MEDIATEK

Raw-HDR algorithm porting guide

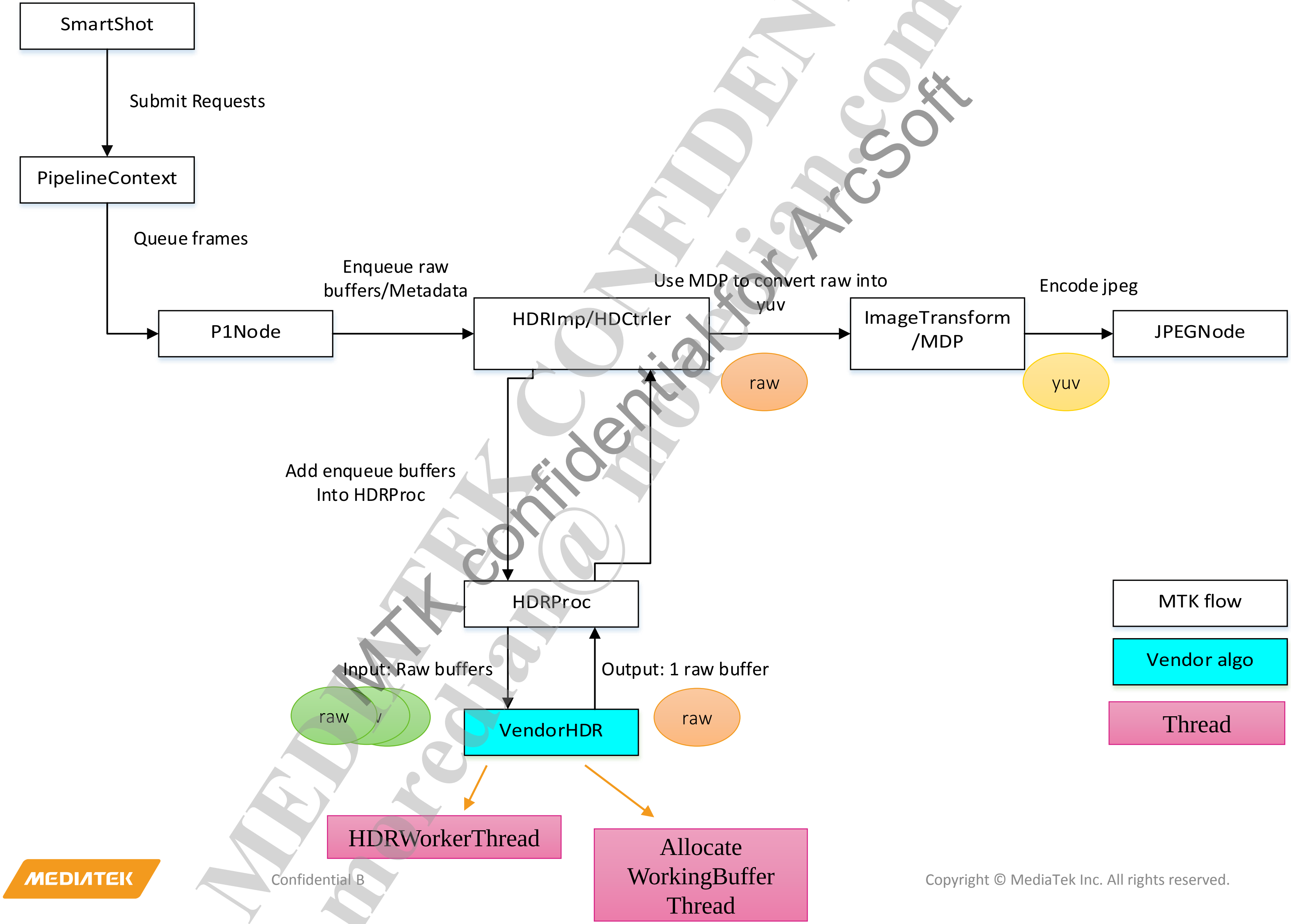
2017/07/07

Confidential B

Outline

- Raw HDR flow control
- Interface introduction
- Methods & members introduction
- Related folders

Raw HDR flow control



SmartShot

Submit Requests

PipelineContext

Queue frames

P1Node

Enqueue raw buffers/Metadata

HDRImp/HDCtrler

Use MDP to convert raw into yuv

raw

ImageTransform /MDP

Encode jpeg

yuv

JPEGNode

Add enqueue buffers Into HDRProc

HDRProc

Input: Raw buffers

raw

Output: 1 raw buffer

raw

VendorHDR

HDRWorkerThread

Allocate WorkingBuffer Thread

MTK flow

Vendor algo

Thread

Interface introduction (MTK)

- **Init/uninit**

  You can do some initialization & un-initialization in these methods.

- **getCaptureInfo**

  We get our exposure setting for each HDR frames.

- **addInputFrame**

  Add input buffers which HDR algorithm needed.

- **Process**

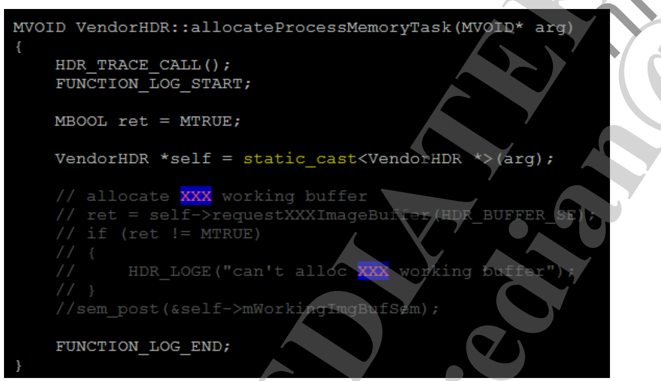
  Create creates a detached worker thread that processes the post-processing task

- **notify**

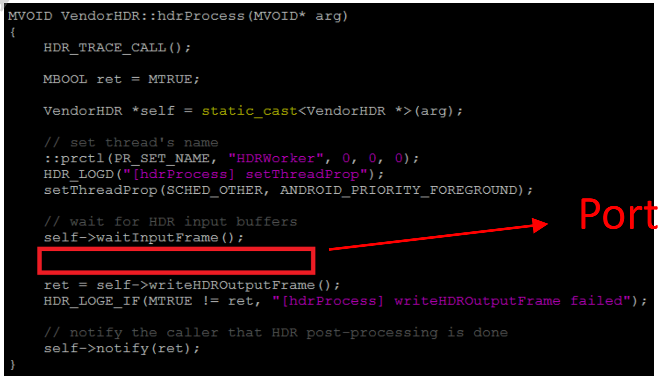
  Indicate post-processing done.

Methods introduction (Vendor)

- **allocateProcessMemoryTask**
  - Vendor can allocate working buffer here. This method executes asynchronously so don’t worry it would block main thread.
- **hdrProcess**
  - Vendor can put post-processing part here. It also a detached thread too.



```
MVOID VendorHDR::allocateProcessMemoryTask(MVOID* arg)
{
    HDR_TRACE_CALL();
    FUNCTION_LOG_START;

    MBOOL ret = MTRUE;

    VendorHDR *self = static_cast<VendorHDR *>(arg);

    // allocate XXX working buffer
    // ret = self->requestXXXImageBuffer(HDR_BUFFER_SE);
    // if (ret != MTRUE)
    // {
    //      HDR_LOGE("can't alloc XXX working buffer");
    // }
    //sem_post(&self->mWorkingImgBufSem);

    FUNCTION_LOG_END;
}
```

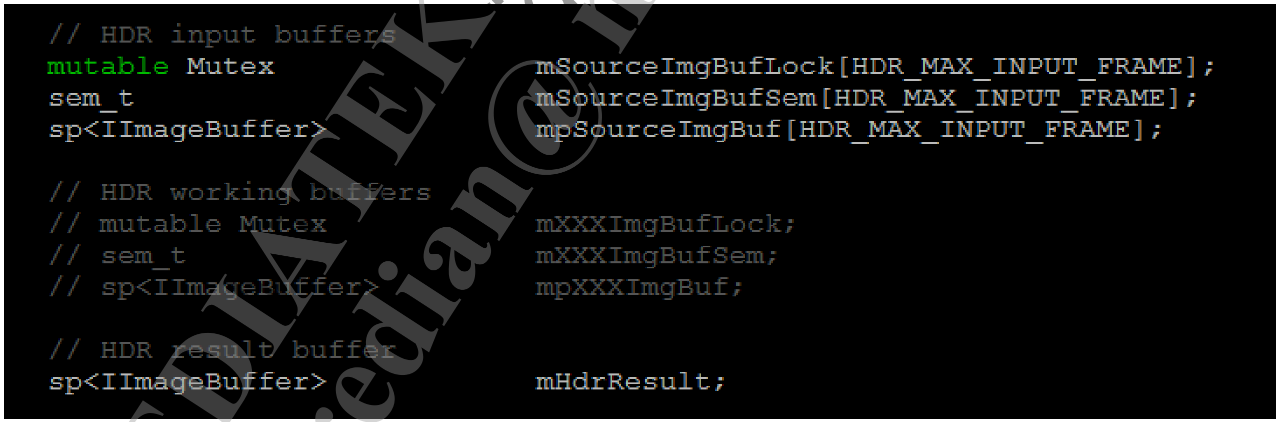


```
MVOID VendorHDR::hdrProcess(MVOID* arg)
{
    HDR_TRACE_CALL();

    MBOOL ret = MTRUE;

    VendorHDR *self = static_cast<VendorHDR *>(arg);

    // set thread's name
    ::prctl(PR_SET_NAME, "HDRWorker", 0, 0, 0);
    HDR_LOGD("[hdrProcess] setThreadProp");
    setThreadProp(SCHED_OTHER, ANDROID_PRIORITY_FOREGROUND);

    // wait for HDR input buffers
    self->waitInputFrame();


    ret = self->writeHDROutputFrame();
    HDR_LOGE_IF(MTRUE != ret, "[hdrProcess] writeHDROutputFrame failed");

    // notify the caller that HDR post-processing is done
    self->notify(ret);
}
```

Porting here

Methods & members introduction (Vendor)

- **writeHDROutputFrame**
  - Write back your result at class member (mHdrResult). HDRVendor will callback mHDRResult back to middleware.
- **mpSourceImgBuf**
  - Input buffers



```
// HDR input buffers
mutable Mutex                 mSourceImgBufLock[HDR_MAX_INPUT_FRAME];
sem_t                         mSourceImgBufSem[HDR_MAX_INPUT_FRAME];
sp<IImageBuffer>              mpSourceImgBuf[HDR_MAX_INPUT_FRAME];

// HDR working buffers
// mutable Mutex              mXXXImgBufLock;
// sem_t                      mXXXImgBufSem;
// sp<IImageBuffer>           mpXXXImgBuf;

// HDR result buffer
sp<IImageBuffer>              mHdrResult;
```

Related folder

- **VendorHDR**
  - vendor/mediatek/proprietary/hardware/mtkcam/feature/hdr/2.3/vendor_algo
  - You can add your methods in VendorHDR header
- **Image buffer**
  - include/mtkcam/utils/imgbuf/IImageBuffer.h
  - You can get color format, strides, resolution by IImageBuffer methods

MEDIATEK

everyday genius