



MIPI Demo 说明

Richard Zhu

richardz@efinixinc.com

MIPI Demo 功能简介

- **MIPI发送**

- 通过 VIO 配置 MIPI-Rx 的工作模式
- 通过 VIO 修改图像参数；
- 通过 VIO 显示自动计算的控制参数，并产生控制信号；
- 产生伪随机数据用于和 MIPI-Rx 对接测试硬件误码；

- **MIPI接收**

- 通过 VIO 配置 MIPI-Rx 的工作模式；
- 通过 VIO 检视 MIPI-Rx 的工作状态；
- 内部自动测量图像参数并通过 VIO 显示；
- 和 MIPI-Rx 配合测试硬件误码；

- **MIPI-RX 波形显示**

- 通过 ILA 对 MIPI-RX 的波形进行显示

- **硬件测试接口**

- 对 MIPI-TX 和 MIPI-RX 的控制信号输出到 GPIO，可以通过示波器进行检测；

用途

- 演示 MIPI-TX 和 MIPI-RX 的功能
- 环回验证 MIPI 的收发功能；
- 调试 MIPI-TX 参数和功能；
- 通过 MIPI-RX 检测外部输入信号特征；

文件组成

与项目（Project）有关的文件：

- TimeConstraint.SDC —— 定时约束文件；
- MIPIDebug.xml —— 项目文件；
- MIPIDebug.peri.xml —— 管脚和外设文件

与调试（Debug）有关的文件：

- Efinity_Dbg.Bat —— 调试批处理；（需要把文件中的路径修改到安装目录）
- Debug_top.v —— Debugger 的顶层代码；
- Debug_profile.json —— Debugger 的信号定义文件；

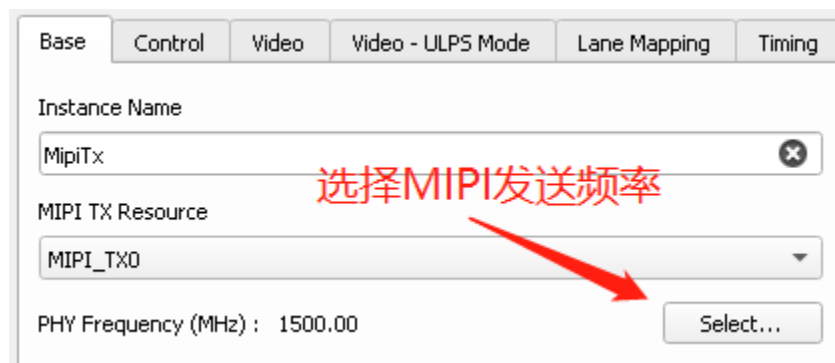
与设计有关的文件：

- MipiPixelData.v —— 像素数据发送和检测；
- MipiRxprocess.v —— MIPI 接收测量和状态检测；
- MipiTest.v —— MIPI 演示顶层文件；
- MipiTxProcess.V —— MIPI 发送参数计算和控制；
- PRBS9.v —— 伪随机码产生和检测；

```
87 localparam [27:0] TxPixelClkFreq_C = 28'd100_000_000;  
88 localparam [27:0] RxPixelClkFreq_C = 28'd100_000_000;  
89 parameter RightCntWidth_C = 20;
```

顶层参数（MIPITest）：

- TxPixelClkFreq_C：发送时钟频率；
- RxPixelClkFreq_C：接收时钟频率；
- RightCntWidth_C：判断数据正确的计数器宽度；



文件组成



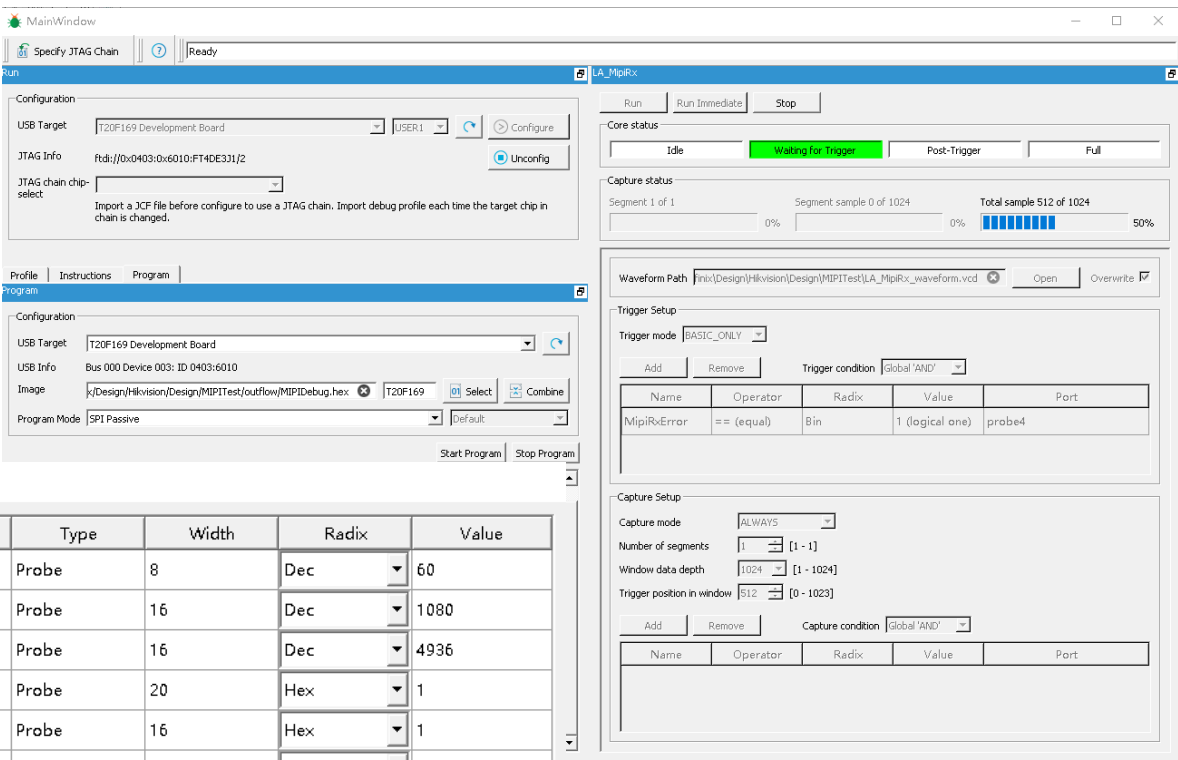
操作界面

MipiTx

	Name	Type	Width	Radix	Value
↑	ConfigTxFrameRate	Source	8	Dec	60
↑	ConfigTxVertPixelNumber	Source	16	Dec	1080
↑	ConfigTxHoriPixelNumber	Source	16	Dec	6000
↑	ConfigTxVBlankTime_us	Source	12	Hex	1
↑	ConfigTxHBlankTime_us	Source	8	Hex	1
↑	configTxLanesNumber	Source	3	Hex	2
↑	ConfigPixelDataType	Source	6	Hex	2a
↑	ConfigTxVCEnable	Source	4	Hex	0
↑	ControlTxDPHYReset	Source	1	Hex	1
↑	ControlTxCSIReset	Source	1	Hex	1
↑	ControlTxDisable	Source	1	Hex	0
↑	ConfigTxFramerMode	Source	1	Hex	1
↑	ParamTxVertSyncLength_Cycle	Probe	24	Dec	1667184
↑	ParamTxHoriSyncLength_Cycle	Probe	16	Dec	1444
↑	paramTxDataValidLength_Cycle	Probe	16	Dec	617
↑	paramTxVertBlankLength_Cycle	Probe	20	Dec	100
↑	paramTxHoriBlankLength_Cycle	Probe	16	Dec	100
↑	RealTxHoriPixelNumber	Probe	16	Dec	4936

MipiRx

	Name	Type	Width	Radix	Value
↑	VidioRxFrameRate	Probe	8	Dec	60
↑	VidioRxVertPixeNumber	Probe	16	Dec	1080
↑	VidioRxHoriPixeNumber	Probe	16	Dec	4936
↑	VidioRxVBlankTime_us	Probe	20	Hex	1
↑	VidioRxHBlankTime_us	Probe	16	Hex	1
↑	ConfigRxLanesNumber	Source	3	Hex	2
↑	StaRxDataType	Probe	6	Hex	2a
↑	StateRxVCActive	Probe	4	Hex	1
↑	ControlRxDPHYReset	Source	1	Hex	1
↑	ControlRxCSIReset	Source	1	Hex	1
↑	StateRxChannelActive	Probe	1	Hex	1
↑	ControlRxErrorClear	Source	1	Hex	0
↑	StateRxError	Probe	18	Hex	200
↑	ConfigRxVCEnable	Source	4	Hex	f
↑	StateRxReslutionChange	Probe	1	Hex	0
↑	StateRxStateChange	Probe	1	Hex	0
↑	StateRxDataRight	Probe	8	Bin	11111111



TX 图像参数设定

Vidio TX 图像参数

ConfigTxFrameRate	发送帧率
ConfigTxVertPixelNumber	列像素个数
ConfigTxHoriPixelNumber	行像素个数
ConfigTxVBlankTime_us	消息信号单位 us

MIPI-TX 控制信号参数

TxVertSyncLength_Cycle	MIPI-Tx 中的 VSYNC 信号的长度；用时钟周期标识
TxHoriSyncLength_Cycle	MIPI-Tx 中的 HSYNC 信号的长度；用时钟周期标识
TxDataValidLength_Cycle	MIPI-Tx 中的 DVLID 信号的长度；用时钟周期标识
TxVertBlankLength_Cycle	两个 VSYNC 中的间隔长度；用时钟周期标识

说明：

1. TxVertBlankLength_Cycle 参数是根据图像参数自动计算的；计算是基于发送时钟 (MipiTxPixelCik) 频率，所以必须确保参数 TxPixelCikFreq_C 和 RealTxHoriPixelNumber 2. TxPixelCik 的频率与 MIPI-Tx 的真实使用的像素个数；MIPI 工作频率设定在 Interface Designer 的 MIPI TX -> Base

3. 可以通过提高 MipiTxPixelCik 的频率和 MIPI-Tx 的工作频率提高图像带宽；如果 1080P 60Fps 的 RGB24 的图像；必须满足：

- MipiTxPixelCik > 65M
- Mipi PHY 工作频率 1.5G，不小于 2Lanes

4. 当设置的参数 MIPI-TX 无法输出时，电路会计算一个行参数进行输出；计算出的行像素个数会在 RealTxHoriNumber 的参数中显示

MipiTx

Name	Type	Width	Radix	Value
ConfigTxFrameRate	Source	8	Dec	60
ConfigTxVertPixelNumber	Source	16	Dec	1080
ConfigTxHoriPixelNumber	Source	16	Dec	6000
ConfigTxVBlankTime_us	Source	12	Hex	1
ConfigTxHBlankTime_us	Source	8	Hex	1
configTxLanesNumber	Source	3	Hex	2
ConfigPixelFormatType	Source	6	Hex	2a
ConfigTxVCEnable	Source	4	Hex	0
ControlTxDPHYReset	Source	1	Hex	1
ControlTxCSIReset	Source	1	Hex	1
ControlTxDisable	Source	1	Hex	0
ConfigTxFramerMode	Source	1	Hex	1
ParamTxVertSyncLength_Cycle	Probe	24	Dec	1667184
ParamTxHoriSyncLength_Cycle	Probe	16	Dec	1444
paramTxDataValidLength_Cycle	Probe	16	Dec	617
paramTxVertBlankLength_Cycle	Probe	20	Dec	100
paramTxHoriBlankLength_Cycle	Probe	16	Dec	100
RealTxHoriPixelNumber	Probe	16	Dec	4936

MIPI-TX 参数和控制信号

MIPI-TX 配置参数

ConfigTxLanesNumber	MIPI-TX 发送的 LANES 的个数（取值 1、2、4）
ConfigPixelDataType	MIPI-TX 发送的数据类型
ConfigTxFramerMode	MIPI-TX 发送工作模式
ConfigTxVCEnable	MIPI-TX 虚拟通道允许
ControlTxDPHYReset	MIPI-TxDPHY 复位信号，低有效
ControlTxCSIRreset	MIPI-TxCSI 复位信号，低有效
ControlTxDisable	MIPI-TX 发送失效，高停止发送

说明：

1. 修改 LANES 时，需要将 **ControlTxDPHYReset**、**ControlTxCSIRreset** 置为有效；
2. **ConfigPixelDataType** 的设置与有效数据宽度有关，详细情况见附表；
3. 如果需要输出严格按我们提供的控制信号输出，需要将 **ConfigTxFramerMode** 配置为 1；
4. **ConfigTxVCEnable** 仅用于 VC 发送测试，每 Bite 对应一个 VC；设为 1，会插入对应 VC 的数据；

	Name	Type	Width	Radix	Value
↑	ConfigTxFrameRate	Source	8	Dec	60
↑	ConfigTxVertPixelNumber	Source	16	Dec	1080
↑	ConfigTxHoriPixelNumber	Source	16	Dec	6000
↑	ConfigTxVBlankTime_us	Source	12	Hex	1
↑	ConfigTxHBlankTime_us	Source	8	Hex	1
↑	configTxLanesNumber	Source	3	Hex	2
↑	ConfigPixelDataType	Source	6	Hex	2a
↑	ConfigTxVCEnable	Source	4	Hex	0
↑	ControlTxDPHYReset	Source	1	Hex	1
↑	ControlTxCSIRreset	Source	1	Hex	1
↑	ControlTxDisable	Source	1	Hex	0
↑	ConfigTxFramerMode	Source	1	Hex	1
↑	ParamTxVertSyncLength_Cycle	Probe	24	Dec	1667184
↑	ParamTxHoriSyncLength_Cycle	Probe	16	Dec	1444
↑	paramTxDataValidLength_Cycle	Probe	16	Dec	617
↑	paramTxVertBlankLength_Cycle	Probe	20	Dec	100
↑	paramTxHoriBlankLength_Cycle	Probe	16	Dec	100
↑	RealTxHoriPixelNumber	Probe	16	Dec	4936

MIPI-RX 参数和控制信号

MIPI-RX 配置参数

ConfigRxLanesNumber	MIPI-RX 通道 (LANES) 数量
ConfigRxVCEnable	MIPI-RX 接收 VC 允许
ControlRxDPHYReset	MIPI-RX DPHY 复位, 低有效
ControlRxCSIRreset	MIPI-RX CSI 复位信号, 低有效
ControlRxErrorClear	MIPI-RX 接收错误清除, 高有效

说明:

1. 修改 LANES 数量 (ConfigRxLanesNumber) 时, 需要将 ControlRxDPHYReset、ControlRxCSIRreset 置为有效;
2. ConfigRxVCEnable 每个 bit 对应一个 VC; 置为 0, MIPI 将屏蔽对应 VC 的数据;
3. ConfigRxLanesNumber 和实际使用通道数对应;

	Name	Type	Width	Radix	Value
↑	ConfigRxLanesNumber	Source	2	Hex	0
↑	ConfigRxVCEnable	Source	4	Hex	0
↑	ControlRxDPHYReset	Source	1	Hex	0
↑	ControlRxCSIRreset	Source	1	Hex	0
↑	ControlRxErrorClear	Source	1	Hex	0
↑	VidioRxFrameRate	Probe	8	Hex	0
↑	VidioRxVertPixeNumber	Probe	16	Hex	0
↑	VidioRxHoriPixeNumber	Probe	16	Hex	0
↑	VidioRxVBlankTime_us	Probe	20	Hex	0
↑	VidioRxHBlankTime_us	Probe	16	Hex	0
↑	StaRxDataType	Probe	6	Hex	0
↑	StateRxVCActive	Probe	4	Hex	0
↑	StateRxError	Probe	18	Hex	0
↑	StateRxReslutionChange	Probe	1	Hex	0
↑	StateRxStateChange	Probe	1	Hex	0
↑	StateRxChannelActive	Probe	1	Hex	0
↑	StateRxDataRight	Probe	8	Hex	0

RX 图像概况和 MIPI-RX 状态

RX 图像概况

VidioRxFrameRate	接收图像的帧率
VidioRxVertPxeNumber	接收图像的列像素个数
VidioRxHoriPxeNumber	接收图像的行像素个数
VidioRxVBlankTime_us	接收图像的场消隐时间，单位 us
VidioRxHBlankTime_us	接收图像的行消隐时间，单位 us
StaRxDataType	MIPI-Rx 的数据类型
StateRxVCActive	MIPI-RX 的活动的虚拟通道（VC）
StateRxError	MIPI-RX 错误指示
StateRxReslutionChange	接收图像分辨率改变指示
StateRxStateChange	MIPI-RX 状态改变指示
StateRxChannelActive	MIPI-RX 通道有效
StateRxDataRight	接收数据正确（与 MIPI-TX 配合）

1. 图像分辨率改变指示（**StateRxReslutionChange**）和 **MIPI-RX** 状态改变指示（**StateRxStateChange**）有效时间会持续最少一秒；
2. 帧率和消隐时间的计算是基于发送时钟（**MipiTxPixelClk**）频率，所以必须确保参数 **TxPixelClkFreq_C** 和实际一致；
3. **StateRxError** 的具体含义见附表；错误指示只能在 **ControlRxErrorClear** 为高时才会被清除；如果指示为 200 表示收到帧正常；
4. **StateRxDataRight** 配合 **MIPI-TX** 的伪随机码发送进行检测；当前支持 **RX** 和 **TX** 数据宽度一样的 **DataType**

	Name	Type	Width	Radix	Value
↑	ConfigRxLanesNumber	Source	2	Hex	0
↑	ConfigRxVCEnable	Source	4	Hex	0
↑	ControlRxDPHYReset	Source	1	Hex	0
↑	ControlRxCSIReset	Source	1	Hex	0
↑	ControlRxErrorClear	Source	1	Hex	0
↑	VidioRxFrameRate	Probe	8	Hex	0
↑	VidioRxVertPxeNumber	Probe	16	Hex	0
↑	VidioRxHoriPxeNumber	Probe	16	Hex	0
↑	VidioRxVBlankTime_us	Probe	20	Hex	0
↑	VidioRxHBlankTime_us	Probe	16	Hex	0
↑	StaRxDataType	Probe	6	Hex	0
↑	StateRxVCActive	Probe	4	Hex	0
↑	StateRxError	Probe	18	Hex	0
↑	StateRxReslutionChange	Probe	1	Hex	0
↑	StateRxStateChange	Probe	1	Hex	0
↑	StateRxChannelActive	Probe	1	Hex	0
↑	StateRxDataRight	Probe	8	Hex	0

参数表

PHY Frequency (MHz) : 1500.00

Clock Timer

T _{CLK-POST} (ns)	113
T _{CLK-TRAIL} (ns)	96
T _{CLK-PREPARE} (ns)	48
T _{CLK-ZERO} (ns)	304
Escape Clock Freq (MHz)	20.00
T _{CLK-PRE} (ns)	250

Data Timer

T _{HS-PREPARE} (ns)	66
T _{HS-ZERO} (ns)	142
T _{HS-TRAIL} (ns)	90

```
//Base Parameter
#define MipiTxPhyClkFrq      32'd1_500_000_000
#define MipiTxPixelClkFreq  27'd 100_000_000
#define MipiTxEscClkFreq    27'd 20_000_000

////////////////////////////////////
//Mipi Tx Timing Parameter
#define MipiTx_TCLK_Post    16'd113
#define MipiTx_TCLK_Trail   16'd96
#define MipiTx_TCLK_Prepare 16'd48
#define MipiTx_TCLK_Zero    16'd304
#define MipiTx_TCLK_Pre     16'd250

#define MipiTx_THS_Prepare   16'd66
#define MipiTx_THS_Zero     16'd142
#define MipiTx_THS_Trail    16'd90
#define MipiTx_THS_Exit     16'd250
```

说明:

1. 内部计算依赖于我们提供的参数;
2. 时钟频率参数的单位为 Hz , 定时参数的单位为 ns
3. MipiTxPhyClk、MipiPixelClk 和 TxEscClk 时钟频率必须和实际使用的一样;
4. 修改配置参数, 需要即使修改这个参数表, 以保证内部计算的数据统一, 否则可能造成 MIPI Tx 工作不正常;