Status of the Embedded GPU Space in Linux

Robert Foss
Collabora Ltd.
robert.foss@collabora.com

***Abstract*—Embedded graphic processing units (GPUs) are used for increasingly diverse types of applications, placing new requirements on the GPUs, only some which are supported by vendors. While most IP vendors still don't support their devices with Open Source drivers, in many case the Open Source community has stepped up to provide these.**

In this article the current landscape of embedded GPUs and current challenges with an emphasis on their respective Open Source support will be discussed.

Keywords—virtual reality; vulkan; open source; linux; graphics; driver;

I. Introduction

The embedded GPU space is seeing rapid changes with the introduction of new standards like Vulkan[1] and new applications like VR and wearables.

This, of course, affects the GPU hardware and what we expect from it, which in turn means that the software ecosystem is seeing many new demands placed on it that spans from low latency output [2] of assisted reality and virtual reality (XR) to low power requirements of wearables [3].

The motivation for product manufacturers to choose using a fully Open Source stack comes down to the fact that in most cases the proprietary stack doesn't meet their needs, including stability, performance and features.

For some manufacturers other factors come into play, like the ability to provide decade long support for their products. In order to provide a secure operating system, the graphics stack has to be kept up to date. For this to happen either an Open Source stack has to be used, or the GPU-vendor has to be contracted to provide support for the term of your product support. To product manufacturers this introduces a risk and a financial burden. What happens if your vendor chooses to no longer offer support? Or if the support comes at an unacceptable cost? Or if the vendor exits the graphics business?

Many drivers have come as a result of reverse engineering (RE) efforts by the Open Source community and are now maintained as a part of the Linux kernel, Mesa, and libdrm. Often the motivation behind the RE effort stems from a desire to fully enable powerful hardware that is already available, for example in cheap development kits or single board computers (SBCs).

New standards like Vulkan also have a large impact on the software stack, with some implementations going as far as implementing legacy standards like OpenGL on top the Vulkan stack.

This paper will look at different GPUs and provide an overview for their Linux support, including a comparison of drivers provided by the vendor against the one provided by the Linux Community, looking at the benefits and disadvantages of each of them.

II. New Developmental Targets

A. *Virtual Reality*

Assisted reality and virtual reality (XR) is a new type of frontier for graphics stacks and input stacks. The importance of latency has previously been much lower, and as a result both graphics drivers and entire graphics stacks are not optimized for this use case. XR also places similar requirements on the input stack, where latency can become an issue too. Unlike most features low latency puts system-wide requirements on the graphics stack, to which there are no simple fixes.

A partial solution, called DRM Leasing[4], has been put forward by Keith Packard. It allows applications to take ownership of hardware components directly, thereby bypassing traditional components of the graphics stack like window servers and desktop environments. By doing so many latency introducing components can simply be avoided.

This approach is based on flagging certain display components in the kernel as unavailable for normal applications and then making these components available directly to user space applications.

Other components like GPU shader-compilers, human interface devices (HID) and displays themselves all have to be optimized for low-latency in order to provide the best possible user experience.

B. *Vulkan*

Vulkan offers a new tier of low-level GPU access, and with it a lot of flexibility. This brings with it the possibility of being more power efficient if the new API is utilized correctly on hardware that supports it. For example power savings of up to 20% can be achieved for multimedia use-cases[5].

Vulkan also enables novel use-cases like implementing OpenGL on top of Vulkan in order to always have an OpenGL driver for any Vulkan enabled GPU, like, e.g. Zink[6], which may be desirable to lessen the burden on maintaining an OpenGL driver when a vulkan driver is already available.

Industry adoption of Vulkan is keeping a high pace. Android[7] is moving over to using Vulkan as its main graphics API, which means not only making it available to applications, but also moving all internal Android 3D-rendering over to Vulkan.

From the standpoint of a product manufacturer Vulkan offers some new benefits, like the Conformance Test Suite (CTS) being available as an Open Source project[8]. This enables manufacturers to ship product and have high levels of confidence in the stability and standards conformance of their products.

III. GPU Drivers

Open Source Linux drivers have existed for a long time, especially for Desktop class GPUs, but during the past few years support for Embedded class GPUs has materialized too.

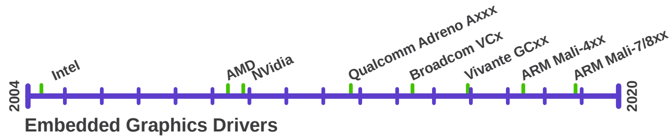

Figure 1: Timeline of Open Source driver becoming available for vendors.

In Fig. 1 the date of introduction for the drivers of GPU vendors has been listed. A closer inspection of the second half reveals that most to Embedded GPUs now have a viable Open Source drivers. The latest additions, the ARM T7/T8/Gxx drivers, are rather new, but quite close to being included in the normal Linux graphics stack.

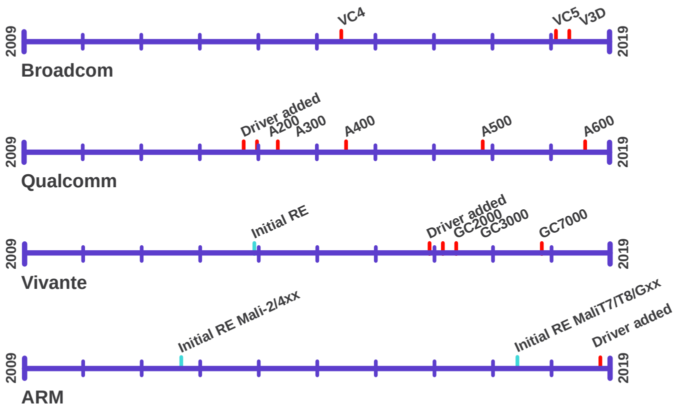

Figure 2: Timeline of embedded GPU driver development

As can be seen from Fig. 2, driver support for different vendors vary, only the ones that target the embedded space and offer some Open Source support have been listed. Currently, the full range of Broadcom GPUs, Qualcomm GPUs and Vivante GPUs are supported, while support for ARM GPUs is still in the early stages.

Another topic related to these drivers is how long drivers stay supported. Naturally that differs between GPUs, but for example the AMD R100 series of GPUs were launched in 2000[9], and the Open Source driver is still maintained and receives patches[10].

Notable exceptions from this list are NVidia and Imagination Technologies GPUs. While both of them make GPUs targeting the embedded space neither have participated in supporting Open Source drivers, and NVidia actively prevents the driver that exists from being usable by contractually limiting the required firmware binaries from being distributed with the Open Source driver. It should be noted however that this does not apply to some NVidia Tegra GPUs, where firmware is freely distributed.

With this overview out of the way, we can have a look into the status of the individual drivers.

A. *Broadcom*

The VC4 and V3D[11] drivers were created by Eric Anholt and are being actively maintained by him and supported by Broadcom.

The initial VC4 driver development was started as a result of it being the GPU shipped in the Raspberry Pi series of SBCs. Since then the VC5 & VC6 GPUs have had support added in a driver called V3D.

Currently the driver is very mature, and it offers the best GPU-vendor support out of the embedded drivers.

B. *Qualcomm*

The Adreno driver, freedreno[12], originated from the Open Source ecosystem and Rob Clark's efforts, and is by now a very mature driver. Additionally, Qualcomm is one of the very few embedded hardware vendors that actually support their Open Source driver to some extent. This support could be more extensive, but it is still far ahead of the competition in the embedded space listed below.

Support for their hardware is complete, in that it is offered for the complete A2-6xx range of GPUs (Fig. 2).

As Qualcomm is an SOC manufacturer, the Adreno GPUs are only available in their Snapdragon system on a chip (SOC).

C. *Vivante*

The Vivante driver, etnaviv[13], has gone from being reverse-engineered to being a very stable and performant driver in a very short time.

The timespan (Fig. 2) between initial reverse-engineering and the first GPUs being support is approximately 3 years. Since then the full range of Vivante GPUs have had support added, including support for their newest GPUs which are still only sparsely available in the marketplace today.

Due to the similarities between different series of Vivante GC GPUs, the effort required for supporting a new GPU has been relatively low historically[14].

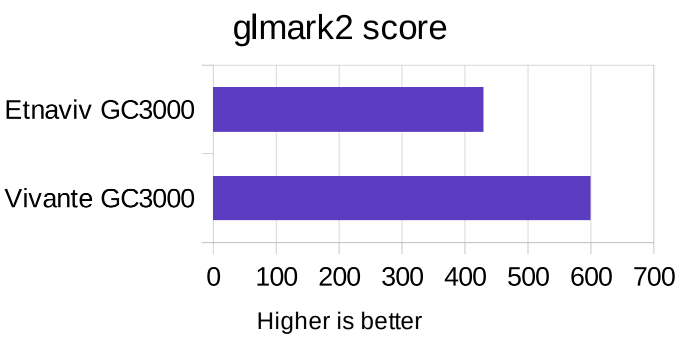

Figure 3: Vivante & Etnaviv GC3000 benchmark

The performance of the Open Source driver (Fig. 3) is comparable to the proprietary one, and has been improving much over time.

This driver has to this moment not received any support from its hardware designer.

D. *ARM*

The ARM drivers, lima[14] and panfrost[15], have been a long time coming. Initial work on the reverse engineering of the Mali-2/3/4xx series of GPUs was presented at FOSDEM 2012 by Luc Verhaagen[16], but the work was never completed or submitted to the Linux graphics stack projects. Nevertheless, recently it has been restarted and it is now close to the point were it will be included in the Linux graphics stack.

Open Source driver support for the Mali-T7/T8/Gxx series has come a very long way in the last two years. The work started with reverse engineering in 2017, and today the panfrost driver is achieved some milestones like correct 3D-rendering of simple scenes. The next step is getting panfrost included in the Linux graphics stack.

The Mali-T7/T800 series of GPUs share some similarities with the Gxx that enable both of them to be supported by the same driver. Much of the panfrost effort has been focused on the Mali-T860 GPU due to it being widely available in development boards, like those based on the Rockchip RK3399 SOC.

At this point the lima and panfrost drivers have received no support from the hardware designer.

IV. Conclusion

Supporting features like XR, together with new standards like Vulkan present the new bleeding edge for graphics, and the challenge is being met by both the proprietary drivers and the Open Source ones alike.

The process of reverse engineering the GPUs of a vendor and producing a first driver seems to take 2-3 years in most cases. After initial support for a GPU has been implemented, later generations of that GPU seem to have support added relatively quickly.

The Open Source graphics stack has several legacy drivers that have been maintained for decades after the last related product was sold. For the embedded space, the importance of this cannot be understated. It means that products can be based on the official Linux repositories and thereby ensure that future bug-fixes will be available and not blocked by proprietary drivers that are incompatible with modern versions of the Linux kernel.

Acknowledgment

The Linux Graphics community has committed large amounts of time and effort towards supporting the hardware released by different vendors. Collabora has sponsored my time for preparing this document.

References

[1] Vulkan Graphics Specification https://www.khronos.org/registry/vulkan/specs/1.0/html/vkspec.html
2018-01-09

[2] LINCOLN, Peter, et al. From motion to photons in 80 microseconds: Towards minimal latency for virtual and augmented reality. IEEE transactions on visualization and computer graphics, 2016, 22. Jg., Nr. 4, S. 1367-1376.

[3] TAN, Cheng, et al. LOCUS: Low-power customizable many-core architecture for wearables. ACM Transactions on Embedded Computing Systems (TECS), 2018, 17. Jg., Nr. 1, S. 16.

[4] Keith Packard, https://keithp.com/blogs/DRM-lease/
2018-01-14

[5] GAMBHIR, Mahak; PANDA, Swati; BASHA, Shaik Jani. Vulkan rendering framework for mobile multimedia. In: SIGGRAPH Asia 2018 Posters. ACM, 2018. S. 64.

[6] Zink, OpenGL implemented in Vulkan https://www.kusma.xyz/blog/2018/10/31/introducing-zink.html
2018-01-09

[7] Android Graphics Developer Guidelines https://developer.android.com/ndk/guides/graphics/
2018-01-09

[8] Vulkan Conformance Test Suite https://github.com/KhronosGroup/VK-GL-CTS
2018-01-10

[9] ATI R100 https://en.wikipedia.org/wiki/ATi_Radeon_R100_Series
2018-01-11

[10] Mesa R300 Driver https://gitlab.freedesktop.org/mesa/mesa/tree/master/src/gallium/drivers/r300
2018-01-11

[11] V3D development log https://anholt.github.io/twivc4/
2018-01-11

[12] Freedreno reverse-engineering project https://github.com/freedreno
2018-01-09

[13] Etnaviv reverse-engineering project https://github.com/etnaviv
2018-01-09

[14] Lima reverse-engineering project https://gitlab.freedesktop.org/lima
2018-01-09

[15] Panfrost reverse-engineering project https://github.com/Panfrost
2018-01-09

[16] Fosdem: Liberating ARM's Mali GPU https://archive.fosdem.org/2012/schedule/event/mali.html
2018-01-10