

— LVDS硬件特性

1.1 相关概念

- **Lane**: 用于连接发送端和接收端的一堆告诉差分线，既可以是时钟 Lane，也可以是数据 Lane
- **Link**: 发送端和接收端之间的时钟Lane 和至少一个数据Lane 组成一个Link，本文中的 link 是一个软件概念，每一个link 包括两个数据lane。
- **LVDS**

低压差分信号，低功耗、低误码率、低串扰和低辐射的差分信号技术，核心是采用极低的电压摆幅告诉差动传输数据，可实现点对点或点对多的连接。

LVDS的基本工作原理如图 1-1所示，其源端驱动器由一个恒流源（通常约为3.5mA，最大不超过4mA）驱动一对差分信号线组成。接收端的接收器本身为高直流输入阻抗，所以几乎全部的驱动电流都流经 100Ω 的终端匹配电阻，并在接收器输入端产生约350mV的电压。当源端驱动状态反变化时，流经匹配电阻的电流方向改变，于是在接收端产生高低逻辑状态的变化。

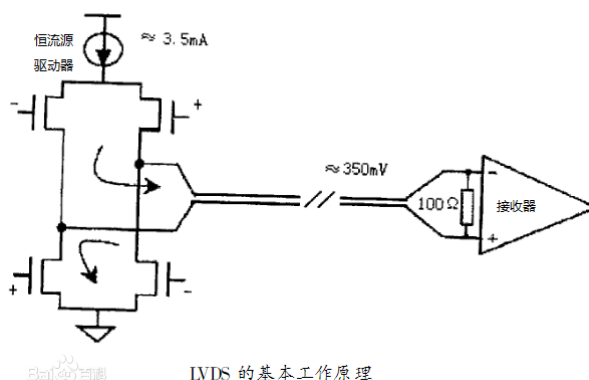


图 1-1 LVDS基本工作原理

特点:

- 高传输能力，理论极限 1.9Gbps
- 低功耗特性，LVDS器件采用CMOS工艺实现
- 供电电压低
- 较强的抗噪声能力，差分信号固有的优点就是噪声以共模的方式在一对差分线上耦合出现，并在接收器中相减，从而可消除噪声，所以LVDS具有较强的抗共模噪声能力
- 有效地抑制电磁干扰
- 时序定位精确

1.2 MIPI/LVDS电气参数

Sub-LVDS/LVDS 差分 DC 电气参数如图 1-2 所示. (《Hi3559A/C V100 ultra-HD Mobile Camera SoC 用户指南-2-25》)

符号	参数		最小值	典型值	最大值	单位
VIDTH(SL)	Differential Input Threshold Voltage $ (VP-VM) $	Sub-LVDS	70	-	-	mV
VIDTH(HS)		HiSPi(SLV S)	70	-	-	
VIDTH(HiV CM)		HiSPi(HiV CM)	100	-	-	
VIDTH(DP)		D-PHY HS	70	-	-	
VIDTH(LV)		LVDS	100	-	-	
VIDTH(ML)		Mini-LVDS	100	-	-	
VCM(SL)	Common Mode	Sub-LVDS	0.5	0.9	1.3	V

符号	参数		最小值	典型值	最大值	单位
VCM(HS)	Voltage Range $(VP+VM)/2$	HiSPi(SLV S)	0.07	0.2	0.35	
VCM(HiV CM)		HiSPi(HiV CM)	0.66	0.90	1.17	
VCM(DP)		D-PHY HS	0.07	0.2	0.33	
VCM(LV)		LVDS	0.925	1.2	1.475	
VCM(ML)		Mini-LVDS	1.025	1.2	1.375	
VISVR (SL)	Single-ended Input Voltage Range VP, VM	Sub-LVDS	0.4		1.4	V
VCM(HS)		HiSPi(SLV S)	-0.04		0.49	
VCM(HiV CM)		HiSPi(HiV CM)	0.55		1.35	
VCM(DP)		D-PHY HS	-0.04		0.46	
VCM(LV)		LVDS	0		1.8	
VCM(ML)		Mini-LVDS	0.825		1.575	
ZID(SL)	Internal Termination Resister Value	Sub-LVDS	80	100	125	Ω
ZID(HS)		HiSPi(SLV S)			125	
ZID(HiV CM)		HiSPi(HiV CM)			125	
ZID(LV)		LVDS			120	
ZID(ML)		Mini-LVDS			120	
ZID(DP)		D-PHY HS			125	

图 1-2 差分 DC 参数

MIPI Rx支持的接口类型如表 1-1 所示：（《Hi3559A/C V100 ultra-HD Mobile Camera SoC 用户指南-10-247》）

Interface Type	Common mode voltage	Differential mode voltage	Maximum clock frequency	Maximum data rate per lane
MIPI D-PHY	200mV	200mV	1250MHz	2.5Gbps
sub-LVDS	900mV	150mV	750MHz	1.5Gbps
LVDS	1.25V	350mV	750MHz	1.5Gbps
HiSPi(HiVCM)	900mV	280mV	750MHz	1.5Gbps
HiSPi(SLVS)	200mV	200mV	750MHz	1.5Gbps

由上图可知 LVDS 共模电压为**典型值 1.25 V**，提供最小摆幅 100mV，**差模电压350mV**，在接收器的输入端连接有一个 100Ω 的电阻

1.3 接口时序

1.3.1 VI 接口时序

VI 接口 CMOS 模式时序如图 1-3 所示。（Hi3559A/C V100 ultra-HD Mobile Camera SoC 用户指南-2-46）

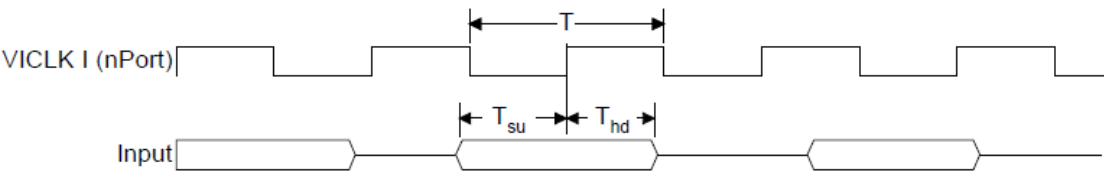


图 1 -3 VI CMOS 模式时序

VI 接口时序参数如表 1-1 所示

表 1-2 VI 接口时序参数

参数	符号	最小值	典型值	最大值	单位
VICLK时钟周期	T	6.73	-	-	ns
输入信号建立时间要求	T _{su}	1.9	-	-	ns
输入信号保持时间要求	T _{hd}	0.8	-	-	ns

1.3.2 MIPI Rx接口时序

MIPI Rx 接口时钟时序如图 1-4 所示：（《Hi3559A/C V100 ultra-HD Mobile Camera SoC 用户指南-2-53》）

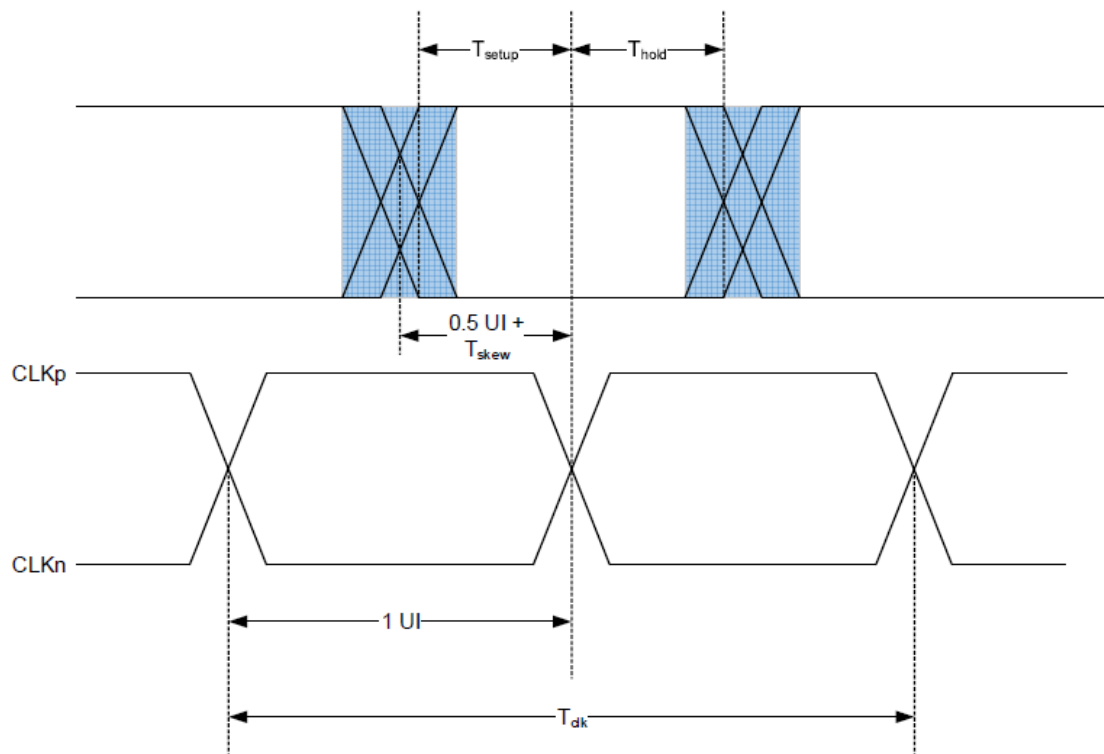


图 1-4 MIPI Rx 接口时钟时序

图中的 T_{skew} 包括 $T_{dynamic_skew}$ 和 T_{static_skew}

MIPI Rx 时序参数如表 1-2 所示：

表 1-3 MIPI Rx 时序参数

符号	参数	最小值	典型值	最大值	单位
FMAX	数据率	-	-	2.5G	bps
T_{clk}	差分时钟周期	0.8	T	-	ns
T_{static_skew}	静态skew时间 (1.5Gbps<data rate<=2.5Gbps)	0.2*UI	-	-	ns
$T_{dynamic_skew}$	动态skew时间 (T _{setup} +T _{hold}) (1.5Gbps<data rate<=2.5Gbps)	0.2*UI	-	-	ns
T_{setup}	差分时钟建立时间	0.15UI (80Mbps<data rate<=1.0Gbps) 0.2UI (1.0Gbps<data rate<=1.5Gbps)	-	-	ns
T_{hold}	差分始终保持时间	0.15UI (80Mbps<data rate<=1.0Gbps) 0.2UI (1.0Gbps<data rate<=1.5Gbps)	-	-	ns

注：UI = T/2

1.4 硬件管脚连接

对于目标板，FPGA与海思的硬件连接关系如图 1-5 。（《17所目标板原理图-MPSoC_HiSilicon_board Sheet Zynq_B64_B65_B66 HP》）

IO_L15P_T2L_N4_AD11P_66_U11	U11	
IO_L14N_T2L_N3_GC_66_Y8	Y8	MIP1_RX1_CK1N
IO_L14P_T2L_N2_GC_66_Y9	Y9	MIP1_RX1_CK1P
IO_L13N_T2L_N1_GC_QBC_66_V8	V8	
IO_L13P_T2L_N0_GC_QBC_66_U8	U8	
IO_T1U_N12_66_Y7	Y7	
IO_L12N_T1U_N11_GC_66_W6	W6	
IO_L12P_T1U_N10_GC_66_W7	W7	
IO_L11N_T1U_N9_GC_66_V6	V6	MIP1_RX1_D0N
IO_L11P_T1U_N8_GC_66_V7	V7	MIP1_RX1_D0P
IO_L10N_T1U_N7_QBC_AD4N_66_U6	U6	MIP1_RX0_CK1N
IO_L10P_T1U_N6_QBC_AD4P_66_U7	U7	MIP1_RX0_CK1P
IO_L9N_T1L_N5_AD12N_66_U4	U4	MIP1_RX0_D3N
IO_L9P_T1L_N4_AD12P_66_U5	U5	MIP1_RX0_D3P
IO_L8N_T1L_N3_AD5N_66_W4	W4	MIP1_RX0_D2N
IO_L8P_T1L_N2_AD5P_66_V4	V4	MIP1_RX0_D2P
IO_L7N_T1L_N1_QBC_AD13N_66_Y5	Y5	MIP1_RX1_D2N
IO_L7P_T1L_N0_QBC_AD13P_66_W5	W5	MIP1_RX1_D2P
IO_T0U_N12_VRP_66_V3	V3	R240 240
IO_L6N_T0U_N11_AD6N_66_U1	U1	MIP1_RX0_D1N
IO_L6P_T0U_N10_AD6P_66_T1	T1	MIP1_RX0_D1P
IO_L5N_T0U_N9_AD14N_66_U3	U3	
IO_L5P_T0U_N8_AD14P_66_U2	U2	
IO_L4N_T0U_N7_DBC_AD7N_66_V1	V1	MIP1_RX0_D0N
IO_L4P_T0U_N6_DBC_AD7P_66_V2	V2	MIP1_RX0_D0P
IO_L3N_T0L_N5_AD15N_66_W1	W1	MIP1_RX0_CK0N
IO_L3P_T0L_N4_AD15P_66_W2	W2	MIP1_RX0_CK0P
IO_L2N_T0L_N3_66_Y1	Y1	MIP1_RX1_D3N
IO_L2P_T0L_N2_66_Y2	Y2	MIP1_RX1_D3P
IO_L1N_T0L_N1_DBC_66_Y3	Y3	MIP1_RX1_D1N
IO_L1P_T0L_N0_DBC_66_Y4	Y4	MIP1_RX1_D1P
VREF_66_W9	W9	

IO_L24N_T3U_N11_PERSTN0_65_AB1	AB1	
IO_L24P_T3U_N10_PERSTN1_I2C_SDA_65_AA1	AA1	
IO_L23N_T3U_N9_65_AA2	AA2	MIP1_RX2_D1N
IO_L23P_T3U_N8_I2C_SCLK_65_AA3	AA3	MIP1_RX2_D1P
IO_L22N_T3U_N7_DBC_AD0N_65_AD1	AD1	MIP1_RX1_CK0N
IO_L22P_T3U_N6_DBC_AD0P_65_AC1	AC1	MIP1_RX1_CK0P
IO_L21N_T3L_N5_AD8N_65_AD2	AD2	MIP1_RX2_D3N
IO_L21P_T3L_N4_AD8P_65_AC2	AC2	MIP1_RX2_D3P
IO_L20N_T3L_N3_AD1N_65_AE2	AE2	MIP1_RX3_CK0N
IO_L20P_T3L_N2_AD1P_65_AE3	AE3	MIP1_RX3_CK0P
IO_L19N_T3L_N1_DBC_AD9N_65_AC3	AC3	MIP1_RX2_CK0N
IO_L19P_T3L_N0_DBC_AD9P_65_AB3	AB3	MIP1_RX2_CK0P
IO_T2U_N12_65_AE7	AE7	
IO_L18N_T2U_N11_AD2N_65_AC4	AC4	MIP1_RX3_D3N
IO_L18P_T2U_N10_AD2P_65_AB4	AB4	MIP1_RX3_D3P
IO_L17N_T2U_N9_AD10N_65_AE4	AE4	MIP1_RX3_D2N
IO_L17P_T2U_N8_AD10P_65_AD4	AD4	MIP1_RX3_D2P
IO_L16N_T2U_N7_QBC_AD3N_65_AE5	AE5	MIP1_RX3_D1N
IO_L16P_T2U_N6_QBC_AD3P_65_AD5	AD5	MIP1_RX3_D1P
IO_L15N_T2L_N5_AD11N_65_AA5	AA5	MIP1_RX2_D2N
IO_L15P_T2L_N4_AD11P_65_AA6	AA6	MIP1_RX2_D2P
IO_L14N_T2L_N3_GC_65_AD6	AD6	MIP1_RX3_D0N
IO_L14P_T2L_N2_GC_65_AC6	AC6	MIP1_RX3_D0P
IO_L13N_T2L_N1_GC_QBC_65_AB5	AB5	MIP1_RX2_CK1N
IO_L13P_T2L_N0_GC_QBC_65_AB6	AB6	MIP1_RX2_CK1P
IO_T1U_N12_65_AE8	AE8	
IO_L12N_T1U_N11_GC_65_AC8	AC8	
IO_L12P_T1U_N10_GC_65_AB8	AB8	
IO_L11N_T1U_N9_GC_65_AD7	AD7	MIP1_RX3_CK1N
IO_L11P_T1U_N8_GC_65_AC7	AC7	MIP1_RX3_CK1P
IO_L10N_T1U_N7_QBC_AD4N_65_AA7	AA7	MIP1_RX2_D0N
IO_L10P_T1U_N6_QBC_AD4P_65_AA8	AA8	MIP1_RX2_D0P
IO_L0N_T1U_N5_AD12N_65_AE9	AE9	

图 1-5 目标板 FPGA 与海思的硬件连接

海思侧的MIPI Rx 管脚定义如图 1-6 所示，以 4*4lane 为例介绍，详细配置参照《Hi3559AV100_PINOUT_CN》

接口	管脚	(4Lane*4)		
		电平 (v)	功能	VI
MIPI_RX0	MIPI_RX0_D2N	1.8	MIPI_RX0_D2N	sensor0
	MIPI_RX0_D2P	1.8	MIPI_RX0_D2P	
	MIPI_RX0_D0N	1.8	MIPI_RX0_D0N	
	MIPI_RX0_D0P	1.8	MIPI_RX0_D0P	
	MIPI_RX0_CK0N	1.8	MIPI_RX0_CK0N	
	MIPI_RX0_CK0P	1.8	MIPI_RX0_CK0P	
	MIPI_RX0_CK1N	1.8	MIPI_RX0_CK1N	
	MIPI_RX0_CK1P	1.8	MIPI_RX0_CK1P	
	MIPI_RX0_D1N	1.8	MIPI_RX0_D1N	
	MIPI_RX0_D1P	1.8	MIPI_RX0_D1P	
	MIPI_RX0_D3N	1.8	MIPI_RX0_D3N	
	MIPI_RX0_D3P	1.8	MIPI_RX0_D3P	
MIPI_RX1	MIPI_RX1_D2N	1.8	MIPI_RX1_D2N	sensor1
	MIPI_RX1_D2P	1.8	MIPI_RX1_D2P	
	MIPI_RX1_D0N	1.8	MIPI_RX1_D0N	
	MIPI_RX1_D0P	1.8	MIPI_RX1_D0P	
	MIPI_RX1_CK0N	1.8	MIPI_RX1_CK0N	
	MIPI_RX1_CK0P	1.8	MIPI_RX1_CK0P	
	MIPI_RX1_CK1N	1.8	MIPI_RX1_CK1N	
	MIPI_RX1_CK1P	1.8	MIPI_RX1_CK1P	
	MIPI_RX1_D1N	1.8	MIPI_RX1_D1N	
	MIPI_RX1_D1P	1.8	MIPI_RX1_D1P	
	MIPI_RX1_D3N	1.8	MIPI_RX1_D3N	
	MIPI_RX1_D3P	1.8	MIPI_RX1_D3P	

接口	管脚	(4Lane*4)		
		电平 (v)	功能	VI
MIPI_RX2	MIPI_RX2_D2N	1.8	MIPI_RX2_D2N	sensor2
	MIPI_RX2_D2P	1.8	MIPI_RX2_D2P	
	MIPI_RX2_D0N	1.8	MIPI_RX2_D0N	
	MIPI_RX2_D0P	1.8	MIPI_RX2_D0P	
	MIPI_RX2_CK0N	1.8	MIPI_RX2_CK0N	
	MIPI_RX2_CK0P	1.8	MIPI_RX2_CK0P	
	MIPI_RX2_CK1N	1.8	MIPI_RX2_CK1N	
	MIPI_RX2_CK1P	1.8	MIPI_RX2_CK1P	
	MIPI_RX2_D1N	1.8	MIPI_RX2_D1N	
	MIPI_RX2_D1P	1.8	MIPI_RX2_D1P	
	MIPI_RX2_D3N	1.8	MIPI_RX2_D3N	
	MIPI_RX2_D3P	1.8	MIPI_RX2_D3P	
MIPI_RX3	MIPI_RX3_D2N	1.8	MIPI_RX3_D2N	sensor3
	MIPI_RX3_D2P	1.8	MIPI_RX3_D2P	
	MIPI_RX3_D0N	1.8	MIPI_RX3_D0N	
	MIPI_RX3_D0P	1.8	MIPI_RX3_D0P	
	MIPI_RX3_CK0N	1.8	MIPI_RX3_CK0N	
	MIPI_RX3_CK0P	1.8	MIPI_RX3_CK0P	
	MIPI_RX3_CK1N	1.8	MIPI_RX3_CK1N	
	MIPI_RX3_CK1P	1.8	MIPI_RX3_CK1P	
	MIPI_RX3_D1N	1.8	MIPI_RX3_D1N	
	MIPI_RX3_D1P	1.8	MIPI_RX3_D1P	
	MIPI_RX3_D3N	1.8	MIPI_RX3_D3N	
	MIPI_RX3_D3P	1.8	MIPI_RX3_D3P	

图 1-6 4lane 模式下的 MIPI Rx 管脚分配

Hi3559AV100 的 MIPI Rx 最大支持 8Lane MIPI 输入或者 16 Lane LVDS 输入，MIPI Rx 能同时对接多个设备（sensor），Hi3559AV100 最多对接的设备数（sensor数）为 8 个。

MIPI Rx 最大能同时对接不同数量的设备 sensor，每个 sensor 需要的 Lane 数量也不同，因此需要用户确定 MIPI Rx 的 Lane 分布模式，具体的 Lane 分布模式如表 1-3 所示：

表 1-3 MIPI Rx Lane 分布模式

芯片类型	Mode	DEV0	DEV1	DEV2	DEV3	DEV4	DEV5	DEV6	DEV7
Hi3559AV100	0	L0~L15	N	N	N	N	N	N	N
	1	L0~L11	N	N	N	N	N	L12~L15	N
	2	L0~L11	N	N	N	N	N	L12 L14	L13 L15
	3	L0~L7	N	N	N	L8~L15	N	N	N
	4	L0~L7	N	N	N	L8~L11	N	L12~L15	N
	5	L0~L7	N	N	N	L8~L11	N	L12 L14	L13 L15
	6	L0~L7	N	N	N	L8 L10	L9 L11	L12 L14	L13 L15
	7	L0~L3	N	L4~L7	N	L8~L11	N	L12~L15	N
	8	L0~L3	N	L4~L7	N	L8~L11	N	L12 L14	L13 L15
	9	L0~L3	N	L4~L7	N	L8 L10	L9 L11	L12 L14	L13 L15
	A	L0~L3	N	L4 L6	L5 L7	L8 L10	L9 L11	L12 L14	L13 L15
	B	L0 L2	L1 L3	L4 L6	L5 L7	L8 L10	L9 L11	L12 L14	L13 L15

注：依据来自《MIPI使用指南-5》

采用 4Lane 分布模式情况下，对应于 图 1-6 的管脚分配。

首先将 MIPI Rx 分为 MIPI RX0、MIPI RX1、MIPI RX2、MIPI RX3 四个组，每个组中具有 4 Lane数据线和 2 Lane 时钟线。

MIPI_RX0_CK0P、MIPI_RX0_CK0N表示差分时钟周期管脚所在位置，MIPI_RX0_D0P、MIPI_RX0_D0N即表示第一对差分线，也就是Lane0；MIPI_RX0_D1P、MIPI_RX0_D1N即表示第二对差分线，也就是Lane1；依次类推，总共有 16 Lane，编号依次是L0、L1、L2、L3、L4、L5、L6、L7、L8、L9、L10、L11、L12、L13、L14、L15。

采用 4 Lane*4的模式，与 FPGA 的连接关系如表 1-4 所示：

表 1-4 Hi3559AV100与FPGA连接关系举例（4Lane*4模式）

FPGA内部管脚	外部接口引脚名	Hi3559AV100	Lane
IO_L3P_T0L_N4_AD15P_66_W2	W2	MIPI_RX0_CK0P	差分时钟
IO_L3N_T0L_N5_AD15N_66_W1	W1	MIPI_RX0_CK0N	差分时钟
IO_L4P_T0U_N6_DBC_AD7P_66_V2	V2	MIPI_RX0_D0P	Lane0
IO_L4N_T0U_N7_DBC_AD7N_66_V1	V1	MIPI_RX0_D0N	Lane0
IO_L6P_T0U_N10_AD6P_66_T1	T1	MIPI_RX0_D1P	Lane1
IO_L6N_T0U_N11_AD6N_66_U1	U1	MIPI_RX0_D1N	Lane1
IO_L8P_T1L_N2_AD5P_66_V4	V4	MIPI_RX0_D2P	Lane2
IO_L8N_T1L_N3_AD5N_66_W4	W4	MIPI_RX0_D2N	Lane2
IO_L9P_T1L_N4_AD12P_66_U5	U5	MIPI_RX0_D3P	Lane3
IO_L9N_T1L_N5_AD12N_66_U4	U4	MIPI_RX0_D3N	Lane3

二 MIPI Rx 介绍

依据《Hi3559A/C V100 ultra-HD Mobile Camera SoC 用户指南-10-244》

2.1 概述

MIPI Rx（移动行业处理器接口）通过低电压差分信号接受原始视频数据（BAYER RGB数据），并将其转化为 DC 时序后传递给下一级模块 VICAP。MIPI Rx包含combo-PHY和Controller两部分，功能框图及在系统中的位置如图 2-1 所示。

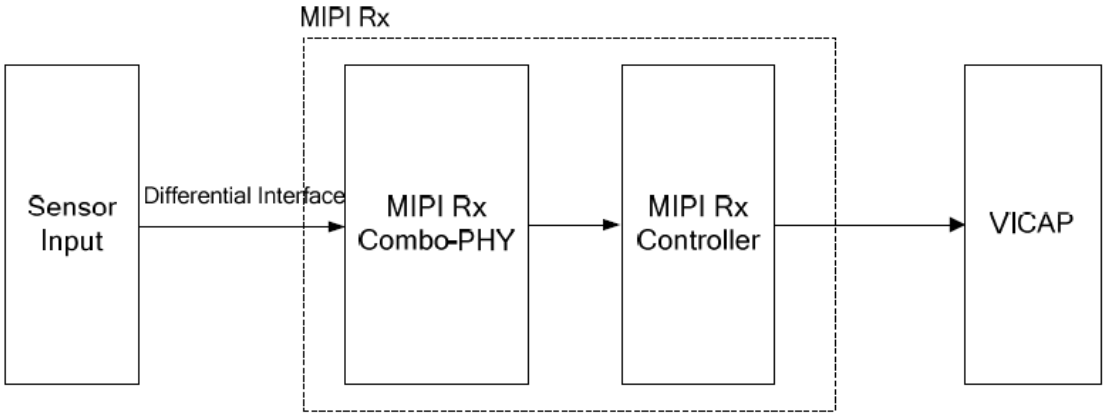


图 2-1 MIPI Rx 功能框图和在系统中的位置

一些概念的说明：

- Lane指差分数据对，MIPI_Rx最多有16 条Lane
- Link 指Lane的分组，每个分组中包含4对数据，MIPI_Rx有4 个Link
- 通道指 MIPI_Rx内部的数据处理通道，对应连接到VICAP 通道，每个通道单独处理一个 Sensor的数据

MIPI Rx 的特点：

- 可同时支持8路sensor输入
- 单路最多支持8-Lane MIPI D-PHY 接口，最大支持2.5Gbps/Lane

- **单路最多支持16-Lane LVDS/ sub-LVDS /HiSPi 接口，最大支持1.5Gbps/Lane**
- 支持RAW8/ RAW10/ RAW12/ RAW14/ RAW16 数据类型的解析
- 支持YUV420 8-bit/ YUV420 10-bit/ YUV422 8-bit/ YUV422 10-bit/Legacy YUV4208-bit数据类型的解析
- 支持 LVDS/HiSPi模式像素/同步码大小端配置
- **通道 0支持一拍双像素输出**

2.2 功能描述

MIPI Rx包括4 个D-PHY，每个PHY 各自有两对差分随路时钟（CLK0/CLK1），每对时钟对应2对数据。MIPI Rx 对接场景分类如图 2-2 所示。

Mode	PHY0		PHY1		PHY2		PHY3	
0	Sensor0(phy0clk0+lane0~15)							
1	Sensor0(phy0clk0+lane0~11)						Sensor1(phy3clk0+lane12~lane15)	
2	Sensor0(phy0clk0+lane0~11)						Sensor1(p hy3clk0+lane12/14)	Sensor2(p hy3clk1+lane13/15)
3	Sensor0(phy0clk0+lane0~7)				Sensor1(phy2clk0+lane8~15)			
4	Sensor0(phy0clk0+lane0~7)				Sensor1(phy2clk0+lane8~11)		Sensor2(phy3clk0+lane12~15)	
5	Sensor0(phy0clk0+lane0~7)				Sensor1(phy2clk0+lane		Sensor2(p	Sensor3(p
					8~11)		hy3clk0+lane12/14)	hy3clk1+lane13/15)
6	Sensor0(phy0clk0+lane0~7)				Sensor1(p hy2clk0+lane8/10)	Sensor2(p hy2_clk1+lane9/11)	Sensor3(p hy3+clk0+lane12/14)	Sensor4(p hy3clk1+lane13/15)
7	Sensor0(phy0clk0+lane0~3)		Sensor1(phy1clk0+lane4~7)		Sensor2(phy2clk0+lane8~11)		Sensor3(phy3clk0+lane12~15)	
8	Sensor0(phy0clk0+lane0~3)		Sensor1(phy1clk0+lane4~7)		Sensor2(phy2clk0+lane8~11)		Sensor3(p hy3+clk0+lane12/14)	Sensor4(p hy3clk1+lane13/15)
9	Sensor0(phy0clk0+lane0~3)		Sensor1(phy1clk0+lane4~7)		Sensor2(p hy2clk0+lane8/10)	Sensor3(p hy2_clk1+lane9/11)	Sensor4(p hy3+clk0+lane12/14)	Sensor5(p hy3clk1+lane13/15)
10	Sensor0(phy0clk0+lane0~3)		Sensor1(p hy1clk0+lane4/6)	Sensor2(p hy1clk1+lane5/7)	Sensor3(p hy2clk0+lane8/10)	Sensor4(p hy2clk1+lane9/11)	Sensor5(p hy3clk0+lane12/14)	Sensor6(p hy3clk1+lane13/15)
11	Sensor0(phy0clk0+lane0/2)	Sensor1(p hy0clk1+lane1/3)	Sensor2(p hy1clk0+lane4/6)	Sensor3(p hy1clk1+lane5/7)	Sensor4(p hy2clk0+lane8/10)	Sensor5(p hy2clk1+lane9/11)	Sensor6(p hy3clk0+lane12/14)	Sensor7(p hy3clk1+lane13/15)

图 2-2 MIPI Rx 对接场景分类

MIPI Rx只完成接口的时序转换，不处理图像的数据格式。Combo-PHY 支持的最大速率为**2.5Gbps/Lane**，总共支持16 Lane 同时传输。对于 MIPI Rx 控制器的来说，**通道的最大工作时钟为600MHz**，所以内部最大处理速度 600M*2pixels/s。

2.3 LVDS 接口数据格式

电压差分信号通过同步码区分消隐区和有效区的数据。

MIPI Rx 的 **combo-PHY** 将差分串行数据转换为并行数据，MIPI Rx控制器把并行数据拆分、拼接，然后提取同步码、解析出像素数据。

在 LVDS 的传输模式中，行场同步信号集成在数据流中，数据流中的特殊码型 **SOF** 和 **EOF** 分别表示帧的起始和结束，**SOL** 和 **EOL** 分别表示行的起始和结束。

SOF/EOF/SOL/EOL由4 个字段构成，**每个字段的位宽与像素数据保持一致**；根据**第4个字段来区分帧/行的起始或结束**，LVDS 同步码格式如图 2-3 所示：

Field	Bit Width	Sync code			
		SOL/SAV (Valid line)	EOL/EAV (Valid line)	SOF/SAV (Invalid line)	EOF/EAV (Invalid line)
1st code	8bit	FFh	FFh	FFh	FFh
	10bit	3FFh	3FFh	3FFh	3FFh
	12bit	FFFh	FFFh	FFFh	FFFh
	14bit	3FFFh	3FFFh	3FFFh	3FFFh
	16bit	FFFFh	FFFFh	FFFFh	FFFFh
2nd code	8bit	00h	00h	00h	00h
	10bit	000h	000h	000h	000h
	12bit	000h	000h	000h	000h
	14bit	0000h	0000h	0000h	0000h
	16bit	0000h	0000h	0000h	0000h
3rd code	8bit	00h	00h	00h	00h
	10bit	000h	000h	000h	000h
	12bit	000h	000h	000h	000h
	14bit	0000h	0000h	0000h	0000h
	16bit	0000h	0000h	0000h	0000h
4th code	8bit	XXh	XXh	XXh	XXh
	10bit	XXXh	XXXh	XXXh	XXXh
	12bit	XXXh	XXXh	XXXh	XXXh
	14bit	XXXXh	XXXXh	XXXXh	XXXXh
	16bit	XXXXh	XXXXh	XXXXh	XXXXh

图 2-3 LVDS 同步码格式

同步码前三个字段固定，第4个字段标识行场的起始或结束。第4个字段的值由图像传感器厂商确定，我们采用文档中的格式，如图 2-4 所示：

Field	Bit Width	Sync code			
		SAV(Valid line)	EAV(Valid line)	SAV(Invalid line)	EAV(Invalid line)
4th code	8bit	80h	9Dh	ABh	B6h
	10bit	200h	274h	2ACh	2D8h
	12bit	800h	9D0h	AB0h	B60h
	14bit	2000h	2740h	2AC0h	2D80h
	16bit	8000h	9D00h	AB00h	B600h

图 2-4 LVDS 同步码的第四个字段格式

注意：每个字段的数据位宽与像素数据保持一致

以 4Lane 为例，LVDS 同步码和像素数据在各个 Lane 上传输方式如图 2-5 所示，H 表示同步码，P 表示像素，H 和 P 的位宽与图像传感器输出的单个像素的位宽一致。各个数据通道首先传输 4 个像素位宽的同步码，紧接着是像素数据，像素数据的分布与通道数有关。

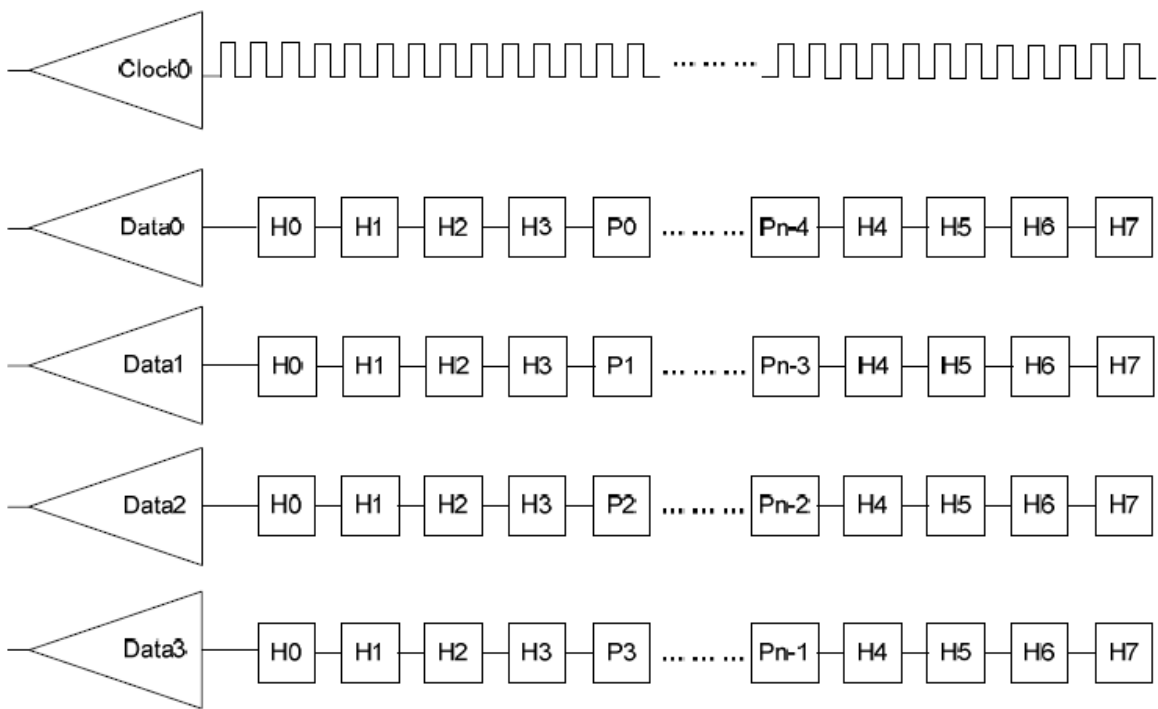


图 2-5 LVDS 同步码和图像传输模式

同步码和像素数据的传输是串行的，MIPI Rx支持数据的大小端可配置。以 RAW12、大端模式为例，图像传感器输出单个像素点的时序如图 2-6 所示。

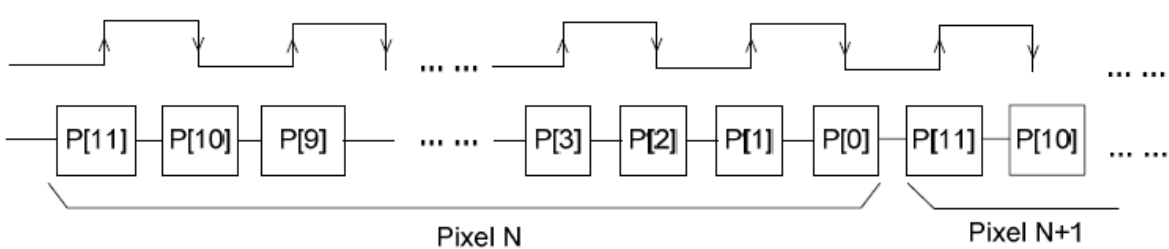


图 2-6 LVDS 单个像素点时序

2.4 LVDS接口线性模式

LVDS 有两种同步方式，一种使用 SAV (Invalid) 和EAV (Invalid) 标识消隐区的无效数据，使用 SAV (Valid)和 EAV (Valid) 标识有效像素区。同步方式如图 2-7 所示。

H.BLK	SAV (Invalid line)	V.BLK	EAV (Invalid line)	H.BLK
H.BLK		V.BLK		H.BLK
H.BLK		V.BLK		H.BLK
H.BLK	SAV (Valid line)	Effective Pixel	EAV (Valid line)	H.BLK
H.BLK		Effective Pixel		H.BLK
⋮		⋮		⋮
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK	SAV (Invalid line)	V.BLK	EAV (Invalid line)	H.BLK
⋮		⋮		⋮
H.BLK		V.BLK		H.BLK
H.BLK		V.BLK		H.BLK

图 2-7 LVDS 同步方式 (SAV-EAV)

2.5 LVDS模式配置流程

LVDS/HiSPi模式下需要配置RAW DATA 类型、数据大小端、同步方式、WDR类型和图像宽高等寄存器。LVDS 模式依靠同步码识别帧/行同步信息，根据RAW DATA 类型的不同，同步码可以为 8/10/12/14/16-bit

- 步骤 1. 上电启动。
- 步骤 2. 根据使用场景将MISC_CTRL130 寄存器中相应通道的mipi_work_mode配置为LVDS模式。
- 步骤 3. 配置CRG寄存器中的PERI_CRG61，打开mipi_bus_clken、mipi_cil_clken 以及对应通道的 mipi_pix_clken。配置mipi总线软复位，撤销复位；配置对应通道pix_core复位，撤销复位。
- 步骤 4. 配置CRG寄存器中的PERI_CRG69，配置sensor复位，撤销复位；打开sensor时钟门控，并配置时钟频率。
- 步骤 5. 配置CRG寄存器中的PERI_CRG60，选择MIP_Rx通道时钟频率。
- 步骤 6. 配置接收数据类型、WDR模式，图像宽高（**LVDS模式下，配置的宽度是图像实际宽度除以Lane数-1**）、同步头、Lane ID 等信息。
- 步骤 7. 配置PHY的工作模式（PHY_MODE_LINK*）、PHY 通道延迟调节（PHY_SKEW_LINK *）、PHY通道使能（PHY_EN_LINK *）、PHY均衡调节（PHY_EQ_LINK *）、PHY 性能调节（PHY_CFG_LINK *）；配置LVDS模式Lane同步头信息（PHY_SYNC_CODE * _LINK *）。
- 步骤 8. 配置系统控制寄存器。场景模式选择（HS_MODE_SELECT）、PHY_EN、LANE_EN，打开 PHY_CIL_CTRL、选择PHYCFG_MODE（对于LVDS 模式，应选择1）。
- 步骤 9. 配置对应的PHYCFG_EN。
- 步骤 10. 配置sensor序列。

注意：统一采用单沿采样，Hi3559AV100 侧需要软件层次的配置。

2.6 MIPI 使用指南

依据《MIPI使用指南-3》

MIPI Rx 是一个支持多种差分视频输入接口的采集单元，通过combo-PHY 接收MIPI/LVDS/sub-LVDS/HiSPi/DC 接口的数据，通过不同的功能模式配置，MIPI Rx 可以支持多种速度和分辨率的数据传输需求，支持多种外部输入设备。最大支持Lane 个数如表 2-1 所示。

表 2-1 最大支持的 Lane 数

芯片类型	最大支持 lane 数
Hi3559AV100	MIPI Rx 最大支持8Lane MIPI 输入或16Lane LVDS 输入。

Hi3559AV100 最大能对接 8 个sensor，MIPI Rx 最大能同时对接不同数量的 sensor，每个 sensor 需要的 Lane 也不尽相同，需要自己定义 LANE 分布模式，具体的分布模式如图 2-8 所示。

芯片类型	Mode	DEV0	DEV1	DEV2	DEV3	DEV4	DEV5	DEV6	DEV7
Hi3559AV100	0	L0~L15	N	N	N	N	N	N	N
	1	L0~L11	N	N	N	N	N	L12~L15	N
	2	L0~L11	N	N	N	N	N	L12 L14	L13 L15
	3	L0~L7	N	N	N	L8~L15	N	N	N
	4	L0~L7	N	N	N	L8~L11	N	L12~L15	N
	5	L0~L7	N	N	N	L8~L11	N	L12 L14	L13 L15
	6	L0~L7	N	N	N	L8 L10	L9 L11	L12 L14	L13 L15
	7	L0~L3	N	L4~L7	N	L8~L11	N	L12~L15	N
	8	L0~L3	N	L4~L7	N	L8~L11	N	L12 L14	L13 L15
	9	L0~L3	N	L4~L7	N	L8 L10	L9 L11	L12 L14	L13 L15
	A	L0~L3	N	L4 L6	L5 L7	L8 L10	L9 L11	L12 L14	L13 L15
	B	L0 L2	L1 L3	L4 L6	L5 L7	L8 L10	L9 L11	L12 L14	L13 L15

图 2-8 MIPI Rx 的 Lane 分布模式

注意：我们采用 4lane*4 的分布模式，即模式 7

Lane id 的配置

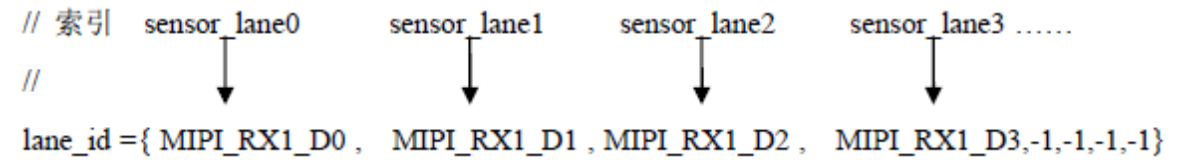
Lane id 的配置对应 mipi_dev_attr_t 中的 short lane_id[MIPI_LANE_NUM]，其中 lane_id 数组的索引号表示的是 sensor 的 Lane ID，lane_id 数组的值表示的是 MIPI Rx 的 LANE ID。

对接 sensor 时，未使用的 lane 将其对应的 lane_id 设置为 0。根据硬件单板与实际sensor 输出通道的对应关系调整lane_id 的配置。下面将举例进行说明。sensor 与 MIPI Rx 的硬件连接关系如表 2-2 所示。

表 2-2 sensor 与 MIPI Rx管脚关系

MIPI Rx 管脚	sensor 管脚
MIPI_RX1_D0	Lane 0
MIPI_RX1_D1	Lane 1
MIPI_RX1_D2	Lane 2
MIPI_RX1_D3	Lane 3

MIPI 的最大Lane 数为8，我们认为SENSOR 的Lane 数目最多8 个，由于sensor 实际只有4 个Lane，只输出数据到MIPI 的4 个Lane，需要将SENSOR 未连接的或者不存在的Lane 的lane_id 配置为-1，所以所以lane_id 配置如图 2-9 所示。



所以最终的 lane_id = {4, 5, 6, 7, -1, -1, -1, -1}