

基于 TDA 处理器的 360 度全景实现 YUV422 输出的方案

Marvin Liang

中国汽车处理器技术支持团队

ABSTRACT

在设计之初，TI Jacinto/TDA 家族汽车处理器的 DSS(显示控制子系统)仅支持 RGB 格式的视频输出和非标准的内嵌同步 YUV422 输出 (BT.656)。这个技术限制让汽车 OEM 厂家在连接车载导航系统和基于 TI Jacinto/TDA 芯片的 ADAS 系统时不得不考虑成本较高的 RGB 传输 LVDS 芯片，特别是当某些车厂希望在高低端车型用同一套车载导航系统兼容传统的 RVC 功能和高级的 SRV 功能时，这个技术限制为车载导航端的硬件系统兼容性设计带来较大麻烦。本文基于 TI Jacinto/TDA 家族处理器，提出了一个输出标准外同步 YUV422 信号的方案，并分别论述了其在 TI VISION SDK 和 PSDKLA 上的实现。

Contents

1	YUV422 视频输出的需求说明	2
2	YUV422 格式以及输出时序	2
	2.1 内嵌同步信号的视频输出	2
	2.2 外同步信号的视频输出	3
3	实现外同步信号 YUV422 输出的方案	4
	3.1 Bypass DSS VID pipeline 内部的所有功能模块	4
	3.2 YUV422 的时分复用传输	5
4	在全景中应用 YUV422 输出的方案	6
	4.1 YUV422 输出的全景应用在 VISION SDK 上的实现	7
	4.2 YUV422 输出的全景应用在 PSDKLA 的实现	8
	4.3 YUV422 输出的验证	9
	Code	10

Figures

Figure 1.	在全景系统中输出 YUV422 的优势	2
Figure 2.	YUV422 的内嵌同步格式 (BT.656)	3
Figure 3.	外同步信号的视频输出时序	3
Figure 4.	通过 DSS VIDx pipeline 实现比特精准的 YUV422 输出	4
Figure 5.	时分复用输出 YUV422	5
Figure 6.	YUV422 输出在全景中的应用	6
Figure 7.	VISION SDK YUV422 输出用例的 chain	8
Figure 8.	YUV422 输出的 DRM 结构	9

1 YUV422 视频输出的需求说明

如 Figure 1.A 所示,当前许多车都已经集成了 RVC 的功能,一般 RVC 的摄像头会通过 FPD LINK 连接到中控导航的系统,这样的应用中,摄像头的输出格式一般为外同步信号的 YUV422。随着汽车工业的发展,越来越多的车型,特别是中高端车型升级 RVC 功能为 360 度全景系统。而对于基于 TI Jacinto/TDA 家族处理器的全景方案,都是输出 RGB888。因为传输视频信号的 FPD LINK 加串器/解串器对一般都是要么支持 YUV 信号的传输(如 TI UB933/UB934 for YUV422),要么支持 RGB 信号的传输(如 TI UB921/UB924 for RGB),这也就要求接入全景或者 RVC 的中控导航系统如果想同时兼容两种接入场景的时候,不得不在软硬件设计的时候考虑同时支持两种解串器。这样的方案无论是对系统的 BOM 成本,软硬件的复杂度来说,都不是一个优化的好方案。

但如果基于 TI Jacinto/TDA 家族处理器的全景方案能够输出 YUV422,如 Figure 1.B 所示。则对于中控导航系统来说,全景系统的接入将如 RVC 接入一样,用比较经济的 YUV 信号加串器/解串器对就可以实现,系统的软硬件设计也将更简单。

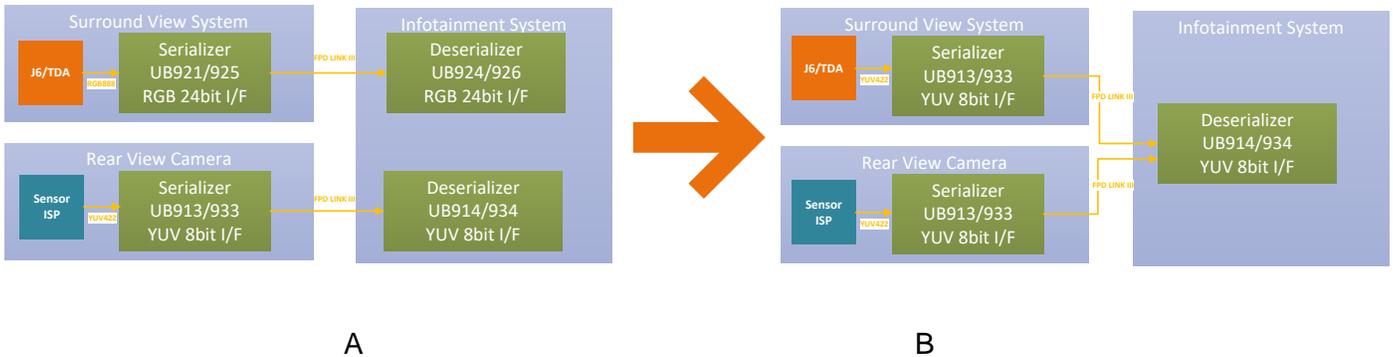


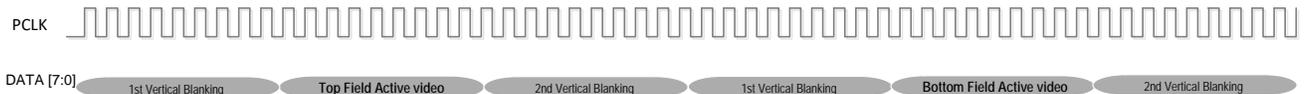
Figure 1. 在全景系统中输出 YUV422 的优势

2 YUV422 格式以及输出时序

根据 Jacinto/TDA 处理器的 TRM 描述,其显示控制子系统支持下面的输出格式:

- A. 外同步信号的 RGB 格式 (RGB888, RGB565, RGB444)
- B. 内嵌同步信号的 YUV422 格式
 - a. 支持 BT656 和 BT1120 标准
 - b. 支持 YUV422 格式 (8 bit 颜色深度)
 - c. 支持 10bit 接口输出 YUV422,实际用到 10 根数据线中的 D[9:2]

2.1 内嵌同步信号的视频输出



A) BT.656 时序和帧格式

EAV Code				Blanking Video				SAV Code				Active Video			
255	0	0	EAV	Cb	Y	Cr	Y	255	0	0	SAV	Cb	Y	Cr	Y
4 Bytes				280 (268) Bytes				4 Bytes				1440 Bytes			

B) BT.656 单行 Active video 的格式

Figure 2. YUV422 的内嵌同步格式 (BT.656)

根据 Figure2.A 和 Figure2.B 可以看出，对于内嵌同步信号的视频输出格式，只需要 PCLK 和数据线，同步信号已经根据 BT 协议的定义打包到了视频数据中了。接收方不再需要 HSYNC 和 VSYNC 信号，而是通过内嵌的列消影，行消影，SAV 和 EAV 字段来解码视频数据。

但是对于 TI Jacinto/TDA 家族的处理器的有一个限制，TRM 里有如下的 CAUTION 段描述：
CAUTION: DISPC supports maximum 256 bytes of horizontal blanking when the LCD outputs are configured in BT modes (bitfield HSW of DISPC_TIMING_H1/2/3 registers is 8 bits wide). This is not compliant with BT.656 and BT.1120 standards, both of which require higher blanking periods. If more than 256 bytes of horizontal blanking are required by the application, then the RGB mode must be used for the LCD outputs.

Blanking Video 在 ITU-R-BT.656 标准中的定义 (如 Figure2.B 是 PAL 制式 280 byte, NTSC 制式 268 byte,所以比较困难在市场上找到能够兼容 TI Jacinto/TDA 家族处理的 blanking video 为 256 byte 的非标准 BT.656 的解码器。

2.2 外同步信号的视频输出

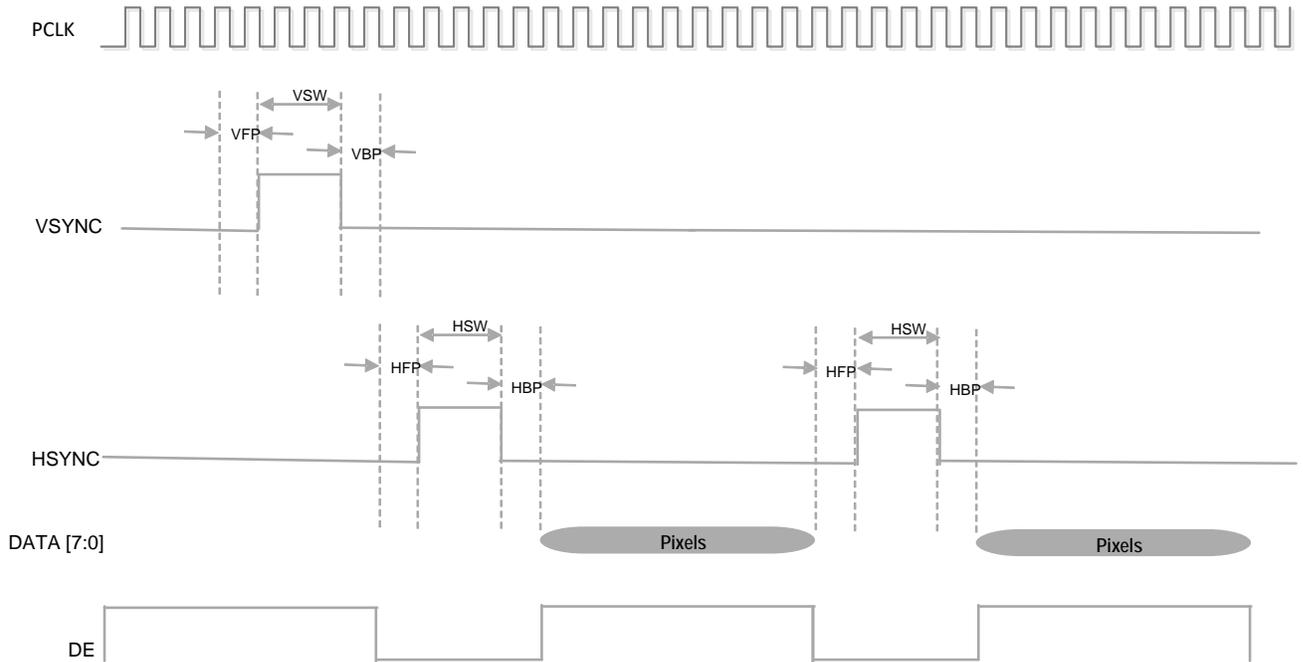


Figure 3. 外同步信号的视频输出时序

如 Figure 3 所示，TI Jacinto/TDA 家族处理器 DSS 输出的外同步信号视频有如下特性：

- A. 支持 HSYNC, VSYNC, FID 和 DE 信号
- B. 当通过 8 位的数据线输出 8bit 像素时, 使用 D[7:0] 数据线
- C. 支持时分复用, 一个像素可以通过多个 PCLK 的时钟周期进行发送

VSYNC: 帧同步信号, 在每帧数据的起始位置跳变。当每一帧数据的所有行都发送完毕的时候, 帧同步信号使能。VSYNC 时序有三个可编程的参数, 分别是 VFP, VBP 和 VSW。

HSYNC: 行同步信号, 在每行数据的起始位置跳变。当每一行数据的所有像素都发送完毕的时候, 行同步信号使能。HSYNC 时序有三个可编程的参数, 分别是 HFP, HBP 和 HSW。

DE: 数据使能信号, 在每行 active video 数据的起始位置跳变。当每一行 active video 数据的所有像素都发送完毕的时候, DE 同步信号使能。DE 的时序依赖于 HFP, HBP 和 HSW 配置

DSS 控制器的寄存器对 HFP, HBP 和 HSW 的配置规则是本身的参数值减 1, 所以对于这些参数而言, 能配置的寄存器最小值是 0, 实际对应的最小参数都是 1。换句话说, 当这些寄存器域配置为 0 的时候, 实际的时序输出 HFP, HBP 和 HSW 都会占一个 PCLK。

因为上述的 BT.656 非标准信号的限制, 本篇 application report 后面所讨论的 YUV422, 都是外同步信号的标准 YUV422 格式。

3 实现外同步信号 YUV422 输出的方案

3.1 Bypass DSS VID pipeline 内部的所有功能模块

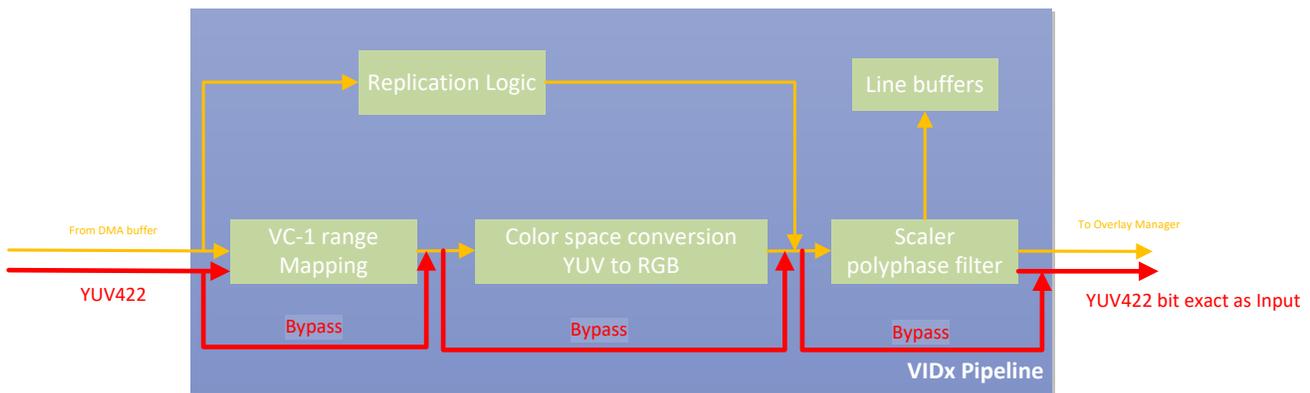


Figure 4. 通过 DSS VIDx pipeline 实现比特精准的 YUV422 输出

首先 DSS 的输出格式不能配置为 YUV422, 因为 DSS 里的颜色空间转换单元并不支持 RGB 到 YUV 的转换。输入的 YUV 格式会在内部被默认转换为 ARGB, 这是因为视频数据经过多个 DSS pipeline 去 overlay 合成的时候, 带 alpha 值的 RGB 更容易做到全局的合成。所以当试图输出 YUV422 时, 无论如何只能将输出格式配置为 16 bit 的 RGB。

再考虑输入格式的配置，如果将 DSS pipeline 的输入格式配置为 YUV422，则 YUV422 视频数据会在 DSS 内部被自动转换为 ARGB 数据。要想输入视频为 YUV422 时 DSS pipeline 不伤害到数据，得到比特精准的输出，必须要让 DSS 认为输入的视频为 RGB 格式。考虑到 YUV422 为每像素 16 bit，RGB565 也是每像素 16 bit，所以最好的办法是在实际输入为 YUV422 格式的情况下，配置该 pipeline 的输入格式为 RGB565，相当于“欺骗”DSS 硬件。

每一个 video pipeline 内部的处理单元其实都有一个 bypass 的路径（如 Figure 4 所示），bypass 后该处理单元不会改变输入的数据。当输入的数据格式为 RGB565 时，VC-1 range mapping 和颜色空间转换单元就会被自动 bypass。当输入该 video pipeline 的视频分辨率（宽和高）和输出该 video pipeline 的视频分辨率被设置为一致时，scaler 单元也被自动 bypass。下面是一组核心寄存器的配置，按照这样的配置，输入 video pipeline 的数据，将会被比特精准的输出到下一级 overlay 模块。

而 overlay 模块如果仅有一个 pipeline 做输入的时候，其全局 overlay 合成不会生效。而禁止像素级 overlay 合成只需要禁止 color key 就可以了。

DSS 相关寄存器配置：

```
DISPC_VID1_ATTRIBUTES.FORMAT = 0x6 //RGB16-565
DISPC_VID1_ATTRIBUTES.RESIZEENABLE = 0x0 //Disable both horizontal and vertical resize
processing
DISPC_VID1_SIZE.SIZEX == DISPC_VID1_PICTURE_SIZE.MEMSIZEX
DISPC_VID1_SIZE.SIZEY == DISPC_VID1_PICTURE_SIZE.MEMSIZEY
DISPC_CONFIG1.TCKLCDENABLE = 0 //Disable the transparency color key for the LCD
DISPC_CONFIG1.CPR = 0 //Color Phase Rotation Disabled
```

3.2 YUV422 的时分复用传输

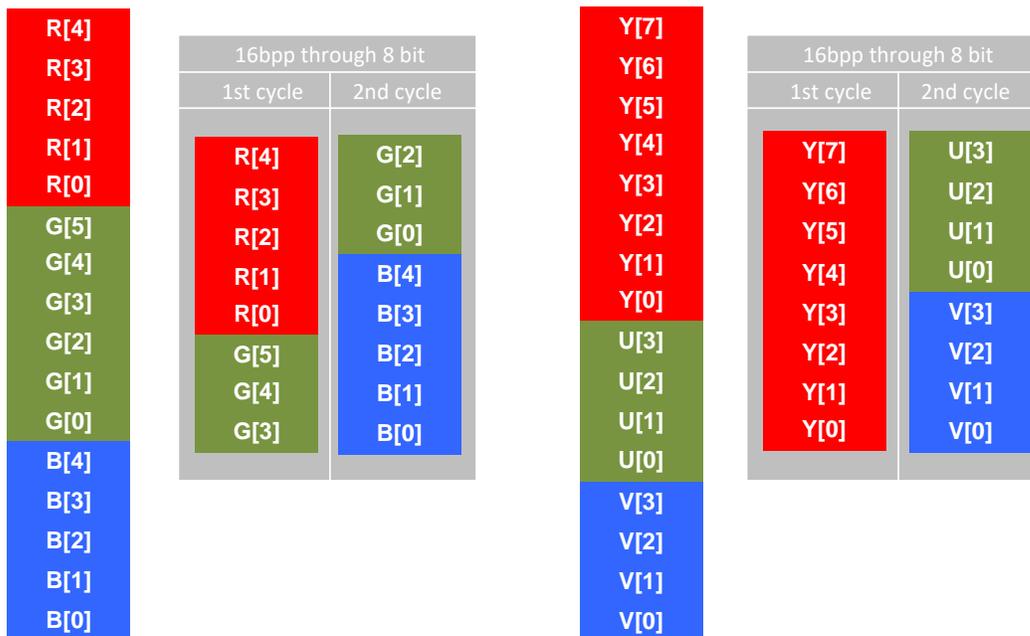


Figure 5. 时分复用输出 YUV422

因为 RGB565 和 YUV422 都是一个像素 16bit，所以当通过 8 位数据线的接口来传递的时候，需要拆分成两个 8 bit，每个 8bit 分别在一个 PCLK 周期内传输完成。Figure 5 指示出了用 TDM 模式通过 8 位数据线传输一个 YUV22 像素的时候，总共需要 2 个 PCLK 周期，第一个 PCLK 周期传输的是 8 bit Y 分量，第二个 PCLK 周期传输的是 UV 分量。下面是对应 YUV422 输出的 TDM 相关寄存器配置示例。

DSS 相关寄存器配置:

```
DISPC_VP1_CONTROL.TDMENABLE = 0x1           // TDM enabled
DISPC_VP1_CONTROL.TDMPARALLELMODE = 0x0     //8-bit parallel output interface selected
DISPC_VP1_CONTROL.TDMCYCLEFORMAT = 0x2     //2 cycles for 1 pixel
DISPC_DATA1_CYCLE1 = 0x8                    //8 bit for 1st cycle
DISPC_DATA1_CYCLE2 = 0x8                    //8 bit for 2nd cycle
DISPC_DATA1_CYCLE3 = 0x0
```

4 在全景中应用 YUV422 输出的方案

考虑典型的 360 度全景应用的显示控制部分, Figure 6 给出了一个系统级的解决方案如何从 RGB888 输出切换到 YUV422 输出. 本例子基于 TI J6 Entry/DRA71x 芯片, 注意该系列芯片没有 VOUT1 接口.

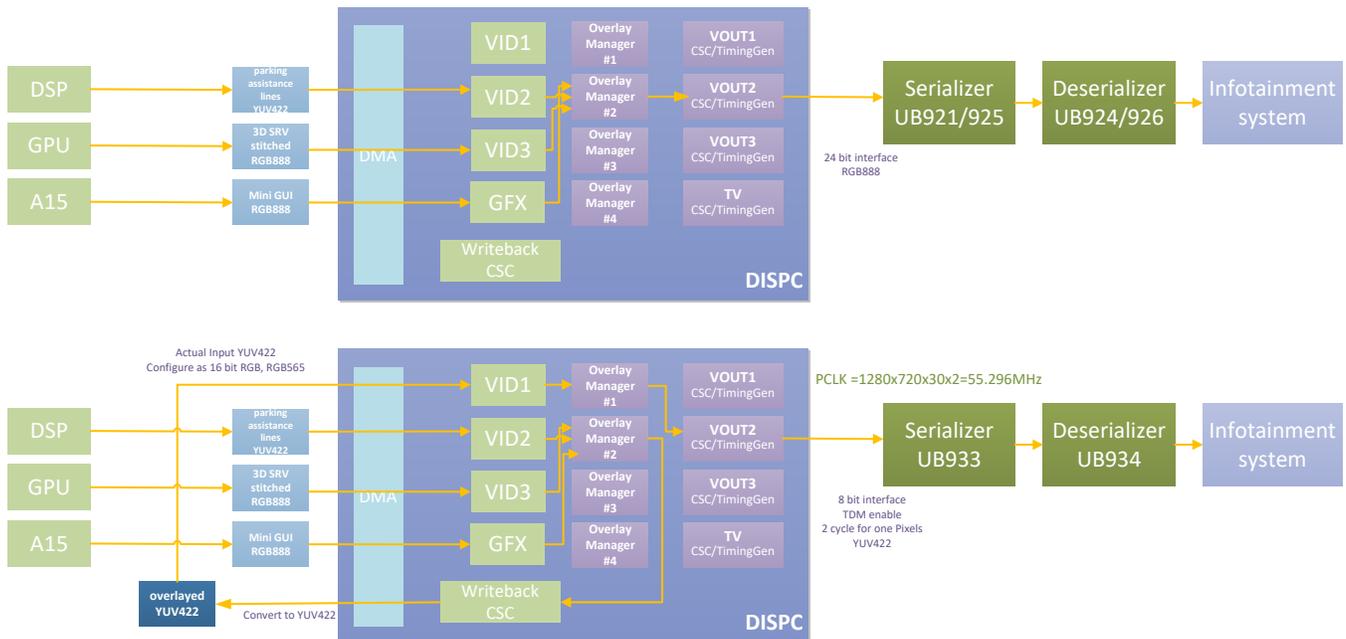


Figure 6. YUV422 输出在全景中的应用

DSP 算法计算并输出 YUV422 格式倒车辅助线 到 DSS VID2 pipeline

GPU 输出合成后的 RGB888 格式 360 度 3D 全景画面到 DSS VID3 pipeline

A15 上的 MinuGUI 绘制 RGB888 格式的简单 UI 到 DSS GFX pipeline

在以前的应用场景中如 Figure 6.1， overlay 管理器会将这个三个 pipeline 上的各个图层按照配置的 alpha 值和 Z order 合成为 RGB888 格式的视频。而为了输出 YUV422，我们必须先要得到 YUV422 格式的合成视频，这个过程不需要 CPU core 参与用软件进行转换，利用 DSS 的 writeback pipeline，可以将原来合成后的 RGB888 格式做颜色空间转换后得到 YUV422 格式。当得到 YUV422 格式的视频后，将其作为 VID1 pipeline 的输入，同时配置 VID1 pipeline 做比特精准的输出（如前 3.1 和 3.2 章节的描述）。那这样在对应的 VOUT 接口上，最终输出的视频为 YUV422 格式。

4.1 YUV422 输出的全景应用 在 VISION SDK 上的实现

基于 VISION SDK 开发的 360 度全景应用，比较容易从 RGB888 输出切换到 YUV422 输出。上述提到的为了做比特精准输出的寄存器都可以通过 DSS Link 的相关配置做到，并不需要修改 M4 上的显示驱动。Use case 级的 link and chain 需要做下列调整, figure 7 是调整后的 chain:

- A. 原来的 RGB888 输出 display link 不要初始化 VID1 pipeline
- B. 后面增加一级 DSS writeback capture link，然后将捕获到的做过颜色空间转换的 YUV422 帧经过一个调度器调度到新的 display link
- C. 新的 display link 只初始化 VID1 pipeline 并配置对应的 VOUT 输出参数

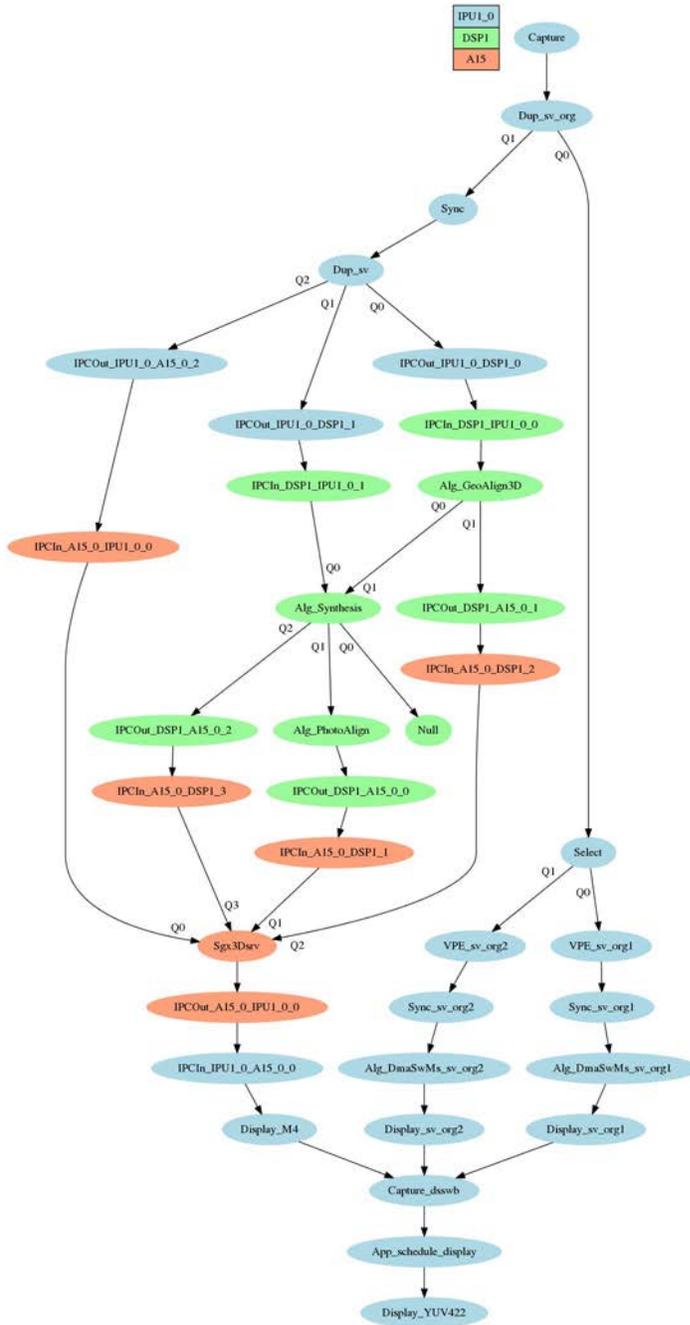


Figure 7. VISION SDK YUV422 输出用例的 chain

4.2 YUV422 输出的全景应用在 PSDKLA 的实现

When implement the YUV422 output in PSDKLA, it is more difficult than VISION SDK since Linux have the display driver framework as DRM. The modification cannot be made straight forward like VISIONSDK M4 display link/driver, it has to be followed the DRM methodology.

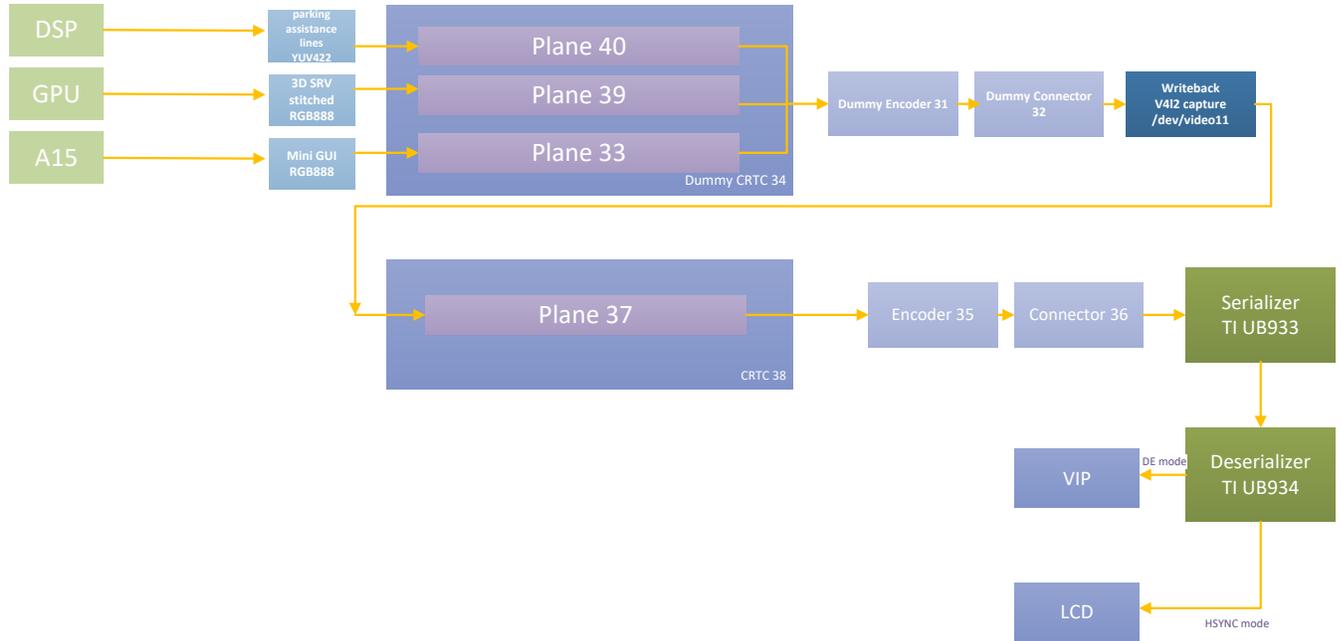


Figure 8. YUV422 输出的 DRM 结构

Figure 8 给出了在 Linux DRM 显示实现框架上输出 YUV422 的办法：

- A. DSP 算法计算并输出 YUV422 格式倒车辅助线 画面绑定到 CRTC34-Plane40 (VID2 pipeline)
- B. GPU 输出合成后的 RGB888 格式 360 度 3D 全景画面绑定到 CRTC34-Plane39(VID3 pipeline)
- C. A15 上的 MinuGUI 绘制 RGB888 格式的简单绑定到 UICRTC34-Plane33 (GFX pipeline)

注意: The CRTC34 是一个 dummy CRTC,它只有 dummy 的 encoder/connector,并不会真正的输出 VOUT 时序

- D. A Linux 用户态程序 使用标准的 Linux V4L2 接口利用 DSS writeback capture 设备 (/dev/video11) 来捕获输出的视频, 当然需要同时配置设备 /dev/video11 的输出格式为 YUV422, 这样实际是利用了 DSS writeback pipeline 里的颜色空间转换器进行了 RGB888 到 YUV422 的转换
- E. 把转换后的 YUV422 格式的视频帧绑定到 CRTC38-Plane37 (VID1 pipeline)
- F. CRTC38 对应的 encoder35 /connector36 输出 YUV422 完整时序

4.3 YUV422 输出的验证

如 Figure 8 所示, YUV422 的输出可以通过两个办法来进行校验:

- A. 环回 DSS 输出的 FPD LINK 加串器到本芯片 VIP 端的解串器。对于这样的场景来说，推荐使用 DSS 的 DE 替代 HSYNC 信号，这样的好处是 VIP 侧不用抓到无用的消影参数，可就不用配置 VIP 侧的 cropper 单元将无用像素删节掉。VIP 抓的数据存储为 YUV422 文件到文件系统，并用其他的第三方 YUV view 进行观察检验。
- B. 连接 DSS 输出的 FPD LINK LCD 端的解串器，这样的场景要求 LCD 侧支持 YUV422 的输入（可能经过其他的转换芯片），因为是输出到 LCD，推荐使用 HSYNC 信号而不是 DE 信号，因为大部分的 LCD 对行消影参数是有要求的。这样的好处是比较直观的看到输出结果，为了更好的比较结果，输出的 YUV422 画面尽可能用带各种色彩并有边沿线的画面。

Code

Below are the key patches  PSDKLA and VISION SDK configuration.

0009-visionsdk-change-display- dataformat-to-SYSTEM_DF_.patch

0013-arm-dts-change- play-timing-for-BT601-uyvy-30fps-o.patch

0005-visionsdk-enable-tdm-mode-in-usecase.patch

References

1. *Chapter 11 of DRA75x, DRA74x SoC for Automotive Infotainment Silicon Revision 2.0, 1.x Technical Reference Manual (Version AD)*
2. *Prasad Konnur, Sivaraj R, DSS Bit Exact Output Application Report*

有关 TI 设计信息和资源的重要通知

德州仪器 (TI) 公司提供的技术、应用或其他设计建议、服务或信息，包括但不限于与评估模块有关的参考设计和材料（总称“TI 资源”），旨在帮助设计人员开发整合了 TI 产品的应用；如果您（个人，或如果是代表贵公司，则为贵公司）以任何方式下载、访问或使用了任何特定的 TI 资源，即表示贵方同意仅为该等目标，按照本通知的条款进行使用。

TI 所提供的 TI 资源，并未扩大或以其他方式修改 TI 对 TI 产品的公开适用的质保及质保免责声明；也未导致 TI 承担任何额外的义务或责任。TI 有权对其 TI 资源进行纠正、增强、改进和其他修改。

您理解并同意，在设计应用时应自行实施独立的分析、评价和判断，且应全权负责并确保应用的安全性，以及您的应用（包括应用中使用的 TI 产品）应符合所有适用的法律法规及其他相关要求。就您的应用声明，您具备制订和实施下列保障措施所需的一切必要专业知识，能够 (1) 预见故障的危险后果，(2) 监视故障及其后果，以及 (3) 降低可能导致危险的故障几率并采取适当措施。您同意，在使用或分发包含 TI 产品的任何应用前，您将彻底测试该等应用和该等应用所用 TI 产品的功能而设计。除特定 TI 资源的公开文档中明确列出的测试外，TI 未进行任何其他测试。

您只有在为开发包含该等 TI 资源所列 TI 产品的应用时，才被授权使用、复制和修改任何相关单项 TI 资源。但并未依据禁止反言原则或其他法律授予您任何 TI 知识产权的任何其他明示或默示的许可，也未授予您 TI 或第三方的任何技术或知识产权的许可，该等许可包括但不限于任何专利权、版权、屏蔽作品权或与使用 TI 产品或服务的任何整合、机器制作、流程相关的其他知识产权。涉及或参考了第三方产品或服务的信息不构成使用此类产品或服务的许可或与其相关的保证或认可。使用 TI 资源可能需要您向第三方获得对该等第三方专利或其他知识产权的许可。

TI 资源系“按原样”提供。TI 兹免除对 TI 资源及其使用作出所有其他明确或默示的保证或陈述，包括但不限于对准确性或完整性、产权保证、无复发故障保证，以及适销性、适合特定用途和不侵犯任何第三方知识产权的任何默认保证。

TI 不负责任何申索，包括但不限于因组合产品所致或与之有关的申索，也不为您辩护或赔偿，即使该等产品组合已列于 TI 资源或其他地方。对因 TI 资源或其使用引起或与之有关的任何实际的、直接的、特殊的、附带的、间接的、惩罚性的、偶发的、从属或惩戒性损害赔偿，不管 TI 是否获悉可能会产生上述损害赔偿，TI 概不负责。

您同意向 TI 及其代表全额赔偿因您不遵守本通知条款和条件而引起的任何损害、费用、损失和/或责任。

本通知适用于 TI 资源。另有其他条款适用于某些类型的材料、TI 产品和服务的使用和采购。这些条款包括但不限于适用于 TI 的半导体产品 (<http://www.ti.com/sc/docs/stdterms.htm>)、[评估模块](http://www.ti.com/sc/docs/sampters.htm)和样品 (<http://www.ti.com/sc/docs/sampters.htm>) 的标准条款。

邮寄地址：上海市浦东新区世纪大道 1568 号中建大厦 32 楼，邮政编码：200122
Copyright © 2018 德州仪器半导体技术（上海）有限公司