

硬核处理器系统 (HPS) 提供 USB On-The-Go (OTG) 控制器的两个实例，可以支持器件和主机功能。控制器支持器件和主机模式中的所有高速、全速和低速传输。控制器完全符合 *On-The-Go and Embedded Host Supplement to the USB Revision 2.0* 规范。控制器可以被编程，以便器件和主机功能通过 USB 协议支持数据转移。


控制器彼此单独地进行操作。**每个 USB OTG 控制器都支持一个通过 USB 2.0 收发器宏单元接口加上 (UTMI+) 低管脚接口 (ULPI) 兼容的 PHY 连接的单 USB 端口。**USB OTG 控制器是 Synopsys® DesignWare® Cores USB 2.0 Hi-Speed On-The-Go (DWC_otg) 控制器的实例。

USB OTG 控制器对于以下的应用和系统而被优化：

- 便携电子器件
- 点到点应用（没有调试集线器，直接连接 HS、FS 或 LS 器件）
- 器件（集线器和分隔支持）的多点应用（作为一个嵌入的 USB 主机）

两个 USB OTG 端口中的每个端口都支持主机和器件模式，如 *On-The-Go and Embedded Host Supplement to the USB Revision 2.0* 规范中所介绍。USB OTG 端口支持所有类型 USB 外设的连接，包括以下外设：

- 鼠标
- 键盘
- 数字摄像机
- 网络适配器
- 硬盘驱动器
- 通用集线器

 额外的信息在 *On-The-Go and Embedded Host Supplement to the USB Revision 2.0* 规范中，可以从 USB Implementers Forum 网站 (www.usb.org) 下载。

USB OTG 控制器的功能

USB OTG 控制器具有以下 USB 指定的功能：

- 符合版本 1.3 和版本 2.0 的 *On-The-Go and Embedded Host Supplement to the USB Revision 2.0* 规范
- 支持 OTG 1.3 和 OTG 2.0 之间的软件可配置模式的操作

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

Portions © 2011 Synopsys, Inc. Used with permission. All rights reserved. Synopsys & DesignWare are registered trademarks of Synopsys, Inc. All documentation is provided "as is" and without any warranty. Synopsys expressly disclaims any and all warranties, express, implied, or otherwise, including the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, and any warranties arising out of a course of dealing or usage of trade.

†Paragraphs marked with the dagger (†) symbol are Synopsys Proprietary. Used with permission.



- 支持所有 USB 2.0 速度：

- 高速 (HS、480-Mbps)
- 全速 (FS、12-Mbps)
- 低速 (LS、1.5-Mbps)



在主机模式中，所有速度都被支持。然而，在器件模式中，只有高速和全速被支持。

- 支持所有 USB 传输类型：

- 控制传输
- Bulk 传输
- 等时传输
- 中断

- 支持自动的 ping 性能

- 支持会话请求协议 (SRP) 和主机流通协议 (HNP)

- 支持挂起、恢复和远程唤醒

- 支持高达 16 个主机通道



在主机模式中，当器件端点数大于主机通道数时，软件会重新编程通道以支持高达 127 个器件，每个器件有 32 个端点 (IN + OUT)，所以总共高达 4,064 个端点。

- 支持高达 16 个双向端点，包括控制端点 0



只有 7 个 **定期器件** IN 端点被支持。

- 支持一个通用根集线器

- 执行硬件中的传输规划

在 USB PHY 层，USB OTG 控制器支持以下功能：

- 一个单 USB 端口连接到每个 OTG 实例
- 一个 ULPI 连接到片外 USB 收发器


- **软件控制的访问，支持供应商指定的或可选的 PHY 寄存器访问以简化调试。**

- OTG 2.0 仅通过一个外部 (片外) ADP 控制器支持 Attach Detection Protocol (附加检测协议，ADP)

在集成方面，USB OTG 控制器支持以下功能：

- 系统和 PHY 接口的不同的时钟
- 直接存储器访问 (DMA) 模式中每个器件 IN 端点的专用 TX FIFO 缓冲器
- **对小 FIFO 缓冲器端点的基于包的，动态 FIFO 存储器分配以及对软件动态地设置容量的 RAM 的灵活、有效地使用**
- 能够在传输期间更改一个端点的 FIFO 存储器容量

- USB 挂起和会话结束模式期间的时钟选通支持
 - PHY 时钟选通支持
 - 系统时钟选通支持
- 数据 FIFO RAM 时钟选通支持

 USB OTG 控制器不支持以下协议：

- 增强主机控制器接口 (Enhanced Host Controller Interface, EHCI)
- 开放式主机控制器接口 (Open Host Controller Interface, OHCI)
- 通用主机控制器接口 (Universal Host Controller Interface, UHCI)

所支持的 PHY

表 18 - 1 列出了一些与 USB OTG 兼容的 PHY。

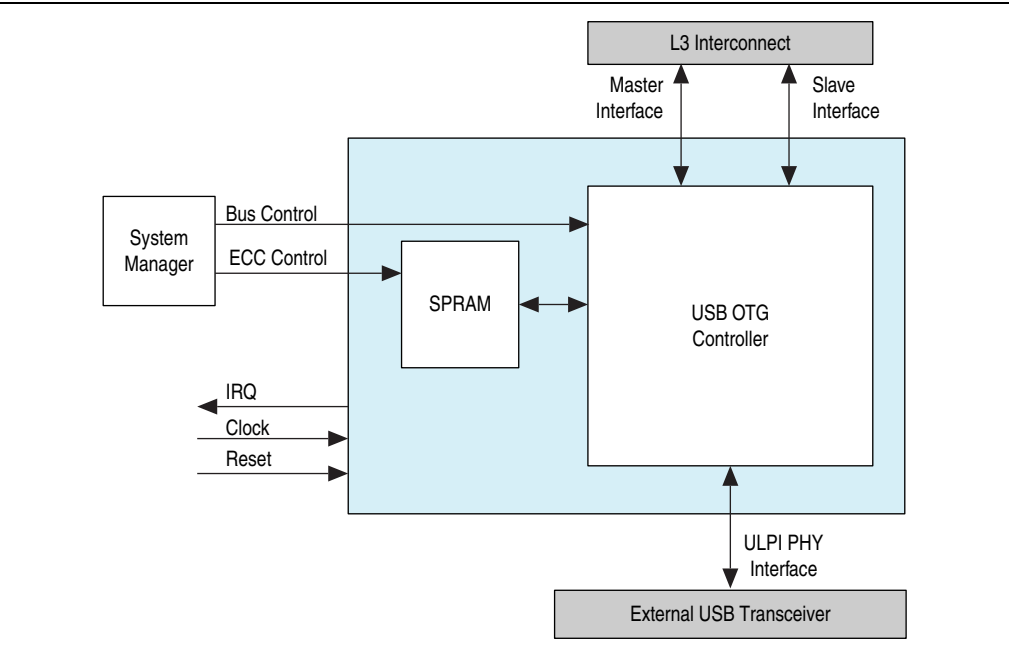
表 18 - 1. 所支持的 PHY

制造商	部件编号
TI	TUSB1210
NXP	ISP1504
Cypress	CY7C68003
SMSC	USB3300

USB OTG 控制器结构图和系统集成

图 18 - 1 显示了 HPS 中的一个 USB OTG 控制器子系统的结构图。HPS 中包含两个子系统。

图 18 - 1. USB OTG 控制器系统集成



USB OTG 控制器通过从接口连接到 level 3 (L3) 互联，从而支持其它的主器件访问控制器中的控制和状态寄存器 (CSRs)。控制器也通过主接口连接到 L3 互联，从而支持控制器中的 DMA 引擎移动外部存储器和控制器之间的数据。

连接到 USB OTG 控制器的单端口 RAM (SPRAM) 用于在主机和器件模式中存储 USB 数据包。它被配置为 FIFO 缓冲器以便接收和发送 USB 路的数据包。

通过系统管理器，USB OTG 控制器可以对使用 and 测试 SPRAM 中的错误纠正代码 (ECC) 进行控制。通过系统管理器，USB OTG 控制器也可以控制 L3 互联的主接口的行为。



要了解更多信息，*Cyclone V 器件手册*第 3 卷的 *System Manager* 章节。

USB OTG 控制器通过 ULPI PHY 接口连接到外部 USB 收发器。该接口也通过管脚多路复用器连接到 HPS 中。管脚多路复用器也通过系统管理器控制。

USB OTG 控制器的额外的连接包括：

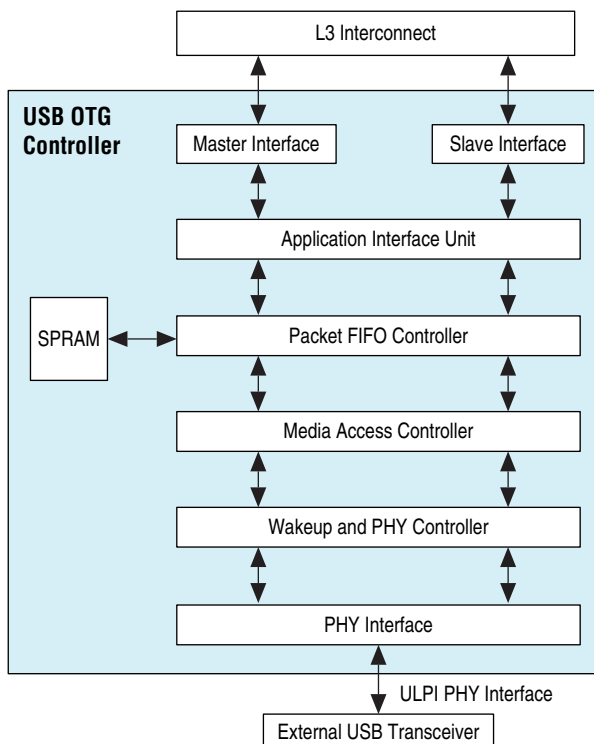
- 从时钟管理器到 USB OTG 控制器的时钟输入
- 从复位管理器到 USB OTG 控制器的复位输入
- 从 USB OTG 控制器到微处理器单元 (MPU) 全局中断控制器 (GIC) 的中断线。

USB OTG 控制器的功能说明

USB OTG 控制器模块说明

图 18-2 是 USB OTG 控制器的结构图。以下部分提供包括 USB OTG 控制器的每个单元的详细信息。

图 18-2. USB OTG 控制器结构图



主接口

主接口包含一个内置 DMA 控制器。DMA 控制器在外部存储器和介质访问控制器 (MAC) 之间移动数据。

主接口的属性通过系统管理器中的 USB L3 主 HPROT 寄存器 (l3master) 控制。这些位提供对 L3 互联的访问信息，包括传输是否可缓存、缓冲或特许的。

 只有当主接口被保证处于非激活状态时，l3master 寄存器中的位才可以被更新。

从接口

从接口支持系统中其它主器件访问 USB OTG 控制器的 CSR。为了测试目的，其它主器件也可访问 SPRAM。

从接口 CSR 单元

从接口可以读取并且写入 USB OTG 控制器中的所有 CSR。所有寄存器访问都是 32 位。

CSR 被划分为以下的寄存器组：

■ 全局

- 主机
- 器件
- 功耗和时钟选通

一些寄存器在主机和器件模式之间共享，因为控制器只可以每次为一种模式。如果一个主器件试图在控制器处于主机模式时访问器件寄存器，或试图在控制器处于器件模式时访问主机寄存器，那么控制器生成一个模式失配中断。未实现的寄存器的写入被忽略。未实现的寄存器的读取返回不确定的值。

应用接口单元

应用接口单元 (AIU) 生成基于可编程的 FIFO 缓冲器阈值的 DMA 请求。AIU 在主机和器件模式中生成 GIC 的中断。DMA 调速器包含在 AIU 中，用于仲裁和控制系统存储器中数据包和它们各自的 USB 端点之间的数据传输。

Packet FIFO 控制器

Packet FIFO 控制器 (PFC) 通过位于 SPRAM 中的数据 FIFO 缓冲器将 AIU 与 MAC 连接。在器件模式中，**一个 FIFO 缓冲器对每个 IN 端点而执行操作**。在主机模式中，一个单 FIFO 缓冲器对所有定期（等时和中断）OUT 端点存储数据，并且一个单 FIFO 缓冲器用于非定期（控制器和 bulk）OUT 端点。主机和器件模式共享一个单接收数据 FIFO 缓冲器。

SPRAM

SPRAM 在主机和器件模式中实现数据 FIFO 缓冲。FIFO 缓冲的容量可以被动态地编程。

SPRAM 支持 ECC。ECC 可以通过在系统管理器中设置 USB0 中的 RAM ECC 使能 (en) 位或 ECC 管理寄存器组 (eccgrp) 中的 USB1 RAM ECC 使能寄存器 (usb0 或 usb1) 而被使能。每个 USB 实例中的单位和双位错误可以使用该寄存器而被注入。

当检测到（和纠正了）单位可纠正的错误，并且检测到双位（不可纠正的）错误时，SPRAM 提供输出以通知系统管理器。当检测到 ECC 错误时，系统管理器生成一个 GIC 的中断。

MAC

MAC 模块实现以下功能性：

- USB 传输支持
- 主机协议支持
- 器件协议支持
- OTG 协议支持
- 联合电源管理 (LPM) 功能

USB 传输

在器件模式中，MAC 解码和检查所有信令包的完整性。对于有效 OUT 或 SETUP 令牌，以下 DATA 数据包也被检查。如果数据包有效，那么 MAC 执行以下步骤：

1. 将数据写入接收 FIFO 缓冲器
2. 当 USB 主机需要时发送相应的握手。

如果一个接收 FIFO 缓冲器不可用，那么 MAC 对主机发送一个 NAK 响应。MAC 也支持 ping 协议。

对于 IN 令牌，如果数据在发送 FIFO 缓冲器中可用，那么 MAC 执行以下步骤：

1. 从 FIFO 缓冲器读取数据
2. 形成数据包
3. 将数据包发送到主机
4. 从主机接收响应
5. 发送更新的状态到 PFC

在主机模式中，MAC 从 AIU 接收一个令牌请求。MAC 执行以下步骤：

1. 建立信令包
2. 将数据包发送到器件

对于 OUT 或 SETUP 传输，MAC 也执行以下步骤：

1. 从发送 FIFO 缓冲器读取数据
2. 组装数据包
3. 将数据包发送到器件
4. 等待一个响应

器件的响应导致 MAC 对 AIU 发送一个状态更新。

对于 IN 或 PING 传输，MAC 等待器件的数据或握手响应。对于数据响应，MAC 执行以下步骤：

1. 验证数据
2. 将数据写入接收 FIFO 缓冲器
3. 将状态更新发送到 AIU
4. 如果适当，将握手发送到器件。

主机协议

在主机模式中，MAC 执行以下功能：

- 检测 USB 链路的连接、中断、远程唤醒事件
- 启动复位
- 启动速度枚举程序
- 生成帧开始 (SOF) 数据包

器件协议

在器件模式中，MAC 执行以下功能：

- 处理 USB 复位序列
- 处理速度枚举
- 检测 USB 链路的 USB 挂起和恢复活动
- 启动远程唤醒

- 解码 SOF 数据包

OTG 协议

MAC 对于 OTG 操作处理 HNP 和 SRP。HNP 对交换主机和器件角色提供一种机制。SRP 提供机制以便主机关闭 V_{BUS} 来节省功耗，并且使器件请求一个新 USB 会话。

LPM 功能

USB OTG 控制器支持主机和器件模式中的 LPM。通过该功能，当成功的 LPM 传输在 USB 链路发生时，USB OTG 控制器会进入睡眠状态。

唤醒和功耗控制

为了减少功耗，USB OTG 控制器支持一种断电模式。在断电模式中，控制器和 PHY 会关闭它们的时钟。控制器在检测以下事件时支持唤醒：

- 恢复
- 远程唤醒
- 会话请求协议
- 新会话开始

PHY 接口单元

USB OTG 控制器支持到 ULPI PHY 的同步 SDR 数据传输。SDR 模式实现一个 8 位数据总线。

ULPI PHY 接口

ULPI PHY 接口同步于来自 PHY 的 ulpi_clk 信号。表 18-2 列出了 ULPI PHY 接口名称和相关信息。

表 18-2. ULPI PHY 接口

端口名称	位宽度	方向	说明
ulpi_clk	1	输入	ULPI 时钟 接收高速 ULPI PHY 提供的 60-MHz 时钟。所有信号都同步于时钟的正边沿。
ulpi_dir	1	输入	ULPI 数据总线控制 1—PHY 具有传输到 USB OTG 控制器的数据。 0—PHY 没有传输的数据。
ulpi_nxt	1	输入	ULPI 下一个数据控制 表明 PHY 已经接受了 USB OTG 控制器的当前字节。当 PHY 发送时，该信号表明一个新的字节可用于控制器。
ulpi_stp	1	输出	ULPI 停止数据控制 控制器驱动该信号为高电平以表明它的数据流的结束。控制器也可以驱动该信号为高电平以请求 PHY 的数据。
ulpi_data[7:0]	8	双向	双向数据总线。闲置期间由控制器驱动为低电平。

时钟

当复位被释放时，所有的时钟都必须是可操作的。在时钟上不需要特殊的处理。

表 18-3 列出了 USB OTG 控制器时钟输入。

表 18-3. USB OTG 控制器时钟输入

时钟信号	频率	功能使用
usb_mp_clk	60 - 200 MHz	驱动主和从接口、DMA 控制器和内部 FIFO 缓冲器
usb0_ulpi_clk	60 MHz	用于外部 ULPI PHY I/O 管脚的 usb0 的 ULPI 参考时钟
usb1_ulpi_clk	60 MHz	用于外部 ULPI PHY I/O 管脚的 usb1 的 ULPI 参考时钟

复位

USB OTG 控制器可以通过硬件复位输入或软件而复位。

复位要求

控制器退出复位之前，ulpi_clk 时钟上必须有最低 12 个周期。复位期间，USB OTG 控制器置位 ulpi_stp 信号。PHY 看到 ulpi_stp 信号被置位时输出一个时钟。然而，如果管脚多路复用器不被编程，那么 PHY 不会看到 ulpi_stp 信号。结果，ulpi_clk 时钟信号不会到达 USB OTG 控制器。

软件必须确保复位保持最低两个 usb_mp_clk 周期。没有最高置位时间。

硬件复位

每个 USB OTG 控制器具有来自复位管理器的一个复位输入。复位信号在冷或暖复位事件期间被置位。复位管理器保持控制器处于复位直到软件释放复位。通过将 HPS 复位管理器中外设模块复位寄存器 (permodrst) 中的相应 USB 位清零，软件释放复位。

复位输入对以下模块复位：

- 主和从接口逻辑
- 集成的 DMA 控制器
- 内部 FIFO 缓冲器
- CSR

复位输入同步于 usb_mp_clk 域。复位输入也同步于 USB OTG 控制器内的 ULPI 时钟并且用于复位 ULPI PHY 域逻辑。

软件复位

通过设置 USB OTG 控制器的全局寄存器 (globgrp) 组中的复位寄存器 (grstctl) 的内核软复位 (csftrst) 位，软件可以复位控制器。

软件复位在以下情况中是有用的：

- 一个 PHY 选择位由软件更改。复位 USB OTG 控制器是清除的一部分以确保 PHY 可以使用新的配置或时钟进行操作。
- 在软件开发和调试期间。

中断

每个 USB OTG 控制器都有一个单中断输出。中断在表 18 - 4 中所示的条件下置位。

表 18 - 4. USB OTG 中断条件

条件	模式
检测到器件启动的远程唤醒。	主机模式
从器件检测到会话请求。	主机模式
检测到器件中断。	主机模式
主机 LPM 入口重试 (entry retry) 已经过期或 LPM 传输完成。	主机模式
主机定期 TX FIFO 缓冲器为空（可以被进一步编程以表示一半为空）。	主机模式
接收到主机通道中断。	主机模式
未完成的定期传输在微帧结束时待定。	主机模式
接收到主机端口状态中断。	主机模式
检测到外部主机启动的恢复。	器件模式
LPM 握手被发送。	器件模式
当在挂起或正常模式时检测到复位。	器件模式
USB 挂起模式被检测到。	器件模式
TX FIFO 缓冲器满或请求队列满导致数据获取被挂起。	器件模式
至少一个等时 OUT 端点在微帧结束时待定。	器件模式
至少一个等时 IN 端点在微帧结束时待定。	器件模式
至少一个 IN 或 OUT 端点中断在微帧结束时待定。	器件模式
到达周期帧结束。	器件模式
没有成功将等时 OUT 数据包写入 RX FIFO 缓冲器。RX FIFO 缓冲器没有足够的空间来适应等时 OUT 端点的最高数据包容量。	器件模式
枚举已经完成。	器件模式
连接器 ID 更改。	通用模式
模式失配。软件访问属于错误模式的寄存器。	通用模式
非定期的 TX FIFO 缓冲器为空。	通用模式
RX FIFO 缓冲器不为空。	通用模式
微帧开始。	通用模式
器件连接去抖动在主机模式中完成。	OTG 中断
等待 B- 器件连接时，A- 器件超时。	OTG 中断
主机协商完成。	OTG 中断
会话请求完成。	OTG 中断
在器件模式中检测到会话结束。	OTG 中断

USB OTG 控制器编程模型

要了解关于使用 USB OTG 控制器的详细信息，请参考操作系统 (OS) 驱动器文档。OS 供应商提供应用编程接口 (APIs) 以控制 USB 主机、器件和 OTG 操作。该部分对以下软件操作提供简单概述：

- 使能 SPRAM ECC

- 主机操作
- 器件操作

使能 SPRAM ECC

要避免不真实的 ECC 错误，必须在使用 ECC 之前初始化 SPRAM 中的 ECC 位。要初始化 ECC 位，软件将数据写入 SPRAM 中的所有位置。

L3 互联通过 USB OTG L3 从接口访问 SPRAM。软件通过 directfifo 存储器空间在 USB OTG 控制器地址空间访问 SPRAM。

SPRAM 包含 8192 (32 KB) 的空间。L3 从接口对 SPRAM 提供 32-bit 访问。物理上，SPRAM 被实现为一个 35-bit 存储器，最高的 3 个位已保留，以便实现 USB OTG 控制器的内部使用。通过 L3 从接口对 SPRAM 执行写操作时，内部数据总线的 32 到 34 位被连接到 1，以便使能 ECC 位使其初始化。

directfifo 存储器空间在控制器地址映射中有所介绍。请参考第 18 - 14 页的“USB OTG 控制器地址映射和寄存器定义”。



软件不可以访问超出 32-KB 范围的 SPRAM。范围外的读传输返回不确定的数据。范围外的写传输被忽略。

主机操作

主机初始化

上电后，USB 端口处于默认模式。没有 V_{BUS} 被应用于 USB 电缆。以下过程将 USB OTG 控制器设置为 USB 主机。

1. 要使能 USB 端口的电源，软件驱动器将主机模式寄存器 (hostgrp) 组的主机端口控制和状态寄存器 (hprt) 中的端口电源 (prtpwr) 位设置为 1。该操作驱动 USB 链路的 V_{BUS} 信号。

控制器等待在 USB 链路检测到一个连接。

2. 当一个 USB 器件连接时，一个中断被生成。hprt 中的端口连接检测 (PrtConnDet) 位被设置为 1。
3. 当检测端口连接时，通过将 hprt 中的端口复位 (prtrst) 位设置为 1，软件驱动器启动端口复位。
4. 软件驱动器必须等待最低 10 ms，以便速度枚举可以在 USB 链路完成。
5. 10 ms 之后，软件驱动器将 prtrst 再设置到 0，以便释放端口复位。
6. USB OTG 控制器生成一个中断。hprt 中的端口使能禁用更改 (prtENCHNG) 和端口速度 (prtSPD) 位被设置，以反映相连的器件的枚举速度。

这时，端口被使能以进行通信。保持进行或 SOF 数据包在端口上发送。如果 USB 2.0- 功能器件没有成功地正确初始化，那么它被报告为 USB 1.1 器件。

主机帧间隔寄存器 (hfir) 使用相应的 PHY 时钟设置进行更新。用于发送 SOF 数据包的 hfir 在主机模式寄存器 (hostgrp) 组中。

7. 软件驱动器必须按以下的顺序编程全局寄存器 (globgrp) 组中的寄存器：
 - a. 接收 FIFO 容量寄存器 (grxfsize) — 选择接收 FIFO 缓冲器的容量。
 - b. 非定期发送 FIFO 容量寄存器 (gnptxfsize) — 对非定期传输选择非定期发送 FIFO 缓冲器的容量和起始地址。
 - c. 主机定期发送 FIFO 容量寄存器 (hptxfsize) — 对定期传输选择定期发送 FIFO 缓冲器的容量和起始地址。
8. 系统软件至少初始化和使能一个通道以便和 USB 器件通信。

主机传输

当配置为主机时，USB OTG 控制器通过两个请求队列的其中之一（一个用于定期传输，一个用于非定期传输）进行 USB 传输。请求队列中的每个入口保持所需 SETUP、IN 或 OUT 通道数以及其它信息，以便执行 USB 链路的传输。请求被写入队列的序列决定 USB 链路的传输序列。

在每个帧或微帧开始时，主机以下列顺序处理请求：

1. 定期请求队列，包括等时和中断传输
2. 非定期请求队列 (bulk 或控制传输)

主机以轮流的方式对每个使能通道规划传输。当主机控制器对通道完成传输时，控制器更新系统存储器中的 DMA 描述符状态。

对于 OUT 传输，主机控制器使用两个发送 FIFO 缓冲器以便保持数据包有效负载被发送。一个发送 FIFO 缓冲器用于所有非定期 OUT 传输，另一个用于所有定期 OUT 传输。

对于 IN 传输，USB 主机控制器将一个接收 FIFO 缓冲器用于所有定期和非定期传输。控制器保持接收 FIFO 缓冲器 USB 器件的数据包有效负载直到数据包被传输到系统存储器。接收 FIFO 缓冲器也保持每个接收到的数据包的状态。状态入口保持 IN 通道数以及其它信息，包括接收到的字节数和有效性状态。

对于通用集线器操作，USB OTG 控制器使用 SPLIT 传输来与集线器的较慢速度器件下游通信。对于这些传输，传输累加或缓冲在通用集线器中执行，并且被相应地规划。USB OTG 控制器确保下游传输完成或累加的数据开始发送到上游时，足够的发送和接收缓冲器被分配。

器件操作

器件初始化

以下过程将 USB OTG 控制器设置为 USB 器件：

1. 上电后，通过写入器件模式寄存器 (devgrp) 组的器件配置寄存器 (dcfg) 中的器件速度 (devspd) 位，USB OTG 控制器必须被设置为所需器件速度。器件速度被设置后，控制器等待 USB 主机检测作为器件端口的 USB 端口。
2. 当外部主机检测 USB 端口时，主机执行一个端口复位，从而生成 USB 器件软件的中断。全局寄存器 (globgrp) 组中的中断 (port reset) 寄存器的 USB 复位 (usbrst) 位被设置。然后器件软件设置数据 FIFO 缓冲器，以接收外部主机的 SETUP 数据包。端点 0 还未被使能。
3. 端口复位完成后，外部主机要求的操作速度为已知。软件读取器件速度状态并且设置所有剩余所需传输域以便使能控制端点 0。

该过程完成后，器件接收 SOF 数据包，并且对 USB 主机设置器件的控制端点准备就绪。

器件传输

当配置为器件时，对于所有 OUT 端点，USB OTG 控制器使用单 FIFO 缓冲器接收数据。接收 FIFO 缓冲器保持接收到的数据包的状态，包括字节数、数据包 ID (PID) 以及接收到数据的有效性。数据被接收到时，DMA 控制器读取 FIFO 缓冲器的数据。如果 FIFO 缓冲器上溢情况发生，那么控制器使用一个 NAK 响应 OUT 数据包，并且从内部倒转指针。

对于 IN 端点，控制器将专用发送缓冲器用于每个端点。应用不需要预测 USB 主机访问非定期端点的顺序。如果 FIFO 缓冲器下溢情况在发送期间出现，那么控制器反转循环冗余码 (CRC) 以在 USB 链路上标记数据包为损坏。

在传输级操作中，应用对于每端点每次处理一个数据包。软件在每个数据包完成时接收到一个中断。根据 USB 链路接收的握手响应，应用决定是否重试传输或继续下一个传输，直到传输中的所有数据包完成。

IN 传输

对于一个 IN 传输，应用执行以下步骤：

1. 使能端点
2. 触发 DMA 引擎，以便将相关的数据包写入相应的发送 FIFO 缓冲器。
3. 等待控制器的数据包完成中断

当端点接收到一个 IN 令牌而相关的发送 FIFO 缓冲器不包含足够的数据时，控制器执行以下步骤：

1. 生成一个中断
2. 将 NAK 握手返回 USB 主机

如果足够的数据可用，那么控制器将数据发送到 USB 主机。

OUT 传输

对于一个 OUT 传输，应用执行以下步骤：

1. 使能端点
2. 等待从 USB OTG 控制器接收到的数据包中断
3. 检索接收 FIFO 缓冲器的数据包

当端点接收到一个 OUT 令牌或 PING 令牌而接收 FIFO 缓冲器没有足够的空间时，控制器执行以下步骤：

1. 生成一个中断
2. 将 NAK 握手返回 USB 主机

如果足够的空间可用，那么控制器将数据存储在接收 FIFO 缓冲器并且将 ACK 握手返回 USB 链路。


控制传输

对于控制传输，应用执行以下步骤：

1. 等待从控制器接收到的数据包中断
2. 检索接收缓冲器的数据包

因为控制传输由 USB 协议控制，所以控制器总是使用 ACK 握手响应。

USB OTG 控制器地址映射和寄存器定义


 地址映射和寄存器定义位于该手册卷附带的 [hps.html](#) 文件中。点击链接以打开文件。

要查看模块说明和基地址，请找到并且点击链接以便查看以下其中一个模块实例：

- **usb0**

- **usb1**

然后要查看寄存器和域说明，找到并且点击寄存器名称。寄存器地址是相对于每个模块实例的基地址的偏移。

 所有模块的基地址也在 *Cyclone V 器件手册* 第 3 卷的 *Introduction to the Hard Processor System* 章节中列出。

文档修订历史

表 18 - 5 显示了该文档的修订历史。

表 18 - 5. 文档修订历史

日期	版本	修订内容
2012 年 11 月	1.2	<div><div>■</div>介绍了中断生成。</div> <div><div>■</div>介绍了主机和器件模式中的软件初始化。</div> <div><div>■</div>介绍了主机和器件模式中的软件操作。</div> <div><div>■</div>简化了功能列表。</div> <div><div>■</div>简化了硬件说明。</div>
2012 年 6 月	1.1	添加了关于 ECC 的说明。
2012 年 1 月	1.0	首次发布。

