

## 第九章 多媒體卡控制器

多媒體卡是由多媒體卡組織所制定的標準。該組織是於 1998 年，由 14 家公司所創始的。其目的是推動全世界採用一種可以在小型與低耗電的設備上儲存與擷取數位資訊的標準。多媒體卡是目前最小的可移除式固態記憶體之一。

多媒體卡的重量約小於兩公克，而大小大約是一張郵票的大小。適合用來做行動裝置的可移除式儲存媒體。可應用在如 MP3 隨身聽，行動電話，PDA，數位相機等等。其容量可至 512MB 以上。

多媒體卡對唯讀的應用使用 ROM 技術，對讀寫的應用則使用快閃記憶體 (flash memory) 的技術。在其封裝上包含七個墊狀的序列介面，使多媒體卡很容易可以整合到許多的裝置中。

未來多媒體卡將可透過平行埠，通用序列匯流排(USB)或 PC 介面卡與 PC 進行資料，影像，與聲音檔案的傳輸。同時多媒體卡採用 DOS/Window 檔案結構，提供與一般 PC 相容的能力。

此外，在安全性上，多媒體卡提供一組安全命令，以符合未來的消費，商業與工業的應用需求。例如行動電子商務，銀行與著作權保護等。特別是在與攜式裝置結合的應用上。

目前所提的安全多媒體卡標準期望與標準的多媒體卡完全相容。這個安全多媒體卡標準包刮特殊的防修改模組，並提供與在智慧卡(SmartCard)及信用卡同類的安全技術。

### 9.1 概觀

PXA250 與 PXA210 應用處理器的多媒體卡 (MMC) 控制器連接存取應用處理器的軟體與 MMC 堆疊 (一個記憶卡集合)。MMC 控制器支援 MMC 系統，為一個低成本的資料儲存與溝通系統。關於 MMC 系統詳細的說明可在 MMC 協會的網站 [www.mmca.org](http://www.mmca.org) 上取得。應用處理器的 MMC 控制器是以 *The MultiMediaCard System Specification Version 2.1* 的標準為基礎，但不支援 1 位元

組與 3 位元組傳送，以及 1023 的最大區塊長度。

MMC 控制器支援從標準 MMC 或序列週邊介面（SPI）匯流排至 MMC 堆疊的解譯協定。用來存取應用處理器的軟體必須指示使用 MMC 或 SPI 模式作為與 MMC 控制器溝通的協定。

MMC 控制器特徵：

- 資料傳送率最高可達 20Mbps
- 回應 FIFO
- 雙重接收資料 FIFO
- 雙重傳送 FIFO
- 不論 MMC 或 SPI 模式皆可支援 2 個 MMC

MMC 控制器包含所有卡片特殊的功能，為 MMC 系統提供匯流排 master 服務，並提供卡片堆疊的標準界面。控制器處理卡片初始化、CRC 的產生與確認、命令、回復，與資料交易。

MMC 控制器為軟體的 slave，包含命令與控制暫存器、回應 FIFO，與資料 FIFO。軟體存取這些暫存器與 FIFO，並且產生命令、中斷回應，與控制後續動作。

圖 9-1 為典型的 MMC 堆疊、MMC 控制器與軟體的互動區塊圖。

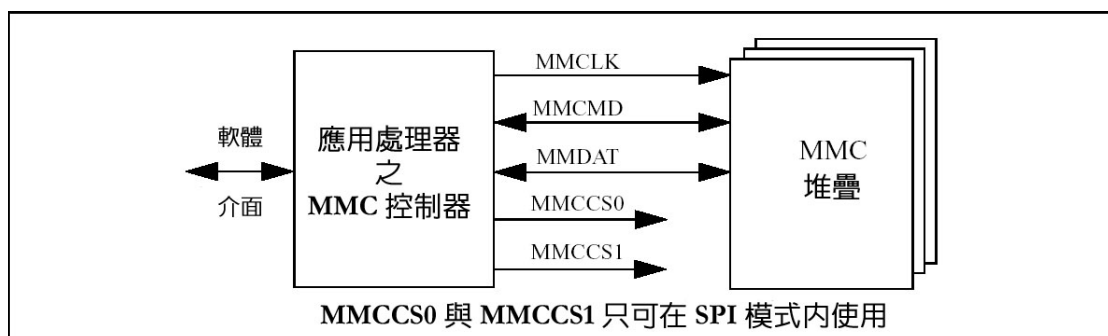


圖 9-1 MMC 系統互動圖

MMC 匯流排連結卡片堆疊與控制器。軟體與控制器可開啟或關閉 MMC 時脈。卡片堆疊與控制器經由命令與資料線連續溝通，並且實作一以訊息為基礎的協定。訊息包含下列的符號（token）：

- 命令：6 位元組命令符號啟動操作。命令集包含卡片初始化、卡片暫存

器讀取與寫入、資料傳送...等。MMC 控制器在 MMCMD 命令時會連續傳送命令符號。命令符號的格式顯示於表 9-1 中。

**表 9-1 命令符號格式**

位元位置	47	46	[45:40]	[39:8]	[7:1]	0
寬 (位元)	1	1	6	32	7	1
值	0	1	×	×	×	1
說明	起始位元	傳送位元	命令索引	參數	CRC7	結束位元

- 回應：回應符號為命令符號的回覆。每個命令都具有特定的回應類型或是無回應類型。回應符號的格式是取決於所預期的回應與卡片的模式。回應符號格式在 *The MultiMediaCard System Specification Version 2.1* 中有詳細說明。
- 資料：資料在控制器與卡片之間以 8 位元區塊做連續的傳送，速率可達 20Mbps。資料符號的格式取決於卡片的模式。表 9-2 為 MMC 模式之資料符號格式，表 9-3 為 SPI 模式之資料符號格式。

**表 9-2 MMC 資料符號格式**

串流資料	1	×	無 CRC	1
區塊資料	0	×	×	1
說明	起始位元	資料	CRC7	結束位元

**表 9-3 SPI 資料符號格式**

值	11111110	×	×
說明	起始位元組	資料	CRC16

在 MMC 模式中，所有的操作都包含命令符號，而大部分的命令具有一個相關聯的回應符號。讀取與寫入命令也具有資料符號。在雙向的 MMCMD 訊號中會傳送與接收命令與回應符號，在雙向的 MMDAT 訊號中會傳送與接收資料符號。圖 9-2 為典型的具有回應及無回應的 MMC 模式命令時序圖表，圖 9-3 為連續的讀取與寫入之典型的 MMC 模式時序圖表。

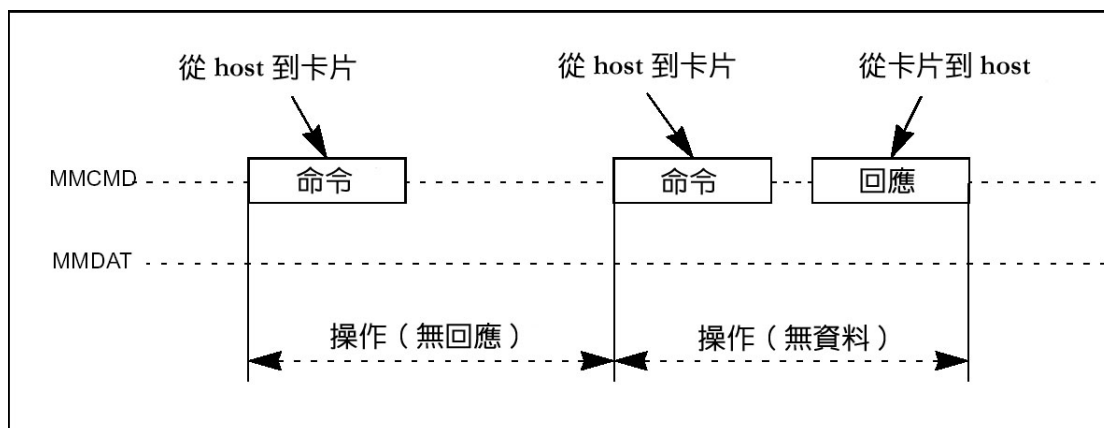


圖 9-2 無資料符號之 MMC 模式操作

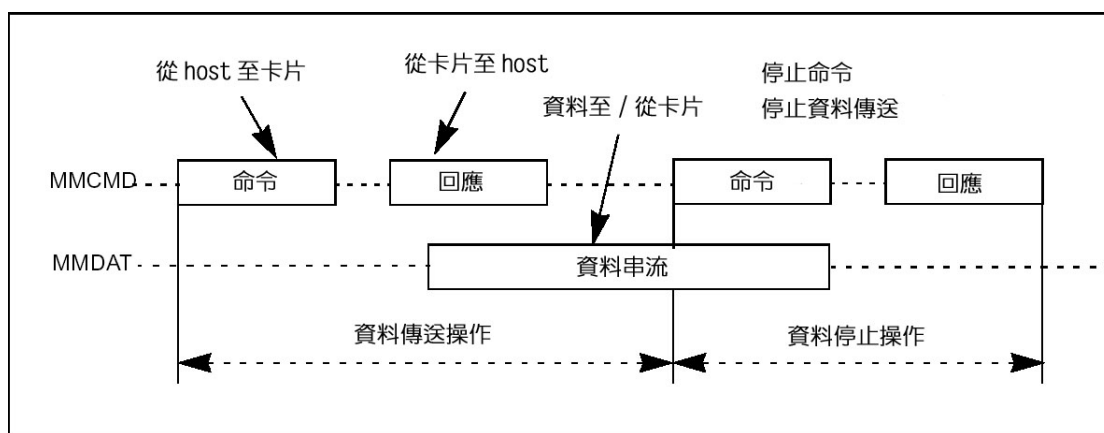


圖 9-3 有資料符號之 MMC 模式操作

在 SPI 模式中，所有的命令都可使用。可使用的命令都具有命令與回應符號。MMCMD 與 MMDAT 訊號在 SPI 模式中為非雙向的。MMCMD 為一與應用處理器相關的輸出，而 MMDAT 為一與應用處理器相關的輸入。在 MMCMD 訊號中會寫入與傳送命令與資料符號，在 MMDAT 訊號中會接收回應與讀取符號。圖 9-4 為典型的無資料符號的 SPI 模式時序圖表。圖 9-5 與圖 9-6 分別為 SPI 模式讀取與寫入時序圖表。

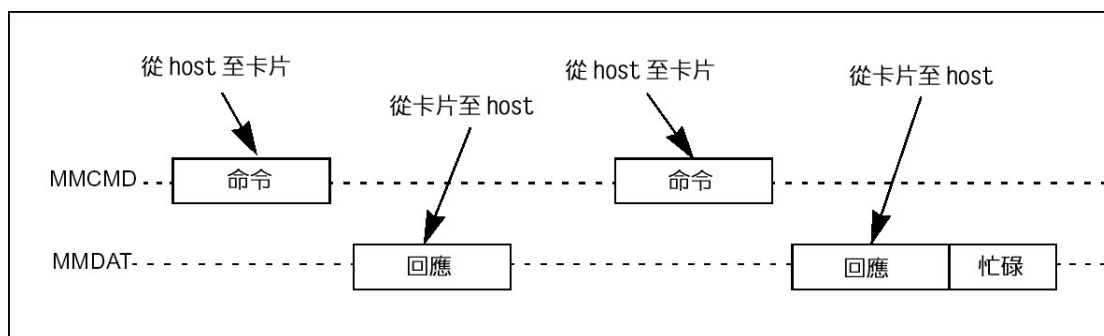


圖 9-4 無資料符號的 SPI 模式操作

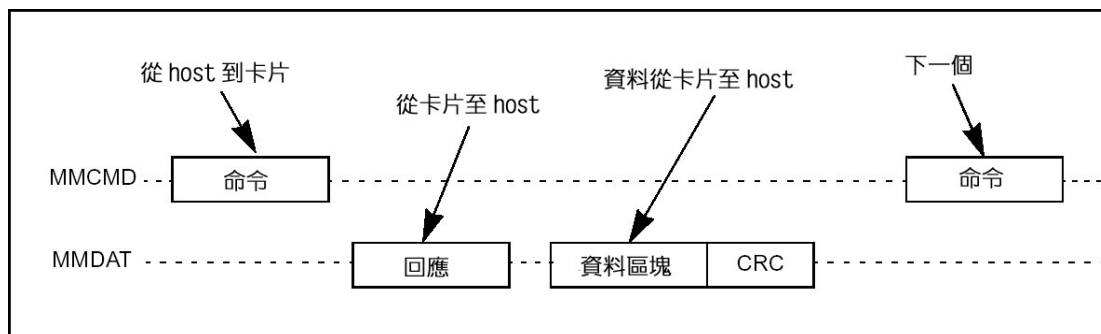


圖 9-5 SPI 模式讀取操作

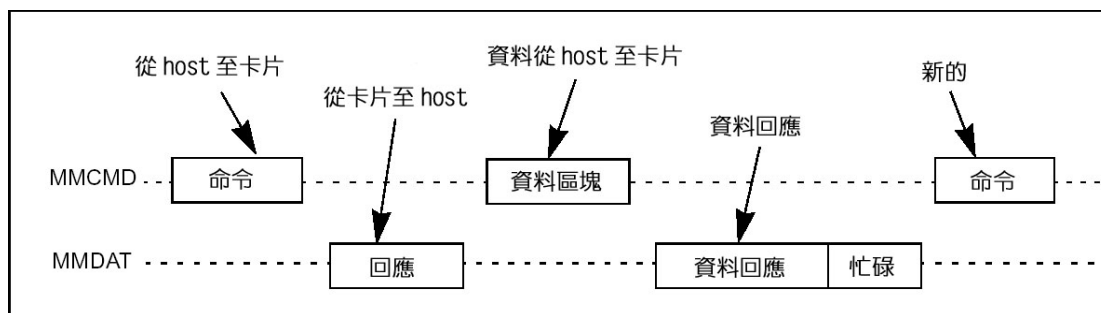


圖 9-6 SPI 模式寫入操作

注意：此控制器不支援 1 位元組與 3 位元組之資料傳送。串流（stream）寫入只支援 10 位元組會更多位元組的資料傳送。

關於 MMC 與 SPI 操作模式的詳細資料請參考 *The MultiMediaCard System Specification*。

## 9.2 MMC 控制器功能說明

軟體必須讀取與寫入 MMC 控制器暫存器與 FIFO 來初始啟動與卡片的溝通。

MMC 控制器為軟體與 MMC 匯流排之間的介面。負責軟體與 MMC 匯流排之間的時序與協定。其包含控制與狀態暫存器、一個 16 位元的回應 FIFO（8 個通道長）、兩個 8 位元的接收資料 FIFO（32 個通道長），與兩個 8 位元的傳送 FIFO（32 個通道長）。暫存器與 FIFO 可由軟體來存取。

MMC 控制器也會藉由緩衝資料與產生並確認 CRC 來啟動最小資料延遲。

範例請參考 9.4 節「多媒體卡控制器操作」。

### 9.2.1 訊號說明

MMC 控制器訊號為 MMCLK、MMCMD、MMDAT、MMCCS0，與 MMCCS1。表 9-4 說明每個訊號的功能。

表 9-4 MMC 訊號說明

訊號名稱	輸入／輸出	說明
MMCLK	輸出	MMC 時脈訊號
MMCMD	雙向	命令線
MMDAT	雙向	資料線
MMCCS0	輸出	晶片選擇 0（僅於 SPI 模式中使用）
MMCCS1	輸出	晶片選擇 1（僅於 SPI 模式中使用）

MMCLK、MMCCS0，與 MMCCS1 訊號會按照路線傳送經過 GPIO 內所選擇的功能。請參考 4.1 節「通用 I/O」來得到分配這些訊號至特定 GPIO 的程序說明。雖然有很多 GPIO 要分配到訊號，但每個訊號必須程式化為其中一個可能的 GPIO。請參考 4.1.2 節「GPIO 選擇功能」以獲得 GPIO 選擇功能的完整說明。

### 9.2.2 MMC 控制器重置

MMC 控制器只可藉由應用處理器的硬重置或軟重置來重置。重置應用處理器與 MMC 控制器的方法可在 2.7 節「重置」中找到。在任何重置之後，會將所有的暫存器與 FIFO 控制設定為其初始值。

### 9.2.3 卡片初始化程序

在重置之後，MMC 卡必須藉由在 MMCLK 訊號上傳送 80 個時脈來初始化。將 MMC\_CMDAT[INIT]位元設定為 1 來初始化 MMC 卡。在目前 MMC\_CMD 暫存器內的命令之前會傳送 80 個時脈。此功能可用來取得已插入匯流排的新卡片。初始化程式期間並不會觸發晶片選擇。

在 80 個時脈初始化程序之後，軟體必須藉由載入適當的命令索引至 MMC\_CMD 暫存器內來連續傳送 CMD1（請參考表 9-19 之命令說明），直到卡片指示已完成供電程序。而後軟體可分配位址給卡片將其設定為 SPI 模式。

## 9.2.4 MMC 與 SPI 模式

在重置之後，MMC 卡會處於 MMC 模式中。卡片可維持在 MMC 模式中，或是藉由設定 MMC\_SPI 暫存器位元來調整組態改變為 SPI 模式。以下的章節簡單說明此兩個模式。

### 9.2.4.1 MMC 模式

在 MMC 模式中，MMCMD 與 MMDAT 訊號為雙向的，而且需要外部 pullup。經由 MMCMD 訊號可傳送與接收命令與回應符號(token)，經由 MMDAT 訊號可讀取與寫入資料。

在 MMC 卡啟動之後，會分配到一個預設的相關卡片位址 (RCA) 0x0001。在初始化程序 (說明於 15.2.3 節「卡片初始化程序」) 期間，軟體會分配不同的位址給每個卡片。而後卡片會由命令符號參數部分的新的相關位址 (由 7 位元的 CRC 所保護，請參考表 9-1) 來定址。關於當多個卡片連結至系統時的識別程序，請參考 *The MultiMediaCard System Specification* 中的卡片識別程序。

回應符號具有五種格式，其中包含一個無回應符號。回應符號長度為 48 或 136 位元，且可能由 7 位元的 CRC 所保護。在 *The MultiMediaCard System Specification* 中可獲得回應符號的詳細資訊。

在寫入資料傳送中，資料尾部會附加來自卡片的 5 位元 CRC 狀態符號。在 CRC 狀態符號之後，卡片可能藉由將 MMDAT 線拉下為低電位來指示其為忙碌狀態。

對讀取操作，開始位址可為卡片記憶體內有效位址空間的任何隨機的位元組位址。對寫入操作，開始位址必須在邊界區段 (boundary sector) 上，而資料長度必須為區段長度的整數倍數。區段為在寫入操作期間會被消除的區塊 (block) 的數目，而且對每一個 MMC 卡皆為固定的。區塊為所傳送的位元組的數目。

MMC 模式支援下列資料傳送模式：

- 單一區塊讀取／寫入：在單一區塊模式中，會傳送單一資料區塊。起始位址由所使用的讀取或寫入命令之命令符號所指定。軟體必須藉由輸入傳送的位元組數目至 MMC\_BLKLEN 暫存器內來設定控制器那個區塊

大小。資料區塊由 16 位元的 CRC 所保護，藉由傳送單元來產生且藉由接收單元來確認。在傳送九最後一個資料位元之後，CRC 會附加在資料後面。

- 多重區塊讀取／寫入：在多重區塊模式下，會傳送多個資料區塊。每個區塊長度與軟體在 MMC\_BLKLEN 暫存器內所指定的長度相同。資料區會儲存或從連續的記憶體位址來檢索，而其起始位址由命令符號所指定。軟體指定 MMC\_NOB 暫存器內所傳送的區塊數目。每個資料區塊由附加的 16 位元 CRC 來保護。多重區塊資料傳送由停止傳送命令所終止。
- 串流讀取／寫入：在串流（stream）模式下，會傳送連續的資料。起始位址由使用的讀取或寫入命令之命令符號所指定。資料串流由停止傳送命令所終止。對寫入傳送，停止傳送命令必須和最後的六個資料位元組一起傳送。確保寫入正確的資料量至卡片。對讀取傳送，停止傳送命令可能在引發資料傳送之後發生。在此模式中，資料並無 CRC 保護。

#### 9.2.4.2 SPI 模式

SPI 模式為一選擇的第二個溝通協定。在 SPI 模式中，MMCMD 與 MMDAT 線為單向的，而且只允許單一區塊資料傳送。MMCMD 訊號為控制器的輸出，用來傳送命令符號與寫入資料至 MMC 卡。MMDAT 訊號為控制器的輸入，用來接收來自 MMC 卡的回應符號與讀取資料。

注意：當卡片為 SPI 模式時，要回復為 MMC 模式為一的方法為開啟卡片的電源。

卡片定址由硬體晶片選擇來實行，為 MMCCS1 和 MMCCS0。所有的命令、回應，與資料符號都為 8 位元長，並且在傳送之後會立即觸發各自的晶片選擇。

命令符號由 7 位元的 CRC 所保護。卡片總是傳送一個回應給命令符號。回應符號具有 4 種格式，包括一個 8 位元的錯誤回應。回應符號的長度為 1、2 或 5 個位元組。

SPI 模式提供一個無保護的模式。在此模式下，命令、回應，與資料符號內仍然需要 CRC 位元，但卡片與控制器會忽略這些位元。

在寫入資料傳送中，資料尾部會加上一個來自卡片的 8 位元的 CRC 狀態符號。如同在 MMC 模式，卡片可能在狀態符號之後藉由將 MMDAT 線拉下為低電位來表示其為忙碌狀態。在讀取資料傳送時，卡片可能回應資料或 1 位元組長的資料錯誤符號。



### 9.2.5 錯誤偵測

MMC 控制器偵測 MMC 匯流排上的下列錯誤，並回報給狀態暫存器 (MMC\_STAT)：

- 回應 CRC 錯誤：在命令回應上發生計算 CRC 錯誤。
- 回應逾時：在特定的時脈數目之前不會開始回應。
- 寫入資料 CRC 錯誤：卡片傳回資料上的 CRC 狀態錯誤。
- 讀取資料 CRC 錯誤：資料上發生計算 CRC 錯誤。
- 讀取逾時：在特定的時脈數目之前不會開始讀取資料操作。
- SPI 資料錯誤：SPI 模式內偵測出讀取資料錯誤符號。

### 9.2.6 中斷

MMC 控制器產生中斷來發出命令程序的狀態訊號。軟體負責適時地遮蔽中斷、驗證中斷，並在需要的時候執行適當的動作。

在 9.5.11 節「MMC\_I\_MASK 暫存器」與 15.5.12 節「MMC\_I\_REG 暫存器」對於中斷與遮蔽有詳細的說明。CMDAT[DMA\_EN] 位元也會遮蔽 MMC\_I\_MASK[RXFIFO\_RD\_REQ] 與 MMC\_I\_MASK[TXFIFO\_WR\_REQ] 中斷位元。

### 9.2.7 時脈控制

MMC 控制器與軟體都可藉由開啟或關閉 MMC 匯流排時脈 (MMCLK) 來控制 MMC 匯流排時脈。這有助於控制資料流 (data flow) 來預防 underrun 與溢位 (overflow)，也可節省電源。軟體也可隨時改變頻率以達到卡片識別頻率所指定的最大資料傳送速率。

MMC 控制器具有一個內部頻率產生器，可啟動、停止，與劃分 MMC 匯流排時脈。軟體可能藉由設定 MMC\_STRPCL 暫存器適當的位元來啟動與停止時脈。MMCLK 頻率由寫入 MMC\_CLKRT 暫存器內的數值來控制。

為了下一個命令程序寫入任何 MMC 控制器暫存器，軟體必須：

1. 停止時脈

2. 寫入暫存器
3. 重新啟動時脈

當欲讀取或接收 FIFO 或寫入傳送 FIFO 時，軟體不可停止時脈。當時脈停止，它會重置 FIFO 內的指標，而且不可傳送或存取留在 FIFO 內的任何資料。當接收 FIFO 為空的且已設定 MMC\_STAT[DATA\_TRAN\_DONE]時，軟體可能停止時脈。

軟體可藉由設定 MMC\_CLKRT 暫存器來指示 20 Mhz 時脈的時脈除數。時脈速率可能會設定為下列數值：

- 20 Mhz
- 20 Mhz 的 1/2，即 10 Mhz
- 20 Mhz 的 1/4，即 5 Mhz
- 20 Mhz 的 1/5，即 2.5 Mhz
- 20 Mhz 的 1/16，即 1.25 Mhz
- 20 Mhz 的 1/32，即 625 khz
- 20 Mhz 的 1/64，即 312.5 khz

控制器也可自動關閉時脈。若在資料讀取期間，兩個接收 FIFO 都變成滿載，或一個接收 FIFO 被軟體所讀取而定一個接收 FIFO 成為滿載，或是在資料寫入期間兩個傳送 FIFO 都變為空的，或是一個傳送 FIFO 被軟體寫入而定一個傳送 FIFO 為空的，則控制器會自動關閉時脈以預防資料溢位或是 underrun。對讀取資料傳送，在接收 FIFO 被清空時控制器會再次啟動時脈。對寫入資料傳送，在傳送 FIFO 不再為空的時，控制器會再次啟動時脈。

警告：當資料在傳送 FIFO 或接收 FIFO 內時停止時脈會造成無法預期的結果。

若軟體在任何時候停止時脈，則在繼續進行之前必須先等待清除 MMC\_STAT[CLK\_EN]狀態位元。

## 9.2.8 資料 FIFO

回應符號、接收資料，與傳送資料的控制器 FIFO 分別為 MMC\_RES、MMC\_RXFIFO，與 MMC\_TXFIFO。這些 FIFO 可由軟體來存取，在以下段落中會做說明。

### 9.2.8.1 回應資料 FIFO (MMC\_RES)

回應 FIFO，即 MMC\_RES，包含控制器傳送命令之後從 MMC 卡所接收的回應。MMC\_RES 為唯讀，為一 8 個通道長的 FIFO。

FIFO 會保持所有可能的回應長度。只有 1 位元組長度的回應位於 FIFO 內的 16 位元通道之 MSB。從回應 FIFO 讀取的第一個半字組為所接收的回應之最高有效半字組。

FIFO 不包含回應 CRC。CRC 確認之狀態位於狀態暫存器 MMC\_STAT 內。

#### 9.2.8.2 接收資料 FIFO (MMC\_RXFIFO)

兩個接收資料 FIFO 對軟體來說是唯讀的，而以單一位元組為基礎則為可讀取的。其為雙重 FIFO (dual FIFO)，每一個 FIFO 之 1 位元組資料為 32 個通道。控制器控制存取 FIFO，而且取決於 FIFO 的狀態。

在系統重置之後與所有命令程序開始時，會清除兩個 FIFO 與其控制來啟動狀態。

FIFO 在軟體與 MMC 匯流排之間替換 (swap)。在任何時候，當軟體讀取其中一個 FIFO 時，MMC 匯流排會寫入至另一個 FIFO。

為了方便舉例，將 FIFO 稱為 RXFIFO1 與 RXFIFO2。在重置之後或是在命令程序的開始，兩個 FIFO 都為空的，而且軟體會讀取 RXFIFO1，MMC 會寫入 RXFIFO2。當 RXFIFO2 變為滿載而 RXFIFO1 為空的時，FIFO 會替換，而軟體會讀取 RXFIFO2，MMC 會寫入 RXFIFO1。當 RXFIFO1 成為滿載而 RXFIFO2 為空的時，FIFO 會替換，而軟體會讀取 RXFIFO1，MMC 會寫入 RXFIFO2。

此替換程序在資料傳送時會繼續執行，而且軟體與 MMC 控制器不會有任何感覺。

若在任何時間兩個 FIFO 都成為滿載而且資料傳送未完成，控制器會關閉 MMCLK，以預防發生溢位。當時脈已關閉，來自卡片的資料傳送會停止，直到再次啟動時脈。在軟體清空所連接的 FIFO 之後，控制器會啟動時脈來繼續執行資料傳送。

軟體所連接的 FIFO 之完整狀態會被註冊至 MMC\_STAT[RECV\_FIFO\_FULL]位元之中。

接收 FIFO 在位元組邊界上為可讀取的，而只有在每個 FIFO 存取（每個可存取的 32 個位元組）時，才會觸發 FIFO 讀取請求。因此除了最後一次讀取可能少於 32 個位元組之外，每個請求都必須讀取 32 個位元組。

使用 DMA 時，必須程式化 DMA 來做 32 個位元組突發傳輸的 1 個位元組讀取。最後一次讀取可少於 32 個位元組突發傳輸。以下為一些範例：

- 接收 96 個資料位元組：
  - 讀取 32 個位元組三次。
  - 對於 DMA，使用三個 32 位元組的敘述元與 32 位元組的突發傳輸。
- 接收 98 個資料位元組：
  - 讀取 32 個位元組三次，然後讀取 2 個位元組。
  - 對於 DMA，使用三個 32 位元組的敘述元與 32 位元組的突發傳輸，以及一個 2 位元組的敘述元與 8、16 或 32 位元組的突發傳輸。
- 接收 105 個位元組：
  - 讀取 32 個位元組三次，然後讀取 9 個位元組。
  - 對於 DMA，使用三個 32 位元組的敘述元與 32 位元組的突發傳輸，以及一個 9 位元組的敘述元與 16 或 32 位元組的突發傳輸。

### 9.2.8.3 傳送資料 FIFO (MMC\_TXFIFO)

兩個傳送資料 FIFO 只可以由軟體寫入，而且在單一位元組基礎上才能被寫入。其為雙重 FIFO (dual FIFO)，每個 FIFO 之 1 位元組資料為 32 個通道。由控制器來控制 FIFO 的存取，並取決於 FIFO 的狀態。

在系統重置後以及所有指令程序的開始，會將兩個 FIFO 和其控制清除為最初狀態。

FIFO 在軟體與 MMC 匯流排之間替換 (swap)。在任何時候，當軟體寫入其中一個 FIFO，MMC 匯流排就會讀取另一個 FIFO。

為了舉例方便，將兩個 FIFO 稱為 TXFIFO1 和 TXFIFO2。在重置之後或是在命令程序的開始時，兩個 FIFO 都為空的，而軟體會寫入 TXFIFO1，MMC 會讀取 TXFIFO2。當 TXFIFO1 成為滿載且 TXFIFO2 為空的時，FIFO 會替換，而軟體會寫入 TXFIFO2，MMC 會讀取 TXFIFO1。當 TXFIFO2 成為滿載而 TXFIFO1 為空的時，FIFO 會替換，而軟體會寫入 TXFIFO1，MMC 會讀取 TXFIFO2。

此替換程序在資料傳送時會繼續執行，而且軟體與 MMC 控制器不會有任

何感覺。

若在任何時間兩個 FIFO 都成為空的而且資料傳送未完成，控制器會關閉 MMCLK，以預防發生 underrun。當時脈已關閉，至卡片的資料傳送會停止，直到再次啟動時脈。當傳送 FIFO 不再為空的時，MMC 控制器會自動重新啟動時脈。

若軟體未填滿所連結的 FIFO，則必須將 MMC\_PRTBUF[BUF\_PART\_FULL] 設定為 1。這會啟動 FIFO 替換而不用填滿 FIFO。

軟體所連結的 FIFO 之空狀態會註冊到 MMC\_STAT[XMIT\_FIFO\_EMPTY] 位元內。

接收 FIFO 在位元組邊界上為可寫入的，而只有在每個 FIFO 存取（每個可存取的 32 個通道）時，才會觸發 FIFO 寫入請求。因此除了最後一次寫入可能少於 32 個位元組之外，每個請求都必須寫入 32 個位元組。

使用 DMA 時，必須程式化 DMA 來做 32 個位元組突發傳輸的 1 個位元組寫入。

最後一次寫入可少於 32 個位元組突發傳輸。

若最後一次的寫入少於 32 個位元組，則必須設定 MMC\_PRTBUF[BUF\_PART\_FULL]位元。當使用 DMA 時，必須程式化最後一個敘述元以允許在寫入資料至 FIFO 之後，DMA 可設定一個中斷。在中斷發生之後，軟體必須設定 MMC\_PRTBUF[BUF\_PART\_FULL]位元。

以下為一些範例：

- 傳送 96 個資料位元組：
  - 寫入 32 個位元組三次。
  - 對於 DMA，使用三個 32 位元組的敘述元與 32 位元組的突發傳輸。
- 傳送 98 個資料位元組：
  - 寫入 32 個位元組三次，然後寫入 2 個位元組。
  - 對於 DMA，使用三個 32 位元組的敘述元與 32 位元組的突發傳輸，以及一個 2 位元組的敘述元與 8、16 或 32 位元組的突發傳輸，並程式化敘述元來設定一個中斷，讓軟體寫入 MMC\_PRTBUF[BUF\_PART\_FULL]位元。

- 傳送 105 個位元組：
  - 寫入 32 個位元組三次，然後寫入 9 個位元組。
  - 對於 DMA，使用三個 32 位元組的敘述元與 32 位元組的突發傳輸，以及一個 9 位元組的敘述元與 16 或 32 位元組的突發傳輸，並程式化敘述元來設定一個中斷，讓軟體寫入 MMC\_PRTBUF[BUF\_PART\_FULL]位元。

#### 9.2.8.4 DMA 與程式 I/O

軟體可能會經由 DMA 或程式 I/O 來和 MMC 控制器做溝通。

為了以 DMA 存取 FIFO，軟體必須程式化 DMA 以單一位元組傳送與 32 位元組突發傳送方式來讀取或寫入 MMC FIFO。舉例說明，為了寫入 64 個資料位元組至 MMC\_TXFIFO，軟體必須程式化 DMA 以 8 個位元的埠大小來寫入 64 個位元組至 MMC 與 32 個位元組突發傳輸。MMC 會發出讀取 MMC\_RXFIFO 與寫入 MMC\_TXFIFO 的請求。

若使用程式 I/O，則在讀取或寫入個別的 FIFO 之前，軟體會等待 MMC\_I\_REG[RXFIFO\_RD\_REQ]或 MMC\_I\_REG[TXFIFO\_WR\_REQ]。

必須將 CMDAT[DMA\_EN]位元設定為 1 來啟動與 DMA 的溝通，而必須設定為 0 來啟動程式 I/O。

### 9.3 卡片溝通協定

此章節說明軟體所負責的工作與 MMC 和卡片之間所使用的溝通協定。

#### 9.3.1 基本、無資料、命令與回應程序

MMC 控制器執行基本的 MMC 或 SPI 匯流排交易。它會從命令暫存器將命令格式化，若可應用則會產生並附加 7 位元 CRC。而後會連續將其解譯至 MMCMD 匯流排、集結回應資料，以及確認回應 CRC。若可應用，MMC 控制器也會確認回應逾時與卡片忙碌。回應資料位於 MMC\_RES FIFO，而交易狀態位於狀態暫存器 MMC\_STAT 內。

軟體之事件協定為：

1. 停止時脈
2. 寫入 0x6 至 MMC\_I\_MASK 暫存器並等待與驗證 MMC\_I\_REG[CLK\_IS\_OFF]中斷。
3. 寫入至下列暫存器：
  - MMC\_CMD
  - MMC\_ARGH
  - MMC\_ARGL
  - MMC\_CMDAT，就算此暫存器內容沒有改變也必須寫入。
  - MMC\_CLKRT
  - MMC\_SPI
  - MMC\_RESTO
4. 啟動時脈
5. 寫入 0x7b 至 MMC\_I\_MASK 暫存器並等待與驗證 MMC\_I\_REG[END\_CMD\_RES]中斷。
6. 讀取 MMC\_RES FIFO 與 MMC\_STAT 暫存器。

命令的結果可能會使某些卡片變為忙碌狀態。軟體可能藉由寫入 MMC\_I\_MASK 暫存器並等待 MMC\_I\_REG[PRG\_DONE]中斷來等待卡片變為不忙碌，或是軟體可啟動與另一個卡片的溝通。軟體可能不會存取相同的卡片，直到此卡片不再忙碌。關於額外的資訊請參考 *The MultiMediaCard System Specification*。

### 9.3.2 資料傳送

資料傳送為一個至卡片的附加資料傳送之命令與回應程序。

請參考 9.4 節「多媒體卡（MMC）控制器操作」之範例。

軟體必須遵照 9.3.1 節「基本、無資料、命令與回應程序」中所描述的步驟。另外，在啟動時脈之前，軟體必須寫入下列暫存器：

- MMC\_RDTO
- MMC\_BLKLEN
- MMC\_NOB

在軟體寫入這些暫存器並啟動時脈之後，軟體必須讀取上述的 MMC\_RES，並讀取或寫入 MMC\_RXFIFO 或 MMC\_TXFIFO。

在完成讀取或寫入資料 FIFO 之後，軟體必須等待適當的中斷。必須讀取狀態暫存器 MMC\_STAT 來確定交易已完成並確認交易狀態。

當使用 DMA 請求訊號時，在 FIFO 準備好讀取或寫入時控制器會指示 DMA。預計所有的 FIFO 讀取與寫入會清空與填滿所連接的 FIFO。若在任何時候軟體未填滿（32 位元組）MMC\_TXFIFO，則軟體必須藉由設定 MMC\_PRTBUF[BUF\_PART\_FULL]來通知控制器。軟體可寫入比需要寫入 MMC\_TXFIFO 的資料更多的資料位元組，但控制器只會傳送 MMC\_BLKLEN 暫存器所指定的位元組數目。

在任何資料傳送的最後，或是 MMC 會流排上的忙碌訊號，在觸發 MMC\_I\_REG[DATA\_TRAN\_DONE]中斷來通知軟體已完成資料傳送之前，MMC 控制器會等待 8 個 MMC 時脈。這可保證 8 個 MMC 時脈的指定最小量會在資料傳送與下一個命令之間發生。

在寫入資料傳送時，當程式化資料時卡片可能變為忙碌狀態。軟體可能會藉由寫入 MMC\_I\_MASK 暫存器與等待 MMC\_I\_REG[PRG\_DONE]中斷來等待卡片變為不忙碌，或是軟體可啟動與另一個卡片的溝通。關於更多的資訊請參考 *The MultiMediaCard System Specification*。

MMC 控制器在所有的模式模式中執行資料交易：單一區塊、多重區塊，與串流模式。

### 9.3.2.1 區塊資料寫入

在單一區塊資料寫入中，會寫入一個資料區塊至卡片。在多重區塊寫入中，控制器執行 MMC 匯流排上的多重單一區塊寫入資料傳送。

在開啟時脈來啟動命令程序之後，軟體必須程式化 MDA 來填滿 MMC\_TXFIFO（寫入 32 個位元組）。軟體必須繼續填滿 FIFO 直到將所有的資料寫入至 FIFO 中。然後軟體必須藉由等待 MMC\_I\_REG[DATA\_TRAN\_DONE]中斷與 MMC\_I\_REG[PRG\_DONE]中斷來等待傳送完成。之後軟體可讀取狀態暫存器 MMC\_STAT 來驗證交易狀態。

對多重區塊寫入，*The MultiMediaCard System Specification* 說明卡片會繼續接收資料區塊，直到接收到停止傳送命令。在控制器傳送 MMC\_NOB 暫存器所指定的位元組數目之後，控制器會停止傳送資料。在偵測到 MMC\_I\_REG[DATA\_TRAN\_DONE]中斷之後，軟體必須設定控制器來傳送停止



傳送命令(CMD12)。關於停止傳送命令的說明可參考 *The MultiMediaCard System Specification*。

若在資料傳送期間兩個傳送 FIFO 都變為空的，則 MMC 控制器會關閉時脈。在寫入 FIFO 之後，控制器會啟動時脈。

在寫入資料區塊中，必須指定以下參數：

- 資料傳送為寫入。
- 區塊長度（若區塊長度不同於來自之前區塊資料傳輸，或是這是第一次指定區塊長度）。
- 傳送的區塊數目。

### 9.3.2.2 區塊資料讀取

在單一區塊資料讀取中，會從卡片讀取資料區塊。在多重區塊讀取中，控制器會在 MMC 匯流排上執行多重單一區塊讀取資料之傳輸。

在啟動時脈來啟動指令程序之後，軟體必須程式化 DMA 以清空 MMC\_RXFIFO（讀取 32 個位元組）。軟體會繼續清空 FIFO，直到已從 FIFO 讀取所有的資料。然後軟體必須藉由等待 MMC\_I\_REG[DATA\_TRAN\_DONE] 中斷來等待完成傳送。然後軟體可讀取狀態暫存器 MMC\_STAT 來驗證交易狀態。

對於多重區塊讀取，*The MultiMediaCard System Specification* 詳細說明卡片會繼續傳送資料區塊，直到接收到停止傳送的命令。在控制器已接收到 MMC\_NOB 暫存器內指定的位元組數目之後，控制器會停止接收資料。當偵測到 MMC\_I\_REG[DATA\_TRAN\_DONE] 中斷時，軟體必須建立控制器來傳送停止傳輸命令 CMD12。關於停止傳送命令的說明請參考 *The MultiMediaCard System Specification*。

在資料傳輸期間，若兩個接收 FIFO 都滿載了，則控制器會關閉時脈。一旦軟體清空所連結的 FIFO，控制器會再次啟動時脈。

在區塊資料讀取中，必須指定以下為參數：

- 資料傳輸為讀取。
- 區塊長度（若區塊長度不同於來自之前區塊資料傳輸，或是這是第一次指定區塊長度）。
- 傳送區塊的數目。

- 接收資料逾時期間。

在逾時期間之前若無接收資料，則控制器會將資料傳輸標記為逾時。逾時期間的延遲定義為：

$$\text{Time-out Delay} = \frac{(\text{MMC\_RDTO}[\text{READ\_TO}]) \times (128)}{10^7} \text{sec}$$

軟體需要計算此數值，並將適當的值寫入 MMC\_RDTO 暫存器。

### 9.3.2.3 串流資料寫入

串流資料寫入類似單一區塊寫入，除了停止傳輸命令是在資料最後 6 個位元組平行傳送。

啟動時脈來開始指令程序之後，軟體必須開始填滿 MMC\_TXFIFO 的程序，並如 9.3.2.1「區塊資料寫入」

所描述來啟動時脈。然後軟體必須等待 MMC\_I\_REG[STOP\_CMD]中斷。此中斷指示 MMC 控制器已準備好停止傳送指令。然後軟體必須停止時脈、為停止傳送指令寫入暫存器，然後啟動時脈。此時軟體必須等待 MMC\_I\_REG[DATA\_TRAN\_DONE]和 MMC\_I\_REG[PRG\_DONE]中斷。

在串流資料寫入中，必須指定以下參數：

- 資料傳送為寫入。
- 資料傳送為串流模式。
- 區塊長度（若區塊長度不同於來自之前區塊資料傳輸，或是這是第一次指定區塊長度）。
- 傳送區塊的數目，如 0xffff。

### 9.3.2.4 串流資料讀取

串流資料讀取如同單一區塊讀取，除了在資料傳送之後必須傳送一個停止傳送命令之外。

在開啟時脈以啟動命令程序之後，軟體必須啟動如 15.3.2.2 節「區塊資料讀取」所描述的 MMC\_RXFIFO 之讀取處理。

當使用 DMA 時，在讀取所有的資料之後，軟體也必須設定 DMA 組態來傳送一個中斷。在 DMA 中斷或程式讀取所有資料之後，軟體必須傳送停止傳送命令。直到軟體傳送停止傳送命令，否則不會設定 MMC\_STAT[DATA\_TRAN\_DONE]位元。

在串流資料讀取中，必須指定以下參數：

- 資料傳送為讀取。
- 資料傳送為串流模式。
- 區塊長度（若區塊長度不同於來自之前區塊資料傳輸，或是這是第一次指定區塊長度）。
- 傳送區塊的數目，如 0xffff。
- 接收資料逾時期間。

### 9.3.3 忙碌程序

在單一與多重區塊寫入操作的每個資料區塊之後，MMC 控制器會自動預期一個來自卡片的忙碌訊號。每次軟體指示預期忙碌訊號（例如：在停止傳送命令、卡片選擇、清除、程式 CID...等等之後會預期忙碌訊號），在命令的最後，MMC 控制器也會預期忙碌訊號。請參考 *The MultiMediaCard System Specification*。

當 MMC 匯流排上有忙碌訊號，軟體只可傳送以下兩個命令其中之一：

- 傳送狀態命令（CMD13）。
- 切斷連結命令（CMD7）。

在卡片為忙碌狀態時，若軟體切斷與卡片的連結，則會關閉忙碌訊號，而軟體可連結另一個不同的卡片。軟體可能不會在相同的卡片上啟動另一個命令程序（當此卡片為忙碌時）。

### 9.3.4 SPI 功能

MMC 控制器最多可將兩個卡片定址為 SPI 模式，使用 MMCCS[1:0]晶片選擇訊號。一旦軟體指定一個晶片選擇在 MMC\_SPI 暫存器內啟動，選擇的訊號在 MMC 時脈的下降邊緣會驅動為主動低電位。晶片選擇維持觸發狀態直到軟體清除晶片選擇啟動位元或選擇另一個卡片。

注意：在寫入至任何暫存器（如 15.3.1 節「基本、無資料、命令與回應程序」中所述）之前，必須停止時脈。

在 SPI 模式中，軟體有執行 CRC 確認的選擇。預設值為無 CRC 確認。

命令與資料在 MMC 匯流排上傳送，並對齊每 8 個時脈，如 *The MultiMediaCard System Specification* 中的 SPI 部分所述。

在讀取程序中，卡片可能傳回資料或資料錯誤符號。若接收到資料錯誤符號，控制器會停止傳送並更新狀態暫存器。

## 9.4 多媒體卡控制器操作

軟體指示卡片與控制器之間所有溝通的方向。先前章節中的操作只對 MMC 模式有效。

### 9.4.1 啟動與停止時脈

藉由停止時脈、寫入暫存器，與啟動時脈來存取暫存器集。

軟體停止時脈如下述：

1. 將 0x01 寫入 MMC\_STRPCL 來停止 MMC 時脈。
2. 將 0x0f 寫入 MMC\_I\_MASK 來遮蔽所有的中斷，除了 MMC\_I\_REG[CLK\_IS\_OFF]中斷。
3. 等待 MMC\_I\_REG[CLK\_IS\_OFF]中斷。

軟體寫入 0x02 至 MMC\_STRPCL 來啟動時脈。

### 9.4.2 初始化

卡片初始化程序必須在前面加上 80 的時脈週期。為了在任何命令之前產生 80 個時脈週期，軟體必須設定 MMC\_CMDAT[INIT]位元。

### 9.4.3 啟動 SPI 模式

為了以 SPI 模式與卡片溝通，軟體必須依照下面來設定 MMC\_SPI 暫存器：

1. MMC\_SPI[SPI\_EN]必須設定為 1。
2. MMC\_SPI[SPI\_CS\_EN]必須設定為 1。
3. 必須設定 MMC\_SPI[SPI\_CS\_ADDRESS]來指定軟體欲定址的卡片。為 1 時會啟動 CS0，而為 0 時會啟動 CS1。

注意：當卡片為 SPI 模式時，回復為 MMC 模式的唯一方法為開啟卡片電源。

### 9.4.4 無資料命令與回應程序

對於基本無資料傳輸、命令，與回應交易，軟體必須：

1. 關閉時脈，如 9.4.1「啟動與停止時脈」所述。
2. 寫入命令索引至 MMC\_CMD[CMD\_INDEX]位元。
3. 寫入命令參數至 MMC\_ARGH 與 MMC\_ARGL 暫存器。
4. 依照下述來寫入 MMC\_CMDAT 暫存器集：
  - a. 寫入 0b00 至 MMC\_CMDAT[RESPONSE\_FORMAT]。
  - b. 清除 MMC\_CMDAT[DATA\_EN]位元。
  - c. 除非卡片可能回應忙碌，否則清除 MMC\_CMDAT[BUSY]位元。
  - d. 清除 MMC\_CMDAT[INIT]位元。
5. 以適當的數值寫入 MMC\_RESTO 暫存器。
6. 寫入 0x1b 至 MMC\_I\_MASK 來反遮蔽 MMC\_I\_REG[END\_CMD\_RES]中斷。
7. 啟動時脈，如 9.4.1 節「啟動與停止時脈」所述。
- 8.

軟體不可改變暫存器內的設定，直到在開啟時脈之後命令與回應程序的最後。開啟時脈之後，軟體必須等待 MMC\_I\_REG[END\_CMD\_RES]中斷，這表示命令與回應程序已結束，而回應在 MMC\_RES FIFO 內。而後軟體可能讀取 MMC\_STAT 暫存器來驗證交易狀態，然後讀取 MMC\_RES FIFO。若發生回應逾時，則 MMC\_RES FIFO 不會包含任何有效的資料。

### 9.4.5 抹除

抹除命令如 9.4.4 節「無資料命令與回應程序」所述一般來執行，並加入下列附加條件：在讀取 MMC\_RES\_FIFO 之後，必須將 MMC\_CMDAT 暫存器內的 BUSY\_BIT 位元設定為 1。

### 9.4.6 單一資料區塊寫入

在單一區塊寫入命令中，軟體必須停止時脈，並如 9.4.4 節「無資料命令與回應程序」所述來設定暫存器。在啟動時脈之前必須設定下列暫存器：

- 設定 MMC\_NOB 暫存器為 0x0001。
- 設定 MMC\_BLKLEN 為每個時脈的位元組數目。
- 更新 MMC\_CMDAT 暫存器如下：
  - 將 0x01 寫入至 MMC\_CMDAT[RESPONSE\_FORMAT]
  - 設定 MMC\_CMDAT[DATA\_EN]位元。
  - 設定 MMC\_CMDAT[WRITE/READ]位元。
  - 清除 MMC\_CMDAT[STREAM\_BLOCK]位元。
  - 清除 MMC\_CMDAT[BUSY]位元。
  - 清除 MMC\_CMDAT[INIT]位元。
- 啟動時脈。

在啟動時脈之後，軟體必須執行下列步驟：

1. 如 9.4.4 節「無資料命令與回應程序」來等待回應。
2. 寫入資料至 MMC\_TXFIFO FIFO 直到寫入所有資料至 FIFO。  
注意：若有一部份小於 32 位元組資料被寫入 FIFO，則必須設定 MMC\_PRTBUF 暫存器。
3. 設定 MMC\_I\_MASK 暫存器為 0x1e 並等待 MMC\_I\_REG[DATA\_TRAN\_DONE]中斷。
4. 設定 MMC\_I\_MASK 為 0x1d。
5. 等待 MMC\_I\_REG[PRG\_DONE]中斷。此中斷表示卡片已結束程式化。軟體可能等待 MMC\_I\_REG[PRG\_DONE]或啟動不同卡片上的另一個命令程序。
6. 讀取 MMC\_STAT 暫存器來驗證交易狀態（例如：CRC 錯誤狀態）。

為了定址不同的卡片，軟體藉由傳送一個基本的無資料命令與回應交易來

傳送一個選擇命令至卡片。為了定址相同的卡片，軟體必須等待 MMC\_I\_REG[PRG\_DONE]中斷。這可確定卡片不在忙碌狀態。

### 9.4.7 單一區塊讀取

在單一區塊讀取命令中，軟體必須停止時脈並如 9.4.4 節「無資料命令與回應程序」中來設定暫存器。

在啟動時脈之前必須先設定下列暫存器：

- 更新 MMC\_CMDAT 暫存器的下列位元：
  - 設定 MMC\_CMDAT[RESPONSE\_FORMAT]位元。
  - 設定 MMC\_CMDAT[DATA\_EN]位元。
  - 清除 MMC\_CMDAT[WRITE/READ]位元。
  - 清除 MMC\_CMDAT[STREAM\_BLOCK]位元。
  - 清除 MMC\_CMDAT[BUSY]位元。
  - 清除 MMC\_CMDAT[INIT]位元。
- 將 MMC\_NOB 暫存器設定為 0x0001。
- 將 MMC\_BLKLEN 暫存器設定為每個區塊的位元組數目。
- 開啟時脈。

開啟時脈之後，軟體必須執行以下步驟：

1. 等待回應，如同 9.4.4 節「無資料命令與回應程序」所述。
2. 當 FIFO 內的資料可使用時，從 MMC\_RXFIFO FIFO 讀取資料，一直讀取直到讀取 FIFO 所有資料。
3. 設定 MMC\_I\_MASK 為 0x1e。
4. 等待 MMC\_I\_REG[DATA\_TRAN\_DONE]中斷。
5. 讀取 MMC\_STAT 暫存器來驗證交易狀態（例如：CRC 錯誤狀態）。

### 9.4.8 多重區塊寫入

多重區塊寫入模式類似單一區塊寫入模式，除了是傳送多重資料區塊之外。每一個區塊長度都相同。除了 MMC\_NOB 暫存器設定為寫入的區塊數目之外，所有暫存器都會被設定來做單一區塊寫入。

多重區塊寫入模式中，在資料傳送至卡片之後，也需要停止傳送命令 CMD12。在發生 MMC\_I\_REG[DATA\_TRAN\_DONE]中斷之後，軟體必須程式化中斷器暫存器來傳送中斷資料傳送命令。

### 9.4.9 多重區塊讀取

多重區塊讀取模式類似單一區塊讀取模式，除了是傳送多重資料區塊之外。每一個區塊長度都相同。除了 MMC\_NOB 暫存器設定為讀取的區塊數目之外，所有暫存器都會被設定來做單一區塊讀取。

多重區塊讀取模式中，在接收卡片資料之後，也需要停止傳送命令 CMD12。在發生 MMC\_I\_REG[DATA\_TRAN\_DONE]中斷之後，軟體必須程式化中斷器暫存器來傳送中斷資料傳送命令。

### 9.4.10 串流寫入

在串流寫入命令中，軟體必須停止時脈，並且設定暫存器，如 9.4.4 節「無資料命令與回應程序」所述。在啟動時脈之前必須設定下列暫存器：

- 將 MMC\_NOB 設定為 ffffh。
- 將 MMC\_BLKLEN 暫存器設定為每個區塊的位元組數目。
- 更新 MMC\_CMDAT 暫存器如下：
  - 將 0b01 寫入至 MMC\_CMDAT[RESPONSE\_FORMAT]。
  - 設定 MMC\_CMDAT[DATA\_EN]位元。
  - 設定 MMC\_CMDAT[WRITE/READ]位元。
  - 設定 MMC\_CMDAT[STREAM\_BLOCK]位元。
  - 清除 MMC\_CMDAT[BUSY]位元。
  - 清除 MMC\_CMDAT[INIT]位元。
- 開啟時脈。

開啟時脈之後，軟體必須執行下列步驟：

1. 等待回應，如 9.4.4 節「無資料命令與回應程序」所述。
2. 持續寫入資料至 MMC\_TXFIFO FIFO 直到將所有資料寫入至 FIFO 內。

注意：當寫入的資料少於 32 個位元組時，必須設定 MMC\_PRTBUF[BUF\_PART\_FULL]位元。



3. 將 MMC\_I\_MASK 設定為 0x77，並等待 MMC\_I\_REG[STOP\_CMD]中斷。
4. 為停止交易命令（CMD12）設定命令暫存器。
5. 等待回應至停止交易命令，如 9.4.4 節「無資料命令與回應程序」所述。
6. 設定 MMC\_I\_MASK 為 0x1e。
7. 等待 MMC\_I\_REG[DATA\_TRAN\_DONE]中斷。
8. 設定 MMC\_I\_MASK 為 0x1d。
9. 等待 MMC\_I\_REG[PRG\_DONE]中斷。此中斷表示卡片已結束程式化。軟體可能等待 MMC\_I\_REG[PRG\_DONE]中斷，或啟動不同卡片上的另一個命令程序。
10. 讀取 MMC\_STAT 暫存器來驗證交易狀態（例如：CRC 錯誤狀態）。

為了定址不同的卡片，軟體必須藉由傳送一個基本的無資料命令與回應交易來傳送一個選擇命令至卡片。為了定址相同的卡片，軟體必須等待 MMC\_I\_REG[PRG\_DONE]中斷。可確定卡片不為忙碌狀態。

### 9.4.11 串流讀取

在串流讀取命令中，軟體必須停止時脈並設定暫存器，如 9.4.4 節「無資料命令與回應程序」所述。在啟動時脈之前必須先設定下列暫存器：

- 將 MMC\_NOB 暫存器設定為 ffffh。
- 將 MMC\_BLKLEN 暫存器設定為每個區塊的位元組數目。
- 更新 MMC\_CMDAT 暫存器如下：
  - 將 0x01 寫入至 MMC\_CMDAT[RESPONSE\_FORMAT]。
  - 設定 MMC\_CMDAT[DATA\_EN]位元。
  - 清除 MMC\_CMDAT[WRITE/READ]位元。
  - 設定 MMC\_CMDAT[STREAM\_BLOCK]位元。
  - 清除 MMC\_CMDAT[BUSY]位元。
  - 清除 MMC\_CMDAT[INIT]位元。
- 開啟時脈。

開啟時脈之後，軟體必須執行下列步驟：

1. 等待回應，如 9.4.4 節「無資料命令與回應程序」所述。
2. 持續寫入資料至 MMC\_TXFIFO FIFO 直到將所有資料寫入至 FIFO 內。
3. 為停止交易命令（CMD12）設定命令暫存器。若 DMA 正被使用，則最後一個敘述元必須設定 DMA 傳送中斷來發出讀取所有資料的訊號。
4. 等待回應至停止交易命令，如 9.4.4 節「無資料命令與回應程序」所述。

5. 將 MMC\_I\_MASK 設定為 0x1e。
6. 等待 MMC\_I\_REG[DATA\_TRAN\_DONE] 中斷。
7. 讀取 MMC\_STAT 暫存器來驗證交易狀態（例如：CRC 錯誤狀態）。

## 9.5 MMC 控制暫存器

MMC 控制器由一套暫存器所控制，在 MMC 匯流排上的每個指令順序之前軟體會組態這些暫存器。

表 9-6 列出 MMC 控制暫存器的位址、名稱和描述。表 9-6 至表 9-24 描述暫存器和 FIFO。

**表 9-5 MMC 控制暫存器**

位址	名稱	描述
0x4110 0000	MMC_TRPCL	控制 MMC 時脈啟動或停止。
0x4110 0004	MMC_STAT	MMC 狀態暫存器（唯讀）。
0x4110 0008	MMC_CLKRT	MMC 時脈率。
0x4110 000c	MMC_SPI	SPI 模式控制位元。
0x4110 0010	MMC_CMDAT	指令／回應／資料順序控制。
0x4110 0014	MMC_RESTO	預期回應逾時。
0x4110 0018	MMC_RDTO	預期資料讀取逾時。
0x4110 001c	MMC_BLKLEN	資料交易的區塊長度。
0x4110 0020	MMC_NOB	區塊模式，區塊的數目。
0x4110 0024	MMC_PRTBUF	局部 MMC_TXFIFO FIFO 寫入。
0x4110 0028	MMC_I_MASK	中斷遮罩。
0x4110 002c	MMC_I_REG	中斷暫存器（唯讀）。
0x4110 0030	MMC_CMD	目前指令索引。
0x4110 0034	MMC_ARGH	目前指令參數的 MSW 部分。
0x4110 0038	MMC_ARGL	目前指令參數的 LSW 部分。
0x4110 003c	MMC_RES	回應 FIFO（唯讀）。
0x4110 0040	MMC_RXFIFO	接收 FIFO（唯讀）。
0x4110 0044	MMC_TXFIFO	傳送 FIFO（唯寫）。

### 9.5.5 MMC\_STRPCL 暫存器

MMC\_STRPCL 允許軟體控制 MMC 匯流排時脈的啟動和停止。此暫存器是

唯寫的，不可讀取。表 9-6 描述暫存器的位元。

表 9-6 MMC\_STRPCL 暫存器

實體位址 4110_0000		MMC_STRPCL 暫存器																MMC															
位元	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	<div>STRPCL</div>																																
重置	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
位元	名稱		描述																														
31:2	保留		保留。																														
1:0	Strpcl		啟動／停止 MMC 時脈 00 = 無動作 01 = 停止 MMC 時脈 10 = 啟動 MMC 時脈 11 = 保留																														

9.5.2 MMC\_STAT 暫存器

MMC\_STAT 暫存器為 MMC 控制器的狀態暫存器。在每個指令順序的開始，暫存器會被清除。

表 9-7 MMC\_STAT 暫存器

實體位址 4110_0004		MMC_STAT 暫存器																MMC																										
位元	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
	保留																		END_CMD_RES		PRG_DONE		DATA_TRAN_DONE		保留		CLK_EN		RECV_FIFO_FULL		XMIT_FIFO_EMPTY		RES_CRC_ERR		SPI_READ_ERROR_TOKEN		CRC_READ_ERROR		CRC_WRITE_ERROR		TIME_OUT_RESPONSE		READ_TIME_OUT	
重置	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0											

位元	名稱	描述
31:14	保留	保留。
13	END_CMD_RE S	指令順序尾端 1 = 指令和回應順序已經完成。 0 = 指令和回應順序還沒完成。

12	PRG_DONE	程式完成 1 = 卡片已經結束程式化且不忙碌。 0 = 卡片還沒結束程式化且在忙碌。
11	DATA_TRAN_DONE	資料傳送完成 1 = 傳送資料到卡片已經完成。 0 = 傳送資料到卡片還沒完成。
10:9	保留	保留。
8	CLK_EN	啟動時脈 1 = MMC 時脈啟動。 0 = MMC 時脈關閉。
7	RECV_FIFO_FULL	接收 FIFO 滿載 1 = 接收 FIFO 滿載。 0 = 接收 FIFO 未滿載。
6	XMIT_FIFO_EMPTY	空的傳送 FIFO 1 = 傳送 FIFO 為空的。 0 = 傳送 FIFO 不是空的。
5	RES_CRC_ERR	回應 CRC 錯誤 1 = 回應發生 CRC 錯誤。 0 = 回應 CRC 無錯誤。
4	SPI_READ_ERROR_TOKEN	SPI 讀取錯誤符號 1 = SPI 資料錯誤符號已經被接收。 0 = SPI 資料錯誤符號還未被接收。
3	CRC_READ_ERROR	CRC 讀取錯誤 0 = 接收的資料發生 CRC 錯誤。 1 = 接收的資料無錯誤。
2	CRC_WRITE_ERROR	CRC 寫入錯誤 1 = 卡片觀察到資料的錯誤傳送。 0 = 資料傳送無錯誤。
1	TIME_OUT_RESPONSE	逾時回應 1 = 卡片回應逾時。 0 = 卡片回應無逾時。
0	READ_TIME_OUT	讀取逾時 1 = 卡片讀取資料未逾時。 0 = 卡片讀取資料逾時。

9.5.3 MMC\_CLKRT 暫存器

MMC\_CLKRT 暫存器說明 MMC 匯流排時脈的頻率段。軟體負責設定此暫存器。

在時脈關閉和軟體已經接收到指示時脈關閉的中斷之後，軟體只可以寫入此暫存器。

表 9-8 MMC\_CLK 暫存器

實體位址 4110_0008		MMC_CLKRT 暫存器																MMC																
位元	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	保留																														CLK_RATE[2:0]			
重置	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
位元		名稱		描述																														
31:3		保留		保留。																														
2:0		CLK_RATE[2:0]		000 = 20 MHz 時脈 001 = 10 MHz 時脈，20 MHz 時脈的 1/2 010 = 5 MHz 時脈，20 MHz 時脈的 1/4 011 = 2.5 MHz 時脈，20 MHz 時脈的 1/8 100 = 1.25 MHz 時脈，20 MHz 時脈的 1/16 101 = 0.625 MHz 時脈，20 MHz 時脈的 1/32 110 = 0.3125 MHz 時脈，20 MHz 時脈的 1/64 111 = 保留																														

9.5.4 MMC\_SPI 暫存器

MMC\_SPI 暫存器只可為 SPI 模式所使用，需藉由軟體來設定。

### 表 9-9 MMC\_SPI 暫存器

實體位址  
4110\_000c

MMC\_SPI 暫存器

MMC

位元

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

保留

SPI\_CS\_ADDRESS  
SPI\_CS\_EN  
CRC\_ON  
SPI\_EN

重置

0 0

位元	名稱	說明
31:4	保留	保留
3	SPI_CS_ADDRESS	指定相關的卡片位址來啟動 SPI CS 1 = 啟動 CS0 0 = 啟動 CS1
2	SPI_CS_EN	SPI 晶片選擇啟動 0 = 啟動 SPI 晶片選擇 1 = 關閉 SPI 晶片選擇
1	CRC_ON	CRC 產生啟動 1 = 啟動 CRC 產生與驗證 0 = 關閉 CRC 產生與驗證
0	SPI_EN	SPI 模式啟動 1 = 啟動 SPI 模式 0 = 關閉 SPI 模式

### 9.5.5 MMC\_CMDAT 暫存器

MMC\_CMDAT 暫存器控制命令程序。當 MMC 匯流排時脈啟動時，寫入至此暫存器會啟動 MMC 匯流排上的命令程序。

**表 9-10 MMC\_CMDAT 暫存器**

實體位址		MMC_CMDAT 暫存器																								MMC							
4110_0010		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
位元		保留																								MMC_DMA_EN	INIT	BUSY	STREAM_BLOCK	WRITE/READ	DATA_EN	RESPONSE_FORMAT[1:0]	
重置		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

位元	名稱	說明
31:8	保留	保留
7	MMC_DMA_EN	DMA 模式啟動 1 = DMA 模式 0 = 程式 I/O 模式 當使用 DMA 模式時，此位元會遮蔽 RXFIFO_RD_REQ 與 TXFIFO_WR_REQ 中斷。
6	INIT	80 個初始化時脈 1 = 在命令程序之前加上 80 個時脈 0 = 在命令程序之前不加上 80 個時脈
5	BUSY	指定在目前命令之後是否預期忙碌訊號 此位元只可為無資料命令／回應交易所使用
4	STREAM_BLOCK	串流模式 1 = 目前命令程序的資料傳送為串流模式 0 = 目前命令程序的資料傳送不為串流模式
3	WRITE/READ	讀取或寫入操作 1 = 指定目前命令的資料傳送為寫入操作 0 = 指定目前命令的資料傳送為讀取操作
2	DATA_EN	資料傳送啟動 1 = 指定目前命令包含資料傳送 0 = 目前命令無資料傳送
1:0	RESPONSE_FORMAT[1:0]	這些位元指定目前命令的回應格式。 00 = MMC 模式中無回應。SPI 模式不支援 01 = MMC 模式中為 R1、R1b、R4 與 R5 格式。SPI 模式中為 R1 與 R1b 格式 10 = MMC 模式中為 R2 格式。SPI 模式中為 R2 格式 11 = MMC 模式中為 R3 格式。SPI 模式中為 R3 格式

### 9.5.6 MMC\_RESTO 暫存器

MMC\_RESTO 暫存器控制 MMC 時脈的數目。若無回應發生，則控制器在命令之後與開啟逾時錯誤之前必須等待。此暫存器的預設值為 64。

表 9-11 MMC\_RESTO 暫存器

實體位址 4110_0014										MMC_RESTO 暫存器										MMC												
位元	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留																								RES_TO							
重置	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
位元	名稱				描述																											
31:7	保留				保留																											
6:0	RES_TO				回應逾時前，MMC 時脈的數目。																											

9.5.7 MMC\_RDTO 暫存器

MMC\_RDTO 暫存器決定時間的長度。若資料未被接收，則控制器在命令之後與開啟逾時錯誤之前必須等待。發生逾時之前的時間長度估計為 256 20MHz 期間的增量，可用以下方程式來計算：

Time-out Delay =  $\frac{(\text{MMC\_RDTO}[\text{READ\_TO}]) \times (256)}{20\text{MHz}} = \frac{(\text{MMC\_RDTO}[\text{READ\_TO}]) \times (128)}{10^7} \text{sec}$

軟體負責計算 READ\_TO 的值。預設值為 0xffff，與 838.848ms 一致。

表 9-12 MMC\_RDTO 暫存器

實體位址 4110_0018																MMC_RDTO 暫存器																MMC															
位元	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
	保留																READ_TO																														
重置	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1															
位元	名稱																描述																														
31:16	保留																保留																														
15:0	READ_TO																說明資料讀取逾時前時間的長度。																														

9.5.8 MMC\_BLKLEN 暫存器

MMC\_BLKLEN 說明資料區塊的位元組數目。

表 9-13 MMC\_BLKLEN 暫存器



實體位址 4110_001c										MMC_BLKLEN暫存器										MMC																		
位元	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
	保留																						BLK_LEN															
重置	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							

位元	名稱	描述
31:10	保留	保留
9:0	BLK_LEN	區塊的位元組數目。

9.5.9 MMC\_NOB 暫存器

在區塊模式，MMC\_NOB 暫存器說明區塊的數目。

表 9-14 MMC\_NOB 暫存器

實體位址 4110_0020																MMC_NOB 暫存器																MMC															
位元	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
	保留																MMC_NOB																														
重置	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																
位元	名稱																描述																														
31:16	保留																保留																														
15:0	MMC_NOB																多重區塊傳輸的區塊數目。																														

9.5.10 MMC\_PRTBUF 暫存器

當 MMC\_TXFIFO 被部分寫入時，使用 MMC\_PRTBUF 暫存器。當 FIFO 滿載（32 位元組）或 MMC\_PRTBUF 暫存器被設定為 1 時，FIFO 替換。

表 9-15 MMC\_PRTBUF 暫存器

實體位址 4110_0024		MMC_PRTBUF 暫存器																MMC														
位元	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留																															BUF_PART_FULL
重置	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位元	名稱	描述
31:1	保留	保留
0	BUF_PART_FULL	緩衝區部分滿載 1 = 緩衝區部分滿載，且必須被替換到其他傳輸緩衝區。 0 = 緩衝區無部分滿載。 在傳送下一個命令前，軟體必須清除此位元。

### 9.5.11 MMC\_I\_MASK 暫存器

在設定為 1 時，MMC\_I\_MASK 暫存器會遮蔽各種中斷。

**表 9-16 MMC\_I\_MASK 暫存器**

實體位址 4110_0028										MMC_I_MASK 暫存器																		MMC							
位元	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
重置	保留																											TXFIFO_WR_REQ	RXFIFO_RD_REQ	CLK_IS_OFF	STOP_CMD	END_CMD_RES	PRG_DONE	DATA_TRAN_DONE	

位元	名稱	描述
31:7	保留	保留
6	TXFIFO_W R_REQ	傳送 FIFO 寫入請求 1 = 遮蔽。 0 = 無遮蔽。
5	RXFIFO_R D_REQ	接收 FIFO 讀取請求 1 = 遮蔽。 0 = 無遮蔽。
4	CLK_IS_O FF	時脈關閉 1 = 遮蔽。 0 = 無遮蔽。
3	STOP_CM D	準備停止傳送命令 1 = 遮蔽。 0 = 無遮蔽。
2	END_CMD _RES	結束命令回應 1 = 遮蔽。

		0 = 無遮蔽。
1	PRG_DON E	程式化完成 1 = 遮蔽。 0 = 無遮蔽。
0	DATA_TRA N_DONE	資料傳輸完成 1 = 遮蔽。 0 = 無遮蔽。

9.5.12 MMC\_I\_REG 暫存器

MMC\_I\_REG 暫存器顯示目前請求的中斷。FIFO 請求中斷（TXFIFO\_WR\_REG）和 RXFIFO\_RD\_REQ 由 MMC\_CMDAT 暫存器的 MMC\_DMA\_EN 位元所遮蔽。軟體負責監視在程式 I/O 模式的這些位元。表 9-9 說明這些位元被清除的描述。

表 9-17 MMC\_I\_REG 暫存器

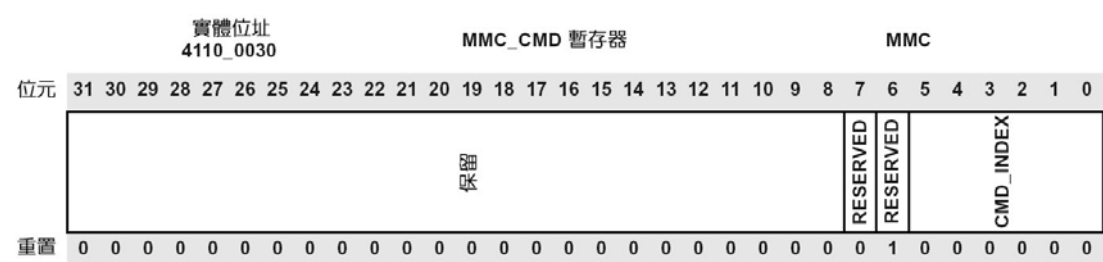
實體位址 4110_002c		MMC_I_REG 暫存器																MMC														
位元	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	圖 9-17																									TXFIFO_WR_REQ	RXFIFO_RD_REQ	CLK_IS_OFF	STOP_CMD	END_CMD_RES	PRG_DONE	DATA_TRAN_DONE
重置	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位元	名稱		描述																													
31:7	保留		保留																													
6	TXFIFO_W R_REQ		傳送 FIFO 寫入請求 1 = 資料寫入 MMC_TXFIFO FIFO 的請求。 0 = 無資料寫入 MMC_TXFIFO FIFO 的請求。 每個寫入後被清除，但是立刻再一次設定，除非 FIFO 內無遺留的通道。																													
5	RXFIFO_R D_REQ		接收 FIFO 讀取請求 1 = 從 MMC_RXFIFO FIFO 讀取資料的請求。 0 = 無從 MMC_RXFIFO FIFO 讀取資料的請求。																													

		每個寫入後被清除，但是立刻再一次設定，除非 FIFO 是空的。
4	CLK_IS_O FF	時脈關閉 1 = MMC 時脈已經被關閉，因為 STRP_CLK 暫存器的停止位元。 0 = MMC 時脈還未被關閉。 當時脈啟動時，被 MMC_STAT[CLK_EN]位元清除。
3	STOP_CM D	串流模式寫入 1 = MMC 準備好停止傳送命令。 0 = MMC 還沒準備好停止傳送命令。 當 CMD12 被載入 MMC_CMD 暫存器且時脈被啟動時，此位元被清除。
2	END_CMD _RES	結束命令回應 1 = MMC 已經接收到回應或是發生回應逾時。 0 = MMC 無接收到回應。 被 MMC_STAT[END_CMD_RES]位元清除。
1	PRG_DON E	程式化完成 1 = 卡片已經結束程式化且不再忙碌。 0 = 卡片未完成程式化且在忙碌。 被 MMC_STAT[PRG_DONE]位元清除。
0	DATA_TRA N_DONE	資料傳輸完成 1 = 資料傳輸已經完成或發生讀取資料逾時。 0 = 資料傳輸未完成。 被 MMC_STAT[DATA_TRAN_DONE]位元清除。

### 9.5.13 MMC\_CMD 暫存器

MMC\_CMD 暫存器說明命令數目。

表 9-18 MMC\_CMD 暫存器



位元	名稱	描述
31:8	保留	保留
7	保留	保留、唯讀、總為 0。命令順序的開始位元且不可以改變。
6	保留	保留、唯讀、總為 1。命令順序的傳送位元且不可以改變。
5:0	CMD_INDEX	命令索引（請參考表 9-19）。

表 9-19 命令索引值

CMD 索引	COMMAND	模式	縮寫
000000	CMD0	MMC/SPI	GO_IDLE_STATE
000001	CMD1	MMC/SPI	SEND_OP_COND
000010	CMD2	MMC	ALL_SEND_CID
000011	CMD3	MMC	SET_RELATIVE_ADDR
000100	CMD4	MMC	SET_DSR
000101	CMD5	保留	
000110	CMD6	保留	
000111	CMD7	MMC	SELECT/DESELECT_CARD
001000	CMD8	保留	
001001	CMD9	MMC/SPI	SEND_CSD
001010	CMD10	MMC/SPI	SEND_CLD
001011	CMD11	MMC	READ_DAT_UNTIL_STOP
001100	CMD12	MMC	STOP_TRANSMISSION
001101	CMD13	MMC/SPI	SEND_STATUS
001110	CMD14	保留	
001111	CMD15	MMC	GO_INACTIVE_STATE
010000	CMD16	MMC/SPI	SET_BLOCKLEN
010001	CMD17	MMC/SPI	READ_SINGLE_BLOCK
010010	CMD18	MMC	READ_MULTIPLE_BLOCK
010011	CMD19	保留	

010100	CMD20	MMC	WRITE_DAT_UNTIL_STOP
010101	CMD21	保留	
010110	CMD22	保留	
010111	CMD23	保留	
011000	CMD24	MMC/SPI	WRITE_BLOCK
011001	CMD25	MMC	WRITE_MULTIPLE_BLOCK
011010	CMD26	MMC	PROGRAM_CID
011011	CMD27	MMC/SPI	PROGRAM_CSD
011100	CMD28	MMC/SPI	SET_WRITE_PROT
011101	CMD29	MMC/SPI	CLR_WRITE_PROT
011110	CMD30	MMC/SPI	SEND_WRITE_PROT
011111	CMD31	保留	
100000	CMD32	MMC/SPI	TAG_SECTOR_START
100001	CMD33	MMC/SPI	TAG_SECTOR_END
100010	CMD34	MMC/SPI	UNTAG_SECTOR
100011	CMD35	MMC/SPI	TAG_ERASE_GROUP_START
100100	CMD36	MMC/SPI	TAG_ERASE_GROUP_END
100101	CMD37	MMC/SPI	UNTAG_ERASE_GROUP
100110	CMD38	MMC/SPI	ERASE
100111	CMD39	MMC	FAST_IO
101000	CMD40	MMC	GO_IRQ_STATE
101001	CMD41	保留	
101010	CMD42	MMC/SPI	LOCK_UNLOCK
101011	CMD43	保留	
101100	CMD44	保留	
101101	CMD45	保留	
101110	CMD46	保留	
101111	CMD47	保留	
110000	CMD48	保留	
110001	CMD49	保留	
110010	CMD50	保留	
110011	CMD51	保留	
110100	CMD52	保留	
110101	CMD53	保留	
110110	CMD54	保留	
110111	CMD55	MMC/SPI	APP_CMD
111000	CMD56	MMC_SPI	GEN_CMD

111001	CMD57	保留	
111010	CMD58	SPI	READ_OCR
111011	CMD59	SPI	CRC_ON_OFF
111100	CMD60	MMC	保留給廠商
111101	CMD61	MMC	保留給廠商
111110	CMD62	MMC	保留給廠商
111111	CMD63	MMC	保留給廠商

9.5.14 MMC\_ARGH 暫存器

MMC\_ARGH 暫存器說明目前命令的參數的上層 16 位元。

表 9-20 MMC\_ARGH 暫存器

實體位址 4110_0034		MMC_ARGH 暫存器																MMC															
位元	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留																ARG_H																
重置	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
位元	名稱																描述																
31:16	保留																保留																
15:0	ARG_H																命令參數的上層 16 位元。																

9.5.15 MMC\_ARGL 暫存器

MMC\_ARGL 暫存器說明目前命令的參數的下層 16 位元。

表 9-21 MMC\_ARGL 暫存器

實體位址 4110_0038		MMC_ARGL 暫存器																MMC															
位元	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留																ARG_L																
重置	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
位元	名稱																描述																
31:16	保留																保留																
15:0	ARG_L																命令參數的下層 16 位元。																

9.5.16 MMC\_RES FIFO（唯讀）

命令之後，MMC\_RES FIFO 包含回應。它為 8 個通道長、16 位元寬。RES FIFO 不包含回應的 7 位元 CRC。CRC 檢查的狀態和回應逾時狀態在 MMC\_STAT 狀態暫存器內。

從回應 FIFO 的第一個半字組讀取為接收回應的最重要的半字組。

表 9-22 MMC\_RES，FIFO 通道

實體位址 4110_003c																MMC_RES FIFO通道																MMC															
位元	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
	保留																RESPONSE_DATA																														
重置	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x															
位元	名稱																描述																														
31:16	保留																保留																														
15:0	RESPONSE_DATA																回應資料的二個位元組。																														

9.5.17 MMC\_RXFIFO FIFO（唯讀）

MMC\_RXFIFO 由兩個雙重 FIFO 所組成，而每個 FIFO 有 32 通道長、8 位元寬。此 FIFO 有從卡片讀取的資料。它對軟體而言是唯讀的 FIFO，且讀取邊界為 8 位元。資料的 8 位元在 32 位元邊界被讀取且佔用最不重要的位元組通道（7:0）。



表 9-23 MMC\_RXFIFO，FIFO 通道

實體位址 4110_0040																MMC_RXFIFO 通道																MMC															
位元	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
	保留																								READ_DATA																						
重置	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x															

位元	名稱	描述
31:8	保留	保留
7:0	READ_DATA	讀取資料的一個位元組。

### 9.5.18 MMC\_TXFIFO FIFO

MMC\_TXFIFO 由兩個雙重 FIFO 所組成，而每個 FIFO 有 32 通道長、8 位元寬。此 FIFO 有寫入卡片的資料。它對軟體而言是唯寫的 FIFO，且寫入邊界為 8 位元寬。資料的 8 位元在 32 位元 APB 被寫入且佔用最不重要的位元組通道 (7:0)。

表 9-24 MMC\_TXFIFO，FIFO 通道

實體位址 4110_0044																MMC_TXFIFO 通道																MMC															
位元	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
	保留																								WRITE_DATA																						
重置	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x															

位元	名稱	描述
31:16	保留	保留
7:0	WRITE_DATA	寫入資料的一個位元組。

### 問題：

1. 請問何為多媒體卡(MMC)？有何特點？
2. 請問在 Intel XScale 應用處理器中的多媒體卡支援哪些模式？
3. 請問在 Intel XScale 應用處理器中的 MMC 有哪些特性？
4. 請問在 Intel XScale 應用處理器中的 MMC，其 SPI 模式與 MMC 模式的資料符號有何不同？
5. 請問在 Intel XScale 應用處理器中的 MMC，其 SPI 模式與 MMC 模式的運作方式有何不同？
6. 請問如何初始化 MMC 卡片？
7. 請說明何為 MMC 模式？

8. 請說明何為 SPI 模式？
9. 請問為何要控制 MMC 匯流排的時脈？如何控制 MMC 匯流排的時脈？
10. 請問 MMC 卡如何接收資料？
11. 請問使用 DMA 來做多媒體卡的讀取時，若要讀取 105 個位元組，其程序為何？
12. 請問使用 DMA 來做多媒體卡的寫入時，若要寫入 98 個位元組，其程序為何？
13. 請問多媒體卡在基本，無資料傳輸時的命令與回應的程序為何？
14. 請問多媒體卡的傳送資料程序為何？