

基于 TDA4 SoC 驱动 CSI2-RX 采集 AWR2188 Radar MMIC 的 Raw ADC 信号



Shen Joe , Xu Liu

Central FAE

摘要

随着 L2 及以上高级辅助驾驶的普及与规范化，高性能的成像级 4D 毫米波雷达传感器和信号处理算法发挥着越来越重要的作用。市场主流 4D 前向毫米波雷达技术正逐步从当前 4T4R 向能提供更远探测距离，更高探测速度，以及兼顾水平和垂直超高角度分辨率与精度的 8T8R 升级换代，但同时也对系统高速数字信号处理的数据量与带宽，算法计算性能和软件设计复杂度提出了更高的要求。目前，TI 发布的 AWR2188 是一款业界领先的、高集成度车规级 RF CMOS 76 to 81Ghz 毫米波射频收发器，可单芯片实现 8T8R 4D 成像雷达前端，其后端可搭载 TI Jacinto TDA4 家族的高性能车规 SoC 处理器以及软件 Processor RTOS SDK 来部署基于内置 Cortex R5 MCU 的无损雷达 ADC raw data 信号接收，基于内置 C7 DSP + MMA 的信号链算法处理，并最终通过高速千兆以太网口输出每帧多达几千个的稠密点云。

本文将着重介绍 AWR2188 通过 CSI2-TX 高速接口输出的 raw ADC 数据格式与排布，并选取 TI TDA4 处理器家族中的一款高性价比入门级 SoC TDA4VEN 的 CSI2-RX 进行配置来正确的采集无损 ADC 数据。该文最后也提供了相应的软件驱动源代码来辅助用户更方便快捷的搭建并调试这个联接毫米波雷达射频前端与高性能信号处理器 SoC 的最重要的关键节点。

内容

1 概述.....	2
2 AWR2188+TDA4VEN 系统设计.....	3
2.1 AWR2188 雷达传感器在汽车产品的应用.....	3
2.2 TDA4VEN 内部计算资源划分和数据流.....	3
2.3 数据格式排布与 Raw ADC data 内存占用.....	4
3 Processor RTOS SDK 中的 CSI2-RX Capture 软件模块驱动配置.....	6
3.1 TDA4VEN 的 CSI2-RX 软件参数，需匹配 AWR2188 SR2.0 的发送格式.....	6
3.2 当 CSI2-RX 接收数据不正常时，在 TDAVEN SoC 端的排查方法.....	7
4 测试脚本工具.....	12
5 参考文献.....	12

插图清单

图 2-1. AWR2188 edge and satellite processing system block diagram.....	3
图 2-2. TDA4VEN software task partitioning and data flow.....	4
图 2-3. AWR2188 SR2.0 ADC data transmitted and stored in non-interleaved mode.....	5

1 概述

TI TDA4VEN SoC 内部集成了符合 MIPI® CSI-2 v1.3 标准和 MIPI® D-PHY 1.2 物理层标准的图像捕获流控 IP，SoC 内部在硬件上设计了 4 组 Camera Serial Interface (CSI2) 接收器（每组配有 4 Lanes D-PHY）。每个数据通道 (Data Lane) 支持高达 2.5Gbps 的传输速率，总物理带宽达到 10Gbps 非常充足。该 CSI2-RX 控制器支持多路虚拟通道 (Virtual Channels)，硬件支持将来自不同输入的帧数据通过不同的 Virtual Channel ID (VC-ID) 加以区分。该控制器不仅负责物理层解包，其内部集成的 DMA 组件能够根据数据流格式，将数据直接搬运存入 SoC 外挂的系统内存 (DDR) 中。CSI2-RX 控制器具备宽泛的数据类型 (Data Type) 识别能力，其主要支持 RAW 格式 (RAW8、RAW10、RAW12、RAW14、RAW16) 和 YUV 格式 (YUV422、YUV420) 等。

TI 的 AWR2188 单芯片 8T8R 毫米波雷达芯片中的 CSI2-TX 必须将 8 个接收通道采集到的海量原始 ADC 数据 (Raw IF Data) 实时、无损地对外发送。AWR2188 的 CSI2-TX 发送控制器和物理层符合 MIPI® CSI-2 v1.2 协议标准，集成符合 MIPI D-PHY v1.1/v1.2 标准的物理层高速波形发生器。AWR2188 SR2.0 版本支持可通过仅 1 个 CSI2-TX 的发送器（配有 4 Lane D-PHY）以最高 1.7Gbps/lane 的速率高效地传输全部 8R 的数据。由于雷达 ADC 采样输出的是实数 (Real) 或复数 (Complex) 信号且和普通视频摄像头的行场数据排布格式差异很大，CSI2-TX 发送控制器在硬件层面上需要将其伪装并打包为标准的 MIPI Data Type 图像数据类型进行流式传输，例如 RAW8/RAW12/RAW16 等格式。

2 AWR2188+TDA4VEN 系统设计

2.1 AWR2188 雷达传感器在汽车产品中的应用

AWR2188 内置 8TX 和 8RX 收发射频和数字前端，并具有丰富的主机接口外设。其中，SPI 用于接收后端处理器发送的初始化固件和实时的 chirp profile 参数配置，CSI2-TX 用于向处理器传输 ADC raw data。当前主流 edge radar 如 4D 毫米波前向雷达和后端处理器通常被设计到同一块 PCB 板卡上，CSI2-RX 到 CSI2-TX 之间的对联无需远距离跨板卡，通过 4 lanes MIPI 高速差分信号直接在板内对接。而卫星雷达往往集成到中央域控制器做雷达信号链算法处理，前端雷达传感器不集成算法处理能力，通过 serdes/TI FPD-Link 单端同轴线缆传输 ADC raw data 数据。

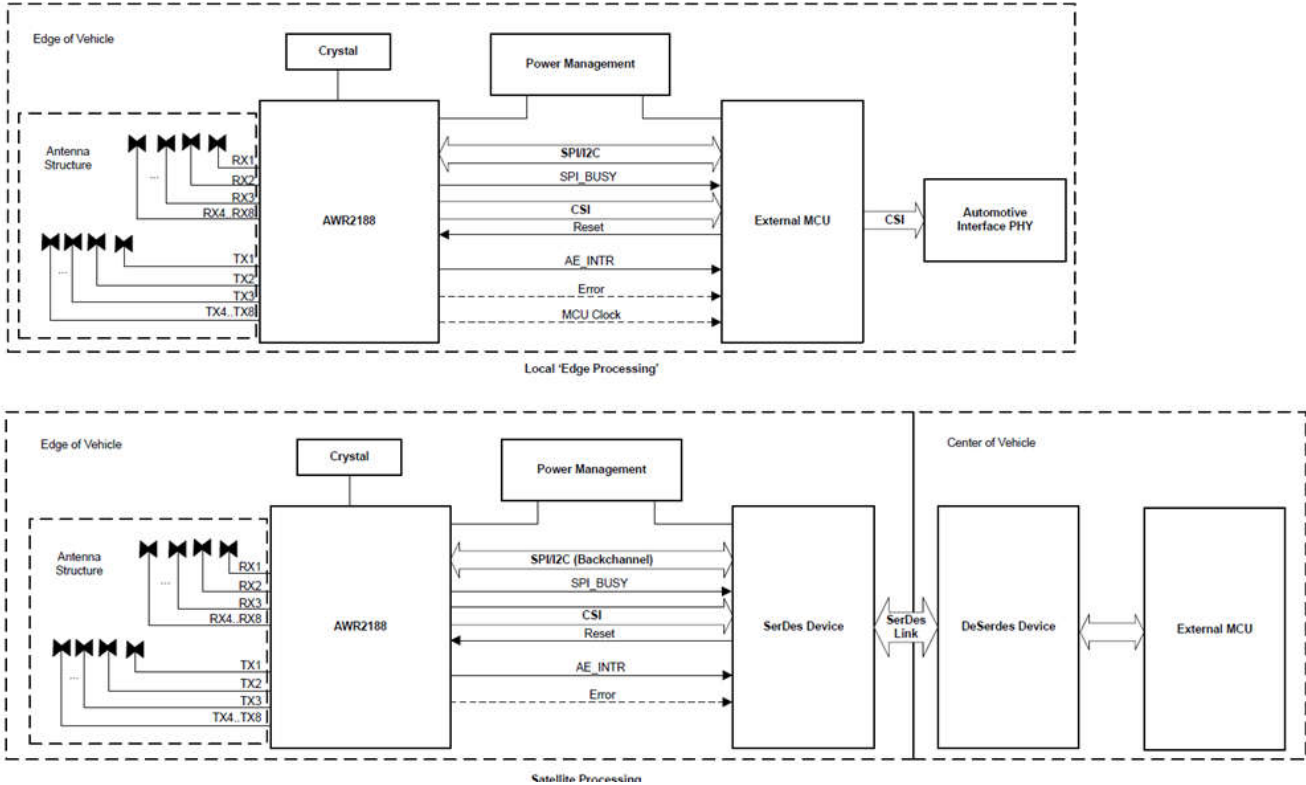


图 2-1. AWR2188 edge and satellite processing system block diagram

2.2 TDA4VEN 内部计算资源划分和数据流

TDA4VEN SoC 内置 3 颗 800MHz ARM cortex R5 和 2 颗 1Ghz 高性能 64+256bits 标矢量混合全浮点 TI C7 DSP，且每颗 dsp 额外挂载了 1Ghz 矩阵乘法加速器 MMA，总共可以支持 80Gflops 32bit 浮点计算和 4Tops/INT8 的神经网络加速或等效的雷达信号处理所必需的 range and doppler FFT 加速。其中 1 颗 Wakeup R5 用于系统 SBL bootloader 启动和设备驱动程序资源管理，1 颗 MCU R5 用于客户专有的 MCU 功能如基于 AutoSar 的软件协议和任务，最后 1 颗 Main R5 用于驱动 FreeRTOS 上的 CSI2-RX 软件 driver，配置 Raw ADC 数据接收格式并初始化和响应 TI 的高速 UDMA 通道将接收到的 Raw ADC 数据搬运至外置 LPDDR4 内存，并通过 IPC 通知 C7 DSP 进行算法计算。C7 DSP 无需 frame 拷贝，通过从 IPC 获取到的共享内存地址，并配置额外的 UDMA 通道将 DDR 中的数据分批搬运到 L2 SRAM 片上内存进行高效运算并在后台同步并发下一次数据搬入和本次结果的搬出，从而大幅度优化并解决了特别影响整体系统性能的内存等待时间问题。最后 DSP 将雷达信号链算法处理后生成的 4D 点云数据放入共享内存并通知 R5 向其驱动的以太网接口进行对外发送。

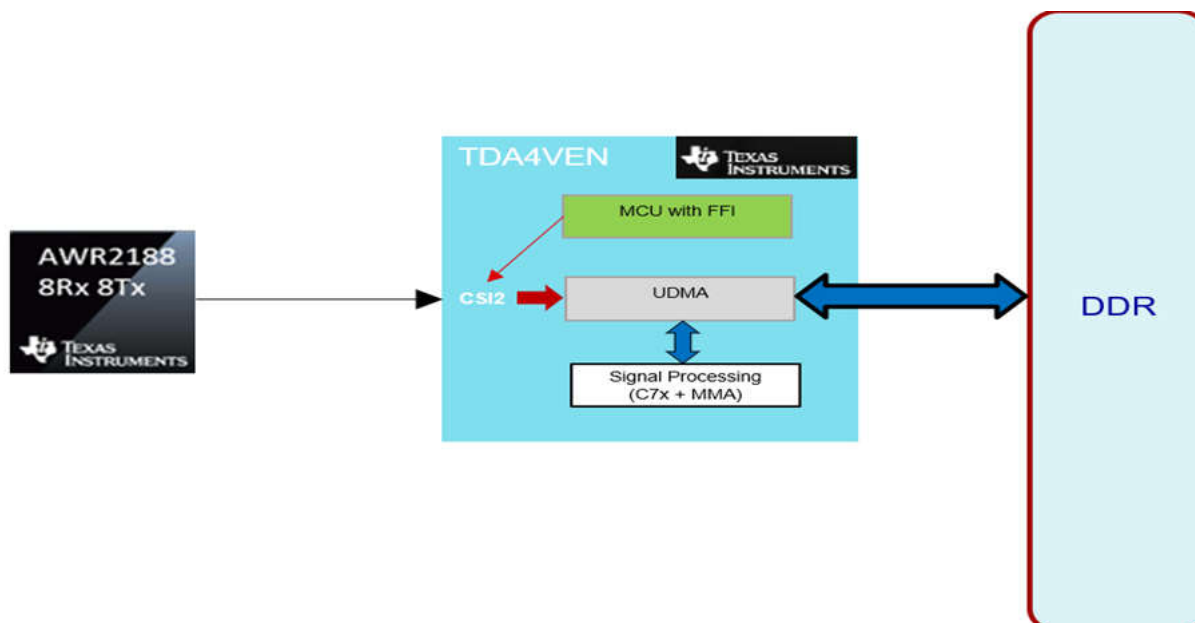


图 2-2. TDA4VEN software task partitioning and data flow

2.3 数据格式排布与 Raw ADC data 内存占用

AWR2188 SR2.0 支持单个 CSI2-TX 发送，配置其 RX0-RX7 全部输出到 Port 0，配置 data type 为 RAW8 的格式，以 850MHz 的 lane speed 进行发送。为了提高速度分辨率，AWR2188 每一个 RX 天线的发送 profile 设置为 1024 个 Chirp。为了提高距离分辨率，每一个 Chirp 设置为 1024 个 ADC 的 16bit 实数 sample 数据（每一个 ADC 的 sample 是 2 个字节）。这样，单帧总数据量为 $2 \text{ Bytes} * 1024 \text{ Samples} * 1024 \text{ chirps} * 8 \text{ Rx} = 16\text{MB}$ 。

接收端 TDA4VEN 的 CSI2-RX 仅需使能 1 个接收器 instance 0，按照 RAW8 的格式接收，只使能 Virtual Channel 0，以一个较大的 width 包含全部 8 个 RX 的 1024 个 16 bits Samples（先排布完 sample，再排布完 RX），Chirps 的数量则对应到 Height 这个维度。

DRAM – AWR2188 ADC – RX*Chirp*ADC = 8*1024*1024*2 = **16MB**

	2-bytes	2-bytes	2-bytes	2-bytes	2-bytes
0x800	Chirp0 RX0 - ADC0	Chirp0 RX0 - ADC1		Chirp0 RX0 - ADC1022	Chirp0 RX0 - ADC1023
0x1000	Chirp0 RX1 - ADC0	Chirp0 RX1 - ADC1		Chirp0 RX1 - ADC1022	Chirp0 RX1 - ADC1023
	⋮	⋮		⋮	⋮
0x3800	Chirp0 RX7 - ADC0	Chirp0 RX7 - ADC1		Chirp0 RX7 - ADC1022	Chirp0 RX7 - ADC1023
0x4000	Chirp1 RX0 - ADC0	Chirp1 RX0 - ADC1		Chirp1 RX0 - ADC1022	Chirp1 RX0 - ADC1023
	⋮	⋮		⋮	⋮
0x8000	Chirp2 RX0 - ADC0	Chirp2 RX0 - ADC1		Chirp2 RX0 - ADC1022	Chirp2 RX0 - ADC1023
	⋮	⋮		⋮	⋮
0x1FC00 0	Chirp1023 RX0 - ADC0	Chirp1023 RX0 - ADC1		Chirp1023 RX0 - ADC1022	Chirp1023 RX0 - ADC1023
	⋮	⋮		⋮	⋮
0x1FF800	Chirp1023 RX7 - ADC0	Chirp1023 RX7 - ADC1		Chirp1023 RX7 - ADC1022	Chirp1023 RX7 - ADC1023

图 2-3. AWR2188 SR2.0 ADC data transmitted and stored in non-interleaved mode

3 Processor RTOS SDK 中的 CSI2-RX Capture 软件模块驱动配置

3.1 TDA4VEN 的 CSI2-RX 软件参数，需匹配 AWR2188 SR2.0 的发送格式

```
gCsiCfgHandle[i]->chFrmCnt = 0U;
gCsiCfgHandle[i]->createPrms.chCfg.chId = 0U;
gCsiCfgHandle[i]->createPrms.chCfg.chType = 0U;
gCsiCfgHandle[i]->createPrms.chCfg.vcNum = 0U;
gCsiCfgHandle[i]->createPrms.chCfg.inCsiDataType = FVID2_CSI2_DF_RAW8;
gCsiCfgHandle[i]->createPrms.chCfg.outFmt.width = 1024(adc)*2(Bytes)*8(RX);
gCsiCfgHandle[i]->createPrms.chCfg.outFmt.height = 1024(Chirp);
gCsiCfgHandle[i]->createPrms.chCfg.outFmt.pitch[0U] = 1024(adc)*2(Bytes)*8(RX);
gCsiCfgHandle[i]->createPrms.chCfg.outFmt.dataFormat = FVID2_DF_RAW08;
gCsiCfgHandle[i]->createPrms.chCfg.outFmt.ccsFormat = FVID2_CCSF_BITS8_PACKED;
gCsiCfgHandle[i]->createPrms.instCfg.enableCsiv2p0Support = (uint32_t)TRUE;
gCsiCfgHandle[i]->createPrms.instCfg.numDataLanes = 4U;
gCsiCfgHandle[i]->createPrms.instCfg.enableErrbypass = (uint32_t)FALSE;
gCsiCfgHandle[i]->createPrms.instCfg.enableStrm[CSIRX_CAPT_STREAM_ID] = 1U;
dphyCfg->leftLaneBandSpeed = CSIRX_LANE_BAND_SPEED_1500_TO_1750_MBPS;
dphyCfg->rightLaneBandSpeed = CSIRX_LANE_BAND_SPEED_1500_TO_1750_MBPS;
chType: 表示 CSIRX 内置的 DMA 把数据搬运到 DDR 中;
vcNum: 表示虚拟通道号;
inCsiDataType: 表示传输格式;
width: 表示传输的宽度;
Height: 表示传输的高度;
Pitch: 表示传输完一行数据后开始下一行需要跳过多少数据;
DataFormat: 表示输出的格式;
ccsFormat: 表示输出的数据在内存里面的排列;
numDataLanes: 表示按照硬件连接使用多少条 data lane;
enableErrbypass: 表示是否要使用错误检查;
leftLaneBandSpeed/rightLaneBandSpeed: 表示按照发送端的速度 bps;
```

小心

MIPI-CSI2 是双沿采样差分信号，data lane 上的数据传输速度是 clock lane 上频率的 2 倍。因此 AWR2188 SR2.0 CSI2-TX Port0 配置的 850Mhz，也就是对应 1700Mbps 的数据发送速度。

小心

需要特别指出的是 TDA4VEN 的 CSI2-RX 控制器必须先准备 ready 后，对端发送端才能开始发送数据。时序上必须要严格满足，否则 CSI2-RX 可能无法稳定地接收信号。

3.2 当 CSI2-RX 接收数据不正常时，在 TDAVEN SoC 端的排查方法

步骤 A：在 SDK source\drivers\csirx\v1\src\csirx_drvUdma.c 中的函数 CsirxDrv_udmaCQEventCb 是当 CSIRX 接收到数据后的 DMA 中断 callback 函数，首先可以通过 CCS 设置断点在如下函数中检查是否接收到数据。

```

429:
430: void CsirxDrv_udmaCQEventCb(Udma_EventHandle eventHandle,
431:                             uint32_t eventType,
432:                             void *appData)
433: {
434:     int32_t retVal = UDMA_SOK, tempVal = UDMA_SOK;
435:     CsirxDrv_ChObj *chObj;
436:     CsirxDrv_QueueObj *qObj;
437:     CsirxDrv_BufManObj *bmObj;
438:     uint64_t pDesc = 0;
439:     uint32_t pTrResp;
440:     uint32_t pTrRespVal, pTrRespStsTypeVal;
441:     uint32_t curQCnt, appCb = 0U, cookie, chldx;
442:     uint64_t timeStamp;
443:     CSL_UdmapTR1 *pTr;
444:
445:     if(eventHandle == NULL)
446:     {
447:         /*To remove unused variable warning*/
448:     }
449:
450:     chObj = (CsirxDrv_ChObj *) appData;
451:     GT_assert(CsirxTrace, (chObj != NULL));
452:
453:     /* Take out completed TRPD from UDMA CQ Ring */
454:     if(eventType == UDMA_EVENT_TYPE_DMA_COMPLETION)
455:     {
456:         /* Disable HW interrupts here */
457:         cookie = HwiP_disable();
458:         bmObj = &chObj->bufManObj;
459:         /* Response received in completion queue, empty up full CQ Ring */
460:         while (retVal == UDMA_SOK)

```

步骤 B：当软件中断未被调用即没有接收到数据时，检查硬件 CSI2-RX IP 是否有接收到 MIPI 的协议包，或是否有接收到 IP 上报的错误。

14.8.6.1.2.36.1 CSIRX_INFO_IRQS Register (Offset = 20h) [reset = 0h]

Information type Interrupt status (non-error conditions)

Return to [Summary Table](#)

Table 14-23120. Instance Table

Instance Name	Physical Address
CSI_RX_IF0	3010 1020h

Figure 14-11440. CSIRX_INFO_IRQS Name Register

31	30	29	28	27	26	25	24
RESERVED							
NONE							
0h							
23	22	21	20	19	18	17	16
RESERVED							
NONE							
0h							
15	14	13	12	11	10	9	8
RESERVED	STREAM3_AB ORT_IRQ	STREAM3_ST OP_IRQ	STREAM2_AB ORT_IRQ	STREAM2_ST OP_IRQ	STREAM1_AB ORT_IRQ	STREAM1_ST OP_IRQ	STREAM0_AB ORT_IRQ
NONE	R/W1TC	R/W1TC	R/W1TC	R/W1TC	R/W1TC	R/W1TC	R/W1TC
0h	0h	0h	0h	0h	0h	0h	0h
7	6	5	4	3	2	1	0
STREAM0_ST OP_IRQ	SP_GENERIC_ RCVD_IRQ	DESKEW_ENT RY_IRQ	ECC_SPARES_ NONZERO_IR Q	WAKEUP_IRQ	SLEEP_IRQ	LP_RCVD_IRQ	SP_RCVD_IRQ
R/W1TC	R/W1TC	R/W1TC	R/W1TC	R/W1TC	R/W1TC	R/W1TC	R/W1TC
0h	0h	0h	0h	0h	0h	0h	0h

Table 14-23121. CSIRX_INFO_IRQS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	DESKEW_ENTRY_IRQ	R/W1TC	0h	Either clock or any datalane has entered deskew
4	ECC_SPARES_NONZER O_IRQ	R/W1TC	0h	Bits 7:6 of the ECC byte are non-zero. Indicates non compliance with the MIPI specification although the core will continue to operate as normal.
3	WAKEUP_IRQ	R/W1TC	0h	Wake-up interrupt.
2	SLEEP_IRQ	R/W1TC	0h	Sleep interrupt.
1	LP_RCVD_IRQ	R/W1TC	0h	Long Packet received by the protocol module
0	SP_RCVD_IRQ	R/W1TC	0h	Short Packet received by the protocol module

注 Bit0/1：表示 MIPI 的 IP 协议层接收到了符合协议的数据。

注 Bit5：当 CSIRX 的 data lane 上的速度大于 1.5Gbps，TDA4VEN 要求发送端发送数据前先做 DESKEW 的操作。如果这个 Bit 没有设置，应该检查发送端的 AWR2188 CSI2-TX 的发送配置是否包含 DESKEW。

步骤 C : 查看状态寄存器是否出现了错误。常见的有信号传输过程中产生的 ECC，CRC 的错误，需要检查发送端的 CSI2-TX 的配置是否满足标准的时序。Fifo 的溢出错误需检查发送端和接收端的 MIPI 的速度是否匹配，DATA ID 不匹配的错误需检查发送端和接收端设置的格式是否一致。TRUNCATED_PACKET 表示发送端和接收端设置的 size 不一致，需检查长宽设置是否一致。

14.8.6.1.2.38.1 CSIRX_ERROR_IRQS Register (Offset = 28h) [reset = 0h]

Datapath error interrupt status. Provides information about data path errors. The host processor can read the interrupt status register to identify the root cause of the event, typically after that the csi2rx_err_irq interrupt line is raised

Return to [Summary Table](#)

Table 14-23124. Instance Table

Instance Name	Physical Address
CSI_RX_IF0	3010 1028h

Figure 14-11442. CSIRX_ERROR_IRQS Name Register

31	30	29	28	27	26	25	24
RESERVED							
NONE							
0h							
23	22	21	20	19	18	17	16
RESERVED				STREAM3_FIFO_OVERFLOW_IRQ	STREAM2_FIFO_OVERFLOW_IRQ	STREAM1_FIFO_OVERFLOW_IRQ	STREAM0_FIFO_OVERFLOW_IRQ
NONE				R/W1TC	R/W1TC	R/W1TC	R/W1TC
0h				0h	0h	0h	0h
15	14	13	12	11	10	9	8
RESERVED			FRONT_TRUNC_HDR_IRQ	PROT_TRUNCATED_PACKET_IRQ	FRONT_LP_NO_PAYLOAD_IRQ	SP_INVALID_R_CVD_IRQ	INVALID_ACCESS_IRQ
NONE			R/W1TC	R/W1TC	R/W1TC	R/W1TC	R/W1TC
0h			0h	0h	0h	0h	0h
7	6	5	4	3	2	1	0
DATA_ID_IRQ	HEADER_CORRECTED_ECC_IRQ	HEADER_ECC_IRQ	PAYLOAD_CRC_IRQ	RESERVED			FRONT_FIFO_OVERFLOW_IRQ
R/W1TC	R/W1TC	R/W1TC	R/W1TC	NONE			R/W1TC
0h	0h	0h	0h	0h			0h

步骤 D : 检查 TDA4VEN 的 CSIRX 的 DMA 设置的 DATA ID 和 VC 通道号是否和发送端一致。

14.8.6.1.2.109.1 CSI_RX_IF_CNTX_CNTL_DMACNTX_J Register (Offset = 0h) [reset = C000000h]

DMA Channel Context. Configuration for each of 32 possible channel contexts. Illegal to program 2 chanCntx with same extraction values.

Return to [Summary Table](#)

Table 14-23266. Instance Table

Instance Name	Physical Address
CSI_RX_IF0	3010 2000h + formula

Figure 14-11513. CSI_RX_IF_CNTX_CNTL_DMACNTX_J Name Register

31	30	29	28	27	26	25	24
EN_CFG	RESERVED	RSV0	RSV1	YUV422_MODE_CFG	RSV2	DUAL_PCK_CFG	
R/W	NONE	R/W	R/W	R/W	R/W	R/W	
0h	0h	0h	0h	3h	0h	0h	
23	22	21	20	19	18	17	16
RESERVED	SIZE_CFG	RESERVED	PCK12_CFG	RESERVED			
NONE	R/W	NONE	R/W	NONE			
0h	0h	0h	0h	0h			
15	14	13	12	11	10	9	8
RESERVED						VIRTCH_CFG	
NONE						R/W	
0h						0h	
7	6	5	4	3	2	1	0
VIRTCH_CFG	DATTYP_CFG						
R/W	R/W						
0h	0h						

9:6	VIRTCH_CFG	R/W	0h	CSI virtual channel index. Supplied by MIPI CSI protocol to DPHY. For CSilver1.3 program 2MSb==0 Reset Source: main_mod_g_srst_n
5:0	DATTYP_CFG	R/W	0h	CSI data type index. Supplied by MIPI CSI protocol to DPHY Reset Source: main_mod_g_srst_n

步骤 E：左右的速度必须是一样的，例如若在软件的 SDK 中对应参数配置为 `#define CSIRX_LANE_BAND_SPEED_1500_TO_1750_MBPS ((uint32_t) 0x13U)`，那么这个寄存器回读的值应是 `0x273 (0010 0111 0011)`。

14.8.6.2.2.5.1 K3_DPHY_RX_PCS_TX_DIG_TBIT0 Register (Offset = B00h) [reset = 0h]

PHY_BAND_CONTROL

[Return to Summary Table](#)

Table 14-23282. Instance Table

Instance Name	Physical Address
DPHY_RX0	3011 0B00h

Figure 14-11520. K3_DPHY_RX_PCS_TX_DIG_TBIT0 Name Register

31	30	29	28	27	26	25	24
UNUSED							
R							
0h							
23	22	21	20	19	18	17	16
UNUSED							
R							
0h							
15	14	13	12	11	10	9	8
UNUSED						BAND_CTL_REG_R	
R						R/W	
0h						0h	
7	6	5	4	3	2	1	0
BAND_CTL_REG_R				BAND_CTL_REG_L			
R/W				R/W			
0h				0h			

Table 14-23283. K3_DPHY_RX_PCS_TX_DIG_TBIT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31:10	UNUSED	R	0h	RESERVED
9:5	BAND_CTL_REG_R	R/W	0h	Data Rate [80_100] MHz
4:0	BAND_CTL_REG_L	R/W	0h	Data Rate [80_100] MHz

4 测试脚本工具

在实际的项目中，可能还需要检查更多的寄存器来确认 TDA4VEN SoC 端的配置和状态是否正常，附件是一个 gel 的脚本文件，这里面包含了 TDA4VEN 的 CSI2-RX 和 DHPY 的所有的寄存器，可以用 CCS 连接上后直接 dump 出来完整的寄存器的列表后和 TI EVM 上正常能接收数据的寄存器做比较，检查是否有不一致的配置和状态。

5 参考文献

1. [J722S TDA4VEN TDA4AEN AM67 Processor Silicon Revision 1.0 Technical Reference Manual \(Rev. C\)](#)
2. [AWR2188 Single Chip 8x8 Cascadable 76-to-81 GHz Transceiver in LOP package datasheet \(Rev. A\)](#)
3. [Software Development Kit for TDA4VEN and TDA4AEN Jacinto processors-Download/PROCESSOR-SDK-RTOS-J722S](#)

重要通知和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、与某特定用途的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他安全、安保法规或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。对于因您对这些资源的使用而对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，您将全额赔偿，TI 对此概不负责。

TI 提供的产品受 [TI 销售条款](#)、[TI 通用质量指南](#) 或 [ti.com](#) 上其他适用条款或 TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。除非德州仪器 (TI) 明确将某产品指定为定制产品或客户特定产品，否则其产品均为按确定价格收入目录的标准通用器件。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

版权所有 © 2026，德州仪器 (TI) 公司

最后更新日期：2025 年 10 月