

# SD 卡读取 BMP 图片显示例程

---

## 1 实验简介

在前面的实验中我们练习了 SD 卡读写, HDMI 视频显示等例程, 本实验将 SD 卡里的 BMP 图片读出, 写入到外部存储器, 再通过 HDMI、LCD 等显示。

本实验如果通过液晶屏显示, 需要有液晶屏模块。

## 2 实验原理

在前面的实验中我们在 HDMI、LCD 上显示的是彩条, 是 FPGA 内部产生的数据, 本实验将彩条替换为 SD 内的 BMP 图片数据, 但是 SD 卡读取速度远远不能满足显示速度的要求, 只能先写入外部高速 RAM, 再读出后给视频时序模块显示。

### BMP 图片格式

本实验直接在 SD 卡中搜索 BMP 文件, 假设每个文件都是从 SD 的某个扇区第一个字节开始, 根据 BMP 文件头的特征找到 BMP。

BMP (全称 Bitmap) 是 Windows 操作系统中的标准图像文件格式, 可以分成两类: 设备相关位图 (DDB) 和设备无关位图 (DIB) , 使用非常广。它采用位映射存储格式, 除了图像深度可选以外, 不采用其他任何压缩, 因此, BMP 文件所占用的空间很大。BMP 文件的图像深度可选 1bit、4bit、8bit 及 24bit。BMP 文件存储数据时, 图像的扫描方式是按从左到右、从下到上的顺序。由于 BMP 文件格式是 Windows 环境中交换与图有关的数据的一种标准, 因此在 Windows 环境中运行的图形图像软件都支持 BMP 图像格式。

对于程序设计来说最重要的是找到 BMP 文件头, BMP 图像文件头格式如下:

1) 1-2: (这里的数字代表的是字节, 下同) 图像文件头。0x4d42=' BM' , 表示是 Windows 支持的 BMP 格式。(注意: 查 ascii 表 B 0x42,M 0x4d,bfType 为两个字节, B 为 low 字节, M 为 high 字节所以 bfType=0x4D42, 而不是 0x424D, 请注意)

- 
- 2) 3-6: 整个文件大小。4690 0000, 为 00009046h=36934。
  - 3) 7-8: 保留, 必须设置为 0。
  - 4) 9-10: 保留, 必须设置为 0。
  - 5) 11-14: 从文件开始到位图数据之间的偏移量( $14+40+4^* (2^{\text{biBitCount}})$ )(在有颜色板的情况下)。4600 0000, 为 00000046h=70, 上面的文件头就是 35 字节=70 字节。

### 位图信息头

- 6) 15-18: 位图图信息头长度。
- 7) 19-22: 位图宽度, 以像素为单位。8000 0000, 为 00000080h=128。
- 8) 23-26: 位图高度, 以像素为单位。9000 0000, 为 00000090h=144。
- 9) 27-28: 位图的位面数, 该值总是 1。0100, 为 0001h=1。
- 10) 29-30: 每个像素的位数。有 1 (单色), 4 (16 色), 8 (256 色), 16 (64K 色, 高彩色), 24 (16M 色, 真彩色), 32 (4096M 色, 增强型真彩色)。1000 为 0010h=16。
- 11) 31-34: 压缩说明: 有 0 (不压缩), 1 (RLE 8, 8 位 RLE 压缩), 2 (RLE 4, 4 位 RLE 压缩, 3 (Bitfields, 位域存放))。
- 12) 35-38: 用字节数表示的位图数据的大小, 该数必须是 4 的倍数, 数值上等于:一行所占的字节数×位图高度。0090 0000 为 00009000h=80×90×2h=36864。假设位图是 24 位, 宽为 41, 高为 30, 则数值=  $(\text{biWidth} * \text{biBitCount} + 31) / 32 * 4 * \text{biHeight}$ , 即=  $(41 * 24 + 31) / 32 * 4 * 30 = 3720$
- 13) 39-42: 用象素/米表示的水平分辨率。A00F 0000 为 0000 0FA0h=4000。
- 14) 43-46: 用象素/米表示的垂直分辨率。A00F 0000 为 0000 0FA0h=4000。
- 15) 47-50: 位图使用的颜色索引数。设为 0 的话, 则说明使用所有调色板项。
- 16) 51-54: 对图象显示有重要影响的颜色索引的数目。如果是 0, 表示都重要。

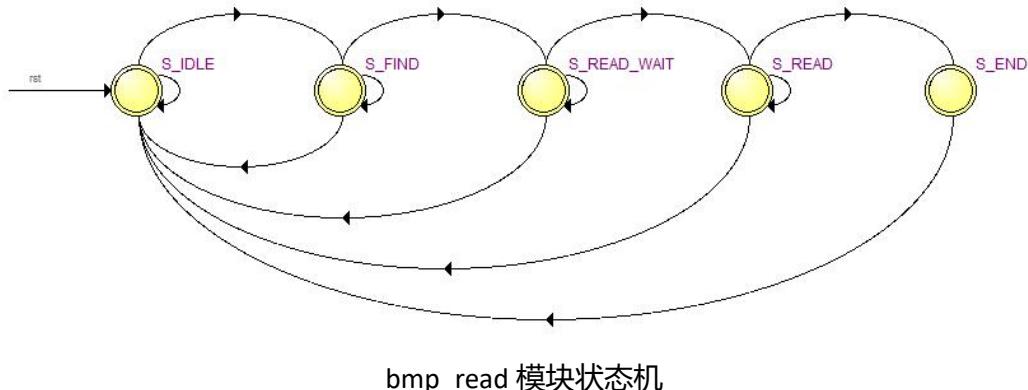
本实验使用不压缩, 24 (16M 色, 真彩色) 的BMP 图片, 文件头大小是 54 个字节, 前两个字节为 “BM”, 紧接着 4 个字节是文件大小, 19-22 字节为图片宽度, 这些信息是程序设计中要使用的重要信息。

### 3 程序设计

由于 SD 卡读写实验, HDMI、LCD 显示实验种我们已经练习过 SD 卡和视频相关知识, 这里不再讲解。

本实验的重点模块是 BMP 图片读取模块 `bmp_read`, `bmp_read` 模块完成 SD 卡中读取一个扇区的数据, 然后和 BMP 的文件头对比, 如果前 2 个字节等于 “BM”, 然后再找到 19-22 字节, 对比图片的宽度和输入要求的宽度是否一致, 如果一致就认为找到一张 BMP 图片, 读取出来, 去掉前面 54 字节的文件头, 写入外部存储器。

`bmp_read` 状态机如下所示, 有搜索命令以后, 进入搜索状态 “`S_FIND`”, 开始不断地读取 SD 卡, 找到符合要求的 BMP 图片, 找到以后进入 “`S_READ_WAIT`”, 判断 FIFO 空间大小, 如果 FIFO 空间足够大, 进入 “`S_READ`” 状态。



`bmp_read` 模块状态机

信号名称	方向	说明
<code>clk</code>	in	时钟输入
<code>rst</code>	in	异步复位输入, 高复位
<code>ready</code>	out	空闲状态指示
<code>find</code>	in	搜索播放请求
<code>sd_init_done</code>	in	sd 卡初始化完成
<code>state_code</code>	out	状态码 0, 表示 sd 还在初始化

---

		1, sd 卡初始化完成, 等待按键按下 2, 正在搜索 BMP 文件 3, 找到 BMP 文件, 正在读取
<b>bmp_width</b>	in	搜索 BMP 图片的宽度
<b>write_req</b>	out	写外部存储器请求
<b>write_req_ack</b>	in	写外部存储器请求应答
<b>sd_sec_read</b>	out	sd 卡读请求
<b>sd_sec_read_addr</b>	out	sd 卡读请求扇区地址
<b>sd_sec_read_data</b>	in	sd 卡读到的数据
<b>sd_sec_read_data_valid</b>	in	sd 卡读数据有效
<b>sd_sec_read_end</b>	in	sd 卡读请求完成
<b>bmp_data_wr_en</b>	out	bmp 文件写使能
<b>bmp_data</b>	out	bmp 文件的数据

bmp\_read 模块端口

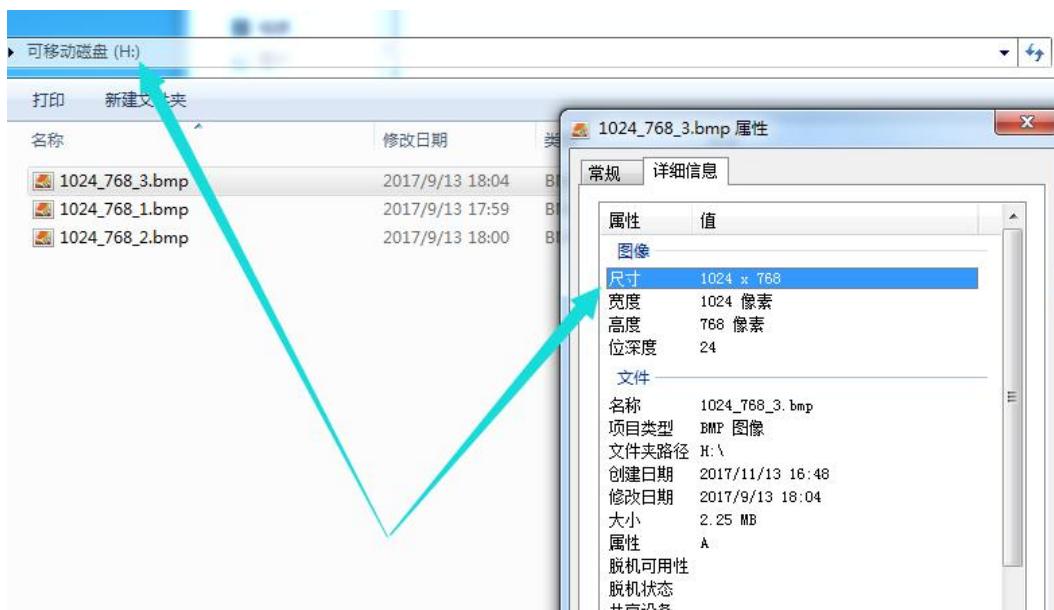
video\_timing\_data 模块完成视频时序到 FIFO 读取的信号的转换, 主要原理就是把视频时序中的“DE”做为 FIFO 的读信号, 但是读出的数据会有延时, 所以做了相应的对齐处理。

## 4 实验现象

- (1) 格式化 sd 卡 (fat32 格式), sd 卡必须是 2.0 以上的版本 (容量大于 4G)



(2) 把 BMP 格式文件放到 sd 卡中，需要注意，BMP 图片数据存储是倒序，所以先用图片处理工具上下颠倒一下。根据显示输出不同的分辨率，放置不同分辨率的图片，HDMI 输出采用 1024x768 分辨率，7 寸液晶屏模块 AN870 分辨率是 800x480,4.3 寸液晶屏模块 AN430 分辨率是 480x272。



(3) 将准备好的 sd 卡注入开发板的 sd 卡槽 (sd 卡不能带电插拔) ，上电，下载实验程序，等待 LED1 变亮时，按下 KEY2，这个时候 LED2 会变亮，表示正在搜索 BMP 文件，如果

找到 BMP 图片 LED1、LED2 会显示会同时亮，这时候显示器（或者液晶屏模块，根据实验工程选择，连接方法在《HDMI 测试实验教程》中已讲述）就会显示相应的图片。如果 sd 卡内有多张 BMP，可以再次按按键 KEY2，会显示下一张图片。



开发板操作



显示效果