

### 第三章 系統設計

當開始發展單晶片系統時，都會選定某個特定晶片上匯流排 (on-chip bus) 系統，讓大家有一相同的匯流排介面方便於整合，但在選用適合的智產模組後到能和匯流排上的其他模組溝通仍有相當的距離。為了配合相同的匯流排介面，我們需要設計一個轉換器(wrapper)，提供兩個不同介面的轉換[6]。如圖 3.1 所示，所設計的轉換器必須將匯流排與智產模組間的信號與傳輸協定做一個適當的轉換。因此在利用 AMBA 整合各個智產時，這些智產的介面必須滿足 AMBA 協定，所以需要對於每個智產發展出一個適當的轉換器。

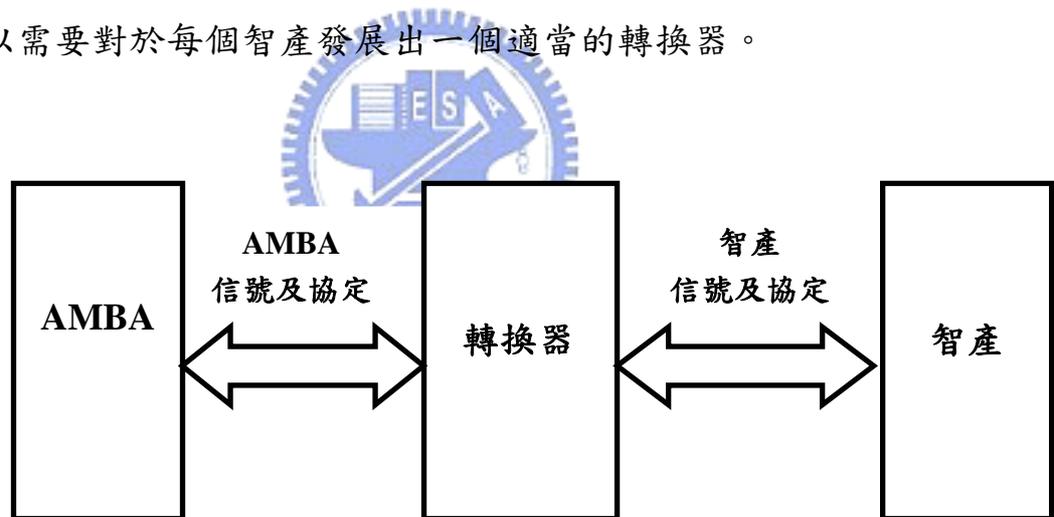


圖 3.1 轉換器連接情形

轉換器的設計主要分成下列 3 個步驟[7]:

1. 確定匯流排協定的行為。
2. 發展匯流排規則與智產行為間的相關性。

3. 建立狀態圖：根據所需，建立適當的狀態機。

這個章節主要根據此步驟設計各個智產轉換器來連接 AMBA 匯流排。由於前一章已經對匯流排協定作了說明，所以本章節將不再對此加以討論。本章節除了設計微控制器及數位信號處理器的轉換器之外，另外還包含了直接記憶體存取(Direct Memory Access, DMA)、同步動態記憶體控制器(SDRAM controller)及 AMBA 架構內部基本元件：仲裁器(Arbiter)，解碼器(Decoder)及橋接器(Bridge)，以完成基本的單晶片系統。圖 3.2 顯示本論文整個系統的架構。

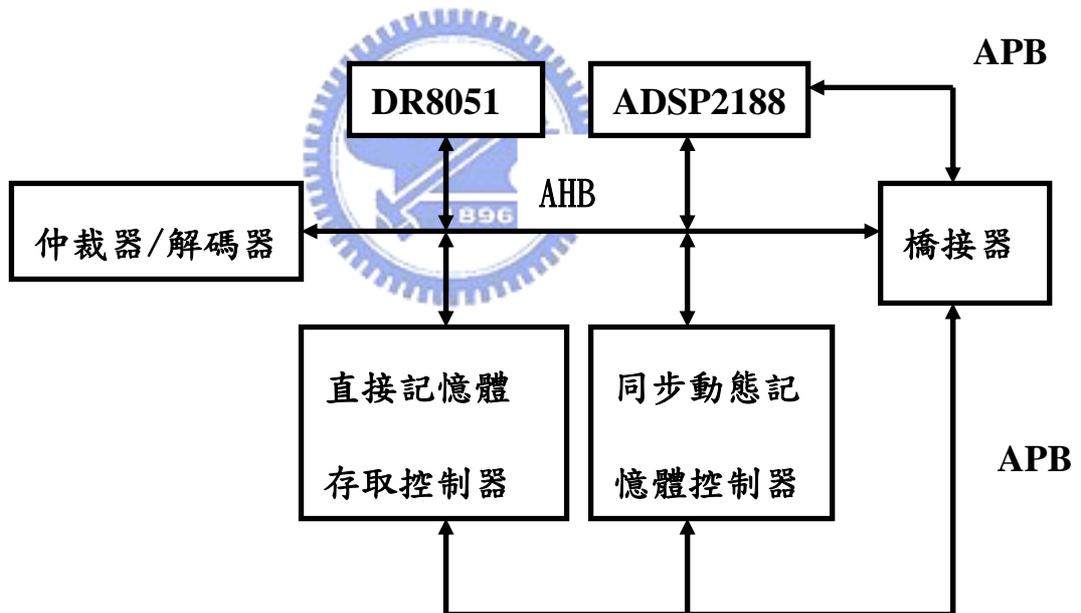


圖 3.2 系統架構

### 3.1 仲裁器

圖 3.3 顯示仲裁器輸入輸出介面。仲裁器的主要功用是同一時間只允許一個主控器使用匯流排，於每一個匯流排週期擷取主控器要求使用匯流排信號(HBUSREQ)，栓鎖匯流排信號(HLOCK)，之後利用仲裁機制選擇適當的主控器，並輸出 HGRANT 及 HMASTER 信號決定匯流排的所有權者去初始資料傳送。而仲裁器也接收從屬器所發出的回應信號及 HSPLIT 信號，完成其 Split 的動作。

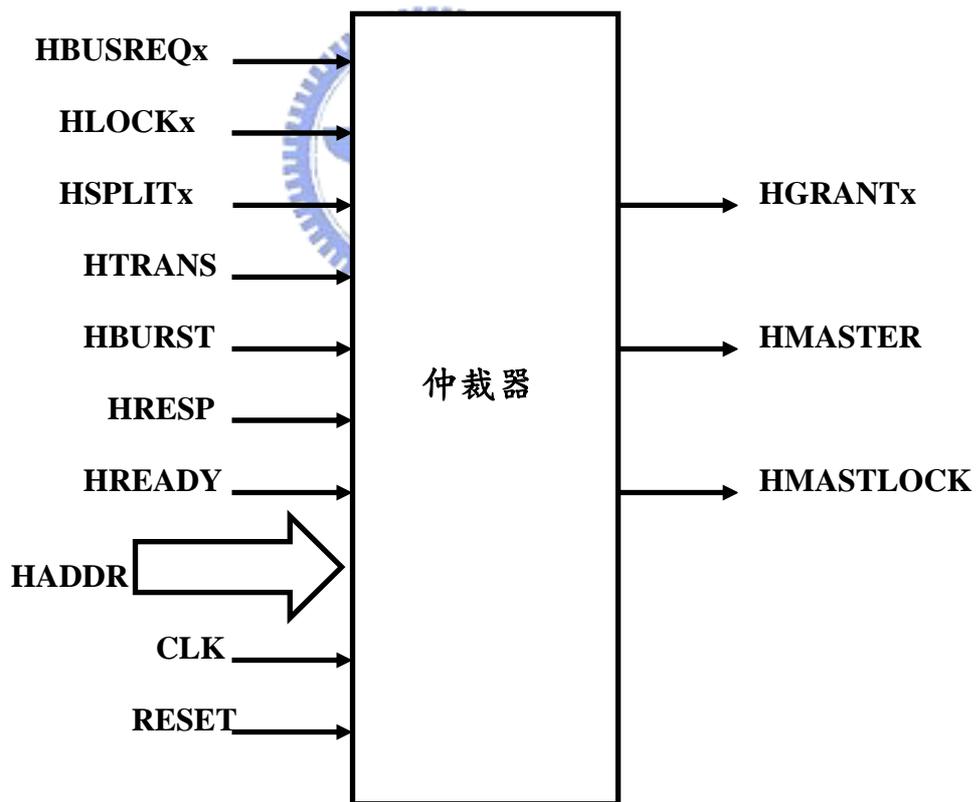


圖 3.3 仲裁器介面

圖 3.4 顯示仲裁器的架構，架構中包含了控制區塊及優先權區塊。

1. 控制區塊：此區塊為接收主控器的控制信號及從屬器的回應信號。再根據這些信號，適時控制優先區塊執行優先權判斷的動作，並發出 HGRANT<sub>x</sub>、HMASTLOCK 及 HMASTER 信號。包括：接收到從屬器回應的 SPLIT 信號，控制優先權區塊遮蔽主控器的優先權，並儲存主控器的號碼；接收到從屬器回應的 HSPLIT 信號，控制優先權區塊重新接收主控器的要求信號；接收到主控器的 HLOCK<sub>x</sub> 信號，發出 HMASTLOCK 信號與固定 HMASTER 信號等狀況。

2. 優先權區塊：執行優先權的判斷。並根據控制區塊遮蔽或開啟某個主控器的要求信號。至於判斷優先權的方法，本論文提供兩種優先權方案：固定(fixed)及循環(rotated)優先權判斷法。

固定優先權：每一個主控器都有其固定的優先順序。仲裁器每次都允許優先權較高的主控器先使用匯流排。

循環優先權：所有的主控器的優先權順序構成一個環狀。當某一個主控器使用完匯流排之後，其環狀順序的下一個主控器即可優先使用匯流排，所以剛使用完匯流排的主控器必須要等到其他要求使用匯流排的主控器使用完之後，才可以再取得使用權。這樣可避免靜態優先權中優先權高的主控器經常佔用

匯流排。

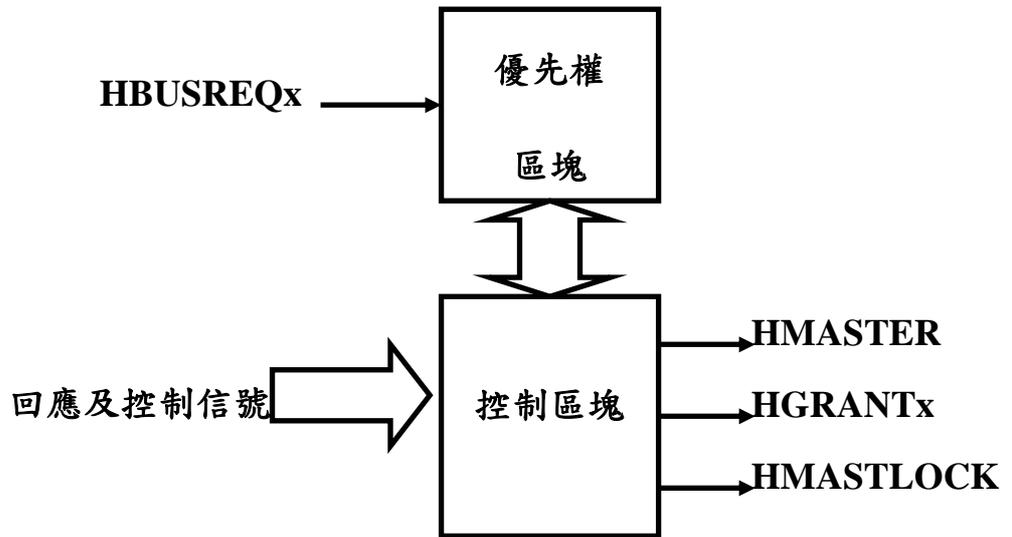


圖 3.4 仲裁器架構圖



### 3.2 解碼器

在 AHB 系統中，有一個中心位址解碼器，提供 HSELx 信號到每一個 AHB 從屬器，每一個 AHB 從屬器皆有其位址空間，解碼器可以根據位址信號判斷所選到的從屬器。這個解碼器本身只負責位址的解碼，當位址信號經解碼後，並未選到任何從屬器時，則由內定從屬器 (default slave) 根據 HTRANS 信號適時的回應 OKAY 或 ERROR 信號：當 HTRANS=IDLE 或 BUSY，則發出無等待週期的 HRESP=OKAY 信號；當 HTRANS=SEQ 或 NONSEQ，則發出 HRESP=ERROR 信號。

圖 3.5 顯示系統解碼器輸出輸入介面與 HSELx 的連接關係。在 AHB 的系統中，每一個從屬器最少必須有 1KB 的位址空間。

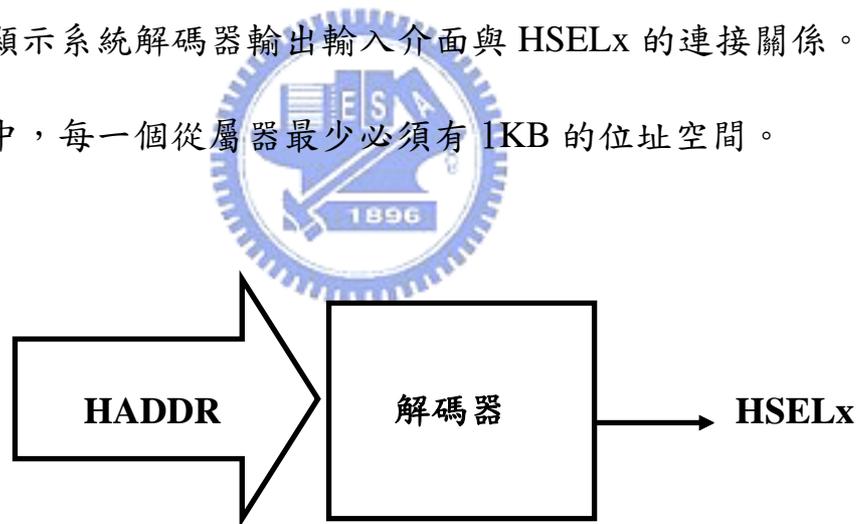


圖 3.5 解碼器

### 3.3 橋接器

橋接器扮演系統匯流排及週邊匯流排之間的介面。系統匯流排送出位址信號，當橋接器被選擇到後，橋接器再根據此位址信號，送出 PSELx 去選擇適當的 APB 從屬器。在傳送期間，只有一個 APB 從屬器可以被選取到。除初之外，橋接器也必須於寫入或讀取傳送時，送出 PENABLE 信號。圖 3.6 顯示了橋接器的輸出入介面。因為橋接器是 AHB 從屬器之一，所以必須要有對 AHB 系統溝通的介面。此外，橋接器也是 APB 系統唯一的主控器，也必須要有與 APB 從屬器溝通的介面。

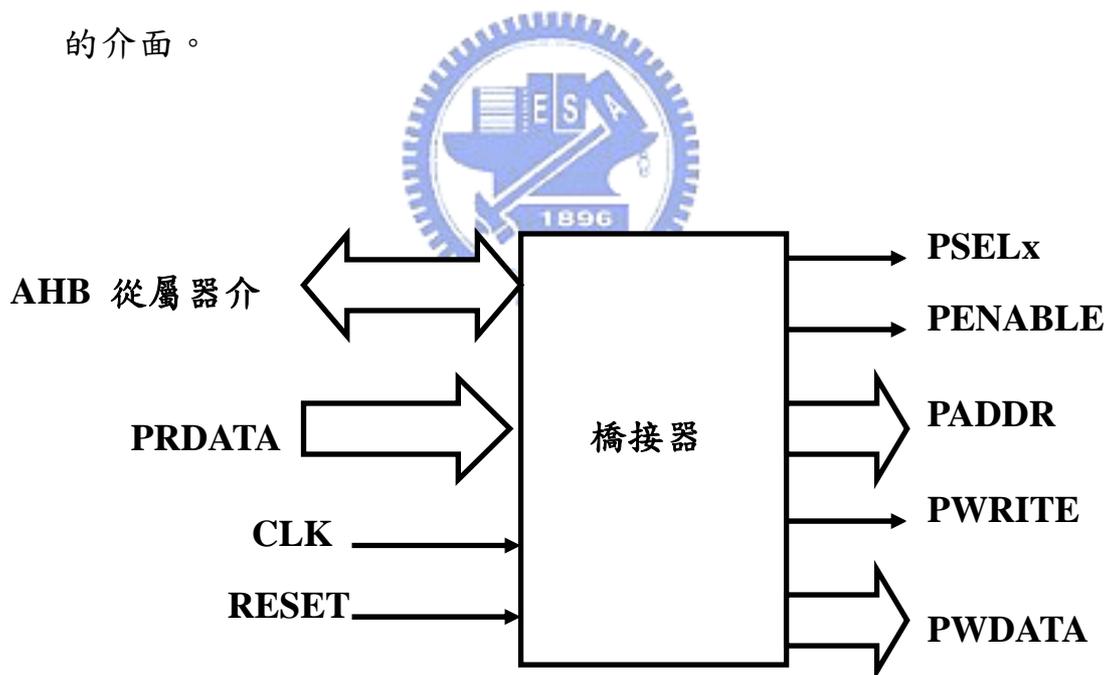


圖 3.6 橋接器

### 3.4 同步動態記憶體控制器

所謂同步動態記憶體是在現有的標準動態記憶體中加入同步控制邏輯(一個狀態機)，利用一個單一的系統時脈同步所有的位址和控制信號。使用同步動態記憶體控制器不但能提高系統表現，還能簡化設計、提供高速的數據傳輸。在功能上，它類似常規的動態記憶體(DRAM)，且也需時脈進行刷新(refresh)。可以說，同步動態記憶體是一種改善了結構的增強型動態記憶體。同步動態記憶體並不像靜態隨機存取記憶體(SRAM)一樣，可以在一個週期內完成一個存取動作，所以同步動態記憶體的存取過程分成好幾個狀態，而且通常需要3~7個週期才能完成。所以通常AHB的傳送需要插入多個等待週期，去等待同步動態記憶體的資料。

圖 3.7 顯示了同步動態記憶體控制器的架構圖。同步動態記憶體控制器包含了AHB及APB從屬器的介面。

APB從屬器的界面是用來接收系統控制者重制(reset)，刷新及模式暫存器[8]的設定指令，並儲存在暫存檔案中，之後同步動態記憶體控制器再根據這些檔案執行重制、定時的刷新或模式暫存器的設定動作。

AHB從屬器的界面是用來接收系統控制者所發出的控制、位址及資料信號，並根據控制信號於同步動態記憶體寫入或讀取資料。

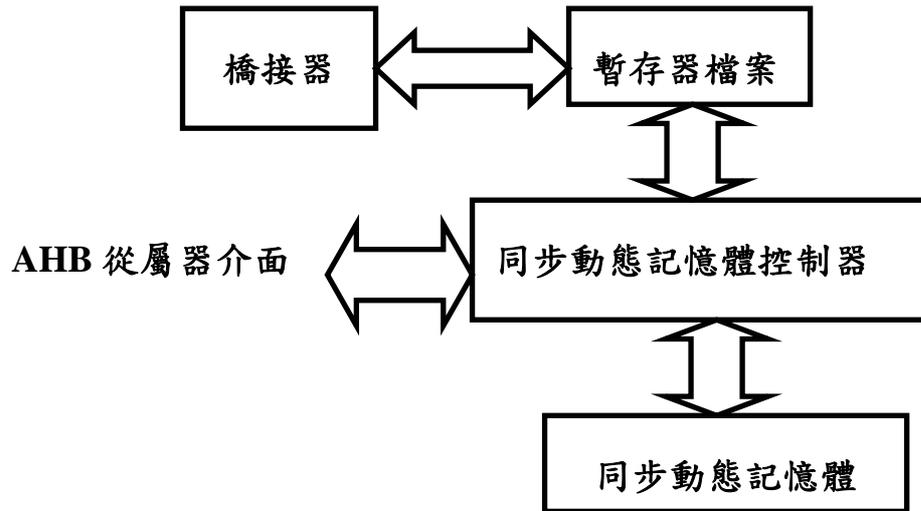


圖 3.7 同步動態記憶體控制器架構圖



### 3.5 直接記憶體存取

近年來直接記憶體存取已經成了每個系統基本且重要的裝置。直接記憶體存取是一種可以讓處理器減輕負擔的存取裝置。直接記憶體存取裝置直接讀取記憶體的資料，而不經過處理器，如此一來，處理器工作負擔減輕，資料傳輸將更快。圖 3.8 顯示了本論文直接記憶體存取控制器的架構圖。直接記憶體存取包含了 AHB 主控器及 APB 從屬器的介面。APB 從屬器介面接收系統控制者所發出的相關要求資料，並將其儲存在暫存器檔案中。根據暫存器檔案的內容，直接記憶體存取控制器開始從起始位址讀取資料，再將資料寫入目的位址。

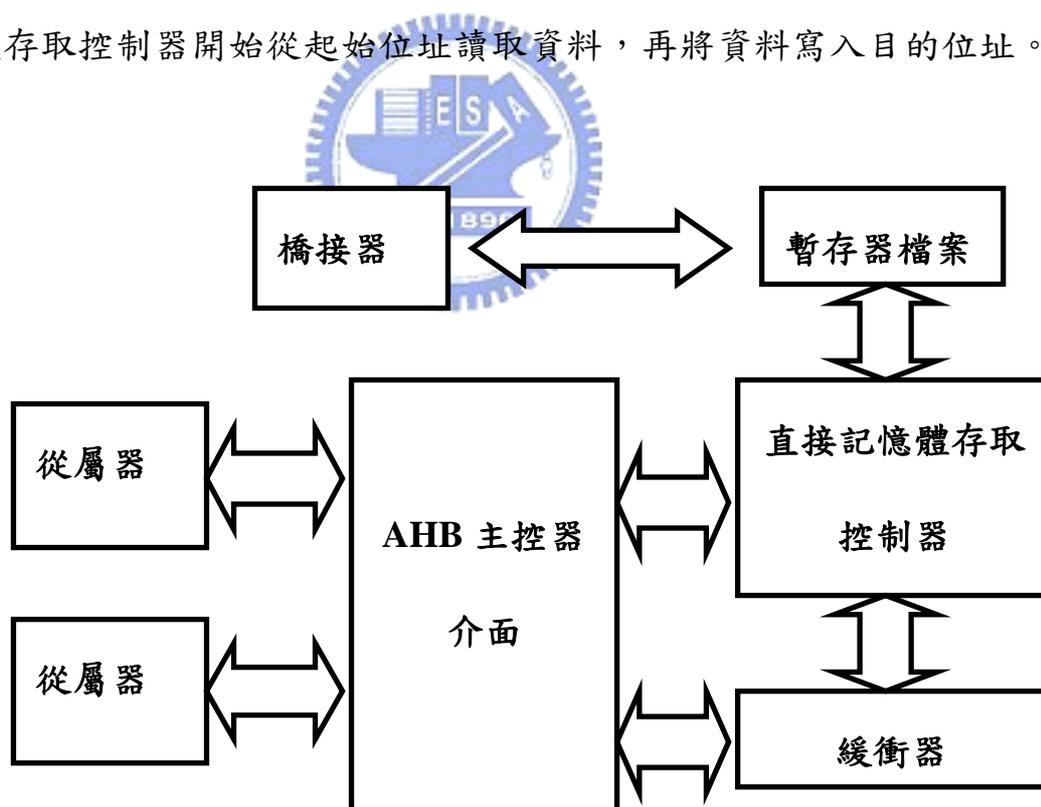


圖 3.8 直接記憶體存取控制器架構圖

表 3.1 列出直接記憶體存取控制器暫存器檔案內容。暫存器檔案儲存著直接記憶體存取控制器的相關狀態。根據暫存器檔案的資料，直接記憶體存取控制器執行所需的動作。

表 3.1 DMA 暫存檔案

位址	信號	描述
0x80000040	STARTADDR	直接記憶體存取讀取資料起始位址
0x80000044	LENGTH	直接記憶體存取傳送資料長度
0x80000048	DESTADDR	直接記憶體存取寫入資料起始位址
0x8000004c	CTRLREG	直接記憶體存取開始傳送信號
0x80000050	ENABLE	直接記憶體存取是否正在使用
0x80000054	COMPLETE	直接記憶體存取是否完成資料傳送
0x80000058	BURST	直接記憶體存取傳輸模式

直接記憶體存取控制器的傳送動作，可分成下列幾個步驟：

1. 系統控制者先寫入相關的暫存器檔案資料，並要求起始直接記憶體存取控制器的動作。
2. 直接記憶體存取控制器向 AHB 仲裁器發出匯流排要求信號 (HBURESQ) 及栓鎖匯流排的信號 (HLOCK)，並等待 AHB 仲

裁器回覆同意使用匯流排的信號(HGRANT)及 HREADY 信號。

3. 等到相對應的 HGRANT 及 HREADY 信號為高準位時，直接記憶體存取控制器開始執行資料的傳送，將某一從屬器的資料搬到某一從屬器。
4. 當傳送計數器為零時，表示傳送結束，此時將栓鎖匯流排的信號改為低準位，並發出傳送結束的信號告知系統控制者。等待下一次的要求。



### 3.6 DR8051

本論文選用 DR8051 作為本系統的微控制器。DR8051 是一個高性能且面積小的 8 位元單晶片嵌入式微控制器軟核心(soft-core)。它和工業標準的 8 位元 8051 微控制器 100% 的二進制相容 (binary-compatible)，且 DR8051 的設計主要是以低功率消耗為主，所以非常適合單晶片系統的設計。

DR8051 在整個系統中扮演 AHB 主控器的角色。它負責提供位址及控制信號去執行傳送動作。由於在 AMBA 規格中，只定義了 AMBA 主控器的介面，所以必須去發展 DR8051 轉換器來作為中間信號的轉換。



#### 3.6.1 DR8051 轉換器介面

DR8051 轉換器主要的功用就是處理 AHB 匯流排及 DR8051 智產間不同的協定。要設計 DR8051 轉換器，則必須發展 DR8051 與 AHB 匯流排間行為的關聯性。DR8051 轉換器的介面如圖 3.9 所示。圖 3.9 中，左半邊的信號為 AMBA 主控器介面，即 AHB 匯流排的信號，而右半邊為連接到 DR8051 智產的信號。表 3.2 列出右半邊信號所代表的意義與 AHB 匯流排信號間的關聯性，沒有列出的 AHB 匯流排信號即表示將在轉換器內處理，而不需傳至 DR8051 智產或是此信號在此

為固定值。至於兩者之間的信號轉換，本論文將利用 DR8051 外部記憶體控制信號[9]及緩衝器配置的方式，完成 DR8051 信號傳送。

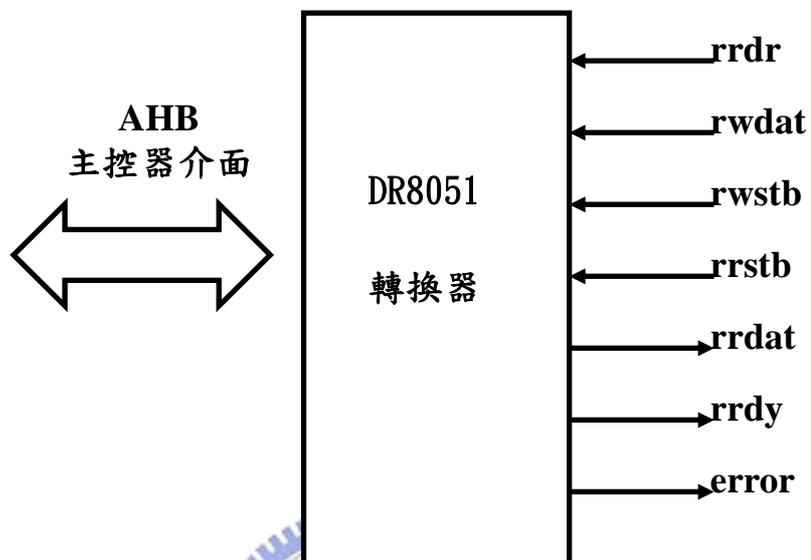


圖 3.9 DR8051 轉換器介面

表 3.2 DR8051 輸出信號意義與 AHB 匯流排信號之相關性

DR8051 信號	意義	相對應 AMBA 信號
<b>radr</b>	DR8051 送出的位址	<b>HADDR</b>
<b>rwdat</b>	DR8051 寫入資料	<b>HWDATA</b>
<b>rrdat</b>	DR8051 讀取資料	<b>HRDATA</b>
<b>rwstb</b>	DR8051 執行寫入動作信號	<b>HBUSREQ</b> 及 <b>HWRITE</b>
<b>rrstb</b>	DR8051 執行讀取動作信號	<b>HBUSREQ</b> 及 <b>HWRITE</b>
<b>rrdy</b>	DR8051 可執行讀取或寫入動作	<b>HRESP=OK</b>
<b>error</b>	DR8051 接收到從屬器錯誤的回應	<b>HRESP=ERROR</b>

### 3.6.2 DR8051 位址及資料緩衝器配置

由於 DR8051 為一個 8 位元的微控制器，而 AHB 位址匯流排為 32 位元，而資料匯流排建議最少為 32 位元，所以需要在兩者位元數不同的情形下，作一個配合。雖然在 AMBA 的規格中，可利用 HSIZE 的信號來決定所需傳送的位元數，但是利用 8 位元數來傳送資料，沒有利用 32 位元來傳送更有效率，因此在設計 DR8051 的 AHB 資料傳送時，固定傳送大小 HSIZE=WORD。所以必須要將 DR8051 中 8 位元的資料作結合，以滿足傳送大小為字元的需求。由於需將 8 位元資料結合成

32 位元資料，所以 DR8051 不提供爆發式傳輸，固定 HBURST = SINGLE。

本論文將利用一個 32 • 8 位元的資料緩衝器及一個 32 • 8 的位址緩衝器來儲存所要傳送或接收 32 位元匯流排信號，並利用外部記憶體控制信號來做為資料儲存及寫入的控制。首先利用外部記憶體位址 (xaddress) 為 0~63 的位址空間來做資料及位址緩衝器位址的控制信號。如圖 3.10 所示，將 xaddress=0~31 作為位址緩衝器的位址信號，xaddress=32~63 作為資料緩衝器的位址信號。

接著，將資料及位址緩衝器配置分割成 8 等份：在位址緩衝器配置中，xaddress=0~3 為第 1 等份，4~7 為第 2 等份 . . . . 28~31 為第 8 等份；在資料緩衝器配置中，xaddress=32~35 為第 1 等份，36~39 為第 2 等份 . . . . 60~63 為第 8 等份。另外，在位址及資料緩衝器配置中，位址緩衝器配置的第 N 等份與資料緩衝器配置的第 N 等份(N=1~8) 中的資料即為相配合所要傳送的位址及資料信號。所以在緩衝器每一等份中，將高位址到低位址內的資料構成的 32 位元資料即為傳送的資料。

之後，再利用 xaddress=64 作為存取 AHB 匯流排的控制信號。當 xaddress=64 時，表示 DR8051 將使用 AHB 匯流排，此時根據輸入資料 (xdatai) 第 2~0 位元(xdatai[2:0])來選擇資料及位址緩衝器的等份，

xdatai[3]用來判斷 DR8051 是要執行讀取或寫入的動作。

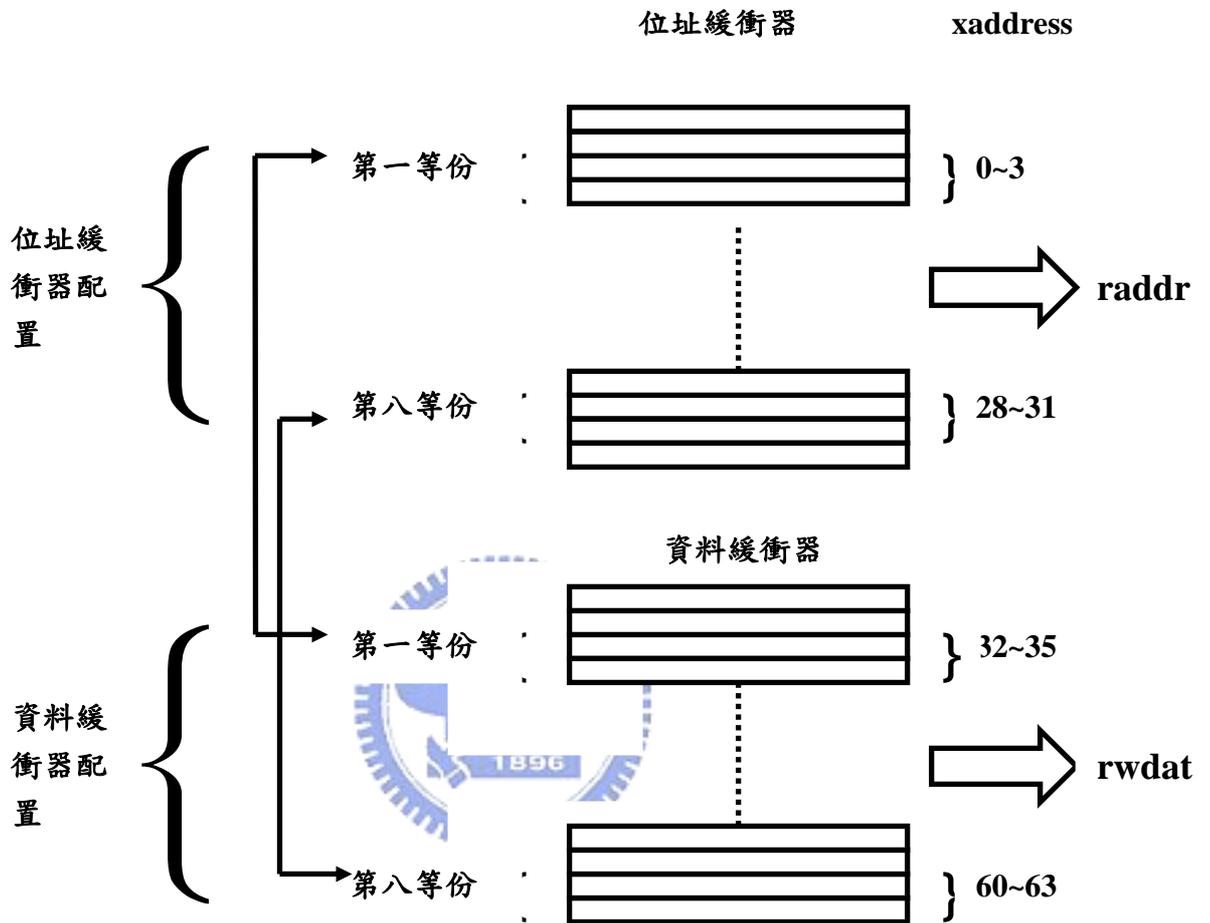


圖 3.10 緩衝器配置

圖 3.11 顯示了 DR8051 模組、外部記憶體及緩衝器的連接情形。

當  $xaddress=0\sim63$  時，即將 DR8051 資料寫入緩衝器或從緩衝器將資料讀入 DR8051； $xaddress=64$  時，即將 AHB 匯流排的資料讀入緩衝器或將緩衝器的資料寫入 AHB 匯流排。至於外部記憶體的規劃，DR8051

共有 256M 位元組的外部記憶體空間，本論文只使用了  $xaddress=0x4000$  至  $0xbfff$  的位址，所以當  $xaddress=0x4000\sim 0xbfff$  時，才是外部記憶體真正的動作。所以，DR8051 模組必須去用多工器選擇正確的輸入。

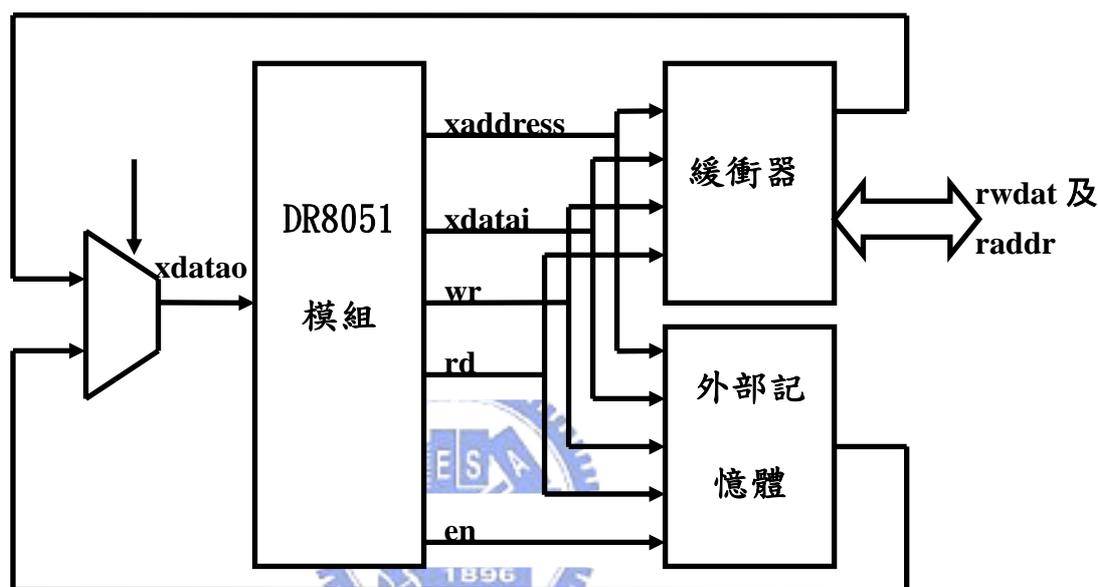


圖 3.11 DR8051 模組、外部記憶體及緩衝器連接情形

除了 DR8051 轉換器的狀態圖外，還需發展 DR8051 位址及資料緩衝器配置的狀態圖，以滿足兩者間的資料傳送。圖 3.12 顯示 DR8051 位址及資料緩衝器配置的狀態圖：

STANDBY：此時為單純的外部記憶體讀寫狀態。當外部記憶體的寫入位址為 64 時，則根據寫入資料  $xdatai$  所選擇到的資料及位址緩衝器配置等份中的資料分別輸出於  $rwdat$  及  $raddr$ ，並根據  $xdatai[3]$  來決定

rwstb 及 rrstb 的信號，接著進入 RBUS\_ACCESS 狀態。

**RBUS\_ACCESS:** 此時在等待 AMBA 匯流排的動作完成。如果 rrdy=1，表示 AMBA 的傳輸動作已經完成，此時將 xdatai[2:0] 選擇到的位址緩衝器配置內的資料加 4，同時若 rrstb=1，則將 rrdat 的資料寫入 xdatai[2:0] 選擇到的資料緩衝器配置中，接著進入 WAIT\_IDLE 狀態；若 error=1，則發出 interrupt 信號，接著進入 WAIT\_IDLE 狀態。

**WAIT\_IDLE:** 等待控制信號取消，以免重複之前的動作，接著進入 IDLE 狀態。

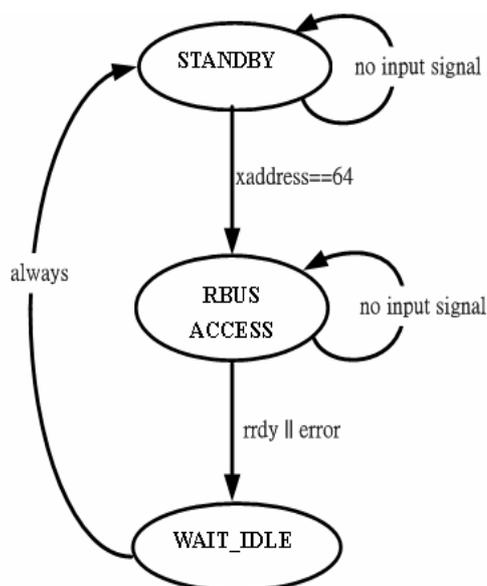


圖 3.12 緩衝器配置狀態圖

### 3.6.3 DR8051 轉換器狀態

確定 DR8051 與 AHB 匯流排之間的關聯性後，接下來將建立轉換

器的狀態圖。圖 3.13 顯示了 DR8051 轉換器的狀態圖。

**INITIAL:** 表示目前 DR8051 沒有資料要傳送或接收。當有資料要傳送時( $rrstb=1$  或  $rwstb=1$ )，此時發出 HBUSREQ 信號給仲裁器要求使用匯流排，接著進入 BUSREQ 狀態。

**BUSREQ:** 等待仲裁器的回應。當相對應的 HGRANT<sub>x</sub> 及 HREADY 信號為高準位時，表示能開始使用匯流排，並將要寫入的控制信號放至匯流排，接著進入 NONSEQ 狀態。

**NONSEQ:** 此時進行資料的傳輸，此狀態至少包含 2 個時脈週期：即位址及資料階段。在資料階段，若從屬器回應的 HREADY 信號為高準位時，且 HRESP=OK，表示傳輸結束，則發出  $rrdy=1$  信號告知外部記憶體可以接收資料，接著回到 INITIAL 狀態；若為 ERROR，則發出  $error=1$  信號告知外部記憶體忽略此次資料，並發出 HTRANS=IDLE 信號，接著回到 ERROR 狀態；若為 SPLIT 或 RETRY，並發出 HTRANS=IDLE 信號，接著回到 RE 狀態。

**ERROR:** 當接收到 HREADY=1 且 HRESP=ERROR，回到 INITIAL 狀態。

**RE:** 若為 HREADY=1 且 HRESP=RETRY 或 HREADY=1 且 HRESP=SPLIT 時，此時再發出 HBUSREQ 信號及相同的傳送信號給仲裁器，重新要求使用匯流排，並回到 BUSREQ 狀態，重新等待仲裁器

的回應。

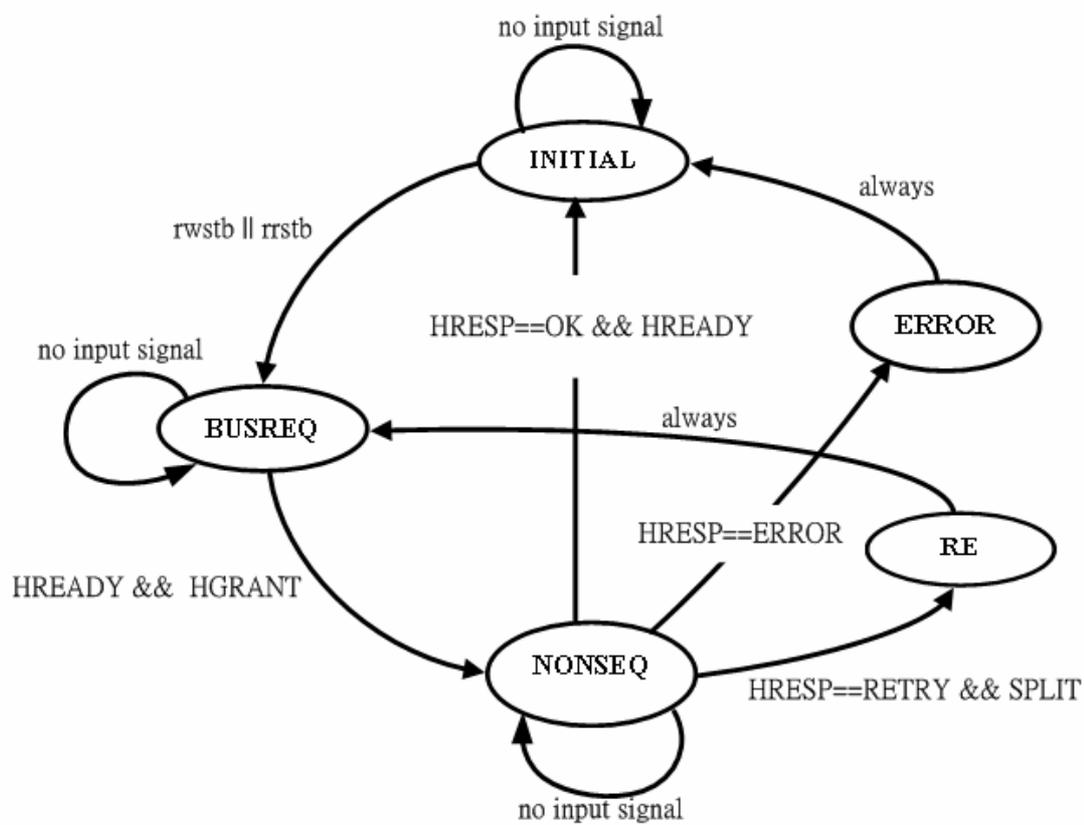


圖 3.13 DR8051 轉換器狀態圖

### 3.6.4 DR8051 信號實現情形

表 3.3 列出 DR8051 對於 AHB 主控器的信號實現情形。

表 3.3 DR8051 信號實現情形

主控器信號	實現與否	實現哪些傳送模式
HADDR[31:0]	實現	--
HTRANS[1:0]	實現	IDLE、NONSEQ
HWRITE	實現	--
HSIZE[2:0]	實現	WORD
HBURST[2:0]	實現	SINGLE
HPORT[3:0]	未實現	--
HWDATA[31:0]	實現	--
HBUSREQ	實現	--
HLOCK	實現	--

### 3.7 ADSP2188

本論文中，選擇 ADSP2188 為本系統的數位信號處理器。ADSP21xx 系列數位信號處理器，本身有一些特性：如為 16 位元定點數位信號處理器，帶 8 位元保護位的 40 位元，單週期執行多數指令等，除此之外，ADSP21xx 系列提供了一整套軟件開發工具及相應的仿真器開發平台 (分別是 VisualDSP 和 VisualDSP++ 系列)，對於系統的調試與仿真提供了極大的方便。它的主要特點是：可以仿真調試從 ADSP2101~2189 全系列的數位信號處理器，因此使得之後的實驗結果驗證，更為簡易正確。

ADSP2188 在整個系統中同時扮演 AHB 從屬器及 APB 從屬器的角色。它負責回應系統控制者所發出的傳送信號。

APB 從屬器的界面是用來接收系統控制者重制(reset)、起始及暫停的指令，並可讀取 ADSP2188 一些重要的旗標值及中斷值。

而 AHB 從屬器的角色是接收系統控制者的控制信號及資料，將資料儲存至資料或程式記憶體中，處理一些有關數位信號處理的程式；或將運算完的資料結果從記憶體中讀出。圖 3.14 顯示 AHB 匯流排與 ADSP2188 資料及位址連接的情形。系統控制者可利用 AHB 匯流排將資料寫入 ADSP2188 資料記憶體及位址記憶體，或從 ADSP2188 資料記憶體及位址記憶體將資料讀出，其中資料記憶體資料為 16 位元，位

址記憶體資料為 24 位元。由圖 3.14 可知，AHB 及 ADSP2188 單元都可使用資料及位址記憶體。

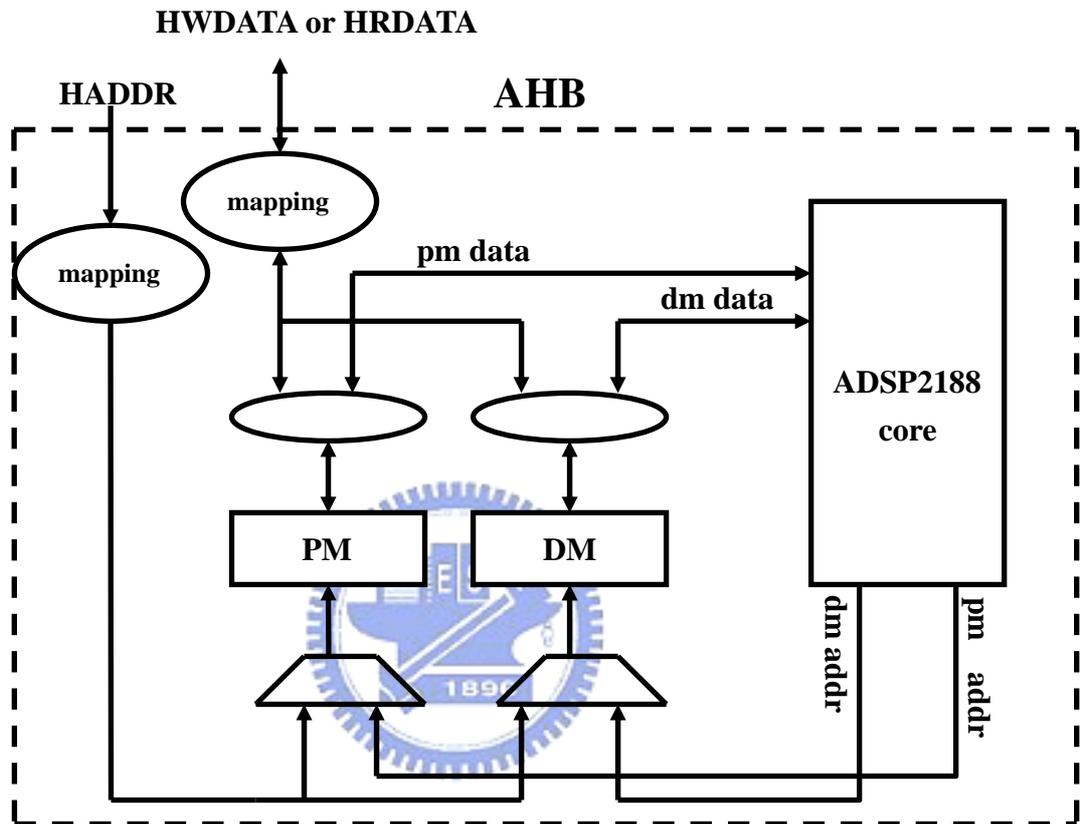


圖 3.14 ADSP2188 記憶體連接情形

### 3.7.1 ADSP2188 轉換器介面

由於 AMBA 規格，只定義了 AMBA 從屬器的介面，所以必須去發展 ADSP2188 轉換器來作為中間信號的轉換。然而由於 ADSP2188 在此系統中扮演加速運算的角色，因此為了維持本身的高效能，ADSP2188 的時脈週期將高於系統的時脈週期，以提高運算的速度。所以 ADSP2188 轉換器必須要對於兩者的時脈週期之間的傳送作一些適

當的調整，才能使資料的傳送正確無誤。

要設計 ADSP2188 轉換器，則必須發展 ADSP2188 與 AHB 匯流排間行為的關聯性。ADSP2188 轉換器的介面如圖 3.15 所示。圖 3.15 中，左半邊的信號為連接 AHB 匯流排的信號，右半邊為連接到 ADSP2188 智產的信號[10]。表 3.4 列出右半邊信號所代表的意義與 AHB 匯流排信號間的關聯性，沒有列出的 AHB 匯流排信號即表示將在轉換器內處理，而不需傳至 ADSP2188 智產或是此信號在此為固定值。

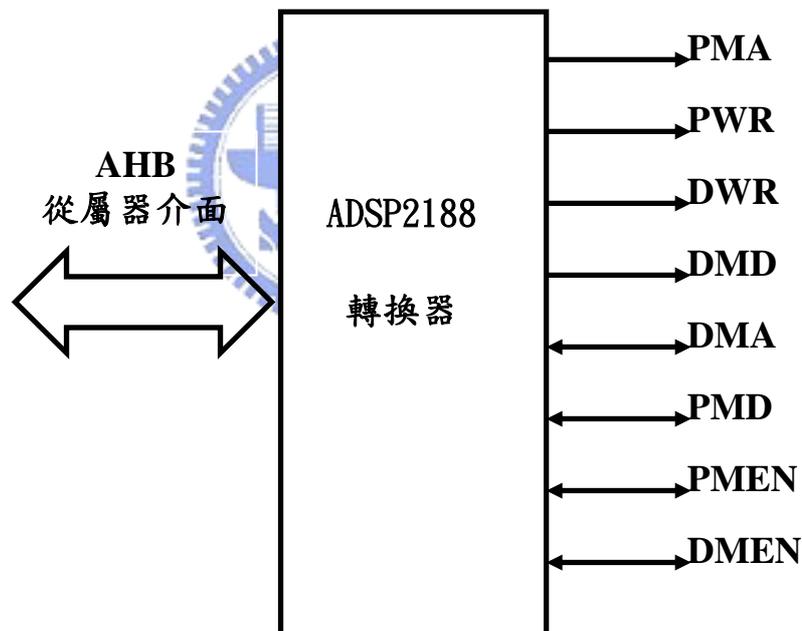


圖 3.15 ADSP2188 轉換器介面

表 3.4 ADSP 轉換器 ADSP 介面信號意義與 AHB 匯流排信號相關性

ADSP2188 信號	意義	對應 AMBA 信號
PMA	ADSP 程式記憶體的位置	HADDR
DMA	ADSP 資料記憶體的位置	HADDR
PMD	ADSP 程式記憶體的資料	HWDATA 或 HRDATA
DMD	ADSP 資料記憶體的資料	HWDATA 或 HRDATA
PWR	寫入或讀取程式記憶體控制信號	HWRITE
DWR	寫入或讀取資料記憶體控制信號	HWRITE
DMEN	資料記憶體的致能信號	HADDR
PMEN	程式記憶體的致能信號	HADDR

由於 ADSP2188 本身的時脈週期不同於系統的時脈週期，所以在 ADSP2188 轉換器的設計中，必須要能處理兩個時脈週期間的資料傳輸。因此，ADSP2188 轉換器就必須有處理兩個時脈週期的介面，並能將資料作適當的轉換，以免由於時脈的不一致，造成錯誤資料的擷取。因此在 ADSP2188 轉換器中，不單單只有一個狀態機。圖 3.16 顯示了

ADSP2188 轉換器中區塊圖。在 ADSP2188 轉換器中主要由 5 個區塊所構成：

- 1.爆發式傳送控制區塊：支援爆發式傳輸模式。將接收相關的爆發式傳輸信號轉成適當的控制信號，控制系統介面控制區塊進行一系列連續的傳輸動作。
- 2.系統介面控制區塊：以系統時脈執行同步時脈的動作。負責將 AHB 匯流排的控制信號轉成適當的信號輸入至 ADSP 介面控制區塊；或將 ADSP 介面控制區塊的回應信號轉成適當的信號輸出至 AHB 匯流排。並適時的發出 VAL 信號，表示將有資料要傳送。
- 3.ADSP 介面控制區塊：以 ADSP 時脈執行同步時脈的動作。負責將系統介面控制區塊的控制信號轉成適當的信號輸入至 ADSP2188 介面；或將 ADSP2188 介面的信號轉成適當的回應信號輸出至系統介面控制區塊。並適時的回應 ACK 信號，表示資料傳送完成。
- 4.位址解碼區塊：負責從系統介面控制區塊擷取正確的位址信號，並將系統控制者與 ADSP2188 的位址信號作適當的解碼。
- 5.資料路徑：負責從系統介面控制區塊擷取正確的資料信號，或從 ADSP2188 擷取回應的資料信號，並將資料作適當的大小轉換。

之後，將對每個區塊作較詳細的說明。

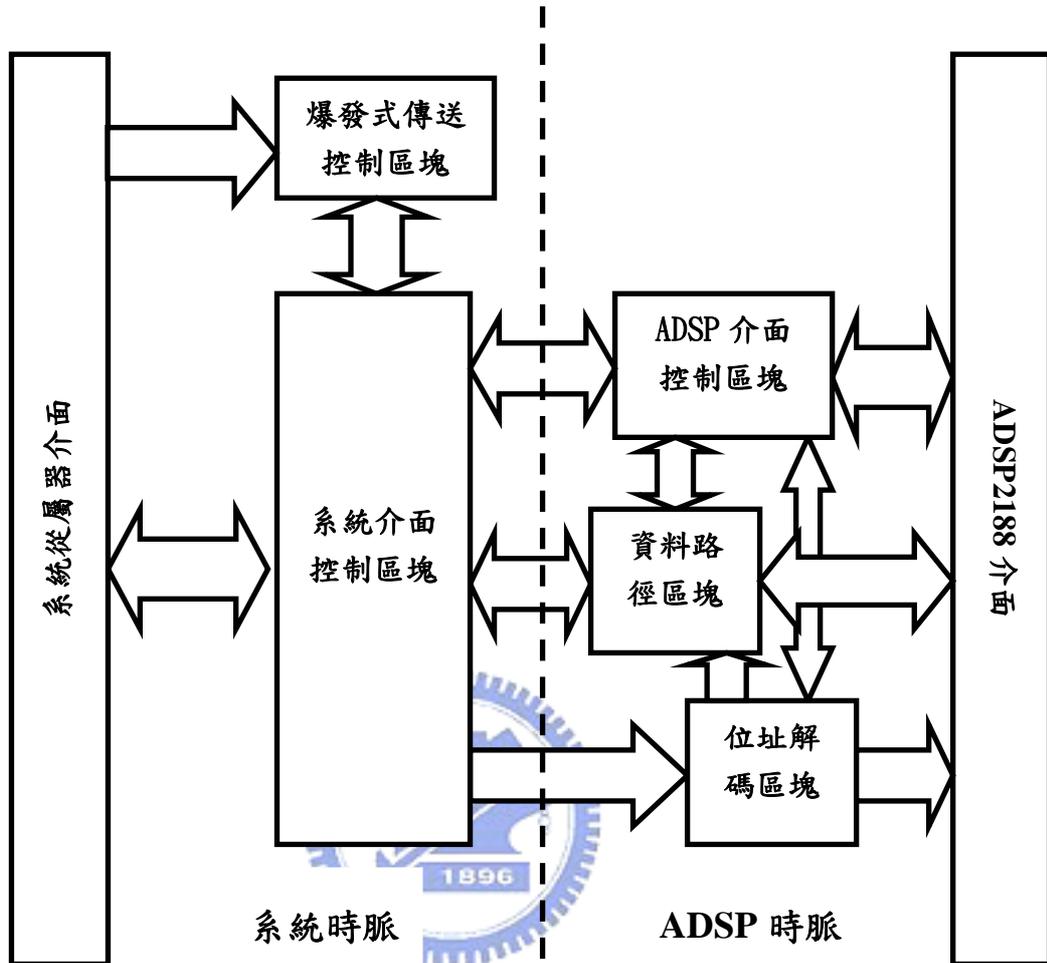


圖 3.16 ADSP2188 轉換器區塊圖

### 3.7.2 爆發式傳送控制區塊

此區塊主要是根據 HBURST 信號，來計算所要連續傳送的個數。在 AHB 匯流排的爆發式傳送模式中，有 4 種固定的連續傳送個數，包含了 1、4、8 及 16 個傳送個數。所以在爆發式傳送控制區塊必須有一個計數器來計算所需要傳送的個數，並根據系統控制介面區塊的狀態來計數。當計數結束，則通知系統控制介面區塊傳輸結束。

### 3.7.3 位址解碼區塊

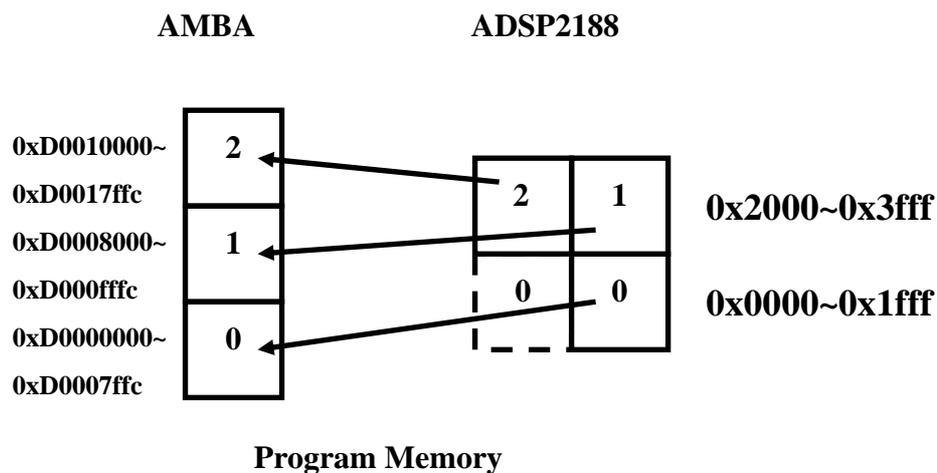
位址匯流排的信號，除了供 AMBA 系統中心解碼器選擇之外，還必須將資料進一步的解碼，供 ADSP2188 轉換器內部的位址解碼器做適當的 ADSP2188 記憶體位址選擇。所以此區塊是根據系統控制介面區塊的狀態，從系統控制介面區塊擷取正確的位址信號，並加以解碼，將正確的位址信號傳給 ADSP2188。

然而在 AMBA 與 ADSP2188 記憶體配置中，有著記憶體重疊的問題。如圖 3.17 所示，在 ADSP2188 的記憶體中，是採用記憶體重疊的方式。在資料記憶體中，位址 0x2000~0x3fff(0x 代表 16 進制)為共用的記憶體，然而 0x0~0x1fff 的記憶體區塊卻是可變的。相對的，在位址記憶體中，位址 0x0~0x1fff 為共用的記憶體，然而 0x2000~0x3fff 的記憶體區塊卻是可變的。



圖 3.17 ADSP2188 程式記憶體示意圖

所以，必須將 ADSP2188 的重疊記憶體配置到 AMBA 中線性的位址，如圖 3.18 所示。



**圖 3.18 位址配置**

根據此一概念，表 3.5 列出 AHB 位址匯流排位元所代表的意義。

位址解碼區塊即根據此表將 AHB 位址匯流排解碼成適當的 ADSP2188

資料或程式記憶體位址及其記憶體重疊區塊。

表 3.5 位址匯流排解碼

位址匯流排位元	意義
31~28	供 AMBA 系統中心解碼器選擇
27	0: 程式記憶體; 1: 資料記憶體
26~21	決定資料大小轉換的方式
19~16	選擇記憶體重疊區塊
15~2	記憶體位址
20, 1~0	保留

#### 3.7.4 資料路徑區塊



此區塊是根據系統控制介面區塊的狀態，從系統控制介面區塊擷取正確的資料信號，將正確的資料信號傳給 ADSP2188；或根據 ADSP 介面控制區塊的狀態，從 ADSP2188 介面擷取回應的資料信號，再傳給系統控制介面區塊。並將兩者間資料大小作適當的轉換。

將資料大小作適當的變換，需根據 HADDR 及 HSIZE 信號。利用 HSIZE 來判斷 AMBA 資料大小，再根據 HADDR[26:20]來決定資料的轉換方式。當資料由小轉大，採用零值擴充(zero extended)；資料由大轉小，則根據 HADDR[26:21]來選擇適當的資料區段，若為整數倍，則資料的切割沒有重疊，若為非整數倍，則資料的切割最後會有重疊的

情形。圖 3.19 顯示 AMBA 資料(32 位元)與 ADSP2188 程式記憶體資料(24 位元)資料轉換情形。

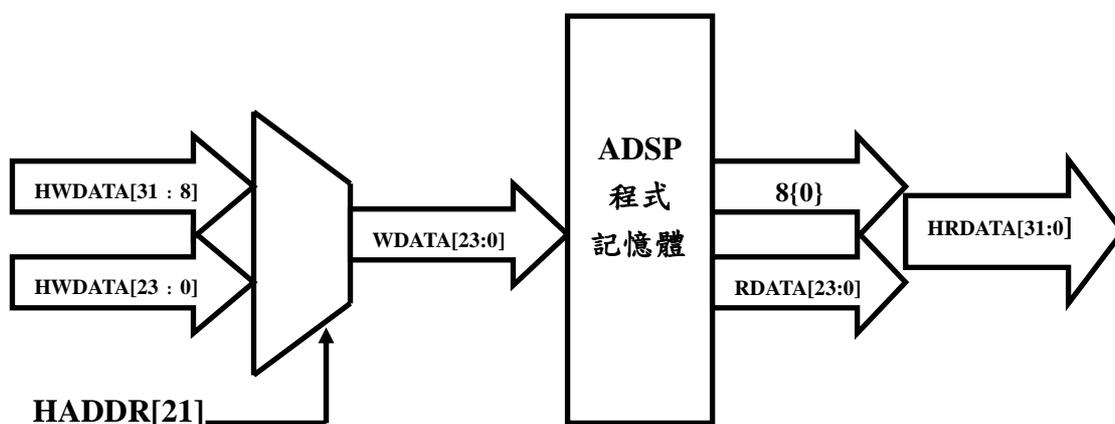


圖 3.19 資料轉換情形

### 3.7.5 系統介面控制區塊

接下來將分別建立系統介面控制區塊與 ADSP 介面控制區塊的狀態圖，以配合兩者不同時脈週期間的資料傳送。

圖 3.20 顯示了系統介面控制區塊的狀態圖。在此介面，主要是以系統時脈週期的正源觸發來處理資料的轉換。

**IDLE:** 表示 ADSP2188 目前沒被系統控制者所選擇到。當 HSEL<sub>x</sub>=1 及 HREADY=1 時，則將系統控制者發出的控制信號栓鎖住，並發出 VAL 信號[11]告知 ADSP 介面控制區塊將要存取記憶體，接著進入 LATCH\_WDATA 狀態。

**LATCH\_WDATA:** 將系統控制者的寫入資料 HWDATA 栓鎖住，之

後進入 BUSY 狀態。此時並判斷栓鎖的 HADDR 信號是否為適當的資訊，若不是則發出 HRESP=ERROR 信號通知系統控制者，之後則是進入 ERROR 狀態。

BUSY: 此時等待 ADSP 介面控制區塊回應的 ACK 信號，接著將讀出的資料放到 HRDATA，並將 HREADY=1。所以此狀態即是在等待從屬器角色讀/寫資料的狀態，而此時利用 HREADY=0 來延長傳送。至於下一個狀態，若 ADSP2188 記憶體正在使用(pms=1 或 dms=1)，則馬上發出 HRESP=SPLIT 信號通知系統控制者，接著回到 SPLIT 狀態；若傳送完成，則需根據爆發式傳送控制區塊判斷是否有資料要繼續傳送，當有資料要繼續傳送(end=0)：若為寫入，則進入 WAIT\_WDATA 狀態，若為讀取，則進入 LATCH\_WDATA 狀態；若無資料要傳輸(end=1)，則進入 IDLE 狀態。

ERROR: 發出 HRESP=ERROR 信號通知系統控制者，並將 HREADY=1，之後回到 IDLE 狀態。

WAIT\_WDATA: 此時在等待下一筆要寫入的資料，並栓鎖住下一筆的控制信號，之後回到 LATCH\_WDATA 狀態。

SPLIT: 發出 HRESP=SPLIT 信號通知系統控制者，並將 HREADY=1，之後回到 DELAY 狀態。

DELAY: 等待 ADSP2188 記憶體使用權，當 ADSP2188 本身使用

完畢(pms=0 或 dms=0)，則根據所紀錄的 HMASTER 信號，發出適當的 HSPLITx 信號給仲裁器，接著回到 IDLE 狀態。

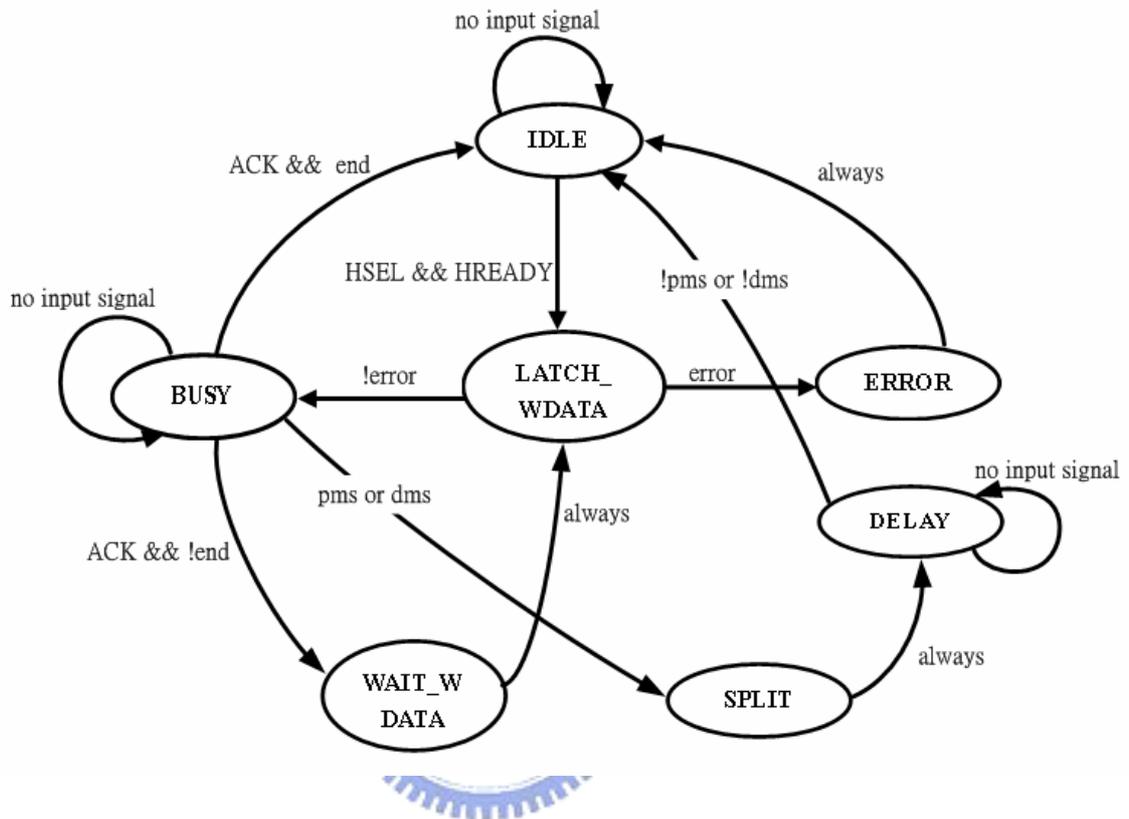


圖 3.20 ADSP2188 包裝器系統介面狀態圖

### 3.7.6 ADSP 介面控制區塊

圖 3.21 顯示了 ADSP 介面控制區塊的狀態圖。在此介面，主要是以 ADSP2188 本身的時脈週期正源觸發來進行狀態的轉換。

IDLE：此時 AHB 並沒有要存取 ADSP2188 記憶體。當接收到系統介面控制區塊 VAL 信號，則根據系統介面控制區塊輸入信號判斷所要存取的是資料或程式記憶體，則進入 WAIT\_PM(HADDR[27]=0)或

WAIT\_DM(HADDR[27]=1)狀態；若為不適當的資訊則回到 IDLE 狀態。

WAIT\_PM: 此狀態是要等待使用 ADSP2188 程式記憶體。當 ADSP2188 本身沒有使用時(pms=0)，則可開始存取記憶體，並根據所要執行的動作，進入 WRITE(HWRITE=1)或 READ(HWRITE=0)狀態。

WAIT\_DM: 此狀態是要等待使用 ADSP2188 資料記憶體。當 ADSP2188 本身沒有使用時(dms=0)，則可開始存取記憶體，並根據所要執行的動作，進入 WRITE(HWRITE=1)或 READ(HWRITE=0)狀態。

WRITE: 此時判斷系統控制者所要寫入的資料是否已經準備好。若已準備好則可以執行寫入。當動作完成，則發出 ACK 信號[11]通知系統介面控制區塊，表示傳輸完成，並根據系統介面控制區塊狀態，取消 AHB 存取 ADSP2188 記憶體的信號，接著進入 IDLE 狀態。若還沒準備好，則必須一直維持此狀態。

READ: 此時判斷系統控制者所要讀取的資料是否已經準備好。若資料已準備好，則發出 ACK 信號通知系統介面控制區塊，並根據系統介面控制區塊狀態，取消 AHB 存取 ADSP2188 記憶體的信號，接著進入 IDLE 狀態。若還沒準備好，則必須一直維持此狀態。

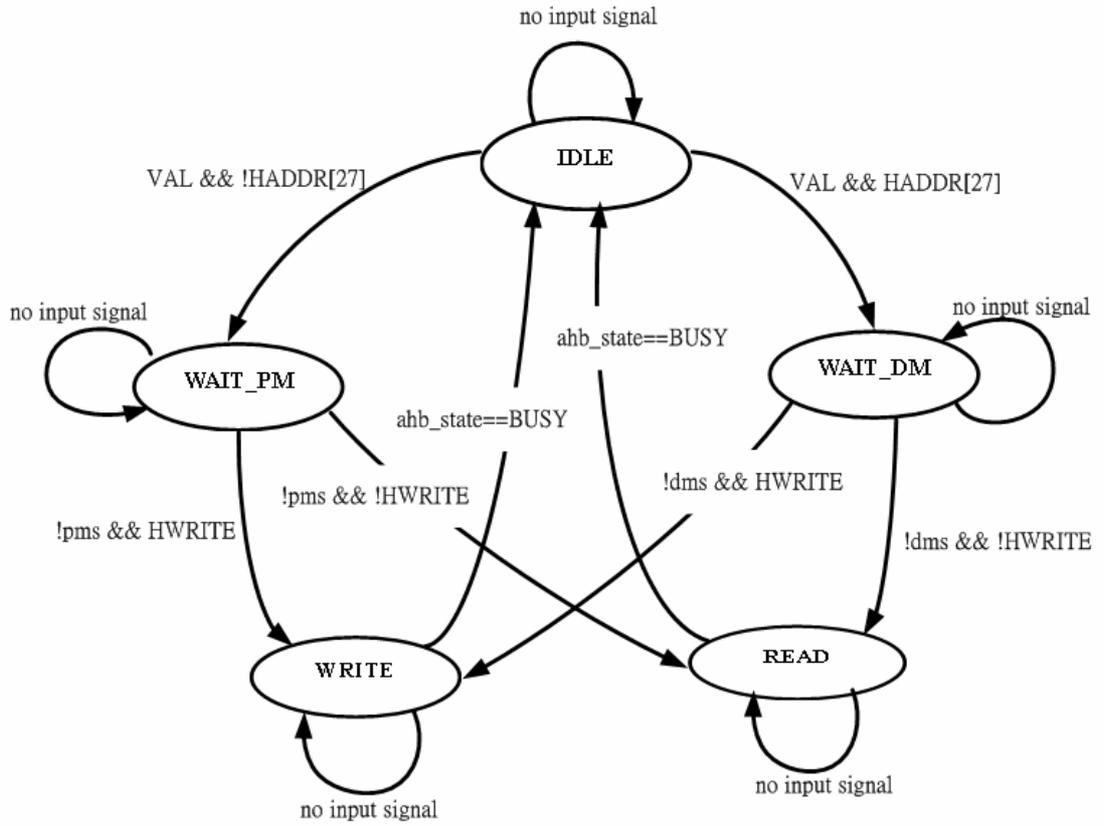


圖 3.21 ADSP2188 轉換器 ADSP2188 介面狀態圖

### 3.7.7 ADSP2188 信號實現情形

表 3.6 列出 ADSP2188 對於 AHB 從屬器的信號實現情況。

表 3.6 ADSP2188 信號實現情形

從屬器信號	實現與否	實現哪些傳送模式
HRDATA[31:0]	實現	--
HREADY	實現	--
HRESP[1:0]	實現	OKAY、ERROR、RETRY、 SPLIT
HSPLIT[15:0]	實現	--

