Qualcomm Technologies, Inc.

Qualcomm Linux Power And Thermal Guide - Addendum

80-70018-30A AA

March 26, 2025

Contents

1 Overview of power management

This addendum serves as a supplementary guide, and it's intended for licensed users with authorized access to Qualcomm® Linux®.

Read this addendum along with the Qualcomm Linux Power and Thermal Guide. The Qualcomm Linux Power and Thermal Guide describes the architecture, supported features, and debug methods to balance power consumption and performance of the Qualcomm Linux software.

This guide describes QCS6490/QCS5430 specifications, configuration and customization, and power dashboards. The power consumption of the device depends on the CPU, and GPU configurations. The following table lists the specifications for the CPU subsystems on QCS6490/QCS5430:

Specifications	**QCS6490**		
Core type	**Kryo Prime**	**Kryo Gold**	**Kryo Silver**
Number of CPUs	1	3	4
CPU maximum frequency	2.7 GHz	2.4 GHz	1.9 GHz
L1 I cache	32 kB	32 kB/core	32 kB/core
L1 D cache	32 kB	32 kB/core	32 kB/core

2 Configure CPU and GPU

It's essential to tune the basic configuration settings of your device before starting the power analysis. These settings play a significant role in the power consumption of the device.

> **Caution:** Modifying the CPU, and GPU configurations impacts the power and the performance of the device. Ensure that you verify the impact across all relevant use cases before any modifications.

2.1 Configure CPU

You can read and update the CPU configurations using the commands specified in the following table:

> **Note:** Run the commands specified in the following table on the Linux device.

Commands to configure the CPU

Command	Purpose
`cat /sys/devices/system/cpuX/online`	Shows the number of online CPU cores. In cpuX, X represents the number of cores, which ranges from 0 to 7.
`echo 1 > /sys/devices/system/cpu/cpuX/online`	Turns on a CPU core. In cpuX, X represents the number of cores, which ranges from 0 to 7.
`echo 0 > /sys/devices/system/cpu/cpuX/online`	Turns off a CPU core. In cpuX, X represents the number of cores, which ranges from 0 to 7.

Command	Purpose
`cat /sys/devices/system/cpu/cpufreq/policy<A>/scaling_cur_freq`	Reads the current frequency of the CPU.
`cat /sys/devices/system/cpu/cpufreq/policy<A>/scaling_available_frequencies`	Reads the supported frequencies of the CPU.
`cat /sys/devices/system/cpu/cpufreq/policy<A>/scaling_min_freq`	Reads the minimum frequency of the CPU.
`echo <cpu freq in KHz> > /sys/devices/system/cpu/cpufreq/policy<A>/scaling_min_freq`	Sets the minimum frequency of the CPU. Replace `<cpu freq in KHz>` with the required frequency and run the commands.
`cat /sys/devices/system/cpu/cpufreq/policy<A>/scaling_max_freq`	Reads the maximum frequency of the CPU.
`echo <cpu freq in KHz> > /sys/devices/system/cpu/cpufreq/policy<A>/scaling_max_freq`	Sets the maximum frequency of the CPU, replace `<cpu freq in KHz>` with the required frequency and run the commands
`cat /sys/devices/system/cpu/cpufreq/policy<A>/stats/trans_table`	Shows the CPU residency.

Command	Purpose

For example, to set the CPU frequency of the Gold core at 1.5 GHz, run the following commands: .. code:
`echo 1516800 > /sys/devices/system/cpu/cpufreq/policy0/scaling_min_freq`
`echo 1516800 > /sys/devices/system/cpu/cpufreq/policy0/scaling_max_freq`
	Sets the CPU frequency at a required level, both `scaling_min_freq` and `scaling_max_freq` are set to the same frequency.

Note: `<A>` refers to 0, 4, 7

- 0 is for the Silver cluster
- 4 is for the Gold cluster
- 7 is for the Prime cluster
- Unit for frequency is KHz

2.2 Configure GPU

You can read and update the GPU settings using the commands specified in the following table:

Note: For these GPU configuration-specific commands, the unit of the output value is in Hz.

Commands to configure the GPU

Command	Purpose
`cat /sys/class/kgsl/kgsl-3d0/gpuclk`	Reads the current frequency of GPU.
`cat /sys/class/kgsl/kgsl-3d0/gpu_available_frequencies`	Reads the supported GPU frequency.
`cat /sys/class/kgsl/kgsl-3d0/devfreq/min_freq`	Reads the minimum frequency of the GPU.
`echo <GPU freq in Hz> > /sys/class/kgsl/kgsl-3d0/devfreq/min_freq`	Sets the minimum frequency of the GPU. Replace `<GPU freq in Hz>` with the intended frequency in Hz and run the command.
`cat /sys/class/kgsl/kgsl-3d0/devfreq/max_freq`	Reads the maximum frequency of the GPU.
`echo <GPU freq in Hz> /sys/class/kgsl/kgsl-3d0/devfreq/max_freq`	Sets the maximum frequency of the GPU. Replace `<GPU freq in Hz>` with the intended frequency in Hz and run the command.
`cat /sys/class/kgsl/kgsl-3d0/gpu_busy_percentage`	Reads the GPU percentage utilization.

For example, to set the GPU frequency at 600000000 Hz, run the following commands:
`echo 600000000 > /sys/class/kgsl/kgsl-3d0/devfreq/min_freq`
`echo 600000000 > /sys/class/kgsl/kgsl-3d0/devfreq/max_freq`
	Sets the GPU frequency at a required level, both `min_freq` and `max_freq` are set to the same frequency level.

Command	Purpose
`cat /sys/class/kgsl/kgsl-3d0/max_pwrlevel` `cat /sys/class/kgsl/kgsl-3d0/min_pwrlevel`	Reads the power levels of the GPU. Level 0 is mapped to the maximum GPU clock. The higher the level, the lower is the GPU clock.
`echo 0 > /sys/class/kgsl/kgsl-3d0/min_pwrlevel`	Sets the GPU frequency to FMAX.

3 Customize CPU frequency

Customization is the process of adjusting various features of the device to tune in the overall power consumption and performance of the system.

You can adjust the parameters of CPU scheduler, CPU frequency governor, and DVFS governor. It's recommended to perform any customization only after gaining a thorough understanding through extensive performance and power analysis.

Caution: Any customization can impact the power and the performance of the device. Therefore, it's crucial to verify the impact across all the relevant use cases before performing any customization.

3.1 Customize CPU frequency governor

You can configure a CPU frequency governor using the `<scaling_governor>` node to enhance CPU power and performance.

Note: The commands specified in the following table should be run on the device.

Commands to customize the CPU frequency governor

Command	Purpose
`cat /sys/devices/system/cpu/cpufreq/policyX/scaling_governor`	Verifies the CPU frequency governor.
`echo schedutil > /sys/devices/system/cpu/cpufreq/policyX/scaling_governor`	Sets the CPU frequency governor to `schedutil`.

Command	Purpose
`echo 1000 > /sys/devices/system/cpu/cpufreq/policyX/schedutil/rate_limit_us`	Customizes `rate_limit_us`, which is a `schedutil` governor parameter. It contains the value in microseconds. The governor waits for `rate_limit_us` time to re-evaluate the load after it has evaluated the load previously. The Qualcomm-tuned value is 1000.

Note: `X` refers to 0, 4, 7

- 0 is for the Silver cluster
- 4 is for the Gold cluster
- 7 is for the Prime cluster

4 Power measuremet procedures and data

Qualcomm Linux defines a set of power dashboard test cases to understand power consumption KPIs of the device. Knowing the details of the measurement platform allows you to perform test case procedures to determine and reference the measured power numbers of the QCS6490/QCS5430 platform.

4.1 Power measurement platform

QCS6490/QCS5430 is the main device for development. By default, these devices support only power measurement at 12V. However, you can measure the system on module (SoM) power at 4.2V with hardware rework.

Following are the instructions for hardware rework:

1. Remove R2406 and R605.
2. Attach TP2400 to R605 pad (VPH_PWR) which isn't connected to SoM.
3. Connect external DC Power supply of 4.2V at TP2401 (Interposer power pin).

For more information about design package and schematics, see Design Package, RB3 Gen 2 SoM and Interposer.

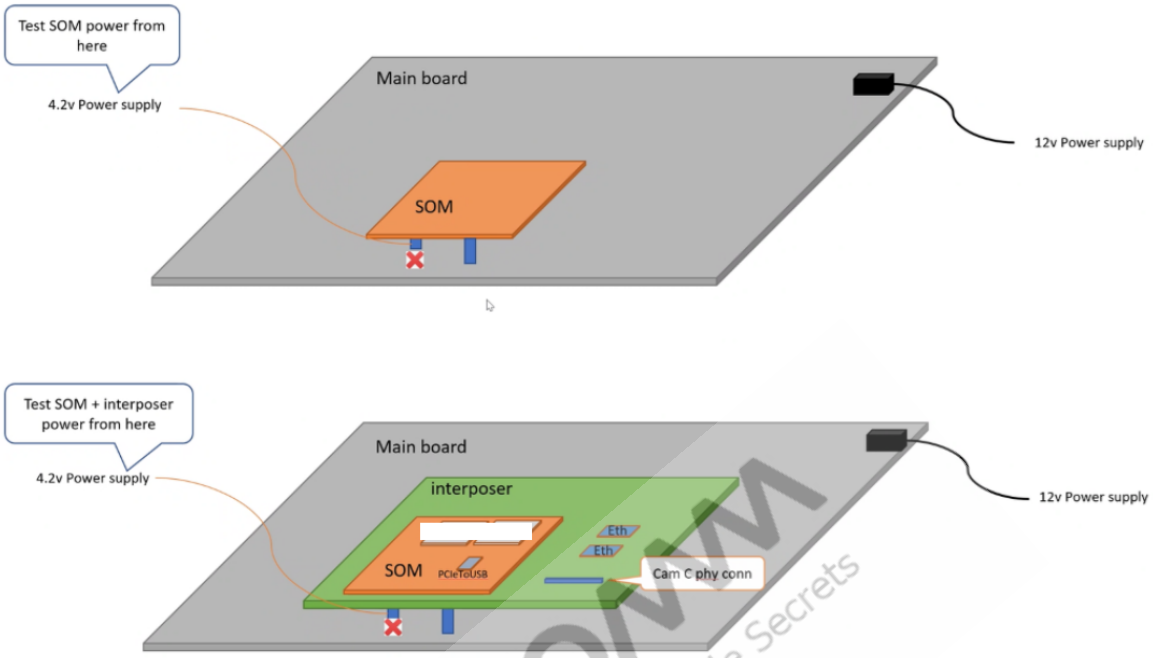

Figure : SoM power

4.2 Test case measurement procedures

This topic provides detailed measurement procedure for the test cases measured on QCS6490/QCS5430.

Test case: Sleep

Measure sleep test cases when the device is in AOSD Sleep state. The lowest power is measured is this state. Follow the instructions to configure the device in the AOSD Sleep state and measure power consumption:

1. Connect USB and run the following commands in the command prompt. The commands ensure that the device enters the AOSD Sleep state.

```
adb root
```

```
adb shell mount -o remount,rw /
```

```
adb shell mount -o remount,rw /usr
```

```
adb shell
```

```
mount -t debugfs none /sys/kernel/debug
```

```
echo device > /sys/kernel/debug/usb/a600000.
usb/mode
```

```
echo device > /sys/kernel/debug/usb/8c00000.
usb/mode
```

```
echo 3000 > /sys/bus/platform/devices/a600000.
usb/power/autosuspend_delay_ms
```

```
echo 3000 > /sys/bus/platform/devices/8c00000.
usb/power/autosuspend_delay_ms
```

```
echo auto > /sys/bus/platform/devices/a600000.
usb/power/control
```

```
echo auto > /sys/bus/platform/devices/8c00000.
usb/power/control
```

```
systemctl stop wlan_daemon
```

```
systemctl stop wlan_daemon.service
```

```
iw phy0 wowlan enable magic-packet
```

```
nohup sh -c 'sleep 10 && echo s2idle > /sys/
power/mem_sleep && echo mem > /sys/power/
autosleep' >/dev/null 2>&1 &
```

2. Disconnect USB and wait for 60 seconds to allow the device enter the AOSD Sleep state.
3. Start capturing power measurements.

Test case: Idle display

Perform the measurement of this test case with the device display on. The test case helps to understand power consumption of a device in active state and without running any applications in the background.

Follow the instructions to measure the power consumption for idle display test case:

1. Connect an external display to the device using a HDMI cable.
2. Wait until the display comes up. The white flower screen should be displayed.

To ensure proper device configuration for idle display use case, use the following commands:

```
adb root
```

```
adb shell mount -o remount,rw /
```

```
adb shell mount -o remount,rw /usr
```

```
adb shell
```

```
mount -t debugfs none /sys/kernel/debug
```

```
echo device > /sys/kernel/debug/usb/a600000.usb/mode
```

```
echo device > /sys/kernel/debug/usb/8c00000.usb/mode
```

```
echo 3000 > /sys/bus/platform/devices/a600000.usb/
power/autosuspend_delay_ms
```

```
echo 3000 > /sys/bus/platform/devices/8c00000.usb/
power/autosuspend_delay_ms
```

```
echo auto > /sys/bus/platform/devices/a600000.usb/
power/control
```

```
echo auto > /sys/bus/platform/devices/8c00000.usb/
power/control
```

```
systemctl stop wlan_daemon
```

```
systemctl stop wlan_daemon.service
```

```
iw phy0 wowlan enable magic-packet
```

```
export GBM_BACKEND=msm && export XDG_RUNTIME_DIR=/
dev/socket/weston && mkdir -p $XDG_RUNTIME_DIR &&
weston --continue-without-input --idle-time=0 &
```

3. Wait for 60 sec and then start capturing power measurements for 120 sec.

Test case: Camera preview 1080p at 30 fps preview to display

Perform the power measurements of the test cases with preview frames captured at 1080p resolution. This test case helps to understand camera power KPI during Camera Preview mode.

To measure the power consumption for camera preview test case, follow these steps:

1. Connect an external display to the device using a HDMI cable.
2. Power on the device and wait until the display comes up. The white flower screen should be displayed.
3. Connect USB and run the following commands at the command prompt. The commands ensures start of camera preview test case.

```
adb root
```

```
adb shell mount -o remount,rw /
```

```
adb shell mount -o remount,rw /usr
```

```
adb shell
```

```
mount -t debugfs none /sys/kernel/debug
```

```
echo device > /sys/kernel/debug/usb/a600000.
usb/mode
```

```
echo device > /sys/kernel/debug/usb/8c00000.
usb/mode
```

```
echo 3000 > /sys/bus/platform/devices/a600000.
usb/power/autosuspend_delay_ms
```

```
echo 3000 > /sys/bus/platform/devices/8c00000.
usb/power/autosuspend_delay_ms
```

```
echo auto > /sys/bus/platform/devices/a600000.
usb/power/control
```

```
echo auto > /sys/bus/platform/devices/8c00000.
usb/power/control
```

```
systemctl stop wlan_daemon
```

```
systemctl stop wlan_daemon.service
```

```
iw phy0 wowlan enable magic-packet
```

```
export XDG_RUNTIME_DIR=/dev/socket/weston &&
export WAYLAND_DISPLAY=wayland-1
```

```
gst-launch-1.0 -ev qtiqmmfsrc camera=0
name=camsrc video_0::type=video ! video/x-raw,
format=NV12,width=1920,height=1080,
framerate=30/1 ! waylandsink fullscreen=true
async=true sync=false
```

4. Disconnect USB.
5. Wait for 60 sec and then start capturing power measurement data for 120 sec.

Note: For all iterations wait for the device to cool down and then take fresh measurements.

Test case: Video Playback 4K @30 fps

Perform the power measurements of the test cases while playing 4K resolution video file. This test case helps to understand power KPI during video playback.

Follow the instructions to measure the power consumption for video playback test case.

1. Connect an external display to the device using a HDMI cable.
2. Wait until the display comes up. The white flower screen should be displayed.
3. Connect USB and run the following commands at the command prompt: Select your own 4K video file.

```
adb root
```

```
adb shell mount -o remount,rw /
```

```
adb shell mount -o remount,rw /usr
```

```
adb shell
```

```
mount -t debugfs none /sys/kernel/debug
```

```
echo device > /sys/kernel/debug/usb/
a600000.usb/mode
```

```
echo device > /sys/kernel/debug/usb/
8c00000.usb/mode
```

```
echo 3000 > /sys/bus/platform/devices/
a600000.usb/power/autosuspend_delay_ms
```

```
echo 3000 > /sys/bus/platform/devices/
8c00000.usb/power/autosuspend_delay_ms
```

```
echo auto > /sys/bus/platform/devices/
a600000.usb/power/control
```

```
echo auto > /sys/bus/platform/devices/
8c00000.usb/power/control
```

```
systemctl stop wlan_daemon
```

```
systemctl stop wlan_daemon.service
```

```
iw phy0 wowlan enable magic-packet
```

```
adb push <4K video file > /opt/
```

```
export XDG_RUNTIME_DIR=/dev/socket/weston
&& export WAYLAND_DISPLAY=wayland-1
```

```
gst-launch-1.0 --gst-debug=2 filesrc
location=<4k video file>.mp4 ! qtdemux !
queue ! h264parse ! v4l2h264dec capture-
io-mode=4 output-io-mode=4 ! video/x-raw,
format=NV12 ! waylandsink enable-last-
sample=false async=false fullscreen=true
```

4. Disconnect USB.
5. Wait for 60 sec and then start capturing power measurement data for 60 sec.

Test case: Camera Encode (4K Encode @30 fps, H264)

Perform the power measurements of the test cases during camera preview (video recording) with frames captured at 4K resolution. This test case helps to understand camera power KPI during Camera Recording mode.

Follow the instructions to measure the power consumption for camera recording test case.

1. Connect an external display to the device using a HDMI cable.
2. Power on the device and wait until the display comes up. The white flower screen should be displayed.
3. Connect USB and run the following commands at the command prompt. The commands ensures start of camera encode test case.

```
adb root
```

```
adb shell mount -o remount,rw /
```

```
adb shell mount -o remount,rw /usr
```

```
adb shell
```

```
mount -t debugfs none /sys/kernel/debug
```

```
echo device > /sys/kernel/debug/usb/a600000.usb/mode
```

```
echo device > /sys/kernel/debug/usb/8c00000.usb/mode
```

```
echo 3000 > /sys/bus/platform/devices/a600000.usb/
power/autosuspend_delay_ms
```

```
echo 3000 > /sys/bus/platform/devices/8c00000.usb/
power/autosuspend_delay_ms
```

```
echo auto > /sys/bus/platform/devices/a600000.usb/
power/control
```

```
echo auto > /sys/bus/platform/devices/8c00000.usb/
power/control
```

```
systemctl stop wlan_daemon
```

```
systemctl stop wlan_daemon.service
```

```
iw phy0 wowlan enable magic-packet
```

```
export XDG_RUNTIME_DIR=/dev/socket/weston && export
WAYLAND_DISPLAY=wayland-1
```

```
gst-launch-1.0 -e qtiqmmfsrc camera=0 name=camsrc !
video/x-raw,format=NV12_Q08C,width=3840,height=2160,
framerate=30/1,interlace-mode=progressive,
colorimetry=bt601 ! queue ! v4l2h264enc capture-io-
mode=4 output-io-mode=5 extra-controls="controls,
video_bitrate_mode=0,video_bitrate=6000000,h264_i_
frame_qp_value=27,h264_b_frame_qp_value=28,h264_p_
frame_qp_value=28,video_gop_size=29,h264_entropy_
mode=1,h264_profile=4;" ! h264parse ! mp4mux ! queue
! filesink location=/opt/mux_4k_avc.mp4
```

4. Disconect USB.
5. Wait for 60 sec and then start capturing power measurement data for 120 sec.

4.3 Power measurement data

The following table presents QCS6490/QCS5430 measured power data based on the software build ID QCM6490.LE.1.0-00269-STD.INT.SL-1.

Note: SoM power includes power consumption of SoC, PMIC, and DDR components. The following measurements exclude the power consumption of the display and camera sensor.

Category	**Power test cases**	**QCS6490/QCS5430 LE 1.0 6 GB LP4x, IMX577, HDMI Display SoM Power (mA@4.2V)**
Sleep	Rock bottom sleep current	7.8
Idle display	Idle display screen over HDMI	62
Video decode	Video playback (4K content @30 fps)	189
Camera preview	Camera preview (1080p resolution @30 fps)	219
Camera encode	Camera recording (4K resolution @30 fps in H.264 format)	292

5 References

Related documents

Title	Number
Qualcomm Technologies, Inc.	
Design Package, RB3 Gen 2 SoM and Interposer	DP25-20659-5

Table : Acronyms and terms

Acronyms or terms	Definition
APSS	Application processor subsystem
CPUSS	CPU subsystem
DDR	Double data rate
DVFS	Dynamic voltage and frequency scaling
SoM	System on module

LEGAL INFORMATION

Your access to and use of this material, along with any documents, software, specifications, reference board files, drawings, diagnostics and other information contained herein (collectively this "Material"), is subject to your (including the corporation or other legal entity you represent, collectively "You" or "Your") acceptance of the terms and conditions ("Terms of Use") set forth below. If You do not agree to these Terms of Use, you may not use this Material and shall immediately destroy any copy thereof.

1) Legal Notice.

This Material is being made available to You solely for Your internal use with those products and service offerings of Qualcomm Technologies, Inc. ("**Qualcomm Technologies**"), its affiliates and/or licensors described in this Material, and shall not be used for any other purposes. If this Material is marked as "**Qualcomm Internal Use Only**", no license is granted to You herein, and You must immediately (a) destroy or return this Material to Qualcomm Technologies, and (b) report Your receipt of this Material to qualcomm.support@qti.qualcomm.com. This Material may not be altered, edited, or modified in any way without Qualcomm Technologies' prior written approval, nor may it be used for any machine learning or artificial intelligence development purpose which results, whether directly or indirectly, in the creation or development of an automated device, program, tool, algorithm, process, methodology, product and/or other output. Unauthorized use or disclosure of this Material or the information contained herein is strictly prohibited, and You agree to indemnify Qualcomm Technologies, its affiliates and licensors for any damages or losses suffered by Qualcomm Technologies, its affiliates and/or licensors for any such unauthorized uses or disclosures of this Material, in whole or part.

Qualcomm Technologies, its affiliates and/or licensors retain all rights and ownership in and to this Material. No license to any trademark, patent, copyright, mask work protection right or any other intellectual property right is either granted or implied by this Material or any information disclosed herein, including, but not limited to, any license to make, use, import or sell any product, service or technology offering embodying any of the information in this Material.

THIS MATERIAL IS BEING PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESSED, IMPLIED, STATUTORY OR OTHERWISE. TO THE MAXIMUM EXTENT PERMITTED BY LAW, QUALCOMM TECHNOLOGIES, ITS AFFILIATES AND/OR LICENSORS SPECIFICALLY DISCLAIM ALL WARRANTIES OF TITLE, MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, SATISFACTORY QUALITY, COMPLETENESS OR ACCURACY, AND ALL WARRANTIES ARISING OUT OF TRADE USAGE OR OUT OF A COURSE OF DEALING OR COURSE OF PERFORMANCE. MOREOVER, NEITHER QUALCOMM TECHNOLOGIES, NOR ANY OF ITS AFFILIATES AND/OR LICENSORS, SHALL BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY EXPENSES, LOSSES, USE, OR ACTIONS HOWSOEVER INCURRED OR UNDERTAKEN BY YOU IN RELIANCE ON THIS MATERIAL.

Certain product kits, tools and other items referenced in this Material may require You to accept additional terms and conditions before accessing or using those items.

Technical data specified in this Material may be subject to U.S. and other applicable export control laws. Transmission contrary to U.S. and any other applicable law is strictly prohibited.

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

This Material is subject to change without further notification.

In the event of a conflict between these Terms of Use and the *Website Terms of Use* on www.qualcomm.com, the *Qualcomm Privacy Policy* referenced on www.qualcomm.com, or other legal statements or notices found on prior pages of the Material, these Terms of Use will control. In the event of a conflict between these Terms of Use and any other agreement (written or click-through, including, without limitation any non-disclosure agreement) executed by You and Qualcomm Technologies or a Qualcomm Technologies affiliate and/or licensor with respect to Your access to and use of this Material, the other agreement will control.

These Terms of Use shall be governed by and construed and enforced in accordance with the laws of the State of California, excluding the U.N. Convention on International Sale of Goods, without regard to conflict of laws principles. Any dispute, claim or controversy arising out of or relating to these Terms of Use, or the breach or validity hereof, shall be adjudicated only by a court of competent jurisdiction in the county of San Diego, State of California, and You hereby consent to the personal jurisdiction of such courts for that purpose.

2) Trademark and Product Attribution Statements.

Qualcomm is a trademark or registered trademark of Qualcomm Incorporated. Arm is a registered trademark of Arm Limited (or its subsidiaries) in the U.S. and/or elsewhere. The Bluetooth® word mark is a registered trademark owned by Bluetooth SIG, Inc. Other product and brand names referenced in this Material may be trademarks or registered trademarks of their respective owners.

Snapdragon and Qualcomm branded products referenced in this Material are products of Qualcomm Technologies, Inc. and/or its subsidiaries. Qualcomm patented technologies are licensed by Qualcomm Incorporated.