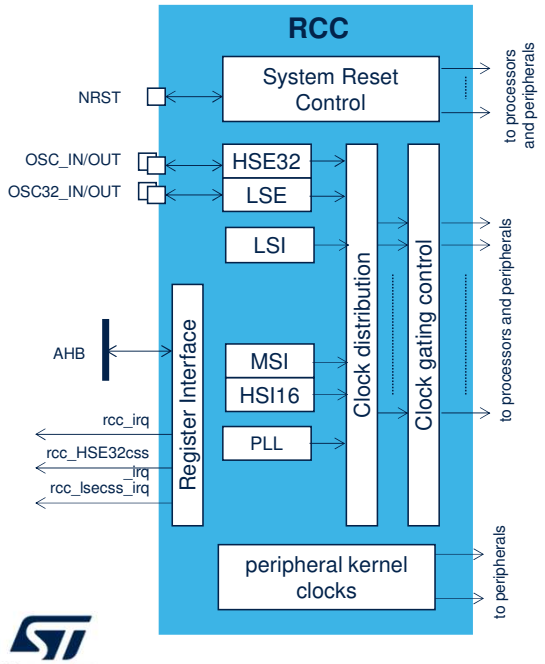
STM32WL5 - RCC

Reset and Clock Controller

Revision 1.0

Hello, and welcome to this presentation of the STM32WL5's reset and clock control.

Overview



- The Reset and Clock Controller (RCC) manages:
  - The generation of all the clocks,
    - CPU1, CPU2 and bus matrix
    - Peripheral kernel clocks
    - PLL, RC oscillators, and crystal oscillators…
  - The gating of all the clocks,
  - The control of all the system and peripheral resets.

Application benefits

- High flexibility in choice of clock sources to meet consumption and accuracy requirements.
- Independent clock control for CPU1, CPU2 sub-systems
- Safe and flexible reset management

2

The reset and clock controller manages the various reset mechanisms and the generation of the system clocks and peripheral kernel clocks.
STM32WL5 microcontrollers embed 3 internal oscillators, two oscillators for an external crystal, resonator or temperature-controlled oscillator, and one phase-locked loop (PLL).
Many peripherals have their own kernel clock, independent of the system clock.
The RCC provides high flexibility in the choice of clock sources, which allows the system designer to meet both power consumption and accuracy requirements.
The numerous independent peripheral clocks allow a designer to adjust the system power consumption without impacting the communication baud rates, and also to keep certain peripherals active in low-power mode.

Reset key features

Safe and flexible reset management without external components

- The RCC generates several types of resets:
  - Power-on reset (rst_por)
  - System reset (nreset)
  - Local peripheral resets (peripheral_rst)
  - sub-GHz radio reset (rfrst)
  - Backup domain reset (rst_vsw)

life.augmented

Safe and flexible reset management without any need for external components reduces application costs.
The RCC manages several types of resets: the power reset, the system reset, the local peripheral resets, the sub-GHz radio reset, and the backup domain reset.

System reset sources (NRST)

No external components are needed due to internal filter and power monitoring

System reset sources can reset external components

- System reset sources:
  - Power-on reset (rst_por)
  - Brown-out reset (also used when exit from Shutdown)
  - Low level on the NRST pin (external reset)
  - WWDG timeout event
  - IWDG timeout event
  - A software-generated reset (SYSRESETREQ)
  - Low-power mode (Stop, Standby, Shutdown) security reset (rst_lpwr)
  - Option byte loading reset (rst_obl)
  - Exit from Standby
  - Illegal sub-GHz radio command

life.augmented

Thanks to the voltages monitoring feature included into the power block (PWR), the filters embedded in the NRST pad, and the RCC reset controller, the amount of external components is reduced to a single external capacitor connected to the NRST pin.

The first type of reset is the system reset, which resets all the registers except certain registers for the Reset and Clock Controller and Power Controller. It also does not reset the Backup domain.

Many sources can generate a system reset:

- An invalid voltage on the VDD or VFBSMPS supply (see PWR block for details),
- An invalid voltage on VDD due to brown-out function. The brown-out function allows the user to choose its own threshold levels for the VDD supply (see PWR block for details),
- An exit from Standby or Shutdown mode

- A low-level on the NRST pad
- A timeout from the independent watchdog
- A timeout from the window watchdog
- Software reset request initiated by the Cortex M4 or Cortex M0+ core
- A low-power-mode security reset (which is generated when Stop, Standby or Shutdown mode is entered but is prohibited by the option byte configuration).
- Option byte loading
- Sending an illegal command to the sub-GHz radio.

Note as well that the system reset (except when generated by a standby reset) asserts the NRST pad, allowing the reset of external components when a system reset occurs.
The reset source flag can be found in the Reset and Clock Controller status register.

Reset coverage

- The Power-on reset resets all the logic located in VDD and VCORE domains. Back-up domain logic is not affected.
  - The System reset resets all registers except certain RCC registers, PWR registers, and the Backup domain. (retained in Standby mode)
- The Backup domain reset resets Backup domain RTC registers, Backup registers, and the RCC_BDCR register.
- Peripherals resets reset the PERxRST bits in RCC registers for the associated peripheral.
- The RCC offers flags in order to identify the system reset source.

life.augmented

5

The power-on reset is the reset having the largest coverage.
The power-on reset, resets all the logic located in the VDD and VFBSMPS domains except those in the Backup domain powered by VBAT which contains the RTC and the external low-speed oscillator.
Note that the power-on reset also triggers the system reset, so the NRST pad is asserted during power-on reset.

The system reset resets most of the logic located in VDD domain except some resources located into the RCC and the PWR blocks. The backup domain is not affected by this reset.
The backup domain reset, resets the backup domain powered by VBAT which contains the RTC and the external low-speed oscillator.
In addition, most peripherals have individual local reset control bits.

Clock key features

Choice of clock sources for low-power, accuracy, and performance

- Three internal clock sources
  - High-speed internal 16 MHz RC oscillator (HSI16)
  - Low-power internal 100 kHz to 48 MHz RC oscillator (MSI)
  - Low-speed low-power internal 32 kHz RC oscillator (LSI)
- Two external oscillators
  - High-speed external 32 MHz oscillator (HSE32) with clock security system and capacitor tuning, optimized for RF performances.
  - Low-speed external 32.768 kHz oscillator (LSE) with clock security system
- One PLL, with three independent outputs

life.augmented

6

The RCC offers a large choice of clock sources, which can be selected depending on low-power, accuracy, and performance requirements.
STM32WL5 microcontrollers embed four internal RC oscillators:

- a high-speed internal RC oscillator (HSI16) which can work at 16 MHz,
- a low-power internal RC oscillator (MSI), working at 100 kHz to 48 MHz,
- a low-speed low power internal 32 kHz RC oscillator (LSI).

STM32WL5 microcontrollers embed 2 oscillators for use with an external crystal, resonator, or temperature-controlled oscillator:

- a high-speed external 32 MHz oscillator (HSE32) with a clock security system and
- a low-speed external 32.768 kHz oscillator (LSE) also with a clock security system.

STM32WL5 microcontrollers also embed one phase-locked loops, with three independent outputs for clocking the CPUs and different peripherals at different frequencies.

High-speed internal (HSI16) clock

1% accuracy, high-speed, and fast wakeup time

Parameters	Values
Accuracy (typ.)	Over-temperature: ± 1 %
Start-up time	1.2 µs (max.)
Consumption (typ.)	150 µA (typ.)

- 16 MHz, factory- and user-trimmed
- HSI16 can be selected as
  - Wakeup clock from Stop mode
  - Backup clock for Clock Security System (CSS)
- Used as system clock when exiting Standby mode
- Can be automatically started when exiting Stop mode.
- Can remain activated during Stop mode, to avoid the start-up penalty.
- Can be used as a kernel clock by peripherals.
- Application trimming with HSE32 using bits MCO and TIM17.

life.augmented

7

The high-speed internal oscillator (HSI16) is a 16 MHz RC oscillator which provides 1% accuracy and fast wakeup times. The HSI16 is trimmed during production testing, and can also be user-trimmed.
The HSI16 can be selected as clock at wakeup from system stop, and as the backup clock if an HSE32 failure is detected by the Clock Security System.
The HSI16 is selected as system clock at wakeup from Standby mode.
The HSI16 can remain powered when the system goes to Stop mode in order to speed up the wakeup time, and use as kernel clock by peripherals in Stop mode.
Some peripherals such as the I2Cs, USART/LPUART and LPTIMs can select the HSI16 as kernel clock.
The HSI16 frequency can be trimmed versus HSE32 by using the MCO and TIM17 bits in Capture mode.

Low-power internal (MSI) clock

Low-power and fast wakeup time

Parameters	Values
Accuracy (typ.)	Over-temperature and supply voltage: ± 3 %
Start-up time	2.5 to 10 µs (typ.) (depending on selected frequency)
Consumption (typ.)	0.6 to 155 µA (typ.) (depending on selected frequency)

- 100 kHz to 48 MHz, factory- and user-trimmed
- MSI can be selected as
  - Wakeup clock from Stop mode
  - Backup clock for Clock Security System (CSS)
- Used as system clock after reset.
- The true RNG can select the MSI as kernel clock.
- Application trimming with HSE32 using bits TIM17

life.augmented

The low-power internal oscillator (MSI) is a multi-frequency RC oscillator in the 100 kHz to 48 MHz range which provides 3% accuracy and fast wakeup times. The MSI is trimmed during production testing, and can also be user-trimmed.
The MSI can be selected as clock at wakeup from system stop.
The MSI is selected as system clock after reset.
Some peripherals such as the true RNG can use the MSI as kernel clock.
The MSI frequency can be trimmed versus HSE32 by using the TIM17 bits in Capture mode.

High-speed external (HSE32) clock

Safe crystal system clock

Parameters	Values
Start-up time	Depends on selected crystal
Consumption (typ.)	50 mA (typ.)

- Features
  - External crystal resonator (32 MHz)
  - Capacitor trimming bank (no external cap)
  - Low-noise, high-performance clock for RF performance
  - Support for TCXO
- Clock Security System (CSS)
  - Automatic detection of HSE32 failure with
    - Non-maskable interrupt generation
    - Break input to TIM1/TIM16/TIM17 => critical applications such as motor control can be put in a safe state.
  - Backup clock is HSI16 or MSI => application software does not stop in case of crystal failure.
- Automatically managed by the sub-GHz radio system

life.augmented

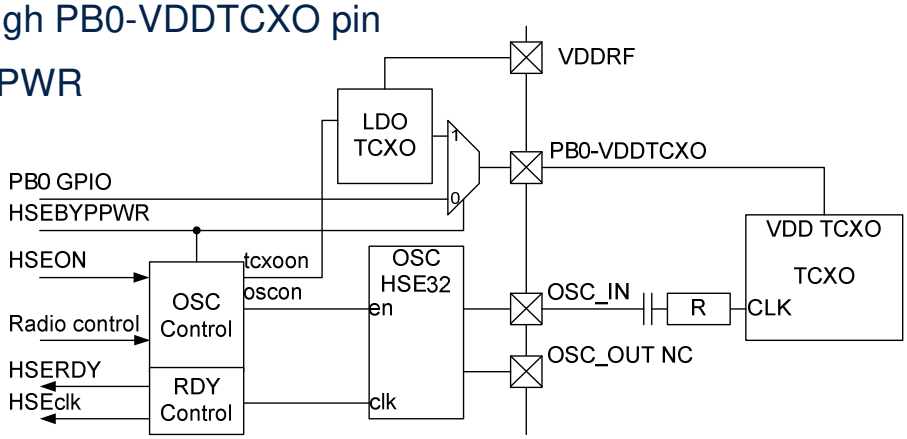
9

The high-speed external oscillator (HSE32) provides a safe crystal system clock.
The HSE32 supports a 32 MHz external crystal or temperature controlled oscillator.
The frequency can be tuned to the required few 1/10 of ppm using on-chip capacitor trimming.
A clock security system allows an automatic detection of HSE32 failure. In this case a Non-Maskable Interrupt is generated, and a break input can be sent to timers in order to put critical applications such as motor control in a safe state. When an HSE32 failure is detected, the system clock is automatically switched to HSI16 or MSI, so the application software does not stop in case of crystal failure.
The use of the high-speed external oscillator clock is mandatory when the sub-GHz radio is active. The high-speed external oscillator will automatically be managed in line with the sub-GHz radio activity.

High-speed external (HSE32) TCXO clock

Temperature compensated crystal oscillator

- External TCXO
- Supply provided through PB0-VDDTCXO pin
- Selected by HSEBYPPWR



life.augmented

The High Speed External (HSE) clock may come from an external temperature-compensated crystal oscillator. The TCXO operation mode is selected by the HSEBYPPWR bit when used by the CPU, and with the sub-Giga Hertz radio command Set_TcxoMode() when used by the sub-Giga Hertz radio. The TCXO supply is available from the PB0-VDDTCXO pin, this is also used to enable and disable the TCXO. The supply level can be selected through the sub-Giga Hertz radio command set_TcxoMode().

Low-speed internal (LSI) clock

internal 32 kHz clock

- The ultra-low-power LSI (available in all modes except Shutdown and VBAT)
  - can be used for RTC, LCD, LPTIMs, and IWDG. (Must not be used for the radio system)
  - Application trimming with HSE32 using bits TIM16.

	LSI 32 kHz
Accuracy (typ.)	Initial: ± 1.6 %
	Over-temperature: ± 1.5 %
	Over VDD: +0.1 / -0.2%
Consumption (typ.)	110 nA

life.augmented

11

STM32WL5 microcontrollers embed an ultra low-power Low Speed Internal oscillators providing 32 kHz named LSI.
Available in all modes except Shutdown and VBAT.
The LSI can be used to clock the RTC, low-power timers, and the independent watchdog.
The accuracy of the LSI is plus or minus 1.6%, plus 1.5% over temperature and plus 0.2% over voltage. The LSI consumption is typically 110 nA.
The LSI frequency can be trimmed versus HSE32 by using the TIM16 bits in Capture mode.

Low-speed external (LSE) clock

32.768 kHz configurable for low-power or high-drive
Available in all power modes and in VBAT mode

- The LSE can be used with external quartz or resonator, or with external clock source in Bypass mode.
- Clock Security System on LSE: Available in all modes except VBAT.
- The LSE can be used for sub-GHz radio system, RTC, true RNG, LPTIMs, USART, and LPUART.

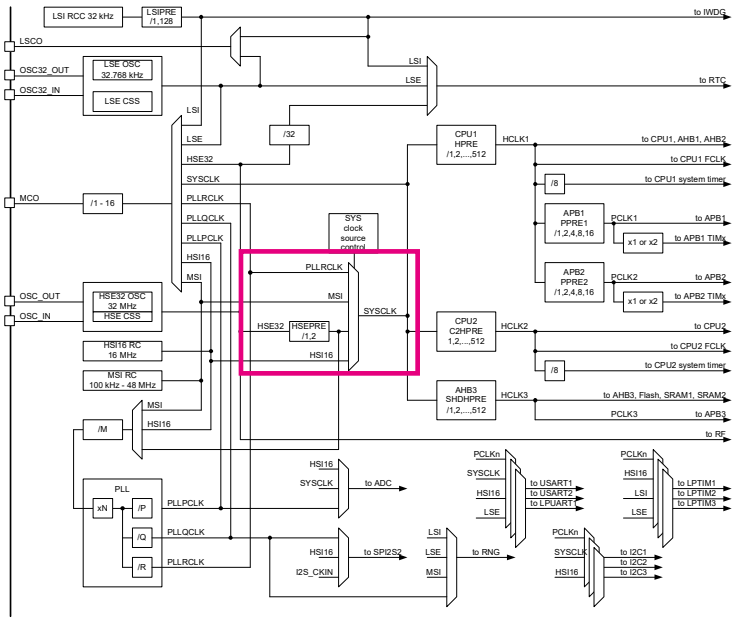
Mode	Maximum critical crystal gm (µA/V)	Consumption (nA)
Ultra-low power	0.5	250
Medium-low driving	0.75	315
Medium-high driving	1.7	500
High driving	2.7	630

life.augmented

12

The 32.768 kHz low-speed external oscillator (LSE) can be used with an external quartz or resonator, or with an external clock source in Bypass mode. The oscillator driving capability is programmable. Four modes are available, from Ultra-low-power mode with a consumption of only 250 nA, to High-driving mode.
A clock security system monitors failure of the LSE oscillator. In case of failure, the application can switch from the RTC clock to the selected LSI clock.
The clock security system is functional in all modes except VBAT. It is also functional under reset.
The LSE can be used to clock the sub-giga Hertz radio system, the RTC, true RNG the low-power timers, the USART, and low-power UART peripherals.

Clock tree



- Dynamic switch for system clock selection
- System clock source can be:
  - HSI16
  - MSI (default after reset and exit from Standby mode)
  - HSE32
  - PLL (PLLRCLK)
- Dynamic frequency dividers allow easy frequency adjustments
  - Cortex-M4 core (HCLK1)
  - Cortex-M0+ radio system (HCLK2)
  - Flash and SRAM memories (HCLK3)

life.augmented

13

The system clock can be derived from the HSI16, MSI, HSE32 or the PLLRCLK output of the PLL system.
The switch used to select the system clock is dynamic, meaning that it is possible to change the frequency on-the-fly according to application performance needs.
The Cortex-M4 core, Cortex-M0+ core and the Flash memory have their independent clock dividers allowing each of them to run on different frequencies. It is recommended to run the Flash memory on the HCLK3 at least at the same speed as the highest frequency selected for the Cortex-M4 and Cortex-M0+ cores.
In addition, all the pre-scalers presented in the figure are dynamic, so they can be changed on-the-fly as well, making the frequency scaling operation very simple.

Clock frequency and voltage scaling

Power optimization for lower frequencies

- When running at lower frequencies, additional power can be saved by changing the operating range.
  - Range 1
    - Max. Cortex-M4 frequency = 48 MHz
    - Max. Cortex-M0+ frequency = 48 MHz
    - Max. PLL VCO frequency = 344 MHz
    - Max. MSI frequency = 48 MHz
  - LPRun
    - Max. Cortex-M4 frequency = 2 MHz
    - Max. Cortex-M0+ frequency = 2 MHz
    - PLL disabled
    - Max. MSI frequency = 2 MHz
  - Range 2
    - Max. Cortex-M4 frequency = 16 MHz
    - Max. Cortex-M0+ frequency = 16 MHz
    - Max. PLL VCO frequency = 128 MHz
    - Max. MSI frequency = 24 MHz

life.augmented

14

To optimize power consumption at lower frequencies the operating range can be changed or Low-power Run mode can be selected.
In Range 1, the clocks of the Cortex-M4 (HCLK1), Cortex-M0+(HCLK2), and Shared bus (HCLK3) must not exceed 48 MHz.
In Range 2, the clocks of the Cortex-M4 (HCLK1), Cortex-M0+(HCLK2), and Shared bus (HCLK3) must not exceed 16 MHz.
In Low-power Run mode, the clocks of the Cortex-M4 (HCLK), Cortex-M0+(HCLK2), and Shared bus (HCLK3) must not exceed 2 MHz.

PLL

Wide input range, accurate output frequency

- One integer PLL:
- PLL
  - 1 dedicated output for system clock
  - 1 output option for SPI2S, true RNG
  - 1 output option for ADC
- PLL clocks:
  - A single unique source HSE32, HSI16 or MSI with pre-divider
  - Wide input frequency range
    - 2.66 to 16 MHz
  - Wide VCO output frequency range
    - 96 to 128 MHz in Range 2
    - 96 to 344 MHz in Range 1
  - 3 outputs
    - PLL (R & Q)
    - With post-divider range from 2 to 8
    - PLL (P)
    - With post-divider range from 2 to 32

life.augmented

15

The PLL embedded into the STM32WL5 microcontroller provides a flexible way to generate the required frequency for the system or peripheral kernel clocks.
It offers a wide input frequency range from 2.66 to 16 MHz.
The PLL can be used with clock source: HSE32, HSI16 or MSI which can be pre-divided.
The PLL VCO has a wide frequency range from 96 to a maximum of 344 MHz in Range 1 and a maximum of 128 MHz in Range 2.
The PLL provides 3 different output clocks which are all derived from the PLL VCO frequency via post-dividers (/P, /Q and /R).
The PLL is used to generate the system clock and SPI2S, true RNG, and ADC peripheral kernel clock.

Clock outputs

- Two clock outputs:
  - MCO available in Run and Stop modes
    - HSI16, MSI, HSE32, PLLSYSCLK, SYSCLK, LSI, or LSE
  - LSCO available in Run, Stop and Standby modes
    - LSI or LSE

life.augmented

16

The Multi-Clock Output is available on GPIO pin PA8 in Run and Stop modes and can select various high- and low-speed clocks.
A Low-Speed Clock Output is available on GPIO pin PA2 in Run, Stop and Standby modes and can select various low-speed clocks.

Peripheral kernel clock

Peripheral interface clock independent from peripheral bus clock

- Some peripherals feature an independent interface kernel clock
  - sub-GHz radio, TrueRNG, SPI2S, ADC, I2C, USART, LPUART, and LPTIM
    - The bus clock allows access to peripheral registers
    - The kernel clock is used to handle the interface function
- Both the peripheral bus clock and kernel clock are gated with the RCC peripheral clock enable bits xxxEN and xxxSMEN.

xxxEN	xxxSMEN	bus clock	kernel clock
0	x	Stopped	
1	0	Active in Run mode, stopped in Sleep and Stop modes	
1	1	Active in Run and Sleep modes, stopped in Stop mode	Active in Run, Sleep and Stop modes

- Some peripherals are able to operate on its kernel clock in Stop mode
  - sub-GHz radio, I2C, USART, LPUSART and LPTIM
  - When the kernel clock selects HSI16, LSI, or LSE.

life.augmented

17

Some peripherals have a separate clock for the processor bus interface and the specific peripheral interface function. The bus clock is used to access the peripheral registers, whereas the kernel clock is used for the specific peripheral interface function.
Having a separate bus clock and kernel clock allows the application to change the interconnect and processor working frequency without affecting the peripheral operation. For example, the USART kernel clock is used to generate the baud rate for the serial interface communication, and the bus clock for the register interface.
The enabling of both the peripheral bus clock and kernel clock is controlled by the Reset and Clock Controller's peripheral enable and sleep mode enable bits. When both bits are set to one, the peripheral is able operate and transfer data in Sleep mode. When HSI16, LSI, or LSE is selected as the kernel clock, the peripheral is able to operate and wake up the system from Stop mode. In Stop mode, the

peripheral is not able to transfer data on the bus matrix, for example to memory. Refer to the specific peripheral training slides for more information.

Clock gating (1)

- The CPU, bus matrix and peripheral clocks are gated depending on the:
  - CPU operating mode (CRun, CSleep, or CStop)
  - Peripheral allocation per CPU
  - Peripheral Sleep mode enable per CPU
- The RCC_busENR.xxxEN and RCC_C2busENR.xxxEN bits:
  - Used to allocate a peripheral to a CPU.
  - The peripheral operating modes will follow the mode of the CPU where it is allocated to.
  - Peripherals may be assigned to both CPUs.
- The RCC_bus_SMENR.xxxEN and RCC_C2busSMENR.xxxEN bits:
  - Used to keep the peripheral in operation when the CPU is in CSleep mode.

life.augmented

18

CPU, bus matrix and peripheral clocks are gated according to the CPU operating mode, the peripheral allocation and the peripheral sleep mode enable bit. A peripheral is allocated to a CPU when the Reset and Clock Controller peripheral enable bit belonging to the CPU is set. The peripheral and the associated bus matrix is clocked whenever the CPU is in CRun mode. Before accessing a peripheral, it must be enabled by the CPU. When both CPUs need to access a peripheral, they must both enable the peripheral by the Reset and Clock Controller peripheral enable bits belonging to the CPUs.

Clock gating (2)

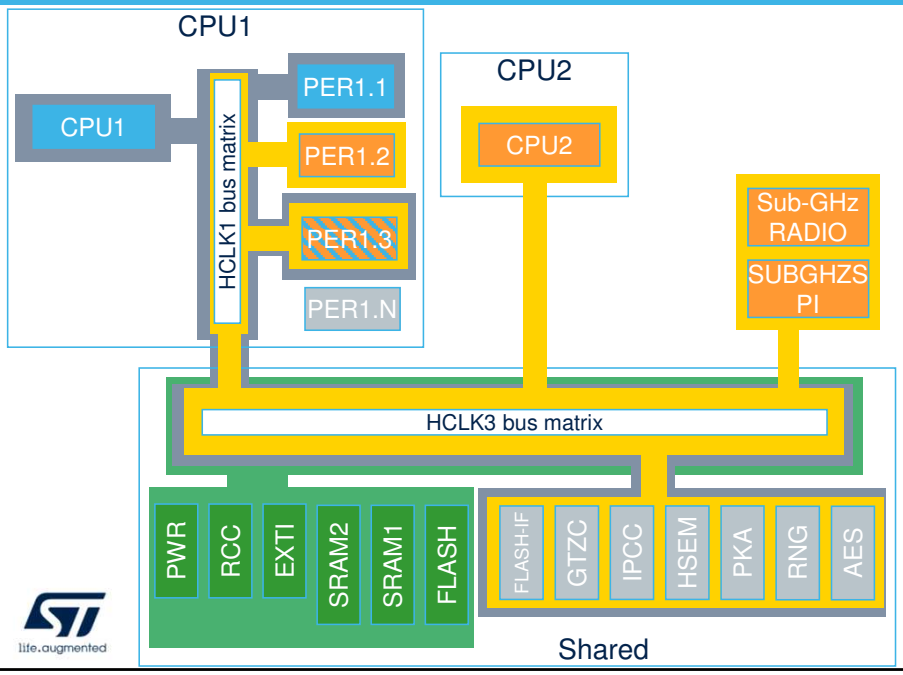
- The CPU
  - Clocked when the CPU is in CRun mode
  - Stopped when the CPU is in CSleep or CStop mode
- Peripherals
  - Clocked when allocated to a CPU in CRun mode
  - Clocked when allocated and enabled for Sleep mode operation to a CPU in CSleep mode.
  - Stopped when not allocated or the allocated CPU is in Cstop mode
  - Stopped when allocated and not enabled for Sleep mode operation and the CPU is in CSleep
- Bus matrix
  - Clocked when a CPU or Peripheral on the bus matrix is clocked
  - Stopped when a CPU and all peripherals on the bus matrix are stopped.

life.augmented

A CPU is only clocked when in CRun mode.
Only allocated peripherals are clocked when the CPU is in CRun, or when the CPU is in CSleep when the peripheral sleep mode operation is enabled.
A bus matrix will be clock when a CPU or peripheral on the bus matrix is clocked.

Peripheral allocation

Optimized low-power clocking



- System peripherals:
  - PWR, RCC, EXTI, FLASH, SRAM1 and SRAM2
  - Active with both CPUs.
- Allocated peripherals: CPU1 CPU2
  - Flash interace, IPCC, PKA, RNG, AES2, SUBGHZSPI, PERn.m
  - Need allocation for CPU1 and CPU2
- De-located peripherals
- Only needed CPU, bus matrix and peripheral clocks will be active.

20

It is important to notice that the Reset and Clock Controller offers two register sets, allowing each processor to allocate (enable) peripherals. A peripheral and the associated bus matrix will only be clocked when allocated by a CPU and the CPU is in CRun mode, or in CSleep mode when the CPU peripheral sleep mode enable bit for this peripheral is also set.

Depending on the peripheral function, peripherals have a different behavior.

Peripherals needed for the system to operate don’t have enable bits and are all time-allocated to both CPUs.

All other peripherals can be allocated by both CPUs.

Before accessing a peripheral the CPU must allocate it. If a peripheral is shared by both CPUs, it must be allocated by both processors; it is up to the application to avoid peripheral access conflicts. The Hardware Semaphore IP exists to help manage accessing shared peripherals.

The peripheral allocation allows dynamical configuration of a

CPU sub-system and to have only the peripherals used by the CPU being clocked. The CPU, plus the peripherals allocated by this CPU, and the associated bus matrixes are considered by the Reset and Clock Controller as a CPU sub-system. To give a simple example, when the CPU1 is active in Run mode, the system peripherals, the CPU1 peripherals and any allocated peripheral will run as well, including the shared bus matrix and the CPU1 bus matrix.
In the example shown, PER1.1 is allocated by CPU1. PER1.2 and the Sub-Giga Hertz radio is allocated by CPU2. PER1.3 is accolated by both CPUs.

Operating states

- Sub-system states
  - in CRun or CSleep mode, its bus matrix and peripherals are clocked.
  - in CStop mode, its bus matrix and peripherals bus clock are stopped.
- System states
  - Is in Run mode when at least one sub-system is in CRun or CSleep mode.
  - Is in Stop, Standby or Shutdown mode when all sub-systems are in CStop mode.

System States	Cortex-M4 sub-system (CPU1)	Cortex-M0+ sub-system (CPU2)
Run*	CRun/CSleep/CStop	CRun/CSleep/CStop
Stop	CStop	CStop
Standby		
Shutdown		

life.augmented

* At least one sub-system shall be in CRun or CSleep

The following table gives a simplified view of the system states versus sub-system states.

- When a sub-system is in CRun or CSleep mode, its bus matrix is clocked.
- When a sub-system is in CStop mode, its bus matrix clock is stopped.
- The system only enters Stop, Standby or Shutdown mode when all sub-systems are in CStop mode.

For more details on system states, please refer to the power controller (PWR) training slides.

Sub-GHz radio system clock

Autonomous radio operation

- The Radio system is able to operate autonomously without the need for a CPU.
  - Directly manages the HSE32 clock sources.
- Clocks needed for the Radio system are automatically enabled.
  - The radio internal and external system clock

life.augmented

The radio system can operate autonomously and is able to wakeup the CPU system from Stop and Standby modes.
The clock for the radio system is managed by the radio stack and in some operating modes by the radio system itself: the HSE32 clock is used for radio TX and RX communication.

Wakeup and startup

- At system power-on and system reset startup:
  - The MSI is selected as system clock. Other clocks and PLLs are disabled.
- When CPU and system wake up from Stop mode:
  - The HSI16 or MSI can be selected as system clock (STOPWUCK). Other clocks and PLLs are disabled.
  - The HSI16 may be kept active during Stop to allow use as peripheral kernel clock. (HSIKERON)
  - The sub-GHz radio my be active during Stop mode, using an internal radio clock or HSE32.
- When CPU and system wake up from Standby mode:
  - The MSI is selected as system clock. Other clocks and PLLs are disabled.
  - The sub-GHz radio my be active during Standby mode, using an internal radio clock or HSE32.
- When CPU wakes up from CStop mode with system in Run mode:
  - The CPU clocks will be the same as when entering CStop mode.

life.augmented

When the system restart, the clock system is reset, all high-speed clocks and PLLs are disabled, except for the high-speed clock used to start up the system.
LSI and LSE are still working, if they were previously enabled.
After power on and a system reset, the MSI clock is enabled as system clock.
When waking up from Stop mode, the system clock can be selected between MSI or HSI16 with Reset and Clock Controller register bit STOPWUCK. In Stop mode, the HSI16 clock may be kept active to allow its use as peripheral kernel clock. In Stop mode the sub-GHz radio may be active using its internal clock or the HSE32.
When waking up from Standby mode, the MSI clock is enabled as system clock. When the CPU is in Standby mode the sub-GHz radio may be active using its internal clock or the HSE32.
When a CPU wakes up from CStop mode, while the system

remained in Run mode, the clock settings are maintained, and the CPU will wake up with the same clock as when it entered CStop mode.
The system wakeup mode can be read from the Power Controller Stop and Standby flags. Refer to Power Controller training slides for more information.
The CPU CSleep mode does not affect the clock settings, but only plays on the CPU sub-system clock gating.

Interrupts

Interrupt event	Type	Description
HSE32 clock security system	**NMI**	Set when a failure is detected in the HSE32 oscillator
LSE clock security system	**IRQ**	Set when a failure is detected in the LSE oscillator
PLL ready interrupt flag	**IRQ**	Clock ready caused by PLL lock
HSE32 ready	**IRQ**	Clock ready caused by the HSE32 oscillator
HSI16 ready	**IRQ**	Clock ready caused by the HSI16 oscillator
MSI ready	**IRQ**	Clock ready caused by the MSI oscillator
LSE ready	**IRQ**	Clock ready caused by the LSE oscillator
LSI ready	**IRQ**	Clock ready caused by the LSI oscillator

life.augmented

This slide lists the Reset and Clock Controller interrupts. The HSE32 and LSE clock security systems, the PLL, and all five oscillator ready signals can generate an interrupt.

Related peripherals

- Refer to these trainings linked to this peripheral, if needed:
  - Power control (PWR)
  - Asynchronous Interrupts and Event Controller (EXTI)

life.augmented

In addition to this training, you may find the Power Control and Extended Interrupt Controller trainings useful.