

etnaviv: status update

Christian Gmeiner

2023-10-19

1



Who Am I?

Long term etnaviv developer

Doing consulting work for Igalia since June 2023

```
git log --since="1 year ago" --until="now"
```

Kernel

- NPU support (Tomeu)
- new HWDB entries
 - GC520 r5341 c204 (2D GPU found on the i.MX8MP SoC)
 - GC7000 r6203 (NXP i.MX8MN SoC)
 - VIP8000 r7120 (Amlogic A311D)
 - VIP8000 Nano r8002 (NPU found on the NXP i.MX8MP SoC)
- Small fixes and improvements

Userspace

- MSAA work by Lucas and me
(ETNA_MESA_DEBUG=msaa)
- TS buffer sharing by Lucas
(ETNA_MESA_DEBUG=shared_ts)
- Performance warnings (ETNA_MESA_DEBUG=perf)
- Compiler work
 - Support negative float inline immediates
 - Improved lowerings
 - Improved linking (glsl-routing)
- Some new extensions



CI

Has proven its value

CI

- baremetal based
- zero maintenance needed

CI - current devices

- 6x GC2000 based devices (imx6q)
- 2x GC3000 based devices (imx6qp)
- 4x GC7000L based device (imx8mq)

Only a handful are available as public runners.

CI - more boardfarms

- David Heidelberg: librem5 devboards
- Sam Ravnborg: i.MX6 Solo with GC880
- Igalia HQ

With ci-tron everybody can jump in and provide some
HW.

The road towards GLES3

With some roadblocks.

The road towards GLES3

- dEQP-GLES3.functional.shaders.texture_functions.*
- Multiple Render Targets

The road towards GLES3

But then ...

```
nir: Transition away from nir_register and abs/neg/sat mo
```

happened and the problems started and I shifted my
focus.

The road towards GLES3

- The master plan to use nir_legacy failed
- Took me quite some time to get it working but shader-db is sad

```
total instructions in shared programs: 228098 -> 249817 (  
instructions in affected programs: 97892 -> 119611 (22.19  
total temps in shared programs: 84665 -> 86158 (1.76%)  
temps in affected programs: 6257 -> 7750 (23.86%)  
total immediates in shared programs: 152272 -> 152148 (-0  
immediates in affected programs: 2304 -> 2180 (-5.38%)
```

The road towards GLES3

Cloned nir_lower_to_source_mods with the following differences:

- we store the source mods in pass_flags
- we do not try to saturate the destination

```
total instructions in shared programs: 234974 -> 235376 (  
instructions in affected programs: 11481 -> 11883 (3.50%)  
total temps in shared programs: 84891 -> 84891 (0.00%)  
temps in affected programs: 0 -> 0  
total immediates in shared programs: 154776 -> 154776 (0.  
total immediates in shared programs: 154776 -> 154776 (0.
```

The road towards GLES3



Alyssa Rosenzweig @alyssa · 2 months ago

Developer



Etnaviv needs a proper backend compiler. <https://lists.freedesktop.org/archives/mesa-dev/2019-May/219103.html> fits the bill -- it 'just' needs a rebase. I will be very happy to review that once the MR is ready. Fundamentally, RA in NIR isn't practical. No other backend does it and for good reason. If the current 'compiler' were up for review today, it would likely not be merged. We can't change history but we can fix things going forward. I don't know the Vivante ISA but I suspect eir will generate better code than the current approach, beyond addressing the hit from this commit.

Digression: current compiler

- uses NIR as backend IR
- does some 64bit nir_const_value pollution

```
#define CONST_VAL(a, b) (nir_const_value) \  
    {.u64 = (uint64_t)(a) << 32 | (uint64_t)(b)}  
  
#define CONST(x) CONST_VAL(ETNA_UNIFORM_CONSTANT, x)  
#define UNIFORM(x) CONST_VAL(ETNA_UNIFORM_UNIFORM, x)
```

Digression: current compiler

- blind flight late in the process

```
/* some custom NIR transformations */  
...  
...  
  
/* call directly to avoid validation */  
nir_convert_from_ssa(shader, true);  
nir_trivialize_registers(shader);
```

Digression: current compiler

- assembler is not aware of different instruction encodings

```
case nir_intrinsic_load_global_etna: {
    struct etna_inst inst = {
        .opcode = INST_OPCODE_LOAD,
        .type = inst_type_from_bitsize(c, ...),
        .tex = {
            .amode = INST_AMODE_ADD_A_W,
        },
    };

    if (nir_src_is_const(intr->src[1])) {
        inst.src[1].amode = INST_AMODE_ADD_A_Y;
        inst.tex.swiz = 128;
    }
}
```

Digression: current compiler

- miss compilations
- quite fragile

etnaviv backend compiler



etnaviv backend compiler

- not a single line of rust
- no SSA based register allocator (vec4 ISA)
- uses isaspec
- uses etnaviv's register allocator
- has a backend IR and an optimizer
- supports control flow

etnaviv backend compiler

```
cp src/asahi/compiler src/etnaviv/compiler -r
```

with some vec4 sugar on top

etnaviv backend compiler

```
kernel void rotate(global int* out, global int* in0, global int*  
    out[get_global_id(0)] =  
        rotate(in0[get_global_id(0)], in1[get_global_id(0)]);  
}
```

etnaviv backend compiler

```
block0 {  
    3.x = add.s32 *12.x, u0.y  
    5.x = imullo0.s32 *3.x, u0.z  
    7.x = load.u32 u0.w [base address], 5.x [offset]  
    9.x = load.u32 u1.x [base address], 5.x [offset]  
    10.x = rotate.u32 *7.x, *9.x  
    store.u32 u1.y [base address], *5.x [offset], *10.x [v  
}
```

etnaviv backend compiler

```
000 add.s32.pack      t0.x____, t0.xyzw, void, u0.yyy
001 imullo0.s32.pack  t0.x____, t0.xyzw, u0.zzzz, voi
002 load.denorm.u32   t0._y__, u0.wwww, t0.xyzw, voi
003 load.denorm.u32   t0.__z_, u1.xxxx, t0.xyzw, voi
004 rotate.u32.pack   t0._y__, t0.yyyy, void, t0.zzz
005 store.denorm.u32  mem.x____, u1.yyyy, t0.xyzw, t0
```

Legacy love

Emulation with shaders

Legacy model love

- etnaviv does support a wide range of GPU modules found in different SoCs
- availability of HW features fluctuates a lot
- binary blob uses some emulation tricks to get the job done

Legacy model love

- Make use of more lowerings NIR provides (logicops, ..)
- Provide own lowerings in NIR (bordercolor)

**etnaviv is more alive than
ever**

Q&A



