
Subject: : AmigaOS4

Topic: : GL4ES: another OpenGL over OpenGL ES2 emulation - some tech. info and porting progress

Re: GL4ES: another OpenGL over OpenGL ES2 emulation - some tech. info and porting progress

Author: : kas1e

Date: : 2019/9/20 20:23:22

URL:

@All

While porting supertuxkart, found another simple shader which fail on our side. Daniel's glslangvalidator_redux compile them fine (so probably ogles2 also fine there, because code from daniel's last glslangvalidator_redux are inside of ogles2 if i correctly remember), but then, compiled .spv version from that shader fail on Nova with reference to problems with shared_ptr

There is shader in question:

```
#version 120
```

```
// motion_blur.vert
```

```
void main()
```

```
{  
  gl_TexCoord[0].st = vec2(gl_MultiTexCoord0.s, gl_MultiTexCoord0.t);  
  gl_Position = gl_Vertex;  
}
```

glslangvalidator compile it fine:

Quote:

```
8/0.Work:Warp3DNova/my_tests> glslangvalidator_redux -G -o motion_blur.vert.spv motion_blur.vert
```

```
Success.
```

Then nova fail to compile it:

```
8/0.Work:Warp3DNova/my_tests> W3DNShaderInfo motion_blur.vert.spv
```

```
W3DNShaderInfo - Get shader information
```

Shader: motion_blur.vert.spv

Compiling motion_blur.vert.spv failed (12) with error: unknown error

Log:

ERROR: An exception occurred during compilation:

ERROR: Assertion failed: px != 0, in T* boost::shared_ptr<T>::operator->() const [with T = GPUProg::CGRegisterAlloc], defined in /SDK/local/common/include/boost/shared_ptr.hpp in line 253

ERROR: An exception occurred during compilation:

ERROR: Assertion failed: px != 0, in T* boost::shared_ptr<T>::operator->() const [with T = GPUProg::CGRegisterAlloc], defined in /SDK/local/common/include/boost/shared_ptr.hpp in line 253

ERROR: Code generation failed for an unknown reason.

Done.

And there is verbose output from nova:

```
8/0.Work:Warp3DNova/my_tests> W3DNShaderInfo -v motion_blur.vert.spv
```

```
W3DNShaderInfo - Get shader information
```

```
Verbose mode
```

```
Shader: motion_blur.vert.spv
```

```
Compiling motion_blur.vert.spv failed (12) with error: unknown error
```

```
Log:
```

```
Shader size: 760 bytes
```

```
Parsing SPIR-V code
```

```
Module Version: 1.2.0
```

```
Generator Magic Number: 0x80003
```

```
Upper bound on ids: 32
```

```
Parsed instructions:
```

```
OpCapability: : Shader
```

```
1: OpExtInstImport: : GLSL.std.450
```

```
OpMemoryModel: : addressing: Logical, memory: GLSL450
```

```
4: OpEntryPoint: : main, execution model: Vertex
```

```
OpSource: : GLSL ver 120
```

```
4: OpName: : main
```

```
12: OpName: : gl_TexCoord
```

```
16: OpName: : gl_MultiTexCoord0
```

```
29: OpName: : gl_Position
```

```
30: OpName: : gl_Vertex
```

```
29: OpDecorate: : BuiltIn(Position)
```

```
2: OpTypeVoid: Void
```

```
3: OpTypeFunction: 2 << func()
```

```
6: OpTypeFloat: Float: 32 bits
```

```
7: OpTypeVector: ???Vector4: num-elements: 4, element type id: 6
```

```
8: OpTypeInt: UInt: 32 bits, unsigned
```

```

9: OpConstant: 8 const9 = 0x1
10: OpTypeArray: ???[]: length id: 9, element type id: 7
11: OpTypePointer: ???Ptr: storage class: Output
12: OpVariable: 11: var12: storage class: Output
13: OpTypeInt: Int: 32 bits, signed
14: OpConstant: 13 const14 = 0x0
15: OpTypePointer: ???Ptr: storage class: Input
16: OpVariable: 15: var16: storage class: Input
17: OpConstant: 8 const17 = 0x0
18: OpTypePointer: ???Ptr: storage class: Input
23: OpTypeVector: ???Vector2: num-elements: 2, element type id: 6
25: OpTypePointer: ???Ptr: storage class: Output
29: OpVariable: 25: var29: storage class: Output
30: OpVariable: 15: var30: storage class: Input
4: OpFunction: func4(type: 3)

5: OpLabel:
19: OpAccessChain: 18: 16[17]
20: OpLoad: 6: tmp20 << 19
21: OpAccessChain: 18: 16[9]
22: OpLoad: 6: tmp22 << 21
24: OpCompositeConstruct: 23: tmp24 << 20, 22
26: OpAccessChain: 25: 12[14]
27: OpLoad: 7: tmp27 << 26
28: OpVectorShuffle: 7: tmp28 << 27, 24, 4, 5, 2, 3
    OpStore: : 28 >> 26
31: OpLoad: 7: tmp31 << 30
    OpStore: : 31 >> 29
    OpReturn:
    OpFunctionEnd:

```

Linking the instructions

Initial Disassembly:

Module Info:

```

OpSource: : GLSL ver 120
1: OpExtInstImport: : GLSL.std.450
OpMemoryModel: : addressing: Logical, memory: GLSL450

```

Capabilities:

```
OpCapability: : Shader
```

Inputs:

```

16: OpVariable: FloatVector4*: gl_MultiTexCoord0: storage class: Input
30: OpVariable: FloatVector4*: gl_Vertex: storage class: Input

```

Outputs:

```

12: OpVariable: FloatVector4[1]*: gl_TexCoord: storage class: Output
29: OpVariable: FloatVector4*: gl_Position: storage class: Output Decorators: BuiltIn(Position)

```

Entry Points:

```
4: OpEntryPoint: : main, execution model: Vertex, Function: Void main()
```

Constants:

9: OpConstant: UInt const9 = 1
14: OpConstant: Int const14 = 0
17: OpConstant: UInt const17 = 0

Disassembled Code:

4: OpFunction: Void main()
5: lb5:
19: OpAccessChain: Float*: gl_MultiTexCoord0[0]
20: OpLoad: Float: tmp20 << gl_MultiTexCoord0[0]
21: OpAccessChain: Float*: gl_MultiTexCoord0[1]
22: OpLoad: Float: tmp22 << gl_MultiTexCoord0[1]
24: OpCompositeConstruct: FloatVector2: tmp24 << tmp20, tmp22
26: OpAccessChain: FloatVector4*: gl_TexCoord[0]
27: OpLoad: FloatVector4: tmp27 << gl_TexCoord[0]
28: OpVectorShuffle: FloatVector4: tmp28 << tmp27, tmp24, 4, 5, 2, 3
OpStore: : tmp28 >> gl_TexCoord[0]
31: OpLoad: FloatVector4: tmp31 << gl_Vertex
OpStore: : tmp31 >> gl_Position
OpReturn:

Performing hardware-independent optimization...

Can't merge stores to array variable: gl_TexCoord
Optimization done.

Optimized Disassembly:

Module Info:

OpSource: : GLSL ver 120
1: OpExtInstImport: : GLSL.std.450
OpMemoryModel: : addressing: Logical, memory: GLSL450

Capabilities:

OpCapability: : Shader

Inputs:

16: OpVariable: FloatVector4*: gl_MultiTexCoord0: storage class: Input
30: OpVariable: FloatVector4*: gl_Vertex: storage class: Input

Outputs:

12: OpVariable: FloatVector4[1]*: gl_TexCoord: storage class: Output
29: OpVariable: FloatVector4*: gl_Position: storage class: Output Decorators: BuiltIn(Position)

Entry Points:

4: OpEntryPoint: : main, execution model: Vertex, Function: Void main()

Constants:

9: OpConstant: UInt const9 = 1
14: OpConstant: Int const14 = 0

17: OpConstant: UInt const17 = 0

Disassembled Code:

4: OpFunction: Void main()

5: lb5:

19: OpAccessChain: Float*: gl_MultiTexCoord0[0]
20: OpLoad: Float: tmp20 << gl_MultiTexCoord0[0]
21: OpAccessChain: Float*: gl_MultiTexCoord0[1]
22: OpLoad: Float: tmp22 << gl_MultiTexCoord0[1]
24: OpCompositeConstruct: FloatVector2: tmp24 << tmp20, tmp22
26: OpAccessChain: FloatVector4*: gl_TexCoord[0]
27: OpLoad: FloatVector4: tmp27 << gl_TexCoord[0]
28: OpVectorShuffle: FloatVector4: tmp28 << tmp27, tmp24, 4, 5, 2, 3
OpStore: : tmp28 >> gl_TexCoord[0]
31: OpLoad: FloatVector4: tmp31 << gl_Vertex
OpStore: : tmp31 >> gl_Position
OpReturn:

Generating the compiled code...

ERROR: An exception occurred during compilation:

ERROR: Assertion failed: px != 0, in T* boost::shared_ptr<T>::operator->() const [with T = GPUProg::CGRegisterAlloc], defined in /SDK/local/common/include/boost/shared_ptr.hpp in line 253

Intermediate disassembly (pre optimization):

Program Type: Vertex

Input Variables:

offset: 0, size: 16, FloatVector4 gl_Vertex

offset: 1, size: 16, FloatVector4 gl_MultiTexCoord0

Output Variables:

offset: 32, size: 16, FloatVector4 gl_TexCoord[1]

Special Output Variables:

offset: 12, size: 16, FloatVector4 gl_Position BuiltIn(Position)

Constants:

UInt32 const9: 1

Int32 const14: 0

UInt32 const17: 0

Instructions:

V_ADD_I32 vDst(VGPR0) src0(SGPR2) src1(VGPR0) // VOP2

Void main()

Function: Void main()

lb5

Label: lb5

19: OpAccessChain: Float*: gl_MultiTexCoord0[0]

20: OpLoad: Float: tmp20 << gl_MultiTexCoord0[0]

S_LOAD_DWORDX4_IMM offset(4) sBase(SGPR[6:7]) sDst(SGPR[8:11])

S_WAITCNT 0

```
BUFFER_LOAD_FORMAT_X offset(0) offEn(0) idxEn(1) glc(0) addr64(0) lds(0) vAddr(VGPR[0:1]) vData(VGP
R2) srSrc(SGPR[8:11]) slc(0) tfe(0) sOffset(0)
S_WAITCNT 0
# 21: OpAccessChain: Float*: gl_MultiTexCoord0[1]
# 22: OpLoad: Float: tmp22 << gl_MultiTexCoord0[1]
S_LOAD_DWORDX4_IMM offset(4) sBase(SGPR[6:7]) sDst(SGPR[12:15])
S_WAITCNT 0
BUFFER_LOAD_FORMAT_XYZW offset(0) offEn(0) idxEn(1) glc(0) addr64(0) lds(0) vAddr(VGPR[0:1]) vData(
VGPR4) srSrc(SGPR[12:15]) slc(0) tfe(0) sOffset(0)
S_WAITCNT 0
V_MOV_B32 vDst(VGPR3) src0(VGPR5)
# 24: OpCompositeConstruct: FloatVector2: tmp24 << tmp20, tmp22
V_MOV_B32 vDst(VGPR8) src0(VGPR2)
V_MOV_B32 vDst(VGPR9) src0(VGPR3)
# 26: OpAccessChain: FloatVector4*: gl_TexCoord[0]
# 27: OpLoad: FloatVector4: tmp27 << gl_TexCoord[0]
```

Performing GPU-specific optimization...

Pre register allocation control-flow processing...

Intermediate disassembly (pre register allocation):

Program Type: Vertex

Input Variables:

offset: 0, size: 16, FloatVector4 gl_Vertex

offset: 1, size: 16, FloatVector4 gl_MultiTexCoord0

Output Variables:

offset: 32, size: 16, FloatVector4 gl_TexCoord[1]

Special Output Variables:

offset: 12, size: 16, FloatVector4 gl_Position BuiltIn(Position)

Constants:

UInt32 const9: 1

Int32 const14: 0

UInt32 const17: 0

Instructions:

```
V_ADD_I32 vDst(VGPR0) src0(SGPR2) src1(VGPR0) // VOP2
```

```
# Void main()
```

```
Function: Void main()
```

```
# lb5
```

```
Label: lb5
```

```
# 19: OpAccessChain: Float*: gl_MultiTexCoord0[0]
```

```
# 20: OpLoad: Float: tmp20 << gl_MultiTexCoord0[0]
```

```
S_LOAD_DWORDX4_IMM offset(4) sBase(SGPR[6:7]) sDst(SGPR[8:11])
```

```
S_WAITCNT 0
```

```
BUFFER_LOAD_FORMAT_X offset(0) offEn(0) idxEn(1) glc(0) addr64(0) lds(0) vAddr(VGPR[0:1]) vData(VGP
```

```
R2) srSrc(SGPR[8:11]) slc(0) tfe(0) sOffset(0)
```

```
S_WAITCNT 0
```

```
# 21: OpAccessChain: Float*: gl_MultiTexCoord0[1]
```

```
# 22: OpLoad: Float: tmp22 << gl_MultiTexCoord0[1]
```

```
S_LOAD_DWORDX4_IMM offset(4) sBase(SGPR[6:7]) sDst(SGPR[12:15])
S_WAITCNT 0
BUFFER_LOAD_FORMAT_XYZW offset(0) offEn(0) idxEn(1) glc(0) addr64(0) lds(0) vAddr(VGPR[0:1]) vData(
VGPR4) srSrc(SGPR[12:15]) slc(0) tfe(0) sOffset(0)
S_WAITCNT 0
V_MOV_B32 vDst(VGPR3) src0(VGPR5)
#      24: OpCompositeConstruct: FloatVector2: tmp24 << tmp20, tmp22
V_MOV_B32 vDst(VGPR8) src0(VGPR2)
V_MOV_B32 vDst(VGPR9) src0(VGPR3)
#      26: OpAccessChain: FloatVector4*: gl_TexCoord[0]
#      27: OpLoad: FloatVector4: tmp27 << gl_TexCoord[0]
ERROR: An exception occurred during compilation:
ERROR: Assertion failed: px != 0, in T* boost::shared_ptr<T>::operator->() const [with T = GPUProg::CGRegist
erAlloc], defined in /SDK/local/common/include/boost/shared_ptr.hpp in line 253
```

Done.

8/0.Work:Warp3DNova/my_tests>

I do some google, and found that this variable used pretty often in all tutorials and even in book "OpenGL Shading language".

Then i find that it all described in section 5.5 of GLSL spec, and in end i find in google some explain , that this "st" part is part of swizzle mask which let you recombine your vector. The texture coordinates are four-component vectors, but st mask selects the first two (you can use "xy", it would be the same).

So i tried also with "xy", but it also fail.

Any ideas if it expected to work (so its a bug), or some non-implemented feature ?