设计 GPIF™ II 主设备接口

关于本文档

适用范围及目的

AN87216 介绍的是如何设计一个 EZ-USB® FX3™主设备接口，用来与外部同步从设备 FIFO 进行通信。该设计使用了 FX3 GPIF™ II Designer 工具，并通过图形状态机入口开发该接口。为测试该设计，请通过 GPIF II 接口以背对背的形式连接两个 FX3 开发套件，其中，一个套件作为主设备（本应用笔记中的主题），另一个作为测试从设备使用。本应用笔记中附带了用于主设备和从设备 FX3 套件的固件源代码和 GPIF II 状态机。

相关项目

有

相关器件系列

CYUSB3014

最新版本的 FX3 SDK

点击此处

相关应用笔记

AN65974

更多代碼示例？我們聽到了

有關 USB SuperSpeed 代碼示例的綜合列表，請訪問 **http://www.cypress.com/101781**

目录

关于本文档 1
目录 1
1 简介 3
2 更多信息 4
2.1 EZ-USB FX3 软件开发套件 4
2.2 GPIF™ II Designer 4
3 GPIF II 简介 6
4 同步从设备 FIFO 接口 8
4.1 同步从设备 FIFO 访问序列和接口时序 8
4.2 同步从设备 FIFO 读序列 10
4.3 同步从设备 FIFO 写序列 11
4.4 FX3 DMA 架构的基本情况 12
4.4.1 套接字（Sockets） 12
4.4.2 线程（Threads） 12
4.5 从设备 FX3 固件中的 DMA 通道配置 13

目录

4.6 标志配置 14
4.6.1 专用线程标志 14
5 GPIF II Designer 工具 15
5.1 接口定义 16
5.2 状态机实现 17
5.3 GPIF II 操作 18
5.3.1 LD_DATA_COUNT （加载数据计数器） 19
5.3.2 LD_ADDR_COUNT（加载地址计数器） 20
5.3.3 DR_GPIO（驱动 GPIO） 20
5.3.4 IN_DATA（输入数据） 21
5.3.5 DR_DATA（驱动数据） 22
5.3.6 IN_ADDR（输入地址） 23
5.3.7 COUNT_DATA、COUNT_ADDR 23
5.3.8 INTR_CPU 23
5.3.9 COMMIT 24
5.4 GPIF 事件 24
6 GPIF II 主设备状态机实现 25
6.1 FX3 GPIF II 主设备完成状态机 25
6.2 FX3 主设备写入到 FX3 从设备 26
6.3 FX3 主设备对 FX3 从设备进行读操作 27
7 FX3 主设备固件实现 29
8 硬件连接 30
9 运行演示版的步骤 32
10 相关的项目文件 36
11 总结 37
12 相关应用笔记 38
13 附录：使用 FX3 开发套件（CYUSB3KIT-001）时的硬件设置 39
文档修订记录 40

1 简介

Cypress FX3 是一个 USB 3.0 外设控制器，可提供高度集成特性和灵活特性以帮助开发人员将 USB 3.0 功能添加到任何系统中。

FX3 具有一个可配置的并行通用可编程接口（GPIF II），即赛普拉斯第二代通用可编程接口。GPIF II 可连接至一个外部处理器（如 ASIC 或 FPGA）。它可同多种接口（如异步 SRAM、异步和同步地址数据复用式接口）建立无缝连接。

同步从设备 FIFO 接口是 GPIF II 的常用领域。在使用该接口的应用中，连接至 FX3 的外部器件将访问 FX3 FIFO，
实现对该接口进行数据读写。并非通过从设备 FIFO 接口执行直接寄存器访问。

本应用笔记侧重于介绍同步 FIFO 主设备接口的设计。主设备启动传输，驱动地址总线（若存在），并通常为从设备提供时钟。本设计中的从设备是另一个 FX3 器件，其 GPIF II 单元被编程为一个从设备 FIFO。要想测试该设计，请通过两个 FX3 开发套件（DVK）上各自的 GPIF II 接口将这两个套件连接起来，此外，其中一个 FX3 DVK 被编程为 FIFO 从设备单元。描述 FIFO 主设备的实现前，本应用笔记将介绍有关该从设备 FIFO 接口的详细信息。更多有关从设备 FIFO 接口的详细信息，请查看 **AN65974，使用 EZ-USB® FX3™从设备 FIFO 接口进行设计**。

在该示例中，单个 PC 通过它的两个 USB 3.0 端口传输测试数据，使用第二个 PC 在两个 PC 间传输数据（如 **Figure 1** 所示）。

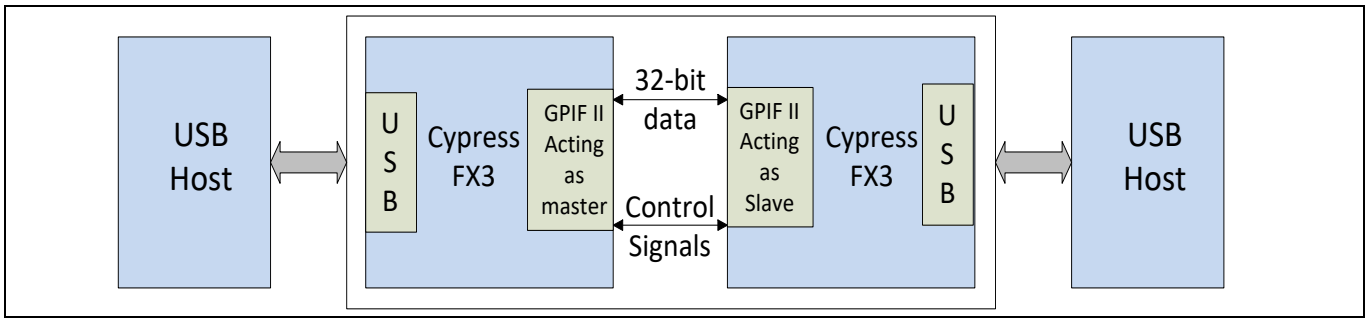

Figure 1 主机至主机的互联电缆

下面一节介绍了 GPIF II、从设备 FIFO 接口及其状态机实现（通过使用 GPIF II Designer 工具）的概述。

2 更多信息

赛普拉斯在 **www.cypress.com** 上提供了大量数据，帮助您为您的设计选择合适的设备，并帮助您快速有效地将设备集成到您的设计中。 有关资源的完整列表，请参阅知识库文章 **KBA87889, 如何使用 FX3/FX3S 进行设计**。

- 概述: **USB 产品组合**, **USB 路线图**
- USB 3.0 产品选择器: **FX3**, **FX3S**, **CX3**, **HX3**, Benicia 应用说明：赛普拉斯提供大量 USB 应用笔记，涵盖从基础到高级的广泛主题。有关 FX3 入门的推荐应用说明如下:
  - **AN75705** – EZ-USB FX3 入门
  - **AN76405** – EZ-USB FX3 引导选项
  - **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** – 如何在 USB 视频类（UVC）框架中使用 EZ-USB FX3 实现图像传感器接口
  - **AN86947** – 使用 EZ-USB FX3 优化 USB 3.0 吞吐量
  - **AN84868** – 使用赛普拉斯 EZ-USB FX3 通过 USB 配置 FPGA
  - **AN68829** – EZ-USB FX3 的从设备 FIFO 接口: 5-Bit 地址模式
  - **AN73609** – EZ-USB FX2LP / FX3 在 Linux 上开发批量循环示例
  - **AN77960** – EZ-USB FX3 高速 USB 主机控制器简介
  - **AN76348** – EZ-USB FX2LP 和 EZ-USB FX3 应用的实现差异
  - **AN89661** – USB RAID 1 Disk Design Using EZ-USB FX3S
- 代码示例 :
  - **USB Full-Speed** 全速 (USB 1.1 & USB 2.0)
  - **USB Hi-Speed** 高速 (USB 2.0)
  - **USB SuperSpeed** 超高速 (USB 3.0)
- 技术参考手册（TRM）:
  - **EZ-USB FX3 Technical Reference Manual**
- 开发套件:
  - **CYUSB3KIT-003, EZ-USB FX3 SuperSpeed Explorer 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 提供完整的软件和固件堆栈，可轻松将 SuperSpeed USB 集成到任何嵌入式应用中。**软件开发工具包**（SDK）附带了工具，驱动程序和应用程序示例，有助于加速应用程序开发。

2.2 GPIF™ II Designer

GPIF II Designer 是一款图形软件，允许设计人员配置 EZ-USB FX3 USB 3.0 设备控制器的 GPIF II 接口。

该工具允许用户从赛普拉斯提供的五个接口中选择一个，或者从头开始选择创建自己的 GPIF II 接口。赛普拉斯提供了行业标准接口，如异步和同步从设备 FIFO，以及异步和同步 SRAM。已经在系统中拥有这些预定义接口之一的设计人员可以简单地选择所选择的接口，从一组标准参数中选择，如总线宽度（x8,16，x32）字节序，时钟设置，然后编译接口。该工具为需要定制界面的用户提供了简化的三步 GPIF 界面开发流程。用户可以先选择其引脚配置和标准参数。其次，他们可以使用可配置的动作设计虚拟状态机。最

更多信息

后，用户可以查看输出时序以验证它是否与预期时序匹配。完成此三步过程后，可以编译界面并与 FX3 集成。

3 GPIF II 简介

GPIF II 是一个可编程状态机，使用它可以灵活地实现一个行业标准或专用接口。GPIF II 可作为外部器件的主设备或从设备使用。

GPIF II 端口（亦即 FX3 的 P 端口）是一个具备多达 32 条双向数据线的并行接口。可将数据线分成地址线，或按时间将其复用到地址线。最多可将 13 条控制线配置为 IN 或 OUT。可编程的状态机引擎控制着 GPIF II 端口操作和控制信号。外部处理器通过控制信号（被配置为 FX3 输入）来控制状态机的操作。

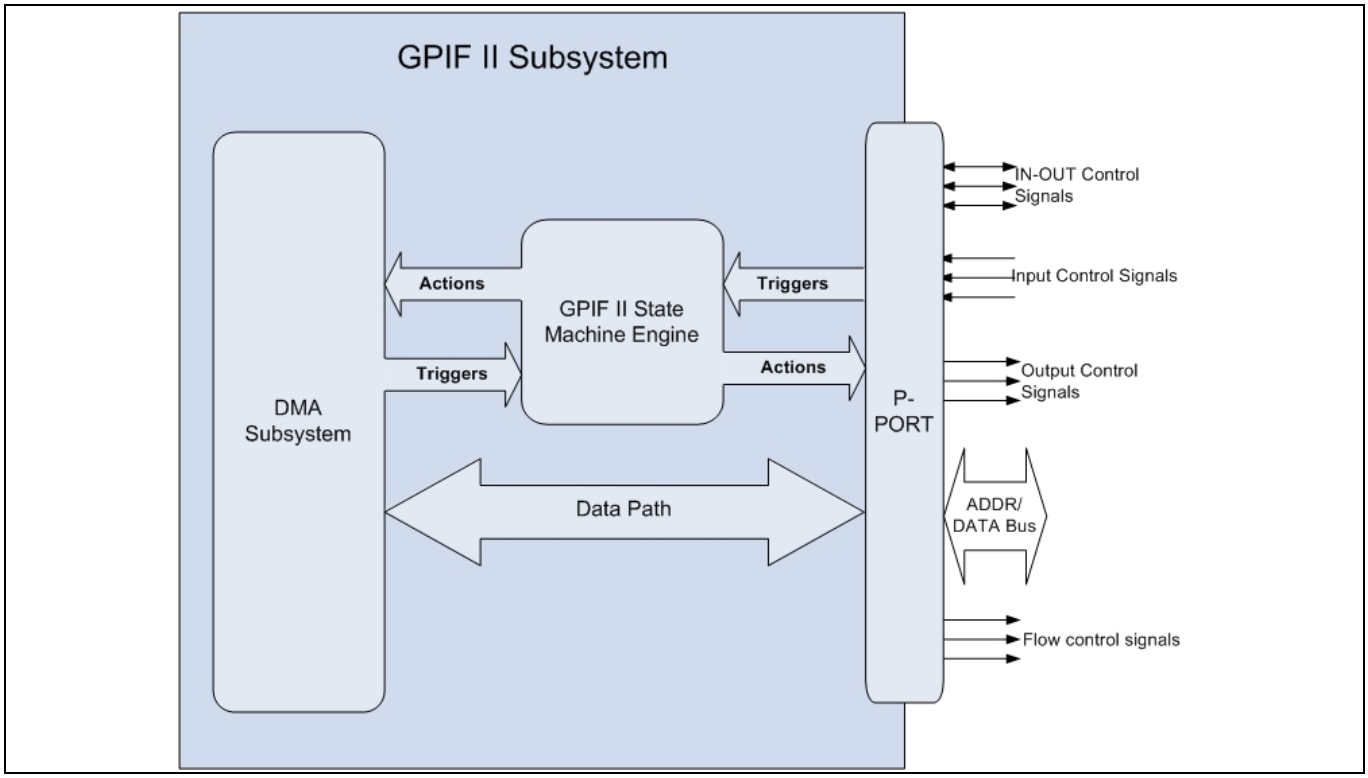

Figure 2 FX3 的 GPIF II 子系统

GPIF 硬件还支持生成一组 DMA 状态标志。这组标志用来指示缓冲区的就绪状态，或基于一个用户编程的阈值。外部处理器可以使用该组标志进行流控制。使用这些标志可确保传输中数据不被丢失。

一组计数器和比较器模块与该 GPIF 硬件相关联，并由状态机使用。比较器检查地址、数据或控制信号的当前状态是否与特定模式相匹配，并生成一个匹配信号。状态机可将该匹配信号作为触发器使用。通过状态机操作，可复位或更新计数器。该计数器也会生成用作触发器的有限匹配信号。

GPIF II Designer 工具生成与设计（用户输入）相应的配置数据。在 FX3 固件上调用 API 将该数据载入，用以初始化 GPIF II 硬件。FX3 SDK 包含一组 API，可用来配置、控制和监控 GPIF 接口。

通过在状态机画布上选中用户指定的操作，GPIF II 硬件可以触发 FX3 板上 ARM 内核的中断，也可以触发外部处理器的中断。GPIF II 状态机还能使用固件生成的输入信号来控制 GPIF 接口的操作。

GPIF II 具有以下特性：

- 可用作主设备或从设备
- 提供 256 种固件可编程状态
- 支持 8 位、16 位和 32 位并行数据总线
- 接口频率可高达 100 MHz

GPIF II 简介

- 使用 32 位数据总线时，该接口支持 14 个可配置的控制引脚；所有的控制引脚都可作为输入/输出或双向引脚使用
- 使用 16 位或 8 位数据总线时，该接口支持 16 个可配置的控制引脚；所有的控制引脚可作为输入/输出或双向引脚使用根据输入信号进行转换 GPIF II 状态，并且该 GPIF II 状态将驱动控制输出信号。状态机的行为由一个描述符定义，为满足所需接口规范而设计。本质上，GPIF II 描述符是可编程寄存器的集合。在 FX3 寄存器空间中，8 kB 的内存空间专门作为 GPIF II 波形存储器，用于存储 GPIF II 描述符。

GPIF II 状态转换基于输入信号发生，控制输出信号由 GPIF II 状态驱动。 状态机的行为由描述符定义，该描述符旨在满足所需的接口规范。 GPIF II 描述符本质上是一组可编程寄存器值。 在 FX3 寄存器空间中，8 千字节专用于 GPIF II 波形存储器，其中存储了 GPIF II 描述符。 有关 GPIF-II Designer 工具的更多信息，请参 **GPIF-II Designer User Guids**。

GPIF II 常用于实现同步从设备 FIFO 接口。下面章节将对其进行详细介绍。随后一节将描述如何使用 FX3 GPIF II Designer 工具设计与同步从设备 FIFO 接口相兼容的主设备接口。

4 同步从设备 FIFO 接口

要想充分利用本应用笔记中下一节的内容，您应该了解同步 FIFO 接口的基本知识。此处介绍了同步从设备 FIFO 接口的详细信息（用以了解 GPIF II 主设备状态机）。

本节介绍了同步从设备 FIFO 接口的互联图，其次是信号的引脚映射情况。有关在从设备 FIFO 接口上执行读/写操作的时序图，请查看**同步从设备 FIFO 访问序列和接口时序**一节。有关从设备 FX3 固件中 P 端口套接字、标志和 DMA 通道配置的详细信息，请参阅**从设备 FX3 固件中的 DMA 通道配置**一节。

在需要进行下述操作的应用中，使用同步从设备 FIFO 接口会很合适：外部处理器或器件需要按顺序对 FX3 的内部 FIFO 缓冲区进行数据读/写访问。并非通过从设备 FIFO 接口执行寄存器访问。同步从设备 FIFO 接口通常是 USB 应用的首选接口，以便支持高吞吐量的要求。

Figure 3 显示了同步从设备 FIFO 接口的接口框图。

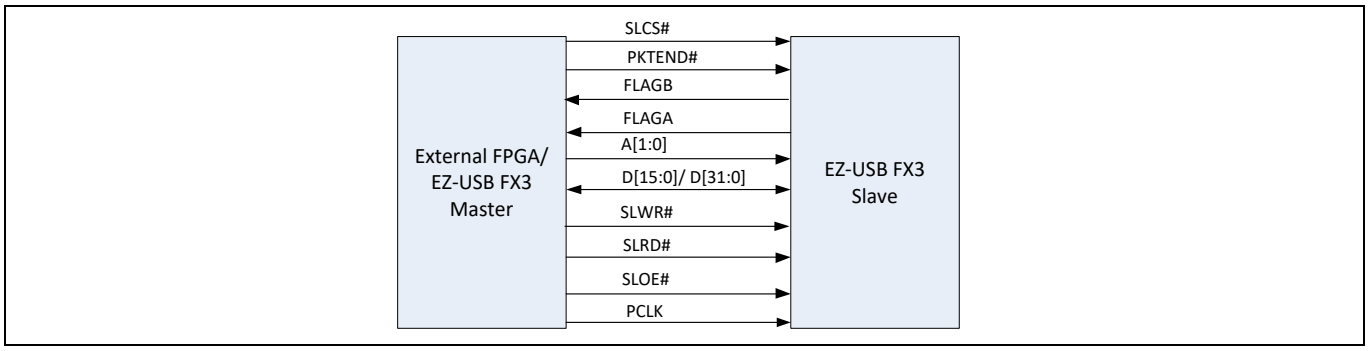

Figure 3 同步从设备 FIFO 接口

Table 1 同步从设备 FIFO 接口信号

信号名称	信号说明
SLCS#	从设备芯片选择，低电平有效。主设备激活该信号以访问从设备接口。
SLWR#	从设备写入选通，低电平有效。主设备激活该信号，以执行写入操作（主设备至从设备）。
SLRD#	从设备读取选通，低电平有效。主设备激活该信号，以执行读取操作（从设备至主设备）。
SLOE#	从设备输出使能，低电平有效。主设备激活该信号，以使从设备驱动其数据总线。如果该信号未被激活，从设备数据总线将悬空。
FLAGA/ FLAGB	FX3 标志输出。可在 FX3 中编程该信号，以指示从设备的不同状态。在该应用中，配置 FLAGA 以指示从设备端数据的可用性，配置 FLAGB 以指示从设备端空闲缓冲区的可用性。
A[1:0]	地址总线。在该 FX3 GPIF II 示例中，要通过使用两个地址行来选择从设备 FX3 上的线程。
D[31:0]	这是从设备 FIFO 接口的 16 位或 32 位数据总线。
PKTEND#	数据包结束，负真值。主设备激活该信号，用以指示 FX3 从设备通过 USB 发送长度为零的数据包或短数据包。
PCLK	主设备提供该时钟给从设备。所有数据和时序操作都参考了该时钟。

4.1 同步从设备 FIFO 访问序列和接口时序

本节介绍了同步从设备 FIFO 接口的访问序列和时序。

同步从设备 FIFO 接口

在本应用笔记附带的设计中，主设备 FX3 对从设备 FX3 的内部 FIFO 缓冲区进行连续数据访问。主设备 FX3 在 ADDR 线上驱动 2 位地址，并激活读选通或写选通信号。从设备 FX3 进行 USB 传输，并激活 FIFO FLAG 信号，以指示空条件（读取 FIFO 时）或满条件（写入 FIFO 时）。

Figure 4 显示的是从外部 FPGA/处理器的角度来看，从设备 FIFO 接口的逻辑框图。**Table 2** 显示的是 FX3 GPIF-II 主设备时序参数。

Table 2　　FX3 GPIF-II 主设备时序参数

参数	说明	最小值	最大值	单位
频率	时钟频率	100	–	MHz
tDS_addr	输入地址建立时间	3	–	ns
tDH_addr	输入数据保持时间	1.5	–	ns
tDO_addr	时钟到输出地址的延迟时间	–	8	ns
tDOH_addr	输出地址保持时间	2	–	ns
tS	输入控制建立时间	3	–	ns
tH	输入控制保持时间	1.5	–	ns
tDS	输入数据设置时间	3	–	ns
tDH	输入数据保持时间	1.5	–	ns
tDO	时钟到输出数据延迟时间	–	8	ns
tDOH	输出数据保持时间	2	–	ns
tCTLO	时钟到控制输出数据的延迟时间	–	8	ns
tCOH	控制输出数据的保持时间	2	–	ns
tHZ	时钟至高阻态的时间	–	8	ns

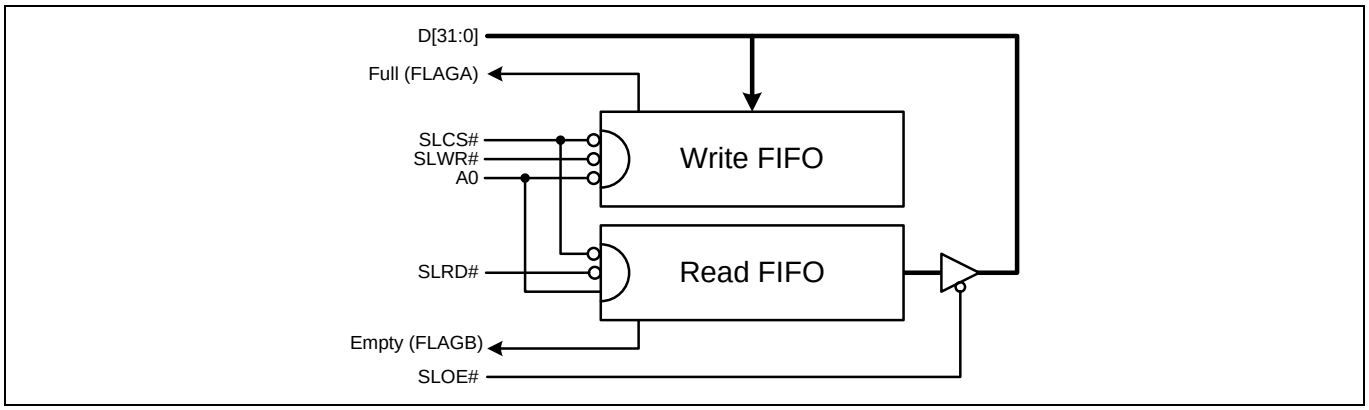

Figure 4　　从设备 FIFO 接口的逻辑框图

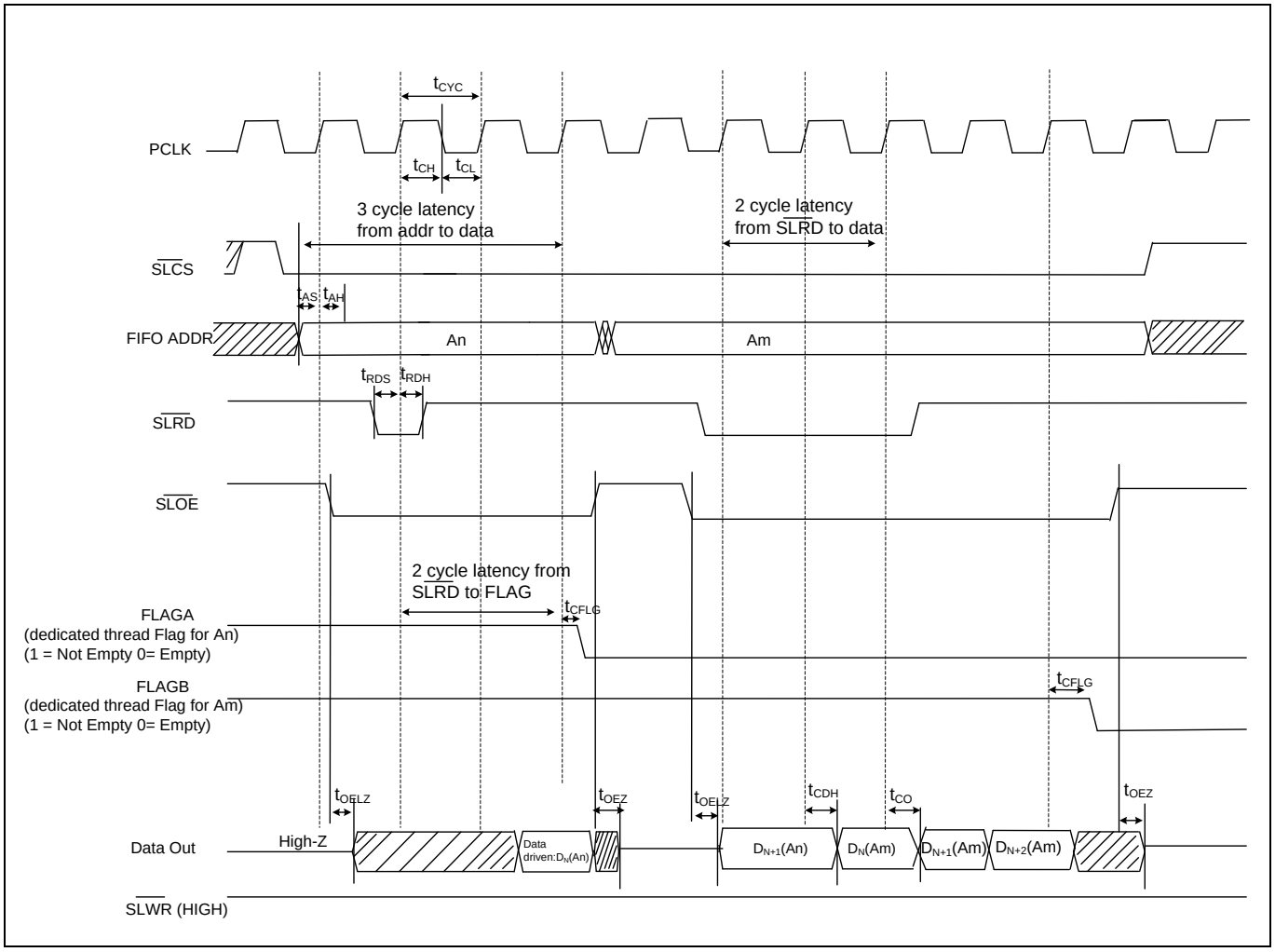

Figure 5　同步从设备 FIFO 读序列

4.2　同步从设备 FIFO 读序列

Figure 5 显示的是主设备如何从从设备（即为预编程的 FX3 DVK）读取单个 32 位字。该图显示的是先对 FIFO 地址“Am”进行连续读取，然后再对 FIFO 地址“An”进行单字读取操作。

请按照下面各步骤执行单 32 位字的读取操作：

1. 主设备先驱动 FIFO 地址“An”，然后激活 SLCS#信号。请注意，主设备必须确保该地址满足建立时间 tAS 的要求（针对采样 SLCS#信号的 PCLK 上升沿）。
2. 主设备激活 SLOE#信号，从而使从设备开始驱动数据总线。
3. 主设备激活 SLRD#，并在一个 PCLK 时长内使其保持低电平状态。这样会启动从寻址的 FIFO 至数据总线的数据传输。

当 SLRD#有效时，FIFO 指针在 PCLK 的上升沿上得到更新。这样会启动从新寻址的 FIFO 至数据总线的数据传输。经过从 PCLK 上升沿算起的两个时钟后，新数据值将在经过时钟边沿后的 tCO 时间内提供。N 是从 FIFO 读取的第一个数据值。要想驱动数据总线，还必须激活 SLOE#。

Figure 5 的下半部分显示了连续读取传输。在连续模式下，在整个读取过程中要持续激活 SLRD#和 SLOE#。首次激活 SLOE#时，使用先前已寻址的 FIFO 中的数据驱动数据总线。当 SLRD#有效时，对于每一个 PCLK 序列的上升沿，FIFO 指针会递增，且下一个数据值会被传输到数据总线上。在 **Figure 5** 示例中，检索的 FIFO 字是 FIFO 中的最后一个字。它是由“空白”FLAGB 信号指示的。

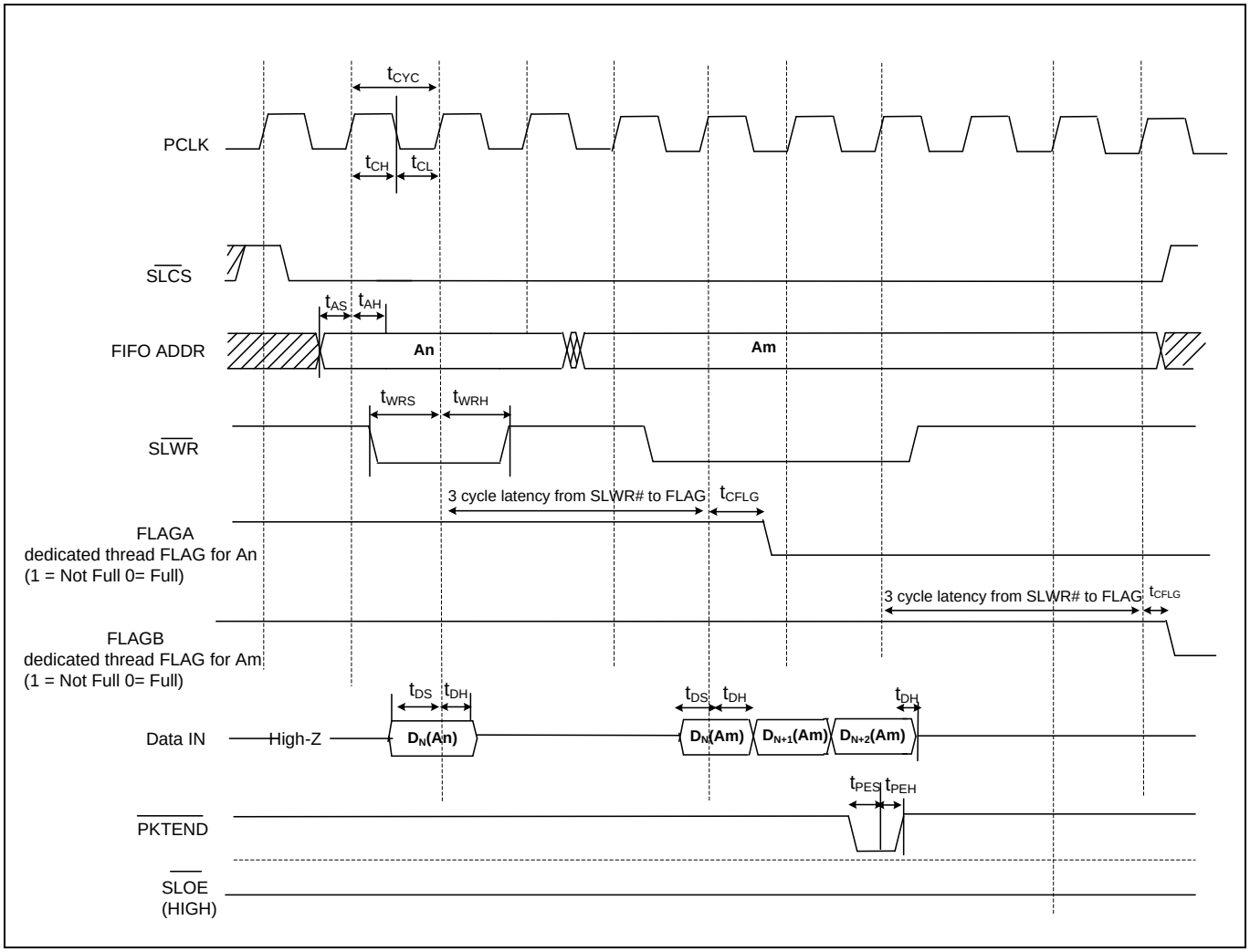

Figure 6　同步从设备 FIFO 写序列

4.3　同步从设备 FIFO 写序列

Figure 6 显示的是对 FIFO 地址“An”进行单个字写入操作然后进行连续写入操作。

请按照下面各步骤执行单 32 位字写入操作：

1. 主设备先驱动 FIFO 地址“An”，然后激活 SLCS#信号。请注意，主设备必须确保该地址满足建立时间 tAS 的要求（针对采样 SLCS#信号的 PCLK 上升沿）。
2. 主设备将它的数据驱动到数据总线上。
3. 激活 SLCS#后，主设备将在下一个时钟周期内激活 SLWR#。
4. 激活 SLWR#时，主设备会将数据写入到 FIFO 内，并且 FIFO 指针会在 PCLK 的上升沿上递增。
5. 从时钟上升沿算起，经过三个时钟周期以及 tCFLG 延迟时间后，FIFO 标志将被更新。

连续写入时显示相同的事件序列。

在连续模式下，主设备将在整个连续写入过程中持续激活 SLWR#和 SLCS#。在连续写入模式下，主设备激活了 SLWR#后，每当 PCLK 的上升沿到来时，数据总线上的数值都被写入到 FIFO 内。激活 SLWR#后，将在 PCLK 的每个上升沿上更新 FIFO 指针。

短数据包：通过使用 PKTEND#信号，可将某个短数据包发送给 USB 主机。主设备 FX3 应该同时激活 PKTEND#，最后的数据字以及与该字相应的 SLWR#脉冲。如果激活 PKTEND#但不激活 SLWR#脉冲，就生

成 ZLP（长度为零的数据包）。主设备必须在激活 PKTEND#过程中保持 FIFOADDR 不变。激活 PKTEND#和 SLWR#，同时从设备 FX3 的 GPIF II 状态机会将数据包作为一个短数据包，并将其发送到 USB 接口。如果协议不要求传输任何短数据包，则 PKTEND#信号可能会被置为高电平。

请注意，在执行读操作时，没有任何特定的信号用于确定是否已从 USB 获取短数据包。必须通过主设备 FX3 监控空 FLAG，以确定完成读取所有数据的时间。

4.4 FX3 DMA 架构的基本情况

FX3 器件有一个内部 DMA 结构，用来将 GPIF II 接口连接至内部系统存储器和其他串行外设。通过 GPIF II 接口传输的所有数据都是传入/传出内部存储器缓冲区的。通过在 FX3 上运行的固件应用将该数据路径（使用 DMA 结构）连接至合适的发送器或接收器（如 USB 主机或串行外设）。

4.4.1 套接字（Sockets）

USB 3.0 器件上的每一个端口都支持与一个数据流终端相应的套接字，并可对其独立寻址。FX3 P 端口（处理器端口）或 GPIF II 端口支持多达 32 个套接字，因此可通过该接口配置总共 32 个独立的数据路径。

与所有被使用的套接字相应的存储器缓冲区是通过固件应用分配的，并将套接字连接至合适的数据发送器或接收器。
请参阅 **FX3 SDK API 指南**中 DMA 通道 API 的内容以了解执行该配置的函数。

4.4.2 线程（Threads）

虽然通过 FX3 器件上的 GPIF 端口可使用 32 个可寻址套接字将数据发送端和接收端连接起来，但只能同时使用四个数据高速传输路径（线程）进行传输数据。这样，应用最多能选用被绑定到相应线程的四个套接字，并且在这些套接字间进行切换不会发生额外的延迟。

通过在每一个终端分配套接字（一个套接字提供数据，另一个接收数据），典型的 FX3 应用初始化了一个或多个线程。通过提供输入地址，或通过指定 GPIF 状态机中 IN_DATA 或 DR_DATA 操作设置（在 **GPIF II 操作** 一节中所述）的目标线程，可直接指定每个传输数据字所使用的线程。

Note: *虽然这个应用不需要使用该技术，但（在应用正在运行时）您可以动态更改套接字至线程的分配。更改绑定某个线程的有效套接字需要额外的延时。通过指定套接字地址（在地址总线宽度范围为 3 到 5 位的情况下），或者通过 FX3 器件中的固件干预，可以自动进行切换。*

GPIF 硬件为每个线程提供了一个组 DMA 状态标志。线程的特定 DMA 标志始终反映了该线程上有效套接字的状态。在某个线程上切换有效套接字时，要有足够长的时间，以确保在使用标志进行控制数据传输前，该标志反映了新套接字的状态。

请注意，套接字至线程的映射可能不完全灵活。可将每个套接字 N 仅绑定给已编号的线程（N MOD 4）。例如，套接字 7 只能使用线程 3（7 除以 4 后的余数）。套接字 11 也可以使用线程 3。这样，不能通过将套接字 0、4、8 等绑定到不同的线程来同时使用这些套接字。但这并不是限制。这是因为套接字的数量远远超过线程的数量（32 与 4），所以仍有很多可用的选项。当选择用于 GPIF 数据传输的套接字时，请留意。

4.5 从设备 FX3 固件中的 DMA 通道配置

固件必须通过使用所需的发送套接字和接收套接字来配置 DMA 通道。

请注意，如果将数据从从设备 FIFO 接口传输到 USB 接口，那么 P 端口或 GPIF II 端口将是发送套接字，USB 是接收套接字，反之亦然。

因此，如果通过从设备 FIFO 接口进行双向数据传输，请配置两个 DMA 通道。使用固件中的 CyU3PDmaChannelCreate API（更多有关 API 的信息，请参阅 **FX3 API 指南**），一个端口用作为发送套接字的 P 端口，另一个用作为接收套接字的 P 端口。

P 端口发送套接字是外部器件通过从设备 FIFO 接口对其进行写入的套接字。P 端口使用套接字是外部器件通过从设备 FIFO 接口对其进行读取的套接字。

在本应用笔记附上的示例中，套接字 0 被配置为 P 端口的生成套接字，套接字 1 被配置为使用套接字（请查看 **Figure 7**）。

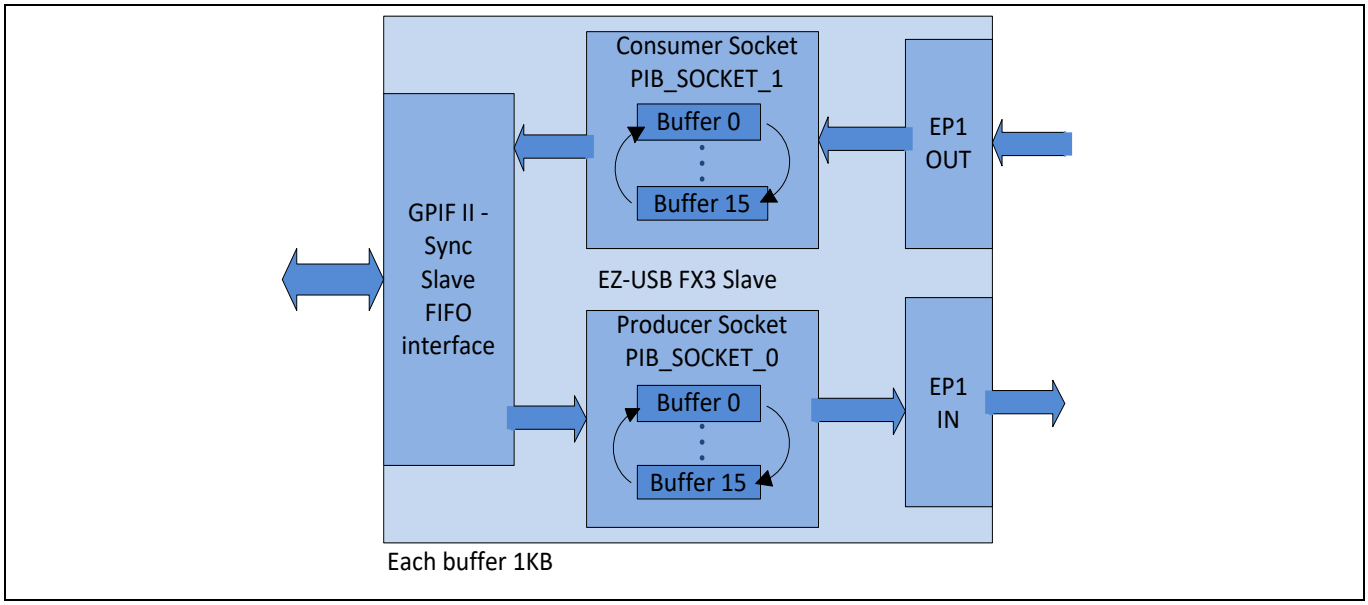

Figure 7 同步从设备 FIFO 固件中所使用的 PIB 套接字

请注意，DMA 通道中的 P 端口套接字编号要与 A1:A0 上所寻址的套接字编号相同。

配置 DMA 通道时，可将多个缓冲区分配到同一个特定的 DMA 通道上。请注意，标志分别表示的是每个缓冲区的满/空状态。（缓冲区的最大容量为 64 KB-16）

例如，如果已将两个大小都是 1024 字节的缓冲区分配给 DMA 通道，当将 1024 字节写入到第一个缓冲区后，满载 FLAG 将显示“满”状态。它持续显示满状态，直至 DMA 通道转到第二个缓冲区为止。DMA 通道转到下一个缓冲区所需的时间不长，但通常需要几微秒。外部主设备必须监控 FLAG，以确定何时完成切换操作，以及何时可用下一个缓冲区来访问数据。

下一节将会说明如何配置 FLAG，以表示不同线程的状态。

同步从设备 FIFO 接口

4.6 标志配置

标志可被配置为空、满、局部空或局部满等信号。这些信号并非由 GPIF II 状态机控制，而是由 EZ-USB FX3 内部的 DMA 硬件引擎控制，该 DMA 用于使 FX3 RAM 中的数据传输到 FIFO。标志可与特定线程相关联，也可以与当前寻址的线程动态关联。在这两种情况下，标志将指示映射到该线程的套接字的状态。

标志会根据（初始化套接字过程中所配置的）套接字方向来指示空或满状态。因此如果从套接字读取数据，标志将表示空状态；如果将数据写入套接字内，标志将表示满状态。

不同类型的 FLAG 可以使用：

- 专用线程标志（空/满或局部空/局部满）
- 当前线程标志（空/满或局部空/局部满）

4.6.1 专用线程标志

可对标志进行配置，以指示特定线程 0-3 的状态。在这种情况下，在地址总线上无论寻址哪个线程，该标志都会专门用于该线程，且始终只表示映射到该特定线程的套接字的状态。

在这种情况下，外部处理器/器件必须跟踪各线程专用的具体标志，并在每次寻址到不同线程时必须监控正确的标志。

例如，如果 FLAGA 专用于线程 1，而 FLAGB 专用于线程 0，则当外部处理器对线程 1 执行访问时，必须监控 FLAGA。当外部处理器访问线程 0 时，必须监控 FLAGB。

每个将要访问的线程均有一个与其相对应的专用标志。如果应用要求访问四个线程，将会使用四个相应的标志。

在该示例应用中，FLAGA 专门用于线程 1、FLAGB 专门用于线程 0。因此，如果主设备 FX3 需要在从设备 FIFO 接口上执行写操作，它要等待 FLAGB 的状态变为高电平后才能开始。FLAGB 的高电平状态表示分配给该数据路径的 DMA 缓冲区尚未满载。当 DMA 缓冲区满载时，FLAGB 将转为低电平。同样，当在从设备 FX3 器件上执行读操作时，主设备 FX3 需要等待 FLAGA 变为高电平后才能开始。FLAGA 的高电平状态表示分配给该数据路径的 DMA 缓冲区尚未空。当 DMA 缓冲区中没有任何数据时，FLAGA 保持低电平状态。

更多有关从设备 FIFO 接口实现的信息，请参阅 **AN65974，使用 EZ-USB FX3 从设备 FIFO 接口进行设计**。

5 GPIF II Designer 工具

因为已明确了从设备 FIFO 接口的要求，现在请学习如何使用 GPIF II Designer 执行 GPIF II 设计。GPIF II Designer 是一个图形工具，它通过使用状态机画布和菜单来配置 FX3 GPIF II 接口。

通过使用 GPIF II Designer，您可以从赛普拉斯先前构建的接口库中选择与所需合适的接口，或者从新开始创建一个 GPIF II 接口。 赛普拉斯提供了符合行业标准的接口，如异步和同步从设备 FIFO、异步和同步 ADMUX（地址/数据复用 RAM）以及外部 SRAM。在具有上述所定义的接口的系统中，开发人员可从一组标准的参数（如总线宽度（x8、16、 x32）、字节顺序和时钟设置）中选择所需接口。然后，编译选定的接口。

要想使用该工具设计一个自定义接口，首先必须在“Interface Definitions”选项卡中选择引脚配置和标准参数。然后，选择“State Machine”选项卡，以通过可配置操作设计状态机。完成该操作后，GPIF II Designer 将该接口编译到一个 C 头文件内，这样 FX3 项目可以包含该安装文件。

GPIF II 是 **FX3 SDK 安装**的一部分。它也是一个独立的安装文件。您可从**赛普拉斯网页**中显示的下述位置下载该安装文件。

Figure 8 显示的是新项目的一部分 GPIF II Designer 的截图。

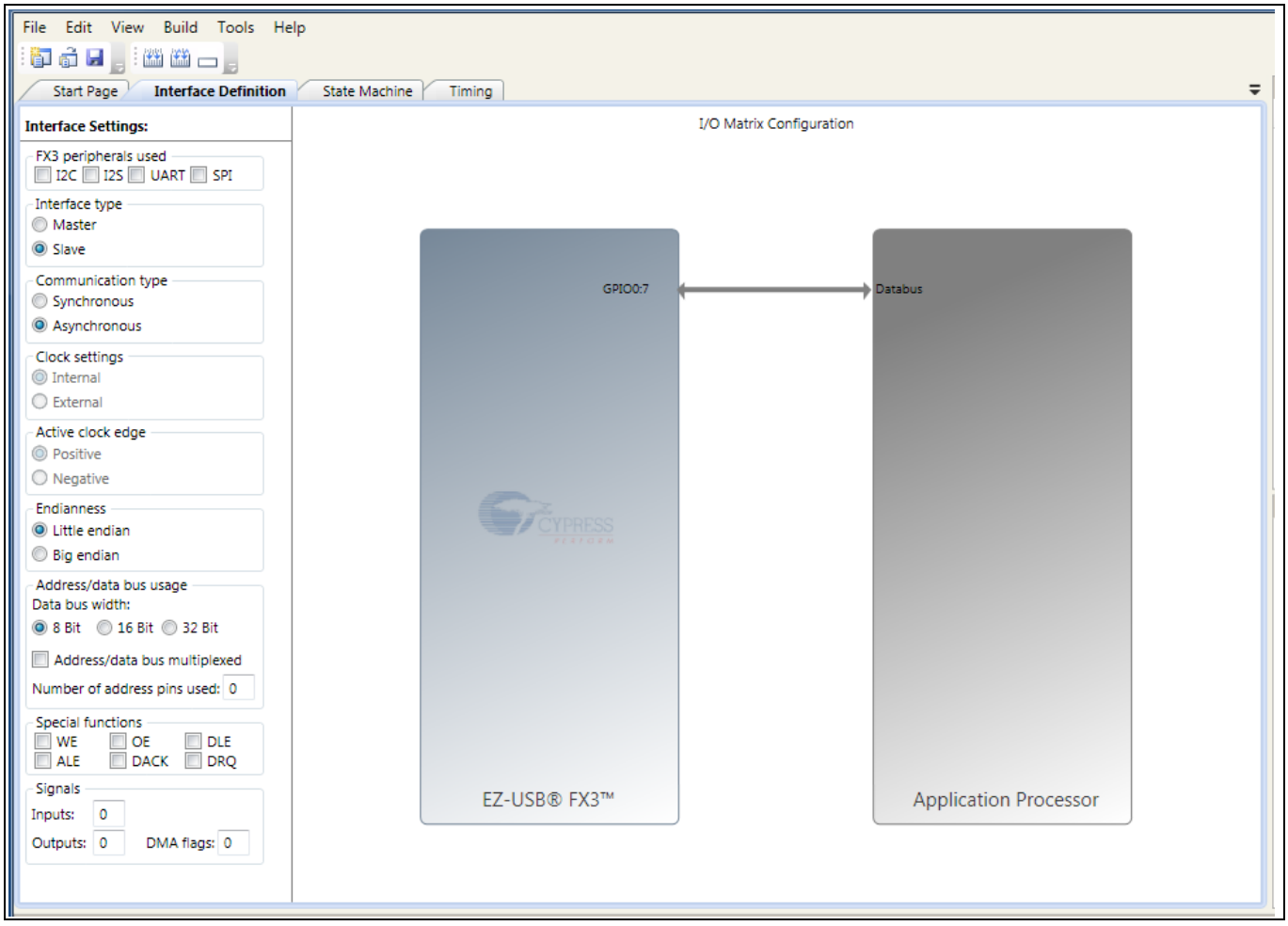

Figure 8 GPIF II Designer 工具

5.1 接口定义

第一步是通过钩选“Interface Settings”选项卡中的条目来配置能与外面连接的 GPIF II 接口（**Figure 8**）。根据您选择的和取消选择的各项，中心面板将发生变化，以反映接口的“活的原理图”。这样，您可以轻松查找 FX3 引脚映射情况，因为 FX3 模块中的 FX3 信号带有标签。

1. 您的整体应用将使用 FX3 的哪个串行外设？通过选中某一框，GPIF II Designer 可避免使用分配给其他 FX3 外设的引脚。如果选中的是“I2C”，则可以避免使用 SCL 和 SDA 引脚。
2. FX3 作为接口的主设备还是从设备使用？如果实现其中一种选择，主设备启动传输并驱动地址总线。当您选择了主设备或从设备时，Action List（操作列表）（右面板）将自动更新，以显示可用的选项。
3. 该接口是否使用时钟？如果使用，请选择“Synchronous”（同步）。异步 SRAM 是一个异步接口示例。它包含了一个地址和数据总线、读选通和写选通，但没有时钟。例如，通过激活地址，然后激活芯片使能并读选通，可以进行读操作。读选通后，将在传输时间内传出数据，且不使用任何时钟。
4. 您是否要 FX3 GPIF II 驱动接口时钟？如果需要驱动，请选择“内部”；否则，选择“外部”。如果第三个问题中没有选中“Synchronous”，该选项将被调暗。
5. GPIF II 何时采样接口信号？在上升沿还是下降沿上？如果第三个问题中没有选中“Synchronous”，该选项将被调暗。
6. 您器件的字节顺序？低位优先或高位优先。“字节顺序”指的是多个字节顺序 – 最高有效（大）或最低有效（小）字节优先。
7. 外部器件（连接至 FX3）的数据总线宽度为：8 位、16 位或 32 位。
8. 外部器件是否要求地址行？如果需要选择各 FX3 资源（如套接字），将使用到地址行，所以选是。是否复用地址和数据总线？
9. “Special features”（特殊特性）指的是唯一的硬件引脚和配置：
10. **WE**：特殊功能只能连接到 GPIO_18。该连接确保在激活 GPIO_18 时，入口路径的数据总线方向（进入 FX3 器件）将被设置。

    OE：该特性仅用于 GPIO_19。使用该功能时，通过激活 GPIO_19，可以直接更改出口路径上数据总线的方向（退出 FX3 器件）。

    DLE：使用数据锁存使能(DLE)功能时，通过取消激活 GPIO_18，可在额外的几纳秒时间内锁存数据行。如果在取消激活 GPIO_18 的边沿到来时，FX3 器件读取了使用于入口路径的数据行，将使用该特性。

11. FX3 需要监控多少个信号，且要控制多少个信号？此外，在您的应用中需要多少个 DMA 标志？当您添加输入或输出时，状态机设计工具也会自动添加它们，以在任何状态中进行测试（输入）或激活（输出）。

5.2 状态机实现

接口定义完成后，下一步是实现状态机。请查看下图的内容。

所有状态机都应该有一个 START（启动）状态。通过右击并选择“Add START State”项，可以添加 START 状态。在 START 状态中，不能发生任何转换。对于来自 START 状态的所有转换，转换公式被固定为‘LOGIC_ONE’。

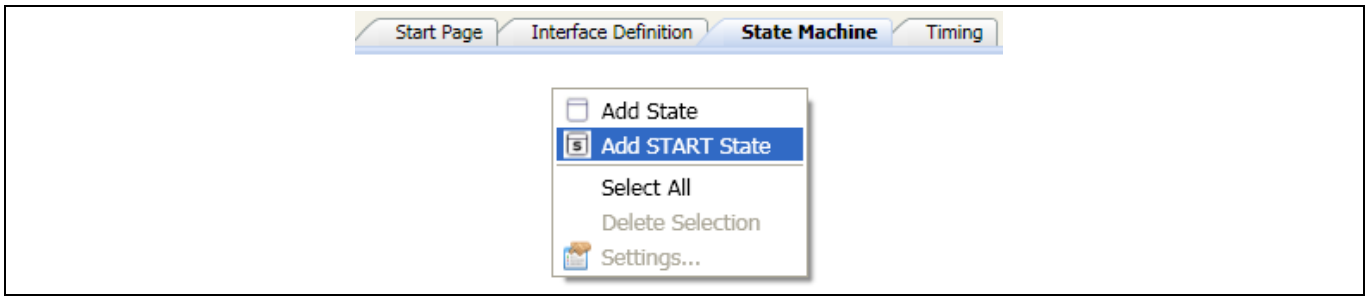

Figure 9 添加 START 状态

通过右击并选择“Add State”项，可以添加一个状态。**Figure 10** 显示了该步骤。

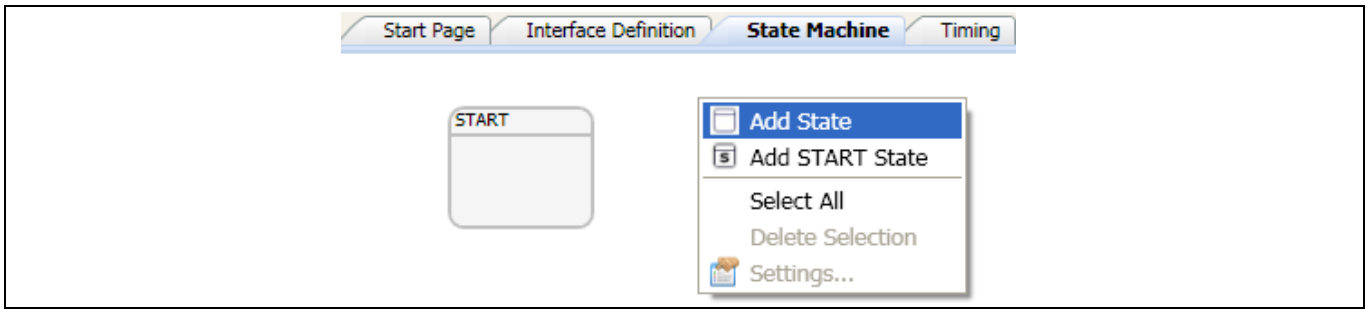

Figure 10 添加一个状态

您可以连接两个状态，如 **Figure 11** 所示。在 **Figure 11** 中，START 状态被转换成 STATE1 状态。

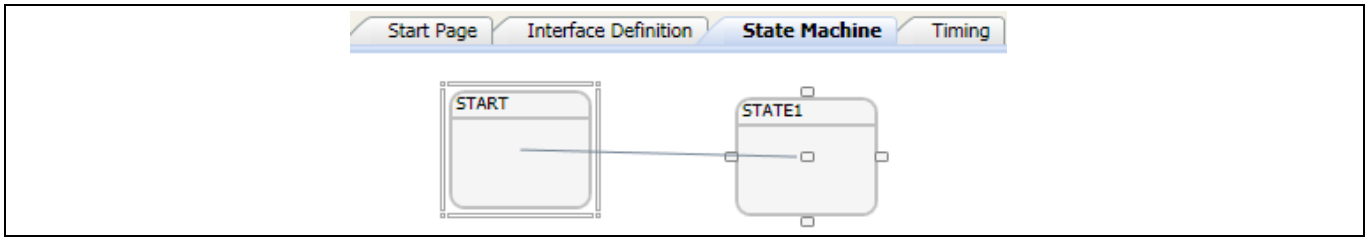

Figure 11 连接不同状态

根据应用的要求，需要给状态添加操作。下一节介绍了执行主从设备 FIFO 接口时需要的操作列表。

Figure 12 显示的是如何将一个操作添加到某个状态内。选择状态框后，右击“Action List”面板中的某个选项，然后选择“Add to Selected State”项。或者，您可将所需操作拖放到选定的状态框内。在每个“操作”条目上悬停鼠标，会显示一个提示，该提示说明了相应操作的用途。要想从状态移除某个操作，右击状态框中的名称文本，然后选择“Delete Action”（删除操作）项。

注意：可给 STATE 框分配多个操作。如果您尝试给同一个状态分配了互相冲突的操作，则 GPIF II Designer 会发出错误信息。例如，您在同一个状态中尝试加载并执行累加数据计数器。

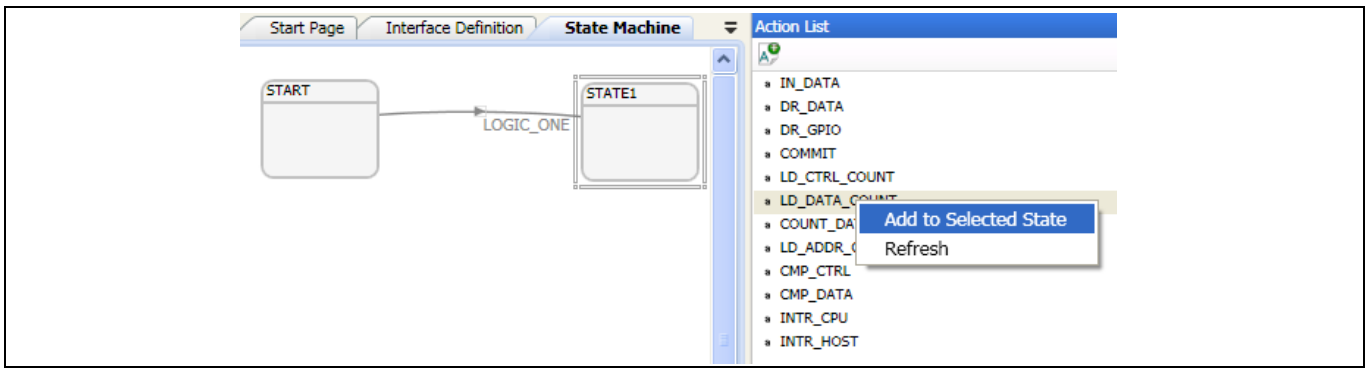

Figure 12　将某个操作添加到状态内

双击转换线后，可获取该状态可用的事件列表。添加的转换公式，如 **Figure 13** 所示。

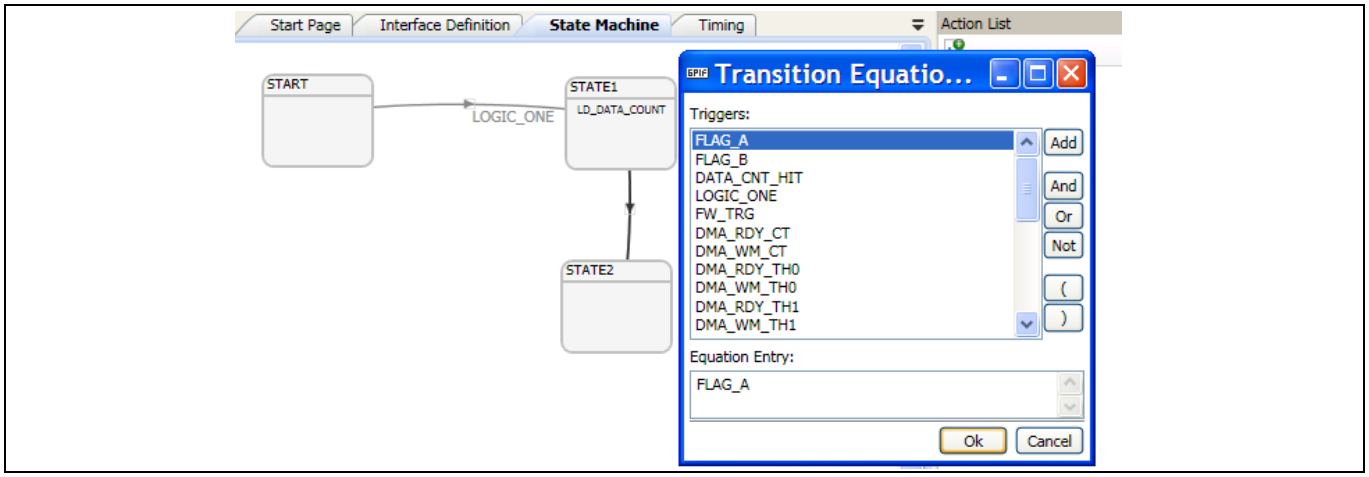

Figure 13　添加转换公式

5.3　GPIF II 操作

可对 GPIF II 状态机的每个状态进行编程，以执行一个或多个 GPIF II 操作。还可以编程某个状态中所执行的操作，只执行一次该操作，或每当时钟上升沿到来时都会执行它。直至转换为另一个状态前，始终执行该操作。此处介绍的是主设备状态机实现中的 GPIF II 操作。有关所有 GPIF II 操作的说明，请打开 GPIF II Designer 中的 *Help*，并点击 *Topics* 或按下 F1 键。在刚打开的 *GPIF II Designer* 窗口中选择 *Contents* 选项卡，然后浏览 *编程 GPIF II 状态机*一节中的 *GPIF II 操作*子节。您可以从 **GPIF II Designer** 网页查看相同的文档。

5.3.1 LD_DATA_COUNT （加载数据计数器）

添加到某一状态后，该操作用于配置数据计数器。配置完成后，不能使用别的值重复 LD_DATA_COUNT 操作。

通过使用 FX3 固件中的 CyU3PGpifInitDataCounter() API 函数，可以执行同样的操作。FX3 API 指南提供了函数参数的相关信息。注意：尽管使用了 CyU3PGpifInitDataCounter()函数配置数据计数器，但仍要将该操作添加到 GPIF II 状态机的状态内。否则，转换公式列表中的 DATA_CNT_HIT 事件会不可用。通过使用该 API，可以在运行时更改数据计数器的值。

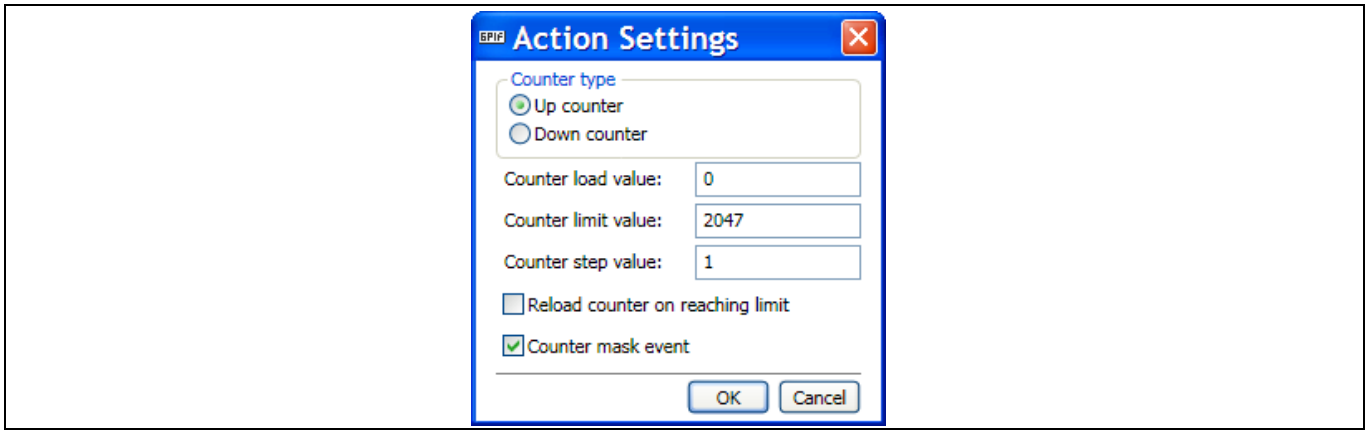

Figure 14 LD_DATA_COUNT 操作设置

下面介绍的是与该操作相关的参数：

- Counter type (Up / Down)：示例中，“Up”项被选中。
- *Counter load value*：执行该操作时，将加载初始计数。示例中，该值等于 0。
- *Counter limit value*：生成事件时的计数值。通过 32 位数据总线读取 8 KB 数据时，该值被设置为 2047。使用下面简单的公式可以计算该值：
  计数器限制值 = （要读取的字节数/ 数据总线宽度（字节）） -1*。
  *只在计数器加载值等于 0 时，才会减去 1。
- *Reload counter on reaching limit*：通过选中该参数框，在到达计数限制值时，可重载计数加载值。否则，它仅作为一次性计数器使用。
- Counter mask event：如果尚未选中该框，当到达计数器限制值时，将生成固件事件。
- *Counter step value*：每次使用 COUNT_DATA 操作时，将要加上/减去计数器步长。该值必须大于 0，以激活 DATA_CNT_HIT 事件。

5.3.2 LD_ADDR_COUNT（加载地址计数器）

该操纵与 LD_DATA_CNT 操作相同。

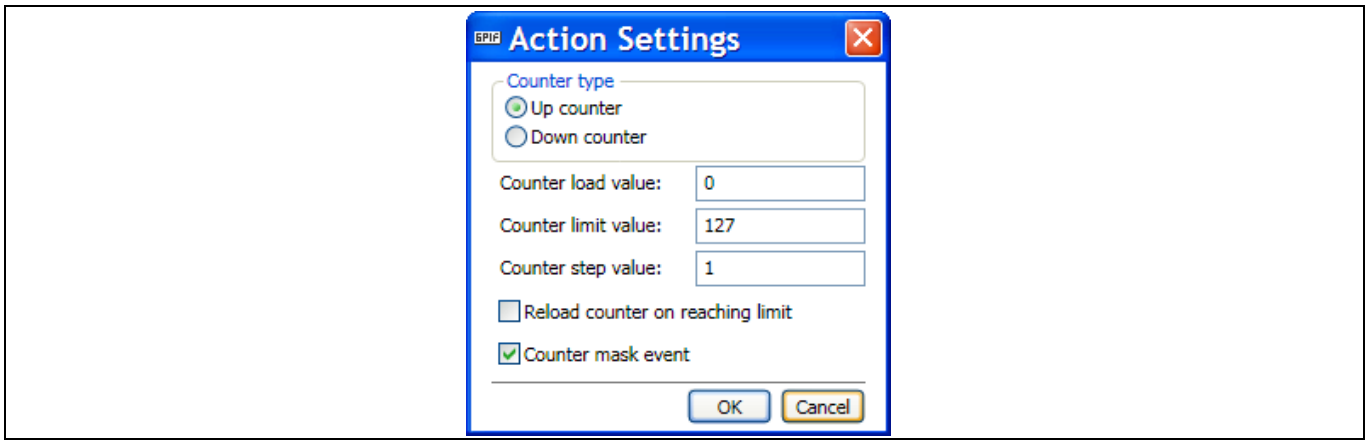

Figure 15 LD_ADDR_COUNT 操作设置

下面介绍的是与该操作相关的参数：

- Counter type (Up / Down)：可配置计数器，使之按照递增/递减顺序计数。
- Counter load value：执行该操作时，将加载初始计数。
- *Counter limit value*：生成事件时的计数值。通过 32 位数据总线读取 512 字节数据时，该值被设置为 127。
- Reload counter on reaching limit：通过选中该参数框，在到达计数限制值时，可重载计数加载值。
- *Counter mask event*：如果尚未选中该框，当到达计数器限制值时，将生成固件事件。
- *Counter step value*：每次使用 COUNT_ADDR 操作时，都会加上/减去计数器步长。

5.3.3 DR_GPIO（驱动 GPIO）

DR_GPIO 操作可将 GPIO 引脚驱动为高电平、低电平或对其进行翻转。在“Assertion Delay”（激活延迟）后，将执行该操作。对于同步操作，该延迟的时长为两个或三个周期；而对于异步操作，该时长为 25 ns 或 30 ns。当转换到下一个状态时，通过该操作驱动的 GPIO 将被取消激活。因此，要想在状态机中所有状态下都将 GPIO 驱动为高电平或低电平，您要将该操作添加到所有状态内。如果不想将它添加到所有状态内，那么请在第一个状态中添加该操作，并选择翻转模式。

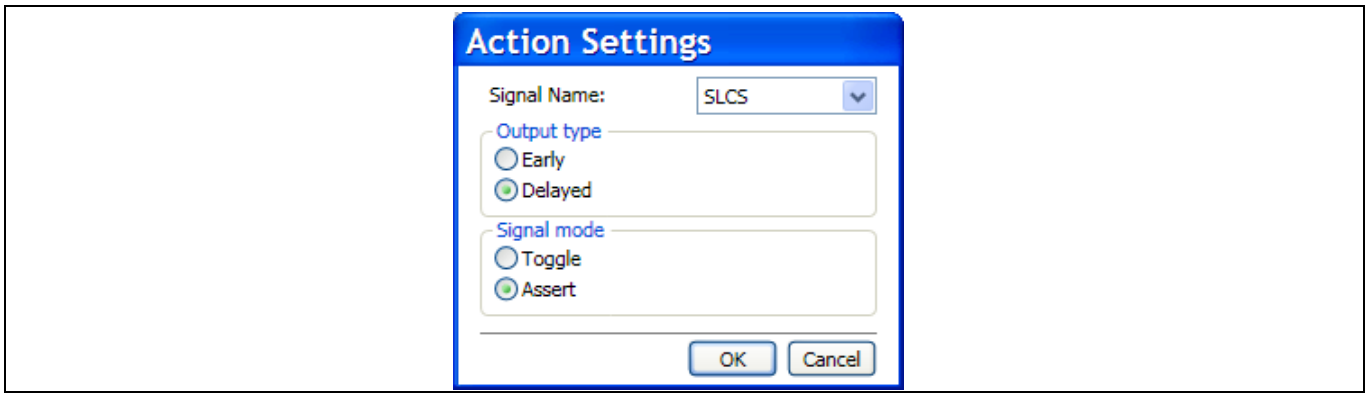

Figure 16 DR_GPIO 操作设置

GPIF **II Designer 工具**

下面介绍的是与该操作相关的参数：

- *Signal name*：用户定义的字母数字字符串，它指示此处可输入的信号。该名称将显示在状态机画布上。
- *Output type*：该参数控制着所驱动的输出信号的延时。当选中 *Early* 参数框时，异步模式的延时为 25 ns，同步模式的延时为两个时钟周期。当选中 *Delayed* 参数框时，异步模式中的延时为 30 ns，同步时钟中的延时为三个时钟周期。
- *Signal mode*：在翻转模式中，当退出包含该信号的状态时，该信号值将被翻转。在激活模式下，信号值被驱动为输出所指定的激活极性。要想选择激活极性，请切换到 Interface Definition 选项卡，然后双击“Application Processor”中的信号名称（例如“OUTPUT0”）。这时，将打开一个对话框，用以设置初始引脚值和高电平有效或低电平有效极性。例如，如果选择低电平有效，当执行“Assert”操作时，引脚将被驱动为低电平。

Note:

1. *每个信号的延迟输出和信号模式设置均是全局设置。因此每次在其他状态下使用该操作时，不能更改这些设置。工具将生成一个与每个输出信号的最后设置相应的配置文件。*
2. *State Settings 对话框中的“Repeat actions until next transition”项并不影响 DR_GPIO 操作的性能。在每个时钟周期内都会重复 DR_GPIO 操作（该时钟可为接口时钟或 FX3 内部时钟）。如果选中“Toggle”项，在当前的状态内，GPIO 将在每个时钟周期内进行翻转。*

5.3.4 IN_DATA（输入数据）

IN_DATA 操作对来自数据总线的数据进行采样，然后将其转移到所指定的位置（可以是 DMA 通道，也可以是固件应用）。 如果将 *Data Sink* 设置为 *Register*，可以使用 *CyU3PGpifReadDataWords()* API 将数据转移到固件内。更多有关该 API 的信息，请参阅 **FX3 API 指南**。注意：指定的目的地为 DMA 通道，且寻址模式被设置为 *Thread selected by State machine (Number of address lines =0)*时，才能直接从四个目的地线程中选择某一个。

能锁存来自数据总线的数据，并不将它保存到所选的位置，或者将先前锁存的数据保存到所选位置。这些选项可满足针对下面情况的协议：总线上的数据可能超前指示信号可用性的信号选通。

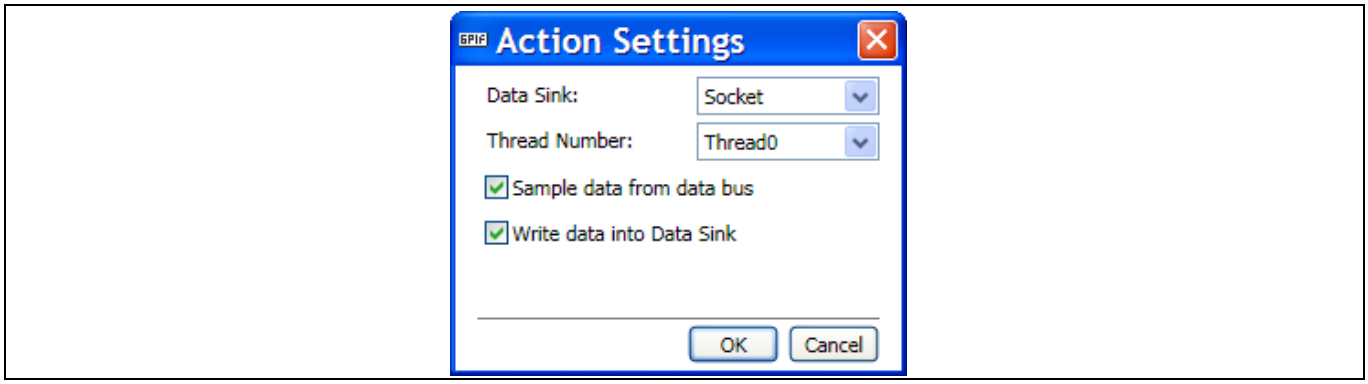

Figure 17 IN_DATA 操作设置

下面介绍的是与该操作相关的参数：

- *Data Sink* — 可用的选项为“Register/ Socket/ PPRegister”

- *Thread Number* — 与数据接收相关联的线程编号。可选的编号为 Thread 0 至 3。（当没有可选择的地址行，且线程编号或主设备模式已被使能时，该特性可用。）
- *Sample data from data bus* — 通过该选项，可对来自数据总线的数据进行采样，但不会将其写入到指定的数据接收器。
- *Write data into Data Sink* — 当用户想要将采样数据推入指定的数据接收器时，请选中该选项。

通过该两个选项，可采样数据，并将数据写入到其目的地。

5.3.5 DR_DATA（驱动数据）

DR_DATA 操作将数据从指定的源驱动到数据总线上。数据源可以是 DMA 通道，也可以是固件应用。如果将 *Data Source* 设置为 *Register*，将使用 *CyU3PGpifWriteDataWords()* API 对来自固件的数据进行写操作。更多有关该 API 的信息，请参阅 **FX3 API 指南** 。注意：仅在数据源为 DMA 通道，且其寻址模式被设置为 *Thread selected by State machine (Number of address lines =0)*时，才能将数据源设置为线程。

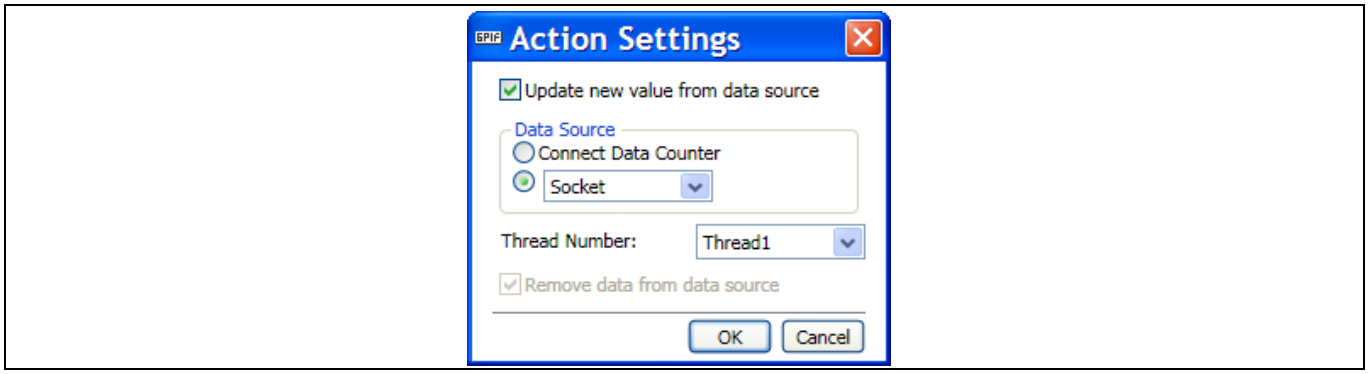

Figure 18 DR_DATA 操作设置

下面介绍的是与该操作相关的参数：

- *Update new value from data source* — 通过该选项，可使用数据源中的数据字更新数据总线，并移除指定源中的数据字。
- *Data Source* — 通过该项，用户可选择“ data counter”和“register/ Socket / PPRegister”。
- *Thread Number* — 与数据源相关联的线程编号。可选的编号为 Thread 0 至 3。当没有用于选择线程编号的地址行时，该特性可用。这时，Interface Definition 选项卡下的 Number of Address Pins Used 字段被设置为 0。
- Remove Data from the Data Source — 禁用该项时，不会消除源数据，因此可在其他状态下使用该数据。

Note: *在 State Settings 对话框中，“Update new value from data source”标志将覆盖“Repeat actions until next transition”标志。尽管已经取消选择 Repeat actions until next transition 标志，但只要选中了 Update new value from data source 标志，就可以在每个时钟周期内更新总线上的数据（GPIF II 处于该状态）。*

5.3.6 IN_ADDR（输入地址）

通过 IN_ADDR 操作，GPIF 硬件采样来自地址总线的值，并使用它选择 DMA 线程或套接字。如果在 Interface Definition 选项卡下指定了 0 地址位，则 Address Selecting 选项被暗掉。

当地址总线宽度小于或等于两位时，地址只能选择四个 DMA 线程中的一个。如果地址宽度介于 3 到 5 位之间，便可以选择 DMA 线程或特定套接字。可使用上面对话框中的“Address Selecting”参数进行选择。

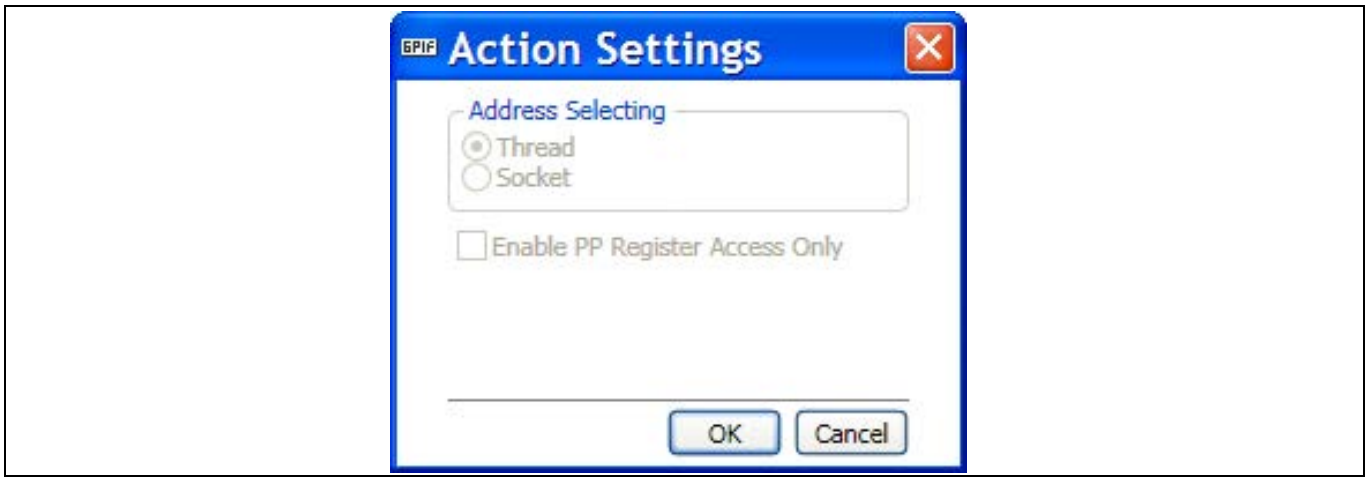

Figure 19 IN_ADDR 操作设置

下面介绍的是与该操作相关的参数：

- Address Selecting — “Thread”或“Socket”

当您更改 Interface Definition 中的值，操纵选项会立即自动更改。例如，如果您将 Number of address pins used 设置为 0，Action List 中的 IN_ADDR 选项将消失。

5.3.7 COUNT_DATA、COUNT_ADDR

使用 LD_DATA_COUNT 操作所配置的步长更新数据计数器的值。如果更新操作使计数达到了所指定的限制值，则 DATA_CNT_HIT 事件的触发将为真。COUNT_ADDR 对地址计数器进行同样的操作。

没有与该操作相应的参数。

5.3.8 INTR_CPU

中断片上 CPU 以生成 CYU3P_GPIF_EVT_SM_INTERRUPT 事件（由固件应用处理）。如果在状态中选择了 INTR_CPU 操作，则转换公式列表中的 INTR_PENDING 事件可用。如果在清除 INTR_PENDING 信号前，需要执行操作，请使用 CyU3PGpifRegisterSMIntrCallback 函数。在 CPU 清除中断前，该回调函数会执行。

没有与该操作相应的参数。

5.3.9 COMMIT

在选定的入口 DMA 通道上传输数据包/缓冲区（将读取数据传入 FX3 器件时所使用的 DMA 通道）。缓冲区将被传输到管道的另一端。通常通过该操作，可强制缓冲区/数据包结束。

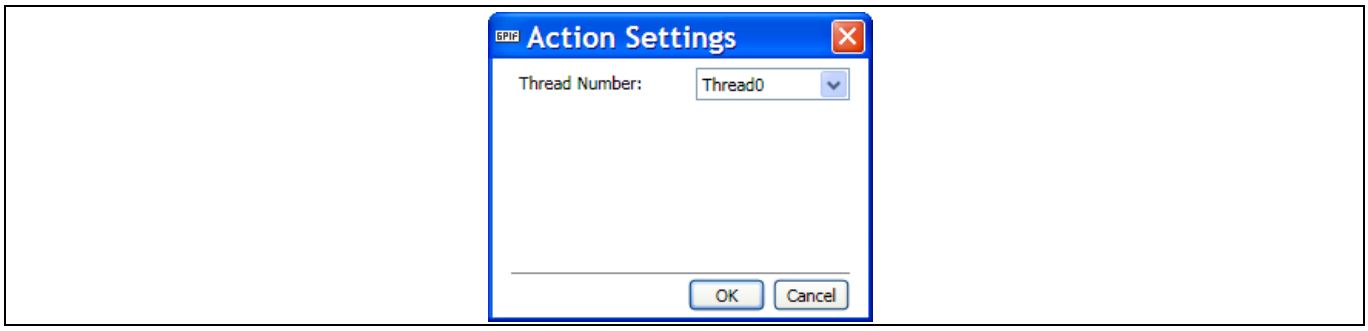

Figure 20 COMMIT 操作设置

下面介绍的是与该操作相关的参数：

- *Thread Number* —Thread0 至 3（仅在地址位的数量被配置为 0 或主控模式被使能时，该参数才可用）。

5.4 GPIF 事件

导致 GPIF II 状态机转换的触发器是通过使用了触发器变量构成的布尔表达式。下表捕获了因执行 GPIF II 操作（在主设备状态机实现使用）而生成的事件。

Table 3 GPIF II 事件说明

GPIF II 事件	引起事件的操作	说明
Input signal name	无；它是一个外部事件	在转换公式中，可使用与任何 GPIO（配置为输入）相关联的名称作为触发器。
LOGIC_ONE	无	无条件地移到下一个状态。
DATA_CNT_HIT	COUNT_DATA	当到达操作设置中的“Counter limit value” 时，该触发器为真。通过 COUNT_DATA 操作，可生成该触发器。
ADDR_CNT_HIT	COUNT_ADDR	当达到操作设置中的“Counter limit value” 时，该触发器为真。通过 COUNT_ADDR 操作，可生成该触发器。
DMA_RDY_CT、DMA_RDY_TH0、DMA_RDY_TH1	IN_DATA DR_DATA	当 DMA 准备好发送或接收数据时，该触发器为真。

6 GPIF II 主设备状态机实现

本节使用上述操作和事件执行主设备状态机。该状态机发送并检索来自同步从设备 FIFO（由第二个 FX3 DVK 实现）的数据。

主设备状态机根据来自从设备 FX3 器件的标志来确定是进行读操作还是进行写操作。

在从设备端，配置 FLAGA 以指示 Thread 1 的就绪状态（Thread_1_DMA_Ready），并配置 FLAGB 以指示 Thread 0 的就绪状态（Thread_0_DMA_Ready）。FLAGA 指示从设备端数据的可用性，FLAGB 指示从设备端空闲缓冲区的可用性。

在主设备固件中创建两个 DMA 通道，以进行双向数据传输。为将数据从 FX3 的 USB 端传输到 GPIF II，请将 PIB 套接字 1 配置为使用套接字，并将 USB 套接字配置为生成套接字。如要将数据从 GPIF II 传输到 USB，请将 PIB 套接字 0 配置为生成套接字，并将 USB 套接字配置为使用套接字（请查看 **Figure 21**）。

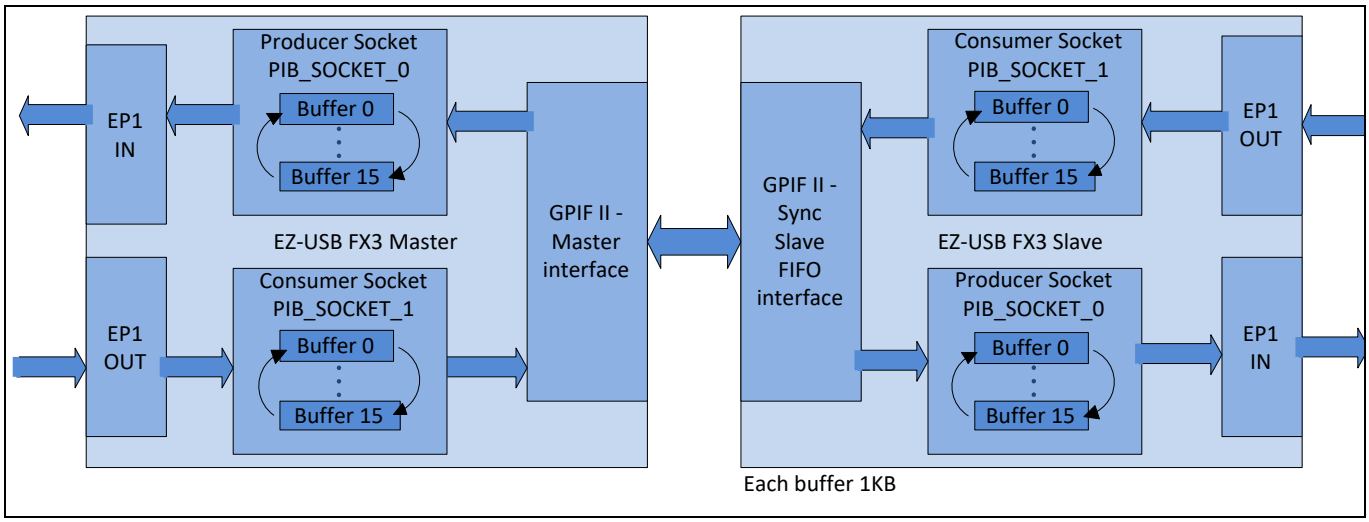

Figure 21 FX3 从设备和主设备的 PIB 套接字

使用套接字/寄存器/地址计数器，主控 FX3 可通过地址总线驱动地址。在本应用中，地址需要选择从设备 FX3 上的某个线程。为方便实现该过程，请将两个地址行（A1 和 A0）配置为 GPIO，并且通过驱动这些 GPIO 寻址从设备 FX3 的线程。

主设备 FX3 提供时钟给从设备 FX3，数据总线宽度为 32 位。从主设备 FX3 的 GPIF II 驱动从设备 FIFO 接口所需要的控制信号。

6.1 FX3 GPIF II 主设备完成状态机

Figure 22 显示的是 GPIF II 主设备状态机。主设备 FX3 和从设备 FX3 间不传输数据时，主设备状态机将进入 RD_WR_IDLE 状态。在该状态中加载并寻址计数器。在附加的示例项目中，数据计数器的值和地址计数器的值被设置为 511（多个数据包大小，USB 2.0 中的 512，USB 3.0 中的 1024），用于对传入/传出缓冲器的数据进行计数。

如果 **Figure 22**、**Figure 23** 和 **Figure 24** 中的转换公式不够清楚，请参阅附带的 GPIF 项目。

该状态（RD_WR_IDLE）会连续检查 FLAGA 或（DMA_RDY_TH1&FLAG_B）事件是否为真，因此它可以转到状态机的读/写操作状态。如果从设备 FX3 上使用套接字 1 的 DMA 缓冲区中存有某些数据，则 FLAGA 将变为高电平。因此，FLAGA 指示从设备上存在有效数据，主设备需要启动读取操作。在驱动读操作所需要的任何控制信号前，主设备必须先检查缓冲区是否空闲。通过使用 DMA_RDY_TH0 来确定缓冲区的状态。

GPIF II 主设备状态机实现

如果要执行写操作，主设备 FX3 上的使用套接字 1 的 DMA 缓冲区中必须存储了有效数据。可通过 DMA_RDY_TH1 查找该条件。此外，从设备端要存在空闲的缓冲区。可通过 FLAGB 验证该条件。

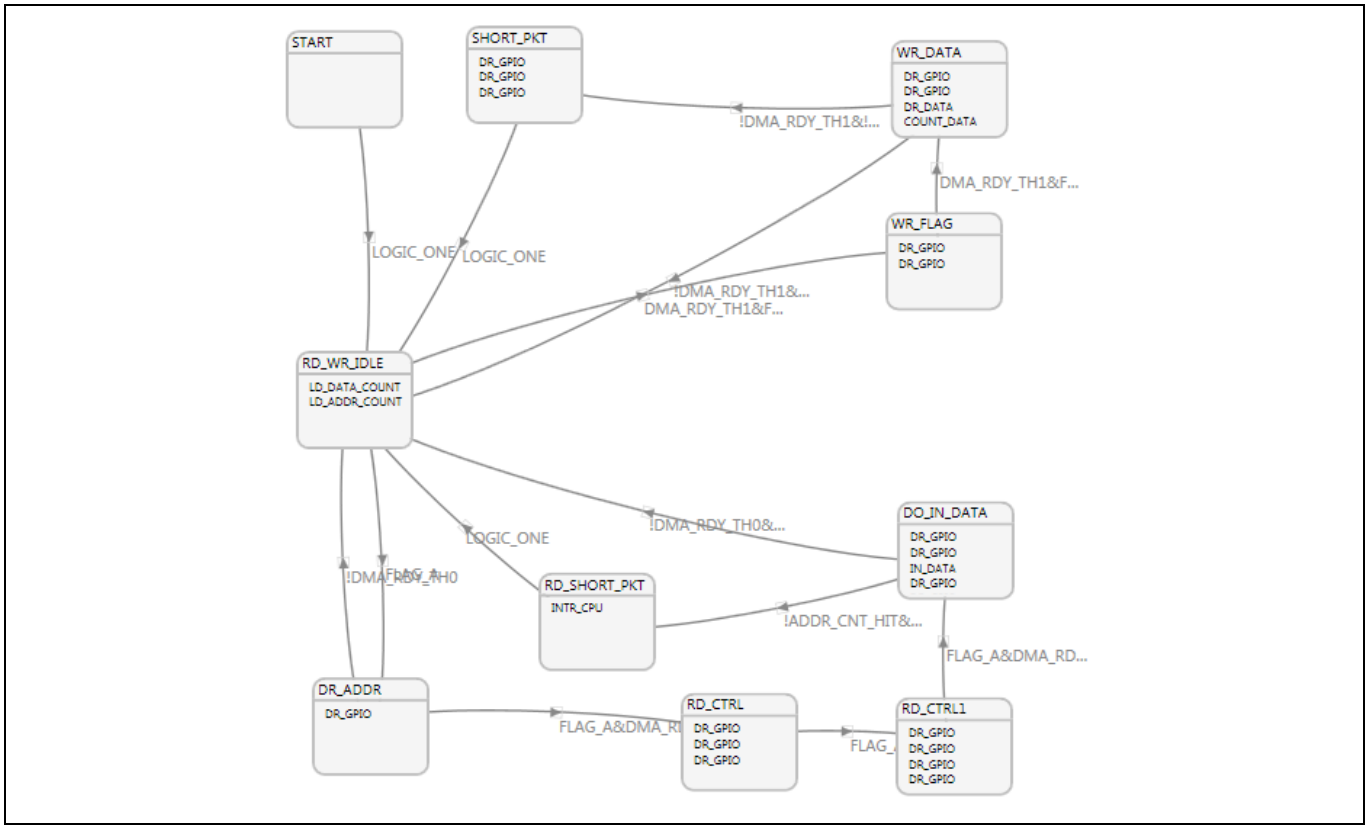

Figure 22　GPIF II 主设备状态机

6.2　FX3 主设备写入到 FX3 从设备

Figure 23 显示的是状态机中主设备 FX3 将数据写入到从设备 FX3 的情况。请查看 **Figure 6**，了解有关需要由主设备 FX3 驱动（以写入到从设备 FIFO）的信号列表。

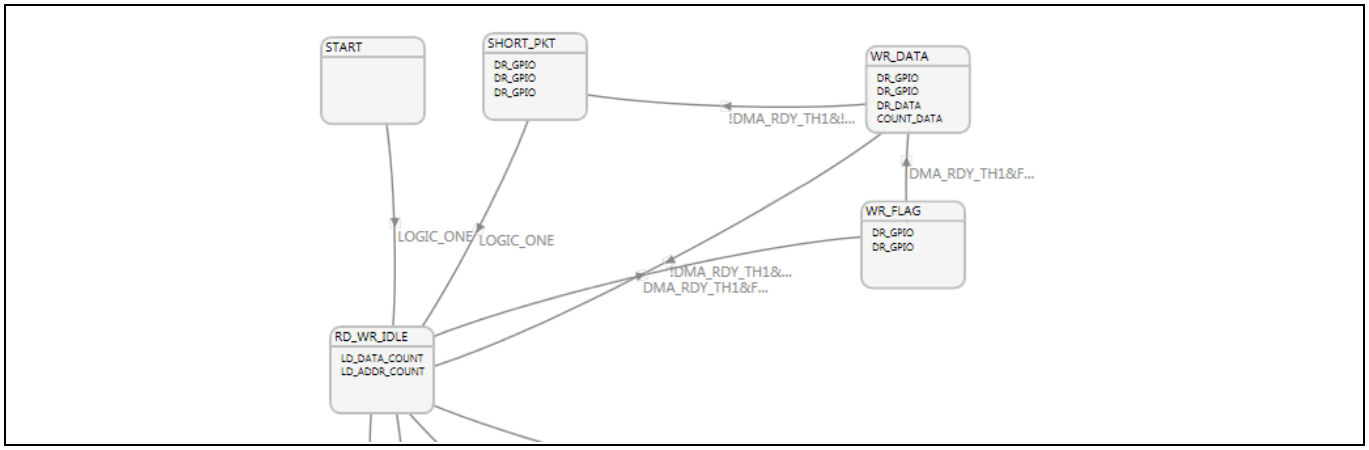

Figure 23　主设备状态机的写入部分

当主设备 DMA 缓冲区中存有某些数据，并且从设备的缓冲区处于空闲状态，则主设备状态机将从 RD_WR_IDLE 状态转换为 WR_FLAG 状态。DMA_RDY_TH1 指示主设备端数据的可用性，FLAGB 指示从设备

端空闲缓冲区的可用性。从该状态切换到 WR_DATA 状态前，主设备将驱动数据和控制信号 SLCS#和 SLWR#。

配置通道时，可以将多个缓冲区分配给一个特殊的 DMA 通道（使用固件中的 CyU3PDmaChannelCreate API 时，请参阅 **FX3 API 指南**，了解 API 的相关信息）。注意：标志分别表示的是每个缓冲区的满/空状态。

单个缓冲器的最大空间为 64 KB-16。例如，如果已经将两个大小均为 512 字节的缓冲区分配给 DMA 通道，当将 512 字节写入到第一个缓冲区后，满载 FLAG 的状态将表示为“满”。它一直表示为该状态，直到 DMA 通道转到第二个缓冲区为止。DMA 通道转到下一个缓冲区所需的时长不长，但通常需要几微秒。

主控状态机进入 RD_WR_IDLE 状态，并在该缓冲器切换延迟器件重新加载地址计数器。通过 ADDR_CNT_HIT 事件查找该缓冲区切换条件。不能使用 DMA_RDY_TH1 标志实现该操作，因为更新该标志需要一个时钟周期的内部延时。当下一个缓冲区可用时，状态机将返回到 WR_DATA 状态。

当 DMA 缓冲区中没有更多数据，且尚未生成 ADDR_CNT_HIT 事件时，会产生短数据包。在驱动短数据包的同时，会驱动 PKTEND#和 SLWR#信号。

6.3 FX3 主设备对 FX3 从设备进行读操作

Figure 24 显示的是状态机中主设备 FX3 读取从设备 FX3 中的数据的情况。请查看 **Figure 5**，了解有关需要由主设备 FX3 驱动（以读取从设备 FIFO）的信号列表。

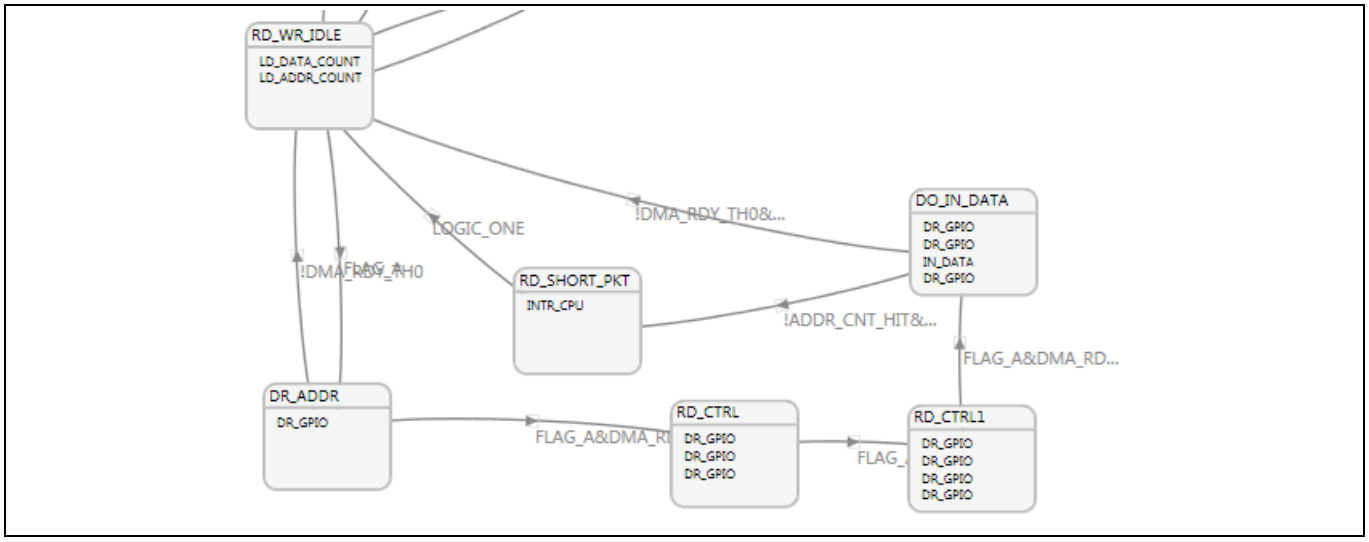

Figure 24 主设备状态机的读取部分

当从设备端 DMA 缓冲区中存在数据时，主设备状态机将从 RD_WR_IDLE 状态转换到 DR_ADDR 状态。在该状态中，主设备驱动地址行。通过将 A0 驱动为高电平来寻址从设备 FX3 的线程 1。然后，它检查是否存在空闲缓冲区，并根据 DMA 缓冲区的可用情况转换到 RD_CTRL 状态。在该状态中，主设备驱动地址和控制信号 SLCS#和 SLOE#。然后，它将进入下一个状态，并驱动 SLRD#信号（除了已驱动的信号外）。随后，它将进入 DO_IN_DATA 状态。在该状态中，通过驱动所有需要的控制信号连续读取数据。

缓冲区切换延时存在，如状态机的写入部分所述。在该延时期间，状态机进入 RD_WR_IDLE 状态，重新加载地址计数器，然后在下一个缓冲器就绪时返回到 DO_IN_DATA 状态。

当从设备 FX3 的 DMA 缓冲区中没有更多数据，且尚未生成 ADDR_CNT_HIT 事件时，会产生短数据包。然后，状态机会进入 RD_SHORT_PKT 状态，并调用 INT_CPU 操作。这样会生成 FX3 固件的中断，并使 DMA 通道得到保护。请在随附的固件中查看 CyFxApplnGPIFEventCB()。

GPIF II 主设备状态机实现

Note: *您也可以使用 COMMIT 操作代替 INT_CPU 操作。但除了 COMMIT 外，还要使用 IN_DATA 操作；否则，会生成长度为零的数据包（ZLP）。因此，如果使用 INT_CPU 替换 COMMIT 和 IN_DATA 操作，请确保在 LD_DATA_COUNTER 操作设置中减去一个计数。*

Note: *通过寄存器执行读/写操作时，请使用 OUT_REG_VALID 和 IN_REG_VALID 事件。（请将 IN_DATA 或 DR_DATA 操作中的“Data Source”项设置为“Registers”）。*

如果您打算将数据写入到外设器件上某个特殊地址位置内，然后将该地址和数据写入到固件的入口寄存器（用于从 FX3 器件传出数据的寄存器），然后，通过监控状态机中的这些事件可以检查是否存在有效的数据。

但是，当您使用 DMA 缓冲区传输大量数据时，需要检查 DMA 的就绪标志。只要缓冲区中存在有效数据，DMA 的就绪标志便会告诉您该信息。

7 FX3 主设备固件实现

除了 GPIF-II 状态机外，FX3 固件还用于创建 DMA 通道，启动这些通道以进行数据传输。该固件也用于设置数据计数器和地址计数器的值。在 USB 套接字和 GPIF 套接字之间创建了两个 AUTO DMA 通道，用来传输数据。每个 DMA 通道包含 16 个缓冲器。如果与 USB 3.0 端口相连接，每个缓冲器大小为 2048 字节。如果与 USB 2.0 端口连接，大小将为 512 字节。在主设备模式下，CyFxApplnSetPibDllParameters() API 用于设置适当的时钟相位。此外，如果与 USB 3.0 端口相连接，数据计数寄存器的初始值为 511（2047 字节）。如果与 USB 2.0 端口相连接，该寄存器的初始值为 127（512 字节）。当从 GPIF-II 状态机得知操作被 INTR_CPU 中断时，当前缓冲器将被打包，并且在 CyFxApplnGPIFEventCB() API 中准备好该缓冲区，以将它传输到 USB。

8 硬件连接

通过使用被编程为同步从设备接口的第二个 FX3 DVK 电路板，可以测试 GPIF II 主设备设计。**Figure 25** 显示了两个 FX3 芯片间的接口连接。**Figure 26** 显示的 FX3 背对背设置使用了两个 FX3 SuperSpeed Explorer 套件（CYUSB3KIT-003）。两个 SuperSpeed Explorer 开发板互相连接，可使他们的 GPIF 接口相连接。插入式连接器（Samtec 器件型号 SSW-120-03-G-D）具有两项功能：隔离开两个开发板，这样 USB 3.0 连接器便不受干扰。应该剪掉插入式连接器上的一些引脚，以避免引起两个开发板间电源引脚和低速外设引脚的短接。**Figure 27** 显示的是在将插入式连接器插入 FX3 开发板前应该剪掉的引脚。

Note: *如果你正在使用的是 CYUSB3KIT-001，请查阅**附录：使用 FX3 开发套件（CYUSB3KIT-001）时的硬件设置**一节，了解有关硬件设置的详细信息。*

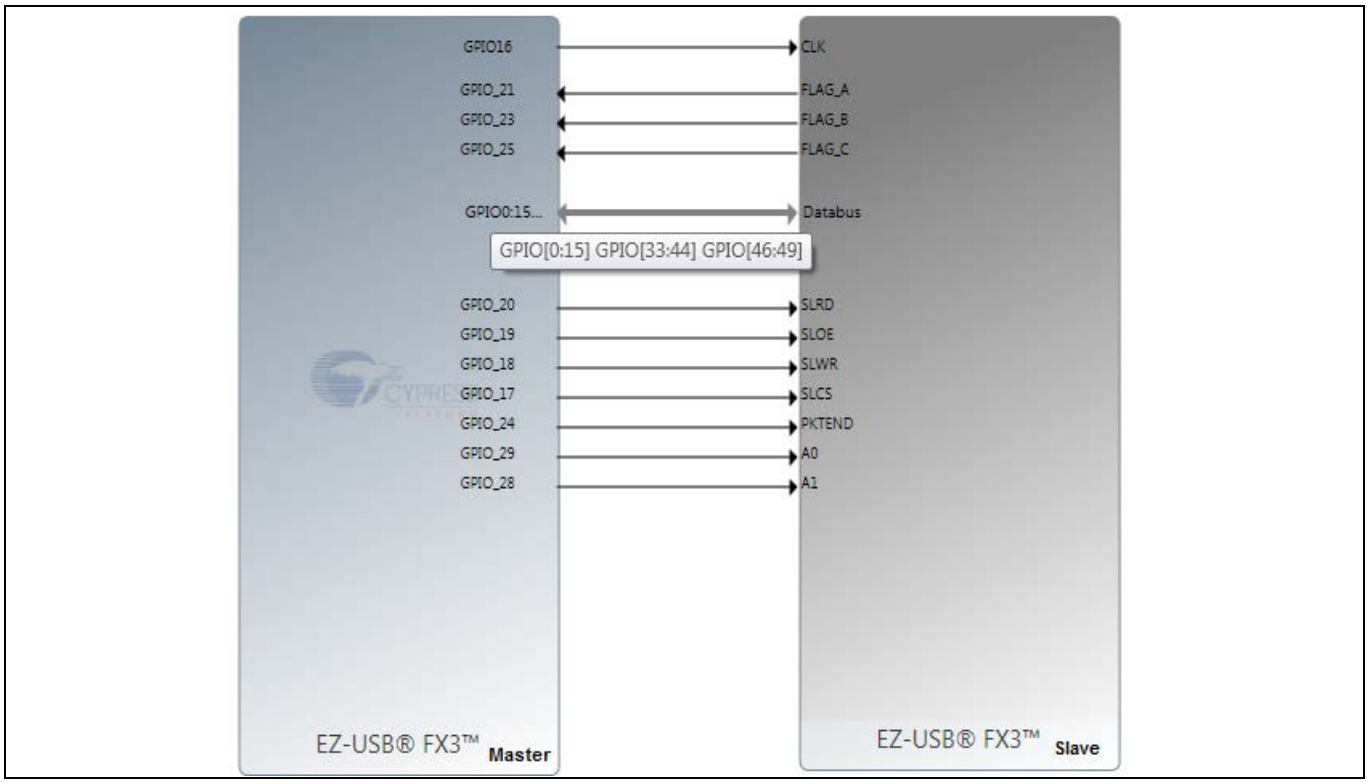

Figure 25 硬件连接

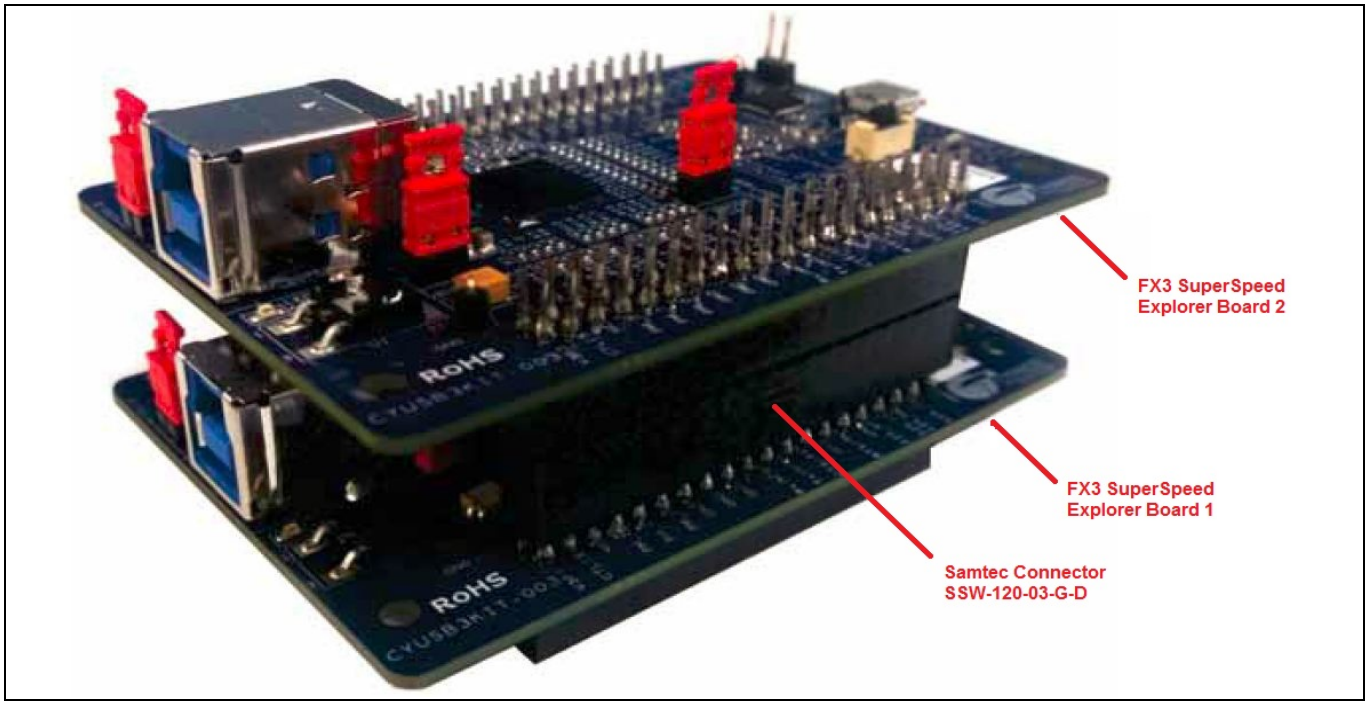

Figure 26　使用 CYUSB3KIT-003 时的 FX3 背对背设置

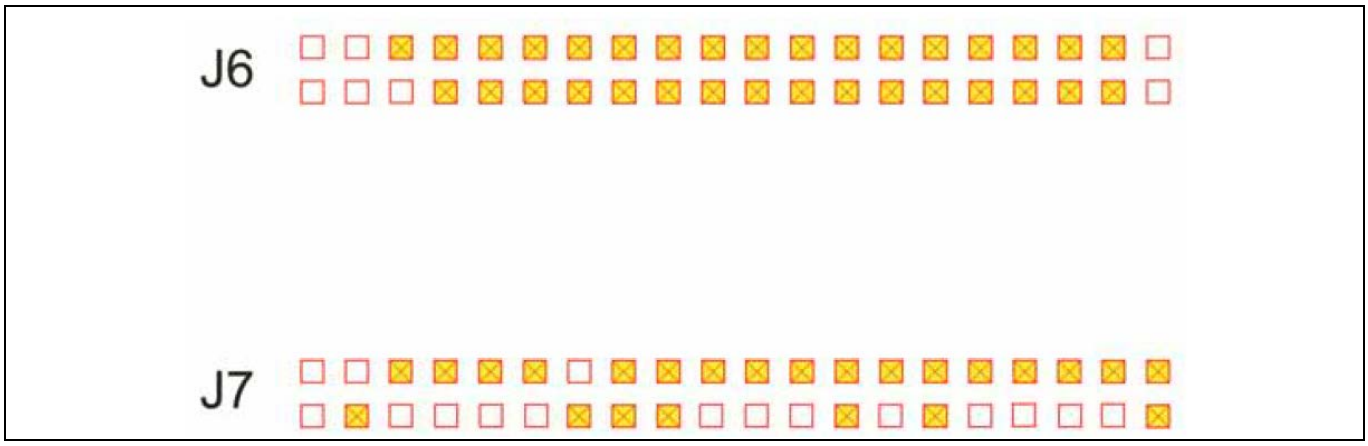

Figure 27　剪掉插入式连接器上的白色引脚

9 运行演示版的步骤

本应用笔记中附上用于主设备和从设备 FX3 的固件源代码和 GPIF II 状态机。本节向您介绍了如何使用本应用笔记所附带的文件来运行演示版的各步骤。

1. 使用互联电路板连接两个 FX3 DVK。请参考硬件连接部分中的 **Figure 26**。
2. 使用 USB 3.0 线缆将两个 FX3 DVK 连接至 PC。这样，该两个器件作为赛普拉斯 Bootloader 器件进行枚举。
3. 使用 USB Control Center 将固件下载到 FX3 器件的 RAM 中。

Note: *请先下载从设备器件的先前构建固件（附件中“bin”文件夹的 Autoslave.img），然后才下载主控器件的固件（附件中“bin”文件夹的 Automaster.img）。*

4. 您可观察从设备和主设备器件的枚举操作，如 **Figure 28** 所示。从设备 FX3 器件使用了 0x4B4 的 VID 和 0x00F2 的 PID。主设备 FX3 使用的是 0x4B4 的 VID 和 0x00F4 的 PID。

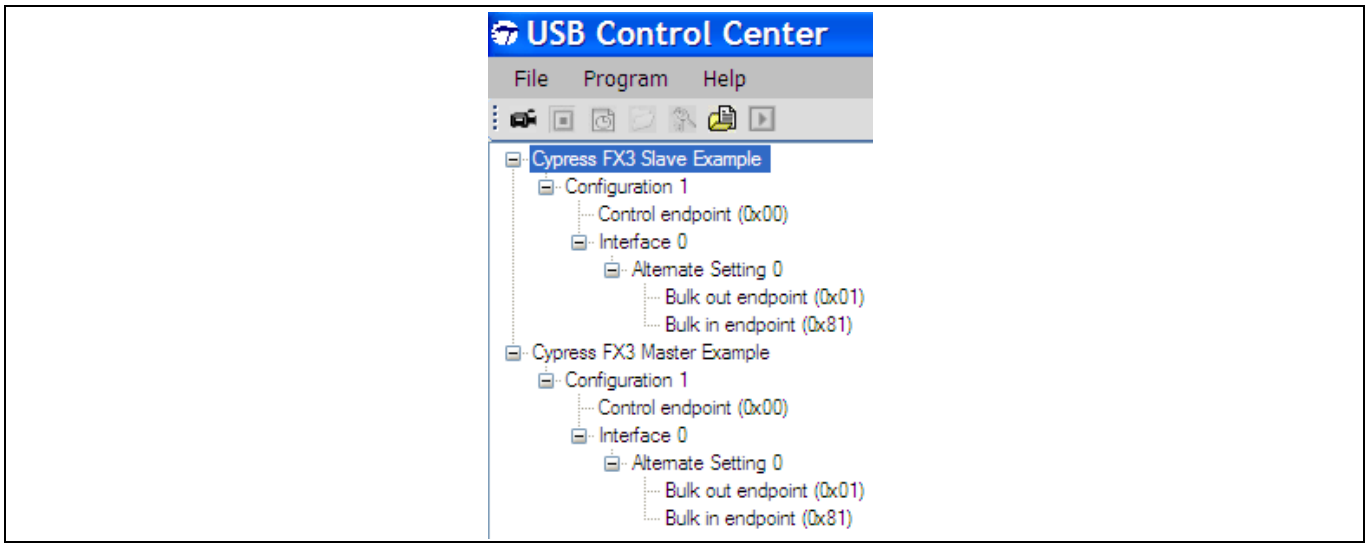

Figure 28 主设备和从设备 FX3 器件

5. 每个器件都有一个 Bulk OUT 端点和一个 Bulk IN 端点。将数据传输到从设备 FX3 的 Bulk OUT 端点，并从主设备 FX3 的 Bulk in 端点读取该数据。请点击 Transfer File-OUT 按键，并浏览到 8192_count.hex（位于附件中的“bin”文件夹内）。该操作可将 8 kB 的数据传输到从设备 FX3 的 Bulk OUT 端点，如 **Figure 29** 所示。**Figure 30** 显示的是从主设备 FX3 的 Bulk in 端点读取的数据。

运行演示版的步骤

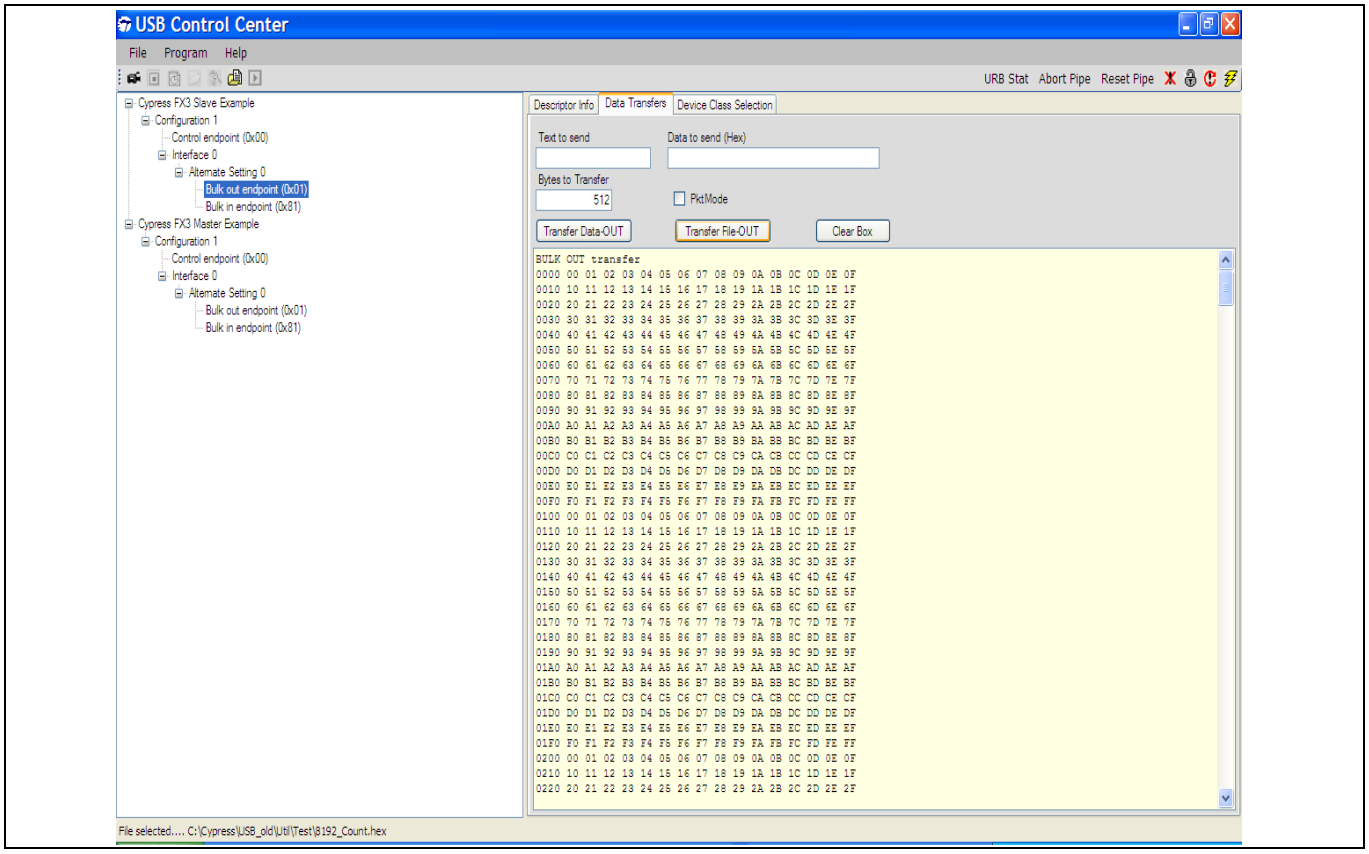

Figure 29　将数据传输到从设备 FX3 的 Bulk OUT 端点

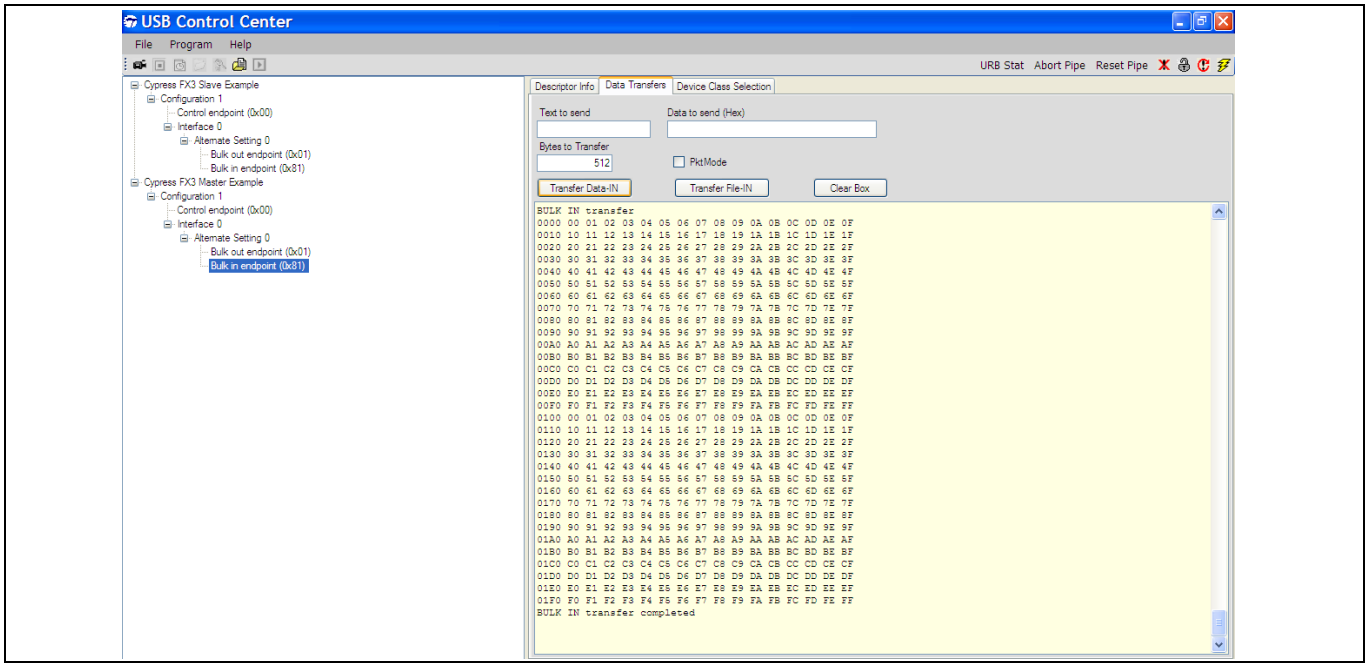

Figure 30　从主设备 FX3 的 Bulk IN 端点读取数据

6. 您也可以将某些短数据包从一端传输到另一端。**Figure 31** 显示的是将 18 个字节传输到主设备 FX3 的 Bulk OUT 端点。**Figure 32** 显示的是读取从设备 FX3 的 Bulk IN 端点上的数据。由于是 32 位数据总线进行传输，所以您可在该图中观察到两个额外的 0 字节。

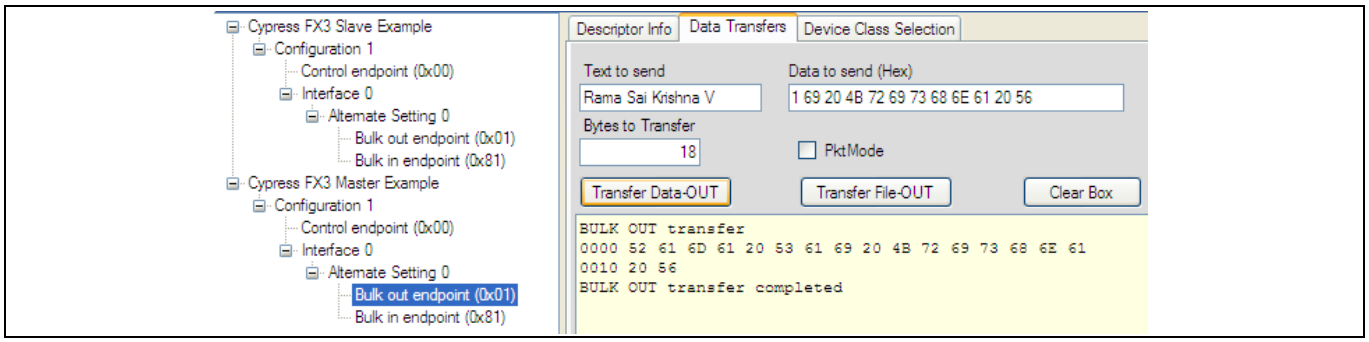

Figure 31　传输一个短数据包到主设备 FX3 的 Bulk OUT 端点

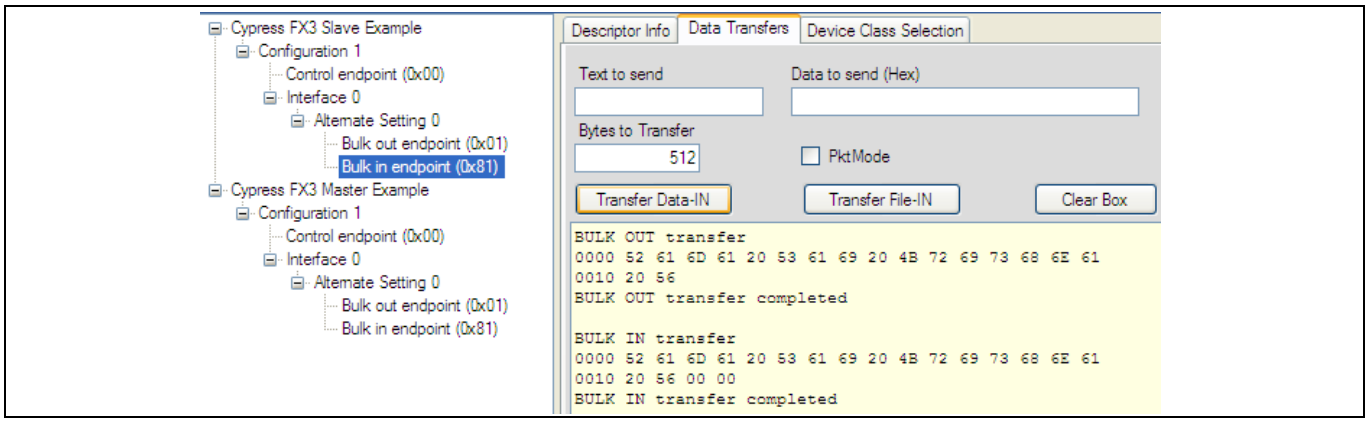

Figure 32　读取来自从设备 FX3 的 Bulk IN 端点上的短数据包

7. **Figure 33** 显示的是传输 8 个字节到主设备 FX3 的 Bulk OUT 端点。**Figure 34** 显示的是读取从设备 FX3 的 Bulk IN 端点上的数据。

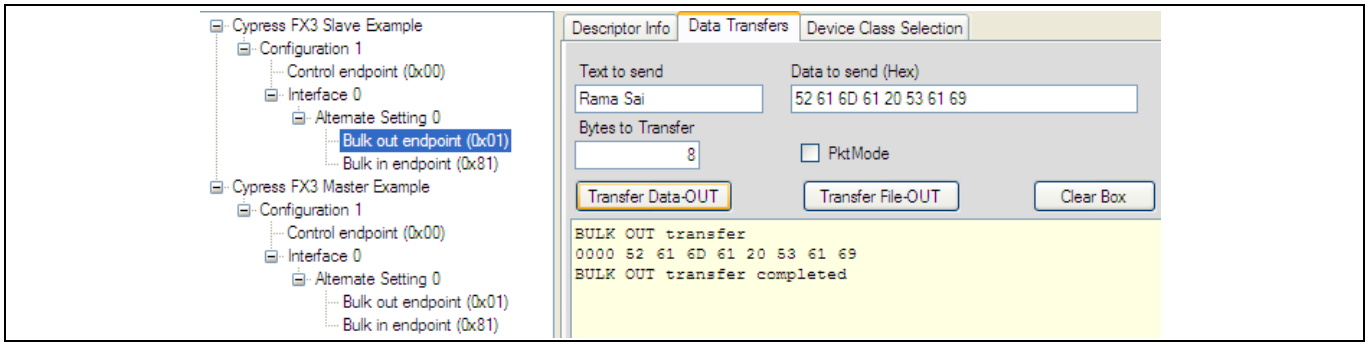

Figure 33　传输一个短数据包到主设备 FX3 的 Bulk OUT 端点

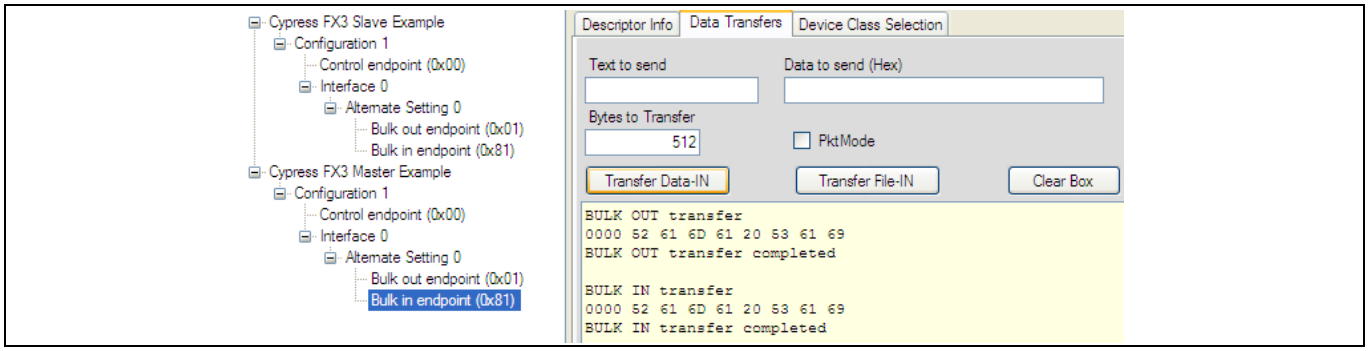

Figure 34　读取从设备 FX3 的 Bulk IN 端点上的短数据包

10 相关的项目文件

Table 4 显示了附带本应用笔记的文件。

Table 4 本应用笔记所附带的文件说明

文件/文件夹	说明
FX3 固件	该文件夹包括下面各文件夹： *AutoSlave* — 从设备 FX3 的 FX3 固件源文件 *AutoMaster* — 主设备 FX3 的 FX3 固件源文件
GPIF II 项目	该文件夹包括下面各文件夹： *master_read_write_sync.cydsn* — 主设备 FX3 的 GPIF II 项目 *sync_slave_fifo_2bit_editable_latestGPIF.cydsn* — 从设备 FX3 的 GPIF II 项目
bin	该文件夹包括下面的文件： *AutoSlave.img* — 从设备 FX3 的 FX3 固件文件 *AutoMaster.img* — 主设备 FX3 的 FX3 固件文件。可将这些固件文件下载到 FX3 DVK 里，用以查看本应用笔记的快速演示。 *8192_count.hex* — 包含 8192 个递增数据的文件。该文件用以在 FX3 的 OUT 端点进行 8 KB 的文件传输。

11 总结

本应用笔记介绍了您需要了解的所有信息，目的是为通过 FX3 的 GPIF II Designer 工具执行一个主设备接口与同步从设备 FIFO 接口相兼容。您可将本设计作为起点，根据应用要求开发一个 GPIF II 主设备状态机。

12 相关应用笔记

AN65974 – 使用 EZ-USB® FX3 从设备 FIFO 接口进行设计

关于作者

姓名： Rama Sai Krishna

职务：高级应用工程师

13 附录：使用 FX3 开发套件（CYUSB3KIT-001）时的硬件设置

Figure 35 显示的是使用两个背对背相连接的 FX3 开发套件（CYUSB3KIT-001）时的硬件设置。

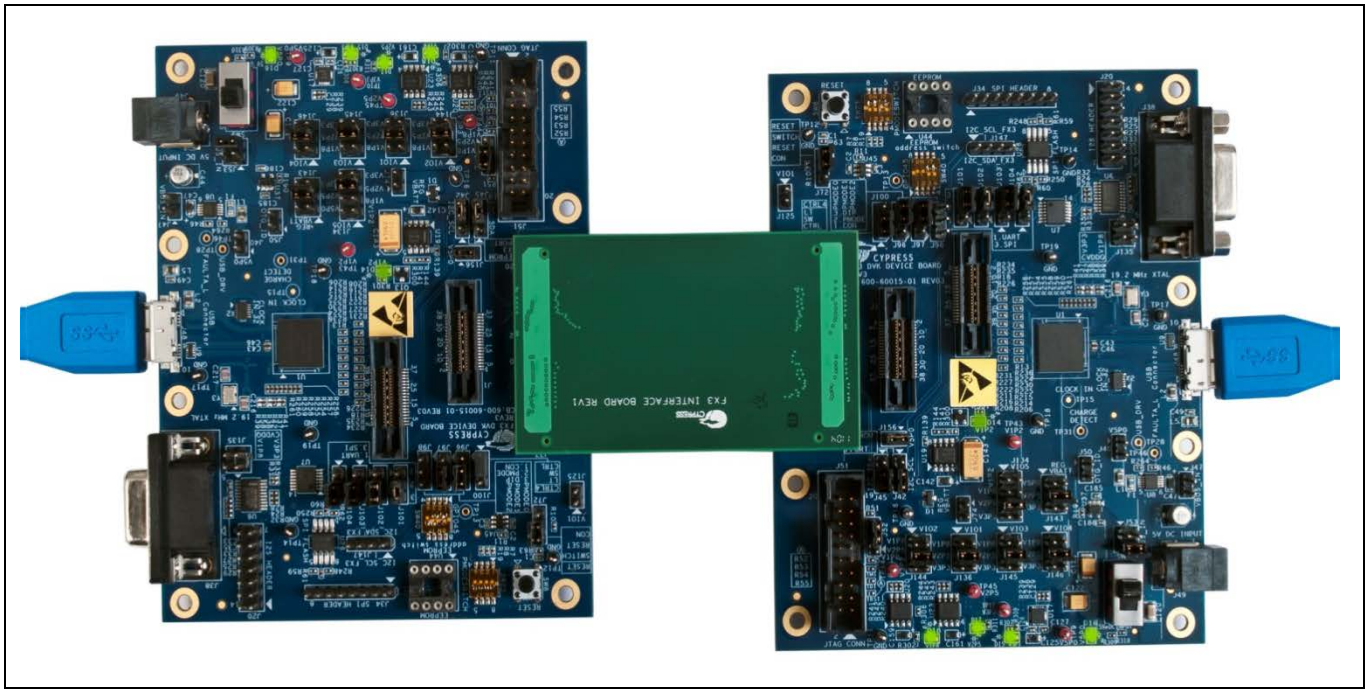

Figure 35 图 1. FX3 Rev3 DVK 的背对背连接

Note: 请确保在两个 FX3 DVK 上都使用了跳线 J100 将柱 1 和柱 2 连接起来。这时，CTRL[4] (GPIO[21]) 将被配置为 FLAGA。要想获取两个 FX3 DVK 间的互联电路板的原理图，请联系赛普拉斯。

文档修订记录

版本	提交日期	变更描述
**	2014-04-11	本文档版本号为 Rev**，译自英文版 001-87216 Rev*A。
*A	2016-02-29	本文档版本号为 Rev*A，译自英文版 001-87216 Rev*B。
*B	2017-04-27	更新徽标和版权。
*C	2019-04-15	本文档版本号为 Rev*C，译自英文版 001-87216 Rev*D。
*D	2021-03-23	本文档版本号为 Rev*D，译自英文版 001-87216 Rev*E 并修改为 Infneon 格式。

商标

所有提及的产品或服务名称以及商标均系其各自所有者财产。

Edition 2021-03-23

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2021 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Go to www.cypress.com/support

Document reference

001-92091 Rev. *D

重要提示

本文档所提供的任何信息绝不应当被视为针对任何条件或者品质而做出的保证（质量保证）。英飞凌对于本文档中所提及的任何事例、提示或者任何特定数值及/或任何关于产品应用方面的信息均在此明确声明其不承担任何保证或者责任，包括但不限于其不侵犯任何第三方知识产权的保证均在此排除。

此外，本文档所提供的任何信息均取决于客户履行本文档所载明的义务和客户遵守适用于客户产品以及与客户对于英飞凌产品的应用所相关的任何法律要求、规范和标准。

本文档所含的数据仅供经过专业技术培训的人员使用。客户自身的技术部门有义务对于产品是否适宜于其预期的应用和针对该等应用而言本文档中所提供的信息是否充分自行予以评估。

如需产品、技术、交付条款和条件以及价格等进一步信息，请向离您最近的英飞凌科技办公室接洽（**www.infineon.com**）。

警告事项

由于技术所需产品可能含有危险物质。如需了解该等物质的类型，请向离您最近的英飞凌科技办公室接洽。

除非由经英飞凌科技授权代表签署的书面文件中做出另行明确批准的情况外，英飞凌科技的产品不应当被用于任何一项一旦产品失效或者产品使用的后果可被合理地预料到可能导致人身伤害的任何应用领域。