Raspberry Pi 4 Vulkan Driver

Open Source Summit + Embedded Linux Conference Europe 2020

Alejandro Piñeiro Iglesias <apinheiro@igalia.com>
Contents

- Development story
- Current state
- Implementation challenges
- Future plans
- Contributing
Development Story
Development Story

- Driver code name: V3DV.
- Development started in a public fork of Mesa.
- Leverages Mesa Vulkan WSI.
- Expands existing V3D NIR compiler.
- Same kernel interface as V3D.
Development Story

- [Nov 19] Development start.
- [Jan 20] Triangle demo.
- [May 20] Bunch of Sascha Willem’s demos running.
- [Jun 20] Moved development to open repositories.
- [Jul 20] All Quake games working.
- [Oct 20] Moved development to Mesa upstream
Development Story

• Initial early milestone to render on hardware.
• Vulkan CTS to help iterative feature development.
  – Requires minimal functionality in the driver first.
  – Helped improve CTS coverage.
Development Story

• Growing subset of CTS for regression testing.
  – Parallel deqp runner for faster execution.
  – Currently ~10K tests (~10% of CTS pass list).
• Weekly rebases and full CTS runs.
• Assert everywhere philosophy.
• Progress updates via blog posts.
Current State
Current State

• Vulkan 1.0 mandatory feature set complete.
  – A bunch of optional features too.
  – Many optional features and extensions missing.

• Current focus on CTS conformance.
  – Passing ~110K tests, ~4 fails to go.
Current State

- VkQuake 1-3 & OpenArena.
- PPSSPP (Vulkan PSP emulator).
Current State

- Many demos from Sascha Willems working:
Current State

• Not much performance work yet.
  – Mostly for the Quake games.
  – VkQuake3 much faster than its GL1 renderer.
Current State

- Aware of some slow paths in the driver.
  - Particularly for some cases of transfer ops.
  - Possibly underused TFU unit.
Implementation Challenges
Implementation Challenges

• Vulkan expects everything to execute in GPU.
  – Not quite possible for us in a few selected cases.
  – Caused some implementation churn.
  – Incurs in additional coordination (flushes).
Implementation Challenges

- Linear display pipeline in Raspberry Pi 4
  - V3D cannot sample from linear images.
  - For now, we don’t support sampling on swapchains.
  - We should be able to sample in windowed mode when running inside a compositor… worth it?
Implementation Challenges

• Vulkan pipeline state not always sufficient.
  – Would like to emit shader variants based on texture formats for optimal performance.
  – We don’t know formats until descriptors are bound.
  – Pre-compile 2 shader variants in advance.
    • Optimal case: use 16-bit return size
    • Fallback case: use 32-bit return size
Implementation Challenges

- Mesa WSI implementation not optimal for us.
  - Optimal path requires PCI GPU and VK_EXT_pci_bus_info.
  - Raspberry Pi display device is not a PCI device.
    - We just want to check that DRI3 device matches.
  - RFC MR with a solution proposed.
Future Plans
Future Plans

• Short term:
  – Vulkan 1.0 conformance.
Future Plans

• Long term:
  - Explore better TFU unit usage.
  - Better WSI platform support.
  - Optimal implementation of input attachments.
  - Optional features & extensions
  - Maybe Vulkan 1.1?
Future Plans

• Long term:
  – Improve code reuse with GLES driver.
  – Maybe port some features to GLES driver:
    • Hardware multisample resolve.
    • Sample rate shading.
    • Robust buffer access.
    • Etc.
Future Plans

• Long term:

More real world testing!!
Contributing
Contributing

- Stable context to enable external contributors.
- V3D 4.2 docs not available to general public.
  - GLES 3.1 open source driver can make up for this.
- Lots of FIXMEs in the source code.
- Many optional features pending.
- Testing and performance feedback.
Contributing

• Resources:
  – #videocore @ freenode
  – mesa-dev mailing list
  – Gitlab issues
Special Thanks

- Mesa community, for NIR, SPIR-V translator, WSI bits, etc.
- Existing Mesa Vulkan driver developers.
- Eric Anholt
- Dave Emett
Q&A

We are hiring: www.igalia.com/jobs