# 1 Experiment Introduction

In the previous experiment, we practiced SD card reading and writing, VGA video display and other routines. In this experiment, the BMP pictures in the SD card are read out and written to the external memory, and then displayed through HDMI, LCD, etc. If the experiment is displayed on the LCD screen module, the LCD module is required.

# 2 Experiment Principle

In the previous experiment, we displayed the color bars on the VGA and LCD, which are the data generated inside the FPGA. In this experiment, the color bars are replaced with the BMP picture data in the SD. However, the SD card reading speed is far from meeting the display speed requirement. It can only be written to the external high-speed RAM first, and then read to the video timing module.

## 2.1 BMP image format

This experiment searches BMP files directly on the SD card. Assume that each file starts from the first byte of a certain sector of SD and finds BMP according to the characteristics of the BMP file header.

BMP (full name Bitmap) is a standard image file format in the Windows operating system and can be divided into two categories: device-dependent bitmap (DDB) and device-independent bitmap (DIB), which are widely used. It uses a bitmapped storage format that does not use any compression other than the image depth option, so the BMP file takes up a lot of space. The image depth of BMP files can be 1bit, 4bit, 8bit and 24bit. When the BMP file stored data, the image is scanned in the order from left to right and bottom to top. Because the BMP file format is a standard for exchanging graph-related data in a Windows environment, graphical image software running in a Windows environment supports the BMP image format.

The most important thing for programming is to find the BMP header. The BMP image header format is as follows:

1) 1-2: (The number here represents the byte, the same below) image file header. “0x4d42=’BM’”, indicating that it is a BMP format supported by Windows. (Note: Check ascii table B 0x42, M 0x4d, “bfType” is two bytes, B is low byte, M is high byte, so “bfType=0x4D42” instead of “0x424D”, please note)

2) 3-6： Entire file size, 4690 0000， is 00009046h=36934

3) 7-8: Reserved, must be set to 0

4) 9-10: Reserved, must be set to 0

5) 11-14: Offset from file start to bitmap data (14+40+4*(2^biBitCount)) (in the case of a color plate). 4600 0000, which is 00000046h=70, the header of the above file is 35 words = 70 bytes.

**Bitmap information header**

6) 15-18: Bitmap information header length

7) 19-22: Bitmap width in pixels. 8000 0000, 00000080h=128

8) 23-26: Bitmap height in pixels. 9000 0000, for 00000090h=144

9) 27-28: The number planes of the bitmap,This value is always 1. 0100, which is 0001h=1

10) 29-30: The number of bits per pixel. There are 1 (monochrome), 4 (16 colors), 8 (256 colors), 16 (64K colors, high color), 24 (16M colors, true colors), 32 (4096M colors, enhanced true colors). 1000 is 0010h=16.

11) 31-34: Compression Description: 0 (no compression), 1 (RLE 8, 8-bit RLE compression), 2 (RLE 4, 4-bit RLE compression, 3 (Bitfields, bit field storage)

12) 35-38: The size of the bitmap data expressed in bytes, which must be a multiple of 4, and is numerically equal to: the number of bytes in a row × the height of the bitmap. 0090 0000 is 00009000h=80×90×2h=36864. Assuming the bitmap is 24 bits, the width is 41, and the height is 30, the value = (biWidth*biBitCount+31)/32*4*biHeight, ie =(41*24+31)/32*4*30=3720

13) 39-42: Horizontal resolution in pixels per meter. A00F 0000 is 0000 0FA0h=4000

14) 43-46: Vertical resolution in pixels per meter. A00F 0000 is 0000 0FA0h=4000

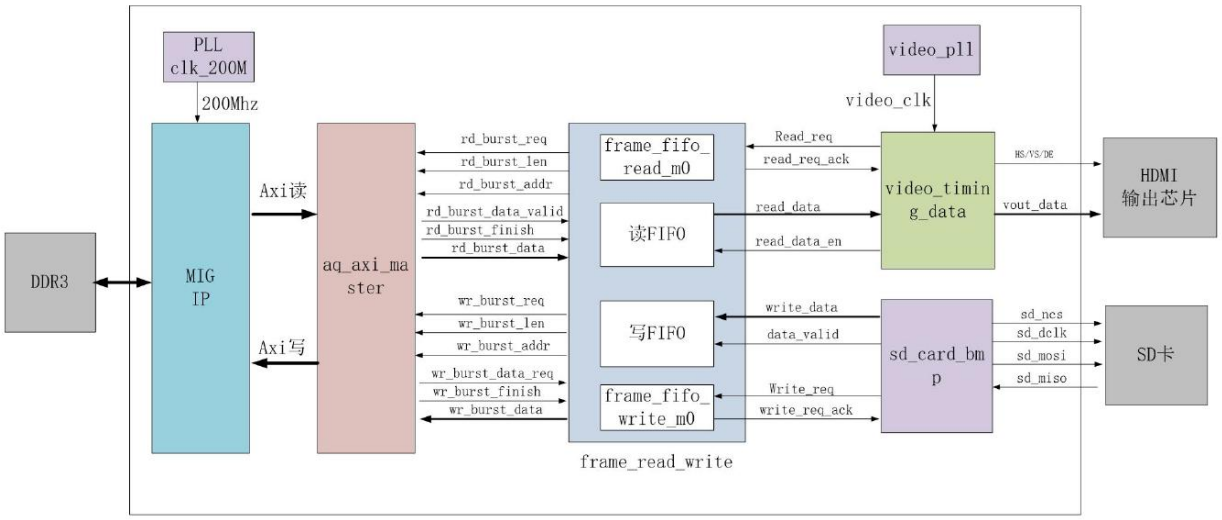
15) 47-50: The number of color indexes used by the bitmap. Set to 0 to indicate that all palette items are used.

16) 51-54: The number of color indices that have a significant impact on the image display. If it is 0, it means that it is important

***This experiment uses a non-compressed, 24 (16M color, true color) BMP picture, the file header size is 54 bytes, the first two bytes are "BM", followed by 4 bytes is the file size, 19-22 The byte is the width of the image, which is important information to be used in the programming.***

# 3 Programming

The program first saves the picture data in the SD card to the external DDR3 memory, and then takes it out and displays it on the HDMI or VGA display. Due to SD card reading and writing experiments, VGA, LCD display experiments, we have already practiced SD card and video related knowledge, no longer explain here.The logic block diagram of the program is shown in the following figure:

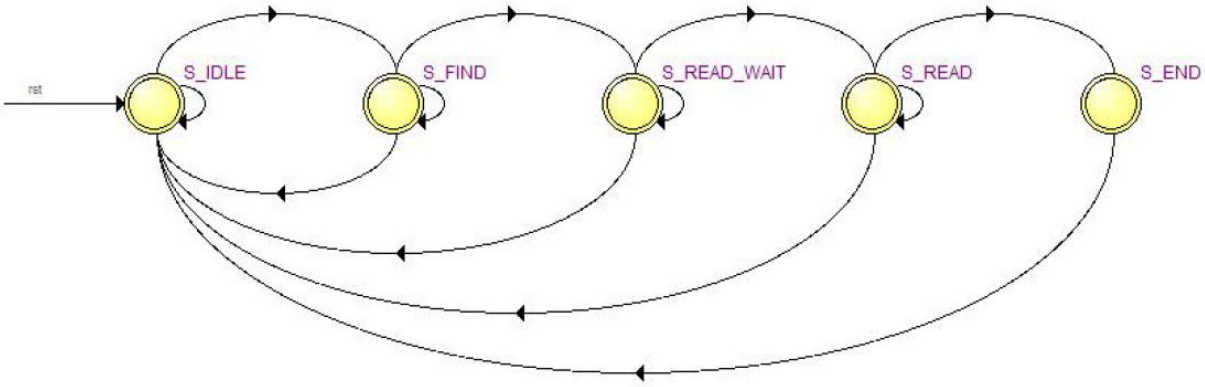



The key module of this experiment is the BMP picture reading module “bmp_read”. The “bmp_read” module completes reading the data of one sector in the SD card, and then compares it with the file header of the BMP. If the first 2 bytes are equal to "BM", then find 19-22 bytes, compare the width of the picture and the width of the input request. If they agree, find a BMP picture, read it, remove the first 54 bytes of the file header, and write to the external memory.

Due to the SD card reading and writing experiments, HDMI and LCD display experiments, we have already practiced SD card and video related knowledge, and will not be explained here.

sd_card_bmp includes 3 modules: BMP picture reading module bmp_read, SD card control module sd_card_top and key detection module ax_debounce. The key module of this experiment is the BMP image reading module bmp_read. The bmp_read module completes reading the data of a sector in the SD card, and then compares it with the BMP file header. If the first 2 bytes are equal to "BM", then find 19 -22 bytes, compare whether the width of the picture is consistent with the width required by the input. If they are consistent, it is considered to find a BMP picture, read it out, remove the first 54 bytes of the file header, and write it into the external memory.

The “bmp_read” state machine is as follows. After the search command, enter the search state "S_FIND", start to read the SD card continuously, find the BMP picture that meets the requirements, after finding it, go to “S_READ_WAIT” and judge the size of the FIFO space. If the FIFO space is large enough, enter the “S_READ” state.




bmp_read module state machine

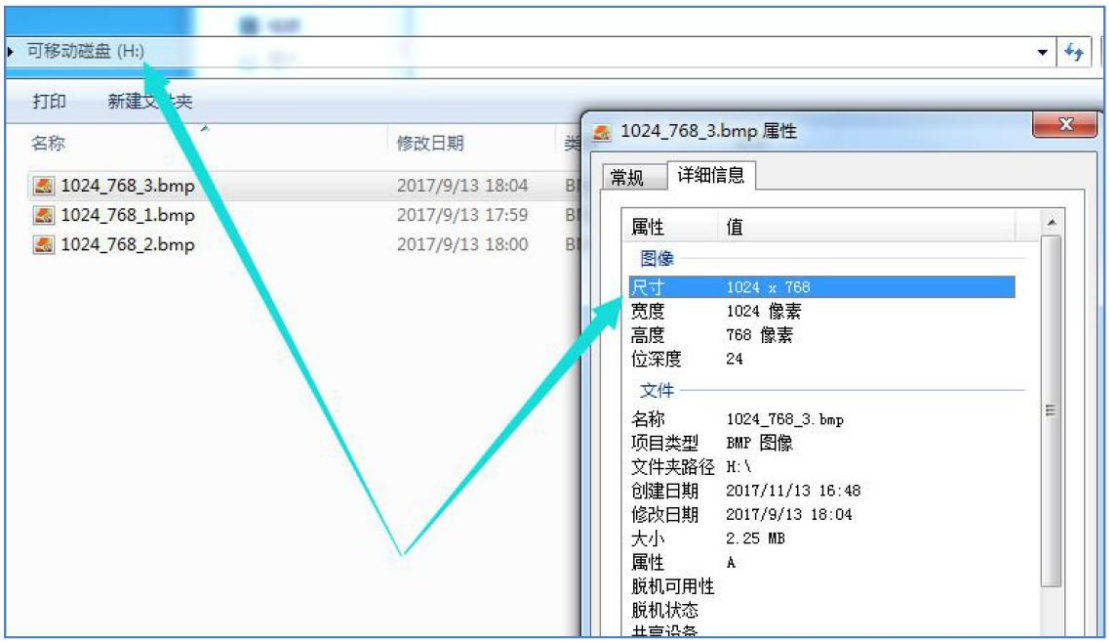
| Signal Name | Direction | Description |
|---|---|---|
| clk | in | clock input |
| rst | in | asynchronous reset input, high reset |
| ready | out | idle status indication |
| find | in | search play request |
| sd_init_done | in | sd card initialization completed |
| state_code | out | status code<br><br>0, indicating that sd is still initializing<br><br>1, sd card initialization is complete, waiting for the button to press<br><br>2, searching for BMP files<br><br>3, find the BMP file, playing |
| bmp_width | in | Search for the BMP image width |
| write_req | out | Write external memory request |
| write_req_ack | in | Write external memory request response |
| sd_sec_read | out | sd card read request |
| sd_sec_read_addr | out | sd card read request sector address |
| sd_sec_read_data | in | Data read by sd card |
| sd_sec_read_data_valid | in | sd card read data is valid |
| sd_sec_read_end | in | sd card read request completed |
| bmp_data_wr_en | | Bmp file write enable |
| bmp_data | | Audio data for bmp files |

The “video_timing_data” module completes the conversion of the video timing to the FIFO read signal. The main principle is to use the "DE" in the video timing as the FIFO read signal, but the read data will be delayed, so the corresponding alignment processing is performed.

# 4 Experiment Result

1) Format sd card (fat32 format), sd card must be version 2.0 or higher (capacity greater than 4G).


2) Put the BMP format file into the sd card. Note that the BMP image data storage is reversed, so use the image processing tool to reverse it upside down. According to the different resolutions of the display output, the pictures with different resolutions are placed, the HDMI output adopts the 1024x768 resolution, the 7-inch LCD module AN070 resolution is 800x480, and the 4.3-inch LCD module AN430 resolution is 480x272.



3) Insert the prepared “sd” card into the sd card slot of the development board (***When the development board is powered on, the sd card cannot be plugged or unplugged***), Power on, Download the experimental program. When waiting for LED1 to turn on, press KEY1, at this time LED2 will turn on, indicating that the BMP file is being searched. If a BMP image is found, LED1 and LED2 will be displayed and will be on at the same time, and the display (or LCD screen module, selected according to the experimental project) will Display the corresponding picture. If there are multiple BMPs in the sd card, you can press the button KEY1 again, and the next picture will be displayed.


Display Effect