

Integration of STM32MP13x MPU product lines and STPMIC1D / STPMIC1A on a wall adapter supply

Introduction

This application note applies to the STM32MP13x product line devices, henceforward referred to as STM32MP13x to ease the reading of this document. The power management integrated circuits used to illustrate this application are the STPMIC1DPQR and is referred to as STPMIC1D and the STPMIC1APQR referred to as STPMIC1A.

This application note provides a hardware reference design example based on a STM32MP13x device powered from a main power supply via the STPMIC1D or STPMIC1A power management IC.

This document is intended for product architects and designers who require information about hardware integration and settings, and it focuses on:

- Reference design block diagram
- Power distribution
- Start up, shutdown and low-power management
- Power on/off and low-power management
- USB high-speed port management
- Enhanced CPU frequency supply management

Table 1. Device summary

Reference	Product lines
STM32MP13x	STM32MP131 ,STM32MP133, STM32MP135

1 General information

This document applies to STM32MP13x dual-core Arm[®]-based microprocessors.

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



2 Overview

This application note covers the STM32MP13x together with the STPMIC1D or STPMIC1A (referred to as PMIC in the rest of this document when referring to one or the other indistinctly) with DDR and flash memory and together with the following peripherals:

- DC input power source from main power supply: 5 V typical (4 V to 5.5 V).
- DDR3/DDR3L with x16 bits bus width.
- A boot device that can be either a 3.3 V or 1.8 V powered eMMC or a 3.3 V powered NAND or NOR.
- 2 x USB HS host port.

This application note also covers a low-cost implementation using STPMIC1A with V_{DDCPU} and V_{DDCORE} powered from a single PMIC SMPS, allowing one free SMPS to be allocated to power supply some peripherals of the application.

Not covered by this application note:

- IpDDR2 and IpDDR3
- Peripherals with an I/O voltage of 1.8 V.

It is assumed that the above design choices are more relevant for battery-powered application (using STPMIC1E).

- USB Type-C receptacle with dual role.

2.1 Reference documents

Table 2. Reference documents

Document number	Title
STMicroelectronics documents.⁽¹⁾	
[1]	<i>Getting started with STM32MP13x Lines hardware development (AN5474)</i>
[2]	<i>Highly integrated power management IC for micro processor units (DS12792)</i>
[3]	<i>STM32MP13x Lines using low-power modes (AN5565)</i>
[4]	<i>STM32MP131, STM32MP133 and STM32MP135 MPU Lines discrete power supply hardware integration (AN5586)</i>
[5]	<i>STM32MP13xx advanced Arm®-based 32-bit MPUs (RM0475)</i>
[6]	<i>Arm® Cortex®-A7 up to 900 MHz, TFT, 30 comm. interfaces, 24 timers, analog, cryptography and advanced security (DS13483)</i>
[7]	<i>STM32MP1 Series lifetime estimates (AN5438)</i>

1. Refer to www.st.com

3 Glossary

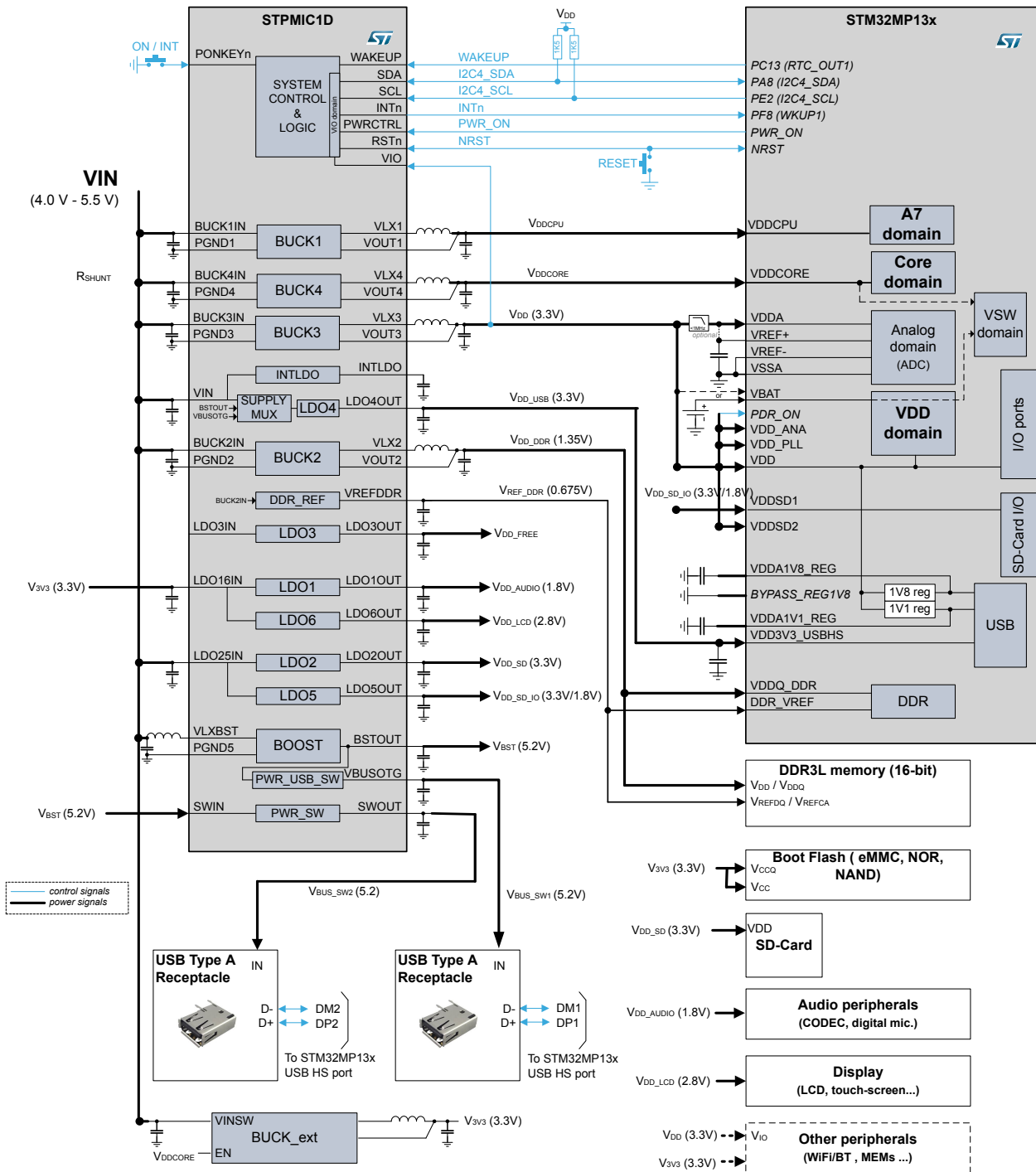
Table 3. Glossary

Term	Definition
FSBL	First stage boot loader
HSI	High speed internal oscillator
IC	Integrated circuit
LDO	Low drop out linear regulator
MPU	Microprocessor unit and refers to the STM32MP13x in this document
PMIC	Power management integrated circuit and stands for the STPMIC1D and STPMIC1A in this document
RTC	Real-time clock
SMPS	Switching mode power supply
SW	Software

4 Wall adapter supply application reference design

This reference design targets an application powered from a main supply adapter with DDR3L, a boot flash memory, an SD-Card, and two USB 2.0 HS host ports. The boot flash memory can be either eMMC, NAND, or NOR. Other peripherals like audio, display, wireless, MEMs are also included to illustrate the application. The main peripheral interfaces work with I/O voltage at 3.3 V.

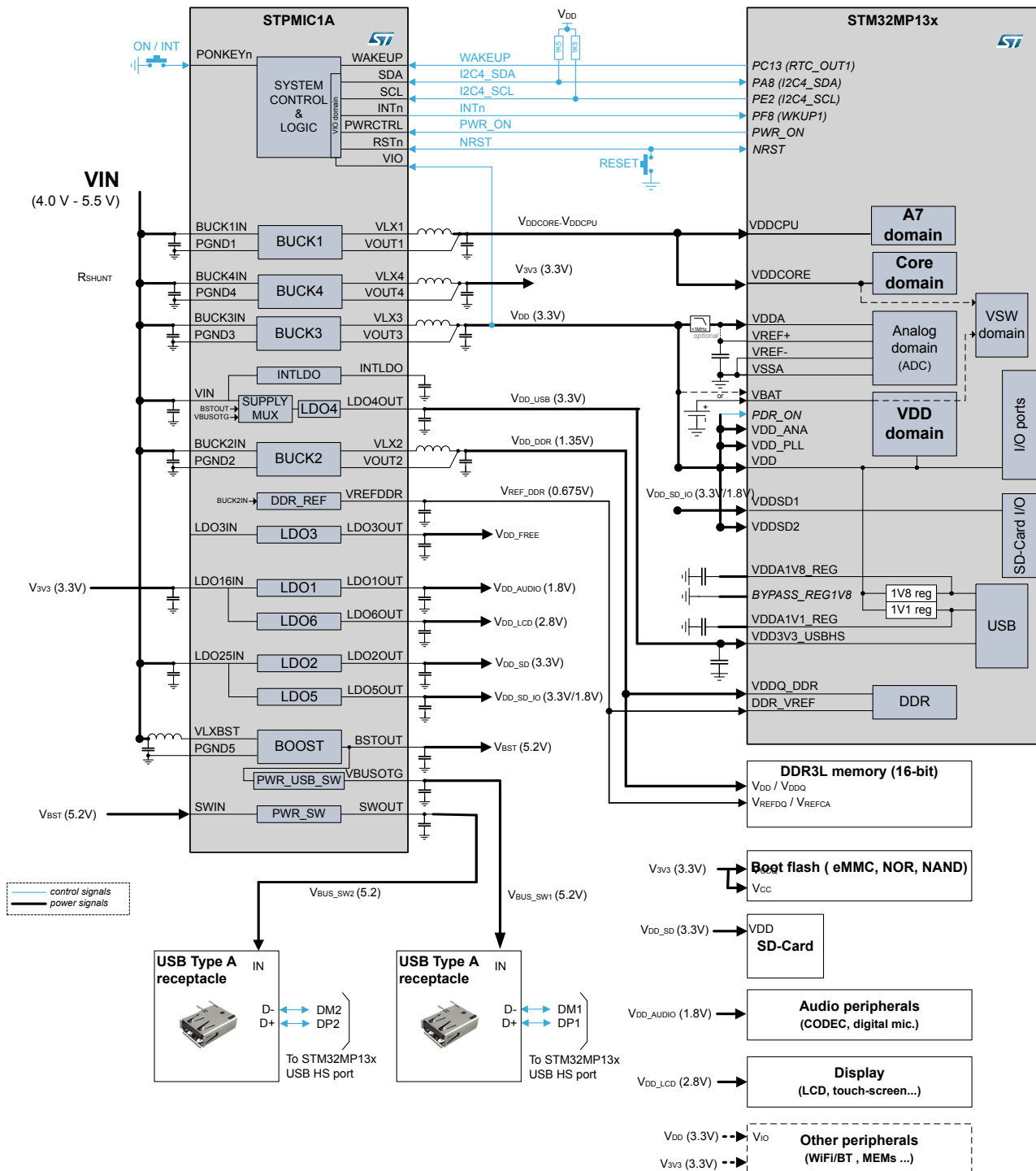
Figure 1. STM32MP13x and STPMIC1D with DDR3L, boot flash, SD-Card, and 2x USB HS



Note: The following are not shown in the diagram:

- MPU decoupling scheme (see [1]).
- PMIC discrete components value (see [2]).
- Additional protection, such as ESD, EMI filtering, over voltage.

Note: It is possible to remove the BUCK_ext in case the application does not use SDCard, and to power the eMMC using BUCK3 (3V3 or 1.8V) for eMMC VCCQ and LDO5 (3V3) for eMMC VCC. A small additional current due to eMMC is then present in Standby mode (BUCK3 active).

Figure 2. Low-cost version V_{DDCPU} merged with V_{DDCORE} using STPMIC1A


Note: The following are not shown in the diagram:

- MPU decoupling scheme (see [1]).
- PMIC discrete components value (see [2]).
- Additional protection, such as ESD, EMI filtering, over voltage.

In this configuration with V_{DDCPU} and V_{DDCORE} merged, LPLV-Stop2, and Run mode overdrive is not possible, however an SMPS (BUCK4) is available to power peripherals.

In [Figure 2](#) the use of STPMIC1A instead of STPMIC1D is recommended because the BUCK4 output voltage used to supply peripherals is 3.3 V by default on the STPMIC1A, whereas on the STPMIC1D it is 1.2 V and is used to supply V_{DDCORE} ([Figure 1](#)). Unless otherwise specified in the text, the STPMIC1A has similar characteristics to the STPMIC1D.

4.1 Power distribution

The PMIC integrates the regulators required to supply power to the STM32MP13x device, in addition to a set of regulators to supply some of the application peripherals.

Separate V_{DDCORE} and V_{DDCPU} supplies can be used, for example for overdrive on V_{DDCPU} or to shut down V_{DDCPU} in LPLV-Stop2 mode. In this case, BUCK4 is used to supply V_{DDCORE} , and an external SMPS BUCK convertor (BUCK_ext) is required to supply additional peripherals (V_{3V3}). BUCK3 supplying V_{DD} is limited to ~500 mA.

4.1.1 V_{DD} power domain (3.3 V)

V_{DD} is the reference design's main IO voltage domain used by the STM32MP13x, STPMIC1D, STPMIC1A, and peripherals.

It is powered from the PMIC BUCK3 step-down SMPS, which has high efficiency and low quiescent current across all load conditions.

The V_{DD} voltage domain is the first voltage available during power-up sequence (PMIC Rank1) and the last disabled during power down sequence.

V_{DD} is enabled in Run, LP-Stop, LPLV-Stop, LPLV-Stop2, and Standby modes. V_{DD} is disabled in Power off and VBAT modes.

The STPMIC1A and the STPMIC1D start V_{DD} (powered from BUCK3) at 3.3 V at power-up. See [\[2\]](#) for details.

V_{DD_PLL} , V_{DD_ANA} should be connected to V_{DD} .

V_{DDSD1} and V_{DDSD2} may be powered from V_{DD} depending on expected usage of the V_{DDSD1}/V_{DDSD2} I/O's usage. In our reference design in [Figure 1](#), V_{DDSD1} is used to supply STM32MP13x I/O's connected to the SD-Card and to avoid external level shifters during UHS-I mode. V_{DDSD1} is not connected to V_{DD} and it can be dynamically set to either 3.3 V or to 1.8 V.

The ADC analog voltage (V_{DDA}) and related analog reference voltage (V_{REF+}) may be powered from V_{DD} depending on expected ADC performance. If high ADC precision is expected, power V_{DDA} (and V_{REF+}) from a low noise power source with a voltage greater than 1.8 V. In that case, a PMIC LDO may be dedicated to supply the STM32MP13x V_{DDA} voltage domain.

4.1.2 V_{DDCORE} power domain

V_{DDCORE} is the main STM32MP13x digital power domain.

It is powered from the STPMIC1D BUCK4 step-down SMPS. This voltage domain is the next power domain to be available during the power-up sequence (STPMIC1D Rank2) and the penultimate to be disabled during the power-down sequence.

V_{DDCORE} is enabled in Run, LP-Stop, LPLV-Stop, and LPLV-Stop2 modes. V_{DDCORE} is reduced in LPLV-Stop and LPLV-Stop2 to save power. V_{DDCORE} is disabled in Standby and in Power off and VBAT modes.

The STPMIC1D starts V_{DDCORE} (powered from BUCK4) at 1.2 V at power-up. However, the software should increase V_{DDCORE} to the STM32MP13x minimum Run voltage after the boot phase (refer to the datasheet).

If the CPU overdrive feature is not required, V_{DDCORE} and V_{DDCPU} are both supplied by the STPMIC1A's BUCK1. In that case, the STPMIC1A's BUCK4 is used to supply peripherals at 3.3 V. Refer to [Section 4.1.6](#).

4.1.3 V_{DDCPU} power domain

The STM32MP13xD and STM32MP13xF devices have an enhanced consumer mission profile (see [\[7\]](#)). This profile allows the Arm® Cortex®-A7 CPU clock frequency run up to 900 MHz (see [\[6\]](#) for details and limitations).

The V_{DDCPU} supply voltage must be increased to the minimum overdrive Run value when the CPU frequency (F_{mpuss_ck}) operates above 650 MHz (refer to datasheet). When it does not operate in Run mode above 650 MHz, the V_{DDCPU} supply voltage must be set back to its nominal Run mode value. Refer to [Section 6](#) for a detail procedure to switch from Run to Run overdrive mode.

V_{DDCPU} is the STM32MP13x Arm® Cortex®-A7 CPU digital power domain.

V_{DDCPU} is powered from the STPMIC1D's BUCK1 step-down converter for applications supporting CPU overdrive (see [Figure 1](#)). V_{DDCPU} is powered with V_{DDCORE} from the STPMIC1A's BUCK1 step-down converter for applications not supporting CPU overdrive (see [Figure 2](#)).

With STPMIC1A, this voltage domain is the second power domain to be available during the power-up sequence (STPMIC1A Rank2). This voltage domain is the penultimate to be disabled during a power-down sequence. When using STPMIC1D, it is the third power domain to be available during a power-up sequence (STPMIC1D Rank3).

V_{DDCPU} is enabled in Run, LP-Stop, and LPLV-Stop modes. V_{DDCPU} voltage is reduced in LPLV-Stop to save power. V_{DDCPU} is disabled in LPLV-Stop2, Standby and in Power off and VBAT modes.

The STPMIC1A and the STPMIC1D start V_{DDCPU} (powered from BUCK1) at 1.2 V at power-up. However, the software should increase V_{DDCPU} to the STM32MP13x minimum Run voltage after the boot phase (refer to the datasheet).

If V_{DDCORE} and V_{DDCPU} are combined, BUCK1 also supplies V_{DDCORE} . In this case, the CPU frequency is limited to 650 MHz and the V_{DDCPU} supply is limited to the nominal Run mode voltage.

4.1.4 VDD_USB power domain (3.3 V)

V_{DD_USB} is dedicated to supplying power to the STM32MP13x USB PHY (VDD3V3_USBHS).

It is powered from the PMIC LDO4 linear regulator, which has been specifically designed for this feature.

This voltage domain is the last domain available during a power-up sequence (PMIC Rank 3). It is the first disabled during a power-down sequence; except for regulators enabled by software, which are disabled before LDO4 in Rank 0. See [2] for details.

The V_{DD_USB} regulator is managed by the software at run time. V_{DD_USB} (and also VBUS_SW1, VBUS_SW2) is disabled in Standby and in Power off and VBAT modes.

For the USB flashing use case (STM32MP13x peripheral boot from ROM), LDO4 (V_{DD_USB}) is needed at power-up to supply the USB PHY. LDO4 has a 3.3 V fixed voltage.

4.1.5 VDD_DDR (1.35 V), VREF_DDR (0.675 V) power domains

- V_{DD_DDR} is dedicated for DDR3L core power supply (VDD and VDDQ) and for the STM32MP13x DDR interface voltage domain (VDDQ_DDR)
- V_{REF_DDR} is dedicated for DDR3L reference voltage (VREFQ/VREFCA) and for STM32MP13x DDR reference voltage (DDR_VREF) at V_{DD_DDR}/2 (0.675 V)
- V_{DD_DDR} (1.35 V) is powered from the PMIC BUCK2 step down SMPS having high efficiency and low quiescent current in any load conditions. BUCK2 is powered from VIN.
- V_{REF_DDR} (0.675 V) is powered from the PMIC REFDDR sink/source LDO. When enabled REFDDR output voltage is equal to V_{DD_DDR} / 2 (BUCK2 output voltage / 2)

The PMIC does not start V_{DD_DDR} and V_{REF_DDR} at power-up. They must be powered up and powered down by STM32MP13x software, respectively at STM32MP13x boot up and shutdown.

DDR3 boot sequence recommended by JEDEC is the following:

1. After 100 μ s, set BUCK2 at 1.35 V and enable BUCK2 (V_{DD_DDR}).
2. Enable DDR_REF LDO (V_{REF_DDR}).

Step 1. and 2. can be inverted since LDO3 and DDR_REF LDO output rise to V_{DD_DDR}/2 once BUCK2 is enabled.

Note: In the reference design in [Figure 1](#), there is no V_{TT_DDR} (Fly-by topology termination) since usually only one memory chip is needed for 16-bit DDR3 configuration.

4.1.6 V3V3 power domain (3.3 V)

When V_{DDCORE} and V_{DDCPU} are merged, V_{3V3} can be powered from the BUCK4 step-down SMPS. However, in this case it is recommended to use the STPMIC1A since the default setting of BUCK4 is 3.3 V (as in [Figure 2](#)).

This configuration saves the cost of one external BUCK however it cannot be used with LPLV-Stop2 mode nor with MPU frequencies > 650 MHz (Run Overdrive).

V_{DDCORE} and V_{DDCPU} can be separated in order to use Run Overdrive mode on V_{DDCPU} or shut down V_{DDCPU} in LPLV-Stop2 mode. When this happens, V_{3V3} can be powered from an additional external BUCK_ext step-down SMPS (as in [Figure 1](#)).

The external BUCK should have the following characteristics:

- EN pin threshold (V_{ih}) < LPLV-Stop mode V_{DDCORE} min value (0.85 V)
- BUCK supports active discharge with 3 ms max discharge duration (same as PMIC rank duration)

V_{3V3} is a general-purpose power supply, which should be instead of V_{DD} for the following:

1. Power supply for boot Flash device (eMMC, NAND, NOR) core domain (VCC).

V_{3V3} power domain:

- is disabled during a reset power cycle, complying with the Flash reboot requirement.
- is automatically enabled at boot.
- is disabled in Standby and Off mode.

V_{3V3} must be enabled before entering Standby in the BUCK4_MAIN_CR register (see [2]). V_{3V3} can then be disabled in Standby (PWR_ON signal low) and reenabled when the MPU leaves Standby mode (PWR_ON signal high). This applies when the following configuration is present:

- V_{DDCORE} and V_{DDCPU} are merged as in Figure 2
- BUCK4 is delivering V_{3V3},

When an external BUCK (BUCK_ext) is used as in Figure 1, V_{DDCORE} controls the enable of the discrete SMPS. The enable threshold of the discrete SMPS should be low enough to guarantee that the V_{3V3} SMPS is kept on in LPLV-Stop (> 0.95 V)

2. LDO16IN preregulation

If the LDOs are not expected to provide more than 3.0 V. Supplying them with a preregulated 3.3 V input enhances the LDO's power efficiency and reduces thermal dissipation.

In this case, V_{3V3} can be disabled to save power only when all peripherals connected to the supplied LDOs are not being used.

4.1.7 SD-Card power (VDD_SD)

SD-Card power domain

The SD-Card has a VDD_SD power supply and its IO's are connected to the STM32MP13x GPIO's powered by VDDSDx supply.

There are two main possible configurations for SD-Card power domains:

- Default speed configuration with
 - STM32MP13x VDDSDx connected to VDD
 - SD-Card VDD_SD powered by LDO5 with the same voltage as VDD
 - LDO2 not used
- UHS-I mode configuration with
 - STM32MP13x VDDSDx powered by LDO5 (VDD_SD_IO), which allows IO's voltage to switch from 3.3 V to 1.8 V dynamically
 - VDD_SD powered by LDO2.

This is shown in Figure 1 and Figure 2. This configuration removes the need for an external SD-Card level shifters since the SD-Card I/Os are connected to the STM32MP13x device VDDSD1 power domain I/Os.

4.1.7.1 Default speed configuration

SD-Card power from LDO5 (VDD_SD), SD-Card I/Os power from VDD (VDD_SD_IO)

On the PMIC, the LDO5 SD-Card (VDD_SD) voltage domain is the second domain to be enabled during the power-up sequence (Rank2). This voltage domain is also the penultimate to be disabled during the power-down sequence.

STPMIC1D has NVM settings to set LDO5 (VDD_SD) at 3.3 V when STPMIC1D powers up.

STPMIC1A has NVM settings to set LDO5 (VDD_SD) at 2.9 V when STPMIC1A powers up, so after the boot sequence the software must reprogram LDO5 to 3.3 V.

If the SD-Card device is the boot Flash peripheral, LDO5 (VDD_SD) is needed at power up to supply the SD-Card. LDO5 allows the STM32MP13x to access this memory from the ROM to boot up. The application software must program the PMIC to power off the SD-card in Standby mode (PWR_ON signal low). It must also program it to power on the SD-card in Run mode (PWR_ON signal high) before the application goes into Standby mode.

When the application recovers from Standby mode to Run mode, the SD-card is powered and ready to be accessed by the STM32MP13x boot ROM (peripheral boot). If STPMIC1A is used, the LDO5 NVM settings must be reprogrammed to select 3.3 V before use in a production application.

4.1.7.2 UHS-I mode configuration

In Figure 1 and Figure 2 configurations, the UHS-I mode is supported allowing $V_{DD_SD_IO}$ to switch dynamically from 3.3 V to 1.8 V.

SD-Card I/Os from LDO5 ($V_{DD_SD_IO}$)

On the PMIC, the LDO5 SD-Card I/Os voltage domain ($V_{DD_SD_IO}$) is the second domain to be enabled during the power-up sequence (Rank2). It is also the penultimate to be disabled during the power-down sequence.

Note: *LDO2 cannot be used in this configuration to supply $V_{DD_SD_IO}$ since it does not power up automatically at reset. The SD-card IO's supply ($V_{DD_SD_IO}$) must always be powered before the SD-Card supply (V_{DD_SD}).*

Software manages the $V_{DD_SD_IO}$ regulator at run time. It is disabled in Standby and in Off modes.

STPMIC1D has NVM settings to set LDO5 ($V_{DD_SD_IO}$) at 3.3 V when STPMIC1D powers up.

STPMIC1A has NVM settings to set LDO5 ($V_{DD_SD_IO}$) at 2.9 V when STPMIC1A powers up, so after a boot sequence the software must reprogram LDO5 to 3.3 V.

SD-Card power from LDO2 (V_{DD_SD})

The SD-Card main supply V_{DD_SD} is powered from the PMIC LDO2 linear regulator.

When enabling the SD-Card, the SD-Card I/Os ($V_{DD_SD_IO}$) must be set to the same value as V_{DD_SD} (IOs at 3.3 V). On PMIC, LDO2 does not start automatically at power-up and it must be enabled by software and set to 3.3 V. Software may disable V_{DD_SD} if no read/write access is expected.

When the UHS-I mode is requested, the software sends a command to the SD-Card to internally change its GPIOs to 1.8 V. Then the software changes the PMIC setting to drive 1.8 V on LDO5 ($V_{DD_SD_IO}$). The software can then change the SDMMCx GPIO to high-drive HSLV mode by setting the SYSCFG_HSLVENxR register to 0x1018. In SYSCFG_HSLVENxR, x=4 for SDMMC1, x=5 for SDMMC2.

Caution: SYSCFG_HSLVENxR should not be set to HSLV before LDO5 is set to 1.8 V or the device could be damaged.

In case an application issue occurs, the software must switch off V_{DD_SD} for a power cycle. The power cycling is mandatory to go back from UHS-I mode (IOs at 1.8 V) to the default speed mode (IOs at 3.3 V). This is why a dedicated regulator LDO2 is used to power V_{DD_SD} . The power cycle is initiated by a software system reset that is handled by the PMIC.

The SYSCFG_HSLVENxR HSLV programming is also reset during this power cycling phase.

If the SD-Card is deactivated by software, the software should first reset SYSCFG_HSLVENxR before setting LDO5 ($V_{DD_SD_IO}$) to 3.3 V. Then, the software can send a command to the SD-card to internally change its GPIOs to 3.3 V.

4.1.7.3 SD-Card as boot device

If the SD-Card device is the boot Flash peripheral, both the SD-Card main supply and the SD-Card IO supply must be powered up at reset.

If STPMIC1A is used as in Figure 1 and Figure 2 configurations, the STPMIC1A LDO5 NVM needs to be reprogrammed to select 3.3 V before use in a production application.

If the LDO2 (V_{DD_SD}) is needed at power up to supply the SD-Card (UHS-I use case), the application software must program the PMIC in order to power off the SD-card in Standby mode (PWR_ON signal low) and power on the SD-card in Run mode (PWR_ON signal high) before the application goes into Standby mode.

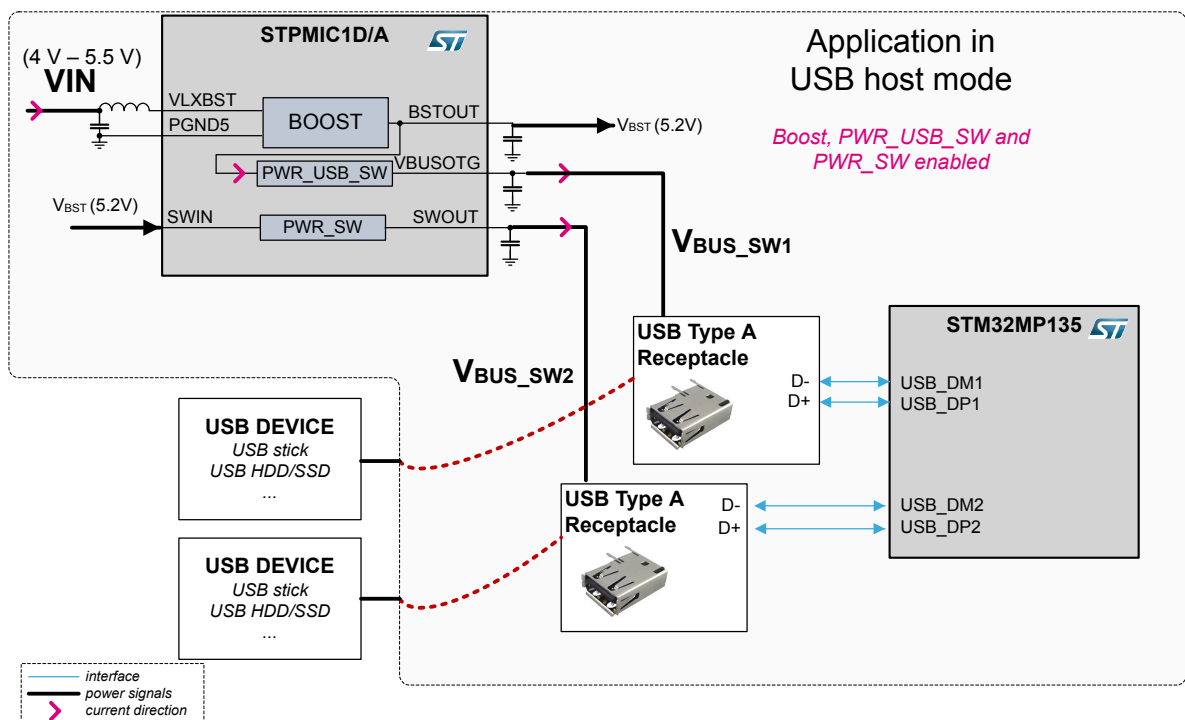
In this case, when the application recovers from standby mode to run mode, the SD-card is powered and ready to be accessed by the STM32MP13x boot ROM (peripheral boot). The LDO2 NVM settings must be reprogrammed to power up LDO2 automatically at power-up (rank 1) before use in a production application.

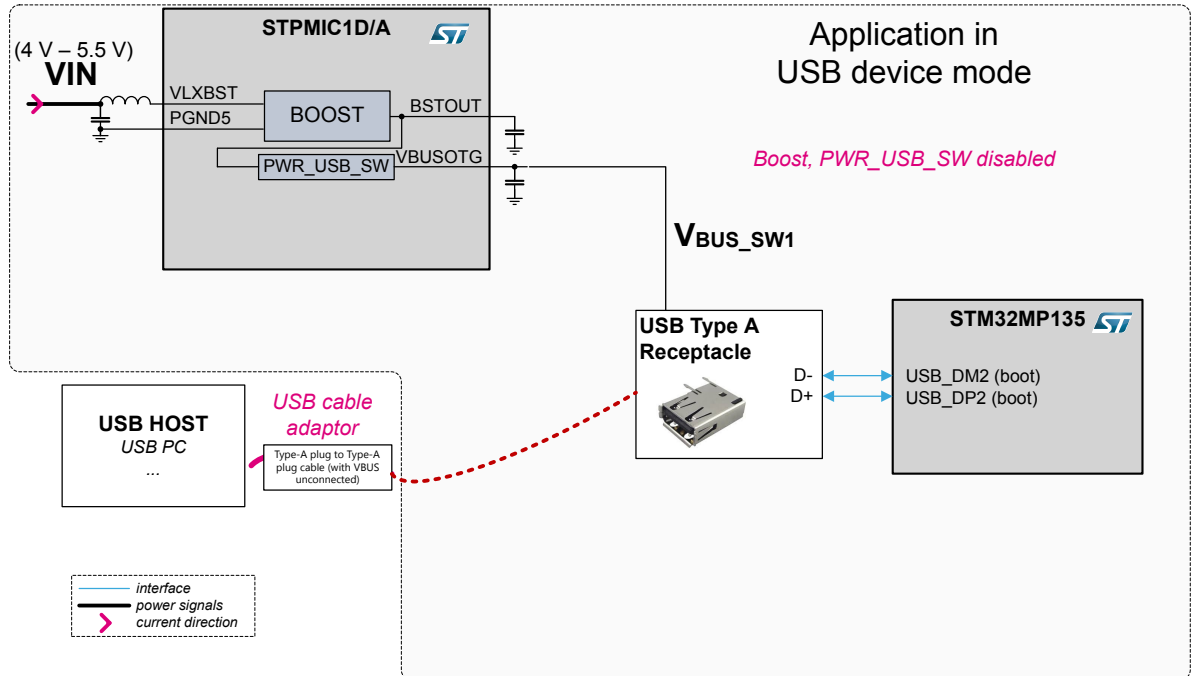
4.1.8 VBUS_SW1 and VBUS_SW2 power domain (5.2 V)

V_{BUS_SW1} and V_{BUS_SW2} are dedicated power domains for the two USB high-speed host interfaces. They are connected to the IN pin of the USB Type-A receptacles.

- Two USB device peripherals are connected to the application and are powered from V_{BUS_SW1}/V_{BUS_SW2} from the PMIC (see Figure 3).
- There is no need for external power supplies. The PMIC delivers compliant USB VBUS voltages from the VIN wall supply (4 V to 5.5 V). This is done thanks to the PMIC's boost SMPS that regulates VBUS at 5.2 V with VIN from 4 V to 5.5 V.
- Software manages boost at run time.
- When at least one of the V_{BUS_SW1}/V_{BUS_SW2} is enabled, the VDD_USB supply must also be enabled in order to enable USB PHY operation.

Figure 3. V_{BUS_SW1} , V_{BUS_SW2} power path in USB host mode



Flashing through USB with Type-A connector
Figure 4. Flashing through USB with Type-A connector


For applications that have only Type-A receptacle USB port, it is still possible to perform serial boot over the USB on the STM32MP13x system. This can be done in USB device mode.

This specific Boot mode is different from the classical boot method that uses the USB Type-B receptacle. To support this specific Boot mode, there are two requirements:

- A USB Type-A receptacle is used in USB device mode
- A dedicated non USB-compliant Type-A to Type-A plug cable is required.

The host PC USB Type-A receptacle must be connected on one side of the dedicated cable while the other side is connected to the STM32MP13x device USB Type-A receptacle.

The STM32Cube Programmer is used on the host PC to flash the Linux distribution on the target STM32MP13x device.

The STM32MP13x must be ready for USB/UART boot:

- Either the Flash memory is empty and the device is automatically set to USB/UART boot
- Or boot pins must be set to USB/UART boot mode (BOOT[2:0]='000' or '110')

In this specific use case, the VBUS_SW1 signal is not connected to the STM32MP13x hence the boot ROM does not probe VBUS to detect a host PC connection.

During the flashing operation, the boost and PWR_USB_SW must be kept de-activated.

1. Initial conditions:
 - PC ready to enumerate USB DFU
 - Boot pin set to USB/UART mode if Flash is not empty
 - Board power supply is OFF
 - USB cable connection between PC host and STM32MP13x board
2. Power supply is switched ON (or reset of the STM32MP13x device)
3. PC enumerates USB DFU
4. Flashing starts with STM32Cube Programmer

Once flashing is complete, the USB cable between the host PC and the STM32MP13x device must be disconnected. This must be done before booting the system in application mode from the Flash.

4.1.9 VDD_AUDIO (1.8 V), VDD_LCD (2.8 V) power domains

V_{DD_AUDIO} and V_{DD_LCD} are provided as shown in the reference design in Figure 1 and Figure 2.

They are respectively powered from the LDO1 and the LDO6 linear regulators.

V_{DD_AUDIO} and V_{DD_LCD} are not enabled by the PMIC at power-up. They are enabled and set at the right voltage after power-up by software when the related peripheral requires them.

Software manages V_{DD_AUDIO} and V_{DD_LCD} regulators at run time. Software can disable V_{DD_AUDIO} and V_{DD_LCD} if no peripheral access is expected. V_{DD_AUDIO} and V_{DD_LCD} are disabled in Standby and in Off modes.

4.2 Control signals and interface between STM32MP13x and PMIC

The following sections outline the way the STM32MP13x microprocessor communicates with the PMIC device. There are several interface choices that can be used depending on the application requirements. Each interface is described in this first part.

I²C interface:

The PMIC can be controlled by the STM32MP13x via the I²C interface to:

- Enable or disable a regulator
- Set a regulator voltage and mode (low-power or high-power)
- Set low-power management (PWRCTRL behavior)
- Set the interrupt controller or read interrupt status
- Set the protection (watchdog, overcurrent, under-voltage) or read protection status
- Reprogram the NVM to change the startup behavior.

Note: The PMIC has special default NVM settings that allow it to boot an STM32MP13x application with 3.3 V IOs either from

- from a USB interface for flashing or loading and then for executing software
- from an SD-Card
- from a flash memory, such as an eMMC.

Once the STM32MP13x is able to execute software, it is also able to reprogram the PMIC NVM on-the-fly to fine-tune the final application; during tests and in mass production.

ON/INT push button:

The user “ON/INT” push button is connected to the PMIC PONKEYn pin (active low). This button allows:

- To power up the PMIC.
- To send an interrupt to the STM32MP13x on a button-press event or a button-release event when the application is operating.
- To force a power-off of the PMIC with a long press (16 s by default).

NRST signal:

NRST is a bidirectional active low signal for the STM32MP13x and the PMIC. The STM32MP13x NRST pin and PMIC RSTn pin are of digital input/open drain output topology:

- When PMIC asserts RSTn (such as during the power-up or the power down sequence), it drives the NRST signal low: the STM32MP13x is forced into a reset state until PMIC releases the NRST.
- PMIC immediately asserts the RSTn pin and performs a noninterruptible power-cycle in the following cases:
 - when the STM32MP13x asserts an NRST signal, such as an STM32MP13x watchdog reset
 - when a user presses on the “RESET” button,

The PMIC performs a power-down sequence followed by a power-up sequence and finally releases the RSTn.

Note: At the end of a power-cycle sequence, PMIC waits for the NRST signal to go high before rearming the reset to avoid an infinite reset loop.

INTn signal:

The INTn is a PMIC output active low interrupt line connected to the STM32MP13x PF8 input pin. PF8 has both interrupt and wake-up capability:

- To manage an interrupt from the PMIC when the STM32MP13x is in either Run or Stop mode.
- To wake up the STM32MP13x when it is in Standby mode.

PWR_ON signal:

The PWR_ON signal is driven by the STM32MP13x PWR_ON pin to control PMIC PWRCTRL pin.

This allows the STM32MP13x to switch the PMIC power strategy very quickly to one of the following application power modes:

- From Run mode to LPLV_Stop or LPLV-Stop2 mode and back
- From Run mode to Standby mode and back.

(See [3] for details about low-power mode management and PWR_ON pin setting when using PMIC)

After a power-up or a reset, the PMIC PWRCTRL pin is disabled. Before going in Low-power mode, the STM32MP13x sets the PMIC via I²C to program the expected power behavior according to the PWR_ON signal state.

WAKEUP signal (optional):

The WAKEUP signal is driven by the STM32MP13x PC13 (RTC_OUT) pin to control the PMIC WAKEUP pin. It allows the STM32MP13x to power up the PMIC; typically, when the real-time clock wake-up timer elapses.

This feature is available if a coin cell battery is connected to the STM32MP13x VBAT pin.

5 Power management

The following power modes are reviewed in the following sections:

- Operating modes
- Application power-up and power-down modes
- Low-power management mode
- User reset and crash recovery management
- Software management examples

5.1 Operating modes

The application can switch to different operating modes depending on the system activity. The operating modes are managed by the STM32MP13x. The operating modes control the power management and the clock distribution (see details in [3]).

Table 4 summarizes the application level operating modes. The PMIC power modes depend on the application operating mode.

LP-Stop mode is not covered since LPLV-Stop or LPLV-Stop2 are more appropriate in the context of this AN.

Table 4. Application operating modes

Operating mode	STPMIC1D Power mode	VIN ⁽¹⁾	PWR_ON	Description	Notes
Run	Power-on-main	> VINOK_fall	1	V _{DD} power on V _{DDCORE} , V _{DDCPU} , power on, system clock on V _{3V3} power on, V _{DD_SD} power on/off DDR3 active/auto refresh	(2)
Stop	Power-on-main	> VINOK_fall	1	V _{DD} , V _{DDCPU} power on V _{DDCORE} power on, system clock off V _{3V3} power on, V _{DD_SD} power on/off DDR3 active/auto refresh	(2)
LPLV-Stop	Power-on-alternate	> VINOK_fall	0	V _{DD} power on V _{DDCORE} , V _{DDCPU} power on at lower voltage, system clock off V _{3V3} power on, V _{DD_SD} power on/off DDR3 auto refresh/self-refresh	(3)
LPLV-Stop2	Power-on-alternate	> VINOK_fall	0	V _{DD} power on V _{DDCPU} power off V _{DDCORE} power on at lower voltage, system clock off V _{3V3} power on, V _{DD_SD} power on/off DDR3 auto refresh/self refresh	(3)
Standby	Power-on-alternate	> VINOK_fall	0	V _{DD} power on V _{DDCORE} , V _{DDCPU} power off, system clock off V _{3V3} power off, V _{DD_SD} power off DDR3 self-refresh/off	(3)
Power-off	Off	> VIN_POR_fall	-	All power off	-
	No_supply	< VIN_POR_fall	-	All power off	-
Coin-cell-VBAT	No_supply	< VIN_POR_fall	-	All power off except the STM32MP1 Series microprocessor VSW	(4)

1. PMIC hardware thresholds. See [2] for details.
2. The difference between Run and Stop modes is only based on the STM32MP1 Series microprocessor clock management. For power management, there is no difference between Run and Stop mode.
3. There is no difference on the PWR_ON control pin when entering LPLV_Stop, LPLV_Stop2, or Standby mode from Run mode. PWR_ON signal goes from high to low in both cases. Before entering LPLV_Stop, LPLV_Stop2 or Standby mode, the STM32MP1 Series microprocessor programs the PMIC via the I²C interface to set the regulators accordingly. This LPLV-Stop2 mode is not available using STPMIC1A as shown in Figure 2 where V_{DDCPU} and V_{DDCORE} are merged.
4. To retain the content of the STM32MP1 Series microprocessor VSW domain (RTC, backup registers, backup RAM, and retention RAM) when V_{DD} is turned off, the STM32MP1 Series microprocessor VBAT pin can be connected to an optional coin cell battery.

5.1.1 Application turn-on/turn-off conditions

When the application is in Power-off mode, a turn-on condition is required to power up the PMIC into Power-on mode. Similarly, if the application needs to go into Power-off mode, a turn-off condition is required to power down the PMIC.

Note: *Power-up: PMIC transitional power-up state where the regulators start sequentially in a predefined order (rank) and voltage and ends by releasing the NRST signal. After this state, the PMIC goes into power-on state and remains there, the application can now be run. This state is reached by one of the following:*

- *from Off mode with a turn-on condition*
- *from NO_SUPPLY with VIN voltage rising higher than VINOK_rise (AUTO turn-on).*

Power-down: PMIC transitional power-down state where the NRST is asserted leading to the regulators stopping sequentially in the reverse order of the power-up sequence. After this state, the PMIC is in the off state and remains as such until a turn-on condition occurs. This state is reached from the power-on state with a turn-off condition.

The PMIC autonomously manages the power-up and the power-down sequence when respectively a turn-on or a turn-off condition occurs (see [2] for details).

Turn-on conditions

The PMIC automatically powers up when the power supply source (VIN) rises above VINOK_rise (an auto turn-on feature enabled by default in PMIC NVM). If the PMIC is in the off state (and VIN > VINOK_rise), it can be powered up by one of three external triggers:

- “ON/INT” user button press: PONKEYn pin voltage falling-edge.
- The STM32MP13x wake up event occurs (for example RTC or tamper wake up via STM32MP13x PC13 pin): WAKEUP pin voltage rising-edge.

Note: *The STM32MP13x wake up feature is available if a coin cell battery is connected to STM32MP13x VBAT pin.*

Turn-off conditions

A turn-off condition causes the PMIC to power down and go into the off state. In the off state, all regulators are turned off. If the PMIC is in the on state, it can be powered down by one of six conditions:

- Software switch-off: I2C command sent by the STM32MP13x to the PMIC.
- “ON/INT” user button long press: when the button is pressed for 16 seconds, the PMIC is turned off (the delay is programmable).
- Thermal shutdown: if overheating, PMIC shuts down and restarts when the temperature returns to a correct level.
- Over-current protection: if enabled by software, an over-current condition on a regulator causes the PMIC to shut down.
- Watchdog: if enabled by software, when the countdown timer reaches 0, the PMIC goes to the off state.
- VINOK_fall: if the supply voltage VIN goes below the VINOK_fall threshold, the PMIC goes to the off state.

Note: *An application can set the PMIC “restart request” feature to automatically restart the application after a turn-off condition (see [2] for details).*

5.1.2 PMIC restart_request and mask_reset options

Before a turn-off condition occurs, the STM32MP13x software can program the PMIC to restart instead of turning it off by setting the restart_request feature in the PMIC. This setting must be done before initiating the turn-off condition; such as after an application power-up.

For example, the software can completely reboot the application by setting the restart request bit in the PMIC (RREQ_EN = 1), then programming a software switch off (SWOFF = 1). The PMIC performs a power cycle sequence: a power down sequence (disabling all regulators) followed by a power up sequence (restarting regulators then releasing NRST signal).

If the application needs one or several PMIC regulators to be kept enabled during a power cycle, the STM32MP13x software can program the PMIC mask_reset option by setting the PMIC BUCKS_MRST_CR register to target the buck converter and LDOS_MRST_CR register to target LDOs (see [2] for details on the PMIC mask_reset option). This setting must be done before a power cycle, such as after the application power-up.

This is typically the case for the BUCK3 powering the STM32MP13x V_{DD} power domains in case V_{BAT} is not present in the application.

In such case, the power cycle on V_{DD} must be masked (BUCKS_MRST_CR[2] = 1) to prevent losing:

- The STM32MP13x backup RAM

- The backup register content.

If BUCKS_MRST_CR[2] is not set, this information is lost when a power cycle is triggered by an NRST from the STM32MP13x (see [Section 5.4 User reset and crash recovery management](#)) or by a turn-off condition with the restart_request bit enabled.

5.2 Application power-up/power-down sequence

The power-up sequence is the transition managed by the STPMIC1D between Power-off and Run operating modes and similarly for the power-down sequence. The application power-up and power-down sequences are shown in [Figure 5](#) according to the reference design in [Figure 1](#).

Note: The power sequences described in the following subsections are only applicable to [Figure 1](#) using STPMIC1D. The same concept applies to [Figure 2](#) using STPMIC1A but with a slightly different sequence: BUCK1 (VDDCORE/VDDCPU) ranking is Rank2 (instead of Rank3). This difference makes VDDCPU to follow the same sequence as VDDCORE. BUCK4 (Rank2) is now used to supply V_{3V3} .

5.2.1 Power-up by main supply (VIN) plugin

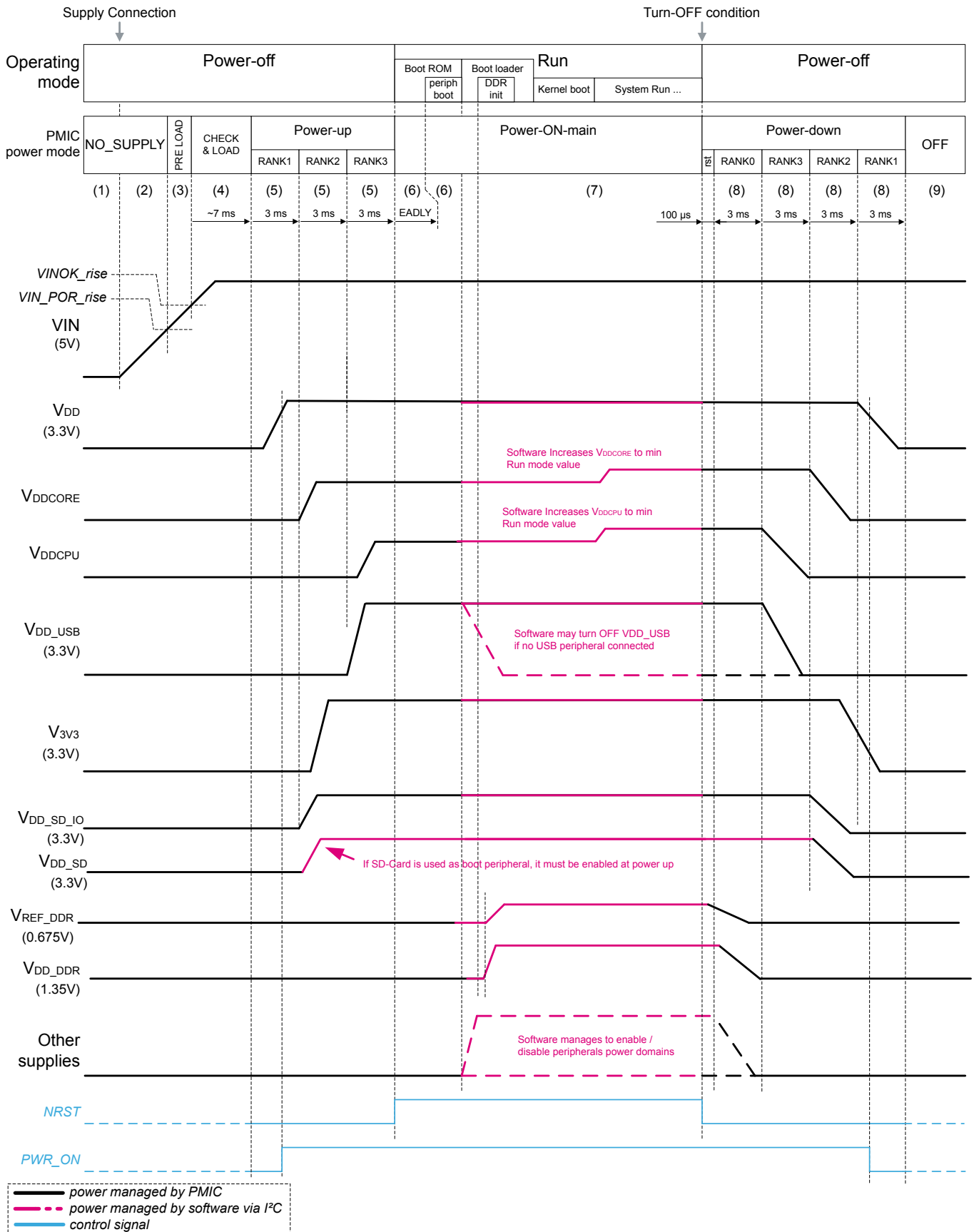
The application has no power initially. When a power supply is connected, the application starts automatically when VIN rises (the STPMIC1D has the auto turn-on enabled by default in its NVM). When the STPMIC1D is powered up, the application boots (including the DDR initialization) and finally the system reaches Run mode. When a turn-off condition occurs, the STPMIC1D powers down and goes into the Off mode: the application goes into Power-off mode. The whole process is detailed below and illustrated in [Figure 5](#):

1. Application has no power or the STM32MP13x is in VBAT mode (powered from a coin cell battery to supply the STM32MP13x V_{SW}).
2. A valid power supply source ($VIN > VINOK_{rise}$) is connected to the application. VIN voltage rises.
3. Once VIN supply is above $VIN_{POR_{rise}}$ (2.3 V):
 - a. The STPMIC1D initializes and preloads its NVM contents.
 - b. The STPMIC1D asserts the NRST.
4. VIN supply rises above $VINOK_{rise}$, the STPMIC1D checks the turn-on condition (auto turn-on is enabled in the STPMIC1D NVM). The STPMIC1D starts a power-up sequence as a valid turn-on condition is detected.
5. The STPMIC1D follows the power-up sequence:
 - a. Rank1: BUCK3 (V_{DD}) is enabled at 3.3 V.
 - b. Rank2: after 3 ms BUCK4 (V_{DDCORE}) is enabled at 1.2 V and then V_{3V3} at 3.3 V through BUCK_ext. LDO5 ($V_{DD_{SD_{IO}}}$) is enabled at 3.3 V. If the SD-Card is used during boot, LDO2 ($V_{DD_{SD}}$) is also enabled. LDO2 ($V_{DD_{SD}}$) must be previously programmed by the NVM and set to 3.3 V Rank2. Once V_{DDCORE} reached the $V_{TH_{VDDCORE}}$ threshold, a V_{DDCORE_TEMPO} is started. After that, the STM32MP13x performs an internal initialization and releases its reset. The STM32MP13x remains in reset as the STPMIC1D is still asserting its NRST signal.
 - c. Rank3: after 3 ms BUCK1 (V_{DDCPU}) is enabled at 1.2 V, LDO4 ($V_{DD_{USB}}$) is enabled at 3.3 V (hard setting). After 3 ms, the STPMIC1D releases the NRST signal.
6. The NRST signal rises since both the STM32MP13x and the STPMIC1D release their respective reset pins:
 - a. The STM32MP13x EADLY delay timer⁽¹⁾ (10 ms) starts.
 - b. When the EADLY delay elapses, the boot ROM software starts accessing external peripherals to load and execute bootloader software. The external peripherals can be either eMMC or SD-Card depending on the STM32MP13x boot pin settings.
 - c. The bootloader can control any STPMIC1D regulator (such as initialize an LCD and plot a splash screen).

7. The bootloader initializes the DDR then loads and executes the Kernel:
 - a. DDR_VREF (V_{REF_DDR}) is enabled.
 - b. BUCK2 (V_{DD_DDR}) is enabled at 1.35 V
 - c. The software waits for at least 1.4 ms for BUCK2 ready
 - d. The software initializes the STM32MP13x DDR controller and DDR3 device.
 - e. The bootloader loads and executes the kernel. The kernel initializes.
 - f. The software increases the STPMIC1D BUCK1 (V_{DDCPU}) and BUCK4 (V_{DDCORE}) to typical Run mode values.
STPMIC1D default BUCK1 and BUCK4 values (1.2 V) allow correct processing of the boot sequence, however the Run mode requires higher voltage value (see [6])
 - g. System is now running.
8. A turn-off condition occurs. The STPMIC1D performs a power-down sequence:
 - a. The STPMIC1D asserts the NRST (STM32MP13x reset) and waits for 100 μ s.
 - b. Rank0: the STPMIC1D disables all regulators it has not enabled at power-up (BUCK2, DDR_REF, LDO1, LDO6, BOOST, PWR_USB_SW, PWR_SW).

Note: As soon as a STPMIC1D regulator is disabled, a pull-down resistor is enabled on its output to discharge the decoupling capacitor voltage. The LDO and BUCK regulator output voltages are discharged in 3 ms and 1.5 ms respectively. Refer to [2] for details.

- c. Rank3: after 3 ms BUCK1 (V_{DDCPU}), LDO4 (V_{DD_USB}) are disabled.
 - d. Rank2: after 3 ms BUCK4 (V_{DDCORE}), then V_{3V3} and LDO5 ($V_{DD_SD_IO}$) are disabled. If an SD-Card is used during boot, LDO2 (V_{DD_SD}) is also disabled. LDO2 (V_{DD_SD}) was programmed by the NVM and set to 3.3 V Rank2.
 - e. Rank1: after 3 ms BUCK3 (V_{DD}) is disabled.
9. After 3 ms, the STPMIC1D is in Off mode: the application is in Power-off.
 1. The EADLY timer prevents the boot ROM from performing any access to the boot peripheral before it is ready. It waits for a stable voltage on the Flash memory (eMMC or SD-Card) to ensure that the boot software is reliably read by the boot ROM. Default delay period after reset is 10 ms. (see [5] for details).

Figure 5. Power-up/power-down sequence


5.2.2 Power-up from the STPMIC1D Off mode

The application in [Figure 1](#) is powered up from NO_SUPPLY state, where the wall supply VIN connection is the turn-on condition. Auto turn-on is enabled in the STPMIC1D NVM.

A power-up from the STPMIC1D Off mode follows a similar sequence as in [Figure 5](#) from Off mode. The difference is when a turn-on condition occurs, the sequence starts from step (4) "CHECK & LOAD" instead of waiting for VIN to rise.

The differences are detailed below:

- The STPMIC1D is initially powered from a VIN voltage that is higher than VINOK_rise. This allows the STPMIC1D to power up. A wall supply in the VIN connection triggers the VIN rise in [Figure 5](#).
- Steps (1), (2), (3) of [Figure 5](#) are replaced by a single step. This step merges the "NO-SUPPLY" and "PRE-LOAD" of the STPMIC1D power modes to "OFF".
- The "Supply connection" event is replaced by the "Power-ON condition" event and is placed between "OFF" and "CHECK & LOAD" of the STPMIC1D state.

5.2.3 Power-down by battery removal

The application in [Figure 1](#) is powered off by a turn-off condition with VIN maintaining a valid voltage.

If the application is powered off by a power removal, the turn-off condition is VIN dropping below VINOK_fall. Once VIN supply is below VINOK_fall, the STPMIC1D asserts an NRST for 100 μ s then powers-down as shown in step (8) onwards in [Figure 5](#).

Limitation: when the main power is removed, VIN voltage drops very quickly to the VINOK_fall value, in less than a few milliseconds (depending on system activity). Only then the power-down sequence start. As soon as the STPMIC1D asserts an NRST, system activity is immediately stopped and power consumption drops, slowing the VIN drop. Nevertheless, VIN may drop below the VIN_POR_fall threshold before the power-down sequence ends. In this case, the STPMIC1D regulator's pull-down discharge resistors are no longer controlled by the STPMIC1D. A bulk decoupling capacitor (a few hundred μ F) may be inserted on the VIN path to limit VIN drop speed.

5.3 Low-power mode management

The STM32MP13x supports several operating modes to reduce power consumption (see [Section 5.1 Operating modes](#)). This section describes the LPLV-Stop, LPLV-Stop2, and Standby low-power modes (see [\[3\]](#) for details).

Note: *Stop mode concerns the STM32MP13x internal clock management without external power management. So, Stop mode is not described in this section.*

Note: *The power sequences described in the following subsections are only applicable to [Figure 1](#) using STPMIC1D. The same concept applies to [Figure 2](#) using STPMIC1A but with a slightly different sequence, since BUCK1 (VDDCORE/VDDCPU) has a Rank2 instead of Rank3, so VDDCPU follows the same sequence as VDDCORE. Also BUCK4 (Rank2) is now used to supply V_{3V3} .*

Low-power modes are managed by the STM32MP13x. The STM32MP13x PWR_ON output pin is connected to the STPMIC1D PWRCTRL input pin. The STPMIC1D states can then be switched: Power-ON-main to Power-ON-alternate and vice versa.

After power-up, the STPMIC1D goes into POWER_ON MAIN until the STM32MP13x tells the STPMIC1D to activate the PWRCTRL pin in active low polarity by setting PWRCTRL_POL = 0 and PWRCTRL = 1 in the STPMIC1D MAIN_CR register. [Table 4](#) summarizes the STPMIC1D states and corresponding operating modes according to the PWR_ON signal state.

When the STPMIC1D goes from POWER_ON MAIN to POWER_ON ALTERNATE, it internally switches from the MAIN control register's (xxxx_MAIN_CR) content to the ALTERNATE control register's (xxxx_ALT_CR) content and vice versa (see [\[2\]](#) for details).

Before entering Low-power mode, the STM32MP13x must set the STPMIC1D ALTERNATE control registers in line with the STPMIC1D regulator settings for the required Low-power mode behavior. If needed, the STM32MP13x must set the STPMIC1D MAIN control registers to guarantee that the application leaves Low-power mode.

5.3.1 LPLV-Stop mode

The application LPLV-Stop mode sequence is shown in [Figure 6](#) according to the implementation shown in [Figure 1](#).

1. The application is powered up and is working in Run operating mode; the STPMIC1D is in POWER_ON MAIN state.
2. When the LPLV-Stop operating mode is requested, the software prepares the LPLV-Stop entry process:
 - a. STM32MP13x settings such as: stopping some clocks, setting DDR to self-refresh, setting PWRLP_TEMPO.
 - b. STPMIC1D settings:
 - BUCK1 (V_{DDCPU}), BUCK4 (V_{DDCORE}): Run mode value in main mode and LPLV-Stop value in alternate mode. Refer to the datasheet for Run mode and LPLV-Stop mode values.
 - Other BUCKs and LDOs keep the same Run mode value in main and alternate mode.
3. The STM32MP13x sets the LPDS and LVDS bits of the PWR_CR1 register to prepare for entering LPLV-Stop: the PWR_ON signal is deasserted when the STM32MP13x enters LPLV-Stop. The STPMIC1D goes to POWER_ON ALTERNATE state:
 - BUCK1 and BUCK4 voltages decrease to LPLV-Stop mode value.
4. On a wake-up event, the STM32MP13x leaves LPLV-Stop mode and asserts the PWR_ON signal:
 - a. The STM32MP13x timer $t_{SEL_VDDCORETEMPO}$ is started to allow V_{DDCORE} and V_{DDCPU} to reach the operating voltage level of the Run mode .

Note: $t_{SEL_VDDCORETEMPO} = 234 \mu s$ min (see [\[5\]](#)).

- b. The STPMIC1D goes to POWER_ON MAIN state:
 - BUCK1 voltage rises from LPLV-Stop mode value to Run mode value. In 152 us max for V_{DDCPU} to go from 0.9 V LPLV-Stop to 1.25 V Run mode
 - BUCK4 voltage rise from LPLV-Stop mode value to Run mode value. In 184 us max for V_{DDCORE} to go from 0.9 V LPLV-Stop to 1.25 V Run mode

Note: *STPMIC1D BUCK1 has 2.3 mV/ μs min slew rate.*

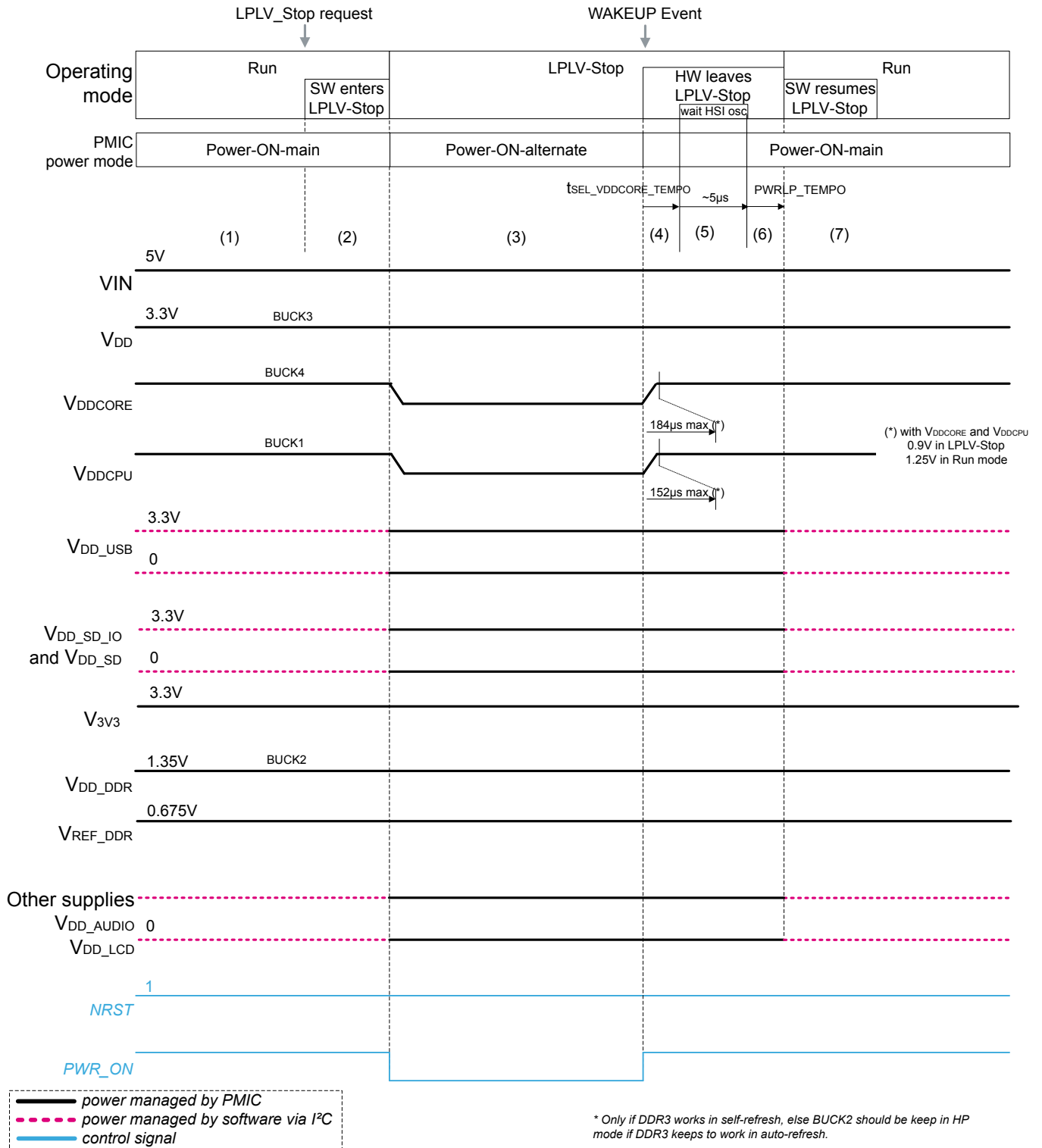
Note: *STPMIC1D BUCK4 has 1.9 mV/ μs min slew rate*

5. Once $t_{SEL_VDDCORETEMPO}$ has elapsed, a clock restore process is performed in the STM32MP13x.

6. Once the STM32MP13x HSI clock oscillator is stable ($\sim 5 \mu\text{s}$), the PWRLP_TEMPO timer is started waiting for the peripherals to be stable.

Note: PWRLP_TEMPO is an STM32MP13x dedicated timer designed to wait for the regulator recovery when the application goes from Low-power mode to Run mode. When using the STPMIC1D, the PWRLP_TEMPO delay can be set to minimum as the $t_{\text{SEL_VDDCORETEMPO}}$ duration is higher than the STPMIC1D total recovery time. However, if an application goes to LP_Stop (instead of LPLV_Stop), the STM32MP13x does not wait for the $t_{\text{SEL_VDDCORETEMPO}}$ delay. So, PWRLP_TEMPO must be set to $100 \mu\text{s}$ to let the STPMIC1D regulators recover from LP to HP mode.

7. When PWRLP_TEMPO elapses, the application goes into Run mode. The software resumes normal operations (such as restoring clocks and restoring DDR from self-refresh).

Figure 6. LPLV-stop mode sequence


5.3.2 LPLV-Stop2 mode

The application LPLV-Stop2 mode sequence is shown in [Figure 7](#) according to the implementation shown in [Figure 1](#).

Note: *LPLV-Stop2 mode is not possible when V_{DDCORE} and V_{DDCPU} share the same supply, as in [Figure 2](#).*

1. The application is powered up and working in Run operating mode; the STPMIC1D is in POWER_ON MAIN state.
2. When the LPLV-Stop2 operating mode is requested, the software prepares the LPLV-Stop2 entry process:
 - a. STM32MP13x settings such as: stopping some clocks, setting DDR to self-refresh, setting PWRLP_TEMPO.
 - b. STPMIC1D settings:
 - BUCK4 (V_{DDCORE}): 1.25 V in main mode, 0.9 V in alternate mode.
 - BUCK1 (V_{DDCPU}): 1.25 V in main mode, OFF in alternate mode.
 - Other BUCKs and LDOs keep the same Run mode value in main and alternate mode.
3. The STM32MP13x sets the LPDS and LVDS bits of the PWR_CR1 register to prepare for entering LPLV-Stop2: the PWR_ON signal is deasserted when the STM32MP13x enters LPLV-Stop2. The STPMIC1D goes to POWER_ON ALTERNATE state:
 - BUCK4 voltage decreases to 0.9 V
 - BUCK1 regulator is powered OFF
4. On a wake-up event, the STM32MP13x leaves LPLV-Stop2 mode and asserts a PWR_ON signal:
 - a. The STM32MP13x timer $t_{SEL_VDDCORETEMPO}$ is started to allow V_{DDCORE} to reach the operating voltage level of the Run mode .

Note: $t_{SEL_VDDCORETEMPO} = 234 \mu s \text{ min}$ (see [\[5\]](#)).

- b. The STPMIC1D goes to POWER_ON MAIN state:
 - BUCK1 voltage rises from 0 V to 1.25 V (in 100 μs to 1ms setup time + 152 μs rise time max)
 - BUCK4 voltage rise from 0.9 V to 1.25 V (in 184 μs max)

Note: *STPMIC1D BUCK1 has 2.3 mV/ μs min slew rate plus setup time between 100 μs to 1 ms.*

Note: *STPMIC1D BUCK4 has 1.9 mV/ μs min slew rate.*

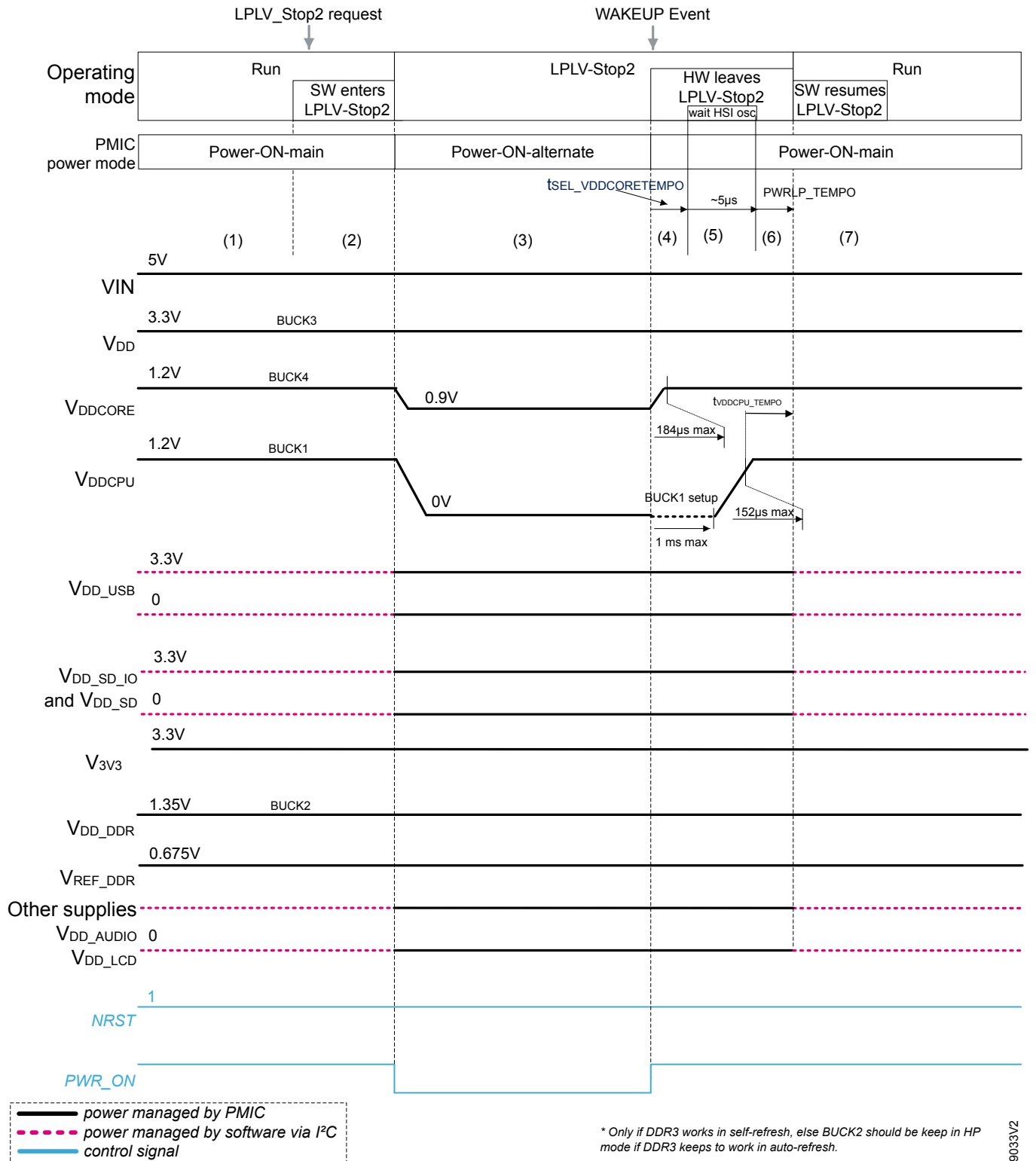
5. Once the $t_{SEL_VDDCORETEMPO}$ has elapsed, a clock restore process is performed in the STM32MP13x.
6. Once the STM32MP13x HSI clock oscillator is stable ($\sim 5 \mu s$), the PWRLP_TEMPO timer is started waiting for the peripheral supplies stabilize. In parallel, the BUCK1 voltage (V_{DDCPU}) increases until it reaches V_{TH_VDDCPU} . At this time, `vddcpu_ok` is set internally and the t_{VDDCPU_TEMPO} is started allowing V_{DDCPU} to reach its minimum Run mode operating voltage (see [\[6\]](#)) during this time.

Note: *PWRLP_TEMPO is an STM32MP13x dedicated timer designed to wait for the regulator recovery when the application goes from Low-power mode to Run mode.*

7. When both PWRLP_TEMPO and t_{VDDCPU_TEMPO} elapse, the application goes into Run mode. The software resumes normal operation (such as restoring clocks and restoring DDR from self-refresh).

Attention: *On wake-up from a very short LPLV-Stop2 transition, the regulator output voltage may remain high enough so that Run mode starts rapidly while the regulator current drive may not have enough time to reach its nominal value. This results in a voltage drop as soon as Run mode requires significant current. Hence, PWRLP_TEMPO must be increased to allow the regulator to recover correctly.*

A PWRLP_TEMPO of around 2ms (PWRLP_DLY[21:0] = 0x20000) is recommended.

Figure 7. LPLV-Stop2 mode sequence


5.3.3 Standby mode

The application Standby mode sequence is shown in Figure 8 according to the implementation shown in Figure 1. In this application, the boot Flash memory voltage (V_{3V3}) is present when leaving Standby mode (as V_{3V3} SMPS is controlled from V_{DDCORE}). The STM32MP13x is able to read the boot software (FSBL). In Standby mode, DDR3 memory is in self-refresh.

1. The application is powered up and is operating in Run mode; the STPMIC1D is in POWER_ON MAIN state.
2. When Standby mode is requested, the software prepares to enter standby by changing:
 - a. The STM32MP13x settings such as:
 - Stopping certain clocks
 - Sets DDR in self-refresh
 - Sets POPL⁽¹⁾ and EADLY timers, and so on
 - b. The STPMIC1D settings:
 - i. BUCK1 (V_{DDCPU}), BUCK4 (V_{DDCORE}): ON in main mode, OFF in alternate mode. The software must set BUCK1 (V_{DDCPU}) and BUCK4 (V_{DDCORE}) to the minimum Run mode operating voltage (refer to the datasheet [6])
 - ii. LDO4 (V_{DD_USB}), LDO5 ($V_{DD_SD_IO}$), LDO1 (V_{DD_AUDIO}), LDO6 (V_{DD_LCD}): OFF in alternate mode. In main mode, it is the software to set at runtime.
 - iii. Other BUCKs and LDOs keep the same Run mode value in main and alternate mode.
3. The STM32MP13x resets the LPDS and LVDS bits to wait while entering Standby mode: PWR_ON signal is deasserted when the STM32MP13x enters standby:
 - a. The POPL timer is started to prevent the STM32MP13x leaving standby before POPL elapses.
 - b. The STPMIC1D goes in POWER_ON ALTERNATE state:
 - i. BUCK1 (V_{DDCPU}), BUCK4 (V_{DDCORE}): regulators are powered OFF
 - ii. LDO4 (V_{DD_USB}), LDO5 ($V_{DD_SD_IO}$), LDO1 (V_{DD_AUDIO}), LDO6 (V_{DD_LCD}) are powered OFF
4. On a wake-up event, the STM32MP13x leaves Standby mode and asserts a PWR_ON signal:
 - a. The STPMIC1D goes in POWER_ON MAIN state:
 - i. BUCK1 (V_{DDCPU}), BUCK4 (V_{DDCORE}): regulators are powered ON
 - ii. Other regulators go back to the main state as programmed by software before entering Standby mode
 - b. When the V_{DDCORE} voltage goes above the $V_{TH_VDDCORE}$ threshold, $t_{VDDCORE_TEMPO}$ is started. As long as the $t_{VDDCORE_TEMPO}$ timer has not elapsed, the STM32MP13x is kept in reset internally.

*Note: The STM32MP13x internal voltage threshold, $V_{TH_VDDCORE}$ rising edge, is 0.95 V min.
The STM32MP13x internal delay, $t_{VDDCORE_TEMPO}$ is 200 μ s min.*
5. When $t_{VDDCORE_TEMPO}$ elapses, the STM32MP13x is taken out of internal reset (V_{DDCORE_OK}):
 - a. V_{DDCORE} voltage is higher than STM32MP13x V_{DDCORE} min operating voltage for boot operation (1.2 V).

Note: BUCK4 has a 1.9 mV/ μ s minimum slew rate. This rate guarantees that a V_{DDCORE} voltage is higher than the STM32MP13x V_{DDCORE} min operating voltage when $t_{VDDCORE_TEMPO}$ elapses.

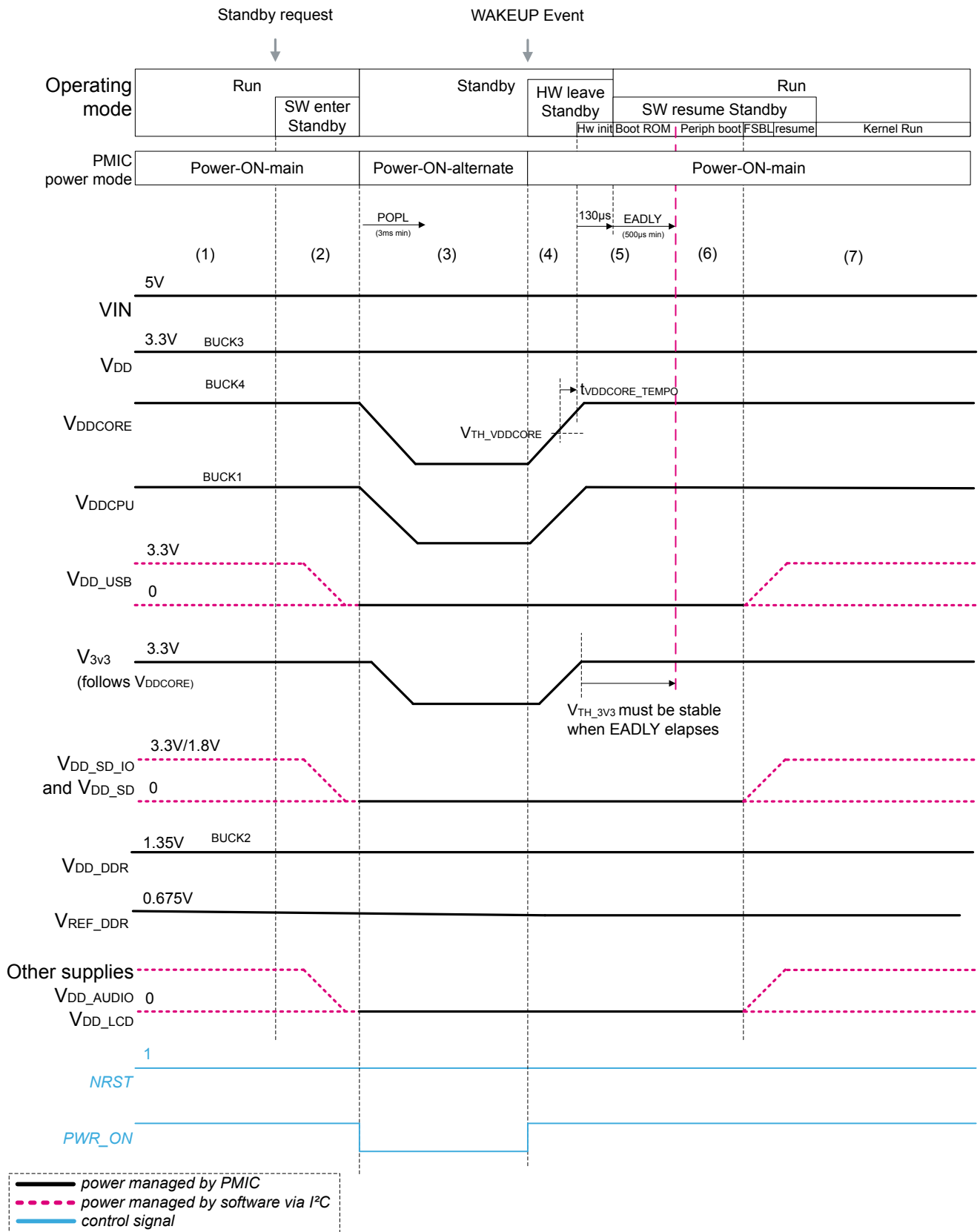
 - b. The STM32MP13x performs internal hardware initialization (enables the HSI and loads OTP settings over a 130 μ s duration).
 - c. When the V_{DDCPU} voltage is above the V_{TH_VDDCPU} threshold, t_{VDDCPU_TEMPO} is started. When t_{VDDCPU_TEMPO} elapses and hardware initialization is done, the device goes into Run mode. The boot ROM executes the EADLY delay
6. When the EADLY delay timer has elapsed, the boot ROM starts accessing external peripherals to load and execute boot software. The external peripherals are the Flash memory. Implicitly, when EADLY has elapsed, the V_{3V3} voltage is stable:
 - a. The boot ROM reads FSBL (first stage bootloader) in the Flash memory. It verifies, and executes the FSBL.
 - b. The software can program the STPMIC1D via the I²C interface to set any regulator at this step.
7. The software detects an “exit from Standby mode”, then resumes and runs the Kernel software. The BUCK1 (V_{DDCPU}) and BUCK4 (V_{DDCORE}) are set to the minimum Run mode operating voltage

- The STM32MP13x POPL timer allows the STM32MP13x to be kept in standby and to assert a PWR_ON signal low for a minimum duration. This action allows the peripheral regulators to stop before restarting them. This is to ensure the peripherals restart properly if a wake-up event occurs just after the application goes into standby. The STPMIC1D has a discharge resistor on each regulator output that allows all of the regulator output voltages to discharge in less than 3 ms. The POPL can be set to a minimum of 3 ms or can be kept with the default value (10 ms) if the wake-up duration from Standby is not critical.*

The STM32MP13x EADLY timer is dedicated to preventing boot ROM performing any access to the boot peripheral before it is ready when recovering from Standby mode. It waits for a stable voltage on the Flash memory to ensure that the boot software is reliably read by the boot ROM. The Flash memory can be the eMMC or the SD-Card.

In this application, V_{3V3} rise time depends on the BUCK_ext characteristics and control so the minimum EADLY must be set accordingly.

In the case where V_{DDCORE} and V_{DDCPU} are merged and provided by BUCK1 (using STPMIC1A), then V_{3V3} can be provided by BUCK4. BUCK4 rises in less than 1.5 ms. The EADLY can be set to 1.5 ms minimum. It can be kept with the default value (10 ms) if the wake up duration from Standby is not critical.

Figure 8. Standby mode sequence


5.4 User reset and crash recovery management

As introduced in [Section 4.2 Control signals and interface between STM32MP13x and PMIC](#), the STM32MP13x and the PMIC both have bidirectional active low reset pins interconnected (see [Figure 1](#) signal NRST).

If an STM32MP13x crash occurs (iwdg1_out_rst or iwdg2_out_rst watchdog elapsing), a reset pulse is generated by the STM32MP13x on NRST signal. The reset pulse is caught by the PMIC which triggers an immediate power cycle sequence: a PMIC power-down sequence followed by a PMIC power-up sequence.

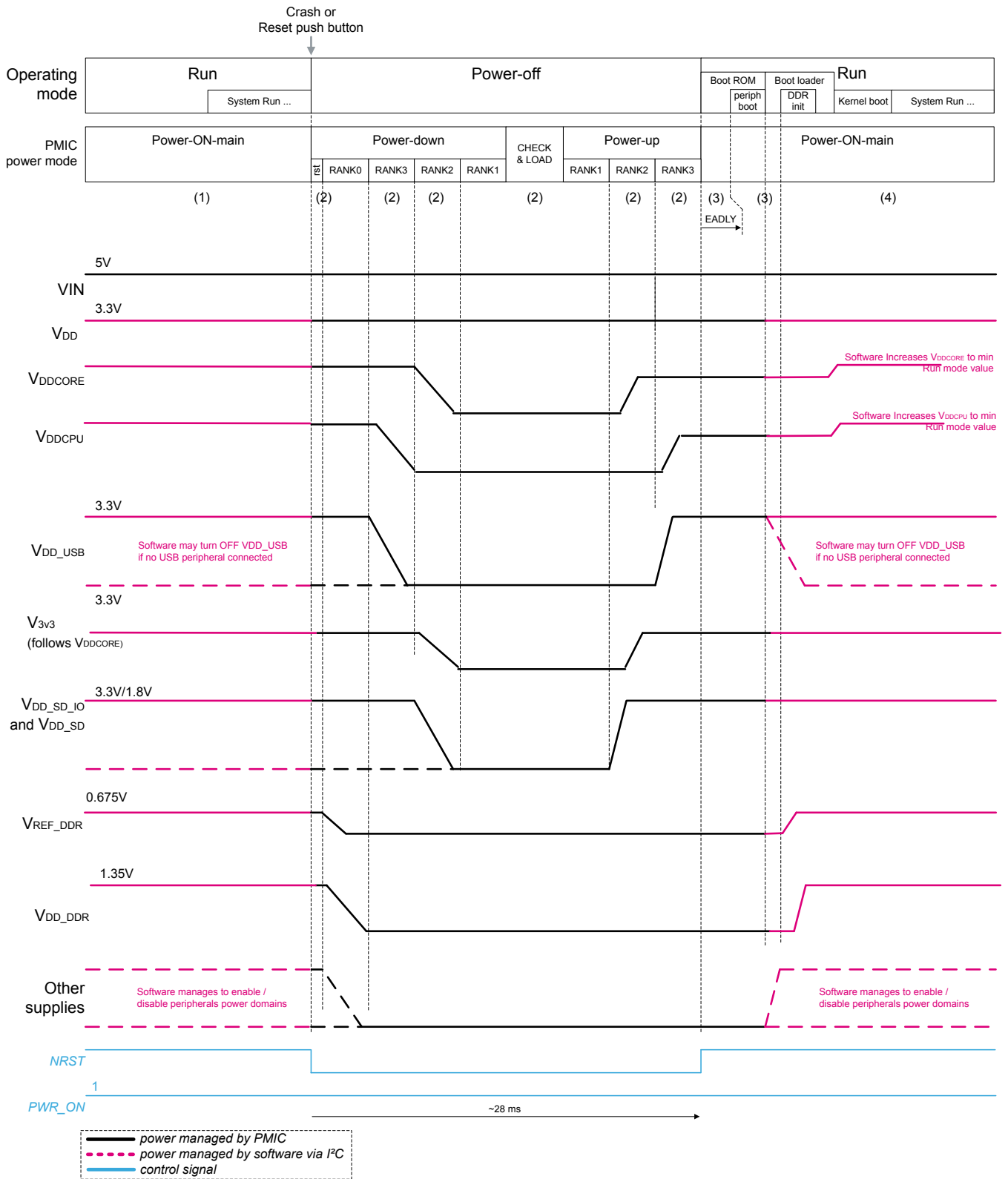
A power cycle allows the peripherals to restart and reset properly after a crash occurs; especially for peripherals that do not have a reset input signal. Power cycling is mainly recommended for peripheral boot devices and Flash memory devices such as: eMMC, NAND, NOR, SD-Card. Power cycling is not performed on the PMIC BUCK3 (VDD) that needs to be kept enabled during reset (see [Section 5.1.2 PMIC restart_request and mask_reset options](#) for details of the PMIC mask_reset option).

If the reset button is pressed by a user, the same power cycle sequence is performed by the PMIC.

5.4.1 Crash recovery management or user reset sequence

The sequence shown in [Figure 9](#) illustrates a crash recovery sequence according to the implementation shown in [Figure 1](#). In this sequence, the crash happens in Run mode (by IWDG reset). Nevertheless, an IWDG reset can occur in all modes, including: Run, Stop, LP-stop, LPLV-Stop, LPLV-Stop2, and Standby modes.

1. The application is powered up and is in Run mode; the PMIC is in Power-ON-main state. A crash occurs (iwdg1_out_rst or iwdg2_out_rst watchdog elapsing) or the user pushes the reset button. The reset button push generates a pulse on the NRST signal.
 2. The PMIC detects the reset assertion (NRST pulse low) and starts a noninterruptible power cycle:
 - a. The PMIC asserts NRST low.
 - b. The PMIC performs a power-down sequence.
 - c. The PMIC checks the conditions to restart (such as VIN, temperature) and reloads the internal NVM.
 - d. The PMIC performs a power-up sequence.
 - e. The PMIC releases NRST⁽¹⁾
 3. The NRST signal rises as the STM32MP13x and the PMIC release their respective reset pins (and reset button released):
 - a. The STM32MP13x EADLY⁽²⁾ delay timer (10 ms) is started.
 - b. When EADLY has elapsed, the boot ROM starts accessing external peripherals to load and execute the bootloader software. The external peripherals can be either an eMMC or an SD-Card, depending on the STM32MP13x boot pins setting.
 - c. The bootloader controls any PMIC regulator (such as to initiate an LCD and plot splash screen).
 4. The bootloader initializes the DDR then loads and executes the Kernel:
 - a. BUCK2 (V_{DD_DDR}) is enabled at 1.35 V
 - b. DDR_VREF (V_{REF_DDR}) is enabled.
 - c. Software waits for at least 1.4 ms for BUCK2 ready
 - d. The software initializes the STM32MP13x DDR controller and DDR3 device.
 - e. The bootloader loads and executes the kernel and the kernel initializes.
 - f. The software must set BUCK1 (V_{DDCPU}) and BUCK4 (V_{DDCORE}) if applicable, to the minimum Run mode operating voltage. Refer to the datasheet for more details. System is running.
1. *If the reset signal (NRST) is still asserted at this step (for example the user is still pressing the reset button), the PMIC waits for the reset signal to be released before rearming the reset circuit. This is to avoid the PMIC repeating a power cycle loop. PMIC power cycle duration is ~28 ms.*
 2. *The EADLY timer is dedicated to preventing boot ROM performing any access to the boot peripheral before it is ready. Typically waiting for a stable voltage of Flash memory (eMMC or SD-card) to ensure that the boot software is reliably read by the boot ROM. Default value after reset is 10 ms delay. (see [\[5\]](#) for details).*

Figure 9. Crash recovery sequence


5.5 Software management example

This section presents one possible software integration strategy of the PMIC management by the STM32MP13x. The OpenSTLinux software distribution integrates the way the PMIC regulators are driven and configured by the STM32MP13x to match the operating mode presented in [Section 5.1 Operating modes](#).

To summarize the main OpenSTLinux integration points:

- Interface with the PMIC is performed in the low level “secure monitor” part of the boot chain split between FSBL (for example: TF-A) and SSBL (for example: U-Boot).
- The PMIC power management strategy is presented in [Table 5](#). It is configured in the secure monitor DTS file (Typ TF-A). This strategy uses the Linux regulator framework-binding terminology.
- Each PMIC power source is seen as a “regulator” on which the software application and driver registers as a “consumer”. Typically, a regulator is enabled when it is requested by one consumer. The exception is made for the core supply. The core supply must be kept alive whatever the consumer registration state and keep the “always-on” option.

To learn more about the power management function in OpenSTLinux, refer to the following online user guide articles:

https://wiki.st.com/stm32mpu/wiki/Power_overview

https://wiki.st.com/stm32mpu/wiki/Regulator_overview .

Also refer to the Linux[®] Kernel binding documentation for the regulator framework.

Table 5. PMIC power management options

Power state						
Application power domain	PMIC supply source	Default Power On state (NVM)	STM32MP13x = Run/Stop PMIC= POWER_ON Main	STM32MP13x = LPLV-stop or LPLV-Stop2 PMIC= POWER_ON alternate	STM32MP13x = STANDBY DDR OFF PMIC= POWER_ON alternate	Options/ Comments
V _{DD}	BUCK3	3.3 V	3.3 V/always-on ⁽¹⁾	3.3 V	3.3 V	Mask_reset ⁽²⁾ , overcurrent protection ⁽³⁾
V _{DDCORE}	BUCK4	1.2 V	Typ Run value (refer to datasheet)/ always-on	0.9 V	Off	Overcurrent protection ⁽⁴⁾
V _{DDCPU}	BUCK1	1.2 V	Typ Run value (refer to datasheet)/ always-on	0.9 V (LPLV-Stop) or Off (LPLV-Stop2)	Off	Overcurrent protection
V _{DD_USB}	LDO4	3.3 V	3.3 V/consumer driven ⁽⁵⁾	3.3 V/consumer driven	Off	-
V _{DD_DDR}	BUCK2	Off	1.35 V/always-on	1.35 V	Off	Overcurrent protection
V _{REF_DDR}	DDR_REF	Off	0.675 V/always-on	0.675 V	Off	-
V _{DD_SD_IO}	LDO5	3.3 V	3.3 V/1.8 V always on/ consumer driven	3.3 V/1.8 V always ON	Off ⁽⁶⁾ boot on ⁽⁷⁾	Can be decreased to 1.8 V in UHS-I mode
V _{DD_SD}	LDO2	Off	3.3 V	3.3 V	Off	Must be set after boot sequence by software
VBST	BOOST	Off	ON/consumer driven	ON/consumer driven	Off	Pure software. No alternate registers
VBUS_SW1	PWR_USB_SW	Off	ON/consumer driven	ON/consumer driven	Off	
VBUS_SW2	PWR_SW	Off	ON/consumer driven	ON/consumer driven	Off	
V _{DD_AUDIO}	LDO1	Off	1.8 V/consumer driven	1.8 V/consumer driven	Off	-
V _{DD_LCD}	LDO6	Off	1.8 V/consumer driven	1.8 V/consumer driven	Off	-
V _{3V3}	BUCK_ext	3.3 V	3.3 V	3.3 V	Off	Controlled by V _{DDCORE} ⁽⁸⁾

1. Always-on: Keeps the core voltage on even if there is no software consumer.
2. Mask_reset: Specify the PMIC mask_reset option to this regulator not to be impacted by a reset power cycle. (see Section 5.1.2 PMIC restart_request and mask_reset options)
3. Overcurrent protection: Specify the PMIC option OCPOFF on this regulator. Overcurrent detection leads to a PMIC shutdown. Refer to [2]
4. Only if V_{DDCORE} and V_{DDCPU} are not merged using STPMIC1D like on Figure 1
5. Consumer driven: Linux driver turns the regulator on/off following consumer demand. When entering low-power mode, the last Run status is applied (Main mode duplicate in Alternate mode). The user has to consider the required status before entering Low-power mode. For example, when the powered peripheral is set as wake up source.
6. V_{DD_SD_IO} is connected to the STM32MP13x V_{DDSD1} power supply. It cannot be switched off if any GPIO is above 3.6 V (typically 5 V tolerant IOs).
7. Boot on: Software must set this regulator on in POWER_ON Main mode before entering into Low-power mode (switch to PMIC Alternate mode). This is needed to turn-on immediately on wake up (switch back to Main mode).
8. Or BUCK4 if V_{DDCORE} connected to V_{DD_CPU} using STPMIC1A.

6 STM32MP13xD and STM32MP13xF enhanced CPU frequency supply management

The STM32MP13xD and STM32MP13xF devices have an enhanced consumer mission profile ([7]). This profile allows the Arm® Cortex®-A7 CPU to run at a higher clock frequency ([6] for details and limitations).

Accordingly, the V_{DDCPU} supply voltage must be increased when the CPU frequency (F_{mpuss_ck}) operates above 650 MHz. Refer to the datasheet for V_{DDCPU} Run Overdrive minimum voltage value. When it does not operate in Run mode above 650 MHz, the V_{DDCPU} supply voltage should be set back to its nominal Run mode voltage.

The V_{DDCPU} voltage is increased by setting BUCK1 of the PMIC to the desired voltage value.

When going from Run mode to Run Overdrive mode above 650 MHz, V_{DDCPU} must be increased before the frequency.

When going from Run Overdrive mode above 650 MHz back to Run mode, the frequency must be decreased before the voltage.

The change of voltage is done by setting the BUCK1_MAIN_CR register to the desired value. The setting is done through the I2C software programming.

When Overdrive mode is needed, the voltage change on the STPMIC1D must be controlled. Also, the MPU_RAM_LOWSPEED bit in the STM32MP13x PWR_CR1 register must be managed by software.

The MPU_RAM_LOWSPEED bit must reset by respecting the two conditions below:

- after the V_{DDCPU} supply has reached the overdrive mode voltage range
- before increasing the MPU frequency into the overdrive frequency range.

The MPU_RAM_LOWSPEED bit must set by respecting the two conditions below:

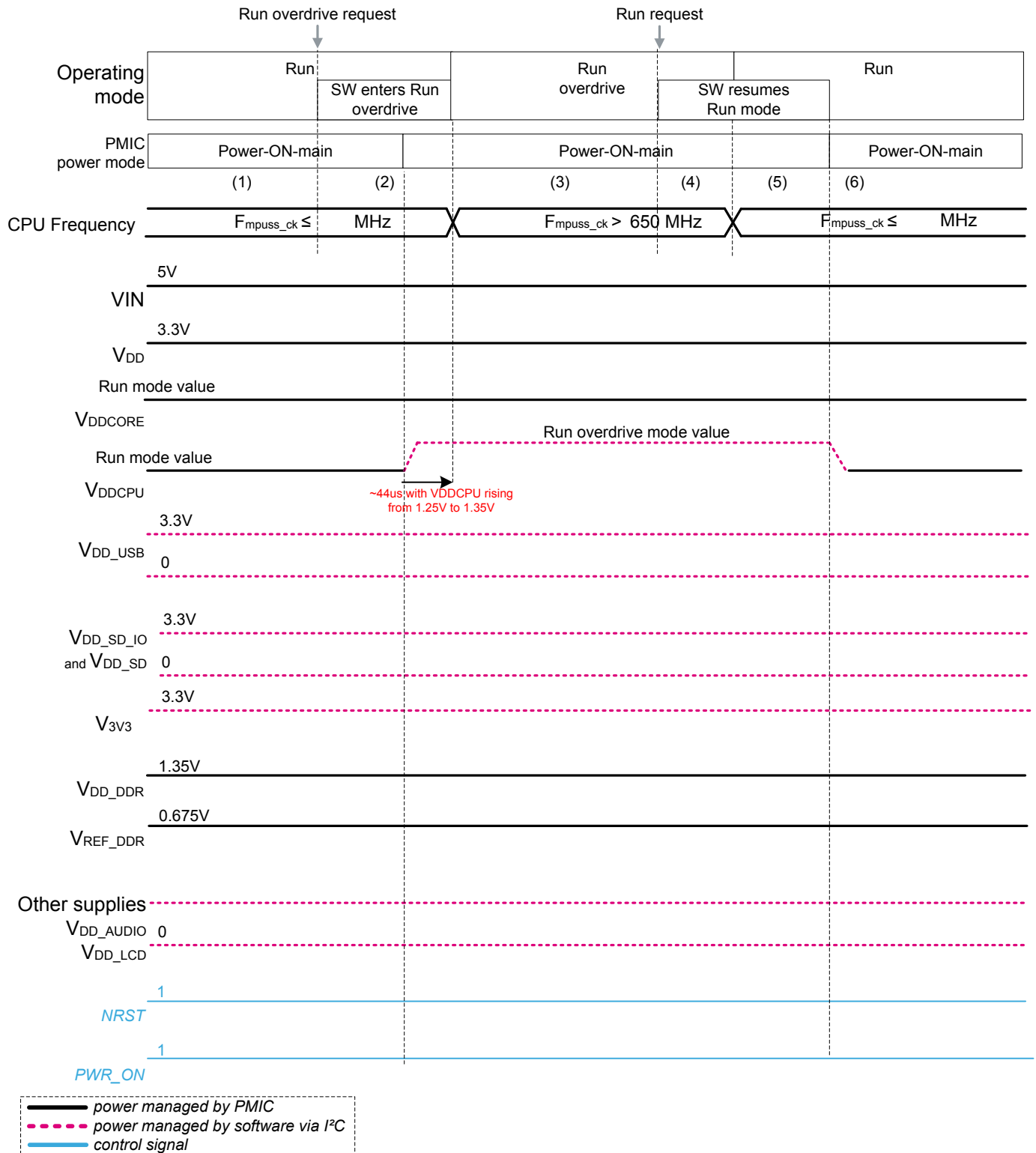
- after decreasing the MPU frequency into the standard frequency range
- before decreasing the MPU voltage below the overdrive mode voltage range.

The application Run Overdrive mode sequence is shown in Figure 10 according to the implementation shown in Figure 1.

1. The application is operating in Run mode with the CPU frequency below 650 MHz; the STPMIC1D is in POWER_ON MAIN state.
2. When Run Overdrive mode is requested, the software prepares to enter Run Overdrive mode. It changes the V_{DDCPU} voltage level through I2C programming.
3. Once V_{DDCPU} has reached the Run Overdrive voltage, the software can reset the MPU_RAM_LOWSPEED bit. After this, it can increase the CPU frequency above 650 MHz. The system is now in Run Overdrive mode above 650 MHz.

Note: STPMIC1D BUCK1 has 2.3 mV/μs min slew rate.

4. When Run mode is requested, the software prepares to resume Run mode. It changes the CPU frequency to below 650 MHz, then it sets the MPU_RAM_LOWSPEED bit.
5. The software decreases the V_{DDCPU} voltage level through I2C programming.
6. The system is now in Run mode below 650 MHz.

Figure 10. Example with IOs at 3.3 V, DDR3L, and V_{DDCORE} voltage scaling


Revision history

Table 6. Document revision history

Date	Version	Changes
20-Jun-2023	1	Initial release.

Contents

1	General information	2
2	Overview	3
2.1	Reference documents	3
3	Glossary	4
4	Wall adapter supply application reference design	5
4.1	Power distribution	8
4.1.1	V _{DD} power domain (3.3 V)	8
4.1.2	VDDCORE power domain	8
4.1.3	VDDCPU power domain	8
4.1.4	VDD_USB power domain (3.3 V)	10
4.1.5	VDD_DDR (1.35 V), VREF_DDR (0.675 V) power domains	10
4.1.6	V3V3 power domain (3.3 V)	10
4.1.7	SD-Card power (VDD_SD)	11
4.1.8	VBUS_SW1 and VBUS_SW2 power domain (5.2 V)	13
4.1.9	VDD_AUDIO (1.8 V), VDD_LCD (2.8 V) power domains	15
4.2	Control signals and interface between STM32MP13x and PMIC	15
5	Power management	17
5.1	Operating modes	17
5.1.1	Application turn-on/turn-off conditions	18
5.1.2	PMIC restart_request and mask_reset options	19
5.2	Application power-up/power-down sequence	20
5.2.1	Power-up by main supply (VIN) plugin	20
5.2.2	Power-up from the STPMIC1D Off mode	23
5.2.3	Power-down by battery removal	23
5.3	Low-power mode management	24
5.3.1	LPLV-Stop mode	24
5.3.2	LPLV-Stop2 mode	27
5.3.3	Standby mode	29
5.4	User reset and crash recovery management	32
5.4.1	Crash recovery management or user reset sequence	32
5.5	Software management example	34
6	STM32MP13xD and STM32MP13xF enhanced CPU frequency supply management	36
	Revision history	38

List of figures

Figure 1.	STM32MP13x and STPMIC1D with DDR3L, boot flash, SD-Card, and 2x USB HS	5
Figure 2.	Low-cost version V_{DDCPU} merged with V_{DDCORE} using STPMIC1A	7
Figure 3.	V_{BUS_SW1} , V_{BUS_SW2} power path in USB host mode	13
Figure 4.	Flashing through USB with Type-A connector	14
Figure 5.	Power-up/power-down sequence	22
Figure 6.	LPLV-stop mode sequence	26
Figure 7.	LPLV-Stop2 mode sequence	28
Figure 8.	Standby mode sequence	31
Figure 9.	Crash recovery sequence	33
Figure 10.	Example with IOs at 3.3 V, DDR3L, and V_{DDCORE} voltage scaling	37

List of tables

Table 1.	Device summary	1
Table 2.	Reference documents	3
Table 3.	Glossary	4
Table 4.	Application operating modes	18
Table 5.	PMIC power management options	35
Table 6.	Document revision history	38

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

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

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

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

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

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

© 2023 STMicroelectronics – All rights reserved