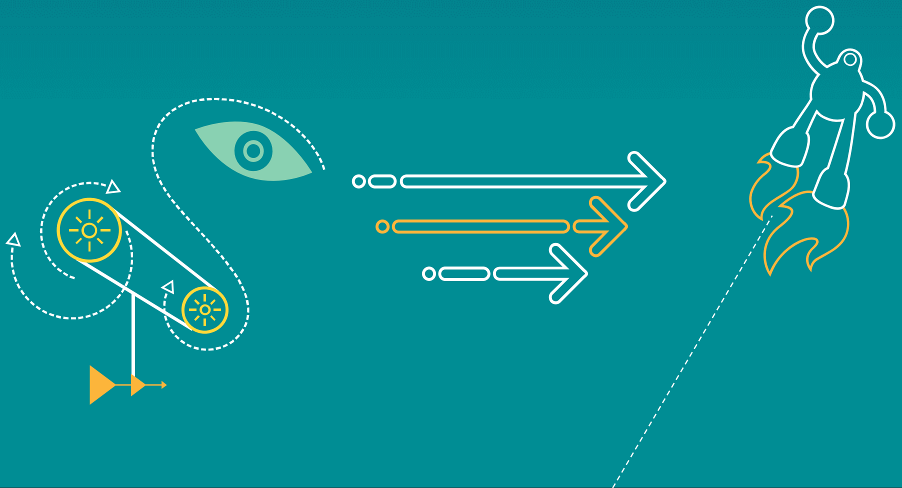
MSM8953 Linux Android Thermal Management Overview

Qualcomm Technologies, Inc.

80-P3255-6 Rev. E



Confidential and Proprietary – Qualcomm Technologies, Inc.

NO PUBLIC DISCLOSURE PERMITTED: Please report postings of this document on public servers or websites to: DocCtrlAgent@qualcomm.com.

Restricted Distribution: Not to be distributed to anyone who is not an employee of either Qualcomm Technologies, Inc. or its affiliated companies without the express approval of Qualcomm Configuration Management.

Confidential and Proprietary – Qualcomm Technologies, Inc.

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

Revision History

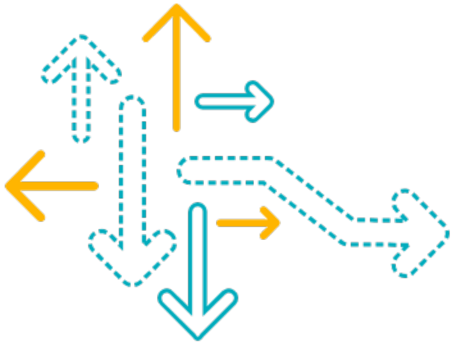
Revision	Date	Description
A	November 2015	Initial release
B	April 2016	Added slides 8 and 9
C	May 2016	Updated slide 32
D	August 2016	Numerous updates were made to the doc; must be read in its entirety
E	October 2017	Numerous changes were made in the section *Modem Thermal Management*; to be read in its entirety.

Contents

- Objectives
- Thermal Mitigation Software Concept Architecture
- Thermal Management Software
- Boot Thermal Management (BTM) Algorithm
- Kernel Thermal Monitor (KTM)
- User Space Thermal Engine
- Modem Thermal Management
- Thermal Configuration
- New Thermal Features
- Thermal Management Mechanism
- Thermal Software Features and Management Devices
- Thermal Debug Overview
- References
- Questions?

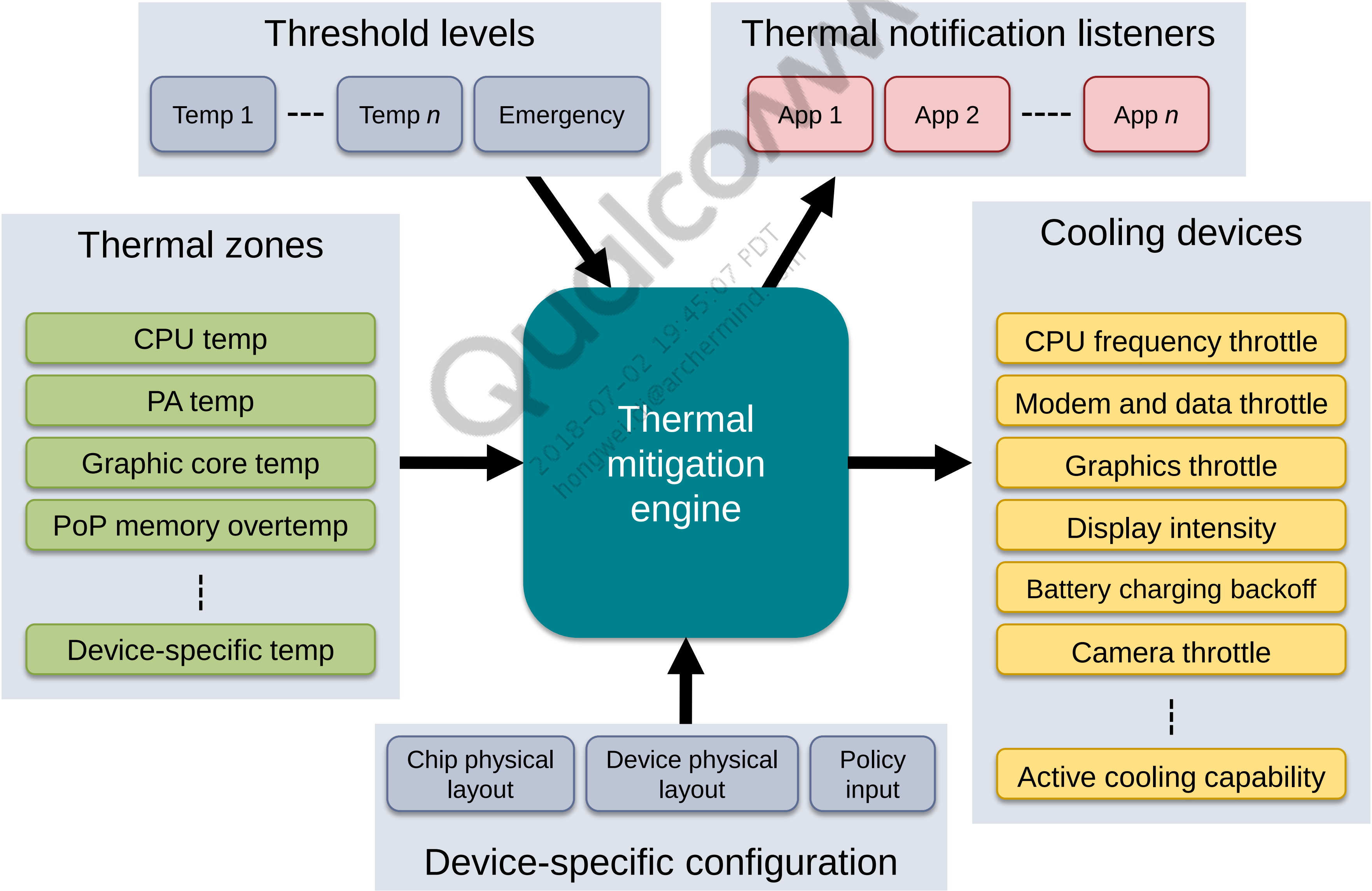
Objectives

- At the end of this presentation, you will understand the Linux Android thermal management features for the MSM8953 chipset.



Thermal Mitigation Software Concept Architecture

Overview


Threshold levels
Temp 1
Temp *n*
Emergency
Thermal notification listeners
App 1
App 2
App *n*
Thermal zones
CPU temp
PA temp
Graphic core temp
PoP memory overtemp
Device-specific temp
Thermal mitigation engine
Cooling devices
CPU frequency throttle
Modem and data throttle
Graphics throttle
Display intensity
Battery charging backoff
Camera throttle
Active cooling capability
Chip physical layout
Device physical layout
Policy input
Device-specific configuration

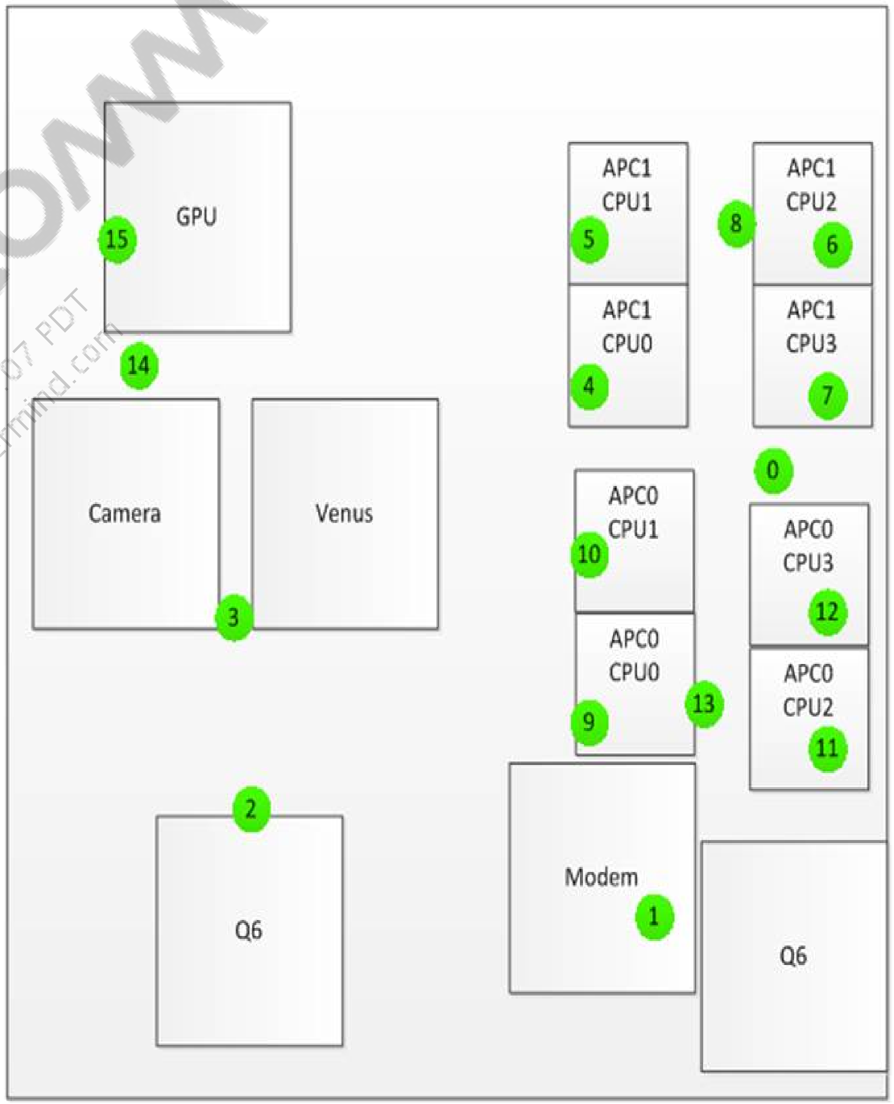
On-Die Temperature Sensor Floor Plan of MSM8953 Chipset

- MSM8953 chipset has 16 on-die sensors. More sensors (thermistors) are needed on devices (PA, XO, BATT, PMIC, and case thermistors) to tune thermal.
- Improved on-die sensor accuracy and sampling rate due to the new hardware architecture.
- Hardwired tsens_reset occurs, when predefined critical high or low threshold in SBL1 is exceeded, to protect the device.

boot_images\core\hwengines\tsens\config\<Target>\TsensBootBsp.c

```
        /* .nCriticalMin   */ -35,
        /* .nCriticalMax   */ 120
```

Cluster_CPU	Logical CPU #	TSENS #
APC0_CPU0	CPU0	9
APC0_CPU1	CPU1	10
APC0_CPU2	CPU2	11
APC0_CPU3	CPU3	12
APC1_CPU0	CPU4	4
APC1_CPU1	CPU5	5
APC1_CPU2	CPU6	6
APC1_CPU3	CPU7	7



Thermal Sensors’ Accuracy Improvement

- Thermal zone on-die sensor accuracy and the sampling rate are greatly improved because of the new hardware architecture supported in MSM8953 chipset. The MSM8953 TSENS hardware can capture the decidegree value compared to all QTI targets.
- The thermal sensor information of kernel in DTSI node lists all sensors that the thermal daemon uses with the information, such as sensor name, sensor type (for example, tsens, ADC, or pmic alarm sensor), sensor alias (for example, core0, pop_mem), and sensor scaling factor.

Example:

```
sensor_information0: qcom,sensor-information-0 {
               qcom,sensor-type = "tsens";
               qcom,sensor-name = "tsens_tz_sensor0";
                           qcom,scaling-factor = <1>;              -> temp reading in degree
   (MSM8937/MSM8976/52/older targets), default value will be 1
                   };


sensor_information0: qcom,sensor-information-0 {
               qcom,sensor-type = "tsens";
               qcom,sensor-name = "tsens_tz_sensor0";
                           qcom,scaling-factor = <10>;             -> temp reading in decidegree
   (MSM8953)
                   };

 sensor_information16: qcom,sensor-information-16 {
               qcom,sensor-type =  "alarm";
               qcom,sensor-name = "pm8953_tz";
               qcom,scaling-factor = <1000>;                        -> temp reading in millidegree
       };
```

Thermal Sensors’ Accuracy Improvement (cont.)

- The thermal sensor information is exported to user space with the following read only sysfs node:

  /sys/module/msm_thermal/sensor_info

- A single sysfs only provides information for all sensors listed in the dtsi node per the following format:

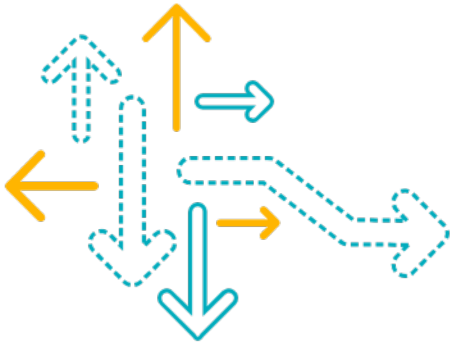
  <sensor_type>:<sensor_name>:<sensor alias if any>:<scaling factor>

Example:

```
root@msm8953_64:/ # cat /sys/module/msm_thermal/sensor_info
tsens:tsens_tz_sensor0::1
tsens:tsens_tz_sensor1::1
tsens:tsens_tz_sensor2:pop_mem:1
tsens:tsens_tz_sensor3::1
tsens:tsens_tz_sensor4:L2_cache_1:1
tsens:tsens_tz_sensor5:cpu0:1
tsens:tsens_tz_sensor6:cpu1:1
tsens:tsens_tz_sensor7:cpu2:1
tsens:tsens_tz_sensor8:cpu3:1
tsens:tsens_tz_sensor9:cpu4-cpu5-cpu6-cpu7:1
tsens:tsens_tz_sensor10:gpu:1 adc:pa_therm0::1 adc:pa_therm1::1 adc:xo_therm::1
adc:xo_therm_buf::1 adc:case_therm::1 alarm:pm8937_tz::1000
```

Decidegree output:

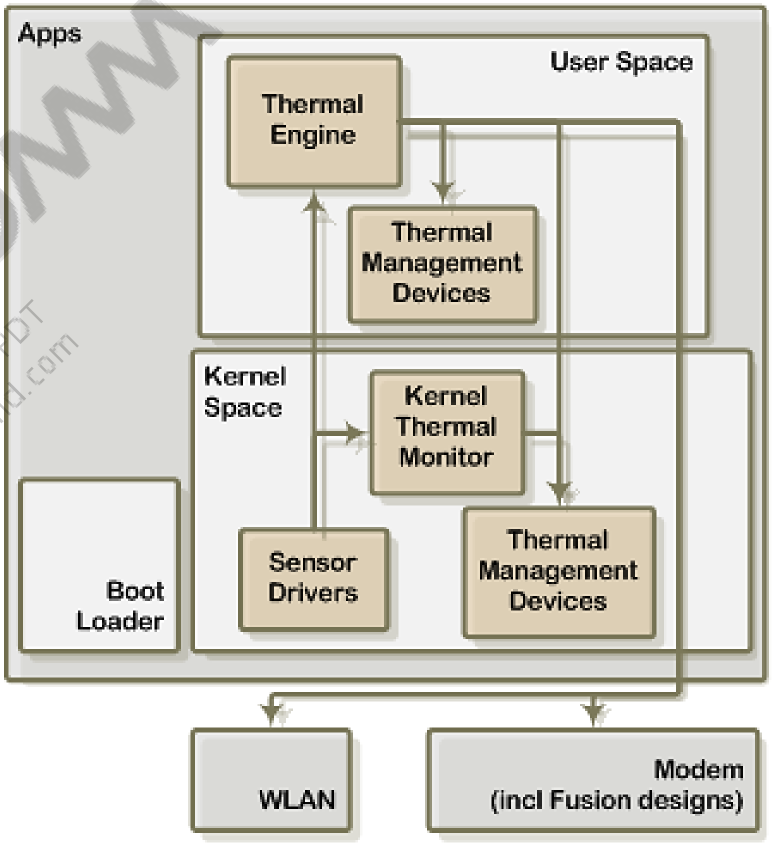
```
# cat /sys/class/thermal/thermal_zone2/temp348
```

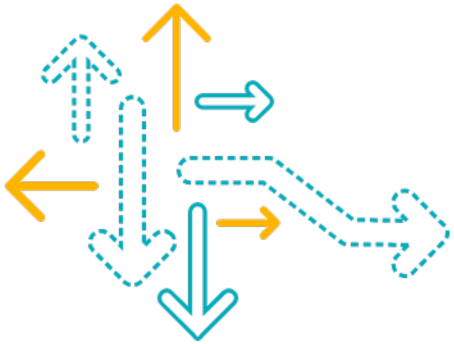


Thermal Management Software

Overview

- Thermal management software goals:
  - Manage the chip junction (Tj) temperature limits (typically 85°C)
  - Manage the external device skin temperature limits (typically 45°C)
- Sensors:
  - Sensors on the MSM™ chipsets die
  - Board thermistors – PMIC, PA, XO, and so on
- Management devices:
  - Passive cooling applied by reducing performance.
  - Device selection and threshold are configurable by the OEM to tune for ID variability through a configuration file.





Boot Thermal Management (BTM) Algorithm

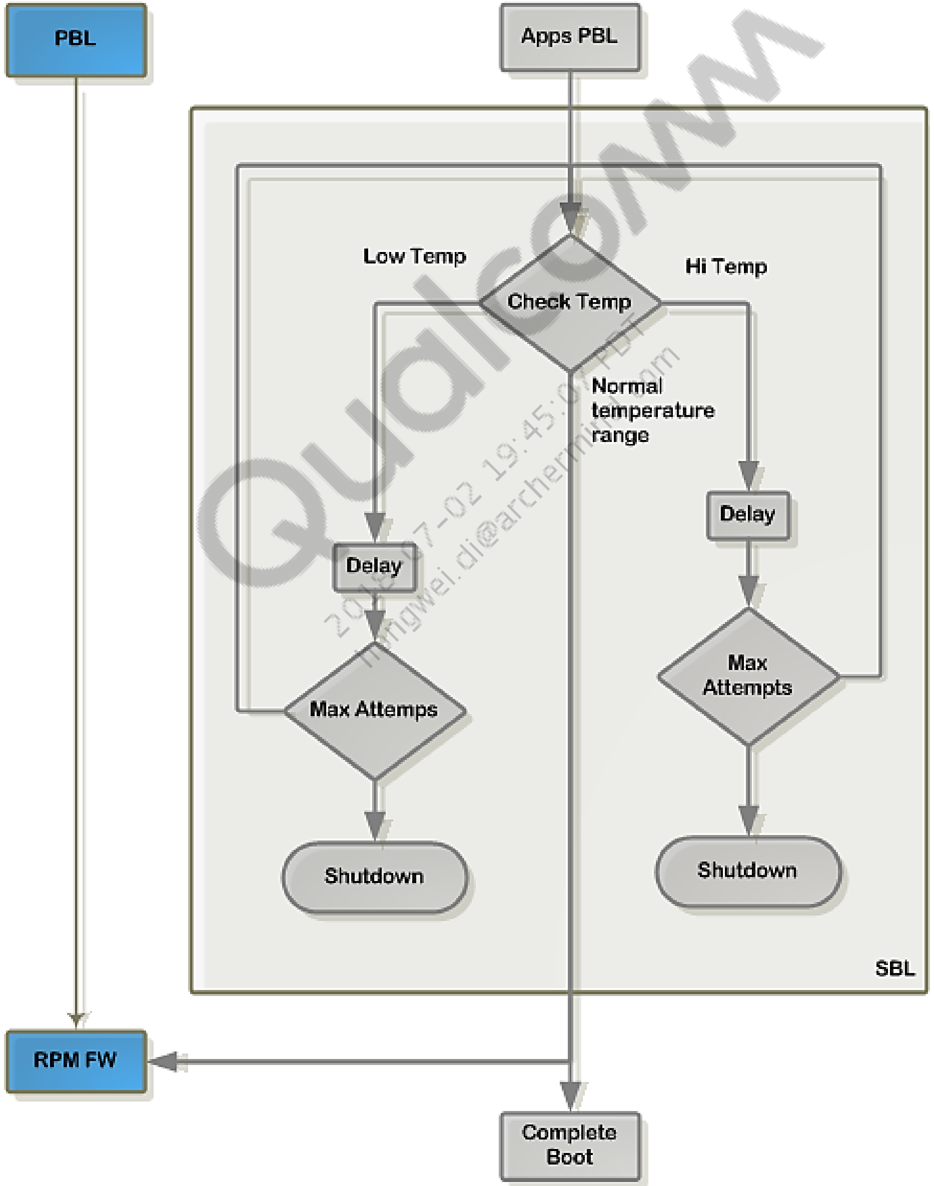
Overview

- Early boot mitigation:
  - Ensures that the temperature is in a valid operating range before allowing the device to boot.
  - Temperature thresholds, delays, and maximum attempts are configured in the boot loader build.
- Configure BTM – Hard coded
- nUpperThresholdDegC and nLowerThresholdDegC are set to maximum value of 150°C and minimum value of -150°C, respectively, by default and BTM is virtually disabled.
- If nUpperThresholdDegC and nLowerThresholdDegC are defined by OEMs with correct temperature threshold, boot can be deferred if the current temperature is higher than nUpper threshold or lower than the nLower threshold and retries; however, if predefined number of retries is exceeded, boot can fail and the system will shut down.

On boot_images/core/hwengines/tsens/config/89xx/BootTempCheckBsp.c

```
ConstBootTempCheckBspTypeBootTempChecksBsp[]={
 {
   /* .nUpperThesholdDegC*/150,
   /* .nLowerThresholdDegC*/-150
   }
  };
```

Overview (cont.)



Kernel Thermal Monitor (KTM)

Kernel bootup mitigation

- KTM is one of the kernel drivers that is initialized early to guarantee correct operation and to protect the device from thermal damage.
- After the driver is initialized, it starts polling for the temperature and mitigates the CPU (or other devices) when the temperature is above a certain threshold.
  - Mitigation or monitoring includes:
    - CPU frequency mitigation
    - CPU core control
    - Thermal reset
    - Vdd restriction
    - Vdd MX voting
  - KTM switches to the Interrupt mode late in the initialization phase. Before performing this transition to the Interrupt mode, the previous mitigations posed during bootup are removed.

Postbootup mitigation

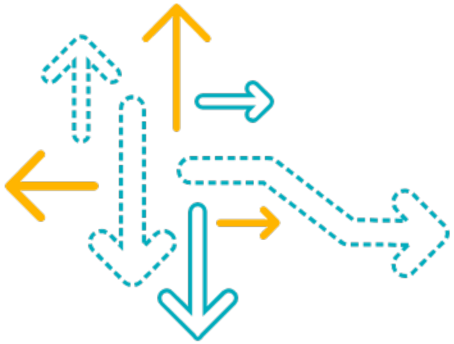
- After the system enters the late initialization phase, the KTM clears the current mitigation or monitor thread and switches to an Interrupt mode-type mitigation. During this phase, the thermal-sys framework is initialized and thermal uses this framework to set the threshold and receive notifications. This switch to the Interrupt mode does not depend on or wait for the thermal engine.
    - Mitigations or monitoring includes:
        - Emergency frequency mitigation
        - Emergency hotplug
        - Thermal reset
        - Vdd restriction
        - VDD MX voting
    - These KTM features permanently co-exist with the thermal engine for enhanced device protection.

KTM Configuration (Kernel Device Tree Example)

- The KTM algorithm is present in the /drivers/thermal/msm_thermal.c file and the associated parameters are defined in the arch/arm/boot/dts/msm8953.dtsi file.

KTM device tree example:

```
qcom,msm-thermal {
      compatible = "qcom,msm-thermal";
      qcom,sensor-id = <4>;
      qcom,poll-ms = <250>;
      qcom,limit-temp = <60>;
      qcom,temp-hysteresis = <10>;
      qcom,therm-reset-temp = <115>;
      qcom,freq-step = <2>;
      qcom,freq-control-mask = <0xff>;
      qcom,core-limit-temp = <80>;
      qcom,core-temp-hysteresis = <10>;
      qcom,core-control-mask = <0xfe>;
      qcom,hotplug-temp = <105>;
      qcom,hotplug-temp-hysteresis = <40>;
      qcom,cpu-sensors =    "tsens_tz_sensor4", "tsens_tz_sensor5",
                        "tsens_tz_sensor6", "tsens_tz_sensor7",
                        "tsens_tz_sensor9", "tsens_tz_sensor9",
                        "tsens_tz_sensor9", "tsens_tz_sensor9";
```



User Space Thermal Engine

Overview

- Thermal daemon was initially commercialized on the MSM8660 chipset and subsequently enhanced and reworked.
- The reworked thermal daemon enables integration with sensor manager and allows multiple algorithms.
  - Thermal engine: The thermal engine supports legacy and advanced dynamic algorithms running in parallel. OEMs will be able to choose the existing algorithm or the new algorithm in the thermal engine configuration file.
  - Dynamic algorithm exhibits significantly reduced tuning effort and improved average DMIPS.
  - Virtual sensor: A combination of more than two sensor readings with associated weights to accurately correlate skin temperature.
  - Dynamic parameter update: Important set of thermal engine configuration parameters can be updated at runtime for better OEM-specific dynamic thermal management.
- Two algorithm types are used in the thermal configuration file:
  - Monitor or threshold algorithm – algo_type monitor
  - Dynamic thermal management (DTM) – algo_type_ss

Threshold or Monitor Algorithm

- Threshold or monitor algorithm performs mitigation based on preconfigured set points. When a temperature threshold is reached, a management device is set to a predetermined performance level.
- OEM determines a series of temperature thresholds and a precise corresponding action per threshold.
- Monitor algorithm is used for LCD, modem, camcorder, and battery mitigation; not recommended for CPU and GPU mitigation, as it requires extensive tuning to find each set point.

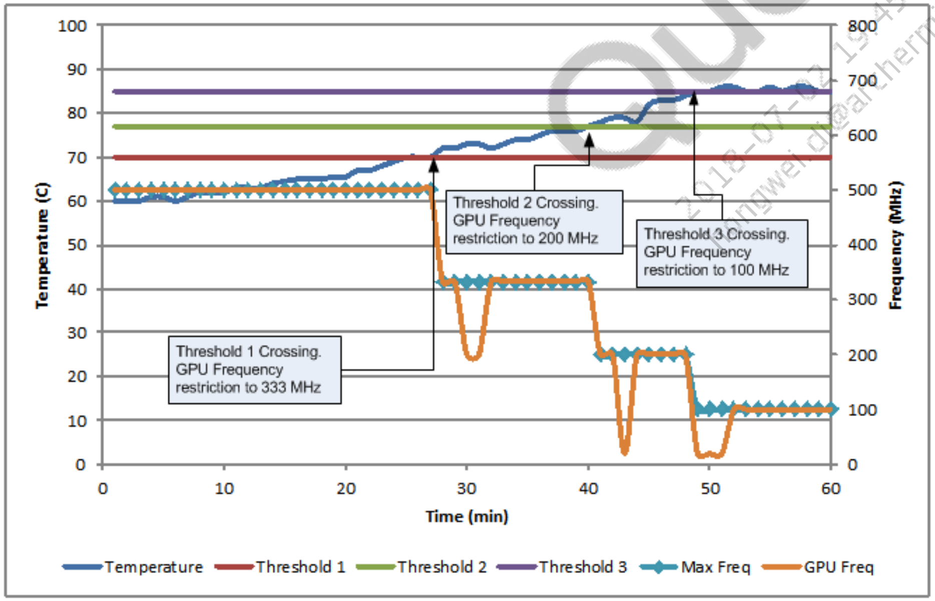
Threshold or monitor configuration example:

```
sampling         1000     : Default sampling period of 1000msec

[MSS_TM]                                      : Thermal Rule name
algo_type         monitor                     : Algorithm Type
sensor            tz_sensor_zone*/Thermistor  : Sensor Type: TSENS/Thermistor
sampling          1000                        : Sampling period of 1000msec
thresholds        95000     100000  105000    : thresholds set in C, 95C, 100C and 105C
thresholds_clr    90000     950000  100000    : clear points set 90C, 95C and 100C
actions          <device>  <device> <device>  : Mitigation Device Type
action_info       1             2        3    : Mitigation Levels
```

Note: * <device> Refer thermal read me file for types mitigation devices supported
/vendor/qcom/proprietary/thermal-engine/readme.txt

Threshold or Monitor Algorithm (cont.)

- Measured temperature crosses a predefined threshold and then sets a predefined mitigation level.
- When sensor temperature reaches 70°C, the GPU frequency is reduced from 500 MHz to 333 MHz.
- Final sensor temperature is maintained at 85°C.



Threshold 1 Crossing. GPU Frequency restriction to 333 MHz

Threshold 2 Crossing. GPU Frequency restriction to 200 MHz

Threshold 3 Crossing. GPU Frequency restriction to 100 MHz

Configuration parameters	Level 1	Level 2	Level 3
Threshold set	70°C	77°C	85°C
Threshold clear	65°C	72°C	80°C
GPU frequency	333 MHz	200 MHz	100 MHz

DTM

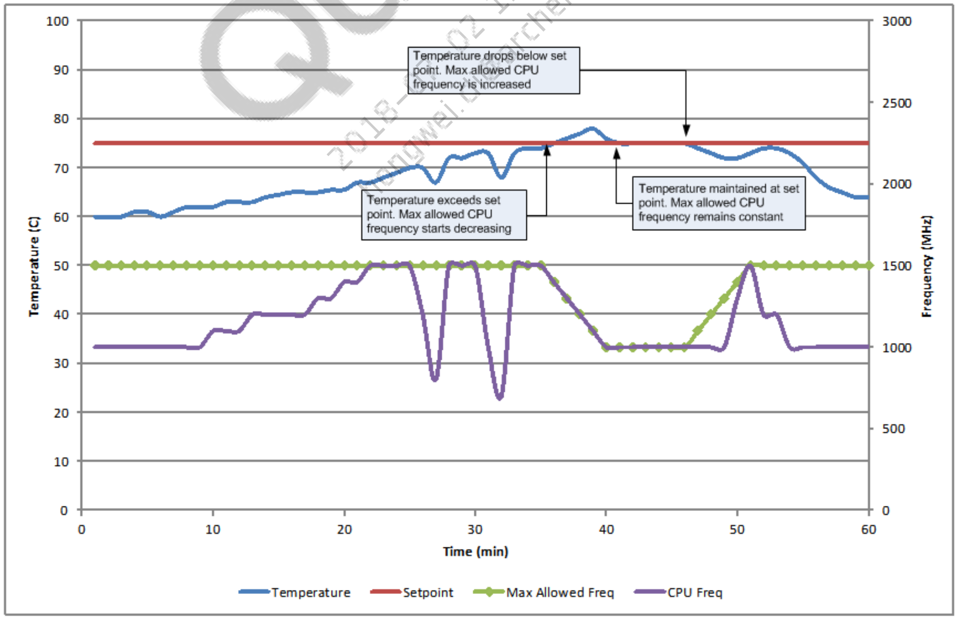
- DTM is recommended over monitor for CPU and GPU mitigation because it greatly decreases tuning effort, boosts performance, and more strictly maintains temperature to OEM set point.
- The dynamic algorithm adjusts performance based on the difference between a sensor measurement and a set point temperature.
- If the measured temperature is above the set point, the algorithm steps the maximum allowed frequency of the CPU and/or GPU down. The algorithm continues to monitor the temperature and adjust the frequency maximum down until the measured temperature is at or below the set point.

DTM configuration example:

```
[DTM_CONF]                                 : Thermal Rule name
algo_type    SS                            : Algorithm Type
sampling     65                            : Sampling period of 65 msec
sensor       tz_sensor_zone*/Thermistor    : Sensor Type: TSENS/Thermistor
device       cpu/gpu                       : Device(cpu/gpu)to be mitigated
set_point     95000                        : Thresholds set point in degree Celsius - 95˚C
set_point_clr 90000                        : Threshold clear set points – 90˚C
```

DTM (cont.)

- When the temperature crosses a set point (75°C), reduce performance until temperature stabilizes. The Polling mode is enabled based on the sampling parameter defined in the rule.
- By reducing performance while above the threshold and as the temperature falls below the setpoint (75°C), the maximum allowed frequency is allowed to ramp back up.
- If the temperature drops further below the setpoint clear (50°C), the Interrupt mode is re-enabled.
- This is used only for CPU and/or GPU control.



Temperature drops below set point. Max allowed CPU frequency is increased

Temperature exceeds set point. Max allowed CPU frequency starts decreasing

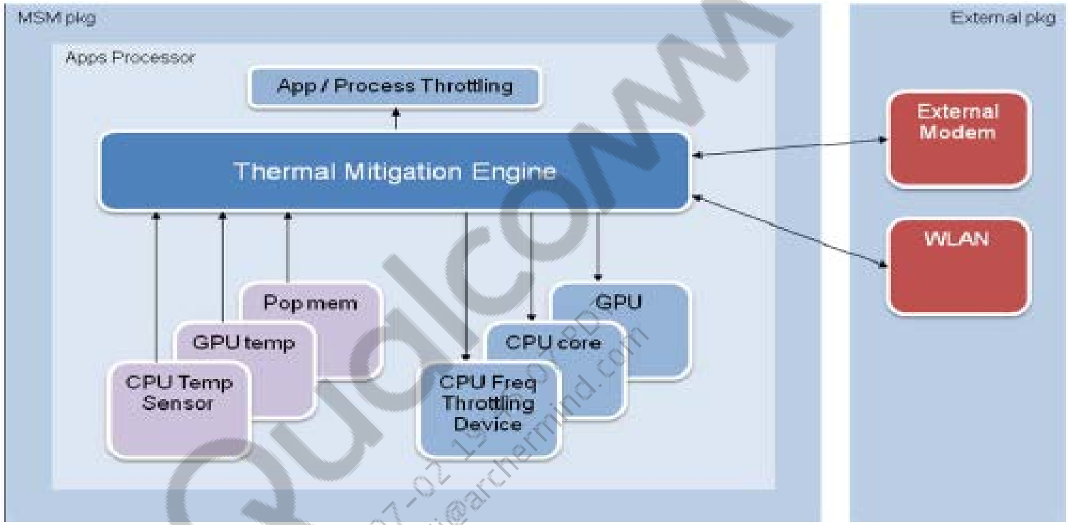
Temperature maintained at set point. Max allowed CPU frequency remains constant

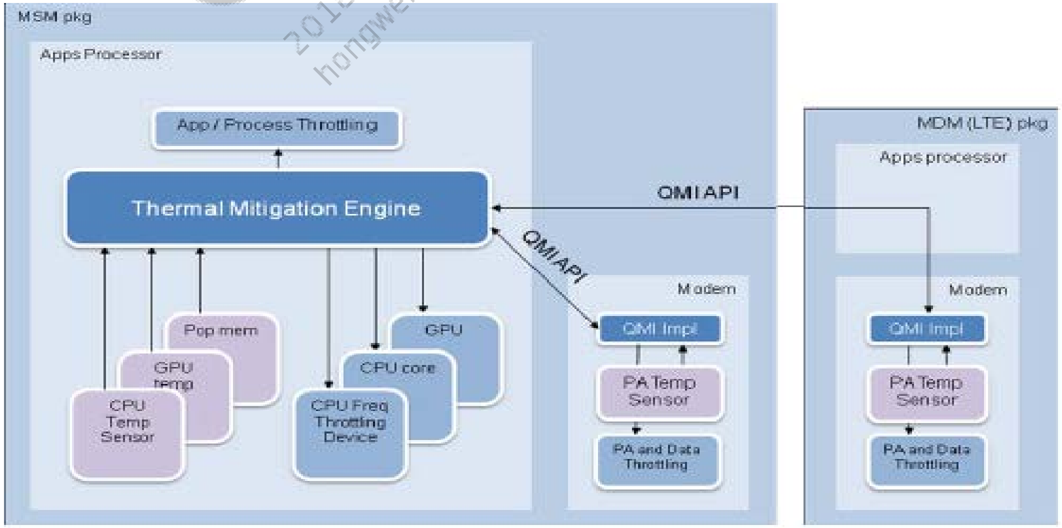
Modem Thermal Management

Overview

- Managed by the thermal engine
- Temperature is reported to the thermal engine by house keeping analog-to-digital converter (HKADC) via Qualcomm modem interface (QMI).
- The thermal engine receives temperature reading from thermistors and thermal zone sensors (TSENS)
- PA is the hottest component in modem-centric scenario.
- Actions and thresholds are defined in the thermal engine configuration file as follows:
  - Thermal-engine.conf – TSENS
  - Modem embedded file system (EFS) – PA
- Perform one of the following methods to control the temperature:
  - Preferred or first method – Keep the original power class and limit the uplink (UL) data throughput, while performing duty-cycling (DTx).
  - Not a preferred method; however, it is required as another tool to reduce the probability of reaching the Emergency state – Reduce the power class of the device and lower the maximum Tx power, to limit the power dissipation of the power amplifier
  - DL data traffic is controlled by user equipment (UE) acknowledgment (ACK) and/or no acknowledgment (NACK) packets to network (physical uplink common control channel (PUCCH) backoff)
  - Limit the downlink (DL) throughput by dropping carrier components (CC) and fallback to 2Rx from 4Rx. Emergency state – Call shutdown; device goes to limited service and allows only E911 calls

Modem Mitigation Legacy PA Thermal Mitigation Algorithms





Modem Mitigation Legacy PA Thermal Mitigation Algorithms (cont.)

- Legacy PA thermal mitigation algorithm is same as previous targets, which use modem as thermal action device.
- The legacy PA thermal mitigation algorithm supports four levels of thermal adjustment:
  - Level 0 – No restriction, full modem performance
  - Level 1 – Requests the modem to run the data throughput reduction algorithms
  - Level 2 – Maximum transmit power limit (MTPL) backoff and/or PUCCH backoff
  - Level 3 – Puts the modem into Limited Service mode, in which only emergency 911 calls are allowed

Sample configuration:

```
[pa_therm0]
  algo_type          monitor
  sensor             pa_therm0
  sampling           1000
  thresholds         75000       95000  105000
  thresholds_clr     65000       85000  100000
  actions            modem       modem  modem
  action_info        1           2          3
```

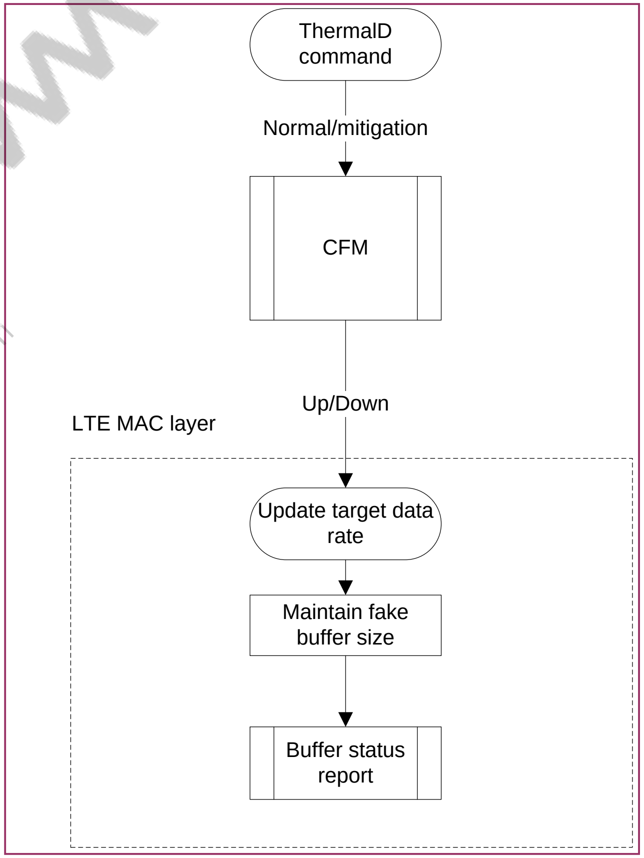
Note: Tx backoff and/or PUCCH backoff power backup is enabled by EFS configuration settings.

Modem Mitigation Legacy PA Thermal Mitigation Algorithms (cont.)

- Tabasco Modem mitigation level 2 supports DL throttling and PA power backoff throttling
- DL throttling is enabled by default:
  - If tm_mechanism is set to 00 and the EFS file tx_power_backoff is pushed, the MTPL backoff will kick-in along with PUCCH backoff
- Tx power or PUCCH backoff alone can be configured by pushing the EFS file:
  - If CR 1105869 is resolved (DL throttling at level 2 thermal mitigation)
- The tx_power_backoff and the tm_mechanism EFS files must be located at /nv/item_files/modem/lte/ML1/

Mitigation Level 1 (PA Sensor) – UL Data Throttle

- NV 65611 defines the flow-control target LTE data rates; these data rates are expressed in number of bytes per millisecond.
- NV 65676 step timer in seconds for changing the UL data rate states; the default value is 15 sec.
- With the centralized flow manager, the UE sends fake buffer status reports to the network based on the target rate; therefore, the network assigns lower grant based on the same.



Mitigation Level 1 (PA Sensor) – UL Data Throttle (cont.)

- The following is the default target MAC-level data rate configuration for UL data throttle in software:

```
Uint8 num_state= 10;
Uint8 default_state= 5;
Uint16 reserved = 0; /* keep this set to 0 */
 /* number of bytes per TTI (ms) */
Uint32 target_rate[0] = 6250 (50 Mb/s)
Uint32 target_rate[1] = 5000 (40 Mb/s)
Uint32 target_rate[2] = 3125 (25 Mb/s)
Uint32 target_rate[3] = 1250 (10 Mb/s)
Uint32 target_rate[4] = 625 (5 Mb/s)
Uint32 target_rate[5] = 125 (1 Mb/s)
Uint32 target_rate[6] = 63 (500 kb/s)
Uint32 target_rate[7] = 13 (100 kb/s)
Uint32 target_rate[8] = 6 (50 kb/s)
Uint32 target_rate[9] = 1 (10 kb/s)
```

Notes:

- Thermal management is active only when using a non-GCF SIM, that is, a SIM that is not programmed with MCC-MNC (1-1).
- For LTE flow control to work, network and/or network simulator must support dynamic scheduling, that is, scheduling based on buffer status reported by the UE.

Mitigation Level 1 (PA Sensor) – UL Data Throttle (cont.)

- To change or configure the data rate, use the following EFS structure and write num_state, default_state, target_rate[0]…target_rate[num_state-1]. The data rate is expressed in bytes per milliseconds:

```
Uint8 num_state = 10;
Uint8 default_state= 5;
Uint16 reserved = 0; /* keep this set to 0 */
 /* number of bytes per TTI (ms) */
Uint32 target_rate[0] = 6250 (50 Mb/s)  //0x186A ( in EFS write it as 6A180000 )
Uint32 target_rate[1] = 5000 (40 Mb/s)  //0x1388 ( in EFS write it as 88130000 )
Uint32 target_rate[2] = 3125 (25 Mb/s)
Uint32 target_rate[3] = 1250 (10 Mb/s)
Uint32 target_rate[4] = 625 (5 Mb/s)
Uint32 target_rate[5] = 125 (1 Mb/s)   //Default state as configured on EFS
Uint32 target_rate[6] = 63 (500 kb/s)
Uint32 target_rate[7] = 13 (100 kb/s)
Uint32 target_rate[8] = 6 (50 kb/s)
Uint32 target_rate[9] = 1 (10 kb/s)
```

- EFS filename – lte_fc_macul_target_rates
- EFS file location – /nv/item_files/modem/lte/common/
- Sample EFS content

```
6A180000 88130000 350C0000 E2040000
71020000 7D000000 3F000000 0D000000
06000000 01000000
```

Mitigation Level 1 (PA Sensor) – UL Data Throttle (cont.)

- To revert to the default configuration, use the QPST EFS Explorer to remove the /nv/item_files/modem/lte/common/lte_fc_macul_target_rates file.
- For flow control target data rates, the default value of the state is 10. The default rate is currently set to 1 Mbps. The minimum rate can be set to a lower value. However, the value must be set to meet the requirement for control channels and delay sensitive applications. The value must not be set to 0 as it shuts down the UL and causes the call to eventually drop.
- The default state is selected to guarantee a timely response that reduces the temperature. Set the default state to 5 in the default configuration. The data rate is initially reduced to 1 Mbps after the UL flow control is triggered.

```
Uint8 num_state = 10;
Uint8 default_state = 5;
Uint16 reserved = 0; /* keep this set to 0 */
 /* number of bytes per TTI (ms) */
Uint32 target_rate[0] = 6250 (50 Mb/s)
                  ::::
Uint32 target_rate[5] = 125 (1 Mb/s)
                  ::::
Uint32 target_rate[9] = 1 (10 kb/s)
```

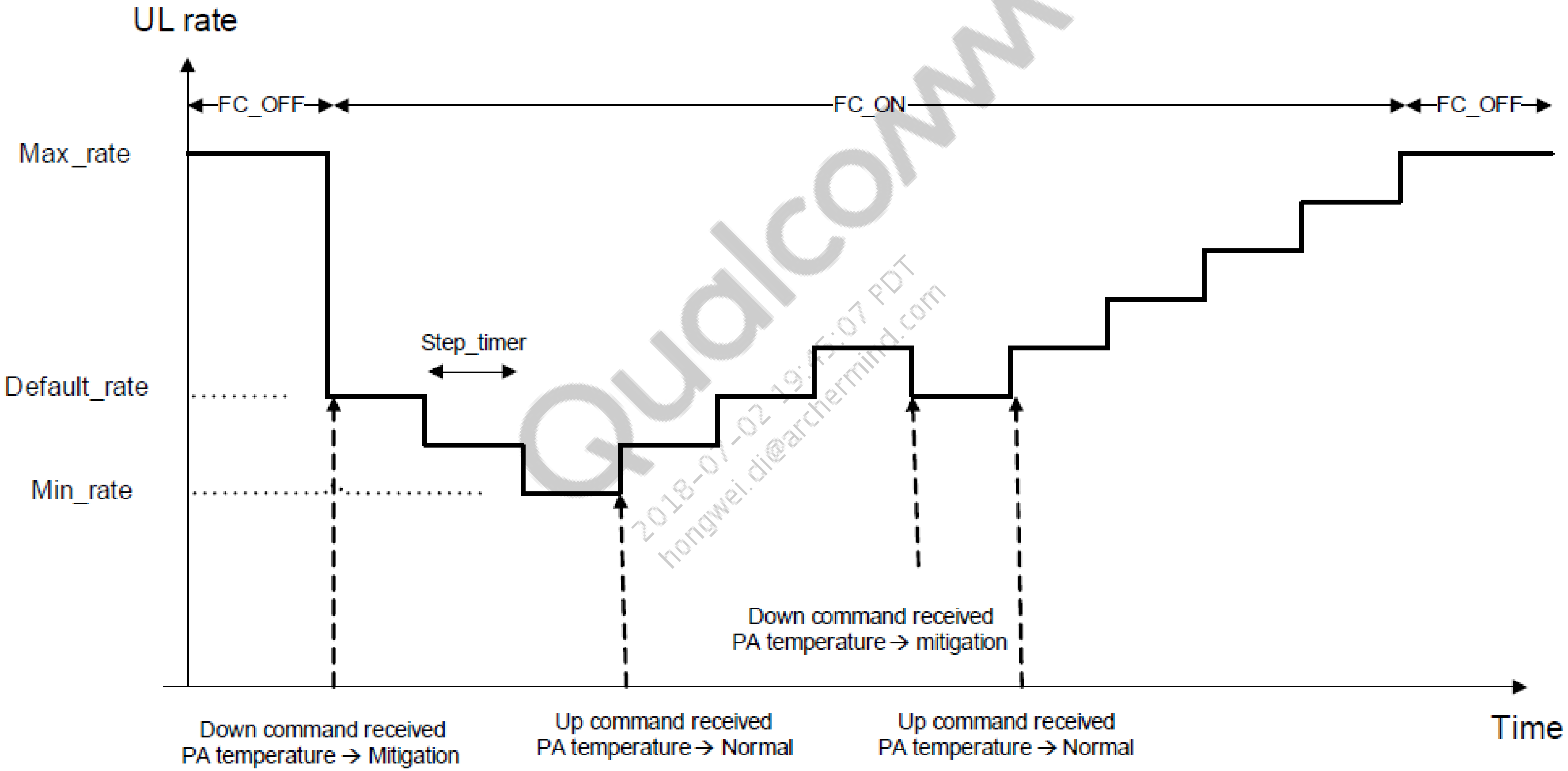
Mitigation Level 1 (PA Sensor) – UL Data Throttle (cont.)

- NV 65676 step timer is in seconds to change the rate states. The default value is 15 seconds.
- With a centralized flow manager, the UE sends fake buffer status reports to the network based on the target rate. Therefore, the network assigns lower grant based on the same.

Notes:

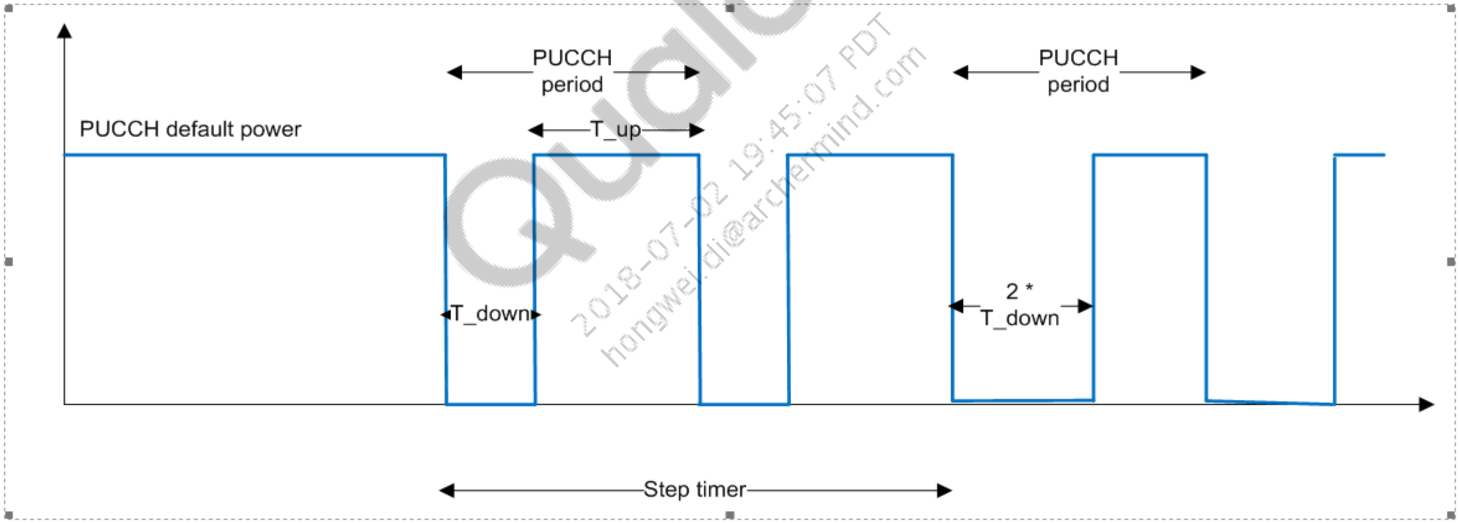
- For LTE flow control to work, network and/or network simulator must support dynamic scheduling, that is, scheduling based on the buffer status reported by the UE.
- NV 65611 cannot be used to configure the data rate at level 1, only the EFS method can be used.

Mitigation Level 1 (PA Sensor) – UL Data Throttle (cont.)

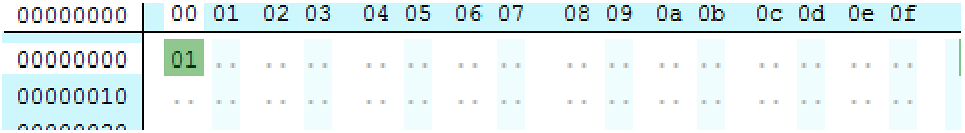


Mitigation Level 2 (PA Sensor) – PUCCH Backoff

- DL throttling is enabled by default
- Data traffic is controlled by UE ACK and/or NACK packets to network.
- During T_down, UE does not transmit any HARQ ACK and/or NACK on PUCCH.
- T_down + T_up is kept constant, called the PUCCH period. PUCCH duty cycle can be adjusted by changing T_down and T_up.



- To enable DL throttle (PUCCH backoff), the EFS hexadecimal file tm_mechanism (content: 01) must be present at /nv/item_files/modem/lte/ML1/.



```
00000000 | 00 01  02 03  04 05  06 07  08 09  0a 0b  0c 0d  0e 0f
00000000 | 01 ..  .. ..  .. ..  .. ..  .. ..  .. ..  .. ..  .. ..
00000010 | .. ..  .. ..  .. ..  .. ..  .. ..  .. ..  .. ..  .. ..
00000020 |
```

Mitigation Level 2 (PA Sensor) – PUCCH Backoff (cont.)

- Default PUCCH backoff settings

```
pucch_cancel_info->default_state_fc = 4;
/* Default state for Thermal mitigation */
pucch_cancel_info->default_state_tm = 4;
pucch_cancel_info->num_states = 6;
pucch_cancel_info->step_timer_fc = 400;

/* Step Timer for each state for thermal  mitigation */
pucch_cancel_info->step_timer_tm = 30000;

pucch_cancel_info->timer_info[0].t_off = 100; /* Off timer */
pucch_cancel_info->timer_info[0].t_on = 100; /* On timer */

pucch_cancel_info->timer_info[1].t_off = 80;
pucch_cancel_info->timer_info[1].t_on = 120;

pucch_cancel_info->timer_info[2].t_off = 60;
pucch_cancel_info->timer_info[2].t_on = 140;

pucch_cancel_info->timer_info[3].t_off = 40;
pucch_cancel_info->timer_info[3].t_on = 160;

pucch_cancel_info->timer_info[4].t_off = 20;
pucch_cancel_info->timer_info[4].t_on = 180;

pucch_cancel_info->timer_info[5].t_off = 10;
pucch_cancel_info->timer_info[5].t_on = 190;
```

Mitigation Level 2 (PA Sensor) – PUCCH Backoff (cont.)

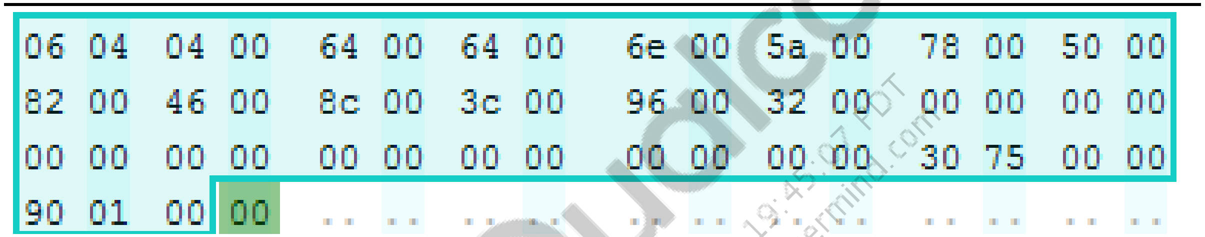
- To change or configure the PUCCH throttle information, use the following EFS structure and write num_states, default_state_tm, default_state_fc, padding, t_on, and t_off as follows:

Structure lte_ml1_nv_cfg_pucch_cancel_info_s

```
struct {
   /* Number of states */
   uint8 num_states;--------------------------------------- 6 //0x06

   /* Default state for Thermal mitigation */
   uint8 default_state_tm;--------------------------------- 4 //0x04

   /* Default state for CPU based Flow control */
   uint8 default_state_fc;--------------------------------- 4 //0x04

  /* Padding */
  uint8 padding;------------------------------------------- 0 //0x00

  struct {
     /* On timer */               ------------------------- 100 0x00 64 ( Write in EFS 64 00)
     uint16 t_on;

     /* Off timer */
     uint16 t_off;                -------------------------100 0x00 64   ( Write in EFS 64 00)

   }timer_info[LTE_ML1_NV_CFG_MAX_PUCCH_CANCEL_STATES];

     /* Step Timer for each state for thermal mitigation */
     uint32 step_timer_tm;---------------------------------3000 //0x00007530(Write in EFS 3075 0000)


     /* Step Timer for each state for CPU flow control */
     uint32 step_timer_fc;---------------------------------400 //0x0000 0190(Write in EFS 9001 0000)
}
```

Mitigation Level 2 (PA Sensor) – PUCCH Backoff (cont.)

- EFS file pucch_cancel
- EFS location: /nv/item_files/modem/lte/ML1/
- Sample EFS content:



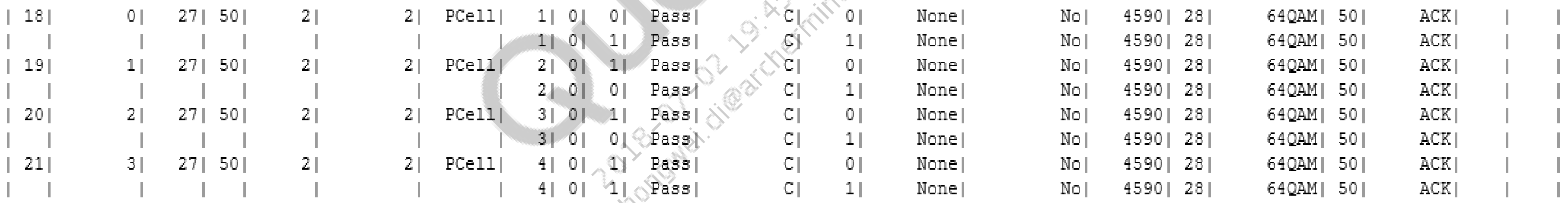
```
06 04  04 00   64 00  64 00   6e 00  5a 00   78 00  50 00
82 00  46 00   8c 00  3c 00   96 00  32 00   00 00  00 00
00 00  00 00   00 00  00 00   00 00  00 00   30 75  00 00
90 01  00 00   .. ..  .. ..   .. ..  .. ..   .. ..  .. ..
```

PUCCH Backoff Log

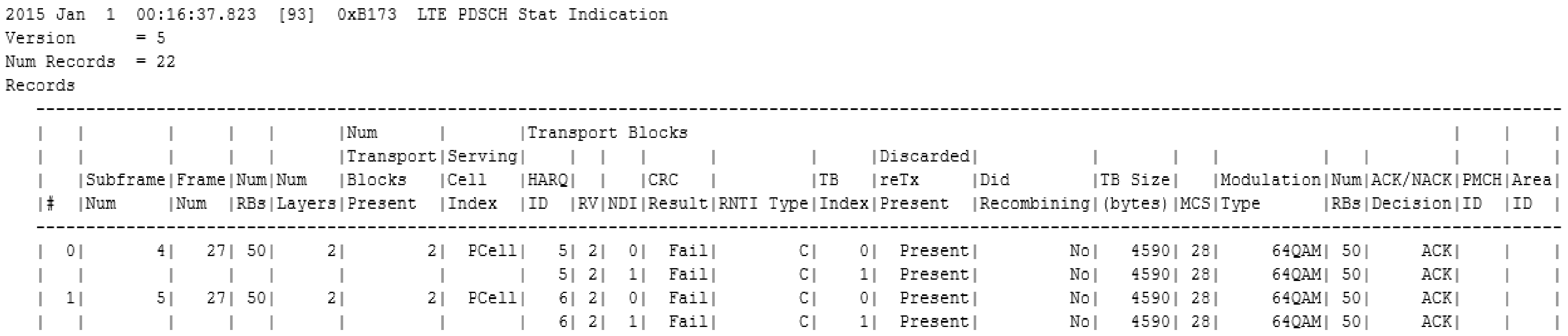
F3 Logs:

```
2015 Jan  1  00:16:37.791  lte_ml1_common_fc.c  1048   H  PUCCH Backoff down received.
2015 Jan  1  00:16:37.811  lte_ml1_common_fc.c  697    H  Off timer expiry. PUCCH
Backoff OFF
2015 Jan  1  00:16:37.841  lte_ml1_common_fc.c  1167   H  PUCCH FC OFF cmd received.
```

Log Packet: Ml1 intentionally punctures the ACK/NACK information to reduce the CPU usage
1980 Jan 6 00:27:54.827 [27] 0xB173 LTE PDSCH Stat Indication



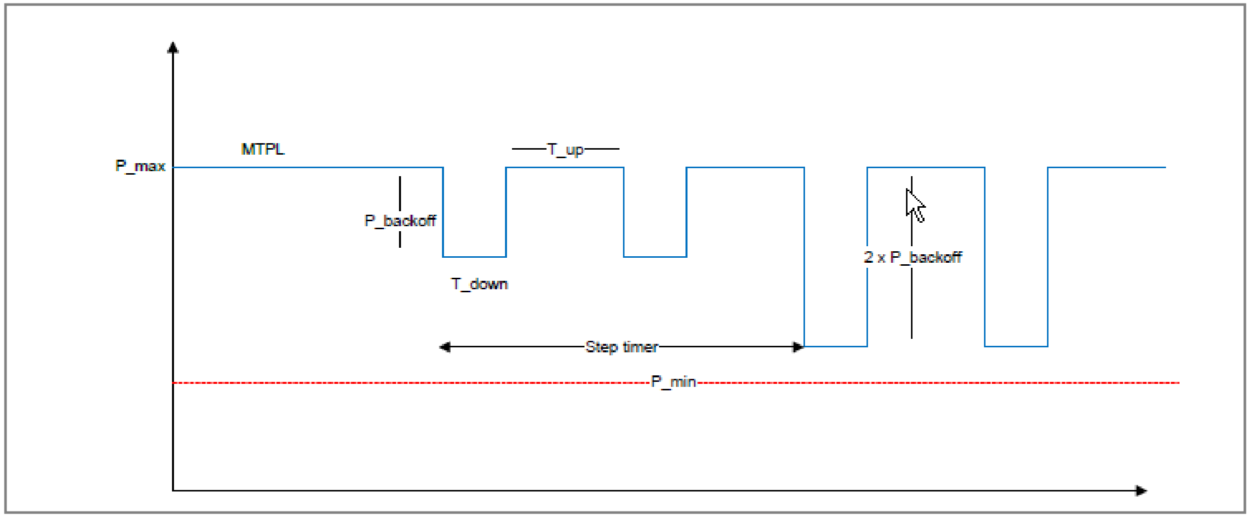
```
| 18|       0|   27| 50|     2|        2|  PCell|   1| 0|  0|  Pass|        C|    0|     None|         No|   4590| 28|     64QAM| 50|     ACK|    |    |
|   |        |     |   |      |         |       |   1| 0|  1|  Pass|        C|    1|     None|         No|   4590| 28|     64QAM| 50|     ACK|    |    |
| 19|       1|   27| 50|     2|        2|  PCell|   2| 0|  1|  Pass|        C|    0|     None|         No|   4590| 28|     64QAM| 50|     ACK|    |    |
|   |        |     |   |      |         |       |   2| 0|  0|  Pass|        C|    1|     None|         No|   4590| 28|     64QAM| 50|     ACK|    |    |
| 20|       2|   27| 50|     2|        2|  PCell|   3| 0|  1|  Pass|        C|    0|     None|         No|   4590| 28|     64QAM| 50|     ACK|    |    |
|   |        |     |   |      |         |       |   3| 0|  0|  Pass|        C|    1|     None|         No|   4590| 28|     64QAM| 50|     ACK|    |    |
| 21|       3|   27| 50|     2|        2|  PCell|   4| 0|  1|  Pass|        C|    0|     None|         No|   4590| 28|     64QAM| 50|     ACK|    |    |
|   |        |     |   |      |         |       |   4| 0|  1|  Pass|        C|    1|     None|         No|   4590| 28|     64QAM| 50|     ACK|    |    |
```



```
2015 Jan  1  00:16:37.823  [93]  0xB173  LTE PDSCH Stat Indication
Version      = 5
Num Records  = 22
Records
   --------------------------------------------------------------------------------------------------------------------------------------------------------
   |   |        |     |   |      |Num      |       |Transport Blocks                                                                            |    |    |
   |   |        |     |   |      |Transport|Serving|    |  |   |      |         |     |Discarded|           |       |   |          |   |        |    |    |
   |   |Subframe|Frame|Num|Num   |Blocks   |Cell   |HARQ|  |   |CRC   |         |TB   |reTx     |Did        |TB Size|   |Modulation|Num|ACK/NACK|PMCH|Area|
   |#  |Num     |Num  |RBs|Layers|Present  |Index  |ID  |RV|NDI|Result|RNTI Type|Index|Present  |Recombining|(bytes)|MCS|Type      |RBs|Decision|ID  |ID  |
   --------------------------------------------------------------------------------------------------------------------------------------------------------
   |  0|       4|   27| 50|     2|        2|  PCell|   5| 2|  0|  Fail|        C|    0|  Present|         No|   4590| 28|     64QAM| 50|     ACK|    |    |
   |   |        |     |   |      |         |       |   5| 2|  1|  Fail|        C|    1|  Present|         No|   4590| 28|     64QAM| 50|     ACK|    |    |
   |  1|       5|   27| 50|     2|        2|  PCell|   6| 2|  0|  Fail|        C|    0|  Present|         No|   4590| 28|     64QAM| 50|     ACK|    |    |
   |   |        |     |   |      |         |       |   6| 2|  1|  Fail|        C|    1|  Present|         No|   4590| 28|     64QAM| 50|     ACK|    |    |
```

Mitigation Level 2 (PA Sensor) – Tx Power Backoff

- DL throttling is enabled by default
  - If tm_mechanism is set to 00 and the EFS file tx_power_backoff is pushed, the MTPL backoff will kickin along with PUCCH backoff
- Tx power or PUCCH backoff alone can be configured by pushing EFS file
  - If CR 1105869 is resolved (DL throttling at level 2 thermal mitigation)
- The PA maximum transmit power is adjusted per the parameters configured in the EFS file (tx_power_backoff) located at /nv/item_files/modem/lte/ML1/.
- Values that can be configured in the .efs file are:
  - P_backoff – Initial value for Tx power backoff in dB (at each step n, the value of power backoff is n × P_backoff)
  - T_on – Length of time when the UE removes the limit on MTPL
  - T_off – Length of time when the UE reduces MTPL
  - Step_timer – Time spent in each step (see P_backoff)



Mitigation Level 2 (PA Sensor) – Tx Power Backoff (cont.)

- Structure of tx_power_backoffis located at /nv/item_files/modem/lte/ML1/

```
/* Initial backoff*/
uint16 p_backoff;
/* Maximum value of the backoff*/
uint16 p_backoff_max;
/* Time for non-backed-off value of power */
uint16 t_on;
/* Time for backed off Value of power */
uint16 t_off;
/* Timer for each step of the backoff*/
uint32 step_timer;
```

- Example
  - If the hexadecimal content of the file is 05000C00 32003200 983A0000, then the following values are set:
    - P_backoff – 5 dB (0500 for 5 dB)
    - Max_backoff – 12 dBm (0C00 for 13 dB)
    - T_on – 50 ms (3200 for 50 ms)
    - T_off – 50 ms (3200 for 50 ms)
    - Step_timer – 15 sec (983 A for 15 sec)

- Same default backoff values (5 dB, 10 dB, and 12 dB) are applied for each carrier for both intraband or interband UL CA.

Modem Thermal Management

- Introduction of new Silvers cores in MPSS.TA.X.Y all versions (Tabasco) modem:
  - Tabasco modem has two Qualcomm® Hexagon™ DSP processors for software and firmware operations(Q6A, Q6B with two Silver cores).
  - The two Silver co-processors in Tabasco perform vector and a minimal set of control firmware operations, like physical layer processing (LTE, WCDMA) and carrier aggregation.
  - Hexagon Silver core runs at the same frequency as Q6A core and it can be power collapsed independently while Hexagon is active.
  - Silver cores are power collapsed by default and come online when ever technical areas are requested.
- As the modem use cases consume more power due to increasing downlink and uplink data rates and increasing complexity of processing algorithms, Tabasco modem is introduced with two mitigation strategies to control Hexagon Silver core die temperature.
  - Modem thermal mitigation algorithm for LTE use cases
  - Modem thermal mitigation algorithm for WCDMA use cases
  - Legacy PA thermal mitigation algorithm

Modem Hexagon Silver Core Thermal Mitigation – LTE and/or WCDMA

- Modem Hexagon Silver core thermal mitigation is required for LTE and/or WCDMA dual carrier HSDPA use cases with VDD_MSS operating point of TURBO.
- The Silver thermal mitigation module runs on the application processor and sends a QMI indication to the LTE ML1/WCDMA (DC-HSDPA) modem client when thermal mitigation is required. The goal is to allow VDD_MSS to reduce from TURBO to NOM.
- The algorithms will be triggered when the temperature sensor (tsens1) located near the Hexagon Silver cores indicates that the junction temperature has crossed the configurable mitigation threshold.

Note: Modem proc mitigation threshold is based on Qualcomm® Reference Design (QRD) thermal profiling and is configured such that the impact performances of stand-alone LTE and/or WCDMA use cases are not impacted.

Modem Hexagon Silver Core Thermal Mitigation – LTE and/or WCDMA (cont.)

- LTE modem thermal mitigation
  - The Silver core thermal mitigation module runs on the Apps and will send a QMI indication to the LTE ML1 modem client when thermal mitigation is required.
  - For level 1, the secondary component carrier (SCC) is dropped by declaring vRLF, which causes the UE to report CQI = 0 to the eNodeB. This allows the VDD_MSS voltage to drop from TURBO and also reduces the Hexagon Silver processing load.
  - The second action defined is level 3 in case the call manager shuts down the LTE data call and allow emergency voice calls only.

Mitigation level	Action	Comment
Level 0	No mitigation	No thermal condition
Level 1	Declare vRLF on SCC	Tj enters level 1 region
Level 2	Not defined	Not defined
Level 3	LTE shutdown	Emergency calls only

Modem Hexagon Silver Core Thermal Mitigation – LTE and/or WCDMA (cont.)

- WCDMA modem thermal mitigation:

  The Silver core thermal mitigation module runs on the Apps and will send a QMI indication to the WCDMA dual carrier HSDPA modem client when thermal mitigation is required. For Tabasco, this corresponds to DC-HSDPA with Qualcomm interference cancellation and equalization (QICE) interference cancellation enabled.

  - If the junction temperature for this use case exceeds the level 1 threshold, then the QICE algorithm will be disabled regardless of how many interfering neighbor cells are being processed by QICE. The goal is to allow VDD_MSS to reduce from TURBO to NOM.
  - The second action defined is level 3 in case the call manager will shut down the WCDMA data call and allow emergency voice calls only.

Mitigation level	Action	Comment
Level 0	No mitigation	No thermal condition
Level 1	Disable QICE	Tj enters level 1 region
Level 2	Not defined	Not defined
Level 3	WCDMA shutdown	Emergency calls only

Modem Hexagon Silver Core Thermal Configuration

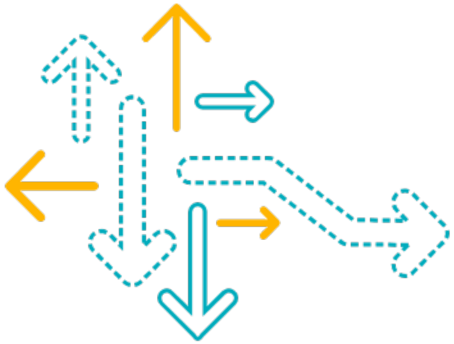
- Based on QRD thermal profiling, the Modem_proc mitigation threshold is configured such that the performance of stand-alone LTE and/or WCDMA use cases are not impacted.

Sample configuration:

MODEM_PROC_TEMP_MITIGATION]

```
algo_type monitor
sampling 1000
sensor tsens_tz_sensor1
thresholds 75000 80000
thresholds_clr 70000 75000
actions modem_proc modem_proc
action_info 1 3
```

Trigger	Mitigation level	Mitigation action LTE and/or WCDMA	Comment LTE and/or WCDMA
Modem temperature	0	No mitigation	–
–	1	Drop SCell by declaring vRLF and/or Disable QICE	VDD_MSS reduced to NOM from TURBO
–	2	Unused	–
–	3	Shutdown (emergency call only)	–



Thermal Configuration

Default Thermal Configuration

- QTI provides a default thermal configuration that is embedded in the device source code. This configuration must be tuned to meet the OEM’s unique requirements; refer to *Thermal Tuning Procedure* (80-N9649-1).
- View the default thermal configuration by using the ADB command `thermal-engine–o`.
  - This ADB command prints the existing thermal rules, including QTI defaults and custom OEM settings.
- The default configuration includes:
  - Rules for junction temperature management (85°C, by default)
    - Label examples – [SS-CPU0], [SS-CPU1], [SS-CPU2], [SS-CPU3], and [SS-CPU4-5-6-7]
    - These rules should not be increased from their default values
    - The following branches contain the source code of embedded rules:
      - /vendor/qcom/proprietary/thermal-engine/ss-data.c
      - /vendor/qcom/proprietary/thermal-engine/thermal_monitor-data.c
  - Rules for skin temperature management
    - Should be added by the OEM to thermal-engine.conf and pushed to /system/etc/ thermal-engine.conf
  - Other default rules should not be changed, for example, VDD_RSTR_MONITOR-TSENSX

Add Custom Thermal Configuration to Device

- Custom thermal configurations can be added to a device without recompiling the source code.
- Add a new rule by placing the new rule in the file named thermal-engine.conf and pushing it to the device (/system/etc/thermal-engine.conf) using ADB.

  ```
  adb push <location_of_thermal-engine.conf> /system/etc/thermal-engine.conf
  ```

- For example, to add a rule for GPU, place the following in thermal-engine.conf:

  ```
  [SS-GPU]
  algo_type ss
  sampling 65
  sensor tsens_tz_sensor12
  device gpu
  set_point 60000
  set_point_clr 57000
  time_constant 0
  ```

- The preceding example adds a rule named [SS-GPU] to the thermal configuration.
- Reboot the device after adding or changing thermal-engine.conf.

Add Custom Thermal Configuration to Device (cont.)

- Replace a default rule by adding a rule to thermal-engine.conf with the same name as the default rule.
- For example, [SS-POPMEM] is a default rule; if a rule with the same name is added to thermal-engine.conf, the default rule is overridden.

```
[SS-POPMEM]
algo_type ss
sampling 250
sensor pop_mem
device cluster1
set_point 65000
set_point_clr 55000
time_constant 2
```

- The preceding rule overrides the default [SS-POPMEM] rule of 80°C and lowers it to 65°C.
- Reboot the device after adding or changing thermal-engine.conf.

Add Custom Thermal Configuration to Device (cont.)

- Disable a default rule by adding a rule to thermal-engine.conf with the name of the rule to disable, followed by disable 1.

  ```
  [SS-POPMEM]
  disable 1
  ```

- The preceding example disables the [SS-POPMEM] rule.
- Reboot the device after adding or changing thermal-engine.conf.
- OEM’s can enlarge battery charging thermal mitigation from 0 mA to 3000 mA.
- /arch/arm/boot/dts/qcom/msm8953-qrd.dtsi

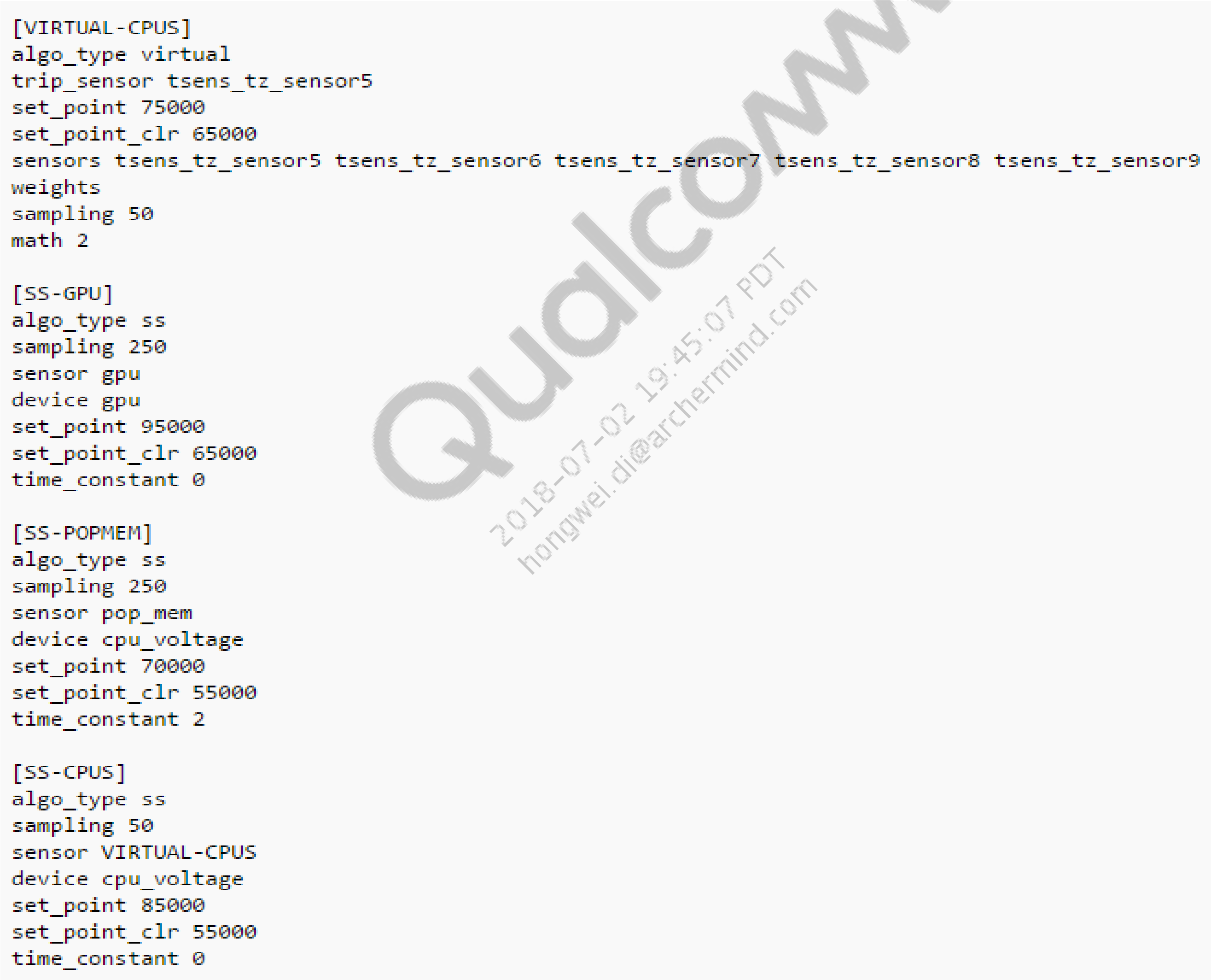
```
&pmi8950_charger {
         qcom,battery-data = <&qrd_batterydata>;
         qcom,float-voltage-mv = <4400>;
         qcom,chg-led-sw-controls;
         qcom,chg-led-support;
         qcom,external-typec;
         qcom,typec-psy-name = "typec";
         qcom,thermal-
mitigation = <3000 2500 2000 1500 1000 500 0>;
         status = "okay";
};
```

Sample config:

```
[BATTERY_CHARGING_CTL]
algo_type monitor
sampling 10000
sensor case_therm
thresholds 38000 40000 43000 48000
thresholds_clr 35000 38000 40000 43000
actions battery battery battery
action_info 2 3 4 5
```

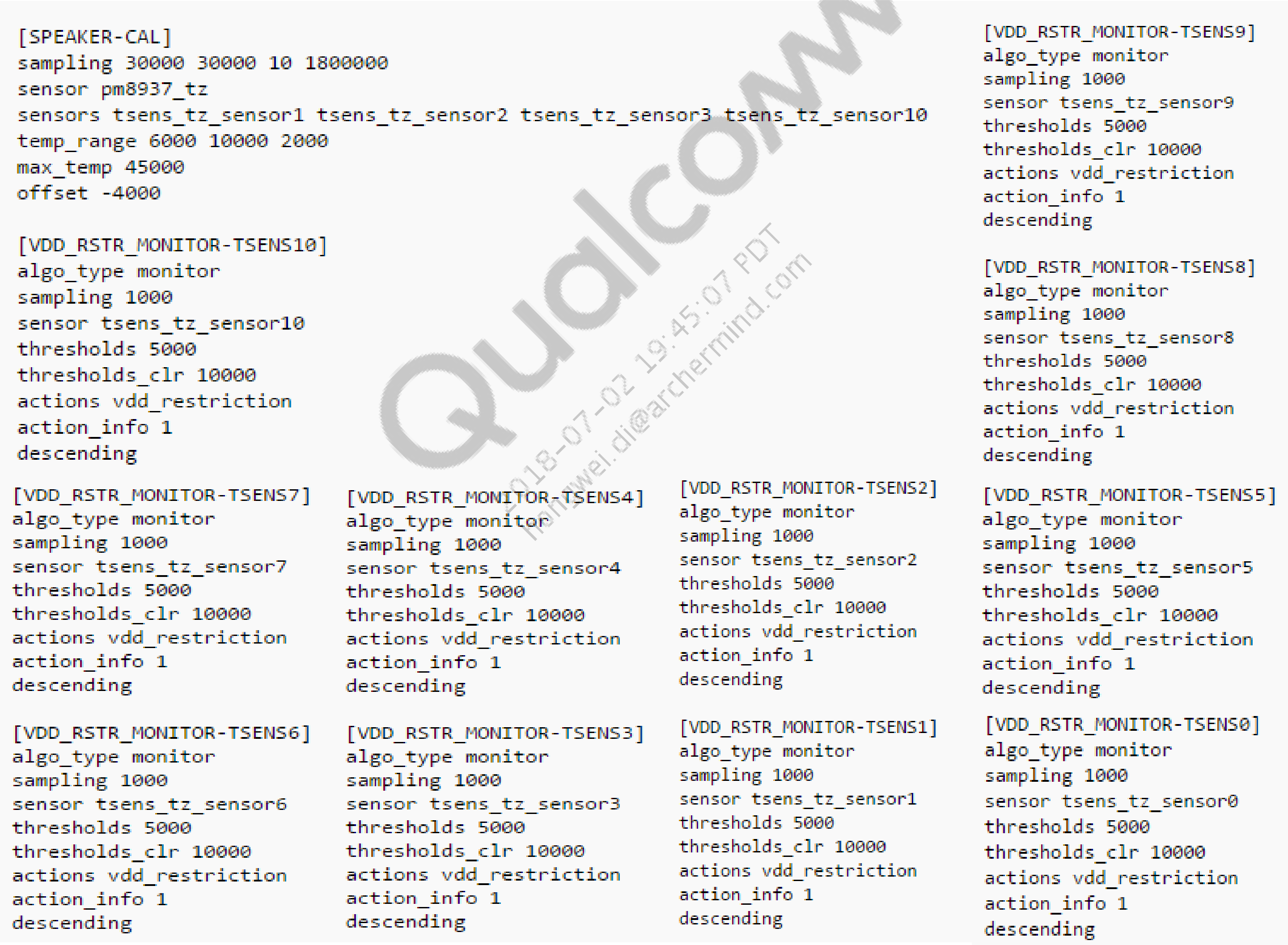
Note: OEMs can refer to the following links on kernel documentation for more information /kernel/Documentation/devicetree/bindings/power/smbxxxx-charger.txt or qpnp-xxxxxcharger.txt

MTP MSM8953 Default Thermal Configuration



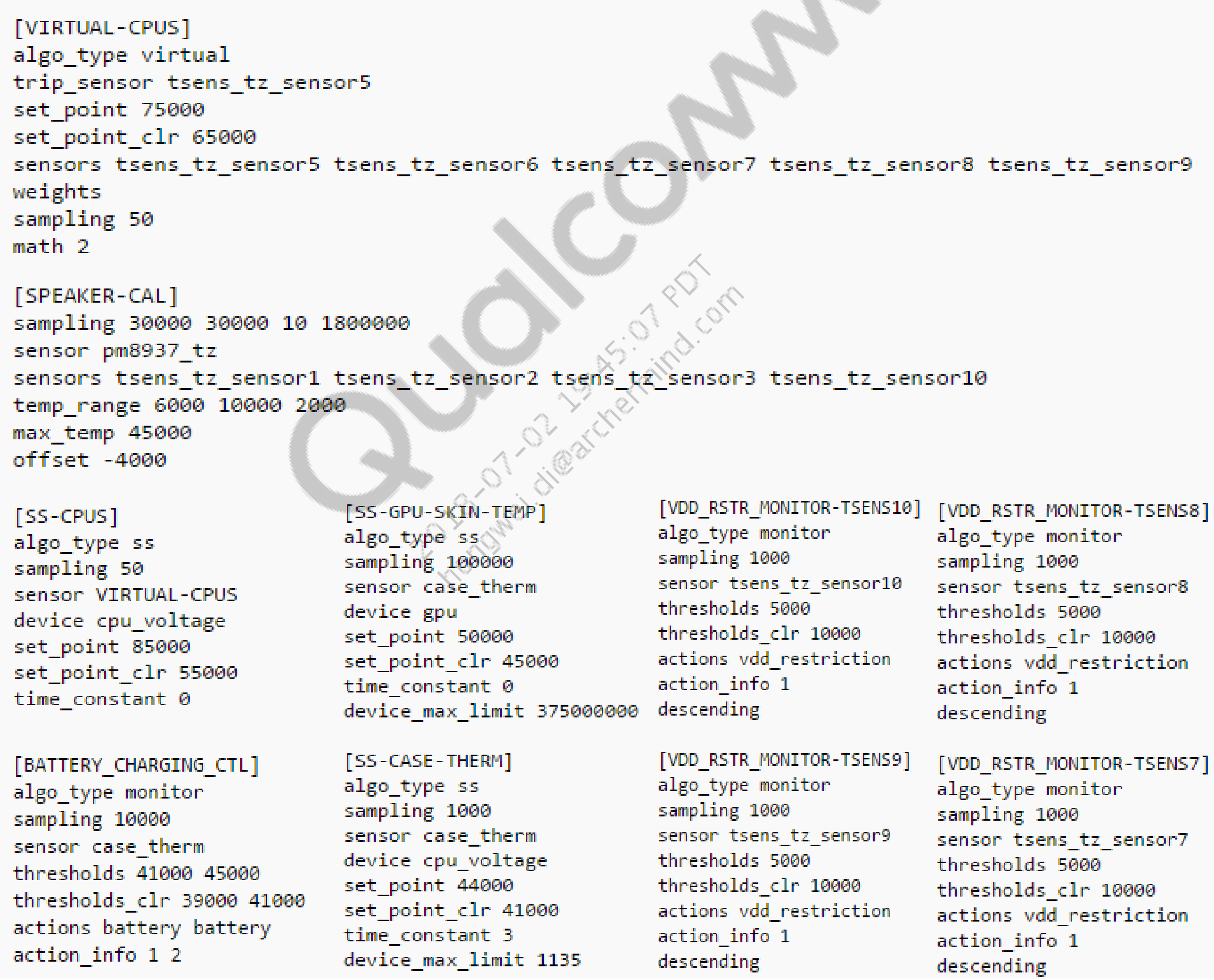
```
[VIRTUAL-CPUS]
algo_type virtual
trip_sensor tsens_tz_sensor5
set_point 75000
set_point_clr 65000
sensors tsens_tz_sensor5 tsens_tz_sensor6 tsens_tz_sensor7 tsens_tz_sensor8 tsens_tz_sensor9
weights
sampling 50
math 2

[SS-GPU]
algo_type ss
sampling 250
sensor gpu
device gpu
set_point 95000
set_point_clr 65000
time_constant 0

[SS-POPMEM]
algo_type ss
sampling 250
sensor pop_mem
device cpu_voltage
set_point 70000
set_point_clr 55000
time_constant 2

[SS-CPUS]
algo_type ss
sampling 50
sensor VIRTUAL-CPUS
device cpu_voltage
set_point 85000
set_point_clr 55000
time_constant 0
```

MTP Default Thermal Configuration



```
[SPEAKER-CAL]
sampling 30000 30000 10 1800000
sensor pm8937_tz
sensors tsens_tz_sensor1 tsens_tz_sensor2 tsens_tz_sensor3 tsens_tz_sensor10
temp_range 6000 10000 2000
max_temp 45000
offset -4000

[VDD_RSTR_MONITOR-TSENS10]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor10
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending

[VDD_RSTR_MONITOR-TSENS9]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor9
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending

[VDD_RSTR_MONITOR-TSENS8]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor8
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending

[VDD_RSTR_MONITOR-TSENS7]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor7
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending

[VDD_RSTR_MONITOR-TSENS6]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor6
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending

[VDD_RSTR_MONITOR-TSENS4]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor4
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending

[VDD_RSTR_MONITOR-TSENS3]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor3
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending

[VDD_RSTR_MONITOR-TSENS2]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor2
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending

[VDD_RSTR_MONITOR-TSENS1]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor1
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending

[VDD_RSTR_MONITOR-TSENS5]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor5
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending

[VDD_RSTR_MONITOR-TSENS0]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor0
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending
```

QRD8953 Thermal Configuration



```
[VIRTUAL-CPUS]
algo_type virtual
trip_sensor tsens_tz_sensor5
set_point 75000
set_point_clr 65000
sensors tsens_tz_sensor5 tsens_tz_sensor6 tsens_tz_sensor7 tsens_tz_sensor8 tsens_tz_sensor9
weights
sampling 50
math 2

[SPEAKER-CAL]
sampling 30000 30000 10 1800000
sensor pm8937_tz
sensors tsens_tz_sensor1 tsens_tz_sensor2 tsens_tz_sensor3 tsens_tz_sensor10
temp_range 6000 10000 2000
max_temp 45000
offset -4000
```

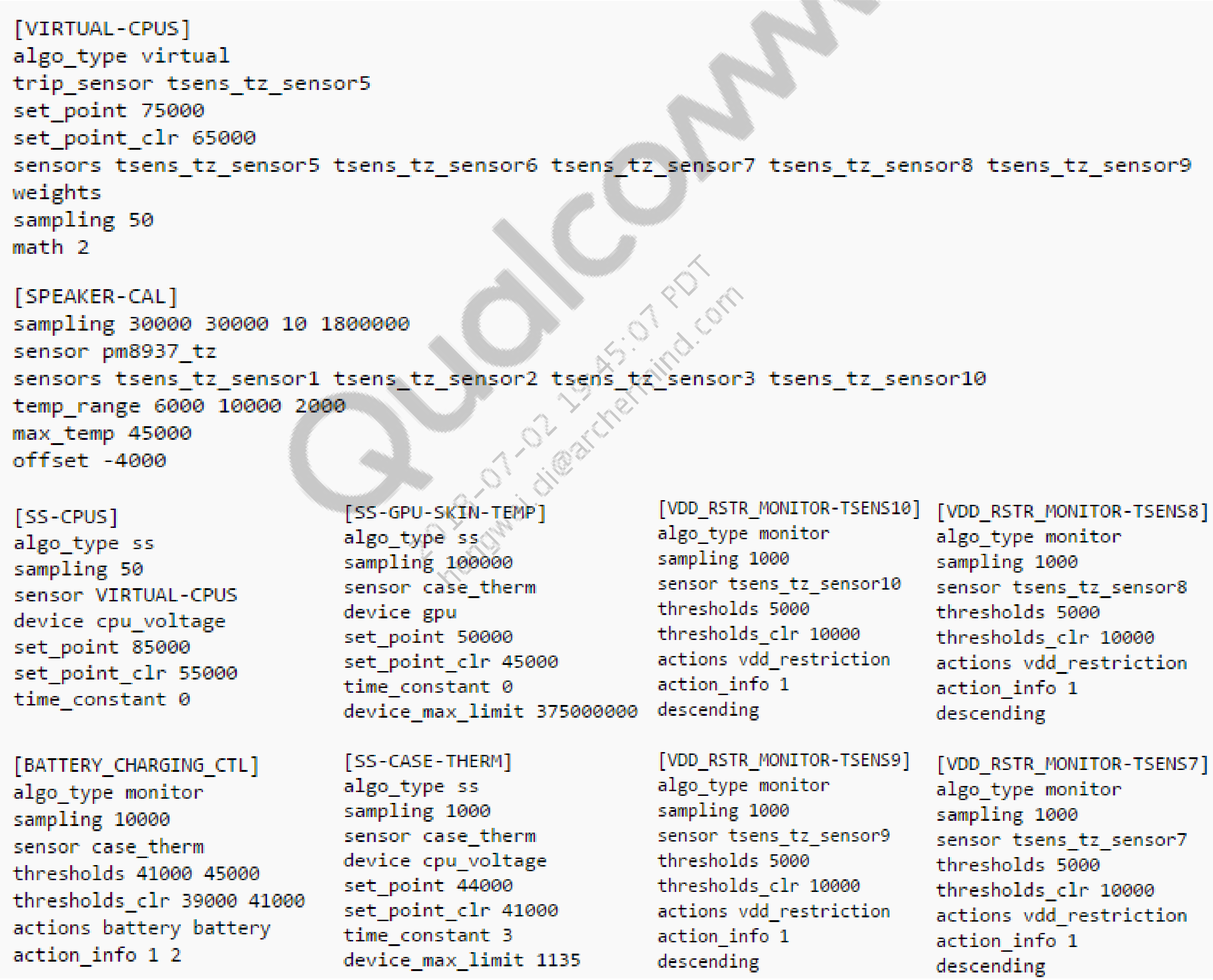
```
[SS-CPUS]
algo_type ss
sampling 50
sensor VIRTUAL-CPUS
device cpu_voltage
set_point 85000
set_point_clr 55000
time_constant 0

[BATTERY_CHARGING_CTL]
algo_type monitor
sampling 10000
sensor case_therm
thresholds 41000 45000
thresholds_clr 39000 41000
actions battery battery
action_info 1 2
```

```
[SS-GPU-SKIN-TEMP]
algo_type ss
sampling 100000
sensor case_therm
device gpu
set_point 50000
set_point_clr 45000
time_constant 0
device_max_limit 375000000

[SS-CASE-THERM]
algo_type ss
sampling 1000
sensor case_therm
device cpu_voltage
set_point 44000
set_point_clr 41000
time_constant 3
device_max_limit 1135
```

```
[VDD_RSTR_MONITOR-TSENS10]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor10
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending

[VDD_RSTR_MONITOR-TSENS9]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor9
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending
```

```
[VDD_RSTR_MONITOR-TSENS8]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor8
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending

[VDD_RSTR_MONITOR-TSENS7]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor7
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending
```

QRD8953 Thermal Configuration (cont.)



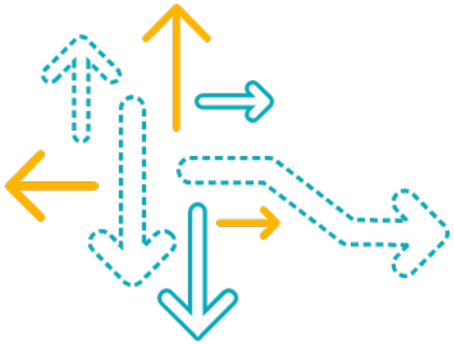
```
[VIRTUAL-CPUS]
algo_type virtual
trip_sensor tsens_tz_sensor5
set_point 75000
set_point_clr 65000
sensors tsens_tz_sensor5 tsens_tz_sensor6 tsens_tz_sensor7 tsens_tz_sensor8 tsens_tz_sensor9
weights
sampling 50
math 2

[SPEAKER-CAL]
sampling 30000 30000 10 1800000
sensor pm8937_tz
sensors tsens_tz_sensor1 tsens_tz_sensor2 tsens_tz_sensor3 tsens_tz_sensor10
temp_range 6000 10000 2000
max_temp 45000
offset -4000
```

```
[SS-CPUS]
algo_type ss
sampling 50
sensor VIRTUAL-CPUS
device cpu_voltage
set_point 85000
set_point_clr 55000
time_constant 0

[BATTERY_CHARGING_CTL]
algo_type monitor
sampling 10000
sensor case_therm
thresholds 41000 45000
thresholds_clr 39000 41000
actions battery battery
action_info 1 2
```

```
[SS-GPU-SKIN-TEMP]
algo_type ss
sampling 100000
sensor case_therm
device gpu
set_point 50000
set_point_clr 45000
time_constant 0
device_max_limit 375000000

[SS-CASE-THERM]
algo_type ss
sampling 1000
sensor case_therm
device cpu_voltage
set_point 44000
set_point_clr 41000
time_constant 3
device_max_limit 1135
```

```
[VDD_RSTR_MONITOR-TSENS10]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor10
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending

[VDD_RSTR_MONITOR-TSENS9]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor9
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending
```

```
[VDD_RSTR_MONITOR-TSENS8]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor8
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending

[VDD_RSTR_MONITOR-TSENS7]
algo_type monitor
sampling 1000
sensor tsens_tz_sensor7
thresholds 5000
thresholds_clr 10000
actions vdd_restriction
action_info 1
descending
```



New Thermal Features

BCL

- In some concurrent use case scenarios, there may be a high-voltage drop and excessive current is drawn from the battery by CPU cores and other peripheral loads such as camera flash, GSM PA, display, especially at a low battery voltage.
- The power management integrated circuit (PMIC) has a BCL hardware to prevent battery undervoltage lockout (UVLO) and overcurrent protection (OCP) situations.
    - When the PMIC input voltage drops below the operational value, the UVLO circuit turns off the power to protect the device.
    - When the battery current exceeds the operating threshold for too long, the OCP in the battery breaks the circuit and the phone is shut down abruptly.
- Abrupt phone shutdown does not provide a good user experience.
- The BCL software mechanism uses the PMIC fuel gauging hardware, to mitigate the current drawn from the battery by reducing the CPU load and ensuring that OCP and UVLO do not occur.

Note: Refer to *Battery Current Limit (BCL) Overview and Tuning* (80-NM328-709) for tuning and more details.

Case Thermistor to Control Device Skin Temperature

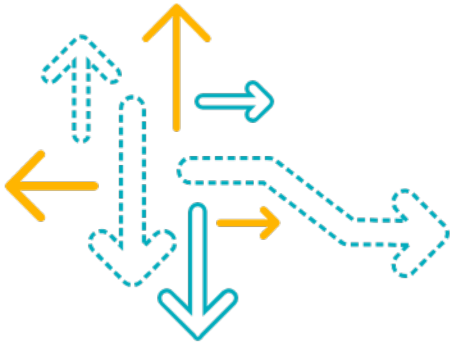
- A new ADC channel has been introduced to read external thermistor “case_therm” for skin temperature control.
- It is recommended that all OEMs include the case thermistor sensor in their design to improve tuning and maintain the device skin temperature within the specification.

Example code:

```
arch/arm/boot/dts/qcom/msm8953.dtsi
   qcom,sensor-information {
   sensor_information6: qcom,sensor-information@6 {
   qcom,sensor-type = "adc“; qcom,sensor-name = "case_therm";
    };
}

arch/arm/boot/dts/qcom/msm8952-qrd-skuc.dtsi
&pm8952_vadc {
   chan@13 {
   label = "case_therm";
   reg= <0x13>;
   qcom,decimation= <0>;
   qcom,pre-div-channel-scaling = <0>;
   qcom,calibration-type = "ratiometric";
   qcom,scale-function = <2>;
   qcom,hw-settle-time = <2>;
   qcom,fast-avg-setup = <0>;
   qcom,vadc-thermal-node;
   };
};
```

Multi Zone Temperature Control Engine (MTC)

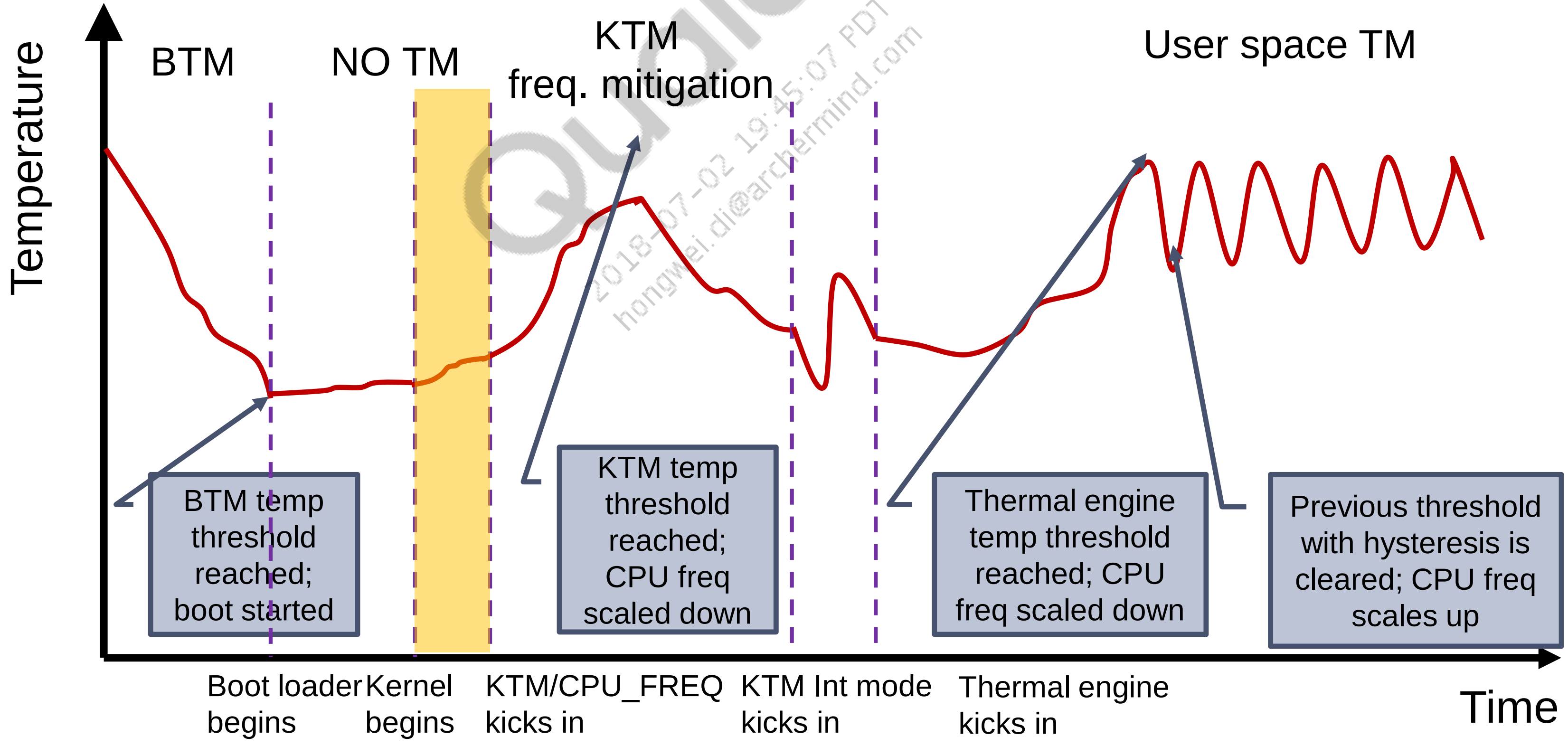
- Motivation
  - Thermal issues critical
  - Latency of thermal issue detection to mitigation is high:
    - Hardware tsense latency is improved 65 ms -> 4 ms
    - Software latency may be the bottleneck: average ~ 5 ms, worst case 30 ms
    - Thermal ramp: about 10^{0}C at 24 ms (75^{0}C to 85^{0}C)
- High latency requires large margins
- MTC hardware block (Tsens_wrapper) issues clock throttling commands to pulse swallower (RCGwTC).
- Software programmable throttling:
  - Throttling table (steps of 3%)
  - Thresholds
  - Disable or enable
- Software DCVS loop stays as is
- Advantages:
  - Simple but fast and effective
  - Generic solution for any subsystem
- MTC will be enabled only when junction temperature crosses 105^{0}C.

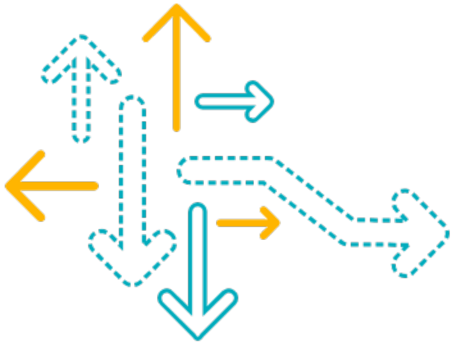


Thermal Management Mechanism

Overall Apps Processor Thermal Management Mechanism

- Three distinct TMs are provided:
  - SBL temperature check
  - Rich set of kernel thermal monitor
  - Full thermal engine with KTM postboot feature enabled

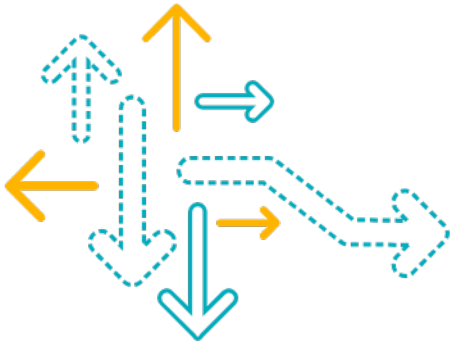

Temperature
BTM
NO TM
KTM
freq. mitigation
User space TM
BTM temp threshold reached; boot started
KTM temp threshold reached; CPU freq scaled down
Thermal engine temp threshold reached; CPU freq scaled down
Previous threshold with hysteresis is cleared; CPU freq scales up
Boot loader begins
Kernel begins
KTM/CPU_FREQ kicks in
KTM Int mode kicks in
Thermal engine kicks in
Time



Thermal Software Features and Management Devices

Overview

Feature	Description
CPU TM	Adjustment of maximum allowed operating frequency per cluster
GPU TM	Adjustment of maximum allowed operating frequency
Hotplug	Takes specific core offline
CPU core shutdown	Safety mechanism to ensure CPU cores shut off before junction temperature limits are exceeded
Modem TM	Adjustment of peak data rates, maximum Tx power, and data call termination
Camcorder TM	Adjustment of encoder frame rate or encoding shutoff
WLAN TM	Adjustment of peak data rates
LCD backlight TM	Adjustment of maximum backlight intensity
Battery charging TM	Adjustment of maximum allowable charge rate
BCL	CPU mitigation based on state-of-charge level of battery
Speaker coil calibration	Automatic calibration of speaker coil resistance vs. temperature enables audio codec to protect against speaker coil damage at high temperatures and high-power output
Voltage restriction	Voltage restriction enables low operating voltage above 0°C by adjusting the required minimum voltage at temperature extremes
KTM	Adjustment of maximum allowed operating frequency during kernel initialization and postboot device protection
Dynamic parameter update	Important parameter sets can be updated at runtime for better OEM-specific dynamic thermal management



Thermal Debug Overview

Thermal Engine Debug Overview

- To enable KTM logging

```
echo 8 > /proc/sys/kernel/printk
echo 'file msm_thermal.c +p' > /sys/kernel/debug/dynamic_debug/control
```

- Thermal engine provides detailed logging on Debug mode.
- To enable Debug mode:

1. Do one of the following
   - Keep “debug” in the first line of thermal-engine.conf and restart thermal-engine service (#thermal-engine &)
   - Start thermal engine in Debug mode manual by using the commands

     ```
     #stop thermal-engine (super user  mode)
     #start thermal-engine  -debug   &
     ```

   The following output is displayed in the command prompt window:

   ```
   adb   logcat –v  time –s ThermalEngine
   ```

2. Enabled thermal configuration on device for thermal mitigation:

   `adb shell  thermal-engine –o` (based on soc_id config is enabled,/sys/devices/soc0/sco_id)

3. Edit configuration file

   ```
   adb pull        /etc/thermal-engine.conf
   adb             remount
   <edit>
   adb push        thermal-engine.conf /etc/
   ```

ADB Commands

- To check the GPU frequencies

```
adb shell mkdir /sys/kernel/debug
adb shell mount -t debugfs none /sys/kernel/debug
adb shell cat /sys/kernel/debug/clk/grp_3d_clk/rate
```

- To check sensor zone type and sensor mapping

```
cat /sys/class/thermal/thermal_zone*/type
```

Example:

```
tsens_tz_sensor0
    :
tsens_tz_sensor4
pm8226_tz
pa_therm0
pa_therm1
```

- To check the sensor temperature

```
adb shell /sys/class/thermal/thermal_zone*/temp  *  zone number
```

- To check the CPU frequency

```
adb shell  cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq
adb shell cat  /sys/devices/system/cpu/cpu0/cpufreq/cpu_max_freq
adb shell cat  /sys/devices/system/cpu/cpu0/cpufreq/cpu_min_freq
```

Note: For more details on further debugging steps, refer to *Linux Android Software Thermal Debugging Guide* (80-NM998-1).

ADB Commands (cont.)

- To check the DDR frequency

```
adb shell cat /sys/kernel/debug/clk/bimc_clk/measure   x (2/1000000)
```

- To check SNoC

```
adb shell cat /sys/kernel/debug/clk/snoc_clk/measure
```

- To check PCNoC

```
adb shell cat /sys/kernel/debug/clk/pcnoc_clk/measure
```

- To check the MDP frequency

```
adb shell cat /sys/kernel/debug/clk/gcc_mdss_mdp_clk/measure
```

Key Points of Thermal Logcat

```
    --------- beginning of /dev/log/system
    --------- beginning of /dev/log/main
01-03 03:50:30.644 I/ThermalEngine(3086): Thermal daemon started
01-03 03:50:30.644 I/ThermalEngine(3086): Debug output enabled   →   if thermal-engine started in Debug mode
01-03 03:50:30.644 D/ThermalEngine(3086): Number of GPU cores 1
01-03 03:50:30.644 I/ThermalEngine(3086): Number of gpus:1
01-03 03:50:30.654 D/ThermalEngine(3086): Number of CPU cores 4
01-03 03:50:30.654 D/ThermalEngine(3086): Number of CPU cores 4
01-03 03:50:30.724 I/ThermalEngine(3086): Using target config file '/system/etc/thermal-engine-8226.conf‘ --thermal configuration file
01-03 03:50:30.724 D/ThermalEngine(3086): parse_tm_section: Parsing section CPU0-1_MONITOR sensor        cpu0-1
01-03 03:50:30.724 D/ThermalEngine(3086): sampling        1000
01-03 03:50:30.724 D/ThermalEngine(3086): thresholds      120000
01-03 03:50:30.724 D/ThermalEngine(3086): thresholds_clr 115000
01-03 03:50:30.724 D/ThermalEngine(3086): actions         shutdown
01-03 03:50:30.724 D/ThermalEngine(3086): action_info     5000
01-03 03:60:35.720 D/ThermalEngine(18336): handle_timer_sig: SS Id SS-POPMEM Read pop_mem 61000mC, Err 0mC, SampleCnt 1
01-03 03:60:35.725 D/ThermalEngine(18336): handle_timer_sig: SS Id SS-POPMEM, E0 0mC, E1 0mC
01-03 03:60:35.720 D/ThermalEngine(18336): settimer: Start timer 1.000(sec)
01-03 03:60:35.720 D/ThermalEngine(18336): algo_monitor: Wait for EV
01-03 03:60:35.720 I/ThermalEngine(348): ACTION: CPU - Setting CPU[0] to 998400
01-03 03:60:35.720 I/ThermalEngine(348): ACTION: CPU - Setting CPU[1] to 998400
01-03 03:60:35.720 I/ThermalEngine(348): ACTION: CPU - Setting CPU[2] to 998400
01-03 03:60:35.720 I/ThermalEngine(348): ACTION: CPU - Setting CPU[3] to 998400
01-03 21:30:01.948 I/ThermalEngine(286): hotplug_ktm_request: write out 2
01-03 21:30:01.958 I/ThermalEngine(286): ACTION: Hot-plugged OFF CPU[1]
01-03 21:30:01.958 E/ThermalEngine(286): TM Id HOTPLUG-CPU1 Sensor cpu1 Reading 105c
01-03 21:30:01.958 E/ThermalEngine(286): handle_thresh_sig: TM Id HOTPLUG-CPU1 Sensor cpu1 Temp 10500
01-03 21:30:01.958 E/ThermalEngine(286): TM Id 'HOTPLUG-CPU1' Sensor 'cpu1' - alarm raised 1 at 105.0°C
```

Pop_mem sensor is the preliminary CPU mitigation sensor. Most of the times, it shows 61ºC

Action item is taken when pop_mem threshold crosses the limit (60°C)

Hot-plugged CPU cores at 105c defined in `qcom,msm-thermal{ };`

Key Points of KTM Kernel Logs (Filtered by Thermal Key Word)

```
Line 162: [1, swapper/0][    1.254433] msm-thermal qcom,msm-thermal.16: msm_thermal:Failed reading node=/soc/qcom,msm-thermal, key=qcom,rpm-phase-resource-type err=-22. KTM continues
Line 163: [1, swapper/0][    1.254454] msm-thermal qcom,msm-thermal.16: msm_thermal:Failed reading node=/soc/qcom,msm-thermal, key=qcom,gfx-phase-warm-temp. err=-22. KTM continues
Line 164: [1, swapper/0][    1.254505] msm-thermal qcom,msm-thermal.16: probe_vdd_mx:Failed reading node=/soc/qcom,msm-thermal, key=qcom,mx-restriction-temp. KTM continues
```

Failed reading nodes due to incomplete driver initialization

```
Line 168: [1, swapper/0][    1.255635] msm_thermal:get_kernel_cluster_info CPU1 topology not initialized.
Line 424: [36, kworker/4:1][    2.265345] msm_thermal:do_cluster_freq_ctrl Limiting CPU0 max frequency to 1344000. Temp:60
Line 425: [36, kworker/4:1][    2.265359] msm_thermal:do_cluster_freq_ctrl Limiting CPU1 max frequency to 1344000. Temp:60
Line 426: [36, kworker/4:1][    2.265369] msm_thermal:do_cluster_freq_ctrl Limiting CPU2 max frequency to 1344000. Temp:60
Line 427: [36, kworker/4:1][    2.265379] msm_thermal:do_cluster_freq_ctrl Limiting CPU3 max frequency to 1344000. Temp:60
Line 428: [36, kworker/4:1][    2.265389] msm_thermal:do_cluster_freq_ctrl Limiting CPU4 max frequency to 533333. Temp:60
Line 429: [36, kworker/4:1][    2.265399] msm_thermal:do_cluster_freq_ctrl Limiting CPU5 max frequency to 533333. Temp:60
Line 430: [36, kworker/4:1][    2.265409] msm_thermal:do_cluster_freq_ctrl Limiting CPU6 max frequency to 533333. Temp:60
Line 431: [36, kworker/4:1][    2.265418] msm_thermal:do_cluster_freq_ctrl Limiting CPU7 max frequency to 533333. Temp:60
```

KTM Monitor mode CPU mitigation (threshold 60ºC, monitors only core0 sensor) and hot plug threshold 80ºC

```
Line 432: [36, kworker/4:1][    2.511507] msm_thermal:do_core_control Set Offline: CPU6 Temp: 81
Line 441: [36, kworker/4:1][    2.761533] msm_thermal:do_core_control Set Offline: CPU5 Temp: 81
```

Core hot plug log

```
Line 1129: [1, swapper/0][   19.833884] msm_thermal:interrupt_mode_init Interrupt mode init
```

KTM switches to Interrupt mode

```
Line 1131: [1, swapper/0][   19.852333] msm_thermal:disable_msm_thermal Max frequency reset for CPU0
Line 1133: [1, swapper/0][   19.869191] msm_thermal:disable_msm_thermal Max frequency reset for CPU1
Line 1135: [1, swapper/0][   19.889191] msm_thermal:disable_msm_thermal Max frequency reset for CPU2
Line 1136: [1, swapper/0][   19.897337] msm_thermal:disable_msm_thermal Max frequency reset for CPU3
Line 1138: [1, swapper/0][   19.917723] msm_thermal:disable_msm_thermal Max frequency reset for CPU4
Line 1140: [1, swapper/0][   19.936190] msm_thermal:disable_msm_thermal Max frequency reset for CPU5
Line 1141: [1, swapper/0][   19.946976] msm_thermal:disable_msm_thermal Max frequency reset for CPU6
Line 1143: [1, swapper/0][   19.973617] msm_thermal:disable_msm_thermal Max frequency reset for CPU7
```

KTM releases all mitigation and sets CPU max frequency while switching to interrupt mode and then monitors for hot plug 105ºC, core0 emergency frequency mitigation, Vdd restriction, and so on.

```
Line 1984: [5.954276 / 01-03 06:47:31.042] msm_thermal:set_enabled enabled = 0
```

KTM Handovers ctrl to thermal-engine

```
Line 2140: [1, swapper/0][   19.936190] msm_thermal:msm_thermal_bite TSENS:3 reached temperature:115. System reset
```

KTM triggers software reset, when any of the thermal zones hits 115ºC

KTM RAM Dump Debug overview

- When reset state is 0x1B, then the RAM dump is not useful because the caches are not flushed during reset.
- When reset state is 0x23 and if the last dmesg log says “msm_thermal:msm_thermal_bite TSENS:8 reached temperature:115. System reset”, then it is a watchdog bite triggered by KTM when tsens hits 115^{o}C.
- KTM polls the temperature and mitigates during boot until the late_init phase of kernel boot. After late initiation of the phase it hands over the temperature monitoring to thermal-engine (user space).
  - enabled == 0 (thermal-engine monitors temperature)
  - enabled == 1 (KTM monitors temperature)
- KTM has three kernel threads:
  - msm_thermal:hot-plug – Aggregates hot plug requests to bring the CPU cores offline or online
  - msm_thermal:freq_mitig – Aggregates the scaling maximum or minimum frequency requests and mitigates the CPU frequency
  - msm_thermal:therm_monitor – Performs watchdog bite

KTM RAM Dump Debug overview (cont.)

- ‘cpus’ variable has the thermal mitigation state for all the cores
  - cpus.cpu – Logical CPU ID
  - cpus.sensor_id – tsens monitors this particular core temperature
  - cpus.offline – If TRUE, KTM has requested this core to be offline
  - cpus.user_offline – If TRUE, user space (thermal-engine) has requested this core to be offline
  - cpus.hotplug_thresh_clear – If TRUE, emergency hot-plug threshold for this core is triggered in the hardware but not yet handled in KTM
  - cpus.user_max_freq – Holds the scaling maximum frequency requested by thermal engine
  - cpus.user_min_freq – Holds the scaling minimum frequency requested by thermal engine
  - cpus.max_freq – If TRUE, KTM has a request to cap the scaling maximum frequency.
  - cpus.limited_max_freq – Holds the last successful scaling maximum frequency requested by KTM
  - cpus.limited_min_freq – Holds the last successful scaling minimum frequency requested by KTM
  - cpus.freq_thresh_clear – If TRUE, the emergency frequency mitigation threshold for this core is triggered in the hardware but not yet handled in KTM.

KTM RAM Dump Debug overview (cont.)

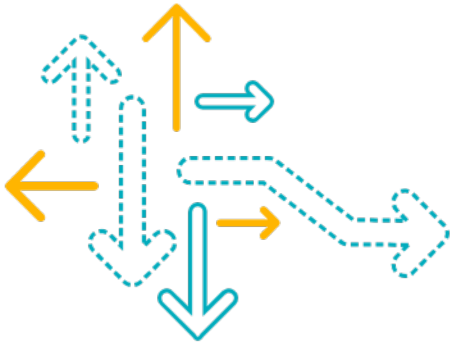
- When user_max_freq is high (for example, 4294967295) then the thermal engine does not place any request for this core. It applies for user_min_freq, if the value is 0.
- When user_max(min)_freq and limited_max(min)_freq are not same, then KTM is in the process of applying a new request or the “freq_mitig” kthread that is waiting on a mutex or blocked.
- ‘cpus_offlined’ variable is a bitmask for the current thermal hot plug request for CPUs. If the value of bit 1 is:
  - 1 – The thermal has a hot plug request for this core.
  - 0 – The thermal has no hot plug request for this core.
- When the request in cpus_offlined does not match the ‘cpus.offline’ and ‘cpus.user_offline’, then the hot-plug thread waits for a mutex or blocked.
- When (cpus.threshold.trip == THERMAL_TRIP_CONFIGURABLE_HI && cpus.threshold.active == 1) then the emergency threshold for this core is not reached. Hence, the temperature of this core is lesser than the emergency threshold (cpus.threshold.temp)

References

Title	Number
Qualcomm Technologies, Inc.	
Thermal Design Checklist	80-VU794-21
Design for Thermal: Key Requirements Why, What, Where, When, and How	80-VU794-24
Thermal Protection Algorithm Overview	80-VT344-1
MSM8974 Thermal Mitigation Algorithm	80-N8633-6
Thermal Tuning Procedure	80-N9649-1
Skin Temperature Measurement Procedure Using IR Camera	80-VU794-15
Linux Android Software Thermal Debugging Guide	80-NM998-1
Coefficient of Thermal Spreading (CTS) - Figure of Merit for Mobile Thermal Management	80-VU794-14
Mobile Devices Hardware Thermal Management	80-VU794-16
Core Control Feature	80-P0106-1
Battery Current Limit (BCL) Overview and Tuning	80-NM328-709

References (cont.)

Acronym or term	Definition
ACK	Acknowledgment
BCL	Battery current limiting
BTM	Boot thermal management
CC	Carrier components
DL	Downlink
DTM	Dynamic thermal management
KTM	Kernel thermal monitor
MTPL	Maximum transmit power limit
NACK	No acknowledgment
OCP	Overcurrent protection
PMIC	Power management integrated circuit
PUCCH	Physical uplink common control channel
QICE	Qualcomm interference cancellation and equalization
SCC	Secondary component carrier
UE	User equipment
UL	Uplink
UVLO	Undervoltage lockout



Questions?

https://createpoint.qti.qualcomm.com