



Gowin Hardened LICD 用户指南

UG1452-1.0, 2025-03-28

版权所有 © 2025 广东高云半导体科技股份有限公司

GOWIN高云、**GOWIN**、、**GOWINSEMI**、**GOWIN**、Gowin、**高云**、晨熙、小蜜蜂、LittleBee、**Arora-V**、GowinPnR、GoBridge 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、翻译本文档内容的部分或全部，并不得以任何形式传播。

免责声明

本文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。高云半导体对文档中包含的文字、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文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文档进行适时的更新。

版本信息

日期	版本	说明
2025/03/28	1.0	初始版本。

目录

插图清单.....	ii
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 概述.....	3
2.1 特性.....	3
2.2 功能结构.....	4
2.2.1 编码功能结构.....	4
2.2.2 解码功能结构.....	6
3 LICD 原语.....	8
3.1 LICD.....	8
3.1.1 LICD 端口介绍.....	8
3.1.2 LICD 参数介绍.....	9
3.1.3 原语例化.....	10
4 IP 的调用.....	11
4.1 LICD IP.....	11
4.1.1 端口介绍.....	11
4.1.2 典型时序.....	13
4.1.3 配置说明.....	13
5 设计参考.....	15
5.1 RGB 格式.....	15
5.1.1 直接连接方式.....	15
5.1.2 单编码器交织模式.....	17
5.2 RGBG 格式.....	18
5.2.1 双编码器交织模式.....	18
5.3 应用须知.....	20
5.3.1 配置说明.....	20

图目录

图 2-1 编码示意图.....	5
图 2-2 解码示意图.....	7
图 4-1 ENCODE PORTS IP 典型时序.....	13
图 4-2 DECODER PORTS IP 典型时序.....	13
图 4-3 LICD IP 配置界面.....	13

1 关于本手册

1.1 手册内容

Gowin LICD 用户指南主要内容包括功能特点、端口描述、配置调用等，主要用于帮助用户快速了解 LICD 的特性及使用方法。

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、查看以下相关文档：

- [DS981, GW5AT 系列 FPGA 产品数据手册](#)
- [DS1118, GW5ART 系列 FPGA 产品数据手册](#)
- [DS1111, GW5AT 系列 FPGA 产品\(车规级\)数据手册](#)

1.3 术语、缩略语

表 1-1 中列出了本手册中出现的相关术语、缩略语及相关释义。

表 1-1 术语、缩略语

术语、缩略语	全称	含义
LICD	Lossless Image Compression and Decompression	无损图像压缩与解压缩
Encoder	-	编码器: 使用特定算法将原始图像样值使用较短的码字替换,从而得到压缩码流
Decoder	-	解码器: 与编码器功能相反,将压缩码流中的码字恢复成原始图像样值
EncDec	-	编解码器: 编码器和解码器的合

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 概述

LICD 是一种 Gowin 开发的一个用于图像无损压缩与解压缩的硬核 IP，方便客户进行图像或数据的压缩与解压缩，降低接口传输带宽。

2.1 特性

Gowin LICD 主要特性如下：

- 支持多个编码器/解码器并行工作，提高数据处理效率。
- 支持不同的数据输入格式（RGB，YCbCr444，YCbCr422，RGBG）以及其他自定义数据，并通过配置灵活调整编码器/解码器的工作模式。
- 最高可处理 4k@60Hz 8-bit RGB 或 YCbCr444 格式的视频
- 支持像素级无损压缩，平均压缩率 25%-70%
- 无需行缓存，节省硬件资源
- 实时性与低延迟
 - 每个时钟周期可处理多达 3 个图像采样点
 - 仅 4-5 时钟周期延迟

2.2 功能结构

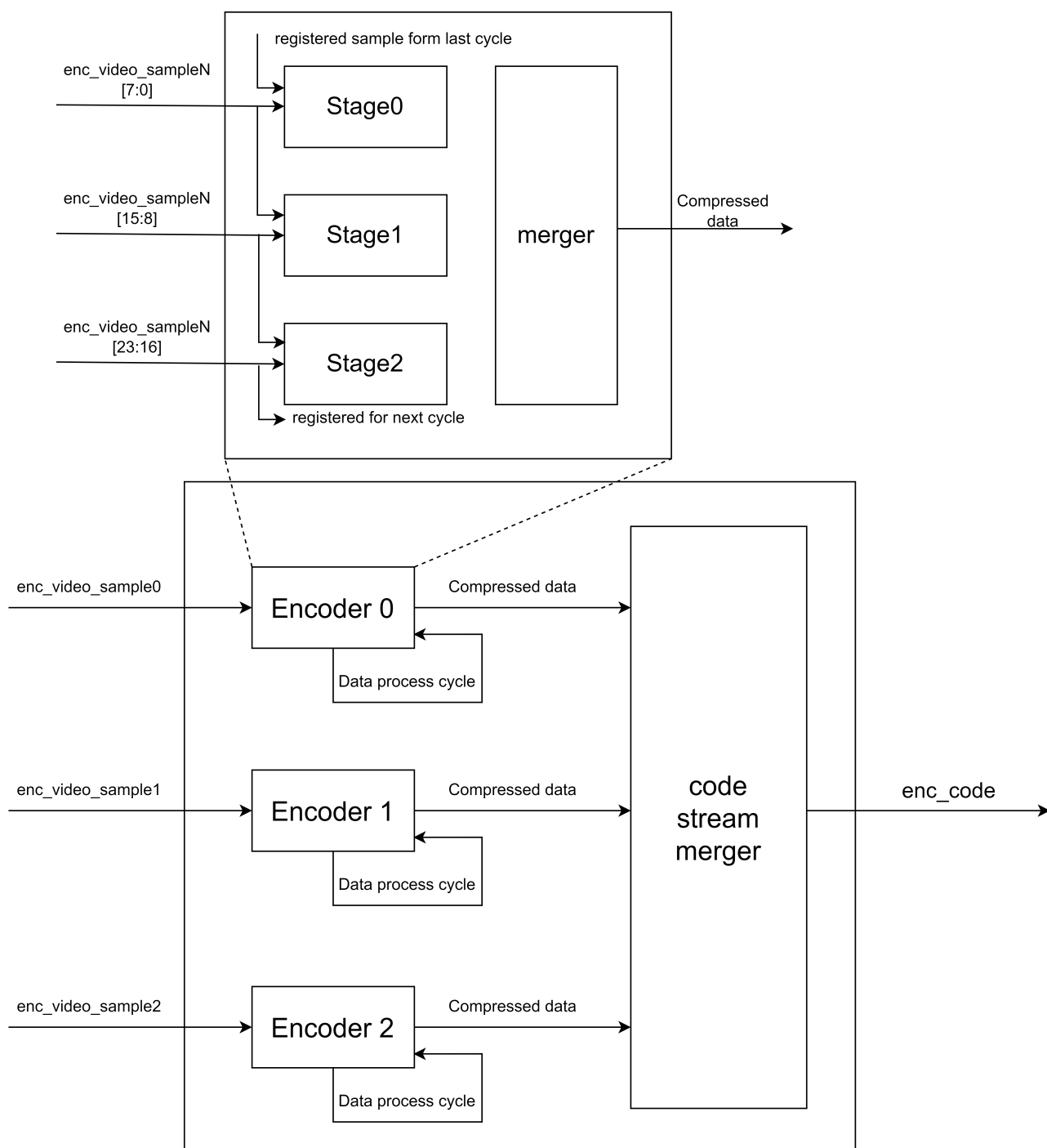
2.2.1 编码功能结构

LICD 编码模块包含三个并行的硬核编码器，每个编码器采用三级编码架构，并可配置为直连模式或交织模式：

- 直连模式：每个解码器仅处理一种数据类型，独立运行。
- 交织模式：每个解码器分时处理不同的数据类型，包括：
 - 交织 AB 模式：解码器交替处理两种数据（A、B）。
 - 交织 ABC 模式：解码器交替处理三种数据（A、B、C）。

具体的处理方法请参考 [5 设计参考](#)。

图 2-1 编码示意图



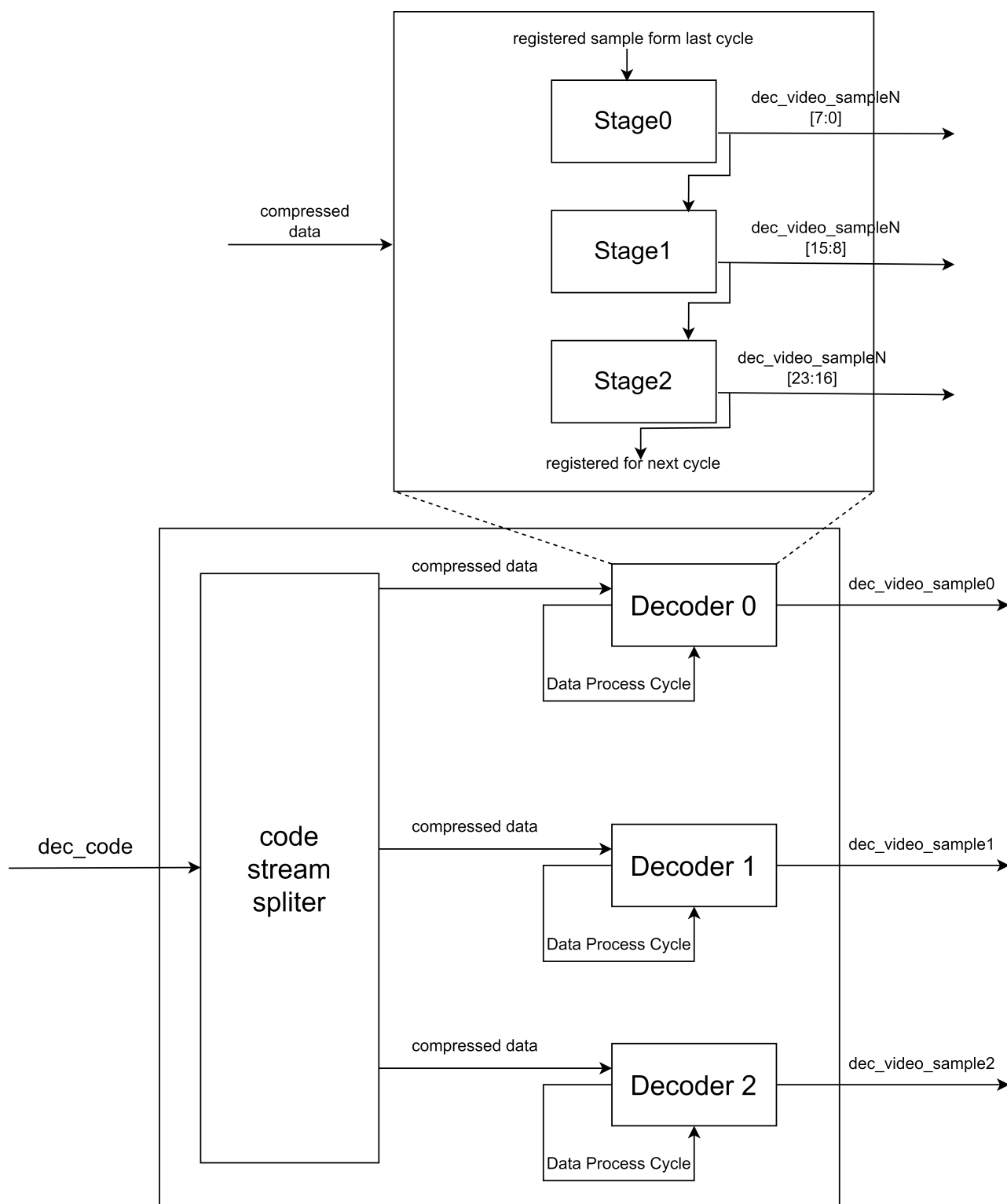
2.2.2 解码功能结构

IP 解码模块包含三个并行的硬核解码器，每个解码器采用三级解码架构，并可配置为直连模式或交织模式：

- 直连模式：每个解码器仅处理一种数据类型，独立运行。
- 交织模式：解码器分时处理不同的数据类型，包括：
 - 交织 AB 模式：解码器交替处理两种数据（A、B）。
 - 交织 ABC 模式：解码器交替处理三种数据（A、B、C）。

具体的处理方法请参考 [5 设计参考](#)。

图 2-2 解码示意图



3 LICD 原语

3.1 LICD

3.1.1 LICD 端口介绍

表 3-1 LICD 端口介绍

端口	I/O	描述
ENC_CLK	Input	编码时钟输入
ENC_CE	Input	编码时钟门控输入（高电平使能时钟）
ENC_RST	Input	编码模块复位输入（高有效）
ENC_VIDDE	Input	编码数据有效指示信号（高有效）
ENC_VIDSAMP2[23:0]	Input	像素分量输入，对应编码器 2 输入 • STAGE_NUM = 1 使用低 8bit • STAGE_NUM = 2 使用低 16bit • STAGE_NUM = 3 使用 24bit
ENC_VIDSAMP1[23:0]	Input	像素分量输入，对应编码器 1 输入 • STAGE_NUM = 1 使用低 8bit • STAGE_NUM = 2 使用低 16bit • STAGE_NUM = 3 使用 24bit
ENC_VIDSAMP0[23:0]	Input	像素分量输入，对应编码器 0 输入 • STAGE_NUM = 1 使用低 8bit • STAGE_NUM = 2 使用低 16bit • STAGE_NUM = 3 使用 24bit
ENC_CODE[95:0]	Output	像素压缩编码输出，压缩编码输出，实际使用位宽跟设置中 STAGE_NUM 和 CDEC_NUM 以及 CODE_WIDTH 有关
ENC_CODEVLD	Output	像素压缩编码有效指示信号
ENC_VIDDEO	Output	ENC_VIDDE 延时输出
DEC_CLK	Input	解码时钟输入，解码像素时钟(频率同编码像素时钟)

端口	I/O	描述
DEC_CE	Input	解码时钟门控输入（高电平使能时钟）
DEC_RST	Input	解码模块复位输入（高电平有效）
DEC_VIDDE	Input	解码数据使能指示（高有效），当该信号有效时，解码模块开始读取压缩数据并将读取的压缩数据解压恢复原始数据
DEC_DATAI[95:0]	Input	像素编码输入，压缩编码输入，实际使用位宽跟设置中 STAGE_NUM 和 CDEC_NUM 以及 CODE_WIDTH 有关
DEC_DFETCH	Output	像素编码信号有效指示（高有效），下一个数据必须在该信号使能前准备好
DEC_VIDSAMP[23:0]	Output	像素分量解码输出，对应解码器 2 输出 • STAGE_NUM = 1 使用低 8bit • STAGE_NUM = 2 使用低 16bit • STAGE_NUM = 3 使用 24bit
DEC_VIDSAMP[23:0]	Output	像素分量解码输出，对应解码器 1 输出 • STAGE_NUM = 1 使用低 8bit • STAGE_NUM = 2 使用低 16bit • STAGE_NUM = 3 使用 24bit
DEC_VIDSAMP[23:0]	Output	像素分量解码输出，对应解码器 0 输出 • STAGE_NUM = 1 使用低 8bit • STAGE_NUM = 2 使用低 16bit • STAGE_NUM = 3 使用 24bit
DEC_VIDDEOO	Output	DEC_VIDDE 信号延时输出

3.1.2 LICD 参数介绍

表 3-2 LICD 参数介绍

参数	默认	描述
STAGE_NUM	2'b00	控制每个 编解码器在每个时钟周期内处理的 video 采样点数量。
ENCDEC_NUM	2'b00	控制并行工作的 编解码器 数量。
CODE_WIDTH	2'b00	定义压缩码流数据的总线宽度。
INTERLEAVE_EN	3'b000	控制每个 编解码器 是否启用交织模式。
INTERLEAVE_MODE	3'b000	为每个 编解码器设置交织模式的具体类型（AB 或 ABC 模式）。

3.1.3 原语例化

```
LICD licd_inst (  
  .ENC_CODE(enc_code),  
  .ENC_CODEVLD(enc_codevld),  
  .ENC_VIDDEO(enc_viddeo),  
  .DEC_DFETCH(dec_dfetch),  
  .DEC_VIDSAMP2(dec_vidsamp2),  
  .DEC_VIDSAMP1(dec_vidsamp1),  
  .DEC_VIDSAMP0(dec_vidsamp0),  
  .DEC_VIDDEO(dec_viddeo),  
  .ENC_CLK(enc_clk),  
  .ENC_CE(enc_ce),  
  .ENC_RST(enc_rst),  
  .ENC_VIDDE(enc_vidde),  
  .ENC_VIDSAMP2(enc_vidsamp2),  
  .ENC_VIDSAMP1(enc_vidsamp1),  
  .ENC_VIDSAMP0(enc_vidsamp0),  
  .DEC_CLK(dec_clk),  
  .DEC_CE(dec_ce),  
  .DEC_RST(dec_rst),  
  .DEC_VIDDE(dec_vidde),  
  .DEC_DATAI(dec_datai)  
);  
defparam licd_inst.STAGE_NUM = 2'b00;  
defparam licd_inst.ENCDEC_NUM = 2'b00;  
defparam licd_inst.CODE_WIDTH = 2'b00;  
defparam licd_inst.INTERLEAVE_EN = 3'b000;  
defparam licd_inst.INTERLEAVE_MODE = 3'b000;
```

4 IP 的调用

为方便使用，高云半导体云源软件对硬核 LICD 进行了 IP 封装。封装后的 IP 适合大部分使用场景，如果有进一步需求，可在技术支持的指导下使用原语方式例化。

4.1 LICD IP

4.1.1 端口介绍

表 4-1 LICD IP 顶层端口

端口	I/O	描述
dec_clk	Input	解码时钟输入，解码像素时钟(频率同编码像素时钟)
dec_ce	Inpu	解码时钟门控输入（高电平使能时钟）
dec_rst	Input	解码模块复位输入（高电平有效）
dec_vidde	Input	解码数据使能指示（高有效），当该信号有效时，解码模块开始读取压缩数据并将读取的压缩数据解压恢复原始数据
dec_datai[95:0]	Input	解码数据输入
enc_clk	Input	编码时钟输入
enc_ce	Input	编码时钟门控输入（高电平使能时钟）
enc_rst	Input	编码模块复位输入（高有效）
enc_vidde	Input	编码数据有效指示信号（高有效）
enc_vidsamp2[23:0]	Input	像素分量输入，对应编码器 2 输入 ● Stage Number = 1 使用低 8bit ● Stage Number = 2 使用低 16bit ● Stage Number = 3 使用 24bit
enc_vidsamp1[23:0]	Input	像素分量输入，对应编码器 1 输入 ● Stage Number = 1 使用低 8bit ● Stage Number = 2 使用低 16bit

端口	I/O	描述
enc_vidsamp1[23:0]	Input	● Stage Number = 3 使用 24bit
enc_vidsamp0[23:0]	Input	像素分量输入，对应编码器 0 输入 ● Stage Number = 1 使用低 8bit ● Stage Number = 2 使用低 16bit ● Stage Number = 3 使用 24bit
enc_code[95:0]	Output	像素压缩编码输出，实际使用位宽跟设置中 Stage Number 和 Coder Number 有关
enc_codevid	Output	像素压缩编码有效指示信号
enc_viddeo	Output	enc_vidde 延时输出
dec_dfetch	Output	像素编码信号有效指示（高有效），下一个数据必须在该信号使能前准备好
dec_vidsmap2[23:0]	Output	像素分量解码输出，对应解码器 2 输出 ● Stage Number = 1 使用低 8bit ● Stage Number = 2 使用低 16bit ● Stage Number = 3 使用 24bit
dec_vidsmap1[23:0]	Output	像素分量解码输出，对应解码器 1 输出 ● Stage Number = 1 使用低 8bit ● Stage Number = 2 使用低 16bit ● Stage Number = 3 使用 24bit
dec_vidsmap0[23:0]	Output	像素分量解码输出，对应解码器 0 输出 ● Stage Numbe = 1 使用低 8bit ● Stage Numbe = 2 使用低 16bit ● Stage Numbe = 3 使用 24bit
dec_viddeo	Output	dec_vidde 信号延时输出

4.1.2 典型时序

图 4-1 ENCODE PORTS IP 典型时序

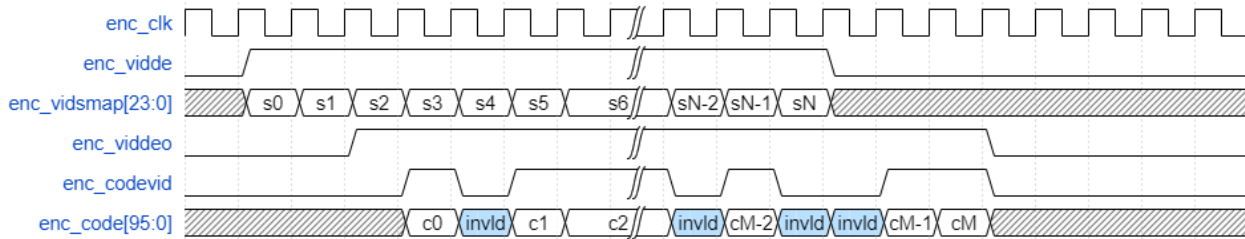
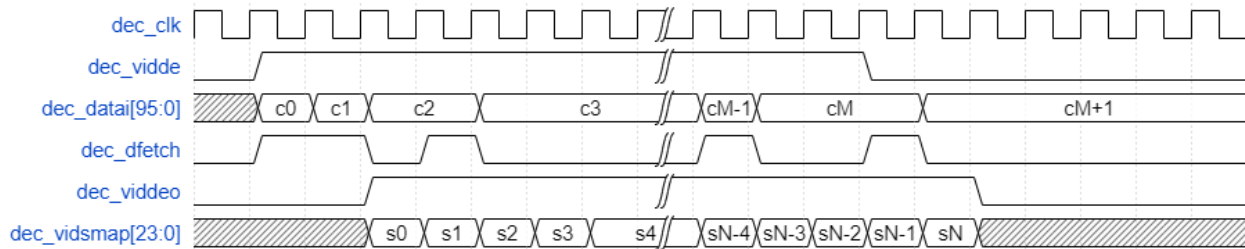
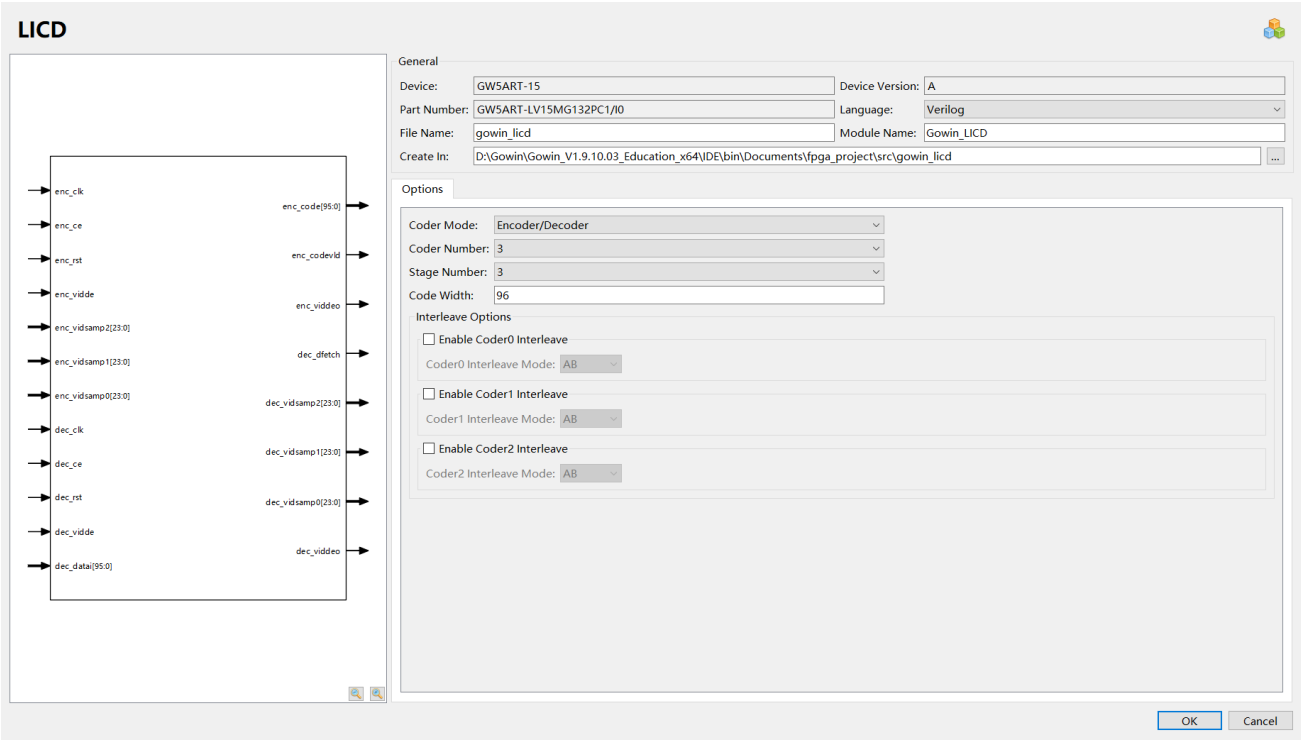


图 4-2 DECODER PORTS IP 典型时序



4.1.3 配置说明

图 4-3 LICD IP 配置界面



General 配置框

General 配置框用于配置产生的 IP 设计文件的相关信息。

- Device: 显示已配置的 Device 信息;
- Part Number: 显示已配置的 Part Number 信息;

- **Language:** 配置产生的 IP 设计文件的硬件描述语言。选择右侧下拉列表框，选择目标语言，支持 Verilog 和 VHDL；
- **Module Name:** 配置产生的 IP 设计文件的 **Module name**。在右侧文本框可重新编辑模块名称。**Module Name** 不能与原语名称相同，若相同，则报出 **Error** 提示；
- **File Name:** 配置产生的 IP 设计文件的文件名。在右侧文本框可重新编辑文件名称；
- **Create In:** 配置产生的 IP 设计文件的目标路径。可在右侧文本框中重新编辑目标路径，也可通过文本框右侧选择按钮选择目标路径。

Option 配置框

- **Coder Mode:** 配置模式，编码/解码模式
- **Coder Number:** 配置编码/解码器的数量(最多选 3 个)
- **Stage Number:** 配置并行处理的样本数(最多选 3 个)
- **Code Width:** 数据位宽，只读
- **Enable Coder0 Interleave:** 使能 Coder0 交错模式
- **Coder0 Interleave Mode:** AB/ABC 模式，配置 Coder0 的交错模式为 AB 模式或 ABC 模式
- **Enable Coder1 Interleave:** 使能 Coder1 交错模式
- **Coder1 Interleave Mode:** AB/ABC 模式，配置 Coder1 的交错模式为 AB 模式或 ABC 模式
- **Enable Coder2 Interleave:** 使能 Coder2 交错模式
- **Coder2 Interleave Mode:** AB/ABC 模式，配置 Coder2 的交错模式为 AB 模式或 ABC 模式

5 设计参考

LICD 模块可以处理两种格式的图像数据：RGB 格式和 RGBG 格式。详细信息请参考 [5.1 RGB 格式](#) 及 [5.2 RGBG 格式](#) 所述。

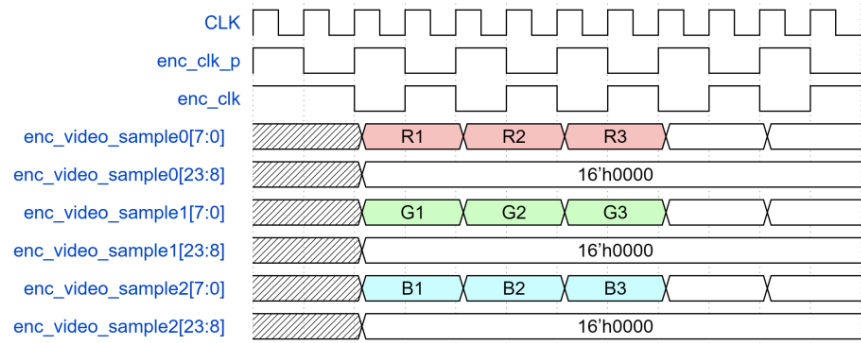
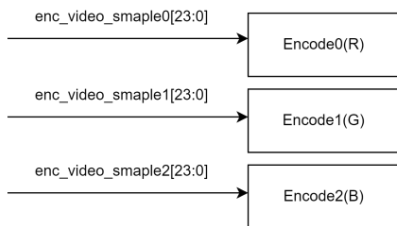
5.1 RGB 格式

RGB 格式的图像数据可以通过以下两种方式进行处理：

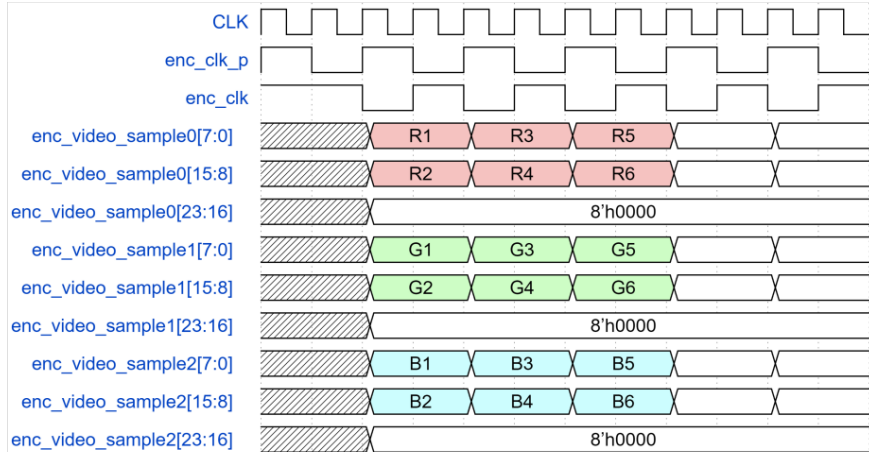
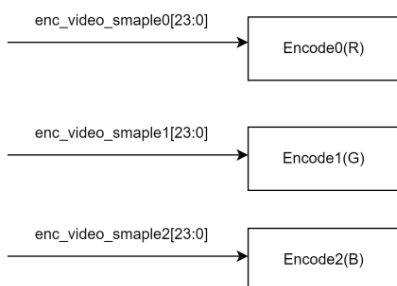
5.1.1 直接连接方式

- 使用使用 3 个编码器，每个编码器分别处理一个颜色分量：
 - Encoder0 处理 R 分量。
 - Encoder1 处理 G 分量。
 - Encoder2 处理 B 分量。
- 每个编码器独立工作，处理各自对应的颜色分量。
- 适用于需要高并行处理的场景。

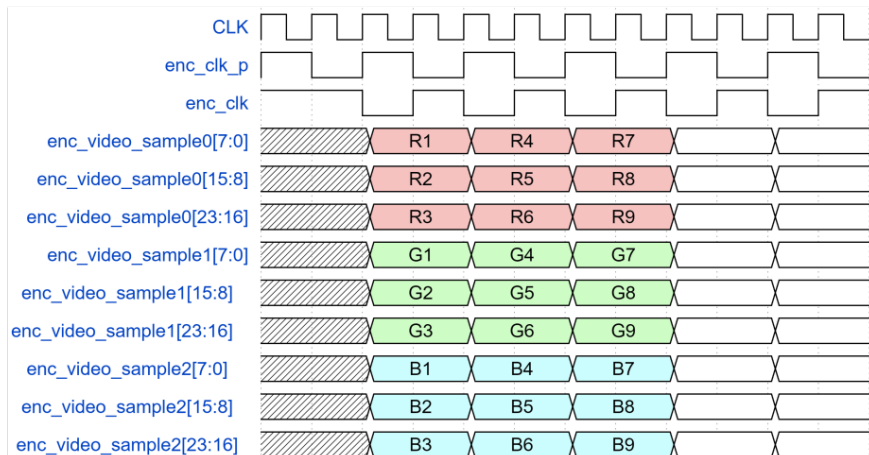
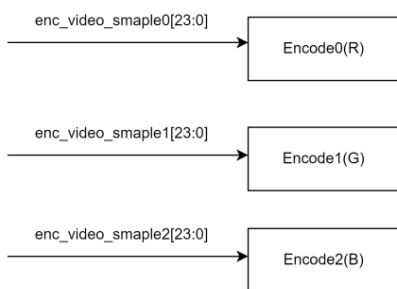
Using 1 Stage



Using 2 Stage



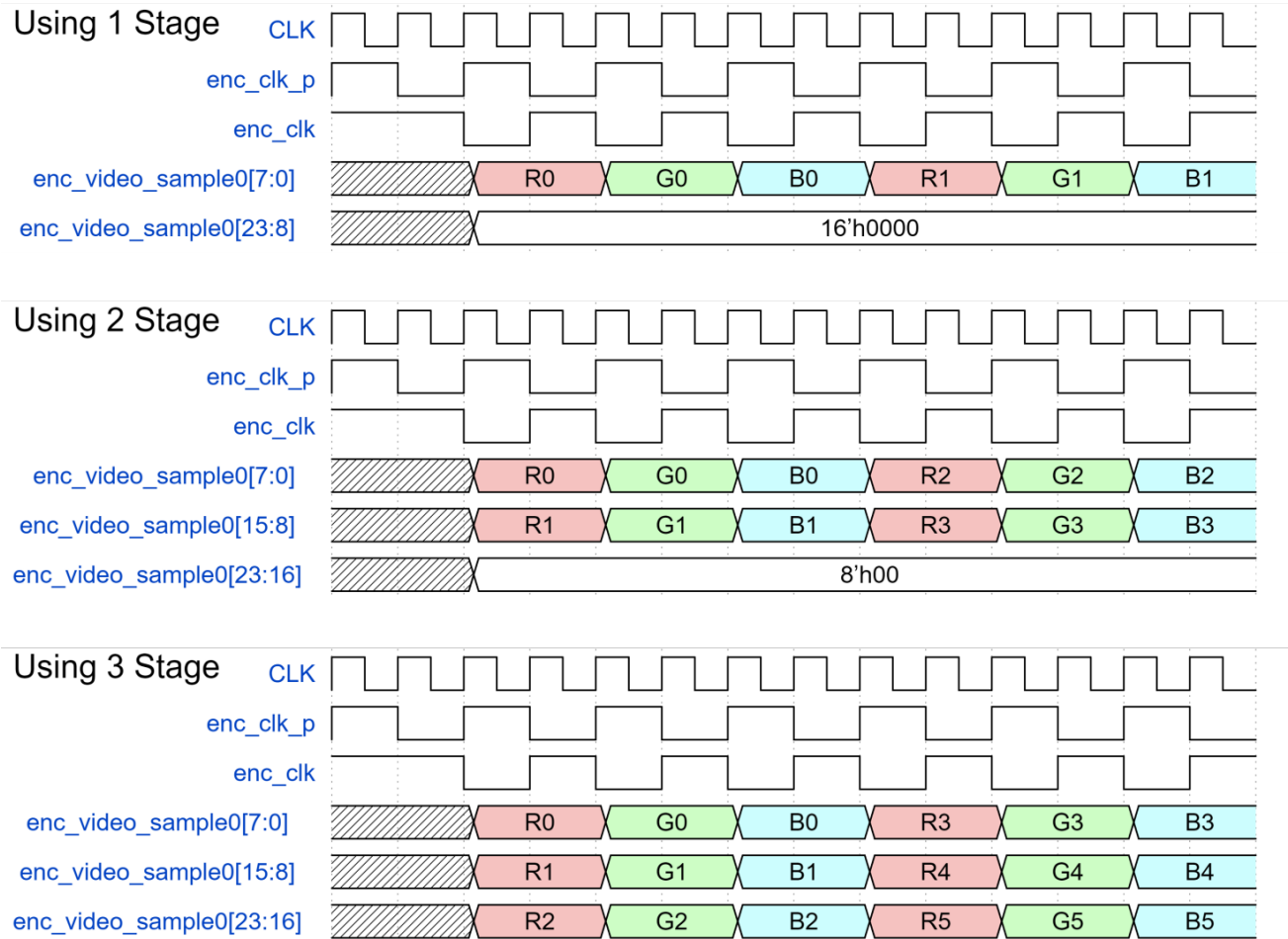
Using 3 Stage



5.1.2 单编码器交织模式

使用 1 个编码器（仅 Encoder0），并通过配置 IPEnable Coder0 interleave，同时配置 Coder0 Interleave Mode 为 ABC 模式，处理 R、G、B 三个分量。

- 通过交织模式，多个颜色分量共享一个编码器。
- 适用于资源有限或需要减少硬件复杂度的场景。



5.2 RGBG 格式

5.2.1 双编码器交织模式

对于 RGBG 的处理，必须使用 2 个 encoder，并且 Interleave Mode=AB 模式。Encode0 处理 R 分量和 B 分量，Encoder1 处理两个 G 分量， Encoder2 不做使用。

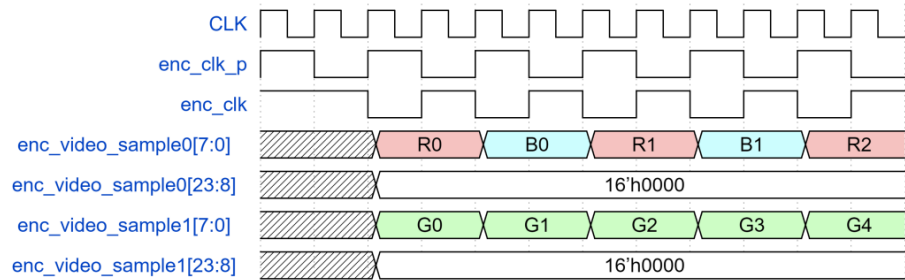
注！

硬核做的是数据的压缩和解压，不考虑像素点的信息，像素点信息的排布，需要上层用户的代码完成，例如在一个内存中，存储着 R 分量的信息，B 分量的信息，和 G 分量的信息（只不过对于 RGBG 的格式，G 分量是 R 和 B 分量的两倍）。

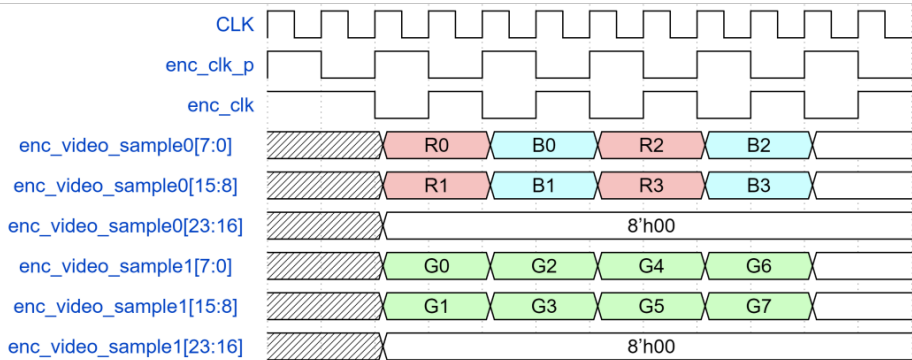
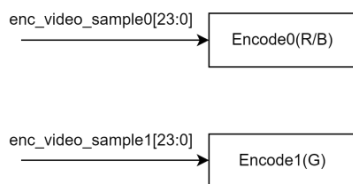
内存1	R0 R1 R2 R3 R4 ……RN	
内存2	B0 B1 B2 B3 B4 ……BN	
内存3	G0 G1 G2 G3 G4 ……GN	GN G(N+1) G(N+2) G(N+3) G4 ……G((2N+1))

其中 R0 B0 G0 G1 并没有什么关系，并不是一个像素点的信息，RGB 具体的排布方式，是由 上层用户代码决定，LICD 模块只是对进入的数据做压缩，LICD 解压之后，将数据送到上次用户代码，再有用户去归类，合并成实际图像的像素点信息。

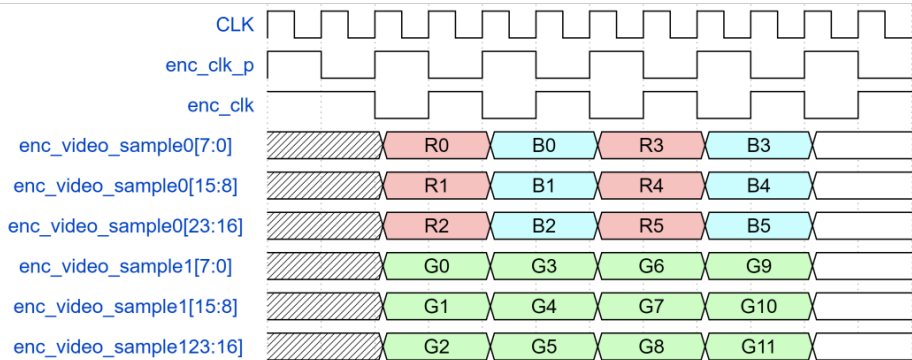
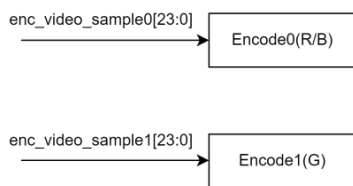
Using 1 Stage



Using 2 Stage



Using 3 Stage



对于交织模式，使用 **stage2/3** 时，注意图中 **G** 分量给的顺序。

5.3 应用须知

5.3.1 配置说明

在复位信号有效后，需要等 5 个时钟周期后，再给正式的数据（确保复位已经释放掉），不然存在前几个数据丢失的情况。

