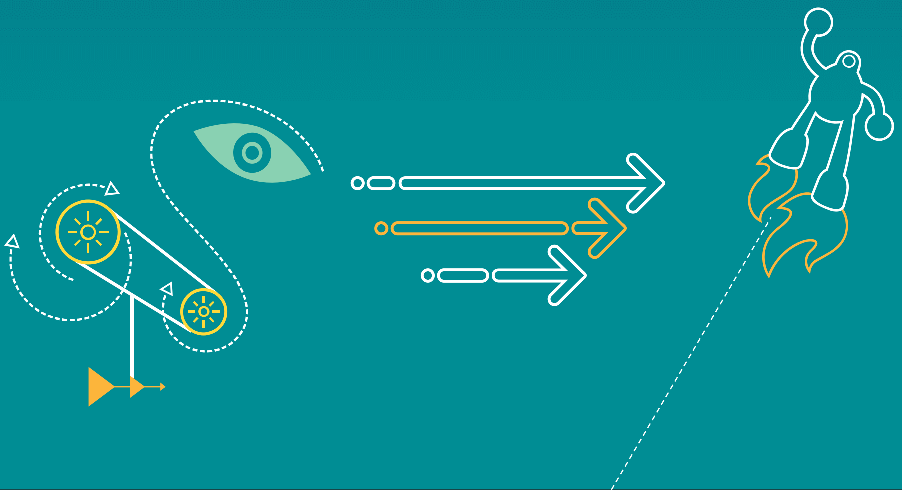
Linux Android PMIC Software Drivers Overview

Qualcomm Technologies, Inc.



80-P2485-2 C

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

Revision History

Revision	Date	Description
A	October 2015	Initial release
B	June 2016	Numerous updates were made to this document to include PMi8940, and MSM8917/MSM8940 information; to be read in its entirety.
C	November 2016	Updated slides 6, 9, and 17 with the MSM8920 details.

Contents

- Overview
- SPMI and Interrupts
- Regulator
- GPIO/MPP
- ADC
- Charger
- Fuel Gauge
- PWM
- LED
- References
- Questions?

PM8937/PMi8937/PMi8940 Overview

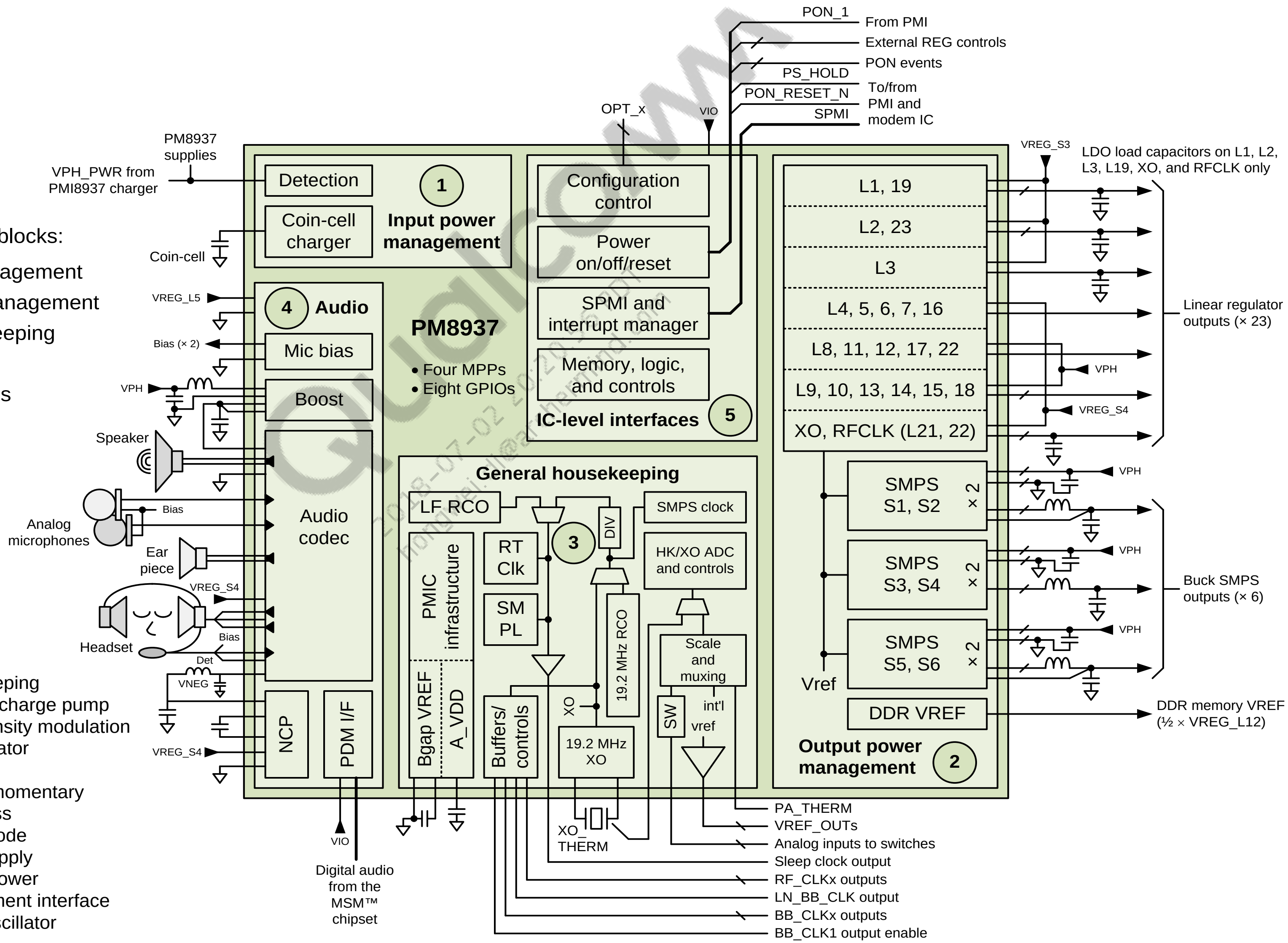
Introduction

This presentation describes the Linux Power Management Integrated Circuit (PMIC) (with Real Time Clock (RTC)) driver functionality in the MSM8937/MSM8917/MSM8940/MSM8920 application processors.

PM8937 High-Level Block Diagram

Five major functional blocks:

1) Input power management
2) Output power management
3) General housekeeping
4) Audio
5) IC-level interfaces

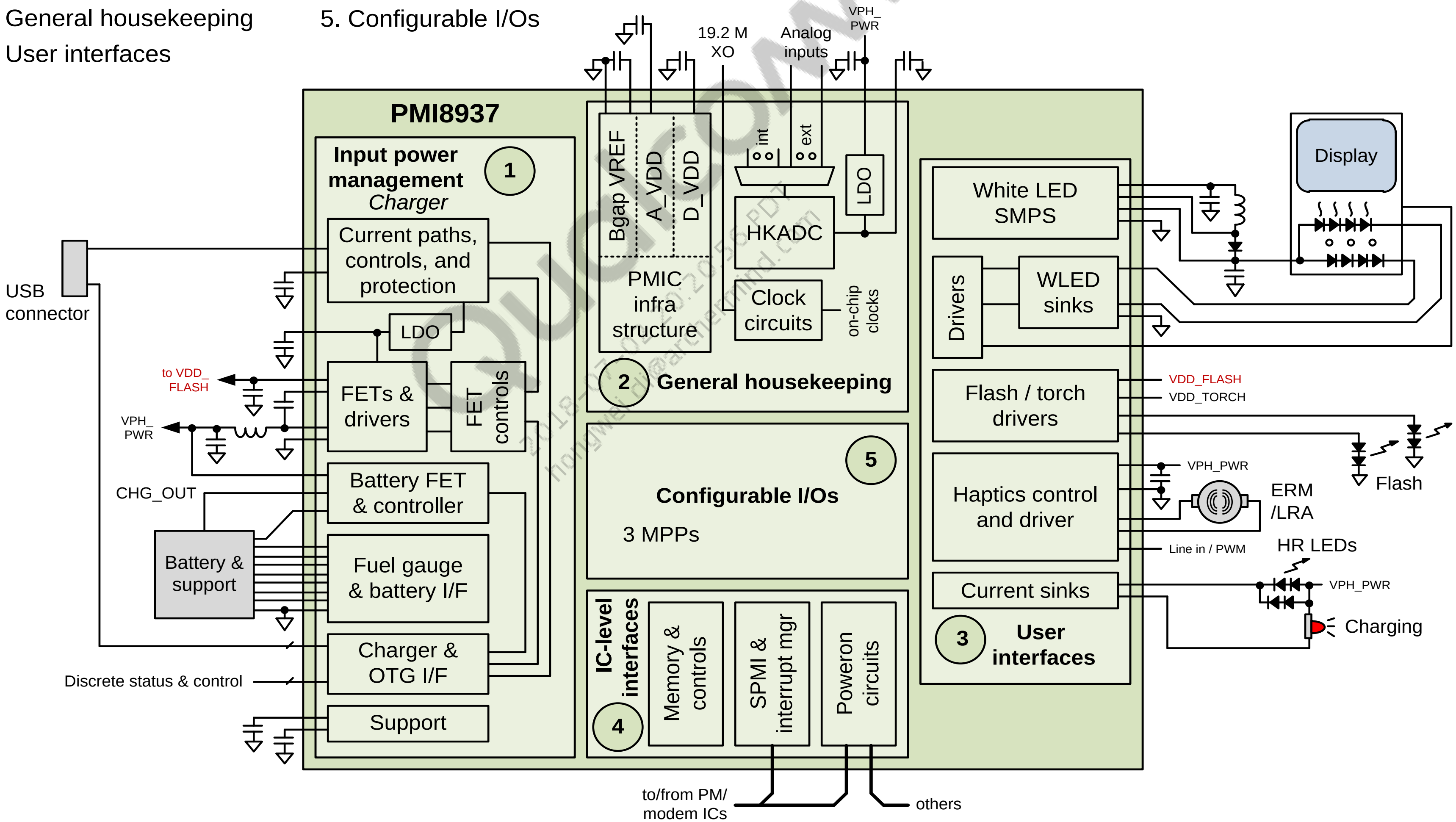


HK = housekeeping
NCP = negative charge pump
PDM = pulse density modulation
RCO = RC oscillator
RT = real-time
SMPL = sudden momentary power loss
SMPS = switch mode power supply
SPMI = system power management interface
XO = crystal oscillator

PMi8937 High-Level Block Diagram

Five major functional blocks:

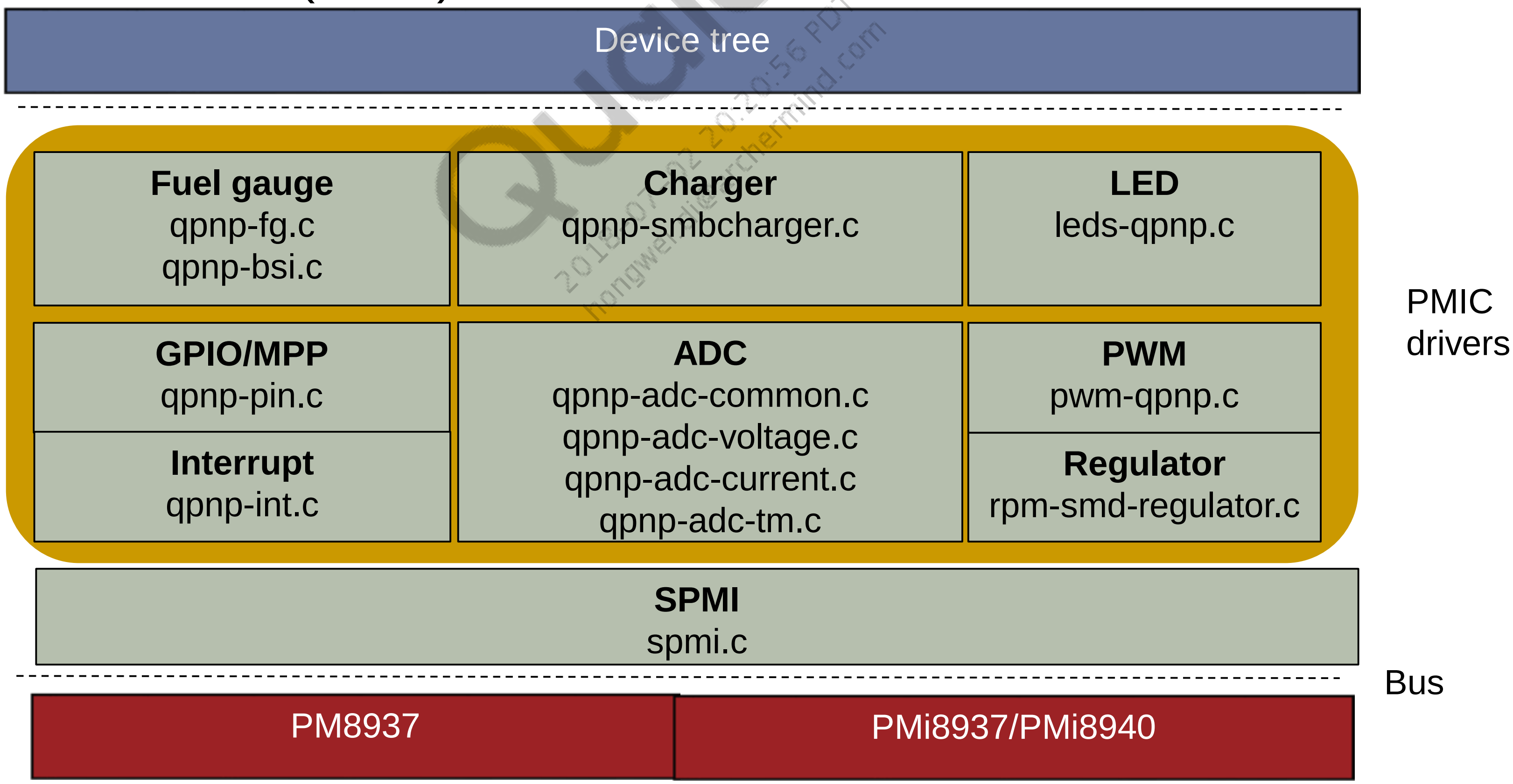
1. Input power management
2. General housekeeping
3. User interfaces
4. IC-level interfaces
5. Configurable I/Os



 80-P2485-2 C November 2016

MSM8937/MSM8917/MSM8940 Linux PMIC Software Architecture

The PM8937, PMi8937 and PMi8940 chips interface with the MSM8937/MSM8917/MSM8940/MSM8920 chipsets through the system power management interface (SPMI) bus and provide features such as analog-to-digital conversion (ADC), charger, fuel gauge, LED, and pulse width modulation (PWM).



MSM8937/MSM8917/MSM8940 Power-On and Reset

- In the kernel, the software performs the following resets:
  - Shutdown
  - Hard reset
  - Warm reset
- Before the system powers down, specific functions are called in msm-poweroff.c:
  - msm_restart_prepare() – For warm and hard reset
  - do_msm_poweroff() – For normal shutdown

Note: No Warm/Soft Reset support for Mission mode. OEMs can still use soft reset for debug, but hard reset must be used for Mission mode.

SPMI and Interrupts

PM8937 and PMi8937 SPMI Driver

- PMIC drivers utilize the SPMI buses to access the PMIC registers.
  - SPMI master – Provides SPMI-related functions in the power controller of the State of Charge (SoC)
  - SPMI slave – Provides SPMI-related functions in a PMIC
  - Request-Capable Slave (RCS) device – Initiates sequences on an SPMI bus
- SPMI driver source – /drivers/spmi/spmi.c

PM8937 and PMi8937/PMi8940 SPMI Slave IDs

- Each PMIC consists of two slave IDs.
  - Each slave ID has 64 k addresses, that is, 128 k of addresses per PMIC.
  - Addresses are subdivided into 256 groups of 256 addresses.
  - Each group is known as a peripheral.

PMIC chips	Reserved slave ID	Example
Primary PMIC	USID0 and USID1	PM8937, PM8994
Interface PMIC	USID2 and USID3	PMi8937/PMi8940, PMI8994

- Internally, the USID is translated into a local slave ID (LSID).
  - USIDs 0, 2, and 4 map to LSID 0.
  - USIDs 1, 3, and 5 map to LSID 1.
- The address is broken down into:
  - LSID
  - Peripheral ID (PID)
  - Register offset
- The LSID is provided in all register maps. The USID is needed when the PMIC is accessed from the SPMI bus.
  - For PM8937, add 0×00000 to the address (no change).
  - For PMi8937, add 0×20000 to the address.

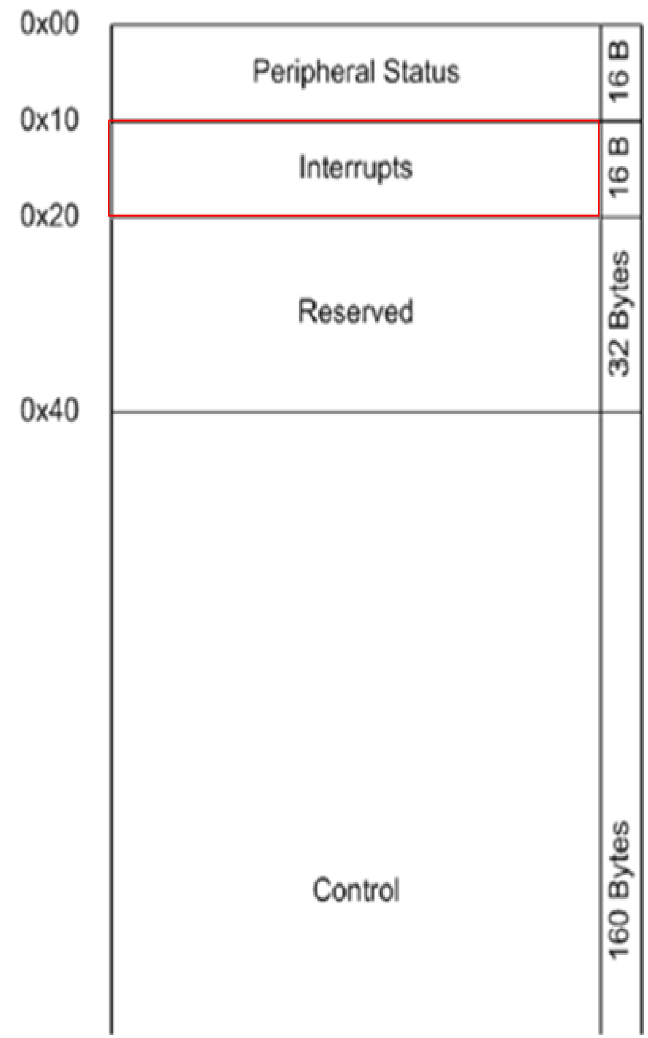
Interrupts

- Each peripheral has interrupts (offset is 0×10 – 0×1E) contained within its peripheral register map.
  - Each register is reserved for a different function.
  - Each bit defines a different interrupt.

Register: **INT_RT_STS** : 0x1300 + 0x0010 (0x1310)

Type:	Read
Reset State:	0x0000

Bits	Field Name	Field Values
6	USBID_CHANGE_INTR_RT_STS	*READ ONLY*
5	AICL_DONE_RT_STS	*READ ONLY*
4	OTG_OVERCURRENT_RT_STS	*READ ONLY*
3	OTG_FAIL_RT_STS	*READ ONLY*
2	USBIN_SRC_DET_STS_RT_STS	*READ ONLY*
1	USBIN_OV_RT_STS	*READ ONLY*
0	USBIN_UV_RT_STS	*READ ONLY*



Interrupt Trigger Types

- FE – Falling edge trigger
- RE – Rising edge trigger
- BE – Both edges; interrupt is triggered on both the interrupt edges, or in other words, on a state change
- H – High-level trigger; triggered when the expression is true; cannot be cleared if the signal is still high
- L – Low-level trigger; triggered when the expression is false; cannot be cleared if the signal is still low

Note: Interrupt trigger types are defined in kernel/include/linux/interrupt.h.

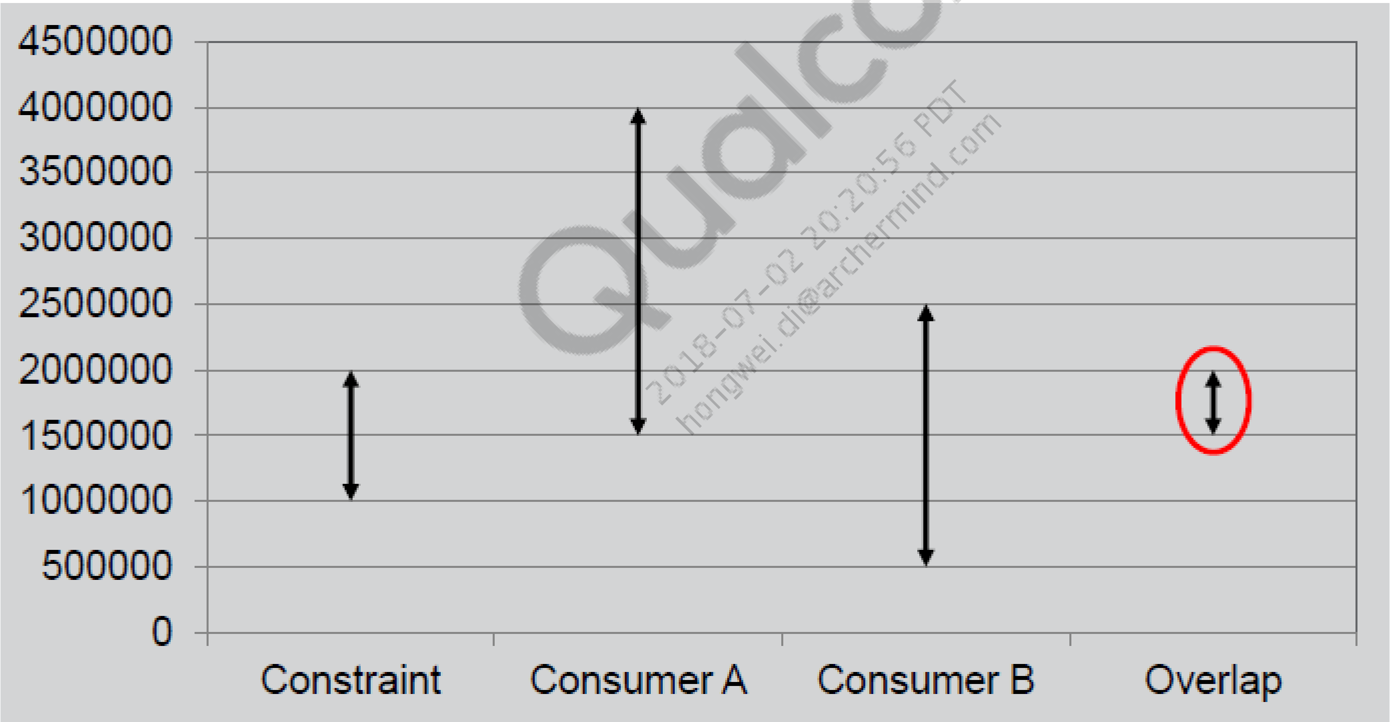
Regulator

PM8937 Regulator Driver and Device Tree Source (DTS) File

- Power grid
  - MSM8937/MSM8917/MSM8940/MSM8920 – The MSM8937/MSM8917/MSM8940/MSM8920 Linux Android Current Consumption Data document is still to be published.
- The MSM8937/MSM8917/MSM8940/MSM8920 power grid is yet to be published.
- The PM8937 regulator driver provides voltage and current requests to the RPM based on the needs of the system.
- Regulator driver source is located at:
  - For bringup – kernel/drivers/regulator/qpnp-regulator.c
  - For official use – kernel/drivers/regulator/rpm-smd-regulator.c
- DTS – kernel/arch/arm/boot/dts/qcom/msm-pm8937-rpm-regulator.dtsi Device Tree Source Include (DTSI) documentation is located at:
  - kernel/Documentation/devicetree/bindings/regulator/qpnp-regulator.txt
  - kernel/Documentation/devicetree/bindings/regulator/rpm-smd-regulator.txt

Voltage Aggregation Example

- Board file constraint for regulator foo − [1000, 2000] mV
- Consumer A request for foo − [1500, 4000] mV
- Consumer B request for foo − [500, 2500] mV



- Regulator driver sets to the minimum physically possible setpoint in the overlap range of [1500, 2000] mV

Load Current Aggregation Example

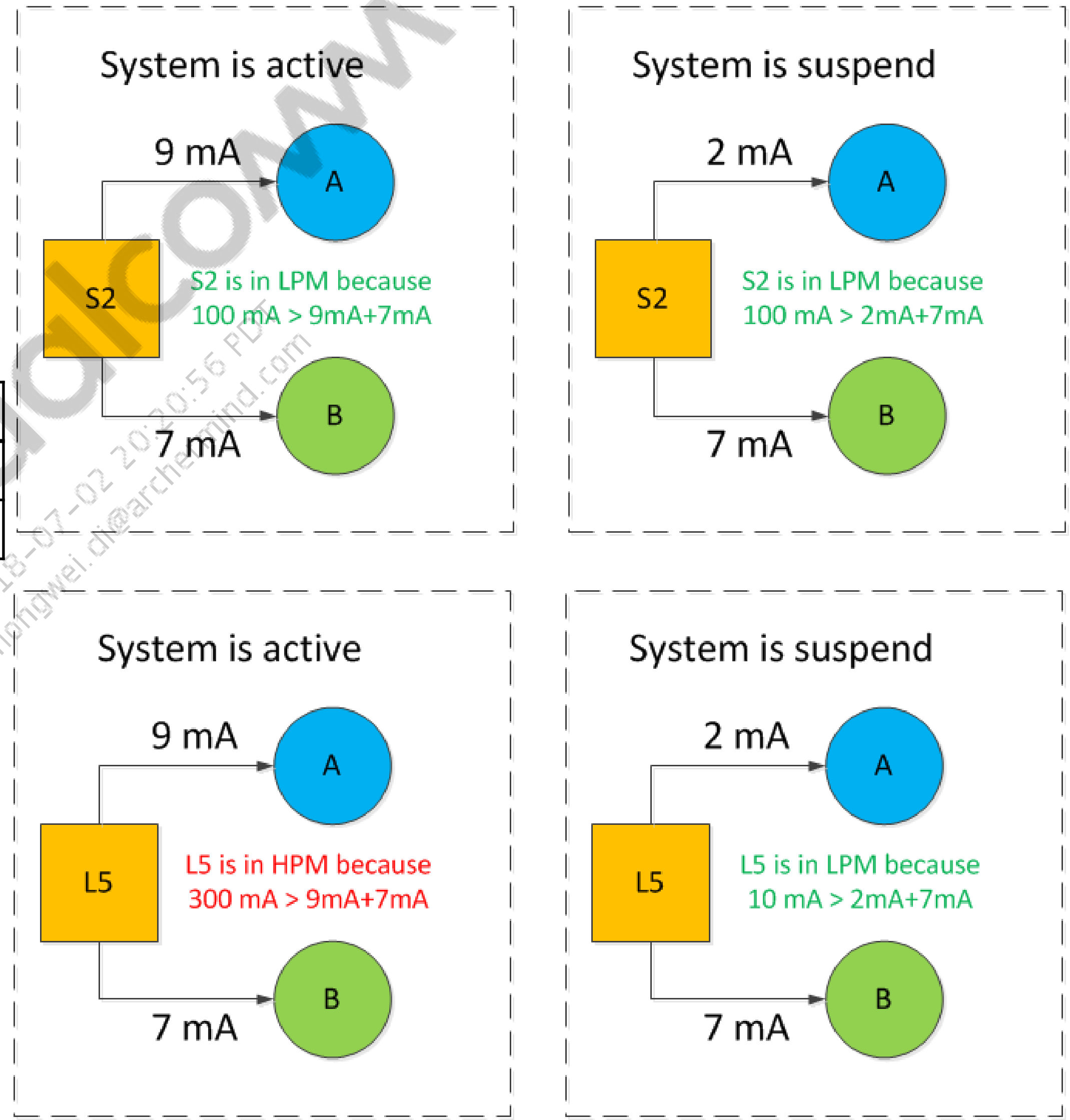
- Two regulators – S2 and L5

Regulator	LPM	HPM
S2	100 mA	1500 mA
L5	10 mA	300 mA

- Two clients – A and B

Client	Active	Suspend
A	9 mA	2 mA
B	7 mA	7 mA

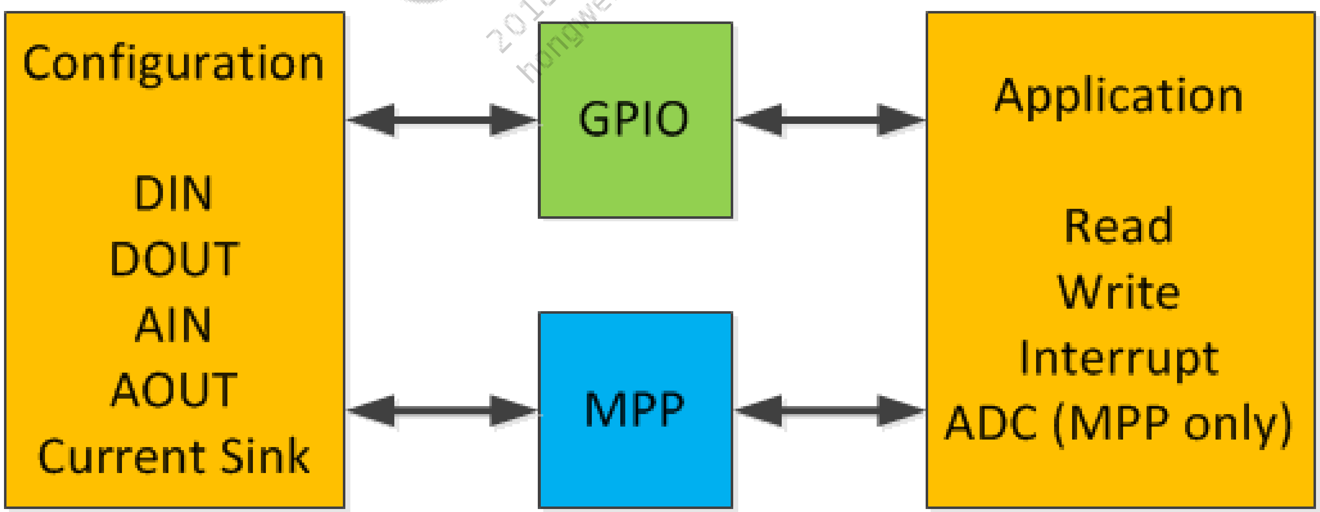
- L5 is in high power mode when the system is active, as the drawn current is larger than its low power mode can support



GPIO/MPP

QPNP Pin Driver and DTS File

- The QPNP pin driver is a GPIO/MPP driver to configure the PMIC pins.
- Pin driver source – kernel/drivers/gpio/qpnp-pin.c
  - DTS for PM8937 – kernel/arch/arm/boot/dts/qcom/msm-pm8937.dtsi
  - DTS for PMi8937 – kernel/arch/arm/boot/dts/qcom/msm-pmi8937.dtsi
  - DTS for PMi8940 – kernel/arch/arm/boot/dts/qcom/msm-pmi8940.dtsi
- DTSI documentation – kernel/Documentation/devicetree/bindings/gpio/qpnp-pin.txt



ADC

ADC Driver and DTS File

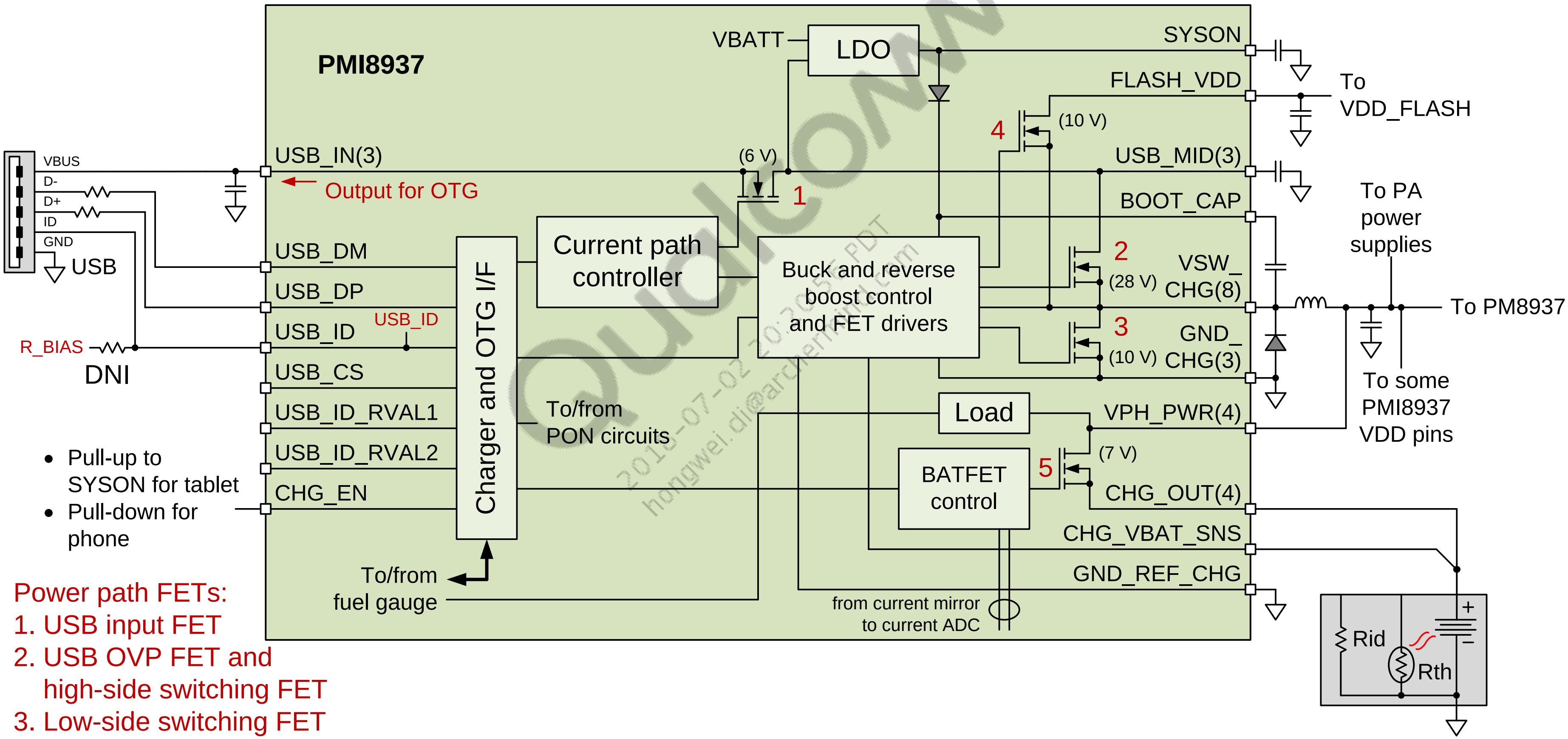
- The ADC drivers read values from the VADC and IADC channels and perform the ADC conversions.
- ADC drivers
  - kernel/drivers/hwmon/qpnp-adc-common.c
  - kernel/drivers/hwmon/qpnp-adc-voltage.c
  - kernel/drivers/hwmon/qpnp-adc-current.c
- ADC DTS
  - kernel/arch/arm/boot/dts/qcom/msm-pm8937.dtsi
- ADC DTSI document
  - kernel/Documentation/devicetree/bindings/hwmon/qpnp-adc-voltage.txt
  - kernel/Documentation/devicetree/bindings/hwmon/qpnp-adc-current.txt

Charger

PMi8937/PMi8940 Charger

- The PMi8937/PMi8940 charger driver acts to configure charger parameters for the hardware finite state-machine to regulate the charging current.
- Charger driver source – kernel/drivers/power/qpnp-smbcharger.c
  - DTS – kernel/arch/arm/boot/dts/qcom/msm-pmi8937.dtsi
  - DTS – kernel/arch/arm/boot/dts/qcom/msm-pmi8940.dtsi
- DTSI documentation – kernel/Documentation/devicetree/bindings/power/qpnp-smbcharger.txt
- USB charger up to 1.5 A, 4 V to 7.2 V input range, 28 V OVP in case of PMi8937
- USB charger up to 2 A, 4 V to 7.2 V input range, 28 V OVP in case of PMi8940
- No DC charger support
- Automatic input current limit for universal USB adapter compatibility
- USB OTG and HDMI/MHL power support (1 A at +5 V)
- Up to 750 mA charging output from 500 mA USB port using Qualcomm Turbo Charge™ mode
- CHG_LED allows four patterns on the notification LED, off, solid on, blink pattern1, and blink pattern2. The software is allowed to override controls of CHG_LED and control the LED regardless of charger presence.

PMi8937/PMi8940 Charger (cont.)



Power path FETs:

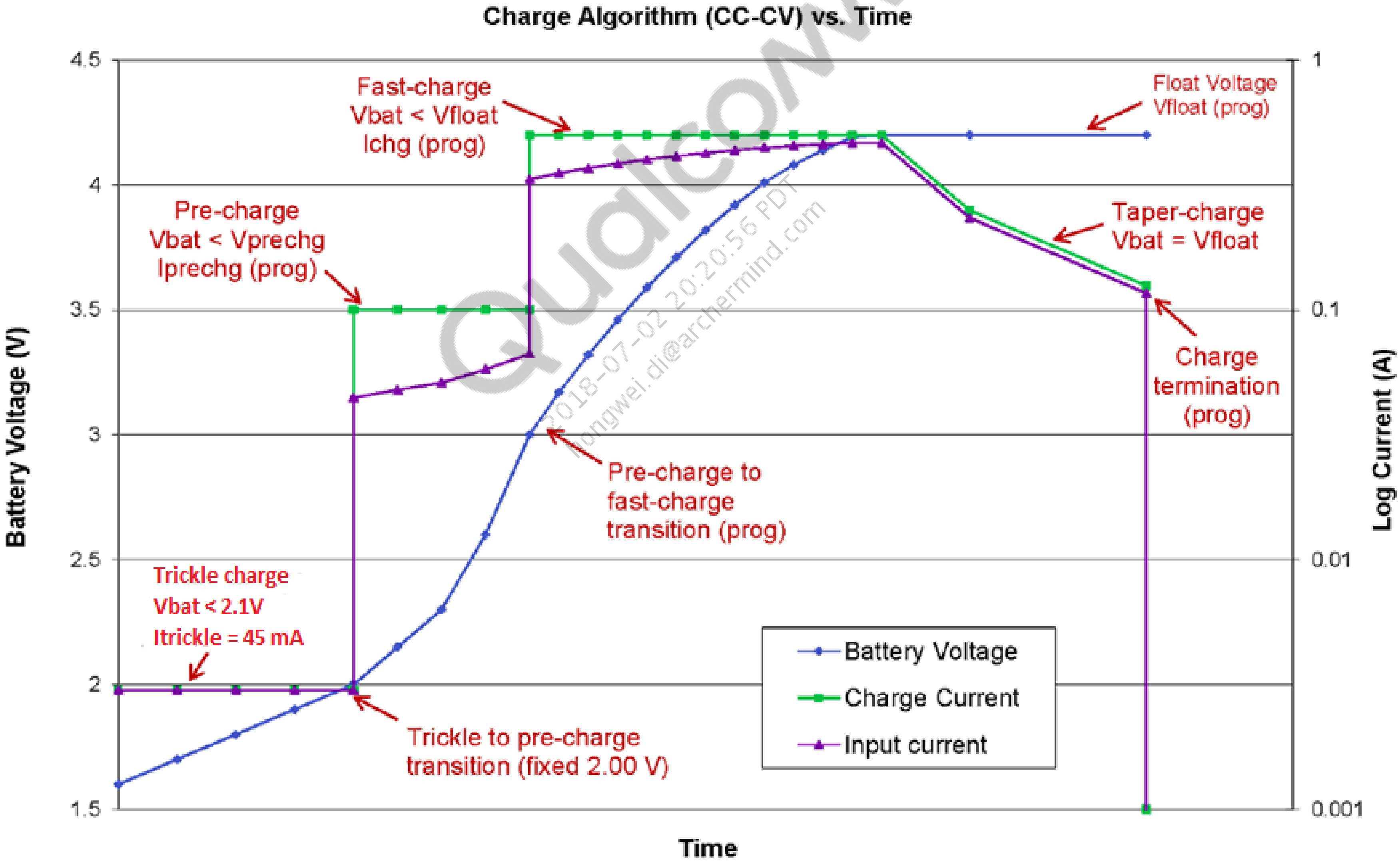
1. USB input FET
2. USB OVP FET and high-side switching FET
3. Low-side switching FET
4. Flash VDD FET
5. BATFET

PMi8937/PMi8940 Charger Cycle

- Charge conditions
  - Input voltage < Over voltage lockout (OVLO)
  - Input voltage > Under voltage lockout (UVLO)
  - Input voltage > VBAT + 0.1 V
- Trickle charge
  - VBAT < 2.1 V
  - IBAT – 45 mA
- Precharge
  - VBAT – 2.4 V to 3.0 V
  - IBAT – 100 to 250 mA
- Constant current charge
  - IBAT – 300 mA to 1500 mA in case of PMi8937
  - IBAT – 300 mA to 2000 mA in case of PMi8940
  - Constant voltage charge
  - VFLOAT – 3.6 V to 4.5 V

PMi8937/PMi8940 Charger Cycle (cont.)

- Charger cycle plot



PMi8937/PMi8940 Charger Cycle (cont.)

- Charge completion – The charge cycle is considered complete when the charge current reaches the programmed termination current threshold
- Automatic battery recharge conditions
  - VBAT < VFLOAT-V_RECH
  - Battery temperature returns to normal
- Charger inhibits threshold
  - Programmable to prevent charging initiation upon power cycling or charge enabling/disabling unless the battery voltage is 50 mV, 100 mV, 200 mV, or 300 mV below the float voltage
  - When the charge inhibit function is enabled, the automatic recharge threshold is overridden to the (higher) charge inhibit voltage threshold

PMi8937/PMi8940 Battery Missing Detection (BMD)

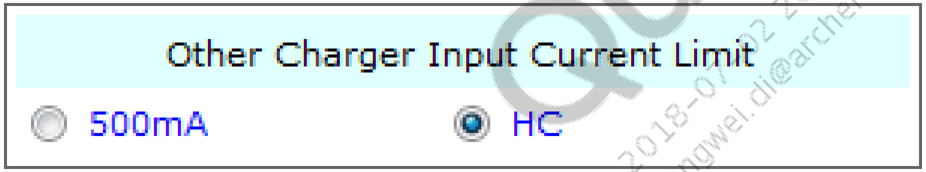
- The battery must be present before the charging starts.
- Battery missing detection can be based on two pin sources:
  - BAT_ID
  - BAT_THERM
- The PMIC can be programmed to check for a missing battery in two ways:
  - BMA – Check at the beginning of the charge cycle (terminal-based).
  - BMA – Check every 2.6 sec (terminal-based); results in higher power consumption.
    - For each poll, the system provides a 10 mA discharge current for a short time (~ 100 ms).
    - BATT_MISSING_INT can be set by the PMi register.

Automatic Power Source Detection (APSD)

- APSD detects the USB charger type based on D+ and D- signal lines
- SDP
  - D+ and D- are grounded
  - USB 1.1 – 100 mA
  - USB 2.0 – 100 mA/500 mA
  - USB 3.0 – 150 mA/900 mA
- DCP
  - D+ and D- are short
  - A standard wall charger capable of at least 500 mA

Automatic Power Source Detection (APSD) (cont.)

- Nonstandard charger
  - D+ and D- are float
  - Input current – 500 mA
- Other Charger Port (OCP)
  - D+ and D- are fixed to specific voltages
  - Input current – 500 mA or run by automatic input current limiting (AICL)



AICL

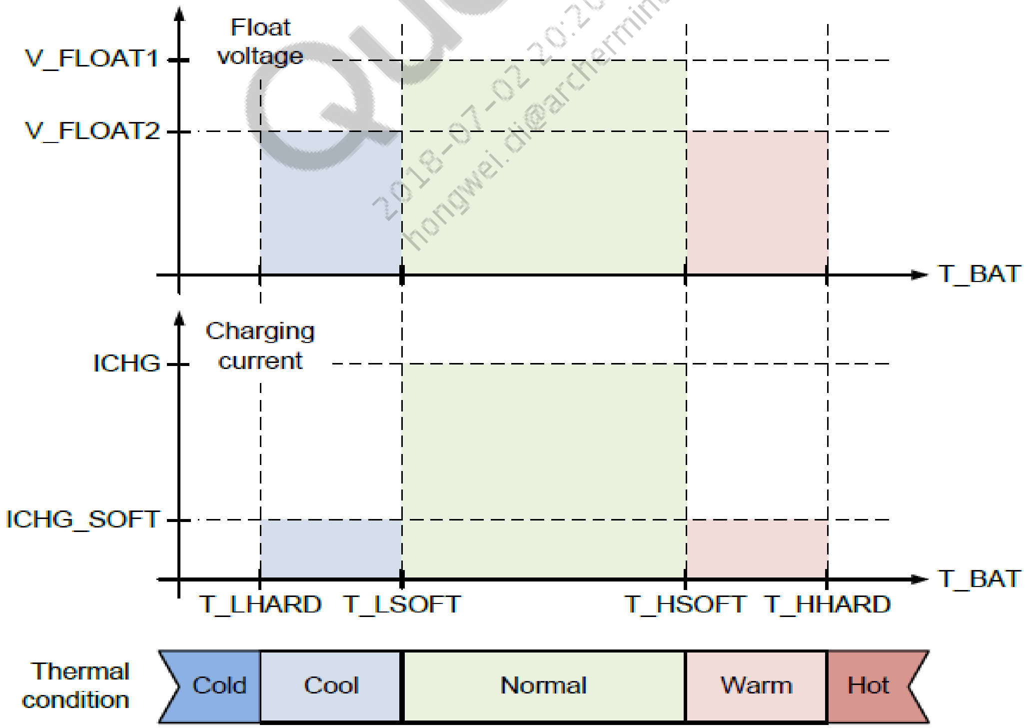
- AICL is hardware-based in PMi8937/PMi8940.
- If AICL is enabled, three events can trigger the algorithm:
  - AICL operation is running but not yet complete
  - AICL operation is complete but source voltage has collapsed
  - Current setting in the volatile register is updated with a value lower than the AICL setting
- Manual AICL rerun reasons are:
  - Hardware AICL may stop at lower current under certain conditions
  - Input current limitation (ICL) is higher

AICL (cont.)

- The hardware periodically reruns AICL after a specific time period if the rerun option is enabled at the start.
- The minimal programmable time of AICL rerun is 45 sec.
- To save power, the software method is used to rerun AICL by enabling and disabling it immediately.
- It is recommended that the ICL must initially be set to the maximum allowable value and then let AICL adjust it.
- ICL is initially set in OTP in the CSIR file.

JEITA Compliance

- Four thresholds divide the battery temperature into five areas, cold, cool, normal, warm, and hot
- Charging current and float voltage adjust is based on thermal zones
- Hardware-based function
- Software also supports configuration of JEITA thresholds through the device tree



Thermal Mitigation

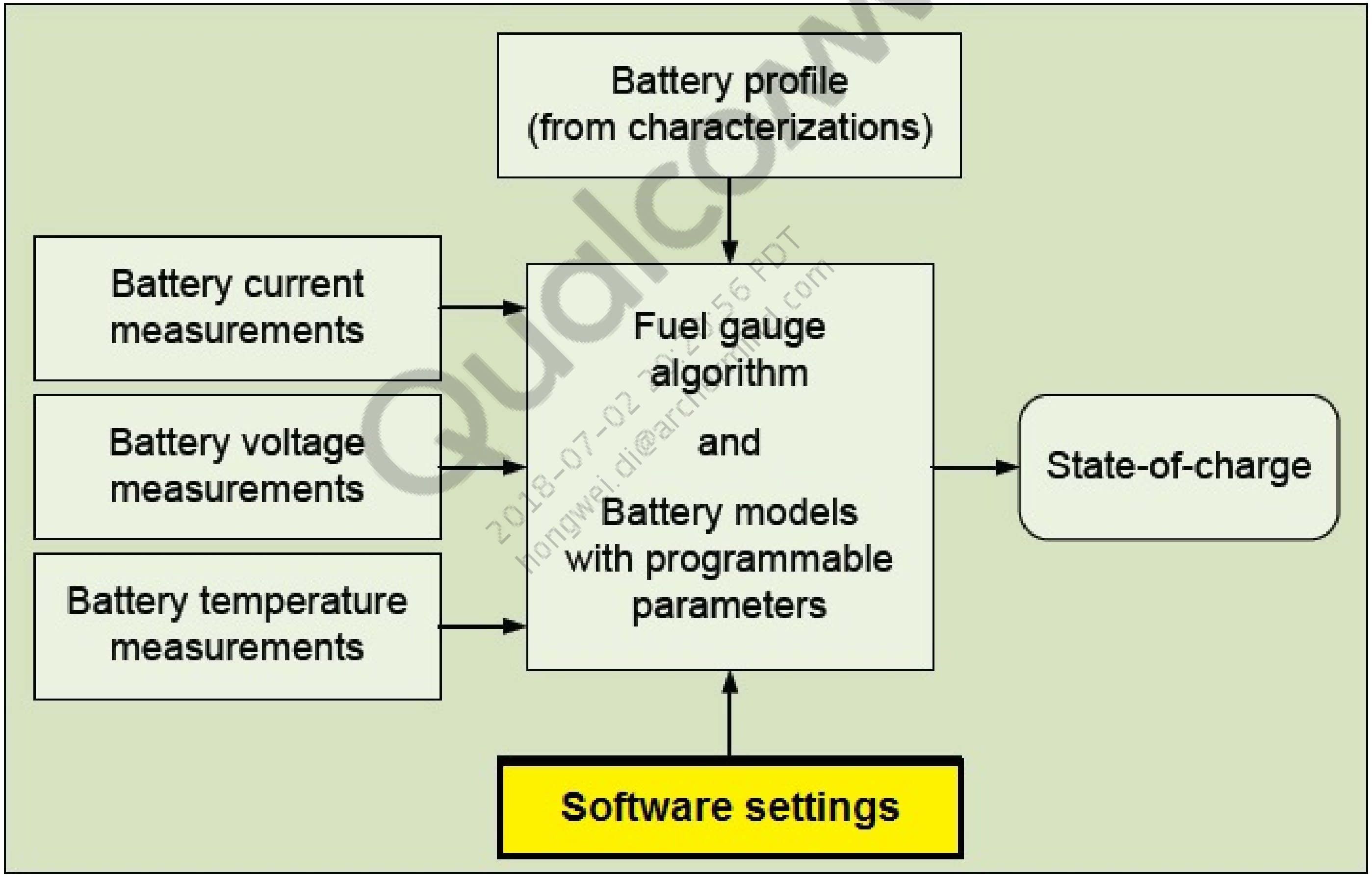
- The PMi8937 features four on-chip temperature sensors that function whenever the chip is powered, either from the battery or one of the inputs.
- Three of these sensors (OTST1, OTST2, and OTST3) are connected to interrupt signals.
- Interrupts trigger whenever the die temperature exceeds the respective threshold.

Sensor name	IRQ	Functions with input missing yes/no	Settings (°C)
OTST1	Yes	Yes	90
OTST2	Yes	Yes	100, 110, 120, 130
OTST3	Yes	Yes	130
TLIM_SD	No	Yes	150

- Once junction temperature at the OTST2 sensor reaches approximately 100°C, 110°C, 120°C, or 130°C, the input current is reduced to prevent high power dissipation.

Fuel Gauge

Flow



Battery profile
(from characterizations)
Battery current
measurements
Battery voltage
measurements
Battery temperature
measurements
Fuel gauge
algorithm
and
Battery models
with programmable
parameters
State-of-charge
Software settings

Driver

- SoC is calculated in the hardware and retrieved by the software via register reading
- Source code driver tree structure
  - Driver – kernel/drivers/power/qpnp-fg.c
  - DTS – kernel/arch/arm/boot/dts/qcom/msm-pmi8937.dtsi
  - DTS – kernel/arch/arm/boot/dts/qcom/msm-pmi8940.dtsi
  - Documentation – Documentation/devicetree/bindings/power/qpnp-fg.txt
- Supplies user space power supply framework properties
  - POWER_SUPPLY_PROP_CAPACITY – Current SoC
  - POWER_SUPPLY_PROP_CURRENT_NOW – Instantaneous battery current
  - POWER_SUPPLY_PROP_VOLTAGE_NOW – Instantaneous battery voltage
- Resolution for current and voltage are 39 mA and 39 mV
- Fuel gauge awakes system at full (100%) or empty (0%)
- Each SoC change generates a DELTA_SOC interrupt

Driver (cont.)

- Battery profiles
  - Up to four profiles are stored in the OTP
  - Selected profile is stored in SRAM, either through OTP or software through SPMI
  - OEM-unique profile is loaded into SRAM at bootup and overwrites generic profiles
- Interrupts
  - high-soc – SoC > configurable high SoC threshold
  - low-soc – SoC < configurable low SoC threshold
  - full-soc – SoC = 100%
  - empty-soc – SoC = 0% at shutdown SoC OR voltage < configurable empty voltage
  - delta-soc – Configurable delta SoC to update user space
  - first-est-done – Triggers after profile is loaded and battery detection is performed in the fuel gauge hardware
  - sw-fallbk-ocv
  - sw-fallbk-new-battrt-sts

PWM

PWM Driver

- The qpnp-pwm driver supports PWM functionality to support a range of applications, for example, varying display brightness, LED dimming, and so on.
- PMi8937 does not support light pulse generator (LPG) and RGB
- PWM driver source
  - kernel/drivers/pwm/pwm-qpnp.c
  - kernel/drivers/pwm/core.c
- PWM documentation – kernel/Documentation/pwm.txt
- DTSI documentation – kernel/Documentation/devicetree/bindings/pwm/pwm-qpnp.txt
- Required bindings to support the PWM feature
  - qcom,period – PWM period time in ms
  - qcom,duty – PWM duty time in ms
  - Label – PWM

PWM Calculation

- PWM duty cycle = $\frac{A}{B}$, where:
  - A = PWM value in decimal
  - B = 2^ (PWM_SIZE)
- PWM frequency = $\frac{master\ clock\ frequency}{B * C * D}$, where:
  - Master clock frequency = 1.024 kHz, 32.764 kHz, 19.2 MHz, or off
  - C = Divide ratio (1, 3, 5, or 6)
  - D = 2M (M = 0 through 7)
- When PWM APIs are used, the PWM driver calculates and configures the master clock, divides ratio, M, and resolution to best fit the requested period.
- Use PWM frequency > 100 Hz to avoid flickering.
- Use a slow PWM frequency to see blinking.

LED

LED Driver and DTSI File

- The QPNP LED driver is used to control LEDs that are part of the PMIC on QTI reference platforms. PMi8937/PMi8940 has WLED, two flash LEDs, and one MPP general-purpose driver that can be used for home row LED.
- LEDs driver source
  - kernel/drivers/leds/leds-qpnp.c
  - kernel/drivers/leds/leds-qpnp-wled.c
- DTSI documentation – kernel/Documentation/devicetree/bindings/leds/leds-qpnp.txt
- Required properties for each child node, WLED, flash, and torch driver
  - Compatible – Should be QTI, leds-qpnp
  - qcom,id – Must be one of the values supported in enum qpnp_led
  - Label – Type of LED used, that is, WLED
  - qcom,max-current – Maximum current the LED can sustain in mA
  - linux,name – Name of the LED used in the LED framework

LED Types

- WLED – Used primarily as display backlight; display subsystem uses LED triggers for WLED to control brightness as needed
- Flash – Used primarily as a camera or video flash
- MPP – Controlled through a multipurpose pin for home row LED

References

Title	Number
Qualcomm Technologies, Inc.	
PM8937 + PMI8937 Power Management IC Design Guidelines/Training Slides	80-P2564-5B
MSM8937 Linux Android Current Consumption Data	80-P2468-7
MSM8917 Linux Android Current Consumption Data	80-P2470-7
MSM8940 Linux Android Current Consumption Data	80-P4978-7

Acronym or term	Definition
ADC	Analog-to-digital conversion
AICL	Automatic input current limiting
APSD	Automatic power source detection
DTS	Device tree source
DTSI	Device tree source include
IADC	Current analog-to-digital conversion
ICL	Input current limitation
LPG	Light pulse generator

References (cont.)

Term	Definition
LSID	Local slave ID
OCP	Other charger port
OVLO	Over voltage lock out
PMIC	Power management integrated circuit (with RTC)
PWM	Pulse width modulation
RCS	Request-capable slave
RTC	Real time clock
SoC	State of charge
SPMI	System power management interface
UVLO	Under voltage lock out
VADC	Voltage analog-to-digital conversion

Questions?

https://createpoint.qti.qualcomm.com