HiSPi Solution Application Guide

(AN04007, V3.0)

(19.05.2021)

Shenzhen Pango Microsystems Co., Ltd.

Revisions History

Document Revisions

Version	Date of Release	Revisions
V3.0	19.05.2021	Initial release.

About this Manual

Terms and Abbreviations

Terms and Abbreviations	Meaning
HiSPi	High-Speed Serial Pixel interface

Table of Contents

Tables

Figures

Chapter 1 Overview

1.1 Introduction

This document is an application solution for HiSPi in FPGA/CPLD products from Shenzhen Pango Microsystems Co., Ltd. It primarily introduces the function list, interface definition, interface timing, supported devices, and reference designs of the HiSPi solution.

1.2 Main Functions

The main functions supported by HiSPi are as follows:

- Number of supported channels are configurable, up to a maximum of 24 lanes
- Automatic alignment of data channels
- Manual adjustment of data channel alignment
- 1:12 serial-to-parallel parsing
- Packetized-SP mode
- Parallel video stream output interface

1.3 Design Information

Table 1-1 Design Information

HiSPi Solution	
Supported Devices	Logos2 FPGA
Supported User Interface	Parallel User Interface
Provided Design Files	
Design File	Encrypted file
Reference Designs	RTL file
Simulation File	Encrypted file
Constraint File	FDC file
Development Tools	
Design Tools	PDS Development Suite Pango Design Suite2022-2-sp6.8

1.4 Resource Usage

Table 1-2 Resource Utilization Overview

Device Family	Configuration	LUT	Register	DRM
PG2L100H	Enable AUTO_TRAINING (Automatic alignment data window), can input 12 lanes	7020	12158	7
	Manual adjustment of sampling window, can input 12 lanes	4709	10030	7

Chapter 2 Function Introduction

The HiSPi protocol process module will extract the HiSPi interface's video data, send it to the video processing module for processing, then to the display driver module, and finally to the display equipment for display.

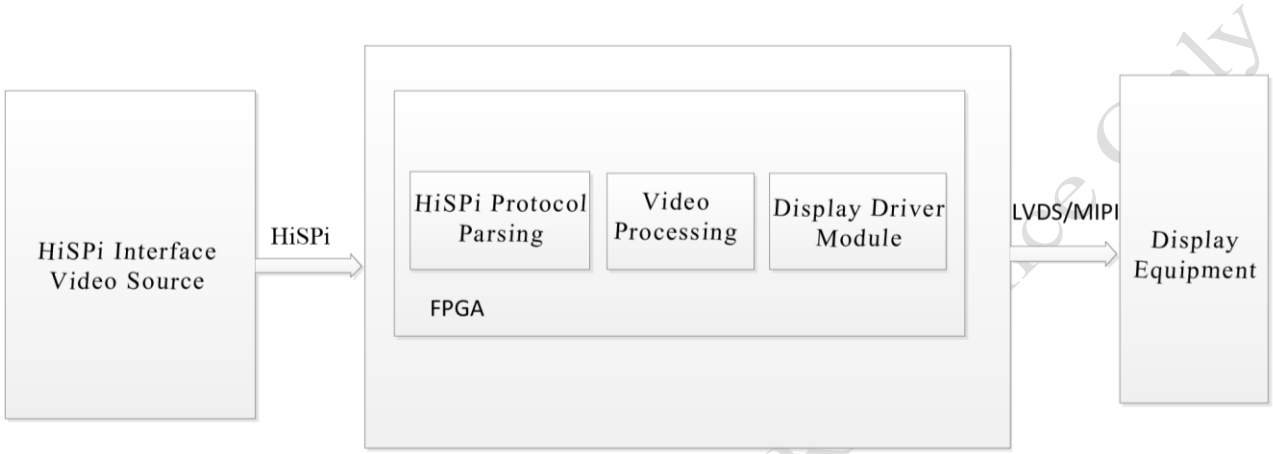

Figure 2-1 Application Introduction

2.1 Design Architecture

Figure 2-2 HiSPi Design Architecture

The HiSPi module includes the following submodules:

ides_phy_io_rx module: Implements deserialization of serial data, using a 1:4 deserialization

method here; gearbox_4 to 12 module: Converts 1:4 deserialized data into 1:12 parallel data;

hispi_training module: Implements automatic data sampling window detection and automatic alignment;

hispi_wordalign module: Implements HiSPi byte alignment;

hispi_ch_bonding module: Implements channel alignment for odd and even number array data pairs;

hispi_decoder module: Implements synchronous signal and video signal separation;

lane_remap module: Implements lane remapping, uniformly converting to 12 regions for processing;

pixel_remap module: Implements pixel reordering;

clk_and_rst module: Provides clock and reset signal;

2.2 Interface List

Table 2-1 List of Interfaces of the HiSPi Top-level Module

Signal Name	I/O	Bit width	Description
Clock Reset Signals			
rx_sysclk	I	1	Parallel byte clock that is high-speed clock rx_ioclk0 divided by 6
rx_io_rstn	I	1	ISERDES reset signal, active-low
rx_rstn	I	1	Input logic reset signal, active-low
key_in_n	I	1	Debugging button, reserved
rx_ioclk0	I	1	Input high-speed clock, which drives even pair data
rx_ioclk0_div	I	1	Input high-speed division clock, generated by dividing rx_ioclk0 by 2
rx_ioclk1	I	1	Input high-speed clock, which drives odd pair data
rx_ioclk1_div	I	1	Input high-speed division clock, generated by dividing rx_ioclk1 by 2
rx_data_p	I	24	HiSPi high-speed data, p side
rx_data_n	I	24	HiSPi high-speed data, n side
Configuration Interfaces			
config_clk	I	1	Configuration clock, 50MHz
config_rst_n	I	1	Configuration module reset signal, active-low
spi_clk	I/O	1	Output SPI clock

Signal Name	I/O	Bit width	Description
spi_cs_n	I/O	1	Output SPI chip select signal, active-low
spi_do	I/O	1	Output SPI data signal
spi_di	I	1	Input SPI data signal
i2c_scl	O	1	Output I2C signal
i2c_sda	I/O	1	Input/Output I2C data signal
sensor_rstn	O	1	Output camera reset signal, active-low
init_done	O	1	Initialization complete indicator signal, a high level indicates that the camera initialization is complete
HiSPi User Interface Signals			
line_test	O	1	Reserved
video_data	O	12*24	Parallel video data signal, each piece of valid data contains 24 continuous pieces of raw12 pixel data
video_vld	O	1	Data enable, active-high, and is a one-clock delay of rd_en
video_eol	O	1	Video line end flag, lasting one rd_clk, active-high
video_sof	O	1	Video frame synchronous signal, lasting one rd_clk, active-high
rd_clk	I	1	Read clock for line buffer RAM
rd_en	I	1	Read enable for line buffer RAM, active-high
wr_oneline_finish	O	1	After writing a line of video data signal to line buffer RAM, assert signal for one rd_clk cycle, active-high

2.3 Parameter Definitions

Table 2-2 HiSPi Top-Level Reference Definition

Parameter	Description
FREQ_CLK	Clock frequency for the configuration module, with a default value of 50,000,000 (representing 50MHz)
FREQ_SPI	SPI interface frequency, with a default value of 1,000,000, representing the SPI clock frequency of 1MHz
CHANNEL	HiSPi data channel number, optional 6/12/24 Default value = 12
DATA_WIDTH	Serialization width of the HiSPi protocol Fixed to 12
H_OUTPUT_SIZE	Horizontal display size of the video source Default value = 4176
V_OUTPUT_SIZE	Vertical display size of the video source Default value = 3088
RD_ADDR_BASE	Read start address for line buffer RAW, with the value ranging from 0~173 and each address of the line buffer RAM storing 24 pieces of raw12 pixel data. For example, if the start address value is 4, then the preceding 0~3 addresses (4*24=96 pixels) will be truncated. Default value = 8'h00
RD_NUM	The number of data read from the line buffer RAM per line, maximum 4176/24=174, the default value is 160, representing 24*160=3840 pixels per line

Parameter	Description
AUTO_TRAINING	Automatic training enable, optional "TRUE", "FALSE" If "TRUE", automatic training is enabled to automatically scan data sampling window; If "FALSE", the data sampling window needs to be searched manually. When "TRUE", users must ensure within the time of HTOTAL * VTOTAL / CHANNEL parallel clocks, there are equal to or more than (VTOTAL - VBLANK - FRAME_MARGIN) SOL signals of the HiSPi protocol (12"h001, 12"h000, 12"h000, 12"hfff) Default value = "FALSE"
HTOTAL	Total number of columns in the horizontal direction, including the blanking period Default value = 4176+1344
VTOTAL	Total number of rows in the vertical direction, including the blanking period Default value = 3088+50
VBLANK	Vertical blanking rows Default value = 50
FRAME_MARGIN	Margin time for one frame, in rows Default value = 450
CH_BONDING	Channel alignment enable signal, optional "TRUE", "FALSE" If "TRUE", enable channel alignment; if "FALSE", do not enable channel alignment; default value="TRUE"
DELAY_D_VALUE	The static delay value of the data channel, effective when AUTO_TRAINING is"FALSE". Each channel has a value of 0–247, and each step delay is 10ps; this value affects the data sampling window and can use either automatic training or manual methods to find the optimal sampling window value. Default value= {{(CHANNEL/2){8'h7c}}, {(CHANNEL/2){8'h90}}}
ISERDES_WIDTH	Serialization ratio width used by ISERDES Fixed to 4
ISERDES_MODE	ISERDES_MODE deserialization mode Fixed to "DDR1TO4"

2.4 Interface Timing

The deserialized user interface timing is shown in Figure 3. video_sof assert for one clock cycle indicates the start of a frame. Every time wr_oneline_finish assert, indicate one line data is written into the line buffer RAM. When the user initiates a read request by asserting rd_en, through video_data interface, and the data is valid when video_vld is high. video_eol asserts in the end of read data.

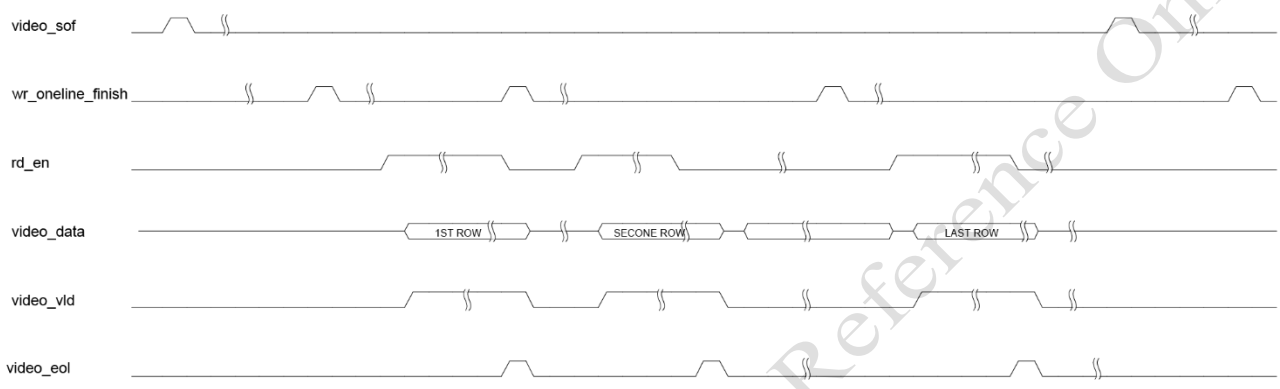

Figure 2-3 Top-level Interface Timing

Chapter 3 Example1

3.1 Example Function Design

The block diagram for reference design 1 is shown in Figure 3-1. The images captured by the camera are processed by the FPGA and displayed on the LVDS panel.

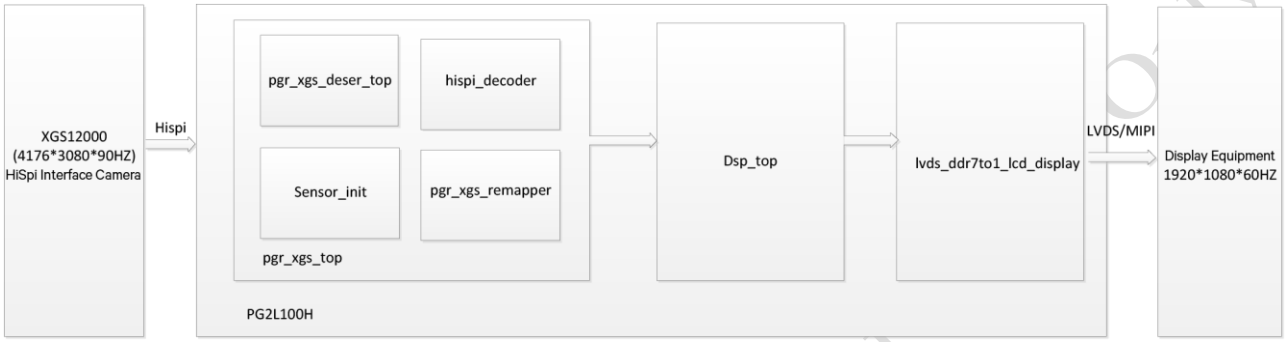

Figure 3-1 Example Design

The main function of the pgr_xgs_top module is to parse the HiSPi interface camera's serial data into parallel video data and sync signals;

The main function of the dsp_top module is to capture parallel video data from the camera, perform frame buffering, pixel rearrangement, and color space conversion;

The lvds_ddr7to1_lcd_display module mainly functions as the LVDS screen's driver module and is responsible for driving the 1920*1080 resolution LVDS screen display.

3.2 Example Interface List

Table 3-1 Example Top-level Interface List

Signal Name	I/O	Bit width	Description
Clock Reset Signals			
ref_clk	I	1	Oscillator input port, 50MHz, system reference clock
ref_clk_p	I	1	Differential oscillator input port, 200MHz, p side, the DDR reference clock
ref_clk_n	I	1	Differential oscillator input port, 200MHz, n side, the DDR reference clock
ext_rst_n	I	1	System reset signal, active-low
key_in0_n	I	1	Debug button, reserved
key_in1_n	I	1	Debug button, reserved

Signal Name	I/O	Bit width	Description
Camera high-speed data port			
rx_data_p	I	24	Receiver data port, p side
rx_data_n	I	24	Receiver data port, n side
rx_clk_p0	I	1	Receiver clock port, p side, drives even pair data, from pin GMCLK port p
rx_clk_n0	I	1	Receiver clock port, n side, driving even pair data, from pin GMCLK port n
rx_clk_p1	I	1	Receiver clock port, p side, drives odd pair data, from pin GMCLK port p
rx_clk_n1	I	1	Receiver clock port, n side, drives odd pair data, from pin GMCLK port n
Camera Configuration Ports			
sensor_rstn	O	1	Camera reset signal, active-low
spi_clk	I/O	1	SPI clock
spi_cs_n	I/O	1	SPI chip select signal, active-low
spi_do	I/O	1	SPI output data signal
spi_di	I	1	SPI input data signal
i2c_scl	O	1	I2C Clock
i2c_sda	I/O	1	I2C data
GA0	O	1	Ic address control signal, fixed to low
GA1	O	1	Ic address control signal, fixed to low
LVDS screen signal			
lvds_bl_pwm_o	O	1	LCD backlight control signal, active when pulled down
lvds_bl_en_o	O	1	LCD backlight enable control signal, active when pulled down
lvds_pwr_en_o	O	1	LCD backlight brightness control signal, which is pulled down by default and represents the maximum brightness
c0_clkout_p	O	1	LVDS output differential clock, p side, drives even number array data
c0_clkout_n	O	1	LVDS output differential clock, n side, drives even number array data
c0_dataout_p	O	4	LVDS output differential data, p side, even number array
c0_dataout_n	O	4	LVDS output differential data, n side, even number array
c1_clkout_p	O	1	LVDS output differential clock, p side, drives odd number array data
c1_clkout_n	O	1	LVDS output differential clock, n side, drives odd number array data
c1_dataout_p	O	4	LVDS output differential data, p side, odd number array
c1_dataout_n	O	4	LVDS output differential data, n side, odd number array

3.3 Example File Directory

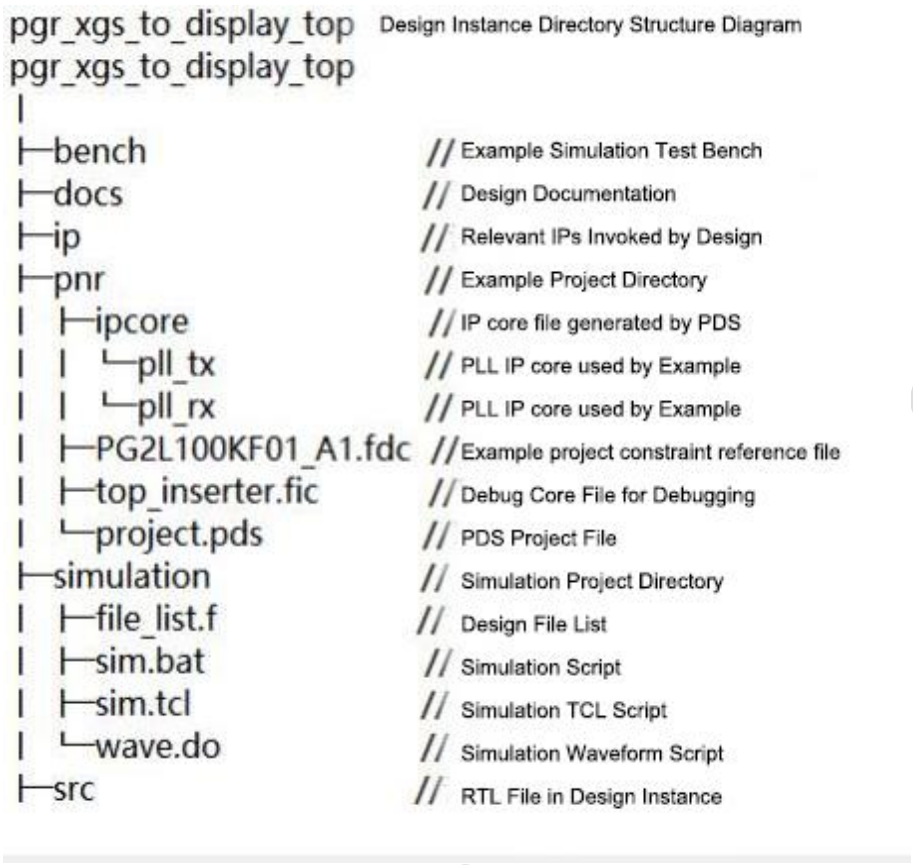

Figure 3-2 File Directory

3.4 Example Simulation

A TCL script instance is available in the reference design file directory. The dsp_sim.tcl script file has a file list of the simulation projects. Based on the PDS software installation directory, the user needs to modify lines 243~266 in the dsp_file_list.f file, to change the simulation library directory to local PDS installation directory, and modify lines 10 and 13 of the dsp_sim.tcl file, to change the version to local PDS version.

After setting up the simulation environment, run the dsp_sim.bat script to start the simulation or run the dsp_sim.tcl script in the simulation tool to start the simulation. The simulation waveform is shown in Figure 3-3.

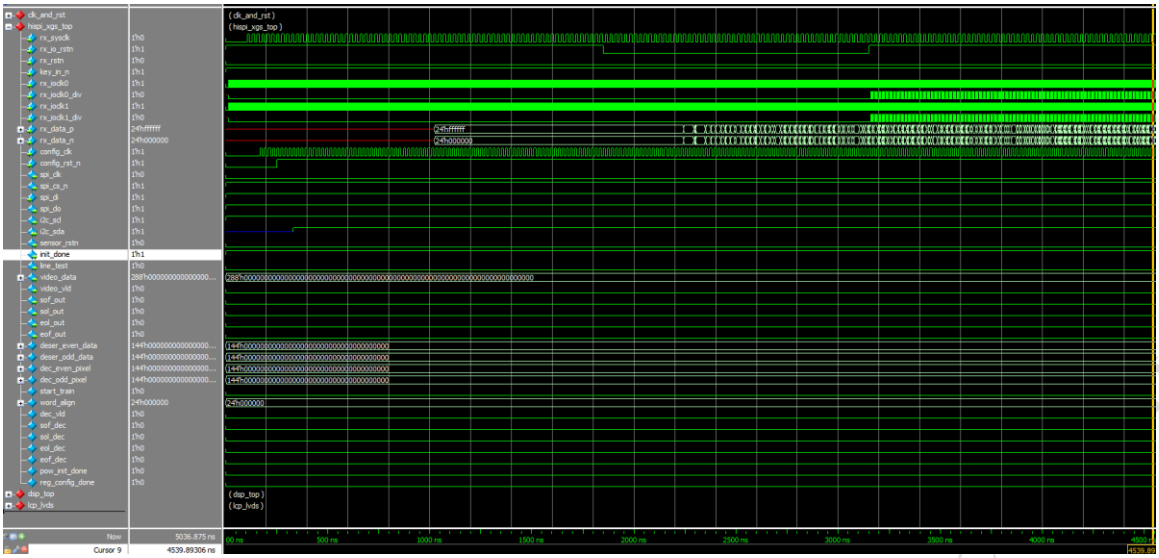

Figure 3-3 Simulation Waveform

3.5 Example Hardware Verification

The hardware verification platform is shown in Figure 3-4. In this platform, the onsemi camera XGS12000 development board is used and connected to the PG2L100H video application board (P04W100AS01_A0) via the FMC connector, which is then connected to the LVDS screen via the 30-pin FPC.

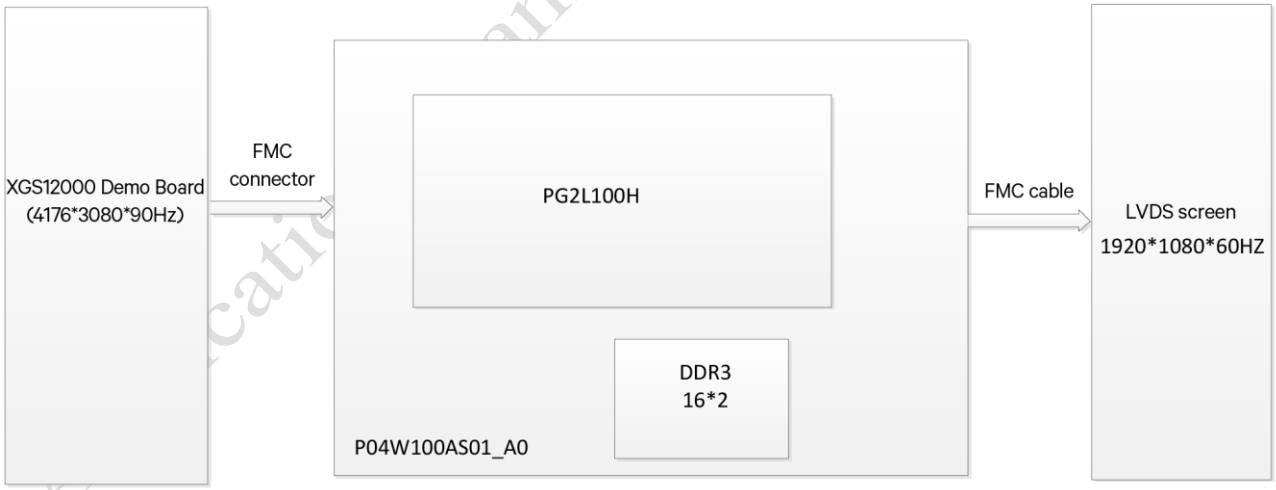

Figure 3-4 Hardware Verification Platform

The video application board is powered by a 12V DC power supply. Ensure before the hardware verification that the jumper J518 is set to 2.5V, J516 is set to 2.5V, and J514 is set to DC5V (supplying 5V power locally to the LVDS panel). Turn on the power switch, download the bitstream, and the image captured by the camera will be displayed on the LVDS screen, as shown in Figure 3-5. As the camera's image size (4176*3080) is larger than the resolution of the LVDS screen (1920*1080), the image displayed in the Example is the upper-left corner of the camera image.

Note: When connecting the 30-pin FPC cable to the LVDS screen, ensure correct orientation. When using a reverse-direction cable, insert it into the development board with the contact surface facing up.

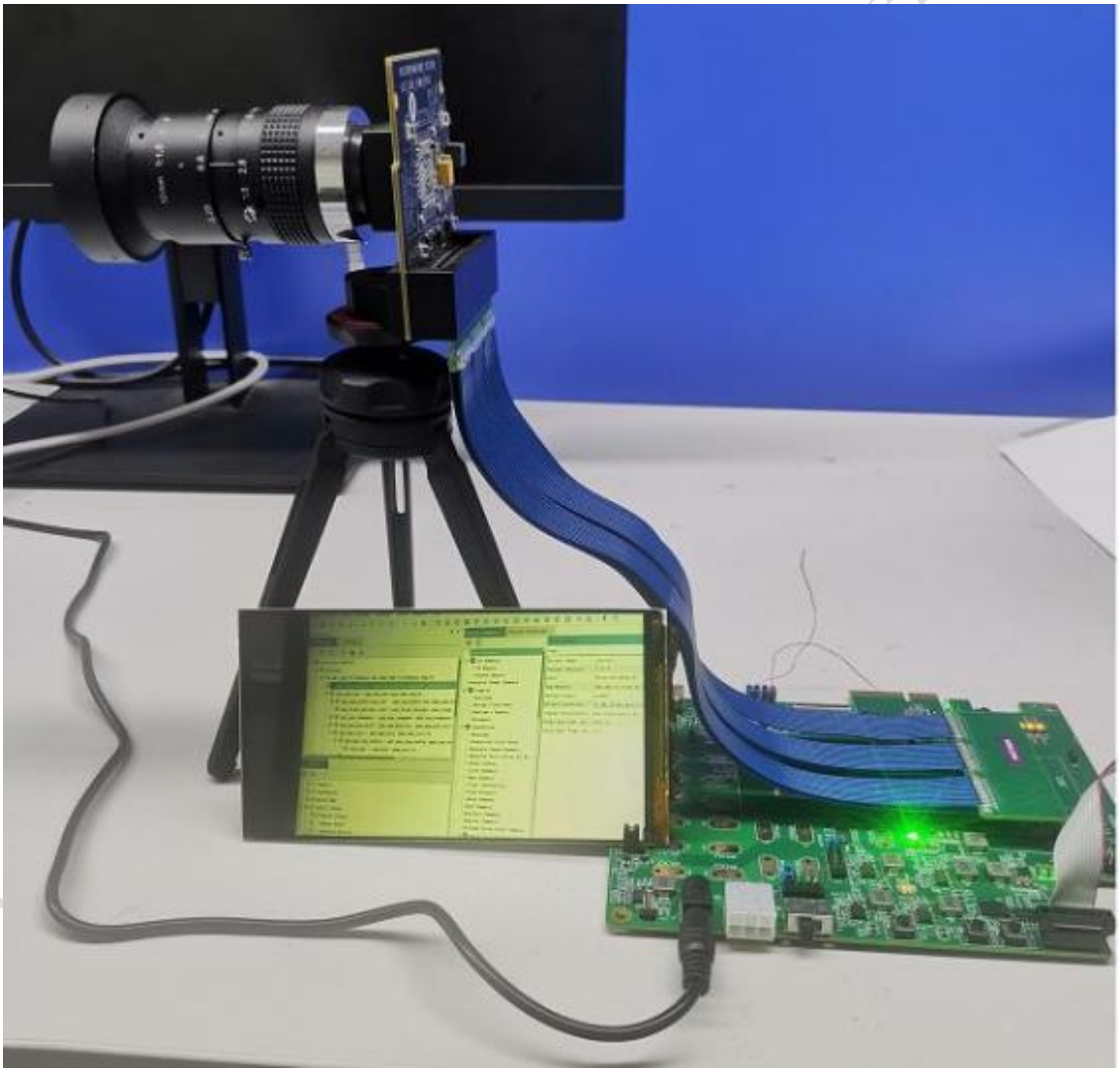

Figure 3-5 Hardware Verification

Chapter 4 Example2

4.1 Example Function Design

The block diagram for reference design 2 is shown in Figure 4-1. The images captured by the camera are processed by the FPGA, then sent to the HiSilicon 3519 for processing, and ~~finally~~ displayed on an HDMI monitor.

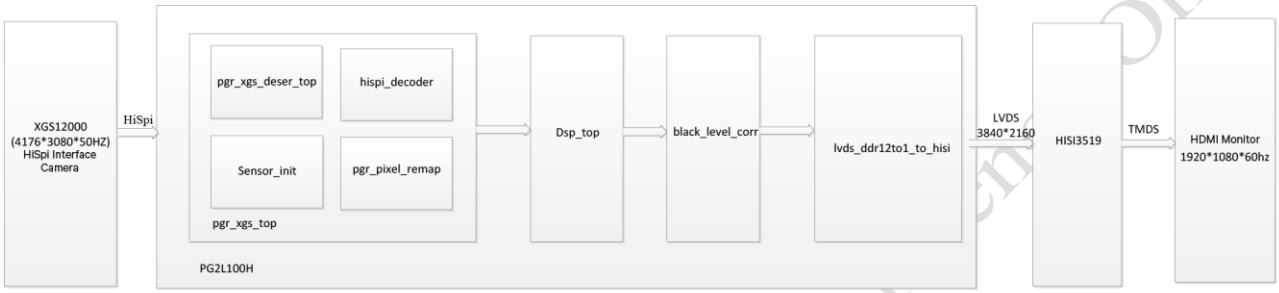

Figure 4-1 Example Design

The main function of the pgr_xgs_top module is to parse the HiSPi interface camera's serial data into parallel video data and sync signals and rearrange pixels according to their physical positions;

The main function of the dsp_top module is to save the parallel video data from the camera to frame buffer and perform frame rate conversion; the main function of the black_level_corr module is to implement black level calibration of the camera data;

The lvds_ddr12to1_to_hisi module primarily functions as an LVDS transmit module, sending 3840*2160*30Hz raw12 image data to the HiSilicon 3519 via the LVDS interface. The HiSilicon 3519 processes the image and crops it to 1920*1080*60Hz to be displayed on an HDMI monitor;

4.2 Example Interface List

Table 4-1 Example Top-Level Interface List

Signal Name	I/O	Bit width	Description
Clock Reset Signals			
ref_clk	I	1	Oscillator Input port, 50MHz, system reference clock
ref_clk_p	I	1	Differential oscillator input port, 200MHZ, p side, reserved
ref_clk_n	I	1	Differential oscillator input port, 200MHZ, n side, reserved

Signal Name	I/O	Bit width	Description
ext_rst_n	I	1	System reset signal, active-low
key_in0_n	I	1	Debug button, reserved
key_in1_n	I	1	Debug button, reserved
Camera High-speed Data Port			
rx_data_p	I	24	Receiver data port, p side
rx_data_n	I	24	Receiver data port, n side
rx_clk_p0	I	1	Receiver clock port, p side, drives even pair data, from pin GMCLK port p
rx_clk_n0	I	1	Receiver clock port, n side, drives even pair data, from pin GMCLK port n
rx_clk_p1	I	1	Receiver clock port, p side, drives odd pair data, from pin GMCLK port p
rx_clk_n1	I	1	Receiver clock port, n side, drives odd pair data, from pin GMCLK port n
Camera Configuration Ports			
sensor_rstn	O	1	Camera reset signal, active-low
spi_clk	I/O	1	Output SPI clock
spi_cs_n	I/O	1	Output SPI chip select signal, active-low
spi_do	I/O	1	Output SPI data signal
spi_di	I	1	Input SPI data signal
i2c_scl	O	1	Output I2C signal
i2c_sda	I/O	1	Input/Output I2C data signal
GA0	O	1	Ic address control signal, fixed to low,
GA1	O	1	Ic address control signal, fixed to low,
LVDS transmission interface signal			
c0_clkout_p	O	1	LVDS output differential clock, p side, group 0 LVDS
c0_clkout_n	O	1	LVDS output differential clock, n side, group 0 LVDS
c0_dataout_p	O	4	LVDS output differential data, p side, group 0 LVDS
c0_dataout_n	O	4	LVDS output differential data, n side, group 0 LVDS
c1_clkout_p	O	1	LVDS output differential clock, p side, group 1 LVDS
c1_clkout_n	O	1	LVDS output differential clock, n side, group 1 LVDS
c1_dataout_p	O	4	LVDS output differential data, p side, group 1 LVDS, using the lower two bits
c1_dataout_n	O	4	LVDS output differential data, n side, group 1 LVDS, using the lower two bits
c2_clkout_p	O	1	LVDS output differential clock, p side, group 2 LVDS, not used
c2_clkout_n	O	1	LVDS output differential clock, n side, group 2 LVDS, not used
c2_dataout_p	O	4	LVDS output differential data, p side, group 2 LVDS, not used
c2_dataout_n	O	4	LVDS outputs differential data, n side, group 2 LVDS, not used

4.3 Example File Directory

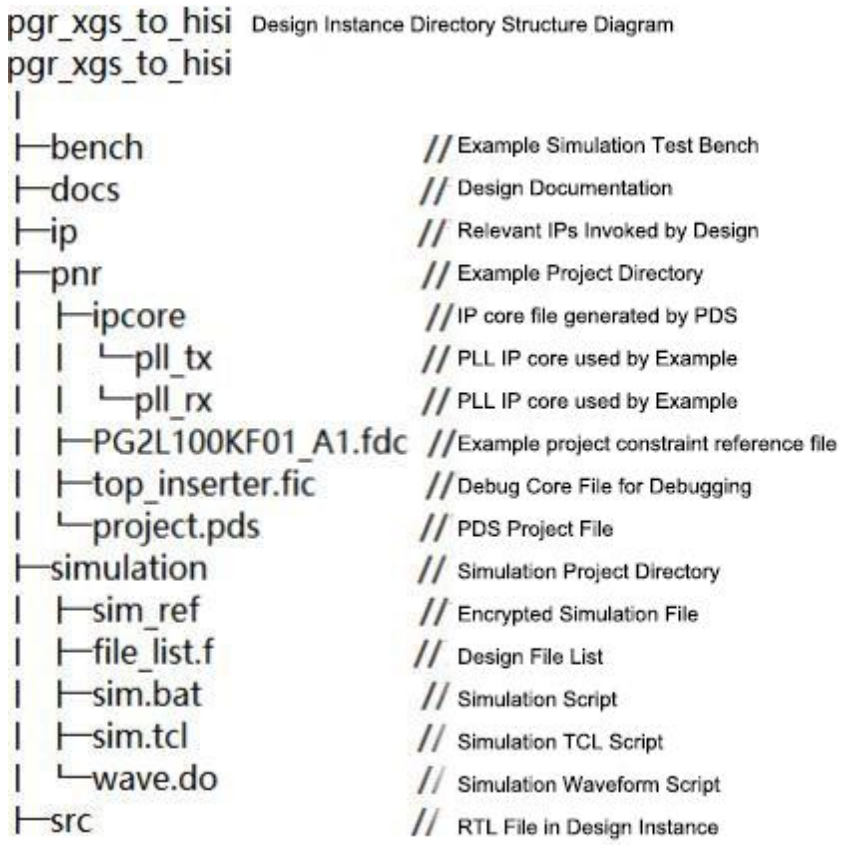

Figure 4-2 File Directory

4.4 Simulation

In the reference design file directory, a TCL script instance is provided. The dsp_sim.tcl script file has the simulation project's filelist. Based on the PDS's software installation directory, the user needs to modify the dsp_file_list.f file from line 223 to line 246, modify the simulation library directory to the local PDS installation directory, and modify lines 10 and 13 of the dsp_sim.tcl file to the locally installed PDS version.

After setting up the simulation environment, run the dsp_sim.bat script to start the simulation or run the dsp_sim.tcl script in the simulation tool to start the simulation. The simulation waveform is shown in Figure 4-3.

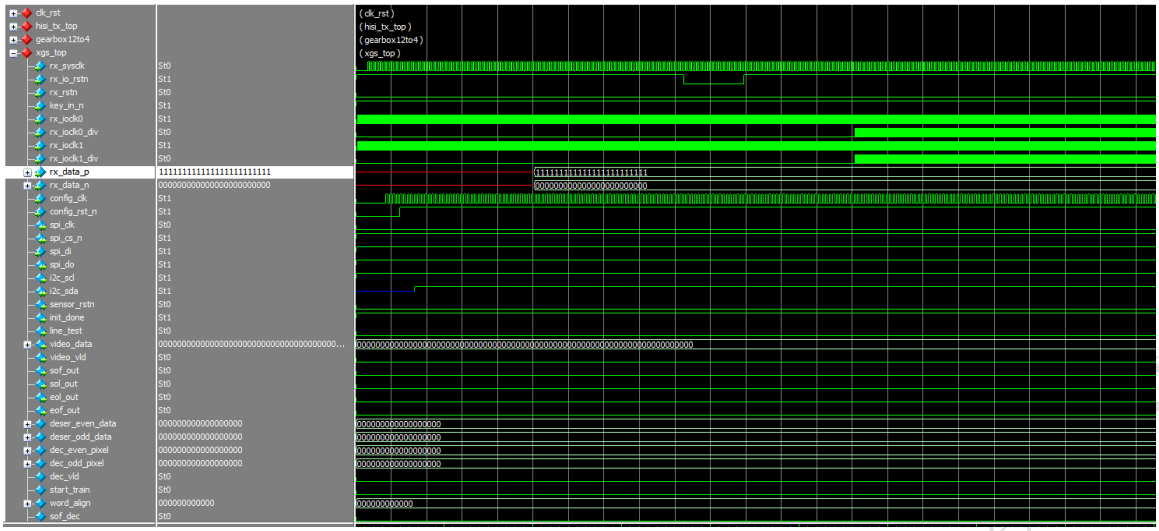

Figure 4-3 Simulation Waveform

4.5 Example Hardware Verification

The hardware verification platform is shown in Figure 4-4. In this platform, an onsemi camera XGS12000 development board is connected to the PG2L100H video application board (P04W100AS01_A0) via the FMC connector, which is then connected to the HiSilicon 3519 development board via the 90-pin direct flexible cable FPC, and finally connected to the monitor via an HDMI cable.

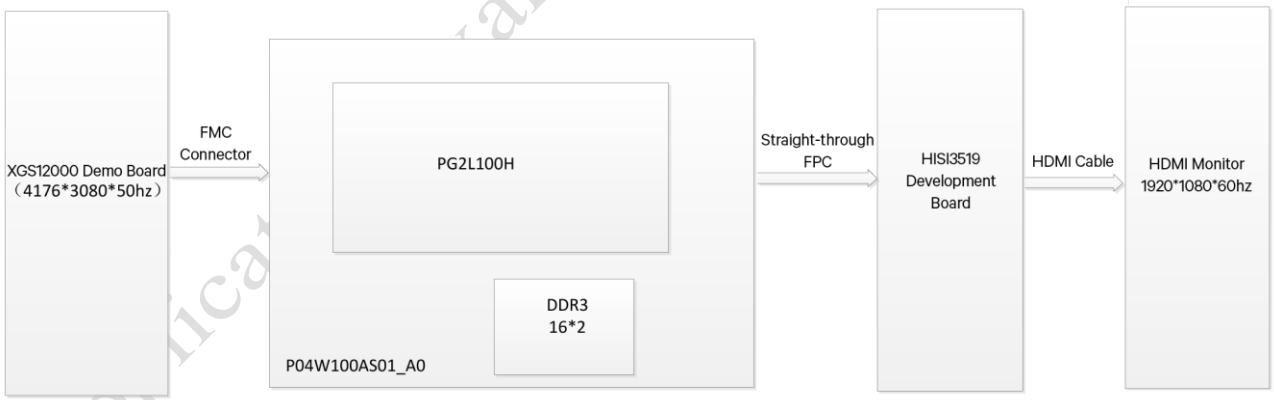

Figure 4-4 Hardware Verification Platform

The video application board is powered by a 12v DC power supply. Before hardware verification, ensure that jumpers J516 and J517 are set to 2.5v, power on the switch, and download the bitstream. The image captured by the camera is received by the FPGA and transmitted via LVDS signal to the

HiSilicon 3519. The HiSilicon 3519 processes the image and crops it to 1920*1080p60.The image is displayed on the HDMI monitor, as shown in Figure 4-5.

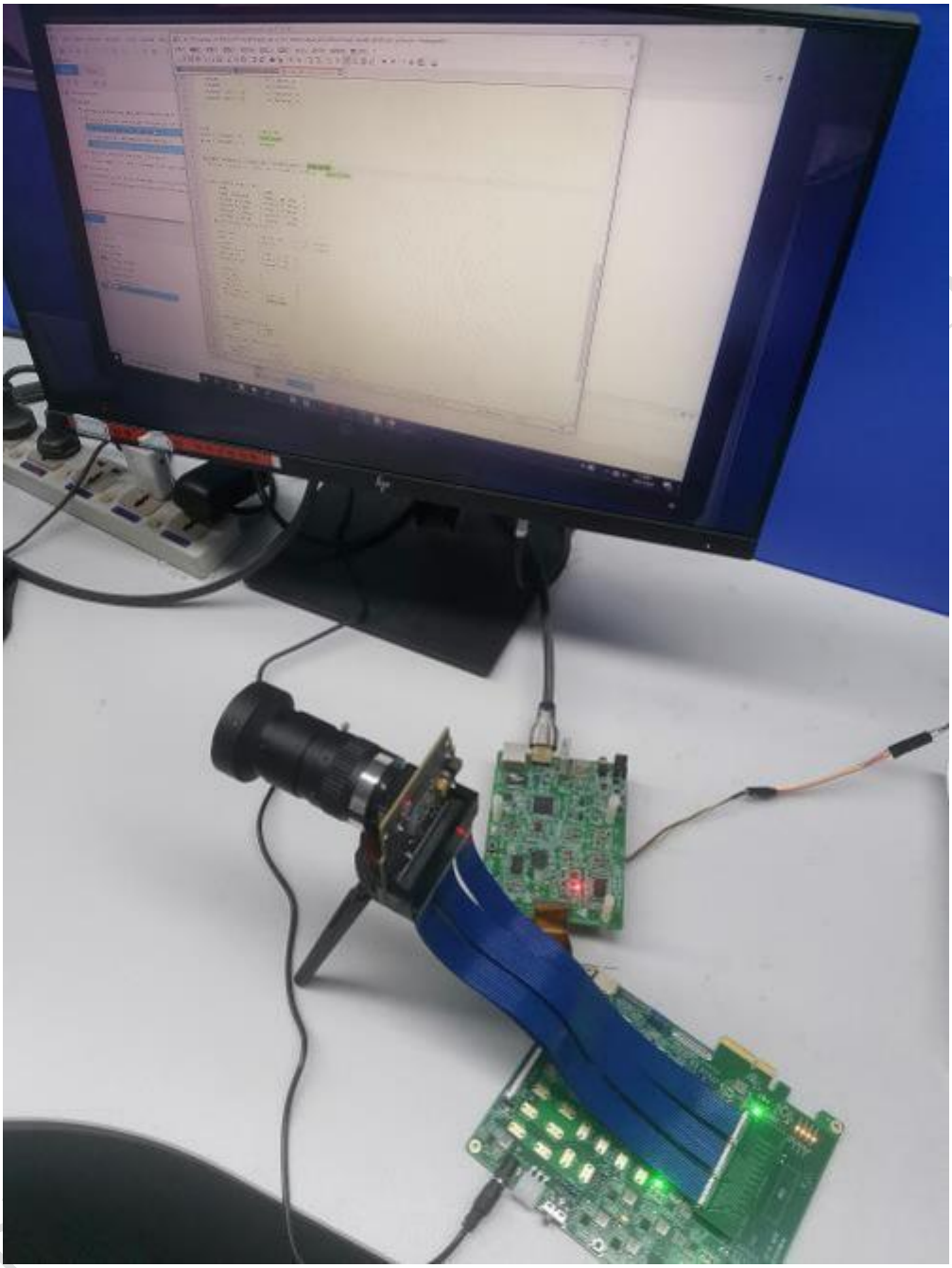

Figure 4-5 Hardware Verification

Disclaimer

Copyright Notice

Disclaimer

1. This document only provides information in stages and may be updated at any time based on the actual situation of the products without further notice. The Company assumes no legal responsibility for any direct or indirect losses caused by improper use of this document.
2. This document is provided "as is" without any warranties, including but not limited to warranties of merchantability, fitness for a particular purpose, non-infringement, or any other warranties mentioned in proposals, specifications, or samples. This document does not grant any explicit or implied intellectual property usage license, whether by estoppel or otherwise.
3. The Company reserves the right to modify any documents related to its series products at any time without prior notice.
4. The information contained in this document is intended to assist users in resolving application-related issues. While we strive for accuracy, we cannot guarantee that the document is entirely free from flaws. Should any functional abnormalities and performance degradation arise due to deviation from the prescribed procedures outlined herein, our company will neither be held liable nor concede that such issues stem from product deficiencies. The solutions presented in this document are just one of the feasible options and cannot cover all application scenarios. Consequently, if users encounter functional abnormalities or performance degradation despite adhering to the prescribed procedures outlined herein, we cannot assure that such issues are indicative of product deficiencies.