Running OpenCL C Kernels on Vulkan for Fun and Profit
Eric Berdahl | Sr. Engineering Manager, Mobile Video Products
Vulkan Compute
Porting OpenCL C to Vulkan
Ralph Potter, Codeplay

CLSPV Compiler
- OpenCL C
- Clang (OpenCL C Frontend)
- SPIR 1.2 (LLVM IR)
- Vulkan SPIR-V
- CLSPV Module Passes
- Descriptor Map

Acknowledgements
- Google
- Adobe
- codeplay
- David Neto, John Kessenich
- Eric Berdahl
- Neil Henning, JinGu Kang
Adobe Premiere Pro CC

= workflows and user experience

+ Mercury Playback Engine
Thought Experiment: Mercury Playback Everywhere?

Mercury Playback Engine

CUDA, OpenCL, Metal

CUDA, OpenCL

Metal

Vulkan?
Managing Complexity

- Great Flexibility
- Expensive to rewrite large amounts of code

- Greatest breadth and impact
- Sensitive to quality and performance

Tremendous leverage!
Observations

- Vulkan and SPIR-V today are sufficient to host compute shaders written in novel languages
- In the App+Driver+Compiler triad, the compiler has huge advantages!
- Clspv has allowed us to compile several hundred thousand lines of OpenCL C code to SPIR-V (graphics) with minimal changes
- The Clspv experience revealed coverage gaps in Vulkan CTS
  - We have improved CTS coverage
  - Compilers are remarkably agile for working around bugs in drivers
- Clspv is today an insanely useful prototype with which I’m building a commercial Android product
Project Rush – Unveiled at VidCon 2018