

## 第六章 系統整合單元

這一章將描述系統整合單元(System Integration Unit - SIU)。SIU 控制幾種處理器的系統功能，SIU 所包含的單元如下：

- 通用 I/O 埠(General-purpose I/O ports)
- 中斷控制器(Interrupt controller)
- 即時時脈(Real-time clock)
- 作業系統計時器(Operating system timer)
- 脈衝寬度調變器(Pulse Width modulator)

### 6.1 通用 I/O

PXA250 應用處理器透過 27 個暫存器來致能或控制通用 I/O(GPIO)的 81 個腳位，控制內容如腳位的方向(輸入或輸出)、腳位功能、腳位狀態(只能輸出)、腳位位準偵測(只能輸入)和交替功能。參考表 2-11「PXA210 應用處理器不支援的信號」即為 PXA210 不支援的信號列表。其中有一部分的 GPIO 腳位可以使應用處理器離開睡眠模式。注意當選擇某個 GPIO 的腳位就是指定某一個 GPIO 的功能，因為 GPIO 的許多腳位具有交替功能，而且可以被設定為可支援應用處理器的周邊設備。

將所有未使用的 GPIO 腳位設定為輸出，將可以把電源的消耗降到最低。

#### 6.1.1 GPIO 的運作

PXA250 應用處理器提供 81 個 GPIO 的腳位來產生及捕捉指定的輸入及輸出信號。每一個腳位都可以被設計成輸入或輸出。當設計成輸入時，GPIO 也可以被當作一中斷來源。當全部都是重置(reset)時，81 個腳位被設定為輸入，並且持續當作輸入直到他們被設定成其他狀態。

使用 GPIO 腳位之方向暫存器(GPIO Pin Direction register - GPDR)來設定 GPIO 腳位為輸出或是輸入。當設定為輸出時，寫入 GPIO 腳位之輸出設定暫存器(GPIO Pin Output Set register - GPSR)將腳位設定為高電位；寫入 GPIO 腳位之輸出清除暫存器(GPIO Pin Output Clear register - GPCR)則將腳位清除為低電位。不論腳位是被設定為輸入或輸出，設定和清除暫存器都可以被寫入。如果腳

位被設定為輸入時，先前所設定的輸出狀態會在腳位被重新設定為輸出時反應原設定的輸出。

讀取 GPIO 腳位之位準暫存器(GPIO Pin Level register - GPLR)可以確認每一個 GPIO 腳位的狀態。你可以隨時讀取這個暫存器來確認腳位目前的狀態。另外，使用 GPIO 正緣偵測致能暫存器(GPIO Rising Edge Detect Enable register - GRER)和 GPIO 負緣偵測致能暫存器(GPIO Falling Edge Detect Enable register - GFER)來偵測每一個腳位是正緣或是負緣。使用 GPIO 邊緣偵測狀態暫存器(GPIO Edge Detect Status register - GEDR)來讀取邊緣偵測狀態。你可以程式化這些邊緣偵測暫存器來產生中斷(參考 6.2 節「中斷控制器」)。並且使用 GPIO[15:0]產生喚醒事件讓 PXA250 應用處理器離開睡眠模式(3.4.8.5 節「離開睡眠模式」)。

當應用處理器進入睡眠模式，電源管理之睡眠狀態暫存器(PGSR0、PGSR1、PGSR2)的內容會載入到輸出資料暫存器(output data register)。如果某一個特定的腳位被設定為輸出，在進入睡眠模式前會依照 PGSR 內的值輸出到該腳位。當應用處理器離開睡眠模式，這些值會持續傳送到 GPIO 腳位，一直到 GPDR、GPSR、GPCR 被重新寫入新的值，並且設定電源管理之睡眠狀態暫存器(PSSR)中的 GPIO 位元，用以指示在離開睡眠模式之後，GPIO 暫存器已經被重新初始化。這是必須的步驟，因為 GPIO 的邏輯電路在睡眠模式中已失去電力。

在應用處理器中，大部分的 GPIO 腳位具備交替的功能。在序列控制器和 LCD 控制器的幾種模式中，有額外腳位的需求。這些功能是在特定的 GPIO 腳位中直接以硬體線路(hardwired)方式加入應用處理器中，他們的使用方式將在以下的段落依序說明。即使一個 GPIO 腳位被其他交替的功能使用，你仍然必須透過 GPDR 來程式化該腳位的方向。詳細的交替功能內容將在 6.1.2 節「GPIO 交替功能」中提出。圖 6-1 為單一 GPIO 腳位的區塊圖。

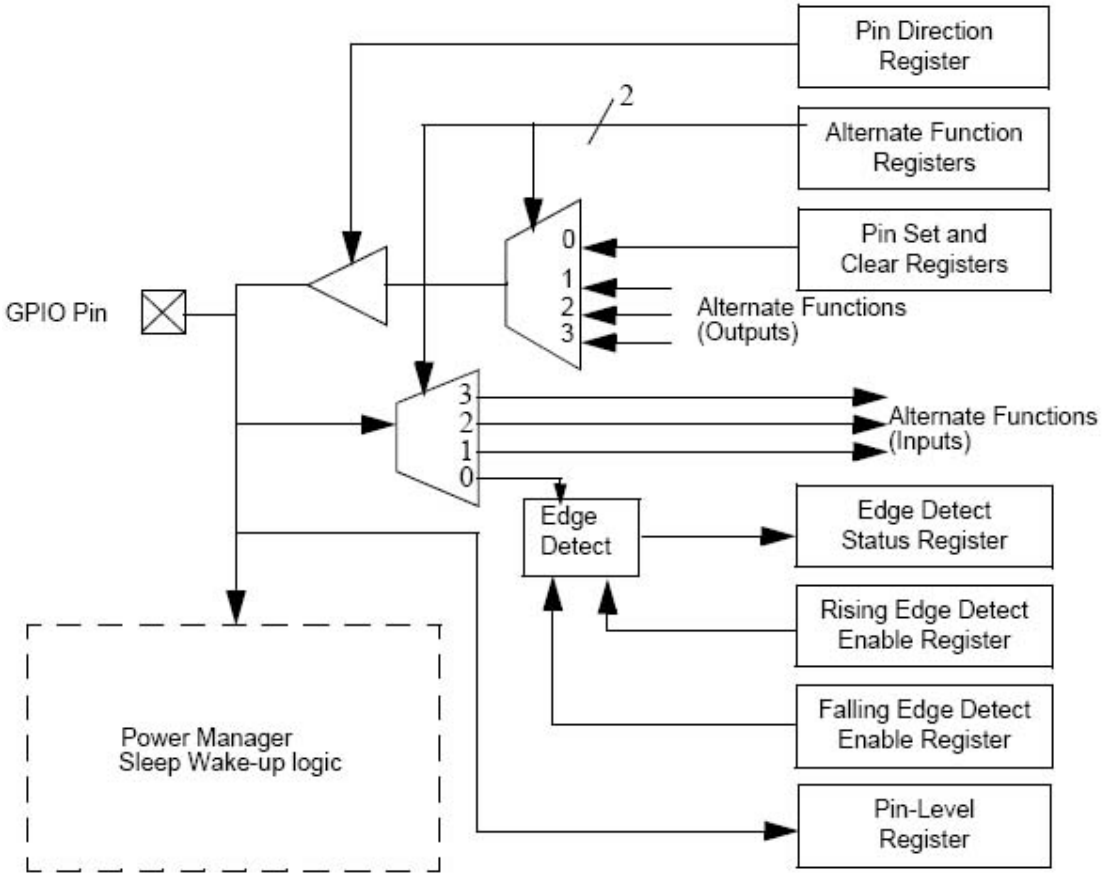


圖 6-1 通用 IO 區塊圖。

6.1.2 GPIO 的交替功能

GPIO的腳位可以容納高達六種的交替功能，可以透過應用處理器來驅動其功能。如果一個GPIO的腳位已經使用其交替功能，那它就不能在同一時間再被用來當作GPIO使用。GPIO[0]是保留的，因為他在睡眠模式中有特殊的用途，所以不能提供交替功能。GPIO[15:0]的功能是從睡眠模式中喚醒應用處理器。這個喚醒的功能在3.4.8.5節「離開睡眠模式」已經解釋過了。表6-1說明每個GPIO腳位以及其他對應的交替功能。

想要瞭解交替功能的更多資訊，請參考表6-1中對應的「來源單元」欄。

表 6-1 GPIO 交替功能

腳位 (Pin)	交替功能名稱 (Alternate Function Name)	交替功能指定 (Alternate Function)	AF(n)編碼 (AF(n) encoding)	來源單元 (Source Unit)	信號描述及註解 (Signal Description and comments)
-------------	--	-----------------------------------	--------------------------------	-----------------------	---

		Assignment)			
GP1	GP_RST	ALT_FN_1_IN	01	時脈及電源管理員單元	低電位動作 GP_reset
GP6	MMCCLK	ALT_FN_1_OUT	01	多媒體卡(MMC)控制器	MMC 時脈
GP7	48MHz clock <sup>+</sup>	ALT_FN_1_OUT	01	時脈及電源管理員單元	48MHz 時脈輸出
GP8	MMCCS0	ALT_FN_1_OUT	01	多媒體卡(MMC)控制器	MMC 晶片選擇 0
GP9	MMCCS1	ALT_FN_1_OUT	01		MMC 晶片選擇 1
GP10	RTCCLK	ALT_FN_1_OUT	01	系統整合單元	即時時脈(1 Hz)
GP11	3.6MHz	ALT_FN_1_OUT	01	時脈及電源管理員單元	3.6MHz 振盪器輸出
GP12	32kHz	ALT_FN_1_OUT	01		32kHz 輸出
GP13	MBGNT	ALT_FN_2_OUT	10	記憶體控制器	記憶體控制器允許
GP14	MBREQ	ALT_FN_1_IN	01		記憶體控制器交替匯流排主要求
GP15	nCS_1	ALT_FN_2_OUT	10	記憶體控制器	低電位動作晶片選擇 1
GP16	PWM0	ALT_FN_2_OUT	10	系統整合單元	PWM0 輸出
GP17	PWM1	ALT_FN_2_OUT	10		PWM1 輸出
GP18	RDY	ALT_FN_1_IN	01	記憶體控制器	外部匯流排預備
GP19	DREQ[1]	ALT_FN_1_IN	01	記憶體控制器	外部 DMA 要求
GP20	DREQ[0]	ALT_FN_1_IN	01		外部 DMA 要求
GP23	SCLK	ALT_FN_2_OUT	10	SSP 串列埠	SSP 時脈
GP24	SFRM	ALT_FN_2_OUT	10		SSP 訊框
GP25	TXD	ALT_FN_2_OUT	10		SSP 傳送
GP26	RXD	ALT_FN_1_IN	01		SSP 接收
GP27	EXTCLK	ALT_FN_1_IN	01		SSP 外部時脈
GP28	BITCLK	ALT_FN_1_IN	01	AC97 控制器單元	AC97 bit_clk
GP28	BITCLK	ALT_FN_2_IN	10	I2S 控制器	I2S bit_clk
	BITCLK	ALT_FN_1_OUT	01		I2S bit_clk
	BITCLK	ALT_FN_2_OUT	10	AC97 控制器單元	AC97 bit_clk
GP29	SDATA_IN0	ALT_FN_1_IN	01	AC97 控制器單元	AC97 Sdata_in0
	SDATA_IN	ALT_FN_2_IN	10	I2S 控制器	I2S Sdata_in
GP30	SDATA_OUT	ALT_FN_1_OUT	01	I2S 控制器	I2S Sdata_out
	SDATA_OUT	ALT_FN_2_OUT	10	AC97 控制器單元	AC97 Sdata_out
GP31	SYNC	ALT_FN_1_OUT	01	I2S 控制器	I2S sync
	SYNC	ALT_FN_2_OUT	10	AC97 控制器單元	AC97 sync
GP32	SDATA_IN1	ALT_FN_1_IN	01	AC97 控制器單元	AC97 Sdata_in1
	SYSCCLK	ALT_FN_1_OUT	01	I2S 控制器	I2S 系統時脈
GP33	nCS[5]	ALT_FN_2_OUT	10	記憶體控制器	低電位動作晶片選擇 5

GP34	FFRXD	ALT_FN_1_IN	01	UARTs	FFUART 接收
	MMCCS0	ALT_FN_2_OUT	10	多媒體卡(MMC)控制器	MMC 晶片選擇 0
GP35	CTS	ALT_FN_1_IN	01	UARTs	FFUART Clear to send
GP36	DCD	ALT_FN_1_IN	01		FFUART Data carrier detect
GP37	DSR	ALT_FN_1_IN	01		FFUART data set ready
GP38	RI	ALT_FN_1_IN	01		FFUART Ring Indicator
GP39	MMCCS1	ALT_FN_1_OUT	01	多媒體卡(MMC)控制器	MMC 晶片選擇 1
	FFTXD	ALT_FN_2_OUT	10	UARTs	FFUART 傳送資料
GP40	DTR	ALT_FN_2_OUT	10	UARTs	FFUART data terminal ready
GP41	RTS	ALT_FN_2_OUT	10		FFUART request to send
GP42	BTRXD	ALT_FN_1_IN	01	UARTs	BTUART 接收資料
GP43	BTTXD	ALT_FN_2_OUT	10	UARTs	BTUART 傳送資料
GP44	CTS	ALT_FN_1_IN	01	UARTs	BTUART clear to send
GP45	RTS	ALT_FN_2_OUT	10		BTUART request to send
GP46	ICP_RXD	ALT_FN_1_IN	01	紅外線通訊埠	ICP 接收資料
	RXD	ALT_FN_2_IN	10	UARTs	STD_UART 接收資料
GP47	TXD	ALT_FN_1_OUT	01	UARTs	STD_UART 傳送資料
	ICP_TXD	ALT_FN_2_OUT	10	紅外線通訊埠	ICP 傳送資料
GP48	nPOE	ALT_FN_2_OUT	10	記憶體控制器	Card Space 輸出致能
GP49	nPWE	ALT_FN_2_OUT	10		Card Space 寫入致能
GP50	nPIOR	ALT_FN_2_OUT	10		Card Space I/O 讀取
GP51	nPIOW	ALT_FN_2_OUT	10		Card Space I/O 寫入
GP52	nPCE[1]	ALT_FN_2_OUT	10		Card Space Card 致能
GP53	nPCE[2]	ALT_FN_2_OUT	10		Card Space Card 致能
GP53	MMCCLK	ALT_FN_1_OUT	01	多媒體卡(MMC)控制器	MMC 時脈
GP54	MMCCLK	ALT_FN_1_OUT	01	多媒體卡(MMC)控制器	MMC 時脈
GP54	nPSKTSEL	ALT_FN_2_OUT	10	記憶體控制器	Card Space 插槽選擇
GP55	nPREG	ALT_FN_2_OUT	10	記憶體控制器	Card Address 位元 26
GP56	nPWAIT	ALT_FN_1_IN	01		Card Space 等待信號
GP57	nIOIS16	ALT_FN_1_IN	01	記憶體控制器	I/O Card Space 匯流排寬度選擇
GP58	LDD[0]	ALT_FN_2_OUT	10	LCD 控制器	LCD data 腳位 0
GP59	LDD[1]	ALT_FN_2_OUT	10		LCD data 腳位 1
GP60	LDD[2]	ALT_FN_2_OUT	10		LCD data 腳位 2
GP61	LDD[3]	ALT_FN_2_OUT	10		LCD data 腳位 3

GP62	LDD[4]	ALT_FN_2_OUT	10		LCD data 腳位 4
GP63	LDD[5]	ALT_FN_2_OUT	10		LCD data 腳位 5
GP64	LDD[6]	ALT_FN_2_OUT	10		LCD data 腳位 6
GP65	LDD[7]	ALT_FN_2_OUT	10		LCD data 腳位 7
GP66	LDD[8]	ALT_FN_2_OUT	10	LCD 控制器	LCD data 腳位 8
	MBREQ	ALT_FN_1_IN	01	記憶體控制器	記憶體控制器交替匯流排主要求
GP67	LDD[9]	ALT_FN_2_OUT	10	LCD 控制器	LCD data 腳位 9
	MMCCS0	ALT_FN_1_OUT	01	多媒體卡(MMC)控制器	MMC 晶片選擇 0
GP68	MMCCS1	ALT_FN_1_OUT	01	多媒體卡(MMC)控制器	MMC 晶片選擇 1
	LDD[10]	ALT_FN_2_OUT	10	LCD 控制器	LCD data 腳位 10
GP69	MMCCCLK	ALT_FN_1_OUT	01	多媒體卡(MMC)控制器	MMC_CLK
	LDD[11]	ALT_FN_2_OUT	10	LCD 控制器	LCD data 腳位 11
GP70	RTCCLK	ALT_FN_1_OUT	01	系統整合單元	即時時脈(1 Hz)
	LDD[12]	ALT_FN_2_OUT	10	LCD 控制器	LCD data 腳位 12
GP71	3.6MHz	ALT_FN_1_OUT	01	時脈及電源管理員單元	3.6MHz 振盪器時脈
	LDD[13]	ALT_FN_2_OUT	10	LCD 控制器	LCD data 腳位 13
GP72	32kHz	ALT_FN_1_OUT	01	時脈及電源管理員單元	32kHz 時脈
	LDD[14]	ALT_FN_2_OUT	10	LCD 控制器	LCD data 腳位 14
GP73	LDD[15]	ALT_FN_2_OUT	10	LCD 控制器	LCD data 腳位 15
	MBGNT	ALT_FN_1_OUT	01	記憶體控制器	記憶體控制器允許
GP74	LCD_FCLK	ALT_FN_2_OUT	10	LCD 控制器	LCD 訊框時脈
GP75	LCD_LCLK	ALT_FN_2_OUT	10		LCD 線時脈
GP76	LCD_PCLK	ALT_FN_2_OUT	10		LCD 像素時脈
GP77	LCD_ACBIAS	ALT_FN_2_OUT	10		LCD AC 偏壓
GP78	nCS[2]	ALT_FN_2_OUT	10	記憶體控制器	低電位動作晶片選擇 2
GP79	nCS[3]	ALT_FN_2_OUT	10		低電位動作晶片選擇 3
GP80	nCS[4]	ALT_FN_2_OUT	10		低電位動作晶片選擇 4

注意：表格中深色部分表示在 PXA210 應用處理器中不支援。

<sup>+</sup> CKEN[11] – 為了使 48MHz 時脈在 GP7 上輸出，USB 時脈致能位元必須致能。

### 6.1.2.1 PXA210 應用處理器操作上的限制

在 PXA210 中，以下幾個 GPIO 腳位是不可使用的：

- GPIO[14:2]
- GPIO[22:19]
- GPIO[38:35]

● GPIO[41:40]

注意：因為 GPIO[14:2]在 PXA210 應用處理器中是不可使用的，所以只有 GPIO[15]和 GPIO[1:0]可以用來當作從睡眠模式中喚醒的功能。

### 6.1.3 GPIO 暫存器定義

在GPIO控制區塊中總共有27個32位元暫存器。這27個暫存器分為三組，每組各對應到九種不同功能的暫存器，供81個GPIO腳位使用。各個GPIO腳位所對應到的暫存器描述如下：

- 三個監視暫存器之腳位狀態(GPLR)
- 六個控制暫存器之輸出腳位狀態(GPSR、GPCR)
- 三個控制暫存器之腳位方向(GPDR)
- 六個控制暫存器之偵測正緣與/或偵測負緣(GRER & GFER)
- 三個指示暫存器之當腳位上所指定的邊緣類型，已經被偵測到(GEDR)
- 六個決定暫存器之腳位是用來當作一般的GPIO之用，或是被用來當作三選一的交替功能(GAFR\_L、GAFR\_U)

表6-2 GPIO暫存器定義

暫存器 類型	暫存器功 能	GPIO[15:0]	GPIO[31:16]	GPIO[47:32]	GPIO[63:48]	GPIO[79:64]	GPIO[80]
GPLR	監控腳位狀 態	GPLR0		GPLR1		GPLR2	
GPSR	控制輸出腳 位狀態	GPSR0		GPSR1		GPSR2	
GPCR		GPCR0		GPCR1		GPCR2	
GPDR	設定腳位方 向	GPDR0		GPDR1		GPDR2	
GRER	偵測正/負緣	GRER0		GRER1		GRER2	
GFER		GFER0		GFER1		GFER2	
GEDR	偵測邊緣類 型	GEDR0		GEDR1		GEDR2	
GAFR	設定交替功 能	GAFR0_L	GAFR0_U	GAFR1_L	GAFR1_U	GAFR2_L	GAFR2_U

注意：對交替功能暫存器而言，\_L表示較低的16個GPIO交替功能被該暫存器所設定，\_U表示較高的16個GPIO交替功能被該暫存器所設定。

注意：GPLR2[31:17]、GPSR2[31:17]、GPCR2[31:17]、GPDR2[31:17]、GRER2[31:17]、GFER2[31:17]、GEDR2[31:17]和GAFR2\_U[31:2]是

保留位元。這些位元寫入為0，並且忽略讀取。

注意：全部的 GPIO 暫存器在重置(reset)時被初始設定為 0x0，並導致全部 GPIO 腳位被初始為輸入。

6.1.3.1 GPIO 腳位位準暫存器(GPLR)

藉由讀取 GPIO 腳位之位準暫存器(GPIO Pin-Level Register - GPLR)來得知每一個 GPIO 腳位的狀態。GPLR 中的每一個位元皆對應到 GPIO 的每一個腳位。GPLR0[31:0]對應到 GPIO[31:0]，GPLR1[31:0]對應到 GPIO[63:32]，GPLR2[16:0]對應到 GPIO[80:64]。使用唯讀的 GPLR0-2 暫存器來確定特定的腳位現在的數值(不論腳位方向已設定為何)。對於讀取保留的位元(GPLR2[31:17])，傳回零值。

表 6-3 到表 6-5 表示 GPIO 腳位之位準暫存器的位元地圖。

表 6-3 GPLR0 暫存器位元地圖

Physical Address 0x40E0_0000										GPIO Pin Level Register0 (GPLR0)																		SysInt				
Bit	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
	PL31	PL30	PL29	PL28	PL27	PL26	PL25	PL24	PL23	PL22	PL21	PL20	PL19	PL18	PL17	PL16	PL15	PL14	PL13	PL12	PL11	PL10	PL9	PL8	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位元	名稱	描述
<31:0>	PL[x]	GPIO 腳位位準 ‘x’ (x = 0 to 31) 這是唯讀的欄位，用來指示每一個 GPIO 當時的數值。 0=腳位狀態為低電位 1=腳位狀態為高電位

表 6-4 GPLR1 暫存器位元地圖

Physical Address 0x40E0_0004								GPIO Pin Level Register1 (GPLR1)																								SysInt	
Bit	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																																
	PL63	PL62	PL61	PL60	PL59	PL58	PL57	PL56	PL55	PL54	PL53	PL52	PL51	PL50	PL49	PL48	PL47	PL46	PL45	PL44	PL43	PL42	PL41	PL40	PL39	PL38	PL37	PL36	PL35	PL34	PL33	PL32	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位元	名稱	描述
<31:0>	PL[x]	GPIO 腳位位準 ‘x’ (x = 32 to 63) 這是唯讀的欄位，用來指示每一個 GPIO 當時的數值。 0=腳位狀態為低電位 1=腳位狀態為高電位



表 6-5 GPLR2 暫存器位元地圖

Physical Address 0x40E0_0008																GPIO Pin Level Register2 (GPLR2)														SysInt															
Bit																																													
																Reserved																													
Reset																0														0															

位元	名稱	描述
<31:17>	—	保留
<16:0>	PL[x]	GPIO 腳位位準 ‘x’ (x = 64 to 80) 這是唯讀的欄位，用來指示每一個 GPIO 當時的數值。 0=腳位狀態為低電位 1=腳位狀態為高電位

6.1.3.2 GPIO 腳位之方向暫存器(GPDR)

每個腳位的輸入或輸出是透過設定 GPIO 腳位之方向暫存器(GPDR0、GPDR1、GPDR2)來控制的。GPDR 暫存器對 GPIO 的 81 個腳位都有一個相對應的方向控制位元。如果方向位元設定為 1，GPIO 為輸出。如果設定為 0，為輸入。保留位元(GPDR2[31:17])必須被寫入 0，並且讀取到保留位元時必須被忽略。

注意：重置(reset)會將GPDR0-2暫存器的位元清除為0，並且設定全部的GPIO腳位為輸入。

表 6-6 到表 6-8 說明 GPIO 腳位之方向暫存器的位元地圖。

表 6-6 GPDR0 暫存器位元地圖

Physical Address 0x40E0_000C																GPIO Pin Direction Register0 (GPDR0)														SysInt															
Bit																																													
Reset																0														0															

位元	名稱	描述
<31:0>	PD[x]	GPIO 第 ‘x’ 個的腳位方向(x = 0 to 31) 0=腳位為輸入 1=腳位為輸出

表 6-7 GPDR1 暫存器位元地圖

		Physical Address 0x40E0_0010								GPIO Pin Direction Register1 (GPDR1)																SysInt							
Bit		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
		PD63	PD62	PD61	PD60	PD59	PD58	PD57	PD56	PD55	PD54	PD53	PD52	PD51	PD50	PD49	PD48	PD47	PD46	PD45	PD44	PD43	PD42	PD41	PD40	PD39	PD38	PD37	PD36	PD35	PD34	PD33	PD32
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位元	名稱	描述
<31:0>	PD[x]	GPIO 第‘x’個的腳位方向(x = 32 to 63) 0=腳位為輸入 1=腳位為輸出

表 6-8 GPDR2 暫存器位元地圖

		Physical Address 0x40E0_0014																GPIO Pin Direction Register2 (GPDR2)																SysInt															
Bit		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																PD80	PD79	PD78	PD77	PD76	PD75	PD74	PD73	PD72	PD71	PD70	PD69	PD68	PD67	PD66	PD65	PD64															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																

位元	名稱	描述
<31:17>	—	保留
<16:0>	PD[x]	GPIO 第‘x’個的腳位方向(x = 64 to 80) 0=腳位為輸入 1=腳位為輸出

### 6.1.3.3 GPIO 腳位之輸出設定暫存器(GPSR)及腳位輸出清除暫存器(GPCR)

當一個GPIO被設定為輸出，你可以藉由寫入GPIO腳位之輸出設定暫存器(GPIO Pin Output Set register - GPSR)或GPIO腳位之輸出清除暫存器(GPIO Pin Output Clear register - GPCR)來控制這個腳位的狀態。一個輸出的腳位被設定為高電位，是藉由寫入1到它GPSR中對應的位元。要清除一個輸出的腳位，寫入1到GPCR中對應的位元。GPSR和GPCR都是唯寫的暫存器。嘗試讀取會傳回一個無法預料的數值。

寫入一個0到GPSR或GPCR中的任一位元對於腳位的狀態是不會有影響的。當腳位被設定為輸入時，寫入一個1到GPSR或GPCR中的位元，其對應到的腳位只有在該腳位被設定為輸出之後才會有效。保留位元(GPSR2[31:17]及GPCR2[31:17])必須被寫入0，並且讀取時必須被忽略。

表6-9到表6-11說明GPIO腳位之輸出設定暫存器的位元地圖。表6-12到表6-14說明GPIO腳位之輸出清除暫存器的位元地圖。

表 6-9 GPSR0 暫存器位元地圖

[illegible]

位元	名稱	描述
<31:0>	PS[x]	GPIO 第‘x’個輸出腳位設定(x = 0 to 31) 0=腳位狀態不受影響 1=若腳位設定為輸出，腳位的位準將被設定為高電位(1)

**表 6-10 GPSR1 暫存器位元地圖**

[illegible]

位元	名稱	描述
<31:0>	PS[x]	GPIO 第‘x’個輸出腳位設定(x = 32 to 63) 0=腳位狀態不受影響 1=若腳位設定為輸出，腳位的位準將被設定為高電位(1)

**表 6-11 GPSR2 暫存器位元地圖**

[illegible]

位元	名稱	描述
<31:17>	—	保留
<16:0>	PS[x]	GPIO 第‘x’個輸出腳位設定(x = 64 to 80) 0=腳位狀態不受影響 1=若腳位設定為輸出，腳位的位準將被設定為高電位(1)

表 6-12 GPCR0 暫存器位元地圖

Physical Address 0x40E0_0024										GPIO Pin Output Clear Register0 (GPCR0)														SysInt									
Bit	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	
	PC31	PC30	PC29	PC28	PC27	PC26	PC25	PC24	PC23	PC22	PC21	PC20	PC19	PC18	PC17	PC16	PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位元	名稱	描述
<31:0>	PC[x]	GPIO 第‘x’個輸出腳位清除(x = 0 to 31) 0=腳位狀態不受影響 1=若腳位設定為輸出，腳位的位準將被清除為低電位(0)

表 6-13 GPCR1 暫存器位元地圖

Physical Address 0x40E0_0028								GPIO Pin Output Clear Register1 (GPCR1)																SysInt								
Bit	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
	PC63	PC62	PC61	PC60	PC59	PC58	PC57	PC56	PC55	PC54	PC53	PC52	PC51	PC50	PC49	PC48	PC47	PC46	PC45	PC44	PC43	PC42	PC41	PC40	PC39	PC38	PC37	PC36	PC35	PC34	PC33	PC32
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位元	名稱	描述
<31:0>	PC[x]	GPIO 第‘x’個輸出腳位清除(x = 32 to 63) 0=腳位狀態不受影響 1=若腳位設定為輸出，腳位的位準將被清除為低電位(0)

表 6-14 GPCR2 暫存器位元地圖

Physical Address 0x40E0_002C																GPIO Pin Output Clear Register2 (GPCR2)																SysInt															
Bit	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																PC80	PC79	PC78	PC77	PC76	PC75	PC74	PC73	PC72	PC71	PC70	PC69	PC68	PC67	PC66	PC65	PC64														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0															

位元	名稱	描述
<31:17>	—	保留
<16:0>	PC[x]	GPIO 第‘x’個輸出腳位清除(x = 64 to 80) 0=腳位狀態不受影響 1=若腳位設定為輸出，腳位的位準將被清除為低電位(0)

#### 6.1.3.4 GPIO 正緣偵測致能暫存器(GRER)與負緣偵測致能暫存器(GFER)

每一個 GPIO 腳位都可以程式化為用來偵測腳位上的正緣、負緣或是正負緣的轉變狀態。當偵測到的邊緣型態與該腳位程式化的邊緣型態符合時，就會設定 (set)相對應的狀態位元(status bit)。中斷控制器也可以設定為當任何一個狀態位元被設定時，就會產生中斷信號通知核心。除此之外，透過對中斷控制器的程式化，可以把某一個子集合的狀態位元設定起來，觸發應用處理器，讓應用處理器從睡眠模式中被喚醒。參考 3.4.8 節「睡眠模式」及 3.5.6 節「電源管理員之 GPIO 邊緣偵測狀態暫存器」，可以得到更多關於狀態位元造成應用處理器從睡眠模式中被喚醒的資訊。

造成 GPIO 邊緣偵測致能暫存器(GEDR)的位元被設定，其邊緣的類型是透過正緣偵測致能暫存器(GRER)和負緣偵測致能暫存器(GFER)來選擇得到的。舉一個例子，某一個 GPIO 腳位該腳位的狀態由邏輯位準零(logic level zero)轉變成邏輯位準一(logic level one)時，使其所對應的 GRER 位元被設定時，將導致相對應的 GEDR 狀態位元被設定。同樣地，由邏輯位準一轉變成邏輯位準零時，相對應的 GFER 狀態位元與 GEDR 狀態位元會被設定。當對應位元的在兩個暫存器(GRER 與 GFER)都被設定時，正緣或負緣轉變都會導致對應的 GEDR 狀態位元被設定。

注意：為確保邊緣偵測，最小的脈衝寬度時間需為1μS。

表 6-15 到表 6-17 說明 GPIO 正緣偵測致能暫存器的位元地圖。

表 6-18 到表 6-20 說明 GPIO 負緣偵測致能暫存器的位元地圖。

注意：GRER2 和 GFER2 中的保留位元，寫入必須為 0 且讀取必須被忽略。

表 6-15 GRER0 暫存器位元地圖

Physical Address 0x40E0_0030										GPIO Rising Edge Detect Enable Register0 (GRER0)														SysInt								
Bit	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
	RE31	RE30	RE29	RE28	RE27	RE26	RE25	RE24	RE23	RE22	RE21	RE20	RE19	RE18	RE17	RE16	RE15	RE14	RE13	RE12	RE11	RE10	RE9	RE8	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位元	名稱	描述
<31:0>	RE[x]	GPIO 第‘x’個腳位正緣偵測致能(x = 0 to 31) 0=關閉正緣偵測致能 1=當 GPIO 腳位偵測到一個正緣時，其對應的 GEDR 狀態位元將被設定

表 6-16 GRER1 暫存器位元地圖

Physical Address 0x40E0_0034								GPIO Rising Edge Detect Enable Register1 (GRER1)																SysInt								
Bit	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
	RE63	RE62	RE61	RE60	RE59	RE58	RE57	RE56	RE55	RE54	RE53	RE52	RE51	RE50	RE49	RE48	RE47	RE46	RE45	RE44	RE43	RE42	RE41	RE40	RE39	RE38	RE37	RE36	RE35	RE34	RE33	RE32
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位元	名稱	描述
<31:0>	RE[x]	GPIO 第‘x’個腳位正緣偵測致能(x = 32 to 63) 0=關閉正緣偵測致能 1=當 GPIO 腳位偵測到一個正緣時，其對應的 GEDR 狀態位元將被設定

表 6-17 GRER2 暫存器位元地圖

Physical Address 0x40E0_0038										GPIO Rising Edge Detect Enable Register2 (GRER2)										SysInt													
Bit	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																RE80	RE79	RE78	RE77	RE76	RE75	RE74	RE73	RE72	RE71	RE70	RE69	RE68	RE67	RE66	RE65	RE64
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位元	名稱	描述
<31:17>	—	保留
<16:0>	RE[x]	GPIO 第‘x’個腳位正緣偵測致能(x = 64 to 80) 0=關閉正緣偵測致能 1=當 GPIO 腳位偵測到一個正緣時，其對應的 GEDR 狀態位元將被設定

表 6-18 GFER0 暫存器位元地圖

Physical Address 0x40E0_003C								GPIO Falling Edge Detect Enable Register0 (GFER0)																SysInt								
Bit	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
	FE31	FE30	FE29	FE28	FE27	FE26	FE25	FE24	FE23	FE22	FE21	FE20	FE19	FE18	FE17	FE16	FE15	FE14	FE13	FE12	FE11	FE10	FE9	FE8	FE7	FE6	FE5	FE4	FE3	FE2	FE1	FE0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位元	名稱	描述
<31:0>	FE[x]	GPIO 第‘x’個腳位負緣偵測致能(x = 1 to 31) 0=關閉負緣偵測致能 1=當 GPIO 腳位偵測到一個負緣時，其對應的 GEDR 狀態位元將被設定

表 6-19 GFER1 暫存器位元地圖

Physical Address 0x40E0_0040										GPIO Falling Edge Detect Enable Register1 (GFER1)										SysInt												
Bit	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
	FE63	FE62	FE61	FE60	FE59	FE58	FE57	FE56	FE55	FE54	FE53	FE52	FE51	FE50	FE49	FE48	FE47	FE46	FE45	FE44	FE43	FE42	FE41	FE40	FE39	FE38	FE37	FE36	FE35	FE34	FE33	FE32
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位元	名稱	描述
<31:0>	FE[x]	GPIO 第‘x’個腳位負緣偵測致能(x = 32 to 63) 0=關閉負緣偵測致能 1=當 GPIO 腳位偵測到一個負緣時，其對應的 GEDR 狀態位元將被設定

表 6-20 GFER2 暫存器位元地圖

Physical Address 0x40E0_0044		GPIO Falling Edge Detect Enable Register2 (GFER2)																SysInt															
Bit	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																FE80	FE79	FE78	FE77	FE76	FE75	FE74	FE73	FE72	FE71	FE70	FE69	FE68	FE67	FE66	FE65	FE64
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位元	名稱	描述
<31:17>	—	保留
<16:0>	FE[x]	GPIO 第‘x’個腳位負緣偵測致能(x = 64 to 80) 0=關閉負緣偵測致能 1=當 GPIO 腳位偵測到一個負緣時，其對應的 GEDR 狀態位元將被設定

### 6.1.3.5 GPIO 邊緣偵測狀態暫存器(GEDR)

GPIO 邊緣偵測狀態暫存器(GPIO Edge Detect Status register - GEDR0、GEDR1、GEDR2)包含總共 81 個狀態位元，並且對應到 81 個 GPIO 腳位。當在腳位上的邊緣偵測發生時，並且符合 GRER 與/或 GFER 暫存器中設定的邊緣類型，則在 GEDR 中所對應的狀態位元將被設定。一旦 GEDR 位元被一個邊緣事件設定後，這個位元會一直保持其設定，直到使用者在狀態位元中寫入 1 來清除它。在 GEDR 狀態位元中寫入 0 是不會有任何效用的。

GPIO[80:0]設定所對應的每個邊緣偵測，其所對應的 GEDR 狀態位元都可以觸發一個中斷請求。GPIO[80:2]可以形成一個單一群組，當 GEDR[80:2]之中任何一個被設定時，都可以導致一個中斷請求被觸發。GPIO[0]和 GPIO[1]則產生獨立的第一等級(first-level)中斷。參考 6.2 節「中斷控制器」，將有關於 GPIO 中

斷程式化的描述。

表 6-21 到表 6-23 表示 GPIO 邊緣偵測狀態暫存器的位元地圖。

**表 6-21 GEDR0 暫存器位元地圖**

Physical Address 0x40E0_0048		GPIO Edge Detect Status Register0 (GEDR0)																SysInt														
Bit	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
	ED31	ED30	ED29	ED28	ED27	ED26	ED25	ED24	ED23	ED22	ED21	ED20	ED19	ED18	ED17	ED16	ED15	ED14	ED13	ED12	ED11	ED10	ED9	ED8	ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位元	名稱	描述
<31:0>	ED[x]	GPIO 第‘x’個腳位邊緣偵測狀態(x = 0 to 31) 讀取 0—GRER 與/或 GFER 所對應的腳位沒有邊緣被偵測 1—GRER 與/或 GFER 所對應的腳位有邊緣被偵測 寫入 0—沒有效用 1—清除邊緣偵測狀態欄位

**表 6-22 GEDR1 暫存器位元地圖**

Physical Address 0x40E0_004C								GPIO Edge Detect Status Register1 (GEDR1)																SysInt								
Bit	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
	ED63	ED62	ED61	ED60	ED59	ED58	ED57	ED56	ED55	ED54	ED53	ED52	ED51	ED50	ED49	ED48	ED47	ED46	ED45	ED44	ED43	ED42	ED41	ED40	ED39	ED38	ED37	ED36	ED35	ED34	ED33	ED32
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位元	名稱	描述
<31:0>	ED[x]	GPIO 第‘x’個腳位邊緣偵測狀態(x = 32 to 63) 讀取 0—GRER 與/或 GFER 所對應的腳位沒有邊緣被偵測 1—GRER 與/或 GFER 所對應的腳位有邊緣被偵測 寫入 0—沒有效用 1—清除邊緣偵測狀態欄位



表 6-23 GEDR2 暫存器位元地圖

Physical Address 0x40E0_0050										GPIO Edge Detect Status Register2 (GEDR2)										SysInt												
Bit	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															ED80	ED79	ED78	ED77	ED76	ED75	ED74	ED73	ED72	ED71	ED70	ED69	ED68	ED67	ED66	ED65	ED64
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位元	名稱	描述
<31:17>	—	保留
<16:0>	ED[x]	GPIO 第‘x’個腳位邊緣偵測狀態(x = 64 to 80) 讀取 0—GRER 和/或 GFER 所對應的腳位沒有邊緣被偵測 1—GRER 和/或 GFER 所對應的腳位有邊緣被偵測 寫入 0—沒有效用 1—清除邊緣偵測狀態欄位

6.1.3.6 GPIO 交替功能暫存器(GAFR)

GPIO 交替功能暫存器(GPIO alternate function register - GAFR2、GAFR1、GAFR0)內含相對應的 81 個 GPIO 腳位的選擇位元。每個 GPIO 可以設定為標準的 GPIO 腳位，三選一的交替輸入功能，或是三選一的交替輸出功能。要選擇任何一種交替的功能，GPDR 暫存器必須設定 GPIO 為輸入，才可以使用交替的輸入功能。同樣的，GPDR 暫存器必須設定 GPIO 為輸出，才可以使用交替的輸出功能。每個 GPIO 腳位分配到的一對位元，該位元值決定此 GPIO 腳位的功能(標準 GPIO、交替功能 1、交替功能 2 或交替功能 3)。GAFR 成對的位元所對應的選擇功能如下：

- 「00」表示標準 GPIO 功能
- 「01」選擇交替輸入功能 1(ALT\_FN\_1\_IN)或交替輸出功能 1(ALT\_FN\_1\_OUT)
- 「10」選擇交替輸入功能 2(ALT\_FN\_2\_IN)或交替輸出功能 2(ALT\_FN\_2\_OUT)
- 「11」選擇交替輸入功能 3(ALT\_FN\_3\_IN)或交替輸出功能 3(ALT\_FN\_3\_OUT)

表 6-24 到表 6-29 說明 GPIO 交替功能暫存器的位元地圖。

警告：設定 GPIO 對應到一個不存在的交替功能時，將會產生不確定的結果。

表 6-24 GAFR0\_L 暫存器位元地圖

		Physical Address 0x40E0_0054																GPIO Alternate Function Register0 Lower(GAFR0_L)																SysInt															
Bit		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																
		AF15	AF14	AF13	AF12	AF11	AF10	AF9	AF8	AF7	AF6	AF5	AF4	AF3	AF2	AF1	AF0																																
Reset		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	0															

位元	名稱	描述
<31:0>	AF[x]	<p>GPIO 第‘x’個腳位之交替功能選擇位元(x = 0 to 15)</p> <p>在暫存器中的一對位元決定GPIO腳位為對應的交替功能之一或是標準的GPIO腳位</p> <p>00—GPIO腳位(GPIO[x])為通用I/O</p> <p>01—GPIO腳位(GPIO[x])為交替功能1</p> <p>10—GPIO腳位(GPIO[x])為交替功能2</p> <p>11—GPIO腳位(GPIO[x])為交替功能3</p>

表 6-25 GAFR0\_U 暫存器位元地圖

		Physical Address 0x40E0_0058																GPIO Alternate Function Register0 Upper(GAFR0_U)																SysInt															
Bit		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																
		AF31	AF30	AF29	AF28	AF27	AF26	AF25	AF24	AF23	AF22	AF21	AF20	AF19	AF18	AF17	AF16																																
Reset		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	0															

位元	名稱	描述
<31:0>	AF[x]	<p>GPIO 第‘x’個腳位之交替功能選擇位元(x = 16 to 31)</p> <p>在暫存器中的一對位元決定GPIO腳位為對應的交替功能之一或是標準的GPIO腳位</p> <p>00—GPIO腳位(GPIO[x])為通用I/O</p> <p>01—GPIO腳位(GPIO[x])為交替功能1</p> <p>10—GPIO腳位(GPIO[x])為交替功能2</p> <p>11—GPIO腳位(GPIO[x])為交替功能3</p>

表 6-26 GAFR1\_L 暫存器位元地圖

		Physical Address 0x40E0_005C								GPIO Alternate Function Register1 Lower(GAFR1_L)																SysInt								
Bit		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	
		AF47	AF46	AF45	AF44	AF43	AF42	AF41	AF40	AF39	AF38	AF37	AF36	AF35	AF34	AF33	AF32																	
Reset		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	0

位元	名稱	描述
<31:0>	AF[x]	GPIO 第‘x’個腳位之交替功能選擇位元(x = 32 to 47)

		<p>在暫存器中的一對位元決定GPIO腳位為對應的交替功能之一或是標準的GPIO腳位</p> <p>00—GPIO腳位(GPIO[x])為通用I/O</p> <p>01—GPIO腳位(GPIO[x])為交替功能1</p> <p>10—GPIO腳位(GPIO[x])為交替功能2</p> <p>11—GPIO腳位(GPIO[x])為交替功能3</p>
--	--	--

表 6-27 GAFR1\_U 暫存器位元地圖

	Physical Address 0x40E0_0060								GPIO Alternate Function Register1 Upper(GAFR1_U)								SysInt															
Bit	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
	AF63	AF62	AF61	AF60	AF59	AF58	AF57	AF56	AF55	FA54	AF53	AF52	AF51	AF50	AF49	AF48																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位元	名稱	描述
<31:0>	AF[x]	<p>GPIO 第‘x’個腳位之交替功能選擇位元(x = 48 to 63)</p> <p>在暫存器中的一對位元決定GPIO腳位為對應的交替功能之一或是標準的GPIO腳位</p> <p>00—GPIO腳位(GPIO[x])為通用I/O</p> <p>01—GPIO腳位(GPIO[x])為交替功能1</p> <p>10—GPIO腳位(GPIO[x])為交替功能2</p> <p>11—GPIO腳位(GPIO[x])為交替功能3</p>

表 6-28 GAFR2\_L 暫存器位元地圖

	Physical Address 0x40E0_0064								GPIO Alternate Function Register2 Lower(GAFR2_L)								SysInt															
Bit	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
	AF79	AF78	AF77	AF76	AF75	AF74	AF73	AF72	AF71	AF70	AF69	AF68	AF67	AF66	AF65	AF64																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位元	名稱	描述
<31:0>	AF[x]	<p>GPIO 第‘x’個腳位之交替功能選擇位元(x = 64 to 79)</p> <p>在暫存器中的一對位元決定GPIO腳位為對應的交替功能之一或是標準的GPIO腳位</p> <p>00—GPIO腳位(GPIO[x])為通用I/O</p> <p>01—GPIO腳位(GPIO[x])為交替功能1</p> <p>10—GPIO腳位(GPIO[x])為交替功能2</p> <p>11—GPIO腳位(GPIO[x])為交替功能3</p>

表 6-29 GAFR2\_U 暫存器位元地圖

		Physical Address 0x40E0_0068								GPIO Alternate Function Register2 Upper(GAFR2_U)																SysInt							
Bit		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																														AF80	
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位元	名稱	描述
<31:2>	—	保留
<1:0>	AF[80]	GPIO 第 80 腳位之交替功能選擇位元 在暫存器中的一對位元決定GPIO腳位為對應的交替功能之一或是標準的GPIO腳位 00—GPIO腳位(GPIO[80])為通用I/O 01—GPIO腳位(GPIO[80])為交替功能1 10—GPIO腳位(GPIO[80])為交替功能2 11—GPIO腳位(GPIO[80])為交替功能3

#### 6.1.3.7 設定交替功能暫存器的範例程序

在這個範例中，GP0 是當作一般的 GPIO，GP(15:1)被設定為交替的功能。參考表 6-1 列出的交替功能，沒有其他的 GPIO 被設定。在這個範例中，任何的重置被移除之後，GPDOR(15:0) 將 GPIO 腳位設定為輸入。GAFR00(31:0)會為 0x0000\_0000，將 GPIO 指定為一般 GPIO 功能。為了簡單起見，假設 GP(16-31)表示設定為一般 GPIO 的輸入。

在此範例中，

- GPIO[0]被設定為標準 GPIO 輸入
- GPIO[1]為輸入，設定為交替功能 1(ALT\_FN\_1\_IN)
- GPIO[5:2]保留，設定為標準 GPIO 輸入
- GPIO[12:6]為輸出，設定為交替功能 1(ALT\_FN\_1\_OUT)
- GPIO[13]為輸出，設定為交替功能 2(ALT\_FN\_2\_OUT)
- GPIO[14]為輸入，設定為交替功能 1(ALT\_FN\_1\_IN)
- GPIO[15]為輸出，設定為交替功能 2(ALT\_FN\_2\_OUT)

為了在離開重置之後可以設定 GPIO 的交替功能，以下的設定程序是必要的：

1. 寫入 GPSR0 0x0000\_8000—將 GPIO[15]被設定為輸出。
2. 寫入 GPDOR 0x0000\_BFC0—GPIO[12:6]、GPIO[13]和 GPIO[15]做為輸出。將 GPIO[15]驅動為高電位，直到交替功能被設定。這是低電位輸出之前所需要

做的動作。

3. 寫入 GAFR0\_L 0x9955\_5004—將 GPIO[15:0]對應到交替功能。

為了將 GPIO 設定為輸出，你必須先設定 GPSR 和 GPCR，使得腳位的方向被改變。藉由設定 GPDR 暫存器的位元來改變腳位的方向。

注意：更多的交替功能資訊，參考這份文件的表 6-1 之來源單元欄位。

## 6.1.4 GPIO 暫存器位置

表 6-30 說明相關的 GPIO 區塊的暫存器及它們的實體位址。

**表 6-30 GPIO 暫存器位址**

位址	名稱	描述
0x40E0_0000	GPLR0	GPIO 腳位之位準暫存器 GPIO[31:0]
0x40E0_0004	GPLR1	GPIO 腳位之位準暫存器 GPIO[63:32]
0x40E0_0008	GPLR2	GPIO 腳位之位準暫存器 GPIO[80:64]
0x40E0_000C	GPDR0	GPIO 腳位之方向暫存器 GPIO[31:0]
0x40E0_0010	GPDR1	GPIO 腳位之方向暫存器 GPIO[63:32]
0x40E0_0014	GPDR2	GPIO 腳位之方向暫存器 GPIO[80:64]
0x40E0_0018	GPSR0	GPIO 腳位之輸出設定暫存器 GPIO[31:0]
0x40E0_001C	GPSR1	GPIO 腳位之輸出設定暫存器 GPIO[63:32]
0x40E0_0020	GPSR2	GPIO 腳位之輸出設定暫存器 GPIO[80:64]
0x40E0_0024	GPCR0	GPIO 腳位之輸出清除暫存器 GPIO[31:0]
0x40E0_0028	GPCR1	GPIO 腳位之輸出清除暫存器 GPIO[63:32]
0x40E0_002C	GPCR2	GPIO 腳位之輸出清除暫存器 GPIO[80:64]
0x40E0_0030	GRER0	GPIO 正緣偵測致能暫存器 GPIO[31:0]
0x40E0_0034	GRER1	GPIO 正緣偵測致能暫存器 GPIO[63:32]
0x40E0_0038	GRER2	GPIO 正緣偵測致能暫存器 GPIO[80:64]
0x40E0_003C	GFER0	GPIO 負緣偵測致能暫存器 GPIO[31:0]
0x40E0_0040	GFER1	GPIO 負緣偵測致能暫存器 GPIO[63:32]
0x40E0_0044	GFER2	GPIO 負緣偵測致能暫存器 GPIO[80:64]
0x40E0_0048	GEDR0	GPIO 邊緣偵測狀態暫存器 GPIO[31:0]
0x40E0_004C	GEDR1	GPIO 邊緣偵測狀態暫存器 GPIO[63:32]
0x40E0_0050	GEDR2	GPIO 邊緣偵測狀態暫存器 GPIO[80:64]
0x40E0_0054	GAFR0_L	GPIO 交替功能選擇暫存器 GPIO[15:0]
0x40E0_0058	GAFR0_U	GPIO 交替功能選擇暫存器 GPIO[31:16]

0x40E0_005C	GAFR1_L	GPIO 交替功能選擇暫存器 GPIO[47:32]
0x40E0_0060	GAFR1_U	GPIO 交替功能選擇暫存器 GPIO[63:48]
0x40E0_0064	GAFR2_L	GPIO 交替功能選擇暫存器 GPIO[79:64]
0x40E0_0068	GAFR2_U	GPIO 交替功能選擇暫存器 GPIO[80]

## 6.2 中斷控制器

中斷控制器控制著應用處理器所有可以取得的中斷來源，並且包含所有第一階層的中斷來源的位置。中斷控制器也決定哪一個中斷會引發哪一個 IRQ 或是哪一個 FIQ 會發生，並且也可以遮蔽(mask)此中斷。中斷控制器只支援單一優先權等級(single priority level)，雖然如此，中斷還是可以被安排為 IRQ 或 FIQ，FIQ 的優先權比 IRQ 還高。

### 6.2.1 中斷控制器之操作

中斷控制器提供所有中斷來源的遮蔽功能，並且產生 FIQ 或 IRQ 的處理器中斷。應用處理器的中斷架構為兩級。

第一級為中斷控制器之遮蔽暫存器(Interrupt Controller Mask register - ICMR)中，所有已致能以及未遮蔽的中斷來源。第一級中斷被以下的暫存器控制：

- 中斷控制器之等候暫存器(Interrupt Controller Pending register - ICPR) – 辨別系統中所有的作用中的中斷。
- 中斷控制器之 IRQ 等候暫存器(Interrupt Controller IRQ Pending register - ICIP) – 內含所有可以產生 IRQ 中斷來源的中斷。中斷控制器之層級暫存器(Interrupt Controller Level register - ICLR)被設定來傳送中斷給 ICIP，並產生一個 IRQ。
- 中斷控制器之 FIQ 等候暫存器(Interrupt Controller FIQ Pending register - ICFP) – 內含所有可以產生 FIQ 中斷來源的中斷。中斷控制器之層級暫存器(Interrupt Controller Level register - ICLR)被設定來傳送中斷給 ICFP，並產生一個 FIQ。

第二級使用在來源裝置內的暫存器(這個裝置已產生相對應的第一級中斷位元)。第二級中斷狀態提供額外的中斷的資訊，並且在中斷服務常式內被使用。一般而言，很多個第二級的中斷會使用 OR 邏輯運算來產生一個第一級相對應的中斷位元。

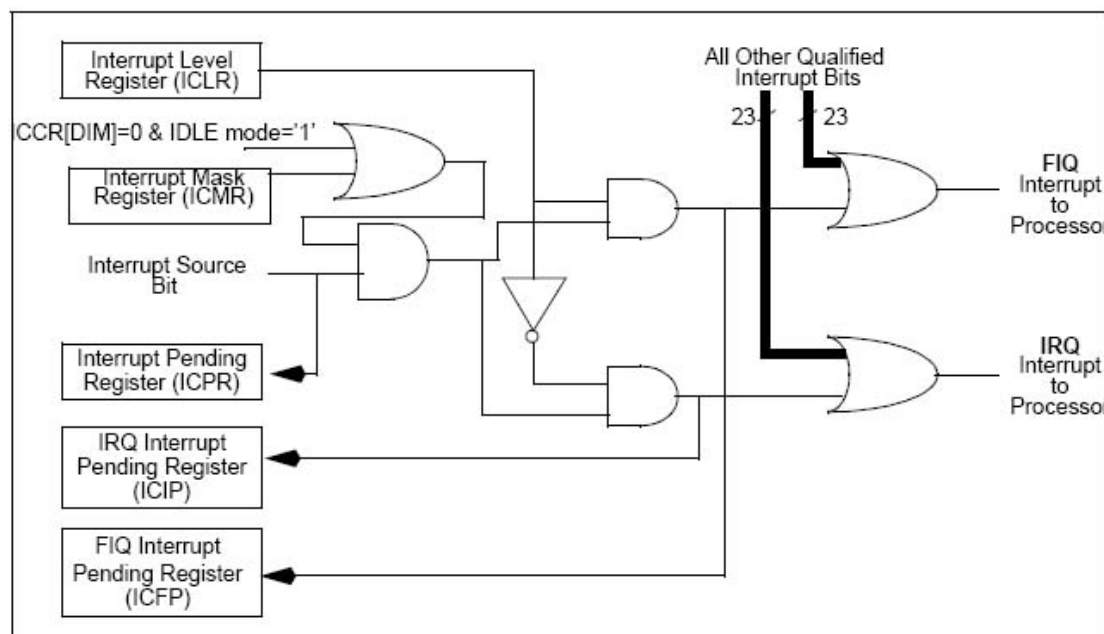
大部分的情況中，中斷的源頭可以藉由讀取兩個暫存器位址來決定：透過 ICIP

來辨識產生 IRQ 中斷的裝置，或透過 ICFP 來辨識產生 FIQ 中斷的裝置。接著你可以讀取裝置內部的狀態暫存器來獲得正確的請求服務功能。

當 ICCR[DIM]位元為 0 時，中斷遮蔽暫存器(Interrupr Mask register)在閒置模式(IDLE mode)期間將被忽略，所有已致能的中斷都可以使處理器離開閒置模式。否則，只有未遮蔽的中斷會使處理器離開閒置模式。ICCR[DIM]的重置狀態為 0。

6-23 頁圖 6-2 「中斷控制器區塊圖」說明中斷控制器的區塊圖。

圖 6-2 中斷控制器區塊圖



## 6.2.2 中斷控制器暫存器定義

中斷控制器包含下列幾個暫存器：

- 中斷控制器之 IRQ 等候暫存器(Interrupt Controller IRQ Pending register - ICIP)
- 中斷控制器之 FIQ 等候暫存器(Interrupt Controller FIQ Pending register - ICFP)
- 中斷控制器之等候暫存器(Interrupt Controller Pending register - ICPR)
- 中斷控制器之遮蔽暫存器(Interrupt Controller Mask register - ICMR)
- 中斷控制器之層級暫存器(Interrupt Controller Level register - ICLR)
- 中斷控制器之控制暫存器(Interrupt Controller Control register - ICCR)

重置之後，CPU 中的 FIQ 和 IRQ 中斷將被關閉，並且所有的中斷控制器暫存器的狀態被設定為 0x0。在 CPU 內的中斷再次被使用前，中斷控制器暫存器必須透過軟體來初始化。

6.2.2.1 中斷控制器之遮蔽暫存器(ICMR)

中斷控制器之遮蔽暫存器(Interrupt Controller Mask register - ICMR)每個等待(pending)中斷位元分別對應一個遮蔽位元(共 22 個)。這個遮蔽位元可以控制等待中斷位元是否會產生一個處理器中斷(IRQ 或 FIQ)。當等待中斷作用時，如果其對應的 ICMR 遮蔽位元為 1 時，該中斷便會被 CPU 處理。當閒置模式時，ICCR[DIM]必須將此遮蔽位元設定為有效，否則任何中斷來源的請求都會去設定其所對應的等待位元，並且將其中斷自動執行，而不去理會目前遮蔽位元的狀態。

為了預防引發中斷，遮蔽位元允許軟體週期性的去輪循(polling)可被中斷的來源。ICMR 暫存器在重置時被初始化為 0，這表示所有的中斷都被遮蔽，必須透過使用者來設定 ICMR，選擇其所要求的中斷。

表 6-31 說明中斷控制器之遮蔽暫存器的位元地圖。表 6-37 說明可使用的第一級中斷和它們在 ICPR 暫存器中的位置。

表 6-31 ICMR 暫存器位元地圖

Physical Address 0x40D0_0004										Interrupt Controller Mask Register (ICMR)										SysInt														
Bit	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		
	IM31	IM30	IM29	IM28	IM27	IM26	IM25	IM24	IM23	IM22	IM21	IM20	IM19	IM18	IM17	Reserved	Reserved	IM14	IM13	IM12	IM11	IM10	IM9	IM8	Reserved									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?		

位元	名稱	描述
<31:8>	IM[x]	第'x'個中斷遮蔽(x=8 to 14,17 to 31) 0=等待中斷被遮蔽(中斷不會被送到 CPU 或電源管理員) 1=等待中斷被允許(中斷會被送到CPU和電源管理員) 注意：在閒置模式中，如果 ICCR[DIM]被清除，則 IM 位元會被忽略。
<7:0>	—	保留

6.2.2.2 中斷控制器之層級暫存器(ICLR)

中斷控制器之層級暫存器(Interrupt Controller level register - ICLR)控制等待



中斷是否會產生 FIQ 或是 IRQ 中斷。如果等待中斷沒有被遮蔽，其所對應的 ICLR 位元會被解碼，並用來要觸發哪一種處理器中斷。如果中斷被遮蔽，則 ICLR 中所對應的位元將會變無效。在重置時，ICLR 暫存器所有位元全部被初始化成為 0，所以軟體必須設定 ICLR，回復到一般操作的值。

表 6-32 說明中斷控制器之層級暫存器的位元地圖。表 6-37 說明可使用的第一級中斷和它們在 ICPR 暫存器中的位置。

表 6-32 ICLR 暫存器位元地圖

		Physical Address 0x40D0_0008																Interrupt Controller Level Register (ICLR)														SysInt							
Bit		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						
		IL31	IL30	IL29	IL28	IL27	IL26	IL25	IL24	IL23	IL22	IL21	IL20	IL19	IL18	IL17	Reserved	Reserved	IL14	IL13	IL12	IL11	IL10	IL9	IL8	Reserved													
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	?					

位元	名稱	描述
<31:8>	IL[x]	第’x’個中斷位準(x=8 to 14,17 to 31) 0=將中斷導向至 IRQ 中斷輸入 1=將中斷導向至 FIQ 中斷輸入
<7:0>	—	保留

6.2.2.3 中斷控制器之控制暫存器(ICCR)

中斷控制器之控制暫存器(Interrupt Controller Control register - ICCR)內有一個控制位元，取消閒置遮蔽(Disable IDLE Mask - DIM)。當被設定時，這個位元禁止閒置模式的動作。在一般的閒置模式下，不管 ICMR 中的值為何，任何已致能的中斷都可以使應用處理器離開閒置模式。如果這個位元被清除，則中斷會依照 ICMR 所定義的值，讓應用處理器離開閒置模式。

注意：這個暫存器在重置之後會被清除。

表 6-33 說明中斷控制器之控制暫存器的位元地圖。表 6-37 說明可使用的第一級中斷和它們在 ICPR 暫存器中的位置。

表 6-33 ICCR 暫存器位元地圖

Physical Address 0x40D0_0014														Interrupt Controller Control Register (ICCR)														SysInt													
Bit	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																															DIM									
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0									

位元	名稱	描述
<31:1>	—	保留
<0>	DIM	取消閒置遮蔽。 0=所有已致能的中斷都可以讓應用處理器離開閒置模式。 1=只有已致能的及未遮蔽的(由ICMR定義)可使應用處理器離開閒置模式。 這個位元在重置後將會被清除。

6.2.2.4 中斷控制器之 IRQ 等待暫存器(ICIP)及 FIQ 等待暫存器(ICFP)

ICIP 及 ICFP 每個中斷都包含相對應的一個位元(每個各 22 個)。這些位元可以指出有單元已發出中斷請求。在中斷服務常式內部，藉由讀取 ICIP 和 ICFP 來確定中斷的來源。一般而言，軟體接著會讀取發生中斷的設備內的狀態暫存器，並決定如何服務這個中斷。為了表示所發生的中斷，會將 ICIP 及 ICFP 內狀態位元作邏輯 OR 運算，並將結果反應至 ICPR 中唯讀的位元(參考 6.2.2.5 節)。一旦中斷服務完畢，中斷處理常式將寫入 1 到相對應的狀態位元，並清除來源端的等待中斷。

清除來源端的中斷狀態位元時，其所對應的ICIP或ICFP旗標也會自動地清除，確保來源單元中沒有其他的中斷狀態位元。

表6-34說明中斷控制器之IRQ等待暫存器的位元地圖。表6-35說明中斷控制器之FIQ等待暫存器的位元位元地圖。表6-37說明可使用的第一級中斷和它們在ICPR暫存器中的位置。

表 6-34 ICIP 暫存器位元地圖

		Physical Address 0x40D0_0000								Interrupt Controller IRQ Pending Register (ICIP)																SysInt										
Bit		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			
		IP31	IP30	IP29	IP28	IP27	IP26	IP25	IP24	IP23	IP22	IP21	IP20	IP19	IP18	IP17	Reserved	Reserved	IP14	IP13	IP12	IP11	IP10	IP9	IP8	Reserved										
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?			

位元	名稱	描述
<31:8>	IP[x]	第'x'個 IRQ 等待(x=8 to 14,17 to 31) 0=沒有任何已致能的來源端之 IRQ 請求 1=已致能的來源端之 IRQ 請求。
<7:0>	—	保留

表 6-35 ICFP 暫存器位元地圖

		Physical Address 0x40D0_000C																Interrupt Controller FIQ Pending Register (ICFP)										SysInt							
Bit		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		
		FP31	FP30	FP29	FP28	FP27	FP26	FP25	FP24	FP23	FP22	FP21	FP20	FP19	FP18	FP17	Reserved	Reserved	FP14	FP13	FP12	FP11	FP10	FP9	FP8	Reserved									
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?		

位元	名稱	描述
<31:8>	FP[x]	第'x'個 IRQ 等待(x=8 to 14,17 to 31) 0=沒有任何已致能的來源端之 FIQ 請求 1=已致能的來源端之 FIQ 請求。
<7:0>	—	保留

#### 6.2.2.5 中斷控制器之等待暫存器(ICPR)

ICPR 為一個 32 位元唯讀的暫存器，它顯示系統裡所有作用中的中斷。這些位元不會受到遮蔽暫存器(ICMR)的狀態所影響。清除來源端的中斷狀態位元，也會自動的清除其所對應的 ICPR 旗標，確保來源單元中不會有其他的中斷狀態位元。表 6-36 說明中斷控制器之等待暫存器的位元地圖。

表 6-37 說明 ICPR 中的每個位元位置被指定到的等待中斷來源。同時表中也包含了中斷來源單元和每個相關連的第二級中斷的號碼。要瞭解第二級中斷更多資訊，請參考來源單元之欄位。

表 6-36 ICPR 暫存器位元地圖

Physical Address 0x40D0_0010										Interrupt Controller Pending Register (ICPR)																		SysInt							
Bit	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			
	IS31	IS30	IS29	IS28	IS27	IS26	IS25	IS24	IS23	IS22	IS21	IS20	IS19	IS18	IS17	Reserved		IS14	IS13	IS12	IS11	IS10	IS9	IS8								Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

位元	名稱	描述
<31>	IS31	RTC警報符合暫存器之中斷等待(RTC Alarm Match Register Interrupt Pending) 0=由於RTC警報符合暫存器，使得中斷沒有被等待。 1=由於RTC警報符合暫存器，使得中斷被等待。
<30>	IS30	RTC赫茲時脈滴答中斷等待(RTC HZ Clock Tick Interrupt Pending) 0=由於RTC赫茲時脈滴答，使得中斷沒有被等待。 1=由於RTC赫茲時脈滴答，使得中斷被等待。
<29>	IS29	OS計時器之符合暫存器3中斷等待(OS Timer Match Register 3 Interrupt Pending) 0=由於OS計時器之符合暫存器3，使得中斷沒有被等待。 1=由於OS計時器之符合暫存器3，使得中斷被等待。
<28>	IS28	OS計時器之符合暫存器2中斷等待(OS Timer Match Register 2 Interrupt Pending) 0=由於OS計時器之符合暫存器2，使得中斷沒有被等待。 1=由於OS計時器之符合暫存器2，使得中斷被等待。
<27>	IS27	OS計時器之符合暫存器1中斷等待(OS Timer Match Register 1 Interrupt Pending) 0=由於OS計時器之符合暫存器1，使得中斷沒有被等待。 1=由於OS計時器之符合暫存器1，使得中斷被等待。
<26>	IS26	OS計時器之符合暫存器0中斷等待(OS Timer Match Register 0 Interrupt Pending) 0=由於計時器符合暫存器 0，使得中斷沒有被等待。 1=由於計時器符合暫存器 0，使得中斷被等待。
<25>	IS25	DMA通道服務請求中斷等待(DMA Channel Service Request Interrupt Pending) 0=由於DMA通道服務請求，使得中斷沒有被等待。 1=由於 DMA 通道服務請求，使得中斷被等待。
<24>	IS24	SSP 服務請求中斷等待(SSP Service Request Interrupt Pending)

		0=由於SSP服務請求，使得中斷沒有被等待。 1=由於 SSP 服務請求，使得中斷被等待。
<23>	IS23	MMC 狀態/錯誤之偵測中斷等待(MMC Status/Error Detection Interrupt Pending) 0=由於MMC狀態/錯誤偵測，使得中斷沒有被等待。 1=由於MMC狀態/錯誤偵測，使得中斷被等待。
<22>	IS22	FFUART傳送/接收/錯誤之中斷等待(FFUART Transmit/Receive/Error Interrupt Pending) 0=由於FFUART傳送/接收/錯誤，使得中斷沒有被等待。 1=由於 FFUART 傳送/接收/錯誤，使得中斷被等待。
<21>	IS21	BTUART傳送/接收/錯誤之中斷等待(BTUART Transmit/Receive/Error Interrupt Pending) 0=由於BTUART傳送/接收/錯誤，使得中斷沒有被等待。 1=由於 BTUART 傳送/接收/錯誤，使得中斷被等待。
<20>	IS20	STUART傳送/接收/錯誤之中斷等待(STUART Transmit/Receive/Error Interrupt Pending) 0=由於STUART傳送/接收/錯誤，使得中斷沒有被等待。 1=由於 STUART 傳送/接收/錯誤，使得中斷被等待。
<19>	IS19	ICP傳送/接收/錯誤之中斷等待(ICP Transmit/Receive/Error Interrupt Pending) 0=由於ICP傳送/接收/錯誤，使得中斷沒有被等待。 1=由於 ICP 傳送/接收/錯誤，使得中斷被等待。
<18>	IS18	I2C服務請求中斷等待(I2C Service Request Interrupt Pending) 0=由於I2C服務請求，使得中斷沒有被等待。 1=由於 I2C 服務請求，使得中斷被等待。
<17>	IS17	LCD控制器服務請求中斷等待(LCD Controller Service Request Interrupt Pending) 0=由於控制器服務請求，使得中斷沒有被等待。 1=由於控制器服務請求，使得中斷被等待。
<16:15>	—	保留
<14>	IS14	AC97中斷等待(AC97 Interrupt Pending) 0=由於AC97單元，使得中斷沒有被等待。 1=由於 AC97 單元，使得中斷被等待。
<13>	IS13	I2S中斷等待(I2S Interrupt Pending) 0=由於I2S單元，使得中斷沒有被等待。 1=由於 I2S 單元，使得中斷被等待。
<12>	IS12	性能監控單元(PMU)中斷等待(Performance Monitoring

		Unit (PMU) Interrupt Pending) 0=由於PMU單元，使得中斷沒有被等待。 1=由於 PMU 單元，使得中斷被等待。
<11>	IS11	USB服務中斷等待(USB Service Interrupt Pending) 0=由於USB服務的請求，使得中斷沒有被等待。 1=由於 USB 服務的請求，使得中斷被等待。
<10>	IS10	GPIO[80:2]邊緣偵測中斷等待(GPIO[80:2] Edge Detect Interrupt Pending) 0=由於一個(或多個)GPIO[80:2]的邊緣偵測，使得中斷沒有被等待。 1=由於一個(或多個)GPIO[80:2]的邊緣偵測，使得中斷被等待。
<9>	IS9	GPIO[1] 邊緣偵測中斷等待(GPIO[1] Edge Detect Interrupt Pending) 0=由於GPIO[1]的邊緣偵測，使得中斷沒有被等待。 1=由於 GPIO[1]的邊緣偵測，使得中斷被等待。
<8>	IS8	GPIO[0] 邊緣偵測中斷等待(GPIO[0] Edge Detect Interrupt Pending) 0=由於GPIO[0]的邊緣偵測，使得中斷沒有被等待。 1=由於 GPIO[0]的邊緣偵測，使得中斷被等待。
<7:0>	—	保留

表 6-37 第一級中斷列表

位元位置	來源單元	第二級來源編號	位元描述
IS<31>	即時時脈 (Real-time clock)	1	RTC 等於警報暫存器(RTC equals alarm register)
IS<30>		1	Hz 時脈 TIC 發生(One Hz clock TIC occurred)
IS<29>	作業系統計時器 (Operating system timer)	1	OS 計時器等於符合暫存器 3(OS timer equals match register 3)
IS<28>		1	OS 計時器等於符合暫存器 2(OS timer equals match register 2)
IS<27>		1	OS 計時器等於符合暫存器 1(OS timer equals match register 1)
IS<26>		1	OS 計時器等於符合暫存器 0(OS timer equals match register 0)
IS<25>	DMA 控制器	16	DMA 通道服務請求(DMA Channel

	(DMA controller)		service request)
IS<24>	同步序列埠 (Synchronous Serial Port)	3	SSP 服務請求(SSP service request)
IS<23>	多媒體卡(Multi Media Card)	9	MMC 狀態/錯誤之偵測(MMC status / error detection)
IS<22>	FFUART	5	FFUART 傳送/接收/錯誤(x-mit, receive, error in FFUART)
IS<21>	BTUART	5	BTUART 傳送/接收/錯誤(x-mit, receive, error in BTUART)
IS<20>	STUART	4	STUART 傳送/接收/錯誤(x-mit, receive, error in STUART)
IS<19>	ICP	6	ICP 傳送/接收/錯誤(x-mit, receive, error in ICP)
IS<18>	I2C	6	I2C 服務請求(I2C service request)
IS<17>	LCD 控制器(LCD controller)	15	LCD 控制器服務請求(LCD controller service request)
IS<16>			保留
IS<15>			保留
IS<14>	AC97	10	AC97 中斷(AC97 interrupt)
IS<13>	I2S	5	I2S 中斷(I2S interrupt)
IS<12>	核心(Core)	1	PMU 中斷(PMU interrupt)
IS<11>	USB	7	USB 中斷(USB interrupt)
IS<10>		79	GPIO<80> 至 GPIO<2>邊緣偵測 之邏輯 OR 運算(“OR” of GPIO edge detects 80-2)
IS<9>		1	GPIO<1>邊緣偵測(GPIO<1> edge detect)
IS<8>		1	GPIO<0>邊緣偵測(GPIO<0> edge detect)
IS<7>			保留
IS<6>			保留
IS<5>			保留
IS<4>			保留
IS<3>			保留
IS<2>			保留
IS<1>			保留
IS<0>			保留

	第二級中斷來源總數 (Total level 2 interrupt sources)	179	
--	--	-----	--

有幾個單元的中斷信號有一個以上的來源。當一個中斷是從這幾個單元發出信號時，中斷處理常式透過中斷控制器的等待暫存器辨識出是哪一個中斷被觸發。這會辨識出發出請求的單元，但無法辨識出哪一個是正確的來源。接著處理常式讀取中斷單元的狀態暫存器，藉以辨識該單元中哪一個來源發出中斷的信號。對所有的中斷而言都有其對應的來源，中斷控制常式只需要使用中斷控制器的暫存器來辨識正確的中斷來源。ICPR[16:15]和 ICPR[7:0]為保留位元，寫入必須為 0。讀取這些位元必須被忽略。

### 6.2.3 中斷控制器暫存器位置

表 6-38 說明與中斷控制器區塊相關連的暫存器及它們的實體位址。

表 6-38 中斷控制器暫存器位址

位址	名稱	描述
0x40D0_0000	ICIP	中斷控制器之IRQ等待暫存器
0x40D0_0004	ICMR	中斷控制器之遮蔽暫存器
0x40D0_0008	ICLR	中斷控制器之層級暫存器
0x40D0_000C	ICFP	中斷控制器之FIQ等待暫存器
0x40D0_0010	ICPR	中斷控制器之等待暫存器
0x40D0_0014	ICCR	中斷控制器之控制暫存器

## 6.3 即時時脈

在廣大的頻率範圍內，我們可以利用即時時脈來對一個時脈來源做設定。一般而言，RTC 被設定為 1Hz 輸出，並且被做為系統時間的維持者(keeper)。如果 RTC 的輸出時脈增加並到達一個預設值時，可以利用警報的功能，將中斷或喚醒事件致能。



### 6.3.1 即時時脈操作

對於您的設計來說，即時時脈(RTC)可以提供一個通用的即時參考。在硬體重置或是看門狗重置後，RTC 計數器暫存器(RTC Counter register - RCNR)將被初始化為 0。這是個不受限制的計數器，在重置被移除之後，其計數值才會開始逐漸增加。計數器只有在 HZ 時脈的正緣才會增加。將計數器設定為所希望的值。如果計數器被設定一個非 0 的值，則將這個要求的值寫入 RCNR。這個計數器的值不會受到進入和離開睡眠或閒置模式轉換的影響。

除了 RCNR 之外，RTC 還包含了一個 32 位元的 RTC 警報暫存器(RTC Alarm register - RTAR)。RTAR 可以被設定一個值來與 RCNR 相比較。在每個 HZ 時脈的正緣階段時，計數器會增加並且與 RTAR 做比較。如果數值符合，且該致能位元被設定，則 RTC 狀態暫存器之警告符合位元(RTSR[AL])會被設定。這個狀態位元會被導向到中斷控制器，若將此位元在中斷控制器內解除遮蔽，則可以用來產生一個處理器中斷。另一個可用的中斷狀態位元為 RTSR[HZ]，其中斷狀態位元無論 HZ 時脈狀態轉變為何，皆會將其中斷狀態反應至該位元。寫入 1 至 RTSR 中的 AL 或 HZ 位元，其狀態位元就會被清除。

HZ 時脈是藉由兩個時脈來源擇一做除法運算所產生，兩個的頻率都接近 32.768kHz。第一個來源是將 3.6864MHz 的晶體振盪器除以 112 後的輸出，其頻率接近 32.914kHz。另一個來源是選配的 32.768kHz 晶體振盪器本身的頻率輸出。你的系統可以由 32.768kHz 晶體振盪器和 3.6864MHz 晶體振盪器兩者共同建構而成。如果在睡眠模式中額外消耗的電力是可接受的話，你的系統可以只使用 3.6864MHz 晶體振盪器。

產生 HZ 時脈的除法器邏輯是可以程式化的。這可以讓你透過對計數器的修整，來調整不精確的晶體頻率，及調整透過 3.6864MHz 振盪器除法所產生的近似的 32kHz 頻率。這個修整的機制可以讓你將 RTC 機確度控制在每月 $\pm$ 五秒之內。調整的程序將在後面的段落中說明。

所有在 RTC 中的暫存器，除了 RTTR 之外，都透過硬體重置或是看門狗重置來重置。RTTR 這個修整暫存器只能透過硬體重置來重置。

### 6.3.2 RTC 暫存器定義

接下來的章節將提供即時時脈暫存器的說明。

6.3.2.1 RTC 修整暫存器(RTTR)

對 RTTR (RTC Trim register)程式化可以用來設定 HZ 時脈的頻率。這個暫存器的重置值(0x0000\_7FFF)(假設是使用一顆完美的 32.768kHz 晶體)將可以產生剛好為 1Hz 的 HZ 時脈輸出。在上述情況下，如果使用 0x0000\_7FFF 之外的數值，就有可能會產生不同的 HZ 時脈頻率。此外，你使用的晶體可能不會剛好為 32.768kHz，你必須寫入 0x0000\_7FFF 之外的數值到 RTTR 中來做補償。6.3.3 節「修整程序」將說明如何計算這個暫存器中的數值。

表 6-39 說明 RTC 修整暫存器的位元地圖。所有的保留位元寫入必須為 0，並且讀取必須被忽略。你只可以使用硬體重置來重置 RTTR。為了保護寫入修整暫存器之資料的有效性，第 31 位元用來當作鎖定位元(Lock Bit)。只有當 RTTR(31)為 0 時，RTTR 中的資料才可以改變。一旦 RTTR(31)被設定為 1 時，只有硬體重置可以清除 RTTR。

表 6-39 RTTR 暫存器位元地圖

Physical Address 0x4090_000C		RTC Trim Register (RTTR)																SysInt															
Bit	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	
	LCK	Reserved					DEL										CK_DIV																
Reset	0	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

位元	名稱	描述
<31>	LCK	修整值的鎖定位元 0=RTTR 的值允許改變 1= RTTR 的值不允許改變
<30:26>	—	保留
<25:16>	DEL	修整刪除計數(Trim delete count) 當時脈修整開始時，這個值代表32kHz時脈會被刪除的數量。
<15:0>	CK_DIV	時脈除數計數 這個值是32kHz時脈週期的數量，每個HZ時脈週期加1。

6.3.2.2 RTC 警報暫存器(RTAR)

即時時脈警報暫存器(real-time clock alarm register)為一個 32 位元的暫存器。應用處理器可以讀取及寫入這個暫存器。在每一個 HZ 時脈的正緣，這個暫

存器都會與 RCNR 做比較。如果兩者相同而且 RTSR[ALE]被設定時，則 RTSR[AL]也會被設定。

相對於處理器時脈，HZ 時脈具備非同步的性質，對此暫存器做寫入是由硬體機制來控制的，當處理器寫入之後，硬體將會延遲 2 個 32kHz 時脈週期後才會實際寫入暫存器。

RTAR 暫存器在重置後被初始化為 0x0。

表 6-40 說明 RTC 警報暫存器的位元地圖。

表 6-40 RTAR 暫存器位元地圖

Physical Address 0x4090_0004										RTC Alarm Register (RTAR)										SysInt												
Bit	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
	RTMV																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位元	名稱	描述
<31:0>	RTMV	RTC 目標符合值(RTC Target Match Value) 這個值與 RTC 計數器做比較。

6.3.2.3 RTC 計數器暫存器(RCNR)

RTC 計數器暫存器(RCNR)為一個可以讀/寫的暫存器。雖然我們建議透過作業系統使用 MMU 保護機制來避免不注意地寫入 RCNR，但是該計數器還是可以在任何時間被處理器寫入，雖然建議作業系統在使用 MMU 保護機制時應該避免不注意地寫入 RCNR。(參考 *Megacell 使用者手冊*瞭解 MMU 詳細的操作)。

相對於處理器時脈，HZ 時脈具備非同步的性質，對此暫存器做寫入是由硬體機制來控制的，當處理器寫入之後，硬體將會延遲 2 個 32kHz 時脈週期後才會實際寫入暫存器。若如果連續多個寫入到 RCNR，最後一個對 RCNR 計數器的更新可能會被延遲 2 個 32kHz 時脈週期。

RCNR 可以在任何時間讀取。在增加或是被寫入之後，讀取計數將反應出暫存器的值，而不會有 2 個 32kHz 時脈週期的延遲。

表 6-41 說明 RTC 計數器暫存器的位元地圖。

表 6-41 RCNR 暫存器位元地圖

Physical Address 0x4090_0000										RTC Counter Register (RCNR)																		SysInt									
Bit	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					
	RCV																																				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					

位元	名稱	描述
<31:0>	RCV	RTC 計數值 RTC 計數器目前的值。

6.3.2.4 RTC 狀態暫存器(RTSR)

RTC 狀態暫存器(RTC Status register - RTSR)在硬體重置後全部被清除為 0。ALE 和 HZE 位元可以同時致能 AL 和 HZ 位元的更新及中斷。若 ALE 和 HZE 位元分別被設定，則 AL 和 HZ 位元這兩個狀態位元將會被 RTC 邏輯所設定。藉由寫入 1 到 AL 和 HZ 位元可以將它們清除。AL 和 HZ 位元被導向至中斷控制器，引發第一級中斷。寫入 0 到所有的保留位元，並且忽略所有保留位元的讀取。

在睡眠模式中，只有 AL 事件可以設定 RTSR 暫存器中的狀態位元。HZ 位元並不會在睡眠模式中被設定，因為這將是一個中斷的循環事件。

表 6-42 說明 RTC 狀態暫存器的位元地圖。

表 6-42 RTSR 暫存器位元地圖

Physical Address		RTC Status Register (RTSR)																SysInt															
0x4090_0008																																	
Bit		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																															
Reset		?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
位元	名稱	描述																															
<31:4>	—	保留																															
<3>	HZE	HZ 中斷致能 0=HZ 中斷不致能 1=HZ 中斷致能																															
<2>	ALE	RTC 警報中斷致能 0=RTC 警報中斷不致能																															

		1=RTC 警報中斷致能
<1>	HZ	HZ 正緣偵測 0=沒有正緣被偵測。 1=正緣被偵測，並且 HZE 位元被設定。
<0>	AL	RTC 警報偵測 0=沒有 RTC 警報被偵測。 1=RTC 警報被偵測(RTNR 與 RCNR 符合)，並且 ALE 位元被設定。

### 6.3.3 修整程序

HZ 時脈透過除以振盪多工器的輸出來驅動 RTC。因為晶體本身所具備的不精確性質，以及電路板上所產生的各種電容現象，就像其他的變數一樣，都可能會導致基準時間不精確。這需要對時脈週期做些許的調整。透過 RTTR，應用處理器可以讓你調整(或修整)HZ 基準時間至小於 1ppm 的誤差。因此如果 HZ 時脈被設定為 1 Hz，每個月應該會有少於五秒的誤差。

每當 nRESET 信號被觸發，RTTR 就會被重置為預設值 0x0000\_7FFFF。如此將產生近似於 1 Hz 的時脈。

當時脈除數計數(clock divisor count)(RTTR[15:0])被設定為 0x0 時，HZ 時脈提供 RTC 一個高位準的信號—實際上就是關閉 RTC。對於其他非零的數值設定在時脈除數計數中，HZ 時脈頻率將會是 32kHz 時脈來源除以時脈除數計數之後再加上 1。

#### 6.3.3.1 振盪器頻率測定(Calibration)

為了確定 RTTR 中程式化的值，你必須先使用一個精確的時間基底，如頻率計數器，來測量振盪多工器的輸出頻率(接近 32kHz)。這個時脈在外部是可取得的，透過選擇 32 位元系統中 GPIO[12]或 GPIO[72]的交替功能，或是 PXA210 中 GPIO[72]的交替功能來取得。為了利於取得時脈，可以將腳位設定為輸出，並且切換到交替功能。參考 6.1 節，以瞭解如何讓時脈為外部取得。為了修整時脈，將振盪器的輸出除以一个整數，並藉由週期性的刪除來做少量的調整。

### 6.3.3.2 RTTR 數值計算

在知道了振盪器的真實頻率之後，必須將該頻率除以所希望的 HZ 時脈頻率，其結果會分成整數與小數兩個部分。數值的整數部分(減 1)被讀取到 RTTR 的時脈除數計數欄位。這個值會跟一個接近 32kHz 的振盪多工器的輸出(由 16 位元計數器所計數)做比較。當兩個值相同時，計數器將會重置並且產生一個脈衝來構成原始的 HZ 時脈信號。

調整的小數的部分是透過週期性地刪除時脈(從時脈串流所驅動的整數計數器)來完成。修整的區間週期被硬體佈線為  $2^{10}-1$  個 HZ 時脈週期。如果 HZ 時脈被設定為 1 Hz，修整區間就會接近 17 分鐘。時脈刪除的個數(修整刪除的數值)為一個 10 位元的可程式化的計數器。在一次的修整區間內，允許從時脈串流中一次刪除 0 到  $2^{10}-1$  個 32kHz 時脈。RTTR[25:16]代表每次修整操作中被刪除的 32kHz 時脈的個數。

總之，每個  $2^{10}-1$  HZ 時脈週期，整數計數器停止計時一段時間，該段時間等於累積的小數部分的誤差。如果小數誤差被程式化為 0，就不會有修整操作的發生，且 RTC 是以原始的 32kHz 時脈供應時脈。HZ 時脈頻率和名義上的 32kHz 時脈(各為 f1 和 f32K)的關係如下列的方程式。

$$f1 = \frac{(2^{10}-1) \cdot (RTTR[CK\_DIV]+1) - RTTR[DEL]}{(2^{10}-1) \cdot (RTTR[CK\_DIV]+1)} * \frac{f32k}{(RTTR[CK\_DIV]+1)}$$

f1 = HZ時脈頻率

f32k = RTC內部時脈—32.678 kHz晶體輸出或3.68 MHz晶體除至32.914 kHz的輸出

RTTR[DEL] = RTTR(25:16)

RTTR[CK\_DIV] = RTTR(15:0)

#### 6.3.3.2.1 修整範例#1—測量值沒有小數成分

在這個範例中，所希望的 HZ 時脈頻率為 1 Hz。振盪器輸出經測量後為 36045.000 cycles/s(Hz)。此振盪器頻率的輸出剛好超過名義上的 32.768kHz 有 3277 週(cycle)之多，且沒有小數部分。因此，只需要整數修整功能，而不需要小數修整功能。因此，RTTR[15:0]被載入相等於 36045-1 的二進位值，或是 0x0000\_8CCC。RTTR[25:16]的值載入為 0(電源開機狀態)來關閉小數修整功能。這個修整例子在修整的過程中留下了為 0 的錯誤。

### 6.3.3.2.2 修整範例#2—測量值有小數成分

這個範例在測量振盪器頻率時，為常見的有小數成分的頻率。一樣地，希望的 HZ 時脈輸出頻率為 1 Hz。如果振盪器輸出經測量後為 32768.92 cycles/s (Hz)，整數修整是必要的，因此在產生 1Hz 時脈之前已計數的週期平均個數為 32768.92。跟上個範例類似，RTTR[15:0]的整數欄位被載入 32768-1 或是十六進位的 0x0000\_7FFF(重設值)。

因為實際的時脈頻率比整數數值快0.92週/每秒，由整數修整產生的HZ時脈稍微比需要的快，所以必須調降下來。因此，設定小數修整功能來刪除0.92週/每秒，讓HZ輸出頻率下降至適當的值。這個修整程序每隔1023(2<sup>10</sup>-1)秒就被執行一次，所以修整必須被設定成每隔1023秒就刪除941.16個時脈(0.92 x 1023 = 941.16)。在計數器中載入941的十六進位值：0x3AD。這個數值的小數部分不能被修整掉，且構成修整中的誤差，說明如下。

修整設定產生一個每1023秒有0.16週誤差的結果，誤差計算結果(parts-per-million - ppm)為：

$$\text{Error} = \frac{0.16 \text{ cycles}}{1023 \text{ sec}} \times \frac{1 \text{ sec}}{32768 \text{ cycles}} = 0.002 \text{ ppm}$$

### 6.3.3.2.3 最大誤差計算與即時時脈的準確性

從修整範例#2 中可知，最大的可能誤差接近 1 個時脈/每 2<sup>10</sup>-1 秒。計算這個例子產生的 ppm 誤差：

$$\text{Error (maximum)} = \frac{1 \text{ cycle}}{1023 \text{ sec}} \times \frac{1 \text{ sec}}{32768 \text{ cycles}} = 0.03 \text{ ppm}$$

為了維持每個月+/-五秒的精確性，所需要的精確性計算為：

$$\text{Error} = \frac{5 \text{ sec}}{\text{month}} \times \frac{1 \text{ month}}{2592000 \text{ sec}} = 1.9 \text{ ppm}$$

這個計算指出經過修整程序之後，HZ 時脈輸出可以設定得非常精準。同樣

地，使用修整程序來補償某個範圍內會影響晶體振盪器的因素。這些因素可以包含如下，但也不限制在以下範圍：

- 製造商和供應商之晶體的差異
- 晶體老化
- 系統電壓之差異
- 系統製造商之差異

調整程序可以透過一個精確的機制來移除差異，以及消除在個別系統中系統製造商和環境的變數，藉以抵銷上述因素。然而，這只是一個量測的解法，並不能當做動態系統與環境改變中的實際解法，只能在擁有設備的工廠中才能如此設定完成。

### 6.3.4 即時時脈暫存器位置

表 6-43 說明即時時脈暫存器的位置。

表 6-43 RTC 暫存器位址

位址	名稱	解釋
0x4090_0000	RCNR	RTC 計數暫存器
0x4090_0004	RTAR	RTC 警報暫存器
0x4090_0008	RTSR	RTC 狀態暫存器
0x4090_000C	RTTR	RTC 修整暫存器

## 6.4 作業系統計時器

應用處理器包含一個由一個3.6864 MHz振盪器所驅動時脈的32位元作業系統計時器。作業系統計數暫存器(Operating System Count register - OSCR)是一個不會受限制的上數計數器。OS計數器包含四個32位元符合暫存器(OSMR3、OSMR2、OSMR1、OSMR0)，你可以讀取和寫入每個暫存器。當OSCR中的值與任何一個符合暫存器中的值相同時，中斷致能位元會被設定，OSSR中所對應的位元也會被設定。這些位元也會導向至中斷控制器，並且也可以被程式化成產生中斷。當符合發生時，OS計時器之看門狗符合致能暫存器(OS Timer Watchdog Enable Register - OWER)被設定，OSMR3也可以當作用來重置應用處理器的看門狗符合暫存器。你必須初始化OSCR和OSMR暫存器，並且在CPU中的FIQ和IRQ中斷產生之前清除任何已設定的狀態位元。



### 6.4.1 看門狗計時器之運作

你也可以使用 OSMR3 來當作看門狗比較暫存器(watchdog compare register)。這個功能藉由設定 OWER[0]來致能。當看門狗被致能，且該比較與暫存器符合時，應用處理器將被重置，其大部分的內部狀態都會被清除。內部重置被觸發 256 個處理器時脈之後移除，並允許應用處理器開機。參考 3.4.2 節「看門狗重置」，瞭解重置功能的細節。

每次當作業系統提供服務給暫存器，並使用 OSMR3 當作看門狗時，建議使用以下程序：

1. 讀取目前計數器中的值。
2. 一個偏移量(offset)被加到所讀取的數值上。這個偏移量相當於下一次逾時之前的時間數(謹慎估計此計數器)。
3. 更新的數值被寫回到 OSMR3。

OS 程式碼必須在每個符合發生之前，週期性地重複上述程序。只要有一個符合產生時，OS 計時器便會觸發一個重置給應用處理器。

### 6.4.2 OS 計時器暫存器定義

#### 6.4.2.1 OS 計時器之符合暫存器 0-3(OSMR0,OSMR1,OSMR2,OSMR3)

這些暫存器都是 32 位元且可以讓處理器讀寫的。這些暫存器在每個 3.6864 MHz 時脈的正緣發生之後與 OSCR 做比較。如果其中任何一個暫存器與計數器暫存器符合，且適當的中斷致能位元被設定時，那麼 OSSR 中所對應的狀態位元會被設定。這些狀態位元會被重導至中斷控制器，他們可以藉由解除遮蔽來產生 CPU 中斷。你也可以使用 OSMR3 來當作看門狗計時器。

表 6-44 說明 OS 計時器之符合暫存器的位元地圖。所有的四個暫存器除了它們的位置之外，其餘都是相同的。雖然只單獨說明一個 OS 計時器之符合暫存器，但是其餘所有的資訊對四個 OS 計時器之符合暫存器來說都是相同的。

表 6-44 OSMR[x]暫存器位元地圖

Physical Address		OS Timer Match Registers (OSMR3, OSMR2, OSMR1, OSMR0)																								SysInt						
0x40A0_0000																																
0x40A0_0004																																
0x40A0_0008																																
0x40A0_000C																																
Bit	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
	OSMV																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位元	名稱	描述
<31:0>	OSMV	OS 計時器符合值 該值為與 OS 計時器之計數器做比較。

#### 6.4.2.2 OS計時器之中斷致能暫存器(OIER)

這個暫存器包含四個致能位元來指示四個符合暫存器中哪一個是符合的，而且OS計時器計數器將會在OSSR中設定一個狀態位元。每個符合暫存器皆有一個對應的致能位元。清除一個致能位元並不表示所對應的已設定的中斷狀態位元會被清除。

表6-45說明OS計時器之中斷致能暫存器的位元地圖。

表 6-45 OIER 暫存器位元地圖

Physical Address 0x40A0_001C																OS Timer Interrupt Enable Register (OIER)																SysInt				
Bit	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																																E3	E2	E1	E0
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0			

位元	名稱	描述
<31:4>	—	保留
<3>	E3	中斷致能通道 3 0=OSMR3 和 OS 計時器符合時，OSSR[M3]不會被觸發。 1=OSMR3 和 OS 計時器符合時，OSSR[M3]會被觸發。
<2>	E2	中斷致能通道 2 0=OSMR2 和 OS 計時器符合時，OSSR[M2]不會被觸發。 1=OSMR2 和 OS 計時器符合時，OSSR[M2]會被觸發。
<1>	E1	中斷致能通道 1 0=OSMR1 和 OS 計時器符合時，OSSR[M1]不會被觸發。 1=OSMR1 和 OS 計時器符合時，OSSR[M1]會被觸發。
<0>	E0	中斷致能通道 0 0=OSMR0 和 OS 計時器符合時，OSSR[M0]不會被觸發。 1=OSMR0 和 OS 計時器符合時，OSSR[M0]會被觸發。

#### 6.4.2.3 OS計時器之看門狗符合致能暫存器(OWER)

看門狗致能暫存器包含一個控制位元(bit 0)來致能看門狗的功能。這個位元藉由寫入一個1來設定，並且只有下列重置可以清除，如硬體重置、睡眠重置、看門狗重置、和GPIO重置。

表6-46顯示OS看門狗之符合致能暫存器(OS Watchdog Match Enable register)的腳位地圖。

**表 6-46 OWER 暫存器腳位地圖**

Physical Address 0x40A0_0018										OS Timer Watchdog Match Enable Register (OWER)										SysInt												
Bit	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																														WME	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位元	名稱	描述
<31:1>	—	保留
<0>	WME	看門狗之符合致能 0=OSMR3 符合時，不會產生應用處理器的重置。 1=OSMR3 符合時，會產生應用處理器的重置。

#### 6.4.2.4 OS計時器之計數暫存器(OSCR)

OS 計時器之計數暫存器是一個 32 位元的計數器，在 3.6864 MHz 時脈的正緣時增加計數。這個計數器可以隨時被讀寫。建議系統透過 MMU 保護機制來對這個暫存器的寫入做保護。

OSCR被寫入後，在暫存器實際被更新之前會有些延遲。軟體在傳遞暫存器的內容之前，必須確定暫存器已經更新為新的數值。

表6-47說明OS計時器之計數暫存器的腳位地圖。

**表 6-47 OSCR 暫存器位元地圖**

Physical Address 0x40A0_0010										OS Timer Count Register (OSCR)																SysInt							
Bit	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	
	OSCV																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位元	名稱	描述
----	----	----

<31:0>	OSCV	OS 計時器之計數器的數值。 目前 OS 計時器之計數器的數值。
--------	------	-------------------------------------

6.4.2.5 OS計時器之狀態暫存器(OSSR)

這個狀態暫存器包含狀態位元，且這些位元會指示出四個比較暫存器和 OSCR 中符合的發生。當發生符合事件時這些位元就會被設定(跟隨著 3.6864 MHz 時脈的正緣之後)，而且 OIER 暫存器中對應的中斷致能位元會被設定。寫入 1 到對應的位元位置就可以清除 OSSR 位元。寫入 0 到這個暫存器並不會有作用。對於保留位元應該寫入為 0 忽略所有的讀取。

表 6-47 說明 OS 計時器之狀態暫存器的位元地圖。

OSSR 暫存器位元地圖		
Physical Address 0x40A0_0014      OS Timer Status Register (OSSR)      SysInt		
Bit	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	
		M3 M2 M1 M0
Reset	? ?	0 0 0 0

位元	名稱	描述
<31:4>	—	保留
<3>	M3	符合狀態通道 3 如果 OIER[3]被設定： 0=自從上次被清除後，OSMR[3]不符合於 OS 計時器之計數器。 1=OSMR[3]符合於 OS 計時器之計數器。
<2>	M2	符合狀態通道 2 如果 OIER[2]被設定： 0=自從上次被清除後，OSMR[2]不符合於 OS 計時器之計數器。 1=OSMR[2]符合於 OS 計時器之計數器。
<1>	M1	符合狀態通道 1 如果 OIER[1]被設定： 0=自從上次被清除後，OSMR[1]不符合於 OS 計時器之計數器。 1=OSMR[1]符合於 OS 計時器之計數器。
<0>	M0	符合狀態通道 0

		<p>如果 OIER[0]被設定：</p> <p>0=自從上次被清除後，OSMR[0]不符合於 OS 計時器之計數器。</p> <p>1=OSMR[0]符合於 OS 計時器之計數器。</p>
--	--	---

### 6.4.3 OS計時器暫存器位置

表6-48說明與OS計時器相關的暫存器以及存取的實體位址。

**表6-48 OS計時器暫存器位置**

位址	名稱	解釋
0x40A0_0000	OSMR0	OS 計時器之符合暫存器 0 (OS timer match register 0)
0x40A0_0004	OSMR1	OS 計時器之符合暫存器 1 (OS timer match register 1)
0x40A0_0008	OSMR2	OS 計時器之符合暫存器 2 (OS timer match register 2)
0x40A0_000C	OSMR3	OS 計時器之符合暫存器 3 (OS timer match register 3)
0x40A0_0010	OSCR	OS 計時器之計數暫存器 (OS timer counter register)
0x40A0_0014	OSSR	OS 計時器之狀態暫存器 (OS timer status register)
0x40A0_0018	OWER	OS 計時器之看門狗致能暫存器 (OS timer watchdog enable register)
0x40A0_001C	OIER	OS 計時器之中斷致能暫存器 (OS timer interrupt enable register)

## 6.5 脈衝寬度調變器

使用脈衝寬度調變器(Pulse Width Modulator - PWM)可以讓應用處理器產生兩個輸出的信號。這個信號是以 3.6864 MHz 時脈為基礎，並且最小值必須為 2 個時脈週期寬。藉由設定 GPIO 讓應用處理器產生輸出信號。

### 6.5.1 脈衝寬度調變器之運作

應用處理器包含兩個脈衝寬度調變器：PWM0 和 PWM1。每個 PWM 個別獨立運作，被各自的暫存器所控制。他們可以在一個外部的腳位上提供一個脈衝寬度調變信號。因為每個 PWM 包含相同的電路，通稱為 PWM $n$ ， $n$  等於 0 或 1。每個 PWM 包含：

- 兩個脈衝寬度調變器之通道。
- 透過 6 位元時脈除法器器和 10 位元週期計數器來加強對週期的控制。
- 10 位元脈衝控制。

圖 6-3 為 PWM 的區塊圖。

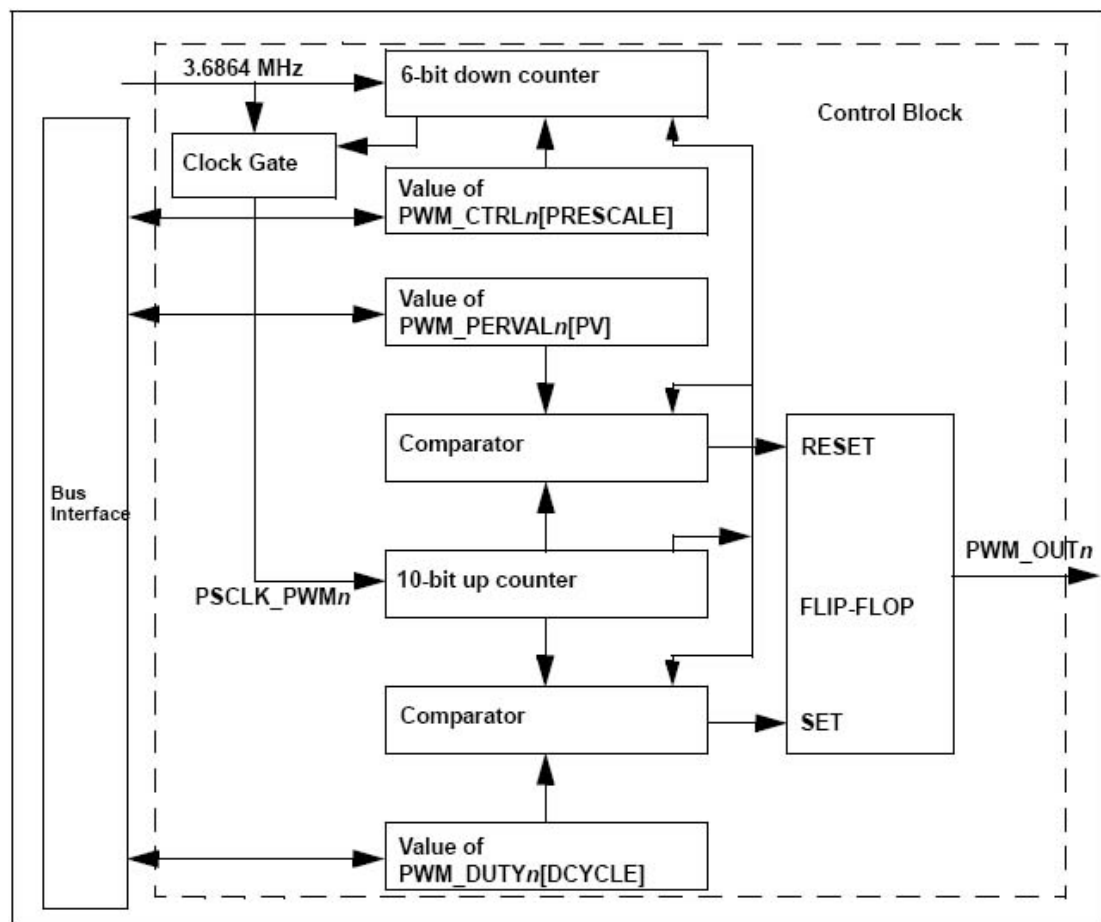


圖 6-3 PWM $n$  區塊圖

#### 6.5.1.1 相依性

PWM 單元由 3.6864 MHz 振盪器的輸出提供來源。  
每個脈衝寬度調變器單元(PWM $n$ )由三個暫存器控制：

脈衝寬度控制暫存器(Pulse Width Control Register - PWM\_CTRL)

工作週期控制暫存器(Duty Cycle Control Register - PWM\_DUTY)

週期控制暫存器(Period Control Register - PWM\_PERVAL)

藉由設定暫存器中的值，PWM $n$  單元產生一個脈衝寬度調變輸出信號。這些暫存器包含 PWM $n$  計數器的值及 PWM $n$  電源管理模式。

每個暫存器包含一個或多個欄位，這些欄位可以用來決定 PWM\_OUT $n$  波形的屬性。PWM\_CTRL $n$ [PRESCALE] 為 PWM 模組時脈的除數。注意，實際使用在 PWM 模組時脈的除數，比程式化在 PWM\_CTRL $n$ [PRESCALE] 中的值還大 1。這個被除過的 PWM 模組時脈會驅動一個 10 位元的上數計數器(up-counter)。這個上數計數器由兩個分開的比較器提供參數。第一個比較器包含 PWM\_DUTY $n$ [DCYCLE] 之值。當值相符時，PWM\_OUT 信號被設定為高電位(High)。另一個比較器包含 PWM\_PERVAL $n$ [PV] 之值，當 PWM\_PERVAL $n$ [PV] + 1 和 10 位元上數計數器相同時，清除 PWM\_OUT 信號為低電位(Low)。PWM\_PERVAL $n$ [PV] 和 PWM\_DUTY $n$ [DCYCLE] 都是 10 位元的欄位。

注意：注意並確認 PWM\_PERVAL $n$  暫存器中的值要比 PWM\_DUTY $n$  暫存器中的值還大。會有一個情況為，當 PWM\_PERVAL $n$  比 PWM\_DUTY $n$  還小時，輸出會一直維持在高電位(High)狀態。

#### 6.5.1.2 重置順序

系統重置時不會有脈衝寬度調變信號。在系統重置時，PWM\_CTRL $n$  和 PWM\_DUTY $n$  暫存器被重置為 0x0，PWM\_PERVAL $n$  被重置為 0x004。這樣會將 PWM\_OUT $n$  腳位以零工作週期來設定。6 位元的下數計數器被重置為 0x0，因此輸入的 3.68 MHz 時脈會直接驅動 10 位元上數計數器。PWM\_OUT $n$  腳位重置後會一直持續低電位(Low)，直到 PWM\_DUTY $n$  暫存器被設定一個非零的數值為止。

圖 6-4 說明一個基本的脈衝寬度波形。

#### 6.5.1.3 電源管理需求

透過每一對時脈致能位元可以用來每個 PWM (參考 3.6.2 節「時脈致能暫存器」)。如果時脈被關閉，這個單元就會被以下列兩種方法關閉：

魯莽—PWM 立即停止。

平順—PWM 在停止之前完成目前的工作週期。

透過選擇 PWM\_CTRL[PWM\_SD]可以關閉，請參考 6.5.2.1 節「PWM 控制暫存器(PWM\_CTRLn)」。

6.5.2 暫存器說明

以下的段落提供脈衝寬度調變器的暫存器說明。

6.5.2.1 PWM 控制暫存器(PWM\_CTRLn)

PWMn 控制暫存器：PWM\_CTRLn，包含兩個區域：

PRESCALE—PERSCALE 欄位包含一個 6 位元的 PRESCALE 計數器載入值。這個欄位允許 3.6864 MHz 輸入時脈到 PSCLK\_PWMn，該輸入時脈被除以 1(PWM\_CTL[PRESCALE]=0)到 64(PWM\_CTL[PRESCALE]=63)之間的值。

注意：除數的值要比程式化在 PRESCALE 欄位的值還大 1。

PWM\_SD—PWMn 可用兩種方法關閉：平順和魯莽，這是依照 PWM\_CTRLn[PWM\_SD]的設定來決定。如果選擇平順的方式，在工作週期計數器完成它的計數之後，才會將 PWMn 關閉。如果選擇了魯莽的方式，PWMn 會馬上關閉，prescale 計數器和工作週期計數器會重置，並且為這兩個暫存器載入它們相關的設定載入值。

注意：在魯莽的關閉時，PWM\_OUTn 信號可能會延遲最多一個 PSCLK\_PWMn 時脈週期。

表 6-49 說明 PWM 控制暫存器的位元地圖。

表 6-49 PWM_CTRLn 暫存器位元地圖																																	
Physical Address 0x40B0_0000 0x40C0_0000																PWM Control Registers (PWM_CTRL0, PWM_CTRL1)																SysInt	
Bit	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																								PWM_SD		PRESCALE						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
位元	名稱					描述																											
<31:9>	—					保留																											



<8>	PWM_SD	PWM 的關閉方式： 0=當 CKEN 暫存器中的時脈致能位元被清除時，優雅地關閉 PWMn。 1=當 CKEN 暫存器中的時脈致能位元被清除時，魯莽地關閉 PWMn。
<5:0>	PRESCALE	PWMn Prescale 除數 決定 PWM 模組時脈的頻率(3.86 MHz 時脈) $PSCLK\_PWMn = 3.6864\text{ MHz} / (PWM\_CTRL[PRESCALE] + 1)$

6.5.2.2 PWM工作週期暫存器(PWM\_DUTYn)

PWM工作週期暫存器：PWM\_DUTYn，包含兩個欄位：

- FDCYCLE
- DCYCLE

FDCYCLE 位元決定 PWM\_OUTn 是否為 PWM\_DUTYn 暫存器中的 DCYCLE 位元的功能，或是被設定為高電位。當 FDCYCLE 位元被清除為低電位時(一般運作)，PWM\_OUTn 輸出的波形為循環的，PWM\_OUTn 維持為高電位輸出，其輸出的期間的 PSCLK\_PWMn 週期數與 DCYCLE 相等。

如果 FDCYCLE=0x0，DCYCLE=0x0，PWM\_OUTn 被設定為低電位，其輸出不會有高低電位的變換(not toggle)。

注意：如果 FDCYCLE 為 0b1，PWM\_OUTn 在整個週期中都為高電位，並且不受到 DCYCLE 位元中的設定值的影響。

表 6-50 說明 PWM 工作暫存器的位元地圖。

表 6-50 PWM\_DUTYn 暫存器位元地圖

Physical Address 0x40B0_0004 0x40C0_0004										PWM Duty Cycle Registers (PWM_DUTY0, PWM_DUTY1)										SysInt																		
Bit	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																						FDCYCLE	DCYCLE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
位元	名稱																描述																					
<31:11>	—																保留																					

<10>	FDCYCLE	<p>PWMn 全工作週期</p> <p>0=PWM 時脈(PWM_OUTn)的工作週期是由 DCYCLE 來決定。</p> <p>1=PWM 被設定為高電位，其輸出不會有高低電位的變換(not toggle)。</p>
<9:0>	DCYCLE	<p>PWMn 工作週期</p> <p>PWMn 時脈的工作週期，例如 PWMn 被觸發後的一個 PWMn 的週期內， PSCLK_PWM 的週期數。</p>

### 6.5.2.3 PWM週期控制暫存器(PWM\_PERVALn)

PWM 週期控制暫存器(PWM\_PERVALn)內有一個 10 個位元的欄位名為 PV。就 PSCLK\_PWMn 時脈而言，這個欄位決定了 PWM\_OUTn 週期的波形。如果這個欄位被清除為 0，PWMn 會有效地關閉而且 PWM\_OUTn 會維持在高電位的狀態。對於任何非零的值寫入到 PV 欄位時，PWMn 的輸出頻率為 PSCLK\_OUTn 的頻率除以(PV+1)的值。時脈輸出的範圍從直通的 PSCLK\_PWMn，變成每個輸出脈衝會延遲  $2^6$  或 64 個輸入時脈。

當10位元上數計數器的值與(PV+1)相同時，上數計數器和正反器被重置，而且PWM\_CTRLn, PWM\_PERVALn 和PWM\_DUTYn 的值被載入到這些暫存器的內部版本中。重置正反器會導致PWM\_OUTn輸出為低電位，而且PWM週期會再度開始。

除非FDCYCLE=0x0、DCYCLE=0x0，否則寫入所有的0到暫存器中會導致輸出維持在高電位狀態。如果FDCYCLE=0x0、DCYCLE=0x0，不論PV位元欄位的值為何，輸出都會維持低電位狀態。

注意：由於內部時序的需要，所有對PWM暫存器的變動都必須在PWM時脈週期終止開始前的4個核心時脈週期內完成，這是為了保證接下來的PWM週期會執行新的值。

表6-51說明PWM週期控制暫存器的位元地圖。

**表 6-51 PWM\_PERVALn 暫存器位元地圖**

Physical Address		PWM Period Control Registers										SysInt																										
0x40B0_0008		(PWM_PERVAL0, PWM_PERVAL1)																																				
0x40C0_0008																																						
Bit	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																						PV															
Reset	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	0	0						

位元	名稱	描述
<31:10>	—	保留
<9:0>	PV	<p>PWMn 週期控制：</p> <p>組成一個 PWM_OUTn 週期的 PSCLK_PWMn 週期數量。</p> <p>注意：如果 PV=0x0，PWMn 時脈(PWM_OUTn)被設定為高電位，其輸出不會有高低電位的變換(not toggle)，除非 FDCYCLE=0x0 以及 DCYCLE=0x0。在這個例子中無論 PV 中的值為何，PWM_OUTn 被設定為低電位。</p>

6.5.3 脈衝寬度調變器之輸出波形範例

圖6-4為提供參考的一個脈衝寬度調變器輸出的範例。

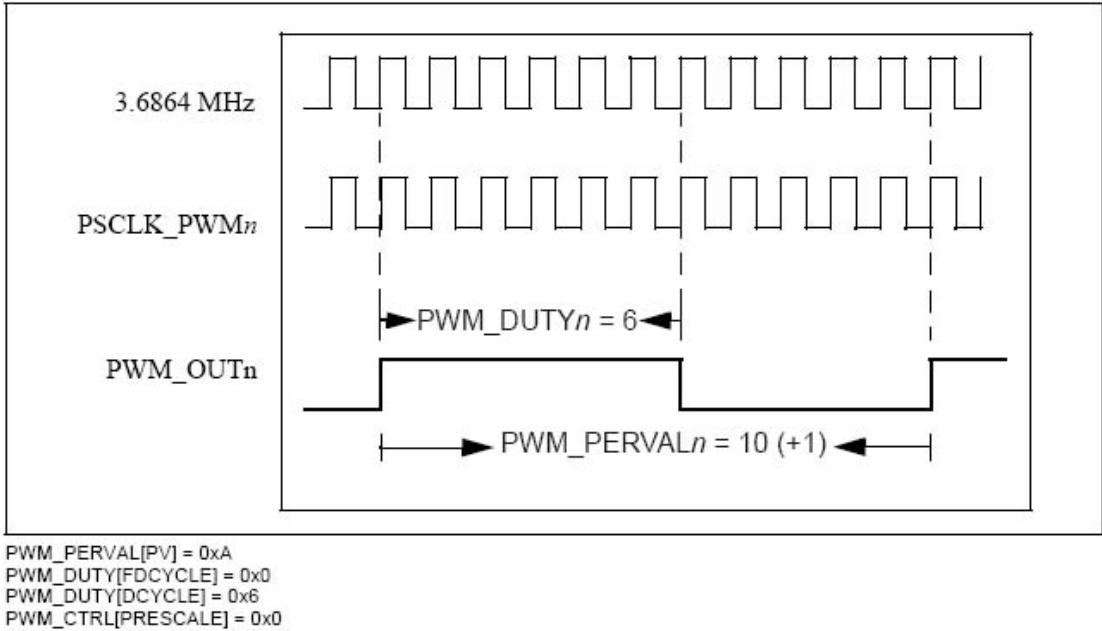


圖6-4 基本脈衝寬度波形

圖 6- 4 是由寫入 PWM\_PERVALn[PV]十進位的 10(11 個時脈)和 PWM\_DUTYn[DCYCLE]為 6 產生的輸出波形圖。圖 6-4 也說明

PWM\_CTRLn[PRESCALE]被載入 0x0，導致 PSCLK\_PWMn 與 3.6864 MHz 輸入時脈有相同的頻率。

## 6.5.4 暫存器總結

表 6-52 說明與 OS 計時器相關的暫存器以及用來存取的實體位址。

表 6-52 PWM 暫存器位置

位址	名稱	解釋
0x40B0_0000	PWM_CTRL0	PWM0 控制暫存器 (PWM0 Control Register)
0x40B0_0004	PWM_PWDUTY0	PWM0 工作週期暫存器 (PWM0 Duty Cycle Register)
0x40B0_0008	PWM_PERVAL0	PWM0 週期控制暫存器 (PWM0 Period Control Register)
0x40C0_0000	PWM_CTRL1	PWM1 控制暫存器 (PWM1 Control Register)
0x40C0_0004	PWM_PWDUTY1	PWM1 工作週期暫存器 (PWM1 Duty Cycle Register)
0x40C0_0008	PWM_PERVAL1	PWM1 週期控制暫存器 (PWM1Period Control Register)

### 問題：

1. 系統整合單元包含哪些單元？
2. 請問在 Intel XScale 應用處理器中的 GPIO 有什麼樣的特性？
3. 請問在 Intel XScale 應用處理器中，如何設定 GPIO 的腳位為輸入或輸出？若為輸出，則須如何設定其腳位為高電位或低電位？若為輸入，則如何得知此腳位的狀態？
4. 請問在 Intel XScale 應用處理器中的 GPIO 交替功能，其功用為何？
5. 請問如何設定 Intel XScale 應用處理器中的 GPIO 交替功能？
6. 請問在 Intel XScale 應用處理器中有哪兩種中斷？其作用為何？其優先權為何？
7. 請問在 Intel XScale 應用處理器中，如何遮蔽中斷？該如何設定？
8. 請問即時時脈有何用途？
9. 一般的處理器可以利用即時時脈來輸出 Hz 訊號，意即每秒鐘一個時脈。請

問在 Intel XScale 應用處理器中，若 Hz 不準確實，該如何修正？

10. 請問有哪些因素可能影響 Hz 的準確度？
11. 請問作業系統佔存器有哪些用途？
12. 請問看門狗重置的運作原理為何？該如何設定？
13. 請問脈衝寬度調變器的功用為何？