# 图像缓存设计

本文讲述下利用sdram缓存从摄像头处得到的数据，并将图像显示到显示屏上的工程架构。本文不涉及具体的代码讲解，只描述其中的实现思路。

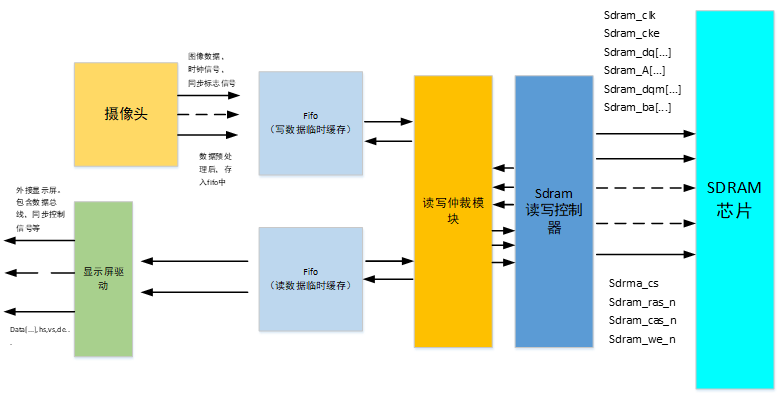
该工程有如下模块组成：

摄像头数据捕获模块，

读fifo，写fifo，

读写仲裁模块，

sdram控制器



每个模块的作用如下。

1，摄像头数据捕获模块。

一般摄像头传输的数据，一个像素是由相邻两个时钟的数据拼接而成的，在数据捕获模块中，要根据同步控制信号，将两个时钟的数据拼接在一起输出。

2，读fifo，写fifo。

用于临时缓存读写数据，解决数据的跨时钟域问题。

3，读写仲裁模块。

用于判断和控制合适从sdram中读写数据。

4，sdram控制器。

用来实现sdram的初始化配置，读写时序的实现。

该工程的总体设计概要图如下，从摄像图获取的的数据先临时缓存到写fifo中，然后再写入到sdram中，需要显示图像时，图像数据先从sdram临时储存到读fifo中，然后显示屏驱动模块读取fifo中的数据，并将图像显示到显示屏上。在sdram中，将图像数据按照从左到右，从上到下的顺序储存在一片连续地址的储存区域内，便可以方便突发读写传输。

1，为什么要用sdram。

由于摄像头产生的数据时钟速率和显示屏的时钟不匹配，而且当摄像头传来数据时，显示屏驱动模块不一定在此时需要数据，况且同一时刻摄像头传进的像素的位置不一定就是显示屏正刷新到的位置。故不能将摄像头的数据直接直接传输到显示屏驱动模块。

一张24bit的的全彩图的，每个像素点占据3byte的空间,常见的不同尺寸的图片一帧所占空间如下：

1080\*1920\*3byte=6,220,800byte=5,9mbyte;

960\*480\*3byte=1，382，400byte=1.3mbyte

480\*272\*3byte=391,680byte=0.3mbyte

通常板载的sdram的大小有16MB,32MB，64MB，128MB，等等，由设计成本决定，但用于储存一帧图片绰绰有余。

只有将图像数据临时储存起来，这样读写便互不所影响。当摄像头传来有效数据时，便将数据存入sdram，当显示屏需要数据时，便从sdram中读取数据，发送给显示屏用于显示。这样便可以将读写储存隔离。

低成本的板载sdram的时钟频率也在100MHz以上，数据宽度为16bit,数据带宽最小为100M\*2byte（16bit）=200Mbyte/s。按60HZ，640\*480的图片计算，一帧图片要进行读和写两次操作，通常由摄像头传入的图像为16位宽，那么图像传输的速率为：640 \* 480 \* 2btye(16bit) \* 2 \* 60 = 70Mbyte,可见sdram的带宽足够，满足图像实时显示要求。

2，为什么要用fifo做数据的临时缓存？

为什么不能将数据直接存入sdram，还要经过fifo临时缓存呢?

一是由于跨时钟域。摄像头产生的数据的速率和sdram的写入时钟速率不匹配。一般而言，sdram的时钟频率都要大于像素时钟的一到两倍。故需要一个读写双时钟的fifo。将像素时钟接入到fifo的写时钟端口，像素数据有效信号接入到fifo的写使能端口，将有效的数据写入到fifo中。将sdram的时钟接到fifo的读时钟端口，用于从fifo中读取数据。

二是由于sdram的突发传输，要保证连续突发传输时每一个数据都是有效数据。如果直接将摄像头的数据传输到sdram端口 ，在连续传输的128个字节中，像素数据不一定是全部连续有效的数据。只将有效数据储存进fifo中后，从fifo中读出的连续数据一定都是连续的有效数据。

对于读fifo，也是同理，显示屏的驱动时钟和sdram的时钟也存在跨时钟域，而且在突发读取时，也不能保证从sdram中传出的数据正好和显示所需要Dev数据相匹配。故需要有一个双时钟端口的用于读数据的fifo做数据的临时缓存。

3，何时将fifo中的数据写入sdram？

sdram的突发长度设置为128，设置当写fifo中的数据量大于128时，进行一次写突发传输，将fifo中的128个数据搬移到sdram中。图像数据不断从摄像头传入，写进fifo中，每当写入的数据大于预设值，就进行一次写突发传输。这个有读写仲裁模块所控制。

4，何时将sdram中的数据读出

显示屏驱动模块要从读fifo中读取数据，设置一个阈值，当读fifo中的数据小于该阈时，便进行一次突发传输，将数据从sdram中临时缓存到fifo中，用于显示驱动的读操作。

5，何为乒乓操作，为何要乒乓操作？

乒乓操作的具体总法为，在sdram中设置两个储存空间，用于储存两张图片。分别用于储存摄像头传来的图像数据，和显示屏读取数据。一帧图片传输完成后，读写区域互换。这样能保证在显示屏显示时，能够显示一张完整的图像。如果利用同一片储存区域来储存图像，当前一帧图像还没有读取显示完成，下一帧图像的数据就将该区域覆盖，那么显示屏上的画面会出现拖影现象，两帧图像会交叠在一起。