

# Altera FPGA 全速漂移 开发指南

## 基于 FPGA-FX3 SlaveFIFO 接口的

### StreamOUT 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](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](mailto: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
2016-07-26	V1.00	创建。

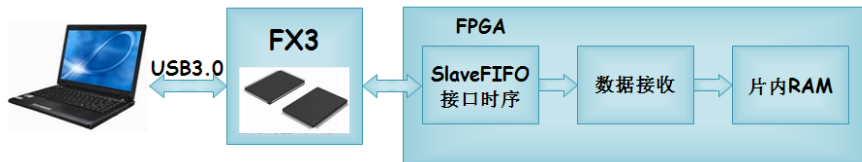


## 目录

Altera FPGA 全速漂移 开发指南 .....	1
基于 FPGA-FX3 SlaveFIFO 接口的 .....	1
StreamOUT 实例 .....	1
1 StreamOUT 功能概述 .....	3
2 Firmware 下载 .....	3
3 FPGA 代码解析.....	4
4 SignalTap II 在线逻辑分析仪查看接口时序 .....	7
5 Streamer 中查看数据吞吐量 .....	13

# 1 StreamOUT 功能概述

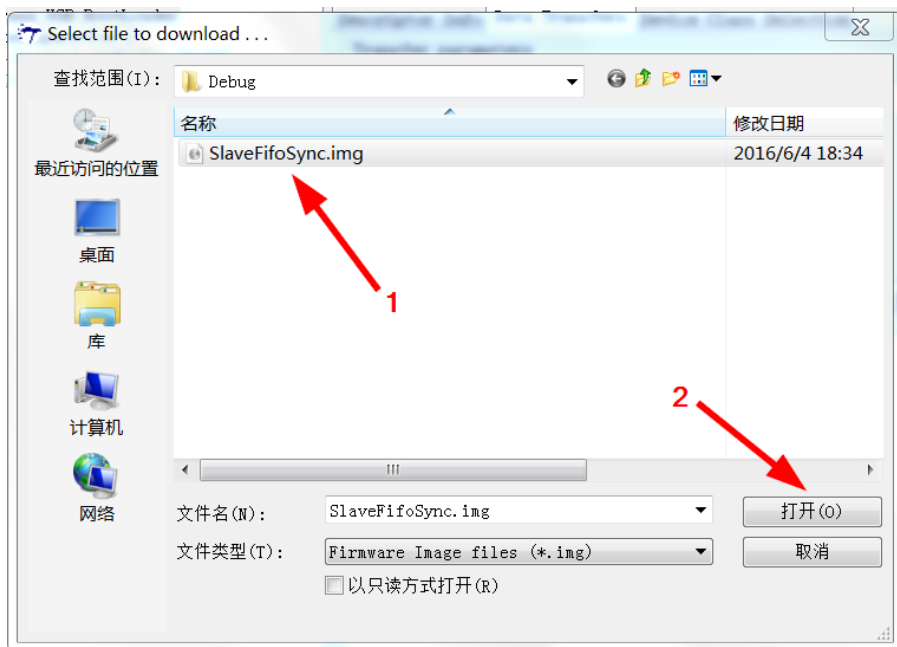
如图所示，本工程主要是 PC 端发送批量数据到 FX3，FX3 通过指示位 flaga 告知 FPGA 有数据待读取，FPGA 端便通过 SlaveFIFO 接口读取 PC 端发送过来的数据缓存到 FPGA 内部的 FIFO 中。整个数据的收发过程，我们在 FPGA 内部可以通过在线逻辑分析仪 SignalTap II 抓取 SlaveFIFO 接口的所有信号进行查看。



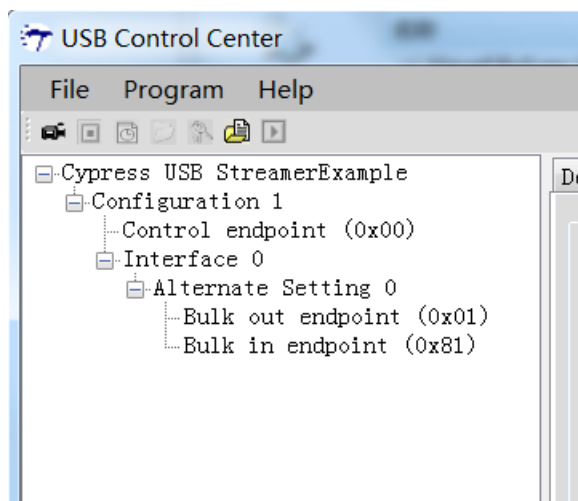
## 2 Firmware 下载

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。

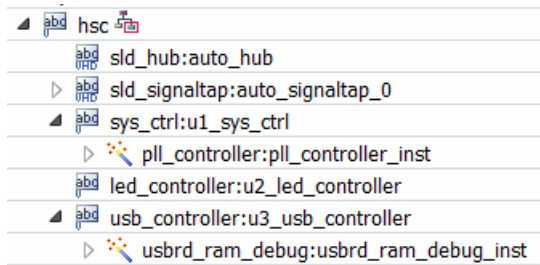


下载完成后，USB Control Center 如图所示。



### 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 是否有可以读取的数据，并读取这些数据，然后写到 FPGA 片内 RAM 中。
	usbrd_ram_debug.v 该 RAM 存储当前读取到 FX3 的最新数据帧，可以在 Quartus II 的 In-System Memory Content Editor 中查看这个 RAM 所存储的数据。
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 读写状态机一旦检测到 FX3 的 SlaveFIFO 有可读取的数据，就进入 FX3 数据读取的状态，读取 SlaveFIFO 中所有的数据，缓存到片内 RAM 中。

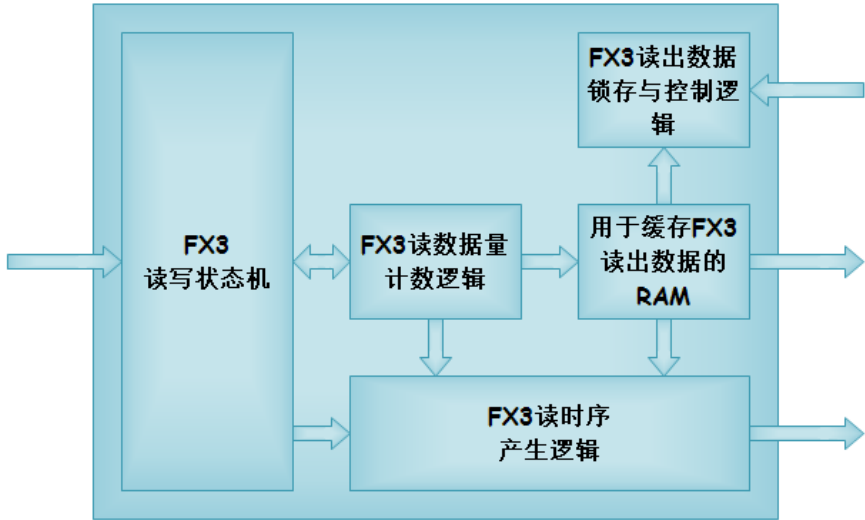


图 USB 读写模块功能框图

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，判断 SlaveFIFO 是否有可读取数据，若有则进入 FXS\_READ 状态读出 FX3 的 SlaveFIFO 中所有的数据，接着进入 FXS\_RSOP 状态停留一个时钟周期，最后回到 FXS\_IDLE 状态，如此反复。

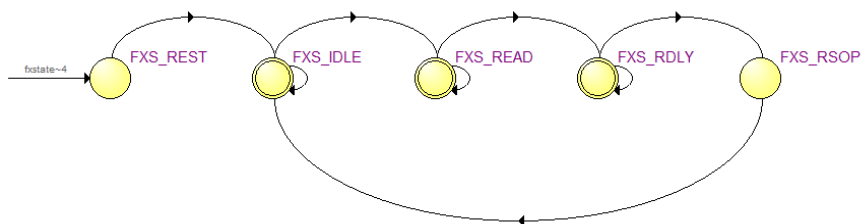
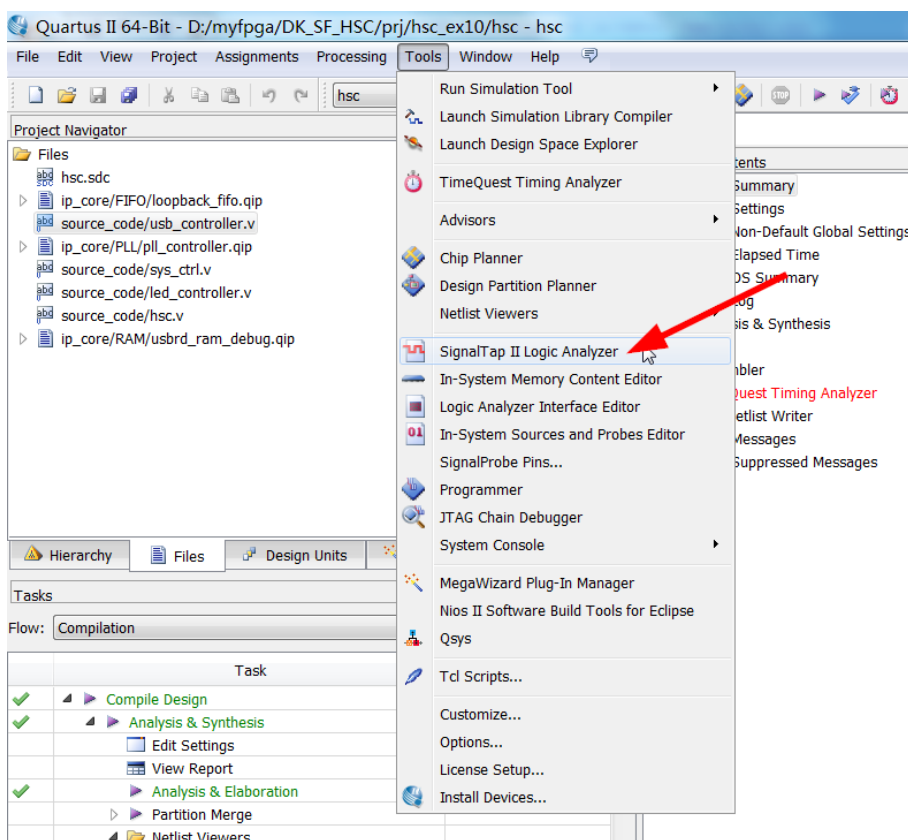


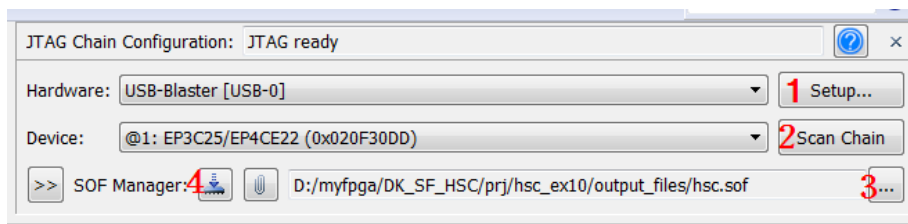
图 USB 读写模块状态转移图

## 4 SignalTap II 在线逻辑分析仪查看接口时序

打开 “...\prj\hsc\_ex10” 路径下的 Quartus II 工程（双击 hsc.qpf）。点击菜单 “Tools → SignalTap II Logic Analyzer”。



如图所示，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。

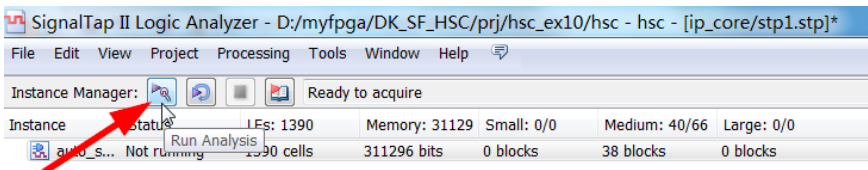


如图所示，设置触发条件为 fx3\_flaga 信号的边沿。

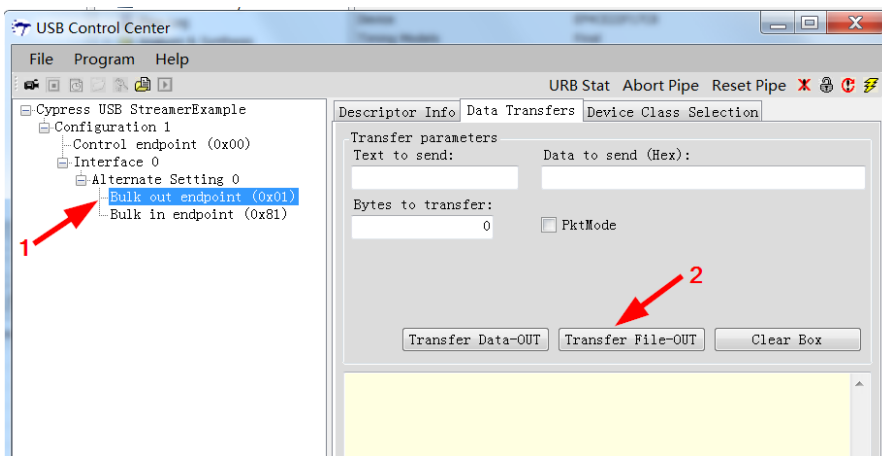


trigger: 2016/06/07 16:34:05 #1		Lock mode:  Allow all changes			
Type	Alias	Node Name	Data Enable 76	Trigger Enable 76	Trigger Conditions 1 Basic AND
		fx3_a	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Xh
		fx3_pktend_n	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		fx3_slwr_n	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		fx3_slcs_n	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		fx3_sloe_n	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		fx3_slrd_n	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		fx3_db	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	xxxxxxxxh
		fx3_flaga	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		fx3_flagb	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		fx3_flagc	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		fx3_flagd	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		...controller fx3_rdb	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	xxxxxxxxh
		...controller fx3_rdb_en	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

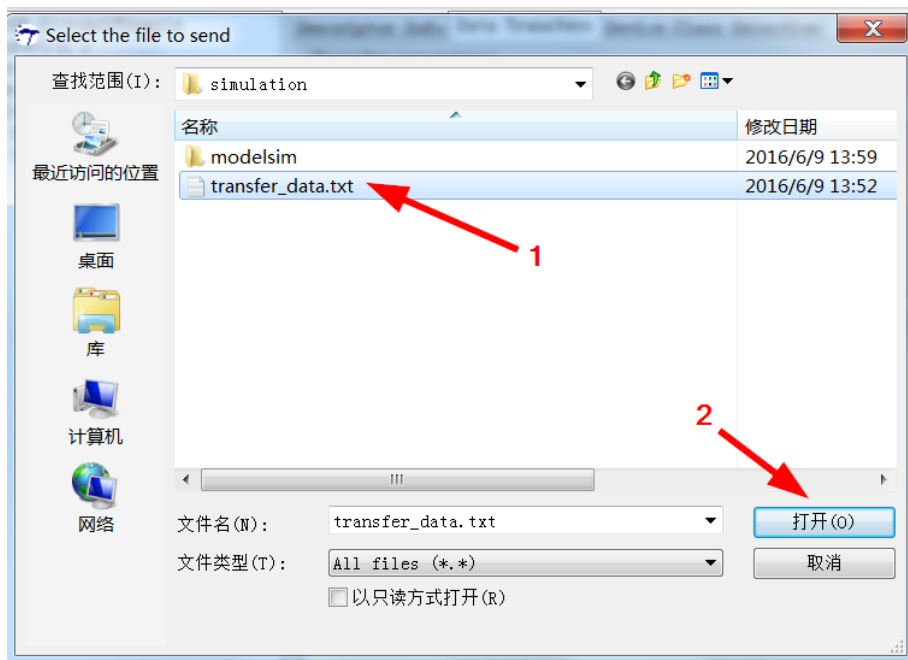
如图所示，选中 Instance 下面的选项，然后点击 Run Analysis 按钮进行单次触发。



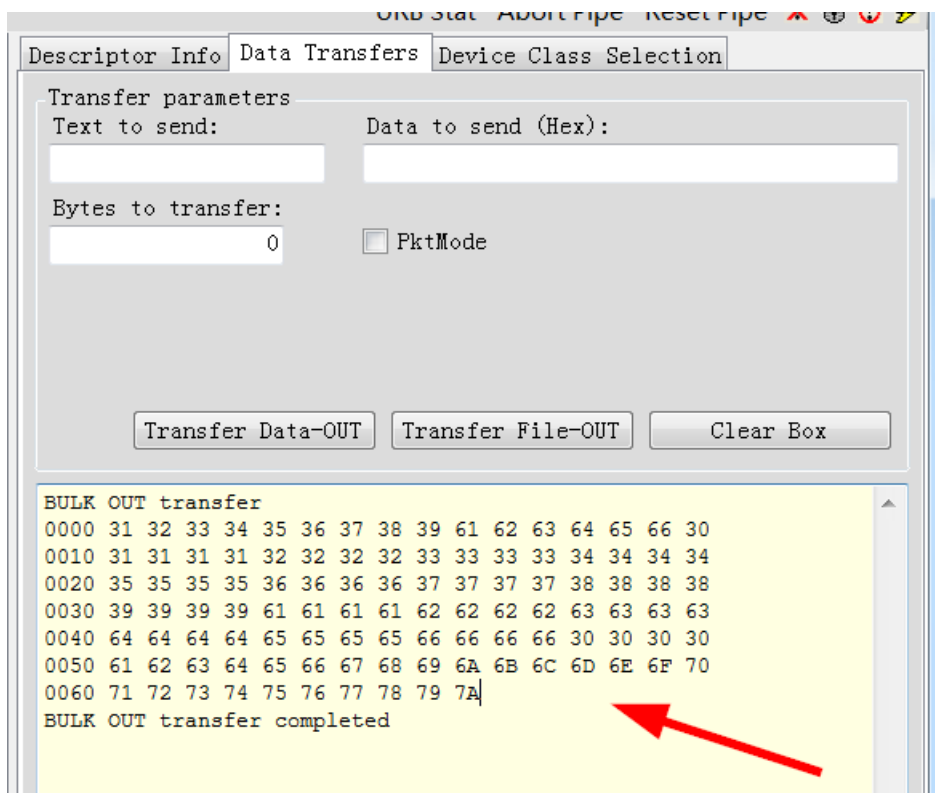
如图所示，在 USB Control Center 的 Bulk out endpoint (0x01)中点击“Transfer File-OUT”。



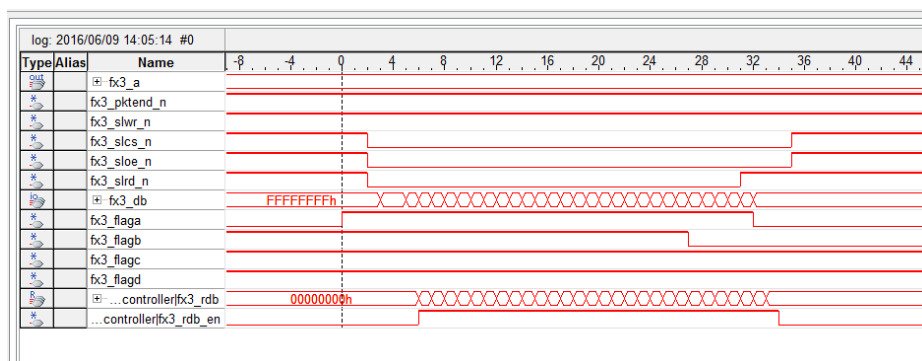
如图所示，在路径“...\prj\hsc\_ex10\simulation”下选中 transfer\_data.txt 文件。



在信息窗口中，如图所示，可以看到“BULK OUT transfer completed”的发送成功的提示信息。

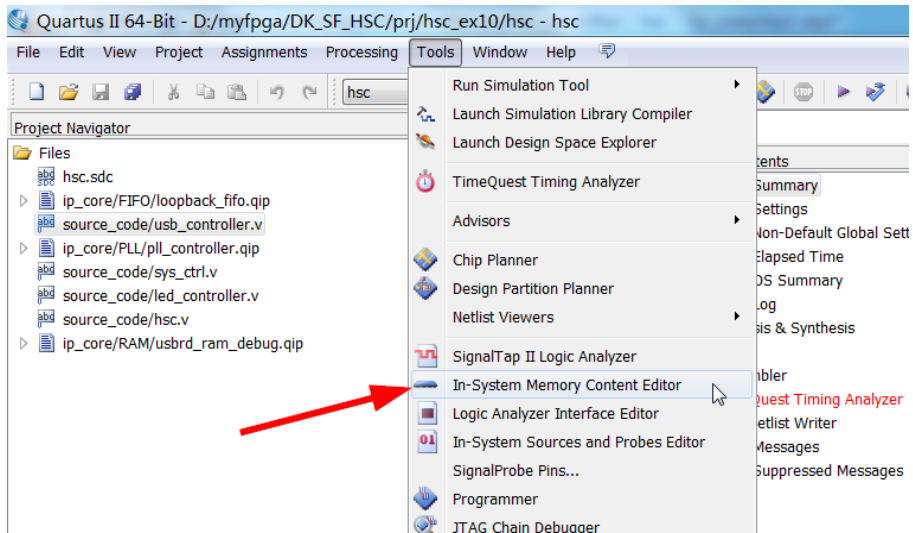


如图所示，此时我们看到 SignalTap II 中也出现了一次波形触发。这个波形中记录了 PC 端传输给 FX3 一大串的数据，和上一个实验不同，这个实验中 FPGA 从 FX3 的 SlaveFIFO 读取这些数据后，不再将这些数据传输给 FX3，而是将这些数据缓存到 FPGA 的片内 RAM 中。我们通过 FPGA 片内 RAM 可以观察这些数据是否和 USB Control Center 发送过来的一致。

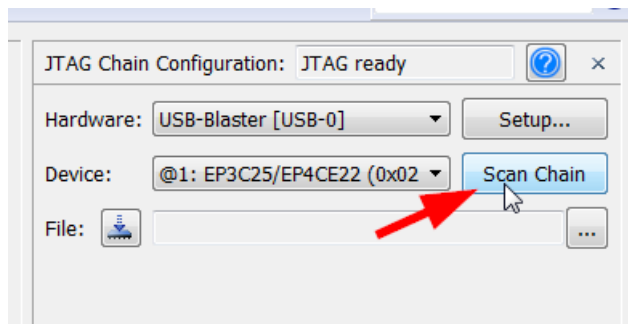


如图所示，Quartus II 中单击菜单 “Tools → In-System Memory Content

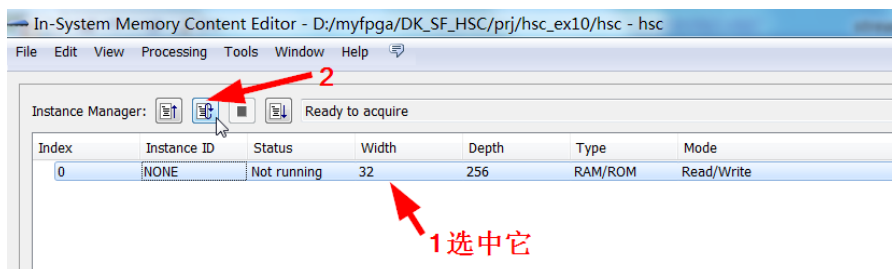
Editor”。



由于已经在 SignalTap II 中完成了 FPGA 的下载配置，所有这里不需要重复。如图所示，只需要点击 Scan Chain 建立连接。



接着如图所示，点击运行即可。



如图所示，我们可以看到接收到的数据，和之前发送的完全一致。大家可以更改 transfer\_data.txt 文件下的数据，重新发送看看，这里的数据会

相应的发生变化。

Instance 0: NONE																											
000000	31	32	33	34	35	36	37	38	39	61	62	63	64	65	66	30	123456789abcdef0										
000004	31	31	31	31	32	32	32	32	33	33	33	33	34	34	34	34	1111222233334444										
000008	35	35	35	35	36	36	36	36	37	37	37	37	38	38	38	38	5555666677778888										
00000c	39	39	39	39	61	61	61	61	62	62	62	62	63	63	63	63	9999aaaabbbbcccc										
000010	64	64	64	64	65	65	65	65	66	66	66	66	30	30	30	30	ddddeeeeffff0000										
000014	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	abcdefghijklmnp										
000018	71	72	73	74	75	76	77	78	79	7A	00	00	00	00	00	00	qrstuvwxyz.....										
00001c	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....										
000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....										
000024	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....										
000028	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....										
00002c	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00											

## 5 Streamer 中查看数据吞吐量

我们也可以打开 PC 端的 Streamer，如图所示配置后进行测试。在 USB2.0 下，数据吞吐量也可以达到 25400KB（实际吞吐量不同 PC 配置差别较大，有限较强的 PC 可以达到 40MB 以上），蛮不错的数据吞吐量了。

Cypress USB StreamerExample

File Help

Endpoint . . . . . ?-0, 512 Byte Bulk out endpoint (0x01)

Packets per Xfer 8 Successes 26752

Xfers to Queue 128 Failures 0

Stop

Throughput (KB/s)

25400