0	RW
6	**M**ust set be 0	0	RW
5	**SDRAM Bank number (sdr_bank)** 0b: 2 banks (column 位址容量只支援 256 words) 1b: 4 banks	1	RW
4-3	**SDRAM Row addressing (sdr_row)** 00b: 2K (A0-A10) 01b: 4K (A0-A11) 1Xb: 8K (A0-A12)	1	RW
2-0	**SDRAM Column addressing (sdr_col)** 000b: 256 (A0-A7) 001b: 512 (A0-A8) 010b: 1024 (A0-A9) 011b: 2048 (A0-A9, A11) 1XXb: 4096 (A0-A9, A11-A12)	0	RW

Reference setting:
128Mb, 16MB, 8Mx16: 0x29; bank no: 4, row size: 4096, col size: 512

PAGE0 REG[E1h] SDRAM mode register & extended mode register (SDRMD)

Bit	Description	Default	Access
7-5	**Partial-Array Self Refresh (sdr_pasr)** *Only for mobile SDR SDRAM 000b: Full array 001b: Half array (1/2) 010b: Quarter array (1/4) 011b: 保留 100b: 保留 101b: One-eighth array (1/8) 110b: One-sixteenth array (1/16) 111b: 保留	0	RW
4-3	**To select the driver strength of the DQ outputs (sdr_drv)** *Only for mobile SDR SDRAM 00b: Full-strength driver 01b: Half-strength driver 10b: Quarter-strength driver 11b: One eighth-strength driver	0	RW

Bit	Description	Default	Access
2-0	**SDRAM CAS latency (sdr-caslat)** 010b: 2 SDRAM clock 011b: 3 SDRAM clock Other: 保留	03h	RW

***NOTE: This register was locked after sdr_initdone bit was set as 1.**

PAGE0 REG[E2h] SDRAM auto refresh interval (SDR_REF_ITVL0)

Bit	Description	Default	Access
7-0	**Refresh interval (Low byte)** SDRAM 內部自動刷新時間，由 SDRAM 時脈計數。 ***** 如果此暫存器設定為 0000h，SDRAM 自動刷新將會被禁能。** 內部刷新時間是根據 SDRAM's Refresh 的週期規格與 row size 來決定。 Ex. 如果 SDRAM 頻率是 140MHz，SDRAM 的刷新周期 Tref 是 64ms，並且 row size 為 4096，那麼內部刷新時間應該是小於 64e-3 / 4096 * 140e6 ~= 2187, 2187-2 = 2185 = 889 h，因此暫存器[E2h][E3h]就是設定 889h。	00h	RW

PAGE0 REG[E3h] SDRAM auto refresh interval (SDR_REF_ITVL1)

Bit	Description	Default	Access
7-0	**Refresh interval (High byte)** SDRAM 內部自動刷新時間，由 SDRAM 時脈計數。 ***** 如果此暫存器設定為 0000h，SDRAM 自動刷新將會被禁能。**	00h	RW

PAGE0 REG[E4h] SDRAM Control register (SDRCR)

Bit	Description	Default	Access
7-6	**Length to break a burst transfer** 00: 256 01: 128 10: 64 11: 32	0	RW
5	必須要被設成 0	0	RW
4	**XMCKE pin state** 為目前 XMCKE 腳位的狀態。 0: SDR 記憶體時脈禁能。 1: SDR 記憶體時脈致能。	1	RO
3	**Report warning condition** 0: 禁能或清除警告旗標。 1: 致能警告旗標。 警告條件是當讀取記憶體位址接近 SDRAM 最大位址 (可能是超過最大位址減去 512bytes) 或是超過可存取的範圍或是讀取 SDRAM 頻寬跟不上幀更新的速率，那麼警告事件將會被鎖定，	0	RW

Bit	Description	Default	Access
	使用者可以檢查這個位元來確定。這個警告旗標可以透過設定這個 bit 為 0 來清除。		
2	**SDRAM timing parameter register enable(SDR_PARAMEN)** 0: 禁能 SDRAM 時序參數暫存器。 1: 致能 SDRAM 時序參數暫存器。	0	RW
1	**SDRAM enter power saving mode (sdr_psaving)** 0 到 1 的變化將會進入省電模式。 1 到 0 的變化將會跳出省電模式。	0	RW
0	**Start SDRAM initialization procedure (sdr_initdone)** 0 到 1 變化將會執行 SDRAM 初始程序。 讀取此位元'1' 表示 SDRAM 已經被初始化並且可以被存取了。 一旦被寫 1 後，就無法被重寫為 0。 1 到 0 的變化不需要其他的操作。	0	RW

***** 下列 SDRAM 時序暫存器只有當 SDR_PARAMEN (REG[E4], b2) 設為 1 時有效。**

PAGE0 REG[E0h] SDRAM timing parameter 1

Bit	Description	Default	Access
7	**NA**	0	RO
6	**NA**	0	RW
5	**NA**	0	RW
4	**NA**	0	RW
3-0	tMRD : Load Mode 命令到 Active 或 Refresh 命令的時間。 00h – 0Fh: 1 ~ 16 SDRAM 時脈。	2	RW

PAGE0 REG[E1h] SDRAM timing parameter 2

Bit	Description	Default	Access
7-4	tRFC : 自動刷新週期。 00h – 0Fh: 1 ~ 16 SDRAM clock。	8	RW
3-0	tXSR : 跳出 SELF REFRESH-to-ACTIVE command。 00h – 0Fh: 1 ~ 16 SDRAM 時脈。	7	RW

PAGE0 REG[E2h] SDRAM timing parameter 3

Bit	Description	Default	Access
7-4	tRP : PRECHARGE 命令的週期時間(15/20ns)。 00h – 0Fh: 1 ~ 16 SDRAM 時脈。	2	RW
3-0	tWR : Time of WRITE recovery time。 00h – 0Fh: 1 ~ 16 SDRAM 時脈。	0	RW

PAGE0 REG[E3h] SDRAM timing parameter 4

Bit	Description	Default	Access
7-4	tRCD : ACTIVE-to-READ 或 WRITE 的延遲時間。 00h – 0Fh: 1 ~ 16 SDRAM 時脈。	2	RW
3-0	tRAS : Time of ACTIVE-to-PRECHARGE。 00h – 0Fh: 1 ~ 16 SDRAM 時脈。	6	RW

20.13 IIC Master 暫存器

PAGE0 REG[E5h] IIC Master Clock Pre-scale Register 0 (IICMCPR0)

Bit	Description	Default	Access
7-0	**IIC Master Clock Pre-scale [7:0]** *XSCL = CCLK / (5*(Pre-scale + 2))*	0	RW

PAGE0 REG[E6h] IIC Master Clock Pre-scale Register 1 (IICMCPR1)

Bit	Description	Default	Access
7-0	**IIC Master Clock Pre-scale [15:8]** *XSCL = CCLK / (5*(Pre-scale + 2))*	0	RW

PAGE0 REG[E7h] IIC Master Transmit Register (IICMTXR)

Bit	Description	Default	Access
7-0	**IIC Master Transmit [7:0]**	0	RW

PAGE0 REG[E8h] IIC Master Receiver Register (IICMRXR)

Bit	Description	Default	Access
7-0	**IIC Master Receiver [7:0]**	0	RW

PAGE0 REG[E9h] IIC Master Command Register (IICMCMDR)

Bit	Description	Default	Access
7	**START** 產生(重覆)開始條件，並且會被硬體自動清除。 *註*：讀取這個 bit 永遠為 0。	0	RW
6	**STOP** 產生停止條件，並且此位元會被硬體自動清除。 *註*: 讀取這個 bit 永遠為 0。	0	RW
5	**READ(READ and WRITE can't be used simultaneously)** 從 slave 讀資料，並且此位元會被硬體自動清除。 *註*: 讀取這個 bit 永遠為 0。	0	RW
4	**WRITE(READ and WRITE can't be used simultaneously)** 對 Slave 做寫入，並且此位元會被硬體自動清除。 *註*: 讀取這個 bit 永遠為 0。	0	RW

Bit	Description	Default	Access
3	**ACKNOWLEDGE** 當 IIC master 接收到資料時。 0 : Sent ACK。 1 : Sent NACK。 *註*： 讀取這個 bit 永遠為 0。	0	RW
2-1	**NA**	0	RO
0	**Noise Filter** 0 :.禁能。 1： 致能。	0	RW

PAGE0 REG[EAh] IIC Master Status Register (IICMSTUR)

Bit	Description	Default	Access
7	**Received acknowledge from slave** 0 : Acknowledge 接收到。 1： 沒有 Acknowledge 接受到。	0	RO
6	**IIC Bus is Busy** 0： 閒置狀態，在 STOP 訊號被偵測到時，此 bit 為 0。 1： 忙碌狀態，在 START 訊號被偵測到時，此 bit 為 1。	0	RO
5-2	**NA**	0	RO
1	**Transfer in progress** 0： 當傳輸完成時。 1： 當傳輸正在進行。	0	RO
0	**Arbitration lost** 當 RA8889 失去 arbitration 時，這個 bit 會設成 1。Arbitration 會失去的狀況有: 一個 STOP 訊號被偵測到，但是並沒有被要求，此時 RA8889 的 master 會驅動 SDA 為 high，但是其他的 master 會將 SDA 驅動 low。	0	RO

20.14 GPI 與 GPO 暫存器

PAGE0 REG[F0h] GPIO-A direction (GPIOAD)

Bit	Description	Default	Access
7-0	**General Purpose I/O, Port A** **GPIO-A_dir[7:0] : General Purpose I/O direction control** 0: 輸出。 1: 輸入。	FFh	RW

PAGE0 REG[F1h] GPIO-A (GPIOA)

Bit	Description	Default	Access
7-0	**General Purpose I/O, Port A** **Only available in parallel 8-bits MPU I/F & serial MPU I/F** **For Write, Port A's General Purpose Output** GPO-A[7:0] : A 埠為通用型輸出，與 DB[15:8]共用腳位。 **For Read, Port A's General Purpose Input** GPI-A[7:0] : A 埠為通用型輸入，與 DB[15:8]共用腳位。	NA	RW

PAGE0 REG[F2h] GPIO-B (GPIOB)

Bit	Description	Default	Access
7-5	**NA**	NA	NA
4	**For Write. Port B's General Purpose Output** 輸出資料與 KOUT[0] 共用腳位。 **For Read, Port B's General Purpose Input** 輸入資料與 KIN[0] 共用腳位。	NA	RW
3-0	**For Read, Port B's General Purpose Input** This bit not writable. Only valid on serial host interface. {XA0, XnWR, XnRD, XnCS} **For Read, Port B's General Purpose Input** 這個 bit 是唯讀的，只有使用在串列主控端介面。 {XA0, XnWR, XnRD, XnCS}	NA	R

PAGE0 REG[F3h] GPIO-C direction (GPIOCD)

Bit	Description	Default	Access
7-0	**General Purpose I/O, Port C** **GPIO-C_dir[7:0] : General Purpose I/O direction control** 0: 輸出。 1: 輸入。	FFh	RW

PAGE0 REG[F4h] GPIO-C (GPIOC)

Bit	Description	Default	Access
7-0	**General Purpose I/O, Port C** **GPIO-C[7] & GPIO_C[4:0] : General Purpose Input / Output** 與 {XPWM0, XnSFCS1, XnSFCS0, XMISO, XMOSI, XSCK} 共用腳位。 GPIO 功能只有在相關的功能被禁能時才能使用。 (ex. PWM, SPI master disabled). *** GPIO_C[6:5] 是無法使用的。	NA	RW

PAGE0 REG[F5h] GPIO-D direction (GPIODD)

Bit	Description	Default	Access
7-0	**General Purpose I/O, Port D** **GPIO-D_dir[7:0] : General Purpose I/O direction control** 0: 輸出。 1: 輸入。	FFh	RW

PAGE0 REG[F6h] GPIO-D (GPIOD)

Bit	Description	Default	Access
7-0	**General Purpose I/O, Port D** **GPIO-D[7:0] : General Purpose Input/Output** 與 PDAT[18, 2, 17, 16, 9, 8, 1, 0] 共用腳位。	NA	RW

PAGE0 REG[F7h] GPIO-E direction (GPIOED)

Bit	Description	Default	Access
7-0	**General Purpose I/O, Port E** **GPIO-E_dir[7:0] : General Purpose I/O direction control** 0: 輸出。 1: 輸入。	FFh	RW

PAGE0 REG[F8h] GPIO-E (GPIOE)

Bit	Description	Default	Access
7-0	**General Purpose I/O, Port E** **GPIO-E[7:0] : General Purpose Input/Output.** 與 XPDAT[12, 11, 10, 7, 6, 5, 4, 3] 共用腳位。	NA	RW

PAGE0 REG[F9h] GPIO-F direction (GPIOFD)

Bit	Description	Default	Access
7-0	**GPIO-F_dir[7:0] : General Purpose I/O direction control** 0: 輸出。 1: 輸入。	FFh	RW

PAGE0 REG[FAh] GPIO-F (GPIOF)

Bit	Description	Default	Access
7-0	**General Purpose I/O, Port F** **GPIO-F[7:0] : General Purpose Input/Output.** 與 XPDAT[23, 22, 21, 20, 19, 15, 14, 13] 共用腳位。	NA	RW

20.15 鍵盤控制暫存器

PAGE0 REG[FBh] Key-Scan Control Register 1 (KSCR1)

Bit	Description	Default	Access
7	保留。 必須被設為 0。	0	0
6	**Long Key Enable Bit** 1： 致能，長按鍵週期被 KSCR2 bit4-2 設定。 0： 禁能。	0	RW
5-4	**Short Key de-bounce Times** 消除鍵盤彈跳時間，以 key-scan 掃描週期為基頻。 00b : 4 01b : 8 10b : 16 11b : 32	0	RW
3	**Repeatable Key enable** 0: 禁能重覆鍵。 1: 致能重覆鍵。 ie, 如果鍵盤始終被按下，並且長按鍵被禁能的情況下，那麼控制器將會重覆以短按鍵的消除彈跳時間發出按鍵中斷，但是使用者必須要去清除中斷旗標，否則會看不到下一個中斷，因為中斷旗標狀態在上一次的中斷已經被記錄到 1；而如果長按鍵被致能那麼發出中斷的時間是以長按鍵的認可時間，同樣的每次中斷產生後，使用者如果要看到下一次的中斷，則必須先清除中斷旗標。	0	RW
2-0	**Row Scan Time** **Period of Key scan controller to scan one row.** $T_{KEYCLK} = \frac{1}{F_{SYSCLK}} \times 2048$ 000: $ROW\_SCAN\_Time = T_{KEYCLK}$ 001: $ROW\_SCAN\_Time = T_{KEYCLK} \times 2$ 010: $ROW\_SCAN\_Time = T_{KEYCLK} \times 4$ 011: $ROW\_SCAN\_Time = T_{KEYCLK} \times 8$ 100: $ROW\_SCAN\_Time = T_{KEYCLK} \times 16$ 101: $ROW\_SCAN\_Time = T_{KEYCLK} \times 32$ 110: $ROW\_SCAN\_Time = T_{KEYCLK} \times 64$ 111: $ROW\_SCAN\_Time = T_{KEYCLK} \times 128$ **This key pad controller supports 5x5 keys. Total Key pad scan time = Row Scan Time * 5**	0	RW

PAGE0 REG[FCh] Key-Scan Controller Register 2 (KSCR2)

Bit	Description	Default	Access
7	**Key-Scan Wakeup Function Enable Bit** 0: Key-Scan 喚醒功能被禁能。 1: Key-Scan 喚醒功能被致能。	0	R/W
6	**Key released interrupt enable** 0: 當所有按鍵被釋放時，沒有中斷產生。 1: 當所有按鍵被釋放時，有中斷產生。	0	RW
5	**NA**	0	RO
4-2	**Long Key Recognition Factor** 這是指定長按鍵認可時間，短按鍵會先被認可後長按鍵才會被認可，數值 0 到 7。 $LongKey\,\mathrm{Re}\,cognitionTime = RowScanTime \times 5 \times (LongKey\,\mathrm{Re}\,cognitionFactor + 1) \times 1024$	0	RW
1-0	**Numbers of Key Hit.** 0：沒有按鍵被按下。 1：一鍵被按下，REG[FDh]是鍵碼。 2：兩個按鍵被按下，REG[FEh]紀錄第二個鍵碼。 3：三個按鍵被按下，REG[FFh]紀錄第三個鍵碼。 如果在超過一個消除彈跳時間內沒有任何按鍵被按下，則這個位元會回到 0。	0	RO

PAGE0 REG[FDh] Key-Scan Data Register (KSDR0)

Bit	Description	Default	Access
7-0	**Key Strobe Data0** 對應的鍵碼 0 被按下。 在超過一個彈跳時間內沒有任何按鍵被按下的化，則此暫存器會回到 FFh。	TBD	RO

PAGE0 REG[FEh] Key-Scan Data Register (KSDR1)

Bit	Description	Default	Access
7-0	**Key Strobe Data1** 對應的鍵碼 1 被按下。 在超過一個彈跳時間內沒有任何按鍵被按下的化，則此暫存器會回到 FFh。	TBD	RO

PAGE0 REG[FFh] Key-Scan Data Register (KSDR2)

Bit	Description	Default	Access
7-0	**Key Strobe Data2** 對應的鍵碼 2 被按下。 在超過一個彈跳時間內沒有任何按鍵被按下的化，則此暫存器會回到 FFh。	TBD	RO

Table 20-3 : Key Code Mapping Table (Normal Key)

	Kin0	Kin1	Kin2	Kin3	Kin4
Kout0	00h	01h	02h	03h	04h
Kout1	10h	11h	12h	13h	14h
Kout2	20h	21h	22h	23h	24h
Kout3	30h	31h	32h	33h	34h
Kout4	40h	41h	42h	43h	44h

Table 20-4 : Key Code Mapping Table (Long Key)

	Kin0	Kin1	Kin2	Kin3	Kin4
Kout0	80h	81h	82h	83h	84h
Kout1	90h	91h	92h	93h	94h
Kout2	A0h	A1h	A2h	A3h	A4h
Kout3	B0h	B1h	B2h	B3h	B4h
Kout4	C0h	C1h	C2h	C3h	C4h

20.16 Media Decoder 相關暫存器

***所有相關的暫存器都在 page 1, ie REG[46h]bit 0=1

PAGE1 REG[0Bh] Interrupt Enable Register (INTEN)

Bit	Description	Default	Access
7-1	N/A	O	RO
0	**IDEC Interrupt Enable Bit** 0: 禁能中斷 1: 致能中斷	0	RW

PAGE1 REG[0Ch] Interrupt Event Flag Register (INTF)

* 如果使用者收到中斷，但是透過這個暫存器卻沒有中斷，那麼使用者應該要去確認 SPI master 狀態暫存器的中斷旗標 REG[BAh]。

Bit	Description	Default	Access
7-1	N/A	0	RW
0	**IDEC Interrupt flag** **Write Function➔Interrupt Clear Bit** 0: 無動作。 1: 清除 IDEC 中斷旗標。 **Read Function➔Interrupt Status** 0: 沒有 IDEC 中斷產生。 1: IDEC 中斷產生。	0	RW

PAGE1 REG[0Dh] Mask Interrupt Flag Register (MINTFR)

*** 如果使用者遮罩中斷旗標，那麼 RA8889 不會發出中斷給 MPU，而 MPU 也不需要去檢查中斷旗標(Interrupt Flag)。但是如果使用只有某些中斷旗標沒有被遮罩掉，那麼 MPU 不會收到中斷，但是 MPU 可以透過檢查中斷旗標以得知中斷產生。

Bit	Description	Default	Access
7-1	N/A	0	RO
0	**Mask IDEC Interrupt Flag** 0: 不遮罩。 1: 遮罩。	0	RW

PAGE1 REG[2Eh] AVI shadow pip start address 0 (avi_spip_sadr0)

Bit	Description	Default	Access
7-2	**memory start address [7:2] for shadow image**	0	RW
1-0	**Fix at 0**	0	RO

PAGE1 REG[2Fh] AVI shadow pip start address 1 (avi_spip_sadr1)

Bit	Description	Default	Access
7-0	**memory start address [15:8] for shadow image**	0	RW

PAGE1 REG[30h] AVI shadow pip start address 2 (avi_spip_sadr2)

Bit	Description	Default	Access
7-0	**memory start address [23:16] for shadow image**	0	RW

PAGE1 REG[31h] AVI shadow pip start address 3 (avi_spip_sadr3)

Bit	Description	Default	Access
7-0	**memory start address [31:24] for shadow image**	0	RW

PAGE1 REG[46h] Page Switch

Bit	Description	Default	Access
7-3	N/A	0	RW
2	**PS8876 Fsck(REG[BBh]) compatible mode** 0: $F_{sck} = F_{core} / (divisor + 1) \times 2$, User don't need modify old program parameter 1: $F_{sck} = F_{core} / (divisor) \times 2$ When user need to use SPI_DIVSOR = 0, $F_{sck} = F_{core}$, set this to 1 Note: this bit is available only on page1	0	RW
1	N/A	0	RW
0	**Page switch** 0: page 0, 少於 256 個暫存器設定 1: page 1, 媒體解碼暫存器設定	0	RW

PAGE1 REG[A0h] – Video Control (VC)

Bit	Description	Default	Access
7	**Media error 媒體格式錯誤，表示不支援的圖像格式或檔頭格式錯誤。 當以上的情形發生時，此位元會變成 1.**	0	RO
6	**Media decoder busy** 0: Media decoder 為閒置 1: Media decoder 為忙碌	0	RO
5	**Media fifo empty** 0: Media FIFO 不是空的 1: Media FIFO 是空的	0	RO
4	**N/A**	0	RO
3	N/A	0	RW
2	N/A	0	RW
1	**N/A**	0	RO
0	**N/A**	0	RO

PAGE1 REG[A1h] – Media Image Height High Byte (MIHH)

Bit	Description	Default	Access
7-0	**Image height 由媒體(BMP/JPEG/AVI) 檔頭得到 Height[15:8]**	0	RO

PAGE1 REG[A2h] – Media Image Height Low Byte (MIHL)

Bit	Description	Default	Access
7-0	**Image height 由媒體(BMP/JPEG/AVI) 檔頭得到 Height[7:0]**	0	RO

PAGE1 REG[A3h] – Media Image Width High Byte (MIWH)

Bit	Description	Default	Access
7-0	**Image width 由媒體(BMP/JPEG/AVI) 檔頭得到 Height[15:8]**	0	RO

PAGE1 REG[A4h] – Media Image Width Low Byte (MIWL)

Bit	Description	Default	Access
7-0	**Image width 由媒體(BMP/JPEG/AVI) 檔頭得到 Height[7:0]**	0	RO

PAGE1 REG[A5h] – Video Frame Period Byte3 (VFPB3)

Bit	Description	Default	Access
7-0	**Video Frame Period 由 AVI 檔頭得到 VFPB[31:24]**	0	RO

PAGE1 REG[A6h] – Video Frame Period Byte2 (VFPB2)

Bit	Description	Default	Access
7-0	**Video Frame Period 由 AVI 檔頭得到 VFPB[23:16]**	0	RO

PAGE1 REG[A7h] – Video Frame Period Byte1 (VFPB1)

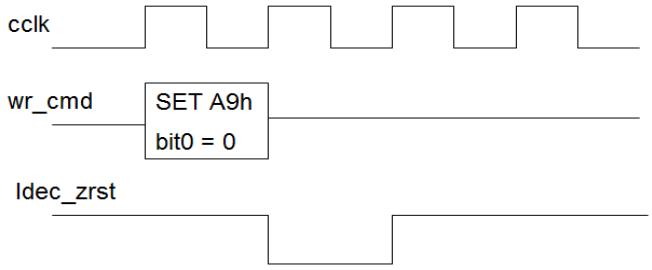
Bit	Description	Default	Access
7-0	**Video Frame Period 由 AVI 檔頭得到 VFPB[15:8]**	0	RO

PAGE1 REG[A8h] – Video Frame Period Byte0 (VFPB0)

Bit	Description	Default	Access
7-0	**Video Frame Period 由 AVI 檔頭得到 VFPB[7:0]**	0	RO

PAGE1 REG[A9h] – Video Control 1 (VC1)

Bit	Description	Default	Access
7-2	**Reserved**	0	RO
1	**Must set 1**	1	RW
0	**Idec reset , clear Idec circuit** 1: 不動作 0: 重置	1	RW

Bit	Description	Default	Access
			

PAGE1 REG[B6h] Serial flash AVI/JPG/BMP (IDEC_CTRL)

Bit	Description	Default	Access
7-6	**IDEC Serial Flash/ROM I/F # Select** 00: 選擇 Serial Flash/ROM 0 I/F 01: 選擇 Serial Flash/ROM 1 I/F 10: 選擇 Serial Flash/ROM 2 I/F 11: 選擇 Serial Flash/ROM 3 I/F	0	RW
5	**N/A**	0	RO
4	**FONT/DMA serial flash sck and data bus select** 0: 選擇 SPI bus 0 以及相關腳位 xmosi, xmiso, xsio2, xsio3 動作 1: 選擇 SPI bus 1 以及相關腳位 xsp1_msio0, xsp1_msio1, xsp1_msio2, xsp1_msio3 動作	0	RW
3	**IDEC sck and data bus select** 0: 選擇 SPI bus 0 以及相關腳位 xmosi, xmiso, xsio2, xsio3) 動作 1: 選擇 SPI bus 1 以及相關腳位 xsp1_msio0, xsp1_msio1, xsp1_msio2, xsp1_msio3) 動作	0	RW
2-1	**IDEC destination Color depth:** 00: 8bit 01: 16 bit 10: 24bit 11: NA	10	RW
0	**Write Function: IDEC Start Bit** MPU 設為 1，然後自動重置為 0 當 fontwr_busy 為 1 時，無法啟動。而且，如果啟用 IDEC，則無法將串行閃存 I / F 設置為文字模式並發送字符代碼。 **Read Function: IDEC Busy Check Bit** 0: Idle 1: Busy 當串行閃存 I / F 處於 IDEC 模式時，SDRAM 中的目標起始位址，目標圖像寬度，顏色深度和地址模式後跟 Canvas 的設置，只能在圖形模式下運行。	0	RW

PAGE1 REG[B7h] Serial flash AVI/JPG/BMP (IDEC_CTRL)

<table>
<tr><th>Bit</th><th>Description</th><th>Default</th><th>Access</th></tr>
<tr><td>7</td><td>Page 1 FONT/DMA Serial Flash/ROM I/F # Select
這個 BIT 需搭配 PAGE0 REG[B7h] bit7 使用
<table><tr><td></td><td>Page0 REG[B7h]Bit7 = 0</td><td>Page0 REG[B7h]Bit7 = 1</td></tr><tr><td>Page1 REG[B7h]Bit7 = 0</td><td>CS0</td><td>CS1</td></tr><tr><td>Page1 REG[B7h]Bit7 = 1</td><td>CS2</td><td>CS3</td></tr></table></td><td>0</td><td>RW</td></tr>
<tr><td>6</td><td>N/A</td><td>0</td><td>RO</td></tr>
<tr><td>5</td><td>Serial Flash/ROM Address Mode
0: 24 bits 定址模式
1: 32 bits 定址模式
如果使用者希望使用 32 bits 定址模式，使用者必須自行輸入 EX4B 命令(B7h) 給串列快閃記憶體，並且設定此 bit 為 1。</td><td>0</td><td>RW</td></tr>
<tr><td>4</td><td>must set to 1</td><td>0</td><td>WO</td></tr>
<tr><td>3-0</td><td>Read Command code & behavior selection
0000b: 1x 讀取命令 03h。讀取速度為 Normal read 速度。資料是由 xmiso 輸入。在位址與資料間不需要空週期。
0010b: 1x 讀取命令 0Bh。為 faster read 速度。資料是由 xmiso 輸入，RA8889 在位址與資料間會塞入 8 個空週期。
0100b: 1x 讀取命令 1Bh。為 fastest read 速度，資料是由 xmiso 輸入。RA8889 在位址與資料間會塞入 16 個空週期
0110b: 2x 讀取命令 3Bh。在 xmiso 與 xmosi 具有交錯資料輸入，在位址與資料間會塞入 8 個空週期 (Dual mode 0)。
1010b: 4x 讀取命令 6Bh。在 xmiso 與 xmosi 與 xsio2 與 xsio3 具有交錯資料輸入。
1100b: 4x read command code – EBh.
4x 讀取命令 EBh。在 xmiso 與 xmosi 與 xsio2 與 xsio3 具有交錯資料輸入。
註：不是所有的 serial flash 都支援以上命令，請根據使用的 serial flash 來選擇正確的讀取命令。</td><td>0</td><td>R/W</td></tr>
</table>

PAGE1 REG[BBh] IDEC Clock divide

Bit	Description	Default	Access
7-0	2'b00: idec_clock = cclk 2'b01: idec_clock = cclk/2 2'b10: idec_clock = cclk/4 2'b11: reserved *註*： 1.暫存器是用來設定 dec_serial flash 及 idec block 的 clock 速度 2.clock 速度必須大於 2 倍的 xosc (10Mhz) 例如: cclk = 50Mhz, Idec_clk = 50/2 = 25>(2*10)Mhz, 則有效 Idec_clk = 50/4,= 12.5 < (2*10)Mhz, 則無效	0	RW

PAGE1 REG[BCh] Serial flash AVI/JPG/BMP Source Starting Address 0 (IDEC_SADR0)

Bit	Description	Default	Access
7-0	**Serial flash IDEC Source START ADDRESS [7:0]** 此暫存器設定串列快閃記憶體的位址 address [7:0]。 直接指定來源圖檔的起始位址。	0	RW

PAGE1 REG[BDh] Serial flash AVI/JPG/BMP Source Starting Address 1 (IDEC_SADR1)

Bit	Description	Default	Access
7-0	**Serial flash IDEC Source START ADDRESS [15:8]** 此暫存器設定串列快閃記憶體的位址 address [15:8]。 直接指定來源圖檔的起始位址。	0	RW

PAGE1 REG[BEh] Serial flash AVI/JPG/BMP Source Starting Address 2 (IDEC_SADR2)

Bit	Description	Default	Access
7-0	**Serial flash IDEC Source START ADDRESS [23:16]** 此暫存器設定串列快閃記憶體的位址 address [23:16]。 直接指定來源圖檔的起始位址。	0	RW

PAGE1 REG[BFh] Serial flash AVI/JPG/BMP Source Starting Address 3 (IDEC_SADR3)

Bit	Description	Default	Access
7-0	**Serial flash IDEC Source START ADDRESS [31:24]** 此暫存器設定串列快閃記憶體的位址 address [31:24]。 直接指定來源圖檔的起始位址。	0	RW

PAGE1 REG[C0h] IDEC (JPG/BMP)Destination Window Upper-Left corner X-coordinates 0 (IDEC_DX0)

Bit	Description	Default	Access
7-0	**Block Mode** 此暫存器定義 IDEC 的底圖 (Canvas) 上目的視窗左上角 X[7:0]。	0	RW

PAGE1 REG[C1h] IDEC (JPG/BMP) Destination Window Upper-Left corner X-coordinates 1 (IDEC_DX1)

Bit	Description	Default	Access
7-0	**Block Mode** 此暫存器定義 IDEC 的底圖 (Canvas) 上目的視窗左上角 X[12:8]。	0	RW

PAGE1 REG[C2h] IDEC (JPG/BMP) Destination Window Upper-Left corner Y-coordinates 0 (IDEC_DY0)

Bit	Description	Default	Access
7-0	**Block Mode** 此暫存器定義 IDEC 的底圖 (Canvas) 上目的視窗左上角 Y[7:0]。	0	RW

PAGE1 REG[C3h] IDEC (JPG/BMP) Destination Window Upper-Left corner Y-coordinates 1 (IDEC_DY1)

Bit	Description	Default	Access
7-0	**Block Mode** 此暫存器定義 IDEC 的底圖 (Canvas) 上目的視窗左上角 Y[12:8]。	0	RW

PAGE1 REG[C5h] IDEC AVI PIP controller (IDEC_PIP)

Bit	Description	Default	Access
7-2	N/A	0	RO
1-0	00b: AVI display buffer use pip1 + shadow pip 01b: AVI display buffer use pip2 + shadow pip 1Xb: AVI display buffer use pip1	00	RW

PAGE1 REG[C6h] IDEC (AVI/JPG/BMP) transfer number 0 (IDEC_TF0)

Bit	Description	Default	Access
7-0	**Image DMA Transfer Number [7:0]** IDEC_TF [31：0]中的數字是圖像容量。	0	RW

PAGE1 REG[C7h] IDEC (AVI/JPG/BMP) transfer number 1 (IDEC_TF1)

Bit	Description	Default	Access
7-0	**Image DMA Transfer Number [15:8]** IDEC_TF [31：0]中的數字是圖像容量。	0	RW

PAGE1 REG[C8h] IDEC (AVI/JPG/BMP) transfer number 2 (IDEC_TF2)

Bit	Description	Default	Access
7-0	**Image DMA Transfer Number [23:16]** IDEC_TF [31：0]中的數字是圖像容量。	0	RW

PAGE1 REG[C9h] IDEC (AVI/JPG/BMP) transfer number 3 (IDEC_TF3)

Bit	Description	Default	Access
7-0	**Image DMA Transfer Number [31:24]** IDEC_TF [31：0]中的數字是圖像容量。	0	RW

PAGE1 REG[CAh] IDEC (JPG/BMP) Destination memory start addr 0 (IDEC_DADR0)

Bit	Description	Default	Access
7-0	**IDEC SDRAM Destination start address [7:0]** 註: 只適用於 JPG/BMP	0	RW

PAGE1 REG[CBh] IDEC (JPG/BMP) Destination memory start addr 1 (IDEC_DADR1)

Bit	Description	Default	Access
7-0	**IDEC SDRAM Destination start address [15:8]** 註: 只適用於 JPG/BMP	0	RW

PAGE1 REG[CCh] IDEC (JPG/BMP) Destination memory start addr 2 (IDEC_DADR2)

Bit	Description	Default	Access
7-0	**IDEC SDRAM Destination start address [23:16]** 註: 只適用於 JPG/BMP	0	RW

PAGE1 REG[CDh] IDEC (JPG/BMP) Destination memory start addr 3 (IDEC_DADR3)

Bit	Description	Default	Access
7-0	**IDEC SDRAM Destination start address [31:24]** 註: 只適用於 JPG/BMP	0	RW

PAGE1 REG[CEh] IDEC (JPG/BMP) Destination Image Width 0(IDEC_DWTH0)

Bit	Description	Default	Access
7-0	**IDEC Destination Image Width [7:0]** 註: 只適用於 JPG/BMP	0	RW

PAGE1 REG[CFh] IDEC (JPG/BMP)　Destination Image Width 1(IDEC_DWTH1)

Bit	Description	Default	Access
7-6	**N/A**	0	RO
5-0	**IDEC Destination Image Width [12:8]** 註: 只適用於 JPG/BMP	0	RW

PAGE1 REG[D3h] – AVI pause

Bit	Description	Default	Access
7-1	**N/A**	0	RO
0	**Pause, the video will be paused when the bit is set** 寫: 1 -- AVI 進入暫停， 0 -- AVI 離開暫停 讀: 1 – AVI 正在暫停 0 – AVI 正在播放	0	RW

PAGE1 REG[D4h] – AVI stop

Bit	Description	Default	Access
7-1	**N/A**	0	RO
0	**Stop, the video will be stopped and exited when the bit is set** 1: 停止功能開啟 0: 無作用	0	RW

21. Summary for GENITOP's Character Supported by RA8889

Table 21-1

● : Supported, — : Not supported

GT21L16T1W supports font	RA8889 Supported Status	Remarks
15X16 dots GB12345 font	●	
15X16 dots BIG5 basic font	●	
15X16 dots JIS0208 basic font	●	The RA8889 can not support the particular fonts which are illustrated in the Table 21-2, caused by the designing bug from GENITOP, but this problem could be solved through the software modification when needed.
15X16 dots Unicode font (Japanese)	●	
5X7 dots ASCII font	—	
7X8 dots ASCII font	—	
6X12 dots ASCII font	—	
8X16 dots ASCII font	●	
8X16 dots bold ASCII font	●	
12 dots ASCII font (Arial)	—	
16 dots ASCII font (Arial)	●	
8X16 dots Latin font	●	
8X16 dots Greek font	●	
8X16 dots Cyril font	●	
12 dots Unicode font (Latin)	—	
12 dots Unicode font (Greek)	—	
12 dots Unicode font (Cyril)	—	
16 dots Unicode font (Latin)	●	
16 dots Unicode font (Greek)	●	
16 dots Unicode font (Cyril)	●	
12 dots Arabia font	—	
12 dots Arabia extendable font	—	
16 dots Arabia font	●	
16 dots Arabia extendable font	●	

Table 21-2: Character code for JIS0208 (RA8889 can not support)

∣ 0135	≦ 0169	≧ 0170	♂ 0173	▽ 0206	▼ 0207	○ 0379	き 0413	ぎ 0414	遡 3344
堕 3436	陳 3636	悌 3680	届 3847	汎 4038	箆 4247	墨 4347	冀 4935	寫 4948	冪 4949
剞 4974	卍 5036	哈 5093	營 5159	堍 5229	幇 5483	憇 5660	掫 5756	斛 5847	晢 5881
桿 5969	淦 6232	箏 6823	絖 6913	繃 6962	閤 7967	霖 8035	騙 8157	熙 8406	˚8503
‖ 8565	≦ 8569	≧ 8570	♂ 8573	￥ 8579					

Table 21-3

GT30L24M1Z supports font	RA8889 Supported Status	Remarks
24X24 dots GB18030 basic font	●	
12X24 dots GB2312 extension font	●	
12X24 dots ASCII font	●	
24 dots ASCII font (Arial)	●	
24 dots ASCII font (Times New Roman)	●	

Table 21-4

GT30L32S4W supports font	RA8889 Supported Status	Remarks
11X12 dots GB2312 basic font	—	
15X16 dots GB2312 basic font	●	
24X24 dots GB2312 basic font	●	
32X32 dots GB2312 basic font	●	
6X12 dots GB2312 extension font	—	
8X16 dots GB2312 extension font	●	
8X16 dots GB2312 special font	●	
12X24 dots GB2312 extension font	●	
16X32 dots GB2312 extension font	●	
5X7 dots ASCII font	—	
7X8 dots ASCII font	—	
6X12 dots ASCII font	—	
8X16 dots ASCII font	●	
12X24 dots ASCII font	●	
16X32 dots ASCII font	●	
12 dots ASCII font (Arial)	—	
12 dots ASCII font (Times New Roman)	—	

GT30L32S4W supports font	RA8889 Supported Status	Remarks
16 dots ASCII font (Arial)	●	
16 dots ASCII font (Times New Roman)	●	
24 dots ASCII font (Arial)	●	
24 dots ASCII font (Times New Roman)	●	
32 dots ASCII font (Arial)	●	
32 dots ASCII font (Times New Roman)	●	

Table 21-5

GT30L16U2W supports font	RA8889 Supported Status	Remarks
11X12 dots Unicode font	—	
15X16 dots Unicode font	●	
8X16 dots Special font	●	
5X7 dots ASCII font	—	
7X8 dots ASCII font	—	
6X12 dots ASCII font	—	
8X16 dots ASCII font	●	
12 dots ASCII font (Arial)	—	
12 dots ASCII font (Times New Roman)	—	
16 dots ASCII font (Arial)	●	
16 dots ASCII font (Times New Roman)	●	
8X16 dots Latin font	●	
8X16 dots Greek font	●	
8X16 dots Cyril font	●	
12 dots Latin font (Arial)	—	
12 dots Greek font (Arial)	—	
12 dots Cyril font (Arial)	—	
12 dots Arabia font (Arial)	—	
12 dots Arabia extendable font (Arial)	—	
16 dots Latin font (Arial)	●	
16 dots Greek font (Arial)	●	
16 dots Cyril font (Arial)	●	
16 dots Arabia font (Arial)	●	
16 dots Arabia extendable font (Arial)	●	

Table 21-6

GT30L24T3Y supports font	RA8889 Supported Status	Remarks
11X12 dots GB2312 basic font	—	
15X16 dots GB2312 basic font	•	
24X24 dots GB2312 basic font	•	
11X12 dots GB12345 basic font	—	
15X16 dots GB12345 basic font	•	
24X24 dots GB12345 basic font	•	
11X12 dots BIG5 basic font	—	
15X16 dots BIG5 basic font	•	
24X24 dots BIG5 basic font	•	
11X12 dots Unicode font	—	
15X16 dots Unicode font	•	
24X24 dots Unicode font	•	
5X7 dots ASCII font	—	
7X8 dots ASCII font	—	
6X12 dots ASCII font	—	
8X16 dots ASCII font	•	
12 dots ASCII font (Arial)	—	
16 dots ASCII font (Arial)	•	
24 dots ASCII font (Arial)	•	

Table 21-7

GT20L24F6Y supports font	RA8889 Supported Status	Remarks
5X7 dots ASCII font	—	
7X8 dots ASCII font	—	
6X12 dots ASCII font	—	
8X16 dots ASCII font	•	
8X16 dots bold ASCII font	•	
12 dots ASCII font (Arial)	—	
12 dots ASCII font (Times New Roman)	—	
16 dots ASCII font (Arial)	•	
16 dots ASCII font (Times New Roman)	•	
24 dots ASCII font (Arial)	•	
8X16 dots Latin font	•	

GT20L24F6Y supports font	RA8889 Supported Status	Remarks
8X16 dots Greek font	●	
8X16 dots Cyril font	●	
8X16 dots Hebrew font	●	
8X16 dots Thai font	●	
12X24 dots Latin font	●	
12X24 dots Greek font	●	
12X24 dots Cyril font	●	
16 dots Arabia font (Arial)	●	
16 dots Latin font (Arial)	●	
16 dots Greek font (Arial)	●	
16 dots Cyril font (Arial)	●	
12 dots Latin font (Arial)	—	
12 dots Greek font (Arial)	—	
12 dots Cyril font (Arial)	—	
24 dots Arabia font (Arial)	●	
8x16 ISO8859-1	●	
8x16 ISO8859-2	●	
8x16 ISO8859-3	●	
8x16 ISO8859-4	●	
8x16 ISO8859-5	●	
8x16 ISO8859-7	●	
8x16 ISO8859-8	●	
8x16 ISO8859-9	●	
8x16 ISO8859-10	●	
8x16 ISO8859-11	●	
8x16 ISO8859-13	●	
8x16 ISO8859-14	●	
8x16 ISO8859-15	●	
8x16 ISO8859-16	●	
5x7 ISO8859-1	—	
5x7 ISO8859-2	—	
5x7 ISO8859-3	—	
5x7 ISO8859-4	—	
5x7 ISO8859-5	—	
5x7 ISO8859-7	—	

GT20L24F6Y supports font	RA8889 Supported Status	Remarks
5x7 ISO8859-8	—	
5x7 ISO8859-9	—	
5x7 ISO8859-10	—	
5x7 ISO8859-11	—	
5x7 ISO8859-13	—	
5x7 ISO8859-14	—	
5x7 ISO8859-15	—	
5x7 ISO8859-16	—	
5x10 LCM Area 0	—	
5x10 LCM Area 1	—	
5x10 LCM Area 2	—	
5x10 LCM Area 3	—	
5x10 LCM Area 8	—	
5x10 LCM Area 11	—	
5x10 LCM Area 12	—	
5x10 LCM Area 13	—	

Table 21-8

GT21L24S1W supports font	RA8889 Supported Status	Remarks
24X24 dots GB2312 basic font	•	
12X24 dots GB2312 extension font	•	
12X24 dots ASCII font	•	
24 dots ASCII font (Arial)	•	

Important Notice

The contents contained in this document are believed to be accurate at the time of publication. RAIO assumes no responsibility for any error in this document, and reserves the right to change the products or specification in this document without notice.

The information contained herein is presented only as a guide or examples for the application of our products. No responsibility is assumed by RAIO for any infringement of patents, copyrights, or other intellectual property rights of third parties which may result from its use. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of RAIO or others.

Any semiconductor devices may have inherently a certain rate of failure. To minimize risks associated with customer's application, adequate design and operating safeguards against injury, damage, or loss from such failure, should be provided by the customer when making application designs.

RAIO‘s products are not authorized for use in critical applications such as, but not limited to, life support devices or system, where failure or abnormal operation may directly affect human lives or cause physical injury or property damage. If products described here are to be used for such kinds of application, purchaser must do its own quality assurance testing appropriate to such applications.