Open ARM GPU drivers:

Where are we today, 1 year after the unveiling of the lima driver.

Luc Verhaegen

libv@skynet.be

The problem.

The solution?

Legal?

(IANAL)

GPU #1:

ARM Mali

ARM Mali-200/400

- OpenGLES 2.0
- Separate Vertex (GP) and Fragment (PP) shaders
- SoCs:
  - Samsung exynos 4: world beater in 2011-2012
  - Allwinner, Amlogic, Rockchip, Telechips, Wondermedia, ...
- Future: Mali-450

ARM Mali T6xx

- OpenGLES 3.0/OpenCL 1.1
- Unified shaders
- SoCs:
  - Samsung exynos 5
- Future: ARM T624/T628/T658/T678

Project: Lima

Luc Verhaegen (libv)

Connor Abbott (cwabbott)

Ben Brewer (flatmush)

Lima

- Status: research (highly advanced)
  - No big secrets left in command-stream
  - Compiler is tough due to Mali architecture
  - Actual driver work will start after FOSDEM
- Full GNU/linux systems available
- Website: http://limadriver.org
- Repository: http://gitorious.org/lima
- Demos at the end of this talk / Lima compiler talk in X.org DevRoom (K.3.401) at 17:30.

GPU #2:

Qualcomm Adreno

Qualcomm Adreno 2xx/3xx

- Former ATI Imageon
- OpenGLES 2.0, 3.0 (3xx)
- Unified shaders
- SoCs:
  - Qualcomm Snapdragon S1-S4
  - Freescale iMX5

Project: Freedreno

Rob Clark
(robclark)

Freedreno

- Status: WIP driver
  - Job submission is like ATI radeon.
  - Command-stream is mostly known.
  - Shader architecture also mostly known.
  - WIP xf86 (exa), mesa (gallium) drivers available
- No proper GNU/linux available :(
- Website: http://freedreno.github.com/
- Further details and demos in the X.org DevRoom (K.3.401) at 17:00 today.

GPU #3:

Nvidia Geforce ULP (Tegra)

Nvidia Geforce ULP/Tegra

- Supposedly a cut down older Geforce but...
- OpenGLES 2.0, 3.0 (tegra4), openCL/renderscript (tegra4)
- Split vertex and fragment shader design.
- SoCs: well duh.

Project: Tegra-re

Erik Faye-Lund
(kusma)

Tegra-re

- Status: Early research.
  - Job submission wrapping.
  - Early shader disassembler.
  - Early command stream capture.
  - Replay is up next.
- Limited availability of GNU/linux systems (AC-100, Trimslice)
- Website: https://github.com/kusma/tegra-re

GPU #4:

Vivante GCxxxx

Vivante GCxxxx

- OpenGLES 2.0/3.0
- Unified shader design
- SoCs:
  - Marvell Armada
  - Freescale iMX6
  - Hisilicon, Ingenic, rockchip, loongson, ...

Project: etna-viv

Wladimir J.
Van Der Laan
(wumpus)

Etna-Viv

- Status: early research (rapidly advancing)
  - Full command stream capture and replay
  - Slowly prying apart command stream
  - Shader disassembler and assembler
- Repository: http://github.com/laanwj/etna_viv

GPU #5:

Broadcom Videocore

Broadcom Videocore

- OpenGLES 2.0, OpenVG, OpenMax, …
- … Media Decoding, Shader Compiler, Modesetting, ARM Core bringup...
- DSP based proprietary architecture.
- Found in Broadcom SoCs and standalone Broadcom Media Chips.

RPi == Closed platform.

- Videocore is a separate undocumented DSP/SIMD architecture.
- Runs a RTOS in a reserved bit of RAM.
- Responsible for much of the system level functionality, and even bringing up the ARM core.
- Communication from ARM userspace through a message-passing interface through the kernel and out some shared ringbuffers.
- Userspace GPU drivers are just a shim.

Project: videocore

Scott Mansel
(phire)

Matthew
Parlane
(matt)

(booto)

(Herman)

(David)

Tiernan
Hubble
(thubble)

Mathias
Gottschlag
(mgottschlag)

(eizo_san)

Mark Marshall
(mm120)

Videocore

- Status: research (rapidly advancing)
  - Scalar processor fully RE-ed.
  - Loads of documentation.
  - Assembler/disassembler.
  - Binutils
  - Compiler work started.
  - Some Hello World code for booting the RPi
- Repository: https://github.com/hermanhermitage/videocoreiv/
- IRC: #raspberrypi-internals

GPU #6:

Imagination PowerVR SGX

PowerVR SGX (5xx)

- OpenGLES 2.0, 3.0, OpenVG, ...
- Unified shader design (USSE)
- SoCs:
  - Texas Instruments Omap.
  - Apple A4, A5, A6.
  - Intel Poulsbo through Medfield.
  - … Everyone really, at one point or another...
- Future: Rogue (6xx)

PowerVR is ...

... a mix of microcode, kernel and userspace.

This makes PVR:

- Highly adaptable
- Scalable
- Versatile

But also...

- Fragile
- Synchronization nightmare
- Hard to debug and maintain...
- … and therefore hard to Reverse Engineer.

Project: ...

(slightly gratuitous, but only slightly ;p)

Lima demo time!

Demo Hardware: Mele A1000

- Allwinner A10
  - Single ARM Cortex A8 at 1152MHz
  - 512MB of 360MHz DDR3
  - Mali-400MP1 at 320MHz
- UART exposed
- http://linux-sunxi.org
  - U-boot
  - Kernel
  - Further utilities and documentation.