Qualcomm Technologies, Inc.

Qualcomm Linux Storage Guide

80-70018-6 AA

March 26, 2025

Contents

1 Storage overview **3**
1.1 Storage use cases . . . 3

2 Storage features **5**
2.1 UFS device features . . . 5
2.2 SD card features . . . 5
2.3 NVMe features . . . 5

3 Configure storage features **7**
3.1 Enable UFS device feature using kernel configuration . . . 7
3.2 Enable SD card using kernel configuration . . . 7
3.3 Modify UFS device power management states . . . 8
3.4 Modify SD card power management states . . . 11
3.5 Configure device using UFS parameters . . . 11
3.6 Configure device using SD card parameters . . . 13
3.7 Configure UFS Linux kernel device tree . . . 13
3.8 Configure SD card Linux kernel device tree . . . 14
3.9 Provision UFS device . . . 15

4 Customize storage features **20**
4.1 Mount the file system . . . 20

5 Debug common storage driver issues **25**
5.1 UFS device issue . . . 25
5.2 Debug with sysfs or debugfs nodes . . . 28

6 References **31**
6.1 Related documents . . . 31
6.2 Acronyms and terms . . . 31

1 Storage overview

The Qualcomm® Linux® storage guide describes the features and supported device types, enabling you to configure, customize, and deploy storage capabilities optimally and efficiently.

The following table lists the supported storage types.

Storage type	Version	Supported SoC
Universal flash storage (UFS)	• v2.1 • v2.2 • v3.1	• All
Non-volatile memory express (NVMe)	• v1.3	• QCS9075 • QCS8275
Secure digital (SD) card	• v3.0	• QCS6490 • QCS5430

1.1 Storage use cases

- Boot from the primary storage device: Store the required images to boot up the device by reading the images such as XBL, AOP, Qualcomm Trusted Execution Environment (TEE), and kernel, from storage.
- Save user-specific data: Store customized information such as videos, photos, and documents.
- Support read-only logical unit number (LUN)/region: Store images such as TEE and CDT. LUN is an independent processing unit on a UFS device.
- Stores data in an authenticated, access-controlled, replay-protected manner in RPMB LUN/partition: Replay protected memory block (RPMB) is a well-known LUN (WLUN) in UFS device.

Note:

- See Hardware SoCs that are supported on Qualcomm Linux.
- QCS8275 supports read or write operations only up to 8 GB.
- Detailed information on the storage hardware architecture is available to licensed developers with authorized access. For more information, see Qualcomm Linux Storage Guide - Addendum.

2 Storage features

The following features are supported for the various storage types on Qualcomm Linux.

2.1 UFS device features

The following UFS device features are supported.

Feature	Description
Number of lanes	Only 2 lanes are supported.
Gear	High-speed (HS) Gear 1, 2, 3, and 4 are supported.
Low-power mode	Supports link hibernate and device sleep modes. For more information about the supported modes, see UFS power management states.
Embedded UFS device	Supports only the embedded UFS device. Hot plug isn't supported.

2.2 SD card features

The following SD card features are supported.

Feature	Description
Number of bits	SD card operates in 4-bit mode.
Speed	• SDR50 • SDR104 • DDR50
Hot-plug capability	Supports hot plug capability.

2.3 NVMe features

The following NVMe features are supported.

Feature	Description
Lanes	4 lanes are supported for NVMe over the PCIe port (PCIe1).
Interface	NVMe provides a high-speed serial interface.
Protocol	The NVMe protocol optimizes NAND flash and next-generation SSD technologies, enabling efficient command processing and reducing overhead.

3 Configure storage features

Use the following storage configurations and device tree parameters for UFS and SD card functionalities.

3.1 Enable UFS device feature using kernel configuration

The following table lists the default values of the kernel configuration to enable UFS device.

UFS device kernel configuration

Configuration	Enabled by default (yes/no)	Description
CONFIG_UFS_FAULT_INJECTION	No	Injects the failure command for debugging.
CONFIG_SCSI_UFSHCD	Yes	Enables the UFS host control driver in the kernel.
CONFIG_SCSI_UFSHCD_PLATFORM	Yes	Supports the UFS controller from the platform bus.
CONFIG_SCSI_UFS_QCOM	Yes	Enables a Qualcomm-specific addition to access the PHY configuration and vendor-specific registers.
CONFIG_PHY_QCOM_QMP	Yes	Supports UFS device QMP PHY driver.
CONFIG_SCSI_UFS_BSG	Yes	Enables UFS device BSG device node.
CONFIG_SCSI_UFS_CRYPTO	Yes	Supports UFS device crypto engine.

3.2 Enable SD card using kernel configuration

The following table lists the default values of kernel configuration to enable SD card functionality.

SDHC kernel configuration

Configuration	Enabled by default (yes/no)	Comments
CONFIG_MMC	Yes	Enables MMC driver
CONFIG_MMC_DEBUG	No	Enables driver debug messages
CONFIG_MMC_SDHCI	Yes	Detects SDHCI driver
CONFIG_MMC_CQHCI	Yes	Enables Command Queue Host Control Interface
CONFIG_MMC_SDHCI_MSM	Yes	Enables MSM SDHCI driver
CONFIG_MMC_BLOCK	Yes	Enables MMC block driver

3.3 Modify UFS device power management states

To optimize power management, modify the UFS device power management states as required. However, altering the states can cause changes in power consumption.

Note: For more information about how to run SSH, see the Use SSH section.

UFS device power management states

State	Description
Auto-hibern8 (AH8)	• AH8 automatically transitions the PHY link to the Hibernate state after 150 ms of idle time. This value is adjustable. • To determine the AH8 idle duration value (in microseconds), run the following SSH command on the host computer. `cat /sys/devices/platform/soc@0/1d84000.ufs/auto_hibern8` • This value can be decreased to reduce power consumption.

State	Description

Clock gating	• When UFS device clocks are scaled up to maximum frequency due to the use case, clocks are gated after 50 ms of idle duration.
	• If clocks are scaled down to optimal frequency, clocks are gated after 10 ms of idle duration. To check clock gating enabled/disabled status, run the following SSH command on the host computer.
	`cat /sys/devices/platform/soc@0/1d84000.ufs/clkgate_enable`
	• Clock gating is enabled by default.
	• To debug UFS device issues related to clock gate or ungate and to disable the clock gating, run the following SSH command on the host computer.
	`echo 0 > /sys/devices/platform/soc@0/1d84000.ufs/clkgate_enable`

State	Description

Clock scaling	• After the UFS device load exceeds the upthreshold value, the clock frequency is set to the maximum frequency with clock scale up.
	• Clock scale-down occurs when the load is lower than the threshold. The UFS device operations determines its load. To check the clock scaling status, run the following SSH command on the host computer.
	`cat /sys/devices/platform/soc@0/1d84000.ufs/clkscale_enable`
	• Clock scaling is enabled by default.
	• Disabling clock scaling can result in increased power consumption. To debug any issue related to clock scaling, run the following SSH command on the host computer.
	`echo 0 > /sys/devices/platform/soc@0/1d84000.ufs/clkscale_enable`

Runtime suspend or resume	• UFS device enters the runtime suspend state after 3 seconds of inactivity. • Runtime resume state occurs with the next UFS device operation. This state is enabled by default and is not configurable.
System suspend or resume	• UFS device enters the system suspend with the suspend event. • UFS device resume state occurs with the system resume event. • This state is enabled by default and is not configurable.

3.4 Modify SD card power management states

SD card power management states

State	Description
Runtime suspend or resume	• The SD card enters the runtime suspend state after 50 ms of inactivity. • Resume occurs with the next SD card operation. This state is enabled by default and is not configurable. • The eMMC transitions to the runtime suspend state after 50 milliseconds of inactivity. It resumes with the next I/O operation. This feature is always enabled and cannot be configured.
System suspend or resume	• The SD card enters system suspend state with the suspend event. • SD card system resume state occurs with the system resume event. • This state is enabled by default and is not configurable. • The eMMC enters system suspend mode upon receiving a suspend event and resumes operation with a system resume event. This functionality is always enabled and cannot be configured.

3.5 Configure device using UFS parameters

The device tree specifies the UFS host configuration parameters such as number of gears and lanes, rate value, and timeout values.

The device tree is at `<workspace_root_path>/boot_images/boot/Settings/Soc/Kodiak/Core/Storage/UFS/ufs.dtsi`.

For QCS9075, see `<workspace_root_path>/boot_images/boot/Setting/soc/LeMans/Core/Storage/UFS/ufs.dtsi`.

The following table lists a sample UFS device tree node in UEFI.

Device tree node/key	Subnode	Device tree value	Description
`init_speed_params`	`EnableHighSpeed`	1	Enables high speed mode
	`NumGears`	4	Specifies the maximum gear to be used
	`NumLanes`	2	Number of lanes used in UFS device
	`Rate`	2	Values are: • 1 = Rate A • 2 = Rate B
`perf_speed_params`	`EnableHighSpeed`	1	Enables high speed mode while in perf mode
	`NumGears`	4	Specifies the maximum gear to be used in perf mode
	`NumLanes`	2	Number of lanes used in UFS device in perf mode
	`Rate`	2	Series in perf mode - 1 = Rate A - 2 = Rate B
`timeout_values`	`fDeviceInitTimeoutUs`	2500000	Denotes timeout (μs) during device init
	`UTRDPollTimeoutUs`	30000000	Denotes timeout (μs) for UTRD
`BatteryThresholdMv`	–	3600	Battery threshold (in milli volts) to move to Gear1
`LinkStartupRetryCount`	–	5	Retry count for Link startup
`FUA_Value`	–	1	Forced unit access
`EnableLogging`	–	0	Enables DT serial debug logging in loader - 1 - Enable - 0 - Disable
`RefClock`	–	19200000	UFS device controller reference clock 19.2 MHz
`MphyInitTable`	–	–	Denotes MphyInit table
`MphyInitEndTable`	–	–	Denotes MphyInitEnd table
`MphyLaneInitTable`	–	–	Denotes Mphy 2 Lane Init table

Compile all changes in the `.dtsi` to `xbl_config.elf` and reflash the image to the device.

Qualcomm tunes the values of the `MphyInitTable`, `MphyInitEndTable`, and `MphyLaneInitTable` configurations for UFS device PHY initialization. For debugging and tuning any PHY-level issue, changes these configuration values only with guidance from Qualcomm.

The default values are for UFS device 3.1-compliant part. For UFS device 2.x-compliant part,

`NumGears` value should be 3. Don't changes other parameters for the UFS device 2.x part.

3.6 Configure device using SD card parameters

The device tree specifies the SD card parameters such as drive strength, maximum speed mode, and HSR settings. The device tree is at `<workspace_root_path>/boot_images/boot/Settings/Soc/Kodiak/Core/Storage/SDCC/sdcc.dtsi`.

The following is an example of the SD card device tree node in UEFI.

Device tree node/key	Device tree value	Description
`DriveStrength`	0x1FE4	`DriveStrength` as recommended, sets SDC CLK, CMD, and data drive strengths to 7, 4, 4, respectively.
`MaxSpeedMode`	`SDCC_SDR104_MODE`	Defines maximum speed mode. For SD card, the value is `SDCC_SDR104_MODE`.

3.7 Configure UFS Linux kernel device tree

The SoC DTSI node specifies register address space, clocks, interrupts, and reset information. The platform DTSI node specifies power supply, voltage, and current levels. The `Ufs_mem_hc` node that describes the on-chip UFS device host controller is at `<workspace_root_path>/sources/kernel/kernel_platform/kernel/arch/arm64/boot/dts/qcom/sc7280.dtsi`.

For `Ufs_mem_hc` node in QCS9075, see `<workspace_root_path>/sources/kernel/kernel_platform/kernel/arch/arm64/boot/dts/qcom/sa8775p.dtsi`.

Note: These parameters are only for information purposes. Don't modify them.

Device tree parameters

Property	Description
`Compatible`	For Qualcomm SoCs, must contain strings such as `qcom`, `ufshc`.
`Interrupts`	Interrupt mapping for UFS host controller IRQ.
`Reg`	UFS host registers address mapping.
`Phys`	`phandle` to UFS device PHY node.
`lanes-per-direction`	Specify the number of lanes available per direction. Either 1 or 2.
`clock-names`	List of clock input name strings.
`Clocks`	List of `phandle` and clock specifier pairs.

Property	Description
freq-table-hz	Array of <minimum maximum> operating frequencies stored in the same order as the clocks property.
reset-gpios	A phandle and GPIO specifier denoting the GPIO that is connected.
Resets	Reset node register.

The Ufs_mem_phy node describes on-chip UFS device PHY hardware and this node is in <workspace_root_path>/sources/kernel/kernel_platform/kernel/arch/arm64/boot/dts/qcom/sc7280.dtsi file.

Property	Description
Compatible	Specify compatible string such as qcom, qmp-ufs-phy.
#phy-cells	Set the property to 0.
Reg	Should contain PHY register address space.
reg-names	• Indicates various resources passed to driver (through the reg property) by name. • The required reg-names is phy_mem.
lanes-per-direction	Number of lanes available per direction; either 1 or 2.
clock-names	List of clock input name strings.
clocks	List of phandle and clock specifier pairs.
vdda-phy-supply	phandle to main PHY supply for analog domain.
vdda-pll-supply	phandle to PHY PLL and Power-Gen block power supply.
Resets	Specifies the PHY reset in the UFS device controller.

For more information about DTS parameters, see /kernel_platform/msm-kernel/Documentation/devicetree/bindings/ufs/qcom,ufs.yaml.

3.8 Configure SD card Linux kernel device tree

The SoC DTSI node specifies register address space, clocks, interrupts, and reset information. The platform DTSI node specifies power supply, voltage, and current levels. The sdhc_2: mmc@8804000 node describes the on-chip SDHC host controller. This node is in the

<workspace_root_path>/sources/kernel/kernel_platform/kernel/arch/arm64/boot/dts/qcom/sc7280.dtsi file.

Note: These parameters are only for information purposes. Don't modify them.

Property	Description
Compatible	Specify compatible strings such as qcom,sc7280-sdhci, qcom, sdhci-msm-v5.
Pinctrl-names	Defines name of the pin control states.
Pinctrl-0 &1	Points to pin control settings, this is an array defines pin control settings for multiple states.
Interrupts	Interrupt mapping for SDHC IRQ.
Interrupt-names	Lists the names corresponding to each interrupt defined in the interrupts property.
reg	SDHC host registers address mapping.
iommus	Iommus specifies the IOMMU node and stream ID.
clock-names	List of clock input name strings.
clocks	List of phandle and clock specifier pairs.
interconnects	Define the interconnect paths for the device.
interrupt-names	Lists the names to each interconnect path defined in interconnects property.
power-domains	Specifies the power domain used by the SDHC.
bus-width	Specifies the number of data lines used for communication.
dma-coherent	Specifies that the SDHCI and CPU maintain cache coherency.
qcom,dll-config	Platform specific settings for DLL_CONFIG reg.
resets	Reset node register.
sdhc2_opp_table	This table defines various operating points supported by the SDHCI.

3.9 Provision UFS device

Creating and defining LUNs on a new UFS device is called Provisioning. A UFS device contains one or more logical units. A logical unit is an independent processing entity within the device. Each logical unit has an address within the UFS device called a Logical Unit Number (LUN).

The configuration file for provisioning is at <build root/common/config/ufs/provision.

The provision XML files for the various UFS device versions are as follows:

- provision_ufs22.xml - This is for UFS device 2.2 part
- provision_ufs31.xml - This is for UFS device 3.1 part

In the UFS device provision XML file, the UFS Device Descriptor is a section that contains

essential information about the UFS device being provisioned. The Device Descriptor includes details such as number of Luns, initial power modes, write boost parameters. Here is a sample device descriptor entry:

```
<ufsbNumberLU=”6” bBootEnable=”1” bDescrAccessEn=”0”
bInitPowerMode=”1” bHighPriorityLUN=”0x5” bSecureRemovalType=”0”
bInitActiveICCLevel=”0” wPeriodicRTCUpdate=”0” bConfigDescrLock=”0”
bWriteBoosterBufferPreserveUserSpaceEn=”1”
bWriteBoosterBufferType=”1”
shared_wb_buffer_size_in_kb=”4194304” />
```

The provision XML file also describes each LUN with Unit Descriptor parameters such as Logical Unit Enable, Boot LUN ID. The following is a sample LUN descriptor entry:

```
<ufs LUNum=”0” bLUEnable=”1” bBootLunID=”0” size_in_kb=”4096”
bDataReliability=”0” bLUWriteProtect=”0” bMemoryType=”0”
bLogicalBlockSize=”0x0c”bProvisioningType=”2”
wContextCapabilities=”0”  wb_buffer_size_in_kb=”0”
desc=”LU 0 - User LUN - Rest of the device” />
```

The following table describes some important parameters used in the *provision XML* files.

UFS device parameters description

Parameter	Description
`bNumberLU`	Number of Logical Units `bNumberLU` does not include well-known logical units
`bBootEnable`	Indicates whether the device is enabled for boot: • 00h: Boot feature disabled • 01h: Bootable feature enabled
`bDescrAccessEn`	Indicates whether the Device Descriptor can be read after the partial initialization phase of the boot sequence: • 00h: Device Descriptor access disabled • 01h: Device Descriptor access enabled

Parameter	Description
`BInitPowerMode`	Defines the Power Mode after device initialization or hardware reset: • 00h: UFS-Sleep Mode • 01h: Active Mode
`bHighPriorityLUN`	Defines the high-priority logical unit
`bSecureRemovalType`	• 00h: Information removed by an erase of the physical memory • 01h: Information removed by overwriting the addressed locations with a single character followed by an erase • 02h: Information removed by overwriting the addressed locations with a character, its complement, then a random character • 03h: Information removed using a vendor-defined mechanism
`bConfigDescrLock`	• 0h: Configuration Descriptor not locked • 1h: Configuration Descriptor locked

Note:

- To reprovision the UFS device, set `bConfigDescrLock="0"`. This parameter is part of the provision XML file.
- To prevent reprovisioning, set `bConfigDescrLock="1"` after the UFS device configuration and provisioning is finalized.

UFS device LUN parameters description

Parameter	Description
`LUNum`	Logical Unit Number
`bLUEnable`	Enables/disables the logical unit: • 00h: Disables the logical unit • 01h: Enables the logical unit

Parameter	Description
`bBootLunID`	Boot LUN ID: • 00h: Not bootable • 01h: Boot LU A • 02h: Boot LU B
`size_in_kb`	Specifies LUN size in KBs
`bLUWriteProtect`	Logical Unit Write Protect: • 00h: LU is not write protected • 01h: LU write protected when `fPowerOnWPEn = 1`
`bMemoryType`	Defines logical unit memory type: • 00h: Normal memory • 01h: System code memory type • 02h: Non-Persistent memory type • 03h: Enhanced memory type 1 • 04h: Enhanced memory type 2 • 05h: Enhanced memory type 3 • 06h: Enhanced memory type 4
`bProvisioningType`	Provisioning type • 00h: Thin Provisioning is disabled • 02h: Enables thin provisioning with `TPRZ = 0`

UFS device partition layout

The default UFS device provisioning creates eight LUNs (LUN0 to LUN7).

- The Linux data and all Linux file system images are stored in LUN0.
- Boot well-known LUN (WLUN) 0xB0 alternates between LUN1 and LUN2 to provide a fail-safe backup for the XBL.
- The rest of the boot chain is stored in LUN4.
- The LUN6 and LUN7 are unused by default.

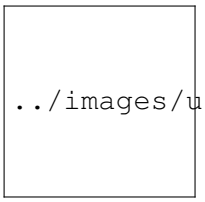

Figure1 Default UFS device layout

4 Customize storage features

This section explains how to customize the storage features, such as adding a new partition and mounting the SD card to a specific path.

Note: All Qualcomm SoC UFS device components are validated for the preferred vendor list (PVL). Select the applicable part from the Memory list.

4.1 Mount the file system

EXT4 is an open-source file system that's used on data partitions on the Qualcomm Linux. The EXT4 file system can support volumes with sizes up to 64 ZB, and single files with sizes up to 16 TB with the standard 4 kB block size. For more information about EXT4, see https://opensource.com/article/17/5/introduction-ext4-filesystem.

An extent is a range of contiguous physical blocks, improving large-file performance and reducing fragmentation. A single extent in EXT4 can map up to 128 MiB of contiguous space with a 4 kB block size. EXT4 doesn't limit the number of subdirectories in a single directory, except by the inherent size limit of the directory itself.

For example:

```
/dev/sda3 on /usr type ext4 (ro,relatime,inlinecrypt)
```

1. Mount `/dev/sda3` to `/data`.

```
mkdir /tmp/data
```

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

```
mount -t ext4 /dev/sda3 /tmp/data
```

2. Verify whether the `/dev/sda3` partition is mounted.

```
df -a
```

Output:

```
...
Filesystem      Size    Used  Avail  Use%  Mounted on

/dev/sda3       108.2G  3.7G  99.9G  4%    /tmp/data
```

3. Mount a file system with specific SELinux context options.

```
mount -o rootcontext=system_u:object_r:etc_t:s0 -t ext4 /dev/
sda3 /home/root/data
```

Note: Run the `mount` command to retrieve the exact `rootcontext=<>` value related to the `/dev/sda3`.

Mount the SD card to a specific path

SD cards are automounted on card insertion. To manually mount the SD card, run the following commands:

1. Remount the root partition as read or write-enabled.

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

2. Create an SD card directory.

```
mkdir sdcard
```

3. Locate the default path for the SD card directory.

```
root@qcs6490-rb3gen2-vision-kit:~# pwd
```

Output:

```
/ (<- default path)
```

4. If the SD card isn't formatted as VFAT, format it using the following command.

```
mkfs.vfat /dev/mmcblk1p1
```

1. Mount the SD card to the default path (`/sdcard`).

   ```
   mount -t vfat /dev/mmcblk1p1 /<default_path>/sdcard
   ```

   Note: If the SD card has multiple partitions, repeat step 4 to mount each partition.

2. Verify whether the SD card is mounted.

   ```
   df -a
   ```

   Output:

   ```
   ...
   Filesystem      Size    Used  Avail  Use%  Mounted on
   /dev/mmcblk1p1  511.0M  4.0K  511.0M  0%   /sdcard
   ```

Mount the NVMe card

To format and mount the new or existing NVMe cards using *fdisk*, do the following:

1. Create a partition table by running the following command.

   ```
   fdisk /dev/nvme0n1
   ```

   Output:

   ```
   Command (m for help): i
   No partition is defined yet!
   Command (m for help): n
   Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
   Select (default p):  (press enter)
   Using default response p.
   Partition number (1-4, default 1):  (press enter)
   First sector (2048-1953525167, default 2048):  (press enter)
   Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-1953525167,
   default 1953525167):
   Created a new partition 1 of type 'Linux' and of size 931.5 GiB.
   Command (m for help): w
   ```

```
The partition table has been altered.
Syncing disks.
root@lemans:~#
```

2. Reboot the board.

Note: Locate the `mkfs.ext4` binary at `/usr/sbin/mkfs.ext4`.

3. Format the `/dev/nvme0n1p1` partition.

```
mkfs.ext4 /dev/nvme0n1p1
```

Output:

```
mke2fs 1.46.5 (30-Dec-2021)
Discarding device blocks: done
Creating filesystem with 244190390 4k blocks and 61054976 inodes
Filesystem UUID: 94ca7f8e-eb27-45ce-84f6-67f0cf977ca8
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632,
2654208,
4096000, 7962624, 11239424, 20480000, 23887872, 71663616,
78675968,
102400000, 214990848
Allocating group tables: done
Writing inode tables: done
Creating journal (262144 blocks): done
Writing superblocks and filesystem accounting information: done
```

4. Mount the NVMe partition1 to the media folder.

```
mount /dev/nvme0n1p1 /media
```

5. Verify whether the `nvme0n1p1` partition is mounted.

```
df -a
```

Output:

```
...
```

```
Filesystem      Size  Used  Avail  Use%  Mounted on
/dev/nvme0n1p1  458G   28K  435G   1%    /var/rootdirs/
media
```

5 Debug common storage driver issues

5.1 UFS device issue

The kernel log records *cmd pending in the device* or *no response from device* when a UFS device issue occurs. For more information, reach out to the UFS device vendor.

Example 1:

```
[ 82.928056][01-01 00:02:05] ufshcd-qcom 1d84000.ufshc:
ufshcd_abort: Device abort task at tag 1
[ 82.928059][01-01 00:02:05] sd 0:0:0:4: [sde] tag#1 CDB:
Read(10) 28 00 00 00 44 49 00 00 20 00
[ 82.936477][01-01 00:02:05] ufshcd-qcom 1d84000.ufshc:
UPIU[1] - issue time 50955434 us
[ 82.936479][01-01 00:02:05] ufshcd-qcom 1d84000.ufshc:
UPIU[1] - complete time 0 us
[ 82.936567][01-01 00:02:05] ufshcd-qcom 1d84000.ufshc:
ufshcd_abort: cmd pending in the device. tag = 1
```

Example 2 :

```
[ 1036.008147] [1230_22:00:49]@0 ufshcd-qcom 1d84000.ufshc:
ufshcd_abort: Device abort task at tag 2
[ 1036.008510] [1230_22:00:49]@0 sd 0:0:0:0: tag#2 CDB:
opcode=0xfa (vendor) fa 02 01 48 07 79 ae 03 50 00
[ 1036.120287] [1230_22:00:50]@1 ufshcd-qcom 1d84000.ufshc:
ufshcd_abort: no response from device. tag = 2, err -110.
```

Physical adapter or data link error

In the kernel log, `pa_err_cnt_total` and `dl_err_cnt_total` with non-zero values indicate a physical adapter or data link (PA/DL) error issue. This issue occurs when the signal quality between the AP and the UFS device isn't strong, and requires modification in all three components, such as hardware, software, and the UFS device vendor to resolve.

To view the PA/DL error count, use the `debugfs` path `/sys/kernel/debug/ufshcd/1d84000.ufs/stats`.

Mount `debugfs` using the following command. This command ensures that the `debugfs` file system remains accessible for the debug nodes.

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

Example:

```
[31709.438001] ufshcd-qcom 1d84000.ufshc: pa_err_cnt_
total=5086, pa_lane_0_err_cnt=3250, pa_lane_1_err_cnt=1405,
pa_line_reset_err_cnt=431
[31709.438008] ufshcd-qcom 1d84000.ufshc: dl_err_cnt_
total=4488, dl_nac_received_err_cnt=289, dl_tcx_replay_
timer_expired_err_cnt=258
[31709.438015] ufshcd-qcom 1d84000.ufshc: dl_afcx_request_
timer_expired_err_cnt=0, dl_fcx_protection_timer_expired_
err_cnt=0, dl_crc_err_cnt=18
```

Active state UFS device power consumption issue

Use the following steps to debug UFS device high power consumption during active state.

1. Check how often the issue occurs. Use the following SSH command to check current clock frequency.

   ```
   cat /sys/devices/platform/soc@0/1d84000.ufs/
   devfreq/1d84000.ufs/cur_freq
   ```

   Note: If the frequency is maximum, skip to Step 3.

2. Check whether any UFS device-related feature (such as WriteBooster) is enabled.
3. Check whether clock scale is disabled.

```
cat /sys/devices/platform/soc@0/1d84000.ufs/
clkscale_enable
```

4. Monitor the UFS device clocks frequency and UFS device rails.

   The issue may occur due to the initial setting or due to an unexpected devfreq event.

Sleep state power consumption issue

During sleep state, if UFS device power consumption is high, use the following steps to debug UFS device suspend-related issues.

1. Check the power management state using sysfs nodes `link_state`, `curr_freq`, and `runtime_status` and PM level (`PM_LVL` 3 has higher power consumption than 5).
2. Check the UFS device state using sysfs node `dev_pm` to determine whether UFS device is in sleep mode.
3. Check whether the HPB and write booster features are enabled because they cause high power consumption.
4. Check the UFS device VCC/VCCQ/VCCQ2 timing from the hardware side if an issue occurs.

Unhandled UFS device SMMU issue

System memory management Unit (SMMU) is a memory management unit (MMU) that connects a direct-memory-access-capable (DMA-capable) I/O bus to the physical memory.

Each device is assigned a unique stream ID (SID). The UFS device controller SID value is 0x80.

For unhandled SMMU fault from UFS device SMMU, disable the UFS device SMMU.

```
arm-smmu 15000000.apps-smmu: Unhandled arm-smmu context fault from
1d84000.ufshc!
```

To disable UFS device SMMU, modify the `ufshc_mem` node in `<workspace_root_path>/sources/kernel/kernel_platform/kernel/arch/arm64/boot/dts/qcom/sc7280.dtsi` file.

```
-    qcom,iommu-dma = "fastmap";
+    qcom,iommu-dma = "bypass";
```

5.2 Debug with sysfs or debugfs nodes

To identify the status or update the state, use the *sysfs* and *debugfs* nodes.

Mount *debugfs* to ensure that the *debugfs* file system is accessible for the debug nodes.

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

UFS device state commands

Required UFS device state information	Command
Check PA/DL error count	`cat /sys/kernel/debug/ufshcd/1d84000.ufs/stats`
Read UFS device gating delay, in ms	`cat /sys/devices/platform/soc@0/1d84000.ufs/clkgate_delay_ms`
Read UFS device clocks current frequency at given point	`cat /sys/devices/platform/soc@0/1d84000.ufs/devfreq/1d84000.ufs/cur_freq`
Read UFS device I/O scheduler details	`cat /sys/devices/platform/soc@0/1d84000.ufs/devfreq/1d84000.ufs/governor`
Read UFS device min frequency during clock scale down	`cat /sys/devices/platform/soc@0/1d84000.ufs/devfreq/1d84000.ufs/min_freq`

Required UFS device state information	Command
Read UFS device max frequency during clock scale up	`cat /sys/devices/platform/soc@0/1d84000.ufs/devfreq/1d84000.ufs/max_freq`
Read UFS device state active or sleep	`cat /sys/devices/platform/soc@0/1d84000.ufs/power_info/dev_pm`
Read UFS device gear information such as HS_G3, HS_G4	`cat /sys/devices/platform/soc@0/1d84000.ufs/power_info/gear`
Read UFS device number of lanes	`cat /sys/devices/platform/soc@0/1d84000.ufs/power_info/lane`
Read UFS device link state, ACTIVE or HIBERN8	`cat /sys/devices/platform/soc@0/1d84000.ufs/power_info/link_state`
Read UFS device rate information such as Rate A or B	`cat /sys/devices/platform/soc@0/1d84000.ufs/power_info/rate`
Read UFS device descriptor information	`ls /sys/devices/platform/soc@0/1d84000.ufs/device_descriptor`
Read UFS device health descriptor information	`ls /sys/devices/platform/soc@0/1d84000.ufs/health_descriptor`
Read UFS device runtime status information active or suspended	`cat /sys/devices/platform/soc@0/1d84000.ufs/power/runtime_status`
Find and/or modify the SPM/RPM level	`cat /sys/devices/platform/soc@0/1d84000.ufs/rpm_lvl` `cat /sys/devices/platform/soc@0/1d84000.ufs/spm_lvl`

SD card state commands

Required SD card state information	Command
Read SD card error stats information	`cat /sys/kernel/debug/mmc1/err_stats`
Read SD card latest error state details	`cat /sys/kernel/debug/mmc1/err_state`
Check the current running clock details	`cat /sys/kernel/debug/mmc1/clock`
Read the SD card runtime status information (active or suspended)	`cat /sys/devices/platform/soc@0/8804000.mmc/power/runtime_status`

Required SD card state information	Command
Read SD card auto suspend delay time	`cat /sys/devices/platform/soc@0/8804000.mmc/power/autosuspend_delay_ms`
Read SD card runtime active time	`cat /sys/devices/platform/soc@0/8804000.mmc/power/runtime_active_time`
Read SD card runtime suspend time	`cat /sys/devices/platform/soc@0/8804000.mmc/power/runtime_suspended_time`
Read SD card uevent details	`cat /sys/devices/platform/soc@0/8804000.mmc/uevent`

NVMe state commands

Required NVMe card state information	Command
View device model name	`cat /sys/class/nvme/nvme0/model`
View device serial number	`cat /sys/class/nvme/nvme0/serial`
View current firmware version	`cat /sys/class/nvme/nvme0/firmware_rev`
View namespace identifier	`cat /sys/class/nvme/nvme0/nvme0n1/nsid`
View namespace size	`cat /sys/class/nvme/nvme0/nvme0n1/size`
View the offset in bytes to align the start of the NVMe namespace to the optimal I/O boundary	`cat /sys/class/nvme/nvme0/nvme0n1/alignment_offset`
View queue depth	`cat /sys/class/nvme/nvme0/queue_count`

6 References

6.1 Related documents

Title	Resource
Qualcomm Technologies, Inc.	
UFS device components	https://www.qualcomm.com/products/internet-of-things/industrial/building-enterprise/qcs6490/peripherals
Qualcomm Linux Storage Guide - Addendum	https://docs.qualcomm.com/bundle/resource/topics/80-70018-6A/overview-addendum.html
Hardware SoCs	https://docs.qualcomm.com/bundle/publicresource/topics/80-70018-115/soc.html
Secure shell	https://docs.qualcomm.com/bundle/publicresource/topics/80-70018-254/how_to.html
Standards	
EXT4	https://opensource.com/article/17/5/introduction-ext4-filesystem

6.2 Acronyms and terms

Acronym or term	Definition
AH8	Auto-hibern8
DTSI	Device tree source inclusion
GPIO	General purpose input output
JEDEC	Joint electron device engineering council
LUN	Logical unit number
NVMe	Non-volatile memory express
PVL	Preferred vendor list
RPMB	Replay protected memory block
SD	Secure digital

Acronym or term	Definition
SDHCI	Secure digital host controller interface
SSD	Solid state drive
UFS	Universal flash storage
WLUN	Well-known logical unit number

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.