Mesh Shaders on NVK

Mary Guillemard

XDC 2024

About me

- Mary Guillemard (fedi: @mary@chaos.social)
- At Collabora since August 2023
  - Work on NVK and Panfrost (PanVK, gallium, bifrost...)
- Behind...
  - Geometry Shader/Barycentric/FP16 for NVK with NAK
  - Research tooling for NVK
  - VK_EXT_graphics_pipeline_library and lot of fixes on PanVK
  - Initial Rusticl support on Panfrost for Valhall Gen 2
  - ...

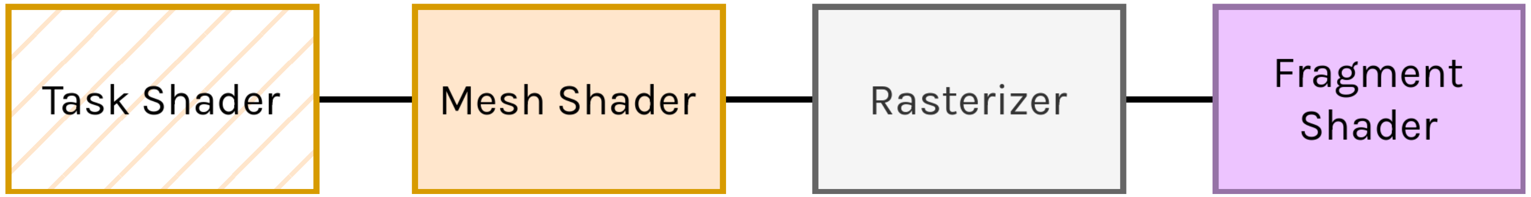
COLLABORA

What are mesh shaders?

What are mesh shaders?

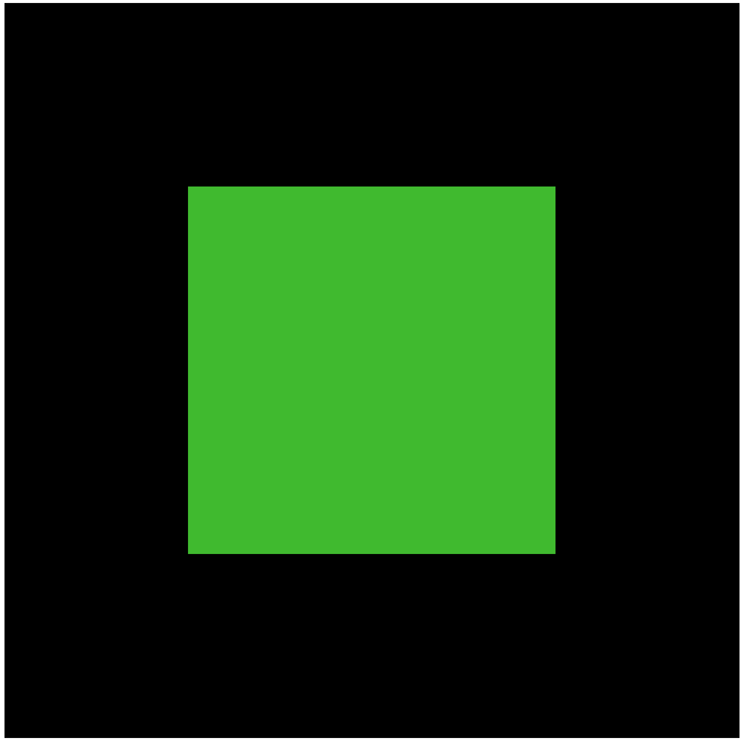
- New type of graphics pipeline with simplified stages
- Geometry generated by the mesh stage
- Closer to the compute model
- Optionally, the task stage can define the grid size of subsequent mesh shader workgroup

What are mesh shaders?



What are mesh shaders?

```glsl
#version 450
#extension GL_EXT_mesh_shader : require

layout(local_size_x = 1, local_size_y = 1, local_size_z = 1) in;
layout(points, max_vertices = 1, max_primitives = 1) out;

void main()
{
    // We are going to generate one vertice and one primitive.
    SetMeshOutputsEXT(1, 1);

    // Set the point size.
    gl_MeshVerticesEXT[0].gl_PointSize = 200.0f;

    // Set the position of the point.
    gl_MeshVerticesEXT[0].gl_Position = vec4(0.0, 0.0, 0.0, 1.0);

    // Finally set the point vertice indice.
    gl_PrimitivePointIndicesEXT[0] = 0;
}

```



COLLABORA

Open First

COLLABORA

How to draw on NVIDIA hardware?

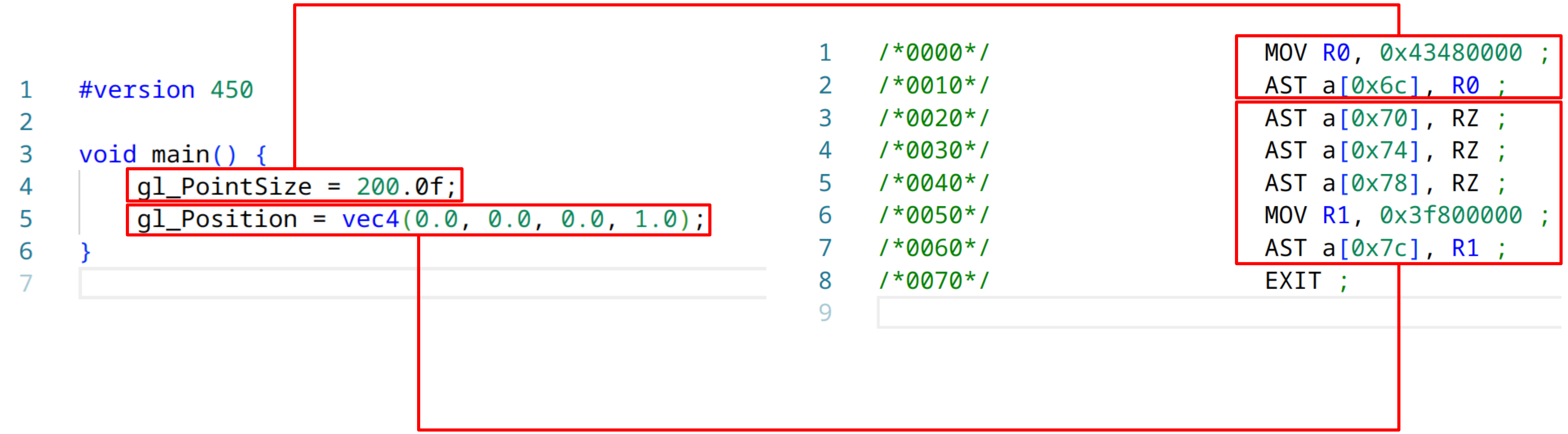
In the command stream...

- Bind resources (constant buffers, attributes,...)
- Bind shader stage programs
- Set DRAW_CONTROL (topology, provoking vertex,...)
- Call DRAW_VERTEX_ARRAY

In the shader program…

- Each shader program have a header (attributes I / O)
- Attribute access in the code is done with ALD / AST
- Each attributes have an unique identifier

In the shader program…



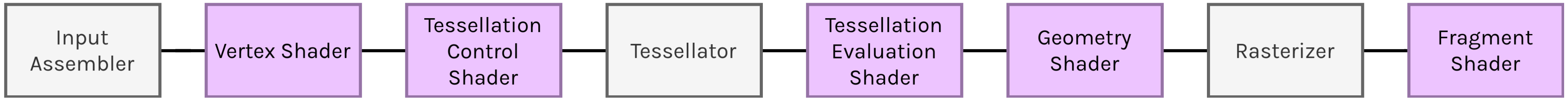
```
#version 450

void main() {
    gl_PointSize = 200.0f;
    gl_Position = vec4(0.0, 0.0, 0.0, 1.0);
}
```

```
/*0000*/                MOV R0, 0x43480000 ;
/*0010*/                AST a[0x6c], R0 ;
/*0020*/                AST a[0x70], RZ ;
/*0030*/                AST a[0x74], RZ ;
/*0040*/                AST a[0x78], RZ ;
/*0050*/                MOV R1, 0x3f800000 ;
/*0060*/                AST a[0x7c], R1 ;
/*0070*/                EXIT ;
```

COLLABORA

Open First

How to draw on NVIDIA hardware?



~1:1 mapping of the traditional pipeline

COLLABORA

How mesh shaders works on NVIDIA?

In the command stream...

In the command stream…

- Bind resources (constant buffers, textures,…)

In the command stream…

- Bind resources (constant buffers, textures,…)
- Bind shader stage programs (vertex?, tessellation??, geometry???)

In the command stream...

- Bind resources (constant buffers, textures,...)
- Bind shader stage programs (vertex?, tessellation??, geometry???)
- Set DRAW_CONTROL

In the command stream…

- Bind resources (constant buffers, textures,…)
- Bind shader stage programs (vertex?, tessellation??, geometry???)
- Set DRAW_CONTROL
- Call DRAW_VERTEX_ARRAY

In the command stream...

- Bind resources (constant buffers, textures,...)
- Bind shader stage programs (vertex?, tessellation??, geometry???)
- Set DRAW_CONTROL
- Call DRAW_VERTEX_ARRAY

... What is going on here?

In the command stream…

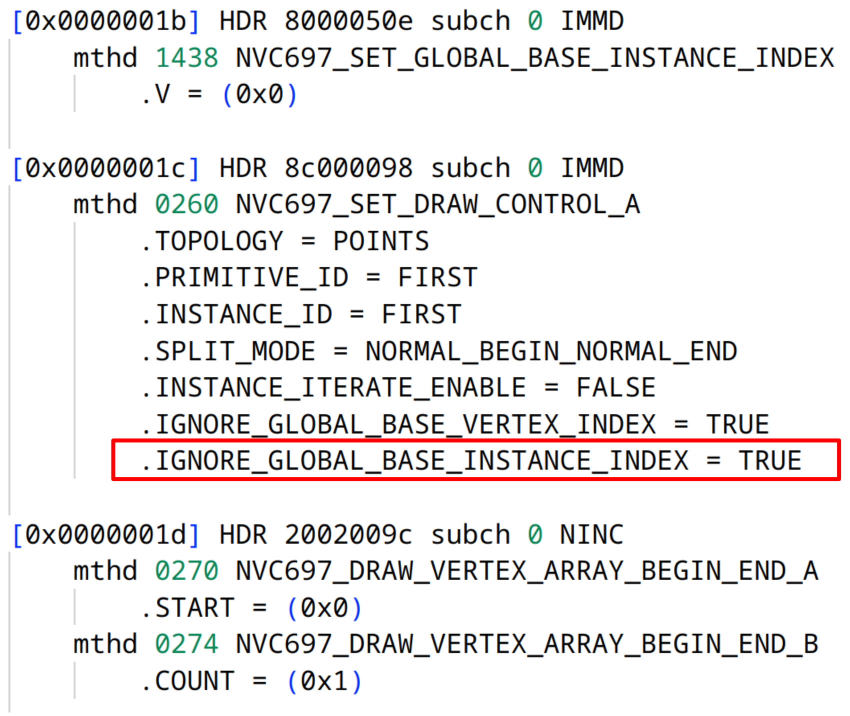
```
[0x0000000b] HDR 80010453 subch 0 IMMD
    mthd 114c NVC697_SET_MESH_CONTROL
        .ENABLE = TRUE

[0x0000000c] HDR 20020454 subch 0 NINC
    mthd 1150 NVC697_SET_MESH_SHADER_A
        .OUTPUT_TOPOLOGY = POINTS
        .MAX_VERTEX = (0x1)
        .MAX_PRIMITIVE = (0x1)
    mthd 1154 NVC697_SET_MESH_SHADER_B
        .SHARED_MEM_LINES = (0x0)
        .THREAD_COUNT = (0x1)
```

- After binding stages, set mesh control methods

COLLABORA

Open First

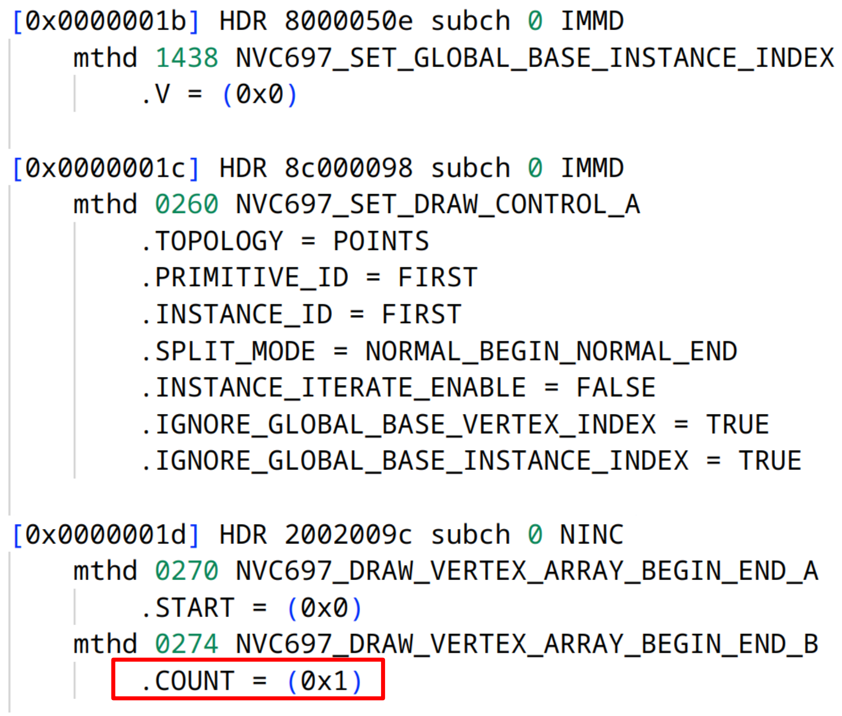
In the command stream...



```
[0x0000001b] HDR 8000050e subch 0 IMMD
    mthd 1438 NVC697_SET_GLOBAL_BASE_INSTANCE_INDEX
        .V = (0x0)

[0x0000001c] HDR 8c000098 subch 0 IMMD
    mthd 0260 NVC697_SET_DRAW_CONTROL_A
        .TOPOLOGY = POINTS
        .PRIMITIVE_ID = FIRST
        .INSTANCE_ID = FIRST
        .SPLIT_MODE = NORMAL_BEGIN_NORMAL_END
        .INSTANCE_ITERATE_ENABLE = FALSE
        .IGNORE_GLOBAL_BASE_VERTEX_INDEX = TRUE
        .IGNORE_GLOBAL_BASE_INSTANCE_INDEX = TRUE

[0x0000001d] HDR 2002009c subch 0 NINC
    mthd 0270 NVC697_DRAW_VERTEX_ARRAY_BEGIN_END_A
        .START = (0x0)
    mthd 0274 NVC697_DRAW_VERTEX_ARRAY_BEGIN_END_B
        .COUNT = (0x1)
```

- DRAW_CONTROL disable global instance index

In the command stream...



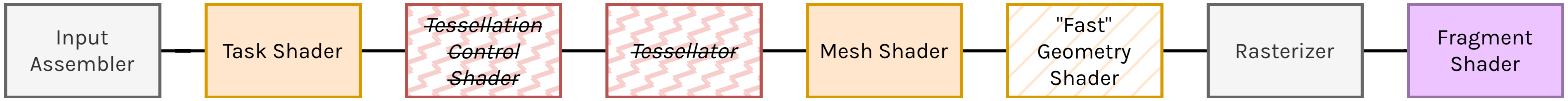
```
[0x0000001b] HDR 8000050e subch 0 IMMD
    mthd 1438 NVC697_SET_GLOBAL_BASE_INSTANCE_INDEX
        .V = (0x0)

[0x0000001c] HDR 8c000098 subch 0 IMMD
    mthd 0260 NVC697_SET_DRAW_CONTROL_A
        .TOPOLOGY = POINTS
        .PRIMITIVE_ID = FIRST
        .INSTANCE_ID = FIRST
        .SPLIT_MODE = NORMAL_BEGIN_NORMAL_END
        .INSTANCE_ITERATE_ENABLE = FALSE
        .IGNORE_GLOBAL_BASE_VERTEX_INDEX = TRUE
        .IGNORE_GLOBAL_BASE_INSTANCE_INDEX = TRUE

[0x0000001d] HDR 2002009c subch 0 NINC
    mthd 0270 NVC697_DRAW_VERTEX_ARRAY_BEGIN_END_A
        .START = (0x0)
    mthd 0274 NVC697_DRAW_VERTEX_ARRAY_BEGIN_END_B
        .COUNT = (0x1)
```

- DRAW_CONTROL disable global instance index
- DRAW_VERTEX_ARRAY is called with the dispatch workgroup size

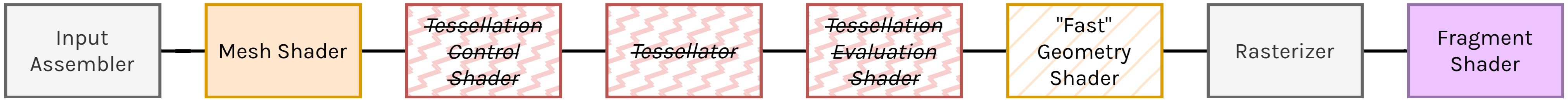
COLLABORA

Open First

Mesh pipeline the NVIDIA way



With Task shader



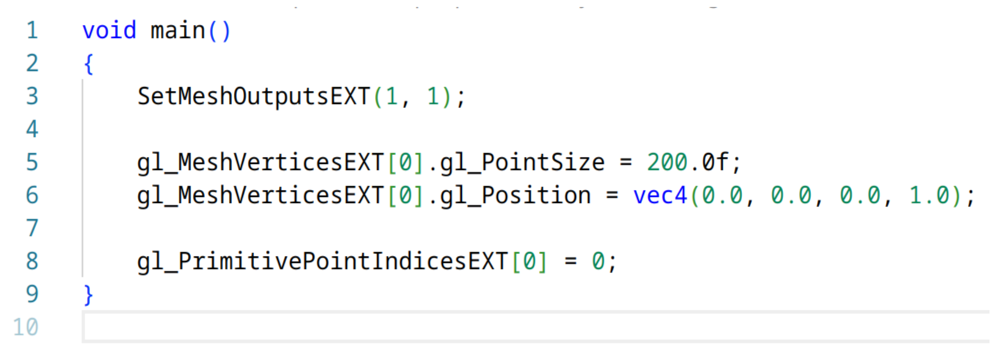
Without Task shader

In the shader program…

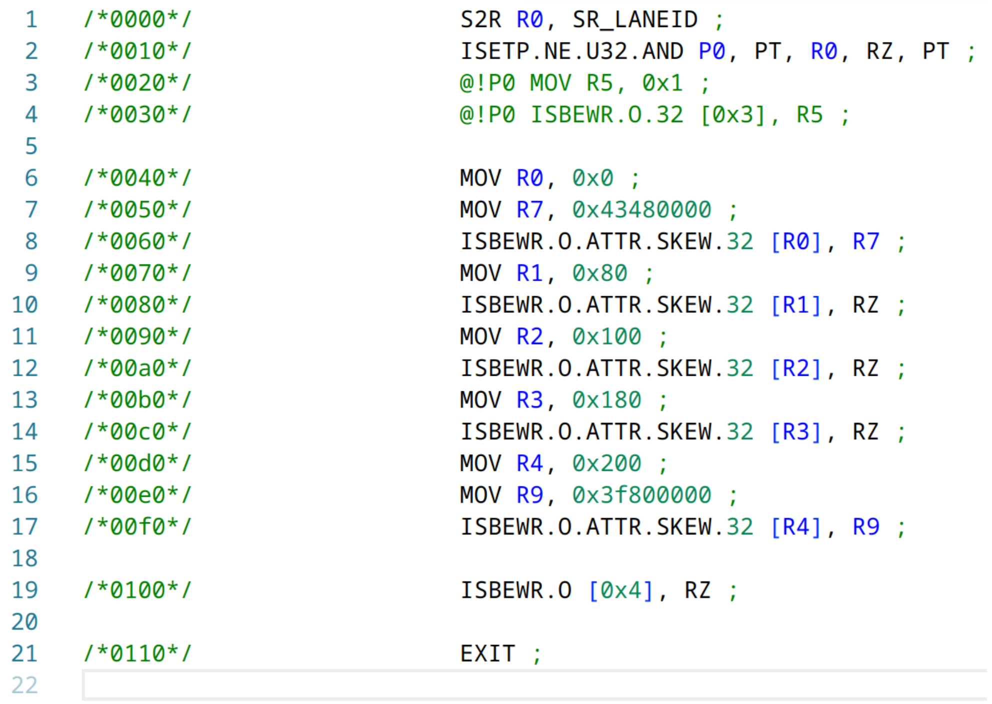
```
void main()
{
    SetMeshOutputsEXT(1, 1);

    gl_MeshVerticesEXT[0].gl_PointSize = 200.0f;
    gl_MeshVerticesEXT[0].gl_Position = vec4(0.0, 0.0, 0.0, 1.0);

    gl_PrimitivePointIndicesEXT[0] = 0;
}
```

```
/*0000*/                      S2R R0, SR_LANEID ;
/*0010*/                      ISETP.NE.U32.AND P0, PT, R0, RZ, PT ;
/*0020*/                      @!P0 MOV R5, 0x1 ;
/*0030*/                      @!P0 ISBEWR.O.32 [0x3], R5 ;

/*0040*/                      MOV R0, 0x0 ;
/*0050*/                      MOV R7, 0x43480000 ;
/*0060*/                      ISBEWR.O.ATTR.SKEW.32 [R0], R7 ;
/*0070*/                      MOV R1, 0x80 ;
/*0080*/                      ISBEWR.O.ATTR.SKEW.32 [R1], RZ ;
/*0090*/                      MOV R2, 0x100 ;
/*00a0*/                      ISBEWR.O.ATTR.SKEW.32 [R2], RZ ;
/*00b0*/                      MOV R3, 0x180 ;
/*00c0*/                      ISBEWR.O.ATTR.SKEW.32 [R3], RZ ;
/*00d0*/                      MOV R4, 0x200 ;
/*00e0*/                      MOV R9, 0x3f800000 ;
/*00f0*/                      ISBEWR.O.ATTR.SKEW.32 [R4], R9 ;

/*0100*/                      ISBEWR.O [0x4], RZ ;

/*0110*/                      EXIT ;
```

In the shader program...



```
void main()
{
    SetMeshOutputsEXT(1, 1);

    gl_MeshVerticesEXT[0].gl_PointSize = 200.0f;
    gl_MeshVerticesEXT[0].gl_Position = vec4(0.0, 0.0, 0.0, 1.0);

    gl_PrimitivePointIndicesEXT[0] = 0;
}
```



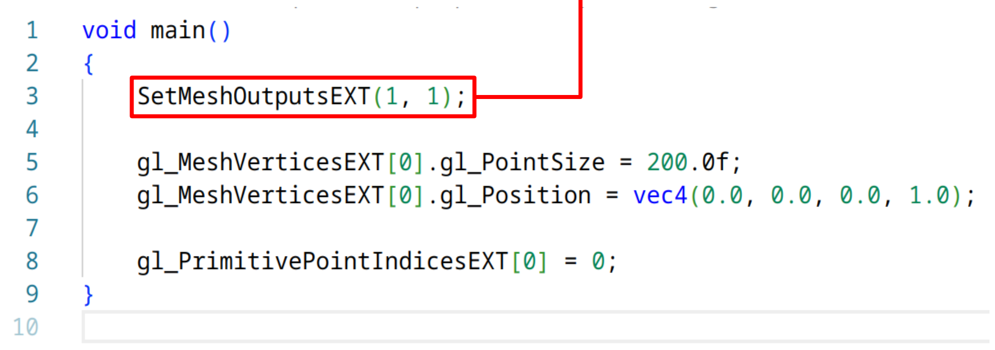
```
/*0000*/                   S2R R0, SR_LANEID ;
/*0010*/                   ISETP.NE.U32.AND P0, PT, R0, RZ, PT ;
/*0020*/                   @!P0 MOV R5, 0x1 ;
/*0030*/                   @!P0 ISBEWR.O.32 [0x3], R5 ;

/*0040*/                   MOV R0, 0x0 ;
/*0050*/                   MOV R7, 0x43480000 ;
/*0060*/                   ISBEWR.O.ATTR.SKEW.32 [R0], R7 ;
/*0070*/                   MOV R1, 0x80 ;
/*0080*/                   ISBEWR.O.ATTR.SKEW.32 [R1], RZ ;
/*0090*/                   MOV R2, 0x100 ;
/*00a0*/                   ISBEWR.O.ATTR.SKEW.32 [R2], RZ ;
/*00b0*/                   MOV R3, 0x180 ;
/*00c0*/                   ISBEWR.O.ATTR.SKEW.32 [R3], RZ ;
/*00d0*/                   MOV R4, 0x200 ;
/*00e0*/                   MOV R9, 0x3f800000 ;
/*00f0*/                   ISBEWR.O.ATTR.SKEW.32 [R4], R9 ;

/*0100*/                   ISBEWR.O [0x4], RZ ;

/*0110*/                   EXIT ;
```

- New shiny instruction: ISBEWR ✨

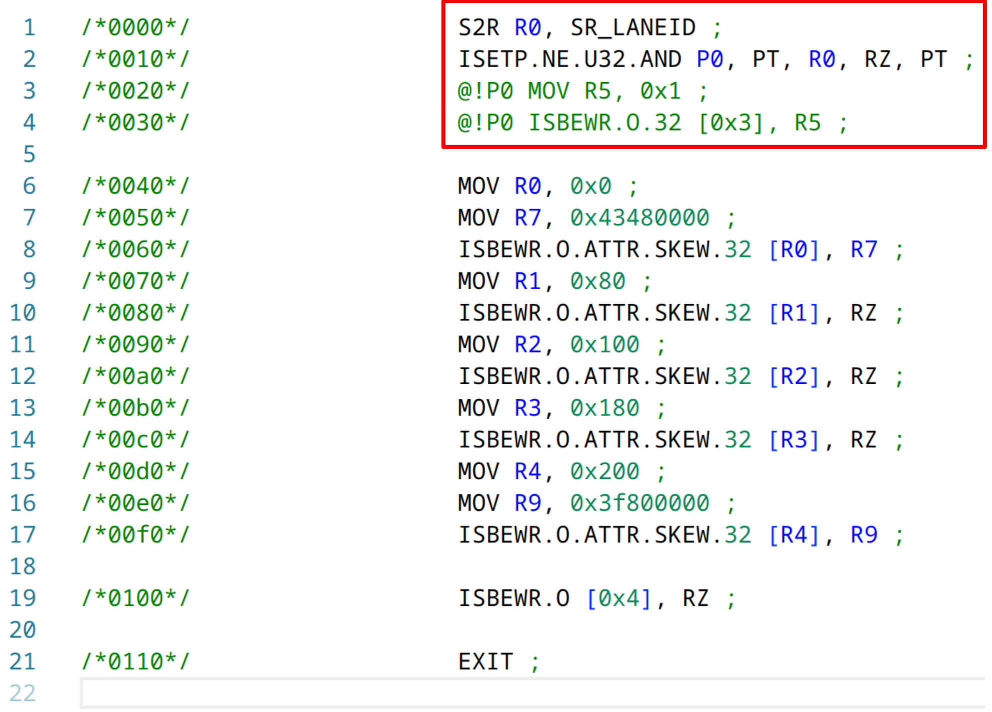
COLLABORA

Open First

In the shader program…



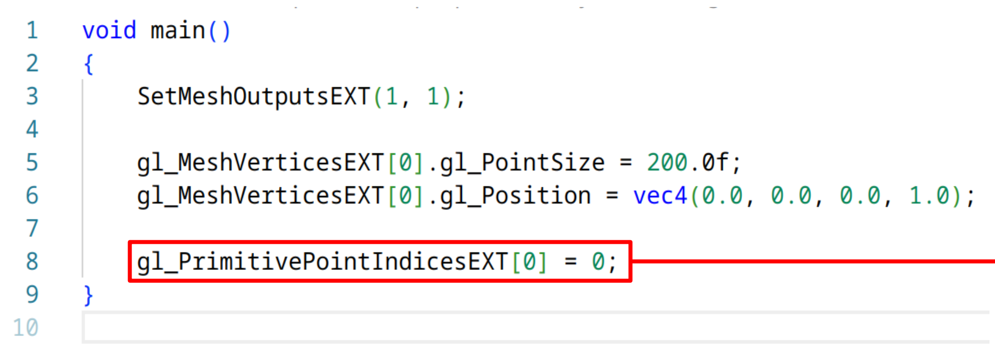
```glsl
void main()
{
    SetMeshOutputsEXT(1, 1);

    gl_MeshVerticesEXT[0].gl_PointSize = 200.0f;
    gl_MeshVerticesEXT[0].gl_Position = vec4(0.0, 0.0, 0.0, 1.0);

    gl_PrimitivePointIndicesEXT[0] = 0;
}
```



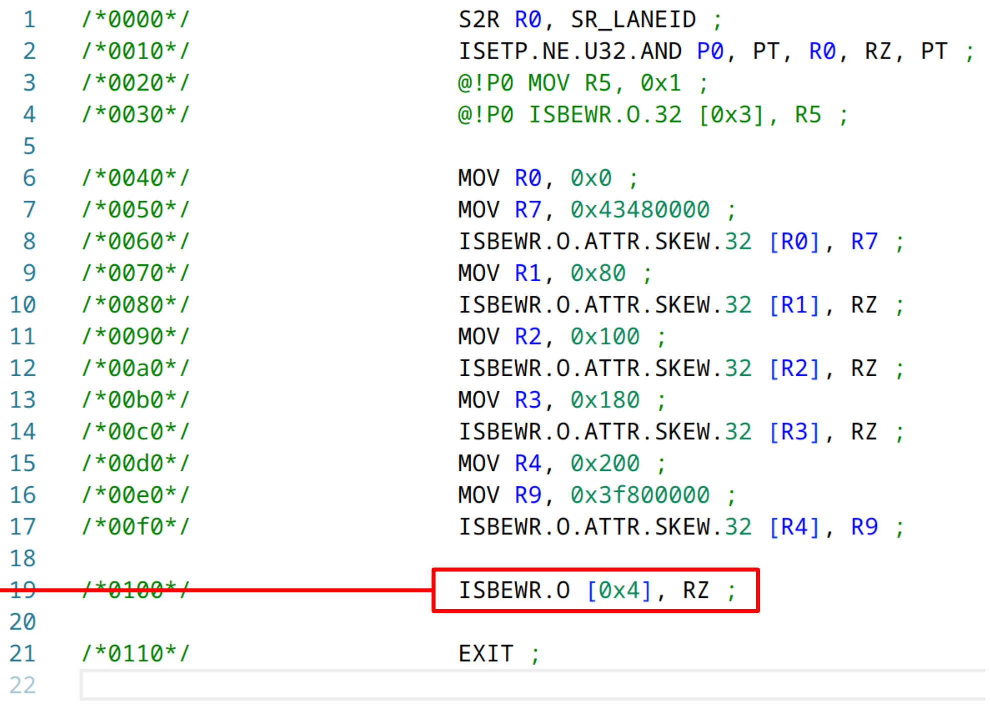
```
/*0000*/                    S2R R0, SR_LANEID ;
/*0010*/                    ISETP.NE.U32.AND P0, PT, R0, RZ, PT ;
/*0020*/                    @!P0 MOV R5, 0x1 ;
/*0030*/                    @!P0 ISBEWR.O.32 [0x3], R5 ;

/*0040*/                    MOV R0, 0x0 ;
/*0050*/                    MOV R7, 0x43480000 ;
/*0060*/                    ISBEWR.O.ATTR.SKEW.32 [R0], R7 ;
/*0070*/                    MOV R1, 0x80 ;
/*0080*/                    ISBEWR.O.ATTR.SKEW.32 [R1], RZ ;
/*0090*/                    MOV R2, 0x100 ;
/*00a0*/                    ISBEWR.O.ATTR.SKEW.32 [R2], RZ ;
/*00b0*/                    MOV R3, 0x180 ;
/*00c0*/                    ISBEWR.O.ATTR.SKEW.32 [R3], RZ ;
/*00d0*/                    MOV R4, 0x200 ;
/*00e0*/                    MOV R9, 0x3f800000 ;
/*00f0*/                    ISBEWR.O.ATTR.SKEW.32 [R4], R9 ;

/*0100*/                    ISBEWR.O [0x4], RZ ;

/*0110*/                    EXIT ;
```

- New shiny instruction: ISBEWR ✨
- Primitive count stored at offset 0x0

In the shader program...



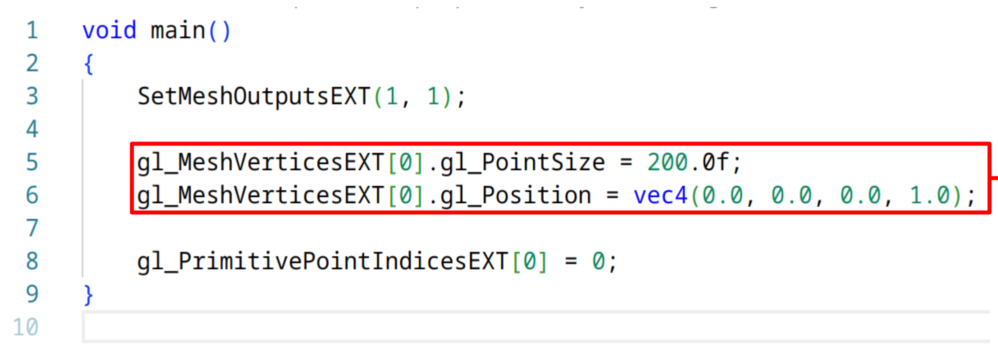
```
void main()
{
    SetMeshOutputsEXT(1, 1);

    gl_MeshVerticesEXT[0].gl_PointSize = 200.0f;
    gl_MeshVerticesEXT[0].gl_Position = vec4(0.0, 0.0, 0.0, 1.0);

    gl_PrimitivePointIndicesEXT[0] = 0;
}
```



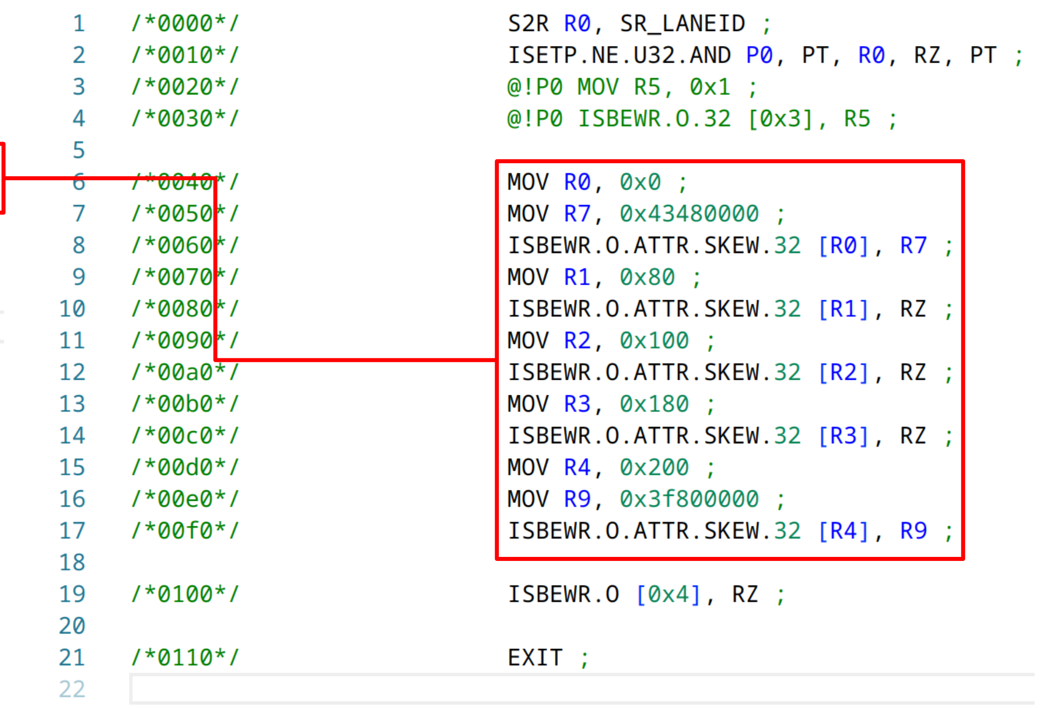
```
/*0000*/                    S2R R0, SR_LANEID ;
/*0010*/                    ISETP.NE.U32.AND P0, PT, R0, RZ, PT ;
/*0020*/                    @!P0 MOV R5, 0x1 ;
/*0030*/                    @!P0 ISBEWR.O.32 [0x3], R5 ;

/*0040*/                    MOV R0, 0x0 ;
/*0050*/                    MOV R7, 0x43480000 ;
/*0060*/                    ISBEWR.O.ATTR.SKEW.32 [R0], R7 ;
/*0070*/                    MOV R1, 0x80 ;
/*0080*/                    ISBEWR.O.ATTR.SKEW.32 [R1], RZ ;
/*0090*/                    MOV R2, 0x100 ;
/*00a0*/                    ISBEWR.O.ATTR.SKEW.32 [R2], RZ ;
/*00b0*/                    MOV R3, 0x180 ;
/*00c0*/                    ISBEWR.O.ATTR.SKEW.32 [R3], RZ ;
/*00d0*/                    MOV R4, 0x200 ;
/*00e0*/                    MOV R9, 0x3f800000 ;
/*00f0*/                    ISBEWR.O.ATTR.SKEW.32 [R4], R9 ;

/*0100*/                    ISBEWR.O [0x4], RZ ;

/*0110*/                    EXIT ;
```

- New shiny instruction: ISBEWR ✨
- Primitive count stored at offset 0x0
- Primitive indices stored starting at offset 0x4

In the shader program...



```
void main()
{
    SetMeshOutputsEXT(1, 1);

    gl_MeshVerticesEXT[0].gl_PointSize = 200.0f;
    gl_MeshVerticesEXT[0].gl_Position = vec4(0.0, 0.0, 0.0, 1.0);

    gl_PrimitivePointIndicesEXT[0] = 0;
}
```



```
/*0000*/                   S2R R0, SR_LANEID ;
/*0010*/                   ISETP.NE.U32.AND P0, PT, R0, RZ, PT ;
/*0020*/                   @!P0 MOV R5, 0x1 ;
/*0030*/                   @!P0 ISBEWR.O.32 [0x3], R5 ;

/*0040*/                   MOV R0, 0x0 ;
/*0050*/                   MOV R7, 0x43480000 ;
/*0060*/                   ISBEWR.O.ATTR.SKEW.32 [R0], R7 ;
/*0070*/                   MOV R1, 0x80 ;
/*0080*/                   ISBEWR.O.ATTR.SKEW.32 [R1], RZ ;
/*0090*/                   MOV R2, 0x100 ;
/*00a0*/                   ISBEWR.O.ATTR.SKEW.32 [R2], RZ ;
/*00b0*/                   MOV R3, 0x180 ;
/*00c0*/                   ISBEWR.O.ATTR.SKEW.32 [R3], RZ ;
/*00d0*/                   MOV R4, 0x200 ;
/*00e0*/                   MOV R9, 0x3f800000 ;
/*00f0*/                   ISBEWR.O.ATTR.SKEW.32 [R4], R9 ;

/*0100*/                   ISBEWR.O [0x4], RZ ;

/*0110*/                   EXIT ;
```

- New shiny instruction: ISBEWR ✨
- Primitive count stored at offset 0x0
- Primitive indices stored starting at offset 0x4
- Attributes in a different ISBE space

In the attribute space madness...

- Entirely dynamic
- Group of 32 values per attribute
- Defined by the shader header I / O definition
- Layout repeat if you have more than 32 vertices

In the attribute space madness...

- Quite complicated...
- Per primitive after and defined in GS header
- GS configured in “fast mode”

ISBE Attribute Layout	
0x000	MeshVertices[0].PointSize
0x004	MeshVertices[1].PointSize
...	
0x07C	MeshVertices[31].PointSize
...	
0x080	MeshVertices[0].Position.x
0x084	MeshVertices[1].Position.x
...	
0x27C	MeshVertices[31].Position.w
0x280	MeshVertices[32].PointSize
...	

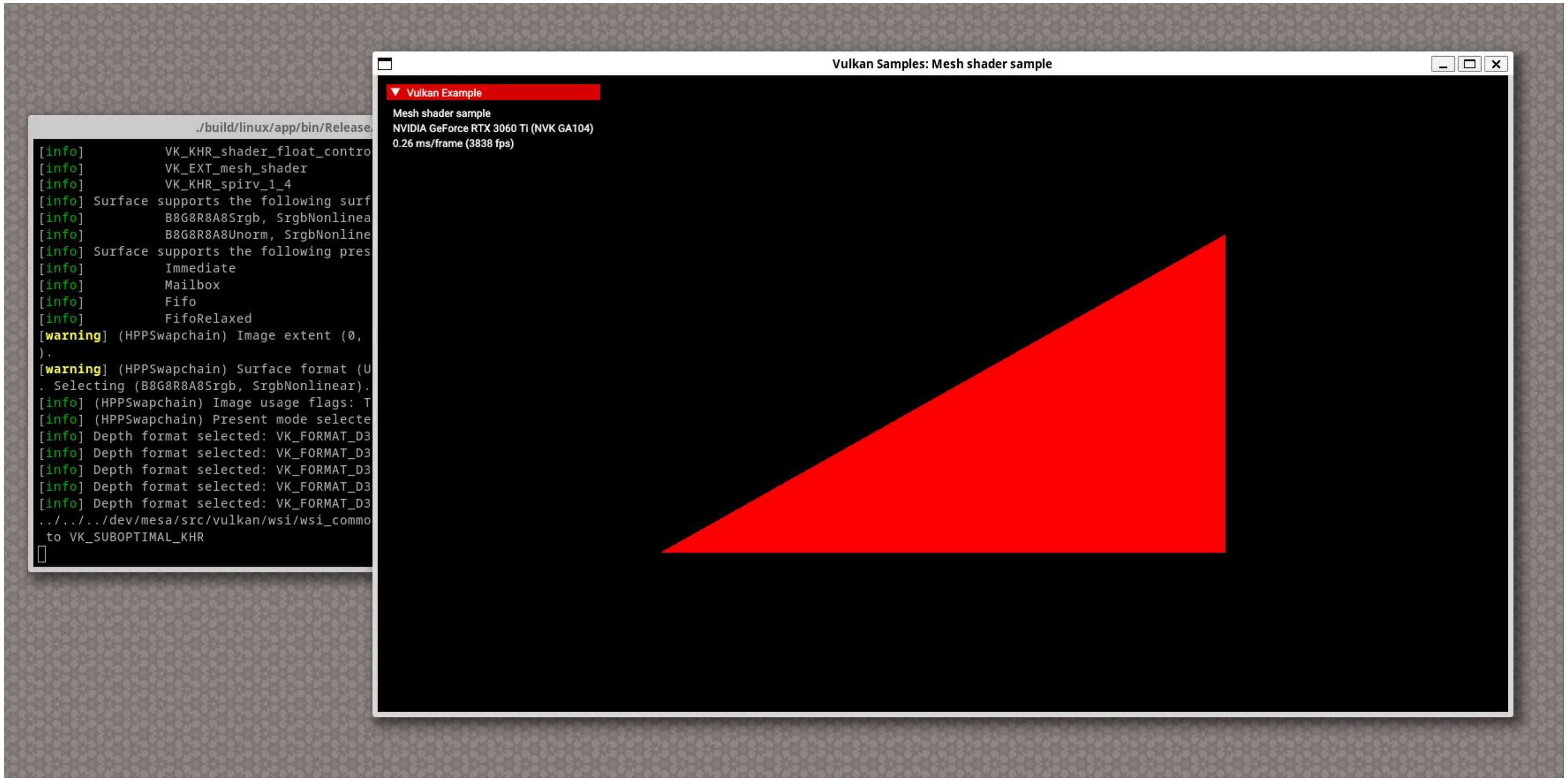
COLLABORA

Open First

COLLABORA

What is the current state?

What is the current state?



```
./build/linux/app/bin/Release
[info]          VK_KHR_shader_float_contro
[info]          VK_EXT_mesh_shader
[info]          VK_KHR_spirv_1_4
[info] Surface supports the following surf
[info]          B8G8R8A8Srgb, SrgbNonlinea
[info]          B8G8R8A8Unorm, SrgbNonline
[info] Surface supports the following pres
[info]          Immediate
[info]          Mailbox
[info]          Fifo
[info]          FifoRelaxed
[warning] (HPPSwapchain) Image extent (0,
).
[warning] (HPPSwapchain) Surface format (U
. Selecting (B8G8R8A8Srgb, SrgbNonlinear).
[info] (HPPSwapchain) Image usage flags: T
[info] (HPPSwapchain) Present mode selecte
[info] Depth format selected: VK_FORMAT_D3
[info] Depth format selected: VK_FORMAT_D3
[info] Depth format selected: VK_FORMAT_D3
[info] Depth format selected: VK_FORMAT_D3
[info] Depth format selected: VK_FORMAT_D3
../../../dev/mesa/src/vulkan/wsi/wsi_commo
 to VK_SUBOPTIMAL_KHR
```

Vulkan Samples: Mesh shader sample

Mesh shader sample
NVIDIA GeForce RTX 3060 Ti (NVK GA104)
0.26 ms/frame (3838 fps)

What is the current state?

Pass: 10792, Fail: 346, Crash: 4, Skip: 28346

Close yet still have a way to go...

Task 💔 Mesh

Pass: 4153, Fail: 3259, Crash: 8, Skip: 17068, Missing: 15000

- Global/local invocation indices seems different with task
- meshShaderQueries missing task/mesh invocations count
- Probably more…

Shared memory

- No true shared memory
- Use ISBE attribute space for shared memory
- Atomics emulation needed
  - nir_opt_uniform_atomics to the rescue!
  - Emulation still needed for xchg/cmpxchg

Invocations

- The hardware only support up to 32 local invocations
- **We need at least 128 by specification**
- We need emulation
  - One hardware invocation = up to 4 invocations
  - Non trivial to materialize with NIR

Thank you!

We are hiring

col.la/careers