Qualcomm Technologies, Inc.

Qualcomm Linux Performance Guide - Addendum for QCS9075

User Guide

80-70018-10A Rev. AA

April 7, 2025

Contents

Tables

1 Performance overview

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

Read this addendum in conjunction with the Qualcomm Linux Performance Guide, which describes the supported features, configuration and customization options to fine-tune and enhance the performance of Qualcomm Linux. It also describes the tools to identify and analyze performance issues in the software and troubleshooting methods.

This guide describes device specifications, QCS9075-specific configuration and customization changes, and performance dashboards.

The performance of the software depends on the CPU, GPU, and DDR subsystems. Qualcomm Linux uses the Qualcomm® Kryo™ CPU. The following table lists the specifications for the subsystems on QCS9075:

Table 1-1 Specifications for QCS9075 CPU, GPU, and DDR subsystems

Specifications	QCS9075	
Core type	Kryo Gold	Kryo Prime
Number of CPUs	4	4
CPU maximum frequency	2.36 GHz	2.36 GHz
L1I cache	64 kB/core	64 kB/core
L1D cache	64 kB/core	64 kB/core
L2 cache	512 kB/core	512 kB/core
L3 cache	2 MB/core (shared per cluster)	
GPU	Qualcomm® Adreno™ 663 GPU	
GPU maximum frequency	800 MHz	
DDRSS	Six-channel high-speed memory–3200 MHz LPDDR5 SDRAM (6 × 16-bit)	

2 Features impacting performance

The Qualcomm® Linux® kernel includes features, such as the CPU scheduler, CPU frequency governor, dynamic voltage and frequency scaling (DVFS), and memory management. Additionally, Qualcomm has added a feature called PerfHAL to enhance the performance of Qualcomm Linux.

For an overview of each feature and related reference links, see Qualcomm Linux Performance Guide → Features impacting performance.

2.1 PerfHAL

PerfHAL is a Qualcomm proprietary service that offers added functionality by making perflock APIs accessible. It's beneficial when you need short-term performance enhancements or power savings.

Resource opcodes

Perflock uses a combination of opcodes and their corresponding values to perform specific operations on a perflock resource.

The following table lists the supported opcodes:

Table 2-1 Supported opcodes

Opcode	Purpose	Sysnode on device
0x44000000	Sets the minimum acceptable performance level for individual tasks and task groups.	`/proc/sys/kernel/sched_util_clamp_min`
0x44004000	Sets the maximum acceptable performance level for individual tasks and task groups.	`/proc/sys/kernel/sched_util_clamp_max`
0x44008100	Sets the minimum frequency of the Gold cluster.	`/sys/devices/system/cpu/cpufreq/policy0/scaling_min_freq`
0x44008000	Sets the minimum frequency of the Gold Prime cluster.	`/sys/devices/system/cpu/cpufreq/policy4/scaling_min_freq`
0x4400C100	Sets the maximum frequency of the Gold cluster.	`/sys/devices/system/cpu/cpufreq/policy0/scaling_max_freq`
0x4400C000	Sets the maximum frequency of the Gold Prime cluster.	`/sys/devices/system/cpu/cpufreq/policy4/scaling_max_freq`

The following are some examples of the resource opcodes:

- 0x44008100, 1958400: This pair of opcode and value indicates that the minimum frequency of the Gold cluster must be set to 1958400 KHz.
- 0x44008100, 1958400, 0x4400C100, 2100000: This pair of opcode and value indicates that the minimum frequency of the Gold cluster must be set to 1958400 KHz. The maximum frequency of the Gold cluster must be set to 2100000 KHz.

For more information about how to use and debug perflock, see Qualcomm Linux Performance Guide → Customize perflock.

3 Configure CPU

It's essential to tune the basic configuration settings of your device before starting the performance analysis. The settings play a significant role in the performance of the device. You can configure the CPU, GPU, and memory settings.

CAUTION Any configuration changes can impact the power and the performance of the device. Ensure that you verify the impact across all relevant use cases before any modifications.

For more information about the configuration settings, see Qualcomm Linux Performance Guide → Configure CPU, GPU, and memory.

You can check and modify the CPU configurations using the commands specified in the following table:

NOTE The commands specified in the following table should be run on the Linux device. The unit of the output value for these commands is in KHz.

Table 3-1 Commands to configure the CPU

Command	Purpose
`cat /sys/devices/system/cpu/online`	Checks the online CPU cores.
`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.
`cat /sys/devices/system/cpu/cpufreq/policy0/scaling_cur_freq` `cat /sys/devices/system/cpu/cpufreq/policy4/scaling_cur_freq`	Reads the current frequency of the CPU.
`cat /sys/devices/system/cpu/cpufreq/policy0/scaling_available_frequencies` `cat /sys/devices/system/cpu/cpufreq/policy4/scaling_available_frequencies`	Reads the supported frequencies of the CPU.

Table 3-1 Commands to configure the CPU (cont.)

Command	Purpose
`cat /sys/devices/system/cpu/cpufreq/policy0/scaling_min_freq` `cat /sys/devices/system/cpu/cpufreq/policy4/scaling_min_freq`	Reads the minimum frequency of the CPU.
`echo <cpu freq in KHz> > /sys/devices/system/cpu/cpufreq/policy0/scaling_min_freq` `echo <cpu freq in KHz> > /sys/devices/system/cpu/cpufreq/policy4/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/policy0/scaling_max_freq` `cat /sys/devices/system/cpu/cpufreq/policy4/scaling_max_freq`	Reads the maximum frequency of the CPU.
`echo <cpu freq in KHz> > /sys/devices/system/cpu/cpufreq/policy0/scaling_max_freq` `echo <cpu freq in KHz> > /sys/devices/system/cpu/cpufreq/policy4/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/policy0/stats/trans_table` `cat /sys/devices/system/cpu/cpufreq/policy4/stats/trans_table`	Checks the CPU residency.

For example, to set the CPU frequency of the Gold core at 1.5 GHz, run the following commands:
`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. Set `scaling_min_freq` and `scaling_max_freq` to the same frequency to keep the CPU frequency at the required level.

4 Customize CPU frequency governor

Customization is a process that includes fine-tuning various aspects of the system, which can significantly affect the overall performance and power of the system.

The CPU scheduler, CPU frequency governor, dynamic voltage and frequency scaling (DVFS) governor, perflock, and memory can be fine-tuned. Undertake any tuning 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.

For more information about the customization settings, see Qualcomm Linux Performance Guide → Customize for performance tuning.

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

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

Table 4-1 Commands to customize the CPU frequency governor

Command	Purpose
`echo performance > /sys/devices/system/cpu/cpufreq/policy0/scaling_governor` `echo performance > /sys/devices/system/cpu/cpufreq/policy4/scaling_governor`	Sets the CPU governor to enhance the system performance.
`cat /sys/devices/system/cpu/cpufreq/policy0/scaling_governor` `cat /sys/devices/system/cpu/cpufreq/policy4/scaling_governor`	Verifies the CPU frequency governor.

Table 4-1 Commands to customize the CPU frequency governor (cont.)

Command	Purpose
`echo schedutil > /sys/devices/system/cpu/cpufreq/policy0/scaling_governor` `echo schedutil > /sys/devices/system/cpu/cpufreq/policy4/scaling_governor`	Sets the CPU frequency governor to `schedutil`.
`echo 1000 > /sys/devices/system/cpu/cpufreq/policyX/schedutil/rate_limit_us`	Customizes `rate_limit_us`. The value of `X` in `policyX` corresponds to clusters 0 and 4. This 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.

5 Performance dashboards

You can access performance dashboards for boot time, system benchmarks, memory map, and product segment key performance indicators (KPIs) on the Qualcomm® Linux® reference devices.

5.1 Boot time

The boot time is the duration from device power-on until the device is up (with network disabled).

The following table lists the measured boot time (values in seconds) on QCS9075:

Use case	Score
Boot time (log-based)	15.66

NOTE A lower boot time score is better.

For information about the measurement procedure, see Boot time measurement.

5.2 System benchmarks

Geekbench is a utility for measuring CPU performance. It provides the following CPU benchmark scores on QCS9075:

Benchmark	Version	Benchmark score
Geekbench ST	6.1.0	1129
Geekbench MT	6.1.0	5781

For information about the measurement procedure, see System benchmark measurement.

NOTE A higher system benchmark score is better.

5.3 Memory map

The following table lists the memory consumption (values in MB) for each partition, such as non-Linux, kernel static, and applications. It also lists the total free memory available to the system after device boot and during the use case such as 1080p resolution encoding at 30 fps.

Memory partitions	After boot memory	1080p30FPS_videodecod_16Streams	1080p30FPS_videodecod_16Streams_AI	1280*720pCamera_encode
Total RAM	36864	36864	36864	36864
Non-Linux	808	808	808	808
Kernel static	775	775	775	775
Applications + framework	626	2517	3812	34555
Total free memory	34655	32764	31469	31826

For information about the measurement procedure, see Memory map measurement.

NOTE A higher total free memory value is better for the system performance.

5.4 Use case KPIs

The following table describes the camera latency measurement data for a QCS9075 camera use case:

Use case	Latency definition	Latency (in seconds)
720p30 encoding first record latency	First time after boot, time taken from GStreamer Connect to the first video-encoded frame	1.532

NOTE A lower value is better.

For measurement procedure, see Camera recording/snapshot latency measurement.

5.5 Measurement procedures

The measurement procedures include KPIs, such as boot time, system benchmark, and record latency.

5.5.1 Boot time measurement

To measure boot time, do the following:

1. Flash a build. See Qualcomm Linux Build Guide → Flash software images to boot the device.
2. After the device is up, reboot it.
3. Install the itsy package file (IPK) to enable systemd-analyze. For more information about the systemd-analyze tool, see Analysis tools → Systemd-analyze.

4. Collect serial logs during the device power-on process, focusing on the boot loader time.

   To collect the serial logs on a Linux host, do the following:

   a. Connect a serial cable between the device and the Linux host PC.

   b. Connect to the UART terminal to obtain serial logs. To set up the UART terminal, see Qualcomm Linux Build Guide → Connect to a UART shell.

   c. Power-off the device.

   d. Power-on the device.

   e. Save the serial logs from the terminal.

5. Collect the systemd-analyze logs by running the following commands:

```
cd /usr/bin
systemd-analyze plot > /var/lib/systemd-plot.svg
systemd-analyze dump > /var/lib/systemd-analyze_dump.txt
systemctl > /var/lib/systemctl.txt
systemd-analyze blame > /var/lib/systemd-analyze_blame.txt
systemd-analyze critical-chain > /var/lib/systemd-
analyze_critical_chain.txt
```

6. Pull the traces by running the following commands using a secure copy protocol (SCP) or a similar tool. Ensure to specify the target IP address in the commands:

```
scp -r root@10.92.162.185:/var/lib/systemd-plot.svg /local/mnt/workspace/
logs
scp -r root@10.92.162.185:/var/lib/systemd-analyze_dump.txt /local/mnt/
workspace/logs
scp -r root@10.92.162.185:/var/lib/systemctl.txt /local/mnt/workspace/logs
scp -r root@10.92.162.185:/var/lib/systemd-analyze_blame.txt /local/mnt/
workspace/logs
scp -r root@10.92.162.185: /var/lib/systemd-analyze_critical_chain.txt /
local/mnt/workspace/logs
```

7. Verify the following files:

   - `systemd-plot.svg` trace file for the Linux (kernel and user space) phases
   - `systemd-analyze_blame.txt` trace file for the individual services duration
   - `systemctl.txt` trace file for the failed services

   The following is a sample output for the non-Linux boot, which includes PBL+XBL, Core UEFI, and Kernel Loader:

```
PBL+XBL: 1405 ms
Core UEFI : 619 ms
Kernel Loader: 293 ms
Total time before non-Linux kernel: 2318 ms
```

   The following is a sample output for the Linux boot for kernel and user space:

```
QCOM Reference Distro with Wayland 1.0 qcs9100-ride-sx (Linux 6.6.52 #1
SMP PREEMPT Sun Dec 1 08:17:08 UTC 2024) arm64 vm-other
Startup finished in 176ms (loader) + 1.468s (kernel) + 22.344s
(userspace) = 23.812s multi-user.target reached after 22.272s in
userspace
```

Boot markers for the boot loader for non-Linux are as follows:

```
"PBL+XBL":  "UEFI Start",
"Core UEFI": "UEFI Total",
"Kernel Loader": "UEFI End - OS Loader"
```

5.5.2 System benchmark measurement

Geekbench is a tool used to measure system performance against established benchmarks.

To measure system performance using Geekbench, do the following:

1. Download Geekbench for the Linux/ARM architecture from upstream https://www.geekbench.com/preview/.

   NOTE The Geekbench 6 for Linux/AArch64 is a preview build. The preview builds require an active Internet connection and automatically upload benchmark results to the Geekbench browser.

2. To measure a CPU benchmark using Geekbench, do the following:
   a. Unzip the Geekbench file and push it from the host into the device, use SCP or a similar tool. Ensure to specify the target IP address in the first command. The following are the example commands:

   ```
   scp -r Geekbench-6.3.0-LinuxARMPreview root@10.92.174.66:/var/cache/
   cd /var/cache
   chmod 777 Geekbench-6.3.0-LinuxARMPreview/*
   ```

   b. To run Geekbench, run the following commands on the device:

   ```
   cd Geekbench-6.3.0-LinuxARMPreview
   ./geekbench_aarch64
   ```

5.5.3 Memory map measurement

A memory map provides information on how memory is allocated for different processes. A memory map measurement allows you to monitor a mapped process and troubleshoot any memory issues.

To calculate the memory map, boot the device and stabilize it. Then, run the following commands to collect logs from the device:

```
cat /proc/meminfo
cat /proc/iomem
cat /proc/vmstat
```

Non-Linux memory

Calculate the non-Linux memory using the following formula:

Non-Linux = Total RAM size − Total Linux

To calculate the total RAM size, run the following command:

```
cat /proc/meminfo | grep -i "MemTotal"
```

MemTotal is 5512456 kB, which corresponds to approximately 6 GB of RAM.

To calculate the total Linux memory from `iomem`, run the following command:

```
cat /proc/iomem | grep System
```

The following is an output of the command:

```
83600000-839fffff : System RAM
9c700000-9d08dfff : System RAM
9d096000-9d0a0fff : System RAM
9d0a9000-9d4ccfff : System RAM
9d4dc000-9d58efff : System RAM
9d598000-9e813fff : System RAM
9e833000-9e87dfff : System RAM
9e887000-9e890fff : System RAM
9e899000-9ed52fff : System RAM
9edcb000-9f7fbfff : System RAM
9f800000-9f9fffff : System RAM
9fc00000-a00cffff : System RAM
e3400000-1ffffffff: System RAM
```

The total Linux memory is the sum of the differences in the system RAM addresses.

For example:

839fffff − 83600000 = 4194303 bytes = 3.99 MB

Kernel static

Calculate the kernel static using the following formula:

Kernel static = Total Linux − MemTotal

MemTotal is available in meminfo as follows:

```
MemTotal: 4513944 kB
```

Application + framework memory calculation

Calculate the memory used by the applications and framework using the following formula:

Application + framework = MemTotal − Free memory

Free memory calculation

Calculate the free memory using the following formula:

Free memory = MemFree + (Cached − shmem) + buffer + ION cache

To obtain the free memory details, run the following command:

```
cat /proc/meminfo
```

The following is an output of the command:

```
MemTotal:     4513944 kB
MemFree:      3471436 kB
MemAvailable: 3816716 kB
Buffers:      9216 kB
Cached:       574856 kB
Shmem:        24776 kB
```

To check the vmstat logs for ION cache, run the following command:

```
cat /proc/vmstat
nr_kernel_misc_reclaimable 16217
```

Here, 16217 pages represent approximately 63.3 MB. The calculation is as follows:

16217 pages × 4 kB/page = 64,868 kB (since 1 kB = 1024 bytes)

To convert to megabytes (MB), calculate the ION cache as follows:

64,868 kB ÷ 1024 = 63.3 MB.

Use case to measure memory map with artificial intelligence

To measure a memory map using an artificial intelligence (AI) use case, follow these steps:

1. To connect to the Weston display, run the following command on the device:

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

2. Push the MP4 video files into the device. The following example shows the location of the MP4 file: `filesrc location=/opt/Draw_1080p_180s_30FPS.mp4`
3. Push the AI model into the device and change the model path in the commands. For example,

   ```
   model=/opt/yolov8_det_quantized.tflite
   ```

4. To run the AI use case, run the following commands:

   NOTE Ensure to replace the MP4 file name, file path, and AI model name with the required values in `filesrc location=/opt/Draw_1080p_180s_30FPS.mp4` and `model=/opt/yolov8_det_quantized.tflite` before running these commands.

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

   ```
   export ADSP_LIBRARY_PATH="/usr/lib/dsp/cdsp;/usr/lib/dsp/
   cdsp1;$ADSP_LIBRARY_PATH"
   ```

   ```
   ulimit -n 4096 && gst-launch-1.0 -ev --gst-debug=2 qtivcomposer name=mix \
   sink_0::position="<0, 0>" sink_0::dimensions="<480, 270>" \
   sink_1::position="<480, 0>" sink_1::dimensions="<480, 270>" \
   sink_2::position="<960, 0>" sink_2::dimensions="<480, 270>" \
   sink_3::position="<1440, 0>" sink_3::dimensions="<480, 270>" \
   sink_4::position="<0, 270>" sink_4::dimensions="<480, 270>" \
   sink_5::position="<480, 270>" sink_5::dimensions="<480, 270>" \
   sink_6::position="<960, 270>" sink_6::dimensions="<480, 270>" \
   sink_7::position="<1440, 270>" sink_7::dimensions="<480, 270>" \
   ```

```
sink_8::position="<0, 540>" sink_8::dimensions="<480, 270>" \
sink_9::position="<480, 540>" sink_9::dimensions="<480, 270>" \
sink_10::position="<960, 540>" sink_10::dimensions="<480, 270>" \
sink_11::position="<1440, 540>" sink_11::dimensions="<480, 270>" \
sink_12::position="<0, 810>" sink_12::dimensions="<480, 270>" \
sink_13::position="<480, 810>" sink_13::dimensions="<480, 270>" \
sink_14::position="<960, 810>" sink_14::dimensions="<480, 270>" \
sink_15::position="<1440, 810>" sink_15::dimensions="<480, 270>" \
sink_16::position="<0, 0>" sink_16::dimensions="<480, 270>" \
sink_17::position="<480, 0>" sink_17::dimensions="<480, 270>" \
sink_18::position="<960, 0>" sink_18::dimensions="<480, 270>" \
sink_19::position="<1440, 0>" sink_19::dimensions="<480, 270>" \
sink_20::position="<0, 270>" sink_20::dimensions="<480, 270>" \
sink_21::position="<480, 270>" sink_21::dimensions="<480, 270>" \
sink_22::position="<960, 270>" sink_22::dimensions="<480, 270>" \
sink_23::position="<1440, 270>" sink_23::dimensions="<480, 270>" \
sink_24::position="<0, 540>" sink_24::dimensions="<480, 270>" \
sink_25::position="<480, 540>" sink_25::dimensions="<480, 270>" \
sink_26::position="<960, 540>" sink_26::dimensions="<480, 270>" \
sink_27::position="<1440, 540>" sink_27::dimensions="<480, 270>" \
sink_28::position="<0, 810>" sink_28::dimensions="<480, 270>" \
sink_29::position="<480, 810>" sink_29::dimensions="<480, 270>" \
sink_30::position="<960, 810>" sink_30::dimensions="<480, 270>" \
sink_31::position="<1440, 810>" sink_31::dimensions="<480, 270>" \
mix. ! queue ! fpsdisplaysink signal-fps-measurements=true text-
overlay=false sync=true video-sink="waylandsink fullscreen=true sync=true"
\
filesrc location=/opt/Draw_1080p_180s_30FPS.mp4 ! qtdemux ! queue !
h264parse ! v4l2h264dec capture-io-mode=5 output-io-mode=5 ! queue ! tee
name=split0 ! queue ! mix. \
filesrc location=/opt/Draw_1080p_180s_30FPS.mp4 ! qtdemux ! queue !
h264parse ! v4l2h264dec capture-io-mode=5 output-io-mode=5 ! queue ! tee
name=split1 ! queue ! mix. \
filesrc location=/opt/Draw_1080p_180s_30FPS.mp4 ! qtdemux ! queue !
h264parse ! v4l2h264dec capture-io-mode=5 output-io-mode=5 ! queue ! tee
name=split2 ! queue ! mix. \
filesrc location=/opt/Draw_1080p_180s_30FPS.mp4 ! qtdemux ! queue !
h264parse ! v4l2h264dec capture-io-mode=5 output-io-mode=5 ! queue ! tee
name=split3 ! queue ! mix. \
filesrc location=/opt/Draw_1080p_180s_30FPS.mp4 ! qtdemux ! queue !
h264parse ! v4l2h264dec capture-io-mode=5 output-io-mode=5 ! queue ! tee
name=split4 ! queue ! mix. \
filesrc location=/opt/Draw_1080p_180s_30FPS.mp4 ! qtdemux ! queue !
h264parse ! v4l2h264dec capture-io-mode=5 output-io-mode=5 ! queue ! tee
name=split5 ! queue ! mix. \
filesrc location=/opt/Draw_1080p_180s_30FPS.mp4 ! qtdemux ! queue !
h264parse ! v4l2h264dec capture-io-mode=5 output-io-mode=5 ! queue ! tee
name=split6 ! queue ! mix. \
```

```
filesrc location=/opt/Draw_1080p_180s_30FPS.mp4 ! qtdemux ! queue !
h264parse ! v4l2h264dec capture-io-mode=5 output-io-mode=5 ! queue ! tee
name=split7 ! queue ! mix. \
filesrc location=/opt/Draw_1080p_180s_30FPS.mp4 ! qtdemux ! queue !
h264parse ! v4l2h264dec capture-io-mode=5 output-io-mode=5 ! queue ! tee
name=split8 ! queue ! mix. \
filesrc location=/opt/Draw_1080p_180s_30FPS.mp4 ! qtdemux ! queue !
h264parse ! v4l2h264dec capture-io-mode=5 output-io-mode=5 ! queue ! tee
name=split9 ! queue ! mix. \
filesrc location=/opt/Draw_1080p_180s_30FPS.mp4 ! qtdemux ! queue !
h264parse ! v4l2h264dec capture-io-mode=5 output-io-mode=5 ! queue ! tee
name=split10 ! queue ! mix. \
filesrc location=/opt/Draw_1080p_180s_30FPS.mp4 ! qtdemux ! queue !
h264parse ! v4l2h264dec capture-io-mode=5 output-io-mode=5 ! queue ! tee
name=split11 ! queue ! mix. \
filesrc location=/opt/Draw_1080p_180s_30FPS.mp4 ! qtdemux ! queue !
h264parse ! v4l2h264dec capture-io-mode=5 output-io-mode=5 ! queue ! tee
name=split12 ! queue ! mix. \
filesrc location=/opt/Draw_1080p_180s_30FPS.mp4 ! qtdemux ! queue !
h264parse ! v4l2h264dec capture-io-mode=5 output-io-mode=5 ! queue ! tee
name=split13 ! queue ! mix. \
filesrc location=/opt/Draw_1080p_180s_30FPS.mp4 ! qtdemux ! queue !
h264parse ! v4l2h264dec capture-io-mode=5 output-io-mode=5 ! queue ! tee
name=split14 ! queue ! mix. \
filesrc location=/opt/Draw_1080p_180s_30FPS.mp4 ! qtdemux ! queue !
h264parse ! v4l2h264dec capture-io-mode=5 output-io-mode=5 ! queue ! tee
name=split15 ! queue ! mix. \
split0. ! queue ! qtimlvconverter ! queue ! qtimltflite delegate=external
external-delegate-path=libQnnTFLiteDelegate.so external-delegate-
options="QNNExternalDelegate,backend_type=htp,htp_device_id=(string)0,htp_p
erformance_mode=(string)2;" model=/opt/yolov8_det_quantized.tflite !
queue ! qtimlvdetection threshold=75.0 results=10 module=yolov8
labels=/opt/yolov5m.labels constants="YOLOv8,q-offsets=<-107.0, -128.0,
0.0>,q-scales=<3.093529462814331, 0.00390625, 1.0>;" ! queue ! mix. \
split1. ! queue ! qtimlvconverter ! queue ! qtimltflite delegate=external
external-delegate-path=libQnnTFLiteDelegate.so external-delegate-
options="QNNExternalDelegate,backend_type=htp,htp_device_id=(string)0,htp_p
erformance_mode=(string)2;" model=/opt/yolov8_det_quantized.tflite !
queue ! qtimlvdetection threshold=75.0 results=10 module=yolov8
labels=/opt/yolov5m.labels constants="YOLOv8,q-offsets=<-107.0, -128.0,
0.0>,q-scales=<3.093529462814331, 0.00390625, 1.0>;" ! queue ! mix. \
split2. ! queue ! qtimlvconverter ! queue ! qtimltflite delegate=external
external-delegate-path=libQnnTFLiteDelegate.so external-delegate-
options="QNNExternalDelegate,backend_type=htp,htp_device_id=(string)0,htp_p
erformance_mode=(string)2;" model=/opt/yolov8_det_quantized.tflite !
queue ! qtimlvdetection threshold=75.0 results=10 module=yolov8
labels=/opt/yolov5m.labels constants="YOLOv8,q-offsets=<-107.0, -128.0,
0.0>,q-scales=<3.093529462814331, 0.00390625, 1.0>;" ! queue ! mix. \
```

```
split3. ! queue ! qtimlvconverter ! queue ! qtimltflite delegate=external
external-delegate-path=libQnnTFLiteDelegate.so external-delegate-
options="QNNExternalDelegate,backend_type=htp,htp_device_id=(string)0,htp_p
erformance_mode=(string)2;" model=/opt/yolov8_det_quantized.tflite !
queue ! qtimlvdetection threshold=75.0 results=10 module=yolov8
labels=/opt/yolov5m.labels constants="YOLOv8,q-offsets=<-107.0, -128.0,
0.0>,q-scales=<3.093529462814331, 0.00390625, 1.0>;" ! queue ! mix. \
split4. ! queue ! qtimlvconverter ! queue ! qtimltflite delegate=external
external-delegate-path=libQnnTFLiteDelegate.so external-delegate-
options="QNNExternalDelegate,backend_type=htp,htp_device_id=(string)0,htp_p
erformance_mode=(string)2;" model=/opt/yolov8_det_quantized.tflite !
queue ! qtimlvdetection threshold=75.0 results=10 module=yolov8
labels=/opt/yolov5m.labels constants="YOLOv8,q-offsets=<-107.0, -128.0,
0.0>,q-scales=<3.093529462814331, 0.00390625, 1.0>;" ! queue ! mix. \
split5. ! queue ! qtimlvconverter ! queue ! qtimltflite delegate=external
external-delegate-path=libQnnTFLiteDelegate.so external-delegate-
options="QNNExternalDelegate,backend_type=htp,htp_device_id=(string)0,htp_p
erformance_mode=(string)2;" model=/opt/yolov8_det_quantized.tflite !
queue ! qtimlvdetection threshold=75.0 results=10 module=yolov8
labels=/opt/yolov5m.labels constants="YOLOv8,q-offsets=<-107.0, -128.0,
0.0>,q-scales=<3.093529462814331, 0.00390625, 1.0>;" ! queue ! mix. \
split6. ! queue ! qtimlvconverter ! queue ! qtimltflite delegate=external
external-delegate-path=libQnnTFLiteDelegate.so external-delegate-
options="QNNExternalDelegate,backend_type=htp,htp_device_id=(string)0,htp_p
erformance_mode=(string)2;" model=/opt/yolov8_det_quantized.tflite !
queue ! qtimlvdetection threshold=75.0 results=10 module=yolov8
labels=/opt/yolov5m.labels constants="YOLOv8,q-offsets=<-107.0, -128.0,
0.0>,q-scales=<3.093529462814331, 0.00390625, 1.0>;" ! queue ! mix. \
split7. ! queue ! qtimlvconverter ! queue ! qtimltflite delegate=external
external-delegate-path=libQnnTFLiteDelegate.so external-delegate-
options="QNNExternalDelegate,backend_type=htp,htp_device_id=(string)0,htp_p
erformance_mode=(string)2;" model=/opt/yolov8_det_quantized.tflite !
queue ! qtimlvdetection threshold=75.0 results=10 module=yolov8
labels=/opt/yolov5m.labels constants="YOLOv8,q-offsets=<-107.0, -128.0,
0.0>,q-scales=<3.093529462814331, 0.00390625, 1.0>;" ! queue ! mix. \
split8. ! queue ! qtimlvconverter ! queue ! qtimltflite delegate=external
external-delegate-path=libQnnTFLiteDelegate.so external-delegate-
options="QNNExternalDelegate,backend_type=htp,htp_device_id=(string)1,htp_p
erformance_mode=(string)2;" model=/opt/yolov8_det_quantized.tflite !
queue ! qtimlvdetection threshold=75.0 results=10 module=yolov8
labels=/opt/yolov5m.labels constants="YOLOv8,q-offsets=<-107.0, -128.0,
0.0>,q-scales=<3.093529462814331, 0.00390625, 1.0>;" ! queue ! mix. \
split9. ! queue ! qtimlvconverter ! queue ! qtimltflite delegate=external
external-delegate-path=libQnnTFLiteDelegate.so external-delegate-
options="QNNExternalDelegate,backend_type=htp,htp_device_id=(string)1,htp_p
erformance_mode=(string)2;" model=/opt/yolov8_det_quantized.tflite !
queue ! qtimlvdetection threshold=75.0 results=10 module=yolov8
labels=/opt/yolov5m.labels constants="YOLOv8,q-offsets=<-107.0, -128.0,
```

```
0.0>,q-scales=<3.093529462814331, 0.00390625, 1.0>;" ! queue ! mix. \
split10. ! queue ! qtimlvconverter ! queue ! qtimltflite delegate=external
external-delegate-path=libQnnTFLiteDelegate.so external-delegate-
options="QNNExternalDelegate,backend_type=htp,htp_device_id=(string)1,htp_p
erformance_mode=(string)2;" model=/opt/yolov8_det_quantized.tflite !
queue ! qtimlvdetection threshold=75.0 results=10 module=yolov8
labels=/opt/yolov5m.labels constants="YOLOv8,q-offsets=<-107.0, -128.0,
0.0>,q-scales=<3.093529462814331, 0.00390625, 1.0>;" ! queue ! mix. \
split11. ! queue ! qtimlvconverter ! queue ! qtimltflite delegate=external
external-delegate-path=libQnnTFLiteDelegate.so external-delegate-
options="QNNExternalDelegate,backend_type=htp,htp_device_id=(string)1,htp_p
erformance_mode=(string)2;" model=/opt/yolov8_det_quantized.tflite !
queue ! qtimlvdetection threshold=75.0 results=10 module=yolov8
labels=/opt/yolov5m.labels constants="YOLOv8,q-offsets=<-107.0, -128.0,
0.0>,q-scales=<3.093529462814331, 0.00390625, 1.0>;" ! queue ! mix. \
split12. ! queue ! qtimlvconverter ! queue ! qtimltflite delegate=external
external-delegate-path=libQnnTFLiteDelegate.so external-delegate-
options="QNNExternalDelegate,backend_type=htp,htp_device_id=(string)1,htp_p
erformance_mode=(string)2;" model=/opt/yolov8_det_quantized.tflite !
queue ! qtimlvdetection threshold=75.0 results=10 module=yolov8
labels=/opt/yolov5m.labels constants="YOLOv8,q-offsets=<-107.0, -128.0,
0.0>,q-scales=<3.093529462814331, 0.00390625, 1.0>;" ! queue ! mix. \
split13. ! queue ! qtimlvconverter ! queue ! qtimltflite delegate=external
external-delegate-path=libQnnTFLiteDelegate.so external-delegate-
options="QNNExternalDelegate,backend_type=htp,htp_device_id=(string)1,htp_p
erformance_mode=(string)2;" model=/opt/yolov8_det_quantized.tflite !
queue ! qtimlvdetection threshold=75.0 results=10 module=yolov8
labels=/opt/yolov5m.labels constants="YOLOv8,q-offsets=<-107.0, -128.0,
0.0>,q-scales=<3.093529462814331, 0.00390625, 1.0>;" ! queue ! mix. \
split14. ! queue ! qtimlvconverter ! queue ! qtimltflite delegate=external
external-delegate-path=libQnnTFLiteDelegate.so external-delegate-
options="QNNExternalDelegate,backend_type=htp,htp_device_id=(string)1,htp_p
erformance_mode=(string)2;" model=/opt/yolov8_det_quantized.tflite !
queue ! qtimlvdetection threshold=75.0 results=10 module=yolov8
labels=/opt/yolov5m.labels constants="YOLOv8,q-offsets=<-107.0, -128.0,
0.0>,q-scales=<3.093529462814331, 0.00390625, 1.0>;" ! queue ! mix. \
split15. ! queue ! qtimlvconverter ! queue ! qtimltflite delegate=external
external-delegate-path=libQnnTFLiteDelegate.so external-delegate-
options="QNNExternalDelegate,backend_type=htp,htp_device_id=(string)1,htp_p
erformance_mode=(string)2;" model=/opt/yolov8_det_quantized.tflite !
queue ! qtimlvdetection threshold=75.0 results=10 module=yolov8
labels=/opt/yolov5m.labels constants="YOLOv8,q-offsets=<-107.0, -128.0,
0.0>,q-scales=<3.093529462814331, 0.00390625, 1.0>;" ! queue ! mix.
```

5. Follow the Memory map measurement procedure to collect a memory map during the AI use case.

6 References

6.1 Related documents

Title	Number
Qualcomm Technologies, Inc.	
Qualcomm Linux Performance Guide	80-70018-10
Qualcomm Linux Build Guide	80-70018-254
Resources	
https://www.geekbench.com/preview/	
https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt	

6.2 Acronyms and terms

Acronym or term	Definition
AI	Artificial intelligence
DVFS	Dynamic voltage and frequency scaling
IPK	Itsy package file
KPI	Key performance indicator
SCP	Secure copy protocol
SoC	System-on-chip

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.