SPIR: Standard Portable Intermediate Representation

Tilmann Scheller
LLVM Compiler Engineer
t.scheller@samsung.com

Samsung Open Source Group
Samsung Research UK

EuroLLVM 2015
London, United Kingdom, April 13 – 14, 2015
Overview

- Introduction
- SPIR-V
- Summary
Introduction
History

- SPIR 1.2/2.0
  - Intermediate representation for OpenCL kernels
  - Subset of LLVM IR, SPIR adds metadata and intrinsics to LLVM to get the functionality needed for OpenCL
SPIR-V

- Announced in March 2015 together with Vulkan (next generation OpenGL API)
- Unified intermediate representation for compute and graphics
- New IR but maps easily to LLVM IR
- IR defined and controlled by the Khronos Group
- Stable binary format for deployment of shaders/compute kernels
- Native support for many graphics idioms
SPIR-V

- Compilation unit has a series of functions which are formed by a CFG of basic blocks
- SSA-form
- Hierarchical type building
- Logical variables with type

```
[SPIR-V Magic #: 0x07230203]
SPIR-V Version 99
Builder's Magic #: 0x051a008B
	<id> bound is 50
0
OpMemoryModel
Logical
GLSL450
OpEntryPoint
Fragment shader
function <id> 4
OpTypeVoid
	<id> is 2
OpTypeFunction
	<id> is 3
return type <id> is 2
OpFunction
Result Type <id> is 2
Result <id> is 4
0
Function Type <id> is 3

Stream of Words...
```
Differences to LLVM IR

- Can be translated to another IR with a very small self-contained translator
- Binary format is a simple linear stream of words which can be decoded with a very small decoder
- Native support for matrices and unsigned integers
- Special built-in variables and many graphics idioms are native to SPIR-V
- No metadata needed to support the features of GLSL
Summary
Summary

- New standardized stable binary format for shaders/compute kernels
- Different from LLVM IR but can be translated easily
- Opportunities for first-class SPIR-V support built around LLVM (SPIR-V frontend/backend)

For more information about Vulkan/SPIR-V check out the Khronos UK Chapter Meeting starting at 1PM in PSH 314!
Thank you.