

Rockchip RV1106 RV1103 性能优化指南

文件标识：RK-YH-YF-A05

发布版本：V1.0.0

日期：2023-05-18

文件密级：☐绝密 ☐秘密 ☐内部资料 ☒公开

免责声明

本文档按“现状”提供，瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自拥有者所有。

版权所有 © 2023 瑞芯微电子股份有限公司

超越合理使用范畴，非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址：福建省福州市铜盘路软件园A区18号

网址：www.rock-chips.com

客户服务电话：+86-4007-700-590

客户服务传真：+86-591-83951833

客户服务邮箱：fae@rock-chips.com

前言

概述

本文旨在指导工程师如何在RV1103、RV1106平台上进行内存和性能优化。

产品版本

芯片名称	内核版本
RV1103、RV1106	Linux 5.10

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

修订记录

版本号	作者	修改日期	修改说明
V1.0.0	jkand.huang	2023-5-18	初始版本

目录

Rockchip RV1106 RV1103 性能优化指南

- 1. 内存裁剪
 - 1.1 ISP
 - 1.1.1 ISP 省内存/带宽模式
 - 1.1.2 IQ 文件使用bin 文件
 - 1.1.3 ISP 在线模式
 - 1.1.4 ISP 离线模式
 - 1.2 VI
 - 1.2.1 用卷绕模式
 - 1.2.2 输出buffer 配置
 - 1.3 VENC
 - 1.3.1 设置编码通道参考帧共享属性
 - 1.3.2 NormalP
 - 1.3.3 智能编码
 - 1.3.4 确认码流输出buffer
 - 1.3.5 设置编码通道Combo属性
 - 1.4 RGN
 - 1.4.1 OSD 使用2bpp
 - 1.4.2 OSD buffer个数
 - 1.5 AVS
 - 1.5.1 提前AVS 的初始化
 - 1.5.2 输出buffer
 - 1.6 system
 - 1.6.1 目前剩余多少CMA，根据实际调整大小
 - 1.6.2 内核配置选项
 - 1.7 典型案例
 - 1.7.1 RV1103 双目双码流

1. 内存裁剪

1.1 ISP

1.1.1 ISP 省内存/带宽模式

说明：

0：不开启，1:开启。默认值为0。

- 正常线性模式，此参数可设置为1。
- hdr模式，此参数建议设置为0，以图像效果为优先。

详见：

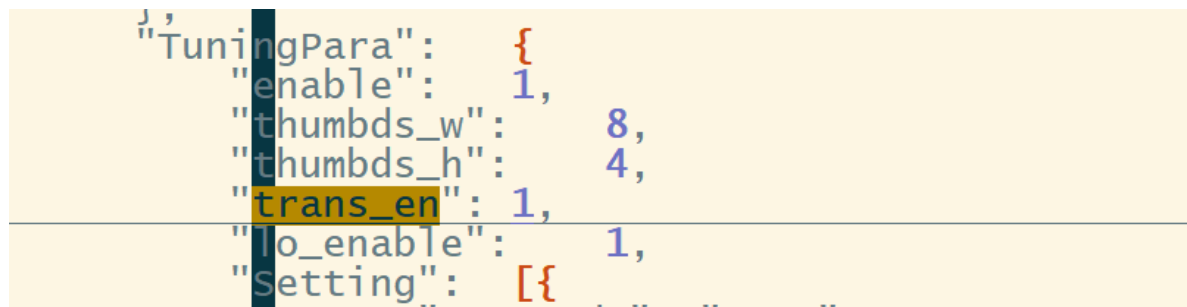
Rockchip_Tuning_Guide_ISP32_CN_v0.1.2.pdf、Rockchip_Development_Guide_ISP32_CN_v0.1.2.pdf

查看：

```
# cat /proc/rkisp-vir0 |grep BAY3D
BAY3D      ON(0xa0012001 0x90089) bypass:0 bwsaving:1 mode:(104x8 sram)
```

修改方法：

IQ（效果文件）文件中：trans_en：1，如下：



```
{
  "TuningPara": {
    "enable": 1,
    "thumbs_w": 8,
    "thumbs_h": 4,
    "trans_en": 1,
    "to_enable": 1,
    "Setting": [{
      "name": "trans_en",
      "value": 1
    }, {
      "name": "to_enable",
      "value": 1
    }
  ]
}
```

1.1.2 IQ 文件使用bin 文件

说明： bin 文件为json文件的紧凑格式，更节省内存。缺点：无法直接修改，不便于调试。

查看： 确认机器中使用的IQ效果文件是*.bin，如下：

```
# ls /etc/iqfiles/sc4336_OT01_40IRC_F16.bin
/etc/iqfiles/sc4336_OT01_40IRC_F16.bin
```

修改方法：

方法一： SDK 编译完后会将json转bin，将bin文件放到机器对应的位置即可，如下：

```
# ls media/isp/out/isp_iqfiles/sc4336_OT01_40IRC_F16*
media/isp/out/isp_iqfiles/sc4336_OT01_40IRC_F16.bin
media/isp/out/isp_iqfiles/sc4336_OT01_40IRC_F16.json
```

方法二：使用j2s4b工具，将json文件转为bin文件，如下：

```
./media/isp/release_camera_engine_rkaiq_rv1106_arm-rockchip830-linux-
uclibcgnueabihost/j2s4b
output/out/media_out/isp_iqfiles/sc4336_OT01_40IRC_F16.json
./sc4336_OT01_40IRC_F16.bin
```

1.1.3 ISP 在线模式

说明： vicap 采集图像完成之后，不过DDR直接输出给ISP。HDR和双目无法使用。

查看：

```
# cat /proc/rkisp-vir0 |grep working
Isp online frame:3675 working time:33ms v-blank:397us
```

1.1.4 ISP 离线模式

说明： vicap采集图像后，经过DDR输出给ISP。

默认为4个buffer，建议修改成2个，如果对帧率要求不高可以尝试修改成1个。

查看：

```
# cat /proc/rkisp-vir0 |grep working
Isp Read frame:3675 working time:33ms v-blank:397us
```

修改方法：

rk_aiq_uapi2_sysctl_preInit_devBufCnt

```
rk_aiq_uapi2_sysctl_preInit_devBufCnt(aiq_static_info.sensor_info.sensor_name,
                                       "rkraw_rx", 2);
```

参考代码：media/samples/example/common/isp3.x/sample_comm_isp.c

1.2 VI

1.2.1 用卷绕模式

说明：

卷绕模式是指ISP 未完全输出一帧，编码器就开始编码，一个你追我赶的过程，所以相比普通模式不需要完整的YUV。

功能限制：

- 不支持mirror/flip 功能，可以使用sensor 的mirror/flip。
- 不支持从通道中获取YUV。
- 不支持帧率控制（SDK V1.5.1 及之前版本）。

详见：Rockchip_Developer_Guide_MPI.pdf，以下两个接口有详细的说明：

```
RK_MPI_VENC_SetChnBufWrapAttr  
RK_MPI_VI_SetChnWrapBufAttr
```

查看：

如下Online 为 1 则是卷绕模式：

```
# cat /proc/vcodec/enc/venc_info|grep Online -A 2  
      ID|   Width|  Height|  Type| ByFrame|  Sequence|   GopMode|  Prio|  
MaxWidth| MaxHeight|  Online|  RefShare  
      0|   1920|   1080|  h264|      y|      16|  normalp|   0|  
  1920|   1080|     1|      1
```

修改方法：

参考代码：

```
media/samples/simple_test/simple_vi_venc_wrap.c
```

1.2.2 输出buffer 配置

VI_ISP_OPT_S:u32BufCount 非卷绕下建议配置2 个，帧率要求不高可以使用1个。

1.3 VENC

1.3.1 设置编码通道参考帧共享属性

说明：不支持超大帧重编功能。

查看：

如下RefShare 为1则是共享参考帧：

```
# cat /proc/vcodec/enc/venc_info|grep RefShare -A 2  
      ID|   Width|  Height|  Type| ByFrame|  Sequence|   GopMode|  Prio|  
MaxWidth| MaxHeight|  Online|  RefShare  
      0|   1920|   1080|  h264|      y|     8225|  normalp|   0|  
  1920|   1080|     1|      1
```

修改方法：RK_MPI_VENC_SetChnRefBufShareAttr

```

VENC_CHN_REF_BUF_SHARE_S stVencChnRefBufShare;
memset(&stVencChnRefBufShare, 0, sizeof(VENC_CHN_REF_BUF_SHARE_S));
stVencChnRefBufShare.bEnable = 1;
s32Ret = RK_MPI_VENC_SetChnRefBufShareAttr(ctx->s32ChnId,
&stVencChnRefBufShare);
if (s32Ret != RK_SUCCESS) {
    printf("RK_MPI_VENC_SetChnRefBufShareAttr failed,ret=0x%x\n", s32Ret);
}

```

1.3.2 NormalP

说明：

编码使用单参考帧，可以节省一张yuv。

修改：

VENC_GOP_MODE_E 设置为VENC_GOPMODE_NORMALP 不开启。

1.3.3 智能编码

说明：

1440p: 可以节省120KB左右。

使能这两项功能可以降低码率和减轻运动拖尾，但是会增加120KB左右内存，建议开启。

详见:Rockchip_Developer_Guide_MPI.pdf 接口说明。

修改：

RK_MPI_VENC_EnableSvc：可降低码率

RK_MPI_VENC_EnableMotionDeblur：减轻运动拖尾

1.3.4 确认码流输出buffer

说明：

保存码流数据（ring_buffer大小），该buffer如果太小则会overflow，用户可根据码率、图片质量灵活调整。

查看：

```

# cat /proc/vcodec/enc/venc_info |grep "ring buf status" -A 3
-----ring buf status-----
-----
ID|   w_pos|   r_pos|  usd_len| total_len|  min_size|   l_w_pos|   l_r_pos
0| 2106824| 2106824| 495263| 2330624|    10240| 2106824| 2106824

```

修改：

方法：VENC_ATTR_S：u32BufSize（推荐值 w*h/3，不小于一个Gop 内码率总大小，如果出现串口刷overflow 将该值调整大）。

1.3.5 设置编码通道Combo属性

说明：

H264/H265 输出一帧编码数据的时候，可以同时输出一张jpeg 图像，节省对YUV占用，特别是在卷绕模式下体现更优。

修改：

RK_MPI_VENC_SetComboAttr

1.4 RGN

1.4.1 OSD 使用2bpp

说明：

OSD 数据使用2bit来表示，所以同时可有两个颜色、透明非透明选项，该功能一般用于人形框。

文档：Rockchip_Developer_Guide_MPI.pdf：表1-5 2BPP格式颜色配置说明。

修改：

画框参考代码：project/app/rkipc/rkipc/src/rv1106_ipc/video/video.c draw_rect_2bpp 函数。

1.4.2 OSD buffer个数

说明：

osd 内存轮转buffer，默认为两个。

修改：

u32CanvasNum = 0 时，两个buffer。

u32CanvasNum = 1 时，一个buffer，但是仅支持 attach 到一个通道上。

1.5 AVS

1.5.1 提前AVS 的初始化

说明：

AVS初始化需要内存来保存中间数据，初始化完之后会释放，所以将初始化放到应用最开始位置，避免峰值内存。

修改：

示例代码：media/samples/example/demo/sample_demo_vi_avs_venc.c

将create和start 分开，create放在最开始位置：

SAMPLE_COMM_AVS_CreateGrp、SAMPLE_COMM_AVS_StartGrp

1.5.2 输出buffer

说明：

AVS支持两个通道的输出，通道1会从通道0缩放下来，如果业务需求两个分辨率就可以满足需求，建议直接使用AVS的两个通道。

输出buffer个数建议两个，帧率要求低可以尝试一个。

1.6 system

1.6.1 目前剩余多少CMA，根据实际情况调整大小

说明：

系统预留出来给媒体用的连续物理内存。

查看：

```
# cat /proc/rk_dma_heap/alloc_bitmap
Total: 67584 KiB
Used: 55436 KiB
```

修改：

方法一：

uboot 下设置命令，如下将cma从66M 改成 40M，注意 setenv sys_bootargs 时中间是不带等号，其他参数不改。

```
=> printenv sys_bootargs
sys_bootargs=ubi.mtd=4 root=ubi0:rootfs rootfstype=ubifs rk_dma_heap_cma=66M
=> setenv sys_bootargs ubi.mtd=4 root=ubi0:rootfs rootfstype=ubifs
rk_dma_heap_cma=40M
=> saveenv
Saving Environment to envf...
=> printenv sys_bootargs
sys_bootargs=ubi.mtd=4 root=ubi0:rootfs rootfstype=ubifs rk_dma_heap_cma=40M
```

方法二：

SDK 的板级配置文件project/cfg/目录下，BoardConfig.mk export RK_BOOTARGS_CMA_SIZE="40M"

build.sh env

烧录evn.img

1.6.2 内核配置选项

说明：

精简内核配置选项，可以参考使用。

修改：

1.7 典型案例

1.7.1 RV1103 双目双码流

demo 源码：

cat /proc/rk_dma_heap/dma_heap_info 如下

cif:

isp:

venc:

osd:

```
---debug_rk /proc/rk_dma_heap/dma_heap_info ---
dma-heap:<rk-dma-heap-cma> -dmabuf i_ino = 120
    Alloc by (vmpi                ) [0x0620f000-0x06216fff] 0x00008000 (32 KiB)
    Attached Devices:
Total 0 devices attached

dma-heap:<rk-dma-heap-cma> -dmabuf i_ino = 119
    Alloc by (vmpi                ) [0x06207000-0x0620efff] 0x00008000 (32 KiB)
    Attached Devices:
Total 0 devices attached

dma-heap:<rk-dma-heap-cma> -dmabuf i_ino = 118
    Alloc by (vmpi                ) [0x05f0f000-0x06206fff] 0x002f8000 (3040
KiB)
    Attached Devices:
Total 0 devices attached

dma-heap:<rk-dma-heap-cma> -dmabuf i_ino = 117
    Alloc by (vmpi                ) [0x05f0e000-0x05f0efff] 0x00001000 (4 KiB)
    Attached Devices:
Total 0 devices attached

dma-heap:<rk-dma-heap-cma> -dmabuf i_ino = 116
    Alloc by (vmpi                ) [0x05eee000-0x05f0dfff] 0x00020000 (128
KiB)
    Attached Devices:
Total 0 devices attached

dma-heap:<rk-dma-heap-cma> -dmabuf i_ino = 115
    Alloc by (vmpi                ) [0x05eed000-0x05eedfff] 0x00001000 (4 KiB)
    Attached Devices:
Total 0 devices attached

dma-heap:<rk-dma-heap-cma> -dmabuf i_ino = 114
    Alloc by (vmpi                ) [0x05ecd000-0x05eecfff] 0x00020000 (128
KiB)
    Attached Devices:
Total 0 devices attached
```

```
dma-heap:<rk-dma-heap-cma> -dmabuf i_ino = 113
  Alloc by (vmpi          ) [0x05b3d000-0x05eccfff] 0x00390000 (3648
KiB)
  Attached Devices:
Total 0 devices attached

dma-heap:<rk-dma-heap-cma> -dmabuf i_ino = 112
  Alloc by (vmpi          ) [0x0583c000-0x05b38fff] 0x002fd000 (3060
KiB)
  Attached Devices:
  ffa00000.rkisp
  ffa00000.rkisp
  ffa00000.rkisp
  ffa00000.rkisp
Total 4 devices attached

dma-heap:<rk-dma-heap-cma> -dmabuf i_ino = 111
  Alloc by (vmpi          ) [0x05544000-0x0583bfff] 0x002f8000 (3040
KiB)
  Attached Devices:
Total 0 devices attached

dma-heap:<rk-dma-heap-cma> -dmabuf i_ino = 110
  Alloc by (vmpi          ) [0x05543000-0x05543fff] 0x00001000 (4 KiB)
  Attached Devices:
Total 0 devices attached

dma-heap:<rk-dma-heap-cma> -dmabuf i_ino = 109
  Alloc by (vmpi          ) [0x05523000-0x05542fff] 0x00020000 (128
KiB)
  Attached Devices:
Total 0 devices attached

dma-heap:<rk-dma-heap-cma> -dmabuf i_ino = 108
  Alloc by (vmpi          ) [0x05522000-0x05522fff] 0x00001000 (4 KiB)
  Attached Devices:
Total 0 devices attached

dma-heap:<rk-dma-heap-cma> -dmabuf i_ino = 107
  Alloc by (vmpi          ) [0x05502000-0x05521fff] 0x00020000 (128
KiB)
  Attached Devices:
Total 0 devices attached

dma-heap:<rk-dma-heap-cma> -dmabuf i_ino = 106
  Alloc by (vmpi          ) [0x05172000-0x05501fff] 0x00390000 (3648
KiB)
  Attached Devices:
Total 0 devices attached

dma-heap:<rk-dma-heap-cma> -dmabuf i_ino = 105
  Alloc by (vmpi          ) [0x04e71000-0x0516dfff] 0x002fd000 (3060
KiB)
  Attached Devices:
  ffa00000.rkisp
  ffa00000.rkisp
  ffa00000.rkisp
```

```

        ffa00000.rkisp
Total 4 devices attached

dma-heap:<rk-dma-heap-cma> -dmabuf i_ino = 1
    Alloc by (vmpi                ) [0x04200000-0x04202fff] 0x00003000 (12 KiB)
    Attached Devices:
Total 0 devices attached

dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (ffa10000.rkcif       ) [0x04203000-0x044aafff] 0x002a8000 (2720
KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (ffa00000.rkisp       ) [0x044ab000-0x044abfff] 0x00001000 (4 KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (ffa00000.rkisp       ) [0x044ac000-0x044acfff] 0x00001000 (4 KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (ffa00000.rkisp       ) [0x044ad000-0x047e9fff] 0x0033d000 (3316
KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (ffa00000.rkisp       ) [0x047ea000-0x04829fff] 0x00040000 (256
KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (rkisp-vir0           ) [0x0482a000-0x0482dfff] 0x00004000 (16 KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (rkisp-vir0           ) [0x0482e000-0x04831fff] 0x00004000 (16 KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (rkisp-vir0           ) [0x04832000-0x04835fff] 0x00004000 (16 KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (rkisp-vir0           ) [0x04836000-0x04839fff] 0x00004000 (16 KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (ffa10000.rkcif       ) [0x0483a000-0x04ae1fff] 0x002a8000 (2720
KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (ffa00000.rkisp       ) [0x04ae2000-0x04ae2fff] 0x00001000 (4 KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (ffa00000.rkisp       ) [0x04ae3000-0x04ae3fff] 0x00001000 (4 KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (ffa00000.rkisp       ) [0x04ae4000-0x04ae20fff] 0x0033d000 (3316
KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (ffa00000.rkisp       ) [0x04e21000-0x04e60fff] 0x00040000 (256
KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (rkisp-vir1           ) [0x04e61000-0x04e64fff] 0x00004000 (16 KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (rkisp-vir1           ) [0x04e65000-0x04e68fff] 0x00004000 (16 KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (rkisp-vir1           ) [0x04e69000-0x04e6cfff] 0x00004000 (16 KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (rkisp-vir1           ) [0x04e6d000-0x04e70fff] 0x00004000 (16 KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (ffa00000.rkisp       ) [0x0516e000-0x05171fff] 0x00004000 (16 KiB)
dma-heap:<rk-dma-heap-cma> -non dmabuf
    Alloc by (ffa00000.rkisp       ) [0x05b39000-0x05b3cfff] 0x00004000 (16 KiB)

Total : 0x2017000 (32860 KiB)

```

