FreeSync™, Adaptive Sync & VRR

Harry Wentland
AGENDA

Static and dynamic refresh rates

DP Adaptive Sync, HDMI™ VRR, FreeSync™

VRR in DRM & Mesa

Next Steps

Conclusion & Questions
Static and dynamic refresh rates
Performance:
Stable
Frames Score:
6136
Test results (average):
<table>
<thead>
<tr>
<th>FPS</th>
<th>CPU(ms)</th>
<th>GPU(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>51</td>
<td>15</td>
<td>20</td>
</tr>
</tbody>
</table>

Test Session Time: 9/28/2019 2:27 PM
~ 68 Hz

~ 48 Hz
Dynamic refresh rates for gaming

- Render rate varies with content
- Latency between render and display matters
Dynamic refresh rates for gaming

- Mismatch between render rate & refresh rate leads to stutter & lag
Dynamic refresh rates for gaming

- Syncing render & refresh rates reduces lag and eliminates stutter

![Diagram showing the concept of dynamic refresh rates and its effects on lag and stutter](image)
Other dynamic refresh rate use cases

- Desktop Content (Nominal 60 Hz)
- Gaming Content (30 ~ 120Hz+)
- Static Desktop Content (Lowest Refresh Rate)
- Video Content (24/25/30/50/60 Hz)
Benefits of Adaptive Sync - Video

- Video frame rates rarely match display refresh rate
- Common video frame rates: 24, 25
- Common display refresh rates: 60, 120
- Would be nice to switch to 24 Hz without a mode change
- With dynamic refresh rates we can seemlessly adjusted to video’s frame rate
- We can save power running at lower refresh rate
Benefits of Adaptive Sync – Power Saving

- Desktop content is mostly static in many cases
- Using adaptive sync system can switch to a lower refresh rate for static content
- Power Savings
DP Adaptive Sync, HDMI™ VRR, FreeSync™
What is Adaptive Sync

- VESA spec introduced variable refresh rate framework called “ignore MSA” with initial eDP spec
- Rolled out to DP and branded as ‘Adaptive-Sync’ with DP 1.2a in 2014
- Protocol to seamlessly vary framerate by changing blank duration and keeping pixel rate the same
- VESA press release addressed three main use cases
  - Seamless variable frame change for smooth gaming use case
  - Seamless change of frame rate to match video rate for judder free video playback
  - Reduce frame rate for power saving in battery

How does adaptive sync work?

- Tx (Transmitter) reads range limits from EDID
- When enabling the display
  - Tx writes ignore_msa bit in DPCD
- When user indicates content is suitable for adaptive sync
  - Set up Tx with range limits
  - Tx will extend vertical blank
  - For low latency use case (i.e., gaming) Tx shall end frame immediately once new frame is presented
Adaptive Sync DP Symbols
What is HDMI™ VRR

- Part of HDMI™ 2.1 spec
- AMD currently doesn’t enable HDMI™ VRR pending HDMI™ VRR CTS
- AMD enables FreeSync™ on HDMI™ via AMD proprietary protocol (Windows only)
What is FreeSync™

- AMD implementation of adaptive sync and VRR
- DP supported via
  - Adaptive Sync spec
  - Proprietary protocol
- HDMI™ supported via
  - Proprietary protocol
- FreeSync™ certification and logo
- FreeSync™ 2
  - Adaptive sync
  - HDR
  - Stricter certification requirements
**Physical Limitations**

- **Static flicker**
  - At very low refresh rates some displays will exhibit flicker due to luminance drop in between frames

- **Dynamic flicker**
  - When switching between short and long frame durations average brightness changes due to larger luminance drop for longer frames
VRR in DRM & Mesa
DRM/KMS VRR interface

CRTC Property

`vrr_enabled`

Indicates if variable refresh rate should be enabled for the CRTC. Support for the requested vrr state will depend on driver and hardware capability - lacking support is not treated as failure.

Connector Property

`vrr_capable_property`

Optional property to help userspace query hardware support for variable refresh rate on a connector. Drivers can add the property to a connector by calling `drm_connector_attach_vrr_capable_property()`. This should be updated only by calling `drm_connector_set_vrr_capable_property()`.
VRR in Userland (X)

- VRR is supported by
  - radeonsi GL
  - orca GL (proprietary AMD GL driver)
  - radv Vulkan

- A free-running variable refresh rate is not suited for all rendered content, such as current implementations of web browsers, compositors, video players

- Mesa has a blacklist through DriConf for such applications: 00-mesa-defaults.conf

- VRR is enabled for GL/Vulkan rendered applications that use the Present extension and are not blacklisted

- Requires xf86-video-amdgpu

- X doesn’t support present flipping unless the application covers the entire X screen, which means VRR generally won’t enable on multi-monitor setups
Adaptive Sync Patches

- **Mesa**: https://patchwork.freedesktop.org/series/51388/
- **xf86-video-amdgpu**: https://gitlab.freedesktop.org/xorg/driver/xf86-video-amdgpu/merge_requests/5
- **Kernel**: https://patchwork.freedesktop.org/series/49487/
FreeSync™ in DC

```
mod_freesync_handle_v_update
  dm_crtc_high_irq
  crtc_state->vrr_params

mod_freesync_build_vrr_params
  crtc_state->vrr_params
  amdgpu_dm_commit_planes
  crtc_state->vrr_params

dc_stream_adjust_vmin_vmax
  v_min, v_max

optc1_set_drr
```
Next Steps
Enabling VRR beyond X, what is needed?

- Currently VRR is only enabled on X
- Good candidates to start adopting VRR would be
  - Wayland
    - Weston
    - Plasma
    - Gnome
  - ChromeOS
  - Etc.
- Looking for community engagement and input here
Enabling more use cases on Linux

- Current solution only covers gaming
- Smooth video playback requires refresh rate to match the content rate
  - With adaptive sync we can dynamically switch the refresh rate without requiring a mode set
- When the desktop is static there is no need to output at full refresh rate
  - Lowering the refresh rate can provide power savings
- Etc.... would love to see what other use cases the community comes up with
A frame duration time

- DRM/KMS exposes a frame duration time
- If userland provides it the kernel driver will program HW to refresh at rate calculated from frame_duration_time
A frame duration time

- Video players can
  - Target the presentation duration, e.g. 1000 / 24 ms
  - Fudge the presentation duration up or down if audio playback drifts
- Compositors can
  - Target a larger presentation time on static screen
A frame duration time – pros and cons

- **Pros**
  - Programming length of frame we're submitting → can use HW to adjust frame
  - No need to recalculate frame time every frame for fixed rates
  - No need to calculate frame duration in driver
A frame duration time – pros and cons

- Cons
  - Potential for dynamic flicker
  - Userspace has to be aware of frame presentation time (vsync)
  - If flip is programmed too late results won’t be as expected
An absolute presentation target

- DRM/KMS exposes an presentation target timestamp
- If userland provides it the kernel driver will program HW in such a way that the start of scanout is no sooner than the timestamp
An absolute presentation target

- Video players can
  - Target the presentation time to be current_time_in_ms + (1000/24)
  - Fudge the presentation time up or down if audio playback drifts
- Compositors can
  - Target a larger presentation time on static screen
An absolute presentation target – pros & cons

- Pros
  - Aligns with existing vdpau interface\(^1\) (earliest\_presentation\_time)
  - Aligns with existing vulkan extension\(^2\) VK\_GOOGLE\_display\_timing
  - Allows SW synchronization of all displays if they all support adaptive sync
  - Userspace doesn’t need to be aware of range limits (vmin/vmax) or vsync
  - Might be useful for VR cases

---

1) [https://http.download.nvidia.com/XFree86/vdpau/doxygen/html/group___vdp_presentation_queue.html#ga5bd61ca8ef5d1bc54ca6921aa57f835a](https://http.download.nvidia.com/XFree86/vdpau/doxygen/html/group___vdp_presentation_queue.html#ga5bd61ca8ef5d1bc54ca6921aa57f835a)

2) [https://github.com/KhronosGroup/Vulkan-Docs/blob/master/appendices/VK_GOOGLE_display_timing.txt](https://github.com/KhronosGroup/Vulkan-Docs/blob/master/appendices/VK_GOOGLE_display_timing.txt)
An absolute presentation target – pros & cons

- Cons
  - Potential for dynamic flicker
  - Userspace has to calculate new target presentation time with each flip
  - Display can’t use HW to target presentation – has to schedule this in SW
  - Need to limit how far in the future presentation target can be
Conclusions
Conclusions

- Dynamic refresh rates greatly improve the gaming experience by reducing
  - Lag
  - Stutter
- There are many displays on the market that support dynamic refresh
- Dynamic/variable refresh rate support is available on X
- Wayland compositors still lack support
- A more explicit interface might be useful to enable other use cases
Questions
DISCLAIMER AND ATTRIBUTIONS

DISCLAIMER

The information contained herein is for informational purposes only, and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD’s products are as set forth in a signed agreement between the parties or in AMD’s Standard Terms and Conditions of Sale. GD-18

AMD, the AMD Arrow logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

©2019 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.