The Lima Driver: Liberating the ARM Mali GPU

Luc Verhaeegen
libv@codethink.co.uk
Why?

- x86 vs ARM
- GNU/Linux vs Android/Linux
- Manufacturers vs Users
ARM GPU Vendors

- Imagination
- Qualcomm
- Nvidia
- ARM
- Vivante
- ...

ARM GPU Vendors
Mali™-200

- Vertex Processor
- Fragment Processor

Memory Management Unit

AMBA® 3 AXI3™ Interface
Availability?

• Mali-200:
  • Telechips 8902
  • Telechips 8803

• Mali-400:
  • Samsung Exynos (Mali-400 MP4)
  • ST/E Novathor
  • Allwinner A10
  • Amlogic 8726-M
ARM's Mali driver stack

Application

libEGL_mali.so

libGLESv2_mali.so

libMali.so

mali.ko

Linux Kernel
Infrastructure vs Compiler

Application

libEGL_mali.so

libGLESv2_mali.so

infrastructure

ESSSL Compiler

mali.ko

Linux Kernel
Userspace-Kernel Interaction

Userspace

mali.ko

GP
PP

Mali

ARM SoC

RAM

userspace

Userspace-Kernel Interaction

[diagram of userspace-kernel interaction with labeled components]
Mali Kernel Interaction

- Retrieve GPU and Memory info
- Map some GPU memory
- [Build up command stream in GPU memory]
- Submit GP job
- Wait for GP job done
- Submit PP job
- Wait for PP job done
All you need is... LD_PRELOAD

To wrap open(), ioctl(), mmap():

- Get /dev/mali fd from open()
- Get memory from mmap()

At GP job start:
  - Dump GP registers
  - Dump memory

At PP job start:
  - Dump PP registers
Limare

- Prototyping only!
- Infrastructure work only:
  - Command stream build up
  - Interface with compiler
  - Linker
  - Job handling
- Small, single frame, tests
- Dumps render to .bmp and fbdev
Limare Methodology

1) Create single frame GLES application
2) Capture command stream
3) Replay command stream
4) Reduce and analyze command stream
5) Adjust Limare infrastructure
6) goto 1
Status: Working

- Mali-200 and Mali-400
- Render to any size the HW supports
- Shader linking
- Assignment of Uniforms, attributes, varyings
- Multiple draws.
- Android app
Polygon Lists

PP

PP Commands

PP (Pixel? Processor)

Polygon List Addresses & Coordinates

Polygon List

Diagram illustrating the flow of data through a PP (Pixel Processor) system, from polygon lists to addresses and coordinates.
Compiler

int __mali_compile_essl_shader(...);
Binary Shader Compiler?

- Depends only on libc
- No setup needed
- Single function call
- Needs source, shader type, 1 struct

→ Quick and easy standalone usage!
Shader instructions

- 128bit VLIW
- Fixed structure
  - vertex: Varying and attribute positions are known
- ? Work for both Mali-200 and Mali-400 ?
- ... [TODO]
Up Next!

- Textures
- Kick-start shader instruction RE-ing
- Setting depth, cull directions, etc...
- More tests/demos!
- Build system, basic memory management, documentation...
- Multiple frames?
Future

- Gallium driver with binary compiler
  - 2-4Months
  - “Should” match performance
- Gallium driver with open compiler
  - Depends...
- DRM driver
  - Next 6 months: Counterproductive
  - Afterwards: Keep old API for compatibility
Contribute!

- Site: http://www.limadriver.org
- Mailinglist: lima@limadriver.org
- Get a device
- Start playing!
http://www.limadriverriver.org/