Introduction to STM32MP13x product lines using low-power modes

Introduction

STM32MP13x product lines are built on an Arm® Cortex®-A7 with a single-core MPU subsystem.

These devices can be configured in various low-power modes to reduce power consumption when necessary.

This application note explains the various low-power modes of STM32MP13x devices, how to configure and how to exit from these modes. This document gives guidelines on how to use low-power modes at the system level. It also presents guidelines when using an external STPMIC1x power-regulator component.

For further information on STM32MP13x devices, refer to the following documents and deliverables available on www.st.com.

- STM32MP13x product line reference manuals
- STM32MP13x product line datasheets
- STPMIC1x datasheet
- *Getting Started with STM32MP13 Series hardware development* (AN5474)
- *STM32MP13x MPU product lines and STPMIC1D / STPMIC1A integration on a wall adapter supply* (AN5587)
- STM32MP13x product line embedded software

Table 1. Applicable products

Type	Product lines
Microcontrollers	STM32MP131, STM32MP133, STM32MP135

1 Overview

This document applies to all STM32MP13x product lines. The table below describes the main characteristics of all the incumbent STM32MP13x devices. In this document, the Cortex®-A7 is called MPU. The present document assumes a full-featured device such as the STM32MP135 microcontroller.

Table 2. Configuration of the STM32MP13x product lines

Lines	Reference manual	Display camera	CAN FD	Ethernet	ADC
STM32MP131	RM0475	No	No	x1	x1
STM32MP133		No	x2	x2	x2
STM32MP135		Yes	x2	x2	x2

In this document, the MPU subsystem is sometimes referred to as *MPU* to facilitate the reading of the document.

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

2 Glossary

The following table contains a non-exhaustive list of terms used in this document.

Table 3. Glossary

Term	Meaning
AHB	Advanced high-performance bus
AVD	Analog voltage detector
BKPSRAM	Backup SRAM
BOR	Brownout reset
BUCK	Step-down switched-mode power-supply converter
CSS	Clock security system
DDR	Double data rate (SRAM)
DDRCTRL	DDR controller
DDRPHYC	DDR physical interface control
ETH	Ethernet controller
FDCAN	Controller area network with flexible data rate. Could also support time-triggered CAN (TT)
GPIO	General-purpose input/output
HDP	Hardware debug port
IRQ	Interrupt request
IWDG	Independent watchdog
LDO	Low dropout regulator
LpDDR	Low-power DDR
LSE	Low-speed external quartz oscillator
LSI	Low-speed internal oscillator
MCU	Microcontroller
MLAHB	Multilayer AHB/AHB-based interconnect
MPU	Microprocessor
PLL	Phase-locked loop
PVD	Programmable voltage detector
PWR	Power control block
PSCI	Power state and coordination interface
QSPI	Quad data lanes serial peripheral interface
RCC	Reset and clock control
RTC	Real time clock
SDMMC	Secure digital and multimedia card interface. Supports SD, MMC, eMMC, and SDIO protocols
SRAM	Static random access memory
STPMIC1x	Power management-integrated circuit. An external circuit that provides various platform power supplies with large controllability through signals and serial interface. STPMIC1D (for 3.3 V VDD application) and STPMIC1E (for 1.8 V VDD application) are available on STM32MP13x product lines.

Term	Meaning
SYSRAM	System SRAM
SW	Software
TEMP	Temperature sensor
TEMPH-L	Temperature sensor high-low monitoring
TF-A	Trusted firmware for Cortex®-A
USART	Universal synchronous/asynchronous receiver transmitter
USB OTG	Universal serial bus (USB) on-the-go (OTG). A standard USB interface able to become a host or device
VBATH-L	VBAT high-low monitoring
VTT	DDR termination resistance power-supply
WFE	Wait for event
WFI	Wait for interrupt

3 Power management concept

This section describes the high-level power management concept of STM32MP13x product line devices. Refer to the corresponding reference manual for more details.

3.1 STM32MP13x product lines system architecture

The architecture of the power modes on the STM32MP13x product lines is derived from the architecture of the STM32MP15x product lines. The main power management differences between STM32MP13x and STM32MP15x devices are:

- STM32MP13x product lines embed a single Cortex®-A7 MPU while STM32MP15x product lines embed a single/dual Cortex®-A7 MPU and a Cortex®-M4 MCU.
- STM32MP13x product lines feature an additional V_{DDCPU} power supply dedicated to Cortex®-A7 MPU. This allows a new LPLV-Stop2 power mode with V_{DDCPU} off and V_{DDCORE} reduced.
- STM32MP13x product lines feature additional wake-up sources from LPLV-Stop/LPLV-Stop2 low-power modes (U(S)ARTx, I^2Cx, SPIx).

Refer to *STM32MP15x product lines using low-power modes* application note (AN5109) for more details on STM32MP15x devices.

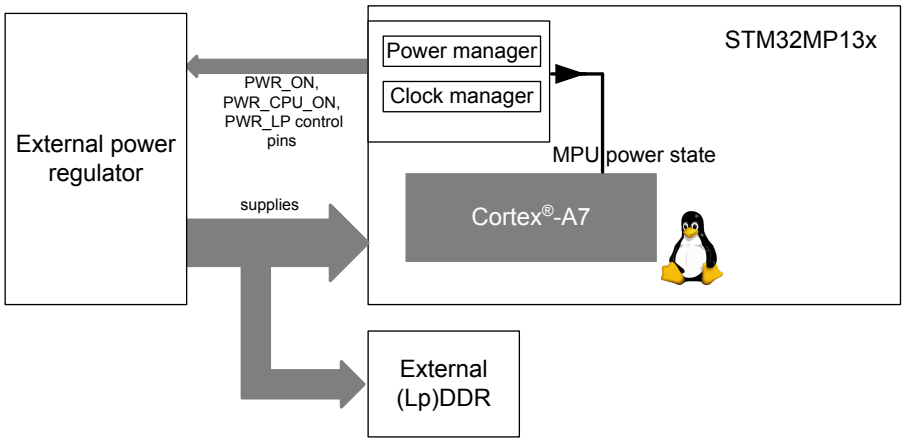
Most STM32 MCUs have an internal LDO (low dropout regulator) to supply the internal digital logic. In STM32MP13x product lines, the internal digital logic is supplied externally. This key difference results in different power supply management for the STM32MP13x product lines. These devices require the use of external signals or they require programming the external regulator (or both), to control the desired voltage supplies.

The power management features are spread between the RCC (reset and clock control) and the PWR (power) blocks of the STM32MP13x product lines device.

- The RCC block ensures the clock tree handling (PLLs, muxes, dividers, clock gating) and the resets (local resets of the peripherals, Cortex®-A7 reset, platform reset).
- The RCC block permits the selection of the power mode based on the power states of the MPU subsystem.
- The PWR block is responsible for low-power entry/exit. It drives the control pins (PWR_ON, PWR_CPU_ON, PWR_LP) to the external regulator based on the power mode.

Figure 1 shows the high-level system architecture of STM32MP13x product lines.

Figure 1. STM32MP13x product lines high-level system architecture



3.1.1 System supplies (V_{DD}, V_{DDCPU}, and V_{DDCORE})

STM32MP13x product line devices require several power supplies (for details refer to the corresponding datasheet and reference manual). Among these power supplies, V_{DD}, V_{DDCPU}, and V_{DDCORE} play a key role in the low-power mode configuration.

- V_{DD}: power supply input for IOs and system analog such as reset, power management oscillators, and PLLs.
- V_{DDCPU}: MPU subsystem digital supply
- V_{DDCORE}: digital core domain supply.
- V_{DD}: must be present before V_{DDCPU} and V_{DDCORE}.

The various power pins of STM32MP13x product lines devices can be supplied either by some external discrete supplies or by using STPMIC1x. The STPMIC1x usage for this purpose is detailed in the reference design section of the application note *Getting Started with STM32MP13 Series hardware development* (AN5474).

The STPMIC1x has several part numbers. The part numbers appropriate for STM32MP13x devices are given below:

- When separated V_{DDCPU} and V_{DDCORE} supplies are required (overdrive OPP)
  - STPMIC1DPQR for application using V_{DD} = 3.3 V
  - STPMIC1EPQR for application using V_{DD} = 1.8 V
- When the same V_{DDCPU} and V_{DDCORE} supplies are required (no-overdrive OPP), the applicable STPMIC1x part numbers are the same as for STM32MP15x product lines devices.
  - STPMIC1APQR for application using V_{DD} = 3.3 V
  - STPMIC1BPQR for application using V_{DD} = 1.8 V

The generic part number "STPMIC1x" is used in the rest of the document. For a specific part number, refer to the above paragraph.

3.1.2 Operating modes description

The operating modes allow the control of the clock distribution towards the different system parts. It also allows the control of the power supplies of the system.

The power management controls the V_{DDCORE} supply following the system operating modes.

The table below presents the operating modes available for the system and the MPU subsystem.

Table 4. Operating modes

-	Power mode	MPU state	Description	Notes
System	Run	CRun/CSleep	V_{DD}, V_{DDCORE}, V_{DDCPU} power on, clock on	-
	Stop	CStop	V_{DD}, V_{DDCORE}, V_{DDCPU} power on, clock off	-
	LP-Stop	CStop	V_{DD}, V_{DDCORE}, V_{DDCPU} power on, clock off	(1)(2)
	LPLV-Stop	CStop	V_{DD} power on. V_{DDCORE} and V_{DDCPU} reduced power-level, and supply load, clock off	(1)
	LPLV-Stop2	CStandby	V_{DD} power ON, V_{DDCORE}, reduced power level, and supply load, V_{DDCPU} power off, clock off	-
	Standby	CStandby	V_{DD} power on. V_{DDCORE}, V_{DDCPU} power off, clock off	-
	VBAT	CStandby	A battery supplies V_{SW}. V_{DD}, V_{DDCORE}, and V_{DDCPU} are off, and RTC clocked by LSE crystal is still active.	(3)
	Power off	CStandby	All power supplies OFF	-
MPU	CRun	-	V_{DDCORE}, V_{DDCPU} power ON, clock ON	-
	CSleep	-	V_{DDCORE}, V_{DDCPU} power on, CPU clock OFF, peripheral clock ON/OFF	(4)
	CStop	-	V_{DDCORE}, V_{DDCPU} power on, CPU subsystem clock off	-
	CStandby	-	V_{DDCORE} power on or off, V_{DDCPU} power off, clock off	-

1. *There is no difference in the output control pins PWR_ON, PWR_CPU_ON, and PWR_LP between LP-Stop, LPLV-Stop, and LPLV-Stop2. While LP-Stop, LPLV-Stop, and LPLV-Stop2 can be activated using STPMIC1x through programming of its internal register before entering the targeted low-power mode, all modes may not be available using other external power regulators.*
2. *The main difference with Stop mode is that the PWR_ON output signal is toggling to 0 when using STPMIC1x external power-regulator, enabling the possibility to take actions like powering off DDR termination resistance power supply.*
3. *To retain the content of the V_{SW} domain (RTC, backup registers, backup RAM, and retention RAM) when V_{DD} is turned off, the VBAT pin can be connected to an optional standby voltage supplied by a battery or from another source.*
4. *The MPU subsystems allocated peripheral(s) clock operate according to RCC PERxLPEN. When the MPU enters CSleep with WFE (wait for event) or WFI (wait for interrupt), the MPU subsystem allocated peripheral(s) clock operates as in MPU CRun mode, irrespective of RCC PERxLPEN.*

It is important to note that VDDCORE=OFF while VDDCPU=ON is strictly forbidden and not supported.

To reach the power-saving target, choose the appropriate low-power mode. The low-power mode to be used depends on the wake-up interrupts and the expected wake-up duration.

The chosen low-power mode must be consistent with the available wake-up sources. For instance, the Standby mode has very limited wake-up source capabilities. Moreover, this mode cannot be used if the wake-up source required is not part of the list of possible wake-up sources in the Standby mode. Refer to the table below.

The wake-up duration is longer if the power-supplies voltage is reduced or switched off by reducing V_{DDCORE} or V_{DDCPU}. It is also longer when switching off the DDR resistance termination supply with the DDR memory in self-refresh or by switching off the full DDR. Switching off the full DDR requires reloading the firmware from the flash memory. The low-power mode wake-up capabilities of the system are shown in the table below.

Table 5. Low-power mode wake-up capabilities of the system

System power mode	Wake-up sources
Stop/LP-Stop	DBG, PVD, AVD, USBH, OTG, CEC, ETH, USARTx, I^2Cx, SPIx, DTS, LPTIMx, GPIOs
LPLV-Stop, LPLV-Stop2	PVD, AVD, USARTx, I^2Cx, SPIx, DTS, LPTIMx, GPIOs
Standby	Six GPIO wake-up pins
All modes	BOR, VBATH/VBATL, TEMPH/TEMPL, LSE CSS, RTC/auto wake-up, tamper pins, IWDGx

Run mode

When in Run mode, the MPU subsystem can be supplied with various V_{DDCPU} values enabling a different operating point (OPP).

By default, the PMIC1X BUCK1 V_{DDCPU} output is set to 1.2 V allowing boot ROM execution, and the software must then set the PMIC1X BUCK1 to the required power-supply level depending on the needed frequency.

The choice of a specific OPP in Run mode does not impact the description provided in the rest of this document regarding low-power mode control features.

Run mode: overdrive OPP

The overdrive OPP requires 1.32 V < V_{DDCPU} < 1.38 V and separated regulators for V_{DDCPU} and V_{DDCORE}.

This operating point allows a higher MPU clock frequency, up to 1000 MHz. However, the mission profile is impacted:

- Overdrive OPP mission profiles:
  - 1000 MHz:
    - 10-year life activity @25% activity rate (21915h)
    - -40 °C < Tj < 105 °C
    - V_{DDCPU_min} = 1.32 V
  - 900 MHz:
    - 10-year life activity @100% activity rate (87660h)
    - -40 °C < Tj < 105 °C
    - V_{DDCPU_min} = 1.32 V
- Non-overdrive OPP mission profile:
  - 650 MHz:
    - 10-year life activity @100% activity rate (87660h)
    - -40 °C < Tj < 125 °C
    - V_{DDCPU_min} = 1.21 V

When overdrive mode is needed, the MPU_RAM_LOWSPEED bit in the PWR_CR1 register must be reset before increasing the MPU frequency into the overdrive frequency range. It must be set after decreasing the MPU frequency into the standard frequency range.

3.1.3 Power strategy

Different configurations can be used to reach the targeted application cost performance. Find below a non-exhaustive list of different possible power strategy configurations. When discrete power regulators are used, it is assumed that only two voltage values are available per regulator.

1. Low-cost application with STPMIC1x
   - Using STPMIC1A/B as on STM32MP15x (common V_{DDCPU} and V_{DDCORE} supply)
   - Up to 650 MHz CPU frequency. No overdrive OPP (common V_{DDCPU} and V_{DDCORE} supply)
   - All low-power modes are possible except for LPLV-Stop2
2. High-speed and low-power modes with STPMIC1x
   - Using STPMIC1D/E (separated V_{DDCPU} and V_{DDCORE} supply).
   - Up to 1000 MHz CPU frequency. Overdrive OPP (separated V_{DDCPU} and V_{DDCORE} supplies).
   - All low-power modes are possible

3. Low-cost application with power-discrete regulators
   - Using optimized discrete power supply (common V_{DDCPU} and V_{DDCORE} supply)
   - Up to 650 MHz CPU frequency. No overdrive OPP (common V_{DDCPU} and V_{DDCORE} supply)
   - Only Stop, LPLV-Stop, Standby, and VBAT low-power modes are possible

V_{DDCORE} voltage	V_{DDCPU} voltage	Mode
V1 (LPLV-Stop value)		LPLV-Stop
V2 (Run value)		Run, Stop

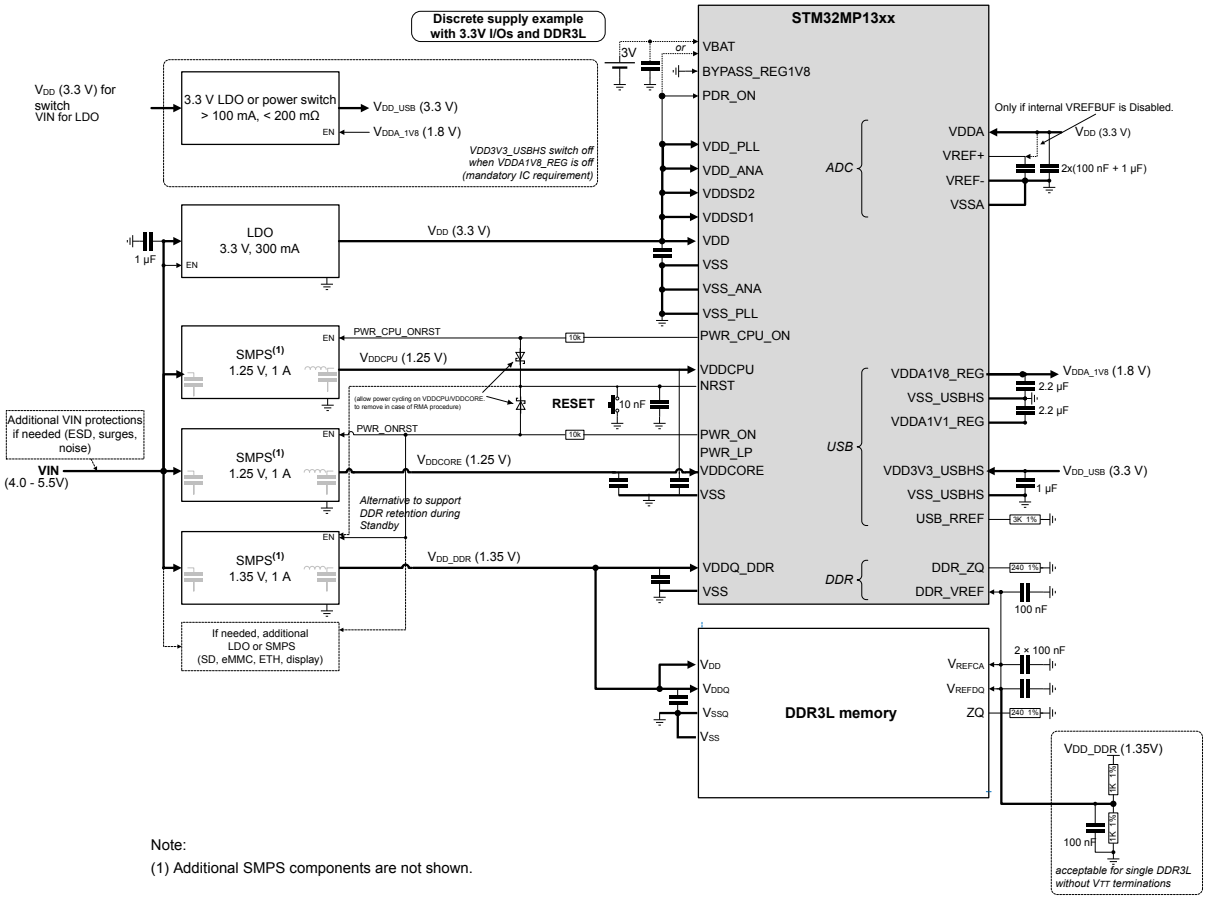
4. High-speed and low-power modes with power-discrete regulators (two OPP in Run mode, only LPLV-Stop2 mode).
   - Using two regulators for V_{DDCPU} and V_{DDCORE}
   - Up to 1000 MHz CPU frequency. Overdrive OPP (separated V_{DDCPU} and V_{DDCORE} supplies)
   - Only LPLV-Stop2, Standby, and VBAT low-power modes are possible
   - Allows low-power mode but it implies a longer wake-up time
   - DDR supply is off during Standby to provide further power gain versus LPLV-Stop2 (at the expense of longer wake-up time since the kernel needs to be reloaded from the flash memory)

V_{DDCORE} voltage	V_{DDCPU} voltage	Mode
V1 (LPLV-Stop value)	0V	LPLV-Stop2
V2 (Run value)	V2 (Run value)	Run, Stop
V2 (Run value)	V3 (Run overdrive value)	Run overdrive

5. High-speed and low-power modes with power-discrete regulators. Overdrive OPP is possible only in Run mode, only LPLV-Stop low-power mode is possible.
   - Using two regulators for V_{DDCPU} and V_{DDCORE}.
   - Up to 1000 MHz CPU frequency. Overdrive OPP.
   - Only LPLV-Stop is possible.
   - DDR is in self-refresh during Standby providing power gain versus LPLV-Stop with reduced wake-up duration compared to DDR supply off.

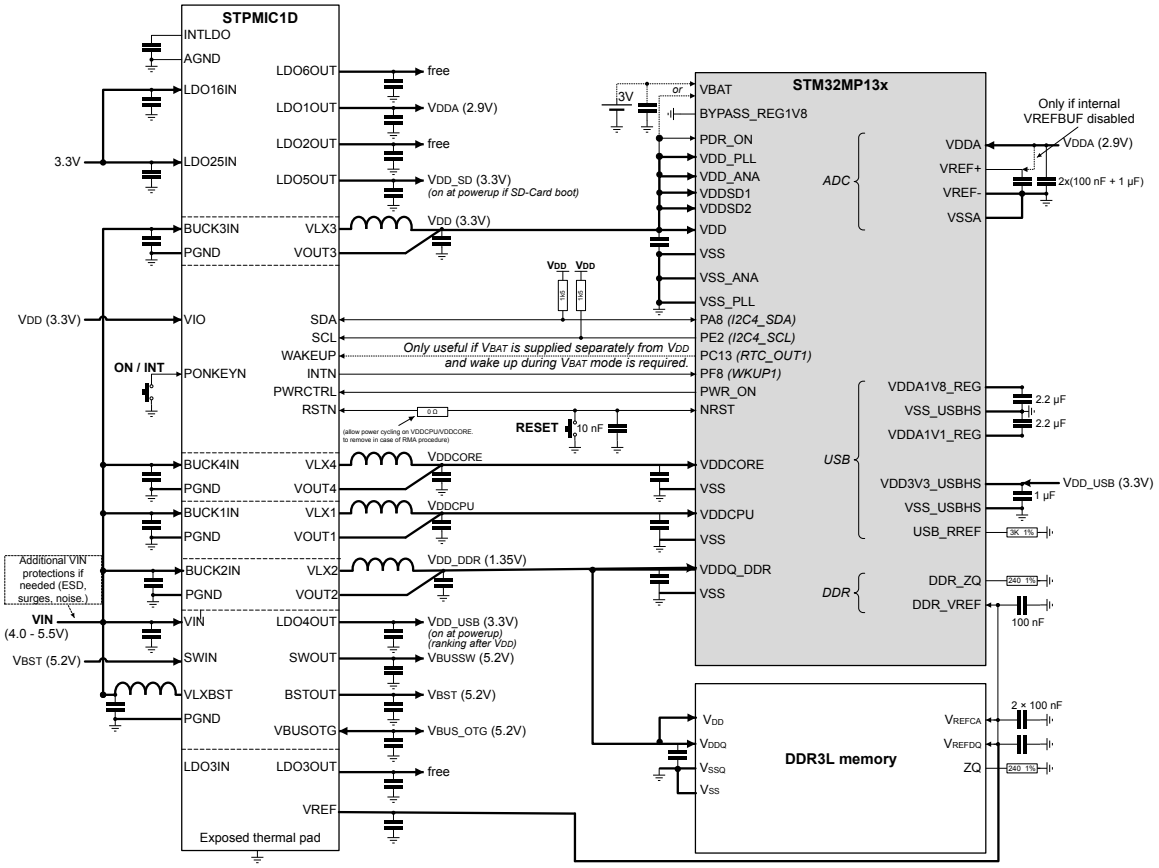
V_{DDCORE} voltage	V_{DDCPU} voltage	Mode
V1 (LPLV-Stop value)	V1 (LPLV-Stop value)	LPLV-Stop
V2 (Run value)	V3 (Run overdrive value)	Run overdrive, Stop

Figure 2. Discrete supply example with 3.3 V I/Os and DDR3L



Note:
(1) Additional SMPS components are not shown.

Figure 3. STPMIC1 supply example 3.3 V I/Os with DDR3L



The STPMIC1x provides power distribution. It is driven by I²C and control signals coming from the STM32MP13x device. The control is done by the Cortex®-A7 MPU subsystem core. The main benefit of using the STPMIC1x instead of using discrete regulator supplies is to have an integrated, smaller, and cheaper solution. Refer to the application note *STM32MP13x MPU product lines and STPMIC1D / STPMIC1A integration on a wall adapter supply* (AN5587) for more details.

Having separated regulators for V_{DDCPU} and V_{DDCORE} on the STPMIC1x may require an additional external buck converter to power the board components.

Example: On the STM32MP13x Discovery board (MB1635, STM32MP135F-DK), an external 3V3 buck converter is used to power the USB port, Ethernet ports, Bluetooth/WiFi, expansion connectors, LCD, and camera.

4 Operating modes control

This section describes the various operating modes available on STM32MP13x product line devices and the ways of activating them.

4.1 Low-power mode control

The low-power modes control register bits are presented below.

- PDDS bit in register PWR MPU control register (PWR_MPUCR)
  - 0: keeps Stop mode when the MPU enters CStop or LPLV-Stop2 mode when the MPU enters CStandby
  - 1: allows Standby mode when the MPU enters CStandby
- STOP2 bit in register PWR control register 1 (PWR_CR1)
  - 0: keeps system Stop mode when PDDS=0 and the MPU enters CStop
  - 1: allows system LPLV-Stop2 mode when PDDS = 0
- STPREQ_P0 bit in RCC stop request set register (RCC_MP_SREQSETR)
  - 0: writing 0 has no effect. Reading 0 means that the MPU processor does not allow the MPU domain to go to CStop/CStandby. Then if the MPU is in WFI, it is kept in CSleep
  - 1: writing one sets the STPREQ_P0 bit. Reading 1 means that the MPU processor allows the MPU domain to go to CStop

The table below summarizes the relationship between the system and the MPU low-power modes.

Table 6. System low-power modes summary

System	MPU	DDR[1]	System oscillators[2]	hclk4	PWR_LP	PWR_ON		PWR_CPU_ON
						LPCFG = 0	LPCFG = 1	
Run	CRun or CSleep	Active/Auto refresh	On	On	1	1	1	1
[3]Stop (LPDS = 0)	CStop (STPREQ_P0 = 1, STOP2 = 0, PDDS = 0)	Self-refresh	On/Off[4]	Off				
[3] LP-Stop, LPLV-Stop (LPDS = 1)					0[5]		0[5]	
LPLV-Stop2 (LPDS = 1)	CStandby (STPREQ_P0 = 1, STOP2 = 1, PDDS = 0)	Off/Self-refresh	Off					0
Standby (LPDS = x)	CStandby (STPREQ_P0 = 1, STOP2 = x, PDDS = 1)				0[6]	0[6]	0[6]	0

1. *The DDR operation state is only presented for information. The DDR may be off when MPU is in CRun, for example. When executing from boot ROM or it may be in self-refresh when the system is in any of its operation modes.*
2. *The system oscillators: HSI/HSE/CSI oscillators configured ON in RCC.*
3. *PDDS bit selects the Stop mode (PPDS bit in registers PWR MPU control register (PWR_MPUCR).*
4. *When the system oscillator HSI, HSE, or CSI is used, the state is controlled by the respective xxxKERON (HSIKERON, HSEKERON, or CSIKERON bits). These bits present in the RCC oscillator clock enable a set register (RCC_OCENSETR) and the RCC oscillator clock enables a clear register (RCC_OCENCLRR)), else the system oscillator is off.*
5. *Settings in PWR control register 3 (PWR_CR3) bits POPL have no impact on the LP-Stop and LPLV-Stop mode PWR_ON and PWR_LP pulse low time.*
6. *A guaranteed minimum PWR_ON, PWR_CPU_ON, and PWR_LP pulse time can be defined in PWR control register 3 (PWR_CR3) bits POPL.*

4.2 External control signals PWR_ON, PWR_CPU_ON, PWR_LP pins

The three output pins PWR_ON, PWR_CPU_ON, and PWR_LP are related to the V_{DDCORE} and V_{DDCPU} supplies. They are used by external components or regulators to identify which voltage level must be applied on V_{DDCORE} and V_{DDCPU}.

- PWR_ON: V_{DDCORE} supply request
  - It is automatically generated by the hardware depending on the state of the STM32MP13x device and on the value of LPCFG (PWR_ON pin configuration) bit in PWR control register 1 (PWR_CR1).
- PWR_CPU_ON: V_{DDCPU} supply request
  - It is automatically generated by the hardware depending on the state of the STM32MP13x device.
- PWR_LP: V_{DDCORE} low-power mode control (active low)
  - It is automatically generated by the hardware depending on the state of the STM32MP13x device and the value of LPDS (low-voltage deep sleep LPLV-Stop mode selection) bit on PWR control register 1 (PWR_CR1).

The table below indicates the PWR_ON, PWR_CPU_ON, and PWR_LP output pin values. These values depend on the various configurations of power modes and LPDS, LPCFG, and LVDS bits.

- LPDS (low-power deep sleep Stop mode selection) is PWR control register 1 (PWR_CR1) bit 0:
  - 0: Stop mode selected, external regulator kept in main power mode (pwr_lp = 1).
  - 1: Low-power Stop mode selected, the external regulator may enter low-power mode (pwr_lp = 0). Further low-power mode selection is provided by LVDS.
- LVDS (low-voltage deep sleep LPLV-Stop mode selection) is PWR control register 1 (PWR_CR1) bit 2:
  - 0: LP-Stop mode V_{DDCORE} and V_{DDCPU} domains supply reset level at the same level as Run mode. The V_{DDCORE} and V_{DDCPU} domains supply level in LP-Stop mode must be kept at the same level as Run mode.
  - 1: LPLV-Stop mode V_{DDCORE} and V_{DDCPU} domains supply reset level at a lower level than Run mode. It permits V_{DDCORE} and V_{DDCPU} domains lower supply levels in LPLV-Stop mode.
- LPCFG (PWR_ON pin configuration) is PWR control register 1 (PWR_CR1) bit 1:
  - 0: PWR_ON pin signals Standby mode (PWR_ON =1 in Run, Stop, LP-Stop, LPLV-Stop, LPLV-Stop2 and 0 in Standby)
  - 1: PWR_ON pin signals Standby, LP-Stop, LPLV-Stop, and LPLV-Stop2 modes (PWR_ON =1 in Run, Stop, and 0 in LP-Stop, LPLV-Stop, LPLV-Stop2, and Standby)

The table below also shows the differences in the usage of pins PWR_ON and PWR_LP depending when STM32MP13x device is used with STPMIC1x or with other external power-supply components. The power modes are detailed in the PWR section of the corresponding reference manual.

Table 7. PRW_ON, PWR_LP levels according to power modes: LPDS, LVDS, and LPCFG bits

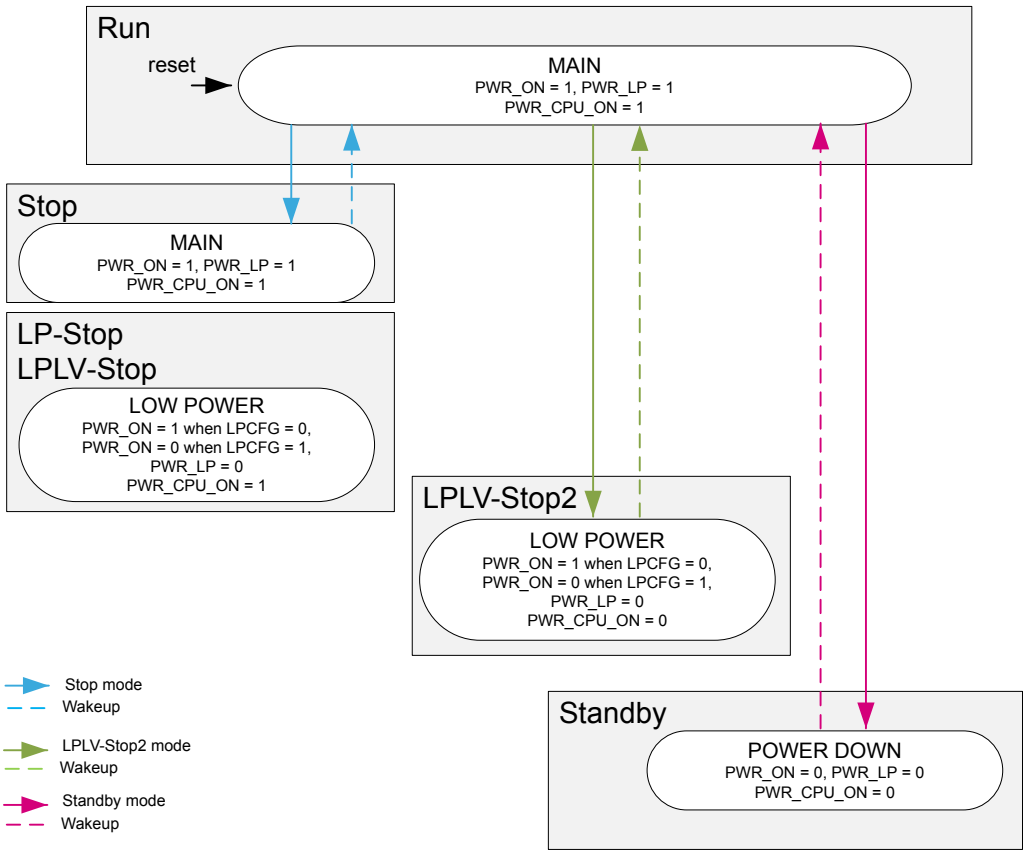
STM32MP13x state	LPDS/LVDS bits	LPCFG bit	PWR_ON/ PWR_CPU_ON	PWR_LP	V_{DDCORE}	V_{DDCPU}
Startup (until V_{DD} reaches POR threshold level)	X/X	X	0/0	0	Off	Off
Run mode	X/X	X	1/1	1	On	On
Stop mode	0/X	X	1/1	1	On	On
LP-Stop mode	1/0	0	1/1	0	On	On
		1(1)	0/1(2)	0(2)		
LPLV-Stop mode	1/1	0(3)	1/1	0	On(4)	On(5)
		1(1)	0/1(2)	0(2)		
LPLV-Stop2 mode	1/1	0(3)	1/0	0	On(4)	Off
		1(1)	0/0(2)	0(2)		
Standby mode	0/0	X	0/0(2)	0(2)	Off	Off
VBAT mode (V_{DD} powered down)	X/X	X	high-Z/high-Z	high-Z	Off	Off

1. *Configuration used with STPMIC1x PWRCTRL pin connected to PWR_ON.*
2. *There is no difference between LP-Stop, LPLV-Stop, LPLV-Stop2, and Standby mode on the PWR_ON, PWR_LP output values '00' with LPCFG bit=1.*
3. *Depending on the external power regulator that is used, this configuration may not be available (not possible to reduce the power supply level).*
4. *V_{DDCORE} supply level and power load can be decreased.*
5. *V_{DDCPU} supply level and power load can be decreased.*

Note: The highlighted cells in Table 7 show that there is no difference between LP-Stop, LPLV-Stop, LPLV-Stop2, and Standby mode on the PWR_ON, PWR_LP output values 00 with LPCFG bit = 1.

The figure below illustrates the relationship between the system states and the PWR_ON, PWR_CPU_ON, PWR_LP output pins for regulator control for STM32MP13x product line devices.

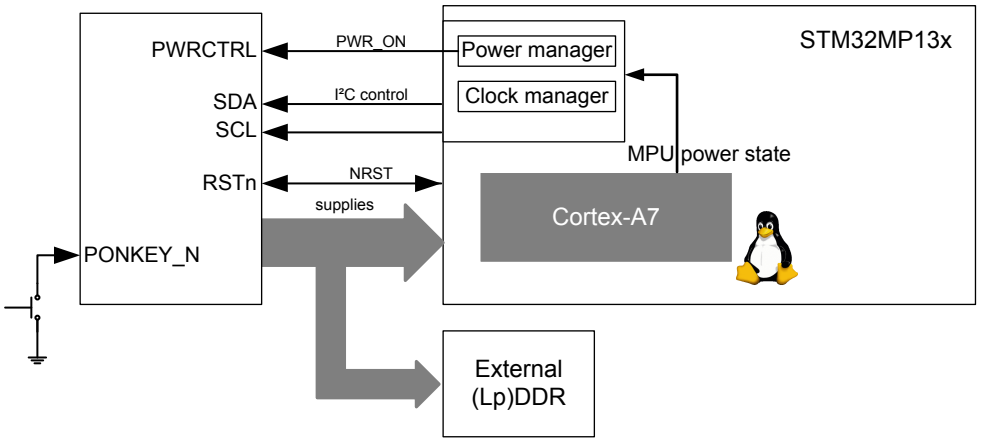
Figure 4. Power states and external regulator control



4.2.1 Using the STPMIC1x power regulator

The figure below shows the main differences at the system level when using the STPMIC1x power regulator.

Figure 5. STM32MP13x product lines high-level system architecture using STPMIC1x



When using the STPMIC1x power regulator, only the PWR_ON output-control pin from the STM32MP13x device is needed for the control of the STPMIC1x power modes. This is through the PWRCTRL pin. In that case, the user must set the LPCFG bit of the PWR_CR1 register to 1.

The highlighted cells in Table 7 show that there is no difference between the LP-Stop, LPLV-Stop, LPLV-Stop2, and Standby modes on the PWR_ON, PWR_LP output values 00 with LPCFG bit = 1.

The STPMIC1x differentiates between the LP-Stop, LPLV-Stop, LPLV-Stop2, and Standby low-power modes and applies the correct V_{DDCORE} and V_{DDCPU} value thanks to the STPMIC1x internal registers. Programming of these registers is done via the I^2C interface and not through the PWR_ON and PWR_LP pins values.

When using the STM32MPU OpenSTLinux distribution, the secure monitor TF-A or OP-TEE handles this programming. Before entering the LP-Stop, LPLV-Stop, LPLV-Stop2, or Standby mode the application must configure the STPMIC1x internal registers to the desired system power-supplies level. These are V_{DDCORE}, V_{DDCPU}, V_{DD_DDR}, and others. When entering LP-Stop mode, it is still possible to program the STPMIC1x even though the V_{DDCORE} level is not decreased. This applies only in LPLV-Stop. This results in other power supplies being shut down if not needed. This is, for example, the case for the power supply of the DDR termination resistances.

The STPMIC1x includes two 8-bit registers for each of the BUCK (1,2,3,4) and LDO (1,2,3,4,5,6):

- A BUCKx/LDOx control register in HP mode that stores the expected output voltage value during the Run mode.
- A BUCKx/LDOx control register in LP mode that stores the expected output voltage during low-power mode.

Each of these registers has the following structure:

7	6	5	4	3	2	1	0
output value <5:0>						hplp	ena

- Bits 7:2: **output value**: the output value on 6 bits corresponds to a specified voltage value (refer to the STPMIC1x datasheet).
- Bit 1: **hplp**: forces high/low load capability (allows more/less load on the STPMIC1x output).
  - 0: High-load capability
  - 1: Low-load capability

Note: hplp bit is applicable only for BUCKx registers, not for LDOx.

- Bit 0: **ena**: enable bit.
  - 0: BUCK/LDO disabled
  - 1: BUCK/LDO enabled

The following tables show how the STPMIC1x can be programmed in Run mode and before entering LP-Stop, LPLV-Stop, LPLV-Stop2, and Standby modes respectively. The below tables list all power supplies and not only relates to the V_{DDCORE}. The settings used on them are in line with the example presented in Figure 3.

Table 8. STPMIC1x (HP mode) programming for Run mode

Supply name	Control register (HP mode) /@	Run
V_{DDCORE}	BUCK4/0x23	0x69 (1.25 V)
V_{DDCPU}	BUCK1/0x20	0x69 (1.25 V) or 0x79 (1.35V) overdrive
V_{DD_DDR}	BUCK2/0x21	0x79 (1.35 V)
V_{DD}	BUCK3/0x22	0xD9 (3.3 V)
V_{REF_DDR}	VREFDDR/0x24	0x1
V_{DDA}	LDO1/0x25	0x51 (2.9 V)
V_{DD_USB}	LDO4/0x28	0x1 (3.3 V)
V_{DD_SD}	LDO5/0x29	0x51 (2.9 V)

Table 9. STPMIC1x (LP mode) programming: LP-Stop LPLV-Stop and Standby mode

Supply name	Control register (LP mode) /@	LP-Stop	LPLV-Stop	LPLV-Stop2	Standby with DDR SR	Standby w/o DDR SR
V_{DDCORE}	BUCK4/0x33	0x69 (1.25 V)	0x33 (0.9 V)	0x33 (0.9 V)	0x30 (off)	
V_{DDCPU}	BUCK1/0x30	0x69 (1.25 V)	0x33 (0.9 V)	0x30 (off)		
V_{DD_DDR}	BUCK2/0x31	0x79 (1.35 V)				0x7A (off)
V_{DD}	BUCK3/0x32	0xD9 (3.3 V)				
V_{REF_DDR}	VREFDDR/0x34	0x1				0x0 (off)
V_{DDA}	LDO1/0x35	0x51 (2.9 V)			0x50 (off)	
V_{DD_USB}	LDO4/0x38	0x1 (3.3 V)			0x0 (off)	
V_{DD_SD}	LDO5/0x39	0x51 (2.9 V)			0x50 (off)	

Note: Setting bit0 to 0 disables the corresponding BUCK/LDO, and its value is set to off.

Example: V_{DDA}= 2.9 V (0x51) when bit0 = 1, and is off (0x50) when bit0 = 0.

The application note *STM32MP13x MPU line and STPMIC1D/STPMIC1A integration on a wall adapter supply* (AN5587) provides more details on how to use STPMIC1x.

The NRST pin of STM32MP13x product lines can be activated by the STPMIC1x RST_N pin when the user is setting the STPMIC1x PONKEY_N pin to 0. Hence, this resets both STPMIC1x and STM32MP13x devices.

4.2.2 Using other external power supplies

When using external power supplies other than STPMIC1x, the use of PWR_ON, PWR_CPU_ON, and PWR_LP output control pins is possible. This allows the control of the external power supply if low-power modes below Stop mode: LP-Stop, LPLV-Stop, LPLV-Stop2, and Standby upon the application requirement.

When the STM32MP13x device goes to LP-Stop, LPLV-Stop, or LPLV-Stop2 mode, PWR_ON must be set to 1. When it is on Standby, PWR_ON must be set to 0. To activate this usage, the user must set the LPCFG bit to 0.

When the STM32MP13x device goes to LP-Stop or LPLV-Stop mode, PWR_CPU_ON must be set to 1. When it is on LPLV-Stop2 or Standby, PWR_CPU_ON must be set to 0.

4.3 Low-power mode entry sequence

The STM32MP13x devices have dedicated power modes at the subsystem level (MPU) and system level. This section details how to enter those power modes at the subsystem and system levels.

MPU subsystem

The MPU enters the low-power modes of the MPU subsystem: CSleep, CStop, and CStandby, when executing the WFI or WFE instructions. For the MPU to enter the low-power modes of the subsystem, the RCC, and PWR registers must be already correctly programmed. Refer to Section 5.1). When using the STM32MPU OpenSTLinux distribution the first stage bootloader handle these mechanisms, refer to Power management under Linux®.

System

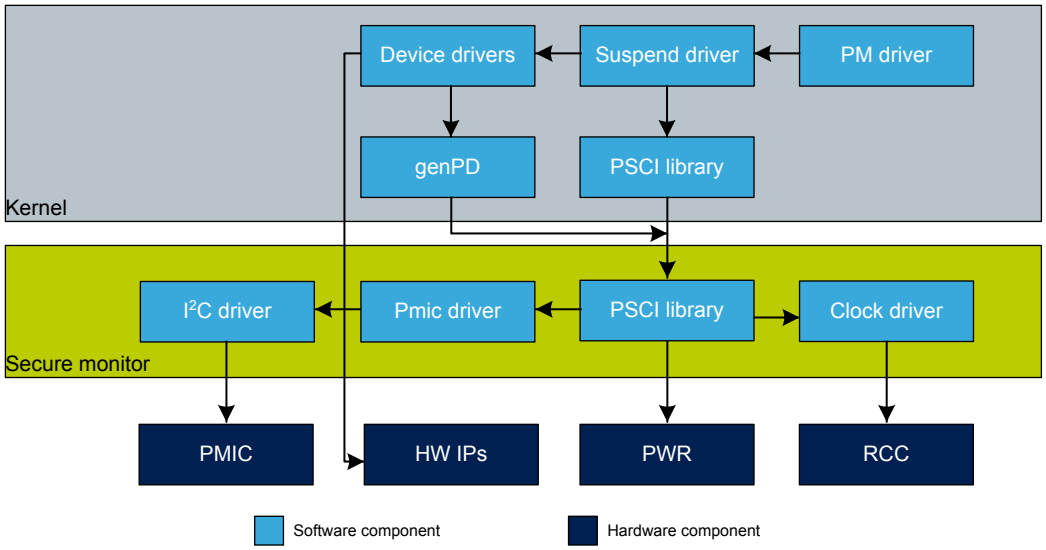
The system may enter the Stop, LP-Stop, LPLV-Stop, LPLV-Stop2, or Standby mode when all the EXTI wake-up sources are cleared. The MPU is set in CStop or CStandby mode.

Note: To avoid overconsumption on VDDQ_DDR, when entering LPLV-Stop or LPLV-Stop2 if VDDQ_DDR is not shut down, the DDR memory must be put in self-refresh. Furthermore, DDR PHY must be set in retention mode. This is done by setting bit DDRRETEN: DDR retention enable bit of the PWR control register 3 (PWR_CR3) register.

4.4 Power management under Linux®

Linux power management is done through the software framework shown in the figure below.

Figure 6. Linux power management software framework



The Linux suspend framework is used to enter the low-power modes. It relies on:

- genPD (generic power-domain) framework. This allows the definition of power domains and the mapping of IPs on these power domains
- PSCI to perform the low-level power management process

Additional information can be found on the *Power overview* wiki article available at http://wiki.st.com/stm32mpu/wiki/Power_overview. The secure monitor regulator framework is used to configure the external power-regulator voltages for each power mode. It relies on an ST STPMIC1x (PMIC) driver and an I²C driver.

Linux power commands mapping to STM32MP13x device power modes

When using the Linux operating system, the power-mode commands are predefined. This section explains how they are mapped to the low-power modes of the STM32MP13x device hardware.

Under Linux, the user can ask the system to enter into low-power mode with the following direct input commands:

To enter in Standby mode with DDR off:

```
'shutdown -h 0'
```

Or to reach any low-power mode among Stop variants and Standby, with DDR in self-refresh:

```
echo 'mem' > /sys/power/state
```

The `'disk'`, `'freeze'`, and `'standby'` commands are not supported on STM32MP13x devices.

The Cortex®-A7 is put in WFI when entering a low-power mode.

The list of available wake-up sources is not the same for all low-power modes. Thus, a software mechanism is implemented through the genPD framework to ensure that the low-power mode and the activated wake-up source are consistent.

Important: The strategy is to allow the MPU to enter the deepest low-power mode available according to the currently activated wake-up source.

Example 1: The user activates the ETH1 as a wake-up source using the `ethtool -s eth1 wol g` command.

Then, calling `echo mem > /sys/power/state` results in the MPU entering CStop allowing Stop or LP-Stop.

Example 2: The user selects the RTC as a wake-up source. The RTC is always a possible wake-up source, whatever the low-power mode status. Then, calling `echo mem > /sys/power/state` results in the MPU entering CStop allowing system Standby.

This mechanism ensures that a blocking situation such as activating ETH as a wake-up source and requesting system Standby can never happen.

Table 10 lists the deepest possible power mode according to wake-up source groups. It also lists the equivalence between Linux standard low-power modes and STM32MP13x devices system low-power modes, including the external (Lp)DDR power state: on, off, self-refresh (SR) as it is implemented in the STM32MPU OpenSTLinux distribution provided environment.

Table 10. Deepest power mode per wake-up source group and equivalence between Linux and STM32MP13x device system power modes

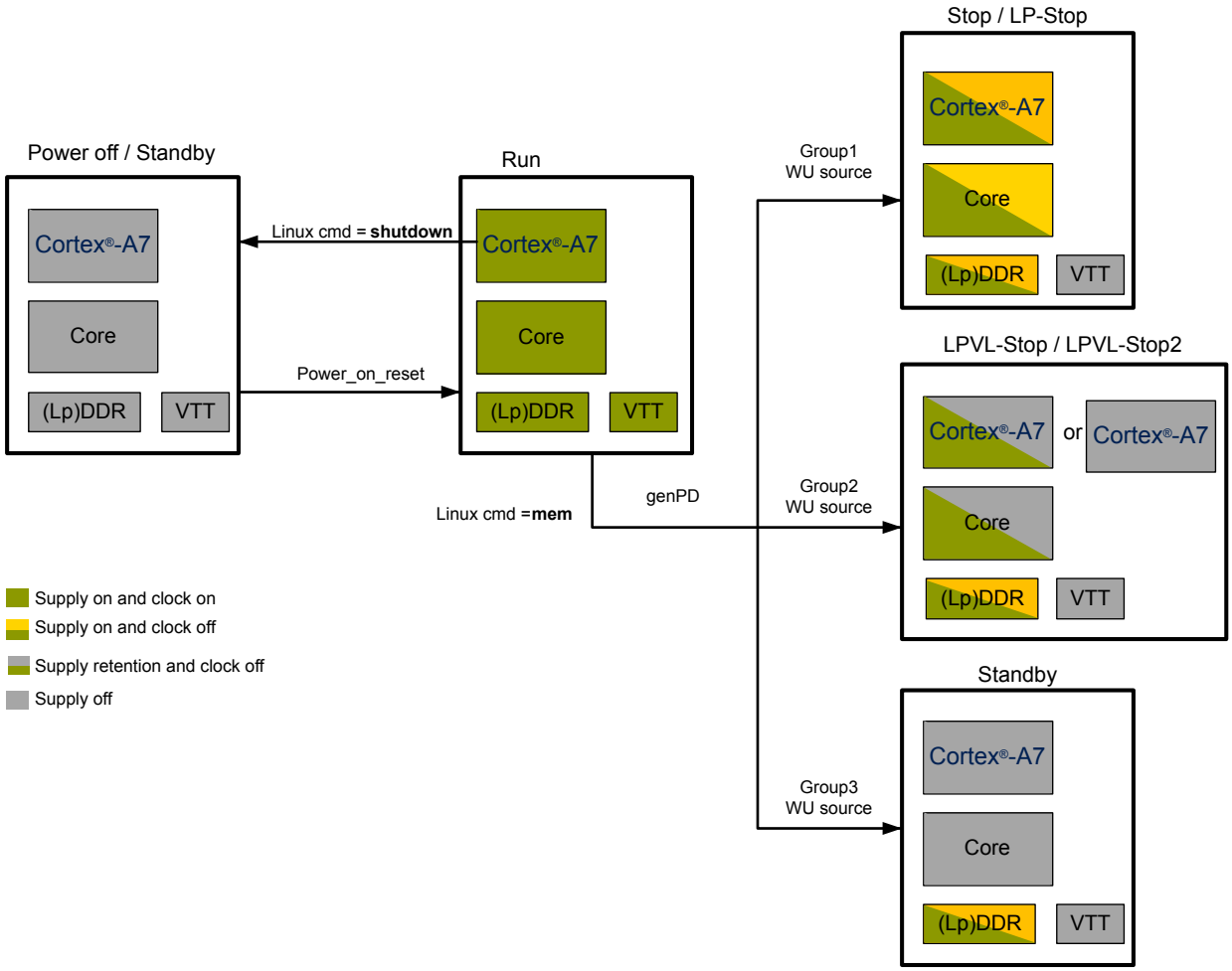
Wake-up source	Linux command	STM32MP13x device system deepest power mode	System DDR	Linux kernel state	Power consuming	Wake-up time	Comment/Application guideline
Group 1: USB, CEC, ETH	"mem"	Stop or LP-Stop	SR (VTT off)	"Suspend-to-ram"	Medium	Medium	LP-Stop: driving external PWR_LP/ PWR_ON permits designing the custom strategy for the external regulator. Typical application is to switch off DDR3 termination supply (VTT) (most likely not needed in 16-bit DDR design)
Group 2: PVD, AVD, DTS, USART, I^2C, SPI, LPTIM, GPIOs	"mem"	LPLV-Stop or LPLV-Stop2	SR (VTT off)	"Suspend-to-ram"	Low	Medium	LPLV-Stop(2): save power thanks to the power retention. Suitable for applications with aggressive power constraints and tolerant with limitations of wake-up source (refer to Table 5. Low-power mode wake-up capabilities of the system)
Group 3: BOR, Vbat mon, Temp mon, LSE CSS, RTC, TAMP, wake-up pins	"mem"	Standby	SR	"Suspend-to-ram"	Low	Medium	Standby saves more power at the expense of wake-up time
	"shutdown"	Off/VBAT	Off	Shutdown	Very low	High	-

Figure 7 shows the power state transitions available in the system. The same definitions for each wake-up source group used in the above table are considered in this figure. Group 3 wake-up sources are also functional with group 1 and group 2 sources. Group 2 is also compatible with group 1.

If several wake-up sources are activated in different groups, the low-power mode is chosen. This must respect the hierarchy stated in the table above.

The choice between Stop or LP-Stop and LPLV-Stop or LPLV-Stop2 is done through settings in the secure monitor device tree. Refer to wiki.st.com/stm32mpu/wiki/Power_overview.

Figure 7. Available power state transitions



4.5 Low-power mode exit sequence

For details on how to exit from low-power modes, refer to the corresponding product's reference manual sections: *Power supply system startup sequence*, *System Stop mode* to *System VBAT mode in PWR*.

When STM32MP13x devices exit from low-power modes, all the output voltages of the external regulators must be set up to the appropriate level. Moreover, all the output voltages of the external regulators must be stable before the Run mode is applied.

4.5.1 Exit from system power-reset

The expectation is that the external flash memory power supply is ready before the STM32MP13x device enters the Run mode, and reads data from the flash memory to boot.

- When using STPMIC1x, this expectation is automatically handled. Indeed, the STPMIC1x does not release the STM32MP13x device NRST pin until all supplies are at their expected value.
- When using a discrete regulator component, the design must make sure that these constraints are correctly handled.

4.5.2 Exit from Stop mode

When exiting from Stop mode, the voltages are not changed. Even if no issue relates to the voltages, note that while the PLL1 and PLL2 settings are restored thanks to the hardware **clock restore** automatic feature, other PLL may have to be reconfigured if needed. This also applies to all other **Stop** modes: LP-Stop, LPLV-Stop, LPLV-Stop2.

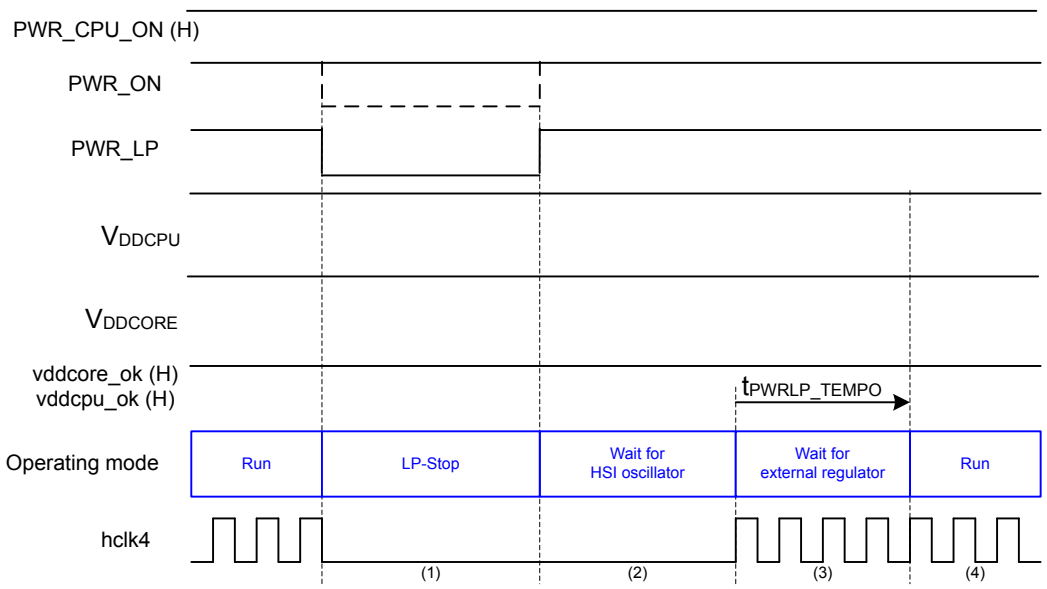
4.5.3 Exit from LP-Stop mode

On exit from LP-Stop mode, V_{DDCORE} is not changed. However, other voltages at the system level may be switched off or reduced and may need to be set back to their Run mode value.

For example, the DDR3/DDR3L (x32-bit bus width) termination-resistance power supply VTT can be switched off during LP-Stop. The STM32MP13x device waits for a programmable delay (t_{PWRLP_TEMPO}) in the **RCC PWR_LP delay control register** (RCC_PWRLPDLYCR) to permit the recovery of the Run mode value of the external regulators.

- When using STPMIC1x, the exit from LP-Stop mode is notified from the STM32MP13x device by the PWR_ON pin. The pin toggles back to 1 thus requesting changing the supply level on STPMIC1x output supplies. The STPMIC1x ensures that the t_{PWRLP_TEMPO} can be set to a minimum value of minimum value around 1000 µs (typical condition).
  - The formula for the STM32MP1 maximum output voltage rise time is:
    For BUCK output: maximum {100 µs; V2-V1/3.6 10-3 µs} (typical)
    with V2: minimum Run mode voltage V1: voltage during low-power mode.
  - For LDO output: ramp-up time is defined by the current limit and the total amount of capacitance on LDO output. For 10 µF total output capacitance on LDO3, ramp-up time is 20 µs typical.
  - For a VDD_USB of 3.3 V (from BUCK output): the maximum rise time is 3.3/3.6 10-3= 917 us (typical).
  - Exact values in the worst-case condition must be computed from the STPMIC1x datasheet. However, if wake-up time is not critical it is recommended to use some reasonable margin, for example, 5 ms.
- When using discrete external regulator components, the hardware must set the required power supplies to their Run mode value within the t_{PWRLP_TEMPO} delay.

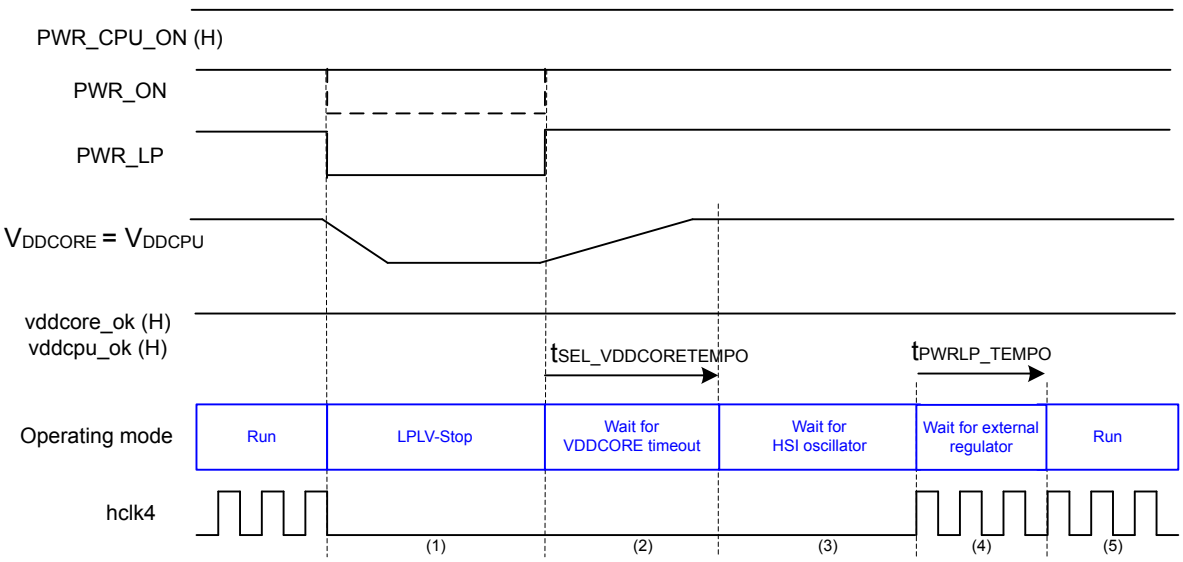
Figure 8. Wake-up sequence from LP-Stop



4.5.4 Exit from LPLV-Stop mode

- When using STPMIC1x, the exit from LP-Stop mode is notified from the STM32MP13x device by the PWR_ON pin. The pin toggles back to 1, thus requesting changing supply level on STPMIC1x output supplies. The STPMIC1x ensures that the V_{DDCORE} output supply reaches its minimum Run mode value in less than $t_{SEL\_VDDCORETEMPO}$ delay.
  - If during LPLV-Stop VDDCORE power supply (from BUCK output) is reduced to 0.9 V, then V1 = 0.9 V, V2 = 1.2 V so the maximum rise time is 0.3/3.6 10-3 = 83 µs (typical). The worst condition value must be computed from the STPMIC1x data sheet and must be less than $t_{SEL\_VDDCORETEMPO}$ (234 us).
  - Same formula to use for other power supplies that should have rise time (in worst condition) less than t_{PWRLP_TEMPO}.
  - If wake-up time is not critical it is recommended to use a reasonable margin, for example, 5 ms.
- When using discrete external regulator components, the hardware must be able to set the V_{DDCORE} supply to its minimum Run mode value within the $t_{SEL\_VDDCORETEMPO}$ delay. It must set the other required power supplies to their Run mode value within the t_{PWRLP_TEMPO} delay.

Figure 9. Wake-up sequence from LPLV-Stop



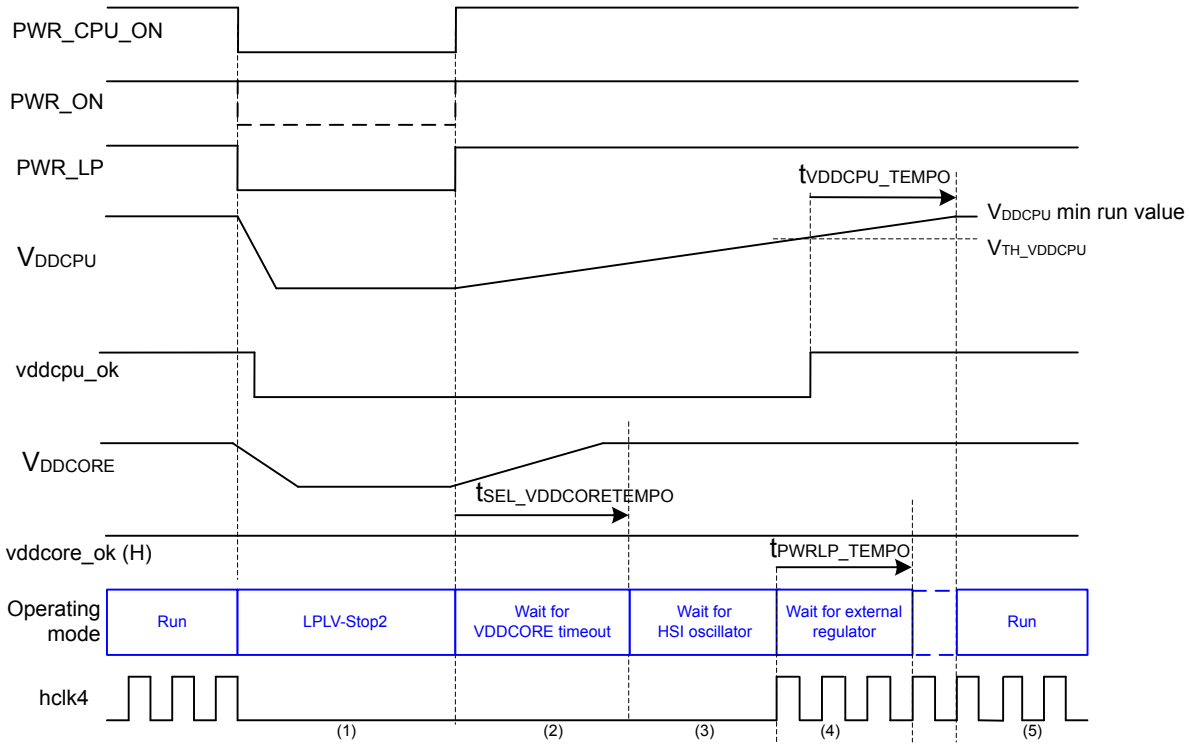
4.5.5 Exit from LPLV-Stop2 mode

The same conditions for the exit from LPLV-Stop apply to exit from LPLV-Stop2 mode. Thus, with the addition of the V_{DDCPU} that was switched off and needs to be set back to its minimum Run mode value.

- When using STPMIC1x, the V_{DDCPU} is raised in parallel to V_{DDCORE} when the PWR_ON pin toggles back to 1.
- When using discrete external regulator components, the PWR_CPU_ON pin toggles back to 1. The Run mode is started when both t_{PWRLP_TEMPO} and t_{VDDCPU_TEMPO} are elapsed. To cope with very short LPLV-Stop2 mode duration, the t_{PWRLP_TEMPO} may need to be increased to around 2 ms to allow the external regulator correct setup.

On exit from LPLV-Stop2 mode, as V_{DDCORE} is kept on, the SYSRAM content is not lost. Then TF-A is still present, program execution restarts with a local MPU reset.

Figure 10. **Wake-up sequence from LPLV-Stop2**



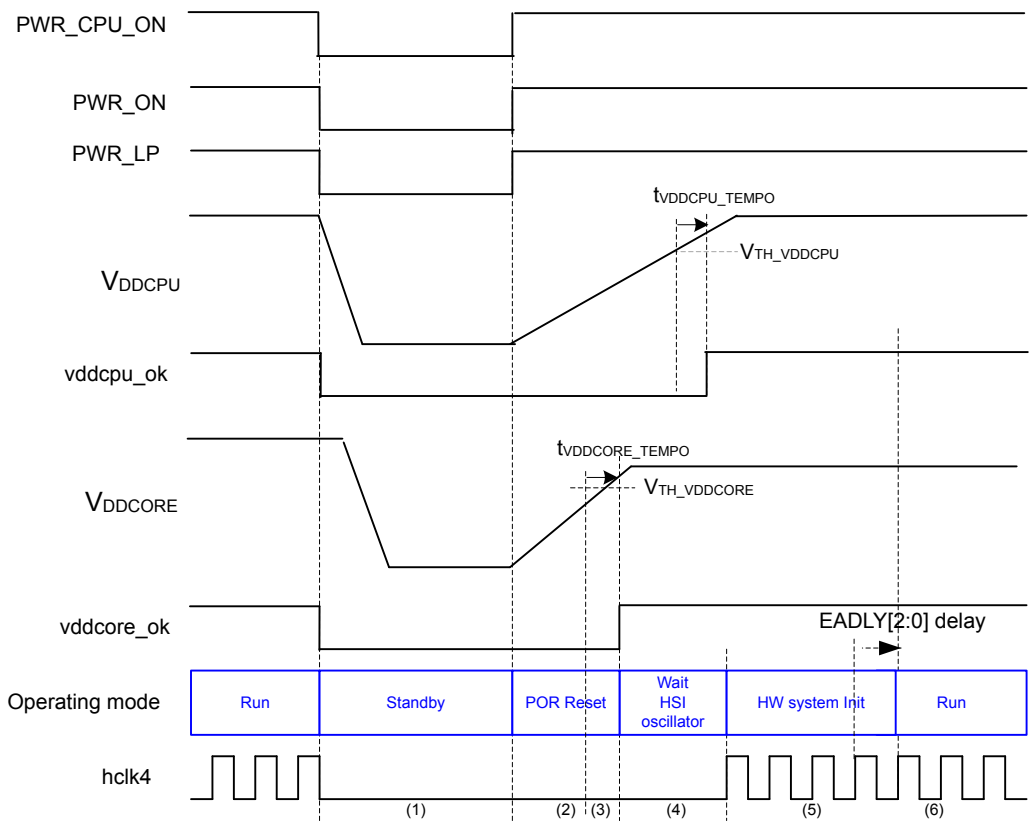
4.5.6 Exit from Standby mode

On exit from Standby mode, V_{DDCORE}, and V_{DDCPU} voltages that are turned off during Standby mode need to be switched on and set to their Run mode values.

During Standby, the V_{DD} supply is kept active. Thus, at wake-up from Standby, the pins PWR_ON and PWR_CPU_ON are set back to 1. Part of the digital logic in the PWR controller is supplied by VDD and so can drive PWR_ON and PWR_CPU_ON pins.

- When using STPMIC1x, the exit from Standby mode is notified from the STM32MP13x device by the PWR_ON pin. The pin toggles back to 1 thus requesting changing supply level on STPMIC1x output supplies. The STPMIC1x device ensures that V_{DDCORE} and V_{DDCPU} output supplies reach their minimum Run mode values in less than $t_{VDDCORETEMPO}$ delay. It ensures that the EADLY[2:0] delay sets to a minimum value defined by the same formula provided in Exit from LP-Stop mode. The recommendation is to use a reasonable margin of 5-10 ms for EADLY[2:0]. The programmable EADLY[2:0] delay is used to make sure that all bootable external interfaces are correctly supplied before the boot ROM starts accessing them.
- When using discrete external regulator components, the hardware must set the V_{DDCORE} supply to its minimum Run mode value within the $t_{VDDCORETEMPO}$ delay. It must also set the V_{DDCPU} supply to its minimum Run mode value before Run mode. The Run mode is within the $t_{VDDCORETEMPO}$ + HSI oscillator startup + EADLY[2:0] delay. Moreover, it must handle the setting of the other required power supplies to their Run mode value, within the programmed EADLY[2:0] delay.

Figure 11. **Wake-up sequence from Standby**



4.5.7 Exit from VBAT mode

When the STM32MP13x device is in VBAT mode, only the V_{SW} supply is maintained through an external VBAT supply.

The available wake-up sources (TAMP, RTC) can activate the PC13 pin to inform the external regulator about the wake-up request. The external regulator then restarts the power sequencing like for a system power reset.

4.5.8 Exit from MPU CStop and CStandby

MPU CStop and CStandby have different exit modes (see Table 6).

- When exiting from MPU CStandby with the system in LPLV-Stop2, program execution restarts with a local MPU reset. The boot ROM does not load anything from LPLV-Stop2; it calls back an address stored in a backup register (BSEC_SCRATCH).
- When exiting from MPU CStandby with the system in Standby, the program execution restarts similarly to a power-on reset: option bytes loading, reset vector fetched.
- When exiting from MPU CStop with the system in Stop, LP-Stop, or LPLV-Stop mode, the program execution restarts through:
  - The interrupt handler, if the WFI instruction or return from ISR was used to enter into low-power mode.
  - The instruction following WFE, if the WFE instruction is used to enter into low-power mode).

5 STM32MP13x lines peripherals configuration for low-power modes

STM32MP13x line devices implement many peripherals. Their configuration plays an important role in power consumption. This section describes how to configure the relevant peripherals for efficient power consumption.

5.1 Peripheral assignment and allocation

There are two available contexts during runtime:

- Cortex®-A7 secure
- Cortex®-A7 nonsecure

Some peripherals can be *assigned* either to the secured or the nonsecured context, others are always secure or always nonsecure. There are also some secure-aware peripherals, that can dynamically manage requests from both the secured and the nonsecured contexts. The peripherals assignment to a secure or nonsecure context is done by the extended TrustZone® protection controller (ETZPC).

Each peripheral is assigned either at reset time, Cortex®-A7 secure, or nonsecure, according to the reference manual register content. Otherwise, it is during boot time when using STM32MPU OpenSTLinux distribution, see Figure 12). The software can later update this default configuration during runtime.

When one of the contexts wants to use a peripheral assigned to it, it has to ensure that the peripheral is clocked. Therefore, before using a peripheral, the MPU has to enable it. Moreover, it can define if this peripheral remains active in CSleep mode. It is the RCC that performs this clocking enabling called *allocation* of peripherals to a processor.

Refer to the sections *Peripheral allocation* and *General clock concept overview* of the corresponding reference manual for more details.

- Enabling a peripheral is done by setting the dedicated PERxEN bit on the RCC_MP_xxxxENSETR register.
- Disabling a peripheral is done by setting the dedicated PERxEN bit on the RCC_MP_xxxxENCLRR register.
- Activating a peripheral during CSleep is done by setting the dedicated PERxLPEN bit on the RCC_MP_xxxxLPENSETR register.
- Disabling a peripheral during CSleep is done by setting the dedicated PERxLPEN bit on the RCC_MP_xxxxLPENCLRR register.

The peripheral allocation is used by the RCC to control automatically the clock gating according to the processor modes.

Depending on the operating power-supply range, some peripherals may be used with limited functionality and performance. For more details, refer to the section “*General operating conditions*” in the corresponding datasheet.

The figure below gives an example of peripheral allocation with the following considerations:

- The MPU enabled the ETH, SDMMC1, and SRAM3.
- SYSRAM, DDRC, DDRPHYC, and IWDG1 are implicitly allocated to MPU.
- The group composed by the MPU: bus matrix 1, bus matrix 2, and peripherals allocated via RCC_MP_xxxx registers are forming the MPU peripheral domain (MPU_DOM).

Note: PWR and RCC are implicitly allocated to the MPU. SRAM1, SRAM2, and SRAM3 are also implicitly allocated to the MPU because the MLAHB bridge is enabled if the processor is in CRun.

Note: The BKPSRAM has a dedicated enable bit to gate the bus interface clock. The BKPSRAM needs to be enabled before using the enable bit.

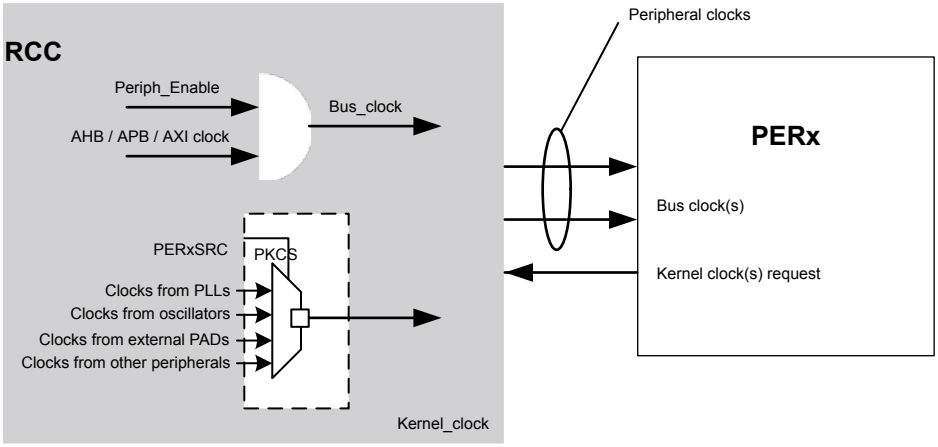
5.2 Peripheral clock distribution

The peripheral clocks are the clocks provided by the RCC to the peripherals. Two kinds of clocks are available:

- The bus interface clocks.
- The kernel clocks.

The following figure describes the peripheral clock distribution on STM32MP13x line devices.

Figure 12. **Peripheral clock distribution**



A kernel clock allows the peripheral to use a different clock frequency for the peripheral function (compared to the peripheral bus clock).

The Linux clock driver framework offers a 'clk_set_parent' service able to select the kernel clock.

For more details about kernel clock distribution and peripheral clock gating, refer to the section "*Peripheral clock gating control*" of the reference manual.

5.3 Stop, LP-Stop, LPLV-Stop, LPLV-Stop2 peripheral allocation

The peripherals using the LSI or the LSE clock are still able to operate in Stop and LP-Stop modes. While peripherals having a kernel clock request are still able to operate by selecting a clock source in RCC, which is able to run in Stop mode.

To allow the I^2Cs or USARTs to work in CStop or system Stop (LP-Stop mode), the user must select an oscillator as kernel clock: hse_ker_ck, hsi_ker_ck or csi_ker_ck according to RCC PERxEN. The selected oscillators are provided to the peripheral when the peripheral generates a kernel clock request.

The oscillators remain activated in Stop, LP-Stop mode if the xxxKERON = 1. This allows immediate delivery of the clock on wake-up from Stop, LP-Stop, or on peripheral kernel clock request.

In LPLV-Stop and LPLV-Stop2 mode, the external power supply is expected to lower its V_{DDCORE} voltage level. As a result, only a limited number of peripherals are still functional (BOR, PVD, AVD, VBATH-L monitoring, TEMPH-L monitoring, LSI, LSE, LSE CSS, RTC, IWDG). Additionally, some peripherals can wake up the system, like DTS, USART, I^2C, SPI, LPTIM, and GPIO.

6 Debug tips

This section describes how to use the debug features in low-power modes.

6.1 Enabling debugging in low-power mode: low-power mode emulation

When entering a low-power mode, the clock of the processors, the subsystem clocks, or the core power supply may be switched off. This action prevents debug accesses to the related unclocked domains. STM32MP13x lines devices support a low-power mode emulation that maintains the debug capability while in low-power mode. This is controlled by the DBGLP bit in the DBGMCU configuration register (DBGMCU_CR).

Note: The naming DBGMCU is kept for consistency with legacy STM32 MCU products.

Comprehensive information can be found in the following parts of the corresponding STM32MP13x lines reference manual:

- *Low-power emulation mode in the RCC chapter*
- *Microcontroller debug unit (DBGMCU) in Debug support chapter*

Note: Using low-power emulation modes has a significant impact on power savings (various clocks and power supply being kept on). Such modes should only be used for debugging purposes. Power consumption measurements are not relevant while using low-power mode emulation.

6.2 Using HDP for debugging the low-power modes

The hardware debug port (HDP) allows the observation of internal signals that can be used to debug the entry in and the exit from low-power modes. These signals can be the output on some pins of the STM32MP13x device and they can be easily monitored (through a scope or a logic analyzer). See the *Hardware debug port* section in the corresponding STM32MP13x lines reference manual for the comprehensive list of signals and their mixing configuration.

HDP and associated GPIOs should be disabled in the final application to reach a lower power consumption.

The most important signals for power-mode debugging are described in the tables below.

Table 11. Monitoring the CSleep modes

Signal	Description
CA7 STANDBYWFI0 (HDP7, HDP_MUX=0) CA7 STANDBYWFE0 (HDP7, HDP_MUX=1)	The CA7 STANDBYWFI0 in case of WFI, or CA7 STANDBYWFE0 in the case of WFE signals is used to monitor the CA7 CSleep mode. These signals indicate whether a core is in WFI or WFE state.

Table 12. Monitoring the entry in CStop and Stop mode

Signal	Description
RCC pwrds_mpu (HDP1, HDP_MUX=6)	The RCC pwrds_mpu signal can be used to monitor the CA7 CStop mode.
RCC pwrds_sys (HDP3, HDP_MUX=6)	The RCC pwrds_sys signal can be used to monitor the STM32MP13x device Stop mode.

Table 13. Monitoring the exit from the Stop mode

Signal	Description
pwrwake_sys (HDP0, HDP_MUX=0)	The pwrwake_sys signal can be used to monitor that the EXTI has received a wake-up interruption from a peripheral.
pwrwake_mpu (HDP2, HDP_MUX=0)	The pwrwake_mpu signal can be used to monitor that the CA7 subsystem is requested to return to CRun mode.

Refer to https://wiki.st.com/stm32mpu/wiki/HDP_Linux_driver for an example of HDP usage.

6.3 Linux debug hints

Checking clock summary using: `cat /sys/debug/kernel/clk/clk_summary`:

- It hierarchically displays the clock tree with frequency and state.

Monitoring the power status flags using the `devregs` command:

- PWR_MPUCR: contains the Stop flags for MPU
- RCC_MP_RSTSR: contains the reset reasons including Standby and Cstandby

If the system does not wake up:

- Check that at least one wake-up source is enabled. As well as, wake-up peripherals are clocked from HSI or HSE clock. Check that wake-up pin polarity and pull up/pull down are set.

If the system does not enter low-power mode:

- Check that a wake-up event is not already pending when calling the mode
- `cat/proc/interrupts` can indicate such events
- Check EXTI setup.

If the overall power consumption is too high:

- Check that clocks are released when a peripheral is not used. This can be checked thanks to the `clk_summary` command.

Revision history

Table 14. **Document revision history**

Date	Version	Changes
3-Mar-2023	1	Initial release.
05-Jun-2024	2	Updated: • Cover page: document title, and added Table 1. Applicable products. • Section 3.1.2: Operating modes description: *Run mode: overdrive OPP* subsection.

Contents

List of tables

List of figures