

Rockchip Developer Guide DVR&DMS Product

ID: RK-SM-YF-398

Release Version: V1.1.9

Release Date: 2022-11-29

Security Level: ☐Top-Secret ☐Secret ☐Internal ☒Public

DISCLAIMER

THIS DOCUMENT IS PROVIDED “AS IS”. ROCKCHIP ELECTRONICS CO., LTD.(“ROCKCHIP”)DOES NOT PROVIDE ANY WARRANTY OF ANY KIND, EXPRESSED, IMPLIED OR OTHERWISE, WITH RESPECT TO THE ACCURACY, RELIABILITY, COMPLETENESS, MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE OR NON-INFRINGEMENT OF ANY REPRESENTATION, INFORMATION AND CONTENT IN THIS DOCUMENT. THIS DOCUMENT IS FOR REFERENCE ONLY. THIS DOCUMENT MAY BE UPDATED OR CHANGED WITHOUT ANY NOTICE AT ANY TIME DUE TO THE UPGRADES OF THE PRODUCT OR ANY OTHER REASONS.

Trademark Statement

"Rockchip", "瑞芯微", "瑞芯" shall be Rockchip's registered trademarks and owned by Rockchip. All the other trademarks or registered trademarks mentioned in this document shall be owned by their respective owners.

All rights reserved. ©2022. Rockchip Electronics Co., Ltd.

Beyond the scope of fair use, neither any entity nor individual shall extract, copy, or distribute this document in any form in whole or in part without the written approval of Rockchip.

Rockchip Electronics Co., Ltd.

No.18 Building, A District, No.89, software Boulevard Fuzhou, Fujian, PRC

Website: www.rock-chips.com

Customer service Tel: +86-4007-700-590

Customer service Fax: +86-591-83951833

Customer service e-Mail: fae@rock-chips.com

Preface

Overview

This document is going to introduce DVR&DMS product solution.

Product Version

Chipset	Kernel Version
RV1126, RV1109	Linux 4.19

Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers

Software development engineers

Revision History

Version	Author	Date	Change Description
V1.0.0	Vicent Chi, Zhihua Wang, Zhichao Yu	2021-01-31	Initial version
V1.0.1	Vicent Chi	2021-03-01	Add MPI+DVP solution introduction
V1.0.2	Ruby Zhang	2021-03-15	Update product version
V1.1.0	Zhichao Yu	2021-04-30	Modify the AD chip access solution introduction; Add VP introduction
V1.1.1	Vicent Chi	2021-06-03	Add FAQ
V1.1.2	Zhichao Yu	2021-07-24	Add audio related introduction
V1.1.3	Ruby Zhang	2021-10-11	Fix some expressions
V1.1.4	Zhichao Yu	2021-10-26	Support Interlaced Frame Composite
V1.1.5	Xing Zheng	2021-10-27	Add multichannel audio record introduction
V1.1.6	Zhichao Yu	2022-03-08	Add additional notes
V1.1.7	Zhichao Yu	2022-06-08	Add instruction for input different resolution cameras
V1.1.9	Zhichao Yu	2022-11-29	Add product performance optimization introduction

Contents

Rockchip Developer Guide DVR&DMS Product

1. Rockchip DVR/DMS Product Solution Introduction
 - 1.1 Advantages of Developing DVR/DMS Product on RV1126 Platform
 - 1.2 Analog HD RX Chip Support List
 - 1.3 RV1126 DVR/DMS Products Application Block diagram
2. Analog HD RX Chip Driver Development Introduction
 - 2.1 Kernel Config
 - 2.2 Kernel dts Configuration
3. Data Streams Channels Introduction
 - 3.1 Two-channel Solution
 - 3.2 Video Format Restrictions of Channels
 - 3.2.1 VICAP Channel
 - 3.2.2 ISP Channel
 - 3.3 Enumeration of Video Nodes of Channels
 - 3.3.1 VICAP Channel
 - 3.3.2 ISP Channel
 - 3.4 Video Capture Limitation of Channels
 - 3.4.1 ISP channel
 - 3.5 Query Resolution and Video Signal of Channels
 - 3.6 Query Hot Plug Interface In Real Time
4. Multi-channel Audio Capture
5. rkmedia_vmix_vo_dvr_test Application Note
 - 5.1 8-channel Video Capture and H264 Encoding
 - 5.2 Support for 8-channel Video Synthesis Display
 - 5.3 Support for 8-channel Video Switching to Front 4-channel and Rear 4-channel Display
 - 5.4 Support for Area Frame
 - 5.5 Support for RGN Cover
 - 5.6 Support for Screen OSD
 - 5.7 Support for Channel Display and Hide
 - 5.8 Support for Obtaining Regional Brightness of Channels
6. VP Module Introduction
7. System Performance Optimization
 - 7.1 CPU Performance Optimization
 - 7.1.1 Turn off High Resolution Timer
 - 7.1.2 Turn off IOMMU in ISPP
 - 7.2 RGA Performance Optimization
 - 7.2.1 To Enable DMA Buffer Cache
8. FAQ
 - 8.1 Pictures are Staggered When Hot Plugging
 - 8.1.1 Solution
 - 8.2 Support Interlaced Frame Composite for CVBS

1. Rockchip DVR/DMS Product Solution Introduction

RV1126 chip has two MIPI interfaces and one DVP interface. In addition, it provides powerful encoding performance and supports up to 8 channels of 1080@15fps simultaneous encoding. With a built-in 2T computing power NPU, it is very suitable for the development of DVR/DMS products.

1.1 Advantages of Developing DVR/DMS Product on RV1126 Platform

- Support up to 8 channels of 1080P analog HD video input;
- Powerful AI processing capabilities, support for operating DMS+ADAS algorithms at the same time;
- Powerful encoding capabilities, up to 8 channels of 1080P@15fps simultaneous encoding;
- Support 8 channels video OSD overlay;
- Support 8 channels video split-screen display demo;

1.2 Analog HD RX Chip Support List

At present, RV1126 platform has been adapted to many analog HD RX chips, and the drivers for these chips have been integrated in the SDK. You can select from the following table:

Model	Manufacturer	Interface	Number of Channels	Maximum Supported Resolution
NVP6188	Nextchip	MIPI	4	4K
N4	Nextchip	MIPI	4	1080P
TP2815	Techpoint	MIPI	4	1080P
TP2855	Techpoint	MIPI	4	1080P
TP9930	Techpoint	DVP	4	2K
TP9950	Techpoint	MIPI/DVP	1	1080P

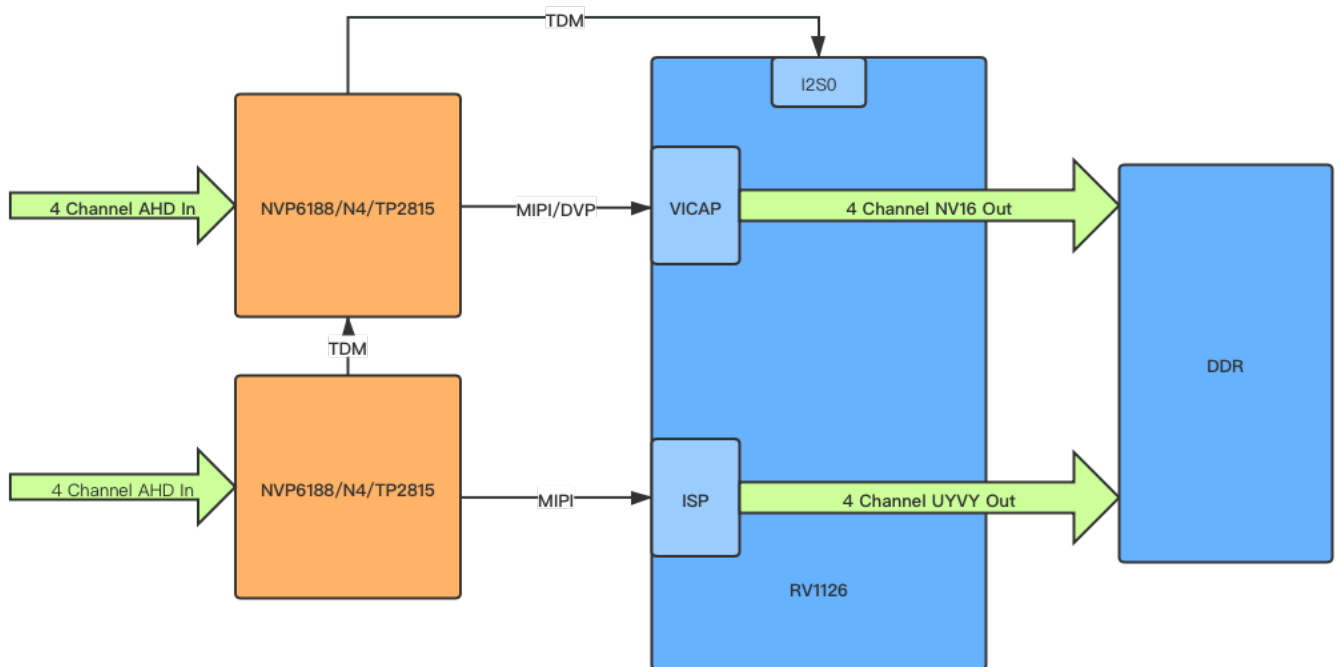
1.3 RV1126 DVR/DMS Products Application Block diagram

RV1126 supports different AD chip access methods, MIPI + MIPI and MIPI + DVP.

Due to the limitation of the RV1126 chip, it is necessary to limit the image output formats of different channels:

- **VICAP channel: all should adopt NV16 format;**
- **ISP access: all should adopt UYVY format;**

The block diagram of the reference solution of camera input is as follows:



NOTE: The DVP port cannot support different resolution camera input, therefore dual MIPI scheme is recommended.

2. Analog HD RX Chip Driver Development Introduction

2.1 Kernel Config

Open the relevant config of the RX chip accordingly:

```
CONFIG_VIDEO_NVP6188=y
```

2.2 Kernel dts Configuration

Take NVP6188 RX chip as an example:

```
nvp6188_0: nvp6188_0@30 {
    compatible = "nvp6188";
    reg = <0x30>;
    clocks = <&cru CLK_MIPICSI_OUT>;
    clock-names = "xvclk";
    power-domains = <&power RV1126_PD_VI>;
```

```

pinctrl-names = "rockchip,camera_default";
pinctrl-0 = <&mipicsi_clk0>;
reset-gpios = <&gpio4 RK_PA0 GPIO_ACTIVE_HIGH>;
power-gpios = <&gpio1 RK_PD4 GPIO_ACTIVE_HIGH>;
vi-gpios = <&gpio3 RK_PC0 GPIO_ACTIVE_HIGH>;
rockchip,camera-module-index = <0>;
rockchip,camera-module-facing = "front";
rockchip,camera-module-name = "nvp6188";
rockchip,camera-module-lens-name = "nvp6188";
port {
    ucam_out0: endpoint {
        remote-endpoint = <&mipi_in_ucam0>;
        data-lanes = <1 2 3 4>;
    };
};

nvp6188_1: nvp6188_1@32 {
    compatible = "nvp6188";
    reg = <0x32>;
    clocks = <&cru CLK_MIPICSI_OUT>;
    clock-names = "xvclk";
    power-domains = <&power RV1126_PD_VI>;
    pinctrl-names = "rockchip,camera_default";
    pinctrl-0 = <&mipicsi_clk1>;
    reset-gpios = <&gpio4 RK_PA1 GPIO_ACTIVE_HIGH>;
    vi-gpios = <&gpio3 RK_PC1 GPIO_ACTIVE_HIGH>;
    rockchip,camera-module-index = <1>;
    rockchip,camera-module-facing = "back";
    rockchip,camera-module-name = "nvp6188";
    rockchip,camera-module-lens-name = "nvp6188";
    port {
        ucam_out1: endpoint {
            remote-endpoint = <&csi_dphy1_input>;
            data-lanes = <1 2 3 4>;
        };
    };
};

```

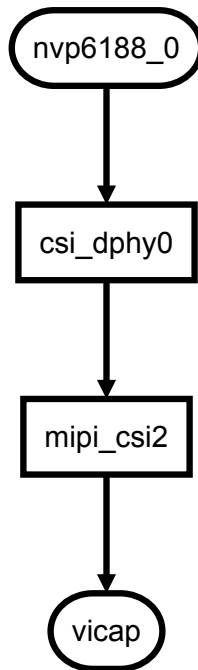
3. Data Streams Channels Introduction

3.1 Two-channel Solution

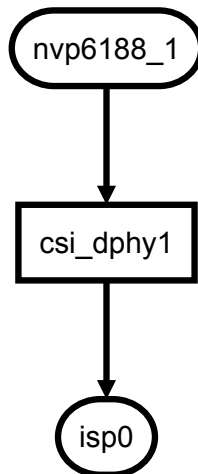
Dual MIPI Solution

Take the dual NVP6188 as an example:

- VICAP channel 0



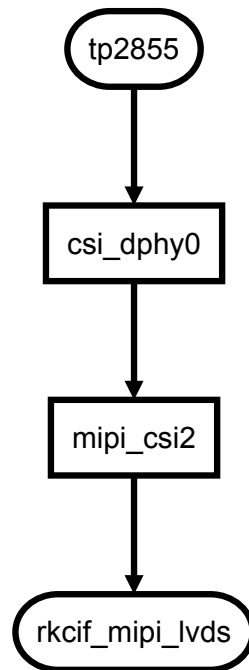
- ISP channel 1



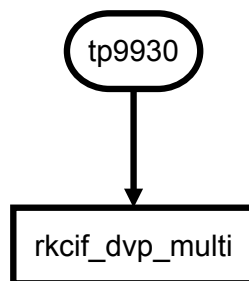
MIPI+DVP solution

Take TP9930+TP2855 for example:

- VICAP Channel 0



- VICAP Channel 1



3.2 Video Format Restrictions of Channels

3.2.1 VICAP Channel

- Use NV16 format for capturing.

3.2.2 ISP Channel

- Use UYVY format for capturing

3.3 Enumeration of Video Nodes of Channels

3.3.1 VICAP Channel

- Capture device node name of stream_cif_mipi_id0/1/2/3 by media-ctl -p -d /dev/mediaX

```
[root@RV1126_RV1109:/]# media-ctl -p -d /dev/media0
Media controller API version 4.19.111

Media device information
-----
driver          rkcif
model           rkcif_mipi_lvds
serial
bus info
hw revision     0x0
driver version  4.19.111

Device topology
- entity 1: stream_cif_mipi_id0 (1 pad, 4 links)
    type Node subtype V4L flags 0
    device node name /dev/video0
    pad0: Sink
        <- "rockchip-mipi-csi2":1 [ENABLED]
        <- "rockchip-mipi-csi2":2 []
        <- "rockchip-mipi-csi2":3 []
        <- "rockchip-mipi-csi2":4 []

- entity 5: stream_cif_mipi_id1 (1 pad, 4 links)
    type Node subtype V4L flags 0
    device node name /dev/video1
    pad0: Sink
        <- "rockchip-mipi-csi2":1 []
        <- "rockchip-mipi-csi2":2 [ENABLED]
        <- "rockchip-mipi-csi2":3 []
        <- "rockchip-mipi-csi2":4 []

- entity 9: stream_cif_mipi_id2 (1 pad, 4 links)
    type Node subtype V4L flags 0
    device node name /dev/video2
    pad0: Sink
        <- "rockchip-mipi-csi2":1 []
        <- "rockchip-mipi-csi2":2 []
        <- "rockchip-mipi-csi2":3 [ENABLED]
        <- "rockchip-mipi-csi2":4 []

- entity 13: stream_cif_mipi_id3 (1 pad, 4 links)
    type Node subtype V4L flags 0
    device node name /dev/video3
```

```

pad0: Sink
    <- "rockchip-mipi-csi2":1 []
    <- "rockchip-mipi-csi2":2 []
    <- "rockchip-mipi-csi2":3 []
    <- "rockchip-mipi-csi2":4 [ENABLED]

```

3.3.2 ISP Channel

- Get device node name of rkisp_mainpath、rkisp_rawwr0/1/2 by media-ctl -p -d /dev/mediaX

```

media-ctl -p -d /dev/media1
Media controller API version 4.19.111

Media device information
-----
driver          rkisp
model           rkisp0
serial
bus info
hw revision     0x0
driver version  4.19.111

Device topology
- entity 17: rkisp_mainpath (1 pad, 1 link)
    type Node subtype V4L flags 0
    device node name /dev/video5
    pad0: Sink
        <- "rkisp-isp-subdev":2 [ENABLED]

- entity 29: rkisp_rawwr0 (1 pad, 1 link)
    type Node subtype V4L flags 0
    device node name /dev/video7
    pad0: Sink
        <- "rkisp-csi-subdev":2 [ENABLED]

- entity 35: rkisp_rawwr1 (1 pad, 1 link)
    type Node subtype V4L flags 0
    device node name /dev/video8
    pad0: Sink
        <- "rkisp-csi-subdev":3 [ENABLED]

- entity 41: rkisp_rawwr2 (1 pad, 1 link)
    type Node subtype V4L flags 0
    device node name /dev/video9
    pad0: Sink

```

```
<- "rkisp-csi-subdev":4 [ENABLED]
```

3.4 Video Capture Limitation of Channels

3.4.1 ISP channel

- pipeline switch

The following code should be added to the boot script(/dev/media1 is determined according to the actual isp registration)

```
media-ctl -d /dev/media1 -l '"rkisp-isp-subdev":2->"rkisp-bridge-isp":0[0] '
media-ctl -d /dev/media1 -l '"rkisp-isp-subdev":2->"rkisp_mainpath":0[1] '
```

- stream on switch

Because there is no separate switch for the stream on switch of the four channels of rkisp_mainpath and rkisp_rawwr0/1/2, therefore, if you want to capture any one of the rkisp_rawwr0/1/2 three channels, you have to ensure that the three channels will not stream out until the rkisp_mainpath channel is in the stream on state.

- rkisp_mainpath format switch

The output is 1080p by default. If you want to switch the format to 720p, you need to execute:

```
media-ctl -d /dev/media1 --set-v4l2 '"m01_b_nvp6188 1-0032":0[fmt:UYVY8_2X8/1280x720] '
media-ctl -d /dev/media1 --set-v4l2 '"rkisp-csi-subdev":1[fmt:UYVY8_2X8/1280x720] '
media-ctl -d /dev/media1 --set-v4l2 '"rkisp-isp-subdev":0[fmt:YUYV8_2X8/1280x720] '
media-ctl -d /dev/media1 --set-v4l2 '"rkisp-isp-subdev":0[crop:(0,0)/1280x720] '
media-ctl -d /dev/media1 --set-v4l2 '"rkisp-isp-subdev":2[fmt:YUYV8_2X8/1280x720] '
media-ctl -d /dev/media1 --set-v4l2 '"rkisp-isp-subdev":2[crop:(0,0)/1280x720] '
```

3.5 Query Resolution and Video Signal of Channels

- Get subdev node name of sensor by media-ctl -p -d /dev/mediaX

```
media-ctl -p -d /dev/media1
Media controller API version 4.19.111
```

Media device information

```
-----  
driver          rkisp  
model           rkisp0  
serial  
bus info  
hw revision     0x0  
driver version  4.19.111
```

Device topology

....

```
- entity 92: m01_b_nvp6188 1-0032 (1 pad, 1 link)  
    type V4L2 subdev subtype Sensor flags 0  
    device node name /dev/v4l-subdev6  
    pad0: Source  
        [fmt:UYVY8_2X8/1920x1080 field:none]  
        -> "rockchip-mipi-dphy-rx":0 [ENABLED]
```

- Get the resolution before opening the channel

```
#include <stdio.h>  
#include <unistd.h>  
#include <fcntl.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <sys/time.h>  
#include <sys/mman.h>  
#include <sys/ioctl.h>  
#include <linux/videodev2.h>  
  
#define RKMODULE_MAX_VC_CH    4  
  
struct rkmodule_vc_fmt_info {  
    __u32 width[RKMODULE_MAX_VC_CH];  
    __u32 height[RKMODULE_MAX_VC_CH];  
    __u32 fps[RKMODULE_MAX_VC_CH];  
} __attribute__((packed));  
  
struct rkmodule_vc_hotplug_info {  
    __u8 detect_status;  
} __attribute__((packed));  
  
#define RKMODULE_GET_VC_FMT_INFO \  
    _IOR('V', BASE_VIDIOC_PRIVATE + 12, struct rkmodule_vc_fmt_info)
```

```

#define RKMODULE_GET_VC_HOTPLUG_INFO \
    _IOR('V', BASE_VIDIOC_PRIVATE + 13, struct rkmodule_vc_hotplug_info)

int main(int argc, char *argv[]) {
    int ch = 0;
    struct rkmodule_vc_hotplug_info status;
    struct rkmodule_vc_fmt_info fmt;
    int fd = open("/dev/v4l-subdev2", O_RDWR, 0);
    ioctl(fd, RKMODULE_GET_VC_FMT_INFO, &fmt);
    ioctl(fd, RKMODULE_GET_VC_HOTPLUG_INFO, &status);
    for(ch = 0; ch < 4; ch++) {
        printf("# ch: %d\n", ch);
        printf("\t width: %d\n", fmt.width[ch]);
        printf("\t height: %d\n", fmt.height[ch]);
        printf("\t fps: %d\n", fmt.fps[ch]);
        printf("\t plug in: %d\n", (status.detect_status & (1 << ch)) ? 1 : 0);
    }
    close(fd);
    return 0;
}

```

3.6 Query Hot Plug Interface In Real Time

- Provide sysfs nodes to user layer for read queries.

```

/sys/devices/platform/ff510000.i2c/i2c-1/1-0032/hotplug_status
/sys/devices/platform/ff510000.i2c/i2c-1/1-0030/hotplug_status

```

4. Multi-channel Audio Capture

The audio of DVR product is generally collected to RV1126 through TDM, and the following commands can be used to collect audio data from TDM:

```

arecord -Dhw:0,0 -c 8 -f S16_LE -r 8000 /tmp/record.wav -vv

```

And then we can record a specific channel follow the steps below:

1. Modifying /etc/asound.conf to config dsnooper

```
# Copyright © 2021 Rockchip Electronics Co. Ltd.
# Author: Xing Zheng <zhengxing@rock-chips.com>

pcm.!default
{
    type asym
    playback.pcm "hw:0,0"
    capture.pcm "hw:0,0"
}

pcm.dsnooper_6 {
    type dsnoop
    ipc_key 5978291 # must be unique for all dmix plugins!!!!
    ipc_key_add_uid yes
    slave {
        pcm "hw:0,0"
        channels 6
    }
    bindings {
        0 0
        1 1
        2 2
        3 3
        4 4
        5 5
    }
}

pcm.cap_ch0_rt {
    type route
    slave {
        pcm dsnooper_6
        channels 6
    }
    ttable {
        0.0 1.0
    }
}

pcm.cap_ch1_rt {
    type route
    slave {
        pcm dsnooper_6
        channels 6
    }
    ttable {
        0.1 1.0
    }
}
```

```
pcm.cap_ch2_rt {
    type route
    slave {
        pcm dsnooper_6
        channels 6
    }
    ttable {
        0.2 1.0
    }
}

pcm.cap_ch3_rt {
    type route
    slave {
        pcm dsnooper_6
        channels 6
    }
    ttable {
        0.3 1.0
    }
}

pcm.cap_ch4_rt {
    type route
    slave {
        pcm dsnooper_6
        channels 6
    }
    ttable {
        0.4 1.0
    }
}

pcm.cap_ch5_rt {
    type route
    slave {
        pcm dsnooper_6
        channels 6
    }
    ttable {
        0.5 1.0
    }
}

pcm.cap_ch0 {
    type plug
    slave.pcm "cap_ch0_rt"
}
```

```
pcm.cap_ch1 {  
    type plug  
    slave.pcm "cap_ch1_rt"  
}  
  
pcm.cap_ch2 {  
    type plug  
    slave.pcm "cap_ch2_rt"  
}  
  
pcm.cap_ch3 {  
    type plug  
    slave.pcm "cap_ch3_rt"  
}  
  
pcm.cap_ch4 {  
    type plug  
    slave.pcm "cap_ch4_rt"  
}  
  
pcm.cap_ch5 {  
    type plug  
    slave.pcm "cap_ch5_rt"  
}
```

2. Use the following command to record the 6th channel(the 1st channel is ch0)

```
arecord -Dcap_ch5 -c 1 -r 16000 -f S16_LE /tmp/record_ch5.wav
```

Finally, in your application, you can open "cap_chX" to record audio data by calling rkmedia interface (X stands for 0 ~ 5).

NOTE: On RV1126/RV1109 SoC, only I2S0 supports TDM mode.

5. rkmedia_vmix_vo_dvr_test Application Note

The rkmedia_vmix_vo_dvr_test is mainly used to realizes 8-channel video capture and encoding, and 8-channel video synthesis display. The source code is located in SDK/external/rkmedia/examples.

5.1 8-channel Video Capture and H264 Encoding

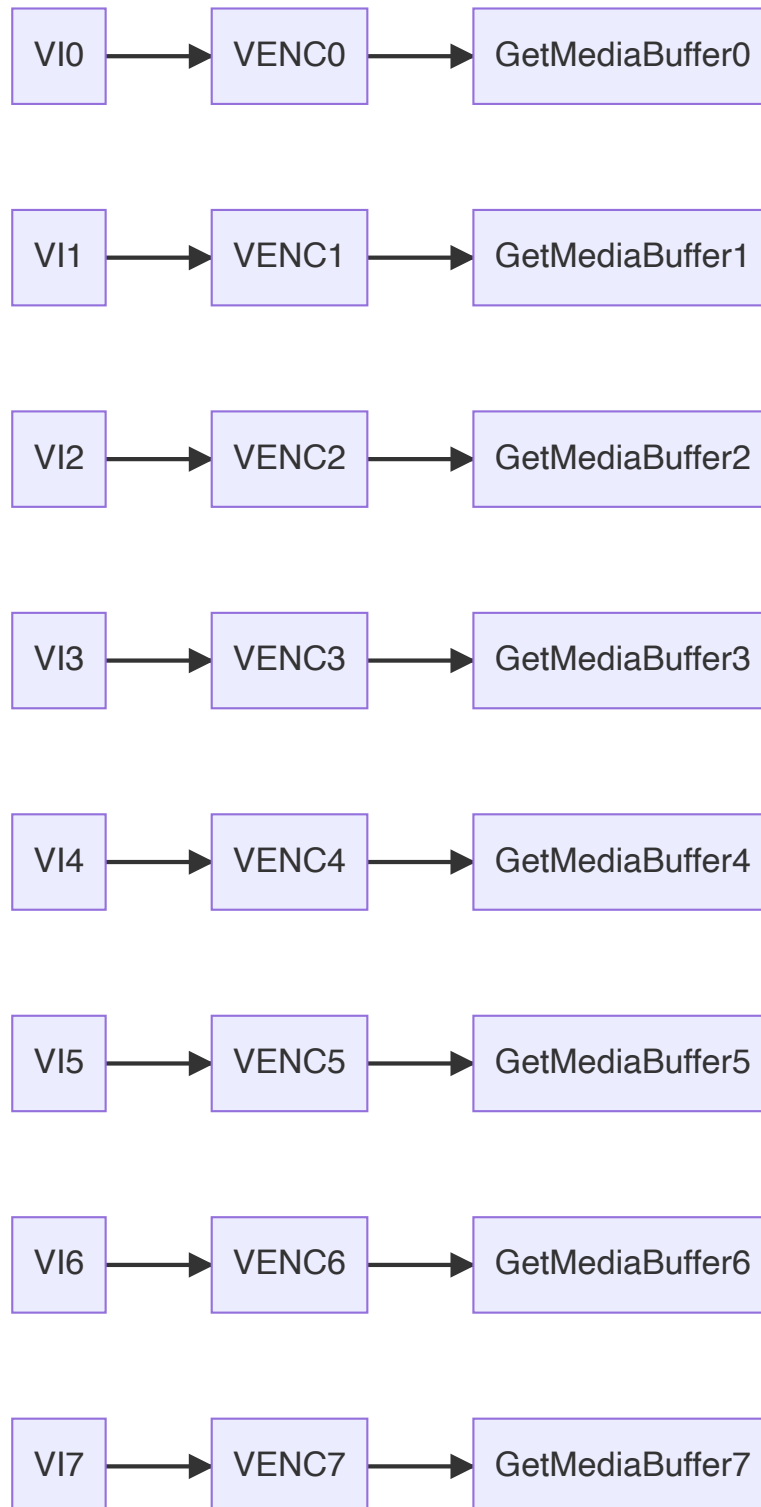
The 8-channel video capture node, resolution, and format are configured through an array, which is convenient for users to modify and debug:

```
stDvr dvr8[8] = {
    {0, "/dev/video30", 1920, 1080, IMAGE_TYPE_NV12, 0},
    {1, "/dev/video31", 1920, 1080, IMAGE_TYPE_NV12, 0},
    {2, "/dev/video32", 1920, 1080, IMAGE_TYPE_NV12, 0},
    {3, "/dev/video33", 1920, 1080, IMAGE_TYPE_NV12, 0},
    {4, "/dev/video37", 1920, 1080, IMAGE_TYPE_NV12, 0},
    {5, "/dev/video38", 1920, 1080, IMAGE_TYPE_NV12, 0},
    {6, "/dev/video39", 1920, 1080, IMAGE_TYPE_NV12, 0},
    {7, "/dev/video40", 1920, 1080, IMAGE_TYPE_NV12, 0},
};
```

According to the recommendation of the dual MIPI solution, it should be modified to:

```
stDvr dvr8[8] = {
    {0, "/dev/video0", 1920, 1080, IMAGE_TYPE_NV16, 0},
    {1, "/dev/video1", 1920, 1080, IMAGE_TYPE_NV16, 0},
    {2, "/dev/video2", 1920, 1080, IMAGE_TYPE_NV16, 0},
    {3, "/dev/video3", 1920, 1080, IMAGE_TYPE_NV16, 0},
    {4, "/dev/video5", 1920, 1080, IMAGE_TYPE_UYVY422, 0},
    {5, "/dev/video7", 1920, 1080, IMAGE_TYPE_UYVY422, 0},
    {6, "/dev/video8", 1920, 1080, IMAGE_TYPE_UYVY422, 0},
    {7, "/dev/video9", 1920, 1080, IMAGE_TYPE_UYVY422, 0},
};
```

The 8-channel video VI implements 8-channel H264 encoding through bind VENC, and the 8-channel VENC-encoded data can be obtained through the GetMediaBuffer thread. Users can realize video transmission requirements base on it.

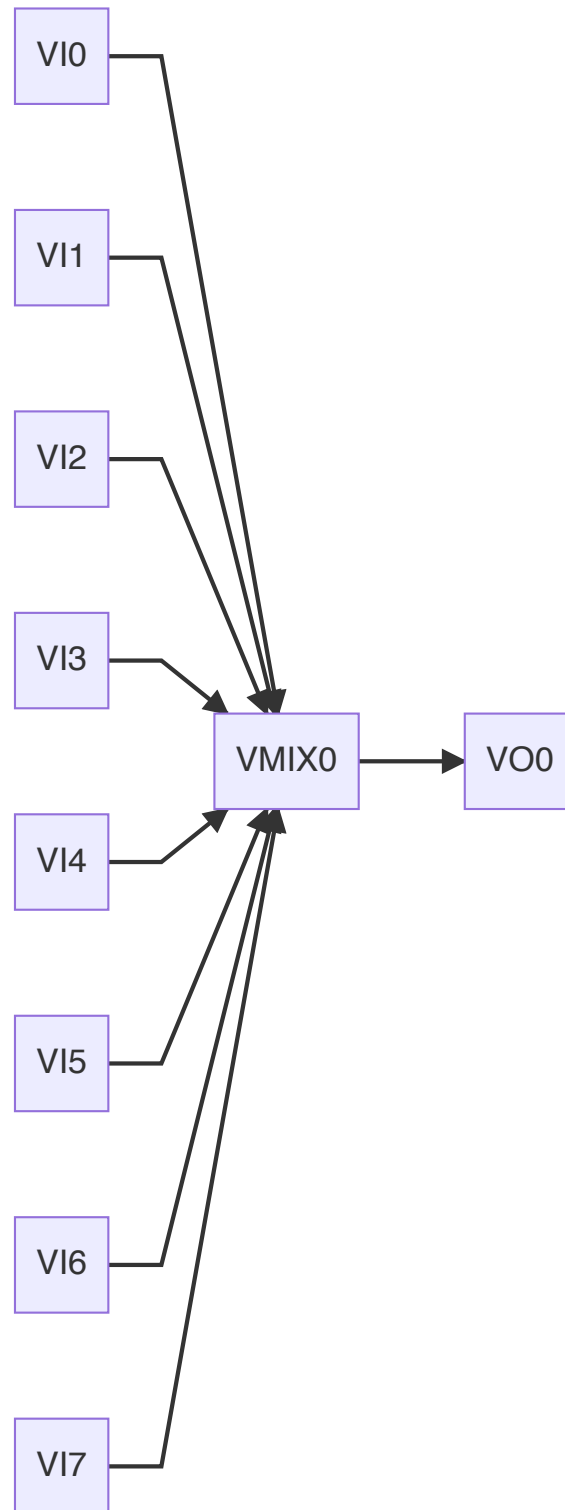


5.2 Support for 8-channel Video Synthesis Display

The 8-channel video specifies the rectangular area of the screen to be displayed through an array, which is convenient for users to modify and debug:

```
RECT_S area_2x4[8] = {
    {0, 0, WIDTH / 2, HEIGHT / 4},
    {WIDTH / 2, 0, WIDTH / 2, HEIGHT / 4},
    {0, HEIGHT / 4, WIDTH / 2, HEIGHT / 4},
    {WIDTH / 2, HEIGHT / 4, WIDTH / 2, HEIGHT / 4},
    {0, HEIGHT / 2, WIDTH / 2, HEIGHT / 4},
    {WIDTH / 2, HEIGHT / 2, WIDTH / 2, HEIGHT / 4},
    {0, HEIGHT * 3 / 4, WIDTH / 2, HEIGHT / 4},
    {WIDTH / 2, HEIGHT * 3 / 4, WIDTH / 2, HEIGHT / 4},
};
```

The 8-channel video synthesis display is realized through the VMIX+VO module:



5.3 Support for 8-channel Video Switching to Front 4-channel and Rear 4-channel Display

Through `dvr_bind` and `dvr_unbind`, the 8-channel video can be switched to the front 4-channel and the rear 4-channel display. Users only need to define the rectangular display area of the front 4-channel and the rear 4-channel.

5.4 Support for Area Frame

Area frame is realized by drawing lines on the whole screen, increasing the area boundary, and specifying the drawing area through an array. The minimum line width is 2, and an even number is required:

```
RECT_S line_2x4[4] = {  
    {0, HEIGHT / 4, WIDTH, 2},  
    {0, HEIGHT / 2, WIDTH, 2},  
    {0, HEIGHT * 3 / 4, WIDTH, 2},  
    {WIDTH / 2, 0, 2, HEIGHT},  
};
```

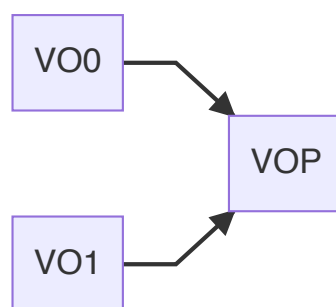
Line drawing area is set through RK_MPI_VMIX_SetLineInfo.

5.5 Support for RGN Cover

It is supported to set sensitive areas for each channel, which is realized through RK_MPI_VMIX_RGN_SetCover.

5.6 Support for Screen OSD

Screen OSD is realized through VO1, users can draw OSD in the buffer (the format is ABGR), after sending to VO1, the OSD can be superimposed on the VO0 video to display the effect through Alpha. In the osd_thread of application, an OSD switching display of two color blocks is drawn every 500ms.



5.7 Support for Channel Display and Hide

Display and hide the channel through RK_MPI_VMIX_ShowChn, RK_MPI_VMIX_HideChn.

5.8 Support for Obtaining Regional Brightness of Channels

Regional brightness of channels is obtained through RK_MPI_VMIX_GetChnRegionLuma, you can get up to 64 regional brightness at a time. The coordinate of each channel is the starting coordinate of the region relative to the channel, not the starting coordinate of the relative screen. You can realize the OSD inverse effect of the screen through the regional brightness.

6. VP Module Introduction

In DVR/DMS products, RGA is used frequently. In order to relieve the pressure of RGA, we use the scaling function of the ISPP module in the RV1126 chip. Therefore, we provide a VP module in rkmedia, through which the scaling function of ISPP can be used.

For details of VP, please refer to the document:

[docs/RV1126_RV1109/Multimedia/Rockchip_Developer_Guide_Linux_RKMedia_CN.pdf](#).

It should be noted that the scaling function of ISPP limits the width of the Buffer and requires 16 Byte alignment.

7. System Performance Optimization

Due to the complicated service logic of DVR products, there may be multi mainstreams or substreams of camera, OSD, algorithm processing and other functions at the same time. System performance often hit bottlenecks. Therefore, this chapter will summarize some regular system performance optimization ways at present, customers should choose according to actual products.

7.1 CPU Performance Optimization

Before CPU performance optimization, you can use top, perf and other common tools to analyze CPU occupation to optimized firstly. In addition, applications should also be used to limit the invalid log printing. Log output also increase CPU consumption.

7.1.1 Turn off High Resoultion Timer

Turn off high resolution timer configuration to reduce CPU consumption. To turn off high resolution timers, you should modify the configuration of kernel. The modification way is as follows:

```
-Config_high_res_timers
```

After modifying the configuration, please re-build kernel.

Note: After the high resolution timer is closed, NPU driver loading may be affected. If the NPU driver fails to load and reports errors, please apply for a NPU driver with a compatible high resolution timer disabled at Redmine.

7.1.2 Turn off IOMMU in ISPP

IOMMU is used to map non-continuous physical memories into continuous physical memories for hardware. IOMMU is enabled in ISPP by default. If using IOMMU, there will be additional CPU usage during memory mapping. You can configure a continuous physical memory for ISPP to achieve the purpose of closing IOMMU. The following patch is a modification example:

```
Diff -Git A/ARCH/ARM/BOOT/DTS/XXX.DTS B/ARCH/ARM/BOOT/DTS/XXX.DTS
index e2501ed7f996..360169b7fdd1 100644
--- A/Arch/ARM/BOOT/DTS/XXX.DTS
+++ B/ARCH/ARM/BOOT/DTS/XXX.DTS
@@ -318,7 +318,15 @@
     };

    & isp_reserved {
-Size = <0x06400000>;
+ size = <0x0b000000>;
+};
+
+& rkispp_mmu {
+ Status = "Disabled";
+};
+
+& rkispp {
+ memory-region = <& isp_reserved>;
+};

    & mipi_csi2 {
```

Note: The size of continuous physical memory size reserved in `isp_reserved` is different from product to product, and a reasonable configuration is needed according to product requirement.

7.2 RGA Performance Optimization

7.2.1 To Enable DMA Buffer Cache

DVR products require using RGA for a large number of scaling and cropping operations. The buffer processed by RGA will be mapped before each operation, and therefore increase CPU consumption. This chapter will provide a method of caching the mapping information of RGA DMA Buffer to reduce CPU resources consumption.

DMA Buffer Cache is enabled in SDKs released after V3.0 by default. If it is an earlier SDK, please use the following patches to enable DMA Buffer Cache(Note: The following three patches are required and should be added one by one according to the serial number).

1. The first patch: 0001-dma-buf-dma-buf-cache-fix-error-case-for-attach-deta.patch

```
From 0423f7d60199d4aaf015924d6dd5983ff90b5998 Mon Sep 17 00:00:00 2001
From: Jianqun Xu <jay.xu@rock-chips.com>
Date: Tue, 28 Dec 2021 14:42:56 +0800
Subject: [PATCH] dma-buf: dma-buf-cache: fix error case for attach / detach

Signed-off-by: Jianqun Xu <jay.xu@rock-chips.com>
Change-Id: I84ff3ac7c1357416bb12ca61aa7134fc652538d6
---
 drivers/dma-buf/dma-buf-cache.c | 13 ++++++-----
 1 file changed, 8 insertions(+), 5 deletions(-)

diff --git a/drivers/dma-buf/dma-buf-cache.c b/drivers/dma-buf/dma-buf-cache.c
index 79f35d359241..7bfa6a5e5fc4 100644
--- a/drivers/dma-buf/dma-buf-cache.c
+++ b/drivers/dma-buf/dma-buf-cache.c
@@ -29,7 +29,7 @@ static int dma_buf_cache_destructor(struct dma_buf *dmabuf, void
*dtor_data)

    mutex_lock(&data->lock);
    list_for_each_entry_safe(cache, tmp, &data->head, list) {
-    if (cache->sg_table)
+    if (!IS_ERR_OR_NULL(cache->sg_table))
        dma_buf_unmap_attachment(cache->attach,
                                cache->sg_table,
                                cache->direction);
@@ -83,6 +83,7 @@ EXPORT_SYMBOL(dma_buf_cache_detach);
    struct dma_buf_attachment *dma_buf_cache_attach(struct dma_buf *dmabuf,
                                                    struct device *dev)
    {
+ struct dma_buf_attachment *attach;
    struct dma_buf_cache_list *data;
    struct dma_buf_cache *cache;

@@ -117,8 +118,13 @@ struct dma_buf_attachment *dma_buf_cache_attach(struct dma_buf
*dmabuf,
    return ERR_PTR(-ENOMEM);

    /* Cache attachment */
- cache->attach = dma_buf_attach(dmabuf, dev);
+ attach = dma_buf_attach(dmabuf, dev);
+ if (IS_ERR_OR_NULL(attach)) {
+     kfree(cache);
+     return attach;

```



```

+ }

+ cache->attach = attach;
+ mutex_lock(&data->lock);
+ list_add(&cache->list, &data->head);
+ mutex_unlock(&data->lock);
@@ -163,9 +169,6 @@ struct sg_table *dma_buf_cache_map_attachment(struct
dma_buf_attachment *attach,
+ cache->sg_table = dma_buf_map_attachment(attach, direction);
+ cache->direction = direction;

- if (!cache->sg_table)
- return ERR_PTR(-ENOMEM);
-
+ return cache->sg_table;
+ }
EXPORT_SYMBOL(dma_buf_cache_map_attachment);

```

2. The second patch: 0001-dma-buf-support-to-cache-dma-buf-attachment.patch

From 04273ec4b3523ed3420e16828b7731c20ec3a167 Mon Sep 17 00:00:00 2001
 From: Jianqun Xu <jay.xu@rock-chips.com>
 Date: Fri, 9 Jul 2021 10:00:21 +0800
 Subject: [PATCH] dma-buf: support to cache dma-buf-attachment

This patch try to fix this issue by caching the dma-buf attachments and stores the cache list to dtor_data of dma-buf structor. The dma-buf attach with cache will try to find cached attachment first and return the valid instance.

This patch also store the deattach operation to dtor of dma-buf structor by dma_buf_set_destructor.

Change-Id: I4778c3328825f6c04f5d2608994e62fe3478bflb
 Signed-off-by: Jianqun Xu <jay.xu@rock-chips.com>
 (cherry picked from commit 36514da674cb71553472a66704eae5981035c44c)

```

drivers/dma-buf/Kconfig          |    7 ++
drivers/dma-buf/Makefile         |    1 +
drivers/dma-buf/dma-buf-cache.c | 171 +++++
include/linux/dma-buf-cache.h    |   32 +++++
4 files changed, 211 insertions(+)
create mode 100644 drivers/dma-buf/dma-buf-cache.c
create mode 100644 include/linux/dma-buf-cache.h

```

```

diff --git a/drivers/dma-buf/Kconfig b/drivers/dma-buf/Kconfig
index cc70e8bee874..414a05a8b7aa 100644
--- a/drivers/dma-buf/Kconfig
+++ b/drivers/dma-buf/Kconfig

```

```

@@ -1,5 +1,12 @@
    menu "DMABUF options"

+config DMABUF_CACHE
+ bool "DMABUF cache attachment"
+ default n
+ help
+   This option support to store attachments in a list and destroy them by
+   set to a callback list in the dtor of dma-buf.
+
    config SYNC_FILE
        bool "Explicit Synchronization Framework"
        default n
diff --git a/drivers/dma-buf/Makefile b/drivers/dma-buf/Makefile
index c33bf8863147..87048e5d5aba 100644
--- a/drivers/dma-buf/Makefile
+++ b/drivers/dma-buf/Makefile
@@ -1,3 +1,4 @@
    obj-y := dma-buf.o dma-fence.o dma-fence-array.o reservation.o seqno-fence.o
+obj-$(CONFIG_DMABUF_CACHE) += dma-buf-cache.o
    obj-$(CONFIG_SYNC_FILE) += sync_file.o
    obj-$(CONFIG_SW_SYNC) += sw_sync.o sync_debug.o
diff --git a/drivers/dma-buf/dma-buf-cache.c b/drivers/dma-buf/dma-buf-cache.c
new file mode 100644
index 000000000000..79f35d359241
--- /dev/null
+++ b/drivers/dma-buf/dma-buf-cache.c
@@ -0,0 +1,171 @@
+// SPDX-License-Identifier: GPL-2.0
+/*
+ * Copyright (c) 2021 Rockchip Electronics Co. Ltd.
+ */
+
+#include <linux/slab.h>
+#include <linux/dma-buf.h>
+#undef CONFIG_DMABUF_CACHE
+#include <linux/dma-buf-cache.h>
+
+struct dma_buf_cache_list {
+ struct list_head head;
+ struct mutex lock;
+};
+
+struct dma_buf_cache {
+ struct list_head list;
+ struct dma_buf_attachment *attach;
+ enum dma_data_direction direction;
+ struct sg_table *sg_table;
+};

```

```

+
+static int dma_buf_cache_destructor(struct dma_buf *dmabuf, void *dtor_data)
+{
+ struct dma_buf_cache_list *data;
+ struct dma_buf_cache *cache, *tmp;
+
+ data = dmabuf->dtor_data;
+
+ mutex_lock(&data->lock);
+ list_for_each_entry_safe(cache, tmp, &data->head, list) {
+     if (cache->sg_table)
+         dma_buf_unmap_attachment(cache->attach,
+             cache->sg_table,
+             cache->direction);
+
+     dma_buf_detach(dmabuf, cache->attach);
+     list_del(&cache->list);
+     kfree(cache);
+ }
+ mutex_unlock(&data->lock);
+
+ kfree(data);
+ return 0;
+}
+
+static struct dma_buf_cache *
+dma_buf_cache_get_cache(struct dma_buf_attachment *attach)
+{
+ struct dma_buf_cache_list *data;
+ struct dma_buf_cache *cache;
+ struct dma_buf *dmabuf = attach->dmabuf;
+
+ if (dmabuf->dtor != dma_buf_cache_destructor)
+     return NULL;
+
+ data = dmabuf->dtor_data;
+
+ mutex_lock(&data->lock);
+ list_for_each_entry(cache, &data->head, list) {
+     if (cache->attach == attach) {
+         mutex_unlock(&data->lock);
+         return cache;
+     }
+ }
+
+ mutex_unlock(&data->lock);
+
+ return NULL;
+}

```

```

+
+void dma_buf_cache_detach(struct dma_buf *dmabuf,
+    struct dma_buf_attachment *attach)
+{
+ struct dma_buf_cache *cache;
+
+ cache = dma_buf_cache_get_cache(attach);
+ if (!cache)
+     dma_buf_detach(dmabuf, attach);
+}
+EXPORT_SYMBOL(dma_buf_cache_detach);
+
+struct dma_buf_attachment *dma_buf_cache_attach(struct dma_buf *dmabuf,
+    struct device *dev)
+{
+ struct dma_buf_cache_list *data;
+ struct dma_buf_cache *cache;
+
+ if (!dmabuf->dtor) {
+     data = kzalloc(sizeof(*data), GFP_KERNEL);
+     if (!data)
+         return ERR_PTR(-ENOMEM);
+
+     mutex_init(&data->lock);
+     INIT_LIST_HEAD(&data->head);
+
+     dma_buf_set_destructor(dmabuf, dma_buf_cache_destructor, data);
+ }
+
+ if (dmabuf->dtor && dmabuf->dtor != dma_buf_cache_destructor)
+     return dma_buf_attach(dmabuf, dev);
+
+ data = dmabuf->dtor_data;
+
+ mutex_lock(&data->lock);
+ list_for_each_entry(cache, &data->head, list) {
+     if (cache->attach->dev == dev) {
+         /* Already attached */
+         mutex_unlock(&data->lock);
+         return cache->attach;
+     }
+ }
+ mutex_unlock(&data->lock);
+
+ cache = kzalloc(sizeof(*cache), GFP_KERNEL);
+ if (!cache)
+     return ERR_PTR(-ENOMEM);
+
+ /* Cache attachment */

```

```

+ cache->attach = dma_buf_attach(dmabuf, dev);
+
+ mutex_lock(&data->lock);
+ list_add(&cache->list, &data->head);
+ mutex_unlock(&data->lock);
+
+ return cache->attach;
+}
+EXPORT_SYMBOL(dma_buf_cache_attach);
+
+void dma_buf_cache_unmap_attachment(struct dma_buf_attachment *attach,
+    struct sg_table *sg_table,
+    enum dma_data_direction direction)
+{
+ struct dma_buf_cache *cache;
+
+ cache = dma_buf_cache_get_cache(attach);
+ if (!cache)
+     dma_buf_unmap_attachment(attach, sg_table, direction);
+}
+EXPORT_SYMBOL(dma_buf_cache_unmap_attachment);
+
+struct sg_table *dma_buf_cache_map_attachment(struct dma_buf_attachment *attach,
+    enum dma_data_direction direction)
+{
+ struct dma_buf_cache *cache;
+
+ cache = dma_buf_cache_get_cache(attach);
+ if (!cache)
+     return dma_buf_map_attachment(attach, direction);
+
+ if (cache->sg_table) {
+     /* Already mapped */
+     if (cache->direction == direction)
+         return cache->sg_table;
+
+     /* Different directions */
+     dma_buf_unmap_attachment(attach, cache->sg_table,
+         cache->direction);
+ }
+
+ /* Cache map */
+ cache->sg_table = dma_buf_map_attachment(attach, direction);
+ cache->direction = direction;
+
+ if (!cache->sg_table)
+     return ERR_PTR(-ENOMEM);
+
+

```

```

+ return cache->sg_table;
+}
+EXPORT_SYMBOL(dma_buf_cache_map_attachment);
diff --git a/include/linux/dma-buf-cache.h b/include/linux/dma-buf-cache.h
new file mode 100644
index 000000000000..d97545560990
--- /dev/null
+++ b/include/linux/dma-buf-cache.h
@@ -0,0 +1,32 @@
+/* SPDX-License-Identifier: GPL-2.0 */
+/*
+ * Copyright (c) 2021 Rockchip Electronics Co. Ltd.
+ */
+#ifndef _LINUX_DMA_BUF_CACHE_H
+#define _LINUX_DMA_BUF_CACHE_H
+
+#include <linux/dma-buf.h>
+
+extern void dma_buf_cache_detach(struct dma_buf *dmabuf,
+                                struct dma_buf_attachment *attach);
+
+extern void dma_buf_cache_unmap_attachment(struct dma_buf_attachment *attach,
+                                           struct sg_table *sg_table,
+                                           enum dma_data_direction direction);
+
+extern struct dma_buf_attachment *
+dma_buf_cache_attach(struct dma_buf *dmabuf, struct device *dev);
+
+extern struct sg_table *
+dma_buf_cache_map_attachment(struct dma_buf_attachment *attach,
+                             enum dma_data_direction direction);
+
+#ifdef CONFIG_DMABUF_CACHE
+/* Replace dma-buf apis to cached apis */
+#define dma_buf_attach dma_buf_cache_attach
+#define dma_buf_detach dma_buf_cache_detach
+#define dma_buf_map_attachment dma_buf_cache_map_attachment
+#define dma_buf_unmap_attachment dma_buf_cache_unmap_attachment
+#endif
+
+#endif /* _LINUX_DMA_BUF_CACHE_H */

```

3. The third patch

```

- a/drivers/video/rockchip/rga2/rga2_drv.c
+++ b/drivers/video/rockchip/rga2/rga2_drv.c
@@ -45,7 +45,7 @@

    #if (LINUX_VERSION_CODE >= KERNEL_VERSION(4, 4, 0))
    #include <linux/pm_runtime.h>
-#include <linux/dma-buf.h>
+#include <linux/dma-buf-cache.h>
    #endif

```

Enable the following configuration in kernel:

```
CONFIG_DMABUF_CACHE=y
```

You can add log printing to confirm whether RGA DMA-Cache takes effect:

```

--- a/drivers/dma-buf/dma-buf-cache.c
+++ b/drivers/dma-buf/dma-buf-cache.c
@@ -61,6 +61,7 @@ dma_buf_cache_get_cache(struct dma_buf_attachment *attach)
    list_for_each_entry(cache, &data->head, list) {
        if (cache->attach == attach) {
            mutex_unlock(&data->lock);
+           printk("%s %d\n", __func__, __LINE__);
            return cache;
        }
    }
}

```

Note: After confirming that the DMA-Cache of RGA is effective, the debugging log should be deleted.

8. FAQ

8.1 Pictures are Staggered When Hot Plugging



8.1.1 Solution

Add the following configuration in dts to solve the hot plug problem:

```
&rkcif_mipi_lvds {
    ....
    rockchip,cif-monitor = <3 2 25 1000 5>;
    ....
};
```

8.2 Support Interlaced Frame Composite for CVBS

There are limitations for "Interlaced Composite" function:

1. MIPI Interface: it should output frame count number (from frame start and frame end short packets) for parity judgement;
2. BT656 Interface: it should output standard SAV/EAV, which has bit 6 for parity judgement;
3. In our driver, only selfpath device support this function, otherwise it will output the following error log:

```
only selfpath support interlaced
```

You can use the following command to check which device is selfpath:

```
media-ctl -p
```

```
entity 3: rkisp_selfpath (1 pad, 1 link)
    type Node subtype V4L flags 0
    device node name /dev/video1
    pad0: Sink
        <- "rkisp-isp-subdev":2 [ENABLED]
```

Modification In Driver

You should set `format.field` to `V4L2_FIELD_INTERLACED` in your driver in `xxxx_get_fmt` function. `driver/media/i2c/tc35874x.c` would be a nice reference.