

MIPI介绍及Demo演示 v1.0

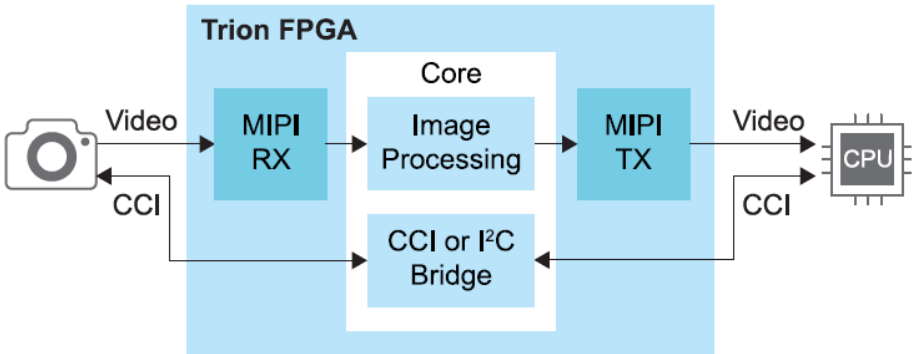
主要内容

- MIPI 主要功能介绍
- MIPI 控制器参数介绍
- MIPI Clk-Lane和Data-Lane工作状态切换介绍
- MIPI demo功能简介

参考文档

- MIPI Timing计算说明v1.1.ppt
- an015-design-for-mipi-v2.1.pdf
- mipi-utility-v1.1.xlsm
- MIPI-CSI-2-协议介绍.ppt
- MIPI演示说明_1V5.ppt
- MIPI参数和波形.pptx

MIPI 主要功能介绍



- MIPI TX/RX接口支持MIPI CSI-2规范v1.3和MIPI D-PHY规范v1.1
- MIPI D-PHY和CSI-2控制器均为硬核，用户不能绕过CSI-2控制器只使用D-PHY
- 支持配置为1、2、4Lanes模式
- 高速模式每条Lane支持高达1.5 Gbps
- 可配置为连续和非连续时钟模式
- 支持超低功耗状态(ULPS)
- 64位像素接口

Supported Data Type	Format
RAW	RAW6, RAW7, RAW8, RAW10, RAW12, RAW14
YUV	YUV420 8-bit (legacy), YUV420 8-bit, YUV420 10-bit, YUV420 8-bit (CSPS), YUV420 10-bit (CSPS), YUV422 8-bit, YUV422 10-bit
RGB	RGB444, RGB555, RGB565, RGB666, RGB888
User Defined	8 bit format

MIPI 控制器主要参数

- MIPI发送10个参数

- DHY时钟频率
- Clk-Lane参数
- Data-Lane参数

Block Editor

LPS Mode Lane Mapping Timing

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)	90
T _{HS-ZERO} (ns)	142
T _{HS-TRAIL} (ns)	106

Red arrows from the text on the left point to the following fields: PHY Frequency (MHz), T_{CLK-POST}, T_{CLK-TRAIL}, T_{CLK-PREPARE}, T_{CLK-ZERO}, T_{CLK-PRE}, T_{HS-PREPARE}, T_{HS-ZERO}, and T_{HS-TRAIL}.

- MIPI接收3个参数

- 校准时钟
- Clk-Lane参数
- Data-Lane参数

Block Editor

Status Lane Mapping Timing

Calibration Clock Freq (MHz) 100.00

Clock Timer

T _{CLK-SETTLE} (ns)	200
------------------------------	-----

Data Timer

T _{HS-SETTLE} (ns)	130
-----------------------------	-----

Red arrows from the text on the left point to the following fields: Calibration Clock Freq, T_{CLK-SETTLE}, and T_{HS-SETTLE}.

MIPI 控制器主要参数

UI : DHY 时钟周期

Block Editor

LPS Mode Lane Mapping Timing

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)	90
T _{HS-ZERO} (ns)	142
T _{HS-TRAIL} (ns)	106

$$T_{\text{CLK-Post}} \geq 60\text{ns} + 52 \cdot \text{UI}$$

$$T_{\text{CLK-Trail}} \geq 60\text{ns}$$

$$T_{\text{CLK-Prepare}} : 38\text{ns} \text{ --- } 95\text{ns}$$

$$T_{\text{CLK-Zero}} \geq 300\text{ns} - T_{\text{CLK-Prepare}}$$

$$F_{\text{Escape-clk}} : 11\text{Mhz} \text{ --- } 20\text{Mhz}$$

T_{CLK-Pre} : Varies depending on the escape clock frequency

$$T_{\text{HS-Prepare}} \geq 85\text{ns} + 4 \cdot \text{UI}$$

$$T_{\text{HS-Zero}} \geq 145\text{ns} + 10 \cdot \text{UI} - T_{\text{HS-Prepare}}$$

$$T_{\text{HS-Trail}} \geq \max(n \cdot 8 \cdot \text{UI}, 60\text{ns} + n \cdot 4 \cdot \text{UI})$$

Block Editor

Status Lane Mapping Timing

Calibration Clock Freq (MHz) 100.00

Clock Timer

T _{CLK-SETTLE} (ns)	200
------------------------------	-----

Data Timer

T _{HS-SETTLE} (ns)	130
-----------------------------	-----

$$F_{\text{Calibration-clk}} : 80\text{Mhz} \text{ --- } 120\text{Mhz}$$

$$T_{\text{CLK-Settle}} : 95\text{ ns} \text{ --- } 300\text{ ns}$$

$$T_{\text{HS-Settle}} : 85\text{ ns} + 6 \cdot \text{UI} \text{ --- } 145\text{ ns} + 10 \cdot \text{UI}$$

MIPI 控制器主要参数

- MIPI控制器参数所要满足的Timing要求
 - MIPI Timing计算说明v1.1.ppt
 - MIPI-CSI-2-协议介绍.ppt
 - mipi-utility-v1.1.xlsm

Block Editor

LPS Mode Lane Mapping Timing

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)	90
T _{HS-ZERO} (ns)	142
T _{HS-TRAIL} (ns)	106

Block Editor

Status Lane Mapping Timing

Calibration Clock Freq (MHz) 100.00

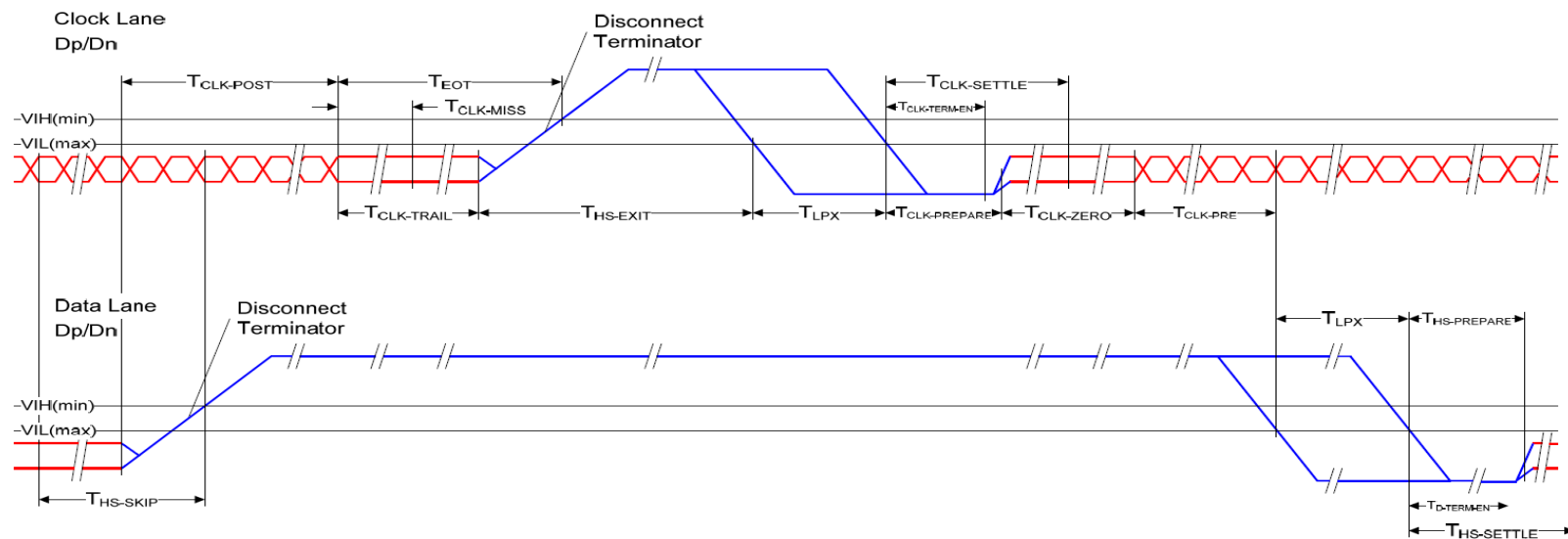
Clock Timer

T _{CLK-SETTLE} (ns)	200
------------------------------	-----

Data Timer

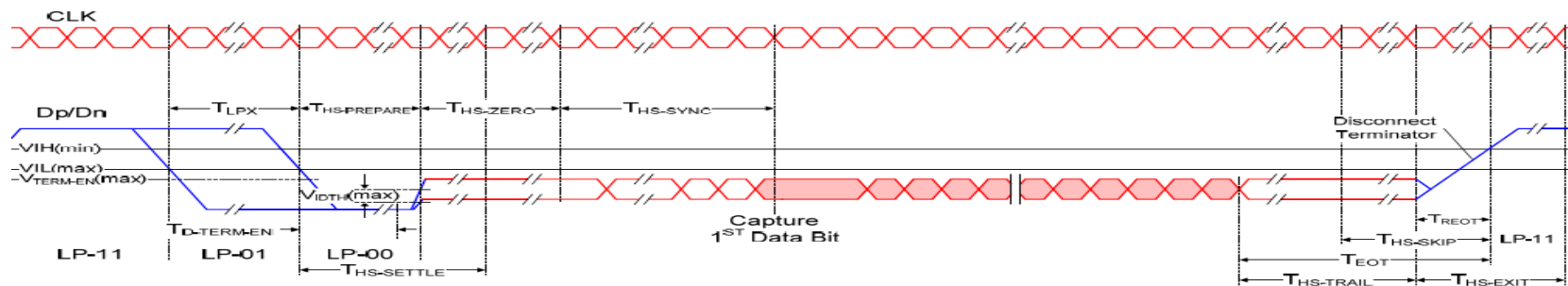
T _{HS-SETTLE} (ns)	130
-----------------------------	-----

MIPI clock lane工作状态切换



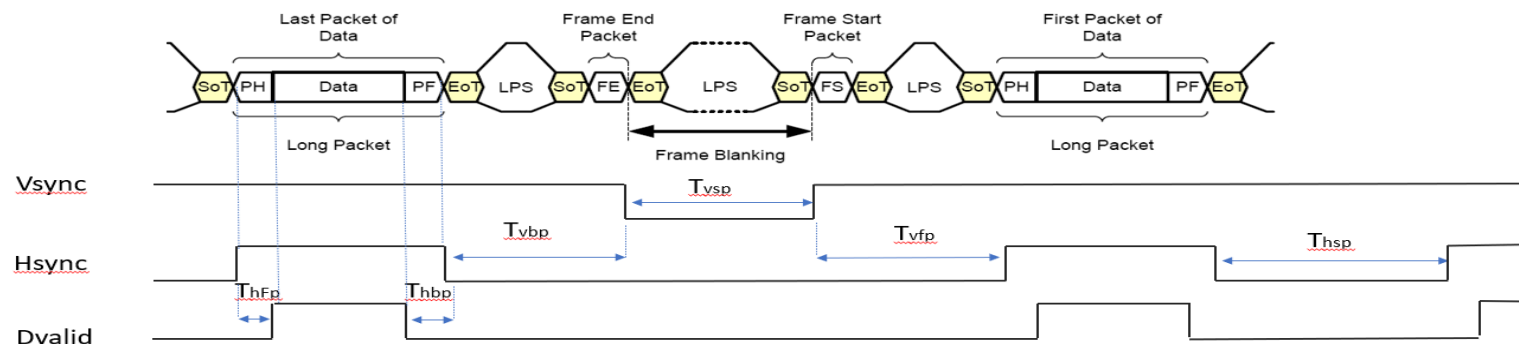
- Clk lane – HS -----> LP
 - 在所有data lane进入LP模式后，时钟运行 $T_{clk-post}$ 时间后，以HS-0状态结束
 - 在HS-0状态保持 $T_{clk-trail}$ 时间
 - CLK Lane 从HS-TX状态到LP-TX状态，直到进入LP-11状态，需要 $T_{HS-EXIT}$
- Clk lane – LP -----> HS
 - 从LP-11 → LP-01,在LP-01状态持续 T_{LPX} 时间
 - 从LP-01 → LP-00,需要 $T_{clk-prepare}$ 时间
 - HS使能，LP不使能，在HS-0状态持续 $T_{clk-zero}$ 时间
 - 在任何一个data lane转到HS之前，CLK-HS保持 $T_{clk-pre}$ 时间

MIPI data lane工作状态切换



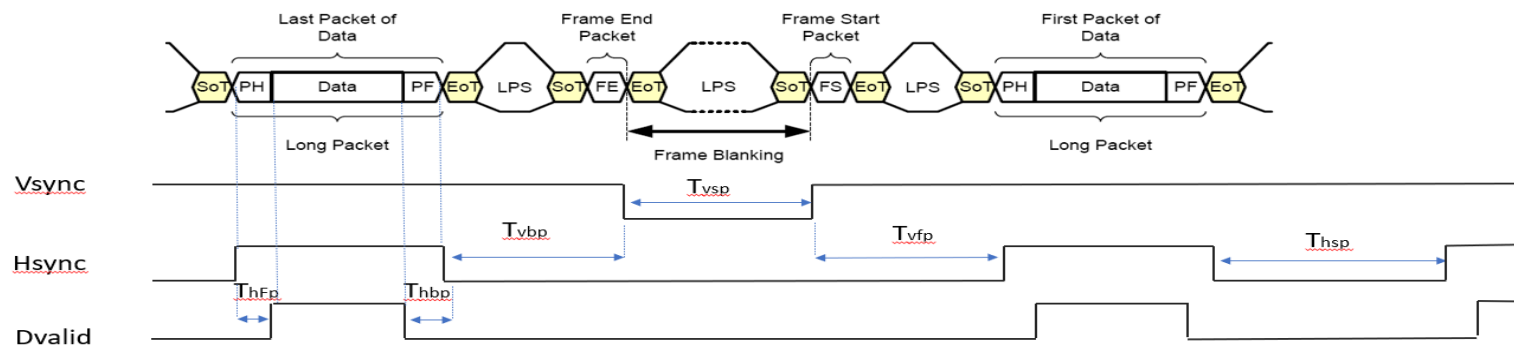
- 从LP -> HS状态
 - 从LP-11---LP-01,HS-Rqst持续 T_{LPX} 时间
 - 进入LP-00状态持续 $T_{HS-PREPARE}$ 时间
 - 使能HS 关闭LP, 进入HS-0持续 $T_{HS-ZERO}$ 时间
 - 发送HS SYNC-Seq '00011101'
 - 持续发送HS数据
- 从HS -> LP状态
 - 传输完最后的数据后保持 $T_{HS-TRAIL}$ 时间
 - 关闭 HS-TX 使能LP-TX, 进入LP-11状态保持 $T_{HS-EXIT}$ 时间

行消隐时间-- T_{hsp} T_{hfp} T_{hbp} 计算



- $T_{hfp} = PH = 1/(DHY-Preq) * 32$, 最少2个pixel clock
- $T_{hbp} = PF = 1/(DHY-Preq) * 16$, 最少2个pixel clock
- T_{hsp} 参数
 - 可由MIPI Utility.xlsm计算文档得到
 - 根据MIPI Timing计算说明v1.1文档中的计算公式得到

场消隐时间--- T_{vsp} T_{vfp} T_{vbp} 计算

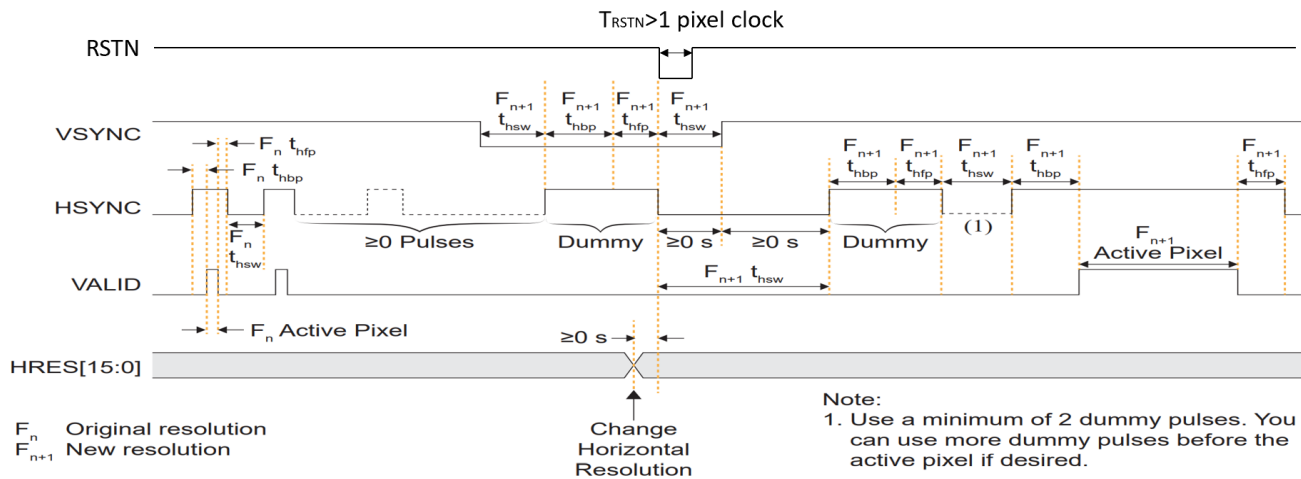


- 根据实测 T_{vsp} , T_{vfp} , T_{vbp} 的值至少是1个blanking line, 以blanking line 为单位增加;
- $T\text{-blanking line} = T_{hfp} + T_{hbp} + T_{hsp}$

Enter the video information	
Data type	RAW8
# of data pixels per line	1920
# of data lines per frame	1080
# of blanking lines per frame	3
# of frames per second (Hz)	30
Horizontal blanking per line (us)	10
Pixel clock frequency (MHz)	100

在MIPI Utility 文档中
#of banking lines per frame的
最小值为3

MIPI 分辨率动态切换时间参数说明



- MIPI TX分辨率进行动态切换需满足上图时序
 - 要进行分辨率动态切换 T_{vsp} 需大于2个banking line
 - 在VSYNC low 状态持续时间大于 T_{hsp} 后改变行像素参数HRES[15:0]
 - 在HRES[15:0]改变大于1个pixel clock 时间后 RSTN low ， 仅复位MIPI CSI
 - RSTN low 时间大于1个Pixel clock

MIPI Demo功能简介

- **MIPI发送**
 - 通过VIO配置MIPI-Tx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；

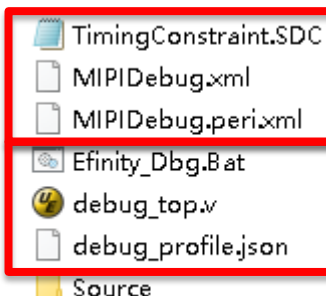
用途

- 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

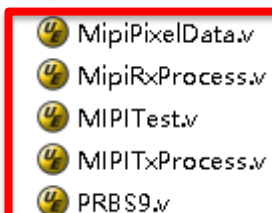
与项目（Project）有关的文件：

- 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：

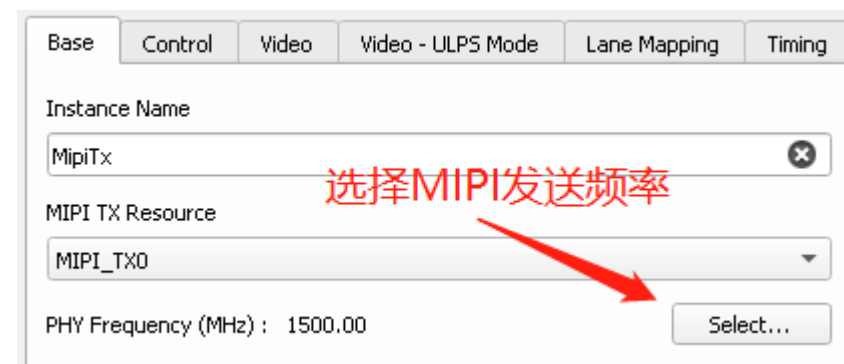
- MipiPixelData.v ——像素数据发送和检测；
- MipiRxprocess.v ——MIPI接收测量和状态检测；
- MipiTest.v ——MIPI演示顶层文件；
- MipiTxProcess.V ——MIPI发送参数计算和控制；
- PRBS9.v ——伪随机码产生和检测；

```
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；



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;

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

MainWindow

Specify JTAG Chain ? Ready

Run

LA_MipiRx

Configuration

USB Target T20F169 Development Board USER1 Configure

JTAG Info Rtd:/Dx0403:0x6010:FT4DE3J1/2 Unconfig

JTAG chain chip-select Import a JCF file before configure to use a JTAG chain. Import debug profile each time the target chip in chain is changed.

Profile Instructions Program

Program

Configuration

USB Target T20F169 Development Board

USB Info Bus 000 Device 003: ID 0403:6010

Image x:\Design\Hik-vision\Design\MIPITest\outflow\MPIIDebug.hex T20F169 Select Combine

Program Mode SPI Passive Default

Start Program Stop Program

Run Run Immediate Stop

Core status Idle Waiting for Trigger Post-Trigger Full

Capture status Segment 1 of 1 Segment sample 0 of 1024 Total sample 512 of 1024 0% 0% 50%

Waveform Path x:\Design\Hik-vision\Design\MIPITest\LA_MipiRx_waveform.vcd Open Overwrite

Trigger Setup Trigger mode BASIC_ONLY Add Remove Trigger condition Global 'AND'

Name	Operator	Radix	Value	Port
MipiRxError	== (equal)	Bin	1 (logical one)	probe4

Capture Setup Capture mode ALWAYS Number of segments 1 [1 - 1] Window data depth 1024 [1 - 1024] Trigger position in window 512 [0 - 1023] Add Remove Capture condition Global 'AND'

Name	Operator	Radix	Value	Port
------	----------	-------	-------	------

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	场消隐时间, 单位us
ConfigTxHBlankTime_us	行消隐时间, 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
TxVertBlankLength_Cycle	两个VSYNC中的间隔长度; 用时钟周期标识
RealTxHoriPixelNumber	MIPI-Tx真实使用的行像素个数

说明:

1. MIPI-TX控制信号参数是根据图像参数自动计算的; 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;
2. TxPixelClk的频率与MIPI的物理接口频率无关; MIPI工作频率设定在Interface Designer的MIPI TX -> Base
3. 可以通过提高MipiTxPixelClk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:
 - MipiTxPixelClk > 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
↑	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-TX参数和控制信号

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

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

MIPI-TX控制信号

ControlTxDPHYReset	MIPI-TxDPHY复位信号，低有效
ControlTxCSIReset	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、ControlTxCSIReset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；

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

MIPI-RX控制信号

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

	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

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;

RX图像概况和MIPI-RX状态

RX图像概况

VidioRxFrameRate	接收图像的帧率
VidioRxVertPixeNumber	接收图像的列像素个数
VidioRxHoriPixeNumber	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

MIPI-RX状态

StaRxDataType	MIPI-Rx的数据类型
StateRxVCAActive	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）

说明：

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
↑	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
↑	StateRxVCAActive	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

Demo 的硬件准备

- T120F324开发板
- USB线
- 2个MIPI转接子板或2个LVDS转接子板，接到MIPI TX0和MIPI RX0接口上
- 扁平线或杜邦线

谢 谢 ！