everyday genius

[Camera] CCU Frame-Sync Debug Guide

Version:	1.3
Release date:	2018-03-30

Document Revision History

Revision	Date	Author	Description
1.0	2018-03-28	Kuanfu Yeh	
1.1	2018-03-29	Kuanfu Yeh	Fix frame-length activation delay description
1.2	2018-03-29	Kuanfu Yeh	Fix mapping of CCU log to delay values description
1.3	2018-03-30	Kuanfu Yeh	Add frame-rate check section

Table of Contents

1 Introduction

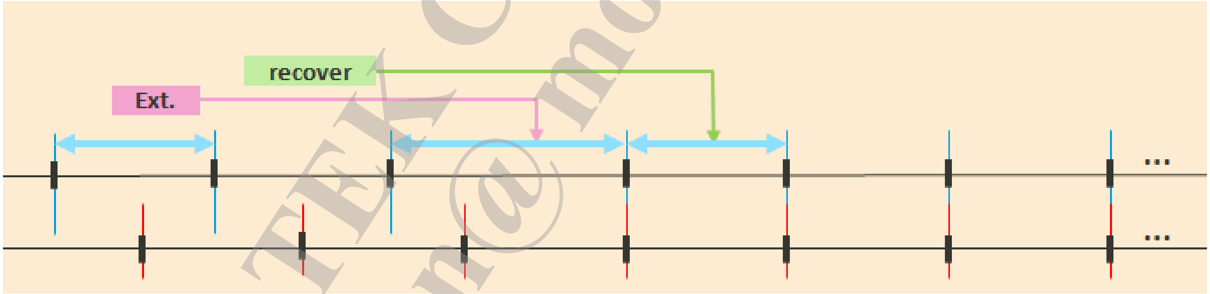
1.1 Overview & Scope

1.1.1 What is Frame-Sync

Image sensor works in frame-based behavior. By receiving SOF(Start-of-Frame) package via MIPI, ISP will generate Vsync signal to APMCU, CCU and related modules. Frame-length is defined as the time period between two consecutive Vsync signals. Frame-sync means make 2 sensor's frame-length and Vsync signal timing identical as possible under stereo scenario.

1.1.2 Purpose of Frame-Sync

Some of stereo camera feature needs the two image received from each side of sensors should be exposed in the very close timing to ensure the quality of feature effect, e.g. Refocus, Bokeh, AE-Sync. Which means that frame start/end timing of two sensors should be nearly the same. CCU will achieve this by calculating frame-length from shutter and fps of two sensors, and add compensation time onto frame-length of one of the two sensors if the two Vsync timing differs over required threshold.



2 System setup

2.1 Settings of sensor driver to make Frame-Sync works

2.1.1 imgsensor_info settings

The pclk, linelength, framelength and max_framerate arguments in sensor driver will affect sensor behavior in each scenario (preview, capture, video…).

```
static imgsensor_info_struct imgsensor_info = {
…
        .cap = {
                .pclk = 272000000,
                .linelength = 3580,
                .framelength = 2530,
                .startx = 0,
                .starty = 0,
                .grabwindow_width = 3264,
                .grabwindow_height = 2448,
                .mipi_data_lp2hs_settle_dc = 85,
                .max_framerate = 300,
        },
…
};
```

Frame-Sync will need these arguments to calculate proper frame length with current frame rate. The arguments must be precise enough to do the calculation; precision can be check with formula 2.1.

$$\left(\boldsymbol{linelenth} \times \boldsymbol{framelength} \times (\boldsymbol{max\_framerate} \div \mathbf{10})\right) - \boldsymbol{pclk} \leq \mathbf{1,000,000}$$

Formula 2.1 – precision constraint of imgsensor_info settings

Frame-Sync might be effective until the criterion of formula 2.1 is met. If the formula is invalid, ask sensor vendor for correct settings.

2.1.2 Frame-length activation delay

Activation delay defines as: the delay frame count for sensor to take effect after particular setting value is set to sensor. E.g. if frame-length takes effect in next frame after frame-length value is set to sensor, the “frame-length activation delay” is 1.

The activation delay of shutter, sensor-gain and frame-length might not be the same, and differs from sensor to sensor. Programmer for sensor driver porting should check these activation delay value with sensor vendor, and fill corresponding value into sensor driver. CCU also refers those arguments while setting exposure settings.

The code snippet below shows a corresponding configuration for a given sensor.

```
static imgsensor_info_struct imgsensor_info = {
…
            .ae_shut_delay_frame = 0,
            .ae_sensor_gain_delay_frame =0,
            .ae_ispGain_delay_frame = 2,
            .frame_time_delay_frame = 2,
…
};
```

A set of exposure settings is calculated by AE algorithm, including shutter, sensor-gain, ISP-gain and frame-length. These four arguments should take effect at the same frame to get correctly exposed image. However, the activation delay frame counts might be different for each argument, hence there are four arguments in imgsensor_info structure in sensor driver that described the timing. For the arguments: ae_shut_delay_frame, ae_sensor_gain_delay_frame and ae_ispGain_delay_frame, please refer to sensor porting guide.

frame_time_delay_frame is used by CCU Frame-Sync only, if it was wrong, Frame-Sync might not success, it can be computed with formula 2.2. In this case, it should be “1 – 1 + 2”, turns out 2.

$$\boldsymbol{frame\_time\_delay\_frame} \ = \boldsymbol{FLAD} - \boldsymbol{SAD} + \boldsymbol{ae\_ispGain\_delay\_frame}$$

Formula 2.2 – Computation of argument “frame_time_delay_frame”

- **FLAD (frame-length-activation-delay):**
  In this case, frame-length takes effect 1 frame after which frame-length value is being set, so frame-length-activation-delay is 1.
- **SAD (shutter-activation-delay):**
  In this case, shutter takes effect 1 frame after which shutter value is being set, so shutter-activation-delay is 1.

2.2 Building the CCU sensor driver binary

Since CCU sensor driver is extracted on build time from original sensor driver source code in kernel, if any modification to sensor driver was made, **CCU sensor driver intermediates folder should be deleted** before re-build, and instead of building kernel only, a **full-build is must**.

- **CCU sensor driver intermediate path** (the parts with italic font and highlight color should be replaced with your actual project name and sensor name)
  "out\target\product\*<project_name>*\obj_arm\EXECUTABLES\libccu_*<sensor_name>*.ddr_intermediates"

For further detail of CCU sensor driver building, please refer to “Build CCU sensor driver” section of document “CCU_sensor_connecition_guide (ISP5.0).pptx”

3 Debug Guide

3.1 Log analysis

3.1.1 Keywords

- **fs_curdiff (F**rame **S**ync **CUR**rent **DIFF**erence)

  e.g. “43084 02-01 20:17:43.786188   553 5940 I CcuDrv : 1706887477: fs_curdiff(33316)”

  - the value means time difference of last 2 Vsync, in unit of micros second(us), the 33316 in above log presents 33316 us = 33.316 ms

- **Vsync_B_cnt, Vsync_A_cnt**

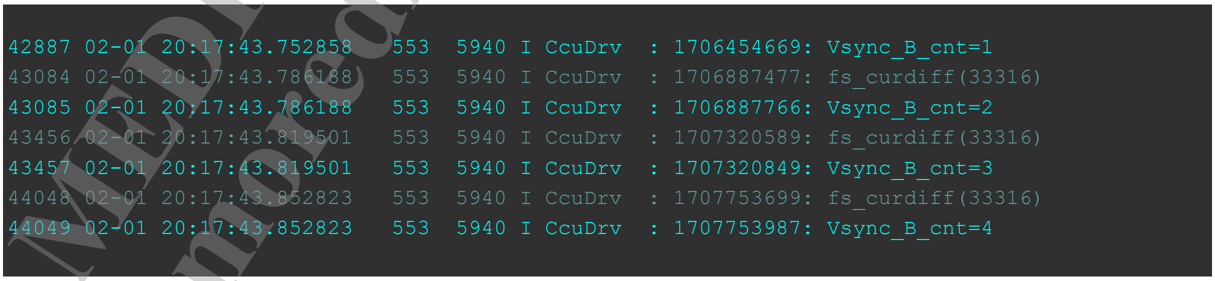
  e.g. “42887 02-01 20:17:43.752858   553 5940 I CcuDrv : 1706454669: Vsync_B_cnt=1”

  - This log prints when Vsync interrupt received by CCU
  - Note: timestamp of CCU log show in unit of clock tick of 13Mhz timer, converting it to ms can be done by divide it by 13000, for example 1706454669 = 131265 ms since CCU booted, it can be used to compare with other CCU logs to calculate time difference, but it cannot match the timestamp of android log.

3.1.2 Log pattern

This section shows how to read CCU Frame-Sync related logs. These logs are default on, no need to apply special command to activate. Following analysis starts from the timing of entering stereo mode.

- In stereo scenario, two sensors will be configured and turned streaming on in serial order, not simultaneously.
- In this case of log, CAM-B (TG-2) is opened first, so that only Vsync_B log is printed. fs_curdiff shows 33316 (33ms) means the time difference between current Vsync and previous Vsync, that will be equal to the frame length of CAM-B for now because only single sensor is turned on.



```
42887 02-01 20:17:43.752858   553  5940 I CcuDrv  : 1706454669: Vsync_B_cnt=1
43084 02-01 20:17:43.786188   553  5940 I CcuDrv  : 1706887477: fs_curdiff(33316)
43085 02-01 20:17:43.786188   553  5940 I CcuDrv  : 1706887766: Vsync_B_cnt=2
43456 02-01 20:17:43.819501   553  5940 I CcuDrv  : 1707320589: fs_curdiff(33316)
43457 02-01 20:17:43.819501   553  5940 I CcuDrv  : 1707320849: Vsync_B_cnt=3
44048 02-01 20:17:43.852823   553  5940 I CcuDrv  : 1707753699: fs_curdiff(33316)
44049 02-01 20:17:43.852823   553  5940 I CcuDrv  : 1707753987: Vsync_B_cnt=4
```

After another sensor be turned on, we can see both Vsync_A and Vsync_B logs are printed, and value of fs_curdiff changes.

- The value of fs_curdiff presents the time difference between Vsync_A and Vsync_B, since Frame-Sync is not started yet, the value is about 6000 (6 ms) now, larger than 500 (0.5ms).
- The value of fs_curdiff will be random every time entering stereo scenario, it’s up to the time difference of two sensors being turned on.

```
44985 02-01 20:17:43.913620   553  5940 I CcuDrv  : 1708539176: Vsync_A_cnt=1
45071 02-01 20:17:43.920460   553  5940 I CcuDrv  : 1708620034: fs_curdiff(6234)
45073 02-01 20:17:43.920460   553  5940 I CcuDrv  : 1708621048: Vsync_B_cnt=6
45479 02-01 20:17:43.946654   553  5940 I CcuDrv  : 1708972141: fs_curdiff(6234)
45480 02-01 20:17:43.946654   553  5940 I CcuDrv  : 1708972429: Vsync_A_cnt=2
45551 02-01 20:17:43.953651   553  5940 I CcuDrv  : 1709059179: fs_curdiff(6688)
45553 02-01 20:17:43.953651   553  5940 I CcuDrv  : 1709060155: Vsync_B_cnt=7
```

When HAL3A receives frame-sync request from middleware, it will then send a command to CCU to make Frame-Sync start, corresponding log is “`sync3a   :  [updateFrameSync]  CCU_SYNC_DBG MSG_TO_CCU_START_FRAME_SYNC`”, if there’s no start frame-sync log found, frame-sync will not start.

After few frames, the frame-sync procedure is done, and value of fs_curdiff shrinks to no more than 500 (0.5 ms), it means the time difference between Vsync_A and Vsync_B is smaller than 0.5 ms.

```
47058 02-01 20:17:44.068170   553  3979 D sync3a  : [updateFrameSync] CCU_SYNC_DBG MSG_TO_CCU_START_FRAME_SYNC
…
50068 02-01 20:17:44.257405   553  5940 I CcuDrv  : 1713011629: fs_curdiff(447)
50069 02-01 20:17:44.257405   553  5940 I CcuDrv  : 1713011947: Vsync_B_cnt=16
50076 02-01 20:17:44.257762   553  5940 I CcuDrv  : 1713017373: fs_curdiff(441)
50077 02-01 20:17:44.257762   553  5940 I CcuDrv  : 1713017690: Vsync_A_cnt=11
50608 02-01 20:17:44.291466   553  5940 I CcuDrv  : 1713450798: fs_curdiff(441)
50609 02-01 20:17:44.291466   553  5940 I CcuDrv  : 1713451116: Vsync_B_cnt=17
50631 02-01 20:17:44.293376   553  5940 I CcuDrv  : 1713456470: fs_curdiff(435)
50632 02-01 20:17:44.293376   553  5940 I CcuDrv  : 1713457404: Vsync_A_cnt=12
```

Upon exiting stereo mode or camera closing, same as starting stereo preview, two sensors will be turned off in serial order, not simultaneously. In this case, CAM-A is turned off first, so that only Vsync_B log is left. HAL3A then will request CCU to stop frame-sync by command, corresponding log is ”`sync3a   : [updateFrameSync] CCU_SYNC_DBG MSG_TO_CCU_STOP_FRAME_SYNC`”.

```
112767 02-01 20:17:48.310381   553  5940 I CcuDrv  : 1765701572: fs_curdiff(33297)
112768 02-01 20:17:48.310381   553  5940 I CcuDrv  : 1765701978: Vsync_B_cnt=136
113343 02-01 20:17:48.344126   553  5940 I CcuDrv  : 1766140358: fs_curdiff(33750)
113344 02-01 20:17:48.344126   553  5940 I CcuDrv  : 1766140789: Vsync_B_cnt=137
113808 02-01 20:17:48.377865   553  5940 I CcuDrv  : 1766579104: fs_curdiff(33750)
113809 02-01 20:17:48.377865   553  5940 I CcuDrv  : 1766579538: Vsync_B_cnt=138
…
120620 02-01 20:17:48.820227   553  5989 D sync3a  : [updateFrameSync] CCU_SYNC_DBG MSG_TO_CCU_STOP_FRAME_SYNC
```

3.2 Further log analysis through detailed log

3.2.1 ADB commands for enable CCU detailed log

With following ADB commands applied, CCU log level will be set to 5, and more detailed logs will be printed out to mobile log (main log)

```
1. adb root
2. adb shell setprop debug.drv.ccu_drv 5
3. adb shell setprop debug.ccuif.ccu_drv 5
4. adb shell getprop debug.ccuif.ccu_drv
       •  Check if above instruction outputs 5, if not, re-do from step 1.
5. adb shell getprop debug.ccuif.ccu_drv
       •  Check if above instruction outputs 5, if not, re-do from step 1.
```

3.3 Check list for trouble-shooting

This section shows a list of known mistake that will cause Frame-Sync fail, check these first if Frame-Sync doesn't working well.

3.3.1 Check if both sensor control on CCU were initialized

CCU Frame-Sync need sensor related arguments to work, thus if only both sensor control on CCU is initialized correctly, Frame-Sync can start to work. Check if the following log shows, and make sure both tg(1) and tg(2) sensor_init_done present in log. Otherwise, check the reason why sensor control initialization cannot be done, or CCU sensor initialization just hasn't been called, then should ask AeMgr & HAL3A owner to check it.

Example for TG-1 sensor control initialization:

```
// this  log means AeMgr called CCU driver to do the initialization (only shows when log level is 5)
AbsCcuCtrlBase: [ccuControl] cotnrol msgId: 1, msg: MSG_TO_CCU_SENSOR_INIT

// following  log means CCU sensor control initialization is done
CcuDrv  : 2109329534: sensor_init, tg(1), fps(300), scenario(1)
CcuDrv  : 2109329870: _sensor_init_done, tg(1), type(2), typeo(2)
```

Example for TG-2 sensor control initialization:

```
// this  log means AeMgr called CCU driver to do the initialization (only shows when log level is 5)
AbsCcuCtrlBase: [ccuControl] cotnrol msgId: 1, msg: MSG_TO_CCU_SENSOR_INIT

// following  log means CCU sensor control initialization is done
CcuDrv  : 2109329534: sensor_init, tg(2), fps(300), scenario(1)
CcuDrv  : 2109329870: _sensor_init_done, tg(2), type(1), typeo(1)
```

3.3.2 Check if frame-sync start has been sent to CCU

HAL3A then will request CCU to start frame-sync by command, corresponding log is ”`sync3a : [updateFrameSync] CCU_SYNC_DBG MSG_TO_CCU_START_FRAME_SYNC`”. If the frame-sync start log is not found, ask HAL3A owner for further analysis.

3.3.3 Check if CCU using the correct delay values

This log is only printed in detailed log, refer to “5.2.1 ADB commands for enable CCU detailed log” to turn detailed log on. With detailed log on, sensor initialization log will printed as following snippet.

```
CcuDrv  : 2107436967: sensor_init, tg(2), fps(300), scenario(1)
CcuDrv  : 2107437066: <I2C-Id=0x00000002
CcuDrv  : 2107437084: <Init-
CcuDrv  : 2107437152: exp_delay_ofst=0x22120011
CcuDrv  : 2107437185: u8SupportedByCCU=0x00000001
CcuDrv  : 2107437233: fs_sinfo, pclk=0x39387000
CcuDrv  : 2107437262: fs_sinfo, linelength=0x000027B0
CcuDrv  : 2107437287: fs_sinfo, min_fl=0x00000C4C
CcuDrv  : 2107437315: fs_sinfo, fl_active_delay=0x00000001
CcuDrv  : 2107437351: _sensor_init_done, tg(2), type(1), typeo(1)
```

Check following items:

- **exp_delay_ofst**: The first, second, and fourth digits shows exposure settings delay timing (described in 4.2.2 Shutter and frame-length activation delay) that CCU get. The three delay value can be mapped in order of shutter delay, sensor-gain delay, frame-length delay, see figure 5.1 for demonstration.

    - First digit must equal to “**ae_ispGain_delay_frame - ae_sensor_gain_delay_frame**”
    - Second digit must equal to “**ae_ispGain_delay_frame - ae_shut_delay_frame**”
    - Fourth digit must equal to “**frame_time_delay_frame**”

  If constraints above don’t meet, re-build CCU sensor binary with the instruction in “4.3 Building the binary”

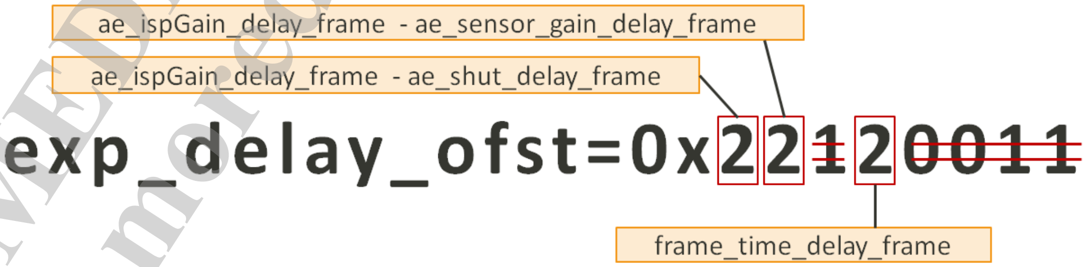

ae_ispGain_delay_frame - ae_sensor_gain_delay_frame
ae_ispGain_delay_frame - ae_shut_delay_frame
exp_delay_ofst=0x22120011
frame_time_delay_frame

Figure 5.1 – mapping of CCU log to delay values

- **fs_sinfo, pclk:** Must equal to “**pclk**” field in corresponding scenario structure of sensor driver (see: 4.2.1 imgsensor_info settings), If not, re-build CCU sensor binary with the instruction in “4.3 Building the binary”

- **fs_sinfo, linelength:** Must equal to “**linelength**” field in corresponding scenario structure of sensor driver (see: 4.2.1 imgsensor_info settings), If not, re-build CCU sensor binary with the instruction in “4.3 Building the binary”

- **fs_sinfo, min_fl:** Must equal to “**framelength**” field in corresponding scenario structure of sensor driver (see: 4.2.1 imgsensor_info settings), If not, re-build CCU sensor binary with the instruction in “4.3 Building the binary”

3.3.4 Check if both sensor have same max frame-rate

Frame-rate (FPS) of two sensors should be identical while Frame-Sync starting, to make sure Frame-Sync works well. At high light environment, both sensor will work at max frame-rate, so both sensor need have same max frame-rate, Max FPS can be checked by log. Following log snippet show a case of fail due to FPS not identical to each other.

```
// The initial FPS for tg-2(Cam-B) is 300 (30 fps)
CcuDrv  : 361933932: sensor_init, tg(2), fps(300), scenario(1)
CcuDrv  : 361934330: _sensor_init_done, tg(2), type(2), typeo(2)
// AE Mgr sends a command to assign FPS of tg-2(Cam-B) to 300 (30 fps)
CcuDrv  : 366178453: $poptask, tg=2, id=8, qn=0
CcuDrv  : 366178512: _set_mfps+
CcuDrv  : 366178921: max_framerate=0x0000012C
CcuDrv  : 366178944: _set_mfps-
  ...
// The initial FPS for tg-1(Cam-A) is 300 (30 fps)
CcuDrv  : 366357169: sensor_init, tg(1), fps(300), scenario(0)
CcuDrv  : 366357482: _sensor_init_done, tg(1), type(1), typeo(1)
...
// AE Mgr sends a command to assign FPS of tg-1(Cam-A) to 200 (20 fps)
CcuDrv  : 368088583: $poptask, tg=1, id=8, qn=0
CcuDrv  : 368088612: _set_mfps+
CcuDrv  : 368088666: max_framerate=0x00000C8
CcuDrv  : 368088689: _set_mfps-

// Here frame-sync start command is sent to CCU
// At this time, FPS of TG-1 is 20, and FPS of TG-2 is 30, that violates, thus frame-sync might fail
sync3a  : [updateFrameSync] CCU_SYNC_DBG MSG_TO_CCU_START_FRAME_SYNC
```