

使用 EZ-USB® FX3™ 优化 USB 3.0 吞吐量

作者：Manaskant Desai、Karthik Sivaramakrishnan

相关项目：有

相关器件系列：EZ-USB FX3

软件版本：FX3 SDK v1.3.3

相关应用笔记：要获取完整列表，请点击[此处](#)

要想获得本应用笔记的最新版本或相关项目文件，请访问 <http://www.cypress.com/go/AN86947>。

与前几代产品相比，USB 3.0 可显著提高系统的性能，因为它的信号传输速度达 5 千兆比特每秒（“超速”）。然而，可实现的最大数据吞吐量还取决于多个关键因素，如主机 PC 控制器类型、操作系统以及 USB 的设计（传输类型和缓冲区大小）。本应用笔记描述了使用 EZ-USB® FX3™ 控制器来最大化 USB 3.0 数据传输速度的各种方法。要想获取完整的 USB SuperSpeed 代码示例列表，请参考 <http://www.cypress.com/?rID=101781>。

目录

1	简介	1	11	操作系统性能的比较	14
2	相关资源	2	12	影响吞吐量的其他因素	14
2.1	EZ-USB FX3 软件开发套件	2	13	其他资源	15
2.2	GPiF II Designer（设计程序）	2	14	总结	15
3	测试装置	3		文档修订记录	16
4	性能总结	5		全球销售和设计支持	17
5	操作说明	5		产品	17
6	同步传输	6		PSoC® 解决方案	17
7	批量传输	8		赛普拉斯开发者社区	17
8	中断传输	10		技术支持	17
9	AUTO DMA 通道上 GPiF 到 USB 的性能	11			
10	主机控制器的性能比较	13			

1 简介

赛普拉斯的 EZ-USB FX3 是新一代 USB 3.0 外设控制器，它可提供高度集成性和多项灵活功能，允许您轻松将 USB 3.0 功能添加到任何系统内。USB 3.0 的信号速率为 5 Gbits 每秒，该速度是 USB 2.0 支持的速率的 10 倍。FX3 器件可提供的数据吞吐量接近于 USB 3.0 规格所支持的理论最大值。

FX3 支持下面所有超速传输类型：控制、同步、批量和中断传输但传输大量数据时不应使用控制传输。因此，本应用笔记总结了获取同步、批量和中断等传输类型的最大吞吐量的各指导。

FX3 通常作为一个桥接器件，用以传输 USB 3.0 主机和外部器件（如图像传感器、ASIC 或 FPGA）之间的数据。基于 FX3 应用获取的数据吞吐量取决于下面各种因素：USB 传输类型，传输数据（如突发大小和 FX3 器件中使用的数据缓冲量），以及主机控制器和所使用的 OS 平台。

使用 FX3 开发板时，本应用笔记将通过使用不同的主机和操作系统对策略进行测试和对比。它使用了简单的 FX3 固件示例集来研究每个因素对吞吐量产生的影响。它描述了给定应用的性能优化指南，最终提供了 FX3 配置的详细信息，实现获取给定应用的最大 USB 3.0 吞吐量。

注意： 本应用笔记列出了只通过 FX3 内部生成的数据而得到的 USB 3.0 的吞吐量。当数据是通过 GPIF II 从外部器件传输到 FX3 时，USB 3.0 吞吐量值比较低。请参考 [AN65974](#)，了解 FPGA 将数据写入到 FX3 时的吞吐量

2 相关资源

在赛普拉斯网站 www.cypress.com 上提供了大量资料，有助于选择符合您设计的器件，并能够快速有效地将该器件集成到您的设计中。有关使用资源的完整列表，请参考知识库文章 [KBA87889 — 如何使用 EZ-USB FX3™ 和 FX3S 进行设计](#)。

- **概况：** [USB 产品系列](#)、[USB 路线图](#)
- **USB 3.0 产品选型器：** [FX3](#)、[FX3S](#)、[CX3™](#)、[HX3](#)、[West Bridge® Benicia™](#)
- **应用笔记：** 赛普拉斯提供了大量的 USB 应用笔记，包括了从基本到高级的广泛主题。下面列出的是 FX3 入门的应用笔记：
 - [AN75705](#) — EZ-USB FX3 入门
 - [AN76405](#) — EZ-USB FX3 启动选项
 - [AN70707](#) — EZ-USB FX3/FX3S 硬件设计指南和原理图检查表
 - [AN65974](#) — 使用 EZ-USB FX3 从设备 FIFO 接口进行设计
 - [AN75779](#) — 如何使用 EZ-USB FX3 在 USB 视频类别（UVC）框架内实现图像传感器连接
 - [AN86947](#) — 使用 EZ-USB FX3 优化 USB 3.0 的吞吐量
 - [AN84868](#) — 使用赛普拉斯 EZ-USB FX3 通过 USB 配置 FPGA
 - [AN68829](#) — 用于 EZ-USB FX3 的从设备 FIFO 接口：5 位地址模式
 - [AN73609](#) — 在 Linux 上的 EZ-USB FX2LP™/FX3 开发 Bulk-Loop 示例
 - [AN77960](#) — EZ-USB FX3 高速 USB 主机控制器简介
 - [AN76348](#) — EZ-USB FX2LP 和 EZ-USB FX3 应用的区别
 - [AN89661](#) — 使用 EZ-USB FX3S 设计 USB RAID 1 磁盘
- **代码示例：**
 - [USB 高速](#)
 - [USB 全速](#)
 - [USB 超速](#)
- **技术参考手册（TRM）：** [EZ-USB FX3 技术参考手册](#)
- **开发套件（DVK）**
 - [CYUSB3KIT-003](#)，EZ-USB FX3 超速浏览器套件
 - [CYUSB3KIT-001](#)，EZ-USB FX3 开发套件
- **模型：** [IBIS](#)

2.1 EZ-USB FX3 软件开发套件

赛普拉斯为 FX3 提供了完整的软件和固件堆栈，这样您很容易便能够将超速 USB 集成到嵌入式应用内。[软件开发套件（SDK）](#) 中带有各种工具、驱动程序和应用示例，有助于加快应用开发程序。

2.2 GPIF II Designer（设计程序）

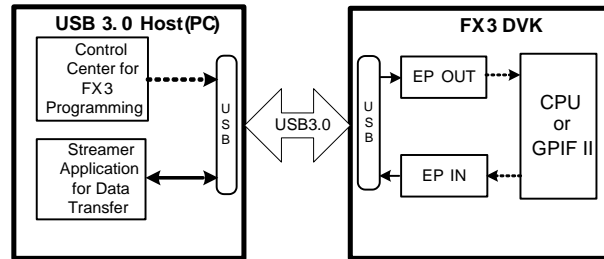
[GPIF II Designer](#) 是一个图形软件，通过该软件您可以配置 EZ-USB FX3 USB 3.0 器件控制器的 GPIF II 接口。它还允许从赛普拉斯所提供的五个接口中选择一个，或从头创建您自己的 GPIF II 接口。赛普拉斯提供了各种符合工业标准的接口，如异步和同步从设备 FIFO、同步 SRAM 以及异步 SRAM。如果您的系统中装有了上面所述接口，您可从标准参数集（如总线宽度（x8、16、x32）、字节顺序、时钟设置）中选择所需要的接口。然后，编译已选定的接口。

如果您需要自定义接口，该工具提供一个简洁的三步骤 **GPIF** 接口开发程序，允许您先选择引脚配置和标准参数。然后，您可以使用可配置操作设计一个虚拟的状态机。最后，通过查看输出时序，您可以验证它是否与所需时序相匹配。一旦完成这三步骤程序，便可以对接口进行编译并将其集成到 **FX3** 内。

3 测试装置

图 1 显示了本应用笔记的测试设置。通过 **USB 3.0** 连接器将 **FX3 DVK** 连接至 **PC 主机**，可以执行测试。

图 1. 测试设置



使用控制中心应用将固件图像下载到 **FX3** 的内部 **RAM** 中。下载后，**FX3** 可以使用 **Streamer** 的应用进行数据传输操作。为了便于使用，附带文件中提供了控制中心、**C++ Streamer** 的应用二进制文件以及赛普拉斯 **USB 3.0** 驱动程序 (**CyUSB3.sys**) 等。附带文件所提供的 **Streamer** 应用已被配置，以实现最佳性能数量。

要想下载最新的 **FX3 SDK**，请访问：www.cypress.com/?rID=57990。

要查找 **Streamer** 应用的 **C++** 版本，请在安装 **SDK** 后访问下面的路径：

C:\Program Files\Cypress\EZ-USB FX3 SDK\1.3\application\cpp\streamer\x86\Release

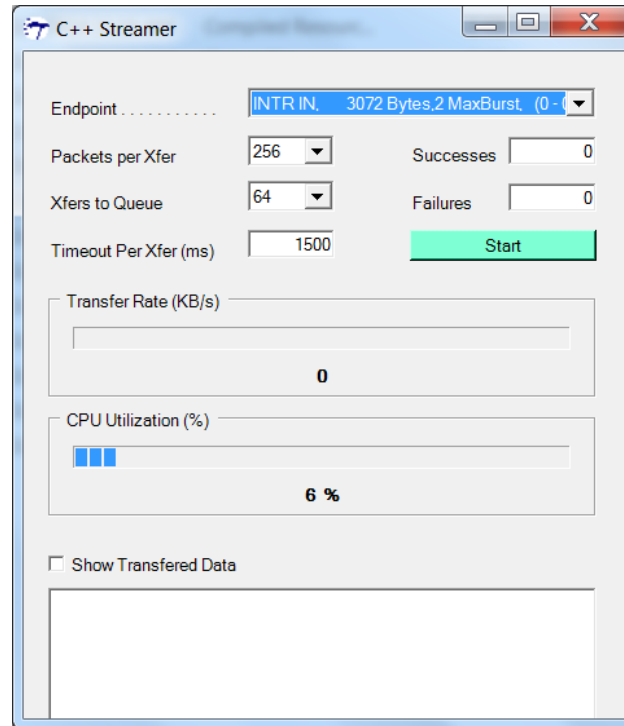
注意： 在 64 位 **Windows** 电脑上，程序文件的目录是“**Program Files (x86)**”。

表 1 列出了四种传输测试。图 2 显示的 **Streamer** 应用支持这些测试，并将其作为“吞吐量表”。该应用笔记使用了 **Streamer** 应用的 **C++** 版本来测量三种传输类型（同步、批量和中断）的性能。请注意，由于应用和驱动器等级开销，**C#** 版本提供了较低的吞吐量。

表 1. 四种传输测试的代码映射

测试	FX3 代码映射	说明
1	USBIsoSourceSink	EP1-IN 是 ISO 数据的常量源；EP1-OUT 是 ISO 数据的常量输出。不存在任何 GPIF II 连接。
2	USBBulkSourceSink	与第 1 种测试相似，但这里使用了 BULK （批量）传输。
3	USBIntrSourceSink	与第 1 种测试相似，但这里使用了中断传输。
4	GpifToUsb	使用 GPIF II 以获得传输到 EP-IN 的最快速度，从而对 PC 进行批量传输操作。这是唯一使用图 1 中从 GPIF II 到 EP1-IN 的虚线路径进行的测试。

图 2. C++ Streamer



该 Streamer 应用包含下面各项：

- **Endpoint**（端点）：通过该下拉列表，您可以选择不同的传输类型和端点缓冲区数量。
- **Packets per Xfer**（每传输操作的数据包数量）：一个传输操作指的是一组数据的数据包集合。每个传输的数据包越多，则开销越少，取得的数据速率也会越高。
- **Xfers to Queue**（队列的传输操作）：该设置有助于启动多个传输，并把它们添加到任务队列中。该项减小了主机应用方上连续传输间的延迟。因此，队列的传输次数越多，数据速率也越高。
- **Successes**（成功）：表示在流式测试期间成功传输的数据包总数的增量。
- **Failures**（失败）：随缓冲区传输的错误报告而增高。器件中没有任何数据会引发故障。
- **Transfer Rate**（传输速率）：提供有关所选端点中 USB 总线和 EZ-USB FX3 当前吞吐量性能的实时更新。
- **CPU Utilization**（CPU 的使用）：提供电脑 CPU 在 USB 流式传输期间利用率的视觉指示。
- **Timeout Per Xfer**（每个 Xfer 的超时）：如果在超时期间没有得到来自器件的数据，则传输失败。默认值为 1500 ms。

表 1 列出了四种代码映射（.img 文件），将这些文件下载到 FX3 DVK 中实现测试带宽。这些模块位于本应用笔记附赠的 .zip 文件中。

- **USBIsoSourceSink** 示例用于测量超速同步传输的性能。该示例使用了一对 IN 和 OUT 端点，以连续输入或输出 FX3 器件端的 ISO（同步）数据。
- **USBBulkSourceSink** 示例用于测量超速批量传输的性能。该示例也使用了一对 IN 和 OUT 端点，以便连续输入或输出 BULK 数据。
- **USBIntrSourceSink** 示例用于测量超速中断传输的性能。
- **GpiToUsb** 示例用于测量 GPIF 到 USB 的数据批量传输的性能。该示例连续读取来自 GPIF II 基本接口的数据，然后再通过 IN 端点将这些数据发送给 USB 主机而不需要任何固件的参与。

注意： 这些测试传输的 FX3 端不会带来开销，如处理 FIFO 中的数据或转移片下数据。开销的处理取决于应用，可以降低本应用笔记中需要测量的最大吞吐量。请谨慎使用 FX3 GPIF II 和 DMA 的功能，以避免降低吞吐量。此文档的 [第 12 节](#) 介绍的是性能提示，用以最大化 FX3 吞吐量。

在所有代码示例中，指令缓存被使能，而数据缓存被禁用。推荐保持数据缓存的 OFF 状态，直到固件应用需要执行正常的数据操作为止。

使用 FX3 DVK 直接连接到 PC 主机来测试各示例（主机使用 Intel C216 芯片集系列 USB 3.0 eXtensible 主机控制器 (XHCI)）。使用的操作系统是 64 位 Windows 7，Intel 主机驱动器的版本为 1.0.5.235。

4 性能总结

[表 2](#) 总结了分别使用三个 SourceSink 固件版本的三种传输类型的性能。该表中的突发长度¹是指 USB 描述符中上报给主机的突发大小，“数据包数量”是指在每个服务间隔内所传输的同步数据包的数量。该参数仅适用于同步和中断这两种传输对于中断传输，数据包数量始终为 1。欲了解吞吐量分析的详情，请参阅第 6、7、8 和 9 的各节。

表 2. 性能总结结果

传输	突发长度	数据包数量	缓冲区大小 (KB)	缓冲区的数量	吞吐量 (KB/s)
同步	16	3	48	2	382,700
批量	16	NA	48	2	454,300
中断	3	1	3	1	23,900

5 操作说明

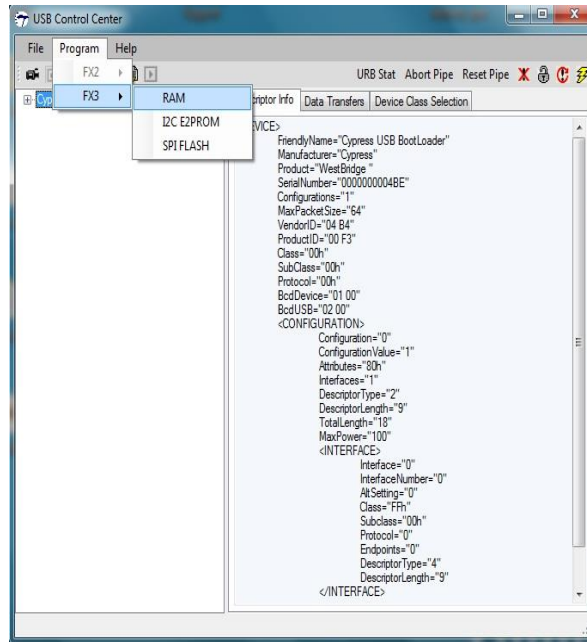
如果第一次使用 FX3DVK，请参考 [AN75705 — EZ-USB FX3 入门应用笔记](#)。您可以根据下面的指导对前面所有传输类型的吞吐量进行测量：

1. 下载本应用笔记附带的 .zip 文件，该文件包含固件、控制中心、C++ Streamer 的应用以及赛普拉斯 USB 3.0 驱动程序。
2. 通过 USB 控制中心（如 [图 3](#) 所示）下载附带文件中四个映射文件的其中一个。也可以在 FX3 SDK 的安装包中找到它（<install directory>\Cypress\EZ-USB FX3 SDK\1.3\application\c_sharp\controlcenter\bin\Release）。
3. 打开附带文件中所提供的 C++ Streamer 应用，选择如 [图 2](#) 所示的参数，然后单击 **Start**（启动）按钮。

数据吞吐量的单位为 KB/s

¹ USB3.0 引用了数据突发，允许通过端点传输某些数据包，而不需要数据包间的交换。单独端点通过与特殊端点相关的超速端点同伴描述符，将上报给主机其突发的能力（即每个突发的最大数据包数量）。

图 3. USB 控制中心



6 同步传输

同步传输类型适用于数据流应用，如音频和视频。同步传输为超速总线提供得到保证的数据带宽，但不包含交换或重试。

超速同步端点支持的最大数据包负载大小为 1024 个字节。因为数据的生成者不需要等待一个 ACK 直到完成指定的突发大小，因此突发传输会提高数据的传输速率。FX3 器件支持的最大突发大小为 16，这是由 USB 规范定义的。但每个超速同步端点可以根据 USB 3.0 规范，在相同的服务间隔²内要求最多三个突发传输（ISO 数据包）。

可将同步传输的服务间隔指定为 $2^{(bInterval-1)} \times 125 \mu s$ ，其中 $bInterval$ 值的范围为 1 到 16。超速同步传输类型在每个服务间隔中都能够传送多达 $1024 \times 16 \times 3$ 个字节。最小的服务间隔为 $125 \mu s$ ，因此可以计算出超速同步传输的最大理论带宽为（最大数据包大小 * 突发大小 * ISO 数据包数量 / 服务间隔）。它的工作速度可高达 $1024 \times 16 \times 3 / 125 \mu \text{ bytes/s}$ 或 375 Mbps（3 Gb/s）。

使用“USBIsosourceSink”固件示例来测量 FX3 的超速同步端点吞吐量。吞吐量取决于突发长度、缓冲区大小³、iso 数据包以及所使用的 DMA 缓冲区数量。

通过 *cyfxisosrcsink.h* 头文件中的定义，可以更改这些参数：

- 使用 CY_FX_ISO_BURST 定义来设置同步端点的突发长度
- 使用 CY_FX_ISO_PKTS 定义来设置每个服务间隔中 ISO 数据包突发的数量
- 使用 CY_FX_ISOSRCSINK_DMA_BUF_SIZE 定义来设置用于传输的每个 DMA 缓冲区的大小。
- 使用 CY_FX_ISOSRCSINK_DMA_BUF_COUNT 定义来设置每个端点所使用的 DMA 缓冲区数量。

默认情况下，分别将 CY_FX_ISO_BURST 和 CY_FX_ISO_PKTS 参数设置为 15 和 3。某些 USB 主机可能不支持该设置所需的传输带宽，因此它们无法选择配置。在这种情况下，您可以通过缩小这些参数的某个值来降低带宽的要求。

² 对于中断和同步端点，主机必须提供服务的端点指定时间间隔被称为服务间隔。通过端点描述符可以指定中断或同步管道的服务间隔。

³ 启动 FX3 固件中的 DMA 通道时，将配置 FX3 DMA 缓冲区。

更改任何一个参数后，您需要重新编译应用。可以将结果固件二进制文件（*USBIsoSourceSink.img*）下载到 FX3 器件的 RAM 内，然后可以通过 Streamer 应用进行测量传输性能。

通过改变突发长度、ISO 数据包和缓冲区数量所得到的同步吞吐量结果被列出在表 3 和表 4 中。从上述结果来看，使用同步端点时，FX3 可提供的性能接近于理论带宽。因此，超速同步吞吐量主要取决于每个传输的突发长度和 ISO 数据包数量。

注意： 与突发长度为 15 的 IN 传输相比，突发长度为 16 的 IN 传输的吞吐量明显降低。原因在于 USB 主机处理请求较慢，不是由 FX3 器件所支持的数据传输速率发生变化而引起的。可以通过改变端点描述符来验证，即上报突发长度为 15 的同时仍然保持器件端点的突发长度配置为 16。

表 3. 同步 IN 端点吞吐量结果

突发长度	Iso 数据包	缓冲区大小 (KB)	缓冲区的数量	吞吐量 (KB/s)	理论吞吐量 (KB/s)
8	1	8	1	63,900	64,000
8	3	24	1	191,900	192,000
12	1	12	1	96,000	96,000
12	3	36	1	287,900	288,000
15	3	15	1	119,800	360,000
15	3	15	2	239,500	360,000
15	3	45	2	359,900	360,000
16	3	48	2	139,400	384,000

注意： 在 Streamer 应用中，选择 **256 Packets per Xfer** 和 **64 Xfers to Queue** 可测量这些吞吐量。如果您正在使用的是附件的 Streamer 应用，则这些设置已被预先选择。

表 4. 同步 OUT 端点吞吐量结果

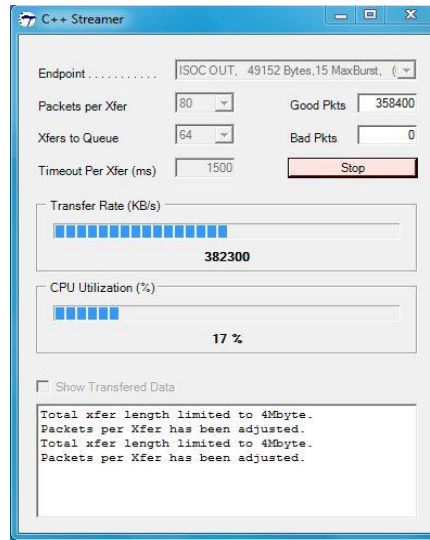
突发长度	Iso 数据包	缓冲区大小 (KB)	缓冲区的数量	吞吐量 (KB/s)	理论吞吐量 (KB/s)
8	1	8	1	63,900	64,000
8	3	24	1	191,900	192,000
12	1	12	1	96,000	96,000
12	3	36	1	287,900	288,000
15	3	45	2	359,900	360,000
16	3	48	1	382,700	384,000
16	3	16	2	251,200	384,000

注意： 在 Streamer 应用中，选择 **256 Packets per Xfer** 和 **64 Xfers to Queue** 可测量这些吞吐量。如果您正在使用的是附件的 Streamer 应用，则这些设置已被预先选择。

如这些结果所示，推荐使用够大的 DMA 缓冲区，这样可以保持一个服务间隔的数据值，从而优化结果。FX3 固件处理每个 DMA 缓冲区的时间约为 40 μ s，该时间比服务间隔持续的时间短得多。如果在服务间隔期间能够将需要传输的所有数据装载到一个 DMA 缓冲区内，那么仅需要一个 DMA 缓冲区即能获取最佳的传输速率。

Streamer 应用显示了同步吞吐量，如图 4 所示。如您所看到，为 **Packets per Xfer** 和 **Xfers to Queue** 选项最大的容许值可以实现最佳性能。

图 4. 同步 OUT 端点吞吐量



7 批量传输

对于需要以不同速率发送大量数据的器件而言，批量传输方式是最合适的。在这种情况下，传输可以使用所有可用的带宽。超速批量传输根据可用带宽来使用总线，并确保数据传输和数据的完整性，但不能保证提供任何带宽。批量传输类型采用了更多的高数据传输速率的应用，如批量存储器件。当主机能够维持所需的传输速率时，还可将它用于视频数据传输。

超速批量端点支持的最大数据包负载大小为 1024 个字节。这些端点也支持的突发大小为 1~16。（“突发”指的是不需要接收端的单独 ACK 信令的一系列 BULK 数据包）。

由于没有给批量传输指定任何固定带宽，因此无法确定吞吐量的最大理论值。在考虑到分配给所有连接至相同 USB 主机的器件后，该值取决于可用的带宽。如果整个带宽可用于一个批量传输，则将 20% 预留给链路和协议级开销的传输后，批量传输的最大理论吞吐量约为 4 Gbit/s。

USBulkSourceSink 示例用于测量 FX3 超速批量端点的吞吐量。该值取决于突发大小、缓冲区大小以及缓冲区的数量。

通过更改 `cyfxbulksrscsink.h` 头文件中的定义，可以更改这些参数：

- 使用 `CY_FX_EP_BURST_LENGTH` 定义来设置端点的突发长度。
- 使用 `CY_FX_BULKSRCSINK_DMA_BUF_SIZE` 定义来设置用于传输数据的每个 DMA 缓冲区的大小。
- 使用 `CY_FX_BULKSRCSINK_DMA_BUF_COUNT` 定义来设置每个端点上使用的 DMA 缓冲区数量。

表 5 和表 6 中列出了因改变突发长度和缓冲区大小而得到的同步吞吐量结果。

表 5. 批量 IN 端点吞吐量结果

突发长度	缓冲区大小 (KB)	缓冲区的数量	吞吐量 (KB/s)
8	16	1	263,100
8	16	2	448,300
12	24	1	305,000
12	24	2	450,000
16	48	2	454,300
16	16	2	351,800
16	16	1	120,500

注意： 在 Streamer 应用中，选择 **256 Packets per Xfer** 和 **64 Xfers to Queue** 可测量这些吞吐量。如果您正在使用的是附件的 Streamer 应用，则这些设置已被预先选择。

表 6. 批量 OUT 端点吞吐量结果

突发长度	缓冲区大小 (KB)	缓冲区的数量	吞吐量 (KB/s)
8	16	1	249,900
8	16	2	360,100
12	24	1	289,000
12	24	2	377,000
16	48	2	405,000
16	16	2	364,900
16	16	1	153,800

注意： 在 Streamer 应用中，选择 **256 Packets per Xfer** 和 **64 Xfers to Queue** 可测量这些吞吐量。如果您正在使用的是附件的 Streamer 应用，则这些设置已被预先选择。

从上述结果可以看出，在最佳状态（没有其他共享总线带宽并连接到 USB 主机的器件）下，批量传输能比同步传输提供的吞吐量更大。这些表指出了超速批量端点吞吐量取决于突发大小、缓冲区大小以及所使用的 DMA 缓冲区数量。

如同步传输，使用更大的 DMA 缓冲区可保存更多的数据突发，从而提高系统性能。通过下面计算，说明了吞吐量对缓冲区大小的关系：

如果突发大小为 16，则可获取的最佳吞吐量是 454,300 KB/s。

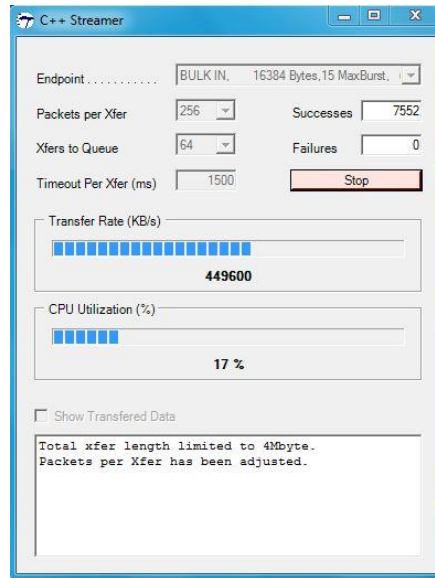
- 每秒处理的 16 KB 突发数量为 28,400
- 传输 16 KB 数据所需的平均时间为 35 μ s
- 传输 32 KB 数据所需的平均时间为 70 μ s

固件应用通过调用 CyU3PDmaChannelGetBuffer 和 CyU3PDmaChannelCommitBuffer API 来轮流使用每个 DMA 缓冲区。性能基准测试表明 CPU 需要大概 40 μ s 的时间来调用 GetBuffer 和 CommitBuffer API。

从表 5 和表 6 可以看出，将每个 DMA 缓冲区限制为一次突发（16 KB）或更低的值时，会引起处理固件操作设置吞吐量范围。如果将每个缓冲区大小增大为 32 KB（两次突发传输，每次 16 KB），可以确保一个缓冲区的传输平均时间比缓冲区的固件处理的平均时间更久。这样的设置可使批量数据传输的性能最佳，固件也参与了该传输操作。

批量 IN 端点的最大吞吐量为 454,300 KB/s 或 3.7 Gbits/s，批量 OUT 端点的最大吞吐量为 405,000 KB/s 或 3.31 Gbits/s。Streamer 应用指出批量吞吐量，如图 5 所示。

图 5. 批量 IN 端点的吞吐量



8 中断传输

对于要求数据可靠性高并要保证最大服务间隔的器件，中断传输类型最为合适。超速中断传输通过使用交换和重试的方式提供了一个受保证的服务间隔和数据传输。中断传输类型更适合于 HID 器件（如鼠标和键盘），而不经常使用在吞吐量的关键应用中。

超速中断端点支持的最大数据包负载大小为 1024 个字节。根据 USB 3.0 的规范，超速中断传输支持的最大突发大小只为 3 个数据包。此外，每个服务间隔仅支持一个突发传输。中断传输的服务间隔定义与同步传输相似。使用一个微帧或 125 μ s 的服务间隔时，超速中断传输的最大理论带宽计算方法为：（最大的数据包大小 * 突发大小 / 服务间隔），即（1024 x 3 / 125 μ ）bytes/s 或 23.43 Mbps。

“USBIntrSourceSink”示例用于测量 FX3 超速中断端点的吞吐量。如同步传输示例，选择 DMA 缓冲区大小等于每个服务端点需要传输的数据量。只需要一个 DMA 缓冲区即可获取最大的吞吐量。

使用 `cyfxintrsrcsink.h` 头文件中的 `CY_FX_INTR_BURST_SIZE` 定义来设置中断端点的突发大小。

中断传输所获取的吞吐量接近于理论范围，并且与 IN 和 OUT 端点的相同。表 7 显示的是中断传输使用不同的突发大小设置时所获取的的吞吐量。

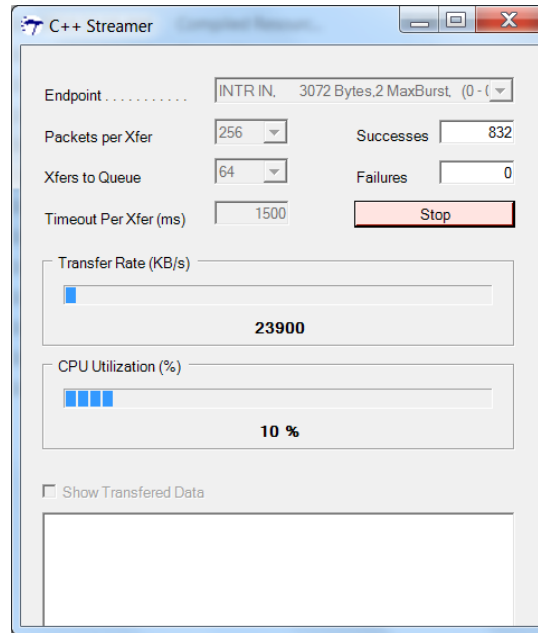
表 7. 中断 IN/OUT 端点吞吐量结果

突发长度	缓冲区大小 (KB)	缓冲区的数量	吞吐量 (KB/s)	理论吞吐量 (KB/s)
1	1	1	7900	8000
2	2	1	15,900	16,000
3	3	1	23,900	24,000

注意： 在 Streamer 应用中，选择 **256 Packets per Xfer** 和 **64 Xfers to Queue** 可测量这些吞吐量。如果您正在使用的是附件的 Streamer 应用，则这些设置已被预先选择。

Streamer 应用指出中断吞吐量，如图 6 所示。

图 6. 中断 IN 端点的吞吐量



9 AUTO DMA 通道上 GPIF 到 USB 的性能

上述所有示例中所获得的传输性能都受固件执行的影响，并且将 DMA 缓冲区大小保持足够大时，得到的结果更好。通过本应用笔记所附加的 `GpifToUsb` 固件示例可测量出在不同端点突发设置、突发大小以及 DMA 缓冲区大小的情况下，数据传输如何变化。

该示例使用了一个微型 GPIF 状态机；每当 GPIF 端的 DMA 缓冲区可用时，该状态机便会持续填充数据。该状态机不需要外部器件来驱动任何信号，并且只在最快的可能速率时才获取填满的 DMA 缓冲区。本应用笔记附带了该状态机的 GPIF II Designer 项目，它位于 `continuous_read.cydsn` 文件夹中。

通过 AUTO DMA 通道（而不需要使用任何固件），GPIF II 数据将经过批量 IN 端点被发送到 USB 主机。在该示例的数据路径中没有任何固件的参与，因此可使用这些结果来确定突发大小、缓冲区大小和缓冲区数量对传输吞吐量的影响。

使用 `cyfxgpiptousb.h` 头文件中的定义可以改变这些参数。

- 使用 `CY_FX_EP_BURST_LENGTH` 定义来设置端点的突发大小。
- 使用 `CY_FX_DMA_BUF_SIZE` 定义来设置所使用的每个 DMA 缓冲区大小。
- 使用 `CY_FX_DMA_BUF_COUNT` 定义来设置使用的 DMA 缓冲区数量。

通过 GPIF 接口的数据速率来决定理论的最大性能。在这种情况下，GPIF 接口在时钟率为 100.8 MHz 和 32 位宽的数据总线下运行。因此最大可能数据速率为 403.2 MB/s。

图 7 说明了在不同的突发长度设置情况下，使用不同的 DMA 缓冲区数量时所产生的不同传输性能。图形中的每条线代表不同的 DMA 缓冲区数量，其范围从单个保存一次突发数据的缓冲区到四个保存三次突发数据的缓冲区。

使用一个 DMA 缓冲区时所得到的性能很低，因为 GPIF 传输暂停的同时，USB 传输则正在执行，反之亦然。该图形显示了传输性能得到提高，到突发长度达到 8 KB 时为止，然后逐渐降低。它还显示当缓冲区大小达到 4 次突发传输的缓冲区总大小时，会提高传输性能，然后逐渐下降。

该图同样显示了基于批量 GPIF 到 USB 传输的应用的最佳设置是：突发长度为 8 KB，缓冲区大小约为 32 KB。进一步增加突发长度和缓冲区大小只能提高边缘的性能，但是将增大使用 RAM 空间。

图 7. 突发大小引起的 USB 性能变化

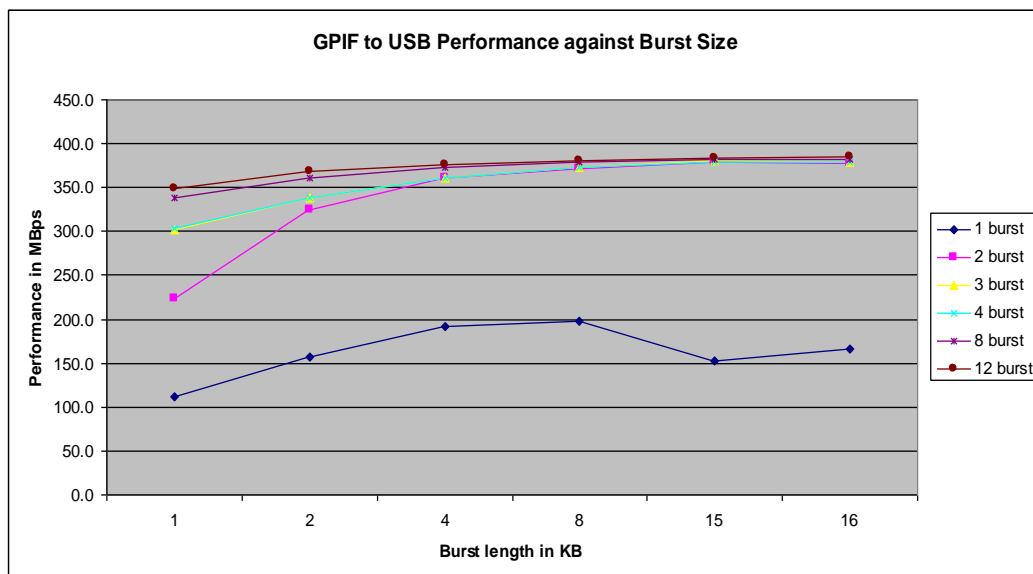


表 8. 显示了通过该固件针对不同的突发长度、DMA 缓冲区大小和 DMA 缓冲区数量所获取的吞吐量。它显示使用不同突发长度的单独 DMA 缓冲区大小时 GPIF 到 USB 传输性能的变化。在使用四个 DMA 缓冲区的情况下，不同之处仅在于单独 DMA 缓冲区的大小。

如图 8 所示，同保存一次突发传输数据的 DMA 相比，能保存两次突发传输数据的缓冲区可显著提高传输的性能。进一步增大缓冲区大小几乎不起任何作用。该图显示了 DMA 缓冲区的理想尺寸为端点突发长度的两倍。

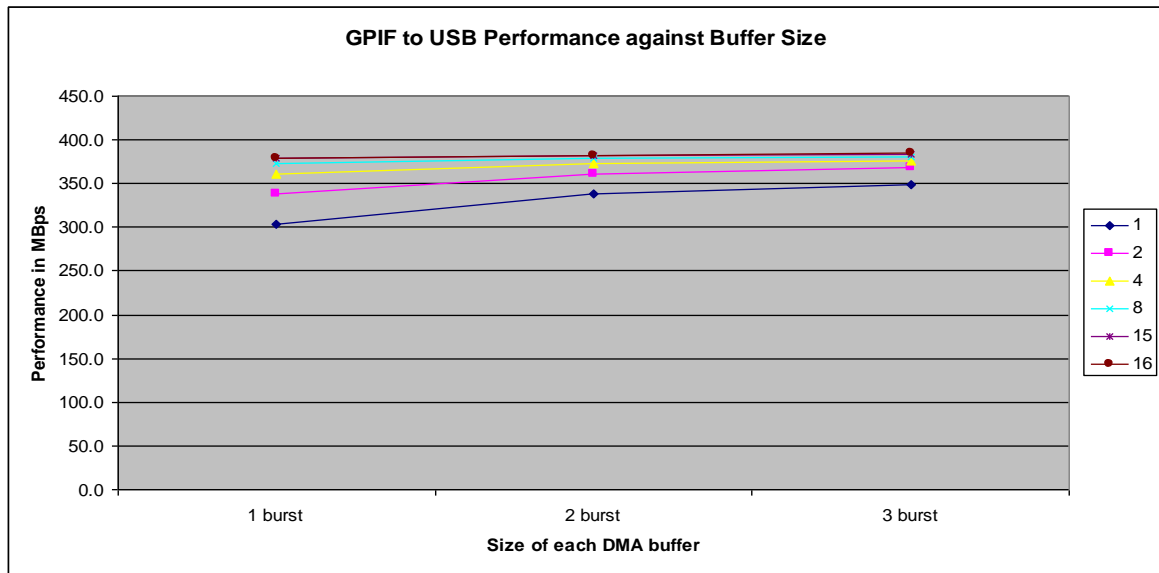
表 8. GPIF 到 USB 批量传输的吞吐量结果

突发长度 (KB)	每个 DMA 缓冲区的大小	使用一个 DMA 缓冲区的传输性能 (KB/s)	使用两个 DMA 缓冲区的传输性能 (KB/s)	使用三个 DMA 缓冲区的传输性能 (KB/s)	使用四个 DMA 缓冲区的传输性能 (KB/s)
1	一个突发	115,200	229,000	308,500	310,800
1	两个突发	—	—	—	346,700
1	三个突发	—	—	—	356,800
2	一个突发	160,600	333,100	346,000	347,100
2	两个突发	—	—	—	369,600
2	三个突发	—	—	—	377,300
4	一个突发	196,000	368,800	369,600	370,100
4	两个突发	—	—	—	381,900
4	三个突发	—	—	—	385,600
8	一个突发	202,100	380,700	382,100	382,100
8	两个突发	—	—	—	387,700
8	三个突发	—	—	—	389,500
16	一个突发	170,700	386,900	387,700	388,600
16	两个突发	—	—	—	391,200

突发长度 (KB)	每个 DMA 缓冲区的大小	使用一个 DMA 缓冲区的传输性能 (KB/s)	使用两个 DMA 缓冲区的传输性能 (KB/s)	使用三个 DMA 缓冲区的传输性能 (KB/s)	使用四个 DMA 缓冲区的传输性能 (KB/s)
16	三个突发	—	—	—	394,000

注意： 在 Streamer 应用中，选择 **256 Packets per Xfer** 和 **64 Xfers to Queue** 可测量这些吞吐量。如果您正在使用的是附件的 Streamer 应用，则这些设置已被预先选择。

图 8.DMA 缓冲区大小引起的 USB 性能变化



10 主机控制器的性能比较

USB 数据的吞吐量还取决于所使用的 USB 主机功能。为了说明这种依赖性，对各种 USB 3.0 主机控制器测量超速批量 IN 的吞吐量，同时保持所有其他测试参数相同。可以在 Windows 7 下运行的多电脑版上使用 USBBulkSourceSink 示例进行测试。

表 9 显示了四个内置 USB 3.0 主机控制器的吞吐量的比较。如下表所示，Intel 的主机控制器的性能比其他 USB 3.0 主机控制器高出约 20%。通常，与内置的 USB 3.0 主控制器相比，卡带 USB 3.0 的吞吐量将更低。

表 9. 主机控制器的 BULK IN 的吞吐量

USB 3.0 主机控制器 (内置)	吞吐量 (KB/s)	驱动程序版本	PC 信息	OS	应用	突发长度	最大数据包大小 (字节)	数据包数量/传输	Xfers to Queue (Xfer 排队)
Intel® USB 3.0 eXtensible 主机控制器	450,400	1.0.9.254	Intel Core™ i5-3210M CPU, Intel(R) 7 系列 /c216 芯片集系列, 2.5 GHz, 4 GB RAM, Service Pack 1	Win 7 64 Bit	C++ Streamer	16	1024	256	64
Renesas Electronics 的 USB 3.0 主机控制器	352,100	2.0.34.0	Intel Core i7 CPU, 芯片集: Intel 3400 系列, 4 GB 的 RAM, Service Pack 1	Win 7 64 位	C++ Streamer	16	1024	256	64

USB 3.0 主机控制器 (内置)	吞吐量 (KB/s)	驱动程序版本	PC 信息	OS	应用	突发长度	最大数据包大小 (字节)	数据包数量/传输	Xfers to Queue (Xfer 排队)
ASMedia 主机控制器	370,200	1.12.5.0	AMD FX™-4100 四线内核处理器, 8GB 的 RAM, 3.60 GHz, Service Pack 1	Win 7 64 位	C++ Streamer	16	1024	256	64
AMD USB 3.0 主机处理器	362,900	1.1.0.153	AMD A6-3670 APU, 2.70 GHz, 8 GB 的 RAM, Service Pack 1	Win 7 64 位	C++ Streamer	16	1024	256	64
NEC Electronics USB 3.0 主机控制器	300,200	1.0.19.0	Intel Core i5-2540M CPU, 2.60 GHz, 4 GB 的 RAM, Service Pack 1	Win 7 64 位	C++ Streamer	16	1024 字节	256	64

11 操作系统性能的比较

USB 吞吐量还取决于操作系统以及 USB 主机上运行的软件。

对各种操作系统测量超速批量 IN 的吞吐量, 同时保持所有其他测试参数相同。通过使用提供最大数据传输速率的 USBBulkSourceSink 应用进行该测试。所有操作系统 (Windows、Mac 和 Linux) 都使用了 Intel 3.0 主机。

针对不同的操作系统测试超速同步 IN 吞吐量。通过使用提供最大数据速率的 USBIsoSourceSink 应用进行该测试。所有操作系统都使用了 Intel 3.0 主机。

通过基于开放源 Libusb 库的控制台应用来测量 Linux 和 Mac OS X 操作系统的吞吐量。要想使用该控制台应用, 请通过下面的网站下载 Linux 的 FX3 SDK 以及 Mac OS 的 FX3 SDK: www.cypress.com/?rID=57990。

表 10 和表 11 对各项性能进行了比较。从结果中可以看出, 与 Windows 和 Mac OS X 相比, Linux 的同步吞吐量较少。这是因为 Linux 内核将 USB 传输的大小限制为小于或等于 32 KB。因此, 应设置突发大小和 ISO-Mult 设置, 使其产品不超过 32 KB。

表 10. 操作系统的 BULK IN 吞吐量

操作系统	Windows 7 (Intel 主机)	Windows 8 (Intel 主机)	Windows 10 (Intel 主机)	Mac OS X 版本 10.10 (Intel 主机)	Linux (kernel 3.11.0-12, Intel 主机)
吞吐量 (KB/s)	454,300	453,500	454,400	420,100	425,400

表 11. 操作系统的同步 IN 吞吐量

操作系统	Windows 7 (Intel 主机)	Windows 8 (Intel 主机)	Windows 10 (Intel 主机)	Mac OS X 版本 10.10 (Intel 主机)	Linux (kernel 3.11.0-12, Intel 主机)
吞吐量 (KB/s)	359,800	359,900	359,900	356,400	256,000

12 影响吞吐量的其他因素

如前面所述, 吞吐量的不同取决于主机控制器和操作系统。如果将多个 USB 器件连接到同一个主机, 则会降低吞吐量。赛普拉斯推荐使用一个符合 USB-IF 标准的 USB 线缆, 以获取最佳结果。与集成的 USB 主机控制器相比, 连接到主机适配器卡的 USB 器件将提供更低的吞吐量。

在需要将数据传出/传入外部器件的应用中（这些器件通过 **GPIO II** 接口进行连接），吞吐量也取决于 **GPIO II** 数据传输速率以及 CPU 处理时间（如果它需要接触数据）。

如果要为这样的传输使用批量端点，请尽量加大 **DMA** 缓冲区的数量。这样有助于抵消 **USB** 主机端在传输数据过程中长期停止对吞吐量产生的影响。

使用同步传输时，赛普拉斯建议每个 **DMA** 缓冲区能够保存一个服务间隔数据。因为在这种情况下主机提供了传输带宽保障，无需使用大量缓冲区。

如果 **FX3** 固件当前没有进行任何数据操作，请使用 **AUTO DMA** 通道，以避免因固件设计所导致的性能限制。如果需要固件进行数据操作，则通过 **DMA** 回调发送数据将比从线程发送数据更快。比如，在 **USB** 视频类别的应用中，固件将标头信息添加到数据块内。此外，赛普拉斯强烈推荐通过删除调试信息将处理代码的关键数据保持为最小值。[AN75779 — 如何在 **USB** 视频类别（**UVC**）框架中使用 **EZ-USB® FX3™** 实现映射传感器接口](#) 提供了固件如何实现 **UVC** 的示例。）

建议使用 **FX3 API** 的发布版本和 **RTOS** 库；将编译器的优化级别置为“-O2”或“-O3”。

13 其他资源

[AN65974 — 使用 **EZ-USB® FX3** 从设备 **FIFO** 接口进行设计](#) 说明了有关从设备 **FIFO** 接口的详细信息，并包含了从设备 **FIFO** 应用的特性测量。请参阅“流式传输的流程”一节，了解使用从设备 **FIFO** 应用所测量到的吞吐量。

请参考 **FX3 SDK** 故障排除指南中第 2.4 节，了解有关 **USB** 连接级别对系统吞吐量的影响因素。本文档是 **FX3 SDK** 安装程序的一部分，位于下面的文件夹内：**C:\Program Files\Cypress\EZ-USB FX3 SDK\1.3\doc**。欲了解更多有关 **USB** 超速协议的信息，请参阅 **USB 3.0 规范**（<http://www.usb.org/developers/docs/>）。

14 总结

本应用笔记提供了使用 **EZ-USB FX3** 来获取 **USB 3.0** 最佳吞吐量的指南。我们已经对不同的传输参数所引起的吞吐量变化进行了分析，从而确认每个传输类型中最重要的参数。同样还对主机控制器的操作系统的吞吐量进行了比较，从而说明影响 **USB** 吞吐量的因素。

文档修订记录

文档标题: AN86947 — 使用 EZ-USB® FX3™ 优化 USB 3.0 的吞吐量

文档编号: 001-98020

版本	ECN	变更者	提交日期	变更说明
**	4802498	WEIZ	07/28/2015	本文档版本号为 Rev**, 译自英文版 001-86947 Rev*A。
*A	5713928	AESATP12	04/27/2017	更新了徽标和版权。
*B	6333009	LIP	04/10/2018	本文档版本号为 Rev*B, 译自英文版 001-86947 Rev*D。

全球销售和设计支持

赛普拉斯公司拥有一个由办事处、解决方案中心、厂商代表和经销商组成的全球性网络。要想查找离您最近的办事处，请访问 [赛普拉斯所在地](#)。

产品

ARM® Cortex® 微控制器	cypress.com/arm
汽车级产品	cypress.com/automotive
时钟与缓冲区	cypress.com/clocks
接口	cypress.com/interface
物联网	cypress.com/iot
存储器	cypress.com/memory
微控制器	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
触摸感应	cypress.com/touch
USB 控制器	cypress.com/usb
无线连接	cypress.com/wireless

PSoC® 解决方案

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6](#)

赛普拉斯开发者社区

[Forums](#) | [WICED IOT Forums](#) | [Projects](#) | [Videos](#) | [Blogs](#) | [Training](#) | [Components](#)

技术支持

cypress.com/support

此处引用的所有其它商标或注册商标都归其各自所有者所有。



赛普拉斯半导体公司
198 Champion Court
San Jose, CA 95134-1709

© 赛普拉斯半导体公司，2013-2018 年。本文件是赛普拉斯半导体公司及其子公司，包括 Spansion LLC（“赛普拉斯”）的财产。本文件，包括其包含或引用的任何软件或固件（“软件”），根据全球范围内的知识产权法律以及美国与其他国家签署条约由赛普拉斯所有。除非在本款中另有明确规定，赛普拉斯保留在该等法律和条约下的所有权利，且未就其专利、版权、商标或其他知识产权授予任何许可。如果软件并不附随有一份许可协议且贵方未以其他方式与赛普拉斯签署关于使用软件的书面协议，赛普拉斯特此授予贵方属人性质的、非独家且不可转让的如下许可（无再许可权）（1）在赛普拉斯特软件著作权项下的下列许可权（一）对以源代码形式提供的软件，仅出于在赛普拉斯硬件产品上使用之目的且仅在贵方集团内部修改和复制软件，和（二）仅限于在有关赛普拉斯硬件产品上使用之目的将软件以二进制代码形式的向外部最终用户提供（无论直接提供或通过经销商和分销商间接提供），和（2）在被软件（由赛普拉斯公司提供，且未经修改）侵犯的赛普拉斯专利的权利主张项下，仅出于在赛普拉斯硬件产品上使用之目的制造、使用、提供和进口软件的许可。禁止对软件的任何其他使用、复制、修改、翻译或汇编。

在适用法律允许的限度内，赛普拉斯未对本文件或任何软件作出任何明示或暗示的担保，包括但不限于关于适销性和特定用途的默示保证。没有任何电子设备是绝对安全的。因此，尽管赛普拉斯在其硬件和软件产品中采取了必要的安全措施，但是赛普拉斯并不承担任何由于使用赛普拉斯产品而引起的安全问题及安全漏洞的责任，例如未经授权的访问或使用赛普拉斯产品。此外，本材料中所介绍的赛普拉斯产品有可能存在设计缺陷或设计错误，从而导致产品的性能与公布的规格不一致。（如果发现此类问题，赛普拉斯会提供勘误表）赛普拉斯保留更改本文件的权利，届时将不另行通知。在适用法律允许的限度内，赛普拉斯不对因应用或使用本文件所述任何产品或电路引起的任何后果负责。本文件，包括任何样本设计信息或程序代码信息，仅为供参考之目的提供。文件使用人应负责正确设计、计划和测试信息应用和由此生产的任何产品的功能和安全性。赛普拉斯产品不应被设计为、设定为或授权用作武器操作、武器系统、核设施、生命支持设备或系统、其他医疗设备或系统（包括急救设备和手术植入物）、污染控制或有害物质管理系统中的关键部件，或产品植入之设备或系统故障可能导致人身伤害、死亡或财产损失其他用途（“非预期用途”）。关键部件指，若该部件发生故障，经合理预期会导致设备或系统故障或会影响设备或系统安全性和有效性的部件。针对由赛普拉斯产品非预期用途产生或相关的任何主张、费用、损失和其他责任，赛普拉斯不承担全部或部分责任且贵方不应追究赛普拉斯之责任。贵方应赔偿赛普拉斯因赛普拉斯产品任何非预期用途产生或相关的所有索赔、费用、损失和其他责任，包括因人身伤害或死亡引起的主张，并使之免受损失。

赛普拉斯、赛普拉斯徽标、Spansion、Spansion 徽标，及上述项目的组合，WICED，及 PSoC、CapSense、EZ-USB、F-RAM 和 Traveo 应视为赛普拉斯在美国和其他国家的商标或注册商标。请访问 cypress.com 获取赛普拉斯商标的完整列表。其他名称和品牌可能由其各自所有者主张为该方财产。