

FPGA 综合实验 数码相框

实验报告

姓名：_____ 赵文亮 _____

学号：_____ 2016011452 _____

班级：_____ 自 64 _____

日期：_____ 2018 年 7 月 13 日 _____

目录

1	实验内容	1
1.1	必做功能	1
1.2	附加功能	1
2	设计方案	1
2.1	图片切换	1
2.1.1	图片翻页	1
2.1.2	菜单	1
2.1.3	幻灯片	1
2.2	图像处理	2
2.2.1	图片翻转	2
2.2.2	黑白与滤镜	2
2.2.3	亮度调节	2
3	系统方框图	2
4	模块选择与功能	2
5	流程图及实验代码	2
5.1	图像的显示	3
5.2	图片的切换	4
5.3	幻灯片及调速	6
5.4	图像翻转	6
5.5	灰度与滤镜	7
5.5.1	亮度调节	7
5.6	触摸控制	8
6	实验结果	8
7	实验中遇到的问题及解决方法	8
8	体会、收获与建议	9
A	数码相框操作指南	11
A.1	电源与睡眠	11
A.2	切换图片	11
A.3	幻灯片放映	11
A.4	图像翻转	11
A.5	菜单与显示	11
A.6	亮度调节	11

1 实验内容

1.1 必做功能

利用彩色 LCD 设计一个简易数码相框，显示图片并能够翻页。

1.2 附加功能

我借助提高实验中用到的红外遥控器，设计了一个多功能的数码相框。它具有显示图片、切换图片、幻灯片播放及调速、图片翻转、亮度调节、黑白及滤镜模式等功能。更加具体的功能及使用方法请参照附录 A 中的数码相框操作指南。

2 设计方案

我使用 FLASH 来完成图片的储存，图片显示的过程实际上是将图片从 FLASH 中读取到 SDRAM，再从 SDRAM 中写入到 LTM 显示屏上。我将数码相框的所有功能划分为两大类：图片的切换和图像的处理。其中，图片的切换将翻页、幻灯片播放、进入菜单等功能包含在内；图片的处理将黑白、滤镜、亮度调节、图片翻转等功能包含在内。更加深入地讲，图片切换决定了下一时刻应该显示的照片序号，图像处理则决定了将每一个像素点从 FLASH 中写入 SDRAM 中所需要进行的操作。

2.1 图片切换

图片切换部分由 photo_manager 模块来完成。该模块可以根据当前的按键，给出下一时刻应该显示的照片的序号，以及显示模式（标准、黑白、滤镜）。

2.1.1 图片翻页

接收到上一张、下一张或者数字键时，将照片序号修改为对应的数值。

2.1.2 菜单

我自己制作了一张图片作为菜单，一并写入到 FLASH 中，正常情况下不会显示出来，只有按下 MENU 键才会进入菜单。我使用一个变量来保存当前的显示模式。在菜单里，用户通过功能键（A、B、C）或触屏就可以设置显示模式。

2.1.3 幻灯片

在幻灯片模式下，我设置了一个计数器，这个计数器计满的时候就会给出一个使能信号，从而切换到下一张。这个计数器的计数值是可变的，所以可以实现调速。

2.2 图像处理

图像处理部分由 `flash_to_sdram_controller` 模块来完成。该模块可根据当前的模式和按键，给出读取 FLASH 中每个像素以及写入到 SDRAM 中的方式。

2.2.1 图片翻转

正常的图片显示是现将指针指向 FLASH 中某一张图片的起始地址，在不断递增指针，使之扫过该图片的所有像素。因此，实现图片翻转只需要将这一过程反过来，即先将指针指向 FLASH 中某一张图片的最后一个像素的地址，再不断递减指针。这里要注意，由于 FLASH 中图片是以 RGB 的顺序保存的，如果仅仅从后向前读取会导致顺序交换。因此在最后写入 SDRAM 时候要注意 RGB 的顺序。

2.2.2 黑白与滤镜

黑白效果即为将 RGB 彩色图转为灰度图。我采用了式 (1) 的算法^[1]。

$$Grey = (R \times 38 + G \times 75 + B \times 15) >> 7 \quad (1)$$

滤镜则是通过将 RGB 的值进行轮换来实现的，既保留了三个分量之间的对比，由营造出奇妙的视觉效果。

2.2.3 亮度调节

我将读取出来的 RGB 值都乘以一个小于等于 1 的系数，再将结果写入 SDRAM 中，即可达到调节亮度的功能。我这里设置了 9 级亮度可供调节。

3 系统方框图

数码相框系统的方框图如图 1 所示。其中 Core 方框中的表示通过 Verilog 编写的模块，在芯片中实现；Board 方框中的模块为开发板上现有的模块；外接的 LTM 即为显示屏。

4 模块选择与功能

数码相框中涉及到的模块已经在图 1 中显示清楚。由于这些模块都是开发板配套的器件或是 Verilog 编写的电路，故不涉及参数选择的问题，仅将它们的功能列举出来，如表 1 所示。

5 流程图及实验代码

数码相框的流程图如图 2 所示，由于内部实现细节过于复杂，这里只将主要的流程画出，下面将分析主要部分的代码实现¹。

¹完整代码见<https://github.com/thu-jw/PhotoFrame>，借鉴了例程 EPHOTO

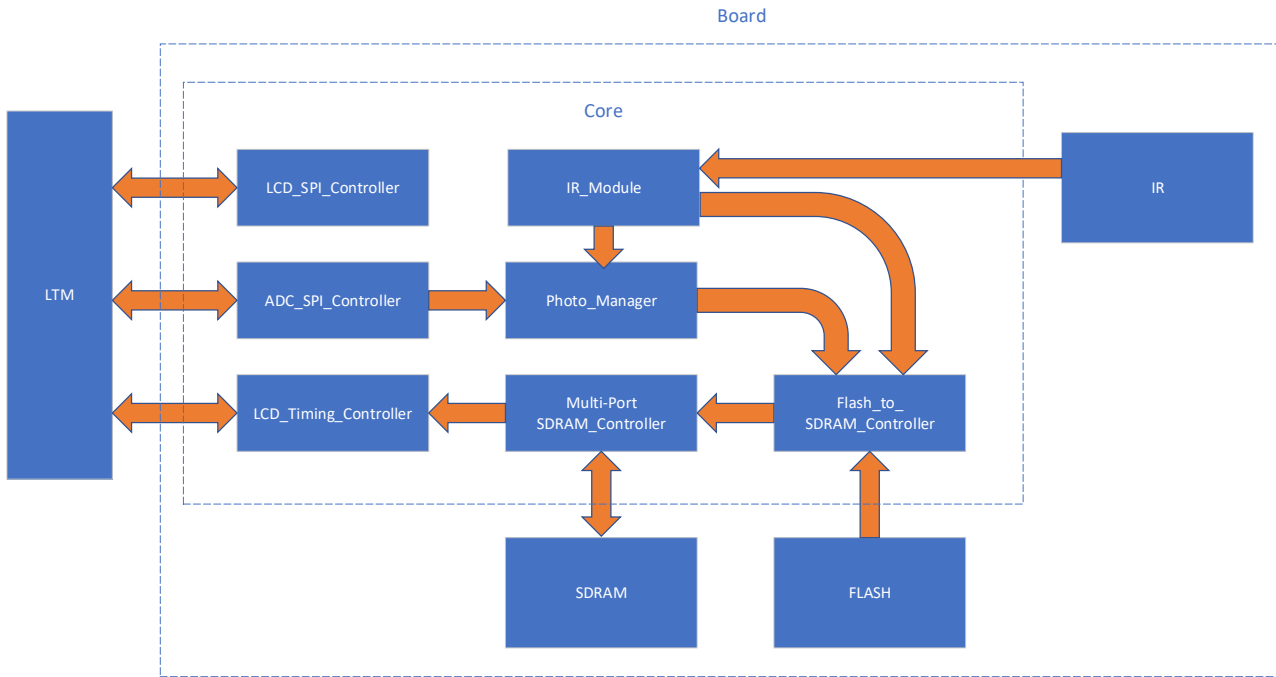


图 1: 数码相框系统方框图

表 1: 数码相框各模块功能

名称	功能
LTM	彩色 LCD 显示屏，带触摸功能
SDRAM	缓存从 FLASH 中读取的图像数据
FLASH	长期保存图片数据
IR	接收红外遥控信号
LCD_SPI_Controller	配置 LCD 驱动
ADC_SPI_Controller	获取屏幕的触摸数据
LCD_Timing_Controller	控制图片数据写入 LTM 的时序
IR_Module	将红外输入从串行转为并行
Photo_Manager	管理当前图片序号和模式
Multi-Port SDRAM_Controller	与 SDRAM 的接口电路
Flash_to_SDRAM_Controller	从 FLASH 中读取像素并处理，控制写入 SDRAM 的顺序

5.1 图像的显示

从 FLASH 中读取每一个像素，每次读取完地址加 1。

```

1         if (!direction)
2             begin
3                 if ( flash_addr_o < flash_addr_max )

```

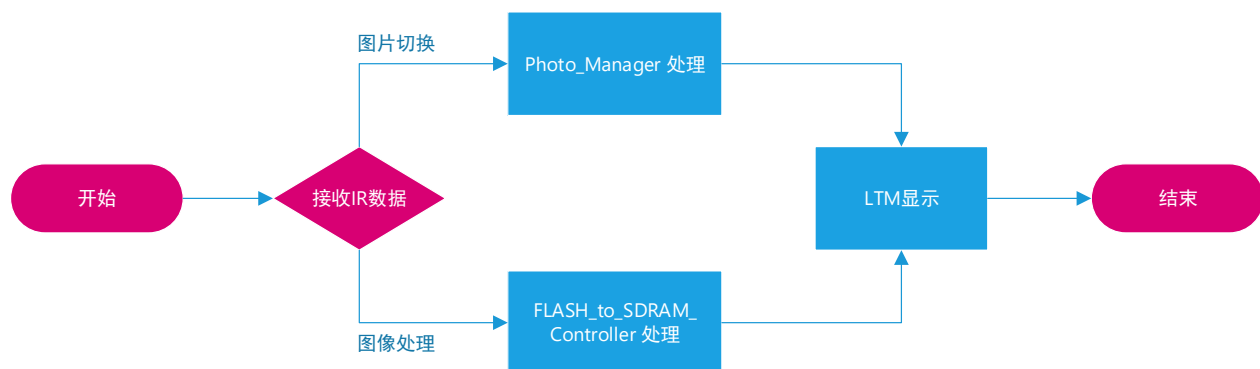


图 2: 数码相框流程图

```

4          flash_addr_o <= flash_addr_o + 1;
5          end

```

确保 RGB 同步之后，写入到 SDRAM 中：

```

1          if (!direction)
2              begin
3                  oRED    <= fl_dq_delay1;
4                  oGREEN  <= fl_dq_delay2;
5                  oBLUE   <= fl_dq_delay3;
6              end

```

之后从 SDRAM 到 LTM 的过程通过 FIFO 缓存，并使用 LCD_Timing_Controller 模块严格控制时序，过程较为复杂，故不再赘述。

5.2 图片的切换

图片切换的过程比较复杂，包含多种情况，具体代码如下：

```

1  //all functions about photo_change
2  always@(posedge iCLK or negedge iRST_n)
3      begin
4          if (!iRST_n)
5              photo_cnt <= 0;
6          else
7              begin
8                  begin
9                      if (!menu && (nextpic_set || (autoplay_next && autoplay)))
10                         begin
11                             if(photo_cnt == (PHOTO_NUM-1))
12                                 photo_cnt <= 0;
13                             else
14                                 photo_cnt <= photo_cnt + 1;
15                         end

```

```

16     else if (!menu && prepic_set)
17         begin
18             if(photo_cnt == 0)
19                 photo_cnt <= (PHOTO_NUM-1);
20             else
21                 photo_cnt <= photo_cnt - 1;
22         end
23     else if (a_set)
24         oMODE <= 0;
25     else if (b_set)
26         oMODE <= 1;
27     else if (c_set)
28         oMODE <= 2;
29     else if (new_key)
30         begin
31             if ((iKEY <= PHOTO_NUM) && (iKEY >= 1))
32                 begin
33                     photo_cnt <= iKEY - 1;
34                     menu <= 0;
35                 end
36             else
37                 case (iKEY)
38                     8'h17:
39                     begin
40                         photo_cnt <= 0;
41                         menu <= 0;
42                     end
43                     8'h12:
44                     begin
45                         oSLEEP <= ~oSLEEP;
46                         photo_cnt <= 0;
47                         menu <= 0;
48                     end
49                     8'h0c:
50                         oSLEEP <= ~oSLEEP;
51                     8'h11:
52                     begin
53                         menu <= ~menu;
54                         if (!menu)
55                             begin
56                                 photo_cnt <= PHOTO_NUM;
57                                 last_photo_cnt <= photo_cnt;
58                             end
59                         else
60                             photo_cnt <= last_photo_cnt;
61                     end
62                     8'h0f:

```

```

63             oMODE <= menu ? 0 : oMODE;
64         8'h13:
65             oMODE <= menu ? 1 : oMODE;
66         8'h10:
67             oMODE <= menu ? 2 : oMODE;
68         endcase
69     end
70 end
71 end
72 end

```

其中 **nextpic_set** 表示通过触屏或者下一页产生的翻页信号，**autoplay_next** 表示幻灯片模式下的翻页信号。这个过程中同样处理了与菜单相关的所有操作。在菜单模式下，所有的手动翻页均失效，只有通过再次按下菜单才能退出。**last_photo_cnt** 记录了进入菜单之前浏览的位置。整个过程将图片切换的相关操作全部处理完毕。

5.3 幻灯片及调速

幻灯片模式通过计数器来完成自动播放：

```

1  always@(posedge iCLK or negedge iRST_n)
2      begin
3          if (!iRST_n)
4              begin
5                  autoplay_cnt <= 0;
6                  autoplay_next <= 0;
7              end
8          else if (autoplay_cnt < autoplay_cmp)
9              begin
10                 autoplay_cnt <= autoplay_cnt + 1;
11                 autoplay_next <= 0;
12             end
13          else
14              begin
15                 autoplay_cnt <= 26'b0;
16                 autoplay_next <= 1;
17             end
18          end

```

其中 **autoplay_cmp** 为可设定的比较值，调节这个值即可改变幻灯片的换页速率。

5.4 图像翻转

图像翻转首先通过倒序读取 FLASH 中的像素：

```

1  assign flash_addr_max_n = 54 + 3*DISP_MODE * (iPHOTO_NUM+1) ; //54(bmp file header)+ 3 x 800x480 (3 800x480)
2  assign flash_addr_min_n = 54 + 3*DISP_MODE * d2_photo_num;
3  ...

```



```

4         flash_addr_o <= flash_addr_max_n ;
5         ...
6     begin
7         if ( flash_addr_o > flash_addr_min_n )
8             flash_addr_o <= flash_addr_o - 1;
9     end

```

其中 flash_addr_max_n 表示反向读取的地址最大值。接着 RGB 按照正确的顺序写入 SDRAM:

```

1     begin
2         oBLUE <= fl_dq_delay3;
3         oRED <= fl_dq_delay2;
4         oGREEN <= fl_dq_delay1;
5     end

```

这里注意 RGB 的值不仅仅是反向，而是先反向后移位。

5.5 灰度与滤镜

灰度使用式 (1) 的算法:

```

1 assign grey1 = (fl_dq_delay1 * 38 + fl_dq_delay2 * 75 + fl_dq_delay3 * 15) >> 7;

```

滤镜则通过交换 RGB 的顺序获得:

```

1     begin
2         oGREEN <= fl_dq_delay1;
3         oBLUE <= fl_dq_delay2;
4         oRED <= fl_dq_delay3;
5     end

```

5.5.1 亮度调节

通过对输入的数据乘以一个小于等于 1 系数来实现。实际操作时，对数据乘以一个小于等于 8 的数，再右移 3 位:

```

1 wire [11:0] fl_dq_buf;
2 assign fl_dq_buf[7:0] = FL_DQ;
3 wire [11:0] fl_dq_adjust = ((intensity[3] == 1) ? (fl_dq_buf << 3) : 0) +
4                             ((intensity[2] == 1) ? (fl_dq_buf << 2) : 0) +
5                             ((intensity[1] == 1) ? (fl_dq_buf << 1) : 0) +
6                             ((intensity[0] == 1) ? (fl_dq_buf) : 0);
7
8 ...
9         fl_dq_delay1 <= fl_dq_adjust[10:3]; //(fl_dq_buf * intensity) >> 3;

```

这里通过移位和加法运算对乘法进行了优化。

5.6 触摸控制

ADC_SPI_Controller 将坐标信息传递给 Photo_Manager，于是可以通过判断坐标的大小来实现触摸效果。通过测量可以确定下面两组参数：

```
1 parameter PHOTO_NUM = 6; // total photo numbers
2 parameter NEXT_PIC_XBD1 = 12'h0;
3 parameter NEXT_PIC_XBD2 = 12'h300;
4 parameter NEXT_PIC_YBD1 = 12'he00;
5 parameter NEXT_PIC_YBD2 = 12'hfff;
6 parameter PRE_PIC_XBD1 = 12'hd00;
7 parameter PRE_PIC_XBD2 = 12'hfff;
8 parameter PRE_PIC_YBD1 = 12'h000;
9 parameter PRE_PIC_YBD2 = 12'h200;
10
11 parameter MODE_XBD1 = 12'h600;
12 parameter MODE_XBD2 = 12'hb00;
13 parameter A_YBD1 = 12'ha50;
14 parameter A_YBD2 = 12'he50;
15 parameter B_YBD1 = 12'h600;
16 parameter B_YBD2 = 12'ha50;
17 parameter C_YBD1 = 12'h200;
18 parameter C_YBD2 = 12'h600;
```

其中第一组参数为上下翻页的触摸区域，下面一组参数是模式切换的触摸区域。再通过设计使能信号，即可完成触摸控制。以标准模式按键为例：

```
1 assign A_en = ((iX_COORD > MODE_XBD1) && (iX_COORD < MODE_XBD2) &&
2               (iY_COORD > A_YBD1) && (iY_COORD < A_YBD2))
3               ? 1:0;
```

6 实验结果

我的数码相框实验得到了老师和助教的一致好评，老师也主动提出要给我的作品录制视频，我想这就是对我最大的肯定了。我在视频中截取了一张图片，如图 3 所示。此时我正在演示滤镜模式，原本黄色的小猫变成偏绿色。

7 实验中遇到的问题及解决方法

本次实验中遇到的问题比较多，下面详细说明：

1. SD 卡读取失败。一开始我尝试使用 SD 卡存储图片的方法，然而运行官方提供的例程都不能读取 SD 卡。后来与同学交流发现，他们也没有成功实现 SD 卡的读取。于是我改用 FLASH 存储图片。
2. 挂载 NIOS II 系统失败。为了能够更好地操纵图片中的像素，我最初想要把整个系统构建基于 NIOS II 的系统，这样就可以在 C 程序里面实现更多的图片处理的算法。我写好了所有模块的接口电路，在



图 3: 数码相框展示截图

挂载到 NIOS II 上却发现, SDRAM_Controller 是一个必要的模块, 而只有它才能用来访问 SDRAM, 于是使用 SDRAM 作为中转的做法就不合适了。于是我又想到, 能不能直接把 FLASH 中的数据读到总线上, 在 C 程序中操纵后再一个一个写入到 LTM 中呢? 我又按照这个想法尝试了一下, 最后也成功挂载在 NIOS II 系统上, 然而在 Eclipse 中编程时, 一个 Hello 程序都不能下载。我明白这应该是硬件的问题。最后, 我还是采用了仅用 FPGA 实现的方式, 顺利地完成了我最初的设想功能。

8 体会、收获与建议

最开始我选择综合实验时, 我就希望能够选择一个有趣的, 有挑战性的实验。我一开始选择了手写数字识别, 但是当我看完实验指导后, 我发现这个实验无非是先训练好一组参数, 再在 FPGA 中进行一次矩阵乘法就可以完成, 对这个实验有些失落。于是我改为了数码相框。我向助教提出后, 两位助教提前给我打了预防针, 说这个实验比较难, 往年有同学做了很多天也没有成功, 最后改做了别的实验。

由于我提高实验做得比较早, 我决定多花一些时间来钻研这个实验。当时仅仅是周二的下午, 我认为我有足够的时间完成数码相框的设计。之后的几天, 我就一直与这个相框为伴。由于自己电脑比较慢, 我在实验室的时候就使用台式机, 回寝室之后就用天空工场的服务器, 一直沉浸在这个实验中。开始几天试图挂载在 NIOS 系统上, 却一次次失败, 也给了我不少的打击。后来和叶老师交流, 老师说不一定在 NIOS 上完成, 于是我打算改变思路, 尝试完全基于 FPGA 完成实验。由于我在前几天试图挂载 NIOS 系统的挣扎中对文档和例程已经相当熟悉, 对实现哪个功能需要修改哪些地方已经了如指掌, 在周五, 我利用了不到一天的时间, 就完成了全部功能的编写。当然, 没有前几天的积累, 这也是不太可能的。

我的数码相框得到了老师、助教、同学的一致认可, 我也感到很有成就感, 这个实验将会成为我这个小学期最有意义的成果。非常感谢老师和助教在我遇到困难的时候帮助我调整思路, 让我最终获得了成功。

一个小建议: 数码相框和 FPGA 通过排线连接时, 排线恰好扭转了 180 度。考虑到排线应该是标准的规格, 这可能是 Altera 公司设计 LTM 接口时没有考虑周全。事实上将 LTM 上的接口旋转 180 度后, 再与

FPGA 连接后效果会比较好。为此，可以尝试制作一个转接头，可能会方便后来的同学进行实验。

参考文献

- [1] 从 RGB 色转为灰度色算法 - CSDN 博客<https://blog.csdn.net/xdrt81y/article/details/8289963>

A 数码相框操作指南

A.1 电源与睡眠

- POWER 键控制数码相框的开机和关机。使用 POWER 键关机后再开机，始终会回到第一张照片。
- MUTE 键控制数码相框的睡眠与唤醒。MUTE 键睡眠后再唤醒，会回到上一次浏览的位置。

A.2 切换图片

- 按键翻页：使用 ADJUST 左右键翻页。
- 触屏翻页：点击屏幕的左上角或右下角翻页。
- 自由跳转：按下数字键即可显示对应的图片。

A.3 幻灯片放映

- 播放/停止键控制幻灯片的播放和停止。
- 在幻灯片模式下，ADJUST 左右控制幻灯片播放速度。

A.4 图像翻转

- CHANNEL 键控制图片的上下方向。CHANNEL UP 使图片朝上，CHANNEL DOWN 使图片朝下。

A.5 菜单与显示

- 通过 MENU 键进入菜单，即可选择显示模式：
 - 标准模式
 - 黑白模式
 - 滤镜模式
- 在菜单中，可以通过功能键（A、B、C）选择显示模式，也可以直接通过触屏选择对应模式。
- 菜单模式下，翻页键被禁用。再次按下菜单键可以回到上次浏览的图片位置。
- 任何情况下按下 RETURN 都会回到第一张图片。

A.6 亮度调节

- 音量键上下调节屏幕的亮度