WIP : Implementation of a double floating point library in GLSL 1.30

Elie TOURNIER

Google Summer of Code 2016
Head of the operation: Ian Romanick

tournier.elie@gmail.com

September 23, 2016

Who am I?

- Elie Tournier
- Google Summer of Code 2016 Student
- Graduate Software and Image Processing Engineer
- Available for hire

Goal

- Iago Toral bring GL_ARB_gpu_shader_fp64 to Intel GPUs.
- A stretch goal would be to create GL_ARB_gpu_shader_fp64 for GPUs without FP64 hardware support.
- Create a FP64 support

GLSL

- The library is written in GLSL.
- Use GLSL in order to use the graphics pipeline without having to use hardware-specific languages.
- Can be convert to GLSL IR thanks to the standalone compiler.
- Like I need to use bitwise operators, GPUs must be GLSL 1.30 (OpenGL 3.0).

Choose a CPU lib

- I don't want to reinvent the wheel.
- Convert a CPU library to a GPU one.
- Berkeley SoftFloat by John R. Hauser.
- IEEE 754 compliant.

Dev environment

- Code in my GitHub.
- Use Shader_runner from Piglit.
- Debug and test shaders.

Store FP64

Example (Code Berkley SoftFloat)

```
typedef struct {
   !!!bits32 high, low;
} float64;
```

Example (Code libSoftFloat)

```
uvec2 fp64;
```

Extract FP64 Exp

Example (Code Berkley SoftFloat)

```
INLINE int16 extractFloat64Exp( float64 a )
{

    return ( a.high>>20 ) & 0x7FF;

}
```

Example (Code libSoftFloat)

```
uint extractFloat64Exp( uvec2 a )
{

    return ( a.x>>20 ) & 0x7FFu;

}
```

Plan

- Finish to implement last FP64 operations.
- Integrate the lib to Mesa.
- Implement GL_ARB_gpu_shader_fp64.

References

John R. Hauser (2015)
Berkeley SoftFloat
U.C. Berkeley.
http://www.jhauser.us/arithmetic/SoftFloat.html

Elie Tournier (2016)
libSoftFloat
https://github.com/Hopetech/libSoftFloat

Thanks.