

## 第十六章 內部整合電路聲音(I2S)控制器

I2S是一種數位立體音訊協定，PXA250與PXA210應用程式處理器的I2S控制器(I2SC)功能區塊控制I2S連結(I2SLINK)，I2S連結是一個立體音訊的低電源四支腳位的序列介面，I2S介面和音訊編碼解碼器'97 (AC'97)介面可能不能同時使用。

I2S是一個為數位立體聲音裝置所設計的串列匯流排，其中至少包含兩條分開的聲音資料線(Audio data)與時脈訊號線(clock)。藉由分開資料線與時脈訊號線，則與時間相關的錯誤，例如抖動(jitter)將可以避免，因此可以免去使用防抖動(anti-jitter)的裝置。I2S的匯流排設計包含下列的串列匯流排：資料線(可以作為輸出或輸入)、一條選擇線(可選擇左、右頻道)、一條時脈線。下面將對I2S作更詳細的介紹。

### 16.1 概觀

I2SC是由緩衝器、狀態暫存器、控制暫存器、平行序列轉換器和計數器組成，使用在應用程式處理器系統記憶體和一個外部I2S編碼解碼器之間的數位化音訊傳輸。

為了重播放數位化音訊的或合成音訊的產生，I2SC從應用程式處理器系統記憶體檢索數位化音訊取樣且將它們經由I2S連結傳送到編碼解碼器，再由編碼解碼器中的外部數位到類比轉換器轉換音訊取樣成為類比音訊波狀。

為了記錄數位化音訊，I2SC從一個編碼解碼器(經由I2S連結)接收數位化音訊取樣且將它們儲存到應用程式處理器系統記憶體。

I2S控制器支援標準I2S與MSB調整I2S格式，控制器以四支或五支腳位連接到一個外部編碼解碼器：

- 一個位元率時脈，能使用一個內部或一個外部來源資料
- 一種格式化或“左/右”控制訊號
- 二個序列音訊腳位，一個做為輸入和另一個做為輸出
- 位元率時脈，一個選擇系統時脈也由I2SC傳送到編碼解碼器

I2S資料能夠藉由DMA控制器或程式I/O二者之一來儲存到系統記憶體或是從系統記憶體取得資料。

對於I2S系統，需要額外的腳位來控制外部編碼解碼器，一些編碼解碼器使用一個L3控制匯流排，它需要3個訊號— L3\_CLK、L3\_DATA和L3\_MODE — 來寫入位元組到L3匯流排暫存器，I2SC經由一般用途I/O (GPIO)腳位的軟體控制來支援L3匯流排協定，I2SC不提供L3匯流排協定的硬體控制。

存在二種透過一個序列路徑傳送數位化立體音訊的相似協定：標準I2S與MSB調整I2S，二者均可工作在多種時脈率，能夠由程式化的分配器或從一個外部時脈源極分配PLL時脈來獲得，更多關於時脈率的詳情見表14-2, “支援取樣頻率”在14-6頁。

## 16.2 訊號說明

SYSCLK是I2S單元中所有其他時脈的基礎時脈，SYSCLK由程式化的分配器分配PLL時脈來產生大約介於2百萬赫茲和12.2百萬赫茲之間的頻率，這個頻率總是為256次的音訊取樣頻率，只有在BITCLK組態為輸出時，SYSCLK由應用程式處理器驅動。

BITCLK支援序列音訊位元率(外部編碼解碼器位元取樣邏輯的基準)，BITCLK的頻率是SYSCLK頻率的四分之一和64次的音訊取樣頻率，每一個BITCLK週期會傳送或接收一個序列音訊資料取樣位元。一個單獨的序列音訊取樣包含了“左”邊和“右”邊訊號，每個取樣含有8、16或32位元。

SYNC的頻率是BITCLK頻率的64分之一，為8千赫茲到48千赫茲的訊號，SYNC的狀態使用來表示目前序列音訊資料取樣是由“左”邊或“右”邊的哪一個通道資料。

SDATA\_IN和SDATA\_OUT資料腳位使用來傳送序列音訊資料到編碼解碼器或從編碼解碼器接收序列音訊資料。

表16-1列出I2S和一個外部編碼解碼器裝置之間的訊號

表16-1 外部介面到編碼解碼器

名稱	方向	說明
GP32/SYSCLK	O	系統時脈= BITCLK * 4 ，只有編碼解碼器能使用
GP28/BITCLK	I 或 O	位元率時脈 = SYNC * 64
GP31/SYNC	O	識別 左/右
GP30/SDATA_OUT	O	序列音訊輸出資料到編碼解碼器
GP29/SDATA_IN	I	序列音訊從編碼解碼器輸入資料

BITCLK能組態為輸入或輸出其中一個，為了設計方向，依循下列這些步驟：

1. 設計SYSUNIT的GPIO方向暫存器(GPDR)，見4.1.3.2節, “GPIO腳位方向暫存器(GPDR)” 在4-8頁有關於GPDR的細節。
2. 設計SYSUNIT的GPIO交替功能暫存器選擇暫存器(GAFR)，見4.1.3.6節, “GPIO交替功能暫存器(GAFR)” 在4-17頁有關於GAFR的細節。
3. 設計I2SC的序列音訊控制暫存器中的BCKD位元，見14.6.1節, “序列音訊控制器廣域控制暫存器(SACR0)”有更多的細節。

注意：在正常操作時更改SACR0[BCKD]位元的狀態會造成BITCLK的跳動且會影響序列的動作。

假如BITCLK是一個輸出，SYSCLK必須組態為輸出。假如BITCLK是由編碼解碼器提供，GPIO腳位GP32能被使用來當作一個交替功能，組態這個腳位為輸出端須依循下列這些步驟：

1. 設計SYSUNIT的GPIO方向暫存器(GPDR)，見4.1.3.2節，“GPIO腳位方向暫存器(GPDR)”在4-8頁有關於GPDR的細節。
2. 設計SYSUNIT的GPIO交替功能暫存器選擇暫存器(GAFR)，見4.1.3.6節，“GPIO交替功能暫存器(GAFR)”在4-17頁有關於GAFR的細節。

組態SYNC與SDATA\_OUT為輸出端須依循下列這些步驟：

1. 設計SYSUNIT的GPIO方向暫存器(GPDR)，見4.1.3.2節，“GPIO腳位方向暫存器(GPDR)”在4-8頁有關於GPDR的細節。
2. 設計SYSUNIT的GPIO交替功能暫存器選擇暫存器(GAFR)，見4.1.3.6節，“GPIO交替功能暫存器(GAFR)”在4-17頁有關於GAFR的細節。

組態SDATA\_IN為輸入端須依循下列這些步驟：

1. 設計SYSUNIT的GPIO方向暫存器(GPDR)，見4.1.3.2節，“GPIO腳位方向暫存器(GPDR)”在4-8頁有關於GPDR的細節。
2. 設計SYSUNIT的GPIO交替功能暫存器選擇暫存器(GAFR)，見4.1.3.6節，“GPIO交替功能暫存器(GAFR)”在4-17頁有關於GAFR的細節。

## 16.3 控制器操作

I2S控制器(I2SC)能經由處理器或DMA控制器其中一個來進行存取。

處理器使用程式I/O指令來存取I2SC並且能存取下列型式的資料：

- I2SC暫存器資料—所有暫存器都是32位元且對齊字組邊界，見14.6節，“暫存器”有更多細節。
- I2SC FIFO資料—藉由寫入到I2SC的序列音訊資料暫存器(SADR)將一個通道放置到傳送FIFO，藉由寫入到SADR來更新一個傳送FIFO通道，藉由讀取SADR來清除一個接收FIFO通道。
- I2S編碼解碼器資料—編碼解碼暫存器能透過L3匯流排來進行存取，L3匯流排操作是由軟體控制3個GPIO腳位來模擬。
- DMA控制器能夠只存取FIFO，存取是透過資料暫存器來進行的，在前面幾節有說明，DMA控制器存取區塊為8、16或32位元組中的FIFO資料，DMA控制器對I2SC發出的下列DMA要求做出回應：

- 傳送FIFO要求是以傳送臨界值(TFTH)設定為基礎，如果所傳送的FIFO數少於傳送臨界值(TFTH)加一個通道時，則傳送FIFO要求被設定。見表 14-3, “SACR0位元說明”在14-9頁有關於TFTH的更多細節。
- 接收FIFO要求是以接收臨界值(RFTH)設定為基礎，如果所接收FIFO數多於接收臨界值(RFTH)加一個通道時，則接收FIFO要求被設定。見表 14-3, “SACR0位元說明”在14-9頁有關於RFTH的更多細節。

### 16.3.1 初始化

1. 藉由設計SYSUNIT的GPIO方向暫存器、SYSUNIT的GPIO交替功能選擇暫存器和I2SC的序列音訊控制器廣域控制暫存器(位元2)來設置BITCLK方向。
2. 選擇標準I2S或MSB調整I2S模式的操作，可以由設計序列音訊控制器I2S/MSB調整控制暫存器(SACR1)的位元0來完成，更多的細節見14.6.2節, “序列音訊控制器I2S/MSB調整控制暫存器(SACR1)” 在14-11頁。
3. 選擇：程式I/O可能會使用來裝填一些取樣(範圍從1到16)到傳送FIFO，假如I2SLINK隨著一個空的傳送FIFO啟動，一個狀態暫存器中的傳送欠載錯誤位元將被設置，更多的細節見14.6.3節, “序列音訊控制器I2S/MSB調整狀態暫存器(SASR0)”，這是一個選擇步驟可以避免這樣的錯誤，假如步驟3沒有執行，則程式I/O必須藉由設定中斷清除暫存器中的位元5來清除傳送欠載狀態位元，更多的細節見 14.6.5節, “序列音效中斷清除暫存器(SAICR)”。
4. 下列的控制位元可以在I2SC的序列音訊控制器廣域控制暫存器(SACR0)中同時設計：
  - a. 藉由設定SACR0的ENB位元(位元0)來啟動I2SLINK
  - b. 由於SACR0暫存器將在步驟2被覆寫，所以在步驟1中須保持BITCLK方向，更改BITCLK方向將會干擾時脈且影響I2SLINK的活動性
  - c. 經由個別規劃SACR0[11:8]和SACR0(15:12)中的TFTH與RFTH位元來設計傳送臨界值和接收臨界值，見14.6.1.2節, “DMA服務所採用的TFTH與RFTH”，有關於允許的臨界值

一旦I2SLINK啟動，假如傳送FIFO仍然是空的，將傳送訊框全部填寫0的值，這將設置SASR0中的傳送欠載狀態位元。可以藉著執行步驟2來避免這個錯誤情況，在填寫至少一個取樣到傳送FIFO之後，有效資料透過I2SLINK傳送，一個取樣是由一個16位元左值和一個16位元右值所組成的32位元值。

啟動I2SLINK將也會造成I2SC記錄到零直到編碼解碼器傳送有效資料。

啟動I2SLINK也會啟動傳送和接收的DMA要求。

### 16.3.2 關閉與啟動音訊重播

當I2SC啟動時，音訊傳輸會自動啟動。傳輸或重播可以藉由觸發SACR1暫存器

中的DRPL位元來停止，更多的細節見14.6.2節，“序列音訊控制器I2S/MSB調整控制暫存器(SACR1)”。

觸發SACR1中的DRPL位元有下列的影響：

1. 所有I2SLINK重播動作是關閉的，在重播關閉時的訊框或資料取樣將有無效的資料(一些資料位元將覆寫為零)，為了避免這種情形，只有在有效的資料傳輸後才可關閉重播，因此全為零的訊框會被傳送
2. 傳送FIFO指標會重置為零
3. 傳送FIFO填充-階層會重置為零
4. 零會透過I2SLINK傳送
5. 傳送的DMA要求是關閉的

### 16.3.3 關閉與啟動音訊記錄

當I2SC啟動時，音訊記錄會自動啟動。記錄也可以藉由觸發SACR1暫存器中的DREC位元來停止，更多的細節見14.6.2節，“序列音訊控制器I2S/MSB調整控制暫存器(SACR1)”。

觸發SACR1中的DREC位元有下列的影響：

1. I2SLINK記錄動作是關閉的，在記錄關閉時的訊框或資料取樣將有無效的資料(一些資料位元將覆寫零)，為了避免這種情形，只有在有效的資料傳輸後才可關閉記錄
2. 接收FIFO指標會重置為零
3. 接收FIFO填充-階層會重置為零
4. 任何由DMA/CPU的讀取操作會以零回傳
5. 接收的的DMA要求是關閉的

### 16.3.4 傳送FIFO錯誤

狀態位元在傳送欠載狀況期間設置，它會觸發一個中斷(假如中斷是致能的話)，更多的細節見14.6.3節、14.6.6節和14.6.5節。在傳送欠載狀況期間，最後有效取樣是連續透過I2SLINK來傳送的，傳送欠載會在下列情況下發生：

1. 有效傳送的資料在記憶體中仍然是可用的但是DMA控制器極需要傳送的FIFO因為它在服務其他有較高優先權的週邊設備。
2. DMA控制器已經從記憶體傳送所有有效資料到傳送的FIFO

第二個情況促使最後的有效取樣透過I2SLINK來回應直到關閉SACR0[ENB]位元來關掉I2SC。

### 16.3.5 接收FIFO錯誤

狀態位元在接收滿溢狀況期間設置，它會觸發一個中斷(假如中斷是致能的)，更多的細節見14.6.3節、14.6.6節和14.6.5節。在接收滿溢狀況期間，由編碼解碼器傳送的資料會遺失(不會被記錄)。

### 16.3.6 尾部位元組

當編碼解碼器已經完成傳送有效資料，零將會被I2SC記錄，且這將繼續直到關閉SACR0[ENB]位元來關掉單元。

假如接收資料的總緩衝區容量小於接收臨界值時，零將被記錄，當所設計的臨界值到達時，接收DMA要求將執行。

## 16.4 序列音訊時脈與取樣頻率

BITCLK是音訊資料位元進入或離開I2SLINK時的速率，編碼解碼器需要SYSCLK來執行類比轉數位操作。

BITCLK能由編碼解碼器或一個內部PLL來提供，假如由內部提供，BITCLK與SYSCLK組態為輸出腳位，且二者能由編碼解碼器提供，假如BITCLK是由編碼解碼器提供，則它組態為輸入腳位，因此，SYSCLK的GPIO腳位能當成一個交替功能使用。

BITCLK列在表14-2，對於不同的取樣頻率有不同的結果，假如BITCLK選擇當作輸出，音訊時脈除數暫存器分割147.46MHz PLL時脈來產生SYSCLK，SYSCLK再除以四來產生BITCLK。取樣頻率是SYNC訊號的頻率，是由BITCLK除以64來產生，見14.6.4節，“序列音訊時脈除數暫存器(SADIV)”，有更多關於暫存器的細節。

48千赫茲的取樣支援MPEG2與MPEG4，44.1千赫茲的頻率支援MP3。

表16-2 支援的取樣頻率

音訊時脈除數暫存器(31:0)	SYSCLK = 147.6 MHz / (SADIV)	BITCLK = SYSCLK / 4	SYNC或取樣頻率= BITCLK / 64
0x0000-000C	12.288 MHz	3.072 MHz	48.000 kHz (closest std = 48 kHz)
0x0000-000D	11.343 MHz	2.836 MHz	44.308 kHz (closest std = 44.1 kHz)
0x0000-001A	5.671 MHz	1.418 MHz	22.154 kHz (closest std = 22.05 kHz)

0x0000-0024	4.096 MHz	1.024 MHz	16.000 kHz (closest std = 16.00 kHz)
0x0000-0034	2.836 MHz	708.92 kHz	11.077 kHz (closest std = 11.025 kHz)
0x0000-0048	2.048 MHz	512.00 kHz	8.000 kHz (closest std = 8.00 kHz)

## 16.5 資料格式

### 16.5.1 FIFO與記憶體格式

FIFO緩衝區為16個字元的深度和32位元的寬度，在每一個方向每個通道可儲存32個取樣。

音訊資料每32位元的字儲存二個取樣(左+右)，即使取樣小於16位元。當右通道資料使用32位元字的位元[31:16]時，左通道資料佔用位元[15:0]。在每一個16位元欄位中，音訊取樣是向左調整，而未使用位元在右側(LSB)填入零。

在記憶體中，立體取樣的映像和FIFO緩衝區中的相同，然而，獨立通道音訊每一個取樣佔用了完全的32位元字，使用較高的半字元或較低的半字元，以這樣來決定是否它是左或右取樣。

### 16.5.2 I2S與MSB調整序列音訊格式

I2S與MSB調整是序列路徑上數位化立體音訊傳送的相似協定。

BITCLK提供序列音訊位元率，為外部編碼解碼器位元取樣邏輯的基礎，它的頻率為64次的音訊取樣頻率，除以64，8千赫茲到48千赫茲的訊號結果表示左邊和右邊序列資料取樣通過序列資料路徑的時間，左/右訊號從SYNC腳位上傳送到編碼解碼器，左/右訊號的每一個欄位伴隨著一個在資料腳位SDATA\_IN與SDATA\_OUT上的序列音訊資料取樣。

圖16-1和圖16-2提供時序圖表為I2S與MSB調整操作模式的格式。

資料在64個BITCLK週期的訊框中傳送和接收，每個訊框由一個左取樣和一個右取樣組成，每個取樣包含有16位元的有效資料，每個取樣的LSB 16位元均裝填零。

在標準I2S模式下，SYNC為低電位時是左取樣而高電位時是右取樣，同樣地，每個資料取樣的MSB是在SYNC邊緣落下後的一個BITCLK週期開始。

在MSB調整模式下，SYNC為高電位時是左取樣而低電位時是右取樣，同樣

地，每個資料取樣的MSB是對齊SYNC的邊緣。

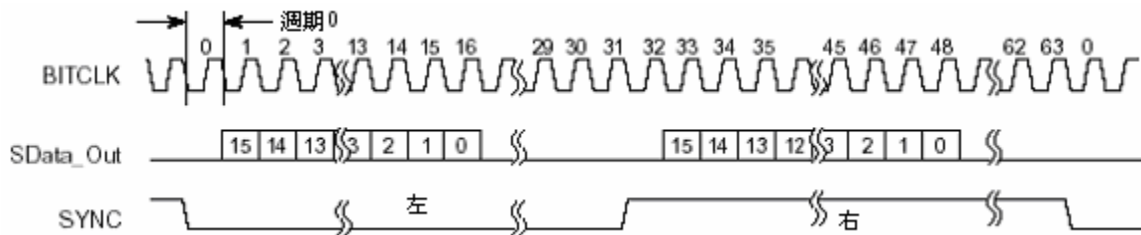


圖16-1 I2S資料格式(16位元)

注意：SData\_In的時序和SData\_Out是相同的

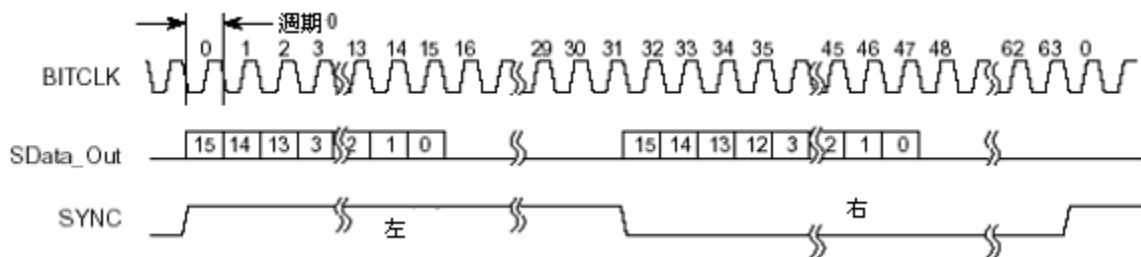


圖16-2 MSB調整資料格式(16位元)

注意：SData\_In的時序和SData\_Out是相同的

## 16.6 暫存器

I2S控制器暫存器都是可定址32位元，範圍從0x4040-0000到0x404FFFFF。

I2S控制器有下列型式的暫存器：

- 控制暫存器使用來設計一般的控制、交替模式特殊控制
- 資料暫存器使用來給傳送和接收FIFO進行存取
- 狀態暫存器發出FIFO緩衝區的狀態和介面的狀態訊號，由一般控制暫存器來選擇
- 中斷暫存器包含了中斷遮罩暫存器、中斷清除暫存器和中斷測試暫存器

### 16.6.1 序列音訊控制器廣域控制暫存器(SACR0)

這個暫存器控制一般I2S功能，所有位元為讀取/寫入，表14-3為SACR0的位元設定。

ENB位元控制I2SLINK為下：



- 清除ENB為零會做下列動作：
  - 關閉任何I2SLINK活動
  - 重置所有接收FIFO指標和控制I2SLINK的計數器
  - 重置接收FIFO
  - 不會影響傳送FIFO
  - 輸出腳位SYNC不會開啟
  - 反觸發所有DMA要求
  - 所有由處理器或DMA控制器進行讀取存取到資料暫存器(SADR)會回傳零
  - 關閉所有中斷
- 設定ENB為壹會做下列動作：
  - 啟動I2SLINK活動
  - 啟動DMA要求

表16-3 SACR0位元說明

實體位址 0x4040-0000		序列音訊控制器廣域 控制暫存器		I <sup>2</sup> S 控制器																														
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																																	
		reserved																RFTH	TFTH	reserved	STRF	EFWR	RST	BCKD	reserved	ENB								
重置																	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0

位元	名稱	說明
31:16	-	保留
15:12	RFTH	接收FIFO中斷或DMA臨界值，設置0到15的值，這個值必須設置為臨界值減1。 接收DMA要求觸發每當接收FIFO的個數 $\geq$ (RFTH+1)通道的個數
11:8	TFTH	傳送FIFO中斷或DMA臨界值，設置0到15的值，這個值必須設置為臨界值減1。 傳送DMA要求觸發每當傳送FIFO的個數 $<$ (TFTH+1)通道的個數
7:6	-	保留
5	STRF	選擇EFWR基本特殊用途功能的傳送或接收FIFO： • 0 = 選擇傳送FIFO • 1 = 選擇接收FIFO 細節見表 14-4
4	EFWR	這個位元啟動特殊用途FIFO寫入/讀取功能： • 0 = 特殊用途FIFO寫入/讀取功能是關閉的 • 1 = 特殊用途FIFO寫入/讀取功能是啟動的 細節見表 14-4
3	RST(1)	重置FIFO邏輯和所有暫存器，除了這個暫存器(SACR0)：

		<ul style="list-style-type: none"> <li>• 0 = 沒有設置</li> <li>• 1 = 重置作用到其他暫存器</li> </ul>
2	BCKD	這個位元說明BITCLK的輸入/輸出方向： <ul style="list-style-type: none"> <li>• 0 = 輸入，BITCLK由外部來源驅動</li> <li>• 1 = 輸出，BITCLK內部產生且驅動到編碼解碼器</li> </ul>
1	-	保留
0	ENB(1)(2)	啟動I2S功能： <ul style="list-style-type: none"> <li>• 0 = I2SLINK是關閉的</li> <li>• 1 = I2SLINK是關閉的</li> </ul>

注意：

1. 假如ENB在正常模式中觸發，RST位元也必須設置且清除來重置所有I2SC暫存器。
2. SACR0[ENB]控制訊號超過時脈範圍，它在一個內部時脈範圍(較BITCLK範圍快)註冊，在SACR0[ENB]傳送到較慢的BITCLK範圍前，它花了四個BITCLK週期和四個內部時脈週期，假如控制設定更改到比(4 BITCLK + 4 內部時脈)週期快的速率，在這個時間範圍最後更新的值會儲存在一個暫時暫存器且傳輸到BITCLK範圍。

### 16.6.1.1 特殊用途FIFO讀取/寫入功能

在表 16-4，EFWR與STRF能設計為特殊用途FIFO存取，在正常操作情況下，處理器或DMA控制器只能夠寫入到傳送FIFO和只能夠讀取接收FIFO，設計這些位元允許處理器或DMA控制器能讀取跟寫入二種FIFO。

表16-4 FIFO 寫入/讀取表

EFWR	STRF	說明
0	x	正常CPU/DMA 寫入/讀取狀況下： <ul style="list-style-type: none"> <li>• 寫入存取到資料暫存器來寫入一個傳送FIFO通道</li> <li>• 讀取存取到資料暫存器來讀出一個接收FIFO通道</li> <li>• I2SLINK從傳送FIFO讀取且寫入到接收FIFO</li> </ul>
1	0	CPU或DMA只能寫入和讀取傳送FIFO： <ul style="list-style-type: none"> <li>• 寫入存取到資料暫存器來寫入一個傳送FIFO通道</li> <li>• 讀取存取到資料暫存器來讀出一個傳送FIFO通道</li> <li>• I2SLINK不能讀取傳送FIFO但能寫入到接收FIFO</li> </ul>
1	1	CPU或DMA只能寫入和讀取接收FIFO： <ul style="list-style-type: none"> <li>• 寫入存取到資料暫存器來寫入一個接收FIFO通道</li> <li>• 讀取存取到資料暫存器來讀出一個接收FIFO通道</li> <li>• I2SLINK能讀取傳送FIFO但不能寫入到接收FIFO</li> </ul>

### 16.6.1.2 DMA服務所採用的TFTH與RFTH

DMA控制器只能設計來傳送8、16或32位元組的資料，這符合2、4或8個FIFO取樣，表16-5介紹TFTH與RFTH的值來預防傳送FIFO滿溢錯誤和接收FIFO欠載錯誤。

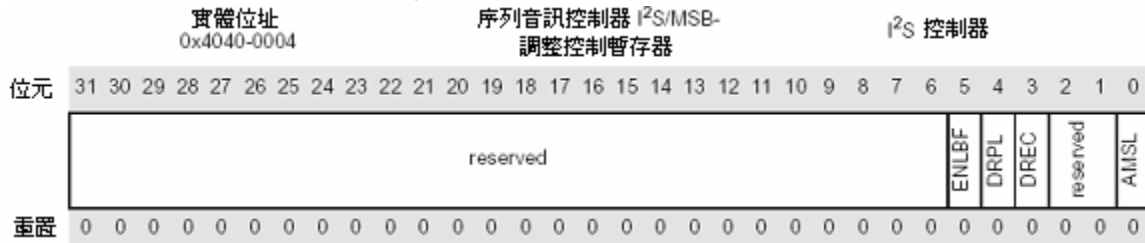
表16-5 對於DMA服務TFTH與RFTH的值

DMA 傳送容量	FIFO通道數	TFTH值		RFTH 值	
		最小值	最大值	最小值	最大值
8位元組	2	0	14	1	15
16位元組	4	0	12	3	15
32位元組	8	0	8	7	15

### 16.6.2 序列音訊控制器I2S/MSB調整控制暫存器(SACR1)

這個暫存器明確地控制I2S與MSB調整模式，所有位元為唯讀，表14-6為SACR1的位元設定。

表16-6 SACR1位元說明



位元	名稱	說明
31:6	-	保留
5	ENLBF	啟動I2S/MSB介面回送功能： <ul style="list-style-type: none"> <li>• 0 = I2S/MSB介面回送功能關閉</li> <li>• 1 = I2S/MSB介面回送功能啟動</li> </ul>
4	DRPL†	關閉I2S或MSB調整介面的重播功能： <ul style="list-style-type: none"> <li>• 0 = 重播功能啟動</li> <li>• 1 = 重播功能關閉</li> </ul>
3	DREC†	關閉I2S或MSB調整介面的記錄功能： <ul style="list-style-type: none"> <li>• 0 = 記錄功能啟動</li> <li>• 1 = 記錄功能關閉</li> </ul>
2:1	-	保留
0	AMSL†	指定交替模式(I2S或MSB調整)操作： <ul style="list-style-type: none"> <li>• 0 = 選擇I2S操作模式</li> <li>• 1 = 選擇MSB調整操作模式</li> </ul>

† SACR1位元DRPL、DREC和AMSL超過時脈範圍，它們在一個內部時脈範圍(較BITCLK範圍快)註冊，在這些控制傳送到較慢的BITCLK範圍前，它花了四個BITCLK週期和四個內部時脈週期，假如上面的控制設定更改到比(4 BITCLK + 4 內部時脈)週期快的速率，在這個時間範圍最後更新的值會儲存在一個暫時暫存器且傳輸到BITCLK範圍。

### 16.6.3 序列音訊控制器I2S/MSB調整狀態暫存器(SASR0)

序列音訊狀態暫存器(SASR0)是使用來記錄FIFO和I2SLINK的狀態，所有位元為唯讀，表14-7為SASR0的位元設定。

只有4位元分配給TFL與RFL，實際上的填充階層說明如下：

```
Actual_TFL(4:0) = {~TNF, TFL(3:0)}
Actual_RFL(4:0) calculation:
    if (RFL(3:0) == 4'b0)
        Actual_RFL(4:0) = {RNE, RFL(3:0)}
    else
        Actual_RFL(4:0) = {1'b0, RFL(3:0)}
```

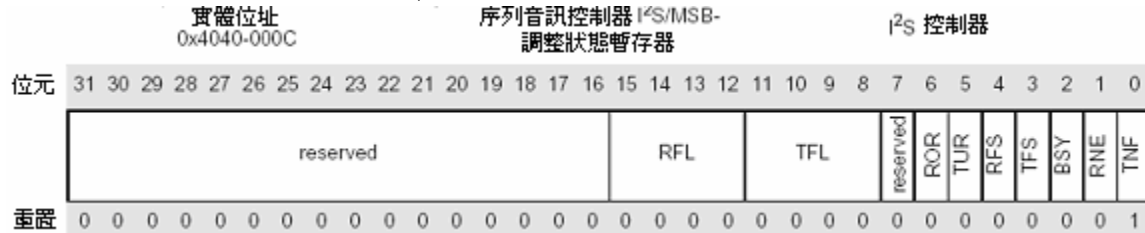
表16-7 SASR0位元說明

實體位址	序列音訊控制器 I <sup>2</sup> S/MSB-調整狀態暫存器		I <sup>2</sup> S 控制器																													
0x4040-000C																																
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
	reserved															RFL		TFL		reserved	ROR	TUR	RFS	TFS	BSY	RNE	TNF					
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	1

位元	名稱	說明
31:16	-	保留
15:12	RFL	接收FIFO階層：接收FIFO中的通道數
11:8	TFL	傳送FIFO階層：傳送FIFO中的通道數
7	-	保留
6	ROR	接收FIFO滿溢： • 0 = 接收FIFO沒有遇到滿溢 • 1 = I2S嘗試資料寫入到滿溢的接收FIFO (可中斷的) 假如序列音訊中斷遮罩暫存器的位元6是設置的，能中斷處理器藉由設定序列音訊中斷清除暫存器的位元6來清除
5	TUR	傳送FIFO欠載： • 0 = 傳送FIFO沒有遇到欠載 • 1 = I2S嘗試從一個空的傳送FIFO讀取資料

		假如序列音訊中斷遮罩暫存器的位元5是設置的，能中斷處理器藉由設定序列音訊中斷清除暫存器的位元5來清除
4	RFS	接收FIFO服務要求： • 0 = 接收FIFO的值在RFL臨界值之下或I2S關閉 • 1 = 接收FIFO的值在RFL臨界值以上 假如序列音訊中斷遮罩暫存器的位元4是設置的，能中斷處理器當接收FIFO通道的個數 < (RFTH + 1) 時會自動清除

表16-7 SASR0位元說明



位元	名稱	說明
3	TFS	傳送FIFO服務要求： • 0 – 傳送FIFO的值超過TFL臨界值或I2S關閉 • 1 – 傳送FIFO的值在TFL臨界值以下 假如序列音訊中斷遮罩暫存器的位元3是設置的，能中斷處理器當傳送FIFO通道的個數 $\geq (TFTH + 1)$ 時會自動清除
2	BSY	I2S忙碌： • 0 = I2S閒置或關閉 • 1 = I2S正在傳送或接收一個訊框
1	RNE	接收FIFO不是為空： • 0 = 接收FIFO為空 • 1 = 接收FIFO不是為空
0	TNF	傳送FIFO不是滿溢： • 0 = 傳送FIFO為滿溢 • 1 = 傳送FIFO不是滿溢

### 16.6.4 序列音訊時脈分配暫存器(SADIV)

這個暫存器使用來產生六個不同的BITCLK頻率和六個不同的取樣頻率，所有位元為讀取/寫入，表14-8為SADIV的位元設定

重置的值，0x0000001A，預設為22.05 千赫茲的取樣頻率。

注意：設定這個暫存器來得到其他的值，在表14-2, “支援的取樣頻率”，在14-6頁是不允許且會造成不預期的活動。

表16-8 SADIV位元說明

實體位址 0x4040-0060		序列音訊時脈分配暫存器	I <sup>2</sup> S 控制器
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	reserved	
重置	0 0	SADIV	
位元	名稱	說明	
31:7	-	保留	
6:0	SADIV	音訊時脈驅動器，有效的SADIV(6:0)為： <ul style="list-style-type: none"> <li>• 000 1100 = BITCLK of 3.072MHz</li> <li>• 000 1101 = BITCLK of 2.836 MHz</li> <li>• 001 1010 = BITCLK of 1.418MHz</li> <li>• 010 0100 = BITCLK of 1.024MHz</li> <li>• 011 0100 = BITCLK of 708.92 KHz</li> <li>• 100 1000 = BITCLK of 512.00 KHz</li> </ul>	

### 16.6.5 序列音訊中斷清除暫存器(SAICR)

序列音訊中斷清除暫存器(SAICR)是中斷控制暫存器，這是一個只能定址的位置且不能儲存資料，這些可定址位置只使用來清除狀態暫存器(SASR0)位元，每個位元位置對應到一個狀態暫存器中的中斷來源位元位置，表14-9為SAICR的位元設定。

這是一個只能寫入的暫存器，讀取操作將被看作是從保留位置讀取，重置的值被保留，因此暫存器不能被讀取。

表16-9 SAICR位元說明

實體位址 0x4040-0018		序列音訊中斷清除暫存器	I <sup>2</sup> S 控制器
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	reserved	
重置	r r	ROR TUR	reserved
位元	名稱	說明	
31:7	-	保留	
6	ROR	清除接收FIFO滿溢中斷和SASR0中的ROR狀態位元	
5	TUR	清除傳送FIFO欠載中斷和SASR0中的TUR狀態位元	
4:0	-	保留	

### 16.6.6 序列音訊中斷遮蔽暫存器(SAIMR)

寫入到中斷遮罩暫存器中對應的位元位置來啟動所對應到的中斷訊號，所有位元為讀取/寫入，表16-10為SAIMR的位元設定。

表14-10 SAIMR位元說明

實體位址 0x4040-0014		序列音訊中斷遮蔽暫存器	I <sup>2</sup> S 控制器
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	reserved	
重置	0 0	ROR	TUR RFS TFS reserved
位元	名稱	說明	
31:7	-	保留	
6	ROR	啟動接收FIFO滿溢狀況產生的中斷	
5	TUR	啟動傳送FIFO欠載狀況產生的中斷	
4	RFS	啟動接收FIFO服務要求產生的中斷	
3	TFS	啟動傳送FIFO服務要求產生的中斷	
2:0	-	保留	

### 16.6.7 序列音訊資料暫存器(SADR)

寫入一個32位元取樣到這個暫存器來更新資料到傳送FIFO，讀取這個暫存器從接收FIFO清除一個32位元取樣。

表16-11為SADR的位元設定，圖16-3說明資料通過FIFO和SADR的流程。

表16-11 SADR位元說明

實體位址 0x4040-0080		序列音訊資料暫存器	I <sup>2</sup> S 控制器
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	DTH	
		DTL	
重置	0 0		
位元	名稱	說明	
31:16	DTH	右資料取樣	
15:0	DTL	左資料取樣	

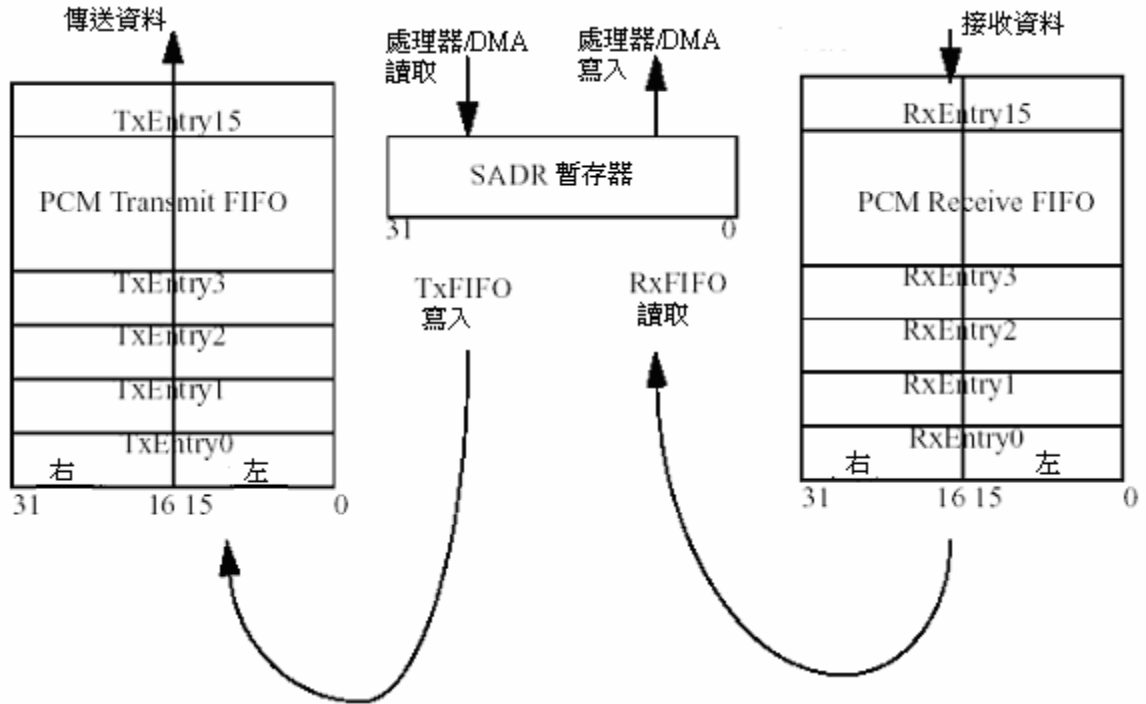


圖16-3 傳送與接收FIFO經由SADR進行存取

### 16.6.8 控制器：暫存器記憶體映像

所有暫存器都是字定址(32位元寬度)而且以0x00004單位遞增，所有I2SC暫存器映像的位址範圍為0x4040-000到0x404F-FFFF，見表16-12。

表16-12 暫存器記憶體映像

位址 (paddr(9:0))	暫存器名稱	說明
0x4040-0000	SACR0	廣域控制暫存器
0x4040-0004	SACR1	序列音訊I2S/MSB調整控制暫存器
0x4040-0008	-	保留
0x4040-000C	SASR0	序列音訊I2S/MSB調整介面和FIFO狀態暫存器
0x4040-0014	SAIMR	序列音訊中斷遮罩暫存器
0x4040-0018	SAICR	序列音訊中斷清除暫存器
0x4040-001C 到 0x4040-005C	-	保留
0x4040-0060	SADIV	音效時脈驅動暫存器，見14.4節，“序列音訊時脈與取樣頻率”
0x4040-0064 到	-	保留



0x4040-007C		
0x4040-0080	SADR	序列音訊資料暫存器(TX和RX FIFO存取暫存器)

## 16.7 中斷

下列SASR0狀態位元，假如啟動的話會中斷處理器：

- 接收FIFO服務DMA要求(RFS)
- 傳送FIFO服務DMA要求(TFS)
- 傳送欠載(TUR)
- 接收滿溢(ROR)

注意：更多的細節見16.6.3節，“序列音訊控制器I2S/MSB調整狀態暫存器(SASR0)”

問題：

1. I2S可以接多少個編解碼器？
2. I2S與外部的編解碼器的介面有幾隻接腳？他們分別代表著什麼功能？
3. I2S編解碼器的取樣頻率範圍為多少？
4. SYSCLK、BITCLK與SYNC之間的關係為何？
5. 如果我們希望設計取樣頻率為48KHz，要如何完成？
6. 為何I2S有尾數填零的功能？它用於何處？
7. 請分辨標準的I2S與MSB調整的I2S之間的差異？
8. I2S有多少的FIFO？
9. 當何種情況發生時，I2S的控制器會產生DMA的要求？
10. 如何設定I2S可以作聲音的錄音或重播的功能？
11. 試述何謂I2S (內部整合電路聲音控制器)。
12. 如何避免抖音 (jitter)。
13. 試述I2S的組成元件。