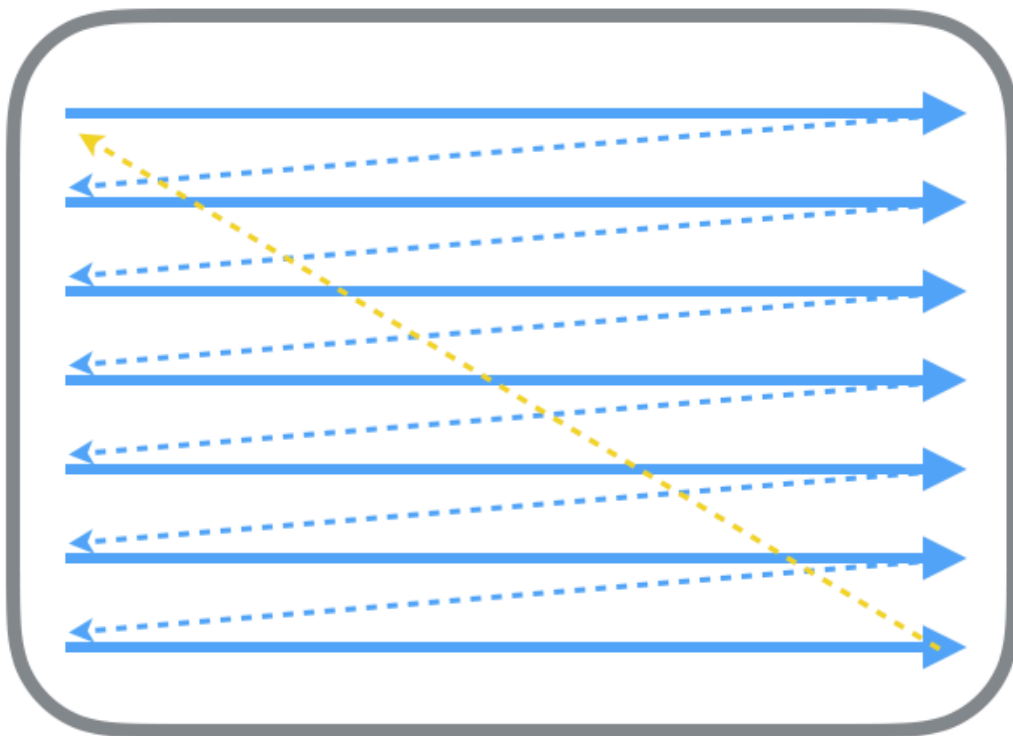


屏幕显示图像原理

电子屏幕显示图像，依靠的是电子枪扫描的方式，从上到下一行行扫描，扫描完成后显示器就可以显示一帧画面。随后电子枪回到初始位置继续下一次扫描。

过程如下图：



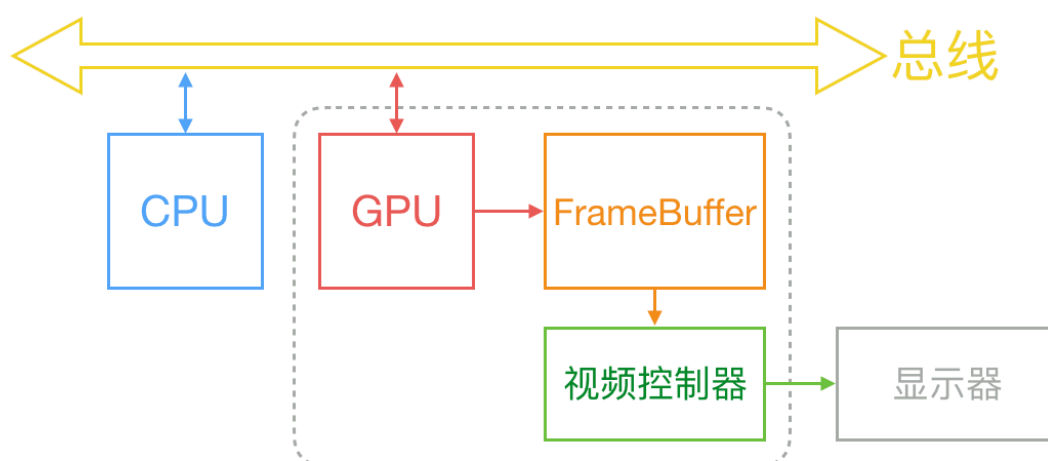
操作系统通常来说都会有视频控制器，显示器需要和视频控制器保持同步。简单来说就是，视频控制器提供内容，显示器负责显示内容。当显示器显示完一帧后，需要通知视频控制器把下一帧内容传过来。

为了把显示器和系统视频控制器保持同步，显示器会用硬件时钟产生一系列的定时信号。当电子枪换到新一行，准备进行扫描时，显示器会发出一个水平同步信号（horizontal synchronization），简称HSync；而当一帧画面绘制完成后，电子枪回到原位，准备画下一帧前，显示器会发出一个垂直同步信号（vertical synchronization），简称VSync。显示器通常以固定频率进行刷新，这个刷新率就是VSync信号产生的频率。

系统渲染过程

计算机系统中，显示器最终显示图像，需要CPU、GPU、显示器共同协作。简单来说，CPU负责计算，将计算好的显示内容提交到GPU，GPU渲染完成后将渲染结果放入帧缓冲区，随后视频控制器会按照VSync信号逐行读取帧缓冲区的数据，经过可能的数模转换结果传递给显示器显示。

整个流程如下图：



在简单情况下，帧缓冲区只有一个，这时帧缓冲区的读取和刷新都会有比较大的效率问题。为了优化，iOS系统采用了双缓冲机制，即两个缓冲区。

双缓冲机制下，GPU会预先渲染好一帧放入一个缓冲区内，让视频控制器读取，当下一帧渲染好后，GPU会直接把视频控制器指针指向第二个缓冲区。这样在效率上有一些提升。

双缓冲机制虽然能解决效率问题，但会引入新的问题。当视频控制器还会读取完成时，即显示器刚显示一半，GPU将新的一帧内容提交到缓冲区，并且GPU把视频控制器指针指向了新提交的缓冲区，最终屏幕就会把下一帧数据的下半段显示到屏幕上。对用户而言，显示的是一张撕裂的图像。如下图：



为了解决撕裂的问题，GPU通常有一个机制叫做垂直同步，简写VSync。当开启垂直同步后，GPU会等待显示器的VSync信号发出后，才会将视频控制器指针指向第二个缓冲区。这样解决了撕裂的现象。

丢帧

丢帧，也就是卡顿。在显示器VSync信号到来后，系统图形服务会通过CADisplayLink等机制通知App，App主线程开始在CPU中计算显示内容，比如视图的创建、布局计算、图片解码、文本绘制等。随后CPU会将计算好的内容提交到GPU去，由GPU进行变换、合成、渲染。随后GPU会将渲染结果提交到帧缓冲区，等待下一次VSync信号到来时显示到屏幕上。由于垂直同步机制，如果在一个VSync时间内，CPU或者GPU没有完成内容提交，则那一帧就会丢弃，等待下一次机会显示，此时显示屏会保留上一帧的内容不变。对于用户来说，就是从前一帧突然变为了后面某帧，中间帧全部丢失，也就是卡顿。