

外形小巧的高集成度 USB Type-C™ 供电 2.0 端口控制器

重要特性

- 外形小巧的 4 x 4 mm 44-WFBGA 封装
- 集成式模拟分立元件可节省物料和设计占用空间
- 符合 USB 供电规范 2.0 的 MAC
- USB Type-C™⁽¹⁾ 连接器支持（包括连接检测和控制功能）
- 用于 CPU/SoC 通信的 I²C/SPI⁽²⁾ 接口
- USB Type-C 备用模式支持
- 双角色电源（Dual Role Power, DRP）和角色互换支持

目标应用

- 笔记本电脑
- 一体机/台式机
- 智能手机
- 平板电脑
- 监视器
- 坞站
- HDTV
- 打印机

主要优势

- 集成模拟分立元件
 - 支持 Rp/Rd 切换的 VCONN FET
 - 电量耗尽 Rd 端接
 - 针对过流情况的可编程电流检测功能
 - 针对过压情况的电压检测功能
- 集成式 5V/3A 端口电源控制器（Port Power Controller, PPC）
 - VBUS 最高支持 5V/3A
 - 根据 USB Type-C 规范提供 500 mA、900 mA、1.5A 和 3.0A 电流
- 集成 3.3V 电源开关
 - 支持电量耗尽的电池
 - 自动在 VBUS 和 +3.3V 主电源之间切换

- USB 供电 MAC
 - 符合 USB 供电规范版本 2.0
 - 供电数据包成帧
 - CRC 校验/生成
 - 4B/5B 编码/解码
 - BMC 编码/解码
 - PD 帧的 EOP/SOP 生成
 - SOP 检测和 SOP 报头处理
 - 独立的 RX/TX FIFO
 - 自动 GoodCRC 报文生成
 - 自动重试生成
 - 错误处理
 - 通过休眠状态实现低待机功耗支持
- USB Type-C 线缆检测逻辑
 - 自动线缆连接与方向检测
 - 基于检测到的方向将基带通信路由至相应 CC 引脚
 - 有源线缆的 VCONN 电源控制
 - 可配置下行端口（Downstream Facing Port, DFP）和上行端口（Upstream Facing Port, UFP）模式
 - 充电电流能力检测
 - 调试附件模式和音频适配器附件模式检测
- +1.8V I²C（1 MHz）接口支持通过配套 CPU/SoC 进行通信/配置
- 备用模式支持
 - DisplayPort™、Thunderbolt™ 和其他主要协议
- 用于选择器件模式的 CFG_SEL0 引脚
- 用于选择 I²C 地址的 CFG_SEL1 引脚⁽²⁾
- 电源和 I/O
 - 集成式 1.8V 稳压器
 - 16 个可配置的通用 I/O 引脚
- 软件
 - C 语言库
- 封装
 - 44 球 WFBGA（4 x 4 x 0.7 mm）
- 环境
 - 商业级温度范围（0°C 至 70°C）

1. USB Type-C™ 和 USB-C™ 是 USB Implementers Forum 的商标。

2. 仅在选择 UPD360 配置时适用。

致 客 户

我们旨在提供最佳文档供客户正确使用 Microchip 产品。为此，我们将不断改进出版物的内容和质量，使之更好地满足您的需求。出版物的质量将随新文档及更新版本的推出而得到提升。

如果您对本出版物有任何问题和建议，请通过电子邮件联系我公司 TRC 经理，电子邮件地址为 CTRC@microchip.com。我们期待您的反馈。

最新数据手册

欲获得本数据手册的最新版本，请访问我公司网站：

<http://www.microchip.com>

查看数据手册中任意一页下边角处的文献编号即可确定其版本。文献编号中紧跟数字串后的字母是版本号，例如：DS30000000A_CN 是文档的 A 版本。

勘误表

现有器件可能带有一份勘误表，描述了实际运行与数据手册中记载内容之间存在的细微差异以及建议的变通方法。一旦我们了解到器件 / 文档存在某些差异时，就会发布勘误表。勘误表上将注明其所适用的硅片版本和文件版本。

欲了解某一器件是否存在勘误表，请通过以下方式之一查询：

- Microchip 网站 <http://www.microchip.com>
- 当地 Microchip 销售办事处（见最后一页）

在联络销售办事处时，请说明您所使用的器件型号、硅片版本和数据手册版本（包括文献编号）。

客户通知系统

欲及时获知 Microchip 产品的最新信息，请到我公司网站 www.microchip.com 上注册。

目录

1.0 前言	4
2.0 简介	7
3.0 焊球说明和配置	9
4.0 寄存器映射	19
5.0 I2C 从控制器（仅限 UPD360-A/UPD360-B）	20
6.0 SPI 从控制器（仅限 UPD360-C）	26
7.0 时钟、复位和功耗管理	31
8.0 系统控制	35
9.0 线缆插头方向和检测	68
10.0 基带 CC 接口（BCI）	107
11.0 供电 MAC	111
12.0 USB 端口电源控制器（PPC）	171
13.0 电源开关	182
14.0 DisplayPort 热插拔检测（HPD）	195
15.0 看门狗定时器（WDT）	202
16.0 工作特性	206
17.0 封装信息	214
附录 A: 数据手册版本历史	217
Microchip 网站	218
变更通知客户服务	218
客户支持	218
产品标识体系	219

UPD360

1.0 前言

1.1 术语表

表1-1: 术语表

术语	定义
ADC	模数转换器
AFE	模拟前端
BCI	基带CC接口
布告栏	USB布告栏设备。支持备用模式的UFP需要该USB设备类来为USB主机提供产品信息。
BIST	内置自检
BMC	双相标记编码
字节	8位
CC	泛指USB Type-C™ 线缆/连接器CC1/CC2引脚
CSR	控制和状态寄存器
DB	电池电量耗尽
DFP	下行端口（USB Type-C规范定义）
DP	DisplayPort（一种VESA标准接口）
DPM	设备策略管理器（PD规范定义）
DRP	双角色电源（USB Type-C规范定义）
DWORD	32位
EC	嵌入式控制器
EP	USB端点
FIFO	先进先出缓冲区
FW	固件
FS	全速
主机	外部系统（包括处理器和应用软件等）
HPD	由DisplayPort和DisplayPort备用模式规范定义的热插拔检测功能
HS	高速
硬件	硬件（指由器件实现的功能）
IC	集成电路
IFC	帧间距
LDO	低压差稳压器
MAC	媒体访问控制器
Microchip	Microchip Technology Incorporated
N/A	不适用
OCS	过流检测
PCS	物理编码子层
PD/UPD	USB供电
PIO	通用I/O
PMIC	功耗管理集成电路
POR	上电复位
PRBS	伪随机二进制序列
QWORD	64位
SA	源地址

表1-1: 术语表 (续)

术语	定义
SBU	边带使用
SCSR	系统控制和状态寄存器
SPM	系统策略管理器 (PD规范定义)
SS	超高速
SVDM	标准/供应商定义的报文 (PD规范定义)
SVID	标准/供应商标识 (PD规范定义)
TCPC	USB Type-C 端口控制器
UFP	上行端口 (USB Type-C 规范定义)
USB	通用串行总线
USB Type-C™	USB Type-C 线缆/连接器
VDO	供应商定义的对象 (PD规范定义)
VSM	供应商特定报文
WORD	16 位
ZLP	长度为零的 USB 数据包

1.2 缓冲器类型

表1-2: 缓冲器类型

缓冲器类型	说明
IS	施密特触发输入
I2C	I ² C 接口
O8	具有 8 mA 灌电流和拉电流能力的输出
OD8	具有 8 mA 灌电流能力的漏极开路输出
PU	70k (典型值) 内部上拉。除非引脚说明中另外说明, 否则始终使能内部上拉。 注: 内部上拉电阻用于防止未连接输入悬空。请勿依靠内部电阻驱动器件外部的信号。连接到必须上拉的负载时, 必须添加外部电阻。
PD	70k (典型值) 内部下拉。除非引脚说明中另外说明, 否则始终使能内部下拉。 注: 内部下拉电阻用于防止未连接输入悬空。请勿依靠内部电阻驱动器件外部的信号。连接到必须下拉的负载时, 必须添加外部电阻。
AIO	双向模拟
P	电源引脚

注: 除非另外说明, 否则数字信号不能承受 5V 电压。

注: 有关各缓冲器电气特性的信息, 请参见第 208 页的第 16.5 节“直流特性”。

1.3 寄存器命名法

表1-3: 寄存器命名法

寄存器位类型符号	寄存器位说明
R	读: 可以对具有该属性的寄存器或位进行读操作。
W	写: 可以对具有该属性的寄存器或位进行写操作。
RO	只读: 只读。写操作不起作用。
RS	读取以置1: 此位在进行读操作时置1。
WO	只写: 如果寄存器或位为只写, 则读操作会返回未指定的数据。
W1S	写1以置1: 写1可将值置1。写0不起作用。
W1C	写1以清零: 写1可将值清零。写0不起作用。
WC	写任意内容以清零: 写任意内容均可将值清零。
LL	锁存低电平: 读取寄存器时清零。
LH	锁存高电平: 读取寄存器时清零。
SC	自清零: 内容在置1后自清零。写零不起作用。内容可被读取。
RO/LH	只读, 锁存高电平: 具有该属性的位在被读取前始终保持为高电平。此位在读取后仍会保持为高电平, 但如果消除使该位保持为高电平的条件, 该位会变为低电平。如果该位未被读取, 则无论高电平条件是否发生变化, 该位都保持高电平。
NASR	不受软件复位影响。 NASR 位的状态在软件复位置为有效时不发生更改。
保留	保留字段: 除非另外说明, 否则保留字段必须写入零, 以确保未来的兼容性。无法在读取时保证保留位的值。

1.4 参考资料

- NXP I²C总线规范 (UM10204, 2014年4月4日): www.nxp.com/documents/user_manual/UM10204.pdf
- USB 供电和USB Type-C规范: http://www.usb.org/developers/docs/usb_31_102015.zip
- VESA DisplayPort备用模式规范 1.0: <http://www.vesa.org>

2.0 简介

2.1 概述

UPD360是一款外形小巧的高集成度USB Type-C供电（Power Delivery, PD）端口控制器，其设计符合 *USB Type-C™ 线缆和连接器规范* 以及 *USB 供电规范2.0*。UPD360为USB Type-C插座提供线缆插头方向和检测，并通过集成式USB供电2.0 MAC实现与USB Type-C伙伴器件之间的基带通信。该器件可工作在独立UFP/DFP模式下，也可使用集成式I²C/SPI接口连接至配套CPU/SoC（具体取决于器件版本，请参见第2.2节“UPD360系列差异汇总”）。

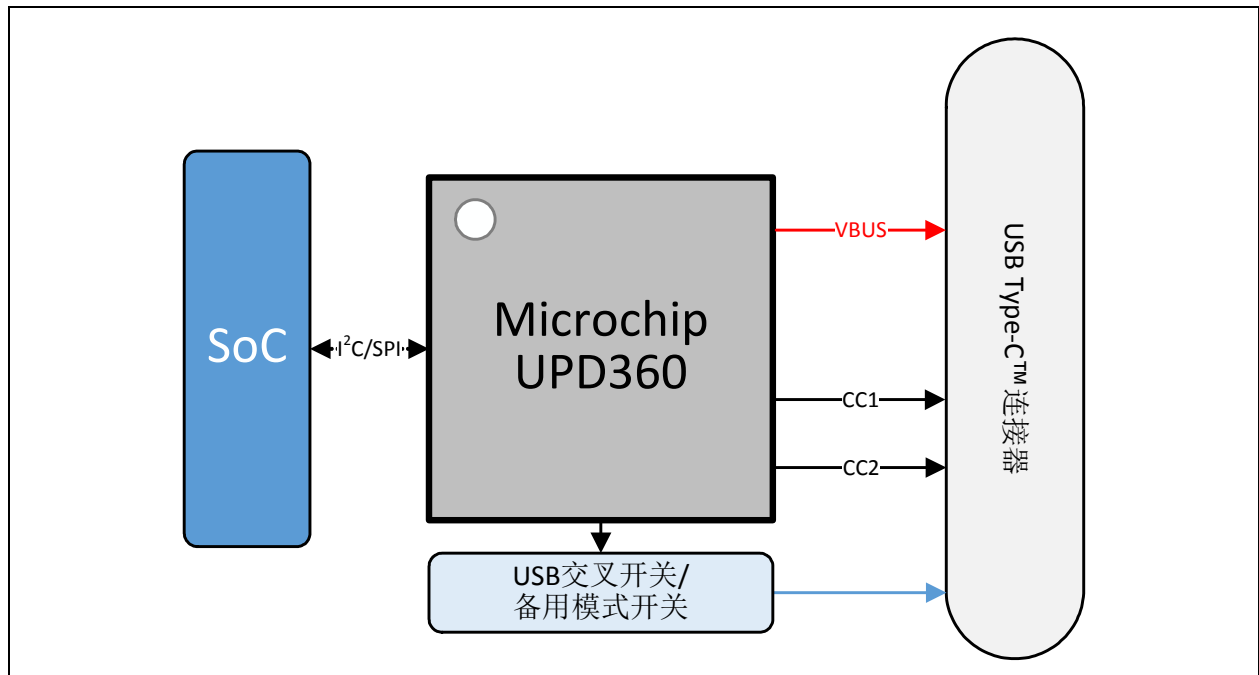
此外，UPD360还集成了USB Type-C PD应用所需的许多模拟分立元件，其中包括两个支持Rp/Rd切换的VCONN FET、一个最高支持5V/3A VBUS的端口电源控制器，以及用于过压/过流检测的电流和电压检测电路。通过集成USB Type-C PD应用所需的许多模拟分立元件，UPD360可为消费类（笔记本电脑、台式机、智能手机、平板电脑、监视器和坞站）应用提供低成本、低功耗且外形小巧（4 x 4 mm）的解决方案。

为了能够有效地支持电池电量耗尽的情况，UPD360还提供了一个集成式电源开关，用于在两个外部3.3V电源（VBUS和主电源）之间进行选择。这样可以有效地实现连接检测和系统唤醒，而无需外部处理器干预（外部处理器处于休眠模式）。

UPD360还能够使用供电2.0协议并通过USB Type-C连接器来协商备用模式。除其他主要协议外，USB Type-C连接器还兼容DisplayPort和Thunderbolt标准。

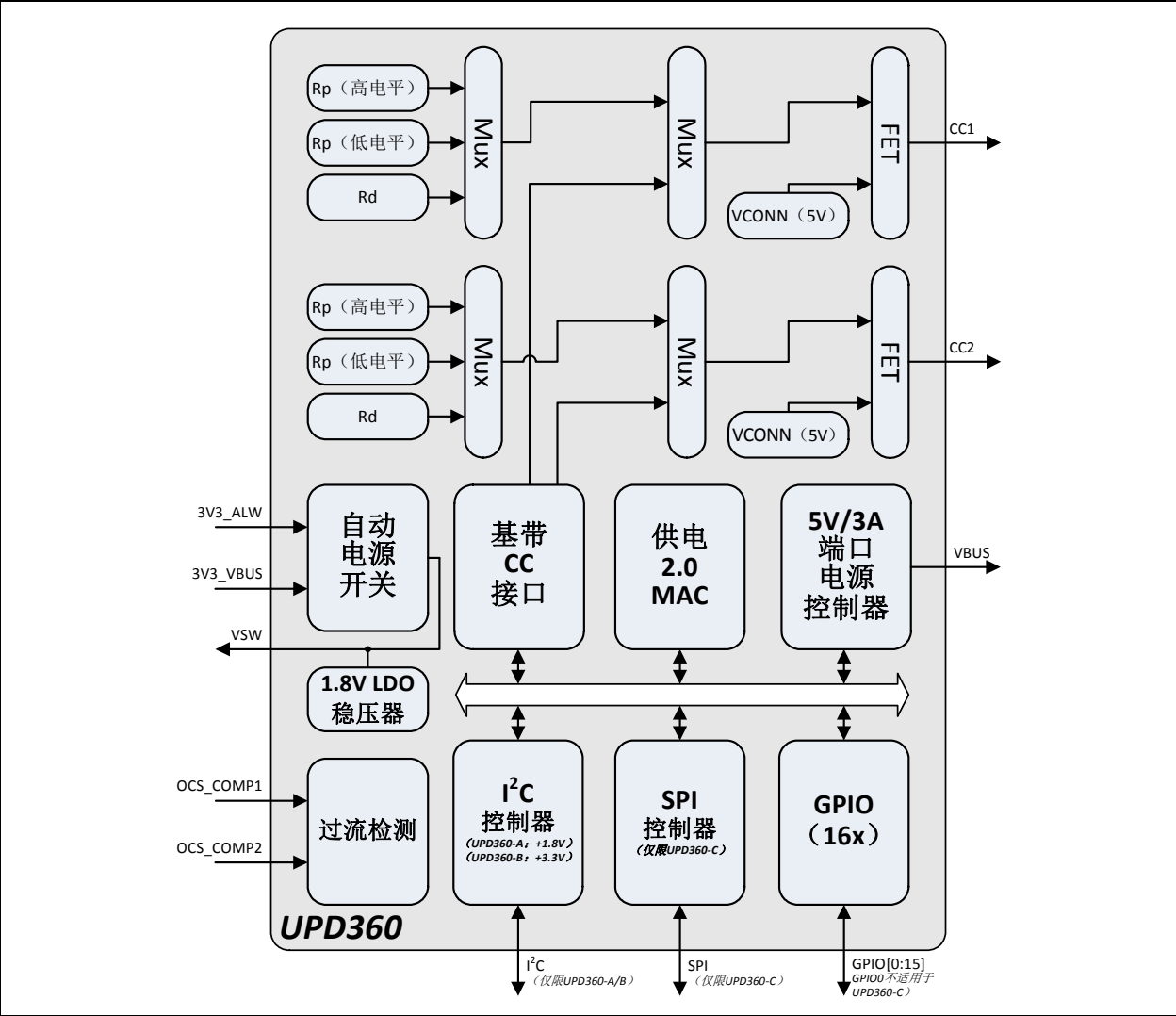
图2-1给出了使用UPD360的系统框图。图2-2给出了UPD360的内部框图。

图2-1： 系统框图



UPD360

图2-2: 内部框图



2.2 UPD360系列差异汇总

UPD360提供以下三个版本:

- UPD360-A
- UPD360-B
- UPD360-C

表2-1汇总了这些版本之间的差异。在本文档中, 将会单独介绍不属于整个UPD360系列的器件的特性。有关订购信息, 请参见第219页的产品标识体系。

表2-1: UPD360系列差异

器件	+1.8V I ² C接口	+3.3V I ² C接口	SPI接口	独立UFP/DFP模式
UPD360-A	X			X
UPD360-B		X		X
UPD360-C			X	

3.0 焊球说明和配置

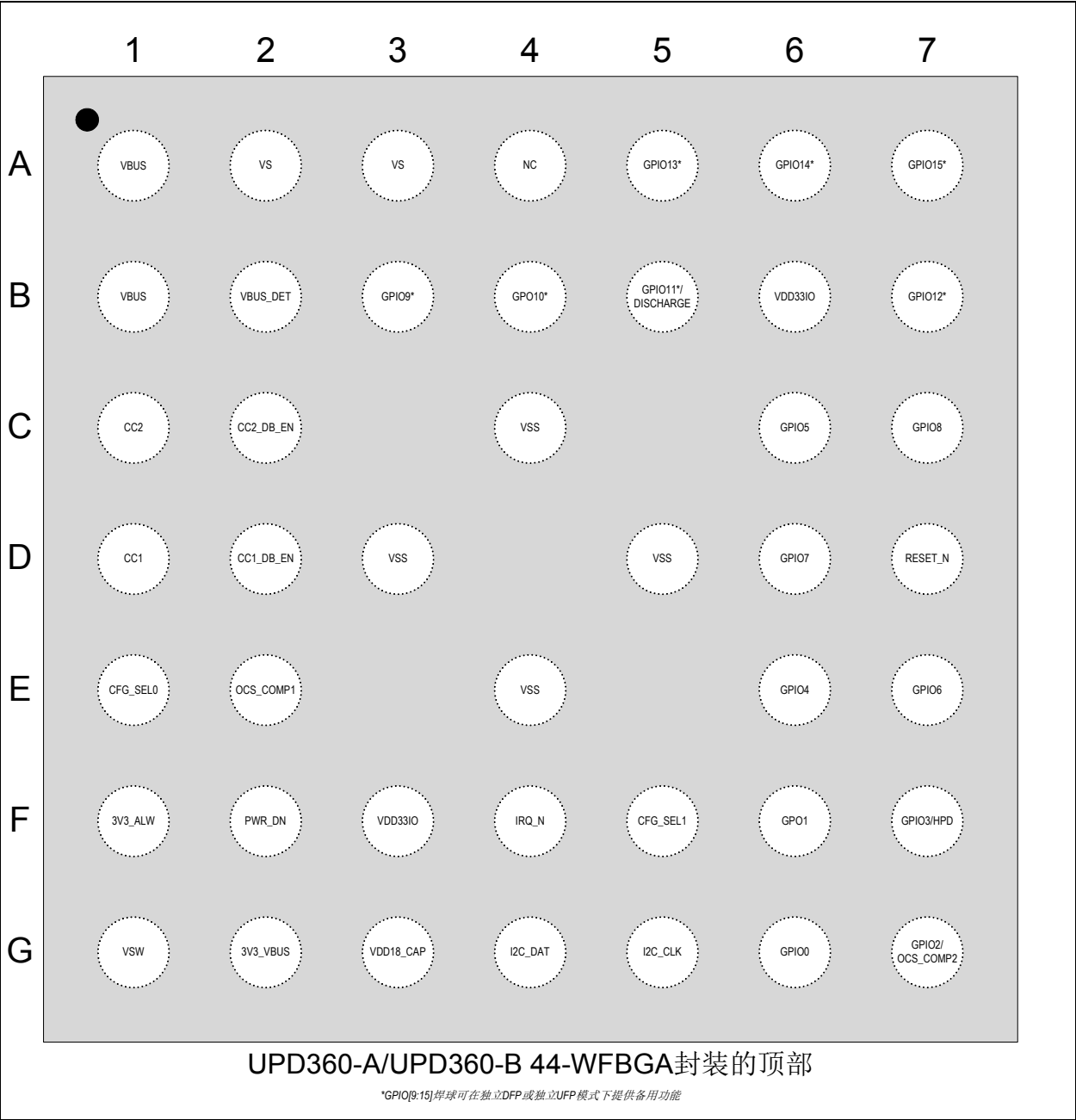
3.1 焊球分配

第9页的第3.1.1节“UPD360-A/UPD360-B焊球分配”对UPD360-A/UPD360-B的焊球分配进行了详细说明。第9页的第3.1.1节“UPD360-A/UPD360-B焊球分配”对UPD360-C的焊球分配进行了详细说明。有关UPD360系列各器件之间差异的信息，请参见第8页的第2.2节“UPD360系列差异汇总”。

3.1.1 UPD360-A/UPD360-B焊球分配

图3-1给出了UPD360-A/UPD360-B的器件焊球图。表3-1为UPD360-A/UPD360-B焊球分配表。第3.2节“焊球说明”中给出了焊球说明。

图3-1： UPD360-A/UPD360-B焊球分配（俯视图）



UPD360

表 3-1: UPD360-A/UPD360-B 焊球分配

焊球	引脚名称	焊球	引脚名称
A1	VBUS	D5	VSS
A2	VS	D6	GPIO7
A3	VS	D7	RESET_N
A4	NC	E1	CFG_SEL0
A5	GPIO13 (注3-1)	E2	OCS_COMP1
A6	GPIO14 (注3-1)	E4	VSS
A7	GPIO15 (注3-1)	E6	GPIO4
B1	VBUS	E7	GPIO6
B2	VBUS_DET	F1	3V3_ALW
B3	GPIO9 (注3-1)	F2	PWR_DN
B4	GPO10 (注3-1) (注3-2)	F3	VDD33IO
B5	GPIO11/DISCHARGE (注3-1)	F4	IRQ_N
B6	VDD33IO	F5	CFG_SEL1
B7	GPIO12 (注3-1)	F6	GPO1 (注3-2)
C1	CC2	F7	GPIO3/HPD
C2	CC2_DB_EN	G1	VSW
C4	VSS	G2	3V3_VBUS
C6	GPIO5	G3	VDD18_CAP
C7	GPIO8	G4	I2C_DAT
D1	CC1	G5	I2C_CLK
D2	CC1_DB_EN	G6	GPIO0
D3	VSS	G7	GPIO2/OCS_COMP2

注 3-1 此焊球在独立 DFP 和独立 UFP 模式下提供备用功能。更多信息，请参见第 3.1.1.1 节“独立 DFP/UFP 模式下的 UPD360-A/UPD360-B GPIO[9:15] 功能”。

注 3-2 此通用信号只能用作输出，不能在 RESET_N 置为有效期间从外部上拉。

3.1.1.1 独立DFP/UFP模式下的UPD360-A/UPD360-B GPIO[9:15]功能

当UPD360-A/UPD360-B配置为独立DFP或独立UFP模式时，以下GPIO焊球将被分配特定的备用功能，详见表3-2。

表3-2: 独立DFP/UFP模式下的GPIO[9:15]备用功能

焊球	I ² C 伙伴模式	独立DFP模式	独立UFP模式
B3	GPIO9	ORIENTATION	ORIENTATION
B4	GPO10	ATTACH	ATTACH
B5	GPIO11	OCS_N	GPIO11
B7	GPIO12	PWR_EN	SINK_5V_LEGACY_N
A5	GPIO13	PWR_CAP0	SINK_5V_1A5_N
A6	GPIO14	PWR_CAP1	SINK_5V_3A0_N
A7	GPIO15	ERR_RECOVER	GPIO15

UPD360

3.1.2 UPD360-C焊球分配

图3-2给出了UPD360-C的器件焊球图。表3-3为UPD360-C焊球分配表。第3.2节“焊球说明”中给出了焊球说明。

图3-2： UPD360-C焊球分配（俯视图）

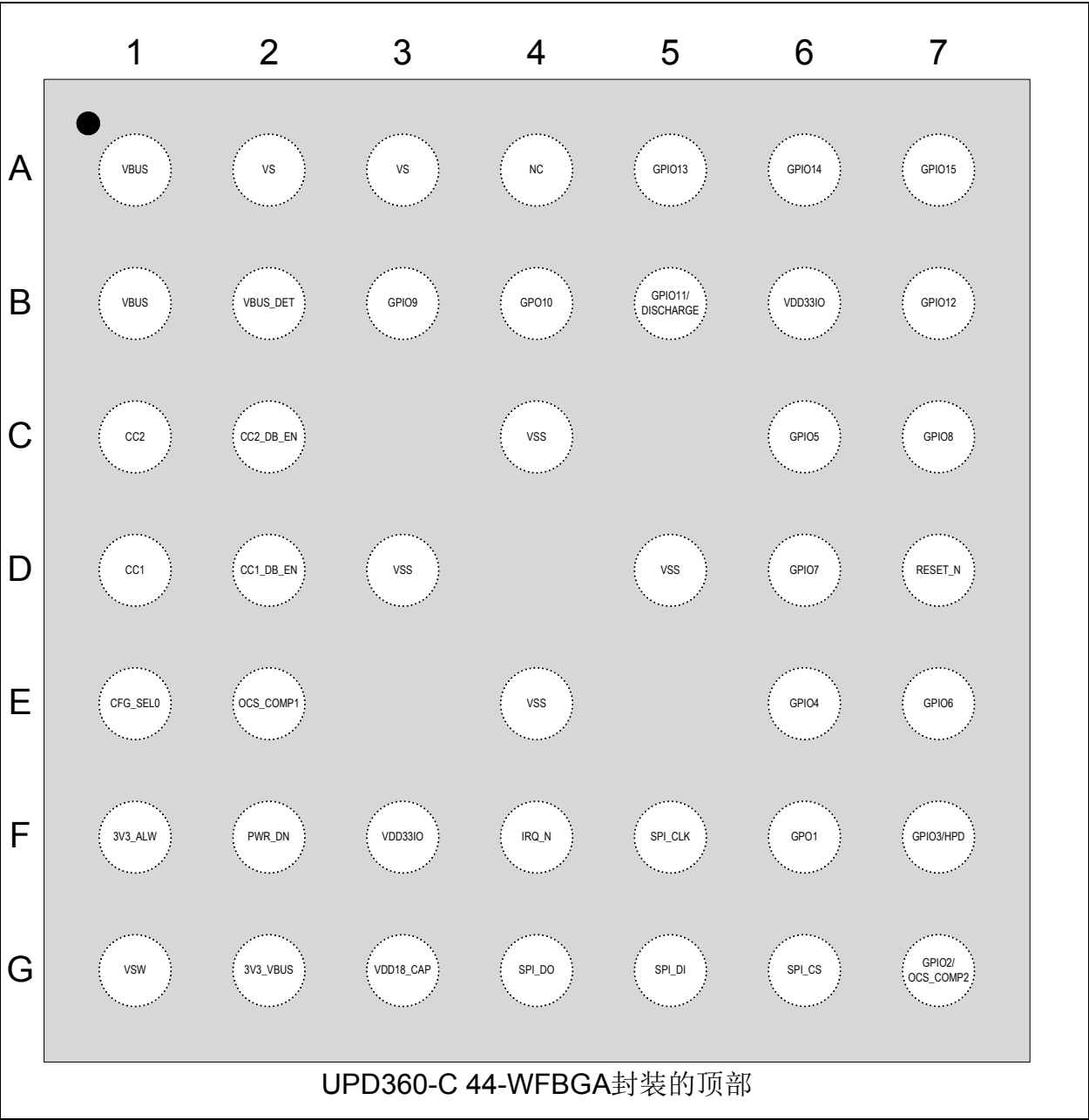


表 3-3: UPD360-C 焊球分配

焊球	引脚名称	焊球	引脚名称
A1	VBUS	D5	VSS
A2	VS	D6	GPIO7
A3	VS	D7	RESET_N
A4	NC	E1	CFG_SEL0
A5	GPIO13	E2	OCS_COMP1
A6	GPIO14	E4	VSS
A7	GPIO15	E6	GPIO4
B1	VBUS	E7	GPIO6
B2	VBUS_DET	F1	3V3_ALW
B3	GPIO9	F2	PWR_DN
B4	GPO10 (注3-3)	F3	VDD33IO
B5	GPIO11/DISCHARGE	F4	IRQ_N
B6	VDD33IO	F5	SPI_CLK
B7	GPIO12	F6	GPO1 (注3-3)
C1	CC2	F7	GPIO3/HPD
C2	CC2_DB_EN	G1	VSW
C4	VSS	G2	3V3_VBUS
C6	GPIO5	G3	VDD18_CAP
C7	GPIO8	G4	SPI_DO
D1	CC1	G5	SPI_DI
D2	CC1_DB_EN	G6	SPI_CS
D3	VSS	G7	GPIO2/OCS_COMP2

注 3-3 此通用信号只能用作输出，不能在 RESET_N 置为有效期间从外部上拉。

UPD360

3.2 焊球说明

本节详细介绍了各种器件信号的功能。

表3-4: 焊球说明

名称	符号	缓冲器类型	说明
USB Type-C™			
配置通道1	CC1	AIO	用于发现、配置和管理USB Type-C线缆连接的配置通道（Configuration Channel, CC）。更多信息，请参见第68页的第9.1节“CC比较器”。
配置通道2	CC2	AIO	用于发现、配置和管理USB Type-C线缆连接的配置通道（CC）。更多信息，请参见第68页的第9.1节“CC比较器”。
CC1 电池电量 耗尽使能	CC1_DB_EN	AIO	确定CC1广播Rd下拉还是高阻态。更多信息，请参见第77页的第9.6节“电池电量耗尽”。 注： 如果提供电量耗尽支持，则连接至CC1，否则接地。
CC2 电量耗尽 电池使能	CC2_DB_EN	AIO	确定CC2广播Rd下拉还是高阻态。更多信息，请参见第77页的第9.6节“电池电量耗尽”。 注： 如果提供电量耗尽支持，则连接至CC2，否则接地。
I²C接口（仅限UPD360-A/UPD360-B）			
I ² C时钟	I2C_CLK	I2C	I ² C时钟信号 （对于UPD360-A为+1.8V，对于UPD360-B为+3.3V）
I ² C数据	I2C_DAT	I2C	I ² C数据信号 （对于UPD360-A为+1.8V，对于UPD360-B为+3.3V）
SPI接口引脚（仅限UPD360-C）			
SPI时钟	SPI_CLK	IS	SPI时钟。 支持的最大SPI时钟频率为25 MHz。
SPI数据输出	SPI_DO	O8	SPI输出数据。
SPI数据输入	SPI_DI	IS	SPI输入数据。
SPI芯片使能	SPI_CS	IS	低电平有效SPI芯片使能输入。
供电控制			
热插拔检测	HPD	IS/O8	DisplayPort热插拔检测。更多信息，请参见第195页的第14.0节“DisplayPort热插拔检测（HPD）”。
放电	DISCHARGE	O8	VBUS放电。通过USB PD软件发出命令时使能外部VBUS放电电路。

表3-4: 焊球说明 (续)

名称	符号	缓冲器类型	说明
Type-C 连接	ATTACH	O8	在独立UFP和独立DFP模式下 (仅限UPD360-A/UPD360-B)，该信号表示线缆已插入两端的USB Type-C插座。此引脚在独立DFP模式下由器件自主驱动。 0b: 无连接 1b: USB Type-C端口已连接两端 更多信息, 请参见第9.10节“独立DFP (仅限UPD360-A/UPD360-B)”和第9.11节“独立UFP (仅限UPD360-A/UPD360-B)”。 注: 该信号在未使用时悬空。 注: UPD360-C中不提供该信号。
Type-C 方向	ORIENTATION	O8	在独立UFP和独立DFP模式下 (仅限UPD360-A/UPD360-B)，该信号用于表示由连接的DFP/UFP端接并在独立DFP模式下由器件自主驱动的CC引脚。 DFP: 0b: CC1引脚由Rd端接。 1b: CC2引脚由Rd端接。 UFP: 0b: CC1引脚被上拉至高于CC2的电压值。 1b: CC2引脚被上拉至高于CC1的电压值。 更多信息, 请参见第9.10节“独立DFP (仅限UPD360-A/UPD360-B)”和第9.11节“独立UFP (仅限UPD360-A/UPD360-B)”。 注: 该信号在未使用时悬空。 注: UPD360-C中不提供该信号。
过流检测	OCS_N	OD8	在独立DFP模式下 (仅限UPD360-A/UPD360-B)，该低电平有效信号表示过流检测。此信号映射至PPC_INT中断。 注: 该信号低电平有效, 在未使用时悬空。 注: UPD360-C中不提供该信号。
VBUS 电源使能	PWR_EN	IS	在独立DFP模式下 (仅限UPD360-A/UPD360-B)，该信号用作USB集线器的端口电源开关使能信号。 注: 该信号应连至VS电源的电源正常信号。 注: 该信号在未使用时接地。 注: UPD360-C中不提供该信号。

UPD360

表3-4: 焊球说明 (续)

名称	符号	缓冲器类型	说明
电源能力0	PWR_CAP0	IS	在独立DFP模式下, 这些信号定义器件支持的充电电流。
电源能力1	PWR_CAP1	IS	00b: USB 2.0 默认电流 01b: USB 3.0 默认电流 10b: 1.5A 11b: 3.0A 注: 在将复位置为无效后, 将无法更改PWR_CAP0和PWR_CAP1引脚的状态。 注: UPD360-C中不提供这些信号。
错误恢复	ERR_RECOVER	IS	在独立DFP模式下 (仅限UPD360-A/UPD360-B), 该引脚用于确定USB Type-C逻辑是否尝试从OCS或其他错误中自动恢复。该信号来自端口电源开关或VCONN FET。 注: UPD360-C中不提供该信号。
传统电流阱	SINK_5V_LEGACY_N	OD8	在独立UFP模式下 (仅限UPD360-A/UPD360-B), 当检测到提供传统USB电流的源时, 该引脚自动置为有效。 注: 该信号在未使用时悬空。 注: UPD360-C中不提供该信号。
1.5A 电流阱	SINK_5V_1A5_N	OD8	在独立UFP模式下 (仅限UPD360-A/UPD360-B), 当检测到提供1.5A USB电流的源时, 该引脚自动置为有效。 注: 该信号在未使用时悬空。 注: UPD360-C中不提供该信号。
3A 电流阱	SINK_5V_3A0_N	OD8	在独立UFP模式下 (仅限UPD360-A/UPD360-B), 当检测到提供3.0A USB电流的源时, 该引脚自动置为有效。 注: 该信号在未使用时悬空。 注: UPD360-C中不提供该信号。
其他			
中断	IRQ_N	OD8	低电平有效中断信号。 注: 该信号在未使用时悬空。
VBUS检测	VBUS_DET	AIO	按比例缩小版VBUS。通过电阻分压器将此信号连至VBUS。
配置选择0	CFG_SEL0	AIO	在系统复位后对此多级配置信号进行采样, 以根据所连接的1%精度电阻值来选择器件的默认工作模式。 更多信息, 请参见第81页的第9.8.1节“配置选择”。

表3-4: 焊球说明 (续)

名称	符号	缓冲器类型	说明
配置选择1	CFG_SEL1	AIO	在系统复位后对此多级配置信号进行采样, 以根据所连接的1%精度电阻值来选择器件的默认I²C从地址 (仅限UPD360-A/UPD360-B)。 更多信息, 请参见第81页的第9.8.1节“配置选择”。 注: UPD360-C中不提供该信号。
通用I/O 0-15	GPIO0、 GPO1、 GPIO2、 GPIO3、 GPIO4、 GPIO5、 GPIO6、 GPIO7、 GPIO8、 GPIO9和 GPO10	IS/O8/ OD8 (PU)	通用I/O信号可完全编程为推挽式输出、漏极开路输出或施密特触发输入 (GPO1和GPO10除外)。可选择使能可编程上拉。 注: 通用信号GPO1和GPO10只能用作输出, 在将RESET_N置为无效时必须保持在低电平状态。 注: 这些信号在未使用时接地。 注: 应在GPIO引脚上连接外部上拉电阻和下拉电阻, 以确保在复位状态时外部器件的输入被驱动为有效状态。 注: UPD360-C中不提供GPIO0。 注: 在独立DFP/UFP模式下 (仅限UPD360-A/UPD360-B), GPIO9-15具有特定的备用功能, 具体见第11页的第3.1.1.1节“独立DFP/UFP模式下的UPD360-A/UPD360-B GPIO[9:15]功能”。 更多信息, 请参见第37页的第8.2节“通用I/O”。
系统复位	RESET_N	IS	低电平有效系统复位。 注: 如果该信号未使用, 则必须将其上拉至VDD33IO。 更多信息, 请参见第34页的第7.7节“复位操作”。
掉电	PWR_DN	AI	该信号置为有效时, 会将器件置于休眠状态。 注: 该信号在未使用时接地。 更多信息, 请参见第31页的第7.2.1节“休眠”。
过流检测 比较器1	OCS_COMP1	AI	集成OCS比较器通过该引脚来检测错误条件。 注: 该信号在未使用时接地。 更多信息, 请参见第39页的第8.3节“外部过流检测”。

UPD360

表3-4: 焊球说明 (续)

名称	符号	缓冲器类型	说明
过流检测比较器2	OCS_COMP2	AI	集成OCS比较器通过该引脚来检测错误条件。 注: 该信号在未使用时接地。 更多信息, 请参见第39页的第8.3节“外部过流检测”。
无连接	NC	-	为确保正常工作, 请勿连接该引脚。
电源/地			
+3.3V 电压开关电源	VSW	P	集成电源开关的+3.3V电源输出。 注: 此引脚还为集成电源开关提供电容, 必须通过1 μ F (<100 Mohm ESR) 电容连接到地。
+3.3V VBUS电源	3V3_VBUS	P	从VBUS到集成电源开关的+3.3V电源输入。 注: 此引脚必须通过2.2 μ F 电容连接到地。
+3.3V 持续电源	3V3_ALW	P	集成电源开关的+3.3V主电源输入。 注: 此引脚必须通过2.2 μ F 电容连接到地。
+3.3V I/O 电源输入	VDD33IO	P	+3.3V I/O 电源输入。
+1.8V 数字 内核电源电容	VDD18_CAP	P	+1.8V 数字内核电源电容。为实现正常工作, 此信号必须通过1 μ F 电容连接到地。
VBUS 端口电源 开关输出	VBUS	P	端口电源开关输出。 注: 两个VBUS信号必须连接在一起。
+5V 端口电源 开关输入	VS	P	端口电源开关的+5V输入。 注: 两个VS信号必须连接在一起。 注: 对于单端口配置, 该信号需要通过一个1 μ F 电容连接到地。如果使用多个端口, 则必须为每个额外的端口额外增加一个1 μ F 的电容。
地	VSS	P	接地引脚。

4.0 寄存器映射

本章提供了器件寄存器映射列表，其中汇总了各种可直接寻址的系统控制和状态寄存器（Control and Status Register, CSR）。所有CSR均可通过器件的内部I2C从控制器（仅限UPD360-A/UPD360-B）或SPI从控制器（仅限UPD360-C）直接访问。有关系统CSR的详细说明，请根据其功能参见对应的章节。表4-1汇总了所有可直接寻址的CSR及其相应的地址。

注： 第6页的第1.3节“寄存器命名法”提供了寄存器位类型定义。

表4-1： 器件寄存器映射

地址	寄存器
3400h-FFFFh	保留
3000h-33FFh	看门狗定时器寄存器
2C00h-2FFFh	保留
2800h-2BFFh	基带CC接口寄存器
2000h-27FFh	保留
1C00h-1FFFh	电源开关寄存器
1800h-1BFFh	供电MAC寄存器
1400h-17FFh	保留
1000h-13FFh	时钟和功耗管理寄存器
0C00h-0FFFh	DisplayPort HPD寄存器
0800h-0BFFh	线缆方向和检测寄存器
0400h-07FFh	端口电源控制器寄存器
0000h-03FFh	系统控制寄存器

注： 在任何情况下都不能对保留的地址空间执行写操作。否则可能导致意外的操作与结果。

5.0 I²C从控制器（仅限UPD360-A/UPD360-B）

本章详细介绍了UPD360-A（+1.8V信号）和UPD360-B（+3.3V信号）中集成的I²C从控制器（I2C_DAT和I2C_CLK）。I²C从控制器可用于主机CPU串行管理和数据传输，并允许主机访问器件的所有配置和状态寄存器（第19页的第4.0节“寄存器映射”）。

5.1 I²C概述

I²C为双向两线数据协议。当前正在发送数据的器件被定义为“发送器”，当前正在接收数据的器件被定义为“接收器”。总线由主器件控制，主器件可生成SCL时钟、控制总线访问并产生启动和停止条件。主器件和从器件可作为发送器或接收器按位传输数据，具体取决于主器件。由于器件的I²C控制器仅用作从器件，因此术语“主机”和“主器件”是同义词，指的都是接口的外部。

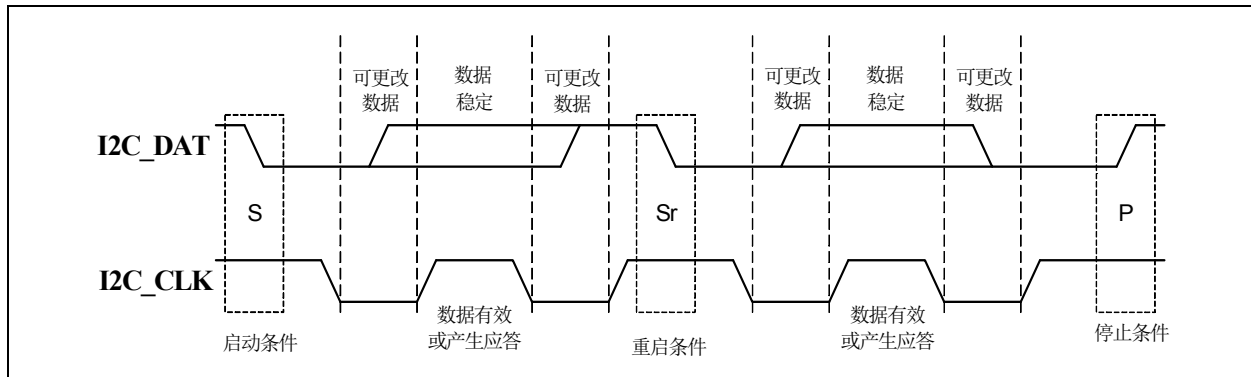
时钟（SCL）和数据（SDA）信号都具有模拟输入滤波器，可抑制小于50 ns的脉冲。当任一接口发送低电平时，数据引脚就会被驱动为低电平，以仿真I²C总线的线与功能。由于从接口永不驱动时钟引脚，因此无需线与功能。

存在以下总线状态：

- **空闲：**当总线空闲时，I2C_DAT和I2C_CLK均为高电平。
- **启动和停止条件：**启动条件（S）被定义为SCL为高电平时SDA线从高电平到低电平的变化。停止条件（P）被定义为SCL为高电平时SDA线从低电平到高电平的变化。在启动条件之后可将总线视为繁忙状态；在停止条件之后可将总线视为空闲4.7 μs/1.3 μs/0.5 μs（分别对应于100 kHz/400 kHz/1 MHz操作）。如果没有停止条件，则在重复启动条件（Sr）后总线保持繁忙状态。停止/启动序列和重复启动在功能上是相同的。
- **数据有效：**在启动条件之后，当SCL为高电平时并且SDA保持稳定时数据有效。只能在时钟为低电平时更改数据。每个时钟脉冲都有一个有效位。每个字节必须为8位长，且先发送MSb。
- **应答：**数据的每个字节后都有一个应答位。主器件生成第九个时钟脉冲作为该位，同时发送器释放SDA（高电平）。为产生肯定的“应答”（ACK），接收器将SDA驱动为低电平，使其在时钟的高电平周期内保持有效，其中包括建立和保持时间。为产生否定的“无应答”（NACK或ACK），接收器将允许该线路在此位时间内保持高电平。接收器可为主器件，也可可为从器件，具体取决于数据的方向。通常，接收器会应答每个字节。如果主器件为接收器，则不会在传输最后一个字节后产生应答。这将通知从器件不要驱动下一个数据字节，进而释放SDA，以使主器件生成停止或重复启动条件。

图5-1显示了典型I²C周期的各总线状态。

图5-1： I²C周期



5.2 I²C从器件操作

I²C从串行接口由一个数据线（I2C_DAT）和一个串行时钟（I2C_CLK）组成。串行时钟由主器件驱动，而数据线是双向的。两个信号均为漏极开路信号，需要外部上拉电阻。

I²C从控制器可实现低电平I²C从串行接口（启动和停止条件检测、数据位发送/接收和应答产生/接收）、处理从器件命令协议并执行系统寄存器读写操作。它容许并提供时钟拉伸，尤其适用于为透明的主机访问唤醒功能（见第32页的第7.3节“异步I2C唤醒（仅限UPD360-A/UPD360-B）”）提供支持。

I²C从控制器符合NXP I²C总线规范（UM10204，2014年4月4日），可满足以下模式的流量要求：

- 标准模式（Sm，100 kbps）
- 快速模式（Fm，400 kbps）
- 增强型快速模式（Fm+，1 Mbps）

有关时序信息，请参见第210页的第16.6.2节“I2C从接口（仅限UPD360-A/UPD360-B）”。

5.2.1 I²C从器件命令格式

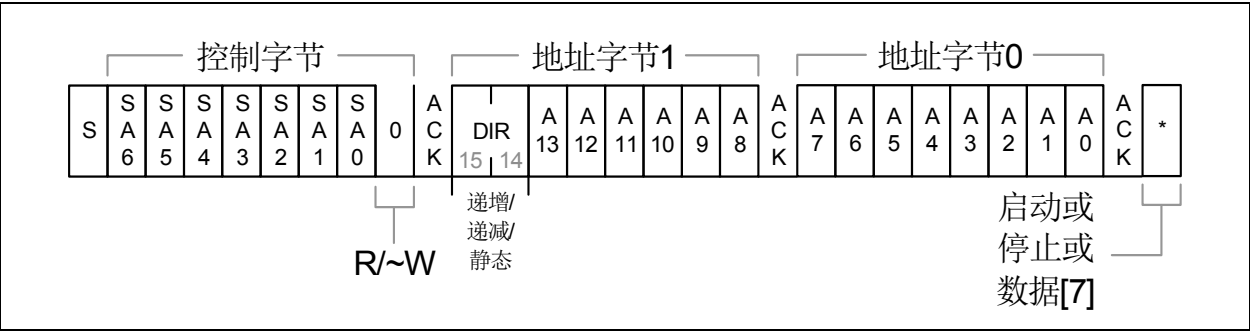
I²C从串行接口支持单寄存器和多寄存器读写命令。主器件先发送一个启动条件，然后发送控制字节，以此来启动读命令或写命令。控制字节由一个7位从地址和一个1位读/写指示（R/~W）组成。器件使用的默认从地址通过CFG_SEL1配置脚选择。假设控制字节中的从地址与此地址匹配，器件将会应答控制字节。否则，在下一个启动条件之前，整个序列都将被忽略。I²C从控制器还支持广播呼叫地址。图5-2、图5-4和图5-5给出了I²C命令格式。

如果控制字节中的读/写指示（R/~W）为0（写），则主器件之后发送的两个字节为寄存器地址，并且这两个字节是必需的。地址字段的高2位表示方向控制（DIR），表示多字节访问会使发出的地址递增、递减还是固定（如静态）（第5.2.2节）。在器件应答地址字节后，主器件可发送数据字节，这些字节将被写入从该地址开始的连续寄存器。它还可能发送另一个启动条件（以开始读取数据）或停止条件（仅设置地址）。后两种情况会在写入任何数据之前终止当前写操作，但会影响用于后续读取的内部寄存器地址的设置。

如果控制字节中的读/写指示（R/~W）为1（读），则器件将在控制字节应答位之后开始发送数据。读命令不能自行指定地址，但可选择以写命令为前缀进行设置（见图5-4，前缀显示为灰色）。不过，如果此读命令紧跟在多寄存器读命令或写命令之后，该地址可能已根据DIR字段在内部递增或递减，因此该读命令将在下一个连续字节地址处开始访问。此外，多字节读命令将会忽略上述访问操作，并根据先前发出的DIR字段继续在内部递增/递减（第5.2.2节）。

寄存器地址字段的长度始终为两个完整字节。某些高位与地址无关。为实现向上兼容，无关的寄存器地址位应始终以0进行发送。

图5-2: I²C从器件寻址



注： 在字节（地址和数据）内，对于数据而言，先传送最高有效位。对于地址而言，先传送最高有效字节。所有寄存器的访问均以字节为单位，寄存器数据以递增字节地址顺序传送。请参见器件寄存器布局，确定这对任一多字节值的有效顺序的影响。

5.2.2 多字节寄存器地址序列

地址字段位[15:14]中的DIR子字段确定如何解析多字节序列。每当发出地址时，该字段都在内部保存，但其不适用于I²C（多字节传输、读或写除外）。DIR字段定义如下：

- **DIR = 00b:** 为多字节数据包中的后续字节访问选择内部保存的寄存器地址的自动递增功能。
- **DIR = 10b:** 为多字节数据包中的后续字节访问选择内部保存的寄存器地址的自动递减功能。
- **DIR = 11b:** 选择一个固定地址。内部寄存器地址不会发生任何修改，即，所有后续访问（单字节或多字节）均针对同一寄存器进行。
- **DIR = 01b:** 保留供将来使用。

请注意，只能通过发布地址来更改DIR字段。在更改之前，它将保持不变，同时对后续多字节读取数据包产生影响。

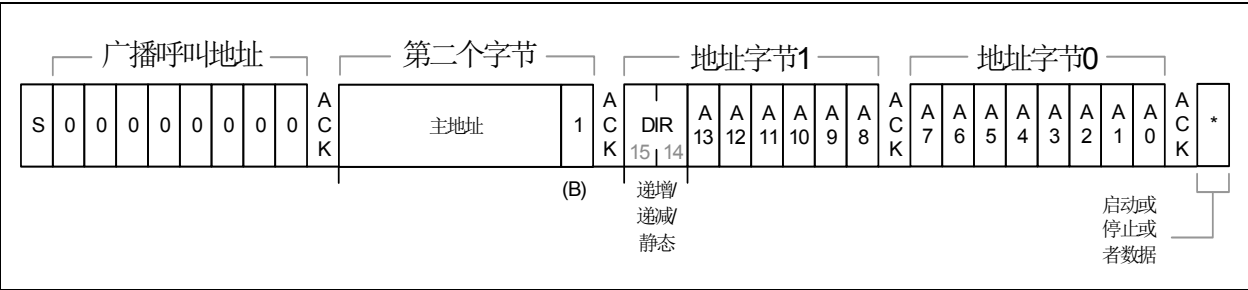
5.2.3 广播呼叫地址

该器件支持I²C广播呼叫地址。此功能的目的是对包含多个UPD360从器件的拓扑进行全局I²C写操作。这最大程度地减少了用于器件复位（通过**硬件控制寄存器（HW_CTL）**）以及各种通用配置寄存器的I²C事务。此工作模式适用于仅由UPD360从器件组成的拓扑。此工作模式可能与共存在I²C总线上的非UPD360从器件不兼容。

仅支持将广播呼叫地址的最低有效位“B”设置为1的情况。器件会忽略将广播呼叫地址的最低有效位“B”设置为0的情况。对于后一种情况，器件将应答广播呼叫地址的第一个字节。器件将忽略并默默丢弃所有后续字节，不会进行应答。广播呼叫地址的第二个字节也会被器件忽略，不会被应答。

图5-3给出了支持的广播呼叫地址格式。

图5-3: I²C广播呼叫地址



5.2.4 器件初始化

在器件自行初始化到各种配置输入均有效的程度之前，I²C从接口将不会响应任何外部引脚活动，也不会受其影响。在该状态下，此器件不应由主器件访问。但是，如果需要这样做，该状态会在控制字节以及主器件发送的任何后续字节的ACK位时间内在外部显示为NACK（高电平）。在内部初始化后，器件将继续以这种方式工作，直至接收到第一个启动条件。在地址写入操作成功完成（图5-2）之前，不应尝试读取事务，否则，读取的值可能会不可预测。或者，也可通过将IRQ_N引脚置为有效来指示器件就绪。

5.2.5 功耗管理期间和功耗管理之后的访问

在低功耗模式下，启动条件会触发器件使其唤醒，器件也会延长 I^2C 时钟（为低电平），直至其内部时钟处于运行状态并被锁定。它随后会释放 I^2C 时钟，并处理传入数据包。

在接收从地址位之前将执行上述步骤，即，如果有多个此类型的器件在同一 I^2C 总线段上处于休眠状态，则这些器件均会延长时钟，并且都会唤醒，而无论其是否真正被寻址。如果 I^2C 事务的从地址与 I^2C 从地址寄存器（ $I2C_ADDR$ ）（仅限UPD360-A/UPD360-B）中指定的值不匹配，则器件将自动掉电。

5.2.6 I^2C 从器件读序列

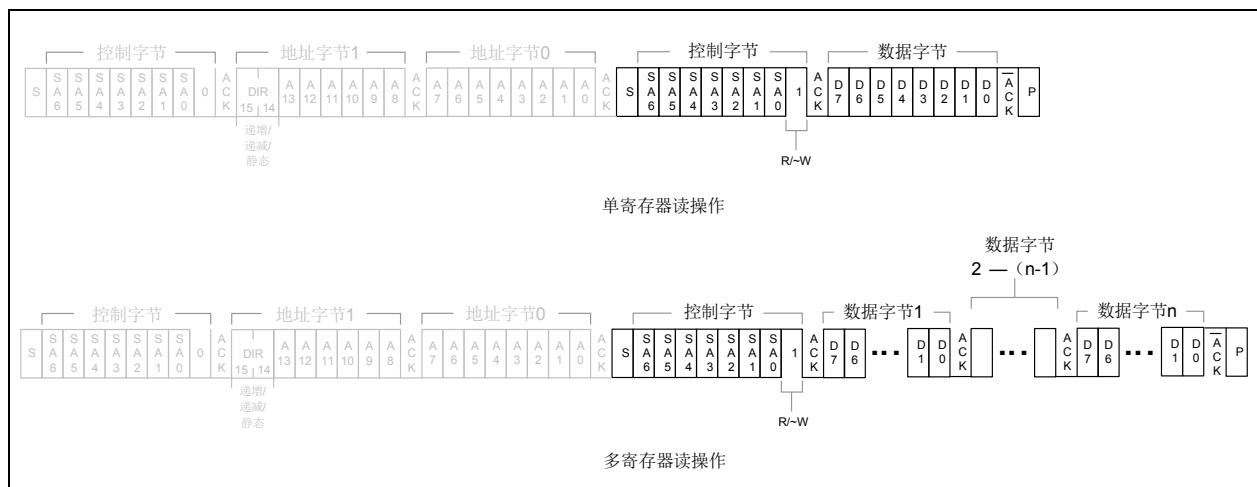
器件寻址后（详见第5.2.1节），当主器件发送一个启动条件和R/~W位置1的控制字节时，会从器件读取寄存器。假设控制字节中的从地址与器件地址匹配，则控制字节会由器件应答。否则，在下一个启动条件之前，整个序列都将被忽略。在应答后，器件将根据最后发出的DIR地址子字段（第5.2.2节）从连续寄存器地址发送1个或多个字节的数据，直至主器件发送一个无应答信号（后跟停止条件）。无应答信号通知器件不要再发送任何字节。

如果只读取一个寄存器字节，则内部寄存器地址不变；否则（多寄存器读取），在每个字节（包括最后一个字节）后，内部寄存器地址都会递增或递减（第5.2.2节）。如果内部地址达到其最大值，则会计满返回0。

如果主器件发送意外启动或停止条件，则器件将立即停止发送，并根据需要响应下一个序列。

图5-4给出了典型的单寄存器读操作和多寄存器读操作。允许先对地址进行可选写操作，以灰色显示。请注意，本示例显示的是简缩情况，其中写操作在读传输开始之前无停止条件。在这种情况下，仍允许停止，但不是必需的。

图5-4: I^2C 从器件读操作



5.2.7 I^2C 从器件写序列

器件寻址后（详见第5.2.1节），当主器件继续发送数据字节时，会将寄存器值写入器件。器件将应答每个字节。在应答后，主器件可在任一数据字节之后发送另一个启动条件或用停止条件暂停序列。在单字节写操作后，内部寄存器地址保持不变。

当主器件在第一个数据字节后发送额外数据字节时，将执行多次写操作。内部地址将自动递增，并会写入下一个寄存器。内部地址达到其最大值后，会计满返回0。当主器件发送另一个启动或停止条件时，多次写入将结束。在执行多次写入时，对于每次写入（包括最后一次写入），内部寄存器地址都会递增或递减（第5.2.2节）。这与新启动条件之后

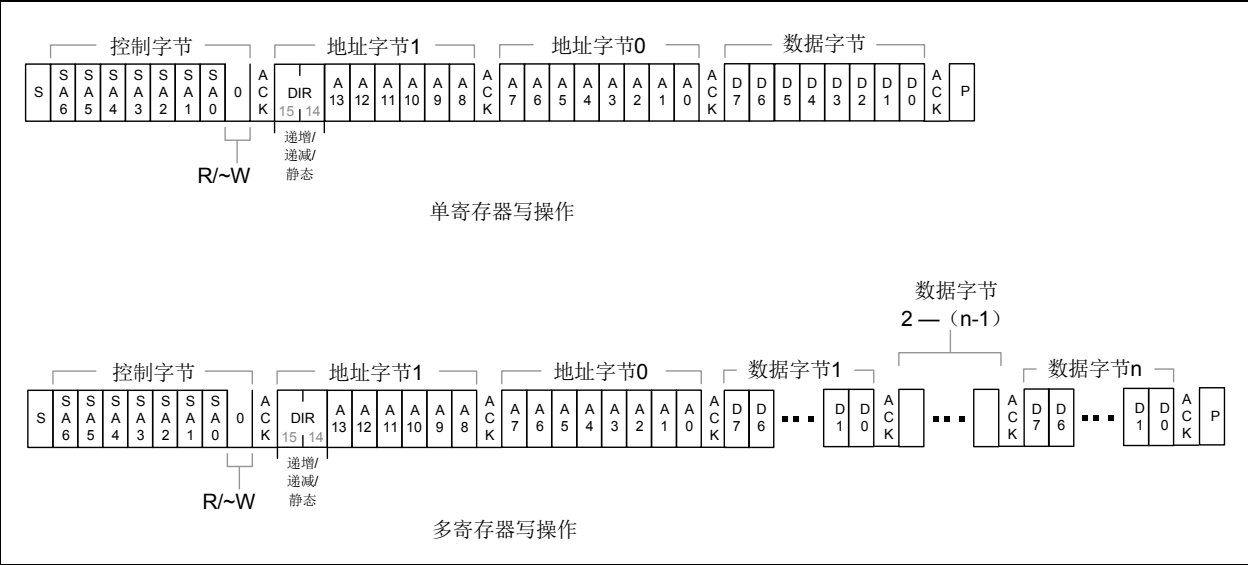
的后续写操作无关，因为随后必须包括新寄存器地址（及其DIR子字段）。但是，如果不先复位寄存器地址，则会影响后续读操作所使用的地址。

对于单次和多次写操作，如果主器件发送意外启动或停止条件，则器件将立即停止，并根据需要响应下一个序列。

对多字节寄存器的数据写入操作可能会延迟，直到输入所有位。如果未写入整个寄存器（主器件意外发送启动或停止条件），则会认为写操作无效，寄存器不受影响。可在一个多次写周期中写入多个寄存器，每个寄存器依次写入。对于未使用的寄存器地址，不能执行I²C写操作。

图5-5给出了典型的单寄存器写操作和多寄存器写操作。

图5-5: I²C从器件写操作



5.2.8 特殊的CSR处理

5.2.8.1 实时位

寄存器值在每次寄存器读取开始时锁存（寄存），以防止主机读取一个变化的值。在多次寄存器读取序列中，每个寄存器单独进行锁存。

5.2.8.2 读取时变化的寄存器和FIFO

因读操作触发副作用的任何单字节寄存器（例如，包含“读取时清零”位或推进FIFO结构）仅在主机开始访问相应值后触发。主器件获知的值始终为原始值，而永远不是因副作用更新后的结果。

对于被视为单个单元的多字节寄存器，更改可能会延迟，直至读取完寄存器的所有字节。如果主机在多字节寄存器的前几个字节之上发送无应答信号，或者在应答整个寄存器之前意外出现开始或停止条件，则相应的读操作可能被视为无效，不会触发副作用。

采用多寄存器读取访问的寄存器读操作将触发多个副作用（在读取时发生）。以下寄存器具有读取副作用：

- 看门狗计数器寄存器（WDT_COUNT）

5.2.8.3 读取时变化的实时位

如上所述，具有实时位的寄存器中的当前值（像任何寄存器一样）作为输出数据进行捕捉和锁存，随后原始寄存器中的读取时变化位将发生改变。为防止数据捕捉之后、“读取时变化”之前发生的硬件事件丢失，这些硬件事件将保持等待状态，直到读操作之后以及因读取而发生任何变化之后才结束。该序列还可确保发生硬件事件时位的边沿。

6.0 SPI从控制器（仅限UPD360-C）

本章详细介绍了UPD360-C中集成的SPI从控制器（**SPI_DI**、**SPI_DO**、**SPI_CLK**和**SPI_CS**）。SPI从控制器可用于主机CPU串行管理和数据传输，并允许主机访问所有器件配置和状态寄存器（[第19页的第4.0节“寄存器映射”](#)）。

6.1 SPI概述

SPI从模块提供低引脚数同步从接口，方便器件与主机系统之间的通信。凭借SPI从器件，可访问系统CSR、内部FIFO和存储器。它支持一条和多条采用递增、递减和静态寻址的寄存器读写命令。SPI模式下仅支持一位通道，频率最高25 MHz。

以下内容对SPI从模块提供的功能进行了概述：

- **快速读取**：4线（时钟、选择、数据输入和数据输出）读操作。串行命令、地址和数据。由于历史原因，这一过程称为“快速”读取，并且是惟一支持的读命令。首次访问时需要一个空字节。采用递增、递减或静态寻址进行单次和多次寄存器读操作。
- **写**：4线（时钟、选择、数据输入和数据输出）写操作，频率最高25 MHz。串行命令、地址和数据。采用递增、递减或静态寻址进行单次和多次寄存器写操作。

6.2 SPI从器件操作

SPI帧在**SPI_CS**的下降沿开始，在**SPI_CS**的上升沿结束。在**SPI_CS**的边沿，**SPI_CLK**时钟可能处于复位状态（即低电平（模式0）或高电平（模式3），由主器件选择）。

SPI_DI引脚（通常称为“MOSI”）上的输入数据在**SPI_CLK**输入时钟的上升沿采样。输出数据在时钟的下降沿通过**SPI_DO**引脚（通常称为“MISO”）发出。当**SPI_CS**片选输入为高电平时，**SPI_DI**和**SPI_CLK**输入被忽略，**SPI_DO**输出悬空。

每个帧以一个8位指令字节开始，由主器件发送，并且从**SPI_CS**变为有效后的输入时钟的第一个上升沿开始由**SPI_DI**接收。

对于写和（快速）读指令，两个地址字节在指令字节之后。地址字段表示一个字节地址。十四个地址位指定地址。剩余的两位[15:14]构成地址字段的DIR子字段，用于指定帧中连续数据字节的地址自动递增（00b）还是自动递减（10b）。特殊的静态地址编码（11b）可使地址在整个数据帧中保持静态，导致帧中传送多个字节时重复访问一个单字节地址。DIR子字段编码01b保留，并且应在实现中解码为与00b相同，以便最大程度减少软件错误（递增至地址空间之外）的影响。

对于快速读取指令，地址字节后面有一个空字节。空字节占用8位，每个时钟1位。

该器件通常不会在指令、地址或空字节周期期间驱动**SPI_DO**，但也存在特殊情况，具体请参见[第27页的第6.2.2节“功耗管理期间和功耗管理之后的访问”](#)。

对于快速读取指令，一个或多个8位数据字段跟在空字节之后。对于写指令，这些数据字段紧跟在地址字节之后。

指令、地址和数据字段中的各个字节先传送最高有效位（most-significant bit, msb）。两字节的地址字段先传送最高有效字节（Most-Significant Byte, MSB）。多字节数据值以地址字段的DIR子字段（bit[15:14]）指定的顺序传送，因此其顺序可以通过使用递增模式（从最低字节地址开始）或递减模式（从最高字节地址开始）有效选择。

SPI接口所支持的连续命令之间的最短时间为50 ns（**SPI_CS**的无效时间最短为50 ns）。

表6-1列出了SPI从控制器支持的指令。不支持的指令保留，不能使用。

表6-1： SPI指令

指令	说明	总线位宽	指令代码	地址字节	空字节	数据字节	最大频率
读							
FASTREAD	读，高速格式	1	0Bh	2	1	1至∞	25 MHz
写							
WRITE	写	1	02h	2	0	1至∞	25 MHz

6.2.1 器件初始化

在器件初始化到各种配置输入均有效的程度之前，SPI接口将不会响应任何外部引脚活动，也不会受其影响。

一旦器件初始化完成，SPI接口将忽略这些引脚，直至检测到SPI_CS的上升沿。

如果器件初始化在有效周期内（SPI_CS为低电平）完成，则帧尾一定在任何内部寄存器受影响或SPI接口的状态发生变化之前出现（SPI_CS恢复高电平）。

器件初始化后的首次SPI访问必须始终为对SPI测试寄存器（SPI_TEST）（仅限UPD360-C）的假读访问。

6.2.1.1 针对初始化完成的SPI从器件读取轮询

如果SPI_DO上连接有一个外部弱上拉电阻，则当器件未初始化时，几乎会从任何内部寄存器中读取到值FFh。通过验证SPI测试寄存器（SPI_TEST）（仅限UPD360-C）中是否至少有一个“0”位，可以确定器件何时初始化。

6.2.2 功耗管理期间和功耗管理之后的访问

SPI通信中的唤醒事件是特定器件的本地事件，不会影响其他器件的状态，即使这些器件与特定器件在同一条SPI总线上也是如此。在唤醒完成之前，SPI接口将使SPI_DO引脚保持低电平，持续时间为SPI_CS的低电平时间。

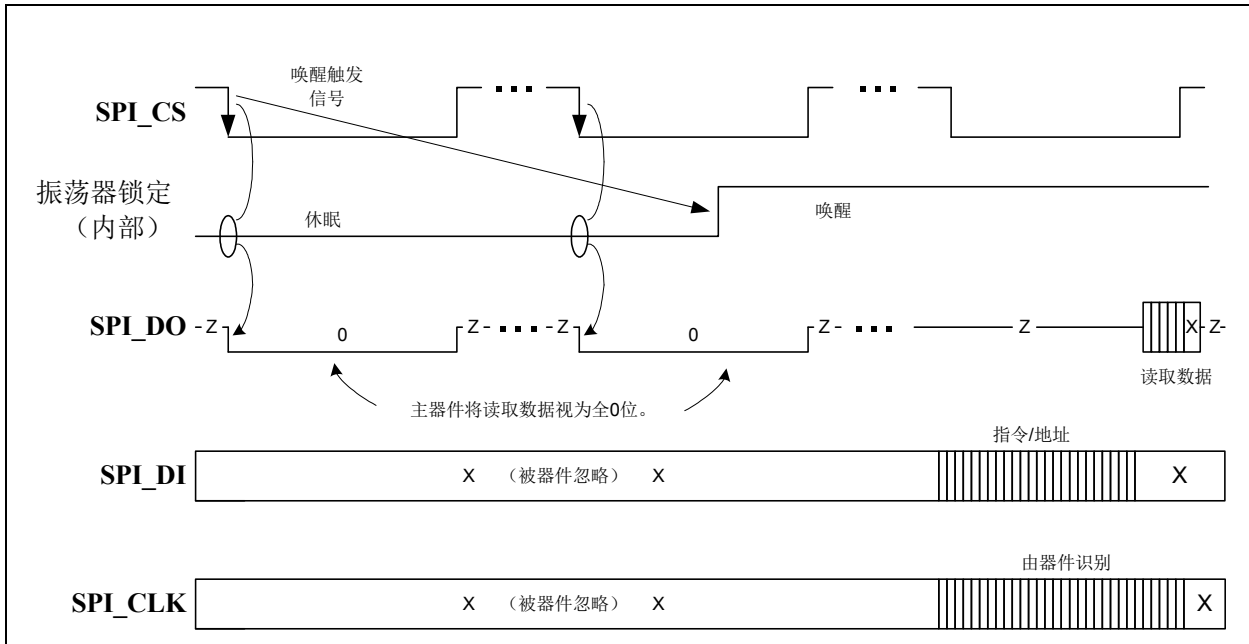
在器件唤醒之前，主器件执行的任何读访问几乎都将返回全“1”位。为了确定器件何时处于唤醒状态以及SPI接口何时正常工作，主器件应在单独的帧中反复轮询SPI测试寄存器（SPI_TEST）（仅限UPD360-C）（SPI_CS由低电平变为高电平）。一旦读取到正确的非零值，即可认为接口正常工作。IRQ_N引脚置为有效可作为轮询的替代方法，用于指示器件已就绪。

一旦功耗管理模式切换回ACTIVE，则在SPI_CS和SPI_DO均处于低电平状态后，SPI接口仍然会忽略SPI_CLK和SPI_DI引脚，直至检测到SPI_CS为高电平。在下一个SPI_CS的下降沿，SPI通信将继续正常进行。

在执行SPI通信后的任何时间，器件均不会回到非通信功耗状态，直至SPI主器件发出明确允许的命令。

图6-1给出了从SPI通信唤醒的序列。

图6-1: SPI通信唤醒功耗管理



6.2.3 SPI读命令（快速读取）

快速读取命令受SPI从器件支持。单个字节或多个字节可以在单个帧中读取（SPI_CS为低电平）。

快速读取是器件支持的唯一读访问形式。该指令在SPI_DI上输入指令代码、地址和一个空字节，然后以每个时钟一位的方式在SPI_DO上输出数据。

首先将SPI_CS置为有效，选择SPI从接口。然后向SPI_DI引脚中输入8位FASTREAD指令0Bh，接着是两个地址字节和一个空字节。地址字节指定器件内的字节寄存器地址，还指定多字节读操作（如下）中连续字节地址的排序方式。空字节的内容是无关的。

在最后一个空位的上升沿之后的时钟下降沿，SPI_DO引脚从所选寄存器字节的最高有效位开始驱动。剩余的寄存器位在随后的时钟下降沿移出。

SPI_CS输入变为无效状态以结束周期。器件会将SPI_DO引脚悬空作为响应。

6.2.3.1 多次字节读操作

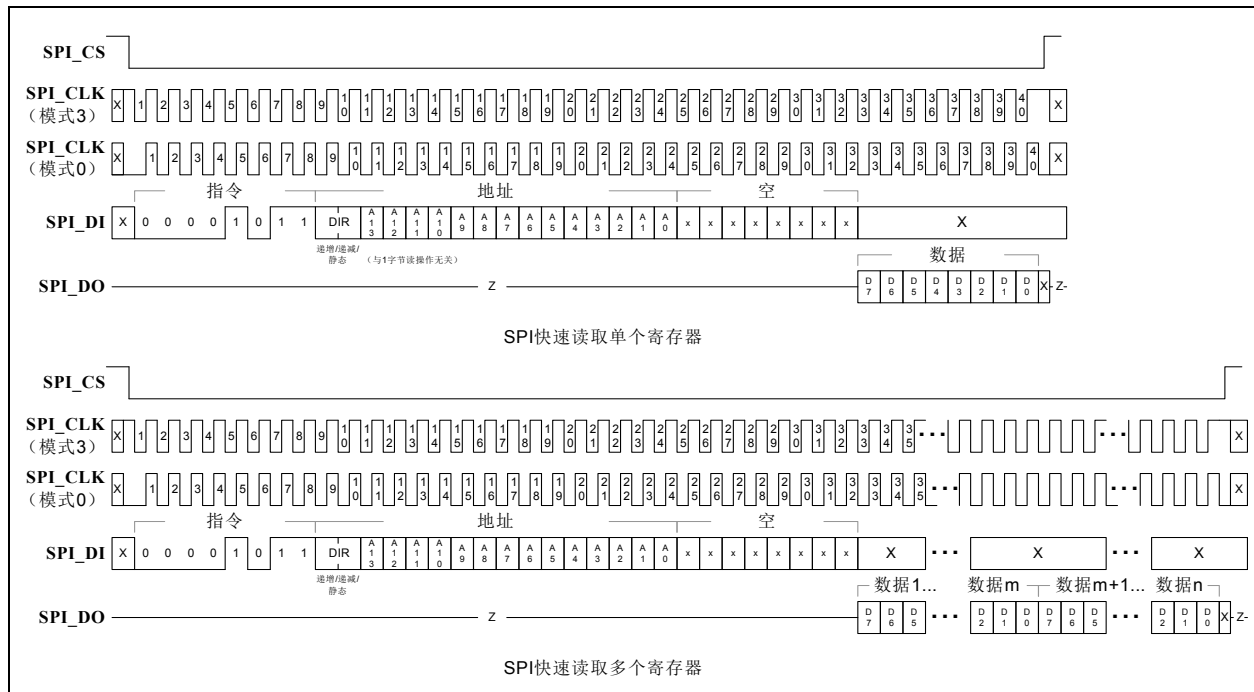
除第一次字节读操作之外的其他字节读操作由主器件在SPI_CS有效时继续提供时钟脉冲的方式来执行。对于连续字节的读操作，地址的高2位bit[15:14]（DIR子字段）指定自动递增（DIR = 00b）、自动递减（DIR = 10b）或静态（固定）（DIR = 11b）。如果主器件支持，可为单个帧内的FIFO读/写或低级寄存器轮询操作维持静态内部地址。

在当前单字节输出移位即将结束时，地址会递增或递减（合适时），并且另一个同步捕捉序列将完成。

6.2.3.2 快速读取

图6-2给出了SPI模式下典型的单次和多次寄存器快速读取操作。

图6-2: SPI快速读取



6.2.4 SPI写命令

SPI从控制器支持以下写命令:

- 写
- 多次写

6.2.4.1 写

写指令在SPI_DI引脚上提供指令代码、地址和数据字节，每个时钟一位。

SPI传送由主器件通过先将SPI_CS驱动为有效状态的方式启动。SPI_DI引脚中先输入8位WRITE指令02h，再输入两个地址字节。地址字节指定器件内的一个字节地址，以及一个方向控制子字段（DIR）。

SPI_DI引脚中输入地址字节后紧接着会输入数据，从第一个字节的最高有效位开始。数据由器件从SPI_DI引脚输入（在随后的每个时钟上升沿移入）。

6.2.4.2 多次写

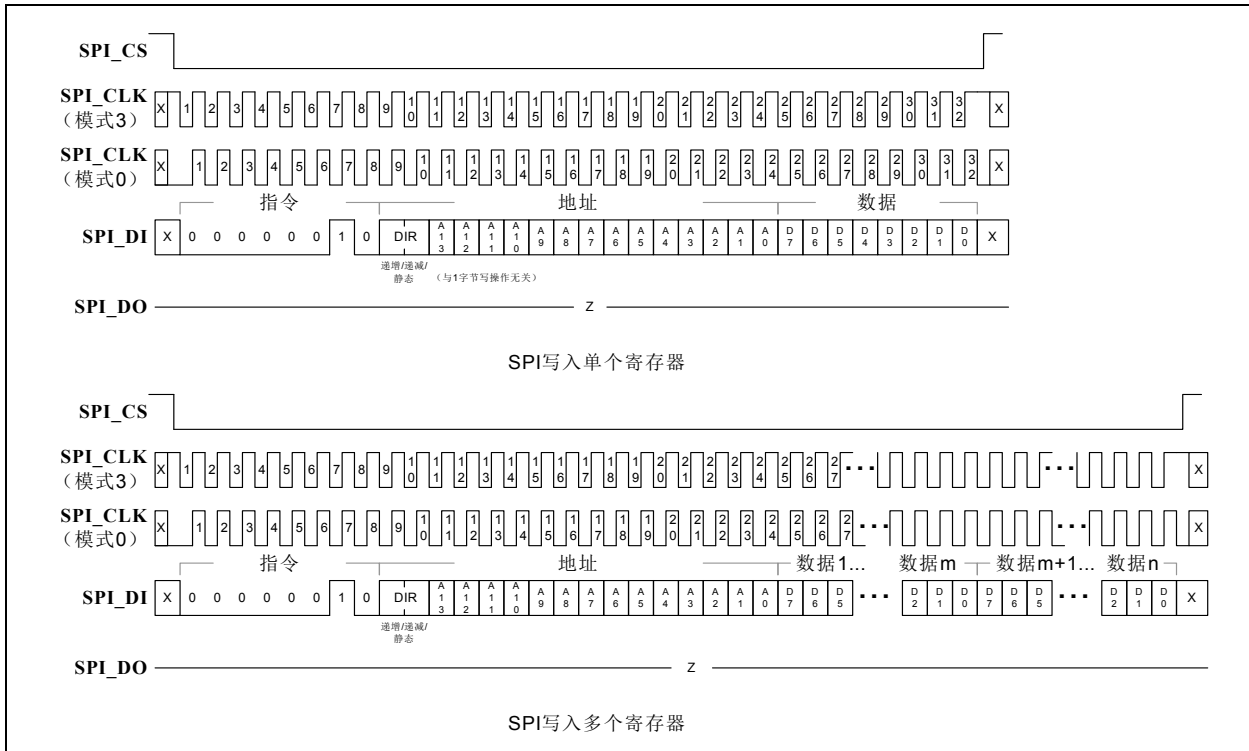
多次写操作由主器件在SPI_CS保持有效时继续提供时钟脉冲和SPI_DI数据的方式来执行。地址的高2位bit[15:14]构成DIR子字段，用于指定自动递增（DIR = 00b）、自动递减（DIR = 10b）或静态寻址。内部字节地址基于这些位递增、递减或保持固定（静态）。保持一个固定的内部地址对FIFO访问、寄存器“位拆裂”或其他重复活动十分有用。

寄存器的数据写操作在整个寄存器的内容输入之后发生：这取决于寄存器定义的大小。如果SPI_CS恢复高电平时未写入整个寄存器，则会认为写操作无效，寄存器不受影响。

SPI_CS输入随后变为无效以结束周期。

图6-3给出了典型的SPI单次和多次寄存器写操作。

图6-3: SPI写



6.2.5 特殊的CSR处理

6.2.5.1 实时位

寄存器值在每次寄存器读取开始时锁存（寄存），以防止主机读取一个变化的值。在多次寄存器读取序列中，每个寄存器单独进行锁存。

6.2.5.2 读取时变化的寄存器和FIFO

因读操作触发副作用的任何单字节寄存器（例如，包含“读取时清零”位或推进FIFO结构）仅在主机开始访问相应值后触发。主器件获知的值始终为原始值，而永远不是因副作用更新后的结果。

对于被视为单个单元的多字节寄存器，更改可能会延迟，直至读取完寄存器的所有字节。如果主机在多字节寄存器的前几个字节之一上发送无应答信号，或者在应答整个寄存器之前意外出现开始或停止条件，则相应的读操作可能被视为无效，不会触发副作用。

采用多寄存器读取访问的寄存器读操作将触发多个副作用（在读取时发生）。以下寄存器具有读取副作用：

- 看门狗计数寄存器（WDT_COUNT）

6.2.5.3 读取时变化的实时位

如上所述，具有实时位的寄存器中的当前值（像任何寄存器一样）作为输出数据进行捕捉和锁存，随后原始寄存器中的读取时变化位将发生改变。为防止数据捕捉之后、“读取时变化”之前发生的硬件事件丢失，这些硬件事件将保持等待状态，直到读操作之后以及因读取而发生任何变化之后才结束。该序列还可确保发生硬件事件时位的边沿。

7.0 时钟、复位和功耗管理

本节详细介绍了器件的各种时钟、复位和功耗管理状态：

7.1 时钟

器件生成以下内部时钟：

- 48 MHz 张弛振荡器
- 20 kHz 保活振荡器
- 环形振荡器

这些振荡器可通过[时钟控制寄存器 \(CLK_CTL\)](#) 手动使能/禁止。

7.2 电源状态

该器件支持四种电源状态，具体在以下小节中进行了定义：

- 休眠
- 待机
- ACTIVE_PPC_OFF
- ACTIVE_PPC_ON

7.2.1 休眠

这是器件的最低功耗状态。将 **PWR_DN** 引脚置为有效即可进入休眠状态。在这种模式下，几乎器件的所有部分都掉电，只保留 3.3V 域内的最基本电路来检测 **PWR_DN** 是否置为无效。

该模式旨在最大限度地减少器件不在电池供电类应用中使用时的功耗。在这些应用中，唤醒事件（如按钮按下）可能会导致主机 CPU 将 **PWR_DN** 置为无效。

7.2.2 待机

待机是器件的最低功耗工作状态。在这种状态下，器件的大部分均掉电。内部 CC 比较器和 20 kHz 振荡器以及必要的模拟组件（1.8V LDO、POR 和偏置等）在该状态下使能。

它们将不断监视 CC 线路以了解连接状态，连接状态会导致主机中断置为有效。如果已建立连接，则该状态会检测到伙伴通告发生变化以及连接断开。

待机是 USB Type-C Unattached.SRC/SNK 状态时 UPD360 器件所处的功耗状态。

该状态下支持以下异步唤醒：

- 异步 I2C 唤醒（仅限 UPD360-A/UPD360-B）
- 异步 SPI 唤醒（仅限 UPD360-C）
- 供电 MAC 唤醒

7.2.3 ACTIVE_PPC_OFF

该状态定义了内部 48 MHz 振荡器使能，但端口电源控制器禁止的情况。该状态的一个用例是希望在一微秒的时间内为一个 GPIO 去抖的情况。在这种情况下，需要通过内部 48 MHz 振荡器来为去抖过程定时。

当通过软件 I²C/SPI 访问寻址或要实现供电通信时，器件也会暂时进入该状态。

UPD360 主要将在 USB Type-C Attached.SNK 状态或 Attached.SRC 状态时处于 ACTIVE_PPC_OFF 状态，此时 VBUS 由外部电源驱动。

7.2.4 ACTIVE_PPC_ON

该状态定义了端口电源控制器正常工作（此时需要为端口电源控制器使能内部 48 MHz 振荡器）的情况。

UPD360

UPD360 将在 USB Type-C Attached.SRC 状态时处于 ACTIVE_PPC_ON 状态，此时 VBUS 由内部 5V 端口电源控制器驱动。

7.3 异步 I²C 唤醒（仅限 UPD360-A/UPD360-B）

该器件支持 I²C 从接口的异步唤醒。凭借时钟延长，触发唤醒的 I²C 事务不会丢失，不必由主机重复发起。该器件的时钟延长不会超过 3 μ s。

以下步骤说明了 I²C 唤醒功能。最初时，环形振荡器和 48 MHz 振荡器处于禁止状态。

1. 主机向器件发起一个 I²C 事务。
2. 器件异步检测启动位的接收，然后在主机将 SCL 驱动为低电平后拉低 I2C_CLK 以使能时钟延长。环形振荡器异步使能，并用作功耗管理逻辑的时钟源。
3. 在大约 5 μ s 的延时之后，振荡器达到稳定状态并为 I²C 控制器提供时钟驱动。
4. 时钟延长禁止，I²C 控制器使能并开始处理待处理的事务。
5. I²C 事务完成。
6. 主机检查器件状态以了解是否有任何未处理的事务。I²C 事务可能已经发起了 PD 传送，或者相反，可能正在接收一致的 PD 事务。
7. 主机确认器件没有待处理的事务后，将通过时钟控制寄存器（CLK_CTL）禁止环形振荡器和 48 MHz 张弛振荡器，以使器件掉电。
8. 器件可随时接收未来的异步 I²C 唤醒事件。

7.4 异步 SPI 唤醒（仅限 UPD360-C）

UPD360 支持 SPI 接口的异步唤醒。该器件的 SPI 协议定义不要求必须重复发起 SPI 事务。

以下步骤说明了 SPI 唤醒功能。最初时，环形振荡器和 48 MHz 振荡器处于禁止状态。

1. 器件掉电。
2. 主机向 SPI 测试寄存器（SPI_TEST）（仅限 UPD360-C）发起一个 SPI 事务，如果器件正在运行，将返回一个非零值。如果器件掉电，则会将 SPI_DO 驱动为 0b。
3. 器件检测 SPI 报文的接收。环形振荡器异步使能，并用作功耗管理逻辑的时钟源。
4. 在大约 5 μ s 的延时之后，振荡器达到稳定状态并为 SPI 控制器提供时钟驱动。
5. 器件处理接收到的下一个 SPI 事务。
6. SPI 事务完成。
7. 主机检查器件状态以了解是否有任何未处理的事务。SPI 事务可能已经发起了 PD 传送，或者相反，可能正在接收一致的 PD 事务。
8. 主机确认器件没有待处理的事务后，将通过时钟控制寄存器（CLK_CTL）禁止环形振荡器和 48 MHz 张弛振荡器，以使器件掉电。
9. 器件将 SPI_DO 驱动为 0b，并等待异步 SPI 唤醒。

7.5 供电MAC唤醒

PD MAC能够在接收到PD数据包时异步唤醒。这会器件置于待机模式，从而最大限度地降低功耗。

以下步骤说明了RX PD MAC唤醒功能。最初时，环形振荡器和48 MHz张弛振荡器处于禁止状态。20 kHz保活振荡器使能，但未在唤醒过程中使用。

1. 要接收PD报文，需通过软件使能RX AFE，并通过CC RX DAC控制寄存器（CC_RX_DAC_CTL）和CC RX DAC过滤器寄存器（CC_RX_DAC_FILT）设置跳变点。
2. PD MAC通过软件配置和使能。
3. 器件掉电。软件通过时钟控制寄存器（CLK_CTL）禁止48 MHz张弛振荡器和环形振荡器（如果使能）。
4. 经过一段时间后，器件会从连接的伙伴处接收到PD报文。
5. PD MAC异步检测CC线路上的前导码活动，并使能48 MHz张弛振荡器。
6. 该振荡器运行大约5 μ s，此时PD MAC也在运行。
7. PD MAC接收前导码的剩余部分并将报文存储到RX FIFO中，然后根据PD协议在需要时用GoodCRC进行响应。
8. 通过将IRQ_N引脚置为有效来发出中断。
9. 软件通过I²C处理中断、读取PD报文并根据需要进行响应。
10. 器件随后掉电。软件通过时钟控制寄存器（CLK_CTL）禁止48 MHz张弛振荡器和环形振荡器。
11. 器件保持掉电，直到接收到下一条PD报文。

7.6 待机模式下的中断有效

当器件处于待机模式时，它能够检测到一些事件，这些事件经软件适当编程可配置为将IRQ_N引脚置为有效。检测此类事件的逻辑基于20 kHz保活振荡器工作。

编程为将IRQ_N置为有效的事件发生后，48 MHz张弛振荡器和环形振荡器立即使能。完成同步后，IRQ_N引脚应置为有效。

在软件处理源中断之后，它应通过时钟控制寄存器（CLK_CTL）禁止48 MHz张弛振荡器和环形振荡器，使器件恢复待机状态。

以下示例序列说明了为检测待机状态下OCS_COMP1引脚上的OCS事件而需遵循的器件配置步骤。

1. 软件使能中断允许寄存器（INT_EN）的OCS_CMP_INT位。
2. 软件使能OCS检测（如第8.3节“外部过流检测”中所定义），以采样OCS_COMP1引脚。
3. 软件通过时钟控制寄存器（CLK_CTL）禁止48 MHz张弛振荡器和环形振荡器。
4. OCS事件发生并在OCS_COMP1上检测到。
5. 环形振荡器使能并用作器件的运行时钟。
6. 器件使能48 MHz振荡器，并等待大约5 μ s，以使振荡器达到稳定状态。
7. 48 MHz振荡器处于稳定状态。
8. 完成同步后，IRQ_N引脚置为有效。
9. 软件检测IRQ_N有效状态并处理中断。
10. 软件通过时钟控制寄存器（CLK_CTL）禁止48 MHz张弛振荡器和环形振荡器，使器件恢复待机状态。

7.7 复位操作

器件支持以下芯片级复位：

- 上电复位（Power-On-Reset, POR）
- 引脚复位（RESET_N）
- 软件复位（[硬件控制寄存器（HW_CTL）](#)中的SRESET）
- 看门狗定时器（WDT_STS）

芯片级复位触发CFG_SEL0和CFG_SEL1配置脚的采样（更多信息，请参见第81页的第9.8.1节“配置选择”）。芯片级复位是否完成可以通过[中断允许寄存器（INT_EN）](#)中的RDY_INT位以及IRQ_N引脚的有效状态来确定。

以下内容总结了芯片级复位后的步骤。

1. 系统级复位事件（POR、RESET_N、SRESET和WDT_STS）发生。
2. 器件使能20 kHz保活振荡器、48 MHz张弛振荡器和1 MHz环形振荡器。
3. 器件采样CFG_SELx引脚。
4. 器件根据CFG_SELx引脚和设置自行配置。
5. 器件使能，[中断允许寄存器（INT_EN）](#)中的RDY_INT位置为有效，IRQ_N引脚置为有效。
6. 器件禁止48 MHz张弛振荡器和环形振荡器。

7.8 时钟和功耗管理寄存器

本节详细介绍了时钟和功耗管理寄存器。有关器件的完整寄存器映射的概述，请参见第19页的第4.0节“寄存器映射”。

表7-1： 时钟和功耗管理寄存器映射

地址	寄存器名称（符号）
1000h	时钟控制寄存器（CLK_CTL）
1001h–13FFh	保留供将来扩展使用

注： 在任何情况下都不能对保留的地址空间执行写操作。否则，可能导致意外的操作与结果。

7.8.1 时钟控制寄存器（CLK_CTL）

地址： 1000h 大小： 8位

Bit	说明	类型	默认值
7:3	保留	RO	-
2	环形振荡器使能 0：禁止振荡器。 1：使能振荡器。	R/W	0b
1	48 MHz张弛振荡器使能 0：禁止振荡器。 1：使能振荡器。	R/W	注7-1
0	20 kHz保活振荡器使能 0：禁止振荡器。 1：使能振荡器。	R/W	1b

注7-1 该位在通过软件读取时始终显示为有效，因为必须使能48 MHz张弛振荡器以方便进行I²C/SPI事务。

8.0 系统控制

本节详细介绍了以下系统控制：

- [系统中断](#)
- [通用 I/O](#)
- [外部过流检测](#)
- [系统控制寄存器](#)

8.1 系统中断

该器件提供了一个漏极开路中断引脚 **IRQ_N**，由系统控制的 [中断状态寄存器 \(INT_STS\)](#) 和 [中断允许寄存器 \(INT_EN\)](#) 共同置为有效。支持下列中断：

- HPD 中断
- 电源开关模块中断
- 看门狗定时器置为有效中断
- 端口电源控制器中断
- 供电 MAC 中断
- OCS 比较器中断
- PIO 改写中断
- PIO 中断
- CC 中断
- VBUS 中断
- 电源控制中断
- 器件就绪中断

要将中断状态位清零，必须将 [中断状态寄存器 \(INT_STS\)](#) 中置为有效的位对应的中断源清除。这一过程通常涉及处理触发中断的模块（例如，PD MAC、CC、端口电源控制器、看门狗定时器、电源开关等）或配置相应模块的中断允许寄存器来禁止有效状态。

通过向 [PIO 中断状态寄存器 \(PIO_INT_STS\)](#) 中置为有效的位写入“1”来清除 PIO 中断。通过向 [PIO 改写中断状态寄存器 \(PIO_OVR_INT_STS\)](#) 中置为有效的位写入“1”来清除 PIO 改写中断。通过向 [OCS 比较器更改状态寄存器 \(OCS_CMP_CHG_STS\)](#) 中置为有效的位写入“1”来清除 OCS 中断。

[图 8-1](#) 说明了器件的中断层级。[图 8-2](#) 说明了 [中断状态寄存器 \(INT_STS\)](#) 中 **CC_INT** 位的产生。

图8-1： 系统中断层级

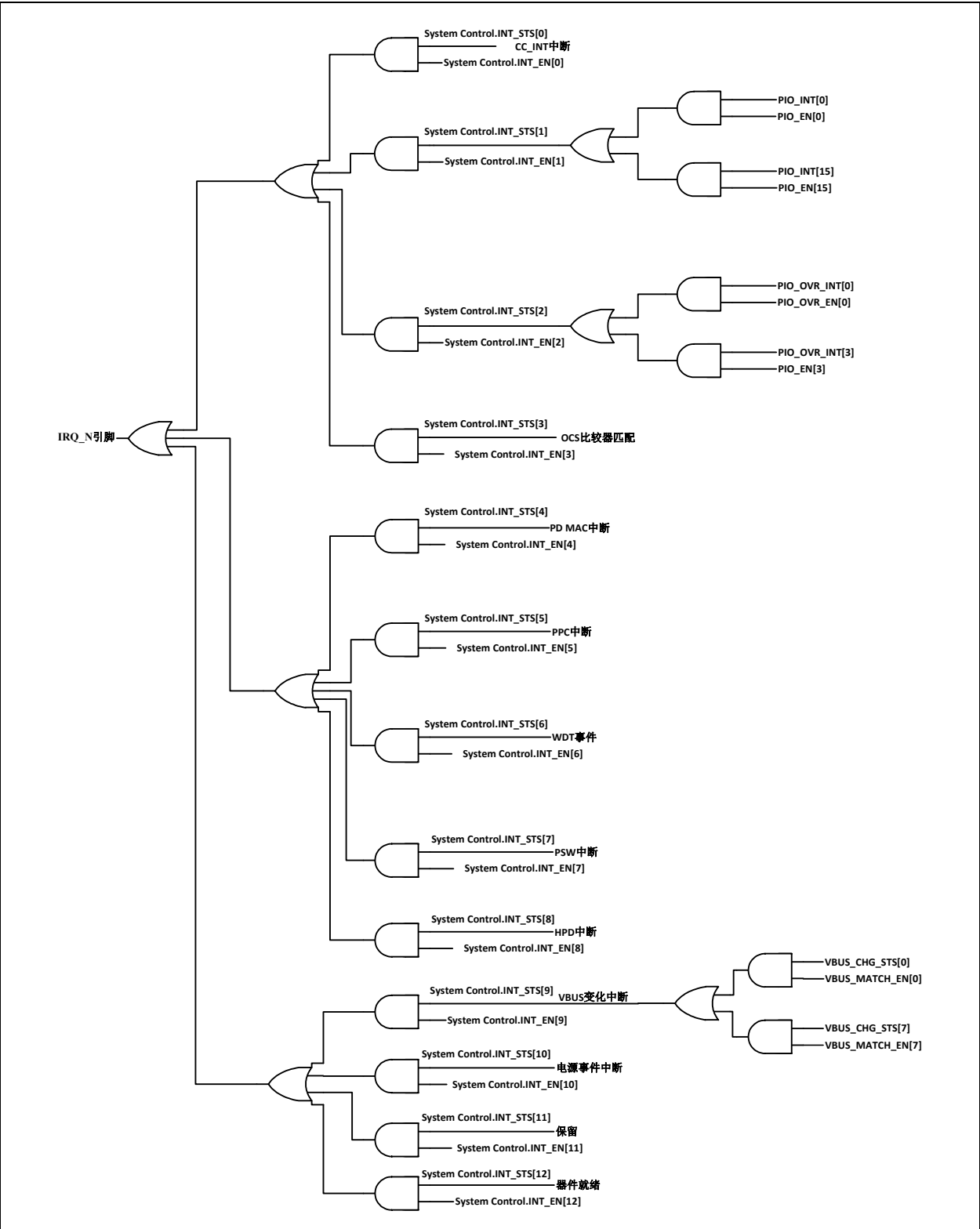
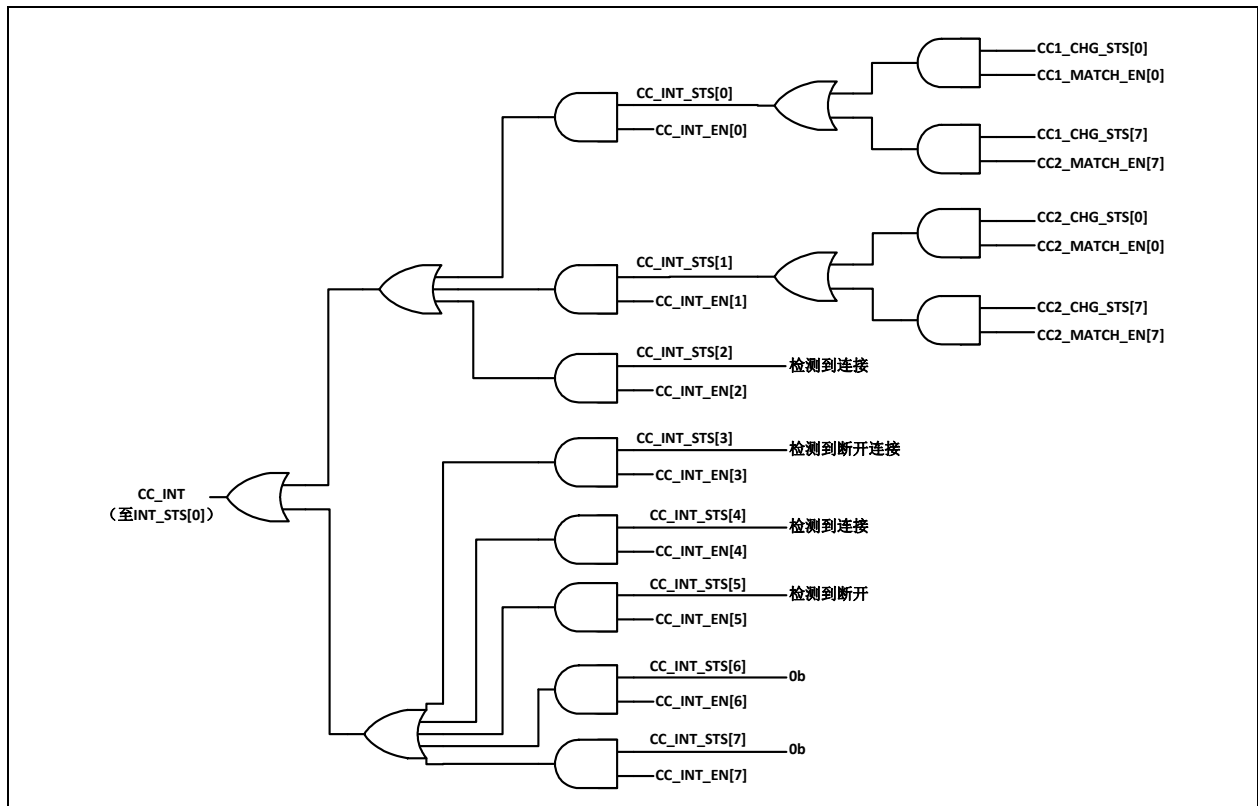


图8-2: CC_INT中断层级



8.2 通用I/O

UPD360的一个关键功能是通过多达十六个PIO管理外部器件。这通常是通过主机软件编程PIO来实现。

在某些情况下，UPD360必须自动改写PIO状态以响应错误条件。为支持此类操作，设计中融合了相关特性。如果依赖软件引入以影响PIO状态的延时太长或不确定，则必须满足这一点。

通过相应配置PIOx寄存器（CFG_PIOx）配置PIO。另外，也可通过配置PIO将中断状态寄存器（INT_STS）中的PIO_INT置为有效。PIO的状态通过读取PIO状态寄存器（PIO_STS）来获知。

有一种机制可通过单次寄存器写操作更新多个PIO输出。这是通过配置PIO输出寄存器（CFG_PIO_OUT）来实现的。如果对该寄存器的写操作将更改PIO的输出值，则会反映在相应配置PIOx寄存器（CFG_PIOx）的DataOutput位中。同样，写入配置PIOx寄存器（CFG_PIOx）中DataOutput将更新配置PIO输出寄存器（CFG_PIO_OUT）中的相应位。

注： 通用信号GPO1和GPO10只能用作输出，且在将RESET_N置为无效时必须保持在低电平状态。

注： UPD360-C中未提供GPIO0。

注： 在独立的DFP/UFP模式下（仅限UPD360-A/UPD360-B），GPIO 9-15具有备用的专用功能，如第11页的第3.1.1.1节“独立DFP/UFP模式下的UPD360-A/UPD360-B GPIO[9:15]功能”所定义。

8.2.1 PIO 同时更新

为能够同时更新多个PIO，引入了一种机制。这对于简化软件、防止UPD360控制的器件进入未知的中间状态是可取的。负责管理电源的器件尤其值得关注。

配置PIOx寄存器（CFG_PIOx） CSR通过此功能更新。当硬件接受同时更新请求时，会同时更新PIO寄存器的内容。

以下示例说明了同时PIO更新操作：

1. 软件根据需要更改**配置PIOx更新寄存器（CFG_PIOx_UPD）**。
2. 软件根据需要更改**配置PIO输出更新寄存器（CFG_PIO_OUT_UPD）**。

注： **配置PIO输出更新寄存器（CFG_PIO_OUT_UPD）**和**配置PIOx更新寄存器（CFG_PIOx_UPD）**必须通过软件一致编程。如果这两个寄存器的输出值位不同，则**配置PIOx更新寄存器（CFG_PIOx_UPD）**中的值将用于更新**配置PIOx寄存器（CFG_PIOx）**以及**配置PIO输出寄存器（CFG_PIO_OUT）**中的相应位。

3. 软件设置**配置PIOx更新寄存器（CFG_PIOx_UPD）**以启动更新。
4. 待处理的PIO改写将阻止接受更新请求。只有待处理的PIO改写功能清除后，器件才会将**PIO_UPD_REQ**清零。待处理的PIO改写通过**PIO改写使能寄存器（PIO_OVR_EN）**清除。
5. 接受更新后，**PIO_UPD_REQ**将自动清零。

应用注意事项： 如果PIO改写待处理，则将阻止PIO同时更新。

8.2.2 PIO 改写

8.2.2.1 概述

本节定义了一种机制，可以在没有软件干预的情况下更改PIO的状态。这提供了一种机制来瞬间禁止电源电路，具体将由UPD360 PIO基于所选PIO输入的状态变化、OCS检测或其他事件进行控制。

8.2.2.2 说明

器件中包含四个改写功能。对于每个功能，必须通过相应**PIO改写x源选择寄存器（PIO_OVRx_SRC_SEL）**选择一个PIO或其他源进行监视。要监视的值在**PIO改写监视值寄存器（PIO_OVR_MON_VAL）**中设置并与相应PIO的状态进行比较。当这些值与改写匹配时，使能的PIO输出相应地更改其状态。

由改写控制的PIO输出通过**PIO改写x输出使能寄存器（PIO_OVRx_OUT_EN）**使能。每个改写在該寄存器中都有一个实例。一个PIO可以用于多个改写。

如果**中断允许寄存器（INT_EN）**适当配置，则固件可获知已通过将**中断状态寄存器（INT_STS）**中的**PIO_OVERRIDE_INT**置为有效执行改写功能。

8.3 外部过流检测

器件集成了一个模拟比较器DAC电路来检测过流情况。该功能通过OCS_COMP1和/或OCS_COMP2引脚来支持。该功能通过以下方式使能：

1. 软件将OCS比较器控制寄存器（OCS_CMP_CTL）中的OCS比较器控制位清零，以确定OCS去抖动器。
2. 软件轮询OCS_DB_ACTIVE直至其清零（表示OCS去抖动器被禁止）。
3. 如果需要使用OCS_COMP2，则必须通过配置PIOx寄存器（CFG_PIOx）禁止共用PIO。
4. 软件根据需要编程OCS比较器x阈值寄存器（OCS_CMPx_THR）中的一个或两个。
5. 软件编程OCS比较器去抖寄存器（OCS_CMP_DEBOUNCE）。
6. 软件通过OCS比较器匹配使能寄存器（OCS_CMP_MATCH_EN）使能OCS_COMPx引脚。
7. 软件在中断允许寄存器（INT_EN）中允许相应的中断。
8. 软件设置OCS比较器控制寄存器（OCS_CMP_CTL）中的OCS比较器控制位，以根据需要使能OCS_COMP1和/或OCS_COMP2的比较操作。
9. 通过比较器检测过流事件将导致OCS比较器更改状态寄存器（OCS_CMP_CHG_STS）中的相应位以及OCS_CMP_INT置为有效。IRQ_N引脚将置为有效（如果已配置）。
10. 软件写入OCS比较器更改状态寄存器（OCS_CMP_CHG_STS）以将OCS_CMP_INT清零。从低电平转换为高电平时，OCS比较器匹配寄存器（OCS_CMP_MATCH）中的相应位将置1；而从高电平转换为低电平时，相应位将清零。

当软件将OCS比较器控制寄存器（OCS_CMP_CTL）中的OCS比较器控制位清零时，将禁止该功能。任一OCS_COMPx引脚均可用于触发一个PIO改写功能。

8.4 通用定时器

该器件集成了一个低功耗通用定时器，其依靠20 kHz振荡器工作，并实现了一个16位单次递减计数器。当定时器下溢时，它会将通用定时器中断源寄存器（GP_TIMER_INT_SRC）中的GP_UFLOW位置为有效并停止计数。这反过来会将中断状态寄存器（INT_STS）中的GP_TIMER_INT置为有效。该中断一直持续到GP_UFLOW位清零。如果中断允许寄存器（INT_EN）中的相应位置1，则IRQ_N引脚将置为有效。

定时器的上限阈值通过通用定时器装载寄存器（GP_TIMER_LOAD）装入。将通用定时器控制寄存器（GP_TIMER_CTL）中的RESET置1将导致定时器的内部计数器装入该值。

定时器通过将通用定时器控制寄存器（GP_TIMER_CTL）中的ENABLE置1来使能。这导致定时器每1 ms递减一次。定时器将继续递减，直至下溢或禁止。

注： 下溢条件定义为递减计数器从0x0000转换为0xFFFF。

将通用定时器控制寄存器（GP_TIMER_CTL）中的ENABLE位清零会禁止定时器，使其停止计数。

定时器的当前计数可从通用定时器计数寄存器（GP_TIMER_COUNT）读取。读取计数之前，应通过将ENABLE清零暂停定时器。读取计数之后，可再次将ENABLE置1，使计数器恢复工作。

通用定时器控制寄存器（GP_TIMER_CTL）中的DIS_ON_UFLOW位将在检测到下溢后自动禁止定时器并将GP_UFLOW置为有效。在这种情况下，将GP_UFLOW置为有效的操作将延迟到定时器完全禁止后进行。

应用注意事项： 该定时器允许在主机CPU中禁止时钟源，这对于DRP等定时事件十分有用，有助于提供总功耗较低的系统解决方案。

以下步骤说明了DRP实现的使用模型：

1. 软件将器件配置为DFP或UFP。更多详细信息，请参见第9.4节“DRP操作”。
2. 软件使能中断允许寄存器（INT_EN）中的GP_TIMER_INT。

3. 软件将通用定时器装载寄存器 (GP_TIMER_LOAD) 编程为符合 USB Type-C 规范中定义的 tDRP 周期范围的随机值。
4. 软件将通用定时器控制寄存器 (GP_TIMER_CTL) 中的 ENABLE、RESET 和 DIS_ON_UFLOW 位置 1。这样会将通用定时器装载寄存器 (GP_TIMER_LOAD) 中的值装入定时器中，随后定时器开始递减计数。与此同时，还会将定时器配置为延迟至定时器完全禁止后将通用定时器中断源寄存器 (GP_TIMER_INT_SRC) 中的 GP_UFLOW 置为有效。

应用注意事项： 对通用定时器装载寄存器 (GP_TIMER_LOAD) 和通用定时器控制寄存器 (GP_TIMER_CTL) 的写操作（在之前的步骤中定义）可以在同一 I²C 写突发中完成。

5. 定时器超时后，GP_UFLOW 位将置 1，中断状态寄存器 (INT_STS) 中的 GP_TIMER_INT 将置为有效。
6. IRQ_N 引脚置为有效。
7. 软件处理 IRQ 并确定是否与 UFP 或 DFP 建立了连接。如果尚未建立连接，软件将更改器件的角色。更多详细信息，请参见第 9.4 节“DRP 操作”。
8. 软件将通用定时器装载寄存器 (GP_TIMER_LOAD) 编程为符合 USB Type-C 规范中定义的 tDRP 周期范围的随机值。
9. 软件会将通用定时器中断源寄存器 (GP_TIMER_INT_SRC) 中的 GP_UFLOW 位清零。
10. 软件将通用定时器控制寄存器 (GP_TIMER_CTL) 中的 ENABLE、RESET 和 DIS_ON_UFLOW 位置 1。这样会将通用定时器装载寄存器 (GP_TIMER_LOAD) 中的值装入定时器中，随后定时器开始递减计数。

应用注意事项： 对通用定时器装载寄存器 (GP_TIMER_LOAD)、通用定时器中断源寄存器 (GP_TIMER_INT_SRC) 和通用定时器控制寄存器 (GP_TIMER_CTL) 的写操作（在之前的步骤中定义）可以在同一 I²C 写突发中完成。

11. 定时器超时并被禁止后，GP_UFLOW 位将置 1，中断状态寄存器 (INT_STS) 中的 GP_TIMER_INT 将置为有效。
12. IRQ_N 引脚置为有效。
13. 软件处理 IRQ 并确定是否与 UFP 或 DFP 建立了连接。如果尚未建立连接，软件将更改器件的角色。更多详细信息，请参见第 9.4 节“DRP 操作”。
14. 转到步骤 8。

8.5 系统控制寄存器

本节详细介绍了系统控制寄存器。有关器件的完整寄存器映射的概述，请参见第19页的第4.0节“寄存器映射”。

表 8-1: 系统控制寄存器映射

地址	寄存器名称 (符号)
0000h	器件ID寄存器 (ID_REV)
0004h	USB 供应商ID寄存器 (VID)
0006h	USB 产品ID寄存器 (PID)
0008h	USB PD 版本寄存器 (PD_REV)
000Ah	USB Type-C™ 版本寄存器 (C_REV)
000Bh-000Dh	保留供将来扩展使用
000Eh	SPI 测试寄存器 (SPI_TEST) (仅限UPD360-C)
0010h	中断状态寄存器 (INT_STS)
0014h	中断允许寄存器 (INT_EN)
0018h	硬件控制寄存器 (HW_CTL)
001Ah	I2C 从地址寄存器 (I2C_ADDR) (仅限UPD360-A/UPD360-B)
001Bh	PPC 状态寄存器 (PPC_BUSY)
001Ch-001Fh	保留供将来扩展使用
0020h	PIO 状态寄存器 (PIO_STS)
0022h	PIO 中断状态寄存器 (PIO_INT_STS)
0024h	PIO 中断允许寄存器 (PIO_INT_EN)
0026h-002Fh	保留供将来扩展使用
0030h	配置PIOx寄存器 (CFG_PIOx) x=0
0031h	配置PIOx寄存器 (CFG_PIOx) x=1
0032h	配置PIOx寄存器 (CFG_PIOx) x=2
0033h	配置PIOx寄存器 (CFG_PIOx) x=3
0034h	配置PIOx寄存器 (CFG_PIOx) x=4
0035h	配置PIOx寄存器 (CFG_PIOx) x=5
0036h	配置PIOx寄存器 (CFG_PIOx) x=6
0037h	配置PIOx寄存器 (CFG_PIOx) x=7
0038h	配置PIOx寄存器 (CFG_PIOx) x=8
0039h	配置PIOx寄存器 (CFG_PIOx) x=9
003Ah	配置PIOx寄存器 (CFG_PIOx) x=10
003Bh	配置PIOx寄存器 (CFG_PIOx) x=11
003Ch	配置PIOx寄存器 (CFG_PIOx) x=12
003Dh	配置PIOx寄存器 (CFG_PIOx) x=13
003Eh	配置PIOx寄存器 (CFG_PIOx) x=14
003Fh	配置PIOx寄存器 (CFG_PIOx) x=15
0040h	配置PIOx更新寄存器 (CFG_PIOx_UPD) x=0
0041h	配置PIOx更新寄存器 (CFG_PIOx_UPD) x=1
0042h	配置PIOx更新寄存器 (CFG_PIOx_UPD) x=2
0043h	配置PIOx更新寄存器 (CFG_PIOx_UPD) x=3
0044h	配置PIOx更新寄存器 (CFG_PIOx_UPD) x=4
0045h	配置PIOx更新寄存器 (CFG_PIOx_UPD) x=5
0046h	配置PIOx更新寄存器 (CFG_PIOx_UPD) x=6
0047h	配置PIOx更新寄存器 (CFG_PIOx_UPD) x=7

UPD360

表 8-1: 系统控制寄存器映射 (续)

地址	寄存器名称 (符号)
0048h	配置PIOx更新寄存器 (CFG_PIOx_UPD) x=8
0049h	配置PIOx更新寄存器 (CFG_PIOx_UPD) x=9
004Ah	配置PIOx更新寄存器 (CFG_PIOx_UPD) x=10
004Bh	配置PIOx更新寄存器 (CFG_PIOx_UPD) x=11
004Ch	配置PIOx更新寄存器 (CFG_PIOx_UPD) x=12
004Dh	配置PIOx更新寄存器 (CFG_PIOx_UPD) x=13
004Eh	配置PIOx更新寄存器 (CFG_PIOx_UPD) x=14
004Fh	配置PIOx更新寄存器 (CFG_PIOx_UPD) x=15
0050h	PIO改写x源选择寄存器 (PIO_OVRx_SRC_SEL) x=0
0051h	PIO改写x源选择寄存器 (PIO_OVRx_SRC_SEL) x=1
0052h	PIO改写x源选择寄存器 (PIO_OVRx_SRC_SEL) x=2
0053h	PIO改写x源选择寄存器 (PIO_OVRx_SRC_SEL) x=3
0054h	PIO改写监视值寄存器 (PIO_OVR_MON_VAL)
0055h	PIO改写使能寄存器 (PIO_OVR_EN)
0056h	PIO改写状态寄存器 (PIO_OVR_STS)
0057h	PIO改写中断状态寄存器 (PIO_OVR_INT_STS)
0058h	PIO改写中断允许寄存器 (PIO_OVR_INT_EN)
0059h	配置PIO输出寄存器 (CFG_PIO_OUT)
005Bh	配置PIO输出更新寄存器 (CFG_PIO_OUT_UPD)
005Dh-005Fh	保留供将来扩展使用
0060h	PIO改写x输出使能寄存器 (PIO_OVRx_OUT_EN) x=0
0062h	PIO改写x输出使能寄存器 (PIO_OVRx_OUT_EN) x=1
0064h	PIO改写x输出使能寄存器 (PIO_OVRx_OUT_EN) x=2
0066h	PIO改写x输出使能寄存器 (PIO_OVRx_OUT_EN) x=3
0068h	PIO改写输出寄存器 (PIO_OVR_OUT)
006Ah	PIO改写方向寄存器 (PIO_OVR_DIR)
006Ch	PIO去抖计数十毫秒寄存器 (PIO_DEBOUNCE_10MS_COUNT)
006Dh	PIO去抖计数一毫秒寄存器 (PIO_DEBOUNCE_1MS_COUNT)
006Eh	PIO去抖计数一微秒寄存器 (PIO_DEBOUNCE_1US_COUNT)
006Fh	PIO去抖状态寄存器 (PIO_DEBOUNCE_STS)
0070h	PIO去抖使能寄存器 (PIO_DEBOUNCE_EN)
0074h-007Fh	保留供将来扩展使用
0080h	OCS比较器控制寄存器 (OCS_CMP_CTL)
0081h	OCS比较器匹配寄存器 (OCS_CMP_MATCH)
0082h	OCS比较器更改状态寄存器 (OCS_CMP_CHG_STS)
0083h	OCS比较器匹配使能寄存器 (OCS_CMP_MATCH_EN)
0084h	OCS比较器x阈值寄存器 (OCS_CMPx_THR) x=1
0086h	OCS比较器x阈值寄存器 (OCS_CMPx_THR) x=2
0088h	OCS比较器去抖寄存器 (OCS_CMP_DEBOUNCE)
0089h-009Ah	保留供将来扩展使用
009Bh	IRQ_N PU/PD控制寄存器 (IRQ_PUPD)
009Ch	稳压器控制寄存器 (VREG_CTL)
009Dh	I2C微调寄存器 (I2C_TRIM)
009E-009Fh	保留供将来扩展使用

表 8-1: 系统控制寄存器映射 (续)

地址	寄存器名称 (符号)
00A0h	通用定时器装载寄存器 (GP_TIMER_LOAD)
00A2h	通用定时器中断源寄存器 (GP_TIMER_INT_SRC)
00A3h	通用定时器控制寄存器 (GP_TIMER_CTL)
00A4h	通用定时器计数寄存器 (GP_TIMER_COUNT)
00A6h–03FFh	保留供将来扩展使用

注: 在任何情况下都不能对保留的地址空间执行写操作。否则, 可能导致意外的操作与结果。

8.5.1 器件ID寄存器 (ID_REV)

地址: 0000h 大小: 32位

Bit	说明	类型	默认值
31:16	器件ID (ID)	RO	0360h
15:0	器件版本 (REV)	RO	注 8-1

注 8-1 该字段的默认值取决于器件的芯片版本。

ID_REV[7:0] = 00h

ID_REV[15:8] = 01h

ID_REV[23:16] = 02h

ID_REV[31:24] = 03h

8.5.2 USB 供应商ID寄存器 (VID)

地址: 0004h 大小: 16位

Bit	说明	类型	默认值
15:0	USB 供应商标识 (VID)	R/W	0424h

VID[7:0] = 04h

VID[15:8] = 05h

8.5.3 USB 产品ID寄存器 (PID)

地址: 0006h 大小: 16位

Bit	说明	类型	默认值
15:0	USB 产品标识 (PID)	R/W	0360h

PID[7:0] = 06h

PID[15:8] = 07h

8.5.4 USB PD 版本寄存器 (PD_REV)

地址: 0008h 大小: 16 位

Bit	说明	类型	默认值
15:0	USB 供电规范版本 (PD_REV)	R/W	2013h

PD_REV[7:0] = 08h

PD_REV[15:8] = 09h

8.5.5 USB TYPE-C™ 版本寄存器 (C_REV)

地址: 000Ah 大小: 16 位

Bit	说明	类型	默认值
15:0	USB Type-C™ 规范版本 (C_REV)	R/W	0011h

C_REV[7:0] = 0Ah

C_REV[15:8] = 0Bh

8.5.6 SPI 测试寄存器 (SPI_TEST) (仅限 UPD360-C)

地址: 000Eh 大小: 8 位

Bit	说明	类型	默认值
7:0	SPI 测试 (SPI_TEST) 通过 SPI 接口唤醒芯片时，主机使用该寄存器来确定芯片何时脱离掉电状态。	RO	FDh

8.5.7 中断状态寄存器 (INT_STS)

地址: 0010h 大小: 16位

Bit	说明	类型	默认值
15	保留	RO	0h
14	GP_TIMER_INT 通用定时器下溢时产生的中断。 该中断反映通用定时器中断源寄存器 (GP_TIMER_INT_SRC) 的 GP_UFLOW 位的值。	RO	0b
13	保留	RO	0b
12	RDY_INT 在发生系统级复位事件后，器件完成配置且准备好供主机CPU访问时产生的中断。 该中断反映硬件控制寄存器 (HW_CTL) 的 DEV_READY 位的值。	RO	0b
11	保留	RO	-
10	PWR_INT 电源事件产生的中断。 该中断通过电源中断状态寄存器 (PWR_INT_STS) 清除。	RO	0b
9	VBUS_INT 表示VBUS更改状态寄存器 (VBUS_CHG_STS) 的状态发生了变化。	RO	0b
8	HPD_INT HPD检测逻辑产生的中断。 该中断通过HPD中断状态寄存器 (HPD_INT_STS) 清除。	RO	0b
7	PSW_INT 电源开关产生的中断。 该中断通过电源开关中断状态寄存器 (PWR_INT_STS) 清除。	RO	0b
6	WDT_INT 看门狗定时器超时产生的中断。 该中断通过看门狗控制寄存器 (WDT_CTL) 中的 WDT_STS 位清除。 注： 当有看门狗中断待处理时，将阻止所有写操作，直至看门狗控制寄存器 (WDT_CTL) 中的 WDT_STS 位清零。	RO	0b
5	PPC_INT PPC产生的中断。 该中断通过PPC中断状态寄存器1 (PPC_INT_STS1) 清除。	RO	0b

UPD360

Bit	说明	类型	默认值
4	MAC_INT PD MAC产生的中断。 该中断通过相应的PD MAC中断寄存器清除，详见第153页的第11.4.3节“PD MAC中断状态和允许寄存器”。	RO	0b
3	OCS_CMP_INT 因检测到过流条件而产生的中断。 该中断通过OCS比较器更改状态寄存器（OCS_CMP_CHG_STS）清除。	RO	0b
2	PIO_OVERRIDE_INT PIO改写产生的中断。 该中断通过PIO改写中断状态寄存器（PIO_OVR_INT_STS）清除。	RO	0b
1	PIO_INT 因配置的PIO状态发生变化而产生的中断。 该中断通过PIO中断状态寄存器（PIO_INT_STS）清除。	RO	0b
0	CC_INT 线缆插头方向和检测模块产生的中断。 该中断通过CC中断状态寄存器（CC_INT_STS）清除。	RO	0b

INT_STS[7:0] = 10h

INT_STS[15:8] = 11h

8.5.8 中断允许寄存器 (INT_EN)

地址: 0014h 大小: 16 位

Bit	说明	类型	默认值
15	保留	RO	0h
14	中断允许13 为“0”时，可阻止产生相应的中断。	R/W	0b
13	保留	RO	-
12	中断允许12 为“0”时，可阻止产生相应的中断。	R/W	1b
11	保留	RO	-
10	中断允许10 为“0”时，可阻止产生相应的中断。	R/W	0b
9	中断允许9 为“0”时，可阻止产生相应的中断。	R/W	0b
8	中断允许8 为“0”时，可阻止产生相应的中断。	R/W	0b
7	中断允许7 为“0”时，可阻止产生相应的中断。	R/W	0b
6	中断允许6 为“0”时，可阻止产生相应的中断。	R/W	1b
5	中断允许5 为“0”时，可阻止产生相应的中断。	R/W	0b
4	中断允许4 为“0”时，可阻止产生相应的中断。	R/W	0b
3	中断允许3 为“0”时，可阻止产生相应的中断。	R/W	0b
2	中断允许2 为“0”时，可阻止产生相应的中断。	R/W	0b
1	中断允许1 为“0”时，可阻止产生相应的中断。	R/W	0b
0	中断允许0 为“0”时，可阻止产生相应的中断。	R/W	0b

INT_EN[7:0] = 12h

INT_EN[15:8] = 13h

UPD360

8.5.9 硬件控制寄存器（HW_CTL）

地址：0018h 大小：16位

Bit	说明	类型	默认值
15:6	保留	RO	-
5	I2C_GEN_CALL_EN 0：不支持I ² C广播呼叫地址。 1：支持I ² C广播呼叫地址。	R/W	0b
4	保留	RO	-
3	I2C FMPLUS 该位置1时，将I ² C接口配置为在增强型快速模式下工作。 注： 仅限UPD360-A/UPD360-B。	R/W	1b
2	DEV_READY 该位置1时，表示在系统级复位事件后，器件已完全配置。	R/SC	0b
1	PIO_UPD_REQ 该位置1时，PIO寄存器通过相应的同时更新寄存器更新。 操作完成后，该位自动清零。 如果有改写功能待处理，请求将不被接受。改写条件被软件清除后，更新请求将被接受。改写通过PIO改写使能寄存器（PIO_OVR_EN）的相应位清除。	R/SC	0b
0	SRESET 软件复位。该位置1时，将发起芯片级复位。	SC	0b

8.5.10 I²C从地址寄存器（I2C_ADDR）（仅限UPD360-A/UPD360-B）

地址：001Ah 大小：8位

Bit	说明	类型	默认值
7	保留	RO	-
6:0	I²C从地址 定义I ² C控制器使用的从地址。	R/W	注8-2

注8-2 该寄存器的默认值通过CFG_SEL1引脚定义。

8.5.11 PPC状态寄存器（PPC_BUSY）

地址： 001Bh 大小： 8位

BIT	说明	类型	默认值
7:1	保留	RO	-
0	PPC_BUSY 该位用于指示PPC何时处理寄存器写入请求。 0b: PPC未忙于处理寄存器写访问。可启动新的寄存器写操作。 1b: PPC正在处理寄存器写访问。直至该位清零，才能启动新的寄存器写操作。	RO	0b

8.5.12 PIO状态寄存器（PIO_STS）

地址： 0020h 大小： 16位

Bit	说明	类型	默认值
15	PIO15 0b: PIO为低电平 1b: PIO为高电平	RO	0b
14	PIO14 0b: PIO为低电平 1b: PIO为高电平	RO	0b
13	PIO13 0b: PIO为低电平 1b: PIO为高电平	RO	0b
12	PIO12 0b: PIO为低电平 1b: PIO为高电平	RO	0b
11	PIO11 0b: PIO为低电平 1b: PIO为高电平	RO	0b
10	PIO10 0b: PIO为低电平 1b: PIO为高电平	RO	0b
9	PIO9 0b: PIO为低电平 1b: PIO为高电平	RO	0b
8	PIO8 0b: PIO为低电平 1b: PIO为高电平	RO	0b
7	PIO7 0b: PIO为低电平 1b: PIO为高电平	RO	0b

UPD360

Bit	说明	类型	默认值
6	PIO6 0b: PIO为低电平 1b: PIO为高电平	RO	0b
5	PIO5 0b: PIO为低电平 1b: PIO为高电平	RO	0b
4	PIO4 0b: PIO为低电平 1b: PIO为高电平	RO	0b
3	PIO3 0b: PIO为低电平 1b: PIO为高电平	RO	0b
2	PIO2 0b: PIO为低电平 1b: PIO为高电平	RO	0b
1	PIO1 0b: PIO为低电平 1b: PIO为高电平	RO	0b
0	PIO0 0b: PIO为低电平 1b: PIO为高电平	RO	0b

PIO_STS[0] = 20h

PIO_STS[1] = 21h

注：通用信号GPO1和GPO10只能用作输出，且在将RESET_N置为无效时必须保持在低电平状态。

注：UPD360-C中未提供GPIO0。

注：在独立的DFP/UFP模式下（仅限UPD360-A/UPD360-B），GPIO 9-15具有备用的专用功能，如第11页的第3.1.1.1节“独立DFP/UFP模式下的UPD360-A/UPD360-B GPIO[9:15]功能”所定义。

8.5.13 PIO中断状态寄存器（PIO_INT_STS）

地址：0022h 大小：16位

Bit	说明	类型	默认值
15:0	PIO_INT[15:0] 置1时，指示PIO[15:0]上发生电平变化。 通过向相应位写入“1”将中断清除。	R/WC	0000h

注：向某位写“1”将清零相应位，并使能对下一个电平转换的检测。如果未通过PIO_EN寄存器中的相应位禁止，则该寄存器中任何位为“1”均会强制将PIO_INT中断置为有效。

8.5.14 PIO中断允许寄存器（PIO_INT_EN）

地址：0024h 大小：16位

Bit	说明	类型	默认值
15:0	PIO_EN[15:0] 置1时，相应的PIO线路状态发生变化时会产生中断。	R/W	0h

8.5.15 配置PIOX寄存器（CFG_PIOX）

地址：x=0: 0030h 大小：8位
x=1: 0031h
x=2: 0032h
x=3: 0033h
x=4: 0034h
x=5: 0035h
x=6: 0036h
x=7: 0037h
x=8: 0038h
x=9: 0039h
x=10: 003Ah
x=11: 003Bh
x=12: 003Ch
x=13: 003Dh
x=14: 003Eh
x=15: 003Fh

Bit	说明	类型	默认值
7	PullUpEnable 0b: 无上拉 1b: 使能上拉。	R/W	0b
6	PullDownEnable 0b: 无下拉 1b: 使能下拉。	R/W	0b
5	FallingAlert 0b: 下降沿无警报 1b: 在下降沿将PIO_INT置为有效	R/W	0b
4	RisingAlert 0b: 上升沿无警报 1b: 在上升沿将PIO_INT置为有效	R/W	0b
3	DataOutput 0b: 无效（推挽或漏极开路时为低电平） 1b: 有效（推挽、高阻态或漏极开路时为高电平） 注： 输入缓冲器状态可通过PIO状态寄存器（PIO_STS）中的相应位确定。 注： 配置PIO输出寄存器（CFG_PIO_OUT）中的相关PIO输出位将更新，以匹配向该位写入的值。	R/W	0b

UPD360

Bit	说明	类型	默认值
2	缓冲器类型 置1时，相应GPIO信号的输出缓冲器配置为推/挽驱动器。 清零时，相应GPIO信号的输出缓冲器配置为漏极开路驱动器。	R/W	0b
1	方向 0b: 输入 1b: 输出	R/W	0b
0	使能 0b: 禁止GPIO 1b: 使能GPIO 注： 当禁止GPIO时，默认功能（如果存在）将正常工作。所有GPIO均默认为使能状态并配置为输入。	R/W	1b

注： 通用信号GPO1和GPO10只能用作输出，且在将RESET_N置为无效时必须保持在低电平状态。

注： UPD360-C中未提供GPIO0。

注： 在独立的DFP/UFP模式下（仅限UPD360-A/UPD360-B），GPIO 9-15具有备用的专用功能，如第11页的[第3.1.1.1节“独立DFP/UFP模式下的UPD360-A/UPD360-B GPIO\[9:15\]功能”](#)所定义。

8.5.16 配置PIOX更新寄存器（CFG_PIOX_UPD）

地址： x=0: 0040h 大小： 8位
 x=1: 0041h
 x=2: 0042h
 x=3: 0043h
 x=4: 0044h
 x=5: 0045h
 x=6: 0046h
 x=7: 0047h
 x=8: 0048h
 x=9: 0049h
 x=10: 004Ah
 x=11: 004Bh
 x=12: 004Ch
 x=13: 004Dh
 x=14: 004Eh
 x=15: 004Fh

Bit	说明	类型	默认值
7	PullUpEnabled	R/W	0b
6	PullDownEnabled	R/W	0b
5	FallingAlert	R/W	0b
4	RisingAlert	R/W	0b
3	OutputValue	R/W	0b
2	BufferType	R/W	0b
1	方向	R/W	0b
0	使能	R/W	0b

注： 通用信号GPO1和GPO10只能用作输出，且在将RESET_N置为无效时必须保持在低电平状态。

注： UPD360-C中未提供GPIO0。

注： 在独立的DFP/UFP模式下（仅限UPD360-A/UPD360-B），GPIO 9-15具有备用的专用功能，如第11页的第3.1.1.1节“独立DFP/UFP模式下的UPD360-A/UPD360-B GPIO[9:15]功能”所定义。

8.5.17 PIO 改写 X源选择寄存器（PIO_OVRX_SRC_SEL）

地址：x=0: 0050h 大小：8位

x=1: 0051h

x=2: 0052h

x=3: 0053h

Bit	说明	类型	默认值
7:5	VBUS 阈值选择 当选择VBUS_DET 来实现改写功能时，该字段用于选择应与哪个VBUS 阈值进行匹配。 000b: VSAFE0V_THR_MATCH 001b: VBUS0_THR_MATCH 010b: VBUS1_THR_MATCH 011b: VBUS2_THR_MATCH 1xxb: VBUS3_THR_MATCH	R/W	0h
4:0	改写选择 该字段选择相应PIO 改写的来源。 0h: PIO0 1h: PIO1 2h: PIO2/OCS_COMP2 3h: PIO3 4h: PIO4 5h: PIO5 6h: PIO6 7h: PIO7 8h: PIO8 9h: PIO9 Ah: PIO10 Bh: PIO11 Ch: PIO12 Dh: PIO13 Eh: PIO14 Fh: PIO15 10h: OCS_COMP1 11h: VBUS_DET 12h-1Fh: 保留	R/W	0h

8.5.18 PIO 改写监视值寄存器 (PIO_OVR_MON_VAL)

地址: 0054h 大小: 8位

Bit	说明	类型	默认值
7:4	保留	RO	-
3	PIO_MON_VAL3 改写功能3在所选PIO上监视该值。	R/W	0b
2	PIO_MON_VAL2 改写功能2在所选PIO上监视该值。	R/W	0b
1	PIO_MON_VAL1 改写功能1在所选PIO上监视该值。	R/W	0b
0	PIO_MON_VAL0 改写功能0在所选PIO上监视该值。	R/W	0b

注: 当PIO映射到OCS比较器或VBUS比较器时, 受监视项将上升/下降至超过相关阈值。改写监视值置1表示监视高于阈值的源。如果该值清零, 则监视低于阈值的源。

8.5.19 PIO 改写使能寄存器 (PIO_OVR_EN)

地址: 0055h 大小: 8位

Bit	说明	类型	默认值
7:4	保留	RO	-
3	PIO_OVR_EN3 置1时, 使能PIO改写3。	R/W	0b
2	PIO_OVR_EN2 置1时, 使能PIO改写2。	R/W	0b
1	PIO_OVR_EN1 置1时, 使能PIO改写1。	R/W	0b
0	PIO_OVR_EN0 置1时, 使能PIO改写0。	R/W	0b

UPD360

8.5.20 PIO 改写状态寄存器 (PIO_OVR_STS)

地址: 0056h 大小: 8位

Bit	说明	类型	默认值
7:4	保留	RO	-
3	PIO_OVR_STS3 所选PIO的改写功能激活。 在与PIO_MON_VAL3相匹配时置1，在改写功能禁止（PIO_OVR_EN3 = 0）时清零。	RO	0b
2	PIO_OVR_STS2 所选PIO的改写功能激活。 在与PIO_MON_VAL2相匹配时置1，在改写功能禁止（PIO_OVR_EN2 = 0）时清零。	RO	0b
1	PIO_OVR_STS1 所选PIO的改写功能激活。 在与PIO_MON_VAL1相匹配时置1，在改写功能禁止（PIO_OVR_EN1 = 0）时清零。	RO	0b
0	PIO_OVR_STS0 所选PIO的改写功能激活。 在与PIO_MON_VAL0相匹配时置1，在改写功能禁止（PIO_OVR_EN0 = 0）时清零。	RO	0b

8.5.21 PIO 改写中断状态寄存器 (PIO_OVR_INT_STS)

地址: 0057h 大小: 8位

Bit	说明	类型	默认值
7:4	保留	RO	-
3	PIO_OVR_INTR3 当改写功能3在所选PIO上检测到编程的监控值时，该中断置1。 中断一直持续到该位清零。	R/WC	0b
2	PIO_OVR_INTR2 当改写功能2在所选PIO上检测到编程的监控值时，该中断置1。 中断一直持续到该位清零。	R/WC	0b
1	PIO_OVR_INTR1 当改写功能1在所选PIO上检测到编程的监控值时，该中断置1。 中断一直持续到该位清零。	R/WC	0b
0	PIO_OVR_INTR0 当改写功能0在所选PIO上检测到编程的监控值时，该中断置1。 中断一直持续到该位清零。	R/WC	0b

8.5.22 PIO 改写中断允许寄存器 (PIO_OVR_INT_EN)

地址: 0058h 大小: 8位

Bit	说明	类型	默认值
7:4	保留	RO	0h
3	PIO_OVR_INTR3 为“1”时，允许产生该中断。	R/W	0b
2	PIO_OVR_INTR2 为“1”时，允许产生该中断。	R/W	0b
1	PIO_OVR_INTR1 为“1”时，允许产生该中断。	R/W	0b
0	PIO_OVR_INTR0 为“1”时，允许产生该中断。	R/W	0b

8.5.23 配置PIO输出寄存器（CFG_PIO_OUT）

地址：0059h 大小：16位

该寄存器允许在一次CSR写操作中更新多个PIO输出值。

Bit	说明	类型	默认值
15:0	CFG_PIO_OUT 当写入该寄存器时，将更新各个PIO的输出值。具体来说，将更新相应CFG_PIOx寄存器中的DataOutput。 直到通过软件写入位于005Ah的字节后，才会更新该寄存器。这是为了防止更新时将外部驱动的组件置于无效状态。	R/W	0000h

8.5.24 配置PIO输出更新寄存器（CFG_PIO_OUT_UPD）

地址：005Bh 大小：16位

Bit	说明	类型	默认值
15:0	CFG_PIO_OUTx_UPD	R/W	0000h

8.5.25 PIO改写X输出使能寄存器（PIO_OVRX_OUT_EN）

地址：x=0: 0060h 大小：16位
x=1: 0062h
x=2: 0064h
x=3: 0066h

Bit	说明	类型	默认值
15:0	PIO_OVR_OUT_EN_X 每个改写功能都具有此CSR的一个实例，指示其控制哪些PIO输出。置为有效时，相关PIO输出的状态由相应改写功能控制。 注： 同一PIO输出可以通过多个改写来使能。	R/W	0h

8.5.26 PIO 改写输出寄存器 (PIO_OVR_OUT)

地址: 0068h 大小: 16 位

Bit	说明	类型	默认值
15:0	PIO 改写输出 该字段定义当相应PIO 针对改写功能使能并且相关改写触发时使用的PIO 输出值。	R/W	0h

8.5.27 PIO 改写方向寄存器 (PIO_OVR_DIR)

地址: 006Ah 大小: 16 位

Bit	说明	类型	默认值
15:0	PIO 改写方向 该字段定义当相应PIO 针对改写功能使能并且相关改写触发时使用的PIO 方向值。	R/W	0h

8.5.28 PIO 去抖计数十毫秒寄存器 (PIO_DEBOUNCE_10MS_COUNT)

地址: 006Ch 大小: 8 位

Bit	说明	类型	默认值
7:0	PIO_DEBOUNCE_10MS_COUNT 该寄存器保存 10 ms 去抖计数器。 去抖周期内的任何电平变化都将被抑制。每个计数对应10 ms，默认值为100 ms。 当去抖粒度为 10 ms 时，使用该寄存器。 注： 实际计数可以是计数或计数-1。	R/W	0Ah

8.5.29 PIO 去抖计数一毫秒寄存器（PIO_DEBOUNCE_1MS_COUNT）

地址： 006Dh 大小： 8位

Bit	说明	类型	默认值
7:0	PIO_DEBOUNCE_1MS_COUNT 该寄存器保存 1 ms 去抖计数器。 去抖周期内的任何电平变化都将被抑制。每个计数对应 1 ms，默认值为 10 ms。 当去抖粒度为 1 ms 时，使用该寄存器。 注： 实际计数可以是计数或计数-1。	R/W	0Ah

8.5.30 PIO 去抖计数一微秒寄存器（PIO_DEBOUNCE_1US_COUNT）

地址： 006Eh 大小： 8位

Bit	说明	类型	默认值
7:0	PIO_DEBOUNCE_1US_COUNT 该寄存器保存 1 μs 去抖计数器。 去抖周期内的任何电平变化都将被抑制。每个计数对应 1 μs，默认值为 10 μs。 当去抖粒度为 1 μs 时，使用该寄存器。 注： 实际计数可以是计数或计数-1。	R/W	0Ah

8.5.31 PIO 去抖状态寄存器 (PIO_DEBOUNCE_STS)

地址: 006Fh 大小: 8位

Bit	说明	类型	默认值
7:1	保留	RO	0h
0	PIO_DB_ACT PIO去抖激活 该位用于指示PIO去抖正在进行中。 1b: PIO去抖处于待处理状态 0b: PIO未处于待处理状态	RO	0b

8.5.32 PIO 去抖使能寄存器 (PIO_DEBOUNCE_EN)

地址: 0070h 大小: 32位

该寄存器按如下定义使能PIO的去抖操作并选择要使用的去抖动器:

- 00b: 禁止去抖
- 01b: 使能去抖 (PIO去抖计数一微秒寄存器 (PIO_DEBOUNCE_1US_COUNT))
- 10b: 使能去抖 (PIO去抖计数一毫秒寄存器 (PIO_DEBOUNCE_1MS_COUNT))
- 11b: 使能去抖 (PIO去抖计数十毫秒寄存器 (PIO_DEBOUNCE_10MS_COUNT))

Bit	说明	类型	默认值
31:30	PIO_DB_EN15	R/W	00b
29:28	PIO_DB_EN14	R/W	00b
27:26	PIO_DB_EN13	R/W	00b
25:24	PIO_DB_EN12	R/W	00b
23:22	PIO_DB_EN11	R/W	00b
21:20	PIO_DB_EN10	R/W	00b
19:18	PIO_DB_EN9	R/W	00b
17:16	PIO_DB_EN8	R/W	00b
15:14	PIO_DB_EN7	R/W	00b
13:12	PIO_DB_EN6	R/W	00b
11:10	PIO_DB_EN5	R/W	00b
9:8	PIO_DB_EN4	R/W	00b
7:6	PIO_DB_EN3	R/W	00b

UPD360

Bit	说明	类型	默认值
5:4	PIO_DB_EN2	R/W	00b
3:2	PIO_DB_EN1	R/W	00b
1:0	PIO_DB_EN0	R/W	00b

注： UPD360-C 中未提供 **GPIO0**。

8.5.33 OCS 比较器控制寄存器（OCS_CMP_CTL）

地址： 0080h 大小： 8 位

Bit	说明	类型	默认值
7	OCS_DB_ACTIVE 当该位读回0b时，将禁止去抖动器。当该位读回1b时，将使能OCS去抖动器。	RO	0b
6:2	保留	RO	0h
1:0	OCS 比较器控制 00b: OCS 比较器和 DAC 掉电 01b: OCS 比较器采样 OCS_COMP1 10b: OCS 比较器采样 OCS_COMP2 11b: OCS 比较器采样 OCS_COMP1 和 OCS_COMP2 当设置为01b或10b时，每100 μs采样一次。当设置为11b时，各 OCS_COMPx 引脚每200 μs有效采样一次。	R/W	00b

8.5.34 OCS 比较器匹配寄存器（OCS_CMP_MATCH）

地址： 0081h 大小： 8 位

当 **OCS 比较器控制** 未针对相应 **OCS_COMP** 引脚使能时，该位始终清零。

Bit	说明	类型	默认值
7:2	保留	RO	0h
1	OCS 比较器匹配2 该位置1时，表示已发生匹配，并在 OCS_COMP2 引脚上检测到OCS条件。	RO	0b
0	OCS 比较器匹配1 该位置1时，表示已发生匹配，并在 OCS_COMP1 引脚上检测到OCS条件。	RO	0b

8.5.35 OCS 比较器更改状态寄存器 (OCS_CMP_CHG_STS)

地址: 0082h 大小: 8位

Bit	说明	类型	默认值
7:6	OCSx匹配有效 (OCSx_MATCH_VLD) 在通过OCS比较器控制首次使能OCS去抖动器且OCS比较器匹配寄存器 (OCS_CMP_MATCH) 中首次匹配变为有效后置为有效。 Bit 7 = OCS2_MATCH_VLD Bit 6 = OCS1_MATCH_VLD	R/WC	0b
5:2	保留	RO	0h
1:0	OCSx更改状态 (OCSx_CHG_STS) 各个位置1时, 表示OCS比较器匹配寄存器 (OCS_CMP_MATCH) 中的相应位发生了更改。 Bit 1 = OCS2_CHG_STS Bit 0 = OCS1_CHG_STS 写入1b会将相应状态位清零。	R/WC	00b

8.5.36 OCS 比较器匹配使能寄存器 (OCS_CMP_MATCH_EN)

地址: 0083h 大小: 8位

Bit	说明	类型	默认值
7:6	OCSx匹配有效使能 (OCSx_MATCH_VLD_EN) 置1时, OCS比较器更改状态寄存器 (OCS_CMP_CHG_STS) 中的相应位可将OCS_CMP_INT中断置为有效。 Bit 7 = OCS2_MATCH_VLD_EN Bit 6 = OCS1_MATCH_VLD_EN	R/W	0b
5:2	保留	RO	0h
1:0	OCSx比较器匹配使能 (OCSx_CMP_MATCH_EN) 置1时, OCS比较器更改状态寄存器 (OCS_CMP_CHG_STS) 中的相应位可将OCS_CMP_INT中断置为有效。 Bit 1 = OCS2_CMP_MATCH_EN Bit 0 = OCS1_CMP_MATCH_EN	R/W	00b

UPD360

8.5.37 OCS 比较器X阈值寄存器（OCS_CMPX_THR）

地址： x=1: 0084h 大小： 16 位
 x=2: 0086h

Bit	说明	类型	默认值
15:10	保留	RO	0h
9:0	OCS 比较器阈值 阈值的低字节必须在高字节之前写入。当发生第二次写入时，将更新全部10位阈值。 单位为~2.5V/1024。 注： 使能OCS去抖器时，不得修改该寄存器。	R/W	0h

8.5.38 OCS 比较器去抖寄存器（OCS_CMP_DEBOUNCE）

地址： 0088h 大小： 8 位

Bit	说明	类型	默认值
7:0	OCS_CMP_DEBOUNCE 表示OCS_COMP1和OCS_COMP2输入去抖的持续时间。 该字段的单位为100 μs。 注： 使能OCS去抖器时，不得修改该寄存器。	R/W	0Fh

8.5.39 IRQ_N PU/PD控制寄存器（IRQ_PUPD）

地址： 009Bh 大小： 8 位

该寄存器控制IRQ_N引脚的PU和PD。

Bit	说明	类型	默认值
7	PullUpEnable 0: 无上拉 1: 使能上拉。	R/W	0b
6	PullDownEnable 0: 无下拉 1: 使能下拉。	R/W	0b
5:0	保留	RO	-

8.5.40 稳压器控制寄存器 (VREG_CTL)

地址: 009Ch 大小: 8位

Bit	说明	类型	默认值
7:1	保留	RO	-
1	VREG_LOW_OVR 置1时, 只要环形振荡器和48 MHz张弛振荡器被禁止, 1.8V稳压器就会进入低功耗模式。 注: 预计该位会在CPU伙伴模式下置1。 注: 在独立模式下, 应将相应OTP位置1以使能该功能并确保最大限度地降低功耗。	R/W	0b
0	VREG低功耗使能 (VREG_LOW_EN) 置1时, 1.8V稳压器处于低功耗模式。 注: 当 VREG_LOW_OVR 位置1时, 该位没有意义。	R/W	0b

8.5.41 I²C微调寄存器 (I²C_TRIM)

地址: 009Dh 大小: 8位

BIT	说明	类型	默认值
7:0	I2C_TRIM 该寄存器用于指定为满足tHD_DAT参数而由I ² C控制器添加的延时。 该寄存器的单位为20.8 ns。	R/W	9h

8.5.42 通用定时器装载寄存器 (GP_TIMER_LOAD)

地址: 00A0h 大小: 16位

BIT	说明	类型	默认值
15:0	GP定时器装载 该寄存器定义要装入通用定时器的值, 通用定时器将从该值开始递减计数。 该寄存器的有效范围为0001h-FFFFh。 该字段的单位为1 ms。 注: 在该寄存器中编程0h是无效的。 注: 软件不得在 RESET 位置1时更改此寄存器的内容, 而应等待至该位清零。	R/W	FFFFh

UPD360

8.5.43 通用定时器中断源寄存器（GP_TIMER_INT_SRC）

地址：00A2h 大小：8位

BIT	说明	类型	默认值
7:1	保留	RO	-
0	GP_UFLOW 通用定时器下溢时，该位将置1。中断状态寄存器（INT_STS）中的GP_TIMER_INT中断一直持续至该位清零。 当通用定时器控制寄存器（GP_TIMER_CTL）的DIS_ON_UFLOW位置1时，将该位置为有效的操作将延迟到定时器完全禁止后进行（100 μs）。	R/WC	0b

8.5.44 通用定时器控制寄存器（GP_TIMER_CTL）

地址：00A3h 大小：8位

BIT	说明	类型	默认值
7	STATUS 该位指示通用定时器的工作状态。 0: 定时器禁止，未在计数 1: 定时器使能，正在计数 该位清零时，定时器已暂停工作。	RO	0b
6:3	保留	RO	-
2	DIS_ON_UFLOW 当该位置1时，一旦检测到下溢条件，通用定时器便会自动禁止。此外，ENABLE位将在检测到下溢后清零，将GP_UFLOW置为有效的操作将延迟到通用定时器暂停工作后进行，并且STATUS位将清零。	R/W	0b
1	RESET 定时器复位。该位置1时，将停止定时器并将内部计数器复位为通用定时器装载寄存器（GP_TIMER_LOAD）中的值。 0: 定时器未处于复位状态 1: 定时器处于复位状态 注： 直到复位的传播完成，该位才清零。这可能需要100 μs。	R/SC	0b

BIT	说明	类型	默认值
0	ENABLE 定时器使能。该位用于启动和停止通用定时器装载寄存器（GP_TIMER_LOAD）。该位不会复位定时器计数。 如果DIS_ON_UFLOW位置1，则该位自动清零。 0：禁止定时器 1：使能定时器	R/W	0b

8.5.45 通用定时器计数寄存器（GP_TIMER_COUNT）

地址：00A4h 大小：16位

BIT	说明	类型	默认值
15:0	通用定时器计数器 该寄存器返回通用定时器的当前值。 读取该寄存器之前，必须通过将ENABLE位清零禁止通用定时器。	RO	0h

9.0 线缆插头方向和检测

本节详细介绍了关于控制和监视CC引脚、监视VBUS_DET引脚、控制VCONN FET以及采样CFG_SEL0和CFG_SEL1引脚的功能。

9.1 CC比较器

器件集成了一个比较器和DAC电路来实现Type-C连接和断开功能。它最多支持八个可编程的阈值，用于在UFP和DFP之间的连接检测。当作为UFP使用时，器件支持检测DFP的通告阈值变化，从而确定拉电流能力。CC比较器检测到的阈值的默认标称值为：

- 0.20V
- 0.40V
- 0.66V
- 0.80V
- 1.23V
- 1.60V
- 2.60V
- 3.0V 专有模式

表9-1： 线缆检测汇总

参数	阈值CSR	说明	最小值	典型值	最大值
DFP_ACT_DEF	CC_THR0	在配置为DFP且通告默认USB电流时检测有源线缆。		0.20V	
UFP_DFP_DEF	CC_THR0	在配置为UFP且DFP通告默认USB电流时检测DFP连接。		0.20V	
DFP_ACT_1A5	CC_THR1	在配置为DFP且通告1.5A电流时检测有源线缆。		0.40V	
UFP_DFP_1A5	CC_THR2	在配置为UFP且DFP通告1.5A电流时检测DFP连接。		0.66V	
DFP_ACT_3A0	CC_THR3	在配置为DFP且通告3.0A电流时检测有源线缆。		0.80V	
UFP_DFP_3A0	CC_THR4	在配置为UFP且DFP通告3.0A电流时检测DFP连接。		1.23V	
DFP_UFP_DEF	CC_THR5	在配置为通告默认USB电流的DFP时检测UFP连接。		1.60V	
DFP_UFP_1A5	CC_THR5	在配置为通告1.5A电流的DFP时检测UFP连接。		1.60V	
DFP_UFP_3A0	CC_THR6	在配置为通告3.0A电流的DFP时检测UFP连接。		2.60V	

以下表格总结了不同配置下**CCx匹配寄存器 (CCx_MATCH)**中要匹配的预期阈值。有关这些配置中使用的去抖器掩码的信息，请参见**CCx去抖清除使能寄存器 (CCx_DBCLR_EN)**。

表9-2: DFP CC匹配汇总

CC 状态	CC THR0	CC THR1	CC THR2	CC THR3	CC THR4	CC THR5	CC THR6	CC THR7
通告默认 USB 电流并连接到供电线缆	0	0	0	0	0	0	0	0
通告 1.5A 电流并连接到供电线缆	0	0	0	0	0	0	0	0
通告 3.0A 电流并连接到供电线缆	0	0	0	0	0	0	0	0
通告默认 USB 电流并连接到 UFP	1	0	0	0	0	0	0	0
通告 1.5A 电流并连接到 UFP	0	1	0	0	0	0	0	0
通告 3.0A 电流并连接到 UFP	0	0	0	1	0	0	0	0
通告默认 USB 电流，无连接 (vOpen)	1	0	0	0	0	1	0	0
通告 1.5A 电流，无连接 (vOpen)	0	1	0	0	0	1	0	0
通告 3.0A 电流，无连接 (vOpen)	0	0	0	1	0	0	1	0
专有模式，无连接 (vOpen)	0	0	0	0	0	0	0	1

表9-3: UFP CC匹配汇总

CC 状态	CC THR0	CC THR1	CC THR2	CC THR3	CC THR4	CC THR5	CC THR6	CC THR7
检测到供电线缆。	0	0	0	0	0	0	0	0
无连接 (SNK.Open)	0	0	0	0	0	0	0	0
连接到 DFP 并且通告默认 USB 电流	1	0	0	0	0	0	0	0
连接到 DFP 并且通告 1.5A 电流	1	0	1	0	0	0	0	0
连接到 DFP 并且通告 3.0A 电流	1	0	1	0	1	0	0	0
连接到 DFP 并且通告专有电流	1	0	1	0	1	0	0	1

以下示例说明了上述表格中预期匹配的基本原理。

DFP: 通告 1.5A 电流并连接到供电线缆，UFP 未连接

在这种情况下，DFP 通告 1.5A 电流并连接到供电线缆。UFP 未连接。根据规范，测得的 v_{Rd} 值将低于 0.35V。

在此配置中，使能以下阈值进行匹配：

- CC_THR1 (DFP_ACT_1A5)

UPD360

- CC_THR5 (DFP_UFP_1A5)

因此，编程到CCx去抖清除使能寄存器 (CCx_DBCLR_EN) 和CCx匹配使能寄存器 (CCx_MATCH_EN) 的值为0x22。

由于vRd小于0.35V，CCx匹配寄存器 (CCx_MATCH) 或CCx更改状态寄存器 (CCx_CHG_STS) 上不会检测到阈值匹配。

DFP：通告3.0A电流并连接到UFP

在这种情况下，DFP通告3.0A电流并连接到UFP。测得的vRd值将大于0.85V且小于2.45V。

在此配置中，使能以下阈值进行匹配：

- CC_THR3 (DFP_ACT_3A0)
- CC_THR6 (DFP_UFP_3A0)

因此，编程到CCx去抖清除使能寄存器 (CCx_DBCLR_EN) 和CCx匹配使能寄存器 (CCx_MATCH_EN) 中的值为0x44。

在这种情况下，只有CC_THR3会发生匹配。在去抖间隔后，CCx匹配寄存器 (CCx_MATCH) 或CCx更改状态寄存器 (CCx_CHG_STS) 中的bit 3将置1。

UFP：连接到DFP并且通告3.0A电流

在这种情况下，UFP连接到通告3.0A电流的DFP。测得的vRd值将大于1.3V且小于2.04V。

在此配置中，使能以下阈值进行匹配：

- UFP_DFP_DEF (CC_THR0)
- UFP_DFP_1A5 (CC_THR2)
- UFP_DFP_3A0 (CC_THR4)

因此，编程到CCx去抖清除使能寄存器 (CCx_DBCLR_EN) 和CCx匹配使能寄存器 (CCx_MATCH_EN) 的值为0x15。

在这种情况下，由于vRd的值必须超过1.23V的CC_THR4，因此所有三个阈值都将发生匹配。Bit 0、Bit 2和Bit 4将在CCx匹配寄存器 (CCx_MATCH) 或CCx更改状态寄存器 (CCx_CHG_STS) 中置1。

9.2 DFP操作

器件用作DFP时，将通过实现电流源来通告两个CC引脚的电流充电能力。

当建立UFP连接时，通过CC引脚驱动电流在UFP的Rd下拉电阻上产生一个电压，该电压可通过集成的CC比较器检测。表9-4总结了监视的电压。连接到有源线缆时，CC引脚上会出现另一个下拉 (Ra)。

DFP可以通过集成的端口电源控制器 (PPC) 提供通告的VBUS电源。

DFP还集成了两个5V FET来实现VCONN功能。第9.7节“VCONN操作”对此进行了进一步讨论。

表9-4： 电流源检测

CC1	CC2	连接状态	CC比较器状态	VBUS	VCONN
开路	开路	无连接	监视两个CC引脚以了解是否存在连接	关闭	关闭
Rd	开路	已连接UFP	监视CC1以了解是否断开	开启	关闭
开路	Rd	已连接UFP	监视CC2以了解是否断开	开启	关闭
Ra	开路	供电线缆，未连接UFP	监视CC2以了解UFP是否连接。 监视CC1以了解线缆是否断开。	关闭	关闭
开路	Ra	供电线缆，未连接UFP	监视CC1以了解UFP是否连接。 监视CC2以了解线缆是否断开。	关闭	关闭

表9-4: 电流源检测 (续)

CC1	CC2	连接状态	CC 比较器状态	VBUS	VCONN
Ra	Rd	供电线缆, 已连接UFP	监视CC2以了解UFP是否断开。 不通过监视CC1来了解是否断开。	开启	开启
Rd	Ra	供电线缆, 已连接UFP	监视CC1以了解UFP是否断开。 不通过监视CC2来了解是否断开。	开启	开启
Rd	Rd	已连接调试附件模式	监视两个CC引脚以了解是否断开	关闭	关闭
Ra	Ra	已连接音频附件模式	监视两个CC引脚以了解是否断开	关闭	关闭

9.2.1 RP 电流源

要通过集成端口电源控制器或外部电源电路来通告器件的电流充电能力, 需使用Rp电流源。电流源可以通过软件使用CC控制寄存器 (CC_CTL) 中的CC1 RP值和CC2 RP值来选择。表9-5总结了与编程值有关的电流源所支持的值。

表9-5: RP 电流源

DFP 通告	电流源 (1.7V至5.5V)	RPx 值
禁止		00b
默认 USB 电源	80 μ A +/- 20%	01b
1.5A @ 5V	180 μ A +/- 8%	10b
3.0A @ 5V	330 μ A +/- 8%	11b

与RP通告所使用的CC引脚耦合的电流源还用于采样CFG_SEL0和CFG_SEL1引脚。当对CFG_SEL0/CFG_SEL1引脚的任意一个进行采样时, 电流将偏离CC引脚并且不会通告RP值。更多详细信息, 请参见VBUS控制寄存器 (VBUS_CTL) 中的VBUS比较器控制字段。

9.2.2 电流源连接检测

以下部分介绍了在配置为电流源时确定是否发生连接的步骤：

1. 软件通过由 I2C/SPI 唤醒器件并使能 48 MHz 振荡器的方式来访问器件。
2. 软件将 **CC 比较器控制** 编程为 00b 并禁止 CC 去抖动器。
3. 软件轮询 **CC_DB_ACTIVE**，直至该位读回表示 CC 去抖动器无效的值。
4. 软件根据需要针对 tPDDebounce 周期编程 **匹配去抖寄存器 (MATCH_DEB)**。
5. 软件根据需要编程 **VBUS 去抖寄存器 (VBUS_DEB)**。
6. 软件根据需要编程 **CCx 去抖清除使能寄存器 (CCx_DBCLR_EN)** 和 **CCx 匹配使能寄存器 (CCx_MATCH_EN)** 以匹配所关注阈值。
7. 软件编程 **CCx 采样使能寄存器 (CCx_SAMP_EN)** 和 **VBUS 匹配使能寄存器 (VBUS_MATCH_EN)** 以检测 **VBUS VSafe0v 匹配 (VSAFE0V_THR_MATCH)**。
8. 软件通过 **CC 中断允许寄存器 (CC_INT_EN)** 允许 **CC_MATCH_VLD**、**CC1_MATCH_CHG** 和 **CC2_MATCH_CHG** 中断。
9. 软件通过 **电源中断允许寄存器 (PWR_INT_EN)** 允许 **VBUS_MATCH_VLD** 中断。

注： **CC_MATCH_VLD** 置为有效表示两个 CC 引脚的初始 CC 去抖操作已经完成，并且 **CCx 匹配寄存器 (CCx_MATCH)** 中存在有效值。**VBUS_MATCH_VLD** 表示 **VBUS 匹配寄存器 (VBUS_MATCH)** 中存在有效值。

10. 软件可通过允许 **中断允许寄存器 (INT_EN)** 中的 **CC_INT**、**VBUS_INT** 和 **PWR_INT** 将 **IRQ_N** 置为有效。
11. 软件通过 **CC 控制寄存器 (CC_CTL)** 中的 **CC1 RP 值** 和 **CC2 RP 值** 编程 RP 电流源。
12. 软件在需要时编程 **CC 阈值 x 寄存器 (CC_THRx)**。
13. 软件编程 **CC 比较器控制** 以采样两个 CC 引脚并使能 CC 去抖动器。**CC_DB_ACTIVE** 位稍后将置为有效。
14. 软件设置 **时钟控制寄存器 (CLK_CTL)** 以禁止所有时钟源接受保活时钟。
15. 当连接 UFP 时，有源线缆的 Rd 下拉电阻或 Ra 下拉电阻将连接到其中一个 CC 引脚。CC 比较器将对此进行检测。
16. CC 引脚的状态变化在编程的去抖周期后记录到 **CCx 匹配寄存器 (CCx_MATCH)** 和 **CCx 更改状态寄存器 (CCx_CHG_STS)** 中。
17. **CC_MATCH_VLD** 以及 **CC1_MATCH_CHG** 或 **CC2_MATCH_CHG** 中断置为有效并且 **CC_INT** 置为有效，从而将 **IRQ_N** 引脚置为有效。同样地，**VBUS_MATCH_VLD** 置为有效会使 **PWR_INT** 置为有效，从而也会将 **IRQ_N** 置为有效。
18. 软件在 tCCDebounce 内对 CC 匹配进一步去抖，以检测连接。
19. 软件必须在 **VBUS_MATCH_VLD** 置为有效后验证 **VBUS 匹配寄存器 (VBUS_MATCH)** 是否指示 VBUS 为 vSafe0v。
20. 在连接之后，软件将器件编程为通过集成的 PPC 为 VBUS 供电（需要时）。如果电源在器件外部，则软件将根据需要配置该电源。
21. 如果连接了有源线缆，则按照 **CCx 匹配寄存器 (CCx_MATCH)**，可以通过适当地将 **VBUS 控制寄存器 (VBUS_CTL)** 中的 **VCONN1 控制** 或 **VCONN2 控制** 置 1 为 VCONN 电源供电。
22. **CC 控制寄存器 (CC_CTL)** 中的 **CC 通信选择** 通过软件置 1，以正确地将基带接口连接到带 Rd 下拉电阻的 CC 引脚。
23. 如果需要，DFP 可尝试利用 PD MAC 与连接的器件进行通信。

9.2.3 电流源断开检测

配置为电流源时，伙伴UFP的断开通过监视相应CC引脚（带Rd下拉电阻）的电压是否超过DFP_UFP_DEF、DFP_UFP_1A5或DFP_UFP_3A0来确定，具体取决于器件通告的充电电流。

以下内容介绍了在伙伴模式下工作时确定是否发生断开的步骤：

应用注意事项： 即使Type-C规范允许，软件也不得在PD合约生效期间查看CC检测状态，以避免PD报文通信繁忙期间Attached.SNK状态呈现错误断开。

应用注意事项： 软件应在电流源配置下的PD合约生效期间设置一个3A通告，以避免PD报文通信繁忙期间Attached.SRC状态呈现错误断开；1.5A通告（尽管Type-C规范允许）不足以避免此问题，因为vOPEN阈值较低。此建议也存在例外情况，即实现“避免冲突”时。

以下步骤假设上一节中已使用CC去抖动器检测连接情况。

1. 软件通过由I2C/SPI唤醒器件并使能48 MHz振荡器的方式来访问器件。
2. 软件根据需要编程CCx去抖清除使能寄存器（CCx_DBCLR_EN）以使能每个阈值的去抖操作。通常这只涉及禁止不对应于vOPEN的阈值。
3. 软件通过CC中断允许寄存器（CC_INT_EN）允许CC1_MATCH_CHG或CC2_MATCH_CHG中断。
4. 软件通过将中断允许寄存器（INT_EN）中的相应CC_INT位置为有效来使能IRQ_N。
5. 软件编程CC比较器控制以采样适当的CC引脚。
6. 器件以20 kHz的工作时钟采样相应的CC引脚。
7. CC的状态变化反映在相应的CCx匹配寄存器（CCx_MATCH）和CCx更改状态寄存器（CCx_CHG_STS）中。
8. CC1_MATCH_CHG或CC2_MATCH_CHG中断置为有效，CC_INT置为有效，从而将IRQ_N引脚置为有效。
9. 软件在tCCDebounce内对CC匹配进一步去抖，以检测断开。
10. 如果正在为VCONN供电，则应通过软件适当地将VBUS控制寄存器（VBUS_CTL）中的VCONN1控制或VCONN2控制置1，从而禁止VCONN FET。在这种情况下，软件还必须按照第9.7.1节“VCONN放电编程模型”中所定义的方式使VCONN放电。
11. 软件禁止PPC，从而使5V内部电源开关断开（需要时）。也可能已由外部电源供电。
12. 内部100Ω VBUS放电开关闭合（需要时）。
13. 放电开关保持闭合，直到超过vSafe0V阈值。
14. 如果VBUS关闭寄存器（VBUS_OFF）中未超过vSafe0V阈值，则会将VBUS放电错误中断置为有效并将IRQ_N引脚置为有效。

注： 发生VBUS放电错误中断表示可能会出现灾难性系统电源问题。

15. 内部放电开关断开。
16. 器件由软件配置为检测UFP连接。禁止48 MHz振荡器。只有保活时钟保持使能状态。

9.3 UFP操作

作为UFP运行时，器件在两个CC线路上应用Rd下拉电阻，并从VBUS置为有效时起等待DFP连接。CC比较器用于确定DFP支持的通告电流充电器能力。

9.3.1 电流阱连接检测

以下步骤说明了如何编程器件来检测作为电流阱工作时的连接情况。下述讨论不包括电池电量耗尽的情况，这类情况将在第9.6节“电池电量耗尽”中说明。

1. 软件通过由I2C/SPI唤醒器件并使能48 MHz振荡器的方式来访问器件。
2. 软件将CC比较器控制编程为00b并禁止CC去抖动器。
3. 软件将VBUS比较器控制编程为00b并禁止VBUS去抖动器。
4. 软件轮询CC_DB_ACTIVE和VBUS_DB_ACTIVE，直至这两个位读回表示CC去抖动器和VBUS去抖动器无效的值0b。软件根据需要针对tPDDebounce间隔编程匹配去抖寄存器（MATCH_DEB）。
5. 软件根据需要编程CCx去抖清除使能寄存器（CCx_DBCLR_EN）以匹配所关注阈值。
6. 软件编程CC控制寄存器（CC_CTL）中的CC1下拉值和CC2下拉值以通告微调的Rd。
7. 软件通过CC中断允许寄存器（CC_INT_EN）允许CC_MATCH_VLD、CC1_MATCH_CHG或CC2_MATCH_CHG中断。
8. CC_MATCH_VLD置为有效表示两个CC引脚的初始CC去抖操作已经完成，并且CCx匹配寄存器（CCx_MATCH）中存在有效值。
9. 软件根据需要编程CCx去抖清除使能寄存器（CCx_DBCLR_EN）以使能相应阈值的去抖操作。
10. 软件通过配置VBUS匹配寄存器（VBUS_MATCH）来允许VBUS_INT，以便检测VBUS。
11. 软件通过配置电源中断允许寄存器（PWR_INT_EN）来使能VBUS_MATCH_VLD。
12. 软件通过为中断允许寄存器（INT_EN）中的CC_INT、PWR_INT和VBUS_INT配置相应位来使能IRQ_N。
13. 软件编程CC比较器控制以采样两个CC引脚。
14. 软件通过VBUS控制寄存器（VBUS_CTL）中的VBUS比较器控制并借助通过VBUS阈值x寄存器（VBUS_THRx）将VBUS_THR0和VBUS_THR1置1的方式编程VBUS比较器/DAC以检测vSafe5V。
15. 软件通过时钟控制寄存器（CLK_CTL）关闭除20 kHz保活时钟以外的所有时钟源以节省功耗。
16. 连接到伙伴DFP后，VBUS由5V电压供电。
17. 在编程的去抖间隔后，相应的CCx更改状态寄存器（CCx_CHG_STS）更新并且相应CC1_MATCH_CHG或CC2_MATCH_CHG中断置为有效，从而依次将CC_INT和IRQ_N置为有效。

注： CC_MATCH_VLD置为有效表示两个CC引脚的初始CC去抖操作已经完成，并且CCx匹配寄存器（CCx_MATCH）中存在有效值。

18. VBUS通过VBUS_DET比较器检测，并在VBUS去抖寄存器（VBUS_DEB）定义的周期内进行去抖。VBUS_INT中断置为有效，从而将IRQ_N置为有效。

注： VBUS_MATCH_VLD置为有效表示两个CC引脚的初始CC去抖操作已经完成，并且VBUS匹配寄存器（VBUS_MATCH）中存在有效值。

19. 软件必须在tCCDebounce内对CC匹配进行去抖并遵循PD去抖寄存器（PD_DEB）对VBUS进行去抖，以检测连接情况。
20. 连接后，软件将配置CC控制寄存器（CC_CTL）中的CC通信选择，以便在需要PD通信时，将基带接口与带Rp上拉电阻的CC正确连接。
21. DFP可与器件通信。

9.3.2 电流阱断开检测

DFP 断开状态通过是否移除VBUS来检测。必须始终为UFP使能VBUS比较器才能检测这种状态。

1. 软件将**VBUS比较器控制**编程为00b并禁止VBUS去抖动器。
2. 软件轮询**VBUS_DB_ACTIVE**，直至该位读回表示VBUS去抖动器无效的值0b。
3. 软件根据需要编程**VBUS去抖寄存器 (VBUS_DEB)**。
4. 软件根据需要编程**CCx采样使能寄存器 (CCx_SAMP_EN)**以使能相应阈值的去抖操作。
5. 软件使能**VBUS_INT**，该中断随后用于通过**VBUS匹配使能寄存器 (VBUS_MATCH_EN)**检测VBUS。
6. 软件通过为**中断允许寄存器 (INT_EN)**中的**VBUS_INT**配置相应位来使能**IRQ_N**。

注： 如有需要，VBUS_THR0和VBUS_THR1可通过**VBUS阈值x寄存器 (VBUS_THRx)**调整为vSafe5V。

7. 软件通过**VBUS控制寄存器 (VBUS_CTL)**中的**VBUS比较器控制**位来使能VBUS比较器/DAC。
8. 软件通过**时钟控制寄存器 (CLK_CTL)**关闭除20 kHz保活时钟以外的所有时钟源以节省功耗。
9. DFP移除VBUS。
10. 在VBUS去抖周期之后，**VBUS匹配寄存器 (VBUS_MATCH)**和**VBUS更改状态寄存器 (VBUS_CHG_STS)**将进行更新。**VBUS_INT**中断置为有效，从而将**IRQ_N**置为有效。
11. 软件继续监视VBUS，以确定其是否在tPdDebounce持续时间内保持在vSafe5V以下。

9.4 DRP 操作

本节介绍了器件实现DRP连接的一种用法。在这种配置中，软件利用器件根据USB Type-C规范以tDRP的间隔在电流源和电流阱通告间交替变化。

最初配置器件以通告电流源能力所采取的步骤遵循第9.2.2节“**电流源连接检测**”中定义的步骤。软件还必须实现tDRP定时器。如果未在这段时间内检测到电流阱，则软件应将器件的角色更改为电流阱，并再次尝试进行连接检测。

1. 软件通过由I2C/SPI唤醒器件并使能48 MHz振荡器的方式来访问器件。
2. 软件将**CC比较器控制**编程为00b并禁止CC去抖动器。
3. 软件轮询**CC_DB_ACTIVE**，直至该位读回表示CC去抖动器无效的值。
4. 软件根据需要针对tPDDebounce周期编程**匹配去抖寄存器 (MATCH_DEB)**。
5. 软件在需要时编程**CC阈值x寄存器 (CC_THRx)**。
6. 软件根据需要编程**CCx去抖清除使能寄存器 (CCx_DBCLR_EN)**和**CCx匹配使能寄存器 (CCx_MATCH_EN)**以匹配所关注阈值。
7. 软件通过**CC中断允许寄存器 (CC_INT_EN)**允许**CC_MATCH_VLD**、**CC1_MATCH_CHG**和**CC2_MATCH_CHG**中断。
8. 软件通过允许**中断允许寄存器 (INT_EN)**中的**CC_INT**将**IRQ_N**置为有效。
9. 软件通过**CC控制寄存器 (CC_CTL)**中的**CC1 RP值**和**CC2 RP值**编程RP电流源。
10. 软件编程**CC比较器控制**以采样两个CC引脚并使能CC去抖动器。**CC_DB_ACTIVE**位稍后将置为有效。
11. 软件设置**时钟控制寄存器 (CLK_CTL)**以禁止所有时钟源接受保活时钟。

注： **CC_MATCH_VLD**置为有效表示两个CC引脚的初始CC去抖操作已经完成，并且**CCx匹配寄存器 (CCx_MATCH)**中存在有效值。

如果在tDRP时间之后没有检测到连接，则软件应将器件的角色配置为电流阱并尝试检测是否存在电流源。这与第9.3.1节“**电流阱连接检测**”中定义的步骤类似。

1. 软件通过由I2C/SPI唤醒器件并使能48 MHz振荡器的方式来访问器件。
2. 软件将**CC比较器控制**编程为00b并禁止CC去抖动器。

3. 软件轮询 **CC_DB_ACTIVE**，直至该位读回表示 **CC** 去抖器无效的值 **0b**。
4. 软件根据需要编程 **CCx** 去抖清除使能寄存器 (**CCx_DBCLR_EN**) 以匹配所关注阈值。
5. 软件通过将 **VBUS** 阈值 **x** 寄存器 (**VBUS_THRx**) 中的 **VBUS_THR0** 和 **VBUS_THR1** 置 **1** 来编程对 **vSafe5V** 的检测。**CCx** 采样使能寄存器 (**CCx_SAMP_EN**) 编程为对这些阈值进行去抖操作。
6. 软件编程 **CC** 控制寄存器 (**CC_CTL**) 中的 **CC1** 下拉值和 **CC2** 下拉值以通告微调的 **Rd**。
7. 软件通过 **CC** 中断允许寄存器 (**CC_INT_EN**) 允许 **CC_MATCH_VLD**、**CC1_MATCH_CHG** 或 **CC2_MATCH_CHG** 中断。
8. 软件通过配置电源中断允许寄存器 (**PWR_INT_EN**) 来使能 **VBUS_MATCH_VLD**。
9. 软件通过为中断允许寄存器 (**INT_EN**) 中的 **CC_INT**、**PWR_INT** 和 **VBUS_INT** 配置相应位来使能 **IRQ_N**。
10. 软件编程 **CC** 比较器控制以采样两个 **CC** 引脚。
11. 软件通过时钟控制寄存器 (**CLK_CTL**) 关闭除 **20 kHz** 保活时钟以外的所有时钟源以节省功耗。

如果在 **tDRP** 时间之后没有检测到连接，则软件应将器件的角色配置为电流源并尝试检测是否存在电流阱。

注： **CC_MATCH_VLD** 置为有效表示两个 **CC** 引脚的初始 **CC** 去抖操作已经完成，并且 **CCx** 匹配寄存器 (**CCx_MATCH**) 中存在有效值。

注： **VBUS_MATCH_VLD** 置为有效表示两个 **CC** 引脚的初始 **CC** 去抖操作已经完成，并且 **VBUS** 匹配寄存器 (**VBUS_MATCH**) 中存在有效值。

9.5 避免冲突

需要一种备用工作模式来支持 **CC** 检测电路以软件形式实现冲突检测，冲突检测已纳入 **USB PD** 规范的版本 **3.0** 中。

为避免因电流阱发送的异步报文 (**AMS**) 引起报文冲突，电流源将 **Rp** 设置为 **SinkTxOk** (**3A@5V**) 以向电流阱指示可以发起 **AMS**。当电流源希望发起 **AMS** 时，它会将 **Rp** 设置为 **SinkTxNG** (**1.5A@5V**)。当电流阱检测到 **Rp** 设置为 **SinkTxOk** 时，可以发起 **AMS**。当电流阱检测到 **Rp** 设置为 **SinkTxNG** 时，不应发起 **AMS**，而只应发送电流源已发起的 **AMS** 的部分报文。

当作为电流阱工作时，需要一种机制来快速确定电流源在 **Rp** 上通告 **SinkTxNG** 还是 **SinkTxOK**。

软件可凭借避免冲突机制指示器件在单个 **CC** 引脚上仅采样单个阈值。这样一来，循环使用两个阈值只需要 **100 μs**，使得软件更容易满足规范中 **SinkTxOk** 规定的时序约束。软件可使用两个 **CCx** 采样使能寄存器 (**CCx_SAMP_EN**) 通过将相应位置 **1** 来指定应采样 **CC** 阈值的哪些子集。为避免冲突，只应在连接的 **CC** 引脚上设置与 **SinkTxOK** 对应的阈值。这对应于 **UFP_DFP_3A0** (**CC_THR4**)。**CC** 控制寄存器 (**CC_CTL**) 中的 **CC** 比较器控制字段应设置为用于 **PD** 通信的 **CC** 引脚。

为了防止在发送或接收 **PD** 数据包时出现 **SinkTXOK** 的误报检测，需要在匹配去抖寄存器 (**MATCH_DEB**) 中编程去抖值 (约为 **7 ms**)。当作为电流阱工作时，如此大的值会影响软件在 **PD** 规范中 **SinkTxTime** 定义的时间内检测 **SinkTXOK** 和发送 **PD** 报文的能力。为处理上述情况以及支持软件编程一个亚毫秒去抖值，引入了 **CC** 硬件控制寄存器 (**CC_HW_CTL**) 中的 **BLK_PD_MSG** 位。使用这种工作模式时，应至少为匹配去抖寄存器 (**MATCH_DEB**) 使用 **100 μs**。

以下序列说明了为电流阱使能“避免冲突”的步骤。该序列在连接并建立 **PD** 合约后使用。

注： 匹配去抖寄存器（MATCH_DEB）的单位由CC硬件控制寄存器（CC_HW_CTL）中的MATCH_DB_UNITS位确定。该位清零时，单位为1.6 ms。该位置1时，单位为100 μ s。

1. 电流阱与电流源建立连接，并协商PD合约。电流阱要向电流源发送AMS。
2. 通过CC硬件控制寄存器（CC_HW_CTL）将MATCH_DB_UNITS设置为1b，以便为CC匹配去抖器使用100 μ s单位。将BLK_PD_MSG位置1以通过去抖器过滤PD通信。
3. 软件通过向匹配去抖寄存器（MATCH_DEB）写入4将该寄存器设置为400 μ s。
4. 软件配置CC_THR4以通过CCx匹配使能寄存器（CCx_MATCH_EN）将中断置为有效。
5. 软件配置CC_THR4以通过CCx采样使能寄存器（CCx_SAMP_EN）进行采样。
6. 通过CC控制寄存器（CC_CTL）的CC比较器控制选择连接到电流源Rp的CC引脚。
7. 软件将等待通过CC_THR4上检测到的匹配来指示电流源Rp为SinkTxOK。
8. 软件启动到电流源的PD报文传送。

应用注意事项： 软件必须保证在连续传送的PD报文之间存在足够的空隙，以使电流源RP值能够在CC线路上趋于稳定。USB-IF成员的研究表明，稳定时间可能会超过40 μ s。

9.6 电池电量耗尽

实现了两种Rd电阻：Rd（电池电量耗尽）和Rd（微调）。CC1_DB_EN和CC2_DB_EN引脚用于确定电池电量耗尽情况下CC引脚的操作，通过PCB从外部连接至相应的CC引脚。当连接到通告上拉电阻的DFP时，CC引脚配置为提供高阻态或未经微调的Rd下拉电阻。

图9-1所示为通过将CCx_DB_EN和CCx一起固定到PCB上来支持电池电量耗尽情况的配置。UFP上拉激活与RD_DB串联的FET，并使能未经微调的电池电量耗尽下拉。

图9-1： CC RD（电池电量耗尽）

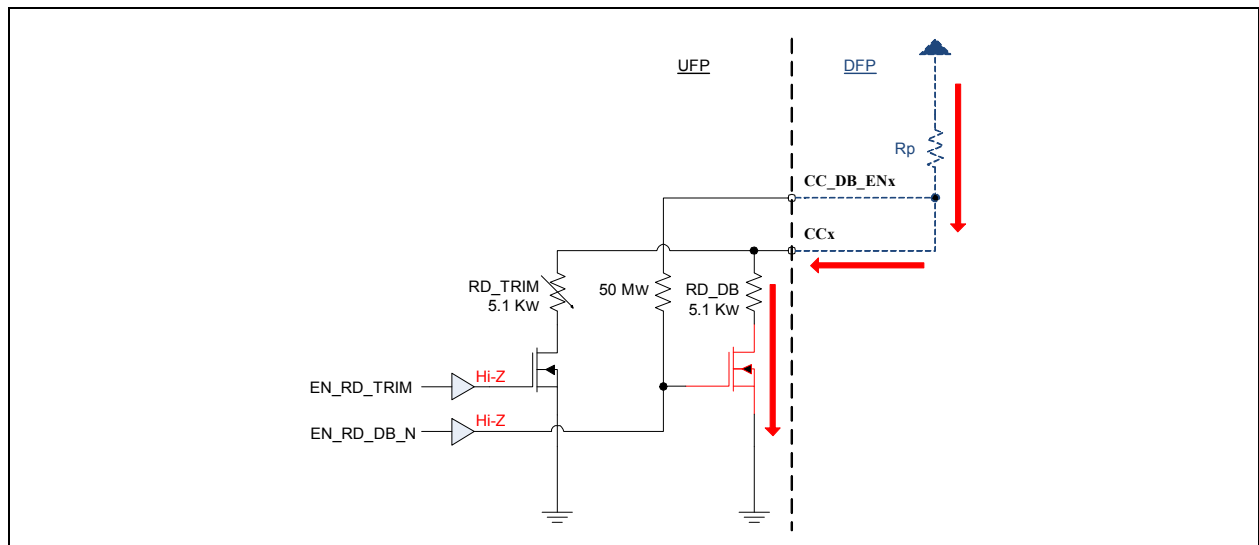
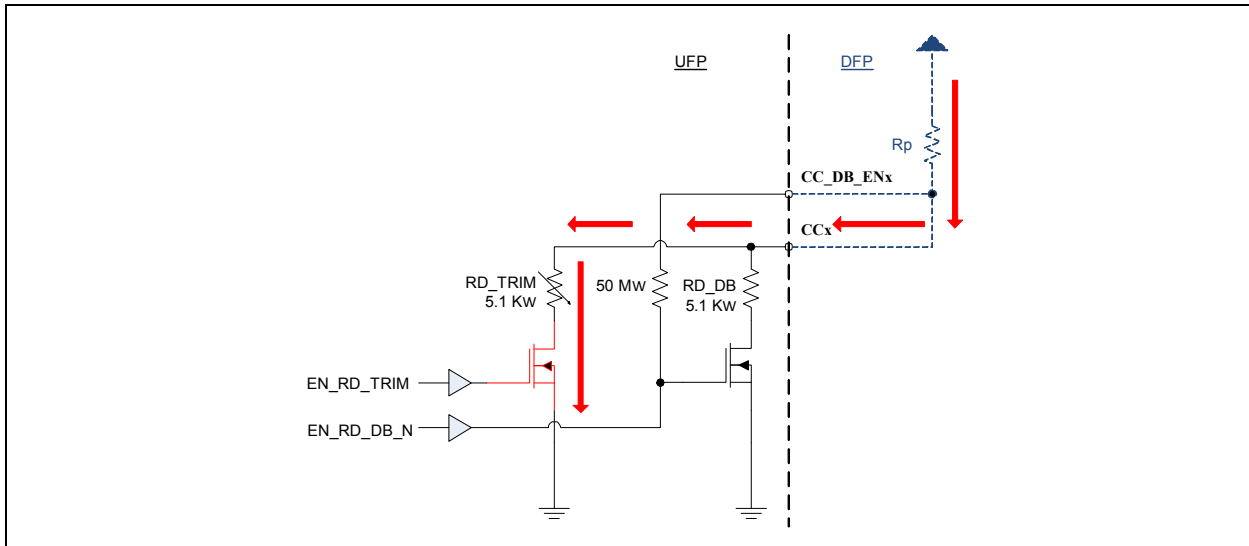


图9-2所示为UFP由DFP通过VBUS供电后的操作。器件上电后，EN_RD_DB默认情况下置为有效，从而使RD_DB下拉保持激活状态。

UPD360

在为主机CPU上电后，软件会将EN_RD_DB置为无效，同时将EN_RD_TRIM置为有效。之后，器件会提供RD_TRIM。

图9-2: CC RD（微调）



所提供的Rd电阻（微调或未经微调）由CC控制寄存器（CC_CTL）中的CC1下拉值和CC2下拉值控制。这些寄存器字段是所描述的EN_RD_TRIM和EN_RD_DB_N控制信号的基础。

9.7 VCONN操作

VCONN是一个5V电源，用于为USB Type-C插头中的电路供电，USB Type-C插头是实现电子标记线缆的必要因素。默认情况下，当连接有源线缆时，DFP始终为VCONN供电。不过，这可以通过软件使用PD VCONN_SWAP来更改。

VCONN FET由软件通过CC控制寄存器（CC_CTL）中的VCONN1控制和VCONN2控制控制位使能/禁止。

在独立DFP模式下，器件单独使能/禁止VCONN FET。该模式适用于没有主机CPU可用或CPU无法管理VCONN的配置。CC硬件控制寄存器（CC_HW_CTL）中的独立操作位用于使能这种工作模式。

应用注意事项： 无法同时使能两个FET。

通过内部监视电路来监视VCONN以了解过流情况。当VCONN过流事件持续的时间超过VCONN OCS和反向驱动去抖寄存器（VCONN_DEB）中规定的时间时，即认为出现过流情况。VCONN OCS监视通过VBUS控制寄存器（VBUS_CTL）中的VCONN OCS使能位使能。

当检测到过流VCONN事件时，电源中断状态寄存器（PWR_INT_STS）中的VCONN放电错误（VCONN_DISCH_ERR）中断置为有效。器件可配置为在检测到CC1反向驱动错误/CC2反向驱动错误或VCONN放电错误（VCONN_DISCH_ERR）时自动禁止VCONN FET。如果检测到去抖动的过流VCONN事件，则将禁止使能的VCONN FET。OCS事件还会导致自动禁止CC控制寄存器（CC_CTL）中相应的VCONN1控制和VCONN2控制控制位。

9.7.1 VCONN放电编程模型

当器件作为DFP工作并且检测到UFP断开时，软件可以使用以下编程模型来实现VCONN放电。

本部分不适用于独立DFP操作。

1. 软件确定连接的UFP已断开并断开VCONN FET。
2. 软件禁止监视与所连接的CC引脚对应的CC阈值。

- 通过CC1 RP值或CC2 RP值禁止两个CC引脚上的Rp电流源。
- 通过CC1下拉值或CC2下拉值选择先前为VCONN供电的CC引脚上的Ra下拉电阻。这将启动VCONN放电。
- 软件通过CC阈值x寄存器（CC_THRx）将CC阈值0设置为150 mV（41d），这是VCONN放电阈值。
- 软件将01h编程到CCx去抖清除使能寄存器（CCx_DBCLR_EN）中。
- 软件将01h编程到CCx匹配使能寄存器（CCx_MATCH_EN）中。
- 软件通过CC中断允许寄存器（CC_INT_EN）允许CC1_MATCH_CHG或CC2_MATCH_CHG中断。
- 通过CC控制寄存器（CC_CTL）的CC比较器控制选择先前为VCONN供电的CC引脚。
- CC引脚的状态变化在编程的去抖周期后记录到CCx匹配寄存器（CCx_MATCH）和CCx更改状态寄存器（CCx_CHG_STS）中。
- 在VCONN放电至150 mV以下后，IRQ_N将置为有效，表示VCONN已经放电。因CC_MATCH_VLD导致IRQ_N置为有效且无阈值匹配也可指示完全放电。
- 放电完成后，器件处于Unattached.SRC状态。

注：软件应通过实现一个定时器来指示 VCONN 放电错误。这可以通过通用定时器来实现。对于最大放电时间，Type-C 规范中没有具体要求。

注：在检测到UFP断开后，固件必须在Type-C规范版本1.1中规定的tVconnOff（35 ms）内断开VCONN电源。

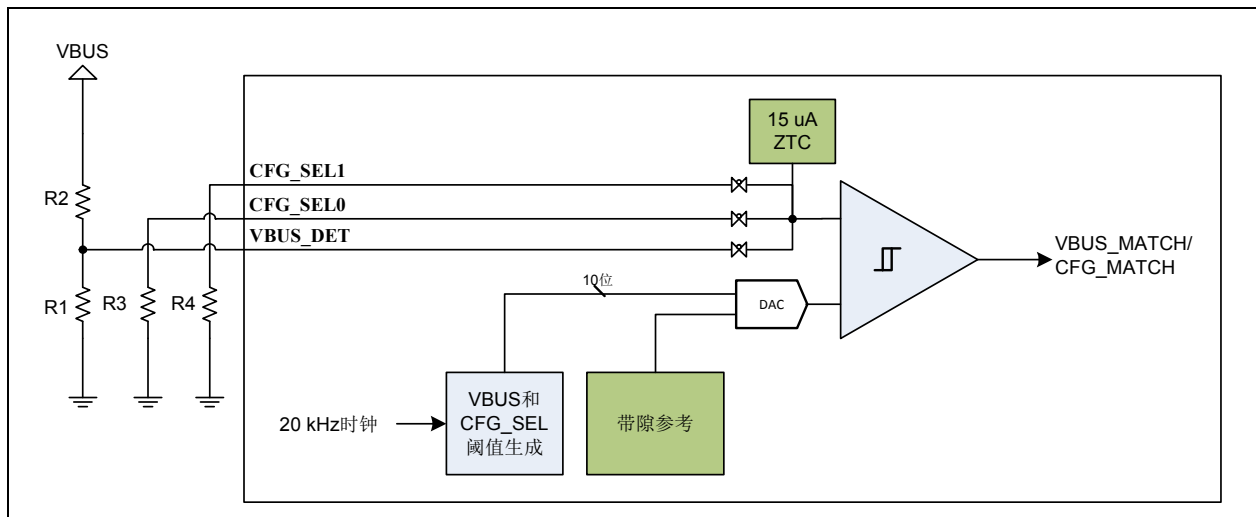
9.8 VBUS检测

器件通过实现一个比较器来确定VBUS何时处于编程范围vSafe5V或vSafe0v内。VBUS通过1:9电阻分压器外部分压，生成VBUS_DET。VBUS_DET与集成DAC生成的8位阈值进行比较。该比较器还由CFG_SEL0和CFG_SEL1引脚共用，这两个引脚在系统复位后自动进行采样。

图9-3所示为VBUS_DET电路。在典型用例中，VBUS_DET阈值编程为跟踪表9-6中定义的以下电压范围。

注：表9-6所示为使用精度为+/-1%的电阻的VBUS_DET的值，其中R1为10 kΩ，R2为90 kΩ。

图9-3: VBUS_DET比较器



对于DFP，VBUS比较器可用于检测VBUS何时处于PD协商规定的理想范围内。当VBUS由器件外部的电源生成时就是这种情况。

UPD360

对于UFP，需要使用VBUS比较器来确定DFP何时连接或断开。还可使用该比较器来确定VBUS何时处于通过PD协商的新电压范围内。

表9-6: VBUS检测阈值

VBUS	范围	VBUS_DET	备注
20	21.5	2.11	
	18.5	1.82	
12	13.1	1.29	
	10.9	1.07	
8	8.9	0.88	
	7.1	0.69	
5	5.5	0.51	vSafe5V
	3.67	0.33	
0.8	0.8	0.08	vSafe0V

如果支持，可以将8V、12V和20V范围编程到VBUS阈值2和VBUS阈值3寄存器中（见VBUS阈值x寄存器（VBUS_THRx））。同样地，可以将5V范围vSafe5v编程到VBUS阈值0和VBUS阈值1寄存器中。

vSafe0V的阈值可以通过VBUS VSafe0V阈值寄存器（VSAFE0V_THR）编程。

VBUS_DET监视逻辑在20 kHz振荡器下以循环使用各个阈值的方式工作。包括vSafe0v在内，一共要比较五个值。

比较的结果在VBUS去抖寄存器（VBUS_DEB）中定义的去抖周期后调整VBUS匹配寄存器（VBUS_MATCH）和VBUS更改状态寄存器（VBUS_CHG_STS）的各个位。

VBUS匹配寄存器（VBUS_MATCH）指示VBUS_DET的值何时高于编程的相应阈值，因此可用于确定VBUS何时处于理想范围内。

如果在电源中断状态寄存器（PWR_INT_STS）中正确配置，VBUS匹配寄存器（VBUS_MATCH）的状态变化可触发IRQ_N引脚置为有效。

VBUS去抖清除使能寄存器（VBUS_DBCLR_EN）的功能相当于CCx去抖清除使能寄存器（CCx_DBCLR_EN），但适用于VBUS去抖器。软件根据需要编程VBUS去抖清除使能寄存器（VBUS_DBCLR_EN）以使能所关注阈值的去抖操作。

以下示例说明了在PD合约协商后需要将VBUS从5V变为20V时使用的编程模型。最初时，使能VBUS去抖器并采样VSafe5v、VBUS_THR0和VBUS_THR1。VBUS_THR2和VBUS_THR3的初始值超过5V，例如25V。有关VBUS_THRx寄存器的详细信息，请参见第9.12.24节。

1. 软件编程VBUS匹配使能寄存器（VBUS_MATCH_EN）以包含新的阈值（VBUS_THR2和VBUS_THR3）。
2. 软件根据需要编程VBUS去抖清除使能寄存器（VBUS_DBCLR_EN）以使能每个阈值的去抖操作。
3. 软件在VBUS_THR2和VBUS_THR3中针对20V编程新的阈值（例如18.5V和21.5V）。该操作会重启VBUS去抖器。
4. 软件等待IRQ_N置为有效。
5. 软件确认VBUS_MATCH_VLD位置1。
6. 软件读取VBUS匹配寄存器（VBUS_MATCH）和VBUS更改状态寄存器（VBUS_CHG_STS）。
7. 如果VBUS更改状态寄存器（VBUS_CHG_STS）是非零的，软件会将状态位清零。

注： 如果VBUS匹配寄存器（VBUS_MATCH）中的位均未置1，则在VBUS_MATCH_VLD置为有效后，VBUS上检测到的电压将小于VSafe0V。

8. VBUS之后的变化将导致IRQ_N置为有效，同时VBUS匹配寄存器（VBUS_MATCH）和VBUS更改状态寄存器（VBUS_CHG_STS）将适当更新。

9.8.1 配置选择

CFG_SEL0和**CFG_SEL1**（仅限UPD360-A/UPD360-B）引脚与**VBUS**共用比较器，如图9-3所示。各个**CFG_SELx**引脚均连接到电阻分压器（通常上拉至VDDIO）。在系统级复位（POR、RESET_N和软件复位）后，将自动对两个**CFG_SELx**引脚进行采样以配置器件。如果检测到独立模式，**CFG_SELx**匹配寄存器（**CFG_SELx_MATCH**）会自动更新，器件将相应地自行配置。

表9-7和表9-8分别详细列出了**CFG_SEL0**和**CFG_SEL1**引脚的各种电阻设置。

注： 有关器件复位的更多信息，请参见第34页的第7.7节“复位操作”。

表9-7： 配置选择0（CFG_SEL0）设置

电阻 (+/-1%)	说明	CFG_SEL0_MATCH 寄存器
GND	独立DFP/USB58xx/USB59xx伙伴DFP 对于和主机不存在I ² C连接的应用，器件将以独立DFP模式工作（仅限UPD360-A/UPD360-B）。 PWR_CAPx 引脚用于选择提供的USB Type-C™ 电流。	0000h
0.475K	独立UFP/USB58xx/USB59xx伙伴UFP 对于和主机不存在I ² C连接的应用，器件将以独立UFP模式工作（仅限UPD360-A/UPD360-B）。	0001h
0.953K	I²C/SPI伙伴模式A（待定）	0003h
1.43K	I²C/SPI伙伴模式B（待定）	0007h
1.87K	I²C/SPI伙伴模式C（待定）	000Fh
2.37K	I²C/SPI伙伴模式D（待定）	001Fh
2.87K	I²C/SPI伙伴模式E（待定）	003Fh
3.32K	I²C/SPI伙伴模式F（待定）	007Fh
3.83K	I²C/SPI伙伴模式G（待定）	00FFh
4.22K	I²C/SPI伙伴模式H（待定）	01FFh
4.75K	I²C/SPI伙伴模式I（待定）	03FFh
5.23K	I²C/SPI伙伴模式J（待定）	07FFh
5.62K	I²C/SPI伙伴模式K（待定）	0FFFh
6.19K	I²C/SPI伙伴模式L（待定）	1FFFh
6.65K	I²C/SPI伙伴模式M（待定）	3FFFh
7.15K	I²C/SPI伙伴模式N（待定）	7FFFh
>7.15K	I²C/SPI伙伴模式O（待定）	FFFFh

注： 除GND（即10.5K）以外的任何**CFG_SEL0**值均将选择I²C/SPI伙伴模式。

表9-8: 配置选择1 (CFG_SEL1) I²C地址设置 (仅限UPD360-A/UPD360-B)

电阻 (+/-1%)	说明	CFG_SEL1_MATCH 寄存器
GND	I ² C从地址 = 1011_111	0000h
0.475K	I ² C从地址 = 1011_110	0001h
0.953K	I ² C从地址 = 1011_101	0003h
1.43K	I ² C从地址 = 1011_100	0007h
1.87K	I ² C从地址 = 1101_011	000Fh
2.37K	I ² C从地址 = 1101_010	001Fh
2.87K	I ² C从地址 = 1101_001	003Fh
3.32K	I ² C从地址 = 1101_000	007Fh
3.83K	I ² C从地址 = 1110_111	00FFh
4.22K	I ² C从地址 = 1110_110	01FFh
4.75K	I ² C从地址 = 1110_101	03FFh
5.23K	I ² C从地址 = 1110_100	07FFh
5.62K	I ² C从地址 = 1110_001	0FFFh
6.19K	I ² C从地址 = 1110_011	1FFFh
6.65K	I ² C从地址 = 1110_000	3FFFh
7.15K	I ² C从地址 = 1110_010	7FFFh
>7.15K	I ² C从地址 = 1001_000	FFFFh

9.9 反向驱动检测

两个CC引脚上均实现了反向驱动检测，用以防止电流反向流动。反向驱动保护电路始终处于工作状态，并在VCCx > VS时触发。

检测到反向驱动条件将导致电源中断状态寄存器 (PWR_INT_STS) 中的CC1反向驱动错误或CC2反向驱动错误位置为有效。

硬件支持在VCONN OCS和反向驱动去抖寄存器 (VCONN_DEB) 中指定的去抖周期之后自动禁止检测到反向驱动的CC引脚上的VCONN FET。该功能通过将VBUS控制寄存器 (VBUS_CTL) 中的CC反向驱动使能置1来使能。

9.10 独立DFP (仅限UPD360-A/UPD360-B)

9.10.1 概述

器件支持独立DFP操作，该模式下没有CPU可用于配置器件。该模式的关键应用是作为Microchip USB58xx/USB59xx系列USB集线器的USB Type-C DFP伙伴。

9.10.2 配置

适当设置CFG_SEL0和CFG_SEL1引脚即可进入该模式。通告和提供的电流充电通过PWR_CAP0和PWR_CAP1引脚定义。器件在系统级复位事件后自行配置。

应用注意事项: 可使用CC硬件控制寄存器 (CC_HW_CTL) 中的独立操作字段禁止独立操作。

9.11 独立UFP（仅限UPD360-A/UPD360-B）

9.11.1 概述

器件支持独立UFP操作，该模式下没有CPU可用于配置器件。该模式的关键应用是作为Microchip USB58xx/USB59xx系列USB集线器的USB Type-C UFP伙伴。

9.11.2 配置

适当设置CFG_SEL0和CFG_SEL1引脚即可进入该模式。器件在系统级复位事件后自行配置。

应用注意事项： 可使用CC硬件控制寄存器（CC_HW_CTL）中的独立操作字段禁止独立操作。

9.12 线缆方向和检测寄存器

本节详细介绍了线缆插头方向和检测寄存器。有关器件的完整寄存器映射的概述，请参见第19页的第4.0节“寄存器映射”。

表9-9: 系统控制和状态寄存器映射

地址	寄存器名称 (符号)
0800h	CC硬件控制寄存器 (CC_HW_CTL)
0803h	CC中断状态寄存器 (CC_INT_STS)
0804h	CCx更改状态寄存器 (CCx_CHG_STS) x=1
0805h	CCx更改状态寄存器 (CCx_CHG_STS) x=2
0806h	CCx匹配寄存器 (CCx_MATCH) x=1
0807h	CCx匹配寄存器 (CCx_MATCH) x=2
0808h	VBUS匹配寄存器 (VBUS_MATCH)
0809h	VBUS更改状态寄存器 (VBUS_CHG_STS)
080Ah	电源中断状态寄存器 (PWR_INT_STS)
080Bh	调试中断状态寄存器 (DBG_INT_STS)
080Ch–0810h	保留供将来扩展使用
0811h	CC中断允许寄存器 (CC_INT_EN)
0812h	CCx匹配使能寄存器 (CCx_MATCH_EN) x=1
0813h	CCx匹配使能寄存器 (CCx_MATCH_EN) x=2
0814h	VBUS匹配使能寄存器 (VBUS_MATCH_EN)
0815h	电源中断允许寄存器 (PWR_INT_EN)
0816h	调试中断允许寄存器 (DBG_INT_EN)
0817h	匹配去抖寄存器 (MATCH_DEB)
0818h	PD去抖寄存器 (PD_DEB)
0819h	VCONN OCS和反向驱动去抖寄存器 (VCONN_DEB)
081Ah	CCx去抖清除使能寄存器 (CCx_DBCLR_EN) x=1
081Bh	CCx去抖清除使能寄存器 (CCx_DBCLR_EN) x=2
081Ch	VBUS去抖清除使能寄存器 (VBUS_DBCLR_EN)
081Dh	CCx采样使能寄存器 (CCx_SAMP_EN) x=1
081Eh	CCx采样使能寄存器 (CCx_SAMP_EN) x=2
081Fh	保留供将来扩展使用
0820h	CC控制寄存器 (CC_CTL)
0822h	CC阈值x寄存器 (CC_THRx) x=0
0824h	CC阈值x寄存器 (CC_THRx) x=1
0826h	CC阈值x寄存器 (CC_THRx) x=2
0828h	CC阈值x寄存器 (CC_THRx) x=3
082Ah	CC阈值x寄存器 (CC_THRx) x=4
082Ch	CC阈值x寄存器 (CC_THRx) x=5
082Eh	CC阈值x寄存器 (CC_THRx) x=6
0830h	CC阈值x寄存器 (CC_THRx) x=7
0832h	CC去抖寄存器 (CC_DEB)
0834h–083Fh	保留供将来扩展使用
0840h	VBUS控制寄存器 (VBUS_CTL)
0842h	VBUS阈值x寄存器 (VBUS_THRx) x=0
0844h	VBUS阈值x寄存器 (VBUS_THRx) x=1
0846h	VBUS阈值x寄存器 (VBUS_THRx) x=2

表9-9: 系统控制和状态寄存器映射 (续)

地址	寄存器名称 (符号)
0848h	VBUS 阈值x 寄存器 (VBUS_THRx) x=3
084Ah	VBUS 去抖寄存器 (VBUS_DEB)
084Bh	VBUS 关闭寄存器 (VBUS_OFF)
084Ch	VBUS 错误寄存器 (VBUS_ERR)
084Dh	保留供将来扩展使用
084Eh	VBUS VSafe0V 阈值寄存器 (VSAFE0V_THR)
0850h	CFG_SELx 匹配寄存器 (CFG_SELx_MATCH) x=0
0852h	CFG_SELx 匹配寄存器 (CFG_SELx_MATCH) x=1
0854h	CFG_SEL 阈值x 寄存器 (CFG_SEL_THRx) x=0
0856h	CFG_SEL 阈值x 寄存器 (CFG_SEL_THRx) x=1
0858h	CFG_SEL 阈值x 寄存器 (CFG_SEL_THRx) x=2
085Ah	CFG_SEL 阈值x 寄存器 (CFG_SEL_THRx) x=3
085Ch	CFG_SEL 阈值x 寄存器 (CFG_SEL_THRx) x=4
085Eh	CFG_SEL 阈值x 寄存器 (CFG_SEL_THRx) x=5
0860h	CFG_SEL 阈值x 寄存器 (CFG_SEL_THRx) x=6
0862h	CFG_SEL 阈值x 寄存器 (CFG_SEL_THRx) x=7
0864h	CFG_SEL 阈值x 寄存器 (CFG_SEL_THRx) x=8
0866h	CFG_SEL 阈值x 寄存器 (CFG_SEL_THRx) x=9
0868h	CFG_SEL 阈值x 寄存器 (CFG_SEL_THRx) x=10
086Ah	CFG_SEL 阈值x 寄存器 (CFG_SEL_THRx) x=11
086Ch	CFG_SEL 阈值x 寄存器 (CFG_SEL_THRx) x=12
086Eh	CFG_SEL 阈值x 寄存器 (CFG_SEL_THRx) x=13
0870h	CFG_SEL 阈值x 寄存器 (CFG_SEL_THRx) x=14
0872h	CFG_SEL 阈值x 寄存器 (CFG_SEL_THRx) x=15
0874h	CFG_SEL 调试寄存器 (CFG_SEL_DBG)
0875h–0885h	保留供将来扩展使用
0886h	VCONN 放电阈值寄存器 (VCONN_DIS_THR)
0888h	VCONN 放电时间寄存器 (VCONN_DIS_TIME)
088Ah–0BFFh	保留供将来扩展使用

注: 在任何情况下都不能对保留的地址空间执行写操作。否则, 可能导致意外的操作与结果。

UPD360

9.12.1 CC 硬件控制寄存器 (CC_HW_CTL)

地址: 0800h 大小: 16 位

Bit	说明	类型	默认值
15:14	SHORT_DET 定义独立DFP模式下处理短路条件时的行为。 0xb: 启动VBUS放电之前未检测到VBUS短路时维持A0功能。 10b: 器件仅在短路条件消除后才启动VBUS放电。 11b: 器件在短路条件消除或时间等于TCYCLE后启动VBUS放电。 注: 该字段仅对独立DFP操作有意义。	R/W	00b
13	保留	RO	-
12	BLK_PD_MSG 该位将导致CC去抖动器在MAC检测到PD报文时暂停。这对于发送和接收报文均适用。报文处理完成后 0b: 禁止PD报文过滤。 1b: 使能PD报文过滤。 注: 必须使能才能进行冲突检测。	R/W	0b
11	MATCH_DB_UNITS 该位定义匹配去抖寄存器 (MATCH_DEB) 的单位。 0b: 匹配去抖的单位为1.6 ms。 1b: 匹配去抖的单位为100 μs。 注: 1.6 ms是从循环使用两个CC引脚的所有阈值所花费的时间得出的。	R/W	0b
10	DEVICE_MODE 指示器件的当前模式。 0b: 器件处于伙伴模式 1b: 器件处于独立模式 (仅限UPD360-A/UPD360-B)	RO	0b
9	CC_DB_ACTIVE 当该位读回0b时, 将禁止去抖动器。当该位读回1b时, 将使能CC去抖动器。 注: 软件可以轮询该位以确定CC去抖动器何时禁止, 这可以用作编程新配置的条件。	RO	0b
8	DEVICE_STATE 指示器件连接的当前状态。 0b: 器件未连接 1b: 器件已连接	RO	0b

Bit	说明	类型	默认值
7:6	功率容量 指示器件的充电电流容量，由PWR_CAPx引脚定义。（仅限UPD360-A/UPD360-B） 该字段仅对可使用PWR_CAPx引脚的配置有意义（仅限UPD360-A/UPD360-B）。	RO	注9-2
5:3	断开阈值选择 定义当作为独立DFP工作时，应使用哪个CC阈值来确定断开情况（仅限UPD360-A/UPD360-B）。	R/W	注9-2
2	器件角色 0b: 器件配置为UFP。 1b: 器件配置为DFP。 注： 使能CC去抖动器时，不得修改该位。	R/W	注9-3
1	保留	RO	-
0	独立操作 0b: 伙伴模式 1b: 独立模式（仅限UPD360-A/UPD360-B） 独立模式： 这是支持独立操作的工作模式。在独立操作期间，I ² C接口不可用，主机CPU无法管理UPD360。 该位置1时，内部逻辑可以处理以下功能： • VCONN使能/禁止 • 连接检测以及将ATTACH置为有效 • 断开检测以及将ATTACH置为无效 • 方向检测以及将ORIENTATION引脚置为有效/置为无效 • 使能CC去抖动器。	R/W	注9-1

注9-1 选择独立配置时，默认值为1b，否则默认值为0b。

注9-2 默认值由PWR_CAP0和PWR_CAP1引脚共同决定（当配置支持这些引脚时）。否则，默认值为0h。

注9-3 默认值由CFG_SEL0和CFG_SEL1引脚共同决定。当配置指示UFP配置时，该位默认为0b。对于DFP操作，该位默认为1b。当没有指定UFP或DFP时，默认值为0b。

UPD360

9.12.2 CC中断状态寄存器（CC_INT_STS）

地址：0803h 大小：16位

Bit	说明	类型	默认值
7	CC_MATCH_VLD 在通过 CC比较器控制 首次使能CC去抖动器且 CCx匹配寄存器（CCx_MATCH） 中首次匹配变为有效后置为有效。	R/WC	0b
6	RP_CHG 当作为独立UFP工作时，该中断表示检测到DFP通告的RP值状态变化（ <i>仅限UPD360-A/UPD360-B</i> ）。 注： 当器件未配置为独立UFP时，该位为只读位，将始终读回0b。 注： 该输入来自一个脉冲信号，一直持续到该位清零。 注： 更新的电流通告可通过 CC控制寄存器（CC_CTL） 中的 DFP电流通告 字段获取。	R/WC	0b
5	DETACH 表示当器件按照 CC控制寄存器（CC_CTL） 中的 独立操作 字段配置为支持独立模式后发生了断开事件（ <i>仅限UPD360-A/UPD360-B</i> ）。 DFP操作：监视CC引脚以检测断开情况。 UFP操作：监视VBUS以检测断开情况。 注： 该输入来自一个脉冲信号，一直持续到该位清零。	R/WC	0b
4	ATTACH 表示当器件按照 CC控制寄存器（CC_CTL） 中的 独立操作 字段配置为支持独立模式后发生了连接事件（ <i>仅限UPD360-A/UPD360-B</i> ）。 DFP操作：监视CC引脚以检测连接情况。 UFP操作：监视CC引脚和VBUS以检测连接情况。 注： 该输入来自一个脉冲信号，一直持续到该位清零。	R/WC	0b
3:2	保留	RO	-
1	CC2_MATCH_CHG 表示相应 CCx更改状态寄存器（CCx_CHG_STS） 的状态发生了变化。 注： 该输入来自一个脉冲信号，一直持续到该位清零。	RO	0b
0	CC1_MATCH_CHG 表示相应 CCx更改状态寄存器（CCx_CHG_STS） 的状态发生了变化。 注： 该输入来自一个脉冲信号，一直持续到该位清零。	RO	0b

9.12.3 CCX更改状态寄存器（CCX_CHG_STS）

地址：x=1: 0804h大小：8位

x=2: 0805h

Bit	说明	类型	默认值
7:0	CCx更改状态 各个位置1时，表示 CCx匹配寄存器（CCx_MATCH） 中的相应位发生了更改。 写入1b会将相应状态位清零。	R/WC	0h

9.12.4 CCX匹配寄存器（CCX_MATCH）

地址：x=1: 0806h大小：8位

x=2: 0807h

Bit	说明	类型	默认值
7:0	CCx阈值匹配（CCx_MATCH） 各个位置1时，表示 CC阈值x寄存器（CC_THRx） 中编程的相应阈值已发生匹配。当测量的电压超过编程的阈值时，即可确定匹配。 处于Unattached状态时，这些寄存器将在 匹配去抖寄存器（MATCH_DEB） 之后更新。处于AttachWait状态时，如果发现vOpen的持续时间长于 PD去抖寄存器（PD_DEB） 中的时间，这些寄存器将在 PD去抖寄存器（PD_DEB） 之后更新。处于Attached Source/Attached Sink状态时（例如检测断开），匹配寄存器将在 PD去抖寄存器（PD_DEB） 之后更新。 注： 在非独立模式下工作时，该寄存器将始终读为默认值，直至使能CC比较器。 注： 该寄存器的内容将按照 CCx去抖清除使能寄存器（CCx_DBCLR_EN） 中的设置进行去抖，否则应显示原始值。	RO	注9-4

注9-4 根据**CC控制寄存器（CC_CTL）**中的**器件角色**字段，当器件配置为DFP时，默认值为FFh；当器件配置为UFP时，默认值为00h。

9.12.5 VBUS匹配寄存器（VBUS_MATCH）

地址：0808h 大小：8位

指示VBUS_DET引脚的哪些VBUS阈值发生匹配。当测量的电压超过编程的阈值时，即可确定匹配。

注： 在非独立模式下工作时，该寄存器将始终读为0h，直至使能VBUS比较器。

注： 该寄存器的内容应按照VBUS去抖清除使能寄存器（VBUS_DBCLR_EN）中的设置进行去抖。

Bit	说明	类型	默认值
7:6	保留	RO	-
5	VBUS阈值3匹配（VBUS3_THR_MATCH）	RO	0b
4	VBUS阈值2匹配（VBUS2_THR_MATCH）	RO	0b
3	VBUS阈值1匹配（VBUS1_THR_MATCH）	RO	0b
2	VBUS阈值0匹配（VBUS0_THR_MATCH）	RO	0b
1	保留	RO	-
0	VBUS VSafe0v匹配（VSAFE0V_THR_MATCH）	RO	0b

9.12.6 VBUS更改状态寄存器（VBUS_CHG_STS）

地址：0809h 大小：8位

Bit	说明	类型	默认值
7:0	VBUS更改状态（VBUS_CHG_STS） 各个位置1时，表示VBUS匹配寄存器（VBUS_MATCH）中的相应位发生了更改。 写入1b会将相应状态位清零。	R/WC	0h

9.12.7 电源中断状态寄存器 (PWR_INT_STS)

地址: 080Ah 大小: 8位

Bit	说明	类型	默认值
7	VBUS_MATCH_VLD 在通过 VBUS比较器控制 首次使能 VBUS去抖动器 且 VBUS匹配寄存器 (VBUS_MATCH) 中首次匹配变为有效后置为有效。	R/WC	0b
6	VCONN放电错误 (VCONN_DISCH_ERR) 该位置1时表示检测到 VCONN放电错误 。 该位仅在独立DFP模式下可用，其他情况下保留。 当 VCONN放电控制 设置为00b或01b时，将跟踪放电错误。如果在经过的时间超过 VCONN放电时间 (VCONN_DIS_TIME) 后， VCONN 无法降至 VCONN放电阈值 (VCONN_DIS_THR) 以下，则认为出现放电错误。	R/WC	0b
5	CC2反向驱动错误 置1时，表示在 CC2 引脚上检测到反向驱动。 注： 该输入来自一个电平信号，一直持续到错误状态停止。	R/WC	0b
4	VBUS放电错误 置1时，表示尝试使 VBUS 放电时所经过的时间已超过 VBUS关闭寄存器 (VBUS_OFF) 中定义的值。 注： 该输入来自一个脉冲信号，一直持续到该位清零。	R/WC	0b
3	VCONN2 FET电源 集成 VCONN2 FET 使能后可提供电源。 该位仅对于独立操作或主机明确使能 VCONN 的情况有用。 注： 该输入来自一个电平信号，一直持续到 VCONN 不再存在。	R/WC	0b
2	VCONN1 FET电源 集成 VCONN1 FET 使能后可提供电源。 该位仅对于独立操作或主机明确使能 VCONN 的情况有用。 注： 该输入来自一个电平信号，一直持续到 VCONN 不再存在。	R/WC	0b
1	CC1反向驱动错误 置1时，表示在 CC1 引脚上检测到反向驱动。 注： 该输入来自一个电平信号，一直持续到错误状态停止。	R/WC	0b
0	VCONN过流错误 表示在集成 VCONN FET 上检测到过流。 注： 该输入来自一个电平信号，一直持续到错误状态停止。	R/WC	0b

UPD360

9.12.8 调试中断状态寄存器 (DBG_INT_STS)

地址: 080Bh 大小: 8位

Bit	说明	类型	默认值
7:4	保留	RO	0h
3	VCONN_DISCH_STS 置1时, 表示先前为VCONN供电的CC引脚正在放电。 该位仅在独立DFP模式下可用, 其他情况下保留。	R/WC	0b
2	VBUS_DISCH 置1时, 表示VBUS正在放电。 注: 该输入来自一个电平信号, 一直持续到放电完成。	R/WC	0b
1	CFG_SEL1完成 置1时, 表示已测量所有CFG_SEL阈值x寄存器 (CFG_SEL_THRx) 来响应通过VBUS比较器控制使能的采样CFG_SEL1引脚的请求, 结果可在相应CFG_SELx匹配寄存器 (CFG_SELx_MATCH) 中读取 (仅限UPD360-A/UPD360-B)。 注: 该输入来自一个脉冲信号, 一直持续到该位清零。	R/WC	0b
0	CFG_SEL0完成 置1时, 表示已测量所有CFG_SEL阈值x寄存器 (CFG_SEL_THRx) 来响应通过VBUS比较器控制使能的采样CFG_SEL0引脚的请求, 结果可在相应CFG_SELx匹配寄存器 (CFG_SELx_MATCH) 中读取。 注: 该输入来自一个脉冲信号, 一直持续到该位清零。	R/WC	0b

9.12.9 CC中断允许寄存器 (CC_INT_EN)

地址: 0811h 大小: 8位

Bit	说明	类型	默认值
7:4	CC中断允许 为“0”时, 可阻止产生相应的中断。	R/W	0000b
3:2	保留	-	00b
1:0	CC中断允许 为“0”时, 可阻止产生相应的中断。	R/W	00b

9.12.10 CCX匹配使能寄存器 (CCX_MATCH_EN)

地址: x=1: 0812h 大小: 8位
x=2: 0813h

Bit	说明	类型	默认值
7:0	CCx匹配使能 置1时, CCx更改状态寄存器 (CCx_CHG_STS) 中的相应位可将相应CCx匹配更改中断 (CC1_MATCH_CHG/CC2_MATCH_CHG) 置为有效。	R/W	00h

9.12.11 VBUS匹配使能寄存器 (VBUS_MATCH_EN)

地址: 0814h 大小: 8位

置1时, **VBUS更改状态寄存器 (VBUS_CHG_STS)** 中的相应位可将**VBUS_INT**中断置为有效。

Bit	说明	类型	默认值
7:6	保留	RO	-
5	VBUS匹配使能[5]	R/W	0b
4	VBUS匹配使能[4]	R/W	0b
3	VBUS匹配使能[3]	R/W	0b
2	VBUS匹配使能[2]	R/W	0b
1	VBUS匹配使能[1]	RO	0b
0	VBUS匹配使能[0]	R/W	0b

9.12.12 电源中断允许寄存器 (PWR_INT_EN)

地址: 0815h 大小: 8位

Bit	说明	类型	默认值
7:0	电源中断允许[7:0] 为“0”时, 可阻止产生相应的中断。	R/W	0h

UPD360

9.12.13 调试中断允许寄存器 (DBG_ENT_EN)

地址: 0816h 大小: 8位

Bit	说明	类型	默认值
7:4	保留	RO	0h
3	调试中断允许3 为“0”时, 可阻止产生相应的中断。 该位仅在独立DFP模式下可用, 其他情况下保留。	R/W	0b
2:0	调试中断允许[2:0] 为“0”时, 可阻止产生相应的中断。	R/W	0h

9.12.14 匹配去抖寄存器 (MATCH_DEB)

地址: 0817h 大小: 8位

Bit	说明	类型	默认值
7:0	匹配去抖 定义在非独立模式下工作时更新 CCx匹配寄存器 (CCx_MATCH) 之前使用的去抖周期。 该寄存器的单位由 CC硬件控制寄存器 (CC_HW_CTL) 中的 MATCH_DB_UNITS 位确定。该位清零时, 单位为1.6 ms。该位置1时, 单位为100 μ s。 注: 使能CC去抖动器时, 不得修改该寄存器。 注: 实际的去抖时间可能为编程的周期时间 ± 1 。 注: 该CSR中编程的值应至少等于 CCx采样使能寄存器 (CCx_SAMP_EN) 中所使能的阈值数。这仅仅为 MATCH_DB_UNITS 设置为1b时会出现的问题。	R/W	2h

9.12.15 PD去抖寄存器 (PD_DEB)

地址: 0818h 大小: 8位

Bit	说明	类型	默认值
7:0	PD去抖 (PD_DEB) 用于实现 $t_{pdDebounce}$ 的周期。 注: 使能CC去抖动器时, 不得修改该寄存器。 注: 该寄存器的单位为1 ms。	R/W	Ah

9.12.16 VCONN OCS和反向驱动去抖寄存器（VCONN_DEB）

地址：0819h 大小：8位

Bit	说明	类型	默认值
7:0	VCONN和反向驱动去抖（VCONN_DEB） 用于对在VCONN FET上检测到的过流以及在CC引脚上检测到的反向驱动执行去抖操作时使用的周期。 注： 该寄存器的单位为1 ms。 注： VCONN OCS 使能置1时，不得更改该寄存器。	R/W	2h

9.12.17 CCx去抖清除使能寄存器（CCx_DBCLR_EN）

地址：x=1: 081Ah x=2: 081Bh 大小：8位

Bit	说明	类型	默认值
7:0	CC去抖清除使能（CC_DBCLR_DEB） 当某个位置1时，CC去抖动器中应包含相应阈值。而当某个位清零时，去抖动器则不应再考虑相应阈值。 当CCx_DBCLR_EN位在运行时置1时，如果当前的原始匹配向量（对应于新的CCx_DBCLR_EN）和先前的原始匹配向量（对应于旧的CCx_DBCLR_EN）存在不匹配，则DB将发生复位。 注： 在运行时将该寄存器中的位清零不会复位去抖动器。 注： CC去抖动器包含CC1/CC2引脚。在任一引脚上检测到阈值发生变化均会导致去抖动器复位。 注： 即使该寄存器可能会即时更改，内部逻辑也只能在扫描周期结束时使能更改，这受CC1/CC2引脚是否主动采样以及CCx采样使能寄存器（CCx_SAMP_EN）的影响。	R/W	注 9-5

注9-5 默认值取决于器件的配置（如表9-10所示，具体取决于CFG_SEL0、PWR_CAP0和PWR_CAP1引脚）。

应用注意事项： 将CCx去抖清除使能寄存器（CCx_DBCLR_EN）中的某个位清零将导致CCx匹配寄存器（CCx_MATCH）中的相应位在相关阈值的状态发生变化时立即更新。这将导致CCx更改状态寄存器（CCx_CHG_STS）中的状态发生变化并且使CC_INT置为有效（如果使能）。为防止出现这种情况，应通过将相关阈值清零在CCx去抖清除使能寄存器（CCx_DBCLR_EN）之前更新CCx匹配使能寄存器（CCx_MATCH_EN）。

表9-10: CCX_DBLCLR_DEB默认值

配置	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
独立UFP (仅限UPD360-A/UPD360-B)	1	0	1	0	1	0	0	0
独立DFP (默认电流) (仅限UPD360-A/UPD360-B)	1	0	0	0	0	1	0	0
独立DFP (1.5A) (仅限UPD360-A/UPD360-B)	0	1	0	0	0	1	0	0
独立DFP (3.0A) (仅限UPD360-A/UPD360-B)	0	0	0	1	0	0	1	0
其他	0	0	0	0	0	0	0	0

9.12.18 VBUS去抖清除使能寄存器 (VBUS_DBCLR_EN)

地址: 081Ch 大小: 8位

当某个位置1时，VBUS去抖器中应包含相应阈值。而当某个位清零时，去抖器则不应再考虑相应阈值。
当VBUS_DBCLR_EN位在运行时置1时，如果当前的原始匹配向量（对应于新的VBUS_DBCLR_EN）和先前的原始匹配向量（对应于旧的VBUS_DBCLR_EN）存在不匹配，则DB将发生复位。
注： 在运行时将该寄存器中的位清零不会复位去抖器。

应用注意事项： 将VBUS去抖清除使能寄存器 (VBUS_DBCLR_EN) 中的某个位清零将导致VBUS匹配寄存器 (VBUS_MATCH) 中的相应位在相关阈值的状态发生变化时立即更新。这将导致VBUS更改状态寄存器 (VBUS_CHG_STS) 中的状态发生变化并且使VBUS_INT置为有效（如果使能）。为防止出现这种情况，应通过将相关阈值清零在VBUS去抖清除使能寄存器 (VBUS_DBCLR_EN) 之前更新VBUS匹配寄存器 (VBUS_MATCH)。

BIT	说明	类型	默认值
7:6	保留	RO	-
5	VBUS去抖清除使能 (VBUS3_DBCLR_DEB)	R/W	注9-6
4	VBUS去抖清除使能 (VBUS2_DBCLR_DEB)	R/W	注9-6
3	VBUS去抖清除使能 (VBUS1_DBCLR_DEB)	R/W	注9-6
2	VBUS去抖清除使能 (VBUS0_DBCLR_DEB)	R/W	注9-6
1	保留	RO	-
0	VSAFE0V去抖清除使能9 (VSAFE0V_DBCLR_DEB)	R/W	注9-6

注9-6 默认值取决于器件的配置（如表9-11所示，具体取决于CFG_SEL0引脚）。

表9-11: VBUS_DBCLR_EN默认值

配置	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
独立UFP (仅限UPD360-A/UPD360-B)	1	0	1	1	0	0	0	0
独立DFP (仅限UPD360-A/UPD360-B)	1	0	1	1	0	0	0	0
其他	0	0	0	0	0	0	0	0

9.12.19 CCX采样使能寄存器 (CCX_SAMP_EN)

地址: x=1: 081Dh 大小: 8位
x=2: 081Eh

BIT	说明	类型	默认值
7:0	CC采样使能 (CC_SAMP_EN) 当某个位置1时，CC去抖动器将采样相应CC阈值。当某个位清除为0b时，CCx匹配寄存器 (CCx_MATCH) 和CCx更改状态寄存器 (CCx_CHG_STS) 中的相应位将始终读为0b。 该寄存器可通过仅采样所关注的阈值来缩短阈值测量延时。 注: 对于独立DFP/UFP操作，该寄存器可保持设置为FFh，因为这种情况下的CC阈值采样延时不是限制因素。 注: 要实现避免冲突，必须使用该寄存器。	R/W	FFh

UPD360

9.12.20 CC控制寄存器 (CC_CTL)

地址：0820h 大小：16 位

该寄存器控制相应CC1/CC2引脚的下拉电阻和电流源。

Bit	说明	类型	默认值
15	RA 检测 置 1 时，表示已检测到 RA 电阻。 当 独立操作 配置为独立模式时，该位由器件置 1。器件在检测到连接后更新该字段。	R/WC	0b
14:13	CC 比较器控制 00b: CC 比较器和 DAC 掉电 01b: CC 比较器采样 CC1 10b: CC 比较器采样 CC2 11b: CC 比较器采样 CC1 和 CC2 注： 每 100 μ s 采样一次。当设置为值 00b 时，电流源参考也应关闭。 注： 该字段为只读字段，使能独立模式时的内容为 11b。 注： 该字段和 CCx 采样使能寄存器 (CCx_SAMP_EN) 配合使用来约束所使能 CC 引脚上监视的阈值数。	注 9-7	00b
12	CC 通信选择 0b: CC1 用于基带通信。 1b: CC2 用于基带通信。 当 独立操作 配置为独立模式时，该位由器件置 1。器件在检测到连接后更新该字段。 该位为只读位，反映了由内部逻辑决定的状态。在独立模式下，该位匹配 ORIENTATION 引脚的状态（仅限 UPD360-A/UPD360-B）。	注 9-7	0b
11:10	CC2 RP 值 00b: 禁止 RP 电流源 01b: 使能 RP 电流源，默认 USB 电流 10b: 使能 RP 电流源，1.5A 11b: 使能 RP 电流源，3.0A 控制 CC2 引脚的 RP 值。	R/W	00b 注 9-8
9:8	CC1 RP 值 00b: 禁止 RP 电流源 01b: 使能 RP 电流源，默认 USB 电流 10b: 使能 RP 电流源，1.5A 11b: 使能 RP 电流源，3.0A 控制 CC1 引脚的 RP 值。	R/W	00b 注 9-8

Bit	说明	类型	默认值
7:6	DFP 电流通告 当 独立操作 配置为独立模式且器件配置为UFP 器件角色 时，该字段表示DFP的通告电流。否则，该字段将读回00b。 器件在检测到连接后更新该字段。 0xb: RP 电流通告默认USB 电流 10b: RP 电流通告 1.5A 11b: RP 电流通告 3.0A	RO	00b
5	保留	RO	-
4:3	CC2 下拉值 00b: 选择电池电量耗尽 RD 电阻 01b: 选择微调 RD 电阻 10b: 选择微调 RA 电阻。 11b: 断开。	R/W	注 9-9
2	保留	RO	-
1:0	CC1 下拉值 00b: 选择电池电量耗尽 RD 电阻 01b: 选择微调 RD 电阻 10b: 选择微调 RA 电阻。 11b: 断开。	R/W	注 9-9

- 注 9-7 在独立模式下工作时，该位为只读位。否则为读/写位。
- 注 9-8 在独立DFP模式下工作时，该字段的默认值由**PWR_CAP0**和**PWR_CAP1**引脚共同决定（见第9.8.1节“配置选择”）（仅限UPD360-A/UPD360-B）。否则，默认值为00b。
- 注 9-9 对于独立DFP和独立UFP，值为00b，直至系统复位完成。之后，独立DFP的默认值为11b，而独立UFP的默认值为01b。

9.12.21 CC 阈值X寄存器（CC_THRX）

地址:	x=0: 0822h	大小:	16 位
	x=1: 0824h		
	x=2: 0826h		
	x=3: 0828h		
	x=4: 082Ah		
	x=5: 082Ch		
	x=6: 082Eh		
	x=7: 0830h		

Bit	说明	类型	默认值
15:10	保留	RO	-
9:0	CC 阈值X（CC_THRX） CC 阈值X 寄存器。 注： 该寄存器的单位约为2.44 mV（由2.5V/1024得出）。 注： 使能CC去抖动器时，不得修改该寄存器。	R/W	注 9-10

- 注 9-10 默认值视情况而定，如表9-12所示。

UPD360

表9-12: CC_THR默认值

CC_THR	Type-C 阈值	R/2R分压比	值
0	0.2	0.13	55
1	0.4	0.27	109
2	0.66	0.44	180
3	0.8	0.53	219
4	1.23	0.82	336
5	1.6	1.07	437
6	2.6	1.73	710
7	3	2	820

应用注意事项： 在更新这些寄存器之前，必须关闭CC比较器。

9.12.22 CC去抖寄存器（CC_DEB）

地址： 0832h 大小： 8位

Bit	说明	类型	默认值
7:0	CC去抖（CC_DEB） 用于实现 $t_{CCDebounce}$ 的周期。 注： 使能CC去抖驱动器时，不得修改该寄存器。 注： 该寄存器的单位为10 ms。	R/W	Fh

9.12.23 VBUS控制寄存器（VBUS_CTL）

地址： 0840h 大小： 16位

Bit	说明	类型	默认值
15:12	保留	RO	-
11	CC反向驱动使能 使能监视两个CC引脚的反向驱动条件。 当检测到反向驱动时，如果在错误的CC引脚上使能VCONN FET，将自动禁止反向驱动。 0b: 禁止CC反向驱动 1b: 使能CC反向驱动	R/W	注9-11
10	IBUS_LOW 确定当PPC的DISCH_SEL清零且VBUS正在放电时IBUS Low在独立模式下是否置为有效。 0b: 不将IBUS Low置为有效 1b: 将IBUS Low置为有效	R/W	0b

Bit	说明	类型	默认值
9:8	OCS_MIN 定义工作在独立DFP模式下时 OCS_N 可保证的最短有效时间（仅限UPD360-A/UPD360-B）。 00b: 5 ms 01b: 10 ms 10b: 20 ms 11b: 30 ms	R/W	00b
7	VBUS_DB_ACTIVE 当该位读回0b时，将禁止去抖动器。当该位读回1b时，将使能VBUS去抖动器。 注： 固件轮询该位以确定去抖动器何时禁止，从而确定何时可编程新的配置。	RO	0b
6	VCONN OCS使能 使能监视内部VCONN FET的过流条件。 0b: 禁止VCONN OCS监视器 1b: 使能VCONN OCS监视器	R/W	注9-11
5:4	VCONN放电控制 该字段确定VCONN放电行为。它仅在DFP独立模式下有意义，在伙伴模式下工作时不得由固件使用。 放电发生在之前为VCONN供电的CC引脚上。 00b: 使VCONN放电，直至达到通过 VCONN放电阈值寄存器（VCONN_DIS_THR） 定义的任一阈值或超出 VCONN放电时间寄存器（VCONN_DIS_TIME） 规定的时间。 01b: 不支持VCONN放电。 10b: 使VCONN放电，直至达到通过 VCONN放电阈值寄存器（VCONN_DIS_THR） 定义的阈值。 11b: 使VCONN放电，并持续通过 VCONN放电时间寄存器（VCONN_DIS_TIME） 规定的时间。 注： 对于选项0xb，如果定时器超时而VCONN尚未放电至 VCONN放电时间寄存器（VCONN_DIS_TIME） 以下，则将检测到VCONN放电错误。	R/W	00b
3	VCONN2控制 使能VCONN2 FET。 0b: 禁止VCONN2 FET 1b: 使能VCONN2 FET 该位在独立模式下没有任何意义（见 独立操作 ）。 VCONN放电错误（VCONN_DISCH_ERR） 置为有效后，该位将在去抖VCONN OCS事件发生时自动清零。	R/W	0b

UPD360

Bit	说明	类型	默认值
2	VCONN1控制 使能VCONN1 FET。 0b: 禁止VCONN1 FET 1b: 使能VCONN1 FET 该位在独立模式下没有任何意义（见 独立操作 ）。 VCONN放电错误（VCONN_DISCH_ERR） 置为有效后，该位将在去抖VCONN OCS事件发生时自动清零。	R/W	0b
1:0	VBUS比较器控制 00b: 禁止比较器和DAC 01b: 采样VBUS 10b: 采样CFG_SEL0 11b: 采样CFG_SEL1（仅限UPD360-A/UPD360-B） CFG_SEL0或CFG_SEL1的采样完成后，该寄存器将自行复位为00b并禁止VBUS比较器。 注： 作为独立UFP工作时，该字段将由硬件强制为01b。	注9-12	00b

注9-11 在独立DFP模式下工作时，该默认值为1。否则，默认值为0。

注9-12 在独立UFP模式下工作时，该字段为只读字段并读回01b。在独立模式下，该值不表示正在主动测量VBUS。是否正在主动测量VBUS可通过读取[VBUS_DB_ACTIVE](#)来确定。

9.12.24 VBUS阈值X寄存器（VBUS_THRX）

地址： x=0: [0842h](#) 大小： 16位
 x=1: [0844h](#)
 x=2: [0846h](#)
 x=3: [0848h](#)

Bit	说明	类型	默认值
15:10	保留	RO	-
9:0	VBUS阈值X（VBUS_THRX） VBUS阈值X寄存器。 阈值的低字节必须在高字节之前写入。当发生第二次写入时，将更新全部10位阈值。 注： 该寄存器的单位约为2.44 mV（由2.5V FS得出）。	R/W	注9-13

注9-13 [表9-13](#)中定义了默认值。

表9-13: VBUS_THR默认值

VBUS_THR	VBUS阈值	1R/9R分压比	值
0	3.67	0.36	148
1	5.5	0.54	222
2	5.5	0.54	222
3	5.5	0.54	222

9.12.25 VBUS去抖寄存器（VBUS_DEB）

地址：084Ah 大小：8位

该寄存器的单位为1 ms。

Bit	说明	类型	默认值
7:0	VBUS去抖（VBUS_DEB） 表示VBUS阈值比较器的去抖时间间隔。 注： 使能VBUS去抖动器时，不得修改该寄存器。	R/W	1h

9.12.26 VBUS关闭寄存器（VBUS_OFF）

地址：084Bh 大小：8位

该寄存器的单位为10 ms。

Bit	说明	类型	默认值
7:0	VBUS关闭 定义VBUS_DET放电至VSafe0V以下后的时序。	R/W	10h

9.12.27 VBUS错误寄存器（VBUS_ERR）

地址：084Ch 大小：8位

Bit	说明	类型	默认值
7:3	保留	RO	-
3	VCONN放电 置1时，将使能VBUS放电错误，使器件处于错误状态。	R/W	0b
2	VBUS放电 置1时，将使能VBUS放电错误，使器件处于错误状态。 注： 当选择PPC进行VBUS放电时，该位不适用。	R/W	0b
1	VCONN OCS 置1时，将使能CC OCS错误，使器件处于错误状态。	R/W	0b
0	CC反向驱动 置1时，将使能CC反向驱动错误，使器件处于错误状态。	R/W	0b

9.12.28 VBUS VSAFE0V 阈值寄存器 (VSAFE0V_THR)

地址：084Eh 大小：16 位

BIT	说明	类型	默认值
15:10	保留	RO	-
9:0	VSAFE0V 阈值 (VSAFE0V_THR) VSAFE0V 阈值寄存器。 阈值的低字节必须 在高字节之前写入。当发生第二次写入时，将更新全部 10 位 阈值。 注： 该寄存器的单位约为 2.44 mV（由 2.5V FS 得出）。	R/W	注 9-14

注 9-14 [表 9-14](#) 中定义了默认值。

表 9-14: VSAFE0V_THR 默认值

VSAFE0V_THR	VBUS 阈值	1R/9R 分压比	值
VSAFE0V_THR	0.8	0.08	32

应用注意事项： 使能 VBUS 比较器时，该寄存器可以通过软件动态写入，前提是遵循寄存器说明中定义的更新规则。

9.12.29 CFG_SELX 匹配寄存器 (CFG_SELX_MATCH)

地址：x=0: 0850h x=1: 0852h 大小：16 位

Bit	说明	类型	默认值
15:0	配置选择 X 匹配 (CFG_SELX_MATCH) 指示 CFG_SELx 引脚上匹配哪些配置选择阈值。当测量的电压超过编程的阈值 时，即可确定匹配。请参见第 81 页的第 9.8.1 节“配置选择”。	RO	0h

9.12.30 CFG_SEL 阈值X寄存器 (CFG_SEL_THRX)

地址:

x=0: 0854h
x=1: 0856h
x=2: 0858h
x=3: 085Ah
x=4: 085Ch
x=5: 085Eh
x=6: 0860h
x=7: 0862h
x=8: 0864h
x=9: 0866h
x=10: 0868h
x=11: 086Ah
x=12: 086Ch
x=13: 086Eh
x=14: 0870h
x=15: 0872h

大小:

16 位

共支持使用 16 个阈值来解码CFG_SELx引脚上的电阻值。

Bit	说明	类型	默认值
15:10	保留	RO	-
9:0	CFG_SEL 阈值 (CFG_SEL_THR) 注：该寄存器的单位约为2.44 mV（由2.5V FS得出）。	R/W	注9-15

注9-15 表9-15中定义了默认值。

表9-15: CFG_SEL_THR 默认值

CFG_SEL_THR	默认值
0	32
1	96
2	160
3	224
4	288
5	352
6	416
7	480
8	544
9	608
10	672
11	736
12	800
13	864
14	928
15	992

UPD360

9.12.31 CFG_SEL 调试寄存器 (CFG_SEL_DBG)

地址: 0874h 大小: 8位

Bit	说明	类型	默认值
7:4	CFG_SEL1_VAL 该寄存器存储器件在系统级复位事件之后采样CFG_SEL1引脚时匹配的最高阈值的快照 (仅限UPD360-A/UPD360-B)。	RO	0h
3:0	CFG_SEL0_VAL 该寄存器存储器件在系统级复位事件之后采样CFG_SEL0引脚时匹配的最高阈值的快照。	RO	0h

9.12.32 VCONN 放电阈值寄存器 (VCONN_DIS_THR)

地址: 0886h 大小: 16位

Bit	说明	类型	默认值
15:10	保留	RO	-
9:0	VCONN 放电阈值 (VCONN_DIS_THR) 该寄存器定义了独立DFP模式下使VCONN放电所使用的阈值。 注: 该寄存器的单位约为2.44 mV (由2.5V/1024得出)。	R/W	41h 注9-16

注9-16 考虑到R/2R分压比, 41相当于150 mV。

9.12.33 VCONN 放电时间寄存器 (VCONN_DIS_TIME)

地址: 0888h 大小: 16位

Bit	说明	类型	默认值
7:0	VCONN 放电时间 (VCONN_DIS_TIME) 定义为VCONN供电的CC引脚放电的时间。 注: 该寄存器的单位为10 ms。	R/W	04h

10.0 基带CC接口 (BCI)

器件集成了基带CC接口 (Baseband CC Interface, BCI) 以方便USB供电通信。该模块桥接PD MAC/BMC和模拟前端。基带通信由与BCI接口的PD MAC发起。BCI实现控制TX基带组件所需的数字功能。

10.1 基带TX数据流

BCI的主要职责是生成基带通信所需的波形。为此，BMC提供了一组共八个寄存器，用于定义所生成BMC信号从低电平到高电平以及从高电平到低电平的转换。

当收到从低电平转换到高电平的指示时，BCI将逐步执行所有BB TX上升x寄存器 (BB_RX_RISEx)。同样，当收到从高电平转换到低电平的指示时，BCI将逐步执行所有BB TX下降x寄存器 (BB_TX_FALLx)。BCI始终首先在BB_TX_RISE0或BB_RX_FALL0上提供值。

应用注意事项： 如果希望为此使用少于十二个惟一值，用户可以复制相应值。

应遵循以下步骤对BCI进行数据传输编程：

1. 软件编程BB TX上升x寄存器 (BB_RX_RISEx) 和BB TX下降x寄存器 (BB_TX_FALLx)，定义上升沿和下降沿的压摆率。
2. 软件使能PD MAC。
3. PD MAC通过固件启动，或通过数据传输 (GoodCRC) 自主启动。PD MAC指示BCI使BB TX模拟组件退出掉电模式。
4. 模拟组件经过足够长的时间上电后，PD MAC开始向驱动模拟组件的BMC编码器传输。
5. 如果MAC请求一个上升沿，BCI将逐步执行BB TX上升x寄存器 (BB_RX_RISEx)。或者，如果MAC请求一个下降沿，BCI将逐步执行BB TX下降x寄存器 (BB_TX_FALLx)。
6. 当PD MAC指示传输已完成时，BCI将使TX模拟组件掉电。

10.2 基带RX数据流

BCI从RX模拟前端接收基带RX数据，并将该数据与软件编程的阈值进行比较。CC RX DAC值定义了用于接收基带数据的跳变点。根据器件的工作模式 (电流源电源、电流阱电源和电源呈中性) 的PD规范中的定义，该字段应编程为比RX眼图中心低175 mV。

要编程所需的跳变点，必须将RX DAC使能位置1且必须编程CC RX DAC控制寄存器 (CC_RX_DAC_CTL) 中的CC RX DAC值字段。

10.3 基带CC接口寄存器

本节详细介绍了基带CC接口寄存器。有关器件的完整寄存器映射的概述，请参见第19页的第4.0节“寄存器映射”。

表10-1： 基带CC接口寄存器映射

地址	寄存器名称（符号）
2800h	CC RX DAC控制寄存器（CC_RX_DAC_CTL）
2802h	CC RX DAC过滤器寄存器（CC_RX_DAC_FILT）
2803h	保留供将来扩展使用
2804h	CC TX DAC过滤器寄存器（CC_TX_DAC_FILT）
2805h-280Fh	保留供将来扩展使用
2810h	BB TX上升x寄存器（BB_RX_RISEx） x=0
2812h	BB TX上升x寄存器（BB_RX_RISEx） x=1
2814h	BB TX上升x寄存器（BB_RX_RISEx） x=2
2816h	BB TX上升x寄存器（BB_RX_RISEx） x=3
2818h	BB TX上升x寄存器（BB_RX_RISEx） x=4
281Ah	BB TX上升x寄存器（BB_RX_RISEx） x=5
281Ch	BB TX上升x寄存器（BB_RX_RISEx） x=6
281Eh	BB TX上升x寄存器（BB_RX_RISEx） x=7
2820h	BB TX上升x寄存器（BB_RX_RISEx） x=8
2822h	BB TX上升x寄存器（BB_RX_RISEx） x=9
2824h	BB TX上升x寄存器（BB_RX_RISEx） x=10
2826h	BB TX上升x寄存器（BB_RX_RISEx） x=11
2828h-282Fh	保留供将来扩展使用
2830h	BB TX下降x寄存器（BB_TX_FALLx） x=0
2832h	BB TX下降x寄存器（BB_TX_FALLx） x=1
2834h	BB TX下降x寄存器（BB_TX_FALLx） x=2
2836h	BB TX下降x寄存器（BB_TX_FALLx） x=3
2838h	BB TX下降x寄存器（BB_TX_FALLx） x=4
283Ah	BB TX下降x寄存器（BB_TX_FALLx） x=5
283Ch	BB TX下降x寄存器（BB_TX_FALLx） x=6
283Eh	BB TX下降x寄存器（BB_TX_FALLx） x=7
2840h	BB TX下降x寄存器（BB_TX_FALLx） x=8
2842h	BB TX下降x寄存器（BB_TX_FALLx） x=9
2844h	BB TX下降x寄存器（BB_TX_FALLx） x=10
2846h	BB TX下降x寄存器（BB_TX_FALLx） x=11
2848h-2BFFh	保留供将来扩展使用

注： 在任何情况下都不能对保留的地址空间执行写操作。否则，可能导致意外的操作与结果。

10.3.1 CC RX DAC控制寄存器（CC_RX_DAC_CTL）

地址：2800h 大小：16位

Bit	说明	类型	默认值
15	RX DAC使能 0：禁止CC RX DAC 1：使能CC RX DAC	R/W	0b
14:10	保留	RO	-
9:0	CC RX DAC值 该寄存器定义了用于接收基带数据的跳变点。 注： 该DAC的满量程范围为1.8V。 注： DAC必须编程为比期望的RX眼图中心低175 mV。	R/W	0h

10.3.2 CC RX DAC过滤器寄存器（CC_RX_DAC_FILT）

地址：2802h 大小：8位

Bit	说明	类型	默认值
7:2	保留	RO	-
1	选择CC Rx过滤器配置	R/W	0b
0	CC RX DAC过滤器使能	R/W	0b

10.3.3 CC TX DAC过滤器寄存器（CC_TX_DAC_FILT）

地址：2804h 大小：8位

Bit	说明	类型	默认值
7:5	保留	RO	-
4	CC TX过滤器使能 使能CC TX过滤器和驱动器。	R/W	0
3:0	CC TX过滤器 选择CC TX过滤器带宽。	R/W	8h

10.3.4 BB TX上升X寄存器 (BB_RX_RISEX)

地址:	x=0: 2810h	大小:	16 位
	x=1: 2812h		
	x=2: 2814h		
	x=3: 2816h		
	x=4: 2818h		
	x=5: 281Ah		
	x=6: 281Ch		
	x=7: 281Eh		
	x=8: 2820h		
	x=9: 2822h		
	x=10: 2824h		
	x=11: 2826h		

BB TX上升寄存器定义了上升沿上基带波形的特性。

Bit	说明	类型	默认值
15:10	保留	RO	-
9:0	BB TX上升值 在实现基带传输上升沿时提供给 CC TX DAC的代码。	R/W	0h

10.3.5 BB TX下降X寄存器 (BB_TX_FALLX)

地址:	x=0: 2830h	大小:	16 位
	x=1: 2832h		
	x=2: 2834h		
	x=3: 2836h		
	x=4: 2838h		
	x=5: 283Ah		
	x=6: 283Ch		
	x=7: 283Eh		
	x=8: 2840h		
	x=9: 2842h		
	x=10: 2844h		
	x=11: 2846h		

BB TX下降寄存器定义了下降沿上基带波形的特性。

Bit	说明	类型	默认值
15:10	保留	RO	-
9:0	BB TX下降值 在实现基带传输下降沿时提供给 CC TX DAC的代码。	R/W	0h

11.0 供电 MAC

PD MAC 实现了通用串行总线供电规范的协议层和物理层的某些功能。PD MAC 的一端与实现大量协议和更高层的软件接口，另一端与 BMC 编码器/解码器模块接口。

除了正常的 TX 和 RX 功能外，PD MAC 还实现了 USB PD 规范中定义的测试模式逻辑（BIST）。

PD MAC 支持以下功能：

- 用于数据包成帧和 CRC32 插入的自动 TX 模式。
- 用于位级数据包控制的原始 TX 模式。
- 对接收到的报文自动进行 GoodCRC 响应。
- BIST RX 模式下的自动 BIST 错误计数报文。
- GoodCRCTimer 实现。
- 重试次数可编程的自动重试。
- 在自动响应模式下自动丢弃冗余接收数据包。
- 74 字节 TX 队列。
- 128 字节 RX 队列。
- 可编程 TX 位时间。允许更改工作频率。
- 可编程前导码长度。
- BIST TX 和 RX 逻辑。
- 可编程 TX 和 RX 队列模式——缓冲区模式和 FIFO 模式。
- 用于 TX 的 CRC32 发生器。
- 用于 RX 的 CRC32 计算器和比较器。

11.1 PD MAC 发送器

PD MAC 发送器由三个主要模块组成：

- TX 队列
- TX 控制
- TX 通信

在 TX 队列中，软件加载要发送的报文。

TX 控制实现了必要的控制逻辑。它可从 TX 队列中读取数据，并根据数据处理模式（自动或原始）处理数据，使其适用于（包含控制信息的半字节）TX 通信。它还可用于生成数据包成帧，并在自动模式下终止数据包，以及为自动响应生成报文（GoodCRC 和 BIST 错误计数）。TX 控制还处理待发送的 SOP 类型的选择。

TX 通信由 TX CRC 发生器、4b5b 编码器、串行器、前导码生成器和 TX 位定时器组成。它可获取半字节数据、计算并插入 CRC、进行 5b 编码并生成基带串行数据。前导码插入也由该逻辑执行。

以下小节更详细地介绍了各个模块和子模块。还介绍了一些支持的 TX 功能。

11.1.1 TX 队列

在 TX 队列中，软件加载要发送的报文。以下小节更详细地介绍了 TX 队列。

11.1.1.1 TX 队列的工作模式

TX 队列的写入接口（MCU 侧）支持两种工作模式：FIFO 模式和缓冲区模式。

11.1.1.1.1 FIFO 模式

该模式通过将 TX 控制寄存器 A（TX_CTL_A）的 EN_FMQ 位置 1 来使能。在该模式下，软件以类似于 FIFO 的方式将数据写入 TX 队列。软件可使用 1800h-1849h 范围内的任一偏移地址（尽管使用 1800h 更符合逻辑）。FIFO 的深度为 74。写入 FIFO 的数据不能被软件回读。

11.1.1.1.2 缓冲区模式

在**TX控制寄存器A (TX_CTL_A)**中的**EN_FMQ**位清零时（POR后的默认设置）选择该模式。在该模式下，软件将数据写入TX队列，如同写入不同地址处的寄存器。偏移地址范围为**1800h-1849h**，缓冲区有**74**个存储单元。

在此模式下访问TX队列时，只应使用字节访问。请注意，缓冲区偏移地址**1800h**包含最低有效字节（Least-Significant-Byte, LSB）（先出字节）。缓冲区偏移地址**0x049**包含最高有效字节（Most-Significant-Byte, MSB）（后出字节）。

软件可任意写入或读取缓冲区中的任意存储单元。一次只能在TX队列中排队一个数据包。不支持排队多个数据包。

11.1.1.2 TX队列数据和处理模式

TX队列中的数据取决于所选处理模式。支持两种模式：“自动模式数据处理”（Auto Mode Data Processing, AMDP）和“原始模式数据处理”（Raw Mode Data Processing, RMDP）。

11.1.1.2.1 自动模式数据处理（AMD P）

该模式通过将**TX控制寄存器A (TX_CTL_A)**的**EN_RMDP**位清零进行选择。在该模式下，仅在TX队列中排队待发送的报文数据（报头和数据对象）。此外，数据按字节排队。由于不存在数据的成帧信息，因此使用**TX数据包长度寄存器 (TX_PKT_LEN)**提供有关报文数据长度的信息。硬件使用此信息来确定附加CRC的时间和位置。数据包成帧（前导码、SOP和EOP）由硬件自动插入。

11.1.1.2.2 原始模式数据处理（RMD P）

该模式通过将**TX控制寄存器A (TX_CTL_A)**的**EN_RMDP**位置1进行选择。在该模式下，软件用于构建包括成帧在内的整个数据包，前导码和CRC除外。由于成帧涉及K代码，因此队列中的数据用控制信息进行编码。队列每个字节中的数据均被视为“半字节”的数据包数据，其需要由硬件进行5b编码或者需要应当在无5b编码时发送的K代码。此外，还有一些特殊控制字节，用于控制CRC的插入和数据包的终止。

当tx_queue_data[5]为0时，会将tx_queue_data[3:0]视为4b常规数据。此4b数据经过CRC32发生器进行CRC计算，并根据USB PD规范R1.0的表5-1编码为5b（表11-1显示了4b5b编码）。

当tx_queue_data[5]为1时，会将tx_queue_data[4:0]视为5b K代码（见表11-1）。该数据从CRC计算中消除，并且在发送之前不经过任何进一步编码。

tx_queue_data[7:0]值8'hFF和8'hFE有特殊含义。8'hFF意味着数据包数据已完成，硬件现应插入计算得到的CRC32（TX_INS_CRC）。8'hFE意味着应立即停止发送（TX_STOP）。

请注意，在RMDP下，软件可计算自身的CRC，并将其作为编码数据置于队列中。在这种情况下，TX_INS_CRC代码将不会被添加到队列中，软件将继续添加EOP并通过TX_STOP终止传送。

表11-1显示了如何在原始模式下通过发送逻辑对字节宽的排队数据进行解析或编码。

表 11-1: 原始模式数据编码

tx_queue_data[5:0]	编码的符号	tx_queue_data[5:0]	编码的符号
6'h00	5'b11110	6'h08	5'b10010
6'h01	5'b01001	6'h09	5'b10011
6'h02	5'b10100	6'h0A	5'b10110
6'h03	5'b10101	6'h0B	5'b10111
6'h04	5'b01010	6'h0C	5'b11010
6'h05	5'b01011	6'h0D	5'b11011
6'h06	5'b01110	6'h0E	5'b11100
6'h07	5'b01111	6'h0F	5'b11101

tx_queue_data[5:0]	K代码	发送的数据
6'b1_11000	Sync-1	5'b11000
6'b1_10001	Sync-2	5'b10001
6'b1_01101	EOP	5'b01101
6'b1_00111	RST1	5'b00111
6'b1_11001	RST2	5'b11001

tx_queue_data[5:0]	特殊含义
8'hFE	TX_STOP
8'hFF	TX_INS_CRC

请注意，tx_queue_data[4:0]在tx_queue_data[5]设置为1时按原样传递。因此，软件可发送保留符号进行错误测试。惟一需要注意的是，不能使用硬件生成和插入CRC。在这种情况下，软件应计算必要的CRC32并将其添加至数据包，以及跳过队列中用于CRC插入的0xFF代码。

11.1.1.2.3 TX 队列编程序列——AMDP

在AMDP下编程序列时应使用以下序列。该示例假设硬件执行数据包成帧和CRC插入。

1. 确保TX控制寄存器B (TX_CTL_B) 的GO位清零，即，硬件完成之前的TX请求。
2. 如果使用FIFO模式，通过向TX控制寄存器B (TX_CTL_B) 的RST_TXQ_FIFO_WRI_PTR位写入1来清零TX队列WRI指针。
3. 写入两个头字节。
4. 写入有效负载数据（如果有）。
5. 向TX数据包长度寄存器 (TX_PKT_LEN) 写入字节数。
6. 通过TX控制寄存器B (TX_CTL_B) 的OK_TO_TX位检查是否可进行发送。
7. 如果OK_TO_TX: 将TX控制寄存器B (TX_CTL_B) 的GO位置1以开始发送。
否则: 重复步骤6。注: 如果某个RX正在进行，则可能需要放弃当前的TX。

11.1.1.2.4 TX 队列编程序列——RMDP

在RMDP下编程序列时应使用以下序列。该示例假设硬件执行CRC插入。

1. 确保TX控制寄存器B (TX_CTL_B) 的GO位清零，即，硬件完成之前的TX请求。
2. 如果使用FIFO模式，通过向TX控制寄存器B (TX_CTL_B) 的RST_TXQ_FIFO_WRI_PTR位写入1来清零TX队列WRI指针。
3. 将SOP K代码写入FIFO *Sync-1、Sync-1、Sync-1和Sync-2。
4. 写入两个头字节，每个字节分为两个半字节。
5. 写入有效负载数据（如果有），每个字节分为两个半字节。
6. 写入值“0xFF”以插入CRC。
7. 写入EOP K代码（EOP）。
8. 写入值“0xFE”以终止发送。如果未写入该值，则将导致发送在TX缓冲区中持续无限循环。
9. 通过TX控制寄存器B (TX_CTL_B) 的OK_TO_TX位检查是否可进行发送。
10. 如果OK_TO_TX：将TX控制寄存器B (TX_CTL_B) 的GO位置1以开始发送。
否则：重复步骤9。注：如果某个RX正在进行，则可能需要放弃当前的TX。

注： 在该模式下，将忽略TX数据包长度寄存器 (TX_PKT_LEN) 的值，因为硬件知道何时根据TX_STOP控制代码的检测来终止数据包。

11.1.2 TX控制

TX控制实现了必要的控制逻辑。它可从TX队列中读取数据，并根据数据处理模式（自动或原始）处理数据，使其适用于（包含控制信息的半字节）TX通信。它还可用于生成数据包成帧，并在自动模式下终止数据包，以及为自动响应生成报文（GoodCRC和BIST错误计数）。TX控制还处理待发送的SOP类型的选择。

11.1.2.1 发送请求

有四个发送请求源：

- 来自接收器的用于软复位的GoodCRC Ack
- 来自接收器的用于正常数据包的GoodCRC Ack
- 来自BIST接收器的BIST错误计数报文
- 从软件发送

11.1.2.2 发送中止

启动数据包发送后，器件将对模拟组件上电，然后等待总线周转定时器和上电延时定时器超时。在此等待期间内，发送可由软件中止，并在接收到良好数据包（包括软复位）、硬复位或者线缆复位（如果已使能并且待处理TX的SOP类型为SOP'、SOP"、SOP'_Debug或SOP"_Debug）时，将由器件中止。

总线周转时间超时（或者如果其已超时）并且上电时间超时后，检查总线是否空闲。如果总线空闲，则开始发送。如果总线不空闲，则丢弃发送，并且通常中止状态位会置1。此丢弃可针对所有数据包通过TX控制寄存器A (TX_CTL_A) 的WAIT4LINE_IDLE位来禁止；也可针对用于已接收软复位的自动响应触发的GoodCRC Ack来禁止（除非通过DIS_SPCL_SR_GCRC_ACK位禁止该功能）。

如果因总线处于非空闲状态而未丢弃发送，则器件将等待总线变为空闲，然后再次等待周转时间。在等待总线空闲期间，发送可由软件中止，并在接收到良好数据包、硬复位或者线缆复位（如果已使能并且待处理TX的SOP类型为SOP'、SOP"、SOP'_Debug或SOP"_Debug）时，将由硬件中止。

软件发出的硬复位和线缆复位不会因接收到良好数据包、硬复位或线缆复位而中止。

请注意，对于接收到的良好数据包，要中止待处理发送，已接收数据包的SOP类型必须与待处理发送的SOP类型匹配，或者待处理发送的SOP类型为非SOP时，已接收数据包的SOP类型必须为SOP（后者可通过TX控制寄存器A（TX_CTL_A）的DIS_SOP_ABRTS_NON_SOP位禁止）；而对于软复位以外的报文，接收到的报文ID必须指示非重复的数据包（始终不将软复位视为重复）。Ping和GoodCRC报文不会导致发送中止。

提供单独的TX中断中止位和单独的中止状态寄存器以用于软件发出的数据包以及自动响应数据包。

11.1.2.3 发送重试

在以下两种情况下重试发送的数据包：总线空闲违例和GoodCRC响应超时。

如果因总线处于非空闲状态而丢弃发送，则存在重试的选项，而不是将其视为中止。为使这种情况发生，数据包必须已由软件启动（而不是硬复位或线缆复位），自动响应模式必须使能，重试计数必须为非零值，并且TX控制寄存器A（TX_CTL_A）的RETRY_ON_LINE_BUSY位必须置1。如果经过指定的重试次数后，因总线繁忙而导致发送失败，TX_FAILED状态位置1（而非TX_ABORTED）。不重试硬复位和线缆复位。

在发送软件启动的帧后（硬复位和线缆复位除外），器件将启动定时器并等待接收器指示GoodCRC响应。前提是假定重试和/或等待GoodCRC已使能。

如果接收到具有正确SOP类型和报文ID的GoodCRC，则完成等待并且发送完成。如果接收到具有错误SOP类型或报文ID的GoodCRC，则会被发送器忽略（甚至不会通知发送器），并被接收器默默丢弃。

如果等待CRC定时器超时，并且剩余重试次数不为零，则重新发送原始数据包。如果剩余重试次数为零，则不会重试数据包，并会指示失败状态。

如果接收到以下任何一项，则等待GoodCRC将被中止：

- 硬复位
- 线缆复位（如果线缆复位接收已使能，且待处理TX的SOP类型为SOP'、SOP"、SOP'_Debug或SOP"_Debug）
- 软复位（如果RX的SOP类型与待处理TX的SOP类型相同，或者RX的SOP类型为SOP而待处理发送的SOP类型为非SOP）
- 除GoodCRC或Ping外的良好数据包（如果RX的SOP类型与待处理TX的类型相同，或者RX的SOP类型为SOP而待处理发送的SOP类型为非SOP（后者可通过TX控制寄存器A（TX_CTL_A）的DIS_SOP_ABRTS_NON_SOP位禁止）且数据包不是重复报文）

将后两个原因视为协议错误。

等待GoodCRC还可通过软件中止。已中止的GoodCRC等待不会重试。

11.1.2.4 发送器禁止

为避免出现软件当前正在发出发送命令而硬件正在接收数据包的冲突情况，TX参数寄存器A（TX_PARAM_A）的EN_FWTX位在接收到硬复位到线缆复位（如果已使能——未完成SOP类型检查，因为发送可能未处于待处理状态）（基于RX中断状态寄存器（RX_IRQ_STAT）的RX_CABLE_RST和RX_HARD_RST位）或者RX FIFO中存在数据时，会自动清零。使用中断和FIFO状态（电平敏感）而非事件发生（边沿），可避免软件在事件后将EN_FWTX位置1的冲突。

11.1.3 TX通信

TX通信用于从TX控制中获取编码的半字节，必要时对其进行编码，然后进行序列化以使其准备好发送。其可用于前导码插入、CRC计算和CRC插入。它还可作为BMC编码器提供TX时钟信号。

11.1.3.1 前导码插入

器件自动向发送的数据包添加交替的0和1前导码。完成前导码的发送时，将在TX控制的指引下处理来自TX队列的数据。

UPD360

发送的前导码位数通过 **TX 参数寄存器 B (TX_PARAM_B)** 的 **PREAMBLE_LEN** 位值进行编程。

注： 将 **PREAMBLE_LEN** 字段设置为奇数值时，将导致违反 USB PD 规范，其中规定“前导码应以 0 开始，并以 1 结束”。奇数值将导致前导码以 0 值开始，但也将以 0 值结束。将此字段设置为奇数值可能会导致也可能不会导致功能问题。因此，为保持符合 USB PD 规范，应将该字段设置为偶数值。

11.1.3.2 TX CRC32

在发送开始（将 **TX 控制寄存器 B (TX_CTL_B)** 的 **GO** 位置 1 或通过硬件触发自动响应）时，发送 CRC 将复位。

编码的半字节数据将传送至 TX CRC32，一次传送一个“半字节”，以用于 CRC 计算。在 CRC 计算中，仅使用有效 D 代码数据。在 CRC 计算中排除所有成帧数据（无效 D 代码）。

注： USB PD 规范中介绍了 CRC32 算法。

11.1.3.3 TX 位时序

基带发送信号的位时间由 **TX 位时间计数器寄存器 (TX_BITTIME_CNT)** 的值控制。该值决定了发送数据速率。

计数值基于时钟频率，可通过以下公式得出：

$((\text{时钟频率 kHz} / \text{比特率 kbps}) - 1)$

例如：

时钟频率 = 48000 kHz (48 MHz)

标称比特率 = 300 kbps

$\text{bit_time_cnt} = (48000 / 300) - 1 = 159$

11.1.4 自动响应模式

器件支持自动响应模式（不要与自动数据处理模式混淆）。在该模式下，器件将在成功接收数据包后自动发送 GoodCRC 报文，或在接收到 BIST PRBS 帧后自动发送 BIST 错误计数报文。该模式可通过将 **TX 控制寄存器 A (TX_CTL_A)** 的 **EN_AUTO_RSP_MODE** 位置 1 来使能。

在正常的自动响应期间，如果发现总线在器件尝试发送自动响应时处于繁忙状态（除非有噪声，否则不应发生这种情况），则会中止发送，**TX 中断状态寄存器 (TX_IRQ_STAT)** 的 **AUTO_RSP_ABORTED** 位将置 1。可通过将 **TX 控制寄存器 A (TX_CTL_A)** 的 **WAIT4LINE_IDLE** 位置 1 改变此行为。将该位置 1 将强制器件等待总线变为空闲状态（更多信息，请参见 **WAIT4LINE_IDLE** 位说明）。即使 TX_CTL_A 的 **RETRY_ON_LINE_BUSY** 置 1，也不会重试 GoodCRC 报文和 BIST 错误计数报文。

在 USB 供电修订版 1.0 中，软复位具有一条致命规则，软复位失败会导致硬复位，进而导致链路断开，因此，对于为响应接收到软复位而进行的 GoodCRC ACK 的发送，其处理方式稍有不同。如果硬件在尝试发送 GoodCRC ACK 时发现线路繁忙，则会等待至线路空闲，然后重新尝试发送。该过程将无限次发生。当 **WAIT4LINE_IDLE** 置 1 时，此行为与正常 GoodCRC ACK 时的行为相同。即使 **TX 控制寄存器 A (TX_CTL_A)** 的 **RETRY_ON_LINE_BUSY** 位置 1，也不会重试 GoodCRC。对于 USB 供电修订版 3.0，通常会重试软复位，因此应通过 **DIS_SPCL_SR_GCRC_ACK** 位禁止该功能。

用于自动发送的 GoodCRC 数据包的 SOP 类型为已接收数据包的 SOP 类型（由接收器解析）。用于 BIST 错误计数报文的 SOP 类型通过 **TX 参数寄存器 A (TX_PARAM_A)** 的 **TX_SOP_SELECT** 位进行选择。

对于 GoodCRC 和 BIST 错误计数报文，数据包头中的 bit 8 取自 **TX 参数寄存器 C (TX_PARAM_C)** 的 **PORT_POWER_ROLE**（对于 SOP）或者 **CABLE_PLUG**（对于 SOP'、SOP''、SOP'_debugs 和 SOP''_debugs）位，具体取决于接收到的 SOP 类型。数据包头中的 bit 5 取自 **PORT_DATA_ROLE** 位（对于 SOP）或设置为 0（对于 SOP'、SOP''、SOP'_debugs 和 SOP''_debugs），具体取决于接收到的 SOP 类型。

为避免丢失自动响应 GoodCRC 请求，使用了一个标志。该标志在出现来自接收器的请求脉冲后置 1，在发送或中止响应时清零。请求标志置 1 后，将保存请求的 SOP 类型和报文 ID 以及软复位响应指示。如果已存在待处理的自动响应 GoodCRC 请求，并接收到会导致其他自动响应 GoodCRC 请求的另一个数据包，除非第一个请求的 SOP 类型为 SOP，

而第二个请求的SOP类型为SOP' _Debug或SOP" _Debug，否则第二个请求将替换第一个请求。无论TX控制寄存器A (TX_CTL_A) 的DIS_SOP_ABRTS_NON_SOP位是否置1，都会进行此替换。即使接收到的数据包未中止待处理的自动响应，待处理的自动响应也可能被新的自动响应替换。

11.1.5 自动重试模式

该器件还支持自动重试模式。在该模式下，如果在适当的时间内（即，在CRCReceiveTimer超时之前）未接收到GoodCRC报文，则重试当前报文。

硬件将重试该报文，直至达到在TX参数寄存器C (TX_PARAM_C) 的N_RETRY_CNT字段中指定的重试值。如果N_RETRY_CNT设置为零，将禁止自动重试模式。

对器件完成事务所使用的重试次数进行跟踪，并将其提供给软件（通过TX状态寄存器 (TX_STAT) 的N_HW_RETRIES字段）。

N_RETRY_CNT值为0意味着仅尝试一次报文，即，硬件不会重试。N_RETRY_CNT指定重试次数，因此尝试次数为(N_RETRY_CNT + 1)。换言之，如果要让硬件尝试发送“N”次报文，则N_RETRY_CNT字段必须设置为“N-1”。

从端口伙伴接收到硬复位或线缆复位（如果已使能，且待处理TX的SOP类型为SOP'、SOP"、SOP' _Debug或SOP" _Debug）、除GoodCRC或Ping报文外的报文（如果RX和TX的SOP类型匹配，或者，如果RX的SOP类型为SOP而TX的SOP类型为非SOP（后者可通过TX控制寄存器A (TX_CTL_A) 的DIS_SOP_ABRTS_NON_SOP位进行禁止））或由软件发出中止命令时，将中止所有待处理的自动重试。

检测到总线冲突会导致器件中止当前传输，其中包括所有待处理的自动重试，除非TX控制寄存器A (TX_CTL_A) 的RETRY_ON_LINE_BUSY位置1。

请注意，通过将N_RETRY_CNT设为0禁止自动重试也会禁止器件在指示成功报文发送之前等待GoodCRC报文的能力。例如，这适用于发送软复位的情况。为规避此限制，可使用TX参数寄存器A (TX_PARAM_A) 的EXPECT_GOODCRC位。将该位置1会使器件等待GoodCRC以响应TX报文，即使将N_RETRY_CNT字段设为零时亦如此。

注： 在GO位置1后将N_RETRY_CNT设为0对当前传输无影响。

GoodCRC报文中的预期报文ID由软件通过TX参数寄存器A (TX_PARAM_A) 的MSG_ID字段进行设置。将其与接收到的GoodCRC报文中的报文ID进行比较。

GoodCRC报文中的预期SOP类型由软件通过TX参数寄存器A (TX_PARAM_A) 的TX_SOP_SELECT字段进行设置。将其与GoodCRC报文中接收到的SOP类型进行比较。为接收到GoodCRC报文，假设SOP类型已通过RX控制寄存器B (RX_CTL_B) 的RX_SOP_ENABLE字段进行使能。

11.1.6 IFG 定时器

TX周转定时器用于确保数据包接收和数据包发送之间的总线空闲时间最短，即，确保在成功接收到数据包时发送GoodCRC报文。

该定时器的值可通过TX周转时间寄存器 (TX_TA_TIME) 设定。该值的单位为μs。该定时器使用自由运行的1 μs脉冲，因此该寄存器的值应比所需值多1，以确保最短时间。

11.1.7 CRC接收定时器

如果在请求发送时器件预计等待GoodCRC ACK（即，自动重试模式由TX参数寄存器C (TX_PARAM_C) 的N_RETRY_CNT字段的非零值使能，或者，该字段值为0时TX参数寄存器A (TX_PARAM_A) 的EXPECT_GOODCRC位将置1），则会使能该定时器。该定时器还可通过RX控制寄存器A (RX_CTL_A) 的EN_CRC_RCV_TMR位进行使用以供软件使用。

如果使能，该定时器将在TX数据包发送停止时启动。

如果软件在使能自动重试模式的情况下中止发送之间的TX数据包，则该定时器不会被触发并将被禁止。如果软件在禁止自动重试的情况下中止TX数据包，则还应通过将RX控制寄存器A (RX_CTL_A) 的EN_CRC_RCV_TMR位清零禁止定时器。

CRCReceiveTimer的超时值可通过RX tReceive时间寄存器（RX_TRECEIVE_TIME）的TRECEIVE字段编程，以10 μs的倍数计。

在BIST TX模式下软件将该定时器用作BISTReceiveErrorTimer（器件不对BISTReceiveError自动计时）；在不使用自动重试或等待GoodCRC模式时，将该定时器用作CRCReceiveTimer。

11.1.8 正在中止TX

软件可通过TX控制寄存器B（TX_CTL_B）的ABORT位发出中止命令来中止已启动的发送。数据包将根据当前发送阶段被中止，具体如下：

- 如果因器件正在等待周转定时器超时或线路变为空闲而尚未启动数据包发送，则发送将立即中止。
- 如果当前处于前导码阶段，则器件将完成当前位的发送，追加EOP，并关闭发送器。
- 如果当前处于SOP、数据或CRC阶段，则器件将完成当前半字节的发送，追加EOP，并关闭发送器。

硬复位应遵循此操作，此为软件的工作范畴。

11.2 PD MAC接收器

PD MAC接收器由三个主要模块组成：

- RX 队列
- RX 控制
- RX 通信

软件从RX队列中读取接收到的报文。

RX控制实现了必要的控制逻辑。它负责验证接收到的数据包、更新RX队列状态以及在需要时触发自动响应。

RX通信由时钟和数据恢复（Clock and Data Recovery, CDR）、RX DES（解串器）（串并转换器、4B5B解码器和成帧检测器）、RX CRC32（CRC计算器和接收定时器）以及用于检测有效数据包接收的其他逻辑组成。

以下小节更详细地介绍了各个模块和子模块。

11.2.1 RX队列

解码的RX数据（报头、数据对象和CRC）保存在一个集成的128字节RX FIFO中。RX FIFO能够存储多个数据包并提供读写指针。

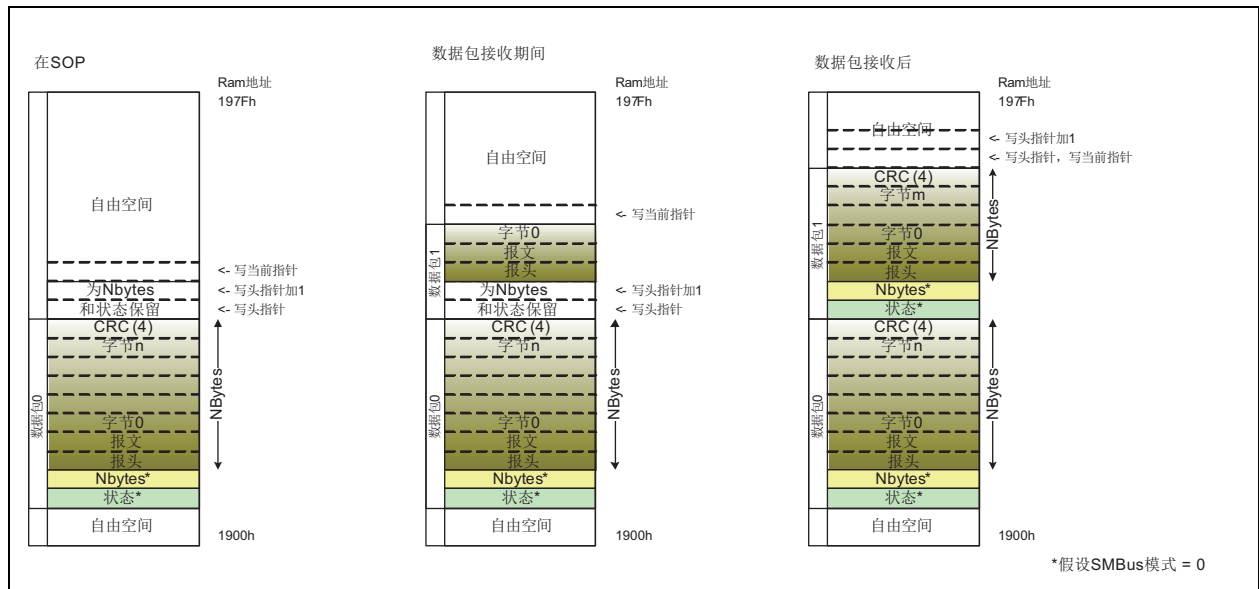
向每个数据包的开头添加两个字节。字节0用于保存数据包状态（SOP类型和传统“缓冲区”有效位），字节1用于保存数据包长度。存在一个选项用于交换这些值（在字节0中保存长度，在字节1中保存状态）以及将数据包长度加1（以将状态字节考虑在内）。此选项用于SMBus（如块读取），其中读取的第一个字节指示传输的长度，其余字节（状态和数据包）紧随其后。请注意，数据包包含4字节CRC，其包含在长度中。

以下小节详细介绍了FIFO的写入和读取接口。

11.2.1.1 RX FIFO 写入接口

图 11-1 显示了在各个接收点的写入接口（从接收硬件的角度）。

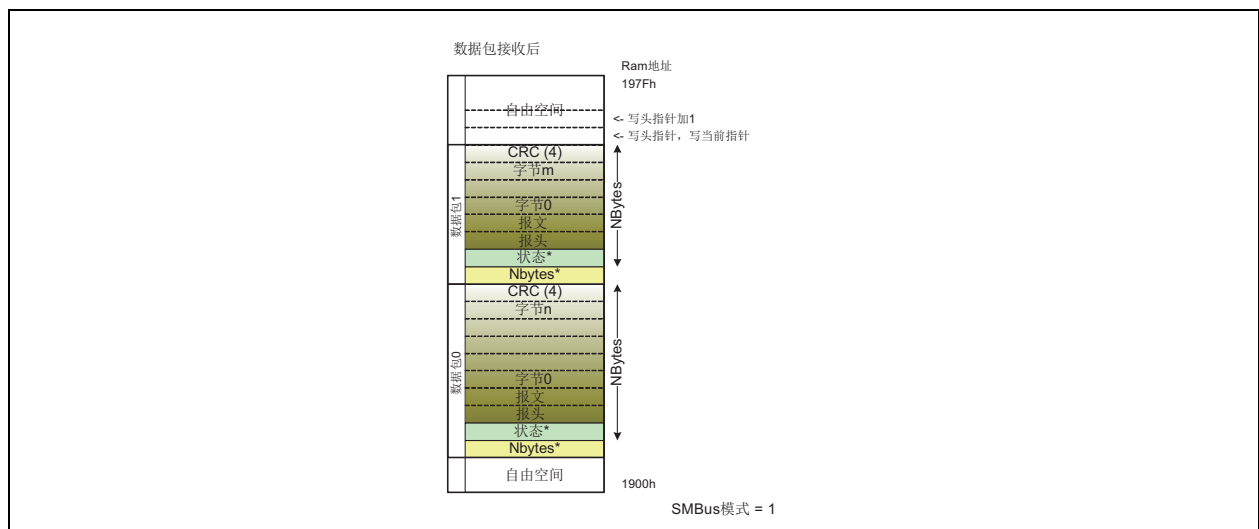
图 11-1: RX FIFO 的写入接口视图



每个RX数据包由两个字节的控制信息和所接收数据包数据组成。状态字节指示数据是否有效以及接收到的SOP类型。状态字节中的值0x00表示数据无效（包括Nbytes字段）。当状态字节的bit 0为1'b1时，Nbytes字段指示数据包的长度，状态字节的bit 6:4指示接收到的SOP类型。空FIFO始终返回值0x00。但是，不应轮询状态字节，而应使用RX中断状态寄存器（RX_IRQ_STAT）的RX_FIFO_NOT_EMPTY位。为传统兼容性保留有效位。当软件从FIFO中读取数据包数据时，应仅读取由Nbytes指示的字节数（除了Nbytes和状态条目）。请注意，Nbytes指示的字节数包含CRC32数据。

SMBus模式使能（通过RX控制寄存器A（RX_CTL_A）的EN_SMBUS_MODE位）时，交换状态和Nbytes字段，Nbytes增加1以包含状态字节。对于该模式，交换将导致无法轮询状态字节（不推荐）。当软件从FIFO中读取数据包数据时，应仅读取由Nbytes指示的字节数（不包括Nbytes条目）。图 11-2 详细给出了SMBus模式下的格式。

图 11-2: RX FIFO SMBUS 模式



UPD360

表 11-2 和表 11-3 分别给出了数据包状态和NBytes 格式。

表 11-2: RXQ 状态字段

Bit	说明	类型
7	保留 始终读为0。	RO
6:4	RX_SOP_TYPE 该字段指示接收到的 SOP 类型。 000b: SOP 001b: SOP' 010b: SOP" 011b: SOP'_Debug 100b: SOP"_Debug 101b - 111b: 保留	RO
3:1	保留	RO
0	状态 队列数据的状态 0b: 数据无效 1b: 数据有效	RO

表 11-3: RXQ 字节数 (NBYTES) 字段

Bit	说明	类型
0	NBYTES 如果 EN_SMBUS_MODE 为0, 则该字段表示数据包中有效数据的字节数。 如果 EN_SMBUS_MODE 为1, 则该字段表示数据包中有效数据的字节数加1。	RO

当器件检测到有效数据包起始 (Start-Of-Packet, SOP) 信号时, 当前写指针将设置为写头指针加2。接收到数据包数据时, 会将其写入FIFO中的当前写指针处, 并在每次写入后, 当前写指针都会递增。当前写指针指向下一个要写入的存储单元。达到RAM顶部时, 当前写指针绕回至0。字节计数由RX控制模块保存。

当接收到有效EOP且CRC逻辑指示良好CRC余数时, 硬件会将状态 (有效位和接收到的SOP类型) 写入写头指针寻址的存储单元, 并将接收到的字节数写入写头指针加1寻址的存储单元。SMBus模式将交换数据 (并使字节数递增)。在写入状态和长度后, 写头指针更新为当前写指针, 写头指针加1更新为当前写指针加1 (考虑到在达到RAM顶部时绕回)。

如果未接收到有效EOP且总线空闲, 则检测到异常终止。在这种情况下, 将丢弃数据包, 即, 器件不会更新NBytes存储单元或状态存储单元。

如果检测到有效EOP, 但CRC发生器指示CRC无效, 则数据包损坏。器件将丢弃数据包, 不会更新NBytes存储单元或状态存储单元。

如果接收到的数据超过FIFO的大小, 则检测到溢出。超出的数据不会写入FIFO中, 当前写指针不会递增。

在数据包起始处进行单独的测试, 以确定是否有足够的空间容纳状态和长度字节。如果“没有用于报头的空间”, 则检测到溢出。超出的数据不会写入FIFO中, 当前写指针不会初始化。

如果数据包已达到允许的最大大小 (由RX最大数据包大小寄存器 (RX_MAX_SIZE) 指定) 且接收到更多数据, 则会检测到数据包过大。超出的数据不会写入FIFO中, 当前写指针不会递增。

所有指针均由一个RAM地址和一个绕回指示组成。每当指针在RAM的顶部绕回时，都会翻转绕回指示。绕回指示用于已满、“没有用于报头的空间”和空比较中，以区分在读指针之前RAM大小已满的写指针（已满条件，在该条件下绕回翻转将相反）和等于读指针的写指针（空条件，在该情况下绕回翻转将相等）。

所有指针和状态位均可由软件读取。所有指针和绕回状态位均可由软件写入。软件负责保持指针和绕回翻转值之间的一致性。

FIFO指针可通过将RX控制寄存器A (RX_CTL_A) 的RST_RECEIVER位置1和清零来复位。

11.2.1.1.1 重复数据包处理

将器件设置为自动响应（发送GoodCRC报文）时，可能会因GoodCRC响应丢失而多次接收某个数据包。如果成功接收到原始数据包，则将自动丢弃具有匹配报文ID的相应SOP类型的后续报文。

器件的操作取决于EOP处CRC生成器的CRC状态。如果CRC良好，并且当前报文的报文ID匹配前一报文的报文ID（根据SOP类型存储的MSG_ID），则器件将不更新数据包的状态和长度存储单元，也不会更新写头指针和写头指针加1。将在下一个数据包的起始处回收FIFO空间。RX错误中断状态寄存器 (RX_ERR_IRQ_STAT) 的RX_PKT_DROPPED位将置1以指示此条件。随后器件将重新发送GoodCRC响应。但是，如果CRC发生器指示不良CRC，或者未接收到EOP，则器件将按照损坏数据包的方式处理数据包。存储的MSG_ID将不会更新。

请注意，如果使能了自动重试、发送器正在等待GoodCRC，以及接收到除GoodCRC外的报文（协议错误），则会丢弃重复数据包。

器件将保持丢弃的重复数据包 (RX重复数据包计数寄存器 (RX_DUP_PKT_CNT)) 和接收到的损坏数据包 (RX BadCRC数据包计数寄存器 (RX_BADCRC_PKT_CNT)) 的单独计数。如果其中的任一计数超过127，RX错误中断状态寄存器 (RX_ERR_IRQ_STAT) 的DBG_EVENT位将置1。

11.2.1.1.2 GoodCRC数据包存储和丢弃

在使能自动重试且发送器正在等待GoodCRC的情况下，将丢弃GoodCRC响应。器件不会更新数据包的状态和长度存储单元，也不会更新写头指针和写头指针加1。将在下一个数据包的起始处回收FIFO空间。请注意，即使SOP类型或报文ID与预期值不匹配，也会丢弃GoodCRC响应。

表 11-4 汇总了存储或丢弃良好数据包的条件和结果。

表 11-4: 丢弃良好数据包的条件和结果

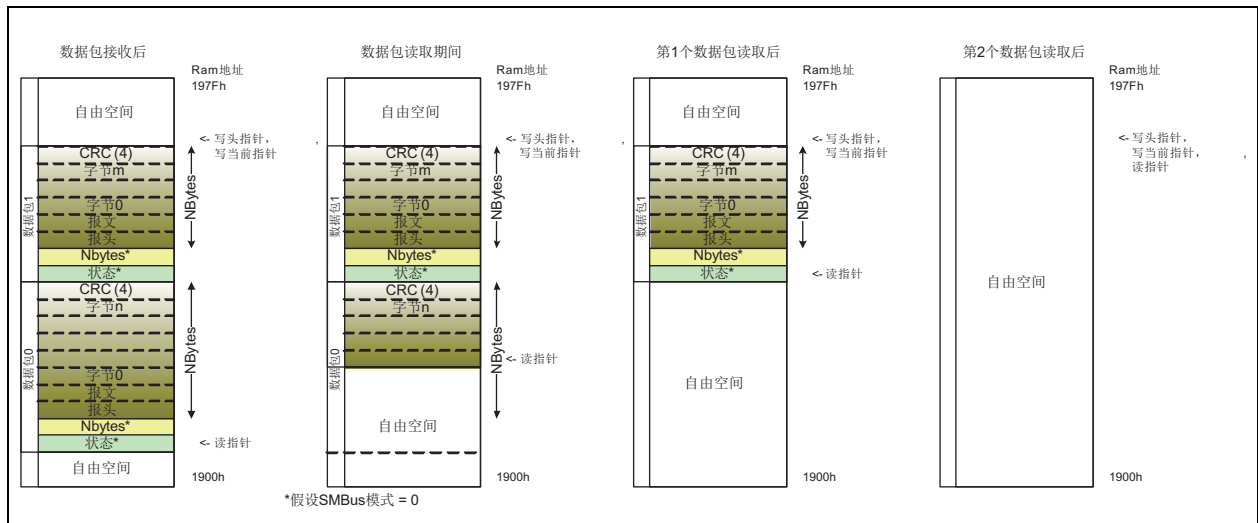
事件		结果	
如果 预期 GoodCRC	接收到 RX 的 SOP 与 TX 的 SOP 匹配的软 复位	数据包已存储 但存在错误状态	RX_PKT_STAT[2] (PCOL_ERROR)
	接收到具有正确 SOP 类型和报文 ID 的 GoodCRC	数据包被丢弃	
	接收到具有错误 SOP 类型或错误报文 ID 的 GoodCRC		
	接收到除 GoodCRC 或 Ping 以外的报文 (RX 的 SOP 匹配 TX 的 SOP, 该 SOP 类型 的第一个报文或报文 ID 与该 SOP 类型的最 后一个报文 ID 不匹配 (即, 不重复))	数据包已存储, 但存在错误状态	RX_PKT_STAT[2] (PCOL_ERROR)
否则, 如果 自动响应已使能	接收到软复位	数据包已存储, 报文 ID 已保存	
	接收到除 GoodCRC 或软复位外的报文以 及此 SOP 类型的第一个报文		
	接收到除 GoodCRC 或软复位外的报文, 报文 ID 与此 SOP 类型的最后一个报文 ID 不 匹配		
	接收到除 GoodCRC 或软复位外的报文, 报 文 ID 与此 SOP 类型的最后一个报文 ID 匹配	数据包被丢弃 且具有重复状态	RX_PKT_STAT[1] (DUPLICATE_PACKET)
	接收到 GoodCRC	数据包已存储	
否则	接收到除 GoodCRC 外的报文 (包括软复位)	数据包已存储, 报文 ID 已保存	
	接收到 GoodCRC	数据包已存储	

注: 如果使能自动响应, 则只要存储除 GoodCRC 外的数据包, 就会发送 GoodCRC 响应。

11.2.1.2 RX FIFO 读取接口

图 11-3 显示了在各个软件读取点的读取接口。

图 11-3: RX FIFO 的读取接口视图



RX FIFO 读取接口始终使用FIFO访问模式。读取具有破坏性，会返回当前数据并释放FIFO空间。尽管软件可出于测试目的而回退读指针，但也不能保证数据未被改写。

通过读取RX中断状态寄存器 (RX_IRQ_STAT) 的RX_FIFO_NOT_EMPTY位确定FIFO队列中数据的有效性后，软件可从RX FIFO读取数据。软件可读取整个数据包，而无需重新检查RX_FIFO_NOT_EMPTY位。

尽管接收到每个字节后都会写入数据，但只有在接收并验证整个数据包后才会指示FIFO中的有效数据。

根据RX控制寄存器A (RX_CTL_A) 的EN_SMBUS_MODE位的设置，可以先读取状态字段或Nbytes字段，然后读取另一个字段。如果使能SMBus模式，则Nbytes字段将先读取并将包含状态字节，否则将后读取Nbytes字段并仅指示数据包长度。请注意，数据包包含4字节的CRC，其包含在长度中。

访问FIFO时，将从FIFO中的读指针处读取数据包数据，并且每次读取后读指针都会递增。读指针指向下一个要读取的存储单元。达到RAM顶部时，读指针绕回至0。

读取数据包时，FIFO空间被释放。软件无需专门释放缓冲区空间。

如果软件需要在不读取数据的情况下清除数据包，则可通过写入RX FIFO读指针寄存器 (RX_FIFO_RD_PTR) 和RX FIFO读指针控制位寄存器 (RX_FIFO_RD_PTR_CTL_BITS) 来实现。考虑到已读取的数据量，软件应为读指针增加适当的值。对于读指针和RX_FIFO_RD_PTR_WRAP控制位，必须考虑到在达到RAM顶部时绕回的情况。为保持一致性，器件在临时寄存器中保存对RX FIFO读指针寄存器 (RX_FIFO_RD_PTR) 的写操作，并在写入RX FIFO读指针控制位寄存器 (RX_FIFO_RD_PTR_CTL_BITS) 时将其传送至实际寄存器。

尽管可作为FIFO访问，但软件也可使用128字节FIFO地址范围内的任何偏移地址。

虽然不推荐这样做，但如果FIFO为空，软件也可从中读取数据。返回的数据将为0，并且FIFO不会下溢。

所有指针均由一个RAM地址和一个绕回指示组成。每当指针在RAM的顶部绕回时，都会翻转绕回指示。绕回指示用于已满、“没有用于报头的空间”和空比较中，以区分在读指针之前RAM大小已满的写指针（已满条件，在该条件下绕回翻转将相反）和等于读指针的写指针（空条件，在该情况下绕回翻转将相等）。

所有指针和状态位均可由软件读取。所有指针和绕回状态位均可由软件写入。软件负责保持指针和绕回翻转值之间的一致性。

FIFO 指针可通过将 **RX 控制寄存器 A (RX_CTL_A)** 的 **RST_RECEIVER** 位置 1 和清零来复位。

RX FIFO 中的每个条目均表示一个字节的已接收数据。写入 FIFO 的数据为报头、有效负载和 CRC32。如前文所述，只有对应于有效 D 代码的 5b 符号才会被处理并写入 RX 队列。K 代码不能传送到 RX 队列，即，不存在“原始模式”接收功能。软件仅查看去帧数据包。

如前文所述，FIFO 中的数据是实际数据而非符号，因此可直接由软件处理。损坏的数据包不会写入队列。

11.2.2 RX 控制

RX 控制实现了验证接收到的数据包、更新 RX 队列状态以及在需要时触发自动响应所需的逻辑。

11.2.2.1 重复数据包检测

为进行重复数据包检测，器件保持根据 SOP 类型接收到的最后一个报文 ID。标志 (**first_msg**) 基于 SOP 类型保持，并指示存储的最后一个报文 ID 是否有效 (已存在前一个数据包)。当接收到良好数据包时，使用此信息来检测报文 ID 是否与针对该 SOP 类型接收到的最后一个报文 ID 相匹配。

器件将当前报文 ID 存储在相应的最后一个报文 ID 中 (由当前已接收数据包的 SOP 选择)。指定 SOP 类型的报文 ID 和标志可通过 **RX 报文 ID 存储寄存器 (RX_MSG_ID_STORED)** 复位。软件通过 **复位控制寄存器 (RESET_CTL)** 发出 **SW_RESET** 或 **PD_RESET** 命令时，所有位均清零。

11.2.2.2 发送中止的 RX 控制

虽然待处理发送的实际中止是由发送控制模块完成的，但由 RX 控制模块作出决定。

对于待处理发送 (发送被触发，但等待总线空闲或总线周转时间)，RX 控制模块在 EOP 处生成一个脉冲。它将检查：
1) 数据包为良好数据包 (良好 CRC、正确的半字节对齐以及无符号错误)，2) 数据包不为 Ping 或 GoodCRC 报文，3a) 接收到的数据包的 SOP 类型匹配待处理发送的 SOP 类型，或 3b) 接收到的数据包的 SOP 类型为 SOP，而待处理发送的 SOP 类型为非 SOP (后者可通过 **TX 控制寄存器 A (TX_CTL_A)** 的 **DIS_SOP_ABRTS_NON_SOP** 位禁止)，以及 4) 数据包不是重复数据包或者是软复位报文。

对于正在等待返回 GoodCRC 的已完成发送，RX 控制将检查：1) 数据包为良好数据包 (良好 CRC、正确的半字节对齐以及无符号错误)，2) 数据包为软复位报文或不为 GoodCRC 报文，3a) 接收到的数据包的 SOP 类型匹配待处理发送的 SOP 类型，或 3b) 接收到的数据包的 SOP 类型为 SOP，而待处理发送的 SOP 类型为非 SOP (后者可通过 **TX 控制寄存器 A (TX_CTL_A)** 的 **DIS_SOP_ABRTS_NON_SOP** 位禁止)，以及 4) 数据包不是重复数据包。

11.2.3 RX 通信

RX 通信由时钟和数据恢复 (CDR)、RX DES (解串器) (串并转换器、4B5B 解码器和成帧检测器)、RX CRC32 (CRC 计算器和接收定时器) 以及用于检测有效数据包接收的其他逻辑组成。

11.2.3.1 CDR

BMC 对接收到的数据进行解码并输出比特流信号。将该信号施加到 CDR 模块的输入端，该 CDR 模块在前导码阶段锁定 (周期和相位) 到数据流，并输出串行数据和捕捉选通 (以数据窗口为中心)。

在前导码阶段，CDR 依赖边沿到边沿时间测量 (两个位时间) 实现周期锁定。使用四个连续良好周期测量值的平均值进行周期锁定。有效周期测量由 **RX 最大比特率位周期计数寄存器 (RX_BIT_PER_CNT_MAX_BR)** 和 **RX 最小比特率位周期计数寄存器 (RX_BIT_PER_CNT_MIN_BR)** 指定的范围决定。相位锁定通过比较同相和正交数据样本以确定相位是需要超前还是滞后来实现。

11.2.3.2 RX DES（解串器）

RX DES将接收到的来自CDR的串行比特流转换为5位码，识别并生成数据包起始/数据包结束/硬复位和线缆复位信号，并将5位码解码为4b数据。

注： 请注意，只有有效的D代码才被处理以写入RX队列。无效/保留代码将被丢弃。

11.2.3.2.1 SOP 检测

通过将最近接收到的4个5位码与预期SOP模式进行比较来检测SOP。根据USB PD规范，检测4个SOP码中的3个已足够。此操作通过创建四个独特的组来完成，每个组包含三个接收码（分别为{a,b,c}、{a,b,d}、{a,c,d}和{b,c,d}），每个组相对于其各自的码进行单独比较。任一成功匹配均表示SOP。

针对每个SOP类型进行单独比较，可通过RX控制寄存器B（RX_CTL_B）的RX_SOP_ENABLE字段使能。对结果进行逻辑或运算。SOP类型被保存并用于数据包处理（自动回复、自动响应和队列状态）。允许软件随时更改RX_SOP_ENABLE字段。内部值在有效接收期间被保留，并在空闲期间从寄存器更新。

为在RX SOP中断状态寄存器（RX_SOP_IRQ_STAT）中生成RX_SOP中断，每个SOP检测均被解码，但不与其RX_SOP_ENABLE进行逻辑与运算。

11.2.3.2.2 复位检测

通过将最近接收到的4个5位码与预期模式进行比较来检测硬复位和线缆复位。根据USB PD规范，检测4个码中的3个已足够。此操作通过创建四个独特的组来完成，每个组包含三个接收码（分别为{a,b,c}、{a,b,d}、{a,c,d}和{b,c,d}），每个组相对于其各自的码进行单独比较。任一成功匹配均分别表示硬复位或线缆复位。

线缆复位必须通过RX控制寄存器A（RX_CTL_A）的EN_CABLE_RESET位使能。

11.2.3.3 RX CRC32

接收数据CRC32由RX CRC32模块计算。使用TX CRC生成所用的逻辑来计算RX CRC。

编码的半字节数据将传送至RX CRC32，一次传送一个“半字节”，以用于CRC计算。数据包结束时，此模块的输出表示为接收到的数据包计算得出的CRC。值0xC704_DD7B表示良好CRC。检测到有效EOP码会锁存CRC值。将锁存的值与预期良好CRC余数值进行比较，此值由RX队列用于确定是否应更新RX队列状态以反映有效数据。

11.3 PD MAC BIST

PD MAC 包含 USB PD 规范中定义的 BIST 功能。它由 TX 和 RX 模块组成。

BIST TX 模块包含 PRBS（伪随机二进制序列）发生器、BIST 模式生成逻辑及其自身的位时序逻辑。TX BIST 测试帧使用的 SOP 类型是一个 20 位静态向量，它根据寄存器设置通过五个 SOP 有序集之间的复用来创建。发送数据包时，得到的 20 位向量仅基于位进行选择。

BIST 控制寄存器 A (BIST_CTL_A) 的 BIST_EN 位置 1 且 BIST_RX_EN 清零时进入 BIST TX。BIST TX 模式 (TX 类型) 通过 BIST 控制寄存器 A (BIST_CTL_A) 的 BIST_TX_MODE 字段设置，并通过先使用 BIST 控制寄存器 B (BIST_CTL_B) 的 BIST_TX_RST 位然后使用 BIST_TX_START/BIST_TX_STATUS 位进行启动。连续 BIST 发送 (模式 0、1、2、3 和 5) 应通过先使用 BIST_TX_RST 位然后再将 BIST_EN 位清零来停止。Tx 可能需要花费 1 个位时间来停止。之后 BIST_EN 才能清零。

BIST RX 模块包含 PRBS 发生器和位错误检测逻辑。BIST RX 仅在 BIST 接收器测试期间使用。当 BIST 控制寄存器 A (BIST_CTL_A) 的 BIST_EN 位以及 BIST_RX_EN 置 1 时进入 BIST RX。BIST 错误计数器使用 BIST 控制寄存器 B (BIST_CTL_B) 的 BIST_CLR_ERR_CNT 位复位。

有关 BIST 寄存器的信息，请参见第 168 页的第 11.4.5 节“PD MAC BIST 寄存器”。

11.4 供电MAC寄存器

本节详细介绍了供电MAC寄存器。有关器件的完整寄存器映射的概述，请参见第19页的第4.0节“寄存器映射”。

表11-5: 供电MAC寄存器映射

地址	寄存器名称 (符号)
1800h-1849h	PD MAC TX 队列/FIFO (74 字节) (更多相关信息, 请参见第11.1.1节“TX队列”)
184Ah-18FFh	保留供将来扩展使用
1900h-197Fh	PD MAC RX FIFO (128 字节) (更多相关信息, 请参见第11.2.1节“RX队列”)
1980h-19FFh	保留供将来扩展使用
1A00h	TX控制寄存器A (TX_CTL_A)
1A01h	TX状态寄存器 (TX_STAT)
1A02h	TX参数寄存器C (TX_PARAM_C)
1A03h	TX数据包长度寄存器 (TX_PKT_LEN)
1A04h	TX参数寄存器A (TX_PARAM_A)
1A05h	TX控制寄存器B (TX_CTL_B)
1A06h	TX参数寄存器B (TX_PARAM_B)
1A07h	TX位时间计数寄存器 (TX_BITTIME_CNT)
1A08h	TX周转时间寄存器 (TX_TA_TIME)
1A09h	TX中止状态寄存器 (TX_ABORT_STAT)
1A0Ah	TX自动响应中止状态寄存器 (TX_AR_ABORT_STAT)
1A0Bh	TX上电时间寄存器 (TX_POWER_UP_TIME)
1A0Ch	TX掉电时间寄存器 (TX_POWER_DOWN_TIME)
1A0Dh-1A3Fh	保留供将来扩展使用
1A40h	RX控制寄存器A (RX_CTL_A)
1A41h	RX控制寄存器B (RX_CTL_B)
1A42h	RX最大比特率位周期计数寄存器 (RX_BIT_PER_CNT_MAX_BR)
1A43h	RX最小比特率位周期计数寄存器 (RX_BIT_PER_CNT_MIN_BR)
1A44h	RX状态寄存器 (RX_STAT)
1A45h	RX数据包状态寄存器 (RX_PKT_STAT)
1A46h	RX tReceive时间寄存器 (RX_TRECEIVE_TIME)
1A47h	RX BadCRC数据包计数寄存器 (RX_BADCRC_PKT_CNT)
1A48h	RX重复数据包计数寄存器 (RX_DUP_PKT_CNT)
1A49h	RX硬复位检测窗口寄存器 (RX_HR_DET_WINDOW)
1A4Ah	RX最后一个GoodCRC数据包高字节寄存器 (RX_LAST_GCRC_PKT_HI)
1A4Bh	RX最后一个GoodCRC数据包低字节寄存器 (RX_LAST_GCRC_PKT_LO)
1A4Ch	RX报文ID存储寄存器 (RX_MSG_ID_STORED)
1A4Dh	RX最大数据包大小寄存器 (RX_MAX_SIZE)
1A4Eh-1A4Fh	保留供将来扩展使用
1A50h	RX FIFO 读指针寄存器 (RX_FIFO_RD_PTR)
1A51h	RX FIFO 读指针控制位寄存器 (RX_FIFO_RD_PTR_CTL_BITS)
1A52h	RX FIFO 写当前指针寄存器 (RX_FIFO_WR_CURRENT_PTR)
1A53h	RX FIFO 写当前指针控制位寄存器 (RX_FIFO_WR_CURRENT_PTR_CTL_BITS)
1A54h	RX FIFO 写头指针寄存器 (RX_FIFO_WR_HEAD_PTR)
1A55h	RX FIFO 写头指针控制位寄存器 (RX_FIFO_WR_HEAD_PTR_CTL_BITS)
1A56h	RX FIFO 写头指针加1寄存器 (RX_FIFO_WR_HEAD_PTR_PLUS_ONE)

UPD360

表11-5: 供电MAC寄存器映射 (续)

地址	寄存器名称 (符号)
1A57h	RX FIFO 写头指针加1控制位寄存器 (RX_FIFO_WR_HEAD_PTR_PLUS_ONE_CTL_BITS)
1A58h-1A7Fh	保留供将来扩展使用
1A80h	MAC 中断状态寄存器 (MAC_IRQ_STAT)
1A81h	TX 中断状态寄存器 (TX_IRQ_STAT)
1A82h	RX 中断状态寄存器 (RX_IRQ_STAT)
1A83h	RX 错误中断状态寄存器 (RX_ERR_IRQ_STAT)
1A84h	功耗管理中断状态寄存器 (PM_IRQ_STAT)
1A85h	RX SOP 中断状态寄存器 (RX_SOP_IRQ_STAT)
1A86h	TX 中断允许寄存器 (TX_IRQ_EN)
1A87h	RX 中断允许寄存器 (RX_IRQ_EN)
1A88h	RX 错误中断允许寄存器 (RX_ERR_IRQ_EN)
1A89h	功耗管理中断允许和控制寄存器 (PM_IRQ_EN)
1A8Ah	RX SOP 中断允许寄存器 (RX_SOP_IRQ_EN)
1A8Bh	复位控制寄存器 (RESET_CTL)
1A8Ch-1A9Fh	保留供将来扩展使用
1AA0h	BMC RX 高电平的最大完整位时间寄存器 (BMC_RX_HI_FB_MAX_TIME)
1AA1h	BMC RX 高电平的最小完整位时间寄存器 (BMC_RX_HI_FB_MIN_TIME)
1AA2h	BMC RX 低电平的最大完整位时间寄存器 (BMC_RX_LO_FB_MAX_TIME)
1AA3h	BMC RX 低电平的最小完整位时间寄存器 (BMC_RX_LO_FB_MIN_TIME)
1AA4h	BMC RX 高电平的最大半位时间寄存器 (BMC_RX_HI_HB_MAX_TIME)
1AA5h	BMC RX 高电平的最小半位时间寄存器 (BMC_RX_HI_HB_MIN_TIME)
1AA6h	BMC RX 低电平的最大半位时间寄存器 (BMC_RX_LO_HB_MAX_TIME)
1AA7h	BMC RX 低电平的最小半位时间寄存器 (BMC_RX_LO_HB_MIN_TIME)
1AA8h	BMC RX 静噪置为有效时间寄存器 (BMC_RX_SQL_ASSERT_TIME)
1AA9h	BMC RX 静噪保持时间寄存器 (BMC_RX_SQL_HOLD_TIME)
1AAAh-1AAFh	保留供将来扩展使用
1AB0h	BMC TX 位时间计数寄存器 (BMC_TX_BITTIME_CNT)
1AB1h	BMC 转换窗口时间寄存器 (BMC_TRANSITION_WINDOW_TIME)
1AB2h-1ABFh	保留供将来扩展使用
1AC0h	BIST 控制寄存器 A (BIST_CTL_A)
1AC1h	BIST 控制寄存器 B (BIST_CTL_B)
1AC2h	BIST 错误计数高字节寄存器 (BIST_ERR_CNT_HI)
1AC3h	BIST 错误计数低字节寄存器 (BIST_ERR_CNT_LO)
1AC4h-1AC6h	保留供将来扩展使用
1AC7h	BIST RX 状态寄存器 (BIST_RX_STAT)
1AC8h-1BFFh	保留供将来扩展使用

注: 在任何情况下都不能对保留的地址空间执行写操作。否则, 可能导致意外的操作与结果。

11.4.1 PD MAC TX 寄存器

以下小节介绍与PD MAC TX逻辑相关的各种寄存器。

11.4.1.1 TX控制寄存器A (TX_CTL_A)

地址: 1A00h 大小: 8位

Bit	说明	类型	默认值
7	保留	RO	-
6	RETRY_ON_LINE_BUSY 通常，如果在硬件尝试发送数据包时线路处于繁忙状态，则硬件将中止TX数据包（包括所有重试尝试）。（请注意，为响应软复位而发送GoodCRC ACK的情况例外，请参见DIS_SPCL_SR_GCRC_ACK位。） 将此位置1将强制硬件不完全中止发送，而是最多重试数据包N_RETRY_CNT次。如果在尝试N_RETRY_CNT次后失败，则会出现TX中断状态寄存器(TX_IRQ_STAT)的TX_FAILED状态，而非TX_ABORTED状态。 以下情况仍会中止TX数据包： • 接收到硬复位 • 接收到线缆复位（如果已使能，且待处理TX的SOP类型为SOP'、SOP"、SOP'_Debug或SOP"_Debug） • 接收到良好、非重复、非Ping和非GoodCRC数据包（包括软复位），且RX和TX SOP类型匹配，或者RX SOP类型为SOP，而TX SOP类型不为SOP（可禁止） • 通过软件将TX控制寄存器B (TX_CTL_B) 的ABORT位置1。	R/W	0b
5	DIS_SOP_ABRTS_NON_SOP 通常，接收到SOP数据包时，待处理的非SOP数据包（包括等待GoodCRC）将被中止。该位置1后，非SOP数据包只会由相同的非SOP类型的数据包中止。	R/W	0b
4	DIS_SPCL_SR_GCRC_ACK 禁止用于响应软复位的GoodCRC ACK的特殊处理。 默认情况下，在自动响应模式下，线路繁忙状态不会中止用于响应软复位接收的GoodCRC ACK发送。将该位置1将禁止此行为，而用于软复位的GoodCRC ACK发送将按照其他数据包的GoodCRC ACK进行处理。 注： 此位置1时，WAIT4LINE_IDLE位也会影响用于软复位的GoodCRC ACK。	R/W	0b

UPD360

Bit	说明	类型	默认值
3	WAIT4LINE_IDLE 通常，如果在硬件尝试发送数据包时线路处于繁忙状态，则硬件将中止TX数据包（包括所有重试尝试）。（请注意，为响应软复位而发送GoodCRC ACK的情况例外，请参见DIS_SPCL_SR_GCRC_ACK位。） 将该位置1将强制硬件不中止发送，而是等到线路空闲，然后发送报文。 RETRY_ON_LINE_BUSY 选项的优先级低于此功能。 以下情况仍会中止TX数据包： • 接收到硬复位 • 接收到线缆复位（如果已使能，且待处理TX的SOP类型为SOP'、SOP"、SOP'_Debug或SOP"_Debug） • 接收到良好、非重复数据包、非 Ping 和非 GoodCRC（包括软复位），且RX和TX SOP类型匹配，或者RX SOP类型为SOP，而TX SOP类型不为SOP（可禁止） • 通过软件将TX控制寄存器B（TX_CTL_B）的ABORT位置1。	R/W	0b
2	EN_AUTO_RSP_MODE 使硬件能够自动发送GoodCRC报文和BIST错误计数报文。 0：禁止自动发送GoodCRC报文（默认值） 1：使能自动发送GoodCRC报文 注： 禁止自动响应模式也将禁止硬件自动重试，即，硬件将忽略TX参数寄存器C（TX_PARAM_C）的N_RETRY_CNT字段。	R/W	0b
1	EN_FMQ 在TX队列的写入接口上使能FIFO模式排队。 0：缓冲区模式——以类似于缓冲区/CSR的方式访问TX队列（默认值） 1：FIFO模式	R/W	0b
0	EN_RMDP 使能原始模式数据处理。 0：自动模式数据处理（默认值） 1：使能原始模式数据处理	R/W	0b

11.4.1.2 TX状态寄存器 (TX_STAT)

地址: 1A01h 大小: 8位

Bit	说明	类型	默认值
7	保留	RO	-
6:4	N_HW_RETRIES 硬件尝试成功发送报文的次数。 如果TX控制寄存器B (TX_CTL_B) 的GO位置1, 则该字段自动清零。 只有在TX成功后, 该字段才有意义。这仅供参考。 0: 初始报文成功。该值必须与TX_ABORTED位相关。如果TX报文被中止, 此位无意义。 >0: 器件执行的重试次数。最大值取决于TX参数寄存器C (TX_PARAM_C) 寄存器的N_RETRY_CNT字段。	RO	000b
3:1	保留	RO	-
0	TX_ACTIVE 发送器激活。 0: 发送器空闲 1: 发送器激活 软件可监视此位, 以确定是否正在进行发送。将该位置为无效并不意味着报文发送成功, 仅表示发送器不再有效且已停止发送。	RO	0b

11.4.1.3 TX参数寄存器C (TX_PARAM_C)

地址：1A02h 大小：8位

Bit	说明	类型	默认值
7	PORT_DATA_ROLE 将该字段用作使用SOP信号的数据包的硬件生成数据包报头的端口数据角色字段。 0: UFP 1: DFP 注： SOP'和SOP"始终为端口数据角色发送一个0位。	R/W	0b
6:4	N_RETRY_CNT 允许的重试次数。 将该字段设置为0可禁止自动硬件重试（即，软件进行重试时）。 如果N_RETRY_CNT设置为零，但预计硬件等待GoodCRC，则TX参数寄存器A (TX_PARAM_A) 的EXPECT_GOODCRC位应置1，否则，数据包发送完成后会立即将TX视为完成。 注： 由于没有反馈用于指示故障，因此硬复位和线缆复位不会自动重试。	R/W	000b
3	CABLE_PLUG 将该字段用作使用SOP'和SOP"信号的数据包的硬件生成数据包报头的线缆插头字段。 0: DFP或UFP 1: 线缆插头	R/W	0b
2	PORT_POWER_ROLE 将该字段用作使用SOP信号的数据包的硬件生成数据包报头的端口电源角色字段。 0: 电流阱 1: 电流源	R/W	0b
1:0	保留	RO	-

11.4.1.4 TX数据包长度寄存器 (TX_PKT_LEN)

地址: 1A03h 大小: 8位

Bit	说明	类型	默认值
7:6	保留	RO	-
5:0	PKT_LEN 排队的数据包长度，以字节为单位。 硬件在自动模式下使用此值来确定何时插入CRC。 PKT_LEN = 报头中的字节数 + 数据对象中的字节数 注: 这仅适用于自动模式队列，不适用于原始模式队列。原始模式队列有特殊的代码来插入CRC和终止发送。 注: 对于该字段，值0无效。	R/W	000000b

11.4.1.5 TX参数寄存器A (TX_PARAM_A)

地址: 1A04h 大小: 8位

Bit	说明	类型	默认值
7	EXPECT_GOODCRC 当TX参数寄存器C (TX_PARAM_C) 的N_RETRY_CNT字段设为0，但预计硬件等待用于响应TX报文的GoodCRC时，使用此位。 如果N_RETRY_CNT非零且TX控制寄存器A (TX_CTL_A) 的EN_AUTO_RSP_MODE置1，则硬件自动等待GoodCRC，此位无效。	R/W	0b
6:4	TX_SOP_SELECT 该字段选择用于正常TX数据包、返回的BIST计数器数据包（用于响应RX BIST测试帧）和TX BIST测试帧的SOP。 在使能自动重试时，该值还用于验证GoodCRC响应中的预期SOP类型。 000: SOP 001: SOP' 010: SOP" 011: SOP'_Debug 100: SOP"_Debug 101 - 111: 保留 注: 对于自动重试模式，假设还在RX控制寄存器B (RX_CTL_B) 中使能预期SOP类型。	R/W	000b

UPD360

Bit	说明	类型	默认值
3	EN_FWTX 使能软件发出的发送。 该位只有在软件将 GO 位置 1 时有效，用于涵盖软件无法从 GO 位置 1 后退以响应有效 RX 的冲突。 如果 RX 中断状态寄存器 (RX_IRQ_STAT) 指示接收到硬复位或线缆复位，该位将由硬件自动保持清零；或者，如果 RX FIFO 中存在数据，则软件必须在每次要发出发送之前将其置 1。 该位清零后，软件发出的发送将被中止。 注： 该位不是中止位。将该位清零不会影响已向硬件发出的发送。硬件将相应地自动中止这类发送。要中止正在进行的发送，请使用 TX 控制寄存器 B (TX_CTL_B) 的 ABORT 位。 注： 将该位置 1 不会阻止硬件发出的发送。 注： 此位通过将 PD_RESET 置为有效进行复位。	R/W	0b
2:0	MSG_ID 报文 ID。该值用于验证 GoodCRC 响应中的预期报文 ID。 注： 硬件不会将此值插入 TX 报文报头中。为实现适当的正常操作，软件必须在报文报头的 bit[11:9] 中设定相同的值。 注： 软件在处理硬复位、线缆复位或软复位时，必须将其设为零。	R/W	000b

11.4.1.6 TX控制寄存器B (TX_CTL_B)

地址: 1A05h 大小: 8位

Bit	说明	类型	默认值
7:6	保留	RO	-
5	TX_CABLE_RESET 将该位和GO位置1会导致为自动模式数据处理配置的硬件发送线缆复位。TX队列和相关寄存器中的任何数据都将被忽略。 此位不适用于原始模式数据处理，因为可在TX队列中明确指定线缆复位数据包。对于原始模式，将忽略此位。 只有在GO位当前未置1时，此位才能置1。 TX_HARD_RESET位具有较高优先级，如果与该位一起置1，则该位不会置1。 数据包发送结束时，该位由硬件清零。此位通过将PD_RESET置为有效进行复位。软件只能向该位写入1，因此在置1后无法清零。 注： 线缆复位不会自动重试（没有反馈用于指示故障）。由软件执行线缆复位重试。 注： 在完成数据包发送后，该位将清零，因此只有在完成前一次发送后GO位清零之后该位才应置1。	W1S/SC	0b
4	OK_TO_TX 通道空闲，可以发送。在GO位置1前，软件应监视此位。 0：软件不应将GO位置1 1：软件可将GO位置1。 在以下情况下将导致硬件将此位设为0： - BMC线路空闲定时器指示线路繁忙。 - 硬件可能正在处理对所接收数据包的自动响应。	RO	1b
3	RST_TXQ_FIFO_WRI_PTR 将该位置1将复位TX队列的写入接口FIFO指针。使用FIFO模式排队时，软件必须始终在将数据放入FIFO之前向此位写入1。 硬件不会锁存此位，它将回读为零。硬件会在GO位清零时自动复位指针（即，硬件完成当前发送）。	WO	0b

UPD360

Bit	说明	类型	默认值
2	TX_HARD_RESET 将该位和GO位置1会导致为自动模式数据处理配置的硬件发送硬复位。TX队列和相关寄存器中的任何数据都将被忽略。 此位不适用于原始模式数据处理，因为可在TX队列中明确指定硬复位数据包。对于原始模式，将忽略此位。 只有在GO位当前未置1时，此位才能置1。 数据包发送结束时，该位由硬件清零。此位通过将PD_RESET置为有效进行复位。软件只能向该位写入1，因此在置1后无法清零。 注： 硬复位不会自动重试（没有反馈用于指示故障）。由软件执行硬复位重试。 注： 在完成数据包发送后，该位将清零，因此只有在完成前一次发送后GO位清零之后该位才应置1。	W1S/SC	0b
1	ABORT 将该位置1可尽快中止当前正在进行的发送。写0不起作用。 硬件不会锁存此位，它将回读为零。 硬件将完成当前半字节发送，发送EOP，然后终止发送。如果该位在前导码阶段置1，则前导码的当前位将在终止前完成。 软件应使用TX中断状态寄存器（TX_IRQ_STAT）的TX_EOP状态位来确定当前数据包何时终止。	WO	0b
0	GO 向该位写入1将： - 开始将TX队列中的数据包发送到线路或： - 在该寄存器的TX_HARD_RESET位置1且数据处理模式设为自动（TX控制寄存器A（TX_CTL_A）的EN_RMDP位未置1）时发送硬复位。 - 在该寄存器的TX_CABLE_RESET位置1且数据处理模式设为自动（TX控制寄存器A（TX_CTL_A）的EN_RMDP位未置1）时发送线缆复位。 发送完成后，硬件将此位清零。此位通过将PD_RESET置为有效进行复位。软件只能向该位写入1，因此在置1后无法清零。硬件将此位清零并不表示报文已成功发送。 硬件可能会检测到总线冲突并中止发送。在这种情况下，TX中断状态寄存器（TX_IRQ_STAT）的TX_ABORTED位将置1，并将产生至MCU的中断。	W1S/SC	0b

11.4.1.7 TX参数寄存器B (TX_PARAM_B)

地址: 1A06h 大小: 8位

Bit	说明	类型	默认值
7	保留	RO	-
6:0	PREAMBLE_LEN 要发送的前导码位数。 范围: 1-127, 规范: 64d (默认值) 值0无效, 硬件不会检查有效性。软件应使用默认值进行正常操作。 将该字段设置为奇数值将导致违反USB PD规范, 规范规定“前导码应以0开始, 并以1结束”。奇数值将导致前导码以0值开始, 但也将以0值结束。这可能会也可能不会导致功能问题。为保持符合规范, 应将该字段设置为偶数值。	R/W	1000000b

11.4.1.8 TX位时间计数寄存器 (TX_BITTIME_CNT)

地址: 1A07h 大小: 8位

Bit	说明	类型	默认值
7:0	BIT_TIME_CNT 位时间计数器值。TX逻辑使用此值来确定发送的位时间 (标称比特率为300 kbps)。可使用该寄存器来调整位时间以进行测试 (调整比特率)。 计数值基于时钟频率, 可通过下式得出: $((\text{时钟频率 kHz} / \text{比特率 kbps}) - 1)$ 即, 时钟频率 = 48000 kHz (48 MHz) 标称比特率 = 300 kbps $\text{BIT_TIME_CNT} = (48000 / 300) - 1 = 159$	R/W	00h

11.4.1.9 TX周转时间寄存器（TX_TA_TIME）

地址：1A08h 大小：8位

该寄存器控制IFG定时器。

Bit	说明	类型	默认值
7:6	保留	RO	-
5:0	TA_TIME 周转时间，单位为μs。 范围：1-63，规范：25 μs 值0无效，硬件不会检查有效性。 硬件使用自由运行的1 μs脉冲发生器，因此该字段的值应比所需值多1，以确保最短时间（26d）。	R/W	010101b

11.4.1.10 TX中止状态寄存器（TX_ABORT_STAT）

地址：1A09h 大小：8位

该字段与TX_ABORTED中断搭配使用，并提供有关TX中止原因的详细信息。该寄存器中的位是粘性的，即，硬件将其中的任一位置1后，该字段便无法由硬件更新，直至软件将该字段清零。

注： 该寄存器中的位由硬件置1，且必须由软件清零。这些位不互斥，即，可根据具体情况将多个位置1。

Bit	说明	类型	默认值
7	TX_ABORT_STAT[7] 在等待GoodCRC期间，TX因协议错误（即，在等待GoodCRC或接收硬复位或线缆复位期间，接收到除GoodCRC或Ping报文外的其他报文）而中止。	R/W1C	0b
6	TX_ABORT_STAT[6] 在软件发出的TX期间，TX因在等待总线周转或等待线路空闲时出现RX（包括硬复位或线缆复位）而中止。	R/W1C	0b
5	TX_ABORT_STAT[5] 自动重试因在等待总线周转或等待线路空闲时出现RX（包括硬复位或线缆复位）而中止。	R/W1C	0b
4	TX_ABORT_STAT[4] 自动重试因总线周转后线路繁忙而中止。	R/W1C	0b
3	保留	RO	-
2	TX_ABORT_STAT[2] 在软件发出的TX期间，总线周转后线路处于繁忙状态。	R/W1C	0b
1	TX_ABORT_STAT[1] 在软件发出的TX正在进行时，软件发出中止命令。	R/W1C	0b
0	TX_ABORT_STAT[0] 软件在TX禁止时发出GO命令。	R/W1C	0b

11.4.1.11 TX自动响应中止状态寄存器（TX_AR_ABORT_STAT）

地址：1A0Ah 大小：8位

该字段与 **AUTO_RSP_ABORTED** 中断搭配使用，并提供有关TX中止原因的详细信息。该寄存器中的位是粘性的，即，硬件将其中的任一位置1后，该字段便无法由硬件更新，直至软件将该字段清零。

注： 该寄存器中的位由硬件置1，且必须由软件清零。这些位不互斥，即，可根据具体情况将多个位置1。

Bit	说明	类型	默认值
7	保留	RO	-
6	TX_AR_ABORT_STAT[6] 自动响应因在等待总线周转或等待线路空闲时出现RX（包括硬复位或线缆复位）而中止。	R/W1C	0b
5:4	保留	RO	-
3	TX_AR_ABORT_STAT[3] 在自动响应期间，经过总线周转时间后线路处于繁忙状态。	R/W1C	0b
2	保留	RO	-
1	TX_AR_ABORT_STAT[1] 软件在自动响应正在进行的过程中发出中止命令。	R/W1C	0b
0	保留	RO	-

11.4.1.12 TX上电时间寄存器（TX_POWER_UP_TIME）

地址：1A0Bh 大小：8位

Bit	说明	类型	默认值
7:0	TX_POWER_UP_TIME 发送前的TX上电时间（单位为8个时钟周期）。 范围：0-255（实际时间要长5个时钟。） 模拟组件至少需要5 μs的时间来上电。 计数值基于时钟频率，可通过下式得出： $TX_POWER_UP_TIME = \text{上电时间} \mu s \times \text{时钟频率} MHz / 8$ 即， 时钟频率 = 48 MHz 上电时间 = 5 μs $TX_POWER_UP_TIME = 5 \times 48 / 8 = 30$	R/W	00h

11.4.1.13 TX掉电时间寄存器（TX_POWER_DOWN_TIME）

地址：1A0Ch 大小：8位

Bit	说明	类型	默认值
7:0	TX_POWER_DOWN_TIME 发送后的TX掉电时间（单位为8个时钟周期）。 范围：0-255（实际时间要长2个时钟。） 模拟组件至少需要经过1 μs的时间后才会掉电。 计数值基于时钟频率，可通过下式得出： TX_POWER_DOWN_TIME = 掉电时间μs x 时钟频率MHz / 8 即， 时钟频率 = 48 MHz 掉电时间 = 1 μs TX_POWER_DOWN_TIME = 1 x 48 / 8 = 6	R/W	00h

11.4.2 PD MAC RX 寄存器

以下小节介绍与PD MAC RX逻辑相关的各种寄存器。

11.4.2.1 RX控制寄存器A (RX_CTL_A)

地址：1A40h 大小：8位

Bit	说明	类型	默认值
7:6	保留	RO	-
5	EN_LOOPBACK 使能环回。该位允许MAC接收其自身的发送。 注： 环回发生在TX和RX共同连接的引脚。	R/W	0b
4	EN_RCV 使能接收器。软件应在配置完接收器后将此位置1。将此位置1会使接收器开始正常工作。 如果在接收活动正在进行时将此位置1，则会在当前接收活动结束后生效。如果在处理接收的过程中将此位清零，则会在当前接收完成后生效。 在接收到硬复位或线缆复位时，该位将由硬件自动清零，以阻止正常的数据包接收。 在POR或SW_RESET后，硬复位和线缆复位接收被禁止，直至EN_RCV置1。之后，将EN_RCV清零（通过软件或硬件）对硬复位或线缆复位的接收无影响，即，即使将EN_RCV清零，硬件也会处理硬复位或线缆复位。 如果软件要在EN_RCV置1后重新初始化接收时序参数，则必须先使用SW_RESET复位UPD_MAC_v2。将PD_RESET置为有效不会复位此位。	R/W	0b
3	RST_RECEIVER 将该位置1会复位接收器队列。如果接收器硬件可能与软件不同步，则软件应将该位置1。硬件会锁存此位，因此该位必须由软件清零以恢复接收器操作。当RESET_CTL寄存器的PD_RESET位置1时，接收器硬件也会复位。	R/W	0b

UPD360

Bit	说明	类型	默认值
2	EN_CRC_RCV_TMR 使能CRCReceiveTimer。 0: 禁止CRCReceiveTimer（默认值） 1: 使能CRCReceiveTimer 只有当软件未使用自动重试模式时，才会使用此位。 使能自动重试（TX控制寄存器A（TX_CTL_A）的EN_AUTO_RSP_MODE置1，N_RETRY_CNT为非零值）时，或者将发送器设置为等待GoodCRC报文（EXPECT_GOODCRC置1，N_RETRY_CNT为0）时，CRCReceiveTimer由硬件自动使能。如果在定时器超时前未接收到GoodCRC报文，则会产生中断。TX中断状态寄存器（TX_IRQ_STAT）的CRC_RCV_TIMEOUT位将置1。 如果自动重试和等待GoodCRC被禁止，则软件可将此位置1以在开始发送前使能CRCReceiveTimer，进而帮助对tReceive计时。使能后，该定时器将在发送结束时自动启动；不过，软件在接收到GoodCRC报文时必须禁止此定时器。如果不禁止，则定时器超时时将产生中断。 超时后，CRCReceiveTimer保持在超时状态，直至重新加载时开始另一个数据包发送。 该定时器还可作为BISTReceiveErrorTimer用于BIST发送。	R/W	0b
1	EN_SMBUS_MODE 使能SMBus块读取FIFO格式。 0: 数据包中的第一个存储单元为状态，第二个存储单元为数据包长度 1: 数据包中的第一个存储单元为数据包长度加1，第二个存储单元为状态	R/W	0b
0	EN_CABLE_RESET 使能线缆复位检测。 0: 禁止线缆复位检测（默认值） 1: 使能线缆复位检测	R/W	0b

11.4.2.2 RX控制寄存器B (RX_CTL_B)

地址: 1A41h 大小: 8位

Bit	说明	类型	默认值
7	保留	RO	-
6:2	RX_SOP_ENABLE 使能要接收的SOP类型。包括RX BIST测试帧。 Bit 2——SOP Bit 3——SOP' Bit 4——SOP" Bit 5——SOP'_Debug Bit 6——SOP"_Debug 允许软件随时更改字段。内部副本在有效接收期间被保留，并在空闲期间从寄存器更新。	R/W	00001b
1:0	保留	RO	-

11.4.2.3 RX最大比特率位周期计数寄存器 (RX_BIT_PER_CNT_MAX_BR)

地址: 1A42h 大小: 8位

该寄存器的值连同RX最小比特率位周期计数寄存器 (RX_BIT_PER_CNT_MIN_BR) 的值共同用于确定RX信号的有效位时间（也称为位周期）。

$RX_BIT_PER_CNT_MIN_BR > \text{有效位时间} > RX_BIT_PER_CNT_MAX_BR$ 。

Bit	说明	类型	默认值
7:0	RX_BIT_PER_CNT_MAX_BR 最大比特率时的位周期计数。 $= (((\text{clock_freq_kHz} / \text{max_bit_rate_kbps}) * (100 - \text{容差})) / 100) - 1$ 警告：如果在C代码中定义宏，则进行数学运算时必须避免整数溢出。	R/W	00h

11.4.2.4 RX最小比特率位周期计数寄存器 (RX_BIT_PER_CNT_MIN_BR)

地址：1A43h 大小：8位

该寄存器的值连同RX最大比特率位周期计数寄存器 (RX_BIT_PER_CNT_MAX_BR) 的值共同用于确定RX信号的有效位时间（也称为位周期）。

RX_BIT_PER_CNT_MIN_BR > 有效位时间 > RX_BIT_PER_CNT_MAX_BR。

Bit	说明	类型	默认值
7:0	RX_BIT_PER_CNT_MIN_BR 最小比特率时的位周期计数。 $= (((\text{clock_freq_kHz} / \text{min_bit_rate_kbps}) * (100 + \text{容差}) / 100) - 1)$ 警告：如果在C代码中定义宏，则进行数学运算时必须避免整数溢出。	R/W	00h

11.4.2.5 RX状态寄存器 (RX_STAT)

地址：1A44h 大小：8位

Bit	说明	类型	默认值
7:2	保留	RO	-
1	RX_IN_PROCESS 接收器激活。 硬件正在接收数据包，即，线路繁忙，发送器未激活。	RO	0b
0	LINE_STATE 0: 线路空闲 1: 线路繁忙（BMC线路空闲定时器指示线路繁忙） 如果此位为1，则可能正在进行接收，软件应等待报文接收完成。如果为0，则软件可将GO位置1。 硬件在最后一刻前执行进一步操作，以避免总线争用。不过，总线争用情况仍可能出现，无法完全避免。 如果硬件执行了预防性措施以避免总线争用，TX数据包将被中止。将通过中断和将TX中断状态寄存器 (TX_IRQ_STAT) 的TX_ABORTED位置1来将这种情况通知给软件。	RO	0b

11.4.2.6 RX数据包状态寄存器 (RX_PKT_STAT)

地址: 1A45h

大小: 8位

该寄存器仅适用于采用自动响应/重试模式的情况。该寄存器与以下位结合使用来提供有关已接收数据包状态的附加信息:

- RX中断状态寄存器 (RX_IRQ_STAT) 的RX_DONE位
- RX错误中断状态寄存器 (RX_ERR_IRQ_STAT) 的RX_PCOL_ERROR位
- RX错误中断状态寄存器 (RX_ERR_IRQ_STAT) 的RX_PKT_DROPPED位
- RX错误中断状态寄存器 (RX_ERR_IRQ_STAT) 的RX_BUF_OVR_RUN位

该寄存器中的位是粘性的, 即, 硬件将其中的任一位置1后, 该字段便无法由硬件更新, 直至软件将该字段清零。

值0x00表示无错误, 而值0xFF表示数据包状态当前未知。其他值按照该寄存器的位定义规定进行解析。

注: 不存在用于表示接收到带有匹配MSG_ID的GoodCRC报文的状态。成功完成TX表示接收到有效GoodCRC ACK。

注: 该寄存器中的位由硬件置1, 且必须由软件清零。这些位不互斥, 即, 可根据具体情况将多个位置1。

Bit	说明	类型	默认值
7	保留	RO	-
6	RX_PKT_STAT[6] ODD_NIBBLES。接收到的数据包因在数据阶段具有奇数个半字节而被丢弃。无论是否存在不良CRC、符号错误或奇数个半字节, 只要RX_PKT_STAT[3] (RX_OVER_SIZE) 置1, 该位就不置1。RX_BAD_PKT中断置为有效。	R/W1C	0b
5	RX_PKT_STAT[5] SYM_ERROR。数据包中存在符号错误, 但未报告BAD_CRC。无论是否存在不良CRC、符号错误或奇数个半字节, 只要RX_PKT_STAT[3] (RX_OVER_SIZE) 置1, 该位就不置1。RX_BAD_PKT中断置为有效。	R/W1C	0b
4	RX_PKT_STAT[4] RX_FIFO_FULL。接收到的数据包因在SOP或数据期间FIFO已满而必须被丢弃。RX_BUF_OVR_RUN中断置为有效。	R/W1C	0b
3	RX_PKT_STAT[3] RX_OVER_SIZE。接收到的数据包因大于RX最大数据包大小寄存器中设置的最大大小而必须被丢弃。RX_BAD_PKT中断置为有效。	R/W1C	0b
2	RX_PKT_STAT[2] PCOL_ERROR。预期接收到GoodCRC时, 接收到除GoodCRC或Ping外的其他报文。RX_PCOL_ERROR中断置为有效。	R/W1C	0b
1	RX_PKT_STAT[1] DUPLICATE_PACKET。适用于自动响应模式。RX_PKT_DROPPED中断置为有效。	R/W1C	0b
0	RX_PKT_STAT[0] BAD_CRC。RX数据包具有不良CRC。RX_BAD_PKT中断置为有效。如果发生符号错误或出现奇数个半字节计数, 则该位也被强制置1。无论是否存在不良CRC、符号错误或奇数个半字节, 只要RX_PKT_STAT[3] (RX_OVER_SIZE) 置1, 该位就不置1。	R/W1C	0b

11.4.2.7 RX tReceive时间寄存器（RX_TRECEIVE_TIME）

地址：1A46h 大小：8位

Bit	说明	类型	默认值
7:0	TRECEIVE CRCReceiveTimer超时值（tReceive）以10倍微秒数为单位。 超时值 = 字段值 * 10（μs） 范围：0-255，规范：900-1100 μs（90-110） 注： 值0无效，硬件不会检查有效性。 注： 硬件使用自由运行的1 μs脉冲发生器，因此，此CRCReceiveTimer可能比预期值少1 μs。 注： 将复位值设定为1000 μs（+/- 2%）。	R/W	64h

11.4.2.8 RX BadCRC数据包计数寄存器（RX_BADCRC_PKT_CNT）

地址：1A47h 大小：8位

该寄存器跟踪接收到的具有不良CRC的数据包（任何具有有效SOP但具有不良CRC、无效EOP或符号错误的数据包）数。该寄存器仅用于调试。
如果使能，则当该寄存器的计数达到128时会产生中断。

Bit	说明	类型	默认值
7:0	BADCRC_PKT_CNT 接收到的不良数据包（即，具有不良CRC、无效EOP或符号错误的数据包）的数量。 该寄存器的值大于127时，RX中断状态寄存器（RX_IRQ_STAT）的DBG_EVENT位将置1。 对该寄存器进行的任何写入操作都会将所有位清零。 注： 该寄存器不受在BIST接收器测试期间接收到的数据包影响。所有这些数据包均损坏。	R/WAC	00h

11.4.2.9 RX重复数据包计数寄存器（RX_DUP_PKT_CNT）

地址：1A48h 大小：8位

该寄存器跟踪接收到的重复数据包的数量。该寄存器仅适用于使能自动响应模式的情况。该寄存器仅用于调试。如果使能，则当该寄存器的计数达到128时会产生中断。

Bit	说明	类型	默认值
7:0	DUP_PKT_CNT 接收到的重复数据包的数量。适用于使能自动响应模式的情况。 该寄存器的值大于127时，RX中断状态寄存器（RX_IRQ_STAT）的DBG_EVENT位将置1。 对该寄存器进行的任何写入操作都会将所有位清零。	R/WAC	00h

11.4.2.10 RX硬复位检测窗口寄存器（RX_HR_DET_WINDOW）

地址：1A49h 大小：8位

Bit	说明	类型	默认值
7	保留	RO	-
6:0	RX_HR_DET_WINDOW 硬复位检测窗口的值，以位数为单位。 该字段决定何时将检测到的硬复位或线缆复位视为有效。	R/W	1011110b

11.4.2.11 RX最后一个GoodCRC数据包高字节寄存器（RX_LAST_GCRC_PKT_HI）

地址：1A4Ah 大小：8位

Bit	说明	类型	默认值
7:0	RX_LAST_GCRC_PKT_HI 接收到的GoodCRC报文的Bit[15:8]。 该寄存器仅适用于使能自动重试的情况。由于在自动重试模式下GoodCRC报文未存储在RX队列中，因此该寄存器为软件提供了一种检查接收到的最后一个GoodCRC报文的方法。	RO	00h

11.4.2.12 RX最后一个GoodCRC数据包低字节寄存器（RX_LAST_GCRC_PKT_LO）

地址：1A4Bh 大小：8位

Bit	说明	类型	默认值
7:0	RX_LAST_GCRC_PKT_LO 接收到的GoodCRC报文的Bit [7:0]。 该寄存器仅适用于使能自动重试的情况。由于在自动重试模式下GoodCRC报文未存储在RX队列中，因此该寄存器为软件提供了一种检查接收到的最后一个GoodCRC报文的方法。	RO	00h

11.4.2.13 RX报文ID存储寄存器（RX_MSG_ID_STORED）

地址：1A4Ch 大小：8位

该寄存器指示是否为每个SOP类型存储最后一个报文ID。写入适当的位会清除所指示SOP类型的最后一个存储的报文ID。清除后，不会将针对相应SOP类型接收到的下一个数据包视为重复。软件在处理软复位时应将相应位清零。

该寄存器中的位由硬件置1，且必须由软件清零。

软件通过复位控制寄存器（RESET_CTL）发出SW_RESET或PD_RESET命令时该寄存器也会清零。

Bit	说明	类型	默认值
7:5	保留	RO	-
4:0	RX_MSG_ID_STORED 该字段指示为SOP类型存储的最后一个报文ID。 Bit 0: SOP Bit 1: SOP' Bit 2: SOP" Bit 3: SOP' _Debug Bit 4: SOP" _Debug	W1C	00000b

11.4.2.14 RX最大数据包大小寄存器（RX_MAX_SIZE）

地址：1A4Dh 大小：8位

该寄存器的值用于设置最大接收数据包大小，其中包括4字节的CRC。

Bit	说明	类型	默认值
7:0	RX_MAX_SIZE 包括4字节CRC的最大接收数据包大小。	R/W	22h

11.4.2.15 RX FIFO 读指针寄存器 (RX_FIFO_RD_PTR)

地址: 1A50h 大小: 8位

该寄存器包含RX FIFO 读指针。软件可根据已知数据包长度和起点来更新此寄存器，进而清除数据包。

如果软件要更新该寄存器，则软件必须保持RX FIFO 读指针控制位寄存器与该寄存器一致。

为保持一致性，对该寄存器的写操作保存在临时寄存器中，并在写入RX FIFO 读指针控制位寄存器时传送至实际寄存器。

Bit	说明	类型	默认值
7	保留	RO	-
6:0	RX_FIFO_RD_PTR 该字段包含RX FIFO 读指针。	R/W	0000000b

11.4.2.16 RX FIFO 读指针控制位寄存器 (RX_FIFO_RD_PTR_CTL_BITS)

地址: 1A51h 大小: 8位

该寄存器包含RX FIFO 读指针的辅助控制位。

如果软件要更新RX FIFO 读指针，则必须使这些位与其保持一致。

Bit	说明	类型	默认值
7	RX_FIFO_EMPTY 该字段表示RX FIFO 为空 (RX FIFO 写头指针等于RX FIFO 读指针)。	RO	1b
6:2	保留	RO	-
1	RX_FIFO_RD_PTR_AT_TOP 该字段表示RX FIFO 读指针处于RAM的顶部 (存储单元127)。	RO	0b
0	RX_FIFO_RX_PTR_WRAP 每当RX FIFO 读指针在RAM的顶部 (存储单元127) 绕回时，该字段都会翻转。如果软件更新RX FIFO 读指针以使其绕回，则必须翻转该位。	R/W	0b

11.4.2.17 RX FIFO 写当前指针寄存器 (RX_FIFO_WR_CURRENT_PTR)

地址：1A52h 大小：8 位

该寄存器包含RX FIFO写当前指针。该指针指示下一个接收到的数据字节的存储位置。该寄存器仅用于调试和测试，通常不应被写入。

软件必须保持该寄存器与其他写指针寄存器之间的一致性。

不存在硬件保持寄存器来保持此寄存器与其他写指针寄存器之间的写一致性。假设该寄存器只在无有效接收的情况下被写入。

Bit	说明	类型	默认值
7	保留	RO	-
6:0	RX_FIFO_WR_CURRENT_PTR 该字段包含RX FIFO写当前指针。	R/W	0000000b

11.4.2.18 RX FIFO 写当前指针控制位寄存器 (RX_FIFO_WR_CURRENT_PTR_CTL_BITS)

地址：1A53h 大小：8 位

该寄存器包含RX FIFO写当前指针的辅助控制位。该寄存器仅用于调试和测试，通常不应被写入。

软件必须保持该寄存器与其他写指针寄存器之间的一致性。

不存在硬件保持寄存器来保持此寄存器与其他写指针寄存器之间的写一致性。假设该寄存器只在无有效接收的情况下被写入。

Bit	说明	类型	默认值
7	RX_FIFO_FULL 该字段表示RX FIFO已满（RX FIFO写当前指针等于RX FIFO读指针）。	RO	0b
6:2	保留	RO	-
1	RX_FIFO_WR_CURRENT_PTR_AT_TOP 该字段表示RX FIFO写当前指针处于RAM的顶部（存储单元127）。	RO	0b
0	RX_FIFO_WR_CURRENT_PTR_WRAP 每当RX FIFO写当前指针在RAM的顶部（存储单元127）绕回时，该位都会翻转。	R/W	0b

11.4.2.19 RX FIFO 写头指针寄存器 (RX_FIFO_WR_HEAD_PTR)

地址: 1A54h 大小: 8位

该寄存器包含RX FIFO写头指针。该指针指示下一个接收到的数据包起始位置。该寄存器仅用于调试和测试，通常不应被写入。

软件必须保持该寄存器与其他写指针寄存器之间的一致性。

不存在硬件保持寄存器来保持此寄存器与其他写指针寄存器之间的写一致性。假设该寄存器只在无有效接收的情况下被写入。

Bit	说明	类型	默认值
7	保留	RO	-
6:0	RX_FIFO_WR_HEAD_PTR 该字段包含RX FIFO写头指针。	R/W	0000000b

11.4.2.20 RX FIFO 写头指针控制位寄存器 (RX_FIFO_WR_HEAD_PTR_CTL_BITS)

地址: 1A55h 大小: 8位

该寄存器包含RX FIFO写头指针的辅助控制位。该寄存器仅用于调试和测试，通常不应被写入。

软件必须保持该寄存器与其他写指针寄存器之间的一致性。

不存在硬件保持寄存器来保持此寄存器与其他写指针寄存器之间的写一致性。假设该寄存器只在无有效接收的情况下被写入。

Bit	说明	类型	默认值
7	RX_FIFO_NO_ROOM_FOR_HEADER 该字段表示RX FIFO没有空间用于RX报头（状态和长度）（RX FIFO写头指针等于RX FIFO读指针或RX FIFO写头指针加1等于RX FIFO读指针）。	RO	0b
6:3	保留	RO	-
2	RX_FIFO_WR_HEAD_PTR_AT_2ND_FROM_TOP 该字段表示RX FIFO写头指针处于RAM顶部减1的位置（存储单元126）。	RO	0b
1	RX_FIFO_WR_HEAD_PTR_AT_TOP 该字段表示RX FIFO写头指针处于RAM的顶部（存储单元127）。	RO	0b
0	RX_FIFO_WR_HEAD_PTR_WRAP 每当RX FIFO写头指针在RAM的顶部（存储单元127）绕回时，该位都会翻转。	R/W	0b

11.4.2.21 RX FIFO 写头指针加1 寄存器 (RX_FIFO_WR_HEAD_PTR_PLUS_ONE)

地址：1A56h 大小：8 位

该寄存器包含RX FIFO写头指针加1。该指针指示下一个接收到的数据包的起始位置加1。该寄存器仅用于调试和测试，通常不应被写入。

软件必须保持该寄存器与其他写指针寄存器之间的一致性。

不存在硬件保持寄存器来保持此寄存器与其他写指针寄存器之间的写一致性。假设该寄存器只在无有效接收的情况下被写入。

Bit	说明	类型	默认值
7	保留	RO	-
6:0	RX_FIFO_WR_HEAD_PTR_PLUS_ONE 该字段包含RX FIFO写头指针加1。	R/W	0000001b

11.4.2.22 RX FIFO 写头指针加1 控制位寄存器 (RX_FIFO_WR_HEAD_PTR_PLUS_ONE_CTL_BITS)

地址：1A57h 大小：8 位

该寄存器包含RX FIFO写头指针加1的辅助控制位。该寄存器仅用于调试和测试，通常不应被写入。

软件必须保持该寄存器与其他写指针寄存器之间的一致性。

不存在硬件保持寄存器来保持此寄存器与其他写指针寄存器之间的写一致性。假设该寄存器只在无有效接收的情况下被写入。

Bit	说明	类型	默认值
7:1	保留	RO	-
0	RX_FIFO_WR_HEAD_PTR_PLUS_ONE_WRAP 每当RX FIFO写头指针加1在RAM的顶部（存储单元127）绕回时，该位都会翻转。	R/W	0b

11.4.3 PD MAC中断状态和允许寄存器

以下小节介绍与PD MAC中断相关的各种寄存器。

有关系统级中断控制的更多信息，请参见第35页的第8.1节“系统中断”。

11.4.3.1 MAC中断状态寄存器（MAC_IRQ_STAT）

地址：1A80h 大小：8位

该寄存器反映了各种中断源的汇总状态。如果其中的任一位置1，则中断状态寄存器（INT_STS）的MAC_INT中断位将有效（如果允许）。

该寄存器中的位是只读的，反映了其他PD MAC中断状态寄存器的组合结果。

不存在相应的中断允许寄存器，因为所有中断均在较低层级被允许。

Bit	说明	类型	默认值
7:5	保留	RO	-
4	RX_SOP_IRQ_STAT RX SOP状态。该位在RX SOP中断状态寄存器（RX_SOP_IRQ_STAT）的任一位置1时置1，并通过RX SOP中断允许寄存器（RX_SOP_IRQ_EN）允许。 该位只能通过清零或禁止RX SOP中断状态寄存器（RX_SOP_IRQ_STAT）的位来清零。	RO	0b
3	PM_IRQ_STAT 功耗管理状态。该位在功耗管理中中断状态寄存器（PM_IRQ_STAT）的任一位置1时置1，并通过功耗管理中中断允许和控制寄存器（PM_IRQ_EN）允许。 该位只能通过清零或禁止功耗管理中中断状态寄存器（PM_IRQ_STAT）的位来清零。	RO	0b
2	RX_ERR_IRQ_STAT RX错误状态。该位在RX错误中断状态寄存器（RX_ERR_IRQ_STAT）的任一位置1时置1，并通过RX错误中断允许寄存器（RX_ERR_IRQ_EN）允许。 该位只能通过清零或禁止RX错误中断状态寄存器（RX_ERR_IRQ_STAT）的位来清零。	RO	0b
1	RX_IRQ_STAT RX状态。该位在RX中断状态寄存器（RX_IRQ_STAT）的任一位置1时置1，并通过RX中断允许寄存器（RX_IRQ_EN）允许。 该位只能通过清零或禁止RX中断状态寄存器（RX_IRQ_STAT）的位来清零。	RO	0b
0	TX_IRQ_STAT TX状态。该位在TX中断状态寄存器（TX_IRQ_STAT）的任一位置1时置1，并通过TX中断允许寄存器（TX_IRQ_EN）允许。 该位只能通过清零或禁止TX中断状态寄存器（TX_IRQ_STAT）的位来清零。	RO	0b

11.4.3.2 TX中断状态寄存器 (TX_IRQ_STAT)

地址: 1A81h 大小: 8位

该寄存器反映了各种TX_IRQ_STAT源的状态。是否产生中断取决于TX中断允许寄存器 (TX_IRQ_EN) 的值。
该寄存器中的位由硬件置1，且必须由软件清零。

Bit	说明	类型	默认值
7	AUTO_RSP_ABORTED TX自动响应报文被中止。TX被中止的原因可通过读取TX自动响应中止状态寄存器 (TX_AR_ABORT_STAT) 确定。	R/W1C	0b
6	AUTO_RSP_SENT 硬件已发送自动响应。硬件发送完自动响应时，该位置1。	R/W1C	0b
5	OK_TO_TX 现在可以发送。TX控制寄存器B (TX_CTL_B) 的OK_TO_TX位从0变为1时该位置1，表示软件可启动发送。	R/W1C	1b
4	CRC_RCV_TIMEOUT CRC接收定时器已过期。这是TX故障。只有当禁止自动重试模式时，才应允许该中断。 RX控制寄存器A (RX_CTL_A) 的EN_CRC_RCV_TMR位必须置1以使能定时器供软件使用。 注： CRC接收定时器还用于BIST模式，以对BIST错误计数报文的接收进行定时。	R/W1C	0b
3	TX_EOP 硬件完成数据包发送。请注意，这并不表示发送了有效EOP成帧符号。这只表示数据包发送结束，即，发送器已关闭。 每次进行TX尝试后该位都会置1。 此位可用于在硬复位或线缆复位完成发送时发出信号，由于其不具备GoodCRC响应，因此TX_DONE位不会置1，或者，如果禁止自动重试模式，在这种情况下硬件完成操作，则发送器将关闭。 该位由软件用来触发内部定时器。软件在允许/禁止该中断时必须谨慎，因为在每次TX完成后该位都会置1。	R/W1C	0b
2	TX_ABORTED TX报文被中止。TX被中止的原因可通过读取TX中止状态寄存器 (TX_ABORT_STAT) 确定。	R/W1C	0b

Bit	说明	类型	默认值
1	TX_FAILED 仅适用于自动重试模式（或使能 GoodCRC 定时器的情况）。 该中断指示 TX 报文请求因以下原因之一而无法完成： • GoodCRC 定时器超时 • 超出最大重试次数 注： 软件可在不使能自动重试的情况下使能 GoodCRC 定时器。在这种情况下，GoodCRC 定时器超时时将产生此中断。由软件负责在接收到 GoodCRC 报文时禁止 GoodCRC 定时器，以阻止产生此中断。	R/W1C	0b
0	TX_DONE 仅适用于自动重试模式。TX 报文请求成功完成，即，报文已由接收方通过 GoodCRC 进行应答。 对于未获得 GoodCRC 报文的报文（即，硬复位或线缆复位），该位未置 1。在这种情况下，应使用 TX_EOP。 禁止自动重试模式时，应使用 TX_EOP 来确定数据包何时完成发送。	R/W1C	0b

11.4.3.3 RX 中断状态寄存器（RX_IRQ_STAT）

地址：1A82h 大小：8 位

该寄存器反映了各种 RX_IRQ_STAT 源的状态。是否产生中断取决于 RX 中断允许寄存器（RX_IRQ_EN）的值。
该寄存器中的位由硬件置 1，且必须由软件清零。

Bit	说明	类型	默认值
7	RX_FIFO_NOT_EMPTY RX FIFO 非空。只要 RX FIFO 中存在待读取的数据，该位就会置 1。通常，该位在接收完有效数据包后置 1，在所有数据均被读取后清零。 注： 该位是只读的，将在 FIFO 为空时清零。	RO	0b
6	LINE_WENT_IDLE 线路现在空闲。硬件检测到 PD 总线上无活动时，该位置 1。 注： 该位不是线路状态，而是一个事件。如果在允许该中断时线路已空闲，该位将立即置 1。如果在允许该中断时线路繁忙，该位将在线路空闲时置 1。 软件可通过此位在 TX 结束或 RX 结束时收到通知。 这仅适用于使能时钟的情况。	R/W1C	1b
5	RX_CABLE_RST 接收到线缆复位报文。	R/W1C	0b
4	保留	RO	-
3	RX_HARD_RST 接收到硬复位报文。	R/W1C	0b

UPD360

Bit	说明	类型	默认值
2	RX_EOP 硬件检测到有效EOP成帧。不良EOP或异常终止将导致数据包被丢弃。 该位置1的情况仅适用于以通过RX控制寄存器B（RX_CTL_B）使能的SOP类型开始的帧。	R/W1C	0b
1	RX_SOP 硬件检测到有效SOP成帧序列。 该位置1的情况适用于通过RX控制寄存器B（RX_CTL_B）使能的SOP类型。	R/W1C	0b
0	RX_DONE RX报文已成功接收。 以下条件适用： - 在自动重试模式下，预期GoodCRC报文被丢弃，因此不进行指示。 - 在自动响应模式下，重复数据包被丢弃，因此不进行指示 注： 在自动响应模式下，该位在发送GoodCRC响应之前置1，而无论发送是否成功。	R/W1C	0b

11.4.3.4 RX错误中断状态寄存器（RX_ERR_IRQ_STAT）

地址：1A83h 大小：8位

该寄存器反映了各种RX_ERR_IRQ_STAT源的状态。是否产生中断取决于RX错误中断允许寄存器（RX_ERR_IRQ_EN）的值。
该寄存器中的位由硬件置1，且必须由软件清零。

Bit	说明	类型	默认值
7	DBG_EVENT 发生了调试事件： - 具有不良CRC的数据包数达到限值。提供了链路质量的指示。 - 重复数据包计数达到限值。提供了接收器丢弃的GoodCRC数据包的指示。 注： 软件必须读取两个计数寄存器才能确定上述两个条件中的哪个条件导致了此事件。 注： 与选通脉冲触发的其他RX_IRQ状态位不同，该位具有持久性，即，如果引起中断的寄存器在清除中断之前未清零，则该位将再次置1。	R/W1C	0b
6:4	保留	RO	-
3	RX_BAD_PKT 接收到损坏的数据包（不良CRC、符号错误、过大或奇数个半字节计数）。 RX数据包状态寄存器（RX_PKT_STAT）的RX_PKT_STAT[0]（BAD_CRC）、RX_PKT_STAT[5]（SYM_ERROR）、RX_PKT_STAT[6]（ODD_NIBBLES）或RX_PKT_STAT[3]（RX_OVER_SIZE）将置1。	R/W1C	0b
2	RX_BUF_OVR_RUN 正在接收数据包，但不存在可用缓冲区空间。在这种情况下，RX数据包状态寄存器（RX_PKT_STAT）的RX_PKT_STAT[4]（RX_FIFO_FULL）位将置1。 接收到的数据包将被丢弃。	R/W1C	0b
1	RX_PCOL_ERROR 仅适用于自动重试模式。硬件检测到协议错误，即，在预期接收到GoodCRC报文时，接收到除GoodCRC或Ping报文之外的报文。 接收到的报文将置于RXQ中，RX_DONE将置1。 自动重试将被中止，TX_ABORTED位将置1。	R/W1C	0b
0	RX_PKT_DROPPED RX数据包因以下原因而被丢弃： - 在自动响应模式下存在重复数据包 - 在自动重试模式下存在GoodCRC报文	R/W1C	0b

11.4.3.5 功耗管理中断状态寄存器 (PM_IRQ_STAT)

地址：1A84h 大小：8位

该寄存器反映了各种PM_IRQ_STAT源的状态。是否产生中断取决于功耗管理中断允许和控制寄存器 (PM_IRQ_EN) 的值。

该寄存器中的位由硬件置1，且必须由软件清零。

Bit	说明	类型	默认值
7:1	保留	RO	-
0	UPD_ACT_WS 检测到USB PD活动。在开始任何活动时，均通过将接收输入信号置为有效来进行异步置1。 为防止出现亚稳定性，只有当软件计划停止时钟时，才应允许该中断。 与其他中断状态位不同，将USB_PD_ACT_WS_EN位清零实际上会阻止该位被置1。 该位不受UPD_CLK_STOP_EN和UPD_WU_EN的影响。 在正常工作期间，USB PD活动状态在其他寄存器位中可用。	R/W1C	0b

11.4.3.6 RX SOP中断状态寄存器 (RX_SOP_IRQ_STAT)

地址：1A85h 大小：8位

该寄存器反映了接收到的各种SOP类型的发生状态。SOP类型无需通过RX控制寄存器B (RX_CTL_B) 使能来将状态位置1。是否产生中断取决于RX SOP中断允许寄存器 (RX_SOP_IRQ_EN) 的值。

该寄存器的位不会级联到RX中断状态寄存器中，但会被使能并与自身的中断信号输出进行逻辑或运算。

该寄存器中的位由硬件置1，且必须由软件清零。

Bit	说明	类型	默认值
7:5	保留	RO	-
4:0	RX_SOP 硬件检测到有效SOP成帧序列。 Bit 0: SOP Bit 1: SOP' Bit 2: SOP" Bit 3: SOP'_Debug Bit 4: SOP''_Debug	RW1C	00000b

11.4.3.7 TX中断允许寄存器 (TX_IRQ_EN)

地址: 1A86h 大小: 8位

将该寄存器的某个位置1将使相应的源产生中断。请注意，只会影响中断产生。TX中断状态寄存器 (TX_IRQ_STAT) 仍反映事件的状态。

在POR后所有中断均被禁止。

Bit	说明	类型	默认值
7	AUTO_RSP_ABORTED_EN 将该位置1可允许AUTO_RSP_ABORTED中断。	R/W	0b
6	AUTO_RSP_SENT_EN 将该位置1可允许AUTO_RSP_SENT中断。	R/W	0b
5	OK_TO_TX_EN 将该位置1可允许OK_TO_TX中断。	R/W	0b
4	CRC_RCV_TIMEOUT_EN 将该位置1可允许CRC_RCV_TIMEOUT中断。	R/W	0b
3	TX_EOP_EN 将该位置1可允许TX_EOP中断。	R/W	0b
2	TX_ABORTED_EN 将该位置1可允许TX_ABORTED中断。	R/W	0b
1	TX_FAILED_EN 将该位置1可允许TX_FAILED中断。	R/W	0b
0	TX_DONE_EN 将该位置1可允许TX_DONE中断。	R/W	0b

11.4.3.8 RX中断允许寄存器 (RX_IRQ_EN)

地址: 1A87h 大小: 8位

将该寄存器的某个位置1将使相应的源产生中断。请注意，只会影响中断产生。RX中断状态寄存器 (RX_IRQ_STAT) 仍反映事件的状态。

在POR后所有中断均被禁止。

Bit	说明	类型	默认值
7	RX_FIFO_NOT_EMPTY_EN 将该位置1可允许RX_FIFO_NOT_EMPTY中断。	R/W	0b
6	LINE_WENT_IDLE_EN 将该位置1可允许LINE_WENT_IDLE中断。	R/W	0b
5	RX_CABLE_RST_EN 将该位置1可允许RX_CABLE_RST中断。	R/W	0b
4	保留	RO	-
3	RX_HARD_RST_EN 将该位置1可允许RX_HARD_RST中断。	R/W	0b
2	RX_EOP_EN 将该位置1可允许RX_EOP中断。	R/W	0b
1	RX_SOP_EN 将该位置1可允许RX_SOP中断。	R/W	0b
0	RX_DONE_EN 将该位置1可允许RX_DONE中断。	R/W	0b

11.4.3.9 RX错误中断允许寄存器（RX_ERR_IRQ_EN）

地址：1A88h 大小：8位

将该寄存器的某个位置1将使相应的源产生中断。请注意，只会影响中断产生。[RX错误中断状态寄存器（RX_ERR_IRQ_STAT）](#)仍反映事件的状态。

在POR后所有中断均被禁止。

Bit	说明	类型	默认值
7	DBG_EVENT_EN 将该位置1可允许 DBG_EVENT 中断。	R/W	0b
6:4	保留	RO	-
3	RX_BAD_PKT_EN 将该位置1可允许 RX_BAD_PKT 中断。	R/W	0b
2	RX_BUF_OVR_RUN_EN 将该位置1可允许 RX_BUF_OVR_RUN 中断。	R/W	0b
1	RX_PCOL_ERROR_EN 将该位置1可允许 RX_PCOL_ERROR 中断。	R/W	0b
0	RX_PKT_DROPPED_EN 将该位置1可允许 RX_PKT_DROPPED 中断。	R/W	0b

11.4.3.10 功耗管理中断允许和控制寄存器（PM_IRQ_EN）

地址：1A89h 大小：8位

将该寄存器的某个位置1将使相应的源产生中断。请注意，只会影响中断产生。[功耗管理中断状态寄存器（PM_IRQ_STAT）](#)仍反映事件的状态。

在POR后所有中断均被禁止。该寄存器还会使能时钟启动和停止请求。

Bit	说明	类型	默认值
7:4	保留	RO	-
3	UPD_CLK_STOP_EN 如果该位置1，则MAC会在空闲时请求系统时钟停止。 如果该位清零，则软件负责将器件返回到休眠模式。 该位不影响唤醒中断（ UPD_ACT_WS ）产生。 注： 该位不影响需要时钟的MAC指示。即使该位清零，MAC在空闲后也会指示无需时钟。	R/W	0b

Bit	说明	类型	默认值
2	UPD_WU_EN 如果该位置1，则MAC会在检测到接收活动时请求系统时钟启动。 该位不影响唤醒中断（ UPD_ACT_WS ）产生。 注： 时钟启动后，该位不影响需要时钟的MAC指示。即使该位清零，MAC在非空闲（即，静噪激活）状态下也会指示需要时钟。只会影响异步唤醒和静噪启动之间的时间周期。	R/W	0b
1	保留	RO	-
0	USB_PD_ACT_WS_EN 将该位置1可允许 UPD_ACT_WS 中断。	R/W	0b

11.4.3.11 RX SOP中断允许寄存器（RX_SOP_IRQ_EN）

地址：[1A8Ah](#) 大小：8位

将该寄存器的某个位置1将使相应的源产生中断。请注意，只会影响中断产生。[RX SOP中断状态寄存器（RX_SOP_IRQ_STAT）](#)仍反映事件的状态。

在POR后所有中断均被禁止。

Bit	说明	类型	默认值
7:5	保留	RO	-
4:0	RX_SOP_EN 将某个位置1可允许相应的 RX_SOP 中断。 Bit 0: SOP Bit 1: SOP' Bit 2: SOP" Bit 3: SOP' _Debug Bit 4: SOP" _Debug	R/W	00000b

11.4.3.12 复位控制寄存器（RESET_CTL）

地址：1A8Bh 大小：8位

Bit	说明	类型	默认值
7	SW_RESET 软件生成的PD MAC复位。软件可使用此位将PD MAC复位为POR状态。 软件应对此位进行翻转，即置1和清零。该位清零后，软件将必须重新初始化PD MAC，其中包括BMC寄存器。 此位通过RESET_N复位。	R/W	0b
6	BMC_SW_RESET 软件生成的BMC复位。软件可使用此位将BMC编码器/解码器复位为POR状态。 软件应对此位进行翻转，即置1和清零。BMC寄存器不受此复位影响。	R/W	0b
5:1	保留	RO	-
0	PD_RESET 处理完PD硬复位、线缆复位或软复位后，软件应将此位置1，以便硬件可返回至PD复位状态。软件应对此位进行翻转，即置1和清零。 以下部分通过PD_RESET复位： • RX_CTL FSM进入POR状态，例外是硬复位和线缆复位的处理保持使能状态（如果之前已通过使能接收器进行使能）。 • TX_CTL FSM进入POR状态。 • TX并串转换和4b5b编码器 • TX控制寄存器B（TX_CTL_B）GO、TX_HARD_RESET和TX_CABLE_RESET位 • TX参数寄存器A（TX_PARAM_A）EN_FWTX位 以下部分不复位： • 除上面列出的寄存器外的所有PD MAC寄存器。 • 各种TX数据包计数器均不复位（NUM_PD_PACKETS_SENT、NUM_MISSING_GCRC_MSG和NUM_MISSING_GCRC_MSG_SENT）。 • 各种RX数据包计数器均不复位（RX_BADCRC_PKT_CNT、RX_DUP_PKT_CNT和NUM_PD_PACKETS_RECVD）。 • RX FIFO不复位。 • RX CDR、4b5b解码器和CRC不复位。 • TX IFG定时器和CRC不复位。 • BIST和功耗管理功能不复位。 • BMC编码器/解码器。	R/W	0b

11.4.4 PD MAC BMC编码器/解码器寄存器

以下小节介绍与PD MAC BMC编码器/解码器相关的各种寄存器。

11.4.4.1 BMC RX高电平的最大完整位时间寄存器（BMC_RX_HI_FB_MAX_TIME）

地址：1AA0h 大小：8位

该寄存器的值与BMC RX高电平的最小完整位时间寄存器（BMC_RX_HI_FB_MIN_TIME）的值共同由BMC解码器用于确定高电平RX信号的有效完整位时间。该时间应考虑到接收信号的比特率范围和占空比失真。

BMC_RX_HI_FB_MAX_TIME > 高电平的有效时间 > BMC_RX_HI_FB_MIN_TIME

Bit	说明	类型	默认值
7:0	BMC_RX_HI_FB_MAX_TIME 最小比特率和最差占空比失真条件下的时间，以时钟周期为单位。 $= (((\text{clock_freq_kHz} / \text{min_bit_rate_kbps}) * (100 + \text{容差}) / 100) - 1)$ 注： 建议该字段采用值202（十进制）。	R/W	00h

11.4.4.2 BMC RX高电平的最小完整位时间寄存器（BMC_RX_HI_FB_MIN_TIME）

地址：1AA1h 大小：8位

该寄存器的值与BMC RX高电平的最大完整位时间寄存器（BMC_RX_HI_FB_MAX_TIME）的值共同由BMC解码器用于确定高电平RX信号的有效完整位时间。该时间应考虑到接收信号的比特率范围和占空比失真。

BMC_RX_HI_FB_MAX_TIME > 高电平的有效时间 > BMC_RX_HI_FB_MIN_TIME

BMC_RX_HI_FB_MIN_TIME > BMC_RX_HI_FB_MAX_TIME

Bit	说明	类型	默认值
7:0	BMC_RX_HI_FB_MIN_TIME 最大比特率和最差占空比失真条件下的时间，以时钟周期为单位。 $= (((\text{clock_freq_kHz} / \text{max_bit_rate_kbps}) * (100 - \text{容差}) / 100) - 1)$ 注： 建议该字段采用值116（十进制）。	R/W	00h

11.4.4.3 BMC RX低电平的最大完整位时间寄存器（BMC_RX_LO_FB_MAX_TIME）

地址：1AA2h 大小：8位

该寄存器的值与BMC RX低电平的最小完整位时间寄存器（BMC_RX_LO_FB_MIN_TIME）的值共同由BMC解码器用于确定低电平RX信号的有效完整位时间。该时间应考虑到接收信号的比特率范围和占空比失真。

BMC_RX_LO_FB_MAX_TIME > 低电平的有效时间 > BMC_RX_LO_FB_MIN_TIME

Bit	说明	类型	默认值
7:0	BMC_RX_LO_FB_MAX_TIME 最小比特率和最差占空比失真条件下的时间，以时钟周期为单位。 $= (((\text{clock_freq_kHz} / \text{min_bit_rate_kbps}) * (100 + \text{容差}) / 100) - 1)$ 注： 建议该字段采用值202（十进制）。	R/W	00h

11.4.4.4 BMC RX低电平的最小完整位时间寄存器（BMC_RX_LO_FB_MIN_TIME）

地址：1AA3h 大小：8位

该寄存器的值与BMC RX低电平的最大完整位时间寄存器（BMC_RX_LO_FB_MAX_TIME）的值共同由BMC解码器用于确定低电平RX信号的有效完整位时间。该时间应考虑到接收信号的比特率范围和占空比失真。

BMC_RX_LO_FB_MAX_TIME > 低电平的有效时间 > BMC_RX_LO_FB_MIN_TIME

BMC_RX_LO_FB_MIN_TIME > BMC_RX_LO_HB_MAX_TIME

Bit	说明	类型	默认值
7:0	BMC_RX_LO_FB_MIN_TIME 最大比特率和最差占空比失真条件下的时间，以时钟周期为单位。 $= (((\text{clock_freq_kHz} / \text{max_bit_rate_kbps}) * (100 - \text{容差}) / 100) - 1)$ 注： 建议该字段采用值116（十进制）。	R/W	00h

11.4.4.5 BMC RX高电平的最大半位时间寄存器（BMC_RX_HI_HB_MAX_TIME）

地址：1AA4h 大小：8位

该寄存器的值与BMC RX高电平的最小半位时间寄存器（BMC_RX_HI_HB_MIN_TIME）的值共同由BMC解码器用于确定高电平RX信号的有效半位时间。该时间应考虑到接收信号的比特率范围和占空比失真。

BMC_RX_HI_HB_MAX_TIME > 高电平的有效时间 > BMC_RX_HI_HB_MIN_TIME

Bit	说明	类型	默认值
7:0	BMC_RX_HI_HB_MAX_TIME 最小比特率和最差占空比失真条件下的时间，以时钟周期为单位。 $= ((1 / 2) * ((\text{clock_freq_kHz} / \text{min_bit_rate_kbps}) * (100 + \text{容差}) / 100) - 1)$ 注： 建议该字段采用值110（十进制）。	R/W	00h

11.4.4.6 BMC RX高电平的最小半位时间寄存器（BMC_RX_HI_HB_MIN_TIME）

地址：1AA5h 大小：8位

该寄存器的值与BMC RX高电平的最大半位时间寄存器（BMC_RX_HI_HB_MAX_TIME）的值共同由BMC解码器用于确定高电平RX信号的有效半位时间。该时间应考虑到接收信号的比特率范围和占空比失真。

BMC_RX_HI_HB_MAX_TIME > 高电平的有效时间 > BMC_RX_HI_HB_MIN_TIME

BMC_RX_HI_FB_MIN_TIME > BMC_RX_HI_HB_MAX_TIME

Bit	说明	类型	默认值
7:0	BMC_RX_HI_HB_MIN_TIME 最大比特率和最差占空比失真条件下的时间，以时钟周期为单位。 $= ((1 / 2) * ((\text{clock_freq_kHz} / \text{max_bit_rate_kbps}) * (100 - \text{容差}) / 100) - 1)$ 注： 建议该字段采用值48（十进制）。	R/W	00h

11.4.4.7 BMC RX低电平的最大半位时间寄存器（BMC_RX_LO_HB_MAX_TIME）

地址：1AA6h 大小：8位

该寄存器的值与BMC RX低电平的最小半位时间寄存器（BMC_RX_LO_HB_MIN_TIME）的值共同由BMC解码器用于确定低电平RX信号的有效半位时间。该时间应考虑到接收信号的比特率范围和占空比失真。

BMC_RX_LO_HB_MAX_TIME > 低电平的有效时间 > BMC_RX_LO_HB_MIN_TIME

Bit	说明	类型	默认值
7:0	BMC_RX_LO_HB_MAX_TIME 最小比特率和最差占空比失真条件下的时间，以时钟周期为单位。 $= ((1 / 2) * ((\text{clock_freq_kHz} / \text{min_bit_rate_kbps}) * (100 + \text{容差}) / 100) - 1)$ 注： 建议该字段采用值110（十进制）。	R/W	00h

11.4.4.8 BMC RX低电平的最小半位时间寄存器（BMC_RX_LO_HB_MIN_TIME）

地址：1AA7h 大小：8位

该寄存器的值与BMC RX低电平的最大半位时间寄存器（BMC_RX_LO_HB_MAX_TIME）的值共同由BMC解码器用于确定低电平RX信号的有效半位时间。该时间应考虑到接收信号的比特率范围和占空比失真。

BMC_RX_LO_HB_MAX_TIME > 低电平的有效时间 > BMC_RX_LO_HB_MIN_TIME

BMC_RX_LO_FB_MIN_TIME > BMC_RX_LO_HB_MAX_TIME

Bit	说明	类型	默认值
7:0	BMC_RX_LO_HB_MIN_TIME 最大比特率和最差占空比失真条件下的时间，以时钟周期为单位。 $= ((1 / 2) * ((\text{clock_freq_kHz} / \text{max_bit_rate_kbps}) * (100 - \text{容差}) / 100) - 1)$ 注： 建议该字段采用值48（十进制）。	R/W	00h

11.4.4.9 BMC RX 静噪置为有效时间寄存器（BMC_RX_SQL_ASSERT_TIME）

地址：1AA8h 大小：8 位

BMC 解码器使用该寄存器的值来确定何时存在有效信号。如果在该时间内检测到接收信号的三个边沿，则“静噪”将激活。该时间应考虑到最小比特率，但无需考虑到接收信号的占空比失真。

Bit	说明	类型	默认值
7:0	BMC_RX_SQL_ASSERT_TIME 最小比特率时的时间（以时钟周期为单位）x 4。 置为有效的时间 $\mu s = (BMC_RX_SQL_ASSERT_TIME \times 4 + 1) / clock_freq \text{ MHz}$ 典型的置为有效时间为3个最大半位时间加上所需容差。 注： 建议该字段采用值70（十进制）。	R/W	00h

11.4.4.10 BMC RX 静噪保持时间寄存器（BMC_RX_SQL_HOLD_TIME）

地址：1AA9h 大小：8 位

BMC 解码器使用该寄存器的值为CDR电路提供时间以输出最后一位数据。典型的保持时间为1个最大半位时间加上所需容差。

注： 静噪下降时间基于BMC RX最大高电平和低电平时间寄存器。

Bit	说明	类型	默认值
7:0	BMC_RX_SQL_HOLD_TIME 最小比特率时的时间（以时钟周期为单位）x 4。 保持时间 $\mu s = (BMC_RX_SQL_HOLD_TIME \times 4 + 1) / clock_freq \text{ MHz}$ 典型值 = $((1 / 2) * ((clock_freq_kHz / min_bit_rate_kbps) * (100 + 容差) / 100) - 1) / 4$ 注： 建议该字段采用值23（十进制）。	R/W	00h

11.4.4.11 BMC TX位时间计数寄存器（BMC_TX_BITTIME_CNT）

地址：1AB0h 大小：8位

Bit	说明	类型	默认值
7:0	BMC_TX_BIT_TIME_CNT 位时间计数器值。BMC编码器使用此值确定最后一位数据的位时序。可使用该寄存器来调整位时间以进行测试（调整比特率）。 计数值基于时钟频率，可通过下式得出： $((\text{时钟频率 kHz} / \text{比特率 kbps}) - 1)$ 即， 时钟频率 = 48000 kHz（48 MHz） 标称比特率 = 300 kbps $\text{bit_time_cnt} = (48000 / 300) - 1 = 159$ 注： 建议该字段采用值159（十进制）。	R/W	00h

11.4.4.12 BMC转换窗口时间寄存器（BMC_TRANSITION_WINDOW_TIME）

地址：1AB1h 大小：8位

BMC解码器使用该寄存器的值来确定线路何时空闲。接收器在一个时间窗口BMC_TRANSITION_WINDOW_TIME内发生3次转换时，检测有效。如果在等待BMC_TRANSITION_WINDOW_TIME时未检测到3次转换，则等待此时间后，将认为总线空闲。USB PD规范要求最短时间为12 μs，最长时间为20 μs。

Bit	说明	类型	默认值
7:0	BMC_TRANSITION_WINDOW_TIME 时间（以时钟周期为单位）x 4 转换时间 $\mu\text{s} = (\text{BMC_TRANSITION_WINDOW_TIME} \times 4) / \text{clock_freq MHz}$ 根据USB PD规范，应设置此值在12 μs和20 μs之间。在48 MHz下，该值应介于144和240之间。 注： 与TX周转时间寄存器（TX_TA_TIME）不同，该硬件不使用预分频器，因此指定的时间准确。 注： 建议该字段采用值200（十进制）。	R/W	00h

11.4.5 PD MAC BIST 寄存器

以下小节介绍与PD MAC内置自检（Built-In Self Test, BIST）相关的各种寄存器。

11.4.5.1 BIST控制寄存器A（BIST_CTL_A）

地址：1AC0h 大小：8位

Bit	说明	类型	默认值
7:5	保留	RO	-
4	BIST_RX_EN 使能BIST接收器。 将该位置1将： • 使能BIST接收器。 • 复位BIST RX逻辑中的PRBS发生器，使其准备好比较传入的BIST PRBS模式。 • 复位BIST错误计数寄存器（BIST错误计数高字节寄存器（BIST_ERR_CNT_HI）和BIST错误计数低字节寄存器（BIST_ERR_CNT_LO））。 通常，该位只有在需要BIST接收器模式时才置1；但是，可将该位置1以使能环回BIST测试，方法是禁止自动响应模式并使能RX环回（将 RX控制寄存器A（RX_CTL_A） 的 EN_LOOPBACK 位置1）。 该位必须清零，并在每次有新BIST RX请求传入时由软件置1。 注： 在BIST环回测试模式下，必须为每个TX数据包手动触发BIST TX，因为功能接收器路径被禁止。	R/W	0b
3	BIST_EN 使能BIST。BIST逻辑激活时将该位置1。 该位与 RX控制寄存器A（RX_CTL_A） 的 EN_LOOPBACK 位共同由硬件用于阻止接收到的数据进入功能路径。即，在BIST模式下使能环回时，硬件无法检测到硬复位。 该位还用于对BMC发送器上电以便在BIST模式下工作。软件应将该位置1，以便在通过 BIST_TX_START/BIST_TX_STATUS 触发BIST帧的发送之前有充足的时间（5 μs）。	R/W	0b
2:0	BIST_TX_MODE 设置BIST测试模式。仅支持以下模式： 010b：BIST载波测试模式2（交替的1和0） 所有其他模式均保留，不应使用。	R/W	000b

11.4.5.2 BIST控制寄存器B (BIST_CTL_B)

地址: 1AC1h 大小: 8位

Bit	说明	类型	默认值
7:3	保留	RO	-
2	BIST_CLR_ERR_CNT 将BIST错误计数寄存器 (BIST错误计数高字节寄存器 (BIST_ERR_CNT_HI) 和BIST错误计数低字节寄存器 (BIST_ERR_CNT_LO)) 清零。	WO	0b
1	BIST_TX_RST 复位TX逻辑和PRBS发生器。软件必须在首次开始BIST测试 (即, BIST_TX_START/BIST_TX_STATUS位置1) 之前将此位置1。 由于该位会复位PRBS发生器, 因此在当前BIST测试结束之前不应再次置1。 该位不被锁存, 始终读为0 PRBS发生器不能通过BIST_TX_START/BIST_TX_STATUS复位, 因为这会针对每个TX BIST帧复位此发生器。 应使用该位停止连续的BIST TX模式 (2)。Tx可能需要花费1个位时间来停止。之后BIST_EN才能清零。 注: 如果此位与 BIST_TX_START/BIST_TX_STATUS 同时置1, 则该位具有较高优先级, 并会使BIST_TX_START/BIST_TX_STATUS无效。	WO	0b
0	BIST_TX_START/BIST_TX_STATUS 开始BIST发送。将该位置1将触发BIST帧的发送。 该位未被锁存, 读取此位时将返回当前BIST发送的状态。 根据BIST模式, 发送将无限次继续 (除非BIST_TX_RST置为有效) 或在发送BIST PRBS帧后停止。 BIST不支持自动发送下一个BIST帧。必须由软件通过先将此位清零然后再重新置1来触发。 注: 该位不能与BIST_TX_RST同时置1。BIST_TX_RST具有较高优先级, 将使BIST_TX_START/BIST_TX_STATUS失效。 在BIST RX模式下, 如果TX控制寄存器A (TX_CTL_A) 的EN_AUTO_RSP_MODE位置1, 则硬件会自动发送BIST错误计数报文。	WO/RO	0b

UPD360

11.4.5.3 BIST 错误计数高字节寄存器 (BIST_ERR_CNT_HI)

地址: 1AC2h 大小: 8位

Bit	说明	类型	默认值
7:0	BIST_ERR_CNT_HI[7:0] BIST 错误计数的高字节 (bit 15-8) (BIST_ERR_CNT[15:8])	RO	00h

11.4.5.4 BIST 错误计数低字节寄存器 (BIST_ERR_CNT_LO)

地址: 1AC3h 大小: 8位

Bit	说明	类型	默认值
7:0	BIST_ERR_CNT_LO[7:0] BIST 错误计数的低字节 (bit 7-0) (BIST_ERR_CNT[7:0])	RO	00h

11.4.5.5 BIST RX 状态寄存器 (BIST_RX_STAT)

地址: 1AC7h 大小: 8位

该寄存器用于帮助进行功能测试。将硬件配置为发送 BIST PRBS 帧。同时，使能环回模式以及使能 BIST 接收模式。现在，发送 PRBS 帧时，也会对其进行接收和比较。该寄存器提供有关 BIST PRBS 帧何时完成接收以及是否检测到错误的信息。

Bit	说明	类型	默认值
7:2	保留	RO	-
1	BIST_RX_ERROR 该位仅在 BIST_RX_DONE 位置 1 时有效。 值 0 表示 BIST PRBS 帧已被接收且无错误。 值 1 表示 BIST PRBS 帧已被接收但有错误。	RO	0b
0	BIST_RX_DONE 成功接收到 BIST RX 帧。 向该位写入 1 可将其清零。	R/W1C	0b

12.0 USB 端口电源控制器（PPC）

该器件的USB 端口电源控制器（PPC）可在VBUS上提供3.0A的连续电流（最高3.2A），并且具有高精度过流限制（Over-Current Limiting, OCL）、欠压和过压锁定以及动态温度管理功能。PPC还包含针对BC 1.2应用的VBUS放电功能和恒流模式电流限制功能。

注： 第209页的表16-6“端口电源控制器直流参数”和第212页的表16-12“端口电源控制器交流参数”分别列出了PPC DC和AC参数。

12.1 PPC 功耗状态

PPC提供以下功耗状态：

- **休眠：**它是可用的最低PPC功耗状态。更多信息，请参见第12.1.1节“休眠状态操作”。
- **错误：**当存在故障条件时，会进入该状态。PPC中断状态寄存器对错误条件进行了详细说明。更多信息，请参见第12.1.3节“错误状态操作”。
- **活动：**在该状态下，PPC处于活动状态且向VBUS提供电流。不存在任何错误。该功耗状态提供完整功能。在该状态下，操作包括激活端口电源开关和电流限制。更多信息，请参见第12.1.2节“活动状态操作”。

注： PPC功耗状态反映在PPC一般配置3寄存器（PPC_GENERAL_CFG3）的PWR_STATE[1:0]字段中。

12.1.1 休眠状态操作

可使用PWR_EN引脚（仅限工作在独立DFP模式下的UPD360-A/UPD360-B）使PPC进入/退出休眠模式。当PPC处于休眠状态时，器件将处于其最低功耗状态。旁路开关和端口电源开关将被禁止。VBUS将接近地电位。

12.1.2 活动状态操作

当PPC进入活动状态时，端口电源开关会闭合。如果存在以下任一条件，则PPC无法处于活动状态（因此，端口电源开关无法闭合）：

- $V_S < V_{S_UVLO}$ （2.5V）。
- PWR_EN被禁止（仅限独立DFP模式）。

12.1.3 错误状态操作

在检测到以下任一事件时，PPC会从活动状态进入错误状态：

- 超出最大允许内部裸片温度 t_{SD} 。
- 检测到过流条件。
- 检测到VS上的欠压条件（见第12.2.1节“VS上的欠压锁定”）。
- 检测到放电错误。
- VS引脚上存在过压条件。

当PPC进入错误状态时，会禁止端口电源开关和VBUS旁路开关，ALERT会置为有效（默认）。在该功耗状态下，它们会保持断开。PPC将根据故障处理选择来退出该状态。

如果使用自动恢复故障处理程序，则PPC会检查所有错误条件是否都已消除。如果所有错误条件都已消除，PPC会恢复活动状态。在独立DFP模式下，ERR_RECOVER引脚用于选择自动恢复算法。

12.2 供电电压

12.2.1 VS上的欠压锁定

为了保持在活动功耗状态，PPC要求在VS引脚上存在最低电压 V_{S_UVLO} （2.5V）。

12.2.2 VS上的过压检测锁定

如果VS引脚电压超出电压 V_{S_OV} 的持续时间超过规定时间 t_{MASK} (5 ms)，端口电源开关会被禁止。这会导致器件进入错误状态。

12.2.3 PWR_EN输入 (仅限UPD360-A/UPD360-B)

在独立DFP模式下，如果满足条件，则PWR_EN引脚可使端口电源开关闭合。如果PWR_EN被禁止，则端口电源开关无法闭合。不过，如果PWR_EN使能，端口电源开关并不一定是闭合的（见第12.1.2节“活动状态操作”）。如果PIN_IGNORE配置位置1，则PPC会忽略PWR_EN引脚状态。否则，会随PWR_EN引脚一起检查PWR_EN_SET配置位。

12.3 USB端口电源开关

为了确保符合USB Type-C，PPC包含了支持跳变限流模式的USB端口电源开关。

限流值 (ILIM) 通过PWR_CAP0和PWR_CAP1引脚设置 (仅限UPD360-A/UPD360-B)，也可通过PPC限流寄存器 (PPC_CURRENT_LIMIT) 更新。该开关还包含了软启动电路和独立的短路限流功能。在活动状态下，端口电源开关会闭合 (VBUS在放电时除外)。

注： 如果消耗电流介于2 mA和7 mA之间的负载连接到端口电源开关，则将在VBUS输出上观察到40-90 mV_{pp}的电压纹波。连接便携式设备时，这种行为是正常的，不会影响充电过程。

12.3.1 限流

限流值 (ILIM) 通过PWR_CAP0和PWR_CAP1引脚设置 (仅限UPD360-A/UPD360-B)。这两个引脚在器件上电时读取。上电后，限流值可通过PPC限流寄存器 (PPC_CURRENT_LIMIT) 进行更改。但是，所设定的限流值不能超过硬件设置的限流值 (由PWR_CAP0和PWR_CAP1引脚定义)。

UPD360预配置有适当的设置，可满足USB Type-C限流要求，工作在独立模式下时无需寄存器写操作即可使用此内部5V端口电源控制器。

12.3.1.1 短路输出限流

当输出电流高于可选的限流值 (PPC限流寄存器 (PPC_CURRENT_LIMIT)) 时，会发生短路限流。器件会检测到该事件，并立即对电流进行限制 (在 t_{SHORT_LIM} 时间内)。如果该条件仍然存在，端口电源开关会标记错误条件，并进入错误状态。

12.3.1.2 软启动

当PWR_EN控制 (仅限UPD360-A/UPD360-B) 改变状态而使能端口电源开关时，PPC会调用软启动程序，其持续时间为VBUS上升时间 (t_{R_BUS})。在软启动程序工作时，它会限制从VS进入VBUS的电流。该电路可以防止由于便携式设备汲取电流阶跃变化而产生的电流尖峰。

当在PWR_EN引脚已使能 (独立DFP模式下) 或PWR_EN_SET位使能的情况下连接便携式设备时，如果总线电流超过ILIM，PPC电流限制器会在指定时间 (t_{SHORT_LIM}) 内做出响应，并在此时正常工作。CBUS电容会根据负载变化的需要输送额外的电流 (如果有)。

12.3.1.3 限流模式

PPC限流功能工作在跳变模式下。当端口电源开关闭合时，此模式激活。在检测功耗状态下工作时，VBUS电流最大值会被限制为 I_{BUS_BYP} 。

12.3.1.3.1 跳变模式

使用跳变限流模式时，PPC USB端口电源开关用作一个低电阻开关，如果超出限流值，它会快速断开。使用跳变限流模式工作时，对于最高为ILIM的所有电流值，VBUS输出电压将保持相对恒定 (等于VS电压减去 $R_{ON_PSW} * I_{BUS}$ 电流)。如果内部温度超过 t_{TSD} 阈值，则端口电源开关将断开。

如果便携式设备汲取的电流超过ILIM，会发生以下情况：

1. 端口电源开关断开（跳变模式操作）。
2. PPC将进入错误状态并将ALERT置为有效。
3. 然后，故障处理电路将决定后续的操作。

图12-1给出了跳变模式下的限流操作，阴影区域表示USB 2.0规定的VBUS范围。虚线表示当超过ILIM时，端口电源开关输出将变为0（即，跳变）。请注意，图12-1中给出的所有ILIM可能值的操作仅供参考；在实际操作中，任一时刻都只有一个ILIM可处于活动状态。

图12-1： 跳变限流操作

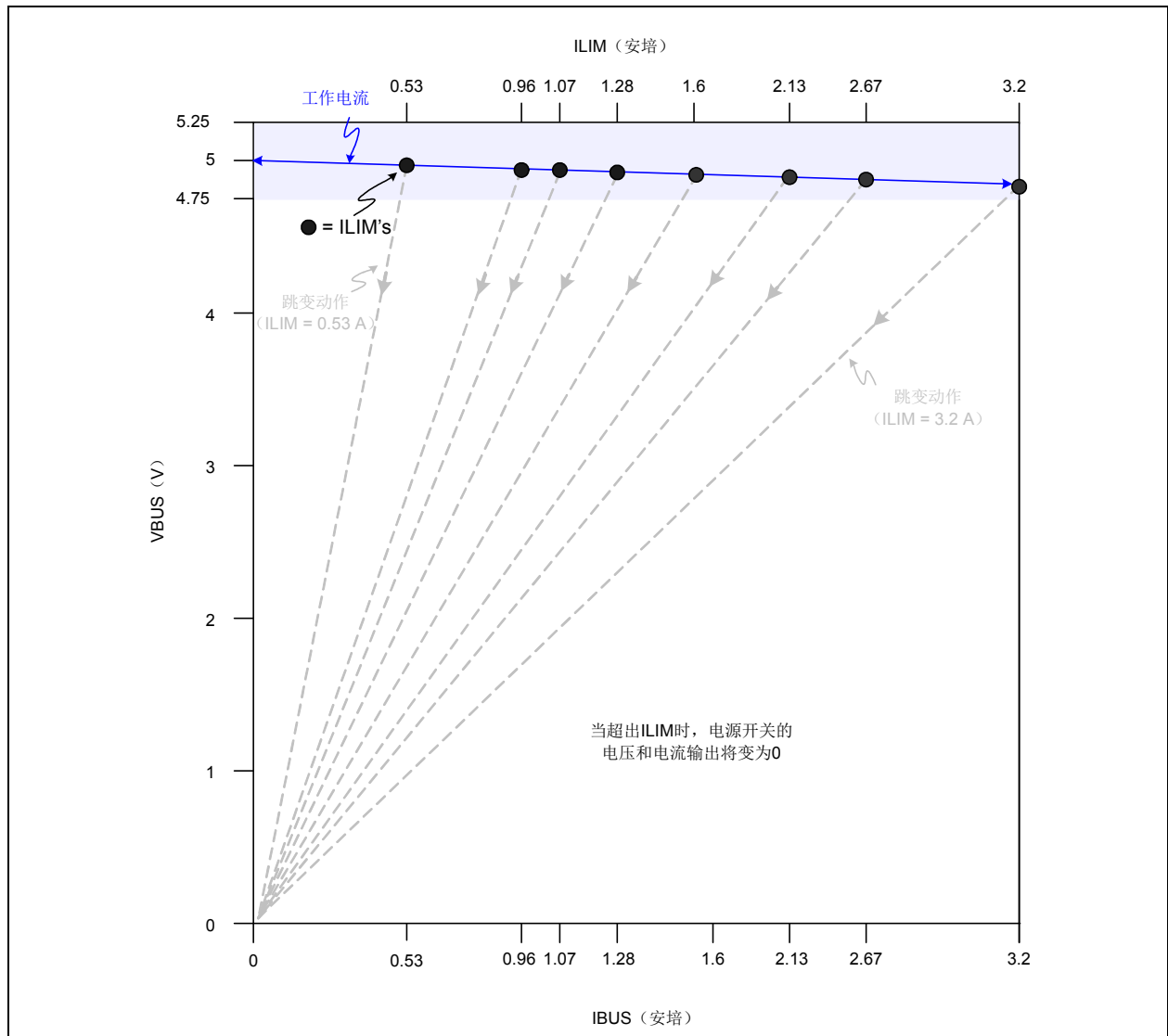
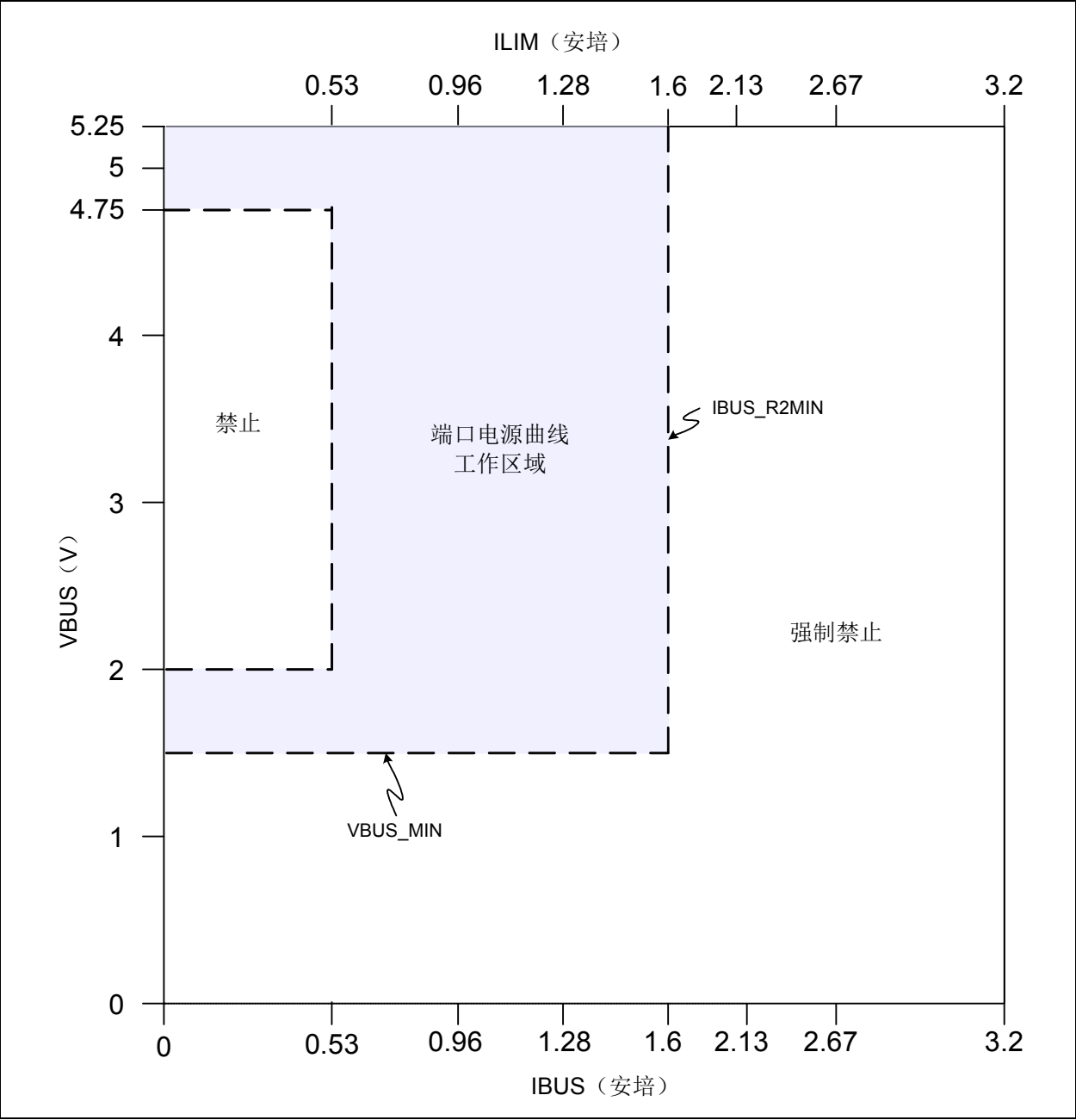


图12-2 说明了这些USB端口电源曲线参数之间的关系。

图12-2: 端口电源曲线参数示例



12.3.2 温度管理

PPC采用内部温度管理：

12.3.2.1 热关断

温度管理包括硬件实现的热关断，对应于最大允许内部裸片温度（ t_{TSD} ）。如果内部温度超过该值，端口电源开关（两个端口）将立即断开，直到温度低于 $t_{TSD} - t_{TSD_HYST}$ 为止。

12.3.3 VBUS放电

至少满足以下条件之一时，PPC会通过内部的 100Ω 电阻对VBUS进行放电：

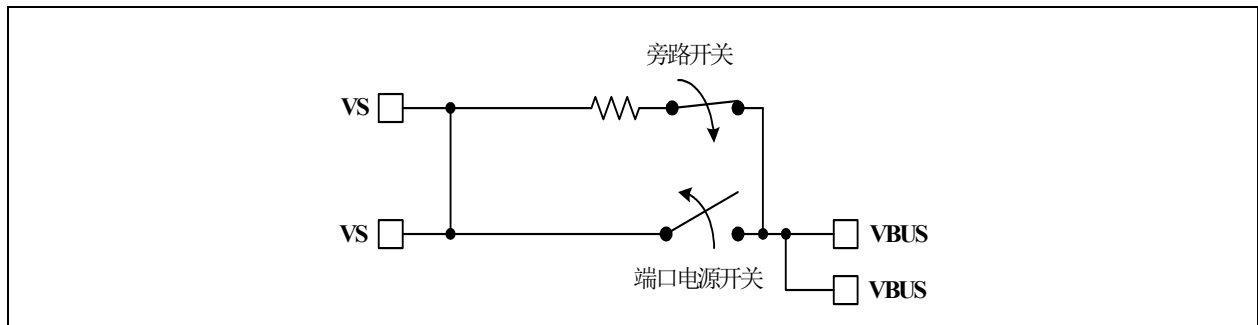
- **PWR_EN**控制（仅限UPD360-A/UPD360-B）被禁止（在**PWR_EN**控制的无效边沿触发）（仅限独立DFP模式）。
- **VS**电压降至低于导致端口电源开关被禁止的指定阈值 V_{S_UVLO} （2.5V）。
- 受命令指示进入休眠功耗状态时。
- 从错误状态恢复时。
- 通过**DISCHARGE**位发出命令时。
- 每次在**VBUS**旁路开关闭合之后激活端口电源开关时。
- 每次在端口电源开关闭合之后激活**VBUS**旁路开关时。

当**VBUS**放电电路被激活时，在 $t_{DISCHARGE}$ 时间结束时，PPC会确认**VBUS**已放电。如果**VBUS**电压不低于 V_{TEST} 电压，则会标记放电错误（通过将**电源中断状态寄存器（PWR_INT_STS）**中的**VBUS放电错误位置1**），并且PPC会进入错误状态。

12.4 VBUS旁路开关

器件包含提供VBUS电流的电路，如图12-3所示。VBUS旁路开关和端口电源开关永远不会同时闭合。

图12-3： VBUS旁路开关



12.5 端口电源控制器寄存器

本节详细介绍了端口电源控制器寄存器。有关器件的完整寄存器映射的概述，请参见第 19 页的第 4.0 节“寄存器映射”。

表 12-1: 端口电源控制器寄存器映射

地址	寄存器名称 (符号)
0400h	PPC VBUS 电流测量寄存器 (PPC_VBUS_CURRENT)
0401h	保留供将来扩展使用
0402h	PPC VBUS 端口状态寄存器 (PPC_VBUS_PORT_STS)
0403h	PPC 中断状态寄存器 1 (PPC_INT_STS1)
0404h	PPC 中断状态寄存器 2 (PPC_INT_STS2)
0405h–040Eh	保留供将来扩展使用
040Fh	PPC 一般状态寄存器 (PPC_GENERAL_STS)
0410h	保留供将来扩展使用
0411h	PPC 一般配置 1 寄存器 (PPC_GENERAL_CFG1)
0412h	保留供将来扩展使用
0413h	PPC 一般配置 3 寄存器 (PPC_GENERAL_CFG3)
0414h	PPC 限流寄存器 (PPC_CURRENT_LIMIT)
0415h–07FFh	保留供将来扩展使用

注： 在任何情况下都不能对保留的地址空间执行写操作。否则，可能导致意外的操作与结果。

12.5.1 PPC VBUS 电流测量寄存器 (PPC_VBUS_CURRENT)

地址: 0400h 大小: 8位

该寄存器用于存储通过VBUS引脚提供给外部设备的测量电流值。当器件处于活动功耗状态时，该值会持续更新。位的权重以mA为单位，范围为0 mA至3400 mA（1 LSB = 13.3 mA）。

当端口电源开关断开（或VBUS放电）时会清零该数据。

注： 电流<100 mA时，不使用此寄存器的数据。

Bit	说明	类型	默认值
7	1706.2 mA	RO	0b
6	853.1 mA	RO	0b
5	426.6 mA	RO	0b
4	213.3 mA	RO	0b
3	106.6 mA	RO	0b
2	53.3 mA	RO	0b
1	26.7 mA	RO	0b
0	13.3 mA	RO	0b

12.5.2 PPC VBUS 端口状态寄存器 (PPC_VBUS_PORT_STS)

地址: 0402h 大小: 8位

Bit	说明	类型	默认值
7	保留	RO	-
6	ALERT 该位在PPC中发生错误（ERR）时置为有效。它也会将中断状态寄存器（INT_STS）中的PPC_INT位置为有效（如果允许）。 该位也可被配置为在LOW_CUR（便携式设备汲取较少电流，可能已完成充电）位置1并通过ALERT_LINK位设为联动时置为有效。当所有可能使ALERT位置为有效的错误条件（错误条件以及设为联动的LOW_CUR）都已消除或根据需要复位时，会将ALERT位置为无效。	RO	0b
5	保留	RO	-
4	CC_MODE 指示VBUS上的IBUS电流已超出ILIM且向区域2提供电流。	RO	0b
3:0	保留	RO	-

UPD360

12.5.3 PPC中断状态寄存器1 (PPC_INT_STS1)

地址: 0403h 大小: 8位

Bit	说明	类型	默认值
7	ERR 指示VBUS引脚上检测到错误且PPC已进入错误状态。向该位写0将清除错误状态并使PPC恢复活动状态。写入1时，会检查所有错误条件。如果所有错误条件都已消除，PPC会恢复活动状态。在进入错误状态时，该位由PPC自动置1。如果该寄存器中的任何其他位置1，器件不会退出错误状态。 0: 未检测到错误 1: 检测到一个或多个错误，且PPC已进入错误状态。	R/W1C	0b
6	保留	RO	-
5	RESET 指示PPC刚刚被复位，应重新编程。该位会在上电时置1。该位在被写入1或PWR_EN引脚（仅限UPD360-A/UPD360-B）翻转时清零（仅限独立DFP模式）。该位置1时，ALERT位置为有效。该数据会在休眠状态下保持。	R/W1C	0b
4	MIN_KEEP_OUT 指示VBUS引脚上的V-I输出已降到V _{BUS_MIN} 以下。如果错误条件已消除或ERR位清零，则该位会在被写入1时清零。该位会使ALERT置为有效以及使器件进入错误状态。	R/W1C	0b
3	TSD 指示内部温度已超过t _{TSD} 阈值，器件已进入错误状态。如果错误条件已消除或ERR位清零，则该位会在被写入1时清零。该位会使ALERT置为有效以及使器件进入错误状态。	R/W1C	0b
2	OVER_VOLT 指示VS电压已超过V _{S_OV} 阈值，器件已进入错误状态。如果错误条件已消除或ERR位清零，则该位会在被写入1时清零。该位会使ALERT置为有效以及使器件进入错误状态。	R/W1C	0b
1	保留	R/W1C	0b
0	OVER_ILIM 指示IBUS电流已超出VBUS引脚上的ILIM阈值（PPC限流寄存器（PPC_CURRENT_LIMIT））和I _{BUS_R2MIN1} 阈值设置（以涵盖CC情况）。如果错误条件已消除或ERR位清零，则该位会在被写入1时清零。该位会使ALERT置为有效以及使器件进入错误状态。	R/W1C	0b

12.5.4 PPC中断状态寄存器2（PPC_INT_STS2）

地址： 0404h 大小： 8位

Bit	说明	类型	默认值
7:6	保留	RO	-
5	VS_LOW 指示VS电压低于 V_{S_UVLO} （2.5V）阈值，且端口电源开关保持断开状态。当VS电压高于 V_{S_UVLO} 阈值时，该位将自动清零。	R/W1C	0b
4:0	保留	RO	-

12.5.5 PPC一般状态寄存器（PPC_GENERAL_STS）

地址： 040Fh 大小： 8位

Bit	说明	类型	默认值
7:5	保留	RO	-
4	CC_MODE 指示IBUS电流已超出ILIM（PPC限流寄存器（PPC_CURRENT_LIMIT））。该位标识端口电源开关限流电路已激活且开关处于限流模式下。当端口电源开关断开时，该位将为“0”。	R/W1C	0b
3	PWR_EN 该位反映端口电源开关控制状态。在PWR_EN引脚（仅限UPD360-A/UPD360-B）和/或PWR_EN_SET位状态发生更改时，该位自动置1和清零。	R/W1C	0b
2	LOW_CUR 指示便携式设备已将其充电电流降至VBUS的阈值以下并可能已完成充电。该位在被写入1时清零，且不会导致ALERT位置为有效，除非ALERT_LINK位置1。	R/W1C	0b
1:0	保留	RO	-

12.5.6 PPC一般配置1寄存器（PPC_GENERAL_CFG1）

地址：0411h 大小：8位

Bit	说明	类型	默认值
7	ALERT_MASK 屏蔽PPC中断状态寄存器1（PPC_INT_STS1）中所有中断的错误（OVER_ILIM和TSD除外）。 0：当检测到错误条件或指示事件时，PPC VBUS端口状态寄存器（PPC_VBUS_PORT_STS）中的ALERT位置为有效。 1：只有在检测到OVER_ILIM或TSD时，PPC VBUS端口状态寄存器（PPC_VBUS_PORT_STS）中的ALERT位才置为有效。	R/W	0b
6	ALERT_LINK 在LOW_CUR位置为有效时，联动将PPC VBUS端口状态寄存器（PPC_VBUS_PORT_STS）中的ALERT位置为有效。 0：如果LOW_CUR指示位置为有效，则PPC VBUS端口状态寄存器（PPC_VBUS_PORT_STS）中的ALERT位不会置为有效。 1：如果LOW_CUR指示位置为有效，则PPC VBUS端口状态寄存器（PPC_VBUS_PORT_STS）中的ALERT位会置为有效。	R/W	0b
5	DISCHARGE 当PPC处于活动状态时，强制VBUS引脚复位和放电。向该位写入逻辑1将使端口电源开关断开以及放电电路激活并对VBUS放电。在DISCHARGE_TIME_SEL[1:0]字段定义的tDISCHARGE时间内，端口电源开关将保持断开状态，且100Ω电阻与GND短接。该位自清零。	R/W	0b
4	PWR_EN_SET 控制端口电源开关是否可闭合并影响功耗状态。此位与PWR_EN引脚进行逻辑或运算（仅限独立DFP模式）。因此，PWR_EN引脚或该位必须为1才能使能端口电源开关。	R/W	0b
3:2	DISCHARGE_TIME_SEL[1:0] 该字段用于设置tDISCHARGE时间，具体如下： 00：100 ms 01：200 ms（默认值） 10：300 ms 11：400 ms	R/W	01b
1:0	保留	RO	-

12.5.7 PPC一般配置3寄存器（PPC_GENERAL_CFG3）

地址：0413h 大小：8位

Bit	说明	类型	默认值
7	PIN_IGNORE 如果该位置1，则在确定工作模式选择和功耗状态时将忽略PWR_EN引脚状态（仅限UPD360-A/UPD360-B）。	R/W	0b
6:5	保留	RO	-
4:3	PWR_STATE[1:0] 该字段指示当前PPC功耗状态，详细信息见下文。这些位在功耗状态发生改变时自动置1和清零。 00: 休眠 01: 检测 10: 活动 11: 错误	R/W	00b
2	BOOST# 指示VBUS上的IBUS电流大于2.0A。	R/W	0b
1:0	保留	RO	-

12.5.8 PPC限流寄存器（PPC_CURRENT_LIMIT）

地址：0414h 大小：8位

该寄存器控制端口电源开关使用的ILIM。在休眠状态下，该寄存器的内容会保持。更多信息，请参见第172页的第12.3.1节“限流”。

Bit	说明	类型	默认值
7:3	保留	RO	-
2:0	ILIM_VBUS[2:0] 用于设置ILIM值，具体如下： 000: 530 mA 001: 960 mA 010: 1070 mA 011: 1280 mA 100: 1600 mA 101: 2130 mA 110: 2670 mA 111: 3200 mA	R/W	注12-1

注12-1 工作在独立DFP模式（仅限UPD360-A/UPD360-B）下时，默认值由PWR_CAP0和PWR_CAP1引脚定义。

13.0 电源开关

为了使器件能够针对电池电量耗尽情况提供有效支持，还提供了一个集成式电源开关，用于在两个外部+3.3V电源之间进行选择：

- **3V3_ALW**：集成电源开关的+3.3V主电源输入。
- **3V3_VBUS**：自VBUS至集成电源开关的+3.3V电源输入。

通过电源开关，可使内核在正常情况下由**3V3_ALW**供电，在**3V3_ALW**不存在时由**3V3_VBUS**供电。这样可以有效进行连接检测和系统唤醒，而无需外部处理器干预（外部处理器处于休眠模式）。

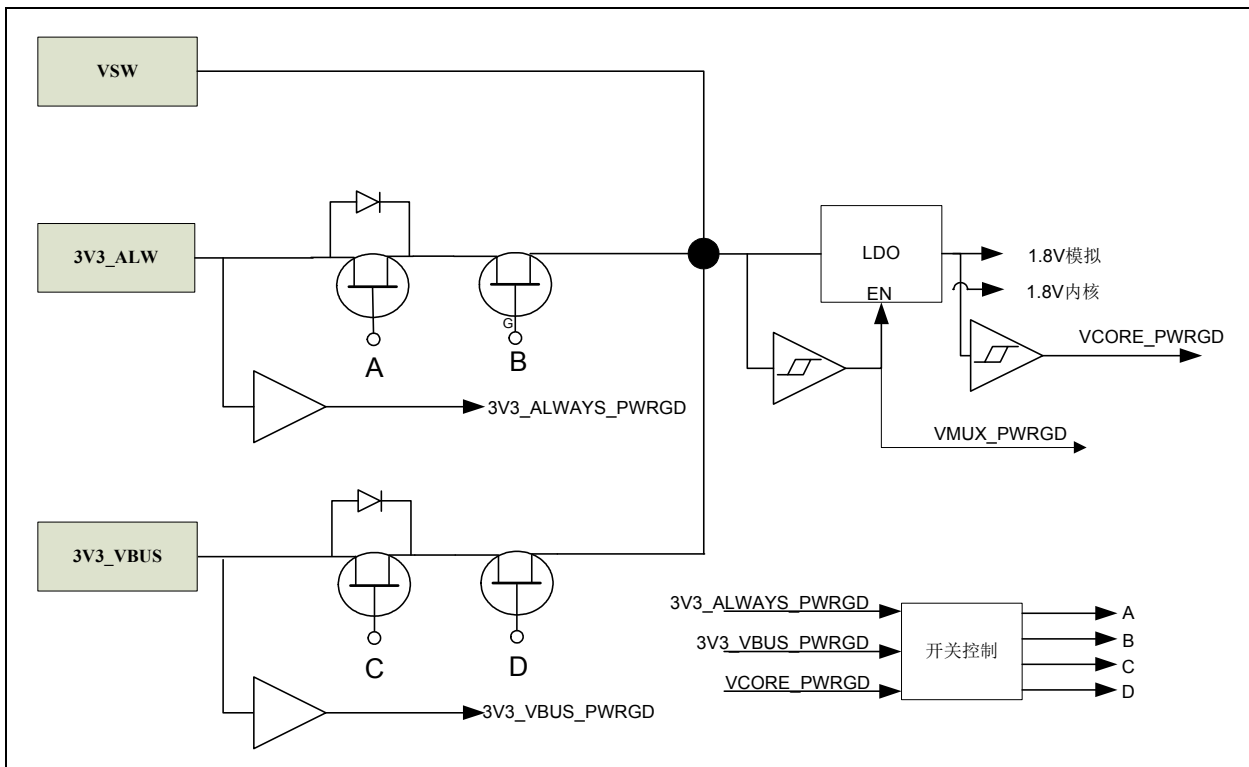
连接**3V3_ALW**和**3V3_VBUS**的是两个FET开关。第一个FET开关在输出两端接有一个二极管。

共有3个电压比较器。当**3V3_VBUS**超出2.7V时，**VBUS_PWRGD**有效。当**3V3_ALW**超出2.7V时，**3V3_ALW_PWRGD**有效。当内核电压达到工作电压时，**VCORE_PWRGD**有效。

如果**VCORE_PWRGD**未置为有效，则器件保持在复位状态。如果**VBUS_PWRGD**和**3V3_ALW_PWRGD**均未置为有效，则无论**VCORE_PWRGD**的状态如何，器件都保持在复位状态。

图13-1给出了内部电源开关的框图。

图13-1： 电源开关框图

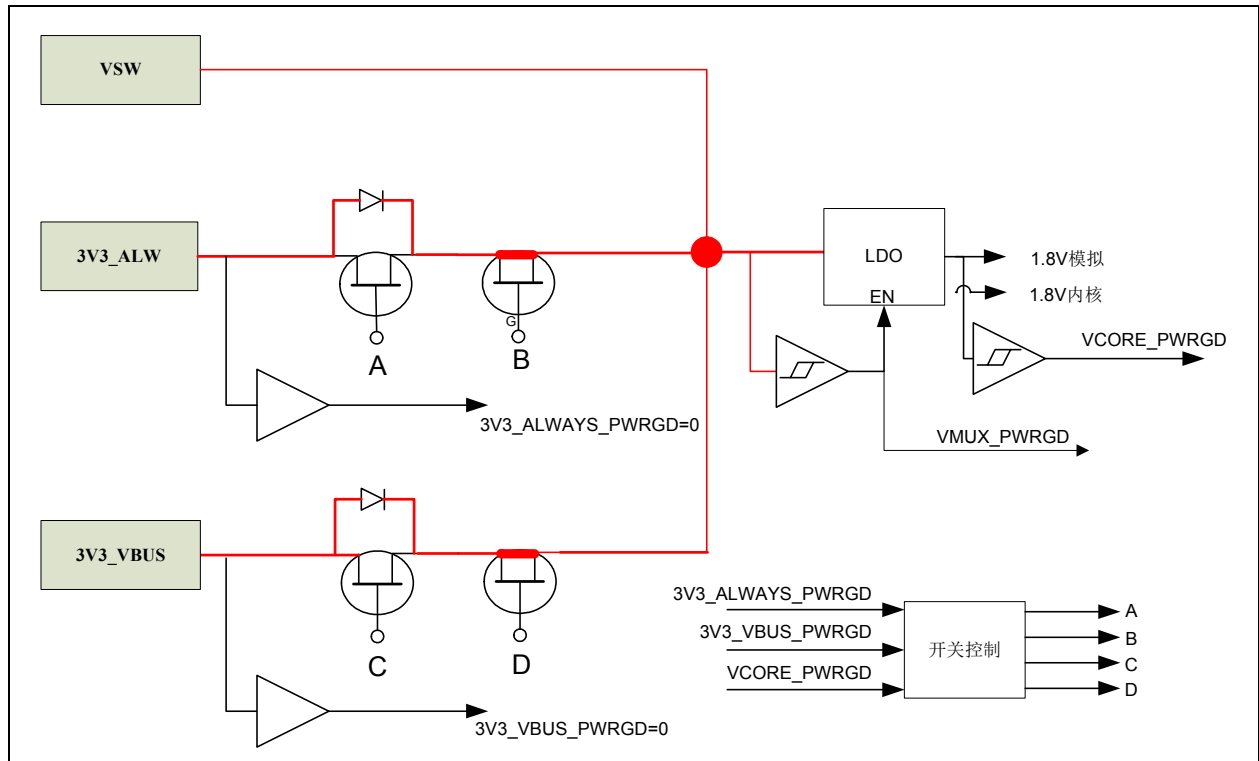


13.1 工作原理

13.1.1 上电

器件上电时，VBUS_PWRGD 和 3V3_ALW_PWRGD 均为低电平。在此类情况下，开关A和C断开，B和D闭合。内核稳压器的输入电压是3V3_ALW和3V3_VBUS中的较高者。二极管可防止反向驱动。

图 13-2: 电源开关：上电



13.1.2 仅3V3_ALW上电

在这种情况下，仅3V3_ALW上电（VBUS_PWRGD = 0, 3V3_ALW_PWRGD = 1）。发生这种情况时，FET开关D断开。50 μs后，FET开关A短路。当短接二极管以消除压降时，该序列可消除发生反向驱动的可能性。

图13-3： 电源开关：上电步骤1

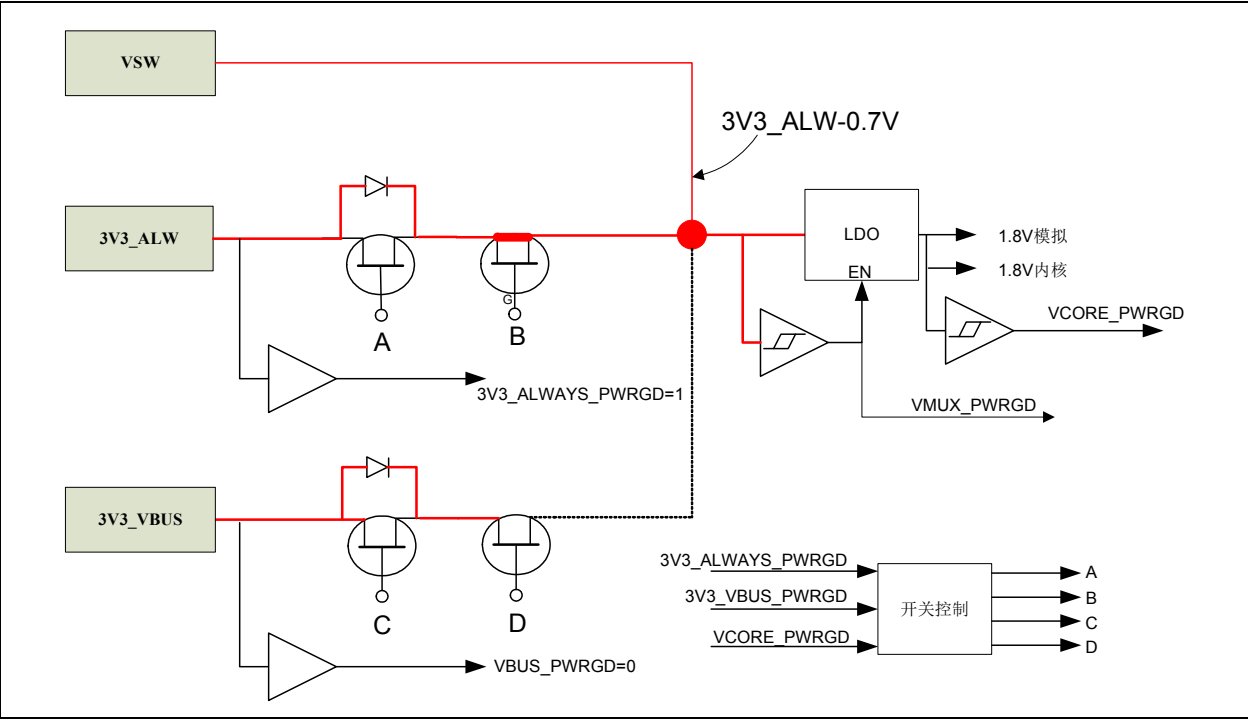
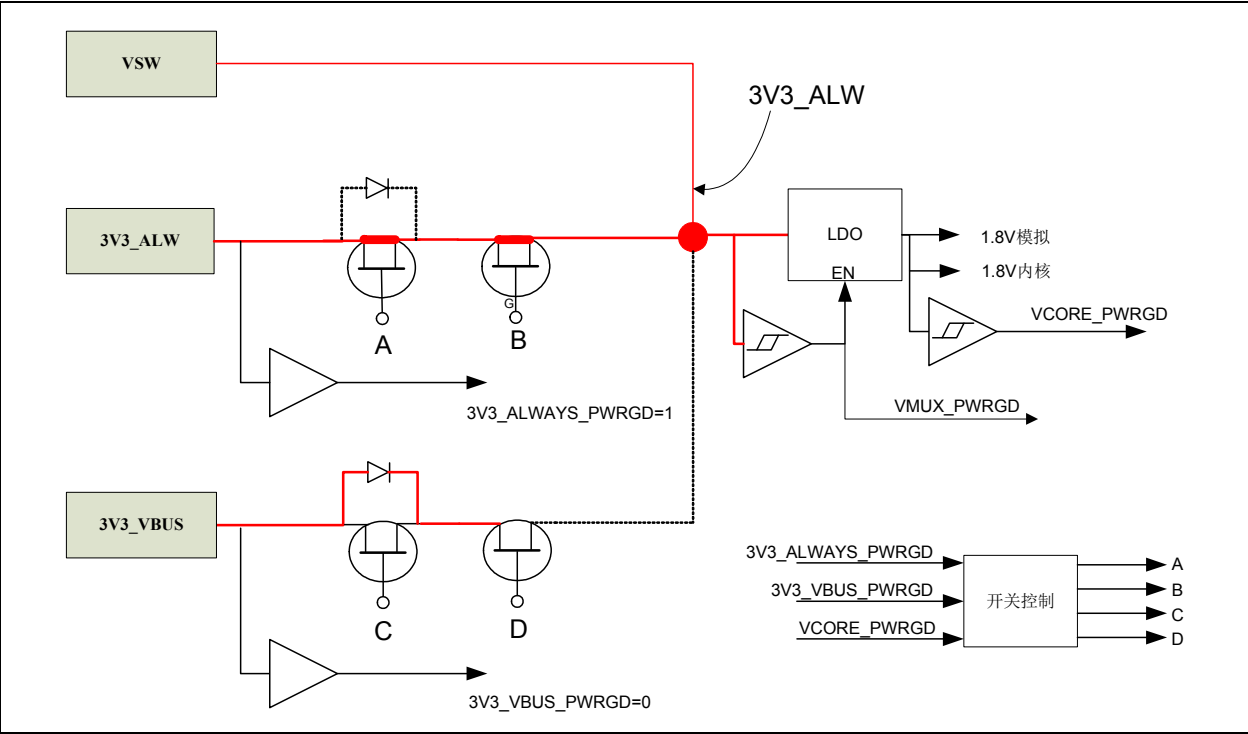


图13-4： 电源开关：上电步骤2



13.1.3 仅VBUS上电

这种情况与上一种情况相反。在这种情况下，仅VBUS上电（3V3_ALW_PWRGD始终等于0，VBUS_PWRGD在达到阈值时等于1）。一旦VBUS_PWRGD有效，内部稳压器即开启并开始正常工作。在此过程期间，3V3_ALW电源轨始终不会反向驱动。

图13-5： 电源开关：仅VBUS上电步骤1

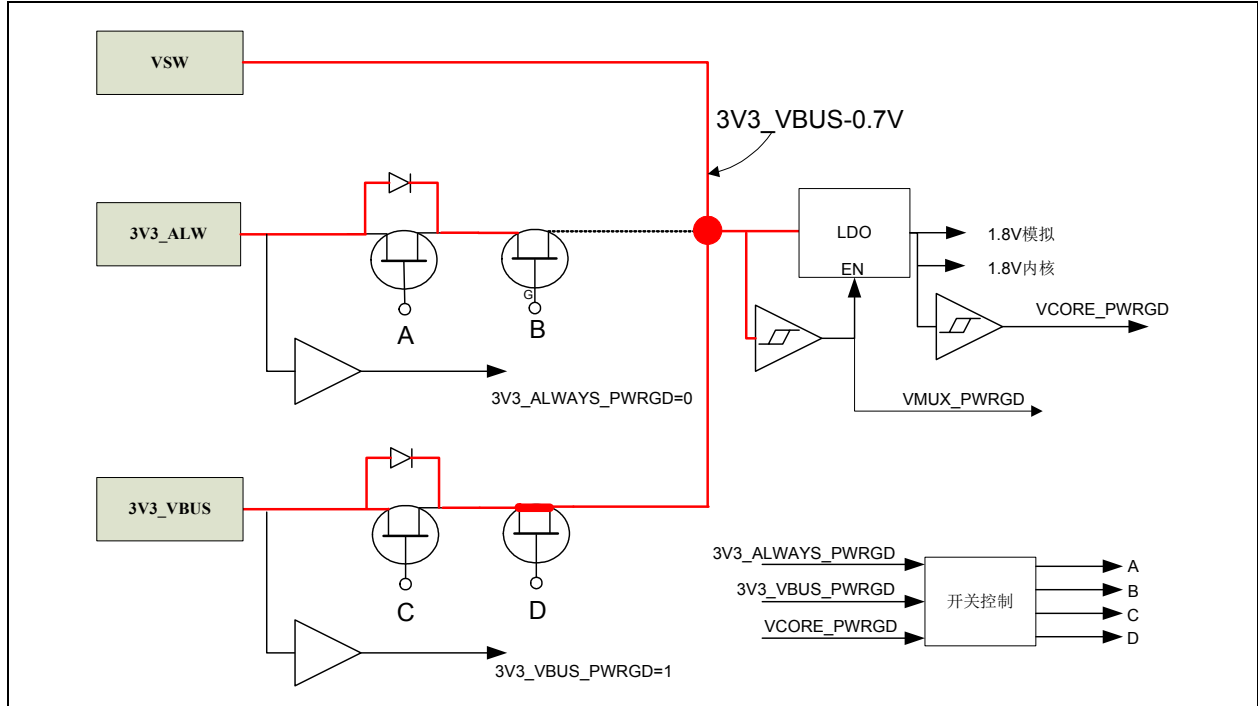
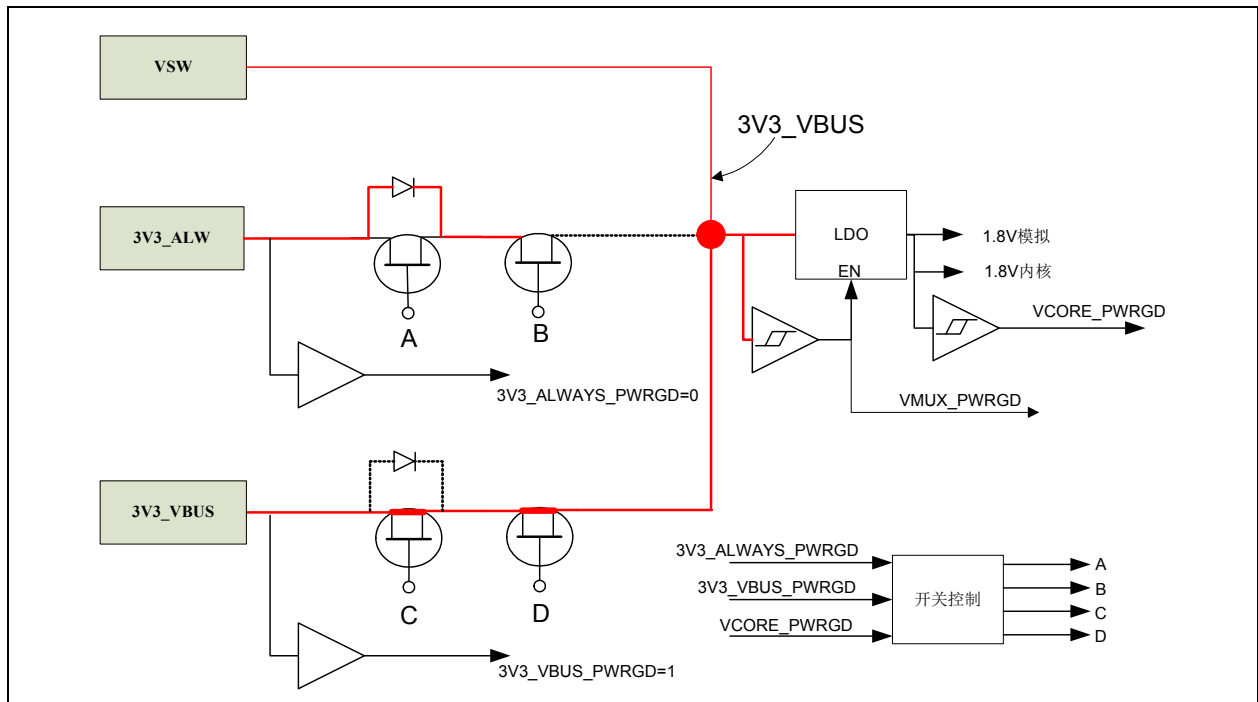


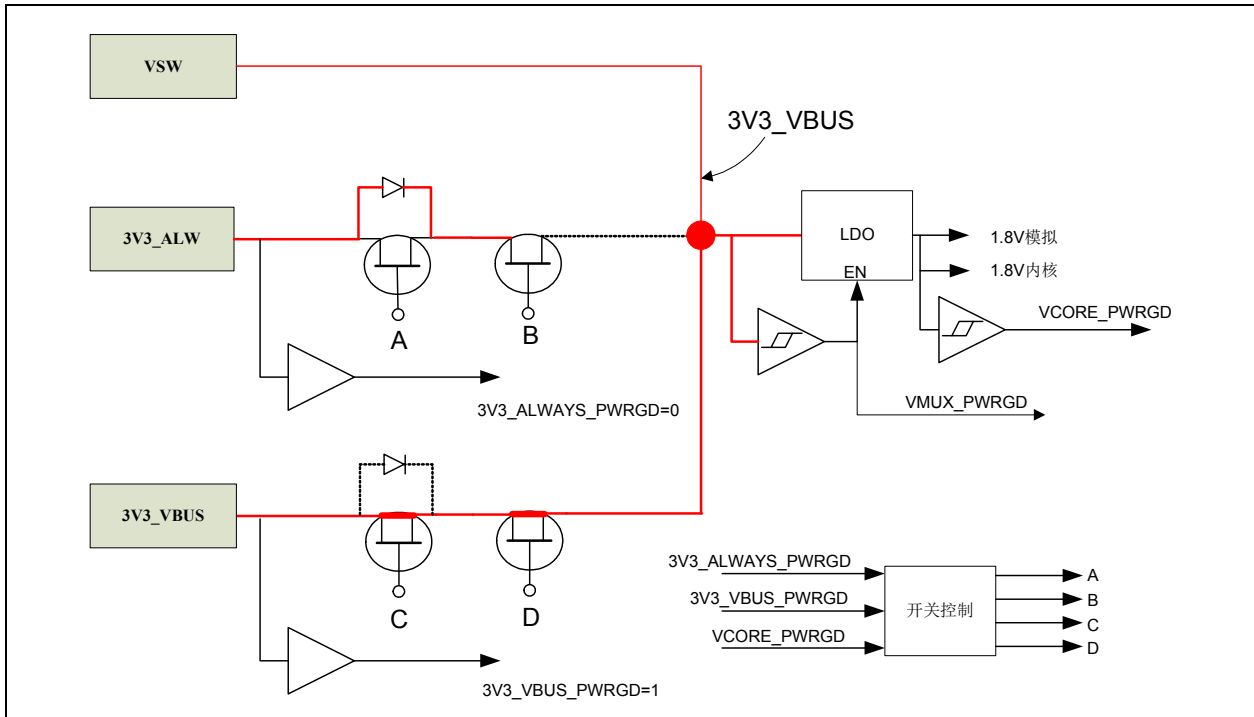
图13-6： 电源开关：仅VBUS上电步骤2



13.1.4 VBUS与3V3_ALW先后上电

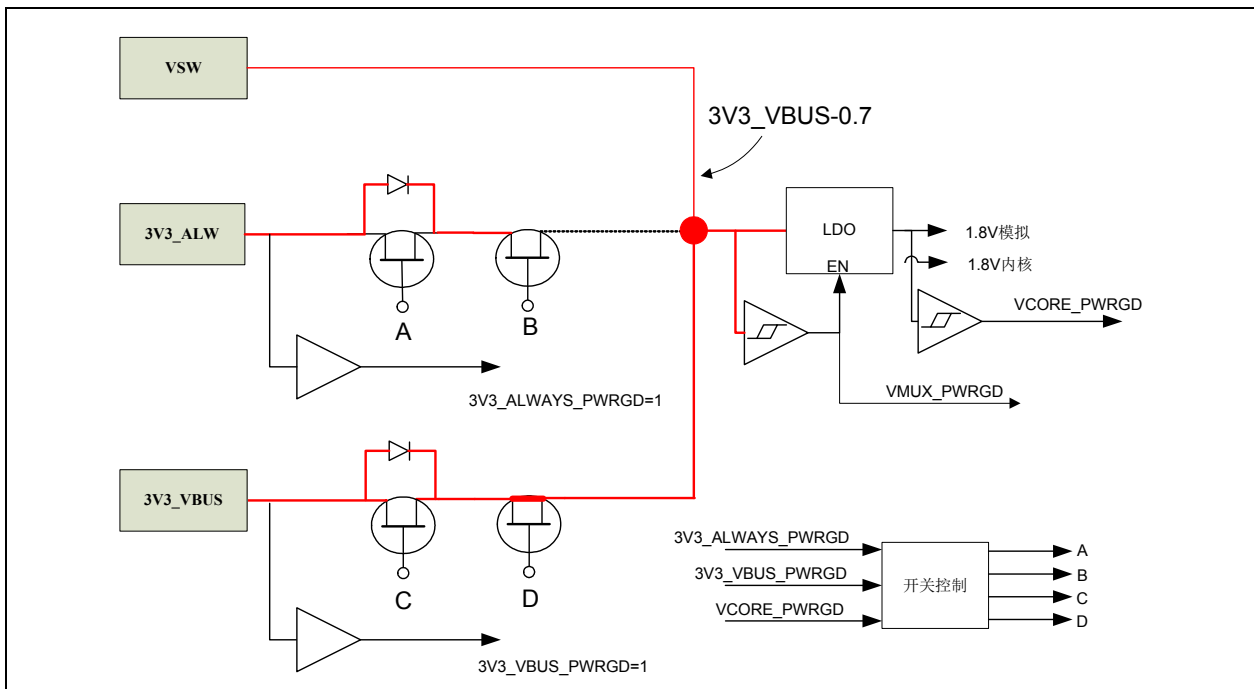
在这种情况下，待VBUS上电后，3V3_ALW开始上电（并将接管内核和为其供电）。图13-7仅显示了VBUS状态。

图13-7： 电源开关：VBUS与3V3_ALW先后上电步骤1



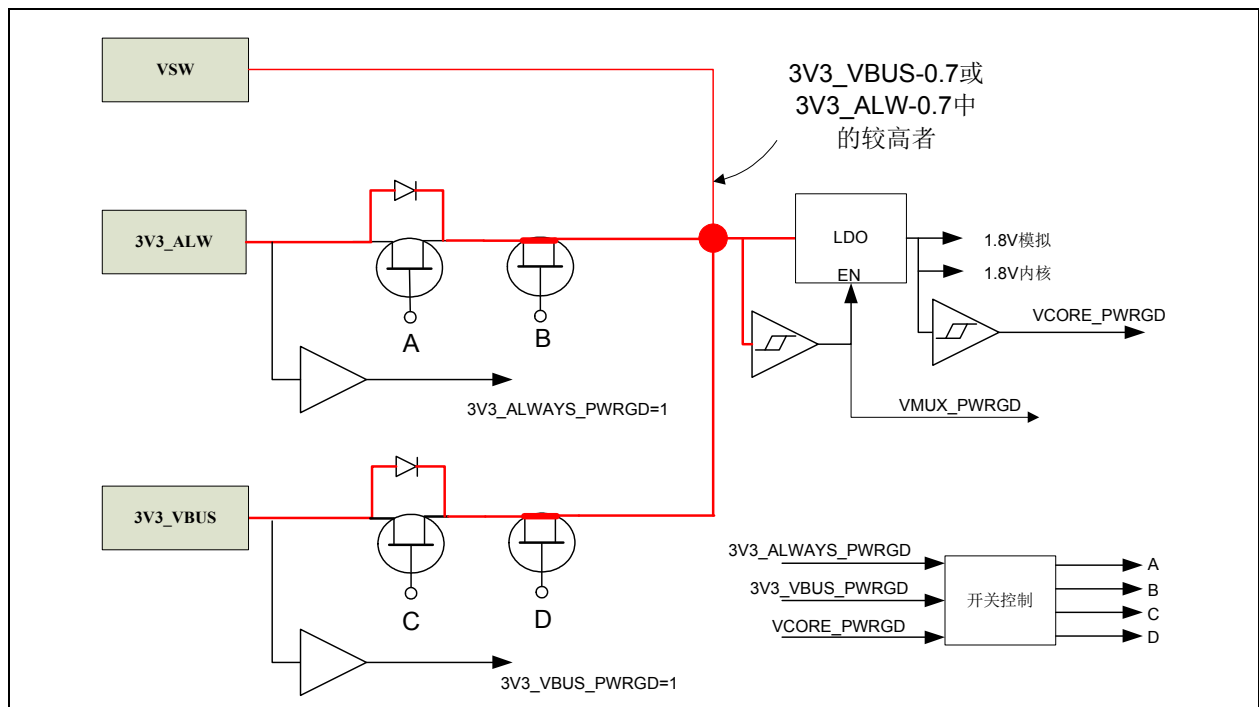
检测到3V3_ALW_PWRGD后，FET开关C会断开。二极管是必需项，因为3V3_ALW可能大于或小于VBUS。LDO输入端的电压降至3V3_VBUS-0.7V。

图13-8： 电源开关：VBUS与3V3_ALW先后上电步骤2



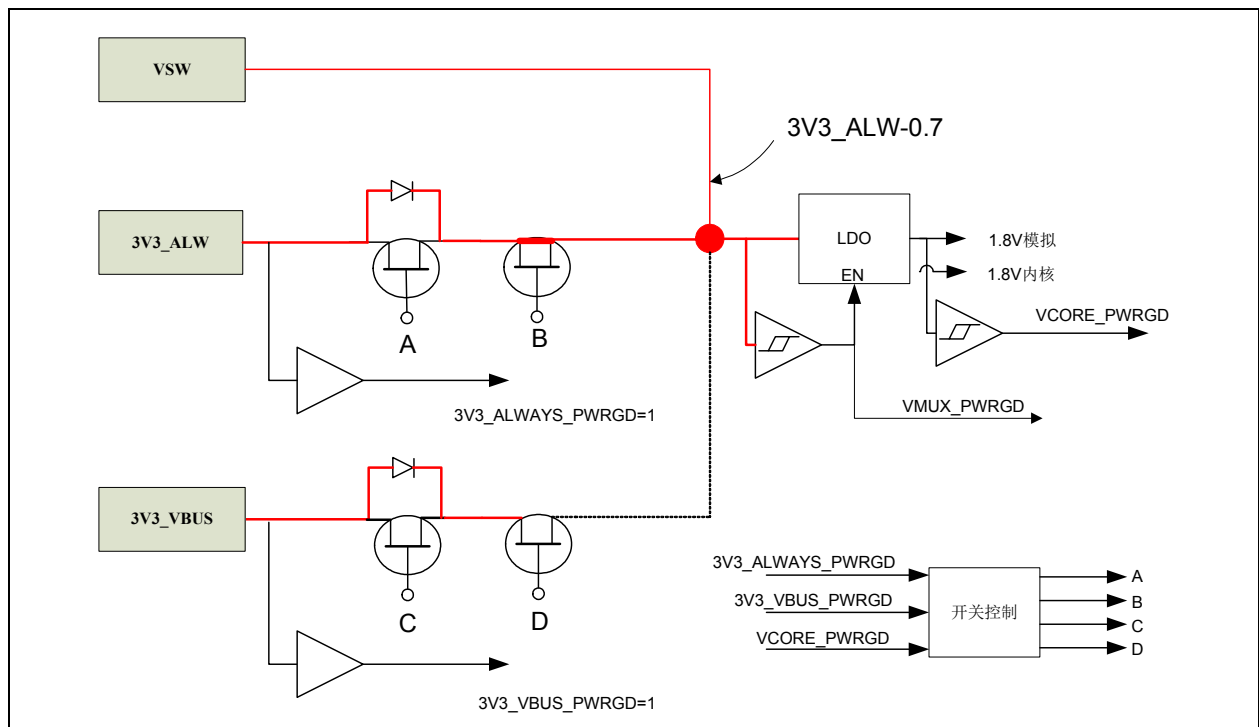
在FET C断开后经过50 μ s的时间，FET B短路。两个二极管可确保无反向驱动。内核稳压器的输入电压将为VBUS - 0.7V或3V3_ALW - 0.7V的较高者。

图13-9： 电源开关：VBUS与3V3_ALW先后上电步骤3



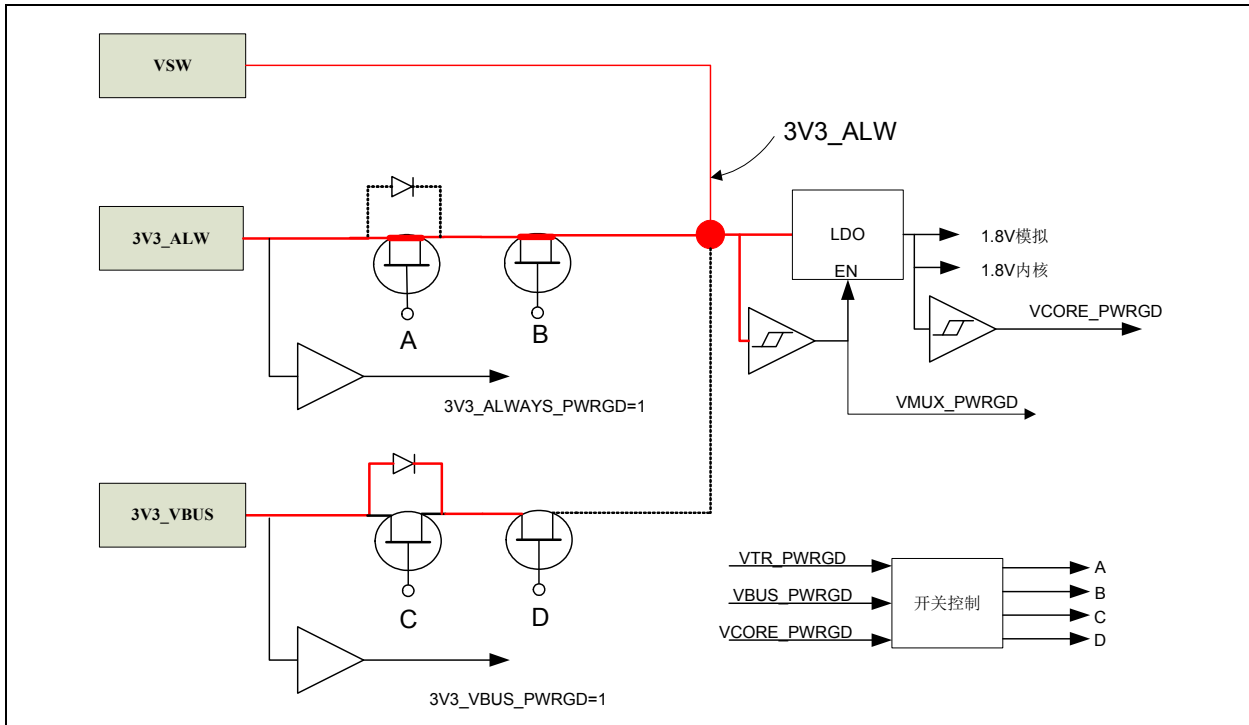
在FET B短路后经过50 μ s的时间，FET D断开。内核稳压器输入现为3V3_ALW - 0.7V。

图13-10： 电源开关：VBUS与3V3_ALW先后上电步骤4



在最后一步中，在FET D断开后经过50 μ s的时间，FET A短路。内核稳压器的输入现变为3V3_ALW。

图13-11： 电源开关：VBUS与3V3_ALW先后上电步骤5



13.1.5 3V3_ALW与VBUS先后上电

在这种情况下，待3V3_ALW上电后，VBUS开始上电。3V3_ALW将仍然是内核的电源，FET开关不会发生任何变化。

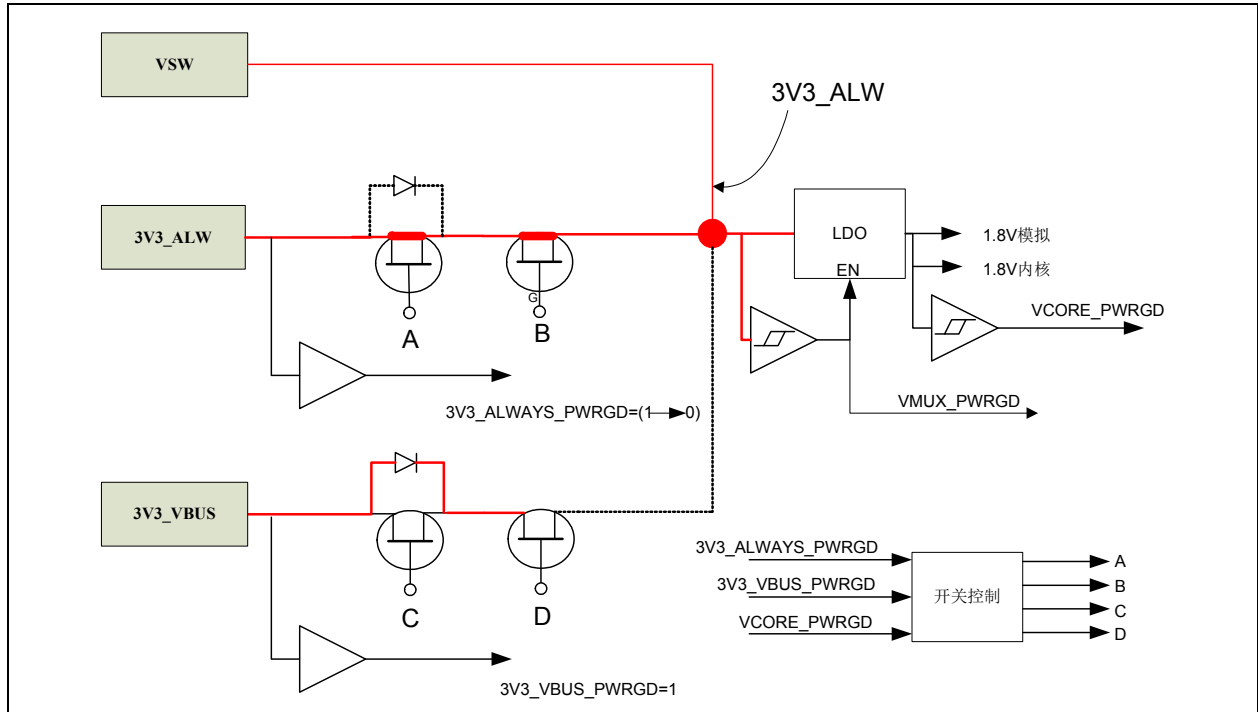
13.1.6 VBUS和3V3_ALW均上电，VBUS消失

在这种情况下，3V3_ALW和VBUS均存在。内核始终由3V3_ALW供电。如果VBUS消失，内核电压无任何变化。此时将向处理器产生中断，使其了解VBUS状态已更改。

13.1.7 VBUS和3V3_ALW均上电，3V3_ALW消失

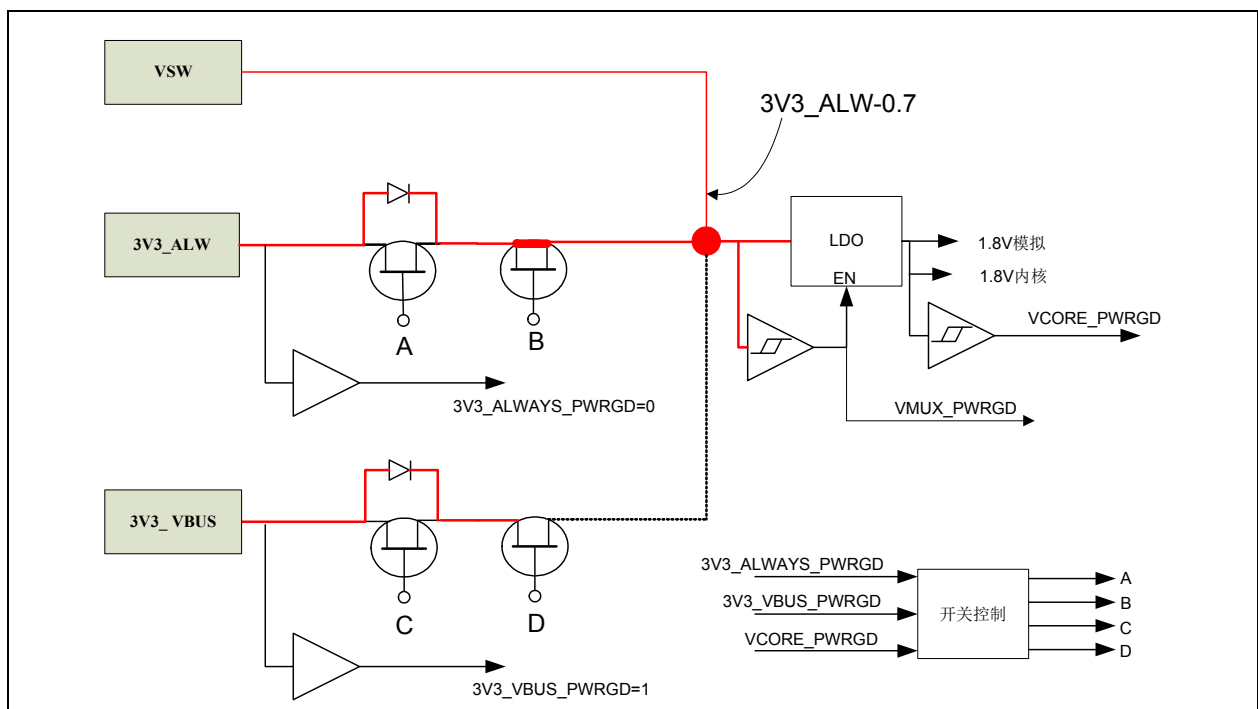
在这种情况下，3V3_ALW和VBUS均存在。内核始终由3V3_ALW供电。如果3V3_ALW消失，开关将先切换到初始上电状态，然后切换到仅VBUS状态。

图13-12: 电源开关: VBUS存在，3V3_ALW消失步骤1



首先，FET A断开。内核稳压器输入的电压变为3V3_ALW-0.7

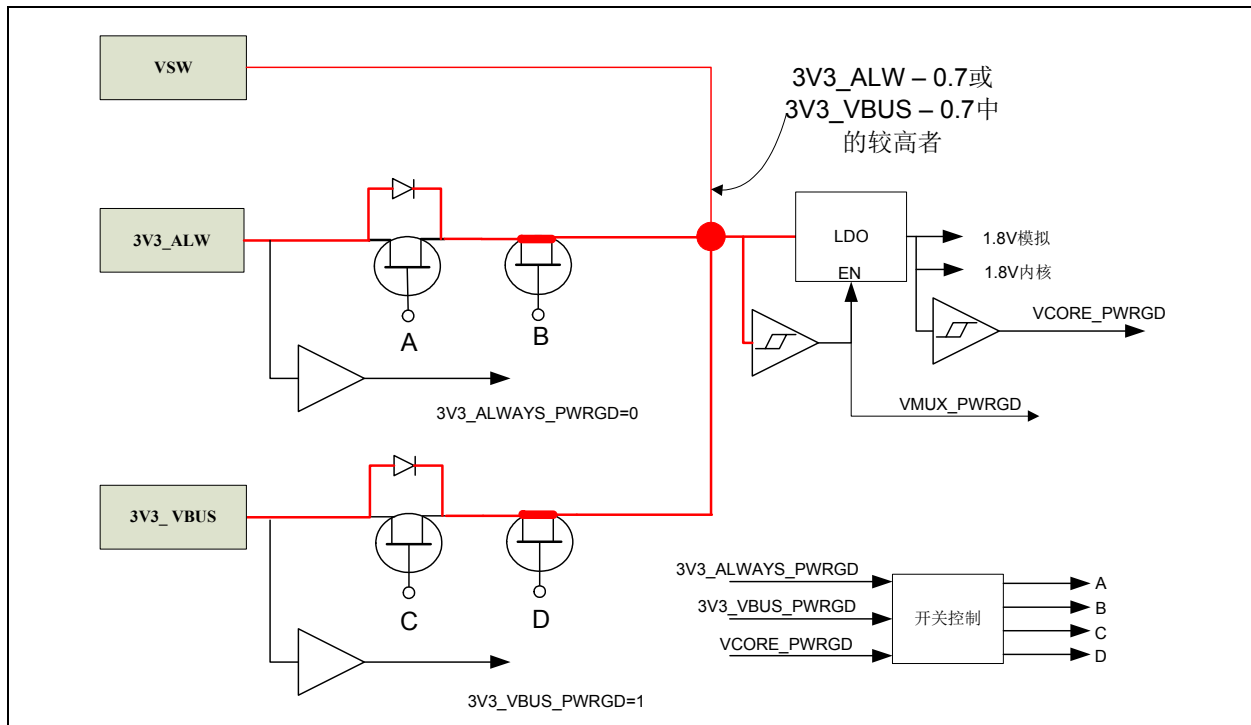
图13-13: 电源开关: VBUS存在，3V3_ALW消失步骤2



UPD360

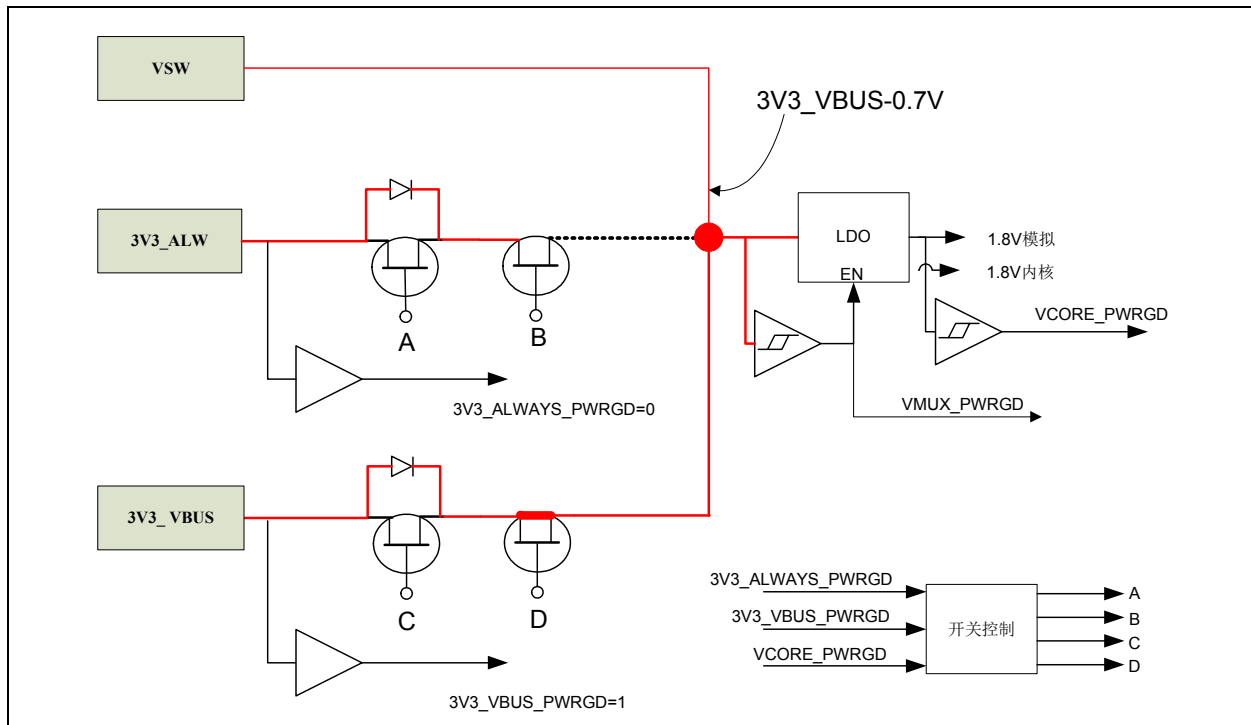
50 μ s后，FET 闭合。内核稳压器输入的电压将是3V3_ALW-0.7和3V3_VBUS-0.7中的较高者。

图13-14： 电源开关：VBUS存在，3V3_ALW消失步骤3



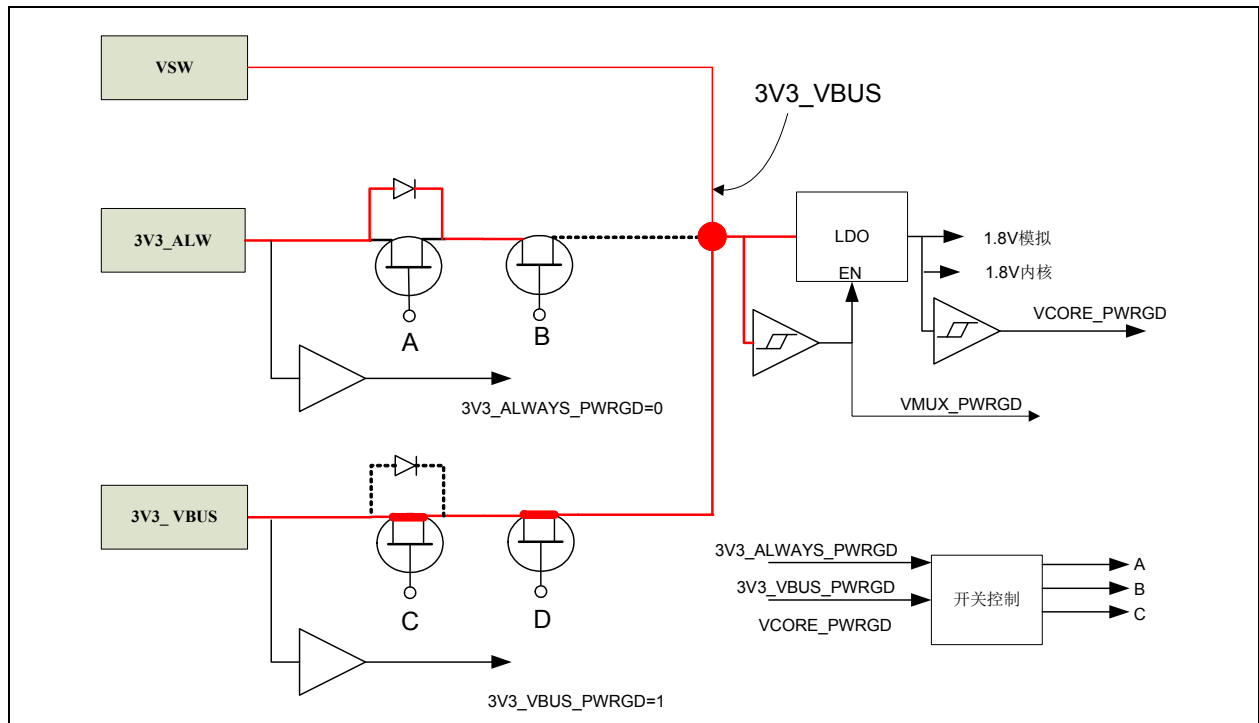
50 μ s后，FET B断开。内核稳压器输入的电压为VBUS-0.7。

图13-15： 电源开关：VBUS存在，3V3_ALW消失步骤4



50 μ s后，FET C短路。内核稳压器输入的电压为3V3_VBUS。

图13-16： 电源开关：VBUS存在，3V3_ALW消失步骤5



13.2 软件改写

如果3V3_VBUS和3V3_ALW均可用，则电源开关会自动选择3V3_ALW进行工作。这可通过电源开关控制寄存器（PWR_SW_CTL）的VBUS开关使能改写（VBUS_SW_EN_OVR）位进行改写，以强制开关工作在VBUS下。该位置1时，开关的自动切换机制被禁止。

13.3 电源开关中断

电源开关中断提醒软件电源开关的状态发生更改。电源开关中断状态寄存器（PWR_INT_STS）中列出了这些事件。在电源开关中断状态寄存器（PWR_INT_STS）中置为有效的位清零之前，中断始终存在。单独的中断事件可通过电源开关中断允许寄存器（PWR_INT_EN）允许。电源开关中断状态源能够触发异步唤醒。

13.4 电源开关寄存器

本节详细介绍了电源开关寄存器。有关器件的完整寄存器映射的概述，请参见第19页的第4.0节“寄存器映射”。

表13-1： 电源开关寄存器映射

地址	寄存器名称（符号）
1C00h	电源开关控制寄存器（PWR_SW_CTL）
1C01h	电源开关状态寄存器（PWR_SW_STS）
1C02h	电源开关中断状态寄存器（PWR_INT_STS）
1C03h	电源开关中断允许寄存器（PWR_INT_EN）
1C04h	电源开关状态寄存器（PWR_SW_STATE_STS）
1C05h – 1FFFh	保留供将来扩展使用

UPD360

13.4.1 电源开关控制寄存器 (PWR_SW_CTL)

地址: 1C00h 大小: 8位

Bit	说明	类型	默认值
7:2	保留	RO	-
1	电源开关使能 (PM_SW_ENB) 此低电平有效信号可使能电源开关。	R/W	0b
0	VBUS开关使能改写 (VBUS_SW_EN_OVR) 该位置为有效时, 电源开关采用3V3_VBUS电源。只有在器件当前工作在3V3_ALW下且VBUS电源存在时, 该位才有意义。 该位可禁止电源开关的自动切换机制。	R/W	0b

13.4.2 电源开关状态寄存器 (PWR_SW_STS)

地址: 1C01h 大小: 8位

Bit	说明	类型	默认值
7:6	保留	RO	-
5	3V3_ALW POR (3V3_ALW_RDY) 主VBAT外部电源的POR输出。电源高于2.85V时, 高电平有效。电源低于2.7V时, 低电平有效。	RO	0b
4	3V3_VBUS POR (VBUS_RDY) VBUS外部电源的POR输出。电源高于4V时, 高电平有效。电源低于2.7V时, 低电平有效。	RO	0b
3	保留	RO	-
2	PWR_RDY 指示VSW和VDD18电源已准备好供电。	RO	1b
1	3V3_ALW开关栅极关断 (3V3_ALW_SW_OKB) 指示VSW相对于3V3_ALW的状态。 0: VSW < 3V3_ALW 1: VSW > 3V3_ALW	RO	0b
0	VBUS开关栅极关断 (VBUS_SW_OKB) 指示VSW相对于VBUS的状态。 0: VSW < VBUS 1: VSW > VBUS	RO	0b

13.4.3 电源开关中断状态寄存器 (PWR_INT_STS)

地址: 1C02h 大小: 8位

Bit	说明	类型	默认值
7:6	保留	RO	-
5	3V3_ALW至VBUS切换中断 (3V3_ALW2VBUS_INTR) 电源开关自动从选择3V3_ALW作为电源切换至选择VBUS作为电源后, 该中断置为有效。 写1可清零该位。写0不起作用。 注: VBUS 开关使能改写 (VBUS_SW_EN_OVR) 位置为有效会导致此中断置为有效。	R/WC	0b
4	VBUS至3V3_ALW切换中断POR (VBUS23V3_ALW_INTR) 电源开关自动从选择VBUS作为电源切换至选择3V3_ALW作为电源后, 该中断置为有效。 写1可清零该位。写0不起作用。 注: VBUS 开关使能改写 (VBUS_SW_EN_OVR) 位置为有效可阻止在VBUS丢失且存在3V3_ALW时将该中断置为有效。	R/WC	0b
3	3V3_ALW丢失中断 (3V3_ALW_LOST_INTR) 该中断在3V3_ALW POR (3V3_ALW_RDY) 从1切换为0时置为有效。它表示3V3_ALW电源已丢失。 写1可清零该位。写0不起作用。	R/WC	0b
2	3V3_ALW就绪中断 (3V3_ALW_RDY_INTR) 该中断在3V3_ALW POR (3V3_ALW_RDY) 从0切换为1时置为有效。它表示已检测到3V3_ALW电源。 写1可清零该位。写0不起作用。	R/WC	0b
1	VBUS丢失中断 (VBUS_LOST_INTR) 该中断在3V3_VBUS POR (VBUS_RDY) 从1切换为0时置为有效。它表示VBUS电源已丢失。 写1可清零该位。写0不起作用。	R/WC	0b
0	VBUS就绪中断 (VBUS_RDY_INTR) 该中断在3V3_VBUS POR (VBUS_RDY) 从0切换为1时置为有效。它表示已检测到VBUS电源。 写1可清零该位。写0不起作用。	R/WC	0b

13.4.4 电源开关中断允许寄存器（PWR_INT_EN）

地址：1C03h 大小：8位

Bit	说明	类型	默认值
7:6	保留	RO	-
5	3V3_ALW2VBUS_EN 为“0”时，阻止产生该中断。	R/W	0b
4	VBUS23V3_ALW_EN 为“0”时，阻止产生该中断。	R/W	0b
3	3V3_ALW_LOST_EN 为“0”时，阻止产生该中断。	R/W	0b
2	3V3_ALW_RDY_EN 为“0”时，阻止产生该中断。	R/W	0b
1	VBUS_LOST_EN 为“0”时，阻止产生该中断。	R/W	0b
0	VBUS_RDY_EN 为“0”时，阻止产生该中断。	R/W	0b

13.4.5 电源开关状态寄存器（PWR_SW_STATE_STS）

该寄存器指示3V3_ALW和3V3_VBUS开关的当前状态。

地址：1C04h 大小：8位

Bit	说明	类型	默认值
7:2	保留	RO	-
1	VBUS开关状态（3V3_VBUS_STATE） 0 = 禁止3V3_VBUS开关。 1 = 使能3V3_VBUS开关。	RO	0b
0	3V3_ALW开关状态（3V3_ALW_STATE） 0 = 禁止3V3_ALW开关。 1 = 使能3V3_ALW开关。	RO	0b

14.0 DISPLAYPORT 热插拔检测 (HPD)

14.1 概述

该器件提供硬件减载支持，用于检测器件的显示端口兼容的 **HPD** 输入引脚的状态。

USB Type-C DisplayPort 备用模式规范基于两个状态标志定义了 **HPD** 状态：

- **HPD_STATE**：指示 HPD 的逻辑状态是高电平还是低电平（分别表示为 **HPD_HIGH** 和 **HPD_LOW**）。为通过 USB 传送 HPD 状态，在接收 **IRQ_HPDP** 时，HPD 的逻辑状态被视为保持高电平；而在基于新机械连接对 HPD 去抖动期间，则为低电平。检测到 HPD 链路上为低电平并持续 2 ms（即，长于最大 **IRQ_HPDP** 脉冲检测时间）时，HPD 的逻辑状态从高电平切换为低电平。HPD 链路上存在毛刺时，HPD 的逻辑状态不变（如 DP v1.3 中所述）。
- **IRQ_HPDP**：指示 **IRQ_HPDP**（即，在 250 μ s 至 2 ms 之后，检测到 HPD 从高电平切换至低电平，然后从低电平切换至高电平，如 DP v1.3 中所述）。

注： HPD 引脚始终配置为推/挽驱动器。

14.2 HPD 接收器

该器件支持通过 **HPD** 引脚进行 HPD 检测（输入）。HPD 检测通过将 **HPD 使能位置 1** 并将 **HPD** 引脚配置为输入（通过 **HPD 控制寄存器 (HPD_CTL)** 的 **HPD 配置**）来使能。该位置 1 后，器件将持续监视 **HPD** 引脚的状态，具体如下：

- 发生 **IRQ_HPDP** 事件时，**HPD 中断状态寄存器 (HPD_INT_STS)** 的 **IRQ_HPDP** 中断位会置为有效。**IRQ_HPDP** 在 **HPD** 最初为高电平时被检测到，且置为无效的时间大于等于 **IRQ_HPDP 最短时间寄存器 (IRQ_HPDP_MIN_TIME)** 的值或小于等于 **IRQ_HPDP 最长时间寄存器 (IRQ_HPDP_MAX_TIME)** 的值。
- 去抖动并检测到从低电平到高电平的切换时，**HPD 中断状态寄存器 (HPD_INT_STS)** 的 **HPD_HIGH** 中断位会置为有效。**HPD_HIGH** 在 **HPD** 引脚最初为低电平时被检测到，且置为有效的时间大于等于 **HPD 高电平检测时间寄存器 (HPD_HIGH_DET_TIME)** 的值。
- 去抖动并检测到从高电平到低电平的切换时，**HPD 中断状态寄存器 (HPD_INT_STS)** 的 **HPD_LOW** 中断位会置为有效。**HPD_LOW** 在 **HPD** 引脚最初为低电平时被检测到，且置为有效的时间大于等于 **HPD 低电平检测时间寄存器 (HPD_LOW_DET_TIME)** 的值。

应用注意事项： **HPD 使能** 清零时，**HPD 中断状态寄存器 (HPD_INT_STS)** 的内容复位为 00h。

应用注意事项： 已允许和置为有效的中断会触发 **中断状态寄存器 (INT_STS)** 的 **HPD_INT** 中断。

已通过 **HPD 事件队列寄存器 (HPD_QUEUE)** 添加一种机制，用于记录最多四个 **HPD** 事件。此功能在 **HPD 使能置 1** 且 **HPD 配置** 指示 **HPD** 引脚用作输入时使能。更多详细信息，请参见第 199 页的第 14.4.4 节“**HPD 事件队列寄存器 (HPD_QUEUE)**”。

应用注意事项： **HPD** 引脚的状态可通过读取 **HPD 控制寄存器 (HPD_CTL)** 的 **HPD 状态** 来获取。

该功能的使用方式如下：

1. 软件配置 **IRQ_HPDP 最短时间寄存器 (IRQ_HPDP_MIN_TIME)**、**IRQ_HPDP 最长时间寄存器 (IRQ_HPDP_MAX_TIME)**、**HPD 高电平检测时间寄存器 (HPD_HIGH_DET_TIME)** 和 **HPD 低电平检测时间寄存器 (HPD_LOW_DET_TIME)**。为符合面向 **IRQ_HPDP** 的 DisplayPort 规范，**IRQ_HPDP 最短时间寄存器 (IRQ_HPDP_MIN_TIME)** 和 **IRQ_HPDP 最长时间寄存器 (IRQ_HPDP_MAX_TIME)** 应分别配置为 250 μ s 和 2 ms。同样，**HPD 低电平检测时间寄存器 (HPD_LOW_DET_TIME)** 应配置为大于 2 ms。
2. 必要时，可通过将 **HPD 中断允许寄存器 (HPD_INT_EN)** 的相应位置为有效来允许中断。
3. **HPD 控制寄存器 (HPD_CTL)** 的 **HPD 使能置 1** 且 **HPD 配置** 清零。

- 4. 发生热插拔事件时，将更新HPD事件队列寄存器（HPD_QUEUE），且中断将置为有效（如果已配置）。
- 5. HPD状态变化通过将HPD中断状态寄存器（HPD_INT_STS）的QUEUE_NOT_EMPTY、HPD_HIGH、HPD_LOW和IRQ_HPDP位置为有效来反映。它们可能导致中断状态寄存器（INT_STS）的HPD_INT置为有效（如果已允许）。
- 6. 软件读取HPD事件队列寄存器（HPD_QUEUE）以确定接收到的热插拔事件。
- 7. 软件写入HPD事件队列寄存器（HPD_QUEUE）并清除有效条目。

14.3 HPD传送

该器件支持通过HPD引脚进行HPD传送（输出）。此过程通过HPD控制寄存器（HPD_CTL）完成，方法是将HPD配置和HPD使能均置1。在HPD引脚上驱动的值由HPD输出值控制。

此外，还提供生成IRQ_HPDP的支持。这是通过在HPD IRQ生成时间寄存器（HPD_IRQ_GEN）中设定为无效的时间然后将HPD控制寄存器（HPD_CTL）的生成IRQ置1实现的。在发出中断脉冲后，生成IRQ会自清零。

该功能的使用方式如下：

- 1. 通过将HPD控制寄存器（HPD_CTL）的HPD配置和HPD使能置1将HPD引脚配置为输出。HPD的值由HPD输出值中设定的值定义。
- 2. 如果需要生成IRQ_HPDP，则HPD IRQ生成时间寄存器（HPD_IRQ_GEN）必须用所需HPD置为无效的时间进行设定。
- 3. 如果需要在生成IRQ_HPDP时接收到中断，则HPD中断允许寄存器（HPD_INT_EN）的相应位应使能。
- 4. 软件通过将HPD控制寄存器（HPD_CTL）的生成IRQ置1来请求IRQ_HPDP生成。HPD输出值必须也设为1b。
- 5. 在生成IRQ_HPDP中断后，生成IRQ位会自清零。
- 6. HPD中断状态寄存器（HPD_INT_STS）的IRQ_HPDP位置为有效，并会使中断状态寄存器（INT_STS）的HPD_INT中断置为有效（如果已允许）。

14.4 DisplayPort HPD寄存器

本节详细介绍了DisplayPort HPD寄存器。有关器件的完整寄存器映射的概述，请参见第19页的第4.0节“寄存器映射”。

表14-1: DISPLAYPORT HPD寄存器映射

地址	寄存器名称（符号）
0C00h	HPD控制寄存器（HPD_CTL）
0C01h	HPD中断状态寄存器（HPD_INT_STS）
0C02h	HPD中断允许寄存器（HPD_INT_EN）
0C03h	HPD事件队列寄存器（HPD_QUEUE）
0C04h	IRQ_HPDP最短时间寄存器（IRQ_HPDP_MIN_TIME）
0C05h	IRQ_HPDP最长时间寄存器（IRQ_HPDP_MAX_TIME）
0C06h	HPD高电平检测时间寄存器（HPD_HIGH_DET_TIME）
0C08h	HPD低电平检测时间寄存器（HPD_LOW_DET_TIME）
0C0Ah	HPD IRQ生成时间寄存器（HPD_IRQ_GEN）
0C0Bh - 0FFFh	保留供将来扩展使用

注： 在任何情况下都不能对保留的地址空间执行写操作。否则，可能导致意外的操作与结果。

14.4.1 HPD控制寄存器（HPD_CTL）

地址： 0C00h 大小： 8位

Bit	说明	类型	默认值
7	HPD 状态 提供HPD引脚的状态。 该位在HPD使能清零时复位为0b。 注： IRQ_HPDP事件被记录为HPD_HIGH状态。 注： 当HP配置为输入或输出时，该位有效。	RO	-
6:4	保留	RO	-
3	HPD 输出值 该位只有在将HPD引脚配置为输出（根据HPD配置）时才有意义。否则，它将始终读为0b。	R/W	0b
2	生成IRQ 置1时，HPD引脚通过将HPD置为无效且持续HPD IRQ生成时间寄存器（HPD_IRQ_GEN）中定义的时间来生成HPD IRQ。 该位只有在将HPD引脚配置为输出（根据HPD配置）时才有意义。否则，它将始终读为0b。 注： 在使用此功能前，HPD输出值必须设为1b。 注： HPD使能设为1b时，软件可能不会更改此位的状态。	R/SC	0b
1	HPD 配置 0：HPD配置为输入 1：HPD配置为输出 注： HPD使能设为1b时，软件可能不会更改此位的状态。	R/W	0b
0	HPD 使能 置1时，使能HPD引脚。	R/W	0b

14.4.2 HPD中断状态寄存器（HPD_INT_STS）

地址：0C01h 大小：8位

HPD控制寄存器（HPD_CTL）的HPD使能位清零时，该寄存器的内容复位为00h。

BIT	说明	类型	默认值
7:4	保留	RO	-
3	QUEUE_NOT_EMPTY 指示HPD事件队列寄存器（HPD_QUEUE）非空。 注： 此中断在读取HPD事件队列寄存器（HPD_QUEUE）后被清除。 注： 该位只有在将HPD引脚配置为输入时才有意义。	RO	0b
2	HPD_HIGH 置1时，指示已检测到HPD从低电平到高电平的切换。 注： 该输入来自一个脉冲信号，一直持续到该位清零。 注： 该位只有在将HPD引脚配置为输入时才有意义。	R/WC	0b
1	HPD_LOW 置1时，指示已检测到HPD从高电平到低电平的切换。 注： 该输入来自一个脉冲信号，一直持续到该位清零。 注： 该位只有在将HPD引脚配置为输入时才有意义。	R/WC	0b
0	IRQ_HPDP 置1时，指示已检测到HPD IRQ。 注： 该输入来自一个脉冲信号，一直持续到该位清零。 注： 将HPD引脚配置为输入时，该位指示已检测到IRQ_HPDP。将HPD引脚配置为输出时，该引脚指示已完成生成IRQ_HPDP的请求。	R/WC	0b

14.4.3 HPD中断允许寄存器（HPD_INT_EN）

地址：0C02h 大小：8位

BIT	说明	类型	默认值
7:4	保留	RO	0h
3:0	HPD中断允许 为“0”时，可阻止产生相应的中断。	R/W	0h

14.4.4 HPD事件队列寄存器（HPD_QUEUE）

地址：0C03h 大小：8位

该寄存器实现了用于记录器件检测到的HPD事件的循环队列。

检测到事件时，会将其写入下一个可用的HPD事件状态字段，最初从退出复位的HPD事件0开始。

读取该寄存器后，当软件将01b写入相应存储单元时，将清除具有有效HPD事件字段的所有条目。器件将跟踪最后一个有效条目，并将下一个接收到的事件存储到下一条目中。例如，如果条目0和1有效，则无论条目0和1是否已被软件清除，器件都将始终将下一事件置于存储单元2中。

器件只会将条目添加至空字段。写入HPD事件3后，器件会将下一事件写入HPD事件0（如可用）。

每个字段的编码如下。

- 00b: HPD事件字段为空
- 01b: 检测到HPD_HIGH
- 10b: 检测到HPD_LOW
- 11b: 检测到HPD_IRQ

将HPD使能设为0b和/或将HPD配置设为1b时，此寄存器将始终读为00h。

在记录四个HPD事件后，任何后续HPD事件都应被忽略，直至软件释放HPD事件存储单元。

BIT	说明	类型	默认值
7:6	HPD事件3	R/WC	00b
5:4	HPD事件2	R/WC	00b
3:2	HPD事件1	R/WC	00b
1:0	HPD事件0	R/WC	00b

14.4.5 IRQ_HPD最短时间寄存器（IRQ_HPD_MIN_TIME）

地址：0C04h 大小：8位

BIT	说明	类型	默认值
7:0	IRQ_HPD最短时间 该字段定义了为识别IRQ事件而必须将HPD置为无效的最短时间。如果置为无效的时间短于此时间，则会被忽略。 设定的时间为(50 μs * IRQ_HPD_MIN_TIME) + 100 μs。 注： HPD使能设为1b时，软件可能不会更改此寄存器的状态。	R/W	0h

14.4.6 IRQ_HPD最长时间寄存器（IRQ_HPD_MAX_TIME）

地址：0C05h 大小：8位

BIT	说明	类型	默认值
7:0	IRQ_HPD最长时间 该字段定义了为识别IRQ事件而应将HPD置为无效的最长时间。如果置为有效的时间长于此时间，则不能将其确定为HPD中断事件。 设定的时间为(50 μs * IRQ_HPD_MAX_TIME) + 100 μs。 注： HPD使能设为1b时，软件可能不会更改此寄存器的状态。	R/W	0h

14.4.7 HPD高电平检测时间寄存器（HPD_HIGH_DET_TIME）

地址：0C06h 大小：16位

BIT	说明	类型	默认值
15:0	HPD高电平检测时间 该字段定义了为检测HPD_HIGH事件HPD在初始低电平状态后必须置为有效的时间。 设定的时间为(50 μs * HPD_HIGH_DET_TIME) + 100 μs。 注： HPD使能设为1b时，软件可能不会更改此寄存器的状态。	R/W	0h

14.4.8 HPD低电平检测时间寄存器（HPD_LOW_DET_TIME）

地址：0C08h 大小：16位

BIT	说明	类型	默认值
15:0	HPD 低电平检测时间 该字段定义了为检测HPD_LOW事件 HPD 在初始高电平状态后必须置为无效的时间。 设定的时间为 $(50 \mu s * \text{HPD_LOW_DET_TIME}) + 100 \mu s$ 。 注： HPD 使能 设为1b时，软件可能不会更改此寄存器的状态。	R/W	0h

14.4.9 HPD IRQ生成时间寄存器（HPD_IRQ_GEN）

地址：0C0Ah 大小：8位

BIT	说明	类型	默认值
7:0	HPD IRQ生成时间 该字段定义在 HPD控制寄存器（HPD_CTL） 的生成 IRQ 位置1后 HPD 应置为无效的时间，单位为50 μs 。 经过此时间后， HPD 引脚将返回到 HPD输出值 中定义的状态。 注： HPD 使能 设为1b时，软件可能不会更改此寄存器的状态。	R/W	0h

15.0 看门狗定时器（WDT）

15.1 概述

看门狗定时器（Watchdog Timer，WDT）的功能是提供一种检测器件是否发生故障的机制。

使能后，如果用户程序无法在称为**WDT 间隔**的指定时间长度内重载WDT，看门狗定时器电路将产生WDT启动的系统复位。

看门狗定时器依靠20 kHz保活振荡器或48 MHz张弛振荡器工作，具体取决于**看门狗控制寄存器（WDT_CTL）**的**WDT_UNITS**位选择的分辨率。

看门狗定时器启动的系统复位通过将**中断允许寄存器（INT_EN）**的**WDT_INT**位置1进行指示。为将**WDT_INT**位清零，**看门狗控制寄存器（WDT_CTL）**的**WDT_STS**位必须清零。

15.2 WDT操作

15.2.1 WDT激活机制

WDT在正常工作期间通过以下操作序列激活：

1. 将**看门狗控制寄存器（WDT_CTL）**的**WDT_EN**位清零。
2. 用计数值装载**看门狗装载寄存器（WDT_LOAD）**。
3. 将**看门狗控制寄存器（WDT_CTL）**的**WDT_EN**位置1。

WDT激活机制启动WDT递减计数器。

15.2.2 WDT禁用机制

通过将**看门狗控制寄存器（WDT_CTL）**的**WDT_EN**位清零禁用WDT。**WDT重载机制**将WDT置于低功耗状态，在该状态下时钟进行门控，计数器停止递减。

15.2.3 WDT重载机制

如果WDT未在短于设定的看门狗时间间隔的时间内重载，则WDT将下溢，将产生WDT复位，并且**看门狗控制寄存器（WDT_CTL）**的**WDT_STS**位将置1。

软件负责持续执行重载看门狗定时器（WDT）的代码段，这会导致通过**WDT激活机制**或写入**喂看门狗寄存器（WDT_KICK）**重载计数器。

15.2.4 WDT间隔

WDT间隔是WDT从**看门狗装载寄存器（WDT_LOAD）**值递减至0h所用的时间。

15.3 I2C/SPI写入

当看门狗中断待处理时，将阻止所有写入操作，直至**看门狗控制寄存器（WDT_CTL）**的**WDT_STS**位清零。

15.4 看门狗定时器寄存器

本节详细介绍了看门狗定时器寄存器。有关器件的完整寄存器映射的概述，请参见第19页的第4.0节“寄存器映射”。

表15-1: 看门狗定时器寄存器映射

地址	寄存器名称（符号）
3000h	看门狗控制寄存器（WDT_CTL）
3002h	喂看门狗寄存器（WDT_KICK）
3004h	看门狗计数寄存器（WDT_COUNT）
3008h	看门狗装载寄存器（WDT_LOAD）
300Ch - 33FFh	保留供将来扩展使用

注： 在任何情况下都不能对保留的地址空间执行写操作。否则，可能导致意外的操作与结果。

15.4.1 看门狗控制寄存器（WDT_CTL）

地址： 3000h 大小： 16位

Bit	说明	类型	默认值
15:4	保留	RO	0h
3	WDT_UNITS 0：使用1 ms单位。 1：使用1 μs单位。 注： 使用1 μs单位时，必须使能48 MHz张弛振荡器，器件的电流消耗将增加超过400 μA。否则，当单位是1 ms时，使用20 kHz的保活振荡器。	R/W	0b
2	WDT_SIM 用于加速看门狗定时器以进行模拟。置1时看门狗工作在1 MHz时钟条件下。 该位置1时，WDT_UNITS应设为1b。	R/W	0b
1	WDT_STS 如果器件的最后一次复位由WDT下溢而引起，则WDT状态由硬件置1。该位必须由软件通过向该位写入1来清零。向该位写入0没有任何作用。	R/WC	注 15-1
0	WDT_EN 置1时，使能看门狗定时器。将该位清零将禁止看门狗定时器。 注： 出于时钟域同步的原因，可能会经过几个时钟周期后才会使此过程生效，以及使WDT计数器开始递减。	R/W	0b

注 15-1 默认值取决于是否发生WDT启动的复位，发生复位时值为1b。否则，值为0b。

15.4.2 喂看门狗寄存器（WDT_KICK）

地址：3002h 大小：8位

Bit	说明	类型	默认值
7:1	保留	RO	0h
0	KICK 向该寄存器写入任何值都会导致看门狗定时器立即重载看门狗装载寄存器（WDT_LOAD）定义的值，并在看门狗控制寄存器（WDT_CTL）的WDT_EN位设为1b时开始递减。WDT_EN为0b时写入此寄存器无任何影响。 注： 读取该寄存器时将始终返回0h。 注： 出于时钟域同步的原因，可能会经过几个时钟周期后才会使此过程生效，以及使计数器重载。	R/W	0h

15.4.3 看门狗计数寄存器（WDT_COUNT）

地址：3004h 大小：32位

Bit	说明	类型	默认值
31:0	WDT_COUNT 此只读寄存器提供了定时器当前状态的快照。 应始终先读取低字节。读取低字节时可获取计数器的快照。这可确保在读取高字节时，计数器的状态保持同步。	RO	FFFF_FFFFh

WDT_COUNT[7:0] = 0x4h
WDT_COUNT[15:8] = 0x5h
WDT_COUNT[23:16] = 0x6h
WDT_COUNT[31:24] = 0x7h

15.4.4 看门狗装载寄存器（WDT_LOAD）

地址：3008h 大小：32位

Bit	说明	类型	默认值
31:0	WDT_LOAD 对该寄存器进行写操作将重载看门狗定时器计数器。该寄存器只能在WDT_EN = 0时进行设定。 零为无效装载值。	R/W	FFFF_FFFFh

WDT_LOAD[7:0] = 0x8h
WDT_LOAD[15:8] = 0x9h
WDT_LOAD[23:16] = 0xAh
WDT_LOAD[31:24] = 0xBh

16.0 工作特性

16.1 绝对最大值*

电源电压 (VS 和 VBUS) (注16-1)	-0.3V 至 +6.0V
电源电压 (VDD33IO、3V3_VBUS 和 3V3_ALW) (注16-1)	0V 至 +4.0V
输入信号引脚相对于地的正电压	+6.0V
输入信号引脚相对于地的负电压	-0.5V
存储温度	-55°C 至 +150°C
引脚温度范围	请参见 JEDEC 规范 J-STD-020
HBM ESD 性能	+/-2 kV

注16-1 通过实验室或系统电源为此器件供电时，务必保证不要超过绝对最大值，否则可能导致器件出现故障。接通或关断交流电源时，某些电源的输出端会出现电压尖峰。此外，交流电源线上的电压瞬变可能会出现在直流输出上。如果存在这种可能性，建议使用钳位电路。

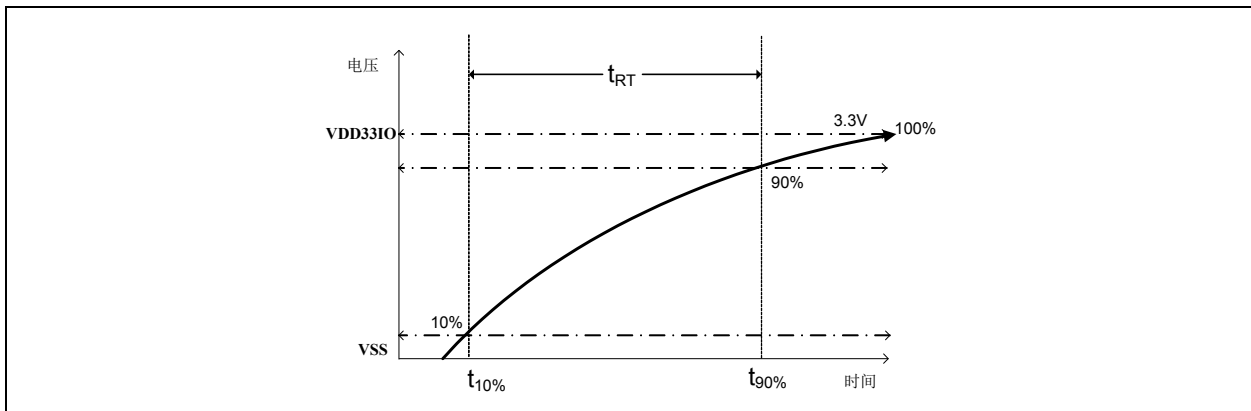
*如果器件的工作条件超过上述绝对最大值，可能对器件造成永久性损坏。上述数值仅是工作条件最大值。器件长时间工作在绝对最大值条件下，其可靠性可能受到影响。我们建议不要使器件工作在超过第16.2节“工作条件**”、第16.5节“直流特性”或本规范任何其他适用章节中规定的条件下。

16.2 工作条件**

电源电压 (VS 和 VBUS)	+4.75V 至 +5.25V
电源电压 (VDD33IO、3V3_VBUS 和 3V3_ALW)	+3.1V 至 +3.47V
输入信号引脚相对于地的正电压	+3.3V
输入信号引脚相对于地的负电压	-0.3V
电源最大上升时间 (T_{RT}) (图16-1)	100 ms
静止空气中的工作环境温度 (T_A)	0°C 至 +70°C

** 只有在本节规定的范围内才能保证器件正常工作。

图16-1： 电源上升时间模型



16.3 封装热规范

表 16-1: 封装热参数

参数	符号	°C/W
结到环境的热阻	Θ_{JA}	74.2
结到外壳顶部的热阻	Θ_{JC}	25.2
结到电路板的热阻	Θ_{JB}	47.9
结到外壳底部的热阻	Ψ_{JT}	8.8

注： 根据 JESDN51 对多层 2S2P PCB 中的各个器件测量或估算热参数。

表 16-2: 功耗

参数	符号	最大值	单位
功耗	P_{dis}	825	mW

注： 这是由于内部VBUS电源开关、VCONN电源开关、3.3V电源O形环开关、模拟模块和内核数字逻辑的最大负载（在限流保护生效之前）造成的最差情况下的功耗。

16.4 电流消耗

表 16-3: 3V3_ALW 器件电流消耗

功耗状态	3V3_ALW 电源电流		
	典型值	最大值	单位
休眠	14		μA
待机	70		μA
ACTIVE_PPC_OFF	70		μA
ACTIVE_PPC_ON	10		mA
ACTIVE 和 PD 数据包传输	+5 请参见注 4		mA

注 1: 该表详细介绍了在不同工作模式下测得的 UPD360 器件的功耗。功耗取决于温度、电源电压和外部拉/灌电流要求。最大值表示短时间内的极短活动突发。典型值表示一段时间内的平均电流消耗。

2: 休眠功耗状态通过将 PWR_DN 引脚置为有效来实现

3: 待机相当于 USB Type-C™ 规范的 Unattached.SRC/Unattached.SNK

4: 3V3_VBUS 电流约为 0 μA，但总线供电状态下除外，此时的功耗相当于“ACTIVE_PPC_OFF”。

5: ACTIVE 和 PD 数据包传输的电源电流是相应 ACTIVE_PPC_ON/OFF 状态的典型值以外的附加电流。

UPD360

表16-4: VS器件电流消耗

功耗状态	VS 电源电流		
	典型值	最大值	单位
休眠	50		μA
待机	60		μA
ACTIVE_PPC_OFF	60		μA
ACTIVE_PPC_ON	请参见注4		mA

注 1: 该表详细介绍了在不同工作模式下测得的 UPD360 器件的功耗。功耗取决于温度、电源电压和外部拉/灌电流要求。最大值表示短时间内的极短活动突发。典型值表示一段时间内的平均电流消耗。

2: 休眠功耗状态通过将 PWR_DN 引脚置为有效来实现

3: 待机相当于 USB Type-C 规范的 Unattached.SRC/Unattached.SNK

4: VS 电流消耗由 VBUS（使用内部 5V PPC 时）和 VCONN 负载两部分组成。

16.5 直流特性

表16-5: 直流电气特性

参数	符号	最小值	典型值	最大值	单位	注
IS 型输入缓冲器						
低电平输入电压	V_{ILI}	-0.3		0.8	V	
高电平输入电压	V_{IHI}	2.0		3.6	V	
下降阈值	V_{ILT}	1.21	1.33	1.8	V	施密特触发器
上升阈值	V_{IHT}	1.31	1.58	1.8	V	施密特触发器
施密特触发器迟滞电压 ($V_{IHT} - V_{ILT}$)	V_{HYS}	100	133	0	mV	
输入泄漏电流 ($V_{IN} = VSS$ 或 $VDDIO$)	I_{IH}	-10		10	μA	注 16-2
输入电容值	C_{IN}			3	pF	
O8 型输出缓冲器						
低电平输出电压	V_{OL}			0.4	V	$I_{OL} = -8 \text{ mA}$
高电平输出电压	V_{OH}	$VDDIO - 0.4$			V	$I_{OH} = 8 \text{ mA}$
OD8 型输出缓冲器						
低电平输出电压	V_{OL}			0.4	V	$I_{OL} = -8 \text{ mA}$
I2C 型缓冲器						
						注 16-3

注 16-2 此规范适用于所有输入引脚及三态双向引脚。内部下拉和上拉电阻可使每个引脚增加或减少 50 μA（典型值）。

注 16-3 I2C 型缓冲器符合 NXP I^2C 总线规范（UM10204，版本 6）。有关更多信息，请参见 I^2C 总线规范。

表 16-6: 端口电源控制器直流参数

参数	符号	最小值	典型值	最大值	单位	注
过压锁定	V_{S_OV}		6		V	
导通电阻	R_{ON_PSW}		40		mΩ	4.75V < VS < 5.25V
热关断阈值	t_{TSD}		135		°C	断开端口电源开关的裸片温度
热关断迟滞	t_{TSD_HYST}		35		°C	由于达到 T_{TSD} 而关断后, 在可以重新闭合端口电源开关之前需要的裸片温度下降量
放电阻抗	$R_{DISCHARGE}$	100			Ω	
ILIM USB 2.0 默认电流	I_{LIM_USB2}	500		600	mA	当 PWR_CAPx 引脚设置为 USB 2.0 默认电流限值或通过 PPC 控制寄存器配置时使用。
ILIM USB 3.0 默认电流	I_{LIM_USB3}	900		1100	mA	当 PWR_CAPx 引脚设置为 USB 3.0 默认电流限值或通过 PPC 控制寄存器配置时使用。
ILIM 1.5A	$I_{LIM_1.5A}$	1500		1800	mA	当 PWR_CAPx 引脚设置为 1.5A 电流限值或通过 PPC 控制寄存器配置时使用。
ILIM 3.0A	I_{LIM_3A}	3000		3700	mA	当 PWR_CAPx 引脚设置为 3.0A 电流限值或通过 PPC 控制寄存器配置时使用。

表 16-7: VCONN 源直流参数

参数	符号	最小值	典型值	最大值	单位	注
ILIM	I_{LIM_VCONN}		600		mA	VS=5V
导通电阻	R_{ON_VCONN}		270		mΩ	

表 16-8: 电源开关直流参数

参数	符号	最小值	典型值	最大值	单位	注
VSW 负载	VSW_Load			100	mA	3V3_ALW/3V3_VBUS = 3.3V
VSW 电阻	R_VSW		500		Ω	

16.6 交流特性和时序

本节详细介绍了器件的各种交流时序规范。

16.6.1 RESET_N时序

图16-2给出了RESET_N时序要求。将RESET_N置为有效并不作为一项要求。但如果使用该引脚，则必须使其在规定的最小周期内保持有效状态。

图16-2: RESET_N时序

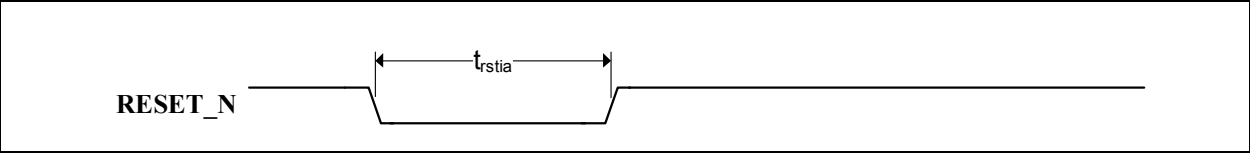


表16-9: RESET_N时序值

符号	说明	最小值	典型值	最大值	单位
t_rstia	RESET_N输入有效时间	1			μs

16.6.2 I²C从接口（仅限UPD360-A/UPD360-B）

图16-3说明了I²C从接口时序要求。I²C从接口可在标准模式、快速模式或增强型快速模式下工作。更多信息，请参见第20页的第5.0节“I²C从控制器（仅限UPD360-A/UPD360-B）”。

图16-3: I²C从器件时序

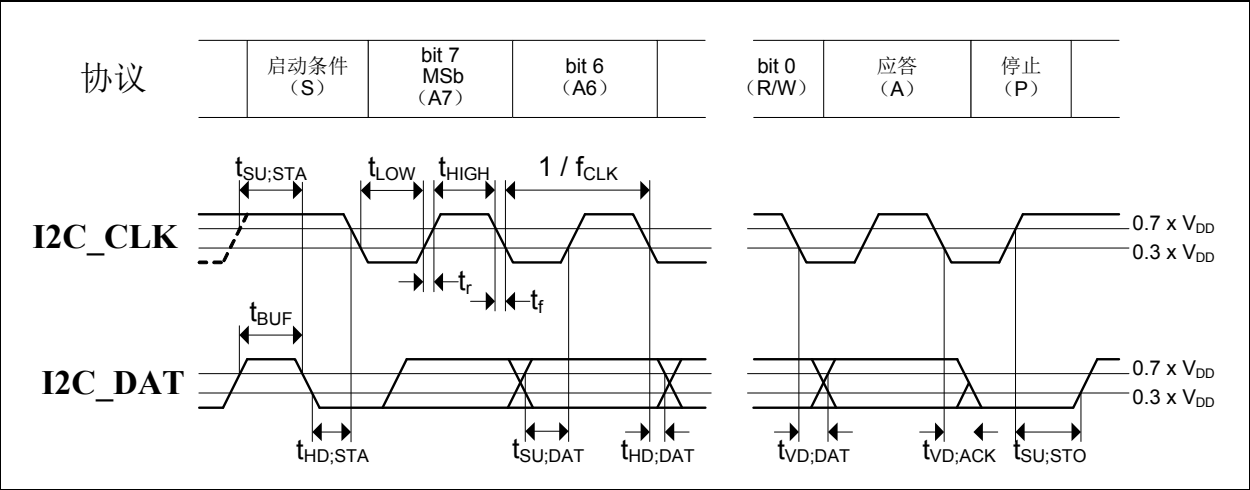


表16-10: I²C从器件时序值

符号	说明	最小值	最大值	单位
f_{CLK}	I2C_CLK时钟频率	0	1000	kHz
t_{BUF}	STOP和START条件之间的总线空闲时间	0.5		μs
$t_{HD;STA}$	(重复)启动条件的保持时间	0.26		μs
$t_{SU;STA}$	重复启动条件的建立时间	0.26		μs
$t_{SU;STO}$	停止条件的建立时间	0.26		μs
$t_{HD;DAT}$	数据保持时间	0		ns
$t_{VD;ACK}$	数据有效应答时间 (注16-4)	0.05	0.45	μs
$t_{VD;DAT}$	数据有效时间 (注16-5)	50	450	ns
$t_{SU;DAT}$	数据建立时间	50		ns
t_{LOW}	I2C_CLK时钟的低电平周期	0.5		μs
t_{HIGH}	I2C_CLK时钟的高电平周期	0.26		μs
t_f	I2C_CLK和I2C_DAT的下降时间 (注16-6) (注16-7)		120	ns
t_r	I2C_CLK和I2C_DAT的上升时间 (注16-6) (注16-7)		120	ns
t_{SP}	必须由输入滤波器抑制的尖峰的脉冲宽度 (注16-8)		50	ns

注16-4 $t_{VD;ACK}$ = 从I2C_CLK低电平到I2C_DAT (输出) 低电平的应答信号的时间。

注16-5 $t_{VD;DAT}$ = I2C_CLK低电平之后I2C_DAT数据输出有效的的时间。

注16-6 主器件必须在内部为I2C_DAT信号提供至少300 ns的保持时间 (见I2C_CLK信号的 V_{IL})，以桥接未定义区域I2C_CLK的下降沿。

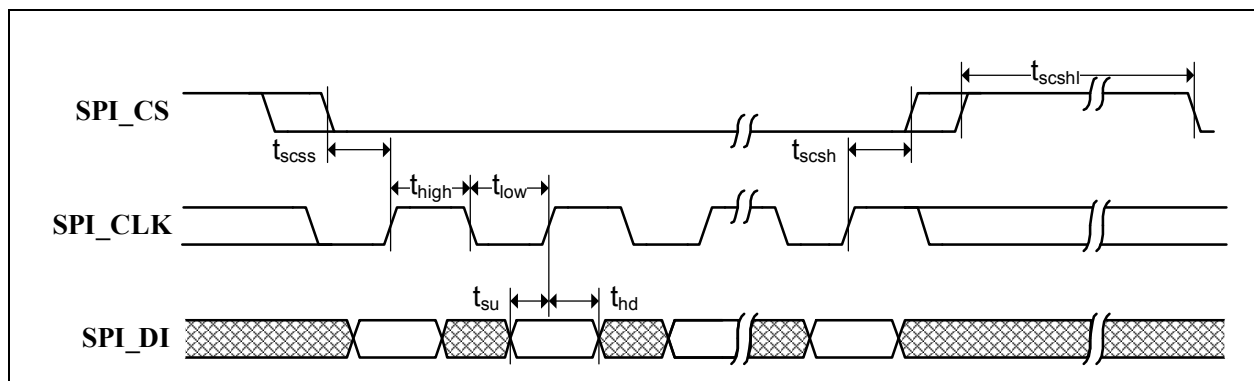
注16-7 I2C_DAT和I2C_CLK总线的最长 t_f 规定为300 ns。I2C_DAT输出阶段的最长下降时间 t_f 规定为250 ns。这样可以在I2C_DAT和I2C_CLK引脚与相应的总线之间连接串联保护电阻，而不过规定的最长值 t_f 。

注16-8 I2C_DAT和I2C_CLK输入上的输入滤波器可抑制50 ns以内的噪声尖峰。

16.6.3 SPI从接口 (仅限UPD360-C)

图16-4和图16-5分别说明了SPI从接口的输入和输出时序要求。更多信息，请参见第26页的第6.0节“SPI从控制器 (仅限UPD360-C)”。

图16-4: SPI从器件输入时序



UPD360

图 16-5: SPI从器件输出时序

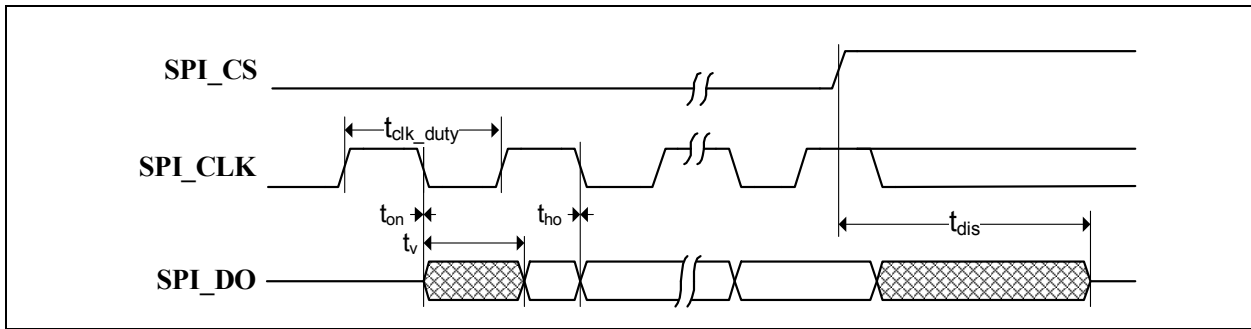


表 16-11: SPI时序值

符号	说明	最小值	典型值	最大值	单位
f_{sck}	SPI_CLK 时钟频率			25	MHz
t_{clk_duty}	SPI_CLK 高/低电平占空比	40		60	%
t_{scss}	自 SPI_CS 建立到 SPI_CLK 的时间	5			ns
t_{scsh}	自 SPI_CLK 起的 SPI_CS 保持时间	5			ns
t_{scshl}	SPI_CS 无效时间	100			ns
t_{su}	自数据输入建立到 SPI_CLK 的时间	10			ns
t_{hd}	自 SPI_CLK 起的数据输入保持时间	4			ns
t_{on}	自 SPI_CLK 起的数据输出导通时间	0			ns
t_v	自 SPI_CLK 起的数据输出有效时间			注 16-9	ns
t_{ho}	自 SPI_CLK 起的数据输出保持时间	0			ns
t_{dis}	自 SPI_CS 无效起的数据输出禁止时间			20	ns

注 16-9 8.5 或 8.0，分别对应于 30 pF 或 10 pF 的负载。

16.6.4 端口电源控制器交流参数

表 16-12: 端口电源控制器交流参数

参数	符号	最小值	典型值	最大值	单位	注
导通延时	t_{ON_PSW}		0.75		ms	PWR_EN 变为有效到开关闭合的时间，VBUS 放电无效。
关断时间	$t_{OFF_PSW_INA}$		0.75		ms	PWR_EN 变为无效到开关断开的时间。 $C_{BUS} = 120 \mu F$
关断时间	$t_{OFF_PSW_ERR}$		1		ms	过流错误、VBUS 最小值错误或放电错误到开关断开的时间。 $C_{BUS} = 120 \mu F$ 注 16-10
关断时间	$t_{OFF_PSW_ERR}$		100		ns	TSD 或反向驱动错误到开关断开的时间。 $C_{BUS} = 120 \mu F$
VBUS 输出上升时间	t_{R_BUS}		1.1		ms	从 VBUS 的 10% 测量至 90%， $C_{LOAD} = 220 \mu F$ $ILIM = 1.0A$
软导通速率	$\Delta I_{BUS}/\Delta t$		100		mA/ μs	

表16-12: 端口电源控制器交流参数 (续)

参数	符号	最小值	典型值	最大值	单位	注
温度更新时间	t_{DC_TEMP}		200		ms	
周期时间	t_{CYCLE}		25		ms	PPC故障确定已清除前的时间段, 作为自动恢复故障处理程序的一部分。
短路响应时间	t_{SHORT_LIM}		1.5		μs	从检测到短路至应用限流的时间。未施加 C_{BUS} 注 16-10
短路检测时间	t_{SHORT}		6		ms	从检测到短路至端口电源开关断开和报警置为有效的的时间。 注 16-10
锁存模式周期时间	t_{UL}		7		ms	从 PWR_EN 边沿由无效变为有效到开始错误恢复的时间。 注 16-10
放电时间	$t_{DISCHARGE}$		200		ms	应用放电电阻的时间量。可编程为 100 ms-400 ms, 列出默认值。
使用跳变模式限流时的端口电源开关操作						
区域2电流禁止值	I_{BUS_R2MIN1}			0.1	A	注 16-10
输出允许的最小VBUS	V_{BUS_MIN1}	2.0			V	注 16-10

注 16-10 仅供设计参考。未经生产测试。

17.0 封装信息

注： 最新封装图请至<http://www.microchip.com/packaging> 查看 Microchip 封装规范。

图 17-1： 封装标识信息

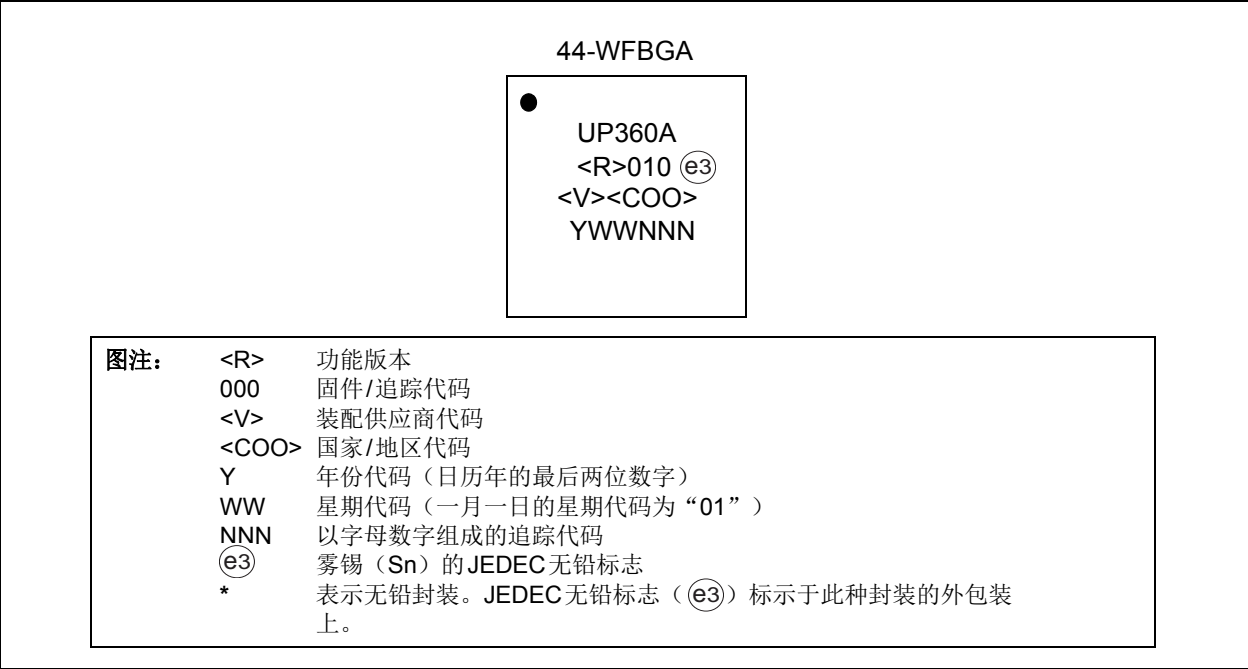
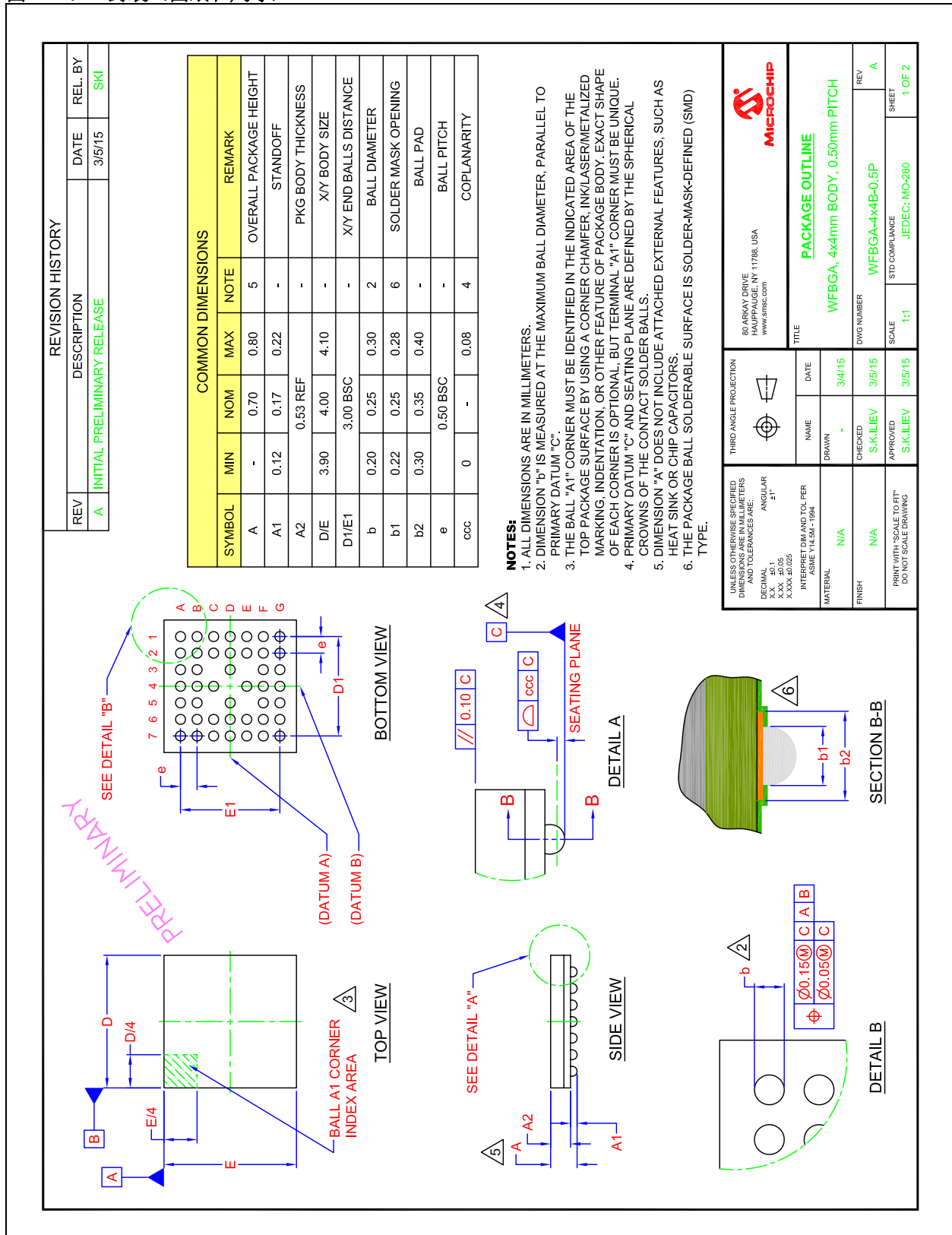


图17-2: 封装 (图纸和尺寸)



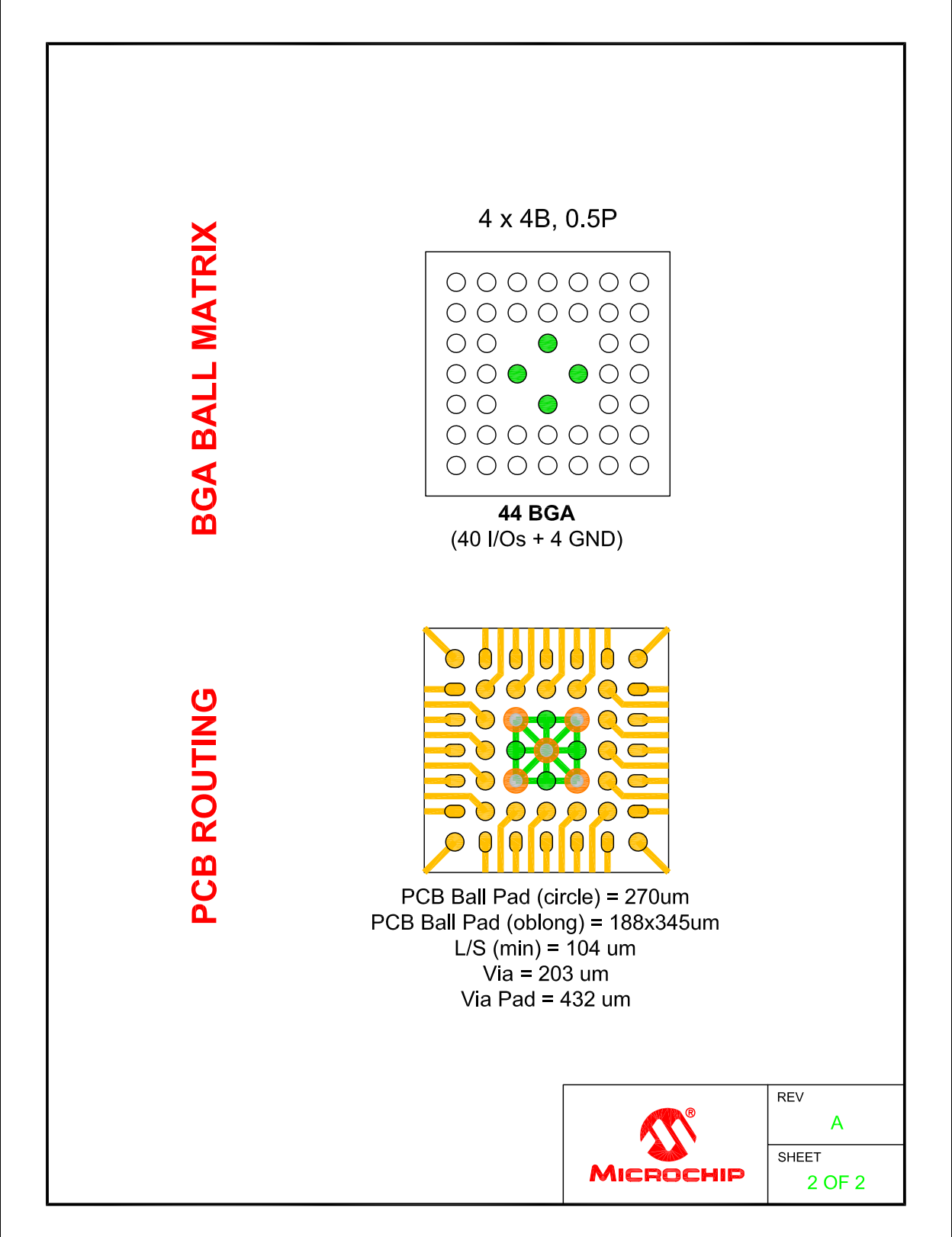
REVISION HISTORY					
REV	DESCRIPTION	DATE	REL. BY		
A	INITIAL PRELIMINARY RELEASE	3/5/15	SKI		

COMMON DIMENSIONS					
SYMBOL	MIN	NOM	MAX	NOTE	REMARK
A	-	0.70	0.80	5	OVERALL PACKAGE HEIGHT
A1	0.12	0.17	0.22	-	STANDOFF
A2	0.53 REF			-	PKG BODY THICKNESS
D/E	3.90	4.00	4.10	-	X/Y BODY SIZE
D1/E1	3.00 BSC			-	X/Y END BALLS DISTANCE
b	0.20	0.25	0.30	2	BALL DIAMETER
b1	0.22	0.25	0.28	6	SOLDER MASK OPENING
b2	0.30	0.35	0.40	-	BALL PAD
e	0.50 BSC			-	BALL PITCH
ccc	0	-	0.08	4	COPLANARITY

NOTES:

- ALL DIMENSIONS ARE IN MILLIMETERS.
- DIMENSION "b" IS MEASURED AT THE MAXIMUM BALL DIAMETER, PARALLEL TO PRIMARY DATUM "C".
- THE BALL "A1" CORNER MUST BE IDENTIFIED IN THE INDICATED AREA OF THE TOP PACKAGE SURFACE BY USING A CORNER CHAMFER, INK/LASER METALIZED MARKING, INDENTATION, OR OTHER FEATURE OF PACKAGE BODY. EXACT SHAPE OF EACH CORNER IS OPTIONAL, BUT TERMINAL "A1" CORNER MUST BE UNIQUE.
- PRIMARY DATUM "C" AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE CONTACT SOLDER BALLS.
- DIMENSION "A" DOES NOT INCLUDE ATTACHED EXTERNAL FEATURES, SUCH AS HEAT SINK OR CHIP CAPACITORS.
- THE PACKAGE BALL SOLDERABLE SURFACE IS SOLDER-MASK-DEFINED (SMD) TYPE.

图17-3: 封装（焊球矩阵和布线）



附录A： 数据手册版本历史

表A-1： 版本历史

版本	节/图/条目	修正
DS00002084C（2017年7月19日）	公开发布版本	

MICROCHIP 网站

Microchip 网站 (www.microchip.com) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的互联网浏览器即可访问。网站提供以下信息:

- **产品支持** —— 数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持** —— 常见问题解答 (FAQ)、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务** —— 产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

变更通知客户服务

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时, 收到电子邮件通知。

欲注册, 请登录 Microchip 网站 www.microchip.com。在 “支持” (Support) 下, 点击 “变更通知客户 (Customer Change Notification)” 服务后按照注册说明完成注册。

客户支持

Microchip 产品的用户可通过以下渠道获得帮助:

- 代理商或代表
- 当地销售办事处
- 应用工程师 (FAE)
- 技术支持

客户应联系其代理商、代表或应用工程师 (FAE) 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 <http://microchip.com/support> 获得网上技术支持。

产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。

部件编号	[X] ⁽¹⁾	-X	[X]	/XXX
器件	卷带式选项	版本	温度范围	封装
器件:	UPD360			
卷带式选项:	空白	= 标准包装 (托盘式)		
	T	= 卷带式 (注1)		
版本:	A	= +1.8V I ² C 接口		
	B	= +3.3V I ² C 接口		
	C	= SPI 接口		
温度范围:	空白	= 0°C 至 +70°C (商业级)		
封装:	6HX	= 44 焊球 WFBGA		

示例:

a) UPD360-A/6HX
+1.8V I²C 接口,
标准封装,
商业级温度,
44 焊球 WFBGA 封装

b) UPD360-T-B/6HX
+3.3V I²C 接口,
卷带式,
商业级温度,
44 焊球 WFBGA 封装

c) UPD360-C/6HX
SPI 接口,
标准封装,
商业级温度,
44 焊球 WFBGA 封装

注 1: 卷带式标识符仅出现在产品目录的部件编号描述中。该标识符用于订货目的, 不会印刷在器件封装上。关于包装是否提供卷带式选项的信息, 请咨询当地的 Microchip 销售办事处。

UPD360

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下，Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案（Digital Millennium Copyright Act）》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

提供本文档的中文版本仅为为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和 / 或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任，并加以赔偿。除非另外声明，在 Microchip 知识产权保护下，不得暗中以其他方式转让任何许可证。

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。Microchip 的 PIC[®] MCU 与 dsPIC[®] DSC、KEELOQ[®] 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品严格遵守公司的质量体系流程。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949 ==

商标

Microchip 的名称和徽标组合、Microchip 徽标、AnyRate、AVR、AVR 徽标、AVR Freaks、BeaconThings、BitCloud、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、Heldo、JukeBlox、KEELOQ、KEELOQ 徽标、Kleer、LANCheck、LINK MD、maXStylus、maXTouch、MediaLB、megaAVR、MOST、MOST 徽标、MPLAB、OptoLyzer、PIC、picoPower、PICSTART、PIC32 徽标、Prochip Designer、QTouch、RightTouch、SAM-BA、SpyNIC、SST、SST 徽标、SuperFlash、tinyAVR、UNI/O 及 XMEGA 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

ClockWorks、The Embedded Control Solutions Company、EtherSynch、Hyper Speed Control、HyperLight Load、IntelliMOS、mTouch、Precision Edge 和 Quiet-Wire 均为 Microchip Technology Inc. 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BodyCom、chipKIT、chipKIT 徽标、CodeGuard、CryptoAuthentication、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet 徽标、Mindi、MiWi、motorBench、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PureSilicon、QMatrix、RightTouch 徽标、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 为 Microchip Technology Inc. 在美国的服务标记。

Silicon Storage Technology 为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

GestIC 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. & KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2018, Microchip Technology Inc. 版权所有。

ISBN: 978-1-5224-2840-4

全球销售及服务网点

美洲

公司总部 **Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277

技术支持:
<http://www.microchip.com/support>

网址: www.microchip.com

亚特兰大 Atlanta
Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

奥斯汀 Austin, TX
Tel: 1-512-257-3370

波士顿 Boston
Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 Chicago
Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

达拉斯 Dallas
Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 Detroit
Novi, MI
Tel: 1-248-848-4000

休斯敦 Houston, TX
Tel: 1-281-894-5983

印第安纳波利斯 Indianapolis
Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453
Tel: 1-317-536-2380

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608
Tel: 1-951-273-7800

罗利 Raleigh, NC
Tel: 1-919-844-7510

纽约 New York, NY
Tel: 1-631-435-6000

圣何塞 San Jose, CA
Tel: 1-408-735-9110
Tel: 1-408-436-4270

加拿大多伦多 Toronto
Tel: 1-905-695-1980
Fax: 1-905-695-2078

亚太地区

中国 - 北京
Tel: 86-10-8569-7000

中国 - 成都
Tel: 86-28-8665-5511

中国 - 重庆
Tel: 86-23-8980-9588

中国 - 东莞
Tel: 86-769-8702-9880

中国 - 广州
Tel: 86-20-8755-8029

中国 - 杭州
Tel: 86-571-8792-8115

中国 - 南京
Tel: 86-25-8473-2460

中国 - 青岛
Tel: 86-532-8502-7355

中国 - 上海
Tel: 86-21-3326-8000

中国 - 沈阳
Tel: 86-24-2334-2829

中国 - 深圳
Tel: 86-755-8864-2200

中国 - 苏州
Tel: 86-186-6233-1526

中国 - 武汉
Tel: 86-27-5980-5300

中国 - 西安
Tel: 86-29-8833-7252

中国 - 厦门
Tel: 86-592-238-8138

中国 - 香港特别行政区
Tel: 852-2943-5100

中国 - 珠海
Tel: 86-756-321-0040

台湾地区 - 高雄
Tel: 886-7-213-7830

台湾地区 - 台北
Tel: 886-2-2508-8600

台湾地区 - 新竹
Tel: 886-3-577-8366

亚太地区

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733

印度 India - Bangalore
Tel: 91-80-3090-4444

印度 India - New Delhi
Tel: 91-11-4160-8631

印度 India - Pune
Tel: 91-20-4121-0141

日本 Japan - Osaka
Tel: 81-6-6152-7160

日本 Japan - Tokyo
Tel: 81-3-6880-3770

韩国 Korea - Daegu
Tel: 82-53-744-4301

韩国 Korea - Seoul
Tel: 82-2-554-7200

马来西亚 Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

马来西亚 Malaysia - Penang
Tel: 60-4-227-8870

菲律宾 Philippines - Manila
Tel: 63-2-634-9065

新加坡 Singapore
Tel: 65-6334-8870

泰国 Thailand - Bangkok
Tel: 66-2-694-1351

越南 Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

欧洲

奥地利 Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

芬兰 Finland - Espoo
Tel: 358-9-4520-820

法国 France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 Germany - Garching
Tel: 49-8931-9700

德国 Germany - Haan
Tel: 49-2129-3766400

德国 Germany - Heilbronn
Tel: 49-7131-67-3636

德国 Germany - Karlsruhe
Tel: 49-721-625370

德国 Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

德国 Germany - Rosenheim
Tel: 49-8031-354-560

以色列 Israel - Ra'anana
Tel: 972-9-744-7705

意大利 Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

意大利 Italy - Padova
Tel: 39-049-7625286

荷兰 Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

挪威 Norway - Trondheim
Tel: 47-7289-7561

波兰 Poland - Warsaw
Tel: 48-22-3325737

罗马尼亚 Romania - Bucharest
Tel: 40-21-407-87-50

西班牙 Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

瑞典 Sweden - Gothenburg
Tel: 46-31-704-60-40

瑞典 Sweden - Stockholm
Tel: 46-8-5090-4654

英国 UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820