

Jacinto6 SoC 上 VISION SDK 和 PSDKLA 之间的 IVA-HD 共享



Fredy Zhang

摘要

Jacinto6 硬件支持图像和视频加速器高清晰度 (IVA - HD)，可与复杂的视频编解码器搭配使用，同时保证功率和性能。

IVA-HD 加速器由图像处理单元 (IPU) 控制，可确保实时数据处理，并采用在帧边界上运行的编解码器算法。

VISION SDK 和 PSDKLA 中有两个不同版本的框架组件和编解码器，但使用两种框架同时处理多媒体流时会发生冲突。本应用报告提供了一种解决方案，可实现在 VISION SDK 和 PSDKLA 之间共享 IVAHD。

本文中讨论的工程配套资料和源代码可从以下 URL 下载：<https://www.ti.com/lit/zip/spracu0>。

内容

1 当前用例中的 IVA-HD 共享问题.....	2
2 IVA-HD 共享设计.....	3
3 IVA-HD 共享实现.....	4
3.1 启动流程.....	4
3.2 uboot 中的 IVA-HD DPLL 配置.....	4
3.3 配置 IPU 以同时支持 IPUMM 和解码链接.....	4
3.4 IVA-HD 配置.....	4
3.5 RPMSG 启动.....	6
4 早期解码演示.....	6
5 参考文献.....	6

插图清单

图 1-1. 在 VISION SDK 和 PSDKLA 之间共享 IVA-HD 问题.....	2
图 1-2. VISION SDK 编解码器.....	2
图 1-3. PSDKLA 编解码器.....	3
图 2-1. 新的 IVA-HD 共享概念.....	3
图 3-1. 启动流程.....	4
图 4-1. 解码示例.....	6

表格清单

表 3-1. 组件比较.....	4
------------------	---

商标

所有商标均为其各自所有者的财产。

1 当前用例中的 IVA-HD 共享问题

越来越多的客户考虑使用在单个 Jacinto/TDA 中支持信息娱乐功能以及 RVC 或动画的系统。最近，需要实时解码 H264 流的系统引入了基于以太网的 RVC，同时，部分客户要求播放带有启动动画的标识视频。

只有 Jacinto/TDA 具有 IVA-HD 硬件实例，因此使用两种不同的软件框架和编解码器驱动器来控制 IVA-HD，但会发生资源冲突。

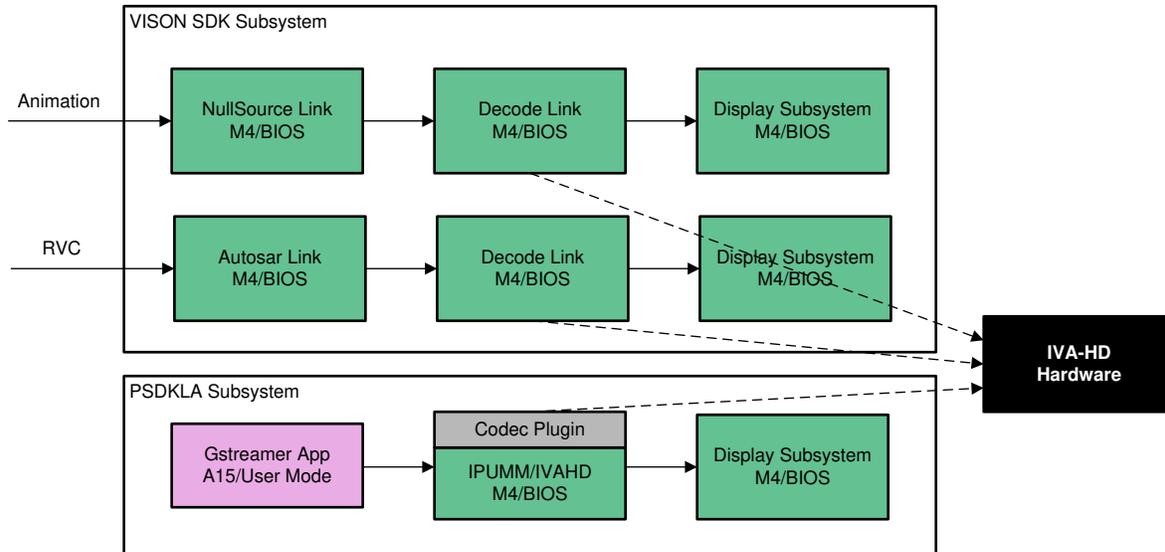


图 1-1. 在 VISION SDK 和 PSDKLA 之间共享 IVA-HD 问题

假设有一个支持这些功能的系统，如图 1-1 所示：

- 用例 A：动画或 RVC

考虑到快速启动的要求，大多数客户选择 VISION SDK 框架来开发 RVC 或动画。具有链接和链概念的 VISION SDK 设计用于实现任何用例。

当视频数据（通常为 h264）通过 NullSource 链接从引导分区读取为视频文件或从以太网帧中提取为原始流时，NullSource 链接或 Autosar 链接会将这些流发送到解码链接进行解码。如图 1-2 所示，解码链接会通过 VISION SDK 编解码器（它位于 M4 内核中，用于配置 IVA-HD 硬件模块来实现解码）调用驱动器，然后将解码帧发送到下一个链接。

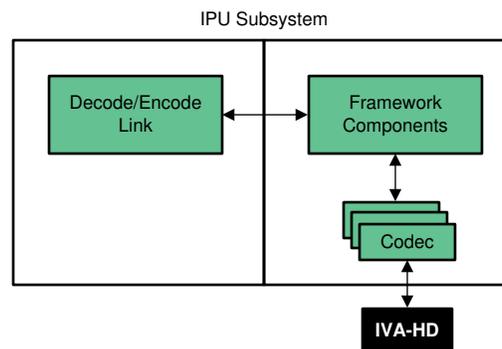


图 1-2. VISION SDK 编解码器

• 用例 B：多媒体播放器

默认情况下，TI PSDKLA 将开源 gstreamer 框架用于任何多媒体播放器用例。TI 提供了 gstreamer 插件：ducatiH264 和 ducaitiMJPEG。这些插件可以在 gstreamer 流水线中工作，以处理多媒体文件。如图 1-3 所示，gstreamer 会调用 ducatiH264 插件，以在 M4 内核上配置 IPUMM（编解码器），从而与 IVAHD 进行通信来解码 H.264 帧。解码插件会调用位于 M4 内核上的 TI IPUMM 驱动器，从而控制 IVA-HD 硬件模块，以实现解码/编码。然后，将帧发送到下一个插件。

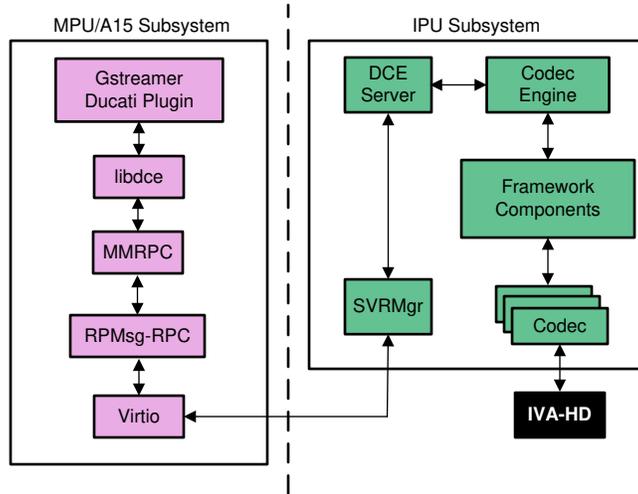


图 1-3. PSDKLA 编解码器

如果 A 和 B 用例同时发生，则将有一个用例失败，因为单个 IVA-HD 硬件实例无法同时由 M4 上的解码链接和 IPUMM 控制。实际上，同时需要 M4 具有解码链接和 IPUMM 支持功能以在 A15 上解码的任何用例都将失败。

2 IVA-HD 共享设计

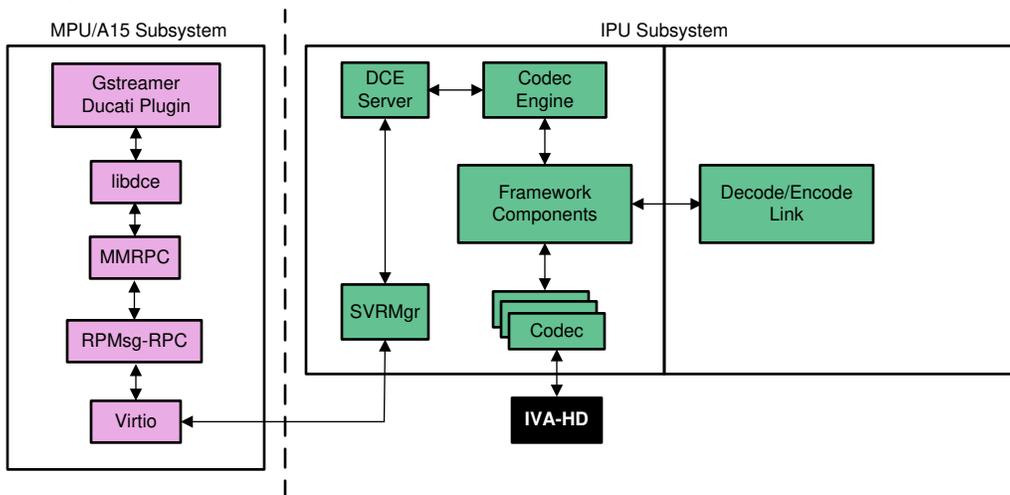


图 2-1. 新的 IVA-HD 共享概念

共享 IVA-HD 硬件要求只能在 M4 中调用驱动器。PSDKLA 和 VISION SDK 都有自己的框架组件和编解码器。它们不兼容。更好的方法是使用一个框架和编解码器，并使其与其他框架和编解码器兼容。

IPUMM 和 VISION SDK 解码链接在同一个内核上运行。如图 2-1 所示，框架组件支持多实例，因此 IVA-HD 可以被两个实体控制，gstreamer 解码插件和解码链接都可以调用框架组件。

- 本文档使用 PSDKLA 框架组件而非 VISION-SDK 框架组件
- 解码链接使用 IPUMM 编解码器替换 VISION SDK 编解码器

3 IVA-HD 共享实现

3.1 启动流程

对于 PSDKLA + VISION-SDK 架构，我们通常使用早期启动晚期附加模式。图 3-1 显示了启动流程。

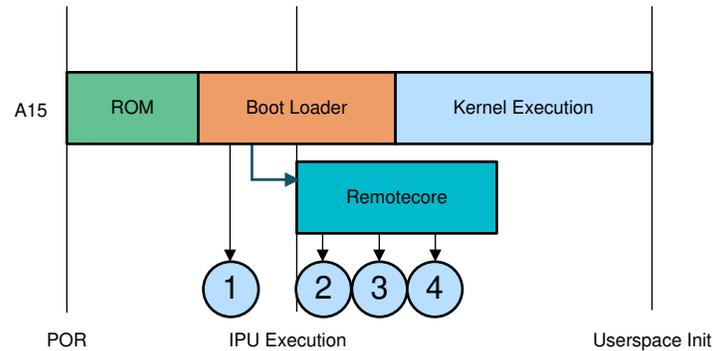


图 3-1. 启动流程

对于 PSDKLA，IVA-HD 的时钟和电源由内核控制。对于 VISION-SDK，使用 IPU 配置 IVA-HD。使用以下启动流程可解决资源冲突。

1. u-boot 中的 IVA-HD DPLL 配置
2. 配置 IPU 以同时支持 IPUMM 和解码链接
3. IVA-HD 配置
 - a. 编解码器引擎和 IPUMM 设置
 - b. 框架组件
 - c. 编解码器
 - d. IVA-HD 启动参数
4. RPMSG 驱动器配置

3.2 u-boot 中的 IVA-HD DPLL 配置

对于一般用例，IVA-HD 时钟和电源在内核中配置。通常在解决方案中，在 u-boot 中配置了 IVAHD DPLL。

如果在 u-boot 中配置了 IVA-HD，您必须删除内核 IVA-HD 配置。否则，内核会将 IVA-HD 复位。

3.3 配置 IPU 以同时支持 IPUMM 和解码链接

Linux 使用 IPUMM 在 PSDKLA 中解码。解码/编码链接位于 VISION-SDK 中，用于 RVC/动画解码。因此，必须在 M4 中支持 IPUMM 和解码链接，这应该在 `cfg.mk` 中配置

例如，在 J6 EVM 中：`sdk/vision_sdk/apps/configs/tda2xx_evm_linux_all/cfg.mk` **IVAHD_INCLUDE** 旨在支持解码/编码链接。**IPUMM_INCLUDE** 旨在支持 IPUMM。参考代码如下所示：

```
# Both IVAHD_INCLUDE & IPUMM_INCLUDE should not be set to "yes"
# Only one should be enabled to avoid IVA-HD resource conflict
IPUMM_INCLUDE=yes
IVAHD_INCLUDE=yes
```

3.4 IVA-HD 配置

比较 PSDKLA 和 VISION SDK 中的驱动器，如表 3-1 所示。它表明框架组件和代码不兼容。

表 3-1. 组件比较

类型	PSDKLA	VISION SDK	兼容性
编解码器引擎	是	否	/
框架组件	是	是	不兼容

表 3-1. 组件比较 (continued)

类型	PSDKLA	VISION SDK	兼容性
编解码器	是 (IPUMM)	是 ⁽¹⁾	不兼容

(1) VISION SDK 使用以下编解码器：

```
ivahd_h264enc_02_00_09_01_production/
ivahd_h264vdec_02_00_17_01_production/
ivahd_hdvicp20api_01_00_00_23_production/
ivahd_jpegvdec_01_00_13_01_production/
ivahd_jpegvenc_01_00_16_01_production/
```

3.4.1 编解码器引擎和 IPUMM 设置

若要设置编解码器引擎和 IPUMM，请参阅 VisionSDK_Linux_UserGuide.pdf：“2.4.2.2 可选组件 ipumm、编解码器引擎和框架组件”来设置编解码器引擎和 IPUMM。

该文档位于以下路径中：PROCESSOR_SDK_VISION_xx_xx_xx_xx/vision_sdk/docs/Linux/VisionSDK_Linux_UserGuide.pdf。

3.4.2 框架组件

根据 VisionSDK_Linux_UserGuide.pdf (位于以下路径：PROCESSOR_SDK_VISION_xx_xx_xx_xx/vision_sdk/docs/Linux/VisionSDK_Linux_UserGuide.pdf)，已知与 VSDK 一同打包的框架组件版本是用于 IVA-HD 分析的补丁版本，该补丁版本不能与 IPUMM 一同使用。

这些解决方案使用 PSDKLA 框架组件。此框架组件不能与 VISION SDK 编解码器一同使用，只能与 IPUMM 一同使用。因此，在 VISION SDK 中将编解码器更改为 IPUMM。

若要安装 PSDKLA 的新框架组件并替换 VISION SDK 版本，请参阅位于以下路径的 VisionSDK_Linux_UserGuide：PROCESSOR_SDK_VISION_xx_xx_xx_xx/vision_sdk/docs/Linux/VisionSDK_Linux_UserGuide.pdf。

3.4.3 编解码器

PSDKLA 使用 IPUMM 编解码器。但是，VISION SDK 使用以下编解码器：

- ivahd_h264enc_02_00_09_01_production
- ivahd_h264vdec_02_00_17_01_production
- ivahd_hdvicp20api_01_00_00_23_production
- ivahd_jpegvdec_01_00_13_01_production
- ivahd_jpegvenc_01_00_16_01_production

在 VISION SDK 中，IPUMM 编解码器用于替换 VISION SDK 代码。主要修改如下所示：

```
----- build/tools_path.mk -----
- hdvicplib_PATH           ?= $(TI_SW_ROOT)/codecs/ivahd_hdvicp20api_01_00_00_23_production
- jpegvenc_PATH            ?= $(TI_SW_ROOT)/codecs/ivahd_jpegvenc_01_00_16_01_production
- jpegvdec_PATH            ?= $(TI_SW_ROOT)/codecs/ivahd_jpegvdec_01_00_13_01_production
- h264venc_PATH            ?= $(TI_SW_ROOT)/codecs/ivahd_h264enc_02_00_09_01_production
- h264vdec_PATH            ?= $(TI_SW_ROOT)/codecs/ivahd_h264vdec_02_00_17_01_production
+hdvicplib_PATH            ?= $(mm_PATH)/extrel/ti/ivahd_codecs
+jpegvenc_PATH             ?= $(mm_PATH)/extrel/ti/ivahd_codecs
+jpegvdec_PATH             ?= $(mm_PATH)/extrel/ti/ivahd_codecs
+h264venc_PATH             ?= $(mm_PATH)/extrel/ti/ivahd_codecs
+h264vdec_PATH             ?= $(mm_PATH)/extrel/ti/ivahd_codecs
```

3.4.4 IVA-HD 启动参数

根据用例，解码链接需要首先启动。因此，在 M4 中配置了 IVA-HD：

```
sdk/links_fw/src/rtos/links_ipu/iva/codec_utils/src/hdvicp2_config.c: icont_boot[]
```

从 IPUMM 中删除配置：

```

sdk/ti_components/codecs/ipumm
--- a/src/ti/framework/dce/ivahd.c

void ivahd_init(uint32_t chipset_id)
...
-   ivahd_boot();
    DEBUG("RMAN_register() for HDVICP is successful")

```

3.5 RPMSG 启动

两个子系统通过 RPMSG 相互通信。`SystemipcConnectToHLOSThread(void)`; 使 rpmsg 驱动器启动，*IPUMM* 启动后删除 `IpMgr_rpmsgStartup()`；。

```

----- links_fw/src/rtos/bios_app_common/tda2ex/ipu2/src/ipu_primary.c -----static void
ipumm_startup_task(uint32_t arg0, uint32_t arg1)
...
-   IpMgr_rpmsgStartup();
    IPUMM_Main(0, NULL);
}

```

4 早期解码演示

下面提供了一个演示来测试此用例。VISION SDK 会从 u-boot 读取 h264 视频数据，然后发送解码链接进行解码。此用例如图 4-1 所示。运行 Linux 后，您可以使用 `gststreamer` 播放视频。

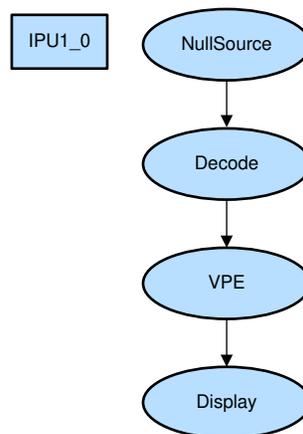


图 4-1. 解码示例

PSDKLA 可以运行 `gststreamer`，命令如下：

```
gst-launch-1.0 playbin uri=file:///home/root/test.mp4 video-sink=waylandsink
```

5 参考文献

- [Linux 中的 Early Boot 和 Late Attach 用例 wiki 网页](#)
- [多媒体](#)

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2022，德州仪器 (TI) 公司