Qualcomm Technologies, Inc.

# FOSS Feature Description

80-NV610-94 B

November 6, 2015

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

# Revision history

| Revision | Date | Description |
|---|---|---|
| A | August 2015 | Initial release |
| B | November 2015 | Updated Chapters 3 and 4, and Appendix A |

# Contents

# Figures



# Tables

# 1 Introduction

## 1.1 Purpose

Fidelity Optimized Signal Scaling (FOSS) is a Qualcomm Technologies, Inc. (QTI) display feature to reduce power consumption for OLED panels. It aims to achieve an optimal visual experience of the displayed image based on Human Visual System (HVS) metrics, while still saving power on the OLED panel. The feature operations are dependent on parameters specific to panel characteristics, display content, and other software configuration. This document is a guide to enable and operate FOSS on OEM target devices. It focuses on FOSS workflow, description and tunings for FOSS parameters, and the software code structure.

The reader must be familiar with the specific target and the MDSS/AMSS source code architecture. To check the availability of this feature on your product, contact QTI support (support.cdmatech@qti.qualcomm.com).

## 1.2 Conventions

Function declarations, function names, type declarations, attributes, and code samples appear in a different font, for example, `#include`.

Shading indicates content that has been added or changed in this revision of the document.

## 1.3 Technical assistance

For assistance or clarification on information in this document, submit a case to QTI at https://createpoint.qti.qualcomm.com/.

If you do not have access to the CDMATech Support website, register for access or send email to support.cdmatech@qti.qualcomm.com.

# 2 FOSS design

This chapter details the design challenges for FOSS and its software design on QTI devices.

## 2.1 FOSS software design

FOSS is a part of the Adaptive Brightness/Backlight Adjustment (ABA) framework that includes:

- Content Adaptive Backlight Scaling (CABL) – Power-savings for LCD panels
- Fidelity Optimized Signal Scaling (FOSS) – Power-savings for OLED panels
- Sunlight Visibility Improvement (SVI) – Improves the visual experience outdoors

ABA shares the hardware block with different software-implemented core algorithms.

Figure 2-1 shows a high-level workflow of FOSS. The FOSS core takes the histogram and panel characteristics. By analyzing the content, FOSS generates a target LUT and gradually updates the actual pixel mapping LUT based on predefined temporal speeds.

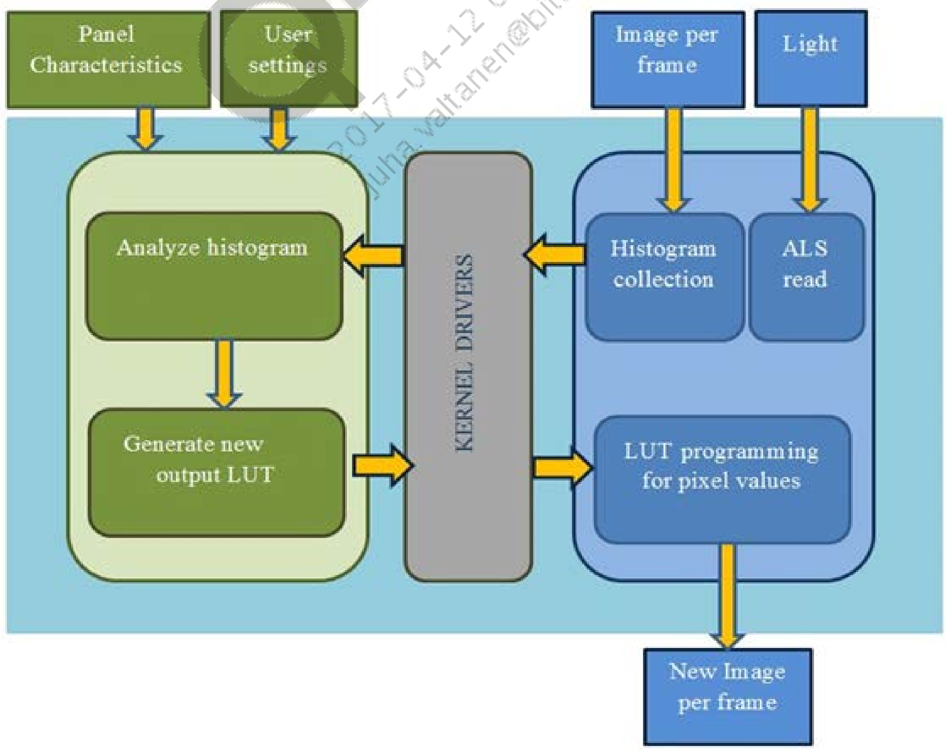


**Figure 2-1 High-level diagram of FOSS workflow**

**NOTE:** If there are other solutions to improve visibility, they are mandated to perform exclusively from FOSS. For example, SVI or Assertive Display (Apical AD) is active in outdoor conditions and

FOSS is active in indoor conditions. Figure 2-2 shows how FOSS works when ambient light is less; beyond a threshold value of ambient light, SVI/AD starts to work and FOSS stops.

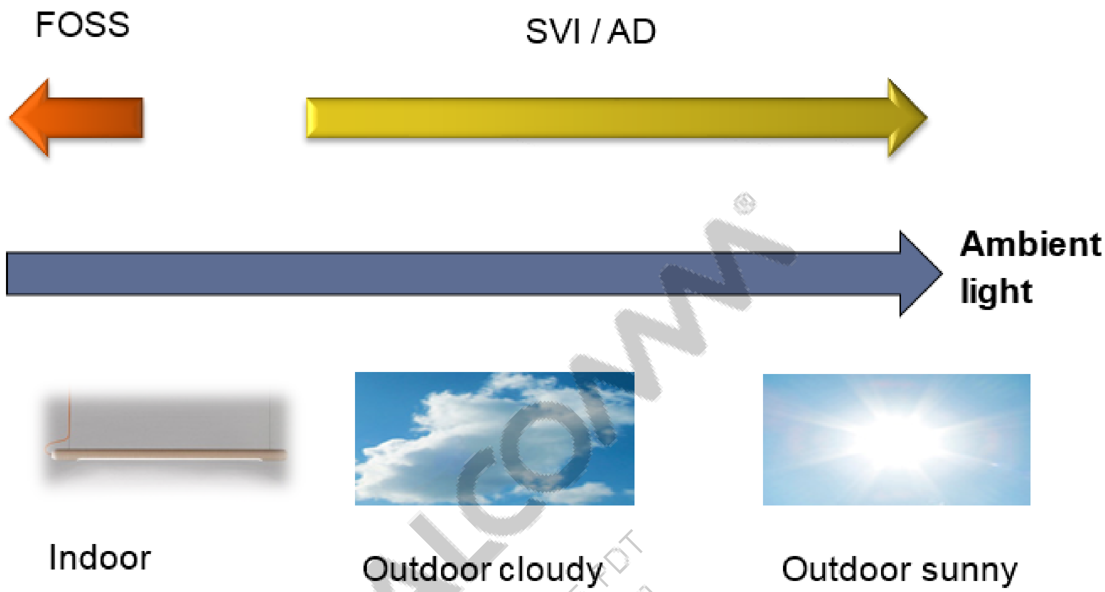


Figure 2-2 FOSS and SVI/AD getting activated based on ambient light

## 2.2 Differences between OLED and LCD panels

LCD and OLED are two physically different types of mobile display panels due to different manufacturing techniques applied. The major difference between OLED and LCD panels is that OLED panels do not need backlight.

An example experiment can be used to understand the power consumption difference:

1. Display a solid gray level image from black (0) to white (255) on the device.
2. Set brightness to the highest (about ~300 nit with 255 pixel value white image).
3. Perform battery level measurement with constant system power (power consumption except for display).

The observations are as follows:

- LCD power remains constant
- OLED power increases exponentially with increase in pixel value

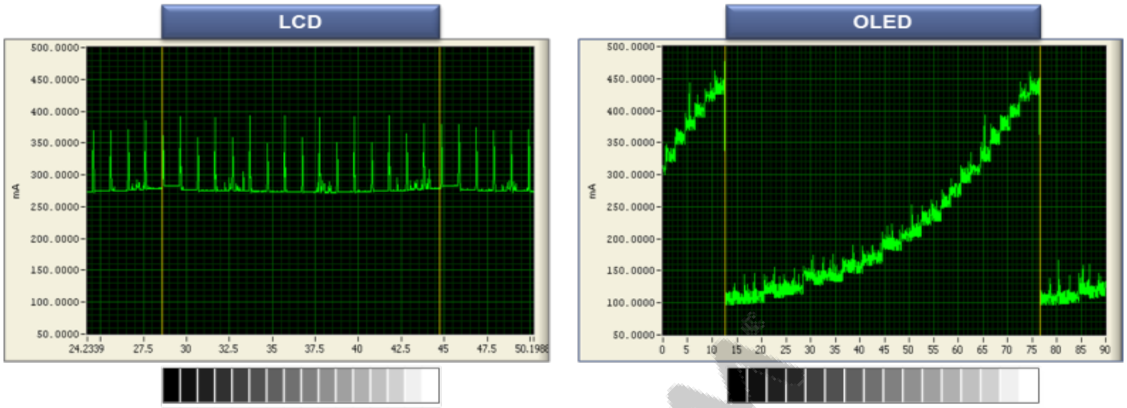


**Figure 2-3  Power consumed by LCD and OLED (baseline power included)**

Power-saving scheme for LCD panels is well developed (CABL) and works on boosting pixel values with lowering backlight. The same technique does not apply to OLED panels since OLED panels do not have backlight. In addition, power consumption of OLED panels is directly proportional to the pixel intensity. The brighter the content, the higher the power consumption.

## 2.3 FOSS design challenges

Developing a power-saving scheme for OLED has the following challenges:

- As OLED panels do not have backlight, global/local backlight dimming algorithms do not work.
- The brightness level and LUT processing in AMOLED cannot be separated because power is directly related to the pixel luminance; scaling down the dynamic range affects the visual quality.
- Power consumption differs among *R*, *G,* and *B* components and the curves are nonlinear. Hence, conventional color processing power-saving methods cannot be used for image/video applications.

## 2.4 FOSS quality

FOSS works on the principal of exploiting panel-side power while providing optimal visual experience of the displayed image based on Human Visual System (HVS) metrics. There are no visual quality degradation metrics applied with FOSS enabled and customers may apply the same at their end to tune the FOSS output.

Refer to Figure 2-4 through Figure 2-6 for illustrations of internal power numbers tested with visual example images.

**Figure 2-4 Panel power with Black content – 115.72 mA**

**Figure 2-5 Panel power with bright content without FOSS applied – 222.42 mA**

Figure 2-6 Panel power with bright content with FOSS – 197 mA

# 3 FOSS configuration process

NOTE: Numerous changes were made in this chapter.

## 3.1 Enable FOSS

The following properties need to be added by updating the device's /system/build.prop file to enable FOSS:

- `ro.qualcomm.cabl = 2`
- `ro.qualcomm.display.paneltype = 1`

**NOTE:** If the "ro.qualcomm.display.paneltype" property is not set, CABL algorithm for LCD device is executed.

## 3.2 Configure FOSS parameters using XML

FOSS parameters can be configured using an XML file based on the panel characteristics and user preference. This can only be performed after enabling FOSS, as described in Section 3.1. The XML interface is not enabled by default; certain properties must be added.

To enable the XML interface:

1. Open the /system/build.prop file.
2. Add the following lines to the file:

```
config.cabl.xml=1
config.cabl.path=<path_to_XML_file_on_device>
```

If the XML file is not found, FOSS falls back to the default parameters. A sample configuration XML file is provided in Appendix A. The procedure to populate FOSS configuration XML is described in Section 3.3.

## 3.3 FOSS tuning

FOSS tuning parameters can be broadly classified into two categories:

- Power-related – Tuned by customers
- Quality-related – Recommended by QTI

### 3.3.1 Power-related, customer-tunable parameters

The core principal of FOSS is to exploit the panel-side power with the same perceptual visual quality. There are two power-related tuning parameters:

- Panel power profile
- Power saving ratio

**Table 3-1 FOSS customer-tunable parameters**

| Parameter | Description | Default Value |
|---|---|---|
| PowerProfileTableLength | An integer value (0-255) representing the number of elements present in the panel power profile table. | 8 |
| PowerProfileTableInput | Each element is an integer value (0-255) representing the pixel brightness, with 0 being the darkest and 255 being the brightest. | 0, 36, 72, 109, 145, 182, 218, 255 |
| PowerProfileTableOutput | Each element is an integer value (0-1023) representing normalized panel power consumption for the corresponding pixel brightness defined in the input list, with 0 being no power consumption and 1023 being the maximum power consumption. | 0, 45, 73, 143, 253, 438, 717, 1023 |
| PowerSavingRatioTableLength | An integer value (0-255) representing the number of elements in the power-saving ratio table. | 4 |
| PowerSavingRatioTableInput | Each element is an integer value (0-1023) representing the normalized total power profile for the displayed frame. FOSS evaluates the power profile for each input frame before applying any power saving. The power profile is a normalized value between 0-1023, proportional to content brightness, with 0 for pure black frame and 1023 for pure white frame. | 32, 48, 700, 788 |
| PowerSavingRatioTableOutput | Each element is an integer value (1-255) representing the target power-saving ratio for the corresponding power profile defined in the input list, with 1 being the maximum power saving and 255 being no power saving. | 255, 204, 204, 192 |

#### Panel power profile

Power saving comes from the cut down of power on panel side. Hence, panel power profile is a critical input for FOSS tuning. FOSS tuning process starts with power profiling for the target OLED panel.

Table 3-2 represents the relationship between the pixel value and panel power consumption.

To generate the panel power profile table:

1. Display a solid black image on the panel and measure its power consumption as baseline power A.

2. Display a set of gray-level images on the device (for example, values 0, 16, 32,.., 255 for 8-bit color depth displays).
3. Measure the power consumption for each image and deduct power A from the measured value.
4. Plot graph of the measured values to generate the panel power profile.
5. Slice the curve to make pixel value and power consumption pairs. In Figure 3-1, eight points have been identified and made members of the profile table input and output entries.

Figure 3-1 shows the sample panel power profile defined in Table 3-2.

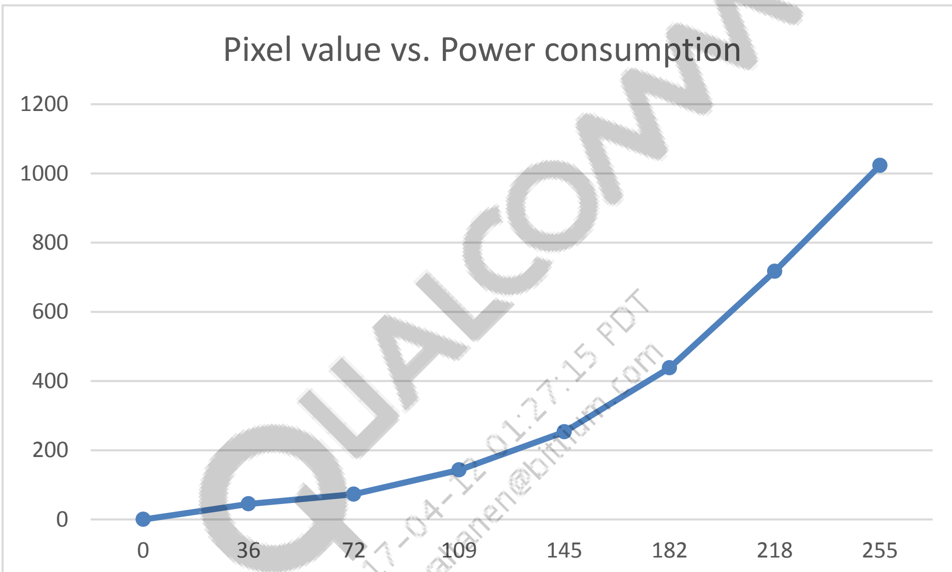




**Figure 3-1 Sample panel power profile**

**Table 3-2 Default panel power profile table**

| | |
|---|---|
| FOSSPowerProfileTableLength | 8 |
| FOSSPowerProfileTableInput | 0, 36, 72, 109, 145, 182, 218, 255 |
| FOSSPowerProfileTableOutput | 0, 45, 73, 143, 253, 438, 717, 1023 |

**NOTE:** If these parameters are not specified, the default parameters (per Table 3-2) are considered. The sample xml file in Appendix A also lists the reference parameters, which are considered as the default parameters when no parameters are specified.

## Power-saving ratio

Figure 3-2 shows the relationship between normalized power consumption for the displayed content and power-saving ratio. Normalized power consumption for displayed content ranges from 0 to 1023, proportional to brightness of the content. For example, the normalized power consumption is 1023 for pure white content and 0 for pure black content. Power-saving ratio ranges from 1 to 255, where 255 represents no saving and 1 represents the maximum saving. The power-saving percentage is calculated as:

$$Psaving = 1 - \frac{Ratio}{255}$$

For example, a ratio value of 204 represents 20% saving.

The power-saving ratio table consists of minimum 2 and maximum 256 pairs of (normalized power consumption, power-saving ratio) elements that define power-saving behavior. For example, Figure 3-2 shows the power-saving behavior defined by the power-saving ratio table listed in Table 3-3

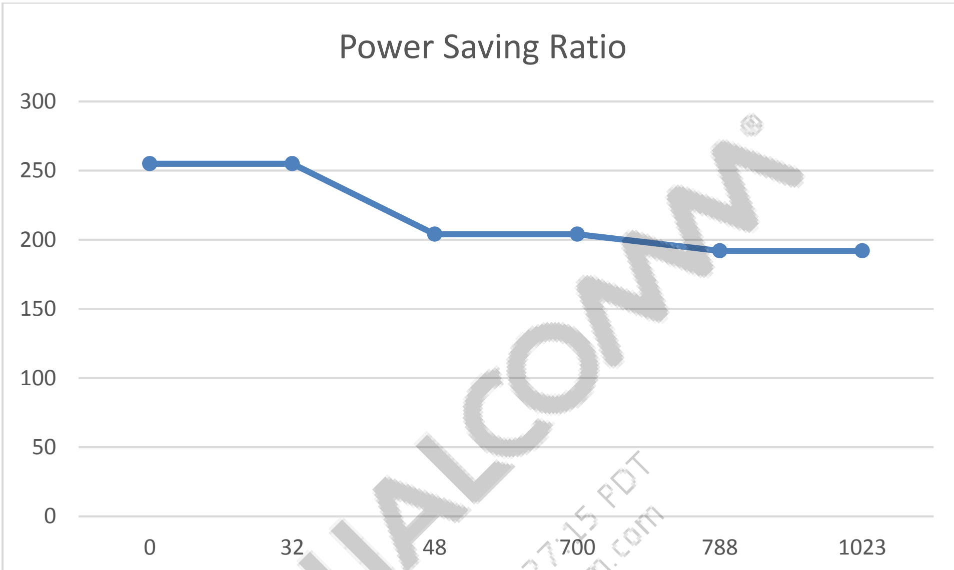


**Figure 3-2 Sample power-saving ratio**

**Table 3-3 Sample power-saving ratio table**

| | |
|---|---|
| FOSSPowerSavingRatioTableLength | 4 |
| FOSSPowerSavingRatioTableInput | 32, 48, 700, 788 |
| FOSSPowerSavingRatioTableOutput | 255, 204, 204, 192 |

In Figure 3-2, FOSS achieves no power saving for content with normalized power consumption ranging from 0 to 32, 0~20% power saving for content with normalized power consumption ranging from 32 to 48, 20% power saving for content with normalized power consumption ranging from 48 to 700, and so on.

## 3.3.2 Quality parameters - QTI recommended

The parameters described in this section are recommended by QTI to be used as **default** parameters, which govern the FOSS algorithm functional behavior. FOSS is a nonlinear tone adjustment method; there are potential quality issues in gradient content, darker content, or contouring lines. The core principal of FOSS is to exploit the panel side power with the same perceptual visual quality. As in case of any power saving algorithm, when quality parameters are configured to achieve better performance (quality), power numbers get affected and vice versa.

- Gradient detection – A gradient detection option is available in FOSS, which once enabled, the displayed content is detected to have higher possibility to be a gradient. In such cases, a linear power-saving LUT is adopted to reduce the artifacts. Default value is off.
- Contrast strength – This parameter determines the level of contrast enhancement applied by FOSS and can be tuned if contrast-related quality issues are observed. Default value is 128.

- LowBrightnessCutoff – This parameter determines the cutoff point beyond which the FOSS provides significant power saving (i.e., FOSS saving – FOSS CPU overhead)
- LuxEndPoint – Ambient light value beyond which FOSS stops working and other picture enhancement features (i.e., SVI/A) start working.
- LutUpdateSpeedOutput – This parameter determines the FOSS working speed (convergence till final frame) and can be tuned in case of flickers observed with scene change use cases. Per internal testing, very minor power impact is observed if very slow speed is used.

## 3.4 FOSS power profiling numbers

The following measurement procedure was used to measure the power numbers using an MSM8952 device:

- Brightness manually set in the UI as indicated in results
- Display Sleep → Never
- Bluetooth → Off
- Wi-Fi → Off
- Airplane mode (explicit even though Wi-Fi is Off to ensure that data is also not used)
- Only portrait (no UI rotation)
- Sounds all set to 0
- Adaptive brightness solution in Android → Turned Off
- Power measure at battery input without USB and only with battery power of 3.8 V
- Device first put to suspend post reboot and it was ensured that device went to XO shutdown. Instantaneous current reaches near zero at ~110 sec.
- Each measurement duration was 10 sec for images, and for video clips it was the entire duration of the video.
- Percentage power savings are computed as follows:
  - A = power at battery while showing black frame
  - B = power at battery while showing content with FOSS off
  - C = power at battery while showing content with FOSS on
  - FOSS savings = 100 * (B – C) / (B – A) [%]

**NOTE:** The file contents (described in Table 3-4) are internal data for QTI reference only and are not shared with customers.

### Table 3-4 Reference content

| Filename (image/video) | Description |
|---|---|
| 10.png | Portrait of woman's face against light gray background |
| 17.png | Photo of Egyptian pyramid against a deep blue sky with some clouds |
| 18.png | Daytime photo of farm land with blue sky |

| Filename (image/video) | Description |
|---|---|
| 34.png | Twilight long exposure photo of city against dark blue sky |
| blue0_1920x1080.png | Photo of castle tower on green grass with CGI blue sky with clouds and moon |
| blue1_1920x1080.png | Photo of rolling hills at edge of a calm lake with reflections of blue sky |
| DSC04415 | Aerial photo of “old world” buildings against a gray sky with clouds |
| DSC04408 | Photo of old castle against cloudy sky with several people standing on walkway |
| DSC03028 | Photo of a residential street showing homes and a red car parked in the foreground |
| Wechat0 | Screenshot of WeChat app showing texting and user icons against gray background |
| Wechat1 | Screenshot of WeChat app showing texting and user icons against blue background |
| Wechat2 | Screenshot of WeChat app showing texting and user icons against brown defocused bokeh photo background |
| Wechat3 | Screenshot of WeChat app showing texting and user icons against blue defocused bokeh photo background |
| 36.png | Portrait of Canadian actress’ face against a gray background |
| red.png | 100% red flat field |
| green.png | 100% green flat field |
| blue.png | 100% blue flat field |
| qct77_scale_100_30fps | Video, 30 Hz, showing animals in arctic ice |
| qct77_scale_80_30fps | Video, 30 Hz, showing animals in arctic ice (luminance scaled to 80%) |
| qct77_scale_50_30fps | Video, 30 Hz, showing animals in arctic ice (luminance scaled to 50%) |

Figure 3-3 shows the power profiling data and power-saving numbers with brightness level set to 100%.

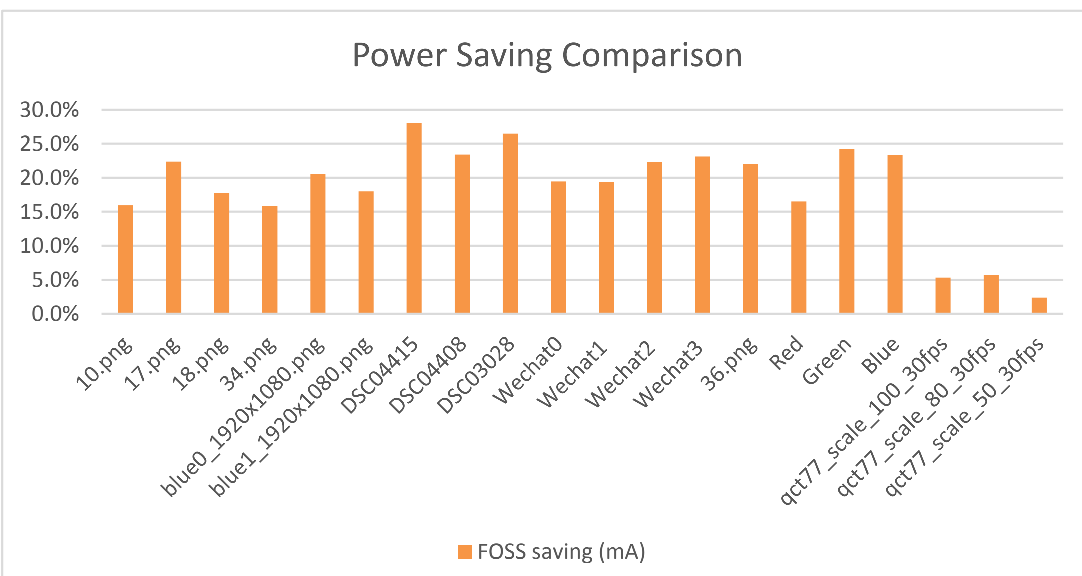


**Figure 3-3 FOSS power savings (%)**

Figure 3-4 shows the power profiling data and power-saving numbers with brightness level set to absolute numbers.

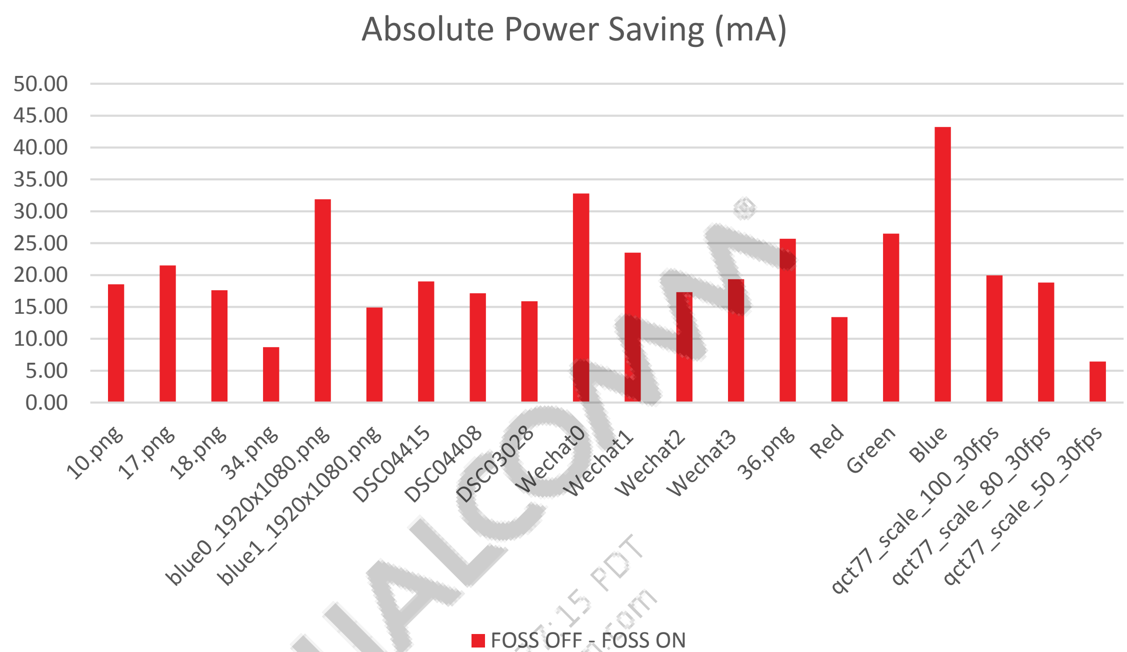


**Figure 3-4 FOSS power savings (absolute)**

# 3.5 FOSS runtime enable/disable

FOSS can be dynamically enabled or disabled. This can be done by securing a socket connection with PPS. Customers can enable their own applications to have FOSS working in particular app/UI using this method. QTI can help with a sample apk to test this; contact QTI customer support if you need a sample apk code.

### FOSS turned off with a socket command

```
adb shell system/vendor/bin/ppd cabl:off
Testing for command: cabl:off
writing cabl:off to the socket
Daemon response: Success
```

### FOSS turned on with a socket command

```
adb shell system/vendor/bin/ppd cabl:on
Testing for command: cabl:on
writing cabl:on to the socket
Daemon response: Success
```

# 4 Limitations

### Global effect

FOSS has an effect on the entire frame as it is a global tone adjustment algorithm. The algorithm cannot perform enhancement or reduce pixel intensity only for a part of the displayed content.

### Temporal effect

As FOSS is a per-histogram processing algorithm and a software feature, it impacts the temporal LUT difference and software frame delay, respectively. This implies that image brightness changes may be noticeable when the LUT convergence is not yet completed. If you observe flickers only with FOSS enabled, contact QTI support (support.cdmatech@qti.qualcomm.com).

### Contour effect

As FOSS is a nonlinear tone adjustment method, there are potential quality issues in ramp/gradient content (contouring lines). If you observe contours only with FOSS enabled, contact QTI support (support.cdmatech@qti.qualcomm.com).

### Content dependent

FOSS works to save power consumption of OLED panels. OLED panel's power consumption is directly proportional to the pixel intensity. The brighter the content, the higher the power consumption. The power-saving numbers achieved for brighter content are higher than that of the darker content.

### Panel dependent

The panel power profile plays a crucial role in determining the exact amount of power savings with FOSS. Due to this, the savings achieved vary with different panels.

# A Sample XML file

NOTE: Numerous changes were made in this chapter.

```
<Group id='FOSS Configuration'>
        <PowerSaveRatioTableLength
units='uint'>4</PowerSaveRatioTableLength>
        <PowerSaveRatioInput units='uint'>32 48 700
788</PowerSaveRatioInput>
        <PowerSaveRatioOutput units='uint'>255 204 204
192</PowerSaveRatioOutput>

        <PowerProfileTableLength units='uint'>8</PowerProfileTableLength>
        <PowerProfileInput units='uint'>0 36 72 109 145 182 218
255</PowerProfileInput>
        < PowerProfileOutput units='uint'>0 45 73 143 253 438 717
1023</PowerProfileOutput>
 </Group>
```

# B References

## B.1 Related documents

| Documents | |
|---|---|
| **Qualcomm Technologies, Inc.** | |
| *Display Postprocessing Features Technical Notes* | 80-NJ164-1 |
| *Sunlight Visibility Improvement (SVI) OEM Design Guide* | 80-NP906-2 |

## B.2 Acronyms and terms

| Term | Definition |
|---|---|
| AD | Assertive Display |
| CABL | Content Adaptive Backlight Scaling |
| FOSS | Fidelity Optimized Signal Scaling |
| HVS | Human Visual System |
| LUT | Look-up table |
| SVI | Sunlight Visibility Improvement |