Linux Power!

(From a view of a PMIC vendor)

Matti Vaittinen

ROHM Semiconductors

Jan 10 2023

Topics

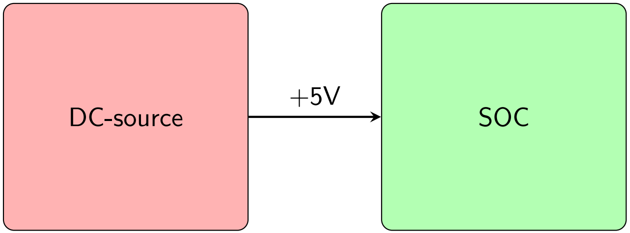
1. What and Why is a PMIC?
2. Regulator provider/consumer
3. Monitoring for abnormal conditions
4. Setting safety limits
5. A helper for sending notifications for “anomalies”

About Me

- Matti Vaittinen
- Kernel/Diver developer at RPHM Semiconductor
- Worked at Nokia BTS projects (networking, clock & sync) 2005 – 2017
- Currently mainly developing/maintaining upstream Linux device drivers for ROHM ICs

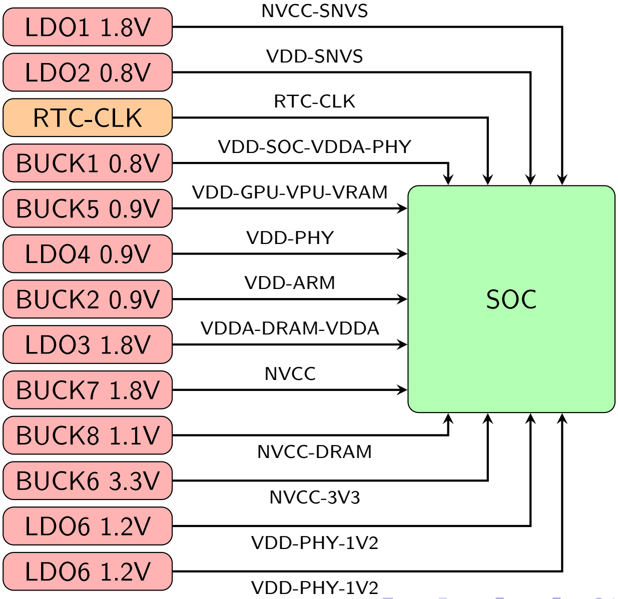
Powering a processor

- Processor and peripherals need power
- Can be as simple as a dummy DC power source with correct voltage



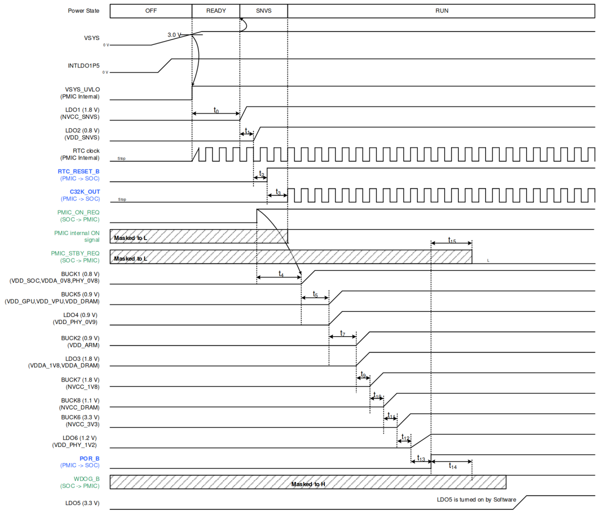
Powering a modern SOC 1/2

Modern SOCs can require multiple specific voltages



Powering a modern SOC 2/2

And specific timings...



Power State
OFF
READY
SNVS
RUN
VSYS
3.0 V
0 V
INTLDO1P5
0 V
VSYS_UVLO (PMIC Internal)
LDO1 (1.8 V) (NVCC_SNVS)
LDO2 (0.8 V) (VDD_SNVS)
RTC clock (PMIC Internal)
Stop
RTC_RESET_B (PMIC -> SOC)
C32K_OUT (PMIC -> SOC)
Stop
PMIC_ON_REQ (SOC -> PMIC)
PMIC internal ON signal
Masked to L
PMIC_STBY_REQ (SOC -> PMIC)
Masked to L
BUCK1 (0.8 V) (VDD_SOC,VDDA_0V8,PHY_0V8)
BUCK5 (0.9 V) (VDD_GPU,VDD_VPU,VDD_DRAM)
LDO4 (0.9 V) (VDD_PHY_0V9)
BUCK2 (0.9 V) (VDD_ARM)
LDO3 (1.8 V) (VDDA_1V8,VDDA_DRAM)
BUCK7 (1.8 V) (NVCC_1V8)
BUCK8 (1.1 V) (NVCC_DRAM)
BUCK6 (3.3 V) (NVCC_3V3)
LDO6 (1.2 V) (VDD_PHY_1V2)
POR_B (PMIC -> SOC)
WDOG_B (SOC -> PMIC)
Masked to H
LDO5 (3.3 V)
LDO5 is turned on by Software
t0
t1
t2
t3
t4
t5
t7
t9
t10
t11
t12
t13
t14
t15

Figure 3-21. Power ON Sequence

More control...

Power savings by:

- Shutting down not needed devices
- Stand-by state(s)
- DVS (Dynamic Voltage Scaling)

Automated power on

Powering-on a system at given time...

- RTC

...Or by an event

- HALL sensor, ...

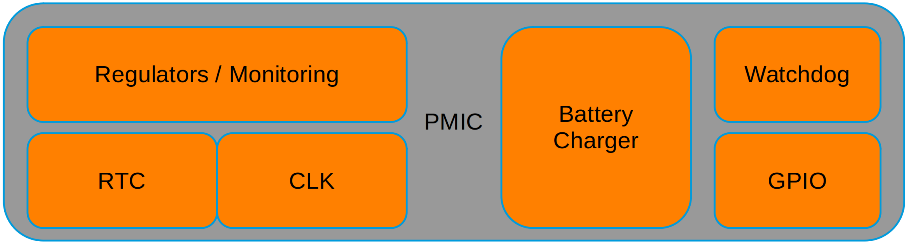
More requirements...

- Battery / charger
- Watchdog
- Functional-safety
  - Voltage monitoring
  - Current monitoring
  - Temperature monitoring

PMICs

PMIC - Power Management Integrated Circuit

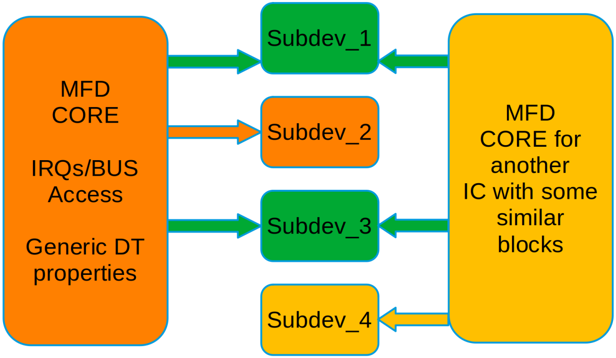
- Multiple DC sources with specific start-up / shut-down sequence
- Voltage control
- Functional-safety
- Auxiliary blocks to support various needs



Multi Function Devices

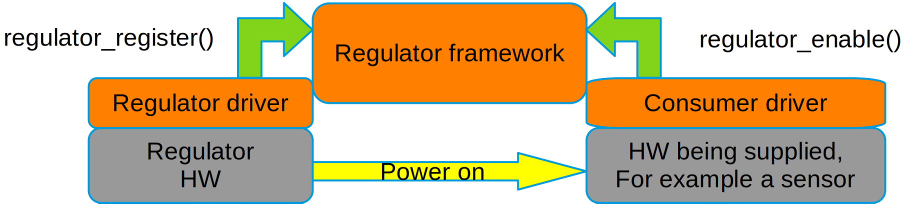
Often MFD drivers which allows re-use

- **Regulator**
- RTC
- Power supply
- Watchdog
- GPIO
- CLK ...



Regulator (provider) and consumer

- Provider is driver interfacing the hardware. Eg, sits “below” the regulator framework. Between regulator framework and HW
- Consumer is driver who wishes to control the regulator using the regulator framework. Eg, sits “on top of” the regulator framework
- PMIC driver is the provider driver (usually just referred as a regulator driver)



Detecting undexpected

Linux has 3 severity categories

- Severity **PROTECTION**
  - Unconditional shutdown by HW
- Severity **ERROR**
  - Irrecoverable error, system not expected to be usable. Error handling by software.
- Severity **WARNING** - NEW(ish)
  - Something is off-limit, system still usable but a recovery action should be taken to prevent escalation to errors