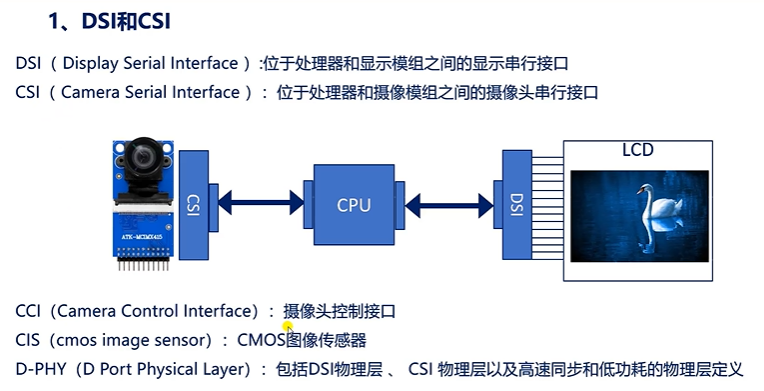
本节了解MIPI接口，再结合Rv1126设备树分析，从sensor出来，到RKMeida之间数据的走向。



1.了解DSI和CSI：



DSI就是针对显示应用的显示接口，位于处理器和显示模组之间的高速串行接口。

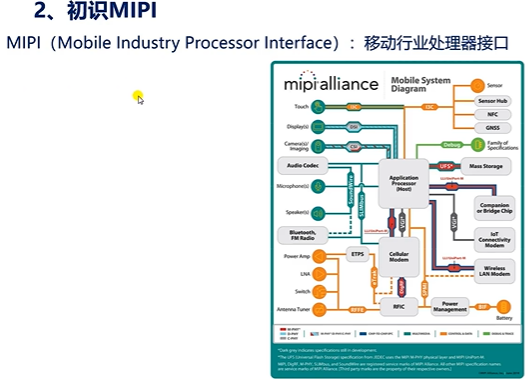
CSI就是摄像头接口，位于处理器和摄像头模组之间的高速串行接口。

CCI就是摄像头控制接口，是传输控制信号的通道接口

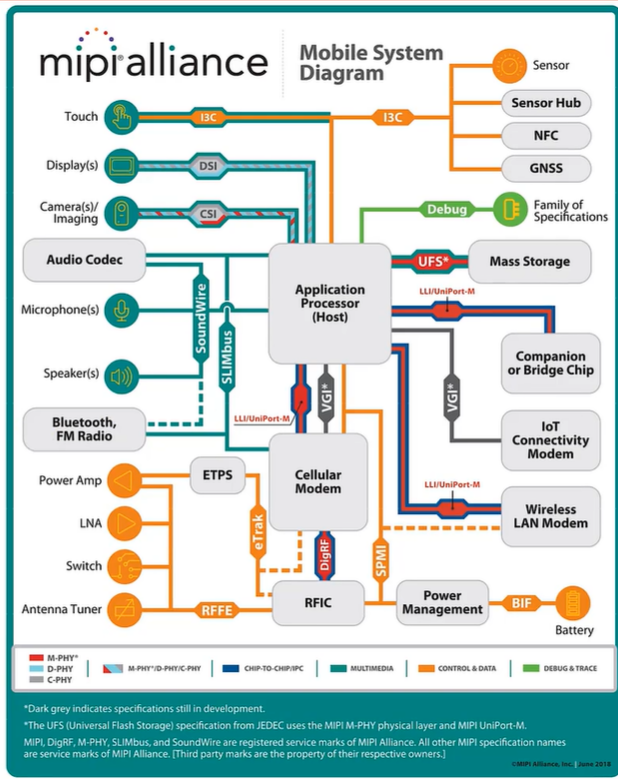
CIS就是CMOS图像传感器

1. PHY就是端口物理层的意思，包括DSI物理层、CSI物理层以及高速同步和低功耗的物理层定义。

MIPI:



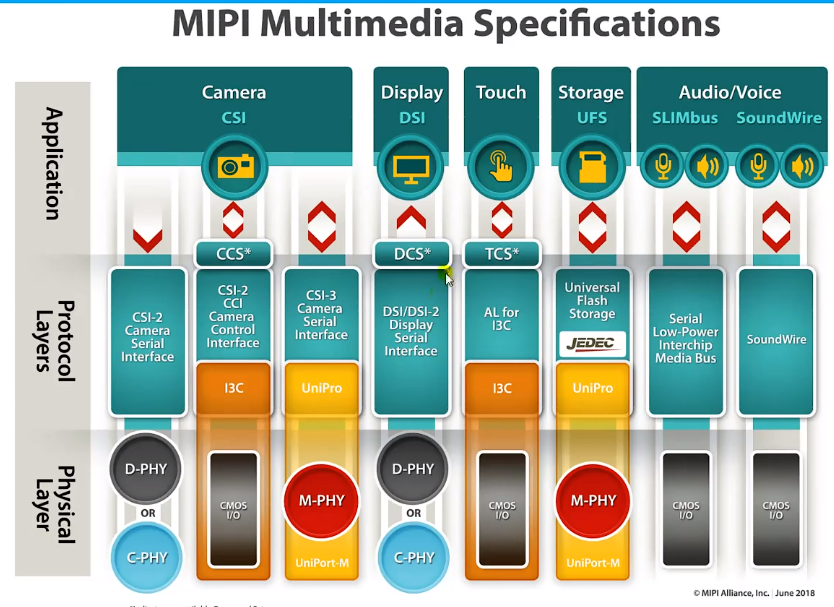
MIPI框架图：



MIPI协议里面也包含许多协议：I3C、I2C、DSI、CSI、RFIC、BIF等协议。

这些协议都是MIPI协议族。

分层形式框架图：



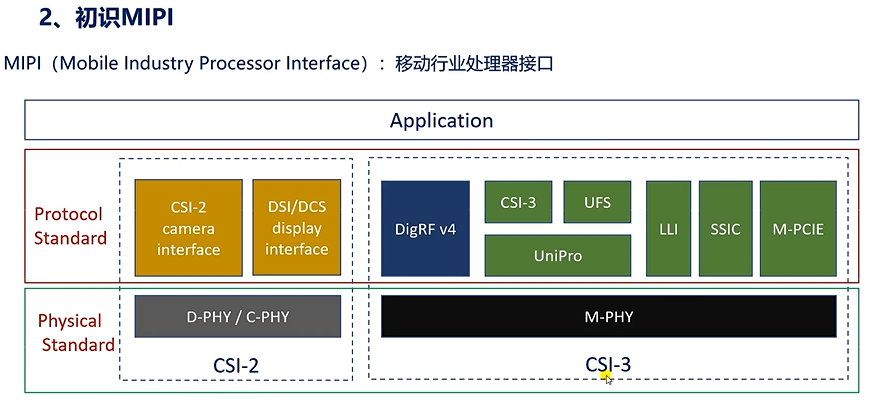
可见分为三层：物理层、协议层、应用层。

我们主要了解MIPI的CSI协议。

CSI-2使用的是D-PHY （或C-PHY)

CSI-3使用的是M-PHY

更简单的框图：



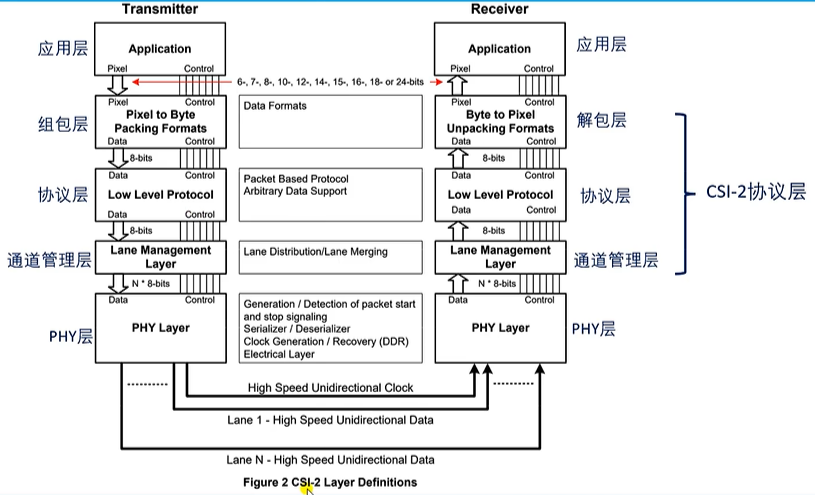
图中，左边的就是CSI-2标准，右边的是CSI-3标准。

在物理层之上就是协议层了，不同版本的CSI是不一样的。我们主要看CSI-2

如果是显示设备就使用DSI协议。如果是摄像头设备就使用CSI-2协议。

在MIPI协议中，不论是底层的物理层还是中间的协议层其内部都是进行了分层设计的。

MIPI中CSI-2协议的层级图：



应用层：主要描述了上层数据流中的数据编码情况，在CSI-2规范中，这个层的作用就是描述了从像素值到字节数据的映射。

组包层（像素字节转换层）：主要用于数据格式的处理。

对于发送端：像素字节转换层主要就是对图像数据进行打包操作，在发送前将数据按照一定的次序。切割成8bit的数据，然后将像素打包成字节，所以在发送端，像素字节转换层也可称为组包层或打包层。

对于接收端：主要就是对数据进行解包操作。接受的时候就是将字节解压为像素。所以在接收端我们称为解包层。

协议层（底层协议层）：

对于发送端：协议层就是根据数据类型添加包头或者包尾等信息。使得数据符合通信协议要求，然后将这些数据流发送给通道管理层。

对于接收端：协议层就是将接收到的数据流去掉包头和包尾，再将数据流发送给上层的解包层。

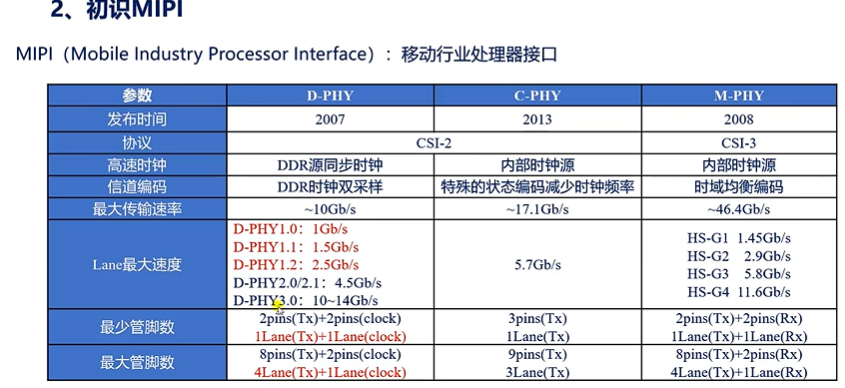
通道管理层（lane管理层）：lane就是MIPI通道中的信息传输单元，数据的收发就是通过lane来完成的。

对于发送端：通道管理层的作用是 根据应用程序的带宽要求选择合适的通道数量，然后将数据流按照一定的顺序和要求发送给PHY物理层。

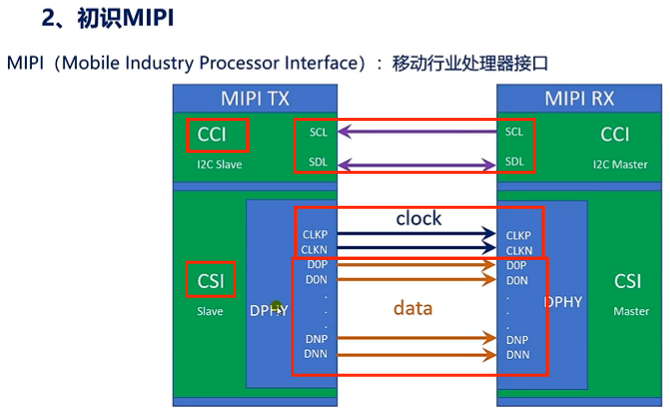
对于接收端：通道管理层就是从PHY物理层就收字节，并将他们合并到一个重新组合的数据流中，从而恢复原始数据流的序列，再输出给协议层。

PHY物理层：指定了传输介质的特性，信号的电气参数、以及时钟和数据通道之间的时序等关系。

各层的详细实现就需要查看MIPI的手册。



MIPI的CSI-2的连接图（D-PHY作为物理层时，camera和SOC之间的硬件连接）：



CSI-2的连接图主要包含两个部分：

CCI和CSI

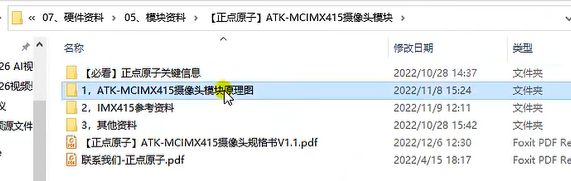
MIPI CSI-2

一般有一对I2C通信引脚。也就是CCI这里。

还有一对差分时钟引脚：CLKP -- CLKP CLKN -- CLKN

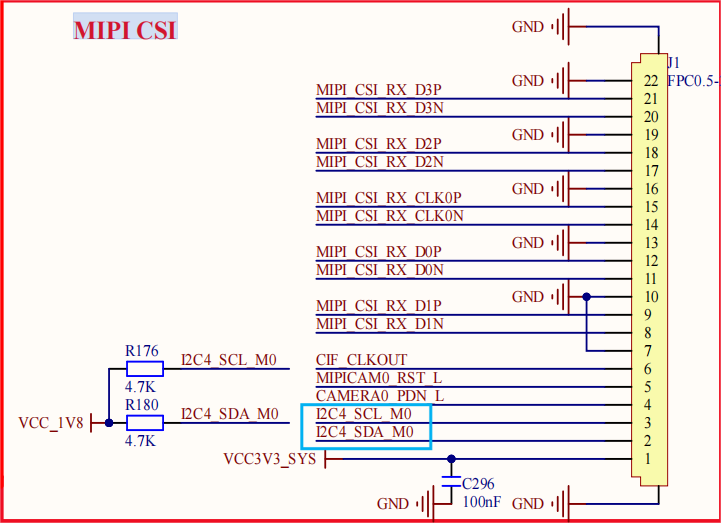
以及1~4对MIPI差分数据信号引脚 D0P~DNN，这里D0P和D0N就是一对差分数据引脚。DNP和DNN也是一对差分数据引脚。

DPHY最大支持4个数据通道（最大管脚数），所以DNN的中间的N最大就是4。一般，摄像头的像素越高，使用的数据通道就会越多。



/\*

通过这里面的原理图我们可以看到：



在设备树目录下：/kernel/arch/arm64/boot/dts/rockchip

有一个rk3568-atk-atompi-ca1.dts文件：

里面显示：

&i2c4 {

status = "okay";

imx335: imx335@1a {

status = "okay";

compatible = "sony,imx335";

reg = <0x1a>;

clocks = <&cru CLK\_CIF\_OUT>;

clock-names = "xvclk";

power-domains = <&power RK3568\_PD\_VI>;

pinctrl-names = "rockchip,camera\_default";

pinctrl-0 = <&cif\_clk>;

reset-gpios = <&gpio4 RK\_PB4 GPIO\_ACTIVE\_LOW>;

pwdn-gpios = <&gpio4 RK\_PB5 GPIO\_ACTIVE\_HIGH>;

rockchip,camera-module-index = <0>;

rockchip,camera-module-facing = "back";

rockchip,camera-module-name = "MTV4-IR-E-P";

rockchip,camera-module-lens-name = "40IRC-4MP-F16";

port {

imx335\_out: endpoint {

remote-endpoint = <&mipi\_in\_ucam0>;

data-lanes = <1 2 3 4>;

};

};

};

imx415: imx415@1a {

status = "okay";

compatible = "sony,imx415";

reg = <0x1a>;

clocks = <&cru CLK\_CIF\_OUT>;

clock-names = "xvclk";

power-domains = <&power RK3568\_PD\_VI>;

pinctrl-names = "rockchip,camera\_default";

pinctrl-0 = <&cif\_clk>;

reset-gpios = <&gpio4 RK\_PB4 GPIO\_ACTIVE\_LOW>;

power-gpios = <&gpio4 RK\_PB5 GPIO\_ACTIVE\_HIGH>;

rockchip,camera-module-index = <0>;

rockchip,camera-module-facing = "back";

rockchip,camera-module-name = "CMK-OT1522-FG3";

rockchip,camera-module-lens-name = "CS-P1150-IRC-8M-FAU";

port {

imx415\_out: endpoint {

remote-endpoint = <&mipi\_in\_ucam1>;

data-lanes = <1 2 3 4>;

};

};

另外在开发板中执行对应的命令来查看地址：

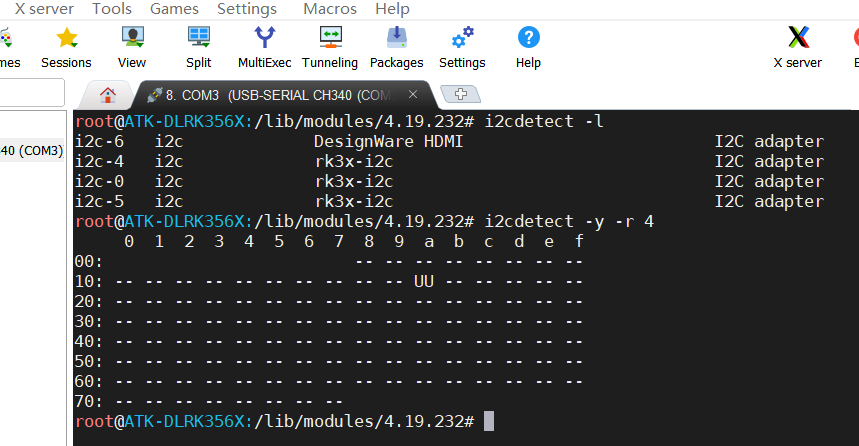
i2cdetect -l



执行命令后就可以看到当前开发板占用了几（3）条I2c总线。分别是I2C0、I2C1、I2C5。

然后我们可以执行命令：

i2cdetect -y -r 5 查看I2C5挂载了哪些设备



（由图可知i2c5挂载了设备地址为0x1a的设备，这个设备地址是imx335或imx415的地址。UU表示已经注册了一个设备了。这个地址表示有设备）

分别查看每个I2C挂载了多少个设备。注意：每个连接到总线上的器件都有唯一的地址。也就是说从机的地址是唯一的。

由上面可知MIPI CSI占用了I2C4，

可以看出摄像头MIPI CSI使用的是I2C4端口。

\*/

总共有4对差分线。并且下面使用了一组I2c。I2C充当了CCI的角色。



由规格书可见I2C的从机地址（器件地址）是0x1a.

也就是说，摄像头的从机地址(器件地址)可以在摄像头的器件规格书中找到。当然也可以在设备树中找到：

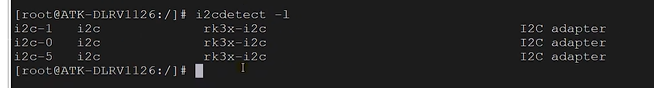
设备树就在该目录下：

IMG_256

通过底板原理图，以及以下命令

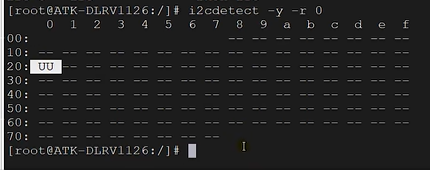
i2cdetect -l

查看当前开发板占用的i2c总线：i2c0、1、5.



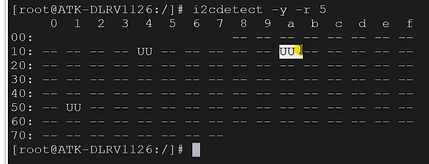
然后执行命令：分别看每个I2C挂载了多少个设备：（从机地址是唯一的）

i2cdetect -y -r 0 查看i2c-0挂载设备。



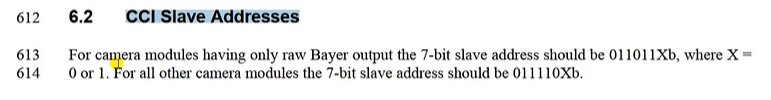
可见其挂载了一个设备。

我们看谁摄像头的i2c的挂载信息：



它挂载了3个设备：0x1a的地址就是IMX415/IMX335的器件地址。UU表示在这个I2C下已经注册了一个设备了。UU就表示这个地址被占用了。

由MIPI手册：



如果摄像头模组输出的图像格式是raw或raw Bayer格式，那么其7位从机地址就是 011011Xb,x为0或1。

00011010 ：0x1a

其他的摄像头从机地址应为：011110Xb.

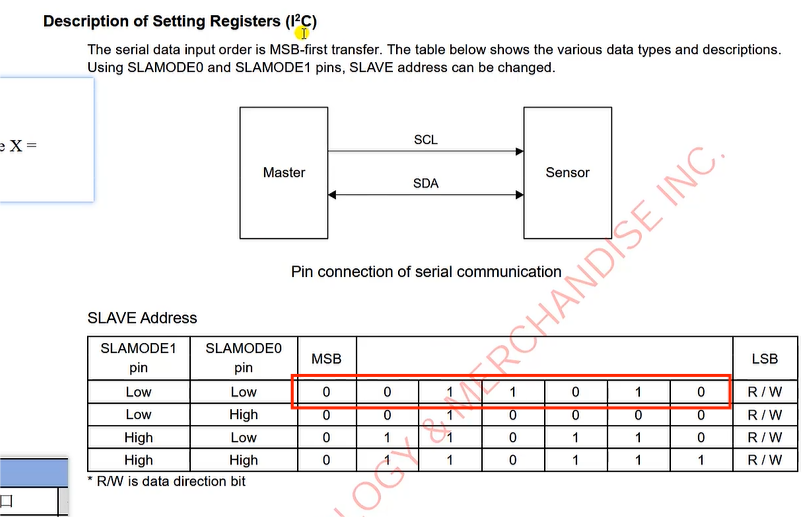
011011Xb,x为0或1。

00011010 ：0x1a

我们分析知：

输出raw格式的摄像头 的从机地址应该是011011Xb，但实际imx415的从机地址却是00011010。与说明的不一样。

查看imx415/imx335的datasheet（"C:\Users\zhongqing\Desktop\笔记\IMX415\_Datasheet\_E19504(产品信息).pdf"）知:

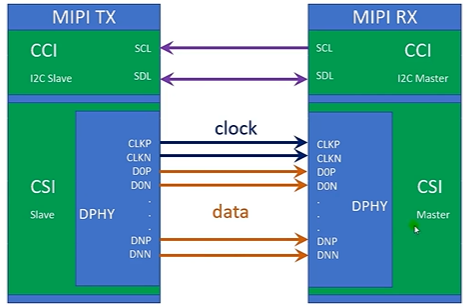


IMX415/335的从机地址是可以（通过SLAMODE1和SLAMODE0管脚）更改的。也就是说摄像头的从机地址可以通过寄存器修改。

我们使用的是第一个：0x1a = 0 0011010

因此，CCI就是一个由双线组成的，双向的，工作模式为半双工的 串行接口。支持7位从机寻址。其作用就是控制通信:通过从机地址 确定与那个从机 进行通信。

CSI部分：



如图：

CSI-2接收器 就是 Master主设备这边。

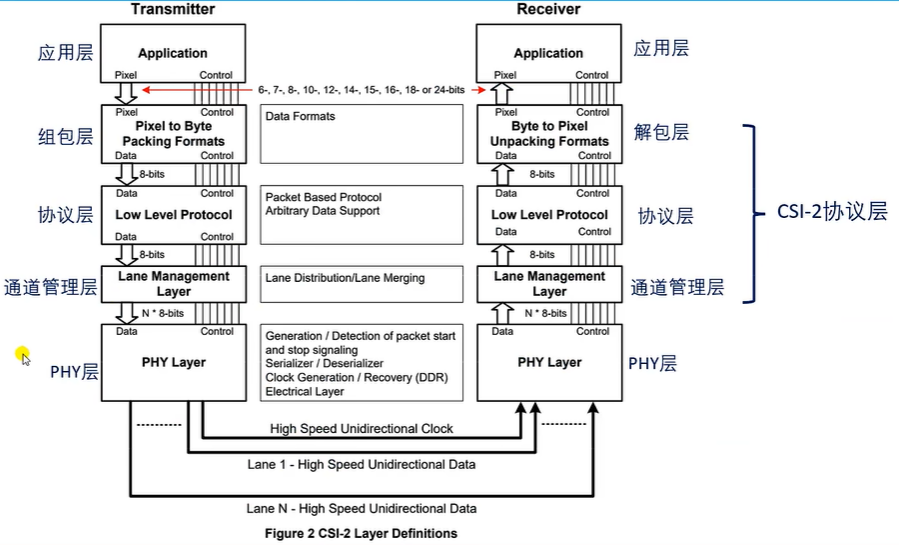
CSI-2发送器 就是 Slave 从设备这边。

当MIPI CSI的主设备与外部的模组进行控制信号交互的时候，在MIPI的发送端，使用的就是IIC的从机地址。

MIPI CSI的发送端就是从设备，

MIPI CSI的接收端就是主设备。就可以通过IIC来配置MIPI CSI发送端的内部的寄存器，从而配置输出数据时需要的通道数等信息。

由之前可知：CSI分为5层：实现原理详细见MIPI手册

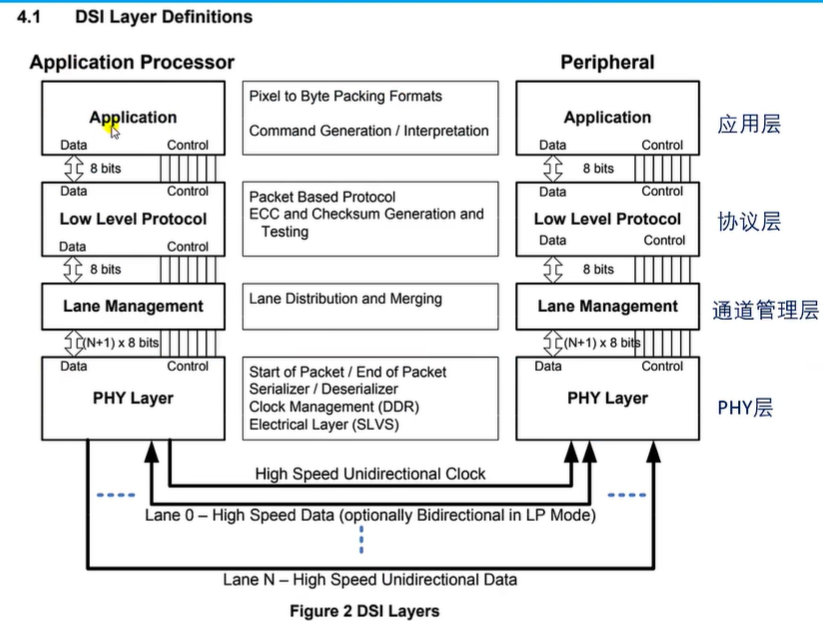
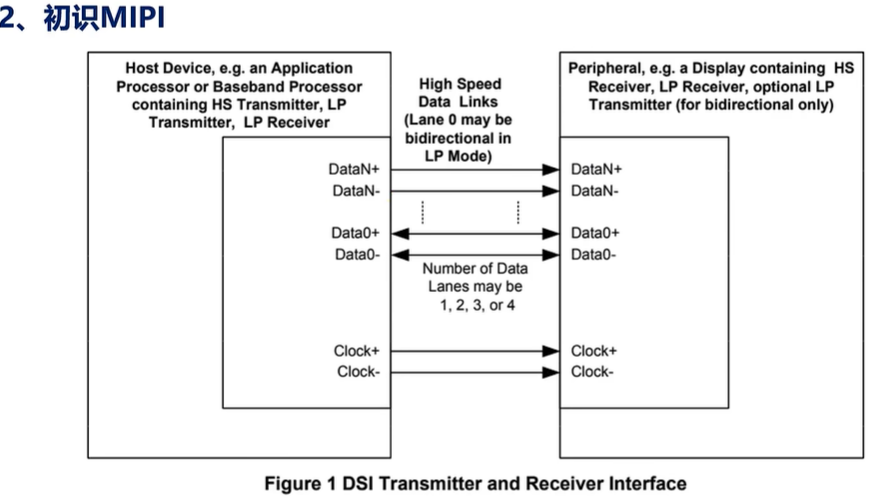


————————————————————————————————

MIPI DSI的设计也是采用了分层的设计，大概分为4层：

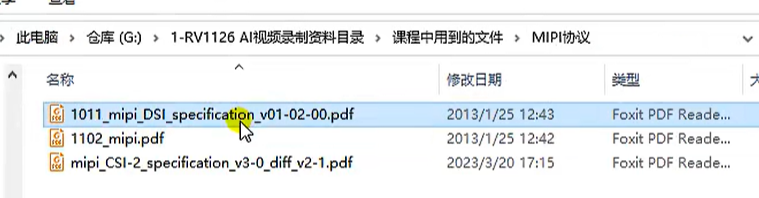
电路连接图：

与CSI-2相比,它没有CCI控制端。没有IIC.



从上到下也是：应用层、协议层、通道层、PHY层。

具体每层作用见MIPI手册。

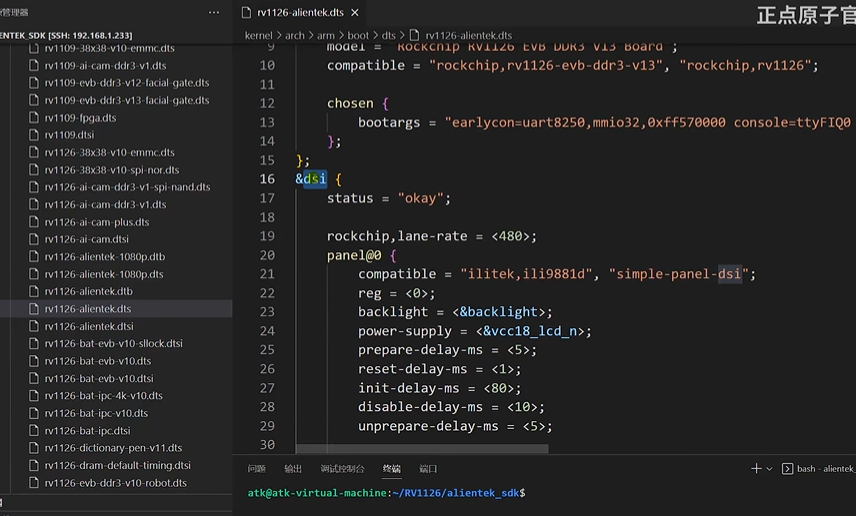


打开设备树，查看从摄像头到rkmedia的数据流是怎么流的：

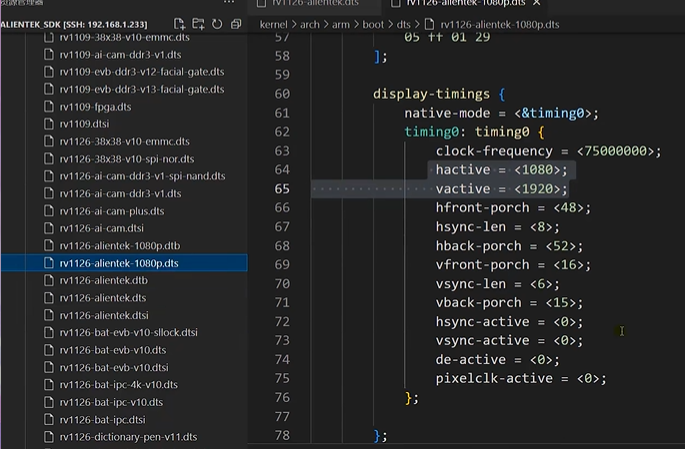
这里就是DSI的配置了：

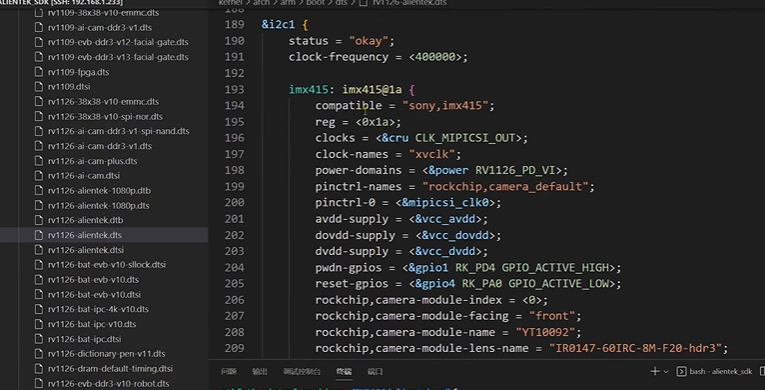
屏幕显示用的就是MIPI DSI。

这个节点下的配置就是MIPI720的配置。

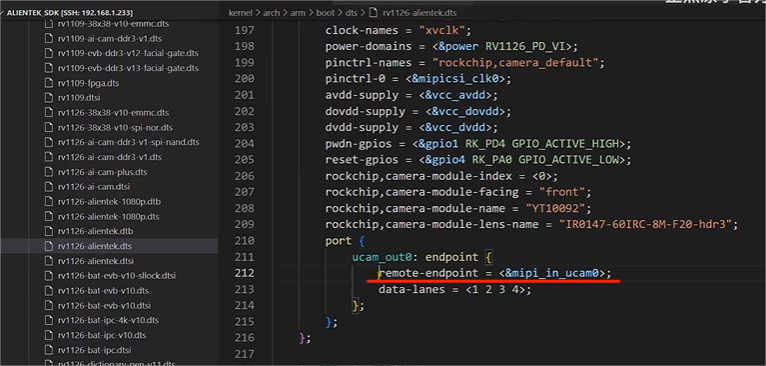


如果是1080p屏幕就是用的：



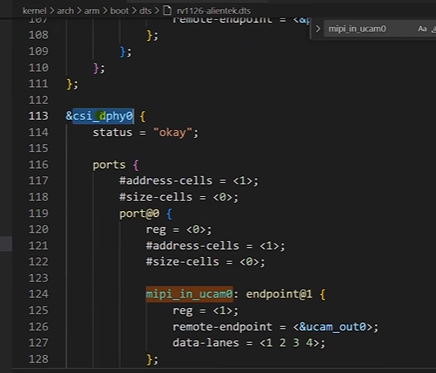


Imx415连接到了remote-endpoint，远端：&mipi\_in\_ucam0:

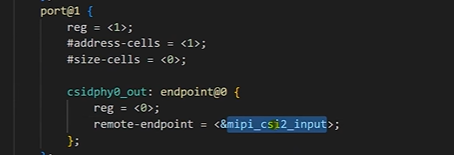


通过搜索mipi\_in\_ucam0看到：

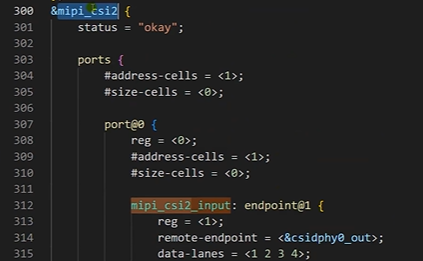
它对应的节点为：CSI\_dphy0



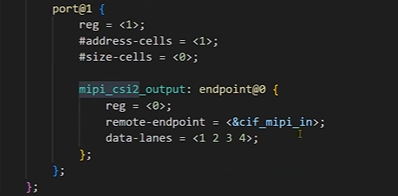
摄像头是端口1的：所以看port@1:



所以又连接到&mipi\_csi2\_input。搜索知道：他又属于：mipi\_csi2节点。



mipi\_csi2节点连接到了：cif\_mipi\_in。



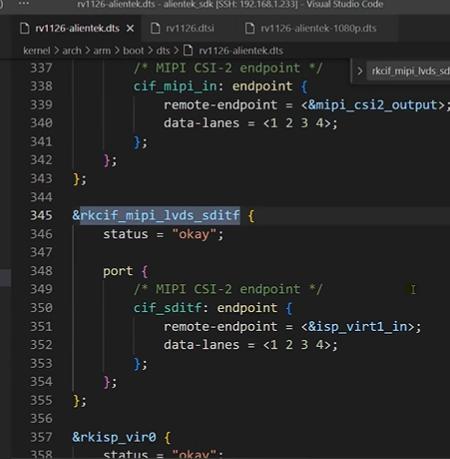
cif\_mipi\_in属于rkcif\_mipi\_lvds节点:



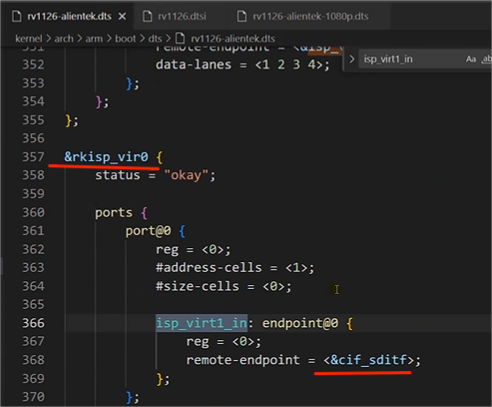
在设备树中搜索：rkcif\_mipi\_lvds，发现他和rkcif\_mipi\_lvds\_sditf关联，



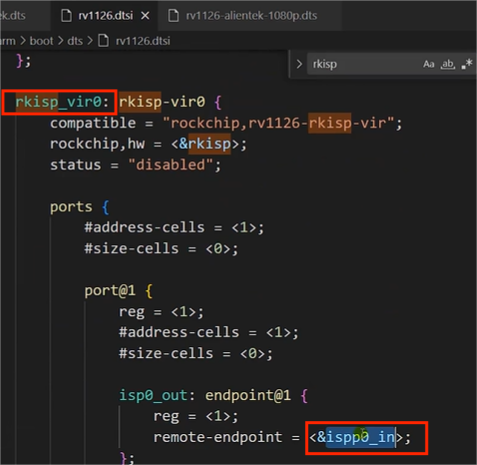
搜索：rkcif\_mipi\_lvds\_sditf得：它连接到：isp\_virt1\_in.



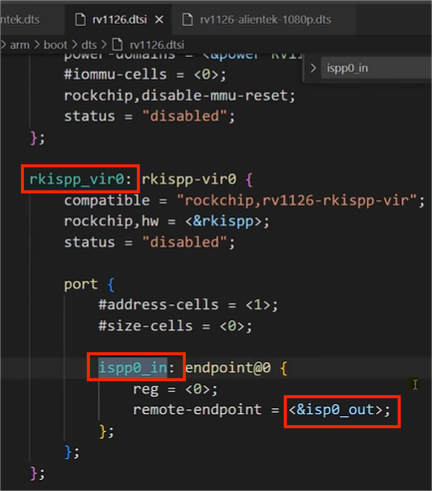
isp\_virt1\_in属于rkisp\_vir0节点，然后又连接到：cif\_sditf.



在设备树搜索：rkisp\_vir0节点:



再搜索：ispp0\_in:属于rkispp\_vir0节点，连接isp0\_out.



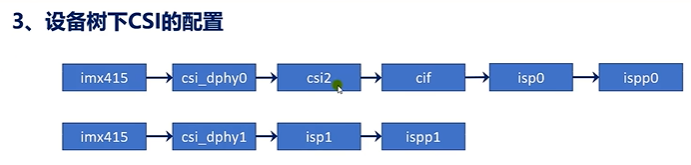
这就是摄像头连接到I2C1时设备节点的连接情况：

Imx415/imx335(I2C1) --- CSI\_dphy0 --- mipi\_csi2 --- rkcif\_mipi\_lvds --- 关联rkcif\_mipi\_lvds\_sditf --- rkisp\_vir0 --- rkispp\_vir0

同理：

Imx415/imx335(I2C5) --- CSI\_dphy1 --- rkisp\_vir1 --- rkispp\_vir1

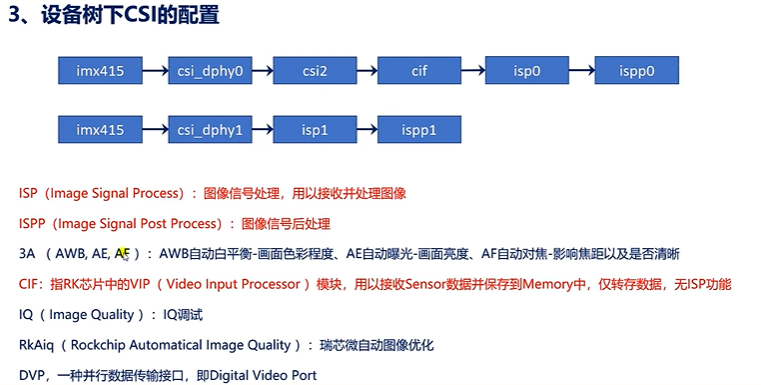
这里的意思是不是说 摄像头模组在设备树上相当于一个I2C从设备吗？CSI接口反而是I2C从设备要附带配的的东西而不是作为一个独立设备来配置？



由图知：

对于i2c1从摄像头sensor出来，连接到了csi\_dphy0,最后连接到ispp0端。Ispp0，rkmedia就是从ispp0获取数据。

对于i2c5从摄像头sensor出来,最终连接到ispp1，rkmedia最终就是从ispp1获取数据的。



本节课大概知道数据流是怎么流的，

下节课具体分析重要节点的数据流的特点 以及 rkmedia从哪些节点获取数据流，以及数据流的特点。并简单介绍v4l2怎么从节点中抓取图片。