Adapting etnaviv to contemporary Vivante hardware

Christian Gmeiner

2025-10-01

igalia

History Lesson: Hardware Generations

- **GCW Zero**: mips (GC860)
- **NXP i.MX6** series: armhf (GC880, GC2000, GC3000)
- **NXP i.MX8** series: arm64 (GC7000 - different variations)

igalia

History Lesson: Markets

- Industrial automation
- Healthcare devices
- Automotive infotainment
- SoC availability > 15 years

igalia

History Lesson: Driver Limitations

- Some features are emulated within shaders
- Some features are impossible to support
- etnaviv is a solid gles2 driver

Contemporary Hardware

```
/* gc7000XSVX_6009 */
{
    0x7000, /* ChipID */
    0x6009, /* ChipRevision */
    0x70008, /* ProductID */
    0x0, /* EcoID */
    0x0, /* CustomerID */
    ...
    0x1, /* gcFEATURE_BIT_REG_GeometryShader */
    ...
    0x1, /* gcFEATURE_BIT_REG_TessellationShaders */
    ...
    0x1, /* gcFEATURE_BIT_REG_Evis */
    ...
},
```


igalia

Contemporary Hardware

i.MX 8QuadMax

- 4x Cortex-A53
- 2x Cortex-A72
- 2x GC7000XSVX (Split GPU architecture)

igalia

The good

- A more recent binary blob driver
- Can run gles3.x and vulkan CTS for RE
- Mainline linux works on my chosen SoM

The good: Hardware based xfb

- 4 * buffers (address, size, stride)
- 4 * 128 descriptors
- pause/resume support
- query support (PIPE_QUERY_PRIMITIVES_EMITTED)
- context buffer
- maps very well to nir_xfb_output_info

The good: geometry shaders

```
<stripe name="GS" brief="Geometry Shader states"> <!-- GE
    <reg32 offset="0x01100" name="CONTROL"/>
    <reg32 offset="0x01104" name="OUTPUT_TYPE" type="GS_C
    <reg32 offset="0x01108" name="UNK01108">
        <bitfield pos="0" name="UNK0"/>
        <bitfield high="24" low="16" name="VERTICES_COUNT
    </reg32>
    <reg32 offset="0x0110C" name="START_PC"/>
    <reg32 offset="0x01110" name="END_PC" value="0x000000
        <doc>index of last instruction + 1</doc>
    </reg32>
    <reg32 offset="0x01114" name="INST_ADDR" type="VIVM"
    ...
</stripe>
```

The good

- fragment operations like alpha_to_coverage
- stencil texturing
- texture gather
- texelfetch
- tessellation
- multisample textures

igalia

The bad

- deqp GLES3 runs take much longer than the GLES2 ones
- around 20 DUTs
- more hardware for ci is needed
- goal: gles2 pre-merge testing for i.MX6Q

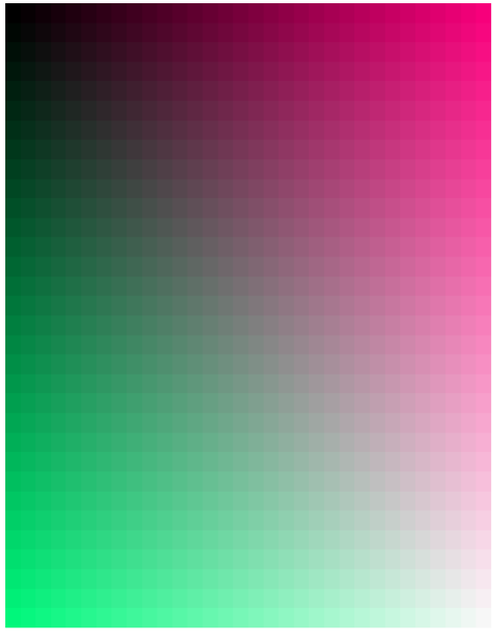
The ugly

128-bit formats are not supported in hardware but required for GLES3.

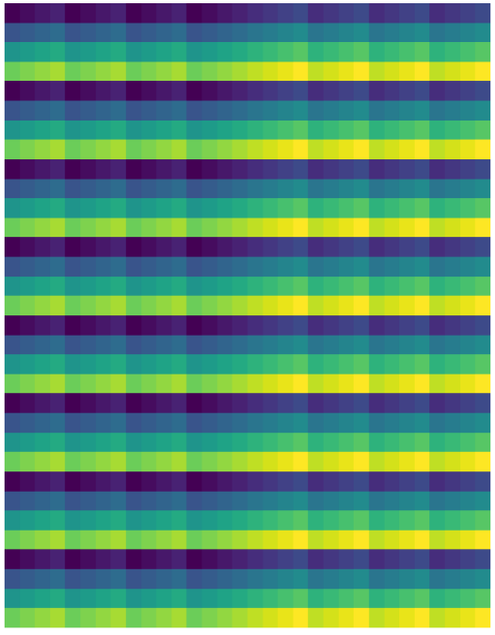
We must emulate them — the challenge is doing it fast.

The ugly - 128bit emulation

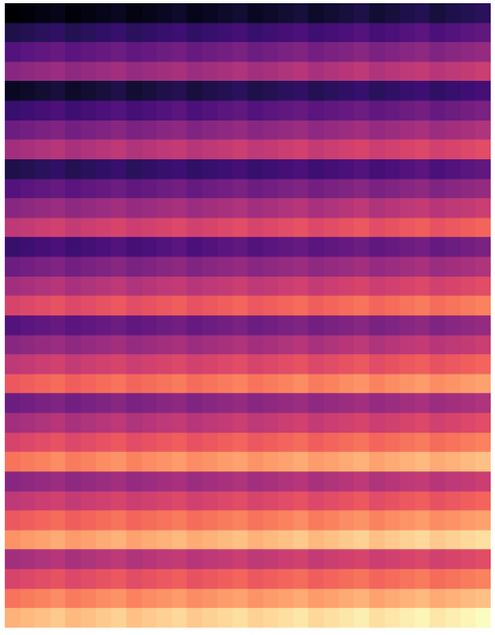
RGBA32F (visualized RGB)



Tiled RG plane



Tiled BA plane



igalia

The ugly - 128bit emulation

- Every clear and blit operation needs to be done twice

```
@@ -290,6 +290,7 @@ etna_blit_clear_color_blt(struct pipe_context *pctx, unsigned idx,
    uint64_t new_clear_value = etna_clear_blit_pack_rgba(dst->format, color);
    bool fast_clear = etna_blt_will_fastclear(dst_level, scissor_state);
    int msaa_xscale = 1, msaa_yscale = 1;
+   bool is_128bit_format = format_is_128bit(dst->format);

    translate_samples_to_xyscale(dst->texture->nr_samples,
                                 &msaa_xscale, &msaa_yscale);
@@ -333,8 +334,21 @@ etna_blit_clear_color_blt(struct pipe_context *pctx, unsigned idx,
       clr.rect_h = dst_level->height * msaa_yscale;
    }

+   if (is_128bit_format) {
+      clr.clear_value[0] = color->ui[0];
+      clr.clear_value[1] = color->ui[1];
+   }
+
    emit_blt_clearimage(ctx->stream, &clr);

+   if (is_128bit_format) {
+      clr.clear_value[0] = color->ui[2];
+      clr.clear_value[1] = color->ui[3];
+      clr.dest.addr.offset += (dst_level->size * dst_level->depth) / 2;
+
+      emit_blt_clearimage(ctx->stream, &clr);
+   }
+
    /* This made the TS valid */
    if (dst_level->ts_size) {
       if (idx == 0) {
```

The ugly - 128bit emulation

8.3 GPU affinity configuration

In the multi-GPU Independent Mode, application can specify to run on a specific GPU among the multiple GPUs through an environment variable VIV_MGPU_AFFINITY. Once an application's GPU affinity is specified, the application will only run on the specified GPU and will not migrate to other GPUs even if those GPUs are idle.

VIV_MGPU_AFFINITY is the environment variable to control the application GPU affinity on multi-GPU platform. The client drivers will assume they are using a standalone GPU through a gcoHARDWARE object no matter how this variable is set. The possible values for the environment variable VIV_MGPU_AFFINITY include:

- Not defined or
- Defined as "0" gcoHARDWARE objects work in gcvMULTI_GPU_COMBINED mode (default)
  - "1:0" gcoHARDWARE objects work in gcvMULTI_GPU_INDEPENDENT mode and GPU0 is used
  - "1:1" gcoHARDWARE objects work in gcvMULTI_GPU_INDEPENDENT mode and GPU1 is used

On a single GPU device, setting VIV_MGPU_AFFINITY to 0 or 1 does not make any difference as all application processes/ threads are bound to GPU0. But the application will fail the GPU context initialization if VIV_MGPU_AFFINITY is set to "1:1" (driver reports error).

The ugly - 128bit emulation

What about rendering into a such a format?

We make use of MRTs to render into each 64bit half, but with some shader magic. There is a nir pass that creates a second output, splits the result and writes 64bit data.

Works well, as we report half of the supported render targets and can use the others for this special 128 emulation case.

The ugly - 128bit emulation

Now it gets really ugly .. sampling from such formats.

On the i.MX8qm there are in total 80 samplers that can be used by any shader stage.

On other GPUs, however, only 32 samplers are available in total, and in some cases they are restricted to a fixed/static assignment across shader stages.

Status update

- i.MX8qm is my main target of interest
- mostly driven by CTS
- there are benefits for other Vivante GPUs

Status update: GLES3

- We are really close - about 600 fails on the i.MX8qm
- Support for some GLES3.1 and GLES3.2 features
- Desktop GL 3.0 is close too
- Biggest blocker is 128bit format emulation

igalia

Thanks! Questions?

Join us!

https://www.igalia.com/jobs

igalia

Status update - contemporary API

- very hacky branch that barely does anything
- Moving more code to src/etnaviv/common
- A shared image library is the next step


igalia

igalia