

Hello-FPGA CoaXPress 2.0 FPGA HOST IP Core User Manual

Hello-FPGA
INFO@HELLO-FPGA.COM

目录

Hello-FPGA CoaXPress 2.0 Host FPGA IP Core.....	3
1 CoaXPress 简介.....	3
2 CoaXPress 2.0 FPGA IP 概述.....	4
2.1 IP 特性.....	4
2.2 器件支持与资源占用.....	5
3 CoaXPress 2.0 FPGA IP 功能说明.....	6
3.1 IP 结构.....	6
3.2 模块描述.....	6
4 CoaXPress 2.0 FPGA IP 参数与接口说明.....	7
4.1 配置参数.....	8
4.2 接口说明.....	9
4.2.1 时钟、复位、启动采集接口.....	9
4.2.2 链路状态相关信号.....	10
4.2.3 低速 uplink PHY 相关信号.....	10
4.2.4 高速外部收发器 phy 接口.....	10
4.2.5 解码统计接口.....	10
4.2.6 Control 控制接口.....	11
4.2.7 Trigger 触发接口.....	12
4.2.8 HeartBeat 心跳接口.....	14
4.2.9 EVENT 事件接口.....	14
4.2.10 Video DMA Stream 接口.....	15
4.2.11 Video Format 接口.....	17
4.2.12 授权接口.....	17
5 CoaXPress 2.0 FPGA IP 地址空间及寄存器描述.....	18
5.1 地址空间.....	18
5.2 寄存器描述.....	19
5.2.1 Host 控制寄存器描述.....	19
5.2.2 解码器 Decoder 寄存器描述.....	25
5.3 Device 寄存器访问模式.....	28
5.3.1 INDIRECT 间接访问.....	28
5.3.2 DIRECT 直接访问.....	29
6 IP 软件配置.....	29
6.1 基本的 Camera 检测和采集启动.....	29
6.2 LINK 链路到 Camera（仲裁器 Arbiter）的链接.....	30
6.3 Camera（仲裁器 Arbiter）与解码器的链接.....	32
Annex A: 参考资料.....	35

图 1-1 CXP 链接结构.....	4
图 4-1 CXP Host FPGA IP(Xilinx IP Integrator).....	8
图 4-2 可配置参数	8
图 4-3 control bus INDIRECT 模式读写示例	12
图 4-4 control bus DIRECT 模式 burst 读写示例(burst 范围 1-256).....	12
图 4-5 uplink trigger timing, Host to Device.....	13
图 4-6 downlink trigger timing, Device to Host	14
图 4-7 Heart Beat interface timing	14
图 4-8 EVENT Timing.....	15
图 4-9 Video DMA timing(以 14bit 像素为例, 图示共 58 个 pixels)	16
图 6-1 2 台相机与 4 个 LINK 的连接设置(1).....	31
图 6-2 2 台相机与 4 个 LINK 的连接设置(2).....	32
图 6-3 2 台相机与 2 个解码器的连接.....	33
图 6-4 3 台相机与 3 个 Decoder 的连接	34
表 2-1 CoaXPress Host IP 支持器件列表与资源占用.....	5
表 4-1 参数说明	8
表 4-2 时钟、复位、启动接口	9
表 4-3 链路状态	10
表 4-4 低速 uplink 物理信号	10
表 4-5 高速 PHY 接口	10
表 4-6 统计接口	10
表 4-7 高速 PHY 接口	11
表 4-8 trigger 接口	13
表 4-9 Heart Beat 接口	14
表 4-10 Event 接口	15
表 4-11 Video Stream DMA 接口.....	15
表 4-12 Video Header 接口.....	17
表 4-13 授权接口	17
表 5-1 DIRECT 模式寄存器地址范围	18
表 5-2 DIRECT 模式寄存器地址范围	19
表 5-3 Host 控制寄存器	19
表 5-4 LINK_STATUS 寄存器描述.....	24
表 5-5 INDIRECT_COMMAND 寄存器描述.....	25
表 5-6 Decoder 寄存器描述	25

Hello-FPGA CoaXPress 2.0 Host FPGA IP Core

1 COAXPRESS 简介

CoaXPress (简称 CXP)是指一种采用同轴线缆进行互联的相机数据传输标准，主要用于替代之前的 Camera Link 协议，常见于科学相机、工业相机、医学图像、航空防务等场景。Camera Link 由于线缆的形式、传输速度等原因，已经不再适用于不断增长的数据带宽需求。

CXP 是一个非对称的高速点对点串行传输协议，主要用于传输视频和静态图像，线缆多使用单条或多条同轴电缆。目前 2.0 标准的最高速度为单条 lane 12.5Gbps，单条 lane 上除了传输图像数据之外，还可以传输低速控制信号(41.6Mbps)、也可以利用该线缆对相机进行供电，即“Power-over-Coax”，单条线缆最长可达 100m。

- 单条同轴线缆最高 12.5Gbps 的线速率，可以使用单条或者多条线缆，比如 4 条线缆可以最高提供 50Gbps 的数据速率
- 较长的线缆长度，比如 3.125 Gbps 速率下线长可以到 100m，12.5Gbps 速率下可以到 35m
- 数据传输具备实时和低延迟的特性，且延迟是固定的
- 精准实时的触发特性，支持通过同轴线缆发送触发，无需额外的通讯线缆
- 可以获得灵活可靠的性能，通过使用标准的同轴线缆，比如 RG59 和 RG6 规格，推荐使用 RG6
- 易于集成的特性，图像、控制通讯、电源可以使用同一条线缆，且线缆价格便宜

CXP 是一种点对点可扩展接口，设备和 HOST 之间的物理介质为 75Ω 同轴电缆。每一个 CXP 接口都包含 1 个 MASTER 连接和若干可选的扩展 SLAVE 连接，每一个连接都需要 1 条同轴线缆，通常设备会对这些连接进行编号，MASTER 固定为 0，SLAVE 扩展接口依次递增。

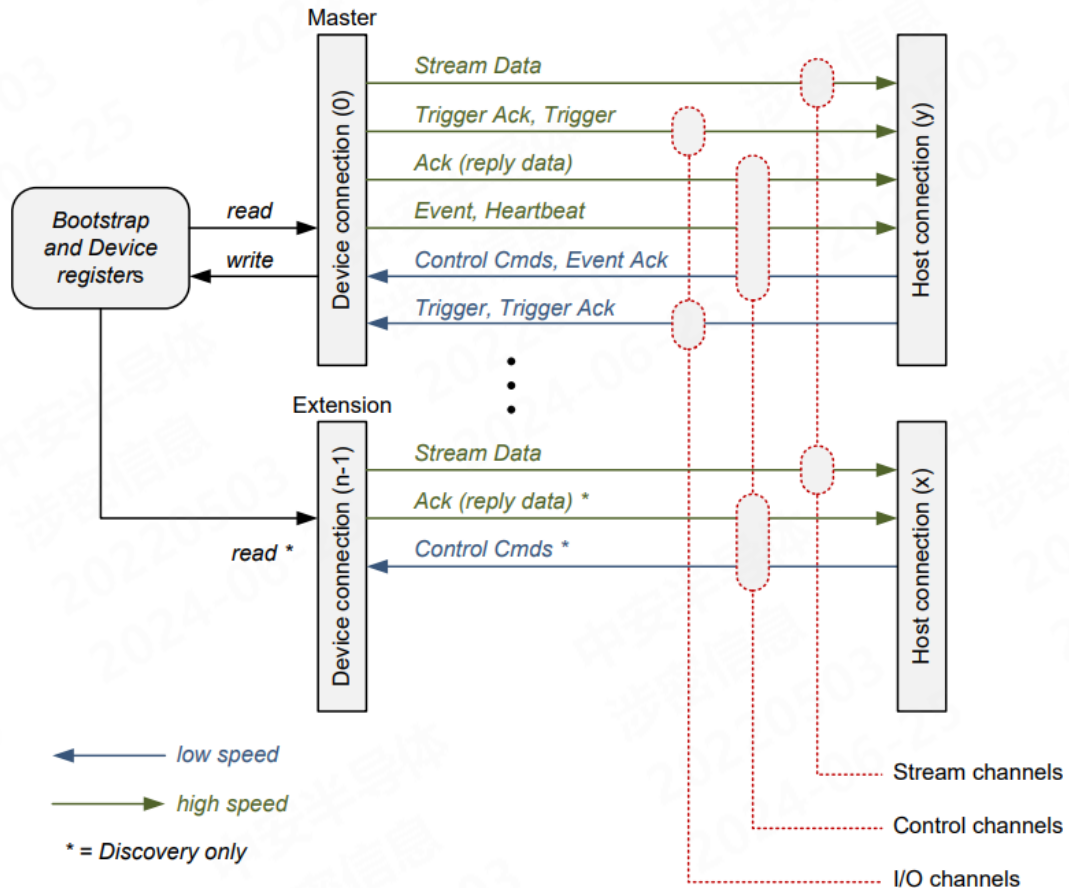


图 1-1 CXP 链接结构

每一个 connection 都包含如下功能

- 高速串行数据(通常为 Device 到 Host downconnection), 最高 12.5 Gbps.
- 低速串行数据(通常为 Host 到 Device upconnection), 最高 41.6 Mbps .
- 供电功能 Power (Host 到 Device), 最高 13W.

2 CoaXPRESS 2.0 FPGA IP 概述

2.1 IP 特性

Hello-FPGA 的 CoaXPRESS 2.0 FPGA IP Core 特点如下:

- 兼容 CoaXPRESS V2.1 标准(JIJA NIF-001-2021) , 向后兼容 CoaXPRESS V2.0 标准(JIJA NIF-001-2019) , V1.1 标准(JIJA NIF-001-2013), V1.0 标准(JIJA NIF-001-2010).
- 支持最高 CXP-12 downlink 高速链接, 41.6 Mbps uplink 低速链接, 支持 CoaXPRESS 定义的所有速率.
- 支持 link 数量 1, 2, 4, 8 配置, 可以满足更高吞吐率的数据传输,
- 支持最多 8 路并行 stream 视频数据流, 比如 1 个 HOST 连接多路相机或者连接 1 个相机的多路图像传感器数据.
- 使用 FPGA 内置的高速收发器作为 PHY 芯片, 使用简单.

- 用户接口的控制总线采用简单的 Avalon 通讯接口，支持 burst 模式的读写，同时提供可选的 AXI-4 接口转换模块；
- 用户接口的数据总线采用简单的 Video 总线接口。
- 专用 GPIO 和触发通讯接口，允许用户发送触发信号到 Device 设备。

2.2 器件支持与资源占用

表 2-1 是当前验证过的器件列表及其资源占用情况，其它器件支持待后续更新。

表 2-1 CoaXPress Host IP 支持器件列表与资源占用

器件类型	Link Num	Stream Num	LUTs	FFs	BRAM	DMA Fmax
Kintex 7	1	1	7.5k	8k	11	250Mhz
Kintex 7	4	1	12k	14k	40	250Mhz
ZYNQ Ultra Scale +	1	1	8k	12k	11	250Mhz
ZYNQ Ultra Scale +	4	1	18k	24k	40	250Mhz

目前测试支持的开发板(应客户需求会不断更新):

KC705, KCU105, ZCU102, ALINX Z19

3 COAXPRESS 2.0 FPGA IP 功能说明

3.1 IP 结构

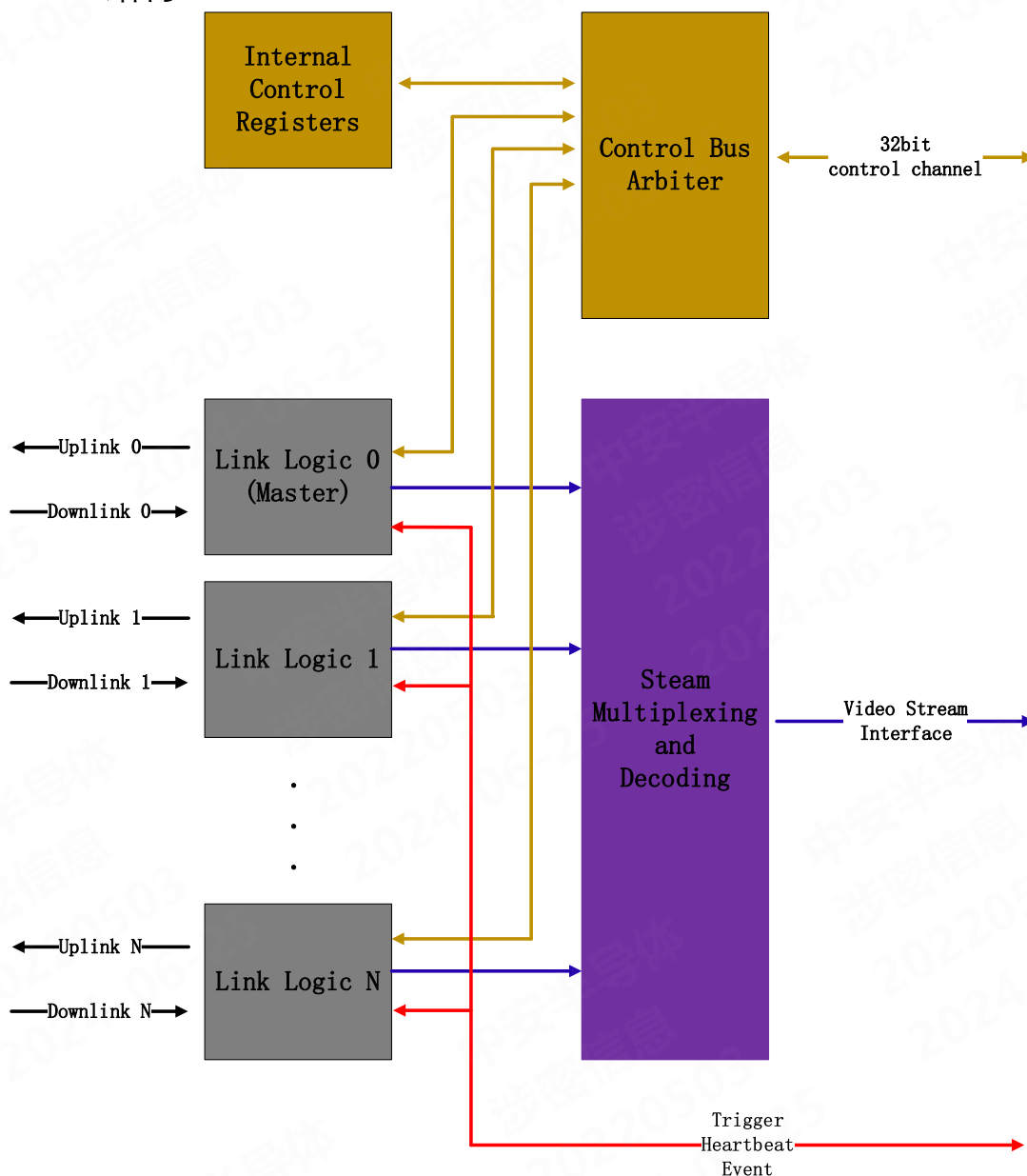


图 3 IP 结构图

其中，紫色绘制的组件代表图像流相关功能块，棕色组件代表控制通道相关功能块，灰色组件代表链路层相关功能块。红色路径代表触发器、心跳和事件机制。

3.2 模块描述

组成IP核的基本组件是：

链路逻辑模块：用于实现高速downlink传输链路、低速uplink传输链路、控制信息传输链接以及测试链路。它对每一个CoaXPRESS链接都是通用的，其中N是

CoaXPress链接的数量，0是主站。链路逻辑采用8B/10B编码，实现对上行和下行数据的组包和解包，同时能够对数据包进行CRC检查并标记，该块还管理着触发、心跳和事件机制。

流复用和解码模块：被设计用来解码和复用从每个链接逻辑到视频接口的输入流数据包。图像包头被剥离，输出数据是原始的视频流。为了在不增加接口时钟速度的情况下增加视频流数据的吞吐量，该接口支持多达16个字节的并行传输。

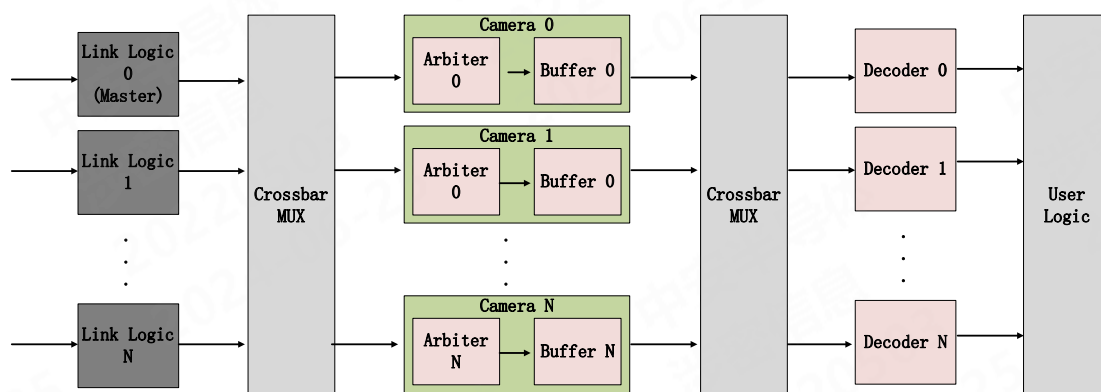


图4 流复用和译码模块结构图

控制总线仲裁器 (Arbiter)：对控制通道事务进行仲裁，按照内部控制寄存器的配置进行。

内部控制寄存器：IP 操作寄存器配置。对外接口为 local control bus 或者 AXI-4 总线，支持 burst 数据传输，寄存器用于 IP 动态配置、状态监测等；

4 COAXPRESS 2.0 FPGA IP 参数与接口说明

以 Xilinx Vivado IP Integrator(IPI)的方式对接口进行说明。

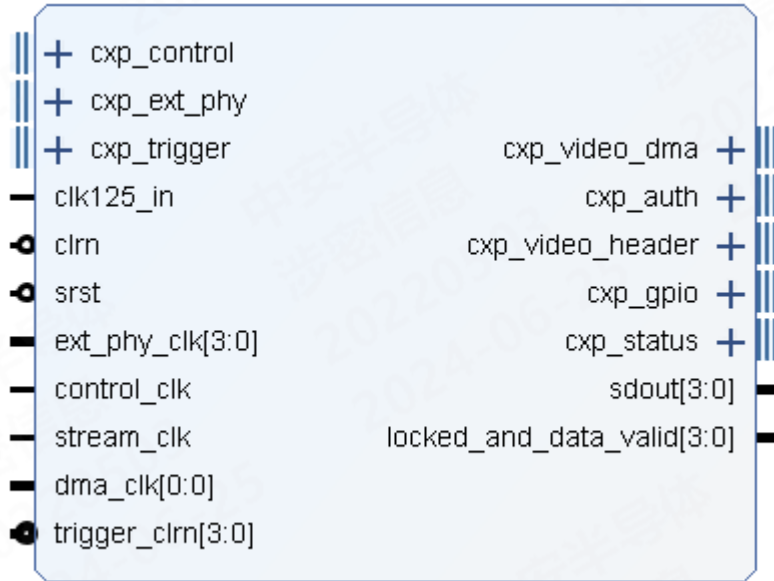


图 4-1 CXP Host FPGA IP(Xilinx IP Integrator)

4.1 配置参数

☐ Show disabled ports

Component Name: **hello_fpga_cyp_host_0**

Device	Kintex 7
CXP2.X Logic En	0
Cam Num	1
Links	4
Streams	1
Stream Words	4
Stream Buffer Bytes	8192
Multi Stream En	0
External Phy	1
Ref Clk Freq	125.0 MHz
Indirect Regs	1
Bigendian En	0
<input type="checkbox"/> Event	
<input type="checkbox"/> Heartbeat	

Diagram of CXP Host FPGA IP ports (repeated from Figure 4-1):

- Inputs: + cyp_control, + cyp_ext_phy, + cyp_trigger, clk125_in, clrn, srst, ext_phy_clk[3:0], control_clk, stream_clk, dma_clk[0:0], trigger_clrn[3:0].
- Outputs: cyp_video_dma, cyp_auth, cyp_video_header, cyp_gpio, cyp_status, sdout[3:0], locked_and_data_valid[3:0].

图 4-2 可配置参数

表 4-1 参数说明(网表文件不可配置)

参数	配置值	默认值	描述说明
Device	器件型号	UltraScale	Kintex UltraScale , Kintex 7 ,Artix 7

LINKS	1 - 8 ⁽¹⁾	4	CoaXPress 链接的总数量。
CAM_NUM	1 - 8	1	可连接到 IP 的 Camera 的最大数量。
STREAM_WORDS	1 - 8	4	通过视频流接口并行传输的字数；可以修改该参数以增加 IP 吞吐量，而不增加流接口频率。
STREAMS	1 - 8	1	IP 中使用的最大流数。
DEVICE	Kintex 7 Kintex7 Ultra Scale Kintex7 Ultra Scale+ ZYNQ 7000 Ultra Scale+ ZYNQ Ultra Scale+	根据客户购买的 FPGA 类型	指定目标器件。
MULTI_STREAM_EN	0 - 不使能 1 - 使能	0	启用多数据流（muti-stream/每台 Camera）逻辑。
STREAM_BUFFER_BYTES	1,024 - 16,384 (1)	4096	流数据缓冲字节数。该参数的值必须至少是最大数据包大小的两倍。
INDIRECT_REGS	0 - 不使能 1 - 使能	0	使能对 Camera 初始化寄存器的间接访问模式。
BIGENDIAN_EN	1 - 大端 0 - 小端	1	设置 DMA 数据的大小端模式。
EXT_PHY	1- 使能外部PHY 2- 内部PHY	0	高速收发器资源，除 K7 系列外，其它系列暂不支持内部 PHY
EVENT	0 - 不使能 1 - 使能	Event 接口	是否使能 Event 接口
HeartBeat	0 - 不使能 1 - 使能	HeartBeat 接口	是否使能心跳接口

4.2 接口说明

4.2.1 时钟、复位、启动采集接口

表 4-2 时钟、复位、启动接口

信号名称	输入/输出	描述说明
clrn	输入	低有效异步复位输入，内部同步。
clk125_in	输入	125Mhz 输入时钟。必须由 PLL 或晶振提供。
stream_clk	输入	流缓冲和同步时钟。要求时钟频率等于链接速率除以 40（ext_phy_clk）。

phy_clk	输出	从高速 PHY 层输出的时钟。该时钟的频率等于链接速率除以 40 (ext_phy_clk)。
acq_trigger	输入	trigger 输入。必须与 stream_clk 同步，单时钟脉冲。其作为定时帧接收的内部触发器。如果不使用，该信号必须被连接到逻辑高电平。

4.2.2 链路状态相关信号

表 4-3 链路状态

信号名称	输入/输出	描述说明
locked_and_data_valid	输出	FSL downlink 被锁定和数据被接收的指示，该信号对每个 LINK 有效，即每个 Link 均有一个对应输出，可以用于检查 link 的物理状态。
fsl_locked_out	输出	FSL downlink 被锁定的指示，该信号在每个 LINK 上有效。
fsl_synced_out	输出	FSL downlink 被同步的指示，该信号在每个 LINK 上有效。

4.2.3 低速 uplink PHY 相关信号

低速 uplink 物理信号

表 4-4 低速 uplink 物理信号

信号名称	输入/输出	描述说明
sdout	输出	串行输出数据。sdout.信号应连接到通用的 FPGA I/O。

4.2.4 高速外部收发器 phy 接口

表 4-5 高速 PHY 接口

信号名称	输入/输出	描述说明
ext_phy_resync	输出	重新同步请求。在同步丢失的情况下，重新同步可以让 phy 重新实现 bit 对齐处理。
ext_phy_din	输入	数据输入，来自 PHY。
ext_phy_kin	输入	控制输入，来自 PHY。
ext_phy_cerr	输入	编码错误。
ext_phy_perr	输入	奇偶校验错误。
ext_phy_valid	输入	数据有效指示，来自 PHY。
ext_phy_clk	输入	phy 恢复出来的时钟。
ext_phy_speed	输出	速度选择。

4.2.5 解码统计接口

表 4-6 统计接口

信号名称	输入/输出	描述说明
crc_count_out	输出	CRC 计数器，每个 STREAM 有 32 位。
dropframe_count_out	输出	丢弃帧计数器，每个 STREAM 有 32 位。
frame_count	输出	帧计数器，每个 STREAM 32 有位。
rx_count_out	输出	RX 数据包计数器，每个 STREAM 有 32 位。

drop_count_out	输出	丢弃的 RX 数据包计数器，每个 STREAM 有 32 位。
notmyid_drop_count_out	输出	错误 ID 丢弃的 RX 数据包计数器，每个 STREAM 有 32 位。
arb_select	输出	仲裁器(Arbiter)选择。

4.2.6 Control 控制接口

负责 Host 本地寄存器与 Device 寄存器读写。Control local 总线与 Avalon 总线完全兼容，支持 burst 触发模式读写，同时我们提供了 AXI4 转 Control bus 模块，用于连接 AXI 总线系统。

表 4-7 高速 PHY 接口

信号名称	输入/输出	位宽	描述说明
control_clk	输入	1	Contro interface 同步时钟，默认该时钟不启用，与 clk125_in 来源相同。
control_address	输入	32	Control interface 地址，单位为 bytes。
control_read	输入	1	Control interface 读请求，高有效。
control_write	输入	1	Control interface 写请求，高有效。
control_readdata	输出	32	Control interface 读数据输出。数据来源于 Host 本地寄存器或者 Device Bootstrap registers。
control_writedata	输入	32	Control interface 写数据，写入待传输的数据。
control_readdatavalid	输出	1	Control interface 读数据有效标志。标识当前 control_readdata 数据 是有效数据。
control_ack_code	输出	8xLINKS	Control commands 回应码
control_waitrequest	输出	1	Control interface handshake 握手信号。高有效，如果 IP 当前无法对读写请求做出回应，则该信号有效。Master 可以根据该信号做出继续等待和 timeout 的动作。
control_burstcount	输入	8	Control interface burst 尺寸，标识当前操作的 burst 大小，单位为 1 个 DWORD(32bits)。最小值为 1。
control_byteenable	输入	4	Control interface byte enable 标识。每 1 个 bit 对应 1 个 byte，如： 1111 writes full 32 bits 0011 writes lower 2 bytes 1100 writes upper 2 bytes 0001 writes byte 0 only 0010 writes byte 1 only 0100 writes byte 2 only 1000 writes byte 3 only

下图展示了 INDIRECT MODE 模式下寄存器的读写时序。

1、对于 read， 需要使用 control_address、control_read， control_read 为高发起一次读操作， control_readdatavalid 为高，表示当前读回的 control_readdata 数据有效。

2、对于 write，需要使用 control_address、control_writedata、control_byteenable 和 control_write 信号，在 control_write 高电平时有效。

3、示例中所有的操作都以 32bit 数据进行，没有使用 burst 模式，用户当然也可以利用 burst 模式完成读写，这里只是简单示例。

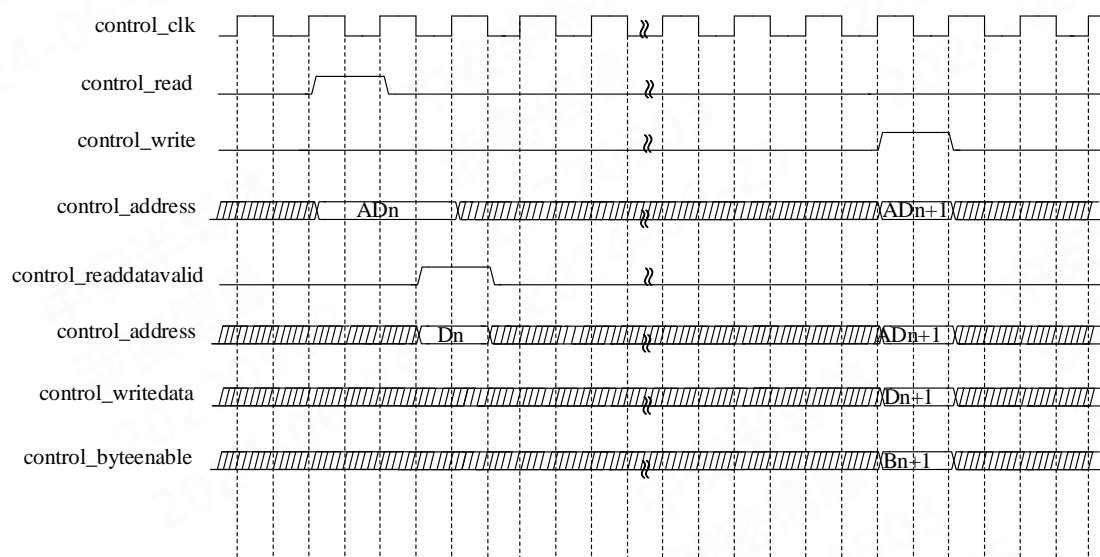


图 4-3 control bus INDIRECT 模式读写示例

下图展示了 DIRECT 模式下的 burst 读写示例，基本时序和 INDIRECT 模式一致。

1、示例写操作的 burst 数量为 4，表示当前写操作需要写入 4 个 DWORD 数据，这里不要要求数据连续，只要 control_write 有 4 个周期高电平即可。

2、示例读操作的 burst 数量为 2，表示当前读操作需要读出 2 个 DWORD 数据，control_read_datavalid 为高即表示 1 个有效的 DWORD 数据。

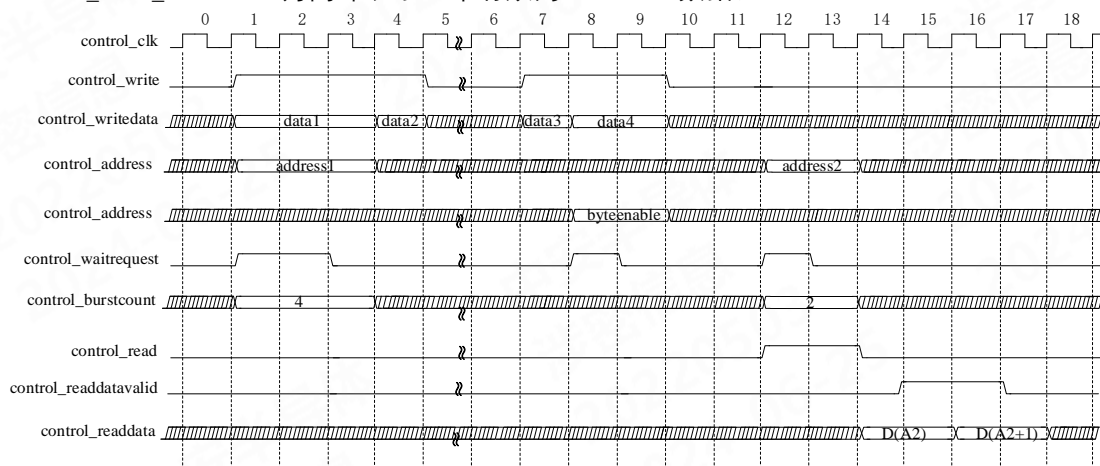


图 4-4 control bus DIRECT 模式 burst 读写示例(burst 范围 1-256)

4.2.7 Trigger 触发接口

用于 Device 和 Host 互相发送、接收触发信号。相机的触发来源通常有 3 个：

- CoaXPress 软件触发，通过 Host 向 Device 设备写寄存器实现；
 - CoaXPress 协议触发，通过 Host LINK 层触发输入信号实现，相对于软件触发实时性强，实际应用中使用较多，无需为相机连接额外的触发接线；
 - CoaXPress 相机外部硬件触发，通过相机的外部硬件接线实现，实时性最强，触发信号发出到相机收到触发信号的延迟可以忽略不计，但是需要额外接触发信号线
- 本章节描述通过 Host IP Link 层向 CoaXPress Device 设备发送触发信号的接口。

表 4-8 trigger 接口

信号名称	输入/输出	描述说明
trigger_clk	输入	可选 trigger 时钟输入。当 TRIGGER_CLK_SYNC 使能后，被用来对 trigger 信号进行同步。默认 disable，使用 125M 时钟输入。
trigger_clrn	输入	可选 trigger 复位输入。当 TRIGGER_CLK_SYNC 使能后，被用来复位同步逻辑。
trigger_in	输入	trigger 边沿选择。trigger_in[n]指示 link n。0 代表下降沿发送 trigger，1 代表上升沿发送 trigger。
trigger_in_chg	输入	Host 向 Device 发送的 trigger 脉冲。trigger_in_chg[n]指示 link n。
trigger_ack	输出	device trigger 响应信号，指示 device 是否收到了 host 发送的触发指令。trigger_ack[n]指示 link n。
trigger_out	输出	Device 到 Host 下行链路 trigger 输出。trigger_out[n*3:n] 指示 link n。实际应用中，下行链路 trigger 很少使用，绝大多数厂商不进行支持，在后续 CXP 协议中会考虑移除。
trigger_delay_out	输出	下行链路 trigger 延迟值。trigger_ack[n*7:n] 指示 link n。
trigger_out_chg	输出	trigger_out 有效指示信号。trigger_out_chg[n] 指示 link n。
missed_triggers_count_out	输出	trigger 丢失计数器输出，每个 link 32bit。
sent_triggers_count_out	输出	发送 trigger 计数器输出，每个 link 32bit。
in_chg_triggers_count_out	输出	有效 trigger 接收计数器，每个 link 32bit。
ack_triggers_count_out	输出	trigger 响应计数器，每个 link 32bit。

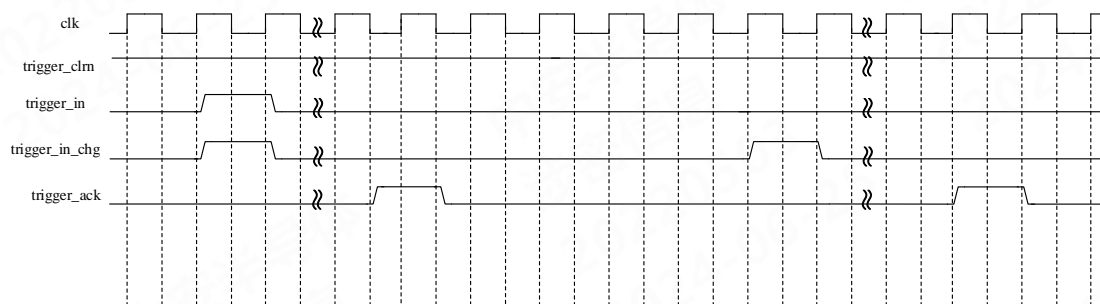


图 4-5 uplink trigger timing，Host to Device

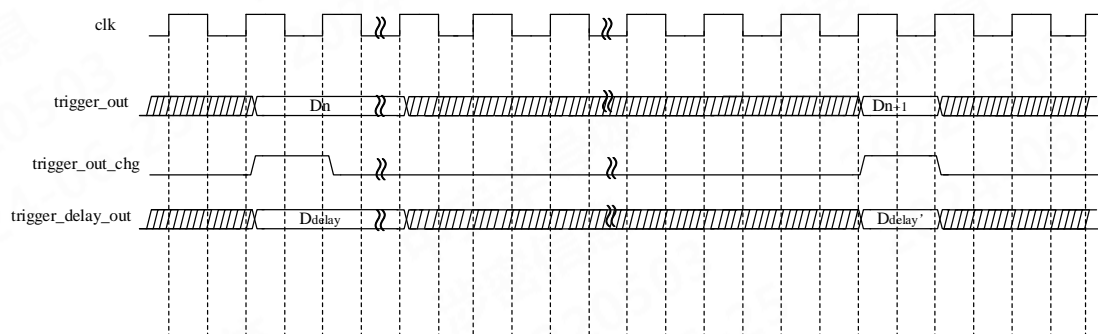


图 4-6 downlink trigger timing, Device to Host

4.2.8 HeartBeat 心跳接口

表 4-9 Heart Beat 接口

信号名称	输入/输出	描述说明
control clk	输入	接口时钟，默认使用 clk125_in
heartbeat_time	输出	心跳时间，由 device 发出。heartbeat_time[63*n:n] 指示 link n。
heartbeat_valid	输出	心跳时间有效信号。heartbeat_time[n] 指示 link n。
heartbeat_error	输出	心跳包错误指示。heartbeat_error[n] 指示 link n。
heartbeat_mhcid	输出	设备 HOST 连接 ID。 heartbeat_time[31*n:n] 指示 link n。

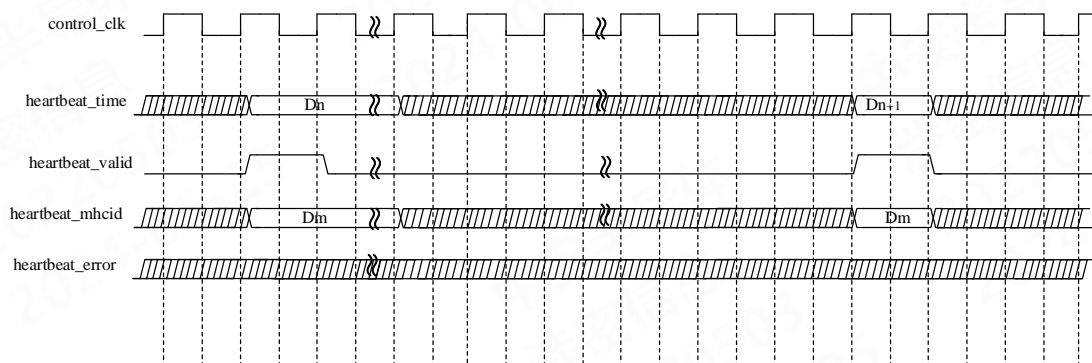


图 4-7 Heart Beat interface timing

4.2.9 EVENT 事件接口

EVENT 事件通道用于为 Device 相机设备提供一种机制，使其能够异步地向主机发送消息和状态更新。一旦 HOST IP 接收到设备发送的事件命令，event_valid 将被置为 1，同时 event_data 将传输事件消息数据的 N words(1 个 word 对应 4 字节)。数据将用 "event_startofpacket" 和 "event_endofpacket" 标识信号的开始和结束。event_dsize 表示事件数据 word 的数量 N。

EVENT 事件接口与 control_clk(默认用 clk125_in 替代)同步，并包含附加的信息信号。

EVENT 的具体使用方式请参考相机厂商定义和 CoaXPress 协议标准。

表 4-10 Event 接口

信号名称	输入/输出	位宽	描述说明
event_data	输出	32*LINKS	事件消息数据。event_data[31*n: n] 指示 link n。
event_dsize	输出	16*LINKS	事件消息数据大小，单位字。 event_dsize[15*n: n] 指示 link n。
event_startofpacket	输出	LINKS	事件消息数据 payload 开始指示。 event_startofpacket[15*n: n] 指示 link n。
event_endofpacket	输出	LINKS	事件消息数据 payload 结束指示。 event_endofpacket[15*n: n] 指示 link n。
event_valid	输出	LINKS	事件有效指示。event_valid[n] 指示 link n。
event_error	输出	LINKS	事件错误指示。event_error[n] 指示 link n。
event_crc_err	输出	LINKS	事件 CRC 校验错误指示。event_crc_err[n] 指示 link n。
event_tag	输出	8*LINKS	事件包 TAG 号。event_tag[7*n:n] 指示 link n。
event_mhcid	输出	32*LINKS	设备 HOST 从机 ID 号。 event_mhcid[31*n:n] 指示 link n。

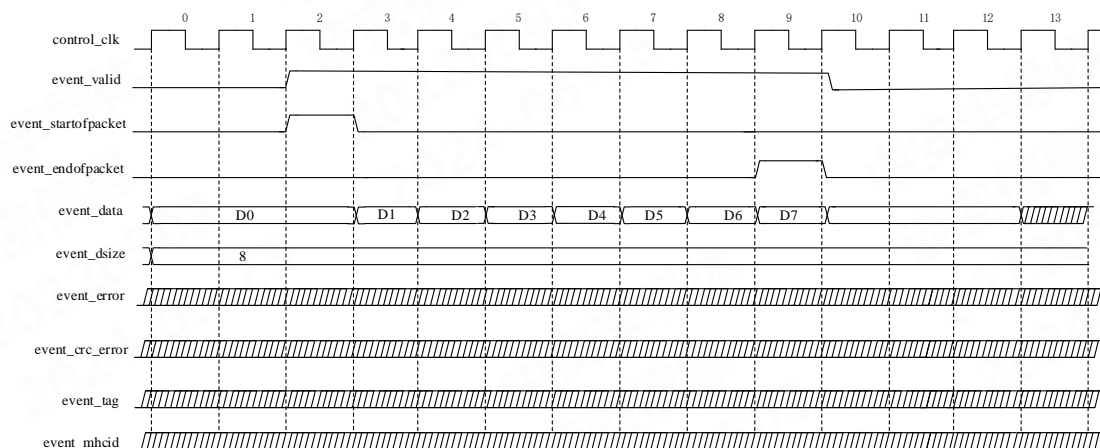


图 4-8 EVENT Timing

4.2.10 Video DMA Stream 接口

视频输出接口定义为 DMA 总线，端口描述见下表

表 4-11 Video Stream DMA 接口

信号名称	输入/输出	位宽	描述说明
dma_clk	输入	STREAMS	dma 流时钟。
dma_data	输出	32*STREAM_WORDS*STREAMS	dma 流数据。
dma_empty	输出	WIDTH_O(1)*STREAMS	dma 空指示，指示一帧的最后一次传输。
dma_valid	输出	STREAMS	dma 流数据有效信号，当次信号变低时，传输暂停。

dma_ready	输入	STREAMS	dma 流数据接收准备信号。
dma_sop	输出	STREAMS	帧开始指示。
dma_eop	输出	STREAMS	帧结束指示。
dma_sol	输出	STREAMS	行开始指示。
dma_eol	输出	STREAMS	行结束指示。

1. The WIDTH_O is defined as $\text{Log}_2(4 \times \text{STREAM_WORDS})$

Video DMA Stream 是通用的视频流接口，具体数据与像素如何对应呢？

当 STREAMS 为 1，STREAM_WORDS 为 4 时，dma_data 位宽为 128，数据排列如下：

- 如果 Device 传输像素是 8bit，则通过 DMA 总线的输出流数据将包含 8 位像素数据，每个时钟周期总共 16 个像素，pixel 0 占据 dma_data 的[7:0]，pixel 1 占据 dma_data 的[15:8]，依次类推。
- 如果 Device 传输像素是 10/12/14/16 位像素图像，则解码流将由 16 位像素组成数据，每个时钟周期总共 8 个像素。数据放在 16bit 的低 10/12/14/16 位，比如 12bit 像素输入时，第 0 个像素在 dma_data[11:0]，第 1 个像素在 dma_data[27:16]，以此类推。

DMA_SOP 和 DMA_EOP 分别指示帧的开始和帧的结束。

DMA_SOL 和 DMA_EOL 分别指示行的开始和行的结束。

DMA_EMPTY 表示最后一次传输帧中的空字节数。

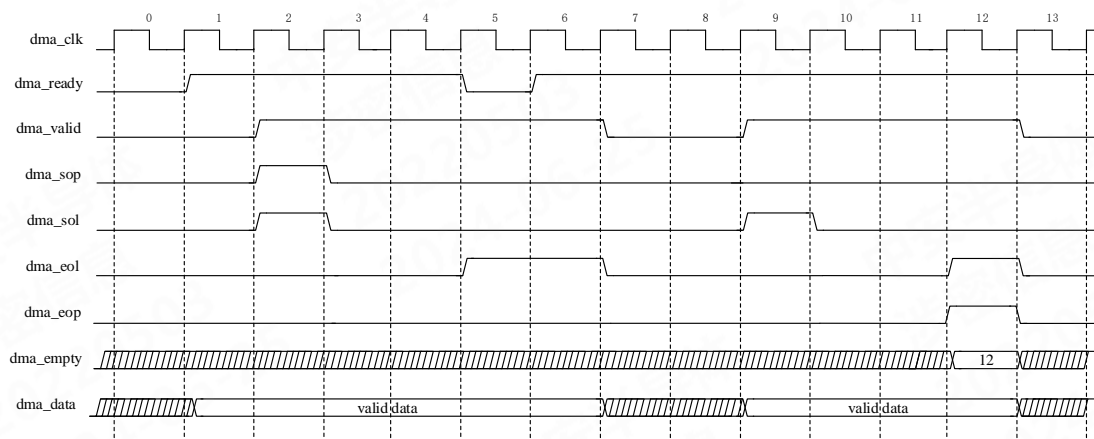


图 4-9 Video DMA timing(以 14bit 像素为例，图示共 58 个 pixels)

1. dma_valid 和 dma_data 在 dma_clk 的上升沿置位，dma_ready 信号拉高，数据完成一次传输。
2. dma_sop 和 dma_sol 在 dma_valid 和 dma_ready 同时为高时置位，指示帧的开始。
3. 一旦 dma_ready 无效，传输就会暂停，此时依靠 IP 内部 FIFO 完成缓存。
4. 一旦 dma_ready 重新置位，传输就会继续。
5. dma_valid 高有效，拉低标识传输暂停。
6. dma_valid 重新有效则传输继续。
7. dma_eop 和 dma_eol 拉高，标识帧结束。dma_empty 的值为 12，表示在数据包末尾，16 个 bytes 中有 12 个无效，4 个有效，对于当前 14bit 像素而言，则有效像素数量为 2。
8. dma_eop 置低，一帧传输完成。

DMA 时钟的频率应根据系统所需的最大数据速率来设置，与当前 LINK 数目、LINK 速率、dma_data 位宽(STREAM_WORDS)、像素格式(影响打包效率)相关。

- 最大数据速率受链路数量 (1/2/4) 和每个链路速度的影响。
- 最大数据速率受到根据图像位数设置的 Packing_Ratio 打包率的影响。即有效像素数据占据 dma_data 数据的比例，对于 8 bit 和 16bit 图像，该比率为“1”，对于 10bit 图

像，该比率为 1.6 (16/10)，对于 12bit 图像，该比率为 1.334 (16/12)，对于 14bit 图像，比率为 1.143 (16/14)。

设置 dma_clk 速率时，应当考虑最坏的情况，即最大打包率、最小数据宽度和最大期望带宽。例如：对于 10bit 图像，数据速率为 20Gbps (4 个链路，每个链路 6.25Gbps) 和 128 位 DMA 数据宽度，则最大频率为 250 MHz 即可。

以下公式描述了如何计算要提供的最大 DMA 时钟频率：

$$MAX_DMA_CLK_FREQUENCY = \frac{LINKS_NUM * LINK_SPEED * Packing_Ratio}{DMA_DATA_WIDTH}$$

4.2.11 Video Format 接口

视频格式接口说明图像的特性，接口说明如下表：

表 4-12 Video Header 接口

信号名称	输入/输出	位宽	描述说明
video_dsize	输出	24*STREAMS	数据大小。
video_xsize	输出	24*STREAMS	数据宽度。
video_ysize	输出	24*STREAMS	数据高度。
video_pixel_format	输出	16*STREAMS	像素格式。
video_image_hdr_valid	输出	STREAMS	数据有效指示信号。

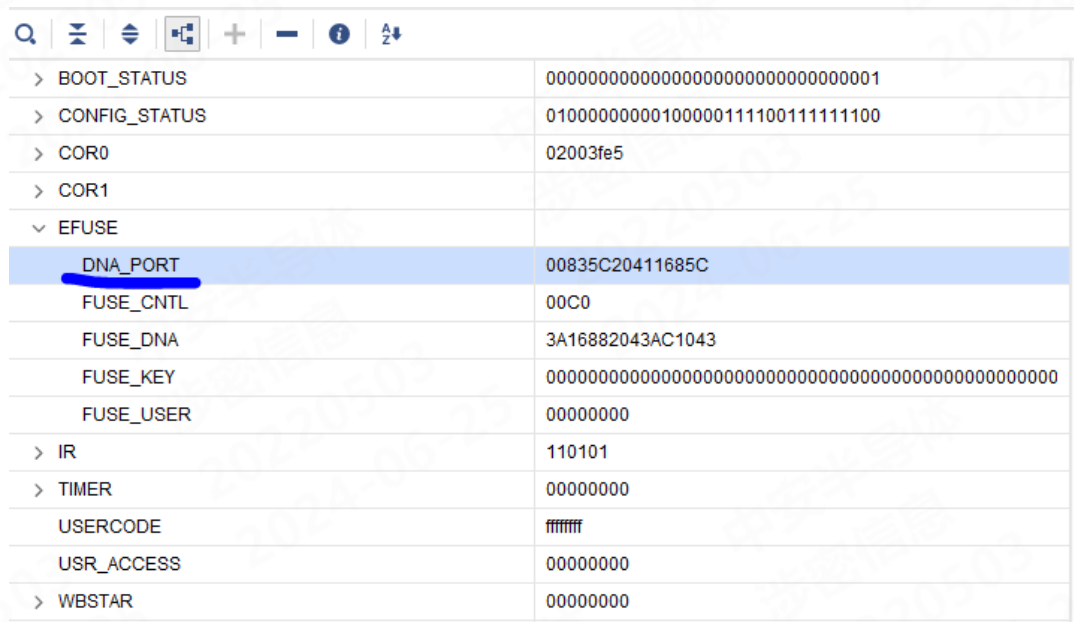
其中 video_pixel_format 定义应当参考 CoaXPress 2.0 标准或者 GenICam 标准。

4.2.12 授权接口

用于 IP 授权，仅适用于购买单个网表的用户，此时用户需要提供需要授权的 FPGA DNA 序列号。

表 4-13 授权接口

信号名称	输入/输出	描述说明
auth_ok	输出	认证通过指示。



> BOOT_STATUS	00000000000000000000000000000001
> CONFIG_STATUS	01000000000100000111100111111100
> COR0	02003fe5
> COR1	
▼ EFUSE	
DNA_PORT	00835C20411685C
FUSE_CNTL	00C0
FUSE_DNA	3A16882043AC1043
FUSE_KEY	00000000000000000000000000000000
FUSE_USER	00000000
> IR	110101
> TIMER	00000000
USERCODE	ffffff
USR_ACCESS	00000000
> WBSTAR	00000000

图 4-10 连接 JTAG 查看当前 FPGA 的 DNA 序列号

5 COAXPRESS 2.0 FPGA IP 地址空间及寄存器描述

5.1 地址空间

根据 GenCam 标准，bootstrap registers (Device 相机的基地址)为 0x00000000，Host 控制端及地址为 0x80000000，而多数设计种，IP 不会分配如此大的地址空间，因此我们支持 2 种寄存器读写方式，分别为 DIRECT 和 INDIRECT，需要在 IP 实例化时指定。推荐使用 INDIRECT 模式，为 IP 分配不超过 16KB 的地址空间，即可访问 Device 所有地址空间，且可以通过寄存器获取当前读写状态，出现错误时更容易定位。

表 5-1 描述了 DIRECT 直接访问模式下的寄存器地址，表 5-2 描述了 INDIRECT 间接访问模式下的寄存器地址

表 5-1 DIRECT 模式寄存器地址范围

寄存器地址入口	地址空间大小	寄存器组名称	描述说明
0x00000000	2G 字节	Camerabootstrap registers	HOST 通过 LINK_CH_SELECT 寄存器，选择链路上设备 (Camera) 的 bootstrap register。将其映射到 bootstrap register 相同的内存范围。根据 CoaXPRESS 标准，按大端 (Big-endian) 模式，对此寄存器空间进行读写。
0x80000000	2K 字节	IP 控制寄存器	控制 HOSTIP 操作的寄存器。详细说明见表 5-3。
0x80002000	1K 字节	解码器 Link 0 控制寄存器	控制解码器 0 操作的寄存器。详细说明见表 24。
0x80003000	1K 字节	解码器 Link 1 控制寄存器	控制解码器 1 操作的寄存器。详细说明见表 24。
0x80004000	1K 字节	解码器 Link 2 控制寄存器	控制解码器 2 操作的寄存器。详细说明见表 24。
0x80005000	1K 字节	解码器 Link 3 控制寄存器	控制解码器 3 操作的寄存器。详细说明见表 24。
0x80006000	1K 字节	解码器 Link 4 控制寄存器	控制解码器 4 操作的寄存器。详细说明见表 24。
0x80007000	1K 字节	解码器 Link 5 控制寄存器	控制解码器 5 操作的寄存器。详细说明见表 24。
0x80008000	1K 字节	解码器 Link 6 控制寄存器	控制解码器 6 操作的寄存器。详细说明见表 24。
0x80009000	1K 字节	解码器 Link 7 控制寄存器	控制解码器 7 操作的寄存器。详细说明见表 24。

表 5-2 DIRECT 模式寄存器地址范围

寄存器地址入口	地址空间大小	寄存器组名称	描述说明
0x00000000	1K 字节	Host 控制寄存器	控制 HOSTIP 操作的寄存器。 详细说明见表 5-3 和 5.13 节。
0x00002000	1K 字节	解码器 0 控制寄存器	控制解码器 0 操作的寄存器。 详细说明见表 24。
0x00003000	1K 字节	解码器 1 控制寄存器	控制解码器 1 操作的寄存器。 详细说明见表 24。
0x00004000	1K 字节	解码器 2 控制寄存器	控制解码器 2 操作的寄存器。 详细说明见表 24。
0x00005000	1K 字节	解码器 3 控制寄存器	控制解码器 3 操作的寄存器。 详细说明见表 24。
0x00006000	1K 字节	解码器 4 控制寄存器	控制解码器 4 操作的寄存器。 详细说明见表 24。
0x00007000	1K 字节	解码器 5 控制寄存器	控制解码器 5 操作的寄存器。 详细说明见表 24。
0x00008000	1K 字节	解码器 6 控制寄存器	控制解码器 6 操作的寄存器。详细说明见表 24。
0x00009000	1K 字节	解码器 7 控制寄存器	控制解码器 7 操作的寄存器。 详细说明见表 24。

5.2 寄存器描述

5.2.1 Host 控制寄存器描述

表 5-3 Host 控制寄存器

偏移地址	寄存器名称	读/写	默认值	寄存器描述说明
0x0	LINK_CH_SELECT	R/W	0x0	选择要访问的通道。下列寄存器生效于所选择的通道：LINK_SPEED, LINK_STATUS, LINK_TST_ERRORS, LINK_TST_MODE, LINK_CTRL_RESET, LINK_RESYNC, LINK_TST_RX, LINK_TST_TX, LINK_CRC,

				MISSED TRIGGERS, SENT TRIGGERS, ACKED TRIGGERS and IN TRIGGERS
0x4	LINK_SPEED	R/W	0x28	高速链接（downlink）速度选择。 0x28: 通道速率 1.250 Gbps 0x30: 通道速率 2.500 Gbps 0x38: 通道速率 3.125 Gbps 0x40: 通道速率 5.000 Gbps 0x48: 通道速率 6.250 Gbps 0x50: 通道速率 10.000 Gbps 0x58: 通道速率 12.500 Gbps
0x8	LINK_STATUS	R	0x0	bit0: FSL 收发器锁定。 bit1: FSL 同步。 bit2~bit7:保留位。 bit8~bit15:控制响应，指示控制命令完成状态，其根据 CoaXPress 标准定义。 bit16:保留位。
0xC	LINK_TST_ERRORS	R/W	0x0	链路测试错误计数器，向该寄存器写 0，可以重置 HOST 端的错误计数器。在链路测试后，该寄存器保存测试过程中发生的错误数量。
0x10	LINK_TST_MODE	R/W	0x0	向此寄存器写 1，可启用 HOST 端测试生成器，用于测试选定的链路。要停止测试，请向此寄存器写 0。
0x14	LINT_CTRL_RESET	R/W	0x0	控制通道复位。写 1 到这个寄存器，重置所选链路的控制通道。待完成后，其会自动清 0。
0x18	CONTROL_TIMEOUT	R/W	0x2710	控制操作超时计数器设置，以 125Mhz 时钟为单位。
0x1C	LINK_RESYNC	W	-	向该寄存器写 1，会触发所选链路的重新同步。
0x20	LINK_NUM	R	LINK_NUM	表示当前 IP 中配置的链接数量。

0x24	INDIRECT_ADDRESS	R/W	0x0	定义 bootstrap register 的地址。仅在间接引导模式中使用。仅用于间接引导模式。
0x28	INDIRECT_COMMAND	R/W	0x0	间接 bootstrap register 访问的命令。仅在间接引导模式下使用。 bit0: 间接写命令。向该位写 1 将开始对 bootstrap register 进行写操作。该位总是读为 0。 bit1: 间接读命令。向该位写 1, 将启动对自启动寄存器的读操作。这个位总是读为 0。 bit2: 间接传输在进行中。在这个位上读 1 表示一个间接交易仍在进行中。写入该位没有任何作用。 bit3~bit7:保留位。 bit8~bit11: 间接字节启用。选择字中哪些字节应该被读/写。 bit12~bit15:保留位。 bit16~bit23: 间接突发计数, 以 32 位字为单位。控制有多少数据将被读/写到 bootstrap register。 bit24~bit31:保留位。
0x2C	LINK_TST_RX_LSB	R/W	0x0	Rx 数据包计数器。向这个寄存器写 0 可以重置 Rx 计数器。在链路测试后, 该寄存器会记录测试期间发送的数据包的数量 (32 LSB bits)。
0x30	LINK_TST_RX_MSB	R	0x0	Rx 数据包计数器。在链接测试后, 该寄存器会记录测试期间发送的数据包数量(32 MSB bits)。
0x34	LINK_TST_TX_LSB	R/W	0x0	Tx 数据包计数器。向这个寄存器写 0 可以重置 Tx 计数器。在链路测试后, 该寄存器会记录测试期间发送的数据包的数量 (32 LSB bits)。
0x38	LINK_TST_TX_MSB	R	0x0	Tx 数据包计数器。在链接测试后, 该寄存器会记录

				测试期间发送的数据包数量(32 MSB bits)。
0x3C	ARB_SEL	R/W	0x0	选择连接到所选仲裁器(Arbiter)的链路(由CAM_SELECT寄存器)。这个寄存器是一个掩码,例如要把选定的仲裁器(Arbiter)连接到链路0,就向这个寄存器写0x01。要同时连接链路0和链路1,写0x03。要连接链路1和链接7,写0x82。
0x40	CAM_SELECT	R/W	0x0	选择将访问的Camera。ARB_SEL寄存器将生效于所选择的通道。
0x44 - 0x48	-	R	-	保留
0x4C	LINK_ORDER	R/W	(*)	Stream仲裁器(Arbiter)链接顺序,仅在使用"新仲裁器(Arbiter)模式"时相关(*)。
0x50	ARBITER RESET	R/W	0x0	向此寄存器写0,重置所选Camera的Stream仲裁器(Arbiter)(根据CAM_SELECT寄存器)。待完成后,其会自动清0。
0x54	LINK_LAYER_BUF_FLUSH	R/W	0x0	复位链路的缓冲区,其8位掩码选择。待完成后,其会自动清0。
0x58	LINK_STREAM_CRC	R/W	0x0	所选链路上发生的Stream CRC错误数量,通过写0x1来清除。
0x5C	USE_TAGS	R/W	-	CXP2.0控制命令标签使能。
0x60	CONTROL_TAG_RESET	R/W	0x0	
0x64	CONTROL_TAG_COUNTER	R	0x0	发生的TAG错误的数量。当发送的命令的TAG不等于接收的确认时,TAG错误被识别。
0x68	LINK_CONTROL_CRC	R/W	0x0	所选链路上发生的控制CRC错误的数量,通过写0x1写入bit0来清除。
0x6C	CORRECTED_CONTROL_ERRORS	R/W	0x0	所选链路上被纠正的错误数量,通过写0x1 0来清除。
0x70	UN_CORRECTED_CONTROL_ERRORS	R/W	0x0	所选链路上未能纠正的错误数量,通过向第0位写0x1来清除。

0x74	CONNECTION_LOSS_COUNTER	R/W	0x0	所选高速链接丢失的次数，通过将 0x1 写入来清除。
0x78	PHY_CERR_COUNTER	R/W	0x0	所选高速链路上的编码错误数量，通过写 0x1 来清除。
0x7C	PHY_PERR_COUNTER	R/W	0x0	所选高速链路上的奇偶校验错误数量，通过将 0x1 来清除。
0x80 - 0x84	-	R	-	保留
0x88	STREAM_BUFFER_BYTES	R	0x2000	Stream 缓冲区的大小，以字节为单位。该大小应与设备所发送的数据包大小相匹配。
0x8C	EVENT_CRC_COUNTER	R/W	0x0	所选链路上发生的事件 CRC 错误的数量。该计数器通过写 0x1 来清除。
0x90	EVENT_PACKETS_ERRORS_COUNTER	R	0x0	在所选链路上发生的错误事件数据包的数量。该计数器通过写 0x1 来清除该计数器。
0x94	CONNECTION_RESET	R/W	0x0	CXP2.0 的连接复位，重置 TAG 计数器。
0x8C - 0xBC	-	R	-	保留
0xC0	IP Revision Number	R	根据修订版本号	16 个 MSB 位是 IP 的主要版本。16 个 LSB 位是 IP 的小修订。
0xC4	CXP_VERSION	R/W	0x2	选择链接的 CoaXPress 版本： 1.0 = 0x1; 1.1 = 0x2; 2.0 = 0x4，默认上电为 1.1 版本，低速链接速率为 20.83Mbps。如果选择了 2.0 版本，当 LINK SPEED 超过 6.25Gbps 后会选择用 41.66Mbps 的低速链接速率。
0xC8 - 0xCC	-	R	-	保留
0xD0	STREAM_BUFFER_USED	R/W	0x0	所选通道的流缓冲器的使用状态。
0xD4	CONTROL_PACKET_COUNTER	R/W	0x0	在所选通道上发送的控制数据包的数量，通过写 0x1 来清除。
0xD8	STREAM_PACKET_COUNTER	R/W	0x0	在所选通道上发送的流数据包的数量，通过写 0x1 来清除。

0xDC	DROPPED_STREAM_PACKET_COUNTER	R/W	0x0	选定通道的流数据包的数量，通过写 0x1 来清除。
0xE0 - 0x3BC	-	R	-	保留
0x3C0	MISSED TRIGGERS	R/W	0x0	保存所选链路其错过触发器的计数值。向该寄存器写 0，可重置该计数器。
0x3C4	SENT_TRIGGERS	R/W	0x0	保存所选链路其发送触发器的计数值。向该寄存器写 0，可重置该计数器。
0x3C8	ACKED TRIGGERS	R/W	0x0	保存所选链路其确认触发器的计数值。向该寄存器写 0，可以重置该计数器。
0x3CC	IN_TRIGGERS	R/W	0x0	保存所选链路其请求触发器的计数值。向该寄存器写 0，可以重置该计数器。
0x3D0-0x3FF		R	-	保留。
0x400-0x7FF	INDIRECT_DATA	R/W	-	用于间接 bootstrap register 访问的数据，仅在间接引导模式下使用。

(*) 应为每台 Camera 设置链接顺序，首先选择 Camera ("CAM SELECT")，然后写入当前有效 Link 的正确顺序。

当使用 4 个链路时(4 个 LINKS)，该顺序应重复 2 次：

32 位控制数据=2 x (4bits 代表第 4 个链接位置，4bits 代表第 3 个链接位置，4bits 代表第 2 个链接位置，4bits 代表第 1 个链接位置)

当使用 2 个链接时，顺序应重复 4 次：

32 位控制数据=4 x (4 位代表第 2 个链接位置，4 位代表第 1 个链接位置)

当使用 1 个链接时，顺序应重复 8 次：

32 位控制数据=8 x (代表第 1 个链接位置的 4 位)

0x4C 寄存器的默认值是根据 LINKS 参数设置的：

对于 1 个链接，其值为 0x0。

对于 2 个 LINKS，其值为 0x10101010。

对于 4 个 LINKS，其值为 0x32103210。

表 5-4 LINK_STATUS 寄存器描述

Bit offset	Width (bits)	Description
0	1	FSL(fast speed link, downlink) 收发器锁定标志。1 标识当前 LINK 已经锁定。
1	1	FSL(fast speed link, downlink) 收发器同步标志。1 标识当前 LINK 已经同步，且正在接收有效数据。
2	6	Reserved 保留
8	8	Control acknowledge code，标识最近一次 control 指令的完成状态*
16	16	Reserved

*control ack code 定义来源于 CoaXPress 标准手册

Acknowledgment code (repeated 4 times):

Success:

0x00 Final, command executed OK, reply data is appended. (i.e. acknowledgment of read command).

0x01 Final, command executed OK, No reply data is appended. (i.e. acknowledgment of write command).

0x04 Wait. The time for the Host to wait shall be sent as a 4 byte integer in the reply data field, using the same packet structure as for a 0x00 acknowledgement. The value shall be in milliseconds, with a range of 100ms to 10s.

Logical Errors (final acknowledgments):

0x40 Invalid address.

0x41 Invalid data for the address.

0x42 Invalid control operation code.

0x43 Write attempted to a read-only address.

0x44 Read attempted from a write-only address.

0x45 Size field too large – command message (write) or acknowledgment message (read) would exceed packet size limit.

0x46 Incorrect size received, message size is inconsistent with message size indication.

0x47 Malformed packet.

Physical Errors (final acknowledgments):

0x80 Failed CRC test in last received command.

Other values are reserved for future use.

表 5-5 INDIRECT_COMMAND 寄存器描述

Bit offset	Width (bits)	Description
0	1	INDIRECT 写指令, 向该 bit 写 1 会发起一次 bootstrap registers 的写操作, 操作完成后自动清 0。当用户去读取该 bit 时, 读取值会始终为 0。
1	1	INDIRECT 读指令, 向该 bit 写 1 会发起一次 bootstrap registers 的读操作, 操作完成后自动清 0。当用户去读取该 bit 时, 读取值会始终为 0。
2	1	INDIRECT 传输正在进行标志. 1 标识当前 INDIRECT 传输尚未完成, 该 bit 不可写。
3	5	Reserved 保留
8	4	INDIRECT byte enable. 选择对当前地址 WORD (1 个 WORD==32bit==4 bytes) 的哪些 byte 进行读写
12	4	Reserved 保留
16	8	INDIRECT burst count(单位为 WORD, 4 个 bytes). 指定当前操作需要从 bootstrap registers 读写多少个 WORD 数据
24	8	Reserved 保留

5.2.2 解码器 Decoder 寄存器描述

表 5-6 Decoder 寄存器描述

偏移地址	寄存器名称	读/写	默认值	寄存器描述说明
0x0	控制权	R/W	0x0	bit0:使能传入 Stream 数据包的处理。 bit1:同步复位, 可以由用户发起。

				bit3: 旁路掉外部缓冲器，如果使用了外部缓冲器的话。其余位为保留位。
0x4		R	-	保留
0x8	CRC_COUNT	R/W	0x0	保存CRC失败的数据包计数值。写此寄存器将重置计数器，不论写入数据是什么。
0xC	RX_COUNT	R/W	0x0	保存收到的stream数据包的计数值。写此寄存器将重置计数器，不论写入数据是什么。
0x10	DROP_COUNT	R/W	0x0	保存因缓冲区使用率过高而被丢弃的流数据包数量。写此寄存器将重置计数器，不论写入数据是什么。
0x14	RX_FRAME_COUNT	R/W	0x0	保存成功接收的视频帧的数量。写此寄存器将重置计数器，不论写入数据是什么。
0x18	MAP_REG	R/W	0x0	bit[15:8]定义解码器将处理的 stream 的 ID。其余为保留位。
0x1C		R	-	保留
0x20	DROP_FRAME_COUNT	R/W	0x0	保存丢弃的帧的数量。写此寄存器将重置计数器，不论写入数据是什么。
0x24	FIFO_THRESHOLD	R/W	0x0	这个寄存器定义了外部缓冲器的阈值。
0x28	FRAMES_PER_BUFFER	R/W	0x1	在触发模式下，Stream 经过系统的帧数。
0x2C		R	-	保留
0x30	OV_COUNT	R/W	0x0	保存缓冲区满的次数。写此寄存器将重置计数器，不论写入数据是什么。
0x34	DEC_SEL	R/W	0x0	解码器与Camera的关联，这个寄存器选择解码器连接到哪个仲裁器 (Arbiter) (Camera)。例如，要将当前的解码器连接到仲裁器 (Arbiter)0，请向该寄存器写入0x00。要将解码器连接到仲裁器 (Arbiter)2，请写0x02。
0x38 - 0x3C		R	-	保留
0x40	EOL_PUSH_EN	R/W	0x0	bit0: 使在每行结束时将

				数据从解码器中推出，而不是在有足够数据时推出。当启用这个位时，"dma_sol"和"dma_eol"信号无效。
0x44	ID_AND_DISABLE_DROP_COUNT	R/W	0x0	保存由于错误的StreamID或由于控制寄存器中的使能位没有设置而丢弃的流数据包的数量。写此寄存器将重置计数器，不论写入数据是什么。
0x48	STREAM_ID	R	0x0	传入Stream数据的ID。
0x4C	PIXEL_FORMAT	R	0x0	来自图像header的像素格式。
0x50	IMAGE_X_SIZE	R	0x0	来自图像header的图像X尺寸。
0x54	IMAGE_Y_SIZE	R	0x0	来自图像header的图像Y大小。
0x58	DATA_SIZE	R	0x0	来自图像header的图像数据大小。
0x5C	WIDTH_ERROR_COUNTER	R/W	0x0	可选宽度的错误计数器，写这个寄存器将重置计数器。
0x60	HEIGHT_ERROR_COUNTER	R/W	0x0	可选高度的错误计数器，写这个寄存器将重置该计数器。
0x64	STREAMID_FILTER_DISABLE	R/W	0x1	禁用可选的Stream ID过滤，当设置位0为高时（默认为禁用）。
0x68	USER_PIXEL_FORMAT	R/W	0x0	可选的用户定义的图像像素格式。
0x6C	USER_WIDTH	R/W	0x0	可选的用户定义的图像宽度。
0x70	USER_HEIGHT	R/W	0x0	可选的用户定义的图像高度。
0x74	USER_DATA_SIZE	R/W	0x0	可选的用户定义的图像数据大小。
0x78	USER_WIDTH_ERROR_COUNTER	R/W	0x0	可选的错误计数器，用于比较图像的宽度和用户定义的宽度（每帧）。写这个寄存器将重置该计数器。
0x7C	USER_HEIGHT_ERROR_COUNTER	R/W	0x0	可选的错误计数器，用于比较图像的高度和用户定义的高度（每一帧）。写这个寄存器将重置该计数器。
0x80	USER_DATA_SIZE_ERROR_COUNTER	R/W	0x0	可选的错误计数器，用于比较图像的数据大小与用户定义的数据大小（每帧）。写这个寄存器将重置该计数器。

0x84	USER_PIXEL_FORMAT_ERROR_COUNTER	R/W	0x0	可选的错误计数器，用于比较图像的像素格式和用户定义的像素格式（每帧）。写这个寄存器将重置该计数器。
------	---------------------------------	-----	-----	---

5.3 DEVICE 寄存器访问模式

5.3.1 INDIRECT 间接访问

如果 Host IP 被配置为间接模式读写 Device 的 bootstrap registers，需要进行一些操作来访问寄存器，将读写动作拆开成多个步骤来实现，不能像 DIRECT 模式一下使用单条指令实现。

操作流程如下(我们提供了 C 代码 demo 源程序实现该流程)：

写bootstrap registers操作，必须按照下述流程：

1. 在LINK_CH_SELECT寄存器中写入你要访问的LINK链路序号，比如4 LINK连接单个相机，那么LINK 0作为master，在访问bootstrap registers之前，需要先选中LINK 0。
2. 将bootstrap registers地址写到INDIRECT_ADDRESS寄存器。
3. 将数据写入INDIRECT_DATA寄存器。如果是burst操作（最多256个32位字），将数据以完整的32位字写入连续的地址中，地址增量为4。如果是单个WORD，将数据写入INDIRECT_DATA寄存器中，并加上对应的字节偏移。
4. 将突发计数值（最小为1个WORD，32bit）写入INDIRECT_COMMAND寄存器，同时将WRITE位打开。
5. 持续读取INDIRECT_COMMAND寄存器，直到 "间接传输中" 传输中位为0。进展位为0。
6. 读取LINK STATUS寄存器以获得完成状态。

读bootstrap registers操作，必须按照下述流程：

1. 在LINK_CH_SELECT寄存器中写入你要访问的LINK链路序号，比如4 LINK连接单个相机，那么LINK 0作为master，在访问bootstrap registers之前，需要先选中LINK 0。
2. 将bootstrap registers地址写到INDIRECT_ADDRESS寄存器。
3. 将burst计数值（最小为1个WORD，32bit）写入INDIRECT COMMAND寄存器，同时将READ bit标志位设为1。
4. 持续读取INDIRECT COMMAND寄存器，直到 "INDIRECT" 正在传输中标志位为0。
5. 读取LINK STATUS寄存器以获取当前操作的完成状态。
6. 如果是burst mode（最多256个32位字），从连续的地址中读取完整的32位字的数据，地址增量为4。如果是单个word，应从INDIRECT DATA寄存器中读取数据，并设置相关的地址偏移。

5.3.2 DIRECT 直接访问

DIRECT 模式从软件上来看会比较直观，与读取 HOST 寄存器没有区别，只是基地址不同，但是在地址寻址时需要小心分配寻址空间。

写 bootstrap registers 操作，必须按照下述流程：

- 1、向 LINK CH SELECT 寄存器中写入你要访问的 LINK 链路号，比如 4 LINK 连接单个相机，那么 LINK 0 作为 master，在访问 bootstrap registers 之前，需要先选中 LINK 0。
- 2、写下要寻址的 bootstrap registers 地址，地址空间被映射到设备的 bootstrap register 的相同内存范围

读 bootstrap registers 操作，必须按照下述流程：

- 1、在 LINK CH SELECT 寄存器中写入你要访问的 LINK 链路号，比如 4 LINK 连接单个相机，那么 LINK 0 作为 master，在访问 bootstrap registers 之前，需要先选中 LINK 0。
- 2、写下要寻址的 bootstrap registers 地址，地址空间被映射到设备的 bootstrap register 的相同内存范围。

6 IP 软件配置

6.1 基本的 CAMERA 检测和采集启动

下文描述了配置为1个LINK，1个Stream的Camera基本检测和采集启动流程（在 INDIRECT 间接访问模式下）。

1. 向HOST控制寄存器中选择链路LINK 0，将0写入 LINK_CH_SELEC 寄存器（在HOST寄存器0x0处）。
2. 通过向Camera的bootstrap register LINK_RESET寄存器（Camera寄存器0x4000）写"1"，在Camera一侧重置链接。
3. 等待200ms。
4. 将HOST LINK 0的速度设置为基本通用速度，写入0x28或0x38（根据Camera的数据手册进行设置）到 LINK_SPEED (HOST寄存器0x4)寄存器，CoaXPress支持1.25Gbps和3.125Gbps 2个发现速率。
5. 读取Camera的bootstrap register LINK_Config寄存器（Camera寄存器0x4014），获取当前默认LINK数量、LINK速度，然后使用默认速度连接Camera。
6. 读取HOST控制寄存器中的LINK_STATUS寄存器（HOST寄存器地址0x8），看LINK链路是否同步（此寄存器的第0位为 "1" 表示同步成功）。
7. 从Camera的 bootstrap register （在Camera寄存器0x0处）读取标准的魔术数字（0xC0A79AE5），验证链接成功。
8. 从Camera的bootstrap register ConnectionConfigDefault(0x4018)，获取相机推荐配置，比如0x40058 表示当前相机推荐配置为4个LINK，每个LINK速率12.5Gbps。
9. 根据推荐LINK数量，在发现速率下检查当前实际连接的LINK数量，比如1个4Lane CXP相机，并不一定4个Lane全部连接到了采集设备，可能只连接了1个Lane。
10. 将新的运行速度(比如0x58)和在LINK数量(比如4)写入Camera的bootstrap register LINK_Config寄存器（Camera寄存器0x4014）。

11. 将相同的操作速度写入HOST控制的LINK_SPEED寄存器（HOST寄存器0x4）。
12. 等待200ms。
13. 读取LINK_STATUS寄存器（HOST寄存器0x8），看链接是否同步（"2'b11"表示同步成功）。
14. 从Camera的 bootstrap register（地址为0x0）中读取标准的魔法号码（0xC0A79AE5）。
15. 从Camera的 bootstrap register（Camera寄存器地址0x0 - 0x1C、0x2000 - 0x20C0、0x3000 - 0x3018和0x4000 - 0x403C）读取Camera的参数，详细请参考camera厂商提供的xml文件或者手册。
16. 向Camera的MasterHostConnectionID寄存器（设备寄存器0x4008）写入一个唯一的Host Link ID。
17. 向Camera的StreamPacketSizeMax寄存器（Camera寄存器0x4010），将最大stream packet包大小设置为0x400。
18. 当使用 "新仲裁器(Arbiter)模式"时，设置活动链接的顺序（见表5-3HOST寄存器地址映射下方注释）。
19. 通过向CONTROL寄存器（HOST寄存器0x2000）写入0x2来重置解码器0。
20. 将StreamID写入解码器0的MAP_REG寄存器（HOST寄存器0x2018）的位[15:8]。
StreamID位于Image1StreamIDAddress（Camera寄存器0x301C），需要从Camera中读取。
21. 在HOST控制寄存器中选择Camera0，向HOSTCAM_SELECT寄存器（HOST寄存器0x040）写0x0。
22. 向HOSTARB_SEL寄存器（HOST寄存器0x3C）写0x1，将链路0连接到仲裁器(Arbiter)0。
23. 写0x0到DEC_SEL(HOST寄存器0x2034)来连接仲裁器(Arbiter)0到解码器0。
24. 通过向Camera bootstrap register 的AcquisitionStart寄存器写 "1"来启动采集（该寄存器的地址写在AcquisitionStartAddress，在Camera寄存器0x300C，寄存器地址从Camera中读出，或者由厂商提供的xml文件给出）

6.2 LINK 链路到 CAMERA（仲裁器 ARBITER）的链接

由于 CoaXPress HOST IP 可以支持多个链路和多个 Camera 同时连接，所以需要通过 **Control Channel interface** 寄存器手动配置连接。为了将正确的链路连接到正确的仲裁器 (Arbiter)（连接到同一台 Camera 的所有链路），应该使用 CAM_SELECT 寄存器选择正确的相关仲裁器(Arbiter)，并向 ARB_SEL 寄存器写入正确的 mask 掩码。

下图演示了两台 Camera 的设置，其中一台 Camera 与 4 个链路相连，另外一台 Camera 不连接。为了正确配置，应该把 0x0 写到 CAM_SELECT 寄存器中选择第一个仲裁器 (Arbiter)，然后将 0x0F 写到 ARB_SEL 寄存器中写 0x0F，将 link0、link1、link2 和 link3 连接到这个仲裁器(Arbiter)。接下来，还应该写 0x01 到 CAM_SELECT 寄存器选择第二个仲裁器 (Arbiter)，然后将 0x00 写到 ARB_SEL 寄存器，以确保没有链接连接到第二个仲裁器(Arbiter)

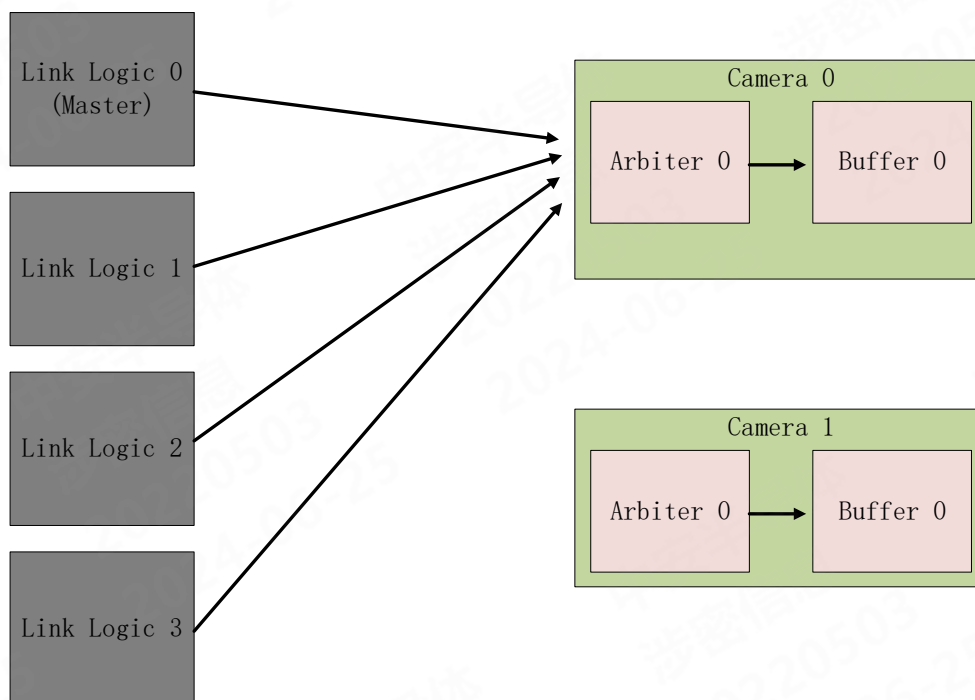


图 6-1 2 台相机与 4 个 LINK 的连接设置(1)

下图演示了一台 Camera 用 2 个 LINK 链接连接到第一个仲裁器(Arbiter)，另一台 Camera 用 1 个链接连接到第二个仲裁器(Arbiter)的设置。为了正确配置这种设置，应该向 CAM_SELECT 寄存器写 0x00 以选择第一个仲裁器(Arbiter)，然后向 ARB_SEL 寄存器写 0x03 以连接链接 0 和链接 1 到这个 Camera 的仲裁器(Arbiter)。接下来，还应该向 CAM_SELECT 寄存器写 0x01 来选择第二个仲裁器(Arbiter)，然后向 ARB_SEL 寄存器写 0x04 来连接 link2 到这个 Camera 的仲裁器(Arbiter)。

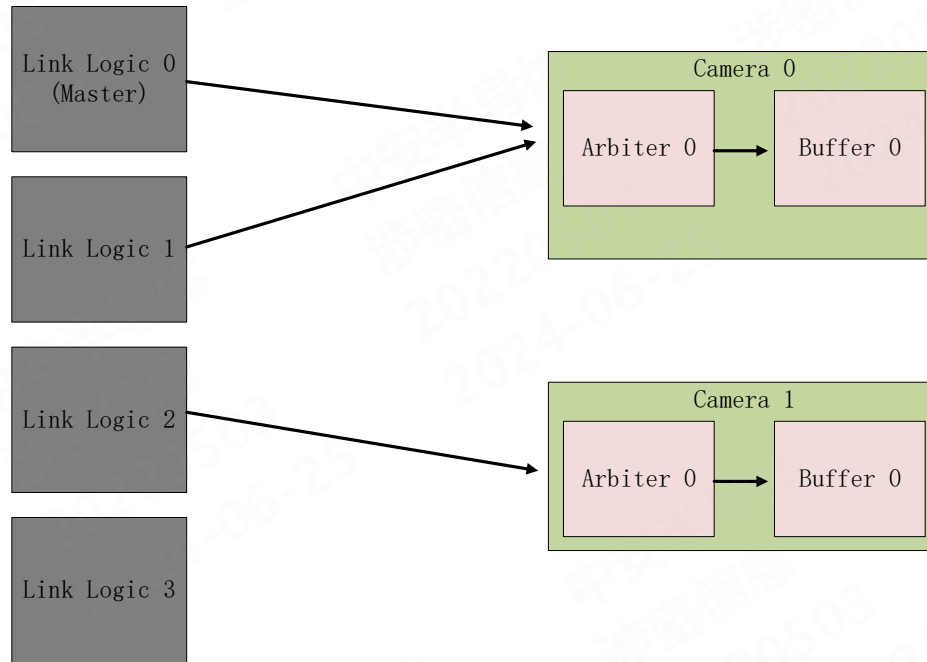


图 6-2 2 台相机与 4 个 LINK 的连接设置(2)

6.3 CAMERA（仲裁器 ARBITER）与解码器的链接

HOST IP 支持同时连接多个 Camera，所以需要通过 *Control Channel interface* 寄存器手动配置连接。为了将正确的仲裁器及其缓冲区连接到正确的解码器（每个流应连接到不同的解码器），需要向 DEC_SEL 寄存器写入正确的 mask 掩码。

下图展示了两个 Camera 和两个 Decoder 解码器的设置，第 0 个 Camera 连接到第 0 个解码器，第 1 个 Camera 连接到第 1 个解码器。为了正确配置，应该访问第一个解码器（间接访问模式的地址空间从 0x00002000 开始，直接访问模式从 0x80002000 开始）并将 0x0 写到 DEC_SEL 寄存器，然后访问第二个解码器（间接访问模式的地址空间从 0x00003000 开始，直接访问模式从 0x80003000 开始），并将 0x1 写到 DEC_SEL。

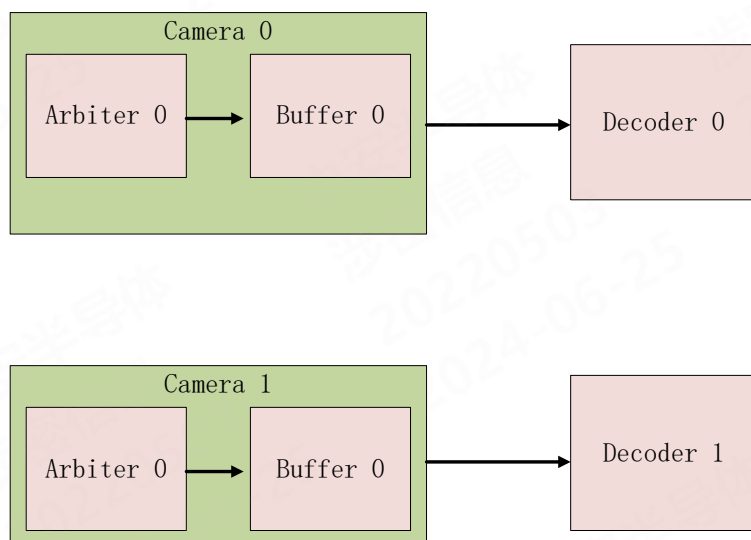


图 6-3 2 台相机与 2 个解码器的连接

下图展示了2台Camera（3个仲裁器Arbiter）和3个解码器的设置，第0台Camera连接到第0和第1解码器，第1台Camera连接到第2解码器，第2台 Camera断开连接。为了正确配置这种设置，应该访问第0个解码器（间接访问模式的地址空间从0x00002000开始，直接访问模式从0x80002000开始）并将0x0写入 DEC_SEL 寄存器写0。接下来，应该访问第1个解码器（间接访问模式的地址空间从0x00003000开始，直接访问模式从0x80003000开始）并将0x0写到DEC_SEL 寄存器。最后，我们应该访问第2个解码器（间接访问模式的地址空间从0x00004000开始，直接访问模式从0x80004000开始），并将0x1写到寄存器中。

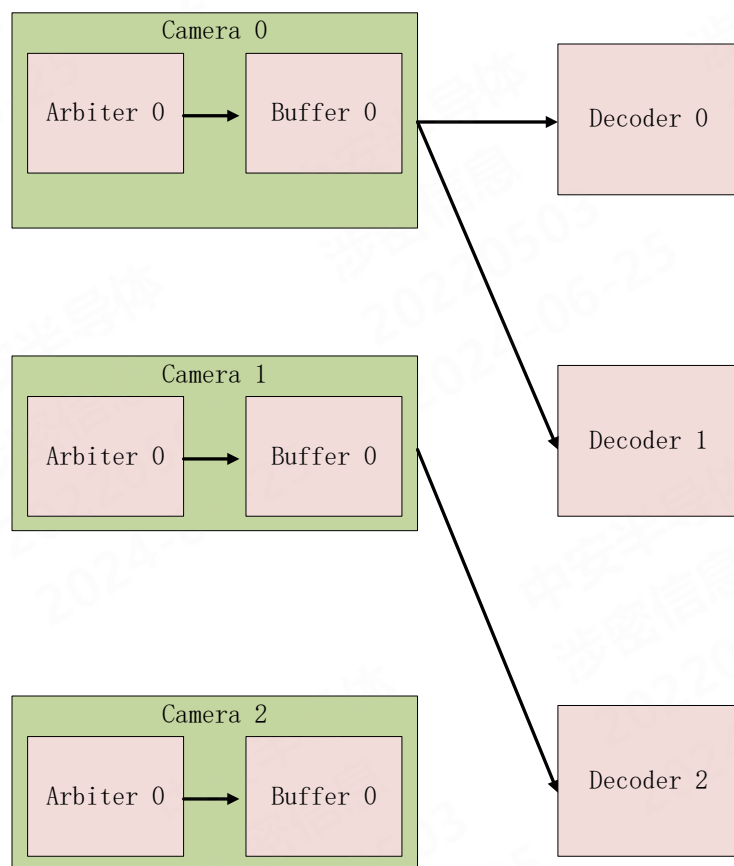


图 6-4 3 台相机与 3 个 Decoder 的连接

Annex A: 参考资料

- 1、CoaXPess JIIA CXP-001-2021
- 2、<https://www.cnblogs.com/xingce/category/2165251.html>

Annex B: 常见问题

实际只接了 1 条 LANE，相机可以用吗？

可以。通常 CoaXPess IP 设置为 4 Lane 设计，那么从 IP 读出的 LINK_NUM 寄存器就固定为 4。如果相机为 4 Lane 相机，但只接了 1 Lane，从相机读出的 ConnectionConfigDefault 寄存器依然会是 0x400xx。用户驱动代码需要通过检查每条 Lane 是否 SYNC 来确定具体的连接状态，从而确定最终相机 ConnectionConfig 寄存器的设置，比如只有 1 条 Lane SYNC，推荐速率为 0x48，那么 ConnectionConfig 寄存器应设置为 0x10048。

```
<Enumeration Name="ConnectionConfig">
  <ToolTip>Holds a valid combination of Device connection speed and number of active connections.</
  ToolTip>
  <Description>Holds a valid combination of Device connection speed and number of active connections.</
  Description>
  <DisplayName>Connection Configuration</DisplayName>
  <Visibility>Beginner</Visibility>
  <pIsLocked>IsRunning</pIsLocked>
  <EnumEntry Name="CONNECTION4SPEED3125">
    <Value>0x40038</Value>
  </EnumEntry>
  <EnumEntry Name="CONNECTION4SPEED5000">
    <Value>0x40040</Value>
  </EnumEntry>
  <EnumEntry Name="CONNECTION4SPEED6250">
    <Value>0x40048</Value>
  </EnumEntry>
  <pValue>ConnectionConfigReg</pValue> </Enumeration>
```

图 5 相机支持的配置示例