

第七章 DMA 控制器

此章節說明 PXA250 與 PXA210 之單晶片 (On-chip) DMA 控制器 (DMAC)。為了回應內部與外部週邊所發出的要求，DMAC 會傳送資料至主記憶體，或從主記憶體傳送資料出來。因為週邊並不支援依據位址與命令，直接將資料傳送至主記憶體。DMAC 具有 16 個 DMA 通道 (DMA Channel)，0-15，每一個自週邊發出的 DMA 要求至少會產生一個記憶體匯流排週期。

DMA 通道是提供許多裝置直接與記憶體收送資訊的系統路徑。DMA 一般是由磁碟，音效裝置，LCD 等設備所使用。

在這一章中，將說明 DMA 控制器與 DMA 通道，以及他們如何運作。在早期的電腦系統中，微處理器處理所有的工作。除了執行程式之外，也要負責週邊裝置的資料收送。不過若由微處理器執行這些資料傳送，由於在傳送與接收資料時微處理器無法執行其他工作，通常系統執行效率較低。

DMA 則扮演了一個中間者的腳色。允許處理器在週邊裝置收送資料的同時能執行其他工作。使得效能更為增加。DMA 是由一些特殊的通道以及控制這些通道的電路所組成。這些通道與電路可以在傳輸資料時不需要由微處理器控制所有的工作。這些電路通常是晶片組或主機板的一部份。

DMA 可大致分為第三方 DMA 或第一方 DMA。

第三方(third party)DMA

由於 DMA 控制器是在資料傳送的過程當中，除了傳送端與接收端兩方以外，實際進行傳輸的另外一方，因此標準的 DMA 有時又稱為第三方 DMA。

第一方(first party)DMA(匯流排主控)

有些週邊設備在傳輸資料時，確實拿到系統匯流排的控制權，支援這種型態傳輸的 DMA 則稱為第一方 DMA。又稱為匯流排主控。

因為較新式的週邊設備具有比舊式的 DMA 控制器更聰明與更快速的硬體線路，因此匯流排主控比一般 DMA 提供更好的效能。目前更新的 DMA 模式如 Ultra DMA 提供非常高的傳輸率。

在較新的設備中，磁碟通常使用程式化 I/O(PIO)或是 PCI 匯流排的第一方 DMA(匯流排主控)來進行資料的傳輸，以增加傳輸的效率。

DMA 控制器

標準的 DMA 傳輸是由內建於系統晶片組的 DMA 控制器來處理。目前許多的系統晶片皆在晶片中內建 DMA 控制器。

DMA 通道

DMA 通道是用來傳輸資料的電路。在 DMA 控制器中，通常會包含幾個通道。如早期 PC 的 DMA 具有 4 個通道。而現代的 PC 則大部份具兩個 DMA，共有八個通道。在 Intel 的 XScale 處理器中的 DMA 控制器則具有 16 個通道。

通常在 DMA 中，不同的通道傳輸的機率可能不同。因此有許多通道通常可作為特定的用途。比如有些通道的優先權比其他通道高，則適合用來作為多媒體資料的傳輸。善用 DMA 提供的功能，可以讓系統的運作更加具有效率。

7.1 DMA 說明

Intel XScale 晶片的 DMAC 只支援 Flow-through 傳送方式。這種傳送方式在目的端的緩衝區／記憶體保留住資料之前，資料會先通過 DMAC。DMA 控制器也可使用 Flow-through 傳送方式，做記憶體至記憶體的移動。

第 7-1 頁，圖 7-1「DMAC 區塊圖」提供一個 DMAC 的概觀。第 7-3 頁，表

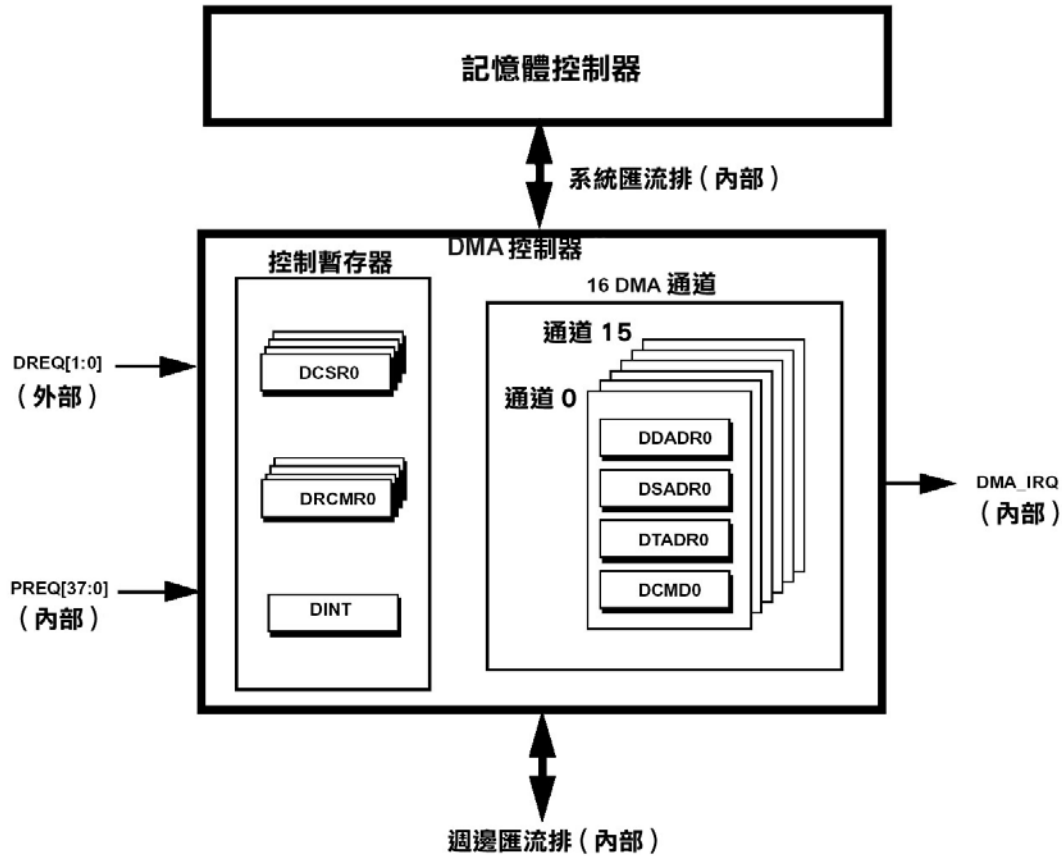


圖 7-1 DMAC 區塊圖

7-1 「DMAC 訊號表」提供 DMAC 訊號與說明表。

7.1.1 DMAC 通道 (Channel)

DMAC 具有 16 個通道，每一個通道由 4 個 32 位元的暫存器所控制。每個通道可以設定其組態，以 Flow-through 傳輸方式來服務任何內部週邊或其中一個外部週邊(輔助晶片)。每個通道會依據週邊的裝置埠寬度為單位傳送。並以週邊裝置的突發傳輸大小的增量由 DMAC 服務其傳送要求。每個裝置的突發傳輸大小與埠的寬度是以裝置的 FIFO 深度與需要的頻寬為基礎，可在通道暫存器內以程式化的方式設定。基於效能的考量，強烈建議使用者將突發傳輸大小設定為與 FIFO DMA 中斷觸發器層級相同，此稱為 FIFO 門檻層級。當多個通道主動執行時，DMAC 使用突發傳輸的傳輸量來服務每個通道。在突發傳輸的資料量傳送之後，DMAC 可能執行一個上下文轉換，切換到另一個有效的通道。DMAC 執行上下文轉換是以通道的活動率為基礎，而不管它的目標設備是否正在要求服務，或是通道正在優先權排程中。

通道資訊的設定必須以個別通道為基礎，設定在 DMAC 暫存器中。請參考第-28 頁，表 7-13 「DMAC 控制暫存器」。DMAC 支援 2 個載入 DMAC 暫存器

的方法—無敘述元抓取 (No-Descriptor Fetch) 與敘述元抓取(Descriptor Fetch) 法。在第 7-6 頁，7.1.4 節「DMA 敘述元」中，對於抓取模式有更詳細的解說。

當設定 DMA 組態時，軟體必須確定快取記憶的一致性。由於 DMAC 不會檢查快取記憶，所以目標和來源位址必須在記憶體管理單元 (Memory Management Unit) 內設定為不可快取 (non-cacheable)。

對於週邊產生的每一個資料傳送的要求，會導致一次記憶體資料的讀取或寫入。除非週邊準備好讀取或寫入整個資料區塊 (8、16 或 32 位元組)，而且設置好來處理讀取與寫入小於一個完整的資料區塊，否則不能要求 DMA 傳送。在 DMA 傳輸的最後可能會讀取或寫入小於一個完整的資料區塊。

7.1.2 訊號說明

表 7-1 指出 DMAC 所控制的 DREQ[1:0]、PREQ[37:0]與 DMA_IRQ 訊號。

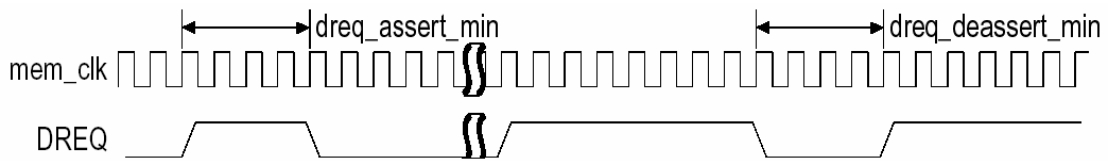
表 7-1 DMAC 訊號表

訊號	訊號類型 輸出／輸入	目標／來源	定義
DREQ[1:0]	輸入	腳位	外部相容晶片要求線。DMA 偵測此正向邊緣並當做一個要求。
DMA_IRQ	輸出	中斷控制器	主動高電位訊號為一個要求。
PREQ[37:0]	輸入	On-chip 週邊	內部週邊 DMA 要求線。On-chip 週邊使用 PREQ 訊號送出要求。 DMAC 不會偵測 PREQ 訊號，直到它自週邊傳送資料至記憶體完成。

7.1.2.1 DREQ[1:0]與 PREQ[37:0]訊號

當需要 DMA 傳輸要求時，外部輔助晶片會產生負緣觸發 DREQ[1:0]訊號。DREQ[1:0]訊號必須維持觸發狀態 4 個 MEMCLK 之久，以允許 DMA 辨別 0 至 1 的轉換。當 DREQ[1:0]訊號反觸發時，它們必須維持反觸發狀態至少 4 個 MEMCLK 之久。DMAC 紀錄從 0 至 1 的轉換來辨別是否是一個新的要求。除非前面的 DMA 資料開始傳送，否則外部輔助晶片不可觸發另一個 DREQ。

圖 7-2 DREQ 時序需求



PREQ[37:0]位元是晶片內部週邊的主動高內部訊號。不同於 DREQ[1:0]，它們是受層級影響的。在 DMAC 完成目前的資料傳輸後，才會再測試 PREQ[37:]訊號。以對晶片內部週邊的寫入要求而言，在 DMAC 傳送完寫入要求之後，它會開始測試 PREQ[37:]訊號。以讀取要求而言，在 DMAC 傳送完與內部匯流排讀取有關的最後一個位元組之後，它會開始測試 PREQ[37:0]訊號。

DCSR[REQPEND]位元指示出通道尚未決定的要求的狀態。

若一個 DREQx 觸發設定了 DMA 通道控制/狀態暫存器 DCSR[REQPEND]位元，且軟體重置 DCSR[RUN]位元來停止通道，DCSR[REQPEND]位元與內部暫存器保留的 DREQx 訊號資訊，可能在通道已經停止後仍維持設定狀態。為了重置 DCSR[REQPEND]位元，軟體必須設定一個仿造的(dummy)敘述元來傳送一些資料。

7.1.2.2 DMA_IRQ 訊號

應用處理器具有 16 個 IRQ 訊號，每一個 DMA 通道都有一個。每個 DMA IRQ 都可在 DMA 中斷暫存器 DINT 內被讀取。關於 DINT 暫存器請參考第 7-17 頁，表 7-6「DINT 暫存器位元表位元定義」。使用者可遮蔽通道的 DMA 命令暫存器 DCMD 的一些位元來引發通道的中斷，例如結束中斷致能位元 ENDIRQEN、起始中斷致能位元 STARTIRQEN 與停止中斷致能位元 STOPIRQEN。

當發生 DMA 中斷時，這個中斷在未決的中斷暫存器的位元 25 (請參考第 6-26 頁，6.2.2.5 節「中斷控制器未決暫存器(ICPR)」) 中是可見的。當一個未決中斷變為主動(active)時，若它所對應的 ICMR 遮蔽位元 25 (請參考第 6-23 頁，6.2.2.1 節「中斷控制器遮蔽暫存器 (ICMR)」) 設定為 1，則這個中斷會傳送給 CPU。

7.1.3 DMA 通道優先權排程

DMA 通道優先權排程允許需要大頻寬的週邊比需要較小頻寬的週邊有較多機會得到服務。DMA 通道內部分為 4 組。每組包含 4 個通道。在每一組中，通道間採用循環(round-robin)優先權。這 4 組 DMA 通道中，第 0 組具有最高的優先權。第 1 組的優先權則比第 2 組及第 3 組的優先權高。第 2 組和第 3 組的優先權最低。詳細內容請參考表-。大頻寬的週邊傳輸必須設定為使用第 0 組。記憶體至記憶體移動與小頻寬的週邊傳輸必須設定為使用第 2 組或第 3 組。當所有通道同時在執行時，在任何連續 8 次的通道服務中，第 0 組會被服務 4 次，第 1 組會被服務 2 次，而第 2 和第 3 組只被服務 1 次。

若 2 個以上的通道為主動且各要求一個 DMA 傳輸時，則使用表 5-2 的優先權方法。

優先權的要求不會影響已經開始執行的要求。當 DCMDx[WIDTH] 或 DCMDx[LENGTH]裡較小的資料傳輸已完成，則需要重新考慮 DMAC 優先權方法。

若所有通道都要求傳送資料，則各組的優先權順序設定如下：

- 第 0 組
- 第 1 組
- 第 0 組
- 第 2 組
- 第 0 組
- 第 1 組
- 第 0 組
- 第 3 組

此模式在下個連續 8 次的通道服務中會繼續重複。在每一組中，通道間給予循環的優先權。

表 7-2 通道優先權順序 (若所有通道同時執行)

組別	通道	優先權	被服務的次數
0	0, 1, 2, 3	最高	4/8
1	4, 5, 6, 7	次高	2/8
2	8, 9, 10, 11	低	1/8
3	12, 13, 14, 15	低	1/8

表 7-3 顯示出使用狀態器 (state machine) 來決定 DMA 通道的優先權。若非所有通道同時執行，則使用此表來決定 DMA 控制器給各個通道確切的順序。

表 7-3 通道優先權

狀態器之狀態	狀態器之各個狀態的 DMA 優先權設定
0	S0 > S1 > S2 > S3
1	S1 > S0 > S3 > S2
2	S0 > S1 > S2 > S3
3	S2 > S3 > S0 > S1
4	S0 > S1 > S2 > S3
5	S1 > S0 > S3 > S2
6	S0 > S1 > S2 > S3
7	S3 > S2 > S1 > S0

通道在每一組中會得到循環的優先權。開機之後，狀態器的狀態為 0。若通道設定為 0 且有未決的要求，則此通道會被服務。若通道設定為 1，而且具有未決的要求，則此通道會被服務，以此類推。一旦要求被服務，狀態器的狀態會增加，到最後會從狀態 7 變回狀態 0。若沒有未決的要求，狀態器會維持目前的狀態直到有未決的要求。請參考表 7-4 之優先權排程範例。

表 7-4 優先權排程範例

提出服務要求的通道	DMA 通道優先權
ch0, ch1	0, 1, 0, 1, 0, 1, 0, 1, ...
ch0, ch15	0, 0, 0, 15, 0, 0, 0, 15, ...
ch0, ch4, ch8, ch12	0, 4, 0, 8, 0, 4, 0, 12, ...
ch0, ch1, ch8, ch12	0, 1, 0, 8, 0, 1, 0, 12, ...
ch0, ch4	0, 4, 0, 0, 0, 4, 0, 4, ...
ch8, ch12	8, 12, 8, 8, 8, 12, 8, 12, ...

7.1.4 DMA 敘述元

DMAC 操作有兩種不同的模式：敘述元抓取模式 (Descriptor Fetch Mode) 與非敘述元抓取模式 (No-Descriptor Fetch Mode)。使用 DCSRx[NODESCFETCH] 位元來決定所使用的模式。

此兩種模式可同時在不同通道上使用，也就是當有些 DMA 通道在一個模式執行時，其他通道可在另一個模式執行。通道必須先停止執行，才可從一種模式換成另一種模式。

若通道內有錯誤發生，它會變成停止狀態，並且維持狀態直到軟體清除錯誤狀態且寫入 1 到 DCSR[RUN]暫存器。

7.1.4.1 非敘述元抓取模式 (No-Descriptor Fetch Mode)

在非敘述元抓取模式，DDADR_x 是保留的。軟體不可對 DDADR_x 做寫入且必須載入 DSADR_x、DTADR_x 與 DCMD_x 暫存器。當設定執行位元時，DMAC 會立刻開始傳輸資料。在傳輸的剛開始，會執行非敘述元抓取。當傳輸結束，通道就會停止。

須確定軟體不可以程式化通道的 DDADR_x 非敘述元抓取模式。

以下是典型的非敘述元抓取模式 (DCSR[NODESCFETCH] = 1) 操作：

1. 在重置之後，通道為未初始化狀態。
2. DCSR[RUN]位元設定為 0，DCSR[NODESCFETCH]位元設定為 1。
3. 軟體將來原位址寫入 DSADR 暫存器，將目標位址寫入 DTADR 暫存器，將指令寫入 DCMD 暫存器。在非敘述元抓取模式中，DDADR 為保留的，而且不可寫入 DDADR。
4. 軟體將 DCSR[RUN]位元寫入 1，執行非敘述元抓取。
5. 由 DCMD[FLOW]來源和目標位元決定通道等待要求或是開始傳輸資料。
6. 通道傳輸位元數等於 DCMD[SIZE]和 DCMD[LENGTH] 的最小值。
7. 通道等待下一個要求或繼續傳輸資料直到 DCMD[LENGTH]為 0。
8. DDADR[STOP]設定為 1 則通道停止。

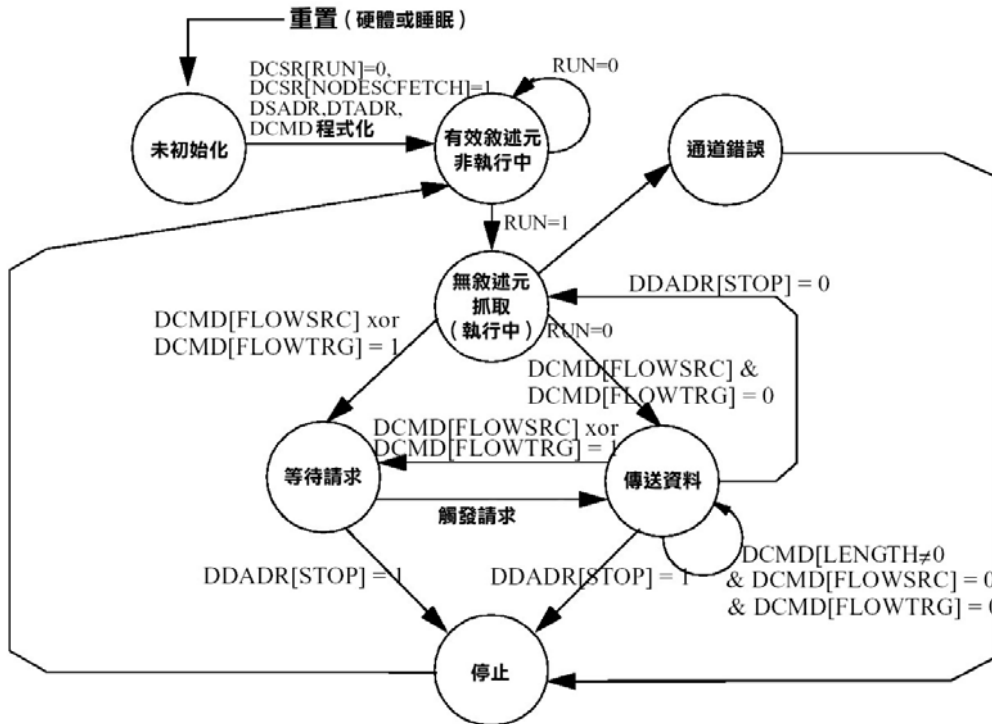


圖 7-3 非敘述元抓取模式通道狀態

7.1.4.2 敘述元抓取模式 (Descriptor Fetch Mode)

在敘述元抓取模式中，DMAC 暫存器由主記憶體裡的 DMA 敘述元載入。多個 DMA 敘述元可以串連在一起形成一個串列。允許 DMA 通道傳輸資料到一些不連續的位置，或從一些不連續的位置傳輸資料到目的。此敘述元協定設計允許敘述元加入到正在執行的 DMA 敘述元串列中。

以下是典型的敘述元抓取模式 (DCSR[NODESCFETCH]=0) 操作：

1. 重置之後，通道為未初始化狀態。
2. 軟體對 DDADR 暫存器寫入一個敘述元位址 (邊界為 16 位元)。
3. 軟體對 DCSR[RUN]位元寫入 1。
4. DMAC 從 DDADR 所指示的記憶體中抓取 4 字組敘述元 (假設已設定記憶體和敘述元鍊)。
5. 在主記憶體邊界為 16 位元組的 4 字組 DMA 敘述元載入以下的暫存器：
 - a. Word[0] → DDADR_x 暫存器和單一的旗標位元。指到下一個 4 字組的敘述元。
 - b. Word[1] → DSADR_x 暫存器。
 - c. Word[2] → DTADR_x 暫存器。
 - d. Word[3] → DCMD_x 暫存器。
6. DCMD[FLW]來源和目標位元決定通道等待要求或是開始傳輸資料。

7. 通道依據 DCMD[SIZE]與 DCMD[LENGTH]的最小值相同的位元組數目進行傳輸。
8. 通道等待下一個要求或繼續傳輸資料直到 DCMD[LENGTH]為 0。
9. DDADR[STOP]位元決定停止通道或繼續從記憶體抓取新的敘述元。

在 DMA 敘述元 (DDADR_x 區域的低位元) Word[0]的 Bit[0] (STOP) 註記是否該敘述元是在敘述元串列的最末端。STOP 位元的值不會影響通道暫存器載入敘述元字組欄位的方式。若一個敘述元與其 STOP 位元被載入到通道暫存器中，在完成與敘述元相關的資料傳輸之後，通道會停止。第 5-8 頁，圖 5-4「敘述元抓取模式通道狀態」有對於此操作的概述。

在軟體載入 DDADR 之後，必須把 DCSR[RUN]位元設定為 1。除非載入 DDADR 暫存器，並且將 DCSR[RUN]位元設定為 1，否則不會執行通道敘述元抓取。

DMAC 優先權方法不會影響 DMA 敘述元抓取。在前一個敘述元執行之後，會緊接著抓取下一個敘述元。

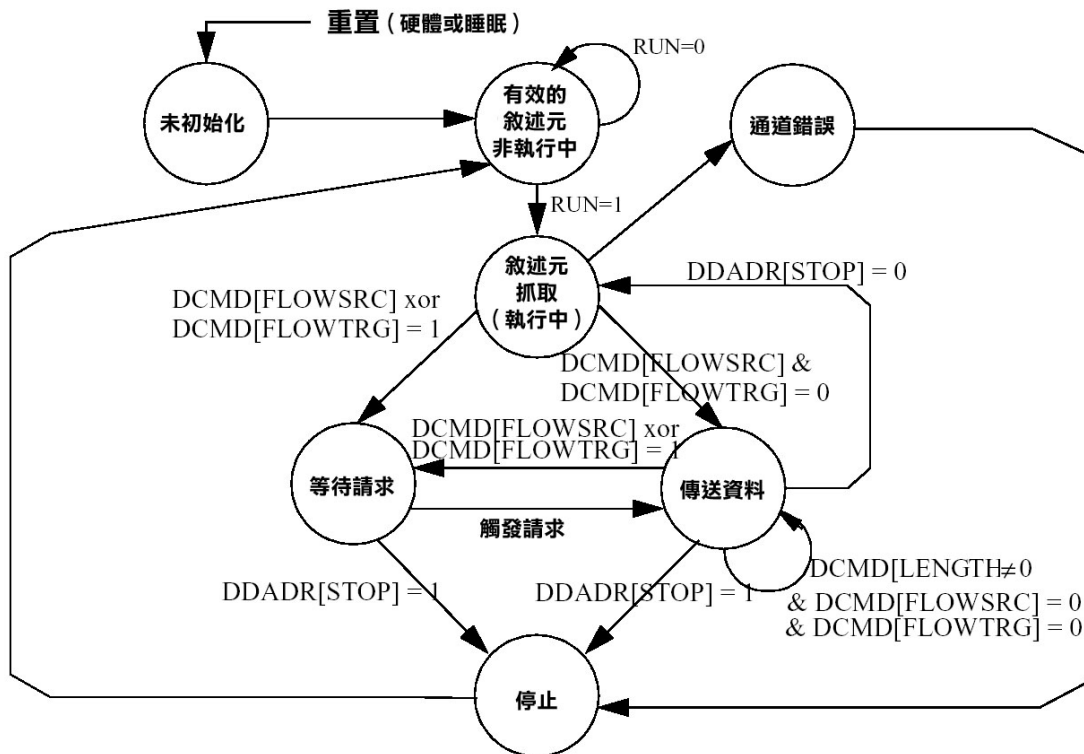


圖 7-4 敘述元抓取模式通道狀態

7.1.5 通道狀態

DMA 通道狀態可經過以下的任何狀態：

- 未初始化 (Uninitialized)：重置之後通道為未初始化狀態。
- 有效敘述元，非執行 (Valid Descriptor, Not Running)：在敘述元抓取模式裡，軟體載入敘述元到通道的 DDADR 中；在非敘述元抓取模式裡，程式化 DSADR、DTADR 與 DCMD 數值。但是 DCSR[RUN]裡對應的執行位元不會被設定為 1。
- 敘述元抓取，執行 (Descriptor Fetch, Running)：從記憶體抓取敘述元的 4 個字組。
- 等待要求 (Wait for Request)：在通道開始傳送資料之前，它會先等待要求。
- 傳送資料 (Transfer Data)：通道正在傳送資料。
- 通道錯誤 (Channel Error)：通道發生錯誤。它會維持停止狀態直到軟體清除錯誤條件，重新初始化通道，並且寫入 1 到 DCSR[RUN]位元。詳細內容請參考 7.3.1 與 7.3.2 節。
- 停止 (Stopped)：通道已停止。

圖 7-3 與 7-4 顯示出狀態之間的進展。

7.1.6 讀取與寫入順序

DMAC 確保資料不會保留在敘述元之間的每個通道緩衝區內。當一個敘述元被處理完成，通道內任何緩衝的讀取資料會被丟棄，而任何的寫入資料則送至記憶體（但資料可能尚未在通道內）。直到敘述元被完全處理後，才會發佈 DMA 中斷。

7.1.7 位元組傳送順序

DCMD[ENDIAN]位元指示當資料從記憶體內讀取或寫入至記憶體時，字組內的位元組順序。詳細內容請參考第 7-10 頁，圖 7-5。DCMD[ENDIAN]位元必須設定為 0，為 Little Endian 傳輸。

第 7-10 頁，圖 7-5「Little Endian 傳輸」顯示出由 DCMD[ENDIAN]與 DCMD[SIZE]位元所決定的傳送資料的順序。

若資料自內部裝置傳送至記憶體，DCMD[ENDIAN]位元設定為 0，DCMD[SIZE]設定為 1，記憶體接收資料的順序如下：

1. Byte[0]
2. Byte[1]

3. Byte[2]
4. Byte[3]

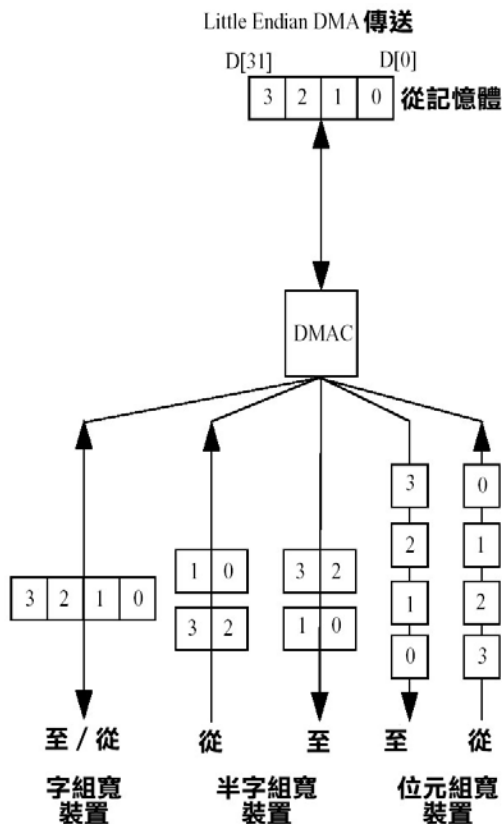


圖 7-5 Little Endian 傳輸

7.1.8 尾端位元組

一般 DMA 傳輸的位元組大小和由 DCMD[SIZE]所指定的處理大小相同。當敘述元處理資料的尾端時，在 DCMD[LENGTH]區域裡的尾端位元組的數目也許會小於傳輸的大小。若 DCMD[FLWOSRC]和 DCMD[FLWOTRG]位元都是 0，或是從內部週邊或輔助晶片接收到對應的要求，則 DMA 可以傳輸確切的尾端位元組數。

尾端位元組必須考慮到以下的情況：

- 記憶體至記憶體移動 (Memory-to-Memory Moves)：DMA 傳輸的位元組數目大小等於 DCMD[LENGTH]或 DCMD[SIZE]的最小值。
- 輔助晶片相關傳輸 (Companion-Chip Related Transfers)：若 DMAC 必須處理尾端位元組，則輔助晶片會觸發要求。若要求被觸發，DMA 傳輸的位元組數等於 DCMD[LENGTH]或 DCMD[SIZE]的最小值。
- 記憶體對內部週邊傳輸 (Memory to Internal Peripheral Transfers)：在記憶體

對內部週邊傳輸期間，大部分的週邊會送一個要求給尾端位元組。週邊操作的細節請參考其他章節。DMA 傳輸位元組等於 DCMD[LENGTH] 或 DCMD[SIZE] 的最小值。

- 內部週邊至記憶體傳輸 (Internal Peripheral to Memory Transfers)：在 On-chip 週邊至記憶體傳輸時，大部分的週邊都不會送要求給尾端位元組。關於週邊操作的詳細內容，請參考本文件其他適當的章節。若週邊傳送出一個要求，DMA 所傳送的位元組數目等於 DCMD[LENGTH] 或 DCMD[SIZE] 的最小值。若軟體必須使用程式化 I/O 來處理尾端位元組，則必須依照以下的操作順序：
 1. 將 0 寫入至 DCSR[RUN] 位元來停止 DMA 通道。
 2. 等待直到通道停止。
 3. 讀取通道的暫存器來檢查通道的狀態。
 4. 執行程式化 I/O 傳送到週邊(非透過 DMA)。
 5. 設定 DCSR[RUN] 位元為 1 並重置 DMA 通道，以使 DMA 通道能進行未來的資料傳輸。

7.2 傳送資料

經由週邊匯流排以及使用 flow-through 資料傳輸方式，將內部週邊連接至 DMAC。DMAC 也可以 flow-through 傳輸模式來傳輸資料到任何記憶體位址與記憶體至記憶體移動。直接連接至外部資料腳位的外部裝置（例如：輔助晶片），必須使用 flow-through 資料傳輸方式。

除了寫入至快閃記憶體之外，主記憶體包含應用處理器所支援的任何記憶體。Intel XScale 處理器不支援寫入快閃記憶體，若寫入快閃記憶體，會引起一個匯流排錯誤。

在 flow-through 傳輸模式中，在資料被目的緩衝區／記憶體保留之前會先經過 DMAC。在 flow-through 傳輸模式下，DMAC 也可執行記憶體至記憶體的移動。

7.2.1 服務內部週邊

DMAC 提供 DMA 要求至通道對應暫存器 (Channel Map Register) (DRCMRx)，對每個可能的 DMA 要求，通道對應暫存器包含 4 個用來指定通道號碼的位元。內部週邊可被對應到 16 個可使用的通道的任何一個。請參考表

7-5「內部週邊 DMA 快速參考表」來設定內部週邊組態作 DMA 存取。內部週邊經由週邊要求匯流排 (PREQ) 來觸發要求位元。來自 PREQ 的訊號在每個週邊時脈 (PCLK) 都會被測試，若有任何一個 PREQ 訊號不為 0，則根據 DRCMRx 對應的位元執行一個查詢的動作。

內部週邊位址若在 DSADR 內，則 DCMDx[FLWSRC]位元必須設定為 1。這允許應用處理器在開始傳送之前等待要求。內部週邊位址若在 DTADR 裡，則 DCMDx[FLOWTRG]位元必須設定為 1。

若 DCMD[IRQEN]被設定為 1，假如傳輸中的一個位元組導致 DCMDx[LENGTH]減少為 0，在該位元組傳輸的最後一個週期結束時會要求一個 DMA 中斷。

7.2.1.1 使用 Flow-Through DMA 讀取週期來服務內部週邊

當內部週邊經 PREQ 匯流排傳送一個要求給 DMAC 通道，而 DMAC 正在執行且設定組態為 flow-through 讀取時，內部週邊的 flow-through DMA 讀取就會開始執行。傳輸的位元組數目由 DCMDx[SIZE]指定，當要求為最高優先權的要求時，會開始進行以下的過程：

1. DMAC 送出一個要求給記憶體控制器，將由 DSADR_x[31:0]定址的位元的數目讀取至 DMAC 內部的一個 32 位元組的臨時緩衝區。
2. DMAC 傳送資料給定址在 DTADR_x[31:0]的 I/O 設備。DCMD[WIDTH]指定內部週邊資料傳輸的寬度。
3. 傳輸的最後，DSADR_x 值會增加，增加的值即為 DCMDx[LENGTH]和 DCMD[SIZE]的兩者中的較小值。DCMDx[LENGTH]也會減少相同的值。

內部週邊的 flow-through DMA 讀取使用以下的內容來設定 DMAC 暫存器位元：

- DSADR[SRCAADDR] = 外部記憶體位址
- DTADR[TRGADDR] = 內部週邊位址
- DCMD[INCSRCAADDR] = 1
- DCMD[FLWSRC] = 0
- DCMD[FLOWTRG] = 0

7.2.1.2 使用 Flow-Through DMA 寫入週期來服務內部週邊

當內部週邊經 PREQ 匯流排傳送一個要求給 DMAC 通道，而 DMAC 正在執行且設定組態為 flow-through 寫入時，內部週邊的 flow-through DMA 讀取就會開始執行。傳輸的位元組數目由 DCMDx[SIZE]指定，當送出最高優先權的要求

時，會開始進行以下的過程：

1. DMAC 從 DSADR_x[31:0]所定址的 I/O 設備傳送所需要的位元組數到 DMAC 寫入緩衝區。
2. DMAC 經由內部匯流排傳送資料給記憶體控制器。DCMD[WIDTH]指定內部週邊資料傳輸的速度。
3. 在傳輸的最後，DTADR_x 值會增加，增加的值即為 DCMD_x[LENGTH]和 DCMD[SIZE]的兩者中的較小值。DCMD_x[LENGTH]也會減少相同的值。

內部週邊的 flow-through DMA 寫入使用以下的內容來設定 DMAC 暫存器位址：

- DSADR[SRCADDDR] = 內部週邊位址
- DTADR[TRGADDR] = 外部記憶體位址
- DCMD[INCTRGADDR] = 1
- DCMD[FLWSRC] = 1
- DCMD[FLOWTRG] = 0

7.2.2 DMA 程式設計快速參考表

表 7-5 為 DMA 程式設計的快速參考表。

表 7-5 內部週邊 DMA 快速參考表

單元	功能	FIFO 位址	寬度 (bytes)	DCMD 寬度 (binary)	突發傳輸 大小 (bytes)	來源 / 目的	DRCMR
I2S	接收	0x40400080	4	11	8, 16, 32	來源	0x40000 0108
	傳送	0x40400080	4	11	8, 16, 32	目的	0x40000 010C
BTUART	接收	0x40200000	1	01	8, 16, 32	來源	0x40000 0110
	傳送	0x40200000	1	01	8, 16, 32 或尾端	目的	0x40000 0114
FFUART	接收	0x40100000	1	01	8, 16, 32	來源	0x40000 0118
	傳送	0x40100000	1	01	8, 16, 32 或尾端	目的	0x40000 011C
AC97	麥克風	0x40500060	4	11	8, 16, 32	來源	0x40000 0120
	數據機 接收	0x40500140	4	11	8, 16, 32	來源	0x40000 0124
	數據機 傳送	0x40500140	4	11	8, 16, 32	目的	0x40000 0128

	聲音接收	0x40500040	4	11	8, 16, 32	來源	0x40000 012C
	聲音傳送	0x40500040	4	11	8, 16, 32	目的	0x40000 0130
SSP	接收	0x41000010	2	10	8, 16	來源	0x40000 0134
	傳送	0x41000010	2	10	8, 16	目的	0x40000 0138
FICP	接收	0x4080000 C	1	01	8, 16, 32	來源	0x40000 0144
	傳送	0x4080000 C	1	01	8, 16, 32 或尾端	目的	0x40000 0148
STUART	接收	0x4070000 C	1	01	8, 16, 32	來源	0x40000 014C
	傳送	0x4070000 C	1	01	8, 16, 32 或尾端	目的	0x40000 0150
MMC	接收	0x41100040	1	01	32 或尾端	來源	0x40000 0154
	傳送	0x41100040	1	01	32 或尾端	目的	0x40000 0158
USB	端點 1 傳送	0x40600100	1	01	32	目的	0x40000 0164
	端點 2 接收	0x40600180	1	01	32	來源	0x40000 0168
	端點 3 傳送	0x40600200	1	01	32	目的	0x40000 016C
	端點 4 接收	0x40600400	1	01	32	來源	0x40000 0170
	端點 6 傳送	0x40600600	1	01	32	目的	0x40000 0178
	端點 7 接收	0x40600680	1	01	32	來源	0x40000 017C
	端點 8 傳送	0x40600700	1	01	32	目的	0x40000 0180
	端點 9 接收	0x40600900	1	01	32	來源	0x40000 0184
	端點 11 傳送 1	0x40600B0 0	1	01	32	目的	0x40000 018C
	端點 12 接收	0x40600B8 0	1	01	32	來源	0x40000 0190

端點 13 傳送	0x40600C0 0	1	01	32	目的	0x40000 0194
端點 14 接收	0x40600E0 0	1	01	32	來源	0x40000 019C

7.2.3 服務輔助晶片和外部週邊

輔助晶片和外部週邊以 flow-through 方式傳輸。DMAC 對通道映象暫存器發出 DMA 要求。通道映象暫存器包含 4 個位元，這 4 個位元指定一個通道號碼給每個可能的 DMA 要求。輔助晶片的要求為 DREQ[1:0]。DREQ 訊號可對應到 16 個可使用的通道之一。DREQ 在每個週邊時脈 (PCLK) 都會測試，而若有任何 DREQ 訊號測試結果不為 0，則 DRCMRx 裡的對應的位元會做檢查。這允許其中之一的通道要求被對應。若外部週邊位址是在 DSADR 或 DTADR 內，則 DCMDx[FLWSRC]位址必須設定為 1。這允許應用處理器在開始傳輸之前先等待要求。

若 DCMDx[IRQEN]設定為 1，在最後一個週期的尾端，會要求一個 DMA 中斷。最後週期是和可使 DCMDx[ENGTH]由 1 減少為 0 的位元組有關。

7.2.3.1 使用 Flow-Through DMA 讀取週期來服務外部週邊

當外部週邊經 DREQ[1:0]匯流排傳送一個要求給 DMAC 通道，而 DMAC 正在執行且設定組態為 flow-through 讀取時，外部週邊的 flow-through DMA 讀取就會開始執行。傳輸的位元組數目由 DCMDx[SIZE]指定，當送出最高優先權的要求時，會開始進行以下的過程：

1. DMAC 送出一個要求給記憶體控制器，將由 DSADR_x[31:0]定址的位元的數目讀取至 DMAC 內部的一個 32 位元組的臨時緩衝區。
2. DMAC 傳輸緩衝區內的資料給定址在 DTADR_x[31:0]的外部設備。
3. 傳輸的最後，DTADR_x 值會增加，增加的值即為 DCMDx[LENGTH]和 DCMD[SIZE]的兩者中的較小值。DCMDx[LENGTH]也會減少相同的值。

注意：外部週邊 flow-through DMA 讀取的行程指示出外部位址增加。一些外部週邊，例如：FIFO，則不需要增加外部位址。

內部週邊的 flow-through DMA 讀取使用以下的內容來設定 DMAC 暫存器位元：

- DSADR[SRCAADDR] = 外部記憶體位址

- DTADR[TRGADDR] = 輔助晶片位址
- DCMD[INCSRCADDR] = 1
- DCMA[INCTRGADDR] = 0
- DCMD[FLWSRC] = 0
- DCMD[FLOWTRG] = 1

7.2.3.2 使用 flow-through DMA 寫入來服務外部週邊

當外部週邊經 DREQ[1:0] 匯流排傳送一個要求給 DMAC 通道，而 DMAC 正在執行且設定組態為 flow-through 讀取時，外部週邊的 flow-through DMA 讀取就會開始執行。傳輸的位元組數目由 DCMDx[SIZE] 指定，當送出最高優先權的要求時，會開始進行以下的過程：

1. DMAC 從 DSADR_x[31:0] 所定址的 I/O 設備傳送所需要的位元組數到 DMAC 寫入緩衝區。
2. DMAC 經內部匯流排傳送資料給記憶體控制器。
3. 在傳輸的最後，DTADR_x 值會增加，增加的值即為 DCMDx[LENGTH] 和 DCMD[SIZE] 的兩者中的較小值。DCMDx[LENGTH] 也會減少相同的值。

注意：外部週邊 flow-through DMA 讀取的行程指示出外部位址增加。一些外部週邊，例如：FIFO，則不需要增加外部位址。

對於一個 flow-through DMA 寫入至外部週邊，使用以下的內容來設定 DMAC 暫存器位元：

1. DSADR[SrcADDR] = 輔助晶片位址
2. DTADR[TRGADDR] = 外部記憶體位址
3. DCMD[INCSRCADDR] = 0
4. DCMD[INCTRGADDR] = 1
5. DCMD[FLWSRC] = 1
6. DCMD[FLOWTRG] = 0

7.2.4 記憶體至記憶體移動

記憶體至記憶體的移動不會引起 DREQ 與 PREQ 要求訊號。處理器寫入至 DCSR[RUN] 位元，並且設定一個通道做記憶體至記憶體移動。DCMDx[FLWSRC] 與 DCMD[FLOWTRG] 位元必須設定為 0。

若 DCMD[IRQEN] 被設定為 1，在最後一個週期會引發一個 DMA 要求，而此週期與引起 DCMDx[LENGTH] 從 1 減少為 0 的位元組有關。

下列為一個 flow-through DMA 記憶體至記憶體讀取與寫入之步驟：

1. 處理器寫入至 DCSR[RUN]暫存器位元，並且開始做記憶體至記憶體的移動。
2. 若應用處理器為敘述元抓取模式，通道會設定組態來抓取 4 字組的敘述元。通道傳送資料不會等待 PREQ 與 DREQ 被觸發。DCMDx[SIZE] 與 DCMDx[LENGTH]兩者中較小的值即為被傳送的位元組數目。
3. DMAC 傳送要求至記憶體控制器，將由 DSADR_x[31:0]定址的位元的數目讀取至 DMAC 內部的一個 32 位元組的臨時緩衝區。
4. DMAC 產生一個寫入週期至 DTADR_x[31:0]內定址的位置。
5. 在傳送的最後，DSADR_x 與 DTADR_x 會增加，而增加的值即為 DCMD[SIZE] 與 DCMDx[LENGTH]兩者中較小的值。

注意：記憶體至記憶體傳送的進展指示出外部位址增加。某些外部週邊，例如 FIFO，並不需要增加外部位址。

7.3 DMAC 暫存器

此章節在說明 DMAC 暫存器。

7.3.1 DMA 中斷暫存器

此唯讀的 DMA 中斷暫存器 (DINT) (圖 5-6) 記錄每一個通道的中斷。

若以下任何一個事件發生，則會產生中斷：

- 與相關通道的相關內部匯流排發生任何類型的執行錯誤。
- 目前的傳送順利完成，且 DCMD:ENDIRQEN 位元設定為 1。
- 目前的敘述元成功載入，且 DCMD:STRATIRQEN 位元設定為 1。
- DCSR:STOPIRQEN 設定為 1，且相關通道處在未初始化或停止的狀態。

軟體必須將 DCSR 暫存器錯誤位元寫入為 1 來重置中斷。

表 7-6 DINT 暫存器位元表位元定義

實體位址 0x4000_00F0		DMA 中斷暫存器 (DINT)																DMA															
位元	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	保留																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																		ChIntr15	ChIntr14	ChIntr13	ChIntr12	ChIntr11	ChIntr10	ChIntr9	ChIntr8	ChIntr7	ChIntr6	ChIntr5	ChIntr4	ChIntr3	ChIntr2	ChIntr1	ChIntr0
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	說明
31:16	保留	讀取為未知，必須寫入為 0。
15:0	CHLINTR _x	通道「x」中斷（唯讀） 0 = 沒有中斷 1 = 中斷

7.3.2 DMA 通道控制／狀態暫存器

讀取與寫入 DMA 通道控制／狀態暫存器 (DCSR_x) (圖 5-7) 包含每個通道的控制與狀態位元。讀取此暫存器可找出中斷的來源。將讀取的值寫回暫存器可清除中斷。

表 7-7 DMA 通道控制／狀態暫存器位元表的位元定義

實體位址 0x4000_0000 - 0x4000_003C		DMA 通道控制 / 狀態暫存器 (DCSR _x)																DMA										
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																REQPEND	保留				STOPSTATE	ENDINTR	STARTINTR	BUSERRINTR		
																		RUN	NODESCFETCH	STOPIRQEN								
重置	0 0	0																0	0	0	0	0	0	0	0	0		

位元	名稱	說明
31	RUN	執行位元（讀取／寫入） 0 = 停止此通道 1 = 啟動此通道 軟體啟動或停止通道。如果在突發傳輸中，執行位元被清除，則突發傳輸會在通道停止前完成。 在軟體設定此位元為敘述元抓取模式之前，軟體必須寫入 DDADR _x 。 通道停止之後，DCSR[STOPSTATE]位元設定為 1。 軟體必須查詢 DCSR[STOPSTATE]位元以決定通道的狀態，或在通道停止後設定 STOPIRQEN 引發中

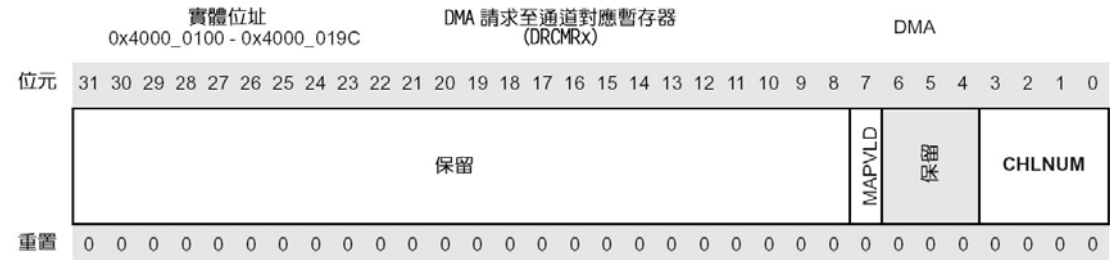
		<p>斷。若要重新啟動停止的通道，軟體必須將位元寫入 1。</p> <p>清除執行位元後，通道會停止。若長度位元 DCMD[LENGTH]為 0，則結束中斷不保證一定會發生。在執行位元清除後，軟體決定是否傳輸。</p>
30	NODESCFETCH	<p>非敘述元抓取（讀／寫）</p> <p>0 = 敘述元抓取模式</p> <p>1 = 非敘述元抓取模式</p> <p>決定頻道是否擁有敘述元。</p> <p>若此位元設定為 0，則通道為敘述元抓取模式。DMAC 暫存器的資訊請參考第 5-7 頁，5.1.4.2 節「敘述元抓取模式」。</p> <p>若此位元設定為 1，則通道為非敘述元抓取模式。DMAC 暫存器的資訊請參考第 5-6 頁，5.1.4.2 節「敘述元抓取模式」。</p>
29	STOPIRQEN	<p>停止中斷啟動（讀取／寫入）</p> <p>0 = 若通道是非初始化或停止狀態，則無中斷。</p> <p>1 = 若通道是未初始化或停止狀態，則發生中斷。</p> <p>若 DCSR[STOPSTATE]為 1，則允許中斷傳至中斷控制器；若為 0，則在通道停止會不會發生中斷。若軟體在通道啟動前將此位元寫入為 1，則會發生中斷。</p>
28:9	保留	讀取為未知，而必須寫入為 0。
8	REQPEND	<p>要求未決定（唯讀）</p> <p>0 = 無未決的要求</p> <p>1 = 通道有未決的要求</p> <p>指示 DMA 通道有未決的要求。</p>
7:4	保留	讀取為未知，而必須寫入為 0。
3	STOPSTATE	<p>停止狀態（唯讀）</p> <p>0 = 執行通道</p> <p>1 = 通道為未初始化或停止狀態。</p> <p>若通道為未初始或停止狀態，狀態位元會被設定。若 DCSR[STOPIRQEN]設定為 1，DMAC 會產生中斷。關於通道狀態請參考第 5-9 頁，5.1.5 節「通道狀態」。</p> <p>軟體必須重新程式化 DDADR，以對 DCSR[RUN]寫入 1 來重新啟動通道和清除此位元。</p> <p>軟體對 DCSR[STOPIRQEN]寫入 0 來重置中斷。</p>

2	ENDINTR	<p>結束中斷（讀取／寫入）</p> <p>0 = 無中斷</p> <p>1 = 產生中斷，因為目前的處理成功完成且 DCMD[LENGTH]=0。</p> <p>發生中斷必須設定 DCMD[ENDIRQEN]位元。軟體必須對此位元寫入 1，重置對應的中斷。對位元寫入為 0 則無效。</p>
1	STARTINTR	<p>開始中斷（讀取／寫入）</p> <p>0 = 無中斷</p> <p>1 = 產生中斷，因為成功抓取敘述元。</p> <p>發生中斷時必須設定 DCMD[STARTIRQEN]位元。軟體必須對此位元寫入 1，重置對應的中斷。對位元寫入為 0 則無效。</p>
0 = 0	BUSERRINTR	<p>匯流排錯誤中斷（讀取／寫入）</p> <p>0 = 無中斷</p> <p>1 = 匯流排錯誤造成中斷</p> <p>指示在資料傳輸時有發生錯誤。當通道有壞的敘述元、來源或目標位址時，傳輸資料會發生錯誤。當位址指向 non-busable 位置或保留空間時，位址會被視為是壞的。軟體必須對此位元寫入 1，重置對應的中斷。對位元寫入為 0 則無效。只有一個錯誤會影響記錄的每個頻道。造成錯誤的通道在傳輸的最後會更新，並且記錄錯誤直到被重新程式化與設定對應的位元。之後通道為可存取的。</p>

7.3.3 DMA 要求至通道映象暫存器（Channel Map Register）

對通道映象暫存器（DRCMR_x）（圖 7-8）的讀取／寫入 DMA 要求，通道映象暫存器會對映每個 DMA 要求到一個通道。詳細內容請參考表 7-13。

表 7-8 DRCMR_x 暫存器位元表的位元定義



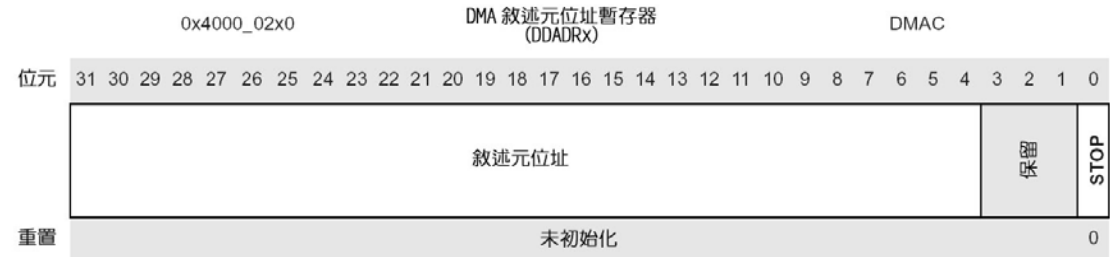
位元	名稱	說明
31:8	保留	讀取為未知，而必須寫入為 0。
7	MAPVLD	映象有效（讀取／寫入） 0 = 要求沒有被對應 1 = 要求被對應至 DRCMRx[3:0]所指示的通道。 決定要求是否被對應。若位元被設定為 1，則要求被對應至 DRCMRx[3:0]所指示的通道。若此位元為 0，則要求不會被對應。此位元也可用來做遮蔽要求之用。
6:4	保留	讀取為未知，而必須寫入為 0。
3:0	CHLNUM	通道數目（讀取／寫入） 若 DRCMR[MAPVLD]設定為 1，則指示出通道數目。不可將 2 個有效的要求對應至相同的通道，否則會產生無法預期的結果。請參考第 5-4 頁，5.1.3 節「DMA 通道優先權排程」來複習通道優先權排程。

7.3.4 DMA 敘述元位址暫存器

DMA 敘述元位址暫存器 (DDADR_x) (請參考第 5-12 頁，表 5-9 「DMA 敘述元位址暫存器位元定義」) 包含特定通道的下一個敘述元的記憶體位址。在啟動電源時，此暫存器內的位元是未定義的。位址必須對齊 16 位元組的邊界。這表示位址的位元[3:1]是保留的，且必須讀取與寫入為 0。DDADR 不可包含任何其他內部週邊暫存器或 DMA 暫存器的位址。

若通道為非敘述元抓取模式，則 DDADR 是保留的。

表 7-9 敘述元位址暫存器位元定義



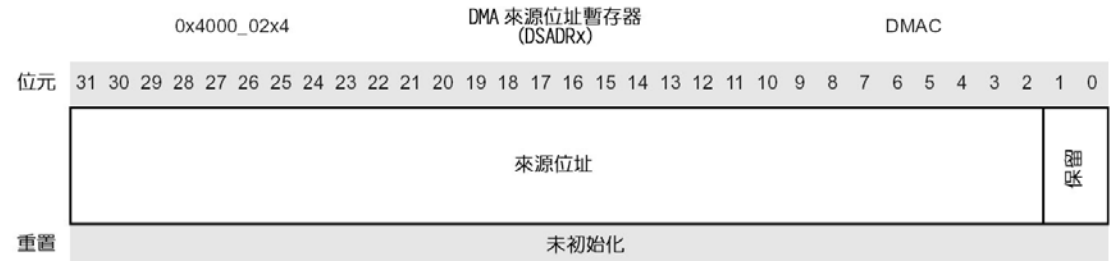
位元	名稱	說明
31:4	DESCRIPTOR ADDRESS	下一個敘述元的位址（讀取／寫入）。
3:1	保留	保留。讀取為未知，而必須寫入為 0。
0	STOP	停止（讀取／寫入） 0 = 執行通道 1 = 在此敘述元完成執行後和抓取下個敘述元之前停止通道。例如：DCMD[LENGTH]=0。 若設定此位元，在通道完成執行後和抓取下個敘述元之前，通道會停止。若 DDADRx[STOP]位元為 0，當目前敘述元完成執行時，會開始抓取以運用 DDADR 的新敘述元。

7.3.5 DMA 來源位址暫存器 (DSADR_x)

在敘述元抓取模式，DSADR_x 只可以讀取；在非敘述元抓取模式，可做讀取和寫入。

DSADR_x (請參考圖 7-10) 包括特定通道目前的敘述元來源位址。來源位址是內部週邊或記憶體位址的位元。在啟動電源時，此暫存器的位元是未定義的。若來源位址是輔助晶片或外部週邊的位址，則來源位址的邊界必須為 8 位元組。這允許位址的[2:0]位元被保留。若來源位址為內部的位址，則位址的邊界必須為 32 位元，並且保留[1:0]位元。DSADR 不能包括任何其他內部 DMA、LCD 或 MEMC 暫存器的位址。

表 7-10 DSADR_x 暫存器位元表位元定義



位元	名稱	說明
31:3	SRCADDR	來源位址 (讀取/寫入) 內部週邊或記憶體位置的位址。 記憶體位置的位址做輔助晶片傳輸。
2	SRCADDR	來源位址位元 2 若 DSADR.SrcAddr 為外部記憶體位置，則保留。 若 DSADR.SrcAddr 為內部記憶體位置，則不保留(讀/寫)。
1:0	保留	讀取為未知，而必須寫入為 0。

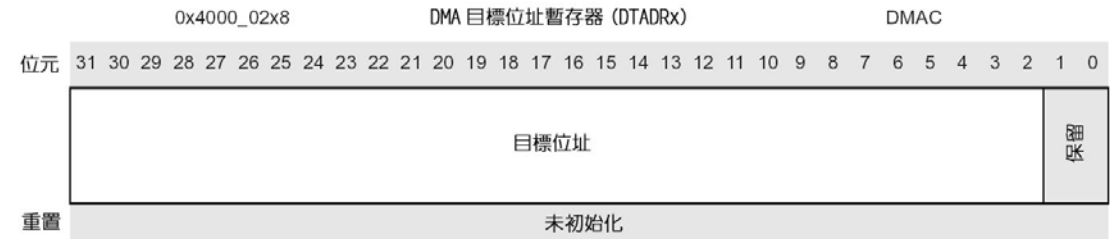
7.3.6 DMA 目標位址暫存器 (DTADR_x)

對軟體而言，DTADR_x (圖 7-11) 在敘述元抓取模式時只可讀取，而在非敘述元抓取模式時可以讀取及寫入。

這些暫存器包含目前通道內的敘述元的目標位址。目標位址為內部週邊或記憶體位置的位址。在啟動電源時，暫存器內的位元是未定義的。若目標位址為輔助晶片或外部週邊的位址，則目標位址必須對齊 8 位元組的邊界。這允許位址的位元[2:0]被保留。若目標位元為內部週邊的位址，則位址必須對齊 32 位元，並保留位元[1:0]。DTADR 不可包含任何內部 DMA、LCD 或 MEMC 暫存器的位址。

DTADR 不可包含快閃記憶體的位址，因為不支援從 DMAC 寫入至快閃記憶體。

表 7-11 DTADR_x 暫存器位元表位元定義



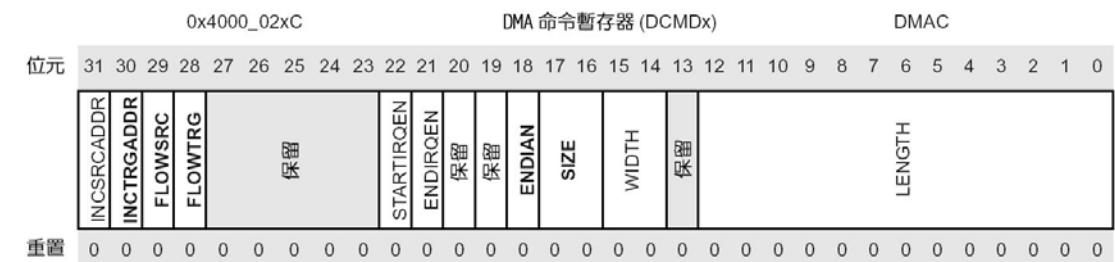
位元	名稱	說明
31:3	TRGADDR	目標位址 (讀取/寫入)： 晶片週邊或是記憶體位置的位址。 輔助晶片傳送的記憶體位置的位址。
2	TRGADDR	目標位址位元 2 若 DTADR.TrigAddr 為外部記憶體位置，則保留。 若 DTADR.TrigAddr 為內部週邊 (讀/寫)，則不保留。
1:0	保留	讀取為未知，而必須寫入為 0。

7.3.7 DMA 命令暫存器 (DCMDx)

對軟體而言，DCMDx (圖 7-12) 在敘述元抓取模式下只可讀取，而在非敘述元抓取模式下則可讀取及寫入。

這些暫存器包含通道的控制位元與目前通道內的傳輸長度。在啟動電源時，此暫存器內的位元會被設定為 0。

表 7-12 DCMDx 暫存器位元表位元定義



位元	名稱	說明
31	INCSRCADR	來源位址增加設定 (讀取/寫入) 0 = 不增加來源位址 1 = 增加來源位址，並在每個內部匯流排處理的最後由 DCMD[SIZE]來啟動 若來源位址為內部週邊的 FIFO 位址，或為外部 IO

		位址，則在每個成功的存取時不會增加位址。在此情況下，此位元必須為 0。
30	INCTRGADDR	目標位址增加設定（讀取／寫入） 0 = 不增加目標位址。 1 = 增加目標位址，並在每個內部匯流排處理的最後由 DCMD[SIZE]來啟動。 若目標位址為內部週邊的 FIFO 位址，或為外部 IO 位址，則在每個成功的存取時不會增加位址。在此情況下，此位元必須為 0。
29	FLWSRC	來源流程控制（讀取／寫入） 0 = 立即開始傳輸資料。 1 = 在開始傳輸資料前先等待要求訊號。 說明來源的流量重。若來源是 on-chip 或外部週邊，則此位元必須為 1。 若 DCMD[FLWSRC]或 DCMD[FLOWTRG]位元被設定，則 DMA 不能傳輸直到他收到要求為止。不可同時將 DCMD[FLWSRC]和 DCMD[FLOWTRG]設定為 1。
28	FLOWTRG	目標流程控制（讀取／寫入） 0 = 立即開始傳輸資料。 1 = 在開始傳輸資料前先等待要求訊號。 說明目標的流量重。若目標是 on-chip 或外部週邊，則此位元必須為 1。 若 DCMD[FLWSRC]或 DCMD[FLOWTRG]位元被設定，則 DMA 不能傳輸直到他收到要求為止。不可同時將 DCMD[FLWSRC]和 DCMD[FLOWTRG]設定為 1。
27:23	保留	讀取為未知，而必須寫入為 0。
22	STARTIRQEN	開始中斷啟動（讀取／寫入） 0 = 無中斷產生。 1 = 當通道的敘述元被載入時，允許中斷通過。 當敘述元被載入時，設定通道的 DCSR[StartIntr]中斷。
21	ENDIRQEN	結束中斷啟動（讀取／寫入） 0 = 無中斷產生 1 = 當 DCMD[LENGTH]減少至 0 時，設定通道的 DCSR[EndIntr]中斷。 說明資料傳輸一完成，就啟動中斷。

20	保留	讀取為未知，而必須寫入為 0。
19	保留	讀取為未知，而必須寫入為 0。
18	ENDIAN	設備 Endian-ness (讀取/寫入) 0 = 位元組排序為 little endian。 1 = 保留
17:16	SIZE	每個傳送資料的突發傳輸最大值 (讀取/寫入) 00 = 保留 01 = 8 位元組 10 = 16 位元組 11 = 32 位元組 若 DCMDx[LENGTH]比 DCMDx[SIZE]小，則傳輸資料大小就等於 DCMDx[LENGTH]。
15:14	WIDTH	On-chip 週邊的寬度 (讀取/寫入) 00 = 保留 01 = 1 位元組 10 = 半字組 (2 位元組) 11 = 字組 (4 位元組) 執行記憶體對記憶體或輔助晶片相關操作時，則必須程式化為 00。
13	保留	讀取為未知，而必須寫入為 0。
12:0	LENGTH	傳輸進入位元組的長度 (讀取/寫入) 說明傳輸為入位元組的長度。DCMD[LENGTH]=0 代表 0 位元組，只適用敘述元抓取模式。DCMD[LENGTH]=0 對非敘述元抓取模式是無效的。傳輸資料的最大值是 (8K-1) 位元組。若傳送影響任何內部週邊，則傳送的長度必須為那週邊寬度的整數倍數。

7.4 範例

此章節包含以下的範例：

- 設定與啟動通道
- 初始化通道內正在執行的敘述元串列
- 增加敘述元至通道內正在執行的敘述元串列的尾端
- 初始化將被 Direct DMA Master 使用的通道

範例 1 · 如何設定與啟動通道

接下來的例子說明如何設定一個通道從 DSADR 位址傳送 LENGTH 長度的字組至 DTADR I/O 位址。此範例也說明了如何啟動傳輸。範例中設定了 DDADR 內的停止位元，因此再完成傳送長度為 LENGTH 的位元組資料之後，DMA 通道會停止。

```
// build real descriptor
desc[0].ddadr = STOP
desc[0].dsadr = DSADR
desc[0].dtadr = DTADR
desc[0].dcmd = DCMD

// start the channel
DMANEXT[CHAN] = &desc[0]
DRUN = 1
```

範例 2 · 如何初始化正在執行的通道之敘述元串列：

```
// Allocate a new descriptor, and make it an end
// descriptor whose “ddadr” field points back at itself
newDesc = New Desc()
newDesc->ddadr = newDesc | STOP
// make it a zero length descriptor
newDesc->dcmd = ZERO
// Start the channel
DMANEXT[CHAN] = newDesc
DRUN = 1
```

通道啟動，然後載入敘述元至暫存器中，最後因傳輸長度為 0 且 STOP 位元被設定而停止。在此範例中未傳送任何資料。可藉由寫入通道的 DDADR 與寫入 1 到 DCSR[RUN]位元來重新啟動通道。

範例 3 · 如何加入敘述元至一個正在執行的通道之敘述元串列的尾端：

此範例假設為啟動敘述元抓取模式。

DMA 敘述元串列用來當網路傳送端的滿緩衝區的佇列，或是網路接收端的

空緩衝區之佇列。因為佇列裡的緩衝區通常很小（尤其是與 ATM cell 一樣小），而動態的（on-the-fly）DMA 敘述元串列的控制必須有效率。

1. 寫入 0 至 DCSR[RUN]
2. 等待直到通道停止。通道的停止狀態反映在 DCSR:STOPSTATE 位元。
3. 在記憶體中，製造要加入的敘述元，並且將停止位元設定為 1。
4. 在記憶體中，控制目前鍊結的最後一個敘述元的 DDADR，如此一來 DDADR 指出步驟 3 所製造的敘述元。
5. 在記憶體中，製造一個新的敘述元，具有與停止的 DMA 通道相同的 DDADR、DSADR、DTADR 與 CMD。
6. 檢查 DMA 通道暫存器並決定鍊結的最後一個敘述元的通道是否停止。
7. 使用步驟 5 所產生的敘述元來程式化通道的 DDADR。
8. 設定 DCSR[RUN] 為 1。

範例 4 · 如何初始化將被 Direct DMA Master 所使用的通道：

在 I/O 通道與主記憶體之間移動資料最有效率的方法為應用處理器的運用敘述元的 DMA 系統。由於每個應用程式需求都不同，因此當運用非敘述元的 DMA 對某些應用程式最佳時，運用敘述元的 DMA 可能對其餘的應用程式最佳。對於無法等待在每個 DMA 傳送之前抓取敘述元所需要的時間的應用程式，必須選擇運用非敘述元的方法。對於可以等待的應用程式，運用敘述元的 DMA 方法可以減少核心干預的數量。

自行修改（Self-Modifying）敘述元：運用敘述元的 DMA 系統可用來提供直接記憶體存取給需要的裝置。

在此範例中，輔助晶片需要下列的需求：

1. 當輔助晶片將 DREQ 從 0 觸發為 1，DMA 必須從晶片的其中一個埠抓取敘述元的 4 個字組。
2. 基於 4 個敘述元字組所包含的資訊，DMA 必須從來源位址傳送資料至目的位址，而不等待輔助晶片的另一個要求。
3. 在傳送完數目為 DCMD:LENGTH 的位元組後，DMA 會回到步驟 1。

具備這些需求的外部裝置可使用記憶體內固定的敘述元。

```
Struct {longaddr;  
        longaddr;  
        longdtadr;  
        shortlength;
```

```

shortdcmd;
} desc[2];
desc[0].ddadr = &desc[1]
desc[0].dsadr = I_ADR + I_DESC_OFFS
desc[0].dtadr = &desc[1].dsadr
desc[0].length = 8;
desc[0].dcmd = CMD_IncTrgAdr | CMD_FlowThru;
desc[1].ddadr = &desc[0]
desc[1].dtadr = I_ADR + I_DATA_OFFS
desc[1].dsadr = 0
desc[1].length = 0
desc[1].dcmd = 0
    
```

當外部設備有資料要傳輸時，它會以標準的方式傳送 DMA 要求。DMAC 會開始從設備的 I_DESC_OFFS 位址讀取 4 個字組 (DMAC 只可傳送 4 個字組，因為第一敘述元的計數為 8 位元組)。這個來自外部設備的 4 個字組會被寫入下個敘述元的 DSADR、DTADR 和 DCMD 區域。然後 DMAC 會進入下個敘述元 (動態修改)，使用外部設備的 I_DATA_OFFS 位址，開始傳送外部設備要求的資料。當傳輸結束，DMAC 會回到第一敘述元，然後重複程序。

7.5 DMAC 暫存器

此節說明 DMAC 暫存器。請參考 7-13。

表 7-13 DMAC 控制暫存器

位址	名稱	說明
0x4000 0000	DCSR0	通道 0 DMA 控制/狀態暫存器
0x4000 0004	DCSR1	通道 1 DMA 控制/狀態暫存器
0x4000 0008	DCSR2	通道 2 DMA 控制/狀態暫存器
0x4000 000C	DCSR3	通道 3 DMA 控制/狀態暫存器
0x4000 0010	DCSR4	通道 4 DMA 控制/狀態暫存器
0x4000 0014	DCSR5	通道 5 DMA 控制/狀態暫存器
0x4000 0018	DCSR6	通道 6 DMA 控制/狀態暫存器
0x4000 001C	DCSR7	通道 7 DMA 控制/狀態暫存器
0x4000 0020	DCSR8	通道 8 DMA 控制/狀態暫存器
0x4000 0024	DCSR9	通道 9 DMA 控制/狀態暫存器
0x4000 0028	DCSR10	通道 10 DMA 控制/狀態暫存器
0x4000 002C	DCSR11	通道 11 DMA 控制/狀態暫存器

0x4000 0030	DCSR12	通道 12 DMA 控制/狀態暫存器
0x4000 0034	DCSR13	通道 13 DMA 控制/狀態暫存器
0x4000 0038	DCSR14	通道 14 DMA 控制/狀態暫存器
0x4000 003C	DCSR15	通道 15 DMA 控制/狀態暫存器
0x4000 00F0	DINT	DMA 中斷暫存器
0x4000 0100	DRCMR0	傳給通道對應暫存器的 DREQ 0 要求 (擴充晶片要求 0)
0x4000 0104	DRCMR1	傳給通道對應暫存器的 DREQ 1 要求 (擴充晶片要求 1)
0x4000 0108	DRCMR2	傳給通道對應暫存器的 12S 接收要求
0x4000 010C	DRCMR3	傳給通道對應暫存器的 12S 傳送要求
0x4000 0110	DRCMR4	傳給通道對應暫存器的 BTUART 接收要求
0x4000 0114	DRCMR5	傳給通道對應暫存器的 BTUART 傳送要求
0x4000 0118	DRCMR6	傳給通道對應暫存器的 FFUART 接收要求
0x4000 011C	DRCMR7	傳給通道對應暫存器的 FFUART 傳送要求
0x4000 0120	DRCMR8	傳給通道對應暫存器的 AC97 麥克風接收要求
0x4000 0124	DRCMR9	傳給通道對應暫存器的 AC97 數據機接收要求
0x4000 0128	DRCMR10	傳給通道對應暫存器的 AC97 數據機傳送要求
0x4000 012C	DRCMR11	傳給通道對應暫存器的 AC97 聲音接收要求
0x4000 0130	DRCMR12	傳給通道對應暫存器的 AC97 聲音傳送要求
0x4000 0134	DRCMR13	傳給通道對應暫存器的 SSP 接收要求
0x4000 0138	DRCMR14	傳給通道對應暫存器的 SSP 傳送要求
0x4000 013C	DRCMR15	保留
0x4000 0140	DRCMR16	保留
0x4000 0144	DRCMR17	傳給通道對應暫存器的 FICP 接收要求
0x4000 0148	DRCMR18	傳給通道對應暫存器的 FICP 傳送要求
0x4000 014C	DRCMR19	傳給通道對應暫存器的 STUART 接收要求
0x4000 0150	DRCMR20	傳給通道對應暫存器的 STUART 傳送要求

		要求
0x4000 0154	DRCMR21	傳給通道對應暫存器的 MMC 接收要求
0x4000 0158	DRCMR22	傳給通道對應暫存器的 MMC 傳送要求
0x4000 015C	DRCMR23	保留
0x4000 0160	DRCMR24	保留
0x4000 0164	DRCMR25	傳給通道對應暫存器的 USB 端點 1 要求
0x4000 0168	DRCMR26	傳給通道對應暫存器的 USB 端點 2 要求
0x4000 016C	DRCMR27	傳給通道對應暫存器的 USB 端點 3 要求
0x4000 0170	DRCMR28	傳給通道對應暫存器的 USB 端點 4 要求
0x4000 0174	DRCMR29	保留
0x4000 0178	DRCMR30	傳給通道對應暫存器的 USB 端點 6 要求
0x4000 017C	DRCMR31	傳給通道對應暫存器的 USB 端點 7 要求
0x4000 0180	DRCMR32	傳給通道對應暫存器的 USB 端點 8 要求
0x4000 0184	DRCMR33	傳給通道對應暫存器的 USB 端點 9 要求
0x4000 0188	DRCMR34	保留
0x4000 018C	DRCMR35	傳給通道對應暫存器的 USB 端點 11 要求
0x4000 0190	DRCMR36	傳給通道對應暫存器的 USB 端點 12 要求
0x4000 0194	DRCMR37	傳給通道對應暫存器的 USB 端點 13 要求
0x4000 0198	DRCMR38	傳給通道對應暫存器的 USB 端點 14 要求
0x4000 019C	DRCMR39	保留
0x4000 0200	DDADR0	DMA 敘述元位址暫存器通道 0
0x4000 0204	DSADR0	DMA 來源位址暫存器通道 0
0x4000 0208	DTADR0	DMA 目標位址暫存器通道 0
0x4000 020C	DCMD0	DMA 命令位址暫存器通道 0
0x4000 0210	DDADR1	DMA 敘述元位址暫存器通道 1
0x4000 0214	DSADR1	DMA 來源位址暫存器通道 1
0x4000 0218	DTADR1	DMA 目標位址暫存器通道 1
0x4000 021C	DCMD1	DMA 命令位址暫存器通道 1
0x4000 0220	DDADR2	DMA 敘述元位址暫存器通道 2
0x4000 0224	DSADR2	DMA 來源位址暫存器通道 2
0x4000 0228	DTADR2	DMA 目標位址暫存器通道 2
0x4000 022C	DCMD2	DMA 命令位址暫存器通道 2
0x4000 0230	DDADR3	DMA 敘述元位址暫存器通道 3

0x4000 0234	DSADR3	DMA 來源位址暫存器通道 3
0x4000 0238	DTADR3	DMA 目標位址暫存器通道 3
0x4000 023C	DCMD3	DMA 命令位址暫存器通道 3
0x4000 0240	DDADR4	DMA 敘述元位址暫存器通道 4
0x4000 0244	DSADR4	DMA 來源位址暫存器通道 4
0x4000 0248	DTADR4	DMA 目標位址暫存器通道 4
0x4000 024C	DCMD4	DMA 命令位址暫存器通道 4
0x4000 0250	DDADR5	DMA 敘述元位址暫存器通道 5
0x4000 0254	DSADR5	DMA 來源位址暫存器通道 5
0x4000 0258	DTADR5	DMA 目標位址暫存器通道 5
0x4000 025C	DCMD5	DMA 命令位址暫存器通道 5
0x4000 0260	DDADR6	DMA 敘述元位址暫存器通道 6
0x4000 0264	DSADR6	DMA 來源位址暫存器通道 6
0x4000 0268	DTADR6	DMA 目標位址暫存器通道 6
0x4000 026C	DCMD6	DMA 命令位址暫存器通道 6
0x4000 0270	DDADR7	DMA 敘述元位址暫存器通道 7
0x4000 0274	DSADR7	DMA 來源位址暫存器通道 7
0x4000 0278	DTADR7	DMA 目標位址暫存器通道 7
0x4000 027C	DCMD7	DMA 命令位址暫存器通道 7
0x4000 0280	DDADR8	DMA 敘述元位址暫存器通道 8
0x4000 0284	DSADR8	DMA 來源位址暫存器通道 8
0x4000 0288	DTADR8	DMA 目標位址暫存器通道 8
0x4000 028C	DCMD8	DMA 命令位址暫存器通道 8
0x4000 0290	DDADR9	DMA 敘述元位址暫存器通道 9
0x4000 0294	DSADR9	DMA 來源位址暫存器通道 9
0x4000 0298	DTADR9	DMA 目標位址暫存器通道 9
0x4000 029C	DCMD9	DMA 命令位址暫存器通道 9
0x4000 02A0	DDADR10	DMA 敘述元位址暫存器通道 10
0x4000 02A4	DSADR10	DMA 來源位址暫存器通道 10
0x4000 02A8	DTADR10	DMA 目標位址暫存器通道 10
0x4000 02AC	DCMD10	DMA 命令位址暫存器通道 10
0x4000 02B0	DDADR11	DMA 敘述元位址暫存器通道 11
0x4000 02B4	DSADR11	DMA 來源位址暫存器通道 11
0x4000 02B8	DTADR11	DMA 目標位址暫存器通道 11
0x4000 02BC	DCMD11	DMA 命令位址暫存器通道 11
0x4000 02B0	DDADR12	DMA 敘述元位址暫存器通道 12
0x4000 024C	DSADR12	DMA 來源位址暫存器通道 12

0x4000 02C8	DTADR12	DMA 目標位址暫存器通道 12
0x4000 02CC	DCMD12	DMA 命令位址暫存器通道 12
0x4000 02D0	DDADR13	DMA 敘述元位址暫存器通道 13
0x4000 02D4	DSADR13	DMA 來源位址暫存器通道 13
0x4000 02D8	DTADR13	DMA 目標位址暫存器通道 13
0x4000 02DC	DCMD13	DMA 命令位址暫存器通道 13
0x4000 02E0	DDADR14	DMA 敘述元位址暫存器通道 14
0x4000 02E4	DSADR14	DMA 來源位址暫存器通道 14
0x4000 02E8	DTADR14	DMA 目標位址暫存器通道 14
0x4000 02EC	DCMD14	DMA 命令位址暫存器通道 14
0x4000 02F0	DDADR15	DMA 敘述元位址暫存器通道 15
0x4000 02F4	DSADR15	DMA 來源位址暫存器通道 15
0x4000 02F8	DTADR15	DMA 目標位址暫存器通道 15
0x4000 02FC	DCMD15	DMA 命令位址暫存器通道 15

問題：

1. 請問何為 DMA？其功用為何？
2. 請問第三方 DMA 與第一方 DMA 有何差異？
3. 請問在 Intel XScale 應用處理器中的 DMA 有什麼樣的特性？
4. 請問何為 Flow-through 傳送方式？
5. 請問在 Intel XScale 應用處理器中的 DMA 可接收哪兩類週邊的要求？由哪些腳位接收這些要求？如何得之一個要求已經發生？
6. 請問 Intel XScale 應用處理器中的 DMA 通道優先權排程為何？
7. 請問若在通道 0、1、2、5、7、10、11、12、15 均有資料等待傳送，則其傳送的順序為何？
8. 請問在 Intel XScale 應用處理器中，DMA 控制器支援哪兩種傳輸模式？各有何特色？
9. 請問非敘述元抓取模式的操作過程為何？
10. 請問敘述元抓取模式的操作過程為何？
11. 在 Intel XScale 應用處理器中，DMA 在傳送的最後尾端位元組有可能小於所指定的傳輸資料長度。此時有些裝置會自動傳送這些尾端位元組，但有些不會。請問在哪些狀況下會自動傳送這些尾端位元組？當不會自動傳輸這些尾端位元組時，請問該如何處理？
12. 請問在 Intel XScale 應用處理器中的 DMA 如何服務內部週邊？
13. 請問在 Intel XScale 應用處理器中的 DMA 如何服務輔助晶片和外部週邊？