AN5727

Application note

How to use STPMIC25 for a wall adapter powered application on STM32MP25x lines MPUs

Introduction

This application note applies to the STM32MP25x MPU devices as detailed in the table below. The devices are referred to as STM32MP25x in the rest of the document. It is powered by the STPMIC25 power management IC companion chip, which is fully featured to supply complete applications.

This document provides an example of a hardware reference design based on a STM32MP25x device. The STM32MP25x is powered by an external 5 V power supply via the STPMIC25APQR power management IC. The STPMIC25APQR peripheral I/O voltage runs at 3.3 V.

This document is intended for product architects and designers who require information about the power management and STPMIC25 settings. This document focuses on:

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

Table 1. Applicable products

Reference	Applicable products
STM32MP25x	STM32MPD251, STM32MP253, STM32MP255, STM32MP257

1 General information

This document applies to STM32MP25x Arm®-based MPUs and STPMIC25x power management IC
Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

2 Overview

This application note describes the interaction between the STM32MP25x and the STPMIC25APQR including the management of the following peripherals:

- DC input power source from main power supply: 5 V typical (4.1 V to 5.5 V).
- DDR4 memory.
- Peripheral I/O interface voltage (V_{DD}) at 3.3 V powered by the STPMIC25.
- USB 3.0 superspeed port supporting power delivery to supply a USB device.
- eMMC flash memory (HS200) as boot device.
- SD card (UHS-I) which is not used as a boot device in this reference design.

Not covered in this application note:

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

The battery-powered application using the STPMIC25 is addressed in [4].

In this document, MPU terminology refers to the STM32MP25x.

2.1 Reference documents

Table 2. **Reference documents**

-	Reference	Title
STMicroelectronics document(1)		
[1]	AN5489	Getting started with STM32MP25x MPUs hardware development
[2]	DS14278	Highly integrated power management IC for microprocessor unit
[3]	RM0457	STM32MP25x advanced Arm®-based 32/64-bit MPUs
[4]	AN5728	How to use STPMIC2x for a battery powered application on STM32MP25 MPUs
[5]	AN5726	Guidline for using low power modes on STM32MP2 MPUs

1. *Refer to www.st.com*

3 Glossary

Table 3. Glossary

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

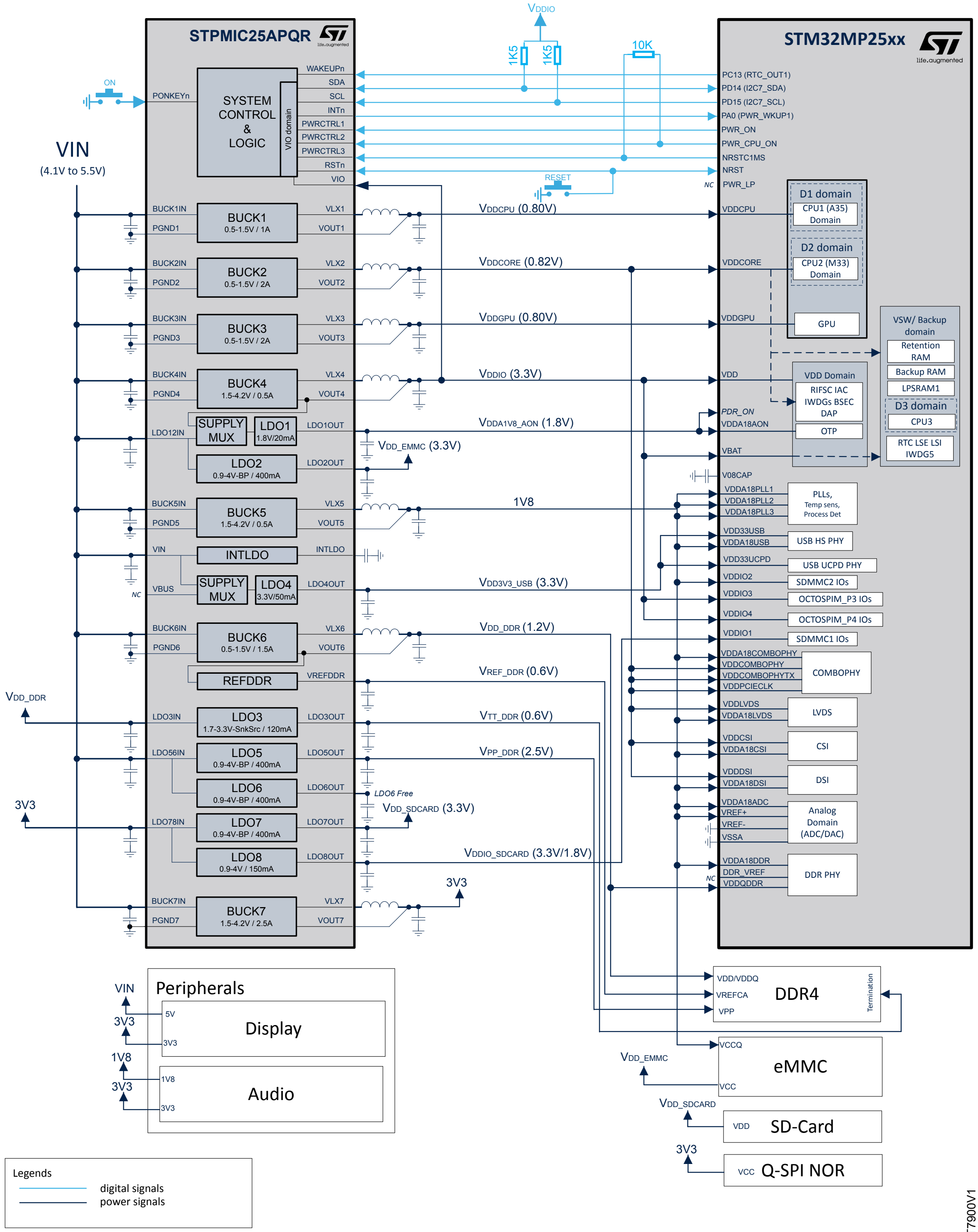
4 5 V power supply application reference design

This reference design targets an application powered by a 5 V power supply with 2 x DDR4, an eMMC, and an SD card.

The boot flash memory on eMMC. Other peripherals like audio and display are added to illustrate the application. A reference design with a USB high speed Type C power deliver is available on [1].

The main peripheral interfaces function with an I/O voltage of 3V3. The overall system is illustrated in Figure 1.

Figure 1. STM32MP25x and STPMIC25APQR with 2xDDR4, eMMC, SD card



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

- *STM32MP25x decoupling scheme (see [1])*
- *STPMIC25APQR discrete components value (see [2])*
- *V_{IN} source and related protection, such as ESD, EMI filtering, overvoltage.*

4.1 Power distribution

The STPMIC25APQR integrates the regulators that supply:
- The STM32MP25x power domains
- The application peripherals.

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

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

V_{DDCPU} is powered from the very efficient STPMIC25 BUCK1 step-down SMPS. The SMPS has an excellent *load transient response* across all operating conditions.

At power-up, the V_{DDCPU} is automatically enabled by the STPMIC25 at 800 mV, which corresponds to the "nominal mode" voltage. (See Section 5.2.1: Power-up triggered main supply (V_{IN}) plugin/power-down by software shutdown).

V_{DDCPU} is enabled in:
- Run1 mode
- Low power LP-Stop1 mode
- Low power LPLV-Stop1 mode.

V_{DDCPU} is disabled in the following modes:
- Run2 mode
- Low power LP-Stop2 mode
- Low power LPLV-Stop2 mode
- Low power Standby mode
- V_{BAT} mode and OFF mode.

In low-power modes, the PWR_CPU_ON output of the MPU manages the STPMIC25 V_{DDCPU} regulator. The PWR_CPU_ON is connected to the PWRCTRL2 input of the STPMIC25.

At runtime, the CPU1 can operate in *nominal mode* or in *overdrive mode*. The V_{DDCPU} voltage is then adjusted to the chosen mode. The MPU manages the transition between *nominal mode* voltage and *overdrive mode* voltage by sending I²C commands to the STPMIC25. The V_{DDCPU} voltage is increased to the "overdrive mode" voltage value (910 mV) when the CPU1 frequency (Fcpu1_overdrive) operates above 1200 MHz. When the CPU1 operates in "nominal mode" at 1200 MHz or below, the V_{DDCPU} must be set back to "nominal mode" voltage value (800 mV).

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

V_{DDCORE} is the main STM32MP25x digital power domain, and is called the D2 domain.

V_{DDCORE} supplies all the digital circuits, which include:
- The Arm® Cortex®-M33 platform (CPU2)
- Some analog IP of the MPU such as:
  - COMBOPHY
  - LVDS
  - CSI
  - DSI, and so on.

V_{DDCORE} is powered from the high efficiency STPMIC25 BUCK2 step-down SMPS. This SMPS has an excellent *load transient response* across operating conditions.

At power-up, V_{DDCORE} is automatically enabled by the STPMIC25 at 820 mV. (See Section 5.2.1: Power-up triggered main supply (V_{IN}) plugin/power-down by software shutdown)

V_{DDCORE} is enabled in:

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

The voltage is lowered to 670 mV in:

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

V_{DDCORE} is disabled in:

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

In low power mode, the PWR_ON output of the MPU manages the STPMIC25 V_{DDCORE} regulator. The PWR_ON output is connected to the PWRCTRL1 input of the STPMIC25. V_{DDCORE} also supplies the D3 backup domain and the retention domain in Run1 and Run2, and Stop1 and Stop 2 modes. (See Section 4.1.10: MPU backup domain and retention domain).

Note: D3 domain integrates all peripherals that must be functional in low power mode.

4.1.3 V_{DDGPU} power domain (800 mV/ 900 mV)

V_{DDGPU} supplies the graphic processing unit (GPU) and neural processing unit (NPU).

V_{DDGPU} is powered from the very efficient STPMIC25 BUCK3 step-down SMPS. This SMPS has an excellent load transient response across operating conditions.

At power-up, V_{DDGPU} is not automatically enabled by the STPMIC25.

At runtime, the MPU manages the V_{DDGPU} regulator by sending I²C commands to the STPMIC25. By default, the V_{DDGPU} regulator is set at 800 mV in “nominal mode” when the GPU frequency operates at 800 MHz or below. V_{DDGPU} regulator is set at 900 mV to run in “overdrive mode” when the GPU frequency ($f_{gpu\_overdrive}$) operates above 800 MHz. The V_{DDGPU} regulator is disabled during low power modes.

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

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

- V_{DD}
- V_{DDIO1}
- V_{DDIO2}
- V_{DDIO3}
- V_{DDIO4}.

V_{DDIO} is also the power supply of the MPU V_{DD} for the retention domain (see Section 4.1.10: MPU backup domain and retention domain for more details). These separate/dedicated I/O supplies can be set to different voltages or be shut down independently.

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

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

V_{DDIO} is powered from the STPMIC25 BUCK4 step-down SMPS, which is dedicated to the supply of sensitive powers domains and has a low output voltage ripple across all operating conditions. V_{DDA1V8_AON} is powered by the dedicated STPMIC25 LDO1 linear regulator, which has a very low quiescent current to reduce power consumption during low power mode.

At power-up, V_{DDIO} and V_{DDA1V8_AON} are automatically enabled to 3.3 V and 1.8 V respectively by the STPMIC25. They are the first regulators switched on at power-up (see Section 5.2.1: Power-up triggered main supply (V_{IN}) plugin/power-down by software shutdown).

The STPMIC25 LDO1 (V_{DDA1V8_AON}) has a built-in power supply multiplexor, which is either powered by LDO12IN (connected to V_{IN} supply at application level) or by the STPMIC25 BUCK4 output (VOUT4). By default, when the STPMIC25 is powered-up, the LDO12IN is selected as the LDO1 power source. Once the MPU is powered on and initialized, the software switches the LDO1 input source from LDO12IN to BUCK4 output (VOUT4) via an I²C command to the STPMIC25. This improves the power efficiency of the LDO1 from 36% (when the LDO4 input source is at 5 V from the LDO12IN) to 54% (when the LDO4 input source is at 3.3 V from the BUCK4 VOUT4).

Note: To set LDO1 input source from LDO12IN to VOUT4, the MPU software reprograms the STPMIC25 LDO1_MAIN_CR[INPUT_SRC] and LDO1_ALT_CR[INPUT_SRC] registers bit to 1 via an I²C commands (see document [2] for details).

In Standby mode, the BUCK4 step-down SMPS is set to low power mode. The MPU sets the STPMIC25 to Standby mode via the PWR_ON output connected to the PWRCTRL1 input of the STPMIC25 (see document [2] for details). V_{DDIO} and V_{DDA1V8_AON} are ON in all modes except in OFF or V_{BAT} mode; when the main power source (V_{IN}) of the application is removed (see Section 4.1.10: MPU backup domain and retention domain for details).

4.1.5 V_{DD3V3_USB} power domain

V_{DD3V3_USB} power domain supplies the USB2 HS PHY (V_{DD3V3_USB}) and the USB PD (power delivery) PHY ($V_{DD33UCPD}$) of the MPU.

Note: Examples of USB implementations with the STM32MP25x are provided in this document [1].

V_{DD3V3_USB} is powered from the dedicated STPMIC25 LDO4 with a fixed output voltage of 3.3 V.

Note: The STPMIC25 LDO4 has a built-in power supply multiplexor powered either from STPMIC25 VIN pin or VBUS pin, which automatically selects the highest input voltage. It is designed specifically for battery-powered applications to keep both the MPU USB PHY and PD PHY working when the battery is discharged. This feature is not applicable to this application note and the STPMIC25 VBUS pin must be left floating (unconnected).

At power-up, the V_{DD3V3_USB} regulator is automatically enabled to 3.3 V by the STPMIC25 (See Section 5.2.1: Power-up triggered main supply (V_{IN}) plugin/power-down by software shutdown).

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

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

V_{DD3V3_USB} is disabled in Standby and OFF mode. In this mode, the USB protection device (TCPP0x) must be shut down, so the USB CC lines are disconnected. In low power mode, MPU software controls the V_{DD3V3_USB} using I²C controls and it can be either switched ON/OFF.

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

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

This application focuses on DDR4 topology.

V_{DD_DDR} (1.2 V) is powered from the STPMIC25 BUCK6 step-down SMPS to power the DDR4 memory ICs (V_{DDR} and V_{DDQ}) and MPU DDR PHY (V_{DDQDDR}) domains.

V_{REF_DDR} (0.6 V) is powered from the STPMIC25 REFDDR sink source LDO. The supply source of the REFDDR LDO is internally connected to the BUCK6 output (V_{OUT6}) and provides a voltage equal to $V_{OUT6}/2$ to power the DDR4 memory V_{REFCA}.

Note:

- *The MPU DDR_VREF must remain unconnected.*
- *If BUCK6 is disabled but REFDDR is enabled, V_{REF_DDR} follows the output voltage of BUCK6 with an output voltage equal to $V_{OUT6}/2$.*

V_{TT_DDR} (0.6 V) is powered from the STPMIC25 LDO3 multipurpose LDO. When set in sink-source mode, the LDO3 provides voltage equal to V_{REF_DDR} (implicitly V_{OUT6} / 2). The LDO3 sink-source mode is dedicated to power the DDR4 memory bus terminations.

Note: To optimize power efficiency, the supply source of LDO3 (LDO3IN) is powered from BUCK6 output (V_{DD_DDR}).

V_{PP_DDR} (2.5 V) is powered from the STPMIC25 general purpose LDO5 to power the DDR4 memory V_{PP}.

At power-up, the STPMIC25 does not automatically start the following regulators:

- V_{DD_DDR}
- V_{TT_DDR}
- V_{PP_DDR}
- V_{REF_DDR}.

The regulators are powered up sequentially by the software bootloader by sending I²C commands to the STPMIC25. This is detailed in the following section and in Section 5.2.1: Power-up triggered main supply (V_{IN}) plugin/power-down by software shutdown.

At runtime, the PWR_ON output of the MPU, connected to the PWRCTRL1 input of the STPMIC25, manages the DDR4 power domains as follows:

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

At power down, and before the software turns OFF the STPMIC25, a power-down sequence must be applied on DDR4 power domains to comply with the JEDEC DDR4 specification (see details in Section 4.1.7: V_{DD_EMMC} power domain (3.3 V)).

Note: The STPMIC25 embeds configurable pull-down discharge resistors on each of the regulator outputs that allow all of regulator output voltages to discharge in less than 1.5 ms. For BUCK regulators, two pull-down values are configurable in NVM to either: slow pull-down and fast pull-down. Depending on the pull-down configuration, the discharge delay is set as follows: Slow PD =1.5 ms and Fast PD=0.3 ms. The "fast pull-down" configuration is used to switch off a power supply quickly than another one. The configuration is used on the DDR uncontrolled power down sequence

Software DDR4 power-up sequence:

1. Power-on event (or standby DDR-OFF mode exit): STPMIC25 power up (or standby DDR-OFF mode recovery)
2. The software bootloader executes DDR4 initialization.
3. The software bootloader sets the STPMIC25 internal pull-down for each DDR regulator as described in the note below:
   a. LDO5 (V_{PP_DDR}) pull down is disabled when LDO5 is disabled:
   LDOS_PD_CR1[LDO5_PD] = 0
   b. BUCK6 (V_{DD_DDR}) fast pull down is activated when BUCK6 is disabled:
   BUCKS_PD_CR2[BUCK6_PD]=10
   c. REFDDR (V_{REF_DDR}) pull down is activated when REFDDR is disabled:
   LDOS_PD_CR2[REFDDR_PD] = 1
   d. LDO3 (V_{TT_DDR}) pull down is activated when LDO3 is disabled:
   LDOS_PD_CR1[LDO5_PD] = 1

Note: It is necessary to set this pull down every time the STM32MP25x is powered up to prevent a "DDR4 uncontrolled power OFF sequence" (see DDR4 uncontrolled power off sequence:)

4. The software bootloader enables the DDR regulators:
   a. Enable LDO5 (V_{PP_DDR}) at 2.5 V
   b. Wait 1 ms tempo
   c. Enable LDO3 in sink-source mode (V_{TT_DDR})
   d. Enable REFDDR (V_{REF_DDR})
   e. Enable BUCK6 (V_{DD_DDR}) at 1.2 V
   f. Wait 1 ms tempo
5. Software initializes the MPU DDR4 memory controller and DDR4 ICs

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

Software DDR4 power down sequence:

1. Software receives an event to enter OFF mode or standby DDR-OFF mode.
2. Assert DDR CKE low
3. Disable BUCK6 (V_{DD_DDR})
4. Wait 1 ms tempo
5. Disable V_{REF_DDR}
6. Disable LDO3 (V_{TT_DDR})
7. Disable LDO5 (V_{PP_DDR})

Note: V_{PP_DDR} voltage falls slowly as LDO5 pull-down discharge when disabled

DDR4 uncontrolled power off sequence:

An uncontrolled power off sequence occurs typically when the main power supply voltage is removed or when a reset occurs.
Then the STPMIC25 manages the power off sequence:

Note: To ensure this uncontrolled power down sequence. At software boot, the STPMIC25 pull-down settings are set (see Software DDR4 power-up sequence:)

1. V_{IN} is removed.
2. V_{IN} crosses the VINOK_fall: STPMIC25 is in *turn off* condition.
3. STPMIC25 starts the power off sequence.
   a. STPMIC25 asserts a reset (NRST): STM32MP25x stops the DDRPHYC especially the DRAM differential clocks (DDRA_CKP/N and DDRB_CKP/N)
   b. STPMIC25 disables Rank0 regulators:
      - LDO5 (V_{PP_DDR})
      - BUCK6 (V_{DD_DDR})
      - LDO3 (V_{TT_DDR})
      - V_{REF_DDR}
4. The voltages of the following regulators are discharged through their respective regulator pulldown resistors:
   – BUCK6 (V_{DD_DDR})
   – LDO3 (V_{TT_DDR})
   – V_{REF_DDR}.
5. LDO5 (V_{PP_DDR}) falls slowly as it discharges through the LDO5 pull-down discharge when LDO5 is disabled.

4.1.7 V_{DD_EMMC} power domain (3.3 V)

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

V_{DD_EMMC} is powered from the STPMIC25 LDO2 general purpose linear regulator.

At power-up, the V_{DD_EMMC} regulator is enabled automatically by the STPMIC25 to 3.3 V (see Section 5.2.1: Power-up triggered main supply (V_{IN}) plugin/power-down by software shutdown).

V_{DD_EMMC} enables the MPU to have read/write access in the following modes:

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

When no read/write access is expected, the MPU software can disable V_{DD_EMMC}. V_{DD_EMMC} is disabled in the following modes:

- Stop mode
- LPLV-Stop2 mode,
- Standby mode
- OFF mode.

The MPU PWRCTRL pin can manage V_{DD_EMMC}. In low power mode, the V_{DD_EMMC} can be either switched ON or OFF by the application software.

eMMC as boot device

If the eMMC device is the boot flash peripheral, the application software must program the STPMIC25 as follows:

1. Power OFF the eMMC in Standby mode
2. Power ON the eMMC in Run1 mode before the application goes into Standby mode.

In this case, V_{DD_EMMC} is required for the first level boot of CPU1 and needs a power cycle to ensure a platform reboot. To carry out a power recycle, the V_{DD_EMMC} of the STPMIC25 regulator is managed by the NRSTC1MS output of the MPU. The NRSTC1MS output is connected to the PWRCTRL3 input of the STPMIC25 (see PWRCTRL1, PWRCTRL2, PWRCTRL3 for details).

4.1.8 SD card power domains (V_{DD_SDCARD}, V_{DDIO_SDCARD})

V_{DD_SDCARD} supplies the SD card memory device (V_{DD}). The V_{DDIO_SDCARD} is dedicated to power the built-in MPU level shifter (V_{DDIO1}) and consequently to support SD card UHS-I mode.

V_{DD_SDCARD} (3.3 V) is powered from the STPMIC25 BUCK7 step-down SMPS (3V3 node) via the STPMIC25 LDO7 acting as a power switch (bypass mode).

V_{DDIO_SDCARD} (3.3 V/1.8 V) is powered from the STPMIC25 BUCK7 step-down SMPS (3V3 node) via the STPMIC25 LDO8 acting as a power switch (LDO8 set in bypass mode) for 3.3 V output voltage or acting as an LDO (LDO8 set in LDO mode at 1.8 V) for 1.8 V output voltage.

Note: *The STPMIC25 LDO7 and LDO8 input (LDO78IN) are powered from BUCK7 output (3V3 node). LDO7 or LDO8 are set in bypass mode when V_{DD_SDCARD} or V_{DDIO_SDCARD} respectively are enabled and set in OFF mode when V_{DD_SDCARD} or V_{DDIO_SDCARD} respectively are disabled. The STPMIC25 bypass mode feature allows excellent power efficiency operation (close to 100%), compared to LDO mode where power efficiency is reaches 66% at best (V_{OUT}/V_{IN} = 3.3 V/5 V).*

At power-up, V_{DD_SDCARD} and V_{DDIO_SDCARD} regulators are not started automatically by the STPMIC25APQR.

Note: *If the SD card is a boot flash peripheral, then the STPMIC25APQR NVM must be reprogrammed to start LDO7 and LDO8 automatically.*

The voltage of the SD card data interface (V_{DDIO_SDCARD}) is changed at runtime depending on the speed settings of the SD card bus:

- Default bus speed (V_{DDIO_SDCARD} = 3.3 V):
  – V_{DD_SDCARD} is powered by LDO7 set in bypass mode (3.3 V)
  – V_{DDIO_SDCARD} is powered by LDO8 set in bypass mode (3.3 V)
- UHS-I bus speed (V_{DDIO_SDCARD} = 1.8 V):
  – V_{DD_SDCARD} is powered by LDO7 set in bypass mode (3.3 V)
  – V_{DDIO_SDCARD} is powered by LDO8 set in LDO mode at 1.8 V

SD card as boot device

As already mentioned in this section, the STPMIC25APQR does not automatically start at power up to the following:

- LDO7
- LDO8.

The SD card is not used as a boot device as described in the Overview.

Note: *If the SD card device requires to be a boot flash peripheral, the production test software (used in mass production to test and to tune the end-product) must reprogram the STPMIC25 NVM to start LDO7 and LDO8 automatically at power-up. For example, to start automatically the LDO7 and LDO8 in bypass mode in STPMIC25 RANK4 (see document [2] for details).*

In addition, as eMMC boot flash, the signal NRSTC1MS connected to PWRCTRL3 is used to power cycle the SD card regulators (LDO7 and LDO8) in case of D1 crash. See Section 5.4.2: CPU1 crash recovery management .

4.1.9 1V8 power domain

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

- PLLs ($V_{DDA18PLLx}$)
- USB2 PHY ($V_{DDA18USB}$)
- COMBOPHY ($V_{DDA18COMBOPHY}$)
- LVDS ($V_{DDA18LVDS}$)
- CSI ($V_{DDA18CSI}$)
- DSI ($V_{DDA18DSI}$)
- ADC/DAC ($V_{DDA18ADC}$)
- DDR PHY($V_{DDA18DDR}$).

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

The 1V8 power domain is powered from the STPMIC25 BUCK5 step-down SMPS, which has reduced output voltage ripple across operating conditions.

At power-up, the 1V8 regulator is automatically enabled by the STPMIC25 (See Section 4.2.2: STPMIC25APQR digital control interface).

At runtime, 1V8 is:

- Enabled in:
  - Run1 mode
  - Run2 mode
  - Low power LP-Stop1 mode
  - Low power LP-Stop2 mode
  - Low power LPLV-Stop1 mode
  - Low power LPLV-Stop2 mode
- Disabled in:
  - Standby mode
  - V_{BAT}/OFF modes.

4.1.10 MPU backup domain and retention domain

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

- The backup domain supplies:
  - RTC
  - Watchdogs
  - LSE
  - LSI
  - LPSRAM1
  - Retention RAM
  - Backup RAM
  - D3 domain.

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

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

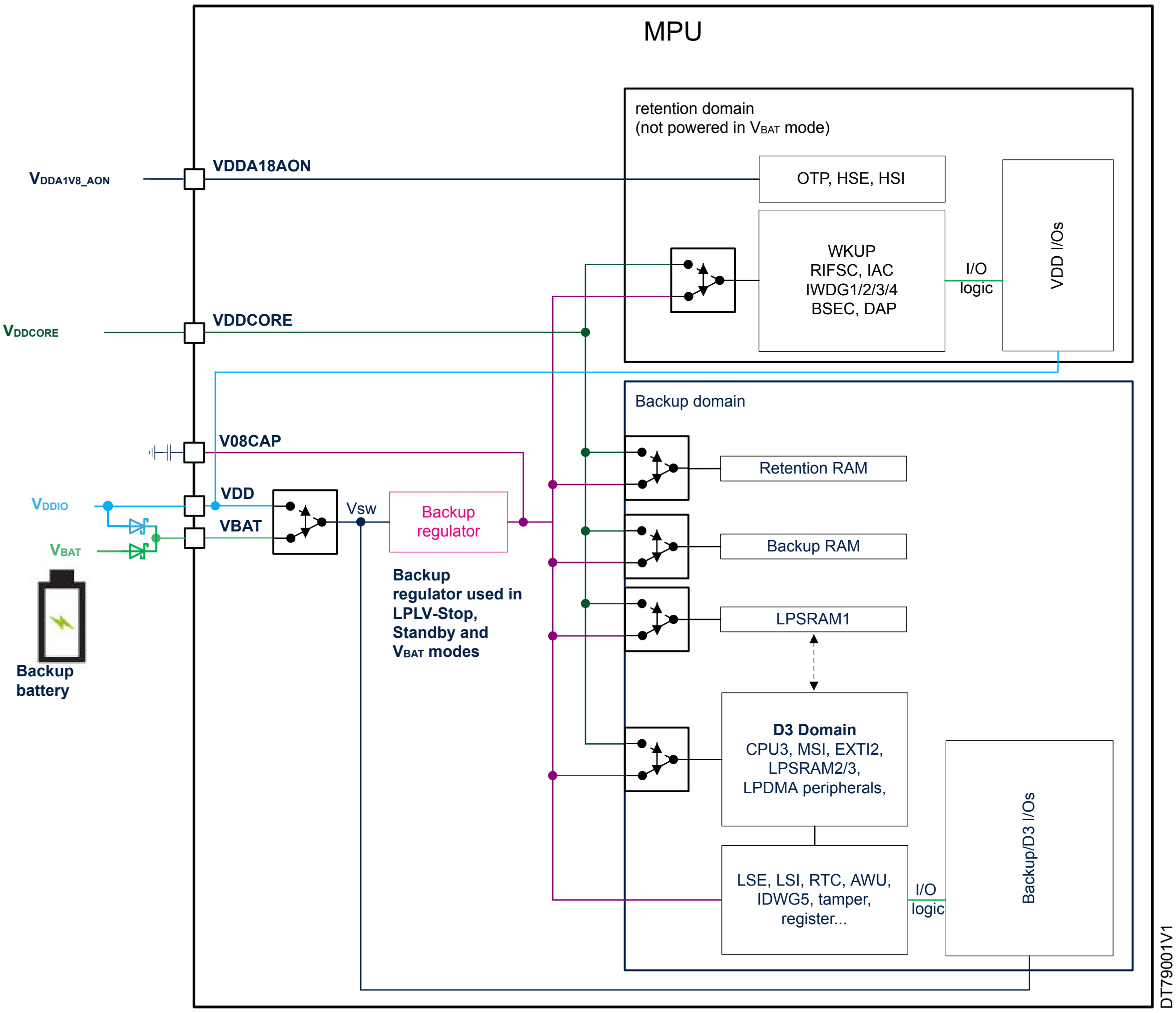
The retention domain is powered in all operating modes, excluding V_{BAT} mode.

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

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

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

Figure 2. Backup and retention domain



For more details on the constraints on V_{BAT} rise time, refer to the document [1].

4.1.11 3V3 external peripherals power domain

The 3V3 power domain is used to power peripherals around the MPU, such as the display, audio, and so on. It is also used to supply the STPMIC25 LDO7 and STPMIC25 LDO8 set in bypass mode in this application to power the SD card.

3V3 is powered from the STPMIC25 BUCK7 step-down general purpose SMPS. This regulator is designed to allow high current consumption.

4.2 Control signals between STPMIC25APQR and STM32MP25x

This section outlines the way the STM32MP25x microprocessor communicates with the STPMIC25APQR device. Several interfaces are available depending on the application requirements.

4.2.1 STPMIC25APQR default behavior with STM32MP25x

The STPMIC25APQR NVM settings are configured to boot the MPU application from flash memory such as eMMC or to boot the MPU from the USB interface. In production, it is used for flashing and then executing the software. The default NVM configuration is available in [2] and summarized in the following table.

Table 4. **Default NVM configuration**

Regulator	Name	Rank	Default output voltage	Default configuration
BUCK1	V_{DDCPU}	RANK3	0.8 V	ON_HP
BUCK2	V_{DDCORE}	RANK2	0.82 V	ON_HP
BUCK3	V_{DDGPU}	RANK0	NA	Software management
BUCK4	V_{DDIO}	RANK1	3.3 V	ON_HP
BUCK5	1V8	RANK3	1.8 V	ON_HP
BUCK6	V_{DD_DDR}	RANK0	NA	ON_HP
BUCK7	3V3	RANK4	3.3 V	Software management
LDO1	V_{DDA1V8_AON}	RANK1	1.8 V	ON
LDO2	V_{DD_EMMC}	RANK4	3.3 V	Software management
LDO3	V_{TT_DDR}	RANK0	N/A	ON
LDO4	V_{DD3V3_USB}	RANK5	3.3 V	Software management
LDO5	V_{PP_DDR}	RANK0	N/A	ON
LDO6	Free (not used)	RANK0	N/A	OFF
LDO7	V_{DD_SDCARD}	RANK0	N/A	OFF
LDO8	V_{DDIO_SDCARD}	RANK0	N/A	OFF
REFDDR	V_{REF_DDR}	RANK0	N/A	ON

In order to start the application, the STPMIC25 regulator start up is spread across five ranks. This is to comply with the MPU power-up sequence constraints and to avoid current peaks on the main power supply. The voltage value of each regulator is defined to fit with the MPU optimum voltage requirements.

For safety management

By default, when a fault occurs the STPMIC25APQR safety management features systematically restarts the application. The STPMIC25 performs a power cycling to systematically restart the application when a fault is detected by the STPMIC25, such as:

- V_{IN} undervoltage
- Over temperature
- Regulator OCP, and so on.

PMIC tuning (optional)

The STPMIC25 NVM can be reprogrammed by the customer to fine-tune the regulator settings, and the safety management behavior to fit with the MPU based application requirements. This action can be done with the STM32CubeProgrammer.

4.2.2 STPMIC25APQR digital control interface

The STPMIC25APQR integrates an I²C interface, five digital input control pins: PONKEYn, PWRCTRL1/2/3, WAKEUPn; a digital output interrupt pin (INTn), and a bidirectional digital reset pin (RSTn).

I²C interface

The MPU is controls the STPMIC25APQR via the I²C interface to:

- Enable/disable, set the voltage, and operating mode of the regulators.
- Dynamic voltage scaling to switch the MPU in nominal or overdrive modes.
- Set regulators external control for low power mechanisms (PWRCTRLx).
- Set the interrupt controller or read interrupt status.
- Set the protection for the watchdog, overcurrent, undervoltage, or read protection status.

- Tune the STPMIC25APQR NVM default configuration for the end-product (power-up sequence, safety management) see Section 5.3: Low power mode management for details

PONKEYn pin (optional)

The STPMIC25APQR PONKEYn pin is a digital active low input signal. It is usually connected to a user push button “ON/INT” to power on the application. Thanks to the STPMIC25 built-in pullup resistor, there is no need for a discrete pullup resistor on this signal.

The PONKEYn signal allows the following operations:

- Turns on the STPMIC25APQR (from PMIC OFF state)
- Wakes up the application from a low-power mode (typically from Standby mode).
- Forces a switch-off or a power cycling condition with a long press. This duration is programmable as described in section Section 5.1.1: Application turn-on/turn-off conditions .

Note: The use of a push “ON/INT” button connected to the PONKEYn is optional as the STPMIC25APQR is automatically turned on when the application is powered (see section Section 4.2.1: STPMIC25APQR default behavior with STM32MP25x).

RSTn pin

The STPMIC25APQR RSTn pin must be connected to the MPU NRST pin. It can also be connected to a “RESET” push button. This is achieved by using the STPMIC25 built-in pullup resistor, and no additional discrete pullup resistor is needed on this signal. Nevertheless, a 10 nF capacitor is fixed to the ground and placed as close as possible to the MPU. This is specifically required on this signal to avoid EMI/coupling as there is no debounce circuitry in neither the MPU, nor the STPMIC25.

The STPMIC25APQR RSTn pin is a digital active low bidirectional signal with a built-in pullup resistor:

- When STPMIC25APQR asserts an RSTn (such as during the power-up or the power-down sequence), it drives the MPU NRST signal low (open drain). The MPU is forced into a system reset until the STPMIC25APQR releases the RSTn.
- When the MPU asserts an NRST signal such as an MPU watchdog event, or by pressing the “RESET” button, the STPMIC25APQR immediately asserts the RSTn pin and performs a noninterruptible power cycle. The STPMIC25APQR performs a power down sequence, followed by a power-up sequence and releases the RSTn.

At the end of the power-cycle sequence, the STPMIC25APQR waits for the MPU NRST signal to go high before rearming the reset detection mechanism to avoid infinite loop reset.

INTn signal

The STPMIC25APQR INTn pin is a digital output (open drain) active low interrupt line connected to the MPU PA0 input pin. Thanks to the STPMIC25 built-in pullup resistor, a discrete pullup resistor is not required on this signal.

PA0 has both interrupt and wake-up capability:

- To manage an interrupt from the STPMIC25APQR when the MPU operates in either run mode or a low power mode (except Standby mode).
- To wake up the MPU when, it operates in Standby mode.

PWRCTRL1, PWRCTRL2, PWRCTRL3

The STPMIC25APQR has three power control digital input signals connected to dedicated MPU control signals.

Each STPMIC25APQR regulator can be controlled from a single PWRCTRL signal; typically to switch on or off, or depending on the PWRCTRL signal state to change the regulator output voltage. Alternatively, a PWRCTRL signal can be configured to reset a regulator at a value defined in STPMIC25APQR NVM (see document [2] for details)

MPU power control connection to STPMIC25APQR:

- The MPU PWR_ON output pin controls the STPMIC25APQR PWRCTRL1 input pin. In this application illustrated in Figure 1, the MPU PWR_ON is multiplexed with the MPU PWR_LP signal. To do this, PWR_D2CR.[LPCFG_D2] must be set to 1 by the software after a system reset.
  In this application, the multiplexing is selected so the MPU PWR_ON signal manages:
  - LPLV-Stop1 mode
  - LPLV-Stop2 mode
  - Standby mode

  When the multiplexing is not selected, the MPU PWR_ON signal only manages the Standby mode.
- The MPU PWR_CPU_ON output pin controls the STPMIC25APQR PWRCTRL2 pin. In the application illustrated in Figure 1, the MPU PWR_CPU_ON controls the MPU D1 domain (V_{DDCPU}) when this domain is in DStandby (see [3] for more details) set in LPLV-Stop2 mode and Standby low power mode. The internal SOC pwr_cpu_on signal can also be multiplexed with the internal SOC pwr_cpu_lp signal. To do this, the PWR_D1CR[LPCFG_D1] must be set to 0 by the software after a system reset. This keeps the PWR_CPU_ON = 1 in LP-Stop1 mode and LPLV-Stop1 mode.
- The MPU NRSTC1MS pin controls the STPMIC25APQR PWRCTRL3 pin. NRSTC1MS pin is used to control the power supplies of the external flash. This power is required for first level boot of CPU1 and needs a power cycle to ensure a platform reboot (eMMC). The NRSTC1MS pin is activated when a system reset is generated. In this application, the NRSTC1MS is linked to MPU PWR_CPU_ON by a 10 kΩ pull-up resistor. This is illustrated in Figure 1.

See Section 5.1.2: STPMIC25APQR power control management (PWRCTRLx) for more details about STPMIC25APQR PWRCTRL settings.

WAKEUPn (optional)

The WAKEUPn signal is driven by the MPU PC13 (RTC_OUT1) pin to control the STPMIC25APQR WAKEUP pin. It allows the MPU to power up the STPMIC25APQR, from STPMIC25APQR OFF state, or in MPU V_{BAT} mode, typically when the real-time clock timer elapses.

This is done by using the STPMIC25APQR built-in pullup resistor, therefore a discrete pullup resistor is not required on this signal.

5 Power management

5.1 Operating modes

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

Table 5. **Operating modes**

<table>
<tr><th>Operating mode</th><th>PMIC state</th><th>PWR_ON</th><th>PWR_CPU_ON</th><th>NRSTC1MS (1)</th><th>Description</th><th>Notes</th></tr>
<tr><td rowspan="7">Run1</td><td rowspan="7">POWER_ON (RUN)</td><td rowspan="7">1</td><td rowspan="7">1</td><td rowspan="7">1</td><td>V_{DDIO} power on</td><td rowspan="14">The difference between Run1/2 and Stop1/2 mode is only based on the STM32MP25x clock management and they have no impact on power management.</td></tr>
<tr><td>V_{DDCORE} power on</td></tr>
<tr><td>V_{DDCPU} power on (in normal or overdrive voltage)</td></tr>
<tr><td>V_{DDGPU} controlled by software</td></tr>
<tr><td>System clock on</td></tr>
<tr><td>Peripherals power on/off</td></tr>
<tr><td>DDR4 active</td></tr>
<tr><td rowspan="7">Run2</td><td rowspan="7">POWER_ON (RUN)</td><td rowspan="7">1</td><td rowspan="7">0</td><td rowspan="7">0</td><td>V_{DDIO} power on</td></tr>
<tr><td>V_{DDCORE} power on</td></tr>
<tr><td>V_{DDCPU} power off</td></tr>
<tr><td>V_{DDGPU} controlled by software</td></tr>
<tr><td>System clock on</td></tr>
<tr><td>Peripherals power on/off</td></tr>
<tr><td>DDR4 active</td></tr>
<tr><td rowspan="7">LP-Stop1 mode</td><td rowspan="7">POWER_ON (RUN)</td><td rowspan="7">0</td><td rowspan="7">1</td><td rowspan="7">1</td><td>V_{DDIO} power on</td><td rowspan="7">-</td></tr>
<tr><td>V_{DDCORE} power on</td></tr>
<tr><td>V_{DDCPU} power on (in normal voltage)</td></tr>
<tr><td>V_{DDGPU} controlled by software</td></tr>
<tr><td>System clock on</td></tr>
<tr><td>Peripherals power on/off</td></tr>
<tr><td>DDR4 self-refresh with V_{TT_DDR} power off</td></tr>
<tr><td rowspan="3">LP-Stop2 mode</td><td rowspan="3">POWER_ON (RUN)</td><td rowspan="3">0</td><td rowspan="3">0</td><td rowspan="3">0</td><td>V_{DDIO} power on</td><td rowspan="3">-</td></tr>
<tr><td>V_{DDCORE} power on</td></tr>
<tr><td>V_{DDCPU} power off</td></tr>
</table>

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

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

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

5.1.1 Application turn-on/turn-off conditions

The STPMIC25 autonomously manages the power-up and the power-down sequence when respectively a turn-on or a turn-off condition occurs.

The STPMIC25APQR automatically powers up when the application is powered from a valid power source: When V_{IN} rises above the V_{INOK_rise}, it triggers an STPMIC25 turn-on condition as the “AUTO turn-on” bit is set by default in the STPMIC25APQR NVM. The V_{INOK_rise} is the STPMIC25 internal threshold (see [2] for the values).

Turn-on conditions

When the application is in OFF mode (STPMIC25 in the OFF state with V_{IN} present), a turn-on condition is required to power up the STPMIC25, and then to run the application. Similarly, if the application needs to go into power-off mode, a turn-off condition is required to power-down the STPMIC25.

If the STPMIC25 is in OFF state, it is powered up by one of three external triggers described in the the table below and two internal ones:

Table 6. STPMIC25 turn-on conditions

Condition	Trigger	Description
AUTO turn-on	Internal	The STPMIC25APQR starts automatically when V_{IN} rises above V_{INOK_rise}. This feature is set by default in the STPMIC25APQR. (see “AUTO turn-ON” section in [2] for details)
PONKEY user button	External	PONKEYn pin voltage falling edge
Wake-up events from the STM32MP25x to STPMIC25 WAKEUPn pin	External	WAKE-UPn pin voltage falling edge

After a turn-on condition, the STPMIC25 carries out a transitional power-up sequence as describes in Section 5.2: Application Power-up/Power-down sequence.

Turn-off conditions

A turn-off condition leads the STPMIC25 to perform a power-down sequence to go into one of the following states:

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

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

Table 7. **Turn off conditions**

Condition	Hard fault	Description
Software switch OFF	NO	I²C commands “SWOFF” sent by the MPU to the STPMIC25
PONKEY user button long press	YES	If the PONKEYn signal is low for 10 s, an application reset is triggered by the user and is treated by the STPMIC25 as a hard fault condition. For practical implementation reasons, the mechanism management for this condition is the same as for hard fault condition. The STPMIC25 triggers a turn-off condition. The STPMIC25 powers down then powers up.(1)
V_{INOK_fall}	YES	If V_{IN} falls below the V_{INOK_fall} threshold, the STPMIC25 triggers a turn-off hard-fault condition. The STPMIC25 powers down and waits for V_{IN} to rise above the V_{INOK_rise} threshold again. At this point, the STPMIC25 powers up.(1)
Thermal shutdown	YES	If the STPMIC25 overheating occurs, the STPMIC25 triggers a turn-off hard fault condition: The STPMIC25 powers down and waits for the temperature to decrease, then the STPMIC25 powers up once again.(1)
Overcurrent protection	YES	If an overcurrent or a short-circuit appears on predefined regulators (see Section 6: Safety management section), the STPMIC25 triggers a turn-off hard-fault condition. The STPMIC25 powers down then powers up.(1)
Watchdog	YES	If the software enables the watchdog, and the watchdog timer has elapsed, the STPMIC25 triggers a turn-off hard-fault condition. The STPMIC25 powers down then powers up.(1)

1. *This is the default STPMIC25APQR behavior set in NVM (see [2] and Section 5.1.1: Application turn-on/turn-off conditions for details).*

After a turn-off condition, the STPMIC25 carries out a transitional power down sequence, starting by asserting the RSTn (MPU NRST), then turning off the regulators in reverse sequence to the power-up state. Once the STPMIC25 ends the power-down sequence, it restarts, or it goes to OFF state or in FAIL_SAFE_LOCK state depending on the turn-off condition source and occurrence. See Section 6: Safety management for details

5.1.2 STPMIC25APQR power control management (PWRCTRLx)

STPMIC25 PWRCTRLx signals are dedicated to managing MPU power modes or special regulator reset features. These signals must be correctly configured, and this before entering into low power mode, to ensure proper MPU low power mode entry and exit transitions.

The STPMIC25 PWRCTRLx signals are all independently controlled and each signal controls an STPMIC25 regulator by setting the appropriate registers as described in [2]. As such, it is possible to define:

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

Note: *One of these power controls can be used to switch the STPMIC25 state machine from Run1 or Run2 (Run) to Standby mode and another one can be used to suspend the STPMIC25 watchdog (typically when the application is in low power mode).*

An STPMIC25 PWRCTRL signal aims to switch between two regulator control registers xxxx_MAIN_CR and xxxx_ALT_CR.

Typically, when a PWRCTRL signal goes to a low state, the STPMIC25 internally switches from the main control register (MAIN) content to the alternate (ALT) control register content and vice versa.

STPMIC25APQR power control management for Standby mode

When the Standby mode is requested, the STPMIC25 state machine must switch to Standby to reach the minimum quiescent current consumption. For this operation, the STPMIC25 uses the PWRCTRL, which is dedicated to ensure the state machine transition from Standby to Run and vice versa. The STANBY_PWRCTRL_SEL[1:0] bit sets the PWRCTRL selection.

When the MPU runs in Standby mode, the consumption must be drastically reduced. The BUCK4 (V_{DDIO}) must be set to low power mode (LP) instead of the high power mode (HP) (see [2] chapter 4.3.1 for details). When this feature is set to a BUCK step down regulator, its performance is reduced, meaning the accepted BUCK rated output current is lowered, and clock synchronization is internally disabled.

Else in other MPU low power modes, the BUCK step down regulator works in high power mode (HP mode). See [2] chapter 4.3.1 for details.

STPMIC25APQR power control management independent reset source

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

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

This feature is specifically suitable to reset application with flash memories in case of D1 crash (see Section 5.4: System and CPU1 crash recovery management). In this application, the regulator independent reset source is activated by the PWRCTRL3 (MPU NRSTC1MS) and mapped to:

- LDO2 (V_{DD_EMMC})
- LDO7 (V_{DD_SDCARD})
- LDO8 (V_{DDIO_SDCARD})

5.1.3 STPMIC25APQR mask-reset option

If the application needs to have one or several STPMIC25 regulators enabled while the STPMIC25 performs a reset sequence, the MPU bootloader software must program the STPMIC25 mask reset option by setting the STPMIC25 BUCK_MRST_CR register to target BUCK converters, and LDO_MRST_CR register to target LDOs. A reset sequence is triggered after the STPMIC25 RSTn signal asserted by the MPU or the user reset push button. Refer to [2] for details on the STPMIC25 mask-reset option.

This is typically the case for the BUCK4 powering the MPU V_{DDIO} power domains and the LDO1 powering the MPU V_{DDA1V8_AON} power domain. The power cycle on V_{DDIO} and on V_{DDA1V8_AON} must be masked by setting these two STPMIC25 registers:

- BUCKS_MRST_CR [3] = 1
- LDOS_MRST_CR [0] = 1

This prevents losing the content in:

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

Note: The MPU software bootloader must program these settings using the I²C command to the STPMIC25, following each application power-up. The content of the BUCKS_MRST_CR, and LDOS_MRST_CR are reset at the end of an STPMIC25 reset cycle.

5.2 Application Power-up/Power-down sequence

The power up sequence is the transition managed by the STPMIC25APQR between power-off and Run mode. The application power-up and power-down sequence shown in Figure 3 is based on the reference designed in Figure 1.

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

When the application is connected to an external power supply from a power-off state, the application starts automatically when V_{IN} rises above the V_{INOK_rise} threshold. This assumes that the STPMIC25 has the AUTO_TURN_ON enabled by default in NVM settings.

When the STPMIC25 is powered-up, the PWRCTRLx signals have no effect (not initialized), the RSTn signal (MPU NRST signal) is released if no hard fault appears, then the application boots (including the DDR4 initialization). Finally, the system reaches Run mode and the MPU is operational.

When a turn-off condition occurs, the STPMIC25 powers down and goes into the OFF mode and the application goes into power-off mode. The whole process is detailed below and illustrated in Figure 3:

1. The application has no power or the MPU is powered by the onboard coin cell V_{BAT}. The V_{BAT} supplies the MPU backup domain.

2. A power supply is connected to the application. V_{IN} voltage rises.
3. Once V_{IN} supply is above V_{INPOR_Rise} (STPMIC25APQR V_{INPOR_Rise} = 2.3 V typ.)
   a. The STPMIC25 preloads its NVM contents and checks its integrity.
   b. If the STPMIC25 NVM integrity is valid, the STPMIC25 initializes, and its states machine goes directly in *CHECK&LOAD* state as the AUTO_TURN_ON bit is set in the NVM. This is defined in the Turn-on conditions section.
   c. Once the STPMIC25 is initialized, it waits for a valid V_{IN} voltage to power up. This means waiting for V_{IN} to rise above the V_{INOK_rise} threshold (STPMIC25APQR V_{INOK_rise} = 4 V typ).

Note: The STPMIC25 PRELOAD and CHECK&LOAD has a typical duration of approximately 6 ms.

4. Once the V_{IN} supply rises above V_{INOK_rise}, the STPMIC25 starts a power-up sequence. The STPMIC25 regulators follow the power-up sequence described below:
   a. RANK1: The following voltage domains are enabled:
      - V_{DDIO} (BUCK4) to 3.3 V.
      - V_{DDA1V8_AON} (LDO1) to 1.8 V.

      A threshold is applied to these supplies that is based on the MPU internal power-on reset threshold (MPU POR). This is described in [3]. Once this threshold is reached the MPU PWR_ON signal goes high, and the MPU enters in system reset. The MPU NRST signal is released at this time (following the $T_{RSTTEMPO}$= 500 µs). But in the application, the STPMIC25 RSTn signal asserts the MPU NRST signal and takes the lead to control the MPU NRST signal.
   b. RANK2: The V_{DDCORE} (BUCK2) assigned to RANK2 is enabled at 0.82 V. An MPU threshold named Vddcore_ok set to 660 mV allows the MPU to enable the MPU PWR_CPU_ON signal and the MPU NRSTC1MS signal. The MPU NRSTC1MS is linked to the MPU PWR_CPU_ON by a 10 kΩ pull-up resistor. The MPU waits for the HSI oscillators to be available. At this point, the STPMIC25 has reached RANK2 and a delay of 1.5 ms is applied.
   c. RANK3: The following power domains are enabled:
      - V_{DDCPU} (BUCK1) to 0.80 V.
      - 1V8 (BUCK5) to 1.8 V.

      An MPU threshold named Vddcpu_ok set at 660 mV is applied on V_{DDCPU} (BUCK1), which allows the MPU to enter in hardware system initialization. At this point, the STPMIC25 has reached RANK3 and a delay of 1.5 ms is applied.
   d. RANK4: Enable both to 3.3 V:
      - 3V3 (BUCK7)
      - V_{DD_EMMC} (LDO2)

      A delay of 1.5 ms is applied.
   e. RANK5: V_{DD3V3_USB} (LDO4) is enabled. A 1.5 ms delay is applied.
   f. When all regulators are ON, the STPMIC25 releases the RSTn signal linked to the MPU NRST signal.
5. As both the MPU and the STPMIC25 release their respective reset pins, then the MPU NRST signal rises:
   a. The NRSTC1MS here asserted by NRST, signal rises.
   b. The MPU EADLY timer starts. Refer to EADLY timer for more information.
   c. The boot ROM starts accessing the external flash memory peripherals (eMMC) when the EADLY timer elapses. The objective is to load, to check, and to execute the bootloader software.
   d. The software controls the V_{DDGPU} (BUCK3), V_{DD_SDCARD} (LDO7), V_{DDIO_SDCARD} (LDO8) regulators once the peripheral boots are completed.

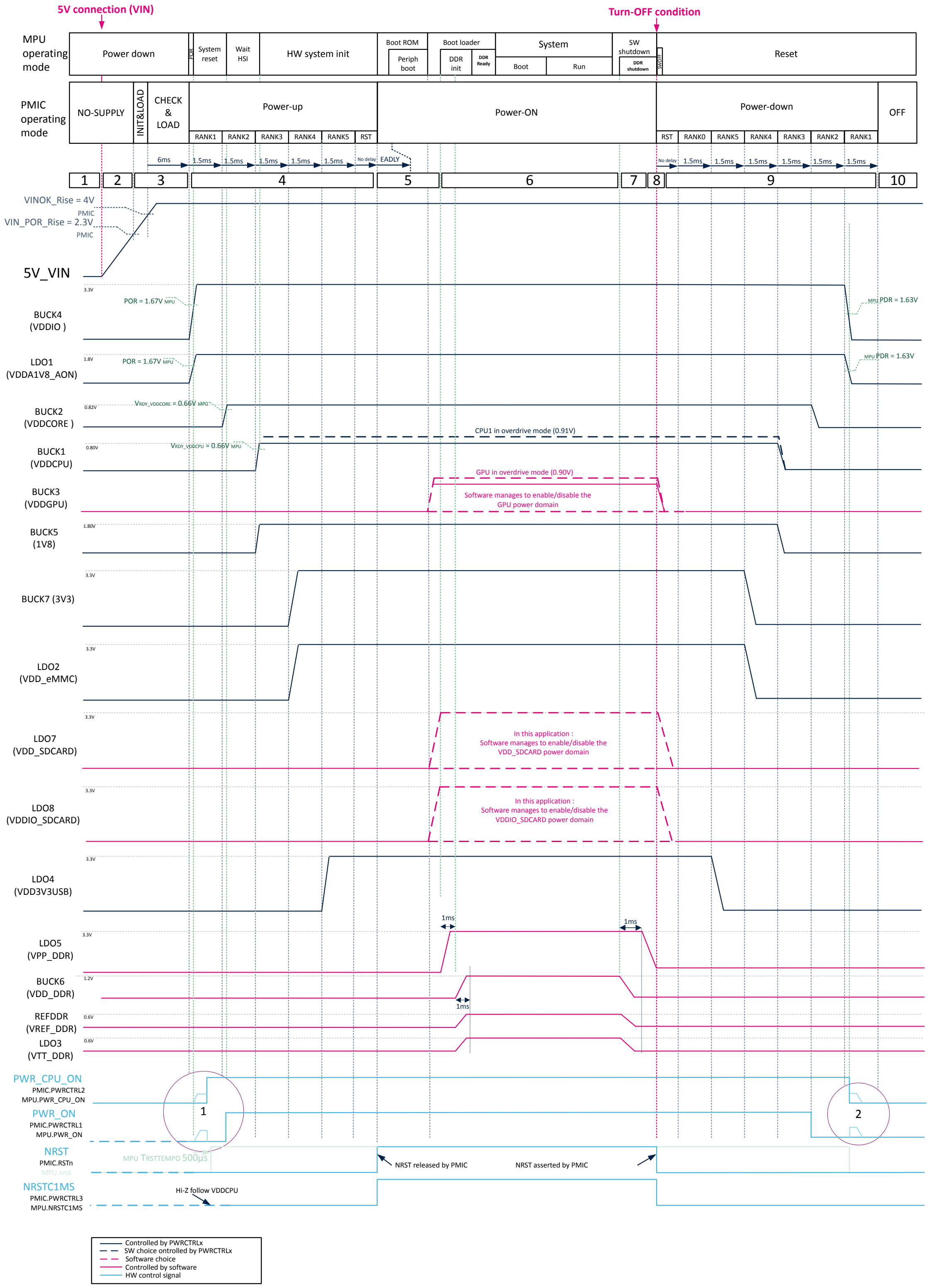
6. The bootloader initializes the DDR then loads and executes the kernel:
   a. The bootloader controls any STPMIC25 regulator.
   b. The V_{PP_DDR} (LDO5) is enabled at 2.5 V.
   c. The software waits for 1 ms.
   d. The software enables the following once the delay has elapsed:
      i. V_{REF_DDR} (REFDDR)
      ii. V_{TT_DDR} (LDO3 in sink-source mode)
      iii. V_{DD_DDR} (BUCK6 at 1.2 V)
   e. The software waits for 1 ms.
   f. The MPU software initializes the DDR4 controller and DDR memory ICs.
   g. The bootloader loads the kernel into DDR4 and executes it. The kernel initializes.
   h. The system is now running.
7. When a shutdown request occurs, the software prepares to power-off properly:
   a. The software shuts down the DDR4 regulators in the following sequence:
      i. Disable in sequence:
         1. V_{DD_DDR} (BUCK6)
         2. V_{TT_DDR} (LDO3)
         3. V_{REF_DDR} (REFDDR).
      ii. Then the software waits 1 ms.
      iii. Disable V_{PP_DDR} (LDO5).
   b. The DDR4 is off, and the MPU is ready to power down, once this sequence is done
8. The software sends the *SWOFF set to 1* command to the STPMIC25 by I²C to trigger an STPMIC25 turn-off condition .
9. The STPMIC25 performs a power-down sequence:
   a. The STPMIC25 asserts the RSTn, asserting the MPU NRST signal.
   b. RANK0: The STPMIC25 disables the V_{DDGPU} (BUCK3), V_{DD_SDCARD} (LDO7), and V_{DDIO_SDCARD} (LDO8) regulator that is not enabled at power-up.
      The system then waits for 1.5 ms.
   c. RANK5: V_{DD3V3_USB} (LDO4) is disabled and waits for 1.5 ms.
   d. RANK4: The power domains are disabled:
      - 3V3 (BUCK7).
      - V_{DD_EMMC} (LDO2).

      The system waits 1.5 ms.
   e. RANK3: The following power domains are disabled:
      - V_{DDCPU} (BUCK1)
      - 1V8 (BUCK5).

      The system then waits for 1.5 ms.
   f. Rank2: V_{DDCORE} (BUCK2) is disabled, the system waits for 1.5 ms.
      Once the V_{DDCORE} voltage goes below an internal MPU threshold, the MPU PWR_CPU_ON signals go low
   g. RANK1: The following power domains are disabled:
      - V_{DDIO} (BUCK4)
      - V_{DDA1V8_AON} (LDO1).

      The system waits 1.5 ms once the V_{DDIO} or V_{DDA1V8_AON} goes below the power-down reset threshold (PDR 1.63 V see [3] for details). The MPU enters in reset mode, the PWR _ON and the PWR_CPU_ON I/Os go in high-Z pull-down.
10. The STPMIC25 is now in OFF mode: the application is powered off.

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



PWR_ON and PWR_CPU_ON signals behavior during the power-up sequence

During the power-up sequence, the STPMIC25 PWRCTRL1/2 internal pull-up resistors, and the MPU PWR_ON and the MPU PWR_CPU_ON internal pull-down resistors induce a particular artifact on the PWR_ON and PWR_CPU_ON signals of the application. See Figure 4 for details.

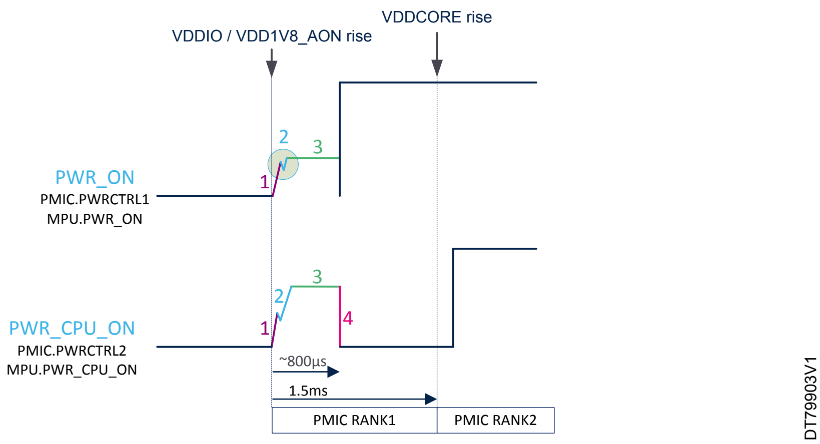
This artifact has no impact on the system behavior as PWRCTRLx signals are only probed by STPMIC25 once the application software is initialized and the STPMIC25 PWRCTRLx signals are allocated to regulators.

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

1. The STPMIC25 PWRCTRL1 and PWRCTRL2 internal pull-ups (80 kΩ typical) are active until the V_{DDIO} and V_{DDA1V8_AON} start. (1)
2. After few µs, the MPU.PWR_ON/PWR_CPU_ON IOs are in high-Z and internal pull-down resistor (40 kΩ typical) is activated on both signals. A resistor divider is formed by the two 80 kΩ pull-up resistors in the STPMIC25 and 40 kΩ pull-down in the MPU. This resistor divider applies to the PWR_ON and PWR_CPU_ON application signals respectively. Both voltages follow the V_{DDIO} rising voltage driven by STPMIC25 PWRCTRLx pull-up resistors. (2)
3. During approximately 800 µs (time for the internal MPU initialization), the MPU internal pull-down and the STPMIC25 internal pull-up are still active, so the V_{DDIO} voltage is divided by the resistor divider ratio. (3)
4. The MPU internal resistors on PWR_ON and PWR_CPU_ON are deactivated and MPU PWR_ON and PWR_CPU_ON pads are internally set in push-pull mode. PWR_ON and PWR_CPU_ON signals are immediately driven by the MPU to the expected level: PWR_ON goes high and PWR_CPU_ON is kept low until the V_{DDCORE} raises. (4)

Note: STPMIC25 PWRCTRLx pull-up resistors remain active and must be disabled by the bootloader software, after each the application powered-up to avoid extra power consumption in low power mode. This is when the PWR_ON and/or PWR_CPU_ON level is low.

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



PWRCTRLs behavior during the power down sequence

During the power down sequence, the STPMIC25 PWRCTRL1/2 internal pull-up resistor and the MPU.PWR_ON and MPU.PWR_CPU_ON internal pull-down resistors induce a particular artifact on the PWR_ON and PWR_CPU_ON signals of the application. The whole process is illustrated in Figure 5.

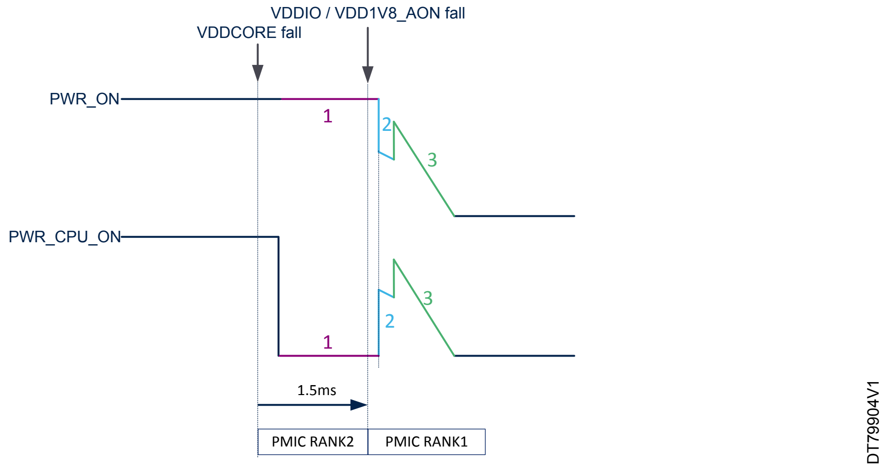
Note: This artifact has no impact on the system behavior as PWRCTRLx signals are not probed by STPMIC25 until the application software initializes and allocates the STPMIC25 PWRCTRLx signals to regulators.

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

1. Once the V_{DDCORE} voltage goes below vddcore_ok thresholds, the MPU.PWR_CPU_ON is deactivated and the MPU.PWR_ON remains active. MPU PWR_ON and PWR_CPU_ON pads are kept in push-pull mode. (1)

2. Once V_{DDIO} and V_{DDA1V8_AON} fall, the MPU.PWR_ON/PWR_CPU_ON internal pull-down resistor (40 kΩ typ.) are activated and the STPMIC25.PWRCTRL1/PWRCTRL2 internal pull-up (80 kΩ typ) are still present. (2). A resistor divider is formed by the two 80 kΩ pull-up resistors in STPMIC25 and 40 kΩ pull-down in MPU and applies to the respective PWR_ON and PWR_CPU_ON application signals. Both voltages follow V_{DDIO} falling voltage driven by STPMIC25 PWRCTRLx pull-up resistors.
3. In this state the STPMIC25 pull-up resistors are still present, but the MPU pull-down resistors are deactivated. Both PWR_ON and PWR_CPU_ON voltages follow V_{DDIO} falling voltage driven by STPMIC25 PWRCTRLx pull-up resistors. (3).

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



5.2.2 Power-down triggered by STPMIC25APQR hard fault (safety management)

The following events trigger:

- A harmful overcurrent
- V_{IN} fall
- An overtemperature
- A watchdog
- PONKEY long key press.

When the STPMIC25 detects a hard fault, it triggers a turn-off condition followed by a power-down sequence. Once the power-down sequence ends, the STPMIC25 can either restart (power up sequence) or it can go in FAIL_SAFE_LOCK state. The STPMIC25 safety management is detailed in Section 6: Safety management .

5.3 Low power mode management

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

Table 8. Low power mode supported by the application

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

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

Table 9. Power control signal connection

MPU output	STPMIC25 input
PWR_ON	PWRCTRL1
PWR_CPU_ON	PWRCTRL2
NRSTC1MS	PWRCTRL3

These signals control the regulators directly from the MPU state machine. This is because, in low power mode, no software is running and the STPMIC25 regulators cannot be controlled by any I²C command from software (see Table 5).

Before entering into low-power mode, the MPU software must prepare the STPMIC25 to enter any of these power modes by setting STPMIC25 xxxx_ALT_CR registers. This also includes changing low power modes by setting the STPMIC25 xxx_MAIN_CR registers with the expected STPMIC25 regulator settings for the targeted low-power mode behavior.

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

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

- Standby mode.

The delays are described in the following section.

5.3.1 STM32MP25x internal timer for low power mode management

EADLY timer

The EADLY timer is a programmable timer used to produce a defined wait period before the boot ROM performs any external access to the flash memory (eMMC, FMC-NAND, OCTOSPI, SD card). This ensures that the boot ROM is able to access the flash memory and reliably read the boot software. The EADLY timer duration is set to 5 ms by default after a system reset. It is recommended to keep this default value.

POPL timers

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

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

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

PODH_D2 timer

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

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

Note: The PODH_D2 timer must override the POPL_D1 timer.

LPLVDLY_D2 timer

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

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

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

The LPLVDLY_D2 timer must be set once by the software at 187µs (PWR_D2CR[LPLVDLY_D2]=0). This value is defined as follows: STPMIC25 has internal 20 µs delay between PWRCTRL rise and V_{DDCORE} regulators state change, in addition to a 150 µs worst case delay of V_{DDCORE} voltage recovery from retention (670 mV) to nominal (820 mV). So, a 170 µs total delay from PWRCTRL signal rising to V_{DDCORE} recovered at 820 mV.

Tdelay_V_{DDCPU}

Tdelay_V_{DDCPU} is an internal and fixed delay (value defined in [2]), that is triggered when V_{DDCPU} reaches the threshold Vcpu_rdy. As long as V_{DDCPU} is below Vcpu_rdy, the domain is reset. Tdelay_V_{DDCPU} is used to wait for V_{DDCPU} to reach its nominal value when the system exits from LP-Stop2 mode or LPLV-Stop2 mode or Standby mode.

Tdelay_V_{DDCORE}

Tdelay_V_{DDCORE} is an internal and fixed delay (value defined in [2]) that is triggered started when V_{DDCORE} reaches the threshold Vcore_rdy. As long as V_{DDCORE} is below Vcore_rdy the domain is reset. Tdelay_V_{DDCORE} is used to wait for V_{DDCORE} to reach its nominal value when the system exits from Standby mode.

PWRLP_TEMPO timer

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

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

C1MSRD timer

The C1MSRD is a programmable timer defining the minimum pulse duration of the NRSTC1MS signal when the system exits Standby mode.

The C1MSRD timer must be set by software at cold boot (boot loader). Recommended values are proposed in Section 5.3.4: Standby mode (DDR4 in self-refresh).

MRD timer

The MRD is a programmable timer defining the minimum pulse duration of the NRST system reset signal. This timer is useful when the supply is provided by a discrete power component, so it can be set to 0 with an STPMIC25.

5.3.2 LP-Stop1/LPLV-Stop1 mode

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

- In LPLV-Stop1, the V_{DDCORE} voltage is reduced
- In LP-Stop1, the V_{DDCORE} is kept at nominal value.

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

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

LP-Stop1 mode

The LP-Stop1 mode is described below and is shown in Figure 6 based on to the implementation shown in Figure 1.

1. The application is powered up and is running in Run1 operating mode; all PWR_ON and PWR_CPU_ON are in high state. In this application, the PWR_LP signal is multiplexed with the PWR_ON signal.
2. When the LP-Stop1 mode is requested, the software prepares to enter LP-Stop1 mode:
    a. The MPU performs internal settings such as:
        i. Set PWRLP_TEMPO timer to 100 µs (in RCC_PWRLPDLYCR register)
        ii. Stop the appropriate clocks
        iii. Set the DDR4 to self-refresh.
    b. The MPU performs STPMIC25 settings (see Table 10)
    c. The MPU PWR_CPU2CR[LPDS_D2] and PWR_CPU1CR[LPDS_D1] bits are enabled. This allows the system to enter into the LP-Stop1 low power mode.

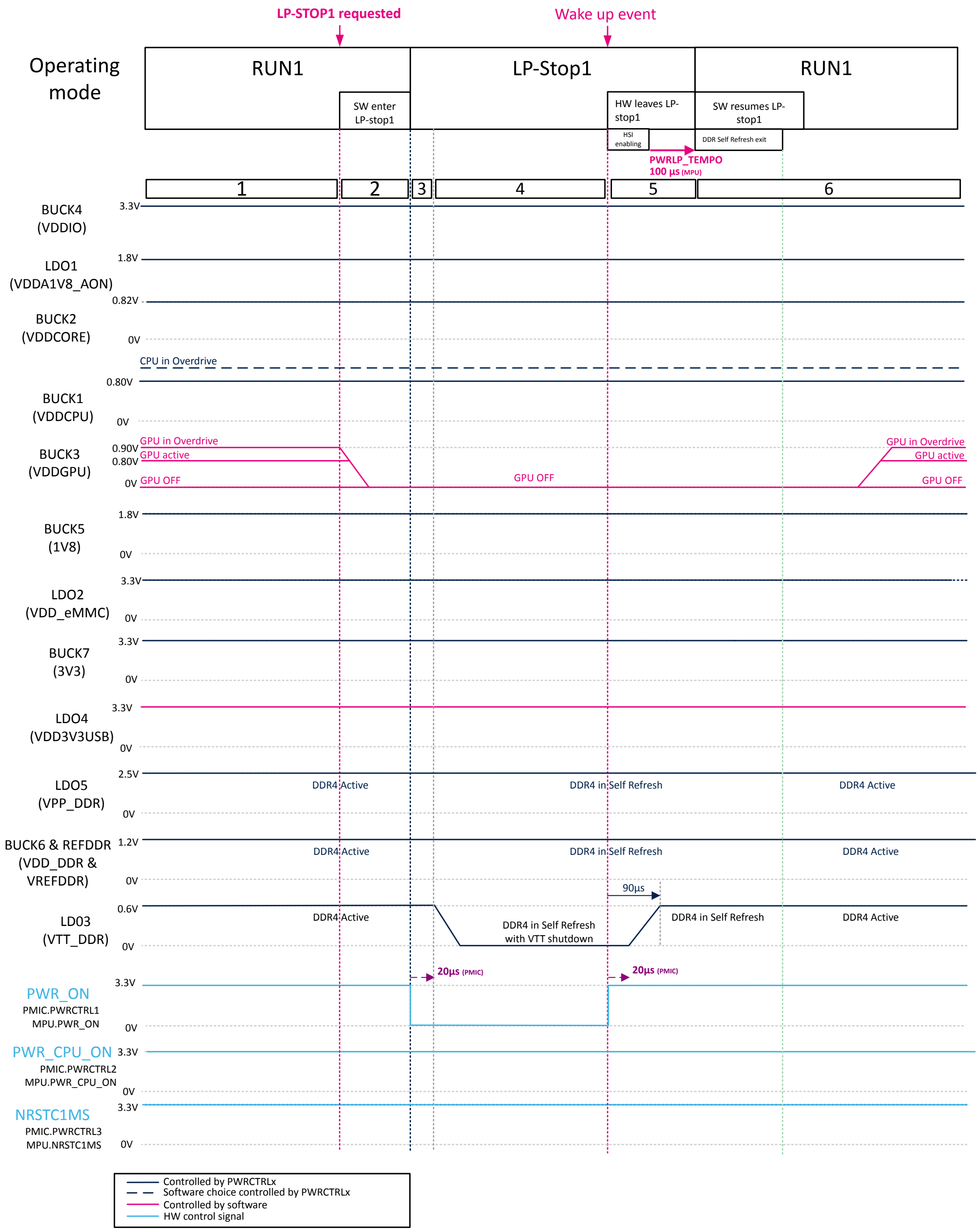
3. Once the system is in LP-Stop1:
   a. The MPU PWR_ON output is deasserted. The STPMIC25 PWRCTRL1 goes low
4. After 20µs internal STPMIC25 delay, the STPMIC25 regulators, which is connected to the PWR_ON (linked to the PWRCTRL1) takes the configuration set in the xxx_ALT_CR registers (see Table 10): the V_{TT_DDR} regulator turns OFF.
5. On a wake-up event, the MPU leaves the LP-Stop1 mode:
   a. The MPU PWR_ON output signal is asserted, driving the STPMIC25 PWRCTRL1 signal high.
   b. After 20µs internal STPMIC25 delay, the STPMIC25 regulators, which is connected to the PWR_ON takes the configuration set in the xxx_MAIN_CR registers (see Table 10):
      i. The V_{TT_DDR} regulator turns ON
      ii. The V_{TT_DDR} voltage rises in 70 µs (90 µs total).
   c. Clocks are enabled and the software executes the PWRLP_TEMPO timer (100 µs). This delay ensures that the V_{TT_DDR} as reaches its nominal value before:
      i. The system enters in Run1
      ii. The software moves the DDR4 out of self-refresh mode.
6. Once the PWRLP_TEMPO timer has elapsed, the application goes into Run1 mode. The software resumes LP-Stop1: DDR4 exit from self-refresh.

Table 10. **STPMIC25 configuration for LP-Stop1 mode**

Regulator	PWRCTRLx affectation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	VOUT	Configuration	VOUT
BUCK1 (V_{DDCPU})	PWR_CPU_ON	ON HP	0.80	OFF	-
BUCK2 (V_{DDCORE})	PWR_ON	ON HP	0.82	ON HP	0.82
BUCK3 (V_{DDGPU})	-	ON/OFF HP	0.80	-	-
BUCK4 (V_{DDIO})	PWR_ON	ON HP	3.3	ON_HP	3.3
LDO1 (V_{DDA1V8_AON})	-	ON	N/A	-	N/A
BUCK5 (1V8)	PWR_ON	ON HP	1.8	ON HP	1.8
BUCK6 (V_{DD_DDR})	PWR_ON	ON HP	1.2	ON HP	1.2
BUCK7 (3V3)	PWR_ON	ON HP	3.3	ON HP	3.3
V_{REF_DDR}	PWR_ON	ON	N/A	ON	N/A
LDO5 (V_{PP_DDR})	PWR_ON	ON	2.5	ON	2.5
LDO2 (V_{DD_EMMC})	NRSTC1MS (1)	ON/OFF	3.3	-	-
LDO3 (V_{TT_DDR})	PWR_ON	SINK SOURCE	N/A	OFF	-
LDO4 (V_{DD3V3_USB})	-	ON/OFF	N/A	-	N/A

1. *The LDO2 is controlled from PWRCTRL3 in reset mode (STPMIC25 PWRCTRL_RST bit set for LDO2)*

Figure 6. LP-Stop1 sequence



LPLV-Stop1 mode

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

1. The application is powered up and is operating in Run1 mode; all PWR_ON and PWR_CPU_ON are in high state. In this application, the PWR_LP signal is multiplexed with the PWR_ON.
2. When the LPLV-Stop1 mode is requested, the software prepares to enter LPLV-Stop1 mode:
   a. The MPU configures the internal settings such as:
      i. Disable the PWRLP_TEMPO timer (set RCC_PWRLPDLYCR[PWRLP_DLY[21:0]]=0).
      ii. Stop the appropriate system clocks.
      iii. Set the DDR into self-refresh.
   b. The MPU configures the STPMIC25 as defined in Table 11.
   c. The MPU PWR_CPU2CR[LPDS_D2] and PWR_CPU2CR[LVDS_D2] bit are enabled. The regulator enters into LPLV-Stop1 low power mode and V_{DDCORE} is then lowered to LPLV-Stop1 mode value.
3. Once the system is in LPLV-Stop1:
   a. The MPU PWR_ON output is deasserted, and the STPMIC25 PWRCTRL1 goes low.
4. After 20 µs internal STPMIC25 delay, the STPMIC25 regulators assigned to PWR_ON (linked to the PWRCTRL1) take the configuration set in the xxx_ALT_CR registers, which is defined in Table 11:
   a. V_{TT_DDR} regulator turns OFF.
   b. V_{DDCORE} regulator output decreases to retention voltage.
5. On a wake-up event, the MPU leaves the LPLV-Stop1 mode:
   a. The MPU PWR_ON output signal is asserted, driving the STPMIC25 PWRCTRL1 signal high and the MPU executes the LPLVDLY_D2 timer (initialized to 187µs) to wait for the V_{DDCORE} to switch from retention voltage to nominal voltage.
   b. After 20 µs internal STPMIC25 delay, the STPMIC25 regulators assigned to PWR_ON take the configuration set in the xxx_MAIN_CR registers (see Table 11):
      i. The V_{TT_DDR} regulator turns ON (V_{TT_DDR} voltage rise in 70 µs)
      ii. The V_{DDCORE} regulator switch from retention voltage (670 mV) to nominal voltage (820 mV) in 150 µs.
   c. Once the LPLVDLY_D2 timer timer elapsed, clocks are enabled.
   d. Once clocks are stable, the MPU goes immediately in Run1 mode (as PWRLP_TEMPO timer is bypassed).
6. The software resumes from LPLV-Stop1, exits DDR4 from self-refresh.

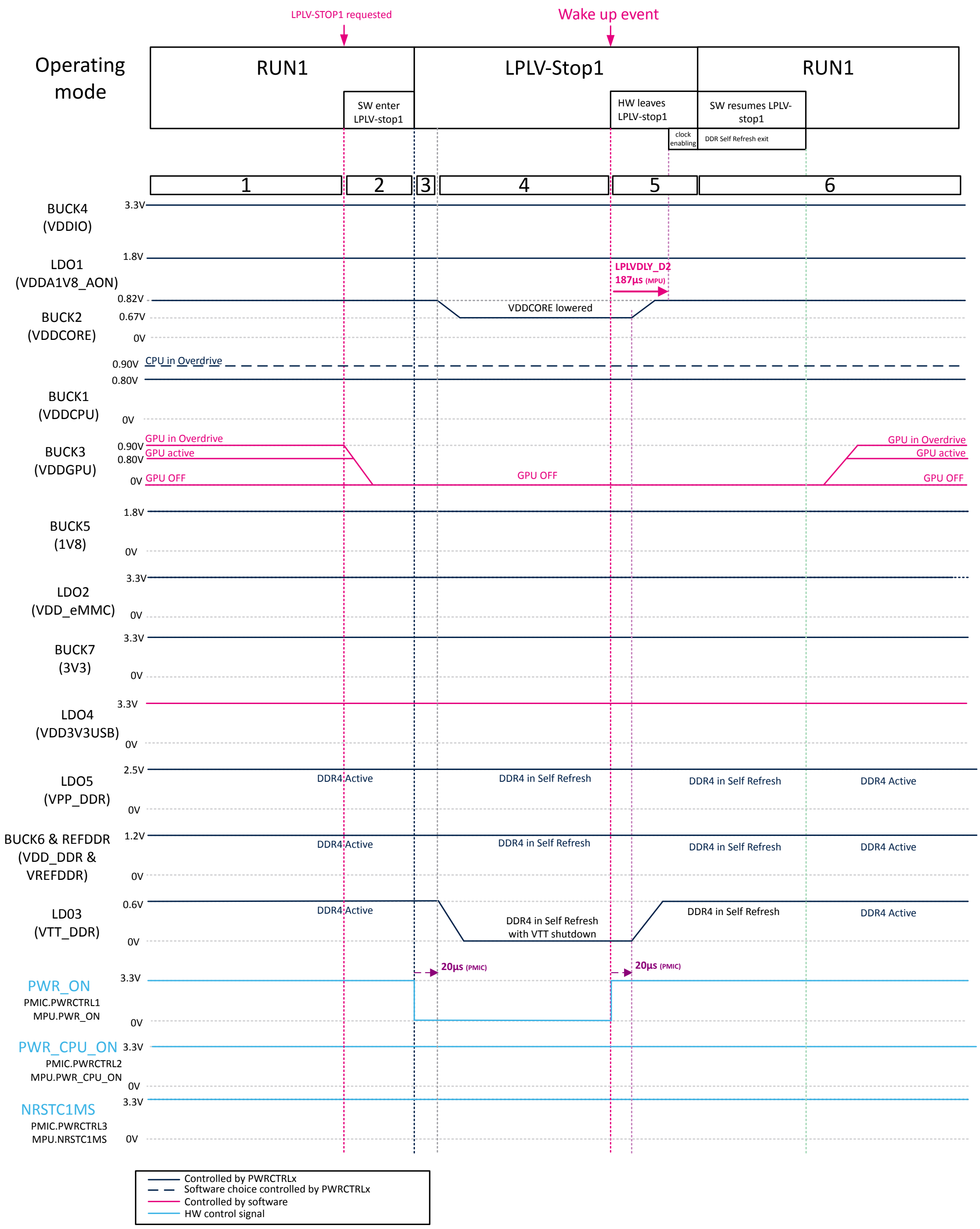
Table 11. STPMIC25 configuration for LPLV-Stop1 mode

Regulator	PWRCTRLx affectation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	VOUT	Configuration	VOUT
BUCK1 (V_{DDCPU})	PWR_CPU_ON	ON HP	0.80	OFF	-
BUCK2 (V_{DDCORE})	PWR_ON	ON HP	0.82	ON HP	0.67
BUCK3 (V_{DDGPU})	-	ON/OFF HP	0.80	-	-
BUCK4 (V_{DDIO})	PWR_ON	ON HP	3.3	ON_HP	3.3
LDO1 (V_{DDA1V8_AON})	-	ON	N/A	-	N/A
BUCK5 (1V8)	PWR_ON	ON HP	1.8	ON HP	1.8

Regulator	PWRCTRLx affectation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	VOUT	Configuration	VOUT
BUCK6 (V_{DD_DDR})	PWR_ON	ON HP	1.2	ON HP	1.2
BUCK7 (3V3)	PWR_ON	ON HP	3.3	ON HP	3.3
V_{REF_DDR}	PWR_ON	ON	N/A	ON	N/A
LDO5 (V_{PP_DDR})	PWR_ON	ON	2.5	ON	2.5
LDO2 (V_{DD_EMMC})	NRSTC1MS (1)	ON/OFF	3.3	-	-
LDO3 (V_{TT_DDR})	PWR_ON	SINK SOURCE	N/A	OFF	N/A
LDO4 (V_{DD3V3_USB})	-	ON/OFF	N/A	-	N/A

1. *The LDO2 is controlled from PWRCTRL3 in reset mode (STPMIC25 PWRCTRL_RST bit set for LDO2)*

Figure 7. LPLV-Stop1 sequence



5.3.3 LP-Stop2/LPLV-Stop2 mode

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

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

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

LP-Stop2 mode

The LP-Stop2 mode is described below and is shown in the Figure 8 based on to the implementation shown in Figure 1.

1. The application is powered up and operating in Run1 mode; all PWR_ON and PWR_CPU_ON are in high state. In this application, the PWR_LP signal is mux with the PWR_ON.
2. When LP-Stop2 mode is requested, the software prepares to enter LP-Stop2 mode:
   a. The MPU configures the internal settings such as:
      i. Set PWR_D1CR[POPL_D1] = 3 ms timer, to define a minimum pulse duration of PWR_CPU_ON. This ensures the full discharge of the V_{DDCPU} voltage before restarting.
      ii. Disable the PWRLP_TEMPO timer (set RCC_PWRLPDLYCR[PWRLP_DLY[21:0]]=0).
      iii. Stop the appropriate system clocks.
      iv. Set the DDR to self-refresh.
   b. The MPU configures the STPMIC25 as described in Table 12.
   c. The PWR_CPU2CR[LPDS_D2] bit is enabled. This allows the regulator to enter in low power LP-Stop2 mode.
   d. The PWR_CPU1CR[PDDS_D1] bit is enabled. This allows the D1 domain to enter in DStandby with V_{DDCPU} switched OFF.
3. Once the system is in LP-Stop2:
   a. The MPU PWR_ON is deasserted and the STPMIC25 PWRCTRL1 goes low.
   b. The MPU PWR_CPU_ON is deasserted and the STPMIC25 PWRCTRL2 goes low.
   c. The NRSTC1MS signal follows the PWR_CPU_ON signal (STPMIC25 PWRCTRL3 goes low).
   d. The MPU POPL_D1 timer is started to keep the D1 domain powered off until the POPL_D1 timer has elapsed. The wake-up event is shifted until POPL_D1 has elapsed.
4. After 20 µs internal STPMIC25 delay, the STPMIC25 regulators assigned to PWR_ON (linked to the PWRCTRL1) and PWR_CPU_ON (linked to the PWRCTRL2) and NRSTC1MS (linked to the PWRCTRL3) takes the configuration set in the xxx_ALT_CR registers (see Table 12):
   a. V_{TT_DDR} regulator turns OFF
   b. V_{DDCPU} regulator turns OFF
   c. V_{DD_EMMC} regulator turns OFF

5. On a wake-up event, the MPU leaves the LP-Stop2 mode:
   a. The MPU PWR_ON signal is asserted (STPMIC25 PWRCTRL1 goes high) and the clock are enabled.
   b. After a 20 µs internal STPMIC25 delay, the STPMIC25 regulators assigned to the PWR_ON takes the configuration set in the xxx_MAIN_CR registers (see Table 12):
      i. The V_{TT_DDR} regulator turns ON
      ii. The V_{TT_DDR} voltage rise in 70 µs (90 µs total)
   c. Once the clocks are stable, the PWRLP_TEMPO timer is bypassed and set to 0. In parallel the MPU PWR_CPU_ON is asserted.
   d. The NRSTC1MS signal follows PWR_CPU_ON.
   e. Following a further 20 µs internal STPMIC25 delay, the STPMIC25 regulators assigned to the PWR_CPU_ON and NRSTC1MS take the configuration set in the xxx_MAIN_CR registers (see Table 12): V_{DDCPU} and V_{DD_EMMC} rise.
   f. Once the V_{DDCPU} voltage reaches the vcpu_rdy threshold, the internal Tdelay_V_{DDCPU} is started to wait for V_{DDCPU} to reach its nominal voltage value.
   g. Once the Tdelay_V_{DDCPU} is elapsed, the CPU1 releases the clock domain to enter in Run1 mode. In this case, the Cortex®-A35 (CPU1) is set as master and the Cortex®-M33 (CPU2) follows the Cortex®-A35.

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

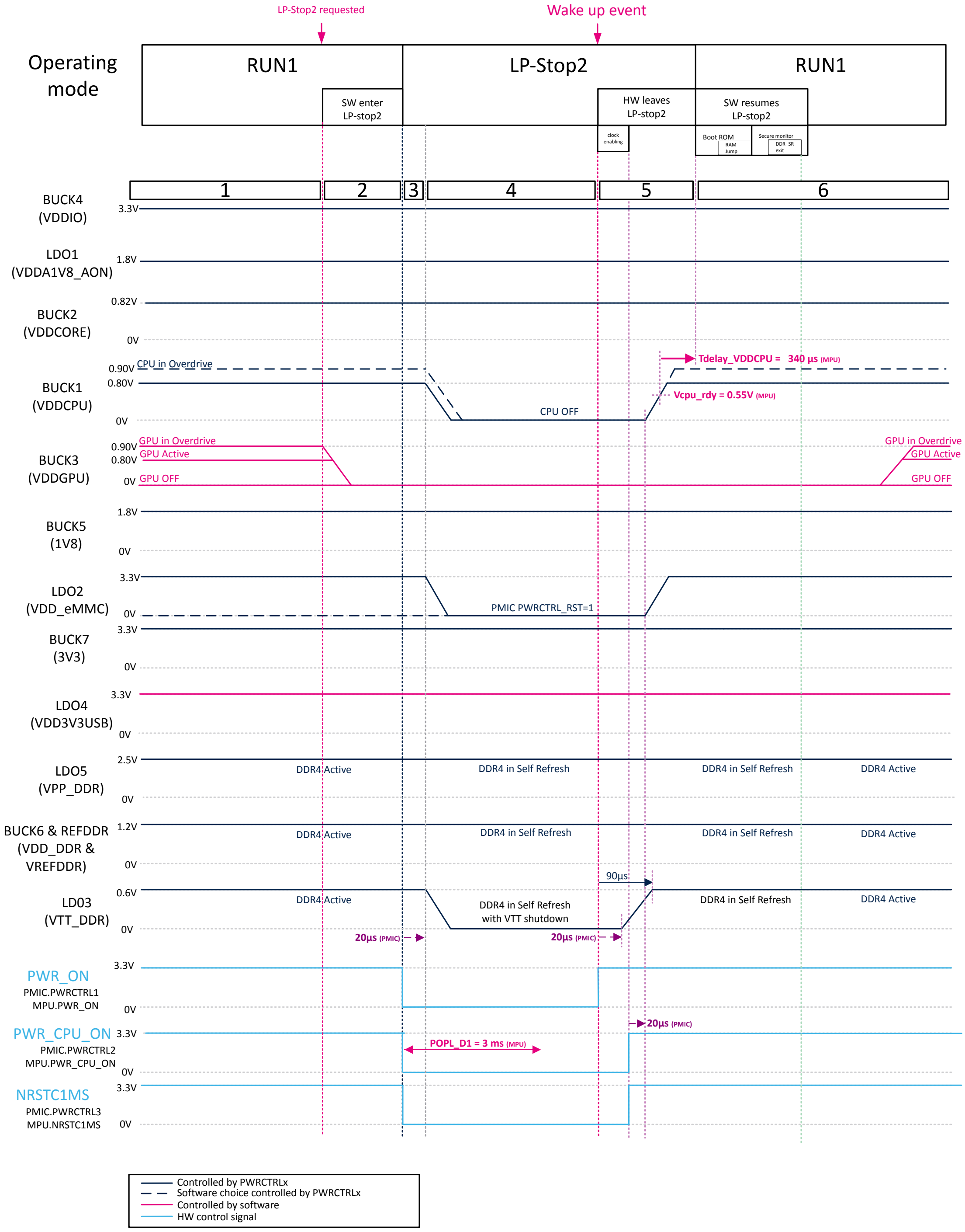
6. The software resumes from LP-Stop2:
   a. A CPU1 reset occurs, then CPU1 reboots from the boot ROM
   b. DDR exits from self-refresh by software running in SYSRAM (secure monitor).

Table 12. **STPMIC25 configuration for LP-Stop2**

Regulator	PWRCTRLx assignation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	VOUT	Configuration	VOUT
BUCK1 (V_{DDCPU})	PWR_CPU_ON	ON HP	0.8	OFF	-
BUCK2 (V_{DDCORE})	PWR_ON	ON HP	0.82	ON HP	0.82
BUCK3 (V_{DDGPU})	-	ON/OFF HP	0.8	N/A	-
BUCK4 (V_{DDIO})	PWR_ON	ON HP	3.3	ON HP	3.3
LDO1 (V_{DDA1V8_AON})	-	ON	N/A	N/A	N/A
BUCK5 (1V8)	PWR_ON	ON HP	1.8	ON HP	1.8
BUCK6 (V_{DD_DDR})	PWR_ON	ON HP	1.2	ON HP	1.2
BUCK7 (3V3)	PWR_ON	ON HP	3.3	ON HP	3.3
V_{REF_DDR}	PWR_ON	ON	N/A	ON	N/A
LDO5 (V_{PP_DDR})	PWR_ON	ON	2.5	ON	2.5
LDO2 (V_{DD_EMMC})	NRSTC1MS (1)	ON	3.3	N/A	0
LDO3 (V_{TT_DDR})	PWR_ON	SINK SOURCE	N/A	OFF	N/A
LDO4 (V_{DD3V3_USB})	-	ON/OFF	N/A	-	N/A

1. *The LDO2 is controlled from PWRCTRL3 in reset mode (STPMIC25 PWRCTRL_RST bit set for LDO2)*

Figure 8. LP-Stop2 sequence



LPLV-Stop2 mode

This section focuses on LPLV-Stop2 mode. The LPLV-Stop2 mode is shown in the Figure 9 based on to the implementation shown in Figure 1.

1. The application is powered up and operates in Run1 mode; all PWR_ON and PWR_CPU_ON are in high state. In this application, the PWR_LP is multiplexed with the PWR_ON.

2. When LPLV-Stop2 mode is requested, the software prepares to enter LPLV-Stop2 mode:
   a. The MPU performs internal settings such as:
      i. Set PWR_D1CR[POPL_D1] = 3 ms timer, to define a minimum pulse duration of PWR_CPU_ON ensuring full discharge of the V_{DDCPU} voltage before restarting.
      ii. Set PWR_D2CR[LPLVDLY_D2] = 187µs timer, to set the LPLVDLY_D2 timer timer used when the MPU leaves the low power mode (see Section 5.3: Low power mode management).
      iii. Disable PWRLP_TEMPO timer (set RCC_PWRLPDLYCR[PWRLP_DLY]]=0).
      iv. Stop the appropriate system clocks.
      v. Set the DDR to self-refresh.
   b. The MPU configures the STPMIC25 as described in Table 13.
   c. The PWR_CPU2CR[LPDS_D2] and PWR_CPU2CR[LVDS_D2] bits are enabled. This allows the regulator to enter into LPLV-Stop2 low power mode and V_{DDCORE} to be lowered to the right value.
   d. The PWR_CPU1CR[PDDS_D1] bit is enabled. This allows the D1 domain to enter in DStandby where V_{DDCPU} switched OFF.
3. Once the system is in LPLV-Stop2:
   a. The MPU PWR_ON is deasserted, and the STPMIC25 PWRCTRL1 signal goes low.
   b. The MPU PWR_CPU_ON is deasserted, and the STPMIC25 PWRCTRL2 signal goes low.
   c. The NRSTC1MS signal follows the PWR_CPU_ON signal, and the STPMIC25 PWRCTRL3 goes low.
   d. The POPL_D1 delay is started to keep the D1 domain powered off until the POPL_D1 timer has elapsed. This means, the wake-up event is shifted until POPL_D1 has elapsed.
4. After 20 µs internal STPMIC25 delay, the STPMIC25 regulators assigned to PWR_ON (linked to the PWRCTRL1) and PWR_CPU_ON (linked to the PWRCTRL2) and NRSTC1MS (linked to the PWRCTRL3) takes the configuration set in the xxx_ALT_CR registers (see Table 13):
   a. V_{TT_DDR} regulators turn OFF.
   b. V_{DDCORE} regulators output decreases to retention voltage.
   c. V_{DDCPU} regulator turns OFF.
   d. V_{DD_EMMC} regulator turns OFF.
5. On a wake-up event, the MPU leaves the LPLV-Stop2:
   a. The MPU PWR_ON output signal is asserted (STPMIC25 PWRCTRL1 goes high) and the MPU executes the LPLVDLY_D2 timer (initialized to 187 µs) to wait for the V_{DDCORE} to switch from retention voltage to nominal voltage.
   b. After a 20 µs internal STPMIC25 delay, the STPMIC25 regulators assigned to the PWR_ON takes the configuration set in the xxx_MAIN_CR registers (see Table 13):
      i. The V_{TT_DDR} regulator turns ON (V_{TT_DDR} voltage rise in 70 µs)
      ii. The V_{DDCORE} regulator switches from retention voltage (670 mV) to nominal voltage (820 mV) in 150 µs.
   c. Once the LPLVDLY_D2 timer timer elapsed (implicitly the V_{DDCORE} voltage has recovered to nominal voltage), clocks are enabled.
   d. After a further 20 µs internal STPMIC25 delay, the STPMIC25 regulators assigned to the PWR_CPU_ON and NRSTC1MS take the configuration set in the xxx_MAIN_CR registers (see Table 13): V_{DD_EMMC} and V_{DDCPU} raise.
   e. Once the V_{DDCPU} voltage reaches the vcpu_rdy threshold, the Tdelay_V_{DDCPU} delay is triggered to wait for the V_{DDCPU} to reach its nominal voltage value.
   f. Once Tdelay_V_{DDCPU} is elapsed, CPU1 releases the clock domain to enter into Run1 mode, the PWRLP_TEMPO timer is not used and set to 0 ms. In this case, the Cortex®-A35 (CPU1) is set as master and the Cortex®-M33 (CPU2) follows the Cortex®-A35

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

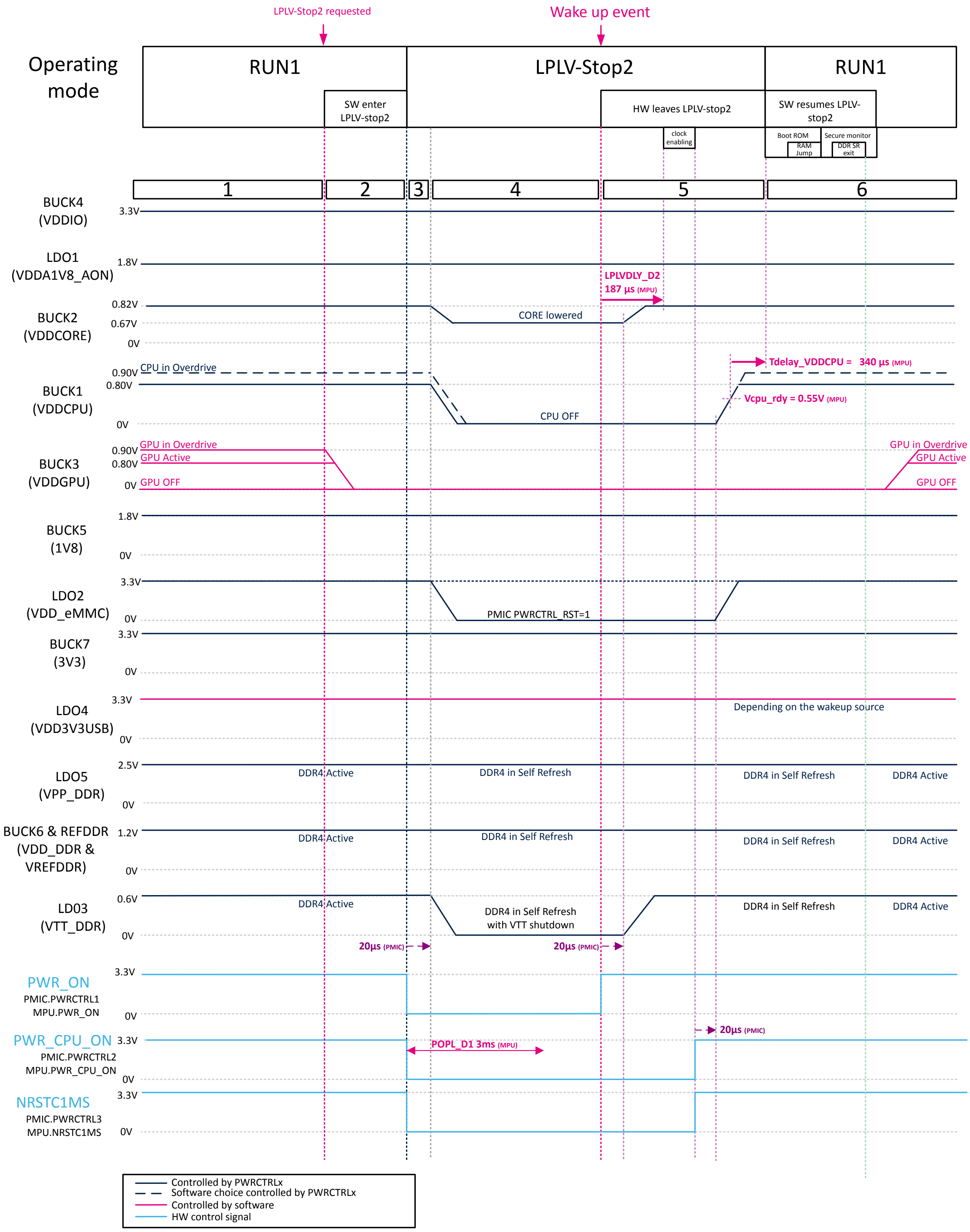
6. The software resumes from LPLV-Stop2:
   a. A CPU1 reset occurs, then CPU1 reboots from the boot ROM, which jumps in software present in SYSRAM.
   b. DDR exits from self-refresh by software running in SYSRAM (secure monitor).

Table 13. STPMIC25 configuration for LPLV-Stop2

Regulator	PWRCTRLx assignation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	VOUT	Configuration	VOUT
BUCK1 (V_{DDCPU})	PWR_CPU_ON	ON HP	0.8	OFF	-
BUCK2 (V_{DDCORE})	PWR_ON	ON HP	0.82	ON HP	0.67
BUCK3 (V_{DDGPU})	-	ON/OFF HP	0.8	-	-
BUCK4 (V_{DDIO})	PWR_ON	ON HP	3.3	ON HP	3.3
LDO1 (V_{DDA1V8_AON})	-	ON	N/A	-	N/A
BUCK5 (1V8)	PWR_ON	ON HP	1.8	ON HP	1.8
BUCK6 (V_{DD_DDR})	PWR_ON	ON HP	1.2	ON HP	1.2
BUCK7 (3V3)	PWR_ON	ON HP	3.3	ON HP	3.3
V_{REF_DDR}	PWR_ON	ON	N/A	ON	N/A
LDO5 (V_{PP_DDR})	PWR_ON	ON	2.5	ON	2.5
LDO2 (V_{DD_EMMC})	NRSTC1MS (1)	ON/OFF	3.3	-	-
LDO3 (V_{TT_DDR})	PWR_ON	SINK SOURCE	N/A	OFF	N/A
LDO4 (V_{DD3V3_USB})	-	ON/OFF	N/A	-	N/A

1. *The LDO2 is controlled from PWRCTRL3 in reset mode (STPMIC25 PWRCTRL_RST bit set for LDO2)*

Figure 9. LPLV-Stop2 sequence



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

5.3.4 Standby mode (DDR4 in self-refresh)

The Standby mode is used when a very-low power consumption is required. In this mode, the MPU PWRCTRLx signals are pulled low. Most of the STPMIC25 regulators are switched off. The content of MPU registers and memories are lost except for the backup and retentions domains (V_{DDIO} and V_{DDA1V8_AON} are kept enabled). The DDR4 is set in self-refresh (V_{DD_DDR}, V_{REF_DDR} and V_{PP_DDR} are kept enabled) to maintain the system in “suspend to RAM state.”

This section focuses on Standby mode with DDR4 in self-refresh. This mode is shown in Figure 10 based on the implementation shown in Figure 1.

1. The application is powered up and operates in Run1 mode. All PWR_ON and PWR_CPU_ON are in high state.
2. When the Standby mode is requested, the software prepares to enter Standby mode:
   a. The MPU performs internal settings such as:
      i. Set the PWR_D1CR[POPL_D1] = 3 ms timer, to define a minimum pulse duration of PWR_CPU_ON ensuring full discharge of the V_{DDCPU} voltage before restarting.
      ii. Set the PWR_D2CR[PODH_D2] = 1 ms timer, to turn off V_{DDCPU} before V_{DDCORE}.
      iii. Set the PWR_D2CR[POPL_D2] = 2 ms, to define a minimum pulse duration of PWR_ON ensuring V_{DDCORE} voltage is fully discharged before restarting.
      iv. Stop the appropriate clocks.
      v. Set the DDR to self-refresh.
   b. The MPU configures the STPMIC25 as described in Table 14.
   c. The PWR_CPU2CR[PDDS_D2] bit is enabled. Standby mode is allowed when CPU2 goes OFF.
   d. The PWR_CPU1CR[PDDS_D1] and PWR_CPU1CR[PDDS_D2] bit are enabled. This allows the D1 domain to enter in DStandby (V_{DDCPU} switched OFF) and the Standby mode is allowed when CPU1 goes OFF.
3. Once the system is in Standby:
   a. The MPU PWR_CPU_ON is deasserted. The STPMIC25 PWRCTRL2 goes low.
   b. The NRSTC1MS signal follows the PWR_CPU_ON signal (STPMIC25 PWRCTRL3 goes low)
   c. The POPL_D1 timer is started to keep the D1 domain powered off until the POPL_D1 timer has elapsed. The wake-up event is shifted until POPL_D1 has elapsed.
   d. The PODH_D2 timer timer is started to keep V_{DDCORE} enabled (PWR_ON keeps high) waiting for the V_{DDCPU} voltage to be powered off before V_{DDCORE}.
4. Following a 20 µs internal STPMIC25 delay, the STPMIC25 regulators assigned to PWR_CPU_ON (linked to PWRCTRL2) and NRSTC1MS (linked to PWRCTRL3) take the configuration set in the xxx_ALT_CR registers (see Table 14):
   a. V_{DDCPU} regulator turns OFF
   b. V_{DD_EMMC} regulator turns OFF
5. Once PODH_D2 timer is elapsed:
   a. The MPU PWR _ON is deasserted. The STPMIC25 PWRCTRL1 signal goes low.
   b. The POPL_D2 delay is started to keep the D2 domain powered off until POPL_D2 has elapsed. The wake-up event is shifted until POPL_D2 has elapsed.
6. Following a 20 µs internal STPMIC25 delay, the STPMIC25 regulators assigned to PWR_ON (linked to PWRCTRL1) take the configuration set in the xxx_ALT_CR registers (see Table 14):
   a. V_{DDCORE} regulator is turned OFF
   b. 1V8 regulator is turned OFF
   c. 3V3 regulator is turned OFF
   d. V_{TT_DDR} regulator is turned OFF

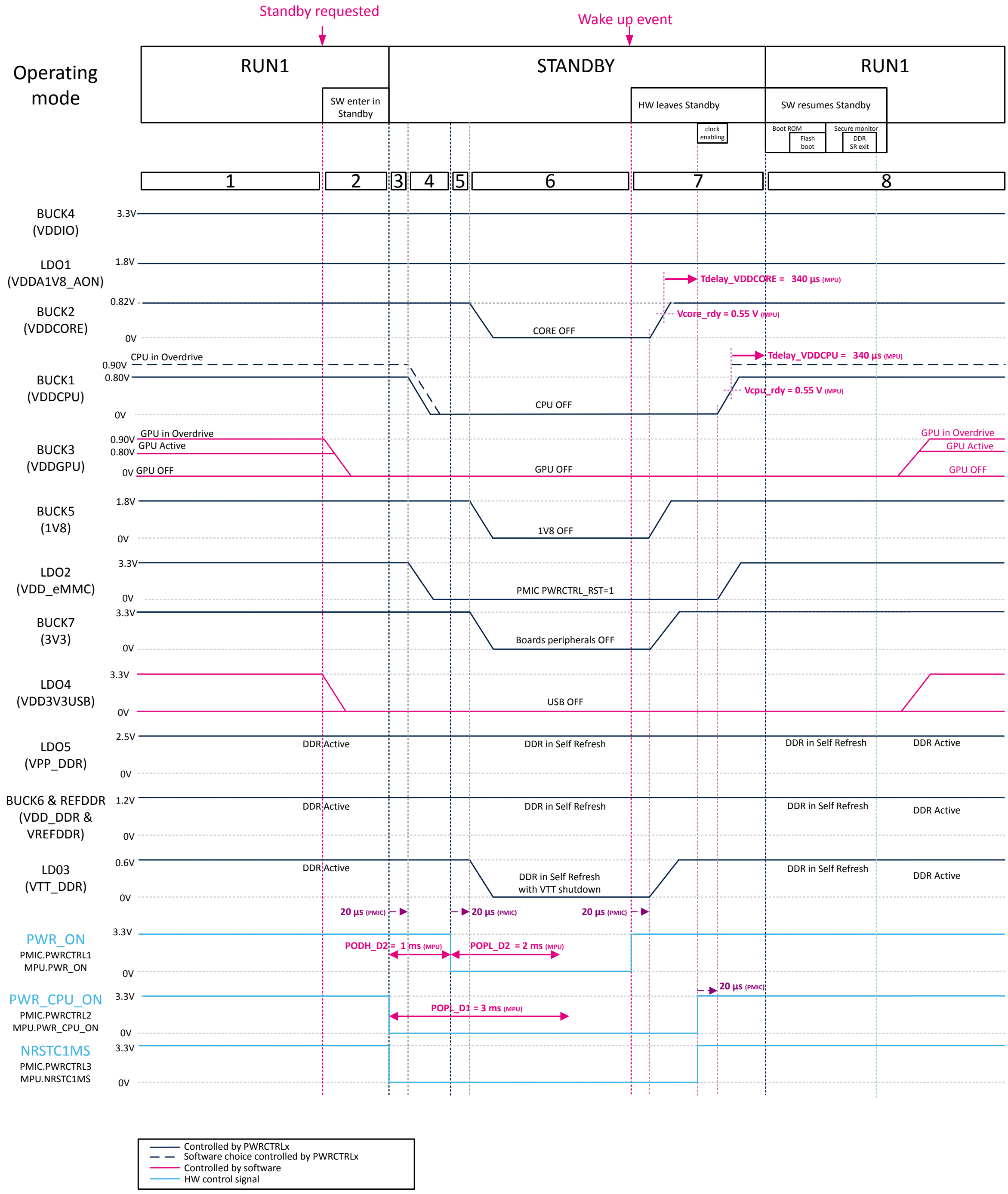
7. On a wake-up event, the MPU leaves the Standby mode:
   a. The MPU PWR_ON signal is asserted (STPMIC25 PWRCTRL1 goes high).
   b. Following the 20 μs internal STPMIC25 delay, the STPMIC25 regulators assigned to PWR_ON take the configuration set in the xxx_MAIN_CR registers (see Table 14):
      i. V_{TT_DDR} regulator is turned ON.
      ii. 3V3 regulators are turned ON.
      iii. 1V8 regulators are turned ON.
      iv. V_{DDCORE} regulator is turned ON.
   c. Once the V_{DDCORE} voltage reaches the vcore_rdy threshold, the Tdelay_V_{DDCORE} internal delay is started. This ensures V_{DDCORE} reaches its nominal voltage value.
   d. Once the Tdelay_V_{DDCORE} delay is elapsed, the clocks are enabled and the MPU PWR_CPU_ON is asserted.
   e. The NRSTC1MS signal follows PWR_CPU_ON.
   f. Following the 20 μs internal STPMIC25 delay, the STPMIC25 regulators assigned to PWR_CPU_ON and NRSTC1MS take the configuration set in the xxx_MAIN_CR registers (see Table 14):
      i. V_{DD_EMMC} and V_{DDCPU} rise.
   g. Once the V_{DDCPU} voltage reaches the vcpu_rdy threshold, the Tdelay_V_{DDCPU} internal delay is started. This is to ensure that V_{DDCPU} reaches its nominal voltage value.
   h. Once Tdelay_V_{DDCPU} is elapsed, CPU1 releases the clock domain to enter in Run1 mode.
   i. In this case, the Cortex®-A35 (CPU1) is set as master and the Cortex®-M33 (CPU2) follows the Cortex®-A35, the system enters in Run1 once the CPU1 clocks are released. The Cortex®-M33 is running once the DDR4 exits from self-refresh.
8. The software resumes from Standby:
   a. The boot ROM is executed then CPU1 jumps to execute software in SYSRAM to resume from Standby (the timer EADLY timer is skipped as no external access to flash memory).
   b. DDR exits from self-refresh by software running is SYSRAM(secure monitor).

Table 14. STM32MP25x configuration for standby DDR in self-refresh

Regulator	PWRCTRLx assignation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	VOUT	Configuration	VOUT
BUCK1 (V_{DDCPU})	PWR_CPU_ON	ON HP	0.8	OFF	-
BUCK2 (V_{DDCORE})	PWR_ON	ON HP	0.82	OFF	-
BUCK3 (V_{DDGPU})	-	ON/OFF HP	0.8	-	-
BUCK4 (V_{DDIO})	PWR_ON	ON HP	3.3	ON HP	3.3
LDO1 (V_{DDA1V8_AON})	-	ON	N/A	-	N/A
BUCK5 (1V8)	PWR_ON	ON HP	1.8	OFF	-
BUCK6 (V_{DD_DDR})	PWR_ON	ON HP	1.2	ON HP	1.2
BUCK7 (3V3)	PWR_ON	ON HP	3.3	OFF	-
REFDDR	PWR_ON	ON	N/A	ON	N/A
LDO5 (V_{PP_DDR})	PWR_ON	ON	2.5	ON	2.5
LDO2 (V_{DD_EMMC})	NRSTC1MS [1]	ON	3.3	-	-
LDO3 (V_{TT_DDR})	PWR_ON	SINK SOURCE	N/A	OFF	N/A
LDO4 (V_{DD3V3_USB}[2])	-	ON/OFF	N/A	-	N/A

1. *The LDO2 is controlled from PWRCTRL3 in reset mode (STM32MP25x PWRCTRL_RST bit set for LDO2)*
2. *V_{DD3V3_USB} must be turned OFF before V_{DDCORE} is turned OFF*

Figure 10. Standby (DDR in self-refresh) sequence



5.3.5 Standby mode (DDR4 OFF)

The Standby mode is used when a very-low power consumption is required. In this mode, the STPMIC25 PWRCTRLx signals are set low. Most of the STPMIC25 regulators are switched off. The content of MPU registers and memories are lost except for the backup and the retention domain (V_{DDIO} and V_{DDA1V8_AON} are kept enabled). The DDR4 is powered off to maintain the system in “suspend to flash state.”

On the STPMIC25 side, the PWRCTRL1 is used to switch the state machine from RUN to STANDBY state (see details in [2]) in addition to switch OFF regulators linked to D2 domain (PWR_ON).

This section focuses on the Standby mode with DDR4 OFF. This mode is described below and is shown in Figure 11 based on the implementation shown in Figure 1.

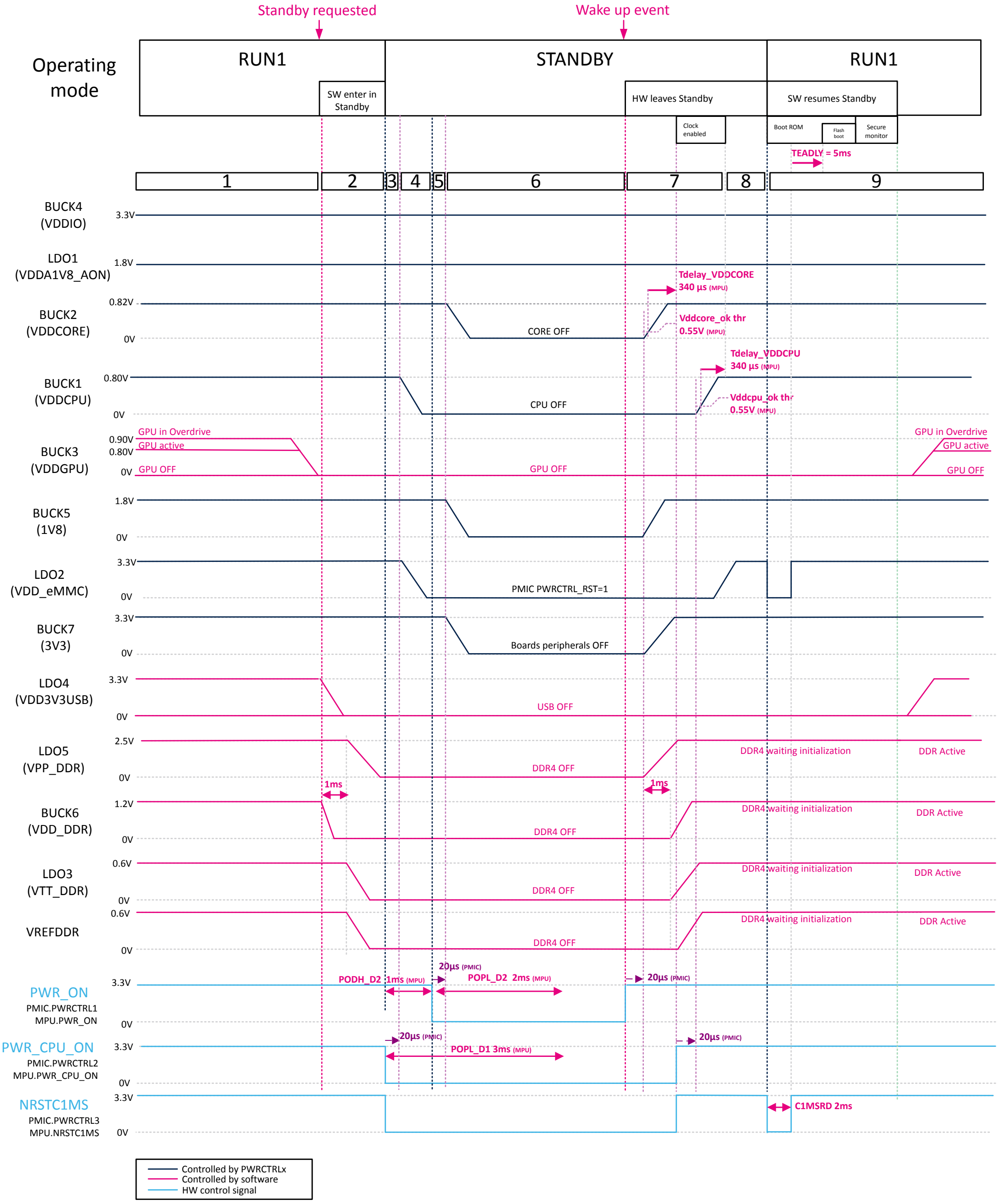
1. The application is powered up and operating in Run1 mode. All PWR_ON and PWR_CPU_ON are in high state.
2. When the Standby mode is requested, the software prepares to enter Standby mode:
   a. The MPU performs internal settings such as:
      i. Set the PWR_D1CR[POPL_D1] = 3 ms timer, to define a minimum pulse duration of PWR_CPU_ON ensuring full discharge of the V_{DDCPU} voltage before restarting.
      ii. Set the PWR_D2CR[PODH_D2] =1 ms timer, to turn off V_{DDCPU} before V_{DDCORE}.
      iii. Set the PWR_D2CR[POPL_D2] = 2 ms timer, to define a minimum pulse duration of PWR_ON ensuring full discharge of the V_{DDCORE} voltage before restarting.
      iv. Stop the appropriate clocks.
      v. Disable DDR.
      vi. Turns OFF the DDR regulators according to the power down sequence defined in Section 4.1.6: DDR power domain (V_{DD_DDR}, V_{TT_DDR}, V_{PP_DDR}, V_{REF_DDR}).
   b. The MPU configures the STPMIC25 as described in Table 15.
   c. The PWR_CPU2CR[PDDS_D2] bit is enabled. Standby mode is allowed when CPU2 goes OFF.
   d. The PWR_CPU1CR[PDDS_D1] and PWR_CPU1CR[PDDS_D2] bit are enabled. This allows the D1 domain to enter in DStandby (V_{DDCPU} switched OFF) and the Standby mode is allowed when CPU1 goes OFF.
3. Once the system is in Standby
   a. The MPU PWR_CPU_ON is de-asserted, the STPMIC25 PWRCTRL2 goes low.
   b. The NRSTC1MS signal follows the PWR_CPU_ON signal (STPMIC25 PWRCTRL3 goes low)
   c. The POPL_D1 timer is started to keep the D1 domain powered off until POPL_D1 timer has elapsed (wakeup event is shifted until POPL_D1 has elapsed).
   d. The PODH_D2 timer is started to keep V_{DDCORE} enabled (PWR_ON keeps high) waiting for V_{DDCPU} voltage to be powered off before V_{DDCORE}.
4. After 20 µs internal STPMIC25 delay, the STPMIC25 regulators assigned to PWR_CPU_ON (linked to PWRCTRL2) and NRSTC1MS (linked to PWRCTRL3) takes the configuration set in the xxx_ALT_CR registers (see Table 15).
   a. V_{DDCPU} regulator turns OFF
   b. V_{DD_EMMC} regulator is turned OFF
5. Once the PODH_D2 timer timer has elapsed
   a. The MPU PWR_ON is deasserted (STPMIC25 PWRCTRL1 goes low)
   b. The POPL_D2 delay is started to keep the D2 domain powered off until the POPL_D2 timer has elapsed. The wake-up event is shifted until POPL_D2 has elapsed.
   c. The POPL_D2 delay is applied on PWRCTRL1.
6. After 20 µs interval STPMIC25 delay, the STPMIC25 regulators assigned to PWR_ON (linked to PWRCTRL1) takes the configuration set in the xxx_ALT_CR registers (see Table 15).
   a. V_{DDCORE} regulator turns OFF
   b. 1V8 regulator turns OFF
   c. 3V3 regulators turn OFF
   d. V_{TT_DDR} regulator turns OFF

7. On a wake-up event, the MPU leaves the low power mode:
   a. The MPU PWR_ON signal is asserted (STPMIC25 PWRCTRL1 goes high).
   b. After 20 µs internal STPMIC25 delay, the STPMIC25 regulators assigned to PWR_ON takes the configuration set in the xxx_MAIN_CR registers (see Table 15):
      - V_{TT_DDR} regulator turns ON.
      - 3V3 regulator turns ON.
      - 1V8 regulator turns ON.
      - V_{DDCORE} regulator turns ON.
      - DDR regulators (V_{PP_DDR}, V_{TT_DDR}, and V_{REF_DDR}) waiting for software initialization.
   c. Once the V_{DDCORE} voltage reaches the vcore_rdy threshold, the Tdelay_V_{DDCORE} internal delay starts to allow the V_{DDCORE} to reach its nominal voltage value.
   d. Once the Tdelay_V_{DDCORE} delay is elapsed, the clocks are enabled and the MPU PWR_CPU_ON is asserted. The NRSTC1MS signal follows PWR_CPU_ON
8. After 20 µs internal STPMIC25 delay, the STPMIC25 regulators assigned to PWR_CPU_ON and NRSTC1MS takes the configuration set in the xxx_MAIN_CR registers (see Table 15): V_{DD_EMMC} and V_{DDCPU} rise.
   a. Once the V_{DDCPU} voltage reaches the vcpu_rdy threshold, the Tdelay_V_{DDCPU} internal delay starts to allow the V_{DDCPU} to reach its nominal voltage value.
   b. Once Tdelay_V_{DDCPU} is elapsed, CPU1 releases the clock domain to enter in Run1 mode.
      In this case, the Cortex®-A35 (CPU1) is set as master and the Cortex®-M33 (CPU2) follows the Cortex®-A35, the system enters in Run1 once the CPU1 clocks are released. The Cortex®-M33 is running once the DDR4 exits from self-refresh.
9. The software resumes from Standby:
   a. A system reset is generated during the C1MSRD timer.
   b. Once this delay is elapsed, the boot ROM must perform any external access during T_{EADLY} = 5 ms. Then the CPU1 starts the boot ROM execution. The application software is running from the flash boot.
   c. Once the boot ROM execution is done, the bootloader carries out a DDR initialization.

Table 15. STPMIC25 configuration standby DDR OFF

Regulator	PWRCTRLx assignation	Register xxxx_MAIN_CR		Register xxxx_ALT_CR	
		Configuration	VOUT	Configuration	VOUT
BUCK1 (V_{DDCPU})	PWR_CPU_ON	ON HP	0.8	OFF	-
BUCK2 (V_{DDCORE})	PWR_ON	ON HP	0.82	OFF	-
BUCK3 (V_{DDGPU})	-	ON/OFF HP	0.8	-	-
BUCK4 (V_{DDIO})	PWR_ON	ON HP	3.3	ON LP	3.3
LDO1 (V_{DDA1V8_AON})	-	ON	N/A	-	N/A
BUCK5 (1V8)	PWR_ON	ON HP	1.8	OFF	-
BUCK6 (V_{DD_DDR})	PWR_ON	ON HP	1.2	OFF	-
BUCK7 (3V3)	PWR_ON	ON HP	3.3	OFF	-
VREF_DDR	PWR_ON	ON	N/A	OFF	-
LDO5 (V_{PP_DDR})	PWR_ON	ON	2.5	OFF	-
LDO2 (V_{DD_EMMC})	NRSTC1MS	ON/OFF	3.3	-	-
LDO3 `(V_{TT_DDR})	PWR_ON	SINK SOURCE	N/A	OFF	N/A
LDO4 (V_{DD3V3_USB})	-	ON	N/A	-	N/A

Figure 11. Standby (DDR OFF) sequence



5.4 System and CPU1 crash recovery management

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

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

5.4.1 System crash recovery management

As introduced in the RSTn pin section, the MPU, and the STPMIC25 both have interconnected bidirectional low reset pins (see Figure 1 NRST signal).

An STM32MP25x crash occurs when one or several of the following fail:

- D1, D2, or D3 domain crash through watchdogs elapsing:
  - iwdg1_out_rst,
  - iwdg2_out_rst
  - iwdg3_out_rst
  - iwdg4_out_rst
  - iwdg5_out_rst
- System reset
- Assertion of NRST by an external source

If an STM32MP25x crash occurs, the MPU generates a reset pulse on the NRST signal.

The reset pulse triggers the STPMIC25 to produce an immediate power cycle sequence. This power cycling is recommended to ensure a correct reset and restart of the peripherals following a global application reset (NRST). This is specifically for peripherals that do not have a reset input signal such as eMMC and so on.

In this application, the power cycling is not performed on the STPMIC25 BUCK4 (V_{DDIO}), neither on the LDO1 (V_{DDA1V8_AON}) due to mask reset option. With this option, the regulators need to be kept enable during reset (see Section 5.1.1: Application turn-on/turn-off conditions for details).

5.4.2 CPU1 crash recovery management

As introduced in the PWRCTRL1, PWRCTRL2, PWRCTRL3 section, the NRSTC1MS pin is dedicated to reset peripherals associated with the CPU1.

If a crash occurs on CPU1, only this domain is reset. Then the boot ROM generates a pulse on NRSTC1MS to reset external peripherals such as boot mass storage memory.

CPU1 reset can reset the complete application but auto reset only the D1 domain (embedding CPU1), whereas the D2 (embedding CPU2) domain is kept alive.

If an SD card is used in the application, power on the SD card in Run mode before the application goes into Standby mode. The V_{DD_SDCARD} and V_{DDIO_SDCARD} are required for the first level boot of CPU1 and need a power cycle to ensure that the platform reboots. An MPU PWRCTRL I/O can be used for this operation.

5.4.3 System crash recovery management sequence

The sequence in Figure 12 illustrates a crash recovery sequence according to the application shown in Figure 1. In this sequence, the crash happens in Run mode using an IWDG reset. An IWDG reset could occur in all low power modes.

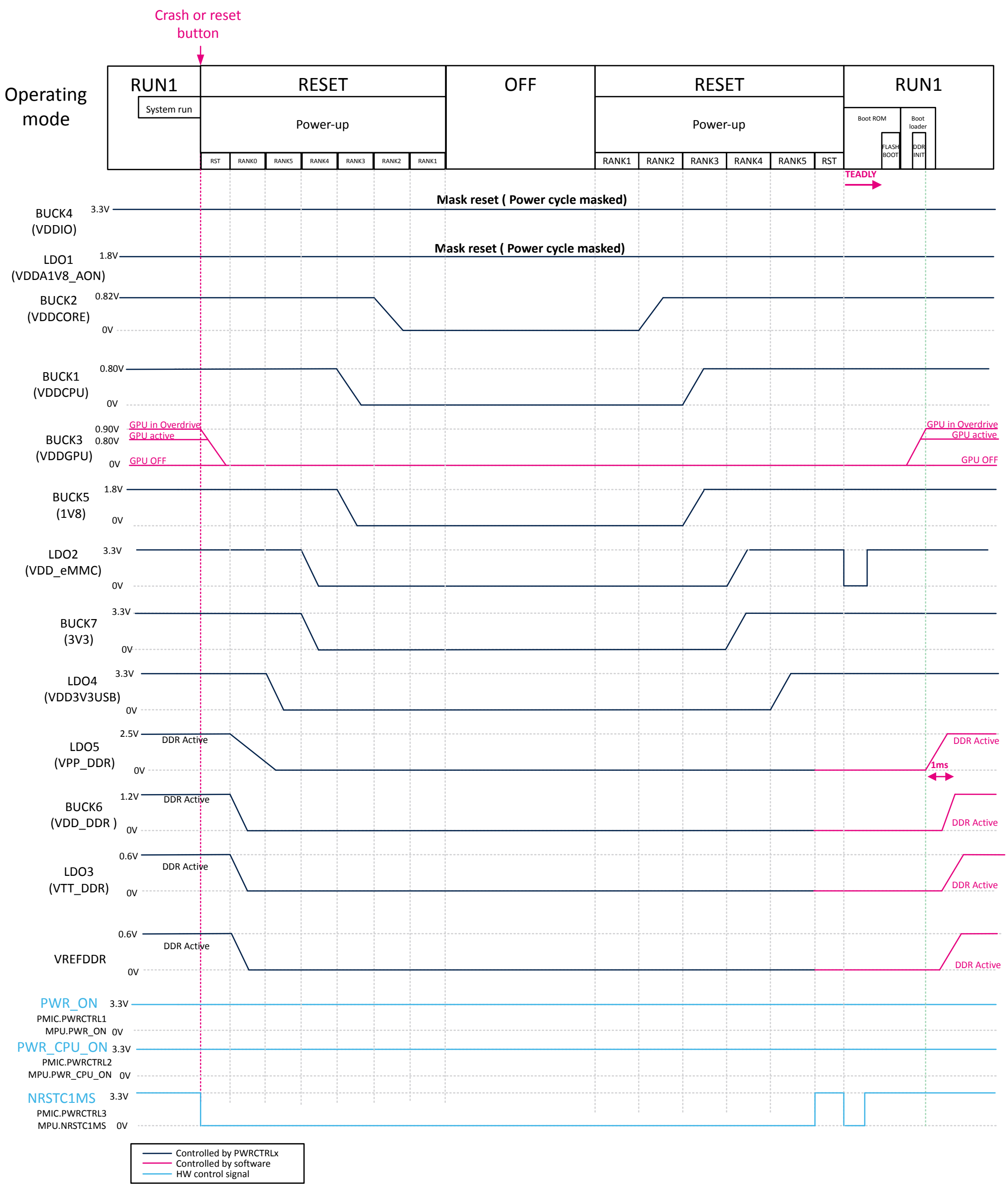
1. The application is powered up and is in Run mode. A crash occurs (watchdogs elapsing), or the user presses the reset button generating a pulse on the NRST signal.
2. The STPMIC25APQR detects the reset assertion (NRST pulse low) and starts a noninterruptible power cycle:
   a. The STPMIC25APQR asserts an NRST low.
3. The STPMIC25APQR performs a powerdown sequence.
4. The STPMIC25APQR checks the condition to restart and reloads the internal NVM.
5. The STPMIC25APQR performs a power-up sequence.

6. The STPMIC25APQR releases NRST.
   If the NRST (reset signal) is still asserted at this stage such as the user is still pressing the reset button, the STPMIC25APQR waits for the reset signal to be released before rearming the reset circuit. This is to avoid the STPMIC25APQR repeating a power cycle loop.

Note: STPMIC25APQR power cycle duration depends on RANK_DLY and RST_DLY configuration (see [3] for details)

7. The NRST signal rises as the STM32MP25x and the STPMIC25APQR release their respective reset pins (and reset button releases):
   a. The STM32MP25x EADLY timer is started.
8. When EADLY timer elapses, the boot ROM starts accessing external peripherals. This can be the eMMC or SD card depending on the STM32MP25x boot settings, to load and execute bootloader software.
9. The bootloader initializes the DDR then loads and executes the kernel: see Section 4.1.6: DDR power domain (V_{DD_DDR}, V_{TT_DDR}, V_{PP_DDR}, V_{REF_DDR}) for more details.

Figure 12. Crash recovery sequence



6 Safety management

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

In this application, the safety management is provided by the MPU software or/and by the STPMIC25 functionalities. In this section, the focus is on the STPMIC25 safety management functionalities.

6.1 STPMIC25 fail-safe management

The STPMIC25 integrates safety elements to manage the following hard fault conditions:

- Over current protection
- Watchdogs
- Thermal shutdown
- V_{INOK_fall}: this is when V_{IN} supply (main) falls below the V_{INOK_fall} threshold.

The general mechanism is as follows:

Each source of hard fault has a dedicated independent fail-safe counter. This counter, named xxx_FLT_CNT, is incremented each time a turn-off hard fault condition occurs.

The counter maximum fault iteration, xxxx_FLT_MAX set by the NVM, is used to define the maximum number of hard fault iterations before the STPMIC25 enters in the FAIL_SAFE_LOCK state. By default, the maximal counter for each hard fault is infinite. The STPMIC25 always restarts when a hard fault condition occurs.

The reset fault timer, RST_FLT_CNT_TMR (set by NVM), is used to define the minimum wait time before reinitializing the hard fault fail-safe counters. If a hard fault condition occurs during this time, the timer is stopped.

As long as the fail-safe counter xxx_FLT_CNT is below xxxx_FLT_MAX, the STPMIC25 carries out a power cycle when a hard fault occurs. A power cycle is defined as a powerdown sequence then wait for FLT_TMR (fault timer) then power up.

Once the number of hard fault iterations exceeds the xxxx_FLT_MAX counter, the system is blocked in FAIL_SAFE_LOCK state. To exit this state, the STPMIC25 must carry out a power-off and power-on or a PONKEYn long press using the rest button can awaken the application (a special NVM setting is necessary).

The following examples illustrate reset mechanisms.

Example 1

When a V_{INOK_fall} hard fault condition occurs in the application.

The STPMIC25 NVM was preconfigurated with the following parameters:

- The VIN_FLT_CNT_MAX counter is configured (NVM fail-safe shadow register NVM_FS_SHR1) at 0001 (one hard fault allowed)
- The RST_FLT_CNT_TMR counter is configured (NVM fail-safe shadow register NVM_FS_SHR2) to 10 (6 min)

The application is powered by a 5 V supply by an external power source. At a set moment, the application supply falls below V_{INOK_fall} (3.5 V). When this condition occurs, a hard fault is triggered by the STPMIC25. The dedicated fail-safe counter (VIN_FLT_CNT) is incremented. The STPMIC25 carries out a power cycle.

- If another V_{INOK_fall} hard fault conditions occur before RST_FLT_CNT_TMR timing elapsed, the VIN_FLT_CNT is incremented one more time and reaches the VIN_FLT_CNT_MAX value, the STPMIC25 enters in FAIL_SAKE_LOCK state.
- If another V_{INOK_fall} hard fault conditions occur after the RST_FLT_CNT_TMR timing elapses, the VIN_FLT_CNT is reinitialized and the system restarts.

Example 2

When a hard fault occurs, the system must go in the OFF state and be woken by the PONKEYn.

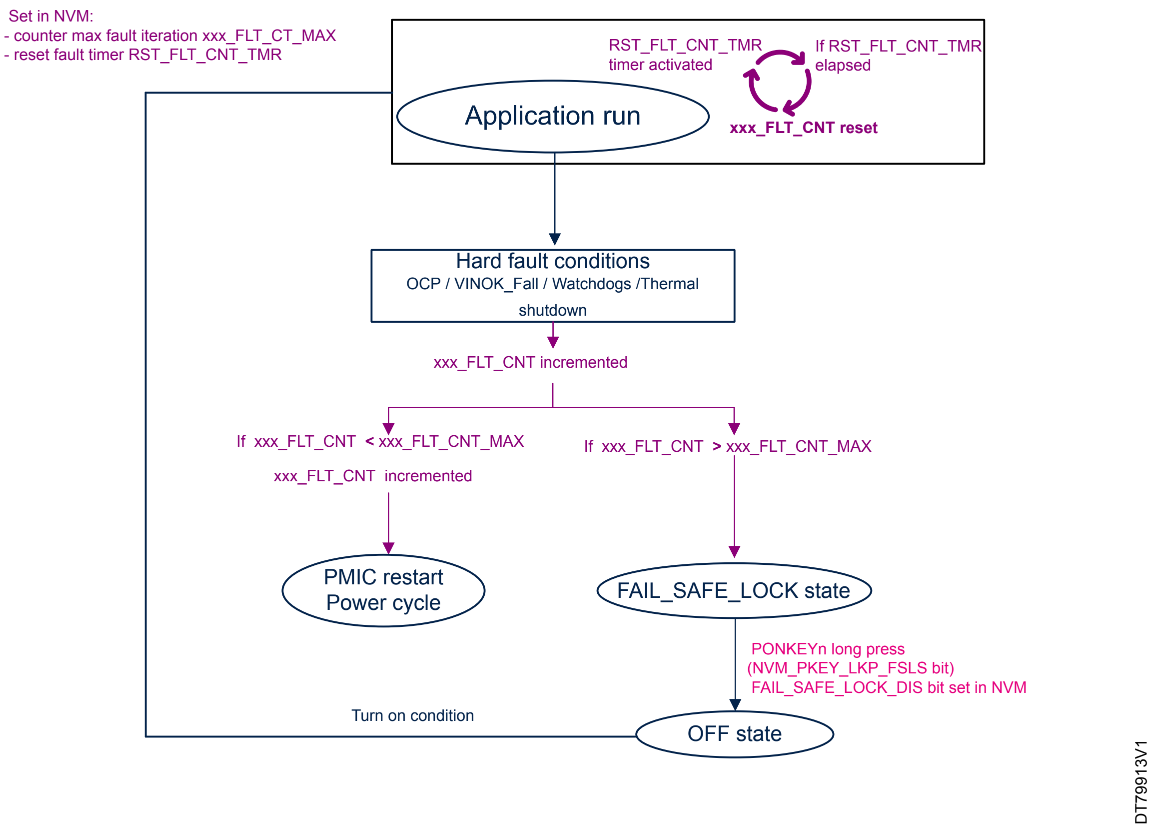
This configuration is defined as follows:

- The xxx_FLT_CNT_MAX must be configured to accept no hard fault (zero hard fault allowed)
- The FAIL_SAFE_LOCK_DIS is disabled (register FAIL_SAFE_LOCK_DIS = 1).

If a hard fault occurs, the xxx_FLT_CNT is incremented and is now greater than xxx_FLT_CNT_MAX. The system enters into the FAIL_SAFE_LOCK state but the FAIL_SAFE_LOCK_DIS setting has disabled the FAIL_SAFE_LOCK state. The system enters directly to the OFF state. The PONKEY button must be pressed to turn the system on to restart it.

The fail-safe management mechanism is summarized in this diagram:

Figure 13. Mechanism fail-safe management



6.1.1 Overcurrent protection (OCP)

To manage overcurrent protection, two levels of protection are implemented:

Hiccup

This protection mode can be set independently for each STPMIC25 regulator, which is not application critical.

Once an OCP occurs, the regulator turns off during the dedicated fault timer - tHICCUP_DLY set in NVM -and then restarts. The fault timer (t_{HICCUP_DLY}) can be set to 0, this means that the regulators restart immediately when an OCP occurs.

With this mode of OCP management, the regulator can restart infinitely until the error disappears.

For example: BUCK7 is used to supply external peripherals, an auto reset of this regulator when an OCP occurs is not critical for the application. BUCK7 (3V3) can be managed in hiccup mode.

OCP fail-safe management

This protection mode is set on application critical regulators such as: DDR4 regulators, V_{DDCORE}, V_{DDCPU}, and so on.

When an OCP occurs on one of the STPMIC25 regulators, this regulator triggers a hard fault condition. When this hard fault is generated, the STPMIC25 triggers a power cycle or enters in FAIL_SAFE_LOCK state

For example: When an OCP occurs on BUCK2 (V_{DDCORE}), the complete application is turned off. The error disappear or was deleted, then a turn-on is possible and the application restarts properly. BUCK2 (V_{DDCORE}) is managed in OCP fail-safe management.

6.1.2 Watchdogs

The STPMIC25 embeds an internal watchdog enabled at runtime by software, or by NVM settings, which are enabled at power-up:

- The watchdog timer duration WDG_TMR_SET is set.
- The watchdog is enabled (WDG_EN bit).

At runtime, the software must periodically reinitialize the watchdog timer by setting the watchdog reset (WDG_RST bit).

A dedicated STPMIC25 PWRCTRLx can be defined to suspend the watchdog typically in low power mode. When this PWRCTRL is active in low power mode, the watchdog timer is suspended. When this PWRCTRL is inactive the watchdog timer is resumed

When the software fails to reinitialize the watchdog following a software crash, the watchdog timer counter (WDG_TMR_CNT) is set to 0.

Once the watchdog timer counter reaches 0, it generates a hard fault condition. Following these hard fault conditions, the dedicated fail-safe counter WDG_FLT_CNT is incremented.

If the WDG_FLT_CNT counter is lower than the WDG_FLT_CNT_MAX, the STPMIC25 carries out a power cycle.

Once the WDG_FLT_CNT counter is higher than the WDG_FLT_CNT_MAX, the STPMIC25 enters into FAIL_SAFE_LOCK state.

6.1.3 Thermal protection

The STPMIC25 embeds a thermal protection to avoid overheating damage.

Two levels of thermal protection are available:

- First level, an interruption is sent to the MPU:
  Once the STPMIC25 junction temperature is below or above defined thresholds (TWRN_Fall or TWRN_Rise), the STPMIC25 generates interrupts that are sent to the MPU.
- A second level, turn off hard fault conditions is generated:
  When the STPMIC25 junction temperature is below TSHDN_Rise thresholds a hard fault condition is generated, the thermal fail-safe counter (TSHDN_FLT_CNT) is incremented. Once the TSHDN _FLT_CNT counter is higher than the TSHDN _FLT_CNT_MAX, the STPMIC25 enters in FAIL_SAFE_LOCK state. But if the STPMIC25 junction temperature is lower than the TSHDN_Fall threshold and the dedicated fault timer (tSHDN_DLY=3 s) is ended, the STPMIC25 restarts.

6.1.4 Main supply (V_{IN}) falls below V_{INOK_Fall} thresholds

The STPMIC25 embeds an internal protection to protect the system if the main supply falls below the V_{INOK_fall} threshold.

When the main supply is lower than V_{INOK_fall} thresholds and the dedicated fault timer (t_{VINOK_Fall} = 100 ms) elapses, a hard fault conditions is generated, the V_{IN} fail-safe counter (VIN_FLT_CNT) is incremented.

Once the VIN_FLT_CNT counter is higher than the VIN_FLT_CNT_MAX, the STPMIC25 enters in FAIL_SAFE_LOCK state.

6.2 OCP management in the application

In the application, the two levels of protection are applied on the regulators as follow:
All regulators critical for the application are managed in OCP fail safe.

Table 16. OCP management application

Regulator	HICCUP	Fail Safe
BUCK1 (V_{DDCPU})	-	YES
BUCK2 (V_{DDCORE})	-	YES
BUCK3 (V_{DDGPU})	YES	-
BUCK4 (V_{DDIO})	-	YES
BUCK5 (1V8)	-	YES
BUCK6 (V_{DD_DDR})	-	YES
BUCK7 (3V3)	YES	-
V_{REF_DDR}	-	YES
LDO1 (V_{DDA1V8_AON})	-	YES
LDO5 (V_{PP_DDR})	-	YES
LDO2 (V_{DD_EMMC})	YES	-
LDO3 (V_{TT_DDR})	-	YES
LDO4 (V_{DD3V3_USB})	YES	-

Revision history

Table 17. **Document revision history**

Date	Version	Changes
24-Sep-2024	1	Initial release.

Contents

List of figures

List of tables