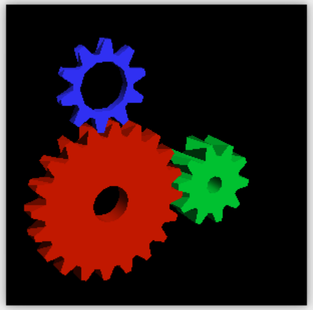
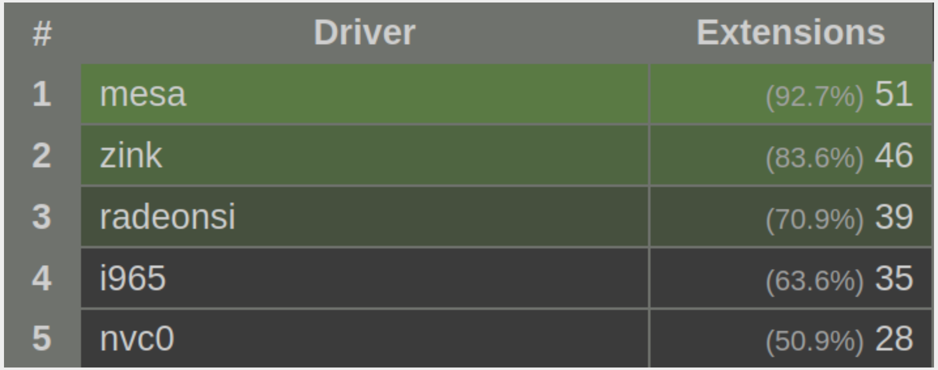
Zink: The Talk



// TODO: finish this

Zink: TL;DR

- Zink is a GL driver
- It does GLES too
- Also WGL
- It has feature support
- MesaMatrix is gray now?

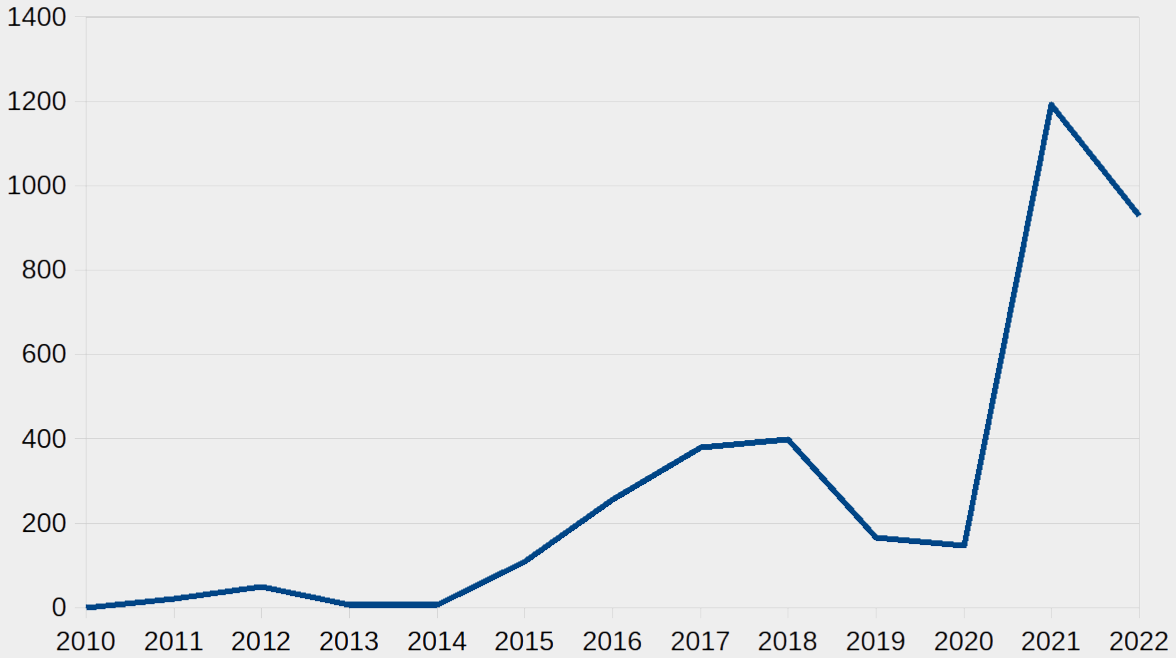


#	Driver	Extensions
1	mesa	(92.7%) 51
2	zink	(83.6%) 46
3	radeonsi	(70.9%) 39
4	i965	(63.6%) 35
5	nvc0	(50.9%) 28

Zink: The Early Years

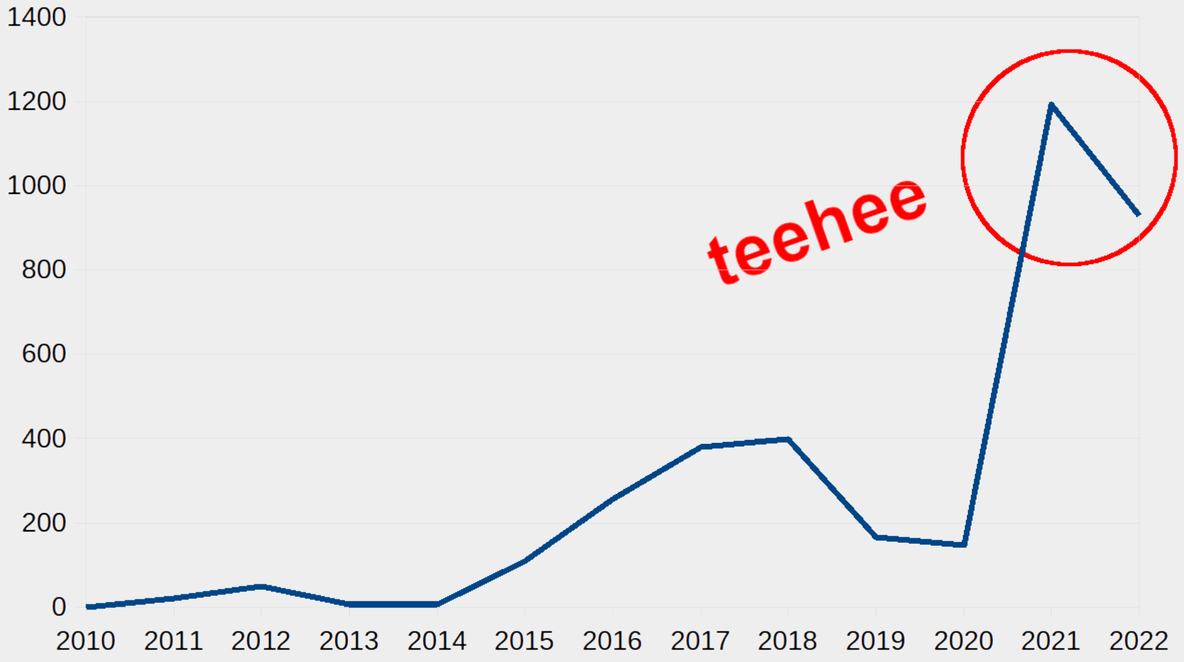
- Started in 2018 by Erik Faye-Lund
  - Merged 31 August 2018
- GL 4.6: 15 February 2021
- ES 3.2: 31 August 2021
  - Exactly 3 years for all versions

Zink: The Early Years Visualized



Dave Airlie: Mesa Reviews/Acks per year

Zink: The Early Years Visualized



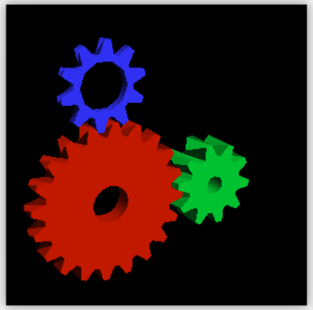
Dave Airlie: Mesa Reviews/Acks per year

Zink: What Took So Long?

- 3 whole years?!
- More like 4
  - My b

Zink: War Stories (Things I Hate)

- Provoking Vertex
  - Needed Vulkan extension
- gl_PointSize
  - No client API in Vulkan, no default



Zink: More Things I Hate

- Transform Feedback
  - Terrible
- Non-seamless Cubemaps
  - Perfect shader emulation is very hard

Zink: Somethin I Really Hate

- 64bit shader emulation
  - Complex
  - Tests take longer than full CI runs
  - Makes everything confusing and hard to understand

Zink: More Things I Hate

- Pixel Buffer Operations
  - PIPE_TEXTURE_TRANSFER_COMPUTE ?
  - pbobench ?
- Alpha/Luminance/Intensity format emulation
  - No Vulkan equivalents

Zink: Even More Things I Hate

- Gallium i/o lowering
  - Vulkan needs derefs, not dwords
- Internet Blog Posts About Vulkan Descriptors
  - Stop writing them

Zink: Still The Topic Of This Talk

- This was the introduction
- Prepare to get technical
  - Seriously
- You’re now breathing manually
  - So am I

Zink: Can I Run My Whole System On It Yet?

No.

Zink: WSI

- Kopper is great, but...
- It took 1.5 years to land
  - The Mesa DRI frontend really is that opaque
  - Also Vulkan WSI is still broken on X11
  - Also still have issues
    - Ancillary invalidation (Sorry, anholt!)
    - Auto-loading (Sorry, MrCooper!)
    - Random corner case explosions (Sorry, everyone else!)

Zink: WSI Solutions

- Collapse DRI frontend
  - Classic drivers are gone
  - This is not maintainable
- More Kopper testing
  - I can’t find all the bugs myself

Gallium: Tiler Optimizations

- Need more info when starting renderpass
  - Layout
  - loadOp
  - storeOp
  - Resolve attachments?

Gallium: Tilers Seeing The Future?

- Introduce threaded-context readahead?
    - Called on pipe_context::set_framebuffer_state
    - Uses driver-provided callbacks to parse command stream
        - pipe_context::bind_fs_state
        - pipe_context::bind_dsa_state
        - pipe_context::blit
        - ???

Gallium: Resolve Attachments?

- Should scanout resolve attachments be provided in framebuffer state?
  - https://gitlab.freedesktop.org/mesa/mesa/-/merge_requests/18695

Vulkan: Future Improvements For Tilers

- Working on something
- Not sure what it will end up being
- Hopefully solves these problems?
  - Pros:
    - Less CPU overhead from Gallium readahead
    - Simpler code in Zink
  - Cons:
    - Will probably be a long time before this materializes
    - More work for Ricardo

Gallium: Slow Vertex State Changes

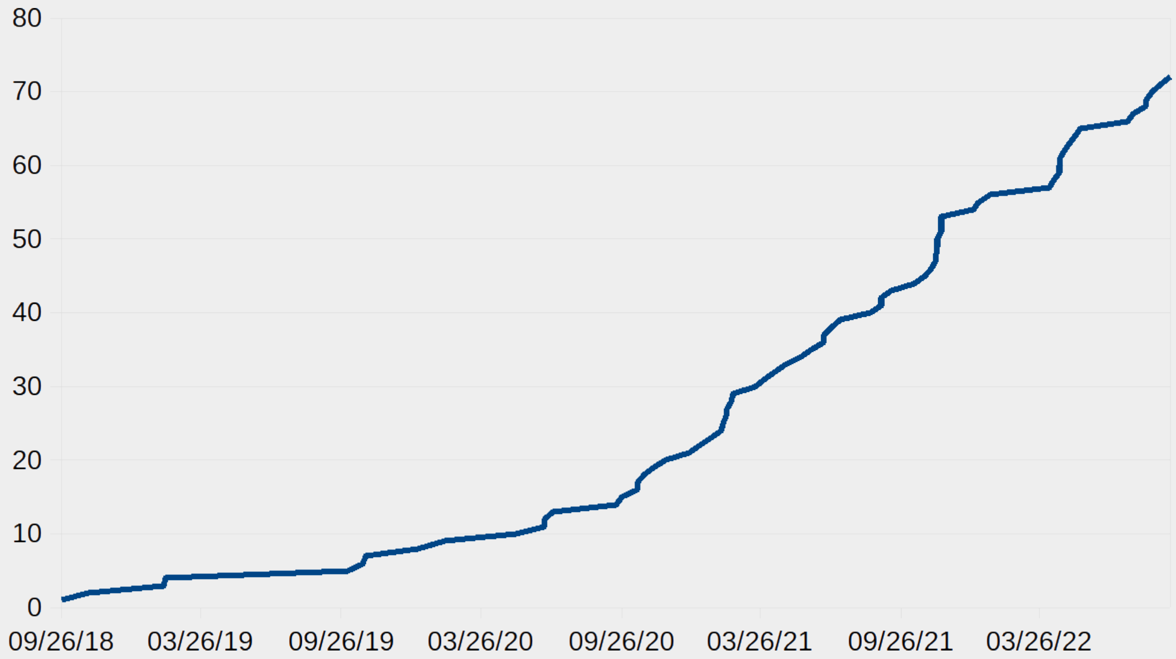
- pipe_context::set_vertex_buffers
  - Has stride
- pipe_context::bind_vertex_elements_state
  - Needs stride
- Overhead created in u_vbuf and Zink
- Hurt recent CPU benchmarks vs ANGLE
  - Zink too heavy on CPU :(

Gallium: Fast Vertex State Changes

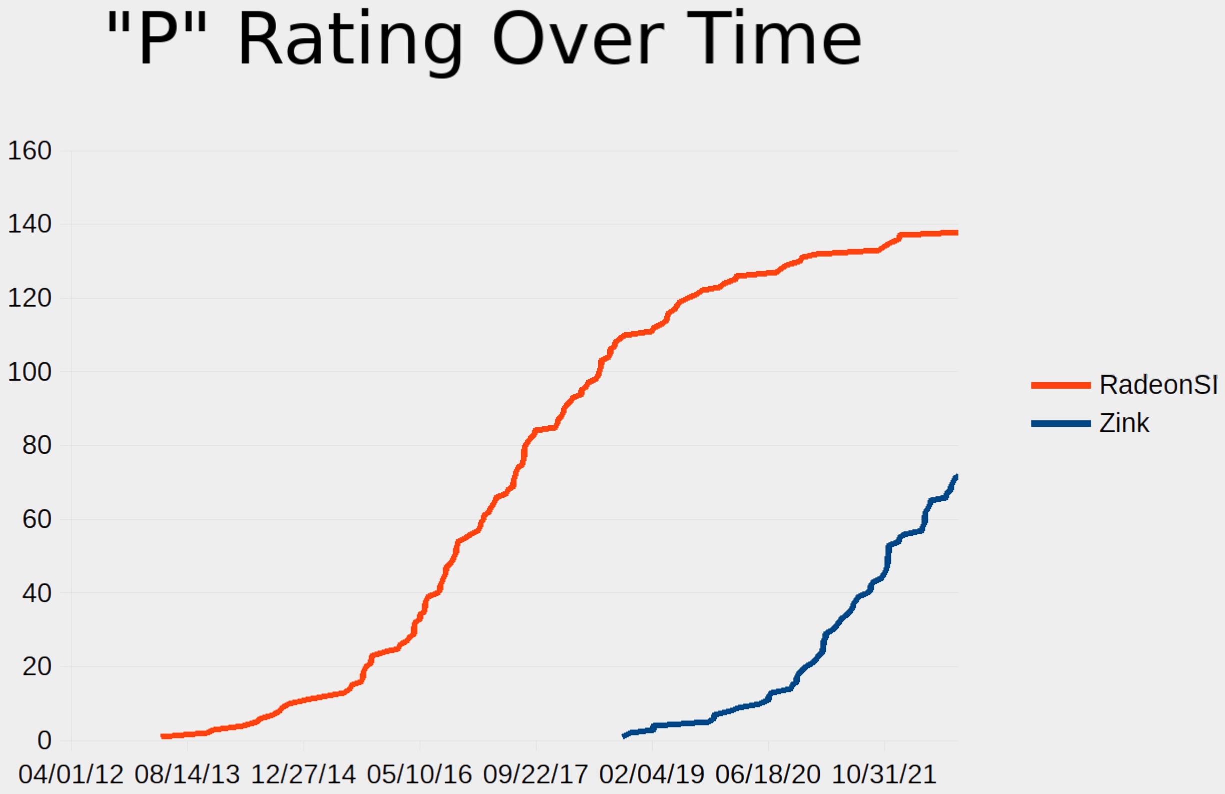
- ???
- pipe_context::set_vertex_buffers_no_stride ?

Zink: Benchmarking

"P" Rating Over Time



Zink: Benchmarking



Zink: What’s Left?

- #1 priority: eliminate shader compile stutters:
  - VK_KHR_dynamic_rendering
  - VK_EXT_graphics_pipeline_library
  - VK_EXT_non_seamless_cube_map
  - VK_EXT_all_the_dynamic_states

=

PRECOMPILATION

Zink: No More Stuttering ...Sometimes

- Only with drivers also supporting:
  - extendedDynamicState2PatchControlPoints
  - graphicsPipelineLibraryFastLinking
  - All the vertex attribute formats
- Also probably VK_EXT_vertex_input_dynamic_state
  - GPL is ~20x slower

- Vulkan is a great API to work with

  ...as long as the driver works as expected

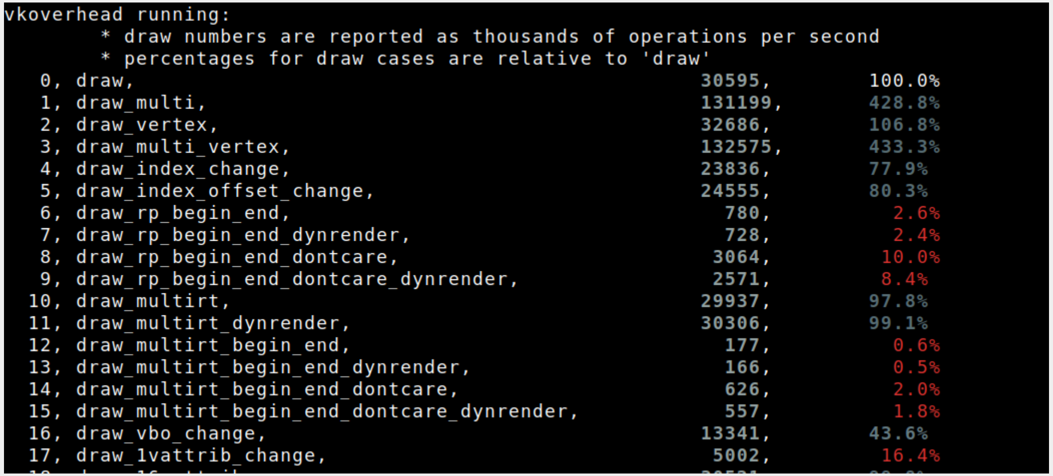
- VKCTS only goes so far
  - Leave Ricardo alone!
- GLCTS+piglit for extra coverage

Vulkan: What Happens To Driver Bugs?

- Report issue
- ???
- Some drivers more responsive than others
- How can this be improved?

Vulkan: What Happens To Slow Drivers?

- You thought I couldn’t plug it here
- https://github.com/zmike/vkoverhead/



```
vkoverhead running:
        * draw numbers are reported as thousands of operations per second
        * percentages for draw cases are relative to 'draw'
   0, draw,                                                   30595,         100.0%
   1, draw_multi,                                             131199,        428.8%
   2, draw_vertex,                                            32686,         106.8%
   3, draw_multi_vertex,                                      132575,        433.3%
   4, draw_index_change,                                      23836,         77.9%
   5, draw_index_offset_change,                               24555,         80.3%
   6, draw_rp_begin_end,                                        780,           2.6%
   7, draw_rp_begin_end_dynrender,                              728,           2.4%
   8, draw_rp_begin_end_dontcare,                              3064,          10.0%
   9, draw_rp_begin_end_dontcare_dynrender,                    2571,          8.4%
  10, draw_multirt,                                           29937,         97.8%
  11, draw_multirt_dynrender,                                 30306,         99.1%
  12, draw_multirt_begin_end,                                   177,           0.6%
  13, draw_multirt_begin_end_dynrender,                         166,           0.5%
  14, draw_multirt_begin_end_dontcare,                          626,           2.0%
  15, draw_multirt_begin_end_dontcare_dynrender,                557,           1.8%
  16, draw_vbo_change,                                        13341,         43.6%
  17, draw_1vattrib_change,                                    5002,          16.4%
```

Vkoverhead: Enlarge Your Perf!

- Found slow VRAM read in Turnip push descriptors
- Found 50x performance loss in RADV sampled image descriptors
- At least one major hardware manufacturer uses it internally
- Don’t wait!
- Try vkoverhead today!
- 300% perf gains or your money back!

Zink: The Future

- TOO MANY PIPE CAPS!
  - Seriously
  - Do you know how many there are?
  - Over 100
  - Over 200
  - Over 250
    - What do they even do?

Zink: Platform Testing

- Zink runs on lots of drivers
- How to effectively test on CI?
  - Is it feasible having jobs for every driver?
    - How about worthwhile?
  - CTS, piglit, traces, ???

Zink: Platform Distribution

- Zink runs on Windows
  - Apparently
  - It even performs better than native GL
    - Supposedly
    - Check reddit for details
- Mesa ships no “official” Windows release
  - Should this change?

Zink Needs Your Help

- I am only one person
  - There are 24 hours in a day
    - 6 hours sleep
    - 3 hours gym (primary workout; legs/chest/shoulders)
    - 8 hours work (email/cts results/bisecting/telecons)
    - 3 hours gym (secondary workout; cardio/arms/core)
    - 4 hours work (maybe actually write code/probably more cts runs)
    - 1 hour meaningful contributions to community discussions

Zink Needs Your Help

- Big ticket with starter tasks
  - https://gitlab.freedesktop.org/mesa/mesa/-/issues/5377
- Also plenty of other work to do on specific platforms
- Send memes too
  - As long as they aren’t better than mine

Zink: State Of Lavapipe

- All required features supported
- GL4.6 CTS passing with old version
  - Still subgroup issues
- ES3.2 mostly passing
  - Guardband clipping broken

Zink: State Of ANV

- Most features supported
  - Missing VK_EXT_vertex_input_dynamic_state
  - extendedDynamicState2PatchControlPoints
    - May add zink workaround for this?
  - No precompile yet (soon?)
  - Sparse binding support?
- GL4.6 CTS passing
- ES3.2 passing
- Variable perf

Zink: State Of RADV

- All features supported
  - Missing some sparse texture features
  - No precompile yet (soon?)
- GL4.6 CTS passing
  - Except one GTF test with uniform buffers
    - Which will surely be fixed
      - Right?
        - It’s not like I’ve been pinging about it for a while or anything
          - Or like there’s a ticket open
- ES3.2 has 2 remaining fails
  - MR open to fix them
    - ...since 2 months ago
- Perf is good.
  - I mean really good

Zink: State Of AMD(PRO)

- Missing (lots) of features
  - Some dynamic state2 (but enabling it somehow breaks hundreds of tests?)
  - VK_EXT_vertex_input_dynamic_state (No, GPL vertex input is NOT the same performance)
  - No dynamic state3 support?
  - GPL fast-linking could be faster
- GL4.6/ES3.2 CTS not passing
  - O O F
  - Tried shouting into the void
    - Did not hear back
  - Tried leaving a message
    - Oddly difficult
- Perf is fine
  - Better than native in some cases on Windows
    - Supposedly?
    - Check reddit for details

Zink: State Of Turnip

- All features supported
  - Precompile: SUPPORTED
  - Sparse binding support?
- GL4.6/ES3.2 CTS nearly passing
  - Failing fewer than 5 tests
  - Great work since a couple months ago when it was in the hundreds
- Perf is pretty good
  - Heroic work by anholt

Zink: State Of NVIDIA

- All features supported
  - Precompile: SUPPORTED
- GL4.6 CTS not passing
  - Still failing a number of tests
    - They have a spreadsheet of all the failures
      - Surely they will be fixed?
- ES3.2 has lots of failures
  - Haven’t reported yet
  - Some failures also cause hangs
- Perf is good.
  - Really good.
  - Unless you hit one of the weird NVIDIA WSI bugs and your app won’t start (looking at you glretrace)

Zink: State of NVK

- All features supported
- Passes GL4.6/ES3.2 CTS
  - First try
- 5000% faster than nouveau
- 200% faster than NVIDIA proprietary in Tomb Raider
- 71% faster in DOOM2016
- Written in ASM
- First driver to use NIR 2.0
- jekstrand blocked me on IRC when I asked if he’d gotten any hangs yet

Zink: State Of $DRIVER

- Your driver here in the next presentation!
  - This could be you!
  - Look at all this space!
  - And I’m talking!
    - But I could be talking about how great you are!
      - Unless you’re not great
        - In which case it’s great that you’re not here!
          - But you should work on that!
            - Find a hobby!
            - Read a book!
            - Hit the gym!
            - Challenge yourself!
          - So much presentation time and slide space to fill!

Zink: Testimonials + Contact

- <mareko> zink+anything is always an interesting combination
- <anholt> well, perhaps zink exploding this nice shader out to being 64k instructions is part of zink perf issues /o\
- <alyssa> zmike: sorry zink has to go
- <jekstrand> Zink is weird.
- <kusma> ??? :)
- <anholt> "Type mismatch for SPIR-V SSA value 369636 bytes into the SPIR-V binary" what if I don't want to look 370kb into a spirv binary, huh? have you considered that
- <dj-death> is is possible to run zink without all the winsys integration?

- <karolherbst> the more I think about using vulkan the more I am convinced in using zink tbh
- <airlied> okay zink and virgl no longer xplode
- <ManMower> like a zink developer vs the unending deluge of CI
- <rg3igalia> but it's true that zink is a very nice use case and I'm glad we have it to dig up bugs and flesh out some test concepts
- <jekstrand> Well, it's definitely a zink bug.
- <hakzsam> this is zink only, makes no sense
- <karolherbst> if a god exists, I am sure that one tries to convince me to just go straight with zink, and I don't listen
- <ajax> how am i still hitting that stupid wait_for_event deadlock

- #zink on OFTC network
- https://gitlab.freedesktop.org/mesa/mesa
- https://www.supergoodcode.com/