The Lima Driver

An update on the command stream / driver side of the open source driver for the ARM Mali GPUs.

Luc Verhaegen <libv@skynet.be>
Some ARM GPU history

- Before: nothing but talk.
- 2011: Idea and initial mali RE-ing.
- 2012: Early days.
  - FOSDEM: first renders.
  - Linuxtag: correctly spinning companion cube.
  - Other ARM GPU projects started.
- 2013: Full proof of concept.
  - FOSDEM: Q3A timedemo.
- 2014: Actual driver for the masses?
Last year.

- Memory management
- Initial mesa driver
- ...

That's it really...
Hardware: Cubietruck

- Allwinner A20
  - Dual core Cortex A7 @ 1GHz
  - Mali M400MP2 @ 312MHz
- Board with all the fixin's
- Schematics available, but not OSHW
- Linux-sunxi code
  - Sunxi u-boot
  - Sunxi-3.4 kernel
  - xf86-video-fbturbo
  - ...
http://linux-sunxi.org/

- Big community with lots going on.
- Easy to replace android with proper linux
- Joined june 2012: no more android!
- Vendor code is a mess.
- Sunxi-kms talk later.
- Danger of throwing the baby out with the bathwater.
Standard, “old”, mesa driver...

- Lima is 2 separate projects
  - 2 insane tasks already.
  - Aids debugging and benchmarking.
- Gallium doesn't allow for that.
  - Symbol layout first, then forced on compiler.
  - No real way to hook in another compiler.
  - “Don't you dare do an intel!”
- Needs just 2 small patches for an external shader compiler.
Standalone Mesa driver.

- Our hardware is:
  - Mix-n-match IP block.
  - Limited resources.
  - Limited free support.
  - Limited audience for a free driver... unless...

- Not hard:
  - Only needs build system changes.
  - Currently supports Mesa-8.0.x through Mesa-9.2.x.
  - Mesa packages available at linux-sunxi.org.

- But… Politics...
Synchronization.

- Limited resources, so all silicon needs to be used.
- Mali is a proper mix-n-match SoC IP block.
- Vendor kernels are a mess...
- No unified command processor to synchronize everything for you.
- `dma-buf` is a good first approach, but for some reason lacks synchronization → `dma-fence`
Frame lifetime

App

GL code

Mesa

Frame buildup

limare

Job threads

Kernel

Mali kernel driver

GPU

GP job

PP job

Start

Flush

Flip

Start

GL code

Flip
Performance: limare tests

Smoothed cube
- limare: 725.75fps
- lima: 524.26fps
- binary: 510.90fps

Textured cube
- limare: 311.65fps
- lima: 237.23fps
- binary: 268.83fps

Companion cube
- limare: 225.88fps
- lima: 231.84fps
- binary: 217.82fps
Performance: es2gears

- lima (-g -O0): 176.05fps
- lima (-O3): 264.65fps
- binary: 321.21fps
Performance: glmark2-es2

- **Build (vbo)**
  - lima: 231fps
  - binary: 300fps

- **Texture (linear)**
  - lima: 221fps
  - binary: 253fps

- **Shading (phong)**
  - lima: 80fps
  - binary: 155fps

- **Bump (high poly)**
  - lima: 54fps
  - binary: 74fps

- **Pulsar**
  - lima: 345fps
  - binary: 384fps

- **Jellyfish**
  - lima: 49fps
  - binary: 53fps
Future.

- Full glmark2-es2 coverage & release.
- Incorporate Connor's glsl work.
- Implement dma-buf/fence.
- Proper tie in with at least sunxi display code.
- Job submission rewrite/kernel driver rewrite???
Tamil driver.

- Mali T6xx/T7xx
- Fresh start.
- Missing community for the chromebook,
  At least it's not android...
- Early days:
  - Capture/replay.
  - Shader Compiler.
A word from our... Beneficiaries?

- “[...] we currently see very little advantage to ARM in the creation of such a driver and several disadvantages.”
- “[...], this project is producing and publicising your versions of documentation about the internals of our hardware, which we do not want, [...]]”
- “Finally, if we wanted an Open Source driver, we could Open Source our own driver...”

Jem Davies,
ARM Fellow, VP of Technology, Media Processing Division,
10th of May 2013
Questions?