

## 摘要

DLPC910 Apps FPGA 指南介绍了 DLPC910 应用 FPGA (Apps FPGA) 的功能和寄存器。Apps FPGA 旨在与 DLP LightCrafter DLPC910 EVM (DLPLCRC910EVM) 和配套的 DMD EVM 配合使用。此外，本指南还概述了 VHDL 代码和实现。

## 内容

<b>1 引言</b> .....	4
1.1 欢迎.....	4
<b>2 概述</b> .....	5
2.1 用途.....	5
2.2 Apps FPGA 硬件目标.....	5
<b>3 接口</b> .....	6
3.1 连接 DLPC910 的 LVDS 高速数据接口.....	6
3.1.1 DLP9000X 和 DLP9000XUV.....	6
3.1.2 DLP6500.....	6
3.2 输出至 DLPC910 的数据加载控制信号.....	6
3.3 DLPC910 的 DMD 复位和块清除信号.....	7
3.4 DLPC910 初始化和控制器复位信号.....	7
3.5 Apps FPGA 复位信号 - apps_resetz .....	7
3.6 DLPC910 状态信息信号.....	7
3.7 USB GPIF ( 接口 ) .....	8
3.7.1 Apps FPGA 寄存器地址读取/写入事务.....	8
3.7.2 FIFO 写入事务.....	11
3.8 DLPLCRC910EVM DIP 开关 (SW2).....	11
3.9 VC-707 DIP 开关 (SW2).....	12
3.10 VC-707 按钮开关.....	12
3.11 VC-707 状态 LED.....	12
3.12 DLPLCRC910EVM Apps FPGA 测试点.....	13
<b>4 操作</b> .....	14
4.1 初始化.....	14
4.1.1 初始化提示.....	14
4.1.2 初始化例程.....	15
4.1.3 GPIO 状态 LED.....	15
4.1.4 错误.....	15
4.2 测试图形发生器 (TPG) 和应用加载程序 - DLP 控制.....	15
4.2.1 测试图形发生器 (TPG).....	16
4.2.2 DMD 数据缓冲区.....	16
4.2.3 DMD 加载状态机.....	16
4.2.4 DMD 复位状态机.....	18
4.2.5 DMD 加载参数.....	18
4.2.6 同步脉冲.....	18
4.3 用户 DLP 控制.....	18
4.3.1 DLP6500 ( 1920 × 1080 ) 用户图像显示示例 ( 全局 ) .....	19
4.3.2 DLP9000X ( 2560 × 1600 ) 用户图像显示示例 ( 全局 ) .....	20
4.3.3 Load4 - 与 DLP6500 DMD 一起使用.....	20
4.3.4 USB GPIF FIFO 数据写入.....	20
4.3.5 外部触发.....	20

4.4 USB GPIF (运行)	21
4.5 时钟和复位	21
4.5.1 参考时钟	21
4.5.2 Clk50 和 Clk100	21
4.5.3 DLP 时钟	21
4.5.4 USB GPIF 时钟	21
4.5.5 逻辑复位	21
4.5.6 时钟域交叉 (CDC)	21
4.6 开关去抖	22
<b>5 USB GPIF 寄存器</b>	<b>23</b>
5.1 寄存器定义	23
5.1.1 状态 (0x000C)	23
5.1.2 数据加载控制 (0x0010)	24
5.1.3 测试图形控制 (0x0014)	24
5.1.4 测试行地址 (0x0018) - [未使用]	25
5.1.5 加载程序复位类型 (0x001C)	25
5.1.6 类型和版本 (0x0020)	25
5.1.7 用户图像缓冲区写入设置 (0x0024)	25
5.1.8 USB GPIF FIFO 读取突发大小 (0x0028) - [已淘汰]	26
5.1.9 用户行命令寄存器 (0x002C)	26
5.1.10 用户块命令寄存器 (0x0030)	26
5.1.11 加载程序行控制 (0x0034)	26
5.1.12 加载程序加载间隔 (0x0038)	27
5.1.13 加载程序曝光时间 (0x003C)	27
5.1.14 地址写入 (0x003F) - [未使用]	27
5.1.15 加载程序控制 (0x0040)	27
5.1.16 Park [PWR_FLOAT] (0x0044)	27
5.1.17 外部触发状态 (0x0048)	28
5.1.18 FPGA 编译日期 (0x0080)	28
5.1.19 主要 - 次要版本 (0x0084)	28
5.1.20 固定值 FPGA 标识符 (0x0088)	28
5.1.21 测试寄存器 (0x008C)	28
<b>6 FPGA 配置</b>	<b>29</b>
<b>7 Apps FPGA 源文件和编译</b>	<b>30</b>
7.1 设计工具	30
7.2 源文件	30
7.2.1 主要 VHDL 和 IP 模块	30
7.2.2 具有多个实例的模块	31
7.2.3 VHDL 软件包	31
7.2.4 Vivado 限制条件	31
7.2.5 存储器 IP 初始化文件	31
7.3 构建 Apps FPGA 代码	31
7.3.1 源代码	31
7.3.2 创建 Vivado 工程	32
7.3.3 编译设计	32
7.3.4 仿真	32
<b>8 德州仪器 (TI) 相关文档</b>	<b>34</b>
<b>9 附录</b>	<b>34</b>
9.1 缩略语和首字母缩写词	34
9.2 注意事项和警告信息	35

## 插图清单

图 2-1. Apps FPGA 硬件目标	5
图 3-1. 寄存器地址事务时序图	9
图 3-2. 寄存器数据写入事务时序图	9
图 3-3. 寄存器数据读取事务时序图	10
图 3-4. FIFO 写入事务时序图	11
图 4-1. Apps FPGA 功能方框图	14

图 4-2. 应用加载程序数据流.....	15
图 4-3. DMD 加载状态机.....	17
图 7-1. 源文件.....	30
图 7-2. 测试台.....	32
图 7-3. 设置对话框.....	33
图 7-4. Vivado 波形窗口.....	33

## 表格清单

表 3-1. Apps FPGA - LVDS 数据接口输出信号名称.....	6
表 3-2. 数据加载控制信号.....	6
表 3-3. 复位和块清零信号.....	7
表 3-4. DLPC910 初始化和控制器复位信号.....	7
表 3-5. DLPC910 状态信息信号.....	7
表 3-6. USB GPIF 接口信号.....	8
表 3-7. USB GPIF 事务类型定义.....	8
表 3-8. 寄存器地址事务示例.....	9
表 3-9. 寄存器数据写入事务示例.....	10
表 3-10. 寄存器数据读取事务示例.....	10
表 3-11. DLPLCRC910EVM DIP 开关 (SW2).....	11
表 3-12. VC-707 DIP 开关 (SW2).....	12
表 3-13. VC-707 按钮开关.....	12
表 3-14. VC-707 状态 LED.....	12
表 3-15. DLPLCRC910EVM Apps FPGA 测试点.....	13

## 商标

Virtex™ and Vivado™ are trademarks of AMD.

所有商标均为其各自所有者的财产。

## 1 引言

DLPC910 Apps FPGA 代码提供了连接 DLPC910 控制器和配套的 DMD 的功能和示例。DLPC910 Apps FPGA 指南详细介绍了 DLPC910 应用 FPGA (Apps FPGA) 的功能和寄存器及其构建所用 VHDL 代码的组织结构。

### 1.1 欢迎

通过 DLP LightCrafter DLPC910 EVM (DLPLCRC910EVM)，可以很好地评估 DLP 芯片产品系列提供的最高带宽数据速率。设计人员可以通过全局、四块、双块和单块模式对所有 DMD 微镜进行像素级精确控制，从而为连续（灯）或固态（开关）照明应用定制 DMD 微镜图形时序。

DLPC910 Apps FPGA 实现支持探索 DLPLCRC910EVM 和配套的 DMD（例如 DLP6500、DLP9000X 和 DLP9000XUV）的功能

通过 DLPLCRC910EVM，客户还可以使用高性能 DLP9000XUV 芯片进行如下系统设计：

- 平版印刷
- 3D 打印和增材制造
- 动态灰度打标和编码
- 工业印刷
- 结构光，例如：
  - 工厂自动化和 3D 机器视觉
  - 3D 在线自动光学检测 (AOI)
  - 机器人视觉
  - 离线 3D 计量
  - 3D 扫描仪
  - 3D 识别和生物识别
- 医疗与生命科学
- 高速成像和显示

## 2 概述

### 2.1 用途

本文档介绍了 DLPC910 应用 FPGA (Apps FPGA) 的功能和寄存器，还概述了 VHDL 代码。

### 2.2 Apps FPGA 硬件目标

Apps FPGA 参考代码专用于搭载在 AMD Xilinx VC-707 评估板上的 AMD Xilinx Virtex™-7 FPGA。如 [Apps FPGA 硬件目标](#) 所示，VC-707 评估板连接到德州仪器 (TI) DLPC910 评估模块 (DLPLCRC910EVM)，该模块连接到三块可用的 TI DMD 板之一。

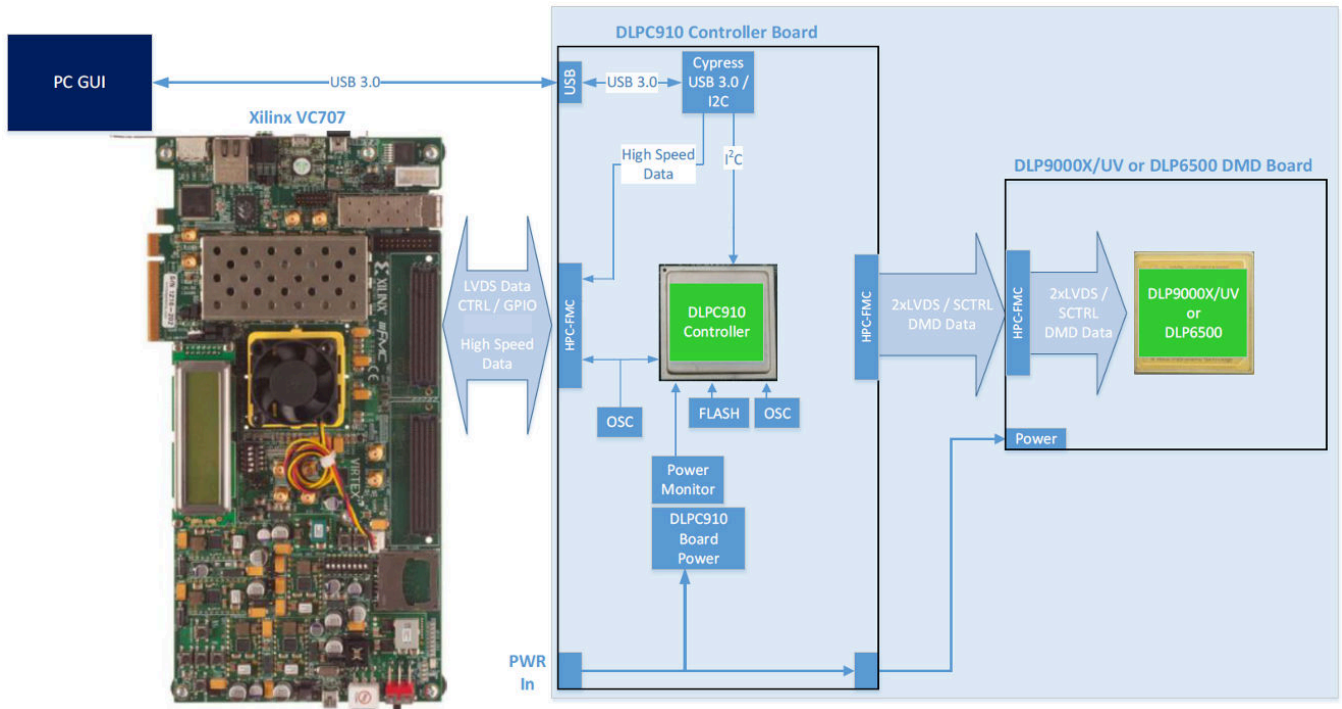


图 2-1. Apps FPGA 硬件目标

### 3 接口

本节介绍了 Apps FPGA 接口信号。Apps FPGA 通过两个高引脚数 FMC 连接器 ( DLPLC910EVM 板上的 J500 和 J501 ) 连接到 DLPC910 EVM 板。除了 DLPC910 接口外, Apps FPGA 还使用 VC-707 评估板上的按钮开关、DIP 开关和 LED。

Apps FPGA 接口信号引脚编号和信号属性在随 Apps FPGA 源代码提供的输入/输出 (I/O) 物理限制条件 (.xdc) 文件中进行了说明。

本节中使用的信号名称与 Apps FPGA 顶层 VHDL 模块中使用的信号名称相匹配。

#### 3.1 连接 DLPC910 的 LVDS 高速数据接口

Apps FPGA 通过 VC-707 FMC HPC1 连接器将图像数据驱动至 DLPC910, 该连接器由 4 条 16 个数据对 LVDS 总线组成。四条总线中的每条总线还具有自己的时钟和数据有效信号对。信号电平、时序和数据映射详细信息可在 DLPC910 数据表中找到。

[Apps FPGA - LVDS 数据接口输出信号名称](#) 中总结了至 DLPC910 的 LVDS 数据接口的 Apps FPGA 输出信号名称。

表 3-1. Apps FPGA - LVDS 数据接口输出信号名称

	LVDS 总线 A	LVDS 总线 B	LVDS 总线 C	LVDS 总线 D
数据	dout_ap(15:0) dout_an(15:0)	dout_bp(15:0) dout_bn(15:0)	dout_cp(15:0) dout_cn(15:0)	dout_dp(15:0) dout_dn(15:0)
时钟	dclk_ap dclk_an	dclk_bp dclk_bn	dclk_cp dclk_cn	dclk_dp dclk_dn
数据有效	dvalid_ap dvalid_an	dvalid_bp dvalid_bn	dvalid_cp dvalid_cn	dvalid_dp dvalid_dn

##### 3.1.1 DLP9000X 和 DLP9000XUV

当使用 DLP9000X 或 DLP9000XUV DMD 时, Apps FPGA 通过所有四条 ( A、B、C、D ) LVDS 数据总线将数据发送到 DLPC910。启动时, Apps FPGA 使用来自 DLPLC910EVM 的 `dmd_type(3:0)` 输入确定连接的 DMD 类型, 并驱动适当数量的 LVDS 总线。此外, 在启动时, Apps FPGA 会根据来自 DLPLC910EVM 的 `dmd_speed_sel(1:0)` 输入将接口时钟速度设置为 400MHz 或 480MHz。

#### 备注

尽管 DLP9000X 和 DLP9000XUV 以 400MHz 的频率运行, 但仅 480MHz 运行进行了充分验证。

##### 3.1.2 DLP6500

当使用 DLP6500 DMD 时, Apps FPGA 通过 LVDS 数据总线 A 和 B 将数据发送到 DLPC910。Apps FPGA 使用 `dmd_type(3:0)` 输入确定连接的 DMD。对于 DLP6500 DMD, 时钟频率固定为 400MHz。

#### 3.2 输出至 DLPC910 的数据加载控制信号

[表 3-2](#) 中列出的数据加载控制信号通过 VC-707 FMC 连接器输出至 DLPC910。DLPC910 数据表中介绍了信号功能。

表 3-2. 数据加载控制信号

名称	Apps FPGA I/O	功能
rowmd(1:0)	out	DMD 行模式
rowad(10:0)	out	DMD 行地址
comp_data	out	补码数据
load4z	out	DMD Load4 功能使能 ( 低电平有效 )
ns_flip	out	DMD 上的顶部/底部图像翻转

### 3.3 DLPC910 的 DMD 复位和块清除信号

表 3-3 列出了与 DLPC910 之间的 DMD 复位和块清零信号。此类信号通过 VC-707 FMC 连接器连接。有关更多信息，请参阅 DLPC910 数据表。

表 3-3. 复位和块清零信号

名称	Apps FPGA I/O	功能
b1kad(3:0)	out	块地址
b1kmd(1:0)	out	块模式
rst2b1kz	out	双块和四块控制
wdt_enablez	out	DMD 复位脉冲看门狗计时器使能
rst_active	in	正在进行 DMD 显微镜时钟脉冲 (MCP)

### 3.4 DLPC910 初始化和控制器复位信号

表 3-4 描述了 DLPC910 控制器的初始化信号。有关其他信号信息，请参阅 DLPC910 数据表。

表 3-4. DLPC910 初始化和控制器复位信号

名称	Apps FPGA I/O	功能
ctrl_rstz	out	DLPC910 控制器复位
pwr_floatz	out	DLPC910 PWR_FLOAT
ecp2_finished	in	从 SPI 闪存状态进行 DLPC910 配置
init_active	in	DLPC910 初始化运行状态

### 3.5 Apps FPGA 复位信号 - apps\_resetz

信号 `apps_resetz` 源自 DLPLC910EVM 板上的按钮开关 (SW1 APP RST)。按下开关后，`apps_resetz` 信号变为低电平，导致 Apps FPGA 中的 `init-run-park` 状态机将 `pwr_floatz` 驱动为低电平来停止 DMD 显微镜。释放按钮开关后，`init-run-park` 状态机会重新初始化 DLPC910 控制器并复位 Apps FPGA 逻辑。

### 3.6 DLPC910 状态信息信号

表 3-5 中列出了 DLPC910 状态信息信号。`ddc_version`、`DMD_type` 和 `DMD_irq` 信号来自 DLPC910 控制器。Apps FPGA 逻辑仅使用 `DMD_type`。版本和 `irq` 信号可通过 USB GPIF 在状态寄存器中获取。

`DMD_speed_sel` 信号来自 DLPLC910EVM 板上的一组跳线。DLPC910 控制器和 Apps FPGA 使用跳线设置来确定 LVDS 高速接口的时钟频率 [DLP9000X](#) 和 [DLP9000XUV](#)。

有关更多信息，请参阅 DLPC910 数据表。

表 3-5. DLPC910 状态信息信号

名称	Apps FPGA I/O	功能
ddc_version(2:0)	in	DLPC910 固件版本
dmd_type(3:0)	in	DLPC910 DMD 类型
dmd_speed_sel(1:0)	in	Apps/DLPC910 LVDS 速度选择跳线
dmd_irq	in	DMD irq 状态信号

### 3.7 USB GPIF (接口)

USB GPIF 端口可连接 Apps FPGA 内部控制状态寄存器。USB GPIF 端口还提供数据输入 FIFO，用于加载用户图像缓冲区和用户图像进行显示。表 3-6 介绍了接口中的信号。

表 3-6. USB GPIF 接口信号

名称	Apps FPGA I/O	功能
gpif_addr(8:0)	in	未使用
usb_fd(15:0)	in/out	USB GPIF 数据，双向
usb_ctr1(5:3)	in	USB GPIF 控制 - 未使用 <sup>(1)</sup>
usb_ctr1(2:0)	in	USB GPIF 控制，识别事务类型
usb_rdy(2:1)	out	USB GPIF 就绪输出 - 未使用 <sup>(2)</sup>
usb_rdy(0)	out	USB GPIF 就绪输出 0 <sup>(3)</sup>
usb_reset	out	未使用 - Apps FPGA 始终驱动低电平
usb_if_clock	in	USB GPIF 时钟，48MHz

(1) 未使用 usb\_ctr1(5:3) 输入。

(2) usb\_rdy(2:1) 输出始终驱动为低电平。

(3) usb\_rdy(0) 是 USB GPIF 输入 FIFO 的空标志。

GPIF 事务类型由 usb\_ctr1(2:0) 定义，如表 3-7 所示。

表 3-7. USB GPIF 事务类型定义

信号	空闲	地址	数据写入	数据读取	FIFO 突发
usb_ctr1(2)	1	0	0	0	1
usb_ctr1(1)	1	1	1	0	1
usb_ctr1(0)	1	1	0	1	0

#### 3.7.1 Apps FPGA 寄存器地址读取/写入事务

读取或写入 Apps FPGA 寄存器需要首先发送寄存器地址，然后发送寄存器数据。寄存器地址和数据的宽度为 32 位。传输 32 位需要两个 16 位总线事务。

#### 备注

在 usb\_if\_clk 的下降沿进行数据采样。



### 3.7.1.1 Apps FPGA 寄存器地址事务

寄存器地址事务时序图显示了寄存器地址事务时序。

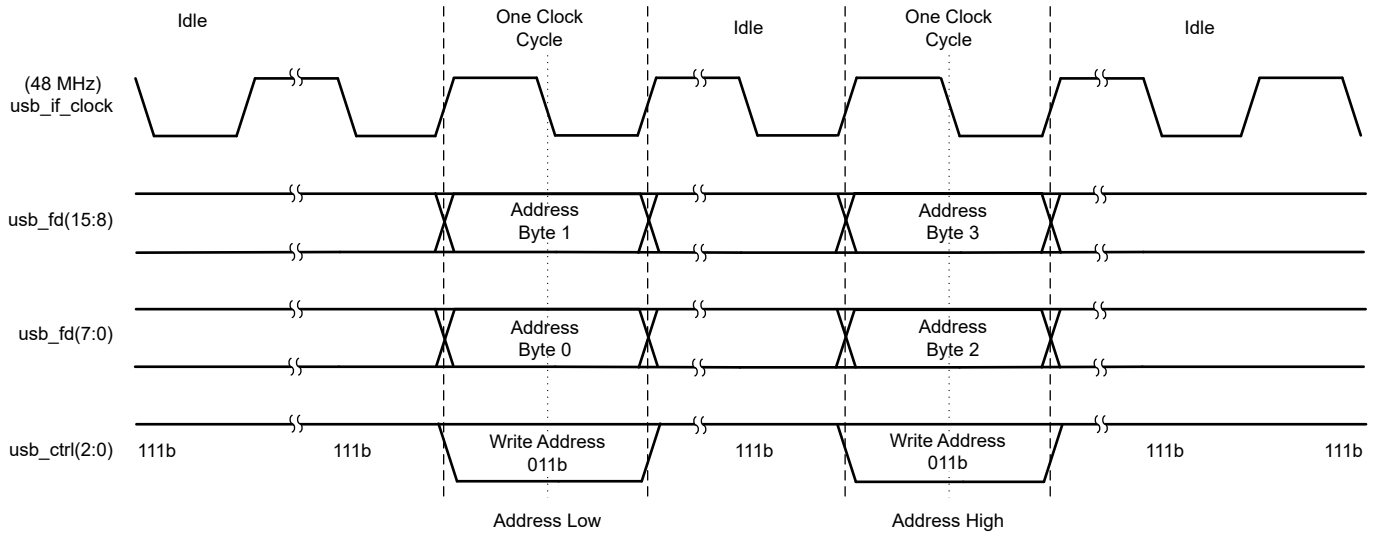


图 3-1. 寄存器地址事务时序图

寄存器地址事务示例显示了寄存器地址事务示例。地址传输之间可以有多个空闲时钟周期。地址事务表显示了 0x00010228 的地址传输。

表 3-8. 寄存器地址事务示例

信号	空闲	写入低地址	空闲	写入高地址	空闲
usb_if_clock		1 个时钟周期		1 个时钟周期	
usb_fd(15:8)	0x00	0x02	0x00	0x00	0x00
usb_fd(7:0)	0x00	0x28	0x00	0x01	0x00
usb_ctr1(2:0)	空闲	地址 - "011"	空闲	地址 - "011"	空闲

### 3.7.1.2 Apps FPGA 寄存器数据写入事务

寄存器数据写入事务时序图显示了寄存器数据写入事务时序。

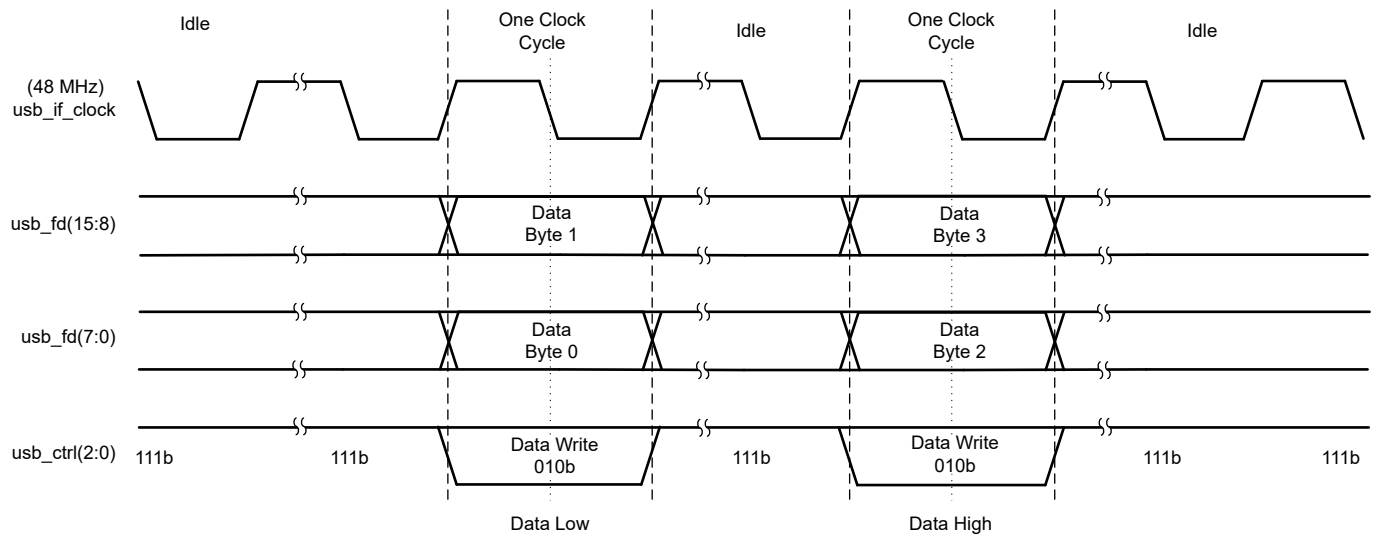


图 3-2. 寄存器数据写入事务时序图

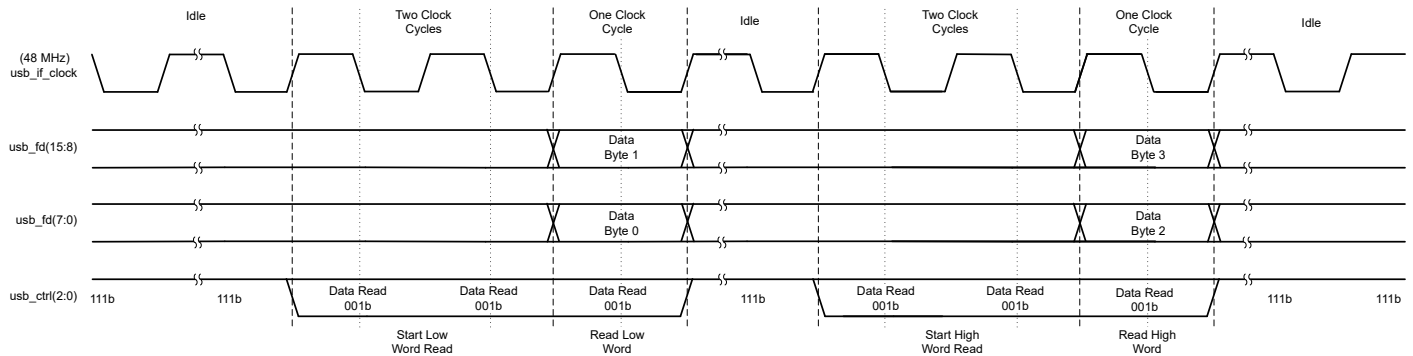
寄存器数据写入事务示例显示了寄存器数据写入事务示例。正在传输寄存器写入数据值 0x04030201。

**表 3-9. 寄存器数据写入事务示例**

信号	空闲	写入低位字	空闲	写入高位字	空闲
usb_if_clock		1 个时钟周期		1 个时钟周期	
usb_fd(15:8)	0x00	0x02	0x00	0x04	0x00
usb_fd(7:0)	0x00	0x01	0x00	0x03	0x00
usb_ctrl(2:0)	空闲	数据写入 - "010"	空闲	数据写入 - "010"	空闲

### 3.7.1.3 Apps FPGA 寄存器数据读取事务

寄存器读取字事务长度为 3 个时钟。寄存器数据读取事务时序图显示了寄存器数据读取事务时序。



**图 3-3. 寄存器数据读取事务时序图**

Apps FPGA GPIF 支持从先前写入的寄存器地址进行多个寄存器读取事务，而无需新的地址事务。这样可以降低轮询状态寄存器时的时间开销。

寄存器数据读取事务示例显示了寄存器读取字事务示例并对寄存器值 0x04030201 的读取进行了说明。

**表 3-10. 寄存器数据读取事务示例**

信号	空闲	开始低位字读取	读取低位字	空闲	开始高位字读取	读取高位字	空闲
usb_if_clock		2 个时钟周期	1 个时钟周期		2 个时钟周期	1 个时钟周期	
usb_fd(15:8)	0x00	--	0x02	0x00	--	0x04	0x00
usb_fd(7:0)	0x00	--	0x01	0x00	--	0x03	0x00
usb_ctrl(2:0)	空闲	数据读取 - "001"	数据读取 - "001"	空闲	数据读取 - "001"	数据读取 - "001"	空闲

### 3.7.2 FIFO 写入事务

USB GPIF FIFO 用于将图像数据从 USB GPIF 传输到 Apps FPGA 中的用户图像缓冲区。FIFO 大小为 256 个字，每个字为 16 位。FIFO 写入事务的长度始终为 256 个字 (512 字节)。

FIFO 写入事务时序图显示了 FIFO 写入事务时序。

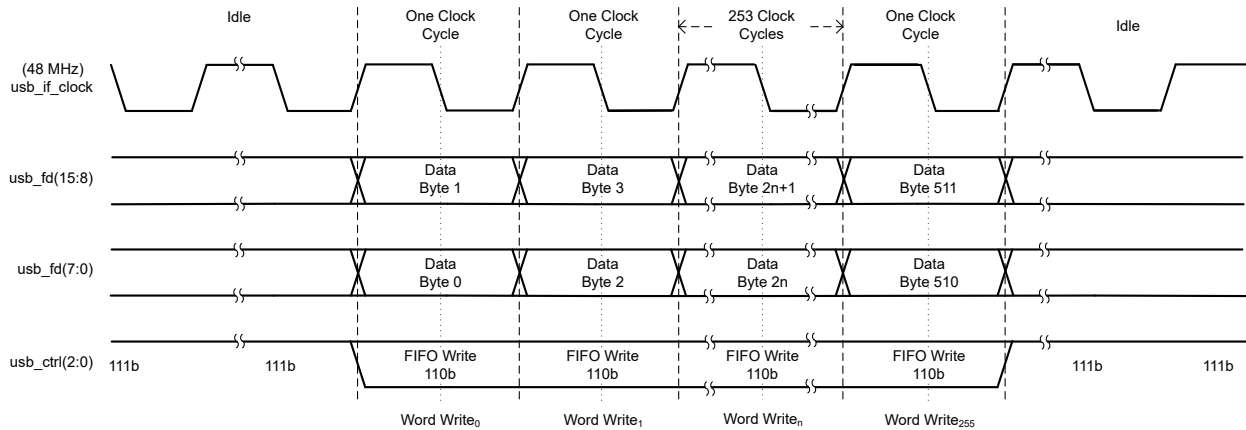


图 3-4. FIFO 写入事务时序图

### 3.8 DLPLCRC910EVM DIP 开关 (SW2)

DLPLCRC910EVM 装有一个 DIP 开关 (SW2)，其开关设置由 Apps FPGA 使用，如表 3-11 中所述。

#### 备注

连接到 DLPLCRC910EVM SW2 的输入在处于 "OFF" 位置 [逻辑 1] 时，通过上拉电阻器拉至高电平，在处于 "ON" 位置 [逻辑 0] 时，拉至低电平。当开关处于 "OFF" [逻辑 1] 状态时，位置 0 和 1 不启用；当开关处于 "ON" [逻辑 0] 状态时，位置 2、3 和 7 不启用。

表 3-11. DLPLCRC910EVM DIP 开关 (SW2)

信号	开关标签	功能	默认位置 <sup>(1)</sup>
evm_in_dip_sw(7)	8	wdt_enblz <sup>(2)</sup>	ON [逻辑 0]
evm_in_dip_sw(6)	7	未使用	ON [逻辑 0]
evm_in_dip_sw(5)	6	未使用	ON [逻辑 0]
evm_in_dip_sw(4)	5	未使用	ON [逻辑 0]
evm_in_dip_sw(3)	4	ns_flip <sup>(2)</sup>	ON [逻辑 0]
evm_in_dip_sw(2)	3	comp_data <sup>(2)</sup>	ON [逻辑 0]
evm_in_dip_sw(1)	2	load4_enz <sup>(2)</sup>	OFF [逻辑 1]
evm_in_dip_sw(0)	1	pwr_float <sup>(3)</sup>	OFF [逻辑 1]

(1) 默认位置如物理开关上所示。

(2) 指示 Apps FPGA 执行 DLPC910 数据表中所述的函数。此类函数在寄存器 0x0010 中复制 ( [数据加载控制 \(0x0010\)](#) 章节 )。

(3) 指示 Apps FPGA 向 DLPC910 发出 pwr\_floatz。此函数在寄存器 0x0044 中复制 ( [Park \[PWR\\_FLOAT\] \(0x0044\)](#) 章节 )。

### 3.9 VC-707 DIP 开关 (SW2)

VC-707 EVM 板装有一个 DIP 开关 (SW2 GPIO DIP SW)，其开关设置由 Apps FPGA 使用，如表 3-12 中所述。正常运行时，将位置 0 设置为 ON，将所有其他开关设置为 OFF。

表 3-12. VC-707 DIP 开关 (SW2)

信号	开关标签	功能	默认位置
vc707_in_dip_sw(7)	8	未使用	关闭
vc707_in_dip_sw(6)	7	未使用	关闭
vc707_in_dip_sw(5)	6	alt_rows_sel <sup>(1)</sup>	关闭
vc707_in_dip_sw(4)	5	未使用	关闭
vc707_in_dip_sw(3)	4	未使用	关闭
vc707_in_dip_sw(2)	3	未使用	关闭
vc707_in_dip_sw(1)	2	未使用	关闭
vc707_in_dip_sw(0)	1	USB GPIF FIFO 输入字交换 <sup>(2)</sup>	打开

(1) 当位置 6 开关为 ON 时，在测试图形缓冲区之后插入水平线图形（偶数行全为 0，奇数行全为 1）。仅用于调试。

(2) 当位置 1 开关为 ON 时，在写入 USB GPIF FIFO 之前、偶数输入的 16 位字将与奇数输入的 16 位字交换。DLPLC910EVM 的实现需要启用该设置 (ON)。

### 3.10 VC-707 按钮开关

VC-707 EVM 装有按钮开关，其开关输入由 Apps FPGA 使用，如表 3-13 中所述。

表 3-13. VC-707 按钮开关

信号	开关标签	功能
vc707_in_pb_sw(7)	SW7	未使用
vc707_in_pb_sw(6)	SW6	未使用
vc707_in_pb_sw(5)	SW5	去抖外部触发器测试 <sup>(1)</sup>
vc707_in_pb_sw(4)	SW4	init-run-park 复位（仅用于调试）
vc707_in_pb_sw(3)	SW3	测试图形循环 <sup>(2)</sup>

(1) VC-707 板上的去抖 SW5 可用作 Apps FPGA 测试接头（DLPLC910EVM 板上的 J3 引脚 3）上的测试外部触发器。要使用 SW5 测试外部触发，请将 J3 引脚 2 短接至 J3 引脚 3。

(2) 当 Apps FPGA 启动时，测试图形会发送到 DLPC910，在 DMD 上显示。测试图形会定期（当前默认值为两秒）更改。按下 SW3 可停止测试图形循环。随后按下 SW3 会更改为下一个测试图形。按下 DLPLC910EVM 板上的 Apps FPGA 复位开关可结束受 SW3 控制的图形时序。

### 3.11 VC-707 状态 LED

Apps FPGA 使用 VC-707 GPIO LED 指示初始化和工作模式，如表 3-14 所述。

表 3-14. VC-707 状态 LED

信号	LED 标签	功能
vc707_led_ds(9)	7	ON = dmd_type 为 0xF，DLP9000X 或 DLP9000XUV
vc707_led_ds(8)	6	ON = dmd_type 为 0xE，DLP6500
vc707_led_ds(7)	5	备用
vc707_led_ds(6)	4	备用
vc707_led_ds(5)	3	ON = USB GPIF 时钟 PLL 被锁定 <sup>(1)</sup>
vc707_led_ds(4)	2	ON = pwr_floatz 置为有效，系统已停止
vc707_led_ds(3)	1	ON = 初始化完成，系统正在运行
vc707_led_ds(2)	0	ON = 初始化错误 <sup>(2)</sup>

(1) 当 USB GPIF PLL 锁定到来自 Infineon - Cypress USB 接口芯片的输入时钟 usb\_if\_clk，GPIO LED 3 会亮起。

(2) 当 DLPC910 初始化期间发生异常时，init-run-park 状态机点亮 GPIO LED 0。

### 3.12 DLPLCRC910EVM Apps FPGA 测试点

DLPLCRC910EVM 装有一个 DIP 开关 (SW2)，其开关设置由 Apps FPGA 使用，如表 3-15 中所述。

**表 3-15. DLPLCRC910EVM Apps FPGA 测试点**

信号	J3 引脚	Apps FPGA 信号
APP_TSTPT7	2	用于用户控制的外部全局复位触发器输入
APP_TSTPT6	3	去抖 VC707 按钮开关 SW5 输出
APP_TSTPT5	4	应用加载程序数据使能，输出
APP_TSTPT4	5	应用加载程序加载繁忙，输出
APP_TSTPT3	6	应用加载程序微镜稳定繁忙，输出
APP_TSTPT2	7	从应用加载程序曝光触发器，输出
APP_TSTPT1	8	应用加载程序有限状态机 (FSM) 微镜复位繁忙，输出
APP_TSTPT0	9	有效复位信号，输出 <sup>(1)</sup>

(1) 在微镜悬空操作期间，有效复位信号不会变为高电平

## 4 操作

图 4-1 显示了 Apps FPGA 的功能方框图。叠加颜色突出显示了执行每组函数的 VHDL 模块，其中 VHDL 模块名称采用斜体。

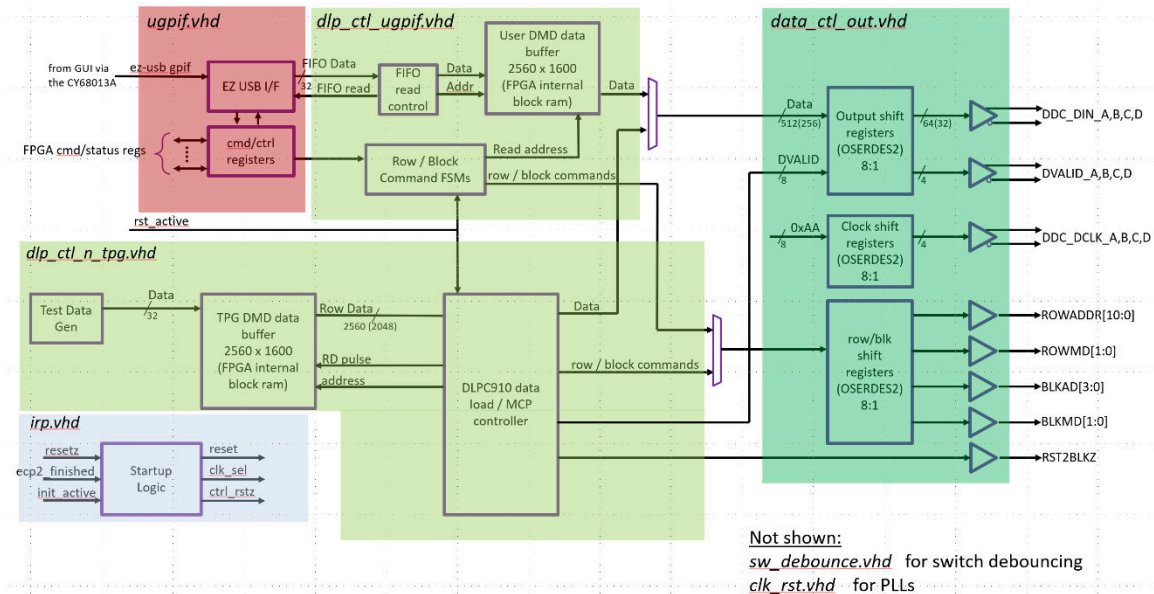


图 4-1. Apps FPGA 功能方框图

Apps FPGA 具有三种运行状态：

1. 正在初始化
2. 初始化错误
3. 运行
  - a. 测试图形/Apps DLP 加载控制
  - b. 用户 DLP 加载控制

Apps FPGA 执行初始化并响应初始化错误，如[加载程序控制 \(0x0040\)](#) 章节所述。初始化完成后，运行状态就开始了。在运行状态下，有两种独立的 DLP 控制方法，可通过加载程序控制寄存器的零位来选择 ([加载程序控制 \(0x0040\)](#))。第一种方法是进行 DLP 组件集的测试图形 Apps FPGA 加载程序控制。第二种方法是当 Apps FPGA 加载程序不受控制时，可以通过 USB GPIF 接口 (用户控制模式) 控制 DLP 组件集。

### 4.1 初始化

Apps FPGA 根据 DLPC910 数据表要求提供 DLPC910 和所附的 DMD 芯片组初始化。VHDL 模块 *IRP.vhd* (init-run-park 控制) 和子模块 *IRP\_FSM.vhd* (init-run-park 状态机) 提供了该启动逻辑。

#### 4.1.1 初始化提示

以下任何事件都会导致 init-run-park 状态机执行初始化例程：

1. 完成 Apps FPGA 配置。
2. 按下 DLPLC910EVM 上的 SW1 APP RST 开关 (apps\_resetz 至 Apps FPGA)。
3. 通过 DLPLC910EVM 上的 DIP 开关 SW2 的位置 1 从 ON 转换为 OFF (使用 DIP 开关解除停止系统)。
4. USB GPIF 寄存器 0x0044 中的 Park 位从 '1' 转换为 '0' (例如，使用寄存器命令位解除停止系统)。

#### 4.1.2 初始化例程

VC-707 执行的初始化例程包括以下基本步骤：

1. 向 DLPC910 控制器发出复位 (`ctrl_rstz`)。
2. 确认 DLPC910 已配置 (`ecp2_finished`)。
3. 确保 Apps FPGA DLP PLL 已锁定 (`dclk_ap/n`)。
4. 采集选定的 DLPC910 接口时钟速度 (`dmd_speed_sel`)。
5. 根据从 `dmd_speed_sel` 中读取的值选择 400MHz 或 480MHz 时钟。
6. 对 AMD/Xilinx oserdes 输出基元进行复位。
7. 确保 DLPC910 开始初始化 (`init_active`)。
8. 开启 DLPC910 训练模式。
9. 释放复位到 DLPC910 (`ctrl_rstz`)。
10. 等待训练完成 (`init_active low`)。
11. 采集 DMD 类型 (`dmd_type`)。
12. 对 Apps FPGA 控制逻辑进行复位 (自动)。
13. 运行。

成功完成初始化后, `init-run-park` 状态机进入运行状态。然后, 可监测 USB GPIF 暂存寄存器、开关 SW1 APP RST 和 SW2 的位置 1。如果这些输入中的任何一个指示需要停止微镜, 则发出 `pwr_floatz`, 状态机等待这些输入返回到运行状态, 然后再次运行初始化例程。

#### 备注

若要对 DLPLCRC910EVM 上的 `dmd_speed_sel` 跳线进行更改, 则需要重新运行初始化 (按 SW1 APP RST 开关)。

#### 4.1.3 GPIO 状态 LED

三个 VC-707 板 GPIO LED 用于显示 `init-run-park` 状态：

- 处于运行状态时, GPIO LED 1 亮起
- 当 DMD 微镜停止时, GPIO LED 2 亮起
- 出现初始化错误时, GPIO LED 0 亮起

#### 4.1.4 错误

如果发生以下任一情况, 则会发出初始化错误标志：

1. DLPC910 不会发出 `ecp2_finished`。
2. Apps FPGA DLP PLL 未锁定。
3. DLPC910 `init_active` 特性不符合预期。
4. 所选时钟速度与 DMD 类型不匹配。

一旦发出初始化错误标志, `init-run-park` 状态机在复位 (SW2 APP RST) 前将保持错误状态, 在复位时尝试另一次初始化。

## 4.2 测试图形发生器 (TPG) 和应用加载程序 - DLP 控制

当应用加载程序控制 DLP 组件集时, 基本数据流如应用加载程序数据流所示。TPG、数据缓冲区和应用加载程序控制功能均包含在 VHDL 模块 `dlp_ctl_n_tpg.vhd` 中。

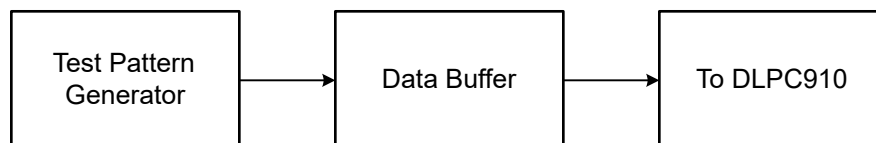


图 4-2. 应用加载程序数据流

### 4.2.1 测试图形发生器 (TPG)

测试图形生成由 VHDL 模块 *tpg.vhd* 以及子模块 *tpg\_trig.vhd* 和 *tpg\_timing.vhd* 提供。VHDL 软件包 *tpg\_pkg.vhd* 为各种测试图形提供测试数据常量。

Apps FPGA 初始化完成后，会定期连续循环 16 个测试图形 ( [测试图形控制 \(0x0014\)](#) 章节 )。修改测试图形控制寄存器的循环间隔字段 ( [测试图形控制 \(0x0014\)](#) ) 来更改循环速率。测试图形控制寄存器还提供控制位来禁用图形循环并持续显示所选的测试图形。

当测试图形在初始化后循环时，按下 VC-707 板上的瞬时按钮开关 SW3 可停止图形循环。连续激活 SW3 将选择序列中的下一个测试图形。按下 DLPLC910EVM 板上的 Apps FPGA 复位开关会重新初始化 Apps FPGA 和 DLP 组件集并返回自动图形循环。

测试图形发生器提供控制和寻址功能，可将测试图形写入 DMD 数据缓冲区。在 DMD 加载状态机的控制下，从数据缓冲区读取图形并将其发送到 DLPC910。

### 4.2.2 DMD 数据缓冲区

DMD 数据缓冲区由 512x1600x1 双端口 RAM 的五个实例组成。双端口 RAM 使用 AMD Xilinx 块存储器生成器工具创建，可从 AMD Vivado™ GUI 中的 IP 目录访问。VHDL 模块 *dmd\_data\_buffer.vhd* 提供双端口数据缓冲功能。

DLP6500 仅使用五个实例中的四个。

### 4.2.3 DMD 加载状态机

VHDL 模块 *dmd\_load\_fsm.vhd* 执行 DLPC910 测试图形数据加载函数 ( “加载程序” 功能 )。加载程序从 DMD 数据缓冲区读取测试图形数据，对数据进行格式化，并将数据发送到 *data\_ctl\_out.vhd* 模块从而进行序列化并缓冲到 DLPC910。此外，还会创建相应的行地址、行命令和 DVALID 控制信号，从而与数据一起发送。

默认情况下，加载程序会加载测试图形图像，从而响应来自 *dload\_trigger.vhd* 模块的加载触发脉冲。触发脉冲的默认周期为 400 μs。加载程序加载间隔寄存器提供了更改触发周期的功能。 [节 5.1.12](#)。



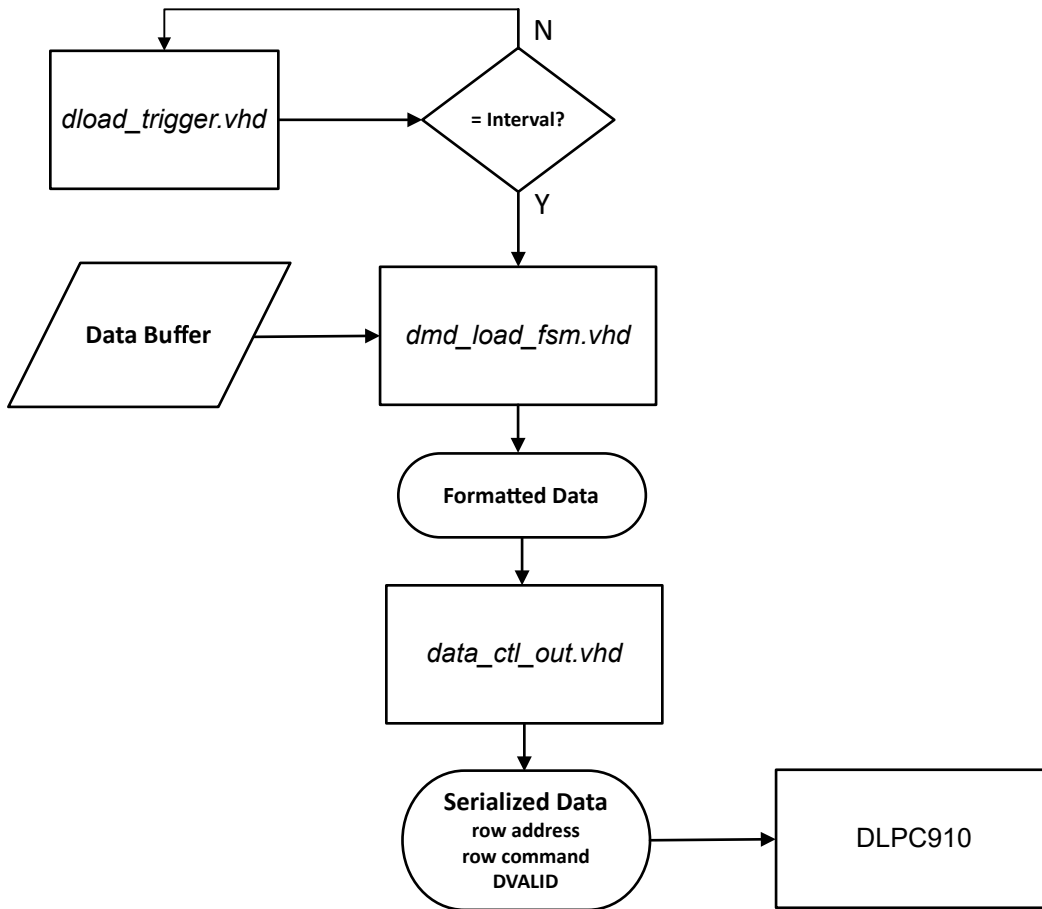


图 4-3. DMD 加载状态机

当复位类型为全局或四块分步时，加载程序可以进入自由运行模式，此时会忽略加载触发，加载程序会尽可能快地连续加载图像。自由运行模式演示了最短 DLP 组件集图像加载时间。[加载程序控制 \(0x0040\)](#)

以下加载程序控制设置可通过 USB GPIF 寄存器进行配置：

1. 复位类型。默认为 `global`。请参阅[加载程序复位类型 \(0x001C\)](#)。
2. 起始 DMD 行和要加载的 DMD 行数。默认为第 0 行和器件满载。请参阅[加载程序行控制 \(0x0034\)](#)。
3. 曝光时间。缺省为 0。请参阅[加载程序曝光时间 \(0x003C\)](#)。
4. 数据加载控制。默认值请参阅[数据加载控制 \(0x0010\)](#)。

#### 备注

为避免在加载程序处于自由运行模式时出现不可预测的结果，请勿更改上述加载程序控制设置。要查看在自由运行模式下更改上述设置的效果，请退出自由运行模式，修改设置，然后重新进入自由运行模式。

加载完整图像或加载单个微镜块后，加载程序会向相控 DMD 复位状态机发送微镜复位请求。加载程序使用 DLP6500 和 DLP9000X 查找表（可按当前器件行号寻址）来确定在哪些行上发送相控复位请求。

#### 4.2.4 DMD 复位状态机

相控复位状态机 *phase\_dmd\_reset\_fsm.vhd* 处理来自加载程序状态机的微镜复位请求。状态机处理**全局**和**相控**复位请求，向 DLPC910 控制器发出所需的块模式和块地址命令。

发出块命令后，状态机会监测来自 DLPC910 控制器的 *rst\_active*。*rst\_active* 变为高电平然后返回到低电平状态后，在指示复位请求完成之前会执行微镜稳定时间间隔。

使用一个小型 FIFO 进行复位命令排队，以应对发生背靠背相控复位请求的情况。当加载分辨率为 1 个微镜块加 1 个 DMD 行的图像时，很可能会出现此类情况。请求该块的复位，然后在 1 行之后请求下一个块的另一个复位。

#### 4.2.5 DMD 加载参数

DMD 加载参数模块 *dmd\_load\_params.vhd* 根据 DMD 类型从包文件 *dmd\_params\_pkg.vhd* 中选择加载参数。加载程序状态机使用所选数据进行分辨率和时序设置。

有关分辨率和时序详细信息，请参阅 [DLPC910 数据表](#)。

#### 4.2.6 同步脉冲

加载程序和 DMD 复位状态机都将其运行与 Apps FPGA 中的内部同步脉冲同步。脉冲宽度为一个 *c1kd* 周期。脉冲周期随 *dmd\_type* 变化。对于 DLP9000X 器件，每五个 *c1kd* 周期就有一个同步脉冲。对于 DLP6500，每八个 *c1kd* 周期有一个同步脉冲。对于每种 DMD 类型，周期表示行周期时间，即为给定的 DMD 行加载数据和发出行块命令所需的时间。

用户 DLP 控制逻辑中的行命令和块命令状态机也会将其运行时间定为行周期同步脉冲。

### 4.3 用户 DLP 控制

使用 Apps FPGA，用户能够通过 USB GPIF 将图像加载到 DLPC910 - DMD 芯片组。两个 VHDL 模块，即 *ugpif.vhd* 和 *dlp\_ctl\_ugpif.vhd*，提供用户控制。提供了全分辨率、1 位深、双端口 RAM 缓冲区来存储图像。

在 DMD 上显示用户图像的基本步骤为：

1. 将图像数据写入用户图像缓冲区 ( USB GPIF FIFO 写入传输, [FIFO 写入事务](#) )。
2. 发送 USB GPIF 行命令将缓冲区数据加载到 DLPC910 - DMD ( [用户行命令寄存器 \(0x002C\)](#) )。
3. 发送 USB GPIF 块命令来创建微镜控制脉冲，显示图像数据 ( [用户块命令寄存器 \(0x0030\)](#) )。

### 4.3.1 DLP6500 ( 1920 × 1080 ) 用户图像显示示例 ( 全局 )

以下步骤介绍了如何在 DLP6500 DMD 上显示图像：

#### 将缓冲区数据从 PC 发送到 Apps FPGA

步进	说明	寄存器地址	数据
1	设置 Apps FPGA 用户加载模式 ( 全零 )	0x0040	使能位 = '0'
2	将缓冲区开始写入设置为零，将行数设置为 1080 (0x438)	0x0024	0x438
3	<b>将 1080p 图像发送到用户图像缓冲区。</b> 需要 507 次写入突发 ( FIFO 写入事务 )		
	突发 1		256 , 16 位字
	突发 2		256 , 16 位字
	⋮		
	突发 n		256 , 16 位字
	⋮		
	突发 507		256 , 16 位字

#### 将数据从 Apps FPGA 数据缓冲器发送到 DLPC910 - DMD

如有需要，设置图像数据修饰符 ( ns\_flip、comp\_data )。使用 DLPLC910EVM 板上的 DIP 开关 (SW2) 或使用寄存器 0x0010。

步进	说明	寄存器地址	数据
1	将第一行数据发送到 DLPC910 - DMD	0x002C	0x00000003
2	发送剩余的 1079 行	0x002C	0x04370001
3	向 DLPC910 - DMD 发出全局微镜复位	0x0030	0x00000183

1. 通过向行命令寄存器 0x002C 写入 0x00000003，将第一行图像数据发送到 DLPC910 - DMD。
2. 通过向寄存器 0x002C 写入 0x04370001 来发送所有剩余行。
3. 通过对寄存器 0x0030 进行写入向 DLPC910 - DMD 发出全局微镜复位。

### 4.3.2 DLP9000X ( 2560 × 1600 ) 用户图像显示示例 ( 全局 )

以下步骤介绍了如何在 DLP9000X 或 DLP9000XUV DMD 上显示图像：

#### 将缓冲区数据从 PC 发送到 Apps FPGA

步进	说明	寄存器地址	数据
1	设置 Apps FPGA 用户加载模式 ( 全零 )	0x0040	使能位 = '0'
2	将缓冲区开始写入设置为零, 将行数设置为 1600 (0x640)	0x0024	0x640
3	<b>将 WQXGA 图像发送到用户图像缓冲区。</b> 这需要 1000 次写入突发 ( FIFO 写入事务 )		
	突发 1		256, 16 位字
	突发 2		256, 16 位字
	⋮		
	突发 <i>n</i>		256, 16 位字
	⋮		
	突发 1000		256, 16 位字

#### 将数据从 Apps FPGA 数据缓冲器发送到 DLPC910 - DMD

如有需要, 设置图像数据修饰符 ( ns\_flip、comp\_data )。使用 DLPLC910EVM 板上的 DIP 开关 (SW2) 或使用寄存器 0x0010。

步进	说明	寄存器地址	数据
1	将第一行数据发送到 DLPC910 - DMD	0x002C	0x00000003
2	发送剩余的 1599 行	0x002C	0x063F0001
3	向 DLPC910 - DMD 发出全局微镜复位	0x0030	0x00000183

### 4.3.3 Load4 - 与 DLP6500 DMD 一起使用

#### 备注

在 Load4 模式下使用 DLP6500 DMD 时, DMD 行指针不接受预加载 (rowmd = "10")。行指针只能清除 (rowmd = "11") 和递增 (rowmd = "01")。此外, 还有 15 个行指针计数不会导致数据写入 DMD 存储器。因此, 要写入 DMD 的第 0 行, 必须先清除行指针 (rowmd = "11"), 然后进行 14 次虚拟写入 (rowmd = "01")。

### 4.3.4 USB GPIF FIFO 数据写入

通过写入 USB GPIF FIFO, 图像可传输至用户图像缓冲区。

图像必须从左上角传输到右下角。USB IF FIFO 复位 ( 寄存器 0x0024 ) 后, 接收到的第一个 16 位字放置在 DMD 的第 0 列至第 15 列中。接收到的第一个 16 位字的 LSB 转至 DMD 第 0 列, MSB 转至第 15 列。

对于 DLP6500 DMD, 传输过程将持续进行, 直至发送了 120 个 16 位 FIFO 字, 其中包括 DMD 上的 1920 列行。接下来发送的 120 个字将放置在下一个 DMD 行中。

对于 DLP9000X 或 DLP9000XUV DMD, 传输过程将持续进行, 直至发送了每个 DMD 行的 160 个 16 位字, 其中包括 DMD 上的 2560 列行。接下来发送的 160 个字将放置在下一个 DMD 行中。

### 4.3.5 外部触发

DLPLC910EVM 上提供了一个外部触发端口 ( 启用外部全局复位触发器 ), 而不是使用寄存器写入来发出全局复位。当触发输入通过高电平有效信号产生脉冲时, 会向 DLP 组件集发出全局复位块命令。

## 4.4 USB GPIF (运行)

Apps FPGA 可通过 USB 通用接口 (GPIF) 连接内部控制状态寄存器。USB GPIF (接口) 中介绍了 USB GPIF 信号和总线事务。USB GPIF 寄存器中介绍了寄存器。USB GPIF 功能的顶层 VHDL 模块是 *ugpif.vhd*。

## 4.5 时钟和复位

*clk\_rst.vhd* 模块包含 Apps FPGA 的时钟 PLL 并对每个时钟域的复位信号进行同步。

### 4.5.1 参考时钟

Apps FPGA 接收两个外部时钟作为内部 PLL 的参考时钟：

- 来自 VC-707 EVM、200MHz 振荡器的 *sysclk\_p/n*
- 来自 Infineon - Cypress USB 接口芯片的 *usb\_if\_clk*

Apps FPGA 不使用由 DLPLCRC910EVM 驱动的第三个外部时钟源 *spare\_clk*。该时钟有两种类型：差分信号类型和单端信号类型。

### 4.5.2 Clk50 和 Clk100

通用时钟 *clk50* 用于开关去抖并由 *init-run-park* 状态机使用，是一个 50MHz 自由运行时钟。

测试图形发生器使用的是自由运行的 100MHz 时钟 *clk100*。

两个时钟均由单个 PLL (AMD - Xilinx IP) 生成，并且进行了相位对齐。PLL 的参考时钟为 *sysclk\_p/n*。

### 4.5.3 DLP 时钟

对于内部 DLP 控制逻辑，Apps FPGA 使用的时钟频率是 DLPC910 高速接口时钟频率的 1/4。该时钟在 VHDL 代码中被命名为 *clkd*。

DLPC910 高速接口是一个双倍数据速率 (DDR) 接口，可实现 8:1 的高速与内部时钟比。这样，Apps FPGA 就可以使用 AMD - Xilinx OSERDESE2 DDR 基元进行数据控制输出。对于每个 Apps FPGA 内部时钟上升沿，8 个数据位会加载到 OSERDESE2 基元中，以便以高速 DDR 时钟速率移出。

Apps FPGA 需要支持两种不同的高速接口时钟频率，即 400MHz 和 480MHz。使用单独的 PLL (AMD/Xilinx IP)，以便创建精确的时钟频率。一个 PLL 生成 100MHz/400MHz 时钟，另一个 PLL 生成 120MHz/480MHz 时钟。两个 PLL 均使用 200MHz *sysclk\_p/n* 作为其参考时钟，并且两个 PLL 都连续运行。

时钟多路复用器基元用于创建 *clkd* 和 *clkd4x* (*clkd4x* 与 *clkd* 相位对齐，供 OSERDESE2 基元使用)。通过单个时钟多路复用器选择 100MHz 或 120MHz 时钟来创建 *clkd*。通过另一个时钟多路复用器选择 400MHz 或 480MHz 时钟来创建 *clkd4x*。初始化期间，由 *init-run-park* 状态机执行一次时钟选择。选择时钟后，*init-run-park* 状态机向 OSERDESE2 基元和 Apps FPGA 逻辑发出复位。

### 4.5.4 USB GPIF 时钟

USB GPIF 逻辑 (*clku*) 的时钟是由参考时钟为 *usb\_if\_clock* 的 PLL 生成的。PLL 输出时钟 (*clku*) 与 PLL 参考时钟的相位频率一致。PLL 的主要用途是对时钟树和缓冲区延迟进行补偿。

### 4.5.5 逻辑复位

根据 AMD - Xilinx 针对 7 系列 FPGA 的建议，所有 Apps FPGA 时钟逻辑均使用同步复位进行编码。该规则有一些例外，例如 PLL 的失锁复位，其异步置为有效，同步置为无效。

同步复位信号名称通常使用与给定域时钟相似的名称。例如，*rstu* 是 *clku* 域的同步复位。

### 4.5.6 时钟域交叉 (CDC)

对于时钟域交叉数据，Apps FPGA 使用 FIFO 和双端口 RAM。这两个组件都是使用 Vivado 中的 AMD - Xilinx IP 生成工具创建的。

对于控制信号和寄存器值域交叉，Apps FPGA 使用 AMD - Xilinx 参数化宏。

Apps FPGA 的异步信号输入通过三个触发器进行同步。这三个触发器应用了 `ASYNC_REG` 属性，以便 Vivado 工具将其尽可能靠近放置，从而最大限度地降低亚稳态。

## 4.6 开关去抖

`sw_debounce.vhd` 模块会对 Apps FPGA 的开关输入进行去抖处理。以下开关输入已去抖：

- DLPLCRC910EVM 上 DIP 开关的 `evm_in_dip_sw(7:0)`
- VC-707 评估板上 DIP 开关的 `vc707_in_dip_sw(7:0)`
- VC-707 评估板上按钮开关的 `vc707_in_pb_sw(7:3)`
- DLPLCRC910EVM (SW1) 上按钮开关的 `resetz`

## 5 USB GPIF 寄存器

Apps FPGA 提供了读取/写入 32 位寄存器接口，可使用 Infineon CY7C68013A-128 USB 外设控制器通过 USB 进行通信。单芯片外设控制器位于 DLPC910 EVM 板上。

USB GPIF (接口) 中介绍了通过 USB GPIF 实现的寄存器读取/写入事务协议。

### 5.1 寄存器定义

本文档的这一部分通篇使用了以下标识：

- R：标识只读寄存器。
- R/W：标识既可读又可写的寄存器。

#### 5.1.1 状态 (0x000C)

地址	位	说明	默认值 <sup>(1)</sup>	读/写
0x000C	(31:16)	未使用	0x0000	R
	15	未使用	'0'	R
	14	未使用	'0'	R
	13	未使用	'0'	R
	12	未使用	'0'	R
	11	未使用	'0'	R
	10	未使用	'0'	R
	9	未使用	'0'	R
	8	未使用	'0'	R
	7	未使用	'0'	R
	6	未使用	'0'	R
	5	DLOK：1 = DLP (DMD) PLL 被锁定。0 = 未锁定	'1'	R
	4	SLOK：1 = 系统 PLL 被锁定。0 = 未锁定	'1'	R
	3	RACT：来自 DLPC910 的 rst_active 信号 <sup>(2)</sup>	'0' 或 '1'	R
	2	IRQ：来自 DLPC910 的 DMD_IRQ 信号 <sup>(2)</sup>	'0'	R
	1	ECP：来自 DLPC910 的 ecp2_finished 信号 <sup>(2)</sup>	'1'	R
	0	CAL：1 = Apps FPGA 正在向 DLPC910 发送训练模式	'0'	R

(1) 默认列显示初始化完成后正常系统运行情况下的状态位值。默认列以外的状态值表明系统有问题。

(2) 该状态位表示 Apps FPGA 传递来自 DLPC910 的信号。

### 5.1.2 数据加载控制 (0x0010)

地址	位	说明	默认值	读/写
0x0010	(31:16)	未使用	0x0000	R
	15	未使用	不适用	
	14	未使用	不适用	
	13	未使用	不适用	
	12	未使用	N/A	
	11	未使用	N/A	
	10	未使用	N/A	
	9	未使用	不可用	
	8	ETRG : 1 = 启用外部全局复位触发器 ; 0 = 未启用 <sup>(1)</sup>	'0'	R/W
	7	FLOT : 1 = 使用 blkmd 命令将 DMD 微镜悬空 ; 0 = 未启用悬空 <sup>(2)</sup>	'0'	读/写
	6	未使用	不适用	
	5	未使用	不适用	
	4	CD : 1 = 显示前的补码数据 ; 0 = 无补码 <sup>(3)</sup>	'0'	读/写
	3	WD : 1 = 启用 DLPC910 看门狗计时器 ; 0 = 禁用 <sup>(3)</sup>	'0'	读/写
	2	NS : 1 = DMD 图像垂直翻转 ; 0 = 无翻转 <sup>(3)</sup>	'0'	R/W
1	L4 : 0 = 启用 DMD Load4 模式 ; 1 = 正常负载 <sup>(3)</sup>	'1'	R/W	
0	PC : 1 = PC GUI 控制 ; 0 = DIP 开关控制 <sup>(3)</sup>	'0'	读/写	

- (1) ETRG 位, 当设置为 '1' 时, 为 GUI 控制的 DMD 负载启用外部触发的全局微镜复位 ( 当寄存器 0x0040 的位 0 = '0' 时 )。该位对 Apps FPGA 图像 ( 测试图形 ) 加载程序没有影响。
- (2) 当设置为 '1' 时, FLOT 位指示 Apps FPGA 图像加载程序使用 blkmd = "11" 和 blkad = "11XX" 将 DMD 微镜悬空。系统将保持此状态, 直到 FLOT 设置为零。该位对 GUI 控制的图像加载没有影响。
- (3) 当 PC 位为 0 时, 位 (4:1) 值取自 DLPC910 EVM 板上的 DIP 开关 SW2。当 PC 位为 1 时, 位 (4:1) 值取自该寄存器。

#### 备注

除 ETRG 位外, 其他控制位同时由应用测试图形加载和 USB GPIF 用户图像加载使用。

### 5.1.3 测试图形控制 (0x0014)

地址	位	说明	默认值	读/写
0x0014	(31:12)	TPG 循环间隔 (19:0) : 定义测试图形发生器的测试图形显示间隔。LSB = 1ms <sup>(1)</sup>	0x7D0	读/写
	11	未使用		
	10	未使用		
	9	未使用		
	8	CEN - 1 = 测试图形循环。0 = 禁用循环 <sup>(2)</sup>	'1'	读/写
	(7:0)	测试图形选择 (7:0) <sup>(3)</sup>	0x00	读/写

- (1) 1ms 表示该值的步长。值 0x000 会导致不可预测的特性。
- (2) 启用图形循环时, 测试图形发生器以 TPG 循环间隔给定的速率循环将图形 0x00 至 0x0E 写入测试图形缓冲区。禁用图形循环后, 测试图形发生器会将所选图形写入测试图形缓冲区。
- (3) 测试图形选择在禁用测试图形循环时选择要显示的测试图形 :
  - 0x00 : 全开图形
  - 0x01 : 全关图形
  - 0x02 : ANSI 棋盘
  - 0x03 : 具有单像素外边框的单像素栅格线图形
  - 0x04 : 自西向东单像素对角线
  - 0x05 : 自东向西单像素对角线



- 0x06 : 水平线
- 0x07 : 垂直线
- 0x08 : Load4 棋盘
- 0x09 : 棋盘
- 0x0A : 反转棋盘
- 0x0B : 1x1 水平线
- 0x0C : 1x1 垂直线
- 0x0D : 随机噪声图形
- 0x0E : 块边界棋盘
- 0x0F : 用户定义的图形 ( 通过 USB/GPIF 加载 )
- 0xFF-0x10 : 未使用

#### 5.1.4 测试行地址 (0x0018) - [未使用]

地址	位	说明	默认值	读/写
0x0018	(31:11)	未使用	零	读/写
	(10:0)	测试行地址 <sup>(1)</sup>	0x000	读/写

(1) Apps FPGA 未使用测试行地址字段。寄存器 0x002C 中的 F1S 位支持强制 1 函数的测试。

#### 5.1.5 加载程序复位类型 (0x001C)

地址	位	说明	默认值	读/写
0x001C	(31:4)	未使用	零	读/写
	(3:0)	复位类型 (3:0) <sup>(1)</sup>	0x02	读/写

(1) 复位类型指示 Apps FPGA 加载程序使用指定的微镜复位类型：

- 0x0 = 相控 × 1
- 0x1 = 相控 × 2
- 0x2 = 全局
- 0x3 = 相控 × 4

#### 5.1.6 类型和版本 (0x0020)

地址	位	说明	默认值	读/写
0x0020	(31:24)	EVM_DIPSW(7:0) <sup>(1)</sup>	从 DLPLCRC910EVM 读取	R
	(23:7)	未使用	零	R
	(6:4)	DDC_VER(2:0) <sup>(2)</sup>	从 DLPC910 读取	R
	(3:0)	DMD_TYPE(3:0) <sup>(3)</sup>	从 DLPC910 读取	R

- (1) EVM\_DIPSW 字段是从 DLPLCRC910EVM 板上的 8 位 DIP 开关向 Apps FPGA 发送信号的 8 位值。开箱即用逻辑的默认值为 0x03。请参阅 [DLPLCRC910EVM DIP 开关 \(SW2\)](#)。
- (2) DDC\_VER 域是在 DDC 版本输入引脚处向 Apps FPGA 发送信号的 3 位值。Apps FPGA 不使用该值。
- (3) DMD\_TYPE 域是在 DMD 类型输入引脚处向 Apps FPGA 发送信号的 4 位值。Apps FPGA 使用 DMD 类型来确定 DMD 分辨率和每个行周期的时钟。有关更多信息，请参阅 [DLPC910 数据表](#)。

#### 5.1.7 用户图像缓冲区写入设置 (0x0024)

地址	位	说明	默认值	读/写
0x0024	(31:16)	buf_wstart_row(15:0) <sup>(1) (2)</sup>	0x0000	读/写
	(15:11)	未使用	0x00	读/写
	(10:0)	buf_wstart_numrows(10:0) <sup>(1) (3)</sup>	1	R/W

(1) 该寄存器包含用户图像缓冲区写入设置。对该寄存器进行写入操作可让用户图像缓冲区写入状态机准备好开始检测 USB/GPIF FIFO，以便将图像传输到用户图像缓冲区。通过 USB/GPIF 接收到的第一行会写入用户图像缓冲区的 buf\_wstart\_row。缓冲区写入状态机接受 buf\_wstart\_numrows 的数据，之后缓冲区停止写入数据，直到对该寄存器进行新的写入操作。此外，对该寄存器的写入操作将复位 USB/GPIF FIFO。

- (2) buf\_wstart\_row 的有效值为 0 至 (TOTAL\_ROWS\_ON\_DMD - 1)。Buf\_wstart\_row 必须小于或等于 TOTAL\_ROWS\_ON\_DMD - buf\_wstart\_numrows。
- (3) Buf\_wstart\_numrows 有效值为 1 至 TOTAL\_ROWS\_ON\_DMD。

### 5.1.8 USB GPIF FIFO 读取突发大小 (0x0028) - [已淘汰]

地址	位	说明	默认值	读/写
0x0028	(31:10)	未使用	零	读/写
	(9:0)	FIFO 读取突发大小 - 已淘汰 <sup>(1)</sup>	不可用	R/W

- (1) Apps FPGA 逻辑不再使用该寄存器。USB GPIF 逻辑仍然支持对该寄存器的读取/写入操作，但当前未使用其内容。

### 5.1.9 用户行命令寄存器 (0x002C)

地址	位	说明	默认值	读/写
0x002C	(31:29)	未使用	零	读/写
	28	F1S <sup>(1)</sup>	0	R/W
	27	未使用	0	R/W
	(26:16)	Numrows <sup>(2) (3)</sup>	1	R/W
	15	未使用	0	R/W
	(14:4)	ROWAD <sup>(2) (3)</sup>	0x000	读/写
	(3:2)	未使用	"00"	读/写
	(1:0)	ROWMD <sup>(2) (3)</sup>	"00"	读/写

- (1) 当 F1S 为 '1' (F1S = 强制为一) 时，所有 '1' 的数据将在给定的行周期内发送到 DLPC910。
- (2) ROWMD 对用户图像缓冲区读取有以下影响：
- "00"：无操作。
  - "01"：递增缓冲区读取地址计数器，然后将已寻址的缓冲区行数据发送到 DMD。继续执行此操作，直到发送了 numrows (位 26:16)。
  - "10"：使用 ROWAD 加载应用 DLP ROWAD 计数器。将 ROWAD 发送到 DLPC910。使用 ROWAD 加载缓冲区读取地址计数器。将已寻址的缓冲区行发送到 DLPC910。
  - "11"：将 ROWAD = 零发送到 DLPC910。将缓冲区读取地址计数器清零。向 DLPC910 发送缓冲区第 0 行数据。
- (3) 用户行命令寄存器用于将数据从用户图像缓冲区移至 DMD。写入寄存器时，ROWMD 和 ROWAD 连同从用户图像缓冲区中读取的数据一起转发到 DLPC910 控制器。有关此类行控制信号功能的更多详细信息，请参阅 DLPC910 数据表。

当控制器未响应对此寄存器的写入操作时，Apps FPGA 会持续向 DLPC910 控制器发送 no-op 行命令。

### 5.1.10 用户块命令寄存器 (0x0030)

地址	位	说明	默认值	读/写
0x0030	(31:9)	未使用	零	读/写
	8	RST2BLKZ <sup>(1)</sup>	1	R/W
	(7:4)	BLKAD <sup>(1)</sup>	0x0	读/写
	(3:2)	未使用	零	读/写
	(1:0)	BLKMD <sup>(1)</sup>	"00"	读/写

- (1) 写入用户块命令寄存器会指示 Apps FPGA 将该寄存器中的块命令转发到 DLPC910 控制器。DLPC910 数据表中介绍了块命令详细信息。转发的块命令包括与 DCLK 和 DVALID 的适当同步。

### 5.1.11 加载程序行控制 (0x0034)

地址	位	说明	默认值	读/写
0x0034	(31:27)	未使用	零	读/写
	(26:16)	起始行 <sup>(1)</sup>	0x000	读/写
	(15:11)	未使用	零	读/写
	(10:0)	加载行 <sup>(2)</sup>	1	R/W

- (1) 起始行字段指示测试图形加载程序在哪一 DMD 行开始加载图像。

(2) 加载行字段指示测试图形加载程序要发送到 DLP 芯片组的测试图形行数。

### 5.1.12 加载程序加载间隔 (0x0038)

地址	位	说明	默认值	读/写
0x0038	(31:10)	未使用	零	读/写
	(9:0)	加载间隔。LSB = 1 $\mu$ s <sup>(1)</sup>	0x190	读/写

(1) 加载间隔字段设置触发测试图形缓冲区读取的间隔。默认间隔为 400  $\mu$ s。LSB 是步长。

#### 备注

如果加载间隔设置为短于配置的 DLP 芯片组加载时间，则加载程序将忽略繁忙时发生的任何触发。不建议使用小于 50  $\mu$ s 的值。

### 5.1.13 加载程序曝光时间 (0x003C)

地址	位	说明	默认值	读/写
0x003C	(31:16)	未使用	零	读/写
	(15:0)	曝光时间计数 <sup>(1)</sup>	0x0000	读/写

(1) 当触发发生时，加载程序会将图形缓冲器发送到 DMD 并执行复位来显示图形。曝光是复位完成后加载程序等待下一次触发之前增加的等待时间。曝光时间以行周期计数衡量：

- 对于 DLP9000X 和 DLP9000XUV DMD，在 400MHz 或 480MHz 下行周期为 20 个 dclk 周期 (50ns 或 41.67ns)。
- 对于 DLP65000 DMD，在 400MHz 下行周期为 32 个 dclk 周期 (80ns)。

例如，默认加载间隔为 400  $\mu$ s 时，在总加载时间加上复位时间再加上曝光时间超过 400  $\mu$ s 的倍数之前，图形加载频率没有变化。因此，有效步长是加载程序加载间隔 (0x0038)。

### 5.1.14 地址写入 (0x003F) - [未使用]

地址	位	说明	默认值	读/写
0x003F	(31:0)	USB GPIF 未使用地址写入数据 <sup>(1)</sup>	零	读/写

(1) Apps FPGA USB GPIF 逻辑将写入未使用寄存器地址的 32 位数据置于该寄存器中。

### 5.1.15 加载程序控制 (0x0040)

地址	位	说明	默认值	读/写
0x0040	(31:2)	未使用	零	R/W
	1	自由运行 <sup>(1)</sup>	0	R/W
	0	加载程序使能 <sup>(2)</sup>	1	R/W

(1) 当设置为 '1' 时，自由运行位指示测试图形加载程序以最大速率运行，忽略加载触发。

(2) 当加载程序使能设置为 '1' 时，Apps FPGA 测试图形加载程序会将测试图形图像发送到 DLP 芯片组。当设置为 '0' 时，测试图形加载停止，DLP 芯片组通过 USB GPIF 进行控制。当加载程序使能位从 '1' 转换为 '0' 时，使用块清除命令清除 DMD 图像，之后用户将通过 USB GPIF 控制 DLP 芯片组。

### 5.1.16 Park [PWR\_FLOAT] (0x0044)

地址	位	说明	默认值	读/写
0x0044	(31:1)	未使用	零	R/W
	0	Park (PWR_FLOAT) <sup>(1)</sup>	0	R/W

(1) 向 Park 位写入值 '1' 会命令 Apps FPGA 将 PWR\_FLOATZ 信号驱动为低电平至 DLPC910。当 Park 位状态从 '1' 变为 '0' 时，Apps FPGA 将 PWR\_FLOATZ 驱动为高电平，复位 (CTL\_RSTZ = '0') 并重新初始化 DLPC910。Apps FPGA 逻辑也会复位。

### 5.1.17 外部触发状态 (0x0048)

地址	位	说明	默认值	读/写
0x0048	(31:1)	未使用	零	R/W
	0	接收到的外部触发 <sup>(1)</sup>	0	R/W

(1) 在 GUI 脚本模式下，第 0 位为 '1' 时表示接收到外部触发。向该寄存器写入任何值即可清除该位。

### 5.1.18 FPGA 编译日期 (0x0080)

地址	位	说明	默认值	读/写
0x0080	(31:16)	以二进制编码小数 (BCD) 格式编译年份。年份中的每个数字 1 个半字节 (例如, 0x2022 == 2022 年)		R
	(15:8)	以 BCD 格式编译月份。月份中的每个数字 1 个半字节 (例如, 0x11 = 第 11 个月 = 11 月)		R
	(7:0)	以 BCD 格式编译月份中的某一天。(例如, 0x12 == 该月的第 12 天)		R

### 5.1.19 主要 - 次要版本 (0x0084)

地址	位	说明	默认值	读/写
0x0084	(31:16)	未使用	零	R
	(15:8)	1 字节主要版本，十六进制		R
	(7:0)	1 字节次要版本，十六进制		R

### 5.1.20 固定值 FPGA 标识符 (0x0088)

地址	位	说明	默认值	读/写
0x0088	(31:0)	32 位固定 FPGA 标识符= 0x000AC910	0x000AC910	R

### 5.1.21 测试寄存器 (0x008C)

地址	位	说明	默认值	读/写
0x008C	(31:0)	USB GPIF 32 位测试读取/写入值 <sup>(1)</sup>	零	读/写

(1) 该寄存器仅用于 USB GPIF 测试。该寄存器的内容不影响 Apps FPGA 的运行。

## 6 FPGA 配置

AMD - Xilinx VC707 板使用 16 位闪存接口进行 FPGA 配置。使用 AMD - Xilinx Vivado 实验室工具通过 JTAG 配置闪存。

### CAUTION

TI 建议在将 VC-707 板连接到 DLPLCRC910EVM 之前，先使用 Apps FPGA 配置存储器文件写入闪存，以避免出厂加载的 VC-707 FPGA 配置和 DLPC910 控制器之间可能出现 FMC 连接器 I/O 冲突。

## 7 Apps FPGA 源文件和编译

### 7.1 设计工具

AMD Xilinx Vivado 版本 2019.2 用于 Apps FPGA 合成、实现和仿真。VC-707 评估套件随附的 SW 工具许可证支持编译 Apps FPGA 所需的所有 Vivado 功能和 IP，并使用提供的 VHDL 测试台文件进行仿真。

### 7.2 源文件

本节列出了构建 Apps FPGA 所需的源文件，并附有简短说明（如图 7-1 所示）：

Module Name	Xilinx IP	Description
<b>primary VHDL and IP modules</b>		
dlpc910_apps_top.vhd		apps FPGA top level
clk_rst.vhd		clocks and resets
pll_50.xci	✓	PLL for 50 MHz & 100 MHz clocks
pll_400.xci	✓	PLL for 100 MHz and 400 MHz DLP clocks
pll_480.xci	✓	PLL for 120 MHz and 480 MHz DLP clocks
pll_ugpif.xci	✓	PLL for 48 MHz USB GPIF clock
sw_debounce.vhd		switch debouncing
db_multi.vhd		debouncing for specified number of switches
debounce.vhd		basic debounce module
irp.vhd		init-run-park control
irp_fsm.vhd		init-run-park state machine
dload_trigger.vhd		load trigger for the dlp_ctl_n_tpg module
mult_trig_count.xci	✓	multiplier for calculating the trigger period counter load value
dlp_ctl_n_tpg.vhd		test pattern generator and DLPC910 chip set load controller
dmd_load_params.vhd		creates DMD specific parameters needed by the load controller
tpg.vhd		test pattern generator
tpg_trig.vhd		test pattern trigger generator
tpg_timing.vhd		test pattern timing generator
dmd_data_buffer.vhd		test pattern buffer
dpram_512x1600.xci	✓	dual port RAM IP, 512 bits wide by 1600 rows
dmd_load_fsm.vhd		DLPC910 load controller state machine (row commands)
dlp9000_lut.xci	✓	timing LUT for the DLP9000. Requires dlp9000_load_lut.coe initialization file.
dlp6500_lut.xci	✓	timing LUT for the DLP6500. Requires dlp6500_load_lut.coe initialization file.
phase_dmd_rst_fsm.vhd		DLPC910 MCP reset controller state machine (block commands)
mcp_req_fifo_16x4.xci	✓	fifo for accumulating mirror block reset requests
dlp_ctl_ugpif.vhd		user DLPC910 chips set load controller
ug_fifo_read_fsm.vhd		state machine that transfers data from the GPIF fifo to the user buffer
ug_data_buffer.vhd		user data buffer
dpram_512x1600x16in.xci	✓	dual port RAM IP, 1600 rows, 16 bit wide input, 512 bit wide output. Uses 16x12800_coe.coe
ug_dmd_load_fsm.vhd		user DLPC910 chip set load state machine (row commands)
ug_dmd_rst_fsm.vhd		user DLPC910 mirror block command state machine
data_ctl_out.vhd		high speed serial data and control outputs, including lvds buffers
ddr_36_oserdes_lvdsout.vhd		32 bit lvds data, clock (2), and dvalid(2) output shift registers
se_serdes_out.vhd		single ended control output shift registers
ddr_1_oserdes.vhd		wrapper for single oserdes2 primitive
ugpif.vhd		USB GPIF control and registers
word_swap.vhd		word swap function
fifo_fwft_1Kx16.xci	✓	USB GPIF fifo. first word fall-through. 16 bit read/write width
ugpif_regs_fsm.vhd		USB GPIF control state machine
<b>modules with multiple instantiations</b>		
inpin_sync.vhd		three flip flop asynchronous input synchronizer.
sync_arst.vhd		wrapper for AMD / Xilinx xpm_cdc_single macro. async reset input flip flop.
sync.vhd		wrapper for AMD / Xilinx xpm_cdc_single macro.
syncp.vhd		wrapper for AMD / Xilinx xpm_cdc_pulse macro.
syncbus_hs.vhd		wrapper for AMD / Xilinx xpm_cdc_handshake macro.
clken.vhd		clock enable generator
<b>vhdl_packages</b>		
dmd_params_pkg.vhd		DMD device specific parameters (resolution, mirror blocks, etc)
tpg_pkg.vhd		test pattern constants
<b>AMD / Xilinx Vivado constraints</b>		
dlpc910_evm_fpa_physio_constraints.xdc		apps FPGA pin assignments and I/O characteristics
dlpc910_evm_fpga_timing_constraints.xdc		apps FPGA timing constraints
<b>memory IP initialization files</b>		
16x12800_coe.coe		init file for the user data buffer dual port RAMs.
dlp9000_load_lut.coe		init file for the dlp9000 lut
dlp6500_load_lut.coe		init file for the dlp6500 lut

图 7-1. 源文件

#### 7.2.1 主要 VHDL 和 IP 模块

主要 VHDL 模块 (.vhd) 提供 Apps FPGA 的主要 DLP 控制功能。AMD - Xilinx IP 模块 (.xci) 用于 PLL、FIFO、存储器和乘法器。

## 7.2.2 具有多个实例的模块

按本标题分组，共有六个 VHDL 模块 (.vhd)。这些模块供几个主要 VHDL 模块使用，具有多个实例。这六个模块包括五个时钟域交叉 (CDC) 和一个时钟使能发生器。

CDC 模块是 Xilinx 参数化 CDC 宏的包装器。使用此类宏可以简化 CDC 分析，因为 Vivado 会自动将其识别为 CDC 函数。

## 7.2.3 VHDL 软件包

*dmd\_params\_pkg.vhd* 文件包含受 Apps FPGA 支持的三个 DMD 的特定 DMD 分辨率和时序参数。这些参数由测试图形加载程序模块 (*dip\_ctl\_n\_tpg.vhd*) 使用。

*tpg\_pkg.vhd* 文件包含测试图形发生器模块 (*tpg.vhd*) 使用的若干测试图形数据常量。提供数据常量来支持 DLP9000X、DLP9000XUV 和 DLP6500 DMD。

## 7.2.4 Vivado 限制条件

除了由 Vivado IP 生成器创建的限制条件外，Apps FPGA 工程还需要两个限制条件文件。物理限制条件文件包含 I/O 引脚和特性。时序限制条件文件提供时钟和时序限制条件以及指令。

## 7.2.5 存储器 IP 初始化文件

两个查找表 (LUT) 初始化 (.coe) 文件用于填充测试图形加载程序状态机的查找表。*16x12800\_coe* 文件用于初始化具有宽垂直条带图形的用户图像缓冲区的前 40 个 DMD 行。其余的用户图像缓冲区行通过 IP 组件中 *dpram\_512x1600x16* 的 Vivado 块存储器生成器 IP 工具内的设置，使用垂直条纹 (0x00FF) 进行初始化。

### 7.2.5.1 查找表

Excel 电子表格 *dlpc910\_apps\_fpga\_luts\_coe.xlsx* 用于创建 DLP9000X、DLP9000XUV 和 DLP6500 DMD 的查找表。查找表由正在加载的 DMD 行寻址。对于每个 DMD 行，LUT 会指示行地址是否用于微镜块中的最后一行。加载程序状态机使用最后一行指示器来请求对刚刚完成加载的块进行 MCP 复位。NS FLIP、相控复位模式 (x1、x2、x4) 和 Load4z 均包含在该表中。该块地址也包含在该表中，以便转发到 DLPC910。

该 Excel 文件包含四个工作表，其中两个用于 DLP9000X - DLP9000XUV DMD，另两个用于 DLP6500 DMD。对于每个 DMD，第一个工作表按行地址创建 LUT 数据。第二个工作表只需复制 LUT 数据从而导出到 CSV 文件。然后使用文本编辑器添加 COE 页眉和页脚信息。

## 7.3 构建 Apps FPGA 代码

### 7.3.1 源代码

<源代码目录> 包含用于创建 *dlpc910\_apps* FPGA 工程的源文件和 .tcl 脚本。包含：

- 源文件夹和文件
- 用于创建工程的 .tcl 文件

#### 7.3.1.1 源文件夹

源文件夹包含四个子文件夹：

- *constr* : Vivado 物理和时序限制条件文件的文件夹
- *ip* : Virtex 7 IP (PLL、存储器、FIFO 等) 的文件夹
- *rtl* : VHDL 模块和软件包的文件夹
- *tbench* : VHDL 测试台文件的文件夹

此外，源文件夹中还包含以下四个文件：

- *16x12800\_coe.coe*
- *dip6500\_load\_lut.coe*
- *dip9000\_load\_lut.coe*

## 备注

仅适用于 DLP9000X 和 DLP9000XUV DMD 的文件。

- dlpc910\_apps\_fpga\_luts\_coe.xlsx

源文件章节提供了所有源文件的说明。

### 7.3.2 创建 Vivado 工程

按照以下步骤创建 dlpc910\_apps FPGA Vivado 工程：

1. 为工程创建文件夹
2. 将源目录从 <source dir> in installation> 复制到创建的文件夹中。源文件夹和 .tcl 文件（如上所述）现在显示在步骤 1 的文件夹中。
3. 打开 Vivado 2019.2 Tcl shell 并导航至步骤 1 的文件夹。
4. 从 Tcl shell 运行 .tcl 文件：`source <tcl_filename>.tcl`
5. .tcl 脚本在文件夹 dlpc910\_apps 中创建 Vivado 工程。
6. 关闭 Tcl shell 并启动 Vivado 2019.2 程序
7. 在“Quick Start”下，选择“Open Project >”并导航至在步骤 5 中创建的 dlpc910\_apps 文件夹。
8. 选择 dlpc910\_apps.xpr 文件并点击“OK”。dlpc910\_apps FPGA 工程在 Vivado 工程管理器 GUI 中打开。

### 7.3.3 编译设计

创建并在工程管理器 GUI 中打开 Vivado 工程后，即可通过点击流程导航器窗格中的 <Generate Bitstream> 来编译 DLPC910\_apps FPGA。工程管理器检测到没有综合和实现结果，并运行实施设计所需的流程。

### 7.3.4 仿真

#### 7.3.4.1 测试台

源代码章节中介绍的源代码存档还包括一个文件夹，其中包含 Apps FPGA 中主要 VHDL 模块的测试台。当执行工程 .tcl 脚本来重新创建 Apps FPGA 工程时，工程 .tcl 脚本会将测试台文件添加到工程中。可在 Vivado 工程管理器 GUI 的源层次结构窗口中查看测试台源。请参阅图 7-2。

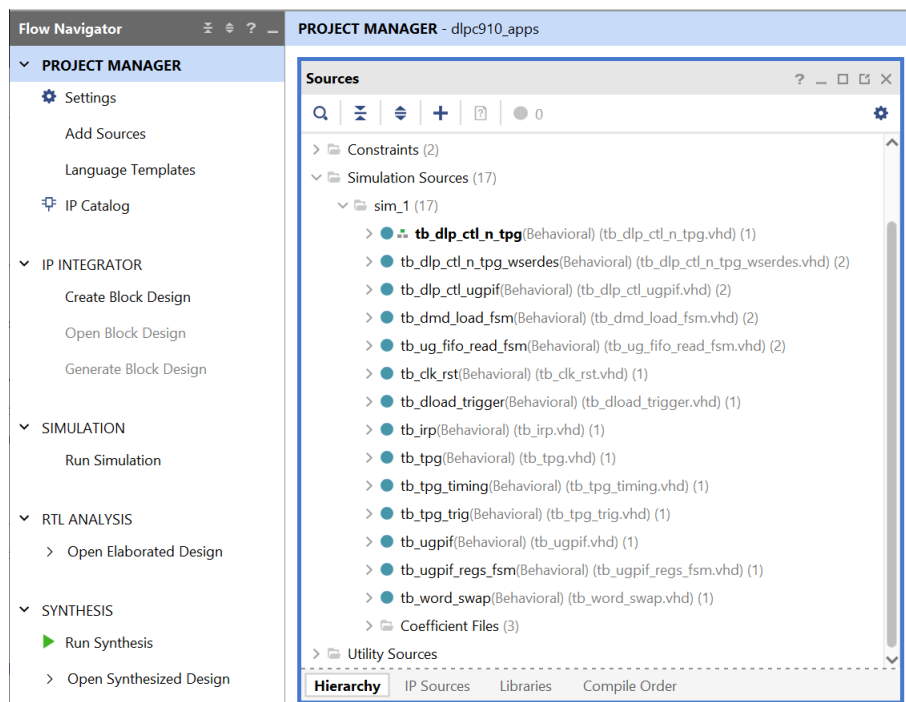


图 7-2. 测试台



### 7.3.4.2 模块仿真步骤

在流程导航窗格 (图 7-2) 中, 右键点击“SIMULATION”, 然后选择“Simulation Settings”。随后会显示如图 7-3 所示的“Settings”对话框。浏览并选择一个相关的测试台文件。该文件将显示在“Simulation top module name”文本框中。然后点击“OK”。

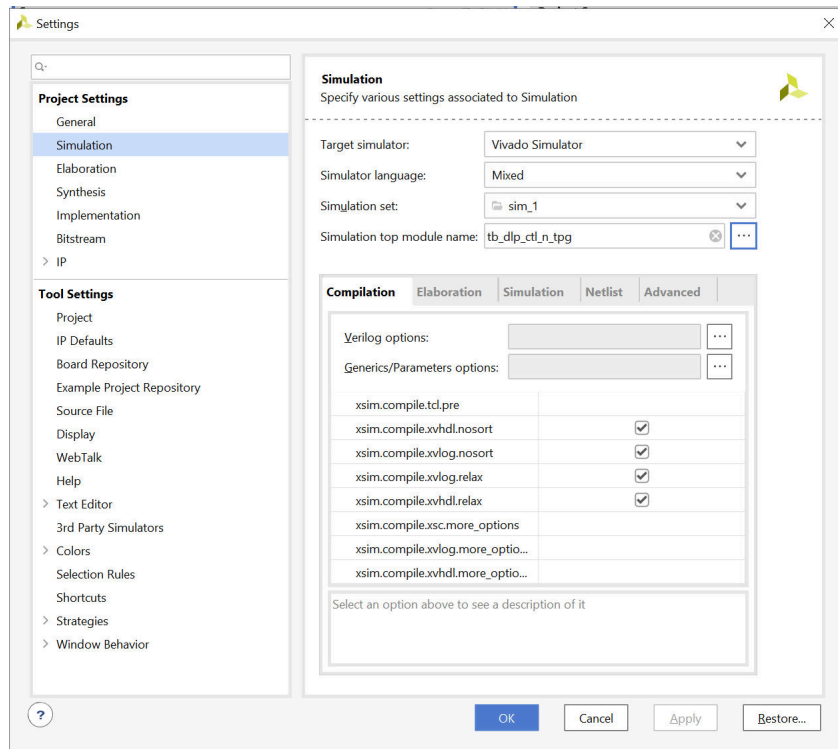


图 7-3. 设置对话框

接下来, 在流程导航窗格中选择“Run Simulation”, 然后选择“Run Behavioral Simulation”。测试台和待测器件 (UUT) 进行编译, Vivado 变更为仿真环境, 其波形窗口如图 7-4 所示。

然后, 用户可以使用默认波形窗口, 或打开关联的波形配置文件 (在“File”下拉菜单下, 选择“Simulation Waveform”)。每个测试台都有一个波形配置文件, 以便开始使用模块仿真。

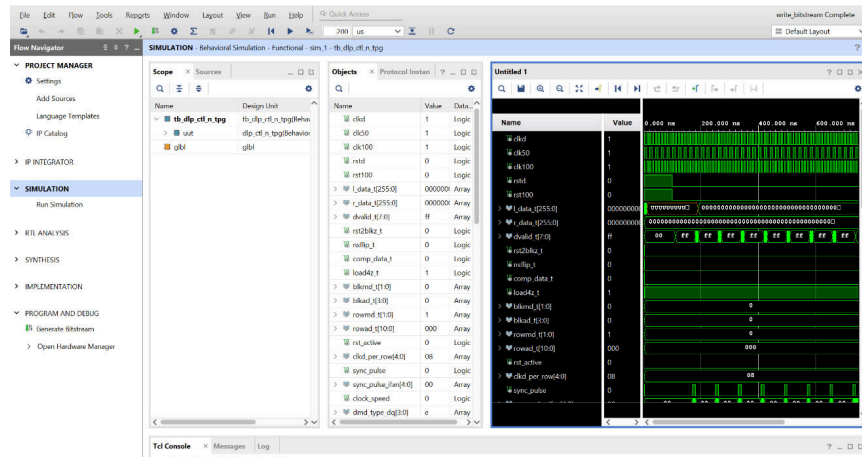


图 7-4. Vivado 波形窗口

## 8 德州仪器 (TI) 相关文档

可以在下述链接查看有关 DLPC910 的组件数据表、技术文档、设计文档和订购信息：

[DLPC910 数字控制器产品文件夹](#)

[LightCrafter DLPC910 EVM 产品文件夹](#)

[DLP6500FLQ DMD 产品文件夹](#)

[DLP6500FYE DMD 产品文件夹](#)

[DLP LightCrafter DLP6500FLQ DMD EVM 产品文件夹](#)

[DLP9000X DMD 产品文件夹](#)

[DLP LightCrafter DLP9000X DMD EVM 产品文件夹](#)

[DLP9000XUV DMD 产品文件夹](#)

[DLP LightCrafter DLP9000XUV DMD EVM 产品文件夹](#)

[适用于 Virtex-7 FPGA 的 AMD/Xilinx VC707 评估板 - 用户指南 \(UG855\)](#)

[AMD/Xilinx VC707 评估平台 - 原理图](#)

## 9 附录

### 9.1 缩略语和首字母缩写词

下面列出了本手册中使用的缩略语和首字母缩写词。

<b>Apps FPGA</b>	VC-707 EVM 或类似电路板上面向客户应用的 AMD Xilinx Virtex 7 FPGA
<b>BCD</b>	二进制编码小数
<b>CDC</b>	时钟域交叉
<b>DDR</b>	双倍数据速率
<b>DLL</b>	动态链接库
<b>DMD</b>	数字微镜器件
<b>FCC</b>	联邦通信委员会
<b>EVM</b>	评估模块 (板)
<b>FIFO</b>	先入先出
<b>FMC</b>	FPGA 夹层连接器
<b>FPGA</b>	现场可编程门阵列
<b>FSM</b>	有限状态机
<b>FW</b>	固件
<b>GPIF</b>	通用接口
<b>GPIO</b>	通用输入输出
<b>GUI</b>	图形用户界面
<b>HPC</b>	高引脚数
<b>HW</b>	硬件
<b>IO</b>	输入/输出
<b>JTAG</b>	联合测试行动组
<b>LED</b>	发光二极管
<b>LUT</b>	查找表
<b>MCP</b>	微镜时钟脉冲
<b>PBC</b>	处理器总线控制
<b>SPI</b>	串行外设接口

<b>TPG</b>	测试图形发生器
<b>SW</b>	开关
<b>USB</b>	通用串行总线
<b>UUT</b>	待测试单元
<b>VHDL</b>	验证和硬件描述语言

## 9.2 注意事项和警告信息

本手册包含注意事项和警告。

### CAUTION

下面是对注意事项声明的说明：注意事项声明描述了可能会损坏软件或设备的情况。

### WARNING

下面是对警告声明的说明：警告声明描述了可能对用户造成伤害的情况。

注意事项或者警告中所提供的信息是为了保护您的安全。请仔细阅读每条注意事项和警告。

**FCC 警告：**本设备仅限于在实验室测试环境中使用。该设备会产生、使用并可能辐射射频能量，并且尚未经过测试，不确定是否符合 FCC 规则第 15 部分 J 子部分关于计算设备的限制规定，该规则旨在针对射频干扰提供合理的保护。在其他环境中操作该设备可能会对无线电通信造成干扰，在这种情况下，用户需独自承担采取措施来消除此干扰所需的费用。

## 重要声明和免责声明

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、某特定用途方面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功能安全、信息安全、监管或其他要求。

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 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/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。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2023，德州仪器 (TI) 公司