How to use STPMIC2L for a wall adapter powered application on STM32MP21x MPUs

Introduction

This application note applies to the STM32MP21x MPU devices as detailed in the table below. The devices are referred to as STM32MP21x in the rest of the document. It is powered by the STPMIC2L power management IC companion chip, which is fully featured to supply a core chipset composed of:

- STM32MP21x
- DDR memory
- flash memory

This document provides an example of a hardware reference design based on a STM32MP21x device. The STM32MP21x is powered by an external 5 V power supply via the STPMIC2L power management IC. The STPMIC2LAPQR is suited for running I/O peripherals at 3.3 V.

This document is intended for product architects and designers who require information about the power management, and STPMIC2L settings. This document focuses on:

- Reference design block diagram
- Power distribution topology
- Power on/off and low power management
- User reset and crash recovery management
- Safety management and PMIC tuning.

Table 1. Applicable products

Reference	Applicable products
STM32MP21x	STM32MP211, STM32MP213, STM32MP215
STPMIC2L	STPMIC2LAPQR

1 General information

This document applies to STM32MP21x Arm® Cortex®-based MPUs and STPMIC2L power management IC.

2 Overview

This application note describes the interaction between the STM32MP21x, and the STPMIC2LAPQR including the management of the following peripherals:

- DC input power source from main power supply: 5 V typical (4.1 V to 5.5 V).
- DDR4 memory.
- Peripheral I/O interface voltage (V_{DDIO}) at 3.3 V powered by the STPMIC2LAPQR.
- USB 2.0 high speed port.
- eMMC flash memory (HS200) as boot device.

Not covered in this application note:

- DDR3L and lpDDR4
- Peripheral interface with I/O voltage (V_{DDIO}) of 1.8 V

In this document, MPU terminology refers to the STM32MP21x. The PMIC terminology refers to the STPMIC2L generic devices. The STPMIC2LA reference is used to highlight specific behaviors predefined in STPMIC2LAPQR NVM.

2.1 Reference documents

Table 2. Reference documents

-	Reference	Title
STMicroelectronics document[1]		
[1]	AN6055	Getting started with STM32MP21x MPUs hardware development
[2]	DS14940	Power management IC for MPU: 3 buck converters and 7 LDOs
[3]	RM0506	STM32MP21x advanced Arm®-based 32/64-bit MPUs
[4]	AN5726	How to implement low-power modes on the STM32MP2 MPUs
[5]	DS14556	Arm® based Cortex®-A35 1.5 GHz + Cortex®-M33 MPU with TFT, 2× USB 2.0, crypto

1. *Refer to www.st.com*

3 Glossary

Table 3. **Glossary**

Term	Meaning
BUCK	Step down regulator
LDO	Low drop out linear regulator
MPU	Microprocessor unit
NVM	Nonvolatile memory
PMIC	Power management integrated circuit
SMPS	Switching mode power supply
SW	Software

4 5 V power supply application reference design

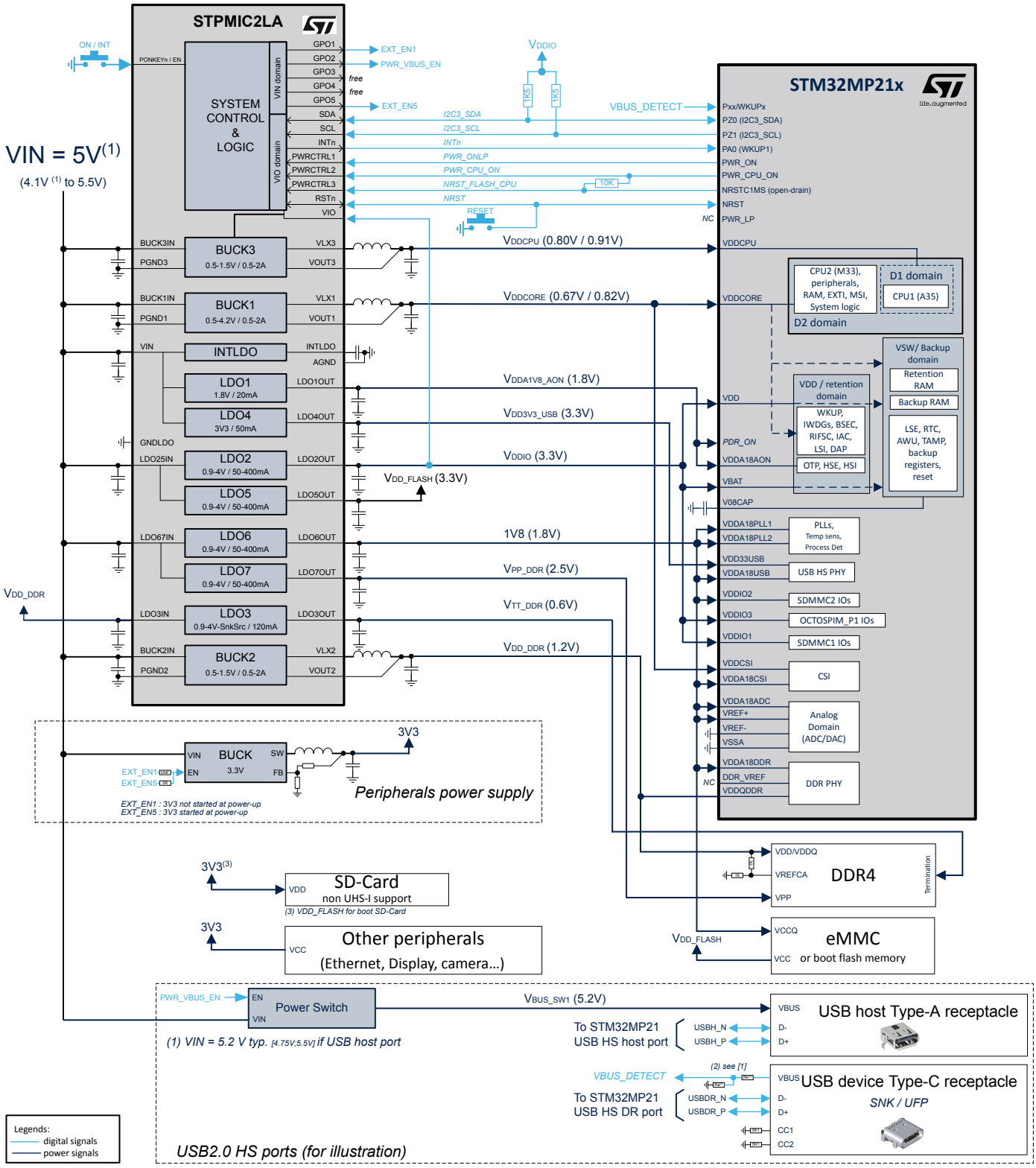
This reference design targets an application powered by a 5 V power supply. The STM32MP21 core voltages, the DDR4 DRAM and the eMMC boot flash memory are powered from the STPMIC2LA. For illustration, application peripherals such as USB ports, SD-Card, Ethernet PHY, and so on are powered from a discrete regulator or a power switch.

Note: The eMMC boot flash memory is used as illustration and can be replaced by any supported flash memory such as SD-Card, NAND flash memory, serial flash memory and so on as described in [1].

The main peripheral interfaces function with an I/O voltage of 3V3.

The overall system is illustrated in Figure 1.

Figure 1. STM32MP21x and STPMIC2LA with DDR4, eMMC



Note: The following are not shown in the diagram:

- *STM32MP21x decoupling scheme (see [1])*
- *STPMIC2LA discrete components value (see [2])*
- *V_{IN} source and related protection, such as ESD, EMI filtering, overvoltage.*

4.1 Power distribution

The PMIC integrates the regulators that supply:

- The STM32MP21x power domains
- The DDR4 DRAM
- A flash memory such as eMMC.

4.1.1 V_{DDCPU} power domain (800 mV/910 mV)

V_{DDCPU} supplies the Arm® Cortex®-A35 platform (CPU1), called the D1 domain.

V_{DDCPU} is powered from the PMIC BUCK3 step-down SMPS. The SMPS has an excellent load transient response across all operating conditions.

At power-up, the V_{DDCPU} is automatically enabled by the PMIC at 800 mV, which corresponds to the “nominal mode” voltage. (See Section 5.2.1: Power-up triggered by main supply (V_{IN}) plugin/power-down by software shutdown).

V_{DDCPU} is enabled in:

- Run1 mode
- Low power LP-Stop1 mode
- Low power LPLV-Stop1 mode.

V_{DDCPU} is disabled in:

- Run2 mode
- Low power LP-Stop2 mode
- Low power LPLV-Stop2 mode
- Low power Standby mode
- V_{BAT} mode and OFF mode.

In low-power modes, the PWR_CPU_ON output of the MPU manages the PMIC V_{DDCPU} regulator. The PWR_CPU_ON is connected to the PWRCTRL2 input of the PMIC.

At runtime, the CPU1 can operate in *nominal mode* or in *overdrive mode*. The V_{DDCPU} voltage is then adjusted to the chosen mode. The MPU manages the transition between *nominal mode* voltage and *overdrive mode* voltage by sending I²C commands to the PMIC. The V_{DDCPU} voltage is increased to the “overdrive mode” voltage value (910 mV) when the CPU1 frequency (Fcpu1_overdrive) operates above 1200 MHz. When the CPU1 operates in “nominal mode” at 1200 MHz or below, the V_{DDCPU} must be set back to “nominal mode” voltage value (800 mV).

4.1.2 V_{DDCORE} power domain (670 mV / 820 mV)

V_{DDCORE} is the main STM32MP21x digital power domain, and is called the D2 domain.

V_{DDCORE} supplies all the digital circuits, which include:

- The Arm® Cortex®-M33 platform (CPU2)
- CSI

V_{DDCORE} is powered from the PMIC BUCK1 step-down SMPS. This SMPS has an excellent load transient response across operating conditions.

At power-up, V_{DDCORE} is automatically enabled by the PMIC at 820 mV. (See Section 5.2.1: Power-up triggered by main supply (V_{IN}) plugin/power-down by software shutdown)

V_{DDCORE} is enabled in:

- Run1 mode
- Run2 mode
- Low power LP-Stop1 mode
- Low power LP-Stop2 mode.

The voltage is lowered to 670 mV in:

- Low power LPLV-Stop1 mode
- Low power LPLV-Stop2 mode.

V_{DDCORE} is disabled in:

- Low power Standby mode
- V_{BAT} mode
- OFF mode.

In low power mode, the PWR_ON output of the MPU manages the PMIC V_{DDCORE} regulator. The PWR_ON output is connected to the PWRCTRL1 input of the PMIC. V_{DDCORE} also supplies the retention domain in Run1 and Run2, and Stop1 and Stop 2 modes. (See Section 4.1.8: MPU backup domain and retention domain).

4.1.3 V_{DDIO} and V_{DDA1V8_AON} power domains

V_{DDIO} is the power supply for the following independent MPU I/Os:

- V_{DD}
- V_{DDIO1}
- V_{DDIO2}
- V_{DDIO3}

V_{DDIO} is also the power supply of the MPU V_{DD} for the retention domain (see Section 4.1.8: MPU backup domain and retention domain for more details). These separate/dedicated I/O supplies can be set to different voltages or be shut down independently.

V_{DDA1V8_AON} domain supplies the MPU V_{DDA1V8_AON} system analog such as:

- Reset block
- Power management (POR/PDR)
- Oscillators (HSE, HSI)
- OTP controller (BSEC)

V_{DDIO} is powered from the PMIC LDO2 linear regulator. V_{DDA1V8_AON} is powered by the dedicated PMIC LDO1 linear regulator, which has a very low quiescent current to reduce power consumption during low power mode.

At power-up, V_{DDIO} and V_{DDA1V8_AON} are automatically enabled to 3.3 V and 1.8 V respectively by the PMIC. They are the first regulators switched on at power-up (see Section 5.2.1: Power-up triggered by main supply (V_{IN}) plugin/power-down by software shutdown).

V_{DDIO} and V_{DDA1V8_AON} are ON in all modes except in OFF or V_{BAT} mode; when the main power source (V_{IN}) of the application is removed (see Section 4.1.8: MPU backup domain and retention domain for details).

4.1.4 V_{DD3V3_USB} power domain

V_{DD3V3_USB} power domain supplies the USB2 HS PHY (V_{DD3V3_USB}) of the MPU.

Note: Examples of USB implementations with the STM32MP21x are provided in the document [1].

V_{DD3V3_USB} is powered from the dedicated PMIC LDO4 with a fixed output voltage of 3.3 V.

At power-up, the V_{DD3V3_USB} regulator is automatically enabled to 3.3 V by the PMIC (See Section 5.2.1: Power-up triggered by main supply (V_{IN}) plugin/power-down by software shutdown).

If a USB peripheral is connected to the application, V_{DD3V3_USB} can be kept enabled in the following modes:

- Run1 mode
- Run2 mode
- Low power LP-Stop1 mode
- Low power LP-Stop2 mode
- Low power LPLV-Stop1 mode
- Low power LPLV-Stop2 mode.

The V_{DD3V3_USB} must be disabled in Standby mode and OFF mode. When 1V8 (MPU VDDA18USB) is disabled to fulfil the following MPU constraint:

Caution: *$V_{DDA18USB}$ must be present whenever $V_{DD33USB}$ is present or 1 ms maximum after V_{DD3V3_USB}. Any other sequense may result is damage to the system.*

4.1.5 DDR power domain (V_{DD_DDR}, V_{TT_DDR}, V_{PP_DDR})

Several power domains are dedicated to supplying the DDR types supported by the MPU: DDR3L, DDR4, and lpDDR4.

This application focuses on DDR4 topology.

V_{DD_DDR} (1.2 V) is powered from the PMIC BUCK2 step-down SMPS to power the DDR4 memory ICs (V_{DDR} and V_{DDQ}) and MPU DDR PHY (V_{DDQDDR}) domains.

V_{TT_DDR} (0.6 V) is powered from the PMIC multipurpose LDO3. When set in sink-source mode, the LDO3 provides voltage equal to V_{REF_DDR} (implicitly V_{OUT2} / 2). The LDO3 sink-source mode is dedicated to power the DDR4 memory bus terminations.

Note: When used in sink-source mode, the BUCK2 output (V_{DD_DDR}) must supply the LDO3 (LDO3IN).

V_{PP_DDR} (2.5 V) is powered from the PMIC general purpose LDO7 to power the DDR4 memory V_{PP}.

At power-up, the PMIC does not automatically start the following regulators:

- V_{DD_DDR}
- V_{TT_DDR}
- V_{PP_DDR}

The regulators are powered up sequentially by the software bootloader by sending I²C commands to the PMIC. This is detailed in the following section and in Section 5.2.1: Power-up triggered by main supply (V_{IN}) plugin/ power-down by software shutdown.

At runtime, the PWR_ON output of the MPU, connected to the PWRCTRL1 input of the PMIC, manages the DDR4 power domains as follows:

- V_{TT_DDR} (LDO3) is switched OFF, and the DDR4 is set in self-refresh in the following modes:
  - Low power LP-Stop1 mode
  - Low power LP-Stop2 mode
  - Low power LPLV-Stop1 mode
  - Low power LPLV-Stop2 mode.
- In Standby mode, there are two possible scenarios:
  - DDR4 in self-refresh: similarly to low power modes, is detailed in Section 5.3.4: Standby mode (DDR4 in self-refresh).
  - DDR4 OFF: all DDR regulators are powered OFF and detailed in Section 5.3.5: Standby mode (DDR4 OFF).

At power down, and before the software turns OFF the PMIC, a power-down sequence must be applied on DDR4 power domains to comply with the JEDEC DDR4 specification (see details in Section 4.1.6: V_{DD_FLASH} power domain (3.3 V)).

Note: The PMIC embeds configurable pull-down discharge resistors on each of the regulator outputs that allow all of regulator output voltages to discharge in less than 1.5 ms. For BUCK regulators, two pull-down values are configurable in NVM with one of the following discharge delays:

- *Slow pull-down set to 1.5 ms*
- *Fast pull-down set to 0.3 ms.*

The "fast pull-down" configuration is used to switch off a power supply quickly than another one. The configuration is used on the DDR uncontrolled power down sequence.

Software DDR4 power-up sequence:

1. Power-on event (or standby DDR-OFF mode exit): PMIC power up (or standby DDR-OFF mode recovery)
2. The software bootloader executes DDR4 initialization.
3. The software bootloader sets the PMIC internal pull-down for each DDR regulator as described in the note below:
   a. LDO7 (V_{PP_DDR}) pull-down is disabled when LDO7 is disabled:
   LDOS_PD_CR1[LDO7_PD] = 0
   b. BUCK2 (V_{DD_DDR}) fast pull-down is activated when BUCK2 is disabled:
   BUCKS_PD_CR2[BUCK2_PD]=10
   c. LDO3 (V_{TT_DDR}) pull-down is activated when LDO3 is disabled:
   LDOS_PD_CR1[LDO3_PD] = 1

Note: It is necessary to set this pull-down every time the MPU is powered up to prevent a "DDR4 uncontrolled power OFF sequence" (see DDR4 uncontrolled power off sequence:)

4. The software bootloader enables the DDR regulators:
   a. Enable LDO7 (V_{PP_DDR}) at 2.5 V
   b. Wait 1 ms tempo
   c. Enable LDO3 in sink-source mode (V_{TT_DDR})
   d. Enable BUCK2 (V_{DD_DDR}) at 1.2 V: V_{TT_DDR} voltage follow V_{DD_DDR} ramp up.
   e. Wait 1 ms tempo
5. Software initializes the MPU DDR4 memory controller and DDR4 devices.

Note: The above sequence aims to fulfill the JEDEC DDR4 where V_{PP_DDR} must ramp at the same time or before V_{DD_DDR}, and V_{PP_DDR} must always be equal to or higher than V_{DD_DDR}. Refer to the latest JEDEC DDR4 specification for more details.

Software DDR4 power down sequence:

1. Software receives an event to enter OFF mode or standby DDR-OFF mode.
2. Assert DDR CKE low
3. Disable BUCK2 (V_{DD_DDR}): V_{TT_DDR} voltage follows V_{DD_DDR} ramp down.
4. Wait 1 ms tempo
5. Disable LDO3 (V_{TT_DDR})
6. Disable LDO7 (V_{PP_DDR})

Note: V_{PP_DDR} voltage falls slowly as LDO7 pull-down discharge is disabled.

DDR4 uncontrolled power off sequence:

An uncontrolled power off sequence occurs typically when the main power supply voltage is removed or when a reset occurs.

In this case, the PMIC manages the power off sequence:

Note: To ensure this uncontrolled power down sequence, the PMIC pull-down settings are set at software boot (see Software DDR4 power-up sequence).

1. V_{IN} is removed.
2. V_{IN} falls below the VINOK_fall threshold: PMIC triggers a *turn off* condition.

3. The PMIC starts the power off sequence.
   a. The PMIC asserts the NRST signal: STM32MP21x stops the DDRPHYC especially the DRAM differential clocks (DDRA_CKP/N and DDRB_CKP/N)
   b. The PMIC disables RANK0 regulators:
      - LDO7 (V_{PP_DDR})
      - BUCK2 (V_{DD_DDR})
      - LDO3 (V_{TT_DDR})
4. The voltages of the following regulators are discharged through their respective regulator pull-down resistors:
   – BUCK2 (V_{DD_DDR})
   – LDO3 (V_{TT_DDR})
5. LDO7 (V_{PP_DDR}) falls slowly as its discharge pull-down resistor is disabled.

4.1.6 V_{DD_FLASH} power domain (3.3 V)

V_{DD_FLASH} supplies the *eMMC flash memory IC core domain* (V_{CC}). The eMMC I/O interface (V_{CCQ}) with the related MPU I/O interfaces (VDDIO2) are powered by the 1V8 power domain to work in high speed mode (HS200).

V_{DD_FLASH} is powered from the PMIC LDO5 general purpose linear regulator.

At power-up, the V_{DD_FLASH} regulator is enabled automatically by the PMIC to 3.3 V (see Section 5.2.1: Power-up triggered by main supply (V_{IN}) plugin/power-down by software shutdown), enabling the MPU CPU1 to access the eMMC as boot flash peripheral.

V_{DD_FLASH} is enabled in the following modes:

- Run1 mode
- Low power LP-Stop1 mode
- Low power LPLV-Stop1 mode.

Note: When no read/write access is expected, the MPU software can disable V_{DD_FLASH} at runtime or prior to enter LP-Stop1 mode or LPLV-Stop1 mode mode.

V_{DD_FLASH} is disabled in the following modes:

- Run2 mode
- Low power LP-Stop2 mode
- Low power LPLV-Stop2 mode
- Low power Standby mode
- OFF mode.

The PMIC embeds an advanced feature allowing to manage an independent reset of the regulator (the LDO5 in this application) that supplies the boot flash peripheral IC (the eMMC in this application). It allows especially to manage the MPU CPU1 independent reboot in case of a CPU1 crash in addition to manage the Standby mode exit. See PMIC power control management independent reset source section for more details.

To control this feature, the MPU NRSTC1MS output is connected to the PWRCTRL3 input of the PMIC. See PWRCTRL1, PWRCTRL2, PWRCTRL3 section for more details.

4.1.7 1V8 power domain

1V8 is an analog power supply domain for the MPU, which are:

- PLLs ($V_{DDA18PLLx}$)
- USB2 PHY ($V_{DDA18USB}$)
- CSI ($V_{DDA18CSI}$)
- ADC ($V_{DDA18ADC}$)
- DDR PHY($V_{DDA18DDR}$).

The 1V8 power domain is also used to supply power to peripherals such as eMMC I/O interface (V_{CCQ}).

The 1V8 power domain is powered from the PMIC LDO6.

At power-up, the 1V8 regulator is automatically enabled by the PMIC (See Section 5.2.1: Power-up triggered by main supply (V_{IN}) plugin/power-down by software shutdown).

1V8 is enabled in following modes:

- Run1 mode
- Run2 mode
- Low power LP-Stop1 mode
- Low power LP-Stop2 mode
- Low power LPLV-Stop1 mode
- Low power LPLV-Stop2 mode

1V8 is disabled in the following modes:

- Low power Standby mode
- V_{BAT}/OFF modes.

4.1.8 MPU backup domain and retention domain

The MPU has two internal power domains to keep critical data and security data in memory (see Figure 2):

- The backup domain supplies:
  - RTC
  - LSE
  - TAMPER
  - Backup registers
  - Backup RAM
  - Retention RAM

They must be retained when V_{DDIO} is turned off to keep critical data or security. The backup domain is powered in all operating modes, including V_{BAT} mode where V_{BAT} is typically powered from a coin cell backup battery.

- The retention domain supplies:
  - OTP controller (BSEC)
  - Independent watchdog (IWDG)
  - Resource isolation framework controller (RIFSC and IAC)
  - Debug access port (DAP).

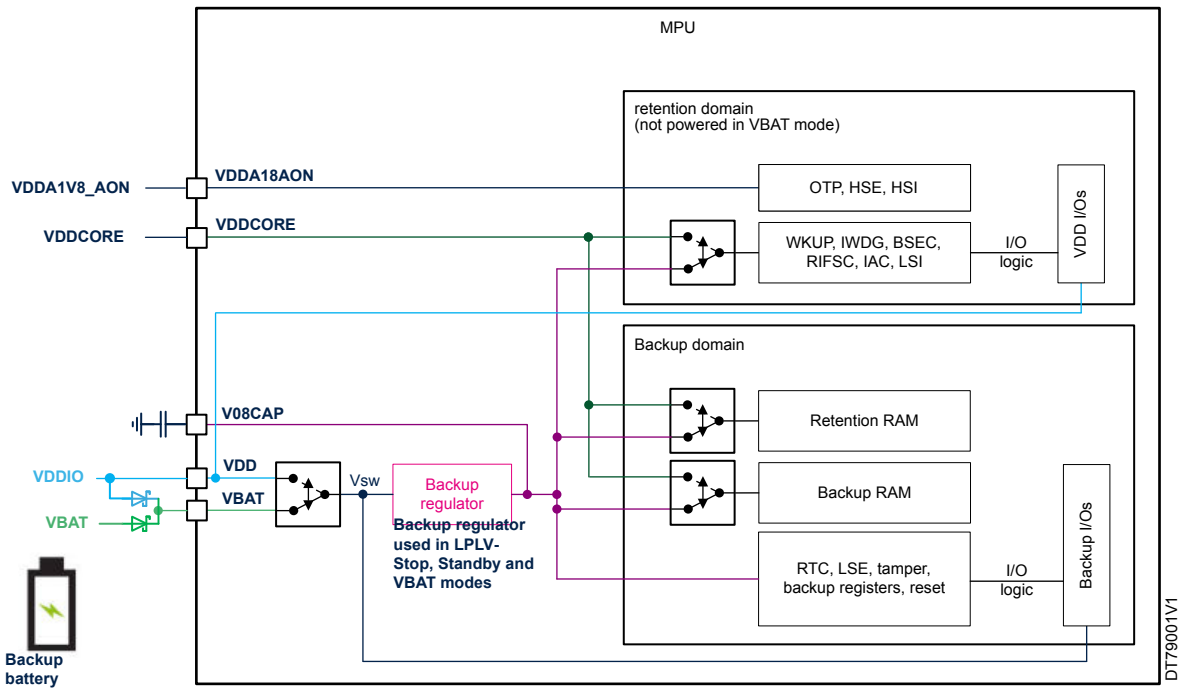
The retention domain is powered in all operating modes, excluding V_{BAT} mode.

These two domains are powered by either: V_{DDCORE}, V_{DD}, or V_{BAT} depending on which supply is available and the operating mode of the MPU:

- The backup domain and the retention domain are powered from the V_{DDCORE} supply in the following modes:
  - Run1 mode
  - Run2 mode
  - Low power LP-Stop1 mode
  - Low power LP-Stop2 mode.
- The backup domain and the retention domain are powered from the V_{DD} supply via the internal backup regulator (decoupling capacitor on V08CAP pin) in the following modes:
  - Low power LPLV-Stop1 mode
  - Low power LPLV-Stop2 mode
  - Low power Standby mode
- In V_{BAT} mode, where V_{BAT} is powered from a backup battery, the internal backup domain is powered from the V_{BAT} supply via the backup regulator.

Note: The retention domain is not powered in V_{BAT} mode.

Figure 2. **Backup and retention domain**



For more details on the constraints on V_{BAT} rise time, refer to the document [5].

4.1.9 3V3 external peripherals power domain

The 3V3 power domain is used to power peripherals around the MPU, such as an SD Card, the display, the ethernet, and so on.

Figure 1 illustrates how the 3V3 domain is powered from a 3V3 discrete regulator to illustrate the interaction with the PMIC GPOs.

The STPMIC2L has five general purpose outputs (GPO). Their push-pull topology on PMIC V_{IN} voltage allows the control of external regulators like internal PMIC regulators (PMIC GPOs are driven with same I^2C registers as internal PMIC regulators). This allows GPOs to be set automatically by PMIC at power-up.

For STPMIC2LA:

- GPO1, GPO2, GPO3 are keep disabled at PMIC power-up
- GPO4, GPO5 are enabled by PMIC during power-up sequence (See [2] for more details)

Note: GPOx disabled = 0 V ; GPOx enabled = V_{in}

4.2 Control signals between STPMIC2LA and STM32MP21x

This section outlines the way the STM32MP21x microprocessor communicates with the STPMIC2LA device. Several interfaces are available depending on the application requirements.

4.2.1 STPMIC2LA default behavior with STM32MP21x

The STPMIC2LA NVM settings are configured to boot the MPU application from flash memory such as eMMC or to boot the MPU from the USB interface. In production, booting from the USB interface is suitable for flashing and then executing the production software and/or the final application software. The default NVM configuration is available in [2] and summarized in the following table.

Table 4. Default STPMIC2LA NVM configuration

Regulator	Name	Rank	Default output voltage	Default configuration
BUCK1	V_{DDCORE}	RANK2	0.82 V	ON
BUCK2	V_{DD_DDR}	RANK0	N/A	OFF
BUCK3	V_{DDCPU}	RANK4	0.8 V	ON
LDO1	V_{DDA1V8_AON}	RANK1	1.8 V	ON
LDO2	V_{DDIO}	RANK1	3.3 V	ON
LDO3	V_{TT_DDR}	RANK0	N/A	OFF
LDO4	V_{DD3V3_USB}	RANK5	3.3 V	ON
LDO5	V_{DD_FLASH}	RANK5	3.3 V	ON
LDO6	1V8	RANK3	1.8 V	ON
LDO7	V_{PP_DDR}	RANK0	N/A	OFF
GPO1	EXT_EN1	RANK0	N/A	OFF
GPO2	PWR_VBUS_EN	RANK0	N/A	OFF
GPO3	-	RANK0	N/A	OFF
GPO4	-	RANK5	N/A	ON
GPO5	EXT_EN5	RANK5	N/A	ON

In order to start the application, the PMIC regulator start up is spread across five ranks. This is to comply with the MPU power-up sequence constraints and to avoid current peaks on the main power supply. The voltage value of each regulator is defined to fit with the MPU optimum voltage requirements.

For safety management

The STPMIC2LA safety management is set by default in NVM to systematically restarts the application. When a hard fault is detected, the STPMIC2LA performs a power cycle in the following sequence:

1. reset assertion
2. power down-sequence
3. power-up sequence
4. reset de-assertion to restart the application

Refer to Section 6: Safety management for more details.

PMIC tuning (optional)

The STPMIC2LA NVM can be reprogrammed by the customer to tune the regulator settings, and the safety management behavior to fit with the MPU based application requirements. This action can be done with the STM32CubeProgrammer.

4.2.2 PMIC digital control interface

The PMIC integrates an I²C interface, five digital input control pins: PONKEYn/EN, PWRCTRL1/2/3, five GPOs; a digital output interrupt pin (INTn), and a bidirectional digital reset pin (RSTn).

I²C interface

The MPU controls the PMIC via the I²C interface to:

- Enable/disable, set the voltage, and operating mode of the regulators.
- Dynamic voltage scaling to switch the MPU in nominal or overdrive modes.
- Set regulators external control for low power mechanisms (PWRCTRLx).
- Set the interrupt controller or read interrupt status.
- Set the protection for the watchdog, overcurrent, undervoltage, or read protection status.
- Tune the PMIC NVM default configuration for the end-product (power-up sequence, safety management).

PONKEYn/EN pin (optional)

The PMIC PONKEYn/EN pin is a digital active low input signal. It is usually connected to a user push button “ON/INT” to power on the application. Thanks to the PMIC built-in pull-up resistor, there is no need for a discrete pull-up resistor on this signal.

The PONKEYn/EN signal allows the following operations:

- Turns on the PMIC (from PMIC OFF state)
- Wakes up the application from a low-power mode (typically from Standby mode).
- Forces a switch-off or a power cycling condition with a long press. This duration is programmable as described in Section 5.1.1: Application turn-on/turn-off conditions .

Note: The use of a push “ON/INT” button connected to the PONKEYn/EN is optional as the STPMIC2LA is automatically turned on when the application is powered.

RSTn pin

The PMIC RSTn pin must be connected to the MPU NRST pin. It can also be connected to a “RESET” user push button. This pin has built-in pull-up resistor ; so no additional discrete pull-up resistor is needed on this signal. Nevertheless, a 10 nF capacitor to GND is required to be placed as close as possible to the MPU NRST pin. This is specifically required to avoid EMI/ESD coupling as there is no debounce circuitry in neither the MPU, nor the PMIC.

The PMIC RSTn pin is a digital active low bidirectional signal with a built-in pull-up resistor:

- When PMIC asserts an RSTn (such as during the power-up or the power-down sequence), it drives the MPU NRST signal low (open drain). The MPU is forced into a system reset until the PMIC releases the RSTn.
- When the MPU asserts an NRST signal such as an MPU watchdog event, or by pressing the “RESET” button, the PMIC immediately asserts the RSTn pin and performs a noninterruptible power cycle. The PMIC performs a power down sequence, followed by a power-up sequence and releases the RSTn.

At the end of the power-cycle sequence, the PMIC waits for the MPU NRST signal to go high before rearming the reset detection mechanism to avoid infinite loop reset.

INTn signal

The PMIC INTn pin is a digital output (open drain) active low interrupt line connected to the MPU PA0 input pin. This pin has built-in pull-up resistor. Therefore, no additional discrete pull-up resistor is needed on this signal

PA0 has both interrupt and wake-up capability:

- To manage an interrupt from the PMIC when the MPU operates in either Run1, Run2 mode or a low power mode (except Standby mode).
- To wake up the MPU when it operates in Standby mode.

PWRCTRL1, PWRCTRL2, PWRCTRL3

The PMIC has three power control digital input signals connected to dedicated MPU control signals.

Each PMIC regulator is controlled from a single PWRCTRL signal typically:

- To switch ON or OFF
- To change the output of a regulator depending on the PWRCTRL signal state.

Alternatively, a PWRCTRL signal can be set to reset a regulator at a value defined in STPMIC2LA NVM (see document [2] for details)

MPU power control connection to PMIC:

- The MPU PWR_ON output pin controls the PMIC PWRCTRL1 input pin. Figure 1 illustrates how the MPU PWR_ON output pin controls the PMIC PWRCTRL1 input pin. The MPU PWR_ON pin is internally multiplexed between either PWR_ON or PWR_LP signal (PWR_D2CR[LPCFG_D2] must be set to 1 by the software after a system reset):
  - PWR_ON pin is multiplexed with PWR_LP signal to manage LP-Stop1/LP-Stop2 or LPLV-Stop1/LPLV-Stop2 modes
  - PWR_ON pin is multiplexed with PWR_ON signal (default) to manage Standby mode
- The MPU PWR_CPU_ON output pin controls the PMIC PWRCTRL2 pin. In the application illustrated in Figure 1, the MPU PWR_CPU_ON controls the MPU D1 domain (V_{DDCPU}) when this domain is in DStandby (see [3] for more details) set in Run2, LP-Stop2 mode, LPLV-Stop2 mode and Standby mode. The internal MPU pwr_cpu_on signal can also be multiplexed with the internal MPU pwr_cpu_lp signal. To do this, the PWR_D1CR[LPCFG_D1] must be set to 0 by the software after a system reset. This keeps the PWR_CPU_ON = 1 in LP-Stop1 mode and LPLV-Stop1 mode.
- The MPU NRSTC1MS pin controls the PMIC PWRCTRL3 pin. NRSTC1MS pin is used to control the power supplies of the external flash. This power is required to carry out the first level boot of CPU1 to ensure the platform is rebooted, a full power cycle must be applied to the mass storage boot flash memory such as eMMC. The NRSTC1MS pin is activated when a system reset, or D1 reset (D1 Standby exit) is generated. In this application, the NRSTC1MS is linked to MPU PWR_CPU_ON by a 10 kΩ pull-up resistor. This is illustrated in Figure 1.

See Section 5.1.2: PMIC power control management (PWRCTRLx) for more details about PMIC PWRCTRL settings.

5 Power management

5.1 Operating modes

The application can switch to different operating modes depending on the system activity. The MPU manages the operating modes and they control the power management as detailed in [4]. The operating modes are described in the table below.

Table 5. Operating modes

Operating mode	PMIC state	PWR_ON	PWR_CPU_ON	NRSTC1MS (1)	Description	Notes
Run1	POWER_ON	1	1	1	V_{DDIO} power on V_{DDCORE} power on V_{DDCPU} power on (in normal or overdrive voltage) System clock on Peripherals power on/off DDR4 active	The difference between Run1/2 and Stop1/2 mode is only based on the STM32MP21x clock management and they have no impact on power management.
Run2	POWER_ON	1	0	0	V_{DDIO} power on V_{DDCORE} power on V_{DDCPU} power off System clock on Peripherals power on/off DDR4 active	
LP-Stop1 mode	POWER_ON	0	1	1	V_{DDIO} power on V_{DDCORE} power on V_{DDCPU} power on (in normal voltage) System clock off Peripherals power on/off DDR4 self-refresh with V_{TT_DDR} power off	-
LP-Stop2 mode	POWER_ON	0	0	0	V_{DDIO} power on V_{DDCORE} power on V_{DDCPU} power off System clock off	-

Operating mode	PMIC state	PWR_ON	PWR_CPU_ON	NRSTC1MS (1)	Description	Notes
LP-Stop2 mode	POWER_ON	0	0	0	Peripherals power on/off DDR4 self-refresh with V_{TT_DDR} power off	-
LPLV-Stop1 mode	POWER_ON	0	1	1	V_{DDIO} power on V_{DDCORE} power on at lower voltage. V_{DDCPU} power on in nominal voltage System clock off Peripherals power on/off DDR4 self-refresh with V_{TT_DDR} power off	In the application specified in Figure 1, the MPU PWR_ON signal is internally muxed with the MPU PWR_LP signals so PWR_ON is low in: • LPLV-Stop1 mode • LPLV-Stop2 mode • Standby mode If this multiplexing does not occur, the PWR_ON signal is • high in: – LPLV-Stop1 mode – LPLV-Stop2 mode • low only in Standby mode.
LPLV-Stop2 mode	POWER_ON	0	0	0	V_{DDIO} power on V_{DDCORE} power on at lower voltage. V_{DDCPU} power off System clock off Peripherals power on/off DDR4 self-refresh with V_{TT_DDR} power off	
Standby mode (Standby mode (DDR4 in self-refresh) Standby mode (DDR4 OFF))	POWER_ON	0	0	0	V_{DDIO} power on V_{DDCORE} power off V_{DDCPU} power off System clock off Peripheral power off DDR4 self-refresh or off with V_{TT_DDR} power off	-
V_{BAT}	NO_SUPPLY	-	-	-	Backup domain powered from backup battery	-
OFF	OFF	-	-	-	All regulators are powered off	-

Operating mode	PMIC state	PWR_ON	PWR_CPU_ON	NRSTC1MS (1)	Description	Notes
					Backup domain powered from backup battery if present	

1. *In this application, the NRSTC1MS is linked to MPU PWR_CPU_ON by a pull-up. When the MPU PWR_CPU_ON is low, the NRSTC1MS is also low.*

5.1.1 Application turn-on/turn-off conditions

The PMIC autonomously manages the power-up and the power-down sequence when respectively a turn-on or a turn-off condition occurs.

The STPMIC2LA automatically powers up when the application is powered from a valid power source: When V_{IN} rises above the V_{INOK_rise}, it triggers a PMIC turn-on condition as the “AUTO turn-on” bit is set by default in the STPMIC2LA NVM. The V_{INOK_rise} is the PMIC internal threshold (see [2] for the values).

Turn-on conditions

When the application is in OFF mode (PMIC in the OFF state with V_{IN} present), a turn-on condition is required to power up the PMIC, and then to run the application. Similarly, if the application needs to go into power-off mode, a turn-off condition is required to power-down the PMIC.

If the PMIC is in OFF state, it is powered up by one of the three triggers described in the table below:

Table 6. PMIC turn-on conditions

Condition	Trigger	Description
AUTO turn-on	Internal	The STPMIC2LA starts automatically when V_{IN} rises above V_{INOK_rise}. This feature is set by default in the STPMIC2LA. (see “AUTO turn-ON” section in [2] for details)
PONKEY user button	External	PONKEYn/EN falling edge pin falling edge (PKEY_EN_CFG = 0 in PMIC NVM: by default in STPMIC2LA NVM).
EN pin asserted	External	PONKEYn/EN pin asserted (PKEY_EN_CFG = 1 in PMIC NVM: not a default setting)

After a turn-on condition, the PMIC carries out a transitional power-up sequence as described in Section 5.2: Application Power-up/Power-down sequence.

Turn-off conditions

A turn-off condition leads the PMIC to perform a power-down sequence to go into one of the following states:

- The OFF state
- The FAIL_SAFE_LOCK state (see [2] for the definition).
- Automatic restart (power cycle).

This depends on whether the source is a software switch-off or a hard fault that has triggered the turn-off condition (see detailed about hard fault is Section 6: Safety management). The turn-off conditions are described in the table below.

Table 7. PMIC turn off conditions

Condition	Hard fault	Description
EN	NO	EN deasserted (when PKEY_EN_CFG = 1 in PMIC NVM: not a default setting)
Software switch OFF	NO	I²C commands “SWOFF” sent by the MPU to the PMIC(1)
PONKEY user button	YES	PONKEYn signal is asserted for 10 s (PKEY_EN_CFG = 0 in PMIC NVM: by default in STPMIC2LA NVM). (See Section 6.2.5: PKEY: power on key user button long press)
V_{IN} undervoltage	YES	V_{IN} voltage falls below the PMIC V_{INOK_fall} threshold.(2) (See Section 6.2.2: V_{IN} undervoltage protection (V_{IN} < V_{INOK_Fall}))
Thermal shutdown	YES	PMIC temperature rises above T_{SHDN_Rise} threshold(3) (See Section 6.2.3: TSHDN: thermal shutdown protection)
Overcurrent protection	YES	Overcurrent or short-circuit on predefined regulators (See Section 6.2.1: OCP overcurrent protection)
Watchdog	YES	PMIC watchdog timer elapsed(4) (See Section 6.2.4: WDG: watchdog timer expiration)

1. *The PMIC goes in a transitional power-down state. It then goes and stays in OFF state until a turn-on condition is met. If the restart request bit (RREQ_EN) is set with the SWOFF bit, the PMIC restarts automatically*
2. *If restart conditions are met, the PMIC waits for the V_{IN} voltage to rise above the V_{INOK_rise} threshold before powering up.*
3. *If restart conditions are met, the PMIC waits for the temperature to decrease below T_{SHDN_Fall} before powering up.*
4. *The watchdog timer is not enabled by default.*

Note: The PMIC restarts automatically after a turn-off condition is triggered by a hard fault source (behavior programed by default in the STPMIC2LA NVM).

5.1.2 PMIC power control management (PWRCTRLx)

The PMIC PWRCTRLx signals are dedicated to managing MPU power modes or special regulator reset features. These signals must be correctly configured, and this before entering into low power mode, to ensure proper MPU low power mode entry and exit transitions.

The PMIC PWRCTRLx signals are all independently controlled and each signal controls a PMIC regulator by setting the appropriate registers as described in [2]. As such, it is possible to define:

- The control source selection of the regulator (PWRCTRL1/2/3)
- The polarity of the respective PWRCTRLx signals, which are used to define if the signals are active low or active high.

Note: One of these power controls can be used to suspend the PMIC watchdog, such as when the application is in low power mode.

A PMIC PWRCTRL signal aims to switch between two regulator control registers xxxx_MAIN_CR and xxxx_ALT_CR.

Typically, when a PWRCTRL signal goes to a low state, the PMIC internally switches from the main control register (MAIN) content to the alternate (ALT) control register content and vice versa.

PMIC power control management independent reset source

The PWRCTRL_RST bit is used to enable the regulator independent reset source. When this bit is set, it behaves in either of the conditions below:

- If PWRCTRL is deasserted, the regulator operates according to xxxx_MAIN_CR .
- If PWRCTRL is asserted, the regulator is disabled and the xxxx_MAIN_CR and xxxx_ALT_CR are reset to the default value defined in the NVM. On PWRCTRL deassertion , the regulator operates according to xxxx_MAIN_CR NVM reset content.

This feature is specifically suitable to reset application with flash memories in case of D1 crash (see Section 5.4.2: CPU1 crash recovery management). In this application, the regulator independent reset source is activated by the PWRCTRL3 (MPU NRSTC1MS) and mapped to LDO5 (V_{DD_FLASH})

5.1.3 PMIC mask-reset option

If the application needs to have one or several PMIC regulators enabled while the PMIC performs a reset sequence, the MPU bootloader software must program the PMIC mask reset option by setting:

- The PMIC BUCKS_MRST_CR register to target BUCK converters
- The PMIC LDOS_MRST_CR register to target LDO regulators

A reset sequence is triggered after the PMIC RSTn signal asserted by the MPU or the user reset push button. Refer to [2] for details on the PMIC mask-reset option.

This is typically the case for the LDO2 powering the MPU V_{DDIO} power domains and the LDO1 powering the MPU V_{DDA1V8_AON} power domain. The power cycle on V_{DDIO} and on V_{DDA1V8_AON} must be masked by setting these two PMIC registers:

- LDOS_MRST_CR [LDO1_MRST] = 1 for LDO1
- LDOS_MRST_CR [LDO2_MRST] = 1 for LDO2

This prevents losing the content in:

- The MPU backup RAM
- The MPU retention RAM
- The MPU backup register content
- The JTAG debug interface included in the OTP controller, see Figure 2

Note: The MPU software bootloader must program these settings using I²C commands to the PMIC, following each application power-up. The content of BUCKS_MRST_CR, and LDOS_MRST_CR is reset at the end of a PMIC reset cycle.

5.2 Application Power-up/Power-down sequence

The power up sequence is the transition managed by the STPMIC2LA from application power-off to application Run mode.

The power down sequence is the transition managed by the STPMIC2LA from application Run mode to application power-off.

The application power-up and power-down sequence shown in Figure 3 is based on the reference designed in Figure 1.

5.2.1 Power-up triggered by main supply (V_{IN}) plugin/power-down by software shutdown

When the application is connected to an external power supply, the PMIC powers-up automatically when V_{IN} rises above the STPMIC2LA V_{INOK_rise} threshold. When the power-up ends, the PMIC release the NRST signal.

Note: The STPMIC2LA has the AUTO_TURN_ON and PONKEYn feature enabled by default in NVM settings.

Once the NRST signal is released, the MPU boots (including the DDR4 initialization by the MPU software). The MPU reaches system Run mode.

Note: When the PMIC is powered-up, the PWRCTRLx signals have no effect, they are not initialized.

When the MPU software sends the "SWOFF" command to the PMIC, a turn-off condition occurs, the PMIC enters power-down, and then goes into the OFF mode.

The above sequence is detailed below and illustrated in Figure 3:

1. The application has no power or the MPU is powered by the onboard coin cell V_{BAT}. The V_{BAT} supplies the MPU backup domain.
2. A power supply is connected to the application. V_{IN} voltage rises.

3. Once V_{IN} voltage is above V_{INPOR_Rise} (STPMIC2LA V_{INPOR_Rise} = 2.3 V typ.) threshold:
   a. The PMIC goes to the *INIT&LOAD* transitional state to preload its NVM contents and checks its integrity. If the PMIC NVM integrity is valid, the PMIC initializes launches and executes the VIN_DLY = 10 ms.
   b. Once VIN_DLY elapsed, the PMIC state machine goes in the *CHECK&LOAD* transition state as the AUTO_TURN_ON bit is set in the NVM. This is defined in the Turn-on conditions.

Note: The VIN_DLY timer is a passive delay used to wait for VIN voltage stabilization. It is set to 10 ms by default in the STPMIC2LA NVM. This delay can be changed by reprogramming the NVM.

Note: The PMIC CHECK&LOAD have a typical duration of approximately 7 ms.

4. Once the *CHECK&LOAD* states ends and the V_{IN} supply rises above V_{INOK_rise}(1), the PMIC starts a power-up sequence. The STPMIC2LA regulators follow the power-up sequence predefined in the NVM:
   a. RANK1 (1.5 ms): The LDO2 (V_{DDIO}) is enabled at 3.3 V and the LDO1 (V_{DDA1V8_AON}) is enabled at 1.8 V. Once V_{DDIO} voltage rises above MPU V_{POR} threshold (1.67 V), and V_{DDA1V8_AON} voltage rises above MPU V_{POR_ANA} threshold (1.67 V), the MPU $t_{RSTTEMPO}$ (440 µs typ.) delay is applied. Once $t_{RSTTEMPO}$ elapses, the MPU enters in system reset. After approximately 230 µs needed to ensure internal initialization, the MPU releases the internal `nrst_por`. The MPU then goes into system reset and the PWR_ON signal goes high.

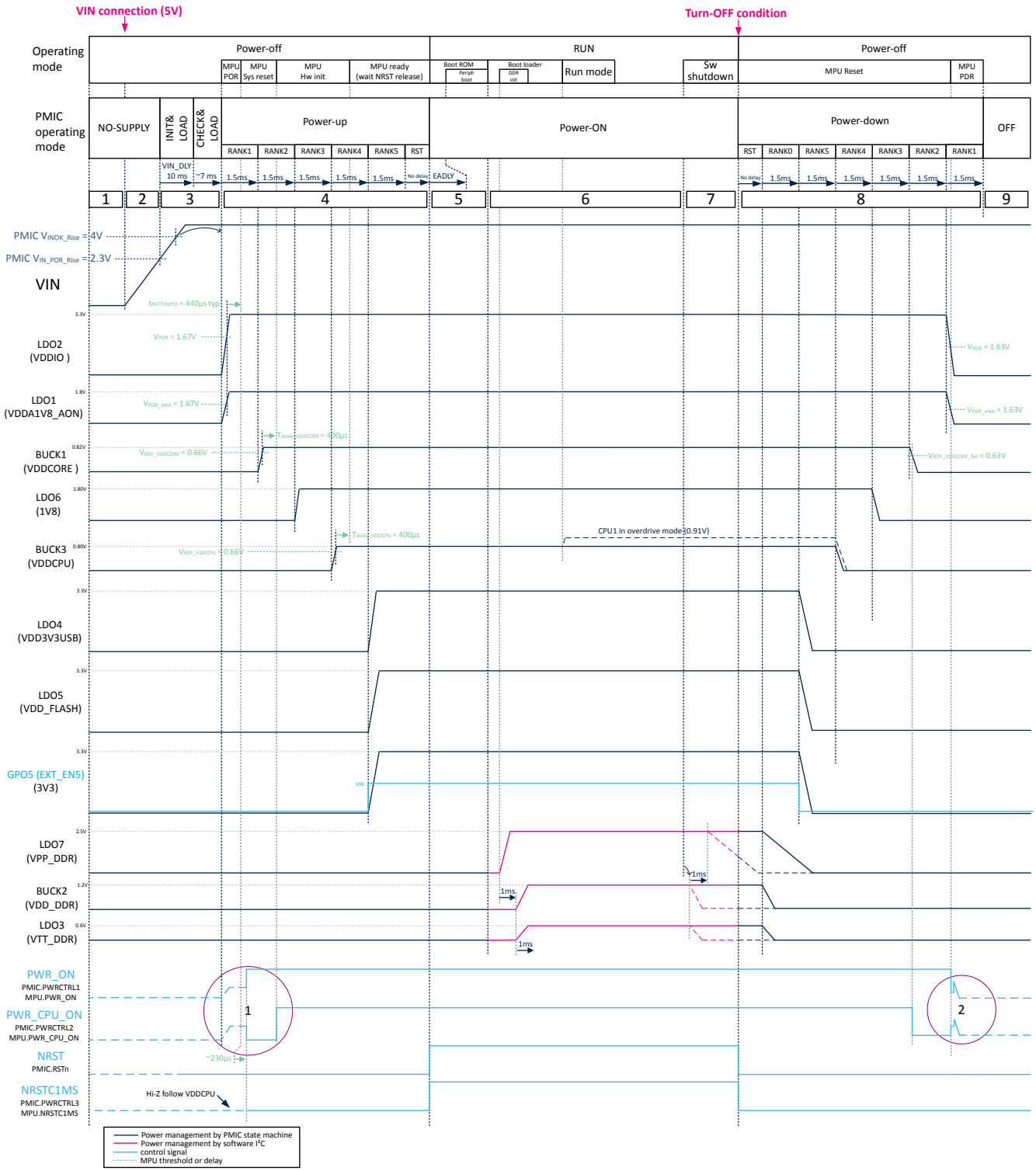
Note: NRST signal is kept low as the PMIC asserts the NRST signal until the end of the power-up sequence.

   b. RANK2 (1.5 ms): The BUCK1 (V_{DDCORE}) is enabled at 0.82 V. Once V_{DDCORE} voltage rises above MPU $V_{RDY\_VDDCORE}$ threshold (0.66 V), an MPU $T_{delay\_VDDCORE}$ (400 µs typ.) delay is started to allow V_{DDCORE} voltage to reach minimum operating voltage. Once $T_{delay\_VDDCORE}$ elapses, the MPU starts hardware initialization, such as starting the HSI oscillator and so on. The PWR_CPU_ON signal goes high.
   c. RANK3 (1.5 ms): The LDO6 (1V8) is enabled at 1.8 V
   d. RANK4 (1.5 ms): The BUCK3 (V_{DDCPU}) is enabled at 0.80 V. Once V_{DDCPU} voltage rises above MPU V_{RDY_VDDCPU} threshold (0.66 V), an MPU T_{delay_VDDCPU} (400 µs typ.) delay is started to allow V_{DDCPU} voltage to reach minimum operating voltage. Once T_{delay_VDDCPU} elapses, the MPU is ready to boot, and is kept in reset until the PMIC releases the NRST signal.
   e. RANK5 (1.5 ms): LDO4 (V_{DD3V3_USB}) and LDO5 (V_{DD_FLASH}) are enabled at 3.3 V. The GPO5 (EXT_EN5) is enabled. The 3V3 discrete SMPS regulator is enabled and the 3V3 voltage rises.
   f. Once RANK5 is ended, the PMIC releases the RSTn that releases MPU NRST.
5. Once the NRST is released, the MPU enters in Run mode:
   a. The MPU releases the NRSTC1MS signal.
   b. The D1 CPU starts to execute the boot ROM: The EADLY timer starts. Refer to STM32MP21x internal timer for low power mode management for more information.
   c. Once the EADLY timer elapses, the boot ROM reads, verifies, and executes the bootloader from the external flash memory (eMMC).
6. The bootloader software performs all the initializations, then loads and executes the application software:
   a. The software enables LDO7 (V_{PP_DDR}) at 2.5 V.
   b. The software waits for 1 ms.
   c. The software enables the following once the delay has elapsed:
      i. LDO3 (V_{TT_DDR}) in sink-source mode
      ii. BUCK2 (V_{DD_DDR}) at 1.2 V.
   d. The software waits for 1 ms.
   e. The MPU software initializes the DDR4 controller and DDR memory ICs.
   f. The bootloader loads the application software into DDR4 and executes it and the kernel initializes.
   g. The system is now running.

7. When a shutdown request occurs, the software prepares to power-off properly:
   a. The software shuts down the DDR4 regulators in the following sequence:
      i. Disable in sequence:
         1. BUCK2 (V_{DD_DDR})
         2. LDO3 (V_{TT_DDR})
      ii. The software waits 1 ms.
      iii. Disable LDO7 (V_{PP_DDR}).
   b. The software sends the *SWOFF* command to the PMIC by I²C to trigger a turn-off condition .
8. The PMIC performs a power-down sequence:
   a. The PMIC asserts the RSTn, asserting the MPU NRST signal.
   b. RANK0 (1.5 ms): If not disabled by software, BUCK2 (V_{DD_DDR}), LDO3 (V_{TT_DDR}), and LDO7 (V_{PP_DDR}) are disabled.
   c. RANK5 (1.5 ms): LDO4 (V_{DD3V3_USB}), LDO5 (V_{DD_FLASH}), and GPO5 (EXT_EN5) are disabled. The 3V3 discrete SMPS is disabled and the 3V3 voltage falls.
   d. RANK4 (1.5 ms): BUCK3 (V_{DDCPU}) is disabled.
   e. RANK3 (1.5 ms): LDO6 (1V8) is disabled.
   f. RANK2 (1.5 ms): BUCK1 (V_{DDCORE}) is disabled. Once the V_{DDCORE} voltage drops below MPU $V_{RDY\_VDDCORE}$ falling threshold, the MPU PWR_CPU_ON signals go low.
   g. RANK1 (1.5 ms): LDO2 (V_{DDIO}) and LDO1 (V_{DDA1V8_AON}) are disabled . Once the V_{DDIO} drops below the MPU V_{PDR} threshold or the V_{DDA1V8_AON} drops below V_{PDR_ANA}threshold, the MPU enters in power-on reset mode. The PWR_ON and the PWR_CPU_ON I/Os go in high-Z pull-down.
9. The PMIC is now in OFF mode: the application is powered off.

1. *For the STPMIC2LA the V_{INOK_rise} is typically 4 V.*

Figure 3. Power up and power-down sequence of the MPU with STPMIC2LA



PWR_ON and PWR_CPU_ON signals behavior during the power-up sequence

During the power-up sequence, the PMIC PWRCTRL1/2 internal pull-up resistors, and the MPU PWR_ON and the MPU PWR_CPU_ON internal pull-down resistors induce a particular artifact on the PWR_ON and PWR_CPU_ON signals of the application. See Figure 4 for details.

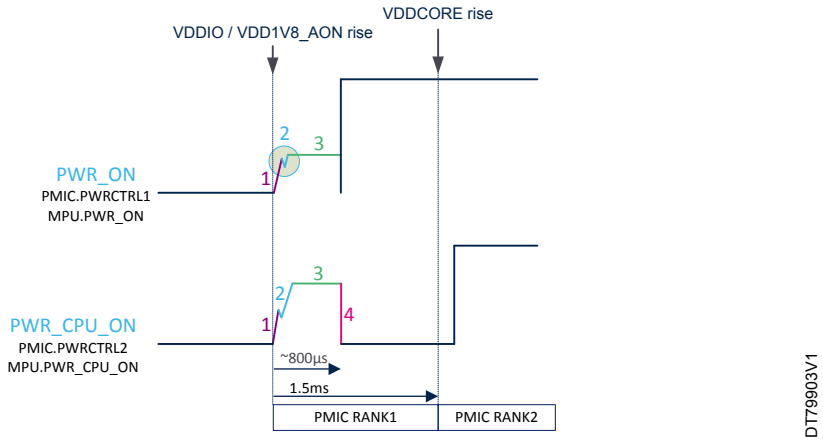
This artifact has no impact on the system behavior as PWRCTRLx signals are only probed by PMIC once the application software is initialized and the PMIC PWRCTRLx signals are allocated to regulators.

Note: The following sequence is to be read with Figure 4. The numbers in the list correspond to the number in Figure 4

1. The PMIC PWRCTRL1 and PWRCTRL2 internal pull-ups (80 kΩ typical) are active until the V_{DDIO} and V_{DDA1V8_AON} start.
2. After few micro seconds, the MPU PWR_ON/PWR_CPU_ON I/Os are in high-Z and internal pull-down resistor (40 kΩ typical) is activated on both signals. A resistor divider is formed by the two 80 kΩ pull-up resistors in the PMIC and 40 kΩ pull-down in the MPU. This resistor divider applies to the PWR_ON and PWR_CPU_ON application signals respectively. Both voltages follow the V_{DDIO} rising voltage driven by PMIC PWRCTRLx pull-up resistors.
3. During approximately 800 µs (V_{DDIO} rise time, together with $t_{RSTTEMPO}$ and internal MPU initialization), the MPU internal pull-down and the PMIC internal pull-up are still active, so the V_{DDIO} voltage is divided by the resistor divider ratio.
4. The MPU internal resistors on PWR_ON and PWR_CPU_ON are deactivated and MPU PWR_ON and PWR_CPU_ON pads are internally set in push-pull mode. PWR_ON and PWR_CPU_ON signals are immediately driven by the MPU to the expected level: PWR_ON goes high and PWR_CPU_ON is kept low until the V_{DDCORE} raises.

Note: PMIC PWRCTRLx pull-up resistors remain active and must be disabled by the bootloader software, after each the application powered-up to avoid extra power consumption in low power mode. This is when the PWR_ON and/or PWR_CPU_ON level is low.

Figure 4. Application PWR_ON and PWR_CPU_ON behavior during the power-up sequence



PWRCTRLs behavior during the power down sequence

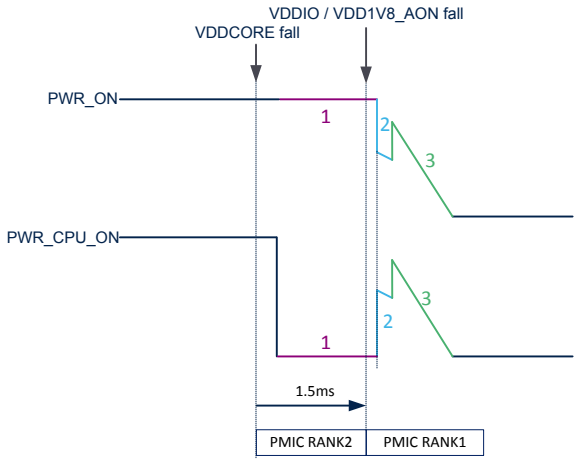
During the power down sequence, the PMIC PWRCTRL1/2 internal pull-up resistor and the MPU PWR_ON and PWR_CPU_ON internal pull-down resistors induce a particular artifact on the PWR_ON and PWR_CPU_ON signals of the application. The whole process is illustrated in Figure 5.

Note: This artifact has no impact on the system behavior as PWRCTRLx signals are not probed by PMIC until the application software initializes and allocates the PMIC PWRCTRLx signals to regulators.

The following sequence is to be read with Figure 5. The numbers in the list correspond to the number in Figure 5.

1. Once the V_{DDCORE} voltage drops below $V_{RDY\_VDDCORE}$ falling threshold, the MPU PWR_CPU_ON is deactivated and the MPU PWR_ON remains active. MPU PWR_ON and PWR_CPU_ON pads are kept in push-pull mode.
2. Once V_{DDIO} drops below the V_{PDR} threshold, or V_{DDA1V8_AON} drops below the V_{PDR_ANA} threshold, the MPU PWR_ON and PWR_CPU_ON I/O pads go in Hi-Z driven with internal pull-down resistor (40 kΩ typ.). The PMIC PWRCTRL1 and PWRCTRL2 pad internal pull-up resistors (80 kΩ typ) are still present. A resistor divider is formed of two 80 kΩ pull-up resistors in the PMIC and 40 kΩ pull-down in the MPU and applies to the respective PWR_ON and PWR_CPU_ON application signals. Both voltages follow V_{DDIO} falling voltage driven by the PMIC PWRCTRLx pull-up resistors.
3. In this state, the PMIC pull-up resistors are still present, but the MPU pull-down resistors are deactivated. Both PWR_ON and PWR_CPU_ON voltages follow V_{DDIO} falling voltage driven by PMIC PWRCTRLx pull-up resistors.

Figure 5. Application PWR_ON and PWR_CPU_ON behavior during power down sequence



5.2.2 Power-down triggered by fast V_{IN} drop

The application in Figure 1 is initially running as illustrated in step 6 of Figure 3. The main supply voltage V_{IN} is removed, causing a drop in V_{IN}. The drop in V_{IN} triggers a PMIC turn off condition (V_{INOK_fall}), as a result the PMIC starts a power-down sequence. Then V_{IN} voltage goes below PMIC V_{INPOR_fall} threshold. As the PMIC is in internal reset, all power supplies drop down in an uncontrolled sequence. Then the main supply voltage is connected back restarting the system once again.

The above sequence is detailed below and illustrated in Figure 6:

1. The application is initially running with a valid main supply voltage (V_{IN})
2. The main supply is removed and V_{IN} starts to drop
3. Once the V_{IN} voltage is below PMIC VINOK_fall threshold (STPMIC2LA VINOK_fall = 3.5V), a turn-off condition occurs and the PMIC starts the power-down sequence:
   a. The PMIC asserts the RSTn, asserting the MPU NRST signal.
   b. RANK0 (1.5 ms): The following regulators are disabled:
      - LDO7 (V_{PP_DDR})
      - BUCK2 (V_{DD_DDR})
      - LDO3 (V_{TT_DDR})
   c. RANK5 (1.5 ms): LDO4 (V_{DD3V3_USB}), LDO5 (V_{DD_FLASH}) and GPO5 (EXT_EN5) are disabled. The 3V3 discrete SMPS is disabled and the 3V3 voltage falls.

Note: The V_{IN} dropping slew rate depends on total decoupling capacitance on V_{IN} and system activity when the main supply is removed. Consequently, the PMIC may enter step4 prior the RANK0 ends.

4. Once the V_{IN} voltage drops below PMIC V_{INPOR_Fall} threshold (STPMIC2LA V_{INPOR_Fall} = 2.1 V), the PMIC enters internal reset. The power-down sequence is stopped, all regulators are disabled with all regulators output discharge disabled, and the PMIC releases the NRST signal. The system enters in an uncontrolled power-down sequence. All power domains voltages drop with uncontrolled slew rate (depends on load and decoupling capacitors on related power domain):
   a. Once the PMIC releases the NRST signal, the NRST signal rises following V_{DDIO} voltage as both V_{DDIO} and V_{DDA1V8_AON} voltage are above respectively MPU V_{PDR} and V_{PDR_ANA} falling thresholds.
   b. The NRSTC1MS signal follows NRST signal.
   c. Once the V_{DDCORE} voltage is below MPU $V_{RDY\_VDDCORE}$ falling threshold, the MPU PWR_CPU_ON signal goes low and the MPU NRSTC1MS follows the MPU PWR_CPU_ON signal.
   d. Once the V_{DDIO} voltage is below MPU V_{PDR} threshold or the V_{DDA1V8_AON} is below V_{PDR_ANA} threshold, the MPU enters in power-on reset mode:
      - The MPU assert the NRST signal and the NRST signal goes low
      - The NRSTC1MS goes floating
      - The PWR_ON and the PWR_CPU_ON I/Os go in high-Z pull-down. The PWR_ON signal goes low and the NRSTC1MS is tied by a voltage divider between internal pull-down and external pull resistor connected to PWR_CPU_ON signal.
      - Once V_{DDIO} is below approximately 1 V, the MPU NRST, NRSTC1MS, PWR_ON, PWR_CPU_ON have uncontrolled voltage.
5. As soon as a power supply is connected to the application, the V_{IN} voltage rises. Once V_{IN} voltage is above V_{INPOR_Rise} (STPMIC2LA V_{INPOR_Rise} = 2.3V typ.) threshold:
   a. The PMIC goes to the *INIT&LOAD* transitional state to preload its NVM content and checks its integrity. If the PMIC NVM integrity is valid, the PMIC initializes launches and executes the VIN_DLY = 10 ms.
   b. Once VIN_DLY elapsed, the PMIC goes in the *CHECK&LOAD* transitional state; as AUTO_TURN_ON bit is set in the STPMIC2LA NVM.

Note: The PMIC CHECK&LOAD has a typical duration of approximately 7 ms.

   c. If the PMIC NVM integrity is valid, the PMIC goes into the *CHECK&LOAD* state, as the AUTO_TURN_ON bit is set in the STPMIC2LA NVM.
   d. Once the PMIC enters in the *CHECK&LOAD* state, output discharge is enabled on all regulators . This allows the discharge of any remaining voltage on all regulators output before the PMIC starts the power-up sequence.

Note: The PMIC INIT&LOAD and CHECK&LOAD have a typical duration of approximately 6 ms.

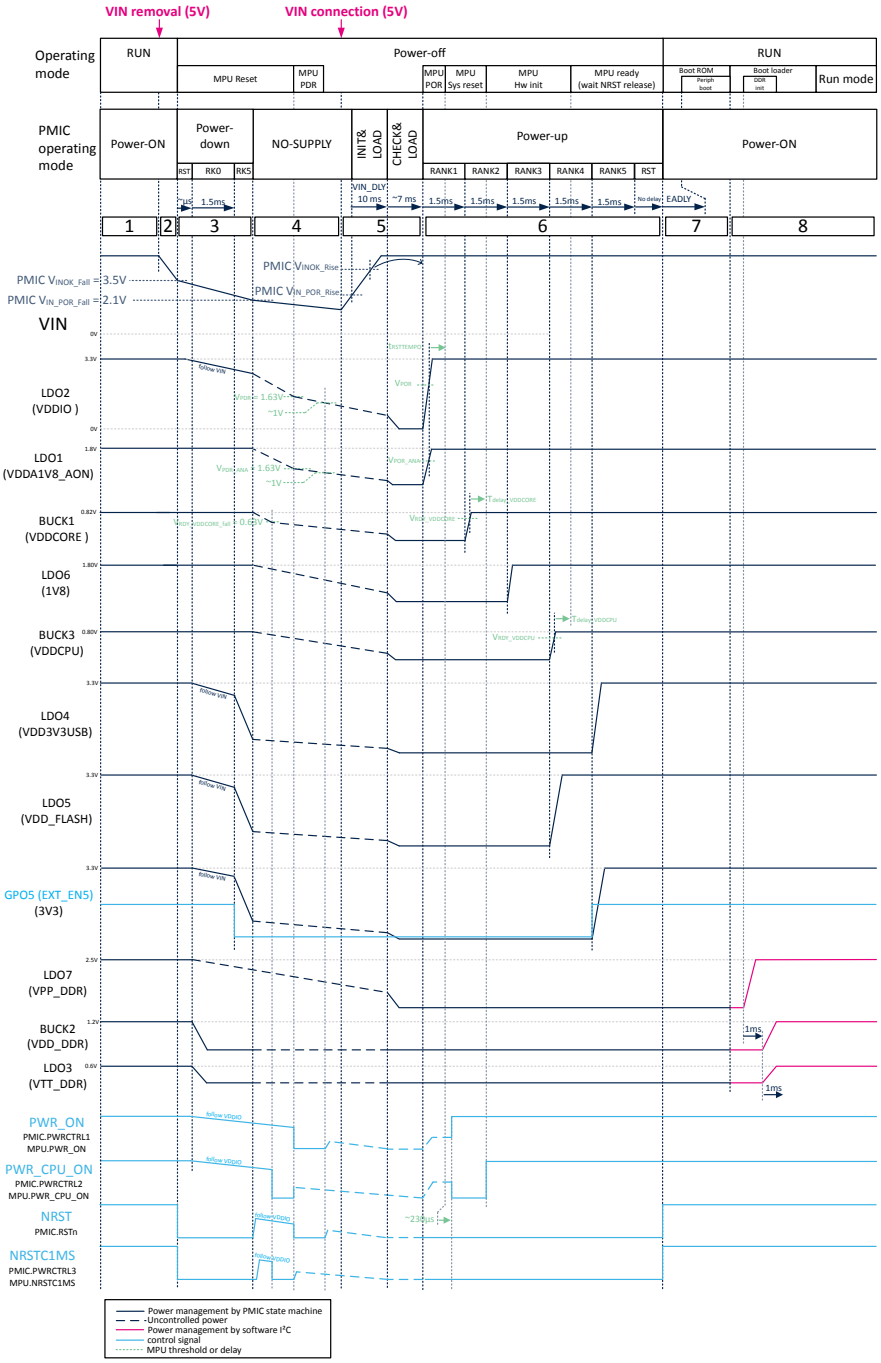
6. Once the *CHECK&LOAD* state ends and the V_{IN} supply rises above V_{INOK_rise} (STPMIC2LA V_{INOK_rise} = 4 V typ.), the PMIC starts a power-up sequence. The PMIC regulators follow the power-up sequence predefined in its NVM:
   a. RANK1 (1.5 ms): The LDO2 (V_{DDIO}) is enabled at 3.3 V and the LDO1 (V_{DDA1V8_AON}) is enabled at 1.8 V. Once V_{DDIO} voltage rises above MPU V_{POR} threshold (1.67 V), and V_{DDA1V8_AON} voltage rises above MPU V_{POR_ANA} threshold (1.67 V), the MPU $t_{RSTTEMPO}$ (440 µs typ.) delay is applied. Once $t_{RSTTEMPO}$ elapses, the MPU enters system reset. After approximately 230 µs, time needed to ensure internal initialization, the MPU release internal nrst_por. The MPU then goes into system reset and the PWR_ON signal goes high.

Note: NRST signal is kept low as the PMIC asserts the NRST signal until the end of the power-up sequence

   b. RANK2 (1.5 ms): The BUCK1 (V_{DDCORE}) is enabled at 0.82 V. Once V_{DDCORE} voltage rises above MPU $V_{RDY\_VDDCORE}$ threshold (0.66 V), an MPU $T_{delay\_VDDCORE}$ (400 µs typ.) delay is started to allow V_{DDCORE} voltage to reach minimum operating voltage. Once $T_{delay\_VDDCORE}$ elapses, the MPU starts hardware initialization, such as starting the HSI oscillator and so on. The PWR_CPU_ON signal goes high.
   c. RANK3 (1.5 ms): The LDO6 (1V8) is enabled at 1.8 V
   d. RANK4 (1.5 ms): The BUCK3 (V_{DDCPU}) is enabled at 0.80 V. Once V_{DDCPU} voltage rises above MPU V_{RDY_VDDCPU} threshold (0.66 V), an MPU T_{delay_VDDCPU} (400 µs typ.) delay is started to allow V_{DDCPU} voltage to reach minimum operating voltage. Once T_{delay_VDDCPU} elapses, the MPU is ready to boot, and it keeps in reset until the PMIC releases the NRST signal.
   e. RANK5 (1.5 ms): LDO4 (V_{DD3V3_USB}) and LDO5 (V_{DD_FLASH}) are enabled at 3.3 V. The GPO5 (EXT_EN5) is enabled. The 3V3 discrete SMPS regulator is enabled and the 3V3 voltage rises.
   f. Once RANK5 is ended, the PMIC releases the RSTn that releases MPU NRST.

7. Once the NRST is released, the MPU enters in Run mode:
   a. The MPU releases the NRSTC1MS signal.
   b. The D1 CPU starts to execute the boot ROM: EADLY timer starts. Refer to EADLY timer for more information.
   c. Once the EADLY timer elapses, the boot ROM reads, verifies, and executes the bootloader from the external flash memory (eMMC).
8. The bootloader software performs all the initializations, then loads and executes the application software:
   a. The software enable LDO7 (VPP_DDR) at 2.5 V.
   b. The software waits for 1 ms.
   c. The software enables the following once the delay has elapsed:
      - LDO3 (V_{TT_DDR}) in sink-source mode
      - BUCK2 (V_{DD_DDR}) at 1.2 V
   d. The software waits for 1 ms.
   e. The MPU software initializes the DDR4 controller and DDR memory ICs.
   f. The bootloader loads the application software into DDR4 and executes it. The kernel initializes.
   g. The system is now running.

Figure 6. Uncontrolled power down sequence



5.3 Low power mode management

The MPU supports several low-power modes to reduce power consumption as detailed in [4]. These are described in Section 5.1: Operating modes. The modes supported by the application and their advantages/ disadvantages are presented in the table below:

Table 8. Low power mode supported by the application

Power mode	Advantages	Disadvantages
LP-Stop1 mode	DDR termination (VTT) is shut down to reduce power consumption Very fast recovery from LP-Stop1 mode to Run mode	Low power consumption gain
LP-Stop2 mode	V_{DDCPU} (D1) is OFF, so power consumption due to D1 current-leakage is saved. Lower power consumption than LP-Stop1 mode.	Longer exit recovery duration than LP-Stop1 mode.
LPLV-Stop1 mode	V_{DDCORE} (D2) voltage is lowered. The D2 domain consumption is reduced.	Few EXTI wake-up sources are available to exit this mode. To exit this mode, more time is needed to restore the lowered supply to its nominal values.
LPLV-Stop2 mode	V_{DDCORE} voltage is lowered and V_{DDCPU} is shut down, the D2 domain consumption is reduced, and the D1 domain current leakage power consumption is saved.	Few EXTI wake-up sources are available to exit this mode. Longer exit recovery duration than LPLV-Stop1 mode.
Standby mode (DDR4 in self-refresh)	Very-low power consumption All MPU power domains are powered off except V_{DDIO} and V_{DDA1V8_AON}. DDR is maintained in self-refresh (suspend to RAM).	Few EXTI wake-up sources are available to exit this mode. Longer exit recovery duration than LPLV-Stop2 mode.
Standby mode (DDR4 OFF)	Lowest power consumption All MPU power domains are powered off except V_{DDIO} and V_{DDA1V8_AON}. DDR is powered off (suspend to flash)	Few EXTI wake-up sources are available to exit this mode. Longer exit recovery duration than Standby mode (DDR4 in self-refresh).

The MPU manages the low-power modes. As described in PWRCTRL1, PWRCTRL2, PWRCTRL3, the power control signals are connected as defined in the table below.

Table 9. Power control signal connection

MPU output	PMIC input
PWR_ON	PWRCTRL1
PWR_CPU_ON	PWRCTRL2
NRSTC1MS	PWRCTRL3

These signals control the regulators directly from the MPU state machine. This is because, in low power mode, no software is running and the PMIC regulators cannot be controlled by any I²C command from software (see Table 5).

Before entering into low-power mode, the MPU software must prepare the PMIC to enter any of these power modes by setting:

- The PMIC xxxx_MAIN_CR registers with the Run mode behavior.
- The PMIC xxxx_ALT_CR registers with the targeted low-power mode behavior.

Note: The term "xxxx" corresponds to the targeted regulator.

The MPU software must also set some internal delays used in the following low power modes:

- LP-Stop1 mode
- LP-Stop2 mode
- LPLV-Stop1 mode
- LPLV-Stop2 mode
- Standby mode

The delays are described in the following section.

5.3.1 STM32MP21x internal timer for low power mode management

EADLY timer

The EADLY timer is a programmable timer used to produce a sufficient delay to ensure that external flash is available for the boot ROM to read the content (eMMC, FMC-NAND, OCTOSPI, SD-Card). This ensures that the boot ROM can reliably read the boot software from the flash boot memory. The EADLY timer duration is set to 5 ms by default after a system reset. It is recommended to keep this default value.

POPL timers

POPL timers are programmable timers used to force the MPU into Standby mode for a minimum duration. When entering in Standby mode, PWR_ON and/or PWR_CPU_ON signals go low for minimum POPL duration forcing peripheral power supply voltages to drop before low power mode exit. This is to ensure MPU peripherals to restart properly if a wake-up event occurs just after MPU enters low power mode. The MPU embeds two POPL timers:

- POPL_D1 linked to the PWR_CPU_ON signal of the MPU. The POPL_D1 defines the minimum duration of PWR_CPU_ON low pulse in the following low power mode:
  - Run2
  - LP-Stop2 mode
  - LPLV-Stop2 mode
  - Standby mode
- POPL_D2 linked to the PWR_ON signal of the MPU. The POPL_D2 defines the minimum duration of PWR_ON low pulse in Standby mode. This delay is not reset by: wake-up from Standby mode, nor MPU reset (NRST, watchdog).

The software sets the POPL timers prior to low power mode entry. Recommended values are proposed in the next sections depending on the low power mode needed.

PODH_D2 timer

The PODH_D2 is a programmable timer used to force the PWR_ON signal high while the PWR_CPU_ON goes low; when entering Standby mode to switch off V_{DDCPU} before V_{DDCORE}. The PODH_D2 timer setting is not reset by wake-up from Standby mode, nor the MPU reset (NRST, watchdog).

The software must set the PODH_D2 timer prior to entering Standby mode. Recommended values are proposed in Section 5.3.4: Standby mode (DDR4 in self-refresh) and Section 5.3.5: Standby mode (DDR4 OFF).

Note: The PODH_D2 timer must override the POPL_D1 timer.

LPLVDLY_D2 timer

The LPLVDLY_D2 is a programmable timer used at LPLV-Stop2 mode exit to wait for the V_{DDCORE} voltage to recover from the retention voltage (670 mV) to the nominal operating voltage (820 mV).

In LPLV-Stop1 mode, the V_{DDCORE} is lowered to 670 mV, the LPLVDLY_D2 is used to wait for V_{DDCORE} to reach the operating supply level in Run mode (820 mV).

In LPLV-Stop2 mode, the V_{DDCORE} is lowered to 670 mV and the V_{DDCPU} is turned OFF, the LPLVDLY_D2 is used to wait for V_{DDCORE} to reach the operating supply level in Run mode (820 mV). Once this delay is elapsed, V_{DDCPU} and the dedicated PWRCTRL (PWR_CPU_ON) starts to rise.

The LPLVDLY_D2 timer must be set once by the software at 187 µs (PWR_D2CR[LPLVDLY_D2]=0). This value is defined as follows:

- PMIC internal 20 µs delay between PWRCTRL1 rise and V_{DDCORE} regulators state change
- A 150 µs worst case delay of V_{DDCORE} voltage recovery from retention (670 mV) to nominal (820 mV).

So, in worst case, there is a 170 µs total delay from PWRCTRL1 signal rising to V_{DDCORE} recovered at 820 mV.

Tdelay_V_{DDCPU}

Tdelay_V_{DDCPU} is an internal and fixed delay (value defined in [5]). When V_{DDCPU} regulator is enabled and has reached the V_{RDY_VDDCPU} threshold, the Tdelay_V_{DDCPU} is started to wait for V_{DDCPU} to reach the Run mode operating supply voltage level. Once Tdelay_V_{DDCPU} elapsed, the CPU1 (D1 domain) is released from reset and the CPU1 starts to run.

As long as V_{DDCPU} is below V_{RDY_VDDCPU} the CPU remains in reset.

The Tdelay_V_{DDCPU} is used at power-up or when the system exits from LP-Stop2 mode, LPLV-Stop2 mode, or Standby mode.

Tdelay_V_{DDCORE}

Tdelay_V_{DDCORE} is an internal and fixed delay (value defined in [5]). When V_{DDCORE} regulator is enabled and has reached the $V_{RDY\_VDDCORE}$ threshold, the Tdelay_V_{DDCORE} is started to wait for V_{DDCORE} to reach the Run mode operating supply voltage level. Once Tdelay_V_{DDCORE} elapsed, the core domain (D2 domain) is released from reset to run. As long as V_{DDCORE} is below $V_{RDY\_VDDCORE}$ the core domain remains in reset.

The Tdelay_V_{DDCORE} is used at power-up or when the system exits from low-power Standby mode.

PWRLP_TEMPO timer

The PWRLP_TEMPO is a delay between the time when the system exits the LP-Stop1 mode, or LP-Stop2 mode, or LPLV-Stop1 mode, or LPLV-Stop2 mode low power mode and the moment when it is allowed to enable the PLLs. It is then able to provide a clock to CPU1 and CPU2, and enters in Run mode. This delay is linked to the D2 power domain (V_{DDCORE}) and must be set in the RCC_PWRLPDLYCR[PWRLP_DLY[21:0]] register bitfield prior to entering low power mode.

The software must set the PWRLP_DLY timer prior to entering low power mode. Recommended values are proposed in the next section.

C1MSRD timer

The CPU1 mass storage reset device (C1MSRD) is a programmable timer defining the minimum pulse duration of the NRSTC1MS signal when the system exits D1 Standby mode. Typically when the system exit from:

- LP-Stop2 mode to Run1 mode
- LPLV-Stop2 mode to Run1 mode
- Run2 mode to Run1 mode.

This timer is also used in case of a D1 independent reset, typically to recover from a D1 crash to reboot the domain without rebooting the complete system. See Section 4.1.6: V_{DD_FLASH} power domain (3.3 V) and PWRCTRL1, PWRCTRL2, PWRCTRL3 section.

The C1MSRD timer must be set by software at cold boot (boot loader). The recommended value for the C1MSRD is 2 ms as the PMIC has built in regulator discharge circuitry to drop regulator output voltage in less than 1.5 ms after the regulator has been turned off.

MRD timer

The MRD is a programmable timer defining the minimum pulse duration of the NRST system reset signal. This timer is useful when the supply is provided by a discrete power component, so it can be set to 0 with a PMIC.

5.3.2 LP-Stop1/LPLV-Stop1 mode

The LP-Stop1 and the LPLV-Stop1 low power modes produce similar regulator behaviors:

- In LPLV-Stop1, the DDR4 termination resistors supply (V_{TT_DDR}) is turned OFF and the V_{DDCORE} voltage is reduced
- In LP-Stop1, the DDR4 termination resistors supply (V_{TT_DDR}) is turned OFF and the V_{DDCORE} is kept at nominal value.

The LPLV-Stop1 mode has lower power consumption than the LP-Stop1 mode, but it requires additional delay to go from LPLV-Stop1 to Run1 mode to wait for V_{DDCORE} nominal voltage to stabilize.

The following section covers both LP-Stop1 and LPLV-Stop1 mode.

LP-Stop1 mode

The LP-Stop1 mode is described below and is shown in Figure 7 based on to the implementation shown in Figure 1.

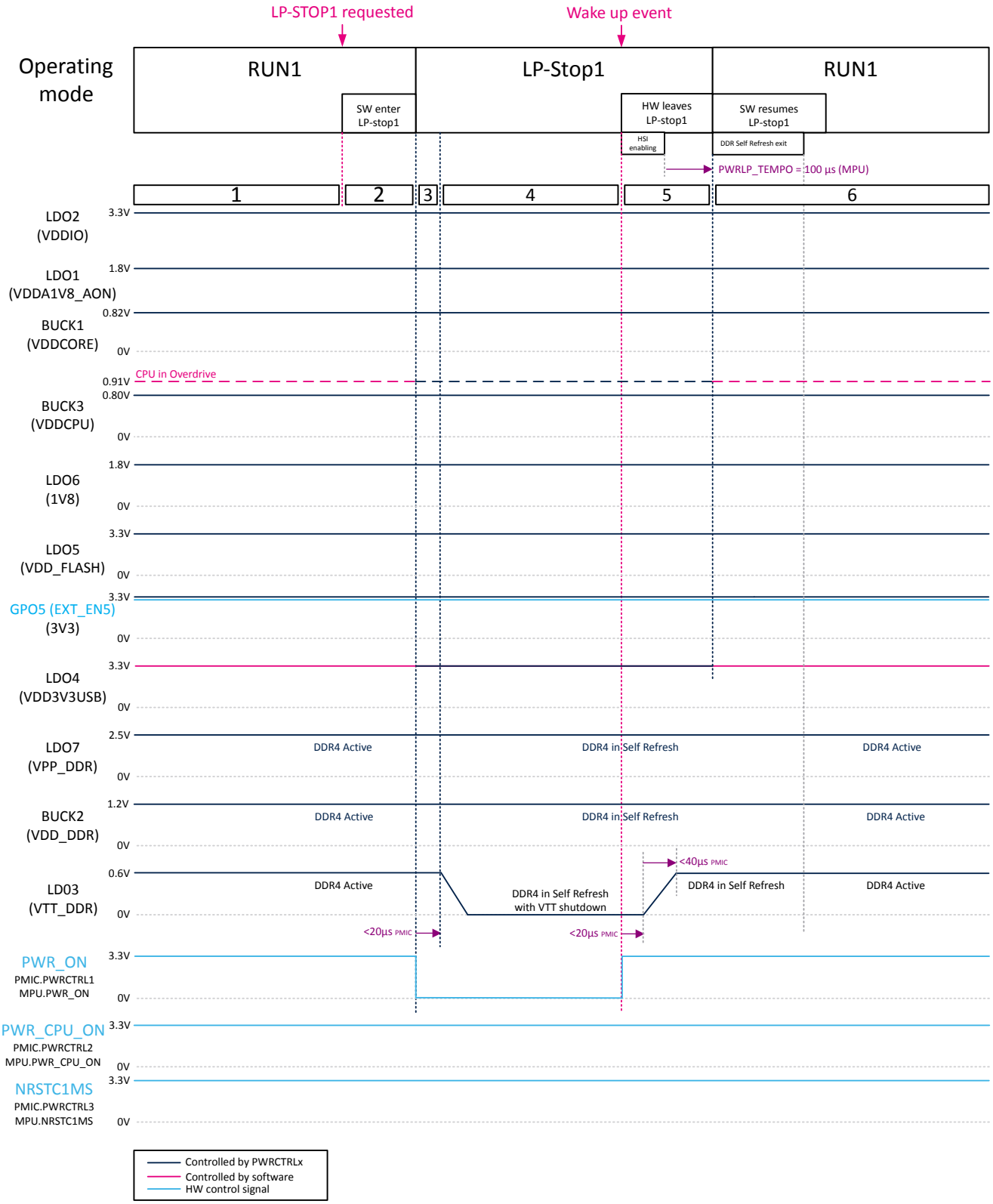
1. The application is powered up and is running in Run1 operating mode; all PWR_ON and PWR_CPU_ON are in high state. In this mode, the PWR_LP signal is internally multiplexed with the PWR_ON pin.
2. When the LP-Stop1 mode is requested, the software prepares to enter LP-Stop1 mode:
   a. The MPU configures the PMIC as defined in Table 10
   b. The MPU performs internal settings such as:
      i. Set PWRLP_TEMPO timer to 100 µs (in RCC_PWRLPDLYCR register)
      ii. Stops all unnecessary system clocks
      iii. Set the DDR4 to self-refresh.
   c. The MPU PWR_CPU2CR[LPDS_D2] and PWR_CPU1CR[LPDS_D1] bits are enabled. This allows the system to enter LP-Stop1 low power mode.
3. Once the system is in LP-Stop1:
   a. The MPU PWR_ON output is deasserted. The PMIC PWRCTRL1 goes low
4. After 20 µs internal PMIC delay, the PMIC regulators, which is connected to the PWR_ON (linked to the PWRCTRL1) takes the configuration set in the xxxx_ALT_CR registers (see Table 10): the V_{TT_DDR} regulator turns OFF.
5. On a wake-up event, the MPU leaves the LP-Stop1 mode:
   a. The MPU PWR_ON output signal is asserted, driving the PMIC PWRCTRL1 signal high.
   b. After 20 µs internal PMIC delay, the PMIC regulators, which is connected to the PWR_ON takes the configuration set in the xxxx_MAIN_CR registers (see Table 10):
      i. The V_{TT_DDR} regulator turns ON
      ii. The V_{TT_DDR} voltage rises in 40 µs.
   c. Clocks are enabled and the software executes the PWRLP_TEMPO timer (100 µs). This delay ensures that the V_{TT_DDR} as reaches its nominal value before:
      i. The system enters in Run1
      ii. The software moves the DDR4 out of self-refresh mode.
6. Once the PWRLP_TEMPO timer has elapsed, the application goes into Run1 mode. The software resumes LP-Stop1: DDR4 exit from self-refresh.

Table 10. PMIC configuration for LP-Stop1 mode

Regulator	PWRCTRLx assignement	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	VOUT	Configuration	VOUT
BUCK1 (V_{DDCORE})	PWR_ON	ON	0.82	ON	0.82
BUCK2 (V_{DD_DDR})	PWR_ON	ON	1.2	ON	1.2
BUCK3 (V_{DDCPU})	PWR_CPU_ON	ON	0.8	OFF	-
LDO1 (V_{DDA1V8_AON})	-	ON	N/A	-	N/A
LDO2 (V_{DDIO})	-	ON	3.3	-	-
LDO3 (V_{TT_DDR})	PWR_ON	SINK SOURCE	-	OFF	-
LDO4 (V_{DD3V3_USB})	-	ON/OFF	N/A	-	N/A
LDO5 (V_{DD_FLASH})	NRSTC1MS(1)	ON/OFF	3.3	-	-
LDO6 (1V8)	PWR_ON	ON	1.8	ON	1.8
LDO7 (V_{PP_DDR})	PWR_ON	ON	2.5	ON	2.5
EXT_EN5 (3V3)	PWR_ON	ON	N/A	ON	N/A

1. *The LDO5 is controlled from PWRCTRL3 in reset mode (PMIC PWRCTRL_RST bit set for LDO5)*

Figure 7. **LP-Stop1 sequence**



LPLV-Stop1 mode

The LPLV-Stop1 mode is described below and is illustrated in the Figure 8 based on to the implementation shown in Figure 1.

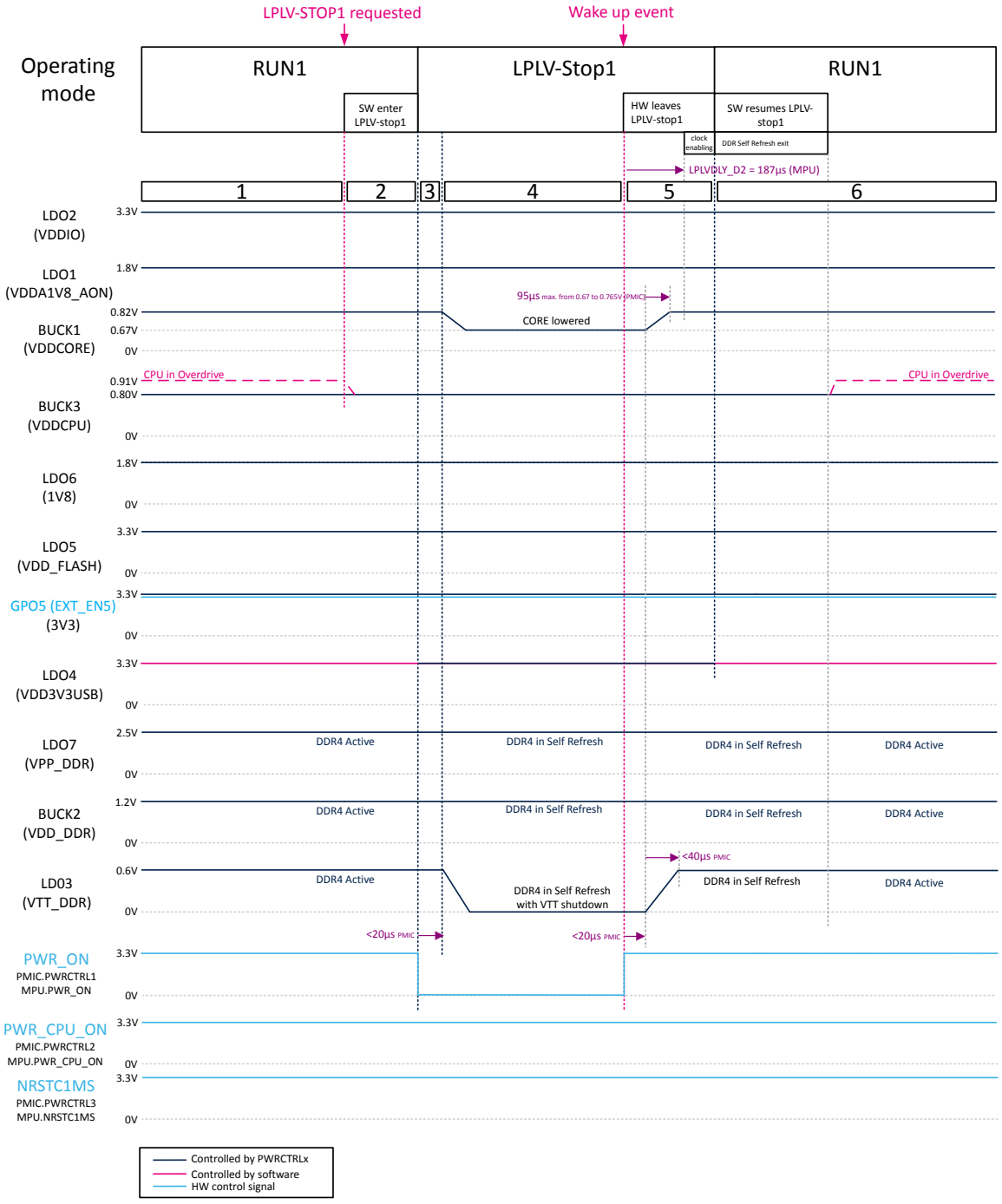
1. The application is powered up and is operating in Run1 mode, all PWR_ON and PWR_CPU_ON are in high state. In this mode, the PWR_LP signal is internally multiplexed with the PWR_ON pin.
2. When the LPLV-Stop1 mode is requested, the software prepares to enter LPLV-Stop1 mode:
   a. The MPU configures the PMIC as defined in Table 11.
   b. If the CPU1 is in overdrive mode, the software sets the CPU1 in nominal mode and sets the V_{DDCPU} to nominal voltage (from 0.91 V to 0.8 V).
   c. The MPU configures the internal settings such as:
      i. Disable the PWRLP_TEMPO timer. This is done by setting RCC_PWRLPDLYCR[PWRLP_DLY[21:0]]=0.
      ii. Set LPLVDLY_D2 timer to 187 µs. This is done by setting PWR_D2CR[LPLVDLY_D2[2:0]=0.
      iii. Stops all unnecessary system clocks
      iv. Set the DDR into self-refresh.
   d. The MPU PWR_CPU2CR[LPDS_D2] and PWR_CPU2CR[LVDS_D2] bit are enabled. This allows the system to enter LPLV-Stop1 low power mode.
3. Once the system is in LPLV-Stop1:
   a. The MPU PWR_ON output is deasserted, and the PMIC PWRCTRL1 goes low.
4. After 20 µs internal PMIC delay, the PMIC regulators assigned to PWR_ON (linked to the PWRCTRL1) takes on the configuration set in the xxxx_ALT_CR registers, which is defined in Table 11:
   a. V_{TT_DDR} regulator is turned OFF.
   b. V_{DDCORE} regulator output decreases to retention voltage (from 820 mV to 670 mV).
5. On a wake-up event, the MPU leaves the LPLV-Stop1 mode:
   a. The MPU PWR_ON output signal is asserted, driving the PMIC PWRCTRL1 signal high and the MPU executes the LPLVDLY_D2 timer to allow the V_{DDCORE} to switch from retention voltage (670 mV) to the minimum operating voltage (765 mV).
   b. After 20 µs internal PMIC delay, the PMIC regulators assigned to PWR_ON take on the configuration set in the xxxx_MAIN_CR registers (see Table 11):
      i. The V_{TT_DDR} regulator is turned ON (V_{TT_DDR} voltage rise in 40 µs)
      ii. The V_{DDCORE} regulator switches from retention voltage (670 mV) to minimum operating voltage (765 mV) in 95 µs. Then it converges to nominal voltage (820 mV).
   c. Once the LPLVDLY_D2 timer elapsed, clocks are enabled.
   d. Once clocks are stable, the MPU goes immediately in Run1 mode (as PWRLP_TEMPO timer is bypassed).
6. The software resumes from LPLV-Stop1, exits DDR4 from self-refresh.

Table 11. PMIC configuration for LPLV-Stop1 mode

Regulator	PWRCTRLx affectation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	VOUT	Configuration	VOUT
BUCK1 (V_{DDCORE})	PWR_ON	ON	0.82	ON	0.67
BUCK2 (V_{DD_DDR})	PWR_ON	ON	1.2	ON	1.2
BUCK3 (V_{DDCPU})	PWR_CPU_ON	ON	0.8	OFF	-
LDO1 (V_{DDA1V8_AON})	-	ON	N/A	-	N/A
LDO2 (V_{DDIO})	-	ON	3.3	-	-
LDO3 (V_{TT_DDR})	PWR_ON	SINK SOURCE	-	OFF	-
LDO4 (V_{DD3V3_USB})	-	ON/OFF	N/A	-	N/A
LDO5 (V_{DD_FLASH})	NRSTC1MS(1)	ON/OFF	3.3	-	-
LDO6 (1V8)	PWR_ON	ON	1.8	ON	1.8
LDO7 (V_{PP_DDR})	PWR_ON	ON	2.5	ON	2.5
EXT_EN5 (3V3)	PWR_ON	ON	N/A	ON	N/A

1. *The LDO5 is controlled from PWRCTRL3 in reset mode (PMIC PWRCTRL_RST bit set for LDO5)*

Figure 8. LPLV-Stop1 sequence



5.3.3 LP-Stop2/LPLV-Stop2 mode

The regulator behavior of both LP-Stop2 mode and LPLV-Stop2 low power modes are similar:

- In LPLV-Stop2, the V_{DDCPU} is shut down and the V_{DDCORE} voltage is reduced.
- In LP-Stop2, the V_{DDCPU} is shut down and the V_{DDCORE} is kept at nominal value.

The LPLV-Stop2 has lower power consumption than the LP-Stop2, but it requires additional time for the device to resume Run1 mode from LPLV-Stop2. This is to allow for V_{DDCORE} to reach its nominal stabilized voltage.

LP-Stop2 mode

The LP-Stop2 mode is described below and is shown in the Figure 9 based on to the implementation shown in Figure 1.

1. The application is powered up and operating in Run1 mode; all PWR_ON and PWR_CPU_ON are in high state. In this mode, the PWR_LP signal is internally multiplexed with the PWR_ON pin.
2. When LP-Stop2 mode is requested, the software prepares to enter LP-Stop2 mode:
   a. The MPU configures the PMIC as described in Table 12.
   b. If the CPU1 is in overdrive mode, the software sets the CPU1 in nominal mode and sets the V_{DDCPU} to nominal voltage (from 0.91 V to 0.8 V).
   c. The MPU configures the internal settings such as:
      i. Set PWR_D1CR[POPL_D1] = 3 ms timer, to define a minimum pulse duration of PWR_CPU_ON. This ensures the full discharge of the V_{DDCPU} voltage before restarting.
      ii. Disable the PWRLP_TEMPO timer (set RCC_PWRLPDLYCR[PWRLP_DLY[21:0]]=0).
      iii. Stops all unnecessary system clocks
      iv. Set the DDR to self-refresh.
   d. The PWR_CPU2CR[LPDS_D2] bit is enabled. This allows the system to enter LP-Stop2 low power mode.
   e. The PWR_CPU1CR[PDDS_D1] bit is enabled. This allows the D1 domain to enter in DStandby, with V_{DDCPU} switched OFF.
3. Once the system is in LP-Stop2:
   a. The MPU PWR_ON is deasserted and the PMIC PWRCTRL1 goes low.
   b. The MPU PWR_CPU_ON is deasserted and the PMIC PWRCTRL2 goes low.
   c. The NRSTC1MS signal follows the PWR_CPU_ON signal (PMIC PWRCTRL3 goes low).
   d. The MPU POPL_D1 timer is started to keep the D1 domain powered off until the POPL_D1 timer has elapsed. The wake-up event is shifted until POPL_D1 has elapsed.
4. After 20 µs internal PMIC delay, the PMIC regulators assigned to PWR_ON (linked to the PWRCTRL1) and PWR_CPU_ON (linked to the PWRCTRL2) and NRSTC1MS (linked to the PWRCTRL3) takes the configuration set in the xxxx_ALT_CR registers (see Table 12):
   a. V_{TT_DDR} regulator is turned OFF
   b. V_{DDCPU} regulator is turned OFF
   c. V_{DD_FLASH} regulator is turned OFF

5. On a wake-up event, the MPU leaves the LP-Stop2 mode:
   a. The MPU PWR_ON signal is asserted (PMIC PWRCTRL1 goes high) and the clock are enabled.
   b. After a 20 µs internal PMIC delay, the PMIC regulators assigned to the PWR_ON takes the configuration set in the xxxx_MAIN_CR registers (see Table 12):
      i. The V_{TT_DDR} regulator is turned ON
      ii. The V_{TT_DDR} voltage rises in 40 µs
   c. Once the clocks are stable, the MPU PWR_CPU_ON is immediately asserted as the PWRLP_TEMPO is bypassed. The PMIC PWRCTRL2 follows the PWR_CPU_ON.
   d. The NRSTC1MS signal follows PWR_CPU_ON and the PMIC PWRCTRL3 goes high.
   e. Following a further 20 µs internal PMIC delay, the PMIC regulators assigned to the PWR_CPU_ON and NRSTC1MS take on the configuration set in the xxxx_MAIN_CR registers (see Table 12):
      - V_{DDCPU} regulator is turned ON
      - V_{DD_FLASH} regulator is turned ON
   f. Once the V_{DDCPU} voltage reaches the V_{RDY_VDDCPU} threshold, the internal $Tdelay\_V_{DDCPU}$ is started to wait for V_{DDCPU} to reach its minimum operating voltage.
6. Once the $Tdelay\_V_{DDCPU}$ has elapsed, the CPU1 releases the clock domain to enter in Run1 mode and the system resumes from LP-Stop2. In this case, the Cortex®-A35 (CPU1) is set as master and the Cortex®-M33 (CPU2) follows the Cortex®-A35:
   a. A CPU1 reset occurs, then CPU1 reboots from the boot ROM, which jumps in software present in SYSRAM.
   b. DDR exits from self-refresh by software running in SYSRAM (secure monitor).

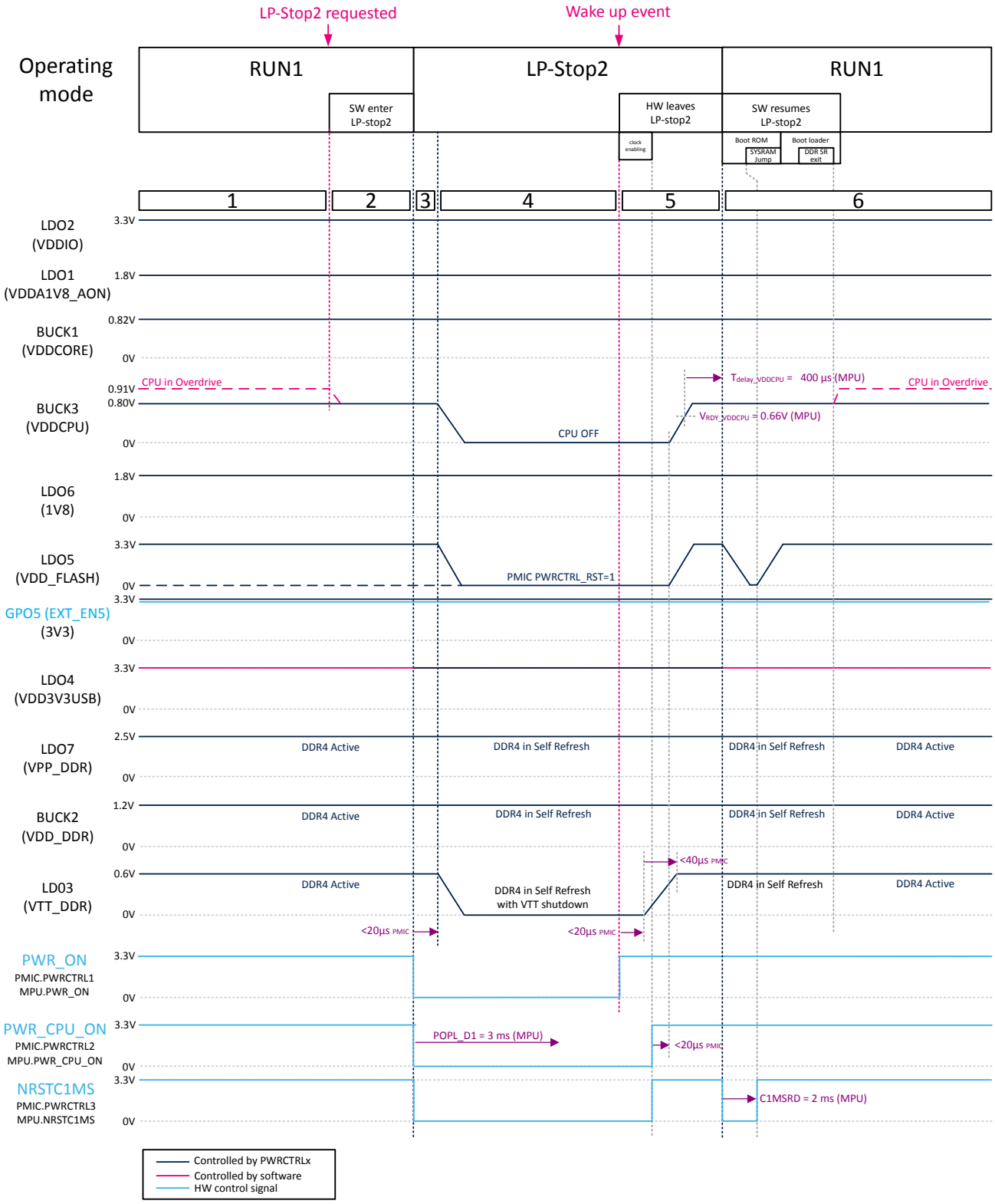
Note: The system enters in Run1 mode only once the CPU1 clocks is released. The Cortex®-M33 is running once the DDR4 exits from self-refresh.

Table 12. PMIC configuration for LP-Stop2

Regulator	PWRCTRLx assignment	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	VOUT	Configuration	VOUT
BUCK1 (V_{DDCORE})	PWR_ON	ON	0.82	ON	0.82
BUCK2 (V_{DD_DDR})	PWR_ON	ON	1.2	ON	1.2
BUCK3 (V_{DDCPU})	PWR_CPU_ON	ON	0.8	OFF	-
LDO1 (V_{DDA1V8_AON})	-	ON	N/A	-	N/A
LDO2 (V_{DDIO})	-	ON	3.3	-	-
LDO3 (V_{TT_DDR})	PWR_ON	SINK SOURCE	-	OFF	-
LDO4 (V_{DD3V3_USB})	-	ON/OFF	N/A	-	N/A
LDO5 (V_{DD_FLASH})	NRSTC1MS(1)	ON	3.3	-	-
LDO6 (1V8)	PWR_ON	ON	1.8	ON	1.8
LDO7 (V_{PP_DDR})	PWR_ON	ON	2.5	ON	2.5
EXT_EN5 (3V3)	PWR_ON	ON	N/A	ON	N/A

1. *The LDO5 is controlled from PWRCTRL3 in reset mode (PMIC PWRCTRL_RST bit set for LDO5)*

Figure 9. LP-Stop2 sequence



LPLV-Stop2 mode

This section focuses on LPLV-Stop2 mode. The LPLV-Stop2 mode is shown in the Figure 10 based on to the implementation shown in Figure 1.

1. The application is powered up and operates in Run1 mode; all PWR_ON and PWR_CPU_ON are in high state. In this mode, the PWR_LP signal is internally multiplexed with the PWR_ON pin.
2. When LPLV-Stop2 mode is requested, the software prepares to enter LPLV-Stop2 mode:
   a. The MPU configures the PMIC as described in Table 13.
   b. If the CPU1 is in overdrive mode, the software sets the CPU1 in nominal mode and sets the V_{DDCPU} to nominal voltage (from 0.91 V to 0.8 V).
   c. The MPU performs internal settings such as:
      i. Set PWR_D1CR[POPL_D1] = 3 ms timer, to define a minimum pulse duration of PWR_CPU_ON ensuring full discharge of the V_{DDCPU} voltage before restarting.
      ii. Set the LPLVDLY_D2 timer to 187 µs (set PWR_D2CR[LPLVDLY_D2[2:0]=0).
      iii. Disable PWRLP_TEMPO timer (set RCC_PWRLPDLYCR[PWRLP_DLY]]=0).
      iv. Stops all unnecessary system clocks
      v. Set the DDR to self-refresh.
   d. The PWR_CPU2CR[LPDS_D2] and PWR_CPU2CR[LVDS_D2] bits are enabled. This allows the system to enter into the LPLV-Stop2 low power mode.
   e. The PWR_CPU1CR[PDDS_D1] bit is enabled. This allows the D1 domain to enter in DStandby (V_{DDCPU} switched OFF).
3. Once the system is in LPLV-Stop2:
   a. The MPU PWR_ON is deasserted, and the PMIC PWRCTRL1 signal goes low.
   b. The MPU PWR_CPU_ON is deasserted, and the PMIC PWRCTRL2 signal goes low.
   c. The NRSTC1MS signal follows the PWR_CPU_ON signal, and the PMIC PWRCTRL3 goes low.
   d. The POPL_D1 delay is started to keep the D1 domain powered off until the POPL_D1 timer has elapsed. This means, the wake-up event is shifted until POPL_D1 has elapsed.
4. After 20 µs internal PMIC delay, the PMIC regulators assigned to PWR_ON (linked to the PWRCTRL1) and PWR_CPU_ON (linked to the PWRCTRL2) and NRSTC1MS (linked to the PWRCTRL3) takes the configuration set in the xxxx_ALT_CR registers (see Table 13):
   a. V_{TT_DDR} regulator is turned OFF.
   b. V_{DDCORE} regulator output decreases to retention voltage (from 820 mV to 670 mV).
   c. V_{DDCPU} regulator is turned OFF.
   d. V_{DD_FLASH} regulator is turned OFF.
5. On a wake-up event, the MPU leaves the LPLV-Stop2:
   a. The MPU PWR_ON output signal is asserted driving the PMIC PWRCTRL1 signal high and the MPU executes the LPLVDLY_D2 timer to wait for the V_{DDCORE} to switch from retention voltage (670 mV) to minimum operating voltage (765 mV).
   b. After a 20 µs internal PMIC delay, the PMIC regulators assigned to the PWR_ON takes the configuration set in the xxxx_MAIN_CR registers (see Table 13):
      i. The V_{TT_DDR} regulator is turned ON (V_{TT_DDR} voltage rise in 40 µs)
      ii. The V_{DDCORE} regulator switches from retention voltage (670 mV) to the minimum operating voltage (765 mV) in 95 µs maximum. Then, it converges to nominal voltage (820 mV).
   c. Once the LPLVDLY_D2 timer elapsed, clocks are enabled.
   d. After a further 20 µs internal PMIC delay, the PMIC regulators assigned to the PWR_CPU_ON and NRSTC1MS take on the configuration set in the xxxx_MAIN_CR registers (see Table 13):
      i. V_{DD_FLASH} regulator is turned ON.
      ii. V_{DDCPU} regulator is turned ON.
   e. Once the V_{DDCPU} voltage reaches the V_{RDY_VDDCPU} threshold, the $Tdelay\_V_{DDCPU}$ delay is triggered to ensure V_{DDCPU} reaches its minimum operating voltage.

6. Once Tdelay_V_{DDCPU} is elapsed, CPU1 releases the clock domain to enter Run1 mode, the system resumes from LPLV-Stop2. In this case, the Cortex®-A35 (CPU1) is set as master and the Cortex®-M33 (CPU2) follows the Cortex®-A35:
   a. A CPU1 reset occurs, then CPU1 reboots from the boot ROM, which jumps in software present in SYSRAM.
   b. DDR exits from self-refresh by software running in SYSRAM (secure monitor).

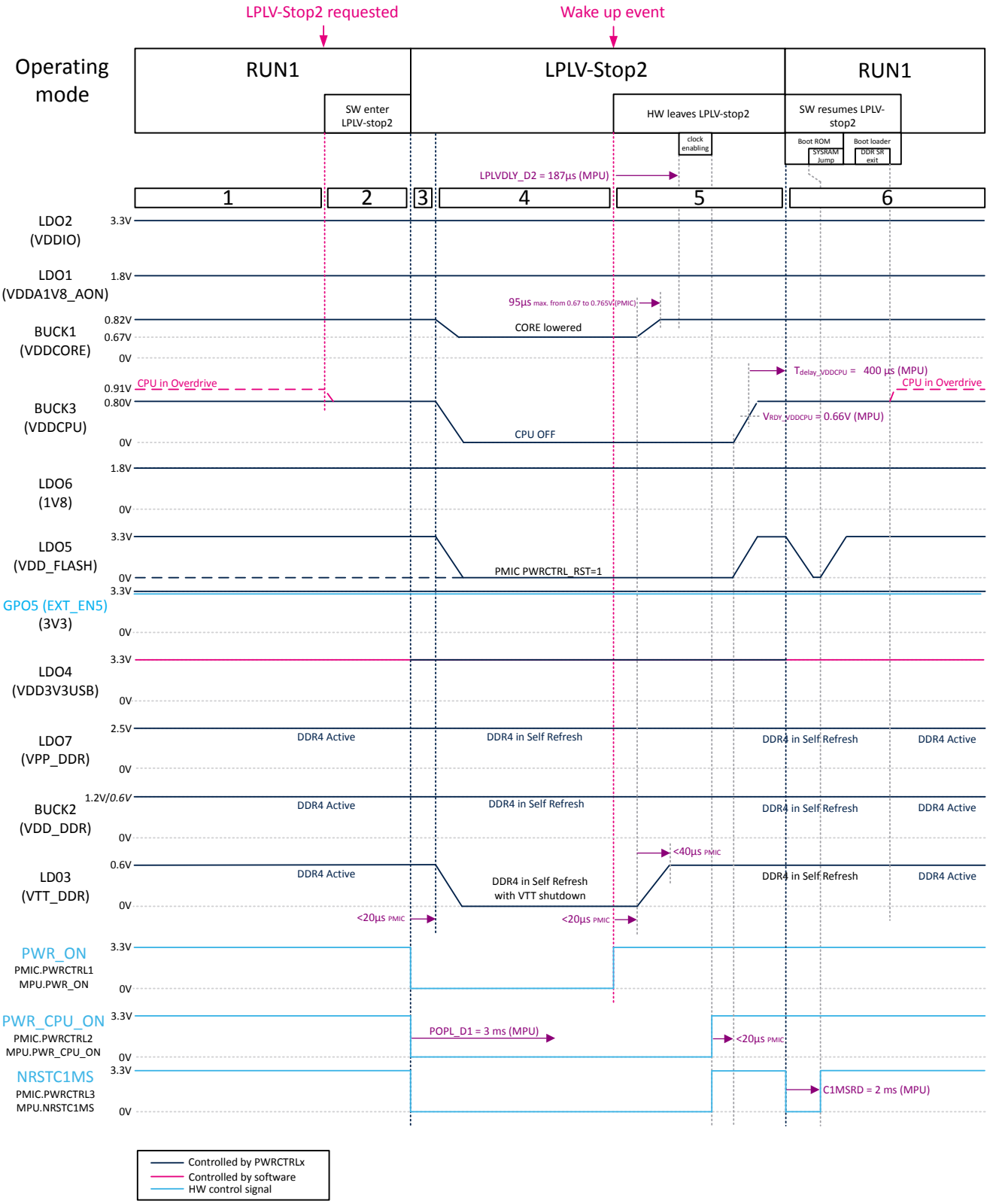
Note: The system enters in Run1 once the CPU1 clocks are released. The Cortex®-M33 is running once the DDR4 exits from self-refresh.

Table 13. PMIC configuration for LPLV-Stop2

Regulator	PWRCTRLx affectation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	VOUT	Configuration	VOUT
BUCK1 (V_{DDCORE})	PWR_ON	ON	0.82	ON	0.67
BUCK2 (V_{DD_DDR})	PWR_ON	ON	1.2	ON	1.2
BUCK3 (V_{DDCPU})	PWR_CPU_ON	ON	0.8	OFF	-
LDO1 (V_{DDA1V8_AON})	-	ON	N/A	-	N/A
LDO2 (V_{DDIO})	-	ON	3.3	-	-
LDO3 (V_{TT_DDR})	PWR_ON	SINK SOURCE	-	OFF	-
LDO4 (V_{DD3V3_USB})	-	ON/OFF	N/A	-	N/A
LDO5 (V_{DD_FLASH})	NRSTC1MS(1)	ON/OFF	3.3	-	-
LDO6 (1V8)	PWR_ON	ON	1.8	ON	1.8
LDO7 (V_{PP_DDR})	PWR_ON	ON	2.5	ON	2.5
EXT_EN5 (3V3)	PWR_ON	ON	N/A	ON	N/A

1. *The LDO5 is controlled from PWRCTRL3 in reset mode (PMIC PWRCTRL_RST bit set for LDO5)*

Figure 10. LPLV-Stop2 sequence



Note: If the wake-up event is generated from one of the WKUP pins or EXTI2, the PWR_CPU_ON and the PWR_ON rise at the same time: the V_{DDCORE} recovers at the same time as the V_{DDCPU} rise. So, the LPLVDLY_D2 timer run in parallel with the $Tdelay\_V_{DDCPU}$ once V_{DDCPU} is higher than V_{RDY_VDDCPU}.

5.3.4 Standby mode (DDR4 in self-refresh)

The Standby mode is used when a very-low power consumption is required. In this mode, the MPU PWRCTRLx signals are pulled low. Most of the PMIC regulators are switched off. The content of MPU registers and memories are lost except for the backup and retentions domains (V_{DDIO} and V_{DDA1V8_AON} are kept enabled). The DDR4 is set in self-refresh (V_{DD_DDR}, and V_{PP_DDR} are kept enabled) to maintain the system in “suspend to RAM state.”

This section focuses on Standby mode with DDR4 in self-refresh. This mode is shown in Figure 11 based on the implementation shown in Figure 1.

1. The application is powered up and operates in Run1 mode. All PWR_ON and PWR_CPU_ON are in high state.
2. When the Standby mode is requested, the software prepares to enter Standby mode:
   a. The MPU configures the PMIC as described in Table 14.
   b. If the CPU1 is in overdrive mode, the software sets the CPU1 in nominal mode and sets the V_{DDCPU} to nominal voltage (from 0.91 V to 0.8 V).
   c. The MPU disables the USB then turns OFF the V_{DD3V3_USB} regulator.
   d. The MPU performs internal settings such as:
      i. Set the PWR_D1CR[POPL_D1] = 3 ms timer, to define a minimum pulse duration of PWR_CPU_ON ensuring full discharge of the V_{DDCPU} voltage before restarting.
      ii. Set the PWR_D2CR[PODH_D2] = 1 ms timer, to turn off V_{DDCPU} before V_{DDCORE}.
      iii. Set the PWR_D2CR[POPL_D2] = 2 ms, to define a minimum pulse duration of PWR_ON ensuring V_{DDCORE} voltage is fully discharged before restarting.
      iv. Stops all unnecessary system clocks.
      v. Set the DDR to self-refresh.
   e. The PWR_CPU2CR[PDDS_D2] bit is enabled. Standby mode is allowed when CPU2 goes OFF.
   f. The PWR_CPU1CR[PDDS_D1] and PWR_CPU1CR[PDDS_D2] bit are enabled. This allows the D1 domain to enter in DStandby (V_{DDCPU} switched OFF) and the Standby mode is allowed when CPU1 goes OFF.
3. Once the system is in Standby mode:
   a. The MPU PWR_CPU_ON is deasserted. The PMIC PWRCTRL2 goes low.
   b. The NRSTC1MS signal follows the PWR_CPU_ON signal (PMIC PWRCTRL3 goes low)
   c. The POPL_D1 timer is started to keep the D1 domain powered off until the POPL_D1 timer has elapsed. The wake-up event is shifted until POPL_D1 has elapsed.
   d. The PODH_D2 timer is started to keep V_{DDCORE} enabled (PWR_ON keeps high) waiting for the V_{DDCPU} voltage to be powered off before V_{DDCORE}.
4. After a 20 µs internal PMIC delay, the PMIC regulators assigned to PWR_CPU_ON (linked to PWRCTRL2) and NRSTC1MS (linked to PWRCTRL3) take on the configuration set in the xxxx_ALT_CR registers (see Table 14):
   a. V_{DDCPU} regulator is turned OFF
   b. V_{DD_FLASH} regulator is turned OFF
5. Once PODH_D2 timer has elapsed:
   a. The MPU PWR _ON is deasserted. The PMIC PWRCTRL1 signal goes low.
   b. The POPL_D2 timer is started to keep the D2 domain powered off until POPL_D2 has elapsed. The wakeup event is shifted until POPL_D2 has elapsed.
6. After a 20 µs internal PMIC delay has elapsed, the PMIC regulators assigned to PWR_ON (linked to PWRCTRL1) take on the configuration set in the xxxx_ALT_CR registers (see Table 14):
   a. V_{DDCORE} regulator is turned OFF
   b. 1V8 regulator is turned OFF
   c. EXT_EN5 (GPO5) goes low, the 3V3 regulator is turned OFF
   d. V_{TT_DDR} regulator is turned OFF

7. On a wake-up event, the MPU leaves the Standby mode:
   a. The MPU PWR_ON signal is asserted (PMIC PWRCTRL1 goes high).
   b. After a 20 µs internal PMIC delay, the PMIC regulators assigned to PWR_ON takes on the configuration set in the xxxx_MAIN_CR registers (see Table 14):
      i. V_{TT_DDR} regulator is turned ON.
      ii. EXT_EN5 (GPO5) signal goes high, the 3V3 regulator is turned ON.
      iii. 1V8 regulator is turned ON.
      iv. V_{DDCORE} regulator is turned ON.
   c. Once the V_{DDCORE} voltage reaches the $V_{RDY\_VDDCORE}$ threshold, the $Tdelay\_V_{DDCORE}$ internal delay is started. This ensures V_{DDCORE} reaches its minimum operating voltage value.
   d. Once the $Tdelay\_V_{DDCORE}$ delay has elapsed, the clocks are enabled and the MPU PWR_CPU_ON is asserted.
   e. The NRSTC1MS signal follows PWR_CPU_ON.
   f. After 20 µs internal PMIC delay, the PMIC regulators assigned to PWR_CPU_ON and NRSTC1MS take the configuration set in the xxxx_MAIN_CR registers (see Table 14):
      i. V_{DD_FLASH} regulator is turned ON
      ii. V_{DDCPU} is turned ON.
   g. Once the V_{DDCPU} voltage reaches the V_{RDY_VDDCPU} threshold, the $Tdelay\_V_{DDCPU}$ internal delay is started. This is to ensure that V_{DDCPU} reaches its minimum operating voltage value.
8. Once $Tdelay\_V_{DDCPU}$ has elapsed, CPU1 releases the clock domain to enter in Run1 mode and the system resumes from the Standby mode. In this case, the Cortex®-A35 (CPU1) is set as master and the Cortex®-M33 (CPU2) follows the Cortex®-A35:
   a. A CPU1 reset occurs, then CPU1 reboots from the boot ROM which jumps to the software present in the SYSRAM.
   b. DDR exits from self-refresh by software running is SYSRAM (secure monitor).

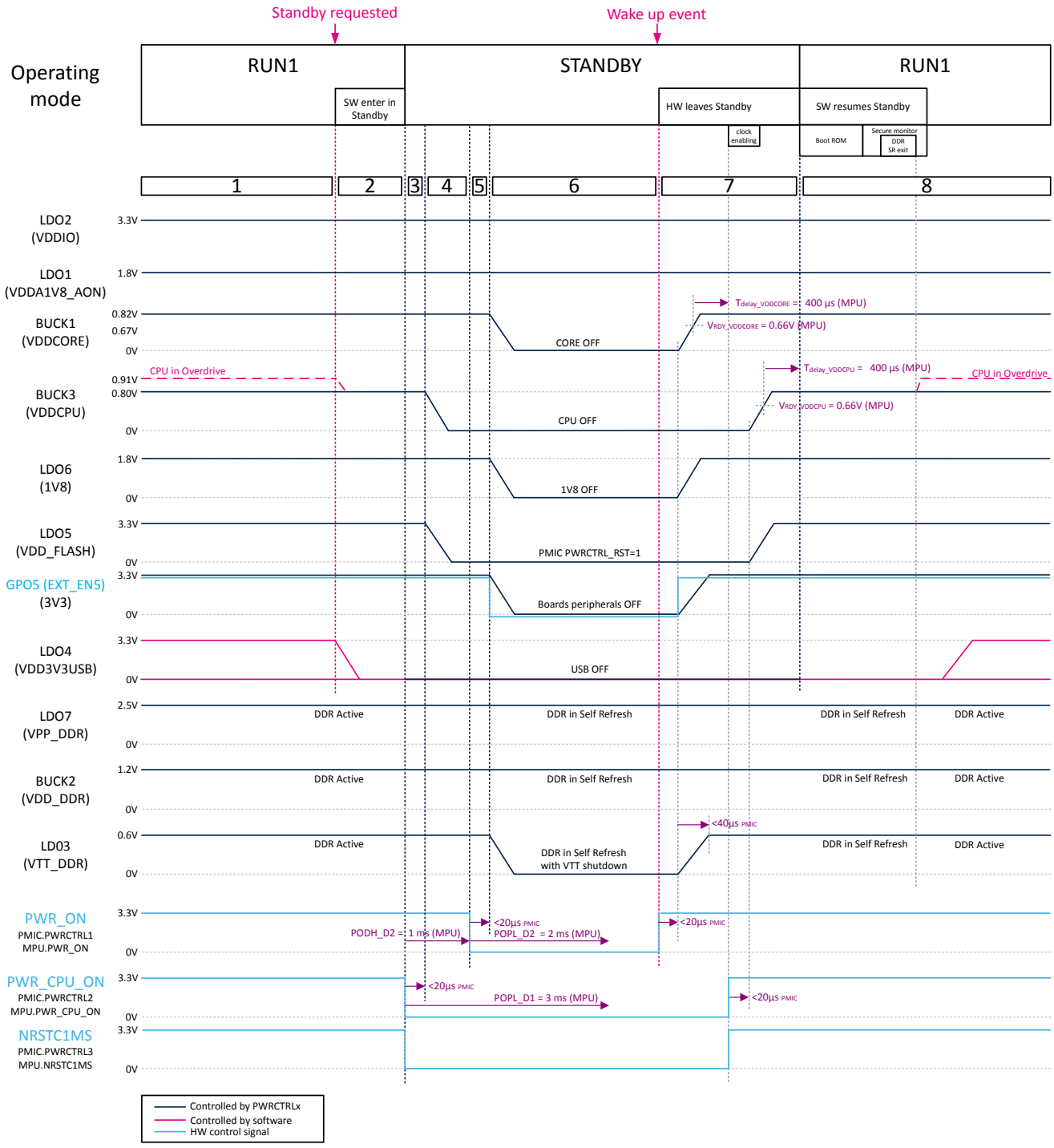
Note: The system enters in Run1 once the CPU1 clocks are released. The Cortex®-M33 is running once the DDR4 exits from self-refresh.

Table 14. PMIC configuration for standby DDR in self-refresh

Regulator	PWRCTRLx assignation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	VOUT	Configuration	VOUT
BUCK1 (V_{DDCORE})	PWR_ON	ON	0.82	OFF	-
BUCK2 (V_{DD_DDR})	PWR_ON	ON	1.2	ON	1.2
BUCK3 (V_{DDCPU})	PWR_CPU_ON	ON	0.8	OFF	-
LDO1 (V_{DDA1V8_AON})	-	ON	N/A	-	N/A
LDO2 (V_{DDIO})	-	ON	3.3	-	-
LDO3 (V_{TT_DDR})	PWR_ON	SINK SOURCE	-	OFF	-
LDO4 (V_{DD3V3_USB})	-	ON/OFF(1)	N/A	-	N/A
LDO5 (V_{DD_FLASH})	NRSTC1MS (2)	ON/OFF	3.3	-	-
LDO6 (1V8)	PWR_ON	ON	1.8	OFF	-
LDO7 (V_{PP_DDR})	PWR_ON	ON	2.5	ON	2.5
EXT_EN5 (3V3)	PWR_ON	ON	N/A	OFF	N/A

1. *V_{DD3V3_USB} must be turned OFF before V_{DDCORE} is turned OFF*
2. *The LDO5 is controlled from PWRCTRL3 in reset mode (PMIC PWRCTRL_RST bit set for LDO5)*

Figure 11. Standby (DDR in self-refresh) sequence



5.3.5 Standby mode (DDR4 OFF)

The Standby mode is used when a very-low power consumption is required. In this mode, the PMIC PWRCTRLx signals are set low. Most of the PMIC regulators are switched off. The content of MPU registers and memories are lost except for the backup and the retention domain (V_{DDIO} and V_{DDA1V8_AON} are kept enabled). The DDR4 is powered off to maintain the system in “suspend to flash state.”

This section focuses on the Standby mode with DDR4 OFF. This mode is described below and is shown in Figure 12 based on the implementation shown in Figure 1.

1. The application is powered up and operating in Run1 mode. All PWR_ON and PWR_CPU_ON are in high state.
2. When the Standby mode is requested, the software prepares to enter Standby mode:
   a. The MPU configures the PMIC as described in Table 15.
   b. The MPU disables the USB then turns OFF the V_{DD3V3_USB} regulator.
   c. If the CPU1 is in overdrive mode, the software sets the CPU1 in nominal mode and sets the V_{DDCPU} to nominal voltage (from 0.91 V to 0.8 V).
   d. The MPU performs internal settings such as:
      i. Set the PWR_D1CR[POPL_D1] = 3 ms timer, to define a minimum pulse duration of PWR_CPU_ON ensuring full discharge of the V_{DDCPU} voltage before restarting.
      ii. Set the PWR_D2CR[PODH_D2] =1 ms timer, to turn off V_{DDCPU} before V_{DDCORE}.
      iii. Set the PWR_D2CR[POPL_D2] = 2 ms timer, to define a minimum pulse duration of PWR_ON ensuring full discharge of the V_{DDCORE} voltage before restarting.
      iv. Stops all unnecessary system clocks
      v. Disable DDR.
      vi. Turns OFF the DDR regulators according to the power down sequence defined in Section 4.1.5: DDR power domain (V_{DD_DDR}, V_{TT_DDR}, V_{PP_DDR}).
   e. The PWR_CPU2CR[PDDS_D2] bit is enabled. Standby mode is allowed when CPU2 is turned OFF.
   f. The PWR_CPU1CR[PDDS_D1] and PWR_CPU1CR[PDDS_D2] bit are enabled. This allows the D1 domain to enter in DStandby (V_{DDCPU} switched OFF), and the Standby mode is allowed when CPU1 is turned off.
3. Once the system is in Standby mode
   a. The MPU PWR_CPU_ON is de-asserted, the PMIC PWRCTRL2 goes low.
   b. The NRSTC1MS signal follows the PWR_CPU_ON signal (PMIC PWRCTRL3 goes low)
   c. The POPL_D1 timer is started to keep the D1 domain powered off until POPL_D1 timer has elapsed (wakeup event is shifted until POPL_D1 has elapsed).
   d. The PODH_D2 timer is started to keep V_{DDCORE} enabled (PWR_ON keeps high) waiting for V_{DDCPU} voltage to be powered off before V_{DDCORE}.
4. After 20 µs internal PMIC delay, the PMIC regulators assigned to PWR_CPU_ON (linked to PWRCTRL2) and NRSTC1MS (linked to PWRCTRL3) takes the configuration set in the xxxx_ALT_CR registers (see Table 15).
   a. V_{DDCPU} regulator is turned OFF
   b. V_{DD_FLASH} regulator is turned OFF
5. Once the PODH_D2 timer has elapsed
   a. The MPU PWR_ON is deasserted (PMIC PWRCTRL1 goes low)
   b. The POPL_D2 delay is started to keep the D2 domain powered off until the POPL_D2 timer has elapsed. The wake-up event is shifted until POPL_D2 has elapsed.
6. After 20 µs internal PMIC delay, the PMIC regulators assigned to PWR_ON (linked to PWRCTRL1) takes the configuration set in the xxxx_ALT_CR registers (see Table 15).
   a. V_{DDCORE} regulator is turned OFF.
   b. 1V8 regulator is turned OFF.
   c. EXT_EN5 (GPO5) goes low: the 3V3 regulator is turned OFF

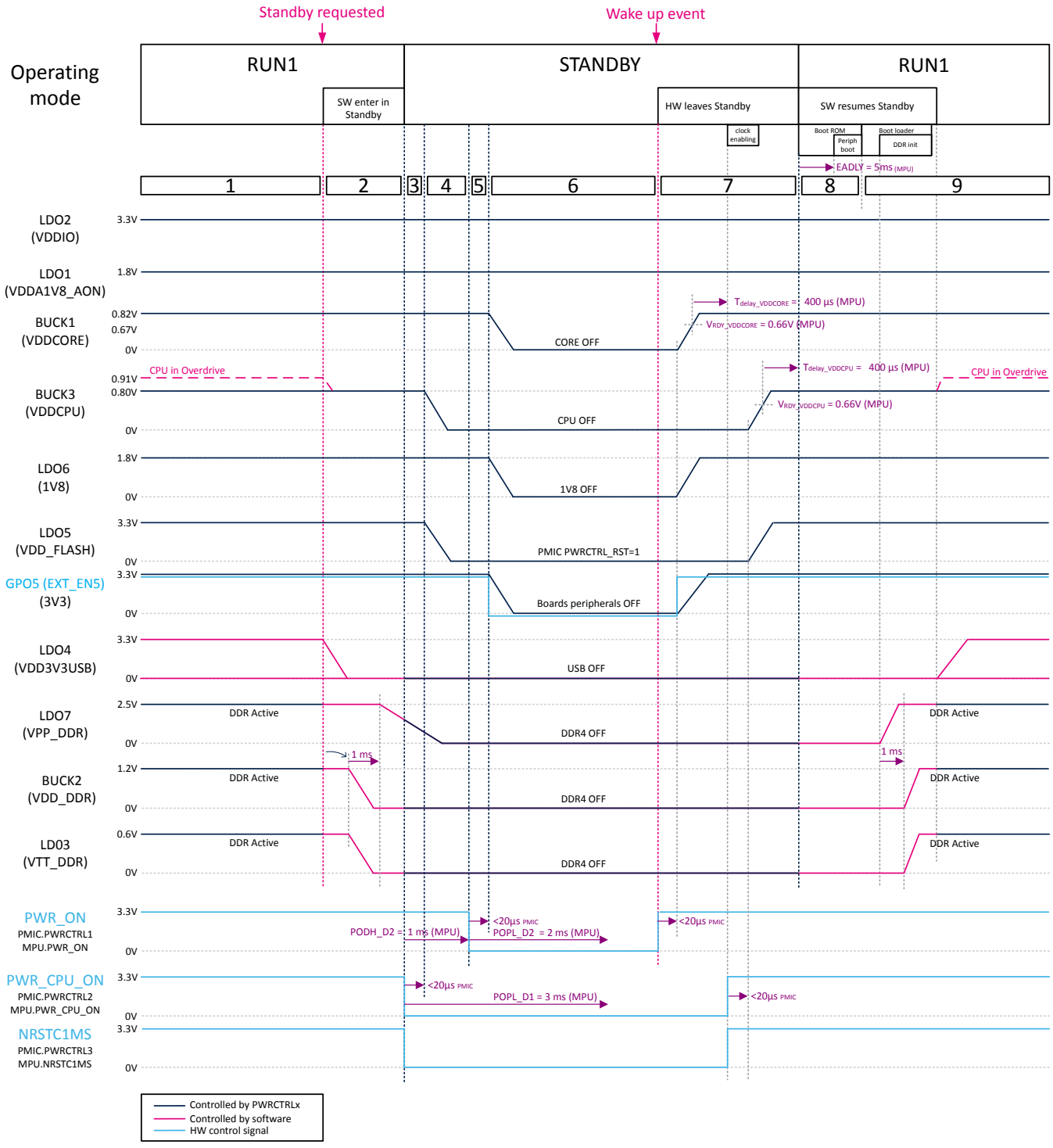
7. On a wake-up event, the MPU leaves the low power mode:
   a. The MPU PWR_ON signal is asserted (PMIC PWRCTRL1 goes high).
   b. After 20 µs internal PMIC delay, the PMIC regulators assigned to PWR_ON takes on the configuration set in the xxxx_MAIN_CR registers (see Table 15):
      - EXT_EN5 (GPO5) goes high: the 3V3 regulator is turned ON
      - 1V8 regulator is turned ON.
      - V_{DDCORE} regulator is turned ON.
   c. Once the V_{DDCORE} voltage reaches the $V_{RDY\_VDDCORE}$ threshold, the Tdelay_V_{DDCORE} internal delay starts to allow the V_{DDCORE} to reach its minimum operating voltage value.
   d. Once the Tdelay_V_{DDCORE} delay is elapsed, the clocks are enabled and the MPU PWR_CPU_ON is asserted. The NRSTC1MS signal follows PWR_CPU_ON
   e. After 20 µs internal PMIC delay, the PMIC regulators assigned to PWR_CPU_ON and NRSTC1MS takes the configuration set in the xxxx_MAIN_CR registers (see Table 15):
      i. V_{DD_FLASH} regulator is turned ON
      ii. V_{DDCPU} regulator is turned ON.
   f. Once the V_{DDCPU} voltage reaches the V_{RDY_VDDCPU} threshold, the T_{delay_VDDCPU} internal delay starts to allow the V_{DDCPU} to reach its nominal minimum operating voltage value.
8. Once Tdelay_V_{DDCPU} is elapsed, CPU1 releases the clock domain to enter in Run1 mode and the system resumes from Standby mode.
   a. The D1 CPU starts to execute the boot ROM, and the EADLY timer starts.
   b. Once the EADLY timer elapses, the boot ROM reads, verifies, and executes the bootloader from the external flash memory (eMMC).
9. The bootloader software performs initializations, then loads and executes the application software:
   a. The bootloader software turns ON the DDR regulators according to the power up sequence defined in Section 4.1.5: DDR power domain (V_{DD_DDR}, V_{TT_DDR}, V_{PP_DDR}).
   b. The bootloader software initializes the DDR4 controller and DDR memory ICs.
   c. The bootloader software resumes Standby mode by loading the application software from eMMC into DDR4 and executes it. The application is resumed, and the system is running.

Table 15. PMIC configuration standby DDR OFF

Regulator	PWRCTRLx assignation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	VOUT	Configuration	VOUT
BUCK1 (V_{DDCORE})	PWR_ON	ON	0.82	OFF	-
BUCK2 (V_{DD_DDR})	PWR_ON	ON(1)	1.2	OFF	-
BUCK3 (VDDCPU)	PWR_CPU_ON	ON	0.8	OFF	-
LDO1 (V_{DDA1V8_AON})	-	ON	N/A	-	N/A
LDO2 (V_{DDIO})	-	ON	3.3	-	-
LDO3 (V_{TT_DDR})	PWR_ON	SINK SOURCE(1)	-	OFF	-
LDO4 (V_{DD3V3_USB})	-	ON/OFF	N/A	-	N/A
LDO5 (V_{DD_FLASH})	NRSTC1MS(3)	ON/OFF	3.3	-	-
LDO6 (1V8)	PWR_ON	ON	1.8	OFF	-
LDO7 (V_{PP_DDR})	PWR_ON	ON(1)	2.5	OFF	-
EXT_EN5 (3V3)	PWR_ON	ON	N/A	OFF	N/A

1. *DDR4 regulators are turned OFF by software prior Standby mode entry*
2. *V_{DD3V3_USB} must be turned OFF before V_{DDCORE} is turned OFF*
3. *The LDO5 is controlled from PWRCTRL3 in reset mode (PMIC PWRCTRL_RST bit set for LDO5)*

Figure 12. Standby (DDR OFF) sequence



5.4 System and CPU1 crash recovery management

The MPU can recover from several levels of crash. A crash could occur on the entire system or only to CPU1. These are described in the following list:

- A system crash: the complete application requires reset (system reset).
- A CPU1 crash: the crash is limited to the Arm® Cortex®-A35 platform (CPU1 is in the D1 domain). So, only the Arm® Cortex®-A35 and associated peripherals need to be reset.

5.4.1 System crash recovery management

As introduced in the RSTn pin section, the MPU, and the PMIC both have interconnected bidirectional low reset pins (see Figure 1 NRST signal).

A system crash occurs when one or several of the following fail:

- MPU D1, or D2 domain crash through watchdogs elapsing:
  - iwdg1_out_rst
  - iwdg2_out_rst
  - iwdg3_out_rst
  - iwdg4_out_rst
- System reset: such as the assertion of NRST by an external source by pressing on the reset button.
- PMIC hard-fault (see Section 6: Safety management)

If a STM32MP21x crash occurs, the MPU generates a reset pulse on the NRST signal.

The reset pulse triggers the PMIC to produce an immediate power cycle sequence. This power cycling is recommended to ensure a correct reset and restart of the peripherals following a global application reset (NRST). This is specifically for peripherals that do not have a reset input signal such as eMMC and so on.

In this application, the power cycling is not performed on the PMIC LDO2 (V_{DDIO}), neither on the LDO1 (V_{DDA1V8_AON}) due to the mask reset option. With this option, the regulators need to be kept enable during reset (see Section 5.1.3: PMIC mask-reset option for details).

The diagram in Figure 13 illustrates a crash recovery or a system reset sequence according to the application shown in Figure 1. For example, the crash is triggered by a IWDG reset occuring in Run mode, or the system reset is triggered by a user pushing the reset button.

Note: An IWDG reset can occur in all low-power modes.

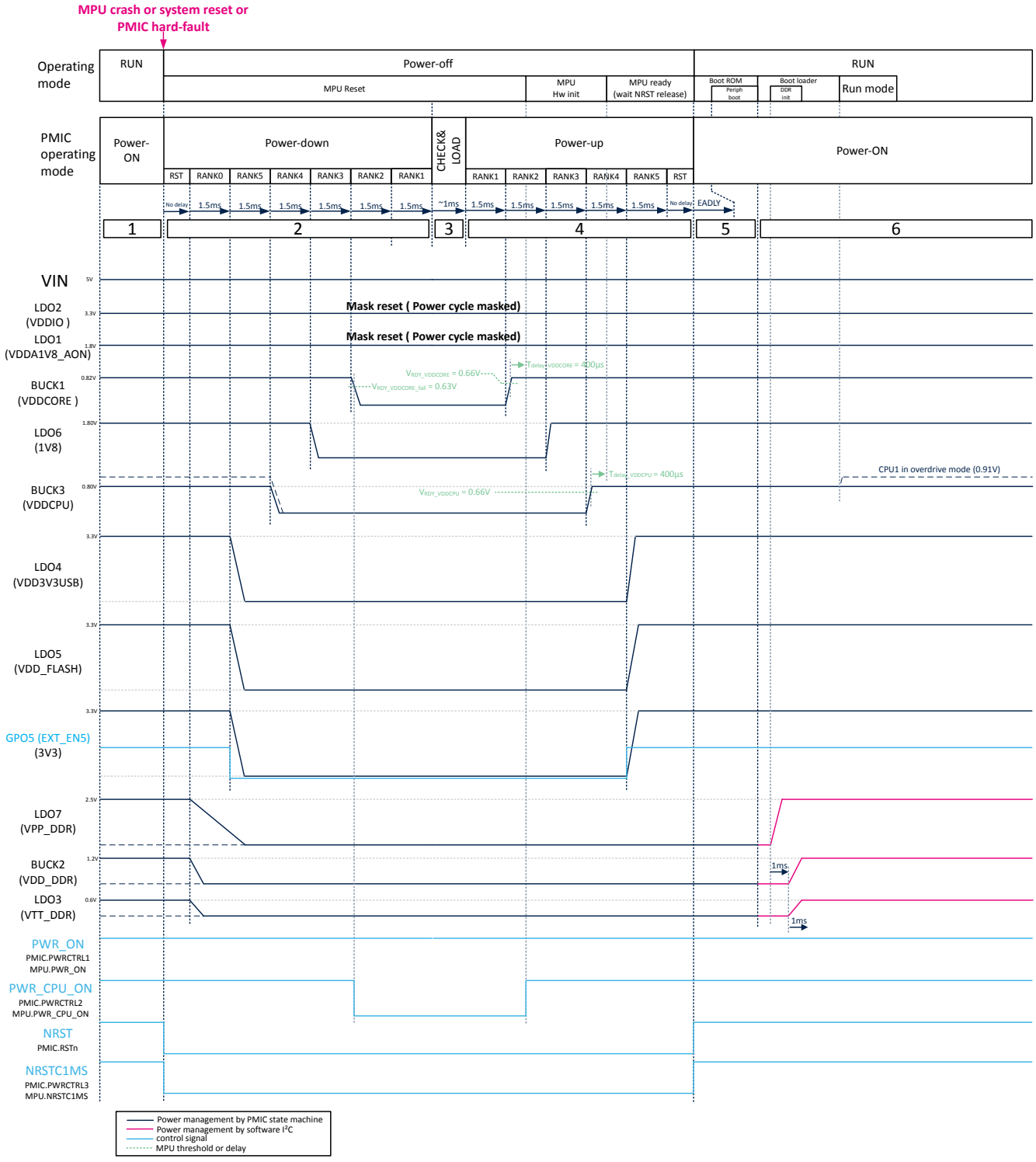
1. The application is powered up and is in Run mode.
2. A crash occurs when one of the MPU watchdogs elapses, or the user presses the reset button generating a pulse on the NRST signal. The PMIC detects the reset assertion (NRST pulse low) and starts a uninterruptible power cycle: The PMIC performs a power-down sequence:
   a. The PMIC asserts the RSTn, asserting the MPU NRST signal.
   b. RANK0 (1.5 ms): LDO7 (V_{PP_DDR}), BUCK2 (V_{DD_DDR}), and LDO3 (V_{TT_DDR}) are disabled.
   c. RANK5 (1.5 ms): LDO4 (V_{DD3V3_USB}), LDO5 (V_{DD_FLASH}) and GPO5 (EXT_EN5) are disabled. The 3V3 discrete SMPS regulator is disabled and the 3V3 voltage falls.
   d. RANK4 (1.5 ms): BUCK3 (V_{DDCPU}) is disabled
   e. RANK3 (1.5 ms): LDO6 (1V8) is disabled.
   f. RANK2 (1.5 ms): BUCK1 (V_{DDCORE}) is disabled. Once the VDDCORE voltage drops below MPU $V_{RDY\_VDDCORE}$ falling threshold, the MPU PWR_CPU_ON signals goes low.
   g. RANK1 (1.5 ms): LDO2 (V_{DDIO}) and the LDO1 (VDDA1V8_AON) are keep enabled as they are masked in reset sequence (See Section 5.1.3: PMIC mask-reset option for details)
3. Once the PMIC ends the power down sequence, it goes in *CHECK&LOAD* state to prepare the power-up sequence and to check if power-up conditions are fulfilled.

4. Once the *CHECK&LOAD* states ends, the PMIC starts a power-up sequence. The STPMIC2LA regulators follow the power-up sequence predefined in its NVM:
   a. RANK1 (1.5 ms): LDO2 (V_{DDIO}) is already enabled at 3.3 V and the LDO1 (V_{DDA1V8_AON}) is already enabled at 1.8 V.

Note: NRST signal is kept low as the PMIC asserts the NRST signal until the end of the power-up sequence

   b. RANK2 (1.5 ms): The BUCK1 (V_{DDCORE}) is enabled at 0.82 V. Once V_{DDCORE} voltage rises above MPU $V_{RDY\_VDDCORE}$ threshold (0.66 V), a MPU Tdelay_VDDCORE (400 µs typ.) delay is started to wait for V_{DDCORE} voltage to reach minimum operating voltage. Once Tdelay_VDDCORE elapses, the MPU starts hardware initialization such as: starting the HSI oscillator and so on, and the PWR_CPU_ON signal goes high.
   c. RANK3 (1.5 ms): LDO6 (1V8) is enabled at 1.8 V.
   d. RANK4 (1.5 ms): BUCK3 (V_{DDCPU}) is enabled at 0.80 V. Once V_{DDCPU} voltage rises above MPU V_{RDY_VDDCPU} threshold (0.66 V), an MPU T_{delay_VDDCPU} (400 µs typ.) delay is started to allow V_{DDCPU} voltage to reach minimum operating voltage. Once T_{delay_VDDCPU} elapses, the MPU is ready to boot, and it keeps in reset until the PMIC releases the NRST signal.
   e. RANK5 (1.5 ms): LDO4 (V_{DD3V3_USB}) and LDO5 (V_{DD_FLASH}) is enabled at 3.3 V. The GPO5 (EXT_EN5) is enabled. The 3V3 discrete SMPS regulator is enabled and the 3V3 voltage rises.
   f. Once RANK5 is ended, the PMIC releases the RSTn that releases MPU NRST.
5. Once the NRST is released, the MPU enters in Run mode:
   a. The MPU releases the NRSTC1MS signal.
   b. The D1 CPU starts to execute the boot ROM: EADLY timer starts.
   c. Once EADLY elapses, the boot ROM reads, verifies, and executes the bootloader from the external flash memory (eMMC).
6. The bootloader software performs initializations, then loads and executes the application software:
   a. The software enable LDO7 (V_{PP_DDR}) at 2.5 V.
   b. The software waits for 1 ms.
   c. The software enables the following once the delay has elapsed:
      - LDO3 (V_{TT_DDR}) in sink source mode
      - BUCK2 (V_{DD_DDR}) at 1.2 V
   d. The software waits for 1 ms.
   e. The MPU software initializes the DDR4 controller and DDR memory ICs.
   f. The bootloader loads the application software into DDR4 and executes it. The kernel initializes.
   g. The system is now running.

Figure 13. **System crash recovery sequence**



5.4.2 CPU1 crash recovery management

As introduced in the PWRCTRL1, PWRCTRL2, PWRCTRL3 section, the NRSTC1MS pin is dedicated to reset peripherals associated with the CPU1, specifically for the flash memory boot device.

If a crash occurs on CPU1, only D1 domain is reset; the CPU2 on D2 domain is not affected and continues to run:

1. The application is powered up and is in Run mode.
2. A CPU1 crash occurs, triggered by the watchdog elapsing on CPU1, then the MPU resets the CPU1.

3. Once the MPU releases the CPU1 reset, the CPU1 executes the boot ROM.
4. The boot ROM software generates a pulse on NRSTC1MS (see C1MSRD timer section):
   a. The C1MSRD timer started
   b. The NRSTC1MS signal is set low by MPU: the LDO5 (V_{DD_FLASH}), assigned to NRSTC1MS (PMIC PWRCTRL3), is powered OFF.
5. Once C1MSRD timer elapses (2 ms), the NRSTC1MS signal is set high by the MPU:
   – The LDO5 is turned ON with the default voltage set in the PMIC NVM
6. The boot ROM starts the EADLY timer (5 ms) to wait for LDO5 (V_{DD_FLASH}) voltage to stabilize and the eMMC flash memory to initialize.
7. Once the EADLY elapses, the boot ROM reads, verifies, and executes the bootloader from the external flash memory (eMMC).
   – The bootloader software performs the CPU1 initializations, then loads and executes the CPU1 application software.

6 Safety management

In this document, the safety management is the concept of implementing mechanisms such as over current protection (OCP), watchdogs, and so on, to maintain the system functional and robust. The objective is to protect the safety and integrity of the application against internal or external errors, or dysfunctions.

The safety management is provided by the MPU software and/or by the PMIC functionalities.

This section focuses on PMIC safety management functionalities. It is based on failure detection (hard fault) and related PMIC behavior such as fail-safe management.

Refer to [2] for more details.

6.1 PMIC fail-safe management

Each source of hard fault as defined in Section 6.2: PMIC hard-faults has a dedicated independent fail-safe counter. This counter, named xxxx_FLT_CNT (where xxxx is the hard fault source), is incremented each time a hard fault event occurs, in addition to a turn-off condition.

The maximum fault iteration counter, xxxx_FLT_CNT_MAX set in PMIC NVM, is used to define the maximum number of the hard fault iterations before the PMIC enters into FAIL_SAFE_LOCK state. By default, there is no limit applied any of the hard-fault counter on the STPMIC2LA. It means that STPMIC2LA always restarts after a hard fault event occurs and never enters in FAIL_SAFE_LOCK state.

As long as the fail-safe counter xxxx_FLT_CNT does not reach the xxxx_FLT_CNT_MAX value, the PMIC carries out a power cycle each time a hard fault event occurs. A power cycle is defined by:

1. a power-down sequence
2. waits for FLT_TMR (fault timer) to end
3. runs a power-up sequence as defined in Section 5.4.1: System crash recovery management

Once the number of hard fault iterations exceed the xxxx_FLT_CNT_MAX counter, the system is blocked in FAIL_SAFE_LOCK state. To exit this state, the PMIC must carry out a main supply removal or a PONKEYn long press (a special NVM setting is necessary).

The RST_FLT_CNT_TMR reset fault timer may be enabled by NVM to automatically clear all xxxx_FLT_CNT fail-safe counters if no hard fault has occurred until RST_FLT_CNT_TMR elapses.

The following examples illustrate fail-safe management mechanisms.

Example 1: STPMIC2LA behavior with a negative voltage glitch on V_{IN}

The initial condition is the STPMIC2LA NVM has the default value.

The VIN_FLT_CNT_MAX counter is configured in the NVM fail-safe shadow register NVM_FS_SHR1, to 1111 (infinite hard fault configuration).

Description:

A 5 V wall adapter is plugged to the application main supply connector. The V_{IN} rises above V_{INOK_rise} (4 V), the PMIC then powers up the application and the application initializes. A negative voltage glitch occurs on V_{IN} due to bad contact at main supply connector. This causes a glitch voltage to fall below VINOK_fall (3.5 V). The PMIC causes a V_{IN} hard fault condition that triggers a power-off sequence and the VIN_FLT_CNT dedicated fail-safe counter is incremented by one. Once the power-off sequence is completed, the PMIC evaluates the state transition and goes to the power-up sequence as long as VIN_FLT_CNT ≤ VIN_FLT_CNT_MAX.

Once the power-up sequence ends, PMIC goes in power-on state and the application initializes and runs.

If several other negative glitches occur on the main supply input (V_{IN}) below VINOK_fall, the STPMIC2LA always restarts as VIN_FLT_CNT ≤ VIN_FLT_CNT_MAX is always true.

This behavior is identical for other hard fault sources: the STPMIC2LA always restarts.

Example 2: STPMIC2LA behavior with negative voltage glitch on V_{IN} and tuned fail-safe management in STPMIC2LA NVM

The initial condition is the STPMIC2LA NVM has been tuned to adjust fail-safe management as follows:

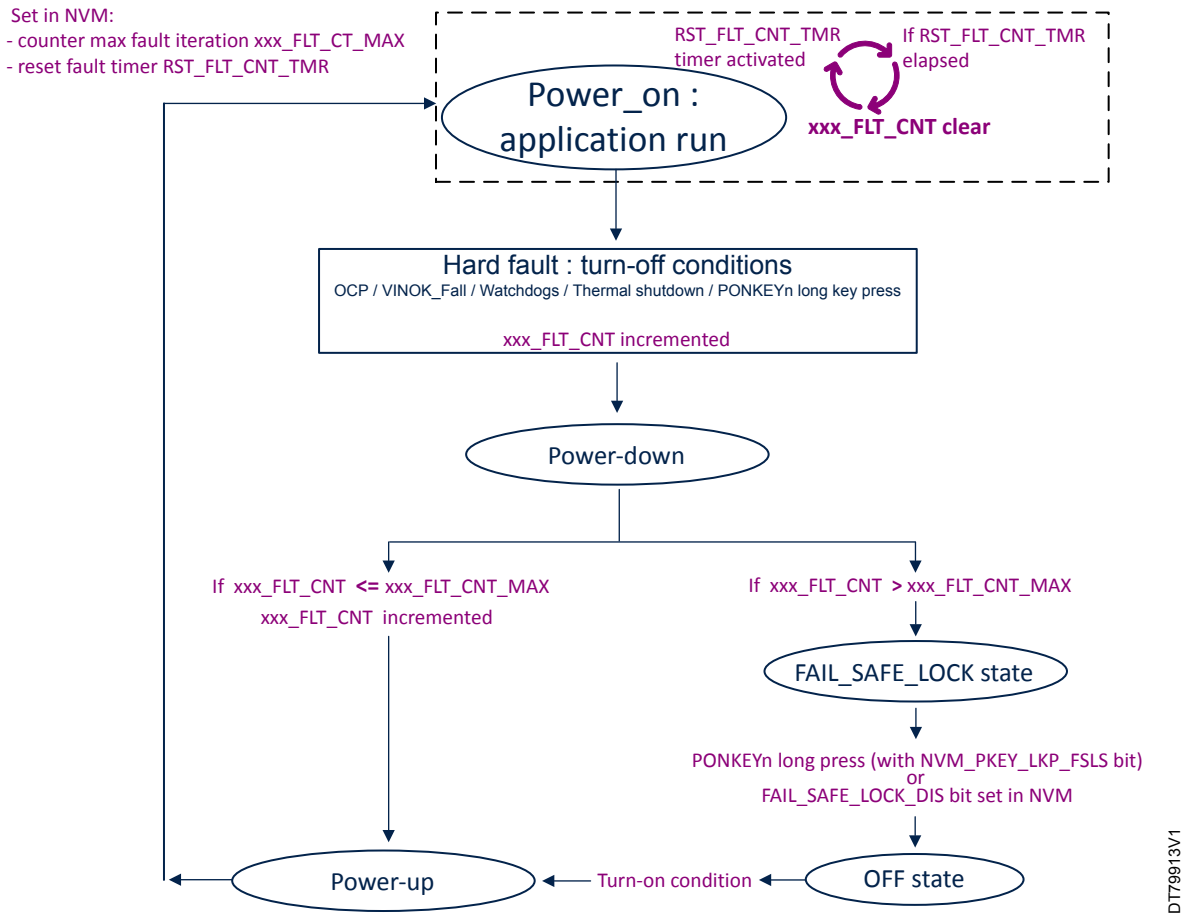
- VIN_FLT_CNT_MAX[3:0] = 0x0001 programmed in PMIC NVM_FS_SHR1 NVM. This specifies that one hard fault on V_{IN} is allowed.
- RST_FLT_CNT_TMR[1:0] = 0x10 programmed in PMIC NVM_FS_SHR2 NVM. This specifies that all fault counters are cleared if no hard fault is detected for six min.

Description

A 5 V wall adapter is plugged to the application main supply connector. The V_{IN} rises above V_{INOK_rise} (4 V), the PMIC then powers up the application and the application initializes. A negative voltage glitch occurs on V_{IN} due to bad contact at main supply connector. This causes a glitch voltage to fall below VINOK_fall (3.5 V). The PMIC causes a V_{IN} hard fault condition that triggers a power off sequence. The VIN_FLT_CNT dedicated fail-safe counter is incremented by one. Once the power-off sequence is completed, the PMIC evaluates the state transition and goes to the power-up sequence as VIN_FLT_CNT ≤ VIN_FLT_CNT_MAX condition is true. Once the power-up sequence ends, the PMIC goes in power-on state and the application initializes and runs.

- If another negative voltage glitch on V_{IN} occurs before the RST_FLT_CNT_TMR elapses, within the following six minutes, the PMIC causes a V_{IN} hard fault condition that triggers a power-off sequence. The dedicated VIN_FLT_CNT fail-safe counter is incremented by one (the content is two). Once the power-off sequence ends, the PMIC evaluates state transition which states that VIN_FLT_CNT ≤ VIN_FLT_CNT_MAX condition is wrong. The PMIC goes in FAIL_SAFE_LOCK_STATE and is locked in this state until next PMIC POR (main supply removal).
- If another negative voltage glitch on V_{IN} occurs after the RST_FLT_CNT_TMR elapses, within the following six minutes, all xxxx_FLT_CNT are cleared including the VIN_FLT_CNT. Then, if a new negative glitch on V_{IN} occurs, the PMIC enters power-off and then it powers up as VIN_FLT_CNT ≤ VIN_FLT_CNT_MAX condition is true.

Figure 14. PMIC fail-safe management mechanism



6.2 PMIC hard-faults

The PMIC considers five hardware source events as hard-fault conditions:

- OCP: Overcurrent protection, including short circuit
- VIN: Undervoltage protection where V_{IN} falls below V_{INOK_fall} threshold
- TSHDN: Thermal shutdown protection
- WDG: Watchdog timer expiration
- PKEY: Power on key button long press

Each hard-fault source is managed in the same way:

- A turn-off condition is triggered (see Turn-off conditions) followed by PMIC power-down sequence
- Then, depending on fail-safe management settings, the PMIC can:
  - Restart automatically (power-up sequence). This is the default behavior of the STPMIC2LA.
  - Not restart automatically: PMIC is kept in FAIL_SAFE_LOCK or OFF state until an authorized turn-on condition is issued.

Note: It is implied, a hard-fault can occur only when PMIC is in power-on state.

6.2.1 OCP overcurrent protection

All the PMIC regulators implement two level of protection against overcurrent or short circuit on their output.

HICCUP (level 0)

In case of overcurrent or short-circuit, each PMIC regulator (set in level 0) operates independently in hiccup mode without impacting any other power domain of the application. This level of protection is suitable for non-critical power domains.

Once an OCP occurs, the regulator turns off for the t_{HICCUP_DLY} timer to elapse. This delay is predefined in NVM, and then restarts.

Accordingly, the regulator restarts infinitely until the overcurrent/short-circuit disappears.

OCP hard-fault management (Level 1)

In case of overcurrent or short-circuit, a PMIC regulator set in level 1 triggers an OCP hard fault turn-off condition. This implies that the PMIC enters power-off state, and then power-on state (depending on fail-safe settings). This level of protection is suitable for the critical power domain, such as MPU V_{DDCORE}, DDR4 regulators, where it is mandatory to restart the application completely in case of overcurrent or short-circuit.

Section 6.3: OCP management in the application illustrates the STPMIC2LA OCP settings based on the application in Figure 1.

6.2.2 V_{IN} undervoltage protection (V_{IN} < V_{INOK_Fall})

The PMIC embeds an undervoltage protection to prevent the MPU or peripherals from crashing in case the PMIC regulators are unable to maintain the correct regulation due to V_{IN} input voltage being too low.

Once the main V_{IN} supply goes below V_{INOK_fall} threshold, even for a very short duration, such as a voltage glitch, generates a hard fault condition. This results in following steps:

1. the V_{IN} fail-safe counter (VIN_FLT_CNT) is incremented
2. the PMIC powers down.

If the VIN_FLT_CNT counter is higher than the VIN_FLT_CNT_MAX when power-down ends, the PMIC enters in FAIL_SAFE_LOCK state.

If the VIN_FLT_CNT counter does not exceed the VIN_FLT_CNT_MAX when power-down ends, the PMIC waits for FLT_TMR fault timer to elapse, where t_{VINOK_Fall} = 100 ms. The PMIC enters power-up and then goes to power-on. The application then initializes and runs.

6.2.3 TSHDN: thermal shutdown protection

The PMIC embeds a thermal protection to avoid any damage from the part overheating.

Two levels of thermal protection are available:

- First level, an interruption is sent to the MPU:
  Once the PMIC junction temperature goes higher than or below temperature thresholds (respectively T_{WRN_Rise} or T_{WRN_Fall}), the PMIC generates an interrupt to be caught and managed by the MPU.
- Second level, hard fault condition is generated:
  Once the PMIC junction temperature exceeds the T_{SHDN_Rise} temperature thresholds:
  1. a hard-fault condition is generated
  2. the thermal fail-safe counter (TSHDN_FLT_CNT) is incremented
  3. the PMIC enters power-down.

  If the TSHDN_FLT_CNT counter is higher than the TSHDN_FLT_CNT_MAX when power-down ends, the PMIC enters in FAIL_SAFE_LOCK state.
  If the TSHDN_FLT_CNT counter does not exceed the TSHDN_FLT_CNT_MAX when power-down ends, the PMIC waits for:
  - FLT_TMR fault timer to elapse, where t_{SHDN_DLY} =3 s
  - PMIC junction temperature goes below T_{SHDN_Fall} threshold

  The PMIC enters power-up, and then goes to power-on. The application then initializes and runs.

6.2.4 WDG: watchdog timer expiration

The PMIC embeds a programmable watchdog that can be enabled at runtime by the software, or enabled by default at power-up (NVM setting):

- WDG_TMR_SET: watchdog timer duration value setting.
- WDG_TMR_CNT: watchdog timer down-counter.
- WDG_EN: watchdog enable bit.
- WDG_RST: watchdog clear bit. This bit must be periodically set by the MPU software to reset the watchdog timer.

Note: The PMIC PWRCTRLx input can be used to suspend the watchdog, typically in low-power mode. When the PWRCTRL is asserted in low-power mode, the watchdog timer is suspended. When this PWRCTRL is desasserted, the watchdog timer is resumed.

If the MPU software fails to clear the PMIC watchdog timer (WDG_RST) bit, the watchdog timer elapses.

Once the PMIC watchdog timer elapses, a hard fault condition is generated:

- the watchdog fault counter (WDG_FLT_CNT) is incremented
- the PMIC enters power-down.

If the WDG_FLT_CNT counter is higher than the WDG_FLT_CNT_MAX when power-down ends, the PMIC enters in FAIL_SAFE_LOCK state.

If the WDG_FLT_CNT counter does not exceed the WDG_FLT_CNT_MAX when power-down ends, the PMIC enters power-up, then goes to power-on. The application then initializes and runs.

6.2.5 PKEY: power on key user button long press

A long press on the PONKEYn user button enables the PMIC hard fault condition to be triggered. It is similar to a system reset except that PMIC performs a power cycle in addition to asserting the reset signal.

The long press duration is set to 10 seconds by default with STPMIC2LA. The NVM can be reprogrammed with one of the following configurations:

- Adjust the duration by modifying NVM_PKEY_LKP_TMR bit
- Disable this feature by setting NVM_PKEY_LKP_OFF bit.

Once the long key press PONKEYn timer elapses, by default set to 10 s, a hard fault condition is generated: the PKEY fail-safe counter (PKEY_FLT_CNT) is incremented and the PMIC enters power-down.

If the PKEY_FLT_CNT counter is higher than the PKEY_FLT_CNT_MAX when power-down ends, the PMIC enters in FAIL_SAFE_LOCK state.

If the PKEY_FLT_CNT counter does not exceed the PKEY_FLT_CNT_MAX when power-down ends, the PMIC enters power-up, and then goes to power-on. The application then initializes and runs.

6.3 OCP management in the application

In the application illustrated in Figure 1, the two levels of protection are applied on the regulators as follows:

- All critical regulators needed for the application are managed in OCP fail-safe (level 1), else in HICCUP (level 0).
- These settings can be modified by reprogramming the NVM_FS_OCP_SHRx registers in PMIC NVM according to Table 16.

Table 16. OCP management application

Regulator	HICCUP (level 0)(1)	Fail Safe (level 1)
BUCK1 (V_{DDCORE})	-	YES
BUCK2 (V_{DD_DDR})	-	YES
BUCK3 (V_{DDCPU})	-	YES
LDO1 (V_{DDA1V8_AON})	-	YES
LDO2 (V_{DDIO})	-	YES
LDO3 (V_{TT_DDR})	-	YES
LDO4 (V_{DD3V3_USB})	YES	-
LDO5 (VDD_FLASH)	YES	-
LDO6 (1V8)	-	YES
LDO7 (V_{PP_DDR})	-	YES

1. *Refer to HICCUP for more details*

Note: The table above applies to STPMIC2LA

Revision history

Table 17. **Document revision history**

Date	Version	Changes
04-May-2026	1	Initial release.

Contents

List of tables

List of figures

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice.

In the event of any conflict between the provisions of this document and the provisions of any contractual arrangement in force between the purchasers and ST, the provisions of such contractual arrangement shall prevail.

The purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

The purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of the purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

If the purchasers identify an ST product that meets their functional and performance requirements but that is not designated for the purchasers' market segment, the purchasers shall contact ST for more information.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.