

Embedded Recipes 2023

# Display support on Embedded Systems:

a tour of Linux implementation &  
limitations

Neil Armstrong - Linaro Developers Service  
Qualcomm Landing Team

29 September 2023, Paris



# Introduction

- Qualcomm Landing Team @ Linaro
  - Qualcomm upstream maintenance
  - Bringup/Addition of new platforms
- I also maintain other upstream pieces
  - Amlogic SoCs
    - Linux & U-Boot architecture
    - Clocks
    - Pinctrl, Serial, CEC ...
  - DRM
    - Bridge drivers
    - Panel drivers
    - Amlogic Display driver
  - Defunct OXNAS platform
- 265 changes, 16369 additions(+), 9544 deletions(-) in the last year

# Introducing Linaro

## Linaro collaborates with businesses and open source communities to:

- Consolidate the Arm code base & develop common, low-level functionality
- Create open source reference implementations & standards
- Upstream products and platforms on Arm

## Why do we do this?

- To make it easier for businesses to build and deploy high quality and secure Arm-based products
- To make it easier for engineers to develop on Arm

## Two ways to collaborate with Linaro:

- 1 Join as a member and work with Linaro and collaborate with other industry leaders
- 2 Work with Linaro Developer Services on a one-to-one basis on a project

For more information go to: [www.linaro.org](http://www.linaro.org)

# Linaro Developer Services

Linaro Developer Services helps companies build, deploy and maintain products on Arm

Arm Software expertise	Specialists in TEE on Arm	Continuous Integration through LAVA	Build, Test and deploy faster
<p>As part of Linaro, Developer Services has some of the world's <b>leading Arm Software experts</b>.</p>	<p>We specialize in security and Trusted Execution Environment (TEE) on Arm.</p>	<p>We offer continuous integration (CI) and automated validation through LAVA (Linaro's Automation &amp; Validation Architecture)</p>	<p>We support every aspect of product delivery, from building secure board support packages (BSPs), product validation and long-term maintenance.</p>

For more information go to: <https://www.linaro.org/services/>

# Linaro membership collaboration



# Agenda

1. A pixel's journey
2. Glossary
3. Display engines architectures
4. Display protocols
5. Support limitations
6. Standards paywall
7. Conclusion
8. Q&A

# Not (entirely) a DRM talk!

If you want some more detailed info about the Linux Direct Rendering Manager:

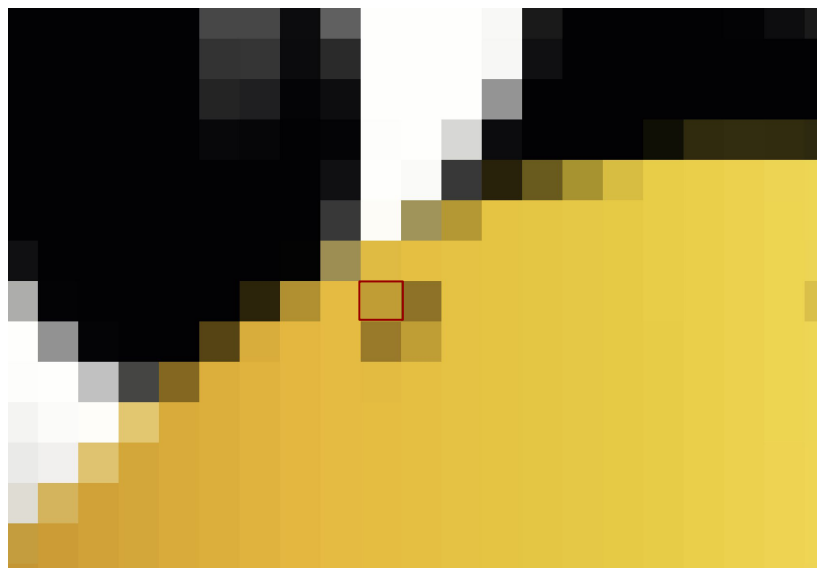
- [Talk "Trading Fbdev for DRM, no returns accepted" by Geert Uytterhoeven](#)
- ["The DRM/KMS subsystem from a newbie's point of view" Boris Brezillon](#)
- ["DRM Driver Development For Embedded Systems" Inki Dae](#)
- ["An introduction to the Linux DRM subsystem" Maxime Ripard](#)

And plenty of others.

And I won't talk about GPUs either, on embedded systems they are separate HW blocks that write pixels on memory.

# A Pixel's Journey

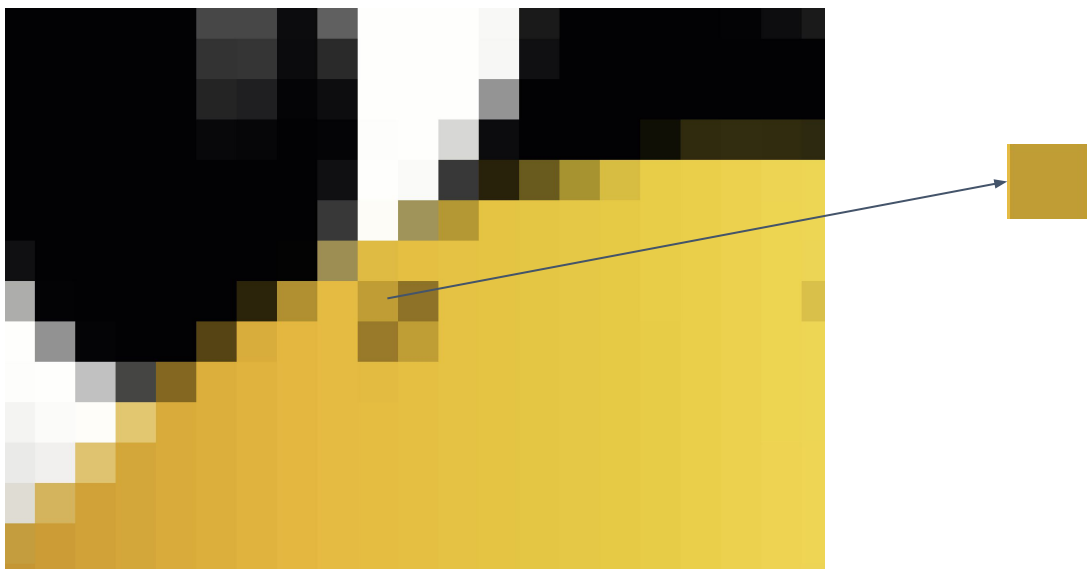
Our little Pixel's journey starts from the memory where  
It's sleeping with his friends:





# A Pixel's Journey

Our little Pixel gets read by the Display Engine



# A Pixel's Journey

But not alone, along it's friends on the same row:

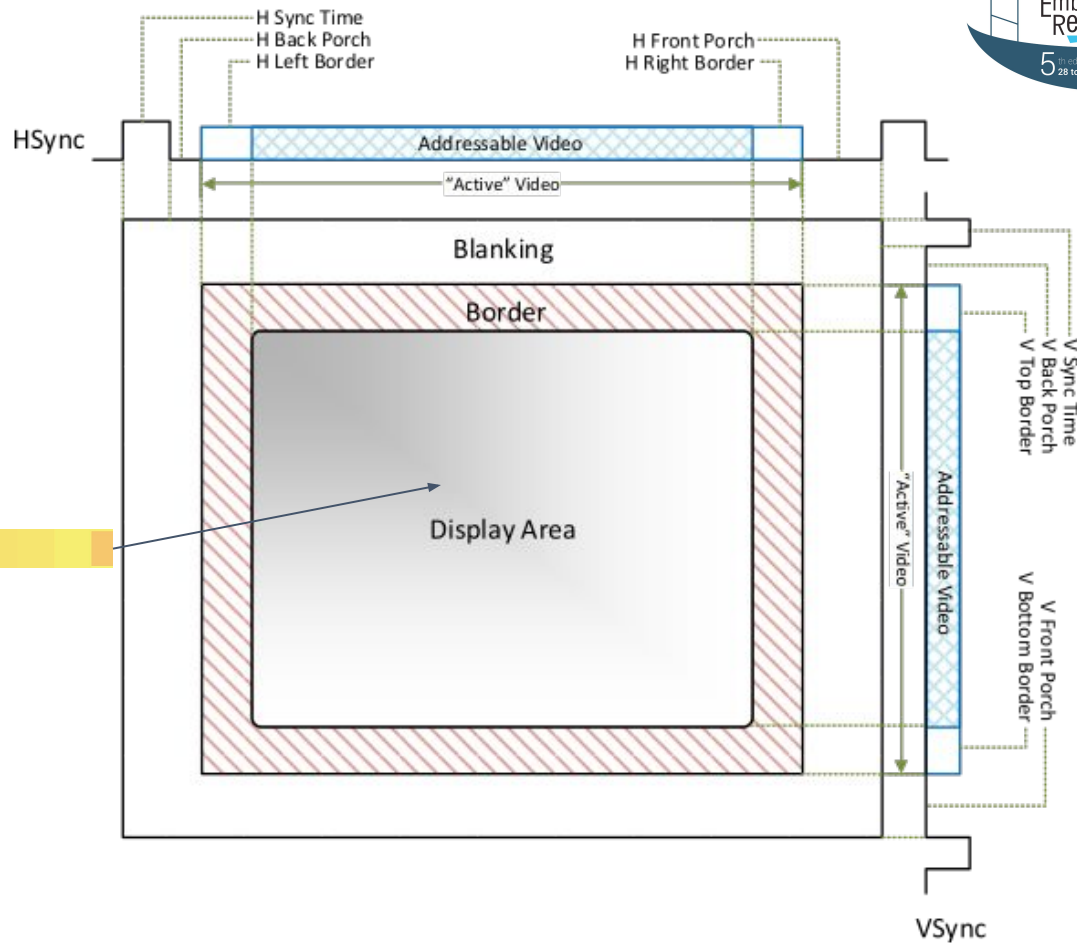


Then it gets tweaked and merged with other pixels:



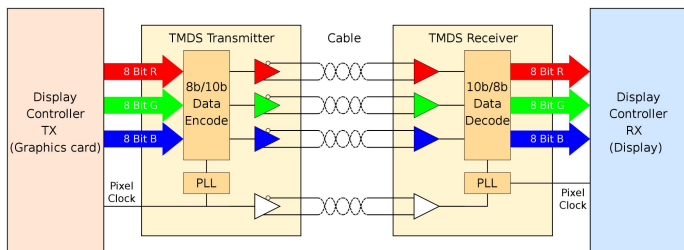
# A Pixel's Journey

Then it gets aligned with display timings:

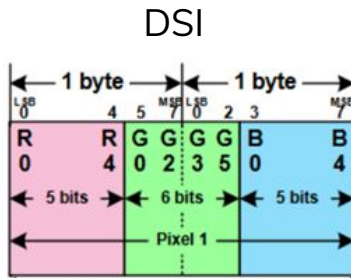


# A Pixel's Journey

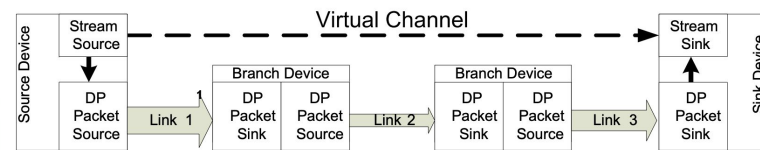
Into TMDS/DSI/DisplayPort packets:



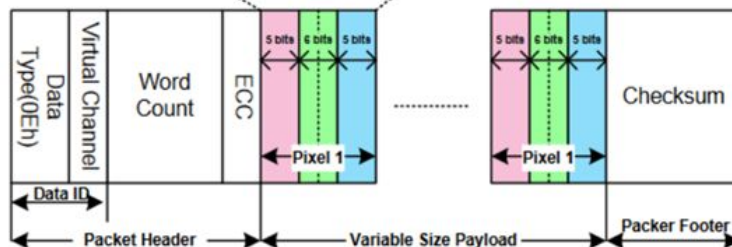
DVI/HDMI



DSI

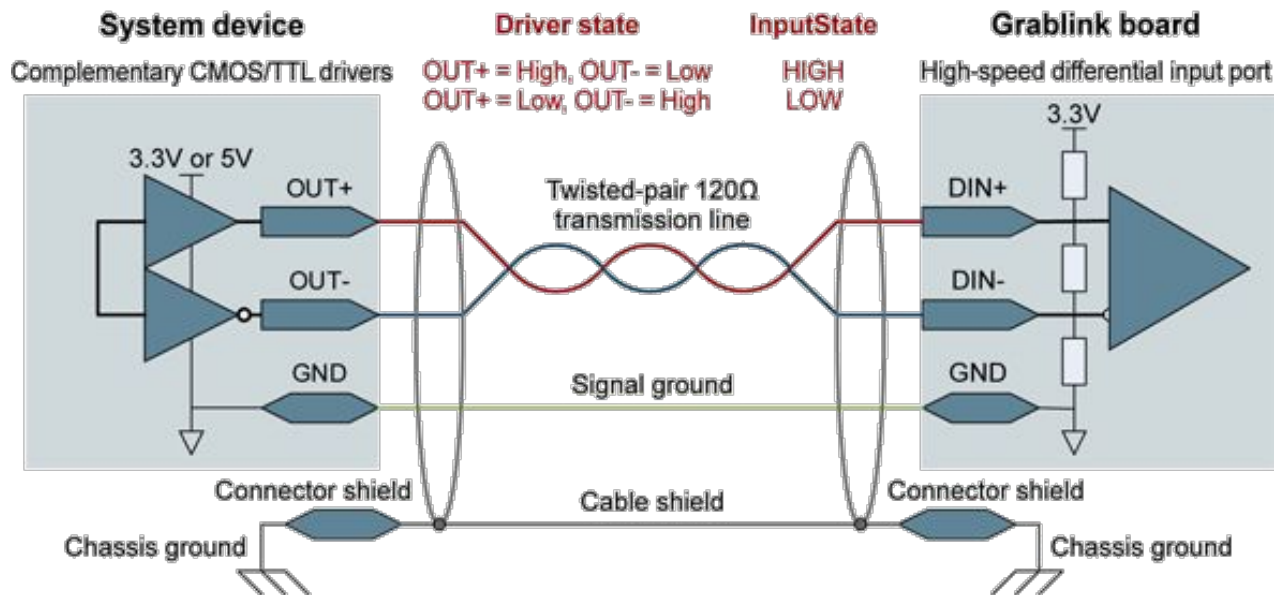


DisplayPort



# A Pixel's Journey

Over a cable wires:



# A Pixel's Journey

Onto a display:



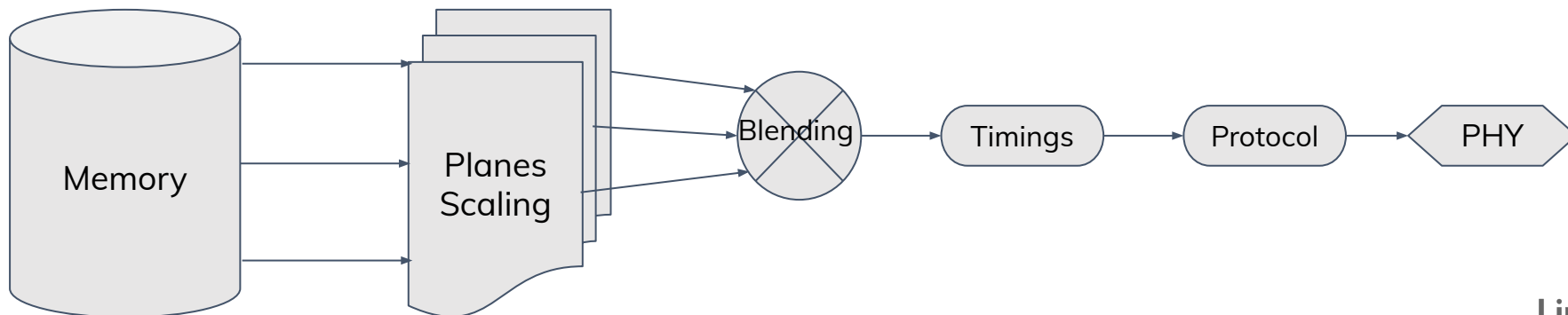
# Glossary

- DRM
  - Direct Rendering Manager
- Plane
  - Layer to be color corrected, scaled & blended with background and other layers
- Source
  - Emitter of the display signal
- Sink
  - Receiver of the display signal
- Timings
  - Out-of-display transmission times used to synchronize the source and the sink
- Packet
  - Minimal group of bits transmitting a payload, usually a group of pixels
- Lanes
  - Number of physical wires (or analog group of wires) used to transmit packets
  - Packets are usually spread over the available lanes to transmit more information

# Display Engines Architectures

Usually, display engine consists of:

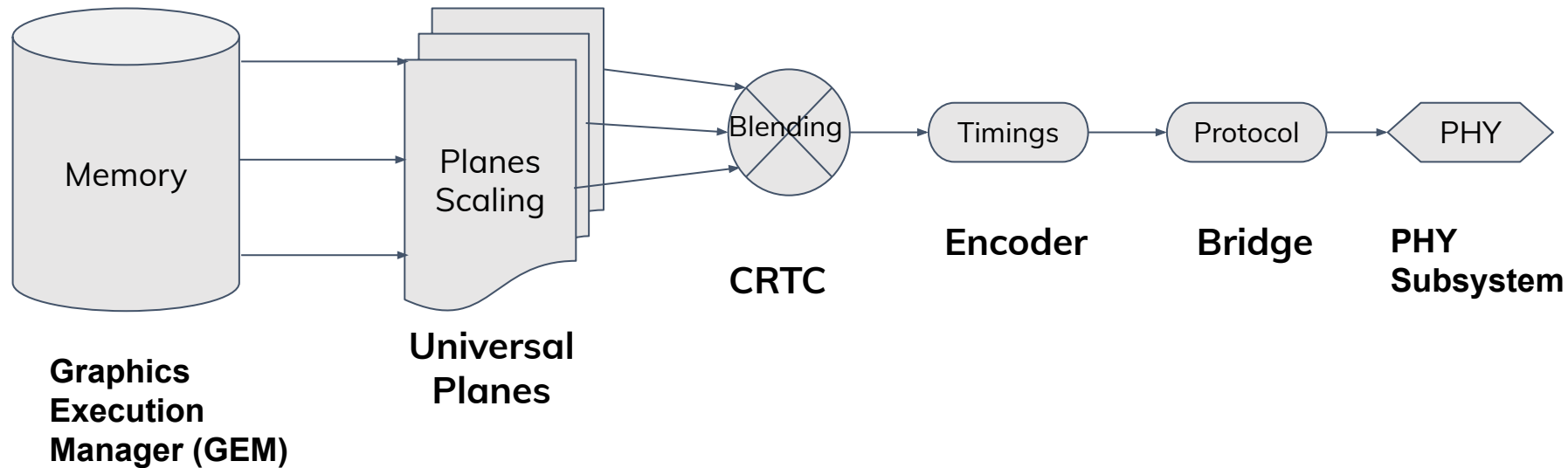
- **Memory reader:** gets pixel from memory
- **Planes blender:** color correct, scaler and merge multiple layers
- **Pixel timings generator:** takes the viewable image and adds necessary synchronization data before and after the display area
- **Protocol Transceiver:** takes the pixel data, cuts it in packets & prepares then to be transmitted over the physical link
- **PHY:** Physical layer converting the digital data into analog signals



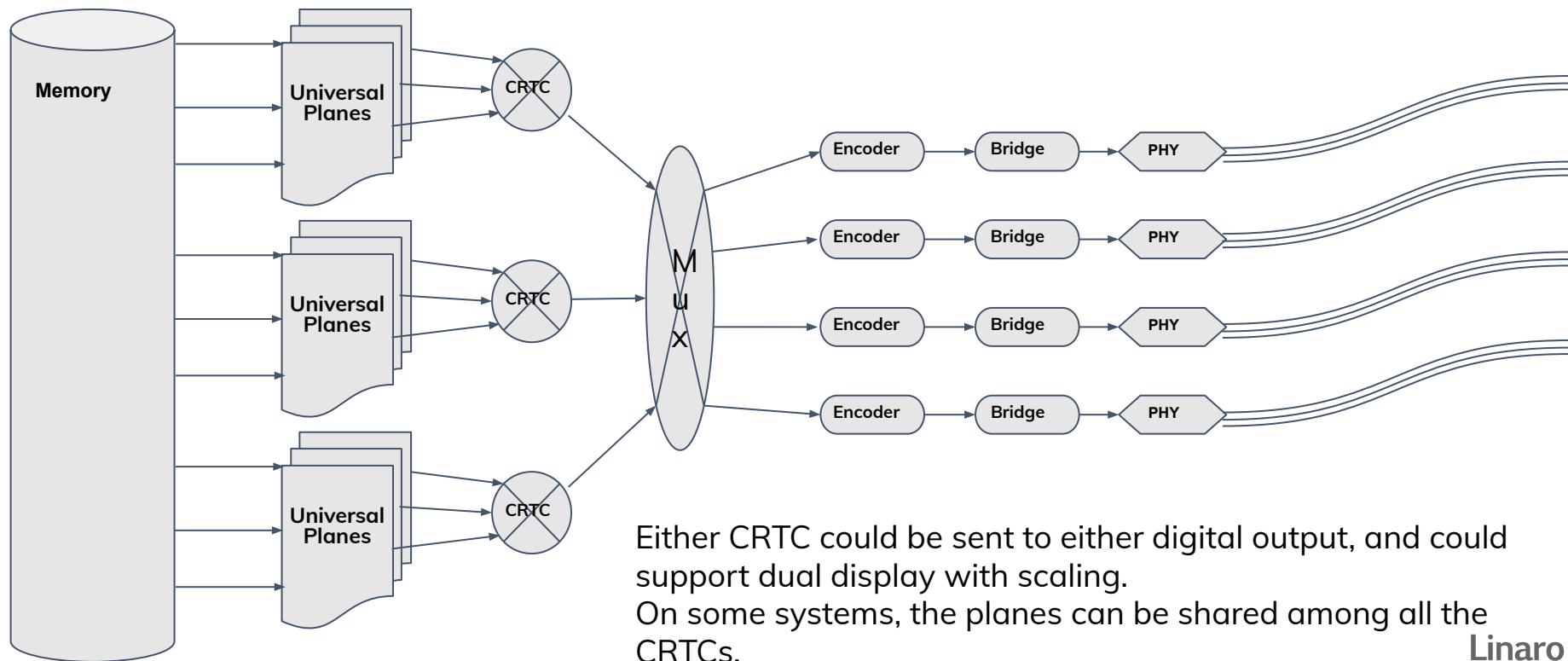


# Display Engines Architectures

Equivalent DRM or Linux subsystems:



# Display Engines architecture



Either CRTC could be sent to either digital output, and could support dual display with scaling.  
On some systems, the planes can be shared among all the CRTCs.

# Display Engines architecture

Qualcomm Snapdragon™ 410E  
 Simplified Display Engine

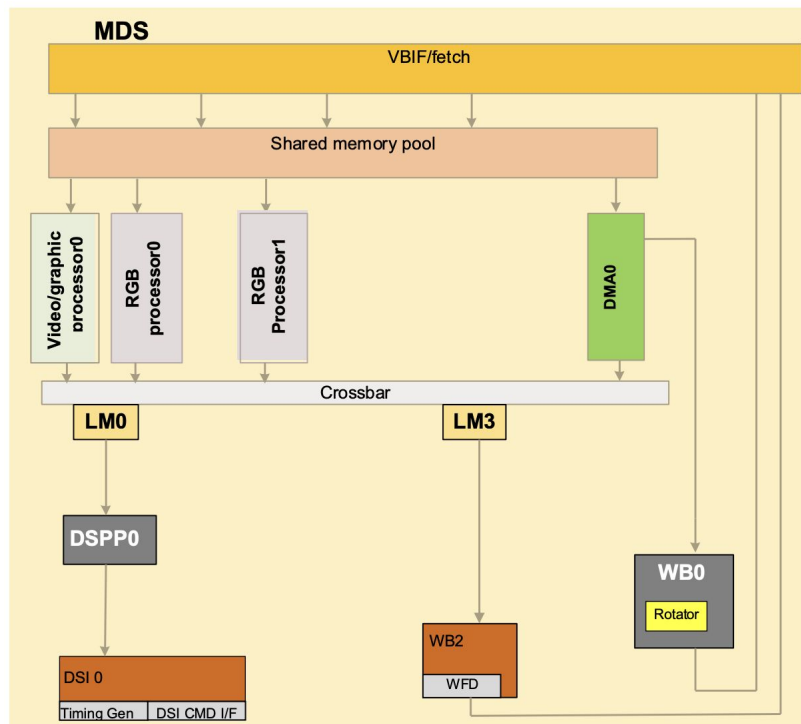
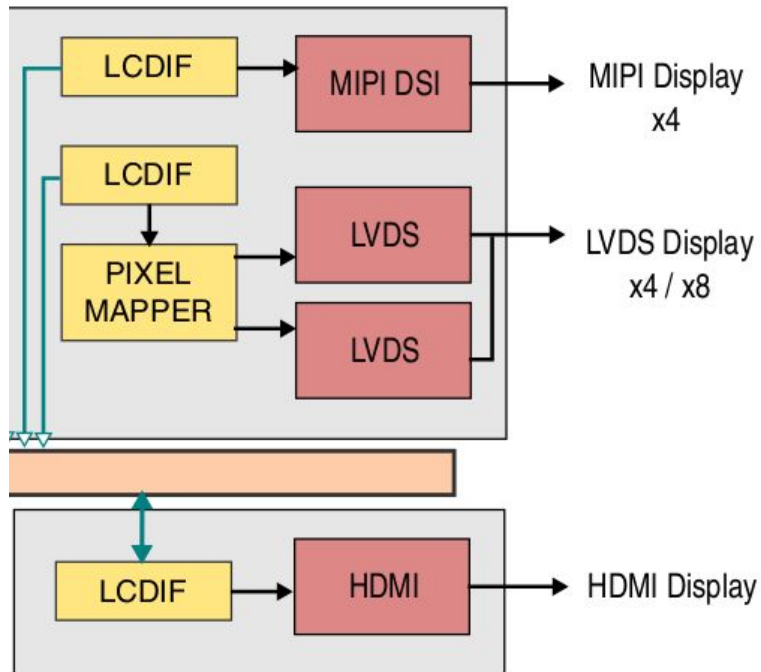


Figure 2-1 Display peripheral

Source: <https://developer.qualcomm.com/download/db410c/android-display-overview.pdf>

# Display Engines architecture

iMX8M+



Source: <https://www.nxp.com/webapp/Download?colCode=IMX8MPRM>

# Display Engines architecture

Allwinner A64

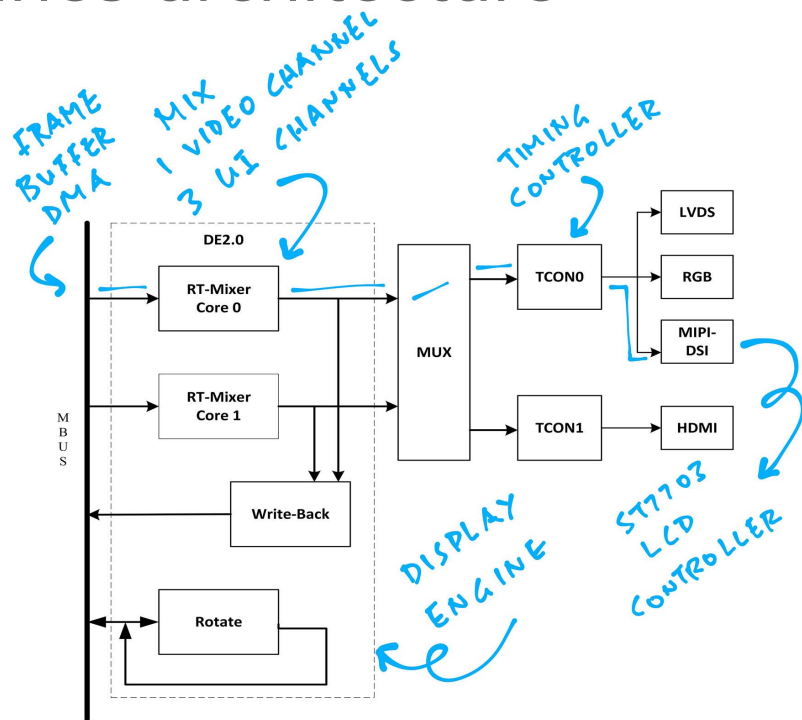
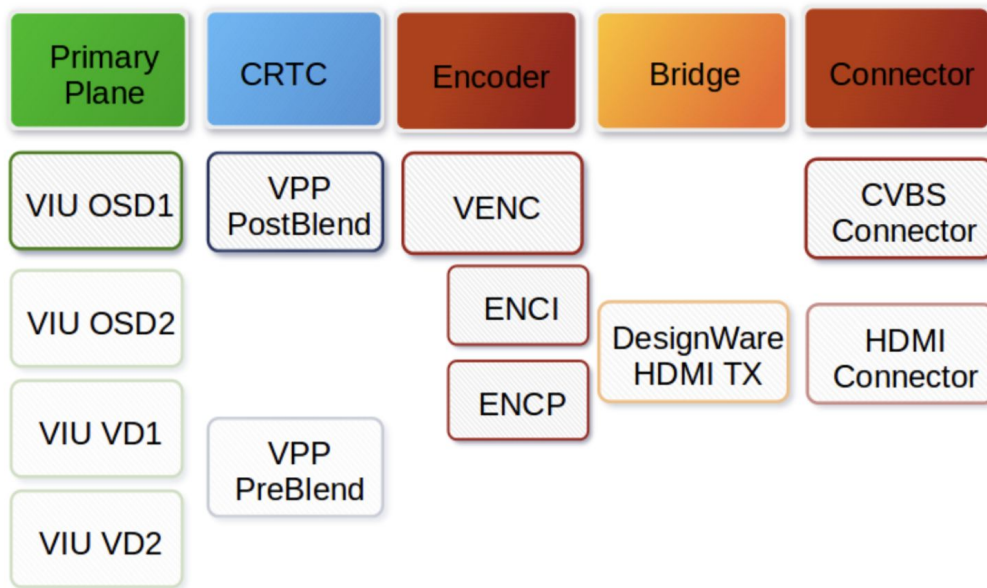


Figure 6-1. Display System Block Diagram

# Display Engines architecture

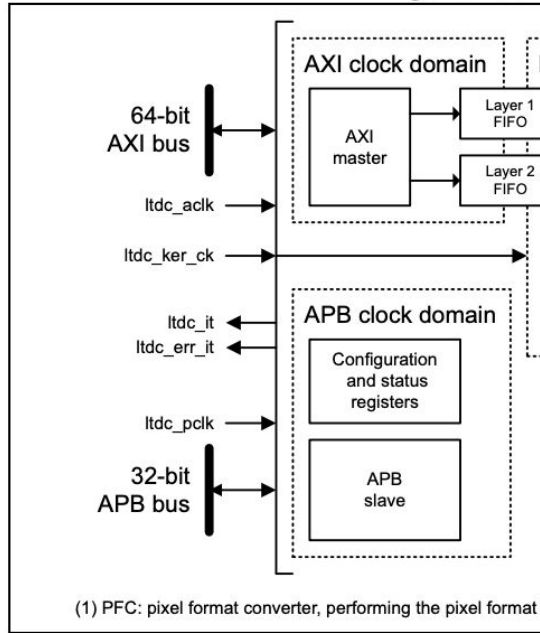
Amlogic GXL



Source: <https://people.freedesktop.org/~narmstrong/elcna-2017-amlogic/#/section-29>

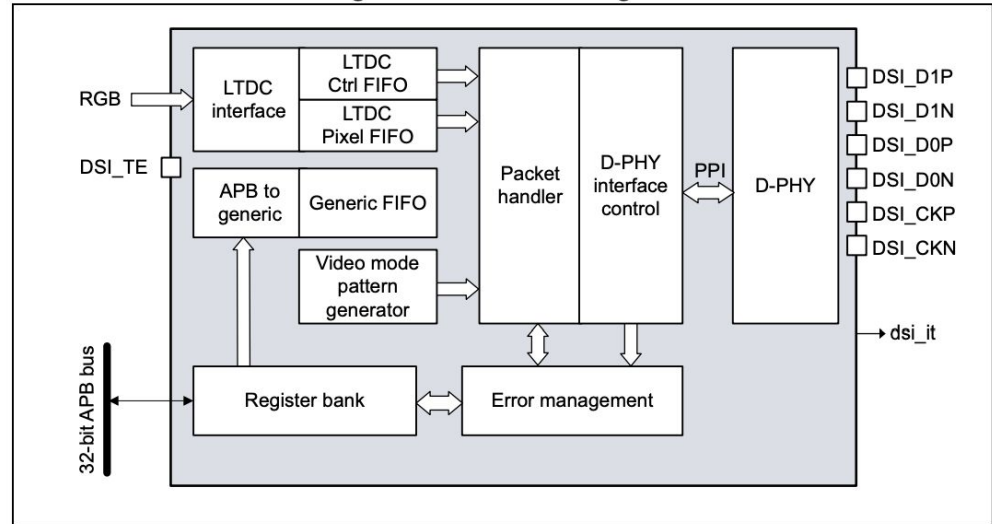
# Display Engines architecture

**Figure 280. LTDC block diagram**

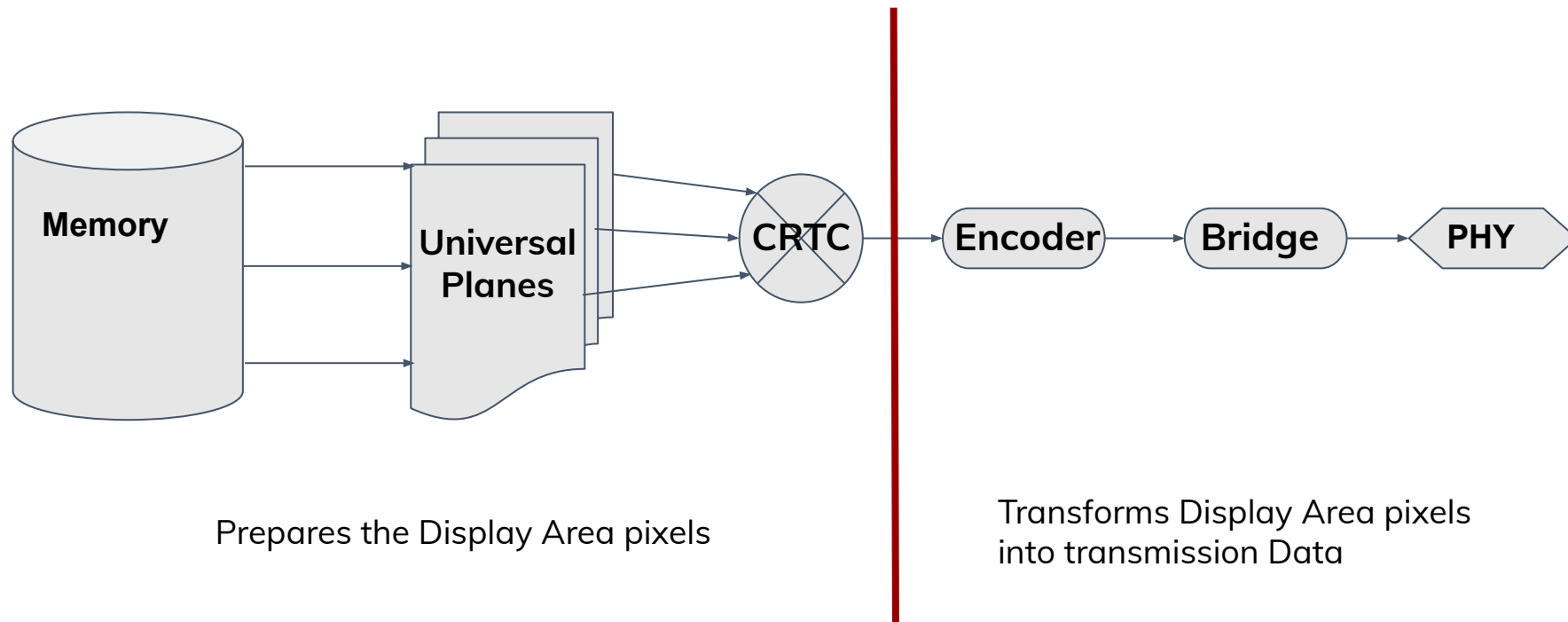


STM32MP1

**Figure 285. DSI block diagram**



# Display Protocols

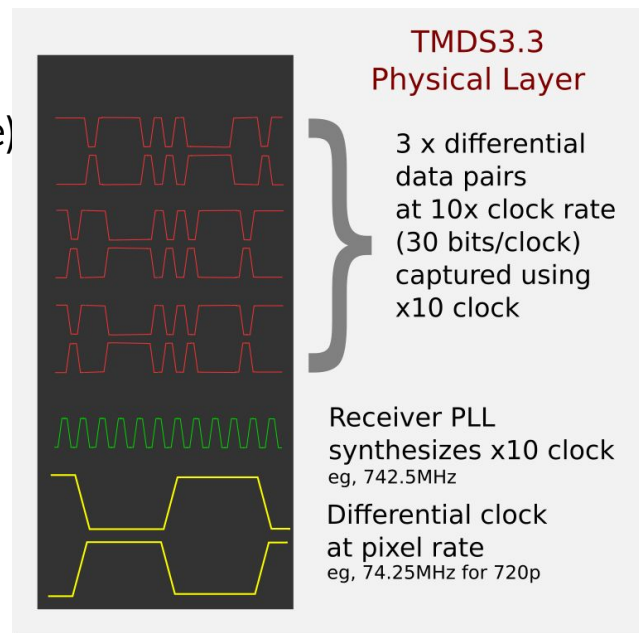




# Display Protocols

## HDMI

- Protocol: 3 × transition minimized differential signaling (TMDS) data and clock
- Is basically single-link DVI-D (Digital Visual Interface)
- Higher link frequencies
  - Up to 4k120 for HDMI2
- Added signaling packets into blanking
  - AVI InfoFrame
  - Audio Frames
- Supplementary Signals
  - CEC bus
  - (e)Audio Return Channel



# Display Protocols

## DisplayPort

- Modern Interface
  - Uses packetized data transmission
  - Embeds clock into micro-packets
  - Transmits Video & Audio
  - Variable number of lanes
- Half-duplex auxiliary channel
  - Handles non-AV communications
  - EDID, ...
  - Can transmit USB
- Extensible
  - Can transmit multiple channels (for multiple monitors)
  - eDP variant for fixed panels (laptops)
  - Supports up to 20.0 Gbit/s per lane in DP 2.0
  - Can transmit up to 8K 120Hz in DP2.0 4 lanes
  - Primary protocol used in USB-C Altmode



# Display Protocols

## MIPI DSI

DSI Spec is well hidden by MIPI Alliance, precise informations are sparse.

=> <https://www.mipi.org/specifications/dsi>

Operates on top of the MIPI D-PHY (or C-PHY for DSI2) physical layer

- High-Speed differential signaling point-to-point serial bus
- MIPI D-PHY is used for Display **and** Cameras

Video transmission is basically packets on the D-PHY bus

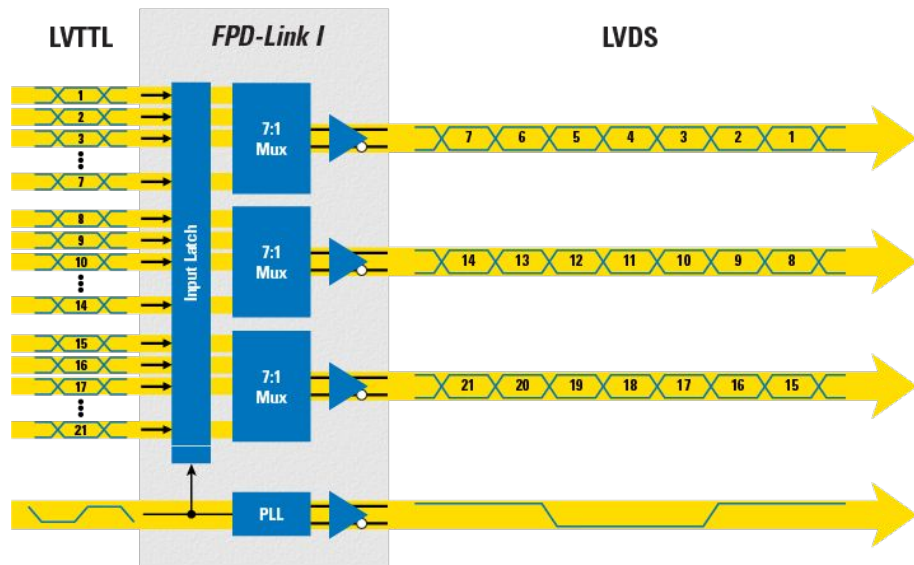
- Either directly translated from video stream
- Either grouped into packets (Burst mode)
- Either command mode
  - Direct write to controller's framebuffer memory

Supports VESA's Display Stream Compression, variable framerate, dual-links to large displays, ....

# Display Protocols

## LVDS

- Low-voltage differential signaling
  - It's the serialized version of the parallel bus signals
  - Can run at high speed
  - Uses inexpensive twisted pair copper
- Only Physical Layer in TIA/EIA-644
- FPD-Link is one of the protocols used
- Is (was ?) mainly used in Laptops
  - Now replaced with eDP



# Display Protocols

## Other protocols:

- Analog TV: Composite/S-Video/Component/SCART
- Analog Monitors: VGA/DVI-I/DVI-A
- MIPI DPI

## Not implemented in upstream kernel:

- V-by-One using SerDes as link layer
- OpenLDI
- ...

# Support limitations

## HDMI

- HDMI1.4 Unsupported features ([https://www.hdmi.org/download/savefile?bucket=hdmi-web-public&fileKey=Specifications/1dot4\\_feature\\_archive.pdf](https://www.hdmi.org/download/savefile?bucket=hdmi-web-public&fileKey=Specifications/1dot4_feature_archive.pdf))
  - HDMI Ethernet Channel
  - Audio Return Channel
  - Content Type (Intel supports it)
  - Additional Color Spaces
- HDMI2 Unsupported features ([https://www.hdmi.org/spec/hdmi2\\_1](https://www.hdmi.org/spec/hdmi2_1))
  - higher video resolutions (8K60 and 4K120, and resolutions up to 10K)
  - Dynamic HDR
  - Source-Based Tone Mapping (SBTM)
  - eARC (Enhanced Audio Return Channel)
  - Variable Refresh Rate (VRR)
  - Auto Low Latency Mode (ALLM)
  - Quick Frame Transport (QFT)
  - Quick Media Switching (QMS)
- Why? Specifications are closed.

# Support limitations

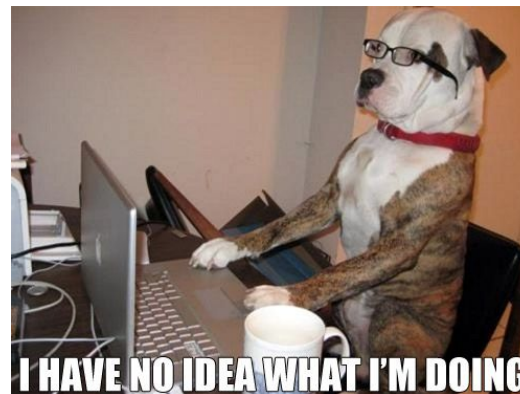
## DisplayPort

- Much Better!
- But almost only on Intel/AMD platforms...
- AMD
  - Supports up to UHBR20 - 20.0 Gbps/Lane
- Embedded platforms
  - Only some Mediatek, Qualcomm, Tegra, Xilinx, Rockchip SoCs supports DP
  - DP2.x Only supported on Intel/AMD
  - MST (Multiple Display support) only on Intel/AMD
  - USB-C Altmode supported on Qualcomm platforms only
- DP is a complex beast, embedded platforms are slowly catching up

# Support limitations

## MIPI DSI

- Hard to say, Why? Specifications are closed.
- At least those are missing:
  - Proper atomic panel initialization
  - Features handshaking between Controller and Bridge/Panel
  - Panel self-refresh support
  - Dynamic mode/rate switching (with Content Type)
- Panel controllers are at best partially documented
  - We're stuck with initialization table
  - Unable to implemented advanced features
  - Unable to support proper calibration data
- Without a DSI analyzer, we are blind
  - At some point it works, no idea why





# Support limitations

- Generally, SoC support for Display Protocols are barely documented
- There's some exceptions
  - STM32MP1 is one of those
- Most of the reference is from the Vendor Linux source
  - But vendor sometimes don't even use DRM
- Using shared IPs is great (like the Synopsys DSI host)
- But it's a challenge to know what's really supported....
  - IP version is known but without the Databook it's useless
  - We don't know how the IP was configured
  - We don't know the exact capabilities of the IP/PHY

# Standards paywall

- HDMI
  - Needs to be an HDMI Adopter to get the HDMI 2.1b spec
  - AFAIK there's no special way for Open-Source developers to get the spec
  - There's leaked HDMI spec up to 2.0 but it's illegal to use them
  - Cost: \$10k/year or \$5k/year + flat \$1/unit
- DisplayPort
  - Free Standards from VESA:  
Adaptive-Sync/Media-Sync CTS, Display Compression-M, High-performance Monitor and Display Compliance Test Specification, Display Stream Compression Standard, Net2Display Remoting, Access Bus, CVT1.2 and Spreadsheet, DCM, DDC/CI 1.1, DisplayID, DMS59, DMT 1.0 Rev 13, DPM Rel. A, DPVL, DSTP, E-DDC 1.2, EDID Extensions, EDID Implementation Guides, E-EDID Rel A 1.0, 2.0, E-EDID EEPROM, FDMI, GTF 1.1, GTF Help Files, M1, MCCS 2.2a, MCCS Updates, MDDI 1.2, MultiDisplay, Multiple Projector Common Data Interchange (MPCDI), NAVI, All Panel Standards, PnP, PSWG 15 and 17 inch, SMT, Stereo, VBE, VIP 2, VSIS 1.0 R2, and VSIS Test Procedure...
  - **But: The DisplayPort standard is available to VESA Members only.**
  - Cost: The fee structure is based on annual sales
  - AFAIK there's no special way for Open-Source developers to get the spec

# Standards paywall

- MIPI DSI/D-PHY/DPI
  - Needs to be a MIPI Alliance member to get access to DSI spec
  - Fees range from \$40,000 for founders and board members to \$4,000 for small companies that join at the adopter level
  - AFAIK there's no special way for Open-Source developers to get the spec

# Conclusion

- Embedded SoCs has non-heterogeneous support
- Variable support due to variable use-cases
  - Phones
  - Set-top-Box
  - Industrial Devices
  - Desktop
  - Laptops
  - ...
- Impossible to entirely define what's support to Userspace
- Impossible to properly implement:
  - All hardware features
  - All protocol features
  - All peripheral features
- Because of documentation:
  - Very limited vendor documentation
  - Paywall of IP specifications/Databooks
  - Paywall for Protocols Specs

Thank you!  
Do you have any questions ?

Get the slides at: [bit.ly/46wrEKy](https://bit.ly/46wrEKy) or Scan:

