

Qualcomm DRM: handling runtime requirements for device components

Dmitry Baryshkov
Senior Staff Engineer

dmitry.baryshkov@oss.qualcomm.com

Snapdragon and Qualcomm branded products are products of Qualcomm Technologies, Inc. and/or its subsidiaries.
Qualcomm patents are licensed by Qualcomm Incorporated.



DRM KMS modesetting

Two stages of modeset

- **atomic_check**
 - Performs a check on the passed commit
 - Can fail
 - Can be executed multiple times without the actual modeset
- **atomic_commit**
 - Reprogram the hardware
 - Is not allowed to fail

DRM/KMS modesetting

Anatomy of the commit

- Execute `atomic_pre_enable()` / `atomic_enable()` / `atomic_disable()` / `atomic_post_disable()` for each KMS object
- None of the callbacks can fail
- KMS objects: CRTC, planes, encoders...
- And bridges
 - Bridges...

DRM/kms: runtime PM

- `drm_atomic_helper_commit_tail_rpm()`
 - call `pm_runtime_get_sync()` in `atomic_enable(crtc)`, one of `pm_runtime_put()` in `atomic_disable(crtc)`
- Bridges
 - Also use `pm_runtime` in callbacks

DRM/kms: runtime PM and not only

- `drm_atomic_helper_commit_tail_rpm()`
 - call `pm_runtime_get_sync()` in `atomic_enable(crtc)`, one of `pm_runtime_put()` in `atomic_disable(crtc)`
- Bridges
 - Also use `pm_runtime` in callbacks
- Non-RPM resources: PHYs, regulators, clocks, etc.

DRM/kms: runtime PM and not only

- `drm_atomic_helper_commit_tail_rpm()`
 - call `pm_runtime_get_sync()` in `atomic_enable(crtc)`, one of `pm_runtime_put()` in `atomic_disable(crtc)`
- Bridges
 - Also use `pm_runtime` in callbacks
- Non-RPM resources: PHYs, regulators, clocks, etc.
- Each can fail to be enabled. The callbacks can't.
 - `Is_enabled` status flags

DRM/KMS: proposal

- Add power_up / power_down callbacks
- Calculate the “power” status of the `drm_atomic_commit`
- Power on new components when we still can atomically fail

DRM/KMS: proposal

- Add power_up / power_down callbacks
- Calculate the “power” status of the `drm_atomic_commit`
- Power on new components when we still can atomically fail

Drawbacks:

- driver and hardware constraints
- Sequence, locking, etc.
- `Phy_configure()` or `clk_set_rate()` still can fail

Alternative proposals?

Thank you

Nothing in these materials is an offer to sell any of the components or devices referenced herein.

© Qualcomm Technologies, Inc. and/or its affiliated companies. All Rights Reserved.

Qualcomm and Snapdragon are trademarks or registered trademarks of Qualcomm Incorporated.
Other products and brand names may be trademarks or registered trademarks of their respective owners.

References in this presentation to “Qualcomm” may mean Qualcomm Incorporated, Qualcomm Technologies, Inc., and/or other subsidiaries or business units within the Qualcomm corporate structure, as applicable. Qualcomm Incorporated includes our licensing business, QTL, and the vast majority of our patent portfolio. Qualcomm Technologies, Inc., a subsidiary of Qualcomm Incorporated, operates, along with its subsidiaries, substantially all of our engineering, research and development functions, and substantially all of our products and services businesses, including our QCT semiconductor business.

Snapdragon and Qualcomm branded products are products of Qualcomm Technologies, Inc. and/or its subsidiaries. Qualcomm patents are licensed by Qualcomm Incorporated.

Follow us on: [in](#) [X](#) [@](#) [v](#) [f](#)

For more information, visit us at [qualcomm.com](https://www.qualcomm.com) & [qualcomm.com/blog](https://www.qualcomm.com/blog)

