AN6312

Application note

How to use STPMIC1L to develop a wall adapter powered application on STM32MP13 MPUs

Introduction

This application note applies to the STM32MP13x MPU devices as detailed in the table below. The devices are referred to as STM32MP13x in the rest of the document. It is powered by the STPMIC1Lx power management IC, which is fully featured to supply a core chipset (STM32MP13x, DDR memory, and a flash memory).

This document provides an example of three hardware reference designs based on an STM32MP13x device powered by a 5 V external power supply source through the STPMIC1LAPQR power management IC for applications not supporting CPU overdrive mode and the STPMIC1LDPQR power management IC supporting CPU overdrive mode. The three applications are suitable for peripheral I/O voltages at 3.3 V.

This document is intended for product architects and designers who require information about power management and STPMIC1Lx settings, and it 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	CPU frequency
STM32MP13x	STM32MP131A, STM32MP131C, STM32MP133A, STM32MP133C, STM32MP135A, STM32MP135C	Up to 650 MHz
STPMIC1Lx	STPMIC1LAPQR	
STM32MP13x	STM32MP131D, STM32MP131F, STM32MP133D, STM32MP133F, STM32MP135D, STM32MP135F	Up to 1000 MHz
STPMIC1Lx	STPMIC1LDPQR	

1 General information

This document applies to STM32MP13x Arm® Cortex®-based MPUs and STPMIC1Lx power management IC.

Note: Arm and Cortex are registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere.

The Arm word and logo are trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved.

2 Overview

This application note describes the interaction between the STM32MP13x and the STPMIC1Lx 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).
- DDR3L memory.
- Peripheral I/O interface voltage (V_{DDIO}) at 3.3 V supplied by the STPMIC1LAPQR or the STPMIC1LDPQR.
- SD card as boot device.

Not covered in this application note:

- Other DDR type (lpDDR3)
- Peripheral interface with I/O voltage (V_{DD}) of 1.8 V

In this document, MPU terminology refers to the STM32MP13x, and PMIC terminology refers to the STPMIC1Lx generic device. The STPMIC1LA highlights specific behaviors predefined in the STPMIC1LAPQR NVM, while the STPMIC1LD highlights specific behaviors predefined in the STPMIC1LDPQR NVM.

2.1 Reference documents

Table 2. Reference documents

-	Reference	Title
STMicroelectronics document (1)		
[1]	DS14839	STPMIC1L highly integrated power management IC for microprocessor units
[2]	RM0475	STM32MP13xx advanced Arm®-based 32-bit MPUs
[3]	AN5474	Getting started with STM32MP13x lines hardware development
[4]	DS13874	STM32MP135A STM32MP135D
[5]	Wiki page DDR config	https://wiki.st.com/stm32mpu/wiki/DDRCTRL_and_DDRPHYC_device_tree_configuration
[6]	AN5438	STM32MP1 series lifetime estimates

1. *Refer to www.st.com.*

3 Glossary

Table 3. Glossary

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

4 5 V power supply application reference designs

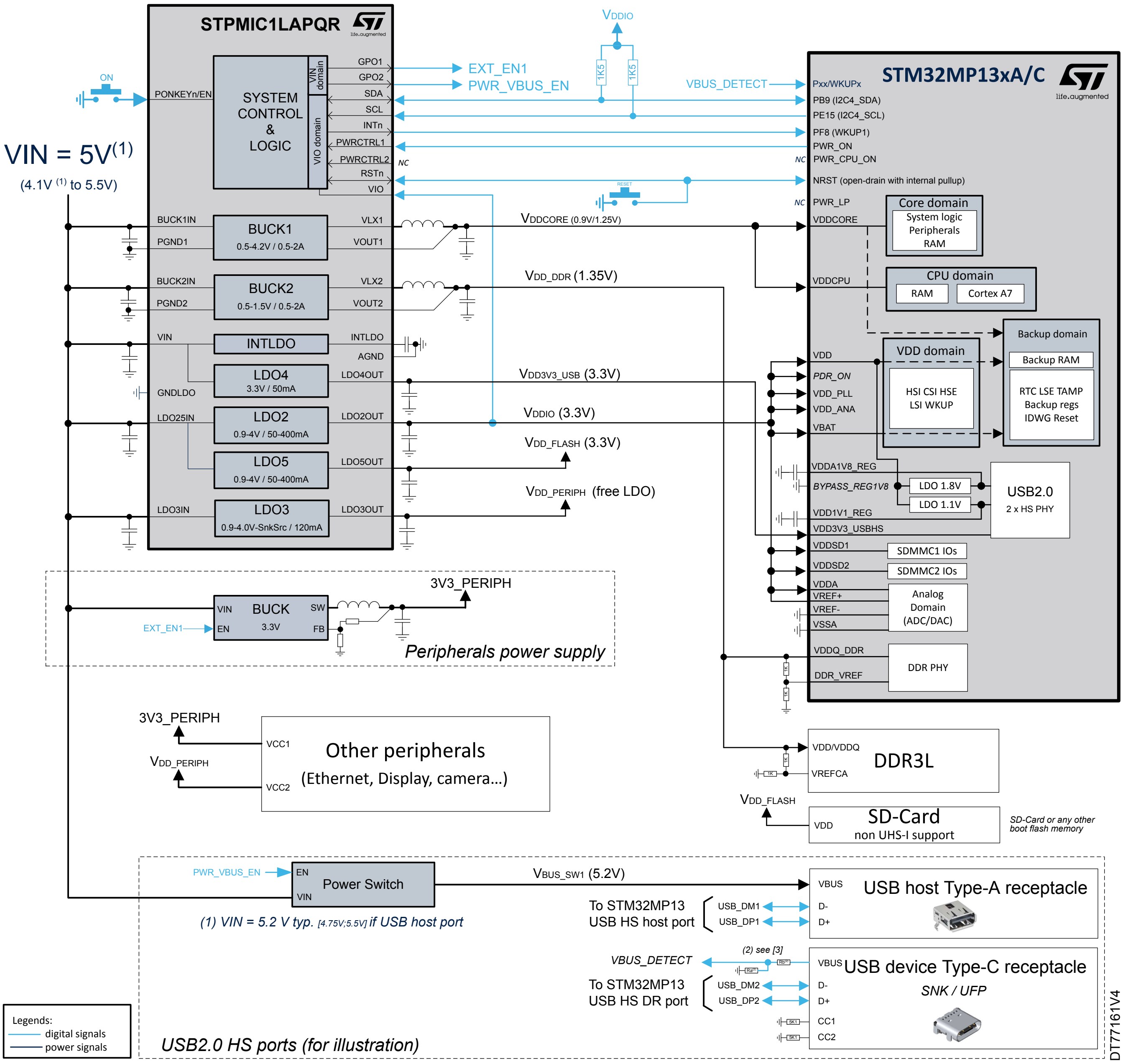
4.1 STPMIC1LA and STM32MP13xA/C with CPU frequency of up to 650 MHz

The reference design shown in Figure 1 targets an application powered by a main supply of 5 V (5.2 V if USB host port). The STM32MP13xA/C core and CPU power domains are merged and powered by the STPMIC1LA BUCK1 SMPS. The DDR3L DRAM, and the SD card boot flash memory are powered by the STPMIC1LA BUCK2 SMPS and LDO5, respectively. For illustration, application peripherals such as USB ports, Ethernet PHY, and others are powered by a discrete regulator or a power switch.

Note: The SD card boot flash memory is used as an illustration. It could be replaced by any supported flash memory such as eMMC, NAND flash memory, or serial flash memory (see [3]).

The main peripheral interfaces operate with an I/O voltage of 3.3 V.

Figure 1. STM32MP13xA/C and STPMIC1LA with DDR3L, SD card, and external SMPS to supply peripherals



Note: The following elements are not shown in the diagram:
- *STM32MP13xA/C decoupling scheme (see [3]).*
- *STPMIC1LA discrete components value (see [1]).*
- *VIN source and related protection, such as ESD, EMI filtering, and overvoltage protection.*

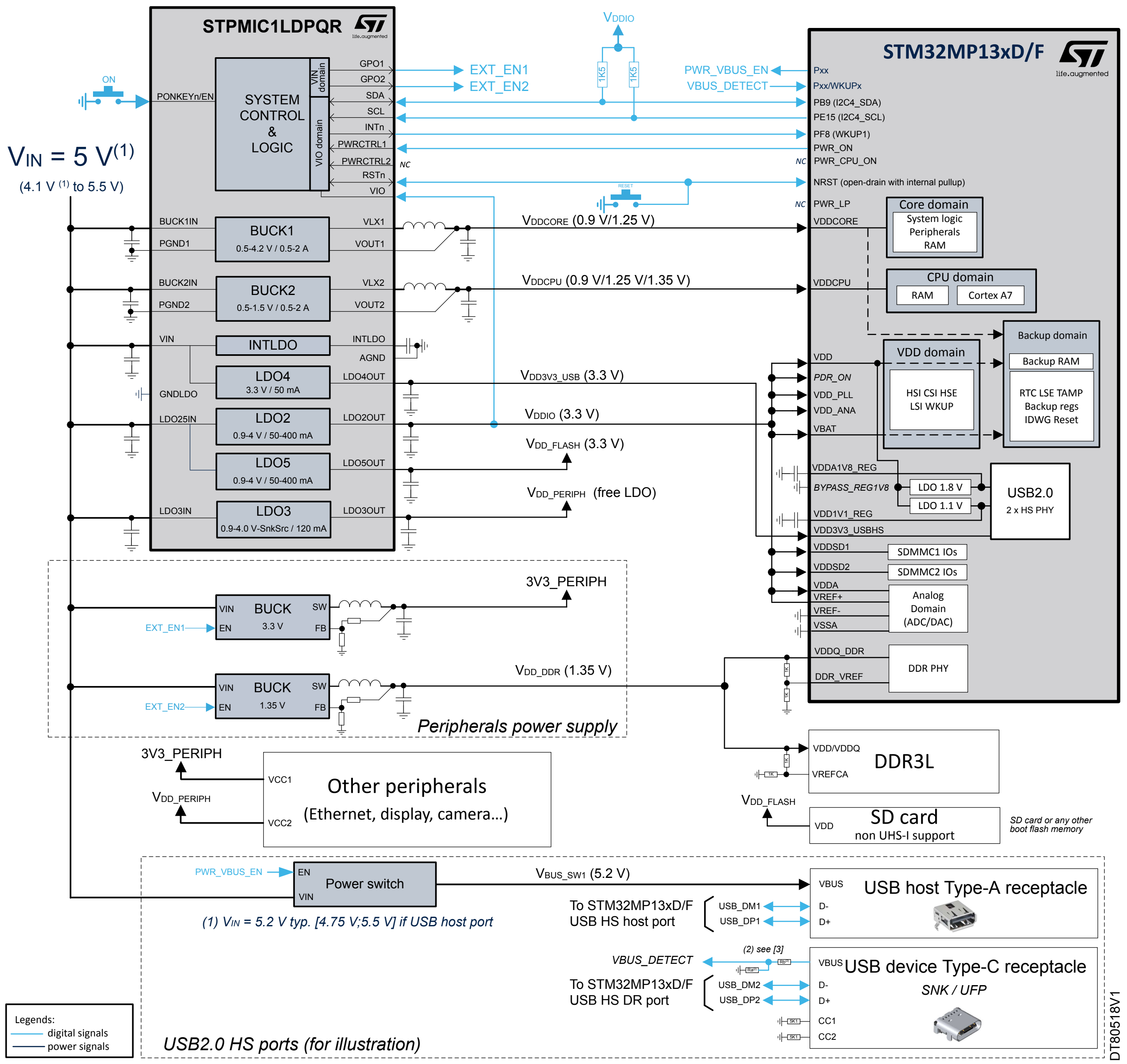
4.2 STPMIC1LD and STM32MP13xD/F with CPU frequency up to 1000 MHz

The reference design shown in Figure 2 targets an application powered by a main supply of 5 V (5.2 V if using a USB host port). The STM32MP13xD/F core and CPU power domains are powered by the STPMIC1LD BUCK1 SMPS and BUCK2 SMPS, respectively. The SD card boot flash memory is powered by the STPMIC1LD LDO5, and the DDR3L DRAM is powered by a discrete SMPS. For illustration purposes, application peripherals such as USB ports and Ethernet PHY are powered by a discrete regulator or a power switch.

Note: The SD card boot flash memory is used for illustration purposes. It could be replaced by any supported flash memory, such as eMMC, NAND flash memory, or serial flash memory (see [3]).

The main peripheral interfaces operate with an I/O voltage of 3.3 V.

Figure 2. STM32MP13xD/F and STPMIC1LD with DDR3L, SD-card, external SMPS to supply DDR3L memory IC and peripherals



Note: The following elements are not shown in the diagram:
- *STM32MP13xD/F decoupling scheme (see [3]).*
- *STPMIC1LD discrete components value (see [1]).*
- *VIN source and related protection, such as ESD, EMI filtering, and overvoltage protection.*

4.3 STPMIC1LD and STM32MP13xD/F with CPU frequency up to 1000 MHz, and V_{DDCPU} merged with V_{DD_DDR}

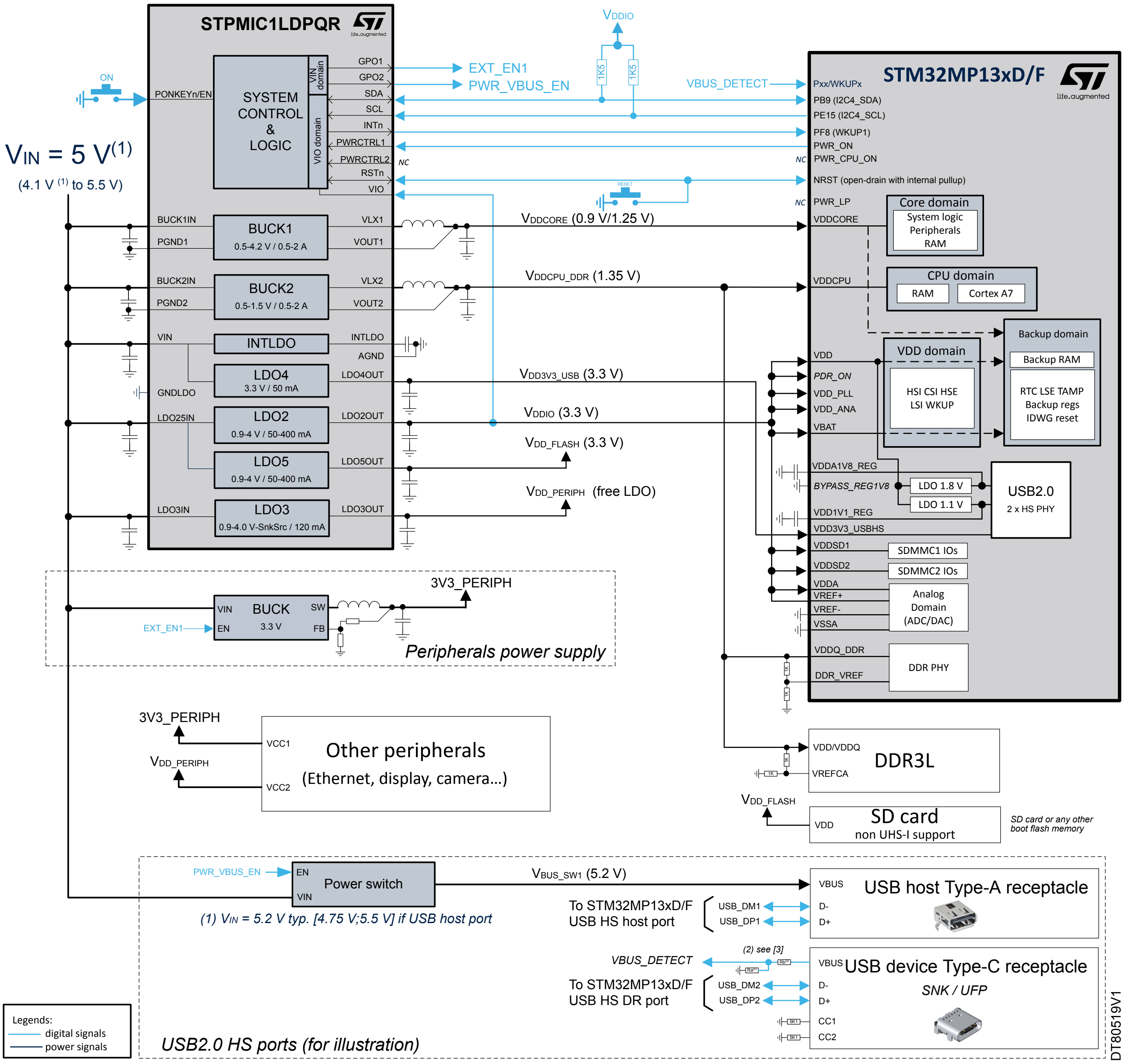
The reference design shown in Figure 3 targets an application powered by a main supply of 5 V (5.2 V if using a USB host port). The STM32MP13 core is powered by the STPMIC1LD BUCK1 SMPS. To save one regulator in the application, the STM32MP13 CPU and the DDR3L DRAM are both powered by the STPMIC1LD BUCK2. The SD card boot flash memory is powered by the STPMIC1LD LDO5. For illustration purposes, application peripherals such as USB ports, SD card, and Ethernet PHY are powered by a discrete regulator or a power switch.

Caution: *The reference design shown in Figure 3 requires setting VDDCPU_DDR to 1.35 V. Setting VDDCPU_DDR to 1.35 V continuously reduces the STM32MP13 lifetime. See [6] for more details.*

Note: The SD card boot flash memory is used for illustration purposes. It can be replaced by any supported flash memory such as eMMC, NAND flash memory, or serial flash memory (see [3]).

The main peripheral interfaces operate with an I/O voltage of 3.3 V.

Figure 3. STM32MP13xD/F and STPMIC1LD with DDR3L, SD-card, external SMPS to supply peripherals



Note: The following elements are not shown in the diagram:

- *STM32MP13xD/F decoupling scheme (see [3]).*
- *STPMIC1LD discrete components value (see [1]).*
- *VIN source and related protection, such as ESD, EMI filtering, and overvoltage protection.*

4.4 Power distribution

4.4.1 V_{DDCPU} power domain (900 mV / 1.25 V / 1.35 V)

V_{DDCPU} supplies the MPU Arm® Cortex®-A7 CPU digital power domain.

With the STPMIC1LA, as illustrated in Section 4.1: STPMIC1LA and STM32MP13xA/C with CPU frequency of up to 650 MHz, the V_{DDCPU} is merged with V_{DDCORE} at the PCB level and powered from the BUCK1 step-down SMPS. Accordingly, the CPU frequency is limited to the nominal value (up to 650 MHz) as overdrive voltage is not allowed on the V_{DDCORE} domain.

V_{DDCPU} is enabled in:

- Run nominal mode at 1.25 V

V_{DDCPU} is lowered to 900 mV in:

- Low-power LPLV-Stop mode

V_{DDCPU} is disabled in:

- Low-power Standby mode
- V_{BAT} mode and off mode

With the STPMIC1LD, as illustrated in Section 4.2: STPMIC1LD and STM32MP13xD/F with CPU frequency up to 1000 MHz, the V_{DDCPU} is independent from V_{DDCORE} and powered from the BUCK2 step-down SMPS. This configuration is specifically suitable for setting V_{DDCPU} at 1.35 V to operate the CPU clock up to 1000 MHz, in addition to supporting LPLV-Stop low-power mode (V_{DDCPU} at 900 mV) and nominal operating mode (V_{DDCPU} at 1.25 V for operating the CPU clock up to 650 MHz). Accordingly, an additional discrete regulator is required to power the DDR3L DRAM power domain compared to designs with the STPMIC1LA.

V_{DDCPU} is enabled in:

- Run nominal mode at 1.25 V
- Run overdrive mode at 1.35 V

V_{DDCPU} is lowered to 900 mV in:

- Low-power LPLV-Stop mode

V_{DDCPU} is disabled in:

- Low-power Standby mode
- V_{BAT} mode and off mode

With the STPMIC1LD, as illustrated in Section 4.3: STPMIC1LD and STM32MP13xD/F with CPU frequency up to 1000 MHz, and V_{DDCPU} merged with V_{DD_DDR}, the V_{DDCPU} is merged with the DDR3L DRAM supply. This configuration is suitable for saving the discrete DDR3L regulator while operating the CPU clock up to 1000 MHz. Accordingly, the V_{DDCPU} is powered at 1.35 V in all modes as the DDR3L DRAM must always be powered at 1.35 V.

V_{DDCPU} is enabled in:

- Run overdrive mode at 1.35 V

V_{DDCPU} is disabled in:

- Low-power Standby DDR off mode (suspend to flash)
- V_{BAT} mode and off mode

Note: *Low-power LPLV-Stop mode and low-power Standby DDR self-refresh mode are not supported as V_{DDCPU} voltage must be kept at 1.35 V in these modes.*

Caution: *Operating the V_{DDCPU} at 1.35 V reduces the STM32MP13 lifetime. See [6] for more details.*

In low-power mode, the PWR_ON output of the MPU manages the PMIC V_{DDCPU} regulator. The PWR_ON output is connected to the PWRCTRL1 input of the PMIC.

4.4.2 V_{DDCORE} power domain (900 mV / 1.25 V)

V_{DDCORE} is the main MPU digital power domain.

V_{DDCORE} supplies all core domains including the system logic, internal RAM, peripherals, and the backup domain.

With STPMIC1LA, as illustrated in Section 4.1: STPMIC1LA and STM32MP13xA/C with CPU frequency of up to 650 MHz, the V_{DDCORE} is merged with V_{DDCPU} at the PCB level and powered by the PMIC BUCK1 step-down SMPS. At power-up, the PMIC BUCK1 is automatically enabled at 1.22 V and must be set by software to the nominal voltage (1.25 V) during MPU initialization (see Section 4.5.2).

Note: *The default output voltage of the STPMIC1LA BUCK1 is set to 1.22 V (instead of 1.25 V) to ensure compatibility with STM32MP15 lines.*

With STPMIC1LD, as illustrated in Section 4.2: STPMIC1LD and STM32MP13xD/F with CPU frequency up to 1000 MHz, and Section 4.3: STPMIC1LD and STM32MP13xD/F with CPU frequency up to 1000 MHz, and V_{DDCPU} merged with V_{DD_DDR}, the V_{DDCORE} is powered by the PMIC BUCK1 step-down SMPS. At power-up, the PMIC BUCK1 is automatically enabled at 1.25 V (see Section 4.5.3).

V_{DDCORE} is enabled in:

- Run mode

V_{DDCORE} is lowered to 900 mV in:

- Low-power LPLV-Stop mode

V_{DDCORE} is disabled in:

- Low-power Standby mode
- V_{BAT} mode and 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 backup domain (see document [4]) in Run, Stop, and LPLV-Stop modes.

4.4.3 V_{DDIO} power domain (3.3 V)

V_{DDIO} is the power supply for the following independent MPU domains:

- V_{DD}
- V_{DDSD1}
- V_{DDSD2}
- V_{DD_PLL}
- V_{DD_ANA}
- V_{DDA}/V_{REF+}
- V_{BAT}

The MPU V_{DD}, V_{DD_ANA}, and V_{DD_PLL} domains must be connected together. They supply the MPU I/Os, the system analog such as oscillators (HSE, HSI), and PLLs.

The MPU V_{DDA} supplies the ADC and the voltage reference buffer (VREFBUF) to generate the V_{REF+} reference voltage of the ADC. The ADC performance is directly impacted by the noise level from the V_{REF+} source, but also by the V_{DDA} source noise level (due to the V_{DDA} power supply rejection ratio).

Note: If V_{DDA} is powered from the V_{DDIO} power domain, a low pass filter with low DC impedance might be inserted in between the V_{DDIO} power source and V_{DDA} depending on the required ADC performance.

Note: V_{REF+} must be connected to the V_{DDIO} power source only if limited ADC performance is expected.

The MPU V_{BAT} supplies the retention domain which includes the backup RAM, RTC, LSE, tamper, backup registers, watchdog, and reset blocks. In this application, the MPU V_{BAT} is powered from V_{DDIO} domain as no backup battery is present. VBAT may be supplied from a backup battery if the application requires keeping the backup domain powered when the main power supply source of the application (VIN) is removed.

VDDIO also supplies the PMIC VIO domain which embeds the I^2C interface, the PWRCTRLx, the RSTn, and the INTn pins. (See Section 4.5.)

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

At power-up, the V_{DDIO} is enabled automatically by the PMIC at 3.3 V. The LDO2 is the first regulator switched on at power-up. (See Section 5.2.1.)

V_{DDIO} is on in all modes except in off mode or V_{BAT} mode, when the main power source (VIN) of the application is removed.

4.4.4 V_{DD3V3_USB} power domain

V_{DD3V3_USB} power domain supplies the USB2.0 HS PHY (V_{DD33_USBHS}) of the MPU.

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

At power-up, the V_{DD3V3_USB} regulator is automatically enabled at 3.3 V by the PMIC. (See Section 5.2.1.)

V_{DD3V3_USB} can be kept enabled in Run, and LPLV-Stop modes if a USB peripheral is connected to the application.

V_{DD3V3_USB} must be disabled in Standby and off mode: when V_{DDCORE} is off.

At runtime, V_{DD3V3_USB} is controlled by the MPU software through an I²C command to the PMIC. In low-power mode, PWR_ON output of the MPU can manage the PMIC V_{DD3V3_USB} regulator via the PWRCTRL1 input of the PMIC.

4.4.5 DDR power domain (V_{DD_DDR})

V_{DD_DDR} is dedicated to DDR3L volatile memory IC power supply (V_{DD} and V_{DDQ}) and the MPU DDR interface (V_{DDQ_DDR}).

With STPMIC1LA, as illustrated in Section 4.1: STPMIC1LA and STM32MP13xA/C with CPU frequency of up to 650 MHz, V_{DD_DDR} (1.35 V) is powered from the PMIC BUCK2 step-down SMPS. At power-up, the BUCK2 regulator is not started automatically by the STPMIC1LA. It is powered up by software by sending I²C commands to the PMIC (see Section 5.2.1 and [5] for more details).

Note: The PMIC BUCK2 does not start automatically to ensure compatibility with multiple DDR types, such as DDR3L and LPDDR3, that use the same PMIC reference. The software sets the voltage of BUCK2 according to the DDR type, for example, 1.35 V for DDR3L and 1.2 V for LPDDR3, before enabling BUCK2.

Note: V_{TT_DDR} is an optional supply to the DDR3L IC bus termination resistor network that is not required for point-to-point topology (single DDR3L IC as illustrated in Figure 1) and recommended for fly-by topology (several DDR3L ICs). When required, V_{TT_DDR} (0.675 V) is powered from the PMIC LDO3 multipurpose LDO and must be set in sink-source mode. It provides voltage equal to BUCK2 output voltage / 2 (V_{OUT2} / 2). In that case, the PMIC LDO3 is dedicated to power supply DDR3L bus termination resistors network.

Note: When the LDO3 is used in sink-source mode (V_{TT_DDR}), it must be supplied from BUCK2 output (LDO3IN = V_{DD_DDR}).

With the STPMIC1LD, as illustrated in Section 4.2: STPMIC1LD and STM32MP13xD/F with CPU frequency up to 1000 MHz, V_{DD_DDR} (1.35 V) is powered by a discrete regulator. This regulator is controlled by PMIC GPO1 (enabled by default) or PMIC GPO2 (disabled by default). The selected GPO is controlled by the MPU software through an I²C command sent to the PMIC (see GPO1 and GPO2 for more details).

With the STPMIC1LD, as illustrated in Section 4.3: STPMIC1LD and STM32MP13xD/F with CPU frequency up to 1000 MHz, and V_{DDCPU} merged with V_{DD_DDR}, V_{DD_DDR} (1.35 V) is merged with V_{DDCPU} and powered by the PMIC BUCK2 step-down SMPS. This configuration eliminates the need for a discrete regulator dedicated to supplying DDR DRAM IC (see Section 4.4.1: V_{DDCPU} power domain (900 mV / 1.25 V / 1.35 V) for more details). At power-up, BUCK2 is automatically started by the STPMIC1LD at 1.25 V. The software must configure BUCK2 output voltage to 1.35 V before initializing the MPU DDR controller and the DDR3L DRAM IC (see Section 5.2.1 and [5] for more details).

Note: When V_{DDCPU} and V_{DD_DDR} are merged, as illustrated in Section 4.3: STPMIC1LD and STM32MP13xD/F with CPU frequency up to 1000 MHz, and V_{DDCPU} merged with V_{DD_DDR}, it is recommended to reprogram the STPMIC1LD NVM to set the BUCK2 default output voltage to 1.35 V instead of 1.25 V. This ensures compliance with the JEDEC (JESD79-3-1A) power-up constraint.

Note: Resistor voltage dividers of 1K/1K are required to supply respectively the DDR3L IC V_{REFCA} and MPU DDR_V_{REF} reference voltages using V_{DD_DDR} as a voltage source for the resistor voltage dividers.

V_{DD_DDR} is enabled in:

- Run mode
- Stop mode
- Low power LPLV-Stop mode
- Standby mode suspend to RAM (see Section 5.3.2: Standby mode (DDR3L in self-refresh))

V_{DD_DDR} is disabled in:

- Standby mode suspend to flash (see Section 5.3.3: Standby mode (DDR3L off))

4.4.6 V_{DD_FLASH} power domain (3.3 V)

V_{DD_FLASH} power domain aims to supply the boot flash memory of the MPU (handled by the MPU boot ROM). As illustrated in Figure 1, the boot flash memory is an SD card. However, any supported flash memory such as eMMC, NAND flash memory, or serial flash memory can replace the SD card (see [3]).

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

At power-up, the V_{DD_FLASH} is automatically enabled by the PMIC at 3.3 V. (See Section 5.2.1.)

The V_{DD_FLASH} regulator is controlled by software at runtime by sending an I2C command to PMIC. Additionally, it is controlled by MPU PWR_ON signal (via PMIC PWRCTRL1) to switch off V_{DD_FLASH} when entering a low-power mode and to switch on V_{DD_FLASH} on low-power mode exit. This is specifically suitable for Standby mode as MPU boot ROM requires access to the boot flash memory on standby mode exit.

4.4.7 3V3 peripherals power supply (3.3 V)

As illustrated in Figure 1, Figure 2, and Figure 3, a 3V3 power domain is used to power supply application peripherals around the MPU (Ethernet PHY, display, camera…).

3V3 is powered from a discrete regulator, such as a general purpose 3.3 V step-down SMPS. A PMIC GPOx can be used to control it:

- STPMIC1LA and STPMIC1LD GPO1: automatically set by PMIC during power-up sequence
- STPMIC1LA and STPMIC1LD GPO2: keep to 0 by PMIC during the power-up sequence

See GPO1 and GPO2 for more details.

4.5 Control signals between PMIC and STM32MP13x

4.5.1 PMIC default behavior overview

The PMIC regulators startup is distributed across five ranks to comply with the MPU power-up sequence constraints and to prevent current peaks on the main power supply. The voltage value of each regulator is defined in NVM to align with the MPU optimum voltage requirements.

The PMIC NVM settings are configured to automatically power up when the main power supply (VIN) is connected (see Section 5.2: Application power-up/power-down sequence for more details). The settings enable the MPU to boot from flash memory, such as an SD card, or from the USB interface. In production, booting from the USB interface is suitable for flashing and then executing the production software, the final application software, or both.

4.5.2 STPMIC1LA default behavior with STM32MP13xA/C

The default NVM configuration is available in [1] and summarized in the following table:

Table 4. Default STPMIC1LA NVM configuration

Regulator	Name	Rank	Default output voltage	Default configuration
BUCK1	V_{DDCORE}	RANK2	1.22 V	On
BUCK2	V_{DD_DDR}	RANK0	N/A	Off
LDO2	V_{DDIO}	RANK1	3.3 V	On
LDO3	*Free*	RANK0	N/A	Off
LDO4	V_{DD3V3_USB}	RANK5	3.3 V	On
LDO5	V_{DD_FLASH}	RANK4	3.3 V	On
GPO1	EXT_EN1	RANK3	N/A	On
GPO2	EXT_EN2	RANK0	N/A	Off

4.5.3 STPMIC1LD default behavior with STM32MP13xD/F

The default NVM configuration is available in [1] and summarized in the following table:

Table 5. **Default STPMIC1LD NVM configuration**

Regulator	Name	Rank	Default output voltage	Default configuration
BUCK1	VDDCORE	RANK2	1.25 V	On
BUCK2	VDDCPU	RANK3	1.25 V	On
LDO2	VDDIO	RANK1	3.3 V	On
LDO3	*Free*	RANK0	N/A	Off
LDO4	VDD3V3_USB	RANK5	3.3 V	On
LDO5	VDD_FLASH	RANK4	3.3 V	On
GPO1	EXT_EN1	RANK5	N/A	On
GPO2	EXT_EN2	RANK0	N/A	Off

4.5.4 PMIC digital control interface

The PMIC integrates:

- An I²C slave interface,
- Three digital input control pins (PONKEYn/EN, PWRCTRL1, PWRCTRL2),
- A digital output interrupt pin (INTn),
- A bidirectional digital reset pin (RSTn),
- Two general-purpose outputs (GPO1/2).

See [1] for details.

I²C interface

The PMIC is controlled by the MPU via the I²C interface to:

- Enable/disable, set the voltage, and operating mode of the regulators.
- Set regulators external control for low-power mechanisms (PWRCTRL1/2).
- Set the interrupt controller or read interrupt status.
- Set the protection (watchdog, overcurrent, undervoltage) or read protection status.
- Tune the PMIC NVM default configuration for end-product (power-up sequence, safety management).

PONKEYn/EN pin

The PONKEYn/EN pin can be muxed either with the PONKEYn or EN digital input feature.

For STPMIC1LA and STPMIC1LD, this pin is muxed as PONKEYn by default. The PMIC NVM can be reprogrammed to change this setting to the EN pin feature using the PKEY_EN_CFG bit in NVM_MAIN_CTRL_SHR3 register:

- When the PKEY_EN_CFG is set to 0, the PONKEYn/EN digital input acts as PONKEYn (default).
- When the PKEY_EN_CFG is set to 1, the PONKEYn/EN digital input acts as EN.

PONKEYn pin description

The STPMIC1L PONKEYn pin is a digital active low input signal with a built-in pull-up resistor. It is usually connected to a user push-button allowing the following operations:

- Turn on the PMIC (from PMIC off state).
- Wake up the application from a low-power mode (typically from Standby mode) by generating an interrupt on signal falling or rising edge.
- Force a switch-off or a power cycling condition with a long press. This duration is programmable as described in Turn-off conditions.

Note: The usage of a user push-button connection to PONKEYn is optional as the STPMIC1LA and the STPMIC1LD are automatically turned on when the application is powered.

EN pin description

The STPMIC1L EN (enable) pin is a digital input signal with a programmable polarity and a programmable pull-up or pull-down resistor. The EN feature powers on the PMIC when the pin is active (for example: EN = 1) or powers off the PMIC when the pin is inactive (for example: EN = 0).

This feature is mainly targeted for USB bus-powered application, where the EN pin of the PMIC is controlled from a USB UCSI power delivery controller to power on/off the application. Alternatively, this new feature enables usage of several PMIC working together into the same application.

RSTn pin

The PMIC RSTn pin must be connected to the MPU NRST pin. Additionally, it can be connected to a “RESET” user push-button. This pin has a built-in pull-up resistor. Therefore, no additional discrete pull-up resistor is needed on this signal. Nevertheless, a 10 nF capacitor to GND must 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 the MPU or the PMIC.

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

- When PMIC asserts RSTn (such as during the power-up or the power-down sequence), it drives the 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 its 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, PMIC waits for the 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 PF8 (wake-up) input pin. This pin has a built-in pull-up resistor. Therefore, no additional discrete pull-up resistor is needed on this signal.

PF8 has both interrupt and wake-up capabilites:

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

PWRCTRL1, PWRCTRL2

The PMIC has two power control digital input signals that can be connected to dedicated MPU control signals.

Each PMIC regulator can be set to be controlled from a single PWRCTRL signal. Thus, typically to switch on/off or to change the output voltage of a regulator depending on the PWRCTRL signal state. Alternatively, a PWRCTRL signal can be set to reset the regulator at the value defined in PMIC NVM (see document [1] for more details).

As illustrated in Figure 1 application, the MPU PWR_ON output pin controls the PMIC PWRCTRL1 input pin. The MPU PWR_ON pin has an internal mux with either PWR_ON or PWR_LP signal:

- PWR_ON pin is mux to PWR_LP signal to manage LPLV-Stop mode.
- PWR_ON pin is mux to PWR_ON signal (default) to manage Standby mode.

(See Section 5.1.3 for more details about PMIC PWRCTRL settings.)

GPO1 and GPO2

The PMIC has two general-purpose push-pull (internally referenced to VIN voltage) outputs. Those two outputs are targeted to control discrete external regulators.

For example, to control regulators supplying application peripherals or discrete power switch as illustrated in Figure 1 application. It can also be used as general-purpose GPOs to control some peripherals or to control another PMIC in slave mode (for example, an STPMIC1L GPOx connected to another STPMIC1L EN input pin).

GPO1 and GPO2 are driven by the PMIC with the same registers set as any other PMIC regulator. Accordingly, they are seen as PMIC regulators by the MPU PMIC software driver (see Table 4 and Table 5 for default value of GPO1 and GPO2 at PMIC power-up).

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, which in turn control the power management. The operating modes are described in the table below according to the application illustrated in Figure 1, Figure 2, and Figure 3.

Table 6. Operating modes

Operating mode	PMIC state	PWR_ON	Description	Notes
Run	POWER_ON	1	V_{DDIO} power-on V_{DDCORE} power-on. V_{DDCPU}(1) power-on (nominal or overdrive). System clock on. Peripherals power-on/off (via GPO1/2). DDR3L active.	V_{DDCPU} fixed to overdrive voltage with application Figure 3.
LPLV-Stop	POWER_ON	0	V_{DDIO} power-on. V_{DDCORE} power-on at lower voltage. V_{DDCPU}(1) power-on at lower voltage. System clocks off. Peripherals power-on/off (via GPO1/2). DDR3L self-refresh.	MPU PWR_ON pin is internally muxed to PWR_LP signal. LPLV-Stop mode not supported with application in Figure 3.
Standby	POWER_ON	0	VDDIO power-on. VDDCORE power-off. VDDCPU(1) power-off. System clocks off. Peripherals power-off (via GPO1/2). DDR3L self-refresh or off.	MPU PWR_ON pin is internally muxed to PWR_ON signal. Standby DDR3L self-refresh not supported with application in Figure 3.
VBAT	NO_SUPPLY	-	Backup domain powered from backup battery if present.	-
Off	Off	-	All regulators power-off. Backup domain powered from backup battery if present.	-

1. *No V_{DDCPU} in application Figure 1 with STPMIC1LA as V_{DDCORE} is merged with V_{DDCPU}.*

5.1.1 Low power modes overview

The MPU supports several low-power modes to reduce power consumption. These modes are described in Section 5.1: Operating modes. The low-power modes supported by the application and their advantages and disadvantages are shown in Table 7.

Table 7. Low-power mode supported by the application

Power mode	Advantages	Disadvantages
Stop	Very fast recovery from Stop to Run mode.	Low power consumption gain.
LPLV-Stop(1)	V_{DDCORE} voltage is lowered. V_{DDCPU} voltage is lowered(2). The current leakage on V_{DDCORE} and V_{DDCPU} domain consumption is reduced.	Few EXTI wake-up sources are available to exit this mode. To exit this mode, time is necessary to restore the lowered supply to its nominal values.
LPLV-Stop2(2)	V_{DDCORE} voltage is lowered. V_{DDCPU} voltage is turned off. The current leakage on V_{DDCORE} is reduced and the current leakage on V_{DDCPU} is saved.	Few EXTI wake-up sources are available to exit this mode. Longer exit recovery duration than LPLV-Stop mode.
Standby (DDR in self-refresh) (1)	Very-low power consumption. All MPU power domains are powered-off except V_{DDIO}. 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-Stop.
Standby (DDR off)	Lowest power consumption. All MPU power domains are powered off except V_{DDIO}. DDR is powered off (suspend to flash).	Few EXTI wakeup sources are available to exit this mode. Longer exit recovery duration than Standby (DDR in self-refresh).

1. *Not supported by the reference design shown in Section 4.3: STPMIC1LD and STM32MP13xD/F with CPU frequency up to 1000 MHz, and V_{DDCPU} merged with V_{DD_DDR}.*
2. *Only supported by the reference design shown in Section 4.2: STPMIC1LD and STM32MP13xD/F with CPU frequency up to 1000 MHz.*

The MPU manages the low-power modes. As described in PWRCTRL1, PWRCTRL2, the power control signals are connected as defined in Table 8.

Table 8. Power control signals

MPU output	PMIC input
PWR_ON	PWRCTRL1

The PWR_ON signal is controlled from the MPU state machine. That is because in low-power mode, no software is running and the PMIC regulators cannot be controlled by any I^2C command from software. (see Table 6. Operating modes)

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

- PMIC xxxx_MAIN_CR registers: settings for run mode behavior
- PMIC xxxx_ALT_CR registers: settings for the targeted low-power mode behavior.

Note: xxxx corresponds to the targeted regulator or GPO.

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

- LPLV-Stop
- Standby mode.

The delays are described in the following section.

5.1.2 PMIC 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 PMIC automatically powers up when the application is powered from a valid power source. When VIN rises above the PMIC V_{INOK_rise} internal threshold (see [1] for more details), it triggers a PMIC turn-on condition as the “AUTO turn-on” bit is set by default in the STPMIC1LA and the STPMIC1LD NVM.

Turn-on conditions

When the application is in off mode (PMIC in off state with V_{IN} present), a turn-on condition is required to power up the PMIC, and then 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 9. PMIC turn-on conditions

Condition	Trigger	Description
AUTO turn-on	Internal	The PMIC starts automatically when the VIN voltage rises above the V_{INOK_rise} threshold. The AUTO turn-on feature is enabled by default in STPMIC1LA and STPMIC1LD NVM.
PONKEY user button pressed	External	PONKEYn/EN pin falling edge (PKEY_EN_CFG = 0 in PMIC NVM: by default in STPMIC1LA and STPMIC1LD 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.

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 10. 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 thePMIC(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)
VIN undervoltage	Yes	VIN 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 above T_{SHDN_Rise} threshold(3). (see Section 6.2.3: TSHDN: thermal shutdown protection)
Overcurrent	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 enters a transitional power-down state. The PMIC then remains in the 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 VIN voltage to rise above the VINOK_rise threshold before the PMIC powers up.*
3. *If restart conditions are met, the PMIC waits for the temperature to decrease below the TSHDN_Fall threshold before the PMIC powers up.*

4. *The watchdog timer is disabled 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 STPMIC1LA and STPMIC1LD NVM).

5.1.3 PMIC power control management (PWRCTRLx)

PMIC PWRCTRL1/2 input signals are dedicated to managing MPU power modes. These signals must be configurated before the MPU enters low-power mode, to ensure expected regulators or GPOx behavior when MPU enters and exits low-power mode.

The PMIC PWRCTRL1/2 signals are independently driven from MPU (typically PWR_ON) to control some PMIC regulators (or GPOs) behavior by setting the appropriate registers as described in [1]. As such, it is possible to define:

- The control source selection of the regulator (PWRCTRL1 or 2).
- The polarity of the respective PWRCTRLx is used to determine whether the signals are active low or active high.

A PMIC PWRCTRL signal aims typically to switch between two control registers: xxxx_MAIN_CR and xxxx_ALT_CR, a PMIC regulator, or a GPO. xxxx corresponds to the targeted regulator or GPO.

For example, when a PWRCTRL signal is driven by MPU from high to low state, the PMIC internally switches from a main (xxxx_MAIN_CR) control register content to an alternate (xxxx_ALT_CR) control register content, and vice versa.

5.1.4 PMIC mask-reset option

If the application needs to have one or several PMIC regulators to be kept 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 targeted BUCK converters
- The PMIC LDOS_MRST_CR register to targeted LDO
- The PMIC GPOS_MRST_CR register to targeted GPO

A reset sequence is triggered after the PMIC RSTn signal is asserted by the MPU or the user reset push button. Refer to [1] for more details on the PMIC mask_reset option.

This is typically the case for the LDO2 powering the MPU V_{DD} power domains. The power cycle on V_{DD} must be masked by setting the PMIC register: LDOS_MRST_CR [1] = 1.

This prevents losing the content in:

- The MPU backup RAM
- The MPU backup register content
- The JTAG debug interface

Note: These settings must be programmed by the MPU software bootloader via an I²C command to the PMIC after each application power-up. This is necessary because the content of BUCKS_MRST_CR, LDOS_MRST_CR and GPOS_MRST_CR is reset at the end of a PMIC reset cycle.

5.1.5 MPU internal timer for low power mode management

EADLY timer

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

POPL timer

The POPL timer is a programmable timer used to force the microprocessor unit (MPU) into standby mode for a minimum duration. When entering standby mode, the PWR_ON signal goes low for the minimum POPL duration, forcing V_{DDCORE} power supply voltages to drop before exiting standby mode. This ensures that the MPU core, CPU, and flash memory domains restart properly if a wake-up event occurs immediately after the MPU enters standby mode.

The software sets the POPL timer prior to standby mode entry.

Recommended values: PWR_CR3.POPL = 3 ms minimum for Section 5.3.2: Standby mode (DDR3L in self-refresh), Section 5.3.3: Standby mode (DDR3L off), Section 5.4.3: Standby mode (DDR3L in self-refresh), Section 5.4.4: Standby mode (DDR3L off), and Section 5.5.1: Standby mode (DDR3L off).

Note: The POPL timer may be set to a higher value if some application peripherals require their power supplies to be fully discharged to restart properly. In such cases, the POPL duration should be set according to the duration required to drop the peripheral power supply voltages. These are supplied by discrete regulators such as the 3.3 V supply illustrated in Figure 1, Figure 2, or Figure 3.

PWRLP_TEMPO timer

The PWRLP_TEMPO is the delay between the time when the system exits LPLV-Stop2 mode and the moment when it is permitted to enable the phase-locked loops (PLLs). This delay ensures the system can provide a clock to the CPU and enter Run mode. The delay is linked to the core domain (V_{DDCORE}) and should be set in the RCC_PWRLPDLYCR.PWRLP_DLY[21:0] register bitfield prior to entering low-power mode.

The PWRLP_DLY timer should be set by software before entering low-power mode. Recommended values are provided in Section 5.4.2, depending on the targeted low-power mode.

MRD timer

The MRD timer 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. It can be set to 0 when using a power management integrated circuit (PMIC).

$t_{VDDCORE\_TEMPO}$ delay

The $t_{VDDCORE\_TEMPO}$ delay (340 µs typical) is an internal and fixed delay initiated to wait for V_{DDCORE} to reach the Run mode operating supply level when V_{DDCORE} is enabled and reaches the threshold VTH_VDDCORE. As long as V_{DDCORE} is below VTH_VDDCORE, the core domain remains in reset. This delay is used during power-up or when the system exits standby low-power mode.

t_{VDDCPU_TEMPO} delay

The t_{VDDCPU_TEMPO} delay (340 µs typical) is an internal and fixed delay initiated to wait for V_{DDCPU} to reach the Run mode operating supply level when V_{DDCPU} is enabled and reaches the threshold VTH_VDDCPU. As long as V_{DDCPU} is below VTH_VDDCPU, the CPU domain remains in reset. This delay is used during power-up or when the system exits standby low-power mode.

$t_{SEL\_VDDCORETEMPO}$ delay

The $t_{SEL\_VDDCORETEMPO}$ delay (380 µs typical) is an internal and fixed delay initiated to wait for V_{DDCORE} to reach the Run mode operating supply level when the system exits LPLV-Stop mode.

$t_{SEL\_VDDCPUTEMPO}$ delay

The $t_{SEL\_VDDCPUTEMPO}$ delay (380 µs typical) is an internal and fixed delay initiated to wait for V_{DDCPU} to reach the Run mode operating supply level when the system exits LPLV-Stop mode.

5.2 Application power-up/power-down sequence

The power-up sequence is the transition managed by the PMIC from application power-off and to application Run mode triggered by a turn-on condition (see Section 5.1.2: PMIC turn-on/turn-off conditions). Typically, when the application is connected to an external power supply, the PMIC power up automatically when V_{IN} rises above PMIC V_{INOK_rise} threshold. When power-up ends, the PMIC releases the RSTn signal. Once the RSTn signal is released, the MPU boots (including the DDR3L initialization by MPU software) and the MPU reaches system run mode.

Note: The STPMIC1LA and the STPMIC1LD have the AUTO_TURN_ON bit and POnKEYn feature enabled by default in PMIC NVM.

The power down sequence is the transition managed by the PMIC from application Run mode to application power-off triggered by a turn-off condition (see Turn-off conditions).

5.2.1 STPMIC1LA and STM32MP13xA/C power-up triggered by main supply (VIN) plugin and power-down by software shutdown

The application power-up and power-down sequence shown in Figure 4 is based on the reference design in Figure 1 and is detailed below:

1. The application has no power.
2. A power supply is connected to the application: VIN voltage rises.

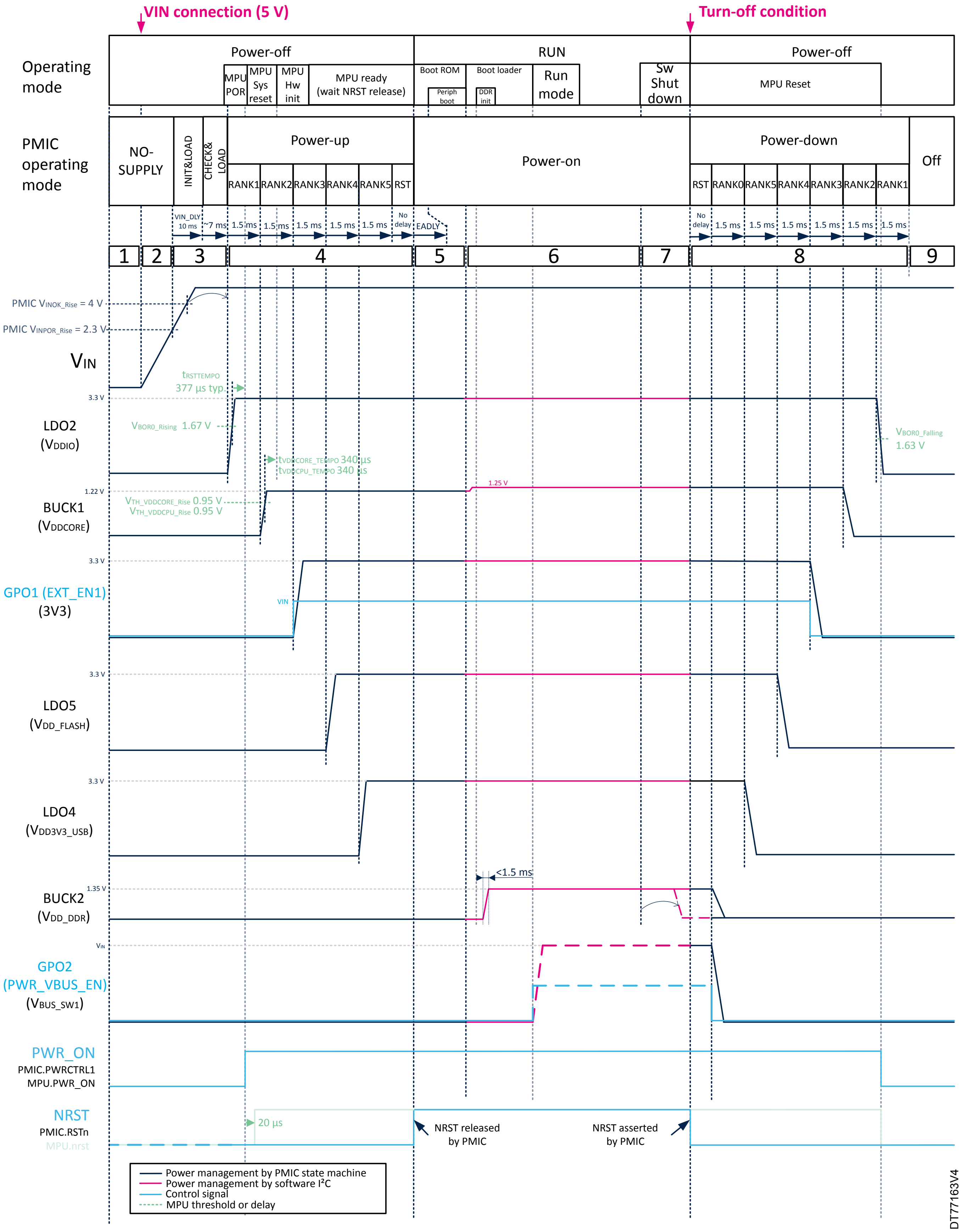
3. Once VIN voltage is above PMIC V_{INPOR_Rise} (2.3 V typ.) threshold:
   a. The PMIC goes to INIT&LOAD transitional state to preload its NVM contents and checks its integrity. If the PMIC NVM integrity is valid, the PMIC initializes, launches, and executes the VIN_DLY = 10 ms.
   b. Once VIN_DLY elapses, the PMIC states machine goes in the CHECK&LOAD transitional state (as the AUTO_TURN_ON bit is set in PMIC NVM: see 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 PMIC NVM. This delay can be changed by reprograming the NVM.

Note: The PMIC CHECK&LOAD duration is about 7 ms typ.

4. Once the CHECK&LOAD state ends and VIN voltage is above the PMIC V_{INOK_rise} (4 V typ.) threshold, the PMIC starts a power-up sequence. The PMIC regulators follow the power-up sequence predefined in PMIC NVM:
   a. The PMIC asserts the RSTn signal.
   b. RANK1 (1.5 ms): The LDO2 (V_{DDIO}) regulator is turned on at 3.3 V. Once V_{DDIO} voltage is above MPU VBOR0 rising threshold (1.67 V typ.), an MPU $t_{RSTTEMPO}$ (377 µs typ.) delay is started. Once $t_{RSTTEMPO}$ elapses, the MPU PWR_ON signal goes high, and the MPU enters in system reset.
   c. RANK2 (1.5 ms): The BUCK1 (V_{DDCORE}) regulator is turned on at 1.22 V. Once V_{DDCORE} voltage is above MPU $V_{TH\_VDDCORE}$ rising threshold (950 mV typ.), an MPU $t_{VDDCORE\_TEMPO}$ delay is started to wait for V_{DDCORE} voltage to reach the minimum operating voltage. Once $t_{VDDCORE\_TEMPO}$ elapses, the MPU starts the HSI oscillators, then performs internal hardware initialization. The MPU then waits for NRST to release.
   d. RANK3 (1.5 ms): The GPO1 (EXT_EN1) regulator is turned on. The 3V3 discrete SMPS regulator is turned on and the 3V3 voltage rises.
   e. RANK4 (1.5 ms): LDO5 (V_{DD_FLASH}) regulator is turned on at 3.3 V.
   f. RANK5 (1.5 ms): LDO4 (V_{DD3V3_USB}) regulator is turned on.
   g. Once RANK5 is ended, the PMIC releases the RSTn that releases MPU NRST.
5. Once NRST is released, the MPU enters Run mode:
   a. The CPU starts to execute the boot ROM: EADLY timer starts (refer to EADLY timer for more information).
   b. Once EADLY elapses, the boot ROM reads, verifies, and executes the bootloader from the external flash memory (for example: the SD card as illustrated in Figure 1).
6. The bootloader software performs initializations, then loads and executes the application software:
   a. Set PMIC BUCK1 (V_{DDCORE}) at 1.25 V (V_{DDCORE} and V_{DDCPU} nominal voltage).
   b. Enable DDR regulators: enable BUCK2 (V_{DD_DDR}) at 1.35 V. Then, set a 1.5 ms timer to wait for DDR voltages stabilization.
   c. Once the 1.5 ms timer elapses, the bootloader software initializes the DDR controller and DDR memory IC.
   d. The bootloader loads the application software into DDR3L and executes it.
   e. The system runs.
7. When a shutdown request occurs:
   a. The software prepares to power-off properly.
   b. The software shuts down the DDR3L: disable BUCK2 (recommended but not mandatory).
   c. The software sends the "SWOFF" command to PMIC to power-down.
8. The PMIC performs a power-down sequence:
   a. The PMIC asserts the RSTn, asserting the MPU NRST signal.
   b. RANK0 (1.5 ms): The BUCK2 (V_{DD_DDR}) regulator is turned off, and GPO2 is turned off.
   c. RANK5 (1.5 ms): The LDO4 (V_{DD3V3_USB}) regulator is turned off.
   d. RANK4 (1.5 ms): The LDO5 (V_{DD_FLASH}) regulator is turned off.
   e. RANK3 (1.5 ms): GPO1 (EXT_EN1) regulator is turned off.
   f. RANK2 (1.5 ms): BUCK1 (V_{DDCORE}) regulator is turned off.
   g. RANK1 (1.5 ms): LDO2 (V_{DDIO}) regulator is turned off. Once V_{DDIO} voltage is below MPU VBOR0 falling threshold (1.63 V typ.), the MPU PWR_ON signal goes low, and the MPU enters in POR.
9. The PMIC is in off mode: the application is powered off.

Figure 4. Power-up and power-down sequence STM32MP13xA/C with STPMIC1LA



5.2.2 STPMIC1LD and STM32MP13xD/F power-up triggered main supply (VIN) plugin and power-down by software shutdown

The application power-up and power-down sequence shown in Figure 5 is based on the reference design in Figure 2 and is detailed below:

1. The application has no power
2. A power supply is connected to the application: V_{IN} voltage rises.
3. Once V_{IN} voltage is above PMIC V_{INPOR_Rise} (2.3 V typ.) threshold:
   a. The PMIC goes to 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 states machine goes in CHECK&LOAD transitional state (as AUTO_TURN_ON bit is set in PMIC NVM: see PMIC turn-on/turn-off conditions)

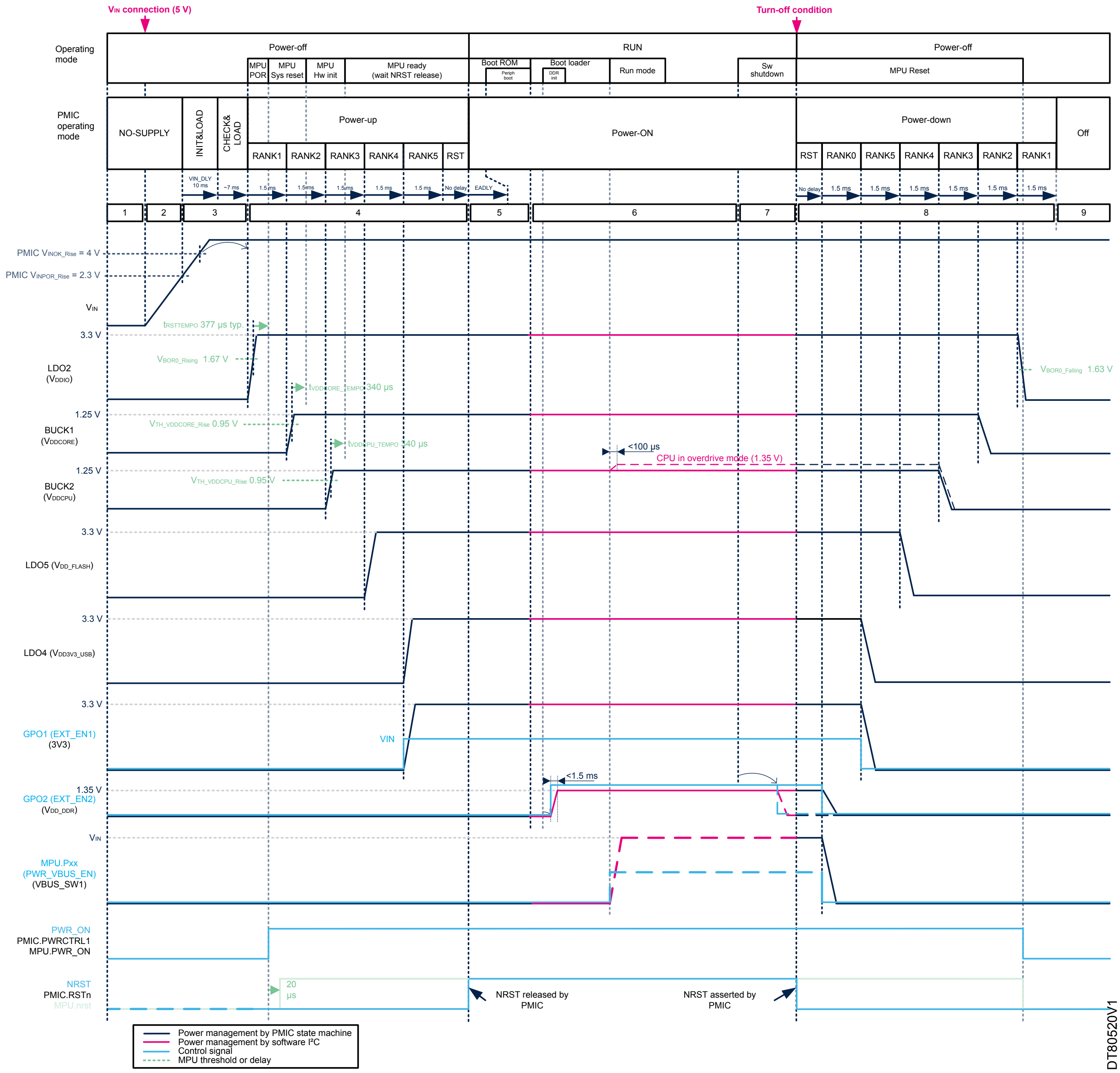
Note: The VIN_DLY timer is a passive delay used to wait for V_{IN} voltage stabilization. It is set to 10 ms by default in PMIC NVM. This delay can be changed by reprograming NVM.

Note: The PMIC CHECK&LOAD duration is about 7 ms typ.

4. Once the CHECK&LOAD states ends and V_{IN} voltage is above PMIC V_{INOK_rise} (4 V typ.) threshold, the PMIC starts a power-up sequence. The PMIC regulators follow the power-up sequence predefined in PMIC NVM:
   a. The PMIC assert RSTn signal
   b. RANK1 (1.5 ms): The LDO2 (V_{DDIO}) regulator is turned on at 3.3 V. Once V_{DDIO} voltage is above MPU VBOR0 rising threshold (1.67 V typ.) a MPU $t_{RSTTEMPO}$ (377 µs typ.) delay is started. Once $t_{RSTTEMPO}$ elapses, the MPU PWR_ON signal goes high, and the MPU enters in system reset.
   c. RANK2 (1.5 ms): The BUCK1 (V_{DDCORE}) regulator is turned on at 1.25 V. Once V_{DDCORE} voltage is above MPU $V_{TH\_VDDCORE}$ rising threshold (950 mV typ.), an MPU $t_{VDDCORE\_TEMPO}$ delay is started to wait for V_{DDCORE} voltage to reach minimum operating voltage. Once $t_{VDDCORE\_TEMPO}$ elapses the MPU starts the HSI oscillators, then perform internal hardware initialization.
   d. RANK3 (1.5 ms): The BUCK2 (V_{DDCPU}) regulator is turned on at 1.25 V. Once V_{DDCPU} voltage is above MPU V_{TH_VDDCPU} rising threshold (950 mV typ.), an MPU t_{VDDCPU_TEMPO} delay is started to wait for V_{DDCPU} voltage to reach minimum operating voltage. The MPU is ready to run and it waits for PMIC to release the NRST.
   e. RANK4 (1.5 ms): LDO5 (V_{DD_FLASH}) regulator is turned on at 3.3 V.
   f. RANK5 (1.5 ms): LDO4 (V_{DD3V3_USB}) regulator is turned on and the GPO1 (EXT_EN1) is turned on. The 3V3 discrete SMPS regulator is enabled and the 3V3 voltage rises.
   g. Once RANK5 is ended, the PMIC releases the RSTn that releases MPU NRST.
5. Once NRST is released, the MPU enters RUN mode:
   a. The CPU starts to execute the boot ROM: EADLY timer starts (refer to EADLY timer for more information)
   b. Once EADLY elapses, the boot ROM reads, verifies, and executes the bootloader from the external flash memory (for example: the SD card as illustrated in Figure 2).
6. The bootloader software performs initializations then loads and executes the application software:
   a. Enable DDR regulator: enable the PMIC GPO2 (EXT_EN2). The V_{DD_DDR} discrete SMPS regulator is enabled and the V_{DD_DDR} voltage rises. Then, set a 1.5 ms timer to wait for DDR voltages stabilization.
   b. Once the 1.5 ms timer elapses, the bootloader software initializes the DDR controller and DDR memory IC.
   c. The bootloader loads the application software into DDR3L and executes it.
   d. The system runs.
   e. The application software can switch to Run from nominal mode (up to 650 MHz) to overdrive mode (up to 1000 MHz):
      i. Set the BUCK2 (V_{DDCPU}) from 1.25 V to 1.35 V
      ii. Set a 100 µs timer to wait for VDDCPU voltage stabilization (see [1] for detail about BUCK2 slew rate)
      iii. Once the 100 µs timer elapses, the application software switches the CPU frequency up to 1000 MHz.
7. When a shutdown request occurs:
   a. The software prepares to power-off properly
   b. The software shuts down the DDR3L: disable the PMIC GPO2 (EXT_EN2). The V_{DD_DDR} discrete SMPS regulator is disabled, and the V_{DD_DDR} voltage drops.
   c. The software sends the “SWOFF” command to PMIC to power-down

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 done by software, the GPO2 (EXT_EN2) is turned off. The V_{DD_DDR} discrete SMPS regulator is disabled, and the V_{DD_DDR} voltage drops.
   c. RANK5 (1.5 ms): The LDO4 (V_{DD3V3_USB}) regulator is turned off and the GPO1 (EXT_EN1) is turned off. The 3V3 discrete SMPS regulator is turned off and the 3V3 voltage drops.
   d. RANK4 (1.5 ms): The LDO5 (V_{DD_FLASH}) regulator is turned off.
   e. RANK3 (1.5 ms): BUCK2 (V_{DDCPU}) regulator is turned off.
   f. RANK2 (1.5 ms): BUCK1 (V_{DDCORE}) regulator is turned off.
   g. RANK1 (1.5 ms): LDO2 (V_{DDIO}) regulator is turned off. Once V_{DDIO} voltage is below MPU VBOR0 falling threshold (1.63 V typ.), the MPU PWR_ON signal goes low, and the MPU enters in POR.
9. The PMIC is in off mode: the application is powered off.

Figure 5. Power-up and power-down sequence STM32MP13xD/F with STPMIC



5.2.3 STPMIC1LD and STM32MP13xD/F power-up triggered main supply (VIN) plugin and power-down by software shutdown (V_{DDCPU} merged with V_{DD_DDR})

The application power-up and power-down sequence shown in Figure 6 is based on the reference design in Figure 3 and is detailed below:

1. The application has no power.
2. A power supply is connected to the application: V_{IN} voltage rises.
3. Once VIN voltage is above the PMIC VINPOR_Rise (2.3 V typ.) threshold:
   a. The PMIC goes to INIT&LOAD transitional state to preload its NVM contents and check its integrity. If the PMIC NVM integrity is valid, the PMIC initializes, launches, and executes the VIN_DLY = 10 ms.
   b. Once VIN_DLY elapses, the PMIC state machine transitions to the CHECK&LOAD state (as the AUTO_TURN_ON bit is set in PMIC NVM; see PMIC turn-on/turn-off 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 PMIC NVM. This delay can be changed by reprogramming NVM.

Note: The PMIC CHECK&LOAD duration is approximately 7 ms typ.

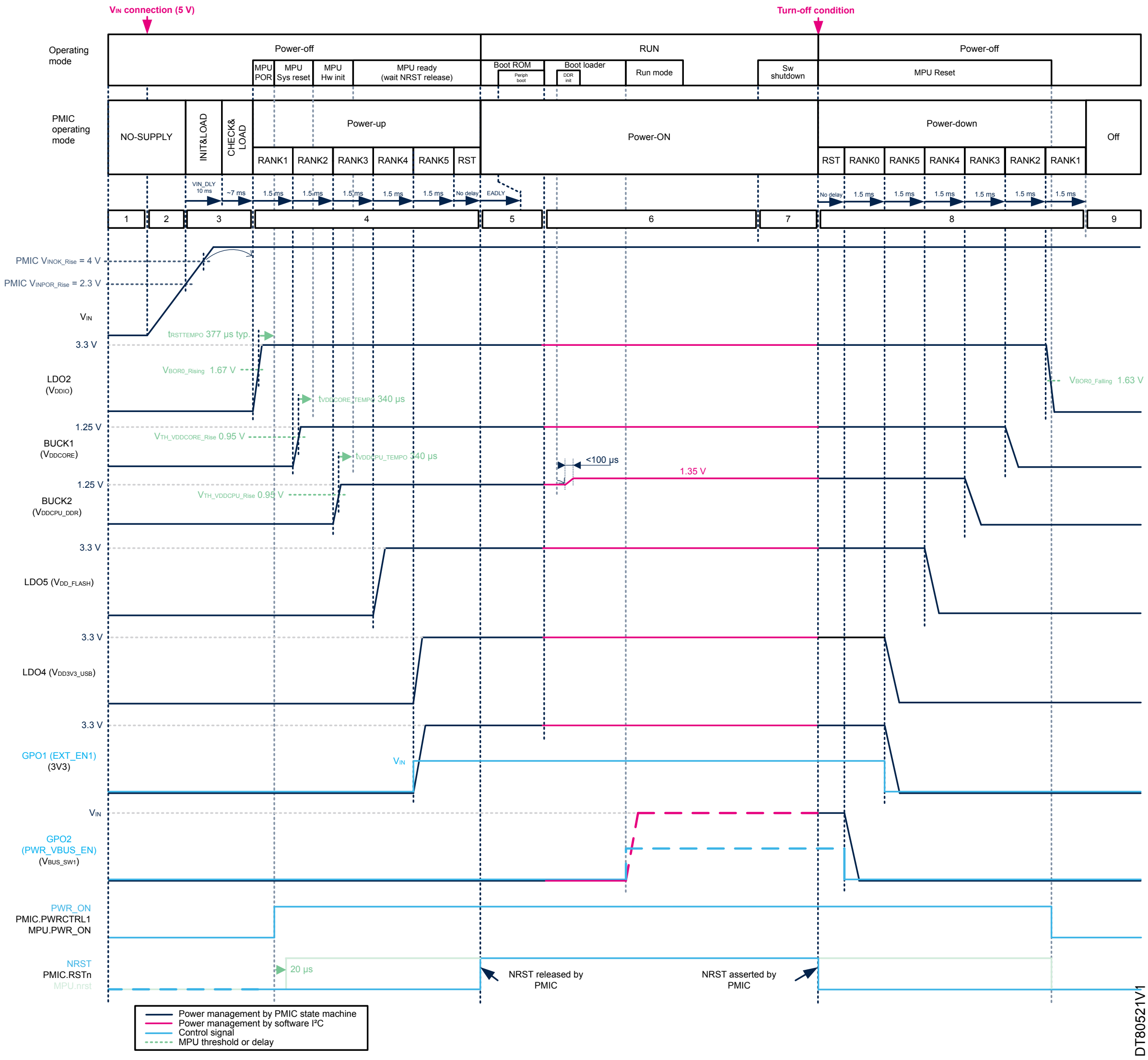
4. Once the CHECK&LOAD state ends and V_{IN} voltage is above the PMIC VINOK_rise (4 V typ.) threshold, the PMIC starts a power-up sequence. The PMIC regulators follow the power-up sequence predefined in PMIC NVM:
   a. The PMIC asserts the RSTn signal.
   b. RANK1 (1.5 ms): The LDO2 (V_{DDIO}) regulator is turned on at 3.3 V. Once V_{DDIO} voltage is above the MPU VBOR0 rising threshold (1.67 V typ.), an MPU $t_{RSTTEMPO}$ (377 µs typ.) delay is started. Once $t_{RSTTEMPO}$ elapses, the MPU PWR_ON signal goes high, and the MPU enters system reset.
   c. RANK2 (1.5 ms): The BUCK1 (V_{DDCORE}) regulator is turned on at 1.25 V. Once V_{DDCORE} voltage is above the MPU VTH_VDDCORE rising threshold (950 mV typ.), an MPU $t_{VDDCORE\_TEMPO}$ delay is started to allow V_{DDCORE} voltage to reach the minimum operating voltage. Once $t_{VDDCORE\_TEMPO}$ elapses, the MPU starts the HSI oscillators and performs internal hardware initialization.
   d. RANK3 (1.5 ms): The BUCK2 (V_{DDCPU_DDR}) regulator is turned on at 1.25 V. Once V_{DDCPU_DDR} voltage is above the MPU VTH_VDDCPU rising threshold (950 mV typ.), an MPU t_{VDDCPU_TEMPO} delay is started to allow V_{DDCPU_DDR} voltage to reach the minimum operating voltage. The MPU is ready to run and waits for the PMIC to release the NRST.

Note: It is recommended to reprogram the STPMIC1LD NVM to set the BUCK2 output voltage to 1.35 V by default instead of 1.25 V to reach the DDR3L nominal operating voltage (1.35 V) at power-up.

   e. RANK4 (1.5 ms): LDO5 (V_{DD_FLASH}) regulator is turned on at 3.3 V.
   f. RANK5 (1.5 ms): LDO4 (V_{DD3V3_USB}) regulator is turned on, and the GPO1 (EXT_EN1) is turned on. The 3V3 discrete SMPS regulator is enabled, and the 3V3 voltage rises.
   g. Once RANK5 ends, the PMIC releases the RSTn, which releases the MPU NRST.
5. Once NRST is released, the MPU enters RUN mode:
   a. The CPU starts to execute the boot ROM: EADLY timer starts (refer to EADLY timer for more information).
   b. Once EADLY elapses, the boot ROM reads, verifies, and executes the bootloader from the external flash memory (for example, the SD card as illustrated in Figure 3).
6. The bootloader software performs initializations, then loads and executes the application software:
   a. Set the BUCK2 (V_{DDCPU_DDR}) from 1.25 V to 1.35 V to reach the DDR3L nominal operating voltage. Then, set a 100 µs timer to wait for V_{DDCPU_DDR} voltage stabilization.
   b. Once the 100 µs timer elapses, the bootloader software initializes the DDR controller and DDR memory IC.
   c. The bootloader loads the application software into DDR3L and executes it.
   d. The system runs with a CPU frequency of up to 1000 MHz.
7. When a shutdown request occurs:
   a. The software prepares to power off properly.
   b. The software sends the “SWOFF” command to the PMIC to power down.

8. The PMIC performs a power-down sequence:
   a. The PMIC asserts the RSTn, asserting the MPU NRST signal.
   b. RANK0 (1.5 ms): The GPO2 is turned off.
   c. RANK5 (1.5 ms): The LDO4 (V_{DD3V3_USB}) regulator is turned off and the GPO1 (EXT_EN1) is turned off. The 3V3 discrete SMPS regulator is turned off, and the 3V3 voltage drops.
   d. RANK4 (1.5 ms): The LDO5 (V_{DD_FLASH}) regulator is turned off.
   e. RANK3 (1.5 ms): BUCK2 (V_{DDCPU_DDR}) regulator is turned off.
   f. RANK2 (1.5 ms): BUCK1 (V_{DDCORE}) regulator is turned off.
   g. RANK1 (1.5 ms): LDO2 (V_{DDIO}) regulator is turned off. Once V_{DDIO} voltage is below the MPU VBOR0 falling threshold (1.63 V typ.), the MPU PWR_ON signal goes low, and the MPU enters POR.
9. The PMIC is in off mode: the application is powered off.

Figure 6. Power-up and power-down sequence STM32MP13xD/F with STPMIC1LD (VDDCPU merged with VDD_DDR)



5.2.4 Power-down triggered by PMIC hard fault (safety management)

When the PMIC detects a hard fault (see Section 6.2), it triggers a turn-off condition followed by a power-down sequence similar to step 8 in Figure 4, Figure 5, or Figure 6. Once the power-down sequence ends, the PMIC can either restart (similar to step 4 in Figure 4, Figure 5, or Figure 6) or go in FAIL_SAFE_LOCK state depending on safety management settings (see Section 6.1).

By default, the STPMIC1LA and the STPMIC1D always restarts after a turn-off condition triggered by a hard fault.

5.2.5 Power-down triggered by main supply removal (VIN)

The application in Figure 1, Figure 5, or Figure 6 is powered off by a power removal (VIN).

Once the VIN supply is below V_{INOK_fall}, the PMIC asserts an NRST and then enters power-down as shown in Figure 4, Figure 5, or Figure 6 step 8.

Limitation: when the main power is removed, the VIN voltage drops very quickly to the V_{INOK_fall} value, in less than a few milliseconds (depending on system activity). Only then the power-down sequence starts. As soon as the PMIC asserts an NRST, system activity is immediately stopped and power consumption drops, slowing the VIN drop. Nevertheless, VIN may drop below the PMIC $V_{IN\_POR\_fall}$ threshold before the power-down sequence ends. In this case, the PMIC regulator pull-down discharge resistors are no longer controlled by the PMIC. The output regulator decoupling capacitors discharge in an uncontrolled way. A bulk decoupling capacitor (a few hundred µF) may be inserted on the VIN path to limit VIN drop speed.

Note: If the application main supply is immediately inserted after VIN drops below PMIC $V_{IN\_POR\_fall}$ threshold, the PMIC goes in INIT&LOAD state then in CHECK&LOAD state (see Figure 4 step 3) where the regulator pull-down discharge resistors are enabled. Thanks to this, output regulator decoupling capacitors are properly discharged before going into the power-up step. (See Figure 4, Figure 5, or Figure 6 step 4.)

5.3 Low-power mode management with STPMIC1LA and STM32MP13xA/C

5.3.1 LPLV-Stop mode

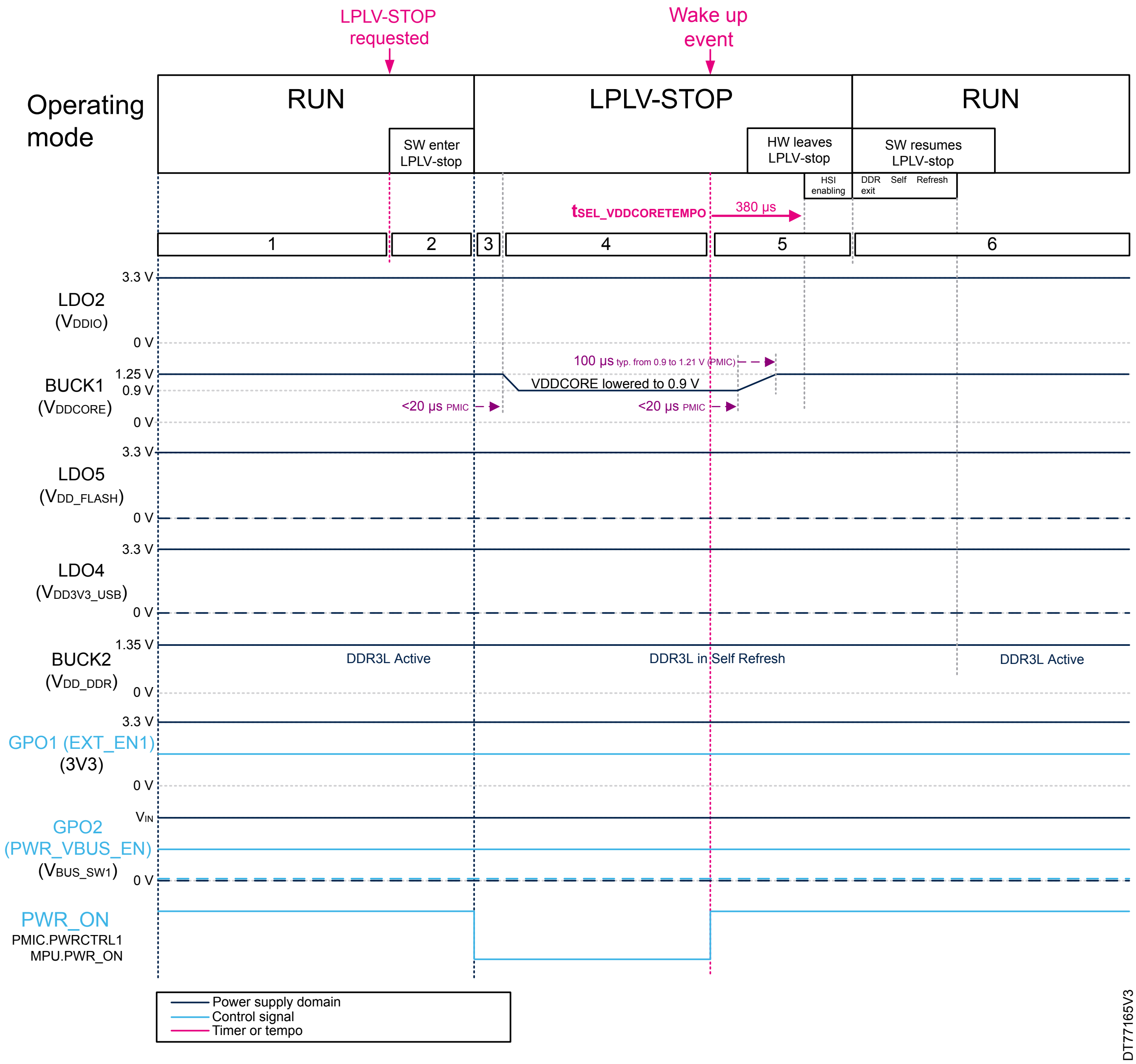
The LPLV-Stop1 mode is described below and is shown in Figure 7 based on the implementation shown in Figure 1.

1. The application is powered up and operates in Run mode. PWR_ON is in high state. In this application, the PWR_LP signal is multiplexed on the PWR_ON pin (PWR_CR1.LPCFG bit is enabled).
2. When the LPLV-Stop mode is requested, the software prepares to enter LPLV-Stop mode:
   a. The MPU performs internal settings such as:
      i. Disabling PWRLP_TEMPO (set in RCC_PWRLPDLYCR.PWRLP_DLY[21:0] = 0).
      ii. Stop non required system clocks.
      iii. Setting the DDR3L to self-refresh.
   b. The MPU performs the PMIC settings (see Table 11).
   c. The MPU PWR_CR1.LPDS and PWR_CR1.LVDS bit are enabled.
3. Once the system is in LPLV-Stop:
   a. The MPU PWR_ON output is deasserted (PMIC PWRCTRL1 signal goes low).
4. The PMIC regulators, which are affected to the PWR_ON, take the configuration set in the xxxx_ALT_CR registers (see Table 11) after ~ 20 µs (internal PMIC delay):
   a. The V_{DDCORE} regulator output decreases to retention voltage (from 1.25 V to 0.9 V).
5. On a wake-up event, the MPU leaves the LPLV-Stop:
   a. The MPU PWR_ON output signal is asserted (the PMIC PWRCTRL1 signal goes high), and the MPU starts the $t_{SEL\_VDDCORETEMPO}$ internal delay to wait for the V_{DDCORE} to switch from retention voltage (0.9 V) to minimum operating voltage (1.21 V).
   b. The PMIC regulators which are affected to the PWR_ON take the configuration set in the xxxx_MAIN_CR registers (see Table 11) after ~ 20 µs (internal PMIC delay):
      i. The V_{DDCORE} regulator switches from retention voltage (0.9 V) to minimal nominal operating voltage (1.21 V) in 100 µs typical and converges to nominal operating voltage (1.25 V). Accordingly, V_{DDCORE} voltage is ready before the $t_{SEL\_VDDCORETEMPO}$ MPU delay elapses.
   c. Once the $t_{SEL\_VDDCORETEMPO}$ internal delay elapses, clocks are enabled.
6. Once the clocks are stable, the MPU goes immediately in Run mode (as the PWRLP_TEMPO is bypassed) and the software resumes LPLV-Stop: DDR3L exits from self-refresh.

Table 11. PMIC configuration for LPLV-Stop mode

Regulator	PWRCTRLx affectation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	V_{OUT}	Configuration	V_{OUT}
BUCK1 (V_{DDCORE})	PWR_ON	On	1.25	On	0.9
BUCK2 (V_{DD_DDR})	PWR_ON	On	1.35	On	1.35
LDO2 (V_{DDIO})	-	On	3.3	-	-
LDO3 (free)	-	-	-	-	-
LDO4 (V_{DD3V3_USB})	-	On/off	N/A	-	N/A
LDO5 (V_{DD_FLASH})	PWR_ON	On/off	3.3	On/off	3.3
GPO1 (3V3)	PWR_ON	On	N/A	On	N/A
GPO2 (V_{BUS_SW1})	-	On/off	N/A	-	N/A

Figure 7. **LPLV-Stop sequence**



5.3.2 Standby mode (DDR3L in self-refresh)

The Standby mode is used when a very-low power consumption is required. Most of the PMIC regulators are switched off. The content of MPU registers and memories are lost except for the backup domains (V_{DDIO} is kept enabled). The DDR3L is set in self-refresh (V_{DD_DDR} is kept enabled) to maintain the system in "suspend to RAM."

The Standby mode with DDR3L in self-refresh is described in the Figure 8 based on the implementation shown in Figure 1.

1. The application is powered up and operates in Run mode. PWR_ON is in a high state.

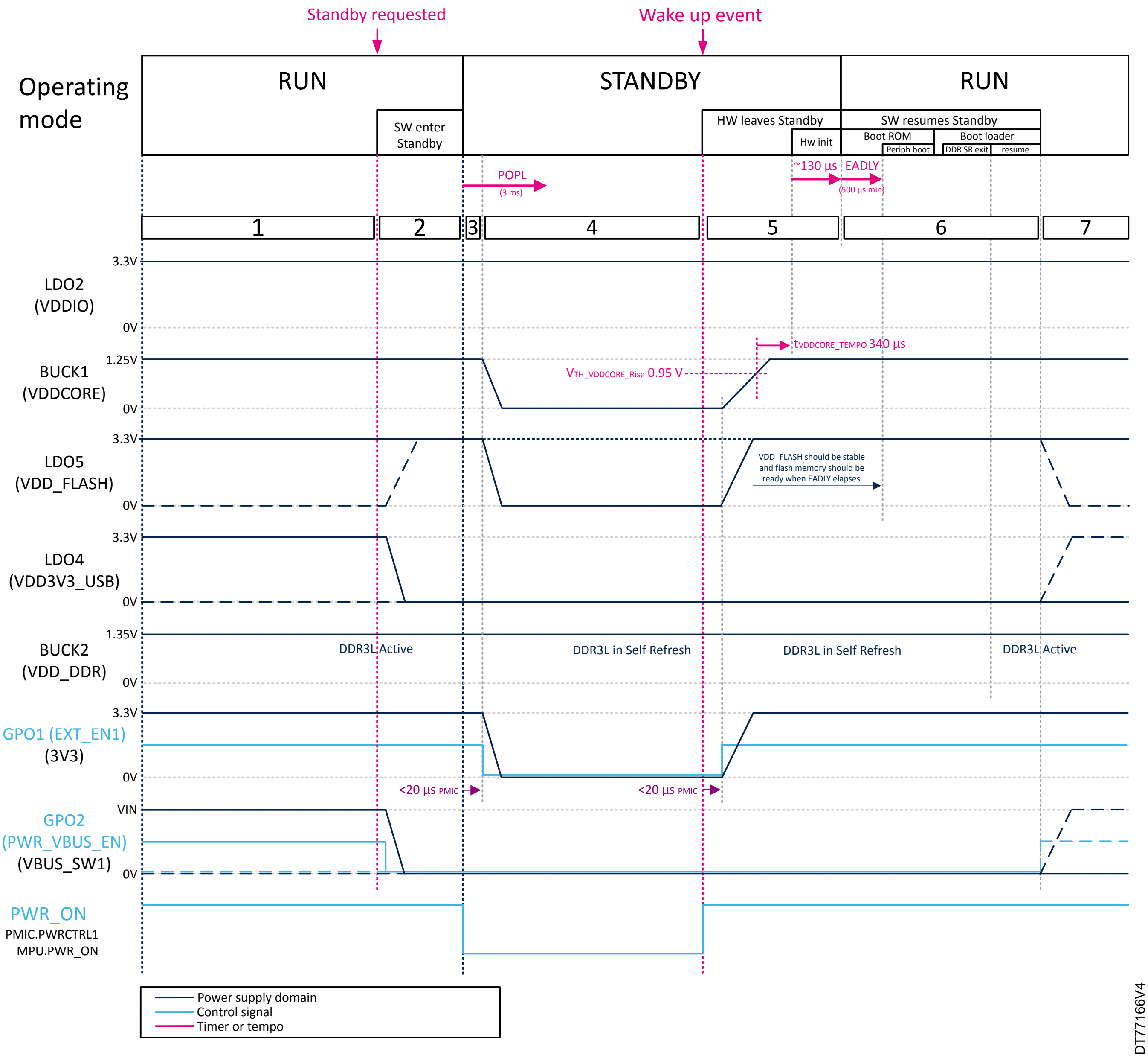
2. When the Standby mode is requested, the software prepares to enter Standby mode:
   a. The MPU performs internal settings such as:
      i. Setting the POPL timer (see POPL timer), to define a minimum pulse duration of PWR_ON ensuring V_{DDCORE} voltage full discharge before restarting.
      ii. Set the EADLY timer (see EADLY timer).
      iii. Stop non required system clocks.
      iv. Setting DDR3L to self-refresh.
   b. The MPU configures the PMIC as described in Table 12.
      i. If turned off, the V_{DD_FLASH} regulator is turned on to enable V_{DD_FLASH} when leaving Standby mode for peripheral boot.
      ii. If turned on, the V_{DD3V3_USB} regulator is turned off.
      iii. If turned on, the V_{BUS_SW1} power switch (controlled by PMIC GPO2) is turned off.
   c. The PWR_MPUCR.PDDS bit is enabled (Standby mode is allowed).
3. Once the system is in Standby mode:
   a. The MPU PWR_ON signal is deasserted (PMIC PWRCTRL1 signal goes low).
   b. The POPL timer is started to keep the application in Standby for a minimum POPL timer duration.
4. The PMIC regulators affected to PWR_ON take the configuration set in the xxxx_ALT_CR registers (see Table 12) after ~ 20 µs (internal PMIC delay):
   a. The V_{DDCORE} regulator is turned off.
   b. The V_{DD_FLASH} regulator is turned off.
   c. The 3V3 regulator (controlled by PMIC GPO1) turns off.
5. On a wake-up event, the MPU leaves Standby mode (MPU waits for the POPL timer to elapse before leaving Standby mode):
   a. The MPU PWR_ON signal is asserted (PMIC PWRCTRL1 signal goes high).
   b. The PMIC regulators affected to the PWR_ON take the configuration set in the xxxx_MAIN_CR registers (see Table 12) after ~ 20 µs (internal PMIC delay):
      i. The V_{DDCORE} regulator is turned on.
      ii. The V_{DD_FLASH} regulator is turned on.
      iii. The 3V3 regulator (controlled by PMIC GPO1) is turned on.
   c. Once the V_{DDCORE} voltage reaches the $V_{TH\_VDDCORE}$ threshold, an MPU $t_{VDDCORE\_TEMPO}$ delay is started to wait for V_{DDCORE} voltage to reach the minimum operating voltage. Once $t_{VDDCORE\_TEMPO}$ elapses, the MPU performs internal hardware initialization.
6. Once the MPU ends internal hardware initialization, the MPU enters Run mode:
   a. The CPU starts to execute the boot ROM: EADLY timer is started (refer to EADLY timer for more information).
   b. Once EADLY elapses, the boot ROM reads, verifies, and executes the bootloader from the external flash memory (for example: the SD card as illustrated in Figure 1).
   c. The software detects an "exit from Standby mode": it exits DDR from self-refresh and resumes the application software.
7. The system runs the application software:
   a. Peripherals switched off before entering Standby mode (such as USB) can be resumed.

Table 12. PMIC configuration for Standby DDR in self-refresh

Regulator	PWRCTRLx affectation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	V_{OUT}	Configuration	V_{OUT}
BUCK1 (V_{DDCORE})	PWR_ON	On	1.25	Off	-
BUCK2 (V_{DD_DDR})	PWR_ON	On	1.35	On	1.35
LDO2 (V_{DDIO})	-	On	3.3	-	-
LDO3 (free)	-	-	-	-	-

Regulator	PWRCTRLx affectation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	V_{OUT}	Configuration	V_{OUT}
LDO4 (V_{DD3V3_USB})	-	Off	N/A	-	N/A
LDO5 (V_{DD_FLASH})	PWR_ON	On	3.3	Off	-
GPO1 (3V3)	PWR_ON	On	N/A	Off	N/A
GPO2 (V_{BUS_SW1})	-	Off	N/A	-	N/A

Figure 8. Standby (DDR in self-refresh) sequence



5.3.3 Standby mode (DDR3L off)

The Standby mode is used when a very-low power consumption is required. Most of the PMIC regulators are switched off. The content of MPU registers and memories are lost except for the backup domain (V_{DDIO} is kept enabled). The DDR3L is powered off (V_{DD_DDR} is disabled), so the system is in "suspend to flash."

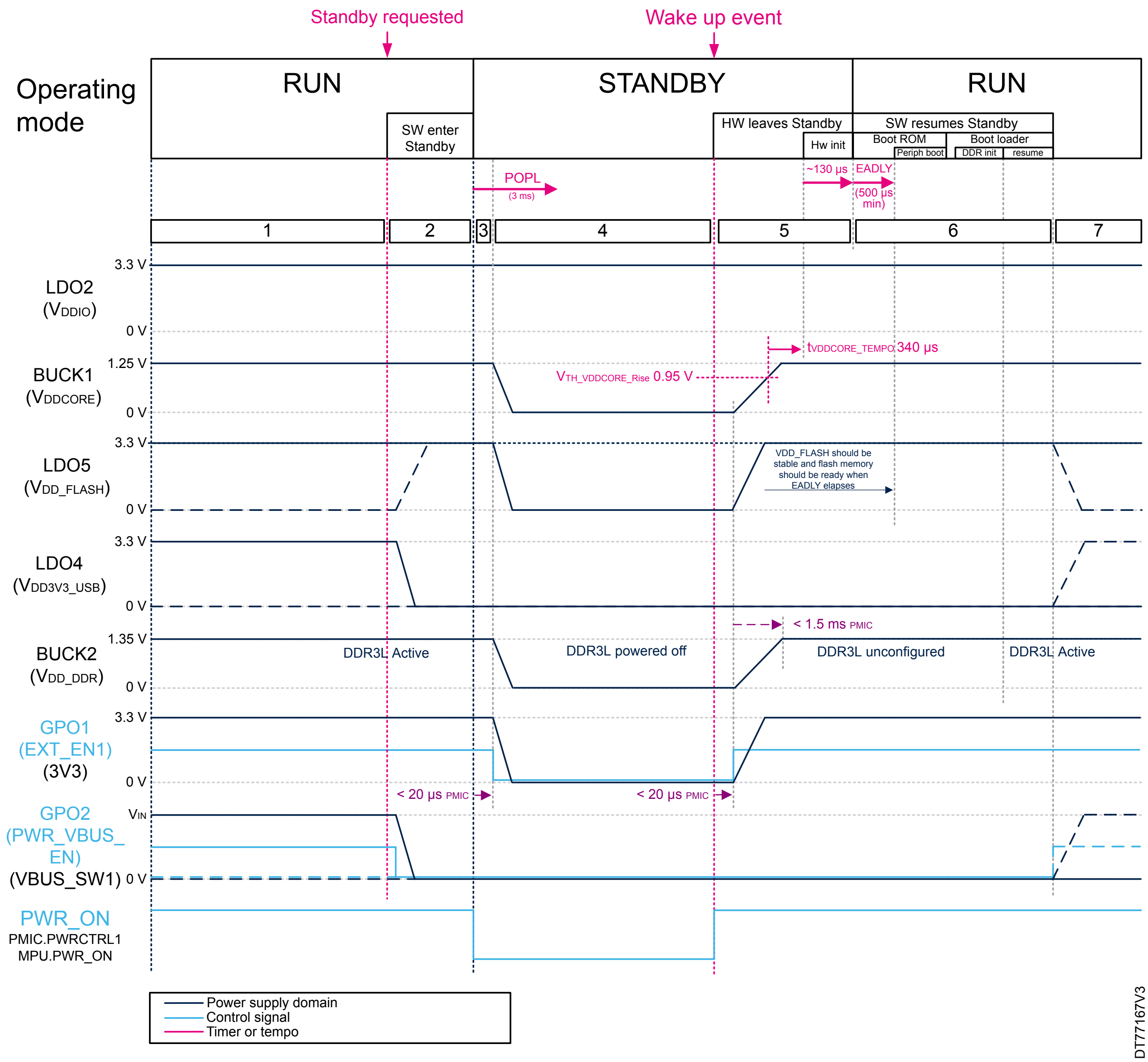
This section focuses on Standby mode with DDR3L off. This mode is described below and is shown in Figure 9 based on the implementation shown in Figure 1.

1. The application is powered up and operates in Run mode. PWR_ON is in high state.
2. When the Standby mode is requested, the software prepares to enter Standby mode:
   a. The MPU performs internal settings such as:
      i. Setting the POPL timer (see POPL timer), to define a minimum pulse duration of PWR_ON ensuring V_{DDCORE} voltage full discharge before restarting.
      ii. Setting the EADLY timer (see EADLY timer).
      iii. Stop non required system clocks.
      iv. Disabling DDR controller.
   b. The MPU performs the STPMIC1L as described in Table 13.
      i. If turned off, the V_{DD_FLASH} regulator is turned on to enable V_{DD_FLASH} when leaving Standby mode for peripheral boot.
      ii. If turned on, the V_{DD3V3_USB} regulator is turned off.
      iii. If turned on, the V_{BUS_SW1} power switch (controlled by PMIC GPO2) is turned off.
   c. The PWR_MPUCR.PDDS bit is enabled (Standby mode is allowed).
3. Once the system is in Standby mode:
   a. The MPU PWR_ON signal is deasserted (PMIC PWRCTRL1 signal goes low).
   b. The POPL timer is started to keep the application in Standby for a minimum POPL timer duration.
4. The PMIC regulators affected to PWR_ON take the configuration set in the xxxx_ALT_CR registers (see Table 13) after ~ 20 µs (internal PMIC delay):
   a. The V_{DDCORE} regulator is turned off.
   b. The V_{DD_FLASH} regulator is turned off.
   c. The V_{DD_DDR} regulator is turned off.
   d. The 3V3 regulator (controlled by PMIC GPO1) is turned off.
5. On a wake-up event, the MPU leaves the Standby mode (MPU waits for the POPL timer to elapse before leaving Standby mode):
   a. The MPU PWR_ON signal is asserted (PMIC PWRCTRL1 signal goes high).
   b. The PMIC regulators which are affected to the PWR_ON, take the configuration set in the xxxx_MAIN_CR registers (see Table 13) after ~ 20 µs (internal PMIC delay):
      i. The V_{DDCORE} regulator is turned on.
      ii. The V_{DD_FLASH} regulator is turned on.
      iii. The V_{DD_DDR} regulator is turned on.
      iv. The 3V3 regulator (controlled by PMIC GPO1) is turned on.
   c. Once V_{DDCORE} voltage reaches the $V_{TH\_VDDCORE}$ threshold, an MPU $t_{VDDCORE\_TEMPO}$ delay is started to wait for V_{DDCORE} voltage to reach the minimum operating voltage. Once $t_{VDDCORE\_TEMPO}$ elapses, the MPU performs internal hardware initialization.
6. Once the MPU ends internal hardware initialization, the MPU enters Run mode:
   a. The CPU starts to execute the boot ROM: EADLY timer is started (refer to EADLY timer for more information).
   b. Once EADLY elapses, the boot ROM reads, verifies, and executes the bootloader from the external flash memory (for example: the SD card as illustrated in Figure 1).
   c. The bootloader detects an "exit from Standby mode":
      i. It initializes DDR controller and DDR3L memory.
      ii. It resumes the application software from flash memory to DDR3L.
7. The system runs the application software:
   a. Peripherals switched off before entering Standby mode (such as USB) can be resumed.

Table 13. PMIC configuration Standby DDR off

Regulator	PWRCTRLx affectation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	V_{OUT}	Configuration	V_{OUT}
BUCK1 (V_{DDCORE})	PWR_ON	On	1.25	Off	-
BUCK2 (V_{DD_DDR})	PWR_ON	On	1.35	Off	-
LDO2 (V_{DDIO})	-	On	3.3	-	-
LDO3 (free)	-	-	-	-	-
LDO4 (V_{DD3V3_USB})	-	Off	N/A	-	N/A
LDO5 (V_{DD_FLASH})	PWR_ON	On	3.3	Off	-
GPO1 (3V3)	PWR_ON	On	N/A	Off	N/A
GPO2 (V_{BUS_SW1})	-	Off	N/A	-	N/A

Figure 9. **Standby (DDR off) sequence**



5.4 Low power mode management with STPMIC1LD and STM32MP13xD/F

5.4.1 LPLV-Stop mode

The LPLV-Stop1 mode is described below and is shown in Figure 10 based on the implementation shown in Figure 2:

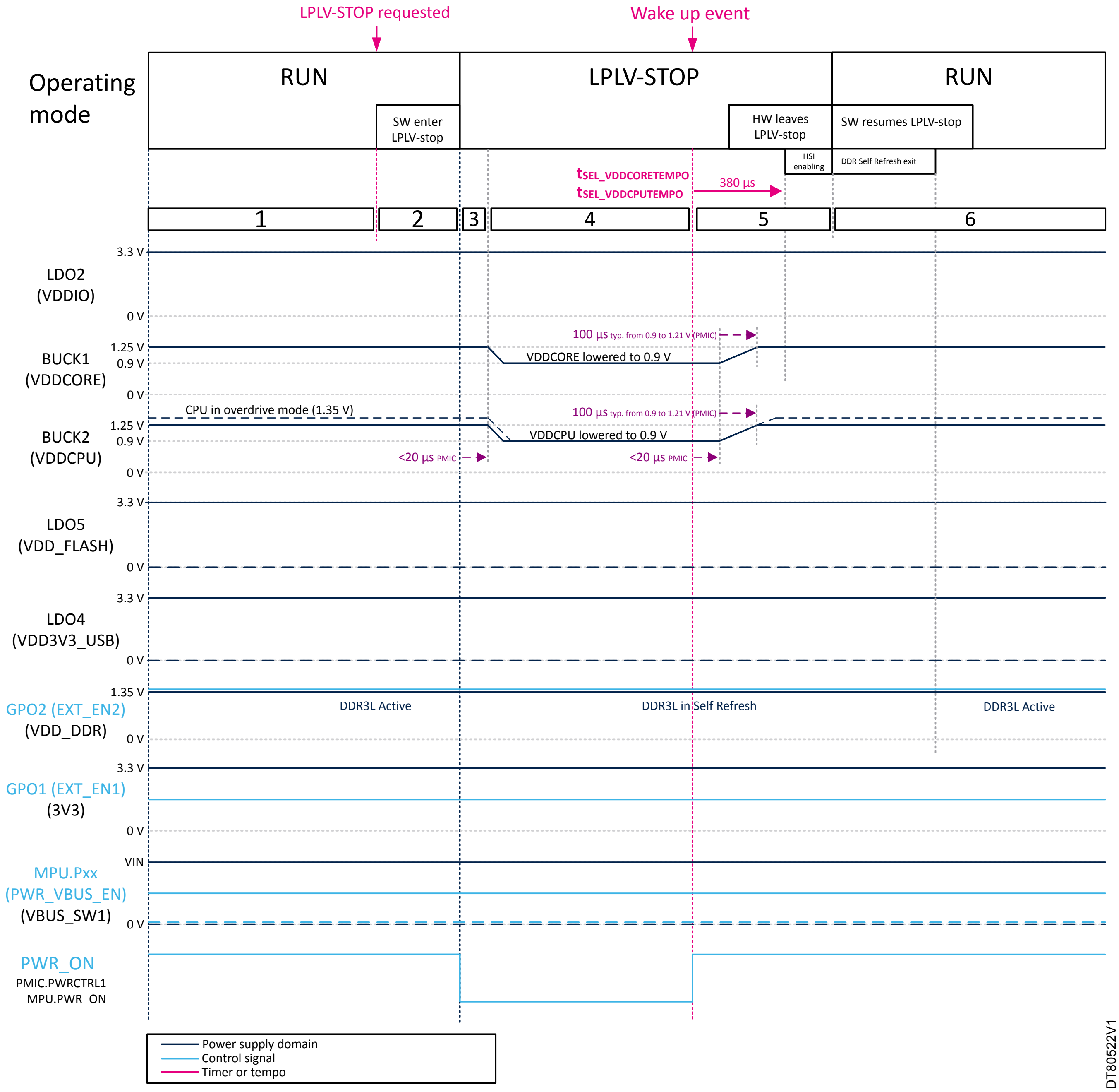
1. The application powers up and operates in Run mode; PWR_ON is in a high state. In this application, the PWR_LP signal is multiplexed with the PWR_ON pin (PWR_CR1.LPCFG bit is enabled).

2. When the LPLV-Stop mode is requested, the software prepares to enter LPLV-Stop mode:
   a. The MPU performs internal settings such as:
      i. Disabling PWRLP_TEMPO (set RCC_PWRLPDLYCR.PWRLP_DLY[21:0] = 0).
      ii. Stop non required system clocks.
      iii. Setting the DDR3L to self-refresh.
   b. The MPU performs the PMIC settings (see Table 14).
   c. The MPU enables the PWR_CR1.LPDS and PWR_CR1.LVDS bits.
3. Once the system enters LPLV-Stop mode:
   a. The MPU PWR_ON output is deasserted (PMIC PWRCTRL1 signal goes low).
4. The PMIC regulators affected by the PWR_ON take the configuration set in the xxxx_ALT_CR registers (see Table 14) after ~20 µs (internal PMIC delay):
   a. The V_{DDCORE} regulator output decreases to retention voltage (from 1.25 V to 0.9 V).
   b. The V_{DDCPU} regulator output decreases to retention voltage (from 1.25 V to 0.9 V).
5. On a wake-up event, the MPU exits LPLV-Stop mode:
   a. The MPU asserts the PWR_ON output signal (the PMIC PWRCTRL1 signal goes high) and starts the $t_{SEL\_VDDCORETEMPO}$ and $t_{SEL\_VDDCPUTEMPO}$ internal delays to wait for the V_{DDCORE} and V_{DDCPU} to switch from retention voltage (0.9 V) to minimum operating voltage (1.21 V).
   b. The PMIC regulators affected by the PWR_ON take the configuration set in the xxxx_MAIN_CR registers (see Table 14) after ~20 µs (internal PMIC delay):
      i. The V_{DDCORE} regulator switches from retention voltage (0.9 V) to minimal nominal operating voltage (1.21 V) in 100 µs typical and converges to nominal operating voltage (1.25 V). Accordingly, the V_{DDCORE} voltage is ready before the $t_{SEL\_VDDCORETEMPO}$ MPU delay elapses.
      ii. The V_{DDCPU} regulator switches from retention voltage (0.9 V) to minimal nominal operating voltage (1.21 V) in 100 µs typical and converges to nominal operating voltage (1.25 V). Accordingly, the V_{DDCPU} voltage is ready before the $t_{SEL\_VDDCPUTEMPO}$ MPU delay elapses.
   c. Once the $t_{SEL\_VDDCORETEMPO}$ internal delay elapses, clocks are enabled.
6. Once the clocks stabilize and the $t_{SEL\_VDDCPUTEMPO}$ internal delay elapses, the MPU immediately enters Run mode (as the PWRLP_TEMPO is bypassed) and the software resumes LPLV-Stop mode: DDR3L exits self-refresh.

Table 14. PMIC configuration for LPLV-Stop mode

Regulator	PWRCTRLx affectation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	V_{OUT}	Configuration	V_{OUT}
BUCK1 (V_{DDCORE})	PWR_ON	On	1.25	On	0.9
BUCK2 (V_{DDCPU})	PWR_ON	On	1.25/1.35	On	0.9
LDO2 (V_{DDIO})	-	On	3.3	-	-
LDO3 (free)	-	-	-	-	-
LDO4 (V_{DD3V3_USB})	-	On/off	N/A	-	N/A
LDO5 (V_{DD_FLASH})	PWR_ON	On/off	3.3	On/off	3.3
GPO1 (3V3)	PWR_ON	On	N/A	On	N/A
GPO2 (V_{DD_DDR})	PWR_ON	On	N/A	On	N/A

Figure 10. LPLV-Stop sequence



5.4.2 LPLV-Stop2 mode

The LPLV-Stop2 mode is described below and shown in Figure 11, based on the implementation in Figure 2.

1. The application powers up and operates in run mode. The PWR_ON signal is high. In this application, the PWR_LP signal is multiplexed with the PWR_ON pin. The PWR_CR1.LPCFG bit is enabled.

2. When LPLV-Stop2 mode is requested, the software prepares to enter LPLV-Stop2 mode:
   a. The MPU performs internal settings:
      i. Set PWRLP_TEMPO to 2 ms (set RCC_PWRLPDLYCR.PWRLP_DLY[21:0] = 0x20000).

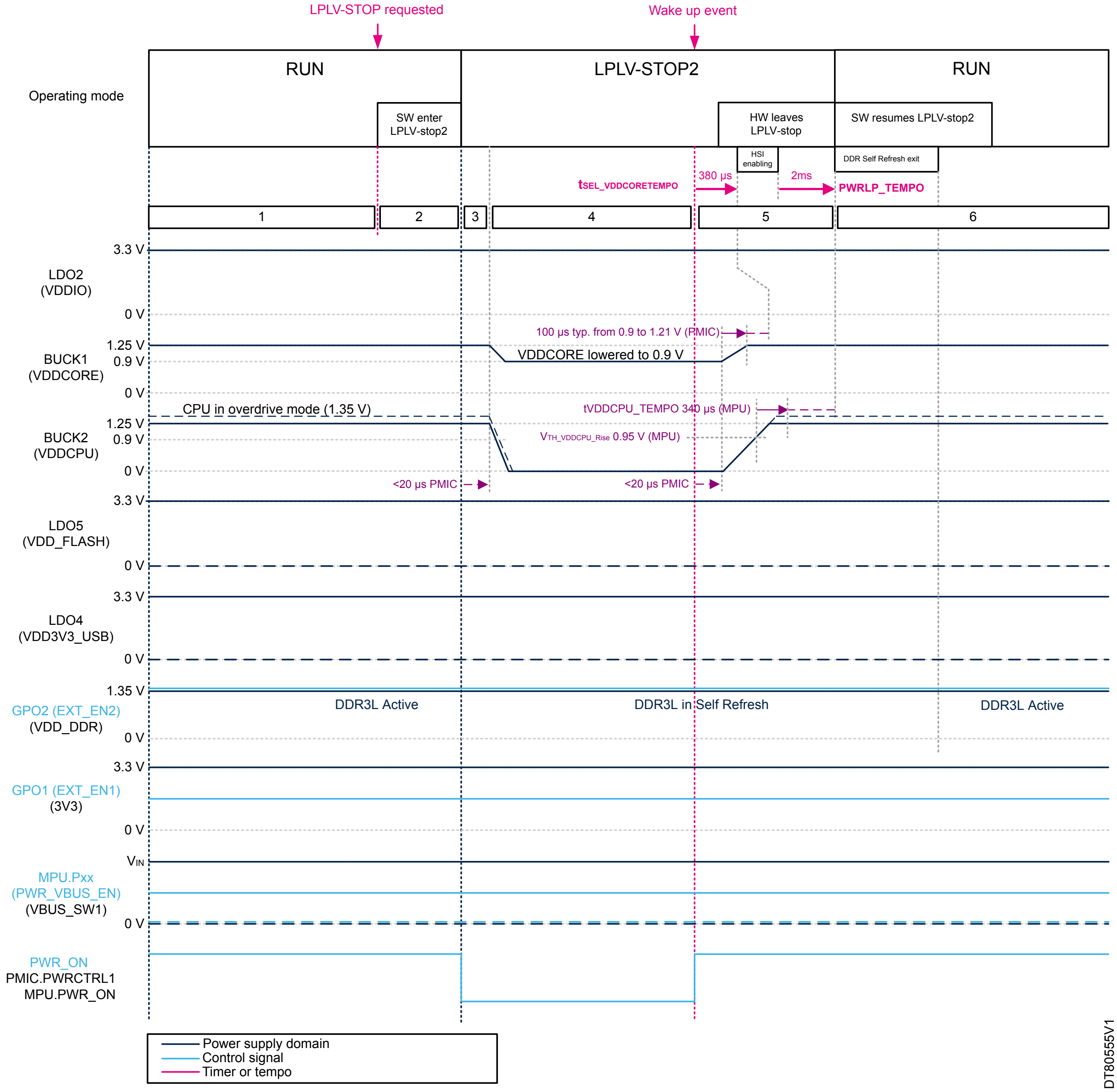
Note: If V_{DDCPU} remains above V_{TH_VDDCPU} during LPLV-Stop2 (very short LPLV-Stop2 duration), set PWRLP_TEMPO to 2 ms to allow enough time for BUCK2 to recover before run mode resumes.

      ii. Stop unnecessary system clocks.
      iii. Set DDR3L to self-refresh.
   b. The MPU performs PMIC settings (see Table 15).
   c. The MPU disables the PWR_MPUCR.PDDS bit and enables the PWR_CR1.STOP2 bit.
3. Once the system is in LPLV-Stop2 mode:
   a. The MPU deasserts the PWR_ON output (the PMIC PWRCTRL1 signal goes low).
4. The PMIC regulators affected to the PWR_ON take the configuration set in the xxxx_ALT_CR registers (see Table 15) after approximately 20 µs (internal PMIC delay):
   a. The V_{DDCORE} regulator output decreases to retention voltage (from 1.25 V to 0.9 V).
   b. The V_{DDCPU} regulator is turned off.
5. On a wake-up event, the MPU exits LPLV-Stop2 mode:
   a. The MPU asserts the PWR_ON output signal (the PMIC PWRCTRL1 signal goes high) and starts the $t_{SEL\_VDDCORETEMPO}$ internal delay to wait for V_{DDCORE} to switch from retention voltage (0.9 V) to minimum operating voltage (1.21 V).
   b. The PMIC regulators affected to the PWR_ON take the configuration set in the xxxx_MAIN_CR registers (see Table 15) after approximately 20 µs (internal PMIC delay):
      i. The V_{DDCORE} regulator switches from retention voltage (0.9 V) to minimum nominal operating voltage (1.21 V) in 100 µs typical and converges to nominal operating voltage (1.25 V). V_{DDCORE} voltage is ready before the $t_{SEL\_VDDCORETEMPO}$ MPU delay elapses.
      ii. The V_{DDCPU} regulator is turned on.
   c. When the $t_{SEL\_VDDCORETEMPO}$ internal delay elapses, clocks are enabled.
   d. When clocks are stable, the PWRLP_TEMPO timer is started.
   e. When V_{DDCPU} voltage is above the MPU V_{TH_VDDCPU} rising threshold (950 mV typical), the MPU starts the t_{VDDCPU_TEMPO} delay to wait for V_{DDCPU} voltage to reach minimum operating voltage.
6. When the PWRLP_TEMPO timer and the t_{VDDCPU_TEMPO} delay elapse, the MPU immediately enters Run mode and the software resumes LPLV-Stop2 mode. DDR3L exits self-refresh.

Table 15. PMIC configuration for LPLV-Stop2 mode

Regulator	PWRCTRLx affectation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	V_{OUT}	Configuration	V_{OUT}
BUCK1 (V_{DDCORE})	PWR_ON	On	1.25	On	0.9
BUCK2 (V_{DDCPU})	PWR_ON	On	1.25/1.35	Off	-
LDO2 (V_{DDIO})	-	On	3.3	-	-
LDO3 (free)	-	-	-	-	-
LDO4 (V_{DD3V3_USB})	-	On/off	N/A	-	N/A
LDO5 (V_{DD_FLASH})	PWR_ON	On/off	3.3	On/off	3.3
GPO1 (3V3)	PWR_ON	On	N/A	On	N/A
GPO2 (V_{DD_DDR})	PWR_ON	On	N/A	On	N/A

Figure 11. LPLV-Stop2 sequence



5.4.3 Standby mode (DDR3L in self-refresh)

The Standby mode is used when a very-low power consumption is required. Most of PMIC regulators are switched off. The content of MPU registers and memories are lost except for the backup domains (V_{DDIO} is kept enabled). The DDR3L is set in self-refresh (V_{DD_DDR} is kept enabled) to maintain the system in "suspend to RAM". The Standby mode with DDR3L in self-refresh is described in the Figure 12 based on the implementation shown in Figure 2.

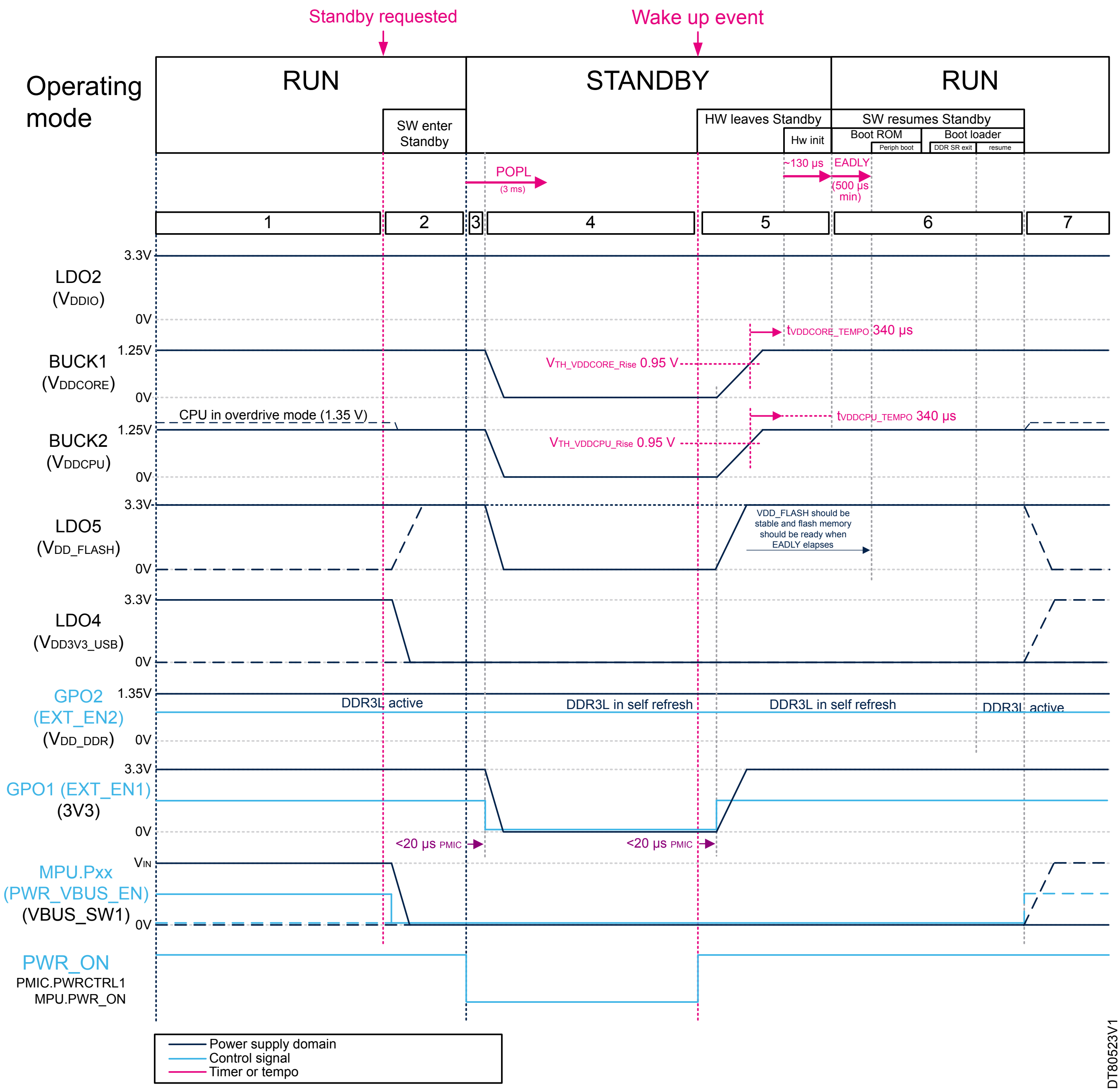
1. The application is powered up and operates in Run mode. PWR_ON is in a high state.

2. When the Standby mode is requested, the software prepares to enter Standby mode:
   a. The MPU performs internal settings such as:
      i. If CPU operates at overdrive frequency, set the CPU at nominal frequency then reduce the V_{DDCPU} voltage at nominal voltage.
      ii. Setting the POPL timer (see POPL timer), to define a minimum pulse duration of PWR_ON ensuring V_{DDCORE} voltage and V_{DDCPU} voltage full discharge before restarting.
      iii. Setting the EADLY timer (see EADLY timer).
      iv. Stop non required system clocks
      v. Setting DDR3L to self-refresh.
   b. The MPU configures the PMIC as described in Table 16:
      i. If turned off, the V_{DD_FLASH} regulator is turned on to enable V_{DD_FLASH} when leaving Standby mode for peripheral boot
      ii. If turned on, The V_{DD3V3_USB} regulator is turned off
      iii. If turned on, the V_{BUS_SW1} power switch (controlled by a MPU I/O) is turned off
   c. The PWR_MPUCR.PDDS bit is enabled (Standby mode is allowed)
3. Once the system is in Standby mode:
   a. The MPU PWR_ON signal is deasserted (PMIC PWRCTRL1 signal goes low)
   b. The POPL timer is started to keep application in Standby for a minimum POPL timer duration.
4. The PMIC regulators, which are affected to PWR_ON take the configuration set in the xxxx_ALT_CR registers (see Table 16) after ~20 µs (internal PMIC delay):
   a. The V_{DDCPU} regulator is turned off.
   b. The V_{DDCORE} regulator is turned off.
   c. The V_{DD_FLASH} regulator is turned off.
   d. The 3V3 regulator (controlled by PMIC GPO1) turns off.
5. On a wakeup event, the MPU leaves the Standby mode (MPU waits for the POPL timer to elapse before leaving Standby mode):
   a. The MPU PWR_ON signal is asserted (PMIC PWRCTRL1 signal goes high)
   b. The PMIC regulators affected to the PWR_ON take the configuration set in the xxxx_MAIN_CR registers (see Table 16) after ~20 µs (internal PMIC delay):
      i. The V_{DDCPU} regulator is turned on.
      ii. The V_{DDCORE} regulator is turned on.
      iii. The V_{DD_FLASH} regulator is turned on.
      iv. The 3V3 regulator (controlled by PMIC GPO1) is turned on.
   c. Once the V_{DDCORE} voltage and the V_{DDCPU} voltage reach the VTH_VDDCORE and the VTH_VDDCPU thresholds respectively, an MPU $t_{VDDCORE\_TEMPO}$ and t_{VDDCPU_TEMPO} delays are started to wait for V_{DDCORE} voltage and for V_{DDCPU} voltage to reach minimum operating voltage. Once $t_{VDDCORE\_TEMPO}$ elapses, the MPU performs internal hardware initialization.
6. Once the MPU ends internal hardware initialization and the t_{VDDCPU_TEMPO} elapses, the MPU enters in Run mode:
   a. The CPU starts to execute the boot ROM: EADLY timer is started (refer to EADLY timer for more information)
   b. Once EADLY elapses, the boot ROM reads, verifies, and executes the bootloader from the external flash memory (for example: the SD card as illustrated in Figure 2).
   c. The software detects an “exit from Standby mode”: It exits DDR from self-refresh and resumes the application software.
7. The system runs the application software:
   a. Peripherals switched off before entering standby mode (such as USB) can be resumed and the software can set the CPU at overdrive frequency.

Table 16. PMIC configuration for standby DDR in self refresh

Regulator	PWRCTRLx affectation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	V_{OUT}	Configuration	V_{OUT}
BUCK1 (V_{DDCORE})	PWR_ON	On	1.25	Off	-
BUCK2 (V_{DDCPU})	PWR_ON	On	1.25	Off	-
LDO2 (V_{DDIO})	-	On	3.3	-	-
LDO3 (free)	-	-	-	-	-
LDO4 (V_{DD3V3_USB})	-	Off	N/A	-	N/A
LDO5 (V_{DD_FLASH})	PWR_ON	On	3.3	Off	-
GPO1 (3V3)	PWR_ON	On	N/A	Off	N/A
GPO2 (V_{DD_DDR})	PWR_ON	On	N/A	On	N/A

Figure 12. Standby (DDR in self-refresh) sequence



5.4.4 Standby mode (DDR3L off)

The standby mode is used when very-low power consumption is required. Most of the PMIC regulators are switched off. The content of MPU registers and memories is lost except for the backup domain (V_{DDIO} remains enabled). The DDR3L is powered off (V_{DD_DDR} is disabled), so the system is in "suspend to flash."

This section focuses on standby mode with DDR3L off. This mode is described below and is shown in Figure 13 based on the implementation shown in Figure 2.

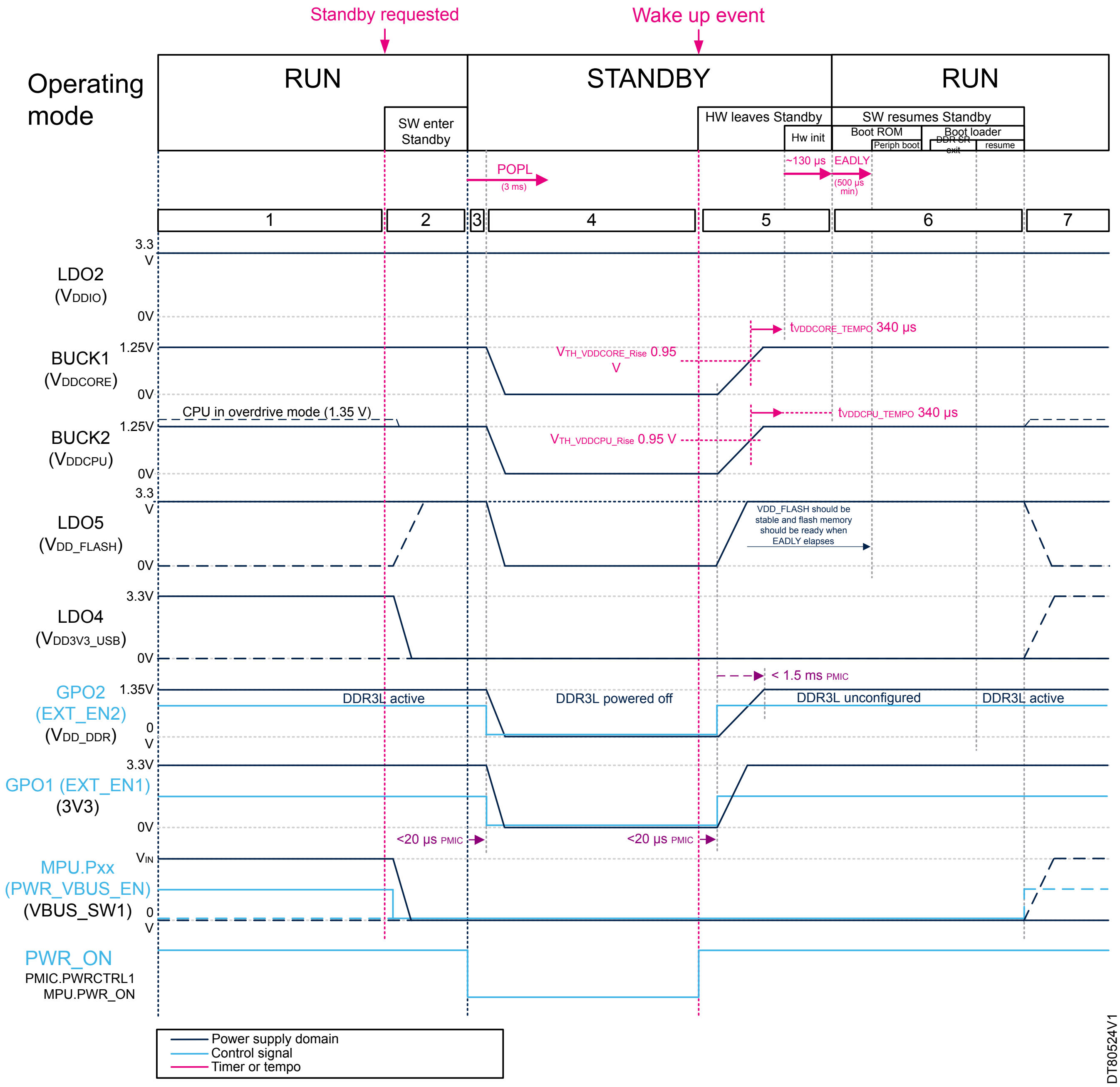
1. The application is powered up and works in run mode; PWR_ON is in a high state.

2. When the standby mode is requested, the software prepares to enter standby mode:
   a. The MPU performs internal settings such as:
      i. If the CPU operates at overdrive frequency, set the CPU at nominal frequency, then reduce the V_{DDCPU} voltage to nominal voltage.
      ii. Set the POPL timer (see POPL timer) to define a minimum pulse duration of PWR_ON, ensuring V_{DDCORE} voltage and V_{DDCPU} voltage fully discharge before restarting.
      iii. Set the EADLY timer (see EADLY timer).
      iv. Stop non required system clocks.
      v. Disable the DDR controller.
   b. The MPU configures the STPMIC1L as described in Table 17:
      i. If turned off, the V_{DD_FLASH} regulator is turned on to enable V_{DD_FLASH} when leaving Standby mode for peripheral boot.
      ii. If turned on, the V_{DD3V3_USB} regulator is turned off.
      iii. If turned on, the V_{BUS_SW1} power switch (controlled by an MPU I/O) is turned off.
   c. Enable the PWR_MPUCR.PDDS bit (standby mode is allowed).
3. Once the system is in standby mode:
   a. The MPU PWR_ON signal is deasserted (PMIC PWRCTRL1 signal goes low).
   b. The POPL timer is started to keep the application in standby for a minimum POPL timer duration.
4. The PMIC regulators affected by PWR_ON take the configuration set in the xxxx_ALT_CR registers (see Table 17) after ~20 µs (internal PMIC delay):
   a. The V_{DDCPU} regulator is turned off.
   b. The V_{DDCORE} regulator is turned off.
   c. The V_{DD_FLASH} regulator is turned off.
   d. The V_{DD_DDR} regulator (controlled by PMIC GPO2) is turned off.
   e. The 3V3 regulator (controlled by PMIC GPO1) is turned off.
5. On a wakeup event, the MPU leaves the standby mode (MPU waits for the POPL timer to elapse before leaving standby mode):
   a. The MPU PWR_ON signal is asserted (PMIC PWRCTRL1 signal goes high).
   b. The PMIC regulators affected by PWR_ON take the configuration set in the xxxx_MAIN_CR registers (see Table 17) after ~20 µs (internal PMIC delay):
      i. The V_{DDCPU} regulator is turned on.
      ii. The V_{DDCORE} regulator is turned on.
      iii. The V_{DD_FLASH} regulator is turned on.
      iv. The V_{DD_DDR} regulator (controlled by PMIC GPO2) is turned on.
      v. The 3V3 regulator (controlled by PMIC GPO1) is turned on.
   c. Once the V_{DDCORE} voltage and the V_{DDCPU} voltage reach the VTH_VDDCORE and VTH_VDDCPU thresholds respectively, the MPU starts $t_{VDDCORE\_TEMPO}$ and t_{VDDCPU_TEMPO} delays to wait for V_{DDCORE} voltage and V_{DDCPU} voltage to reach minimum operating voltage. Once $t_{VDDCORE\_TEMPO}$ elapses, the MPU performs internal hardware initialization.
6. Once the MPU completes internal hardware initialization and t_{VDDCPU_TEMPO} elapses, the MPU enters run mode:
   a. The CPU starts to execute the boot ROM; the EADLY timer is started (refer to EADLY timer for more information).
   b. Once EADLY elapses, the boot ROM reads, verifies, and executes the bootloader from the external flash memory (for example, the SD card as illustrated in Figure 2).
   c. The bootloader detects an "exit from standby mode":
      i. It initializes the DDR controller and DDR3L memory.
      ii. It resumes the application software from flash memory to DDR3L.
7. The system runs the application software:
   a. Peripherals switched off before entering standby mode (such as USB) can be resumed, and the software can set the CPU to overdrive frequency.

Table 17. PMIC configuration Standby DDR off

Regulator	PWRCTRLx affectation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	V_{OUT}	Configuration	V_{OUT}
BUCK1 (V_{DDCORE})	PWR_ON	On	1.25	Off	-
BUCK2 (V_{DDCPU})	PWR_ON	On	1.25	Off	-
LDO2 (V_{DDIO})	-	On	3.3	-	-
LDO3 (free)	-	-	-	-	-
LDO4 (V_{DD3V3_USB})	-	Off	N/A	-	N/A
LDO5 (V_{DD_FLASH})	PWR_ON	On	3.3	Off	-
GPO1 (3V3)	PWR_ON	On	N/A	Off	N/A
GPO2 (V_{DD_DDR})	PWR_ON	On	N/A	Off	N/A

Figure 13. Standby (DDR off) sequence



5.5 Low power mode management with STPMIC1LD and STM32MP13xD/F with V_{DDCPU} merged with V_{DD_DDR}

5.5.1 Standby mode (DDR3L off)

The Standby mode is used when very-low power consumption is required. Most of the PMIC regulators are switched off. The content of MPU registers and memories is lost except for the backup domain (V_{DDIO} remains enabled). The DDR3L is powered off (V_{DDCPU_DDR} is disabled), so the system is in "suspend to flash."

This section focuses on Standby mode with DDR3L off. This mode is described below and illustrated in Figure 14, based on the implementation shown in Figure 3.

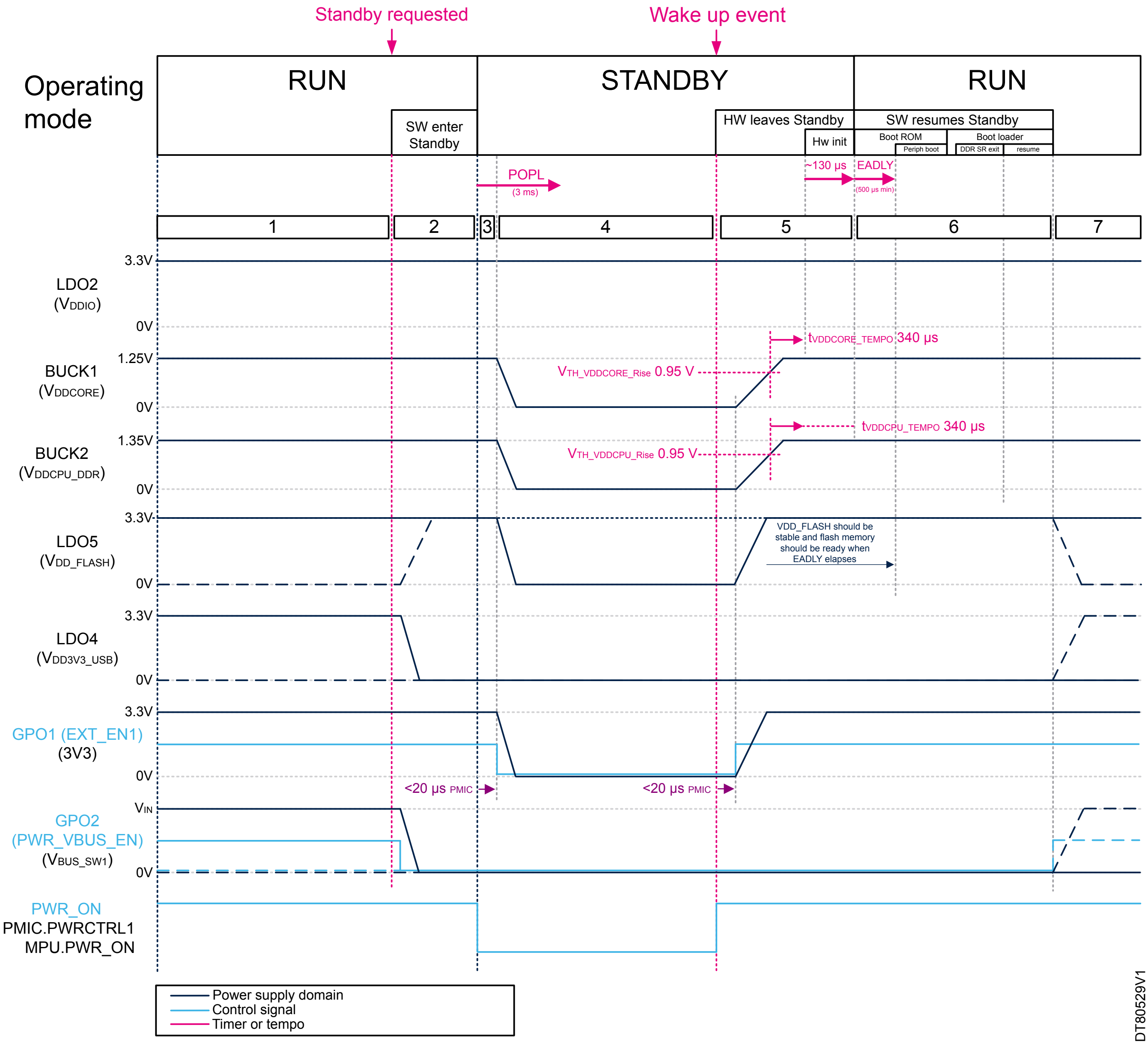
1. The application is powered up and operates in Run mode; PWR_ON is in a high state.

2. When Standby mode is requested, the software prepares to enter Standby mode:
   a. The MPU performs internal settings, such as:
      i. Configuring the POPL timer (refer to POPL timer) to define a minimum pulse duration of PWR_ON, ensuring full discharge of V_{DDCORE} and V_{DDCPU} voltages before restarting.
      ii. Configuring the EADLY timer (refer to EADLY timer).
      iii. Stop non required system clocks.
      iv. Disabling the DDR controller.
   b. The MPU configures the STPMIC1L as described in Table 18:
      i. If turned off, the V_{DD_FLASH} regulator is turned on to enable V_{DD_FLASH} when leaving Standby mode for peripheral boot.
      ii. If turned on, the V_{DD3V3_USB} regulator is turned off.
      iii. If turned on, the V_{BUS_SW1} power switch (controlled by PMIC GPO2) is turned off.
   c. The PWR_MPUCR.PDDS bit is enabled (Standby mode is allowed).
3. Once the system enters Standby mode:
   a. The MPU PWR_ON signal is deasserted (PMIC PWRCTRL1 signal goes low).
   b. The POPL timer starts, keeping the application in Standby for the minimum POPL timer duration.
4. The PMIC regulators affected by PWR_ON adopt the configuration set in the xxxx_ALT_CR registers (refer to Table 18) after approximately 20 µs (internal PMIC delay):
   a. The V_{DDCPU_DDR} regulator is turned off.
   b. The V_{DDCORE} regulator is turned off.
   c. The V_{DD_FLASH} regulator is turned off.
   d. The 3V3 regulator (controlled by PMIC GPO1) is turned off.
5. On a wakeup event, the MPU exits Standby mode (the MPU waits for the POPL timer to elapse before exiting Standby mode):
   a. The MPU PWR_ON signal is asserted (PMIC PWRCTRL1 signal goes high).
   b. The PMIC regulators affected by PWR_ON adopt the configuration set in the xxxx_MAIN_CR registers (refer to Table 18) after approximately 20 µs (internal PMIC delay):
      i. The V_{DDCPU_DDR} regulator is turned on.
      ii. The V_{DDCORE} regulator is turned on.
      iii. The V_{DD_FLASH} regulator is turned on.
      iv. The 3V3 regulator (controlled by PMIC GPO1) is turned on.
   c. Once the V_{DDCORE} voltage and V_{DDCPU_DDR} voltage reach the VTH_VDDCORE and V_{TH_VDDCPU} thresholds, respectively, the MPU initiates $t_{VDDCORE\_TEMPO}$ and t_{VDDCPU_TEMPO} delays to allow the voltages to reach their minimum operating levels. After $t_{VDDCORE\_TEMPO}$ elapses, the MPU performs internal hardware initialization.
6. After the MPU completes internal hardware initialization and t_{VDDCPU_TEMPO} elapses, the MPU enters Run mode:
   a. The CPU begins executing the boot ROM; the EADLY timer starts (refer to EADLY timer for more information).
   b. Once EADLY elapses, the boot ROM reads, verifies, and executes the bootloader from external flash memory (for example: the SD card, as illustrated in Figure 3).
   c. The bootloader detects an "exit from Standby mode":
      i. It initializes the DDR controller and DDR3L memory.
      ii. It resumes the application software from flash memory to DDR3L.
7. The system runs the application software:
   a. Peripherals switched off before entering Standby mode (for example: USB) can be resumed.

Table 18. **PMIC configuration Stanby DDR off**

Regulator	PWRCTRLx affectation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	V_{OUT}	Configuration	V_{OUT}
BUCK1 (V_{DDCORE})	PWR_ON	On	1.25	Off	-
BUCK2 (V_{DDCPU_DDR})	PWR_ON	On	1.35	Off	-
LDO2 (V_{DDIO})	-	On	3.3	-	-
LDO3 (free)	-	-	-	-	-
LDO4 (V_{DD3V3_USB})	-	Off	N/A	-	N/A
LDO5 (V_{DD_FLASH})	PWR_ON	On	3.3	Off	-
GPO1 (3V3)	PWR_ON	On	N/A	Off	N/A
GPO2 (V_{BUS_SW1})	-	Off	N/A	-	N/A

Figure 14. Standby (DDR off) sequence



5.6 Crash recovery/user reset/hard fault management

An MPU crash, a user reset, or a PMIC hard fault are managed similarly by a power cycling sequence handled by PMIC. Accordingly, only application illustrated in Figure 1 is described in this section, considering that application illustrated in Figure 2 and in Figure 3 with the STPMIC1LD have similar behavior

An MPU crash (iwdg1_out_rst, iwdg2_out_rst) or MPU system reset or a user reset (push button) generates a reset pulse on the NRST signal to PMIC. As introduced in RSTn pin, the MPU, and the PMIC both have interconnected bidirectional reset pins (see Figure 1 signal NRST). The reset pulse triggers the PMIC to produce an immediate power cycle sequence. This power cycling ensures a correct reset and restart of the peripherals following a global application reset (NRST).

A default STPMIC1LA hard fault (see Table 10 and Section 5.2.4) is managed by a power cycling; like a system reset. This behavior programmed into the STPMIC1LA NVM may be adjusted depending on expected safety management behavior (see Section 6.1).

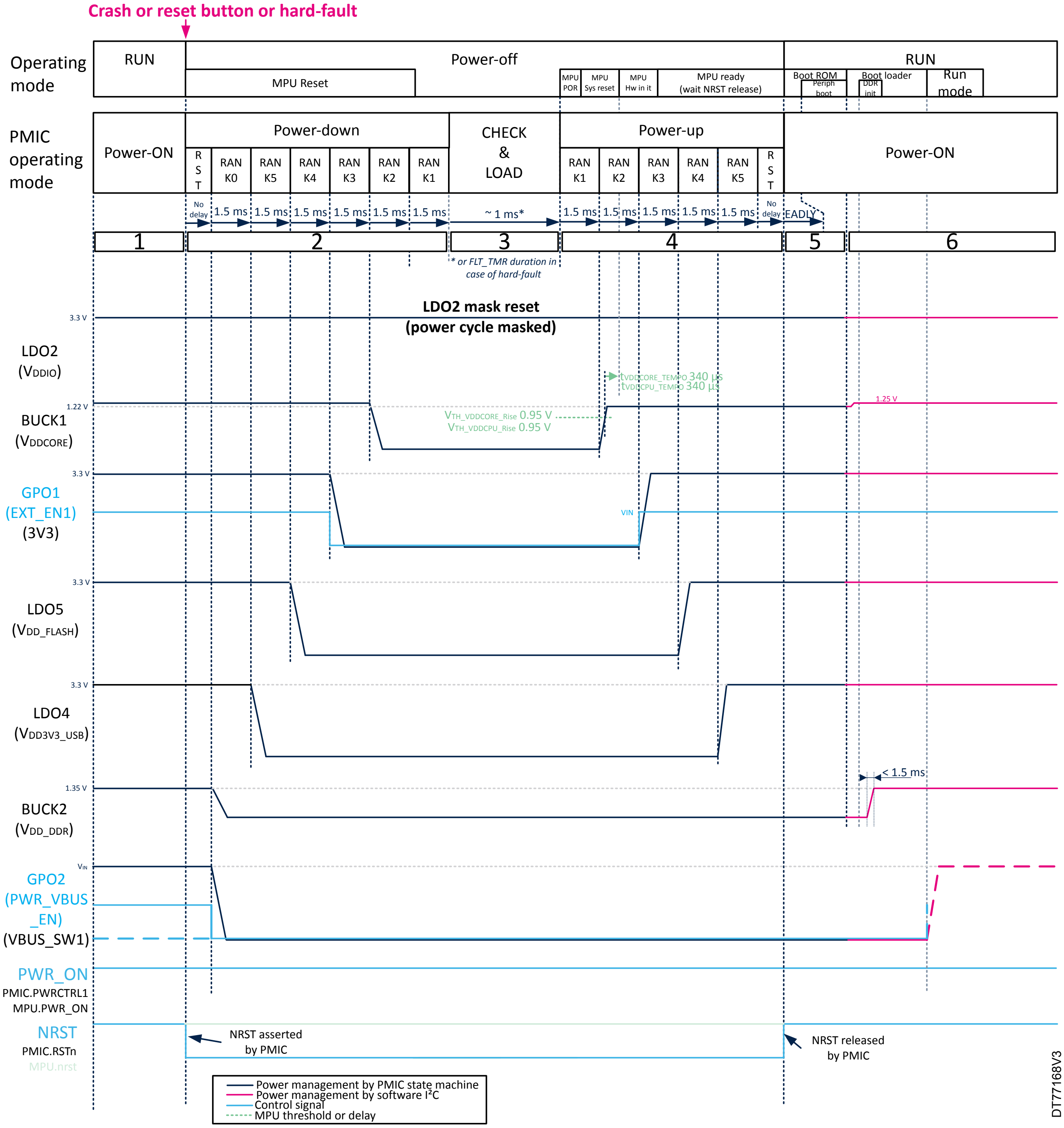
The sequence in Figure 15 illustrates an MPU crash recovery or user push-button reset or PMIC hard fault behavior according to the application shown in Figure 1. In this application, the mask-reset (see Section 5.1.4) is enable for the V_{DDIO} (PMIC LDO2) allowing to keep the LDO2 enabled during the power cycling sequence. Implicitly, the mask-reset on LDO2 has been set by software prior to the sequence; typically by the bootloader during application initialization.

1. The application is powered up and operates in Run mode. An MPU crash occurs or a user presses the reset button generating a pulse on the NRST signal, or the PMIC detects a hard fault.
2. Once the NRST signal is asserted or a hard fault is detected by the PMIC, the PMIC performs a power-down sequence:
   a. The PMIC asserts the RSTn, asserting the MPU NRST signal.
   b. RANK0 (1.5 ms): the BUCK2 (V_{DD_DDR}) regulator is turned off and GPO2 is turned off.
   c. RANK5 (1.5 ms): the LDO4 (V_{DD3V3_USB}) regulator is turned off.
   d. RANK4 (1.5 ms): the LDO5 (V_{DD_FLASH}) regulator is turned off.
   e. RANK3 (1.5 ms): GPO1 (EXT_EN1) is turned off. The 3V3 discrete SMPS regulator is disabled.
   f. RANK2 (1.5 ms): BUCK1 (V_{DDCORE}) regulator is turned off.
   g. RANK1 (1.5 ms): the LDO2 (V_{DDIO}) is kept turned on as mask-reset set on LDO2.
3. Once the power-down sequence ends, the PMIC goes in CHECK&LOAD state (see [1] for more details) to prepare for the power-up sequence.

Note: STPMIC1LA always restarts after a hard fault (default behavior).

4. Once the CHECK&LOAD ends (after ~ 7 ms for an NRST assertion or after FLT_TMR duration for a hard fault (see [1]) PMIC performs a power-up sequence:
   a. RANK1 (1.5 ms): the LDO2 (V_{DDIO}) is already turned on (mask-reset).
   b. RANK2 (1.5 ms): the BUCK1 (V_{DDCORE}) regulator is turned on at 1.22 V. Once V_{DDCORE} voltage is above MPU $V_{TH\_VDDCORE}$ rising threshold (950 mV typ.), an MPU $t_{VDDCORE\_TEMPO}$ delay is started. This is done to wait for V_{DDCORE} voltage to reach the minimum operating voltage. Once $t_{VDDCORE\_TEMPO}$ elapses, the MPU starts and the HSI oscillators then perform internal hardware initialization. The MPU then waits for NRST to release.
   c. RANK3 (1.5 ms): the GPO1 (EXT_EN1) is turned on. The 3V3 discrete SMPS regulator is enabled and the 3V3 voltage rises.
   d. RANK4 (1.5 ms): LDO5 (V_{DD_FLASH}) regulator is turned on at 3.3 V.
   e. RANK5 (1.5 ms): LDO4 (V_{DD3V3_USB}) regulator is turned on.
   f. Once RANK5 ends, the PMIC releases the RSTn that releases MPU NRST.
5. Once NRST is released, the MPU enters Run mode:
   a. The CPU starts to execute the boot ROM: EADLY timer is started (refer to EADLY timer for more information).
   b. Once EADLY elapses, the boot ROM reads, verifies, and executes the bootloader from the external flash memory (for example: the SD card as illustrated in Figure 1).
6. The bootloader software performs initializations then loads and executes the application software:
   a. Set PMIC BUCK1 (V_{DDCORE}) at 1.25 V (V_{DDCORE} and V_{DDCPU} nominal voltage).
   b. Enable DDR regulators: turn on BUCK2 (V_{DD_DDR}) at 1.35 V then set a 1.5 ms timer to wait for DDR voltages stabilization.
   c. Once the 1.5 ms timer elapses, the bootloader software initializes the DDR controller and DDR memory IC.
   d. The bootloader loads the application software into DDR3L and executes it.
   e. The system runs.

Figure 15. Crash recovery sequence



6 Safety management

In this documentation, *safety management* is the concept of implementing mechanisms such as OCP (Over Current Protection), or watchdogs 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 (fail-safe management).

See [1] 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 counter 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 in FAIL_SAFE_LOCK state. By default, there is no limit applied to any of the hard-fault counter on the STPMIC1LA or the STPMIC1LD. It means that STPMIC1LA or STPMIC1LD always restart 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. A power-up sequence as defined in Section Section 5.6: Crash recovery/user reset/hard fault management.

Once the number of hard fault iterations exceeds 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: STPMIC1LA behavior with a negative voltage glitch on VIN

Initial condition: the STPMIC1LA NVM has the default value:

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

Description:

A 5 V wall adaptor is plugged to the application main supply connector. The VIN rises above V_{INOK_rise} (4 V), the PMIC then powers up the application and the application initializes. A negative voltage glitch occurs on VIN (for example: due to bad contact at main supply connector). This causes a glitch voltage to fall below V_{INOK_fall} (3.5 V). The PMIC causes a VIN 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 (VIN) below V_{INOK_fall}, the STPMIC1LA PMIC always restarts as VIN_FLT_CNT ≤ VIN_FLT_CNT_MAX is always true.

This behavior is identical for other hard fault sources: the STPMIC1LA always restarts.

Example 2 (PMIC behavior with negative voltage glitch on VIN and tuned fail-safe management in PMIC NVM)

The initial condition is that the STPMIC1LA NVM has been tuned to adjust fail-safe management as follows:

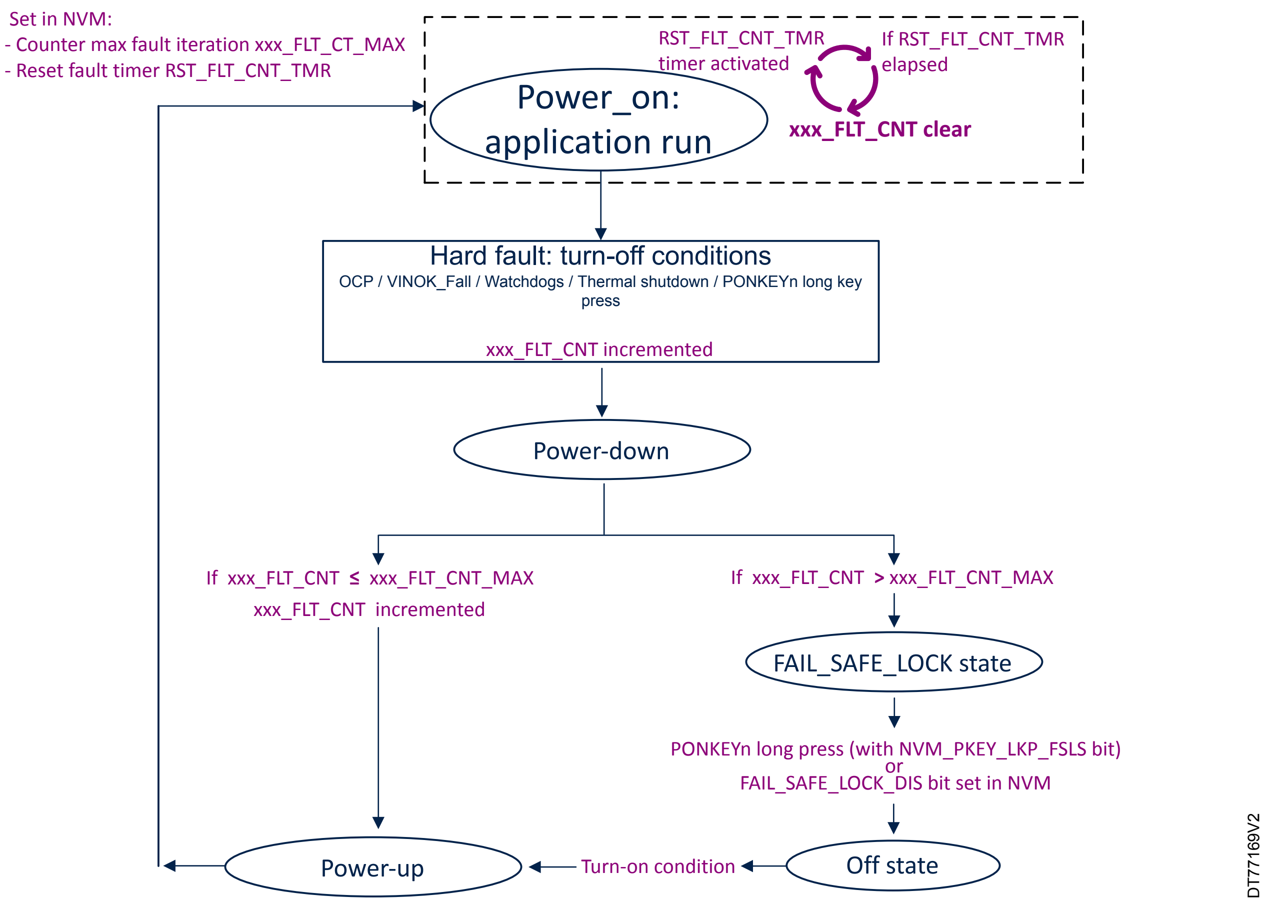
- VIN_FLT_CNT_MAX[3:0] = 0x0001 programmed in PMIC NVM_FS_SHR1 NVM. This specifies that one hard-fault on VIN is allowed.
- RST_FLT_CNT_TMR[1:0] = 0x10 programmed in PMIC NVM_FS_SHR2 NVM. This specifies that all fault counters are cleared if no hard fault is detected for six minutes.

Description:

A 5V wall adaptor 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 VIN due to bad contact at main supply connector. Tgis causes a glitch voltage to fall below V_{INOK_fall} (3.5 V). The PMIC causes a VIN 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 ends, 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, 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 (so 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 16. PMIC fail-safe management mechanism



6.2 PMIC hard faults

PMIC has five hardware source of events considered as hard fault conditions:

- OCP: overcurrent protection, including short circuit
- VIN: undervoltage protection (VIN fall below V_{INOK_fall} threshold)
- TSHDN: thermal shutdown protection
- WDG: watchdog timer expiration
- PKEY: power-on key button long press

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

- It triggers a turn-off condition (see Turn-off conditions) followed by a 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 STPMIC1LA and the STPMIC1LD.
  – Not restart automatically: PMIC is kept in FAIL_SAFE_LOCK or off state until an allowed turn-on condition is met.

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

6.2.1 OCP overcurrent protection

All PMIC regulators implement two levels 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 another power domain of the application. This level of protection is suitable for the non-critical power domain.

Once an OCP occurs, the regulator turns off for t_{HICCUP_DLY} timer (predefined in NVM), 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. Implicitly, the PMIC enters power-off, and then power-on (depending on fail-safe settings). This level of protection is suitable for critical power domain (such as MPU V_{DDCORE}, DDR3L regulators) where it is mandatory to restart the application completely in case of overcurrent or short-circuit.

The Section 6.3 illustrates PMIC OCP settings according to 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 MPU or peripherals from crashing in case PMIC regulators go out of regulation due to too low V_{IN} input voltage.

Once the main V_{IN} supply goes below V_{INOK_Fall} thresholds, even for a very short duration (such as a voltage glitch), a hard fault condition is generated: the VIN fail-safe counter (VIN_FLT_CNT) is incremented and the PMIC powers down.

If the VIN_FLT_CNT counter does not exceed the VIN_FLT_CNT_MAX when power-down ends, the PMIC enters in FAIL_SAFE_LOCK state. Otherwise, the PMIC waits for FLT_TMR fault timer (t_{VINOK_Fall} = 100 ms), then 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 for 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 goes higher than 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 does not exceed the TSHDN _FLT_CNT_MAX when power-down ends, the PMIC wait for:
  – FLT_TMR fault timer, where tSHDN_DLY = 3 s
  – PMIC junction temperature goes below TSHDN_Fall threshold
  The PMIC powers 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 stop 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), 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 STPMIC1LA and STPMIC1LD. PMIC 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 (infinite by default) 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 settings in the application

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

All regulators critical for the application are managed in OCP fail-safe (level 1), else in OCP Hiccup (level 0).

These settings can be modified by reprogramming the NVM_FS_OCP_SHRx registers in PMIC NVM according to Table 19 for STPMIC1LA and Table 20 for STPMIC1LD.

Table 19. OCP management application with STPMIC1LA

-	HICCUP (level 0)	Fail-safe (level 1)
BUCK1 (V_{DDCORE})	-	Yes
BUCK2 (V_{DD_DDR})	-	Yes
LDO2 (V_{DDIO})	-	Yes
LDO3 (free)	Yes	-
LDO4 (V_{DD3V3_USB})	Yes	-
LDO5 (V_{DD_FLASH})	Yes	-

Table 20. OCP management application with STPMIC1LD

-	HICCUP (level 0)	Fail Safe (level 1)
BUCK1 (V_{DDCORE})	-	Yes
BUCK2 (V_{DDCPU})	-	Yes
LDO2 (V_{DDIO})	-	Yes
LDO3 (free)	Yes	-
LDO4 (V_{DD3V3_USB})	Yes	-
LDO5 (V_{DD_FLASH})	Yes	-

7 PMIC tuning (optional)

The PMIC NVM can be reprogrammed by customers to adjust regulator settings and safety management behavior to align with MPU-based application requirements. This process can be performed using ST tools, such as STM32CubeProgrammer.

Revision history

Table 21. Document revision history

Date	Version	Changes
21-Jul-2025	1	Initial release.
13-Nov-2025	2	Removed VTT_DDR DDR3L termination resistor network in the document. Updated Section 2: Overview. Updated Figure 1. Updated introduction to Section 4: 5 V power supply application reference designs. Updated Section 4.4: Power distribution. Updated Table 4. Default STPMIC1LA NVM configuration. Uodated Section 4.5.4: PMIC digital control interface. Updated Table 6. Operating modes. Updated Section 5.2.1: STPMIC1LA and STM32MP13xA/C power-up triggered by main supply (VIN) plugin and power-down by software shutdown. Updated Figure 4. Power-up and power-down sequence STM32MP13xA/C with STPMIC1LA. Updated Section 5.3: Low-power mode management with STPMIC1LA and STM32MP13xA/C. Updated Table Low-power mode supported by the application. Updated Table 11. PMIC configuration for LPLV-Stop mode. Updated Figure 7. LPLV-Stop sequence. Updated Table 12. PMIC configuration for Standby DDR in self-refresh. Updated Figure 8. Standby (DDR in self-refresh) sequence. Updated Table 13. PMIC configuration Standby DDR off. Updated Figure 9. Standby (DDR off) sequence. Updated Figure 15. Crash recovery sequence. Updated Table 19. OCP management application with STPMIC1LA.
09-Apr-2026	3	Whole document edited except: • Section 1: General information • Section 4.4.3: V_{DDIO} power domain (3.3 V) • Section 4.4.4: V_{DD3V3_USB} power domain • Section 4.4.6: V_{DD_FLASH} power domain (3.3 V) • Section 5.3.3: Standby mode (DDR3L off) • Section 6: Safety management • Section 6.2.1: OCP overcurrent protection • Section 6.2.2: V_{IN} undervoltage protection ($V_{IN} < V_{INOK\_Fall}$) • Section 6.2.3: TSHDN: thermal shutdown protection • Section 6.2.4: WDG: watchdog timer expiration

Contents

List of tables

List of figures