Altera FPGA 全速漂移 开发指南

基于 FPGA-FX3 SlaveFIFO 接口的

loopback 实例

欢迎加入 FPGA/CPLD 助学小组一同学习交流：

EDN:

http://group.ednchina.com/GROUP_GRO_14596_1375.HTM

ChinaAET: http://group.chinaaet.com/273

淘宝店链接：http://myfpga.taobao.com/

技术咨询：orand_support@sina.com

特权 HSC 最新资料例程下载地址：

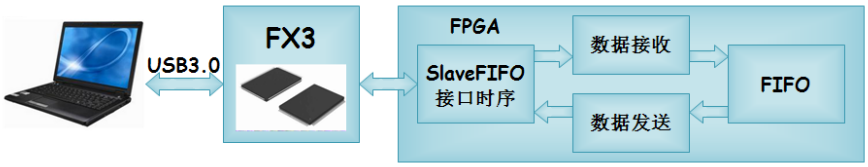
http://pan.baidu.com/s/1pLmZaFx

版本信息		
时间	版本	状态
2016-07-25	V1.00	创建。

目录

1 Loopback 功能概述

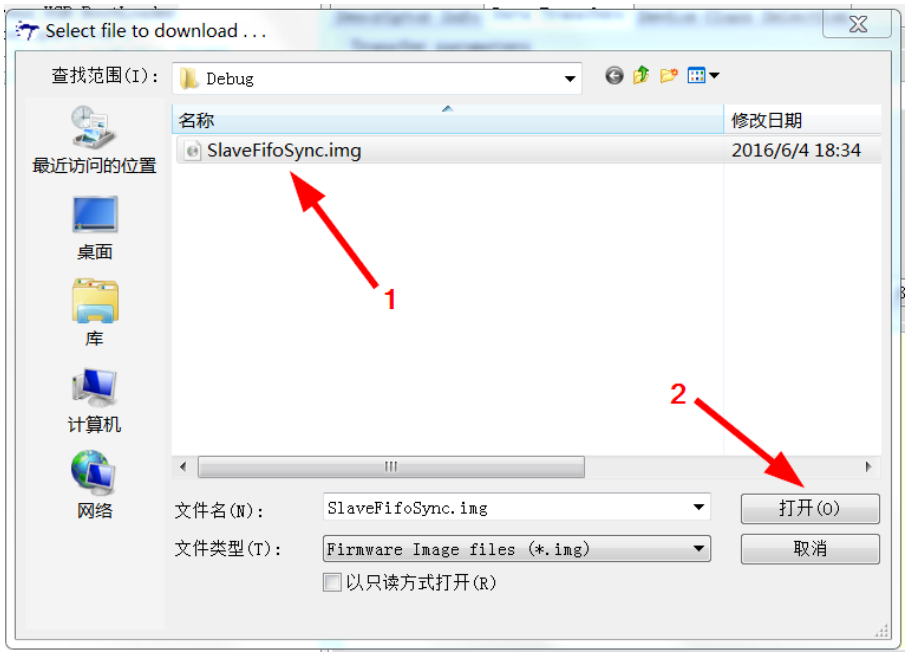
如图所示，工程主要是 PC 端发送数据到 FX3，FX3 通过指示位 flaga 告知 FPGA 有数据待读取，FPGA 端便通过 SlaveFIFO 接口读取 PC 端发送过来的数据缓存到 FPGA 内部的 FIFO 中，FPGA 在完成读取操作后，发起一次 SlaveFIFO 的写入操作，将接收到的数据通过 FX3 最终返回到 PC 端。整个数据的收发过程，我们在 FPGA 内部可以通过在线逻辑分析仪 SignalTap II 抓取 SlaveFIFO 接口的所有信号进行查看。



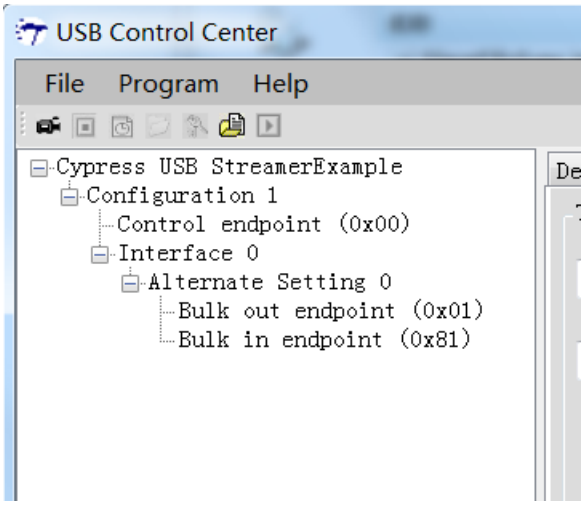
2 Firmware 下载

注意本实验需要同时对 FX3 和 FPGA 进行下载编程配置，FX3 要先下载，随后下载 FPGA。否则时序可能错乱，实验可能失败。

FX3 中下载“...\prj\hsc_ex8\SlaveFifoSync\Debug”路径下的 SlaveFifoSync.img 文件。

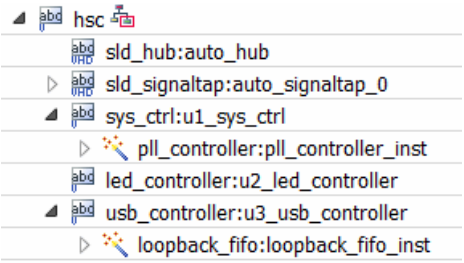


下载完成后，USB Control Center 如图所示。



3 FPGA 代码解析

本实例分为 3 个层级，共计 6 个模块，其层次结构如图所示。



各个模块的基本功能定义如表所示。

表 Verilog 各个模块功能描述

模块名	功能描述
hsc.v	该模块是顶层模块，其下例化了 3 个子模块。该模块仅仅用于子模块间的接口连接，以及和 FPGA 外部的接口定义，该模块中未作任何的逻辑处理。
usb_controller.v	该模块产生 FX3 的 SlaveFIFO 读写操作时序，该模块检测 FX3 的 SlaveFIFO 中是否有发送来的数据，并将这些数据读取和缓存到 FPGA 的片内 FIFO 中，随后将这些数据写会到 FX3 的 SlaveFIFO 中。
	loopback_fifo.v 该 FIFO 用于缓存 FX3 读出的数据。
led_controller.v	进行 24 位计数器的循环计数，产生分频信号用于实现 LED 指示灯的闪烁。
sys_ctrl.v	该模块中例化了 PLL 模块，并且对输入 PLL 的复位信号以及 PLL 锁定后的复位信号进行“异步复位，同步释放”的处理，确保系统的复位信号稳定可靠。
	pll_controller.v 该模块为 FPGA 器件特有的 IP 硬核模块，其主要功能是产生多个特定输入时钟的分频、倍频、相位调整后

	的输出时钟信号。

usb_controller.v 模块是最主要的 SlaveFIFO 及其相关功能实现的模块，我们可以重点看看它的功能。

该模块的功能框图如图所示。FX3 读写状态机在检测到 FX3 的 SlaveFIFO 有可读数据的标志信号后，进入 FX3 数据读取的状态，将 SlaveFIFO 中的数据全部读取出来，缓存到 FPGA 片内的 FIFO 中。接着状态机切换到 FX3 写数据状态，将缓存在 FIFO 中的数据全部回写到 FX3 的 SlaveFIFO 中。这样，我们在 PC 端的 USB 调试工具上所发送的数据，将会被原样送回。

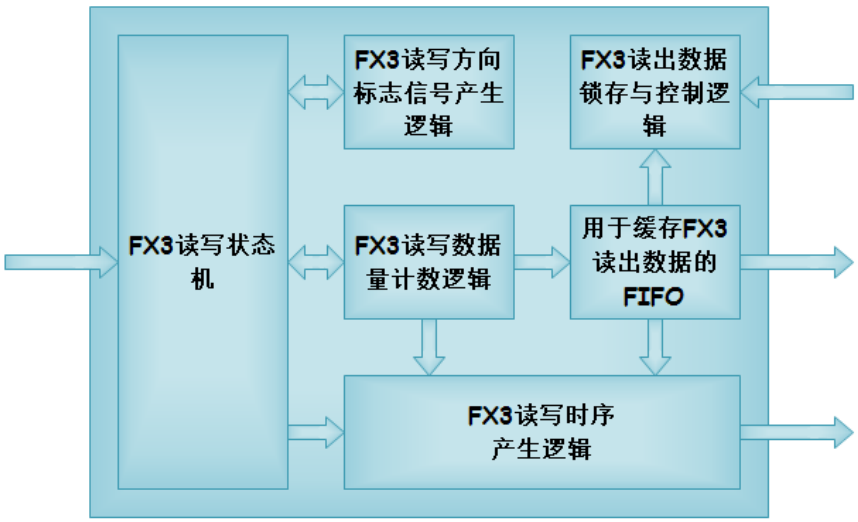


图 USB 读写模块功能框图

FX3 读写状态机的状态迁移如图所示。上电状态为 FXS_REST ，随后就进入 FXS_IDLE 状态，首先等待 FX3 的 SlaveFIFO 有可读数据，然后进入 FXS_READ 状态读取数据，接着进入 FXS_RSOP 状态停留一个时钟周期，然后回到 FXS_IDLE 状态；此时判断 SlaveFIFO 是否可写入，若可以写入则进入 FXS_WRIT 状态写入刚刚读取的所有数据到 FX3 的 SlaveFIFO 中，接着进

入 FXS_WSOP 状态停留一个时钟周期，最后回到 FXS_IDLE 状态，如此反复。

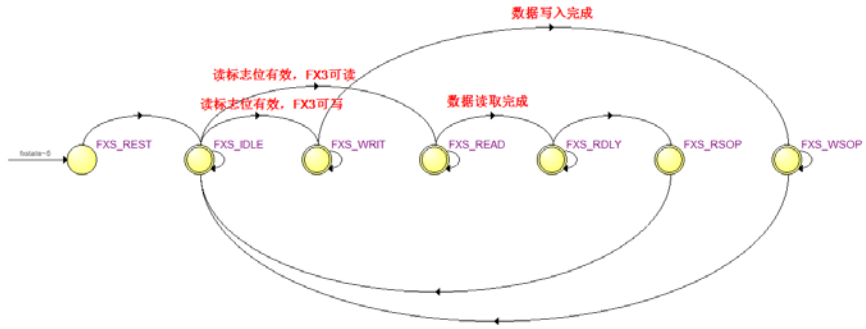


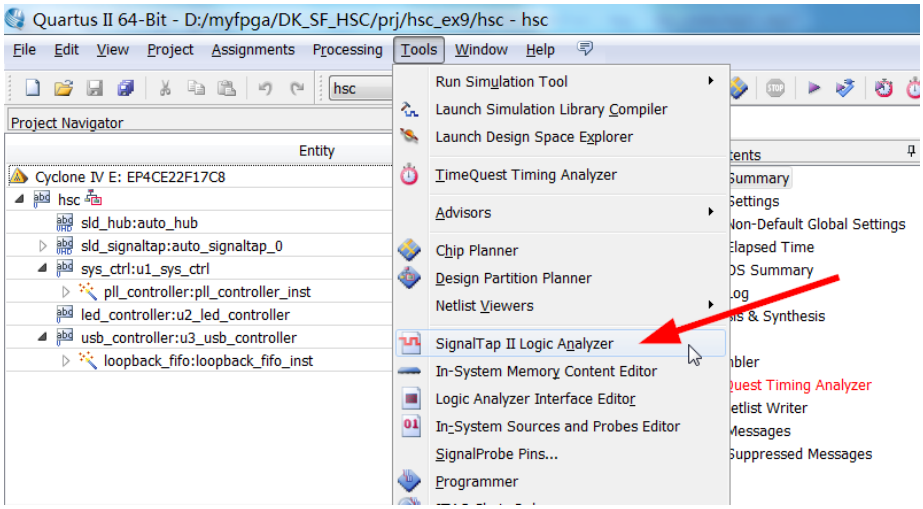
图 USB 读写模块状态转移图

FXS_IDLE 状态判断是读取还是写入 SlaveFIFO 有一个专门的读写方向控制的标志信号 fx3_dir。它为高电平时，FX3 读写状态机只能进行 FX3 的 SlaveFIFO 读操作；它为低电平时，FX3 读写状态机只能进行 FX3 的 SlaveFIFO 写操作。由于这个实验的功能是 USB 数据传输的“loopback”，即作为 USB 主机的 PC 端送出的数据，需要原样返回，因此，fx3_dir 信号的控制原则是：初始状态为 SlaveFIFO 的读操作（等待 PC 端发送数据），只有当读操作触发后，紧接着要进行 SlaveFIFO 的写操作（接收到的数据原样返回）。代码如下所示，这里状态 FXS_RSOP 为刚刚对 FX3 的 SlaveFIFO 读取完数据，FXS_WSOP 为刚刚对 FX3 的 SlaveFIFO 写入完数据。这时候切换更改 fx3_dir 的状态正符合我们的预期。

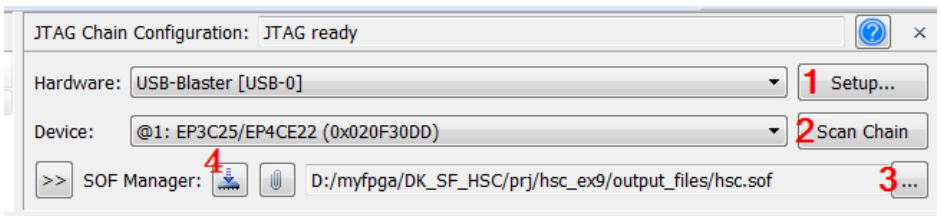
```
////////////////////////////////////////////////////////////////
reg fx3_dir;    //FX3 读写方向指示信号，1--read, 0--write

always @(posedge clk or negedge rst_n)
    if(!rst_n) fx3_dir <= 1'b1;     //read
    else if(fxstate == FXS_RSOP) fx3_dir <= 1'b0;       //write
    else if(fxstate == FXS_WSOP) fx3_dir <= 1'b1;       //read
```

4 SignalTap II 在线逻辑分析仪查看接口时序

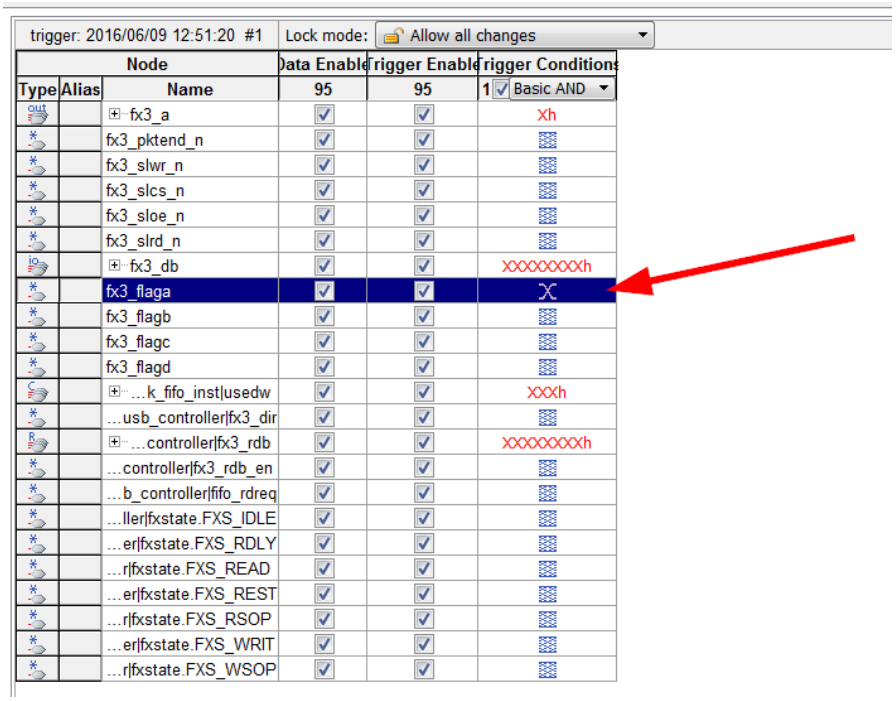
打开“...\prj\hsc_ex9”路径下的 Quartus II 工程（双击 hsc.qpf）。点击菜单“Tools → SignalTap II Logic Analyzer”。



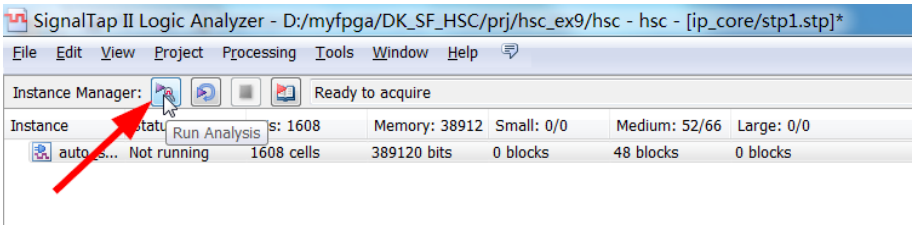
依次点击右侧 JTAG Chain Configuration 下的按钮“Step...”、“Scan Chain”和“...”。确保连接好 FPGA，最后点击第 4 步下载按钮，对 FPGA 进行下载配置。



设置触发条件为 fx3_flaga 信号的边沿。

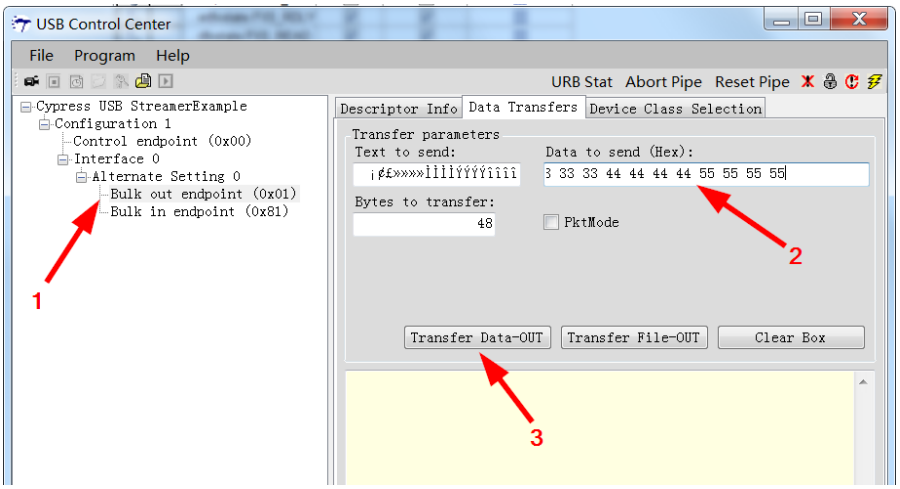


点击 Run Analysis 按钮进行单次触发。

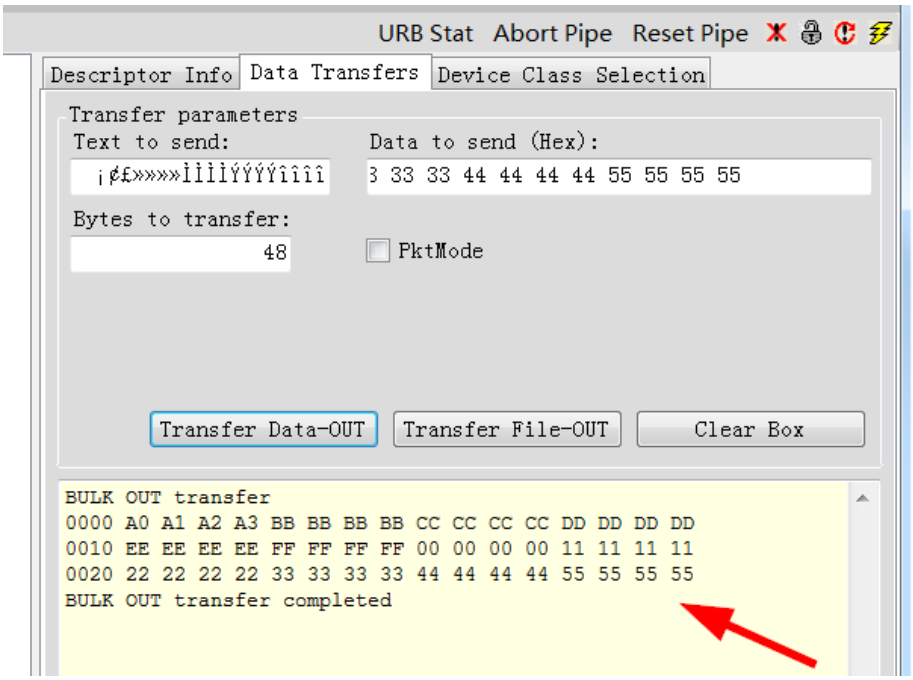


在 USB Control Center 的 Bulk out endpoint (0x01)中输入下面的 48Bytes 数据，点击“Transfer Data-OUT”进行传输。

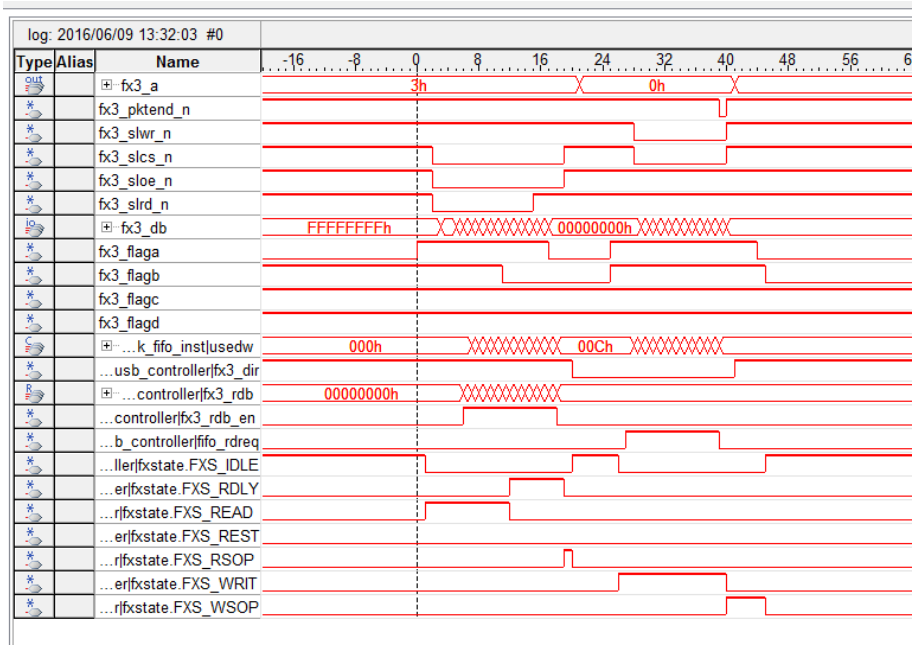
```
A0A1A2A3BBBBBBBBCCCCCCCCDDDDDDDDEEEEEEEEFFFFFFFF000000001111111
1222222223333333344444444555555555
```



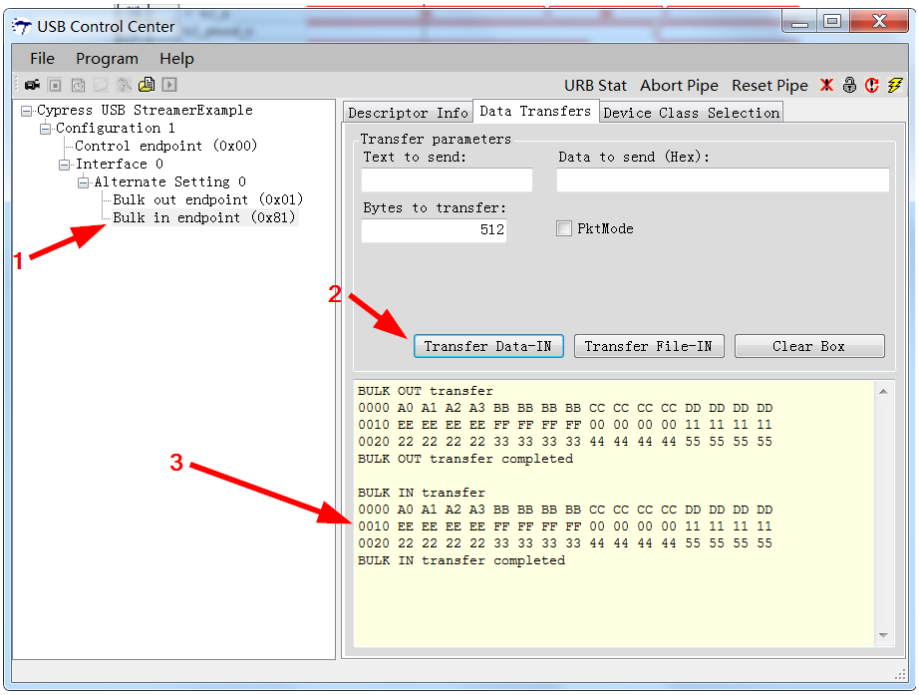
如图所示，在信息窗口中出现了“BULK OUT transfer completed”表示发送成功。



此时我们看到 SignalTap II 中也出现了一次波形触发。这个波形中记录了 PC 端传输给 FX3 一共 48Bytes 数据，然后 FPGA 从 FX3 的 SlaveFIFO 读取这些数据（地址 00），又原封不动的将这些数据传输给 FX3（地址 11）。



为了确认 FPGA 是否真的原封不动的将 48Bytes 数据传输回来了，在 USB Control Center 中，选择“Bulk in endpoint (0x81)”，然后点击 Transfer Data-IN 按钮，随后信息窗口出现了“BULK IN transfer completed”表示数据接收成功，同时大家可以比对收发的数据。比对之后，我们可以发现数据确实完全一致。说明本实例已经实现既定功能。


USB Control Center
File
Program
Help
URB Stat
Abort Pipe
Reset Pipe
Cypress USB StreamerExample
Configuration 1
Control endpoint (0x00)
Interface 0
Alternate Setting 0
Bulk out endpoint (0x01)
Bulk in endpoint (0x81)
1
Descriptor Info
Data Transfers
Device Class Selection
Transfer parameters
Text to send:
Data to send (Hex):
Bytes to transfer:
512
PktMode
2
Transfer Data-IN
Transfer File-IN
Clear Box
3
BULK OUT transfer
0000 A0 A1 A2 A3 BB BB BB BB CC CC CC CC DD DD DD DD
0010 EE EE EE EE FF FF FF FF 00 00 00 00 11 11 11 11
0020 22 22 22 22 33 33 33 33 44 44 44 44 55 55 55 55
BULK OUT transfer completed
BULK IN transfer
0000 A0 A1 A2 A3 BB BB BB BB CC CC CC CC DD DD DD DD
0010 EE EE EE EE FF FF FF FF 00 00 00 00 11 11 11 11
0020 22 22 22 22 33 33 33 33 44 44 44 44 55 55 55 55
BULK IN transfer completed