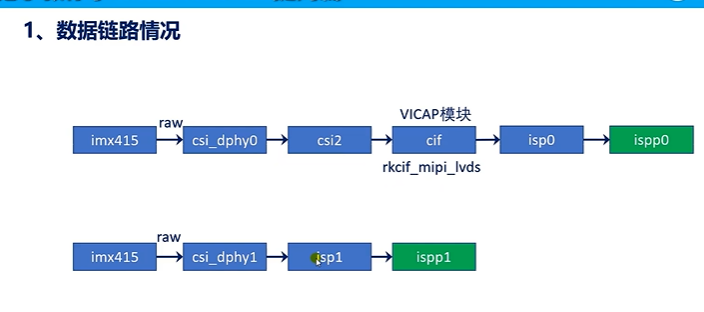


第一节：

简单分析从摄像头的sensor出来到rkmedia的视频输入模块vi模块它的链路情况是什么样的，

第二节：

Rkmedia的视频输入vi模块就是从ispp数据流出节点来获取数据的。所以先来了解ispp数据流输出节点的数据的特点。了解这些节点输出数据的特点。

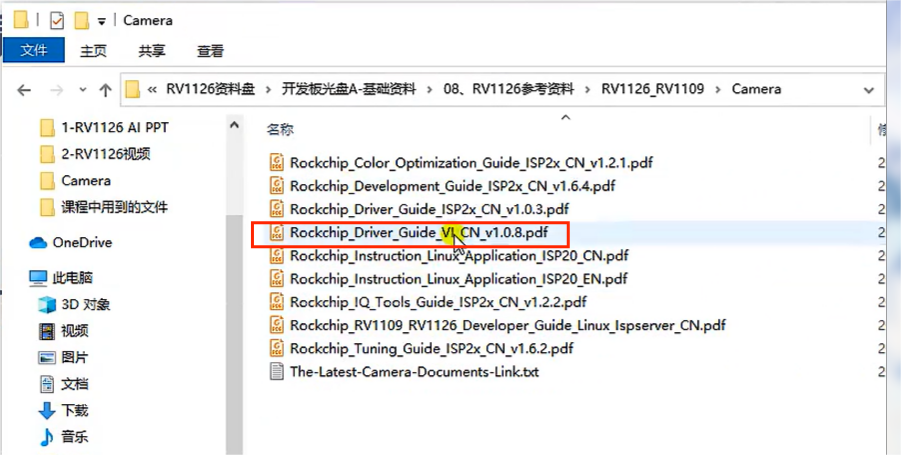


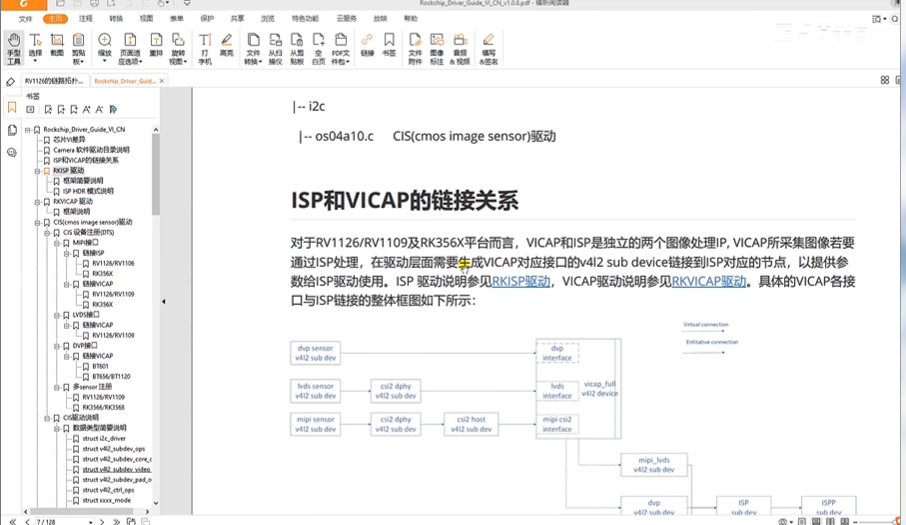
CIF节点也称为VICAP模块节点。

VICAP和ISP模块一样，都是图像处理ip，不过VICAP模块没有对图像进行ISP处理的能力。

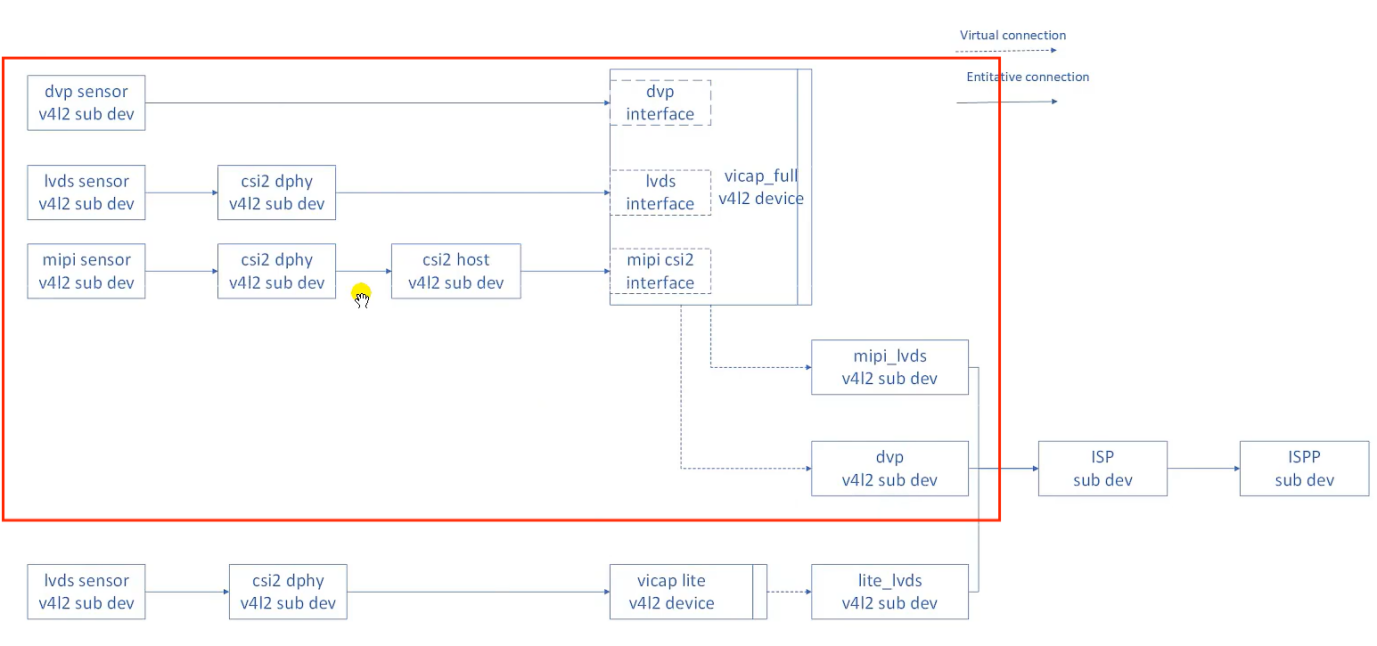
也就是说VICAP模块 与 ISP模块的差别就是：VICAP没有处理图像的能力。但ISP可以对图像进行优化。

由于VICAP不能处理图像，所以如果摄像头输出的是raw数据，那么VICAP节点获取到的数据就是raw数据。即VICAP节点只起到转存数据的作用。它没有ISP功能。





如果VICAP要把数据丢给ISP处理的话：链路如图:



左上角就是v4l2子设备。也就是说：VICAP对应的接口，它先连接到v4l2子设备，然后他才能通过v4l2子设备连接到ISP。

它的开发板配置的IMX415摄像头，输出的就是raw数据，所以获取的数据都是需要经过ISP处理的。

注意：

通过VICAP手册知道:rkvicap\_mipi\_lvds节点实际就是rkcif\_mipi\_lvds节点。

rkvicap\_mipi\_lvds在手册中搜索不到，但rkcif\_mipi\_lvds在手册中可以搜索看到。

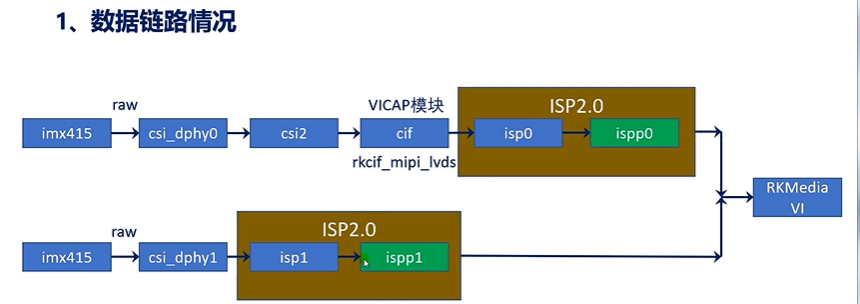
也就是说：

rkvicap\_mipi\_lvds ——实际上是———— rkcif\_mipi\_lvds

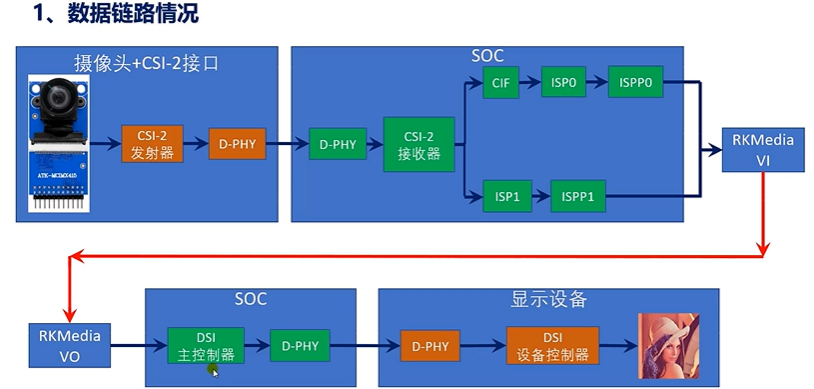
rkvicap ——实际上是———— rkcif

rkvicap\_dvp ——实际上是———— rkcif\_dvp

使用IMX415一定要开启ISP.



需要使用ISP处理raw图像，然后rkmedia再从ISPP对应的Video节点获取数据。

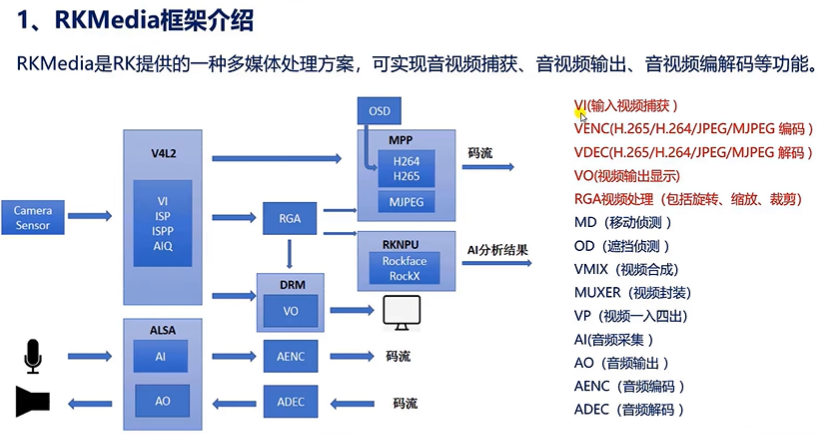


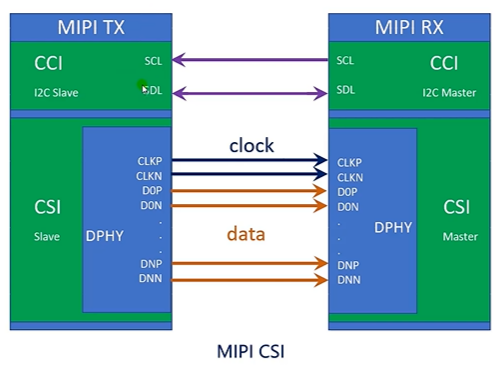
数据链路：

数据从IMX415的sensor出来，经过CSI-2发射器连接到D-PHY物理层，物理连接到RK3568芯片的D-PHY物理层，再连接到CSI-2接收器，再经过右边的那些节点。

由于IMX415出的图像是raw数据，所以要经过ISP处理后，rkmedia模块才能从ISP对应的节点来获取ISP处理后的图像。（IMX415的sensor是发送端，rk3568是接收端），所以左侧称为 CSI-2发射器，右侧的为 CSI-2接收器。

RKMedia的VU模块得到图像后，在内部经过处理后，可以通过DSI来显示。也就是下面这一部分，下面这一部分就是MIPI DSI显示 相关的电路。

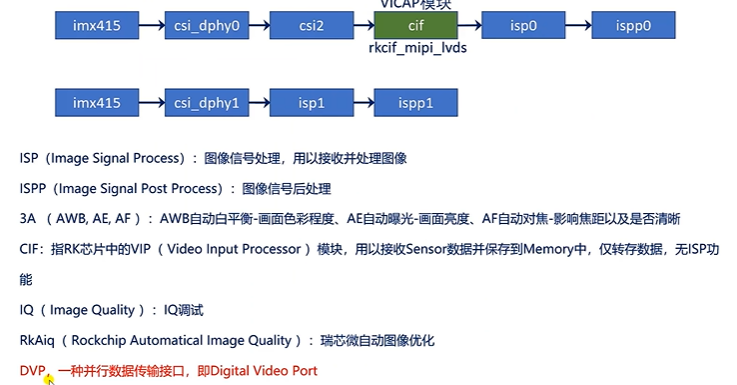
RK3568的DSI主控制器 连接到 D-PHY物理层，然后通过物理层连接到显示设备的D-PHY物理层，再经过DSI设备控制器，最后屏幕就可以将处理的图像显示。



CCI 这里用到了 I2C，I2C主要就是用来控制Sensor工作的，CCI就是摄像头控制接口的意思。它通过I2C控制摄像头的工作。RK3568就是通过I2C对Sensor进行初始化，就是通过I2C配置寄存器还有摄像头的参数的。例如：配置寄存器可以选择开启多少个数据通道等等，主要就是这些控制和配置操作。

CCI主要就是通过I2C对Sensor控制和配置操作。

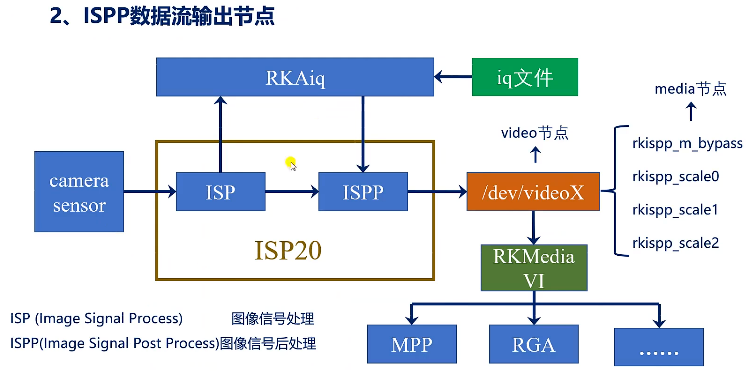
CSI主要就是用来传输图像数据的，这里的D0~DN就是数据通道，可以开启1~4路数据通道，如果摄像头像素达到800w以后，一般就是使用4个数据通道了。



DVP:是一种并行数据传输接口，如OV5640摄像头，即使DVP并口传输。

MIPI CSI接口使用的是串行接口，电路采用的是差分线，（像素大，就用MIPI）

RV1126支持DVP和MIPI接口，MIPI接口就是用到了RV1126的MIPI CSI接口，DVP接口就是用到了CIF接口，不论是DVP还是CIF接口的摄像头，只要输出数据不是raw格式，如：输出的是YUV格式，就可以直接从CIF对应的video节点来获取图像数据，也就是说，可以直接从上图的CIF节点获取图像数据。（CIF只有转存数据的作用）

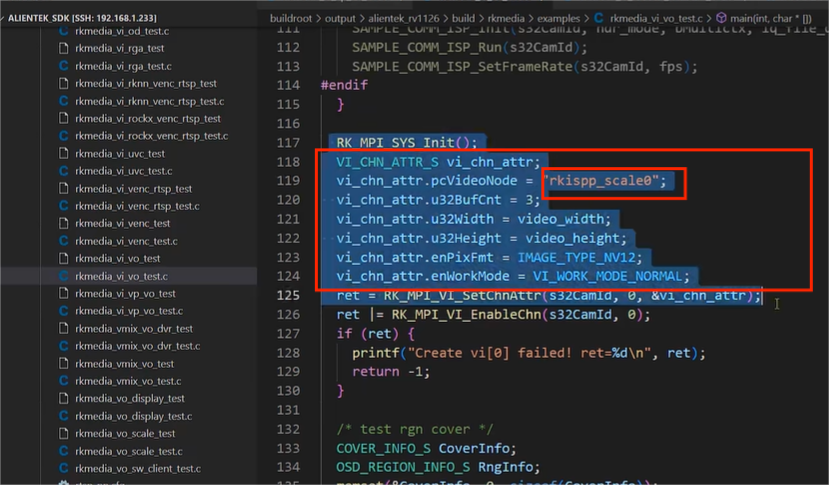


再次分析该图：

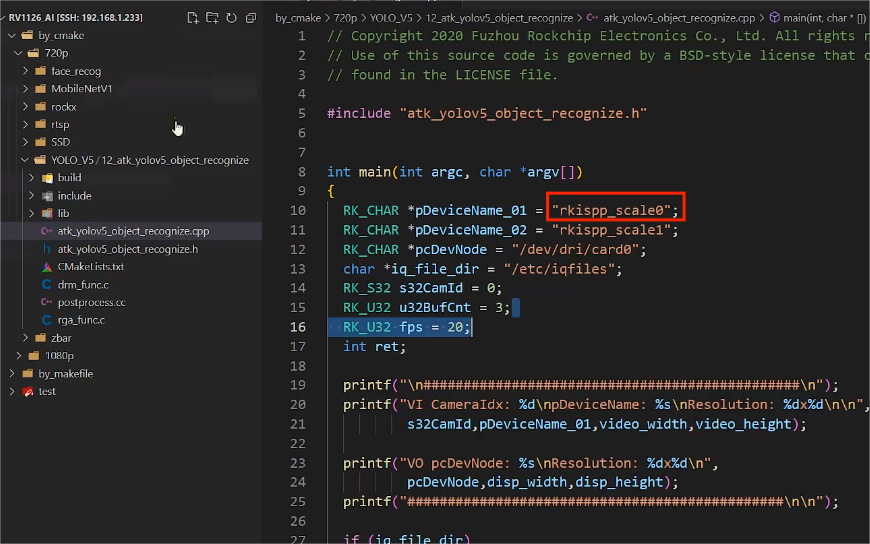
从摄像头的sensor出来经过RV1126的ISP20处理之后，最终RKMedia的vi模块可以从ISPP对应的video节点来获取数据，这些video节点还有对应的media节点，也就是媒体节点。Media节点也就是下面这四路，一般在编写代码的时候，我们调用的就是这个Media节点。而不是video节点。

这四路节点，也成为ISPP输出节点。

可以参考RKMedia例程中的代码:



这里调用的RKMedia的vi模块,调用的就是rkispp\_scale0媒体节点，而不是video节点。（因为video节点内容太多了不好记，而媒体节点只有四个）



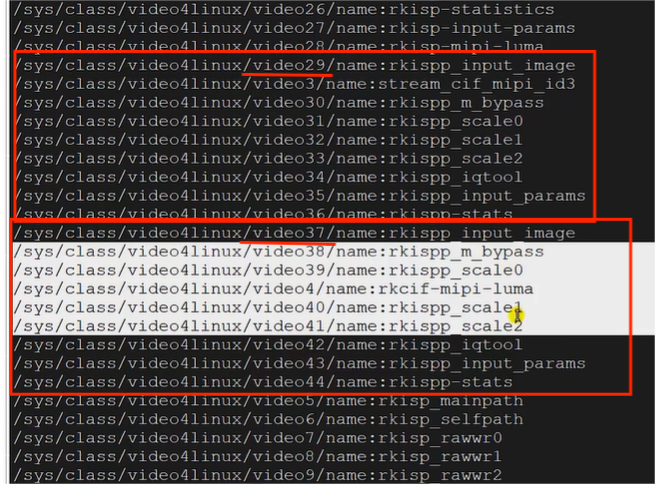


媒体节点对应哪个video节点呢？

通过命令： grep “” /sys/class/video4linux/\*/name

查看。

输出结果如下：

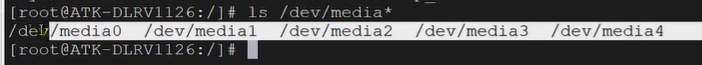


冒号前面的可以看出：媒体节点对应的哪一个video节点。

冒号后面的部分就是媒体节点了。

并且可以看出：媒体节点的名称可以是一样的，但video节点的名称一定是唯一的。之所以是存在两个名称相同的媒体节点，是因为他们属于不同的设备下：

执行 命令：ls /dev/media\* 可以看到：



其系统下有5个媒体设备。

由PPT图：



又知：ISPP输出节点名有四个。右边红色的是MIPI CSI0接口，紫色对应的是开发板MIPI CSI1接口。

也就是说，

开发板的MIPI CSI0接口的 ISPP输出节点(rkispp\_m\_bypass)对应的video节点就是 /dev/video30.

MIPI CSI1接口的 该ISPP节点对应的是 /dev/video38.

对于rkispp\_m\_bypass节点，他比较特殊，它不支持设置分辨率，输入是多大的图像，就只能输出多大，并且这一路输出的图像不支持对图像的缩放操作。也就是说RKMedia不能对图像进行缩放，这一路ISP给ISPP输出多大的图像，RKMedia就只能接收多大的图像。