

如何使用 DS90UB947-Q1 设计IVI 端的 FPD-Link III 系统

Violet Lei/Joey Jin

摘要

DS90UB947-Q1 是 TI 推出的符合 AEC-Q100 认证的车规级 OpenLDI 到 FPD-Link III 串行器芯片，广泛用于高清视频显示链路（如车载中控、液晶仪表等），其主要功能是将 OpenLDI 接口输入的并行 LVDS 图像信号串行化后，输出至 FPD-Link III 接口进行远距离高速传输。配合解串器可实现稳定、低 EMI、高分辨率（最高 1080p@60Hz）的视频传输。本应用笔记以 DS90UB947-Q1 为例，从系统设计到具体软硬件实现及其调试思路 and 手段，帮助读者快速熟悉并开发 FPD-Link III 串行解串应用系统。

内容

1	概述	2
1.1	系统设计.....	2
2	OpenLDI 协议	3
2.1	协议包 3.....	
3	基础设计规范	4
3.1	选型规范.....	4
3.2	硬件设计规范.....	5
3.3	软件设计规范.....	10
3.3.1	参考代码.....	10
3.3.2	关键寄存器.....	14
4	系统建立及调试指南	15
4.1	车载温度实验.....	16
4.1.1	PLL 设计.....	16
4.1.2	温度设置.....	17
4.2	系统调试框图.....	19
4.3	生成测试图片 PATGEN.....	21
5	总结	22
6	参考文献	22

Figures

Figure 1.	DS90UB947-Q1 经典应用系统	3
Figure 2.	DS90UB947-Q1 支持的 OLDI 传输编码方式	4
Figure 3.	DS90UB947-Q1 经典 I²C 设计通路	6
Figure 4.	DS90UB947-Q1 的模式设置	7
Figure 5.	FPD-Link 信号线 DOUT 在 STP 双绞线下的推荐器件	8

Figure 6.	DS90UB947-Q1 标准上电时序	8
Figure 7.	初始化时序选择一	9
Figure 8.	初始化时序选择二	9
Figure 9.	初始化时序选择三	10
Figure 10.	DS90UB947-Q1 与 DS90UB948-Q1 的典型应用场景	16
Figure 11.	锁相环 PLL 设计电路	16
Figure 12.	压控振荡器 VCO 的压控曲线及初始电压区间	18
Figure 13.	DS90UB947-Q1 温度上升或下降实验可满足的温度范围	19
Figure 14.	DS90UB947-Q1 系统调试指南	19
Figure 15.	测试图片生成器 (PATGEN) 的经典图案	21

1 概述

随着智能驾驶、座舱集成与高清化趋势的不断发展，车载显示系统正从传统的低分辨率、多屏分离架构，逐步迈向高分辨率、大屏融合、信息集成化的新阶段。未来，在智能座舱一体化、车载娱乐多媒体系统以及 HUD 抬头显示等高带宽视频应用中，DS90UB947-Q1 及其配套解串器将持续作为关键连接元件，支撑更高分辨率与更复杂的异构接口协议协同工作。通过系统级架构优化为下一代汽车电子提供更稳定、更可靠、更智能的显示传输解决方案。

FPD-Link III 接口能够通过同一条差分或单端链路同时传输视频、音频数据；以及实现全双工控制信号的传输（包括 I²C 和 SPI 通信）。OLDI 采用差分信号传输，具有共模抑制能力，能够有效抑制接地噪声和电磁干扰。通过采用两个差分对来整合视频数据和控制信号，能够有效降低互连线的尺寸与重量，从而简化系统设计。采用低压差分信令（LVDS）在单一差分链路上传输高达 WXGA 和 1080p 分辨率（24 位色深）的信号。

本文将从系统架构设计、软硬件设计指南以及调试流程等方面，系统地介绍如何设计一个标准化规范化的 FPD-Link III 系统。

1.1 系统设计

如 Figure 1 所示，DS90UB947-Q1 串行器需与 DS90UB940-Q1、DS90UB948-Q1、DS90UB926Q-Q1、DS90UB928Q-Q1 等解串器配合使用。该串行器及配套解串器集成 I²C 兼容接口，支持主机控制器对远端器件编程。此外，器件内置双向控制通道（Bidirectional Control Channel, 后续简称 BCC）支持串行器与解串器之间以及远程 I²C 从设备的通信。BCC 通过高速前向通道（串行器→解串器）的嵌入式信令与低速反向通道（解串器→串行器）的低速信令协同实现。

DS90UB947-Q1 将单路或双路 FPD-Link（Open LDI）接口（最高支持 8 条 LVDS 数据通道 + 1 个时钟）转换为 FPD-Link III 接口。串行数据流包含嵌入式时钟、视频控制信号、RGB 视频数据和音频数据，其有效载荷经过直流平衡处理以增强信号质量并支持交流耦合。

该器件可以支持 OLDI 时钟可达 25 MHz 至 96 MHz (单路 single port), 或 50 MHz 至 170 MHz (双路)。FPD-Link III 串行流速率最大为 3.36 Gbps (最低 875 Mbps), 或在两个通道上传输时, 每个通道最大为 2.975 Gbps (最低 875 Mbps)。

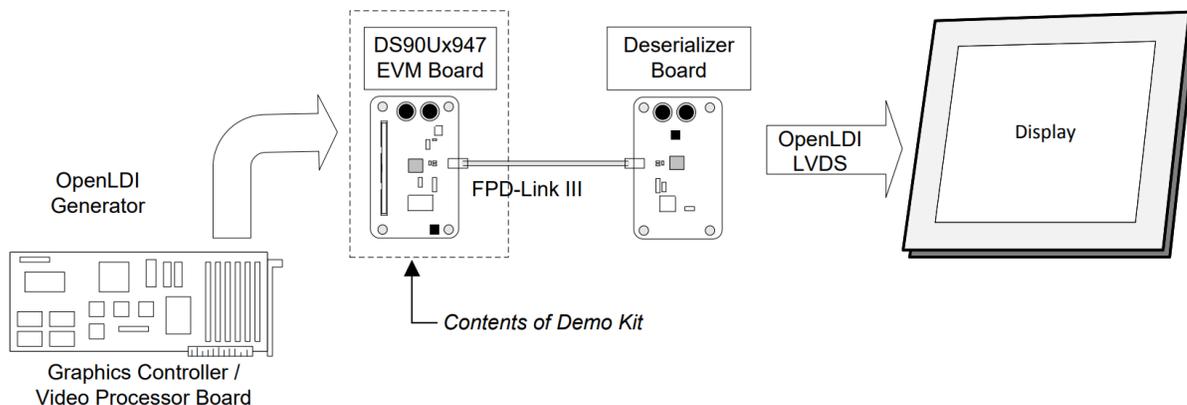


Figure 1. DS90UB947-Q1 经典应用系统

2 OpenLDI 协议

2.1 协议包

Open LVDS Display Interface (OLDI) 协议遵循 LVDS 物理层的基本设计。LVDS 是一种很常见的高速信号差分信号接口, 以其较低的功耗和较强的抗干扰能力使得 LVDS 成为许多应用的首选接口。OLDI 以 LVDS (ANSI/TIA-EIA-644) 作为物理层传输标准, 高速信号通常在 $100\ \Omega$ 差分阻抗负载下传输, 典型传输电流约 $3.5\ \text{mA}$, 差分电压约 $350\ \text{mV}$ 。其差分信号结构具备极佳的抗共模噪声能力与低 EMI 特性。为支持长线缆传输并减少直流偏移, OLDI 引入 DC 平衡机制: 在每一个传输周期中的七位串化数据流中, 有一位用于指示接收端是否需要反转其余六位, 以维持直流平衡。根据传输协议, 每对 LVDS 信号线可传输 6 位有效数据。LVDS 本身通过差分信号结构自带抗 EMI 机制, 而 OLDI 的 DC 平衡设计则进一步增强了其对直流失衡与长距离衰减的处理能力, 以适配 5 至 10 米长距离的视频链路。

DS90UB947-Q1 可配置 24bit (8bit-RGB) 编码方式, 具有 2 种不同的映射方案, 如 Figure 2 所示, 映射方案选择可以从 MODE_SEL [1:0] 引脚 (PIN18\32) 或寄存器 0x4F [7:6] 配置。每一帧对应一个像素时钟 (PCLK) 周期。CLK 的 LVDS 时钟输入遵循 4:3 占空比。

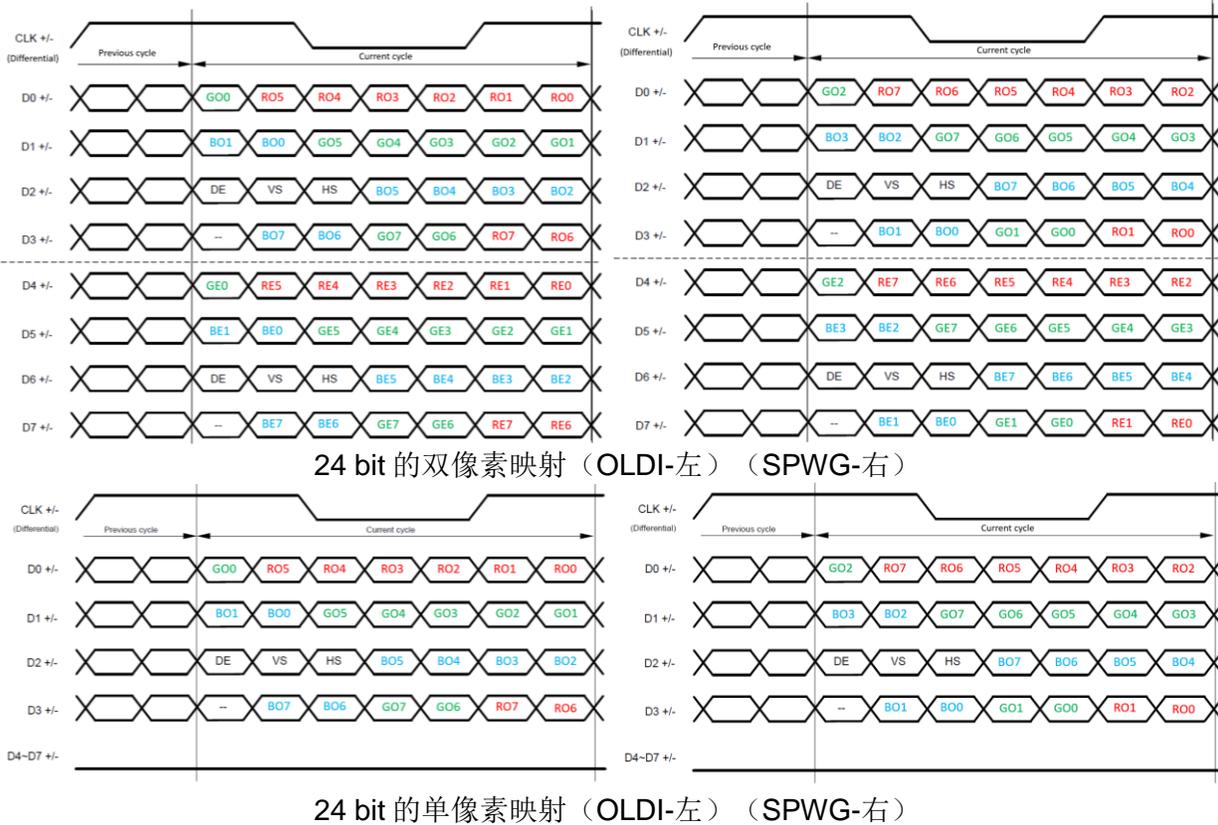


Figure 2. DS90UB947-Q1 支持的 OLDI 传输编码方式

3 基础设计规范

3.1 选型规范

DS90UB947-Q1 支持单路或双路的 FPD-Link III 输出；选型过程中的以下参数可参考评估：

像素时钟(Pixel clock-PCLK/MHz)单路支持 25-96MHz，双路支持 50-170MHz；像素时钟是显示系统中驱动像素数据传输的基准时钟信号。表示每秒可传输的像素数量（例：60MHz = 6 千万像素/秒）。每个时钟周期对应一个像素的数据传输（RGB/YUV 等色彩数据），即为有效区（Active）数据；而消隐区（Blanking）包含前后沿和同步脉冲等控制时序信号。FPS 为帧率。

$$\text{Pixel Clock} = (\text{Hactive} + \text{Hblanking}) * (\text{Vactive} + \text{Vblanking}) * \text{FPS}$$

链路速率(Line rate/bps)单路支持 0.875-3.36Gbps；双路支持最高 2.975Gbps/lane；FPD-Link III 的高速正向通道(forward channel)由 35bit 数据组成，其中包含从串行器传输到解串器的 RGB 数据，同步信号，I2C，GPIO 和 I2S 音频等。因此 line rate 和 PCLK 存在固定的 35 倍关系；

$$\text{Line rate} = \text{Pixel Clock} * 35$$

信号频率(Frequency/Hz)单路支持 0.4375~1.68GHz；双路支持最高 1.4875GHz/lane；因为 FPD-LINK III 采用 NRZ(Non Return Zero Code)编码方式，即通过恒定的电压电平直接表示二进制数据，则 FPD-Link III 的信号频率与链路速率存在固定的 2 倍关系：

$$\text{FPD Frequency} = \text{line rate} / 2$$

计算示例屏参，当帧率为 60Hz 时，PCLK = 2200*1125*60=148.5MHz；Line rate = 148.5*35=5.1975 Gbps；FPD 频率为 2.59875GHz，DS90UB947-Q1 可以在双路模式下支持该屏幕。

参数	参数名称	Min	Typ	Max	Unit
水平时序 Horizontal Timing	水平总像素 Horizontal total data	2000	2200	2400	DCLK
	水平有效像素 Horizontal active data		1920		
	水平前沿 Horizontal front porch	16	88	152	
	水平同步宽度 Horizontal sync width	16	44	128	
	水平后沿 Horizontal back porch	48	148	200	
	水平消隐总量 Horizontal blanking	80	280	480	
垂直时序 Vertical Timing	垂直总行数 Vertical total data	1100	1125	1150	H
	垂直有效行数 Vertical active data		1080		
	垂直前沿 Vertical front porch	2	4	10	
	垂直同步宽度 Vertical sync width	2	5	10	
	垂直后沿 Vertical back porch	16	36	50	
	垂直消隐总量 Vertical blanking	20	45	70	

Table 1. 常见 1080p 屏幕参数示例

3.2 硬件设计规范

DS90UB947-Q1 的经典应用如图所示，硬件原理图的经典设计请参考 DS90UB947-Q1 datasheet 8.1 章节，TI 官网提供了 EVM 板的用户设计指南，其中也包含了整体的电路设计以供客户参考；本章节将对 I2C、MODE_SEL、OLDI、FPD-Link 的 DOUT\RIN 的设计展开分析。

(1) I2C: I2C 广泛应用于电子设备之间，是一种同步半双工串行通信协议，即数据在同一时刻只能单向传输，但可以通过切换方向实现双向通信。通信过程中，主设备 (Master) 负责生成时钟信号 (SCL)，并控制通信的开始和结束。从设备 (Slave) 则根据地址被选中，并在需要时响应主设备的请求。如 Figure 3 左侧是经典 SerDes 的 I2C 通信总线。

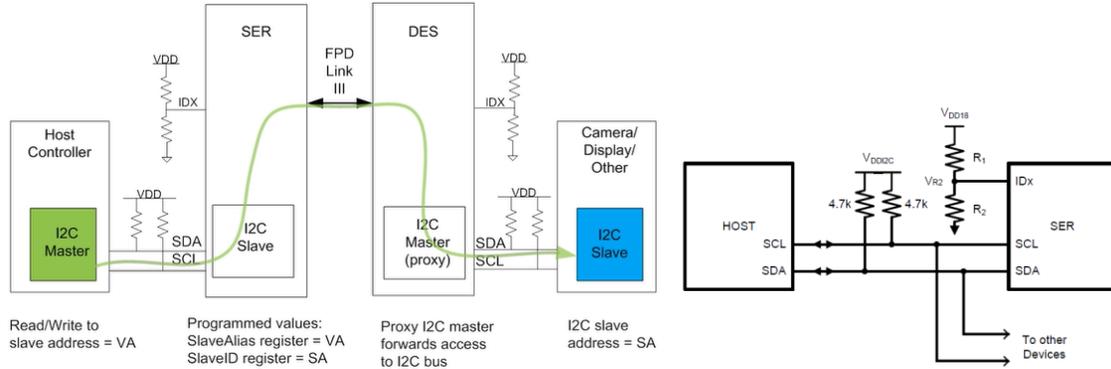


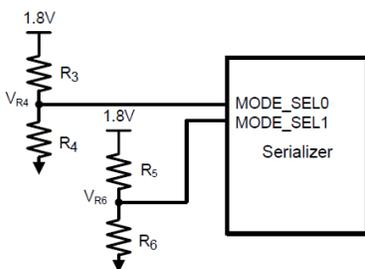
Figure 3. DS90UB947-Q1 经典 I²C 设计通路

为了正确与 I²C 从器件通信、主机发送从机地址并监控来自从机的响应 (ACK)。如果总线上从机被正确寻址、则会通过将 SDA 总线驱动为低电平来确认(ACK)主器件。如果地址与器件的从器件地址不匹配、它会通过让 SDA 拉为高电平来取消确认(NACK)主器件。947 的 I²C 可以通过上下拉电阻约束设备物理地址 (PIN47\48), 如表 3-1 所示。

#	RATIO V_{R2} / V_{DD18}	IDEAL V_{R2} (V)	SUGGESTED RESISTOR R1 kΩ (1% tol)	SUGGESTED RESISTOR R2 kΩ (1% tol)	7-BIT ADDRESS	8-BIT ADDRESS
1	0	0	Any value less than 100	40.2	0x0C	0x18
2	0.212	0.381	133	35.7	0x0E	0x1C
3	0.327	0.589	147	71.5	0x10	0x20
4	0.442	0.795	115	90.9	0x12	0x24
5	0.557	1.002	90.9	115	0x14	0x28
6	0.673	1.212	66.5	137	0x16	0x2C
7	0.789	1.421	21.5	80.6	0x18	0x30
8	1	1.8	Any value less than 100	OPEN	0x1A	0x34

Table 2. 串行通信协议 I2C 地址

(2) **MODE_SEL**: 947 的模式配置主要是约束了 OLDI 接口的以及 FPD-Link 的线缆类型选择; 可以通过 **MODE_SEL[0:1]** 输入引脚 (PIN18\PIN32) 的上拉电阻和下拉电阻阻值搭配或通过配置寄存器 (0x4F[6:7]\0xC2[5]\0x5B[7]) 来设置。关于如 Figure 4 的 R3\R4\R5\R6 的选择, 请查看下列表格:



MODE	SEL	功能
OpenLDI	0	单像素 OLDI 接口
	1	双像素 OLDI 接口
复制模式	0	关闭复制模式(参考 datasheet7.3.14)
	1	打开复制模式
OpenLDI Bit Mapping	0	OpenLDI bit 映射 (如 2.1 所述)
	1	SPWG bit 映射
线缆类型	0	STP 双绞线
	1	COAX 同轴线

Figure 4. DS90UB947-Q1 的模式设置

#	RATIO V_{R4}/V_{DD18}	TARGET V_{R4} (V)	SUGGESTED RESISTOR PULL-UP R3 kΩ (1% tol)	SUGGESTED RESISTOR PULL-DOWN R4 kΩ (1% tol)	OLDI_DUAL	REPEATER
1	0	0	OPEN	Any value less than 100	0	0
2	0.213	0.383	115	30.9	0	1
5	0.560	1.008	82.5	105	1	0
6	0.676	1.216	51.1	107	1	1

Table 3. 配置模式 MODE_SEL0

#	RATIO V_{R6}/V_{DD18}	TARGET V_{R6} (V)	SUGGESTED RESISTOR PULL-UP R5 kΩ (1% tol)	SUGGESTED RESISTOR PULL-DOWN R6 kΩ (1% tol)	MAPSEL	COAX
1	0	0	OPEN	Any value less than 100	0	0
2	0.213	0.383	115	30.9	0	0
3	0.328	0.591	107	52.3	0	1
4	0.444	0.799	113	90.9	0	1
5	0.560	1.008	82.5	105	1	0
6	0.676	1.216	51.1	107	1	0
7	0.792	1.425	30.9	118	1	1
8	1	1.8	Any value less than 100	OPEN	1	1

Table 4. 配置模式 MODE_SEL1

(3) OLDI/FPD-Link III: 系统设计人员应密切注意高速信号链规范，包括 OpenLDI 信号接口和 FPD-Link III 信号接口。LVDS 的电气特性应符合 TIA/EIA-644-A 标准及 OpenLDI 相关规范：注意采用差分布线，避免单端信号干扰。做好阻抗控制、等长布线、提供完整的电源和地参考平面，以减少噪声干扰。避免串扰：相邻信号线应避免平行走线，以减少串扰。TI 建议根据 947 及其配对解串器的数据手册要求，限定 OpenLDI 输入频率和抖动。DS90UB947-Q1 与其配对的解串器作为系统整体，其之间的 FPD-Link 通道必须符合 TI 的要求通道建议，以确保高速通道具有足够的链路质量。

另外请注意在 947-SNLS454A 的规格书中 STP 双绞线的推荐应用方案中，针对 940/948 的 C1-C4 推荐使用 0.033uF 的电容，此处为书写错误，本文进行修正如下，在搭配 940/948 应用时，TI 依然建议使用 0.033uF 或者 0.1uF 串联耦合电容吸收高速突发噪声，提供局部能量缓冲，如图 Figure 5。

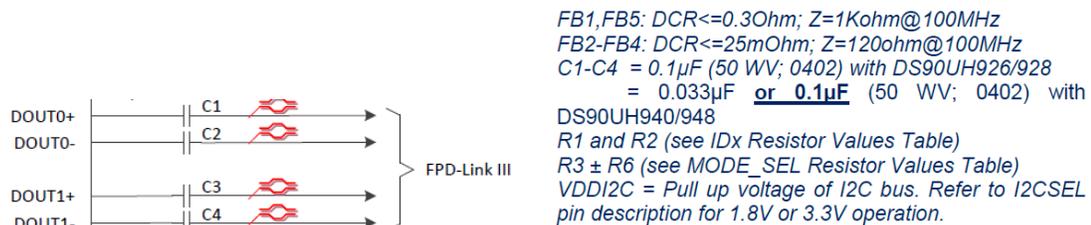
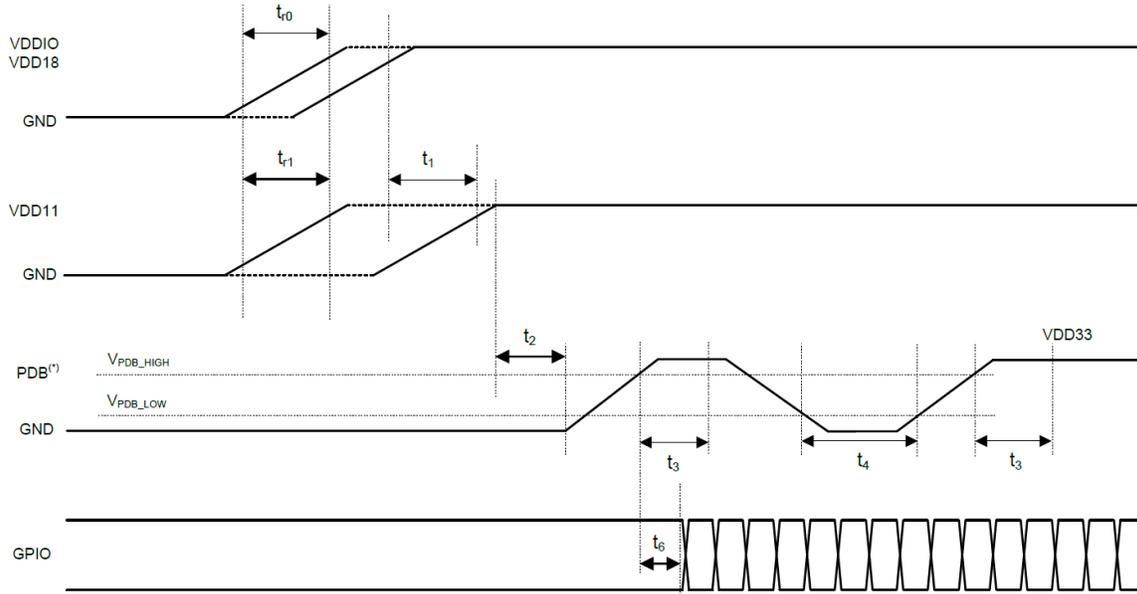


Figure 5. FPD-Link 信号线 DOUT 在 STP 双绞线下的推荐器件

(4) 上电时序：请严格按照下 Figure 6 的上电时序设计 DS90Ux947-Q1，建议使用微控制器（MCU）来拉高 PDB（高电平有效）信号，而非采用 RC 滤波网络，以确保在电源稳定之后能够正确时序地拉高 PDB 引脚。



^(*) It is recommended to assert PDB (active High) with a microcontroller rather than an RC filter network to help ensure proper sequencing of PDB pin after settling of power supplies.

Figure 6. DS90UB947-Q1 标准上电时序

关于初始化时序，947 可有以下三种选择；无论哪种都假设 OpenLDI 的时钟信号与数据信号同步到达：

选择一：OLDI 信号稳定之后再拉高 947 的 PDB；并在 t3 的延时后对 947 进行初始化配置；

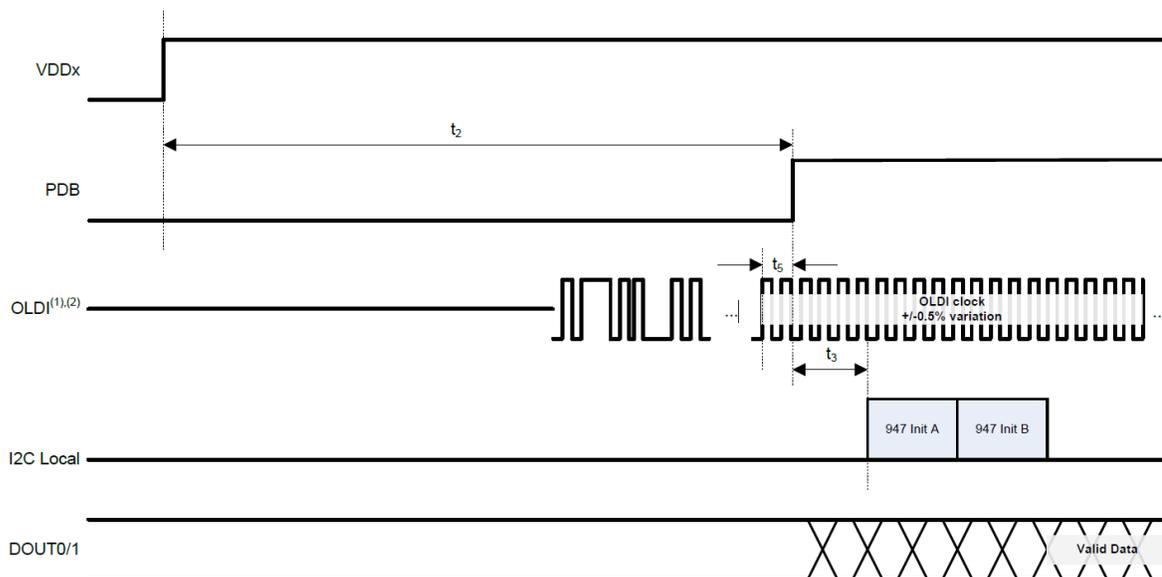


Figure 7. 初始化时序选择一

选择二：如果输入的 OLDI 时钟必须在 947 的 PDB 拉高之后才能输出，那么 TI 建议在 OLDI 时钟及数据输入稳定之后执行 PDB 复位，如 Figure 8；并在 t4 之后执行 3.3 代码里的 initA/initB。

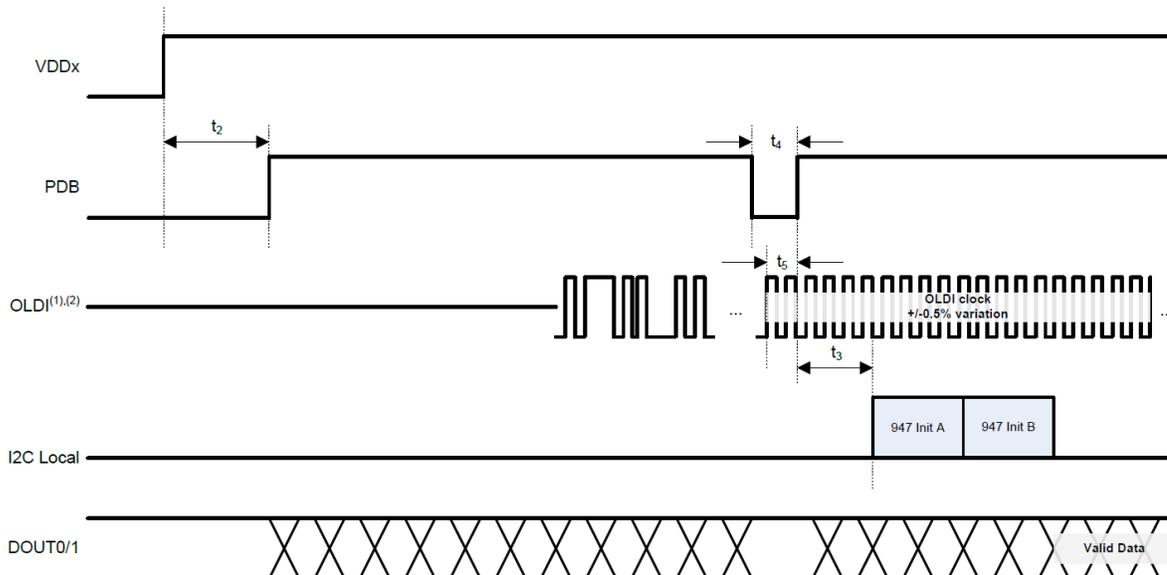


Figure 8. 初始化时序选择二

选择三：或者在 OLDI 时钟还未输出的时候执行 initA，之后让 OLDI 输出稳定的时钟和数据流，那么无需额外的 PDB 复位，在确保 OLDI 时钟稳定之后执行 initB 即可，请注意 eFuse 芯片无需 initB，关于 eFuse 的批次确认请联系 TI 支持团队，一般认为年份/月份代码 = 95 之后的芯片被认为是 eFuse 芯片。

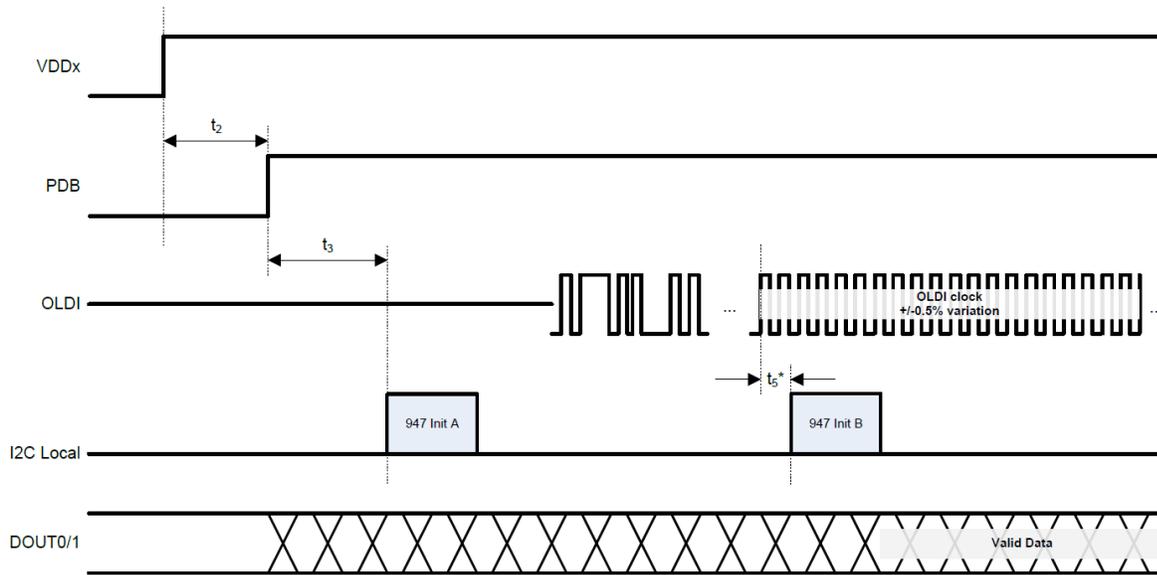


Figure 9. 初始化时序选择三

Symbol	Description	Test Conditions	Min	Typ	Max	Units
tr0	VDDIO, VDD18 Rise time	These time constants are specified for rise time of power supply voltage ramp (10% - 90%)	0.2		1.5	ms
tr1	VDD11 Rise time		0.04		1.5	ms
t1	VDD11 Delay time	Power supplies may be ramped simultaneously. If sequenced, VDD18/VDDIO ramps-up first	0			ms
t2	PDB delay time	PDB should be released after all supplies are stable	0			ms
t3	I2C Ready time	Starting from PDB high, the local I2C access is available after this time	2			ms
t4	Hard Reset time	PDB negative pulse width required for the device reset	2			ms
t5*	oLDI Clock Stable to Hard (PDB) Reset or PLL Reset (947 Init B) delay time	oLDI Clock must be within 0.5% of the target frequency and stable	1			μs
t6	PDB to GPIO delay time	Keep GPIOs low or high until after PDB release	0			ms

Table 5. 初始化时序指南

3.3 软件设计规范

3.3.1 参考代码

DS90Ux947-Q1 的代码请参考如下，示例为 947 与 948 在 Dual FPD-Link STP, 1920x1080@60Hz 模式下的配置代码；其中包含了 Errata 代码，核心为两段 initA 和 initB；

init A 中主要修正了 AV mute 的异常，这是因为在用 FPD-LINK III 器件传输视频数据的时候，在 Blanking 期间，如果 SOC 此时输出特定格式的视频数据（如 0x666666），就有可能导致 SerDes 系统进入 AV Mute 模式；在 AV Mute 模式下，SerDes 之间的 I2C 通信，GPIO 透传功能都不会有影响，但是解串器没有视频数据输出，只有 PCLK 时钟以及行场同步信号的输出，所以此时会没有显示，但是背光触摸等功能正常；通过 0x5B[5] = 0 可防止 947 器件进入 AV Mute 模式。

initB 中主要是温度补偿代码，首先通过 pollstate 函数轮询 947 的状态机，确保其达到正常状态之后，再进行 B 段代码配置，即启动系统时，读取外部温度传感器（SoC 板载传感器或主板中其他传感器）以确定 947 系统周围的启动环境温度。根据环境温度，SoC/MCU 将对 947 的 PLL page 的寄存器写入不同的值，请参考 initB 中注释；正确配置完 947 的 errata 之后，需要手动复位 OLDI PLL 模块（Errata #3）。

请注意任何对 947 或其配对的解串器进行复位操作都会清除这些软件配置，因此请确保每次复位后都重新配置了所有的 errata 代码；且注意代码中所有 PLL 的复位都会导致瞬态失锁（可恢复）；经测试这些软件 errata 将导致串行器/解串器链路锁定时间增加达 6.6ms。

```
#####
# DS90UB947-Q1 Init Code Example v1.1 2024
# 947->948, Dual FPD-Link STP, 1920x1080@60Hz
#####

import time
import sys

# System dependant variables - adjust according to system
UB947 = 0x18 # 947 I2C Address
UB948 = 0x68 # Remote 948 Address
StartAmbTemp = 25 # Example temperature measured by local sensor on PCB (Ambient temp around 947). In real system this should be read back
from temp sensor during initialization
Hres = 1920 # Expected horizontal active resolution
Vres = 1080 # Expected vertical active resolution

def pollstate():
    count=0
    board.WriteI2C(UB947,0x23,0x80)
    board.WriteI2C(UB947,0x24,0x80)
    mask = int('11111',2)
    state = board.ReadI2C(UB947,0x24,1)
    while (state & mask) != 27: #27 = 5'b11011
        time.sleep(0.01) #10ms delay
        state = board.ReadI2C(UB947,0x24,1)
        count = count +1
    if count == 10:
        print "Error: Serializer is not in Normal state"
        return(1)

    board.WriteI2C(UB947,0x23,0x0)
    board.WriteI2C(UB947,0x24,0x0)

def initsequence():

    board.WriteI2C(UB947,0x03,0xDA) # Set I2C passthrough

    linked = 0
    retry = 0
    mask = int('10000000',2)
    while linked == 0:
        STS = board.ReadI2C(UB947,0x5A,1)
        if ((STS & mask) != 0):
```

```

    linked = 1
    print "Deserializer Detected"
else:
    time.sleep(0.1)
    retry = retry + 1
    if retry == 10:
        print "Error: No Deserializer Detected"
        return()

board.WriteI2C(UB948,0x4A,0x01) # Disable OLDI outputs on 948 to prevent screen glitches from appearing during the init process

##### Init A Sequence

# Errata #5 and 6
board.WriteI2C(UB947,0x5B,0x03) # Force dual FPD-Link, disable reset on PLL frequency change

# Errata #10
board.WriteI2C(UB947,0x16,0x02) # Adjust BCC watchdog timer to minimum

# Errata #11
board.WriteI2C(UB947,0x04,0x90) # Prevent AVMUTE for non-HDCP systems

state_error = pollstate() # Poll 947 state machine prior to applying init B errata

if state_error == 1:
    return()

##### Init B Sequence

## Errata #2 (only for older non-eFuse devices)
# board.WriteI2C(UB947,0x40,0x10)  #//set Page to oLDI PLL registers
# board.WriteI2C(UB947,0x41,0x4A)
# board.WriteI2C(UB947,0x42,0x3F)
# board.WriteI2C(UB947,0x41,0x4B)

# if StartAmbTemp < 10:  #//based on temperature reading from external sensor:
#   board.WriteI2C(UB947,0x42,0x88)
# else:
#   board.WriteI2C(UB947,0x42,0x89)
#   board.WriteI2C(UB947,0x41,0x49) #Reset oLDI PLL
#   board.WriteI2C(UB947,0x42,0x10)
#   board.WriteI2C(UB947,0x42,0x00)
#   board.WriteI2C(UB947,0x40,0x14) #set Page to FPD PLL registers
#   board.WriteI2C(UB947,0x41,0x4A)
#   board.WriteI2C(UB947,0x42,0x3F)
#   board.WriteI2C(UB947,0x41,0x4B)
# if StartAmbTemp < 10:
#   board.WriteI2C(UB947,0x42,0x88)
# else:
#   board.WriteI2C(UB947,0x42,0x89)
#   board.WriteI2C(UB947,0x41,0x49) #Reset oLDI PLL
#   board.WriteI2C(UB947,0x42,0x10)
#   board.WriteI2C(UB947,0x42,0x00)

# Errata #3
board.WriteI2C(UB947,0x40,0x10)
board.WriteI2C(UB947,0x41,0x49)
board.WriteI2C(UB947,0x42,0x16)
board.WriteI2C(UB947,0x41,0x47)
board.WriteI2C(UB947,0x42,0x20)
board.WriteI2C(UB947,0x42,0xA0)
board.WriteI2C(UB947,0x42,0x20)
board.WriteI2C(UB947,0x42,0x00)
board.WriteI2C(UB947,0x41,0x49)
board.WriteI2C(UB947,0x42,0x00)

time.sleep(0.01)

```

```

##### Timing Confirmation

retry = 0
Hconfirm = 0
Vconfirm = 0
while (retry < 3):
    board.WriteI2C(UB948,0x68,0x08) # Enable PATGEN BIST
    time.sleep(0.1) # Critical time delay - do not adjust to a shorter value
    board.WriteI2C(UB948,0x68,0x19) # H active High monitor
    Hhigh = board.ReadI2C(UB948, 0x69, 1)
    board.WriteI2C(UB948,0x68,0x09) # H active Low monitor
    Hlow = board.ReadI2C(UB948, 0x69, 1)
    board.WriteI2C(UB948,0x68,0x39) # V active High monitor
    Vhigh = board.ReadI2C(UB948, 0x69, 1)
    board.WriteI2C(UB948,0x68,0x29) # V active Low monitor
    Vlow = board.ReadI2C(UB948, 0x69, 1)
    board.WriteI2C(UB948,0x68,0x00) # Disable PATGEN BIST

    mask = int('00111111',2)

    hlowmask = Hlow & mask
    hhighmask = Hhigh & mask
    vlowmask = Vlow & mask
    vhighmask = Vhigh & mask

    hhighmask = hhighmask << 6
    vhighmask = vhighmask << 6

    Hactive = hhighmask | hlowmask
    Vactive = vhighmask | vlowmask

    if Hactive != Hres:
        retry = retry + 1
        board.WriteI2C(UB947,0x40,0x10) # Reset OLDI PLL
        board.WriteI2C(UB947,0x41,0x49)
        board.WriteI2C(UB947,0x42,0x16)
        board.WriteI2C(UB947,0x41,0x47)
        board.WriteI2C(UB947,0x42,0x20)
        board.WriteI2C(UB947,0x42,0xA0)
        board.WriteI2C(UB947,0x42,0x20)
        board.WriteI2C(UB947,0x42,0x00)
        board.WriteI2C(UB947,0x41,0x49)
        board.WriteI2C(UB947,0x42,0x00)
        time.sleep(0.1)
    else:
        Hconfirm = 1
    if Vactive != Vres:
        retry = retry + 1
        board.WriteI2C(UB947,0x40,0x10) # Reset OLDI PLL
        board.WriteI2C(UB947,0x41,0x49)
        board.WriteI2C(UB947,0x42,0x16)
        board.WriteI2C(UB947,0x41,0x47)
        board.WriteI2C(UB947,0x42,0x20)
        board.WriteI2C(UB947,0x42,0xA0)
        board.WriteI2C(UB947,0x42,0x20)
        board.WriteI2C(UB947,0x42,0x00)
        board.WriteI2C(UB947,0x41,0x49)
        board.WriteI2C(UB947,0x42,0x00)
        time.sleep(0.1)
    else:
        Vconfirm = 1
    if (Hconfirm == 1) and (Vconfirm == 1):
        break

if (retry > 2):
    print "Error: Timing confirmation failed. Double check expected video resolution and OLDI video input"
    return()

```

```

else:
    print "Initialization Complete"
    board.WriteI2C(UB948,0x01,0x01) # Digital reset to restart AEQ
    time.sleep(0.1) # Allow time for 948 to relock
    board.WriteI2C(UB948,0x4A,0x00) # Enable OLDI outputs on 948

initsequence()
    
```

3.3.2 关键寄存器

947 可以通过主页寄存器 0x40\0x41\0x42 映射间接寄存器；关于上述代码中的关键寄存器补充解释如下：

Table 6. OLDI PLL Analog 间接寄存器

ADDRESS	REGISTER NAME	BIT	FIELD	TYPE	DEFAULT	DESCRIPTION
0x49	oLDI_PLL_STATE_MC_CTL	7:5	RESERVED	R	0	Reserved
		4	OLDI_STATE_MC_RESET	RW	0	Enable oLDI PLL reset state: 0 : Disable state machine reset (normal operation) 1 : Enable state machine reset
		3:0	OLDI_STATE_MC_CTL5	RW	0	0000 = RESET 0001 = INITIAL_VCO_SETTLING 0010 = SELECT_DIVIDER_PPM 0011 = FREQ_DAC_TIMER 0100 = FREQ_DAC_PPM 0101 = FREQ_DAC_SELECT 0110 = PFD_CLOSE_LOOP_TIMER 0111 = MONITOR_FREQ_LOCK 1000 = CHECK_FREQ_LOCK 1001 = FREQ_LOCK_TIMER
0x4A	oLDI_PLL_PPM_COUNTER					
		7:0	OLDI_PPM_CNTER_DPTH	RW	7	oLDI PLL reference clock to feedback clock frequency skew parts-per-million (PPM) counter depth 00: 8-bit counters 01: 9-bit counters 03: 10-bit counters 07: 11-bit counters 0F: 12-bit counters 1F: 13-bit counters 3F: 14-bit counters 7F: 15-bit counters FF: 16-bit coutners
0x4B	oLDI_PLL_SETTING					
		7	RESERVED	RW	1	
		6:4	RESERVED	RW	0	
		3:0	oLDI_INIT_LOOP_FLTR_VLTG	RW	0xA	oLDI PLL loop filter initial charge voltage setting 0: 0.244 1: 0.325 2: 0.407 3: 0.488 4: 0.569 5: 0.651 6: 0.732 7: 0.813 8: 0.895 9: 0.976 A: 1.057 B: 1.139 C: 1.22 D: 1.301 E: 1.383 F: 1.464

Table 7. FPD PLL Analog 间接寄存器

ADDRESS	REGISTER NAME	BIT	FIELD	TYPE	DEFAULT	DESCRIPTION
0x49	FPD_PLL_STATE_MC_CTL	7:5	RESERVED	R	0	Reserved
		4	FPD_STATE_MC_RESET	RW	0	Enable FPD PLL reset state: 0 : Disable state machine reset (normal operation) 1 : Enable state machine reset
		3:0	RESERVED	RW	0	Always write '0000b' to this field
0x4A	FPD_PLL_PPM_COUNTER					FPD PLL reference clock to feedback clock frequency skew parts-per-million (PPM) counter depth 00: 8-bit counters 01: 9-bit counters 03: 10-bit counters 07: 11-bit counters 0F: 12-bit counters 1F: 13-bit counters 3F: 14-bit counters 7F: 15-bit counters FF: 16-bit counters
		7:0	FPD_PPM_CNTER_DPTH	RW	7	
0x4B	FPD_PLL_SETTING					
		7	RESERVED	RW	1	
		6:4	RESERVED	RW	0	
						FPD PLL loop filter initial charge voltage setting 0: 0.244 1: 0.325 2: 0.407 3: 0.488 4: 0.569 5: 0.651 6: 0.732 7: 0.813 8: 0.895 9: 0.976 A: 1.057 B: 1.139 C: 1.22 D: 1.301 E: 1.383 F: 1.464
		3:0	FPD_INIT_LOOP_FLTR_VLTG	RW	0xA	

4 系统建立及调试指南

如 Figure 10 是由一对 DS90UB947-Q1 与 DS90UB948-Q1 组成的典型应用场景，实现从图形处理器到显示端的高带宽、低 EMI 视频传输链路。图形处理器（SoC/MCU）的输出采用符合 OLDI 标准的并行差分信号。串行器 DS90UB947-Q1 将输入的 OLDI 视频信号通过高速差分信号链路（FPD-Link III）进行串行化处理后输出。信号通过同轴电缆 COAX 或 STP 双绞线传输至 DS90UB948-Q1 解串器，完成端到端数据通信。在此过程中，除了视频数据外，链路还嵌入了 I²C 和 SPI 控制信号，借助内嵌通道（BCC）支持全双工控制通信。接收到串行数据的 DS90UB948-Q1 解串器将其还原为 OLDI 兼容的并行输出格式。该输出与输入一致，包含时钟与数据通道，可直接驱动 LVDS 接口的显示面板或另一个图形处理模块。I²C 用于寄存器配置与状态监测；GPIO/SPI 可拓展为通用控制通道，支持设备间握手、复位、同步等操作。不同电压域分别供电（如 1.8V, 1.2V, 3.3V 等），以匹配系统级功耗与兼容性需求。

本节将对系统搭建过程中可能遇到的问题进行详细阐述：

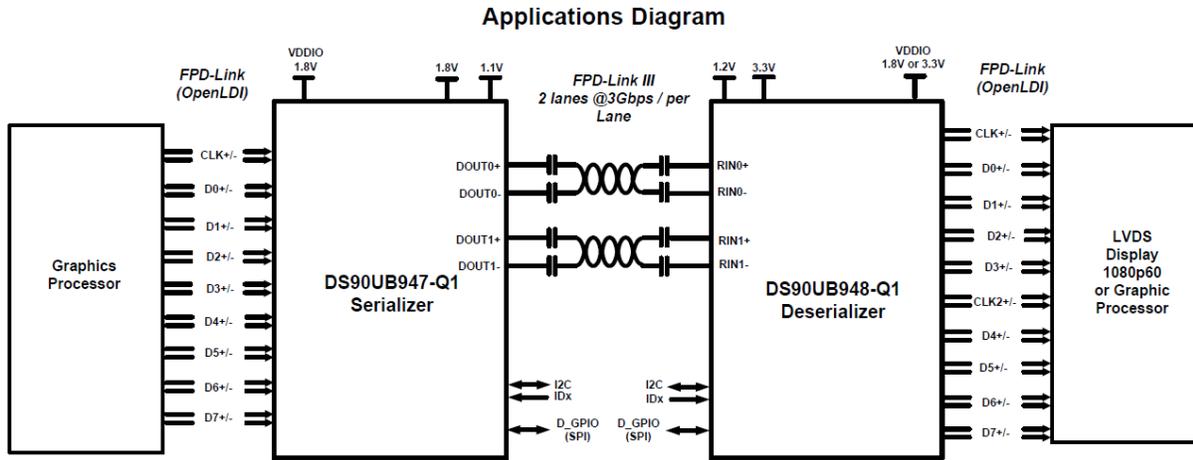


Figure 10. DS90UB947-Q1 与 DS90UB948-Q1 的典型应用场景

4.1 车载温度实验

4.1.1 PLL 设计

PLL 是一种通过反馈机制实现频率和相位锁定的电路，如 Figure 11，其核心组件包括相位检测器、环路滤波器、压控振荡器（VCO）和反馈回路。VCO 作为 PLL 的核心组件，通过控制电压调整输出频率，实现对信号的精确控制。

相位检测器（Phase Detector, PD）比较输入参考信号（如参考时钟）与 VCO 输出信号之间的相位差。如果两者存在相位差，相位检测器会输出一个误差信号。环路滤波器（Loop Filter, LPF）对相位检测器输出的误差信号进行平滑处理，生成一个控制电压（Vcont），用于调整 VCO 的频率。压控振荡器（VCO）的输出频率由输入的控制电压（Vcont）决定。当控制电压变化时，VCO 的输出频率也会相应变化。反馈回路：VCO 的输出信号通过设置 R counter/N counter（分频器），通过负反馈鉴相电路调整环路电压，确保 VCO 输出频率锁定到参考时钟。反馈到相位检测器，形成一个闭环系统。通过不断调整 VCO 的频率，使输出信号与输入参考信号的相位和频率保持一致。

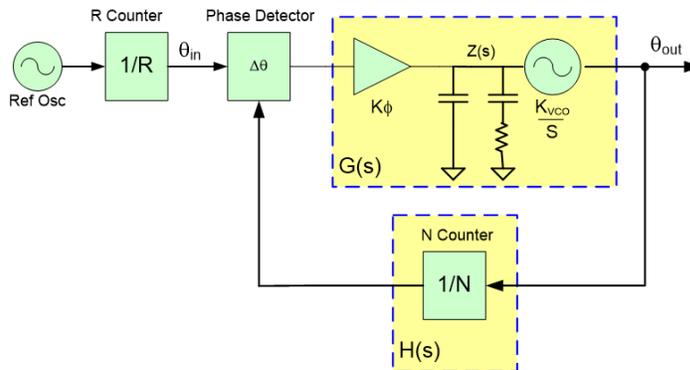


Figure 11. 锁相环 PLL 设计电路

4.1.2 温度设置

当器件周围的环境温度变化约 60° C 或以上时，相位锁定环（PLL）中的压控振荡器（VCO）频率有极低概率发生漂移。由于该频率漂移，系统显示可能出现图像失真或闪烁等异常。为了解决这一问题，自 2019 年 5 月起，TI 在芯片制造中为器件引入了额外的修正以解决温度爬升引发的问题。自 2019 年 5 月（年份/月份代码 = 95）及之后生产的器件，无需采用以下的软件优化方案（initB），即可达到数据手册中的温度爬升性能（最优性能）。

完整的 initB 代码在 3.3 节中已有示例代码，本节将对其中的温度优化代码进行详细阐述：

```
board.WriteI2C(UB947,0x40,0x14)
board.WriteI2C(UB947,0x41,0x4B)
if ambient temp < 10°C
    board.WriteI2C(UB947,0x42,0x88)
else
    board.WriteI2C(UB947,0x42,0x89)
```

上述软件优化可确保器件在冷启动条件下，在环境温度连续漂移范围为 -20° C 至 80° C 时正常工作；在热启动条件下，则支持从 80° C 降至 25° C 的温度变化过程。若希望在热启动条件下实现更宽的温度范围内稳定运行，则应优化为下列代码：

```
board.WriteI2C(UB947,0x40,0x14)
board.WriteI2C(UB947,0x41,0x4B)
if ambient temp < 10°C
    board.WriteI2C(UB947,0x42,0x88)
if 10°C ≤ ambient temp is < 60°C
    board.WriteI2C(UB947,0x42,0x89)
if ambient temp is ≥ 60°C
    board.WriteI2C(UB947,0x42,0x8A)
```

如上代码中，通过主页寄存器 0x40 选择间接寄存器的目录，通过间接寄存器 0x4B 设置系统启动时的 VCO 控制电压；相同的输入数据速率（f）下，如果 0x4B 初始值写入不同，则初始的压控电压有所不同，如下图的 A, B, C 点。假定 0x4B = 0x8A，则初始电压为点 B，VCO 选择了 VCO_x。由于半导体振荡电路的温度特性，VCO 的频率随着温度升高，相同的压控电压下，振荡频率会下降（意味着 VCO_x 的压控曲线将由左下图红色变为绿色）；但输入数据速率保持不变，为了保持锁定这个数据频率，维持环路稳定，锁相环的反馈电路会自动升高压控电压到点 D，保持电路稳定。当温度降低时反之亦然。当温度升高（或者降低）到一定范围，VCO_x 的压控电压已经到达电源轨的极限（A/C 点），VCO_x 以及环路将不能锁定，电路会失锁并选择下一个 VCO_{x+1} 自动锁定，且初始电压仍然由 0x4B 的寄存器设置决定。

在 Errata 中，为了保证从低温启动到高温的变化当中，不会出现闪屏、水波纹屏等，TI 建议初始化的 VCO 压控电压选择偏小一点，以确保选择的 VCO 有足够的余量空间来保证在 -40~105°C 内不会出现 CDR VCO 环路失锁。

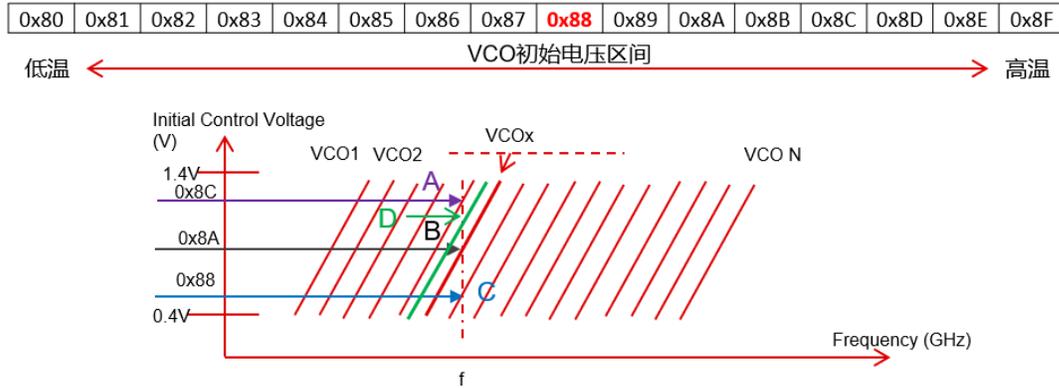


Figure 12. 压控振荡器 VCO 的压控曲线及初始电压区间

请注意数据表 SNLS454A 在 TCHL1 和 TCHL2 的说明中存在拼写错误。文本应显示“... 环境温度下降 ambient temperature is failling...”而不是“... 环境温度上升...”。此拼写错误将在下一个数据表修订版中更正。更正后的文本如下表所示：

T _{CLH1}	Allowable ending ambient temperature for continuous PLL lock when ambient temperature is rising under the following condition: -40°C ≤ starting ambient temperature (T _s) < 0°C. ⁽¹⁾	T _s	80	°C
T _{CLH2}	Allowable ending ambient temperature for continuous PLL lock when ambient temperature is rising under the following condition: 0°C ≤ starting ambient temperature (T _s) ≤ 105°C. ⁽¹⁾	T _s	105	°C
T _{CHL1}	Allowable ending ambient temperature for continuous PLL lock when ambient temperature is falling under the following condition: 45°C < starting ambient temperature (T _s) ≤ 105°C. ⁽¹⁾	25	T _s	°C
T _{CHL2}	Allowable ending ambient temperature for continuous PLL lock when ambient temperature is falling under the following condition: -20°C ≤ starting ambient temperature (T _s) ≤ 45°C. ⁽¹⁾	T _s - 20	T _s	°C

Table 8. DS90UB947-Q1 温度循环修正描述（数据手册 6.3）

如 Figure 13 所示，VCO 的频率漂移在温度突变下会有低概率锁相环 LOCK 丢失（PLL unlock），进而影响整个系统的数据完整性与显示质量，一般表象为闪烁、图像抖动或花屏等异常现象。针对这种情况，系统通常需要评估和限定器件在不同温度爬升/下降过程中的 PLL 能力。因此 947 定义了多个场景下 PLL 能够保持连续锁定的环境温度条件（T_s）。当环境温度由低温段（-40° C ≤ T_s < 0° C）开始上升时，PLL 能保持锁定状态，直到最高温度 80° C。起始温度处于中温段（0° C ≤ T_s ≤ 105° C）范围内，升温过程中 PLL 可一直保持锁定，直到 105° C。起始温度为高温段（45° C ≤ T_s ≤ 105° C），在降温过程中，最低可容忍环境温度为 25° C。高温热启动情况下，PLL 对降温较为敏感，尤其在电路材料热胀冷缩引发的参数偏移下，必须设定一个下限。中低温段启动（-20° C ≤ T_s ≤ 45° C），可接受的降温幅度为最多 20° C。此为一个相对温差指标，意味着系统能容忍的降温梯度为 20° C，而非绝对温度值。

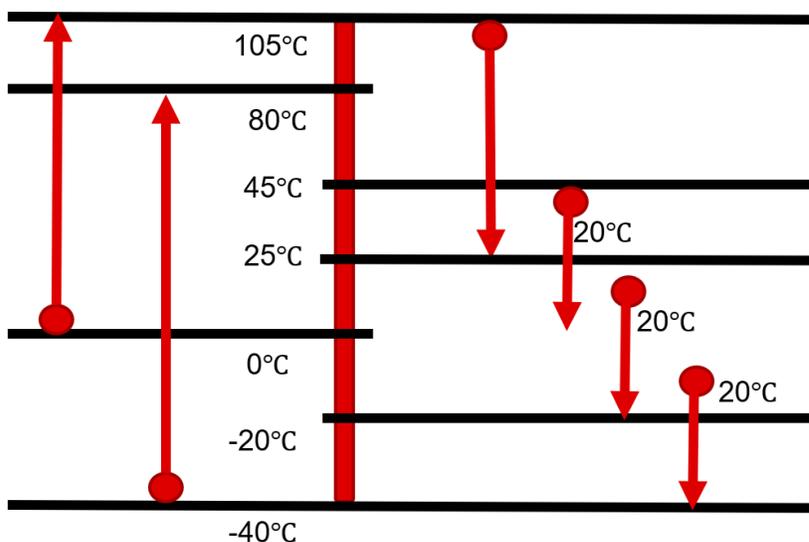


Figure 13. DS90UB947-Q1 温度上升或下降实验可满足的温度范围

4.2 系统调试框图

请设计 SerDes 的工程师始终从系统级角度考虑视频显示链路；如 Figure 14 列出来在异常情况下的调试思路；本节将分点阐述；

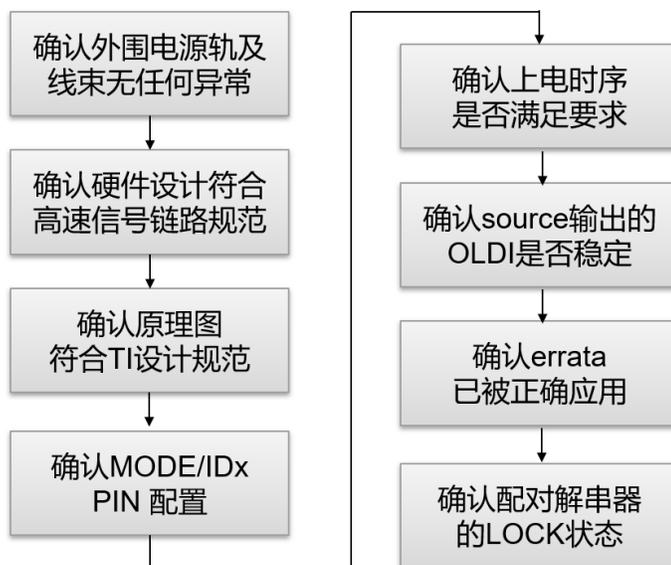


Figure 14. DS90UB947-Q1 系统调试指南

在调试过程中也可以使用 TI 提供的 DS90UB947-Q1 EVM（评估板）进行对比测试；使用 ALP 软件可简化寄存器配置验证过程。

调试思路	测试方法
------	------

<p>(1) 确认外部电源轨与线路完整性</p>	<ol style="list-style-type: none"> 1. VDD 等供电引脚必须满足启动时间和稳压精度要求； 2. 电源轨噪声 满足数据手册第 6 章要求； 3. 电源引脚旁路电容合理布局； 4. 无短路或浮空情况。 	<ol style="list-style-type: none"> 1. 使用示波器检查电源上升顺序及其纹波； 2. 推荐使用开关电源+LDO 组合，提升干净度； 3. 加装 TVS 管防止突波。
<p>(2) 确认硬件布线是否符合高速信号链路要求</p>	<p>关键信号：</p> <ol style="list-style-type: none"> 1. OpenLDI 接口：CLK±、D0± ~ D7±，每对需匹配 100Ω 差分阻抗； 2. FPD-Link 串行输出：DOUT0±、DOUT1±，需要控制长度、串扰、差分对平衡。 	<ol style="list-style-type: none"> 1. 差分对长度匹配误差 ≤ 100mil； 2. 差分走线间距 3W 原则（差分间距 ≥ 3×线宽）； 3. 使用连续地参考层并加接地孔围栏。
<p>(3) 确认原理图是否符合 TI 官方推荐设计</p>	<ol style="list-style-type: none"> 1. MODE、IDx 引脚拉高/拉低方式； 2. 电源分布网络（PDN）； 3. I2C 地址配置、REM_INTB 控制逻辑； 4. GPIO 功能复用与配置。 	<ol style="list-style-type: none"> 1. 对照 DS90UB947-Q1 数据手册和 EVM 原理图逐项比对； 2. 使用 TI 提供的 ALP（Analog LaunchPad）软件工具检查器件 ID、版本号和配置寄存器。
<p>(4) 确认 MODE 和 IDx PIN 配置是否匹配系统</p>	<ol style="list-style-type: none"> 1. MODE[2:0] 决定视频流输入格式（OpenLDI/TTL）、通道数（single/dual）、位宽； 2. IDx 决定设备 I2C 地址； 3. 错误配置将导致串行器无法锁定或无法正确通信。 	<ol style="list-style-type: none"> 1. 使用万用表或逻辑分析仪检查上拉/下拉电阻是否如设计； 2. 开机后通过 I2C 读取配置寄存器，确认生效。
<p>(5) 确认上电时序是否满足要求</p>	<p>参考 3.2 上电时序要求</p>	<ol style="list-style-type: none"> 1. 使用示波器或逻辑分析仪抓取波形并对比数据手册要求； 2. 时序异常常导致器件无法初始化或进入错误状态机。
<p>(6) 确认 source 输出的 OLDI 信号稳定性</p>	<p>关注参数：</p> <ol style="list-style-type: none"> 1. OpenLDI 输出频率（如 74.25MHz）是否恒定； 2. 差分信号摆幅、对称性、边沿抖动； <p>常见错误：</p> <ol style="list-style-type: none"> 1. GPU 输出默认不开启 OLDI； 	<ol style="list-style-type: none"> 1. 使用高速示波器测量 CLK± 和 D0±； 2. 使用眼图测试确保信号质量通过；

	2. 连线错误导致相位颠倒、串扰或丢包。	
确认 errata 是否正确应用	参考 3.3 及 4.1.2	请严格按照初始化代码配置
确认与解串器配对状态 (LOCK)	<ol style="list-style-type: none"> 1. LOCK 引脚应在初始化后拉高; 2. 通过寄存器 0x1E[0] (LOCK 状态位) 确认; 3. I2C 可通信表示链路成功握手。 	LOCK 失锁常见原因: <ol style="list-style-type: none"> 1. DOUT 差分信号未连接或反接; 2. 解串器处于错误 MODE; 3. PLL 未锁定, 温度波动超限。

4.3 生成测试图片 PATGEN

测试图片生成器 (Pattern Generator-PATGEN) 是有效的隔离问题, 调试系统的手段。分为外部时序模式与内部时序模式。

在外部时序模式下, 图案发生器会基于输入信号中的 DE (Data Enable) 与 VS (Vertical Sync) 信号自动检测当前视频帧的时序结构。若 VS 引脚未检测到垂直同步信号, 器件将通过统计连续无效像素时钟 (即 DE = 0 的周期数), 并判断其是否超过当前有效行长度的两倍, 从而推断出垂直消隐区 (Vertical Blanking) 的边界。在内部时序模式中, 图案发生器依据控制寄存器中预设的视频参数进行图像时序生成。此外, 内部时序模式下也可选择由外部输入时钟驱动。947 经典的测试图案如 Figure 15 所示。

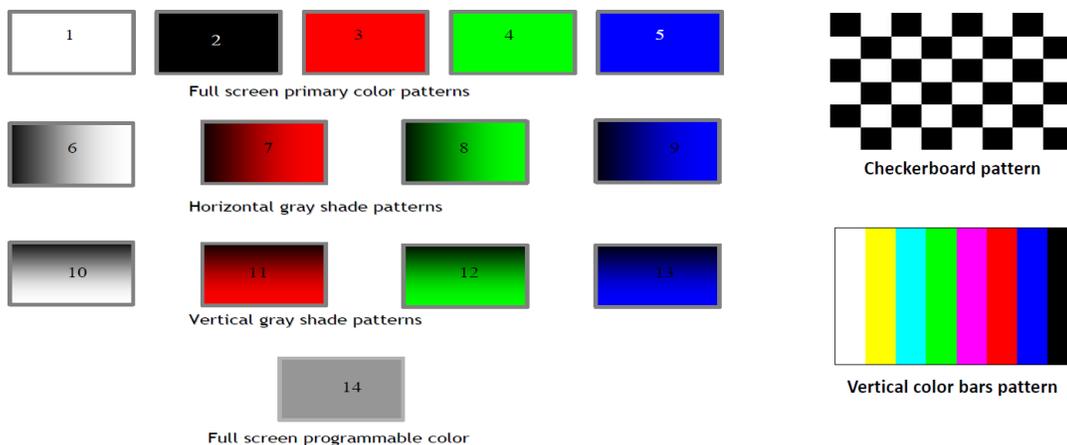


Figure 15. 测试图片生成器 (PATGEN) 的经典图案

下列代码是以外外部 PCLK 参考时钟为例输出 1920x1080@60Hz 的测试图案; 如需修改屏幕参数或选择不同的参考时钟, 请查看代码注释修改对应寄存器。

```

import UB947=0x18
board.WriteI2C(UB947,0x66,0x07) # enable PGAFS1
board.WriteI2C(UB947,0x67,0x80) # Active Horizontal Width.
board.WriteI2C(UB947,0x66,0x08) # enable PGAFS2
board.WriteI2C(UB947,0x67,0x87) # Active Vertical and Horizontal Widths
board.WriteI2C(UB947,0x66,0x09) # enable PGAFS3
board.WriteI2C(UB947,0x67,0x43) # Active Vertical Width
    
```

```

board.WriteI2C(UB947,0x66,0x04) # enable PGTF1
board.WriteI2C(UB947,0x67,0x98) # Total Horizontal Width
board.WriteI2C(UB947,0x66,0x05) # enable PGTF2
board.WriteI2C(UB947,0x67,0x58) # Total Vertical and Horizontal Widths
board.WriteI2C(UB947,0x66,0x06) # enable PGAF3
board.WriteI2C(UB947,0x67,0x46) # Total Vertical Width
board.WriteI2C(UB947,0x66,0x0C) # enable PGHBP
board.WriteI2C(UB947,0x67,0x94) # Horizontal Back Porch Width
board.WriteI2C(UB947,0x66,0x0D) # enable PGVBP
board.WriteI2C(UB947,0x67,0x24) # Vertical Back Porch Width
board.WriteI2C(UB947,0x66,0x0A) # enable PGHSW
board.WriteI2C(UB947,0x67,0x2C) # Horizontal Sync Width
board.WriteI2C(UB947,0x66,0x0B) # enable PGVSW
board.WriteI2C(UB947,0x67,0x05) # Vertical Sync Width
board.WriteI2C(UB947,0x66,0x0E) # enable PBSC
board.WriteI2C(UB947,0x67,0x00) # horizontal and vertical sync polarities to "Positive"

board.WriteI2C(UB947,0x65,0x0C) # set external clock and internal PATGEN timing
board.WriteI2C(UB947,0x64,0x05) # enable PATGEN with color bars

```

5 总结

本文围绕 TI 推出的车规级串行器芯片 DS90UB947-Q1，从系统架构设计、OpenLDI 协议介绍、硬件电路规范、软件初始化方法、PLL 温漂特性、关键调试要点及 PATGEN 功能等多个方面，阐述了其在 FPD-Link III 架构下的应用实践。

6 参考文献

ZHCAA76-深入理解 FPD-LINK 产品的速率评估方法

LVDS Owner's Manual Design Guide, 4th Edition [M/OL]. Texas Instruments.

SNLA359 – DS90Ux947-Q1 Startup Procedure

DS90UB947-Q1, DS90UH947-Q1 Errata

SNLA132-Exploring Int Test PatGen Feat of 720p FPD-Link III Devices

SNLS454A- DS90UB947-Q1 1080p OpenLDI to FPD-Link III serializer

重要通知和免责声明

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

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

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

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

TI 反对并拒绝您可能提出的任何其他或不同的条款。

版权所有 © 2025，德州仪器 (TI) 公司

最后更新日期：2025 年 10 月