Dual Lens Camera Module AN5642 User Manual

ALINX

Version Record

Version	Date	Release By	Description
Rev 1.0	2022-04-30	Rachel Zhou	First Release

The ALINX official Documents are in Chinese, and the English version was translated by **Shanghai Tianhui** Trading Company. They has **not been officially Review by ALINX** and are for reference only. If there are any errors, please send email feedback to support@aithtech.com for correction.

Amazon Store: https://www.amazon.com/alinx

Aliexpress Store:

https://alinxfpga.aliexpress.com/store/911112202?spm=a2g0o.detail.1000007.1.704e2bedqLBW90

Ebay Store: https://www.ebay.com/str/alinxfpga

Customer Service Information

Skype: rachelhust@163.com

Wechat: rachelhust

Email: support@aithtech.com and rachehust@163.com

Table of Contents

Part 1：Dual Lens Camera Module General Description

The dual lens camera module AN5642, use 2 pieces of CMOS chip image sensor OV5640 from OmniVision Corporation of the United States, which supporting two independent or simultaneous display functions. The CMOS OV5640 chip supports DVP and MIPI interfaces. On the OV5642 module, images are transmitted through the DVP interface and the FPGA connection. Figure 1-1 detailed the AN5642 module product photo.

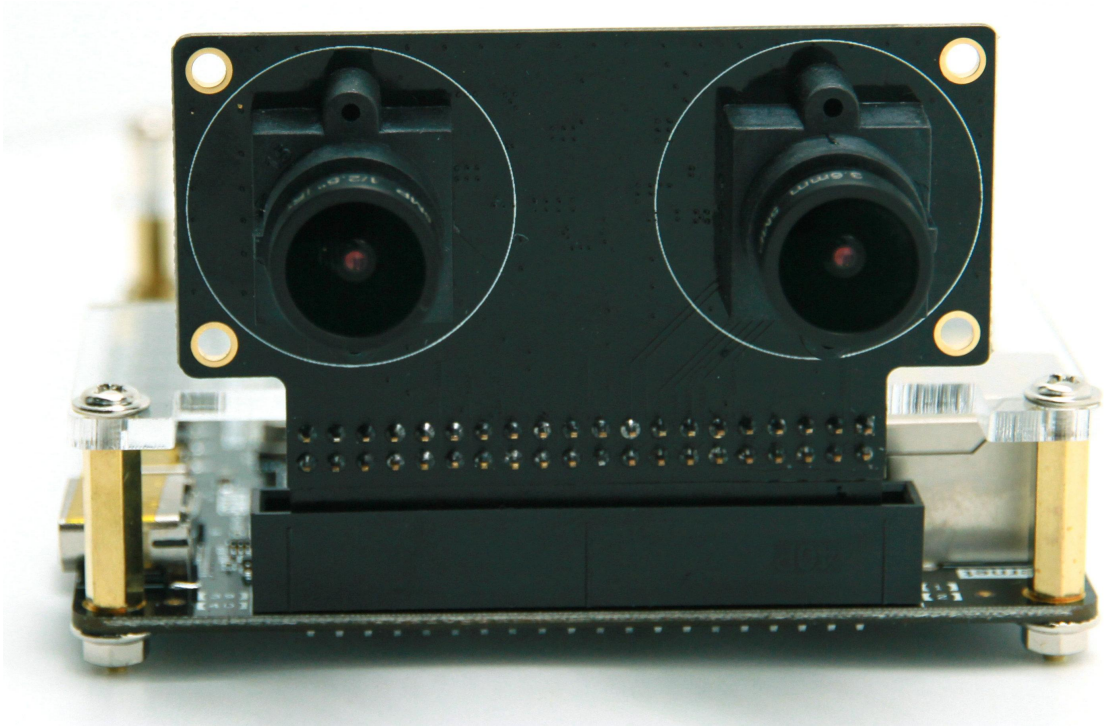


Figure 1-1: AN5642 module product photo

Part 1.1：AN5642 Dual Lens Module Detail Parameter

- AN5642 dual lens camera module detail parameter listed as below:
- Module Interface: 40-pin 0.1 spacing female header, 2-way camera with separate DVP interface
- Spacing：The distance between the 2-way cameras is 40mm
- support for images sizes: 5 megapixel
- Photosensitive chip: 2 pieces of OV5640

- optical size of 1/4"
- Module content: OV5640 power supply circuit and clock
- automatic image control functions: Manual focus, automatic exposure control (AEC), automatic white balance (AWB)
- support for output formats: RAW RGB, RGB565/555/444, CCIR656, YUV422/420, YCbCr422, and compression
- maximum image transfer rate:

  QSXGA (2592x1944): 15 fps

  1080 30 fps

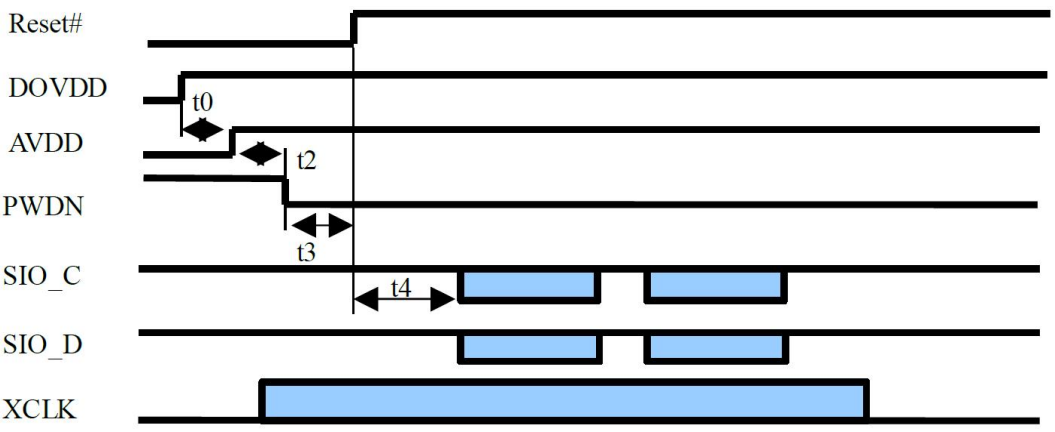
  1280x960: 45 fps

  720p: 60 fps

  VGA (640x480): 90 fps

  QVGA (320x240): 120 fps
- Working temperature: -30~70°C, stable working temperature is 0~50°C

Part 1.2：Chip OV5640 power-on requirements



t0:>=0ms. Delay from DOVDD stable to AVDD stable

t2:>=5ms. Delay from AVDD stable to sensor power up stable

t3:>=1ms. Delay from sensor power up stable to Reset# pull high

t4:>=20ms. Delay from Reset pull high to SCCB initialization

The power-on steps of the chip OV5640 are as follows:

- Step 1: ResetB is pulled low, reset AN5640, PWDN is pulled high
- Step 2: Both DOVDD and AVDD are powered up simultaneously, which is implemented in the power supply design of the module.
- Step 3: After 5ms of AVDD reaching stable, pull PWDN to low.
- Step 4: After 1ms of PWDN go low, pull high ResetB
- Step 5: After 20ms, initialize OV5640 by SCCB initialization:

Part 1.3：Register Configuration of CMOS Chip OV5640

The register configuration of the OV5640 is configured through the I2C interface of the FPGA (or other CPU). The user needs to configure the correct register value to let the OV5640 output the image format we need. In our example, we configure the OV5640 as the image format of the 720P (1280x720) video output image and the RGB565 frame rate of 30fps.

To facilitate debugging, the user can configure registers to enable internal test images of the OV5640, such as color bars and color squares.

Color bar

write_i2c(0x503d, 0x80);

write_i2c(0x4741, 0x00);

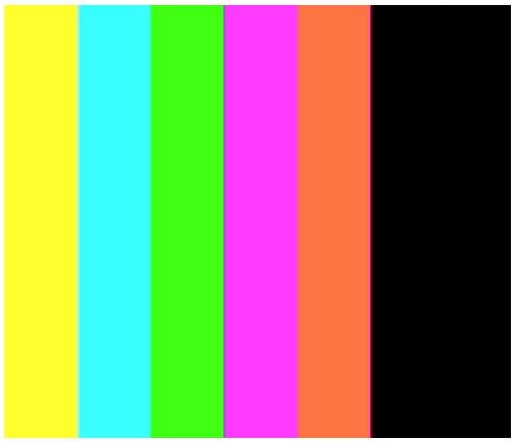


Figure 1-2: Color Bar

Color square

write_i2c(0x503d, 0x82);

write_i2c(0x4741, 0x0);

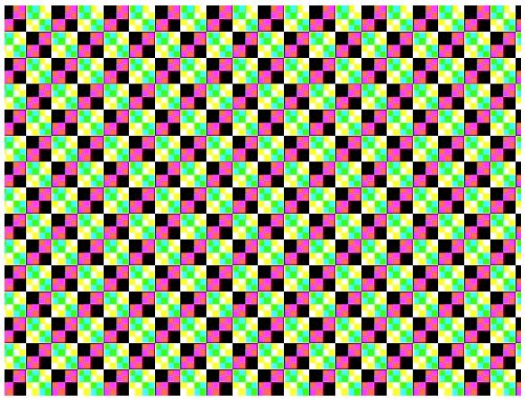


Figure 1-3: Color Square

The data format of the OV5640's camera output is configured in the following 0x4300 registers. In our example, the OV5640 is configured as an RGB565 output format.

0x4300	FORMAT CONTROL 00	0xF8	RW

Format Control 00
Bit[7:4]: Output format of formatter module
0x0: RAW
Bit[3:0]: Output sequence
0x0: BGBG... / GRGR...
0x1: GBGB... / RGRG...
0x2: GRGR... / BGBG...
0x3: RGRG... / GBGB...
0x4~0xF: Not allowed
0x1: Y8
Bit[3:0]: Does not matter
0x2: YUV444/RGB888 (not available for full resolution)
Bit[3:0]: Output sequence
0x0: YUVYUV..., or GBRGBR...
0x1: YVUYVU..., or GRBGRB...
0x2: UYVUYV..., or BGRBGR...
0x3: VYUVYU..., or RGBRGB...
0x4: UVYUVY..., or BRGBRG...
0x5: VUYVUY..., or RBGRBG...
0x6~0xE: Not allowed
0xF: UYVUYV..., or BGRBGR...
0x3: YUV422
Bit[3:0]: Output sequence
0x0: YUYV...
0x1: YVYU...
0x2: UYVY...
0x3: VYUY...
0x4~0xE: Not allowed
0xF: UYVY...
0x4: YUV420

There are many more registers on the OV5640, but many register users do not need to understand, the configuration of the registers can be configured according to the OV5640 application guide. If you want to know more about the register information, you can refer to the register description in the OV5640 datasheet.

Part 2：Hardware Connection

The following ALINX AX516 development board is an example to introduce the hardware connection of the dual lens camera module AN5642 and ALINX serial FPGA development kit. The 40-pin female headers of module plug into the expansion board of FPGA development kit. Figure 2-1 detailed the 40-pin female headers of module.

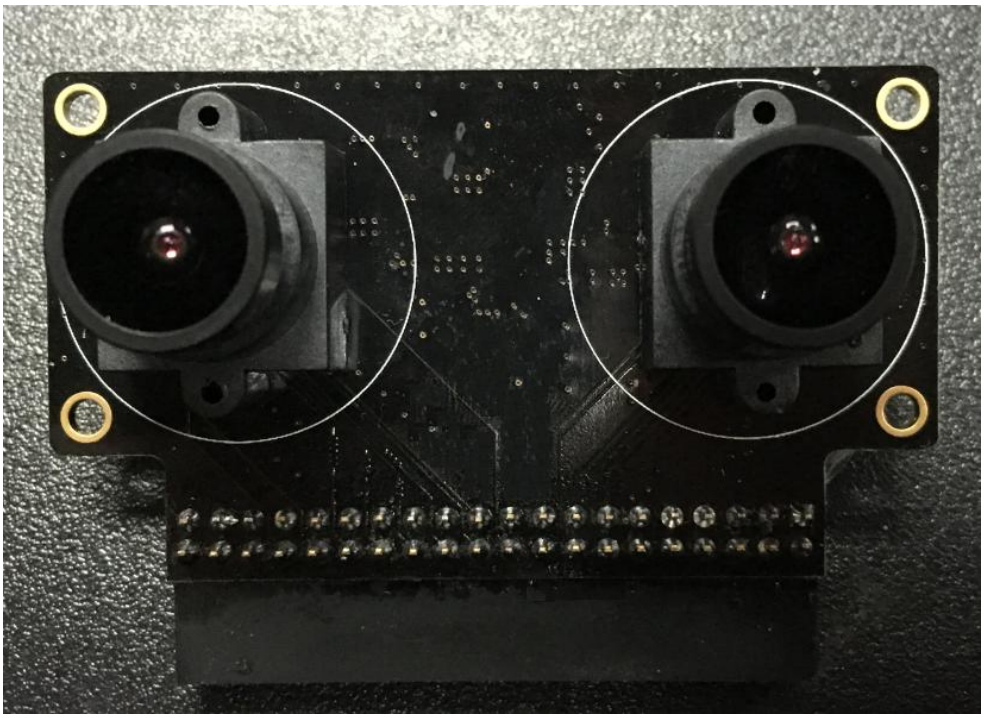


Figure 2-1: 40-pin Female header of Camera Module

The 40-pin female header of camera module detailed as below:

Pin	Pin Name	Pin	Pin Name
Pin1	Ground	Pin2	+5V

Pin3	CMOS2_D9	Pin4	CMOS2_SDA
Pin5	CMOS2_D6	Pin6	CMOS2_SCL
Pin7	CMOS2_D7	Pin8	CMOS_D2
Pin9	CMOS2_HREF	Pin10	CMOS2_D8
Pin11	CMOS2_D3	Pin12	CMOS2_RESET
Pin13	CMOS2_D4	Pin14	CMOS2_D5
Pin15	CMOS2_D1	Pin16	CMOS2_D0
Pin17	CMOS2_VSYNC	Pin18	CMOS2_PCLK
Pin19	NC	Pin20	CMOS1_SDA
Pin21	CMOS1_D9	Pin22	CMOS1_D8
Pin23	CMOS1_SCL	Pin24	CMOS1_D5
Pin25	CMOS1_D3	Pin26	CMOS1_D4
Pin27	CMOS1_D6	Pin28	CMOS1_D0
Pin29	CMOS1_D7	Pin30	CMOS1_D1
Pin31	CMOS1_D2	Pin32	CMOS1_PCLK
Pin33	CMOS1_HREF	Pin34	CMOS1_VSYNC
Pin35	CMOS1_RESET	Pin36	NC
Pin37	Ground	Pin38	Ground
Pin39	+3.3V（or NC）	Pin40	+3.3V(or NC）

CMOSx_D0~D9 is the video image captured by the camera OV5640. If the input selects the RGB format, only the upper 8 bits D2~D9 are needed. CMOSx_PCLK is the pixel clock signal, and the rising edge of the clock collects data. CMOSx_HREF is a line valid signal. When the signal is high, one line of data is valid. CMOSx_VSYNC is a column sync signal. When configured in VGA mode (640*480), the timing of the CMOS camera output is shown below:

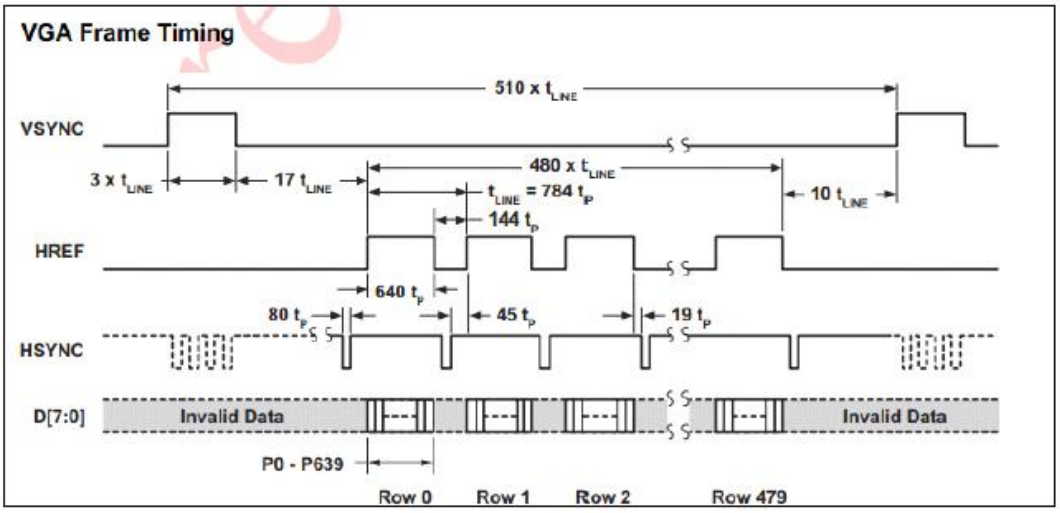


Figure 2-2: VGA Frame Timing

The connection between the camera module and the development board is relatively simple. Just plug the camera module directly into the expansion port of the development board. When inserting, you need to pay attention to the alignment of the PIN1 pin. The Figure 2-3 is a schematic diagram of the connection between the AN5642 module and the AX516 development board (if the AX515 or AX530 development board needs to be plugged into the J3 port).

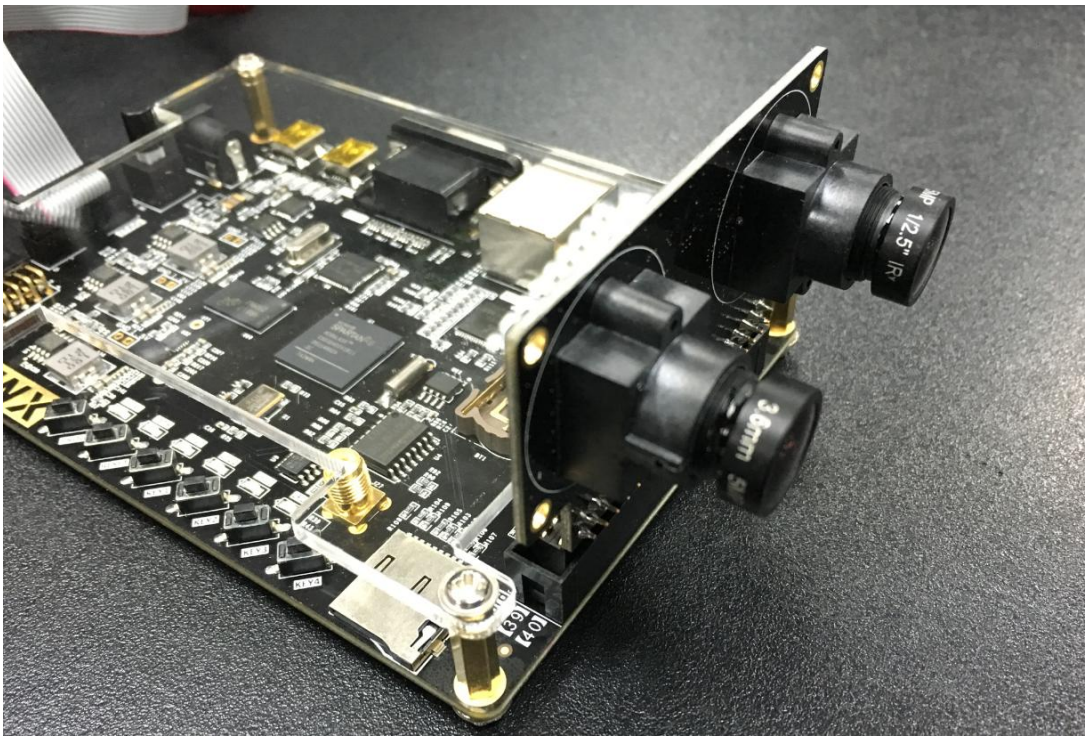


Figure 2-3: The Camera Module Connected with AX516 FPGA Board

Part 3：Binocular switching display experiment

In this experiment, the image of the 2-way camera is displayed on the VGA display, and the 2-way video image is switched by the button KEY1 on the development board. The size of each video image displayed on the VGA display is 720P.

Part 3.1：Programming

The logic block diagram of the FPGA program design is shown in figure 3-1 below:

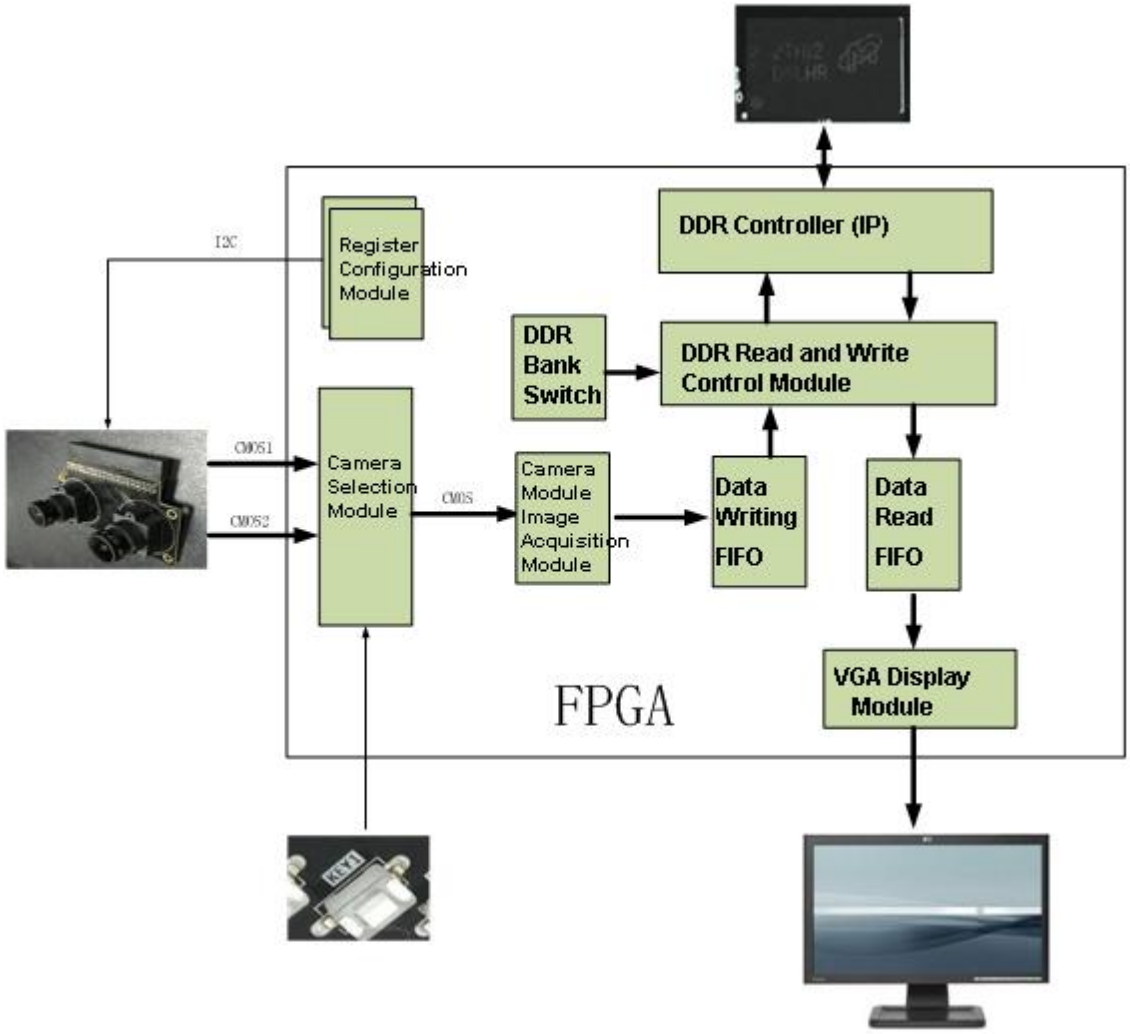


Figure 3-1: The logic block diagram of the FPGA program

After power-on, the FPGA program first configures the binocular camera register through the I2C bus, and configures the OV5640 camera to work in 720P 30-frame video output. After the data output by the 2-way camera is sent to the FPGA, the camera signal selection module is used to select one of the videos to enter the camera data acquisition module.

The camera data acquisition module converts the video image into 32-bit or 64-bit wide data and stores it in the write FIFO. When the data in the write FIFO reaches a certain amount, a burst write signal is generated to the DDR read/write control module, and the DDR read/write control module will write a certain length of data from the write FIFO and writes it to the DDR chip. If the data in the read FIFO is less than a certain amount, a burst read signal will be generated to the DDR read/write control module, and the DDR read/write control module will read a certain length of data from the DDR into the read FIFO. The VGA display module reads the data in the read FIFO and displays the video image onto the VGA display.

The completed engineering architecture is shown in the figure below (Take AX516 as an example):

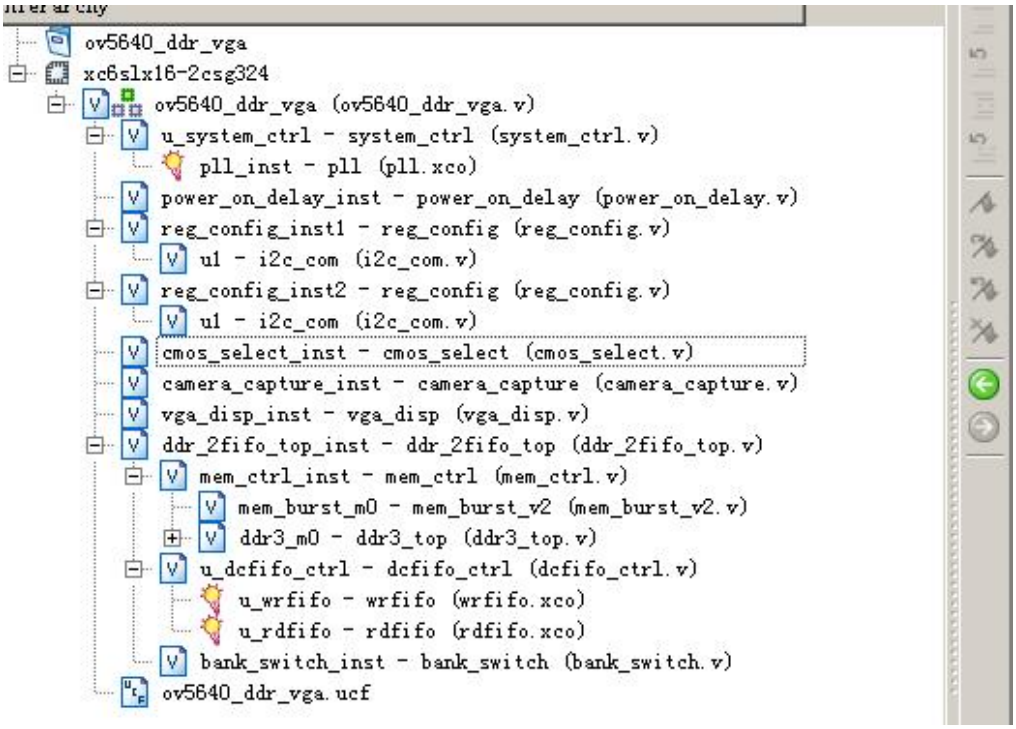


Figure 3-2: Completed Engineering Architecture of AX516

The function description of each module, detailed as below:

1). Power-on waiting program：power_on_delay.v

Because the OV5640 chip has power-on timing requirements, this program is waiting for a period of time after the FPGA is powered on and then enabling the OV5640 register to meet the timing requirements of the OV5640.

2). OV5640 register configuration module：reg_config.v

After power-on the FPGA, the register configuration program of the OV5640 calls the I2C communication program to set the parameters of the registers of the two OV5640 chips. Here the images output by the two OV5640 chips are set to RGB565 format, and the numbers of image is 1280*720.

3). Camera Signal Selection Module：cmos_select.v

The program detects the button Key1, and if the button KEY1 is pressed once, it switches to the signal of the other camera. The default is to select to capture CMOS1 video images.

4). Camera Image Acquisition Module：COMS_Capture.v

The camera image acquisition program converts the 8-bit image from the OV5640 module into a 32-bit or 64-bit data width and produces a FIFO write signal. Because the camera RGB565 format output, an image requires 16 bits of data, it needs to be divided into two 8-bit output, which occupies 2 clocks. So if it is a 720P image, a row of data needs to have 1280 * 2 PCLK clocks.

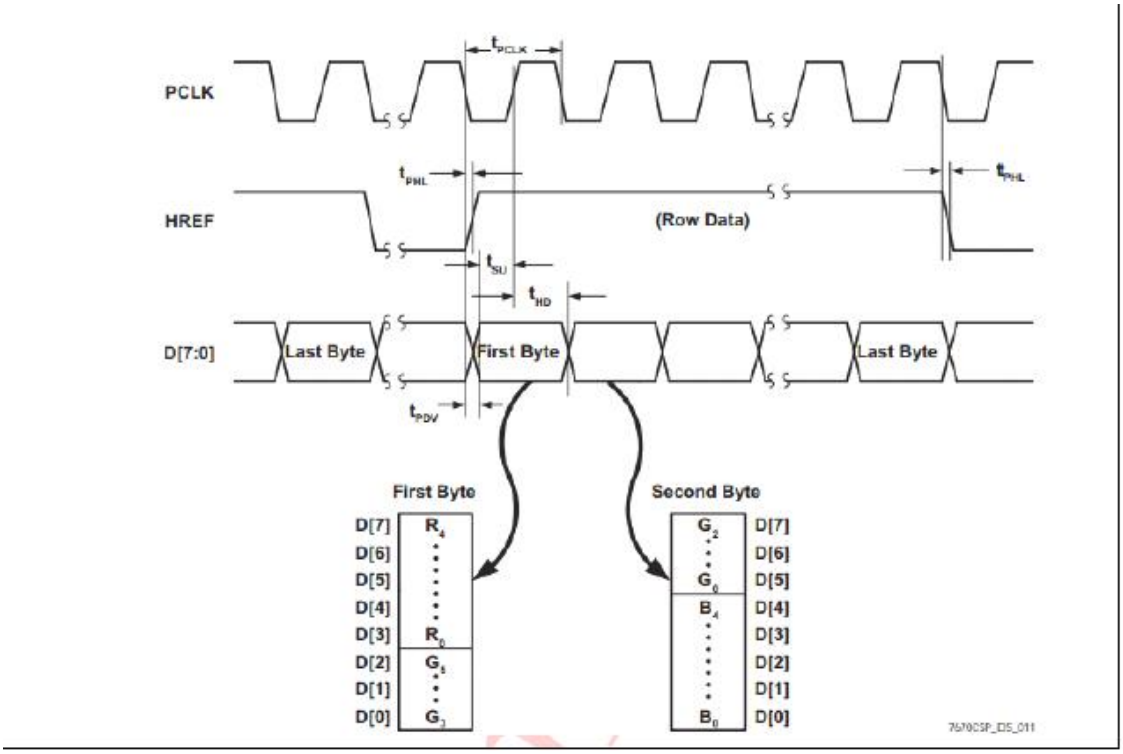


Figure 3-3: RGB 585 Output Timing Diagram

5). FIFO Control Program Module: dcfifo_ctrl.v

When storing the captured video image to the DDR or reading the video image from the DDR to the VGA display module, the data needs to be cached. So we initialized two FIFOs in the Dcfifo_ctrl.v module, a data write FIFO to buffer the data collected from the camera, and a data read FIFO to store the data read from the DDR. In this experiment, the data written to the DDR is first stored in the write FIFO, and the data read from the DDR is first stored in the read FIFO.

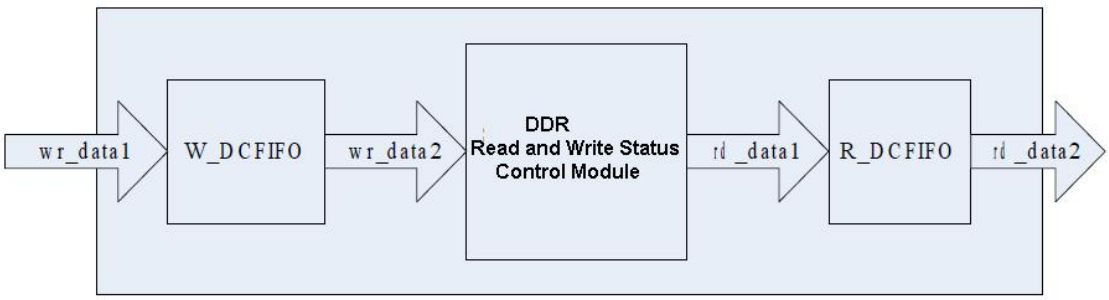


Figure 3-4: FIFO Control Program Module

A DDR burst write request is generated when the amount of data in the write FIFO is greater than the set write burst length (128)

```
158     end
159     else if(ddr_init_done == 1'b1)
160       begin                          //写入优先，带宽内防止数据丢失
161       if(wrf_use >= wr_length & !wr_flag)  //写入FIFO的数据数量超过burst长度,写DDR开始
162         begin                        //wrfifo满突发长度
163         ddr_wr_req <= 1;      //写ddr使能
164         wr_flag <= 1;
165         end
166       else if(ddr_wr_finish)//读FIFO里的数据数量小于burst长度,读DDR开始
167         begin              //rdfifo满突发长度
168         ddr_wr_req <= 0;      //写ddr空闲
169         wr_flag <= 0;
```

Figure 3-5: Generate DDR Burst write request program

The Sdram read request is generated when the amount of data in the read FIFO is less than the set read burst length (128).

```
191     else if(ddr_init_done == 1'b1)
192       begin                          //写入优先，带宽内防止数据丢失
193       if(rdf_use <= rd_length &  !rd_flag)//读FIFO里的数据数量小于burst长度,读DDR开始
194         begin              //rdfifo满突发长度
195         ddr_rd_req <= 1;      //读ddrm使能
196         rd_flag <= 1;
197         end
198       else if(ddr_rd_finish)
```

Figure 3-6: Generate Sdram readr request program

6). DDR Read and Write Control Module: mem_ctrl.v

The function of the mem_burst_v2 program is to convert the external burst read request and write request into the required signals and timing of the Local Bus of the DDR IP interface. This makes the user not need to care about the complex timing of the DDR interface, making it easy and convenient to read and write DDR. The mem_burst_v2 module completes the control of DDR with the DDR IP generated by XILINX/Altera software. The port signal of the circle in the figure 3-6 below is the signal of the user terminal. By controlling these signals, the data of DDR can be read and written.

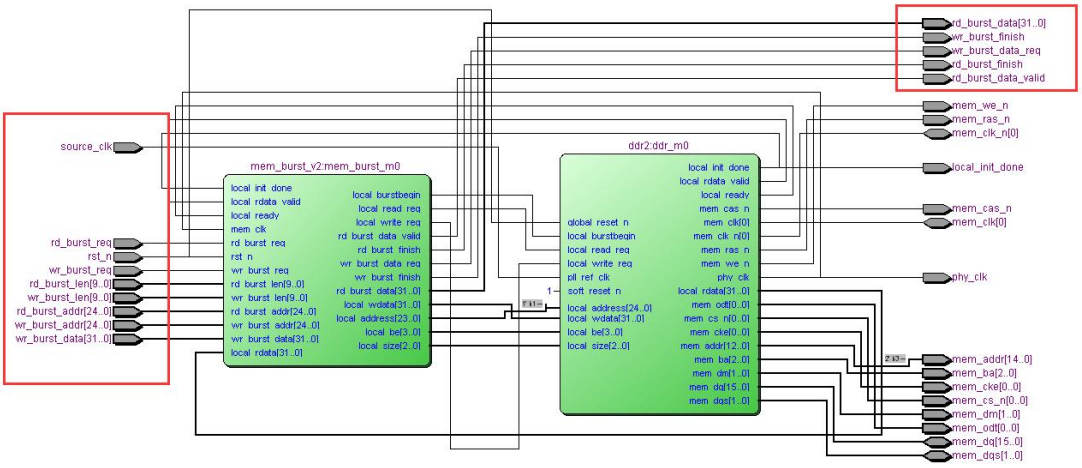


Figure 3-7: The signal of the user terminal

Through the DDR read and write control module, DDR read and write operations become very simple. For example, when performing a Burst read on DDR, set a read request (rd_burst_req is high), and set the read burst length rd_burst_len and the read address rd_burst_addr. Waiting for the rd_burst_data_valid signal to be high indicates that the data is valid, reading the data, and after reading the burst length data, the rd_burst_finish signal goes high.

7). DDR Controller Module：ddr3_top.v

The DDR controller module and its submodules are generated by the IP provided by Xilinx or Altera. For details on how to generate the DDR controller, please refer to the corresponding development board (DDR read and write routines).

8). VGA Display Program: vga_disp.v

The vga_disp.v module implements image display of a VGA display, and generates line synchronization, column synchronization, and timing of image data signals according to the VGA timing standard. VGA clock frequency: Take 1280x768@79.5MHz (60Hz) as an example. Each field corresponds to 798 line

periods, of which 1280 is the display line. Each display line includes 1664 points of clock, of which 1280 points are valid display areas. It can be seen that the clock frequency of VGA is required: 1664*798*60 is about 79.5MHz. Figure 3-7 and figured 3-8 detailed the timing diagram of VGA:

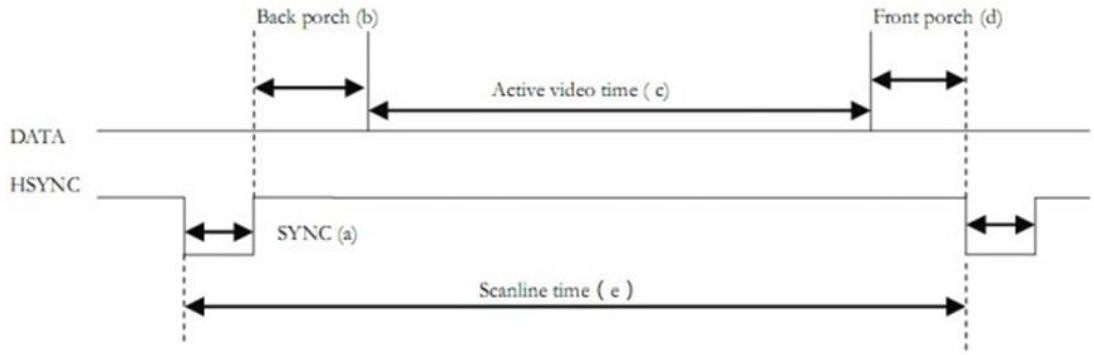


Figure 3-8: VGA Line Time

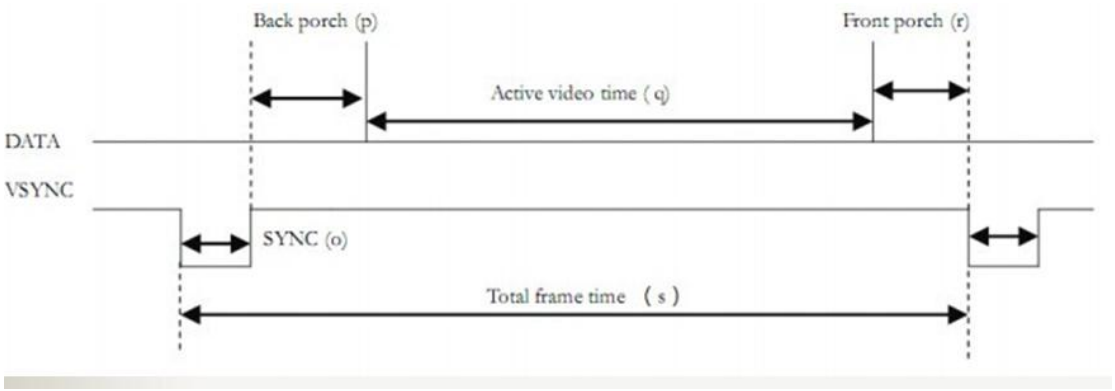


Figure 3-9: VGA Filed Timing

In addition, because the video image output by OV5640 is 1280x720 image size, but the VGA display is 1280x768 image, in order to make the video image in the middle of the VGA display, the image is not displayed in the first 24 lines and the last 24 lines of the VGA image. (Default is black):

```
38    parameter V_FrontPorch=3;          //显示前沿（Front porch r）
39    parameter Vde_start=51;            //27+24 实际显示720
40    parameter Vde_end=771;             //51+720个实际显示720
41
//------------------------------------------------------------//
```

Figure 3-10: Video Image in the Middle of the VGA display

9). ddr Bank Exchange Progrm: bank_switch.v

Bank exchange program realizes ddr read and ddr write respectively operate in different Bank operations. For example, when Bank0 writes the image captured by the camera, VGA reads the data display of Bank3; After Bank0 writes a video image, it switches to DDR to write the next frame image. Bank1 writes the next frame of image. After one frame of VGA is displayed, the display of the next frame will be read from bank0 of DDR. This is displayed and stored in different BANKs to avoid image distortion.

10). System Control Program: system_ctrl.v

The camera's clock (24Mhz) and VGA's image clock (79.5Mhz) are generated, and the program also generates a reset signal for the entire system.

Part 3.2：OV5642 VGA binocular display experiment

After writing the program, assign the Pin of the FPGA, and after recompilation, we can start the camera VGA display experiment. The development board connects with the camera module OV5642 and VGA interface to connect VGA display, then downloads the sof file to the FPGA, we can see 1280x768 video images on the VGA display. The default VGA display shows the video image of the camera CMOS1. We just press the development board button KEY1, and the VGA display will display the CMOS2 video image. Figure 3-11 and Figure 3-12 detailed the video image effect.

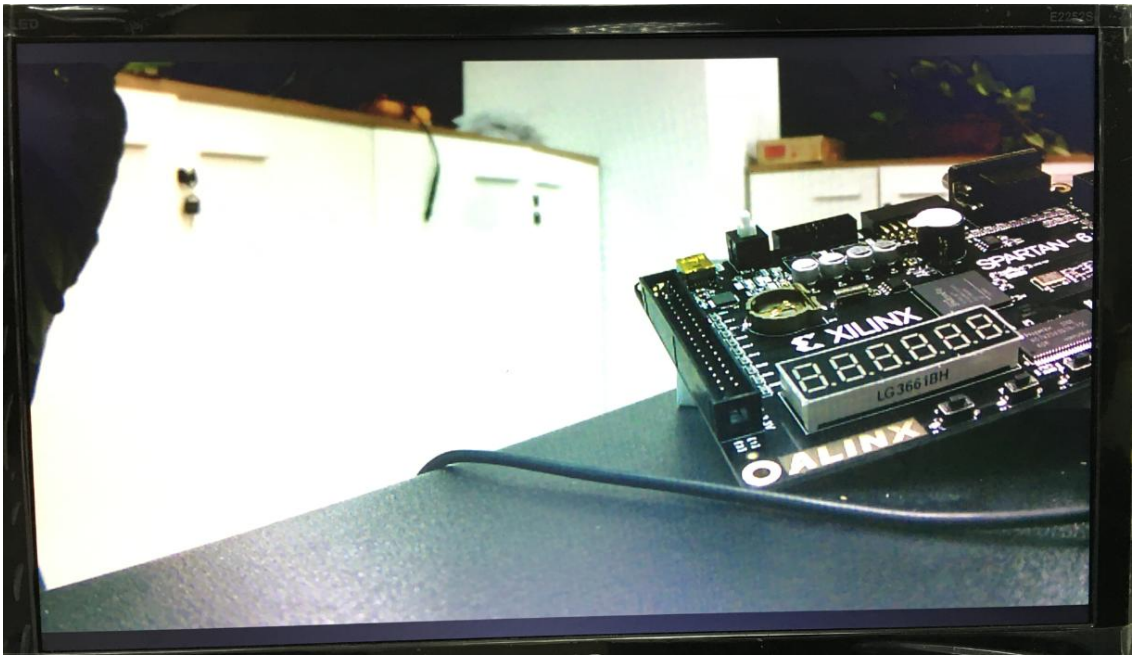


Figure 3-11: OV5642 Video image display effect (CMOS1)

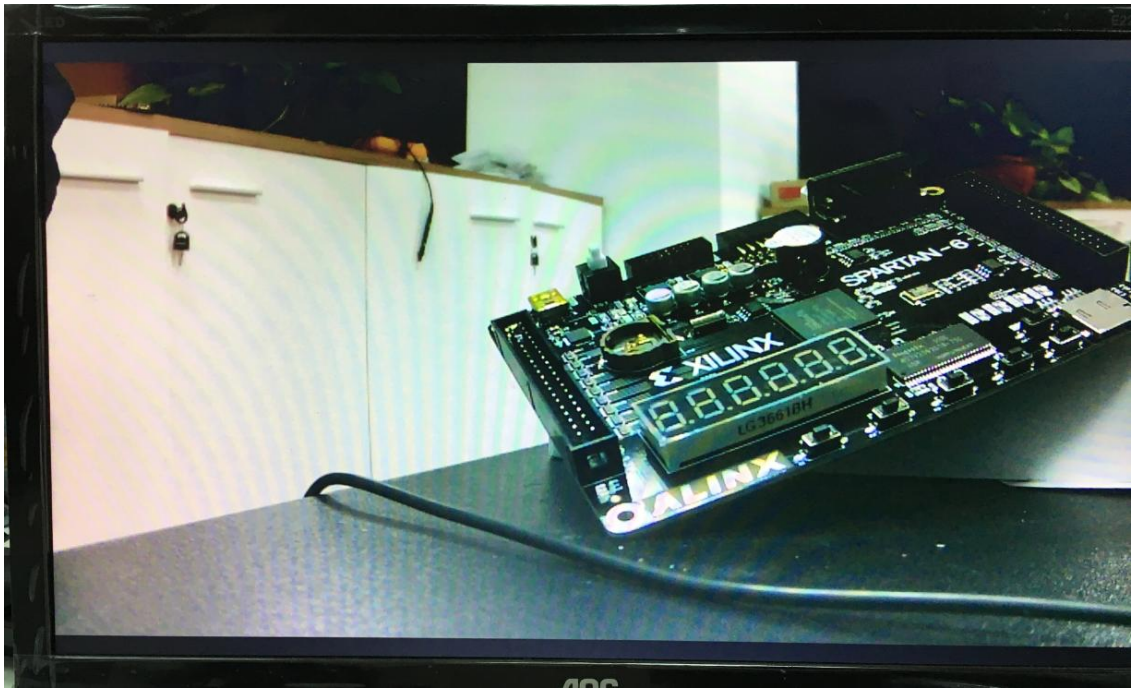


Figure 3-12: OV5642 Video image display effect (CMOS2)

If the captured video image is found to be unclear, the user can rotate the corresponding camera lens to adjust the focus.

Part 4：OV5642 binocular display simultaneously

In this experiment, the video images of the two cameras are simultaneously displayed on the VGA display, the CMOS1 camera is displayed on the left side of the VGA display, and the CMOS2 camera is displayed on the right side of the VGA display. The resolution of the VGA output is unchanged at 1280*768. The resolution of the CMOS1 camera and the CMOS2 camera output are 640*480.

Part 4.1：Programming

Because there are two cameras for simultaneous input and storage and two camera displays, we need to add another read data FIFO and write data FIFO as the video storage of another camera. In addition, we added multiple port arbitrage programs for DDR read and write in the program (ddr multiple port read arbitration program and ddr multiple port write arbitration program). The logical block diagram of the FPGA program design is shown below:

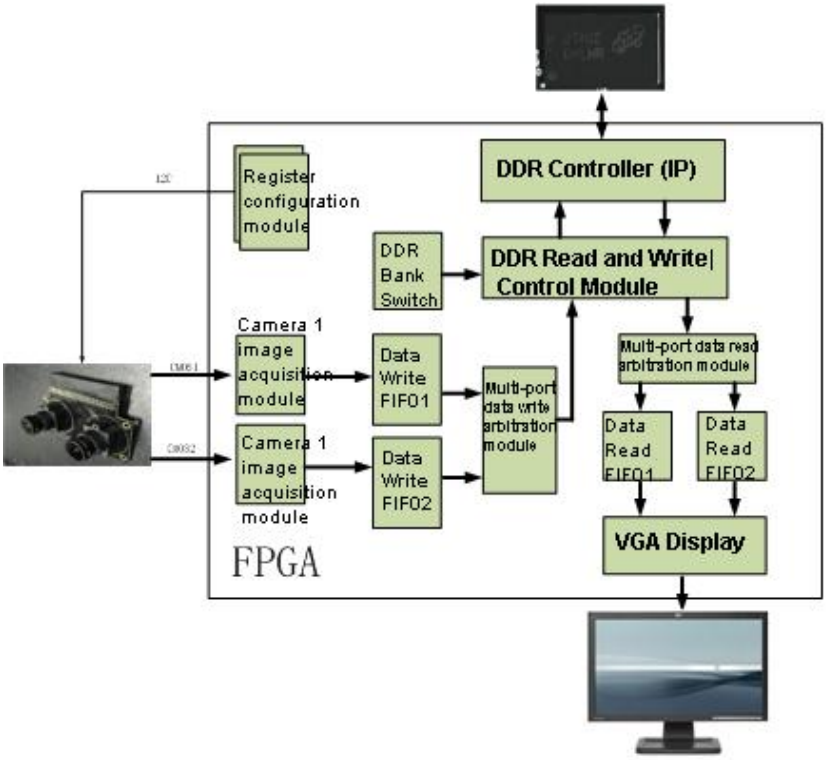


Figure 4-1: The logic block diagram of the FPGA program

The completed engineering architecture is shown in the figure below (Take AX516 as an example):

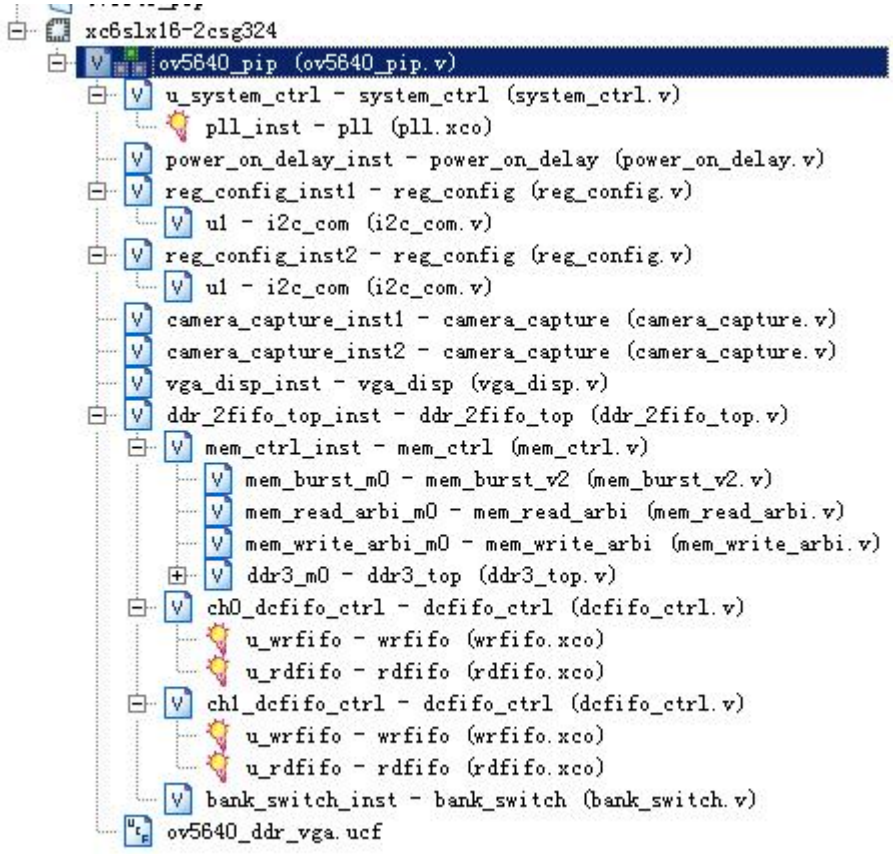


Figure 4-2: Completed Engineering Architecture of AX516

Regarding most of the programs are the same as the previous binocular switching display experiment, we will only explain the multi-port ddr read arbitration module and the multi-port ddr write arbitration module below:

1). Multi-port DDR read arbitration module: mem_read_arbi.v

In the program, the four ports of the input are polled in turn., from ch0~ch3, if ch0 has a read request and the read burst length is not 0, then enter the read state of ch0, output the burst read request of ch0; otherwise continue to query ch1, the same if ch1 has a read request and the read burst length is not 0, it enters the read state of ch1, outputs a burst read request of ch1, and sequentially queries to ch3. Because the read request for each port ch0~ch3 will

remain valid until the request is answered, there will be no read requests being lost.

2). Multi-port ddr write arbitration module: mem_write_arbi.v

The principle of the multi-port ddr write arbitration module is the same as that of the multi-port read arbitration module, and it is also a method of polling in turn.

Part 4.2: OV5642 binocular display simultaneously Experiment

After writing the program, assign the Pin of the FPGA, and after recompilation, we can start the OV5642 binocular display simultaneously experiment. The development board connects with the camera module OV5642 and the VGA port to connect the VGA monitor, and then downloads the sof file to the FPGA. We can see the video images displayed by the two cameras simultaneously on the VGA monitor. Figure 4-3 detailed the OV5642 video image binocular display effect.

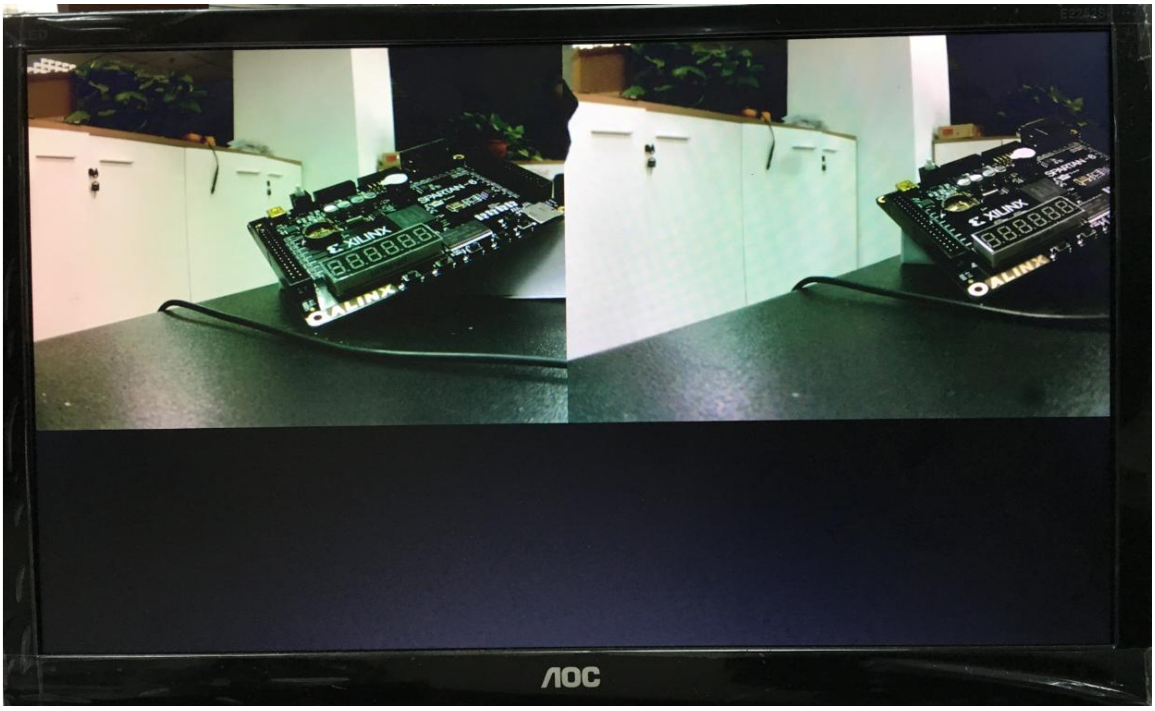


Figure 4-3: OV5642 video image binocular display effect

Users can also modify the program by themselves, so that the two cameras display different video image sizes, such as CMOS1 display 1280*720, CMOS2 display 640*480, and then display the picture-in-picture effect on the VGA display (Refer to Figure 4-1). Here will not introduce about how to achieve. That reserved for everyone to study and learn.

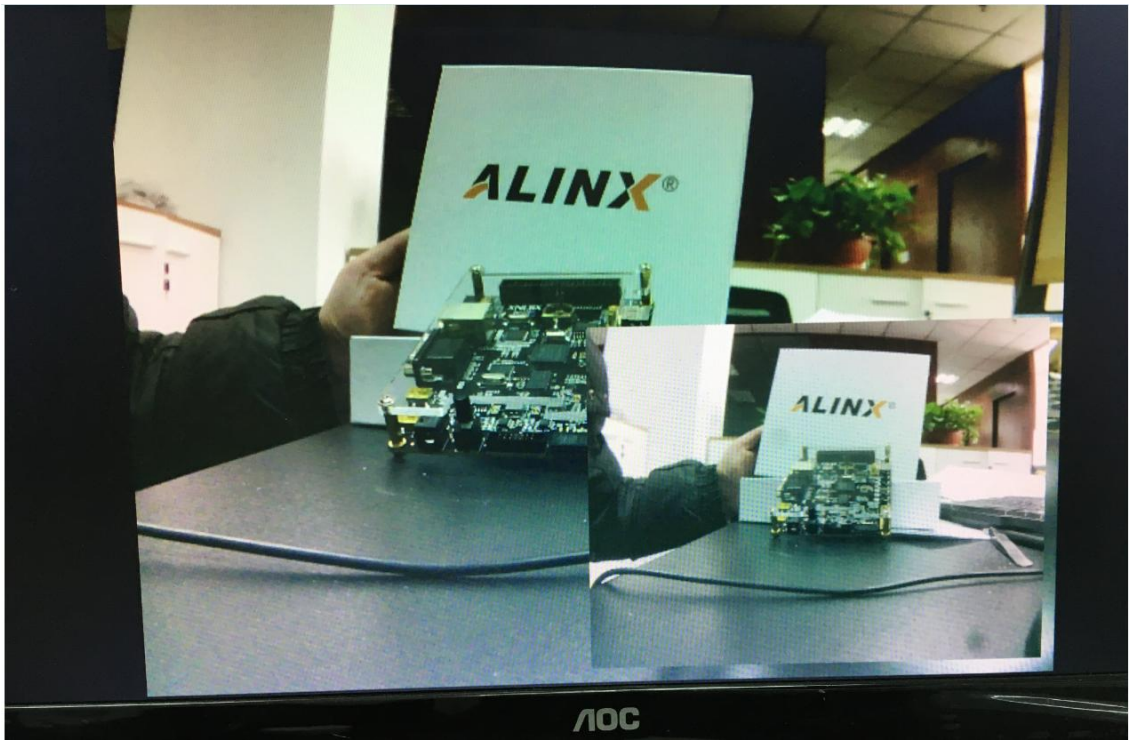


Figure 4-4: OV5642 Picture-in-picture Effect on the VGA Display