Smooth playback of adaptive video streams on Raspberry Pi with gst-mmal

John Sadler,

Krzysztof Konopko,

Tomasz Szkutkowski

YouView TV Ltd.

youview®

What were we trying to achieve?

The usual things really:

- Efficient handling of H.264 & MPEG-2 video
- Judder-free playback
- Proper A/V sync
- Smooth representation changes on adaptive streams
- Decent quality deinterlacing
- Efficient scaling and positioning of video window
- Fancy GL transformations *not* a priority
- Wanted to avoid ending-up with a single “blob” element

STB-like use cases on Raspberry Pi (2) “toy” platform

youview

Did we try gst-omx?

Yes, but we had some issues:

- Performance with *glimagesink* was not good (gst 1.6.0). Dropped frames and stuttering with 1080p streams
- Works much better with *eglglessink* from gst 1.2 - after patching for y-flip
- But, hangs-up on representation-change when playing adaptive streams
- Missing accelerated deinterlacing (s/w deinterlace not fast enough)

For more on gst-omx, see [1]

youview

So, we tried extending gst-omx...

- Tried hacking a new *omxvideosink* element
- Worked quite well for non-adaptive streams, but still some issues:
  - Splitting elements awkward due to OMX tunnelling
  - Dynamic reconfiguration very tricky to get right due to OMX state machine
  - Some issues with A/V sync & clocking
- Problems are really with OMX itself, more than gst-omx
- After *much* fiddling, we reluctantly gave up on OMX (for video)
- Then a question on RPi forums [2] pointed us to MMAL...

youview

So what's this MMAL thing?

- Multi-Media Abstraction Layer from Broadcom
- Host-side interface to VideoCore multimedia components
- Based on the concept of components, ports and buffer headers
  - Component -> Element, Port -> Pad
- Supports tunnelled configuration like OMX
- **Also supports "opaque buffers"** via userspace "buffer-headers"
  - Buffer headers are allocated by upstream elements & refcounted
  - Headers point to the "real" buffers in VC memory
  - GST elements pass buffers between MMAL components by passing headers
  - Makes it easier to split-out elements - no need to fake data-flow (like in tunneled case)
- For more on MMAL, see online docs [3]

youview

Current Elements

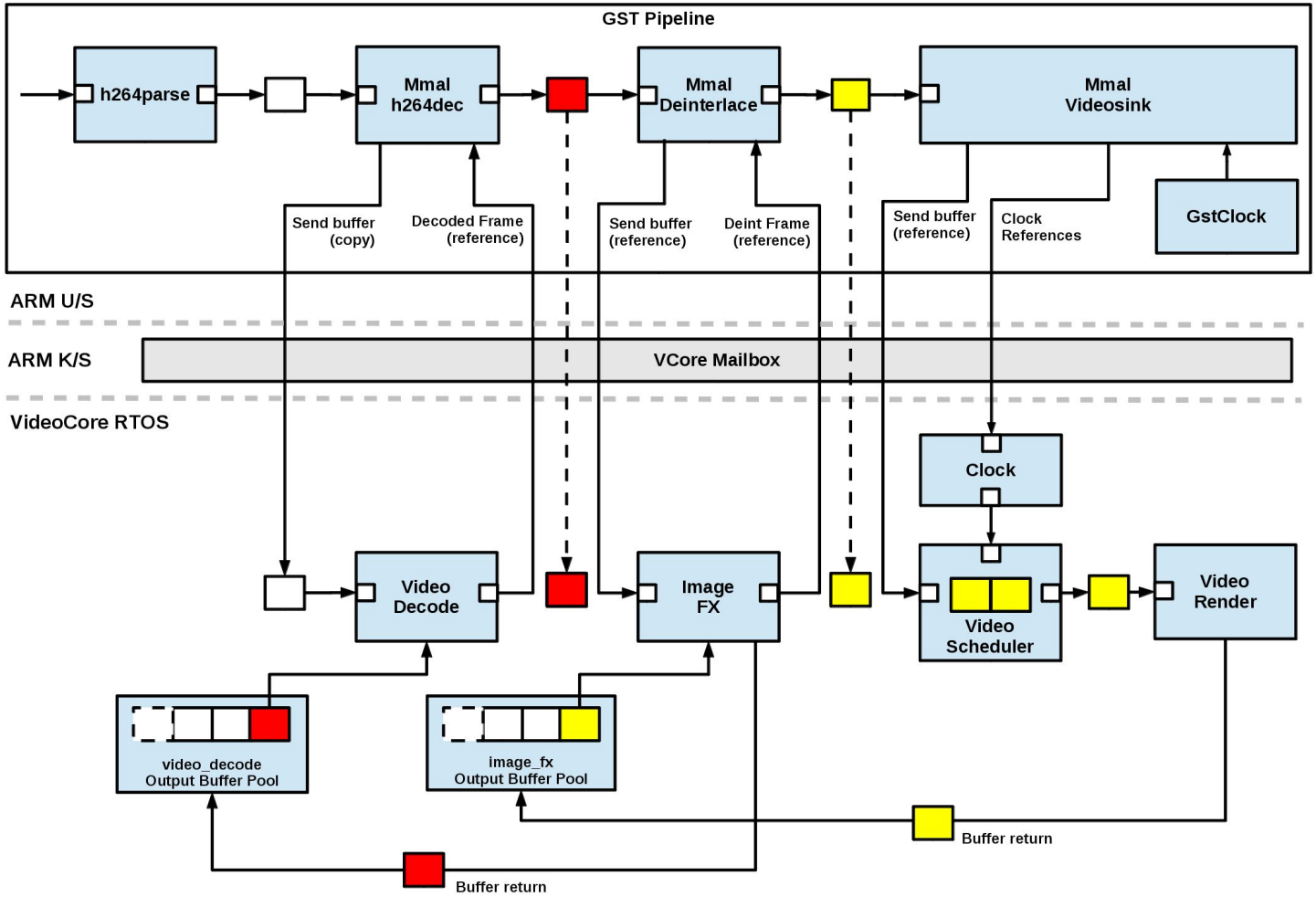
- Video decode: **- e.g. mmalh264dec**
  - H.264
  - MPEG-1, MPEG-2
  - VC-1 , WMV1,2,3
  - VP6,8
- Deinterlace - **mmaldeinterlace**
  - Uses “image_fx” VC component
- Video Sink - **mmalvideosink**
  - Uses VC-side clock & scheduler for smooth playback
- GL Uploader - **mmalglupload**
  - Use with glimagesinkelement
  - Experimental - use mmalvideosink instead for best results

youview

A few challenges we had to overcome

- VSync-aligned presentation & Pipeline/HDMI clock jitter
  - Solution: use VC Video Scheduler & Clock to pace final presentation
- GST & VC clock drift
  - Solution: video sink provides GST clock samples to VC clock component
- MMAL doesn’t like resizing buffers on rep change
  - Solution: always size buffers for max 1080
- Thread-safety in MMAL header refcounting
  - Solution: reimplemented those parts with atomics for now

youview


GST Pipeline
h264parse
Mmal
h264dec
Mmal
Deinterlace
Mmal
Videosink
Send buffer
(copy)
Decoded Frame
(reference)
Send buffer
(reference)
Deint Frame
(reference)
Send buffer
(reference)
Clock
References
GstClock
ARM U/S
ARM K/S
VCore Mailbox
VideoCore RTOS
Clock
Video
Decode
Image
FX
Video
Scheduler
Video
Render
video_decode
Output Buffer Pool
image_fx
Output Buffer Pool
Buffer return
Buffer return

A few outstanding issues

- Needs a good testsuite!
- Transition *between progressive and interlaced* representations is not as smooth as we'd like
- Deinterlacer could be smarter:
  - Doesn't always correctly detect mode
  - Image_fx is configured with undocumented "magic numbers"
- Video codecs other than H.264, MPEG-2 not well tested
- GL support could be improved
- No audio elements - OMX is fine for our audio needs
- Other components could be implemented - e.g. camera, encode, scaling/resize

youview

OK, that's it. Thanks for listening!

Source-code available on GitHub:

https://github.com/youviewtv/gst-mmal

Feedback & contributions welcome!

References:

[1] https://cgit.freedesktop.org/gstreamer/gst-omx

[2] https://www.raspberrypi.org/forums/viewtopic.php?f=70&t=127842

[3] http://www.jvcref.com/files/PI/documentation/html/

youview