

Display Power Saving Feature Overview

Contents:

UBWC (Universal Bandwidth Compression)

SDE(Snapdragon Display Engine) Rotator UBWC

Mixed Mode Composition(GPU and SDE)

Static Display Refresh

Partial Update Feature

DSC (Display Stream Compression)

MIPI DSI UPLS Mode

Dynamic FPS Changes

Idle Power Collapse for SDE

Low Resolution Rendering Feature

CABL (Content Adaptive BackLight Scaling)

FOSS (Fidelity Optimized Signal Scaling)

UBWC(Universal BandWidth Compression)

- UBWC
 - Reduce the bandwidth
 - Optimize the power saving
- How to disable UWBC from display/GFX perspective
 - adb shell setprop debug.gralloc.enable_fb_ubwc 0
 - adb shell setprop debug.gralloc.gfx_ubwc_disable 1
 - adb shell stop
 - adb shell start

- Note
 - UBWC is used by MultiMedia side, such as Venus, GPU, CPP, etc.
- Document:
 - 80-NM328-41_A_UBWC_Introduction
 - 80-NV396-100_A_Universal_Bandwidth_Compression

SDE(Snapdragon Display Engine) Rotator UBWC

- Rotator UBWC
 - Rotator is used by Video use-case
 - Rotator is used by Camera use-case
- How to disable rotator UBWC
 - adb shell setprop sdm.debug.rotator_disable_ubwc 1
 - adb shell stop
 - adb shell start
- Note
 - SDE name is same to MDP(Mobile Display Process).
 - Reduce the memory bandwidth and power for downscaled video

Mixed Mode Composition (GPU and SDE)

- Mixed mode composition
 - Use combination of GPU and SDE composition when SDE resources are insufficient
 - One strategy is static layers composed by GPU, and updating layers by SDE
- The legacy platform(such as 8939, 8976, 8952)
 - How to disable mixed mode composition
 - Øset debug.mdpcomp.mixedmode.disable to 1 in build.prop file
 - Øadb root
 - Øadb shell setprop debug.mdpcomp.mixedmode.disable 1
 - Øadb shell stop
 - Øadb shell start
- The latest platform (8996, 8953, 8937 based on SDM)
 - Not have the property to directly disable mixed mode composition
- Enable SDM user space logs:
 - adb shell service call display.qservice 15 i32 0 i32 1 i32 1
- The relevant document:
 - 80-NV396-37_C_Snapdragon_Display_Manager_User_Guide.pdf

Static Display Refresh

- Static display refresh is only applicable for video mode panel
 - When idle screen is detected (4 refresh cycles worth of inactivity)

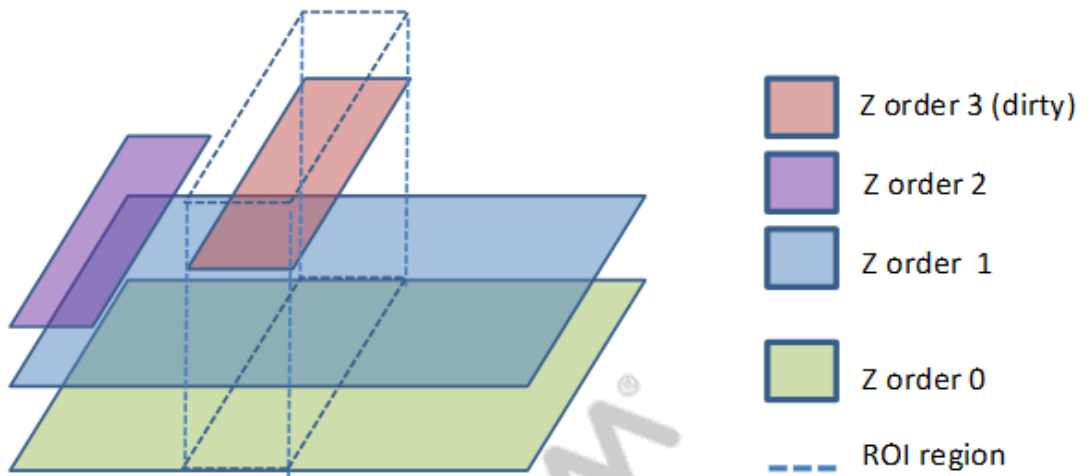
- Copy all layers once to FBT (frame buffer target) and SDE fetches from the FBT subsequently
- 4 refresh cycles which value is the experienced value.
- The legacy platform(such as 8916, 8939, 8952, 8976)
 - How to change the idle timeout value
 - Øadb root
 - Øadb shell setprop debug.mdpcomp.idletime 600
 - Øadb shell stop
 - Øadb shell start
- How to set idle timeout based on SDM
 - adb shell service call display.qservice 16 i32 1000
 - 16 is the code for SET_IDLE_TIMEOUT, 1000 is time in ms
- How to disable idle timeout
 - adb shell service call display.qservice 16 i32 0
- Note
 - Set the sdm.idle_time in build.prop
 - The default value is 70ms, in other words, 4 refresh cycles

Partial Update Feature

- Partial Update
 - Update only dirty regions
 - e.g. Wi-Fi icon updates on status bar, and blinking cursor on messaging apps
- How to enable partial update
 - qcom,partial-update-enabled;
 - qcom,panel-roi-alignment = <4 2 4 2 20 20>;
- Note:
 - In /kernel/Documentation/devicetree/bindings/fb/mdss-dsi-panel.txt

qcom,panel-roi-alignment: Specifies the panel ROI alignment restrictions on its left, top, width, height alignments and minimum width and height values, please check with panel vendor.

 - qcom,partial-update-roi-merge is for dual DSI
- Document:
 - 80-NL235-1_A_Partial_Update_Feature_Smart_Panels_Android
- Partial update overview



Z order 2 doesn't overlay with the ROI stack and will not be considered during composition.

- Note that

- Partial Update and Post Processing(CABL/FOSS/SVI/AD) will be mutually exclusive.
- When PU is enabled, /sys/class/graphics/fb0/dyn_pu node should be true.
- ROI in the SurfaceFlinger. If the partial update is enabled, it can be smaller than the full screen size.

- Enable partial update with Qservice

- adb shell service call display.qservice 19 i32 0 i32 1

- Disable partial update with Qservice

- adb shell service call display.qservice 19 i32 0 i32 0

DSC(Display Stream Compression)

- DSC

- Display Stream Compression
- Compression ratios is 3:1

- FBC

- Frame Buffer Compression
- Compression ratios is 2:1 and 3:1

- Note:

- DSC V1.1 is supported on 8996
- FBC is supported on 8994
- If want to support DSC or FBC, need panel side support this as well.

- There are 2 independent DSC encoders within SDE for 8996/8976 and 8998.

- Picture dimension

- Spec - it is the uncompressed size of the image and conveyed to DSC decoder through PPS (Picture Parameter Set) command.

-SDE - It is the uncompressed size of the image being transferred, Layer Mixer out size

ØWhen split LM is used, Picture Width is summation of both LM width.

- Slice - Minimum encoded unit

- Per encoder maximum 2 slices per line.

- When 2 encoders are working in tandem, maximum 4 slices per line.

- Multiple of Slice width and height must be equal to Picture width and height, respectively.

- DSC split panel

- This field is set when both DSC encoders are working in tandem where each is compressing half of Picture Width.

- This is available for dual DSI interface.

- DSC Merge

- It merges the output of two DSC encoders and sends the compressed bit-stream to single DSI controller.

- DSC Topology, e.g

- 2CTL->2LM->2DSC_EN->2DSI (max 4 slices per line)

- Picture Dimension = 2160x3840

- Minimum slice width = $(2160 / 4) = 540$

- DSC topology configuration

- Need to enable DSC for 4K panel, also both DSC encoders

- For 2K panel, it better to enable DSC as well.

- How to enable DSC for 2K panel (1440x2560)

- qcom,compression-mode = "dsc";

- qcom,config-select = <&dsi_dual_nt35597_sharp_video_config0>;

- dsi_dual_nt35597_sharp_video_config0: config0 {

- qcom,mdss-dsc-encoders = <1>;

- qcom,mdss-dsc-slice-height = <16>;

- qcom,mdss-dsc-slice-width = <720>;

- qcom,mdss-dsc-slice-per-pkt = <2>;

- qcom,mdss-dsc-bit-per-component = <8>;

- qcom,mdss-dsc-bit-per-pixel = <8>;

- qcom,mdss-dsc-block-prediction-enable;

- Note:

- Please see kernel/msm-3.18/Documentation/devicetree/bindings/fb/mdss-dsi-panel.txt

- How to configure the DSC parameters:

- Split panel is false

- ØIf split_panel=0, slice_width must be equal to either picture_width, or picture_width/2

- ØIf slice_width = picture_width/2, picture_width must be even

-Split panel is true

ØIf split_panel=1, slice_width must be equal to either picture_width/2 or picture_width/4

ØIf slice_width = picture_width/2, picture_width must be divisible by 2

ØIf slice_width = picture_width/4, picture_width must be divisible by 4

•Note

-In all use-cases, slice_height must be equal to picture_height/N, where $N \geq 1$

-slice_height * slice_width should be $\geq 10,500$ pixels, as per the DSC Task Group recommendation

MIPI DSI ULPS mode

•ULPS mode

-MIPI DSI PHY Spec - Ultra Low Power State (ULPS) mode

-DSI PHY driving '00' to the data lanes

•Enter ULPS sequence:

-Escape mode entry command:11>10>00>01>00

-Ultra-Low Power State Entry Command:00011110

•Exit ULPS sequence:

-LP00 -> LP10 -> LP11

•How to enable ULPS mode

-qcom,ulps-enabled;

•Note:

-ULPS mode is to reduce the power consumption from DSI Slave side.

•ULPS entry sequence waveform



•Note:

- Need to check panel side if support the ULPS mode as well

Dynamic FPS Changes

•Dynamic FPS

- Changing H/V porch value, DSI Bit clock is no changing
- Changing PLL

•Note:

- Dynamic FPS is applicable for video mode panel only.
- If want to enable dynamic FPS function, please confirm with panel vendor if can support the dynamic FPS changes from panel side as well
- We know that on some panels, when enable dynamic FPS function, panel flickering issue happened.

•Document:

- 80-P1155-1_A_Dynamic_FPS_Feature

Idle Power Collapse for SDE

•IPC:

- Reduce the power consumption in idle use-cases.

•How to enable IPC

- Add the below parameters in dtsi file
- qcom,mdss-idle-power-collapse-enabled;

- Note

- This function is only support for smart panel (command mode panel).
- This is supported on 8998, 8996, 8953, 8937.
- This is not supported on 8952, 8976

Low Resolution Rendering Feature

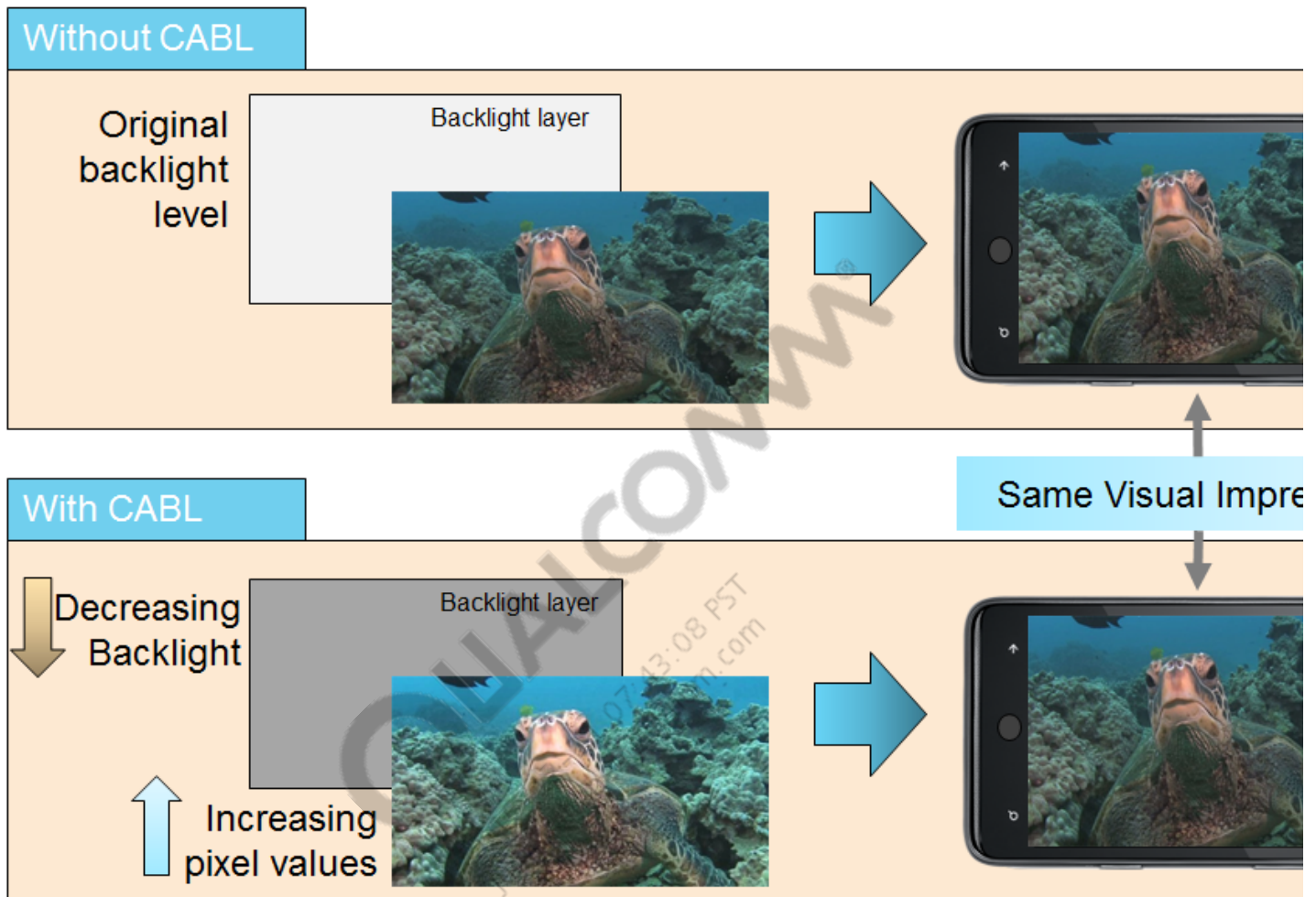
- Low resolution rendering for some specified application
 - GPU rendering in smaller resolution, use SDE to upscale
 - Reduce the game application resolution to lower than the display panel resolution and then upscale the content as part of composition
 - Reduce the load on the GPU and help improve power and performance numbers. e.g: Game is rendered at 720x1280, while display resolution is 1080x1920.
 - The touch coordinates also need appropriate mapping.
 - This feature is only applicable for primary display, no external/virtual display
- Document:
 - 80-P1830-1_A_Low_Resolution_Rendering_Feature
- This feature is disabled by default.
- How to enable or disable this feature
 - For 1080p display resolution, set the override resolution to 720p.
- Enable:
 - adb shell setprop "persist.debug.app_res_override" 720X1280
- Disable:
 - Setting the property to 0 or undefined will disable the feature.
- Note:
 - According to OEMs requirement, can add the specified app name information in whitelistedapp.xml file

CABL (Content Adaptive BackLight Scaling)

- This feature is available for LCD panel, not OLED panel.
 - Boosting up the pixel
 - Reduce the backlight level for power saving
- How to enable/disable CABL
 - Disable - set ro.qualcomm.cabl to 2 in build.prop file
 - Enabled - set ro.qualcomm.cabl to 0
- Note
 - This feature is similar with CABG (this panel solution), which is from panel side.
 - CABL feature is free feature
 - CABL is used for indoor in general
- Document

-80-NP952-1_A_Content_Adaptive_Backlight_OEM_Design_Guide.pdf

-80-NP952-1_B_Content_Adaptive_Backlight_Design_Guide.pdf



- How to use the CABL configuration xml

- Add below property in build.prop

- Øconfig.cabl.xml=1

- Øconfig.cabl.path=/data/misc/display/CABLConfig.xml

- Øconfig.cabl.xml.print=1

- Use the PPD Socket to disable/enable CABL on the fly

- Relevant document:

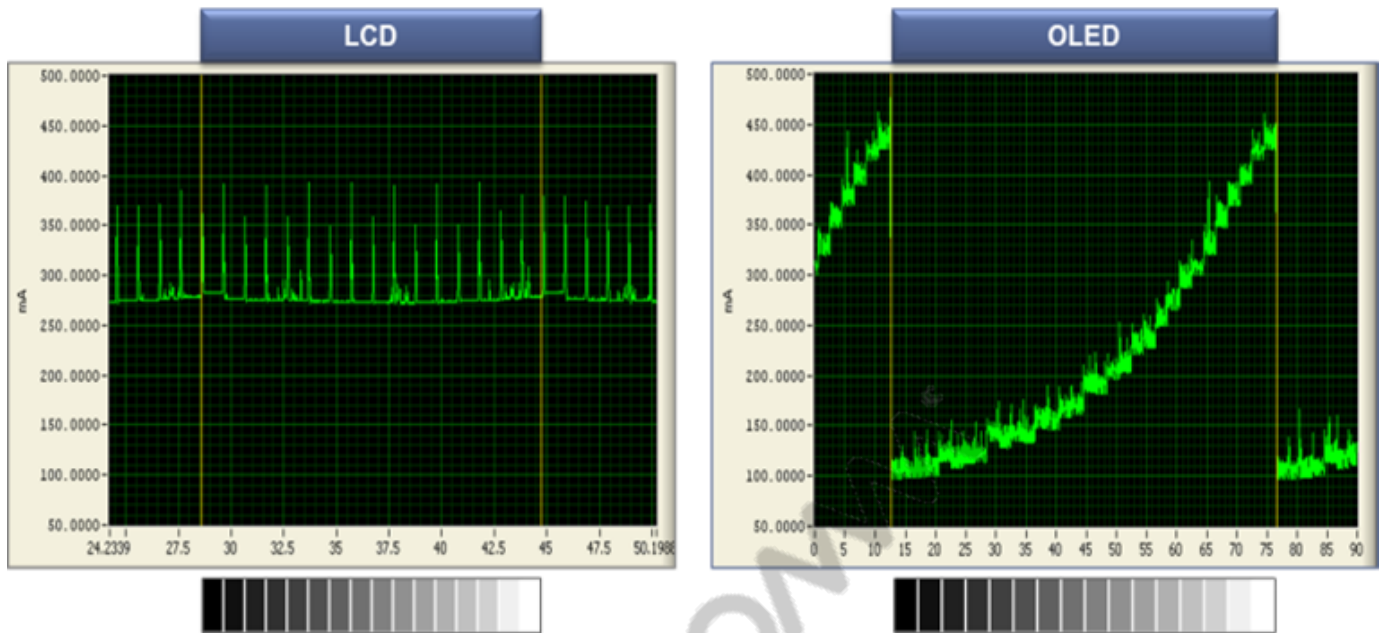
- 80-P2271-1_A_Display Postprocessing Service.pdf

- Note

- We have integrated the default CABL configuration in codes, if OEMs want to tune those CABL parameters, please raise the SR to Qualcomm support team to obtain the configuration xml file

FOSS (Fidelity Optimized Signal Scaling)

- Power consumption curve by LD and OLED natively



- FOSS Power savings for OLED panel.
 - Decrease the pixel value, but meanwhile make sure the visual quality
- How to enable FOSS (for example on 8996)
 - Set ro.qualcomm.foss to 1
 - Set ro.qualcomm.display.paneltype to 1
- Note:
 - If set this to 0, it means that LCD panel, FOSS is not supported on LCD panel
 - FOSS is used for indoor in general
 - By panel side, have the similar function, e.g. ACL
- Document
 - 80-NV610-94_B_FOSS_Feature_Description.pdf
- Note that target power could be adjusted based on content.



Color could be different!

Panel	Original (left)	Processed (right)	Power Saving
Panel I	449mA	350mA	22%

- How to use the FOSS configuration xml
 - Add below propertied in build.prop (for example on 8996)
 - Øconfig.foss.xml=1
 - Øconfig.foss.path=/data/misc/display/FOSSConfig.xml
 - Øconfig.foss.xml.print=1
- Use the PPD Socket to disable/enable FOSS on the fly
- Relevant document:
 - 80-P2271-1_A_Display Postprocessing Service.pdf
- Note
 - We have integrated the default FOSS configuration in codes, if OEMs want to tune those FOSS parameters, please raise the SR to Qualcomm support team to obtain the configuration xml file
- FOSS can be supported on 8952, 8956/76, and 8996, 8953, 8937, 8998 and onwards
- How to enable FOSS on the legacy platform (such as 8952)
 - Add the following in build.prop:
 - Øro.qualcomm.cabl=2
 - Øro.qualcomm.display.paneltype=1 for OLED panel.
- Note
 - ro.qualcomm.display.paneltype=0 is reserved for LCD