

Fabric Debugger 用户手册

(Version 1.1)

深圳市紫光同创电子有限公司

版权所有 侵权必究

文档版本修订记录

版本号	发布日期	修订记录
V1.0	2022.07.12	初始版本
V1.1	2023.04.24	支持 DVIO

目录

1 软件简介.....	8
2 下载电缆介绍.....	8
2.1 USB 下载电缆.....	8
2.2 并口下载电缆:	9
2.3 下载电缆的指示灯状态.....	10
3 软件使用入门.....	11
3.1 连接 JTAG SERVER.....	11
3.2 选择 CABLE 参数.....	11
3.3 设置 CABLE 参数.....	12
3.4 扫描器件.....	13
3.5 下载位流文件.....	14
4 FABRIC DEBUGGER 软件说明.....	16
4.1 用户界面介绍.....	16
4.2 菜单基本操作说明.....	17
4.3 FILE 菜单.....	17
4.2.1 View 菜单.....	18
4.2.2 JTAG Chain 菜单.....	18
4.2.3 Device 菜单.....	18
4.2.4 Settings 菜单.....	20
4.2.5 Help 菜单.....	20
4.3 工具栏基本操作说明.....	21
4.3.1 debugger core 调试模式工具栏.....	21
4.3.2 ADC 调试模式菜单栏.....	22
4.3.3 iScan 菜单栏.....	22
4.3.4 DVIO 菜单栏.....	22
5 DEBUGGER 功能介绍.....	24
5.1 器件栏.....	24
5.2 信号栏.....	26
5.2.1 debugger core 信号栏.....	26
5.2.2 adc 信号栏.....	29
5.3 TRIGGER SETUP 对话框.....	29
5.3.1 Match Funcitons.....	29
5.3.2 Trigger Condition.....	31
5.3.3 Capture Settings.....	35
5.4 WAVEFORM 对话框.....	37
5.4.1 Signal Browser.....	37
5.4.2 Waveform Display.....	38
5.5 LISTING 对话框.....	41
5.6 BUS PLOT 对话框.....	43
5.7 ADC 调试对话框.....	44
5.8 ADC 配置对话框.....	45
5.9 ADC 寄存器.....	46
5.9.1 Logos 系列.....	47
a) 配置寄存器.....	47

b) 模式控制寄存器.....	50
c) 扫描控制寄存器.....	53
d) 温度检测控制寄存器.....	56
e) 状态寄存器.....	56
5.9.2 Logos2、Titan2 系列.....	57
a) 控制寄存器.....	57
b) 状态寄存器.....	66
5.10 波形搜索.....	67
5.11 波形分析.....	69
5.12 系统设置菜单.....	70
5.13 版本匹配功能.....	72
5.14 多核同时调试功能.....	72
5.15 ISCAN.....	73
5.15.1 总体介绍.....	73
5.15.2 功能描述.....	73
5.16 DVIO 调试.....	81
5.16.1 总体介绍.....	81
5.16.2 功能描述.....	82
6 TCL 命令.....	86
6.1 DEBUGGER TCL COMMAND 简介.....	86
6.2 DEBUGGER TCL COMMAND 详细介绍.....	87
6.2.1 通用命令.....	87
1) source.....	87
2) dbg_help.....	88
3) dbg_close.....	88
4) dbg_connect.....	88
5) dbg_disconnect.....	88
6) dbg_delay.....	89
7) dbg_scan_chain.....	89
8) dbg_program.....	89
9) dbg_open_project.....	89
10) dbg_save_project.....	90
6.2.2 Debug core 相关命令.....	90
1) dbg_run.....	90
2) dbg_trig_immd.....	90
3) dbg_stop.....	90
4) dbg_read_user_code.....	91
5) dbg_read_device_id.....	91
6) dbg_read_status_register.....	91
7) dbg_read_instruction_register.....	91
8) dbg_power_init.....	92
9) dbg_get_cur_core.....	92

10) dbg_set_cur_core.....	92
11) dbg_get_buses.....	92
12) dbg_add_bus.....	93
13) dbg_add_to_bus.....	93
14) dbg_del_bus.....	93
15) dbg_reverse_bus.....	93
16) dbg_set_bus_radix.....	94
17) dbg_set_unit.....	94
18) dbg_set_condition.....	95
19) dbg_set_capture.....	96
20) dbg_set_storage.....	97
21) dbg_list.....	97
22) dbg_import_fic.....	98
23) dbg_export_fic.....	98
24) dbg_add_all_to_waveform.....	98
25) dbg_add_all_to_listing.....	99
26) dbg_clear_all_to_waveform.....	99
27) dbg_clear_all_to_listing.....	99
28) dbg_waveform_zoom.....	99
29) dbg_read_adc_register.....	100
30) dbg_write_adc_register.....	100
 <i>6.2.3 iScan 相关命令.....</i>	 100
1) dbg_write_hsst_register.....	101
2) dbg_read_hsst_register.....	101
3) dbg_hsst_update.....	101
4) dbg_hsst_start.....	101
5) dbg_hsst_stop.....	102
 <i>6.2.4 DVIO 相关命令.....</i>	 102
1) dbg_select_current_dvio.....	102
2) dbg_show_current_dvio.....	102
3) dbg_dvio_refresh_input.....	102
4) dbg_dvio_refresh_output.....	102
5) dbg_dvio_reset_output.....	103
6) dbg_dvio_set_output.....	103
7) dbg_dvio_commit_output.....	103
8) dbg_dvio_print.....	103
 7 常见问题.....	 104
8 免责声明.....	106

图目录

图 2- 1 USB Cable I 下载电缆连接示意图.....	8
图 2- 2 USB Cable II 下载电缆连接示意图.....	9
图 2- 3 并口下载电缆连接示意图.....	9
图 3- 1 连接服务端.....	11
图 3- 2 选择 Cable 参数.....	12
图 3- 3 设置 Cable 参数.....	13
图 3- 4 扫描器件.....	14
图 3- 5 Debugger 加载文件图.....	14
图 3- 6 Debugger 成功下载文件到芯片.....	15
图 4- 1 Debugger 主界面.....	16
图 4- 2 File 菜单.....	17
图 4- 3 View 菜单.....	18
图 4- 4 JATG Chain 菜单.....	18
图 4- 5 device 菜单.....	18
图 4- 6 setting 菜单.....	20
图 4- 7 help 菜单.....	20
图 4- 8 debugger core 调试工具栏.....	21
图 4- 9 ADC 调试工具栏.....	22
图 4- 10 iScan 调试工具栏.....	22
图 4-11 DVIO 调试工具栏.....	22
图 5- 1 器件栏信息显示树形图.....	24

图 5- 2 debugger core 右键菜单.....	25
图 5- 3 Signal Tree-Data Ports 右键菜单选项.....	26
图 5- 4 Signal Tree 信号右键菜单选项.....	27
图 5- 5 总线的右键菜单选项.....	28
图 5- 6 ADC 模块通道树形图.....	29
图 5- 7 Match Funciton 配置.....	30
图 5- 8 Match Counter 配置对话框.....	31
图 5- 9 Trigger Condition.....	32
图 5- 10 配置 Trigger Condition 布尔表达式.....	33
图 5- 11 配置 Trigger Condition 顺序触发表达式.....	34
图 5- 12 Capture Settings.....	35
图 5- 13 Storage Qualification Condition 选择 All Data.....	36
图 5- 14 Storage Qualification Condition 选择布尔表达式.....	37
图 5- 15 放大指定区域的波形.....	39
图 5- 16 Go To Waveform.....	40
图 5- 17 Listing 对话框.....	42
图 5- 18 Bus Plot 对话框： data vs time.....	43
图 5- 19 Bus Plot 对话框 : data vs data.....	44
图 5- 20 ADC 对话框.....	45
图 5- 21 Edit Register 配置界面.....	46
图 5- 22 配置 bit 分配.....	48
图 5- 23 模式控制寄存器 bit 分配.....	50

图 5- 24 扫描控制寄存器 bit 分配	53
图 5- 25 扫描控制寄存器 bit 分配	56
图 5- 26 control registers 组成&对应功能	59
图 5- 27 波形搜索界面	68
图 5- 28 波形测量模式设置	70
图 5- 29 硬件采样率设置界面	70
图 5- 30 waveform 测量方式设置界面	71
图 5- 31 波形触发设置界面	72
图 5- 32 iScan 主界面	74
图 5- 33 Hsst Links 右键菜单	74
图 5- 34 Create Links 对话框	75
图 5- 35 Hsst Links 窗口	76
图 5- 36 Create Scan 对话框	77
图 5- 37 Hsst Scans 窗口	78
图 5- 38 Hsst Scan 右键菜单	79
图 5- 39 Scan Plot 界面	80
图 5- 40 DVIO 主界面	82
图 5- 41 器件栏 DVIO 右键菜单	82
图 5- 42 ProbIn 行右键菜单	84
图 5- 43 ProbeOut 行右键菜单	85

1 软件简介

Fabric Debugger 是一款界面化的 FPGA 芯片调试工具，能够针对公司目前以及未来的 FPGA 芯片进行在线调试。本软件直接与 JtagHub、DebugCore 交互，能够实时的配置目标 FPGA、设置触发条件并且观测结果。

a.支持的功能:

支持 FPGA 的 ID、USER CODE、状态寄存器及指令寄存器的读取功能、逻辑位流下载、触发条件配置及信号捕捉、回读 trigger setup 寄存器值、捕获上电初始化数据、波形分析、读取 ADC 数据等。

b.支持的下载电缆

支持电缆：Pango USB Cable I, Pango USB Cable II, Pango Parallel Cable I

2 下载电缆介绍

2.1 USB 下载电缆

使用 USB 下载电缆需要安装对应的驱动，安装步骤参见文档 Pango Design Suite Windows Install Guide 中软件安装部分的介绍。驱动安装成功后插入 USB 下载电缆，如果在设备管理器中多了新的设备“Programming cables”，则说明下载电缆可以正常使用了。

目前支持 Pango USB Cable I 和 Pango USB Cable II，USB 下载电缆负责把 PC 机的 USB 信号转换为器件所需要的 JTAG 信号或 SPI 信号。下载线的物理连接以下载线包装盒内提供的说明文档为准，其连接方式如下图所示：

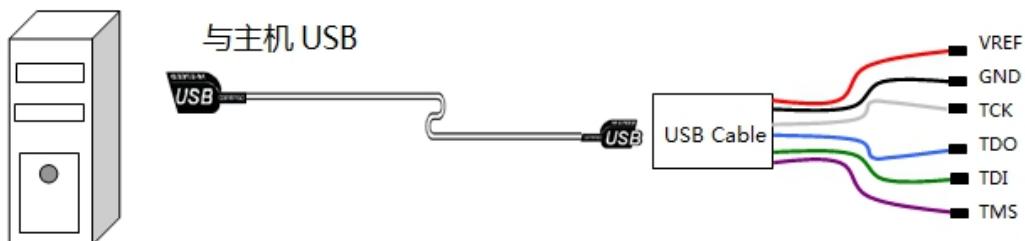


图 2-1 USB Cable I 下载电缆连接示意图

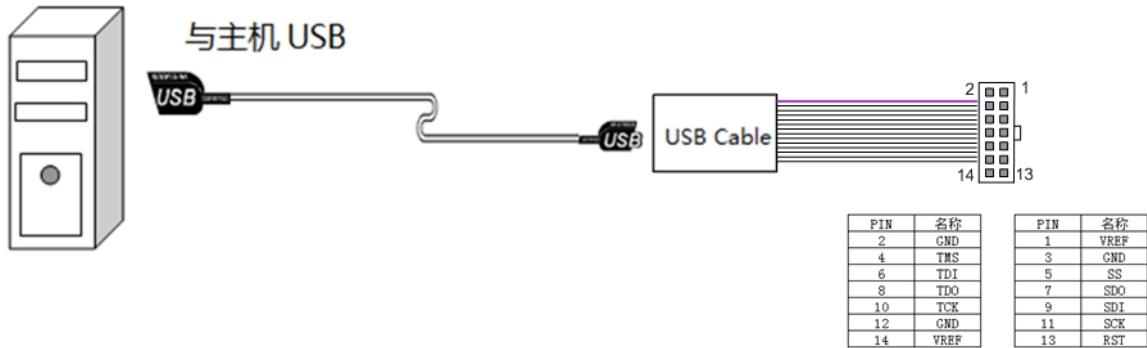


图 2-2 USB Cable II 下载电缆连接示意图

注：图中 14PIN 的连接头中，PIN1 的 VREF 给 JTAG 接口供电，PIN14 的 VREF 给 SPI 接口供电，两者可不同。

2.2 并口下载电缆

使用并口下载电缆，首先要确定 PC 机的主板上有集成的并口卡，并且在 BIOS 中设置启用，如果要使用 ECP 模式，还需要在 BIOS 中设置启用并口 ECP 模式。其次要安装并口驱动，安装步骤参见文档 Pango Design Suite Windows Install Guide 中软件安装部分的介绍。

目前支持 Pango Parallel Cable I，并口下载电缆负责把 PC 机的并行信号转换为器件所需要的 JTAG 信号或 SPI 信号。下载线的物理连接以下载线包装盒内提供的说明文档为准，其连接方式如下图所示：

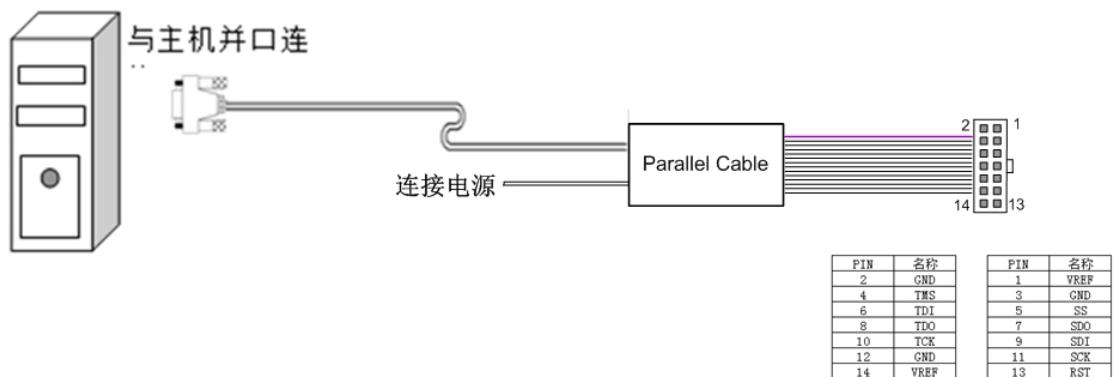


图 2-3 并口下载电缆连接示意图

注：图中 14PIN 的连接头中，PIN1 的 VREF 给 JTAG 接口供电，PIN14 的 VREF 给 SPI 接口供电，两者可不同。

2.3 下载电缆的指示灯状态

Pango USB Cable II 和 Pango Parallel Cable I 都有一个指示灯可以显示红色和绿色，Pango USB Cable I 无指示灯功能，定义每次对 FPGA 和 Flash 的读写都是一次操作，指示灯的显示规律如下：

- a) 初始状态红灯常亮，绿灯灭
- b) 操作进行中绿灯闪，红灯灭
- c) 操作结果出错红灯常亮，绿灯灭
- d) 操作结果正常绿灯常亮，红灯灭

3 软件使用入门

3.1 连接 Jtag Server

启动本软件后，点击工具栏按钮 ，弹出连接到 Cable 的向导，第一步是连接到服务端，本软件采用 TCP/IP 协议与服务端通信，所以需要指定 IP 地址和端口，IP 地址既可以指定为本机 IP 地址也可以指定为远端的 IP 地址。

当指定为本地 IP 地址时，直接点击连接即可；如果指定为远程 IP 地址，需要先在远程启动服务端，可以使用命令行方式启动，启动方法为：切换到可执行程序所在目录然后启动操作系统的 shell 终端，输入命令 `cdt_js -port 65420` 即可，更多使用参数可以输入命令 `cdt_js -help` 查看。

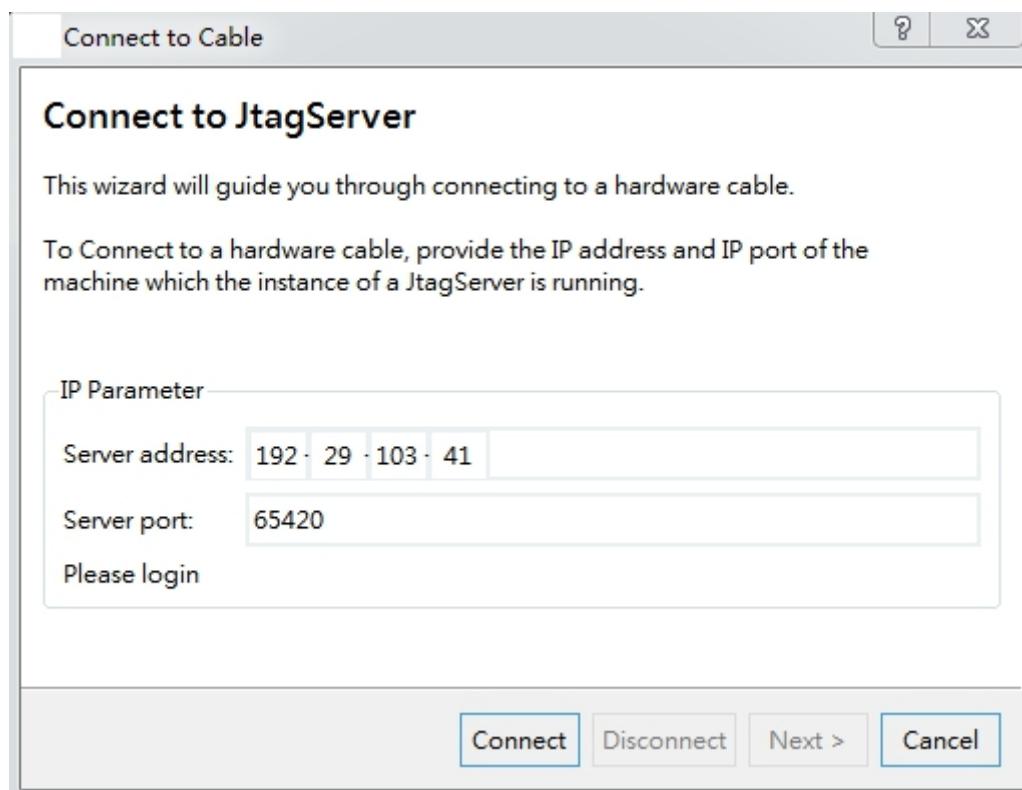


图 3- 1 连接服务端

3.2 选择 Cable 参数

连接到服务端成功后，会列出当前所有可用的 Cable，可以通过单击选中一个 Cable，还可以通过下拉菜单设置 TCK 频率，默认选择第一个 Cable 和第一个

频率。

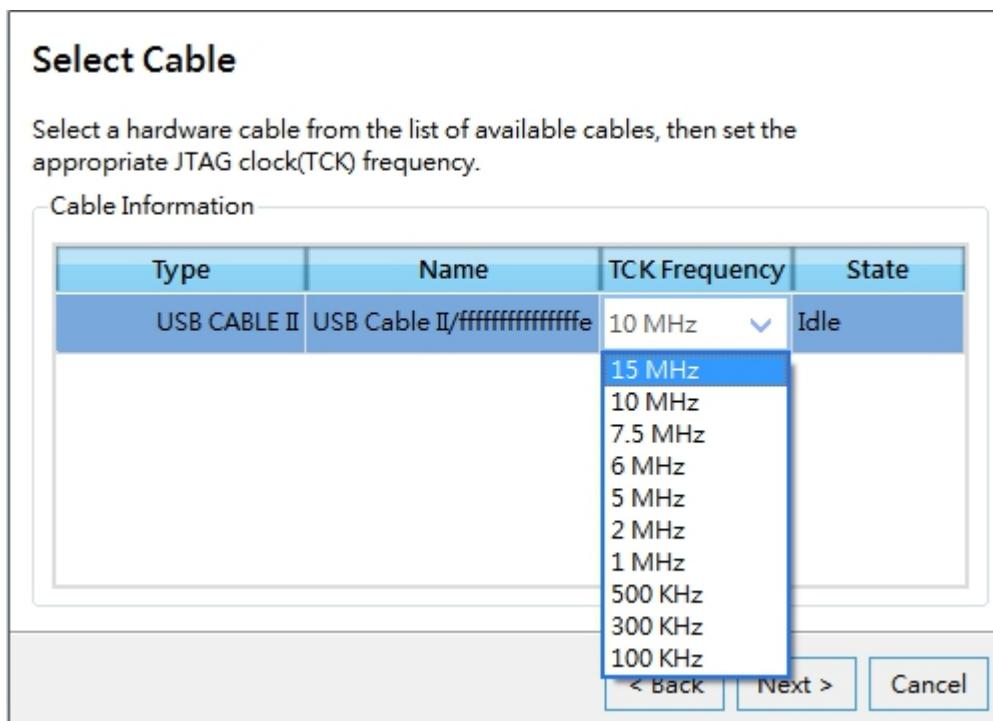


图 3- 2 选择 Cable 参数

3.3 设置 Cable 参数

点击 Next 跳到总结页面，该页面会列出服务端连接信息和 Cable 参数，此时 Cable 参数设置还未生效，点击 Back 可以回到上一步重新选择，点击 Cancel 可以取消本次设置，点击 Next 会将配置信息发送到服务端。



图 3-3 设置 Cable 参数

3.4 扫描器件

同时，软件支持在扫链时按照界面自动连接 server，扫描到器件如下图所示。

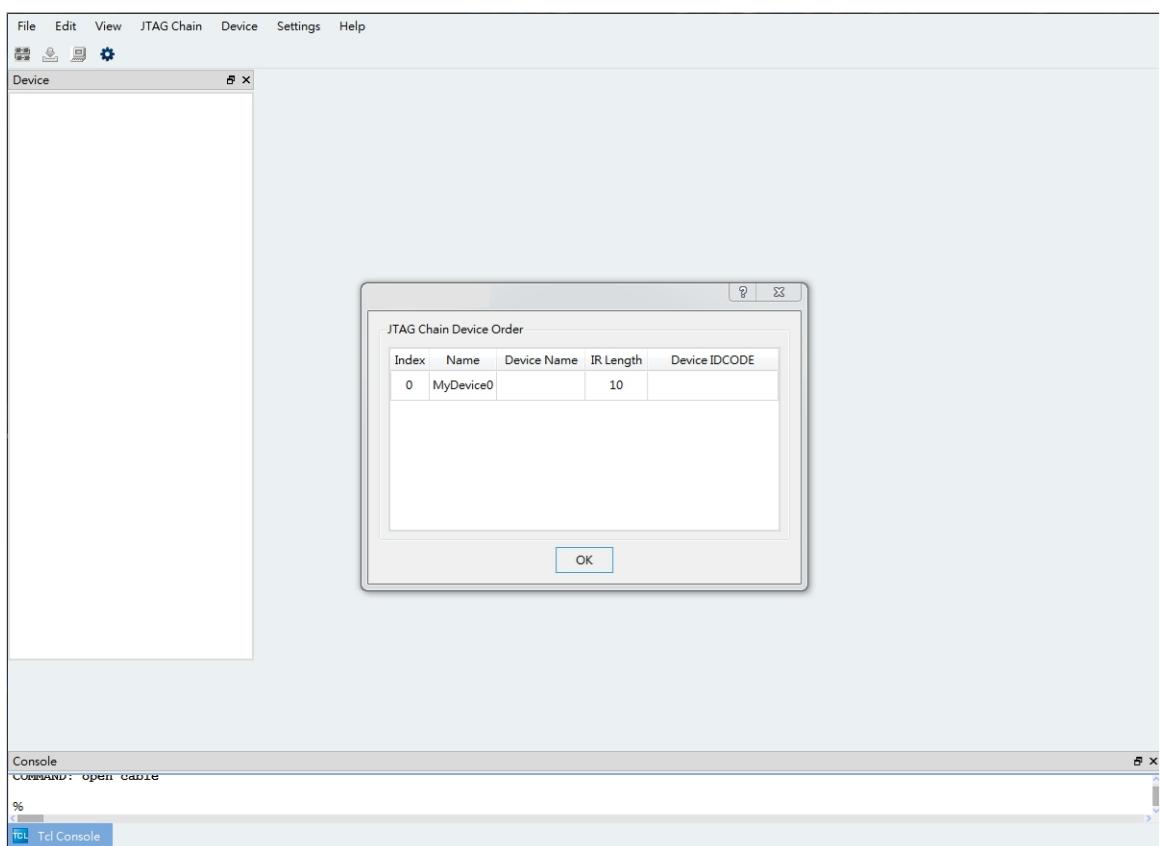


图 3-4 扫描器件

3.5 下载位流文件

自动扫描 JTAG 链，查询链上的器件，如查询成功后点击 Configure BitStream File 按钮或器件右键菜单打开位流配置界面，选择指定文件，同时可以通过手动或是自动的方式加载 fic 文件，自动加载 fic 方式会搜索位流文件所在目录以及上层目录，当存在与位流文件同名的 fic 文件或是_trs.fic 文件时，会在下载位流后，自动导入 fic 文件，位流加载操作界面如下图所示：

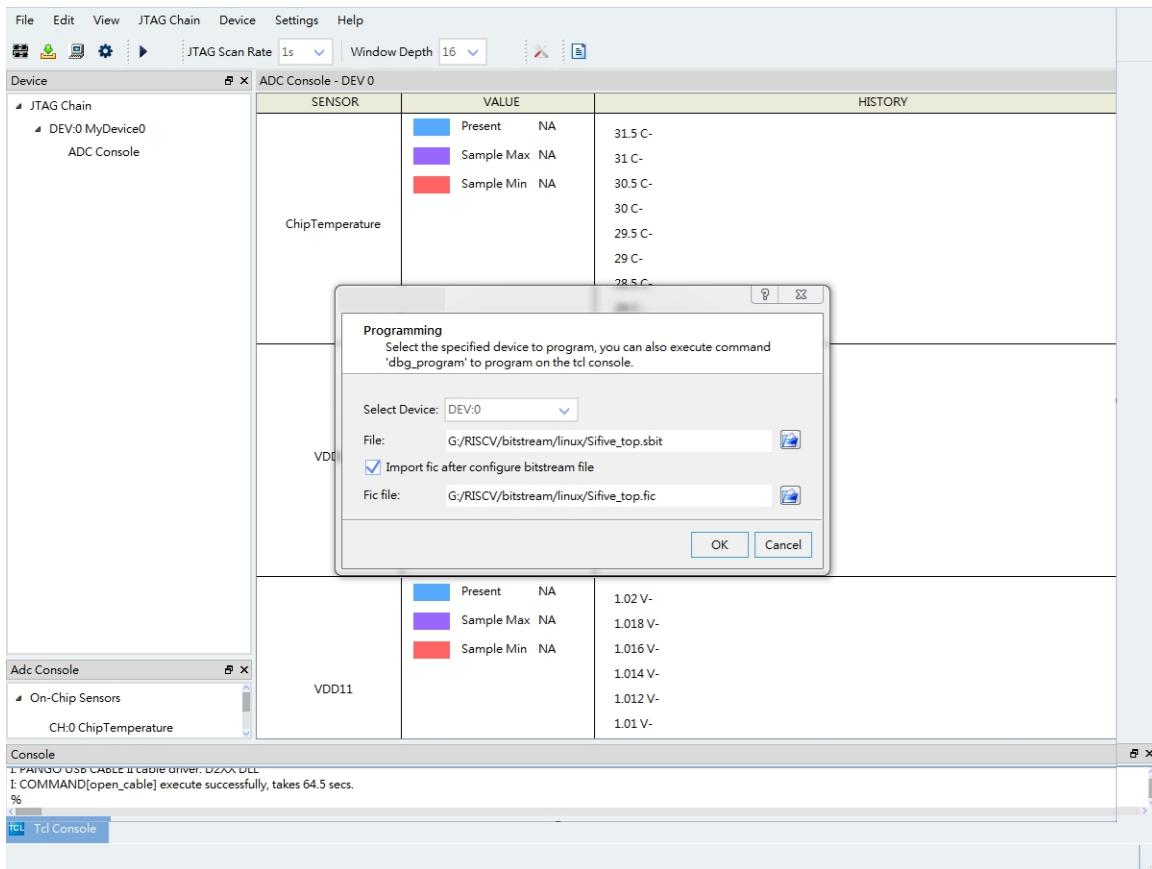


图 3-5 Debugger 加载文件图

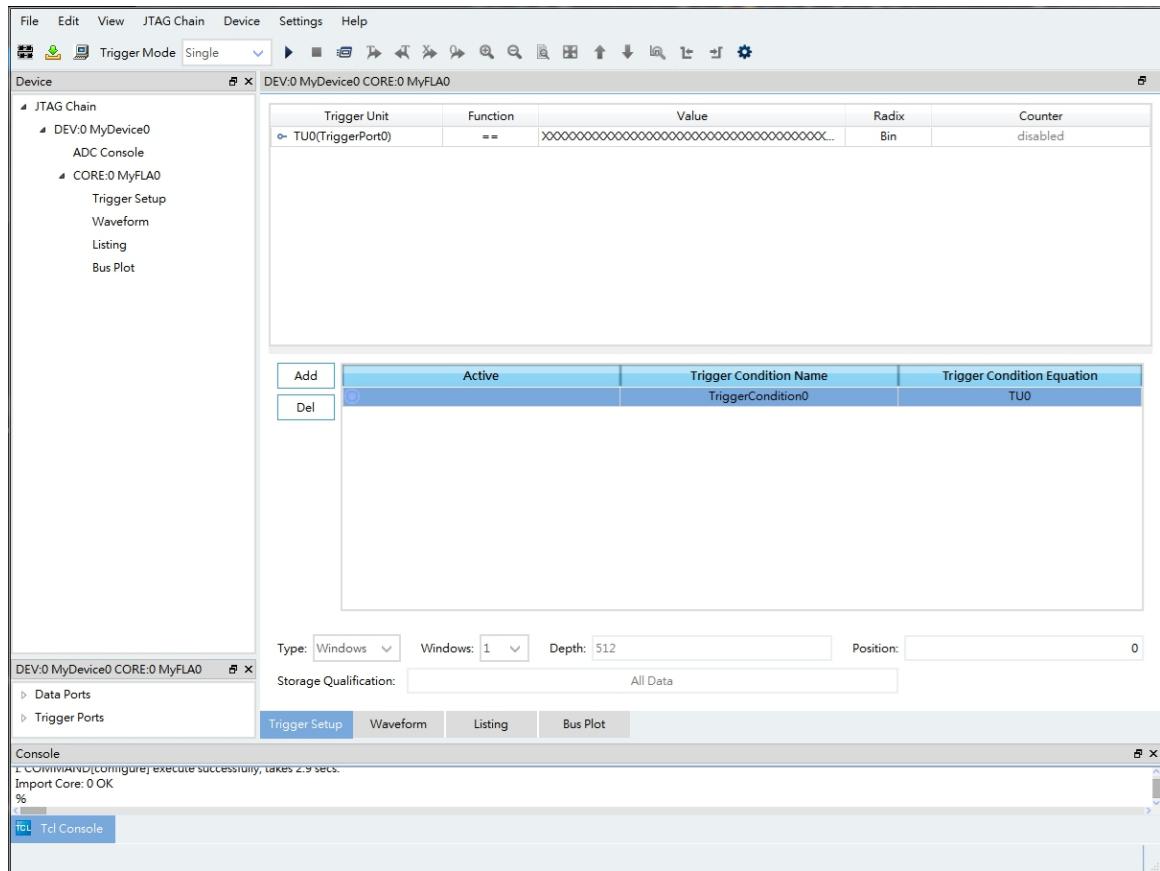


图 3- 6 Debugger 成功下载文件到芯片

4 Fabric Debugger 软件说明

4.1 用户界面介绍

启动 Fabric Debugger 软件，主界面如图

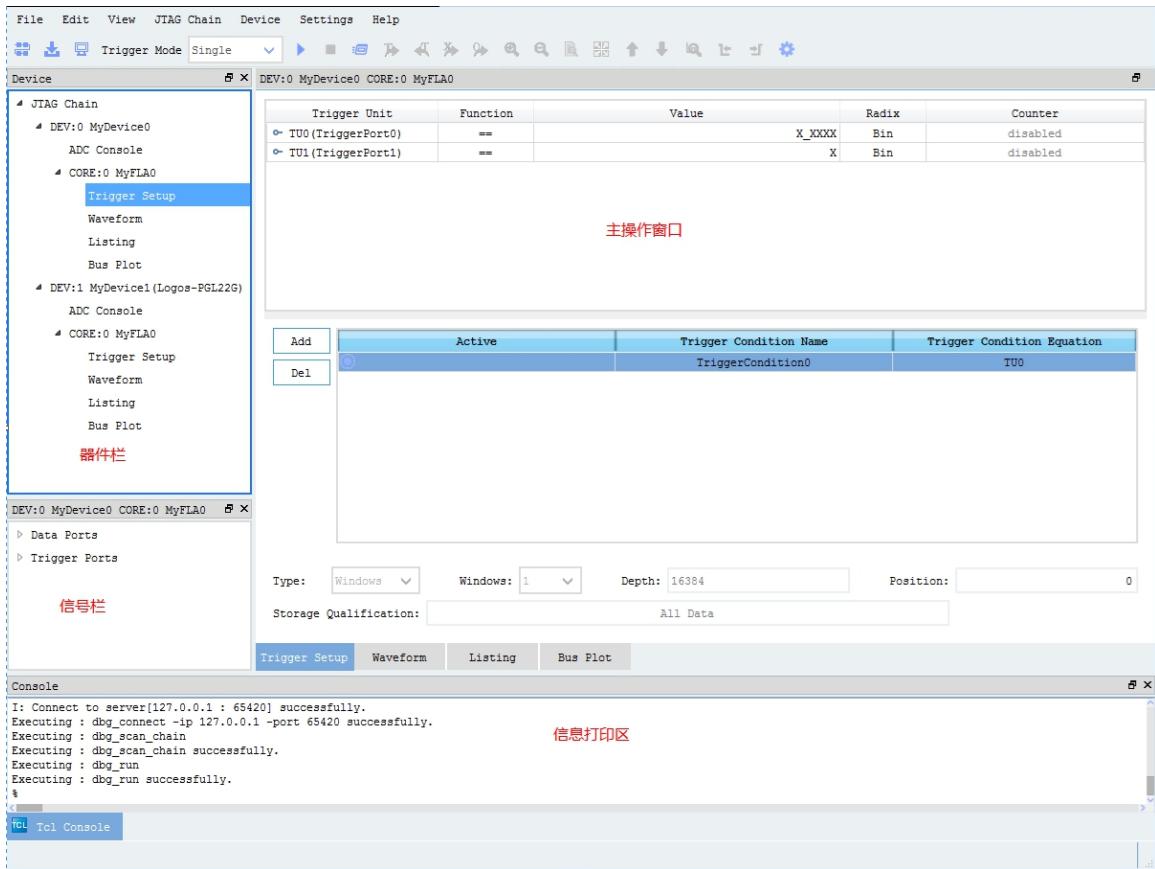


图 4- 1 Debugger 主界面

【器件栏】: 负责显示器件信息

【信号栏】: 负责显示信号、通道等信息

【主操作窗口】: 调试主窗口区

【信息打印区】: 负责软件信息输出

4.2 菜单基本操作说明

4.3 File 菜单

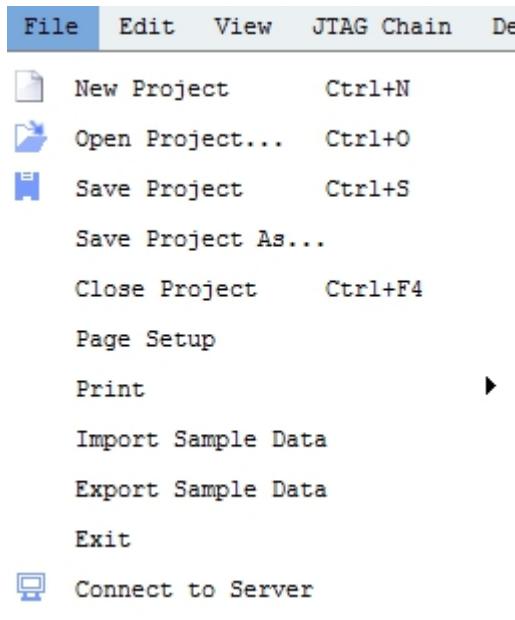


图 4-2 File 菜单

【New Project】: 负责为工程创建新的项目文件(会生成.dprj 项目文件.wdf 的波形文件)

【Open Project】: 打开项目文件

【Save Project】: 保存项目文件

【Save Project As】: 另存项目文件

【Close Project】: 关闭项目

【Connect to server】: 打开连接 server 对话框

【Import sample data】: 导入 fic 替换信号名称或 wdf 波形数据文件

【Export sample data】: 导出 ASCII, WDF, FIC, VCD, TXT 文件, 其中 TXT 文件可导出所有 Bus 的值或者指定界面 (Waveform, Listing 或 Bus Plot) 上的 Bus 的值。

【Exit】: 退出程序

4.2.1 View 菜单

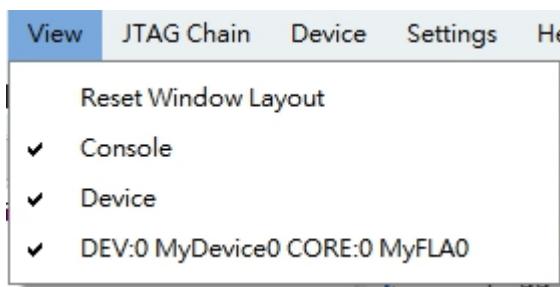


图 4- 3 View 菜单

【Reset window layout】: 将软件布局恢复为默认情况

【Console】 控制信息打印区的显示隐藏

【Device】 控制器件栏的显示隐藏

【Signal】 控制信号栏的显示隐藏

4.2.2 JTAG Chain 菜单



图 4- 4 JATG Chain 菜单

【JTAG Chain setup】: 显示器件链信息

4.2.3 Device 菜单

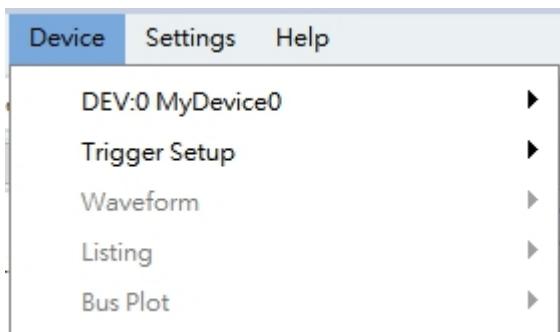


图 4- 5 device 菜单

Device 菜单会显示当前的器件链

【device】 右键显示 device 的操作菜单

【rename】: 重命名器件

【Delete Device】: 删除器件

【Configure Bitstream File】: 为器件配置位流

【Show IDCODE】: 打印器件的 IDCODE

【show USERCODE】: 打印器件的 USERCODE

【show configure status】: 读状态寄存器

【show JTAG Instruction Register】: 读取 JTAG 指令寄存器

【ADC Console】: (PGL22 器件支持)

【Jtag scan rate】: 改变获得 adc 数据频率

【Window Depth】: 改变 adc 窗口深度

【Trigger setup】: 配置 debugger core 界面菜单

【Run】: 按照 debugger core 配置条件抓取波形

【Stop Acquisition】: 停止抓取波形，并清除 debugger core 配置信息

【Trigger Immediate】: 不读取 debugger core 配置直接抓取波形

【Waveform】: debugger 波形菜单

【Next Trigger】: 将波形图定位到下个触发位置

【Previous Trigger】: 将波形图定位到上个触发位置

【Trigger Markers】: 设置界面是否显示触发标志

【Go to X Cursor】: 将界面定位到 X 光标

【Go to O Cursor】: 将界面定位到 O 光标

【Place X Cursor】: 放置 X cursor 到鼠标的当前位置

【Place O Cursor】: 放置 O cursor 到鼠标的当前位置

【zoom】: 实现对波形界面的缩放

【search wave】: 对波形数据搜索

【Measure】: 设置波形上显示的数据信息

【Sample in window/ Sample in buffer】: 设置显示数据的格式， in window 以窗口的形式显示数据， in buffer 显示所有数据

【Negative Time/Sample】: 使能负时间显示

【add cursor】 添加辅助光标

【delete cursor】 删除辅助光标

【Listing】 debugger 的 listing 菜单

【Go to】 将界面定位到特定光标

【Place cursor】 将制定光标放到特定位置

【Busplot】 debugger 的 busplot 菜单

【zoom】 对 busplot 界面进行缩放

4.2.4 Settings 菜单

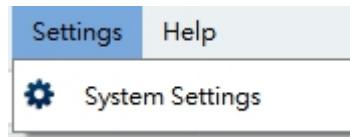


图 4- 6 setting 菜单

【system settings】 详见 debugger 功能系统菜单

4.2.5 Help 菜单

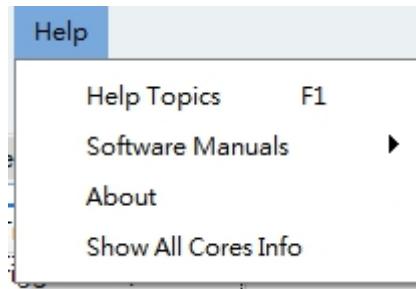


图 4- 7 help 菜单

菜单栏的 Help 可以查看 Fabric Debugger 的软件信息以及 DebugCore 参数信息。

【Help Topics】 会弹出 Pango Assistant, 查看用户手册。

【Software Manuals】 → **【Show User Help】** 会打开 pdf 用户帮助文档。

【About】 会弹出软件信息对话框。

【Show All Cores Info】 会弹出对话框, 每个标签页列出一个 DebugCore 核的所有参数信息。

4.3 工具栏基本操作说明

4.3.1 debugger core 调试模式工具栏



图 4- 8 debugger core 调试工具栏

工具栏按钮（从左至右）说明如下：

Search JTAG Chain: 自动扫描 JTAG 链，查询链上的器件；

Configure Bitstream File: 导入配置文件；

Connect to server: 连接服务器对话框；

Trigger Mode: 设置触发方式:单触发和连续触发

Run: 等同于菜单栏“Trigger Setup”→“Run”；

Stop: 等同于菜单栏“Trigger Setup”→“Stop”；

Trigger Immediately: 等同于菜单栏“Trigger Setup”→“Trigger Immediately”；

Go To X Cursor: 等同于菜单栏“Waveform”→“Go To”→“Go To X Cursor”；

Go To O Cursor: 等同于菜单栏“Waveform”→“Go To”→“Go To O Cursor”；

Go To Next Trigger: 等同于菜单栏“Waveform”→“Go To”→“Trigger”→“Next”；

Go To Previous Trigger: 等同于菜单栏“Waveform”→“Go To”→“Trigger”→“Previous”；

Zoom In: 等同于菜单栏“Waveform”→“Zoom”→“Zoom In”；

Zoom Out: 等同于菜单栏“Waveform”→“Zoom”→“Zoom Out”；

Zoom Fit: 等同于菜单栏“Waveform”→“Zoom”→“Zoom Fit”；

Zoom Full: 等同于菜单栏“Waveform”→“Zoom”→“Zoom Full”；

Move Up: 向上移动 Waveform 中选择的总线及信号；

Move Down: 向下移动 Waveform 中选择的总线及信号。

Analyse wave: 统计波形数据

Previous Transition: 跳到前一个与当前位置值不同的值

Next Transition: 跳到后一个与当前位置值不同的值

System Settings: 系统设置菜单

4.3.2 ADC 调试模式菜单栏



图 4- 9 ADC 调试工具栏

工具栏按钮（从左至右）说明如下：

Search JTAG Chain: 自动扫描 JTAG 链，查询链上的器件；

Configure Bitstream File: 导入配置文件；

Connect to server: 连接服务器对话框；

System Settings: 系统设置菜单

JTAG Scan Rate: 修改获取 ADC 数据的频率

Window Depth: 修改 ADC 波形数据的窗口深度

Clear: 清空 ADC 数据

ADC Settings: 对 ADC 进行配置

logging: 打印 ADC 数据

4.3.3 iScan 菜单栏



图 4- 10 iScan 调试工具栏

工具栏按钮（从左至右）说明如下

Update iScan Status: 刷新眼图界面显示，更新眼图寄存器值

4.3.4 DVIO 菜单栏

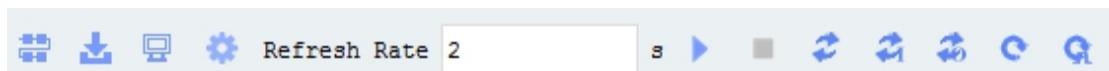


图 4-11 DVIO 调试工具栏

工具栏按钮（从左至右）说明如下：

Search JTAG Chain: 自动扫描 JTAG 链，查询链上的器件；

Configure Bitstream File: 导入配置文件;

Connect to server: 连接服务器对话框;

System Settings: 系统设置菜单

Refresh Rate: DVIO 自动刷新频率设置

Start DVIO: 启动 DVIO 自动刷新

Stop DVIO: 停止 DVIO 自动刷新

Refersh DVIO: 刷新 DVIO

Refresh Probe In: 刷新 DVIO 的 ProbeIn

Refresh Probe Out: 刷新 DVIO 的 ProbeOut

Reset DVIO: 初始化 DVIO 的 ProbeOut

Reset Activity Detector: 重置活动探测器

5 Debugger 功能介绍

5.1 器件栏

用树状结构显示 JTAG 链以及链上器件中的核，如果器件支持 ADC 模块，同时会显示对应 ADC Console 节点，树中会列出 JTAG 链上所有的器件，如果其中的某 FPGA 载入了 DebugCore 核或是支持 ADC 模块，则该 FPGA 会出现叶节点。可以通过叶节点的右键菜单对器件，ADC 以及 Debugger core 进行操作。

对于支持 ADC 以及含有 debugger core 的器件栏显示如下图

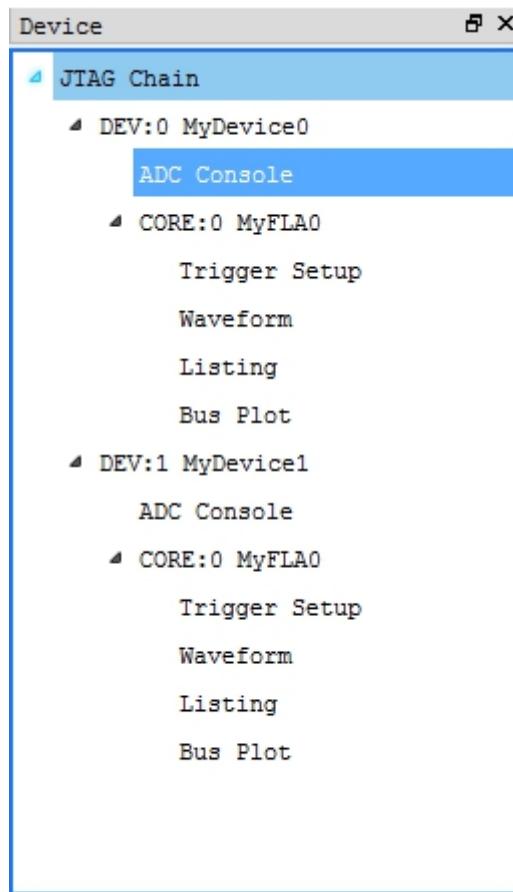


图 5-1 器件栏信息显示树形图

对于器件以及 ADC 节点的右键菜单同第四章的菜单介绍，对于 debugger core 的节点右键菜单显示如下。

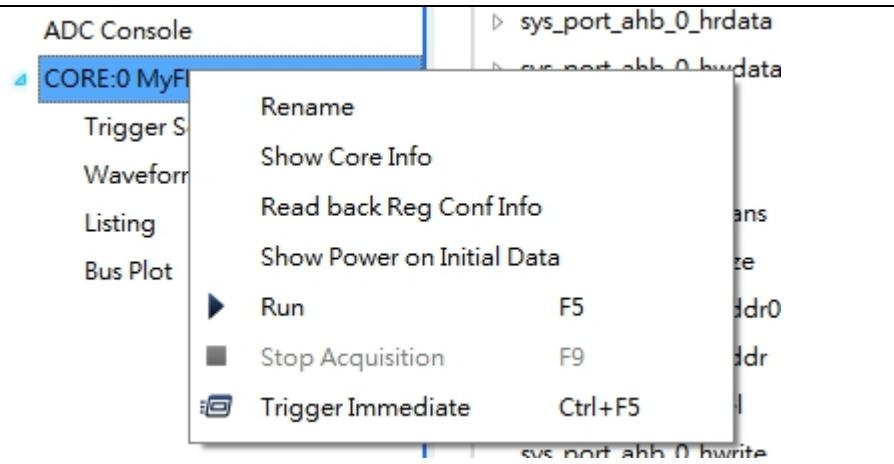


图 5-2 debugger core 右键菜单

【rename】对 debugger core 进行重新命名

【show core info】显示 debugger core 信息

【read back reg conf info】回读 trigger unit, trigger condition, storage style 的寄存器值

【run】在配置好触发条件后，点击 Run 触发条件会被载入到当前的 DebugCore 中，该触发条件被满足或者用户手动结束前会一直存在于 DebugCore 中。一旦触发条件被满足，当前的 DebugCore 会根据 Capture Settings 的设置捕捉数据直到 buffer 被填满，接着数据被 Debugger 接受并显示于 Waveform、Listing 对话框中。现支持同一器件多个核同时抓取。

【Stop Acquisition】从当前 DebugCore 中卸掉触发条件

【Trigger Immediate】忽略掉触发条件及存储条件，捕捉到的数据仅有一个窗口，且从窗口的 sample0 开始触发，当填满 buffer 后，数据被 Debugger 接受并显示于 Waveform、Listing 对话框中。

【Show Power on Initial Data】按照预先设置好的触发条件在 FPGA 上电时就抓取到的数据显示到数据窗口中。在点击了 Run, Stop 或者是 Trigger Immediate 选项后，Power on Initial Data 将会变为无效，此时，如果用户仍然点击获取上电数据选项，界面会给出报错信息。

5.2 信号栏

5.2.1 debugger core 信号栏

Signal Tree 显示在 Project Tree 中所选中 DebugCore 的所有信号，信号可以被重命名、组合成总线、通过右键菜单加入到各个数据显示窗口中。

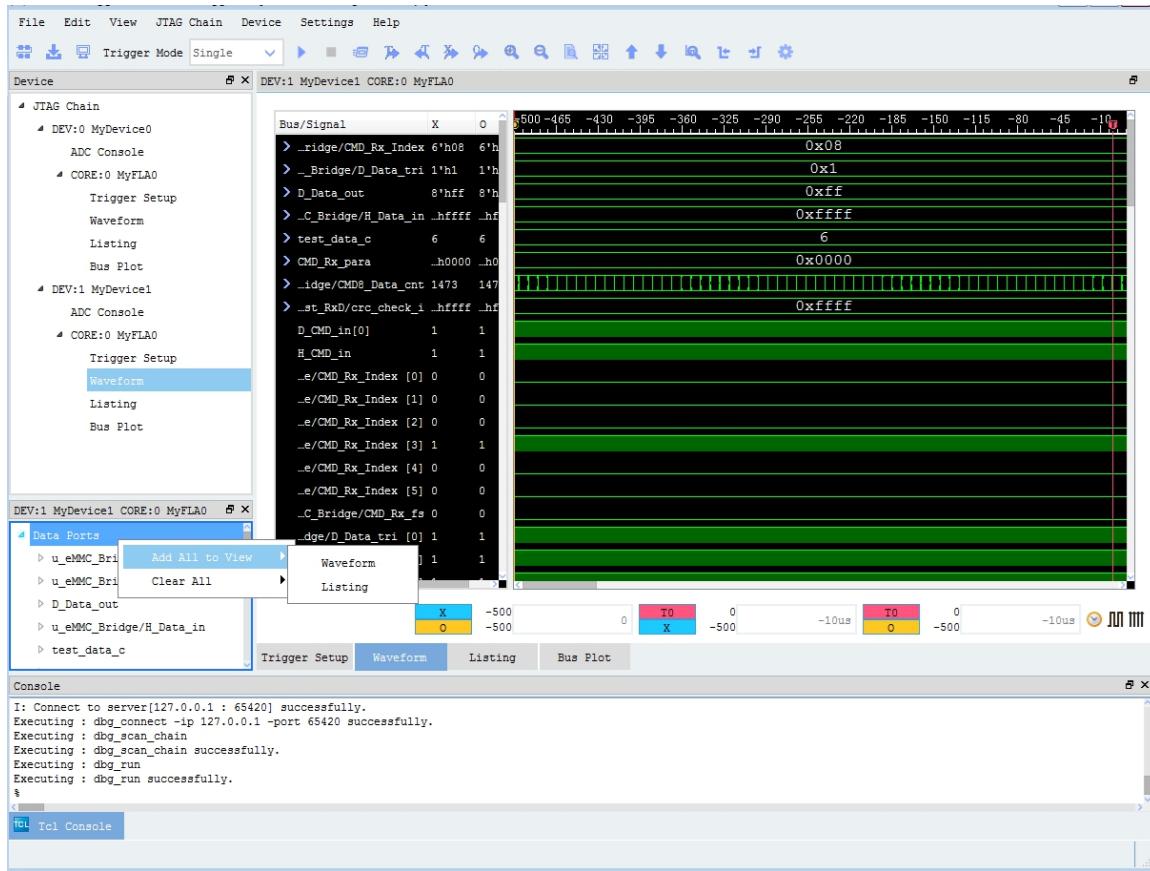


图 5-3 Signal Tree-Data Ports 右键菜单选项

a)添加所有信号和总线到 Waveform 或是 Listing 对话框

右键单击 Data Ports 显示菜单，单击 Add All to View 选项，弹出 Waveform 以及 Listing，通过单击可以选择将所有信号和总线添加到 Waveform 或 Listing。

b)删除 Waveform 或 Listing 上所有信号和总线

右键单击 Data Ports 显示菜单，单击 Clear All 选项，弹出 Waveform 以及 Listing，通过单击可以选择将 Waveform 或 Listing 上的所有信号和总线清除。

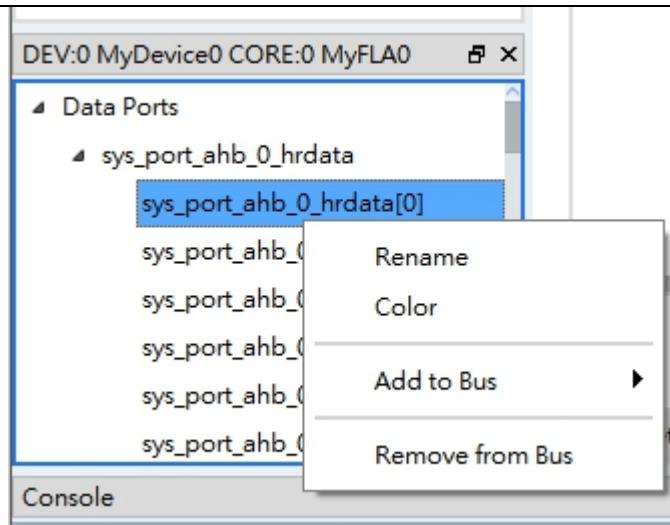


图 5-4 Signal Tree 信号右键菜单选项

a)对信号重命名以及修改信号在 Waveform 以及 Listing 上的显示颜色

通过点击菜单中的 rename 选项可以修改信号的名字。修改后，Trigger setup Waveform 以及 Listing 对应的信号的名字也会同步更新。

b)添加单个信号或是多个信号到 waveform 或是 listing 对话框

通过点击 add to view 菜单选项，通过选择 Waveform 或是 Listing 弹出选项，可以将信号添加到 Waveform 或是 Listing。同时，可以搭配 shift 键进行连续选取多个信号，配合 ctrl 键可以随意选取多个信号。

c)组合信号为总线

仅有数据信号可以被组合为总线。可以通过上面提到的按住 shift 或 ctrl 键的方法选择需要被组合的信号。如果用 shift 键选择信号，则最上面的信号是总线的最低位；如果用 ctrl 键选择信号，则第一根被选择的信号是总线的最低位。

当信号被选好后，在任意被选节点上右键选择“Add to Bus”→“New Bus”，一条新的总线被加入到 Data Port 的最上面；如果想向已有总线添加新的信号，可以右键选择“Add to Bus”，然后选择其子菜单中一根总线的名字，被选信号被加入到已有总线中，新添加的信号总是按序加入到已有信号的最高位。

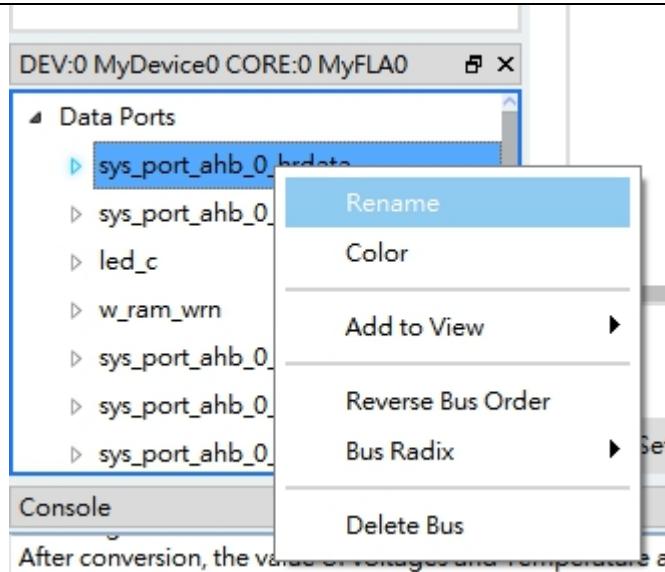


图 5-5 总线的右键菜单选项

a) 反转总线中的信号

在总线节点上右键选择“Reverse Bus Order”，总线中的信号顺序会反转，同时所有显示了该总线的对话框会更新该总线，并重新计算总线的值。

b) 总线的进制

总线可以按照以下的进制显示：

- Binary: 二进制
- Hex: 十六进制
- Octal: 八进制
- Signed: 有符号十进制
- Unsigned: 无符号十进制

c) 删除总线

在总线上右键选择“Delete Bus”，该总线会从 Signal Tree 中删除，同时所有显示了该总线的对话框中删除。

d) 重命名总线，改变总线的显示颜色，添加总线信息到显示对话框中

通过点击 `rename` 可以修改总线的名字，同时 `waveform`, `listing`, `busplot` 也会同步更新。

点击 `color` 可以选取总线在 `waveform` 以及 `listing` 上的显示颜色。

`Add to view` 选项操作与对信号的操作方式相同。

5.2.2 adc 信号栏

通过点击器件栏的 ADC console 即可切换到 ADC 工作模式，信号栏也显示为监测温度，电压的所有通道名称。

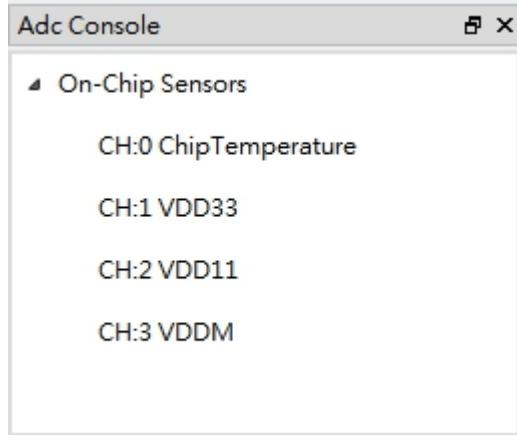


图 5-6 ADC 模块通道树形图

5.3 Trigger Setup 对话框

下载包含 DebuggerCore 位流的器件，双击器件栏中的‘Trigger Setup’叶节点，打开 Debugger Core 图形配置对话框，实时配置 DebuggerCore 无需重新编译，主要从以下三个方面进行配置。

- a) Match Funciton: 为每个 Trigger Unit 定义一个等式或者一个比较式。
- b) Trigger Condition: 定义一个触发条件，该触发条件是任意个 Trigger Unit 的逻辑与、或、非组合，或者它们的顺序触发。
- c) Capture Settings: 定义了数据采样的窗口数、每个窗口的采样点数、以及每个窗口的触发位置。

5.3.1 Match Funcitons

Match Function 定义了每个‘Trigger Unit’在触发条件中的值，如图 5-7。

Trigger Unit	Function	Value	Radix	Counter
• TU0(TriggerPort0)	==	XXXXXXXXXXXXXXXXXXXX	Bin	disabled
counter_0_Z		X		
▼ counter_0		X		
counter_0[0]		X		
▼ counter_3		X		
counter_3[4]		X		
▼ counter_c		X		
counter_c[1]		X		
counter_c[2]		X		
counter_c[3]		X		
▼ dsp_join_kb		X		
dsp_join_kb[4]		X		
dsp_join_kb[5]		X		
dsp_join_kb[6]		X		
dsp_join_kb[7]		X		
N_1_i_i_0		X		
N_2		X		
N_3		X		
N_4		X		
un3_counter_ac0_5_x_...		X		
un3_counter_ac0_5_x_Z		X		

图 5-7 Match Funciton 配置

a) Trigger Unit

Trigger Unit 列出了当前 DebugCore 中所有的 Trigger Unit, 点击 , 该行可以展开, 该 Trigger Unit 对应的 Trigger Port 中所有的信号将被列出, 如图 5-7 所示, 每根信号可以被单独配置。

b) Function

Function 列用来选择 Trigger Unit 的 function。

c) Value

Value 列被单击后, 会变为可编辑状态, 用来输入 Trigger Unit 的值, 值的显示与 Radix 列的选择有关。当输入光标位于待编辑字符前时, 该字符可被编辑, 当输入了一个可被接受的字符后, 该位置原来的字符被替换。不同进制下可接受的字符如下:

- Hex: X, ?, 0-9, 以及 A-F。X 指其所代表的 4bit 值不被关心, 即可以取任意值。当出现“?”, 指其所代表的 4bit 值包含 0-9, A-F, X、R、F、B、N 多个值的组合, 此符号不可输入;
- Octal: X、?、0-7; 当出现“?”, 指其所代表的 3bit 值包含 0-7, X、R、F、B、N 多个值的组合, 此符号不可输入;
- Binary: X (不关心值为多少)、0、1、R (上升沿)、F (下降沿)、B (任意翻转)、N(不翻转), 当 Trigger Unit 可以检测沿(Basic w/edges、Extended w/edges、Range w/edges) 时 R、F、B、N 可输入;

- Unsigned: 0-9(0 至 2 的 n 次幂减一, n 为该 Trigger Unit 对应 Trigger Port 中的信号数)

d) Radix

Radix 列被单击后, 变为下拉列表, 用来选择 Trigger Unit 值以何种进制显示。可以选 Hex、Octal、Bin (Function 为 In Range 及 Out of Range 时不可选)、以及 Unsigned。

e) Counter

Counter 用来选择该 Trigger Unit 的 Function 被满足多少次后, 该 Trigger Unit 的表达式才算满足。如果该 Trigger Unit 中的 Counter 可用, 则该列项为黑色字, 如果不可用, 则该列项为灰色。Counter 列被单击后, 弹出 Match Counter 配置对话框, 如下图:

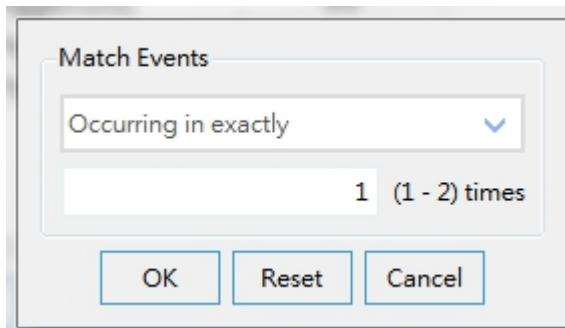


图 5-8 Match Counter 配置对话框

Match Events 下拉菜单中共有 4 个选项:

- counter disable: 不使用计数器
- Occurring in exactly: 精确匹配 N 次 Trigger Unit 设置的条件 (N 为文本框中输入的数值)。
- Occurring in at least: 至少匹配 N 次 Trigger Unit 设置的条件 (N 为文本框中输入的数值)。
- Lasting for at least: 至少连续匹配 N 次 Trigger Unit 设置的条件 (N 为文本框中输入的数值)。

5.3.2 Trigger Condition

Trigger Condition 是一个或多个 Trigger Unit 组成的布尔表达式或者序列, 用来指导 DebugCore 数据的捕捉。可以定义多个 Trigger Condition, 点击“Add”按钮可以新增一个 Trigger Condition; 若想删除 Trigger Condition, 可以先选择它使

其高亮显示，然后点击“Del”按钮。尽管可以添加多个 Trigger Condition，但是每次仅能有一个被使能。

a) Active

Active 列是单选按钮，用来使能当前 Trigger Condition。

b) Trigger Condition Name

Trigger Condition Name 列用来编辑 Trigger Condition 的名字，默认名为“TriggerCondition n”，如下图。

	Active	Trigger Condition Name	Trigger Condition Equation
Add	<input checked="" type="radio"/>	TriggerCondition0	TU0
Del			

图 5-9 Trigger Condition

c) Trigger Condition Equation

Trigger Condition Equation 用来显示 Trigger Unit 组成的布尔表达式或者序列，该序列可以通过点击列项弹出 Trigger Condition 配置对话框来更改。

Trigger Condition 配置对话框的 Boolean 标签页是一个包含了所有 Trigger Unit 的表，每个 Trigger Unit 占用表的一行。“Enable”列用来选择是否将该 Trigger Unit 作为表达式的一部分；“Negate”列用来决定是否将该 Trigger Unit 取逻辑非。所有使能的 Trigger Unit 可以通过逻辑与、或进行组合，可以通过点击单选按钮“Add Equation”、“Or Equation”来选择。“Negate Whole Equation”用来将表达式整体取非。被选择的表达式显示在对话框的最底部，如下图：

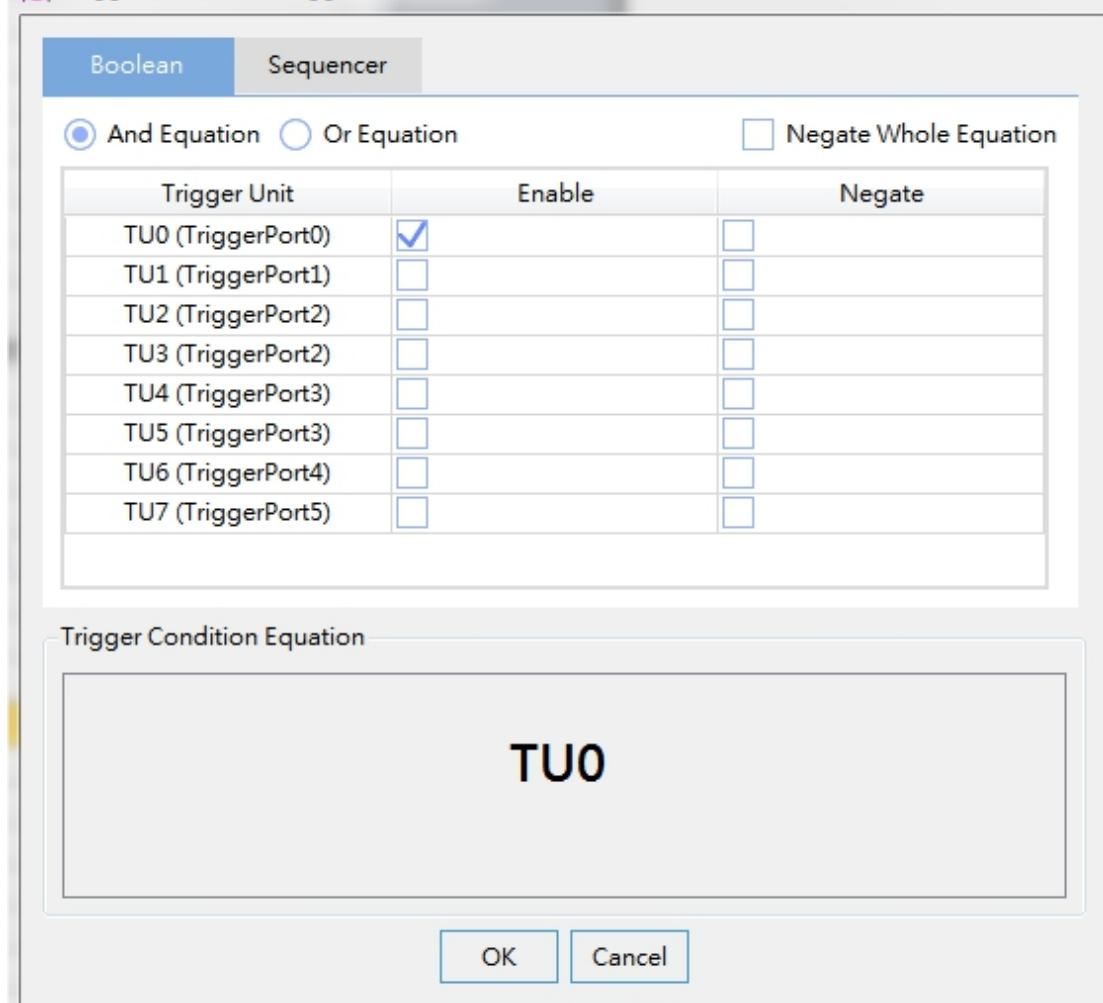


图 5- 10 配置 Trigger Condition 布尔表达式

Trigger Condition 配置对话框的 Sequencer 标签页包含了一个“Number of Levels”下拉列表用来选择触发条件的级数 n, 该级数是当前 DebugCore 的参数之一，最大不超过 16 级；以及一个包含了 n 行的表，用来选择每级触发的 Trigger Unit，该表的“Trigger Unit”列的每个列项被点击后变成一个下拉列表，列表列出了当前 DebugCore 中的所有 Trigger Unit，可以选择任一个，“Negate”列用来将其对应的 Trigger Unit 取非；被选择的 Trigger Unit 按照 Level 顺序等待触发，顺序触发从 Level1 开始，直到 Level1 被满足后，Level2 才能被等待触发，依次类推。所有的触发可以是连续的，也可以是不连续的，通过“Use Continuous Match Event Only”选择。触发条件表达式显示于对话框底部，如下图。

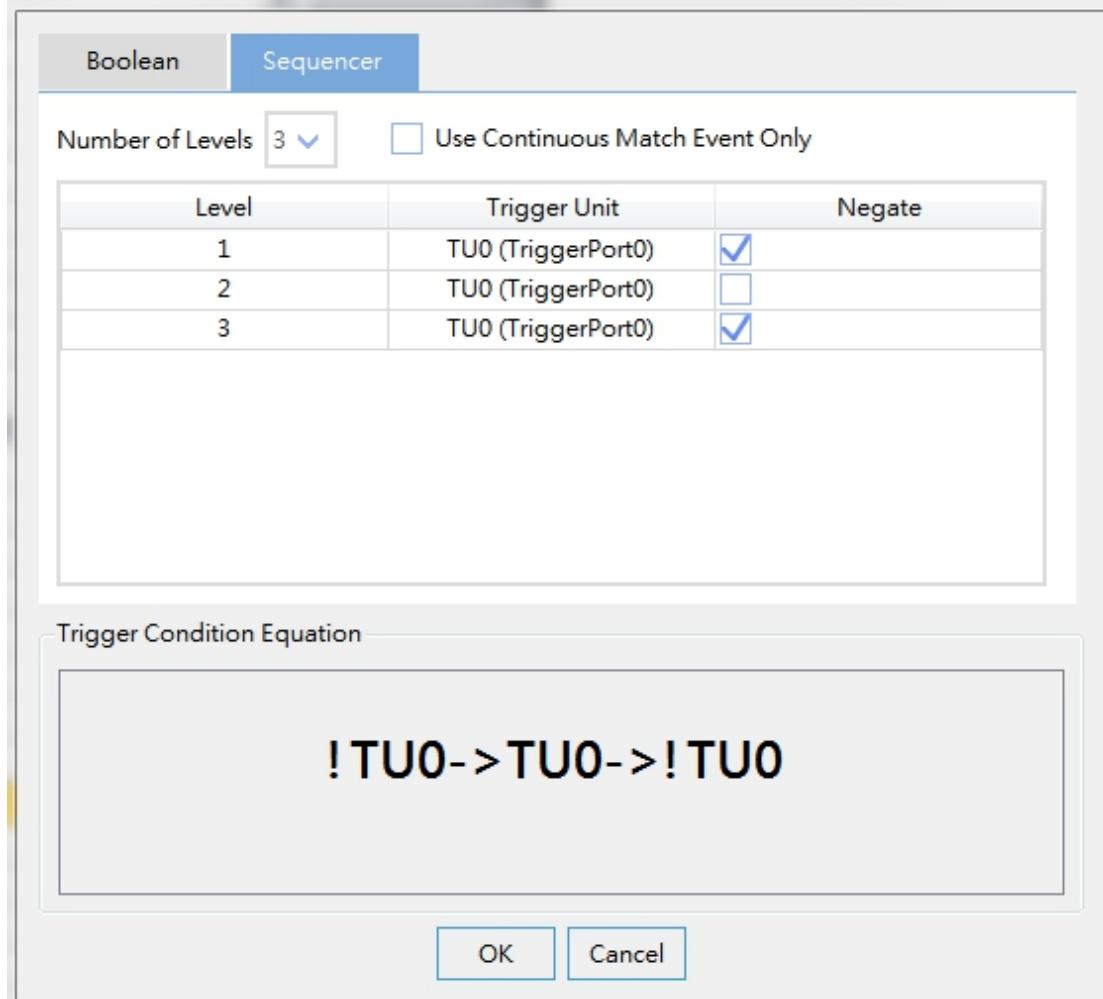


图 5-11 配置 Trigger Condition 顺序触发表达式

d) Running the Trigger

在配置好触发条件后，点击“Trigger Setup”→“Run”，触发条件会被载入到当前的 DebugCore 中，该触发条件在被满足或者用户手动结束前会一直存在与 DebugCore 中。一旦触发条件被满足，当前 DebugCore 会根据 Capture Settings 的设置捕捉数据直到 buffer 被填满，接着数据被 Debugger 接收并显示于 Waveform、Listing 对话框中。

如果想要强制触发，则选择“Trigger Setup”→“Trigger Immediate”。这种触发方式会立即触发，并忽略触发条件及存储条件，捕捉到的数据仅有一个窗口，且从窗口的 sample0 开始触发，当填满 buffer 后，数据被 Debugger 接收并显示于 Waveform、Listing 对话框中。

e) Stopping the Trigger

“Trigger Setup”→“Stop Acquisition”可以从当前 DebugCore 中卸载掉触发条

件。

5.3.3 Capture Settings

Capture Settings如下图所示，它定义了窗口数、每个窗口的数据深度及触发位置。每个窗口显示一个连续的采样序列，有且仅有一个触发点。如果Capture Settings设置有误，则相应区域会变为红色，用以警告用户。

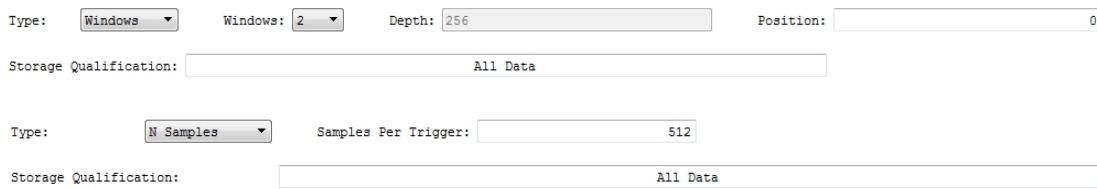


图 5- 12 Capture Settings

a) Type

Type 是个下拉列表，有两个可选项。如果选择“Windows”，则每个窗口中捕捉的采样点数必须为 2 的幂，触发点可以是窗口中的任何位置；如果选择“N Samples”，则会尽可能对多的捕捉窗口，但触发点一定是每个窗口的第一个采样点。

b) Windows

Windows 是个输入框，仅当 Type 选择“Windows”时可用。Windows 用来定义每次数据捕捉的窗口数目，可以为正整数 N ($1 \leq N \leq$ 当前 DebugCore 的最大采样深度)。

c) Depth

Depth 是个下拉列表，仅当 Type 选择“Windows”时可用。Depth 定义了每个窗口的采样深度，下拉列表中的选项可根据 Windows 的输入值重新计算并更新可用的值，且一定为 2 的幂。

d) Position

Position 是个输入框，仅当 Type 选择“Windows”时可用。Postion 定义了每个窗口触发点的位置，可取整数 N ($0 \leq N \leq$ Depth 输入框的值-1)，若输入的值大于“Depth”值，输入框的背景变红色。

e) Samples Per Trigger

Samples Per Trigger 是个输入框，仅当 Type 选择“N Samples”时可用。Samples Per Trigger 定义了每个窗口的采样点数，可取整数 N ($1 \leq N \leq$ 当前 DebugCore 的

最大采样深度), 若输入的值大于“Depth”值, Position 输入框的背景变红色。

f) Storage Qualification Condition

Storage Qualification Condition 是个文字显示框, 点击后弹出“Storage Condition”对话框。Storage Qualification Condition 定义了一个布尔表达式, 它由一个或多个 Trigger Unit 进行逻辑与、或、非组成。这个表达式用来指导触发条件满足后数据的存储, 即触发后, 满足表达式的数据才被存储。

Storage Condition 对话框有一个包含了所有 Trigger Unit 的表, 每个 Trigger Unit 占用表的一行。“Enable”列用来选择是否将该 Trigger Unit 作为表达式的一部分; “Negate”列用来决定是否将该 Trigger Unit 取逻辑非。

Storage Qualification Condition 表达式可以选择捕捉所有的数据, 或者捕捉满足布尔表达式的数据, 布尔表达式可以通过选择“Negate Whole Equation”整体取非。表达式被显示于对话框的下方。

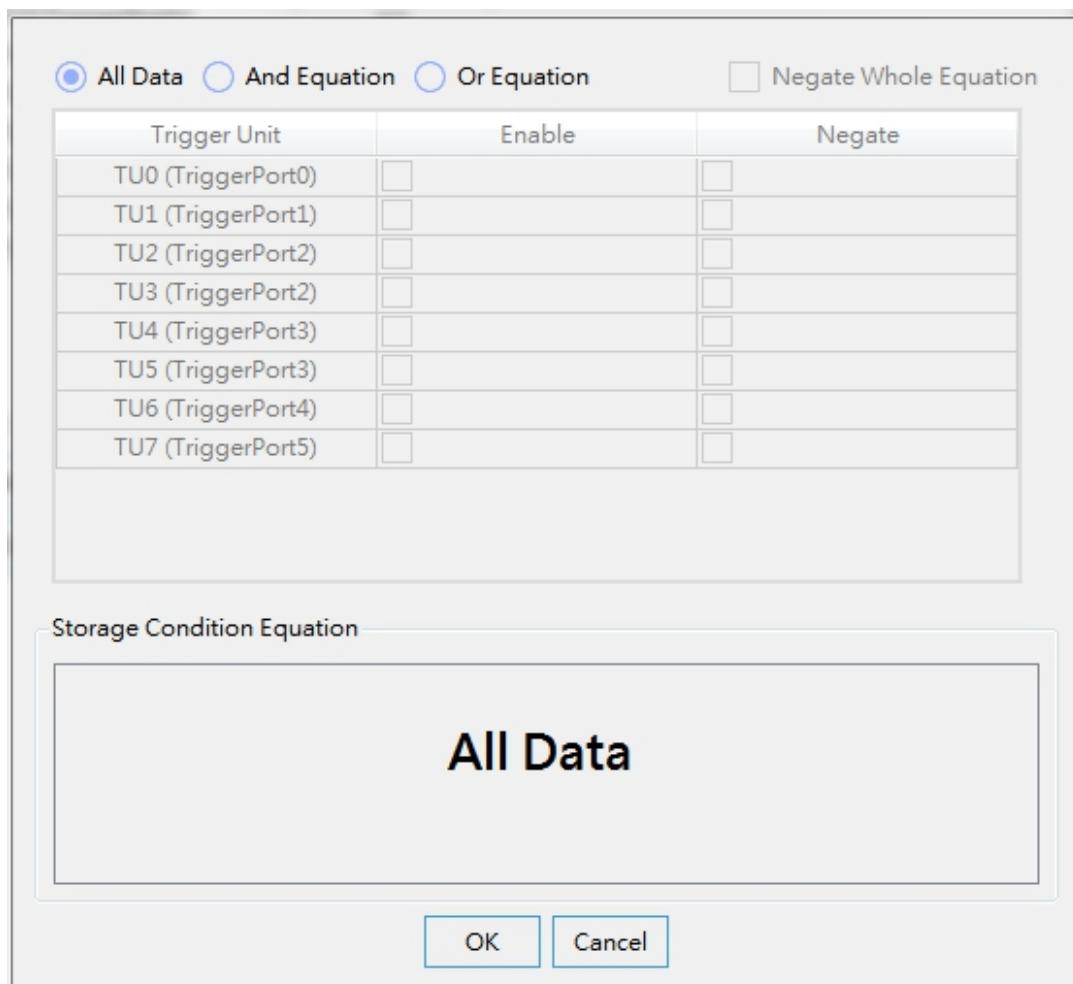


图 5- 13 Storage Qualification Condition 选择 All Data

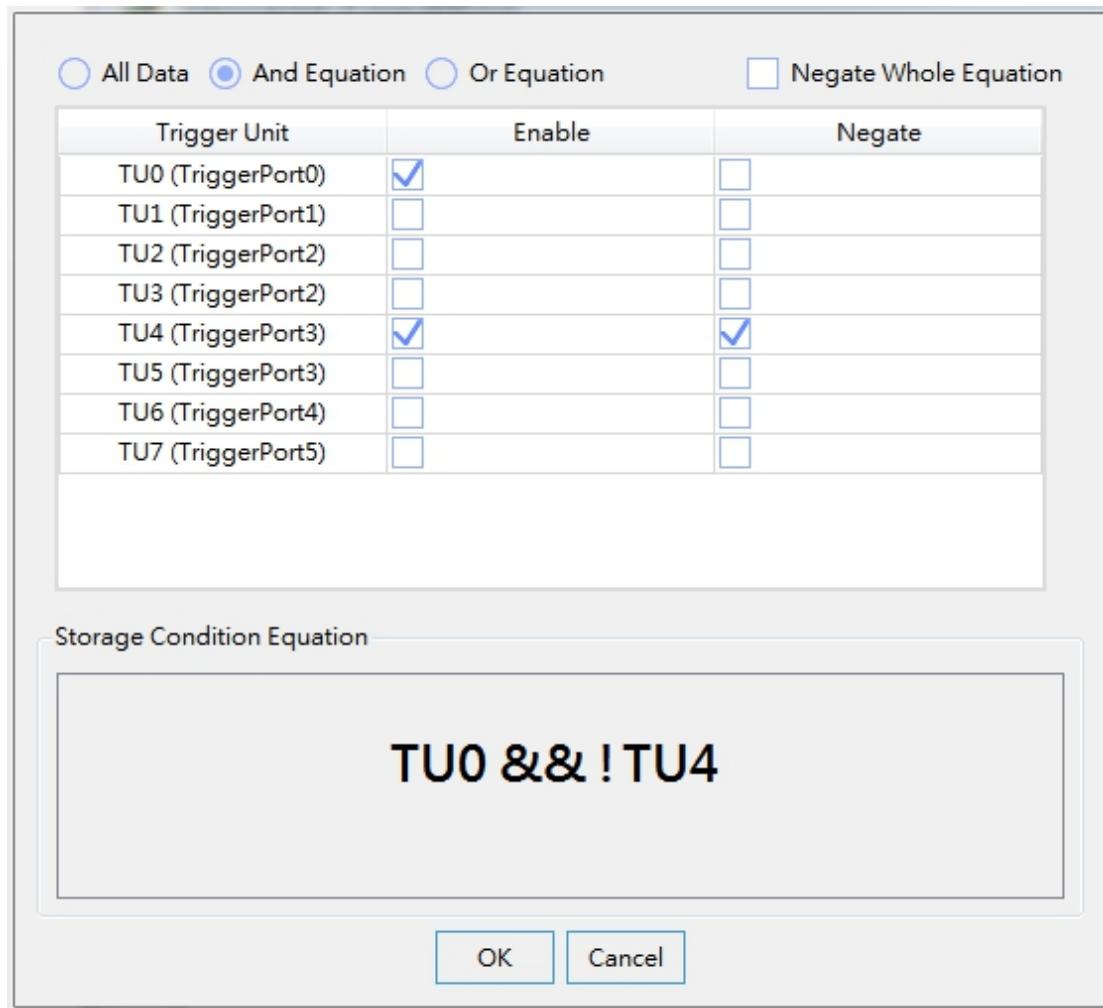


图 5- 14 Storage Qualification Condition 选择布尔表达式

5.4 Waveform 对话框

Waveform 对话框用来以波形的形式显示 DebugCore 捕捉到的数据，双击 Pro 器件栏中的 ‘Waveform’ 叶节点，Waveform 对话框被打开，Waveform 分为两部分：

- Signed Browser：以树状结构显示对话框中所有的信号及总线；
- Waveform Display：以波形的形式显示捕捉到的数据。

5.4.1 Signal Browser

所有 Signal Tree 中的操作都可在此实现。

- 重排序总线及信号

总线或信号可以在 Signal Browser 中重新排列顺序。选中树中某些条目，右键菜单或者在菜单栏上点击 Move Up、Move Down，对选择信号进行上下移动排序。也可以通过鼠标左键对信号拖拽来进行排序，但不支持总线中的单个信号的拖拽。

➤ 剪切、复制、粘贴、删除总线及信号

选择一个或多个信号或总线，右键点击“Cut”、“Copy”、“Paste”、“Remove from View”会执行相应的操作。

➤ 总线的拆分

支持新添加的总线进行拆分，原有的总线不支持拆分。选择一个新添加的总线，通过鼠标右键菜单中的“Split Bus”进行操作。

➤ 显示 short name

Settings>System Settings>waveform 设置 wave signal name，可选择显示 short name，修改信号全名会同步修改简称，修改简称不会影响到信号全名。

5.4.2 Waveform Display

Waveform 对数据的显示于普通波形显示商业软件或者逻辑分析仪相似。

a) 缩放

菜单栏点击“Waveform”→“Zoom”→“Zoom In”或者在波形显示区域右键菜单中选择“Zoom”→“Zoom In”，波形会以当前中线处为中心放大；菜单栏点击“Waveform”→“Zoom”→“Zoom Out”或者在波形显示区域右键菜单中选择“Zoom”→“Zoom Out”，波形会以当前中线处为中心缩小；，若要波形自适应调整缩放系数，则点击菜单栏“Waveform”→“Zoom”→“Zoom Fit”或者在波形显示区域右键菜单中选择“Zoom”→“Zoom Fit”；若要显示整个波形，则点击菜单栏“Waveform”→“Zoom”→“Zoom Full”或者在波形显示区域右键菜单中选择。

在波形显示部分（不包括 Sample 刻度部分以及波形下面的空白部分）左键点击，按住左键不放拖动，向右下角拖动鼠标会拖出一个蓝色矩形框，在某位置放开鼠标（该位置必须为波形显示部分），则鼠标按下与放开之间的波形被显示于波形显示区，如下图。

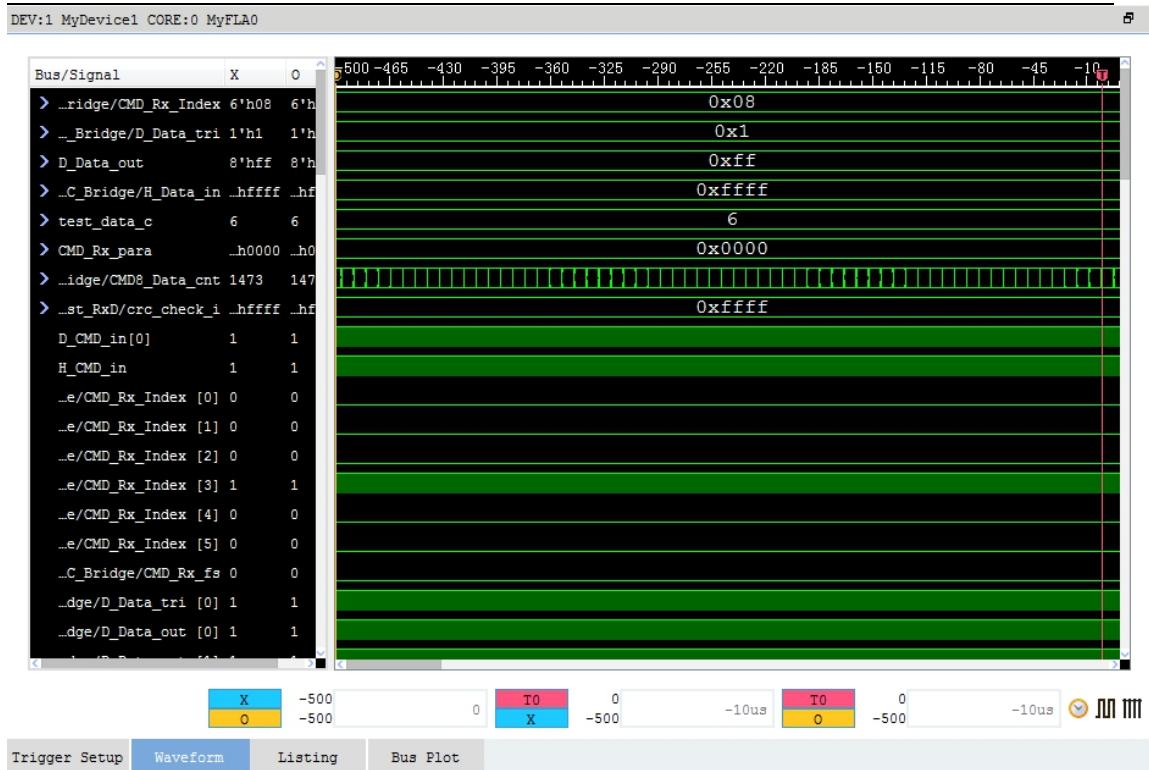


图 5-15 放大指定区域的波形

通过点击菜单栏“Waveform”→“Zoom”→“Zoom X 、 O” 或者在波形显示区域右键菜单中选择“Zoom”→“Zoom X 、 O”，还可以放大 X、O 光标之间的波形。还可以通过“Zoom”→“Zoom Previous”缩放到前一次的大小，或通过“Zoom”→“Zoom Forward”缩放到下一次的大小。

b) Go To

菜单栏点击“Go To X/O cursor”或者在波形显示区域右键菜单中选择“Go To”例：“Go To X Cursor”、“Go To O Cursor”会让 X、O 光标显示于中心，“Go To Next/Previous Trigger”可以跳转到上一触发或者下一触发位置

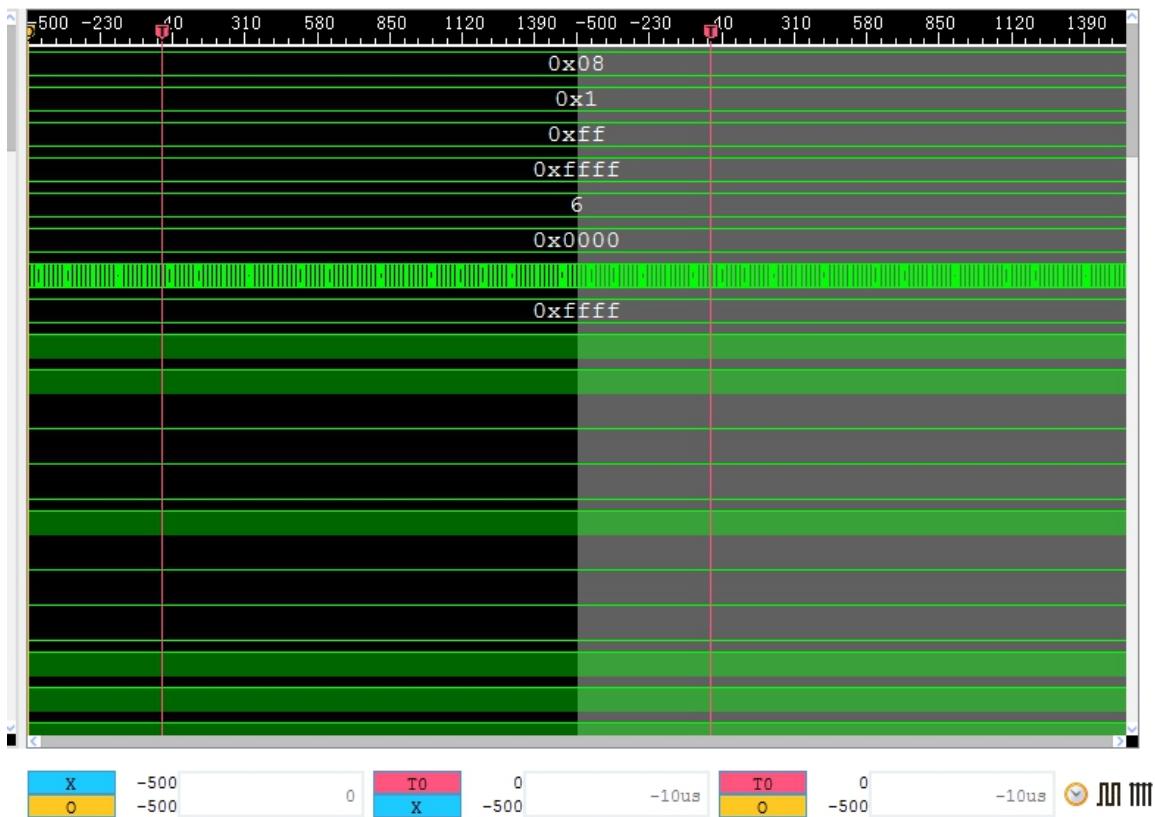


图 5-16 Go To Waveform

c) 光标

波形显示区有两个光标，分别为 X、O。可以通过右键菜单选择“Place X Cursor”、“Place O Cursor”将光标移动到右键点击的位置。同时在 Signal Browser 中，对应的 X、O 列会更新当前总线或信号的值。光标的初始位置为 Sample 0。同时，可以通过右键菜单的 Add Cursor 增加用户光标 A/B/C/D/E/F/G，同 X、O 光标，可以通过 place Cursor 菜单对光标进行放置，通过 Go To 菜单进行光标的定位，也可以通过快捷键 shift+对应光标名称 (shift+A) 对光标进行放置，光标名称进行光标的定位 (A)，同时用户光标可以通过菜单进行删除，快捷键 delete 也可以快速删除用户光标。

光标也可被拖动。将鼠标停留在光标上，鼠标形状会变为“<->”，这时按住鼠标拖动，光标会随着鼠标移动位置。

d) 时间刻度

时间刻度位于波形显示区的顶端，默认情况下时间刻度尺上的数字显示当前窗口的当前 sample 值。同时，时间刻度上会用红色刻度线动态显示鼠标所在位

置，鼠标移到时间栏，可以显示时间刻度的具体值。若需要显示每个 Sample 在整个数据 buffer 中的位置，则右键点击“Sample # in Buffer”；若要显示每个 Sample 在其所在窗口中的位置，则右键点击“Sample # in Window”；是否显示负时间，通过右键点击“Negative Time/Samples”，打勾表示显示负时间，否则不显示负时间。

e) 分隔线

红色的光标线标记出每个窗口的触发点位置，不同窗口依据不同颜色区分，分隔线可以被显示，也可以不显示，“Trigger Markers”，打勾表示显示，不打勾表示不显示。

f) 撷取时间

debugger core 界面标题显示波形时间，波形数据更新同步更新撷取时间。

5.5 Listing 对话框

Listing 对话框用来以列表的形式显示 DebugCore 捕捉到的数据，双击 Project Tree 中的“Listing”叶节点，Listing 对话框被打开。

Listing 列表中，每根信号或总线为一列，如下图。列表中的信号和总线可以被重命名及更改颜色，通过在列头右键菜单点击“Rename”、“Color”。信号和总线也可被 Cut、Copy、Paste。

Sample	s_port_ahb_0_haddr[0]	s_port_ahb_0_haddr[1]	s_port_ahb_0_hdata[0]	s_port_ahb_0_hsize[0]	s_port_ahb_0_htrans[0]	s_port_ahb_0_hwdata[0]	led_c[0](Hex)	W
0	400005F4	40000150	797FCE7	2	1	0	0	
1	40000154	400005F4	0	2	1	400001F0	0	
2	40000124	40000154	87930000	2	1	0	0	
3	40000128	40000124	2783A015	2	1	0	0	
4	4000012C	40000128	8713FEC4	2	1	0	0	
5	400005F4	4000012C	26230017	2	1	0	0	
6	40000130	400005F4	0	2	1	400001F0	0	
7	40000134	40000130	717FEE4	2	1	0	0	
8	400005F4	40000134	7130000	2	1	0	0	
9	40000138	400005F4	0	2	1	1	0	
10	4000013C	40000138	78A0CE7	2	1	0	0	
11	40000140	4000013C	71797BA	2	1	0	0	
12	40000144	40000140	7130000	2	1	0	0	
13	40000144	40000144	C398FD47	2	0	0	0	
14	40000148	40000144	C398FD47	2	1	0	0	
15	4000014C	40000148	FEC42703	2	1	0	0	
16	40000200	4000014C	DCE34785	2	1	0	0	
17	40000150	40000200	0	2	1	40000112	0	
18	400005F4	40000150	797FCE7	2	1	0	0	
19	40000154	400005F4	0	2	1	400001F0	0	
20	40000124	40000154	87930000	2	1	0	0	
21	40000128	40000124	2783A015	2	1	0	0	
22	4000012C	40000128	8713FEC4	2	1	0	0	
23	400005F4	4000012C	26230017	2	1	0	0	
24	40000130	400005F4	0	2	1	400001F0	0	
25	40000134	40000130	717FEE4	2	1	0	0	
26	400005F4	40000134	7130000	2	1	0	0	

X: 0 O: 0 (X-O): 0

Trigger Setup Waveform Listing Bus Plot

图 5-17 Listing 对话框

a) 重新排列总线及信号

鼠标点击于目标列的列头部分，拖动至目标位置。

b) 从 Listing 中移除信号或总线

在目标列列头右键点击“Remove from Viewer”，被选列被移除。如果要移除 Listing 中全部的信号和总线，则在任意列列头右键点击“Clear All”，所有的信号和总线将被移除。

c) 光标

Listing 中的光标与 Waveform 中的光标始终保持位置一致。可以通过右键点击“Place X Cursor”、“Place O Cursor”将光标置于右键点击的位置；或者可以通过鼠标拖动其位置。

d) Go To

Listing 可以通过“Go To X Cursor”、“Go To O Cursor”自动滚动至 X、O 光标的位置。

5.6 Bus Plot 对话框

Bus Plot 对话框用来显示总线，双击 Project Tree 中的“Bus Plot”叶节点，Bus Plot 对话框被打开。

所有总线都可以在 Bus Plot 对话框中显示，显示方式有两种：一种按照总线值随着时间的变化显示，另一种按照总线 vs 另一总线值显示。

a) Plot 种类

对话框左上角的单选按钮可以选择 Plot 的种类，有两种 Plot 种类：data vs time 和 data vs data。选择 data vs time 时，所有总线都可以同时显示于对话框中，如图 32。选择 data vs data 时，需要选择 X、Y 坐标的总线，X 坐标轴上的每个点对应 X 总线在某时刻的值，Y 坐标轴上的点是 Y 总线在同一时刻的值，如下图。图中总线的值均按照 Unsigned 计算。

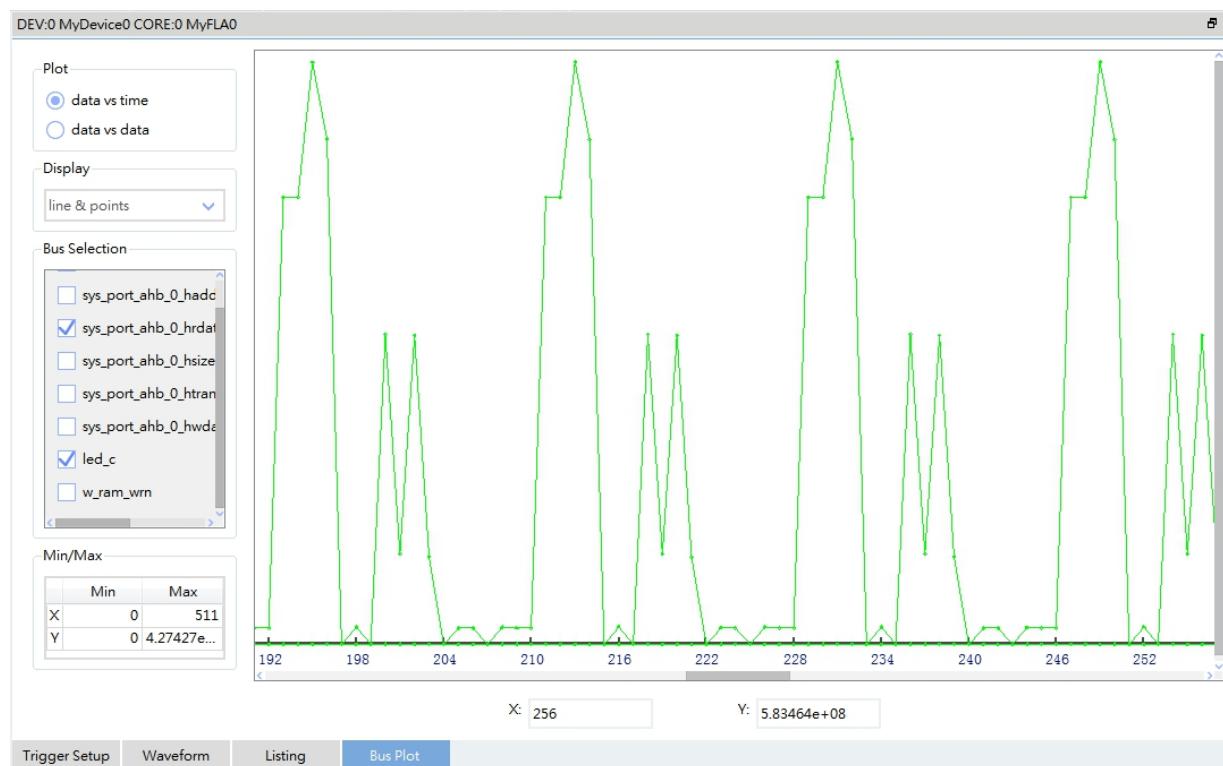


图 5-18 Bus Plot 对话框：data vs time

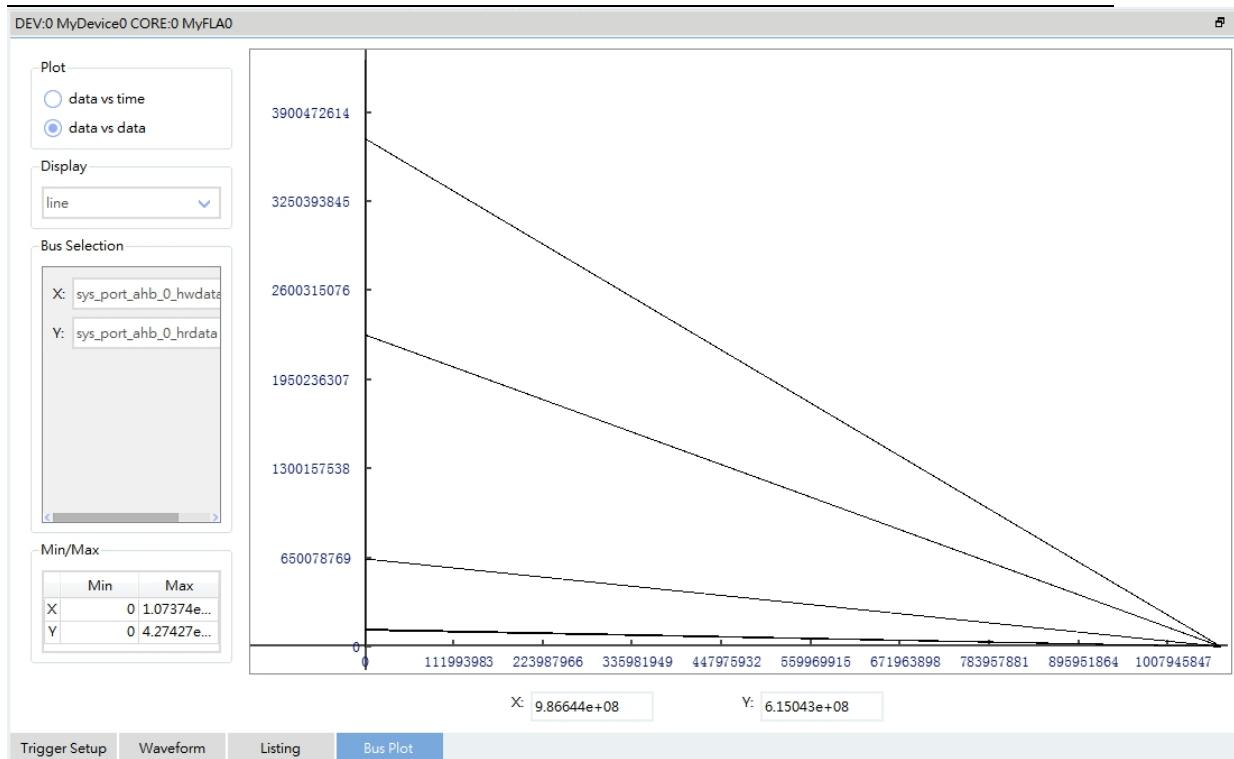


图 5- 19 Bus Plot 对话框 : data vs data

b) 显示方式

Bus Plot 可以有三种显示方式：点、线、及点线组合。显示方式的选择通过 Display 下拉列表选择，显示方式影响所有总线的显示。

c) 总线的选择

data vs time 时可以选择任意总线，data vs data 时可以通过 X、Y 下拉列表选择 X、Y 坐标轴对应的总线。

d) 最大、最小值

Min/Max 表用来显示当前 Bus Plot 显示区内纵横坐标的最大值与最小值。

e) 鼠标跟踪

对话框底部的 X: 和 Y: 显示框显示当前鼠标所在位置对应在 X、Y 坐标轴上的值。

5.7 ADC 调试对话框

ADC 对话框用来显示 ADC 调试界面，观测芯片电压温度数据，当前只有

PGL12G, PGL22G, PG2L100H, PG2L25H, PG2T390H 器件支持 ADC 模块，双击 Project Tree 中的“ADC Console”叶节点，ADC Console 对话框被打开。如图 5-14 所示

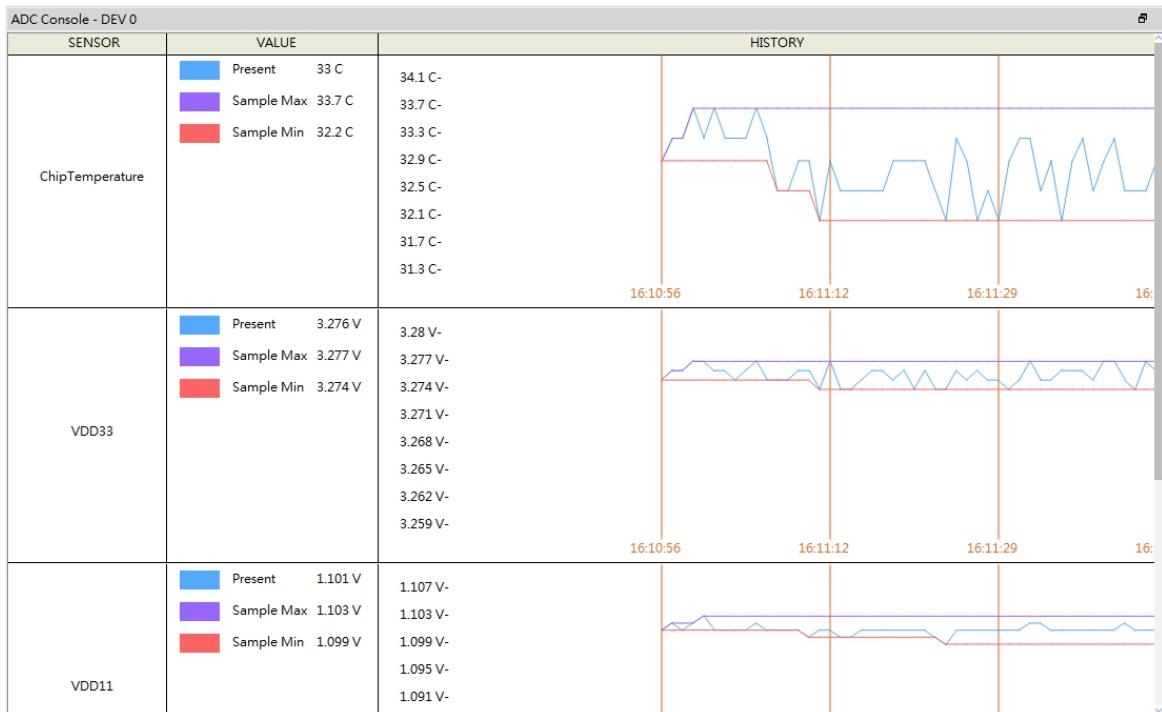


图 5- 20 ADC 对话框

a) sensor

sensor 用来显示当前监测通道的名称

b) value

value 动态显示通道的当前值以及监测过程中的最大值和最小值，同时显示对应波形颜色。

c) history

history 动态显示监测通道值的变化情况，并从右侧向左侧滚动，右侧为当前值，同时显示采集数据时间，通过工具栏或是菜单栏可以修改窗口大小

5.8 ADC 配置对话框

ADC Settings 用于配置或调整 ADC 工作参数，选择工具栏  图标即可打开 ADC Settings 界面。ADC 参数主要通过配置 ADC 的控制寄存器，Edit Register 配置对话框如下图所示：

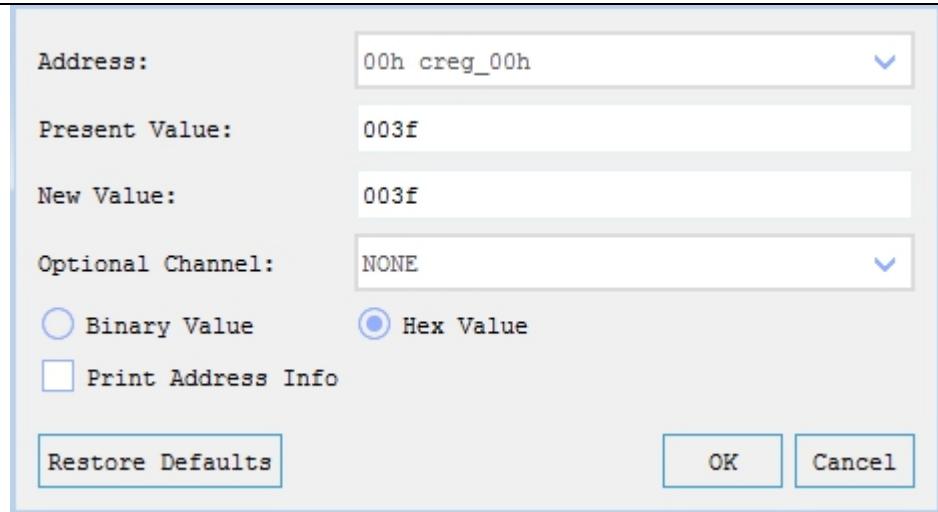


图 5-21 Edit Register 配置界面

【Address】 器件可配置寄存器地址名称列表，下拉框选择地址。

【Present Value】 显示“Address”中选择地址的当前值，不可编辑。

【New Value】 默认状态下，显示“Address”中选择地址的当前值，可通过修改该值，实现控制寄存器的配置，点击“OK”按钮后生效。

【Optional Channel】: 选择是否在 ADC 界面显示外部通道（外部电压），默认状态“NONE”，即不显示外部通道，如果下拉列表选择其它的外部通道，点击“OK”按钮后，该通道将在 ADC 界面显示，并在 ADC 信号栏中显示选择的通道。

【Binary Value / Hex Value】 切换“Present Value”和“New Value”的显示进制及输入格式，支持二进制和十六进制，默认是十六进制。

【Print Address Info】 使能打印 ADC 寄存器信息到控制台，默认不使能，点击“OK”按钮后生效。

【Restore Defaults】 点击后，弹出询问对话框，询问是否恢复所有寄存器的默认值，点击询问对话框的“OK”后，器件所有可配置寄存器的值将恢复默认状态值。

5.9 ADC 寄存器

ADC 的寄存器分为控制寄存器（Control Register）和状态寄存器（Status Register）。用户通过配置 ADC 的控制寄存器来对温度和电压进行监控。

5.9.1 Logos 系列

Logos 系列器件只有 PGL12G 和 PGL22G 支持 ADC 功能。

Logos 器件的 ADC 工作模式包括默认模式、单次扫描模式、连续扫描模式、单通道模式。

在默认模式下，ADC 将自动对芯片上的温度和电压进行扫描转化，并将转化值存入状态寄存器中。ADC 会自动进行校准，并且都是经过 16 次采样求平均值之后得到的。

在单次扫描模式下，ADC 会从通道的 LSB 位扫到 MSB 位，然后停止，即只进行一次扫描。

连续扫描模式下 ADC 会一直进行扫描，直到更改扫描模式。

单通道模式，需要将Mode register (01h) 中的SEQ1到SEQ0设置为2'b11才会有效。在这种模式下，用户需要设置想要进行数模转化的通道，通道的选择是通过mode reg (01h) 中的CH3到CH0并配合DS和BU两位进行控制。

控制寄存器名称及地址如下表所示：

Name	Address	Description
Configuration Register	00h	ADC的配置位（见配置寄存器（Configuration Register））
Mode Register	01h	ADC的模式控制位（见模式控制寄存器（Mode Register））
Sequence Register.#0	02h	ADC的Sequence模式下的配置控制位（见扫描控制寄存器（sequence registers））
Sequence Register.#1	03h	ADC的Sequence模式下的配置控制位（见扫描控制寄存器（sequence registers））
Temperature sensor Register	04h	ADC的Temperature sensor阈值的配置控制位（见温度检测控制寄存器（Temperature sensor registers））

表 1 控制寄存器名称及地址表

a) 配置寄存器

配置寄存器 (00h) 用来对ADC内部的一些模块进行具体配置。其各个bit的具体分配如图所示。

Config Reg.	DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
ADDR<4:0>=00h	AVG1	AVG0	CAL1	CAL0	REF	CREF	FS2	FS1	FS0	VCM2	VCM1	VCM0	DIV3	DIV2	DIV1	DIV0

图 5-22 配置 bit 分配

配置寄存器详细说明如下表：

Bit	Name	Description	对应输出给 ADC_CORE_TOP的信号
DI0 to DI3	DIV0 to DIV3	通过配置不同的值对输入时钟进行分频（见ADC时钟），来改变ADC采样时钟的频率（具体配置见表3）	
DI4 to DI6	VCM0 to VCM2	通过配置改变ADC内部比较器输入的共模电压（具体配置见表4）	AD_VCM_G<2:0>
DI7 to DI9	FS0 to FS2	通过配置改变ADC的量程（具体配置见表5）	AD_FS_G<2:0>
DI10	CREF	对ADC的calibration的参考源进行选择 0：接ADC电源VDD11 1：外部参考源（与D11控制的外接参考源相同）	AD_CREF_SEL
DI11	REF	对ADC的参考源进行选择 0：内部参考源（BGP产生） 1：外部参考源	AD_REF_SEL
DI12 to DI13	CAL0 to CAL1	对ADC的calibration进行选择（具体见表6）	
DI14 to DI15	AVG0 to AVG1	通过配置对ADC输出的CODE多次求平均（具体配置见表7）	

表 2 配置寄存器详细表

DIV3	DIV2	DIV1	DIV0	Description
0	0	0	0	2
0	0	0	1	2
0	0	1	0	3

0	0	1	1	4
...
1	1	1	1	16

表 3 采样时钟控制位

VCM2	VCM1	VCM0	Description
0	0	0	VCM=0.8V
0	0	1	VCM=0.9V
...
1	1	1	VCM=1.5V

表 4 共模电压控制位

FS2	FS1	FS0	Description
0	0	0	Bipolar mode VFS=1V; unipolar and single-end mode VFS=0.5V
0	0	1	Bipolar mode VFS=1.2V; unipolar and single-end mode VFS=0.6V
...
1	1	1	Bipolar mode VFS=2.4V; unipolar and single-end mode VFS=1.2V

表 5 量程控制位

NOTE: VFS表示满量程。

CAL1	CAL0	Description
0	0	ADCs offset and gain calculation disable
0	1	ADCs offset calculation enable
1	X	ADCs offset and gain calculation enable

表 6 校准选择控制位

AVG1	AVG0	Description
0	0	1
0	1	16
1	0	64
1	1	256

表 7 求平均控制位

b) 模式控制寄存器

模式控制寄存器 (01h) 用来对ADC的工作模式进行配置，各个bit的分配如图所示。1MSPS模式是由寄存器1M (Mode Reg, 01h) 来控制，设为1时，进入1MSPS模式。进入此模式后，如果CAL1和CAL0 (config reg 00h中) 不为2'b00，则会先做一次校准值计算，并将offset error 校准值和 gain error 校准值写入相应的status register，之后再开始所选择的通道信号的转换。各个bit的定义见表：

Mode Reg. ADDR<4:0>=01h	DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
	SEQ1	SEQ0	CF	1M	clksw	0	0	0	MUX1	MUX0	CH3	CH2	CH1	CH0	BU	DS

图5- 23 模式控制寄存器bit分配

Note: 设置为0的bit要一直配置为0.

Bit	Name	Description	对应输出给 ADC_CORE_TOP的信号
DI0 to DI1	DS and BU	此控制位在single channel模式和1MSPS模式(见SEQ1、SEQ0的定义和1M的定义)下有效，其中DS选择使用single-end还是diff模式，由于diff模式中包括bipolar和unipolar，因此由BU来进行选择，[BU,DS]的编码具体如下： X0: single-end; 01: unipolar; 11: bipolar	AD_MODE_DIFF_EN: X0 single-end: 1'b0; 01 unipolar: 1'b0; 11 bipolar: 1'b1
DI2 to DI5	CH0 to CH3	此控制位在single channel模式和1MSPS模式下有效，用来进行通道选择，要配合DS和BU使用。	见下表
DI6 to DI7	MUX0 to MUX1	此控制位在single channel模式和1MSPS模式下有效，用来进	AD_MUX<1:0>取值如下： <MUX1,MUX0>=0X时为

		行channel 13输入信号的选择, [MUX1, MUX0]编码如下: 0X: 选择VDD33; 10: 选择VDD11; 11: 选择VDDM	2'b01; <MUX1,MUX0>=10时为 2'b10; <MUX1,MUX0>=11时为 2'b11;
DI11	clksw	时钟切换控制位	进入用户模式后,若控制位为 1'b0: 选择clk_osc作为adc时 钟; 1'b1: 选择dclk作为adc时钟;
DI12	1M	1MSPS模式控制位	1'b0: 工作模式由 [SEQ1:SEQ0]设置; 1'b1: 工作模式为1MSPS模 式;
DI13	CE	连续采样模式 (continus sampling mode) 和主动控制采 样模式 (event-drive sampling mode) 0: continue 1: event-drive	
DI14 to DI15	SEQ0 to SEQ1	Sequence模式选择 (具体见表 9)	

表 8 模式控制寄存器详细表

SEQ1	SEQ0	Description	备注
0	0	Default sequence Mode 默认扫描模 式	最大支持0.5MSPS
0	1	Single pass sequence mode单次扫描 模式	最大支持0.5MSPS
1	0	Continue sequence mode连续扫描模 式	最大支持0.5MSPS
1	1	Single channel mode (sequence mode off) 单通道模式	最大支持0.5MSPS

表 9 模式选择配置表

CH3:CH0取值为0000~1011时，对应表格如下：

[BU,DSI]	CH 3	CH 2	CH 1	CH 0	AD_SELA<3: 0>	AD_SELB<3: 0>	AD_MODE_DIFF_ EN	
X0 (single-en d)	0	0	0	0	4'd0	4'd12	0	
	0	0	0	1	4'd1			
	0	0	1	0	4'd2			
	0	0	1	1	4'd3			
	0	1	0	0	4'd4			
	0	1	0	1	4'd5			
	0	1	1	0	4'd6			
	0	1	1	1	4'd7			
	1	0	0	0	4'd8			
	1	0	0	1	4'd9			
	1	0	1	0	4'd10			
	1	0	1	1	4'd11			
01 (unipolar)	0	0	0	0	4'd1	4'd0	0	
	0	0	0	1				
	0	0	1	0	4'd3	4'd2		
	0	0	1	1				
	0	1	0	0	4'd5	4'd4		
	0	1	0	1				
	0	1	1	0	4'd7	4'd6		
	0	1	1	1				
	1	0	0	0	4'd9	4'd8		
	1	0	0	1				
	1	0	1	0	4'd11	4'd10		
	1	0	1	1				
11 (bipolar)	0	0	0	0	4'd1	4'd0	1	
	0	0	0	1				
	0	0	1	0	4'd3	4'd2		
	0	0	1	1				
	0	1	0	0	4'd5	4'd4		
	0	1	0	1				
	0	1	1	0	4'd7	4'd6		

	0	1	1	1				
	1	0	0	0				
	1	0	0	1				
	1	0	1	0				
	1	0	1	1				

根据上表，可以作如下总结：

当BU/DS的设置为single-end (00或10) 模式时，AD_SELA<3:0>的取值即为对应CH3~CH0的值，AD_SELB<3:0>设置为4'd12 (选择VSS为负向端)；

当BU/DS的设置为unipolar/bipolar (01或11) 模式时，channel 0和channel 1、channel 2和 channel 3、...、channel 10和channel 11会组成pair，例如，当CH3~CH0设置为0000或0001时，即选择channel 0和channel 1这个pair，依此类推。

CH3:CH0取值为1100~1111时，对应表格如下：

CH 3	CH 2	CH 1	CH 0	[BU,DS]	AD_SELA<3: 0>	AD_SELB<3: 0>	AD_MODE_DIFF_ EN
1	1	0	0				
1	1	0	1	XX	4'd13	4'd12	0
1	1	1	0				
1	1	1	1	XX	4'd15	4'd14	0

在此情况下，不论BU/DS如何取值，AD_SELA和AD_SELB的值是固定的。

c) 扫描控制寄存器

扫描控制寄存器 (02h to 03h) 用来对sequence模式下的ADC进行配置与控制。其中sequence register0用来对sequence的通道进行选择，sequence register1控制通道的工作模式 (diff和single-end、 unipolar和bipolar)。寄存器的各个bit的分配如图。

Sequence Reg#0 ADDR<4:0>=02h	DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
	0	0	0	0	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0
Sequence Reg#1 ADDR<4:0>=03h	DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
	0	0	0	0	BUS5	DS5	BU4	DS4	BU3	DS3	BU2	DS2	BU1	DS1	BU0	DS0

图5-24 扫描控制寄存器bit分配

其中DS<0:5>是分别对VAUX<0>/VAUX<1>、 VAUX<2>/VAUX<3>、 VAUX<4>/VAUX<5>…VAUX<8>/VAUX<9>进行single-end和diff模式选择。0:

single-end; 1: diff。

在前面选择为diff模式的前提下，BU<0:5>是分别对VAUX<0>/VAUX<1>、VAUX<2>/VAUX<3>、VAUX<4>/VAUX<5>…VAUX<8>/VAUX<9>进行diff的形式选择。0: unipolar; 1: bipolar。

在single-end模式下，一对通道如VAUX<0>和VAUX<1>分别工作在single-end模式；在unipolar模式下，VAUX<0>和VAUX<1>组成pair，VAUX<0>为N端，VAUX<1>为P端；在bipolar模式下VAUX<0>和VAUX<1>组成差分对。

Name (sequence reg2)	控制的通道	Description	AD_MODE_DIFF_EN的取值
BU0/DS0	VAUX<0>/VAUX<1>	X0: single-end 01: unipolar (VAUX<0>/VAUX<1>组成一对) 11: bipolar (VAUX<0>/VAUX<1>组成一对)	X0: 0 01: 0 11: 1
BU1/DS1	VAUX<2>/VAUX<3>	X0: single-end 01: unipolar 11: bipolar	同上
BU2/DS2	VAUX<4>/VAUX<5>	X0: single-end 01: unipolar 11: bipolar	同上
BU3/DS3	VAUX<6>/VAUX<7>	X0: single-end 01: unipolar 11: bipolar	同上
BU4/DS4	VAUX<8>/VAUX<9>	X0: single-end 01: unipolar 11: bipolar	同上
BU5/DS5	VA<0>/VA<1>	X0: single-end 01: unipolar 11: bipolar	同上

在single-end模式下

Name (sequence reg0)	Description	实时AD_SELA<3:0> AD_SELB<3:0>	

C0	控制输入通道VAUX<0> C0控制是否扫描, 0: disable, 1: enable	AD_SELA<3:0>=2'b0000 AD_SELB<3:0>=2'b1100	
C1	控制输入通道VAUX<1> C1控制是否扫描, 0: disable, 1: enable	AD_SELA<3:0>=2'b0001 AD_SELB<3:0>=2'b1100	
...	...		
C9	控制输入通道VAUX<9> C10控制是否扫描, 0: disable, 1: enable	AD_SELA<3:0>=2'b0010 AD_SELB<3:0>=2'b1100	
C10	控制输入通道VA0 C10控制是否扫描, 0: disable, 1: enable	AD_SELA<3:0>=2'b1010 AD_SELB<3:0>=2'b1100	
C11	控制输入通道VA1 C11控制是否扫描, 0: disable, 1: enable	AD_SELA<3:0>=2'b1011 AD_SELB<3:0>=2'b1100	

在diff模式下

Name(sequence reg0)	Description	实时AD_SELA<3:0> AD_SELB<3:0>	
C0/C1	控制输入通道 VAUX<0>/VAUX<1> C0/C1控制是否扫描, 00/01/10: disable, 11: enable	AD_SELA<3:0>=2'b0001 AD_SELB<3:0>=2'b0000	
C2/C3	控制输入通道 VAUX<2>/VAUX<3> C2/C3控制是否扫描, 00/01/10: disable, 11: enable	AD_SELA<3:0>=2'b0011 AD_SELB<3:0>=2'b0010	
...	...		
C8/C9	控制输入通道 VAUX<8>/VAUX<9> C8/C9控制是否扫描, 00/01/10: disable, 11: enable	AD_SELA<3:0>=2'b1001 AD_SELB<3:0>=2'b1000	

C10/C11	控制输入通道VA<0>/VA<1> C10/C11控制是否扫描， 00/01/10: disable, 11: enable	AD_SELA<3:0>=2'b1011 AD_SELB<3:0>=2'b1010	
---------	--	--	--

d) 温度检测控制寄存器

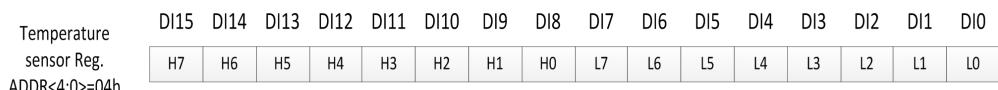


图 5- 25 扫描控制寄存器 bit 分配

温度检测控制寄存器（04h）用来对默认模式下温度检测阈值（上限与下限）进行配置与控制。其中高8bit (DI15~DI8) 设置温度检测上限，低8bit (DI7~DI0) 设置温度检测下限。

设置为0时对应0K (-273.15度)，阈值设置step为5.8K (16*0.364K)。

常用设置值如下：

	code	温度
DI15:DI8	8'b01000101	128度
DI7:DI0	8'b000111000	53度

e) 状态寄存器

状态寄存器 (status registers, 05h to 18h) 用来存放各通道的转换结果和用来 calibration 的 offset 和 gain error 值 (其中 05h 为 flag reg)。所有的状态寄存器，用户都只可以通过 DRP 进行读取，不能进行写操作。状态寄存器 06h~11h 的定义分 single-end 模式和 diff 模式，single-end 模式如下表所示。

Name	Address	Description
VAUX<0:9>	06h to 0Fh	存放了 VAUX<0:9>的转换值
VA0	10h	存放了 VA0 的转换值
VA1	11h	存放了 VA1 的转换值

Diff 模式如下表所示

Name	Address	Description
VAUX<1>/ VAUX<0>	06h to 07h	存放了 VAUX<1>/ VAUX<0>的差分转换值

VAUX<3>/ VAUX<2>	08h to 09h	存放了VAUX<3>/ VAUX<2>的差分转换值
VAUX<5>/ VAUX<4>	0Ah to 0Bh	存放了VAUX<5>/ VAUX<4>的差分转换值
VAUX<7>/ VAUX<6>	0Ch to 0Dh	存放了VAUX<7>/ VAUX<6>的差分转换值
VAUX<9>/ VAUX<8>	0Eh to 0Fh	存放了VAUX<9>/ VAUX<8>的差分转换值
VA1/VA0	10h to 11h	存放了VA1/VA0的差分转换值

其它registers的定义：

Name	Address	Description
SU OFFSET	12h	存放了 single-end 和 unipolar 模式的 ADC 的 offset 校准值
BI OFFSET	13h	存放了 bipolar 模式的 ADC 的 offset 校准值
GAIN	14h	存放了 Gain error 的校准值
VDD33	15h	存放了电源 VDD33 的转换值
VDD11	16h	存放了电源 VDD11 的转化值
VDDM	17h	存放了电源 VDDM 的转化值
Temperature	18h	存放了检测片上温度的转换值

5.9.2 Logos2、Titan2 系列

目前Logos2系列PG2L100H,PG2L25H器件以及Titan2系列PG2T390H器件支持ADC功能，共用同一套寄存器。

a) 控制寄存器

控制寄存器主要有以下功能：

1. 用于设定ADC的工作模式，如时钟频率，ADC_A/B开关情况，alarm指示信号开关。
2. 用于选定扫描的channel，扫描对应的bipolar或unipolar模式，以及是否使能average。
3. 用于设定电压和片上温度的最大阈值和最小阈值。

可配置寄存器详细见下表：

Address	Description
---------	-------------

00h	creg_00h[15:0]														
01h	creg_01h[15:0]														
02h	creg_02h[15:0]														
03h	seq_03h[15:0]														
04h	seq_04h[15:0]														
05h	seq_05h[15:0]														
06h	seq_06h[15:0]														
07h	seq_07h[15:0]														
08h	seq_08h[15:0]														
09h	seq_09h[15:0]														
10h	seq_0Ah[15:0]														
12h	seq_0Ch[15:0]														
14h	seq_0Eh[15:0]														
20-2Bh	alm_20h~alm_2Bh[15:0]														
30h	jtag_abp_30h[15:0]														
31h	creg_31h[13:0]														

表 10 寄存器地址描述

DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	
creg_00h[15:0]	PD1	PDO	CLKSW	SEQ	RDATA		DB2	DB1	DB0		ALM4	ALM3	ALM2	ALM1	ALMO	OT
creg_01h[15:0]	CAL1	CAL0		Avg1	AVG0				DIV7	DIV6	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
creg_02h[15:0]	VCM1	VCM0	VREF	SCALE1	SCALE0		INPUT RES_A1	INPUT RES_A0		INPUT RES_B1	INPUT RES_B0	E_B_VPN	C1k gen	N_SEQ	CE	E_OSC
seq_03h[15:0]	AC15	AC14	AC13	AC12	AC11	AC10	AC9	AC8	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
seq_04h[15:0]	AXC15	AXC14	AXC13	AXC12	AXC11	AXC10	AXC9	AXC8	AXC7	AXC6	AXC5	AXC4	AXC3	AXC2	AXC1	AXC0
seq_05h[15:0]	ABU15	ABU14	ABU13	ABU12	ABU11	ABU10	ABU9	ABU8	ABU7	ABU6	ABU5	ABU4	ABU3	ABU2	ABU1	ABU0
seq_06h[15:0]	AXB15	AXB14	AXB13	AXB12	AXB11	AXB10	AXB9	AXB8	AXB7	AXB6	AXB5	AXB4	AXB3	AXB2	AXB1	AXB0
seq_07h[15:0]	ACAV15	ACAV14	ACAV13	ACAV12	ACAV11	ACAV10	ACAV9	ACAV8	ACAV7	ACAV6	ACAV5	ACAV4	ACAV3	ACAV2	ACAV1	ACAV0
seq_08h[15:0]	AXCAV15	AXCAV14	AXCAV13	AXCAV12	AXCAV11	AXCAV10	AXCAV9	AXCAV8	AXCAV7	AXCAV6	AXCAV5	AXCAV4	AXCAV3	AXCAV2	AXCAV1	AXCAV0
seq_0Ah[15:0]	BXC15	BXC14	BXC13	BXC12	BXC11	BXC10	BXC9	BXC8	BXC7	BXC6	BXC5	BXC4	BXC3	BXC2	BXC1	BXC0
seq_0Ch[15:0]	BXB15	BXB14	BXB13	BXB12	BXB11	BXB10	BXB9	BXB8	BXB7	BXB6	BXB5	BXB4	BXB3	BXB2	BXB1	BXB0
seq_0Eh[15:0]	BXCAV15	BXCAV14	BXCAV13	BXCAV12	BXCAV11	BXCAV10	BXCAV9	BXCAV8	BXCAV7	BXCAV6	BXCAV5	BXCAV4	BXCAV3	BXCAV2	BXCAV1	BXCAV0
alm_20h~2Bh [12:0]	Alarm threshold registers															
JTAG_ABH_30h [15:0]	Jlock	Jbusy	Jreq								Alock	Abusy	Areq			
creg_31h[15:0]			IBIAS13	IBIAS12	IBIAS11	IBIAS10	IBIAS9	IBIAS8	IBIAS7	IBIAS6	IBIAS5	IBIAS4	IBIAS3	IBIAS2	IBIAS1	IBIAS0

图 5-26 control registers 组成&对应功能

各个控制寄存器详细描述如下表所示：

Bit	Name	Description
creg_00h		
DI0	OT	over temperature 使能标志位，高电平有效
DI1 to DI5	ALM0 to ALM4	alarm 输出使能控制位： 0 : enable the alarm 1 : disable the alarm
DI7 to DI9	DB0 to DB2	设置 over_temp 信号 debounce 时的计数器数值
DI11	RDATA	不进行 error correction 过程的标志位，即将 raw data 直接 store 在 status registers 中。 Temperature&supply sensor 不支持此功能。 1 有效
DI12	SEQ	Sequence 模式选择位： 1 : Scan Sequence Mode 0 : Power Up Mode
DI13	CLKSW	时钟转换控制位，控制 clkswitch 信号， 1 : 系统选择 pclk 用于时钟分频产生 ad_clk 0 : 系统选择 clk_osc 用于时钟分频产生 ad_clk
DI14 to DI15	PD0 to PD1	power-down 控制位
creg_01h		
DI0 to DI7	DIVA0 to DIVA7	作用于 ADC 的时钟分频信号
DI11 to DI12	AVG0 to AVG1	ADC 平均次数控制位
DI14 to DI15	CAL0 to CAL1	calibration 配置位
creg_02h		
DI0	E_OSC	clk_osc 开关使能信号。

		0 : 根据系统的 clkswitch 和 PDO 控制位来控制 clk_osc 的开关 ; 1 : clk_osc 常开 , 不受其他信号影响。
DI1	CE	连续采样模式和主动采样模式选择位 0 : Continuous 1 : event-drive
DI2	N_SEQ	处于主动控制状态下时 , 用于产生扫描信号的控制位。
DI3	CLKGEN	Clock generate 使能信号
DI4	E_B_VPN	ADC_B 扫描 vp/vn 通道的使能信号 : 0 : seq_03h/05h/07h[14] 控制 ADC_A 对 VP/VN 通道的扫描情况 ; 1 : seq_03h/05h/07h[14] 控制 ADC_B 对 VP/VN 通道的扫描情况。
DI5 to DI6	INPUT_RESB[0] , INPUT_RESB[1]	For test; ADC B 辅助端口输入电阻配置: 00: 0 ; 01: R ; 10:2R ; 11:3R R=864Ohm Default:00 档位预留 , 现模拟固化 , 不支持调节。
DI8 to DI9	INPUT_RESA[0] , INPUT_RESA[1]	For test; ADC A 辅助端口输入电阻配置: 00:0 ; 01: R ; 10:2R ; 11:3R R=864Ohm Default:00 档位预留 , 现模拟固化 , 不支持调节。
DI11 to DI12	SCALE[0] ,SCALE[1]	FS 范围档位控制

		00:FS=1V , VREF_CM=0.65V; 01: FS=0.9V , VREF_CM=0.675V; 10: FS=0.8V , VREF_CM=0.6V; 11: FS=0.8V , VREF_CM=0.6V; Default : 00 档位预留，现模拟固化，不支持调节。
DI13	VREF	对 ADC 的参考源进行选择 0 : 内部参考源 (BGP 产生) 1 : 外部参考源 PS : power up 模式下选择内部参考源 , trim mode 模式下可以配置
D14 to D15	VCM0、VCM1	共模电压控制位 (详情见表 19)
Creg_09h		
DI11 to DI0	IBIAS[13:0]	偏置电流档位控制 (详情见表 12)
Seq_03h Seq_04h , Seq_0Ah	Enable channel: 1 enable; 0 disable (ADC_B 不扫描内部 temp&voltage)	
Seq_05h seq_06h seq_0Ch	Choose scan mode: 1 bipolar; 0 unipolar	
Seq_07h seq_08h Seq_0Eh	Enable averaging: 1 enable; 0 disable (if disable, take the last sample value)	
Alm_20h	The threshold value of Temperature upper	
Alm_21h	The threshold value of Temperature lower	
Alm_22h	The threshold value of VCC upper	
Alm_23h	The threshold value of VCC lower	
Alm_24h	The threshold value of VCCA upper	
Alm_25h	The threshold value of VCCA lower	
Alm_26h	The threshold value of VCC_CRAM upper	
Alm_27h	The threshold value of VCC_CRAM lower	
Alm_28h	The threshold value of VCC_DRM upper	

Alm_29h	The threshold value of VCC_DRM lower
Alm_2Ah	The threshold value of OT alarm limit
Alm_2Bh	The threshold value of OT alarm reset

表 11 configuration registers 各个控制位功能列表

IBIAS_CTRL<11:0>	IBIAS_CTRL<1:0>	IBIAS_SH_ADC_A/B
	00(default)	80uA
	01	120uA
	10	60uA
	11	100uA
	IBIAS_CTRL<3:2>	IBIAS_TEMP_SENSOR_AMP
	00(default)	10uA
	01	15uA
	10	7.5uA
	11	12.5uA
	IBIAS_CTRL<5:4>	IBIAS_CMP_A/B
	00(default)	10uA
	01	15uA
	10	7.5uA
	11	12.5uA
	IBIAS_CTRL<7:6>	IBIAS_VREF_BUF_A/B
	00(default)	10uA
	01	15uA
	10	7.5uA
	11	12.5uA
	IBIAS_CTRL<9:8>	IBIAS_VCM_BUF_A/B
	00(default)	10uA
	01	15uA
	10	7.5uA
	11	12.5uA
	IBIAS_CTRL<11:10>	IBIAS_VREF_GEN
	00(default)	10uA
	01	15uA
	10	7.5uA
	11	12.5uA
	IBIAS_CTRL<13:12>	IBIAS_TEMP_SENSOR_CORE

	00(default)	10uA
	01	15uA
	10	7.5uA
	11	12.5uA

表 12 电流配置位信息

DIV7	DIV6	DIV5	DIV4	DIV3	DIV2	DIV1	DIVA0	clock divide ratio
0	0	0	0	0	0	0	0	2
0	0	0	0	0	0	0	1	2
0	0	0	0	0	0	1	0	4
0	0	0	0	0	0	1	1	4
0	0	0	0	0	1	0	0	6
0	0	0	0	0	1	0	1	6
...
1	1	1	1	1	1	0	0	254
1	1	1	1	1	1	0	1	254
1	1	1	1	1	1	1	0	256
1	1	1	1	1	1	1	1	256

表 13 时钟分频控制

PD1	PD0	Description
0	0	Default. All blocks powered up
0	1	N/A
1	0	ADC B powered down
1	1	All blocks powered down

表 14 Power-down 控制

ALM0	Alarm for Temperature
ALM1	Alarm for VCC
ALM2	Alarm for VCCA
ALM3	Alarm for VCC_CRAM
ALM4	Alarm for VCC_DRM

表 15 alarm 使能控制位

AVG1	AVG0	Function
0	0	No averaging
0	1	average 16 samples
1	0	average 64 samples
1	1	average 256 samples

表 16 Averaging 控制

CAL1	CAL0	Description
0	0	ADCs offset and gain correction not enable
0	1	ADCs offset correction enable
1	0	ADCs gain correction enable; only for debug mode; not valid for custum;
1	1	ADCs offset and gain correction enable

表 17 Calibration 控制

Clkswitch	DB0 to DB2	Counter	corresponding clk cycle of clk_mux
1	000	20'h 03000	1000ns > clk cycle \geq 850ns
1	001	20'h 04000	850ns > clk cycle \geq 650ns
1	010	20'h 05000	650ns > clk cycle \geq 500ns
1	011	20'h 09000	500ns > clk cycle \geq 300ns
1	100	20'h 20000	300ns > clk cycle \geq 150ns
1	101	20'h 40000	150ns > clk cycle \geq 50ns
1	110	20'h F5000	50ns > clk cycle \geq 10ns
1	111	N/A	N/A
0	XXX	20'h 80000	20ns

表 18 over_temp 信号 debounce 时的计数器

VCM1	VCM0	Description
0	0	VCM=0.65V
0	1	VCM=0.6V
1	0	VCM=0.55V
1	1	VCM=0.7V

表 19 共模电压控制位

seq_03h	seq_05h	seq_07h	Corresponding object
	0		carry out the ADC calibration(code 0)
	1		carry out the ADC calibration(code 1)
	2		Temperature
	3		VCC
	4		VCCA
	5		VCC_CRAM
	6		VCC_DRM
	7		TRIM_V1
	8		TRIM_V2
	9		VTBD3 (not used)
	10		VTBD4 (not used)
	11		VTBD5 (not used)
	12		VTBD6 (not used)
	13		VTBD7 (not used)
	14		VP/VN
seq_04h &seq_0Ah	seq_06h &seq_0Ch	seq_08h &seq_0Eh	Corresponding object
	0		VAAP[0], VAAN[0] —— Auxiliary channel 0
	1		VAAP[1], VAAN[1] —— Auxiliary channel 1
	2-15		VAAP[2:15], VAAN[2:15] —— Auxiliary channel 2-15

表 20 每个 bit 对应的对象

每次复位上电后，存储最小值的寄存器将会复位到 FFFFh，存储最大值得寄存器将会复位到0000h。每一个待测信号被检测后，都要与设定的阈值进行比较。如果检测后的值大于阈值的上限值时，检测后的值将被存入status registers中对应的最大值寄存器中。同理，检测值小于阈值的下限值时，也被存入status registers中对应的最小值寄存器中。

b) 状态寄存器

状态寄存器主要用于存储各通道的转换结果，calibration中的offset和gain的值，以及片上传感器的检测到的最大值和最小值，状态寄存器是只读的。

状态寄存器中的各个地址位的定义：

Address	Description
40h	Storing the convert value of temperature
41h	Storing the convert value of VCC
42h	Storing the convert value of VCCA
43h	Storing the convert value of VCC_CRAM
44h	Storing the convert value of VCC_DRM
45h	Storing the convert value of VP/VN
46h	Storing the convert value of calibration code0
47h	Storing the convert value of calibration code1
48h	Storing the convert value of ADC_A offset(unipolar)
49h	Storing the convert value of ADC_A offset(bipolar)
4Ah	Storing the convert value of ADC_A gain(unipolar)
4Bh	Storing the convert value of ADC_A gain(bipolar)
4Ch	Storing the convert value of ADC_A supply sensor offset(unipolar)
50-5Fh	Storing the convert value of VAAP[0:15]/VAAN[0:15] — Auxiliary channel 0-15
60h	Storing the convert value of ADC_B offset(unipolar)
61h	Storing the convert value of ADC_B offset(bipolar)
62h	Storing the convert value of ADC_B gain(unipolar)
63h	Storing the convert value of ADC_B gain(bipolar)
65h	Storing the maximum convert value of Temperature since the last power up or last reset
66h	Storing the maximum convert value of VCC since the last power up or last reset
67h	Storing the maximum convert value of VCCA since the last power up or last reset
68h	Storing the maximum convert value of VCC_CRAM since the last power up or last reset
69h	Storing the maximum convert value of VCC_DRM since the last power up or last reset
6Ah	Storing the minimum convert value of Temperature since the last power up or

	last reset
6Bh	Storing the minimum convert value of VCC since the last power up or last reset
6Ch	Storing the minimum convert value of VCCA since the last power up or last reset
6Dh	Storing the minimum convert value of VCC_CRAM since the last power up or last reset
6Eh	Storing the minimum convert value of VCC_DRM since the last power up or last reset
70h	Storing the convert value of TRIM_V1 from ADC_A
71h	Storing the convert value of TRIM_V2 from ADC_A
72h	Storing the convert value of VTBD3 from ADC_A
73h	Storing the convert value of VTBD4 from ADC_A
74h	Storing the convert value of VTBD5 from ADC_A
75h	Storing the convert value of VTBD6 from ADC_A
76h	Storing the convert value of VTBD7 from ADC_A
77-7Fh	Undefined

表 21 PG2L100H 状态寄存器地址定义

5.10 波形搜索

在 Waveform 界面按快捷键 Ctrl+F 调出搜索框，或者选中信号，右键菜单点击搜索， 波形搜索界面如下图。

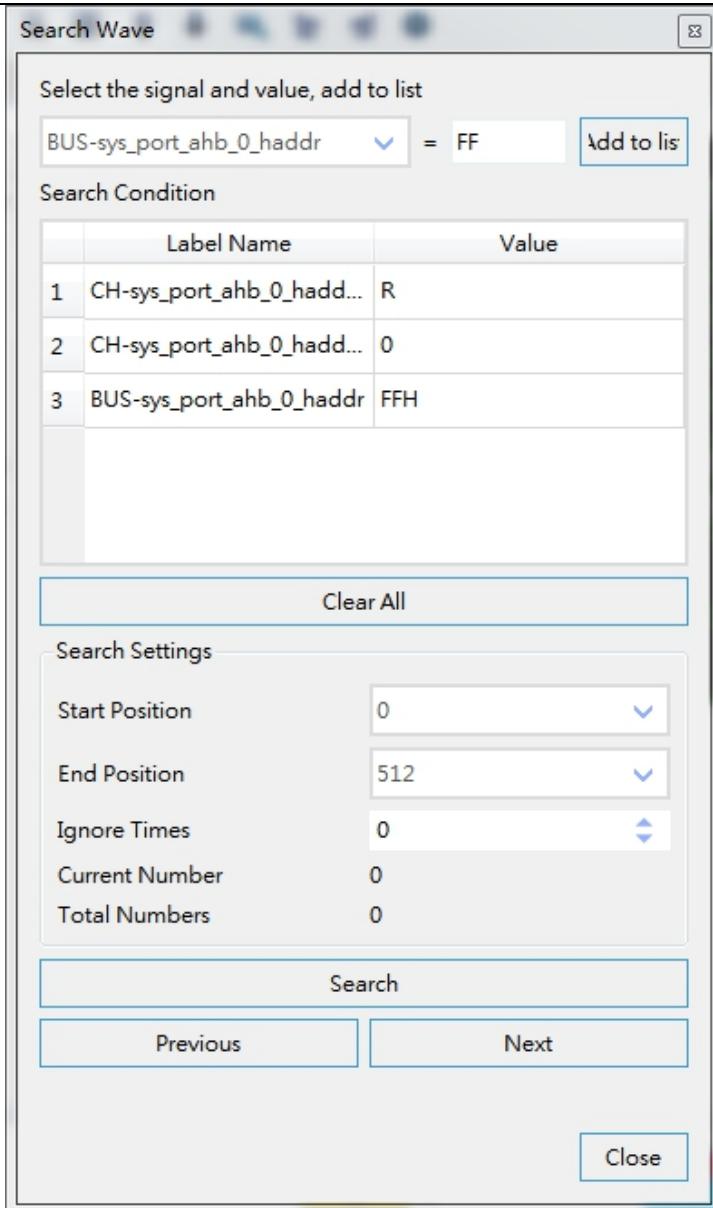


图 5-27 波形搜索界面

搜索信号分为单通道信号和总线信号

单通道信号：选择单条数据信号，选择对应的搜索条件，点击 add to list 添加到搜索条件列表，对应的值选项表示如下：

- 1) 0: 0 值
- 2) 1: 1 值
- 3) R: 上升沿
- 4) F: 下降沿

总线信号：选择总线信号，文本框输入对应信号的十六进制格式，点击 add to

list 添加到搜索条件列表

1、添加信号

选中信号和值，点击 add to list 添加该条件到搜索条件列表

2、更新信号

重新选择该信号和值，点击 add to list，搜索条件列表会自动更新条件

3、删除信号

搜索条件列表选择若干列，右键菜单点击删除

搜索设置选项

- 1) Start Position: 搜索采样点起始位置
- 2) End Position: 搜索采样点结束位置
- 3) Ignore Times: 忽略次数，即连续满足所有搜索条件的次数
- 4) Current Number: 当前光标所在搜索结果的序列号
- 5) Total Number: 上次搜索满足搜索条件的总个数

Clear All: 清除搜索条件列表

Search: 按照已选择的搜索条件进行与搜索，查找同时满足所有条件的采样点

Previous: 通过 X 光标定位，跳转到上一处满足条件的搜索结果

Next: 通过 X 光标定位，跳转到下一处满足条件的搜索结果

5.11 波形分析

波形测量与分析, 测量工具栏如下:

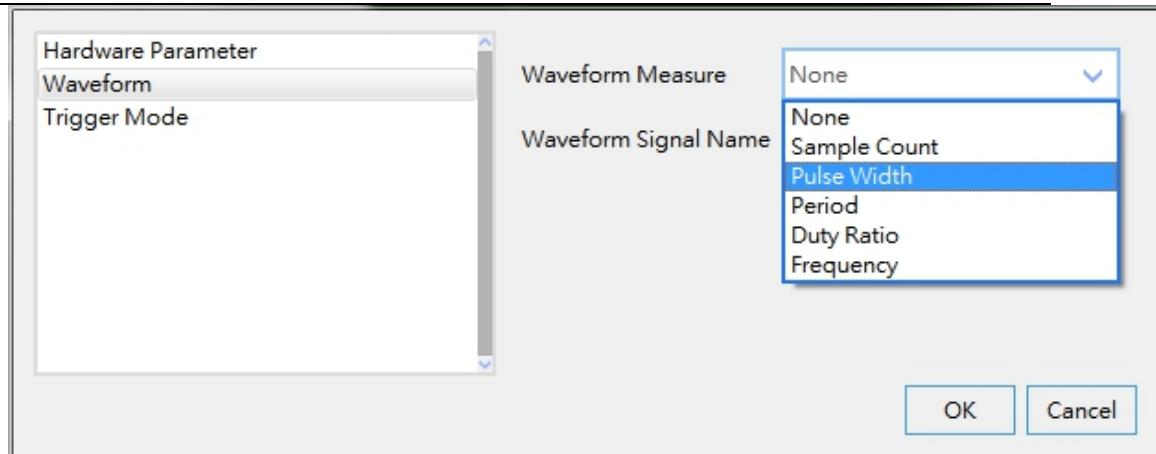


图 5- 28 波形测量模式设置

从上到下功能分别为:

- Measure None: 无测量信息
- Measure Count: 测量采样点数
- Measure Pulse: 测量脉宽时间 (显示每个脉宽时间)
- Measure Period: 测量周期时间 (显示周期时间, 两个上升沿为一个完整周期)
- Measure Frequency: 测量频率 (显示频率, 两个上升沿为一个完整周期)

5.12 系统设置菜单

- 1) 波形测量值根据硬件采样率计算 Settings->SystemSettings->Hardware Parameter 界面, Hardware Sample Rate 设置硬件采样率.

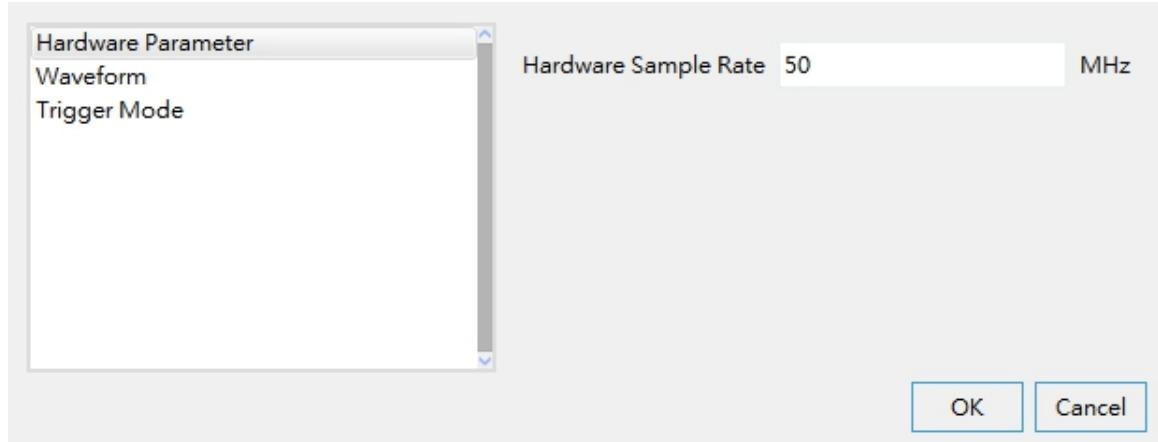


图 5- 29 硬件采样率设置界面

2) Waveform 界面, 可以设置测量方式, 并且支持设置波形数据总线显示格式。

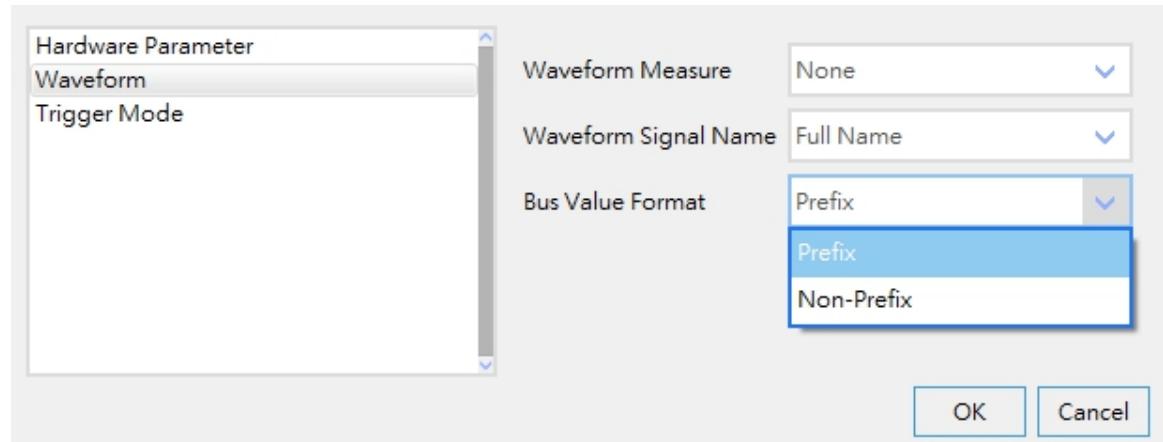


图 5- 30 waveform 测量方式设置界面

总线数值格式设置说明:

- (1) 通过设置下拉框 “System Setting->Waveform->Bus Value Format” 的值, Waveform 和 Listing 界面总线的数据将显示为同一种格式。
- (2) 根据所选 Radix (Binary, Octal, Hex) 的不同, 总线数据显示对应的前缀 (0b, 0o, 0x)。特别地, 十进制不显示前缀。
- (3) 默认状态下, “System Setting->Waveform->Bus Value Format”中选择 "Prefix", 即总线数据默认显示带前缀的数值, 且 Waveform 界面上, "Bus/Signal Tree"中显示 X、O 位置的值为当前所选进制的完整位数, 如, 总线值为 “0x01”, "Bus/Signal Tree"中 X、O 位置的值显示 “8'h01”。
- (4) 如果“System Setting->Waveform->Bus Value Format”中选择"Non-Prefix", Waveform 和 Listing 界面中所有的总线数值切换为不带前缀,"Bus/Signal Tree"中显示 X、O 位置的值为当前所选进制的不为 0 的位数, 如, 总线值为 “1” , "Bus/Signal Tree" 中 X、O 位置的值显示 “8'h1”。
- (5) 当切换数据的显示格式时, 链中所有设备, 所有 Core 的 Waveform 和 Listing 界面的总线数据格式都将被切换。

3) Trigger Mode 波形连续触发保存功能

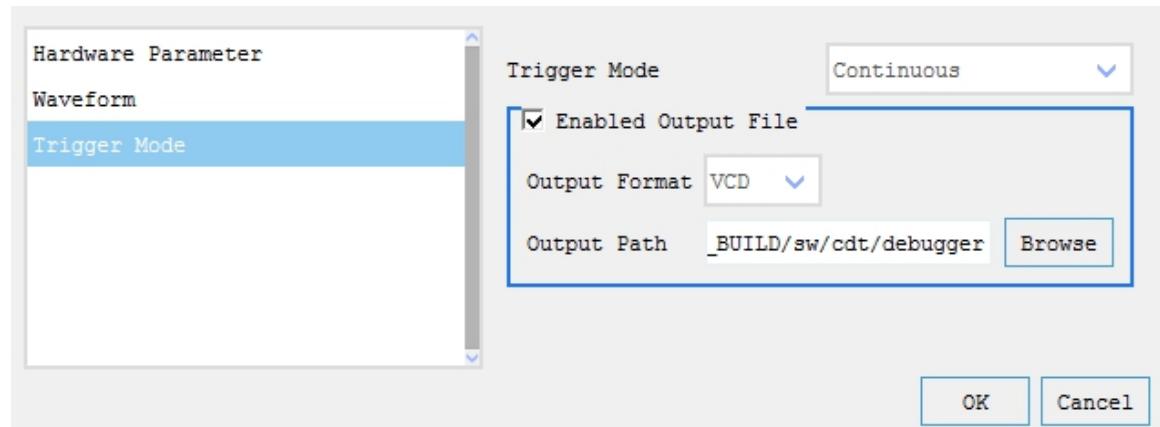


图 5- 31 波形触发设置界面

触发模式：

Single：单次触发，不保存

Continuous：自动连续触发，直到执行出错或者手动 stop。

Enabled OutPut File：使能输出文件功能，只在 Continuous 模式下生效，勾选该项后，会把每次触发的波形数据，输出到指定的路径的指定文件格式（VCD/ASCII/WDF）的文件中。

5.13 版本匹配功能

Debugger 与 jtag server 连接时，自动匹配版本信息（匹配 suit 版本，pro 版本还会匹配 svn 版本信息），不匹配则不允许与 server 通信。

5.14 多核同时调试功能

目前 Debugger 支持以下场景下多核调试功能：

1. 同一器件下，多个核同时执行单次触发
2. 同一器件下，单个核连续触发，其余核单次触发

不同器件下的各个核不能同时触发，且同一时间内只能有一个核连续触发。

5.15 iScan

5.15.1 总体介绍

串行通信技术由于传输过程中噪声等因素的影响，串行信号可能会出现码间串扰，信号畸变等问题。眼图作为一种直观，快速地分析串行信号质量的方法，可衡量通信系统性能的优劣。一般而言，接收端信号的眼图是通过在发送端发送随机比特流，在接收端使用示波器进行观察所得的，但在高速传输链路中，使用示波器测量眼图的成本十分昂贵。因此，在现代高速传输系统中，一般在发送端使用片上伪随机比特发生器来产生比特流，在接收端使用片上眼开监视器来获取眼图，测量误码率，以此降低测试成本。

iScan 可以对高速信号眼图的水平与垂直打开大小进行测量，并将结果保存在寄存器中，通过对寄存器的读取得到眼图打开大小信息，进而可观测到信号传输误码率的大小。

5.15.2 功能描述

下载带有眼图功能的位流后，点击“扫链”将会出现如下图所示界面：

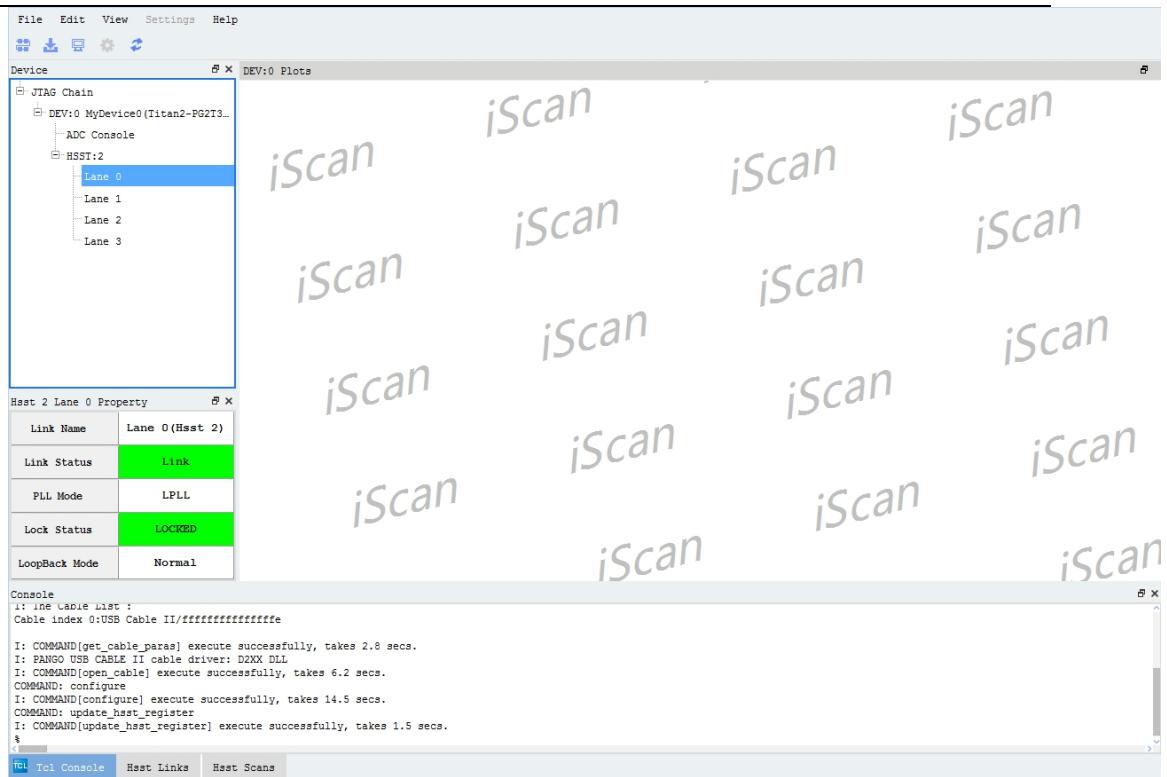


图 5- 32 iScan 主界面

在主界面左侧上方，显示的是当前位流包含的 HSST 列表，目前最多支持 4 个 HSST (HSST1~HSST4)。每个 HSST 最多有 4 个 Lane (Lane0~Lane3)，点击选中某条 Lane，左侧下方会对应显示该条通道属性。下方控制台包含 Hsst Links 和 Hsst Scans 功能窗口，用于创建通道、扫描眼图等操作。

当需要扫描眼图时，第一步需要先在“Hsst Links”界面创建链路通道，右键点击空白处弹出如下菜单选项：

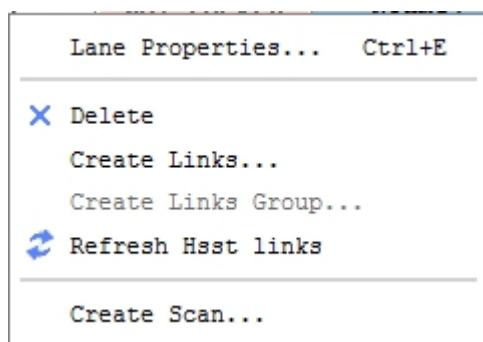


图 5- 33 Hsst Links 右键菜单

菜单选项对应操作说明如下：

菜单项	功能
-----	----

Lane Properties	在属性栏中显示当前通道的属性
Delete	删除选中的通道
Create Links	打开 Create Links 对话框, 创建通道
Create Links Group	选中多项 link items 创建 link group
Refresh Hsst Links	刷新 Hsst links
Create Scan	打开 Create Scan 对话框, 创建扫描

选择 “Create Links...” 创建通道，此时会弹出如下对话框：

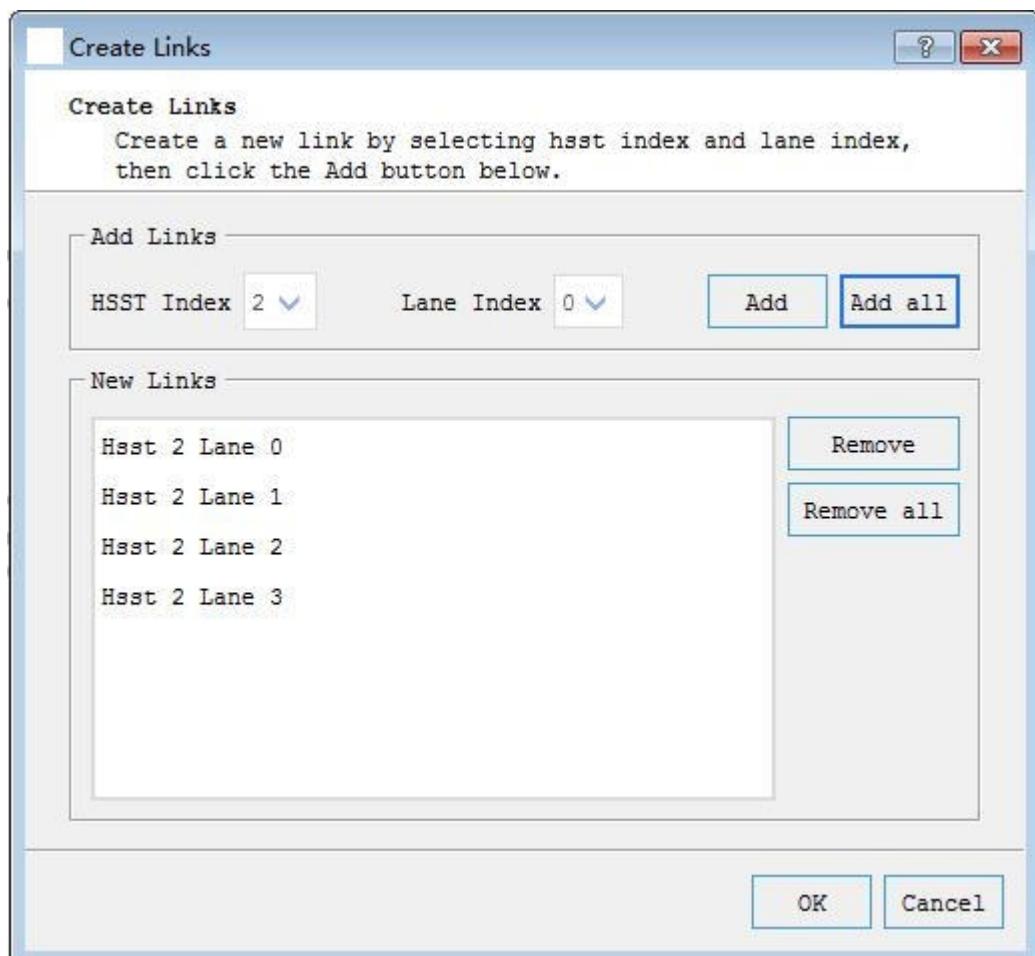


图 5-34 Create Links 对话框

在“Add Links”，通过设置 HSST 序号和 Lane 序号指定想要创建的通道，点击“Add”按钮将其添加到“New Links”栏中，或点击“Add all”将该器件所有有效的通道添加至“New Links”栏中，并可以通过“Remove”、“Remove all”移除“New Links”栏中的通道。在添加完毕后点击“OK”即可成功创建通道。创建成功后“Hsst Links”窗口如下图所示：

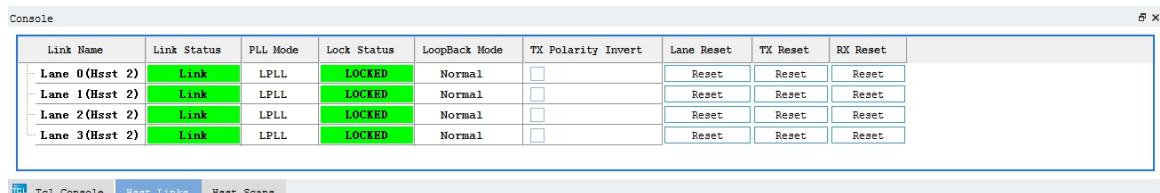


图 5- 35 Hsst Links 窗口

界面从左到右各项属性说明如下：

功能名称	说明
Link Name	显示通道名称
Link Status	显示 HSST 各个 Lane 的连接状态 (No Link/Link)
PLL Mode	显示当前通道锁向环类型： LPPLL/HPLL
Lock Status	显示 LPPLL 或者 HPLL 各个 Lane 的锁定状态 (Not Locked/ Locked)
LoopBack Mode	环回模式配置，包括 Near-End PCS PLOOP、Near-End PMA SLOOP、Near-End PMA PLOOP、Far-End PMA PLOOP、Far-End PCS PLOOP、Normal 六种模式。
TX Polarity Invert	TX 极性反转，勾选表示配置位反转。
Lane Reset	Lane 复位按钮，对当前 TX lane 与 RXlane 同时进行复位
TX Reset	TX 复位按钮，可对每个 Lane 进行 TX 复位操作。
RX Reset	RX 复位按钮，可对每个 Lane 进行 RX 复位操作。

在创建完通道后右键选中某条通道，选择“Create Scan...”创建扫描，

打开创建扫描对话框如下所示：

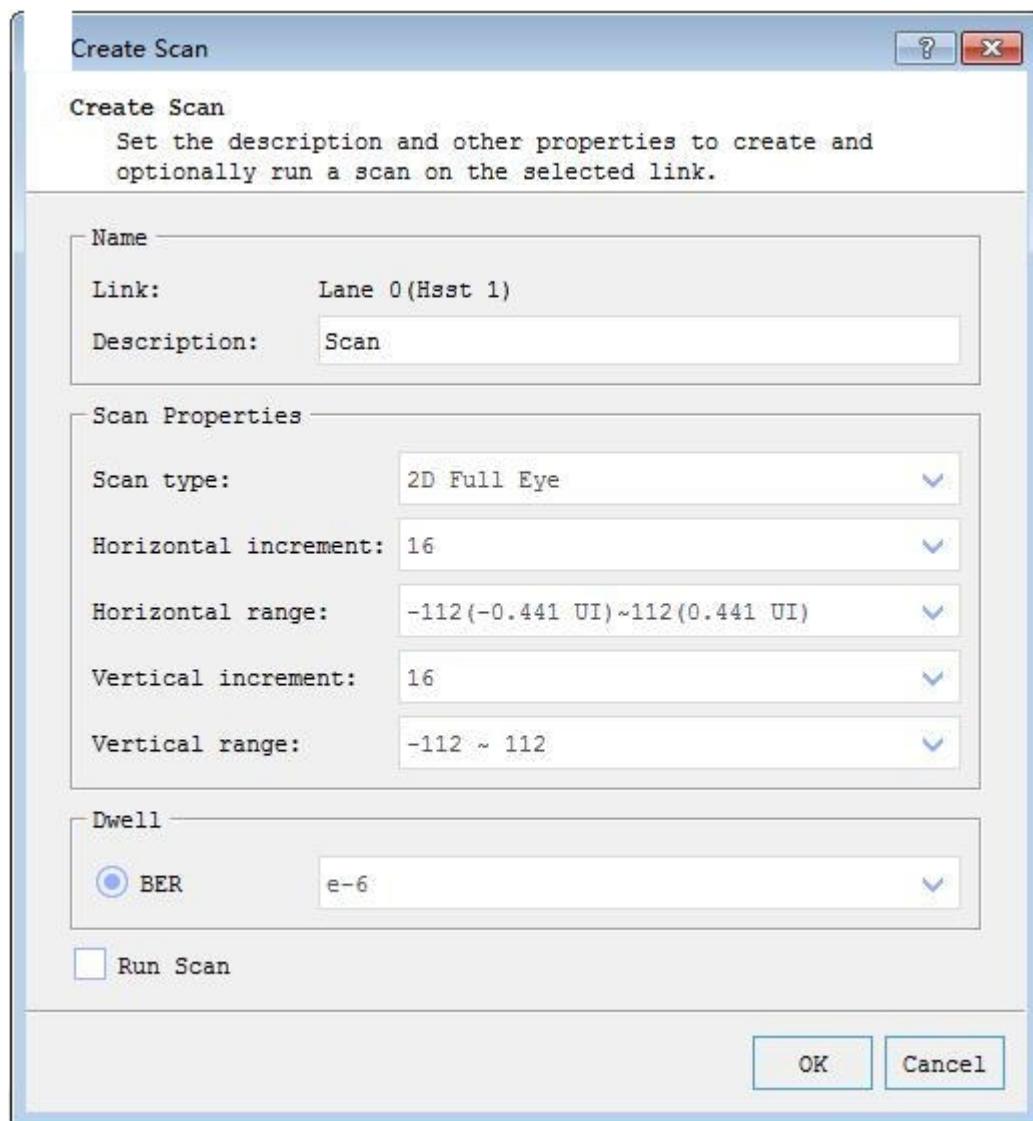


图 5-36 Create Scan 对话框

在该对话框中通过下拉框来配置扫描属性，点击“OK”即可创建扫描，若勾选“Run Scan”会在创建扫描后直接开始扫描眼图。扫描属性从上至下说明如下：

功能	说明
Link:	链路名称
Description	扫描名称
Scan type	生成眼图类型，目前仅支持 2D Full Eye

Horizontal increment	水平方向扫描步进值, 下拉列表选择, 包括 1~16。
Horizontal range	水平方向扫描范围, 下拉列表选择, 包括: -127~127, -126~126, ..., -2~2, -1~1
Vertical increment	竖直方向扫描步进值, 下拉列表选择, 包括 1~16。
Vertical range	竖直方向扫描范围, 下拉列表选择, 包括: -127~127, -126~126, ..., -2~2, -1~1
BER	切换扫描偏移时需达到的最低误码率: e-6 – e-11

在创建扫描后, 界面将自动跳转至“Hsst Scans”窗口, 如下图所示:

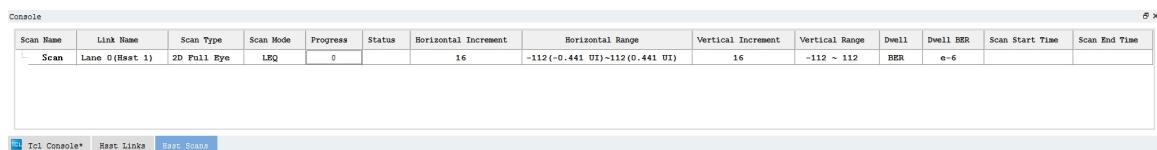


图 5-37 Hsst Scans 窗口

在该窗口下可再次配置上述扫描设置, 窗口各项属性说明如下:

功能	说明
Scan Name	扫描名称
Link Name	扫描所属通道名称
Scan type	生成眼图类型, 目前仅支持 2D Full Eye
Scan mode	扫描模式: LEQ/DFE, 该项参数只支持在 IP 中配置。
Progress	扫描进度条 (0-100%)
Status	扫描状态: Failed to Start: 开始扫描失

	败 Running: 正在扫描 Incomplete: 扫描未完成 Done: 扫描成功	
Horizontal increment	水平方向扫描步进值, 下拉列表选择, 包括 1~16。	
Horizontal range	水平方向扫描范围, 下拉列表选择, 包括: -127~127, -126~126, , -2~2, -1~1	
Vertical increment	竖直方向扫描步进值, 下拉列表选择, 包括 1~16。	
Vertical range	竖直方向扫描范围, 下拉列表选择, 包括: -127~127, -126~1265, , -2~2, -1~1	
Dwell	指定在特定的扫描偏移位置的测试时间	
BER	切换扫描偏移时需达到的最低误码率: e-6 – e-11	
Scan Start Time	开始扫描时间	
Scan End Time	结束扫描时间	

当再次配置完毕后, 可以通过右键选中该扫描, 此时弹出右键菜单, 选择“Start Scan”开始扫描眼图, 或使用快捷键“Ctrl + F5”启动扫描。右键菜单说明如下:

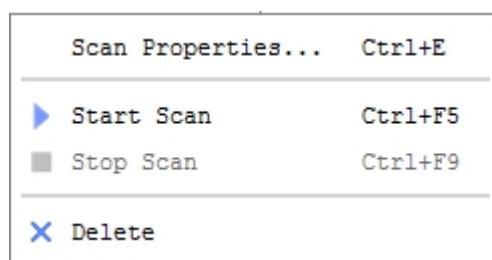


图 5- 38 Hsst Scan 右键菜单

功能	说明
Scan Properties	扫描属性
Start Scan	开始扫描眼图
Stop Scan	停止扫描眼图
Delete	删除选中的扫描，同时删除对应已生成的眼图

扫描过程中，可随时通过右键菜单中的“Stop Scan”或快捷键“Ctrl + F9”停止扫描。当扫描完成后，主界面右侧上方会显示生成的 Scan Plot 界面：

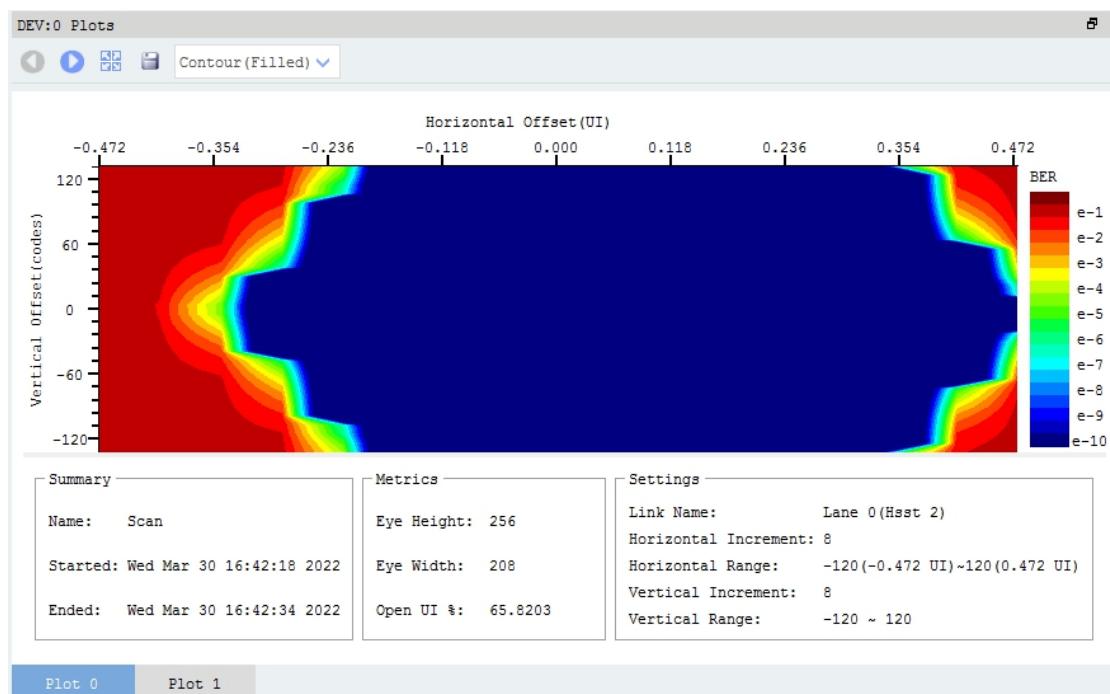


图 5-39 Scan Plot 界面

可以通过上方工具栏进行切换眼图、导出数据等操作：



工具栏从左到右说明如下：

【Go to previous plot】：切换到上一张眼图

【Go to next plot】：切换到下一张眼图

【Full Screen】：全屏显示眼图

【Export eye data】：输出眼图数据

【Contour】：切换画图方式：Coutour(Filled)/Coutour(Line)（填充型/线型）

- 眼图从左到右、从上到下属性说明：

功能	说明
Name	扫描名称
Started	开始扫描时间
Ended	结束扫描时间
Eye Height	眼图高度
Eye Width	眼图宽度
Open UI%	眼图打开百分比
Link Name	链路名称
Horizontal increment	水平方向扫描步进值
Horizontal range	水平方向扫描范围
Vertical increment	竖直方向扫描步进值
Vertical range	竖直方向扫描范围

5.16 DVIO 调试

5.16.1 总体介绍

DVIO(Debugger Virtual Input Output)功能是一种能够实时监控和驱动 FPGA 内部信号的调试手段，它代替了 LED 灯和操作按钮等硬件设备，帮助用户灵活得对 FPGA 内部运行情况进行了解与控制。

它的实现依赖于往用户设计中嵌入 DVIO 核，该核的 ProbeIn 端口与需要监视的用户设计信号相连，ProbeOut 端口与需要控制的用户设计信号相连。在调试过程中，软件可以固定的频率扫描 ProbeIn 所连信号，捕捉监控信号的实时值和跳变信息，并将用户对 ProbeOut 所连信号的操作及时反应到芯片内部，控制用户设计。

目前一个 Device 最多支持 15 个 DVIO 核，每个 DVIO 最多可开通 256 个 ProbeIn，256 个 ProbeOut，每个 ProbeOut 或 ProbeIn 最多可打开 256 位。

5.16.2 功能描述

下载带有 DVIO 功能的位流后，点击“扫链”将会出现如下图所示界面：

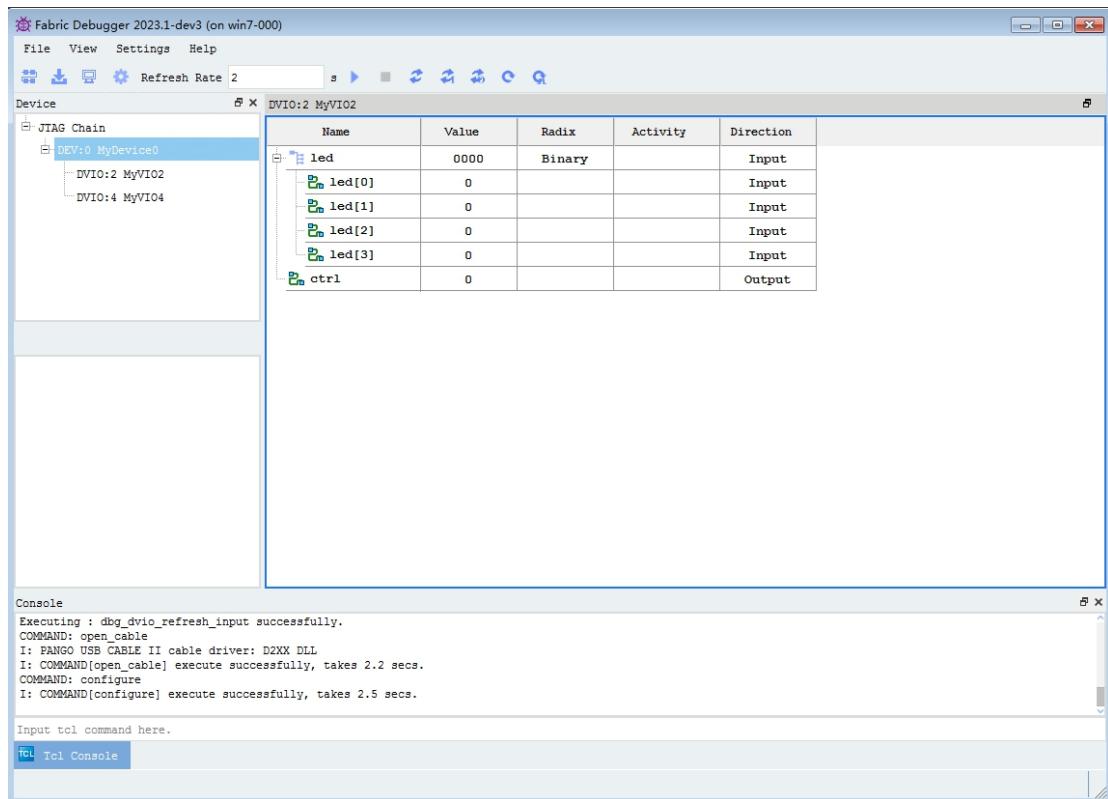


图 5- 40 DVIO 主界面

- 当位流中有 DebugCore，扫链后默认显示 DebugCore 的操作界面，当位流中仅有 DVIO，扫链后默认显示 DVIO 的操作界面。

器件栏显示当前位流包含的 DVIO，右键 DVIO 项，跳出操作菜单栏如下图所示：

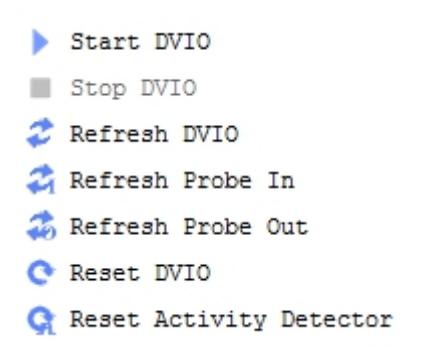


图 5- 41 器件栏 DVIO 右键菜单

各操作描述如下：

菜单项	功能
Start DVIO	启动 DVIO 自动刷新
Stop DVIO	停止 DVIO 自动刷新
Refresh DVIO	刷新 DVIO
Refresh Probe In	刷新 DVIO 的 ProbeIn
Refresh Probe Out	刷新 DVIO 的 ProbeOut
Reset DVIO	初始化 DVIO 的 ProbeOut
Reset Activity Detector	重置活动探测器

- 若当前 DVIO 核没有打开活动探测器功能，Reset Activity Detector 菜单项将为置灰不可操作状态。

DVIO 操作界面显示当前核的所有 ProbeIn 和 ProbeOut 端口，操作界面为一个信号表格，表格行分为 Bus 和 Net 两种级别，Bus 可折叠可展开。表格各列数据说明如下：

列名	说明
Name	Bus 或 Net 的名称
Value	Bus 或 Net 的值
Radix	Bus 值的显示进制
Activity	ProbeIn 的活动探测器状态
Direction	Bus 或 Net 的信号方向

- Radix 只对 Bus 行有意义，点击 Bus 的 Radix 列会跳出进制选择下拉框，包括二进制、八进制、十六进制、无符号整型和有符号整型。
- 只有使能了活动探测器的 DVIO 核才有 Activity 这一列，且只对 ProbeIn 有意义，活动探测器状态分为上升跳变、下降跳变、脉冲跳变。
- Direction 列用来区分该行是 ProbeIn 还是 ProbeOut。
- ProbOut 行可修改值，修改后会立刻写入到芯片。

行右键菜单，在信号表格的 ProbeIn 行按下鼠标右键，跳出右键菜单如图所示：

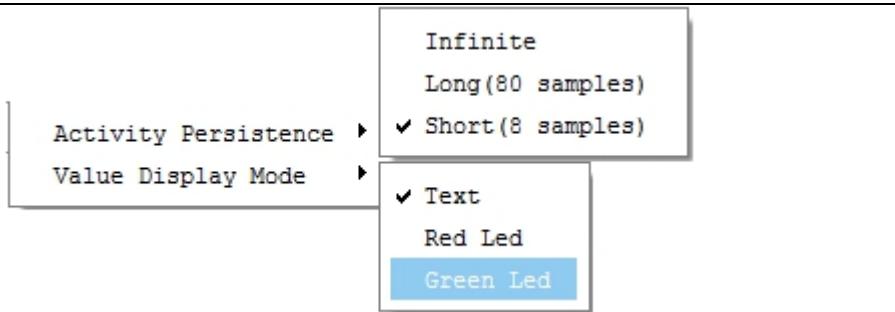


图 5- 42 ProbeIn 行右键菜单

各菜单项说明如下

菜单项	功能
Activity Persistence	活动探测器显示周期
Infinite	无限周期
Long (80 Samples)	显示 80 个扫描周期
Short (8 Samples)	显示 8 个扫描周期
Value Display Mode	ProbeIn Net 值显示模式
Text	ProbeIn Net 值文本显示
Red Led	ProbeIn Net 值以红色 LED 灯显示
Yellow Led	ProbeIn Net 值以绿色 LED 灯显示

- Activity Persistence 菜单项只有在该 DVIO 核的活动探测器功能打开了，才会显示。该菜单项用来设置自动刷新 DVIO 时，Activity Detector 的显示模式。设置为 Infinite 模式，自动刷新过程中，Activity Detector 一直积累且显示直至停止自动刷新才消失，即显示周期为开始到停止。设置为 Long 模式，Activity Detector 的显示周期为 80 个刷新周期，若在这 80 个刷新周期内值又发生了变化，重新进入新一轮显示周期。设置为 Short 模式，Activity Detector 的显示周期为 8 个刷新周期，若在这 8 个刷新周期内值又发生了变化，重新进入新一轮显示周期。

在信号表格的 ProbeOut 行按下鼠标右键，跳出右键菜单如图所示：

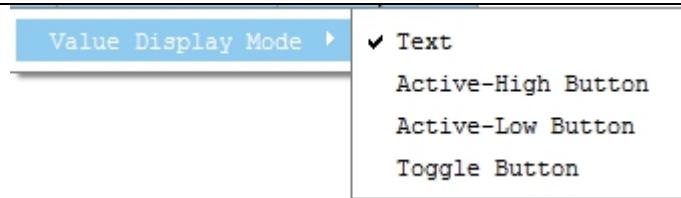


图 5- 43 ProbeOut 行右键菜单

各菜单项说明如下

菜单项	功能
Value Display Mode	ProbeOut Net 值操作模式
Text	ProbeOut Net 值文本操作
Active-High Button	ProbeOut Net 值常低按钮操作
Active-Low Button	ProbeOut Net 值常高按钮操作
Toggle Button	ProbeOut Net 值开关按钮操作

- 切换为常低按钮操作方式，net 值会首先切换为 0，在按下状态，net 切换为 1，松开，切换为 0；
- 切换为常高按钮操作方式，net 值会首先切换为 1，在按下状态，net 切换为 0，松开，切换为 1；

DVIO 操作界面支持批量式修改信号值显示方式，框选同方向型的信号行，鼠标右键跳出右键菜单，可批量式修改信号值显示方式。

6 TCL 命令

6.1 Debugger TCL Command 简介

现阶段 Debugger 支持的 tcl 命令如下：

```
source
dbg_help
dbg_close
dbg_connect
dbg_disconnect
dbg_delay
dbg_scan_chain
dbg_program
dbg_open_project
dbg_save_project
dbg_run
dbg_trig_immd
dbg_stop
dbg_read_user_code
dbg_read_device_id
dbg_read_status_register
dbg_read_instruction_register
dbg_power_init
dbg_get_cur_core
dbg_set_cur_core
dbg_get_buses
dbg_add_bus
dbg_add_to_bus
dbg_del_bus
dbg_reverse_bus
dbg_set_bus_radix
dbg_set_unit
dbg_set_condition
```

```
dbg_set_capture
dbg_set_storage
dbg_list
dbg_import_fic
dbg_export_fic
dbg_add_all_to_waveform
dbg_add_all_to_listing
dbg_clear_all_to_waveform
dbg_clear_all_to_listing
dbg_waveform_zoom
dbg_read_adc_register
dbg_write_adc_register
dbg_write_hsst_register
dbg_read_hsst_register
dbg_hsst_update
dbg_hsst_start
dbg_hsst_stop
```

可通过 `dbg_help` 获取系统可用命令, 具体命令使用方法: 命令名 `-help`, 如:
`dbg -help`, 所有输入到 TCL Console 的正确命令都会被记录下来, 保存在 log 目录下的 `TclCmd.tcl` 文件, 命令执行的调试信息输出到同目录下的 `TclCmd.log` 文件

6.2 Debugger TCL Command 详细介绍

6.2.1 通用命令

1) source

功能: 执行 TCL 脚本文件, 逐行解析并串行执行命令

命令名: `source`

参数说明: TCL 脚本文件路径(后缀名必须为.tcl)

用法示例: `source D:/TclCmd.tcl`

以 ‘#’ 字符开头的为注释行, 可参照以下格式:

#Fabric Debugger TCL cmd

dbg_connect -ip 192.29.103.191 -port 65420

dbg_scan_chain

如果脚本文件中某条指令执行失败，则中断执行，不再执行下一行指令

文件不能嵌套使用。

2) **dbg_help**

功 能： 列出当前所有支持的 TCL 命令

命令名： dbg_help

参数说明： 无

用法示例： dbg_help

3) **dbg_close**

功 能： 关闭 debugger 程序

命令名： dbg_close

参数说明： 无

用法示例： dbg_close

4) **dbg_connect**

功 能： 连接到指定服务器

命令名： dbg_connect

参数说明：

-ip： （必选）服务器 ip 地址

-port：（必选）服务器连接端口

如果两个参数任一为空， 默认以界面设置信息进行连接

用法示例： dbg_connect -ip 192.29.103.191 -port 65420

5) **dbg_disconnect**

功 能： 断开与服务器的连接

命令名: `dbg_disconnect`

参数说明: 无

用法示例: `dbg_disconnect`

6) `dbg_delay`

功 能: 开启一段时间的延时

命令名: `dbg_delay`

参数说明:

-time: (必选) 延迟时间, 单位为秒

用法示例: `dbg_delay -time 1`

7) `dbg_scan_chain`

功 能: 扫链, 获取挂在 Jtag 上的设备链

命令名: `dbg_scan_chain`

参数说明: 无

用法示例: `dbg_scan_chain`

8) `dbg_program`

功 能: 下载位流文件

命令名: `dbg_program`

参数说明:

-device: (可选) 选中设备索引, 默认选择当前设备

-file: (必选) 位流文件

用法示例: `dbg_program -device 0 -file C:/add.sbit`

9) `dbg_open_project`

功 能: 打开工程

命令名: `dbg_open_project`

参数说明：

-file: (必选) 工程名

用法示例：dbg_open_project -file ./dbg.dprj

10) **dbg_save_project**

功 能：保存工程

命令名：dbg_save_project

参数说明：

-file: (可选) 工程名

用法示例：dbg_save_project -file ./dbg.dprj

6.2.2 Debug core 相关命令

1) **dbg_run**

功能：触发，根据当前调试核设置条件抓取波形

命令名：dbg_run

参数说明：无

用法示例：dbg_run

2) **dbg_trig_immd**

功 能：立即触发，抓取波形

命令名：dbg_trig_immd

参数说明：无

用法示例：dbg_trig_immd

3) **dbg_stop**

功 能：停止正在触发的请求动作

命令名：dbg_stop

参数说明：无

用法示例：dbg_stop

4) **dbg_read_user_code**

功 能：从 jtagsever 回读 user code 信息

命令名：dbg_read_user_code

参数说明：无

用法示例：dbg_read_user_code

5) **dbg_read_device_id**

功 能：从 jtagsever 回读 device ID 信息

命令名：dbg_read_device_id

参数说明：无

用法示例：dbg_read_device_id

6) **dbg_read_status_register**

功 能：从 jtagsever 回读状态寄存器信息

命令名：dbg_read_status_register

参数说明：无

用法示例：dbg_read_status_register

7) **dbg_read_instruction_register**

功 能：从 jtagsever 回读指令寄存器信息

命令名：dbg_read_instruction_register

参数说明：无

用法示例：dbg_read_instruction_register

8) **dbg_power_init**

功能：从 jtagsever 回读上电初始化数据

命令名：dbg_power_init

参数说明：无

用法示例：dbg_power_init

9) **dbg_get_cur_core**

功 能：获取当前选中核

命令名：dbg_get_cur_core

参数说明：无

用法示例：dbg_get_cur_core

在 tcl console 中列出当前设备当前核

10) **dbg_set_cur_core**

功 能：设置当前选中核

命令名：dbg_set_cur_core

参数说明：

-device: (必选) 当前设备索引

-core: (必选) 当前核索引

用法示例：dbg_set_cur_core -device 0 -core 0

11) **dbg_get_buses**

功 能：获取当前设备当前核的总线 bus 信息

命令名：dbg_get_buses

参数说明：无

用法示例：dbg_get_buses

在 tcl console 中列出当前设备当前核的总线 bus 信息

12) **dbg_add_bus**

功能：在当前设备当前核中添加一条总线

命令名： `dbg_add_bus`

参数说明：

-name: (必选) 新增总线名称

-channel: (必选) 总线包含的数据通道索引，以 ‘/’ 为通道分割线，依次按顺序添加，当添加多个相连通道时，可使用’ - ‘连接，例如 1-16 或者 8-5

用法示例： `dbg_add_bus -name MyBus -channel 3/1-16/6/8-5/2/1`

13) **dbg_add_to_bus**

功能：添加数据通道到指定总线

命令名： `dbg_add_to_bus`

参数说明：

-name: (必选) 总线名称

-channel: (必选) 总线包含的数据通道索引，以 ‘/’ 为通道分割线，依次按顺序添加，当添加多个相连通道时，可使用’ - ‘连接，例如 1-16 或者 8-5

用法示例： `dbg_add_to_bus -name MyBus -channel 3/1-16/6/8-5/2/1`

14) **dbg_del_bus**

功 能：在当前设备当前核中删除指定总线

命令名： `dbg_del_bus`

参数说明：

-name: (必选) 总线名称

用法示例： `dbg_del_bus -name MyBus`

15) **dbg_reverse_bus**

功 能：反转指定总线

命令名： `dbg_reverse_bus`

参数说明：

-name: (必选) 总线名称

用法示例：dbg_reverse_bus -name MyBus

16) **dbg_set_bus_radix**

功 能：设置指定总线以指定进制表现

命令名：dbg_set_bus_radix

参数说明：

-name: (必选) 总线名称

-radix: (必选) 总线进制

0: 二进制

1: 八进制

2: 十六进制

3: 无符号十进制

4: 带符号十进制

用法示例：dbg_set_bus_radix -name MyBus -radix 0

17) **dbg_set_unit**

功能：设置 Trigger Unit 的触发格式

命令名：dbg_set_unit

参数说明：

-unit: (可选) Trigger Unit 索引号, 0~15 之间

-func: (可选) 匹配表达式, 可使用的表达式范围：“==”, “<>”, “>”, “>=”, “<”, “<=” “IN Range”, “Out of Range” , 当表达式中有空格时必须使用双引号将表达式引起来

-value: (可选) Trigger Unit 的值, 值的设定与进制有关不同进制下可接受的字符如下：

Hex: 0-9 以及 A-F

Octal: 0-7

Binary: X (不关心值为多少)、0、1、R (上升沿)、F (下降沿)、B (任意翻转)、N (不翻转), 当 Trigger Unit 可以检测沿 (Basic w.edges、Extended w.edges、Range w.edges) 时 R、F、B、N 可输入; (此进制下, 每一位的值都必须显示指定, 格式为: XX_XXXX_XXXX)

Unsigned: 0-9

-radix: (可选) 进制, 可接受的参数为对应进制的序号:

0: Hex

1: Octal

2: Binary

3: Unsigned

-cnt_type: (可选) 计数器类型, 可接受的参数为对应类型的序号:

0: Counter Disable

1: Occurring in exactly

2: Occurring in at least

3: Lasting for at least

-cnt_num: (可选) 计数次数, 当计数器类型不等于 disable 可用, 计数次数最少为 1, 最大不得超过 2 的 n 次幂, n 为计数器的位宽。

用法示例(三种用法):

显示所有 Trigger Unit 及其详细信息, 不带选项

dbg_set_unit

计数器不可用时, 设置 Trigger Unit(以 Trigger Port 位宽为 10 示例:)

dbg_set_unit -unit 1 -func "==" -value 11_XXXX_XXXX -radix 0

计数器可用时, 设置 Trigger Unit

dbg_set_unit -unit 1 -func "==" -value 68 -radix 2 -cnt_type 2 -cnt_num 1

18) **dbg_set_condition**

功 能: 对触发条件进行增加, 删除, 激活和组合操作

命令名: **dbg_set_condition**

参数说明:

-add: (可选) 增加一个 Trigger Condition, 默认参数 1

-delete: (可选) 删除一个 Trigger Condition, 默认参数 1

-active: (可选) 要激活的 Trigger Condition 索引号, 0~15 之间且小于总的 Trigger Condition 数

-select: (可选) 该选项不能单独使用, 需要与-set 选项同时使用。
要设定的 Trigger Condition 索引号, 0~15 之间且小于总的 Trigger Condition 数

-set: (可选) 该选项不能单独使用, 需要与-select 选项同时使用。

设定-select 指定的 Trigger Condition 的 TU 组合, 如 TU0&&TU1

用法示例(五种用法):

a) 显示所有 Trigger Condition 及其详细信息, 不带选项

 dbg_set_condition

b) 增加一个 Trigger Condition

 dbg_set_condition -add 1

c) 删 除一个 Trigger Condition

 dbg_set_condition -delete 1

d) 激活指定的 Trigger Condition

 dbg_set_condition -active 1

e) 设定指定的 Trigger Condition

 dbg_set_condition -select 2 -set "TU0&& TU1&& TU2"

19) dbg_set_capture

功 能: 设置捕获条件和存储条件

命 令 名: dbg_set_capture

参 数 说 明:

-type: (可选) 捕获风格, 可设置为 Windows 或 Nsamples, “W”, “w” 或“0” 均表示 Windows 风格, “N”, “n” 或 “1” 表示 Nsamples

-windows: (可选) 窗口数目, 范围为 1 到最大数据深度; 不可与选项-depth 同时使用。仅在 Windows 风格下可用

-depth: (可选) 每个窗口的数据深度, 范围为 1 到最大数据深度除以窗口数, 且必须为 2 的 n 次幂; 不可与选项-window 同时使用。仅在 Windows 风格下可用

-position: (可选) 触发位置, 范围为 1 到最大数据深度除以窗口数。仅在 Windows 风格下可用

-samples: (可选) 采样个数, 范围为 1 到最大数据深度。仅在 Nsamples 风格下可用

用法示例(三种用法):

a) 显示当前的捕获设置和存储条件, 不带选项

 dbg_set_capture

b) 设置 windows 风格的捕获条件

 dbg_set_capture -type 0 -windows 1 -position 10

c) 设置 Nsamples 风格的捕获条件

 dbg_set_capture -type 1 -samples 512

20) **dbg_set_storage**

功 能: 设置存储条件

命令名: **dbg_set_storage**

参数说明:

-storage: (必选) 存储设定, 设定当前核存储数据的条件, 当该功能禁用时, 默认值为 “All Data”; 当该功能可用时, 可设置

为 TU 组合, 如 TU0&&TU1, 或设置为 “All Data”

用法示例: `dbg_set_storage -storage "TU0 && TU1"`

21) **dbg_list**

功 能: 显示所有核详细信息或指定选项的信息显示

命令名: **dbg_list**

参数说明:

-device (可选) 显示所有设备名称, 默认参数 1

-core (可选) 显示所有核名称， 默认参数 1
-data_port (可选) 显示所有 data port , 默认参数 1
-trig_port (可选) 显示所有 trigger port , 默认参数 1
-trig_unit (可选) 显示所有 trigger unit , 默认参数 1

用法示例 (两种用法):

1) 显示所有核的详细信息, 不带选项

dbg_list

2) 显示指定选项的信息, 带一个或多个选项

dbg_list -data_port 1 -trig_port 1

22) **dbg_import_fic**

功 能: 导入 fic 文件

命令名: `dbg_import_fic`

参数说明:

-file: (必选) fic 文件

用法示例: `dbg_import_fic -file ./dbg.fic`

23) **dbg_export_fic**

功 能: 导出 fic 文件

命令名: `dbg_export_fic`

参数说明:

-file: (必选) fic 文件

用法示例: `dbg_export_fic -file ./dbg.fic`

24) **dbg_add_all_to_waveform**

功 能: 添加所有信号到 waveform

命令名: `dbg_add_all_to_waveform`

参数说明: 无

用法示例: `dbg_add_all_to_waveform`

25) `dbg_add_all_to_listing`

功 能: 添加所有信号到 listing

命令名: `dbg_add_all_to_listing`

参数说明: 无

用法示例: `dbg_add_all_to_listing`

26) `dbg_clear_all_to_waveform`

功 能: 清除 waveform 所有信号

命令名: `dbg_clear_all_to_waveform`

参数说明: 无

用法示例: `dbg_clear_all_to_waveform`

27) `dbg_clear_all_to_listing`

功 能: 清除 listing 所有信号

命令名: `dbg_clear_all_to_listing`

参数说明: 无

用法示例: `dbg_clear_all_to_listing`

28) `dbg_waveform_zoom`

功 能: waveform 的放大缩小

命令名: `dbg_waveform_zoom`

参数说明:

-func 缩放功能

in: zoom in

out: zoom out

fit: zoom fit

full: zoom full

xo: zoom XO

previous: zoom previous

forward: zoom forward

用法示例: `dbg_waveform_zoom -func in`

29) `dbg_read_adc_register`

功 能: 读 ADC 寄存器

命令名: `dbg_read_adc_register`

参数说明:

不加参数

打印所有 ADC 寄存器的值

-address

带一个 ADC 地址参数, 只打印指定地址 ADC 的值

用法示例: `dbg_read_adc_register -address 0x18`

30) `dbg_write_adc_register`

功 能: 写 ADC 寄存器

命令名: `dbg_write_adc_register`

参数说明:

-address

指定 ADC 寄存器地址

-value

指定写入 ADC 寄存器的值

用法示例: `dbg_write_adc_register -address 0 -value 0x55`

6.2.3 iScan 相关命令

此类 tcl 命令只针对 iScan 功能的器件, 并且显示 iScan 功能相关界面后可执
(下载的位流中需包含 iScan 功能)。,

1) **dbg_write_hsst_register**

功 能：写 HSST 寄存器

命令名：dbg_write_hsst_register

参数说明：

-address

指定 HSST 寄存器地址

-value

指定写入 HSST 寄存器的值

用法示例：dbg_write_hsst_register -address 0x8000003 -value 0x22

2) **dbg_read_hsst_register**

功 能：读 HSST 寄存器

命令名：dbg_read_hsst_register

参数说明：

-address

指定 HSST 寄存器地址

用法示例：dbg_read_hsst_register -address 0x8000004

3) **dbg_hsst_update**

功 能：读取所有 HSST 寄存器的值并更新到 APB 界面

命令名：dbg_hsst_update

参数说明：无。

用法示例：dbg_hsst_update

4) **dbg_hsst_start**

功 能：开始当前 HSST 眼图扫描测试。

命令名：dbg_hsst_start

参数说明：

-scan

指定扫描序号

用法示例： `dbg_hsst_start -scan 0`

5) **`dbg_hsst_stop`**

功 能：停止当前 HSST 眼图扫描测试。

命令名： `dbg_hsst_stop`

参数说明：无。

用法示例： `dbg_hsst_stop`

6.2.4 DVIO 相关命令

1) **`dbg_select_current_dvio`**

功 能：选择当前操作的 DVIO 核

参数说明： `-device` 操作的设备 `-dvio` 操作的 DVIO 核

用法示例： `dbg_select_current_dvio -device 0 -dvio 2`

2) **`dbg_show_current_dvio`**

功 能：显示当前操作的 DVIO 核

用法示例： `dbg_show_current_dvio`

3) **`dbg_dvio_refresh_input`**

功 能：获取当前操作 DVIO 核的值

用法示例： `dbg_dvio_refresh_input`

4) **`dbg_dvio_refresh_output`**

功 能：获取当前操作 DVIO 核的 ProbeOut 值

用法示例： `dbg_dvio_refresh_output`

5) `dbg_dvio_reset_output`

功 能：初始化当前操作 DVIO 核的 ProbeOut 值

用法示例：`dbg_dvio_reset_output`

6) `dbg_dvio_set_output`

功 能：设置当前操作 DVIO 核的某个 ProbeOut 值

参数说明：`-probe_out` 设置的 ProbeOut `-value` 设置的 ProbeOut 值，只支持二进制数值，且位数必须与 ProbeOut 的位宽一致

用法示例：`dbg_dvio_set_output -probe_out 2 -value 11110000`

7) `dbg_dvio_commit_output`

功 能：将当前操作 DVIO 核的某个 ProbeOut 值写入芯片

参数说明：`-probe_out` 写入芯片的 ProbeOut

用法示例：`dbg_dvio_commit_output -probe_out 1`

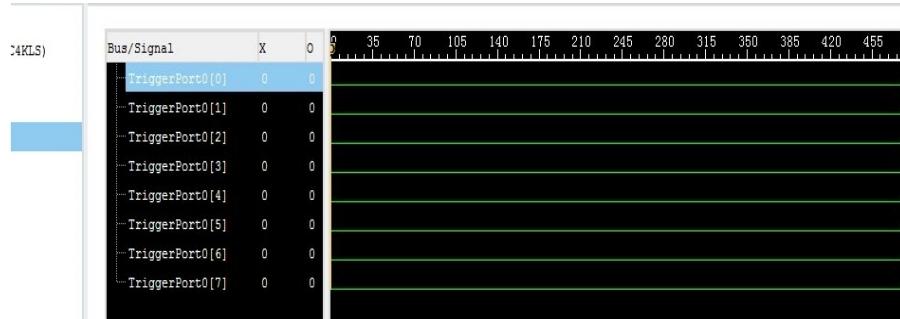
8) `dbg_dvio_print`

功 能：打印当前操作 DVIO 核的信息

用法示例：`dbg_dvio_print`

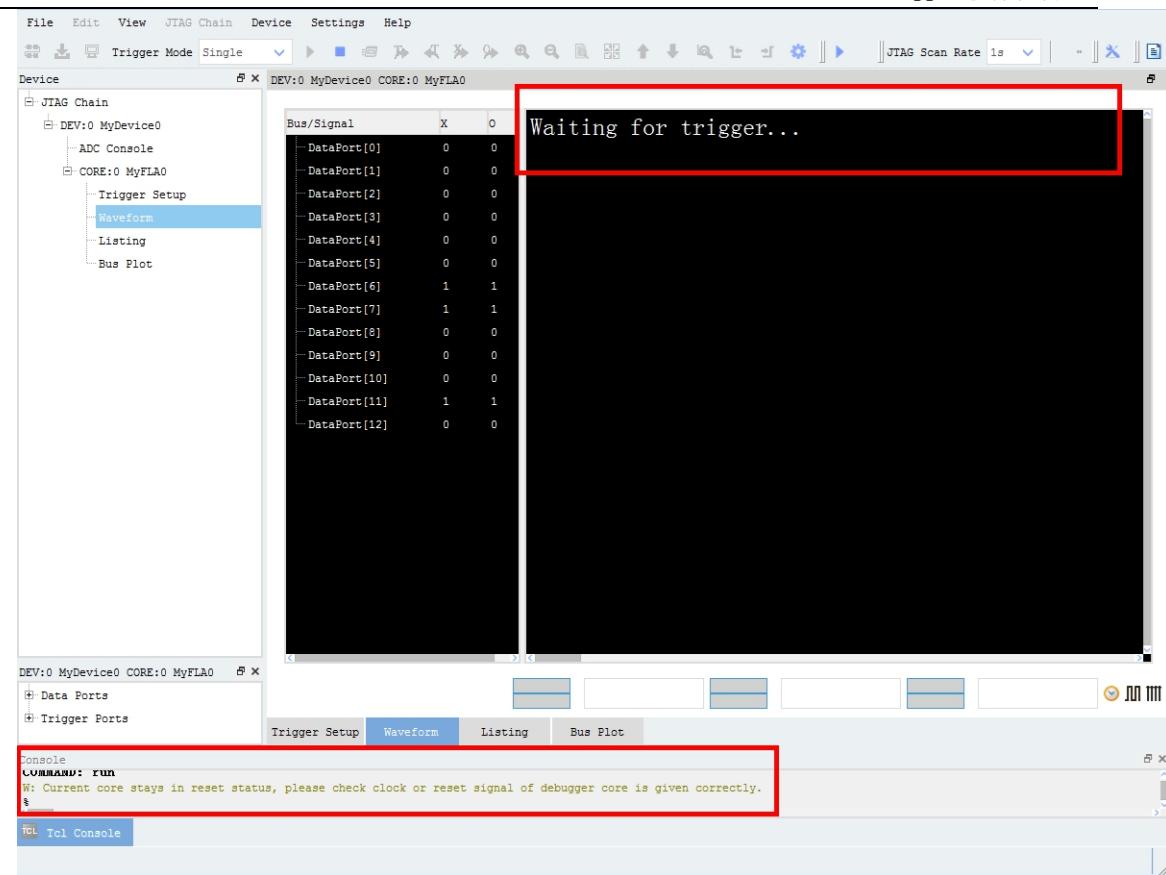
7 常见问题

1. 点击“Run”抓取数据时，抓取到的波形数据异常，波形图显示所有 channel 的数据均为 0 如下图所示：



参考办法：

- A) 查看约束文件，复位管脚约束是否正确。
 - B) 查看是否存在时序违例。
2. 点击“Run”抓取数据时，波形界面显示“Waiting for trigger...”，同时“Console”中提示警告信息“W: Current core stays in reset status, please check clock or reset signal of debugger core is given correctly.”，如下图所示：



参考办法：

- A) 信号源（晶振）是否存在。
- B) 查看约束文件，时钟管脚约束是否正确。

3. DPIO 界面设置好 ProbeOut 值后写入芯片不成功，刷新 ProbeOut 值，从芯片中读取到的还是原来的值。

参考办法：

- A) 检查对 DPIO Core 的时钟 clk 信号时钟引脚约束是否正确。

8 免责声明

版权声明

本文档版权归深圳市紫光同创电子有限公司所有，并保留一切权利。未经书面许可，任何公司和个人不得将此文档中的任何部分公开、转载或以其他方式披露、散发给第三方。否则，公司必将追究其法律责任。

免责声明

- 1、本文档仅提供阶段性信息，所含内容可根据产品的实际情况随时更新，恕不另行通知。如因本文档使用不当造成的直接或间接损失，本公司不承担任何法律责任。
- 2、本文档按现状提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。
- 3、公司保留任何时候在不事先声明的情况下对公司系列产品相关文档的修改权利。