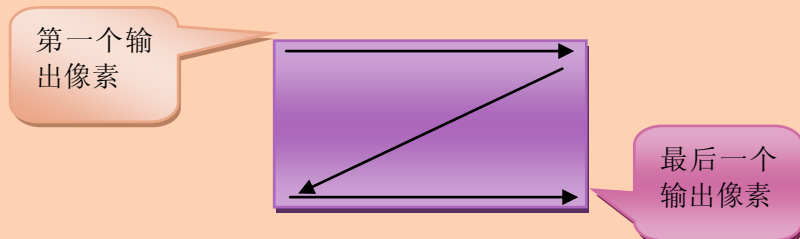


ov7725 数字摄像头编程基本知识笔记

这里以 ov7725 为例，对数字摄像头的时序进行分析。其他数字摄像头的时序也大同小异。

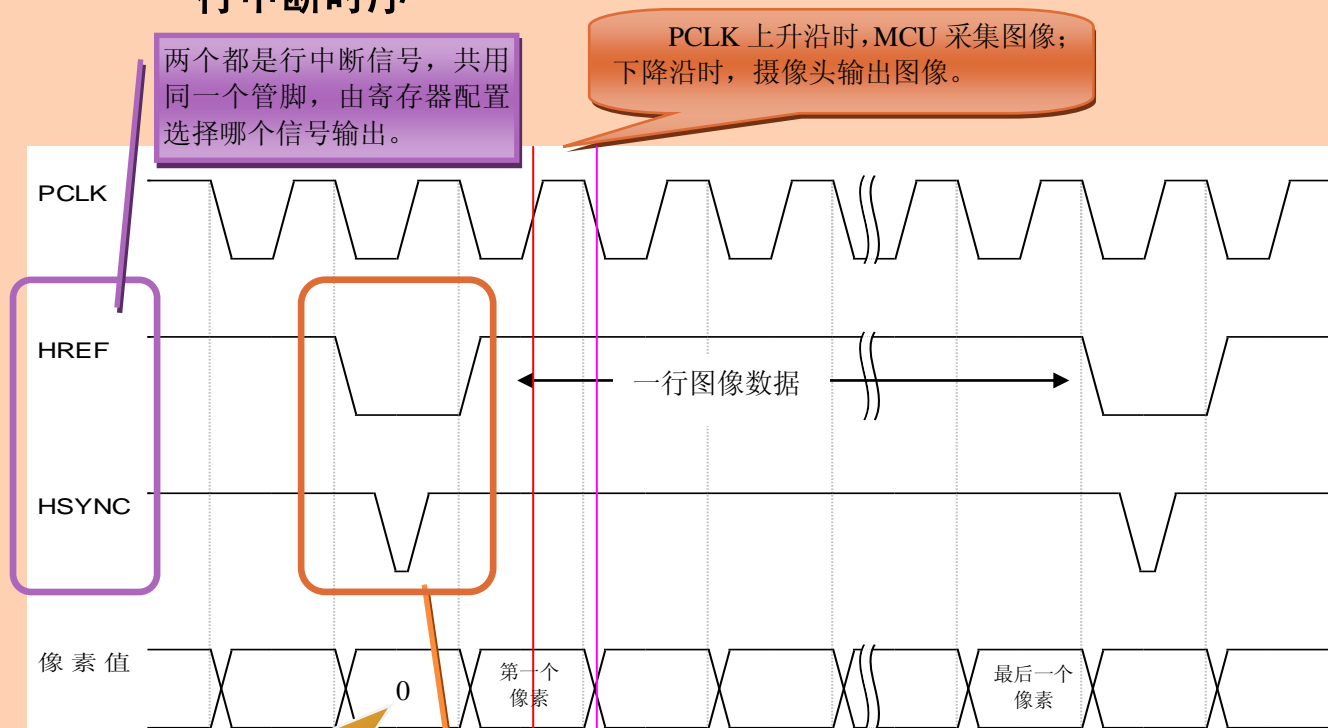
像素输出顺序

数字摄像头输出图像时，一般都是从左到右，有上到下逐个输出（部分芯片可配置输出顺序）：



有些摄像头有奇偶场，是采用隔行扫描方法，把一帧图象分为奇数场和偶数场两场。（ov7725 没有奇偶场之分）

行中断时序

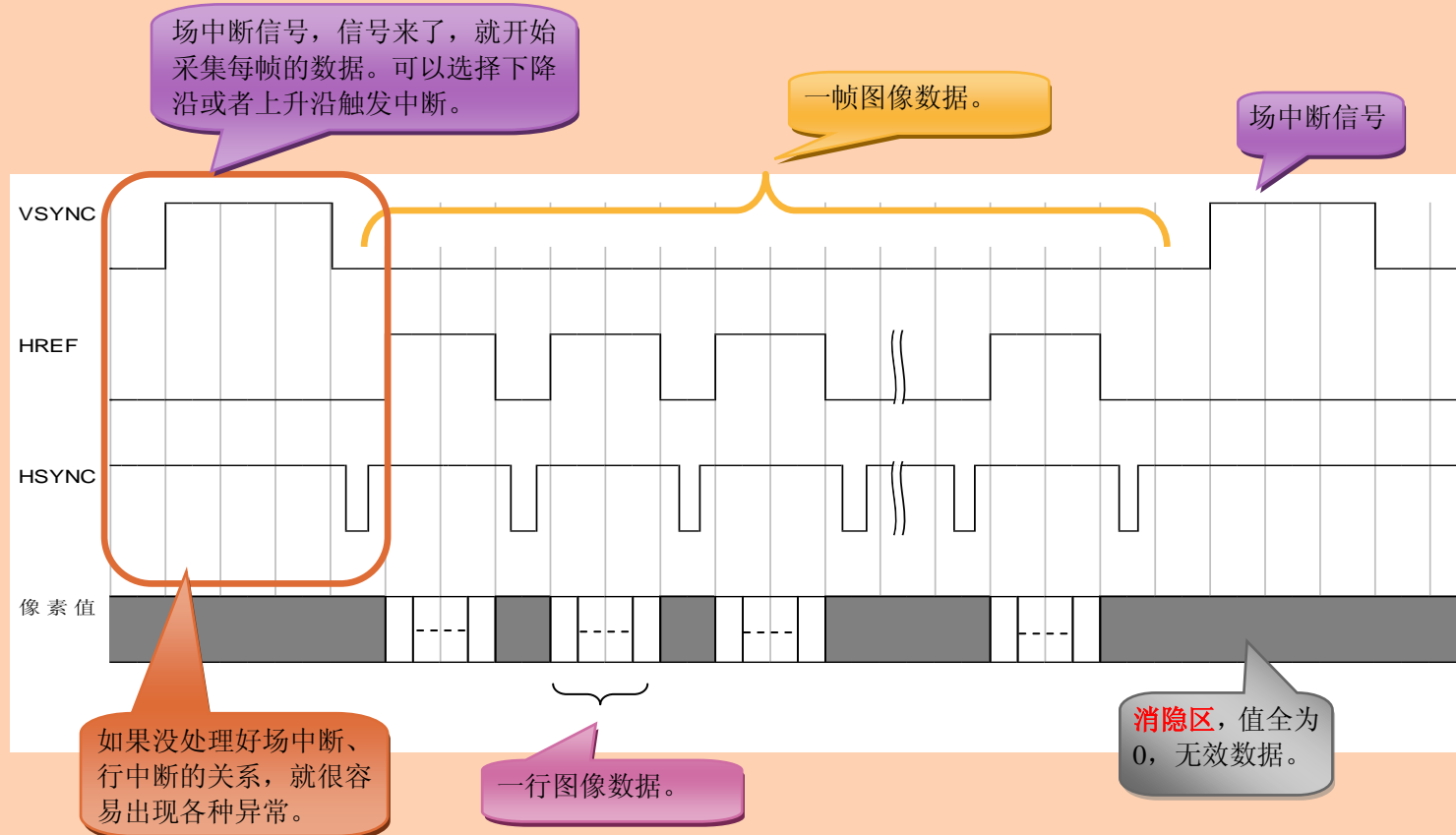


消隐区，如果不按照时序来采集，就有可能采集到消隐区，值为 0，即黑色。行与行之间，场与场之间都有消隐区的。

HREF 和 HSYNC 都用于行中断信号，但时序有点区别。

HREF 上升沿就马上输出图像数据，而 HSYNC 会等待一段时间再输出图像数据，如果行中断里需要处理事情再开始采集，则显然用 HREF 的上升沿是很容易来不及采集第一个像素。

场中断时序



采集图像思路

这种方法是最简单，最多人用，但也是采集图像最不稳定的方法，经常出现消隐区等问题。

常见图像采集问题里有对应的伪代码。

①使用 for 循环延时采集

1. 需要采集图像时，开场中断
2. 场中断来了就开启行中断，关闭场中断
3. 行中断里用 for 循环延时采集像素，可以在行中断里添加标志位，部分行不采集，即可跨行采集。
4. 行中断次数等于图像行数时即可关闭行中断，标志图像采集完毕。

②使用场中断和行中断，DMA 传输

1. 需要采集图像时，开场中断
2. 场中断来了，开行中断和初始化 DMA 传输
3. 行中断来了就设置 DMA 地址，启动 DMA 传输。如果先过滤部分行不采集，则设置一个静态变量，每次行中断来了都自加 1，根据值来选择采集或不采集某些行。
4. 每个 PCLK 上升沿来了都触发 DMA 传输，把摄像头输出的值读取到内存数组里。当触发 n 次 (n=图像列数目) 后就停止 DMA 传输。
5. 行中断次数等于一幅图像的行数，或者等待下一个场中断来临 就结束图像采集，关闭行中断和场中断。

如果不用 DMA 传输，则直接用 for 循环加延时来采集就好，PCLK 也不需要。不过延时值需要设置合适，不然要不就采集到消隐区，要不就只采集图像的左边部分。

③使用场中断，DMA 传输

不使用行中断信号，直接 DMA 模块计数来完成一场结束的判断

1. 需要采集图像时, 开场中断
2. 场中断来了, 初始化 DMA 传输, 并启动 DMA 传输
3. 每个 PCLK 上升沿来了都触发 DMA 传输, 把摄像头输出的值读取到内存数组里。
当触发 n 次 (n =图像像素数目) 后就停止 DMA 传输。
4. DMA 停止传输时触发中断, 中断里关闭场中断, 图像采集完毕。
或者等待下一个场中断来临才关闭场中断, 标记图像采集完毕

常见图像采集问题

1. 图像分成上下两幅图图像

没处理好场中断信号: 不根据场中断信号来了才开始每帧的采集, 就会出现采集到一幅图片里由上下两幅图片组成: 上半幅图是前一帧图像的底部, 下半幅图是这帧的顶部图像。

或者场中断来了后, 先处理太多东西, 然后才开始采集, 导致跳过了开头的数据, 后续采集的时候又根据采集的行数来判断是否停止采集, 就好采集到下一幅图像的开头



是下一帧的顶部
图像。

论坛帖子: <http://www.znczz.com/forum.php?mod=viewthread&tid=98576>
<http://www.znczz.com/forum.php?mod=viewthread&tid=24642>

2. 图像分成左右两半图像, 左半幅图是右半幅图的右边图像。

没处理好行中断, 可能行中断来了后处理太多东西才开始采集。

3. 图像分成左右两半相同的图片, 而且中间有消隐区

读取行的数目太多, 或者中间延时太久, 导致读取一行变成的读取两行



论坛帖子: <http://www.znczz.com/forum.php?mod=viewthread&tid=98767>
<http://www.znczz.com/forum.php?mod=viewthread&tid=97071>

4. 图像采集没居中, 图像中心偏左或偏右了。

排除摄像头镜头歪了, 或者摄像头芯片焊接歪了的硬件问题外, 一般情况下是出现在用 for 循环延时采集上。

这些先写一下 for 循环延时的伪代码:

```
1. //for 循环延时采集, 都是在行中断里采集一行数据
2. void Hsr()
3. {
4.     static uint16 Hn=0; //正在采集的行数
5.
6.     //行中断来了延时一下, 跳过消隐区
7.     DELAY_A();
8.
9.     for(i=0; i<每行元素数目; i++)
10.    {
11.        //采集图像值, 并保存在对应的内存数组里
12.        port2buff(Hn++, i);
13.
14.        //单片机速度较快时, 如果不加延时, 那么很有可能同一个像素
15.        //值采集多次, 而且图像右边的像素没采集到, 即会出现图像中心偏右。
16.        DELAY_B();
17.    }
18.
19.    Hn++;
20. }
```

如果 **DELAY_A()**延时值**过大**, 图像左边的部分像素就没采集到, 图像中心偏左。

如果 **DELAY_A()**延时值**过小**, 图像左边就会出现消隐区。**HREF** 上升沿触发 行中断 就不需要添加 **DELAY_A()**延时值。

如果 **DELAY_B()**延时值**过大**, 右边图像就会出现消隐区。当不添加 **DELAY_B()**还是出现右边有消隐区时, 说明单片机速度太慢, **每行采集像素数目太多**, 可以减少每行采集的数目。

如果 **DELAY_B()**延时值**过小**, 右边图像没采集到, 图像中心偏右

每行元素数目过大, 即上面所说的出现右边有消隐区情况。

每行元素数目过小, 显然采集图片太窄, 影响路况识别。

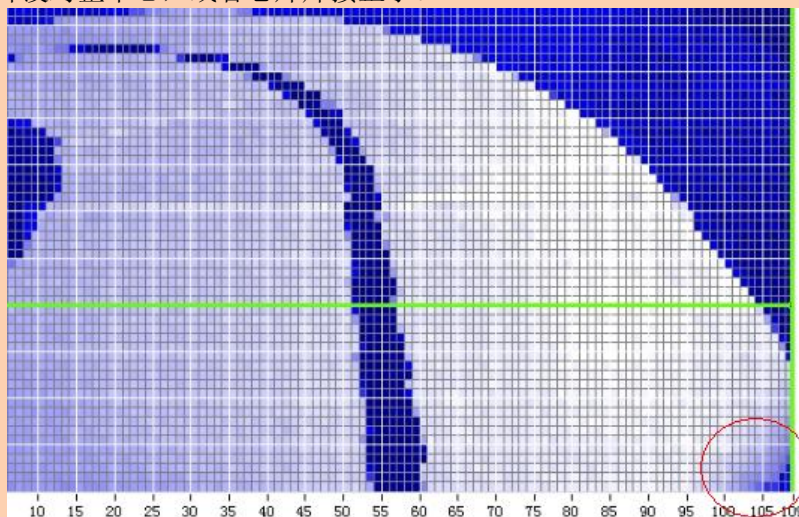
偏左了:



论坛帖子: <http://www.znczz.com/forum.php?mod=viewthread&tid=98644>

5. 图像有四个角落有阴影

镜头与芯片没对正中心, 或者芯片焊接歪了。



论坛帖子: <http://www.znczz.com/forum.php?mod=viewthread&tid=96179>

SCCB 介绍

I2C 协议 + 读操作时停止条件 = SCCB 协议
兼容 I2C 协议，可直接用 I2C 模块来控制……

SCCB 是 ov 系列摄像头的控制信号通信的协议，兼容 I2C 协议，与 I2C 协议唯一不同的是 SCCB 在读时，增加一个总线停止条件（SCCB 信号线需要接上拉电阻，理由与 I2C 一样）。

ov 系列摄像头的寄存器是 EEPROM，不稳定，数据容易丢失，因此程序每次初始化时都重新写入寄存器配置。

ov 系列摄像头通常都是配置好默认的参数，但特定的场合需要特定的配置，因此有必要学习 SCCB 控制摄像头，从而发挥摄像头的最大采集效果。

常见需要修改的寄存器有，PCLK 速率，帧率、图像亮度、对比度、色饱和度、镜像等功能。

野火鹰眼摄像头简介

鹰，拥有一双敏捷、锐利的眼睛。野火硬件二值化摄像头，速率可达 150 帧每秒，去噪点能力极强，二值化效果非常理想，如鹰眼般的敏捷，故取名为：鹰眼。



野火鹰眼摄像头，是野火为智能车比赛而专门量身打造的摄像头模块，采用 ov 系列三十万像素当中成像质量最棒、低照度极好的 ov7725 芯片，通过特殊的硬件结构完成二值化操作，图像处理效果极佳，采集图像速度极快，为目前智能车比赛选用摄像头中 fastest、二值化处理效果最佳的无敌摄像头。巅峰决战之刻，百虎相争，**野火鹰眼**助你傲视群雄，勇创佳绩。

野火鹰眼选用 ov7725 的理由

ov7725 信噪比更高、速度更快、稳定性更好和微光灵敏度更高、绝非 ov7620、ov7670、ov6620 这类可比，ov 系列三十万像素当中成像质量最棒的摄像头，是智能车比赛的最佳选

择。

低照度好（即微光灵敏度高）有什么好处？

正如我们平时拍摄高速运动中物体的图片一样，高速奔跑中的小车拍摄的图像非常容易变模糊的，目前智能车比赛所用的摄像头中，没一个卖家的摄像头采集效果图片是晃动中拍摄的，原因你懂的。

如何克服拍摄运动图像变模糊问题呢？没错，减少曝光时间……但摄像头芯片，减少曝光时间必须通过提高频率来完成，但当速度加快时，单片机的 I/O 速度就限制了速度的加快，目前常见的解决方法是使用高速 FIFO 来解决。

为什么野火鹰眼传输速度更快，反而号称更加稳定？

野火鹰眼，是一个硬件二值化摄像头，直接输出二值化图像，一次传输 8 个像素，相比与黑白摄像头一次传输一个像素快 8 倍、彩色摄像头两次传输一个像素快 16 倍。

假设黑白摄像头每秒传输 30 帧，同样的数据线总线频率下，则野火鹰眼对应的速度是 $30 \text{ 帧} \times 8 = 240 \text{ 帧}$ ……

假设彩色摄像头每秒传输 30 帧，同样的数据线总线频率下，则野火鹰眼对应的速度是 $30 \text{ 帧} \times 16 = 480 \text{ 帧}$ ……

元芳，你怎么看？恐怖吧？

野火鹰眼设置为最高速度 150 帧，换句话说，比 30 帧的黑白摄像头和彩色摄像头的数据总线频率都低很多，从而传输更加稳定。对应普通摄像头模块，不知道你们有没有出现过信号线过长的时候，图像无法采集，或者采集图像乱了的问题呢？这就是因为数据总线频率高了，对电气要求更高，容易出现振铃现象，从而导致传输异常。

另外一方面，野火鹰眼，采用 ov7725 芯片，BGA 封装，而市场上目前智能车比赛常用的摄像头芯片大部分都是 CLCC 封装，电气特性远不如 BAG 封装。

BGA 封装的优点如下：

- 电气特性好
 - 管脚短，寄生电感、电容、电阻小，串扰小；
 - 管脚引线长度易于控制，延时一致性好；
- 高速、高频信号完整性好
 - 高速、高频信号完整性好信号完整性好——互扰小
 - 抗干扰性好接地完整性好——地弹小
 - 电源完整性好——电弹小
- 适合多层板——配合接地层/电源层，信号返回路径好
- 焊点机械强度高——抗热应力，抗冲击应力

——BGA 优点的介绍引用出处《JTAG 为什么是 BGA 器件最佳测试解决方案？》

还有哦！野火鹰眼，4 层板，信号返回路径好；沉金工艺加工，导电性能更高……

o(￣▽￣)o 唉，太多优点了，限于篇幅，就不一一详细讲了……

为什么不用 FIFO？

1. 飞思卡尔智能车组委会对是否允许购买带 FIFO 摄像头模块态度不确定，冒着违规的风险。
2. FIFO 仅仅是加快摄像头的采集速度，但会减慢单片机采集图像的速度，在这个比拼速度的比赛，别的摄像头 30 帧每秒的采集速度都不稳定，而我们的摄像头却高达 150 帧每秒稳定采集，高的何止一个档次。

硬件二值化适应性不佳？

其他的摄像头模块，硬件二值化是全局阈值二值化，处理效果难以称得上理想，而且某些硬件二值化摄像头模块的阈值固定，不可调，适应性确实不佳。基于上述理由，因此市场

上还没有针对智能车比赛的硬件二值化摄像头模块。

但对于野火鹰眼而言,答案是 NO……野火鹰眼,使用的二值化方式绝非全局阈值二值化那么简单。不同的均匀光照情况下,依旧保存较好的图像二值化效果。

注:过暗、过亮异常情况下,野火鹰眼没法很好地二值化,二值化后基本上都是白色,这一点是所有的摄像头也没法克服。关于反光问题,野火鹰眼的二值化处理可以减轻反光问题所带来的影响(不妨把野火鹰眼与其他摄像头模块对比,你就会发现野火鹰眼对反光问题处理更好),但没法完全克服,这也是摄像头芯片目前无法克服的问题。如果要更好地克服反光问题,就必须从镜头方面着手,例如镜头镀膜加滤光片,但价格也贵很多。

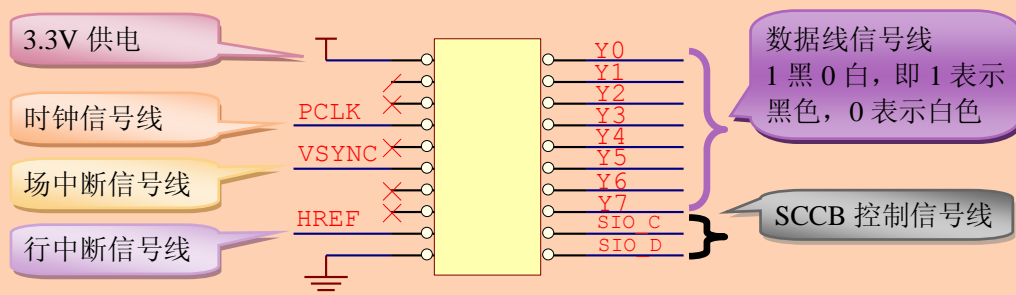
为什么图像歪了?

目前市场上专门为飞思卡尔智能车比赛设计的摄像头模块,大部分的芯片都是 CLCC 封装,而非 BGA 封装,焊接过程中容易出现芯片焊接偏移问题,从而导致了采集到的图像也是偏移的。因此购买这类的摄像头模块时,拼的是 RP 值,RP 值不好,就购买到焊接偏移的摄像头。

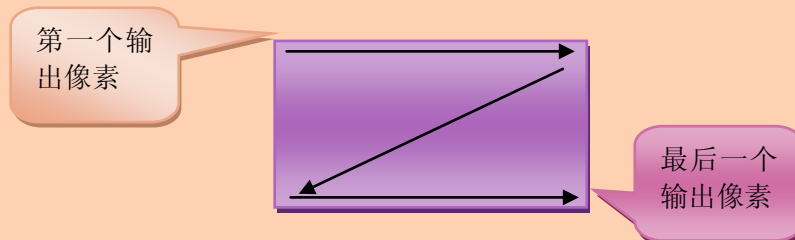
野火鹰眼,采用 ov7725 芯片, BGA 封装,焊接过程中会自动居正,不存在焊接偏移问题,而且 BGA 封装的电气特性更佳,这也是为什么较为高端的摄像头芯片都是清一色的 BGA 封装。

买摄像头模块还得拼 RP,而且图像质量还难以称得上理想,传输速度慢而且不稳定,还要为二值化处理而烦恼,但居然还有人继续购买,元芳,你怎么看?

野火鹰眼摄像头接口及数据格式



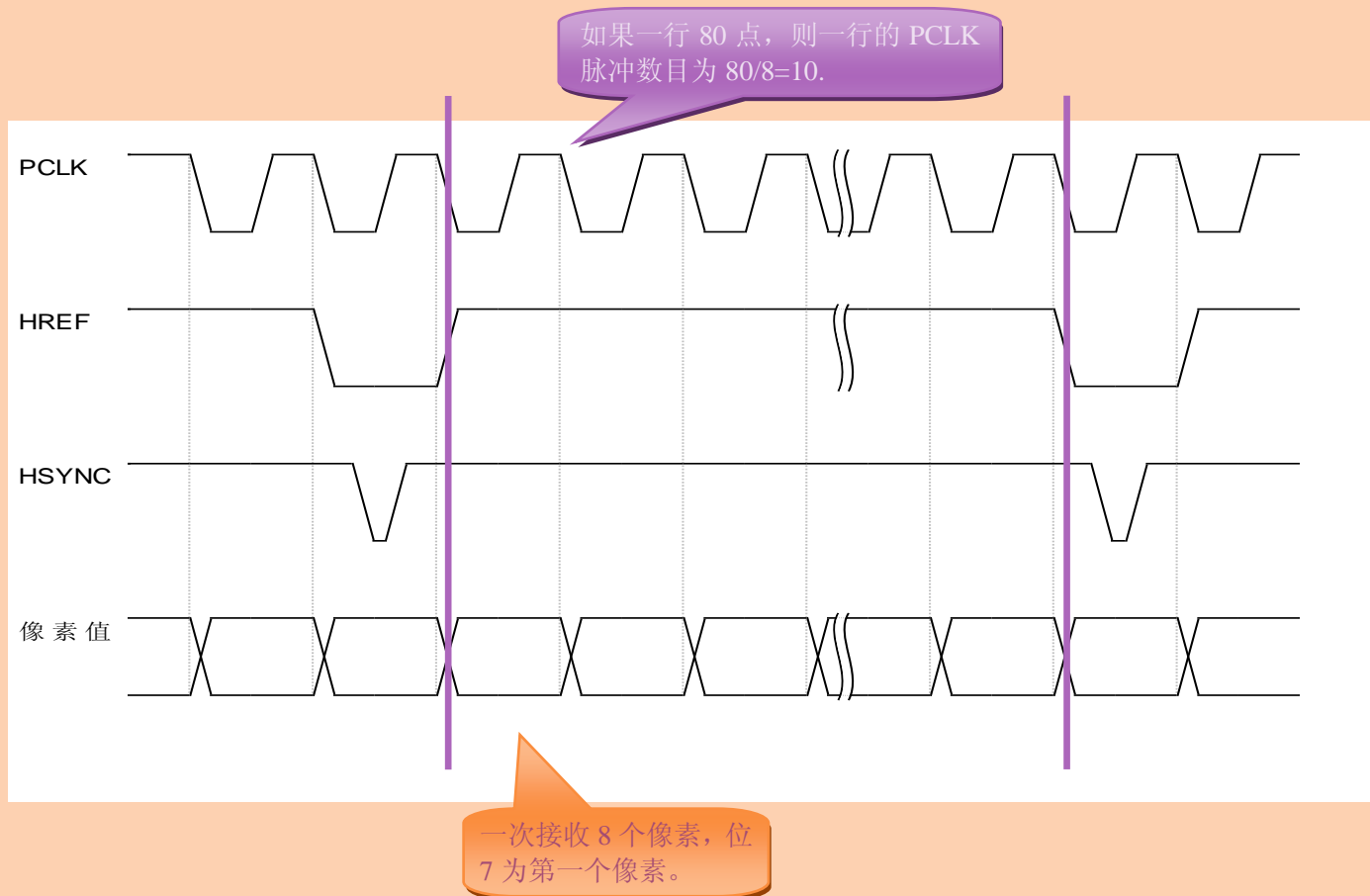
野火鹰眼是硬件二值化摄像头,8 位数据线一次传输 8 个像素,ov7725 的图像像素先输出则在高位,后输出则低位。



Ov7725 默认图像的输出顺序是从左到右,从上到下,即第一个接收字节的最高位为第一个像素(最左上角像素),最低位为第 8 个像素。

野火鹰眼摄像头时序

野火鹰眼, 时序与 ov7725 完全一样的, 唯一不同的是数据格式:



假设第一行接收了的前 5 个字节数据如下: 0x80,0x01,0x55,0xAA,0x01
则我们看到的图像为:

