

君正®

T41 SDK 调试指南

Date: 2023-12



北京君正集成电路股份有限公司
Ingenic Semiconductor Co., Ltd.

Copyright © 2005-2022 Ingenic Semiconductor Co. Ltd. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Ingenic Semiconductor Co. Ltd.

Trademarks and Permissions



、 Ingenic and Ingenic icons are trademarks of Ingenic Semiconductor Co.Ltd. All other trademarks and trade names mentioned in this document are the property of their respective holders.

Disclaimer

All the deliverables and data in this folder serve only as a reference for customer development. Please read through this disclaimer carefully before you use the deliverables and data in this folder. You may use the deliverables in this folder or not. However, by using the deliverables and data in this folder, you agree to accept all the content in this disclaimer unconditional and irrevocable. If you do not find the content in this disclaimer reasonable, you shall not use the deliverables and data in this folder.

The deliverables and data in this folder are provided "AS IS" without representations, guarantees or warranties of any kind (either express or implied). To the maximum extent permitted by law, Ingenic Semiconductor Co., Ltd (Ingenic) provides the deliverables and data in this folder without implied representations, guarantees or warranties, including but not limited to implied representations, guarantees and warranties of merchantability, non-infringement, or fitness for a particular purpose. Deviation of the data provided in this folder may exist under different test environments.

Ingenic takes no liability or legal responsibility for any design and development error, incident, negligence, infringement, and loss (including but not limited to any direct, indirect, consequential, or incidental loss) caused by the use of data in this folder. Users shall be responsible for all risks and consequences caused by the use of data in this folder.

北京君正集成电路股份有限公司

地址：北京市海淀区西北旺东路 10 号院东区 14 号楼君正大厦

电话：**(86-10)56345000**

传真：**(86-10)56345001**

Http: //www.ingenic.com.cn

前言

概述

本文为 Ingenic-T41 SDK 调试指南,为用户在使用 DEMO 板的过程中遇到问题提供帮助和解决方法。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
T41 SDK 调试指南	V2.1

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

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

日期	版本	修订章节
2022-01	1.0	第一次正式版本发布
2022-08	1.1	2 帧率不足分析 新增 2.3 IVDC 下的相关统计 5 功能优化 新增 5.8 查看模块之间的绑定设置
2023-07	2.0	新增 第 6 章内存优化
2023-12	2.1	新增 第 7 章调试工具

目录

目录.....	2
1 图像异常分析.....	4
1.1 如何抓取 RAW 原始数据?	4
1.2 如何抓取 nv12 图?	4
1.3 如何获取编码裸流?	5
2 帧率不足分析.....	6
2.1 sensor 帧率信息.....	6
2.2 isp 帧率信息.....	6
2.3 IVDC 相关帧率信息.....	6
2.4 编码帧率信息.....	7
2.5 模块时钟.....	7
3 图像问题.....	8
3.1 如何解决图像颜色不正常?	8
3.2 如何解决图像颜色偏绿?	8
3.3 如何解决图像颜色偏红?	9
3.4 如何解决图像夜视异常?	9
3.5 如何解决没有图像?	9
3.6 如何解决图像条纹?	9
3.7 如何解决图像闪烁?	9
4 MSC.....	10
4.1 SD 卡分区自动分配.....	10
4.2 关于卡检测驱动分析.....	10
4.3 为什么卡启 uboot 需要跳 17K.....	10
5 功能优化.....	11
5.1 如何调整 nna 优先级?	11
5.2 如何调整 DDR 的优先级?	11
5.3 如何绑定中断到对应的核上?	11
5.4 如何读写 sensor 寄存器?	11
5.5 查看模块之间的绑定设置.....	11
5.6 查看 FrameSource 参数设置.....	12
5.7 如何保留&恢复现场.....	12
5.8 如何抓取 Log.....	12
5.9 如何读写 eth phy 寄存器.....	13
6 内存优化.....	14

6.1 Rmem 查看与设置	14
6.2 查看内存池使用情况	15
6.3 查看 mem 使用情况	15
6.4 nmem 使用查看	15
6.5 内存节省方式-直通功能	15
6.6 其他节省内存方式	16
7 调试工具	17
7.1 Coredump	17
7.2 gdbserver	18
7.3 IMPSDK 调试命令	20

1 图像异常分析

1.1 如何抓取 RAW 原始数据？

【分析】

当视频流出现异常时，抓取 raw 原始数据可以辅助判断是编码前还是编码后导致的图像问题。

Raw 原始数据正常：编码问题

Raw 原始数据异常：ISP/sensor 端异常

【解决】

命令：抓取 raw 数据

```
echo saveraw savenum >/proc/jz/isp/isp-w02
```

```
echo snapraw >/proc/jz/isp/isp-w02
```

抓取的 raw 数据，存放为/tmp/snap.raw，第二种方式需要额外分配 $ispmem=W*H*2$ ，但两种方式放到文件系统下，都需缓存，使用 mem。

1.2 如何抓取 nv12 图？

【分析】

当视频花屏或者图像出现问题时，抓取 nv12 图可以辅助判断是编码前还是编码后导致的图像问题。

【注意事项】

此方法只在非直通下有效，直通下不能使用该命令，直通时 isp 直接把数据给编码，不经过其他模块，只能获取直通给编码的 NV12 数据。

【解决】

命令：抓取 nv12

```
mkdir /tmp/mountdir
```

```
主码流：touch /tmp/mountdir/ispsnap0.nv12
```

```
次码流：touch /tmp/mountdir/ispsnap1.nv12
```

获取编码的 NV12

```
主码流：touch /tmp/mountdir/encsnap0.nv12
```

```
次码流：touch /tmp/mountdir/encsnap1.nv12
```

获取 osd 叠加后的 NV12

主码流: touch /tmp/mountdir/osdsnap0.nv12

次码流: touch /tmp/mountdir/osdsnap1.nv12

直通下获取 IVDC 后的 NV12 图片:

echo snapyuv > /proc/jz/isp/isp-ivdc

1.3 如何获取编码裸流?

【解决】

touch /mnt/mountdir/dump.bs //先建立 dump.bs 文件, 然后重新运行程序, 会产生裸流。

2 帧率不足分析

2.1 sensor 帧率信息

【功能】

- (1) 查看 sensor 的帧数统计。
- (2) 可检查图像是否缺帧。

【解决】

(1) `cat /proc/jz/isp/isp-w02;sleep10;cat /proc/jz/isp/isp-w02 //` 查看 sensor 输出的帧率。

2.2 isp 帧率信息

【功能】

- (1) 获取图像基本信息（帧数、驱动版本、ISP 工作模式等）。
- (2) 可检查图像是否缺帧。

【解决】

命令：

```
T21/T31/T40/T41: cat /proc/jz/isp/isp-m0
```

isp 处理完成的帧数

```
while true;do cat /proc/jz/isp/isp-m0 | grep debug;sleep 1;done
```

【补充】

查看 ISP clk 频率：

```
cat /sys/module/tx_isp_t41/parameters/isp_clk
```

查看 sensor data_interface : 1:Mipi 0:DVP

```
cat /sys/module/sensor_imx335_t41/parameters/data_interface
```

2.3 IVDC 相关帧率信息

【功能】

查看 IVDC 的基本信息

【解决】

```
cat /proc/jz/isp/isp-ivdc
```

2.4 编码帧率信息

【功能】

检查是否有码流产生，tf 代表编码完成的，df 代表编码丢失的。

【解决】

```
while true;do ./impdbg --enc_info | grep CHANNEL;sleep 1;done
```

2.5 模块时钟

【功能】

查看模块时钟频率。

【解决】

```
cat /proc/jz/clock/clocks
```

【补充】

建议:

T41 主要频率为以下值:

ISPM clock 不超过 300M

ISPA clock 不超过 600M

ISPS clock 不超过 600M

VPU clock 不超过 550M

DDR clock 不超过 750M (T41N)

DDR clock 不超过 700M (T41NQ/XQ)

DDR clock 不超过 600M (T41L/LQ)

T40 主要频率为以下值:

ISP clock 不超过 350M

VPU clock 不超过 654M

DDR clock 不超过 700M (T40XP)

DDR clock 不超过 500M (T40N/NN)

3 图像问题

3.1 如何解决图像颜色不正常？

【分析】

可能是 Sensor Data 线对齐方式设置错误。

【解决】

需要根据 SCH 的设计，insmod 时加入参数。

例：\$ insmod sensor_jxh42.ko sensor_gpio_func=1 ，其中参数值的意义为：

- 0: GPIO-A 低 10bit
- 1: GPIO-A 高 10bit
- 2: GPIO-A 低 12bit
- 3: GPIO-A 低 8bit
- 4: GPIO-A 高 8bit

3.2 如何解决图像颜色偏绿？

【分析】

可能是没有在文件系统中放置 bin 文件(/etc/sensor/xxx.bin)，在此情况下串口打印：no bin file on the system!!! 或者 Sensor bin 文件与 ISP driver 不匹配，会出现如下打印：

```
#####  
##  
#### The version of ov9750.bin doesn't match with driver! ####  
#### The version of ov9750.bin is xxx, the driver is yyy ####  
#####  
##
```

【解决】

需要在文件系统的/etc/sensor 目录下放置正确的 bin 文件。

3.3 如何解决图像颜色偏红？

【分析】

可能是没有加 IR-Cut

3.4 如何解决图像夜视异常？

【分析】

- (1) 是否使用对应的夜视 setting
- (2) 是否将 IR-CUT 切换到白片
- (3) 红外灯是否打开

3.5 如何解决没有图像？

【分析】

- (1) 检查硬件的连接是否良好，可以尝试更换软排线或者 sensor 板
- (2) 是否加载对应 sensor 的驱动 ko
- (3) 分辨率、裁剪、缩放等配置是否正确

3.6 如何解决图像条纹？

【分析】

- (1) 检查 sensor 初始化 setting 是否正确
- (2) 检查 sensor 板温度是否正常
- (3) sensor 的 AVDD 以及 DVDD 是否正确，AGAIN 是否设置过大

3.7 如何解决图像闪烁？

【分析】

- (1) 判断是工频干扰的条纹还是整帧闪烁
- (2) 检查 sensor 驱动中配置的 delay time 是否正确
- (3) 检查 sensor 驱动中 gain 的转换算法是否正确

4 MSC

4.1 SD 卡分区自动分配

Uboot/board/ingenic/isvp_t41/partitions.tab

4.2 关于卡检测驱动分析

【分析】

通常 CD 脚控制 mos 管会控制 SD 卡的供电。（硬件处理）

软件上识别卡，添加卡的一系列操作：

驱动中通过 setup_timer 设置定时器，一直执行 jzmmc_detect_change 函数判断 CD 脚状态，当 CD 脚状态变化即可被检测到。

4.3 为什么卡启 uboot 需要跳 17K

【分析】

GPT 分区表信息最大不会超过 17K 且 GPT 分区表信息的 uboot 可以直接为 SD 卡进行分区。

5 功能优化

5.1 如何调整 nna 优先级？

【解决】

```
devmem 0x1301206c 32 0
devmem 0x13012038 32 0x88404002
```

5.2 如何调整 DDR 的优先级？

【解决】

```
devmem 0x1301206c 32 0
devmem 0x13012028 32 0x88404007
devmem 0x13012038 32 0x88404007
```

5.3 如何绑定中断到对应的核上？

【解决】

```
echo 2 > /proc/irq/对应的中断号/irq_affinity
```

5.4 如何读写 sensor 寄存器？

【解决】

```
读取：echo r sen_reg 0x0a >/proc/jz/isp/isp-w00; cat /proc/jz/isp/isp-w00
```

```
写入：echo w sen_reg 0x0a 0x01>/proc/jz/isp/isp-w00
```

5.5 查看模块之间的绑定设置

【解决】

```
impdbg --sys
impdbg --system_info
```

5.6 查看 FrameSource 参数设置

【解决】

```
impdbg --fs //查看 framesource 通道信息
touch /tmp/fsattr;logcat //查看 framesource 参数配置
```

5.7 如何保留&恢复现场

【解决】

对于 Nor Flash, 可以把 Flash 内容保存下来。方法如下:

```
$ dd if=/dev/mtd0 of=/tmp/mtd0
$ dd if=/dev/mtd1 of=/tmp/mtd1
$ dd if=/dev/mtd2 of=/tmp/mtd2
$ dd if=/dev/mtd3 of=/tmp/mtd3
$ cd tmp
$ cat mtd0 mtd1 mtd2 mtd3 > flash.bin
$ rm mtd0 mtd1 mtd2 mtd3
```

把 flash.bin 取出即可

或者:

```
$ cat /dev/mtd0 /dev/mtd1 /dev/mtd2 /dev/mtd3 > flash.bin
```

在需要恢复现场时, 在 u-boot 中把 Nor Flash 完全擦除, 再把 flash.bin 完全写入即可。

对于 Nand Flash, NAND 打包整体和 NOR 并没有太大区别, 不过注意生成的整包固件最好比 flash 的容量小。

5.8 如何抓取 Log

【解决】

出现问题时, 需要抓取 kernel 以及 lib 的 log。

kernel log 抓取命令:

```
$ dmesg > /tmp/time.dmesg。
```

lib log 抓取方法:

```
$ logcat -c time > /tmp/time.log
```

命令大概运行 3 秒后 ctrl+c 终止程序 (logcat 命令在 sdk tools 目录下)。

5.9 如何读写 eth phy 寄存器

【解决】

ETH-PHY 是 IEEE802.3 中定义的一个标准模块，可以对 PHY 的行为、状态进行管理和控制，而具体管理和控制动作是通过读写 PHY 内部的寄存器实现的。

Uboot 环境下：

读格式：ethphy read addr

```
isvp_t41# ethphy read 0x0
```

写格式：ethphy write addr data

```
isvp_t41# ethphy write 0x0 0x0
```

Kernel 环境下：

读格式：reg:phy 地址-phy 寄存器地址(phy 地址默认是 0，phy 寄存器地址根据标准手册查看)

```
$ echo r reg:0-0x0 > /proc/jz/mdio/cmd
```

写格式：reg:phy 地址-phy 寄存器地址-写入数据

```
$ echo w reg:0-0x0-0x0 > /proc/jz/mdio/cmd
```

6 内存优化

6.1 Rmem 查看与设置

1) 内存的使用说明

在 uboot 的 cmdline 中有下面内存配置

```
mem=42012K@0x0 ispmem=8100K@0x2907000 rmem=15424K@0x30F0000
```

mem 是分配给 linux 系统使用的内存。

rmem 是 SDK 保留使用的内存，使用该内存的地方包括，ISP dmaout 的轮转 buf、编码模块的输入输出 buf，OSD 模块的叠加图片。

2) rmem 内存大小如何确定

A. rmem 的内存可以使用 SDK 中的 rmem 计算器 excel 表格来计算得到。

B. 先把 rmem 设置大一点，先让系统跑起来，然后执行 `impdbg --sys;logcat` 可以看到 rmem 实际使用的内存和剩余内存，然后把剩余内存从 rmem 拿出来给 mem 使用。

3) rmem 内存的优化

OSD 内存占用默认是 1M，可以根据实际产品功能调小，例如 512K。可以通过下面接口设置 OSD 的 rmem 的内存占用。OSD 占用的内存大小是叠加最大图片的 $width*height*4$ 。

内存优化调用(默认是 1M)

```
extern "C" {  
    //设置 OSD 模块使用大小  
    extern int IMP_OSD_SetPoolSize(int size);  
}  
初始化 SDK 之前调用  
IMP_OSD_SetPoolSize(512*1024);
```

4) Rmem 使用详细信息

【解决】

```
./impdbg --sys;logcat
```


6.2 查看内存池使用情况

【解决】

```
cat /tmp/mempool0
```

6.3 查看 mem 使用情况

【解决】

```
echo 3 >/proc/sys/vm/drop_caches //释放缓存
cat /proc/meminfo
```

6.4 nmem 使用查看

【解决】

用 libdriver.m.so 替换 libdriver.so，会在/tmp 下生成 nmem.txt；里面有详细的 nmem 的使用情况。

6.5 内存节省方式-直通功能

- 1) 直通功能只能在主码流上使用；
- 2) 直通功能相比传统方式可以节省 1~1.75 个 buffer；
- 3) 直通功能下，主码流只能使用 isp osd，其他通道使用 IPU OSD；
- 4) 测试可以使用 sample-Encoder-video-direct.c ；
- 5) jpeg 使用 IVPU，参考 sample-Encoder-video-direct.c 。

6.5.1 直通功能使用方式：

一、加载驱动配置

A. direct_mode 参数

```
$ insmod tx-isp-t41.ko direct_mode=1
```

0: 代表使用非直通，使用传统方式，先 copy 到 buffer 再给编码；

1: 代表使用直通，占用内存少；

2: 代表使用半直通，占用内存相比直通多一点，4K 分辨率建议使用半直通。

B. ivdc_mem_line 参数，配置 IVDC 中 DDR 存储的行数，默认为 height/4

```
$ insmod tx-isp-t41.ko direct_mode=1 ivdc_mem_line=
height / 2
```

C. ivdc_threshold_line 参数，配置 IVDC 编码缓存的阈值（单位：行数 width，

256 对齐)； 0: 配置为 height/2, 表示阈值为半帧 (默认值), 大于 0: 表示缓存为 width * ivdc_threshold_line

```
$ insmod tx-isp-t41.ko direct_mode=1 ivdc_threshold_line=
height / 2
```

二、编码初始化配置

结构体为: IMPEncoderChnAttr channel_attr;

结构体下参数: channel_attr.bEnableIvdc

使用时机: 创建编码通道时

```
if (0 == chnNum)
    channel_attr.bEnableIvdc = true;
ret = IMP_Encoder_CreateChn(chnNum, &channel_attr);
```

6.6 其他节省内存方式

- 1) 减少 nrvbs 个数;
- 2) 减少编码 sreamcnt 个数;
- 3) 加载 isp 驱动时候加上参数 isp_memopt=1;
- 4) 使用内存池, 根据内存池实际使用情况减少 rmem 配置, 需要注意 cmdline 中内存大小配置需要 4K 对齐;
- 5) IPU OSD 使用 IPU_RMEM_PIC 方式;
- 6) 裁剪内核大小, 去掉不必要功能;
- 7) 减少音频 buffer 缓存帧数;
- 8) 算法可以使用 nocopy 方式直接使用 nrvbs 的数据。

7 调试工具

7.1 Coredump

7.1.1 coredump 介绍

当程序运行的过程中异常终止或者崩溃，操作系统会将程序当时的内存状态记录下来，保存在一个文件中，这种行为就叫做 core dump。可以认为 core dump 是“内存快照”，但实际上，除了内存信息之外，还有些关键的程序运行状态也会同时 dump 下来，例如寄存器信息（包括程序指针、栈指针等）、内存管理信息、其他处理器和操作系统状态和信息。core dump 对于编程人员诊断和调试程序是非常有帮助的，因为对于有些程序错误是很难重现的，例如指针异常，而 core dump 文件可以再现程序出错时的情景。

7.1.2 开启 coredump

可以使用命令 ulimit 开启，也可以在程序中通过 setrlimit 系统调用开启。

在终端输入命令 `$ ulimit -c` 输出结果为 0，说明默认是关闭 coredump 的，即当程序异常终止时，不会生成 coredump 文件。可以使用 `$ ulimit -c unlimited` 来开启 coredump 功能，并且不限制 coredump 文件大小；如果需要限制文件大小，将 unlimited 改成你想生成 core 文件的大小，注意单位是 KB。

7.1.3 调试 core 文件

调试使用 PC 上的 gdb 查看程序挂载位置。但调试开发板程序需要使用交叉编译工具链中的 gdb 命令：

```
glibc:mips-linux-gnu-gdb
uclibc: mips-linux-uclibc-gnu-gdb
```

使用 `mips-linux-***-gdb [program] [core]` 查看 core 文件，其中 program 是可执行程序文件名，core 是生成的 core 文件名。例如：

```
$ mips-linux-uclibc-gnu-gdb ./build core-build
```

7.2 gdbserver

7.2.1 gdbserver 使用方法

`gdbserver` 是嵌入式开发调试的主要工具，依赖开发板上的 `gdbserver` 程序以及交叉编译工具链中的 `mips-linux-gnu-gdb` 命令。`gdbserver` 需要调试的 PC 和开发板网络相通。

本文仅介绍 `gdbserver` 的开启及连接方法，相关命令遵循标准 `gdbserver` 命令，详细内容请在互联网查询。

`gdb` 以及 `gdbserver` 命令位置：

- A. `gdb` 在 `toolchain` 文件夹下，位于 `bin/mips-linux-gnu-gdb`
- B. `gdbserver` 位于 `toolchain` 文件夹下：
 - `glibc: mips-linux-gnu/libc/usr/lib/bin/gdbserver`
 - `uclibc: mips-linux-gnu/libc/uclibc/usr/lib/bin/gdbserver`

1) PC 运行 `gdb`

```
$ mips-linux-gnu-gdb [PC 端应用程序路径]
```

例如：

```
$ mips-linux-gnu-gdb sample-Encoder-h264
GNU gdb (Ingenic 2015.02) 7.4.50.20120716-cvs
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
<http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show
copying"and "show warranty"for details.
This GDB was configured as "--host=i686-pc-linux-gnu
--target=mips-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from sample-Encoder-h264...done.
(gdb)
```

A. 如果出现`(no debugging symbols found)`的提示，可能是因为应用程序 strip 过了（PC 端），删除了 debug 需要的段，无法进行 gdb 调试

B. 开发板上程序是可以 strip 的，Symbol 的读取在 PC 端 gdb 工具完成。这也就是 gdbserver 相对与 local gdb 以及 core dump 的优势——开发板端可以 strip——因为前两者都需要在开发板端 load symbol，因此应用程序会变得非常大，无法放在 flash 中。

2) 设置动态库搜索路径

应用程序往往动态链接 libc、ld 等库，当运行于动态库中 break 时（比如 memcpy）往往无法跟踪到当前 pc 所在位置。这是因为动态库的加载地址是不固定的。因此需要在 PC 端 gdb 上指定动态库加载路径。方法是：

```
set solib-search-path
/opt/mips-gcc472-glibc216-64bit/mips-linux-gnu/libc/lib/:/opt/mips-gc
c472-glibc216-64bit/mips-linux-gnu/lib
```

多个路径之间通过“:”来隔开。注：上述例子中仅指定了 toolchain 中 glibc 的 libc 系列库和 libstdc++ 等主要外部库。

A. glibc 的动态库在 toolchain 文件夹下的相对目录为：mips-linux-gnu/libc/lib 与 mips-linux-gnu/lib

B. uclibc 的动态库在 toolchain 文件夹下的相对目录为：
mips-linux-gnu/libc/uclibc/lib 与 mips-linux-gnu/lib/uclibc

3) 开发板启动 gdbserver

```
$ gdbserver [PC 机 IP:端口, 与步骤 2 中端口需一致] [应用程序路径]
例如:
```

```
$ gdbserver 192.168.1.100:1234 /tmp/sample-Encoder-h264
```

4) 链接开发板

```
$ target remote [开发板 IP:端口]
```

例如：

```
$ target remote 192.168.1.101:1234
```

5) gdbserver 连通，开发板会出现如下信息：

```
(gdb) target remote 192.168.1.101:1234
Remote debugging using 192.168.1.101:1234
Reading symbols from
/opt/mips-gcc472-glibc216-64bit/mips-linux-gnu/libc/lib/ld.so.1...don
e.
Loaded symbols for
```

```
/opt/mips-gcc472-glibc216-64bit/mips-linux-gnu/libc/lib/ld.so.1
0x77fc7070 in __start () from
/opt/mips-gcc472-glibc216-64bit/mips-linux-gnu/libc/lib/ld.so.1
```

至此，即可在 PC 端运行 gdb 命令。



注意

与 gdbserver 与 local gdb 不同，启动程序的命令为 “c” (continue)。

7.3 IMPSDK 调试命令

1) Rmem

rmem 的内存可以使用 SDK 中的 rmem 计算器 excel 表格来计算得到；SDK remem 内存使用信息可以使用下面命令查看：

```
impdbg --sys;logcat
```

2) 查看数据流处理统计

update_cnt 是每个 group 模块线程的处理次数统计 可以通过下面命令来获取 10 秒间隔的统计来计算处理帧率。

```
[root@IPC:tmp]# ./impdbg --sys;sleep 10;./impdbg --sys
tree item: 3
Framesource-0      update_cnt=81540(qframecnt=81538,
dqframecnt=81540, sem_msg_cnt=16, sem_cnt=0)
OSD-0              update_cnt=81540(sem_msg_cnt=16, sem_cnt=0)
Encoder-0          update_cnt=81540(sem_msg_cnt=16, sem_cnt=0)
---Framesource-0---OSD-0-----Encoder-0
tree item: 4
Framesource-1      update_cnt=108828(qframecnt=108823,
dqframecnt=108823, sem_msg_cnt=16, sem_cnt=0)
IVS-0              update_cnt=108828(sem_msg_cnt=16, sem_cnt=0)
OSD-1              update_cnt=108828(sem_msg_cnt=16, sem_cnt=0)
Encoder-1          update_cnt=108828(sem_msg_cnt=16, sem_cnt=0)
---Framesource-1---IVS-0-----OSD-1-----Encoder-1
```

3) 查看编码信息统计

可以通过下面命令获取编码 group, channel 信息，编码类型，码流控制等参数。

-tf 信息，编码通道码流 buf 成功被上次取走的码流统计。

-df 信息，编码通道码流 buf 慢导致的丢帧数统计。

```
[root@IPC:tmp]# ./impdbg --enc_info;sleep 10;./impdbg --enc_info
GROUP 0
          CHANNEL 0      1920x 1080      START H265 tf:69198      df:6050
encdur:3343131
-----
ch->index = 0
rcAttr->outFrmRate->frmRateNum = 25(0x19) offset:size = 0:4
rcAttr->outFrmRate->frmRateDen = 1(0x1) offset:size = 4:4
rcAttr->maxGop = 50(0x32) offset:size = 8:4
rcAttr->rcMode->rcMode = 3(0x3) offset:size = 12:4
rcAttr->rcMode->Smart->maxQp = 45(0x2d) offset:size = 16:4
rcAttr->rcMode->Smart->minQp = 15(0xf) offset:size = 20:4
rcAttr->rcMode->Smart->staticTime = 2(0x2) offset:size = 24:4
rcAttr->rcMode->Smart->maxBitRate = 2048(0x800) offset:size = 28:4
rcAttr->rcMode->Smart->iBiasLvl = -3(0xfffffff) offset:size = 32:4
rcAttr->rcMode->Smart->changePos = 80(0x50) offset:size = 36:4
rcAttr->rcMode->Smart->qualityLvl = 2(0x2) offset:size = 40:4
rcAttr->rcMode->Smart->frmQPStep = 3(0x3) offset:size = 44:4
rcAttr->rcMode->Smart->gopQPStep = 15(0xf) offset:size = 48:4
rcAttr->rcMode->Smart->flucLvl = 2(0x2) offset:size = 52:4
rcAttr->frmUsed->enable = 0(0x0) offset:size = 56:1
rcAttr->frmUsed->frmUsedMode = 0(0x0) offset:size = 60:4
rcAttr->frmUsed->frmUsedTimes = 0(0x0) offset:size = 64:4
rcAttr->denoise->enable = 0(0x0) offset:size = 68:1
rcAttr->denoise->dnType = 0(0x0) offset:size = 72:4
rcAttr->denoise->dnIQp = 0(0x0) offset:size = 76:4
rcAttr->denoise->dnPQp = 0(0x0) offset:size = 80:4
rcAttr->hSkip->hSkipAttr->skipType = 0(0x0) offset:size = 84:4
rcAttr->hSkip->hSkipAttr->m = 49(0x31) offset:size = 88:4
rcAttr->hSkip->hSkipAttr->n = 1(0x1) offset:size = 92:4
rcAttr->hSkip->hSkipAttr->maxSameSceneCnt = 1(0x1) offset:size =
96:4
rcAttr->hSkip->hSkipAttr->bEnableScenecut = 0(0x0) offset:size =
100:4
rcAttr->hSkip->hSkipAttr->bBlackEnhance = 0(0x0) offset:size = 104:4
rcAttr->hSkip->maxHSkipType = 1(0x1) offset:size = 108:4
GROUP 1
          CHANNEL 1      640x 480      START H265 tf:83485      df:1506
encdur:3343132
-----
ch->index = 1
rcAttr->outFrmRate->frmRateNum = 25(0x19) offset:size = 0:4
```

```

rcAttr->outFrmRate->frmRateDen = 1(0x1) offset:size = 4:4
rcAttr->maxGop = 50(0x32) offset:size = 8:4
rcAttr->rcMode->rcMode = 3(0x3) offset:size = 12:4
rcAttr->rcMode->Smart->maxQp = 45(0x2d) offset:size = 16:4
rcAttr->rcMode->Smart->minQp = 15(0xf) offset:size = 20:4
rcAttr->rcMode->Smart->staticTime = 2(0x2) offset:size = 24:4
rcAttr->rcMode->Smart->maxBitRate = 546(0x222) offset:size = 28:4
rcAttr->rcMode->Smart->iBiasLvl = -3(0xfffffff) offset:size = 32:4
rcAttr->rcMode->Smart->changePos = 80(0x50) offset:size = 36:4
rcAttr->rcMode->Smart->qualityLvl = 2(0x2) offset:size = 40:4
rcAttr->rcMode->Smart->frmQPStep = 3(0x3) offset:size = 44:4
rcAttr->rcMode->Smart->gopQPStep = 15(0xf) offset:size = 48:4
rcAttr->rcMode->Smart->flucLvl = 2(0x2) offset:size = 52:4
rcAttr->frmUsed->enable = 0(0x0) offset:size = 56:1
rcAttr->frmUsed->frmUsedMode = 0(0x0) offset:size = 60:4
rcAttr->frmUsed->frmUsedTimes = 0(0x0) offset:size = 64:4
rcAttr->denoise->enable = 0(0x0) offset:size = 68:1
rcAttr->denoise->dnType = 0(0x0) offset:size = 72:4
rcAttr->denoise->dnIQp = 0(0x0) offset:size = 76:4
rcAttr->denoise->dnPQp = 0(0x0) offset:size = 80:4
rcAttr->hSkip->hSkipAttr->skipType = 0(0x0) offset:size = 84:4
rcAttr->hSkip->hSkipAttr->m = 49(0x31) offset:size = 88:4
rcAttr->hSkip->hSkipAttr->n = 1(0x1) offset:size = 92:4
rcAttr->hSkip->hSkipAttr->maxSameSceneCnt = 1(0x1) offset:size =
96:4
rcAttr->hSkip->hSkipAttr->bEnableScenecut = 0(0x0) offset:size =
100:4
rcAttr->hSkip->hSkipAttr->bBlackEnhance = 0(0x0) offset:size = 104:4
rcAttr->hSkip->maxHSkipType = 1(0x1) offset:size = 108:4

GROUP 0
CHANNEL 0 1920x 1080 START H265 tf:69319 df:6050
encdur:3353252
-----
ch->index = 0
rcAttr->outFrmRate->frmRateNum = 25(0x19) offset:size = 0:4
rcAttr->outFrmRate->frmRateDen = 1(0x1) offset:size = 4:4
rcAttr->maxGop = 50(0x32) offset:size = 8:4
rcAttr->rcMode->rcMode = 3(0x3) offset:size = 12:4
rcAttr->rcMode->Smart->maxQp = 45(0x2d) offset:size = 16:4
rcAttr->rcMode->Smart->minQp = 15(0xf) offset:size = 20:4
rcAttr->rcMode->Smart->staticTime = 2(0x2) offset:size = 24:4
rcAttr->rcMode->Smart->maxBitRate = 2048(0x800) offset:size = 28:4

```



```

rcAttr->rcMode->Smart->iBiasLvl = -3(0xffffffff) offset:size = 32:4
rcAttr->rcMode->Smart->changePos = 80(0x50) offset:size = 36:4
rcAttr->rcMode->Smart->qualityLvl = 2(0x2) offset:size = 40:4
rcAttr->rcMode->Smart->frmQPStep = 3(0x3) offset:size = 44:4
rcAttr->rcMode->Smart->gopQPStep = 15(0xf) offset:size = 48:4
rcAttr->rcMode->Smart->flucLvl = 2(0x2) offset:size = 52:4
rcAttr->frmUsed->enable = 0(0x0) offset:size = 56:1
rcAttr->frmUsed->frmUsedMode = 0(0x0) offset:size = 60:4
rcAttr->frmUsed->frmUsedTimes = 0(0x0) offset:size = 64:4
rcAttr->denoise->enable = 0(0x0) offset:size = 68:1
rcAttr->denoise->dnType = 0(0x0) offset:size = 72:4
rcAttr->denoise->dnIQp = 0(0x0) offset:size = 76:4
rcAttr->denoise->dnPQp = 0(0x0) offset:size = 80:4
rcAttr->hSkip->hSkipAttr->skipType = 0(0x0) offset:size = 84:4
rcAttr->hSkip->hSkipAttr->m = 49(0x31) offset:size = 88:4
rcAttr->hSkip->hSkipAttr->n = 1(0x1) offset:size = 92:4
rcAttr->hSkip->hSkipAttr->maxSameSceneCnt = 1(0x1) offset:size =
96:4
rcAttr->hSkip->hSkipAttr->bEnableScenecut = 0(0x0) offset:size =
100:4
rcAttr->hSkip->hSkipAttr->bBlackEnhance = 0(0x0) offset:size = 104:4
rcAttr->hSkip->maxHSkipType = 1(0x1) offset:size = 108:4
GROUP 1
        CHANNEL 1      640x  480      START H265 tf:83738      df:1506
encdur:3353253
-----
ch->index = 1
rcAttr->outFrmRate->frmRateNum = 25(0x19) offset:size = 0:4
rcAttr->outFrmRate->frmRateDen = 1(0x1) offset:size = 4:4
rcAttr->maxGop = 50(0x32) offset:size = 8:4
rcAttr->rcMode->rcMode = 3(0x3) offset:size = 12:4
rcAttr->rcMode->Smart->maxQp = 45(0x2d) offset:size = 16:4
rcAttr->rcMode->Smart->minQp = 15(0xf) offset:size = 20:4
rcAttr->rcMode->Smart->staticTime = 2(0x2) offset:size = 24:4
rcAttr->rcMode->Smart->maxBitRate = 546(0x222) offset:size = 28:4
rcAttr->rcMode->Smart->iBiasLvl = -3(0xffffffff) offset:size = 32:4
rcAttr->rcMode->Smart->changePos = 80(0x50) offset:size = 36:4
rcAttr->rcMode->Smart->qualityLvl = 2(0x2) offset:size = 40:4
rcAttr->rcMode->Smart->frmQPStep = 3(0x3) offset:size = 44:4
rcAttr->rcMode->Smart->gopQPStep = 15(0xf) offset:size = 48:4
rcAttr->rcMode->Smart->flucLvl = 2(0x2) offset:size = 52:4
rcAttr->frmUsed->enable = 0(0x0) offset:size = 56:1
rcAttr->frmUsed->frmUsedMode = 0(0x0) offset:size = 60:4

```

```

rcAttr->frmUsed->frmUsedTimes = 0(0x0) offset:size = 64:4
rcAttr->denoise->enable = 0(0x0) offset:size = 68:1
rcAttr->denoise->dnType = 0(0x0) offset:size = 72:4
rcAttr->denoise->dnIQp = 0(0x0) offset:size = 76:4
rcAttr->denoise->dnPQp = 0(0x0) offset:size = 80:4
rcAttr->hSkip->hSkipAttr->skipType = 0(0x0) offset:size = 84:4
rcAttr->hSkip->hSkipAttr->m = 49(0x31) offset:size = 88:4
rcAttr->hSkip->hSkipAttr->n = 1(0x1) offset:size = 92:4
rcAttr->hSkip->hSkipAttr->maxSameSceneCnt = 1(0x1) offset:size =
96:4
rcAttr->hSkip->hSkipAttr->bEnableScenecut = 0(0x0) offset:size =
100:4
rcAttr->hSkip->hSkipAttr->bBlackEnhance = 0(0x0) offset:size = 104:4
rcAttr->hSkip->maxHSkipType = 1(0x1) offset:size = 108:4

```

impdbg 其它命令说明:

usage: impdbg --option [args]

--enc_info	get encoder info
--enc_rc_s chn:offset:size:data	set encoder rc
--fs_info	get frame source info
--save_pic [path]	save picture
--system_info	get system info
--ui_mode	access ui mode
--miscmd	misc cmd
--ai_dev_info	get ai parameter info
--ai_get_frm	get ai frame info
--ao_dev_info	get ao parameter info
--ao_get_frm	get ao frame info