

GTP 光纤通信测试例程

黑金动力社区 2020-03-13

1 实验简介

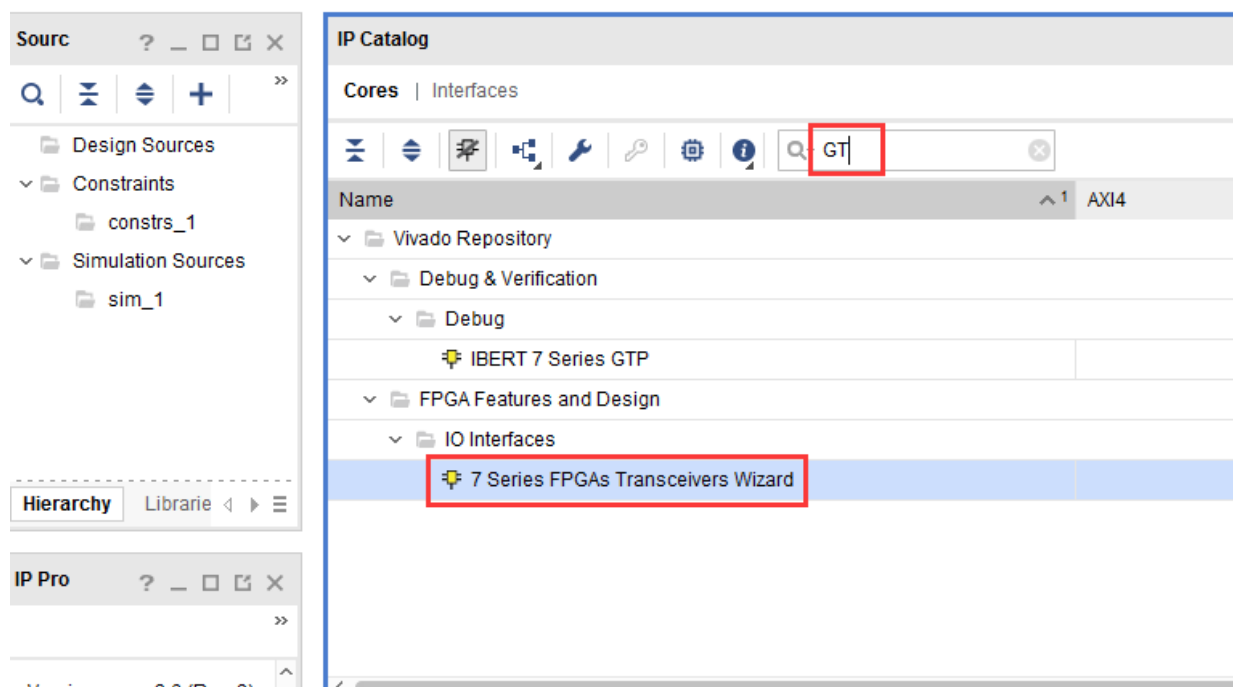
本实验将介绍通过光纤实现视频图像的传输，视频图像由黑金双目摄像头模块 AN5642 采集，再通过开发板上的其中两路光模块进行视频信号的光纤发送和接收，然后在 VGA 显示器上显示出来。

2 实验原理

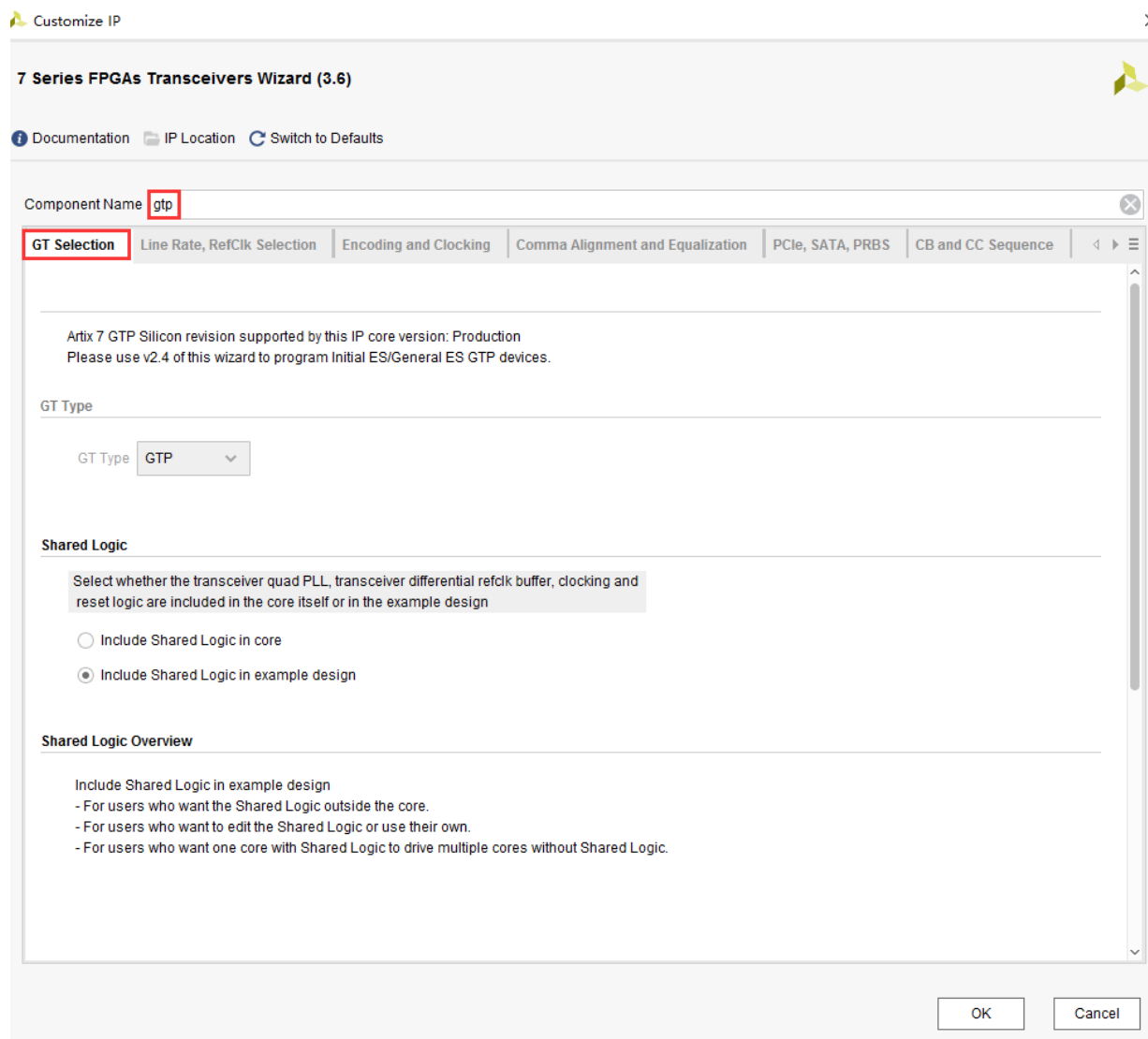
2.1 GTP IP 设计

XILINX 的 Vivado 软件已经为用户设计好了 GTP IP，用户无需关心 GTP 的内部具体工作就可以使用 IP 来实现 GTP 的高速的数据收发。下面我具体的 GTP IP 的生成和配置方法：

1. 在 IP Catalog 界面中双击 FPGA Features and Design\IO Interface 目录下的"7 Series FPGAs Transceivers Wizard"图标。



2. Component Name 栏输入"gtp"为取名，在 GT selection 界面里无需修改，保留默认。



3. 在 Line Rate, Refclk Selection 界面里，首先设置 GTP 的传输协议位“Aurora 8B10B single lane 4byte”，我们在前面一章讲过，Xilinx 的 GTP 是支持很多种协议的，Aurora 8B/10B 协议是一个可扩展的、轻量级的链路层协议，可以用于通过一条或多条串行链路将数据点到点传输。这里我们用的光模块传输是单路的，所以选择 single lane, 数据接口为 4byte, 就是 32 位数据。再选择 TX 和 RX 的 Line Rate 速度，这个 Line Rate 速度是需要是 GTP 参考时钟的整数倍，开发板上的 GTP 参考时钟为 125Mhz, 这里我们 Line Rate 为参考时钟的 25 倍，所以 Line Rate 设置为 3.125Gbps，Reference Clock 为 125Mhz。

Customize IP

7 Series FPGAs Transceivers Wizard (3.6)

Documentation IP Location Switch to Defaults

Component Name: gtp

GT Selection Line Rate, RefClk Selection Encoding and Clocking Comma Alignment and Equalization PCIe, SATA, PRBS CB and CC Sequence Summary

Protocol: aurora 8b10b single lane 4byte

TX RX

Line Rate (Gbps): 3.125 [0.5 - 6.25] TX off Line Rate (Gbps): 3.125 [0.5 - 6.25] RX off

Reference Clock (MHz): 125.000 Range: 60..660 Reference Clock (MHz): 125.000 Range: 60..660

gt row: Top Row Use Common DRP

Diagram: REFCLK1_Q0, REFCLK0_Q0, PLL0, PLL1, Quad1, GTP_X0Y7, GTP_X0Y8, GTP_X0Y5, GTP_X0Y4

PLL Selection: TX PLL0 RX PLL0 Extend reset to 3 ms

Transceiver Selection: Use GTP X0Y5 TX Clock Source: REFCLK1 Q0 RX Clock Source: REFCLK1 Q0 Advanced Clocking Option PRBS pattern generator and checker Vivado Lab Tools

Active Transceivers = 1

OK Cancel

默认软件只有使能一个 Channel (GTP_X0Y4) 的 GTP, 这里要分别选中 GTP_X0Y5, GTP_X0Y6, GTP_X0Y6, 然后选择右边的"Use GTP X0Y5/6/7"前面的钩, 这样 PLL0 都连接到了 4 个 GPT Channel 模块。

Customize IP

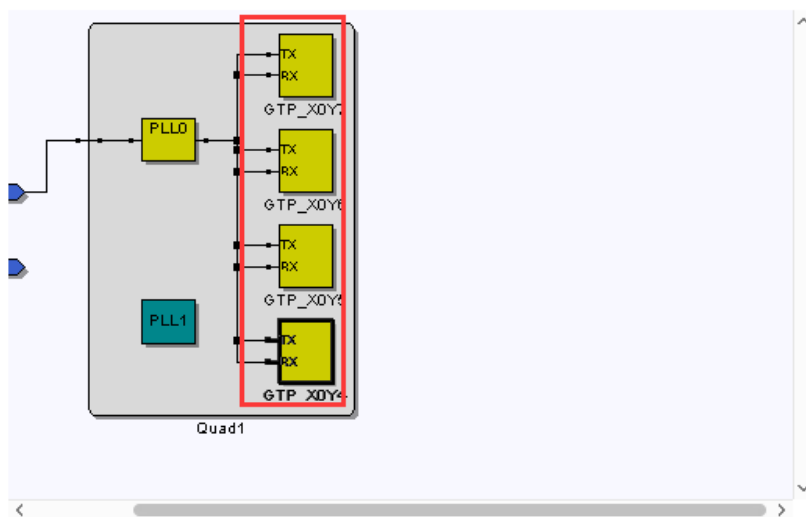
7 Series FPGAs Transceivers Wizard (3.6)

[Documentation](#) [IP Location](#) [Switch to Defaults](#)Component Name GT Selection | **Line Rate, RefClk Selection** | Encoding and Clocking | Comma Alignment and Equalization | PCIe, SATA, PRBS | CB and CProtocol

TX

Line Rate (Gbps) [0.5 - 6.25] ☐ TX offReference Clock (MHz) Range: 60..660gt row ☐ Use Common DRP

RX

Line Rate (Gbps) Reference Clock (MHz) 

PLL Selection

TX RX

Transceiver Selection

☒ Use GTP X0Y4TX Clock Source RX Clock Source ☐ Advanced Clocking Option☐ PRBS pattern generator and checker☐ Vivado Lab Tools

Active Transceivers = 4

另外 TX Clock Source 和 RX Clock Source 需要都选择 REFCLK0 Q0, 因为电路设计上 125Mhz 的参考时钟是连接到 REFCLK0 的管脚上的。

Component name: gtp

GT Selection | **Line Rate, RefClk Selection** | Encoding and Clocking | Comma Alignment and Equalization | PCIe, SATA, PRBS | CB and CC Sequence | Summary

Protocol: **aurora 8b10b single lane 4byte**

TX

Line Rate (Gbps): 3.125 [0.5 - 6.25] ☐ TX off

Reference Clock (MHz): 125.000 Range: 60..660

gt row: Top Row ☐ Use Common DRP

RX

Line Rate (Gbps): 3.125 [0.5 - 6.25] ☐ RX off

Reference Clock (MHz): 125.000 Range: 60..660

PLL Selection

TX: PLL0 RX: PLL0 ☐ Extend reset to 3 ms

Transceiver Selection

☒ Use GTP X0Y4

TX Clock Source: REFCLK0 Q0

RX Clock Source: REFCLK0 Q0

☐ Advanced Clocking Option

☐ PRBS pattern generator and checker

☐ Vivado Lab Tools

Diagram: A block diagram showing a PLL0 block connected to four GTP_X0Y blocks (GTP_X0Y7, GTP_X0Y6, GTP_X0Y5, GTP_X0Y4). Each GTP block has TX and RX ports. A PLL1 block is also shown, connected to the GTP_X0Y4 block. The diagram is labeled "Quad1".

4. 在 Encoding and Clocking 界面里，设置 TX 和 RX 的外部数据宽度，8B/10B 使能，内部数据宽度等信息，这里保留默认配置。因为外部数据宽度是 32，是内部数据宽度的 2 倍，所以这里 GTP 的内部时钟频率是外部接口的 2 倍。

Documentation IP Location Switch to Defaults

Component Name: gtp

GT Selection Line Rate, RefClk Selection **Encoding and Clocking** Comma Alignment and Equalization PCIe, SATA, PRBS CB and CC Sequence Summary

☒ Use DRP DRP/System Clock Frequency (MHz) 100 [0.0 - 175.0]

Optional Ports

☐ TXBYPASS8B10B ☒ TXCHARDISPMODE ☒ TXCHARDISPVAL
☒ RXCHARISCOMMA ☒ RXCHARISK ☐ RXSTARTOFSEQ

Synchronization and Clocking

TX **RX**

☒ Enable TX Buffer ☒ Enable RX Buffer

TXUSRCLK Source TXOUTCLK RX Buffer Bypass Mode Auto

TXOUTCLK Source ☐ Use TXPLLREFCLK RXUSRCLK Source RXOUTCLK


RXOUTCLK Source ☐ Use RXPLLREFCLK

Optional Ports

☒ TXPCSRESET ☒ TXPMARESET ☐ TXSYSCLKSEL ☐ TXRATE ☒ TXBUFSTATUS ☐ TX8B10BEN
☒ RXPCSRESET ☒ RXPMARESET ☐ RXSYSCLKSEL ☐ RXRATE ☒ RXBUFSTATUS
☒ RXBUFRESET ☒ RXCDRHOLD ☐ SIGVALIDCLK ☐ PLL0 PD ☐ PLL1 PD
☐ TXPIPPMEN ☐ TXPIPPMOVRDEN ☐ TXPIPPMSEL ☐ TXPIPPMPD ☐ TXPIPPMSTEPSIZE

OK Cancel

4. 在 Comma Alignment and Equalization 界面里，设置 Comma 值为 K28.5, K28.5 是一种用以表示 Fibre Channel 操作开始的特殊 10 比特字符。8B/10B 编码中将 K28.5 作为 K 码的控制字符，称为“comma”，所以可以用 comma 字符指示帧的开始和结束标志，或始终修正和数据流对齐的控制字符。

 Customize IP**7 Series FPGAs Transceivers Wizard (3.6)**[Documentation](#) [IP Location](#) [Switch to Defaults](#)Component Name

| GT Selection | Line Rate, RefClk Selection | Encoding and Clocking | Comma Alignment and Equalization | PCIe, SATA, PRBS | CB and |
|--|---------------------------------------|-----------------------|---|--|--------|
| <input checked="" type="checkbox"/> Use comma detection | | | Comma Value <input type="text" value="K28.5"/> | Comma Mask <input type="text" value="1111111111"/> | |
| <input type="checkbox"/> Decode valid comma only | | | Plus Comma <input type="text" value="0101111100"/> | Align to <input type="text" value="Two Byte Bou"/> | |
| <input type="checkbox"/> Combine plus/minus commas (double-length comma) | | | Minus Comma <input type="text" value="1010000011"/> | | |
| Optional Ports | | | | | |
| <input checked="" type="checkbox"/> ENPCOMMAALIGN (Enables positive Comma Alignment) <input checked="" type="checkbox"/> ENMCOMMAALIGN (Enables negative Comma Alignment) | | | | | |
| <input type="checkbox"/> RXSLIDE <input checked="" type="checkbox"/> RXBYTEISALIGN <input checked="" type="checkbox"/> RXBYTEREALIGN <input checked="" type="checkbox"/> RXCOMMADET | | | | | |
| Termination and Equalization | | | | | |
| Differential Swing and Emphasis Mode <input type="text" value="Custom"/> | | | | | |
| RX Equalization | | | RX Termination | | |
| Mode | <input type="text" value="LPM-Auto"/> | Voltage | <input type="text" value="Programmable"/> | | |
| Automatic Gain Control | <input type="text" value="Auto"/> | Trim Value (mV) | <input type="text" value="800"/> | | |
| Optional Ports | | | | | |
| <input checked="" type="checkbox"/> TXPOLARITY <input type="checkbox"/> TXINHIBIT <input checked="" type="checkbox"/> TXDIFFCTRL <input checked="" type="checkbox"/> TXPOSTCURSOR <input checked="" type="checkbox"/> TXPRECURSOR <input checked="" type="checkbox"/> TXMAINCURSOR | | | | | |
| <input checked="" type="checkbox"/> RXPOLARITY | | | | | |

5. PCIe,SATA, PRBS 页面修改如下。

Component Name: gtp

GT Selection | Line Rate, RefClk Selection | Encoding and Clocking | Comma Alignment and Equalization | **PCIe, SATA, PRBS** | CB and CC Sequence | Summary

☐ Enable PCI Express

SATA COM sequence

Bursts: 4 [0 - 7] Idles: 4 [0 - 7]

PCI Express Parameters

Transition Time

To P2: 100 [0 - 255] From P2: 60 [0 - 4095] To/From Non P2: 60 [0 - 255]

Optional Ports

| | | | |
|--|---|---|--------------------------------------|
| <input checked="" type="checkbox"/> LOOPBACK | <input type="checkbox"/> RXCOMWAKEDET | <input type="checkbox"/> TXDETECTRX | <input type="checkbox"/> RXSTATUS |
| <input type="checkbox"/> TXCOMINIT | <input type="checkbox"/> TXELECIDLE | <input type="checkbox"/> RXVALID | <input type="checkbox"/> TXCOMSAS |
| <input type="checkbox"/> PHYSTATUS | <input type="checkbox"/> RXCOMINITDET | <input type="checkbox"/> TXCOMWAKE | <input type="checkbox"/> RXCOMSASDET |
| <input type="checkbox"/> TXCOMFINISH | <input checked="" type="checkbox"/> TXPOWERDOWN | <input checked="" type="checkbox"/> RXPOWERDOWN | |

OOB signalling and PRBS

☐ Use RX OOB Signal Detection

PRBS

| | | | |
|---|--|---|--|
| <input checked="" type="checkbox"/> Use PRBS Detector | <input checked="" type="checkbox"/> Use Port TXPRBSSEL | <input checked="" type="checkbox"/> Use Port TXPRBSFORCEERR | <input type="checkbox"/> RXPRBS_LOOPBACK |
|---|--|---|--|

OK Cancel

6. 在 CB and Sequence 页面中不用修改。

Component Name

GT Selection

Line Rate, RefClk Selection

Encoding and Clocking

Comma Alignment and Equalization

PCIe, SATA, PRBS

CB and CC Sequence

Summary

Channel Bonding

☐ Use Channel Bonding

☐ Use Two Channel Bonding Sequences

Sequence Max Skew

Sequence length

Clock correction

☒ Use Clock Correction

PPM Offset +/- [-1250 - 1250]

☐ Use Two Clock Correction Sequences

Periodicity of the CC sequence (bytes)

Sequence length

Sequence Definition

| | Sequence | K Character | Inverted Disparity | Don't Care |
|------------------|---------------------------------------|-------------------------------------|--------------------------|--------------------------|
| Sequence1, Byte1 | <input type="text" value="11110111"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Sequence1, Byte2 | <input type="text" value="11110111"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Sequence1, Byte3 | <input type="text" value="11110111"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Sequence1, Byte4 | <input type="text" value="11110111"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Sequence2, Byte1 | <input type="text" value="00000000"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Sequence2, Byte2 | <input type="text" value="00000000"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Sequence2, Byte3 | <input type="text" value="00000000"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Sequence2, Byte4 | <input type="text" value="00000000"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

7. 在 Summary 界面中检查一下配置情况，点击 OK 完成。

Documentation IP Location Switch to Defaults

Component Name gtp

GT Selection Line Rate, RefClk Selection Encoding and Clocking Comma Alignment and Equalization PCIe, SATA, PRBS CB and CC Sequence **Summary**

Summary

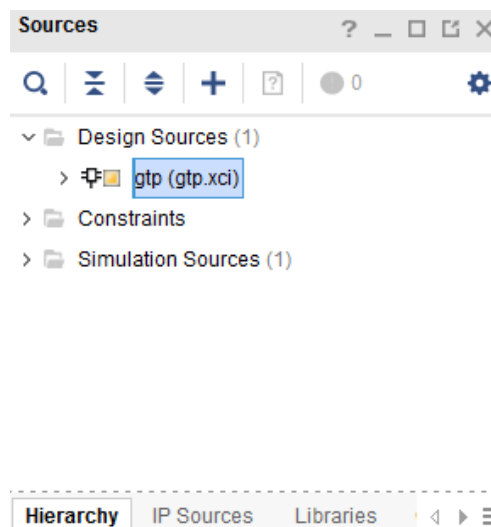
| Features | GT |
|-------------------------|--------------------------------|
| Protocol File | aurora_8b10b_single_lane_4byte |
| TX Line Rate(Gbps) | 3.125 |
| TX reference clock(MHz) | 125.000 |
| Encoding | 8B/10B |
| TX Internal Data width | 20 |
| TX External Data width | 32 |
| TXUSRCLK(MHz) | 156.25 |
| TXUSRCLK2(MHz) | 78.125 |
| TX Buffer Enabled | true |
| RX Line Rate(Gbps) | 3.125 |
| RX reference clock(MHz) | 125.000 |
| Decoding | 8B/10B |
| RX Internal Data width | 20 |
| RX External Data Width | 32 |
| RXUSRCLK(MHz) | 156.25 |
| RXUSRCLK2(MHz) | 78.125 |
| RX Buffer Enabled | true |

OK

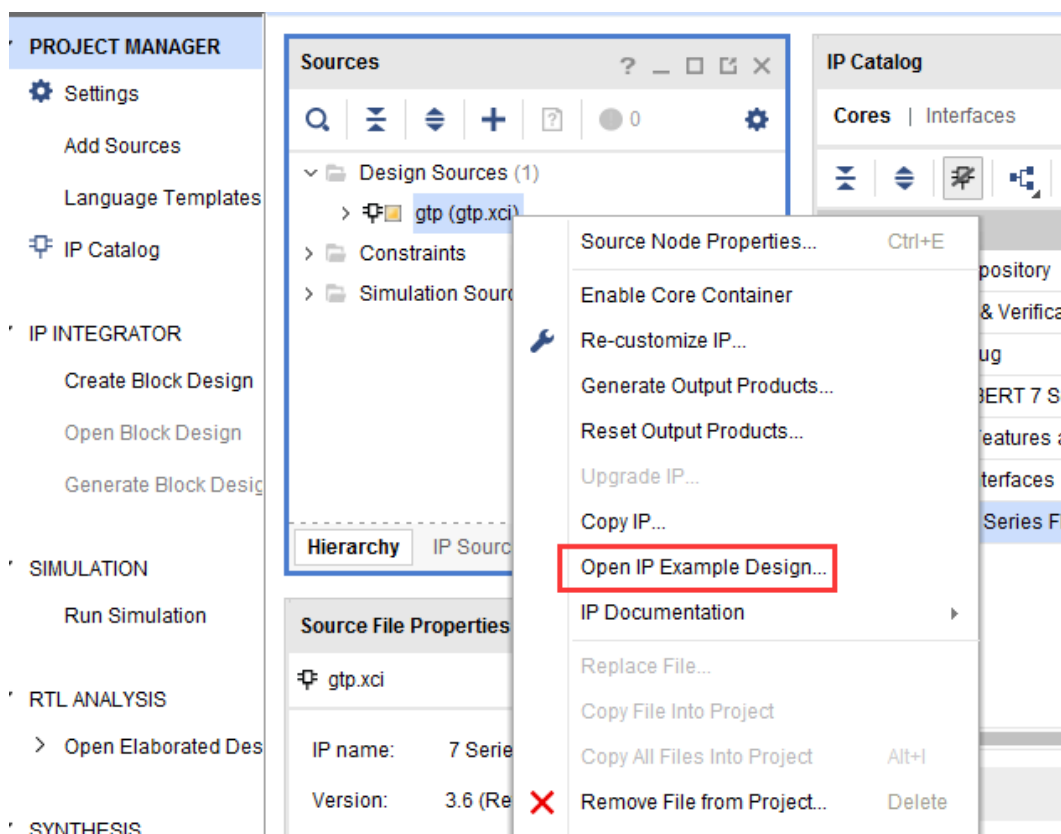
Cancel

关于更多 GTP IP 的配置信息，大家请参考 Xilinx 提供的文档“pg168-gtwizard.pdf”和“ug482_7Series_GTP_Transceivers.pdf”，这两个文档上有对 GTP IP 各个配置参数做了详细的介绍。

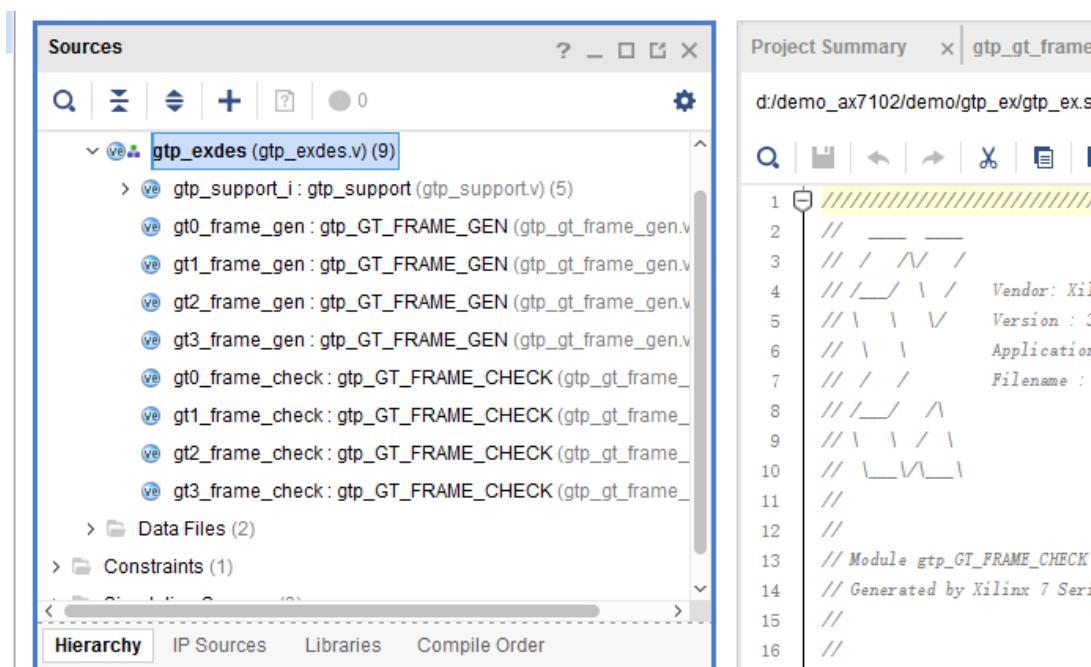
8. 配置完成在工程中自动添加刚刚生成 gtp 的 IP。



9. 我们来生成 GTP IP 的 example 工程，右键选择 gtp,在下拉菜单里选择"Open IP Example Design..."。



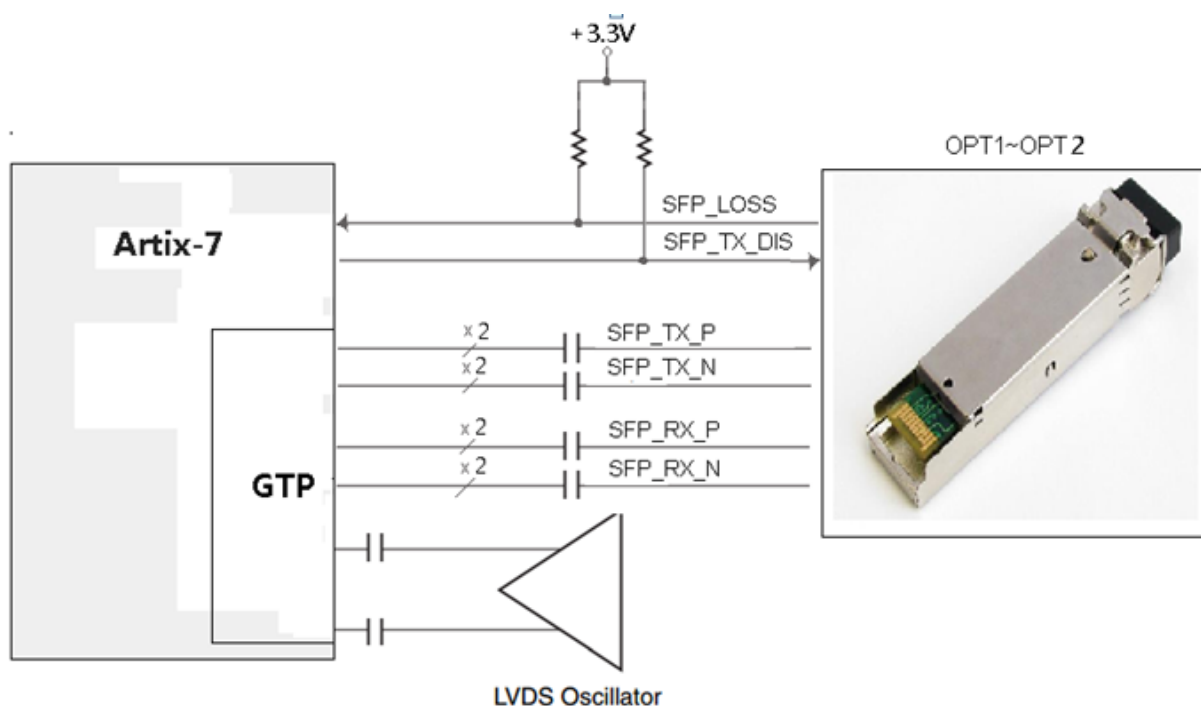
10. 生成的 example 工程如下图所示，在这个例子工程中，程序会在 gt0_frame_gen 模块中产生测试数据进行 GTP 的数据传输，在 gt0_frame_check 模块接收并检查是否正确，如果不正确，错误统计值增加。



关于 example 的工程代码和测试我们这里不做介绍，大家自己去看去测试就可以了。在后面的视频图像光纤传输的软件开发中我们会用到这里 example 工程中的一些文件和 IP。

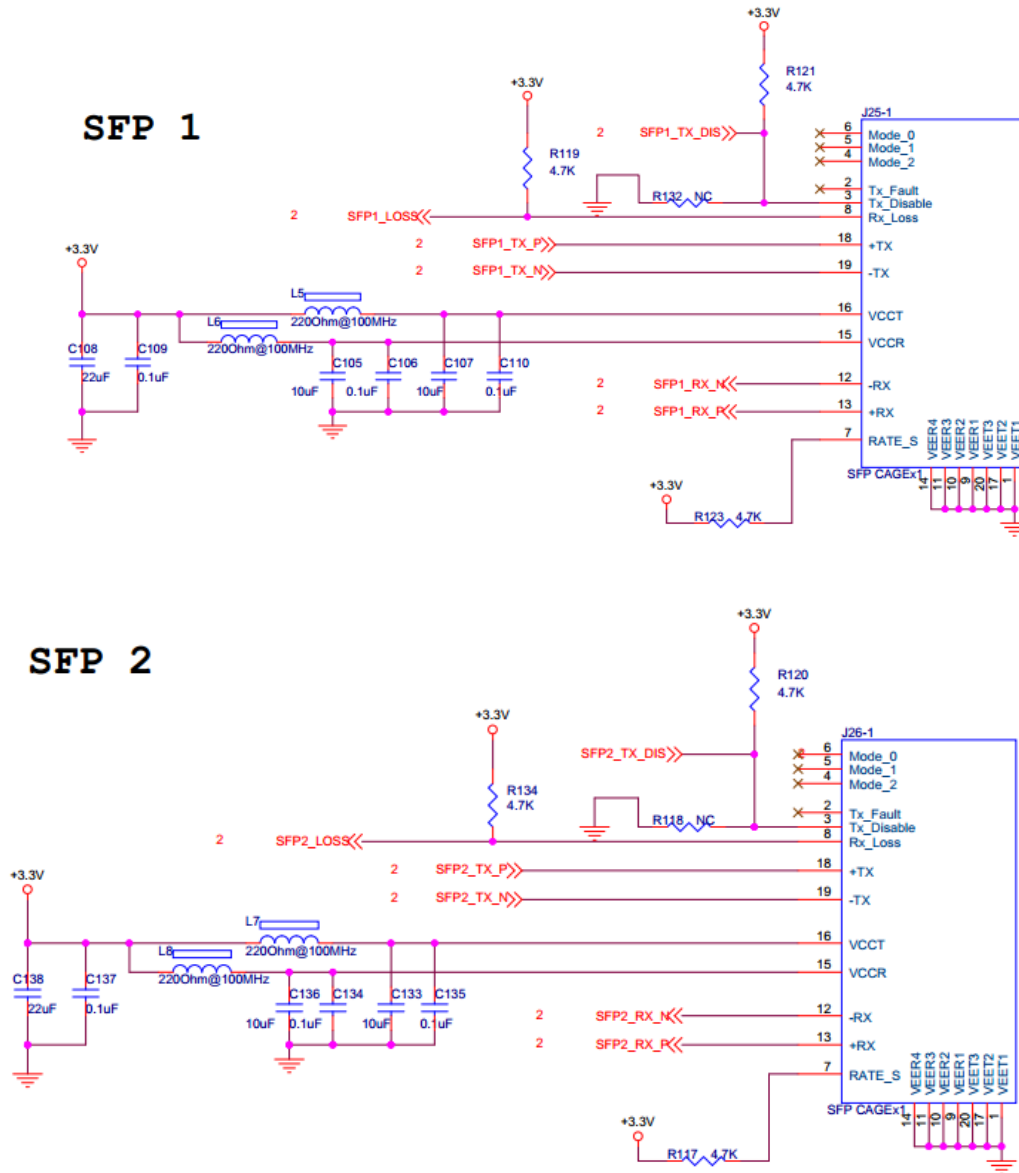
2.2 硬件介绍

在 AX7102 开发板上，有 2 路光纤接口 OPT1~OPT2, 分别连接到 FPGA 芯片的 GTP 的通道上。FPGA 和光纤连接的设计示意图如下图所示:



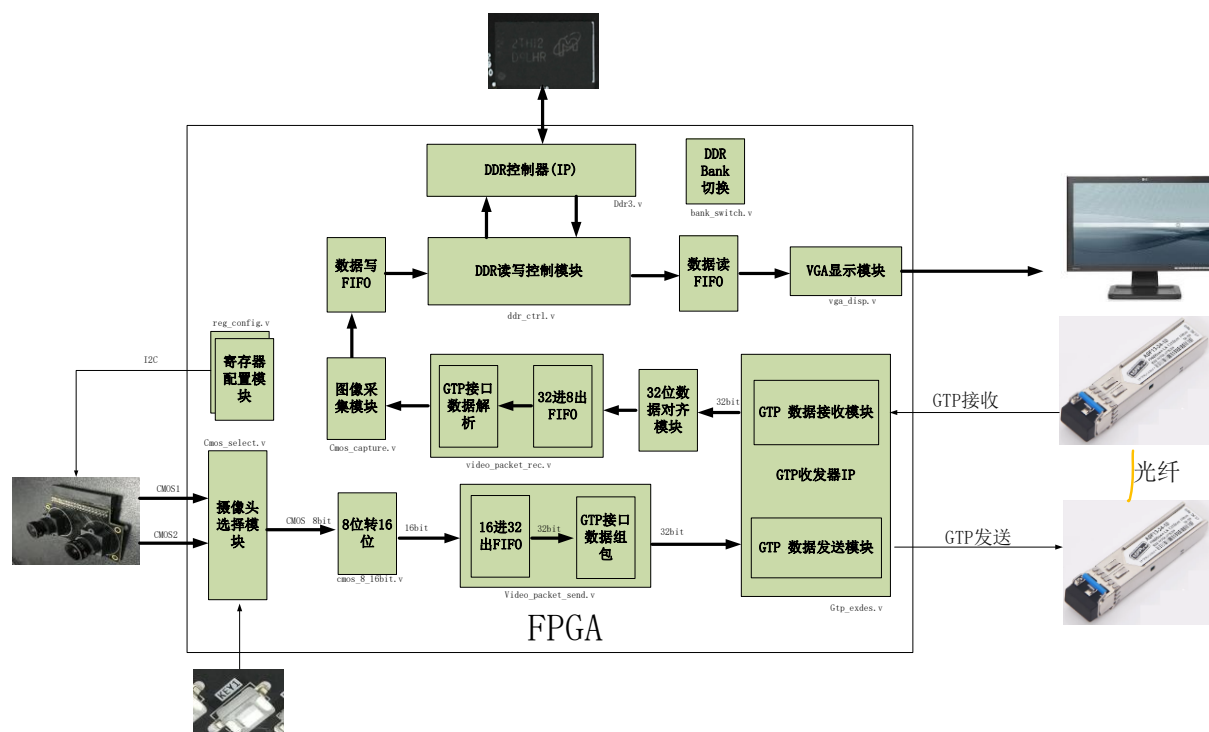
其中 OPT1 光模块接口连接到 GTP 的 Channel0 上，OPT2 跟 GTP 的 Channel1 相连。光模块和 FPGA 之间用 0.1uf 的电容隔开，使用 AC Couple 的模式。

光模块的 LOSS 信号和 TX_Disable 信号连接到 FPGA 的普通 IO 上。LOSS 信号用来检测光模块的光接收是否丢失，如果没有插入光纤或者 Link 上，LOSS 信号为高，否则为低。TX_Disable 信号用来使能或者不使能光模块的光发射，如果 TX_Disable 信号为高，光发射关闭，否则光发送使能，正常使用的时候需要拉低此信号。硬件原理图如下：



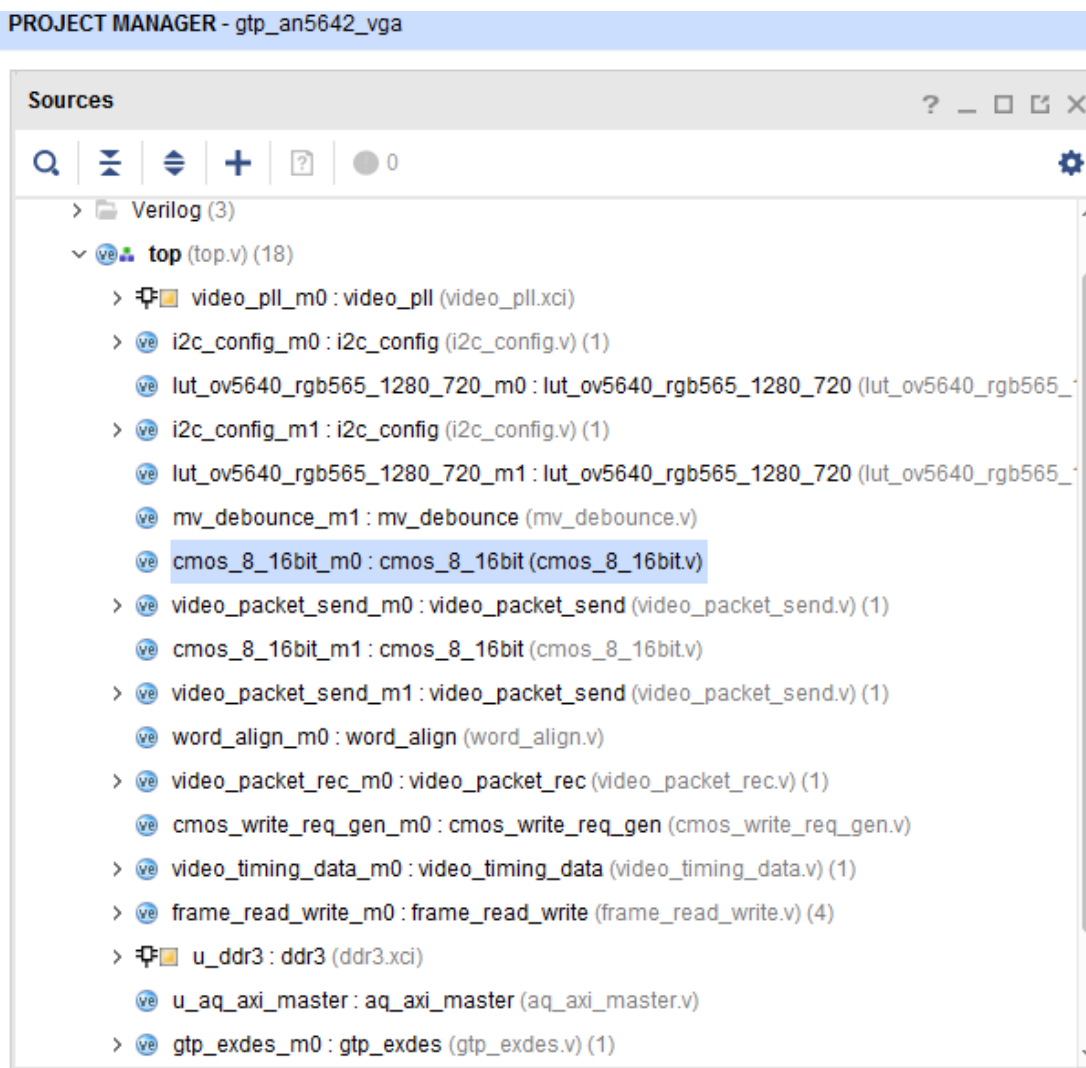
3 程序设计

视频图像光纤传输的 FPGA 程序设计的逻辑框图如下图所示：



视频图像从双目摄像头 AN5642 传入 FPGA 后，先通过摄像头切换模块来选择其中的一路，切换使用开发板上的按键来实现，视频图像先转换成 16 位的数据宽度，再存入到一个 16 位进，32 位出的 FIFO 中。当 FIFO 中的数据达到一定量的时候（1 行视频数据），从 FIFO 中取出数据使用 GTP IP 发送给外部的光模块 1，光模块 1 把电信号转换成光信号通过光纤传输到光模块 2，光模块 2 又把光信号转换成电信号输入到 FPGA 的 GTP 接收，GTP 接收到的数据需要做一个 32 位数据对齐之后，并解析出视频图像部分的数据，再把数据存入到 FIFO 中。后面就是把图像传到 DDR 的缓存中，在 VGA 显示器上显示。

视频图像光纤传输设计好的工程如下图所示：



因为大部分程序跟前面的双目 VGA 切换显示实验一样，我们这里只对增加部分的模块做一下功能说明：

1) gtp 数据通信模块

在前面我们已经生成过 gtp IP 的 example 工程，这里只有添加 gtp IP 的 example 工程中程序和 IP 就可以了（不添加 gt0_frame_gen.v 模块和 gt0_frame_check.v 模块）。因为 gt0_frame_gen.v 模块和 gt0_frame_check.v 模块这两个是测试数据的产生和检查模块，本例程用不上。另外需要对添加的 gtp_exdes.v 文件进行修改，主要是删除 gt0_frame_gen.v 模块和 gt0_frame_check.v 模块的例化，再在模块的 Port 处添加以下的 4 个 channel 的用户接口信号，添加后的四个通道的信号如下图所示：

The screenshot displays the ALINX IDE interface. On the left, the 'Sources' window shows a project hierarchy with various modules. The 'gtp_exdes_m0 : gtp_exdes (gtp_exdes.v)' module is selected. Below it, the 'Source File Properties' window shows the 'General' tab for 'gtp_exdes.v'. On the right, the 'gtp_exdes.v' Verilog module is open, showing its port definitions. A red box highlights the port list, and a yellow box highlights the 'tx2_kchar' input.

Sources Window:

- video_pll_m0 : video_pll (video_pll.xci)
- i2c_config_m0 : i2c_config (i2c_config.v)
- lut_ov5640_rgb565_1280_720_m0 : lut_...
- i2c_config_m1 : i2c_config (i2c_config.v)
- lut_ov5640_rgb565_1280_720_m1 : lut_...
- mv_debounce_m1 : mv_debounce (mv_c...
- cmos_8_16bit_m0 : cmos_8_16bit (cmo...
- video_packet_send_m0 : video_packet_s...
- cmos_8_16bit_m1 : cmos_8_16bit (cmo...
- video_packet_send_m1 : video_packet_s...
- word_align_m0 : word_align (word_align...
- video_packet_rec_m0 : video_packet_rec...
- fifo_2048_32i_8o_m0 : fifo_2048_3...
- cmos_write_req_gen_m0 : cmos_write_r...
- video_timing_data_m0 : video_timing_da...
- frame_read_write_m0 : frame_read_writ...
- u_ddr3 : ddr3 (ddr3.xci)
- u_aq_axi_master : aq_axi_master (aq_ax...
- gtp_exdes_m0 : gtp_exdes (gtp_exdes.v)**
- gtp_support_i : gtp_support (gtp_sup...

Source File Properties:

Source File Properties
gtp_exdes.v

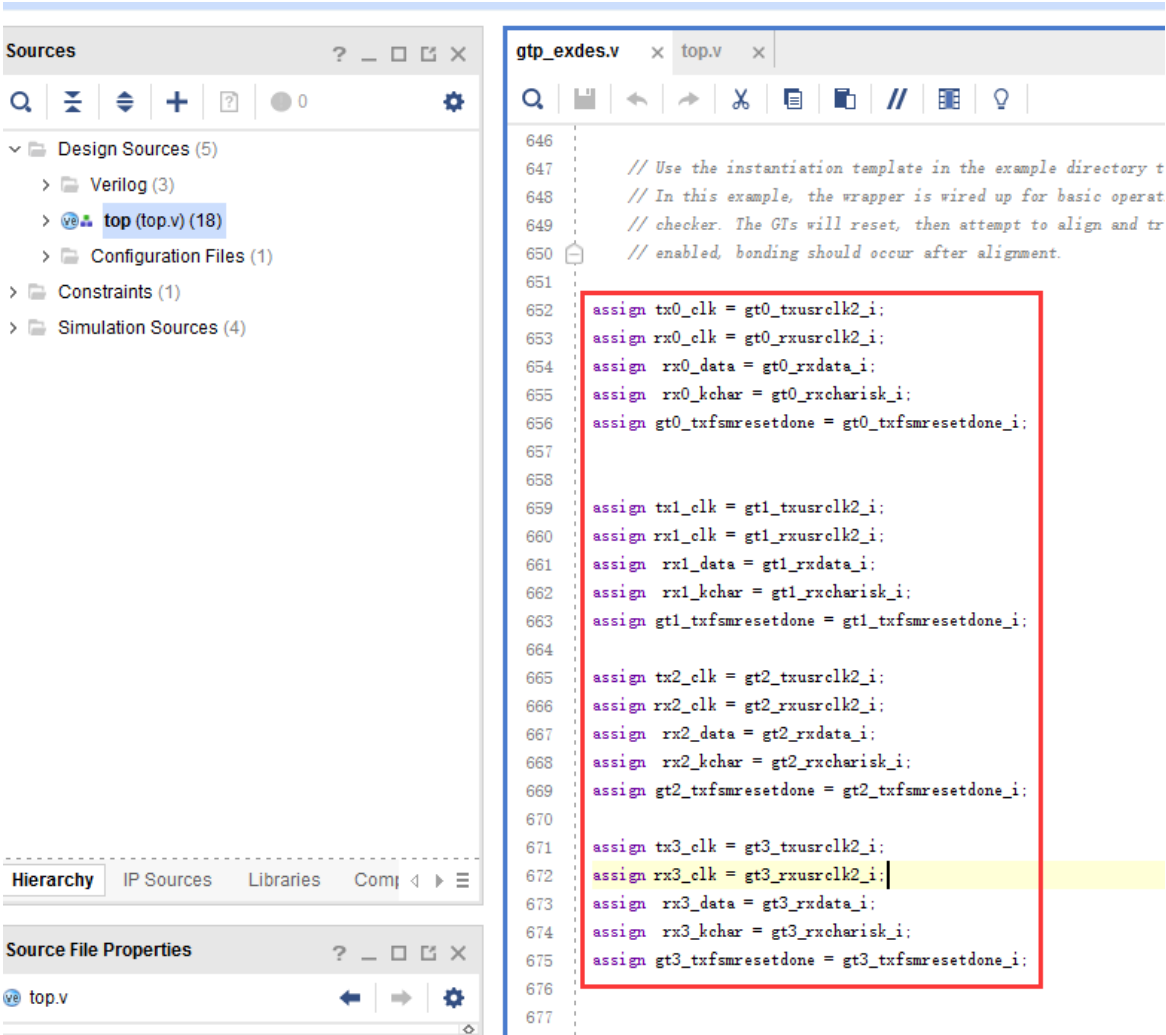
gtp_exdes.v:

```

3 module gtp_exdes
4 (
5     output tx0_clk,
6     output gt0_txfsresetdone,
7     input[31:0] tx0_data,
8     input[3:0] tx0_kchar,
9     output rx0_clk,
10    output[31:0] rx0_data,
11    output[3:0] rx0_kchar,
12
13    output tx1_clk,
14    output gt1_txfsresetdone,
15    input[31:0] tx1_data,
16    input[3:0] tx1_kchar,
17    output rx1_clk,
18    output[31:0] rx1_data,
19    output[3:0] rx1_kchar,
20
21    output tx2_clk,
22    output gt2_txfsresetdone,
23    input[31:0] tx2_data,
24    input[3:0] tx2_kchar,
25    output rx2_clk,
26    output[31:0] rx2_data,
27    output[3:0] rx2_kchar,
28
29    output tx3_clk,
30    output gt3_txfsresetdone,
31    input[31:0] tx3_data,
32    input[3:0] tx3_kchar,
33    output rx3_clk,
34    output[31:0] rx3_data,
35    output[3:0] rx3_kchar,
36

```

再把端口定义四个 Channel 的信号和 gtp_support 子模块里的信号进行连接。



下面对在 gtp_exdes.v 端口中添加的几个 GTP 用户接口信号做一下介绍，以下以 channel0 的 GTP 接口为例：

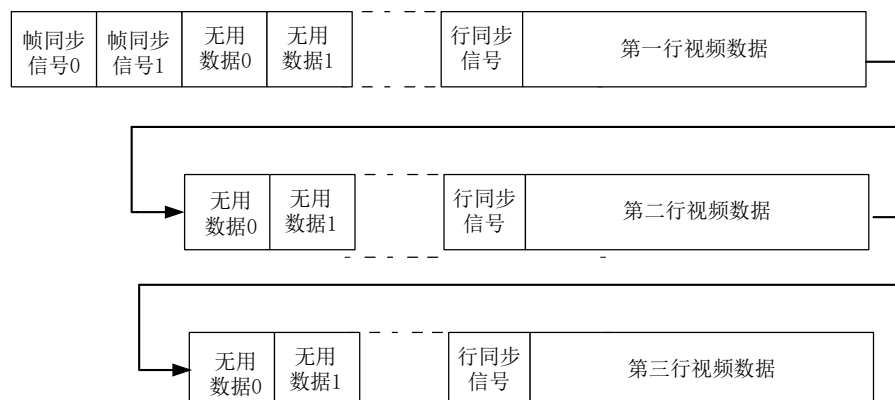
| 信号名 | 位数 | 输入输出 | 说明 |
|-----------|----|------|--|
| tx0_clk | 1 | 输出 | 数据发送时钟，也就是第十四部分介绍的 TXUSRCLK2，频率为 GTP 的参考时钟 125Mhz。数据在上升沿有效。 |
| tx0_data | 32 | 输入 | GTP 发送数据。 |
| tx0_kchar | 4 | 输入 | GTP 发送的 K 控制字，用来指示发送的数据是 K 码控制字符还是正常传输数据。高电平表明是 K 码控制字符，4 位对应发送数据的 4 个 Byte。 tx0_kchar [3] 对应 tx0_data [31:24] tx0_kchar [2] 对应 tx0_data [23:16] tx0_kchar [1] 对应 tx0_data [15:8] |

| | | | |
|---------------------|----|----|---|
| | | | tx0_kchar [0] 对应 tx0_data [7:0] |
| rx0_clk | 1 | 输出 | 数据接收时钟，也就是第十四部分介绍的 RXUSRCLK2，频率为 GTP 的参考时钟 125Mhz。数据在上升沿有效。 |
| rx0_data | 32 | 输出 | GTP 接收数据。 |
| rx0_kchar | 4 | 输出 | GTP 接收 K 控制字，用来指示接收的数据是 K 码控制字符还是正常传输数据。高电平表明是 K 码控制字符，4 位对应接收数据 32 位的 4 个 Byte。 rx0_kchar [3] 对应 rx0_data [31:24] rx0_kchar [2] 对应 rx0_data [23:16] rx0_kchar [1] 对应 rx0_data [15:8] rx0_kchar [0] 对应 rx0_data [7:0] |
| gt0_tx fsmresetdone | 1 | 输出 | GTP 初始化完成信号。 |

知道了 GTP 用户接口信号的含义，我们就能通过用户接口实现光纤数据的收发了。

2) gtp 数据包准备模块 video_packet_send.v

接收到的视频图像数据存放在 16 位进，32 位出的 FIFO 中，在 video_packet_send.v 中会由一个状态机来发送视频图像的数据，首先一帧图像开始传输前，GTP 会发送帧同步信号。再判断这个 FIFO 中的数据量，如果 FIFO 内的数据还没有一行的视频数据，GTP 则发送无用的数据，当 FIFO 内已经有一行视频的数据时，GTP 会先发送行同步信号，然后再把这一行视频的数据通过 GTP 发送出去。一行数据发送完成，再重新判断 FIFO 内的数据量，FIFO 数据量达到一行视频数据时，接着发送第二行视频图像。GTP 数据发送一帧图像的流程如下图：



所有的 GTP 发送的数据位数为 32 位，帧同步信号、行同步信号、无用数据定义如下：

图像帧同步信号 0：定义为 32 位的 “ff_00_00_bc”

图像帧同步信号 1：定义为 32 位的 “ff_00_01_bc”

GTP 插入的无用数据 0：定义为 32 位的 “ff_55_55_bc”

GTP 插入的无用数据 1：定义为 32 位的 “ff_aa_aa_bc”

图像行同步信号：定义为 32 位的 “ff_00_02_bc”

这些同步信号和无用数据的高 24 位数据是用户自己定义的，低 8 位“bc”是 K28.5 码控制字符。
K 码特征字定义在 Xilinx 的 “ug482_7Series_GTP_Transceivers.pdf” 文档里有描述。

Table C-2: Valid Control K Characters

| Special Code Name | Bits HGF EDCBA | Current RD – abcdei fghj | Current RD + abcdei fghj |
|----------------------|----------------|--------------------------|--------------------------|
| K28.0 | 000 11100 | 001111 0100 | 110000 1011 |
| K28.1 | 001 11100 | 001111 1001 | 110000 0110 |
| K28.2 | 010 11100 | 001111 0101 | 110000 1010 |
| K28.3 | 011 11100 | 001111 0011 | 110000 1100 |
| K28.4 | 100 11100 | 001111 0010 | 110000 1101 |
| K28.5 | 101 11100 | 001111 1010 | 110000 0101 |
| K28.6 | 110 11100 | 001111 0110 | 110000 1001 |
| K28.7 ⁽¹⁾ | 111 11100 | 001111 1000 | 110000 0111 |
| K23.7 | 111 10111 | 111010 1000 | 000101 0111 |
| K27.7 | 111 11011 | 110110 1000 | 001001 0111 |
| K29.7 | 111 11101 | 101110 1000 | 010001 0111 |
| K30.7 | 111 11110 | 011110 1000 | 100001 0111 |

向 GTP 发送 K28.5 码控制字符时，需要拉高 gt_tx_ctrl 信号的对应位，标示发送数据里的某个字节位为 K 码控制字。所以这里在向 GTP 发送同步信号和无用数据的时候，gt_tx_ctrl 信号设置为 0001，发送视频数据的时候则置为 0000。

```
105 case(state)
106     SEND_FRAME_SYNC0: //发送视频的帧同步信号0
107     begin
108         state <= SEND_FRAME_SYNC1;
109         gt_tx_data <= 32'hff_00_00_bc;
110         gt_tx_ctrl <= 4'b0001;
111     end
112     SEND_FRAME_SYNC1: //发送视频的帧同步信号1
113     begin
114         state <= SEND_OTHER0;
115         gt_tx_data <= 32'hff_00_01_bc;
116         gt_tx_ctrl <= 4'b0001;
117     end
118     SEND_OTHER0: //发送视频的无用的信号ff_55_55_bc
119     begin
120         state <= SEND_OTHER1;
121         gt_tx_data <= 32'hff_55_55_bc;
122         gt_tx_ctrl <= 4'b0001;
123     end
124     SEND_OTHER1: //发送视频的无用的信号ff_aa_aa_bc
125     begin
126         if(buffer_out == min_width) //判断fifo中是否有一行的视频图像数据
```

3) 位数据对齐模块 word_align.v

GTP 收发器外部用户数据接口的宽度为 32 位，内部数据宽度为 20 位(8b/10b 转换)。在实际测试过程中发现，发送的 32 位数据会有可能出现 16 位的数据的移位，就是说发送的数据和接收到的数据会有 16 位的错位，下表演示 GTP 发送数据和接收数据移位的情况：

| GTP 发送的数据 | | GTP 接收的数据 | |
|-----------|----------|-----------|-----------|
| 数据 1 | 11111111 | 数据 1 | 11112222 |
| 数据 2 | 22222222 | 数据 2 | 22223333 |
| 数据 3 | 33333333 | 数据 3 | 33334444 |
| 数据 4 | 44444444 | 数据 4 | 44445555 |
| 数据 5 | 55555555 | 数据 5 | 5555..... |
| | | | |

因为我们在 GTP 发送同步信号和无用数据的时候加入了 K 码控制字，并且设置 gt_tx_ctrl 信号为 0001, 如果出现 16 位数据移位的情况，接收到的同步信号和无用数据时，K 码控制字也会跟着移位，gt_tx_ctrl 的信号就会变成 0100。所以我们在程序可以通过判断 gt_tx_ctrl 信号的值来判断接收到的 GTP 数据是否移位，如果接收到的 gt_tx_ctrl 为 0001，跟我们发送的时候一样，说明数据没有移位；如果接收到的 gt_tx_ctrl 为 0100，接收到的数据移位，需要重新组合，在 word_align.v 模块里完成。

```
always@(posedge rx_clk)
begin
    case(align_bit)
        4'b0001:
            rx_data_align <= gt_rx_data;
        4'b0100:
            rx_data_align <= {gt_rx_data[15:0], gt_rx_data_d0[31:16]};
        default:
            rx_data_align <= 32'd0;
    endcase
end
```

4) GTP 视频数据解析模块 video_packet_rec.v

因为接收到的 32 位数据中只有一部分是视频图像的数据，其它的是帧同步，行同步和无用的数据，在 video_packet_rec.v 模块里需要把视频图像的数据解析出来存入到一个 32 位进，8 位出的 FIFO 中。

程序的一个功能是检测 GTP 数据中的行同步信号（数据为 ff_00_02_bc），如果接收到行同步信号，就把后面接收的一行视频数据存放到 FIFO 中。

```
//解析行同步信号
always@(posedge rx_clk or posedge rst)
begin
    if(rst)
        wr_en <= 1'b0;
    else if(gt_rx_ctrl == 4'b0001 && gt_rx_data == 32'hff_00_02_bc)
        wr_en <= 1'b1;
    else if(wr_cnt == ({1'b0, vout_width[15:1]} - 16'd1))
        wr_en <= 1'b0;
end
```

程序的另一个功能是恢复视频图像的帧同步信号（数据为 ff_00_00_bc），如果接收到帧同步信号，则置位视频图像的帧信号 vs。

```

24  //恢复帧同步信号
25  always@(posedge rx_clk or posedge rst)
26  begin
27      if(rst)
28          vs_r <= 1'b0;
29      else if(gt_rx_ctrl == 4'b0001 && gt_rx_data == 32'hff_00_00_bc)
30          vs_r <= 1'b1;
31      else if(vs_cnt > 16'd100)
32          vs_r <= 1'b0;
33  end

```

另外模块中会判断 FIFO 中存入的视频数据，如果 FIFO 内的数据量大于一行视频的数据，则产生 FIFO 的读使能信号，把 FIFO 的一行视频数据输出给外部接口模块。

其它模块在这里我们不做介绍了。

5) 管脚约束

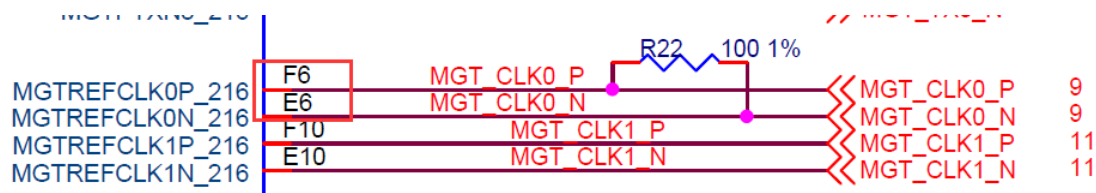
这里的管脚约束是在 gtp IP 的 example 工程中的 gtp_exdes.xdc 文件中修改而来，比如 GTP 的参考时钟输入管脚，这里需要跟开发板上的管脚对应。

```

##### RefClk Location constraints #####
set_property PACKAGE_PIN E6 [get_ports Q0_CLK0_GTREFCLK_PAD_N_IN]
set_property PACKAGE_PIN F6 [get_ports Q0_CLK0_GTREFCLK_PAD_P_IN]

```

对应 AX7102 开发板原理图上的对应管脚为 F6, E6。



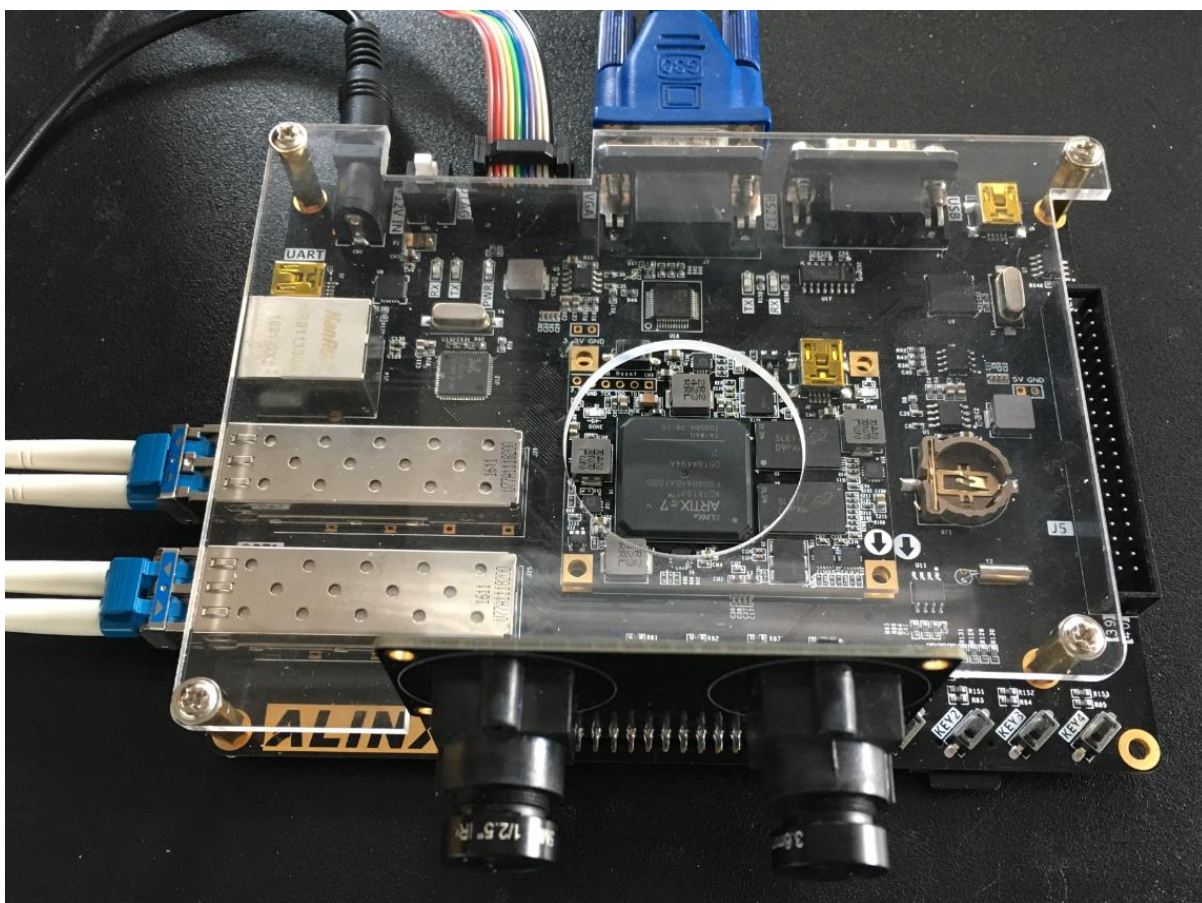
添加 SFP 光模块的发送控制管脚的定义如下：


```
set_property IOSTANDARD LVCMOS33 [get_ports {tx_disable[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {tx_disable[0]}]
set_property PACKAGE_PIN C22 [get_ports {tx_disable[0]}]
set_property PACKAGE_PIN C20 [get_ports {tx_disable[1]}]
```

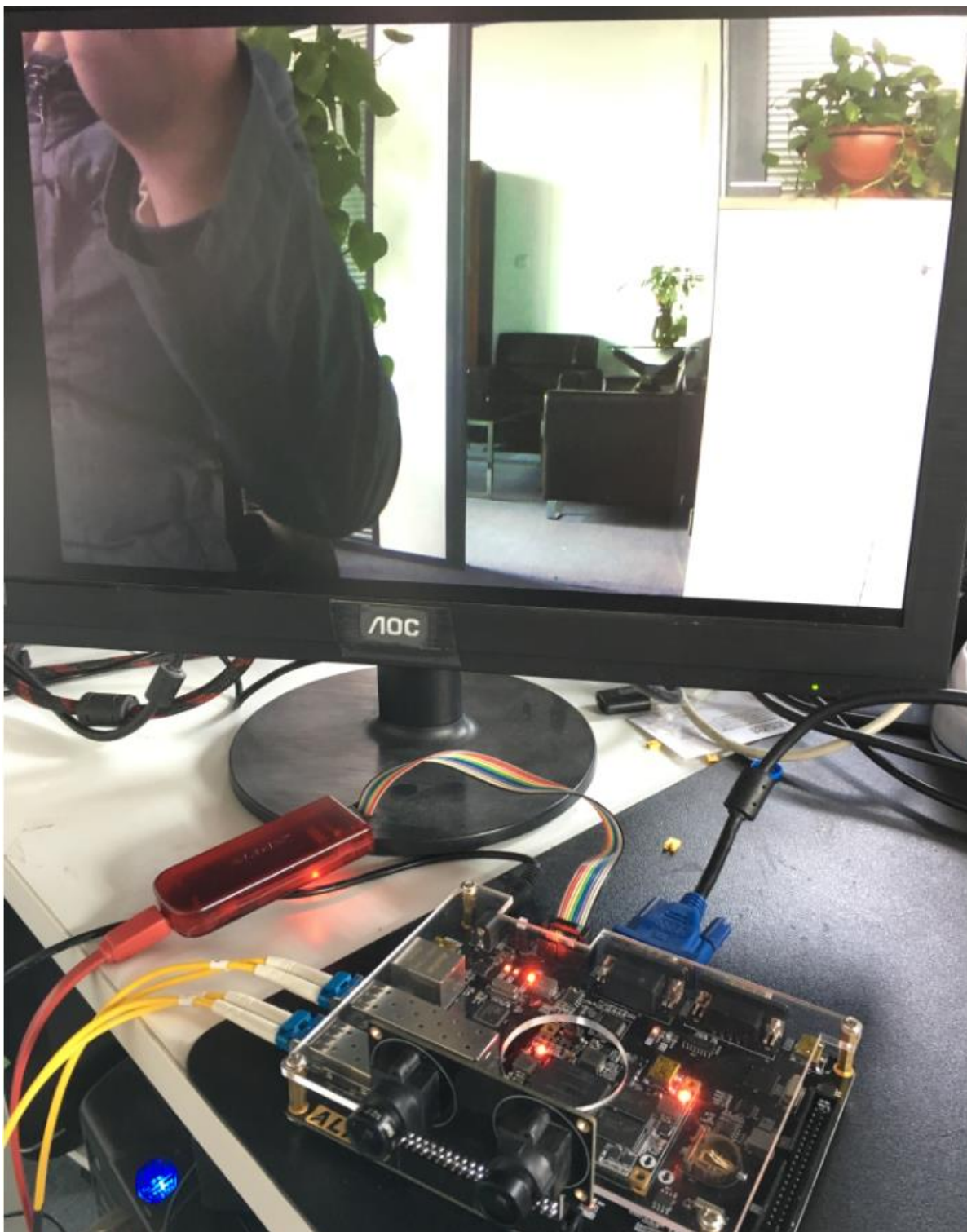
另外在 gtp_exdes.xdc 文件中添加程序中用到的复位信号，双目摄像头的接口信号，VGA 显示信号的管脚定义和 DDR 的管脚定义。

4 光纤视频传输现象

编译项目通过后我们就可以开始 AN5642 双目光纤视频传输的实验了。开发板的扩展口(J4)上插上摄像头 AN5642 模块，开发板的 VGA 口连接 VGA 显示器，光模块插入到 OPT1~OPT2 的接口上，再连接光纤。硬件连接后如下图所示：



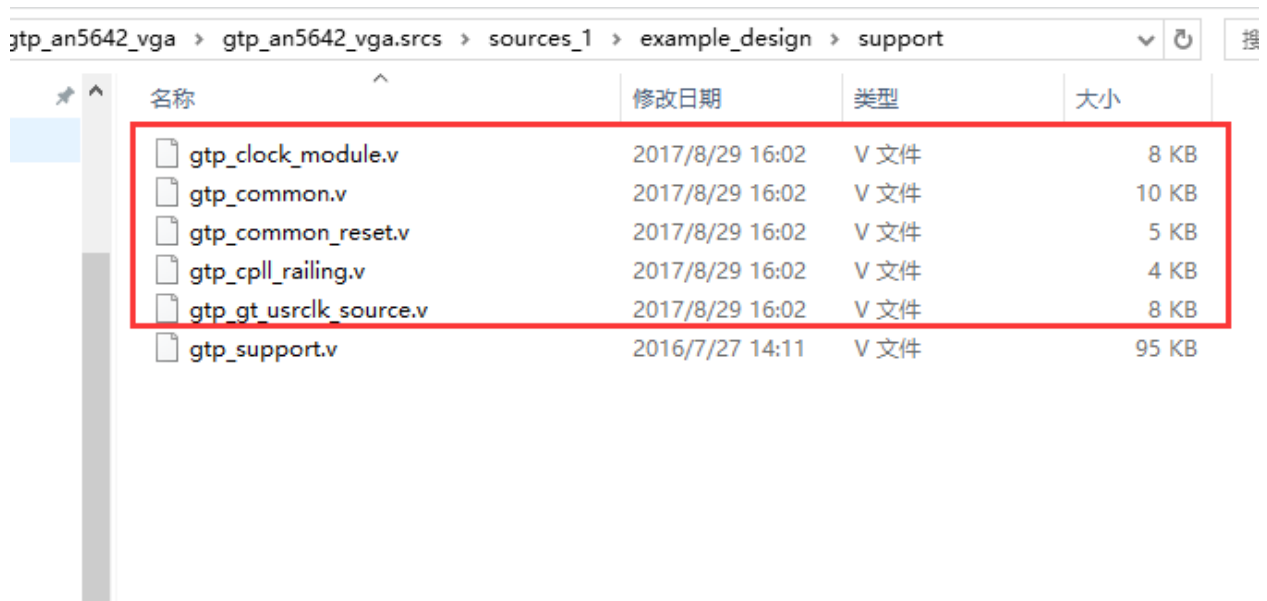
再下载 bit 文件到 FPGA, 我们就可以在 VGA 显示器上看到视频图像从 AN5642 采集后 FPGA 先通过光纤传输，再环路回 FPGA，然后在 VGA 显示器上显示的视频图像了。



按一下开发板上的 KEY2 按钮，视频图像会切换到另外一路的摄像头。程序中设计的是 4 个 GTP Channel (Channel0~3) 同时发送视频数据 (在 AX7102 开发板上只有 Channel0 和 Channel1 连

接了光模块)，但只有接收 GTP Channel1 的视频图像数据显示，因为 GTP Channel1 对应的是 OPT2，所以这里的视频图像是从 OPT1 光模块传输到 OPT2 的光模块。

如果用户需要对 gtp_ov5640_vga 工程中的 gtp IP 重新配置的话，如果只是修改工程里的 gtp IP 配置的话，编译的时候会报错。用户需要在修改 gtp_ov5640_vga 工程中的 gtp IP 配置的同时，还是需要重新生成 gtp IP 的 example 文件，然后用 example 工程中生成的文件去替换 gtp_ov5640_vga 工程中的以下文件。



| gtp_an5642_vga > gtp_an5642_vga.srscs > sources_1 > example_design > support | | | | | ▼ | 🔄 | 🔍 |
|--|-----------------|------|-------|--|---|---|---|
| 名称 | 修改日期 | 类型 | 大小 | | | | |
| gtp_clock_module.v | 2017/8/29 16:02 | V 文件 | 8 KB | | | | |
| gtp_common.v | 2017/8/29 16:02 | V 文件 | 10 KB | | | | |
| gtp_common_reset.v | 2017/8/29 16:02 | V 文件 | 5 KB | | | | |
| gtp_cppll_railing.v | 2017/8/29 16:02 | V 文件 | 4 KB | | | | |
| gtp_gt_usrclk_source.v | 2017/8/29 16:02 | V 文件 | 8 KB | | | | |
| gtp_support.v | 2016/7/27 14:11 | V 文件 | 95 KB | | | | |