olor\_Stage0 = {0, 0, 0, 0};

static float2 \_vTransformedCoords\_0\_Stage0 = {0, 0};

static float4 gl\_Color[1] =

{

float4(0, 0, 0, 0)

};

cbuffer DriverConstants : register(b1)

{

struct SamplerMetadata

{

int baseLevel;

int internalFormatBits;

int wrapModes;

int padding;

};

SamplerMetadata samplerMetadata[1] : packoffset(c4);

};

#define GL\_USES\_FRAG\_COLOR

float4 gl\_texture2D(uint samplerIndex, float2 t, float bias)

{

return textures2D[samplerIndex].SampleBias(samplers2D[samplerIndex], float2(t.x, t.y), bias);

}

void gl\_main()

{

float4 \_outputColor\_Stage0 = {0, 0, 0, 0};

{

(\_outputColor\_Stage0 = \_vcolor\_Stage0);

}

float4 \_output\_Stage1 = {0, 0, 0, 0};

{

float4 \_child = {0, 0, 0, 0};

{

(\_child = gl\_texture2D(\_uTextureSampler\_0\_Stage1, \_vTransformedCoords\_0\_Stage0, -0.5).xyzw);

}

(\_output\_Stage1 = (\_child \* \_outputColor\_Stage0.w));

}

{

(gl\_Color[0] = \_output\_Stage1);

}

}

struct PS\_INPUT

{

float4 dx\_Position : SV\_Position;

float4 gl\_Position : TEXCOORD2;

float4 v0 : TEXCOORD0;

float2 v1 : TEXCOORD1;

};

@@ PIXEL OUTPUT @@

PS\_OUTPUT main(PS\_INPUT input)

{

\_vcolor\_Stage0 = input.v0;

\_vTransformedCoords\_0\_Stage0 = input.v1.xy;

gl\_main();

return generateOutput();

}