Bringing Stateless Video Decoder Support To Linux

Nicolas Dufresne
Principal Software Engineer
About me

- Over 11 years at Collabora
- Core GStreamer developer
- Contribute to Linux Media
The beginning of Linux CODECs

- Google partners with Samsung and Asus
- Produce the first ARM Chromebook
- Based on Exynos 5 SoC
- Includes Samsung MFC Decoder
- MFC V4L2 M2M driver landed mainline

2011
CODA Driver

- CODA driver was added
  - Design from Chips&Media
  - CODA Hx4 and 960 support
- Enabling i.MX51 and i.MX6
- Reversed engineering
State-Full

Processor

ACC

ACC

ACC
A V4L2 output queue is used for the bitstream
A V4L2 capture queue is used for the decoded pictures
Additional control flow are added to support draining, flushing
Inter-queue configuration flow is needed
Pros

- Minimal per CODEC code needed

Cons

- Requires a firmware
- Harder to multiplex
The beginning of Linux State-Less CODECs

- Google partners with Rockchip
- 2nd gen of ARM Chromebook
- New type of CODEC hardware
- Rockchip VDPU?
State-Less

Reference(s)
Bitstream
Parameters

Picture

ACC
ACC
ACC
V4L2 M2M + Request

- Per-frame (or slice) controls are associated with bitstream buffers using requests
- References are signaled using user defined timestamps
- Better identification can be done using the Media Controller topology
V4L2 Specific Process

- Allocate a Request (an FD)
- Set per-frame/slice params for this request
- Queue a v4l2_buffer for the request
- Queue the request
- Poll the request FD for completion
MediaTek VPU

- State-full H.264, VP8 and VP9 decode
- Tiled output only (requires HW converter)
Qualcomm Venus

- State-full MPEG4, MPEG2, H.264, VC1, H.264, VP8, VP9 and HEVC decode
- MPEG4, H.263, H.264, VP8, HEVC

2017
Upstreaming Stalled

- Could not settle on the Request / Job API
- Low knowledge of CODEC decoding process by the linux-media maintainers
- Only one hardware to test the API design
- No formal specification (not that state-full CODEC had any either)
Allwinner VPU support
Kickstarter by Bootlin

- Request API is finalized
- MPEG2 Support landed in Staging
- H.264 support was progressing (but only sliced based)
- Reversed engineered from binary userspace blob
- VA-API userspace drivers
The crowd funding had gain good momentum

A formal specification was merged

H264, VP8, HEVC uAPI added as staging control API

Cedrus gained H264 and HEVC support

RK3288 driver was mainlined (MPEG2, H264, VP8)

RK3288 driver was renamed Hantro!
RK3399 JPEG, MPEG2, H.264 support landed

GStreamer gained base classes for state-less CODEC, with already merge DXVA2 and NVDEC support

GStreamer H.264 and VP8 V4L2 support landed

Embedded World Conference 2020 canceled

VA V4L2 Request driver was abandoned
H.264 uAPI stable in 5.11
VP8 uAPI stable in 5.13
HEVC uAPI is in bad shape
VP9, MPGE2 and HEVC uAPI being worked on
Starting to use conformance test (with fluster*)
Encoder support still under discussion

* https://github.com/fluendo/fluster