ISP50 Report

Agenda

1. Introduction ISP
   - 1.1 isp简介.
   - 1.2 seninf部分reg 说明(结合isp40 report录像 即可, 不再复述)
   - 1.3 分析pipecheck的code flow, 作用说明.
   - 1.4 常规问题解题SOP
2. 常用Debug CMD
3. Case Share(按照常见问题数量)
   - Sensor 吐数据不够
   - Sensor fifo overrun
   - Sensor ecc crc 异常
   - ISP fifo overrun
   - MW flow / ap 问题
   - 其他isp部分已经修正bug
   - 其他模块问题
4. 常见log 解释

1.1 isp简介.
1.2 seninf部分reg 说明(结合isp40 report录像 即可, 不再复述)
1.3 分析pipecheck的code flow, 作用说明.
1.4 常规问题解题SOP

INTRODUCTION ISP

1.1 ISP 简介

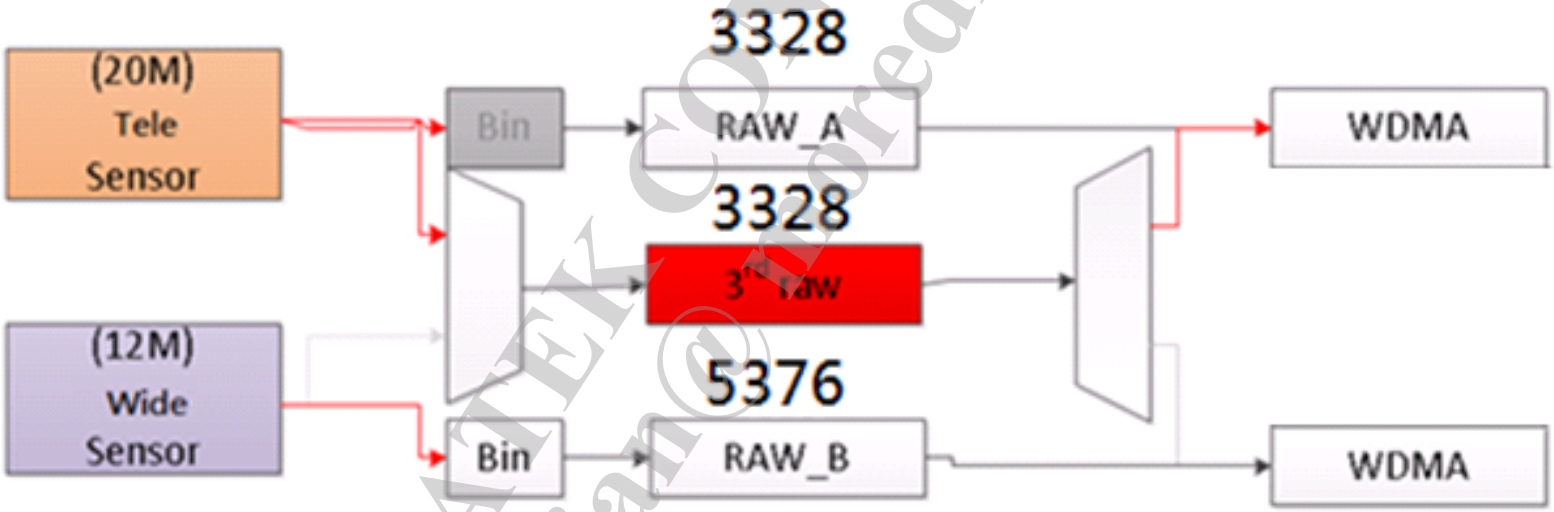
Hardware Spec

	Sylvia/P40/mt6771 (ISP 5.0)	Bianco(ISP 4.5)
Clock	546MHz/364MHz	450MHz/300MHz
Resolution	24M 30fps 20M+16M 30fps	21M 30fps 13M+13M 30fps
TG#	2	2
Cam#	3	2
UNI dmao	N/A	flko/eiso/rsso
Cam dmao	imgo/rrzo/lcso/ufeo/ufgo aao/pso/afo/pdo/ **flko/eiso/rsso**	imgo/rrzo/lcso aao/pso/afo/pdo/

Pass 1 → Cam
Pass 2 → DIP

Hardware Spec

- #define CAM_A_MAX_LINE_BUFFER_IN_PIXEL (3328)
- #define CAM_B_MAX_LINE_BUFFER_IN_PIXEL (5376)
- #define CAM_C_MAX_LINE_BUFFER_IN_PIXEL (3328)



Cam HW Path - Interrupt

```
/* 0x1A004024 */
30   SW_PASS1_DON_ST SW PASS1 DONE interrupt enable, it can be down sample and drop when
rrzo/imgo/lcso/lmvo/rsso image fifo full
29   DMA_ERR_ST   DMA error status
28   PDO_ERR_ST   PDO error status
27   UFEO_ERR_ST UFEO error status
26   UFGO_ERR_ST UFGO error status
25   LSC_ERR_ST   LSC error status, lsc tabe can't read on time
24   BNR_ERR_ST   BNR error status, bpc tabe can't read on time
23   LCSO_ERR_ST LCSO error status
22   PSO_ERR_ST   PSO error status
21   AAO_ERR_ST   AEerror status
20   IMGO_ERR_ST IMGO overrun status
19   AFO_ERR_ST   AFO error interrrupt status
18  RRZO_ERR_ST RRZO error status
12   SOF_INT_ST TG1 sof interrupt status
11   PASS1_DON_ST     PASS1  done  interrupt status
10   RRZO_DROP_FRAME_ST   RRZO drop frame status
9    IMGO_DROP_FRAME_ST   IMGO drop frame status
8    CQ_VS_ERR_ST     CQ cover next vsync error
7    CQ_APB_ERR_ST    CQ APB out of range
6    CQ_CODE_ERR_ST   CQ Code out of range
5    TG_GBERR_ST TG_GBERR
4    TG_ERR_ST    "TG1 ERR status 1 mease TG overrun occur, it normal hint bandwidth is not
enough"
3    EXPDON_ST    Exposusre1 done interrupt status
2    TG_INT2_ST   TG interrupt2 status
1    TG_INT1_ST   TG interrupt1 status
0    VS_INT_ST    Vsync1 interrupt status
```

HW pass1 done. Just like SOF, HW p1 done will also signal even if current frame is FBC's drop frame. This signal is for CQ's trig signal.

sensor output data < TG grab size

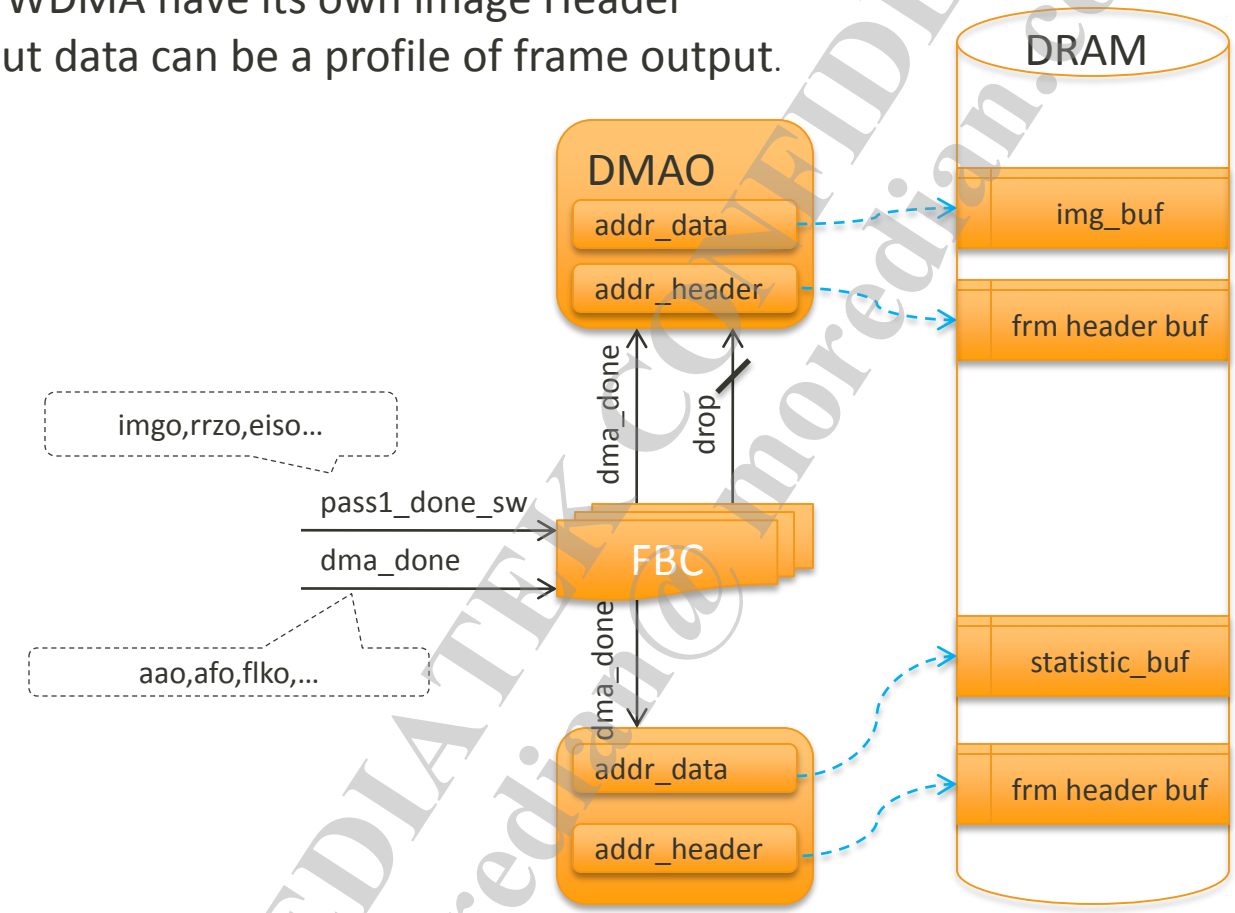
1. data through put > isp clk.
2. isp setting error.
3. SMI 回擋.

Note:
0x1a004024:pre-frame read
0x1a004028: read when error occurred.

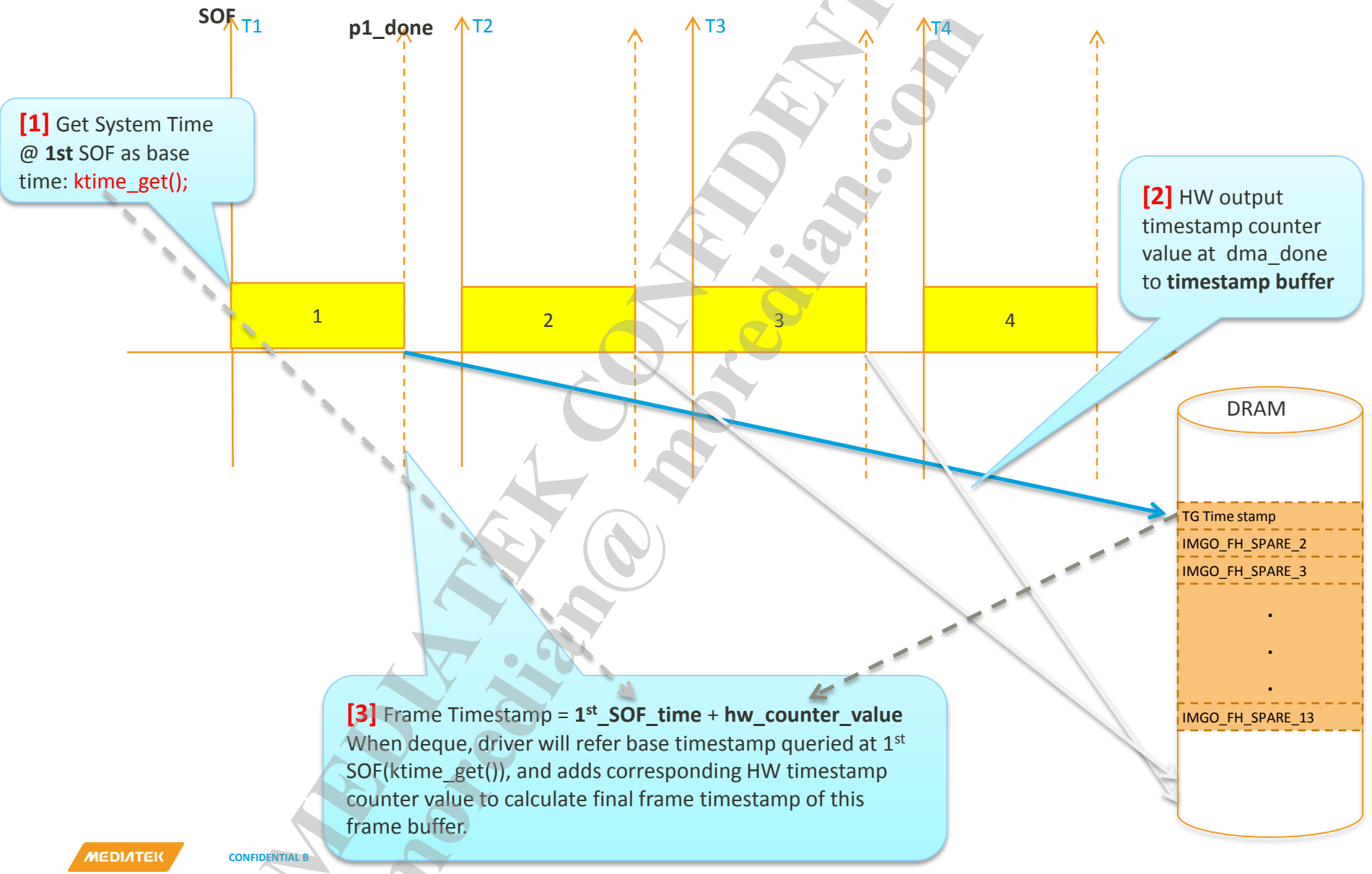
Cam HW Path - DMA

architecture of Image Header:

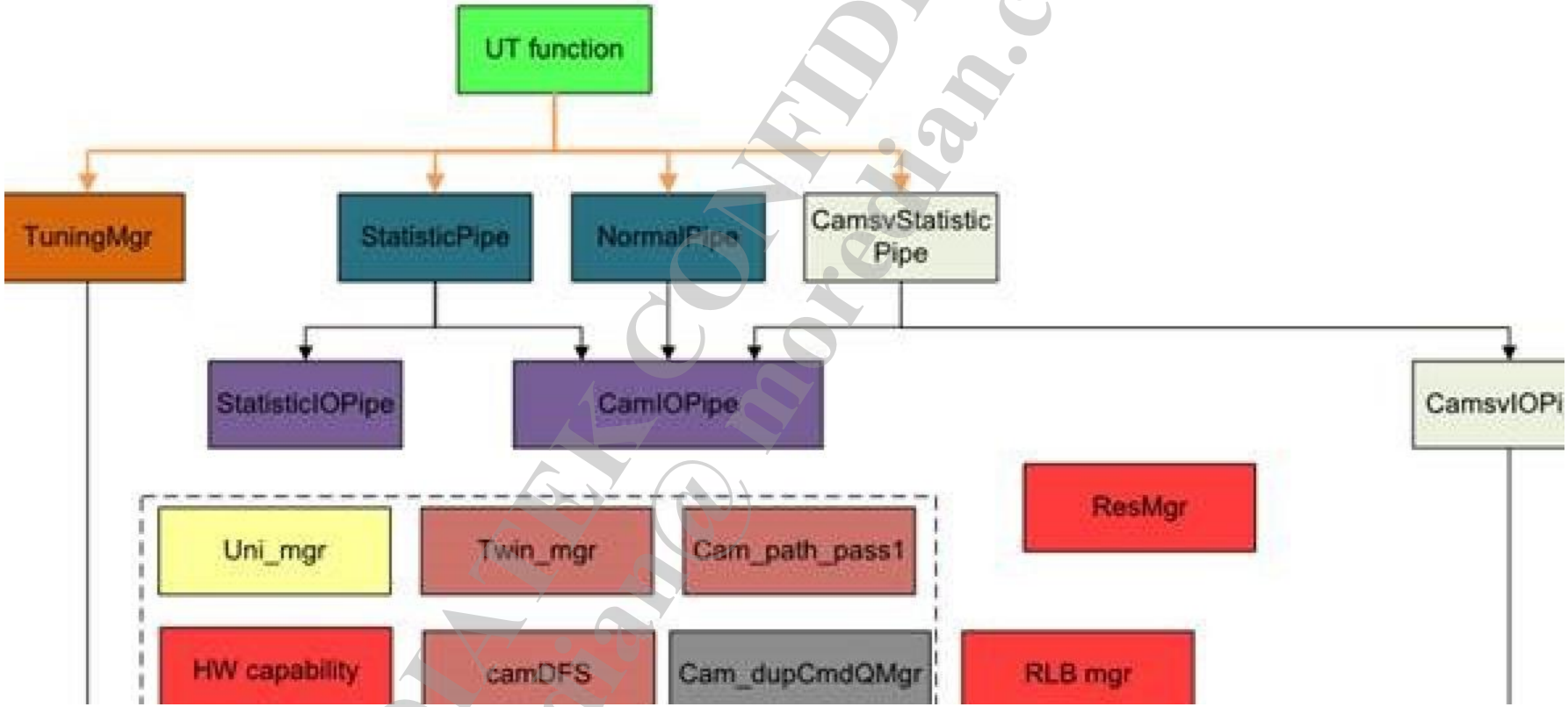
- Each WDMA have its own image Header
- Output data can be a profile of frame output.



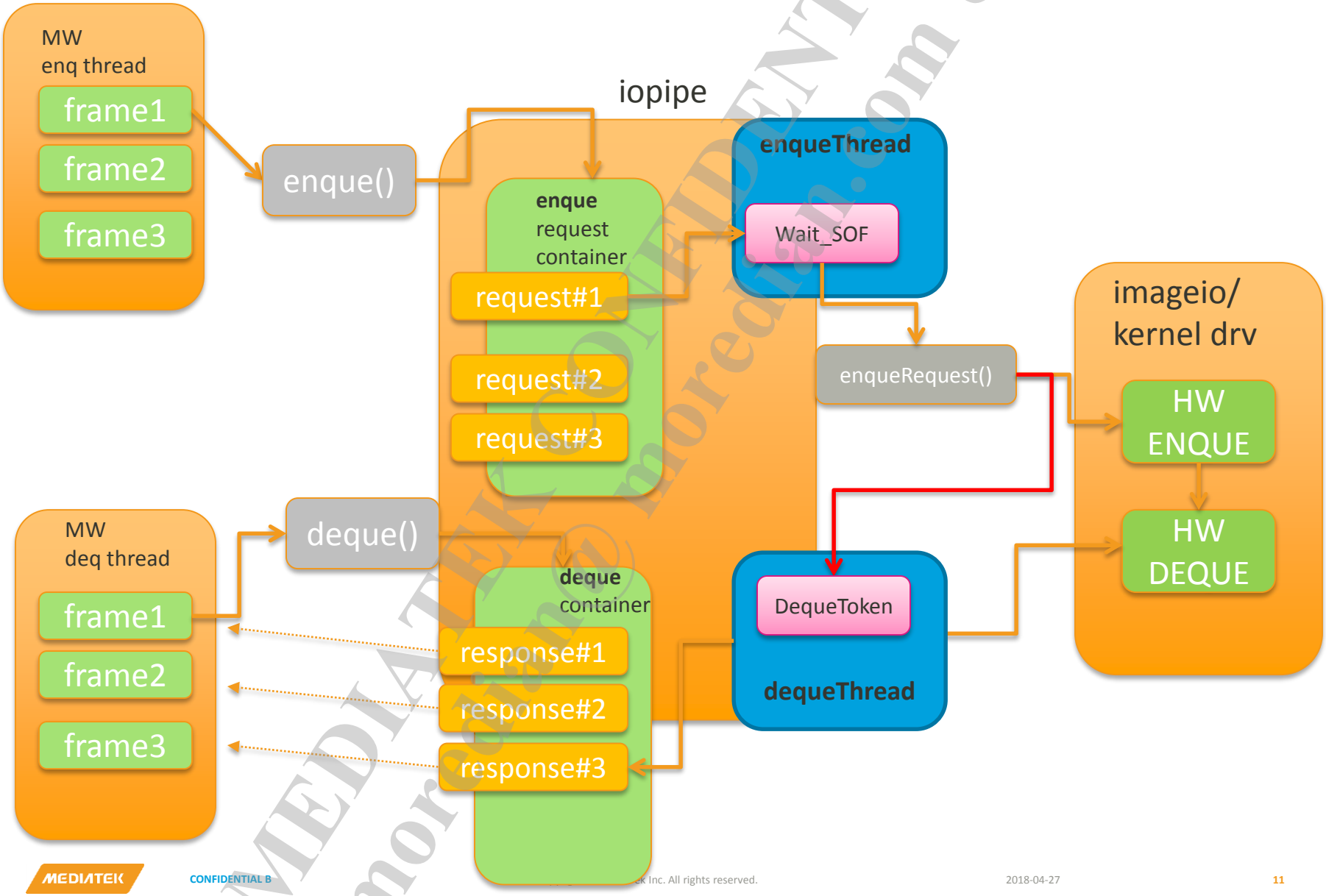
Cam HW Path – HW Timestamp



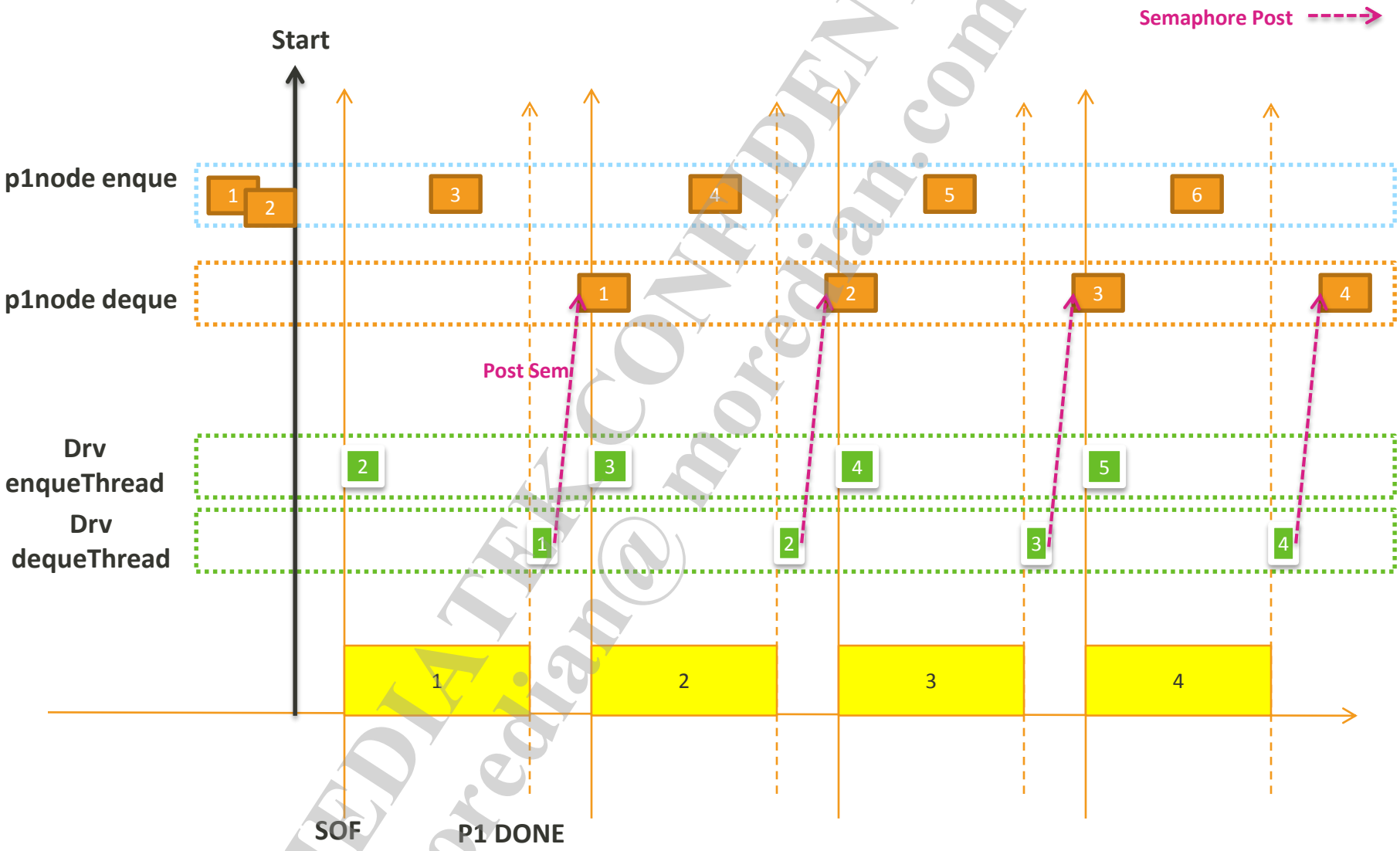
Driver Architecture - Driver Stack



Driver working/sync timing

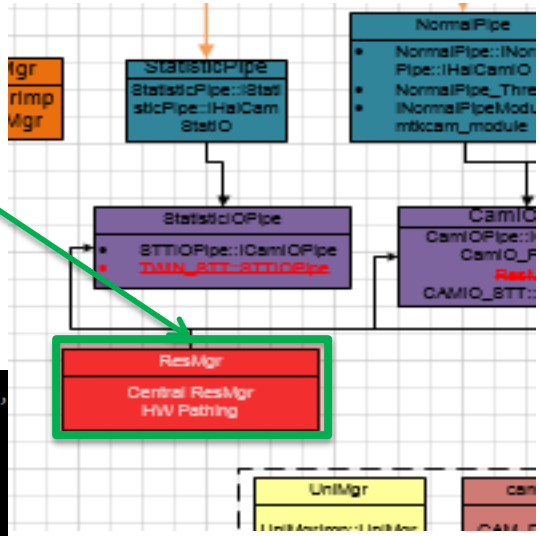


Driver working/sync timing



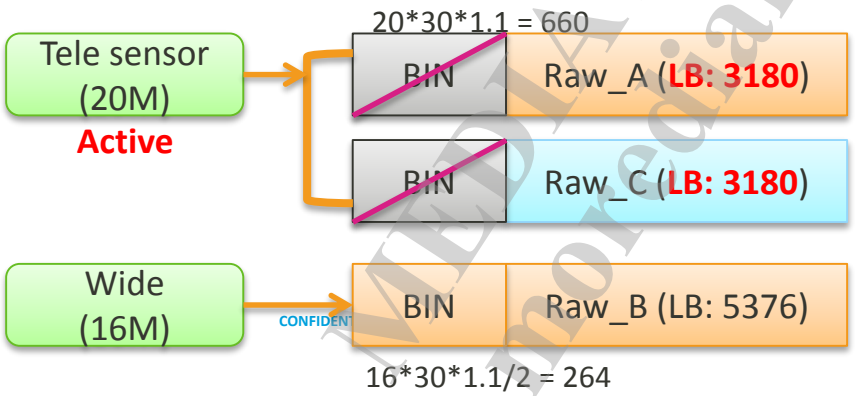
Twin resource allocate

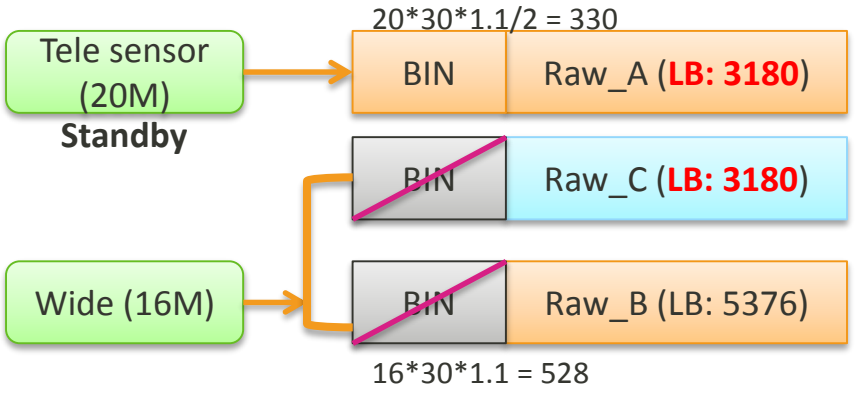
- Central control logic in **ResMgr**:
  - default: 1st TWIN, 2nd BIN, 3rd ISP_CLK
  - off-bin: 1st TWIN, 2nd ISP_CLK




```
MBOOL Cam_ResMgr::Res_Meter_kernel(const Res_Meter_IN& cfgInParam, Res_Meter_OUT &cfgOutParam,MUINT32& thru_put,
{
    //check thru put, always counting from clk_level = 0 for isp minimum request
    //start calc. thru put
    //priority : 1:twin.  2:bin.  3:Clk.
    while(thru_put >= (MUINT32)((MUINT64)cfgInParam.vClk.at(cfgOutParam.clk_level) * ClkMargin))
    {
        if((cfgInParam.offTwin == MFALSE) && (availCam.size() != 0)){
            this->K_TWIN_CASE(cfgOutParam,thru_put,availCam);
        }
        else if(bBin_useful == MTRUE){    //2nd
            this->K_BIN_CASE(cfgOutParam,thru_put,bBin_useful,bin_pix_mode);
        }
        else if(cfgOutParam.clk_level < (cfgInParam.vClk.size()-1)){    //3rd
            this->K_CLK_CASE(cfgOutParam);
```

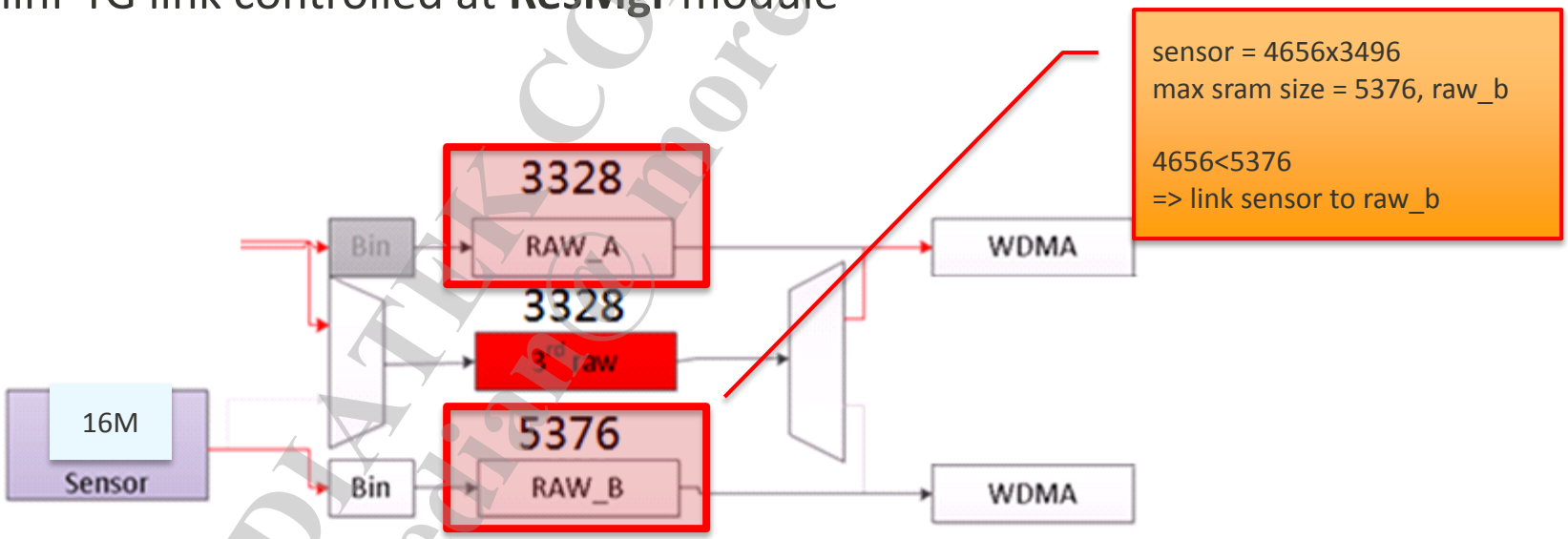
LPM : 364MHz
HPM: 546MHz





Seninf-TG link rule

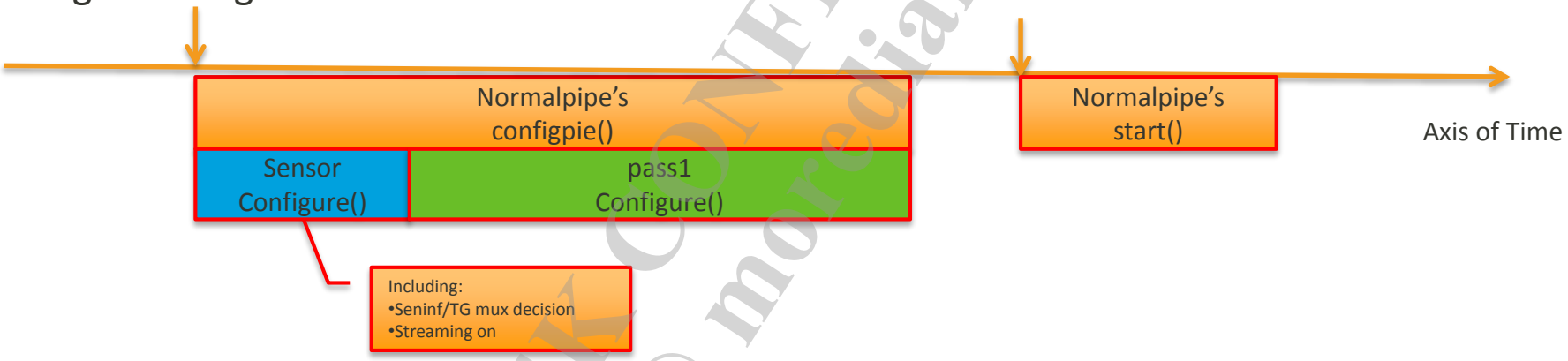
- Seninf link to which TG will be decided through pass1 driver
- Link logic:
  - If SENSOR_H < maximum sram size(5376)(single path)
    => open maximum sram size for this sensor
- Seninf-TG link controlled at **ResMgr** module



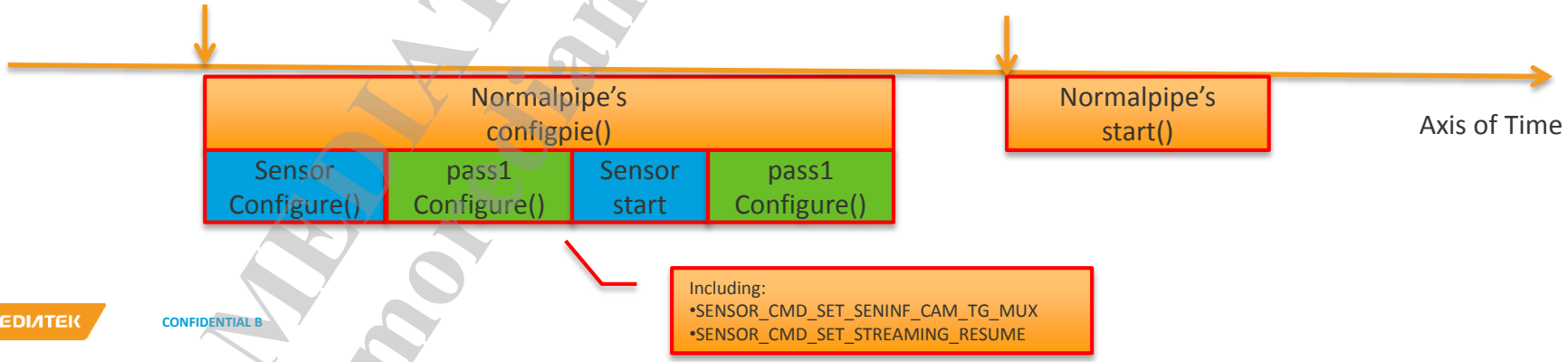
Seninf-TG link rule

- Decision timing

Original design:

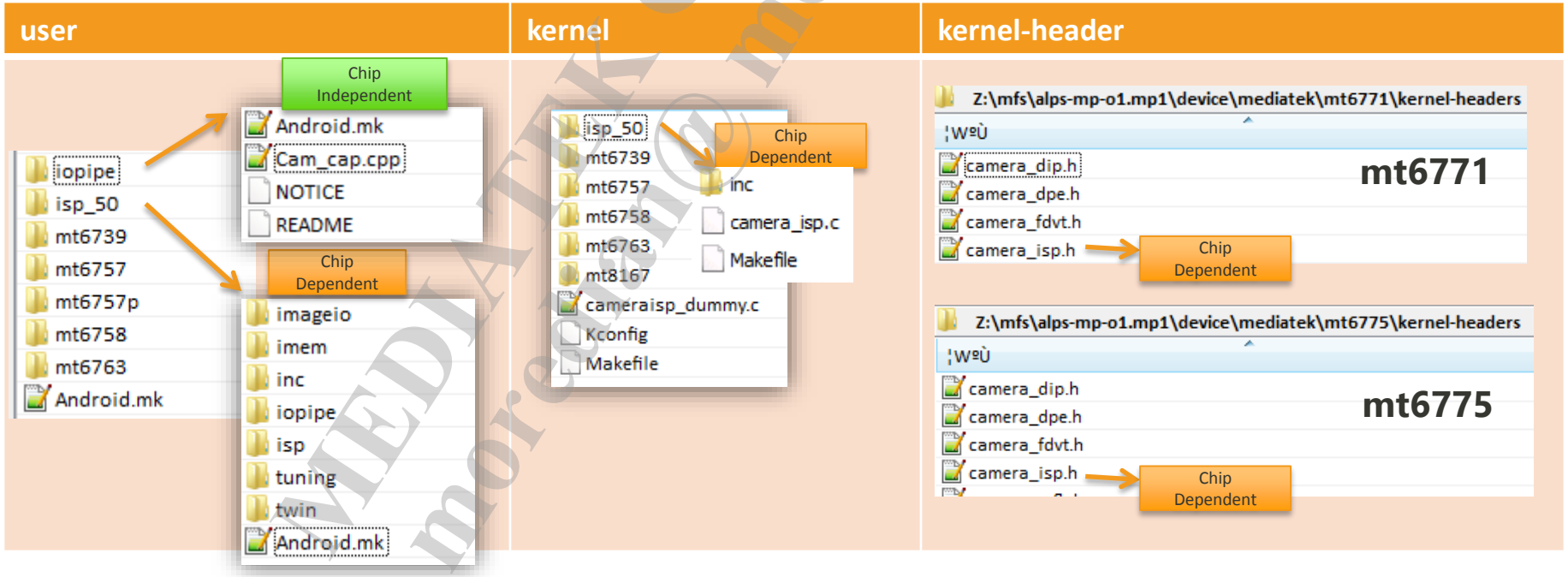


ISP50 design:



Pass1 File Tree

- isp_50 folder for sylvia & cannon user/kernel code, except kernel-header
  - sylvia: mt6771, p40
  - cannon: mt6775, p70
- Users pace path
  - include: vendor\mediatek\proprietary\hardware\mtkcam\drv\include\isp_50
  - src: vendor\mediatek\proprietary\hardware\mtkcam\drv\src\isp\isp_50
- kernel space path
  - kernel-4.4\drivers\misc\mediatek\cameraisp\src
- kernel-header
  - sylvia: device\mediatek\mt6771\kernel-headers
  - cannon: device\mediatek\mt6775\kernel-headers



ISP COMMON KNOWLEDGE<1>

- VSYNC, SOF, P1DONE,

Vsync:sensor 打出第一条line的信号后, 触发seninf 产生vsync 信号.

sof : vsync 信号通过VF到tg 后,tg会模拟出sof 信号.

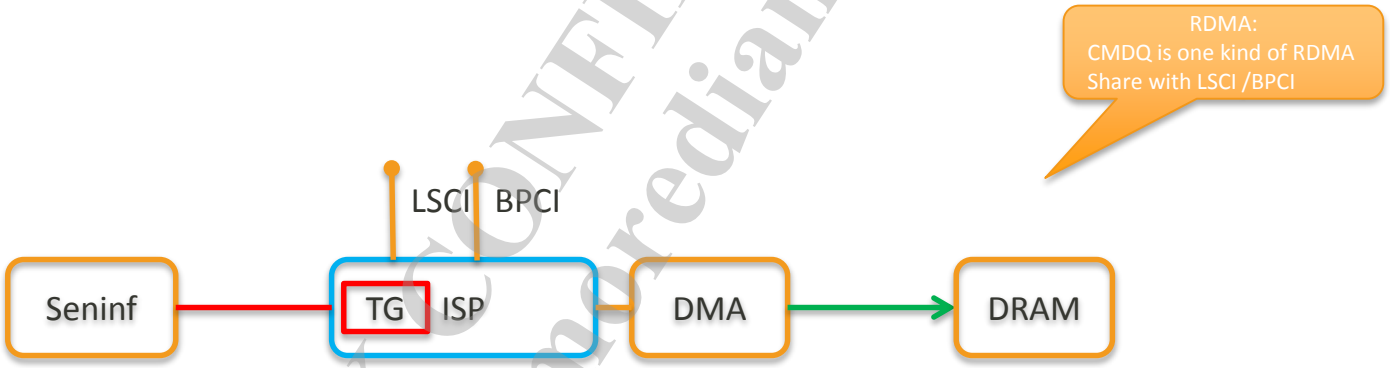
P1DONE: tg收满一个frame,通过特定的DMA out (imgo rrzo etc…)发到对应的DRAM后,若DRAM接受完毕.说明这个frame成功收下.

- LOST P1 DONE.

特定DMA out的数据没有完全发到DRAM中,引起lost P1 done.
原因:

1. 没有通过enque申请DRAM , 导致dma out无处可吐.
2. Sensor 没有吐数据,导致没有数据送出去.
3. Dma out发到DRAM速度太慢, sensor的下一个P1 done已经来了.
4. 其他原因. isp设定, seninf设定,etc…

ISP COMMON KNOWLEDGE<2>



ISP COMMON KNOWLEDGE<3>

- twinmode,

就是说一个sensor 同时开了两个tg(isp)一起处理数据.
这个时候另外一个sensor 就不能再打开了.

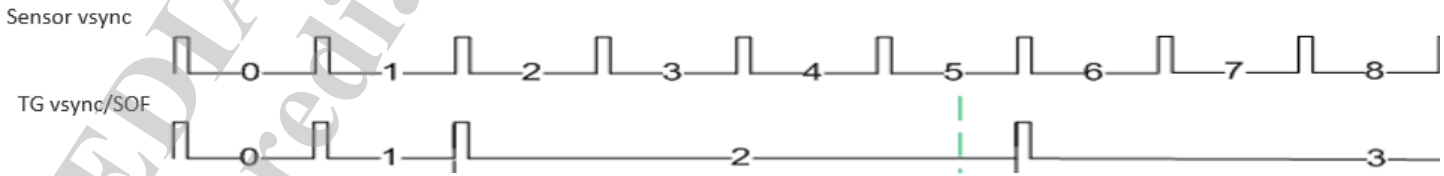
- Front binning.

特定当sensor resolutionXFPS 太大,一个isp处理不了, 可以在isp端把sensor的数据做binning average的动作

- Subsample.

多个seninf 的sof信号才触发一次tg的vsync 信号, 用在high speed video场景较多.

Subsample = 4:
ISR handler have receive event at each subsample signal



ISP COMMON KNOWLEDGE<4>

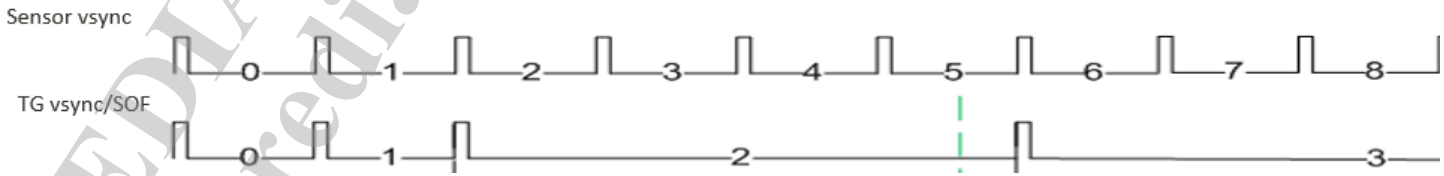
- ESD机制

最初按照字母含义就是打静电时,sensor防呆机制.
后来常用于sensor各种原因没有数据时的场景.
原理是:当isp一定时间等不到sensor p1done信号后,重启sensor的过程.(卡顿)

- Subsample.

多个seninf 的sof信号才触发一次tg的vsync 信号, 用在high speed video场景较多.

Subsample = 4:
ISR handler have receive event at each subsample signal



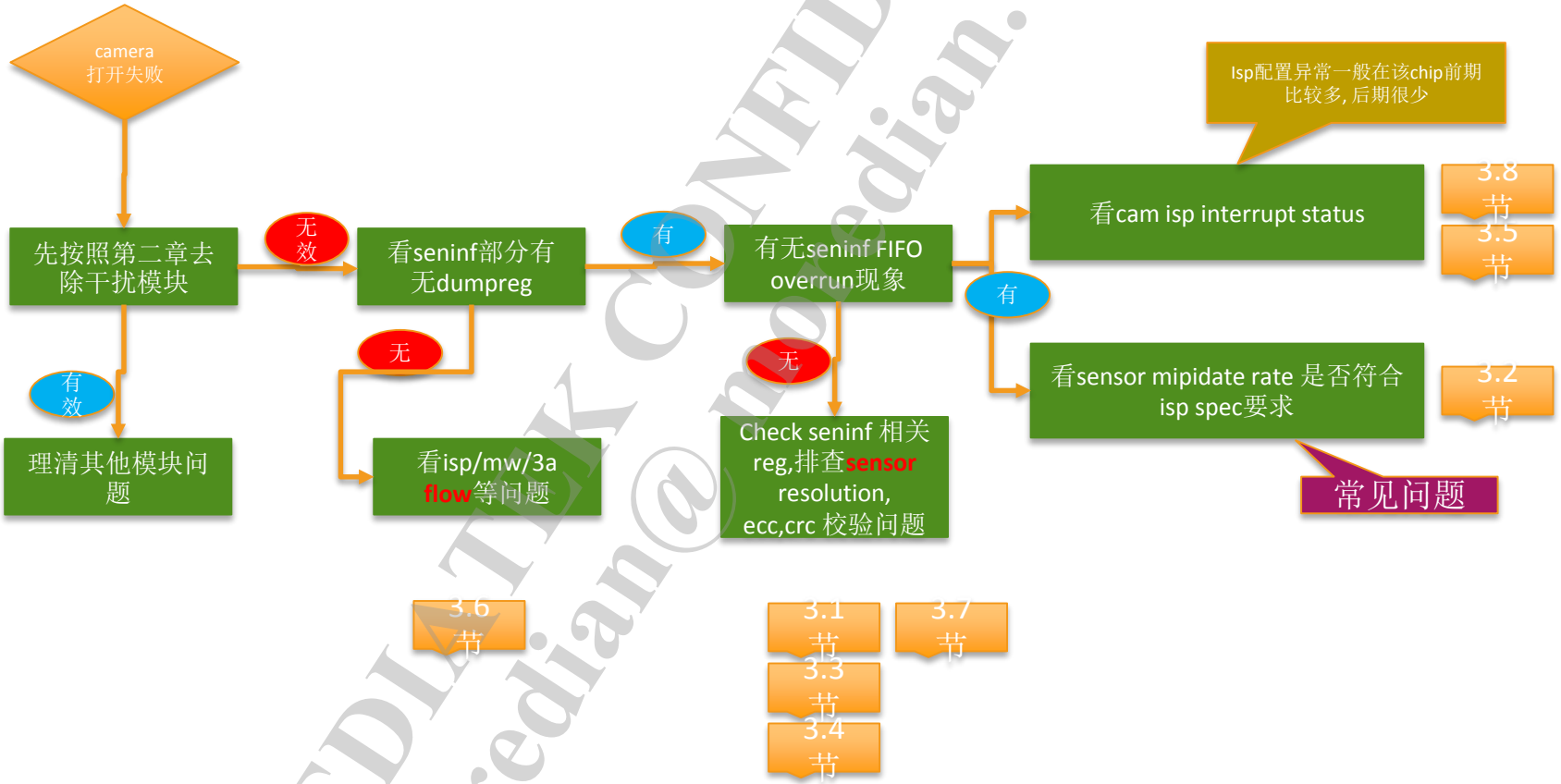
1.2 分析PIPECHECK的CODE FLOW, 作用说明.

PIPE CHECK

- PIPE check 是为方便客户定位isp相关问题加的判断机制.
- 当有isp相关异常时, 会打印所有'嫌疑犯'信息方便排查.
- 使用方法:
  - 参考第三章:CASE SHARE 案例分析.

1.3 解决问题SOP

1.3 解决问题SOP



2. 常用DEBUG COMMAND

2.1 畫面顯示magic number方法

- 可以快速定位屏幕畫面和錄影過程在log上面的相對位置

adb shell setprop debug.cam.drawid 1

2.2 开启常规isp的log

- 在eng版本下, 起码是user-debug版本:

```
adb shell setprop debug.camera.log.p1node 2
adb shell setprop debuglog.imageio.iopipe 3
adb shell setprop debuglog.imageio.iopipet 3
adb shell setprop debuglog.imageio.pipe 3
adb shell setprop debug.isp 2
```

- 若后期需要mtk协助分析isp异常, 建议按照这里的说明提供mtklog.不然后面还要再麻烦抓取.

2.3 ccu关闭命令

- Ccu是mt6771 上使用较多, 若配置错误也可能会有isp部分错误.
- 可以先关闭CCU 理清问题点.

```
adb shell setprop debug.ae_mgr_ccu.enable 0
adb shell setprop debug.af_ccu.disable 1
adb shell setprop debug.ccu_mgr_ccu.enable 0
```

2.4 pdaf关闭方法

- 理清是PDAF还是isp问题, 可以通过adb cmd 直接关闭pdfa 测试.

```
adb shell setprop debug.pdflow.disable 1
```

- 版本在mt6771开始支持.
- 若要看贵司codebase是否有支持,可以在AfMgr::getPdInfoForSttCtrl()函数中看是否有上述cmd.

2.5 isp module关闭命令

- 对于图像画面条纹或颜色异常,要理清问题可以:

1. 先拿raw看看,若raw没问题.
2. 关isp module

```
adb shell setprop isp.obc2.disable 1
adb shell setprop isp.pgn.disable 1
adb shell setprop isp.ggm.disable 1
adb shell setprop isp.lce.disable 1

adb shell setprop isp.dbs.disable 1
adb shell setprop isp.obc.disable 1
adb shell setprop isp.bpc.disable 1
```

```
adb shell setprop isp.nr1.disable 1
adb shell setprop isp.rpg.disable 1
adb shell setprop isp.pgn.disable 1
adb shell setprop isp.udm.disable 1
adb shell setprop isp.ccm.disable 1
adb shell setprop isp.g2c.disable 1
adb shell setprop isp.anr.disable 1
adb shell setprop isp.ccr.disable 1
adb shell setprop isp.pca.disable 1
adb shell setprop isp.ee.disable 1
adb shell setprop isp.anr2.disable 1
adb shell setprop isp.bok.disable 1
adb shell setprop isp.anr_tbl.disable 1
```

2.6 raw yuv dump cmd

- 和isp40 dump 命令兼容.

6. p2-0000-wroto-5120x2880-10240_0_0-yuy2.yuv

b. **Pure MDP**

i. **Select dump buffer**

1. dump MDP all output

adb shell "setprop debug.camera.dump.mdp 1"

ii. **File name format**

1. mdp-[FrameNo]-out-[Hight]x[Width].yuv
2. **Example**

mdp-1234-out-1920x1080.yuv

3. **Pull buffer from device**

adb pull /sdcard/camera_dump/

4. **Clear buffer of device**

adb shell rm -rf /sdcard/camera_dump/

a. **ISP + MDP**

i. **Select dump buffer port** (using bitmask)

1. dump Pass2 IMGO input

adb shell "setprop debug.camera.dump.p2.in 1"

2. dump Pass2 RRZO input

adb shell "setprop debug.camera.dump.p2.in 2"

3. dump Pass2 WDMA output

adb shell "setprop debug.camera.dump.p2.out 1"

4. dump Pass2 WROT output

adb shell "setprop debug.camera.dump.p2.out 2"

5. dump Pass2 IMG2Ooutput

adb shell "setprop debug.camera.dump.p2.out 4"

ii. **Dump all buffer**

1. dump Pass2 all input/output

adb shell "setprop debug.camera.dump.p2 1"

iii. **Indicate dump condition** (Not necessary)

1. Only dump Pass2 if IMGO input

adb shell "setprop debug.camera.dump.p2.cond.in 1"

2. Only dump Pass2 if RRZOinput

adb shell "setprop debug.camera.dump.p2.cond.in 2"

iv. **File name format**

1. p2-[FrameNo]-[Port]-[Hight]x[Width]-[Stride 1]_[Stride 2]_[Stride 3]-[Format].yuv
2. **Port**

imgo, rrzo, wdmao, wroto, img2o

3. **Format**

bayer10, fg_bayer10, yv12, nv21, yuy2

4. **Bayer Order**

RAW_B = "b"

RAW_Gb = "gb"

RAW_Gr = "gr"

RAW_R = "r"

2.7 isp走高dvfs cmd.

- 拉高dvfs的频率电压测试是否是isp分配不合理

```
adb shell "echo 1 > /sys/module/mmdvfs_pmqos/parameters/force_step"
adb shell "echo 0 > /sys/module/mmdvfs_pmqos/parameters/force_step"
```

2.8 拉高DDR 频率

- 理清是否是DDR写入太慢问题.

adb shell "echo kir_emi 0 > /sys/devices/platform/10012000.dvfsrc_top/helio-dvfsrc/dvfsrc_debug"

3. CASE SHARE

3.1 Sensor 吐数据不足

- 先看PIPE CHECK 结果:

17556 03-12 01:38:38.800968 609 4752 E ifunc_cam: [vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.pipechk.cpp, Check_Start, line0182] ERROR: [0x0]:accumulated err int_status:0x20000020, int3_status:0x0

0x20000020 =0b10000000000000000000000100000 bit5, grab error,bit29 dma error. 说明tg 没有数据

17574 03-12 01:38:38.801684 609 4752 E ifunc_cam: [vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.pipechk.cpp, DMAO_STATUS, line0751] ERROR: [0x0]:[dmao fifo-F]

17607 03-12 01:38:38.802819 609 4752 E ifunc_cam: [vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.pipechk.cpp, TG_CHECK, line2908] ERROR: [0x0]:current seninf vertical data is small than grab window_v:**1917_1944**

17610 03-12 01:38:38.804053 609 4752 E ifunc_cam: [vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.pipechk.cpp, TG_CHECK, line2960] ERROR: [0x0]:TG checkl fail

17613 03-12 01:38:38.804229 609 4752 E ifunc_cam: [vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.pipechk.cpp, TG_CHECK, line2968] ERROR: [0x0]:start dump seninf info

3.1 Sensor 吐数据不足

再看seninf dump结果:

20147 03-12 01:38:39.122888 609 4752 D SeninfDrv: [debug]SENINF1_CSI2_CTL(0x209617f), SENINF2_CSI2_CTL(0x86160), SENINF3_CSI2_CTL(0x86160), SENINF5_CSI2_CTL(0x2096173)

20148 03-12 01:38:39.122911 609 4752 D SeninfDrv: [debug]SENINF1_PkCnt(0x58b), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x2a6)

20149 03-12 01:38:39.124013 609 4752 D SeninfDrv: [debug]SENINF1_PkCnt(0x605), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x2a6)

20151 03-12 01:38:39.125131 609 4752 D SeninfDrv: [debug]SENINF1_PkCnt(0x682), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x2a6)

20152 03-12 01:38:39.126245 609 4752 D SeninfDrv: [debug]SENINF1_IRQ(0x44), SENINF2_IRQ(0x0), SENINF3_IRQ(0x0), SENINF5_IRQ(0x0), EXT_IRQ(1:0x0, 2:0x0, 3:0x0, 4:0x2)

20153 03-12 01:38:39.126282 609 4752 D SeninfDrv: [debug]SENINF1_MUX_CTRL(0xa6df8080), SENINF1_MUX_INTSTA(0x18), SENINF1_MUX_DEBUG_2(0x12000d80)

20154 03-12 01:38:39.126304 609 4752 D SeninfDrv: [debug]SENINF2_MUX_CTRL(0xa6df8080), SENINF2_MUX_INTSTA(0x18), SENINF2_MUX_DEBUG_2(0xa20077c)

20155 03-12 01:38:39.126324 609 4752 D SeninfDrv: [debug]SENINF3_MUX_CTRL(0x6df0080), SENINF3_MUX_INTSTA(0x0), SENINF3_MUX_DEBUG_2(0x0)

20156 03-12 01:38:39.126344 609 4752 D SeninfDrv: [debug]SENINF4_MUX_CTRL(0x6df0080), SENINF4_MUX_INTSTA(0x0), SENINF4_MUX_DEBUG_2(0x0)

20157 03-12 01:38:39.126363 609 4752 D SeninfDrv: [debug]SENINF5_MUX_CTRL(0x6df0080), SENINF5_MUX_INTSTA(0x0), SENINF5_MUX_DEBUG_2(0x0)

20158 03-12 01:38:39.126382 609 4752 D SeninfDrv: [debug]SENINF6_MUX_CTRL(0x6df0080), SENINF6_MUX_INTSTA(0x0), SENINF6_MUX_DEBUG_2(0x0)

3.1 Sensor 吐数据不足

- Tips:为什么pipe check 中拿到的resolution和seninf debug的多一条line?
- 答: isp 部分设计上tg register 多count 一条line, 是正常现象.

3.2 SENSOR FIFO OVERRUN 问题

3.2 sensor fifo overrun 问题

- 先看pipe check信息

E ifunc_camsv:
[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_camsv.cpp, PipeCheck, line0880] ERROR: [0x8]:start PipeCheck when deque fail at wait signal
E ifunc_camsv:
[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_camsv.cpp, TG_CHECK, line1553] ERROR: [0x8]:seninf horizontal data is small than grab window_w:0_4656
E ifunc_cam:
[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.pipechk.cpp, TG_CHECK, line2881] ERROR: [0x0]:current seninf vertical data is small than grab window_v:0_3496
E ifunc_cam:
[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.pipechk.cpp, TG_CHECK, line2933] ERROR: [0x0]:TG checkl fail
E ifunc_cam:
[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.pipechk.cpp, TG_CHECK, line2941] ERROR: [0x0]:start dump seninf info

问题点:
上面看log, 其实异常log很多, 这里没贴.当中间有说标红的部分,就要看一下seninf的状态了.

3.2 sensor fifo overrun 问题

- 再看seninf debug 信息

D SeninfDrv: [debug]SENINF_TOP_MUX_CTRL(0x43010) SENINF_TOP_CAM_MUX_CTRL(0x76543210) ISP_clk(546000)
D SeninfDrv: [debug]SENINF1_CSI2_INT_STATUS(0x5c000044), CLR SENINF1_CSI2_INT_STATUS(0x5c000044)
D SeninfDrv: [debug]SENINF2_CSI2_INT_STATUS(0x0), CLR SENINF2_CSI2_INT_STATUS(0x0)
D SeninfDrv: [debug]SENINF3_CSI2_INT_STATUS(0x0), CLR SENINF3_CSI2_INT_STATUS(0x0)
D SeninfDrv: [debug]SENINF5_CSI2_INT_STATUS(0x0), CLR SENINF5_CSI2_INT_STATUS(0x0)
D SeninfDrv: [debug]SENINF1_CSI2_CTL(0x8096160), SENINF2_CSI2_CTL(0x86160), SENINF3_CSI2_CTL(0x86160), SENINF5_CSI2_CTL(0x86160)
D SeninfDrv: [debug]SENINF1_PkCnt(0x0), SENINF2_PkCnt(0x0), SENINF3_PkCnt(0x0), SENINF5_PkCnt(0x0)
D SeninfDrv: [debug]SENINF1_IRQ(0x5c009044), SENINF2_IRQ(0x0), SENINF3_IRQ(0x0), SENINF5_IRQ(0x0), EXT_IRQ(1:0x0, 2:0x0, 3:0x0, 4:0x0)
D SeninfDrv: [debug]SENINF1_MUX_CTRL(0xa6df8080), SENINF1_MUX_INTSTA(0x19), SENINF1_MUX_DEBUG_2(0x12300da8)
D SeninfDrv: [debug]after reset overrun, SENINF1_MUX_CTRL(0xa6df8080), SENINF1_MUX_INTSTA(0x19), SENINF1_MUX_DEBUG_2(0x12300da8)

问题点:
SENINF1_MUX_INTSTA reg 状态错误, 说明sensor 吐数据太快, 考虑降低mipi daterate

3.3 SENINF ECC CRC校验问题

3.3 seninf ecc crc校验问题(isp20 isp30)

- 看seninf debug 信息(isp30 如下)

337339 01-02 10:07:03.694 <3>[36215.973262] [ISP][0x15008120 A6DF8180],[0x15008124 8000007F],[**0x15008128 0000001A**],[0x1500812C 00000000] (1)[16038:PDFlowCtrlThd][ISP][0x15008130 0700C303],[**0x15008134 0A200798**],[0x15008138 0A200798],[0x1500813C 00000000]

337344 01-02 10:07:03.694 <3>[36215.973344] [ISP][0x150083B0 80007FFF],**[0x150083B4 00001015]==> 这里crc 错误.**,[0x150083B8 00000012],[0x150083BC 00000799] (1)[16038:PDFlowCtrlThd][ISP][0x150083C0 00000000],[0x150083C4 00000000],[0x150083C8 00000001],[0x150083CC 01010101]

337423 01-02 10:07:03.695 <3>[36215.974315] [ISP]0x15008D30 0000C303 (1)[16038:PDFlowCtrlThd][ISP]**0x15008D34 00000000 ==> vc size**

3.3 seninf ecc crc校验问题(isp40 isp50)

- 再看seninf debug 信息

D SeninfDrv: [debug]SENINF_TOP_MUX_CTRL(0x43010) SENINF_TOP_CAM_MUX_CTRL(0x76543210) ISP_clk(546000)
D SeninfDrv: [debug]SENINF1_CSI2_INT_STATUS(0x5c000044), CLR SENINF1_CSI2_INT_STATUS(0x5c000044)
D SeninfDrv: [debug]SENINF2_CSI2_INT_STATUS(0x0), CLR SENINF2_CSI2_INT_STATUS(0x0)
D SeninfDrv: [debug]SENINF3_CSI2_INT_STATUS(0x0), CLR SENINF3_CSI2_INT_STATUS(0x0)
D SeninfDrv: [debug]SENINF5_CSI2_INT_STATUS(0x0), CLR SENINF5_CSI2_INT_STATUS(0x0)
D SeninfDrv: [debug]SENINF1_CSI2_CTL(0x8096160), SENINF2_CSI2_CTL(0x86160), SENINF3_CSI2_CTL(0x86160), SENINF5_CSI2_CTL(0x86160)
D SeninfDrv: [debug]SENINF1_PkCnt(0x0), SENINF2_PkCnt(0x0), SENINF3_PkCnt(0x0), SENINF5_PkCnt(0x0)
D SeninfDrv: [debug]SENINF1_IRQ(0x5c009044), SENINF2_IRQ(0x0), SENINF3_IRQ(0x0), SENINF5_IRQ(0x0), EXT_IRQ(1:0x0, 2:0x0, 3:0x0, 4:0x0)

问题点:

SENINF1_CSI2_INT_STATUS reg 状态错误, 要考虑看硬件连接状态,也可能是sensor不吐数据.

3.3 seninf ecc crc校验问题(isp40 isp50)

18040A14	SENINF1_CSI2_INT_STATUS	csi2 Interrupt status	00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		MERGE_FIFO_AF	TRIO3_ESCAPE_CODE_DETECT_STA	TRIO2_ESCAPE_CODE_DETECT_STA	TRIO1_ESCAPE_CODE_DETECT_STA	TRIO0_ESCAPE_CODE_DETECT_STA		ERR_FRAME_SYNC_S7_STA	ERR_FRAME_SYNC_S6_STA	ERR_LANE_RESYNC_STA	ERR_FRAME_SYNC_S5_STA	ERR_FRAME_SYNC_S4_STA	ERR_FRAME_SYNC_S3_STA	ERR_FRAME_SYNC_S2_STA	ERR_FRAME_SYNC_S1_STA	ERR_FRAME_SYNC_S0_STA
Type		RO	RO	RO	RO	RO		RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset		0	0	0	0	0		0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FE_STA	GS_STA	LS_STA	FS_STA	ERR_SOT_SYNC_HS_LNRD3_STA	ERR_SOT_SYNC_HS_LNRD2_STA	ERR_SOT_SYNC_HS_LNRD1_STA	ERR_SOT_SYNC_HS_LNRD0_STA	ERR_MULTI_LANE_SYNC_STA		ERR_CRC_STA	ERR_ECC_DOUBLE_STA	ERR_ECC_CORRECTED_STA	ERR_ECC_NO_ERROR_STA	ERR_ID_STA	ERR_FRAME_SYNC_STA
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO		RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0

Bit(s)	Name	Description
30	MERGE_FIFO_AF	Asserted when the merge fifo is almost full
29	TRIO3_ESCAPE_CODE_DETECT_STA	Asserted when the escape code of trio2 is detected
28	TRIO2_ESCAPE_CODE_DETECT_STA	Asserted when the escape code of trio2 is detected

➢Normal IRQ : 0x9004

seninf1 18040A00
seninf2 18041A00
seninf3 18042A00
seninf4 18043A00
seninf5 18044A00

3.4 SENINF CLOCK设定BUG

3.4 seninf clock设定bug

- 先看PipeCheck
- 这题pipecheck 错误点在seninf v = 0. 但seninf dump reg是正常的.

7553 02-23 18:25:07.115 4764 14423 E ifunc_cam: [vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.pipechk.cpp, TG_CHECK, line2915] ERROR: [0x1]:current seninf vertical data is small than grab window_v:0_1728

7555 02-23 18:25:07.115 4764 14423 E ifunc_cam: [vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.pipechk.cpp, TG_CHECK, line2954] ERROR: [0x1]:TG is in idle status:0x1

7558 02-23 18:25:07.115 4764 14423 E ifunc_cam: [vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.pipechk.cpp, TG_CHECK, line2967] ERROR: [0x1]:TG checkl fail

7560 02-23 18:25:07.115 4764 14423 E ifunc_cam: [vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.pipechk.cpp, TG_CHECK, line2975] ERROR: [0x1]:start dump seninf info

3.4 seninf clock设定bug

- 再看seninf debug 信息

02-23 18:25:07.146 4764 14423 D **SeninfDrv: [debug]f_fseninf_ck(312000)**
7225 02-23 18:25:05.141 4764 14423 D SeninfDrv: [debug]SENINF1_CSI2_INT_STATUS(0x0), CLR SENINF1_CSI2_INT_STATUS(0x0)
7230 02-23 18:25:05.144 4764 14423 D SeninfDrv: [debug]SENINF1_PkCnt(0x6c4), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x1)
7232 02-23 18:25:05.146 4764 14423 D SeninfDrv: [debug]SENINF1_PkCnt(0x6c4), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x1)
7234 02-23 18:25:05.147 4764 14423 D SeninfDrv: [debug]SENINF1_MUX_CTRL(0xa6df8080), SENINF1_MUX_INTSTA(0x18), SENINF1_MUX_DEBUG_2(0x91806c0)

问题点:

1. 2328 X 1728 是imx519 的preview setting / 正常.
2. SENINF1_CSI2_INT_STATUS reg 0
3. SENINF1_PkCnt(0x6c4) 两次没有递增.

最终发现seninf clk isp内部设定太大导致.最终office版本为:

SeninfDrv: [debug]f_fseninf_ck(208000)

3.5 ISP FIFO OVERRUN问题

ISP FIFO 概念

- DMA fifo工作机制:

每个dma有其自己的fifo,其值大小固定,当data在fifo中存到fifo的一定percent后(比如说50%),就会call EMI 写data出去,或者换种说法, 提前发request 给EMI ,要求其写data出去吧.
fifo一边从source端收数据, 一边通过EMI 写data出去.这样一般情况下, source端in 产生的数据量(resolution X fps, 还要考虑vblk等信息) 小于 EMI out 消耗的数据 , pipe正常.
注意EMI 不仅要处理camera isp dmao的数据, 在分片时序中还要处理其他非camera
的数据.(可能此時頻寬忙碌中 emi 沒辦法快速服務 camera isp dmao)

- Seninf fifooverrun 和 isp fifo overrun 概念区别

p1和seninf有各自的fifo.

seninf fifo overrun,看SENINF1_MUX_INTSTA
p1 fifo overrun 看 CAM_B: raw_int_err:0x100000_0x10

ISP FIFO 概念

- DMA_ERR_ST定义:

/kernel-4.4/drivers/misc/mediatek/cameraisp/src/isp_50/inc/cam_regs.h

```
33/* err status */
34#define TG_ERR_ST               (1L<<4)
35#define TG_GBERR_ST             (1L<<5)
36#define CQ_CODE_ERR_ST      (1L<<6)
37#define CQ_APB_ERR_ST       (1L<<7)
38#define CQ_VS_ERR_ST        (1L<<8)
39#define AMX_ERR_ST           (1L<<15)
40#define RMX_ERR_ST           (1L<<16)
41#define BMX_ERR_ST           (1L<<17)
42#define RRZO_ERR_ST          (1L<<18)
43#define AFO_ERR_ST           (1L<<19)
44#define IMGO_ERR_ST          (1L<<20)
45#define AAO_ERR_ST           (1L<<21)
46#define PSO_ERR_ST           (1L<<22)
47#define LCSO_ERR_ST          (1L<<23)
48#define BNR_ERR_ST           (1L<<24)
49#define LSC_ERR_ST            (1L<<25)
50#define UFGO_ERR_ST         (1L<<26)
51#define UFEO_ERR_ST         (1L<<27)
52#define PDO_ERR_ST           (1L<<28)
53#define DMA_ERR_ST           (1L<<29)
```

```
1.  6114 ErrStatus = IrqStatus & IspInfo.IrqInfo.ErrMask[module][SIGNAL_INT];
2.  6115 WarnStatus = IrqStatus & IspInfo.IrqInfo.WarnMask[module][SIGNAL_INT];
3.  6116 IrqStatus = IrqStatus & IspInfo.IrqInfo.Mask[module][SIGNAL_INT];
```

```
1.  130/**
2.  131 *    IRQ Error Mask
3.  132 */
4.  133#define INT_ST_MASK_CAM_ERR     (\
5.  134     TG_ERR_ST |\
6.  135     TG_GBERR_ST |\
7.  136     CQ_CODE_ERR_ST |\
8.  137     CQ_APB_ERR_ST |\
9.  138     CQ_VS_ERR_ST |\
10. 139     AMX_ERR_ST |\
11. 140     RMX_ERR_ST |\
12. 141     BMX_ERR_ST |\
13. 142     BNR_ERR_ST |\
14. 143     LSC_ERR_ST |\
15. 144     DMA_ERR_ST)
16. 145
```

3.5 isp fifo overrun 的问题

- 先看seninf debug 信息, 数据正常

291059 01-08 17:44:24.335900 590 26679 D SeninfDrv: [debug]SENINF_TOP_MUX_CTRL(**0x43214**) SENINF_TOP_CAM_MUX_CTRL(0x76543200) ISP_clk(0)
291060 01-08 17:44:24.339127 590 26679 D SeninfDrv: [debug]SENINF1_CSI2_INT_STATUS(0x0), CLR SENINF1_CSI2_INT_STATUS(0x0)
291061 01-08 17:44:24.342338 590 26679 D SeninfDrv: [debug]SENINF2_CSI2_INT_STATUS(0x0), CLR SENINF2_CSI2_INT_STATUS(0x0)
291062 01-08 17:44:24.345528 590 26679 D SeninfDrv: [debug]SENINF3_CSI2_INT_STATUS(0x0), CLR SENINF3_CSI2_INT_STATUS(0x0)
291063 01-08 17:44:24.348617 590 26679 D SeninfDrv: [debug]SENINF5_CSI2_INT_STATUS(0x44), CLR **SENINF5_CSI2_INT_STATUS**(0x44)
291064 01-08 17:44:24.348641 590 26679 D SeninfDrv: [debug]SENINF1_CSI2_CTL(0x86160), SENINF2_CSI2_CTL(0x86160), SENINF3_CSI2_CTL(0x86160), SENINF5_CSI2_CTL(0x809617f)
291065 01-08 17:44:24.348654 590 26679 D SeninfDrv: [debug]SENINF1_PkCnt(0x0), SENINF2_PkCnt(0x0), SENINF3_PkCnt(0x295), **SENINF5_PkCnt**(0x6db)
291066 01-08 17:44:24.348669 590 26679 D SeninfDrv: [debug]SENINF1_IRQ(0x0), SENINF2_IRQ(0x0), SENINF3_IRQ(0x0), **SENINF5_IRQ**(0x44), EXT_IRQ(1:0x0, 2:0x0, 3:0x0, 4:0x0)
291067 01-08 17:44:24.348681 590 26679 D SeninfDrv: [debug]SENINF1_MUX_CTRL(0xa6df8180), **SENINF1_MUX_INTSTA**(0x18), **SENINF1_MUX_DEBUG_2**(0xa200794)

3.5 isp fifo overrun 的问题

- 再看pipe check 部分有如下error 信息.

554923 02-19 14:16:05.836443 604 10299 I MtkCam/P1NodeImp: [hardwareOps_start] [Cam::0] InitRqeFlow return 4 4 0 - Cam::0 Sensor(4656x2618) Raw(1,0x2)-Proc(4656x2618)-Pure(4656x2618) Bin(4656x2618) BinEn=0 TG(0:0) DTwin(1@0)=0 LSM(1) QLV(2) Ratio(6) SensorCfg(i:0 4656x2618 s:2 b:0 c:1, h:0 f:30 t:0 d:0) ConfigPort[5]:(0x1f) InitParam[R:0 B:10 D:1 Nd:0 Ul:0 Pb:1 Dt:0 Iq:2 F(DataPattern:x0 OffBin:x0 SensorNum:x0 RAW:x0)]

570040 02-19 14:16:13.955178 604 10365 E ifunc_cam: [vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.pipechk.cpp, Check_Start, line0182] ERROR: [0x1]:accumulated err int_status:0x10, int3_status:0x0

- Kernel log 也有这样的信息.

01-09 15:49:26.637482 19911 19911 I
[7474.306329].(1)[19911:kworker/1:1][ISP][ISP_BH_Workqueue] [7471.486204]CAM_B: raw_int_err:0x40000_0x10, raw_int3_err:0x0

SeninfDrv user count 回收问题

mw 概率没有enque buffer 进来

3.6 MW FLOW 问题/ AP问题

3.6.1 SeninfDrv user count 回收问题

- 异常log.

107625 01-29 08:04:17.022351 8392 24321 D SeninfDrv: [init][init]: Entry count 1
107626 01-29 08:04:17.022364 8392 24321 D SeninfDrv: [init]Already inited
116518 01-29 08:04:19.092917 8392 8213 D MtkCam/P1NodeImp: [uninit] [Cam::0] API
116521 01-29 08:04:19.093105 8392 8213 D MtkCam/P1NodeImp: [uninit] [Cam::0] API
117922 01-29 08:04:19.250228 8392 8213 D ImgSensorDrv: [uninit][uninit] imgsensor_drv
117923 01-29 08:04:19.250258 8392 8213 D MtkCam/HalSensor: [powerOff] - <DefaultCa aDevice1> mSensorIdx = 0x0
117924 01-29 08:04:19.250278 8392 8213 D SeninfDrv: [uninit][uninit]: 2
117925 01-29 08:04:19.250294 8392 8213 E SeninfDrv: [uninit]Err(287):Still users
117926 01-29 08:04:19.250304 8392 8213 D MtkCam/Utils/CamMgr: [decSensorCount] [mtkcam-dev1] current sensor count [1]
118193 01-29 08:04:19.278447 8392 24512 D SeninfDrv: [init][init]: Entry count 1
118194 01-29 08:04:19.278464 8392 24512 D SeninfDrv: [init]Already inited
118195 01-29 08:04:19.278469 8392 24512 D ImgSensorDrv: [init][init] imgsensor_drv
118203 01-29 08:04:19.278752 8392 24512 D MtkCam/HalSensor: [powerOn] sensorIdx : 0

注意, 当pip 或vsdof 时, 退出一个camera时也会有这样的log, 是正常现象.因为还有个camera要用.

3.6.1 SeninfDrv user count 回收问题

- 正常的对比log

```
109241 01-01 13:09:01.139014  8797  8804 D MtkCam/P1NodeImp: [uninit] [Cam::0] API +
109244 01-01 13:09:01.139280  8797  8804 D MtkCam/P1NodeImp: [uninit] [Cam::0] API -
109914 01-01 13:09:01.226000  8797  8804 D MtkCam/HalSensor: [powerOff] mSensorIdx : 0
109915 01-01 13:09:01.226000  8797  8804 D MtkCam/HalSensor:
109916 01-01 13:09:01.226139  8797  8804 D MtkCam/HalSensor: [get_boot_mode] Boot Mode 0
109920 01-01 13:09:01.226281  8797  8804 D ImgSensorDrv: [featureControl][featureControl] Skip due to no FeatureId
109921 01-01 13:09:01.226336  8797  8804 D ImgSensorDrv: [close]setFlashproc close sensorIdx:0
109922 01-01 13:09:01.226358  8797  8804 D ImgSensorDrv: [setFlashproc]setFlashproc 11 id:54
109923 01-01 13:09:01.226515  8797  8804 D ImgSensorDrv: [setFlashproc]setFlashproc id:54
109924 01-01 13:09:01.236816  8797  8804 D ImgSensorDrv: [uninit][uninit] imgsensor_drv
109925 01-01 13:09:01.236882  8797  8804 D ImgSensorDrv: [featureControl][featureControl] Skip due to no FeatureId
109926 01-01 13:09:01.236917  8797  8804 D SeninfDrv: [configMclk][Tg0clk]: pcEn=0 freq=24
109927 01-01 13:09:01.237061  8797  8804 D SeninfDrv: [setMclk][setTg1Phase]pcEn(0), freq(24)
109928 01-01 13:09:01.237098  8797  8804 D MtkCam/HalSensor: [setSensorMclk] sensorIdx 0, mclk_src 1, SensorMCLKPLL -
475392373
109929 01-01 13:09:01.237145  8797  8804 D MtkCam/HalSensor: [powerOff] - <DefaultCameraDevice1> mSensorIdx = 0x0
109930 01-01 13:09:01.237170  8797  8804 D SeninfDrv: [uninit][uninit]: 1
109931 01-01 13:09:01.237252  8797  8804 D SeninfDrv: [uninit][uninit]: 0, mfd(16)
109932 01-01 13:09:01.237328  8797  8804 D MtkCam/Utils/CamMgr: [decSensorCount] [mtkcam-dev1] current sensor count [0]
```

3.6.2 mw 概率没有enque buffer 进来(isp40 isp50)

- seninf status log

18580 03-11 22:43:24.682087 572 5281 D SeninfDrv: [debug]SENINF_TOP_MUX_CTRL(0x43210) SENINF_TOP_CAM_MUX_CTRL(0x76543200)
18581 03-11 22:43:24.683791 572 5281 D SeninfDrv: [debug]SENINF1_CSI2_INT_STATUS(0x0), CLR SENINF1_CSI2_INT_STATUS(0x0)
18585 03-11 22:43:24.687134 572 5281 D SeninfDrv: [debug]SENINF1_CSI2_CTL(0x209617f), SENINF2_CSI2_CTL(0x86160), SENINF3_CSI2_CTL(0x86160), SENINF5_CSI2_CTL(0x86160)
18586 03-11 22:43:24.687168 572 5281 D SeninfDrv: [debug]SENINF1_PkCnt(0x9f), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x1)
18587 03-11 22:43:24.688363 572 5281 D SeninfDrv: [debug]SENINF1_PkCnt(0x119), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x1)
18588 03-11 22:43:24.689580 572 5281 D SeninfDrv: [debug]SENINF1_PkCnt(0x196), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x1)
18589 03-11 22:43:24.690695 572 5281 D SeninfDrv: [debug]SENINF1_IRQ(0x1044),
18590 03-11 22:43:24.690731 572 5281 D SeninfDrv: [debug]SENINF1_MUX_CTRL(0xa6df8080), SENINF1_MUX_INTSTA(0x18), SENINF1_MUX_DEBUG_2(0x50002d0)

3.6.2 mw 概率没有enque buffer 进来(isp40 isp50)

- seninf status log

18580 03-11 22:43:24.682087 572 5281 D SeninfDrv: [debug]SENINF_TOP_MUX_CTRL(0x43210) SENINF_TOP_CAM_MUX_CTRL(0x76543200)
18581 03-11 22:43:24.683791 572 5281 D SeninfDrv: [debug]SENINF1_CSI2_INT_STATUS(0x0), CLR SENINF1_CSI2_INT_STATUS(0x0)
18585 03-11 22:43:24.687134 572 5281 D SeninfDrv: [debug]SENINF1_CSI2_CTL(0x209617f), SENINF2_CSI2_CTL(0x86160), SENINF3_CSI2_CTL(0x86160), SENINF5_CSI2_CTL(0x86160)
18586 03-11 22:43:24.687168 572 5281 D SeninfDrv: [debug]SENINF1_PkCnt(0x9f), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x1)
18587 03-11 22:43:24.688363 572 5281 D SeninfDrv: [debug]SENINF1_PkCnt(0x119), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x1)
18588 03-11 22:43:24.689580 572 5281 D SeninfDrv: [debug]SENINF1_PkCnt(0x196), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x1)
18589 03-11 22:43:24.690695 572 5281 D SeninfDrv: [debug]SENINF1_IRQ(0x1044),
18590 03-11 22:43:24.690731 572 5281 D SeninfDrv: [debug]SENINF1_MUX_CTRL(0xa6df8080), SENINF1_MUX_INTSTA(0x18), SENINF1_MUX_DEBUG_2(0x50002d0)

3.6.2 mw 概率没有enque buffer 进来(isp40 isp50)

- P1 flow上看,进入high speed video时, 就没有enque sof 给isp了.

16223 03-11 22:43:22.231031 572 572 D MtkCam/FeatureFlowControl: [0:constructRecordingPipeline] +
16235 03-11 22:43:22.231441 572 572 D MtkCam/FeatureFlowControl: [0:constructRecordingPipeline] sensor mode:3, rawType:0, size:1280x720, fps:120 pixel:2 vhdrMode:0
16264 03-11 22:43:22.235198 572 572 I MtkCam/FeatureFlowControl: [0:constructRecordingPipeline] RRZO Count 16/32
18083 03-11 22:43:22.374464 572 5226 I NormalPipe_Thread: [enqueRequest] N:0 dma:x8 no request sof_idx:8, FrameStatus(2)
18090 03-11 22:43:22.414637 572 5226 I NormalPipe_Thread: [enqueRequest] N:0 dma:x8 no request sof_idx:12, FrameStatus(2)
18097 03-11 22:43:22.454540 572 5226 I NormalPipe_Thread: [enqueRequest] N:0 dma:x8 no request sof_idx:16, FrameStatus(2)
18128 03-11 22:43:22.494650 572 5226 I NormalPipe_Thread: [enqueRequest] N:0 dma:x8 no request sof_idx:20, FrameStatus(2)
18133 03-11 22:43:22.534557 572 5226 I NormalPipe_Thread: [enqueRequest] N:0 dma:x8 no request sof_idx:24, FrameStatus(2)
18137 03-11 22:43:22.574700 572 5226 I NormalPipe_Thread: [enqueRequest] N:0 dma:x8 no request sof_idx:28, FrameStatus(2)

3.6.2 mw 概率没有enque buffer 进来(isp40 isp50)

- P1 flow上看,后面就出现这些常见的错误log,pipecheck仅仅说deque fail fail

18463 03-11 22:43:24.282639 572 5227 E ifunc_cam_buf: [vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.buf_ctrl.cpp, waitBufReady, line0616] ERROR: [0x0]:dma:BUF_CTRL_RRZO start wait:[enque record:SW_buf_cnt:0x4, SW_enq_cnt:0x0, FbcCnt:0x0],[2]time,[3]time,[4]time,[5]time,, waitbufready fail. start fail check
18470 03-11 22:43:24.282706 572 5282 E ifunc_cam_buf: (waitBufReady){#3039:vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.buf_ctrl.cpp}
18471 03-11 22:43:24.282725 572 5282 E ifunc_cam_buf: [vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.buf_ctrl.cpp, PipeCheck, line0190] ERROR: [0x0]:start PipeCheck when deque fail at wait signal
18472 03-11 22:43:24.282725 572 5282 E ifunc_cam_buf: (PipeCheck){#190:vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.buf_ctrl.cpp}

3.6.2 mw 概率没有enque buffer 进来(isp40 isp50)

- 参考如下正常log, 在user 版本应该每个frame都应该有如下信息,表示有enque buffer 进来.

ifunc_cam_buf: [enqueueHwBuf]
[0x1]:BUF_CTRL_RRZO:(0x0_0x0),PA(0xDE00000_0x14A01400),FH_VA(0xF502D400),size(0x2c8f80),enque_sof(86),isUF(0)
ifunc_cam_buf: [enqueueHwBuf]
[0x1]:BUF_CTRL_IMGO:(0x0_0x0),PA(0x27300000_0x14A00400),FH_VA(0xF502C400),size(0x12fc000),enque_sof(86),isUF(0)

- 若需要更多enque buffer detail的信息,可以参考<2.2 开启常规isp的log>, 可以看到如下log.

197765 01-01 04:51:52.282794 520 3142 D NormalPipe_Thread: [enque] Thd: E197774 01-01 04:51:52.283195 520 3142 D NormalPipe_Thread: [enque] FrameMgr::enque+, 0x9500000197776 01-01 04:51:52.283273 520 3142 D NormalPipe_Thread: [enque] Thd: X197777 01-01 04:51:52.283329 520 3142 D NormalPipe_Thread: [dmaCfg] +197779 01-01 04:51:52.283403 520 3142 D NormalPipe_Thread: [dmaCfg] N:0 dma:x6 cmd:0x101c, crop:(0,0,5344,4016), outsize:(5344,4016)197788 01-01 04:51:52.283755 520 3142 D NormalPipe_Thread: [dmaCfg] -197791 01-01 04:51:52.283827 520 3142 D NormalPipe_Thread: [enque] Thd: E197796 01-01 04:51:52.283990 520 3142 D NormalPipe_Thread: [enque] FrameMgr::enque+, 0xB000000197797 01-01 04:51:52.284057 520 3142 D NormalPipe_Thread: [enque] Thd: X197799 01-01 04:51:52.284111 520 3142 D NormalPipe_Thread: [dmaCfg] +197802 01-01 04:51:52.284288 520 3142 D NormalPipe_Thread: [dmaCfg] -

3.6.2 mw 概率没有enque buffer 进来(isp40 isp50)

- 故说明code上mw flow异常,从kernel log找更多的证据.

(4)[5227:IspDeqThd_TG1]: OV16885_REAR_camera_sensor[hs_video_setting] E
……..
(6)[566:kworker/6:2]: [ISP][ISP_BH_Workqueue] [125.086141]CAMA
P1_SOF_12_12(0x0_0x0,0x18000_0x101,0x0,0x17700000,0x1),int_us:33307,cq:0x10800000
(6)[566:kworker/6:2]: [ISP][ISP_BH_Workqueue] [125.119446]CAMA
P1_SOF_16_16(0x0_0x0,0x18000_0x101,0x0,0x17700000,0x1),int_us:33306,cq:0x10800000
(6)[566:kworker/6:2]: [ISP][ISP_BH_Workqueue] [125.152753]CAMA
P1_SOF_20_20(0x0_0x0,0x18000_0x101,0x0,0x17700000,0x1),int_us:33305,cq:0x10800000
(6)[566:kworker/6:2]: [ISP][ISP_BH_Workqueue] [125.186059]CAMA
P1_SOF_24_24(0x0_0x0,0x18000_0x101,0x0,0x17700000,0x1),int_us:33307,cq:0x10800000
(6)[566:kworker/6:2]: [ISP][ISP_BH_Workqueue] [125.219365]CAMA
P1_SOF_28_28(0x0_0x0,0x18000_0x101,0x0,0x17700000,0x1),int_us:33306,cq:0x10800000
(7)[1222:kworker/7:2]: [ISP][ISP_BH_Workqueue] [125.252672]CAMA
P1_SOF_32_32(0x0_0x0,0x18000_0x101,0x0,0x17700000,0x1),int_us:33307,cq:0x10800000
(6)[566:kworker/6:2]: [ISP][ISP_BH_Workqueue] [125.285980]CAMA
P1_SOF_36_36(0x0_0x0,0x18000_0x101,0x0,0x17700000,0x1),int_us:33309,cq:0x10800000

参考:
4.6 P1_SOF P1_DON Lost p1 done

3.6.3 mw 概率没有enque buffer 进来(isp20)

- 先这类问题一般出现在客户客制化ap 时, 现象也是 isp_wait_irq fail. (这个log是旧平台的.mt6771暂时无客户cr)

正常时候, rrzo有enque进来的log
8616 01-04 07:33:22.042367 482 5007 D iio/ifunc: [enqueueHwBuf] pass1:dma(4),id(0),size(0x249f00),VA(0x9be6c000),PA(0x4600000),S/C(0/0),bufidx(2),replace:new(0x3c00000),bufidx(2),bImdMode(0)
8630 01-04 07:33:22.042844 482 5007 D iio/ifunc: [enqueueHwBuf] pass1:dma(8),id(0),size(0x8ca00),VA(0x9c587000),PA(0x4100000),S/C(0/0),bufidx(2),replace:new(0x5200000),bufidx(2),bImdMode(0)
异常时候, RRZO 没有enque 进来,只有imgo了.
8756 01-04 07:33:22.173277 482 5007 D iio/ifunc: [enqueueHwBuf] pass1:dma(4),id(0),size(0x249f00),VA(0x9bb95000),PA(0x4a00000),S/C(0/0),bufidx(0),replace:new(0x4200000),bufidx(0),bImdMode(0)
这里把最后一次正常的rrzo enque buffer消耗掉PA(0x5200000)
8880 01-04 07:33:22.393588 482 5007 D iio/ifunc: [waitBufReady] waitBufReady[8]:[1,ffffffff] rtBufCtrl.ctrl(4)/id(8)/ptr(0x0)
8881 01-04 07:33:22.393725 482 5007 D iio/ifunc: [dequeueHwBuf] pass1:i(0),dma(8),id(0x0),size(0x8ca00),xsize(0x4af),VA(0x9b7a4000),PA(0x5200000),crop(0x0,0x0,0x4b0,0x1e0),count(1),cur sof(42),frm_cnt(41),mag(0x25),rawType(0)
后面就开始报错了.
9041 01-04 07:33:22.968880 482 5009 E IspDrv : {IspDrv} [vendor/mediatek/proprietary/hardware/mtkcam/legacy/platform/mt6739/core/drv/isp/isp_drv.cpp, waitIrq, line2380] ERROR: ISP_WAIT_IRQ fail(-1). Type(0), Status(0x00000400), Timeout(501).

3.7 ISP FIFO OVERRUN问题

3.7 sensor 无vsync 信号问题

- 先看seninf debug 信息, 数据正常

打静电后, seninf 记录原来的数据,容易让人误解,但还好**SENINF5_CSI2_INT_STATUS 有指示问题点.**

275007 02-28 19:54:47.030875 612 18983 D SeninfDrv: [debug]SENINF1_PkCnt(0x1), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x400)

275008 02-28 19:54:47.031967 612 18983 D SeninfDrv: [debug]SENINF1_PkCnt(0x1), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x453)

275009 02-28 19:54:47.033062 612 18983 D SeninfDrv: [debug]SENINF1_PkCnt(0x1), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x4a6)

sensor吐出來的數據就有問題，bit22 异常.沒收到frame start 和frame end，因此無VSYNC訊號

02-28 19:54:47.030808 612 18983 D SeninfDrv: [debug]SENINF5_CSI2_INT_STATUS(0x400044), CLR SENINF5_CSI2_INT_STATUS(0x400044)

275011 02-28 19:54:47.034190 612 18983 D SeninfDrv: [debug]SENINF1_MUX_CTRL(0xa6df8080), SENINF1_MUX_INTSTA(0x18), SENINF1_MUX_DEBUG_2(0xcc0072b)

275289 02-28 19:54:47.043020 612 18983 W NormalPipe: [reset] WARNING: ESD flow start +

275290 02-28 19:54:47.043035 612 18983 W NormalPipe: [reset] WARNING: Reset+, SenIdx=1

276544 02-28 19:54:51.987660 612 18983 E Isp_Drv : Isp_Drv[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/isp/isp_drv.cpp, waitIrq, line0522] ERROR: ISP(0x1)_WAIT_IRQ fail(14). Wait Status(0x00000001), Timeout(5000).

3.7 sensor 无vsync 信号问题

- 再看pipe check 部分有如下error 信息.

274854 02-28 19:54:46.703848 612 18983 E Isp_Drv : Isp_Drv[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/isp/isp_drv.cpp, waitIrq, line0522] ERROR: ISP(0x1)_WAIT_IRQ fail(14). Wait Status(0x40000000), Timeout(400).
274862 02-28 19:54:46.706724 612 18983 E ifunc_cam: [vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.pipechk.cpp, Check_Start, line0148] ERROR: [0x1]:accumulated err int_status:0x20000020, int3_status:0x0

- isp部分已经说TG_GBERR_ST了, 但为什么seninf reg看,有收到的数据.
- ==>MUX的值有可能停留在最後一次正常收到的寬高。以這題來說interupt status不正常。後面其實已經收不到data了

3.8 内部开发版本遇到的问题SHARE

内部开发版本遇到的问题share

- 先看seninf debug 信息:

02-04 18:08:24.217 603 25456 D SeninfDrv: [debug]SENINF1_CSI2_INT_STATUS(**0x1c400**
SENINF1_CSI2_INT_STATUS(0x1c400044)
02-04 18:08:24.220 603 25456 D SeninfDrv: [debug]SENINF1_PkCnt(0x28b), SENI[illegible]2_PkCnt(0x1),
SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x1)
02-04 18:08:24.221 603 25456 D SeninfDrv: [debug]SENINF1_PkCnt(0x32e), SENINF2_PkCnt(0x1),
SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x1)
02-04 18:08:24.222 603 25456 D SeninfDrv: [debug]SENINF1_PkCnt(0x3d5), SENINF2_PkCnt(0x1),
SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x1)
02-04 18:08:24.224 603 25456 D SeninfDrv: [debug]SENINF1_IRQ(**0x1c408044**), SENINF2_IRQ(0x0),
SENINF3_IRQ(0x0), SENINF5_IRQ(0x0), EXT_IRQ(1:0x0, 2:0x0, 3:0x0, 4:0x0)
02-04 18:08:24.224 603 25456 D SeninfDrv: [debug]SENINF1_MUX_CTRL(0xa6df8080),
SENINF1_MUX_INTSTA(0x19), SENINF1_MUX_DEBUG_2(0x7800438)
02-04 18:08:24.225 603 25456 D SeninfDrv: [debug]after reset overrun,
SENINF1_MUX_CTRL(0xa6df8080), SENINF1_MUX_INTSTA(0x19),
SENINF1_MUX_DEBUG_2(0x7800438)

为什么这个 value 会变化

3.8.1 SMI 繁忙

1. 变化原因是因为 isp回堵报错了, 这些reg status 已经错乱.没有参考价值.
2. 有fifo overrun, 先看isp 是否正常,再考虑sensor mipi clk

- 再看pipe check 部分有如下error 信息.

E ifunc_cam:
[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.pipechk.cpp, PATH_CHECK, line0406] ERROR: [0x0]:CQ SMI request error
E ifunc_cam:
[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.pipechk.cpp, DMAO_STATUS, line0723] ERROR: [0x0]:CQ can't read data from DRAM
E ifunc_cam:
[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_function_cam.pipechk.cpp, PS_CHECK, line1655] ERROR: [0x0]:warning, quality lost

3.8.1 SMI繁忙

- 再看seninf debug信息.

SeninfDrv: [debug]SENINF_TOP_MUX_CTRL(0x43210) SENINF_TOP_CAM_MUX_CTRL(0x76543210)
SeninfDrv: [debug]SENINF1_CSI2_INT_STATUS(0x1c008044), CLR SENINF1_CSI2_INT_STATUS(0x1c008044)
SeninfDrv: [debug]SENINF1_CSI2_CTL(0x2096160), SENINF2_CSI2_CTL(0x86160), SENINF3_CSI2_CTL(0x86160), SENINF5_CSI2_CTL(0x86160)
SeninfDrv: [debug]SENINF1_PkCnt(0x19d), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x1)
SeninfDrv: [debug]SENINF1_PkCnt(0x254), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x1)
SeninfDrv: [debug]SENINF1_PkCnt(0x30d), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x1)
SeninfDrv: [debug]SENINF1_IRQ(0x1c009044), SENINF2_IRQ(0x0), SENINF3_IRQ(0x0), SENINF5_IRQ(0x0), EXT_IRQ(1:0x0, 2:0x0, 3:0x0, 4:0x0)
SeninfDrv: [debug]SENINF1_MUX_CTRL(0xa6df8080), SENINF1_MUX_INTSTA(0x19), SENINF1_MUX_DEBUG_2(0x7800438)
SeninfDrv: [debug]after reset overrun, SENINF1_MUX_CTRL(0xa6df8080), SENINF1_MUX_INTSTA(0x19), SENINF1_MUX_DEBUG_2(0x7800438)

3.8.1 SMI繁忙

- 再看kernel部分有如下error 信息.

[4204.833211] -(6)[20120:ccu_worker][ISP][ISP_DumpDmaDeepDbg]
[4204.832993]CAM_A:raw_int_err:0x0_0x100,
raw_int3_err:0x0\x0a[4204.833144]CAM_A:raw_int_err:0x0_0x20000000,
raw_int3_err:0x0\x0a[4204.833144]camsys:0x3a[4204.833144]CAM_A:IMGO:0xffff0000,R
RZO:0xffff0000,AAO=0xffff0000,AFO=0xffff0000,LCSO=0xffff0000,UFEO=0xffff0000,PDO=0
xffff0000,PSO=0xffff0000\x0aEISO=0xffff0000,RSSO=0xffff0000,UFGO=0xffff0000,FLKO=0x
ffff0000 DMA_DBG_SEL=0x0
TOP_DBG_PORT=0x0\x0a[4204.833144](IMGO1:DMA_DBG_SEL=0x80403
DBG_PORT=0x40000)[4204.833144](IMGO2:DMA_DBG_SEL=0x403
DBG_PORT=0x488600c)[4204.8331

具体分析方法

Case study – mismatch/TG_ERR/WDMA_ERR (ALPS03663492)

[User log]

```
E ifunc_cam_buf:
[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_functio
n_cam.buf_ctrl.cpp, dequeueHwBuf, line1487] ERROR: [0x1]:BUF_CTRL_AFO:PA in header is
mismatch with PA in list [0x0_0xFE00000]
E ifunc_cam:
[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_functio
n_cam.pipechk.cpp, Check_Start, line0135] ERROR: [0x1]:accumulated err int_status:0x100010,
int3_status:0x0
E ifunc_cam:
[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_functio
n_cam.pipechk.cpp, PATH_CHECK, line0315] ERROR: [0x1]:current image pattern: 0
E ifunc_cam:
[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_functio
n_cam.pipechk.cpp, DMAO_STATUS, line0594] ERROR: [0x1]:[dmao status:0xe17d]_imgo_
ERR:0xffff82a0 _rrzo_ ERR:0xffff82a0 _aao_ ERR:0xffff8280 _lcso_ ERR:0xffff8280 _lmvo_
ERR:0xffff8280 _flko_ ERR:0xffff8280 _pso_ ERR:0xffff8280 _rawi_ ERR:0xffff0044
E ifunc_cam:
[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/isp_50/imageio/drv/cam/isp_functio
n_cam.pipechk.cpp, DMAO_STATUS, line0636] ERROR: [0x1]:[dmao fifo-F]
```

Case study – mismatch/TG_ERR/WDMA_ERR (ALPS03663492)

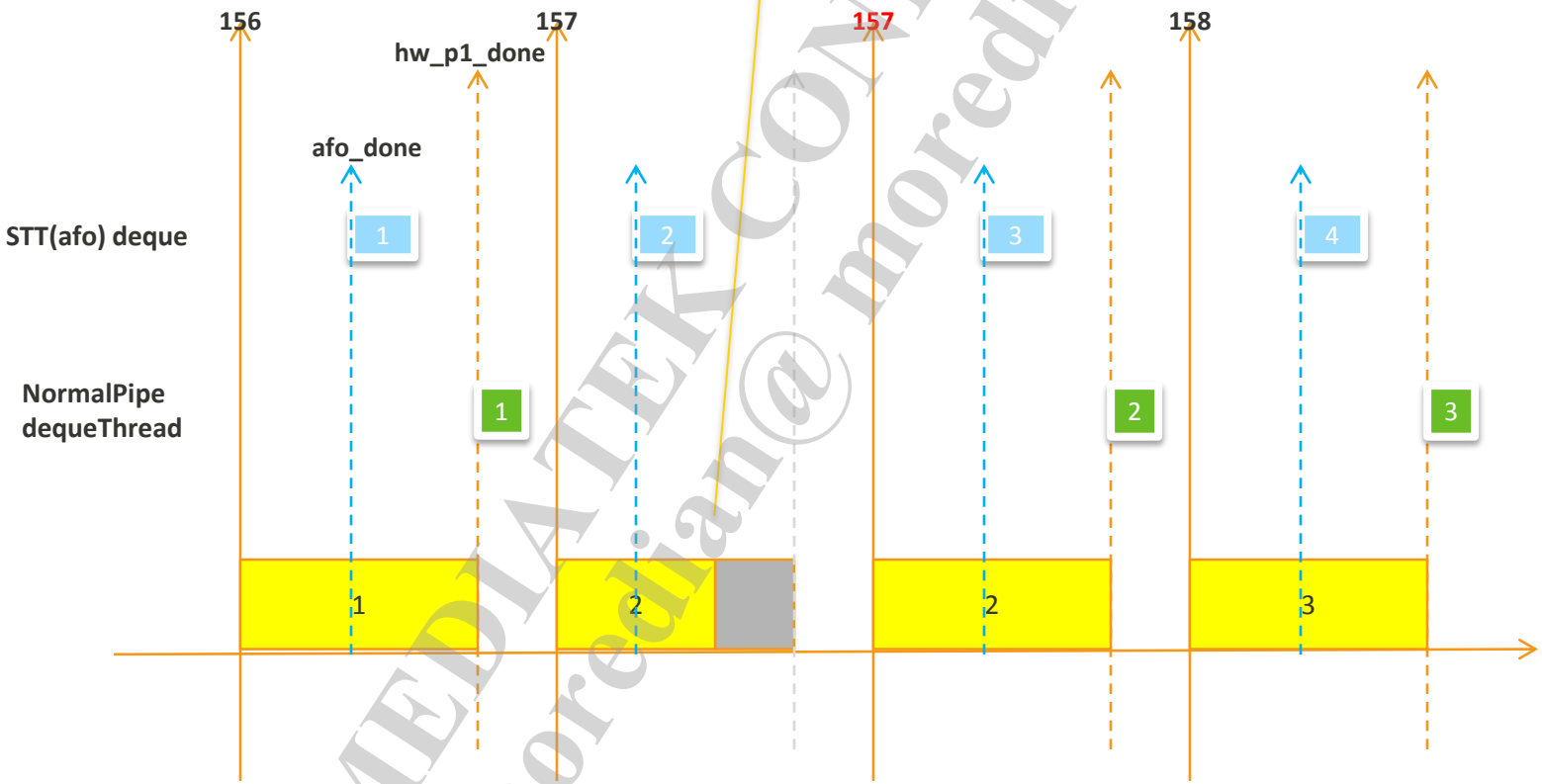
```
[Kernel log]
[  641.871494] (3)[3091:kworker/3:1]: [ISP][ISP_BH_Workqueue] [641.871345]CAMB
P1_SOF_157_157 (0x18000_0x21b1c01,0x18000_0x21b1c01,0x29d00000,0xba00000,0x9c),
int_us:50113,cq:0xe700000
[  641.921580] (3)[3091:kworker/3:1]: [ISP][ISP_BH_Workqueue]
[641.886161]CAM_B:raw_int_err:0x100000_0x10, raw_int3_err:0x0

[  641.921580][641.921441]CAM_B: raw_int_err:0x0_0x20000000, raw_int3_err:0x0
[  641.921580][641.921441]camsys:
0x3a[641.921441]CAM_B:IMGO:0xffff82a0,RRZO:0xffff82a0,AAO=0xffff8280,AFO=0xffff8
280, LCSO=0xffff8280,UFEO=0xffff0000,PDO=0xffff0000,PSO=0xffff8280
[  641.921580]: EISO=0xffff8280,RSSO=0xffff8280,UFGO=0xffff0000,FLKO=0xffff8280
[  641.921580][641.921441]CAM_B:
BPCI:0xffff0000,LSCI=0xffff0044,PDI=0xffff0000,RAWI=0xffff0000
[  641.921596] (3)[3091:kworker/3:1]: [ISP][ISP_BH_Workqueue] [641.921441]CAMB
P1_SOF_158_157 (0x18000_0x21b1c01,0x18000_0x21b1c01,0x29d00000,0xba00000,0x9c),
int_us:50111,cq:0xdd00000
```

Case study

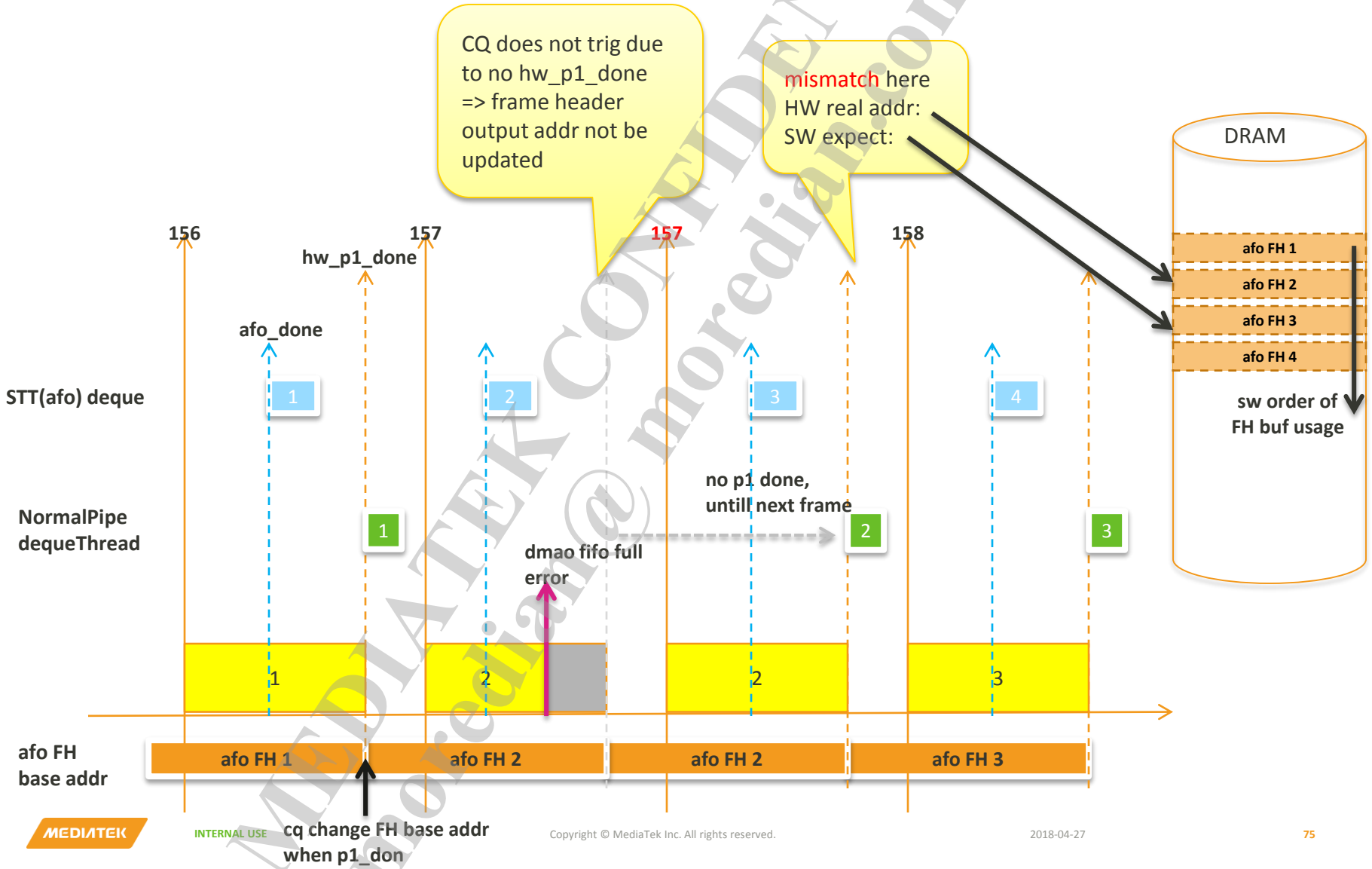
[641.871345]CAMB P1_SOF_157_157

[641.886161]CAM_B:raw_int_err:0x100000_0x10

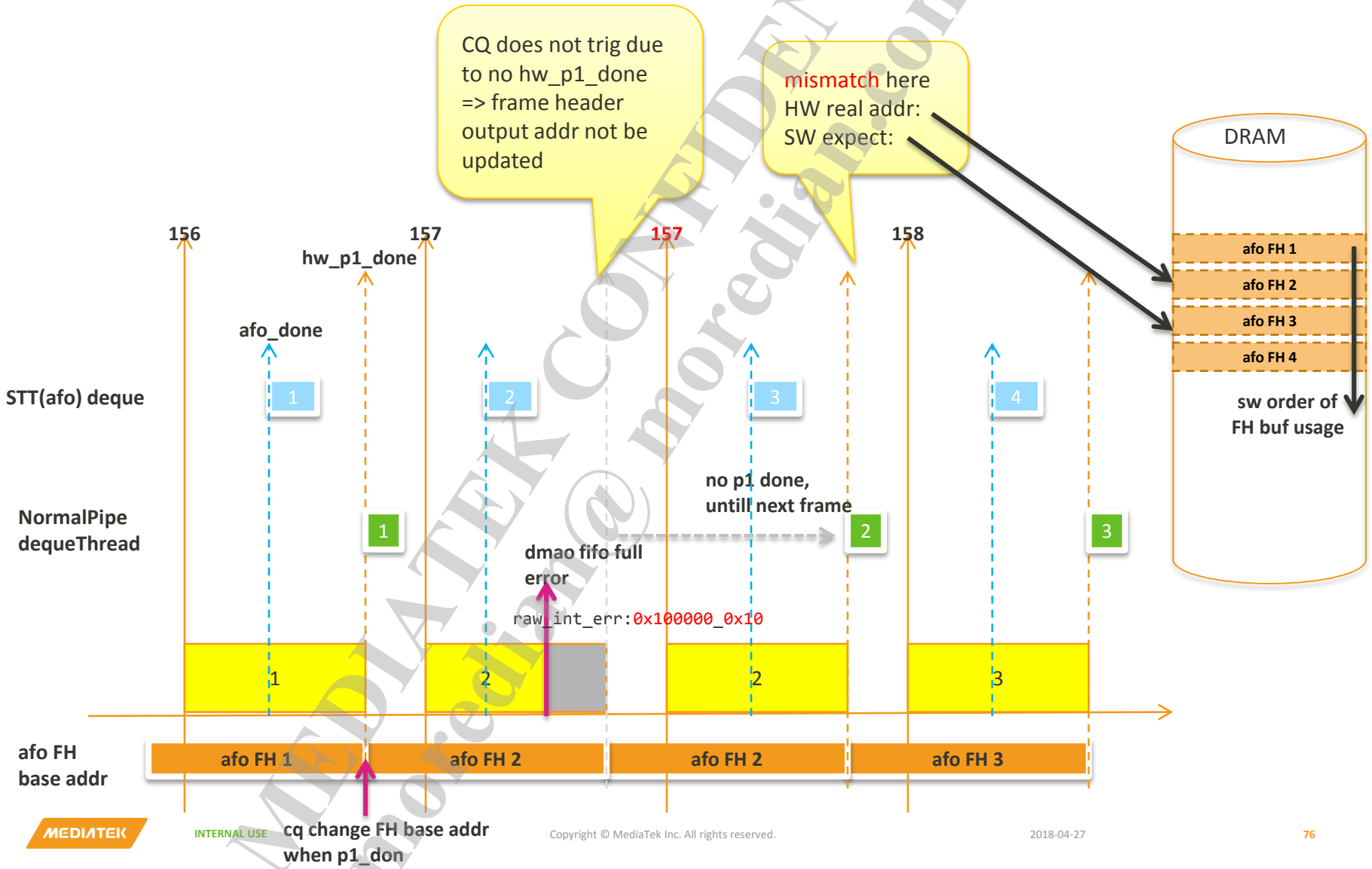
[641.921441]CAM_B:IMGO:0xffff82a0,RRZO:0xffff82a0,AAO=0xffff8280,AFO=0xffff8280,
[641.921441]CAMB P1_SOF_158_157



Case study



Case study



3.4 seninf clock设定bug

- 再看seninf debug 信息

02-23 18:25:07.146 4764 14423 D **SeninfDrv: [debug]f_fseninf_ck(312000)**
7225 02-23 18:25:05.141 4764 14423 D SeninfDrv: [debug]SENINF1_CSI2_INT_STATUS(0x0), CLR SENINF1_CSI2_INT_STATUS(0x0)
7230 02-23 18:25:05.144 4764 14423 D SeninfDrv: [debug]SENINF1_PkCnt(0x6c4), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x1)
7232 02-23 18:25:05.146 4764 14423 D SeninfDrv: [debug]SENINF1_PkCnt(0x6c4), SENINF2_PkCnt(0x1), SENINF3_PkCnt(0x1), SENINF5_PkCnt(0x1)
7234 02-23 18:25:05.147 4764 14423 D SeninfDrv: [debug]SENINF1_MUX_CTRL(0xa6df8080), SENINF1_MUX_INTSTA(0x18), SENINF1_MUX_DEBUG_2(0x91806c0)

问题点:
1. 2328 X 1728 是imx519 的preview setting / 正常.
2. SENINF1_CSI2_INT_STATUS reg 0
3. SENINF1_PkCnt(0x6c4) 两次没有递增.

最终发现seninf clk isp内部设定太大导致.最终office版本为:
SeninfDrv: [debug]f_fseninf_ck(208000)

4. 常见LOG 说明

4.1 previous not consumed 含义

每一楨enque前會先判斷software cq update count和physical cq update count,如果不一致,即打印” previous not consumed”,意即上一次enque的dmao未完成(cannot wait hw p1 done),如果大量出現則不合理,必須查一下為何會造成cannot wait hw p1 done.

1. Sensor 没吐数据.

4.1.1 sensor 不吐数据导致

- 首先看到这样的错误:

1139236 11-15 14:20:32.898294 512 17134 D NormalPipe: [ispEnqueThread] N:0: cur SOF: 105
1139237 11-15 14:20:32.898328 512 17134 D NormalPipe: [enqueRequest] +N:0 CurSof:105 enqCnt:19 cqCnt(p/v):18/19 FrameStatus(2)
1139238 11-15 14:20:32.898351 512 17134 W NormalPipe: [enqueRequest] WARNING: N:0 cqCnt phy:18 vir:19 reserve request, previous not consumed.

4.1.1 sensor 不吐数据导致

- 分析seninf debug 信息, 数据异常

从如下log看,是sensor imx386 数据异常.仅仅吐了0xfc0 X 0x37e的大小
359583 11-15 14:19:55.679644 512 14036 I NormalPipe: [configPipe] N:0 TG=1 Scen=5 tgFmt=x20d devID=1 eRawPxlID=3 PixelMode=0 W/H=[4032,3016] fps=24/24 Clk=1000 khz burst=1

1140236 11-15 14:20:33.850750 512 17156 D SeninfDrvImp: [sendCommand]SENINF1_MUX_CTRL 0x1A040120(0x9eff8080), SENINF1_MUX_INTSTA 0x1A040128(0x1a), SENINF1_MUX_DEBUG_2 0x1A040134(0xfc0037e),==> imx386 数据不足
1140237 11-15 14:20:33.850790 512 17156 D SeninfDrvImp: [sendCommand]SENINF2_MUX_CTRL 0x1A040520(0x9eff8080), SENINF2_MUX_INTSTA 0x1A040528(0x1a), SENINF2_MUX_DEBUG_2 0x1A040534(0x10700c30),
1140238 11-15 14:20:33.850832 512 17156 D SeninfDrvImp: [sendCommand]SENINF3_MUX_CTRL 0x1A040920(0x86df0080), SENINF3_MUX_INTSTA 0x1A040928(0x0), SENINF3_MUX_DEBUG_2 0x1A040934(0x0)
1140239 11-15 14:20:33.850865 512 17156 D SeninfDrvImp: [sendCommand]SENINF4_MUX_CTRL 0x1A040D20(0x9effa180), SENINF4_MUX_INTSTA 0x1A040D28(0x1a), SENINF4_MUX_DEBUG_2 0x1A040D34(0x13b00000)

4.2 关于isp fifo 的概念讨论

- 参考3.5节.

4.3 “INT_STATUS_”

- /kernel-4.4/drivers/misc/mediatek/cameraisp/src/isp_50/inc/cam_regs.h

```
33/* err status */
34#define TG_ERR_ST               (1L<<4)
35#define TG_GBERR_ST             (1L<<5)
36#define CQ_CODE_ERR_ST          (1L<<6)
37#define CQ_APB_ERR_ST           (1L<<7)
38#define CQ_VS_ERR_ST            (1L<<8)
39#define AMX_ERR_ST              (1L<<15)
40#define RMX_ERR_ST              (1L<<16)
41#define BMX_ERR_ST              (1L<<17)
42#define RRZO_ERR_ST             (1L<<18)
43#define AFO_ERR_ST              (1L<<19)
44#define IMGO_ERR_ST             (1L<<20)
45#define AAO_ERR_ST              (1L<<21)
46#define PSO_ERR_ST              (1L<<22)
47#define LCSO_ERR_ST             (1L<<23)
48#define BNR_ERR_ST              (1L<<24)
49#define LSC_ERR_ST              (1L<<25)
50#define UFGO_ERR_ST             (1L<<26)
51#define UFEO_ERR_ST             (1L<<27)
52#define PDO_ERR_ST              (1L<<28)
53#define DMA_ERR_ST              (1L<<29)
```

```
130/**
131 *    IRQ Error Mask
132 */
133#define INT_ST_MASK_CAM_ERR     (\
134         TG_ERR_ST |\
135         TG_GBERR_ST |\
136         CQ_CODE_ERR_ST |\
137         CQ_APB_ERR_ST |\
138         CQ_VS_ERR_ST |\
139         AMX_ERR_ST |\
140         RMX_ERR_ST |\
141         BMX_ERR_ST |\
142         BNR_ERR_ST |\
143         LSC_ERR_ST |\
144         DMA_ERR_ST)
145
146
```

检查tg fifo overrun@3.5

检查SMI优先级 @3.8.1,sensor数据量

检查Sensor数据@3.1-3.3

4.4.1 isp wait irq fail<mw异常>

- 复现时查看dumpsys meminfo发现手机cameraservice 达到了2.7 G，怀疑cameraservice存在内存泄漏，因为没有内存所以桌面一起来就会被lmk杀掉。mainlog 中出现了大量如下 error log，能否帮忙看下什么原因？

11-14 10:53:40.125 23523 29572 E Isp_Drv :
Isp_Drv[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/mt6797/isp/isp_drv.cpp, waitIrq, line0600] ERROR: ISP(0x0)_WAIT_IRQ fail(-1). Wait Status(0x00000080), Timeout(400).
11-14 10:53:40.125 23523 29572 E Isp_Drv :
(waitIrq){#600:vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/mt6797/isp/isp_drv.cpp}
11-14 10:53:40.305 23523 29574 E Isp_Drv :
Isp_Drv[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/mt6797/isp/isp_drv.cpp, waitIrq, line0600] ERROR: ISP(0x0)_WAIT_IRQ fail(-1). Wait Status(0x00000020), Timeout(400).
11-14 10:53:40.305 23523 29574 E Isp_Drv :
(waitIrq){#600:vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/mt6797/isp/isp_drv.cpp}
11-14 10:53:40.515 23523 29572 E Isp_Drv :
Isp_Drv[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/mt6797/isp/isp_drv.cpp, waitIrq, line0600] ERROR: ISP(0x0)_WAIT_IRQ fail(-1). Wait Status(0x00000080), Timeout(400).

4.4.2 isp wait irq fail<mw异常>

- 分析结论:原因是mw没有enque buffer进来.

11-19 21:14:22.683951 27227 18103 E ifunc_cam:
[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/mt6797/imageio/drv/cam/isp_function_cam.buf_ctrl.cpp, waitBufReady, line1946] ERROR: [0x0]:queue depth is empty for deque
11-19 21:14:22.683951 27227 18103 E ifunc_cam:
(waitBufReady){#1946:vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/mt6797/imageio/drv/cam/isp_function_cam.buf_ctrl.cpp}
11-19 21:14:22.684028 27227 18103 E sttiopipe:
[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/mt6797/imageio/pipe/StatisticIOPipe/StatisticIOPipe.cpp, dequeOutBuf, line0930] ERROR: [0x0]:waitBufReady fail
(dequeOutBuf){#930:vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/mt6797/imageio/pipe/StatisticIOPipe/StatisticIOPipe.cpp}
11-19 21:14:22.684072 27227 18103 E StatisticPipe:
[vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/mt6797/iopipe/CamIO/StatisticPipe.cpp, deque, line0986] ERROR: error:TG1 port_13 deque fail
(deque){#986:vendor/mediatek/proprietary/hardware/mtkcam/drv/src/isp/mt6797/iopipe/CamIO/StatisticPipe.cpp}
11-19 21:14:22.684115 27227 18103 E afo_buf_mgr: [dequeueHwBuf] m_pSttPipe deque fail
(dequeueHwBuf){#229:vendor/mediatek/proprietary/hardware/mtkcam/aaa/source/mt6797/buf_mgr/AFOBufMgr.cpp}

4.4.2 isp wait irq fail<sensor异常>

- 另外要注意的是有这样的错误也有可能是sensor 吐数据异常.
  具体参考第三章seninf 分析部分.

--

4.5 "no request sof_idx:"

- 经常帮忙协助分析帧率不够的log.” CAMA Lost p1 done_” =➔ 没有任何意义

从log看233-243丢6帧,lost P1 Done,
8096 01-13 07:12:13.629248 2396 2396 I [135.942460] (2)[2396:kworker/2:2][name:camera_isp&]: [ISP][ISP_BH_Workqueue] [135.942322]CAMA Lost p1 done_237 (0xed): [135.942322]CAMA P1_SOF_237_237(0x18000_0xe5090900,0x18000_0xe5090900,0x28200000,0x21f00000,0x110),int_us:41629,cq:0x23700000
8118 01-13 07:12:13.712629 2068 2068 I [136.025841] (6)[2068:kworker/6:2][name:camera_isp&]: [ISP][ISP_BH_Workqueue] [136.025616]CAMA Lost p1 done_239 (0xef): [136.025616]CAMA P1_SOF_239_239(0x18000_0xe60a0a00,0x18000_0xe60a0a00,0x2be00000,0x25700000,0x111),int_us:41641,cq:0x23b00000
8131 01-13 07:12:13.795848 2068 2068 I [136.109060] (6)[2068:kworker/6:2][name:camera_isp&]: [ISP][ISP_BH_Workqueue] [136.108921]CAMA Lost p1 done_241 (0xf1): [136.108921]CAMA P1_SOF_241_241(0x18000_0xe70b0b00,0x18000_0xe70b0b00,0x2af00000,0x25400000,0x112),int_us:41642,cq:0x23f00000
8153 01-13 07:12:13.879191 136 136 I [136.192403] (0)[136:kworker/0:1][name:camera_isp&]: [ISP][ISP_BH_Workqueue] [136.192232]CAMA Lost p1 done_243 (0xf3): [136.192232]CAMA P1_SOF_243_243(0x18000_0xe80c0c00,0x18000_0xe80c0c00,0xa500000,0x19800000,0x113),int_us:41669,cq:0x23700000

4.5 "no request sof_idx:"

- 要看pipe check 部分有如下error 信息.

2774 user enque thread 一直阻塞中

28121 01-13 07:11:53.003870 517 2774 I NormalPipe: [enqueRequest] N:0 dma:x8 no request sof_idx:1, FrameStatus(2)
28123 01-13 07:11:53.004099 517 2741 I NormalPipe: [enqueRequest] N:2 dma:x8 no request sof_idx:7, FrameStatus(1)
28174 01-13 07:11:53.045481 517 2774 I NormalPipe: [enqueRequest] N:0 dma:x8 no request sof_idx:2, FrameStatus(1)
28175 01-13 07:11:53.045771 517 2741 I NormalPipe: [enqueRequest] N:2 dma:x8 no request sof_idx:8, FrameStatus(1)
28259 01-13 07:11:53.087153 517 2774 I NormalPipe: [enqueRequest] N:0 dma:x8 no request sof_idx:3, FrameStatus(1)
28260 01-13 07:11:53.087621 517 2741 I NormalPipe: [enqueRequest] N:2 dma:x8 no request sof_idx:9, FrameStatus(1)
28464 01-13 07:11:53.128797 517 2774 I NormalPipe: [enqueRequest] N:0 dma:x8 no request sof_idx:4, FrameStatus(1)
28465 01-13 07:11:53.128797 517 2741 I NormalPipe: [enqueRequest] N:2 dma:x8 no request sof_idx:10, FrameStatus(1)
28519 01-13 07:11:53.170462 517 2774 I NormalPipe: [enqueRequest] N:0 dma:x8 no request sof_idx:5, FrameStatus(1)
28520 01-13 07:11:53.170462 517 2741 I NormalPipe: [enqueRequest] N:2 dma:x8 no request sof_idx:11, FrameStatus(1)
28539 01-13 07:11:53.212164 517 2774 I NormalPipe: [enqueRequest] N:0 dma:x8 no request sof_idx:6, FrameStatus(1)
28540 01-13 07:11:53.212175 517 2741 I NormalPipe: [enqueRequest] N:2 dma:x8 no request sof_idx:12, FrameStatus(1)

4.5 "no request sof_idx:"

- 好不容易才enque一个buffer给isp drv, 故帧率掉了

28883 01-13 07:11:53.360236 517 2691 I NormalPipe: [acceptEnqRequest] MyEnq:0: dma:x8
pa(0x18600000)io(0_0_3968_2976_1440_1080)M:x3, dma:x6
pa(0x7E00000)io(0_0_3968_2976_3968_2976)M:x3, dma:xf
pa(0x20A00200)io(0_0_256_1_256_1)M:x3, dma:xb pa(0x1DD00000)io(0_0_384_384_384_384)M:x3,
28893 01-13 07:11:53.362491 517 2694 I NormalPipe: [acceptEnqRequest] MyEnq:2: dma:x8
pa(0x9300000)io(0_0_1600_1200_1280_960)M:x3, dma:x6
pa(0x6F00000)io(0_0_1600_1200_1600_1200)M:x3, dma:xb
pa(0x6A00000)io(0_0_384_384_384_384)M:x3,
……
29185 01-13 07:11:53.454560 517 2691 I NormalPipe: [acceptEnqRequest] MyEnq:0: dma:x8
pa(0x19800000)io(0_0_3968_2976_1440_1080)M:x4, dma:x6
pa(0xA500000)io(0_0_3968_2976_3968_2976)M:x4, dma:xf
pa(0x20A00300)io(0_0_256_1_256_1)M:x4, dma:xb pa(0x1F000000)io(0_0_384_384_384_384)M:x4,
29269 01-13 07:11:53.503617 517 2774 I NormalPipe: [enqueRequest] N:0 dma:x8 no request
sof_idx:13, FrameStatus(2)
29270 01-13 07:11:53.503628 517 2741 I NormalPipe: [enqueRequest] N:2 dma:x8 no request
sof_idx:19, FrameStatus(2)

4.6 P1_SOF P1_DON Lost p1 done

- P1_SOF_XXX表示平台开始拿到第几个frame的sof信号
- P1_DON_XXX 表示平台已经拿到了第几个frame信号
- Lost p1 done_XXX 表示平台没有正常拿到第几个frame信号
- 若连续两帧之间没有Lost P1 done, 可以通过两帧之间的P1_SOF 或者 P1_DON时间间隔推算framerate.
  - 若这个时间不符合预期,会导致waitirq 时间久, 影响3a的处理.
- 从P1_SOF 到对应的P1_DON时间即为sensor readout时间.
- 从P1_DON 到P1_SOF 即为sensor VBlk 时间,根据平台不一样, 这个时间要求在500-1000ms 以上. 否则就sensor fifo overrun了.

[137.445032]CAMB P1_DON_128(0x0_0x0,0x18000_0xd32e2e00)
[137.455825]CAMB Lost p1 done_129 (0x81): [137.455825]CAMB
P1_SOF_129_129(0x0_0x0,0x18000_0xd42e2e00,0x0,0x1b300000,0xaf),int_us:33774,cq:0xdb00000
[137.489599]CAMB P1_SOF_130_130(0x0_0x0,0x18000_0xd42e2f01,0x0,0xca00000,0xb0),int_us:33774,cq:0xd300000
[137.512615]CAMB P1_DON_130(0x0_0x0,0x18000_0xd42f2f00)
[137.523371]CAMB Lost p1 done_131 (0x83): [137.523371]CAMB
P1_SOF_131_131(0x0_0x0,0x18000_0xd52f2f00,0x0,0xca00000,0xb0),int_us:33771,cq:0xd300000
[137.557147]CAMB Lost p1 done_132 (0x84): [137.557147]CAMB
P1_SOF_132_132(0x0_0x0,0x18000_0xd62f2f00,0x0,0xca00000,0xb0),int_us:33778,cq:0xd300000
[137.613897]CAMB P1_DON_133(0x0_0x0,0x18000_0xd6303000)
[137.624690]CAMB Lost p1 done_134 (0x86): [137.624690]CAMB
P1_SOF_134_134(0x0_0x0,0x18000_0xd7303000,0x0,0x9900000,0xb1),int_us:33771,cq:0xd700000

everyday genius