




Gowin DC-SCM LTPI IP 用户指南

IPUG1086-1.0,2024-02-08

版权所有 © 2024 广东高云半导体科技股份有限公司

GOWIN高云, , Gowin以及高云均为广东高云半导体科技股份有限公司注册商标, 本手册中提到的其他任何商标, 其所有权利属其拥有者所有。未经本公司书面许可, 任何单位和个人都不得擅自摘抄、复制、翻译本档内容的部分或全部, 并不得以任何形式传播。

免责声明

本档并未授予任何知识产权的许可, 并未以明示或暗示, 或以禁止反言或其它方式授予任何知识产权许可。除高云半导体在其产品的销售条款和条件中声明的责任之外, 高云半导体概不承担任何法律或非法律责任。高云半导体对高云半导体产品的销售和 / 或使用不作任何明示或暗示的担保, 包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等, 均不作担保。高云半导体对档中包含的文字、图片及其它内容的准确性和完整性不承担任何法律或非法律责任, 高云半导体保留修改档中任何内容的权利, 恕不另行通知。高云半导体不承诺对这些档进行适时的更新。

版本信息

日期	版本	说明
2024/02/08	1.0	初始版本。

目录

目录	i
图目录	iii
表目录	iv
1 关于本手册	1
1.1 手册内容	1
1.2 相关文档	1
1.3 术语、缩略语	1
1.4 技术支持与反馈	2
2 功能简介	3
2.1 概述	3
2.2 主要特征	4
2.3 资源利用	4
3 功能描述	5
3.1 系统框图	5
3.1.1 DC-SCM 2.0 LTPI 结构	5
3.1.2 IP 实现框图	6
3.2 端口列表	8
3.3 接口时序	11
3.3.1 配置接口时序	11
3.3.2 数据接口时序	11
3.3.3 GPIO 接口时序	11
3.3.4 UART 接口时序	11
3.3.5 IIC/SMBus 接口时序	11
3.4 寄存器描述	12
4 设计说明	17
4.1 线速率	17
4.2 时钟	17
4.2.1 系统时钟	17

4.2.2 参考时钟.....	17
4.2.3 配置时钟(cfg clk).....	17
4.2.4 LVDS 时钟.....	17
4.2.5 RX/TX DIV5 时钟.....	17
4.2.6 时钟结构.....	18
4.3 外部通道接口处理.....	18
4.3.1 GPIO 通道.....	19
4.3.2 UART 通道.....	20
4.3.3 IIC/SMBus 通道.....	21
4.3.4 OEM 通道.....	25
4.3.5 数据通道.....	25
4.4 LTPI 帧格式.....	26
4.4.1 帧格式.....	26
4.4.2 具体帧定义.....	27
4.5 链路初始化以及运行流程.....	31
4.5.1 状态图.....	31
4.5.2 链路训练.....	32
4.5.3 链路配置.....	33
4.5.4 链路运行.....	34
5 界面配置.....	35
6 参考设计.....	40
6.1 应用.....	40
6.2 参考设计.....	40
6.2.1 硬件平台.....	40
6.2.2 实现框图.....	41
6.2.3 总线协议与地址规划.....	42
6.3 板级测试.....	43
6.3.1 操作说明.....	43
6.3.2 操作步骤.....	43
7 文件交付.....	46
7.1 文档.....	46
7.2 设计源代码（加密）.....	46
7.3 参考设计.....	46

图目录

图 3-1 LTPI 结构图	5
图 3-2 LTPI 各个通道传输示意图.....	6
图 3-3 DC-SCM LTPI IP 实现框图.....	6
图 3-4 Gowin DC-SCM LTPI IP 端口示意图	8
图 3-5 配置接口写时序图.....	11
图 3-6 配置接口读时序图.....	11
图 4-1 典型时钟方案	18
图 4-2 UART 采样与恢复示意图.....	20
图 4-3 IIC 总线事务交换	22
图 4-4 LTPI 链路状态图	31
图 5-1 打开 DC-SCM LTPI IP	35
图 5-2 Gowin DC-SCM LTPI IP 配置界面 1.....	36
图 5-3 Gowin DC-SCM LTPI IP 配置界面 2.....	37
图 5-4 Gowin DC-SCM LTPI IP 配置界面 3.....	37
图 5-5 Gowin DC-SCM LTPI IP 配置界面 4.....	38
图 5-6 Gowin DC-SCM LTPI IP 配置界面 5.....	38
图 5-7 Gowin DC-SCM LTPI IP 配置界面 6.....	38
图 5-8 Gowin DC-SCM LTPI IP 配置界面 7.....	39
图 5-9 Gowin DC-SCM LTPI IP 配置界面 8.....	39
图 6-1 HPM 和 SCM 的连接	40
图 6-2 平台框图	40
图 6-3 参考设计实现框图.....	41

表目录

表 1-1 术语、缩略语	1
表 2-1 Gowin DC-SCM LTPI IP 概述	3
表 2-2 Gowin DC-SCM LTPI IP 占用资源	4
表 3-1 端口定义	9
表 3-2 Gowin DC-SCM LTPI IP 寄存器	12
表 3-3 速度描述	15
表 3-4 通告帧能力描述	15
表 3-5 串口通道最大波特率定义表	16
表 4-1 外部通道接口处理	18
表 4-2 GPIO 字段配置	19
表 4-3 UART 字段配置	21
表 4-4 IIC/SMBus 通道事务定义	21
表 4-5 IIC/SMBus 字段配置	22
表 4-6 IIC/SMBus 事务流程	23
表 4-7 数据通道事务定义	25
表 4-8 数据通道字段配置	25
表 4-9 帧格式说明	26
表 4-10 链路训练阶段帧类型	27
表 4-11 Link Detect 帧定义	27
表 4-12 Link Speed 帧定义	27
表 4-13 链路配置阶段帧类型	28
表 4-14 Advertise 帧定义	28
表 4-15 Configure 帧定义	28
表 4-16 Accept 帧定义	29
表 4-17 运行阶段帧类型	30
表 4-18 Default I/O 帧定义	30
表 4-19 Default Data 帧定义	31
表 6-1 KEY 和 LED 的映射路线表	42
表 6-2 寄存器地址定义	42

表 7-1 文档列表	46
表 7-2 Gowin DC-SCM LTPI IP 设计源代码列表.....	46
表 7-3 Gowin DC-SCM LTPI IP RefDesign 文件夹内容列表.....	46

1 关于本手册

1.1 手册内容

Gowin® DC-SCM LTPI IP 用户指南主要包括功能简介、功能描述、GUI 调用参考设计等，旨在帮助用户快速了解 Gowin DC-SCM LTPI IP 的产品特性、特点及使用方法。本手册中的软件界面截图参考的是 1.9.9.01 (64-bit) 版本，因软件版本升级，部分信息可能会略有差异，具体以用户软件版本的信息为准。

1.2 相关文档

通过登录高云®半导体网站 www.gowinsemi.com.cn 可以下载、查看 FPGA 产品相关文档：

- [SUG100, Gowin 云源软件用户指南](#)
- [DS961, GW2ANR 系列 FPGA 产品数据手册](#)
- [DS102, GW2A 系列 FPGA 产品数据手册](#)
- [DS226, GW2AR 系列 FPGA 产品数据手册](#)
- [DS971, GW2AN-18X & 9X 器件数据手册](#)
- [DS976, GW2AN-55 器件数据手册](#)

1.3 术语、缩略语

本手册中出现的相关术语、缩略语及相关释义如表 1-1 所示。

表 1-1 术语、缩略语

术语、缩略语	全称	含义
APB	Advanced Peripheral Bus	外围总线
CRC	Cycle Redundancy Check	循环冗余检查
CSR	Control and Status Register	控制和状态寄存器
DC-SCI	Datacenter-ready Secure Control Interface	数据中心就绪的安全控制接口
DC-SCM	Datacenter-ready Secure Control Module	数据中心就绪的安全控制模块

术语、缩略语	全称	含义
FPGA	Field Programmable Gate Array	现场可编程门阵列
GPIO	Gowin Programmable I/O	Gowin可编程通用管脚
HPM	Host Processor Module	主机处理器模块
IIC	Inter-Integrated Circuit	内部集成电路
IP	Intellectual Property	知识产权
LPTI	LVDS Tunneling Protocol and Interface	LVDS隧道协议和接口
LVDS	Low Voltage Differential Signaling	低压差分信号
MCSI	Multi-Channel Serial Interface	多通道串行接口
OEM	Original Equipment Manufacturer	原始设备制造商
SCM	Secure Control Module	安全控制模块
SerDes	Serializer/Deserializer	串行器和解串器
SMBus	System Management Bus	系统管理总线
TDM	Time Division Multiplexing	时分复用
UART	Universal Asynchronous Receiver/Transmitter	通用异步接收器/发送器

1.4 技术支持与反馈

高云半导体提供全方位技术支持，在使用过程中如有任何疑问或建议，可直接与公司联系：

网址：www.gowinsemi.com.cn

E-mail：support@gowinsemi.com

Tel: +86 755 8262 0391

2 功能简介

2.1 概述

LTPI (LVDS Tunneling Protocol & Interface, LVDS 隧道协议和接口)是一种协议和接口, 设计用于在主机处理器模块 (Host Processor Module, HPM) 和安全控制模块 (Secure Control Module, SCM) 之间传输各种低速信号。它不仅支持 GPIO 的隧道传输, 还支持 IIC、UART 等低速串行接口的隧道传输。它还可以通过额外的专有 OEM 接口进行扩展, 并为 HPM FPGA 和 SCM FPGA 之间的原始数据隧道提供支持。Gowin DC-SCM LTPI IP 符合数据中心就绪安全控制模块 DC-SCM 2.0 LTPI 规范, 并具有标准化的数据中心就绪安全控制接口(DC-SCI), 聚合多个数据通道, 例如 IIC/SMBbus、GPIO 和 UART, 为客户的系统 and 电路板设计增加更多灵活性。

表 2-1 Gowin DC-SCM LTPI IP 概述

Gowin DC-SCM LTPI IP	
逻辑资源	参见表 2-2
交付文件	
设计文件	Verilog(encrypted)
参考设计	Verilog
TestBench	Verilog
测试设计流程	
综合软件	GowinSynthesis®
应用软件	Gowin Software (V1.9.9.Beta-1 及以上)

注!

可登录[高云半导体网站](#)查看芯片支持信息。

2.2 主要特征

- 符合 DC-SCM2.0 协议规范
- 链路初始化、发现和协商
- 支持多通道串行接口
- 支持 GPIO、IIC/SMBus、UART、OEM 和数据通道聚合
- 一共支持多达 5 个通道的聚合/分解
- 支持多达 1023 个正常延迟 GPIO
- IIC/SMBus 的每个通道都可以选择切换 Controller 或者 Target
- OEM 通道扩展为类似低延迟 GPIO 接口
- 支持 AMBA 3 APB 协议 v1.0，用于软 IP 和数据通道的寄存器访问
 - 支持 PREADY，指示 APB 传输完成的就绪信号
 - PSLVERR 仅在与数据通道相关的访问中受支持（传输失败的错误指示信号）
- 在 GW2A 器件上支持 LVDS 数据速率最高为 300Mbps（暂定）。

2.3 资源利用

通过 Verilog 语言实现 DC-SCM LTPI IP。因使用器件的密度、速度和等级不同，其性能和资源利用情况可能不同。以高云 GW2A 系列 FPGA 为例，DC-SCM LTPI IP 资源利用情况如表 2-2 所示。

表 2-2 Gowin DC-SCM LTPI IP 占用资源

配置	占用资源					
	Register	LUT	ALU	BSRAM	SSRAM	PLL
配置为 SCM 使能 GPIO、IIC、UART、OEM、Data 通道 LLGPIO 设置为 16 NLGPIO 设置为 32 UART 接口设置为 1 IIC 接口设置为 1 OEM GPIO 设置为 32	2535	2708	581	0	0	1
配置为 HPM 使能 GPIO、IIC、UART、OEM、Data 通道 LLGPIO 设置为 16 NLGPIO 设置为 32 UART 接口设置为 1 IIC 接口设置为 1 OEM GPIO 设置为 32	2611	2794	596	0	1	1

3 功能描述

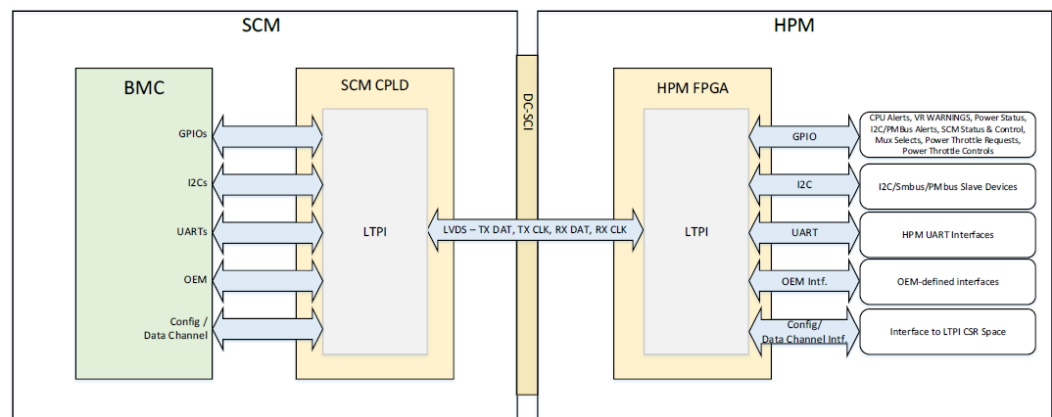
3.1 系统框图

3.1.1 DC-SCM 2.0 LTPI 结构

LTPI 接口可以通过两个 FPGA 器件来实现，以便与典型的 BMC 器件进行配置：

- HPM FPGA: 提供本地 HPM 接口与 LTPI 的桥接
- SCM FPGA: 提供到本地 SCM 接口的 LTPI 桥接

图 3-1 LTPI 结构图



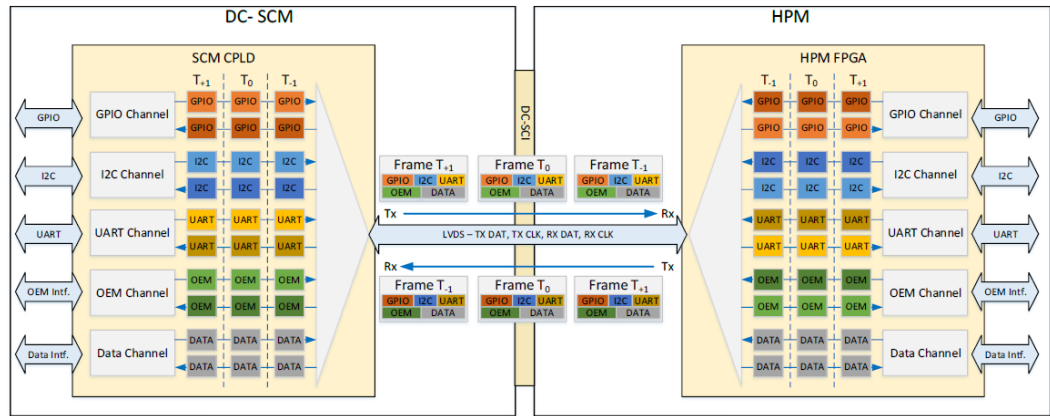
DC-SCM LTPI 接口实现了通道的概念，其中每个通道都映射到 DC-SCM 和 HPM 上的特定类型的物理接口。LTPI 接口定义了以下通道：

- GPIO 通道
- IIC/SMBus 通道
- UART 通道
- OEM 渠道
- 数据通道

LTPI 使用高速 LVDS 全双工链路的时分复用（TDM）在 SCM 和 HPM 之间发送和接收 LTPI 信道。如图 3-2 所示，在 LVDS 链路上的每个相等时隙

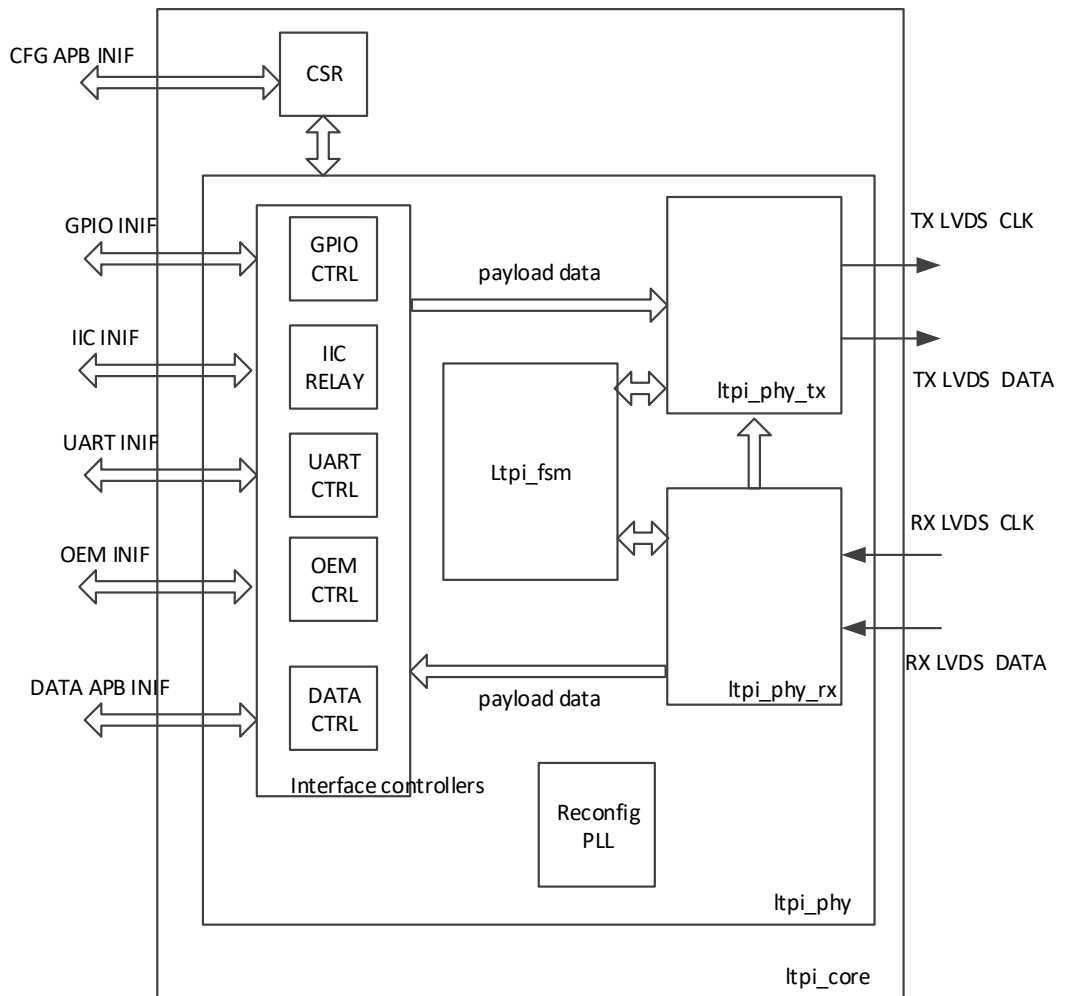
中，都有一个 LTPI 帧正在传输。在通过 LTPI 接口发送的每个帧中，分配给特定信道的比特数与专用于每个信道的 LTPI 带宽成正比。为了最大限度地减少对 LTPI 信道的延迟影响，假设 LTPI 帧是没有任何帧间间隙的。在无法避免帧间间隙的实现中，建议帧间间隙时间保持恒定，以最小化延迟变化并简化实现。

图 3-2 LTPI 各个通道传输示意图



3.1.2 IP 实现框图

图 3-3 DC-SCM LTPI IP 实现框图



DC-SCM LTPI IP 包含:

- **interface controllers** 模块: 接收来自于外部接口的信号, 通过内部的各个控制器/中继器来采集或者事务编码, 将采集数据和编码数据发送到 **ltpi_phy_tx** 模块中组帧; 接收来自于 LTPI 运行帧的内容, 将其还原为外部接口的信号时序。包含以下接口:
 - IIC/SMBus 中继器 - IIC/SMBus 接口
 - GPIO 控制器-GPIO 接口
 - UART 控制器-UART 接口
 - OEM 控制器-OEM 接口
 - 数据总线控制器-数据通道桥接至外部接口
- **ltpi_phy_tx** 模块功能
 - 每个 LTPI 信道都在 LTPI 帧内的专用位字段上编码
 - 计算每帧的 CRC, 并放与帧尾
 - 然后使用 8b/10b 编码对帧进行编码
 - 帧标志用帧逗号符号扩展 (K 码)
 - 10b 编码帧在 LVDS TX 链路上串行化
- **ltpi_phy_rx** 模块功能
 - LVDS 流量被捕获并反序列化到 10b 编码的 FIFO 缓冲器中
 - 在 10b 数据流中, 帧 K 码被捕获为 LTPI 帧的开始
 - 该帧被解码为 8b 编码
 - 再次检查帧的 CRC 校验和, 并将其分解为 LTPI 信道
 - 使用信道控制器/中继器在各个信道上重建信道状态
- **ltpi_fsm** 模块功能
 - 链路初始化以及运行流程状态机
- **Reconfig PLL** 模块功能
 - 根据 SCM 和 HPM 的速率协商结果, 通过改变 PLL 的参数来生成对应的 LVDS 速率。

3.2 端口列表

Gowin DC-SCM LTPI IP 核的 IO 端口如图 3-4 所示，详细介绍如表 3-1 所示。

图 3-4 Gowin DC-SCM LTPI IP 端口示意图

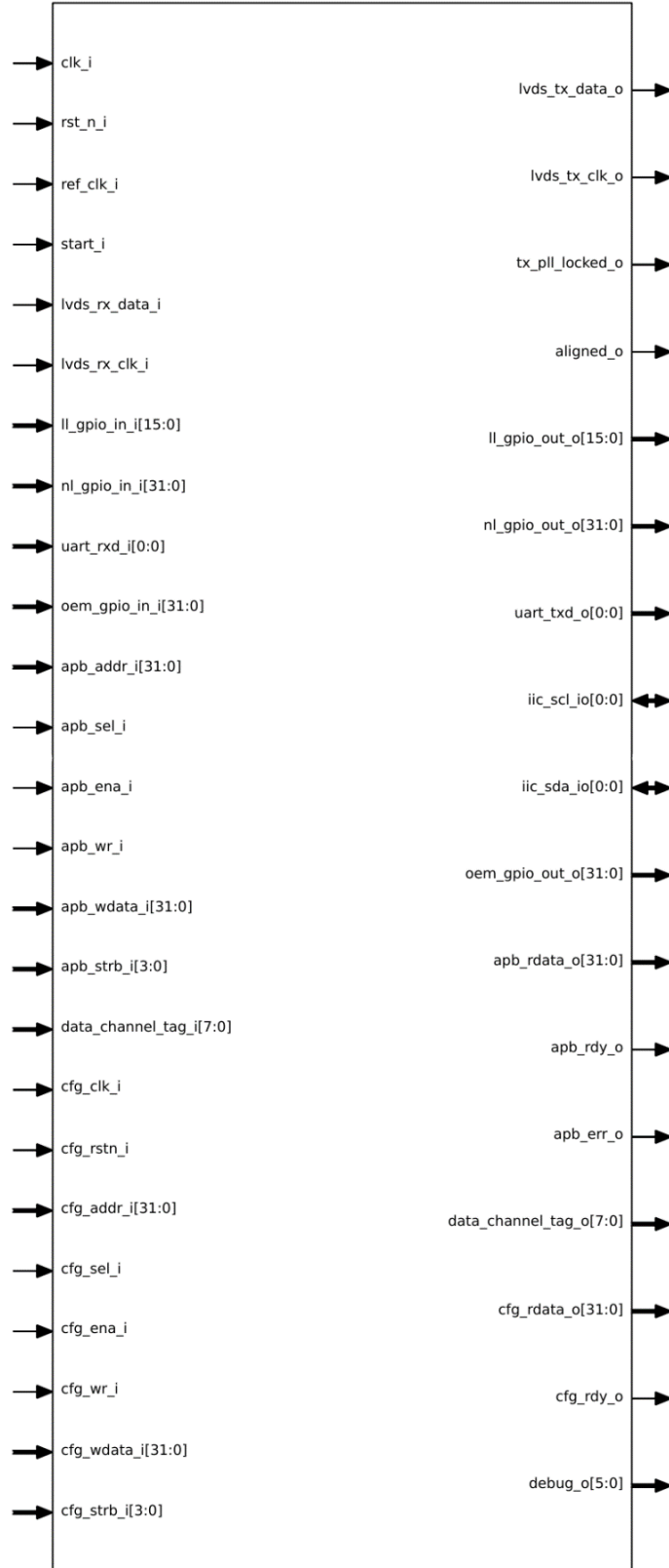


表 3-1 端口定义

名称	I/O	位宽	说明
clk_i	input	1	系统时钟，必须为 60MHz 输入
rst_n_i	input	1	系统复位
ref_clk_i	input	1	参考时钟，必须为 25MHz 输入
start_i	input	1	IP 核开始工作信号。 为高的时候开始 IP 开始运行
lvds_tx_data_o	output	1	LTPI 串行输出数据
lvds_tx_clk_o	output	1	LTPI 串行输出时钟
lvds_rx_data_i	input	1	LTPI 串行输入数据
lvds_rx_clk_i	input	1	LTPI 串行输入时钟
tx_pll_locked_o	output	1	内部 PLL 锁定标志，该 PLL 用于驱动 TX 差分时钟和驱动 TX 差分数据
aligned_o	output	1	字节对齐标志
低延迟 GPIO 通道（GPIO 接口使能后有效）			
ll_gpio_in_i	input	[N ^[1] -1 : 0]	低延迟 GPIO 输入管脚
ll_gpio_out_o	output	[N-1 : 0]	低延迟 GPIO 输出管脚
正常延迟 GPIO 通道（GPIO 接口使能后有效）			
nl_gpio_in_i	input	[M ^[2] -1 : 0]	正常延迟 GPIO 输入管脚
nl_gpio_out_o	output	[M-1 : 0]	正常延迟 GPIO 输出管脚
UART 通道（UART 接口使能后有效）			
uart_rxd_i	input	[L ^[3] -1 : 0]	UART 输入管脚
uart_cts_i	input	[L-1 : 0]	UART 流控管脚
uart_txd_o	output	[L-1 : 0]	UART 输出管脚
uart_rts_o	output	[L-1 : 0]	UART 流控管脚
IIC/SMBus 通道（IIC/SMBus 接口使能后有效）			
iic_scl_io	inout	[K ^[4] -1 : 0]	IIC/SMBus SCL 管脚
iic_sda_io	inout	[K-1 : 0]	IIC/SMBus SDA 管脚
OEM 通道（OEM 接口使能后有效）			
oem_gpio_in_i	input	[P ^[5] -1 : 0]	OEM 输入管脚
oem_gpio_out_o	output	[P-1 : 0]	OEM 输出管脚
Data 通道（Data 接口使能后有效）			
data_channel_tag_i	input	[7:0]	数据通道数据帧的 Tag 字段内容输入
data_channel_tag_o	output	[7:0]	数据通道数据帧的 Tag 字段内容输出
Data 通道 APB 从端（SCM），（Data 接口使能后有效）			
apb_addr_i	input	[31:0]	总线配置地址
apb_sel_i	input	1	总线配置选择
apb_ena_i	input	1	总线配置使能

名称	I/O	位宽	说明
apb_wr_i	input	1	总线方向信号 1: 写 0: 读
apb_rdata_o	output	[31:0]	读取的数据
apb_wdata_i	input	[31:0]	写入的数据
apb_rdy_o	output	1	准备好信号
apb_err_o	output	1	错误指示
apb_strb_i	input	[3 :0]	写选通信号
Data 通道 APB 主端 (HPM), (Data 接口使能后有效)			
apb_addr_o	output	[31:0]	总线配置地址
apb_sel_o	output	1	总线配置选择
apb_ena_o	output	1	总线配置使能
apb_wr_o	output	1	总线方向信号 1: 写 0: 读
apb_rdata_i	input	[31:0]	读取的数据
apb_wdata_o	output	[31:0]	写入的数据
apb_rdy_i	input	1	准备好信号
apb_err_i	input	1	错误指示
apb_strb_o	output	[3 :0]	写选通信号
CSR 接口			
cfg_clk_i	input	1	CSR 接口时钟, CSR 接口也是 APB 接口, 当数据通道启用时, 这个时钟被用于数据通道
cfg_rstn_i	input	1	配置复位
cfg_addr_i	input	[31:0]	总线配置地址
cfg_sel_i	input	1	总线配置选择
cfg_ena_i	input	1	总线配置使能
cfg_wr_i	input	1	总线方向信号 1: 写 0: 读
cfg_rdata_o	output	[31:0]	读取的数据
cfg_wdata_i	input	[31:0]	写入的数据
cfg_rdy_o	output	1	准备好信号
cfg_strb_i	input	[3 :0]	写选通信号, 内部固定为 4'hf
Debug 接口			
debug_o	output	[5:0]	-

注!

- ¹⁾表中 N 是 LL GPIO 数量, 通过 GUI 设置。
- ²⁾表中 M 是 NL GPIO 数量, 通过 GUI 设置。

- ^[3]表中 L 是 UART 数量，通过 GUI 设置。
- ^[4]表中 K 是 IIC/SMBus 数量，通过 GUI 设置。
- ^[6]表中 P 是 OEM 数量，通过 GUI 设置。

3.3 接口时序

3.3.1 配置接口时序

内部寄存器通过总线进行配置。图 3-5 和图 3-6 是总线配置写时序图和读时序图，该接口与 APB slave 时序一致，可以直接接 APB master 模块。

图 3-5 配置接口写时序图

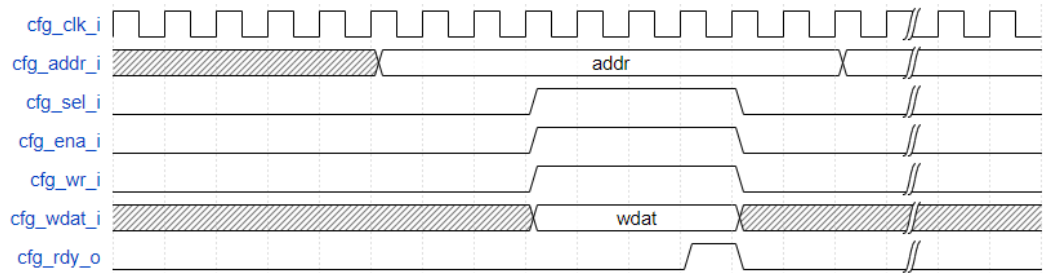
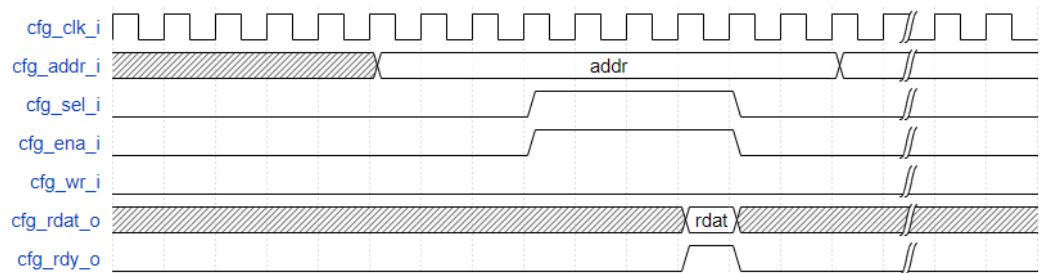


图 3-6 配置接口读时序图



3.3.2 数据接口时序

同配置接口时序，也是 APB 接口。

3.3.3 GPIO 接口时序

GPIO 接口直连即可。

3.3.4 UART 接口时序

标准 UART 接口时序。

3.3.5 IIC/SMBus 接口时序

标准 IIC/SMBus 总线时序。

3.4 寄存器描述

Gowin DC-SCM LTPI IP 中包含 LTPI 控制和状态寄存器（CSR），通过总线接口进行配置，地址以字节为单位。

表 3-2 Gowin DC-SCM LTPI IP 寄存器

地址偏移	比特	默认值	W/R	描述
0x00	31:24	—	—	Reserved
	19:16	—	R	本地 LTPI 链路状态 0: 链路检测阶段 1: 速率协商阶段 2: 通告阶段 3: 配置阶段（SCM）、接受阶段（HPM） 4: 运行状态 5-7: 保留
	15:12	—	R	对端 LTPI 链路状态 0: 链路检测阶段 1: 速率协商阶段 2: 通告阶段 3: 配置阶段（SCM）、接受阶段（HPM） 4: 运行状态 5-7: 保留
	11:8	—	R	LTPI 速度等级（LVDS 时钟频率） 0: 基频 X1 1: 基频 X2 2: 基频 X3 3: 基频 X4 4: 基频 X6 5: 基频 X8 6: 基频 X10 7: 基频 X12 8: 基频 X16 9: 基频 X24 10: 基频 X32 11: 基频 X40 12-15: 保留 基频为 25MHz
	7	—	R	DDR 模式 0: SDR 模式 1: DDR 模式
	6	—	—	保留
	5	—	R	链路配置/接受握手超时错误
	4	—	R	链路速率协商超时错误
	3	—	R	非法逗号符号错误

地址偏移	比特	默认值	W/R	描述
	2	–	R	帧 CRC 错误
	1	–	R	LTPI 链路丢失错误
	0	–	R	链路对齐状态 0: 链路没对齐 1: 链路对齐
0x04	31:24	–	–	Reserved
	23:8	GUI 配置	RW	本地链路速度能力, 详细信息见表 3-3
	7:4	GUI 配置	R	本地LTPI Major版本 使用 BCD 编码
	3:0	GUI 配置	R	本地LTPI Minor版本 使用 BCD 编码
0x08	31:24	–	–	Reserved
	23:8	–	R	对端链路速度能力, 见表 3-4
	7:4	–	R	对端LTPI Major版本 使用 BCD 编码
	3:0	–	R	对端LTPI Minor版本 使用 BCD 编码
0x0C	31:16	–	–	Reserved
	15:0	0x0	RW	LTPI 本地 Platform ID
0x10	31:16	–	–	Reserved
	15:0	–	R	LTPI 远端 Platform ID
0x14	31:0	GUI 配置	RW	本地能力通告帧内容低 32bit, 详细约定见表 3-4
0x18	31:0	GUI 配置	RW	本地能力通告帧内容高 32bit, 详细约定见表 3-4
0x1C	31:0	–	R	对端能力通告帧内容低 32bit, 详细约定见表 3-4
0x20	31:0	–	R	对端能力通告帧内容高 32bit, 详细约定见表 3-4
0x24	31:0	SCM:GUI 配置时默认与本地能力通告帧相同	SCM:RW HPM:R	SCM: LTPI 默认能力配置(低32bits) HPM: LTPI 接受并生效的配置(低 32bits) 详细约定见表 3-4
0x28	31:0	SCM:GUI 配置时默认与本地能力通告帧相同	SCM:RW HPM:R	SCM: LTPI 默认能力配置(高32bits) HPM: LTPI 接受并生效的配置(高 32bits) 详细约定见表 3-4
0x2C	31:0	0x0	RWC	LTPI 链路对齐错误计数器
0x30	31:0	0x0	RWC	LTPI 链路丢失错误计数器

地址偏移	比特	默认值	W/R	描述
0x34	31:0	0x0	RWC	LTPI CRC 错误计数器
0x38	31:0	0x0	RWC	非法 Comma 错误计数器
0x3C	31:0	0x0	RWC	链路速度协商超时错误计数器
0x40	31:0	0x0	RWC	SCM: 链路配置超时错误计数器 HPM: 链路接受超时错误计数器
0x44	31:24	0x0	RWC	配置帧 (Configure Frame) 接收计数器 (仅 HPM)
	23:16	0x0	RWC	速度帧 (Speed Frame) 接收计数器
	15:0	0x0	RWC	检测帧 (Detect Frame) 接收计数器
0x48	31:0	0x0	RWC	通告帧 (Advertise Frame) 接收计数器
0x4C	31:24	0x0	RWC	配置帧 (Configure Frame) 发送计数器 (仅 HPM)
	23:16	0x0	RWC	速度帧 (Speed Frame) 发送计数器
	15:0	0x0	RWC	检测帧 (Detect Frame) 发送计数器
0x50	31:0	0x0	RWC	运行帧 (Operational Frame) 接收计数器
0x54	31:0	0x0	RWC	运行帧 (Operational Frame) 发送计数器
0x80	31:12	–		保留
	10	0x0	RW	触发配置状态
	9	0x0	RW	自动跳转到配置状态 1: 自动跳转 0: 需要触发
	8	0x0	RW	数据通道复位 1: 表示复位 0: 表示工作
	7:2	0x0	RW	LTPI IIC 通道复位, 每个 bit 表示每个 IIC 通道 1: 表示复位 0: 表示工作
	1	0x0	RW	LTPI 链路重新训练请求 上升沿会进行重新训练请求
	0	0x0	RW	LTPI 链路软复位 上升沿会进行软复位

注!

R 表示读, W 表示写, WC 表示写清除

表 3-3 速度描述

字节	Bit 定义							
	7	6	5	4	3	2	1	0
0	基频 x12 300MHz	基频 x10 250MHz	基频 x8 200MHz	基频 x6 150MHz	基频 x4 100MHz	基频 x3 75MHz	基频 x2 50MHz	基频 x1 25MHz
1	0: SDR 1:DDR	保留	保留	保留	基频 x40 1GHz	基频 x32 800MHz	基频 x24 50MHz	基频 x16 400MHz

表 3-4 通告帧能力描述

字节	bit 定义							
	7	6	5	4	3	2	1	0
0	Supported Channels							
	Reserved			OEM channel	Data channel	UART channel	IIC channel	GPIO channel
1	GPIO Channel Capabilities							
	Number of NL GPIOs[7:0]							
2	Reserved						Number of NL GPIOs[9:8]	
3	IIC Channel Capabilities							
	Reserved	EchoSupport 1: support 0: no support	IIC5 En 1: enable 0: disable	IIC4 En 1: enable 0: disable	IIC3 En 1: enable 0: disable	IIC2 En 1: enable 0: disable	IIC1 En 1: enable 0: disable	IIC0 En 1: enable 0: disable
4	Reserved		IIC5Speed 1:400K 0:100K	IIC4Speed 1:400K 0:100K	IIC3Speed 1:400K 0:100K	IIC2Speed 1:400K 0:100K	IIC1Speed 1:400K 0:100K	IIC0Speed 1:400K 0:100K
5	IIC Channel Capabilities							
	Reserved	UART1 En 1: enable 0: disable	UART0 En 1: enable 0: disable	Flow control	Max Baud Rate 定义见表 3-5			
6	OEM Capabilities							
7								

表 3-5 串口通道最大波特率定义表

Baud rate(bps)	Max baud rate encoding
300	0h00
600	0h01
1200	0h02
1800	0h03
2400	0h04
4800	0h05
9600	0h06
19200	0h07
38400	0h08
57600	0h09
115200	0h0A
230400	0h0B
460800	0h0C
576000	0h0D
921600	0h0E
Reserved	0h0F

寄存器设置举例如下：

假如本端速度支持 X1、X2、X3、X4、X6、DDR；通道支持 GPIO、IIC/SMBus、UART、OEM、Data 通道，且

- LLGPIO 有 16 个
- NLGPIO 有 32 个
- IIC 有 1 个，100K 速率，支持 Echo
- OEM 的 GPIO 有 32 个
- Data 通道支持
- UART 有 1 个，不支持流控，支持 115200bps 的最大波特率

则寄存器设定为：0x04[23:8] = 16h801f, 0x14[31:0] = 32h4100201f, 0x18[31:0] = 32h00002A00。

4 设计说明

4.1 线速率

DC-SCM 2.0 LTPI 规范定义了 12 种 LVDS 时钟，包括 X1(25MHz)、X2(50MHz)、X3(75MHz)、X4(100MHz)、X6(150MHz)、X8(200MHz)、X10(250MHz)、X12(300MHz)、X16(400MHz)、X24(600MHz)、X32(800MHz)、X40(1000MHz)。模式有 SDR 和 DDR 模式，如果是 SDR 模式，那么线速率就和时钟频率相同，例如工作在 X6(150MHz)下的 SDR 模式，那么线速率为 150Mbps，工作在 X6(150MHz)下的 DDR 模式，那么线速率为 300Mbps。

4.2 时钟

4.2.1 系统时钟

必须为固定为 60MHz。

4.2.2 参考时钟

必须固定为 25MHz，且参考时钟需要与系统时钟同源。

4.2.3 配置时钟(cfg clk)

该内核通过配置接口进行寄存器的读写，同时如果使能了数据通道，则该时钟也作用于数据通道的 APB 接口。该接口的时钟是独立于系统时钟或参考时钟的时钟，此时钟与系统时钟或参考时钟之间没有依赖关系。

4.2.4 LVDS 时钟

IP 核内部存在动态可配置 PLL 模块，该 PLL 模块默认生成两个相位差 90°（此相位差可以通过寄存器调整）的时钟 LINKCLK 和 LINKCLK90，根据速率协商的结果，进行 LVDS 频率动态配置，用于驱动 TX LVDS CLK 和 TX LVDS Data 信号的产生

4.2.5 RX/TX DIV5 时钟

IP 核内部存在串并转换原语 ISER10/OSER10，并行数据的时钟为 LVDS CLK 的 5 分频。在发送部分，TXDIV5 由时钟 LINKCLK 通过原语 CLKDIV 进行 5 分频；在接收部分，RXDIV5 由接收的 LVDS CLK 通过原语 CLKDIV

进行 5 分频。

4.2.6 时钟结构

图 4-1 典型时钟方案

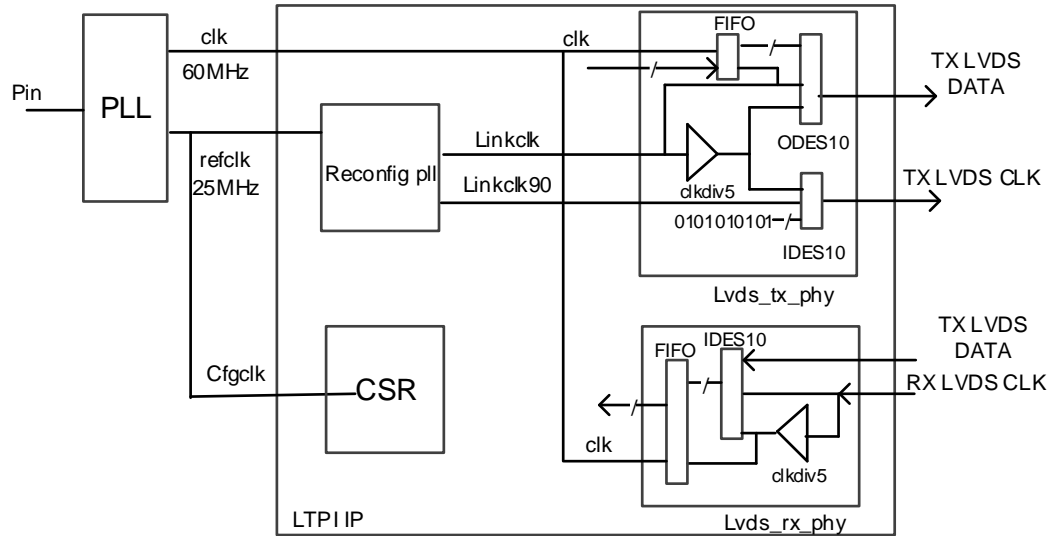


图 4-1 显示了一种时钟方案，基于同一个时钟源生成了系统时钟 clk，为 60MHz；参考时钟 refclk，为 25MHz；同时配置时钟 cfgclk 也可与参考时钟相同。

4.3 外部通道接口处理

LTP1 使用以下通用方法来采集和传输各种 LTP1 通道：

- **采样：** I/O 状态由 LTP1 进行采样，采样在 LTP1 帧中进行隧道传输。
- **事件/状态检测：** 根据接口状态，为通道定义一组事件/状态，并在 LTP1 帧中进行隧道传输。
- **随机存取读/写请求：** 外部接口触发随机存取内存映射通道。

LTP1 支持的接口也可以根据给定接口的 RX 和 TX 方向的同步来分类，即接口的隧道传输是否需要 TX 和 RX 方向在彼此之间同步：

- **异步：** 对于给定的信道和接口/链路，LTP1 TX 和 RX 路径是独立的。
- **同步：** 对于给定的信道和接口/链路，LTP1 TX 和 RX 路径需要同步，以允许接口在隧道传输后正确工作。

表 4-1 外部通道接口处理

通道类型	采集方式	TX 和 RX 同步方式	通道特性
GPIO	采样	异步	直接采集信号电平 低延迟 GPIO 每个 LPTI 帧都更新 正常延迟 GPIO 可以设置数量更多,但由于 LPTI 帧给定的正常延迟 GPIO 的位置有限,因此可能会隔多个帧才更新一次,同时 LPTI 帧还会有一个帧 ID

通道类型	采集方式	TX 和 RX 同步方式	通道特性
UART			标志用于识别正常延迟 GPIO 的位置
			直接采集信号电平 UART 信号进行过采样 每个 LTPI 帧包含多个采样值
IIC/SMBus	事务/状态检测	同步	IIC/SMBus 由于需要来回交互，因此 IIC/SMBus 状态被编码成 LTPI 事务通过 LTPI 传输 在 LTPI 事务发送之后，需要等待另一端的 LTPI 接口的事务回应，因此需要使用 IIC/SMBus 时钟扩展功能来等待同步完成
Data Bus	随机访问		数据总线传输如数据读和写被编码为 LTPI 事务 总线操作完成指示用于同步对数据总线的访问

4.3.1 GPIO 通道

对于 GPIO 通道，GPIO 逻辑状态（高和低）直接编码在 LTPI 帧位上。低延迟（LL）GPIO 在每个 LTPI 帧被刷新，并且在发送的每个帧中具有用于每个 GPIO 的专用位。每 N 个帧刷新一次正常延迟（NL）GPIO，其中 N 定义为：

$$N = \left(\frac{\text{Total Number of NL GPIOs tunneled over LTPI}}{\text{Number of NL GPIO bits per LTPI IO Frame}} \right)$$

N = 需要传递的 GPIO 总数/每个 IO 帧的 NL GPIO 比特数

为了解码在给定帧中隧道传输的 NL GPIO，使用在每个帧中递增的帧计数器来解码来自 LTPI 帧的 NL GPIO 索引。LTPI 帧中 LTPI GPIO 通道的编码如下表所示，其中 LL 和 NL GPIO 分别定义在 1 个字节上。对于 LL GPIO，一个字节中的每一位都被定义为数字为 0 到 7 的 GPIO。对于 NL GPIO，为了解码每个位位置 0 到 7 的 GPIO 编号，需要添加偏移 X，其中 X 定义为：

$$X = (\text{Frame Counter} \% N) \\ * (\text{Number of NL GPIO bits per LTPI Frame})$$

表 4-2 GPIO 字段配置

名称	Bit 定义							
	7	6	5	4	3	2	1	0
低延迟 GPIO (LLGPIO)	LL GPIO7 0: 低 1: 高	LL GPIO6 0: 低 1: 高	LL GPIO5 0: 低 1: 高	LL GPIO4 0: 低 1: 高	LL GPIO3 0: 低 1: 高	LL GPIO2 0: 低 1: 高	LL GPIO1 0: 低 1: 高	LL GPIO0 0: 低 1: 高
正常延迟 GPIO (NLGPIO)	NL GPIO [X+7] 0: 低 1: 高	NL GPIO [X+6] 0: 低 1: 高	NL GPIO [X+5] 0: 低 1: 高	NL GPIO [X+4] 0: 低 1: 高	NL GPIO [X+3] 0: 低 1: 高	NL GPIO [X+2] 0: 低 1: 高	NL GPIO [X+1] 0: 低 1: 高	NL GPIO [X] 0: 低 1: 高

默认 I/O 帧为低延迟 GPIO 分配 2 个字节，为正常延迟 GPIO 配置 2 个字节。这意味着每个 IO 帧中可以发送 16 个 NL GPIO，并且传输所有 NL GPIO 需要 N 个帧。如果某个实现需要在单个 IO 帧中定义更多 LL 或 NL GPIO，则可以使用 OEM 字段来扩展默认 I/O 帧。

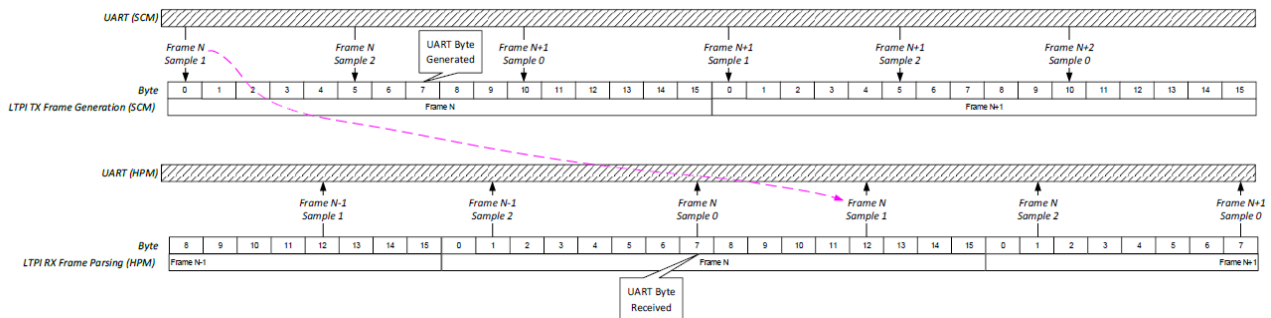
正常延迟 GPIO 的总数在“NL GPIO 数量”字段的能力帧中定义。能力在 LTPI 训练阶段进行交换。通过这种方式，SCM 中的 LTPI 和 HPM 中的 LTPI 能够基于上面定义的帧计数器和 N 参数来正确地解码给定帧中的 NL GPIO 的索引。

4.3.2 UART 通道

UART 通道的工作原理类似于 LL GPIO 通道。不同之处在于，对 UART 发送数据 (TXD) 和接收数据 (RXD) 线的采样频率是 LL GPIO 采样频率的 3 倍。

UART 采样数据如何通过 LTPI 帧从 TX 端到 RX 端，并且在接收端恢复 UART 数据，如下图 4-2 所示。如本例所示，当生成字节号 7 并通过 LTPI 传输时，UART 样本需要准备就绪。这意味着在此之前，需要收集 3 个样本，假设样本随时间均匀分布。对于帧 N，当发送前一帧 (N-1) 的字节 10 时，实际采集样本 0。当通过 LTPI 发送字节 0 时采集样本 1，当生成并发送带有 UART 样本的字节 7 时采集样本 2。通过这种方式，在帧之间保持样本随时间的均匀分布。在接收器侧，当接收到的样本用于在 UART 接口上生成信号时，需要保持相同的分布。当接收到帧 N 的字节 7 时，样本 0 用于在 UART 接口上生成信号，当接收到字节 12 时，从帧 N 生成下一个样本 1。最后一个采样是在下一帧 (N+1) 的字节 1 上生成的。

图 4-2 UART 采样与恢复示意图



UART 信道与 RXD 和 TXD 信号一起支持 RTS 和 CTS 支持的流控制信号隧道。流控制信号不是过采样的，而是类似于低延迟 GPIO 在每个 LTPI 帧中编码。表 4-3 显示了 UART 通道的编码。每个信道被编码在用于 TX 的 4 比特和用于 RX 方向的 4 比特上。这些编码将通过 LTPI 接口单独进行隧道传输。在该示例中，从 SCM 的角度考虑方向，因此 UART0/1 TX 编码适用于从 SCM 到 HPM 的隧道传输的 LTPI 帧。UART0/1 RX 编码适用于从 HPM 通过隧道传输到 SCM 的 LTPI 帧。

表 4-3 UART 字段配置

名称	Bit 定义							
	7	6	5	4	3	2	1	0
UART0/1 TX 编码	UART1				UART0			
	RTS	TXD[2]	TXD[1]	TXD[0]	RTS	TXD[2]	TXD[1]	TXD[0]
	0: 低 1: 高	0: 低 1: 高	0: 低 1: 高	0: 低 1: 高	0: 低 1: 高	0: 低 1: 高	0: 低 1: 高	0: 低 1: 高
UART0/1 RX 编码	CTS	RXD[2]	RXD[1]	RXD[0]	CTS	RXD[2]	RXD[1]	RXD[0]
	0: 低 1: 高	0: 低 1: 高	0: 低 1: 高	0: 低 1: 高	0: 低 1: 高	0: 低 1: 高	0: 低 1: 高	0: 低 1: 高

4.3.3 IIC/SMBus 通道

IIC/SMBus SCL 和 SDA 信号是双向的，并且可以在 IIC/SMBus 事件期间由控制器或目标设备主动驱动。因此，SCL 和 SDA 信号不能像 GPIO 或 UART 信号那样直接通过 LTPI 进行隧道传输。在 LTPI 中，IIC/SMBus 通道实现了基于事件的模型，其中编码的 IIC/SMBus 事件/状态在 LTPI 的一侧被捕获并在另一侧被恢复。当 IIC/SMBus 事件由控制器生成时（例如，START 条件），控制器的本地 SCL 线由 LTPI IP 来进行拉伸，直到接收到远端事件。当接收到远端事件时，它在控制器本地总线上恢复，然后 SCL 从拉伸状态释放。

表 4-4 IIC/SMBus 通道事务定义

总线事件	方向	描述	事务编码(4bit)
Idle	双向	空闲状态	4b0000
Start	控制端 (Controller) 到目标端 (Target)	控制端检测到开始事件	4b0001
Start Received	目标端到控制端	开始事件在目标端恢复	4b0010
Stop	控制端到目标端	控制端检测到停止事件	4b0011
Stop Received	目标端到控制端	停止事件在目标端恢复	4b0100
Data Received	双向	SDA 值在对端恢复	4b0101
Data0	双向	发送 SDA 值 0	4b0110
Data1	双向	发送 SDA 值 1	4b0111
Start Echo	目标端到控制端	开始事件被目标端正确接收的反馈事务	4b1000
Stop Echo	目标端到控制端	停止事件被目标端正确接收的反馈事务	4b1001
Data0 Echo	双向	发送的 SDA 值 0 正确接收	4b1010
Data1 Echo	双向	发送的 SDA 值 1 正确接收	4b1011
Data Received Echo	双向	表示 Data Received 事件被正确接收。它用于区分字节	4b1100

总线事件	方向	描述	事务编码(4bit)
		传输过程中连续的 Data 0/1 事件。	
Reserved	保留	保留	4b1101-4b1111

每个 IIC/SMBus 总线相当于 4 位有效负载数据，如表 4-4 所示。在表 4-5 的示例中，对于 6 个 IIC/SMBus 总线，每帧相应地为 IIC/SMBus 通道分配了 3 个字节的的有效负载。

表 4-5 IIC/SMBus 字段配置

IIC/SMBus 载荷字节	Bit 定义							
	7	6	5	4	3	2	1	0
0	IIC/SMBus BUS 1				IIC/SMBus BUS 0			
	Event[3]	Event[2]	Event[1]	Event[0]	Event[3]	Event[2]	Event[1]	Event[0]
1	IIC/SMBus BUS 3				IIC/SMBus BUS 2			
	Event[3]	Event[2]	Event[1]	Event[0]	Event[3]	Event[2]	Event[1]	Event[0]
2	IIC/SMBus BUS 5				IIC/SMBus BUS 4			
	Event[3]	Event[2]	Event[1]	Event[0]	Event[3]	Event[2]	Event[1]	Event[0]

图 4-3 IIC/SMBus 总线事务交换

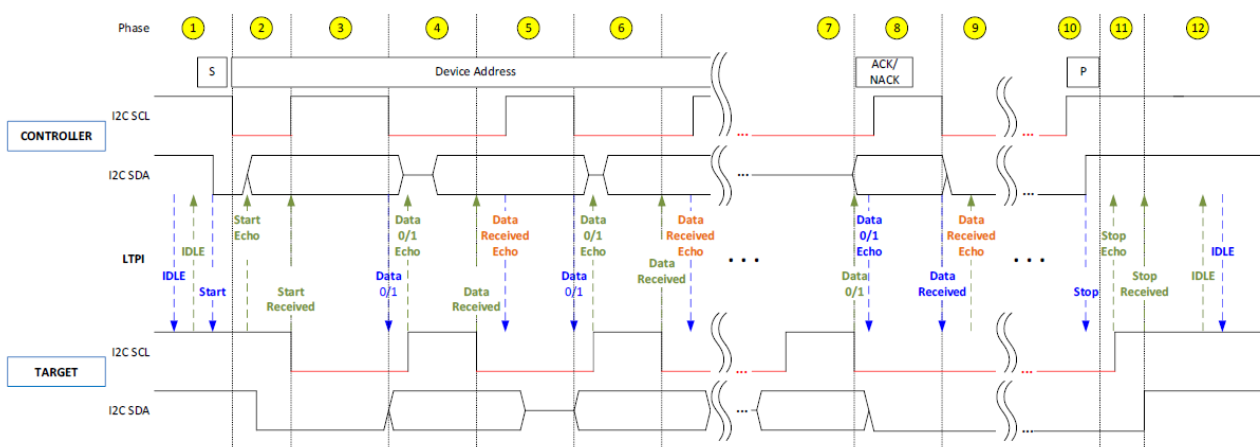


图 4-3 显示了 IIC/SMBus 总线事件交换。事件由控制器发起。一旦在 LTPI 中检测到事件请求，通信方就会通过 LTPI 将编码的事件 echo 发送回请求方来确认接收，并且当事件成功时，它会发送事件接收数据包。当任一方等待通信方的反馈时，本地 SCL 总线上会进行时钟拉伸，以防止当前事务正在进行时出现新事务。编码数据包持续传输，直到生成新事件。SCL 接口中的粗线表示在 SCL 总线上完成的时钟延伸。

表 4-6 IIC/SMBus 事务流程

序号	控制端	目标端	IIC/SMBus 事件	LTPI 事件方向
1	IP 检测 SDA 下降沿和 SCL 高电平, 开始向 LTPI 接口发送 Start 事件; IP 触发本地 SCL 上的 SCL 延伸并等待接收到 START Received.	总线空闲状态	Start	控制端到目标端
2	时钟拉伸	从 LTPI 收到 Start 事件, 发送 start echo 事件进行反馈, 在本地 SCL 和 SDA 总线生成开始条件时序; 发送 Start Received 事件到 LTPI; 触发 SCL 拉伸并等待下一个事件	Start Echo Start Received	目标端到控制端
3	从 LTPI 接收到了 start Received 事件后停止本地总线的时钟拉升; 控制器开始发送数据 bit; IP 检测到本地总线的 SCL 上升沿和 SDA 数据; IP 检测 SCL 下降沿, 并发送 Data0/1 事务到 LTPI. IP 触发本地 SCL 上的 SCL 延伸并等待接收到 Data Received 事件	时钟拉伸	Data0/1	控制端到目标端
4	时钟拉伸	从 LTPI 收到 Data0/1 事件, 发送 Data echo 事件进行反馈, 释放 SCL 并在本地 SCL 和 SDA 总线发送数据 0/1; 在 SCL 高电平保持时间之后 SCL 拉低; 发送 Data Received 事件到 LTPI; 触发 SCL 拉伸并等待下一个事件	Data Echo Data Received	目标端到控制端
5	从 LTPI 接收到了 Data Received 事件后停止本地总线的时钟拉升; 发送 Data Received Echo 事件到 LTPI 接口, 控制器开始发送数据 bit; IP 检测到本地总线的 SCL 上升沿和 SDA 数据; IP 检测 SCL 下降沿, 并发送 Data0/1 事务到 LTPI. IP 触发本地 SCL 上的 SCL 延伸并等待接收到 Data Received 事件	时钟拉伸	Data Received Echo Data 0/1	控制端到目标端
6	时钟拉伸	首先从 LTPI 收到 Data Received Echo 事件, 表明上一个 bit 发送完成, 接下来接收一个新的 Data 0/1 事件; 发送 Data echo 事件进行反馈, 释放 SCL 并在本地 SCL 和 SDA 总线发送数据	Data Echo Data Received	

序号	控制端	目标端	IIC/SMBus 事件	LTPI 事件方向
		0/1; 在 SCL 高电平保持时间之后 SCL 拉低; 发送 Data Received 事件到 LTPI; 触发 SCL 拉伸并等待下一个事件		
...
数据事务方向的改变, 即本例中的 ACK/NACK, 但对于读事务中的数据位来说是相同的				
...
7	时钟拉伸	目的端将数据 bit 置于 SDA; IP 检测到本地总线上的 SCL 上升沿且 SDA 数据; IP 检测到 SCL 下降沿, 且发送 Data0/1 事件到 LTPI; 触发 SCL 拉伸并等待下一个事件	Data 0/1	目标端到控制端
8	从 LTPI 收到 Data0/1 事件, 发送 Data echo 事件进行反馈, 释放 SCL 并在本地 SCL 和 SDA 总线发送数据 0/1; 在 SCL 高电平保持时间之后 SCL 拉低; 发送 Data Received 事件到 LTPI; 触发 SCL 拉伸并等待下一个事件	时钟拉伸	Data Echo Data Received	控制端到目标端
9	时钟拉伸	从 LTPI 接收到了 Data Received 事件; 开始发送 Data Received Echo 事件	Data Received Echo	目标端到控制端
...
10	当本地总线上的 SCL 为高电平时 (本地总线上的停止条件), IP 检测到 SDA 上升沿。IP 向 LTPI 发送 STOP 事件。	时钟拉伸	Stop	控制端到目标端
11	等待 Stop Echo 和 Stop Received 事件; IP 等待新的 IIC 起始条件, 如果检测到新的 START 条件, 则应将其推迟 (使用时钟拉伸), 直到收到先前 STOP 条件的 Stop Received 事件。	收到 Stop 事件; 发送 Stop Echo 事件并开始在本本地总线 SCL 和 SDA 上生成停止条件时序信号; 通过 LTPI 接口发送 Stop Received; 触发总线状态为空闲。	Stop Echo Stop Received	目标端到控制端
12	总线空闲状态	总线空闲状态	Idle	N/A

4.3.4 OEM 通道

OEM 通道将类似于与 LLGPIO，与 LLGPIO 不同的地方在于，OEM 通道将在所有非数据通道帧传输时更新，而 LLGPIO 在数据通道帧传输也会更新。

4.3.5 数据通道

数据通道允许将标准化读取和写入请求传送到可寻址内存空间。DC-SCM LTPI 软 IP 支持数据通道的 APB 接口，并且在启用时可以按需访问。在 LTPI 中，数据通道实现了基于事件的模型，其中编码的数据事件/状态在 LTPI 的一侧被捕获并在另一侧被恢复。

表 4-7 数据通道事务定义

指令	参数	描述	8bit 事件编码
读请求	地址	读指定地址的请求指令	0x00
写请求	地址和数据	写指定地址和数据的请求指令	0x01
读完成	地址和数据	读操作完成指令	0x02
写完成	地址	写操作完成指令	0x03
CRC 错误	无	接收的数据帧 CRC 错误，并且操作不能完成	0x04
保留	保留	保留	0x05~0xFF

根据指令请求，每个数据通道字节被映射到数据帧中的每个有效负载字节。

表 4-8 数据通道字段配置

名称	比特编码							
	7	6	5	4	3	2	1	0
读操作	Command = 8'h00							
	地址字节 3							
	地址字节 2							
	地址字节 1							
	地址字节 0							
	保留					字节 3 使能	字节 2 使能	字节 1 使能
读操作完成	Command = 8'h02							
	地址字节 3							
	地址字节 2							
	地址字节 1							
	地址字节 0							
	读操作状态 0:成功, 1: 非法访问, 2-F 保留					字节 3 使能	字节 2 使能	字节 1 使能

名称	比特编码				
	数据字节 3				
	数据字节 2				
	数据字节 1				
	数据字节 0				
写操作	Command = 8'h01				
	地址字节 3				
	地址字节 2				
	地址字节 1				
	地址字节 0				
	保留	字节 3 使能	字节 2 使能	字节 1 使能	字节 0 使能
	数据字节 3				
	数据字节 2				
	数据字节 1				
	数据字节 0				
写操作完成	Command = 8'h01				
	地址字节 3				
	地址字节 2				
	地址字节 1				
	地址字节 0				
	写操作状态 0:成功, 1: 非法访问, 2-F 保留	保留			

4.4 LTPI 帧格式

4.4.1 帧格式

IP 中所设计的帧格式全部与 DC-SCM 2.0 LTPI 协议中规定的一致，所有类型的帧格式都是固定长度的，一共 16 个字节，其格式如表 4-9 所示。

表 4-9 帧格式说明

字节数	说明
0	帧起始, K 码
1	帧类型
2-14	帧数据
15	CRC8 校验, 帧尾

4.4.2 具体帧定义

1. 链路训练阶段

该阶段有两种子帧类型，如表 4-10 所示。

表 4-10 链路训练阶段帧类型

帧类型	逗号符号	帧类型值
Link Detect	K28.5	0x00
Link Speed		0x01

a. Link Detect Frame

首先双方一开始发送的是 Link Detect Frame 帧，具体定义见表 4-11。

表 4-11 Link Detect 帧定义

帧内字节	Bit 内容								描述
	7	6	5	4	3	2	1	0	
0	逗号符号								根据上表定义
1	帧类型								根据上表定义
2	LTPI Major 版本				LTPI Minor 版本				BCD 编码 [Major].[Minor]
3	速度能力								格式见表 3-3
4									
5-14	保留								保留
15	CRC								CRC8 checksum

b. Link Speed Frame

Link Speed Frame 帧格式与 Link Detect Frame 帧格式完全相同，只是 Link Speed Frame 帧中 Speed Select 是已经协商后的速率，如表 4-12 所示。

表 4-12 Link Speed 帧定义

帧内字节	Bit 内容								描述
	7	6	5	4	3	2	1	0	
0	逗号符号								根据上表定义
1	帧类型								根据上表定义
2	LTPI Major 版本				LTPI Minor 版本				BCD 编码 [Major].[Minor]
3	协商后速度								格式见表 3-3
4									
5-14	保留								保留
15	CRC								CRC8 checksum

2. 链路配置阶段

该阶段有三种子帧类型，如表 4-13 所示。

表 4-13 链路配置阶段帧类型

帧类型	逗号符号	帧类型值
Advertise	K28.6	0x00
Configure		0x01
Accept		0x02

a. LTPI Advertise Frame

速率协商完成之后，SCM 和 HPM 都会发送 Advertise 帧，互相通告能力，具体定义如表 4-14。

表 4-14 Advertise 帧定义

帧内字节	Bit 内容								描述
	7	6	5	4	3	2	1	0	
0	逗号符号								根据上表定义
1	帧类型								根据上表定义
2	Platform 类型								-
3									
4	能力类型 0x00 默认能力类型 0x01-0x80 保留 0x81-0xFF OEM 自定义								能力类型允许自定义 LTPI 能力
5	LTPI 能力								格式见表 3-4
6									
7									
8									
9									
10									
11									
12									
13-14	保留								保留
15	CRC								CRC8 checksum

互相通告能力之后，接下来，SCM 会发送 Configure 帧。Configure 帧与 Advertise 帧格式一致，不同在于，SCM 会将 SCM 预设的能力或者是 BMC 实时设置的能力放置于请求 LTPI 能力字段。

b. LTPI Configure Frame

表 4-15 Configure 帧定义

帧内字节	Bit 内容								描述
	7	6	5	4	3	2	1	0	
0	逗号符号								根据上表定义
1	帧类型								根据上表定义

帧内字节	Bit 内容	描述
2	Platform 类型	-
3		
4	能力类型 0x00 默认能力类型 0x01-0x80 保留 0x81-0xFF OEM 自定义	能力类型允许自定义 LTPI 能力
5	请求 LTPI 能力	格式见表 3-4
6		
7		
8		
9		
10		
11		
12		
13-14	保留	保留
15	CRC	CRC8 checksum

c. LTPI Accept Frame

HPM 接收到了 Configure 帧之后，会与本地能力进行对比，如果本地能力包含 Configure 帧内所描述的能力，则发送 Accept 帧。Accept 帧与 Configure 帧格式一致。接受 LTPI 能力和请求 LTPI 能力部分内容一致。

表 4-16 Accept 帧定义

帧内字节	Bit 内容								描述
	7	6	5	4	3	2	1	0	
0	逗号符号								根据上表定义
1	帧类型								根据上表定义
2	Platform 类型								-
3									
4	能力类型 0x00 默认能力类型 0x01-0x80 保留 0x81-0xFF OEM 自定义								能力类型允许自定义 LTPI 能力
5	接受 LTPI 能力								格式见表 3-4
6									
7									
8									
9									
10									
11									
12									

帧内字节	Bit 内容	描述
13-14	保留	保留
15	CRC	CRC8 checksum

SCM 收到 Accept 帧之后，开始进入运行阶段，发送 Operation 帧，而 HPM 收到 SCM 发来的 Operation 帧后进入运行阶段。

3. 运行阶段

当 SCM 和 HPM 都进入了 operation 状态时，开始发送运行帧。

运行帧有三种，一种是默认 I/O 帧，一种是默认数据帧，还有 OEM 自定义帧，如表 4-17 所示。当前的 IP 只支持默认 I/O 帧和默认数据帧。

表 4-17 运行阶段帧类型

帧类型	逗号符号	帧类型值
Default I/O	K28.7	0x00
Default Data		0x01
Reserved		0x02-0x0F
OEM Defined		0x10-0xFF

当没有使能 Data 通道的时候，则只有 Default I/O 帧，LTPI Default I/O Frame。

a. LTPI Default I/O Frame

该帧传输 GPIO/UART/IIC/OEM 等通道的信息，具体定义如表 4-18 所示。

表 4-18 Default I/O 帧定义

帧内字节	Bit 内容								描述
	7	6	5	4	3	2	1	0	
0	逗号符号								根据上表定义
1	帧类型								根据上表定义
2	帧计数器								帧计数器用于解码 NL GPIO 的位置
3	低延迟 GPIO0								定义见表 4-2
4	低延迟 GPIO1								定义见表 4-2
5	正常延迟 GPIO0								定义见表 4-2
6	正常延迟 GPIO1								定义见表 4-2
7	UART0 和 UART1								定义见表 4-3
8	IIC0 和 IIC1								定义见表 4-5
9	IIC2 和 IIC3								定义见表 4-5
10	IIC4 和 IIC5								定义见表 4-5
11-14	OEM 保留								OEM 保留
15	CRC								CRC8 checksum

b. LTPI Default Data Frame

该帧用于传送总线数据，具体定义如表 4-19 所示。

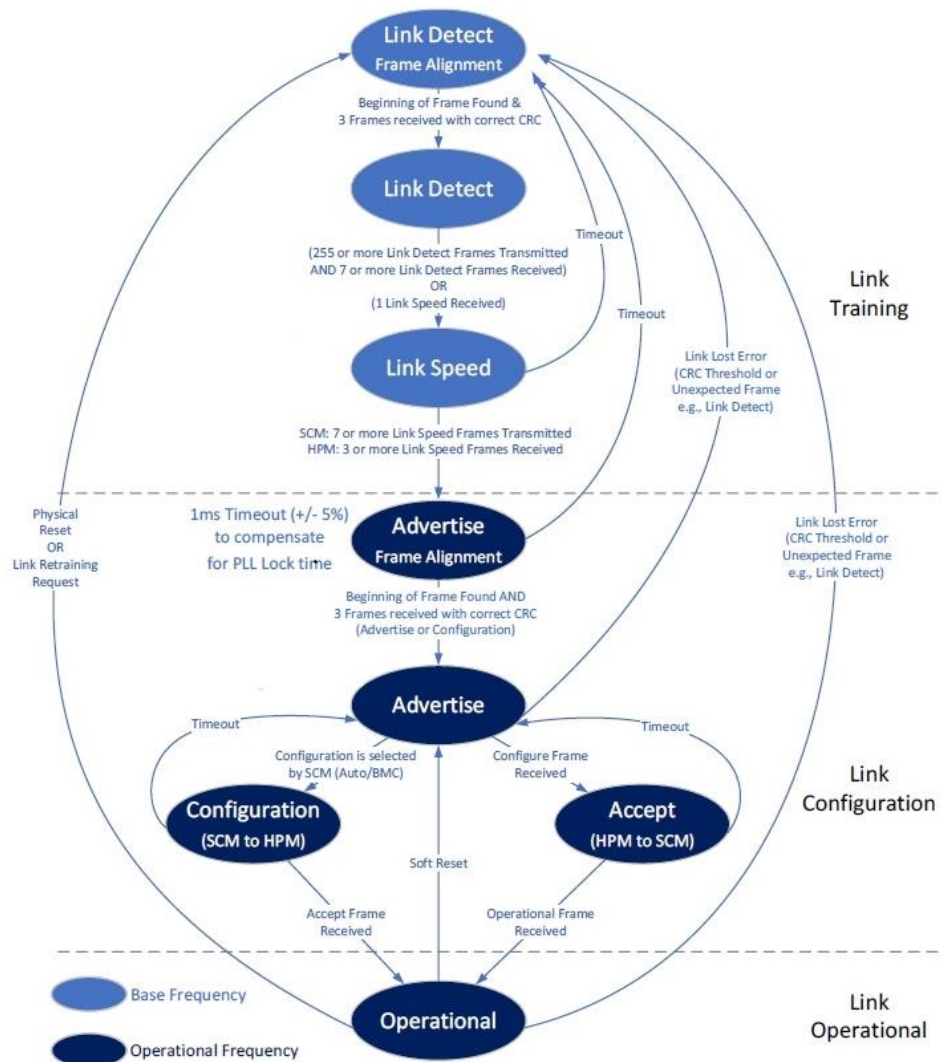
表 4-19 Default Data 帧定义

帧内字节	Bit 内容								描述
	7	6	5	4	3	2	1	0	
0	逗号符号								根据上表定义
1	帧类型								根据上表定义
2	低延迟 GPIO0								定义见表 4-2
3	低延迟 GPIO1								定义见表 4-2
4	Tag								
5-14	数据通道载荷								定义见表 4-8
15	CRC								CRC8 checksum

4.5 链路初始化以及运行流程

4.5.1 状态图

图 4-4 LTPI 链路状态图



LTPI 链路训练和切换到操作模式被定义为状态机，可分为 3 个主要阶段：

- 链路训练：链路初始化的初始状态
- 链路配置：LTPI 特性的配置
 - 交换 LTPI 特性功能
 - 启用 LTPI 的选定操作配置
 - 切换到操作模式
- 链路运行：说明链路在 LTPI 生命周期的大部分时间内处于可用状态
整个链路的操作都是按照这个状态机的流程来进行运作的。

4.5.2 链路训练

SCM 和 HPM 退出复位后，链路训练阶段开始。它在单数据速率 SDR 模式下以 25MHz 的基本频率执行。链接训练阶段的主要目的是：

- 达到链路直流平衡条件
- 交换链路速度功能
- 切换到操作链路频率

链接训练阶段基于 2 个状态：

1. 链路检测状态（Link Detect）
2. 链路速度协商状态（Link Speed）

在链路检测状态下，SCM 和 HPM 都开始发送链路检测帧。帧在两个方向上传输。主要目的是建立链路直流平衡并表明支持的工作频率。此阶段可分为 2 个部分：

- 链路检测-帧对齐：在这部分中，SerDes 逻辑尝试通过识别逗号符号并处理帧以验证 CRC 来锁定帧的开头。为了进入下一部分，至少需要接收 3 个具有正确 CRC 的帧。
- 链路检测：这是链路检测的主要部分，其中帧用于解释另一端的操作速度能力。在这一部分中使用了 TX 和 RX 计数器。由于 SCM 和 HPM 在实现 DC 平衡寻找帧开头方面存在时序差异，因此预计 SCM 和 HPM TX/RX 计数器将不对齐。

SCM 和 HPM 均应：

- 在 TX 链路上发送至少 255 个链路检测帧
- 在 RX 链路上接收至少 7 个具有正确 CRC 校验和的连续帧

但如果一方较快地移动到下一个阶段并开始发送链路速度帧，则“较慢”的一方应立即移动到下一个状态。

在链路速度协商状态下，SCM 和 HPM 都开始发送链路速度帧。SCM 应根据接收到的链路检测功能及其自身的功能来选取最高的共同目标频率。

在某些情况下，例如系统集成、系统调试或恢复模式，可能需要将工作频率降低到最大能力以下。为此，LTPI 可以向 BMC 或 SCM 上的其他控制器公开设置，以允许修改 SCM LTPI 链路检测功能。

必须满足以下条件才能进入下一阶段：

- **SCM**：在 TX 链路上发送至少 7 个链路速度帧
- **HPM**：接收至少 3 个链路速度帧

4.5.3 链路配置

链路配置在链路训练阶段完成后开始。它是以在链路速度协商状态中选择的操作频率来执行。链路配置阶段的主要目的是：

- 切换到操作链路频率
- 交换 LTPI 能力
- 启用 LTPI 的选定操作配置
- 切换到操作模式

链接训练阶段基于 3 个状态：

- 通告状态 Advertise
- 配置状态 Configure
- 接受状态 Accept

通告状态是链路从基本频率切换到工作频率时的第一个状态。在通告状态下，SCM 和 HPM 都开始发送通告帧。帧在两个方向上传输。切换到工作频率后，RX Serdes 需要重新执行字节对齐，找帧头，与链路检测阶段类似。该阶段也可以分为 2 个部分：

- **通告-帧对齐**：在这部分中，Serdes 逻辑尝试通过识别逗号符号并处理帧以验证 CRC 来锁定帧的开头。为了进入下一部分，至少需要接收 3 个具有正确 CRC 的帧。
- **通告**：这是 Advertise 的主要部分，其中框架用于解释对方的 LTPI 能力。

SCM 和 HPM 均应持续发送通告帧至少 1ms，以允许链路稳定在工作频率。如果 SCM 先完成 1ms 传输要求并自动开始发送 Configure Frame，则 HPM 应立即切换到 Accept 而不完成 1ms 传输要求。在通告状态下，SCM 和 HPM 持续发送广告帧。该状态没有进入下一个状态的超时时间，这是为了允许 BMC 读取 Advertise 信息并选择 LTPI 配置。BMC 可以使用 CSR 寄存器来验证 LTPI 通告状态是否处于活动状态、读取通告信息并写入 LTPI 的目标配置。

对于 SCM 和 HPM，接下来的状态是不同的。SCM 进入配置状态和 HPM 进入接受状态。要进入下一个状态，SCM 需要开始发送配置帧：

- 正确进入 Advertise 状态后自动根据 SCM LTPI 配置
- 根据 BMC 的请求，遵循 BMC 应用的配置

如果在通告状态下连续丢失 3 个通告帧，则 LTPI 应返回到链路检测状态。

SCM 进入配置状态，HPM 进入接受状态，配置帧仅由 SCM 传输，接受帧仅由 HPM 传输。在此状态下，SCM 开始发送配置帧，并等待 HPM 发回接受帧。

要移至下一个状态，则需：

1. SCM 开始向 HPM 发送配置帧
2. 当 HPM 接收到至少 1 个配置帧并且它与 HPM 功能匹配时，它将具有与 SCM 请求的功能相同的功能的接受帧发送回 SCM。
3. 当接收到接受帧并且它与配置帧功能匹配时，SCM 通过在链路上发送第一个操作帧来移动到下一个状态。

如果 SCM 在发送 32 配置帧后没有收到匹配的接受帧，它将返回通告状态。

4.5.4 链路运行

工作模式是 LTPI 生命周期大部分时间的主要工作条件。在操作模式下，SCM 和 HPM 都不断地双向发送 I/O 帧。

当使用数据通道时，当通过以下方式触发对数据通道的访问时，将按需生成数据帧。当生成数据通道帧时，它将插入 I/O 帧流中。

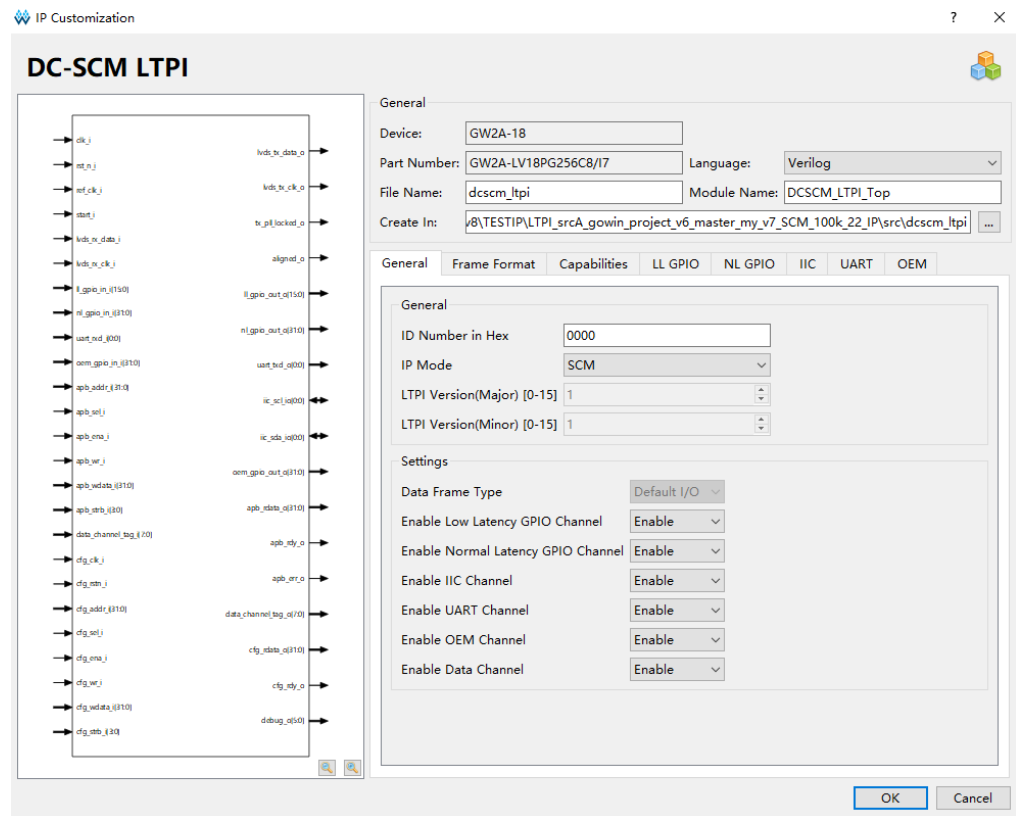
SCM 和 HPM 保持运行状态，直到发生以下情况：

1. 软复位被触发：切换到 **Advertise State**
2. 硬复位被触发（FPGA 复位、断电）：在链路检测状态下启动
3. 连续丢失 7 个帧后，链路丢失：从链路检测状态开始

5 界面配置

在 IP 库中找到 Soft IP Core，在 Interface and Interconnect 一栏中点击 DC-SCM LTPI IP 出现图 5-1。

图 5-1 打开 DC-SCM LTPI IP



Gowin DC-SCM LTPI IP 配置界面如图 5-2 所示。

图 5-2 Gowin DC-SCM LTPI IP 配置界面 1

- ID Number in Hex: ID 号, 输入 16 位位宽十六进制数, 填充 Platform ID 字段
- IP Mode: IP 模式, 选择本端是 SCM 还是 HPM
- LTPI Version(Major): 版本号, 固定为 1, 不能编辑
- LTPI Version(Minor): 版本号, 固定为 1, 不能编辑
- Data Frame Type: 数据帧类型, 暂时不支持自定义类型, 只支持默认类型
- Enable Low Latency GPIO Channel: 使能低延迟 GPIO 通道
- Enable Normal Latency GPIO Channel: 使能正常延迟 GPIO 通道
- Enable IIC Channel: 使能 IIC/SMBus 通道
- Enable UART Channel: 使能串口通道
- Enable OEM Channel: 使能 OEM 通道
- Enable Data Channel: 使能数据通道

图 5-3 Gowin DC-SCM LTPI IP 配置界面 2

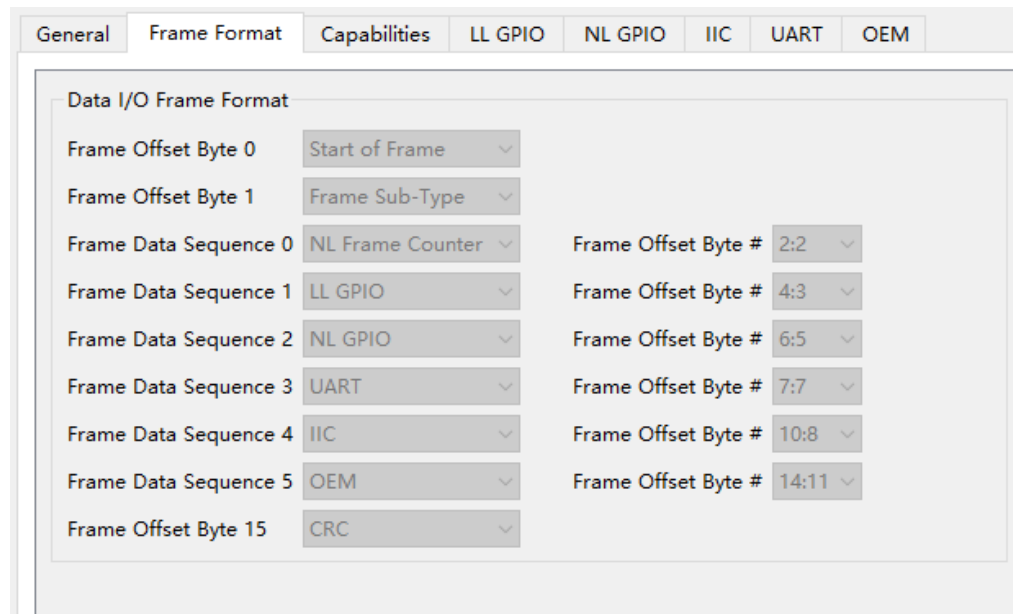
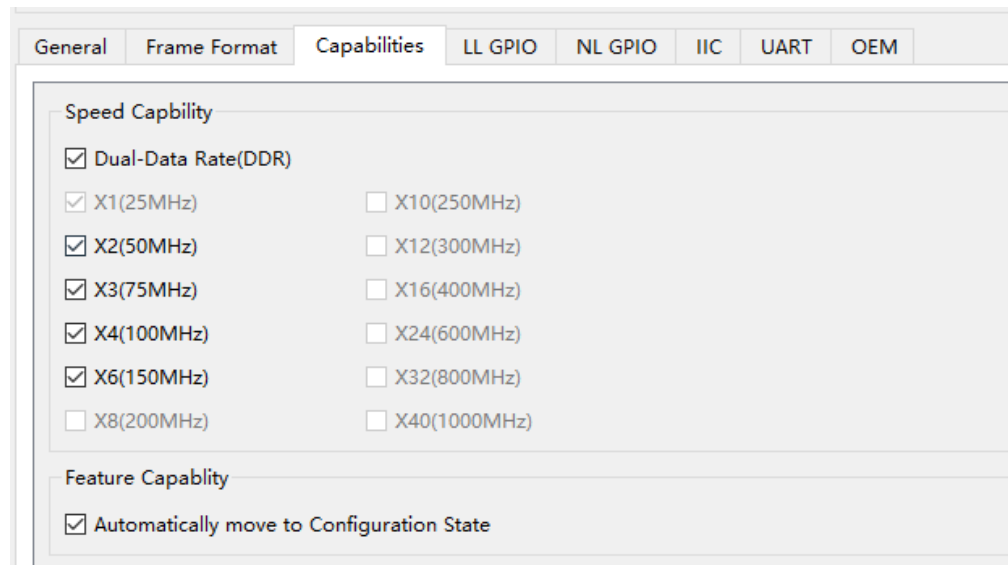


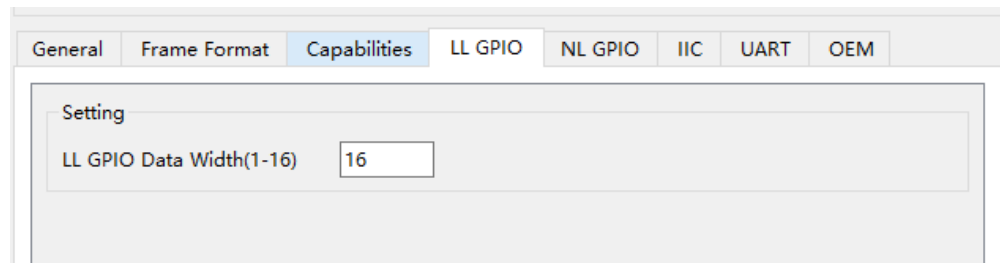
图 5-3 主要显示 Data I/O 帧的结构，因为目前只有默认模式，因此各种通道的位置都是固定的，因此整个页面处于不可编辑状态。

图 5-4 Gowin DC-SCM LTPI IP 配置界面 3



- **Speed Capability:** 速度能力，目前 IP 支持 X1、X2、X3、X4、X6 以及 DDR 模式，可以根据器件能力和实际应用来选择 IP 需要支持的速度和模式。
- **Automatically move to Configuration State:** 自动跳转到配置状态，仅仅 SCM 可编辑，当勾选时，通告状态结束之后将自动进入配置状态，否则需要通过寄存器触发才能进入配置状态。

图 5-5 Gowin DC-SCM LTPI IP 配置界面 4



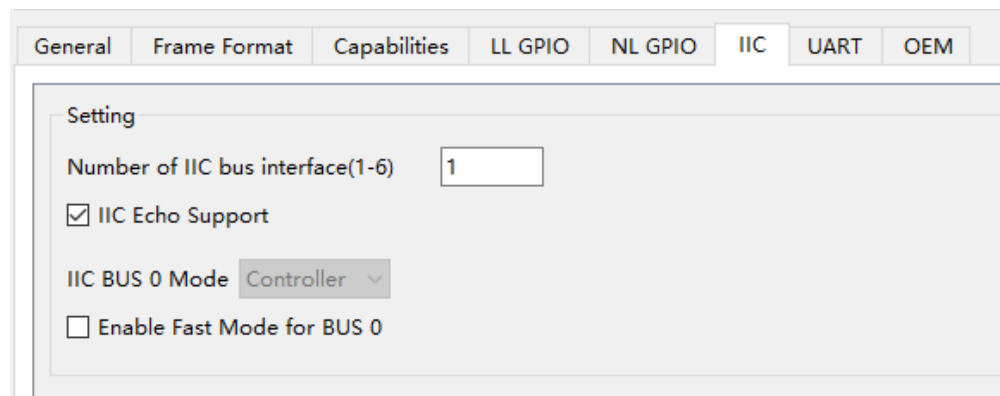
- LL GPIO Data Width: 低延迟 GPIO 个数，最多支持 16 个

图 5-6 Gowin DC-SCM LTPI IP 配置界面 5



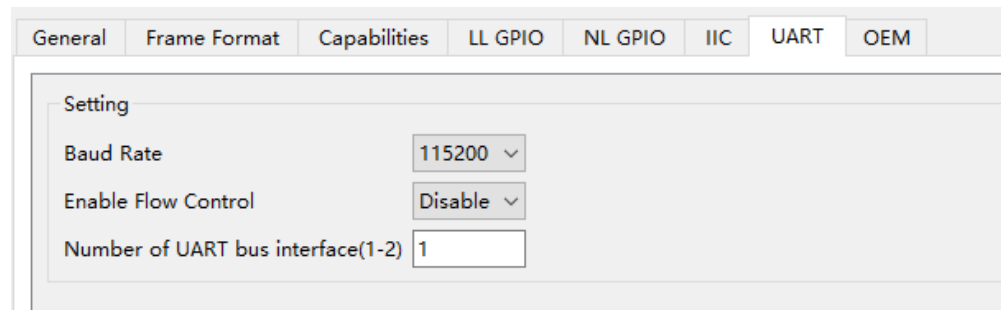
- NL GPIO Data Width: 正常延迟 GPIO 个数，最多支持 1023 个

图 5-7 Gowin DC-SCM LTPI IP 配置界面 6



- Number of IIC bus interface: IIC 总线接口的个数，最大支持 6 个。
- IIC Echo Support: 默认支持。
- IIC BUS 0 Mode: IP 模式为 SCM 的时候，默认为 Controller，IP 模式为 HPM 的时候，默认为 Target，用户可配置。
- Enable Fast Mode for BUS: 勾选后 IIC/SMBus 支持 400KHz，不勾选则是 100KHz。

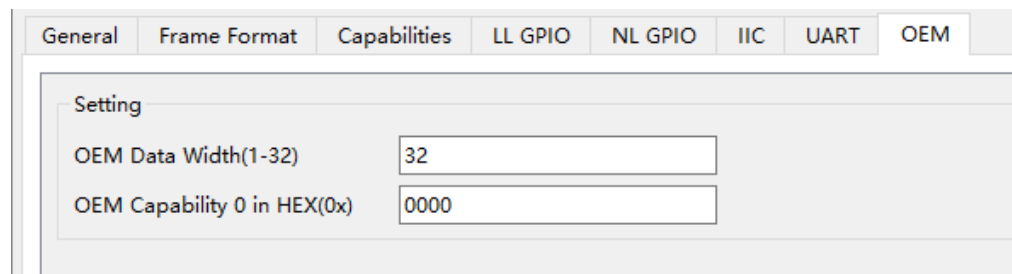
图 5-8 Gowin DC-SCM LTPI IP 配置界面 7



Setting	Value
Baud Rate	115200
Enable Flow Control	Disable
Number of UART bus interface(1-2)	1

- Baud Rate: 波特率设置。
- Enable Flow Control: 流控使能。
- Number of UART bus interface: UART 接口个数，最大可以支持 2 个。

图 5-9 Gowin DC-SCM LTPI IP 配置界面 8



Setting	Value
OEM Data Width(1-32)	32
OEM Capability 0 in HEX(0x)	0000

- OEM Data Width: OEM 数据，最大支持 32bit。
- OEM Capability: 仅填充字段，暂无其他含义。

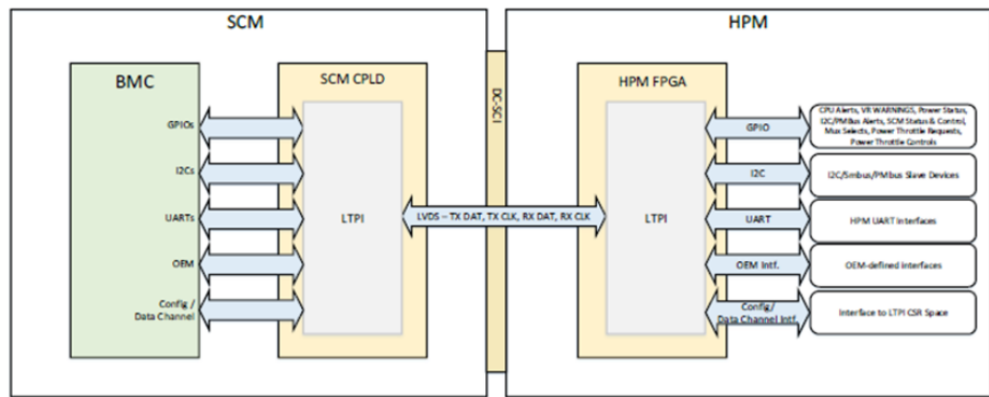
6 参考设计

详细信息请参见高云半导体官网 DC-SCM LTPI IP 相关参考设计。

6.1 应用

DC-SCM LTPI 设计主要用于在 HPM 和 SCM 之间传输各种低速信号。

图 6-1 HPM 和 SCM 的连接

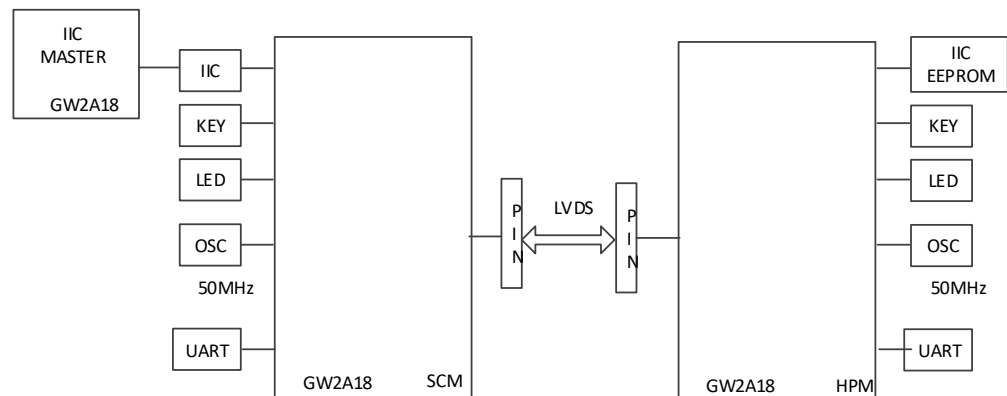


6.2 参考设计

本节主要介绍 Gowin DC-SCM LTPI IP 的参考设计实例的搭建及其使用方法。

6.2.1 硬件平台

图 6-2 平台框图

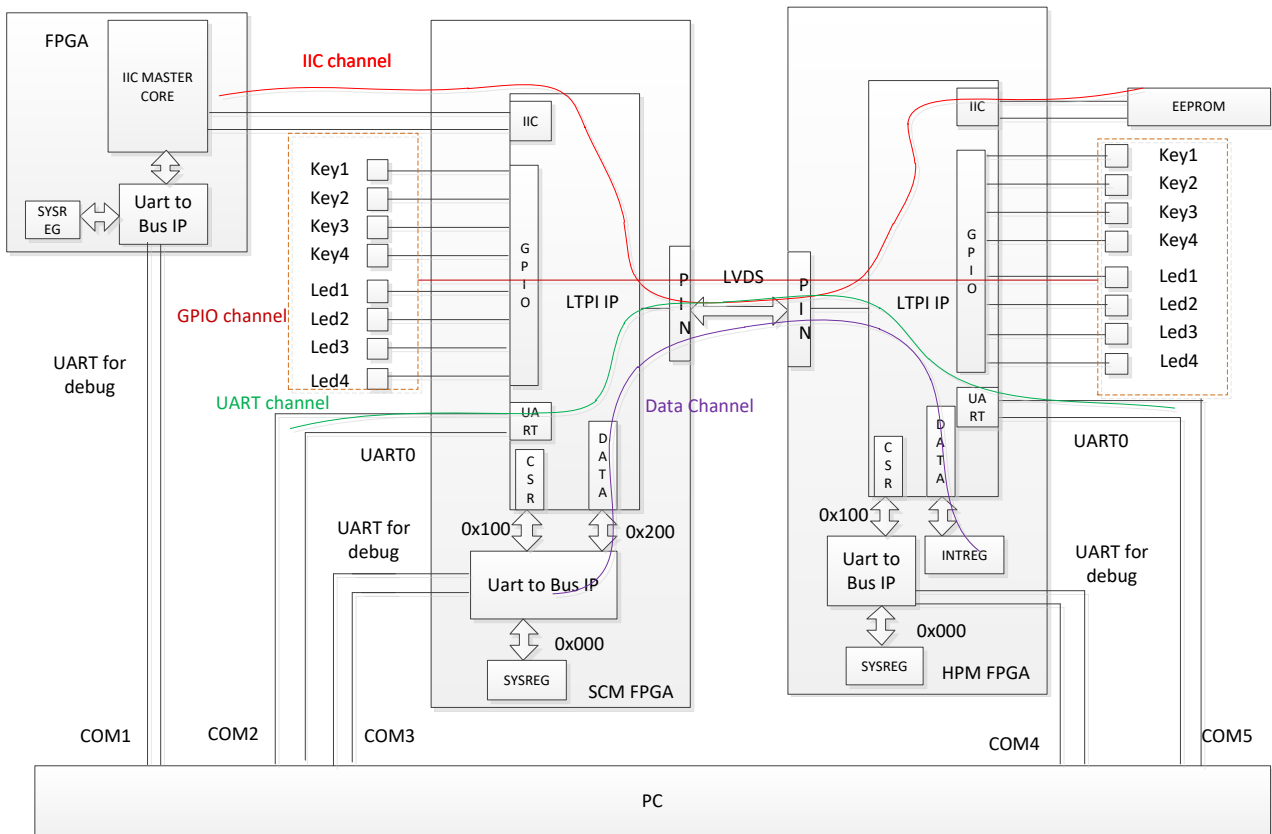


用 3 块 GW2A18 板卡上实现了一个这样的参考实例,其硬件条件如下:

1. 有一个板载 50MHz 的晶振,用于生成系统时钟和参考时钟,总线配置时钟等时钟。
2. 板载 KEY 和 LED 用于 LLGPIO 和 NLGPIO 通道的测试。
3. 排针引出 UART 接入 PC,用于测试 UART 通道的测试。
4. 作为 SCM 的 FPGA 板卡外接另一个 GW2A18 板卡用于模拟 IIC MASTER,作为 HPM 的 FPGA 板卡外接 IIC 接口的 EEPROMz 作为 IIC SLAVE。
5. 通过 UART 接口以及 UART to Bus IP 模拟 APB 时序测试 Data 通道。

6.2.2 实现框图

图 6-3 参考设计实现框图



主要模块如下:

- DC-SCM LTPI IP 模块

该模块设置如该模块的配置如图 5-2 到图 5-9 所示。

- LLGPIO 设置为 16 个
- NLGPIO 设置为 32 个
- OEM Data 宽度设置为 32 个
- IIC/SMBus 个数设置为 1 个

- UART 个数设置为 1 个

其中 KEY 和 LED 的映射路线为下表，用于测试 GPIO 和 OEM 通道：

表 6-1 KEY 和 LED 的映射路线表

起点	路径	终点
KEY1	LLGPIO[0]	LED1
KEY2	NLGPIO[0]	LED2
KEY3	NLGPIO[31]	LED3
KEY4	OEM[0]	LED3

测试 GPIO 和 OEM 通道是双向的。SCM FPGA 的 KEY1 按下去时，HPM FPGA 的 LED1 会点亮；HPM FPGA 的 KEY1 按下去时，SCM FPGA 的 LED1 会点亮。

- UART_to_Bus 模块：串口转总线模块，将串口助手或上位机发送过来的串口指令转化为总线控制时序，用于访问 CSR 寄存器以及测试 Data 通道
- sysreg 模块：用于产生本地系统配置寄存器

6.2.3 总线协议与地址规划

UART_to_Bus IP 模块自定义串口总线协议约定如下：

写总线 “W AAAA BBBBBBBB”

读总线 “R AAAA”

读总线将会返回如下指令 “G AAAA BBBBBBBB”

UART to Bus IP 具体用法可以参考 [IPUG1022, Gowin UART to Bus IP 用户指南](#)。

参考工程中详细的寄存器地址定义如下：

表 6-2 寄存器地址定义

基地址	描述	
0x000	用于控制FPGA工程的系统寄存器	
	0x00[0]	0:开启LTPI, 1:关闭LTPI
	0x01[7:0]	Data通道Tag字段输入
	0x81[0]	可配置PLL锁定指示 1: 锁定 0: 未锁定
	0x82[0]	字节对齐指示 1: 对齐 0: 未对齐
	0x84[7:0]	Data通道Tag字段输出
0x100	用于接LTPI CSR接口	

基地址	描述
0x200	SCM: 用于接LTPI Data通道接口 HPM: 没有此基地址

6.3 板级测试

6.3.1 操作说明

该示例工程是在 DK_START_GW2A18 V2.0 开发板上实测通过的。如果您手上没有此块开发板，但您又想运行参考设计工程，您只要对串口管脚、时钟输入管脚根据您的板卡的实际情况进行修改即可用于测试。

修改完成之后，生成 bit 文件，下载到 FPGA 中即可开始测试

6.3.2 操作步骤

1. 连线

- 将 SCM 和 HPM 之间的 LVDS 线连接好，注意接口要满足 LVDS 接口的电气规范。
- 用于做 IIC Master 的 FPGA 与 SCM FPGA 连接好，注意，IIC 的 SDA 和 SCL 需要在硬件上置为上拉，注意接口要满足 IIC 接口的电气规范。
- 将串口接口连接与 PC，记录各个串口的串口号，方便后续调试。

2. 三块 FPGA 板卡上电，查询 LTPI 状态

默认情况下 SCM 和 HPM 上电就会开始运行，此时可以通过连接 CSR 寄存器的串口来查询 LTPI 内部的状态。

通过 COM3 查询 SCM 的 CSR 寄存器，通过 COM4 查询 HPM 的 CSR 寄存器(具体 COM 号需要查询 PC 机设备管理器)，波特率设置为 460800bps，数据为 8bit，无校验位，停止位为 1bit。

- 发送指令 R 81 反馈为 G 0081 00000001，表明内部 PLL 锁定
- 发送指令 R 82 反馈为 G 0082 00000001，表明字节对齐
- 发送指令 R 100 反馈为 G 0100 00044481

根据 CSR 寄存器定义，表明 SCM 和 HPM 都已经处于运行状态

3. 测试 GPIO 和 OEM 通道

按下 SCM 或 HPM 的 KEY1~KEY4，则对端 LED1~LED4 对应的会点亮，此表明 GPIO 和 OEM 通道正常。

4. 测试 UART 通道

连接 SCM 和 HPM 的 UART 通道对应的串口 COM2 和 COM5 (具体 COM 号需要查询 PC 机设备管理器)，两边都设置波特率为 115200。两边发送任意内容，对端都能正确收到，则表示 UART 通道正常。

5. 测试 Data 通道

Data 通道通过 UART to Bus IP 将串口指令转变成了 APB 总线时序，映射的基地址是 0x200。

在 SCM 的调试串口 COM3 输入 W 200 12345678, 然后进行回读 R 200, 如果返回结果为 G 0200 12345678 则表明 Data 通道正常

6. 测试 IIC 通道

测试 IIC 通道需要 SCM 外部有一个 IIC 主控制器来发出 IIC 时序，同时需要一个 IIC slave 器件接到 HPM 上，这里选择了 EEPROM 器件。因此还有一块开发板专门用于实现 IIC 主控制器，IIC 主控制器使用 Gowin IIC Master IP 来实现，具体使用可以查询 Gowin IIC Master IP 用户手册，这里同样也是使用 UART to Bus IP 进行调试，将 0x100 基地址接入 IIC Master IP 的内部总线端口。

此处使用的串口是 COM1（具体 COM 号需要查询 PC 机设备管理器），波特率设置为 460800bps，数据为 8bit，无校验位，停止位为 1bit。

调试指令如下：

1. 首先对 IIC EEPROM 的 0x01 地址写入 0x11, 0x02 地址写入 0x12, 指令如下：

W 0100 c8	——设置 IIC 速率
W 0101 00	——设置 IIC 速率
R 0100	——回读 IIC 速率 返回值 G 0100 000000C8
W 0102 80	——启动 IIC 核
W 0103 A0	——设置 IIC 器件地址，并设置为写操作
W 0104 90	——生成 start 时序和写器件地址时序
R 0104	——等待操作完成，返回值 G 0104 00000041
W 0103 01	——设置 IIC 器件内部地址 0x01
W 0104 10	——内部地址 0x01 写入，生成写内部地址时序
R 0104	——等待操作完成，返回值 G 0104 00000041
W 0103 11	——把 0x11 写入当前 IIC 地址(0x01)
W 0104 10	——生成写数据时序
R 0104	——等待操作完成，返回值 G 0104 00000041
W 0103 12	——把 0x12 写入下一个 IIC 地址(0x02，自动递增)
W 0104 50	——生成写数据时序和停止时序
R 0104	——等待操作完成，返回值 G 0104 00000001

2. 对 IIC EEPROM 的 0x01, 0x02 地址进行回读，操作指令如下：

W 0100 c8	——设置 IIC 速率
W 0101 00	——设置 IIC 速率

R 0100	——回读 IIC 速率 返回值 G 0100 000000C8
W 0102 80	——启动 IIC 核
W 0103 A0	——设置 IIC 器件地址，并设置为写操作
W 0104 90	——生成 start 时序和写器件地址时序
R 0104	——等待操作完成，返回值 G 0104 00000041
W 0103 01	——设置 IIC 器件内部地址 0x01
W 0104 10	——内部地址 0x01 写入，生成写内部地址时序
R 0104	——等待操作完成，返回值 G 0104 00000041
W 0103 A1	——设置 IIC 器件地址，并设置为读操作
W 0104 90	——生成 start 时序和写地址时序
R 0104	——等待操作完成，返回值 G 0104 00000041
W 0104 20	——读当前 IIC 地址(0x01)数据
R 0103	——等待返回值,返回指令 G 0103 00000011
W 0104 28	——最后一次读 IIC 地址(0x02)数据
R 0103	——等待返回值,返回指令 G 0103 00000012
W 0104 40	——生成 STOP 时序
R 0104	——等待操作完成，返回指令 G 0104 00000081

7 文件交付

Gowin DC-SCM LTPI IP 交付文件主要包含三个部分，分别为：文档、设计源代码和参考设计。

7.1 文档

文件夹主要包含用户指南 PDF 文档。

表 7-1 文档列表

名称	描述
IPUG1086, Gowin DC-SCM LTPI IP 用户指南	高云 DC-SCM LTPI IP 用户手册，即本手册

7.2 设计源代码（加密）

加密代码文件夹包含 Gowin DC-SCM LTPI IP 的 RTL 加密代码，供 GUI 使用，以配合高云半导体云源®软件产生用户所需的 IP 核。

表 7-2 Gowin DC-SCM LTPI IP 设计源代码列表

名称	描述
ltpi_core.v	IP 核顶层文件，给用户接口信息，加密。

7.3 参考设计

Gowin DC-SCM LTPI IP RefDesign 文件夹主要包含 Gowin DC-SCM LTPI IP 的网表文件，用户参考设计，约束文件、顶层文件及工程文件夹等。

表 7-3 Gowin DC-SCM LTPI IP RefDesign 文件夹内容列表

名称	描述
top.v	参考设计的顶层 module
top.cst	工程物理约束文件
top.sdc	工程时序约束文件
dcscm_ltpi.v	生成 DC-SCM LTPI IP 顶层文件，加密
uart_to_bus.v	串口转总线模块，加密
sysreg.v	SysReg 生成模块
syspll.v	PLL IP 文件

