

黑金开发板之 OV5642 双目摄像头 用户手册 Rev. 1.00

版本记录

版本	时间	作者	描述
Rev1.00	2016-12-14		First Release

第一部分 OV5642 双目摄像头模组介绍

OV5642 摄像头模组采用 2 片美国 OmniVision(豪威)CMOS 芯片图像传感器 OV5640 ,支持两路独立或者同时显示的功能。CMOS OV5640 芯片支持 DVP 和 MIPI 接口, 在 OV5642 模组上是通过 DVP 接口和 FPGA 连接实现图像的传输。

1.1 OV5642 摄像头模组的参数说明

以下为 OV5642 模组的详细参数:

- 接口：40 针母座，2 路摄像头采用单独的 DVP 接口;
- 间距：2 路摄像头之间的间距为 40mm;
- 像素：硬件像素 500W;
- 感光芯片：2 片 OV5640;
- 感光尺寸：1/4;
- 模组内容：含 OV5640 电源电路和时钟;
- 功能支持：手动对焦, 自动曝光控制(AEC),自动白平衡(AWB);
- 图像格式： RAW RGB, RGB565/555/444,CCIR656, YUV422/420, YCbCr422 和压缩;
- 捕获画面：QSXGA(2592x1944), 1080p, 1280x960, VGA(640x480), QVGA(320x240) ;
- 工作温度：-30~70°C, 稳定工作温度为 0~50°C

1.2 OV5640 的上电要求

为了让 OV5640 上电后能正常工作，OV5640 的程序设计中需要考虑 OV5640 的上电时序的要求。OV5640 的上电步骤如下：

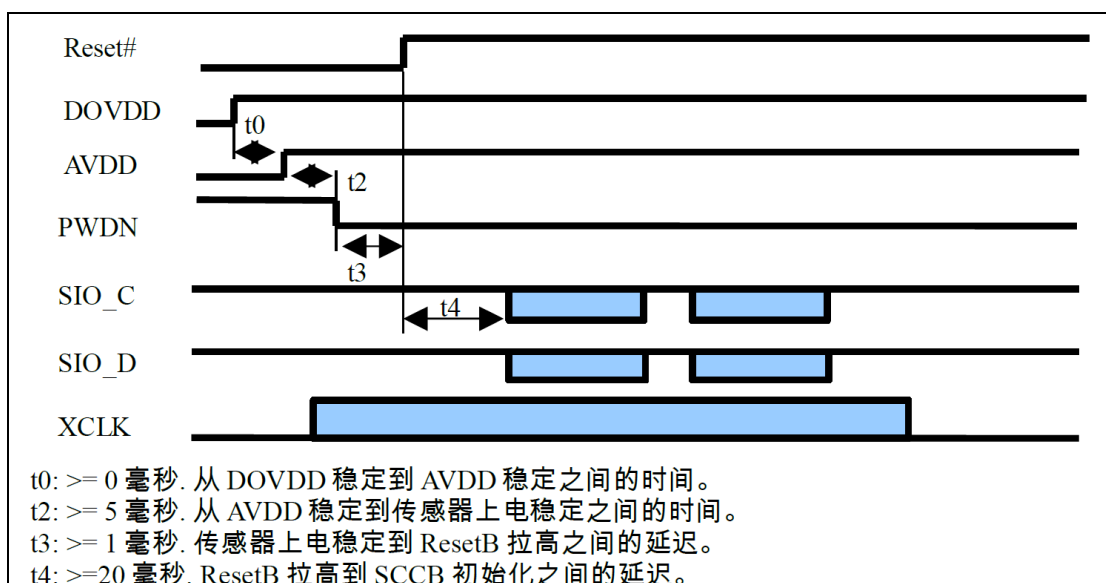
步骤 1：ResetB 拉低，复位 OV5640。 PWDN 引脚拉高。

步骤 2：DOVDD 和 AVDD 两路最好同时上电,这在模组的电源设计中实现。

步骤 3：等电源稳定 5 毫秒后，拉低 PWDN。

步骤 4：PWDN 置低 1 毫秒后, 拉高 ResetB。

步骤 5：20 毫秒后, 初始化 OV5640 的 SCCB 寄存器设置



1.3. CMOS 芯片 OV5640 的寄存器配置

OV5640 的寄存器配置是通过 FPGA (或其它 CPU) 的 I2C 接口来配置。用户需要配置正确的寄存器值让 OV5640 输出我们需要的图像格式，在我们提供的实例中我们配置 OV5640 为视频输出图像为 720P (1280x720)，帧频为 30fps 的 RGB565 的图像格式。具体的寄存器配置大家可以参考文档“OV5640_自动对焦照相模组应用指南”。

为了方便调试，用户可以配置寄存器来使能 OV5640 的内部测试图像，比如显示彩色条和彩色四方形。

Color bar 彩色条

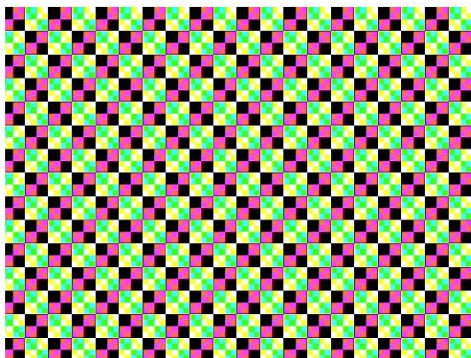
```
write_i2c(0x503d, 0x80);
write_i2c(0x4741, 0x00);
```



Color bar

Color square 彩色四方形

```
write_i2c(0x503d, 0x82);
write_i2c(0x4741, 0x0);
```



Color square

OV5640 的摄像头输出的数据格式在以下的 0x4300 的寄存器里配置，在我们的例程中 OV5640 配置成 RGB565 的输出格式。

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: YUYUYV..., or
GBRGRB...

0x1: YVUYVU..., or
GRBGRB...

0x2: UYVUYV..., or
BGRBGR...

0x3: VYUVYU..., or
RGRGRB...

0x4: UVYUVY..., or
BRBGRB...

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

Bit[3:0]: Output sequence

0x0: YUYV...

0x1: YVYU...

0x2: UYVY...

0x3: VYUY...

0x4~0xE: Not allowed

0xF: UYVY...

0x4300 FORMAT CONTROL
00 0xF8 RW

关于 OV5640 的寄存器还有很多很多，但很多寄存器用户无需去了解，寄存

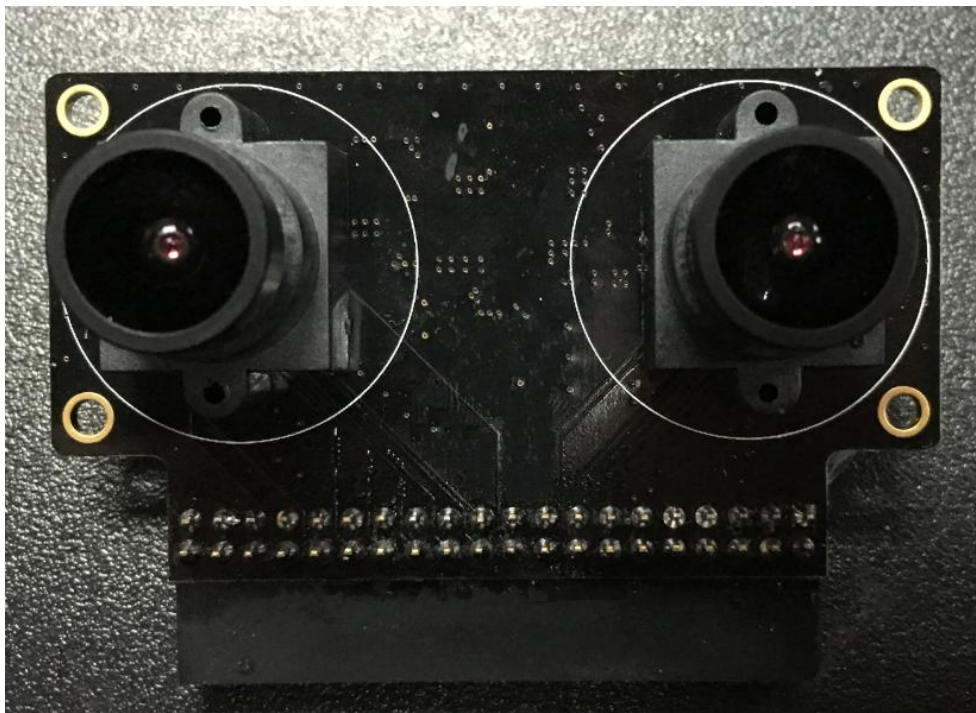
器的配置用户可以按照 OV5640 的应用指南来配置就可以了。如果您想了解更多的寄存器的信息，可以参考 OV5640 的 datasheet 中的寄存器说明。

第二部分 硬件连接

下面以黑金的 AX516 开发板为例介绍 OV5640 摄像头模组和开发板的硬件连接。OV5642 摄像头模组是 40 针的排针，可以和黑金开发板的扩展口直接无缝对插。

实物图和排针的引脚定义如下：

OV5642 模组实物图：

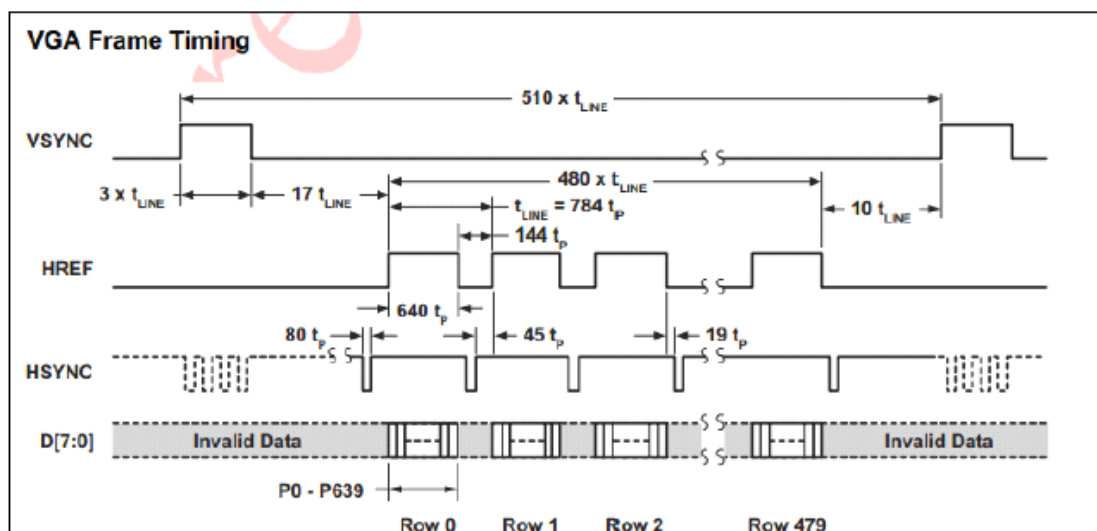


OV5642 模组接口定义：

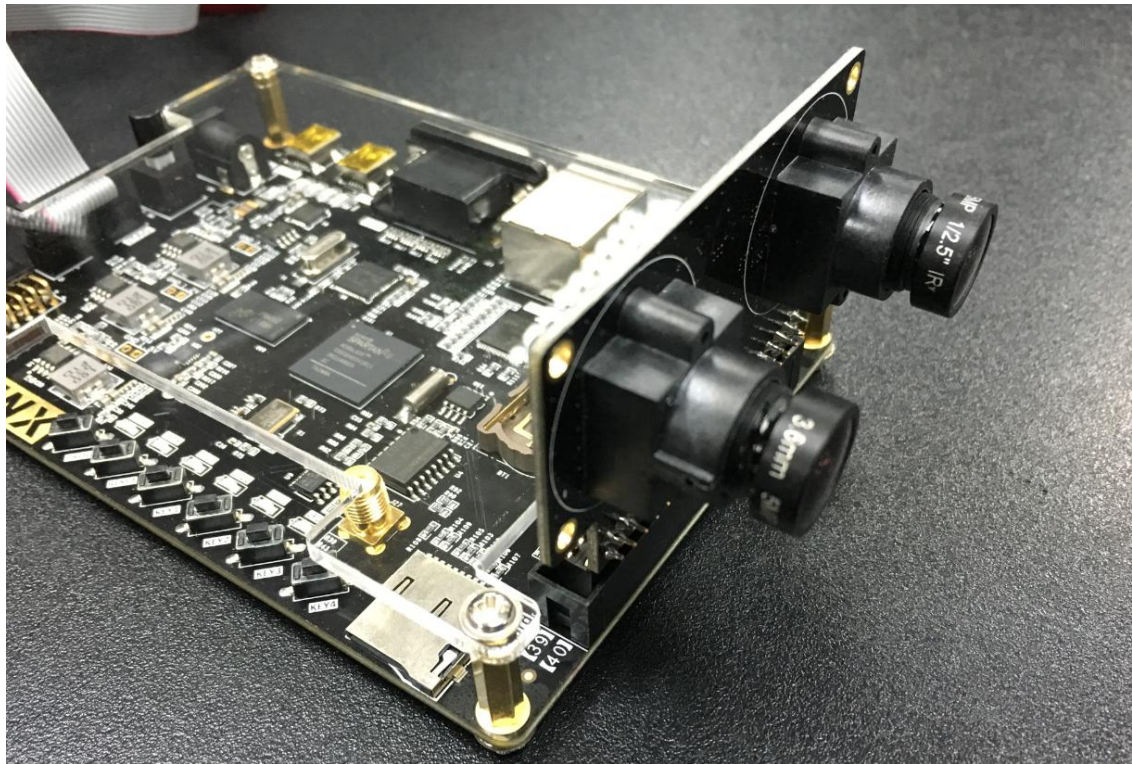
Pin 脚	信号名	Pin 脚	信号名
Pin1	地	Pin2	+5V
Pin3	CMOS2_D9	Pin4	CMOS2_SDA
Pin5	CMOS2_D6	Pin6	CMOS2_SCL
Pin7	CMOS2_D7	Pin8	CMOS_D2
Pin9	CMOS2_HREF	Pin10	CMOS2_D8
Pin11	CMOS2_D3	Pin12	CMOS2_RESET
Pin13	CMOS2_D4	Pin14	CMOS2_D5

Pin15	CMOS2_D1	Pin16	CMOS2_D0
Pin17	CMOS2_VSYNC	Pin18	CMOS2_PCLK
Pin19	空脚	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	空脚
Pin37	地	Pin38	地
Pin39	+3.3V (可不接)	Pin40	+3.3V (可不接)

其中 CMOSx_D0~D9 为摄像头 OV5640 采集的视频图像，如果输入选择 RGB 格式的话，只需要用到高 8 位 D2~D9。CMOSx_PCLK 为像素时钟信号，时钟的上升沿采集数据。CMOSx_HREF 是行有效信号，信号为高的时候，一行数据有效。CMOSx_VSYNC 是列同步信号。当配置成 VGA 模式 (640*480) 的时候，CMOS 摄像头输出的时序如下图所示：



摄像头模组和开发板连接比较简单，只要把摄像头模组直接插到开发板的扩展口上就可以了，对插的时候需要注意 PIN1 脚对齐，千万不要插偏。以下是 AN5642 模块和 AX516 开发板连接的示意图（如果是 AX515 或 AX530 开发板需要插到 J3 口）。

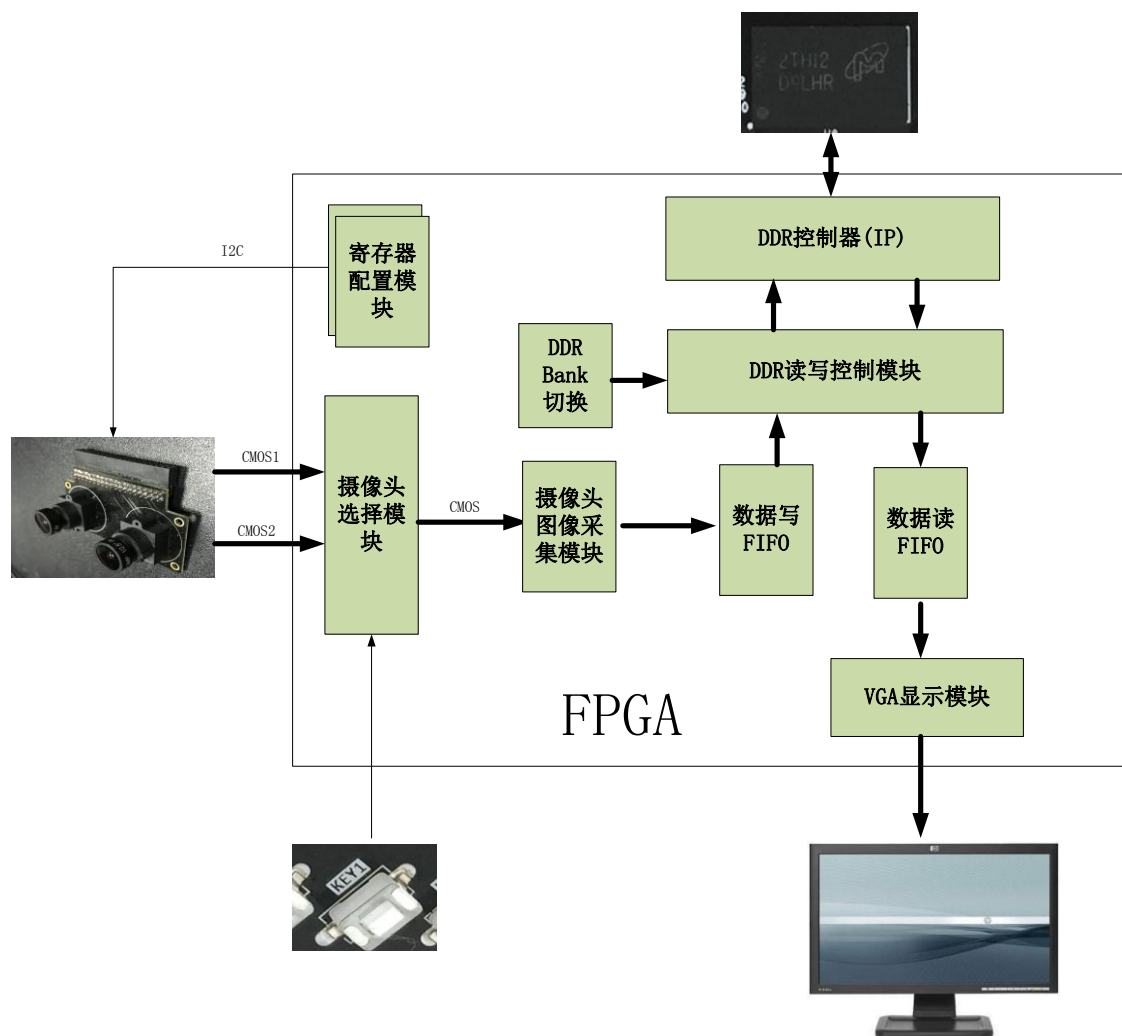


第三部分 双目切换显示实验

本实验是显示 2 路摄像头的图像到 VGA 显示器上, 2 路的视频图像是通过开发板上的按键 KEY1 来切换显示。VGA 显示器上显示的每路视频图像大小为 720P。

3.1.程序设计

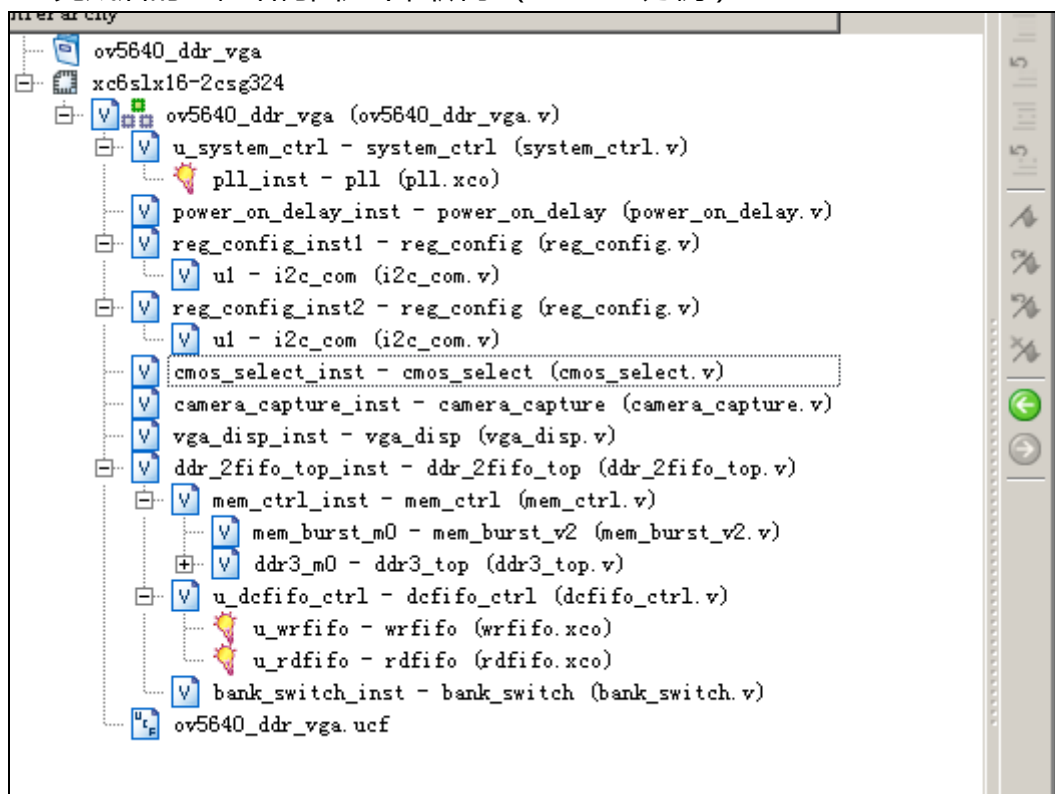
FPGA 程序设计的逻辑框图如下图所示：



上电后首先 FPGA 程序通过 I2C 总线对双目摄像头进行寄存器配置，配置 OV5640 摄像头工作在 720P 30 帧的视频输出。2 路摄像头输出的数据到 FPGA 后通过摄像头信号选择模块来选择其中一路的视频进入到摄像头数据采集模块。摄像头数据采集模块把视频图像转化成 32 位或者 64 位宽度的数据存储在写 FIFO 中，写 FIFO 里的数据达到一定的数量的时候会产生一个 burst 写信号给 DDR 读

写控制模块，DDR 读写控制模块会从写 FIFO 里读取一定长度的数据并写入到 DDR 芯片中。读 FIFO 里的数据如果少于一定的数量的时候，会产生一个 burst 读信号给 DDR 读写控制模块，DDR 读写控制模块会从 DDR 里读取一定长度的数据数据到读 FIFO 里。VGA 显示模块会读取读 FIFO 里的数据，并显示视频图像到 VGA 显示器上。

完成后的工程结构图如下图所示（AX516 为例）：



下面对每个模块的功能做一下说明：

1). 上电等待程序：power_on_delay.v

因为 OV5640 芯片有上电时序的要求，这个程序我们是 FPGA 上电后等待一段时间再使能配置 OV5640 寄存器，从而来满足 OV5640 这个时序的要求。

2). OV5640 寄存器配置模块：reg_config.v

OV5640 的寄存器配置的程序在 FPGA 启动后调用 I2C 的通信程序对两个 OV5640 芯片的寄存器的进行参数设置，这里我们设置两个 OV5640 芯片输出的图像为 RGB565 格式，像数都为 1280*720 的像数大小。

3). 摄像头信号选择模块：cmos_select.v

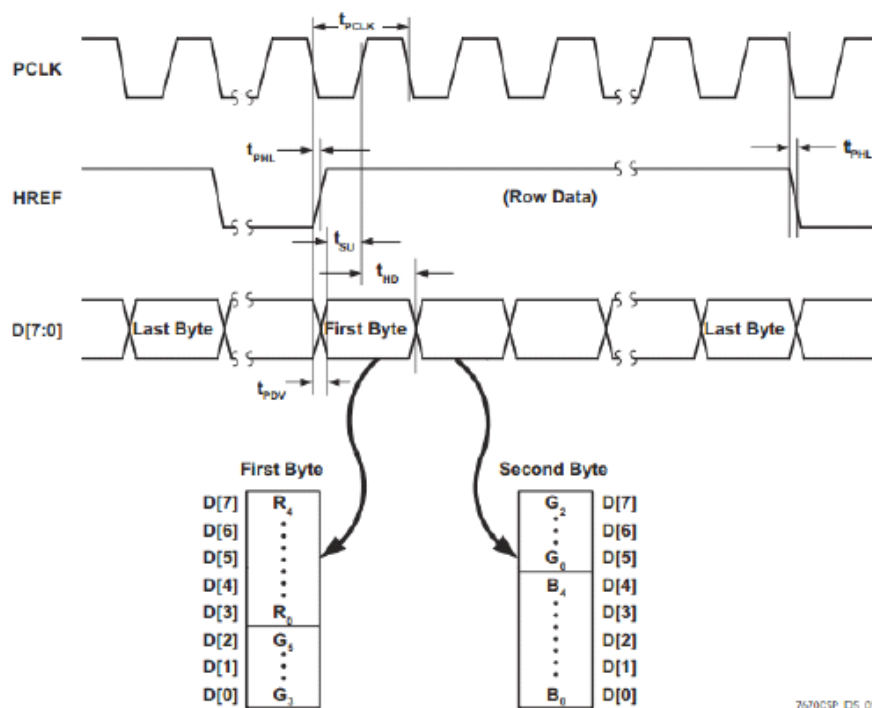
程序通过检测按键 Key1，如果按键 KEY1 按下一次有效，则切换到另一路摄

像头的信号。默认是选择采集 CMOS1 视频图像。

4). 摄像头图像采集模块 : COMS_Capture.v

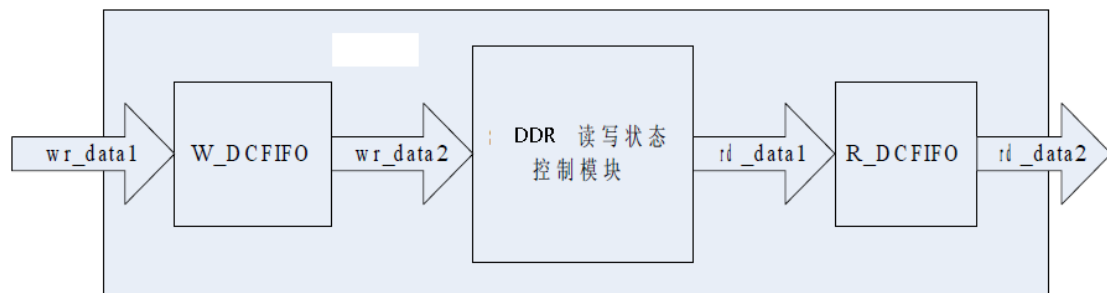
摄像头图像采集程序把从 OV5640 模组传来的 8 位图像转化为 32 位或者 64 位数据宽度，并产生 FIFO 的写信号。因为摄像头 RGB565 格式输出时，一个像数需要 16 位的数据，需要分两个 8 比特输出，这样占 2 个时钟。所以如果是 720P 的图像，一行数据需要有 1280×2 个 PCLK 时钟。

Figure 11 RGB 565 Output Timing Diagram



5). FIFO 控制程序 : dcfifo_ctrl.v

在存储采集的视频图像到 DDR 之前或者从 DDR 读出视频图像到 VGA 显示模块的时候，需要缓存数据。所以我们在 Dcfifo_ctrl.v 模块里实例化了两个 FIFO，一个数据写 FIFO，用来缓存从摄像头采集的数据；一个数据读 FIFO，用来存储从 DDR 读出的数据。本实验中，向 DDR 写入的数据首先存放在写 FIFO 中，从 DDR 中读出的数据首先存放在读 FIFO 中。



当写 FIFO 内数据量大于设定的写 burst 长度(128)的时候产生 DDR burst 写请求。

```

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

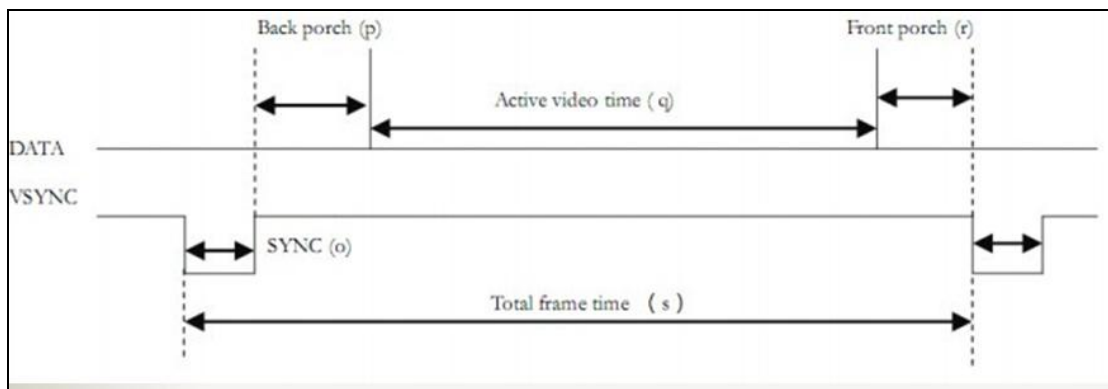
当读 FIFO 内的数据量长度小于设定的读 burst 长度(128)的时候产生 Sdram 读命令。

```

191     else if(DDR_init_done == 1'b1)
192         begin
193             //写入优先，带宽内防止数据丢失
194             if(rdf_use <= rd_length & !rd_flag) //读FIFO里的数据数量小于burst长度，读DDR开始
195                 begin
196                     //rdfifo满突发长度
197                     ddr_rd_req <= 1; //读ddr使能
198                     rd_flag <= 1;
199                 end
200             else if(DDR_rd_finish)
201                 begin
202                     //rdfifo满突发长度
203                     ddr_rd_req <= 0; //读ddr空闲
204                     rd_flag <= 0;
205                 end
206             end
207         end
208     end
209 end
  
```

6). DDR 读写控制模块:mem_ctrl.v

mem_burst_v2 程序的功能就是把外部的 burst 读请求和写请求转化成 DDR IP 接口的 Local Bus 的所需的信号和时序。使得用户不需要关心 DDR 接口的复杂时序，使得读写 DDR 变得方便简单。mem_burst_v2 模块配合 XILINX/Altera 软件生成的 DDR IP 完成了对 DDR 的控制，下图中画圈圈的端口信号为用户端的信号，通过控制这些信号，就可以读写 DDR 的数据。

VGA 场时序：

另外因为 OV5640 输出的视频图像是 1280x720 的像数大小,但 VGA 显示为 1280x768 的图像,程序中为了让视频图像现在在 VGA 显示器的中间,已经在 VGA 图像的前 24 行和后 24 行不显示图像 (默认为黑色)：

```

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
42 //-----//

```

9). ddr Bank 交换程序: bank_switch.v

Bank 交换程序实现 ddr 读和 ddr 写分别在不同的 Bank 操作,比如当 Bank0 在写入摄像头采集的图像时,VGA 读取 Bank3 的数据显示;当 Bank0 写入一幅视频图像完成后,切换到 DDR 的 Bank1 写入下一帧图像。VGA 的一帧显示完毕后,下一帧的显示就会从 DDR 的 bank0 开始读取。这样显示和存储在不同的 BANK,避免图像变形的情况。

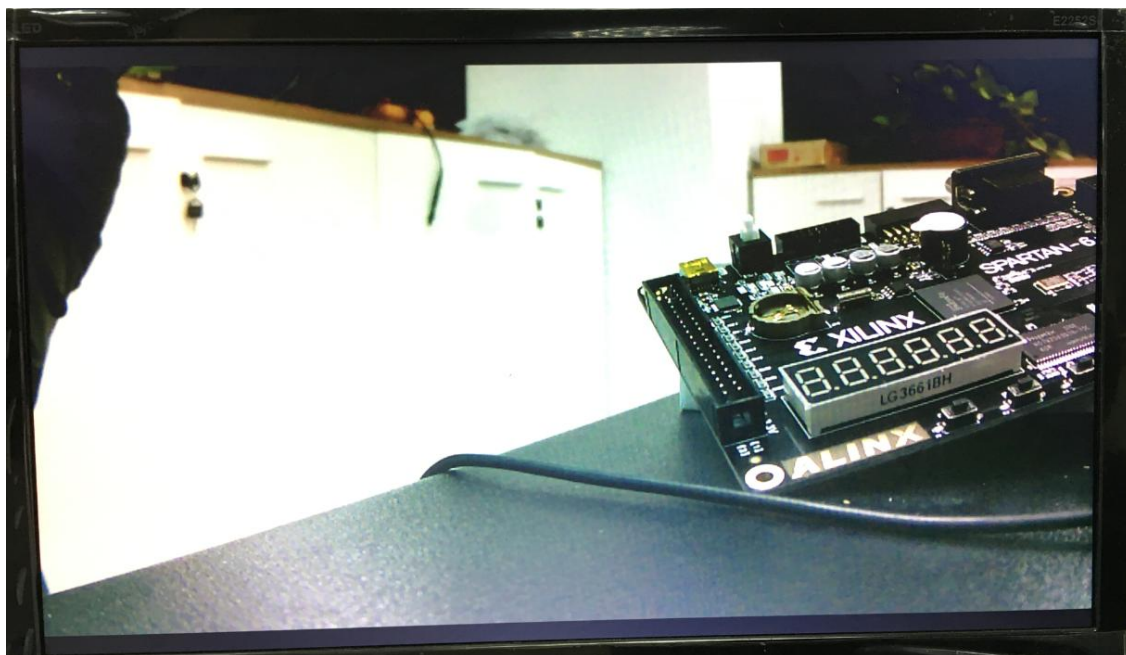
10). 系统控制程序: system_ctrl.v

产生摄像头的时钟(24Mhz)和 VGA 的像数时钟(79.5Mhz),另外程序也产生整个系统的一个复位信号。

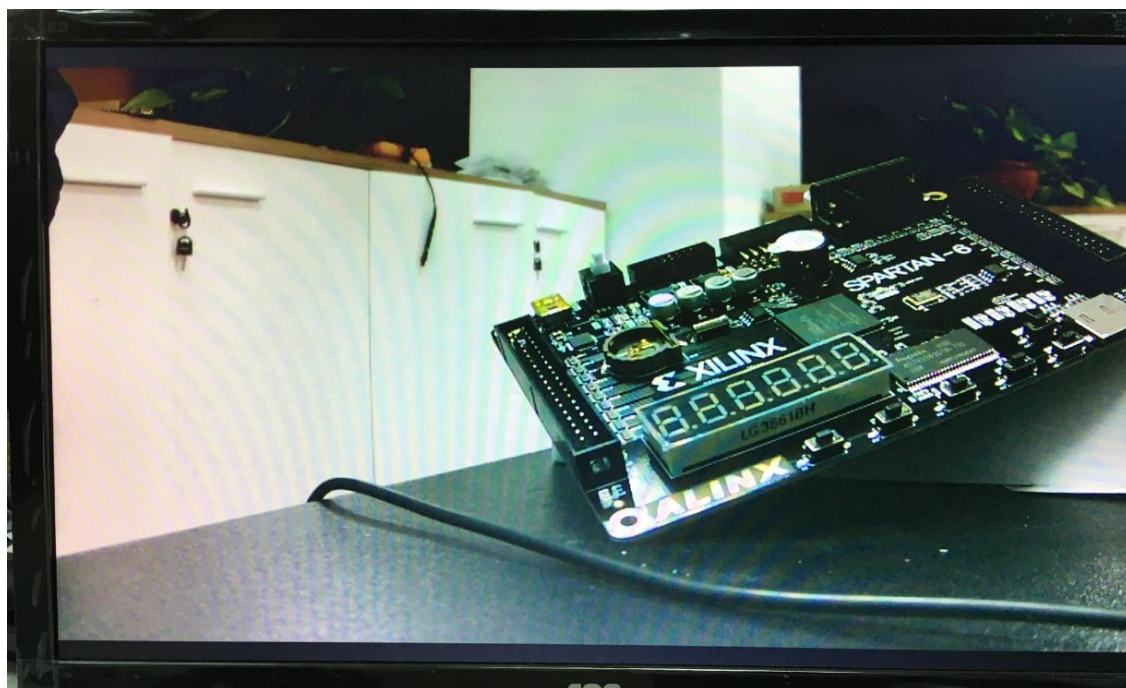
3.2. OV5642 VGA 双目显示实验

编写完程序后,分配 FPGA 的 Pin 脚,重新编译通过后我们就可以开始摄像头 VGA 显示实验了。开发板插上摄像头 OV5642 和 VGA 接口连接 VGA 显示器,再下载 sof 文件到 FPGA,我们就可以在 VGA 显示器上看到 1280x768 的视频图像了。默认 VGA 显示器上显示的是摄像头 CMOS1 的视频图像,我们只要按一下开发板按键 KEY1,VGA 显示器会显示 CMOS2 的视频图像。

OV5642 视频图像显示效果 (CMOS1):



OV5642 视频图像显示效果 (CMOS2):



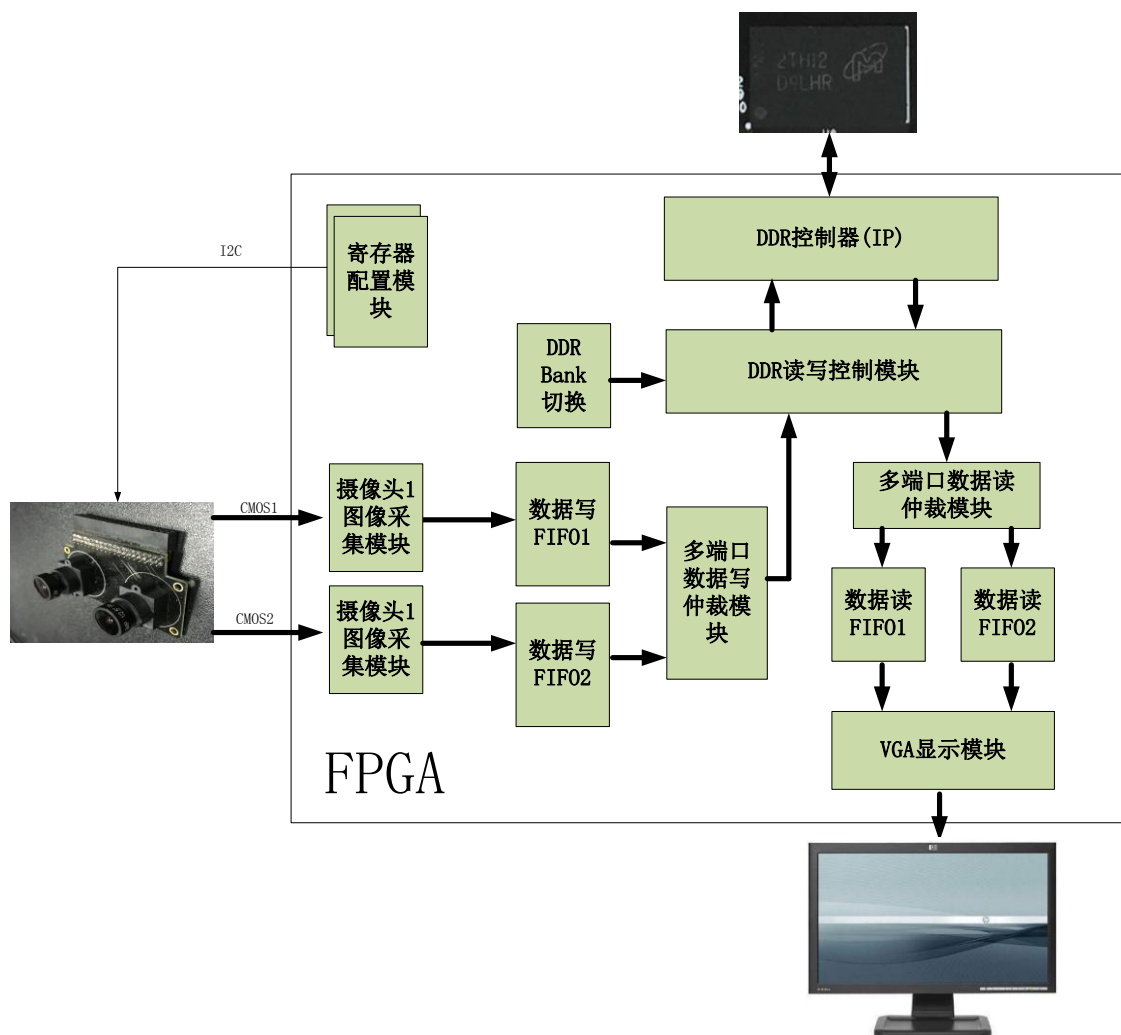
如果发现拍摄的视频图像不清晰，用户可以自己旋转相应的摄像头镜头来调焦。

第四部分 OV5642 双目同时显示实验

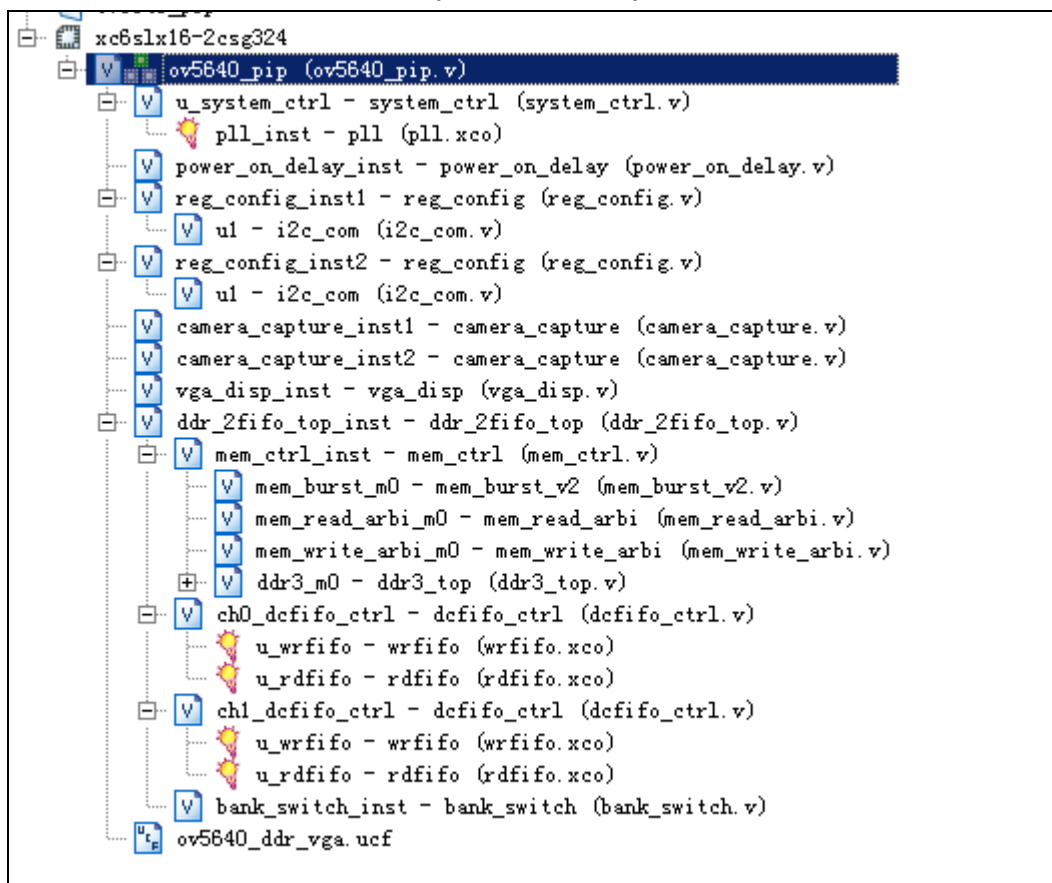
本实验将两个摄像头的视频图像同时显示在 VGA 的显示器上，CMOS1 摄像头显示在 VGA 显示器的左边，CMOS2 摄像头显示在 VGA 显示器的右边。VGA 输出的分辨率不变，还是 1280*768，这时 CMOS1 摄像头和 CMOS2 摄像头输出的分辨率都是 640*480。

4.1 程序设计

因为有 2 路摄像头的同时输入和存储和 2 个摄像头显示，程序上我们需要再添加一个读数据 FIFO 和写数据 FIFO，作为另外一路摄像头的视频存储。另外我们在程序里添加了多个端口对 DDR 读写的仲裁程序(DDR 多个端口的读仲裁程序和 ddr 多个端口的写仲裁程序。FPGA 程序设计好的逻辑框图如下图所示：



完成后的工程结构图如下图所示 (AX516 为例):



因为大部分程序跟前面的双目切换显示实验一样，我们这里只对多端口 ddr 读仲裁模块和多端口 ddr 写仲裁模块做一下说明：

1). 多端口 ddr 读仲裁模块:mem_read_arbi.v

程序中对输入四个端口进行轮流查询，从 ch0~ch3，如果 ch0 有读请求并且读 burst 长度不为 0，则进入 ch0 的读状态，输出 ch0 的 burst 读请求；否则继续查询 ch1，同样如果 ch1 有读请求并且读 burst 长度不为 0，则进入 ch1 的读状态，输出 ch1 的 burst 读请求；依次查询到 ch3。因为每个端口 ch0~ch3 的读请求会一直有效，直到请求被应答，所以不会有读请求被遗失的情况。

2). 多端口 ddr 写仲裁模块:mem_write_arbi.v

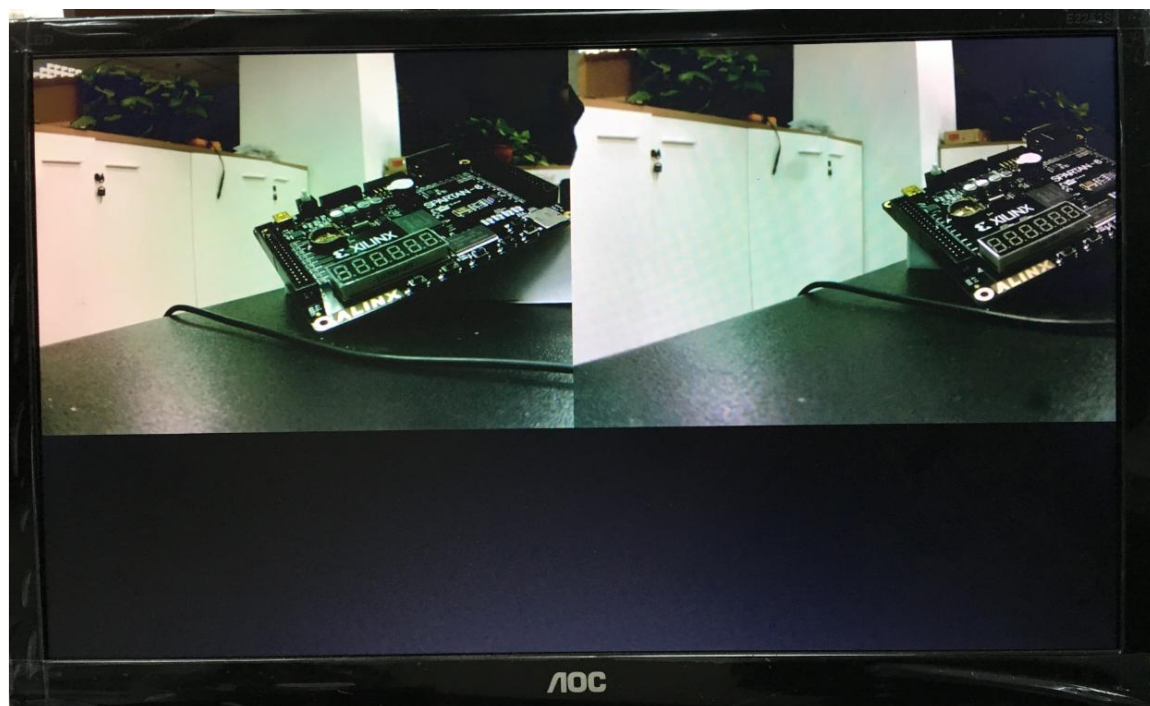
多端口 ddr 写仲裁模块的原理跟多端口读仲裁模块的方法是一样，也是采用轮流查询的方式。

4.2 双目同时显示实验

编写完程序后，分配 FPGA 的 Pin 脚，重新编译通过后我们就可以开始 OV5642 双目同时显示实验了。开发板插上摄像头 OV5642 和 VGA 口连接 VGA

显示器，再下载 sof 文件到 FPGA, 我们就可以在 VGA 显示器上看到两个摄像头同时显示的视频图像了。

OV5642 视频图像双目显示效果



用户也可以自己修改程序，让两个摄像头显示不同的视频图像大小，比如 CMOS1 显示 1280*720, CMOS2 显示 640*480，然后在 VGA 显示器上显示画中画的效果，具体如何实现我们就不讲了，还是留给大家深入研究和学习之用。

