36.USB 设备端口（UDP）

36.1 概述

USB 设备端口（UDP）适用于通用串行总线（USB）V2.0 全速设备规范。

每个端点可配置为几种 USB 传输类型中的一种。可以和双端 RAM 的一段或两段联合用于存储当前数据有效负载。如果使用两段，则处理器读写一个 DPR 段，而 USB 器件外设读写另一个。同步端点必须遵循此特性。因此器件工作于有两 DPR 段端点时维持最大带宽（1M 字节/秒）。

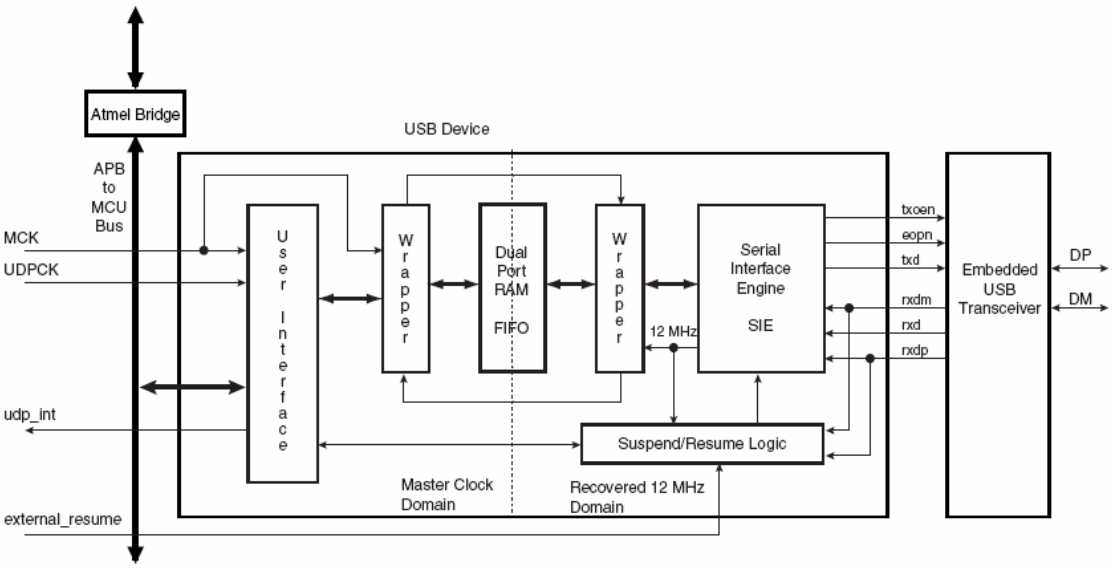
表 36-1。USB 端点说明

端点数	助记符	Dual-Bank	最大端点容量	端点类型
0	EP0	否	8	Control/Bulk/Interrupt
1	EP1	是	64	Bulk/Iso/Interrupt
2	EP2	是	64	Bulk/Iso/Interrupt
3	EP3	否	64	Control/Bulk/Interrupt
4	EP4	是	256	Bulk/Iso/Interrupt
5	EP5	是	256	Bulk/Iso/Interrupt

USB 器件自动检测挂起与恢复，通过出现中断来停止处理器。某些产品可利用外部信号唤醒 USB 主机控制器发送。

36.2 方框图

图 36-1　方框图



通过 APB 总线接口访问 UDP。通过对 APB 寄存器 8 位值读写来对数据 FIFO 进行读写。UDP 外设需要两个时钟：用于 MCK 域的外设时钟和用于 12MHz 域的 48MHz 时钟。

USB2.0 全速垫内置并且由串行接口引擎（SIE）控制。

external_resume 信号可选。它允许在系统模式下唤醒 UDP 外设。接着主机通知请求恢复的器件。列举时，此特性必须由主机处理。

36.3　附属产品

更多细节请看 USB 设备硬件实现，见产品属性文件说明。

USB 物理收发器集成在产品中。双向差分信号 DP 和 DM 对于产品边界有效。

一个 I/O 口可被用于检查来自主机的 VBUS 是否仍然有效。可使用此入口通知自供电设备主机断电。此时，必须禁用 DP 上拉电阻来防止电流流入主机。此应用应该断开收发器，然后去掉上拉电阻。

36.3.1 I/O 口

PIO 控制器不控制 DP 和 DM。内置 USB 物理收发器由 USB 器件外设控制。

为保留一个检查 VBUS 的 I/O 口，必须先对 PIO 控制器编程，将该 I/O 配置为输入 PIO 模式。

36.3.2 电源管理

USB 设备外设需要 48MHz 时钟。此时钟必须由精度为±0.25%的 PLL 产生。

因此，USB 设备从电源控制器（PMC）接收两个时钟：主控时钟，MCK，用于驱动外围用户接口，UDPCK，用于连接总线 USB 信号（恢复的 12MHz 域）。

WARNING：电源管理器（PMC）中的 UDP 外围时钟必须在对 UDP 寄存器（包括 UDP_TXCV 寄存器）任意读写操作前使能。

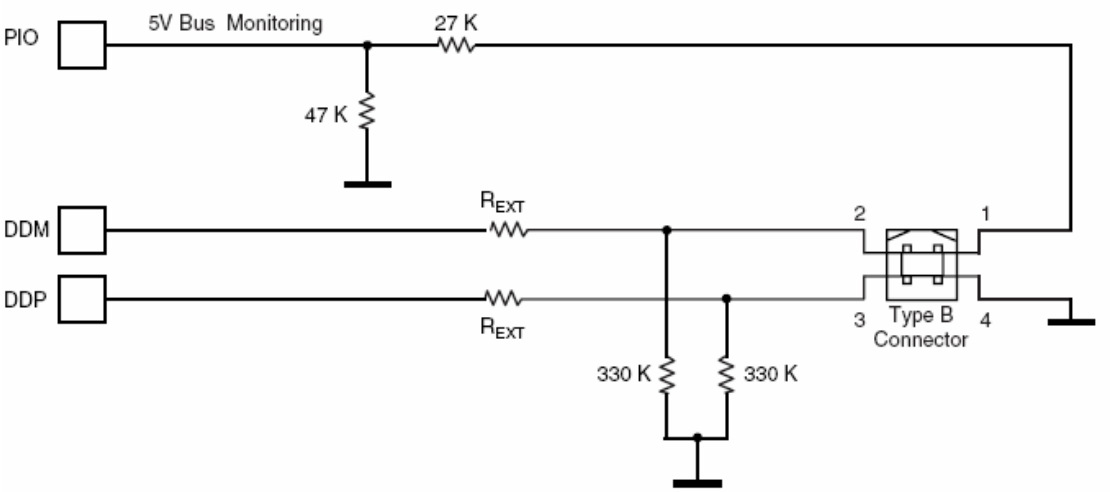
36.3.3 中断

USB 设备接口有一个连接于高级中断控制器（AIC）的中断线。

处理 USB 设备中断需要在配置 UDP 前编程 AIC。

36.4 典型连接图

图 36-2 连接器件外设的连接图



36.4.1 USB 设备收发器

USB 设备收发器内置在产品中。一些需要的分离元件如下：

- 该应用检测所有设备在第 9 章 USB 规范中定义的状态；
  ——VBUS 监控
- 为了减少功率损耗断开主机
- 口线终端

36.4.2 VBUS 监控

VBUS 监控需要检测主机连接。VBUS 监控通过使用一个可禁用的内部上拉电阻的标准 PIO。当关掉主机，则主机应该看作是断开的，必须禁用上拉电阻来防止主机通过上拉电阻寄存器耗电。

当主机断开并且使能收发器，则 DDP 和 DDM 悬空。

这可能导致过损耗。一个解决方案是在 DP 和 DM 上连接 330KΩ的下拉电阻。这些下拉电阻不改变 DDP 和 DDM 信号的完整性。

终端串行电阻必须连接到 DP 和 DM。寄存器值定义在产品(R_{EXT})的电气规范

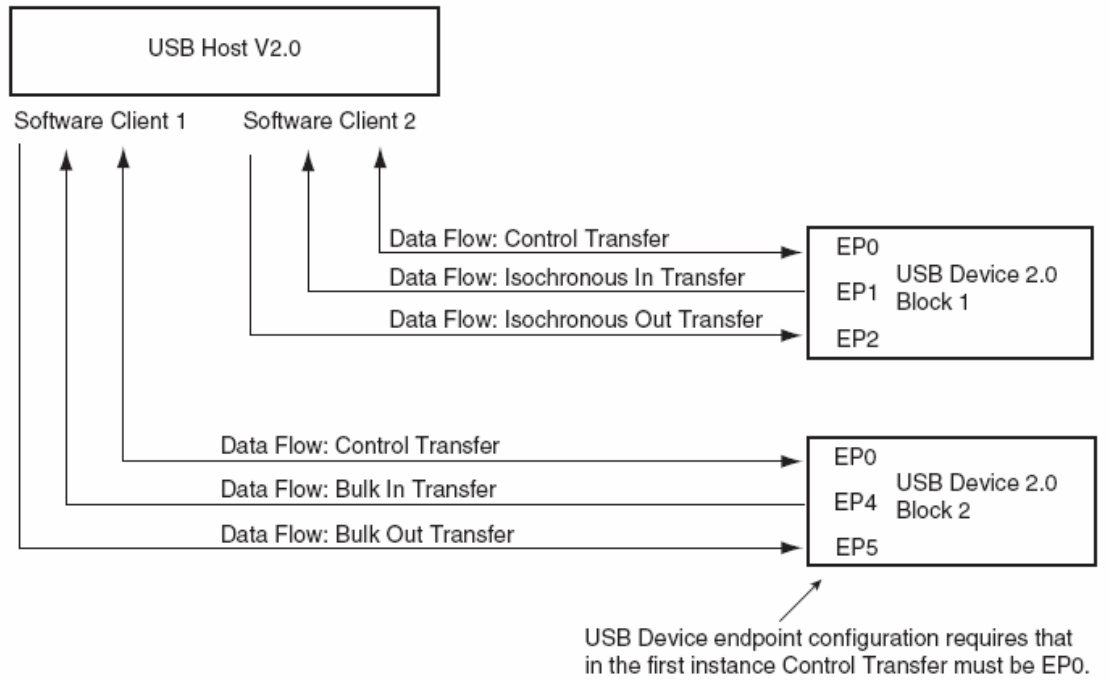
中。

36.5 功能说明

36.5.1 USBV2.0 全速介绍

USBV2.0 全速提供主机和附属 USB 器件间的通信服务。每个器件提供与每个端点相关的通信流。软件通过通信流实现主机与 USB 器件的通信。

图 36-3 USBV2.0 全速通信控制举例



当首先配置 USB 器件（USBV2.0 规范）时总是使用控制传输端点 EP0。

36.5.1.1 USBV2.0 全速传输类型

由 USB 器件自身定义的传输类型携带通信流。

表 36-2 USB 通信流

传输	方向	带宽	端点大小	错误检测	重试
控制	双向	无保证	8，16，32，64	是	自动
同步	单向	保证	256	是	否
中断	单向	无保证	8，16，32，64	是	是
批	单向	无保证	8，16，32，64	是	是

36.5.1.2 USB 总线处理

每次传送将在 USB 总线上进行一次或多次处理。总线上有以下三种数据包处理：

1.设置处理

2.数据传入处理

3.数据传出处理

36.5.1.3 USB 传输事件定义

如下表，传输在 USB 总线上顺序进行。

表 36-3 USB 传输事件

控制传输（1）（3）	设置处理>数据传入处理>状态传出处理 设置处理>数据传出处理>状态传入处理 设置处理>状态传入处理
中断进入传输（设备到主机）	数据传入处理>数据传入处理
中断输出传输（主机到设备）	数据传出处理>数据传出处理
同步传入传输（2）（主机到设备）	数据传入处理>数据传入处理
同步传出传输（2）（设备到主机）	数据传出处理>数据传出处理
批传入传输（设备到主机）	数据传入处理>数据传入处理
批传出传输（主机到设备）	数据传出处理>数据传出处理

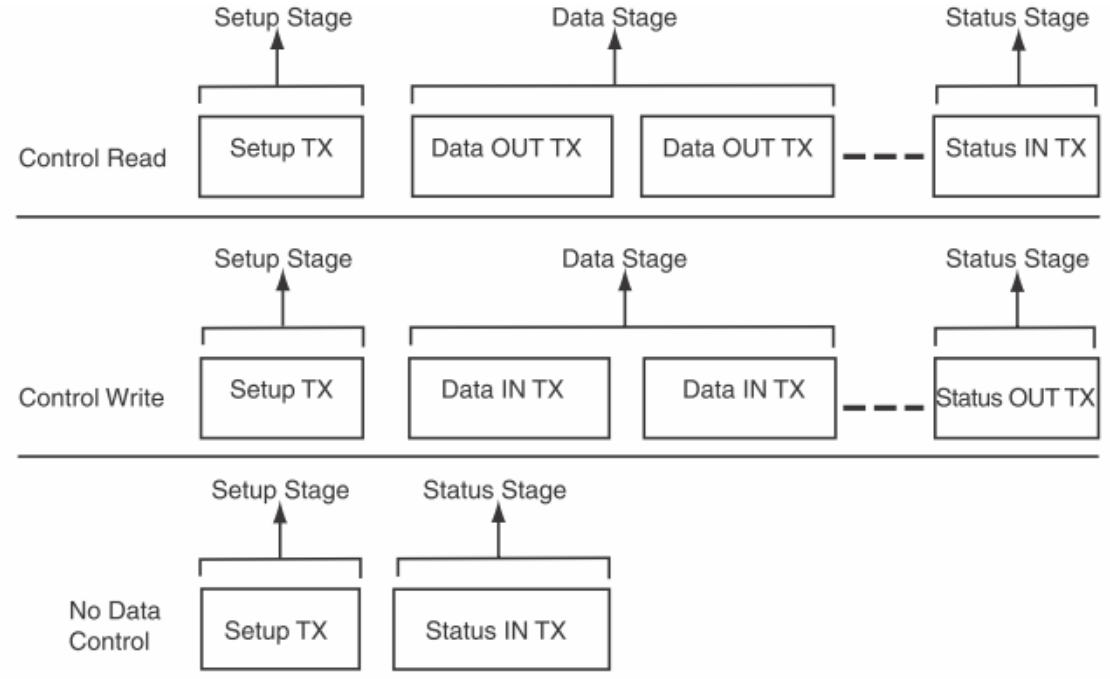
注意：1.控制传输必须使用无 ping-pang 属性的端点

2.同步传输必须使用 ping-pang 属性的端点

3.可使用延迟握手中止控制传输

状态处理是一个只用一个控制传输的主机到设备处理的特殊类型。此控制传输必须使用无 ping-pong 属性的端点。根据控制顺序（读或写），USB 设备发送或接收一次状态处理。

图 36-4 控制读写顺序



注意：1.在 Status IN 阶段，主机等待一个从设备来的使用 DATA1 PID 的长度为 0 的数据包（无数据的 Data IN 处理）。更多关于协议层的细节参考 Universal Serial Bus Specification, Rev. 2.0 第 8 章

2.在 Status OUT 阶段，主机发射一个长度为 0 的数据包到设备（无数据的 Data OUT 处理）。

36.5.2 与 USB V2.0 器件外设交互处理

36.5.2.1 设置处理

设置是在控制传输期间使用的主机到器件之间的一个特殊类型。必须使用无 ping-pong 特性的端点执行控制传输。设置处理必须尽可能快的由固件处理。它使用从主机到设备的传送请求。这些请求由 USB 设备处理并可能需要更多的变量。

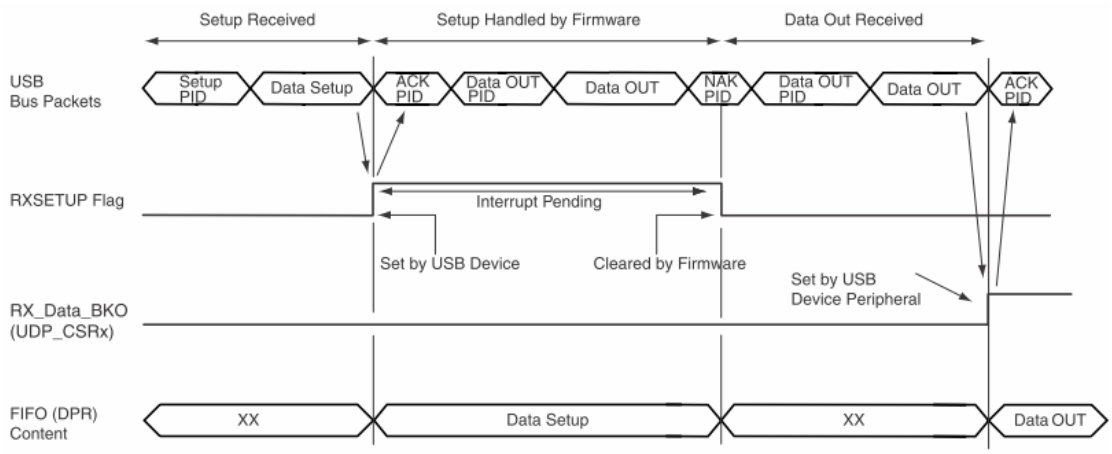
这些变量由设置处理后的 Data OUT 处理发送到器件。这些请求也可能返回数据。返回的数据被设置处理后的 Data IN 处理送入主机。状态处理结束整个控制传输。

当 USB 端点接收到一个设置传输：

- USB 器件自动应答此设置数据包
- 在 UDP_CSRx 寄存器设置 RXSETUP
- 当 RXSETUP 未被清零时产生端点中断，如果该设备中断使能，则该中断被送到微控制器执行。

因此，固件必须轮询 UDP_CSRx 检测 RXSETUP 或获取中断，读取 FIFO 中的设置数据包，然后清零 RXSETUP。在未读取 FIFO 中的设置数据包前不能清零 RXSETUP。否则，USB 设备将接收下一次 Data OUT 传输并覆盖 FIFO 中的设置数据包。

图 36-5 设置处理和 Data OUT 处理



36.5.2.2 Data IN 处理

Data IN 处理被用在控制传输，同步传输，批传输和中断传输，并引导数据从设备到主机的传输。同步传输中的 Data IN 处理必须使用 ping-pong 属性的端点。

36.5.2.3 使用无 ping-pong 属性的端点

使用无 ping-pong 属性执行 Data IN 处理：

1.应用程序检查是否通过轮询端点 UDP_CSRx 寄存器的 TXPKTRDY 对 FIFO 写入 (TXPKTRDY 必须清零)。

2.应用程序将准备发送的第一个数据写入端点的 FIFO 中，给端点的 UDP_FDRx 寄存器写入 0 或多字节值。

3.应用程序通过设置端点的 UDP_CSRx 寄存器中的 TXPKTRDY 通知 USB 外设它已完成。

4.当端点的 UDP_CSRx 寄存器中的 TXCOMP 被置位时应用程序通知端点的 FIFO 已经被 USB 器件释放。然后当 TXCOMP 置位时相应端点的中断被挂起。

5.微处理器将准备发送的第二个数据包写入到端点 FIFO，在端点 UDP_

FDRx 寄存器写入 0 或多字节值。

6.微控制器通过设置端点的 UDP_CSRx 寄存器中的 TXPKTRDY 通知 USB 外设已完成。

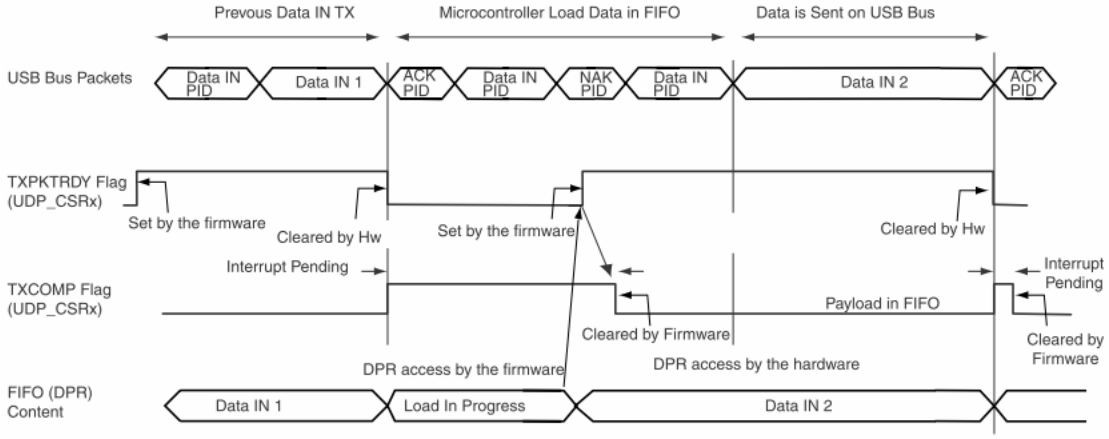
7.应用程序清零端点的 UDP_ CSRx 中的 TXCOMP。

最后一个数据包被发送后，只要 TXCOMP 被置位则应用程序必须清零 TXCOMP。当接收到一个 Data IN 数据包的 ACK PID 信号时由 USB 器件置位 TXCOMP。当 TXCOMP 置位时中断挂起。

警告：TX_PKTRDY 被置位后必须清零 TX_COMP。

注意：关于 Data IN 协议层的更多细节请参考 Universal Serial Bus Specification, Rev 2.0 的第 8 章。

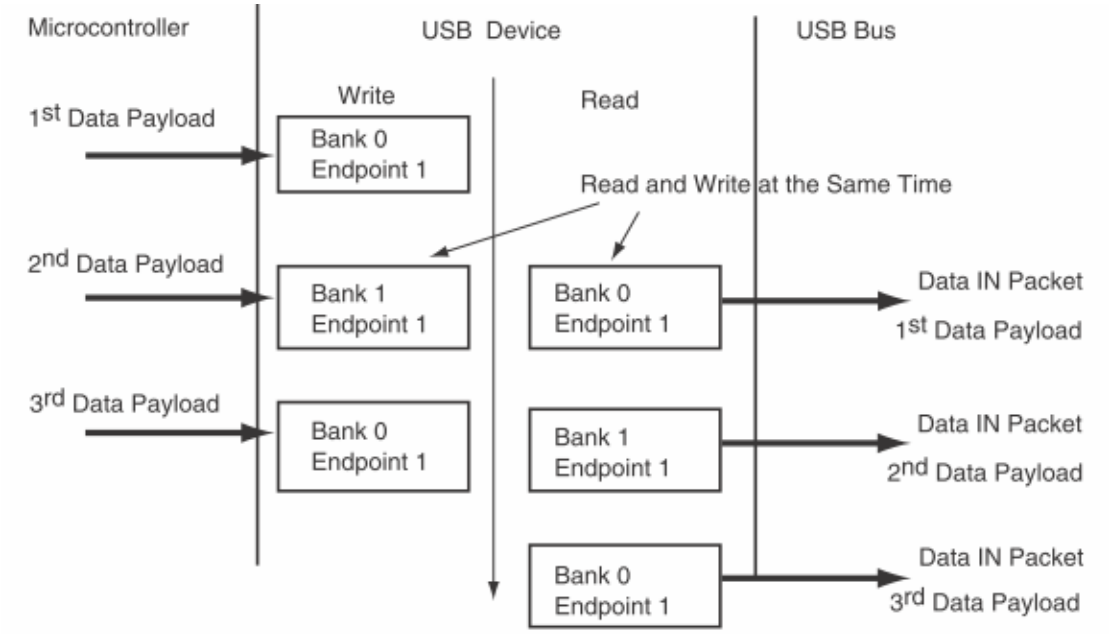
图 36-6 无 Ping- pong 端点的 Data IN 传输



36.5.2.4 使用 ping_pong 属性的端点

在同步传输期间需要使用到 ping_pong 属性的端点。也允许在此传输期间处理 USB 规范定义的最大带宽。为保证带宽为常数或最大带宽，在当前数据被 USB 器件发送的同时微控制器必须准备下一个要发送的数据的有效负载。因此要使用两个存储器段。一个给微控制器，另一个由 USB 器件锁定。

图 36-7 ping_pong 端点 Data IN 传输的段交换



当使用一个 ping-pong 端点，执行 Data IN 处理时需要以下程序：

1.微控制器通过轮询端点 UDP_CSRx 寄存器的 TXPKTRDY 是否清零来检查是否能对 FIFO 写入 。

2.微控制器将准备发送的第一个数据的有效负载写入端点的 FIFO 中，给端点 UDP_FDRx 寄存器写入 0 或多字节值。

3.微控制器通过设置端点的 UDP_CSRx 寄存器中的 TXPKTRDY 通知 USB 外设，它已完成对 FIFO(bank 0）的写入。

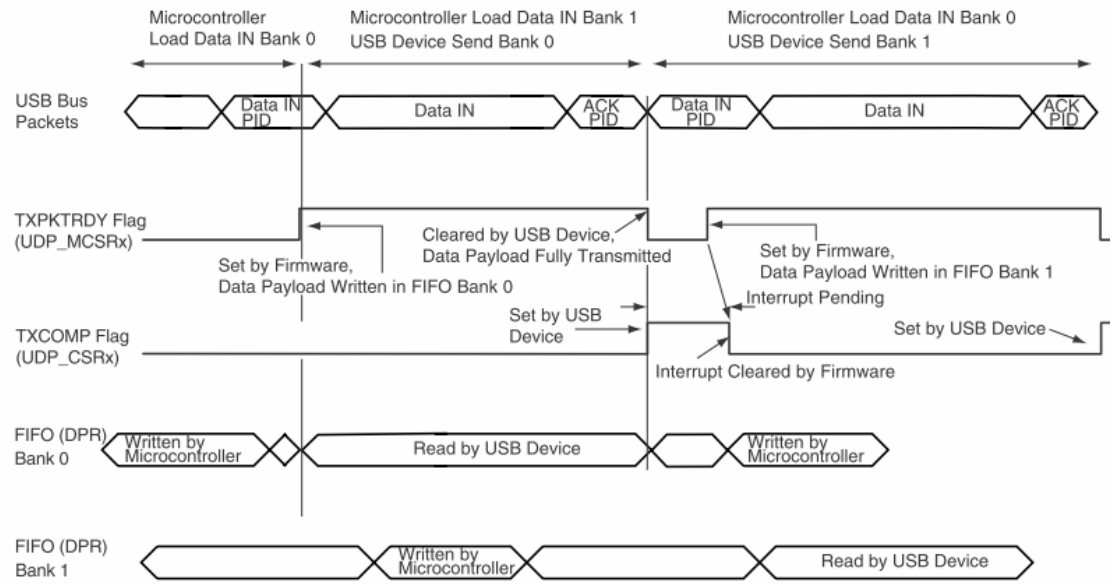
4.不必等待 TXPKTRDY 被清零，微控制器将准备发送的第二个数据的有效负载写入到 FIFO（bank 1），对端点 UDP_ FDRx 寄存器写入 0 或多个字节值。

5.当端点 UDP_ CSRx 寄存器中的 TXCOMP 置位时，由 USB 器件通知微控制器已经释放第一个存储器段。当 TXCOMP 被置位则中断挂起。

6.一旦微控制器接收到第一个存储器段的 TXCOMP 时，则拉高端点 UDP_ CSRx 寄存器的 TXPKTRDY 通知 USB 器件已准备好将要发送的第二个存储段。

7.此步骤中，Bank 0 可用并且微控制器可以准备将要发送的第三个数据的有效负载。

图 36-8 ping-pong 端点的 Data IN 传输



警告：这是软件关键路径，由于一旦第二个段已满，则驱动器不得不等待 TX_COMP 置位 TX_PKTRDY。如果在接收 TX_COMP 置位 和 TX_PKTRDY 置位之间的延迟太长，则一些 Data IN 数据包可能无应答，减小带宽。

警告： TX_PKTRDY 被置位后必须清零 TX_COMP。

36.5.2.5 Data OUT 处理

Data OUT 处理被用在控制，同步，批和中断传输中并引导数据从主机到器件的传输。同步传输中的 Data OUT 处理必须使用 ping-pong 属性的端点。

36.5.2.6 无 ping-pong 属性的 Data OUT 处理

使用无 ping-pong 属性的端点执行 Data OUT 处理：

1.主机产生一个 Data OUT 数据包。

2.此数据包由 USB 器件端点接收。当与该端点有关的 FIFO 正在被微控制器使用时，则返回一个 NAK PID 到主机。一旦 FIFO 可用，则由 USB 器件将数据写入到 FIFO 并且自动发送 ACK 到主机。

3.通过轮询端点 UDP_ CSRx 寄存器的 RX_DATA_BK0 通知微控制器 USB 器件已接收到数据的有效负载。当 RX_DATA_BK0 置位则挂起该端点的中断。

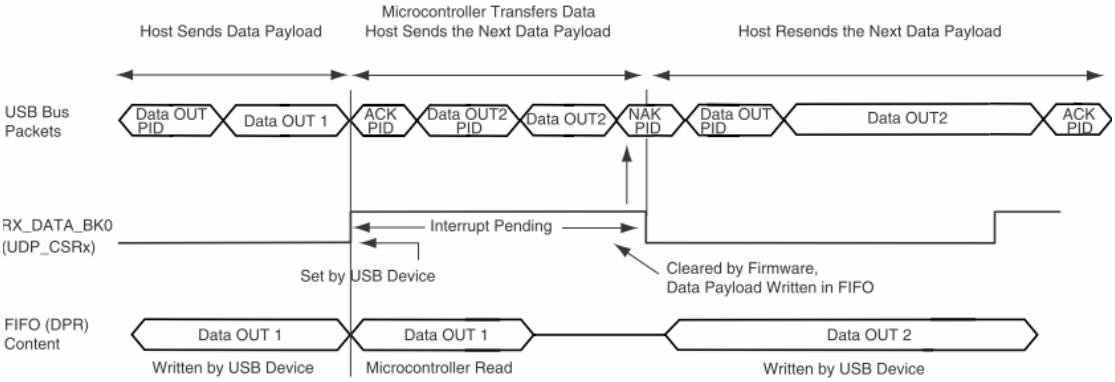
4.通过读取端点 UDP_ CSRx 寄存器的 RXBYTECNT 得到 FIFO 中的字节数。

5.微控制器接收端点存储器的数据到其存储器中。通过读取端点 UDP_ FDRx 寄存器得到接收的数据。

6.微控制器通过清零端点 UDP_ CSRx 寄存器中的 RX_DATA_BK0 通知 USB 器件已完成传输。

7.USB 器件可接收一个新的 Data OUT 数据包。

图 36-9 无 ping-pong 端点的 Data OUT 传输

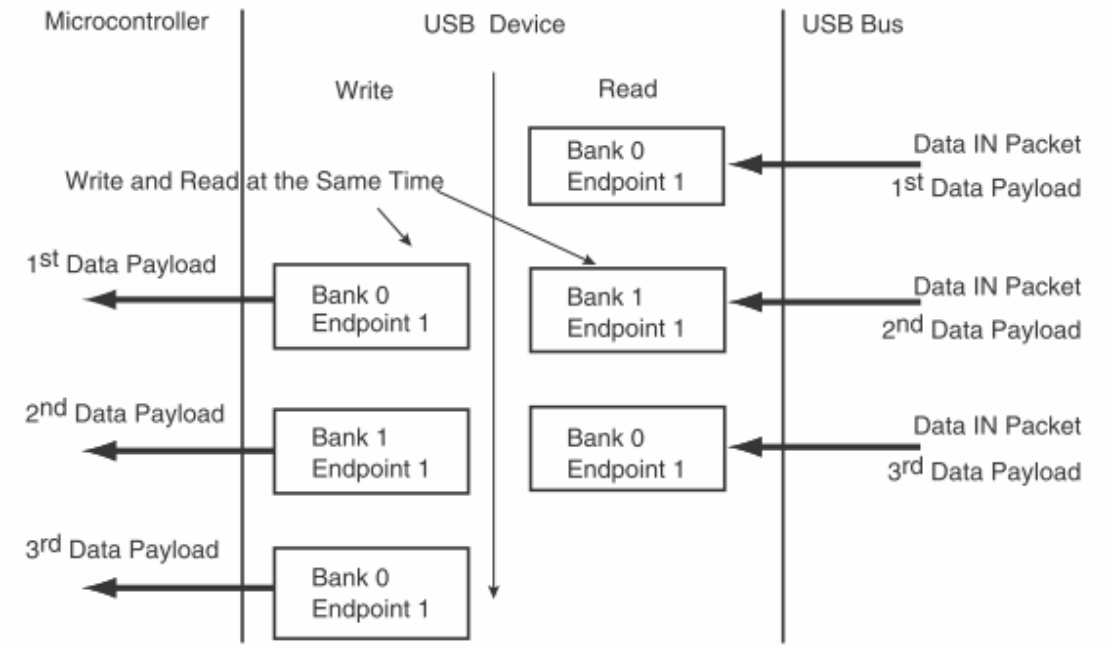


当 RX_DATA_BK0 置位时中断挂起。在 RX_DATA_BK0 被清零后，USB 器件，FIFO 和微控制器存储器间的存储器传输不能进行。否则，USB 器件将接收下一个 Data OUT 传输并且覆盖 FIFO 中当前 Data OUT 数据包。

36.5.2.7 使用 ping-pong 属性的端点

同步传输期间，必须使用 ping-pong 属性的端点。为保证一个常数带宽，当 USB 器件接收到当前数据有效负载时，微控制器必须读取由主机发送的前一个数据有效负载。所以使用两个存储器段。一个给微控制器，另一个由 USB 器件锁定。

图 36-10 ping-pong 端点 Data OUT 传输中的段交换



当使用 ping-pong 端点，执行 Data OUT 处理需要以下步骤：

1.主机产生一个 Data OUT 数据包。

2.该数据包由 USB 器件端点接收。写入到端点的 FIFO 的 Bank0.

3.USB 器件发送一个 ACK PID 数据包到主机。主机可立即发送第二个数据包。由器件接收并复制到 FIFO 的 Bank1.

4.通过轮询端点 UDP_ CSRx 寄存器的 RX_DATA_BK0 通知微控制器 USB 器件已接收到一个数据有效负载。当 RX_DATA_BK0 置位时则挂起该端点的中断。

5.通过读取端点 UDP_ CSRx 寄存器的 RXBYTECNT 得到 FIFO 中的字节数。

6.微控制器将接收到的数据从端点存储器传输到微控制器的存储器。通过读取端点 UDP_ FDRx 寄存器获得接收到的数据。

7.微控制器通过清零端点 UDP_ CSRx 寄存器的 RX_DATA_BK0 通知 USB 外设已完成传输。

8.可以由 USB 外设接收第三个 Data OUT 数据包并且复制到 FIFO 的 Bank0

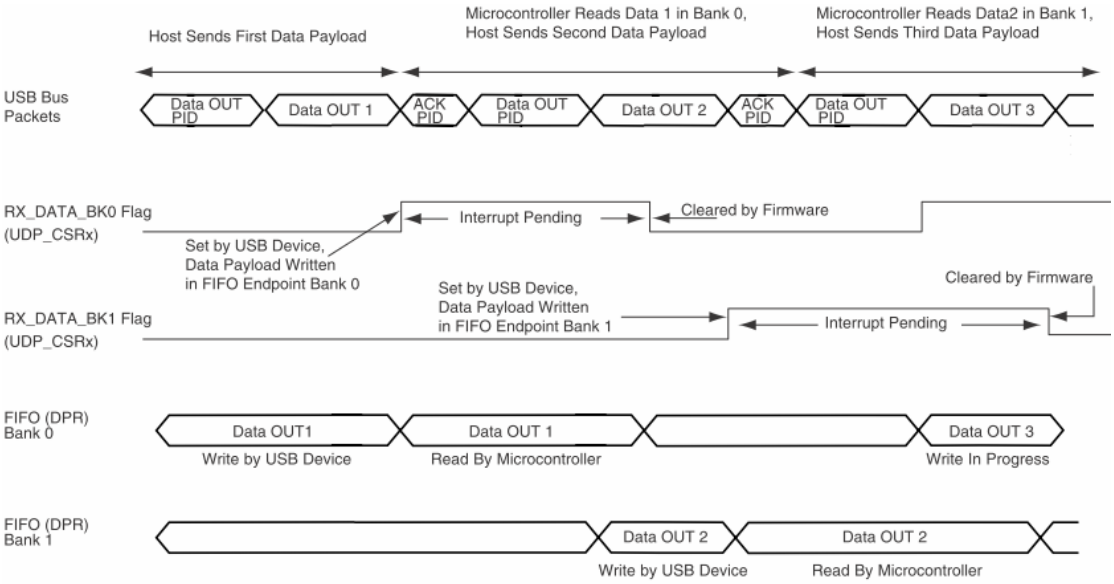
9.如果已经接收到第二个 Data OUT 数据包，通过置位端点 UDP_ CSRx 寄存器的 RX_DATA_BK1 通知微控制器。当 RX_DATA_BK1 置位时则挂起该端点的中断。

10.微控制器将接收到的数据从端点存储器传输到微控制器的存储器。可通过读取端点 UDP_ FDRx 寄存器获得接收到的数据。

11.微控制器通过清零端点 UDP_ CSRx 寄存器的 RX_DATA_BK1 通知 USB 器件已完成传输。

12.USB 器件可接收第四个 Data OUT 数据包并且复制到 FIFO 的 Bank0.

图 36-11 ping-pong 端点的 Data OUT 传输



注意：当 RX_DATA_BK0 或 RX_DATA_BK1 置位则挂起中断。

警告：当 RX_DATA_BK0 和 RX_DATA_BK1 都被置位，则无法确定首先对哪一个清零。因此必须用软件设立一个内部计数器确保能先清零 RX_DATA_BK0 然后清零 RX_DATA_BK1 。该状况可能在应用软件在别处忙且两个存储段被 USB 主机填满时出现。一旦应用程序返回 USB 驱动器，则两个标志位置位。

36.5.2.8 停止握手

停止握手可用在以下两种情形之一。(更多关于停止握手，参考 Universal Serial Bus Specification, Rev 2.0 第 8 章)

- 当和端点相关的停止特性设置时使用功能停止（更多关于停止特性，参考 Universal Serial Bus Specification, Rev 2.0 第 9 章）。
- 中端当前请求，使用协议停止，仅用于控制传输。

以下步骤产生一个停止数据包：

1.微控制器设置端点 UDP_ CSRx 寄存器的 FORCESTALL 标志位。

2.主机接收停止数据包

3.通过轮询 STALLSENT 设置来通知微控制器，器件已停止发送数据。当 STALLSENT 置位，挂起端点中断。微控制器必须清零 STALLSENT 来清除中断。

当停止握手后接到一个设置处理，必须清零 STALLSENT 以防止由于 STALLSENT 被置位产生的中断。

图 36-12 停止握手（Data IN 传输）

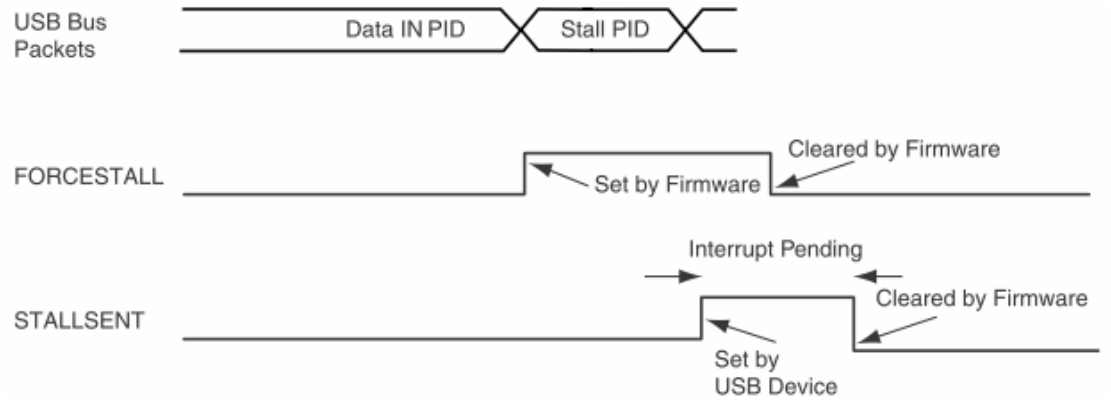
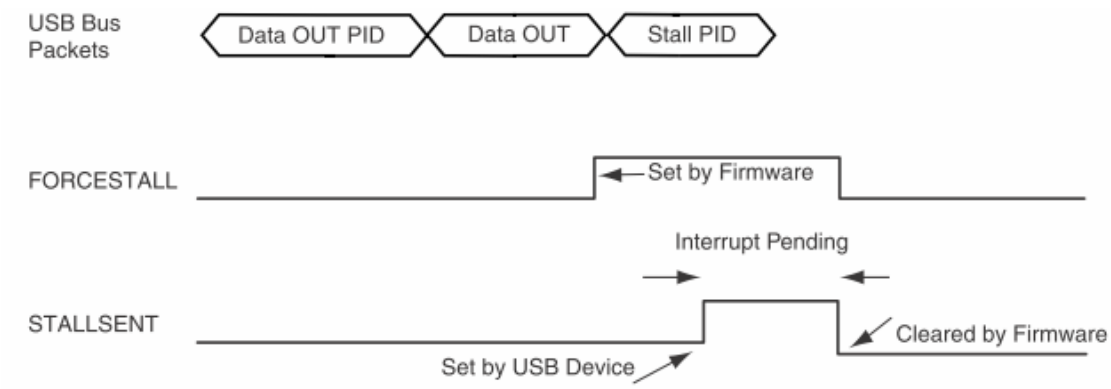


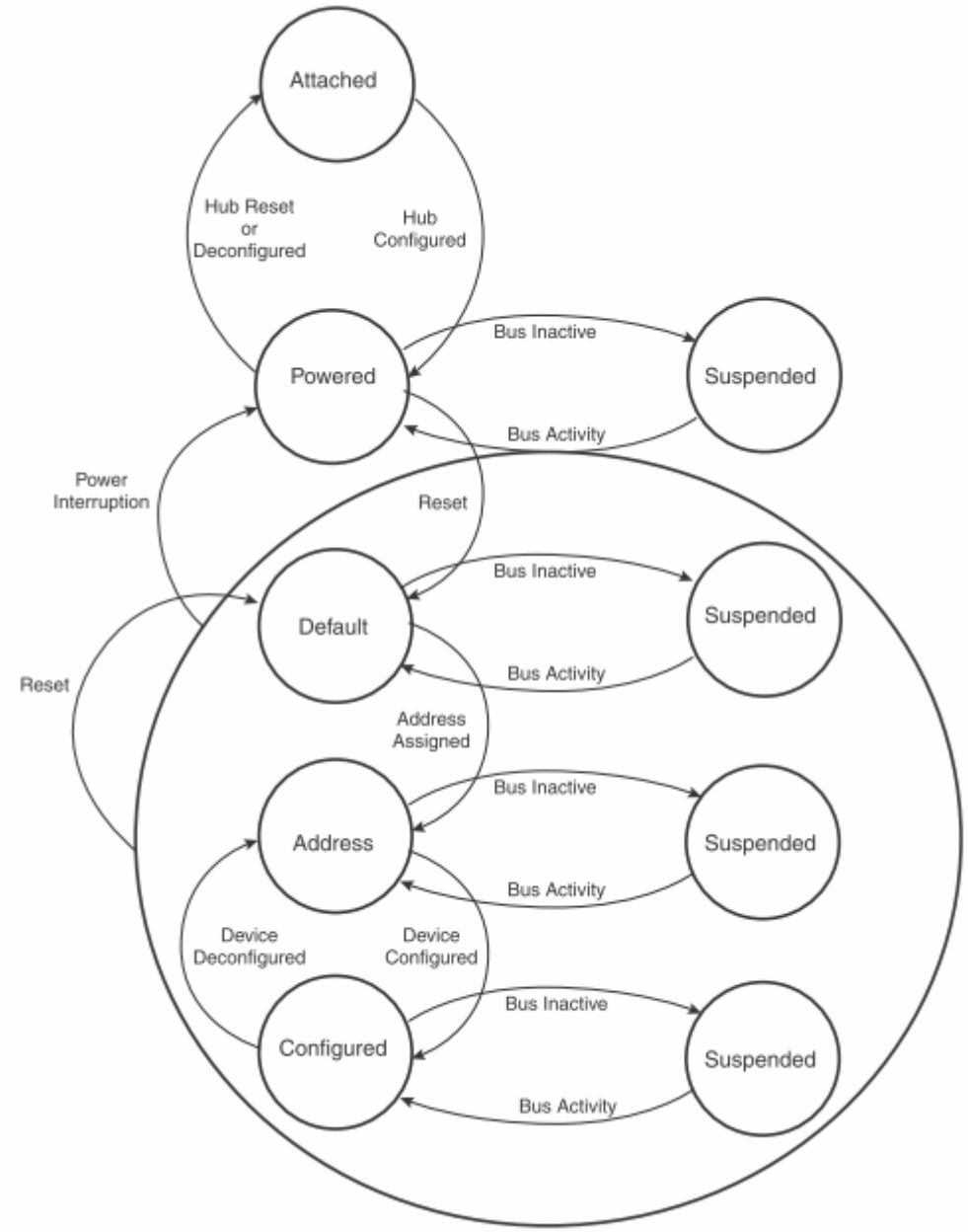
图 36-13 停止握手（Data OUT 传输）



36.5.3 控制器件状态

一个 USB 器件有几种可能的状态。参考 Universal Serial Bus Specification, Rev 2.0 第 9 章。

图 36-14 USB 器件状态框图



一种状态到另一种状态的转换取决于 USB 总线状态或通过默认端点（端点 0）的控制处理发送标准请求。

在一个总线无效周期后，USB 设备进入悬挂模式。强制从 USB 主机接受挂起/恢复请求。挂起模式对总线供电应用的要求非常严格；器件在 USB 总线上消耗不超过 500 微安。

在挂起模式，主机可通过发送一个恢复信号（总线激活）来唤醒一个器件或者 USB 器件可以发送一个唤醒请求给主机，例如，通过移动鼠标唤醒 PC。

唤醒特性并不是对所有器件都强制执行，并必须与主机协商。

36.5.3.1　无供电状态

自供电设备可使用一个称作典型连接单元的 PIO 检测到 5V VBUS。当设备未连接到主机，可通过对 UDP 禁用 MCK，禁用 UDPCK 和禁用收发器减小设备功耗。DDP 和 DDM 接口需要 330 千欧电阻下拉电阻。

36.5.3.2 进入连接状态

当未连接设备时，USB DP 和 DM 信号被通过集成在集线器下游端口的 15 千欧下拉电阻连接到 GND。当器件连接到集线器下游端口，则器件在 DP 上连接了一个 1.5 千欧的上拉电阻。USB 总线接口进入 IDLE 状态，DP 被器件的 1.5 千欧电阻上拉到 3.3V，DM 被主机的 15 千欧电阻拉低。为使能集成的上拉功能，UDP_TXVC 寄存器中的 PUON 位必须置位。要使能集成的上拉，USB_PUCR 总线矩阵寄存器的 UDP_PUP_ON 位必须置位。

警告：为能对 UDP_TXVC 寄存器写入数据，必须在 UDP 上使能 MCK 时钟。在电源管理控制器中实现。

上拉连接后，设备进入供电状态。该状态下，必须在电源管理控制器中使能 UDPCK 和 MCK。收发器可保持禁用。

36.5.3.3 从上电状态到默认状态

连接到一个 USB 主机后，USB 器件要等待一个 end-of-bus 的复位。在 UDP_ISR 寄存器中置位不可屏蔽标志位 ENDBUSRES 并且触发一个中断。

一旦 ENDBUSRES 中断已被触发，则设备进入默认状态。该状态下，UDP 软件必须：

- 使能默认端点，置位 UDP_CSR[0]寄存器中的 EPEDS 标志位，并通过向 UDP_IER 寄存器写入 1 有选择的使能端点 0 的中断。
- 配置已被 USB 复位检测器复位的中断屏蔽寄存器 。
- 清零 UDP_TXVC 寄存器中的 TXVDIS 标志位使能收发器

该状态必须使能 UDPCK 和 MCK。

警告：每次触发 ENDBUSRES 中断时，确保已复位中断屏蔽寄存器和 UDP_CSR 寄存器

36.5.3.4 从默认状态到地址状态

在设置地址标准器件请求后，USB 主机外设进入地址状态。

警告：在器件进入地址状态前，必须完成控制传输的 Status IN 处理，也就是，一旦已接收到 UDP_CSR[0] 寄存器中的 TXCOMP 标志位并被清零则 UDP 器件要设置新的地址。

为了转换到地址状态，驱动软件在 UDP_GLB_STAT 寄存器中设置 FADDEN 标志位，设置其寻址，并设置 UDP_FADDR 寄存器中的 FEN 位。

36.5.3.5 从地址状态到配置状态

一旦已接收到一个有效的设置地址标准器件请求并应答后，则器件使能相应当前配置的端点。通过设置 UDP_CSRx 寄存器中的 EPEDS 和 EPTYPE 域完成，并可使能 UDP_IER 寄存器中的相关中断（可选）。

36.5.3.6 进入挂起状态

当检测到挂起状态（在 USB 总线上无总线动作），则 UDP_ISR 寄存器中的 RXSUSP 信号置位。如果 UDP_IMR 寄存器中的相关位置位则触发中断。通过向 UDP_ICR 寄存器写入数据清零该标志位。然后设备进入挂起状态。

该状态下总线供电设备从 5V VBUS 的漏电流必须小于 500 微安。例如微控制器转换到低速时钟，禁用 PLL 和主振荡器，并进入空闲状态。还可能关闭板上其他器件。

可关闭 USB 器件外设时钟。异步检测 Resume 事件。可在电源管理控制器中关闭 MCK 和 UDPCK 并且可通过 UDP_TXVC 寄存器中的 TXVDIS 域禁用 USB 收发器。

警告：仅在 MCK 对外设使能时允许向 UDP 寄存器读，写。关闭 UDP 外设的 MCK 必须是在向 UDP_TXVC 写入并应答 RXSUSP 后的最后一个操作。

36.5.3.7　接收主机恢复

挂起模式下，异步检测 USB 总线上的恢复事件，收发器和时钟是禁用的（但是上拉仍有效）。

一旦在总线上检测到恢复事件，UDP_ISR 中的 WAKEUP 信号置位。如果 UDP_IMR 寄存器的相关位置位则可能产生中断。该中断可用于唤醒内核，使能 PLL 和主振荡器并配置时钟。

警告：仅当 UDP 外设的 MCK 使能时允许向 UDP 寄存器读写。必须在清零 UDP_ICR 寄存器中的 WAKEUP 位和 UDP_TXVC 寄存器中的 TXVDIS 位前使能 UDP 的 MCK。

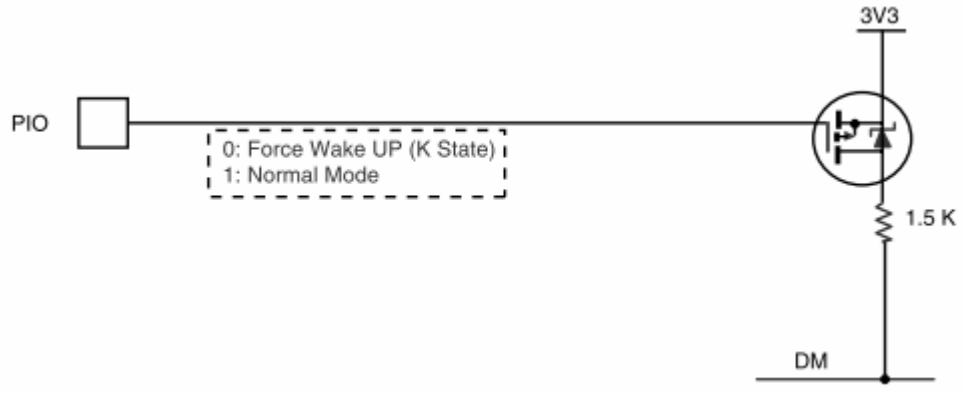
36.5.3.8　发送设备远程唤醒

挂起模式下发送一个外部恢复事件唤醒主机。

- 在器件进入挂起状态后，至少等待 5ms 才可发送外部恢复事件。
- 器件有 10ms 时间开始灌入电流并驱动 K 状态来恢复主机。
- 器件必须在 1 到 15ms 内强制一个 K 状态来恢复主机

为了向总线驱动 K 状态（DM3.3V，DP 连接到 GND），可连接一个晶体管上拉到 DM。可通过禁用 DP 的上拉和使能 DM 的上拉获得 K 状态。可在应用程序控制。

图 36-15 驱动 K 状态的原理图



36.6 USB 器件端口（UDP）用户接口

警告：必须在向 UDP 寄存器包括 UDP_TXCV 寄存器的任何读写前使能电源管理控制器中的 UDP 外设时钟。

表 36-4 UDP 存储器分配

偏移量	寄存器	名称	访问类型	复位状态
0x000	帧数寄存器	UDP_ FRM_NUM	读	0x0000_0000
0x004	全局状态寄存器	UDP_ GLB_STAT	读写	0x0000_0000
0x008	功能地址寄存器	UDP_ FADDR	读写	0x0000_0100
0x00c	保留	-	-	
0x010	中断使能寄存器	UDP_ IER	写	
0x014	中断禁用寄存器	UDP_ IDR	写	
0x018	中断屏蔽寄存器	UDP_ IMR	读	0x0000_1200
0x01c	中断状态寄存器	UDP_ ISR	读	0x0000_XX00
0x020	中断清除寄存器	UDP_ ICR	写	
0x024	保留	-	-	
0x028	复位端点寄存器	UDP_ RST_EP	读写	
0x02c	保留	-		
0x030	端点 0 控制和状态寄存器	UDP_CSR0	读写	0x0000_0000
见注 1	端点 5 控制和状态寄存器	UDP_CSR5	读写	0x0000_0000
0x050	端点 0 FIFO 数据寄存器	UDP_ FDR0	读写	0x0000_0000
见注 2	端点 5 FIFO 数据寄存器	UDP_ FDR5	读写	0x0000_0000
0x070	保留	-		
0x074	收发器控制寄存器	UDP_ TXVC (3)	读写	0x0000_0000
0x078-0x FC	保留			

注意：1.UDP_ CSRx 寄存器地址算法： 0x030 + 4(端点数 - 1)

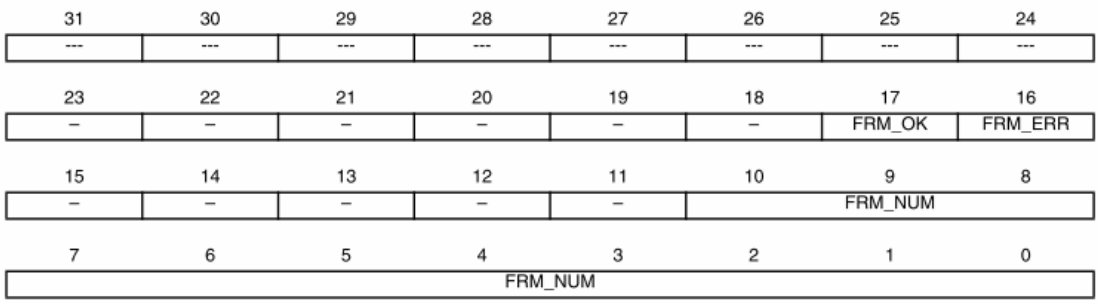
2.UDP_ FDRx 寄存器地主算法： 0x050 + 4(端点数 - 1)

3.见此页“UDP 存储器分配”上面的警告。

36.6.1 UDP 帧数寄存器

寄存器名称：UDP_ FRM_NUM

访问类型：只读



- FRM_NUM[10:0]:在数据包域格式中定义的帧数

每有一帧，主机将这 11 位值加一。每帧开始时更新此值。

在 SOF_EOP（帧开始，数据包结束） 时更新此值。

- FRM_ERR: 帧错误

当接收到的 SOF 包中有错时，在 SOF_EOP 中的该位置位。

收到 SOF_PID 后该位复位。

- FRM_OK: 帧正确

当接收到的 SOF 包中无错时，在 SOF_EOP 中的该位置位。

收到 SOF_PID 后该位复位 (包标识)。

中断状态寄存器中，收到 SOF_PID 时更新 SOF 中断。该位置位不必等待 EOP。

注意:在 8 位寄存器接口中， FRM_OK 为 FRM_NUM_H 的位 4， FRM_ERR 为 FRM_NUM_L 的位 3。

36.6.2 UDP 全局状态寄存器

寄存器名称 :UDP_GLB_STAT

访问类型 :读 / 写

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–		–	CONFG	FADDEN

该寄存器用于按照 USB Serial Bus Specification, Rev.2.0 第 9 章来获得并设置器件状态。

- FADDEN: 功能地址使能

读：

0 = 器件不处于地址状态。

1 = 器件处于地址状态。

写：

0 = 无效，只有复位能将器件带回默认状态。

1 = 将器件设置为地址状态。在成功设置地址请求后出现。在此之前，必须用设置地址参数将 UDP_FADDR 寄存器初始化。在设置 FADDEN 前设置地址必须完成状态阶段，详见 Universal Serial Bus Specification, Rev. 2.0 第 9 章。

- CONFG: 配置

读：

0 = 器件不处于配置状态。

1 = 器件处于配置状态。

写：

0 = 设置器件到非配置状态。

1 = 设置器件到配置状态。

当处于地址状态且接收到成功设置配置请求时将器件置于配置状态，详见 Universal Serial Bus Specification, Rev. 2.0 第 9 章。

36.6.3 UDP 功能地址寄存器

寄存器名称 ： UDP_FADDR

访问类型 :读 / 写

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	FEN
7	6	5	4	3	2	1	0
–	FADD						

- FADD[6:0]: 功能地址值

一旦从主机接收到设置地址请求，并处于无数据控制序列的状态阶段，固件必须对功能地址值编程，参考 Universal Serial Bus Specification，Rev. 2.0 。上电或复位后，功能地址值设置为 0。

- FEN: 功能使能

读：

0 = 功能端点禁用。

1 = 功能端点使能。

写：

0 = 禁用功能端点。

1 = 默认值。

功能使能位 (FEN) 允许微控制器使能或禁用功能端点。从主机接收到复位后微控制器设置该位。一旦该位被设置， USB 器件被主机接受并可与主机传输数据包。

36.6.4 UDP 中断使能寄存器

寄存器名称 :UDP_IER

访问类型 :只写

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	WAKEUP	–	SOFINT	–	RXRSM	RXSUSP
7	6	5	4	3	2	1	0
–	–	EP5INT	EP4INT	EP3INT	EP2INT	EP1INT	EP0INT

- EP0INT: 使能端点 0 中断
- EP1INT: 使能端点 1 中断
- EP2INT: 使能端点 2 中断
- EP3INT: 使能端点 3 中断
- EP4INT: 使能端点 4 中断
- EP5INT: 使能端点 5 中断

0 = 无效。

1 = 使能相应端点中断。

- RXSUSP: 使能 UDP 挂起中断

0 = 无效。

1 = 使能 UDP 挂起中断。

- RXRSM: 使能 UDP 恢复中断

0 = 无效。

1 = 使能 UDP 恢复中断。

- SOFINT: 使能帧起始中断

0 = 无效。

1 = 使能帧起始中断。

- WAKEUP: 使能 UDP 总线唤醒中断

0 = 无效。

1 = 使能 USB 总线中断。

36.6.5 UDP 中断禁用寄存器

寄存器名称 :UDP_IDR

访问类型 :只写

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	WAKEUP	–	SOFINT	–	RXRSM	RXSUSP
7	6	5	4	3	2	1	0
–	–	EP5INT	EP4INT	EP3INT	EP2INT	EP1INT	EP0INT

- EP0INT: 禁用端点 0 中断
- EP1INT: 禁用端点 1 中断
- EP2INT: 禁用端点 2 中断
- EP3INT: 禁用端点 3 中断
- EP4INT: 禁用端点 4 中断
- EP5INT: 禁用端点 5 中断

0 = 无效。

1 = 禁用相应端点中断。

- RXSUSP: 禁用 UDP 挂起中断

0 = 无效。

1 = 禁用 UDP 挂起中断。

- RXRSM: 禁用 UDP 恢复中断

0 = 无效。

1 = 禁用 UDP 恢复中断。

- SOFINT: 禁用帧起始中断

0 = 无效。

1 = 禁用帧起始中断。

- WAKEUP: 禁用 USB 总线唤醒中断

0 = 无效。

1 = 禁用 USB 总线中断。

36.6.6 UDP 中断屏蔽寄存器

寄存器名称 :UDP_IMR

访问类型 :只读

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12[1]	11	10	9	8
–	–	WAKEUP	–	SOFINT	–	RXRSM	RXSUSP
7	6	5	4	3	2	1	0
–	–	EP5INT	EP4INT	EP3INT	EP2INT	EP1INT	EP0INT

注意：1.不允许屏蔽 UDP_IMR 的 Bit 12 并且总是读到 1

- EP0INT: 屏蔽端点 0 中断
- EP1INT: 屏蔽端点 1 中断
- EP2INT: 屏蔽端点 2 中断
- EP3INT: 屏蔽端点 3 中断
- EP4INT: 屏蔽端点 4 中断
- EP5INT: 屏蔽端点 5 中断

0 = 禁用相应端点中断。

1 = 使能相应端点中断。

- RXSUSP: 屏蔽 UDP 挂起中断

0 = 禁用 UDP 挂起中断。

1 = 使能 UDP 挂起中断。

- RXRSM: 屏蔽 UDP 恢复中断

0 = 禁用 UDP 恢复中断。

1 = 使能 UDP 恢复中断。

- SOFINT: 屏蔽帧起始中断

0 = 禁用帧起始中断。

1 = 使能帧起始中断。

- WAKEUP: USB 总线唤醒中断

0 = 禁用 USB 总线唤醒中断。

1 = 使能 USB 总线唤醒中断。

注意: 当 USB 块处于挂起模式时，应用程序可能关闭 USB 逻辑。此时，所有 USB 主机恢复请求必须记录，因此，UDP_IMR 寄存器的 RXRSM 位复位值必须使能。

36.6.7 UDP 中断状态寄存器

寄存器名称 :UDP_ISR

访问类型 :只读

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	WAKEUP	ENDBUSRES	SOFINT	–	RXRSM	RXSUSP
7	6	5	4	3	2	1	0
–	–	EP5INT	EP4INT	EP3INT	EP2INT	EP1INT	EP0INT

- EP0INT: 中断 0 中断状态
- EP1INT: 中断 1 中断状态
- EP2INT: 中断 2 中断状态
- EP3INT: 中断 3 中断状态
- EP4INT: 中断 4 中断状态
- EP5INT: 中断 5 中断状态

0=无端点 0 中断挂起

1 = 端点 0 中断已发生。

几个信号可产生该中断，通过读 UDP_ CSR0:可得：

RXSETUP 置为 1

RX_DATA_BK0 置为 1

RX_DATA_BK1 置为 1

TXCOMP 置为 1

STALLSENT 置为 1

EP0INT 为 粘着位。写相应 UDP_ CSR0 位对 EP0INT 清零前中断有效。

- RXSUSP: UDP 挂起中断状态

0 = 无 UDP 挂起中断等待。

1 = UDP 挂起中断已发生。

当 USB 器件检测到 3ms 无工作时设置该位。 USB 器件进入挂起模式。

- RXRSM: UDP 恢复中断状态

0 = 无 UDP 恢复中断等待。

1 =UDP 恢复中断已发生。

当 USB 器件端口检测到 USB 恢复信号时设置该位。

复位后，该位的状态未定义，应用程序必须清零该位，可通过设置 UDP_ ICR 寄存器中的 RXRSM 标志位。

- SOFINT: 帧开始中断状态

0 = 无帧开始中断等待。

1 = 帧开始中断已发生。

每次检测到 SOF 时发生中断。它可被同步端点作为同步信号使用。

- ENDBUSRES: 总线结束复位中断状态

0 = 无总线结束复位中断等待。

1 = 总线结束复位中断已发生。

UDP 复位序列结束后发生中断。 USB 器件必须在端点 0 准备接收请求。主机开始列举，

然后开始进行配置。

- WAKEUP: UDP 恢复中断状态

0 = 无唤醒中断等待。

1 = 上次清零后出现唤醒中断 (USB 主机发送 RESUME 或 RESET)。

复位后，该位的状态未定义，应用程序必须清零该位，可通过设置 UDP_ ICR 寄存器中的 WAKEUP 标志位。

36.6.8 UDP 中断清除寄存器

寄存器名称 :UDP_ICR

访问类型 :只写

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	WAKEUP	ENDBUSRES	SOFINT	–	RXRSM	RXSUSP
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- RXSUSP: 清除 UDP 挂起中断

0 = 无效。

1 = 清除 UDP 挂起中断。

- RXRSM: 清除 UDP 恢复中断

0 = 无效。

1 = 清除 UDP 恢复中断。

- SOFINT: 清除帧开始中断

0 = 无效。

1 = 清除帧开始中断。

- ENDBURST: 清除总线复位结束中断

0 = 无效。

1 = 清除总线复位结束中断。

- WAKEUP: 清除唤醒中断

0 = 无效。

1 = 清除唤醒中断。

36.6.9 UDP 复位端点寄存器

寄存器名称 :UDP_RST_EP

访问类型 :读 / 写

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	EP5	EP4	EP3	EP2	EP1	EP0

- EP0: 复位端点 0
- EP1: 复位端点 1
- EP2: 复位端点 2
- EP3: 复位端点 3
- EP4: 复位端点 4
- EP5: 复位端点 5

该标志用于复位与端点相关的 FIFO 及 UDP_CSRx 寄存器中的 RXBYTECOUNT 位。它还复位数据切换到 DATA0。它在删除 BULK 端点 HALT 状态后有用。参见 USB Serial Bus Specification, Rev.2.0. 的 5.8.5 节。

警告：复位结束时该标志必须清除。不清除 UDP_CSRx 标志。

0 = 无复位。

1 = 强制将相应端点 FIF0 指向 0，因此 UDP_CSRx 寄存器 RXBYTECNT 域为 0。

36.6.10 UDP 端点控制与状态寄存器

寄存器名称 :UDP_CSRx [x = 0. 5]

访问类型 :读 / 写

31	30	29	28	27	26	25	24
–	–	–	–	–	RXBYTECNT		
23	22	21	20	19	18	17	16
RXBYTECNT							
15	14	13	12	11	10	9	8
EPEDS	–	–	–	DTGLE	EPTYPE		
7	6	5	4	3	2	1	0
DIR	RX_DATA_ BK1	FORCE STALL	TXPKTRDY	STALLSENT ISOERROR	RXSETUP	RX_DATA_ BK0	TXCOMP

警告：由于 MCK 和 UDPCK 之间的同步，通过轮询必须被置位/清零的位，应用软件必须在执行另一次写之前等待写操作结束。

```
//! Clear flags of UDP UDP_CSR register and waits for synchronization
#define Udp_ep_clr_flag(pInterface, endpoint, flags) { \
while (pInterface->UDP_CSR[endpoint] & (flags)) \
pInterface->UDP_CSR[endpoint] &= ~(flags); \
}
//! Set flags of UDP UDP_CSR register and waits for synchronization
#define Udp_ep_set_flag(pInterface, endpoint, flags) { \
while ( (pInterface->UDP_CSR[endpoint] & (flags)) != (flags) ) \
pInterface->UDP_CSR[endpoint] |= (flags); \
}
```

- TXCOMP: 产生一个有前数据写入 DPR 的入包

该标志设置为 1 时产生中断。

写 (由固件清零) ；

0 = 清除标志，清除中断。

1 = 无效。

读 (由 USB 外设设置):

0 = 主机未应答数据入处理。

1 = 数据入处理完成，主机应答。

数据入处理设置 TXPKTRDY 后，器件固件等待 TXCOMP 以确保主机已应答处理。

- RX_DATA_BK0: 接收数据段 0

该标志设置为 1 时产生中断。

写 (由固件清零) ；

0 = 通知 USB 外设器件， FIFO 段 0 数据已被读取。

1 = 无效。

读 (由 USB 外设设置):

0 = FIFO 段 0 未收到数据包。

1 = 已收到数据包，并已存入 FIFO 段 0。

当器件固件已轮询该位或由该信号中断，必须将数据由 FIFO 送到微控制器存储器。RXBYTCENT 域中收到的字节数有效。通过 UDP_FDRx 寄存器读 FIFO 段 0 值。一旦

发送完成，器件固件必须通过清除 RX_DATA_BK0 将 USB 外设器件段 0 释放。

- RXSETUP: 接收设置

该标志设置为 1 时产生中断。

读：

0 = 无有效设置包。

1 = 主机已发送设置包并在 FIFO 中有效。

写：

0 = 器件固件通知 USB 外设器件，它已读取 FIFO 中设置数据。

1 = 无效。

该标志用来通知 USB 器件固件，主机已收到有效设置数据包并由 USB 器件成功接收。USB 器件固件可通过读 UDP_FDRx 寄存器将 FIFO 中设置数据发送到微控制器处理器。一旦发送完成， RXSETUP 必须由器件固件清除。

当 RXSETUP 置位时，不接收随后的数据出数据。

- STALLSENT: 发送停止 (控制、批中断端点) / ISOERROR (同步端点)

该标志设置为 1 时产生中断。

STALLSENT：结束停止握手。

读：

0 = 主机未应答 STALL。

1 = 主机应答 STALL。

写：

0 = 复位 STALLSENT 标志，清除中断。

1 = 无效。

强制器件固件清除该标志，否则中断保持。

关于 STALL 握手，参见通用串行总线规范，Rev. 2.0 8.4.5 及 9.4.5 节。

ISOERROR：同步发送中检测到 CRC 错误。

读：

0 = 之前的同步传输中无错误。

1 = 检测到 CRC 错误， FIFO 中可用数据被破坏。

写：

0 = ISOERROR 标志复位，清除中断。

1 = 无效。

- TXPKTRDY: 发送包就绪

该标志由 USB 器件清除。

该标志由 USB 器件固件置位。

读：

0 = 数据值可写入 FIFO。

1 = 数据值不能写入 FIFO。

写：

0 = 无效。

1 = 新数据有效负载通过固件写入 FIFO 并即将发送。

该标志用来产生数据入处理 (器件到主机)。器件固件检查它能否在 FIFO 中写入数据有效负载，检查 TXPKTRDY 是否清零。通过写 UDP_FDRx 寄存器实现数据到 FIFO 的发送。一旦数据有效否则发送到 FIFO，固件通知 USB 器件将 TXPKTRDY 设置为 1 。USB 总线处理开始。一旦主机收到数据有效负载， TXCOMP 置位。

- FORCESTALL: 强制停止 (用于控制、批及同步端点)

读：
0=普通状态
1=停止状态
写：
0 = 返回到普通状态。
1 = 向主机发送 STALL。
关于 STALL 握手，参见通用串行总线规范，Rev. 2.0 8.4.5 及 9.4.5 节。
控制端点：数据筹备与状态筹备过程中，表明微控制器不能完成请求。
批与中断端点：通知主机端点已停止。
主机应答 STALL， STALLSENT 标志通知器件固件。

- RX_DATA_BK1: 接收数据段 1 (只用于有 ping-pong 特性的端点)

该标志设置为 1 时产生中断。
写 (由固件清零) ；
0 = 通知 USB 器件， FIFO 段 1 数据已被读取。
1 = 无效。
读 (由 USB 外设设置):
0 = FIFO 段 1 未收到数据包。
1 = 已收到数据包，并已存入 FIFO 段 1。
当器件固件已轮询该位或由该信号中断，必须将数据由 FIFO 送到微控制器存储器。RXBYTCENT 域中收到的字节数有效。通过 UDP_FDRx 寄存器读 FIFO 段 1 值。一旦发送完成，器件固件必须通过清除 RX_DATA_BK1 将 USB 外设器件段 1 释放。

- DIR: 发送方向 (仅对控制端点有效)

读 / 写
0 = 在控制数据筹备时允许数据出处理。
1 = 控制数据筹备时使能数据入处理。
关于控制数据筹备，参见通用串行总线规范，Rev. 2.0 8.5.3 节。
在设置筹备结束清除 UDP_CSRx/RXSETUP 前该位必须置位。根据设置数据包发送请求，数据筹备为器件到主机 (DIR = 1)或主机到器件 (DIR = 0) 数据发送。状态筹备时不需检验该位反向。

- EPTYPE[2:0]: 端点类型

读写

000	控制
001	同步 OUT
101	同步 IN
010	批 OUT
110	批 IN
011	中断 OUT
111	中断 IN

- DTGLE: 数据切换

只读
0 = 识别 DATA0 包。
1 = 识别 DATA1 包。

更多 DATA0、 DATA1 包定义见通用串行总线规范，Rev. 2.0 第八章。

- EPEDS: 端点使能禁用

读：

0 = 端点禁用。

1 = 端点使能。

写：

0 = 禁用端点。

1 = 使能端点。

- RXBYTECNT[10:0]: FIFO 中可用字节数

只读

当主机向器件发送数据包时， USB 器件将数据存入 FIFO 并通知微控制器。微控制器可通过读取 UDP_FDRx 寄存器的 RXBYTECENT 字节载入 FIFO 中数据。

36.6.11 UDP FIFO 数据寄存器

寄存器名称：UDP_ FDRx [x = 0..5]

访问类型：读写

- FIFO_DATA[7:0]: FIFO 数据值

微控制器可通过该寄存器将数据推入或弹出 FIFO。

相应 UDP_CSRx 寄存器的 RXBYTECNT 中是由 FIFO 中读取的字节数 (主机发送)。能写入的最大字节数由标准端点描述符的最大包尺寸确定。它不能大于相关端点物理存储器大小，详见通用串行总线规范，Rev. 2.0 。

36.6.12 UDP 收发器控制寄存器

寄存器名称： UDP_ TXVC

访问类型：读写

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	TXVDIS
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

警告：必须在向 UDP 寄存器包括 UDP_TXCV 寄存器的任何读写操作前使能电源管理控制器（PMC）中的 UDP 外设时钟。

- TXVDIS: 禁用收发器

当禁用 UDP，可通过禁用内置的收发器来有效的降低功耗。可通过设置 TXVDIS 域实现。

注意：如果 USB 上拉未连接到 DP，用户不能向任何 UDP 寄存器写入，除了 UDP_ TXVC 寄存器。

- PUON：上拉

0：断开集成在 DP 上的 1.5 千欧上拉电阻

1：连接集成在 DP 上的 1.5 千欧上拉电阻

Powered by Team Mcuzone

QQ:8204136

Website: www.mcuzone.com

2009

MCUZone 微控电子

www.mcuzone.com