

第十七章 I2C匯流排介面單元

傳統上，當有多項裝置要連接到處理器時，各項裝置的住址線以及資料線會個別接到處理器上，如此一來，就佔用了處理器的腳位，使得處理器的IC腳數目增加。為了解決這個問題，飛利浦公司在1980年代發展出所謂的交互整合電路（I2C）匯流排。I2C是一個低頻寬、短距離的通訊協定。所有的裝置藉由兩條線連接在一起，這兩條線分別為串列資料線(SDA)和串列時脈線(SCL)。由於所有的通訊只在這兩條線上動作，所以每一個裝置必須有一個獨一無二的住址，讓處理器來辨識它。下面將更深入的來介紹交互整合電路（I2C）匯流排介面單元，內容包含了PXA250和PXA210應用程式處理器的操作模式和安裝。

17.1 概觀

I2C匯流排是由飛利浦公司研發的一個二支腳位的序列匯流排，SDA資料腳位是作為輸入和輸出功能使用而SCL時脈腳位是用來控制和參考I2C匯流排。I2C單元允許應用程式處理器透過I2C匯流排來服務master和slave裝置。

I2C單元啟動應用程式處理器與I2C周邊設備作溝通以及使用微控制器達成系統管理功能，I2C匯流排需要極少數的硬體來傳送有關於應用程式處理器系統到一個外部裝置的狀態和資料。

I2C單元是屬於應用程式處理器內部匯流排的一個周邊設備，資料是經由一個緩衝介面來傳送到I2C匯流排以及從I2C匯流排來接收，控制和狀態資訊是透過一組記憶體映像暫存器來傳送，請參考*I2C-Bus Specification*有I2C匯流排操作的完整細節。

注意：I2C單元不支援硬體通用呼叫、10位元定址或是CBUS相容性。

17.2 信號說明

I2C單元信號為SDA和SCL，表17-1 說明每個信號的功能。

表17-1 MMC信號說明

信號名稱	輸入/輸出	說明
SDA	雙向	I2C序列資料/位址信號
SCL	雙向	I2C序列時脈線信號

17.3 功能說明

I2C匯流排定義一個序列協定給I2C匯流排上的代理器處理來處理通過的資訊，此序列協定由序列資料/位址（SDA）線和序列時脈線（SCL）所組成的二腳位介面，I2C匯流排上的每個裝置是由一個獨特的7位元位址來識別而且可以在master或slave模式下當成傳送端或接收端來操作，表17-2列出I2C的操作模式。

表17-2 I2C匯流排定義

I2C 裝置	定義
傳送端	發送資料到I2C匯流排
接收端	從I2C匯流排接收資料
Master	起始傳送，產生時脈信號，還有終止傳送
Slave	由master來定址裝置
Multi-master	多個master能同時試著控制匯流排而不損毀資訊
調停	當多個master同時嘗試要控制匯流排時，保證只有一個master能控制匯流排，這樣能保證資訊不會被損毀

例如，當應用程式處理器I2C單元在匯流排上作用為master時,它定址一個EEPROM為slave來接收資料（見圖17-1）。當I2C單元定址EEPROM時，它是一個master傳送端而EEPROM是一個slave接收端。當I2C讀取資料時，它是一個master接收端而EEPROM是一個slave傳送端。不管它是傳送端或是接收端，master會產生時脈信號，起始傳送，還有終止傳送。

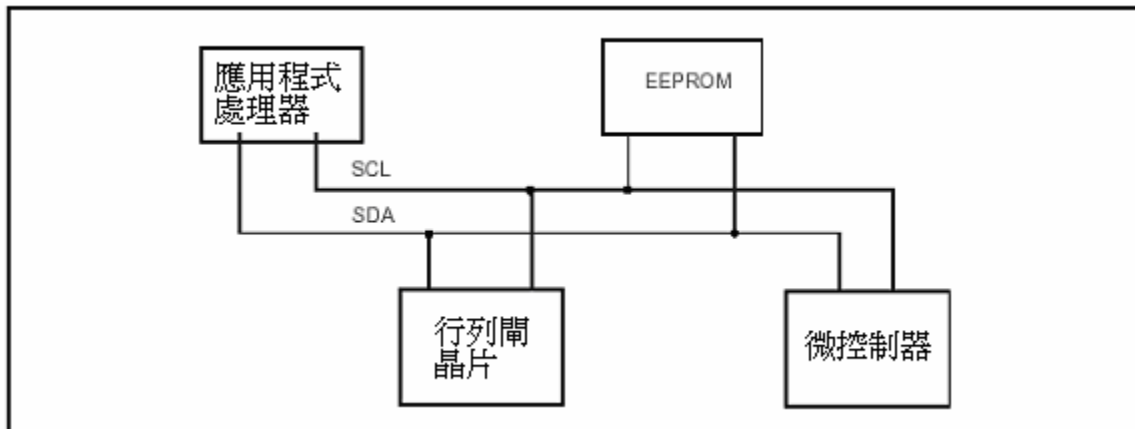


圖17-1 I2C匯流排規劃範例

I2C匯流排允許multi-master系統，意思是當多個裝置能同時起始資料傳輸。為了支援這項特性，I2C匯流排調停藉著硬體接線及閘連接所有I2C介面到I2C匯流排。二個masters能同時驅動匯流排，前提是他們所傳送的資料是相同的。假如一個master嘗試去驅動SDA高電位而另一個master驅動SDA為低電位時，則調停失敗。SCL線是一個由所有的master使用硬體接線及閘連接到SCL線所產生的同步時脈組合。

I2C匯流排序列操作使用一個接線及開匯流排架構，它允許多個裝置驅動匯流排線和傳送調停、等待狀態、錯誤情況等的事件狀態。例如，當master在資料傳送期間驅動時脈（SCL）線，它會在每個時脈是高電位的時候傳送一個位元。當slave不能接受

在master所要求的速率來驅動資料時，slave能在高電位狀態下插入等待狀態信號間隔來維持時脈線為低電位，master的時脈能夠只被另一個master在調停期間或慢速slave週邊設備維持時脈線為低電位時改變。

I2C動作不是由應用程式處理器當成master來啟動就是由應用程式處理器當作slave裝置來接收。上述的兩個狀況可能會導致讀取、寫入或讀寫到 I2C匯流排。

17.3.1 操作區塊

I2C單元是連接到周邊的匯流排，應用程式處理器使用中斷機制來通知CPU在I2C匯流排上的動作。中斷可以使用輪詢來取代。I2C單元是由二條接線介面到I2C匯流排，由一個8位元緩衝器來傳送與接收從應用程式處理器來的資料、一組控制與狀態暫存器和平行/序列轉換的移位暫存器所組成。

當緩衝器滿溢了、緩衝器是空的、檢測到I2C單元slave位址、調停失敗、或是匯流排錯誤情況發生時，I2C單元會初始啟動中斷應用程式處理器。所有中斷情況必須由軟體確實地清除，17.9.4節「I2C狀態暫存器」有詳細內容。

當接收到資料時，8位元I2C資料緩衝區暫存器（IDBR）從移位暫存器介面載入一個位元組的資料到I2C匯流排；當寫入資料時，8位元I2C資料緩衝區暫存器（IDBR）從應用程式處理器內部匯流排載入一個位元組的資料到I2C匯流排。序列移位暫存器使用者不可進行存取。

I2C控制暫存器（ICR）和I2C狀態暫存器（ISR）是儲存在I2C記憶體映像位址空間，暫存器和他們的功能定義在17.9節「暫存器定義」。

I2C單元支援以每秒400千位元的快速模式操作和以每秒100千位元的標準模式操作，參考*The I2C-Bus Specification*有詳細資料。

17.3.2 I2C匯流排介面模式

I2C單元能在不同的操作模式下完成傳輸，表17-3概述這些不同的模式。

表17-3 操作模式

模式	說明
Master傳送	<ul style="list-style-type: none"> · I2C單元作為master · 為寫入操作使用 · I2C單元傳送資料 · I2C單元回應時脈 · Slave裝置在slave接收模式下
Master接收	<ul style="list-style-type: none"> · I2C單元作為master · 為讀取操作使用 · I2C單元接收資料 · I2C單元回應時脈 · Slave裝置在slave傳送模式下
Slave傳送	<ul style="list-style-type: none"> · I2C單元作為slave · 為master讀取操作使用 · I2C單元傳送資料 · Master裝置在master接收模式下
Slave接收（預設）	<ul style="list-style-type: none"> · I2C單元作為slave · 為master寫入操作使用 · I2C單元接收資料 · Master裝置在master傳送模式下

當I2C單元為閒置時，它預設為slave接收模式。這樣允許介面去監視匯流排而且接收任何應用程式處理器的slave位址。

當I2C單元接收到一個位址與I2C Slave位址暫存器（ISAR）中的7位元位址或一般呼叫位址（見17.4.7節「一般呼叫位址」）相符合時，介面不是仍在slave接收模式下就是轉換到slave傳送模式。讀取/寫入位元（R/nW）將決定介面要進入哪一個模式，R/nW位元是包含有slave位址的位元組中的最低有效位元。假如R/nW位元是low，master開始動作想要寫入資料而且I2C單元仍然保持在slave接收模式下。假如R/nW為high，master開始動作想要讀取資料而且I2C單元將轉換到slave傳送模式下。17.4.6節「Slave操作」進一步定義slave操作。

當I2C單元在I2C匯流排上開始一個讀取或寫入時，它從預設的slave接收模式轉換到master傳送模式。假如動作是寫入，在位址傳輸完成後I2C單元仍在master傳送模式下。假如動作是讀取，I2C單元傳送起始位址，然後轉換到master接收模式下。17.4.5節「Master操作」進一步定義master操作。

17.3.3 START與STOP匯流排狀態

I2C匯流排規格書定義一個動作START，在傳輸一開始時使用；以及一個動作STOP匯流排狀態，在傳輸結束時使用。假如當SCL是高電位時，SDA線上發生高電位轉換到低電位的話，START狀況將發生。假如當SCL是高電位時，SDA線上發生低電位轉換到高電位的話，STOP狀況將發生。

I2C單元使用ICR[START]和ICR[STOP]位元來：

- 開始一個額外位元組的傳輸
- 在I2C匯流排上開始START狀況
- 資料鏈致能（重複START）
- 在I2C匯流排上開始STOP狀況

表17-4在ICR中定義START和STOP位元。

表17-4 START和STOP位元定義

STOP位元	START位元	狀況	注意
0	0	無START或STOP	I2C單元傳送一個沒有START或STOP狀況，當多個資料位元組需要傳輸時使用。
0	1	START狀況而且重複START	I2C單元傳送一個START狀況而且傳送8位元IDBR的內容，在一個START開始前IDBR必須包含7位元位址和R/nW位元。為了重複START，IDBR 包含有目標slave位址和R/nW位元，這樣允許一個master能執行多路傳輸到不同的slaves而不用讓出匯流排。介面為了寫入停留在master傳送模式而且為了讀取轉換到master接收模式。
1	X	STOP狀況	在master傳送模式下，I2C單元傳送8位元IDBR和在I2C匯流排上發送一個STOP狀況。在master接收模式下，ICR[ACKNAK]必須改變成否定ACK（見9.4.3節，“I2C認可”），I2C單元傳送NAK位元，在IDBR中接收資料位元組，還有在I2C匯流排上發送一個STOP狀況。

圖17-2為SDA和SCL線之間啟動與停止狀況的關係。

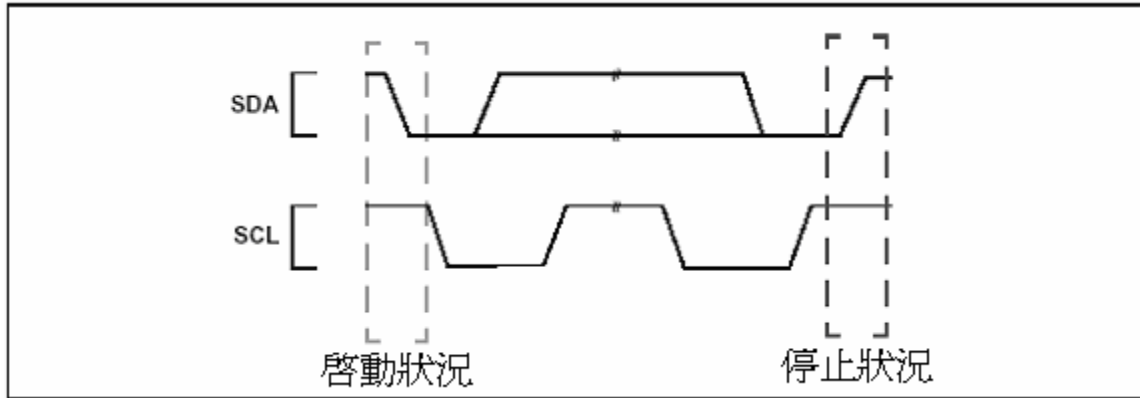


圖17-2 START和STOP狀況

17.3.3.1 START狀況

START狀況 ($ICR[START]=1$, $ICR[STOP]=0$) 開始一個master動作或重複START，在它設置START ICR位元前，軟體必須載入目標slave位址和IDBR中的R/nW位元 (見17.9.2節「I2C資料緩衝區暫存器 — IDBR」)。START和IDBR的內容在設置ICR[TB]位元後於I2C匯流排上傳輸，對於寫入要求，I2C匯流排停留在master傳送模式下；對於讀取要求，則進入master接收模式下。為了重複啟動、改變讀取或寫入或改變目標slave位址，IDBR包含了更新的目標slave位址和R/nW位元。一個重覆啟動可以使master執行多路傳輸到不同的slaves而不用讓出匯流排。

START狀況不是被I2C單元清除。當開始一個START時，假如I2C調停失敗，則當匯流排空閒時它可能會再嘗試START。見17.4.4節「調停」有在那些情況下I2C單元功能的詳細內容。

17.3.3.2 無START或STOP狀況

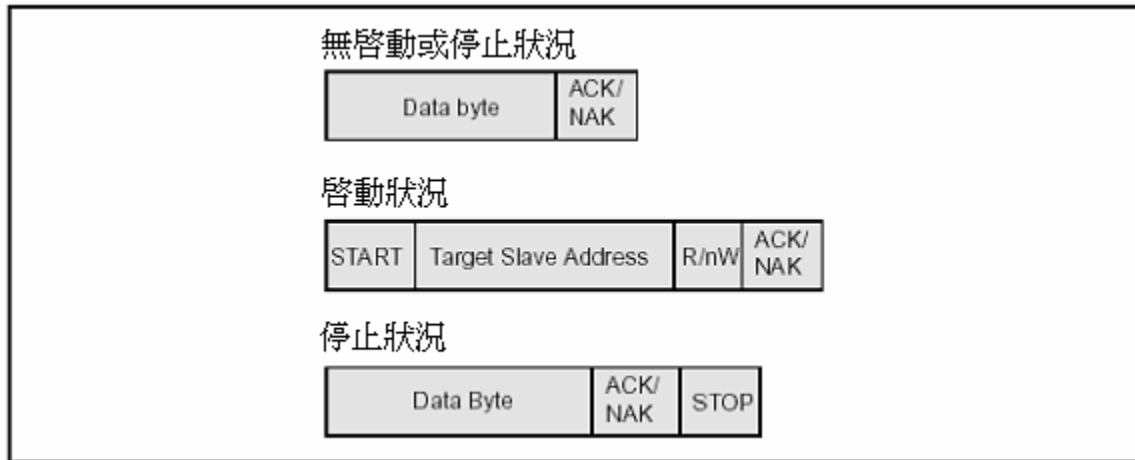
無START或STOP狀況 ($ICR[START]=0$, $ICR[STOP]=0$) 是使用在master傳送模式而I2C單元傳送多個資料位元組 (見圖17-2)。軟體寫入資料位元組而且I2C單元設置ISR[ITE]位元和清除ICR[TB]位元，軟體接下來寫入一個新位元組到IDBR並設置ICR[TB]位元來開始新位元組傳輸。這個步驟繼續到軟體設置ICR[START]或ICR[STOP]位元。ICR[START]和ICR[STOP]位元在START、STOP或重複START傳輸後不是由I2C單元自動清除。

在每個位元組傳輸後，包含ICR[ACKNAK]位元，I2C單元保持SCL線低電位來插入等待狀態直到設置ICR[TB]位元。這個動作通知I2C單元釋放SCL線而且允許下一個資訊傳送進行。

17.3.3.3 STOP狀況

STOP狀況 (ICR[START]=X, ICR[STOP]=1) 終止資料的傳輸。在master傳送模式下，ICR[STOP]位元和 ICR[TB]位元必須設置來開始最後位元組傳輸 (見圖 17-2)。在master接收模式下，I2C單元必須設置ICR[ACKNAK]位元、ICR[STOP]位元和ICR[TB]位元來開始最後的傳輸，在它傳送後軟體必須清除ICR[STOP]狀況

圖17-3 START和STOP狀況



17.4 I2C匯流排操作

I2C單元以1位元組增量來傳輸資料而且總是隨著下列順序：

1. START
2. 7位元Slave位址
3. R/nW位元
4. 認可脈衝
5. 8位元的資料
6. ACK/NAK脈衝
7. 重複步驟5和6給必須的位元組號碼
8. 重複START (重複步驟1) 或STOP

17.4.1 序列時脈線 (SCL) 之產生

當I2C單元在master傳送模式或master接收模式下，它產生I2C時脈輸出。SCL時脈藉由設置ICR[FM]位元為每秒100千位元或每秒400千位元操作來產生。

17.4.2 資料與定址管理

由I2C資料緩衝區暫存器 (IDBR) 和I2C Slave位址暫存器 (ISAR) 管理資料和定址slave。IDBR (見17.9.2節「I2C資料緩衝區暫存器 — IDBR」) 包含一個位元組的資料或7位元slave位址加上R/nW位元, ISAR 包含應用程式處理器可程式化slave位址。I2C單元在接收和認可全位元組後把接收的資料放進IDBR。為了傳送資料, CPU寫入到IDBR, 而且當設置ICR[TB]位元時I2C單元傳遞資訊到序列匯流排, 見17.9.3節「I2C控制暫存器 — ICR」。

當I2C單元在master或slave傳送模式下：

1. 軟體透過內部匯流排寫入資料到IDBR, 在設置ISR[ITE]位元後開始master動作或傳送下一個資料位元組
2. 當設置ICR[TB]位元時, I2C單元從IDBR傳送資料
3. 一個位元組在匯流排上傳輸而且認可週期完成時IDBR傳送清空的中斷信號以告知
4. 當I2C單元準備好傳輸下一個位元組, 而在CPU已經寫入IDBR並且還沒進入STOP狀況前, I2C單元插入等待狀態直到CPU寫入新的值到IDBR和設置ICR[TB]位元

當I2C單元在master或slave接收模式下：

1. 在IDBR接收到滿溢的中斷信號之後, 應用程式處理器透過內部匯流排讀取IDBR資料。
2. 在認可週期完成後I2C單元從移位暫存器傳輸資料到IDBR。
3. I2C單元插入等待狀態直到IDBR讀取, 參考17.4.3節「I2C認可」有在接收模式下的認可脈衝資訊。
4. 在CPU讀取IDBR之後, I2C單元寫入ICR[ACKNAK]位元和ICR[TB]位元, 允許下一個位元組資料傳輸。

17.4.2.1 定址Slave裝置

作為一個master裝置, I2C單元必須做組合和傳送第一個位元組的動作, 這個位元組由預期裝置的slave位址和動作定義的R/nW位元組成, 最高有效位元優先傳送, slave位址和R/nW位元寫入到IDBR (見圖17-4)。

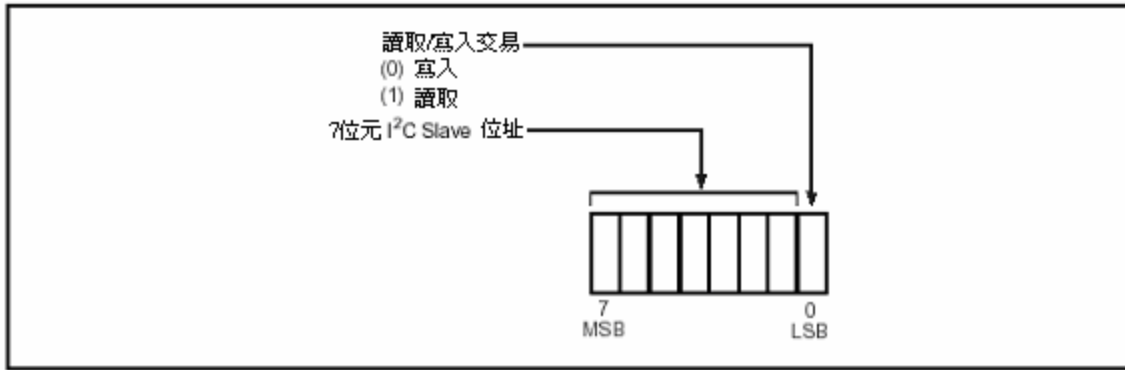


圖17-4 Master動作中第一個位元組的資料格式

第一個位元組的傳輸必須接在從定址slave的ACK脈衝之後，當動作是寫入時，I2C單元仍在master傳送模式而且定址slave裝置停留在slave接收模式下。當動作是讀取時，I2C單元隨著接收到ACK立即轉換到master接收模式而定址slave裝置轉換到slave傳送模式下。當傳回NAK時，I2C單元經由自動發送STOP和設置ISR[BED]位元來放棄動作。

當I2C單元致能且閒置時，它仍在slave接收模式下，它偵測匯流排上的START信號。當它檢測到START脈衝時，I2C單元讀取前七個位元，然後跟ISAR與一般呼叫位址（0x00）中的比對。當位元匹配ISAR暫存器中的內容時，I2C單元讀取第八個位元（R/nW位元）且傳送ACK脈衝，I2C單元不是仍在slave接收模式下（R/nW = 0）就是轉換到slave傳送模式下（R/nW = 1），見17.4.7節「一般呼叫位址」當檢測到一般呼叫位址時的作用。

17.4.3 I2C認可

每個I2C位元組傳輸必須伴隨在由master接收端或slave接收端所產生的認可脈衝。傳送者必須釋放SDA線給接收者傳送認可脈衝（見圖17-5）。

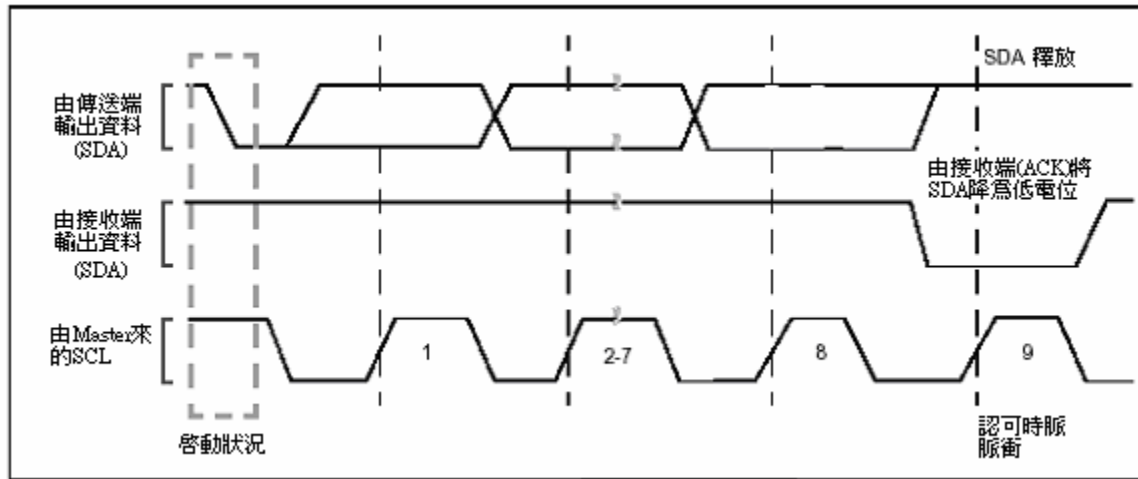


圖17-5 I2C匯流排上的認可

在master傳送模式下，假如目標slave接收裝置無法產生認可脈衝，SDA線仍為高電位。當啟動時缺少認可NAK會造成I2C單元設置ISR[BED]位元並產生關聯的中斷，I2C單元自動產生停止狀況並放棄動作。

在master接收模式下，I2C單元傳送沒有認可（NAK）來告知slave傳送器停止傳送資料，ICR[ACKNAK]位元控制I2C匯流排所驅動的ACK/NAK位元值。在I2C匯流排協定中，ISR[BED]位元並不是設置為master接收模式NAK。I2C單元自動傳送ACK脈衝在它從序列匯流排接收每個位元組之後。在單元接收到最後一個位元組前，軟體必須設置ICR[ACKNAK]位元為1（NAK），NAK脈衝在最後一個位元組指出最後一個位元組已經傳送之後才發送。

在slave模式下，I2C單元自動通知它的slave位址而不受ICR[ACKNAK]位元中的值支配。在slave接收模式下，ACK隨著一個資料位元組來自動回應而不受ICR[ACKNAK]位元中的值支配。I2C單元發送ACK的值在它接收到一個位元組中的第八個資料位元之後。

在slave傳送模式下，I2C單元從master接收到NAK來指示最後一個位元已經傳送，master隨後發送STOP或是重複START，ISR[UB]位元仍為設置直到接收到STOP或是重複START。

17.4.4 調停

I2C匯流排的multi-master功能需要I2C匯流排調停，當二或多個master在最短的停留時間產生START狀況時調停將發生。

調停可能會花很長的時間。假如位址位元跟R/nW一樣的話，調停系統會視為

資料，因為I2C匯流排有接線及閘，假如多個master發出同樣匯流排狀態的信號，傳輸不會遺失資料。假如他們包含的位址和R/nW位元或是資料

不一樣時，master發出高電位狀態來來表示調停失敗並關閉它的資料驅動器。假如I2C單元調停失敗，對於剩餘的位元傳輸它會關閉SDA或SCL的驅動器來停止，且設置ISR[ALD]位元，並回到slave接收模式。

17.4.4.1 SCL調停

每個I2C匯流排上的master為了在SCL線上的資料傳輸會產生它自己的時脈，因此結果是，可能會有不同頻率的時脈連接到SCL線，因為當時脈是在高電位期間時資料是有效的，位元接著位元的調停需要明確的時脈同步程序。

時脈同步是透過I2C介面到SCL線的硬體接線及閘連接。當master的時脈從高電位改變到低電位時，master在它的關聯執行期間保有SCL線（見圖17-6）。時脈不能從低電位轉換到高電位假如另一個master還沒完成它的執行期間，所以最長低電位期間的master使得SCL線維持為低電為，有較短執行期間的masters會在高電位等待狀態，直等到有最長期電位期間的master完成。在有最長低電位期間的master完成後，SCL線改變到高電位狀態而較短低電位期間的masters繼續資料週期。

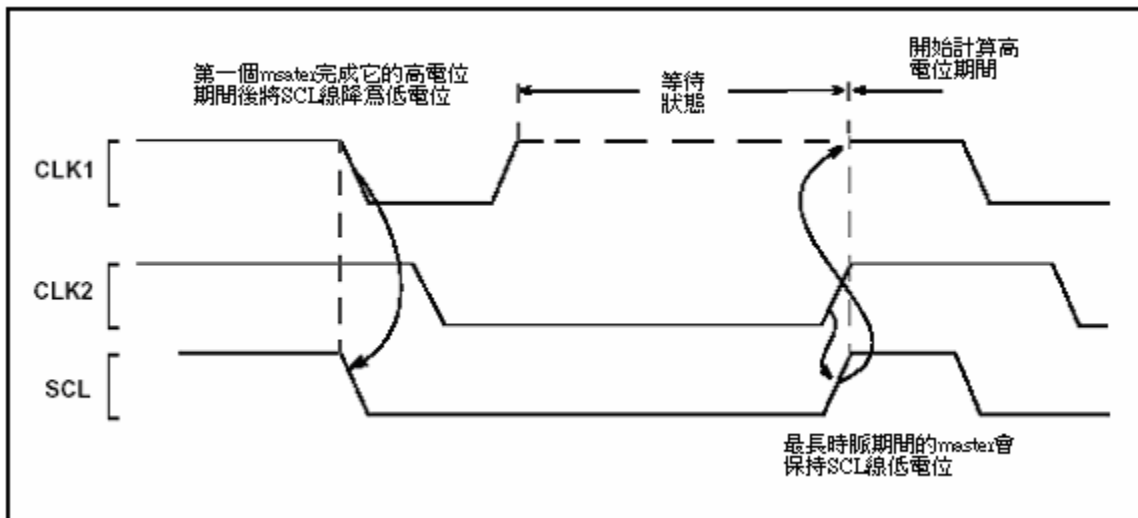


圖17-6 調停程序期間的時脈同步

17.4.4.2 SDA調停

在SDA線上的調停可以繼續一段長時間因為它由位址和R/nW位元開始且繼續透過資料位元，圖17-7有二個master的調停程序，假如二個以上的master連接到匯流排，二個以上的master可能會受干擾，假如位址位元跟R/nW一樣的話，調停系統會

認為是資料，因為I2C匯流排有接線及閘，假如多個master發出同樣匯流排狀態的信號，傳輸不會遺失資料。假如他們包含的位址和R/nW位元或是資料不一樣時，master發出第一個低電位資料位元來表示調停失敗並關閉它的資料驅動器。假如I2C單元調停失敗，對於剩餘的位元傳輸它會關閉SDA或SCL的驅動器來停止，且設置ISR[ALD]位元，並回到slave接收模式。

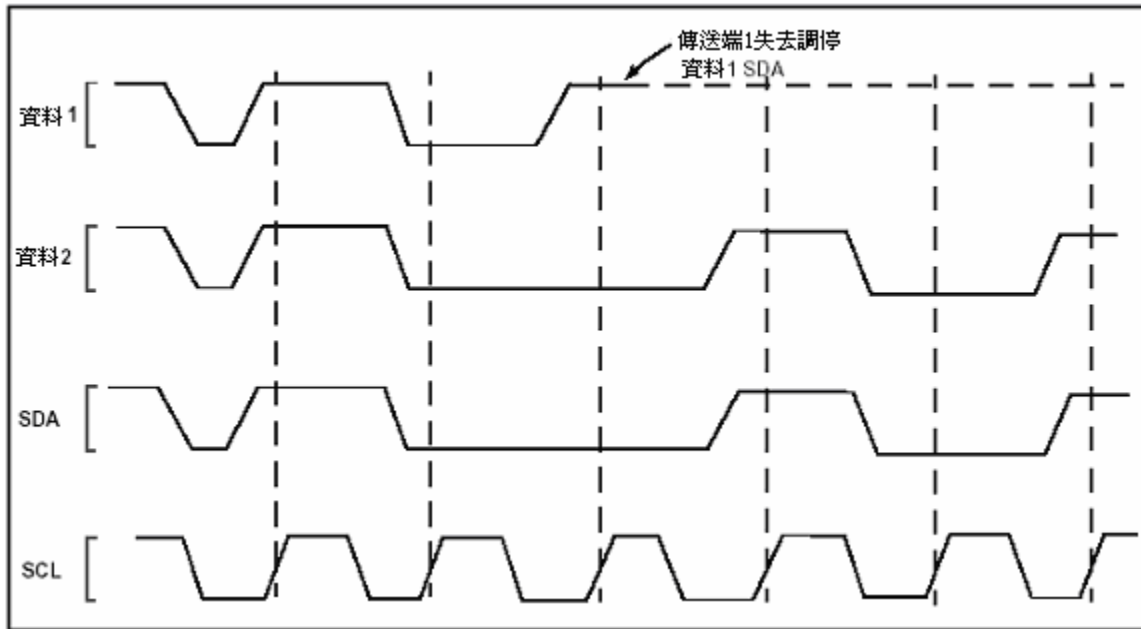


圖17-7 二個Master的調停程序

假如 I2C單元調停失敗是當位址位元傳送且它並不是由位址位元定址時，則I2C單元當匯流排空閒時重送位址。因為當調停失敗後IDBR和ICR暫存器不會被重新寫入，所以重送是可能的，。

假如I2C單元調停失敗是因為另一個匯流排master定址應用程式處理器為slave裝置，I2C單元轉換到slave接收模式且重寫I2C資料緩衝區暫存器中的原始資料，軟體能清除啟動和重新開始master動作。

注意：軟體必須預防I2C單元開始一個動作到它自己的slave位址，因為這樣的動作會造成I2C單元在一個不確定的狀態。

調停有邊界條件假使調停處理由重複START或STOP狀況在I2C匯流排上傳送產生中斷，為了避免錯誤，I2C單元作用為master假如沒有調停在下列的情況發生：

- 重複START狀況和資料位元之間
- 資料位元和STOP狀況之間
- 重複START狀況和STOP狀況之間

假如不同的master同時寫入同一筆資料到相同的目標而且調停在第一個資料位

元組傳輸之後無法解決時，這些情況將會發生。

注意：軟體保證調停會快速解決，例如，軟體能保證master發送唯一的資料由於需要每個master傳送它的I2C位址當成任何動作的第一個資料位元組。當調停被解決時，成功的master發送一個重新啟動且開始有效資料的傳輸，slave捨棄master的位址且使用其他資料。

17.4.5 Master操作

當軟體在I2C匯流排上開始進行讀取或寫入時，I2C單元從預設的slave接收模式轉換到master傳送模式，7個位元的slave位址和R/nW位元隨著啟動脈衝，在master接收到一個認可之後，I2C單元進入下列二種master模式之一：

- Master傳送 — I2C單元寫入資料
- Master接收 — I2C單元讀取資料

CPU寫到ICR暫存器來開始master動作，資料從I2C單元讀取和寫入到記憶體映像暫存器，表17-5說明了I2C單元作為master裝置的作用。

表17-5 Master動作

I2C Master 作用	操作模式	定義
一般時脈輸出	Master傳送 Master接收	·Master驅動SCL線 ·ICR[SCLE]位元必須設置 ·ICR[IUE] 位元必須設置
把目標slave位址寫入到IDBR	Master傳送 Master接收	·在START狀況啟動前CPU寫入到IDBR的7-1位元 ·START後前七個位元在匯流排上發送 ·見17.3.3節「START與STOP匯流排狀態」
把R/nW位元寫入到IDBR	Master傳送 Master接收	·CPU把目標slave位址寫入到IDBR的最低有效位元 ·假如是低電位，master仍是master傳送者；假如是高電位，master轉換到master接收 ·見17.4.2節「資料和定址管理」
發出START狀況信號	Master傳送 Master接收	·見前述的“產生時脈輸出” ·找出IDBR中slave位址和R/nW位元後執行。 ·軟體設置ICR[START]位元 ·軟體設置ICR[TB]位元來開始啟動狀況 ·見17.3.3節「START與STOP匯流排狀態」
開始第一個資料位元組傳輸	Master傳送 Master接收	·CPU寫入位元組到IDBR ·當ICR[TB]位元設置時，I2C單元傳送位元組 ·當傳輸完成時，I2C單元清除ICR[TB]位元和設置

		ISR[ITE]位元
I2C匯流排調停	Master傳送 Master接收	<ul style="list-style-type: none"> ·假如二個以上的master在同一個時脈期間發出啟動信號，則調停必須發生 ·I2C單元只要需要時就會調停，在slave位址和R/nW位元或是資料傳送和持續直到一個master失去匯流排時調停發生，沒有資料遺失 ·假如I2C單元調停失敗，它設置ISR[ALD]位元在位元組傳送完成之後且轉換到slave接收模式下 ·假如I2C單元調停失敗當它嘗試去發送目標位址位元組時，I2C單元當匯流排空閒時會嘗試重新發送 ·系統設計者必須保證邊界條件（17.4節「I2C匯流排操作」有描述）不會發生
寫入一個資料位元組到IDBR	只有Master傳送	<ul style="list-style-type: none"> ·I2C master操作資料傳送模式 ·當設置ISR[ITE]位元和清除ICR[TB]位元時發生，假如IDBR傳送清空的中斷致能時，它會發信號到應用程式處理器 ·CPU寫入一個資料位元組到IDBR，設置適當的START/STOP位元組合，且設置ICR[TB]位元來傳送資料。八個位元從移位暫存器取出然後寫入到序列匯流排，假如要求的話，八個位元隨在STOP之後
從slaver接收端等待認可	只有Master傳送	<ul style="list-style-type: none"> ·當master傳送端時，I2C單元為認可脈衝產生時脈，I2C單元釋放SDA線來允許slave接收端ACK傳送 ·見17.4.3節「I2C認可」
從IDBR讀取I2C資料中的一個位元組	只有Master接收	<ul style="list-style-type: none"> ·I2C master操作資料接收模式 ·從序列匯流排讀取八位元，收集到移位暫存器，然後在讀取到ICR[ACKNAK]位元後傳輸到IDBR ·當設置ISR[IRF]位元和清除ICR[TB]位元時，CPU讀取IDBR。假如IDBR接收到滿溢的中斷致能時，它會發出信號給CPU ·當讀取IDBR時，假如ISR[ACKNAK]是清除的（指出為ACK），應用程式處理器寫入ICR[ACKNAK]位元且設置ICR[TB]位元來開始下一個位元組讀取 ·假如ISR[ACKNAK]位元是設置的（指出為NAK），ICR[TB]位元是清除的，ICR[STOP]位元是設置的，和ISR[UB]位元是設置的，那麼最後一個資料位元組已經讀取到IDBR且I2C單元發送STOP ·假如ISR[ACKNAK]位元是設置的（指出為NAK），ICR[TB]位元是清除的，但ICR[STOP]位元是清除的，那麼CPU有二種選擇：1.設置ICR[START]位

		元，寫入一個新目標位址到IDBR，且設置ICR[TB]位元，它將發送重複START狀況或是。2.設置ICR[MA]位元且交給ICR[TB]位元清除，它將只發送STOP
傳送認可到slave傳送端	只有Master接收	<ul style="list-style-type: none"> 當master接收端時，I2C單元會為認可脈衝產生時脈，I2C單元在ACK週期期間也會驅動SDA線來回應 假如下一個資料位元組是最後一個動作，CPU會設置ICR[ACKNAK]位元來產生NAK 見17.4.3節「I2C認可」
產生一個重複START來約束I2C動作	Master傳送 Master接收	<ul style="list-style-type: none"> 假如期待一串資料鏈時，重複START狀況會使用來取代STOP狀況 這會發生在動作的最後一個資料位元組已經寫入到匯流排後 CPU將會寫入下一個目標slave位址和R/nW位元到IDBR，設置ICR[START]位元和設置ICR[TB]位元 見17.3.3節「START和STOP匯流排狀態」
產生一個STOP	Master傳送 Master接收	<ul style="list-style-type: none"> 在CPU透過匯流排寫入最後一個資料位元組之後產生 CPU藉由設置ICR[STOP]位元來產生STOP狀況 見17.3.3節「START和STOP匯流排狀態」

當CPU需要讀取資料時，I2C單元從slave接收模式轉換到master傳送模式來傳送起始位址、R/nW位元和ACK脈衝。在它發送ACK脈衝之後，I2C單元轉換到master接收模式並且等待接收從slave裝置來的讀取資料（見圖17-8）。多重動作在I2C操作期間會發生，例如，透過重複啟動從master接收轉換到master傳送。

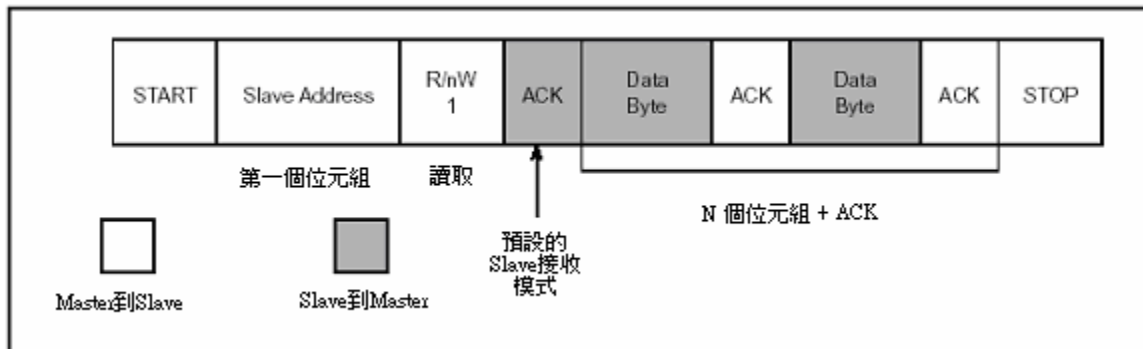


圖17-8 Master接收端從Slave傳送端讀取

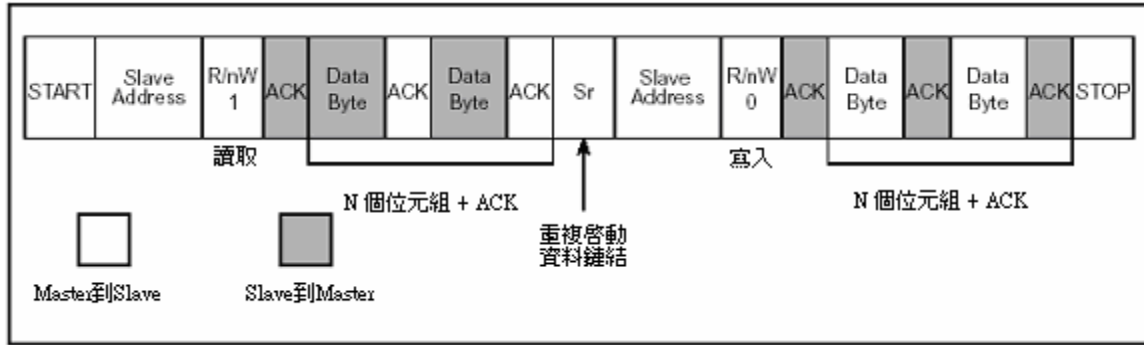


圖17-9 Master接收端從Slave傳送端/重複START/Master傳送端寫入到Slave接收端

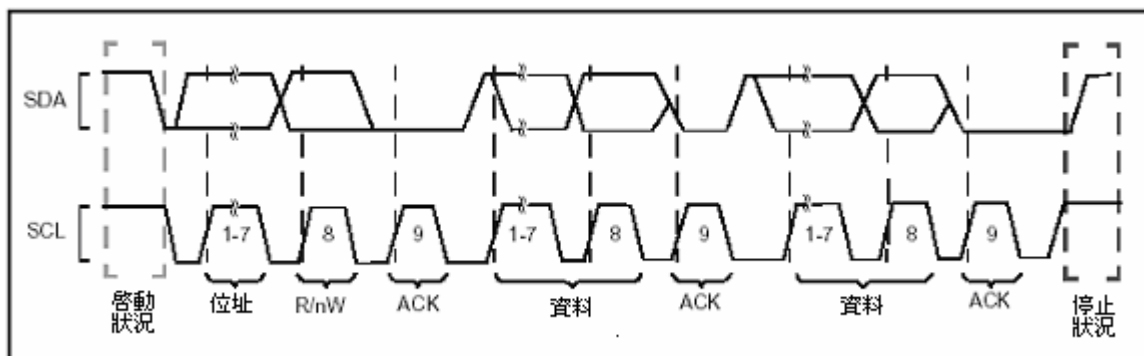


圖17-10 完整的資料傳輸

17.4.6 Slave 操作

表17-6描述I2C單元如何操作為slave裝置。

表17-6 Slave動作

I2C Slave 作用	操作模式	定義
Slave接收 (預設模式)	只有 Slave接收	<ul style="list-style-type: none"> ·I2C單元監視所有slave位址動作 ·ICR[IUE]位元必須設置 ·I2C單元監測匯流排START狀況，當檢測到START時，介面讀取前八個位元並且跟ISAR的7個最高有效位元和一般呼叫位址(0x00)作比對，假如相符合的話，I2C單元發送一個ACK ·假如前八個位元都是零的話，這是一般呼叫位址，假如ICR[GCD]位元是清除的，ISR[GCAD]位元和ISR[SAD]位元都將被設置，見17.4.7節「一般呼叫位址」 ·假如第一個位元組的第八個位元(R/nW位元)是低電位，I2C單元停留在slave接收模式並且ISR[SAD]位元將

		被清除，假如R/nW位元是高電位，I2C單元轉換到slave傳送並且ISR[SAD]位元將被設置
設置Slave位址檢測位元	Slave接收Slave傳送	<ul style="list-style-type: none"> 指出介面已經檢測到I2C操作定址應用程式處理器，包括一般呼叫位址，應用程式處理器能夠識別ISAR藉著從一般呼叫讀取到的ISR[GCAD]位元相比對 在接收且認可要比對的slave位址後，一個中斷發出信號（假如中斷是致能的話）
從IDBR讀取I2C資料中的一個位元組	只有Slave接收	<ul style="list-style-type: none"> I2C slave操作的資料接收模式 從序列匯流排讀取八位元到移位暫存器，當接收到完整的位元組且ACK/NAK位元完成時，位元組從移位暫存器傳輸到IDBR 當設置ISR[IRF]位元且清除ICR[TB]位元時發生，IDBR接收滿溢的中斷發出信號到CPU（假如中斷是致能的話） 軟體從IDBR讀取一個資料位元組，當讀取IDBR時，應用程式處理器寫入需要的ICR[ACKNAK]位元和設置ICR[TB]位元，這樣造成I2C單元停止插入等待狀態並且使master傳送端寫入下一段資訊
傳送認可到master傳送端	只有Slave接收	<ul style="list-style-type: none"> 當slave接收端時，I2C單元在SCL高電位時期使SDA線為低電位來產生ACK脈衝 ICR[ACKNAK]位元控制I2C單元驅動的ACK資料，見17.4.3節「I2C認可」
寫入一個位元組的I2C資料到IDBR	只有Slave傳送	<ul style="list-style-type: none"> I2C slave操作的資料傳輸模式 當設置ISR[ITE]位元且清除ICR[TB]位元時發生，IDBR傳送清空的中斷來發出信號到應用程式處理器（假如中斷是致能的話） 應用程式處理器寫入資料位元組到IDBR且設置ICR[TB]位元來開始傳輸
從master接收端等待認可	只有Slave傳送	<ul style="list-style-type: none"> 當slave傳送端時，I2C單元釋放SDA線來允許master接收端把線降低到電位來接收 見17.4.3節「I2C認可」

圖17-11到圖17-13是I2C動作的範例和master與slave裝置之間的關係。

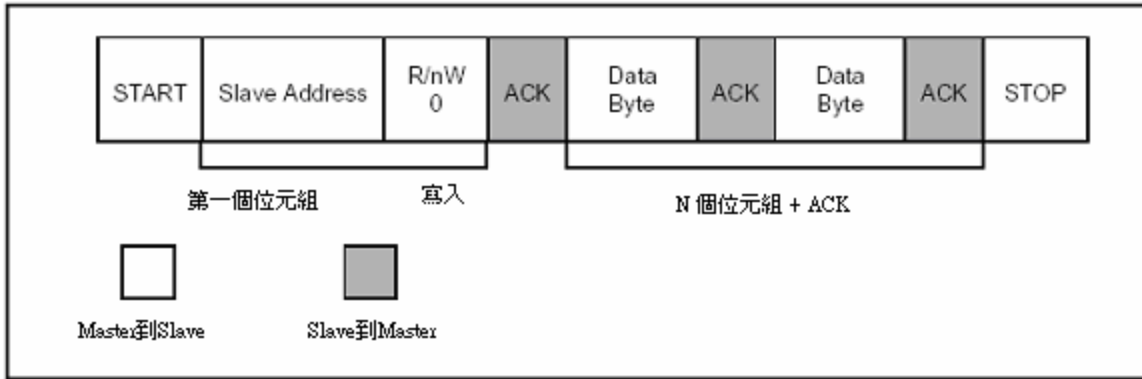


圖17-11 Master傳送端寫入到Slave接收端

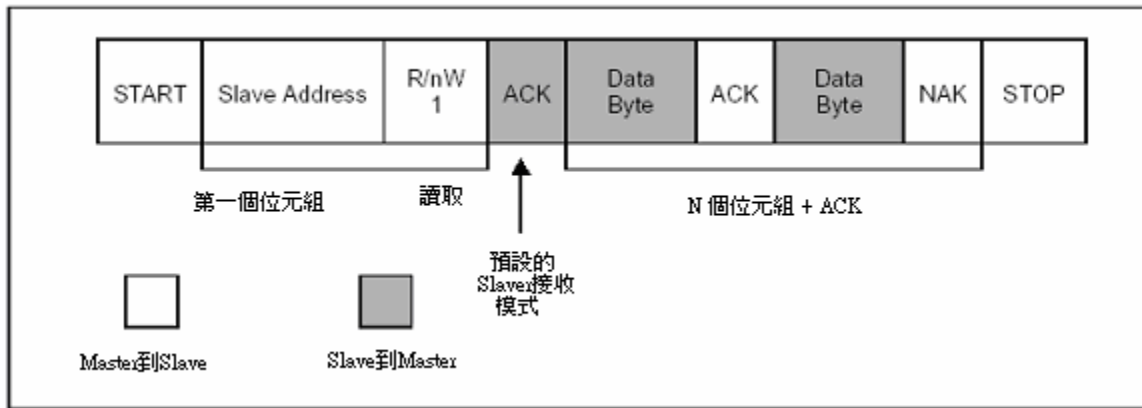


圖17-12 Master接收端讀取到Slave傳送端

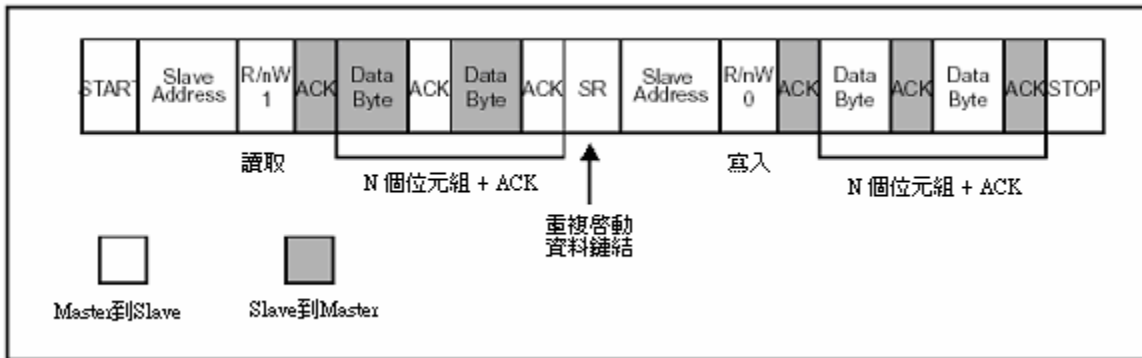


圖17-13 Master接收端讀取到Slave傳送端、重複啟動、Master傳送端寫入到Slave接收端

17.4.7 一般呼叫位址

一般呼叫位址是以slave位址0x00來進行動作，當一個裝置需要從一般呼叫位址來的資料時，它會認可動作且停留在slave接收端模式，否則裝置會忽略一般呼叫位址。一般呼叫動作中的其他位元組是由每個在匯流排上使用它的裝置來認可，裝置不使用這些位元組時不必發送ACK，這表示一般呼叫位址是定義在第二個位元組而是由master傳送端發送。

圖17-14為一般呼叫位址動作，第二個位元組的最低有效位元命名為B，定義為動作，表17-7指出當B=0時的有效值與定義。

I2C單元支援在I2C匯流排上發送和接收一般呼叫位址的傳輸，當軟體從I2C單元發送一般呼叫訊息時，它必須設置ICR[GCD]位元來避免I2C單元當成slave來回應，假如ICR[GCD]沒有設置，I2C匯流排進入一個未知狀態。

假如I2C單元作用為slave且接收一般呼叫位址當ICR[GCD]位元清除時，它會：

- 設置ISR[GCAD]位元
- 設置ISR[SAD]位元
- 中斷應用程式處理器（假如中斷致能時）

假如I2C單元接收到一般呼叫位址且ICR[GCD]位元設置時，它會忽略一般呼叫位址。

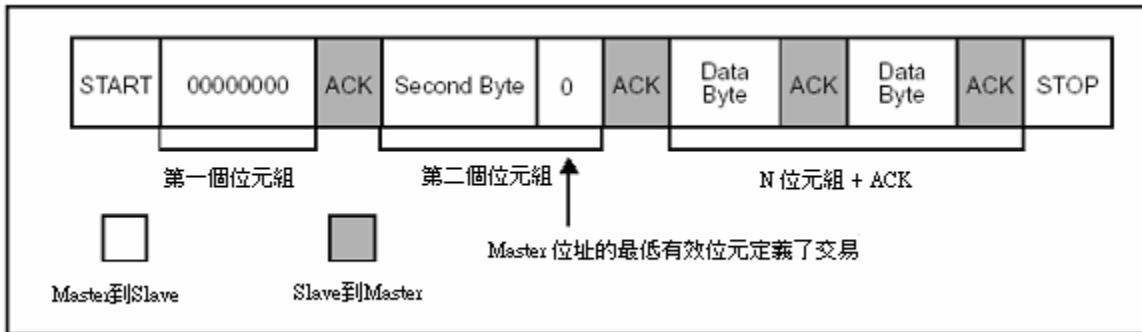


圖17-14 一般呼叫位址

表17-7 呼叫位址第二個位元組定義

第二個位元組的最低有效位元 (B)	第二個位元組值	定義
0	0x06	第二個位元組的2位元組動作告訴slave重置並儲存這個值到它位址可程式化部分
0	0x04	第二個位元組的2位元組動作告訴slave儲存這個值

		到它位址可程式化部分。沒有重置
0	0x00	不允許為第二個位元組
注意： 其他值並不是固定的，而且必須忽略		

軟體必須保證在I2C單元在觸發重置之前它不是忙碌的，軟體也必須保證I2C匯流排是空閒的當單元在重置之後是啟動的。當直接重置時，除了ISAR之外的I2C單元，回到預設的重置狀態，ISAR不會受重置影響。

當B=1時，順序是硬體一般呼叫且I2C單元不支援，參考*The I2C-Bus Specification*有硬體一般呼叫的資訊。

I2C 10位元位址和CBUS相容是不支援的。

17.5 Slave模式程式設計範例

17.5.1 初始單元

1. 設置ISAR中的slave位址
2. 啟動ICR中要求的中斷
3. 設置ICR[IUE]位元來啟動I2C單元

17.5.2 寫入n個位元組為Slave

1. 當Slave位址檢測到中斷發生時，讀取ISR:Slave位址檢測 (1)、單元忙碌 (1)、R/nW位元 (1)、ACK/NAK (0)
2. 寫入1到ISR[SAD]位元來清除中斷
3. 從中斷返回
4. 載入IDBR中的資料位元組來傳輸
5. 設置ICR[TB]位元
6. 當IDBR傳送空的中斷發生，讀取ISR:IDBR傳送空的 (1)、ACK/NAK (0)、R/nW位元 (0)
7. 載入IDBR中的資料位元組來傳輸
8. 設置ICR[TB]位元
9. 寫入1到ISR[ITE]位元來清除中斷
10. 從中斷返回
11. 重複步驟6到10 n-1次。假如slave在任何時間內都沒有資料，I2C單元保持SCL在低電位直到有可用的資料
12. 當IDBR傳送空的中斷發生，讀取ISR:IDBR傳送空的 (1)、ACK/NAK (1)、R/nW位元 (0)

13. 寫入1到ISR[ITE]位元來清除中斷
14. 從中斷返回
15. 當Slave停止檢測中斷發生，讀取ISR:單元忙碌(0)、Slave停止檢測(1)
16. 寫入1到ISR[SSD]位元來清除中斷

17.5.3 讀取 n 個位元組為Slave

1. 當Slave位址檢測中斷發生，讀取ISR:Slave位址檢測(1)、單元忙碌(1)、R/nW位元(0)
2. 寫入1到ISR[SAD]位元來清除中斷
3. 從中斷返回
4. 設置ICR[TB]位元來開始傳輸
5. 當IDBR接收到滿溢的中斷發生，讀取ISR: IDBR接收滿溢(1)、ACK/NAK(0)、R/nW位元(0)
6. 讀取IDBR來得到接收的位元組
7. 寫入1到ISR[IRF]位元來清除中斷
8. 從中斷返回
9. 重複步驟4到8 $n-1$ 次，一旦IDBR滿溢時，I2C單元將保持SCL在低電位直到讀取資料
10. 設置ICR[TB]位元來釋放I2C匯流排且允許下一個傳輸
11. 當Slave停止檢測中斷發生，讀取ISR:單元忙碌(0)、Slave停止檢測(1)
12. 寫入1到ISR[SSD]位元來清除中斷

17.6 Master程式設計範例

17.6.1 初始單元

1. 設置ISAR中的slave位址
2. 啟動ICR中要求的中斷，不要啟動調停失去檢測中斷
3. 設置ICR[IUE]和ICR[SCLE]位元來啟動I2C單元與SCL

17.6.2 寫入1個位元組為Master

1. 載入IDBR中的目標slave位址和R/nW位元，R/nW為寫入必須設定為0
2. 開始寫入：設置ICR[START]，清除ICR[STOP]，清除ICR[ALDIE]，設置ICR[TB]
3. 當IDBR傳送清空的中斷發生，讀取ISR:IDBR傳送空的(1)，單元忙碌(1)，R/nW位元(0)
4. 寫入1到ISR[ITE]位元來清除中斷

5. 假如設置的話寫入1到ISR[ALD]位元。假如master調停失敗，它當匯流排變成空閒時執行位址重試，調停失敗檢測中斷不允許位址重試
6. 載入IDBR中的資料位元組傳輸
7. 開始寫入：清除 ICR[START]，設置ICR[STOP]，設置ICR[ALDIE]，設置ICR[TB]
8. 當IDBR傳送空的中斷發生（單元發送STOP），讀取ISR:IDBR傳送空的（1），單元忙碌（x），R/nW位元（0）
9. 寫入1到ISR[ITE]位元來清除中斷
10. 清除ICR[STOP]位元

17.6.3 讀取1個位元組為Master

1. 載入IDBR中的目標slave位址和R/nW位元，R/nW為讀取必須設定為1
2. 開始寫入：設置ICR[START]，清除ICR[STOP]，清除ICR[ALDIE]，設置ICR[TB]
3. 當IDBR傳送清空的中斷發生，讀取ISR: IDBR傳送空的（1），單元忙碌（1），R/nW位元（1）
4. 寫入1到ISR[ITE]位元來清除中斷
5. 開始讀取：清除 ICR[START]，設置 ICR[STOP]，設置 ICR[ALDIE]，設置 ICR[ACKNAK]，設置ICR[TB]
6. 當IDBR接收到滿溢的中斷發生（單元發送STOP），讀取ISR: IDBR接收滿溢（1），單元忙碌（x），R/nW位元（1），ACK/NAK位元（1）
7. 寫入1到ISR[IRF]位元來清除中斷
8. 讀取IDBR資料
9. 清除ICR[STOP]和ICR[ACKNAK]位元

17.6.4 寫入2個位元組與重複START讀取1個位元組為Master

1. 載入IDBR中的目標slave位址和R/nW位元，R/nW為寫入必須設定為0
2. 開始寫入：設置ICR[START]，清除ICR[STOP]，清除ICR[ALDIE]，設置ICR[TB]
3. 當IDBR傳送清空的中斷發生，讀取ISR:IDBR傳送空的（1），單元忙碌（1），R/nW位元（0）
4. 寫入1到ISR[ITE]位元來清除中斷
5. 載入IDBR中的資料位元組來傳輸
6. 開始寫入：清除ICR[START]，清除ICR[STOP]，設置ICR[ALDIE]，設置ICR[TB]
7. 當IDBR傳送清空的中斷發生，讀取ISR:IDBR傳送空的（1），單元忙碌（1），R/nW位元（0）
8. 寫入1到ISR[ITE]位元來清除中斷
9. 重複步驟5-8一次
10. 載入IDBR中的目標slave位址和R/nW位元，R/nW為讀取必須設定為1
11. 發送重複START當作master，設置ICR[START]，清除ICR[STOP]，清除

- ICR[ALDIE]，設置ICR[TB]
12. 當IDBR傳送清空的中斷發生，讀取ISR:IDBR傳送空的(1)，單元忙碌(1)，R/nW位元(1)
 13. 寫入1到ISR[ITE]位元來清除中斷
 14. 開始讀取：清除ICR[START]，設置ICR[STOP]，設置ICR[ALDIE]，設置ICR[ACKNAK]，設置ICR[TB]
 15. 當IDBR接收到滿溢的中斷發生(單元發送STOP)，讀取ISR:IDBR接收滿溢(1)，單元忙碌(x)，R/nW位元(1)，ACK/NAK位元(1)
 16. 寫入1到ISR[IRF]位元來清除中斷
 17. 讀取IDBR資料
 18. 清除ICR[STOP]和ICR[ACKNAK]位元

17.6.5 讀取2個位元組為Master — 使用放棄傳送STOP

1. 載入IDBR中的目標slave位址和R/nW位元，R/nW為讀取必須設定為1
2. 開始寫入：設置ICR[START]，清除ICR[STOP]，清除ICR[ALDIE]，設置ICR[TB]
3. 當IDBR傳送清空的中斷發生，讀取ISR:IDBR傳送空的(1)，單元忙碌(1)，R/nW位元(1)
4. 寫入1到ISR[ITE]位元來清除中斷
5. 開始讀取：清除ICR[START]，清除ICR[STOP]，設置ICR[ALDIE]，清除ICR[ACKNAK]，設置ICR[TB]
6. 當IDBR接收到滿溢的中斷發生，讀取ISR:IDBR接收滿溢(1)，單元忙碌(1)，R/nW位元(1)，ACK/NAK位元(0)
7. 寫入1到ISR[IRF]位元來清除中斷
8. 讀取IDBR資料
9. 清除ICR[STOP]和ICR[ACKNAK]位元
10. 開始讀取：清除ICR[START]，清除ICR[STOP]，設置ICR[ALDIE]，設置ICR[ACKNAK]，設置ICR[TB]，ICR[STOP]沒設置是因為STOP或重複START將決定位元組讀取
11. 當IDBR接收到滿溢的中斷發生，讀取ISR:IDBR接收滿溢(1)，單元忙碌(1)，R/nW位元(1)，ACK/NAK位元(1)
12. 寫入1到ISR[IRF]位元來清除中斷
13. 讀取IDBR資料
14. 開始停止放棄狀況(STOP則無資料傳輸)，設置ICR[MA]

注意：假如NAK沒在步驟11發送的話，下一個動作必須包含另一個資料位元組讀取

17.7 雜訊抑制邏輯

I2C單元有內建雜訊抑制邏輯來抑制小於等於60ns的雜訊，這是在50ns範圍內的雜訊抑制規格。

17.8 重置條件

軟體必須保證在I2C單元觸發重置前它不是忙碌的，軟體也必須保證當單元啟動且重置之後I2C匯流排是空閒的。當直接重置時，除了ISAR之外的I2C單元，回到預設的重置狀態，ISAR不會受重置影響。

當ICR[UR]位元設置時，I2C單元重置但關聯的I2C MMRs仍不受改變，當以ICR的單元重置來重置I2C單元時，使用下列方針：

1. 設置ICR暫存器中的重置位元且清除暫存器中的餘物
2. 清除ISR暫存器
3. 清除ICR中的重置

17.9 暫存器定義

在表17-8中的暫存器關聯到I2C單元且定位於應用程式處理器週邊記憶體映像位址空間。

表17-8 I2C暫存器位址

暫存器位址	名稱	節
0x4030 1680	IBMR	17.9.1節「I2C匯流排監視暫存器 — IBMR」
0x4030 1688	IDBR	17.9.2節「I2C資料緩衝區暫存器 — IDBR」
0x4030 1690	ICR	17.9.3節「I2C控制暫存器 — ICR」
0x4030 1698	ISR	17.9.4節「I2C狀態暫存器」
0x4030 16A0	ISAR	17.9.5節「I2CSlave位址暫存器 — ISAR」

17.9.1 I2C匯流排監視暫存器 — IBMR

I2C匯流排監視暫存器 (IBMR) 追蹤SCL和SDA腳位的狀態，這些腳位的值記錄在唯讀的IBMR中，所以當I2C匯流排中止且I2C單元必須重置時軟體可以決定。

表17-9 I2C匯流排監視暫存器 — IBMR

實體位址 4030_1680		I ² C 匯流排監視暫存器																I ² C														
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																SCLS	SDAS														
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	1	1
31:2	Reserved		保留																													
1	SCLS		SCL狀態：這個位元持續反應SCL腳位的值																													
0	SDAS		SDA狀態：這個位元持續反應SDA腳位的值																													

17.9.2 I2C資料緩衝暫存器 — IDBR

應用程式處理器使用I2C資料緩衝暫存器從I2C匯流排來傳送和接收資料，IDBR由一邊的程式I/O和另一邊的I2C移位暫存器來存取，IDBR在接收且認可一個完整位元組後從I2C單元接收資料，應用程式處理器核心從I2C單元寫入資料到IDBR並發送它到序列匯流排。

當I2C單元在傳送模式下時 (master或slave)，應用程式處理器透過內部匯流排寫入資料到IDBR，當master動作開始時或是當IDBR傳送空的中斷來發出信號時，應用程式處理器寫入資料到IDBR。當傳輸位元組位元設置時，資料從IDBR移動到移位暫存器，當一個位元組在I2C匯流排上傳輸且認可週期完成時，IDBR傳送空的中斷來發出信號 (假如已啟動)。假如IDBR在I2C匯流排準備就緒傳輸下一個位元組封包前還沒被應用程式處理器寫入而且尚未進入停止狀況時，I2C單元插入等待狀態直到應用程式處理器寫入到IDBR且設置傳輸位元組位元。

當I2C單元在接收模式下 (master或slave)，應用程式處理器透過內部匯流排讀取IDBR資料，當IDBR接收到滿溢的中斷信號時，應用程式處理器從IDBR讀取資料。當ACK週期完成時，資料從移位暫存器移動到IDBR，I2C單元插入等待狀態直到讀取IDBR，參考17.4.3節「I2C認可」有更多在接收模式下的認可脈衝資訊。在應用程式處理器讀取IDBR之後，ACK/NAK控制位元被寫入且傳輸位元組也被寫入，來允許下一個位元組繼續傳輸到I2C匯流排，IDBR暫存器在重置後為0x00。

表17-10 I2C資料緩衝暫存器 — IDBR

實體位址 4030_1688		i ² C 資料緩衝暫存器		i ² C													
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										IDB					
重置	0 0																
31:8	Reserved	保留															
7:0	IDB	I2C資料緩衝區：為I2C匯流排傳送/接收資料的緩衝區															

17.9.3 I2C控制暫存器 — ICR

應用程式處理器使用I2C控制暫存器（ICR）中的位元來控制I2C單元。

表17-11 I2C控制暫存器 — ICR

實體位址 4030_1690		i ² C 控制暫存器		i ² C																							
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										FM	UR	SADIE	ALDIE	SSDIE	BEIE	IRFIE	ITEIE	GCD	IUE	SCLE	MA	TB	ACKNAK	STOP	START
重置	0 0																										
31:16	Reserved	保留																									
15	FM	速度模式： 0 = 100千位元/秒操作 1 = 400千位元/秒操作																									
14	UR	單元重置： 0 = 不重置 1 = 只重置I2C單元																									
13	SADIE	Slave位址檢測中斷致能： 0 = 中止中斷 1 = 啟動I2C單元來中斷應用程式處理器當它檢測到符合slave位址或是一般呼叫位址																									
12	ALDIE	調停失去檢測中斷致能： 0 = 中止中斷 1 = 啟動I2C單元來中斷應用程式處理器當它在master模式下調停失敗																									
11	SSDIE	Slave停止檢測中斷致能： 0 = 中止中斷 1 = 啟動I2C單元來中斷應用程式處理器當它在slave模式下檢測到STOP狀況																									
10	BEIE	匯流排錯誤中斷致能：																									

		<p>0 = 中止中斷</p> <p>1 = 因為下列I2C匯流排錯誤所以啟動I2C單元來中斷應用程式處理器：</p> <ul style="list-style-type: none"> ·當作master傳送端時，在一個位元組傳送後沒有檢測ACK ·當作slave接收端時，I2C單元產生NAK脈衝 <p>注意：軟體回應保證START和STOP狀況錯位情況不會發生，見17.7節「雜訊抑制邏輯」</p>
9	IRFIE	<p>IDBR接收滿溢中斷啟動：</p> <p>0 = 中止中斷</p> <p>1 = 當IDBR從I2C匯流排接收資料位元組時，啟動I2C單元來中斷應用程式處理器</p>
8	ITEIE	<p>IDBR傳送空的中斷啟動：</p> <p>0 = 中止中斷</p> <p>1 = 在傳送一個位元組到I2C匯流排後，啟動I2C單元來中斷應用程式處理器</p>
7	GCD	<p>一般呼叫中止：</p> <p>0 = 啟動I2C單元來回應一般呼叫訊息</p> <p>1 = 中止I2C單元當作slave來回應一般呼叫訊息</p> <p>當I2C單元發送master模式的一般呼叫訊息時必須設置</p>
6	IUE	<p>I2C單元致能：</p> <p>0 = 中止單元且不回應任何master動作或回應任何slave動作</p> <p>1 = 啟動I2C單元（預設為slave接收模式）</p> <p>軟體必須保證在它設置這個位元前I2C匯流排是閒置的</p>
5	SCLE	<p>SCL致能：</p> <p>0 = 由驅動SCL線來中止I2C單元</p> <p>1 = 啟動master模式操作的I2C時脈輸出</p>
4	MA	<p>Master中止：當I2C單元在master模式下，產生一個停止而不用傳送另一個資料位元</p> <p>0 = I2C單元只使用STOP ICR位元來傳送STOP</p> <p>1 = I2C單元發送STOP而沒有資料傳輸</p> <p>在master傳送模式下，在資料位元傳送後，ICR的傳輸位元組位元是清除的且IDBR傳送空的位元是設置的。當沒有多餘的資料位元組需要傳送時，設置master中止位元來發送STOP，傳輸位元組位元（03）必須仍為清除。</p> <p>在master接收模式下，當發送一個NAK而沒有一個STOP（STOP ICR位元沒有設置）且應用程式處理器沒有發送重複START時，再一次設置這個位元來發送STOP，傳輸位元組位元（03）必須仍為清除。</p>
3	TB	<p>傳輸位元組：使用在I2C匯流排上傳送/接收一個位元組</p> <p>0 = 當位元組傳送/接收時由I2C單元清除</p> <p>1 = 傳送/接收一個位元組</p>

		當位元組傳輸完成時，應用程式處理器能監視這個位元來決定，在master或slave模式下，在每個位元組（包含ACK/NAK位元）傳輸後，I2C單元保持SCL線低電位（插入等待狀態）直到傳輸位元組是設置的
2	ACKNAK	ACK/NAK控制 ：定義當在master接收模式下，I2C單元發送的ACK脈衝型式 0 = I2C單元在接收到一個資料位元組後發送ACK脈衝 1 = I2C單元在接收到一個資料位元組後發送否定的ACK（NAK） I2C單元自動發送ACK脈衝當它回應到它的slave位址或當它回應到slave接收模式時，而不受ACK/NAK控制位元設置的影響
1	STOP	STOP ：在master模式I2C匯流排上的資料傳輸後，開始STOP狀況，在master接收模式下，ACK/NAK控制位元必須和這個位元一起設置，見17.3.3.3節「STOP狀況」有STOP狀態的詳細內容 0 = 不發送一個STOP 1 = 發送一個STOP
0	START	START ：當在master模式時，開始START狀況到I2C單元模式，見17.3.3.1節「START狀況」有START狀態的詳細內容 0 = 不發送一個START 1 = 發送一個START

17.9.4 I2C狀態暫存器

ISR發出I2C中斷信號到應用程式處理器中斷控制器，軟體可以使用ISR位元來核對I2C單元與匯流排的狀態，ISR位元（位元9-5）在I2C匯流排上完成ACK/NAK位元之後更新。

ISR也清除下列從I2C單元來的中斷信號：

- IDBR接收滿溢
- IDBR傳送空的
- Slave位址檢測
- 匯流排錯誤檢測
- STOP狀況檢測
- 調停失敗

表17-12 I2C狀態暫存器 — ISR

實體位址 4030_1698		I2C 狀態暫存器	I2C																									
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																BED	SAD	GCAD	IRF	ITE	ALD	SSD	IBB	UB	ACKNAK	RWM
重置	0 0																0	0	0	0	0	0	0	0	0	0	0	0
31:11	Reserved	保留																										
10	BED	匯流排錯誤檢測： 0 = 沒有檢測到錯誤 1 = I2C單元設置這個位元當它檢測到下列其中一個錯誤情況： ·當作master傳送端時，在一個位元組傳送後，沒有在介面上檢測到ACK ·當作slave接收端時，I2C單元產生NAK脈衝 注意： 當錯誤發生時，I2C匯流排繼續動作，軟體必須保證START和STOP狀況錯位情況不會發生，見17.4.4節「調停」 寫入1來清除它																										
9	SAD	Slave位址檢測： 0 = 沒有檢測到slave位址 1 = I2C單元檢測到一個7位元位址跟一般呼叫位址或ISAR相符合，當SADIE中斷設置為1時，中斷會發出信號 寫入1來清除它																										
8	GCAD	一般呼叫位址檢測： 0 = 沒有接收到一般呼叫位址 1 = I2C單元接收到一般呼叫位址																										
7	IRF	IDBR接收滿溢： 0 = IDBR沒有接收到一個新的資料位元組或是I2C單元閒置 1 = IDBR暫存器從I2C匯流排接收到一個新資料位元組，當IRFIE設置為1時中斷會發出信號 寫入1來清除它																										
6	ITE	IDBR傳送空的： 0 = 資料位元組仍舊在傳送 1 = I2C單元已經在I2C匯流排上完成傳送資料，當ITEIE中斷設置為1時，中斷會發出信號 寫入1來清除它																										
5	ALD	調停失敗檢測： 在multi-master操作期間使用 0 = 當調停獲得或從未發生時清除 1 = 當I2C單元調停失敗時設置 寫入1來清除它																										
4	SSD	Slave停止檢測： 0 = 沒有檢測到停止																										

		1 = 在slave接收或是slave傳送模式下當I2C單元檢測到停止時設置 寫入1來清除它
3	IBB	I2C匯流排忙碌： 0 = I2C匯流排是閒置的或I2C單元正在使用匯流排（即單元忙碌） 1 = 當I2C匯流排是忙碌但I2C單元並沒進行動作時設置
2	UB	單元忙碌： 0 = I2C單元不是忙碌 1 = 當I2C單元忙碌時設置，定義為第一個START和STOP之間的時間
1	ACKNAK	ACK/NAK狀態： 0 = I2C單元在匯流排上接收或傳送ACK 1 = I2C單元接收或傳送NAK 當傳輸的位元組是最後一個時，使用在slave傳送模式下來決定，在接收到每個位元組和ACK/NAK資訊後更新
0	RWM	讀取/寫入模式： 0 = I2C單元在master傳送或slave接收模式下 1 = I2C單元在master接收或slave傳送模式下 slave位址的R/nW位元，在停止狀態後由硬體自動清除

17.9.5 I2C Slave位址暫存器 — ISAR

ISAR（見表17-13）定義了I2C單元的7個位元的slave位址。在slave接收模式下，當7個位元的位址跟暫存器裡的值相符合的話，應用程式處理器回應。應用程式處理器寫入這個暫存器在它啟動I2C操作前，ISAR是可完全程式化的（沒有位址被分配到I2C單元）所以它能設置一個值，除了那些系統內固定接線的I2C slave 週邊裝置之外，假如應用程式處理器重置，ISAR不會受影響，ISAR暫存器預設值為0000000₂。

表17-13 I2C Slave位址暫存器 — ISAR

實體位址 4030_16A0		I ² C Slave 位址暫存器	I ² C
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	ISA
重置	0 0		
31:7	Reserved	保留	
6:0	ISA	I ² C Slave位址：當在slave接收模式下I2C單元回應7個位元的位址	

問題：

1. I2C有幾隻接腳？他們分別代表著什麼功能？
2. 裝置連接到I2C時，處理器如何分辨不同的裝置來源？
3. 裝置連接到I2C時，它可以當做哪兩種不同的身分？
4. 使用I2C來傳送資料時，是由誰發起起始訊號？
5. Slave的住址有多少個位元？
6. 當何種情況發生時，I2C調停會失敗？
7. 當多個Master要傳送資料時，其時脈線會如何處理？
8. I2C作序列的資料傳送是先送MSB(Most Significant Bit)還是先送LSB(Less Significant Bit)?
9. 請解釋圖17-11、圖17-12與圖17-13的動作原理？
10. Slave作讀入或寫出的動作是如何決定的？
11. 試述何謂IC（交互整合電路），以及其優點為何。