Graphic debug notes

Kernel program

```
const char *KernelSource = "\n" \
" __kernel void normal_fragment_shader(__global float3* payload,
"                                      __global float3* return_color,
"                                      const unsigned int n)
" {
"     int tid = get_global_id(0);
"     if (tid < n) {
"         float3 color = (float3)(payload[tid].x + 1.0f, payload[tid].y + 1.0f, payload[tid].z +
"         return_color[tid] = (float3)(color.x * 255, color.y * 255, color.z * 255);
"     }
" }
```

```
        .text
        .section .mdebug.abi32
        .previous
        .file   "_cl_03_shader"
        .section        .rodata.cst4,"aM",@progbits,4
        .p2align        2               # -- Begin function normal_fragment_shader
CPI0_0:
        .word   1065353216              # float 1
CPI0_1:
        .word   1056964608              # float 0.5
CPI0_2:
        .word   1132396544              # float 255
        .text
        .globl  normal_fragment_shader
        .type   normal_fragment_shader,@function
        .ent    normal_fragment_shader  # @normal_fragment_shader
normal_fragment_shader:
        .frame  $sp,56,$ra
        .mask   0x00008000,-4
        .fmask  0x00000000,0
# %bb.0:                                # %entry
        sub     $zero, $zero, $zero
        add.si  $sp, $sp, -56
        st.w    $ra, 52($sp)
        st.w    $6, 32($sp)             # (n: 10)
        st.w    $5, 36($sp)             # return_color 0xc0000100
        st.w    $4, 40($sp)             # payload 0xc0000000
        add.si  $4, $zero, 0            # $4 = 0
        jplnk   _Z13get_global_idj      # $2 = id
        ld.w    $3, 32($sp)             # $3 = 10
        setlt.u $3, $2, $3              # $3 = 1   rd ← (rs1 < rs2)
        beq     $3, $zero, .LBB0_2          # if $3 == 0 then .LBB0_2
        jp      .LBB0_1
.LBB0_1:                                # %if.then
        sfll.i  $2, $2, 4               # $2 = 0
        ld.w    $3, 40($sp)             # $3 = 0xc0000000
        add     $3, $3, $2                  # $3 = (address)payload[0]
        ld.w    $4, 36($sp)             # $4 = 0xc0000100
        add     $2, $4, $2                  # $2 = return_color[0]
        mvup.i  $4, (CPI0_0)            # $4 = 0x00 (address)(float)1
        or.i    $4, $4, (CPI0_0)        # $4 = 0x00 (float)1
        ld.w    $4, 0($4)               # $4 = (float)1 *(0x00)
        ld.w    $5, 8($3)               # $5 = payload[0].z
        fadd    $5, $5, $4              # $5 = payload[0].z + 1
        mvup.i  $6, (CPI0_1)            # $6 = 0x08 (address)(float)0.5
        or.i    $6, $6, (CPI0_1)        # $6 = 0x08
        ld.w    $6, 0($6)               # $6 = (float)0.5 *(0x08)
        fmul    $5, $5, $6              # $5 = (payload[0].z + 1) * (float)0.5
        mvup.i  $7, (CPI0_2)            # $7 = 0x10 (float)255
        or.i    $7, $7, (CPI0_2)        # $7 = 0x10 (address)(float)255
```

```
        ld.w    $7, 0($7)            # $7 = (float)255 *(0x10)
        fmul    $5, $5, $7           # $5 = ((payload[0].z + 1) * (float)0.5) * 255
        ld.w    $8, 4($3)            # $8 = payload[0].y
        st.w    $5, 8($2)            # return_color[0].z = $5
        fadd    $5, $8, $4           # $5 = payload[0].y + (float)1
        fmul    $5, $5, $6           # $5 = (payload[0].y + (float)1) * 0.5
        fmul    $5, $5, $7           # $5 = ((payload[0].y + (float)1) * 0.5) * 255
        st.w    $5, 4($2)            # return_color[0].y = $5
        ld.w    $3, 0($3)            # $3 = payload[0].x
        fadd    $3, $3, $4           # $3 = payload[0].x + 1
        fmul    $3, $3, $6           # $3 = (payload[0].x + 1) * 0.5
        fmul    $3, $3, $7           # $3 = ((payload[0].x + 1) * 0.5) * 255
        st.w    $3, 0($2)            # return_color[0].x = $3
.LBB0_2:                            # %if.end
        ld.w    $ra, 52($sp)
        add.si  $sp, $sp, 56
        jr      $ra
        .set    macro
        .set    reorder
        .end    normal_fragment_shader
$func_end0:
        .size   normal_fragment_shader, ($func_end0)-normal_fragment_shader
                                        # -- End function

        .ident  "clang version 7.0.0 (tags/RELEASE_700/final) (ssh://lidajun@10.0.10.208:29418/llvm-7.0.0 cec070c
        .section        ".note.GNU-stack","",@progbits
```

Data Comparison

reference data:

```
[input]
x=349 y=275 (-0.474291, -0.819276, -0.322235)
x=350 y=274 (-0.439626, -0.850920, -0.287515)
x=351 y=273 (-0.403335, -0.879764, -0.251667)
x=351 y=274 (-0.437188, -0.849520, -0.295266)
x=352 y=273 (-0.401025, -0.878535, -0.259530)
x=352 y=274 (-0.434736, -0.848052, -0.303007)
x=353 y=273 (-0.398655, -0.877272, -0.267336)
x=353 y=274 (-0.432255, -0.846526, -0.310723)
x=354 y=274 (-0.429732, -0.844947, -0.318425)
x=355 y=274 (-0.427196, -0.843309, -0.326088)
[output]
x=349 y=275 (67.027870, 23.042351, 86.415070)
x=350 y=274 (71.447662, 19.007750, 90.841866)
x=351 y=273 (76.074806, 15.330095, 95.412407)
x=351 y=274 (71.758545, 19.186167, 89.853584)
x=352 y=273 (76.369339, 15.486829, 94.409889)
x=352 y=274 (72.071205, 19.373405, 88.866592)
x=353 y=273 (76.671539, 15.647819, 93.414688)
x=353 y=274 (72.387482, 19.567909, 87.882759)
x=354 y=274 (72.709206, 19.769199, 86.900787)
x=355 y=274 (73.032494, 19.978088, 85.923813)
```

shader data:

```
[input]
i=0 (-0.474291, -0.819276, -0.322235)
0xbef2d645
0xbf51bc12
0xbea4fbfc
i=1 (-0.439626, -0.850920, -0.287515)
i=2 (-0.403335, -0.879764, -0.251667)
i=3 (-0.437188, -0.849520, -0.295266)
i=4 (-0.401025, -0.878535, -0.259530)
i=5 (-0.434736, -0.848052, -0.303007)
i=6 (-0.398655, -0.877272, -0.267336)
i=7 (-0.432255, -0.846526, -0.310723)
i=8 (-0.429732, -0.844947, -0.318425)
i=9 (-0.427196, -0.843309, -0.326088)
[output]
```

Dead lock

GPGPU-Sim Cycle 9553: WARP_SCHEDULER - Core 0 - Scheduler 0 - scheduler_unit::cycle()
GPGPU-Sim Cycle 9553: WARP_SCHEDULER - Core 0 - Scheduler 0 - Testing (warp_id 12,

dynamic_warp_id 12)
GPGPU-Sim Cycle 9553: WARP_SCHEDULER - Core 0 - Scheduler 0 - Warp (warp_id 12, dynamic_warp_id 12) has valid instruction (<no instruction at address 0x3b0>)
GPGPU-Sim Cycle 9553: WARP_SCHEDULER - Core 0 - Scheduler 0 - Warp (warp_id 12, dynamic_warp_id 12) return from diverged warp flush
GPGPU-Sim Cycle 9553: WARP_SCHEDULER - Core 0 - Scheduler 0 - Testing (warp_id 0, dynamic_warp_id 0)
GPGPU-Sim Cycle 9553: WARP_SCHEDULER - Core 0 - Scheduler 0 - Warp (warp_id 0, dynamic_warp_id 0) fails as ibuffer_empty
GPGPU-Sim Cycle 9553: WARP_SCHEDULER - Core 0 - Scheduler 0 - Testing (warp_id 4, dynamic_warp_id 4)
GPGPU-Sim Cycle 9553: WARP_SCHEDULER - Core 0 - Scheduler 0 - Warp (warp_id 4, dynamic_warp_id 4) fails as ibuffer_empty
GPGPU-Sim Cycle 9553: WARP_SCHEDULER - Core 0 - Scheduler 0 - Testing (warp_id 8, dynamic_warp_id 8)
GPGPU-Sim Cycle 9553: WARP_SCHEDULER - Core 0 - Scheduler 0 - Warp (warp_id 8, dynamic_warp_id 8) fails as ibuffer_empty

> beq 指令跳转到一个非法地址，导致死锁
> beq 指令的实现有问题，未映射到正确的跳转地址

kernel debuger

给单个线程打断点

```
(kernel debugger) b ./tempfiles/_cl_Rasterizer.s:23 1
(kernel debugger) c
(kernel debugger) s
136 [thd=1][i=5] : ctaid=(0,0,0) tid=(0,0,0) icount=4 [pc=24] (./tempfiles/_cl_Rasterizer.s:23 -
Output Registers:
     $zero   .s32 0
Input Registers:
     $zero   .s32 0
Register File Contents:
        $2   .s32 0
       $sp   .s32 1024
        $6   .s32 10 (n: 10)
     $zero   .s32 0
        $5   .s32 -1073741568 (return_color: 0xc0000100)
        $4   .s32 -1073741824 (payload: 0xc0000000)
(kernel debugger) b ./tempfiles/_cl_Rasterizer.s:37 1
(kernel debugger) s
MVPGPU-Sim Kernel DBG: reached breakpoint 2 at ./tempfiles/_cl_Rasterizer.s:37 thread uid = 1 (s
MVPGPU-Sim Kernel DBG: reached by thread uid=1, sid=0, hwtid=0
MVPGPU-Sim Kernel DBG: PC=0x080  opcode: ld operand name: $3 operand value: 40 operand name: $sp

(kernel debugger)
```

1. mvp_beq_impl 指令
2. m_label 标签指令为0

Reference

- float2hex