# 笔记本嵌入式控制器(EC)

## 关于IT5576E特性

### 内置 8051 嵌入式控制器

* eSPI接口
* LPC 总线接口
* Flash 接口
* SMBus 控制器
* 系统唤醒控制
* EC唤醒控制(外部唤醒事件，引脚)
* 中断控制
* 3个8051内部定时器，2个外置8位定时器和2个外置24位定时器在ETWD模块，基于32.768 k时钟源；一个外部16位看门狗定时器。
* 串口功能
* 5个电源管理通道
* BRAM
* 82个GPIO 口
* 外部GPIO控制
* 键盘接口（KBC）
* 11个10bit ADC 通道（8个外部通道）
* 4个8bit DAC通道
* 8个PWM通道
* PECI(平台环境控制接口)
* PS/2 接口
* 矩阵键盘扫描功能
* SPI从机接口(SSPI)
* 系统内部编程
* 红外线装置
* 能耗管理
* 消费电子控制通道 Consumer Electronics Control (CEC) 适用于HDMI 1.4 标准
* RTC控制器

## 通用描述

The IT5576 is a highly integrated embedded controller with system functions suitable for mobile system applications. The IT5576 directly interfaces to the LPC bus and provides ACPI embedded controller function, keyboard controller (KBC) and matrix scan, PWM and ADC for hardware monitor, PS/2 interface for external keyboard/mouse devices, BRAM, CIR and system wake-up functions for system power management

IT5576是一款高度集成的嵌入式控制器，具有适用于移动系统应用程序的系统功能。 IT5576直接连接到LPC总线，并提供ACPI嵌入式控制器功能，键盘控制器（KBC）和矩阵扫描，用于硬件监视器的PWM和ADC，用于外部键盘/鼠标设备的PS / 2接口，BRAM，CIR和系统唤醒系统电源管理功能

# 专业缩写词

|  |  |  |
| --- | --- | --- |
| **缩写** | **全拼** | **备注** |
| LPC Bus | Low Pin Count Interface | 少针脚型接口 |
| ACPI | Advanced Configuration And Power Management | 高级配置和电源管理接口 |
| KBC | Keyboard Controller | 键盘控制 |
| CIR | Consumer IR | 类似红外的消费类控制接口 |
| EC | 8051 Embedded controller | 8051 控制器 |
| SMFI | Shared Memory Flash Interface Bridge | 共享内存接口 |
| eSPI | The Enhanced serial Perpheral Interface | 增强型SPI |
| PNPCFG | Plug and Play Configuration | 即插即用配置 |
| i-Bus | Internal Bus | 内部总线 |
| EC2I | EC Access to the Host Controlled Modules | EC 访问主机控制模块 |
| SWUC | System Wake-Up Control | 系统唤醒控制器 |
| PMC | Power Management Channel | 电源管理通道 |
| SSPI | Serial Peripheral Interface |  |
| PECI | Platform Environment Control Interface | 环境控制接口 |
| SMB | SMBus Interface | 系统管理总线 |
| WUC | Wake-Up Control | 唤醒控制 |
| ECPM | EC Clock and Power Management Controller | EC时钟控制器和电源管理控制器 |
| ETWD | External Timer and External Watchdog | 外部时钟和看门狗时钟 |
| CEC | Consumer Electronics Control | 消费类电子控制器 |
| TMR | Timer | 定时器 |
| EC Bus | Embedded controller Bus | EC 控制接口 |
| PP | Parallel Port | 并行端口 |
| DBGR | Debugger | 调试器 |
| RTCT | RTC-like Timer | RTC 定时器 |
|  |  |  |

# EC的工作原理

# 引脚功能定义

# 系统框图

## 系统框图

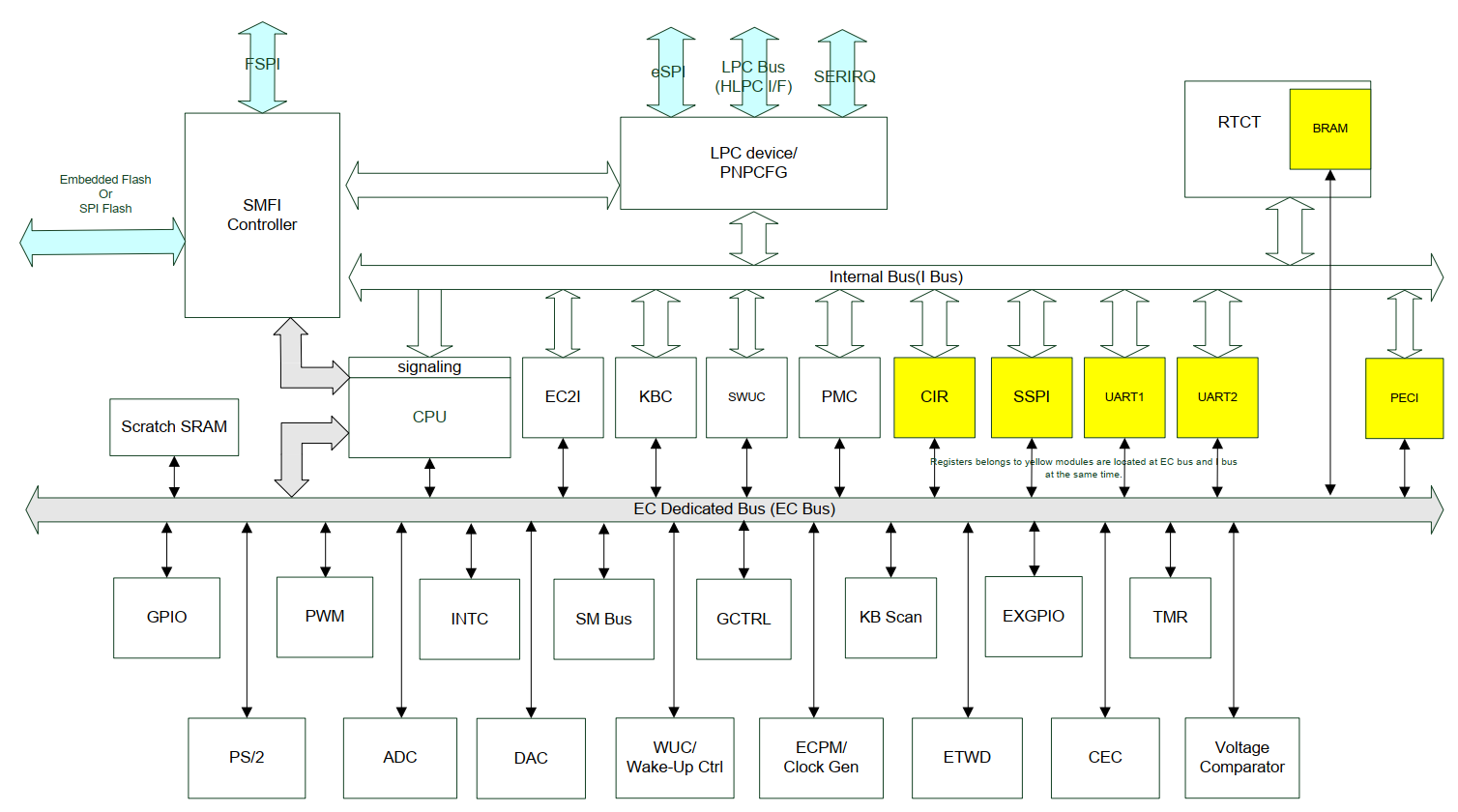


图3‑1：系统框图

## EC映射存储空间

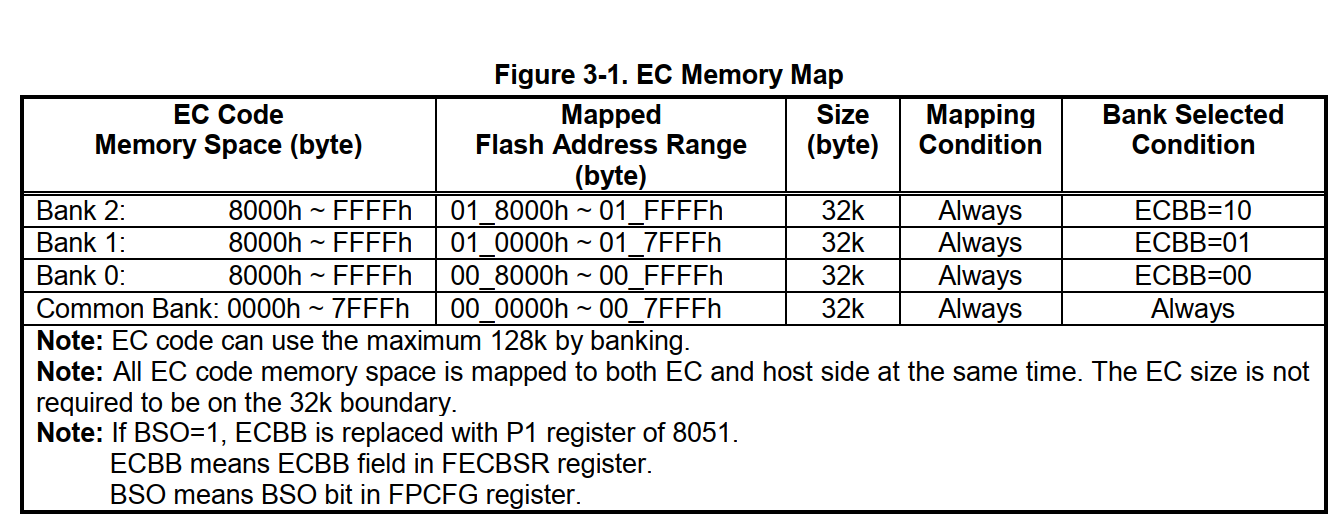


图3‑2：内存存储空间

## EC 内部数据存储空间

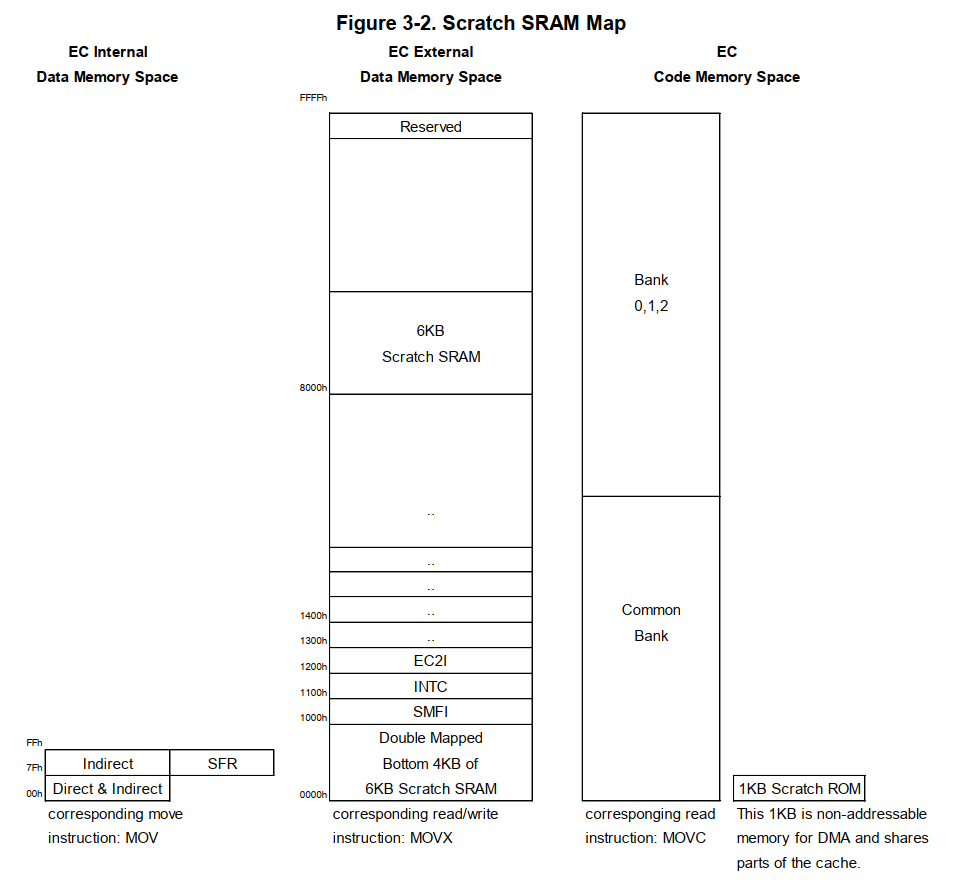


图3‑3：数据存储空间

## 电源状态

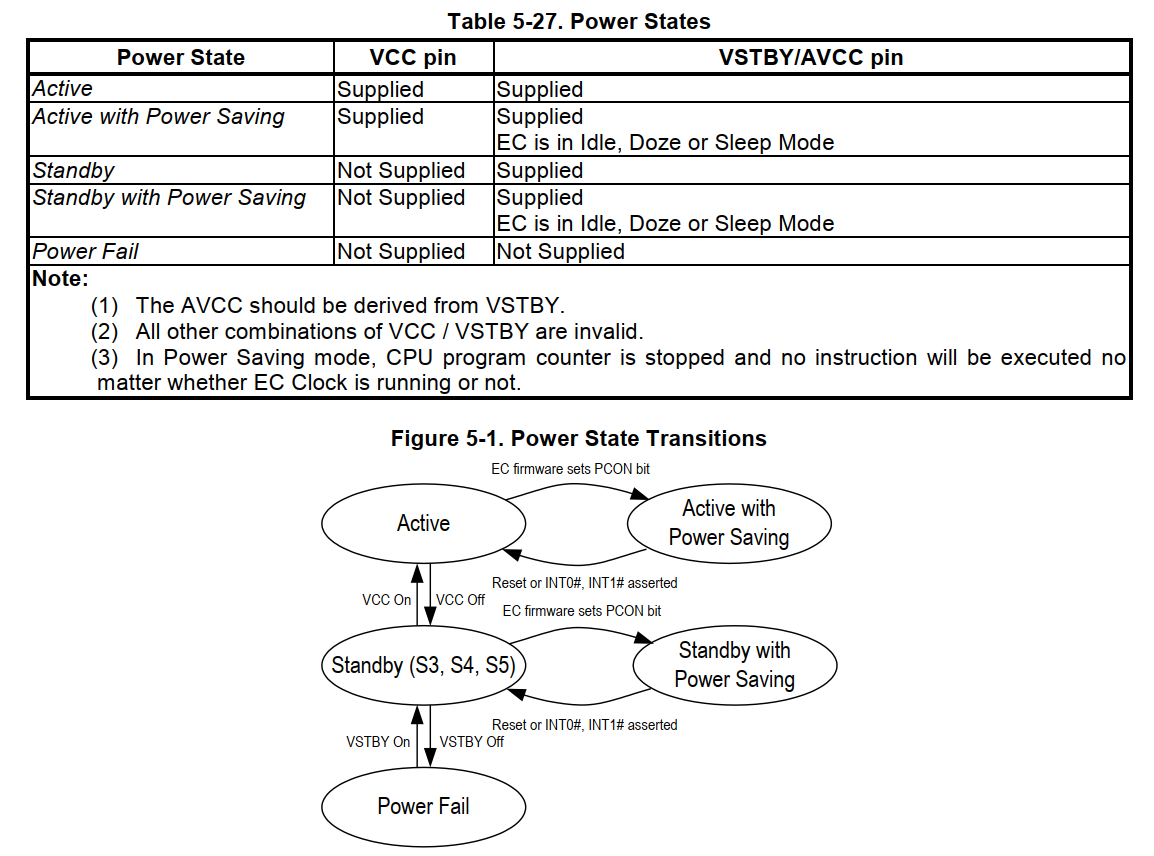


图3‑4：电源管理状态

# Host Domain Functions

## eSPI

实际上eSPI是用于取代的LPC接口，ESPI总线借鉴和复用了SPI电气时钟规范，但是在协议层用了全新的定义，所以eSPI无论从功能还是应用于SPI完全是两回事。eSPI 在同一物理总线上传递了四大信息流，为了保证四类信息流之间相互独立互不干扰，eSPI 引入了channel 的概念。

在协议层面eSPI为每个channel定义了自己专用的指令和payload;在硬件层面，每一个channel 都有属于自己的data buffer。

### eSPI外设通讯通道

eSPI拥有外设通讯通道，在eSPI总线上能收发消息，具体流程如下:

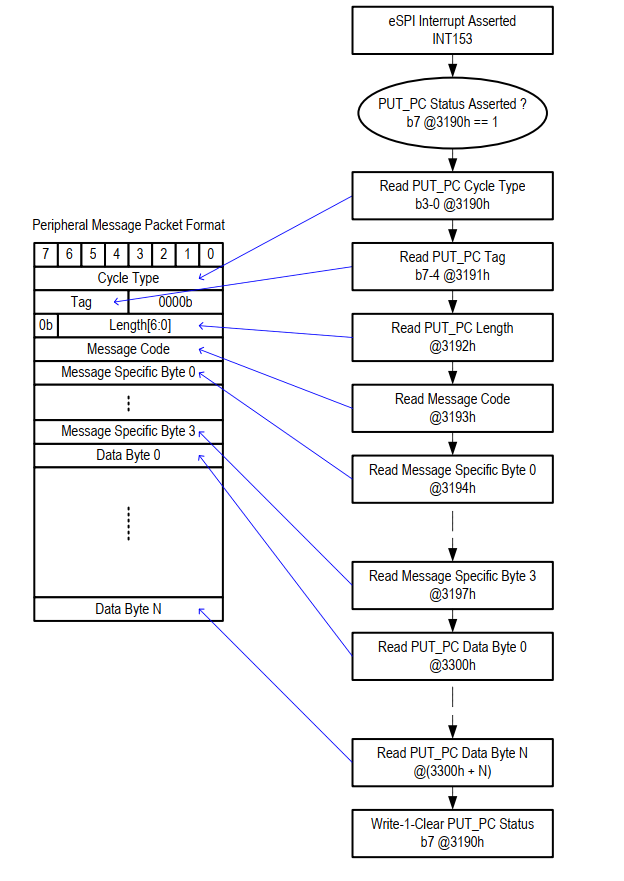


图5‑1：eSPI总线上接收外设消息流程

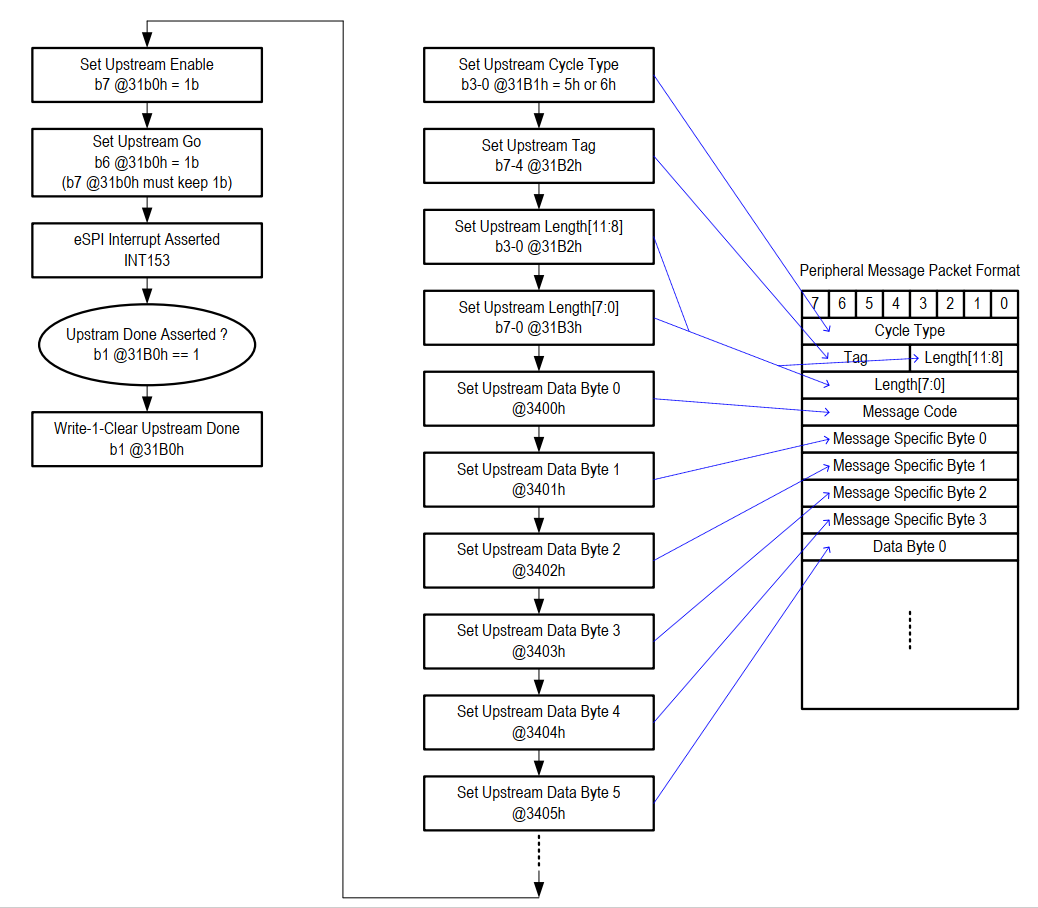


图5‑2：eSPI总线上初始化消息

### eSPI flash访问通道

#### MAFS(Master-Attached Flash access)

eSPI flash 共享空间中与flash中的代码切换流程

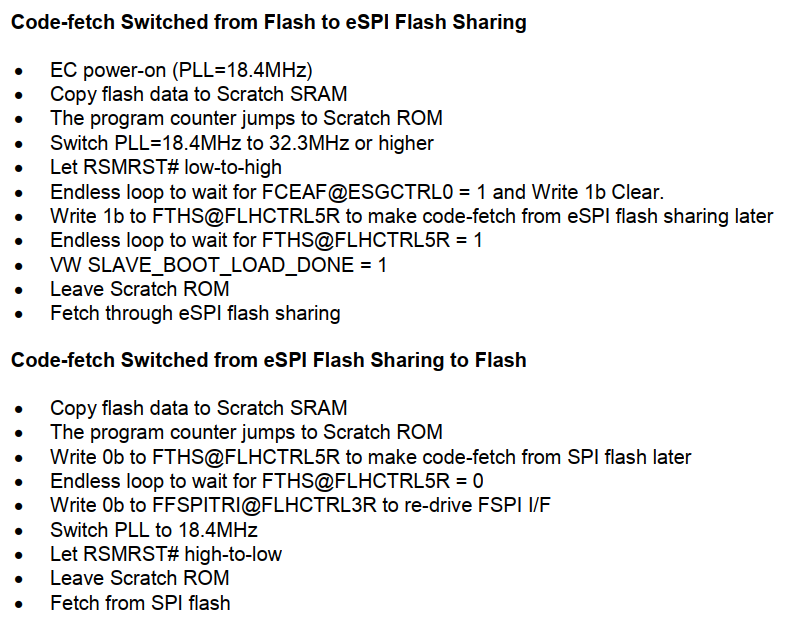


图5‑3：代码切换流程

EC 能通过EC总线访问共享的flsah 组件，流程如下所示：

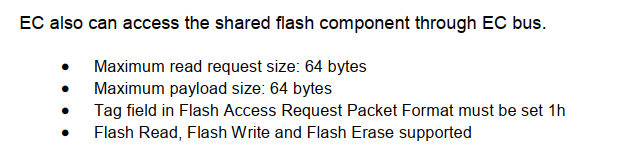


图5‑4：EC访问共享flash配置

sSPI读、写以及擦除flash 操作如下所示：

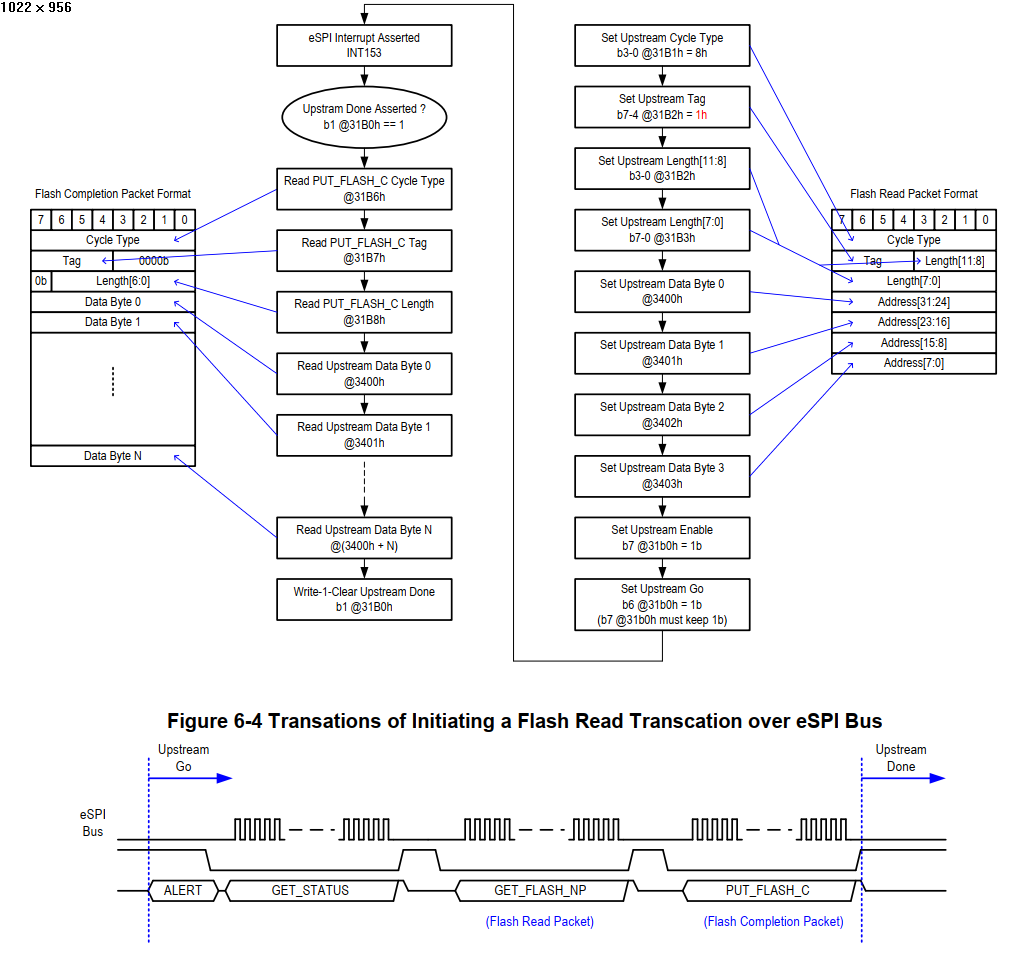


图5‑5：执行读操作

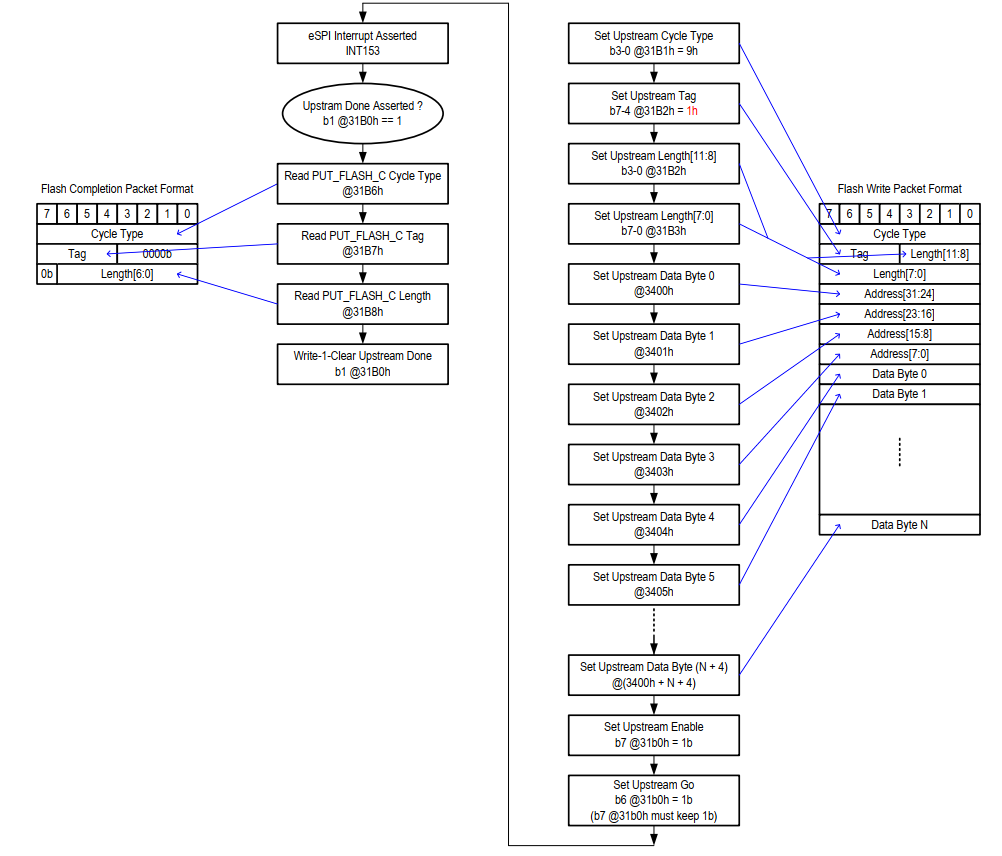


图5‑6：执行写操作

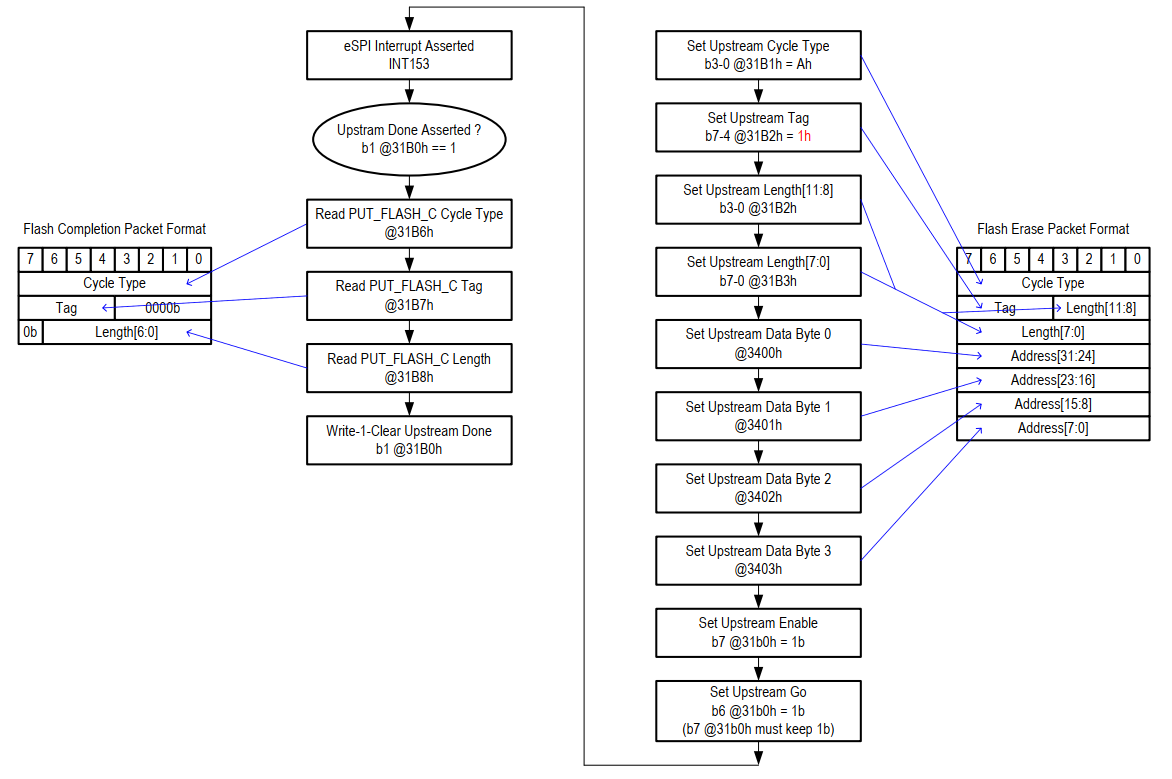


图5‑7：执行擦除flash操作

#### SAFS

SAFS相对于MAFS就是从机，所以他获取的是主机的读写指令操作。操作列表如下所示：

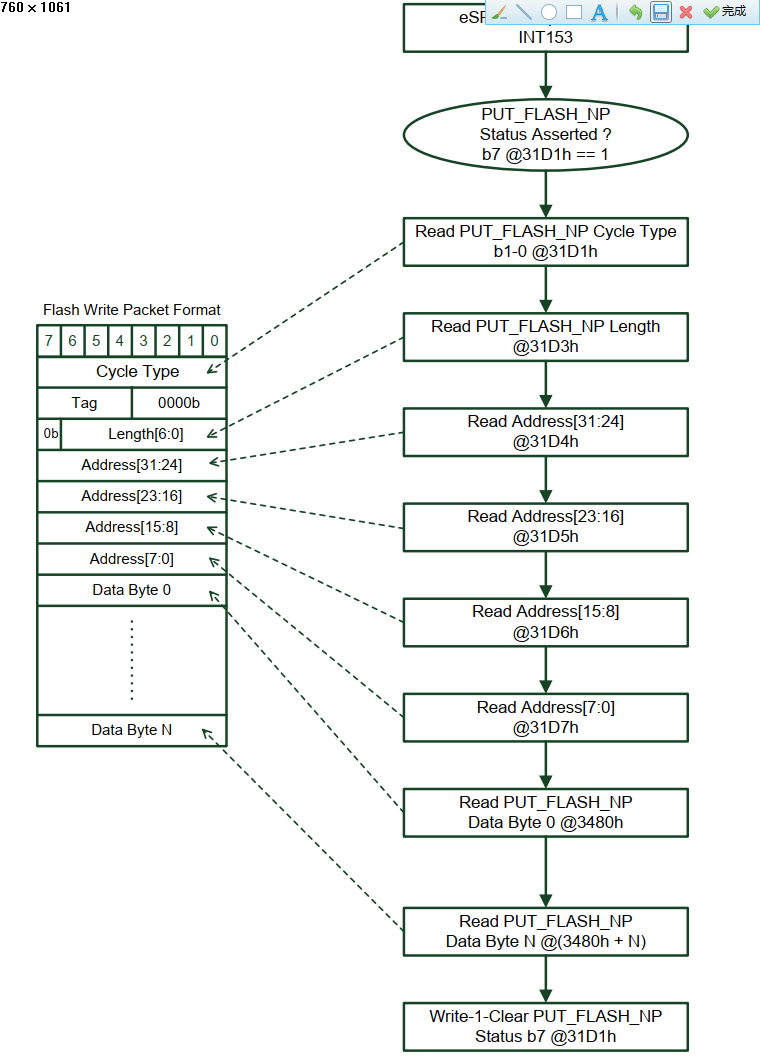


图5‑8：获取flash写操作

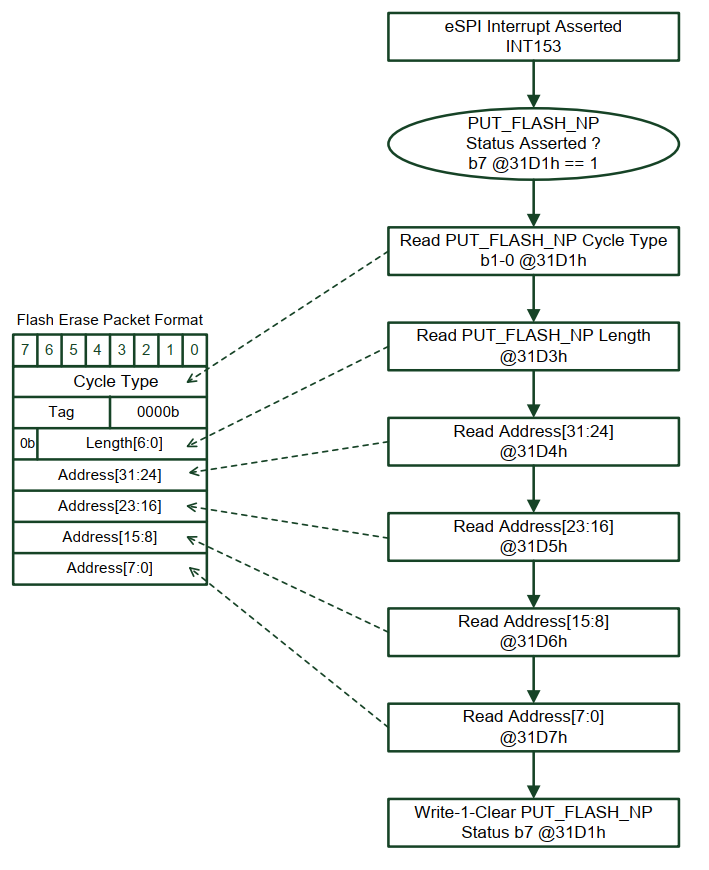


图5‑9：获取flash擦除操作

### OOB Message Channel(out of band)

OOB通道用于处理OOB处理器和EC之间的事务。EC能够启动上游OOB消息事务，使用消息预定义的从地址和命令代码读取SKL-PCH HW信息，包括温度和RTC时间/日期。

关于OOB 信息通道的消息接收传输机制如下所示：

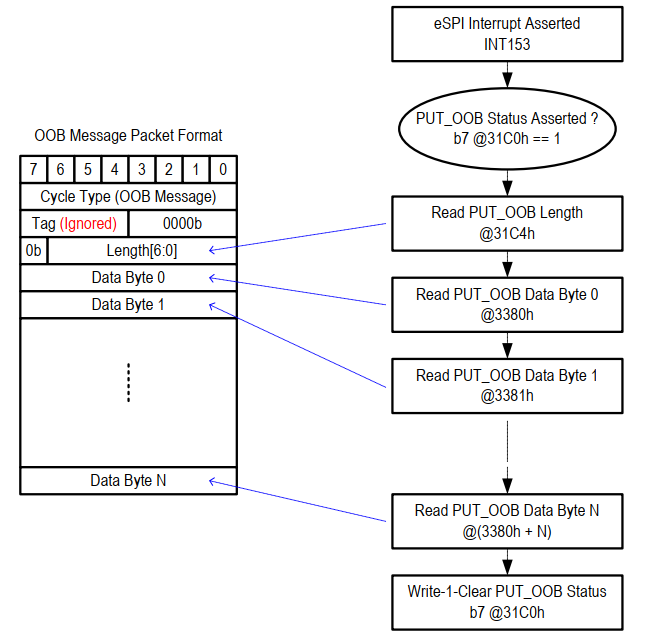


图5‑10：通过总线接收OOB的消息事件

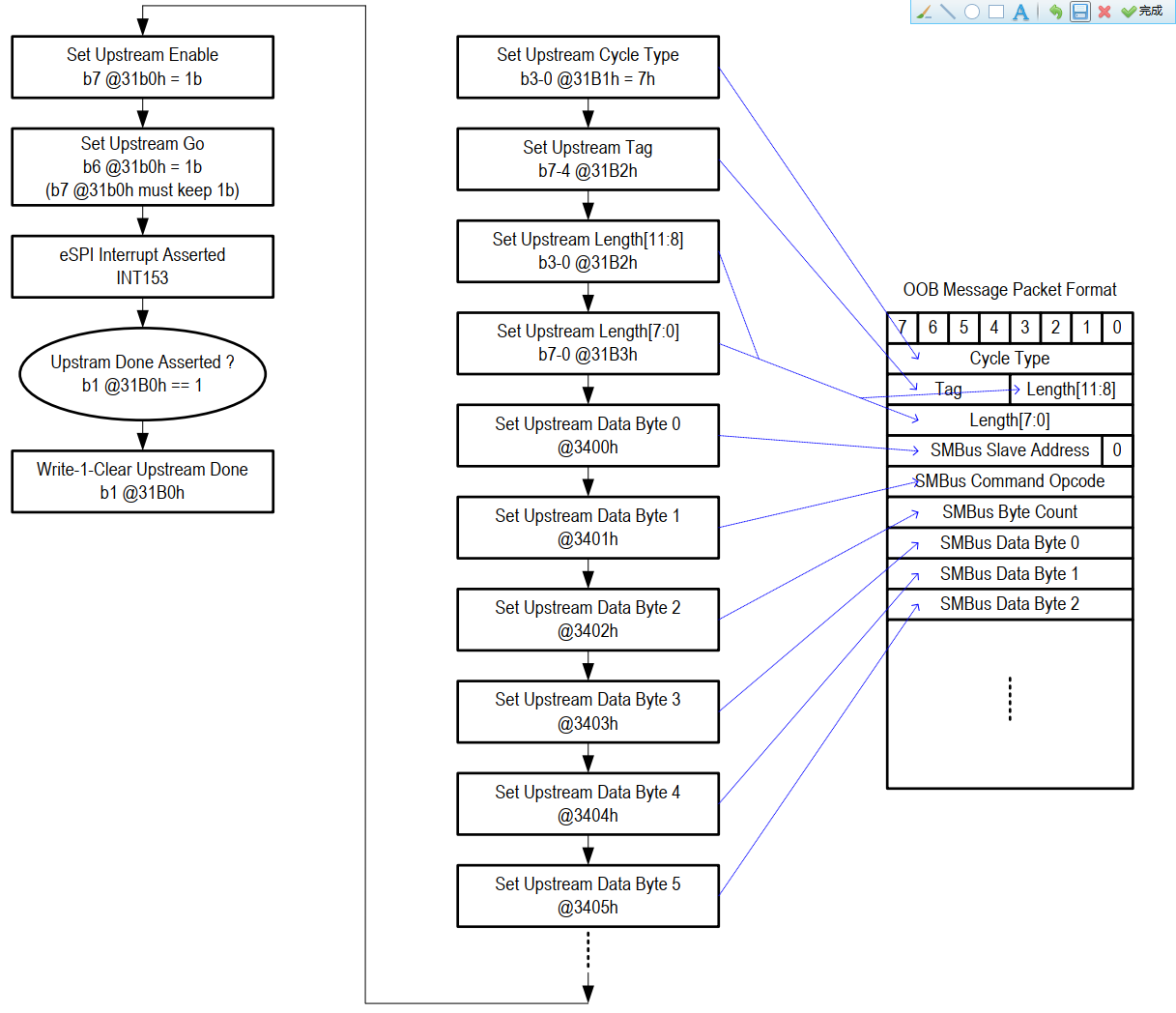


图5‑11：启动OOB的消息事件

### 虚拟接口通道(virtual wire channel)

支持7个最大虚拟接口，支持IRQ0~IRQ15的中断事件等。虚拟接口接收带内数据。以设备接收virtual wire index 2 带内数据为例，过程如下所示：

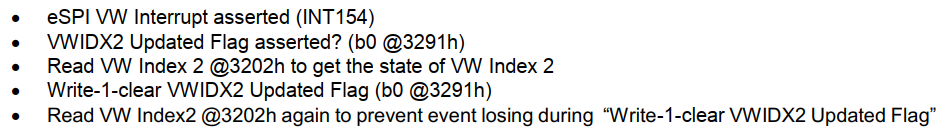


图5‑12：通过eSPI虚拟线通道接收带内消息

以EC能够发起一个在带内消息通过eSPI虚拟电线通道。下面列出了启动VW Index 4带内消息的两种方法。

Method 1: Write VW Index 4 @3204h to change the state of VW Index 4

If the state of VW Index 4 is different from that sent during the previous transaction, EC initiates an in-band message over the eSPI bus for updating VW Index 4. If not, EC does not initiate an in-band message.

**方法1:写入VW Index 4 @3204h，改变VW Index 4的状态**

如果VM index 4的状态与前一次传输期间发送的不同，EC通过eSPI总线启动带内消息以更新VM index 4。如果不是，EC不发起一个带内消息。

Method 2: Write 1 to VW Index 4 Resend @3293h

Writing 1 to VW Index 4 Resend forces EC to resend the state of VW Index 4 regardless of that sent during the previous transaction. VW Index 4 Resend is automatically cleared after the in-band message is sent through the eSPI Virtual Wires channel

**方法2:将1写入到VW index 4 Resend @3293h**

写1到VM index 4强制重发VW index 4的状态，无论在之前的传输中发送的是什么。VW index 4重发后自动清除带内消息发送通过eSPI虚拟接口通道。

## LPC接口

关于LPC 周期访问类型：

The supported LPC cycle types are listed below:

\* LPC I/O Read(16-bit address, 8-bit data)

\* LPC I/O Write (16-bit address, 8-bit data)

\* LPC Memory Read(32-bit address, 8-bit data)

\* LPC Memory Write(32-bit address, 8-bit data)

\* FWH Read (32-bit address, 8-bit data)

\* FWH Write (32-bit address, 8-bit data)

如果EC在启动时LPC内存或I/O周期中失败，请首先检查以下建议项

•LPC/FWH memory cycles

•Check whether LPCRST# reset source from GPD2 or GPB7 is logic low if it is in alternative

function.

Check whether LPCPD# signal from GPE6 is logic low if it is in alternative function.

Check whether HBREN bit is enabled in HCTRL2R register.

Check whether the firmware doesn’t change the read protection control.

•LPC I/O cycles

Check whether LPCRST# reset source from GPD2 or GPB7 is logic low if it is in alternative function.

Check whether LPCPD# signal from GPE6 is logic low if it is in alternative function.

Check whether BADDR1-0 field in BADRSEL register are in correct setting.

Check whether EC2I is not locking PNPCFG access from the host side

## 即插即用配置接口(PNPCFG)[待完善它的寄存器配置]

要访问PNPCFG寄存器，请将目标索引写入地址端口，然后通过数据端口访问该PNPCFG寄存器。

### 逻辑设备号(page 75)

|  |  |
| --- | --- |
| **LDN** | **Functional Block** |
| 01h | Serial Port 1 (UART1) |
| 02h | Serial Port 2 (UART2) |
| 04h | System Wake-Up Control (SWUC) |
| 05h | KBC/Mouse Interface |
| 06h | KBC/Keyboard Interface |
| 0Ah | Consumer IR (CIR) |
| 0Fh | Shared Memory/Flash Interface (SMFI) |
| 10h | RTC-like Timer (RTCT) |
| 11h | Power Management I/F Channel 1 (PMC1) |
| 12h | Power Management I/F 2 (PMC2) |
| 13h | Serial Peripheral Interface (SSPI) |
| 14h | Platform Environment Control Interface (PECI) |
| 17h | Power Management I/F Channel 3 (PMC3) |
| 18h | Power Management I/F Channel 4 (PMC4) |
| 19h | Power Management I/F Channel 5 (PMC5) |

下图指示PNPCFG寄存器与Super I / O配置寄存器和逻辑设备配置寄存器结合在一起。 仅当逻辑设备号寄存器中填入第75页的表6-10中列出的相应逻辑设备号时，才能访问指定逻辑设备的逻辑设备配置寄存器。

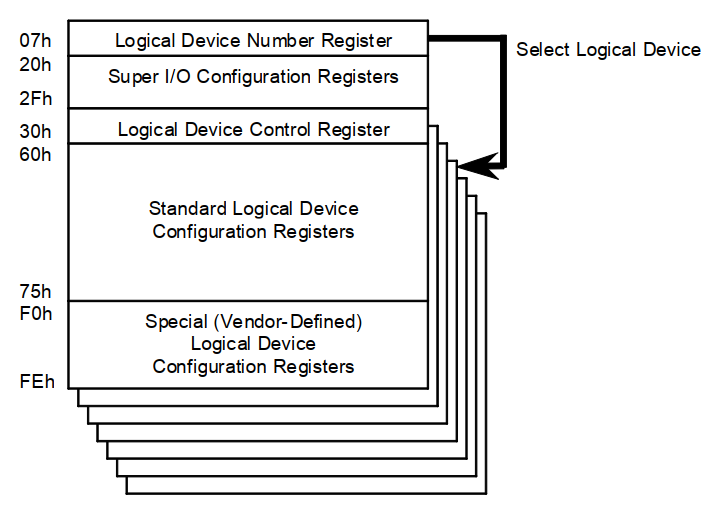


图5‑13：映射表（page 76）

注意，在操作这个逻辑设备配置寄存器功能之前，必须先写入带配置的逻辑设备号。该功能似乎是通过逻辑设备号（index = 07H）索引配置信息。

## 共享内存接口(SMFI)

功能：

The bridge provides the host to access the shared memory. It also provides EC code address space mapped into the host domain address space, and locking mechanism for read/write protection

桥接器提供主机访问共享内存的功能。 它还提供映射到主机域地址空间的EC代码地址空间，以及用于读/写保护的锁定机制

### 通过HLPC 总线进行flash编程

When programming flash via HLPC Follow Mode 0 is processing, the flash will be busy and code fetch from flash by CPU and will be invalid and cause CPU fail to execute instructions. It means the firmware must copy necessary instructions from code space to Scratch SRAM, enable mapping Scratch SRAM to Scratch ROM, and jump to Scratch ROM before programming flash.

当正在通过HLPC跟随模式0对flash进行编程时，flash将处于繁忙状态，CPU将从flash中获取代码，并且该flash将无效并导致CPU无法执行指令。 这意味着固件必须将必要的指令从代码空间复制到Scratch SRAM，将Scratch SRAM映射到Scratch ROM，并在编程flash之前跳转到Scratch ROM。

步骤：

Flash Programming Steps:

(a) The host side communicates with the EC side via KBC/PMC extended or semaphore registers

(b) EC side: Write 1 to HOSTWA bit in SMECCS register

(c) EC side: Copy necessary code to Scratch RAM

(d) EC side: Enable code space mapping of Scratch SRAM

(e) EC side: Make the host processor enter SMM mode if necessary

(f) EC side: Jump instruction to Scratch ROM

(g) Host side: Set related memory-write registers in South-Bridge

(h) Host side: Start flash programming

(i) End flash programming and reset EC domain if necessary

(Refer to section 5.4 on page 25)  
**Note:** Do not let EC enter Idle/Doze/Deep Doze/Sleep mode while processing flash programming flow

### 串行flash编程

对于串行flash编程有两种模式。其中模式0内部没有硬件连接机制来监视flash 的WIP状态；反之，模式1中存在硬件连接机制来监视flash 状态。然而无论是模式0还是1，仍然需要主机端以轮循的方式判断flash 状态。

关于模式0 与模式1 的flash 编程在Page 116

### EC 与host 的信号交互

针对H2RAM主机信号量的LPC存储器/ FWH / IO写周期将产生一个中断（INT83）通知EC，并且在H2RAM EC信号量上执行的数据写入指令还会产生对主机的IRQ。 通过这些信号量，主机和EC可以在H2RAM数据传输期间相互通信

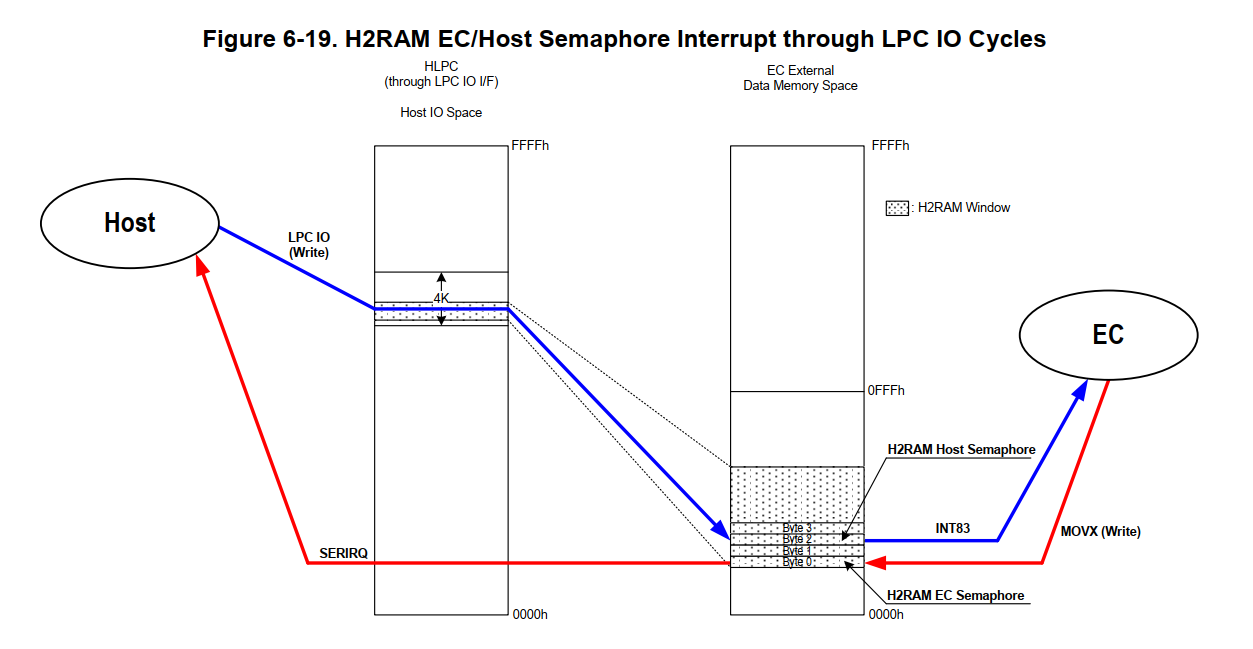


图5‑14：LPC IO 通讯周期

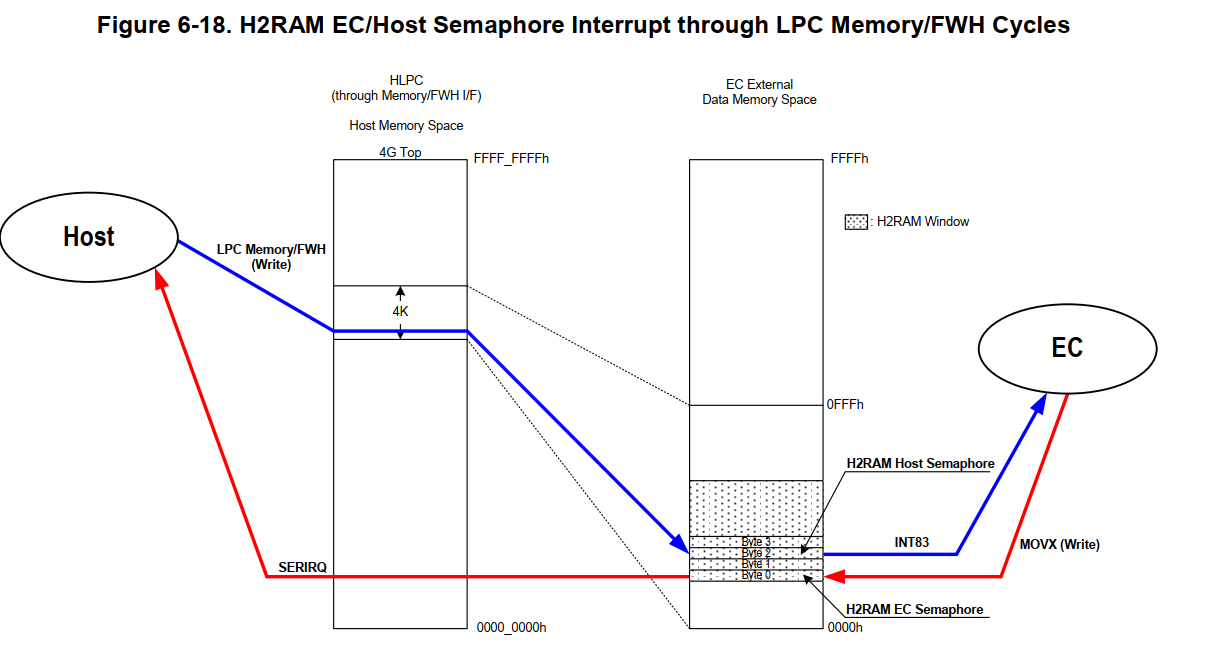


图5‑15： LPC menory/FWH 周期

## 系统唤醒控制(SWUC)【待完善】

## 键盘控制器(KBC) 【待完善】

## 电源管理通道(PMC)

## RTC

支持计时和日历管理，直到关闭VSTBY。

支持two-band SRAM。

时间警报从每秒一次到每月一次不等。

提供四种中断功能：定期中断，警报1中断，警报2中断和更新结束中断。

支持BCD或二进制格式表示时间。

支持夏令时补偿功能。

支持自动闰年补偿功能。

锁存LPC I / O端口80h将数据写入RTCT bank 1的SRAM（P80L功能）

RTCT包含每个存储区的SRAM（存储区0的128B；存储区1的64B；总共192B SRAM），并提供计时和日历功能，直到VSTBY关闭。

RTCT具有两个SRAM库，分别是Bank 0 SRAM和Bank 1 SRAM。

所有RTCT时间，警报数据和控制寄存器都可以通过RTCT索引和数据寄存器进行访问。 索引寄存器指向要访问的寄存器位置，数据寄存器包含要从该位置传送或从该位置传送的数据。 可通过第一对索引和数据寄存器（传统索引70h，71h）访问Bank 0 SRAM。 可通过第二对索引和数据寄存器（传统索引272h，273h）访问Bank 1 SRAM。 Bank 0 SRAM的前14个字节和两个可编程字节是RTCT时间，警报1数据和控制寄存器。 Bank 1 SRAM的前5个字节是RTCT警报2数据寄存器。 对RTCT SRAM的访问可能会被锁定。

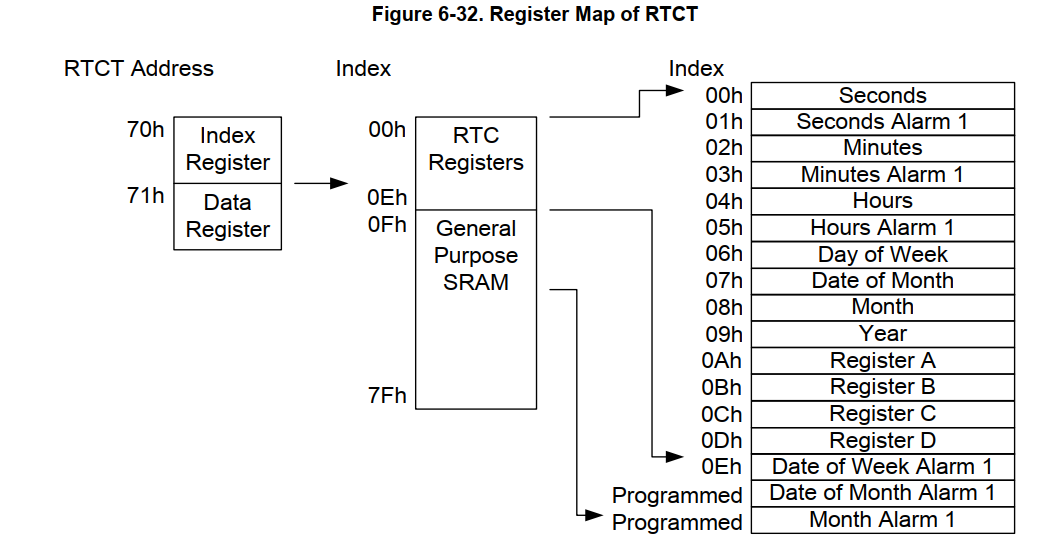


图5‑16： RTC band 0 寄存器地址映射

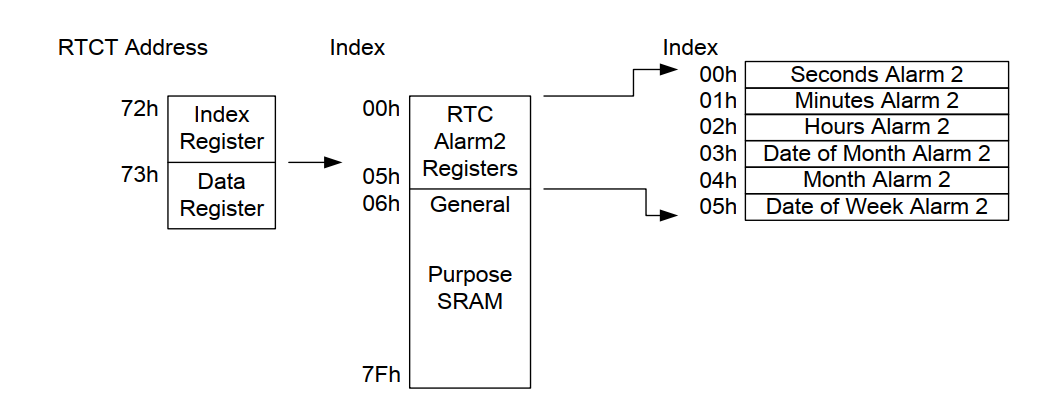


图5‑17： RTC band 1 寄存器地址映射

## 消费性红外端口(CIR)

## SSPI

## 平台环境控制接口(PECI) 未完成

## 串口 完成

两个串口，波特率高达115200，如果选择了告诉波特率选择（HHS）或EC高速选择（ECHS）,则最高波特率可以高达230.4K和460.8K，这个取决于波特率发生器。

与

# EC Domain Functions

## 8051 控制器(EC) 完成

支持睡眠（也称为掉电）和空闲模式

支持INT1＃外部中断

支持64K代码/数据空间和代码存储

支持256字节内部RAM

支持定时器0/1/2

支持全双工UART

## 中断控制 完成

数字较高，中断优先级最高

当检测到外部中断时,CPU会从sleep 模式和瞌睡模式中唤醒。

通常，来自WUC的中断是高电平触发的。 Page209页7.2.3.1有介绍

详细的中断清单在page217。

## 唤醒操作（WUC） 完成

WUC的输入源是外部输入，例如关于PS / 2，GPIO和KB矩阵扫描的引脚，或者是来自内部总线的输入，例如处理外部输入的SWUC，LPC和SMBus。

## 矩阵键盘控制 完成

Page 265 GPIO 要求最少 24 个GPIO（16 \* 8）, 要求能控制这24个GPIO,每个引脚功能可选择为按键扫描模式或者GPIO模式；16 \*8 合计构成128个按键控制。Page233

## GPIO 端口 完成

I / O引脚分别配置为输入，输出或备用功能

支持带有串行闪存的82端口GPIO

可配置的内部上拉电阻

可配置的内部下拉电阻

## EC 时钟和电源管理（ECPM） 完成

与时钟相关 ，略；

## SMBus(SMB) 待完成

## 平台环境控制接口(PECI) 找不到资料

## PS/2 接口 完成

The PS/2 device uses a two-wire bi-directional interface for data transmission. The device consists of three identical channels. Each of the three channels provides two signals (CLK and DATA line) to communicate with the auxiliary device. The PS/2 interface also connects the CLK line and DATA line to the WUC (WU10-WU17) to wake-up the CPU when these lines are toggled. CLK line and DATA line are the same as PS2CLKn and PS2DATn (n=0 or 2) pins. Refer to Table 5-9 on page 15 for the details

PS / 2设备使用两线双向接口进行数据传输。 该设备包含三个相同的通道。 三个通道中的每个通道都提供两个信号（CLK和DATA线）以与辅助设备进行通信。 PS / 2接口还将CLK线和DATA线连接到WUC（WU10-WU17），以在切换这些线时唤醒CPU。 CLK线和DATA线与PS2CLKn和PS2DATn（n = 0或2）引脚相同。 有关详细信息，请参见第15页的表5-9。

The PS/2 Interface has two operation methods: Hardware mode and software mode.

PS / 2接口有两种操作方法：硬件模式和软件模式。

### 硬件模式选择

接收方式

主机从PS / 2设备接收数据的步骤如下。

1.启用硬件模式，选择接收模式，然后释放CLK线和DATA线（将07h写入PS / 2控制寄存器）。

2.启用中断。（PS / 2中断控制寄存器中的TDIE位必须设置为1，因为完成数据传输后，需要读取PS / 2数据寄存器中的数据。）

这些步骤之后，PS / 2接口已准备好接收数据。数据传输完成后，将中断信号设置为高电平（“事务完成”中断）。可以从PS / 2状态寄存器读取状态（事务完成状态），并且可以从PS / 2数据寄存器读取接收到的数据。 PS / 2 CLK线将保持低电平，直到读取PS / 2数据寄存器为止。

传输模式

主机将数据发送到PS / 2设备的步骤如下。

1.启用硬件模式，选择发送模式，然后将CLK线拉低，将DATA线拉高（将0Dh写入PS / 2控制寄存器）。

2.启用中断。 （PS / 2中断控制寄存器中的TDIE位必须设置为1，因为完成数据传输后，需要读取PS / 2状态寄存器中的数据。）

3.将要发送的数据写入PS / 2数据寄存器。

4.拉低DATA线（将0Ch写入PS / 2控制寄存器）。

5.拉高CLK线（将0Eh写入PS / 2控制寄存器）。

这些步骤之后，PS / 2接口已准备好传输数据。数据传输完成后，将中断信号设置为高电平（“事务完成”中断）。可以从PS / 2状态寄存器中读取状态（事务完成状态）。 CLK线将保持低电平，直到读取PS / 2状态寄存器。

输入信号去抖动

在确定其逻辑值之前，此PS / 2接口对CLK输入信号执行去抖动操作。使能该操作时（PS / 2控制寄存器中的DCEN位置1），CLK输入信号必须稳定至少4个时钟周期。

### 软件模式选择

**软件控制PS / 2 CLK线和DATA线**

选择软件模式时（PS / 2控制寄存器中的PSHE = 0），软件可以控制PS / 2 CLK线和DATA线。 PS / 2控制寄存器中的CCLK位和CDAT位控制CLK线和DATA线。 当这些位之一清零时，相关引脚保持低电平。 当这些位之一置位时，相关引脚被拉高。

**软件控制中断**

当PS / 2硬件使能位被清除（PS / 2控制寄存器中的PSHE = 0）并且软件模式中断使能位置位（PS / 2中断控制寄存器中的SMIE = 1）时，软件可以控制PS / 2个中断。 当PS / 2控制寄存器中的CCLK位设置为高电平时，该中断设置为高电平。 如果不需要这样的中断，请清除软件模式中断使能位（PS / 2中断控制寄存器中的SMIE = 0）

## DAC 完成

DAC接口具有四个通道。 每个通道以0位至8位分辨率在0V至AVCC范围内产生输出。

## ADC 完成

该ADC模块分辨率为10bit, 拥有11个通道，其中有3个电压比较器和8个ADC通道。

## PWM 完成

该模块拥有8个8bit PWM输出。该PWM主要用于控制风扇转速及其LED灯光调节。

## 8位定时器(TMR) 完成

支持四个通道。

每个通道支持四个计数器，周期时间寄存器和占空比寄存器。

支持四个时钟源和两个时钟预分频器。

支持8位脉冲模式，16位脉冲模式和切换模式。

支持中断使能和中断禁止，分别用于周期时间匹配和占空比匹配

## CEC控制接口 完成（具体看附件总线）

CEC在HDMI系统中提供了许多高级控制功能。 若干CEC功能旨在增强用户环境中各种视听产品之间的互操作性。

CEC协议是一个单总线协议，一般CEC协议要一个上拉，协议分为数据位和起始位。

**CEC TX Flow**

If CEC operates in initiator mode and want to transimit 2 byte, include header.(1 header + 1 data)

1. Write header information into FIFO

2. Write data byte into FIFO

3. Switch Device to Initiator mode

4. Issue CEC frame by writing 1 to bit ICC.

5. Wait Interrupt. If transmitting normally, status DBD will be set high

6. Set EOM high to terminate this transfer.

7. Wait Interrupt. If transmitting normally, status DBD will be set high

8. When last byte is transmitted. CEC will auto switch to follower mode

CEC TX流

如果CEC在发起方模式下运行并想要发送2个字节，请包含标头。（1个标头+1个数据）

1.将标头信息写入FIFO

2.将数据字节写入FIFO

3.将设备切换到启动器模式

4.通过将1写入ICC位来发出CEC帧。

5.等待中断。 如果传输正常，则状态DBD将设置为高

6.将EOM设置为高电平以终止此传输。

7.等待中断。 如果传输正常，则状态DBD将设置为高

8.最后一个字节发送时。 CEC将自动切换到关注者模式

CEC RX Flow

1. Wait interrupt. If header receives normally, status HDRCV will be set high

2. F/W read CECRH to get header information which is sent from initiator device.

3. Wait interrupt. If the following data receives normally

4. F/W could check FFCNT to monitor how many data bytes are received.

5. Read CECDR to get data byte which is sent from initiator device.

6. F/W also monitor EB@CECOPSTS to know when the initiator want to terminate this transfer.

7. Follower can response nack during ACK bit by setting NKEN, however default response is ack

CEC接收流程

1.等待中断。 如果标题正常接收，则状态HDRCV将设置为高

2. F / W读取CECRH以获取从启动器设备发送的标头信息。

3.等待中断。 如果以下数据正常接收

4. F / W可以检查FFCNT来监视接收到多少数据字节。

5.读取CECDR以获取从发起方设备发送的数据字节。

6. F / W还监视EB @ CECOPSTS以了解发起方何时要终止此传输。

7.跟随者可以通过设置NKEN在ACK位期间响应nack，但是默认响应为ack

## EC访问主机控制模块(EC2C ) 完成

For EC Read Operation from a Host Controlled module register, refer to the followings:

1. Set CSAE bit in IBCTL register.

2. Make sure that both CRIB and CWIB bits in IBCTL register are cleared.

3. Setting its enable bit in IBMAE register for access module, only one module can enable at a time.

4. Assign the offset of the register in the device in IHIOA register.

5. Write 1 to CRIB bit in IBCTL register.

6. Read the CRIB bit in IBCTL until it returns 0.

7. Read the data from IHD register

有关从主机控制的模块寄存器进行EC读取操作的信息，请参考以下内容：

1.将IBCTL寄存器中的CSAE位置1。

2.确保同时清除了IBCTL寄存器中的CRIB和CWIB位。

3.在访问模块的IBMAE寄存器中设置其启用位，一次只能启用一个模块。

4.在IHIOA寄存器中分配寄存器在设备中的偏移量。

5.将1写入IBCTL寄存器的CRIB位。

6.读取IBCTL中的CRIB位，直到它返回0。

7.从IHD寄存器读取数据

For EC Write Operation from a Host Controlled module register, refer to the followings:

1. Set CSAE bit in IBCTL register.

2. Make sure that both CRIB and CWIB bits in IBCTL register are cleared.

3. Setting its enable bit in IBMAE register for access module, only one module can enable at a time.

4. Assign the offset of the register in the device in IHIOA register.

5. Write the data to IHD register, which begins a write transaction.

6. Read the CWIB bit in IBCTL until it returns 0, which represents that a write transaction has been finished.

有关从主机控制的模块寄存器进行EC写操作的信息，请参考以下内容：

1.将IBCTL寄存器中的CSAE位置1。

2.确保同时清除了IBCTL寄存器中的CRIB和CWIB位。

3.在访问模块的IBMAE寄存器中设置其启用位，一次只能启用一个模块。

4.在IHIOA寄存器中分配寄存器在设备中的偏移量。

5.将数据写入IHD寄存器，开始写入事务。

6.读取IBCTL中的CWIB位，直到返回0，这表示写事务已完成。

## 外部定时器及看门口定时器 完成

ETWD模块中的2个外部8位定时器和2个外部24位定时器，基于32.768 k时钟源

ETWD模块中有1个外部16位WDT，基于32.768 k时钟源

## GCTRL 通用控制器 完成

保存片上信息，产商信息，片上复位等通用配置功能。

## 外部GPIO控制器 完成

支持82端口GPIO

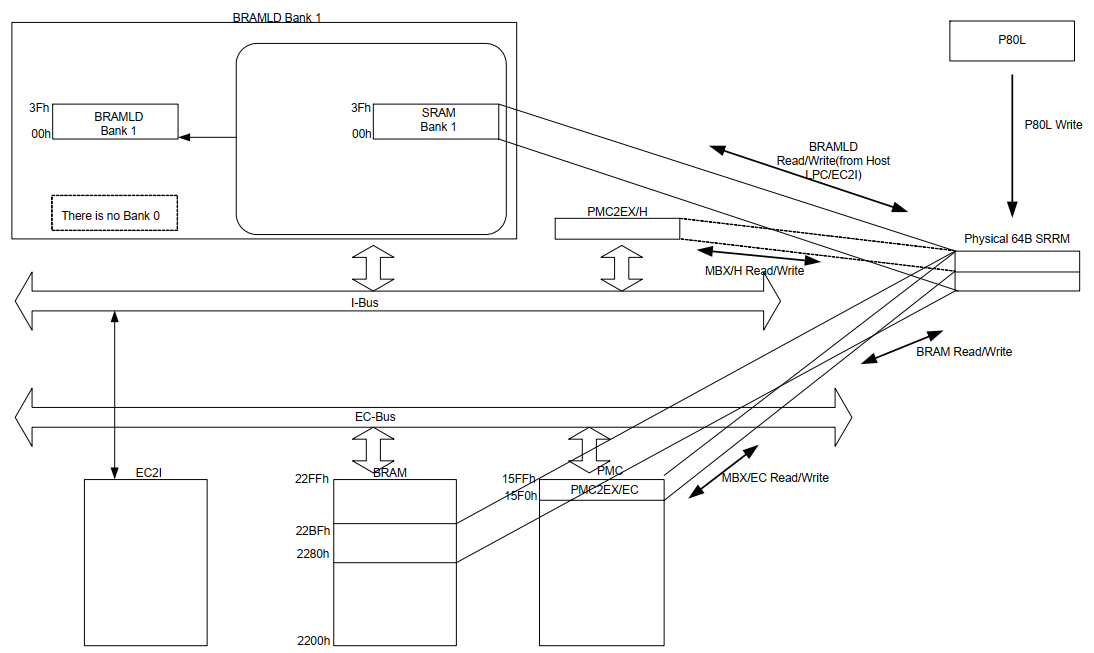
可编程上拉/下拉

施密特触发器输入

支持26端口1.8V输入电平

## BRAM 完成

该模块提供192字节的SRAM用于主机共享数据保存功能。



要注意的是，在使用共享内存的时候，在ITE出的芯片都有做保护处理，防止数据存储错误。

## 红外控制器(CIR) 完成

通过调整频率，波特率除数值和灵敏度范围，CIR寄存器能够支持主要协议，例如RC-5，NEC和RECS-80。

◼Supports 1 CIR channels

◼Supports 28 kHz ~ 57 kHz (low frequency) or 400 kHz ~ 500 kHz (high frequency) carrier transmission

◼The baud rate up to 115200 BPS (high frequency)

◼Demodulation optional

◼Supports transmission run-length encoding and deferral functions

◼Supports two dedicated 32-byte FIFO, one for data transmission and the other for data reception

支持1个CIR通道

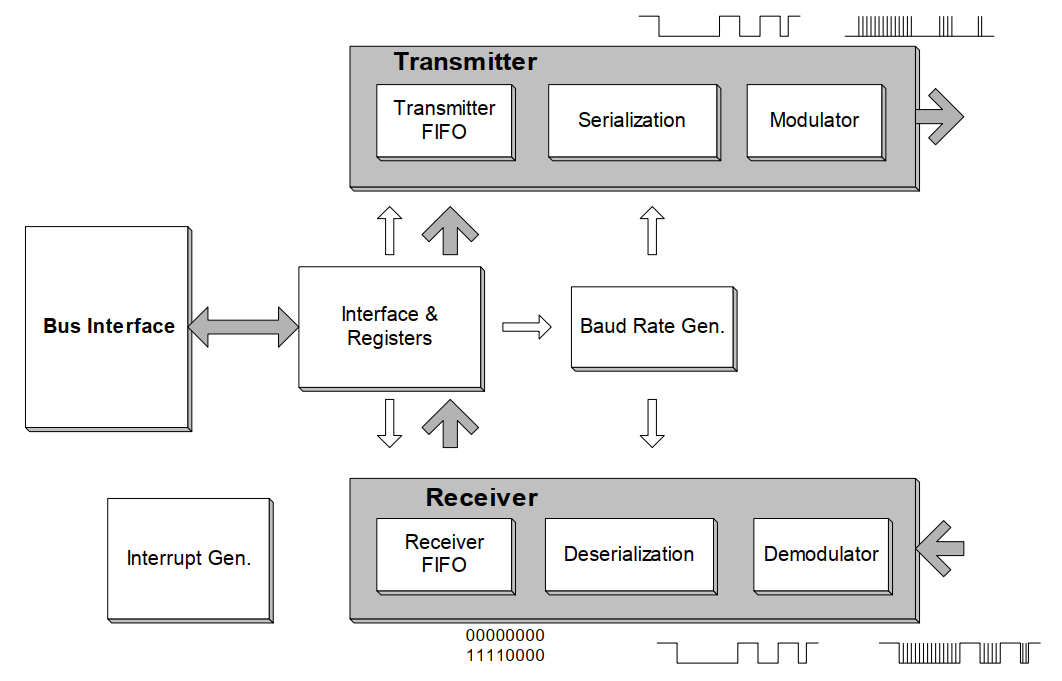
支持28 kHz〜57 kHz（低频）或400 kHz〜500 kHz（高频）载波传输

波特率高达115200 BPS（高频）

解调可选

支持传输行程编码和延迟功能

支持两个专用的32字节FIFO，一个用于数据传输，另一个用于数据接收



The CIR channel consists of two main elements, Transmitter and Receiver. The Transmitter transmits data to the FIFO, processes the FIFO data by serialization, modulation and sends out the data through the LED device. The Receiver is responsible for receiving the FIFO data, processing data by demodulation and deserialization, and storing data into the Receiver FIFO.

CIR通道由两个主要元素组成，即发送器和接收器。 发送器将数据发送到FIFO，通过序列化，调制处理FIFO数据，并通过LED器件发送数据。 接收器负责接收FIFO数据，通过解调和反序列化处理数据，并将数据存储到接收器FIFO中。

### 传输操作

Before the data transmission can be started, code byte write operations must be performed to the Transmitter FIFO C0DR. The bit TXRLE in the C0TCR should be set to “1” before the run-length decode data can be written into the Transmitter FIFO. The bit width of the serialized bit string is determined by programming the baud rate divisor registers, C0BDLR and C0BDHR. When bits HCFS and CFQ[4:0] in the C0CFR are set, either the high-speed or low-speed carrier range is selected, and the corresponding carrier frequency will also be determined. Bits TXMPM[1:0] and TXMPW[2:0] in the C0TCR specify the pulse numbers in a bit width and the required duty cycles of the carrier pulse according to the communication protocol. Only a logic 0 can activate the Transmitter LED in the format of a series of modulating pulses

在开始数据传输之前，必须对发送器FIFO C0DR执行代码字节写操作。 在将行程解码数据写入发送器FIFO之前，应将C0TCR中的TXRLE位设置为“ 1”。 串行化位串的位宽通过编程波特率除数寄存器C0BDLR和C0BDHR来确定。 当C0CFR中的HCFS和CFQ [4：0]位置1时，选择高速或低速载波范围，并且还将确定相应的载波频率。 C0TCR中的TXMPM [1：0]位和TXMPW [2：0]位根据通信协议指定位宽中的脉冲数和载波脉冲所需的占空比。 只有逻辑0才能激活一系列调制脉冲格式的发送器LED.

写操作大概操作为：page 267

1. 开启设备数据发送使能

2. 将行程解码数据写入发送器FIFO之前，使能传输控制

3. 设置数据位宽

4. 设置载波频率（table 7-36）

5. 设置调制器的脉冲宽度和设置脉冲模式。

### 接收操作

The Receiver function is enabled if bit RXEN in the C0RCR is set to “1”. Either demodulated or modulated RX# signal is loaded into the Receiver FIFO, and bit RXEND in the C0RCR determines how the demodulation logic should be used. When bits HCFS and CFQ[4:0] in the C0CFR are set, either the high-speed or low-speed carrier range is selected, and the corresponding carrier frequency will also be determined. Bit RXACT in the C0RCR is set to “1” if the serial data or the selected carrier is incoming, and the sampled data will then be kept in the Receiver FIFO. Write “1” to clear bit RXACT and then stop operation of Receiver FIFO; write “0” to bit RXEN to disable all the Receiver functions. It is strongly suggested that software clear RXACT every time when you will change the expected carrier frequency.

如果C0RCR中的RXEN位置1，则使能接收器功能。 已解调或已调制的RX＃信号被加载到接收器FIFO中，而C0RCR中的RXEND位决定了应如何使用解调逻辑。 当C0CFR中的HCFS和CFQ [4：0]位置1时，选择高速或低速载波范围，并且还将确定相应的载波频率。 如果输入串行数据或选定的载波，则将C0RCR中的RXACT位置1，然后将采样数据保存在接收器FIFO中。 写“ 1”清除RXACT位，然后停止接收FIFO的操作； 将“ 0”写入RXEN位以禁用所有接收器功能。 强烈建议您在每次更改预期的载波频率时，都要清除RXACT。

读操作大概操作为：page 267

1. 使能红外接收器功能

2.使用设备接收器解调载波

3.设置载波频率和范围

4. 使能接收数据使能，数据保存FIFO

5. 清除FIFO操作，并禁用接收器功能

## SSPI 完成

时序与SPI相同。

在mode0 下， 空闲时，SCLK处于低电平，在上升沿数据采样，下降沿数据发送；

在mode1 下， 空闲时，SCLK处于低电平，在下降沿数据采样，上升沿数据发送；

在mode2 下， 空闲时，SCLK处于高电平，在下降沿数据采样，上升沿数据发送；

在mode1 下， 空闲时，SCLK处于高电平，在上升沿数据采样，下降沿数据发送；

## 串口(UART) 完成

两个串口，波特率高达115200，如果选择了告诉波特率选择（HHS）或EC高速选择（ECHS）,则最高波特率可以高达230.4K和460.8K，这个取决于波特率发生器。

## Debugger(DBGR) 完成

DBGR模块提供访问SRAM指令，数据及其EC端外设模块，除此之外，它还提供一种下载flash方法。就是在线调试工具接口

提供断点调试以及单步调试。。

## 并口(PP) 完成

IT5576支持IEEE 1284并行端口接口，以允许进行系统内编程（ISP）或进行系统内调试（ISD）。

并口的连线与矩阵键盘扫描引脚相同。

The DBGR/EPP debug mode takes place when VSTBY is supplied (other types of power are don’t-care) and both EC chip and the flash are soldered on PCB. Parallel port interface occupies pins with the same interface as that of KBS to use the existing KBS connector.

当提供VSTBY（无关电源）并且EC芯片和闪存都焊接在PCB上时，就会发生DBGR / EPP调试模式。 并行端口接口占用的引脚与KBS的接口相同，以使用现有的KBS连接器.（是不是说只要有接无关电源就是在线调试模式，否则就是矩阵键盘扫描？？？？）

# 其他