A JOURNEY THROUGH UPSTREAM ATOMIC KMS TO ACHIEVE DISPLAYPORT COMPLIANCE

XDC 2017

Manasi Navare
Intel Open Source Technology Center
Every end user’s dream – no black screens

Make Intel graphics kernel driver **DisplayPort** compliant and **upstream** it

*Other names and brands may be claimed as the property of others.*
“A journey of a thousand miles begins with a single step”
What is DP Compliance?

*Other names and brands may be claimed as the property of others.*
What happens when you connect a DP cable?

[Diagram showing a DP cable connecting a PC (DP Source) to a Monitor (DP Sink).]

- **PC (DP Source):**
  - Main-Link
  - AUX CH
  - HPD

- **Monitor (DP Sink):**
  - HPD
  - AUX CH
  - Main-Link

- **DP Cable:**
  - Hot Plug Detect Signal
  - DPCD Read/Write
  - Serialized/Encoded Data at Link Clock
Display Port Link Training

- Main Link Parameters:
  - Lanes – 1, 2 or 4
  - Link Rate – 1.62, 2.7, 5.4 or 8.1 Gbps/lane

- Link Training:
  - DP source configures the main link through link training sequence
How to test DP compliance?

- **Device Under Test**: DP source
- **DP Monitor**: DP sink
- **DP reference sink**: Monitor Out
- **Compliance Test Suite SW**: USB

*Other names and brands may be claimed as the property of others.*
Who is responsible for DP Compliance?

Intel Integrated Graphics Device - GPU

**Surfaceflinger/HW Composer**

**3D Graphics**

**Libdrm**

**Mesa (3D Graphics Driver)**

**Userspace**

**Linux Kernel**

**Intel Graphics Driver I915**

**Intel Integrated Graphics Device - GPU**

*Other names and brands may be claimed as the property of others.*
Kernel’s Responsibility - Mode setting

Process of setting up clocks, scanout buffers, initializing the chip and lighting up displays

- Frame Buffer (memory)
- CRTC
- Encoder
- Connector
- Display 1
- Display 2
- Display 3
- Planes (memory)
- Scanning buffers
- Configuring the chip
- Lighting up displays
Atomic KMS – Making Kernel and GPU play nicely

An attempt of making “Every Frame Perfect”

An Atomic Operation is the one that appears to take place as a single indivisible operation
Atomic KMS – Making Kernel and GPU play nicely

An attempt of making “Every Frame Perfect”

Diagram:
- Properties
  - Userspace
  - DRM_IOCTL_MODE_ATOMIC
- Kernel
  - state
    - plane
    - CRTC
      - connector
      - ->atomic_check()
    - device
      - plane
      - CRTC
      - connector
      - ->atomic_commit()
- DP cable
Finally got the ball rolling!!!

Wait, did I

????
“Anything that can go wrong, will go wrong”

DP compliance test FAILURE…
Problem: Does the Atomic KMS driver handle link failures?

Link Failures, black screen…
Solution:

Stable link = Successful Modeset = Perfect frame

Userspace

Kernel

Modeset 3840x2160@60

Validate mode

Check()

Commit()

Link Training

Link Training fails at 5.4 Gbps, 4 lanes

Uevent indicating HW configuration changed

Link Status = BAD

Fallback to 2.7Gbps, 4 lanes

Retry Modeset on the new preferred mode

Success, mode set to 2560x1600@60

Link Status = GOOD

Link Re-Training

Link trained at 2.7 Gbps, 4 lanes

Success mode
Failure is always an Option….

1. Atomic check guarantees the requested mode
   - Can only check GPU parameters, not the physical DP cable
   - Link training can still fail

2. Link Failure can be asynchronous
   - Link might fail after a successful modeset

3. Atomic allows non blocking commits
   - Return to userspace before modeset has completed

*Asynchronous handling of link failures extremely critical
Upstreaming Challenge - Coordinating Compliance

Both Userspace and Kernel responsible to ensure 100% DP Compliance

- Handle
- Uevent
- Check Link Status
- Retry
- Modeset

- Link Train Fallback
- Set Link Status
- Send Uevent
- Link Retraining

*Both Userspace and Kernel responsible to ensure 100% DP Compliance*

- Old Userspace with new Kernel – Not Compliant
- New Userspace with old Kernel – Not Compliant
Intel’s Responsibility – NO REGRESSIONS

- Handle Compliance Test request
- Set compliance test_active
- Read DPCD and write to debugfs node
- Uevent on link failure

Compliance Test Request from DPR 120

./Intel_dp_compliance
IGT
DRM Master

Debugfs/Hotplug Uevent

DRM_IOCTL_GETCONNECTOR
DRM_IOCTL_SETCRTC

Short Pulse

INTEL GRAPHICS I915

- Set up FB/ video pattern, do a modeset on requested mode
- Handle uevent and redo a modeset

- Handle Compliance Test request
- Set compliance test_active
- Read DPCD and write to debugfs node
- Uevent on link failure
Future steps

- Replace Unigraf’s DPR-120 with Google’s Chamelium Board
  - Allows testing all connector types (external displays)
  - Open Source HW - extend to add all corner cases
- Port the DP compliance test suite (as per VESA CTS) to Chamelium
- Port the IGT intel_dp_compliance tool to Chamelium testing tool.
- Add DP Compliance testing to Pre-merge CI systems
My Journey of thousand miles.....

- Steep Learning curve
- Submitting patches – First step out of Comfort Zone
- Community was very Helpful, constructive feedback
- You will see a finish line – Don’t give up!
Thank you

Kernel - Daniel Vetter, Jani Nikula, Ville Syrjala, Rodrigo Vivi, Jim Bride, Matt Roper, Harry Wentland, Sean Paul

X - Martin Peres, Chris Wilson, Eric Anholt, Adam Jackson

IGT - Petri Latvala
We are DP Compliant as of Linux Kernel 4.12 and xorg-server-1.19.3

Questions?

Manasi Navare: manasi.d.navare@intel.com