

Introduction

This reference manual targets application developers. It provides complete information on how to use the STM32G4 Series microcontroller memory and peripherals.

The STM32G4 Series is a family of microcontrollers with different memory sizes, packages and peripherals.

For ordering information, mechanical and electrical device characteristics refer to the corresponding datasheets.

For information on the Arm® Cortex®-M4 core, refer to the *Cortex®-M4 Technical Reference Manual*.

The STM32G4 Series microcontrollers include ST state-of-the-art patented technology.

Related documents

- Cortex®-M4 Technical Reference Manual, available from: <http://infocenter.arm.com>
- STM32G4xx datasheets
- STM32G4xx erratasheets
- STM32F3, STM32F4, STM32G4 and STM32L4 Series Cortex®-M4 programming manual (PM0214)

Contents

| | | |
|----------|---|-----------|
| 1 | Documentation conventions | 73 |
| 1.1 | General information | 73 |
| 1.2 | List of abbreviations for registers | 73 |
| 1.3 | Glossary | 74 |
| 1.4 | Product category definition | 74 |
| 1.5 | Availability of peripherals | 74 |
| 2 | System and memory overview | 78 |
| 2.1 | System architecture | 78 |
| 2.1.1 | I-bus | 79 |
| 2.1.2 | D-bus | 79 |
| 2.1.3 | S-bus | 80 |
| 2.1.4 | DMA-bus | 80 |
| 2.1.5 | BusMatrix | 80 |
| 2.2 | Memory organization | 81 |
| 2.2.1 | Introduction | 81 |
| 2.2.2 | Memory map and register boundary addresses | 82 |
| 2.3 | Bit banding | 86 |
| 2.4 | Embedded SRAM | 87 |
| 2.4.1 | Parity check | 88 |
| 2.4.2 | CCM SRAM Write protection | 88 |
| 2.4.3 | CCM SRAM read protection | 89 |
| 2.4.4 | CCM SRAM erase | 89 |
| 2.5 | Flash memory overview | 90 |
| 2.6 | Boot configuration | 90 |
| 2.6.1 | Boot configuration | 90 |
| 3 | Embedded Flash memory (FLASH) for category 3 devices | 93 |
| 3.1 | Introduction | 93 |
| 3.2 | FLASH main features | 93 |
| 3.3 | FLASH functional description | 94 |
| 3.3.1 | Flash memory organization | 94 |

| | | |
|--------|---|-----|
| 3.3.2 | Error code correction (ECC) | 96 |
| 3.3.3 | Read access latency | 97 |
| 3.3.4 | Adaptive real-time memory accelerator (ART Accelerator) | 99 |
| 3.3.5 | Flash program and erase operations | 101 |
| 3.3.6 | Flash main memory erase sequences | 102 |
| 3.3.7 | Flash main memory programming sequences | 103 |
| 3.3.8 | Read-while-write (RWW) available only in dual bank mode (DBANK=1) | 107 |
| 3.4 | FLASH option bytes | 109 |
| 3.4.1 | Option bytes description | 109 |
| 3.4.2 | Option bytes programming | 117 |
| 3.5 | FLASH memory protection | 120 |
| 3.5.1 | Read protection (RDP) | 120 |
| 3.5.2 | Proprietary code readout protection (PCROP) | 123 |
| 3.5.3 | Write protection (WRP) | 125 |
| 3.5.4 | Securable memory area | 126 |
| 3.5.5 | Disabling core debug access | 127 |
| 3.5.6 | Forcing boot from Flash memory | 127 |
| 3.6 | FLASH interrupts | 128 |
| 3.7 | FLASH registers | 129 |
| 3.7.1 | Flash access control register (FLASH_ACR) | 129 |
| 3.7.2 | Flash Power-down key register (FLASH_PDKEYR) | 130 |
| 3.7.3 | Flash key register (FLASH_KEYR) | 131 |
| 3.7.4 | Flash option key register (FLASH_OPTKEYR) | 131 |
| 3.7.5 | Flash status register (FLASH_SR) | 132 |
| 3.7.6 | Flash control register (FLASH_CR) | 133 |
| 3.7.7 | Flash ECC register (FLASH_ECCR) | 135 |
| 3.7.8 | Flash option register (FLASH_OPTR) | 137 |
| 3.7.9 | Flash PCROP1 Start address register (FLASH_PCROP1SR) | 139 |
| 3.7.10 | Flash PCROP1 End address register (FLASH_PCROP1ER) | 140 |
| 3.7.11 | Flash Bank 1 WRP area A address register (FLASH_WRP1AR) | 141 |
| 3.7.12 | Flash Bank 1 WRP area B address register (FLASH_WRP1BR) | 141 |
| 3.7.13 | Flash PCROP2 Start address register (FLASH_PCROP2SR) | 142 |
| 3.7.14 | Flash PCROP2 End address register (FLASH_PCROP2ER) | 142 |
| 3.7.15 | Flash Bank 2 WRP area A address register (FLASH_WRP2AR) | 143 |
| 3.7.16 | Flash Bank 2 WRP area B address register (FLASH_WRP2BR) | 144 |
| 3.7.17 | Flash Securable area bank1 register (FLASH_SEC1R) | 144 |

| | | |
|----------|---|------------|
| 3.7.18 | Flash Securable area bank2 register (FLASH_SEC2R) | 145 |
| 3.7.19 | FLASH register map | 146 |
| 4 | Embedded Flash memory (FLASH) for category 4 devices | 148 |
| 4.1 | Introduction | 148 |
| 4.2 | FLASH main features | 148 |
| 4.3 | FLASH functional description | 149 |
| 4.3.1 | Flash memory organization | 149 |
| 4.3.2 | Error code correction (ECC) | 150 |
| 4.3.3 | Read access latency | 150 |
| 4.3.4 | Adaptive real-time memory accelerator (ART Accelerator) | 151 |
| 4.3.5 | Flash program and erase operations | 154 |
| 4.3.6 | Flash main memory erase sequences | 155 |
| 4.3.7 | Flash main memory programming sequences | 156 |
| 4.4 | FLASH option bytes | 160 |
| 4.4.1 | Option bytes description | 160 |
| 4.4.2 | Option bytes programming | 164 |
| 4.5 | FLASH memory protection | 166 |
| 4.5.1 | Read protection (RDP) | 166 |
| 4.5.2 | Proprietary code readout protection (PCROP) | 169 |
| 4.5.3 | Write protection (WRP) | 170 |
| 4.5.4 | Securable memory area | 171 |
| 4.5.5 | Disabling core debug access | 172 |
| 4.5.6 | Forcing boot from Flash memory | 172 |
| 4.6 | FLASH interrupts | 173 |
| 4.7 | FLASH registers | 174 |
| 4.7.1 | Flash access control register (FLASH_ACR) | 174 |
| 4.7.2 | Flash Power-down key register (FLASH_PDKEYR) | 175 |
| 4.7.3 | Flash key register (FLASH_KEYR) | 176 |
| 4.7.4 | Flash option key register (FLASH_OPTKEYR) | 176 |
| 4.7.5 | Flash status register (FLASH_SR) | 177 |
| 4.7.6 | Flash control register (FLASH_CR) | 178 |
| 4.7.7 | Flash ECC register (FLASH_ECCR) | 180 |
| 4.7.8 | Flash option register (FLASH_OPTR) | 181 |
| 4.7.9 | Flash PCROP1 Start address register (FLASH_PCROP1SR) | 183 |

| | | |
|----------|---|------------|
| 4.7.10 | Flash PCROP1 End address register (FLASH_PCROP1ER) | 184 |
| 4.7.11 | Flash WRP area A address register (FLASH_WRP1AR) | 184 |
| 4.7.12 | Flash WRP area B address register (FLASH_WRP1BR) | 185 |
| 4.7.13 | Flash Securable area register (FLASH_SEC1R) | 185 |
| 4.7.14 | FLASH register map | 187 |
| 5 | Embedded Flash memory (FLASH) for category 2 devices | 189 |
| 5.1 | Introduction | 189 |
| 5.2 | FLASH main features | 189 |
| 5.3 | FLASH functional description | 190 |
| 5.3.1 | Flash memory organization | 190 |
| 5.3.2 | Error code correction (ECC) | 191 |
| 5.3.3 | Read access latency | 191 |
| 5.3.4 | Adaptive real-time memory accelerator (ART Accelerator) | 192 |
| 5.3.5 | Flash program and erase operations | 195 |
| 5.3.6 | Flash main memory erase sequences | 196 |
| 5.3.7 | Flash main memory programming sequences | 197 |
| 5.4 | FLASH option bytes | 201 |
| 5.4.1 | Option bytes description | 201 |
| 5.4.2 | Option bytes programming | 205 |
| 5.5 | FLASH memory protection | 207 |
| 5.5.1 | Read protection (RDP) | 207 |
| 5.5.2 | Proprietary code readout protection (PCROP) | 210 |
| 5.5.3 | Write protection (WRP) | 211 |
| 5.5.4 | Securable memory area | 212 |
| 5.5.5 | Disabling core debug access | 213 |
| 5.5.6 | Forcing boot from Flash memory | 213 |
| 5.6 | FLASH interrupts | 214 |
| 5.7 | FLASH registers | 215 |
| 5.7.1 | Flash access control register (FLASH_ACR) | 215 |
| 5.7.2 | Flash Power-down key register (FLASH_PDKEYR) | 216 |
| 5.7.3 | Flash key register (FLASH_KEYR) | 217 |
| 5.7.4 | Flash option key register (FLASH_OPTKEYR) | 217 |
| 5.7.5 | Flash status register (FLASH_SR) | 218 |
| 5.7.6 | Flash control register (FLASH_CR) | 219 |

| | | |
|----------|---|------------|
| 5.7.7 | Flash ECC register (FLASH_ECCR) | 221 |
| 5.7.8 | Flash option register (FLASH_OPTR) | 222 |
| 5.7.9 | Flash PCROP1 Start address register (FLASH_PCROP1SR) | 224 |
| 5.7.10 | Flash PCROP1 End address register (FLASH_PCROP1ER) | 225 |
| 5.7.11 | Flash WRP area A address register (FLASH_WRP1AR) | 225 |
| 5.7.12 | Flash WRP area B address register (FLASH_WRP1BR) | 226 |
| 5.7.13 | Flash Securable area register (FLASH_SEC1R) | 226 |
| 5.7.14 | FLASH register map | 228 |
| 6 | Power control (PWR) | 230 |
| 6.1 | Power supplies | 230 |
| 6.1.1 | Independent analog peripherals supply | 231 |
| 6.1.2 | USB transceivers supply | 232 |
| 6.1.3 | Battery backup domain | 232 |
| 6.1.4 | Voltage regulator | 233 |
| 6.1.5 | Dynamic voltage scaling management | 234 |
| 6.2 | Power supply supervisor | 236 |
| 6.2.1 | Power-on reset (POR) / power-down reset (PDR) / brown-out reset (BOR) | 236 |
| 6.2.2 | Programmable voltage detector (PWD) | 237 |
| 6.2.3 | Peripheral Voltage Monitoring (PVM) | 238 |
| 6.3 | Low-power modes | 239 |
| 6.3.1 | Run mode | 244 |
| 6.3.2 | Low-power run mode (LP run) | 244 |
| 6.3.3 | Low power modes | 245 |
| 6.3.4 | Sleep mode | 246 |
| 6.3.5 | Low-power sleep mode (LP sleep) | 247 |
| 6.3.6 | Stop 0 mode | 248 |
| 6.3.7 | Stop 1 mode | 250 |
| 6.3.8 | Standby mode | 251 |
| 6.3.9 | Shutdown mode | 254 |
| 6.3.10 | Auto-wake up from low-power mode | 255 |
| 6.4 | PWR registers | 256 |
| 6.4.1 | Power control register 1 (PWR_CR1) | 256 |
| 6.4.2 | Power control register 2 (PWR_CR2) | 257 |
| 6.4.3 | Power control register 3 (PWR_CR3) | 257 |
| 6.4.4 | Power control register 4 (PWR_CR4) | 259 |

| | | |
|----------|---|------------|
| 6.4.5 | Power status register 1 (PWR_SR1) | 260 |
| 6.4.6 | Power status register 2 (PWR_SR2) | 261 |
| 6.4.7 | Power status clear register (PWR_SCR) | 262 |
| 6.4.8 | Power Port A pull-up control register (PWR_PUCRA) | 263 |
| 6.4.9 | Power Port A pull-down control register (PWR_PDCRA) | 263 |
| 6.4.10 | Power Port B pull-up control register (PWR_PUCRB) | 264 |
| 6.4.11 | Power Port B pull-down control register (PWR_PDCRB) | 264 |
| 6.4.12 | Power Port C pull-up control register (PWR_PUCRC) | 265 |
| 6.4.13 | Power Port C pull-down control register (PWR_PDCRC) | 265 |
| 6.4.14 | Power Port D pull-up control register (PWR_PUCRD) | 266 |
| 6.4.15 | Power Port D pull-down control register (PWR_PDCRD) | 266 |
| 6.4.16 | Power Port E pull-up control register (PWR_PUCRE) | 267 |
| 6.4.17 | Power Port E pull-down control register (PWR_PDCRE) | 267 |
| 6.4.18 | Power Port F pull-up control register (PWR_PUCRF) | 268 |
| 6.4.19 | Power Port F pull-down control register (PWR_PDCRF) | 268 |
| 6.4.20 | Power Port G pull-up control register (PWR_PUCRG) | 269 |
| 6.4.21 | Power Port G pull-down control register (PWR_PDCRG) | 269 |
| 6.4.22 | Power control register (PWR_CR5) | 270 |
| 6.4.23 | PWR register map and reset value table | 271 |
| 7 | Reset and clock control (RCC) | 273 |
| 7.1 | Reset | 273 |
| 7.1.1 | Power reset | 273 |
| 7.1.2 | System reset | 273 |
| 7.1.3 | RTC domain reset | 275 |
| 7.2 | Clocks | 275 |
| 7.2.1 | HSE clock | 279 |
| 7.2.2 | HSI16 clock | 280 |
| 7.2.3 | HSI48 clock | 281 |
| 7.2.4 | PLL | 281 |
| 7.2.5 | LSE clock | 281 |
| 7.2.6 | LSI clock | 282 |
| 7.2.7 | System clock (SYSCLK) selection | 282 |
| 7.2.8 | Clock source frequency versus voltage scaling | 283 |
| 7.2.9 | Clock security system (CSS) | 283 |
| 7.2.10 | Clock security system on LSE | 283 |
| 7.2.11 | ADC clock | 284 |

| | | |
|--------|--|-----|
| 7.2.12 | RTC clock | 284 |
| 7.2.13 | Timer clock | 285 |
| 7.2.14 | Watchdog clock | 285 |
| 7.2.15 | Clock-out capability | 285 |
| 7.2.16 | Internal/external clock measurement with TIM5/TIM15/TIM16/TIM17 | 285 |
| 7.2.17 | Peripheral clock enable register (RCC_AHBxENR, RCC_APBxENRy) | 288 |
| 7.3 | Low-power modes | 288 |
| 7.4 | RCC registers | 290 |
| 7.4.1 | Clock control register (RCC_CR) | 290 |
| 7.4.2 | Internal clocksources calibration register (RCC_ICSCR) | 291 |
| 7.4.3 | Clock configuration register (RCC_CFGR) | 292 |
| 7.4.4 | PLL configuration register (RCC_PLLCFGR) | 294 |
| 7.4.5 | Clock interrupt enable register (RCC_CIER) | 297 |
| 7.4.6 | Clock interrupt flag register (RCC_CIFR) | 298 |
| 7.4.7 | Clock interrupt clear register (RCC_CICR) | 299 |
| 7.4.8 | AHB1 peripheral reset register (RCC_AHB1RSTR) | 300 |
| 7.4.9 | AHB2 peripheral reset register (RCC_AHB2RSTR) | 302 |
| 7.4.10 | AHB3 peripheral reset register (RCC_AHB3RSTR) | 304 |
| 7.4.11 | APB1 peripheral reset register 1 (RCC_APB1RSTR1) | 304 |
| 7.4.12 | APB1 peripheral reset register 2 (RCC_APB1RSTR2) | 307 |
| 7.4.13 | APB2 peripheral reset register (RCC_APB2RSTR) | 307 |
| 7.4.14 | AHB1 peripheral clock enable register (RCC_AHB1ENR) | 309 |
| 7.4.15 | AHB2 peripheral clock enable register (RCC_AHB2ENR) | 310 |
| 7.4.16 | AHB3 peripheral clock enable register (RCC_AHB3ENR) | 312 |
| 7.4.17 | APB1 peripheral clock enable register 1 (RCC_APB1ENR1) | 313 |
| 7.4.18 | APB1 peripheral clock enable register 2 (RCC_APB1ENR2) | 315 |
| 7.4.19 | APB2 peripheral clock enable register (RCC_APB2ENR) | 317 |
| 7.4.20 | AHB1 peripheral clocks enable in Sleep and Stop modes register (RCC_AHB1SMENR) | 318 |
| 7.4.21 | AHB2 peripheral clocks enable in Sleep and Stop modes register (RCC_AHB2SMENR) | 320 |
| 7.4.22 | AHB3 peripheral clocks enable in Sleep and Stop modes register (RCC_AHB3SMENR) | 322 |
| 7.4.23 | APB1 peripheral clocks enable in Sleep and Stop modes register 1 (RCC_APB1SMENR1) | 322 |
| 7.4.24 | APB1 peripheral clocks enable in Sleep and Stop modes register 2 (RCC_APB1SMENR2) | 325 |

| | | |
|----------|--|------------|
| 7.4.25 | APB2 peripheral clocks enable in Sleep and Stop modes register (RCC_APB2SMENR) | 326 |
| 7.4.26 | Peripherals independent clock configuration register (RCC_CCIPR) | 327 |
| 7.4.27 | RTC domain control register (RCC_BDCR) | 330 |
| 7.4.28 | Control/status register (RCC_CSR) | 332 |
| 7.4.29 | Clock recovery RC register (RCC_CRRCR) | 334 |
| 7.4.30 | Peripherals independent clock configuration register (RCC_CCIPR2) | 335 |
| 7.4.31 | RCC register map | 335 |
| 8 | Clock recovery system (CRS) | 340 |
| 8.1 | Introduction | 340 |
| 8.2 | CRS main features | 340 |
| 8.3 | CRS implementation | 340 |
| 8.4 | CRS functional description | 341 |
| 8.4.1 | CRS block diagram | 341 |
| 8.4.2 | Synchronization input | 341 |
| 8.4.3 | Frequency error measurement | 342 |
| 8.4.4 | Frequency error evaluation and automatic trimming | 342 |
| 8.4.5 | CRS initialization and configuration | 343 |
| 8.5 | CRS low-power modes | 344 |
| 8.6 | CRS interrupts | 344 |
| 8.7 | CRS registers | 345 |
| 8.7.1 | CRS control register (CRS_CR) | 345 |
| 8.7.2 | CRS configuration register (CRS_CFGR) | 346 |
| 8.7.3 | CRS interrupt and status register (CRS_ISR) | 347 |
| 8.7.4 | CRS interrupt flag clear register (CRS_ICR) | 349 |
| 8.7.5 | CRS register map | 349 |
| 9 | General-purpose I/Os (GPIO) | 351 |
| 9.1 | Introduction | 351 |
| 9.2 | GPIO main features | 351 |
| 9.3 | GPIO functional description | 351 |
| 9.3.1 | General-purpose I/O (GPIO) | 354 |
| 9.3.2 | I/O pin alternate function multiplexer and mapping | 354 |
| 9.3.3 | I/O port control registers | 355 |
| 9.3.4 | I/O port data registers | 355 |

| | | |
|--------|--|------------|
| 9.3.5 | I/O data bitwise handling | 355 |
| 9.3.6 | GPIO locking mechanism | 356 |
| 9.3.7 | I/O alternate function input/output | 356 |
| 9.3.8 | External interrupt/wakeup lines | 356 |
| 9.3.9 | Input configuration | 357 |
| 9.3.10 | Output configuration | 357 |
| 9.3.11 | Alternate function configuration | 358 |
| 9.3.12 | Analog configuration | 359 |
| 9.3.13 | Using the HSE or LSE oscillator pins as GPIOs | 359 |
| 9.3.14 | Using the GPIO pins in the RTC supply domain | 359 |
| 9.3.15 | Using PB8 as GPIO | 360 |
| 9.3.16 | Using PG10 as GPIO | 360 |
| 9.4 | GPIO registers | 361 |
| 9.4.1 | GPIO port mode register (GPIOx_MODER) (x = A to G) | 361 |
| 9.4.2 | GPIO port output type register (GPIOx_OTYPER) (x = A to G) | 361 |
| 9.4.3 | GPIO port output speed register (GPIOx_OSPEEDR) (x = A to G) | 362 |
| 9.4.4 | GPIO port pull-up/pull-down register (GPIOx_PUPDR) (x = A to G) | 362 |
| 9.4.5 | GPIO port input data register (GPIOx_IDR) (x = A to G) | 363 |
| 9.4.6 | GPIO port output data register (GPIOx_ODR) (x = A to G) | 363 |
| 9.4.7 | GPIO port bit set/reset register (GPIOx_BSRR) (x = A to G) | 363 |
| 9.4.8 | GPIO port configuration lock register (GPIOx_LCKR) (x = A to G) | 364 |
| 9.4.9 | GPIO alternate function low register (GPIOx_AFRL) (x = A to G) | 365 |
| 9.4.10 | GPIO alternate function high register (GPIOx_AFRH) (x = A to G) | 366 |
| 9.4.11 | GPIO port bit reset register (GPIOx_BRR) (x = A to G) | 367 |
| 9.4.12 | GPIO register map | 368 |
| 10 | System configuration controller (SYSCFG) | 370 |
| 10.1 | SYSCFG main features | 370 |
| 10.2 | SYSCFG registers | 370 |
| 10.2.1 | SYSCFG memory remap register (SYSCFG_MEMRMP) | 370 |

| | | |
|-----------|---|------------|
| 10.2.2 | SYSCFG configuration register 1 (SYSCFG_CFGR1) | 371 |
| 10.2.3 | SYSCFG external interrupt configuration register 1 (SYSCFG_EXTICR1) | 373 |
| 10.2.4 | SYSCFG external interrupt configuration register 2 (SYSCFG_EXTICR2) | 374 |
| 10.2.5 | SYSCFG external interrupt configuration register 3 (SYSCFG_EXTICR3) | 375 |
| 10.2.6 | SYSCFG external interrupt configuration register 4 (SYSCFG_EXTICR4) | 377 |
| 10.2.7 | SYSCFG CCM SRAM control and status register (SYSCFG_SC_SR) | 378 |
| 10.2.8 | SYSCFG configuration register 2 (SYSCFG_CFGR2) | 379 |
| 10.2.9 | SYSCFG CCM SRAM write protection register (SYSCFG_SWPR) . | 380 |
| 10.2.10 | SYSCFG CCM SRAM key register (SYSCFG_SKR) | 380 |
| 10.2.11 | SYSCFG register map | 381 |
| 11 | Peripherals interconnect matrix | 382 |
| 11.1 | Introduction | 382 |
| 11.2 | Connection summary | 383 |
| 11.3 | Interconnection details | 385 |
| 11.3.1 | From timer (TIMx, HRTIM) to timer (TIMx) | 385 |
| 11.3.2 | From timer (TIMx, HRTIM) and EXTI to ADC (ADCx) | 387 |
| 11.3.3 | From ADC (ADCx) to timer (TIMx, HRTIM) | 389 |
| 11.3.4 | From timer (TIMx, HRTIM) and EXTI to DAC (DACx) | 389 |
| 11.3.5 | From HSE, LSE, LSI, HSI16, MCO, RTC to timer (TIMx) | 390 |
| 11.3.6 | From RTC, COMPx to low-power timer (LPTIM1) | 391 |
| 11.3.7 | From timer (TIMx) to comparators (COMPx) | 392 |
| 11.3.8 | From internal analog source to ADC (ADCx), comparator (COMPx) and OPAMP (OPAMPx) | 392 |
| 11.3.9 | From comparators (COMPx) to timers (TIMx, HRTIM) | 396 |
| 11.3.10 | From system errors to timers (TIMx) and HRTIM | 402 |
| 11.3.11 | From timers (TIM16/TIM17) to IRTIM | 402 |
| 12 | Direct memory access controller (DMA) | 403 |
| 12.1 | Introduction | 403 |
| 12.2 | DMA main features | 403 |
| 12.3 | DMA implementation | 404 |
| 12.3.1 | DMA1 and DMA2 | 404 |
| 12.3.2 | DMA request mapping | 404 |

| | | |
|-----------|---|------------|
| 12.4 | DMA functional description | 404 |
| 12.4.1 | DMA block diagram | 404 |
| 12.4.2 | DMA pins and internal signals | 406 |
| 12.4.3 | DMA transfers | 406 |
| 12.4.4 | DMA arbitration | 407 |
| 12.4.5 | DMA channels | 407 |
| 12.4.6 | DMA data width, alignment and endianness | 411 |
| 12.4.7 | DMA error management | 412 |
| 12.5 | DMA interrupts | 413 |
| 12.6 | DMA registers | 413 |
| 12.6.1 | DMA interrupt status register (DMA_ISR) | 413 |
| 12.6.2 | DMA interrupt flag clear register (DMA_IFCR) | 416 |
| 12.6.3 | DMA channel x configuration register (DMA_CCRx) | 417 |
| 12.6.4 | DMA channel x number of data to transfer register (DMA_CNDTRx) . | 420 |
| 12.6.5 | DMA channel x peripheral address register (DMA_CPARx) | 420 |
| 12.6.6 | DMA channel x memory address register (DMA_CMARx) | 421 |
| 12.6.7 | DMA register map | 421 |
| 13 | DMA request multiplexer (DMAMUX) | 424 |
| 13.1 | Introduction | 424 |
| 13.2 | DMAMUX main features | 425 |
| 13.3 | DMAMUX implementation | 425 |
| 13.3.1 | DMAMUX instantiation | 425 |
| 13.3.2 | DMAMUX mapping | 425 |
| 13.4 | DMAMUX functional description | 428 |
| 13.4.1 | DMAMUX block diagram | 428 |
| 13.4.2 | DMAMUX signals | 428 |
| 13.4.3 | DMAMUX channels | 429 |
| 13.4.4 | DMAMUX request line multiplexer | 429 |
| 13.4.5 | DMAMUX request generator | 432 |
| 13.5 | DMAMUX interrupts | 433 |
| 13.6 | DMAMUX registers | 434 |
| 13.6.1 | DMAMUX request line multiplexer channel x configuration register (DMAMUX_CxCR) | 434 |
| 13.6.2 | DMAMUX request line multiplexer interrupt channel status register (DMAMUX_CSR) | 435 |

| | | |
|-----------|---|------------|
| 13.6.3 | DMAMUX request line multiplexer interrupt clear flag register (DMAMUX_CFR) | 435 |
| 13.6.4 | DMAMUX request generator channel x configuration register (DMAMUX_RGxCR) | 436 |
| 13.6.5 | DMAMUX request generator interrupt status register (DMAMUX_RGSR) | 437 |
| 13.6.6 | DMAMUX request generator interrupt clear flag register (DMAMUX_RGCFR) | 437 |
| 13.6.7 | DMAMUX register map | 438 |
| 14 | Nested vectored interrupt controller (NVIC) | 440 |
| 14.1 | NVIC main features | 440 |
| 14.2 | SysTick calibration value register | 440 |
| 14.3 | Interrupt and exception vectors | 441 |
| 15 | Extended interrupts and events controller (EXTI) | 445 |
| 15.1 | Introduction | 445 |
| 15.2 | EXTI main features | 445 |
| 15.3 | EXTI functional description | 445 |
| 15.3.1 | EXTI block diagram | 446 |
| 15.3.2 | Wake up event management | 446 |
| 15.3.3 | Peripherals asynchronous Interrupts | 447 |
| 15.3.4 | Hardware interrupt selection | 447 |
| 15.3.5 | Hardware event selection | 447 |
| 15.3.6 | Software interrupt/event selection | 447 |
| 15.4 | EXTI interrupt/event line mapping | 447 |
| 15.5 | EXTI registers | 450 |
| 15.5.1 | Interrupt mask register 1 (EXTI_IMR1) | 450 |
| 15.5.2 | Event mask register 1 (EXTI_EMR1) | 450 |
| 15.5.3 | Rising trigger selection register 1 (EXTI_RTSR1) | 451 |
| 15.5.4 | Falling trigger selection register 1 (EXTI_FTSR1) | 451 |
| 15.5.5 | Software interrupt event register 1 (EXTI_SWIER1) | 452 |
| 15.5.6 | Pending register 1 (EXTI_PR1) | 454 |
| 15.5.7 | Interrupt mask register 2 (EXTI_IMR2) | 454 |
| 15.5.8 | Event mask register 2 (EXTI_EMR2) | 455 |
| 15.5.9 | Rising trigger selection register 2 (EXTI_RTSR2) | 455 |
| 15.5.10 | Falling trigger selection register 2 (EXTI_FTSR2) | 457 |

| | | |
|-----------|---|------------|
| 15.5.11 | Software interrupt event register 2 (EXTI_SWIER2) | 457 |
| 15.5.12 | Pending register 2 (EXTI_PR2) | 458 |
| 15.5.13 | EXTI register map | 459 |
| 16 | Cyclic redundancy check calculation unit (CRC) | 460 |
| 16.1 | Introduction | 460 |
| 16.2 | CRC main features | 460 |
| 16.3 | CRC functional description | 461 |
| 16.3.1 | CRC block diagram | 461 |
| 16.3.2 | CRC internal signals | 461 |
| 16.3.3 | CRC operation | 461 |
| 16.4 | CRC registers | 463 |
| 16.4.1 | CRC data register (CRC_DR) | 463 |
| 16.4.2 | CRC independent data register (CRC_IDR) | 463 |
| 16.4.3 | CRC control register (CRC_CR) | 464 |
| 16.4.4 | CRC initial value (CRC_INIT) | 465 |
| 16.4.5 | CRC polynomial (CRC_POL) | 465 |
| 16.4.6 | CRC register map | 466 |
| 17 | CORDIC co-processor (CORDIC) | 467 |
| 17.1 | CORDIC introduction | 467 |
| 17.2 | CORDIC main features | 467 |
| 17.3 | CORDIC functional description | 467 |
| 17.3.1 | General description | 467 |
| 17.3.2 | CORDIC functions | 467 |
| 17.3.3 | Fixed point representation | 474 |
| 17.3.4 | Scaling factor | 474 |
| 17.3.5 | Precision | 475 |
| 17.3.6 | Zero-overhead mode | 478 |
| 17.3.7 | Polling mode | 479 |
| 17.3.8 | Interrupt mode | 480 |
| 17.3.9 | DMA mode | 480 |
| 17.4 | CORDIC registers | 481 |
| 17.4.1 | CORDIC control/status register (CORDIC_CSR) | 481 |
| 17.4.2 | CORDIC argument register (CORDIC_WDATA) | 483 |
| 17.4.3 | CORDIC result register (CORDIC_RDATA) | 484 |

| | | |
|-----------|---|------------|
| 17.4.4 | CORDIC register map | 484 |
| 18 | Filter math accelerator (FMAC) | 485 |
| 18.1 | FMAC introduction | 485 |
| 18.2 | FMAC main features | 485 |
| 18.3 | FMAC functional description | 486 |
| 18.3.1 | General description | 486 |
| 18.3.2 | Local memory and buffers | 487 |
| 18.3.3 | Input buffers | 487 |
| 18.3.4 | Output buffer | 490 |
| 18.3.5 | Initialization functions | 492 |
| 18.3.6 | Filter functions | 493 |
| 18.3.7 | Fixed point representation | 497 |
| 18.3.8 | Implementing FIR filters with the FMAC | 497 |
| 18.3.9 | Implementing IIR filters with the FMAC | 499 |
| 18.3.10 | Examples of filter initialization | 501 |
| 18.3.11 | Examples of filter operation | 502 |
| 18.3.12 | Filter design tips | 504 |
| 18.4 | FMAC registers | 505 |
| 18.4.1 | FMAC X1 buffer configuration register (FMAC_X1BUFCFG) | 505 |
| 18.4.2 | FMAC X2 buffer configuration register (FMAC_X2BUFCFG) | 505 |
| 18.4.3 | FMAC Y buffer configuration register (FMAC_YBUFCFG) | 506 |
| 18.4.4 | FMAC parameter register (FMAC_PARAM) | 507 |
| 18.4.5 | FMAC control register (FMAC_CR) | 508 |
| 18.4.6 | FMAC status register (FMAC_SR) | 509 |
| 18.4.7 | FMAC write data register (FMAC_WDATA) | 510 |
| 18.4.8 | FMAC read data register (FMAC_RDATA) | 511 |
| 18.4.9 | FMAC register map | 511 |
| 19 | Flexible static memory controller (FSMC) | 513 |
| 19.1 | Introduction | 513 |
| 19.2 | FMC main features | 513 |
| 19.3 | FMC block diagram | 514 |
| 19.4 | AHB interface | 514 |
| 19.4.1 | Supported memories and transactions | 515 |
| 19.5 | External device address mapping | 516 |

| | | |
|---------|--|------------|
| 19.5.1 | NOR/PSRAM address mapping | 516 |
| 19.5.2 | NAND Flash memory address mapping | 517 |
| 19.6 | NOR Flash/PSRAM controller | 518 |
| 19.6.1 | External memory interface signals | 519 |
| 19.6.2 | Supported memories and transactions | 521 |
| 19.6.3 | General timing rules | 523 |
| 19.6.4 | NOR Flash/PSRAM controller asynchronous transactions | 523 |
| 19.6.5 | Synchronous transactions | 541 |
| 19.6.6 | NOR/PSRAM controller registers | 548 |
| 19.7 | NAND Flash controller | 556 |
| 19.7.1 | External memory interface signals | 556 |
| 19.7.2 | NAND Flash supported memories and transactions | 557 |
| 19.7.3 | Timing diagrams for NAND Flash memory | 558 |
| 19.7.4 | NAND Flash operations | 559 |
| 19.7.5 | NAND Flash prewait functionality | 559 |
| 19.7.6 | Computation of the error correction code (ECC) in NAND Flash memory | 560 |
| 19.7.7 | NAND Flash controller registers | 561 |
| 19.7.8 | FMC register map | 567 |
| 20 | Quad-SPI interface (QUADSPI) | 569 |
| 20.1 | Introduction | 569 |
| 20.2 | QUADSPI main features | 569 |
| 20.3 | QUADSPI functional description | 569 |
| 20.3.1 | QUADSPI block diagram | 569 |
| 20.3.2 | QUADSPI pins | 570 |
| 20.3.3 | QUADSPI command sequence | 571 |
| 20.3.4 | QUADSPI signal interface protocol modes | 573 |
| 20.3.5 | QUADSPI indirect mode | 575 |
| 20.3.6 | QUADSPI status flag polling mode | 576 |
| 20.3.7 | QUADSPI memory-mapped mode | 577 |
| 20.3.8 | QUADSPI Flash memory configuration | 578 |
| 20.3.9 | QUADSPI delayed data sampling | 578 |
| 20.3.10 | QUADSPI configuration | 578 |
| 20.3.11 | QUADSPI usage | 579 |
| 20.3.12 | Sending the instruction only once | 581 |
| 20.3.13 | QUADSPI error management | 581 |

| | | |
|-----------|---|------------|
| 20.3.14 | QUADSPI busy bit and abort functionality | 581 |
| 20.3.15 | nCS behavior | 582 |
| 20.4 | QUADSPI interrupts | 583 |
| 20.5 | QUADSPI registers | 584 |
| 20.5.1 | QUADSPI control register (QUADSPI_CR) | 584 |
| 20.5.2 | QUADSPI device configuration register (QUADSPI_DCR) | 587 |
| 20.5.3 | QUADSPI status register (QUADSPI_SR) | 588 |
| 20.5.4 | QUADSPI flag clear register (QUADSPI_FCR) | 589 |
| 20.5.5 | QUADSPI data length register (QUADSPI_DLR) | 589 |
| 20.5.6 | QUADSPI communication configuration register (QUADSPI_CCR) | 590 |
| 20.5.7 | QUADSPI address register (QUADSPI_AR) | 592 |
| 20.5.8 | QUADSPI alternate bytes registers (QUADSPI_ABR) | 593 |
| 20.5.9 | QUADSPI data register (QUADSPI_DR) | 593 |
| 20.5.10 | QUADSPI polling status mask register (QUADSPI_PSMKR) | 594 |
| 20.5.11 | QUADSPI polling status match register (QUADSPI_PSMAR) | 594 |
| 20.5.12 | QUADSPI polling interval register (QUADSPI_PIR) | 595 |
| 20.5.13 | QUADSPI low-power timeout register (QUADSPI_LPTR) | 595 |
| 20.5.14 | QUADSPI register map | 595 |
| 21 | Analog-to-digital converters (ADC) | 597 |
| 21.1 | Introduction | 597 |
| 21.2 | ADC main features | 598 |
| 21.3 | ADC implementation | 599 |
| 21.4 | ADC functional description | 600 |
| 21.4.1 | ADC block diagram | 600 |
| 21.4.2 | ADC pins and internal signals | 601 |
| 21.4.3 | ADC clocks | 602 |
| 21.4.4 | ADC1/2/3/4/5 connectivity | 604 |
| 21.4.5 | Slave AHB interface | 609 |
| 21.4.6 | ADC Deep-power-down mode (DEEPPWD) and ADC voltage regulator (ADVREGEN) | 609 |
| 21.4.7 | Single-ended and differential input channels | 610 |
| 21.4.8 | Calibration (ADCAL, ADCALDIF, ADC_CALFACT) | 610 |
| 21.4.9 | ADC on-off control (ADEN, ADDIS, ADRDY) | 613 |
| 21.4.10 | Constraints when writing the ADC control bits | 614 |
| 21.4.11 | Channel selection (SQRx, JSQRx) | 615 |
| 21.4.12 | Channel-wise programmable sampling time (SMPR1, SMPR2) | 616 |

| | | |
|---------|--|-----|
| 21.4.13 | Single conversion mode (CONT = 0) | 618 |
| 21.4.14 | Continuous conversion mode (CONT = 1) | 618 |
| 21.4.15 | Starting conversions (ADSTART, JADSTART) | 619 |
| 21.4.16 | ADC timing | 620 |
| 21.4.17 | Stopping an ongoing conversion (ADSTP, JADSTP) | 620 |
| 21.4.18 | Conversion on external trigger and trigger polarity (EXTSEL, EXTEN, JEXTSEL, JEXTEN) | 622 |
| 21.4.19 | Injected channel management | 627 |
| 21.4.20 | Discontinuous mode (DISCEN, DISC NUM, JDISC EN) | 629 |
| 21.4.21 | Queue of context for injected conversions | 630 |
| 21.4.22 | Programmable resolution (RES) - Fast conversion mode | 638 |
| 21.4.23 | End of conversion, end of sampling phase (EOC, JEOC, EOSMP) | 639 |
| 21.4.24 | End of conversion sequence (EOS, JEOS) | 639 |
| 21.4.25 | Timing diagrams example (single/continuous modes, hardware/software triggers) | 640 |
| 21.4.26 | Data management | 642 |
| 21.4.27 | Dynamic low-power features | 648 |
| 21.4.28 | Analog window watchdog (AWD1EN, JAWD1EN, AWD1SGL, AWD1CH, AWD2CH, AWD3CH, AWD_HTx, AWD_LTx, AWDx) | 653 |
| 21.4.29 | Oversampler | 657 |
| 21.4.30 | Dual ADC modes | 663 |
| 21.4.31 | Temperature sensor | 677 |
| 21.4.32 | VBAT supply monitoring | 679 |
| 21.4.33 | Monitoring the internal voltage reference | 680 |
| 21.5 | ADC in low-power mode | 681 |
| 21.6 | ADC interrupts | 682 |
| 21.7 | ADC registers (for each ADC) | 683 |
| 21.7.1 | ADC interrupt and status register (ADC_ISR) | 683 |
| 21.7.2 | ADC interrupt enable register (ADC_IER) | 685 |
| 21.7.3 | ADC control register (ADC_CR) | 687 |
| 21.7.4 | ADC configuration register (ADC_CFGR) | 690 |
| 21.7.5 | ADC configuration register 2 (ADC_CFGR2) | 694 |
| 21.7.6 | ADC sample time register 1 (ADC_SMPR1) | 697 |
| 21.7.7 | ADC sample time register 2 (ADC_SMPR2) | 697 |
| 21.7.8 | ADC watchdog threshold register 1 (ADC_TR1) | 698 |
| 21.7.9 | ADC watchdog threshold register 2 (ADC_TR2) | 699 |
| 21.7.10 | ADC watchdog threshold register 3 (ADC_TR3) | 700 |
| 21.7.11 | ADC regular sequence register 1 (ADC_SQR1) | 700 |

| | | |
|-----------|--|------------|
| 21.7.12 | ADC regular sequence register 2 (ADC_SQR2) | 701 |
| 21.7.13 | ADC regular sequence register 3 (ADC_SQR3) | 702 |
| 21.7.14 | ADC regular sequence register 4 (ADC_SQR4) | 703 |
| 21.7.15 | ADC regular data register (ADC_DR) | 704 |
| 21.7.16 | ADC injected sequence register (ADC_JSQR) | 704 |
| 21.7.17 | ADC offset y register (ADC_OFRy) | 707 |
| 21.7.18 | ADC injected channel y data register (ADC_JDRy) | 708 |
| 21.7.19 | ADC analog watchdog 2 configuration register (ADC_AWD2CR) | 709 |
| 21.7.20 | ADC analog watchdog 3 configuration register (ADC_AWD3CR) | 709 |
| 21.7.21 | ADC differential mode selection register (ADC_DIFSEL) | 710 |
| 21.7.22 | ADC calibration factors (ADC_CALFACT) | 711 |
| 21.7.23 | ADC Gain compensation Register (ADC_GCOMP) | 711 |
| 21.8 | ADC common registers | 712 |
| 21.8.1 | ADCx common status register (ADCx_CSR) (x = 12 or 345) | 712 |
| 21.8.2 | ADCx common control register (ADCx_CCR) (x = 12 or 345) | 713 |
| 21.8.3 | ADCx common regular data register for dual mode (ADCx_CDR) (x = 12 or 345) | 717 |
| 21.9 | ADC register map | 717 |
| 22 | Digital-to-analog converter (DAC) | 721 |
| 22.1 | Introduction | 721 |
| 22.2 | DAC main features | 721 |
| 22.3 | DAC implementation | 722 |
| 22.4 | DAC functional description | 723 |
| 22.4.1 | DAC block diagram | 723 |
| 22.4.2 | DAC pins and internal signals | 724 |
| 22.4.3 | DAC channel enable | 730 |
| 22.4.4 | DAC data format | 730 |
| 22.4.5 | DAC conversion | 732 |
| 22.4.6 | DAC output voltage | 733 |
| 22.4.7 | DAC trigger selection | 733 |
| 22.4.8 | DMA requests | 734 |
| 22.4.9 | Noise generation | 735 |
| 22.4.10 | Triangle-wave generation | 736 |
| 22.4.11 | DAC sawtooth wave generation | 737 |
| 22.4.12 | DAC channel modes | 738 |
| 22.4.13 | DAC channel buffer calibration | 741 |

| | | |
|---------|---|-----|
| 22.4.14 | Dual DAC channel conversion modes (if dual channels are available) | 743 |
| 22.5 | DAC in low-power modes | 748 |
| 22.6 | DAC interrupts | 748 |
| 22.7 | DAC registers | 749 |
| 22.7.1 | DAC control register (DAC_CR) | 749 |
| 22.7.2 | DAC software trigger register (DAC_SWTRGR) | 752 |
| 22.7.3 | DAC channel1 12-bit right-aligned data holding register (DAC_DHR12R1) | 753 |
| 22.7.4 | DAC channel1 12-bit left aligned data holding register (DAC_DHR12L1) | 754 |
| 22.7.5 | DAC channel1 8-bit right aligned data holding register (DAC_DHR8R1) | 754 |
| 22.7.6 | DAC channel2 12-bit right aligned data holding register (DAC_DHR12R2) | 755 |
| 22.7.7 | DAC channel2 12-bit left aligned data holding register (DAC_DHR12L2) | 755 |
| 22.7.8 | DAC channel2 8-bit right-aligned data holding register (DAC_DHR8R2) | 756 |
| 22.7.9 | Dual DAC 12-bit right-aligned data holding register (DAC_DHR12RD) | 756 |
| 22.7.10 | Dual DAC 12-bit left aligned data holding register (DAC_DHR12LD) | 757 |
| 22.7.11 | Dual DAC 8-bit right aligned data holding register (DAC_DHR8RD) | 757 |
| 22.7.12 | DAC channel1 data output register (DAC_DOR1) | 758 |
| 22.7.13 | DAC channel2 data output register (DAC_DOR2) | 758 |
| 22.7.14 | DAC status register (DAC_SR) | 759 |
| 22.7.15 | DAC calibration control register (DAC_CCR) | 760 |
| 22.7.16 | DAC mode control register (DAC_MCR) | 761 |
| 22.7.17 | DAC channel1 sample and hold sample time register (DAC_SHSR1) | 762 |
| 22.7.18 | DAC channel2 sample and hold sample time register (DAC_SHSR2) | 763 |
| 22.7.19 | DAC sample and hold time register (DAC_SHHR) | 763 |
| 22.7.20 | DAC sample and hold refresh time register (DAC_SHRR) | 764 |
| 22.7.21 | DAC channel1 sawtooth register (DAC_STR1) | 765 |
| 22.7.22 | DAC channel2 sawtooth register (DAC_STR2) | 765 |
| 22.7.23 | DAC sawtooth mode register (DAC_STMODR) | 766 |
| 22.7.24 | DAC register map | 768 |

| | | |
|-----------|---|------------|
| 23 | Voltage reference buffer (VREFBUF) | 770 |
| 23.1 | Introduction | 770 |
| 23.2 | VREFBUF functional description | 770 |
| 23.3 | VREFBUF trimming | 771 |
| 23.4 | VREFBUF registers | 771 |
| 23.4.1 | VREFBUF control and status register (VREFBUF_CSR) | 771 |
| 23.4.2 | VREFBUF calibration control register (VREFBUF_CCR) | 772 |
| 23.4.3 | VREFBUF register map | 772 |
| 24 | Comparator (COMP) | 773 |
| 24.1 | COMP introduction | 773 |
| 24.2 | COMP main features | 773 |
| 24.3 | COMP functional description | 773 |
| 24.3.1 | COMP block diagram | 773 |
| 24.3.2 | COMP pins and internal signals | 774 |
| 24.3.3 | COMP reset and clocks | 774 |
| 24.3.4 | COMP LOCK mechanism | 775 |
| 24.3.5 | COMP hysteresis | 775 |
| 24.3.6 | COMP output blanking | 775 |
| 24.4 | COMP low-power modes | 776 |
| 24.5 | COMP interrupts | 777 |
| 24.6 | COMP registers | 777 |
| 24.6.1 | Comparator x control and status register (COMP_CxCSR) | 777 |
| 24.6.2 | COMP register map | 779 |
| 25 | Operational amplifiers (OPAMP) | 781 |
| 25.1 | Introduction | 781 |
| 25.2 | OPAMP main features | 781 |
| 25.3 | OPAMP functional description | 781 |
| 25.3.1 | OPAMP output redirection to internal ADC channels | 781 |
| 25.3.2 | OPAMP reset and clocks | 781 |
| 25.3.3 | Initial configuration | 782 |
| 25.3.4 | Signal routing | 782 |
| 25.3.5 | OPAMP modes | 784 |
| 25.3.6 | OPAMP PGA gain | 789 |
| 25.3.7 | Calibration | 790 |

| | | |
|-----------|--|------------|
| 25.3.8 | Timer controlled Multiplexer mode | 791 |
| 25.4 | OPAMP low-power modes | 792 |
| 25.5 | OPAMP registers | 793 |
| 25.5.1 | OPAMP1 control/status register (OPAMP1_CSR) | 793 |
| 25.5.2 | OPAMP2 control/status register (OPAMP2_CSR) | 796 |
| 25.5.3 | OPAMP3 control/status register (OPAMP3_CSR) | 799 |
| 25.5.4 | OPAMP4 control/status register (OPAMP4_CSR) | 802 |
| 25.5.5 | OPAMP5 control/status register (OPAMP5_CSR) | 805 |
| 25.5.6 | OPAMP6 control/status register (OPAMP6_CSR) | 808 |
| 25.5.7 | OPAMP1 timer controlled mode register (OPAMPx_TCMR) (x = 1..6) | 811 |
| 25.5.8 | OPAMP2 timer controlled mode register (OPAMPx_TCMR) (x = 1..6) | 813 |
| 25.5.9 | OPAMP3 timer controlled mode register (OPAMPx_TCMR) (x = 1..6) | 815 |
| 25.5.10 | OPAMP4 timer controlled mode register (OPAMPx_TCMR) (x = 1..6) | 817 |
| 25.5.11 | OPAMP5 timer controlled mode register (OPAMPx_TCMR) (x = 1..6) | 819 |
| 25.5.12 | OPAMP6 timer controlled mode register (OPAMPx_TCMR) (x = 1..6) | 821 |
| 25.5.13 | OPAMP register map | 823 |
| 26 | True random number generator (RNG) | 825 |
| 26.1 | Introduction | 825 |
| 26.2 | RNG main features | 825 |
| 26.3 | RNG functional description | 826 |
| 26.3.1 | RNG block diagram | 826 |
| 26.3.2 | RNG internal signals | 826 |
| 26.3.3 | Random number generation | 827 |
| 26.3.4 | RNG initialization | 829 |
| 26.3.5 | RNG operation | 830 |
| 26.3.6 | RNG clocking | 831 |
| 26.3.7 | Error management | 831 |
| 26.3.8 | RNG low-power usage | 832 |
| 26.4 | RNG interrupts | 833 |
| 26.5 | RNG processing time | 833 |
| 26.6 | RNG entropy source validation | 833 |
| 26.6.1 | Introduction | 833 |
| 26.6.2 | Validation conditions | 833 |
| 26.6.3 | Data collection | 834 |
| 26.7 | RNG registers | 834 |

| | | |
|-----------|--|------------|
| 26.7.1 | RNG control register (RNG_CR) | 834 |
| 26.7.2 | RNG status register (RNG_SR) | 835 |
| 26.7.3 | RNG data register (RNG_DR) | 836 |
| 26.7.4 | RNG register map | 836 |
| 27 | High-resolution timer (HRTIM) | 837 |
| 27.1 | Introduction | 837 |
| 27.2 | Main features | 838 |
| 27.3 | Functional description | 839 |
| 27.3.1 | General description | 839 |
| 27.3.2 | HRTIM pins and internal signals | 841 |
| 27.3.3 | Clocks | 844 |
| 27.3.4 | Timer A..F timing units | 848 |
| 27.3.5 | Master timer | 869 |
| 27.3.6 | Up-down counting mode | 871 |
| 27.3.7 | Set / reset events priorities and narrow pulses management | 878 |
| 27.3.8 | External events global conditioning | 881 |
| 27.3.9 | External event filtering in timing units | 886 |
| 27.3.10 | Delayed protection | 892 |
| 27.3.11 | Register preload and update management | 898 |
| 27.3.12 | PWM mode with "greater than" comparison | 902 |
| 27.3.13 | Events propagation within or across multiple timers | 903 |
| 27.3.14 | Output management | 908 |
| 27.3.15 | Burst mode controller | 910 |
| 27.3.16 | Chopper | 918 |
| 27.3.17 | Fault protection | 919 |
| 27.3.18 | Auxiliary outputs | 924 |
| 27.3.19 | Synchronizing the HRTIM with other timers or HRTIM instances | 927 |
| 27.3.20 | ADC triggers | 930 |
| 27.3.21 | DAC triggers | 934 |
| 27.3.22 | Interrupts | 938 |
| 27.3.23 | DMA | 940 |
| 27.3.24 | HRTIM initialization | 944 |
| 27.3.25 | Debug | 945 |
| 27.4 | Application use cases | 946 |
| 27.4.1 | Buck converter | 946 |
| 27.4.2 | Buck converter with synchronous rectification | 947 |

| | | |
|---------|---|-----|
| 27.4.3 | Multiphase converters | 948 |
| 27.4.4 | Transition mode power factor correction | 949 |
| 27.5 | HRTIM registers | 952 |
| 27.5.1 | HRTIM master timer control register (HRTIM_MCR) | 952 |
| 27.5.2 | HRTIM master timer interrupt status register (HRTIM_MISR) | 955 |
| 27.5.3 | HRTIM master timer interrupt clear register (HRTIM_MICR) | 956 |
| 27.5.4 | HRTIM master timer DMA interrupt enable register (HRTIM_MDIER) | 957 |
| 27.5.5 | HRTIM master timer counter register (HRTIM_MCNTR) | 959 |
| 27.5.6 | HRTIM master timer period register (HRTIM_MP PER) | 959 |
| 27.5.7 | HRTIM master timer repetition register (HRTIM_MREP) | 960 |
| 27.5.8 | HRTIM master timer compare 1 register (HRTIM_MCMP1R) | 960 |
| 27.5.9 | HRTIM master timer compare 2 register (HRTIM_MCMP2R) | 961 |
| 27.5.10 | HRTIM master timer compare 3 register (HRTIM_MCMP3R) | 961 |
| 27.5.11 | HRTIM master timer compare 4 register (HRTIM_MCMP4R) | 962 |
| 27.5.12 | HRTIM timer x control register (HRTIM_TIMxCR) (x = A to F) | 963 |
| 27.5.13 | HRTIM timer x interrupt status register (HRTIM_TIMxISR) (x = A to F) | 968 |
| 27.5.14 | HRTIM timer x interrupt clear register (HRTIM_TIMxICR) (x = A to F) | 970 |
| 27.5.15 | HRTIM timer x DMA interrupt enable register (HRTIM_TIMxDIER) (x = A to F) | 971 |
| 27.5.16 | HRTIM timer x counter register (HRTIM_CNTxR) (x = A to F) | 975 |
| 27.5.17 | HRTIM timer x period register (HRTIM_PERxR) (x = A to F) | 975 |
| 27.5.18 | HRTIM timer x repetition register (HRTIM_REPxR) (x = A to F) | 977 |
| 27.5.19 | HRTIM timer x compare 1 register (HRTIM_CMP1xR) (x = A to F) | 977 |
| 27.5.20 | HRTIM timer x compare 1 compound register (HRTIM_CMP1CxR) (x = A to F) | 979 |
| 27.5.21 | HRTIM timer x compare 2 register (HRTIM_CMP2xR) (x = A to F) | 979 |
| 27.5.22 | HRTIM timer x compare 3 register (HRTIM_CMP3xR) (x = A to F) | 981 |
| 27.5.23 | HRTIM timer x compare 4 register (HRTIM_CMP4xR) (x = A to F) | 982 |
| 27.5.24 | HRTIM timer x capture 1 register (HRTIM_CPT1xR) (x = A to F) | 983 |
| 27.5.25 | HRTIM timer x capture 2 register (HRTIM_CPT2xR) (x = A to F) | 984 |
| 27.5.26 | HRTIM timer x deadtime register (HRTIM_DTxR) (x = A to F) | 985 |
| 27.5.27 | HRTIM timer x output 1 set register (HRTIM_SETx1R) (x = A to F) | 987 |
| 27.5.28 | HRTIM timer x output 1 reset register (HRTIM_RSTx1R) (x = A to F) | 990 |
| 27.5.29 | HRTIM timer x output 2 set register (HRTIM_SETx2R) (x = A to F) | 990 |
| 27.5.30 | HRTIM timer x output 2 reset register (HRTIM_RSTx2R) (x = A to F) | 991 |

| | | |
|---------|---|------|
| 27.5.31 | HRTIM timer x external event filtering register 1 (HRTIM_EEFxR1) (x = A to F) | 992 |
| 27.5.32 | HRTIM timer x external event filtering register 2 (HRTIM_EEFxR2) (x = A to F) | 994 |
| 27.5.33 | HRTIM timer A reset register (HRTIM_RSTAR) | 995 |
| 27.5.34 | HRTIM timer B reset register (HRTIM_RSTBR) | 997 |
| 27.5.35 | HRTIM timer C reset register (HRTIM_RSTCR) | 999 |
| 27.5.36 | HRTIM timer D reset register (HRTIM_RSTDR) | 1001 |
| 27.5.37 | HRTIM timer E reset register (HRTIM_RSTER) | 1003 |
| 27.5.38 | HRTIM timer F reset register (HRTIM_RSTFR) | 1005 |
| 27.5.39 | HRTIM timer x chopper register (HRTIM_CHPxR) (x = A to F) | 1007 |
| 27.5.40 | HRTIM timer A capture 1 control register (HRTIM_CPT1ACR) | 1008 |
| 27.5.41 | HRTIM timer B capture 1 control register (HRTIM_CPT1BCR) | 1008 |
| 27.5.42 | HRTIM timer C capture 1 control register (HRTIM_CPT1CCR) | 1008 |
| 27.5.43 | HRTIM timer D capture 1 control register (HRTIM_CPT1DCR) | 1009 |
| 27.5.44 | HRTIM timer E capture 1 control register (HRTIM_CPT1ECR) | 1009 |
| 27.5.45 | HRTIM timer F capture 1 control register (HRTIM_CPT1FCR) | 1009 |
| 27.5.46 | HRTIM timer x capture 2 control register (HRTIM_CPT2xCR) (x = A to F) | 1010 |
| 27.5.47 | HRTIM timer x output register (HRTIM_OUTTxR) (x = A to F) | 1016 |
| 27.5.48 | HRTIM timer x fault register (HRTIM_FLTxR) (x = A to F) | 1019 |
| 27.5.49 | HRTIM timer x control register 2 (HRTIM_TIMxCR2) (x = A to F) | 1020 |
| 27.5.50 | HRTIM timer x external event filtering register 3 (HRTIM_TIMxEEFR3) (x = A to F) | 1023 |
| 27.5.51 | HRTIM control register 1 (HRTIM_CR1) | 1024 |
| 27.5.52 | HRTIM control register 2 (HRTIM_CR2) | 1026 |
| 27.5.53 | HRTIM interrupt status register (HRTIM_ISR) | 1027 |
| 27.5.54 | HRTIM interrupt clear register (HRTIM_ICR) | 1029 |
| 27.5.55 | HRTIM interrupt enable register (HRTIM_IER) | 1030 |
| 27.5.56 | HRTIM output enable register (HRTIM_OENR) | 1031 |
| 27.5.57 | HRTIM output disable register (HRTIM_ODISR) | 1032 |
| 27.5.58 | HRTIM output disable status register (HRTIM_ODSR) | 1033 |
| 27.5.59 | HRTIM burst mode control register (HRTIM_BMCR) | 1034 |
| 27.5.60 | HRTIM burst mode trigger register (HRTIM_BMTRGR) | 1036 |
| 27.5.61 | HRTIM burst mode compare register (HRTIM_BMCMPR) | 1038 |
| 27.5.62 | HRTIM burst mode period register (HRTIM_BMPER) | 1038 |
| 27.5.63 | HRTIM timer external event control register 1 (HRTIM_EECR1) | 1039 |
| 27.5.64 | HRTIM timer external event control register 2 (HRTIM_EECR2) | 1041 |

| | | |
|-----------|---|-------------|
| 27.5.65 | HRTIM timer external event control register 3 (HRTIM_EECR3) | 1042 |
| 27.5.66 | HRTIM ADC trigger 1 register (HRTIM_ADC1R) | 1043 |
| 27.5.67 | HRTIM ADC trigger 2 register (HRTIM_ADC2R) | 1044 |
| 27.5.68 | HRTIM ADC trigger 3 register (HRTIM_ADC3R) | 1045 |
| 27.5.69 | HRTIM ADC trigger 4 register (HRTIM_ADC4R) | 1047 |
| 27.5.70 | HRTIM DLL control register (HRTIM_DLLCR) | 1049 |
| 27.5.71 | HRTIM fault input register 1 (HRTIM_FLTINR1) | 1050 |
| 27.5.72 | HRTIM fault input register 2 (HRTIM_FLTINR2) | 1052 |
| 27.5.73 | HRTIM burst DMA master timer update register (HRTIM_BDMUPR) . | 1054 |
| 27.5.74 | HRTIM burst DMA timer x update register (HRTIM_BDTxUPR) (x = A to F) | 1055 |
| 27.5.75 | HRTIM burst DMA data register (HRTIM_BDMADR) | 1056 |
| 27.5.76 | HRTIM ADC extended trigger register (HRTIM_ADCER) | 1057 |
| 27.5.77 | HRTIM ADC trigger update register (HRTIM_ADCUR) | 1060 |
| 27.5.78 | HRTIM ADC postscaler register 1 (HRTIM_ADCPS1) | 1061 |
| 27.5.79 | HRTIM ADC postscaler register 2 (HRTIM_ADCPS2) | 1062 |
| 27.5.80 | HRTIM fault input register 3 (HRTIM_FLTINR3) | 1063 |
| 27.5.81 | HRTIM fault input register 4 (HRTIM_FLTINR4) | 1065 |
| 27.5.82 | HRTIM register map | 1067 |
| 28 | Advanced-control timers (TIM1/TIM8/TIM20) | 1077 |
| 28.1 | TIM1/TIM8/TIM20 introduction | 1077 |
| 28.2 | TIM1/TIM8/TIM20 main features | 1077 |
| 28.3 | TIM1/TIM8/TIM20 functional description | 1078 |
| 28.3.1 | Block diagram | 1078 |
| 28.3.2 | TIM1/TIM8/TIM20 pins and internal signals | 1079 |
| 28.3.3 | Time-base unit | 1084 |
| 28.3.4 | Counter modes | 1086 |
| 28.3.5 | Repetition counter | 1098 |
| 28.3.6 | External trigger input | 1099 |
| 28.3.7 | Clock selection | 1100 |
| 28.3.8 | Capture/compare channels | 1104 |
| 28.3.9 | Input capture mode | 1107 |
| 28.3.10 | PWM input mode | 1108 |
| 28.3.11 | Forced output mode | 1109 |
| 28.3.12 | Output compare mode | 1109 |
| 28.3.13 | PWM mode | 1111 |

| | | |
|---------|--|------|
| 28.3.14 | Asymmetric PWM mode | 1119 |
| 28.3.15 | Combined PWM mode | 1120 |
| 28.3.16 | Combined 3-phase PWM mode | 1121 |
| 28.3.17 | Complementary outputs and dead-time insertion | 1122 |
| 28.3.18 | Using the break function | 1125 |
| 28.3.19 | Bidirectional break inputs | 1131 |
| 28.3.20 | Clearing the tim_ocxrefsignal on an external event | 1132 |
| 28.3.21 | 6-step PWM generation | 1134 |
| 28.3.22 | One-pulse mode | 1135 |
| 28.3.23 | Retriggerable One-pulse mode | 1137 |
| 28.3.24 | Pulse on compare mode | 1138 |
| 28.3.25 | Encoder interface mode | 1140 |
| 28.3.26 | Direction bit output | 1158 |
| 28.3.27 | UIF bit remapping | 1159 |
| 28.3.28 | Timer input XOR function | 1159 |
| 28.3.29 | Interfacing with Hall sensors | 1159 |
| 28.3.30 | Timer synchronization | 1161 |
| 28.3.31 | ADC triggers | 1166 |
| 28.3.32 | DMA burst mode | 1166 |
| 28.3.33 | TIM1/TIM8/TIM20 DMA requests | 1167 |
| 28.3.34 | Debug mode | 1167 |
| 28.4 | TIM1/TIM8/TIM20 low-power modes | 1168 |
| 28.5 | TIM1/TIM8/TIM20 interrupts | 1168 |
| 28.6 | TIM1/TIM8/TIM20 registers | 1169 |
| 28.6.1 | TIMx control register 1 (TIMx_CR1)(x = 1, 8, 20) | 1169 |
| 28.6.2 | TIMx control register 2 (TIMx_CR2)(x = 1, 8, 20) | 1170 |
| 28.6.3 | TIMx slave mode control register (TIMx_SMCR)(x = 1, 8, 20) | 1174 |
| 28.6.4 | TIMx DMA/interrupt enable register (TIMx_DIER)(x = 1, 8, 20) | 1178 |
| 28.6.5 | TIMx status register (TIMx_SR)(x = 1, 8, 20) | 1179 |
| 28.6.6 | TIMx event generation register (TIMx_EGR)(x = 1, 8, 20) | 1182 |
| 28.6.7 | TIMx capture/compare mode register 1 [alternate] (TIMx_CCMR1)(x = 1, 8, 20) | 1183 |
| 28.6.8 | TIMx capture/compare mode register 1 [alternate] (TIMx_CCMR1)(x = 1, 8, 20) | 1185 |
| 28.6.9 | TIMx capture/compare mode register 2 [alternate] (TIMx_CCMR2)(x = 1, 8, 20) | 1188 |
| 28.6.10 | TIMx capture/compare mode register 2 [alternate] (TIMx_CCMR2)(x = 1, 8, 20) | 1189 |

| | | |
|-----------|---|-------------|
| 28.6.11 | TIMx capture/compare enable register (TIMx_CCER)($x = 1, 8, 20$) | 1192 |
| 28.6.12 | TIMx counter (TIMx_CNT)($x = 1, 8, 20$) | 1196 |
| 28.6.13 | TIMx prescaler (TIMx_PSC)($x = 1, 8, 20$) | 1196 |
| 28.6.14 | TIMx auto-reload register (TIMx_ARR)($x = 1, 8, 20$) | 1197 |
| 28.6.15 | TIMx repetition counter register (TIMx_RCR)($x = 1, 8, 20$) | 1197 |
| 28.6.16 | TIMx capture/compare register 1 (TIMx_CCR1)($x = 1, 8, 20$) | 1198 |
| 28.6.17 | TIMx capture/compare register 2 (TIMx_CCR2)($x = 1, 8, 20$) | 1198 |
| 28.6.18 | TIMx capture/compare register 3 (TIMx_CCR3)($x = 1, 8, 20$) | 1199 |
| 28.6.19 | TIMx capture/compare register 4 (TIMx_CCR4)($x = 1, 8, 20$) | 1200 |
| 28.6.20 | TIMx break and dead-time register (TIMx_BDTR)($x = 1, 8, 20$) | 1201 |
| 28.6.21 | TIMx capture/compare register 5 (TIMx_CCR5)($x = 1, 8, 20$) | 1205 |
| 28.6.22 | TIMx capture/compare register 6 (TIMx_CCR6)($x = 1, 8, 20$) | 1206 |
| 28.6.23 | TIMx capture/compare mode register 3 (TIMx_CCMR3) ($x = 1, 8, 20$) | 1207 |
| 28.6.24 | TIMx timer deadtime register 2 (TIMx_DTR2)($x = 1, 8, 20$) | 1208 |
| 28.6.25 | TIMx timer encoder control register (TIMx_ECR)($x = 1, 8, 20$) | 1209 |
| 28.6.26 | TIMx timer input selection register (TIMx_TISEL)($x = 1, 8, 20$) | 1210 |
| 28.6.27 | TIMx alternate function option register 1 (TIMx_AF1)($x = 1, 8, 20$) | 1211 |
| 28.6.28 | TIMx alternate function register 2 (TIMx_AF2)($x = 1, 8, 20$) | 1214 |
| 28.6.29 | TIMx DMA control register (TIMx_DCR)($x = 1, 8, 20$) | 1216 |
| 28.6.30 | TIMx DMA address for full transfer (TIMx_DMAR)($x = 1, 8, 20$) | 1217 |
| 28.6.31 | TIMx register map | 1218 |
| 29 | General-purpose timers (TIM2/TIM3/TIM4/TIM5) | 1221 |
| 29.1 | TIM2/TIM3/TIM4/TIM5 introduction | 1221 |
| 29.2 | TIM2/TIM3/TIM4/TIM5 main features | 1221 |
| 29.3 | TIM2/TIM3/TIM4/TIM5 implementation | 1222 |
| 29.4 | TIM2/TIM3/TIM4/TIM5 functional description | 1223 |
| 29.4.1 | Block diagram | 1223 |
| 29.4.2 | TIM2/TIM3/TIM4/TIM5 pins and internal signals | 1224 |
| 29.4.3 | Time-base unit | 1228 |
| 29.4.4 | Counter modes | 1230 |
| 29.4.5 | Clock selection | 1242 |
| 29.4.6 | Capture/compare channels | 1246 |
| 29.4.7 | Input capture mode | 1248 |
| 29.4.8 | PWM input mode | 1249 |
| 29.4.9 | Forced output mode | 1250 |

| | | |
|---------|---|------|
| 29.4.10 | Output compare mode | 1250 |
| 29.4.11 | PWM mode | 1252 |
| 29.4.12 | Asymmetric PWM mode | 1260 |
| 29.4.13 | Combined PWM mode | 1261 |
| 29.4.14 | Clearing the tim_ocxref signal on an external event | 1262 |
| 29.4.15 | One-pulse mode | 1264 |
| 29.4.16 | Retriggerable one-pulse mode | 1265 |
| 29.4.17 | Pulse on compare mode | 1266 |
| 29.4.18 | Encoder interface mode | 1268 |
| 29.4.19 | Direction bit output | 1268 |
| 29.4.20 | UIF bit remapping | 1287 |
| 29.4.21 | Timer input XOR function | 1287 |
| 29.4.22 | Timers and external trigger synchronization | 1287 |
| 29.4.23 | Timers synchronization | 1291 |
| 29.4.24 | ADC triggers | 1296 |
| 29.4.25 | DMA burst mode | 1297 |
| 29.4.26 | TIM2/TIM3/TIM4/TIM5 DMA requests | 1298 |
| 29.4.27 | Debug mode | 1298 |
| 29.4.28 | TIM2/TIM3/TIM4/TIM5 low-power modes | 1298 |
| 29.4.29 | TIM2/TIM3/TIM4/TIM5 interrupts | 1299 |
| 29.5 | TIM2/TIM3/TIM4/TIM5 registers | 1300 |
| 29.5.1 | TIMx control register 1 (TIMx_CR1)(x = 2 to 5) | 1300 |
| 29.5.2 | TIMx control register 2 (TIMx_CR2)(x = 2 to 5) | 1301 |
| 29.5.3 | TIMx slave mode control register (TIMx_SMC_R)(x = 2 to 5) | 1303 |
| 29.5.4 | TIMx DMA/Interrupt enable register (TIMx_DIER)(x = 2 to 5) | 1307 |
| 29.5.5 | TIMx status register (TIMx_SR)(x = 2 to 5) | 1308 |
| 29.5.6 | TIMx event generation register (TIMx_EGR)(x = 2 to 5) | 1310 |
| 29.5.7 | TIMx capture/compare mode register 1 [alternate] (TIMx_CCMR1)(x = 2 to 5) | 1311 |
| 29.5.8 | TIMx capture/compare mode register 1 [alternate] (TIMx_CCMR1)(x = 2 to 5) | 1313 |
| 29.5.9 | TIMx capture/compare mode register 2 [alternate] (TIMx_CCMR2)(x = 2 to 5) | 1315 |
| 29.5.10 | TIMx capture/compare mode register 2 [alternate] (TIMx_CCMR2)(x = 2 to 5) | 1316 |
| 29.5.11 | TIMx capture/compare enable register (TIMx_CCER)(x = 2 to 5) | 1317 |
| 29.5.12 | TIMx counter (TIMx_CNT)(x = 3, 4) | 1319 |
| 29.5.13 | TIMx counter (TIMx_CNT)(x = 2, 5) | 1319 |

| | | |
|-----------|---|-------------|
| 29.5.14 | TIMx prescaler (TIMx_PSC)($x = 2$ to 5) | 1320 |
| 29.5.15 | TIMx auto-reload register (TIMx_ARR)($x = 3, 4$) | 1320 |
| 29.5.16 | TIMx auto-reload register (TIMx_ARR)($x = 2, 5$) | 1321 |
| 29.5.17 | TIMx capture/compare register 1 (TIMx_CC_R1)($x = 3, 4$) | 1321 |
| 29.5.18 | TIMx capture/compare register 1 (TIMx_CC_R1)($x = 2, 5$) | 1322 |
| 29.5.19 | TIMx capture/compare register 2 (TIMx_CC_R2)($x = 3, 4$) | 1323 |
| 29.5.20 | TIMx capture/compare register 2 (TIMx_CC_R2)($x = 2, 5$) | 1324 |
| 29.5.21 | TIMx capture/compare register 3 (TIMx_CC_R3)($x = 3, 4$) | 1325 |
| 29.5.22 | TIMx capture/compare register 3 (TIMx_CC_R3)($x = 2, 5$) | 1326 |
| 29.5.23 | TIMx capture/compare register 4 (TIMx_CC_R4)($x = 3, 4$) | 1327 |
| 29.5.24 | TIMx capture/compare register 4 (TIMx_CC_R4)($x = 2, 5$) | 1328 |
| 29.5.25 | TIMx timer encoder control register (TIMx_ECR)($x = 2$ to 5) | 1329 |
| 29.5.26 | TIMx timer input selection register (TIMx_TISEL)($x = 2$ to 5) | 1330 |
| 29.5.27 | TIMx alternate function register 1 (TIMx_AF1)($x = 2$ to 5) | 1331 |
| 29.5.28 | TIMx alternate function register 2 (TIMx_AF2)($x = 2$ to 5) | 1332 |
| 29.5.29 | TIMx DMA control register (TIMx_DCR)($x = 2$ to 5) | 1333 |
| 29.5.30 | TIMx DMA address for full transfer (TIMx_DMAR)($x = 2$ to 5) | 1334 |
| 29.5.31 | TIMx register map | 1335 |
| 30 | General purpose timers (TIM15/TIM16/TIM17) | 1338 |
| 30.1 | TIM15/TIM16/TIM17 introduction | 1338 |
| 30.2 | TIM15 main features | 1338 |
| 30.3 | TIM16/TIM17 main features | 1339 |
| 30.4 | TIM15/TIM16/TIM17 functional description | 1340 |
| 30.4.1 | Block diagram | 1340 |
| 30.4.2 | TIM15/TIM16/TIM17 pins and internal signals | 1341 |
| 30.4.3 | Time-base unit | 1345 |
| 30.4.4 | Counter modes | 1347 |
| 30.4.5 | Repetition counter | 1351 |
| 30.4.6 | Clock selection | 1352 |
| 30.4.7 | Capture/compare channels | 1354 |
| 30.4.8 | Input capture mode | 1356 |
| 30.4.9 | PWM input mode (only for TIM15) | 1358 |
| 30.4.10 | Forced output mode | 1359 |
| 30.4.11 | Output compare mode | 1359 |
| 30.4.12 | PWM mode | 1361 |
| 30.4.13 | Combined PWM mode (TIM15 only) | 1366 |

| | | |
|---------|---|------|
| 30.4.14 | Complementary outputs and dead-time insertion | 1367 |
| 30.4.15 | Using the break function | 1370 |
| 30.4.16 | Bidirectional break input | 1373 |
| 30.4.17 | Clearing the tim_ocxref signal on an external event | 1375 |
| 30.4.18 | 6-step PWM generation | 1376 |
| 30.4.19 | One-pulse mode | 1377 |
| 30.4.20 | Retriggerable one pulse mode (TIM15 only) | 1378 |
| 30.4.21 | UIF bit remapping | 1379 |
| 30.4.22 | Timer input XOR function (TIM15 only) | 1379 |
| 30.4.23 | External trigger synchronization (TIM15 only) | 1380 |
| 30.4.24 | Slave mode – combined reset + trigger mode (TIM15 only) | 1382 |
| 30.4.25 | Slave mode – combined reset + gated mode (TIM15 only) | 1382 |
| 30.4.26 | Timer synchronization (TIM15 only) | 1383 |
| 30.4.27 | Using timer output as trigger for other timers (TIM16/TIM17 only) | 1383 |
| 30.4.28 | ADC triggers (TIM15 only) | 1383 |
| 30.4.29 | DMA burst mode | 1383 |
| 30.4.30 | TIM15/TIM16/TIM17 DMA requests | 1384 |
| 30.4.31 | Debug mode | 1384 |
| 30.5 | TIM15/TIM16/TIM17 low-power modes | 1385 |
| 30.6 | TIM15/TIM16/TIM17 interrupts | 1385 |
| 30.7 | TIM15 registers | 1386 |
| 30.7.1 | TIM15 control register 1 (TIM15_CR1) | 1386 |
| 30.7.2 | TIM15 control register 2 (TIM15_CR2) | 1387 |
| 30.7.3 | TIM15 slave mode control register (TIM15_SMCR) | 1389 |
| 30.7.4 | TIM15 DMA/interrupt enable register (TIM15_DIER) | 1390 |
| 30.7.5 | TIM15 status register (TIM15_SR) | 1391 |
| 30.7.6 | TIM15 event generation register (TIM15_EGR) | 1393 |
| 30.7.7 | TIM15 capture/compare mode register 1 [alternate] (TIM15_CCMR1) | 1394 |
| 30.7.8 | TIM15 capture/compare mode register 1 [alternate] (TIM15_CCMR1) | 1395 |
| 30.7.9 | TIM15 capture/compare enable register (TIM15_CCER) | 1398 |
| 30.7.10 | TIM15 counter (TIM15_CNT) | 1401 |
| 30.7.11 | TIM15 prescaler (TIM15_PSC) | 1401 |
| 30.7.12 | TIM15 auto-reload register (TIM15_ARR) | 1402 |
| 30.7.13 | TIM15 repetition counter register (TIM15_RCR) | 1402 |
| 30.7.14 | TIM15 capture/compare register 1 (TIM15_CCR1) | 1403 |

| | | |
|---------|--|------|
| 30.7.15 | TIM15 capture/compare register 2 (TIM15_CCR2) | 1404 |
| 30.7.16 | TIM15 break and dead-time register (TIM15_BDTR) | 1404 |
| 30.7.17 | TIM15 timer deadtime register 2 (TIM15_DTR2) | 1407 |
| 30.7.18 | TIM15 inputselection register (TIM15_TISEL) | 1408 |
| 30.7.19 | TIM15 alternate function register 1 (TIM15_AF1) | 1409 |
| 30.7.20 | TIM15 alternate function register 2 (TIM15_AF2) | 1411 |
| 30.7.21 | TIM15 DMA control register (TIM15_DCR) | 1412 |
| 30.7.22 | TIM15 DMA address for full transfer (TIM15_DMAR) | 1412 |
| 30.7.23 | TIM15 register map | 1413 |
| 30.8 | TIM16/TIM17 registers | 1416 |
| 30.8.1 | TIMx control register 1 (TIMx_CR1)(x = 16 to 17) | 1416 |
| 30.8.2 | TIMx control register 2 (TIMx_CR2)(x = 16 to 17) | 1417 |
| 30.8.3 | TIMx DMA/interruptenable register (TIMx_DIER)(x = 16 to 17) | 1418 |
| 30.8.4 | TIMx status register (TIMx_SR)(x = 16 to 17) | 1419 |
| 30.8.5 | TIMx event generation register (TIMx_EGR)(x = 16 to 17) | 1420 |
| 30.8.6 | TIMx capture/compare mode register 1 [alternate] (TIMx_CCMR1)(x = 16 to 17) | 1421 |
| 30.8.7 | TIMx capture/compare mode register 1 [alternate] (TIMx_CCMR1)(x = 16 to 17) | 1422 |
| 30.8.8 | TIMx capture/compare enable register (TIMx_CCER)(x = 16 to 17) | 1424 |
| 30.8.9 | TIMx counter (TIMx_CNT)(x = 16 to 17) | 1427 |
| 30.8.10 | TIMx prescaler (TIMx_PSC)(x = 16 to 17) | 1427 |
| 30.8.11 | TIMx auto-reload register (TIMx_ARR)(x = 16 to 17) | 1428 |
| 30.8.12 | TIMx repetition counter register (TIMx_RCR)(x = 16 to 17) | 1428 |
| 30.8.13 | TIMx capture/compare register 1 (TIMx_CC_R1)(x = 16 to 17) | 1429 |
| 30.8.14 | TIMx break and dead-time register (TIMx_BDTR)(x = 16 to 17) | 1430 |
| 30.8.15 | TIMx timer deadtime register 2 (TIMx_DTR2)(x = 16 to 17) | 1433 |
| 30.8.16 | TIMx inputselection register (TIMx_TISEL)(x = 16 to 17) | 1434 |
| 30.8.17 | TIMx alternate function register 1 (TIMx_AF1)(x = 16 to 17) | 1434 |
| 30.8.18 | TIMx alternate function register 2 (TIMx_AF2)(x = 16 to 17) | 1437 |
| 30.8.19 | TIMx option register 1 (TIMx_OR1)(x = 16 to 17) | 1437 |
| 30.8.20 | TIMx DMA control register (TIMx_DCR)(x = 16 to 17) | 1438 |
| 30.8.21 | TIM16/TIM17 DMAaddress for full transfer (TIMx_DMAR)(x = 16 to 17) | 1438 |
| 30.8.22 | TIM16/TIM17 register map | 1440 |
| 31 | Basic timers (TIM6/TIM7) | 1442 |
| 31.1 | TIM6/TIM7 introduction | 1442 |

| | | |
|-----------|--|-------------|
| 31.2 | TIM6/TIM7 main features | 1442 |
| 31.3 | TIM6/TIM7 functional description | 1443 |
| 31.3.1 | TIM6/TIM7 block diagram | 1443 |
| 31.3.2 | TIM6/TIM7 internal signals | 1443 |
| 31.3.3 | TIM6/TIM7 clocks | 1444 |
| 31.3.4 | Time-base unit | 1444 |
| 31.3.5 | Counting mode | 1446 |
| 31.3.6 | UIF bit remapping | 1453 |
| 31.3.7 | ADC triggers | 1454 |
| 31.3.8 | TIM6/TIM7 DMA requests | 1454 |
| 31.3.9 | Debug mode | 1454 |
| 31.3.10 | TIM6/TIM7 low-power modes | 1454 |
| 31.3.11 | TIM6/TIM7 interrupts | 1454 |
| 31.4 | TIM6/TIM7 registers | 1455 |
| 31.4.1 | TIMx control register 1 (TIMx_CR1)(x = 6 to 7) | 1455 |
| 31.4.2 | TIMx control register 2 (TIMx_CR2)(x = 6 to 7) | 1457 |
| 31.4.3 | TIMx DMA/Interrupt enable register (TIMx_DIER)(x = 6 to 7) | 1457 |
| 31.4.4 | TIMx status register (TIMx_SR)(x = 6 to 7) | 1458 |
| 31.4.5 | TIMx event generation register (TIMx_EGR)(x = 6 to 7) | 1458 |
| 31.4.6 | TIMx counter (TIMx_CNT)(x = 6 to 7) | 1458 |
| 31.4.7 | TIMx prescaler (TIMx_PSC)(x = 6 to 7) | 1459 |
| 31.4.8 | TIMx auto-reload register (TIMx_ARR)(x = 6 to 7) | 1459 |
| 31.4.9 | TIMx register map | 1460 |
| 32 | Low-power timer (LPTIM) | 1461 |
| 32.1 | Introduction | 1461 |
| 32.2 | LPTIM main features | 1461 |
| 32.3 | LPTIM implementation | 1462 |
| 32.4 | LPTIM functional description | 1462 |
| 32.4.1 | LPTIM block diagram | 1462 |
| 32.4.2 | LPTIM input and trigger mapping | 1462 |
| 32.4.3 | LPTIM reset and clocks | 1463 |
| 32.4.4 | Glitch filter | 1464 |
| 32.4.5 | Prescaler | 1465 |
| 32.4.6 | Trigger multiplexer | 1465 |
| 32.4.7 | Operating mode | 1466 |

| | | |
|---------|---|------|
| 32.4.8 | Timeout function | 1468 |
| 32.4.9 | Waveform generation | 1468 |
| 32.4.10 | Register update | 1469 |
| 32.4.11 | Counter mode | 1470 |
| 32.4.12 | Timer enable | 1470 |
| 32.4.13 | Timer counter reset | 1471 |
| 32.4.14 | Encoder mode | 1471 |
| 32.4.15 | Debug mode | 1473 |
| 32.5 | LPTIM low-power modes | 1473 |
| 32.6 | LPTIM interrupts | 1474 |
| 32.7 | LPTIM registers | 1474 |
| 32.7.1 | LPTIM interrupt and status register (LPTIM_ISR) | 1475 |
| 32.7.2 | LPTIM interrupt clear register (LPTIM_ICR) | 1476 |
| 32.7.3 | LPTIM interrupt enable register (LPTIM_IER) | 1476 |
| 32.7.4 | LPTIM configuration register (LPTIM_CFGR) | 1477 |
| 32.7.5 | LPTIM control register (LPTIM_CR) | 1480 |
| 32.7.6 | LPTIM compare register (LPTIM_CMP) | 1482 |
| 32.7.7 | LPTIM autoreload register (LPTIM_ARR) | 1482 |
| 32.7.8 | LPTIM counter register (LPTIM_CNT) | 1483 |
| 32.7.9 | LPTIM option register (LPTIM_OR) | 1483 |
| 32.7.10 | LPTIM register map | 1485 |
| 33 | Infrared interface (IRTIM) | 1486 |
| 34 | AES hardware accelerator (AES) | 1487 |
| 34.1 | Introduction | 1487 |
| 34.2 | AES main features | 1487 |
| 34.3 | AES implementation | 1488 |
| 34.4 | AES functional description | 1488 |
| 34.4.1 | AES block diagram | 1488 |
| 34.4.2 | AES internal signals | 1488 |
| 34.4.3 | AES cryptographic core | 1489 |
| 34.4.4 | AES procedure to perform a cipher operation | 1494 |
| 34.4.5 | AES decryption round key preparation | 1497 |
| 34.4.6 | AES ciphertext stealing and data padding | 1498 |
| 34.4.7 | AES task suspend and resume | 1498 |

| | | |
|---------|---|------|
| 34.4.8 | AES basic chaining modes (ECB, CBC) | 1499 |
| 34.4.9 | AES counter (CTR) mode | 1504 |
| 34.4.10 | AES Galois/counter mode (GCM) | 1506 |
| 34.4.11 | AES Galois message authentication code (GMAC) | 1511 |
| 34.4.12 | AES counter with CBC-MAC (CCM) | 1513 |
| 34.4.13 | AES data registers and data swapping | 1518 |
| 34.4.14 | AES key registers | 1520 |
| 34.4.15 | AES initialization vector registers | 1520 |
| 34.4.16 | AES DMA interface | 1521 |
| 34.4.17 | AES error management | 1522 |
| 34.5 | AES interrupts | 1523 |
| 34.6 | AES processing latency | 1523 |
| 34.7 | AES registers | 1524 |
| 34.7.1 | AES control register (AES_CR) | 1524 |
| 34.7.2 | AES status register (AES_SR) | 1527 |
| 34.7.3 | AES data input register (AES_DINR) | 1528 |
| 34.7.4 | AES data output register (AES_DOUTR) | 1528 |
| 34.7.5 | AES key register 0 (AES_KEYR0) | 1529 |
| 34.7.6 | AES key register 1 (AES_KEYR1) | 1530 |
| 34.7.7 | AES key register 2 (AES_KEYR2) | 1530 |
| 34.7.8 | AES key register 3 (AES_KEYR3) | 1530 |
| 34.7.9 | AES initialization vector register 0 (AES_IVR0) | 1531 |
| 34.7.10 | AES initialization vector register 1 (AES_IVR1) | 1531 |
| 34.7.11 | AES initialization vector register 2 (AES_IVR2) | 1531 |
| 34.7.12 | AES initialization vector register 3 (AES_IVR3) | 1532 |
| 34.7.13 | AES key register 4 (AES_KEYR4) | 1532 |
| 34.7.14 | AES key register 5 (AES_KEYR5) | 1532 |
| 34.7.15 | AES key register 6 (AES_KEYR6) | 1533 |
| 34.7.16 | AES key register 7 (AES_KEYR7) | 1533 |
| 34.7.17 | AES suspend registers (AES_SUSPxR) | 1533 |
| 34.7.18 | AES register map | 1534 |
| 35 | Real-time clock (RTC) | 1536 |
| 35.1 | Introduction | 1536 |
| 35.2 | RTC main features | 1536 |
| 35.3 | RTC functional description | 1537 |

| | | |
|---------|---|------|
| 35.3.1 | RTC block diagram | 1537 |
| 35.3.2 | RTC pins and internal signals | 1538 |
| 35.3.3 | GPIOs controlled by the RTC and TAMP | 1539 |
| 35.3.4 | Clock and prescalers | 1541 |
| 35.3.5 | Real-time clock and calendar | 1542 |
| 35.3.6 | Programmable alarms | 1543 |
| 35.3.7 | Periodic auto-wake up | 1543 |
| 35.3.8 | RTC initialization and configuration | 1544 |
| 35.3.9 | Reading the calendar | 1546 |
| 35.3.10 | Resetting the RTC | 1547 |
| 35.3.11 | RTC synchronization | 1547 |
| 35.3.12 | RTC reference clock detection | 1548 |
| 35.3.13 | RTC smooth digital calibration | 1548 |
| 35.3.14 | Timestamp function | 1550 |
| 35.3.15 | Calibration clock output | 1551 |
| 35.3.16 | Tamper and alarm output | 1551 |
| 35.4 | RTC low-power modes | 1552 |
| 35.5 | RTC interrupts | 1553 |
| 35.6 | RTC registers | 1553 |
| 35.6.1 | RTC time register (RTC_TR) | 1553 |
| 35.6.2 | RTC date register (RTC_DR) | 1554 |
| 35.6.3 | RTC sub second register (RTC_SSR) | 1555 |
| 35.6.4 | RTC initialization control and status register (RTC_ICSR) | 1555 |
| 35.6.5 | RTC prescaler register (RTC_PRER) | 1557 |
| 35.6.6 | RTC wakeup timer register (RTC_WUTR) | 1558 |
| 35.6.7 | RTC control register (RTC_CR) | 1558 |
| 35.6.8 | RTC write protection register (RTC_WPR) | 1561 |
| 35.6.9 | RTC calibration register (RTC_CALR) | 1562 |
| 35.6.10 | RTC shift control register (RTC_SHIFTR) | 1563 |
| 35.6.11 | RTC timestamp time register (RTC_TSTR) | 1564 |
| 35.6.12 | RTC timestamp date register (RTC_TSDDR) | 1564 |
| 35.6.13 | RTC timestamp sub second register (RTC_TSSSR) | 1565 |
| 35.6.14 | RTC alarm A register (RTC_ALRMAR) | 1566 |
| 35.6.15 | RTC alarm A subsecond register (RTC_ALRMASSR) | 1567 |
| 35.6.16 | RTC alarm B register (RTC_ALRMBR) | 1568 |
| 35.6.17 | RTC alarm B subsecond register (RTC_ALRMBSSR) | 1569 |
| 35.6.18 | RTC status register (RTC_SR) | 1569 |

| | | |
|-----------|---|-------------|
| 35.6.19 | RTC masked interrupt status register (RTC_MISR) | 1570 |
| 35.6.20 | RTC status clear register (RTC_SCR) | 1571 |
| 35.6.21 | RTC register map | 1573 |
| 36 | Tamper and backup registers (TAMP) | 1575 |
| 36.1 | Introduction | 1575 |
| 36.2 | TAMP main features | 1575 |
| 36.3 | TAMP functional description | 1576 |
| 36.3.1 | TAMP block diagram | 1576 |
| 36.3.2 | TAMP pins and internal signals | 1577 |
| 36.3.3 | TAMP register write protection | 1577 |
| 36.3.4 | Tamper detection | 1578 |
| 36.4 | TAMP low-power modes | 1580 |
| 36.5 | TAMP interrupts | 1580 |
| 36.6 | TAMP registers | 1580 |
| 36.6.1 | TAMP control register 1 (TAMP_C_R1) | 1581 |
| 36.6.2 | TAMP control register 2 (TAMP_C_R2) | 1582 |
| 36.6.3 | TAMP filter control register (TAMP_FLTCR) | 1583 |
| 36.6.4 | TAMP interrupt enable register (TAMP_IER) | 1584 |
| 36.6.5 | TAMP status register (TAMP_SR) | 1585 |
| 36.6.6 | TAMP masked interrupt status register (TAMP_MISR) | 1586 |
| 36.6.7 | TAMP status clear register (TAMP_SCR) | 1587 |
| 36.6.8 | TAMP backup x register (TAMP_BKPxR) | 1588 |
| 36.6.9 | TAMP register map | 1589 |
| 37 | Universal synchronous/asynchronous receiver transmitter (USART/UART) | 1590 |
| 37.1 | USART introduction | 1590 |
| 37.2 | USART main features | 1591 |
| 37.3 | USART extended features | 1592 |
| 37.4 | USART implementation | 1592 |
| 37.5 | USART functional description | 1593 |
| 37.5.1 | USART block diagram | 1593 |
| 37.5.2 | USART signals | 1594 |
| 37.5.3 | USART character description | 1595 |
| 37.5.4 | USART FIFOs and thresholds | 1597 |

| | | |
|---------|---|------|
| 37.5.5 | USART transmitter | 1597 |
| 37.5.6 | USART receiver | 1601 |
| 37.5.7 | USART baud rate generation | 1608 |
| 37.5.8 | Tolerance of the USART receiver to clock deviation | 1609 |
| 37.5.9 | USART Auto baud rate detection | 1611 |
| 37.5.10 | USART multiprocessor communication | 1613 |
| 37.5.11 | USART Modbus communication | 1615 |
| 37.5.12 | USART parity control | 1616 |
| 37.5.13 | USART LIN (local interconnection network) mode | 1617 |
| 37.5.14 | USART synchronous mode | 1619 |
| 37.5.15 | USART single-wire Half-duplex communication | 1623 |
| 37.5.16 | USART receiver timeout | 1623 |
| 37.5.17 | USART Smartcard mode | 1624 |
| 37.5.18 | USART IrDA SIR ENDEC block | 1628 |
| 37.5.19 | Continuous communication using USART and DMA | 1631 |
| 37.5.20 | RS232 Hardware flow control and RS485 Driver Enable | 1633 |
| 37.5.21 | USART low-power management | 1636 |
| 37.6 | USART in low-power modes | 1639 |
| 37.7 | USART interrupts | 1640 |
| 37.8 | USART registers | 1641 |
| 37.8.1 | USART control register 1 [alternate] (USART_CR1) | 1641 |
| 37.8.2 | USART control register 1 [alternate] (USART_CR1) | 1645 |
| 37.8.3 | USART control register 2 (USART_CR2) | 1648 |
| 37.8.4 | USART control register 3 (USART_CR3) | 1652 |
| 37.8.5 | USART baud rate register (USART_BRR) | 1657 |
| 37.8.6 | USART guard time and prescaler register (USART_GTPR) | 1657 |
| 37.8.7 | USART receiver timeout register (USART_RTOR) | 1658 |
| 37.8.8 | USART request register (USART_RQR) | 1659 |
| 37.8.9 | USART interrupt and status register [alternate] (USART_ISR) | 1660 |
| 37.8.10 | USART interrupt and status register [alternate] (USART_ISR) | 1666 |
| 37.8.11 | USART interrupt flag clear register (USART_ICR) | 1671 |
| 37.8.12 | USART receive data register (USART_RDR) | 1673 |
| 37.8.13 | USART transmit data register (USART_TDR) | 1673 |
| 37.8.14 | USART prescaler register (USART_PRESC) | 1674 |
| 37.8.15 | USART register map | 1675 |

| | | |
|-----------|---|-------------|
| 38 | Low-power universal asynchronous receiver transmitter (LPUART) | 1677 |
| 38.1 | LPUART introduction | 1677 |
| 38.2 | LPUART main features | 1678 |
| 38.3 | LPUART implementation | 1679 |
| 38.4 | LPUART functional description | 1680 |
| 38.4.1 | LPUART block diagram | 1680 |
| 38.4.2 | LPUART signals | 1681 |
| 38.4.3 | LPUART character description | 1681 |
| 38.4.4 | LPUART FIFOs and thresholds | 1682 |
| 38.4.5 | LPUART transmitter | 1683 |
| 38.4.6 | LPUART receiver | 1686 |
| 38.4.7 | LPUART baud rate generation | 1690 |
| 38.4.8 | Tolerance of the LPUART receiver to clock deviation | 1691 |
| 38.4.9 | LPUART multiprocessor communication | 1692 |
| 38.4.10 | LPUART parity control | 1694 |
| 38.4.11 | LPUART single-wire Half-duplex communication | 1695 |
| 38.4.12 | Continuous communication using DMA and LPUART | 1695 |
| 38.4.13 | RS232 Hardware flow control and RS485 Driver Enable | 1698 |
| 38.4.14 | LPUART low-power management | 1700 |
| 38.5 | LPUART in low-power modes | 1703 |
| 38.6 | LPUART interrupts | 1704 |
| 38.7 | LPUART registers | 1705 |
| 38.7.1 | LPUART control register 1 [alternate] (LPUART_CR1) | 1705 |
| 38.7.2 | LPUART control register 1 [alternate] (LPUART_CR1) | 1708 |
| 38.7.3 | LPUART control register 2 (LPUART_CR2) | 1711 |
| 38.7.4 | LPUART control register 3 (LPUART_CR3) | 1713 |
| 38.7.5 | LPUART baud rate register (LPUART_BRR) | 1716 |
| 38.7.6 | LPUART request register (LPUART_RQR) | 1717 |
| 38.7.7 | LPUART interrupt and status register [alternate] (LPUART_ISR) | 1717 |
| 38.7.8 | LPUART interrupt and status register [alternate] (LPUART_ISR) | 1722 |
| 38.7.9 | LPUART interrupt flag clear register (LPUART_ICR) | 1725 |
| 38.7.10 | LPUART receive data register (LPUART_RDR) | 1726 |
| 38.7.11 | LPUART transmit data register (LPUART_TDR) | 1726 |
| 38.7.12 | LPUART prescaler register (LPUART_PRESC) | 1727 |
| 38.7.13 | LPUART register map | 1728 |

| | | |
|-----------|--|-------------|
| 39 | Serial peripheral interface / integrated interchip sound (SPI/I²S) | 1730 |
| 39.1 | Introduction | 1730 |
| 39.2 | SPI main features | 1730 |
| 39.3 | I ² S main features | 1731 |
| 39.4 | SPI/I ² S implementation | 1731 |
| 39.5 | SPI functional description | 1732 |
| 39.5.1 | General description | 1732 |
| 39.5.2 | Communications between one master and one slave | 1733 |
| 39.5.3 | Standard multi-slave communication | 1735 |
| 39.5.4 | Multi-master communication | 1736 |
| 39.5.5 | Slave select (NSS) pin management | 1737 |
| 39.5.6 | Communication formats | 1738 |
| 39.5.7 | Configuration of SPI | 1740 |
| 39.5.8 | Procedure for enabling SPI | 1741 |
| 39.5.9 | Data transmission and reception procedures | 1741 |
| 39.5.10 | SPI status flags | 1751 |
| 39.5.11 | SPI error flags | 1752 |
| 39.5.12 | NSS pulse mode | 1753 |
| 39.5.13 | TI mode | 1753 |
| 39.5.14 | CRC calculation | 1754 |
| 39.6 | SPI interrupts | 1756 |
| 39.7 | I ² S functional description | 1757 |
| 39.7.1 | I ² S general description | 1757 |
| 39.7.2 | Supported audio protocols | 1758 |
| 39.7.3 | Start-up description | 1765 |
| 39.7.4 | Clock generator | 1767 |
| 39.7.5 | I ² S master mode | 1770 |
| 39.7.6 | I ² S slave mode | 1771 |
| 39.7.7 | I ² S status flags | 1773 |
| 39.7.8 | I ² S error flags | 1774 |
| 39.7.9 | DMA features | 1775 |
| 39.8 | I ² S interrupts | 1775 |
| 39.9 | SPI and I ² S registers | 1776 |
| 39.9.1 | SPI control register 1 (SPIx_CR1) | 1776 |
| 39.9.2 | SPI control register 2 (SPIx_CR2) | 1778 |
| 39.9.3 | SPI status register (SPIx_SR) | 1780 |

| | | |
|-----------|--|-------------|
| 39.9.4 | SPI data register (SPIx_DR) | 1782 |
| 39.9.5 | SPI CRC polynomial register (SPIx_CRCPR) | 1782 |
| 39.9.6 | SPI Rx CRC register (SPIx_RXCRCR) | 1782 |
| 39.9.7 | SPI Tx CRC register (SPIx_TXCRCR) | 1783 |
| 39.9.8 | SPIx_I2S configuration register (SPIx_I2SCFGR) | 1783 |
| 39.9.9 | SPIx_I2S prescaler register (SPIx_I2SPR) | 1785 |
| 39.9.10 | SPI/I2S register map | 1787 |
| 40 | Serial audio interface (SAI) | 1788 |
| 40.1 | Introduction | 1788 |
| 40.2 | SAI main features | 1788 |
| 40.3 | SAI implementation | 1789 |
| 40.4 | SAI functional description | 1789 |
| 40.4.1 | SAI block diagram | 1789 |
| 40.4.2 | SAI pins and internal signals | 1791 |
| 40.4.3 | Main SAI modes | 1791 |
| 40.4.4 | SAI synchronization mode | 1792 |
| 40.4.5 | Audio data size | 1793 |
| 40.4.6 | Frame synchronization | 1793 |
| 40.4.7 | Slot configuration | 1796 |
| 40.4.8 | SAI clock generator | 1798 |
| 40.4.9 | Internal FIFOs | 1801 |
| 40.4.10 | PDM interface | 1803 |
| 40.4.11 | AC'97 link controller | 1811 |
| 40.4.12 | SPDIF output | 1812 |
| 40.4.13 | Specific features | 1815 |
| 40.4.14 | Error flags | 1819 |
| 40.4.15 | Disabling the SAI | 1822 |
| 40.4.16 | SAI DMA interface | 1822 |
| 40.5 | SAI interrupts | 1823 |
| 40.6 | SAI registers | 1825 |
| 40.6.1 | SAI configuration register 1 (SAI_ACR1) | 1825 |
| 40.6.2 | SAI configuration register 1 (SAI_BCR1) | 1827 |
| 40.6.3 | SAI configuration register 2 (SAI_AC R2) | 1830 |
| 40.6.4 | SAI configuration register 2 (SAI_BC R2) | 1832 |
| 40.6.5 | SAI frame configuration register (SAI_AFRCR) | 1834 |

| | | |
|-----------|---|-------------|
| 40.6.6 | SAI frame configuration register (SAI_BFRCR) | 1836 |
| 40.6.7 | SAI slot register (SAI_ASLOTR) | 1837 |
| 40.6.8 | SAI slot register (SAI_BSLOTR) | 1838 |
| 40.6.9 | SAI interrupt mask register (SAI_AIM) | 1839 |
| 40.6.10 | SAI interrupt mask register (SAI_BIM) | 1841 |
| 40.6.11 | SAI status register (SAI_ASR) | 1842 |
| 40.6.12 | SAI status register (SAI_BSR) | 1844 |
| 40.6.13 | SAI clear flag register (SAI_ACLRFR) | 1846 |
| 40.6.14 | SAI clear flag register (SAI_BCLRFR) | 1847 |
| 40.6.15 | SAI data register (SAI_ADR) | 1848 |
| 40.6.16 | SAI data register (SAI_BDR) | 1849 |
| 40.6.17 | SAI PDM control register (SAI_PDMCR) | 1849 |
| 40.6.18 | SAI PDM delay register (SAI_PDMDLY) | 1851 |
| 40.6.19 | SAI register map | 1853 |
| 41 | Inter-integrated circuit (I2C) interface | 1855 |
| 41.1 | Introduction | 1855 |
| 41.2 | I2C main features | 1855 |
| 41.3 | I2C implementation | 1856 |
| 41.4 | I2C functional description | 1856 |
| 41.4.1 | I2C block diagram | 1857 |
| 41.4.2 | I2C pins and internal signals | 1858 |
| 41.4.3 | I2C clock requirements | 1858 |
| 41.4.4 | Mode selection | 1858 |
| 41.4.5 | I2C initialization | 1859 |
| 41.4.6 | Software reset | 1864 |
| 41.4.7 | Data transfer | 1865 |
| 41.4.8 | I2C slave mode | 1867 |
| 41.4.9 | I2C master mode | 1876 |
| 41.4.10 | I2C_TIMINGR register configuration examples | 1888 |
| 41.4.11 | SMBus specific features | 1889 |
| 41.4.12 | SMBus initialization | 1892 |
| 41.4.13 | SMBus: I2C_TIMEOUTR register configuration examples | 1894 |
| 41.4.14 | SMBus slave mode | 1895 |
| 41.4.15 | Wakeup from Stop mode on address match | 1902 |
| 41.4.16 | Error conditions | 1902 |
| 41.4.17 | DMA requests | 1904 |

| | | |
|-----------|---|-------------|
| 41.4.18 | Debug mode | 1905 |
| 41.5 | I2C low-power modes | 1905 |
| 41.6 | I2C interrupts | 1906 |
| 41.7 | I2C registers | 1907 |
| 41.7.1 | I2C control register 1 (I2C_CR1) | 1907 |
| 41.7.2 | I2C control register 2 (I2C_CR2) | 1910 |
| 41.7.3 | I2C own address 1 register (I2C_OAR1) | 1912 |
| 41.7.4 | I2C own address 2 register (I2C_OAR2) | 1913 |
| 41.7.5 | I2C timing register (I2C_TIMINGR) | 1914 |
| 41.7.6 | I2C timeout register (I2C_TIMEOUTR) | 1915 |
| 41.7.7 | I2C interrupt and status register (I2C_ISR) | 1916 |
| 41.7.8 | I2C interrupt clear register (I2C_ICR) | 1918 |
| 41.7.9 | I2C PEC register (I2C_PECR) | 1919 |
| 41.7.10 | I2C receive data register (I2C_RXDR) | 1920 |
| 41.7.11 | I2C transmit data register (I2C_TXDR) | 1920 |
| 41.7.12 | I2C register map | 1921 |
| 42 | Independent watchdog (IWDG) | 1923 |
| 42.1 | Introduction | 1923 |
| 42.2 | IWDG main features | 1923 |
| 42.3 | IWDG functional description | 1923 |
| 42.3.1 | IWDG block diagram | 1923 |
| 42.3.2 | Window option | 1924 |
| 42.3.3 | Hardware watchdog | 1925 |
| 42.3.4 | Low-power freeze | 1925 |
| 42.3.5 | Register access protection | 1925 |
| 42.3.6 | Debug mode | 1925 |
| 42.4 | IWDG registers | 1926 |
| 42.4.1 | IWDG key register (IWDG_KR) | 1926 |
| 42.4.2 | IWDG prescaler register (IWDG_PR) | 1927 |
| 42.4.3 | IWDG reload register (IWDG_RLR) | 1928 |
| 42.4.4 | IWDG status register (IWDG_SR) | 1929 |
| 42.4.5 | IWDG window register (IWDG_WINR) | 1930 |
| 42.4.6 | IWDG register map | 1931 |
| 43 | System window watchdog (WWDG) | 1932 |

| | | |
|-----------|--|-------------|
| 43.1 | Introduction | 1932 |
| 43.2 | WWDG main features | 1932 |
| 43.3 | WWDG functional description | 1932 |
| 43.3.1 | WWDG block diagram | 1933 |
| 43.3.2 | Enabling the watchdog | 1933 |
| 43.3.3 | Controlling the down-counter | 1933 |
| 43.3.4 | How to program the watchdog timeout | 1933 |
| 43.3.5 | Debug mode | 1935 |
| 43.4 | WWDG interrupts | 1935 |
| 43.5 | WWDG registers | 1935 |
| 43.5.1 | WWDG control register (WWDG_CR) | 1935 |
| 43.5.2 | WWDG configuration register (WWDG_CFR) | 1936 |
| 43.5.3 | WWDG status register (WWDG_SR) | 1937 |
| 43.5.4 | WWDG register map | 1937 |
| 44 | FD controller area network (FDCAN) | 1938 |
| 44.1 | Introduction | 1938 |
| 44.2 | FDCAN main features | 1940 |
| 44.3 | FDCAN functional description | 1941 |
| 44.3.1 | Bit timing | 1942 |
| 44.3.2 | Operating modes | 1943 |
| 44.3.3 | Message RAM | 1952 |
| 44.3.4 | FIFO acknowledge handling | 1960 |
| 44.3.5 | FDCAN Rx FIFO element | 1961 |
| 44.3.6 | FDCAN Tx buffer element | 1963 |
| 44.3.7 | FDCAN Tx event FIFO element | 1965 |
| 44.3.8 | FDCAN Standard message ID Filter element | 1966 |
| 44.3.9 | FDCAN Extended message ID filter element | 1967 |
| 44.4 | FDCAN registers | 1968 |
| 44.4.1 | FDCAN core release register (FDCAN_CREL) | 1968 |
| 44.4.2 | FDCAN endian register (FDCAN_ENDIAN) | 1968 |
| 44.4.3 | FDCAN data bit timing and prescaler register (FDCAN_DBTP) | 1969 |
| 44.4.4 | FDCAN test register (FDCAN_TEST) | 1970 |
| 44.4.5 | FDCAN RAM watchdog register (FDCAN_RWD) | 1970 |
| 44.4.6 | FDCAN CC control register (FDCAN_CCCR) | 1971 |
| 44.4.7 | FDCAN nominal bit timing and prescaler register (FDCAN_NBTP) | 1973 |

| | | |
|-----------|--|-------------|
| 44.4.8 | FDCAN timestamp counter configuration register (FDCAN_TSCC) | 1974 |
| 44.4.9 | FDCAN timestamp counter value register (FDCAN_TSCV) | 1975 |
| 44.4.10 | FDCAN timeout counter configuration register (FDCAN_TOCC) | 1976 |
| 44.4.11 | FDCAN timeout counter value register (FDCAN_TOCV) | 1976 |
| 44.4.12 | FDCAN error counter register (FDCAN_ECR) | 1977 |
| 44.4.13 | FDCAN protocol status register (FDCAN_PSR) | 1977 |
| 44.4.14 | FDCAN transmitter delay compensation register (FDCAN_TDCR) | 1980 |
| 44.4.15 | FDCAN interrupt register (FDCAN_IR) | 1980 |
| 44.4.16 | FDCAN interrupt enable register (FDCAN_IE) | 1983 |
| 44.4.17 | FDCAN interrupt line select register (FDCAN_ILS) | 1985 |
| 44.4.18 | FDCAN interrupt line enable register (FDCAN_IIE) | 1986 |
| 44.4.19 | FDCAN global filter configuration register (FDCAN_RXGFC) | 1986 |
| 44.4.20 | FDCAN extended ID and mask register (FDCAN_XIDAM) | 1988 |
| 44.4.21 | FDCAN high-priority message status register (FDCAN_HPMS) | 1988 |
| 44.4.22 | FDCAN Rx FIFO 0 status register (FDCAN_RXF0S) | 1989 |
| 44.4.23 | CAN Rx FIFO 0 acknowledge register (FDCAN_RXF0A) | 1990 |
| 44.4.24 | FDCAN Rx FIFO 1 status register (FDCAN_RXF1S) | 1990 |
| 44.4.25 | FDCAN Rx FIFO 1 acknowledge register (FDCAN_RXF1A) | 1991 |
| 44.4.26 | FDCAN Tx buffer configuration register (FDCAN_TXBC) | 1991 |
| 44.4.27 | FDCAN Tx FIFO/queue status register (FDCAN_TXFQS) | 1992 |
| 44.4.28 | FDCAN Tx buffer request pending register (FDCAN_TXBRP) | 1993 |
| 44.4.29 | FDCAN Tx buffer add request register (FDCAN_TXBAR) | 1994 |
| 44.4.30 | FDCAN Tx buffer cancellation request register (FDCAN_TXBCR) | 1994 |
| 44.4.31 | FDCAN Tx buffer transmission occurred register (FDCAN_TXBTO) | 1995 |
| 44.4.32 | FDCAN Tx buffer cancellation finished register (FDCAN_TXBCF) | 1995 |
| 44.4.33 | FDCAN Tx buffer transmission interrupt enable register (FDCAN_TXBTIE) | 1996 |
| 44.4.34 | FDCAN Tx buffer cancellation finished interrupt enable register (FDCAN_TXBCIE) | 1996 |
| 44.4.35 | FDCAN Tx event FIFO status register (FDCAN_TXEFS) | 1997 |
| 44.4.36 | FDCAN Tx event FIFO acknowledge register (FDCAN_TXEFA) | 1997 |
| 44.4.37 | FDCAN CFG clock divider register (FDCAN_CKDIV) | 1998 |
| 44.4.38 | FDCAN register map | 1999 |
| 45 | Universal serial bus full-speed device interface (USB) | 2003 |
| 45.1 | Introduction | 2003 |
| 45.2 | USB main features | 2003 |

| | | |
|-----------|--|-------------|
| 45.3 | USB implementation | 2003 |
| 45.4 | USB functional description | 2004 |
| 45.4.1 | Description of USB blocks | 2005 |
| 45.5 | Programming considerations | 2006 |
| 45.5.1 | Generic USB device programming | 2006 |
| 45.5.2 | System and power-on reset | 2007 |
| 45.5.3 | Double-buffered endpoints | 2012 |
| 45.5.4 | Isochronous transfers | 2014 |
| 45.5.5 | Suspend/Resume events | 2015 |
| 45.6 | USB and USB SRAM registers | 2018 |
| 45.6.1 | Common registers | 2018 |
| 45.6.2 | Buffer descriptor table | 2031 |
| 45.6.3 | USB register map | 2034 |
| 46 | USB Type-C®/USB Power Delivery interface (UCPD) | 2036 |
| 46.1 | Introduction | 2036 |
| 46.2 | UCPD main features | 2036 |
| 46.3 | UCPD implementation | 2036 |
| 46.4 | UCPD functional description | 2037 |
| 46.4.1 | UCPD block diagram | 2038 |
| 46.4.2 | UCPD reset and clocks | 2039 |
| 46.4.3 | Physical layer protocol | 2040 |
| 46.4.4 | UCPD BMC transmitter | 2046 |
| 46.4.5 | UCPD BMC receiver | 2048 |
| 46.4.6 | UCPD Type-C pull-ups (Rp) and pull-downs (Rd) | 2049 |
| 46.4.7 | UCPD Type-C voltage monitoring and de-bouncing | 2050 |
| 46.4.8 | UCPD fast role swap (FRS) signalling and detection | 2050 |
| 46.4.9 | UCPD DMA Interface | 2050 |
| 46.4.10 | Wake up from Stop mode | 2050 |
| 46.4.11 | UCPD programming sequences | 2051 |
| 46.5 | UCPD low-power modes | 2055 |
| 46.6 | UCPD interrupts | 2056 |
| 46.7 | UCPD registers | 2057 |
| 46.7.1 | UCPD configuration register 1 (UCPD_CFGR1) | 2057 |
| 46.7.2 | UCPD configuration register 2 (UCPD_CFGR2) | 2059 |
| 46.7.3 | UCPD control register (UCPD_CR) | 2060 |

| | | |
|-----------|--|-------------|
| 46.7.4 | UCPD interrupt mask register (UCPD_IMR) | 2062 |
| 46.7.5 | UCPD status register (UCPD_SR) | 2064 |
| 46.7.6 | UCPD interrupt clear register (UCPD_ICR) | 2067 |
| 46.7.7 | UCPD Tx ordered set type register (UCPD_TX_ORDSETR) | 2068 |
| 46.7.8 | UCPD Tx payload size register (UCPD_TX_PAYSZR) | 2068 |
| 46.7.9 | UCPD Tx data register (UCPD_TXDR) | 2069 |
| 46.7.10 | UCPD Rx ordered set register (UCPD_RX_ORDSETR) | 2069 |
| 46.7.11 | UCPD Rx payload size register (UCPD_RX_PAYSZR) | 2070 |
| 46.7.12 | UCPD receive data register (UCPD_RXDR) | 2071 |
| 46.7.13 | UCPD Rx ordered set extension register 1 (UCPD_RX_ORDEXTR1) | 2071 |
| 46.7.14 | UCPD Rx ordered set extension register 2 (UCPD_RX_ORDEXTR2) | 2072 |
| 46.7.15 | UCPD register map | 2072 |
| 47 | Debug support (DBG) | 2075 |
| 47.1 | Overview | 2075 |
| 47.2 | Reference Arm® documentation | 2076 |
| 47.3 | SWJ debug port (serial wire and JTAG) | 2076 |
| 47.3.1 | Mechanism to select the JTAG-DP or the SW-DP | 2077 |
| 47.4 | Pinout and debug port pins | 2077 |
| 47.4.1 | SWJ debug port pins | 2078 |
| 47.4.2 | Flexible SWJ-DP pin assignment | 2078 |
| 47.4.3 | Internal pull-up and pull-down on JTAG pins | 2079 |
| 47.4.4 | Using serial wire and releasing the unused debug pins as GPIOs | 2080 |
| 47.5 | STM32G4 Series JTAG TAP connection | 2080 |
| 47.6 | ID codes and locking mechanism | 2081 |
| 47.6.1 | MCU device ID code | 2082 |
| 47.6.2 | Boundary scan TAP | 2082 |
| 47.6.3 | Cortex®-M4 with FPU TAP | 2082 |
| 47.6.4 | Cortex®-M4 with FPU JEDEC-106 ID code | 2083 |
| 47.7 | JTAG debug port | 2083 |
| 47.8 | SW debug port | 2085 |
| 47.8.1 | SW protocol introduction | 2085 |
| 47.8.2 | SW protocol sequence | 2085 |
| 47.8.3 | SW-DP state machine (reset, idle states, ID code) | 2086 |
| 47.8.4 | DP and AP read/write accesses | 2086 |

| | | |
|----------|--|------|
| 47.8.5 | SW-DP registers | 2087 |
| 47.8.6 | SW-AP registers | 2088 |
| 47.9 | AHB-AP (AHB access port) - valid for both JTAG-DP and SW-DP | 2088 |
| 47.10 | Core debug | 2089 |
| 47.11 | Capability of the debugger host to connect under system reset | 2089 |
| 47.12 | FPB (Flash patch breakpoint) | 2090 |
| 47.13 | DWT (data watchpoint trigger) | 2090 |
| 47.14 | ITM (instrumentation trace macrocell) | 2091 |
| 47.14.1 | General description | 2091 |
| 47.14.2 | Time stamp packets, synchronization and overflow packets | 2091 |
| 47.15 | ETM (Embedded Trace Macrocell™) | 2093 |
| 47.15.1 | General description | 2093 |
| 47.15.2 | Signal protocol, packet types | 2093 |
| 47.15.3 | Main ETM registers | 2093 |
| 47.15.4 | Configuration example | 2094 |
| 47.16 | MCU debug component (DBGMCU) | 2094 |
| 47.16.1 | Debug support for low-power modes | 2094 |
| 47.16.2 | Debug support for timers, RTC, watchdog and I ² C | 2095 |
| 47.16.3 | Debug MCU configuration register (DBGMCU_CR) | 2095 |
| 47.16.4 | Debug MCU APB1 freeze register1 (DBGMCU_APB1FZR1) | 2096 |
| 47.16.5 | Debug MCU APB1 freeze register2 (DBGMCU_APB1FZR2) | 2098 |
| 47.16.6 | Debug MCU APB2 freeze register (DBGMCU_APB2FZR) | 2098 |
| 47.17 | TPIU (trace port interface unit) | 2100 |
| 47.17.1 | Introduction | 2100 |
| 47.17.2 | TRACE pin assignment | 2100 |
| 47.17.3 | TPUI formatter | 2102 |
| 47.17.4 | TPUI frame synchronization packets | 2103 |
| 47.17.5 | Transmission of the synchronization frame packet | 2103 |
| 47.17.6 | Synchronous mode | 2103 |
| 47.17.7 | Asynchronous mode | 2104 |
| 47.17.8 | TRACECLKIN connection inside the STM32G4 Series | 2104 |
| 47.17.9 | TPIU registers | 2105 |
| 47.17.10 | Example of configuration | 2106 |
| 47.18 | DBG register map | 2107 |

| | | |
|-----------|-------------------------------------|-------------|
| 48 | Device electronic signature | 2108 |
| 48.1 | Unique device ID register (96 bits) | 2108 |
| 48.2 | Flash size data register | 2109 |
| 48.3 | Package data register | 2110 |
| 49 | Revision history | 2111 |

List of tables

| | | |
|-----------|--|-----|
| Table 1. | STM32G4 Series memory density | 74 |
| Table 2. | Product specific features | 75 |
| Table 3. | STM32G4 Series memory map and peripheral register boundary addresses | 83 |
| Table 4. | CCM SRAM organization | 88 |
| Table 5. | Boot modes | 90 |
| Table 6. | Memory mapping versus boot mode/physical remap | 92 |
| Table 7. | Flash module - 512/256/128 KB dual bank organization (64 bits read width) | 95 |
| Table 8. | Flash module - 512/256/128 KB single bank organization (128 bits read width) | 96 |
| Table 9. | Number of wait states according to CPU clock (HCLK) frequency | 98 |
| Table 10. | Option byte format | 109 |
| Table 11. | Option byte organization | 109 |
| Table 12. | Flash memory read protection status | 120 |
| Table 13. | Access status versus protection level and execution modes | 122 |
| Table 14. | PCROP protection | 124 |
| Table 15. | WRP protection | 126 |
| Table 16. | Flash interrupt request | 128 |
| Table 17. | Flash interface - register map and reset values | 146 |
| Table 18. | Flash module - 256/512 Kbytes organization (64 bits read width) | 149 |
| Table 19. | Number of wait states according to CPU clock (HCLK) frequency | 150 |
| Table 20. | Option byte format | 160 |
| Table 21. | Option byte organization | 160 |
| Table 22. | Flash memory read protection status | 166 |
| Table 23. | Access status versus protection level and execution modes | 168 |
| Table 24. | PCROP protection | 170 |
| Table 25. | WRP protection | 171 |
| Table 26. | Flash interrupt request | 173 |
| Table 27. | Flash interface - register map and reset values | 187 |
| Table 28. | Flash module - 64/128 Kbytes organization (64 bits read width) | 190 |
| Table 29. | Number of wait states according to CPU clock (HCLK) frequency | 191 |
| Table 30. | Option byte format | 201 |
| Table 31. | Option byte organization | 201 |
| Table 32. | Flash memory read protection status | 207 |
| Table 33. | Access status versus protection level and execution modes | 209 |
| Table 34. | PCROP protection | 211 |
| Table 35. | WRP protection | 212 |
| Table 36. | Flash interrupt request | 214 |
| Table 37. | Flash interface - register map and reset values | 228 |
| Table 38. | Range 1 boost mode configuration | 234 |
| Table 39. | PVM features | 238 |
| Table 40. | Low-power mode summary | 241 |
| Table 41. | Functionalities depending on the working mode | 242 |
| Table 42. | Low-power run | 245 |
| Table 43. | Sleep | 247 |
| Table 44. | Low-power sleep | 248 |
| Table 45. | Stop 0 mode | 250 |
| Table 46. | Stop 1 mode | 251 |
| Table 47. | Standby mode | 253 |

| | | |
|-----------|--|-----|
| Table 48. | Shutdown mode | 255 |
| Table 49. | PWR register map and reset values | 271 |
| Table 50. | Clock source frequency | 283 |
| Table 51. | RCC register map and reset values | 335 |
| Table 52. | CRS features | 340 |
| Table 53. | Effect of low-power modes on CRS | 344 |
| Table 54. | Interrupt control bits | 344 |
| Table 55. | CRS register map and reset values | 349 |
| Table 56. | Port bit configuration table | 353 |
| Table 57. | GPIO register map and reset values | 368 |
| Table 58. | BOOSTEN and ANASWVDD set/reset | 373 |
| Table 59. | SYSCFG register map and reset values | 381 |
| Table 60. | STM32G4 Series peripherals interconnect matrix | 383 |
| Table 61. | Interconnect 1 | 385 |
| Table 62. | Interconnect 12 | 386 |
| Table 63. | Interconnect 13 | 386 |
| Table 64. | Interconnect 14 | 386 |
| Table 65. | Interconnect 19 | 387 |
| Table 66. | Interconnect 20 | 389 |
| Table 67. | Interconnect 17 | 391 |
| Table 68. | Interconnect 15 | 392 |
| Table 69. | Interconnect 21 | 392 |
| Table 70. | Interconnect 22 | 393 |
| Table 71. | Interconnect 23 | 394 |
| Table 72. | Interconnect 24 | 395 |
| Table 73. | Interconnect 2 | 396 |
| Table 74. | Interconnect 3 | 397 |
| Table 75. | Interconnect 4 | 398 |
| Table 76. | Interconnect 5 | 399 |
| Table 77. | Interconnect 6 | 399 |
| Table 78. | Interconnect 7 | 399 |
| Table 79. | Interconnect 8 | 400 |
| Table 80. | Interconnect 9 | 400 |
| Table 81. | Interconnect 10 | 401 |
| Table 82. | Interconnect 11 | 401 |
| Table 83. | Interconnect 18 | 402 |
| Table 84. | Interconnect 16 | 402 |
| Table 85. | DMA1 and DMA2 implementation | 404 |
| Table 86. | DMA internal input/output signals | 406 |
| Table 87. | Programmable data width and endian behavior (when PINC = MINC = 1) | 411 |
| Table 88. | DMA interrupt requests | 413 |
| Table 89. | DMA register map and reset values | 421 |
| Table 90. | DMAMUX instantiation | 425 |
| Table 91. | DMAMUX: assignment of multiplexer inputs to resources | 426 |
| Table 93. | DMAMUX: assignment of synchronization inputs to resources | 427 |
| Table 92. | DMAMUX: assignment of trigger inputs to resources | 427 |
| Table 94. | DMAMUX signals | 428 |
| Table 95. | DMAMUX interrupts | 433 |
| Table 96. | DMAMUX register map and reset values | 438 |
| Table 97. | STM32G4 Series vector table | 441 |
| Table 98. | EXTI lines connections | 448 |
| Table 99. | Extended interrupt/event controller register map and reset values | 459 |

| | |
|--|-----|
| Table 100. CRC internal input/output signals | 461 |
| Table 101. CRC register map and reset values | 466 |
| Table 102. CORDIC functions | 468 |
| Table 103. Cosine parameters | 468 |
| Table 104. Sine parameters | 469 |
| Table 105. Phase parameters | 469 |
| Table 106. Modulus parameters | 470 |
| Table 107. Arctangent parameters | 471 |
| Table 108. Hyperbolic cosine parameters | 471 |
| Table 109. Hyperbolic sine parameters | 472 |
| Table 110. Hyperbolic arctangent parameters | 472 |
| Table 111. Natural logarithm parameters | 473 |
| Table 112. Natural log scaling factors and corresponding ranges | 473 |
| Table 113. Square root parameters | 474 |
| Table 114. Square root scaling factors and corresponding ranges | 474 |
| Table 115. Precision vs. number of iterations | 477 |
| Table 116. CORDIC register map and reset value | 484 |
| Table 117. Valid combinations for read and write methods | 498 |
| Table 118. FMAC register map and reset values | 511 |
| Table 119. NOR/PSRAM bank selection | 516 |
| Table 120. NOR/PSRAM External memory address | 517 |
| Table 121. NAND memory mapping and timing registers | 517 |
| Table 122. NAND bank selection | 517 |
| Table 123. Programmable NOR/PSRAM access parameters | 519 |
| Table 124. Non-multiplexed I/O NOR Flash memory | 519 |
| Table 125. 16-bit multiplexed I/O NOR Flash memory | 520 |
| Table 126. Non-multiplexed I/Os PSRAM/SRAM | 520 |
| Table 127. 16-Bit multiplexed I/O PSRAM | 521 |
| Table 128. NOR Flash/PSRAM: example of supported memories and transactions | 522 |
| Table 129. FMC_BCRx bitfields (mode 1) | 525 |
| Table 130. FMC_BTRx bitfields (mode 1) | 525 |
| Table 131. FMC_BCRx bitfields (mode A) | 527 |
| Table 132. FMC_BTRx bitfields (mode A) | 527 |
| Table 133. FMC_BWTRx bitfields (mode A) | 528 |
| Table 134. FMC_BCRx bitfields (mode 2/B) | 530 |
| Table 135. FMC_BTRx bitfields (mode 2/B) | 530 |
| Table 136. FMC_BWTRx bitfields (mode 2/B) | 531 |
| Table 137. FMC_BCRx bitfields (mode C) | 532 |
| Table 138. FMC_BTRx bitfields (mode C) | 533 |
| Table 139. FMC_BWTRx bitfields (mode C) | 533 |
| Table 140. FMC_BCRx bitfields (mode D) | 535 |
| Table 141. FMC_BTRx bitfields (mode D) | 536 |
| Table 142. FMC_BWTRx bitfields (mode D) | 536 |
| Table 143. FMC_BCRx bitfields (Muxed mode) | 538 |
| Table 144. FMC_BTRx bitfields (Muxed mode) | 539 |
| Table 145. FMC_BCRx bitfields (Synchronous multiplexed read mode) | 544 |
| Table 146. FMC_BTRx bitfields (Synchronous multiplexed read mode) | 545 |
| Table 147. FMC_BCRx bitfields (Synchronous multiplexed write mode) | 546 |
| Table 148. FMC_BTRx bitfields (Synchronous multiplexed write mode) | 547 |
| Table 149. Programmable NAND Flash access parameters | 556 |
| Table 150. 8-bit NAND Flash | 556 |

| | | |
|------------|--|-----|
| Table 151. | 16-bit NAND Flash | 557 |
| Table 152. | Supported memories and transactions | 557 |
| Table 153. | ECC result relevant bits | 566 |
| Table 154. | FMC register map and reset values | 567 |
| Table 155. | QUADSPI pins | 570 |
| Table 156. | QUADSPI interrupt requests | 584 |
| Table 157. | QUADSPI register map and reset values | 595 |
| Table 158. | ADC features | 599 |
| Table 159. | ADC internal input/output signals | 601 |
| Table 160. | ADC input/output pins | 601 |
| Table 161. | Configuring the trigger polarity for regular external triggers | 622 |
| Table 162. | Configuring the trigger polarity for injected external triggers | 622 |
| Table 163. | ADC1/2 - External triggers for regular channels | 623 |
| Table 164. | ADC1/2 - External trigger for injected channels | 624 |
| Table 165. | ADC3/4/5 - External triggers for regular channels | 625 |
| Table 166. | ADC3/4/5 - External triggers for injected channels | 626 |
| Table 167. | TSAR timings depending on resolution | 639 |
| Table 168. | Offset computation versus data resolution | 642 |
| Table 169. | Analog watchdog channel selection | 653 |
| Table 170. | Analog watchdog 1 comparison | 654 |
| Table 171. | Analog watchdog 2 and 3 comparison | 655 |
| Table 172. | Maximum output results versus N and M (gray cells indicate truncation) | 658 |
| Table 173. | Oversampler operating modes summary | 663 |
| Table 174. | Effect of low-power modes on the ADC | 681 |
| Table 175. | ADC interrupts per each ADC | 682 |
| Table 176. | DELAY bits versus ADC resolution | 716 |
| Table 177. | ADC global register map | 717 |
| Table 178. | ADC register map and reset values for each ADC (offset = 0x000 for master ADC, 0x100 for slave ADC) | 718 |
| Table 179. | ADC register map and reset values (master and slave ADC common registers) offset = 0x300 | 720 |
| Table 180. | DAC features | 722 |
| Table 181. | DAC input/output pins | 724 |
| Table 182. | DAC input/output signals | 724 |
| Table 183. | DAC1 interconnection | 725 |
| Table 184. | DAC2 interconnection | 726 |
| Table 185. | DAC3 interconnection | 727 |
| Table 186. | DAC4 interconnection | 728 |
| Table 187. | Data format (case of 12-bit data) | 731 |
| Table 188. | HFSEL description | 732 |
| Table 189. | Sample and refresh timings | 739 |
| Table 190. | Channel output modes summary | 741 |
| Table 191. | Effect of low-power modes on DAC | 748 |
| Table 192. | DAC interrupts | 748 |
| Table 193. | DAC register map and reset values | 768 |
| Table 194. | VREF buffer modes | 770 |
| Table 195. | VREFBUF register map and reset values | 772 |
| Table 196. | COMP _x non-inverting input assignment | 774 |
| Table 197. | COMP _x inverting input assignment | 774 |
| Table 198. | Blanking sources | 776 |
| Table 199. | Comparator behavior in low-power modes | 776 |
| Table 200. | COMP register map and reset values | 779 |

| | |
|---|------|
| Table 201. Operational amplifier possible connection | 782 |
| Table 202. Operating modes and calibration | 790 |
| Table 203. Effect of low-power modes on the OPAMP | 792 |
| Table 204. OPAMP register map and reset values | 823 |
| Table 205. RNG internal input/output signals | 826 |
| Table 206. RNG interrupt requests | 833 |
| Table 207. RNG configurations | 833 |
| Table 208. RNG register map and reset map | 836 |
| Table 209. HRTIM inputs/outputs summary | 841 |
| Table 210. External events mapping and associated features | 843 |
| Table 211. Update enable inputs and sources | 843 |
| Table 212. Burst mode clock sources from general purpose timer | 843 |
| Table 213. Fault inputs | 844 |
| Table 214. HRTIM DAC triggers connections | 844 |
| Table 215. Timer resolution and min. PWM frequency for $f_{HRTIM} = 170$ MHz | 845 |
| Table 216. Period and compare registers min and max values | 849 |
| Table 217. Timer operating modes | 849 |
| Table 218. Events mapping across timer A to F | 855 |
| Table 219. Interleaved mode selection | 856 |
| Table 220. Compare 1..3 values in interleaved mode | 857 |
| Table 221. Deadtime resolution and max absolute values | 868 |
| Table 222. Roll-over event destination and mode programming | 875 |
| Table 223. EExFLTR[3:0] codes depending on UDM bit setting | 877 |
| Table 224. External events features | 883 |
| Table 225. Output set/reset latency and jitter versus external event operating mode | 884 |
| Table 226. Filtering signals mapping per timer | 887 |
| Table 227. Windowing signals mapping per timer (EEFLTR[3:0] = 1111) | 889 |
| Table 228. HRTIM preloadable control registers and associated update sources | 899 |
| Table 229. Master timer update event propagation | 904 |
| Table 230. TIMx update event propagation | 904 |
| Table 231. Reset events able to generate an update | 906 |
| Table 232. Update event propagation for a timer reset | 906 |
| Table 233. Output state programming, x=A..F, y = 1 or 2 | 908 |
| Table 234. Timer output programming for burst mode | 911 |
| Table 235. Fault inputs | 920 |
| Table 236. Sampling rate and filter length vs FLTFxF[3:0] and clock setting | 922 |
| Table 237. Fault input blanking events | 922 |
| Table 238. Fault 1..6 counter reset source | 923 |
| Table 239. Effect of sync event versus timer operating modes | 928 |
| Table 240. DAC dual channel trigger example | 936 |
| Table 241. HRTIM interrupt summary | 939 |
| Table 242. HRTIM DMA request summary | 941 |
| Table 243. HRTIM Register map and reset values – master timer | 1067 |
| Table 244. HRTIM register map and reset values – TIMx (x= A..F) | 1068 |
| Table 245. HRTIM register map and reset values – common functions | 1073 |
| Table 246. TIM input/output pins | 1079 |
| Table 247. TIM internal input/output signals | 1079 |
| Table 248. Interconnect to the tim_t11 input multiplexer | 1080 |
| Table 249. Interconnect to the tim_t12 input multiplexer | 1081 |
| Table 250. Interconnect to the tim_t13 input multiplexer | 1081 |
| Table 251. Interconnect to the tim_t14 input multiplexer | 1081 |
| Table 252. TIMx internal trigger connection | 1081 |

| | |
|---|------|
| Table 253. Interconnect to the tim_etr input multiplexer..... | 1082 |
| Table 254. Timer break interconnect | 1082 |
| Table 255. Timer break2 interconnect | 1083 |
| Table 256. System break interconnect | 1083 |
| Table 257. Interconnect to the ocref_clr input multiplexer | 1083 |
| Table 258. CCR and ARR register change dithering pattern | 1117 |
| Table 259. CCR register change dithering pattern in center-aligned PWM mode | 1118 |
| Table 260. Behavior of timer outputs versus tim_brk/tim_brk2 inputs | 1130 |
| Table 261. Break protection disarming conditions | 1132 |
| Table 262. Counting direction versus encoder signals (CC1P = CC2P = 0) | 1141 |
| Table 263. Counting direction versus encoder signals and polarity settings | 1145 |
| Table 264. DMA request | 1167 |
| Table 265. Effect of low-power modes on TIM1/TIM8/TIM20 | 1168 |
| Table 266. Interrupt requests | 1168 |
| Table 267. Output control bits for complementary tim_ocx and tim_ocxn channels with break feature | 1195 |
| Table 268. TIMx register map and reset values | 1218 |
| Table 269. STM32G4 Series general purpose timers | 1222 |
| Table 270. TIM input/output pins | 1224 |
| Table 271. TIM internal input/output signals | 1224 |
| Table 272. Interconnect to the tim_ti1 input multiplexer | 1225 |
| Table 273. Interconnect to the tim_ti2 input multiplexer | 1225 |
| Table 274. Interconnect to the tim_ti3 input multiplexer | 1226 |
| Table 275. Interconnect to the tim_ti4 input multiplexer | 1226 |
| Table 276. TIMx internal trigger connection | 1226 |
| Table 277. Interconnect to the tim_etr input multiplexer | 1227 |
| Table 278. Interconnect to the tim_ocref_clr input multiplexer | 1228 |
| Table 279. CCR and ARR register change dithering pattern | 1259 |
| Table 280. CCR register change dithering pattern in center-aligned PWM mode | 1260 |
| Table 281. Counting direction versus encoder signals(CC1P = CC2P = 0) | 1269 |
| Table 282. Counting direction versus encoder signals and polarity settings | 1274 |
| Table 283. DMA request | 1298 |
| Table 284. Effect of low-power modes on TIM2/TIM3/TIM4/TIM5 | 1298 |
| Table 285. Interrupt requests | 1299 |
| Table 286. Output control bit for standard tim_ocx channels | 1318 |
| Table 287. TIM2/TIM3/TIM4/TIM5 register map and reset values | 1335 |
| Table 288. TIM input/output pins | 1341 |
| Table 289. TIM internal input/output signals | 1342 |
| Table 290. Interconnect to the tim_ti1 input multiplexer | 1342 |
| Table 291. Interconnect to the tim_ti2 input multiplexer | 1343 |
| Table 292. TIMx internal trigger connection | 1343 |
| Table 293. Timer break interconnect | 1344 |
| Table 294. System break interconnect | 1344 |
| Table 295. Interconnect to the ocref_clr input multiplexer | 1344 |
| Table 296. CCR and ARR register change dithering pattern | 1365 |
| Table 297. Break protection disarming conditions | 1374 |
| Table 298. DMA request | 1384 |
| Table 299. Effect of low-power modes on TIM15/TIM16/TIM17 | 1385 |
| Table 300. Interrupt requests | 1385 |
| Table 301. Output control bits for complementary tim_ocx and tim_ocxn channels with break feature (TIM15) | 1400 |
| Table 302. TIM15 register map and reset values | 1413 |

| | |
|---|------|
| Table 303. Output control bits for complementary tim_oc1 and tim_oc1n channels with break feature (TIM16/TIM17) | 1426 |
| Table 304. TIM16/TIM17 register map and reset values | 1440 |
| Table 305. TIM internal input/output signals | 1443 |
| Table 306. TIMx_ARR register change dithering pattern | 1453 |
| Table 307. DMA request | 1454 |
| Table 308. Effect of low-power modes on TIM6/TIM7 | 1454 |
| Table 309. Interrupt request | 1454 |
| Table 310. TIMx register map and reset values | 1460 |
| Table 311. STM32G4 Series LPTIM features | 1462 |
| Table 312. LPTIM1 external trigger connection | 1462 |
| Table 313. LPTIM1 input 1 connection | 1463 |
| Table 314. LPTIM1 input 2 connection | 1463 |
| Table 315. Prescaler division ratios | 1465 |
| Table 316. Encoder counting scenarios | 1472 |
| Table 317. Effect of low-power modes on the LPTIM | 1473 |
| Table 318. Interrupt events | 1474 |
| Table 319. LPTIM register map and reset values | 1485 |
| Table 320. AES internal input/output signals | 1488 |
| Table 321. CTR mode initialization vector definition | 1505 |
| Table 322. GCM last block definition | 1507 |
| Table 323. Initialization of AES_MR _x registers in GCM mode | 1508 |
| Table 324. Initialization of AES_MR _x registers in CCM mode | 1515 |
| Table 325. Key endianness in AES_KEYR _x registers (128- or 256-bit key length) | 1520 |
| Table 326. AES interrupt requests | 1523 |
| Table 327. Processing latency for ECB, CBC and CTR | 1523 |
| Table 328. Processing latency for GCM and CCM (in clockcycles) | 1524 |
| Table 329. AES register map and reset values | 1534 |
| Table 330. RTC input/output pins | 1538 |
| Table 331. RTC internal input/output signals | 1538 |
| Table 332. RTC interconnection | 1539 |
| Table 333. PC 13 configuration | 1539 |
| Table 334. RTC_OUT mapping | 1541 |
| Table 335. Effect of low-power modes on RTC | 1552 |
| Table 336. RTC pins functionality over modes | 1552 |
| Table 337. Interrupt requests | 1553 |
| Table 338. RTC register map and reset values | 1573 |
| Table 339. TAMP input/output pins | 1577 |
| Table 340. TAMP internal input/output signals | 1577 |
| Table 341. TAMP interconnection | 1577 |
| Table 342. Effect of low-power modes on TAMP | 1580 |
| Table 343. Interrupt requests | 1580 |
| Table 344. TAMP register map and reset values | 1589 |
| Table 345. USART / LPUART features | 1592 |
| Table 346. Noise detection from sampled data | 1607 |
| Table 347. Tolerance of the USART receiver when BRR[3:0] = 0000 | 1610 |
| Table 348. Tolerance of the USART receiver when BRR[3:0] is different from 0000 | 1611 |
| Table 349. USART frame formats | 1616 |
| Table 350. Effect of low-power modes on the USART | 1639 |
| Table 351. USART interrupt requests | 1640 |
| Table 352. USART register map and reset values | 1675 |
| Table 353. USART / LP UART features | 1679 |

| | |
|--|------|
| Table 354. Error calculation for programmed baud rates at lpuart_ker_ck_pres = 32.768 kHz | 1690 |
| Table 355. Error calculation for programmed baud rates at fCK = 100 MHz | 1691 |
| Table 356. Tolerance of the LPUART receiver | 1692 |
| Table 358. Effect of low-power modes on the LPUART | 1703 |
| Table 359. LPUART interrupt requests | 1704 |
| Table 360. LPUART register map and reset values | 1728 |
| Table 361. STM32G4 Series SPI and SPI/I2S implementation | 1731 |
| Table 362. SPI interrupt requests | 1756 |
| Table 363. Audio-frequency precision using 48 MHz clock derived from HSE | 1769 |
| Table 364. I2S interrupt requests | 1775 |
| Table 365. SPI/I2S register map and reset values | 1787 |
| Table 366. STM32G4 Series SAI features | 1789 |
| Table 367. SAI internal input/output signals | 1791 |
| Table 368. SAI input/output pins | 1791 |
| Table 369. MCLK_x activation conditions | 1798 |
| Table 370. Clock generator programming examples | 1801 |
| Table 371. TDM settings | 1808 |
| Table 372. TDM frame configuration examples | 1810 |
| Table 373. SOPD pattern | 1813 |
| Table 374. Parity bit calculation | 1813 |
| Table 375. Audio sampling frequency versus symbol rates | 1814 |
| Table 376. SAI interrupt sources | 1823 |
| Table 377. SAI register map and reset values | 1853 |
| Table 378. I2C implementation | 1856 |
| Table 379. I2C input/output pins | 1858 |
| Table 380. I2C internal input/output signals | 1858 |
| Table 381. Comparison of analog vs. digital filters | 1860 |
| Table 382. I2C-SMBus specification data setup and hold times | 1863 |
| Table 383. I2C configuration | 1867 |
| Table 384. I2C-SMBus specification clock timings | 1878 |
| Table 385. Examples of timing settings for fI2CCLK = 8 MHz | 1888 |
| Table 386. Examples of timing settings for fI2CCLK = 16 MHz | 1888 |
| Table 387. Examples of timing settings for fI2CCLK = 48 MHz | 1889 |
| Table 388. SMBus timeout specifications | 1891 |
| Table 389. SMBus with PEC configuration | 1893 |
| Table 390. Examples of TIMEOUTA settings for various I2CC LK frequencies (max t _{TIMEOUT} = 25 ms) | 1894 |
| Table 391. Examples of TIMEOUTB settings for various I2CCLK frequencies | 1894 |
| Table 392. Examples of TIMEOUTA settings for various I2CC LK frequencies (max t _{IDLE} = 50 µs) | 1894 |
| Table 393. Effect of low-power modes on the I2C | 1905 |
| Table 394. I2C interrupt requests | 1906 |
| Table 395. I2C register map and reset values | 1921 |
| Table 396. IWDG register map and reset values | 1931 |
| Table 397. WWDG register map and reset values | 1937 |
| Table 398. CAN subsystem I/O signals | 1938 |
| Table 399. DLC coding in FDCAN | 1946 |
| Table 400. Possible configurations for Frame transmission | 1958 |
| Table 401. Rx FIFO element | 1961 |
| Table 402. Rx FIFO element description | 1961 |
| Table 403. Tx buffer and FIFO element | 1963 |
| Table 404. Tx buffer element description | 1963 |

| | | |
|------------|---|------|
| Table 405. | Tx event FIFO element | 1965 |
| Table 406. | Tx event FIFO element description | 1965 |
| Table 407. | Standard Message ID Filter element | 1966 |
| Table 408. | Standard Message ID Filter element Field description | 1966 |
| Table 409. | Extended Message ID Filter element | 1967 |
| Table 410. | Extended Message ID Filter element field description | 1967 |
| Table 411. | FDCAN register map and reset values | 1999 |
| Table 412. | STM32G4 Series USB implementation | 2003 |
| Table 413. | Double-buffering buffer flag definition | 2013 |
| Table 414. | Bulk double-buffering memory buffers usage | 2013 |
| Table 415. | Isochronous memory buffers usage | 2015 |
| Table 416. | Resume event detection | 2016 |
| Table 417. | Reception status encoding | 2029 |
| Table 418. | Endpoint type encoding | 2029 |
| Table 419. | Endpoint kind meaning | 2029 |
| Table 420. | Transmission status encoding | 2030 |
| Table 421. | Definition of allocated buffer memory | 2033 |
| Table 422. | USB register map and reset values | 2034 |
| Table 423. | UCPD implementation | 2037 |
| Table 424. | UCPD signals on pins | 2038 |
| Table 425. | UCPD internal signals | 2039 |
| Table 426. | 4b5b Symbol encoding table | 2040 |
| Table 427. | Ordered sets | 2042 |
| Table 428. | Validation of ordered sets | 2042 |
| Table 429. | Data size | 2043 |
| Table 430. | Coding for ANAMODE, ANASUBMODE and link with TYPEC_VSTATE_CCx | 2051 |
| Table 431. | Type-C sequence (source: 3A); cable/sink connected (Rd on CC 1; Ra on CC 2) | 2053 |
| Table 432. | Effect of low power modes on the UCPD | 2055 |
| Table 433. | UCPD interrupt requests | 2056 |
| Table 434. | UCPD register map and reset values | 2072 |
| Table 435. | SWJ debug port pins | 2078 |
| Table 436. | Flexible SWJ-DP pin assignment | 2078 |
| Table 437. | JTAG debug port data registers | 2083 |
| Table 438. | 32-bit debug port registers addressed through the shifted value A[3:2] | 2084 |
| Table 439. | Packet request (8-bits) | 2085 |
| Table 440. | ACK response (3 bits) | 2086 |
| Table 441. | DATA transfer (33 bits) | 2086 |
| Table 442. | SW-DP registers | 2087 |
| Table 443. | Cortex®-M4 with FPU AHB-AP registers | 2088 |
| Table 444. | Core debug registers | 2089 |
| Table 445. | Main ITM registers | 2092 |
| Table 446. | Main ETM registers | 2093 |
| Table 447. | Asynchronous TRACE pin assignment | 2100 |
| Table 448. | Synchronous TRACE pin assignment | 2101 |
| Table 449. | Flexible TRACE pin assignment | 2101 |
| Table 450. | Important TPIU registers | 2105 |
| Table 451. | DBG register map and reset values | 2107 |
| Table 452. | Document revision history | 2111 |

List of figures

| | | |
|------------|---|-----|
| Figure 1. | System architecture | 79 |
| Figure 2. | Memory map | 82 |
| Figure 3. | Sequential 16-bit instructions execution (64-bit read data width) | 100 |
| Figure 4. | Changing the read protection (RDP) level | 122 |
| Figure 5. | Example of disabling core debug access | 127 |
| Figure 6. | Sequential 16-bit instructions execution (64-bit read data width) | 153 |
| Figure 7. | Changing the read protection (RDP) level | 168 |
| Figure 8. | Example of disabling core debug access | 172 |
| Figure 9. | Sequential 16-bit instructions execution (64-bit read data width) | 194 |
| Figure 10. | Changing the read protection (RDP) level | 209 |
| Figure 11. | Example of disabling core debug access | 213 |
| Figure 12. | STM32G4 Series power supply overview | 231 |
| Figure 13. | Brown-out reset waveform | 237 |
| Figure 14. | PVD thresholds | 238 |
| Figure 15. | Low-power modes possible transitions | 240 |
| Figure 16. | Simplified diagram of the reset circuit | 274 |
| Figure 17. | Clock tree | 278 |
| Figure 18. | HSE/ LSE clock sources | 279 |
| Figure 19. | Frequency measurement with TIM15 in capture mode | 286 |
| Figure 20. | Frequency measurement with TIM16 in capture mode | 286 |
| Figure 21. | Frequency measurement with TIM17 in capture mode | 287 |
| Figure 22. | Frequency measurement with TIM5 in capture mode | 287 |
| Figure 23. | CRS block diagram | 341 |
| Figure 24. | CRS counter behavior | 342 |
| Figure 25. | Basic structure of an I/O port bit | 352 |
| Figure 26. | Basic structure of a 5-Volt tolerant I/O port bit | 352 |
| Figure 27. | Input floating/pull up/pull down configurations | 357 |
| Figure 28. | Output configuration | 358 |
| Figure 29. | Alternate function configuration | 358 |
| Figure 30. | High impedance-analog configuration | 359 |
| Figure 31. | DMA block diagram | 405 |
| Figure 32. | DMAMUX block diagram | 428 |
| Figure 33. | Synchronization mode of the DMAMUX request line multiplexer channel | 431 |
| Figure 34. | Event generation of the DMA request line multiplexer channel | 431 |
| Figure 35. | Configurable interrupt/event block diagram | 446 |
| Figure 36. | External interrupt/event GPIO mapping | 448 |
| Figure 37. | CRC calculation unit block diagram | 461 |
| Figure 38. | CORDIC convergence for trigonometric functions | 475 |
| Figure 39. | CORDIC convergence for hyperbolic functions | 476 |
| Figure 40. | CORDIC convergence for square root | 477 |
| Figure 41. | Block diagram | 486 |
| Figure 42. | Input buffer areas | 488 |
| Figure 43. | Circular input buffer | 489 |
| Figure 44. | Circular input buffer operation | 490 |
| Figure 45. | Circular output buffer | 491 |
| Figure 46. | Circular output buffer operation | 492 |
| Figure 47. | FIR filter structure | 494 |
| Figure 48. | IIR filter structure (direct form 1) | 496 |

| | | |
|-------------|---|-----|
| Figure 49. | X1 buffer initialization | 501 |
| Figure 50. | Filtering example 1 | 502 |
| Figure 51. | Filtering example 2 | 503 |
| Figure 52. | FMC block diagram | 514 |
| Figure 53. | FMC memory banks | 516 |
| Figure 54. | Mode 1 read access waveforms | 524 |
| Figure 55. | Mode 1 write access waveforms | 524 |
| Figure 56. | Mode A read access waveforms | 526 |
| Figure 57. | Mode A write access waveforms | 526 |
| Figure 58. | Mode 2 and mode B read access waveforms | 528 |
| Figure 59. | Mode 2 write access waveforms | 529 |
| Figure 60. | Mode B write access waveforms | 529 |
| Figure 61. | Mode C read access waveforms | 531 |
| Figure 62. | Mode C write access waveforms | 532 |
| Figure 63. | Mode D read access waveforms | 534 |
| Figure 64. | Mode D write access waveforms | 535 |
| Figure 65. | Muxed read access waveforms | 537 |
| Figure 66. | Muxed write access waveforms | 538 |
| Figure 67. | Asynchronous wait during a read access waveforms | 540 |
| Figure 68. | Asynchronous wait during a write access waveforms | 541 |
| Figure 69. | Wait configuration waveforms | 543 |
| Figure 70. | Synchronous multiplexed read mode waveforms - NOR, PSRAM (CRAM) | 544 |
| Figure 71. | Synchronous multiplexed write mode waveforms - PSRAM (CRAM) | 546 |
| Figure 72. | NAND Flash controller waveforms for common memory access | 558 |
| Figure 73. | Access to non 'CE don't care' NAND-Flash | 560 |
| Figure 74. | QUADSPI block diagram when dual-flash mode is disabled | 569 |
| Figure 75. | QUADSPI block diagram when dual-flash mode is enabled | 570 |
| Figure 76. | An example of a read command in quad mode | 571 |
| Figure 77. | An example of a DDR command in quad mode | 574 |
| Figure 78. | nCS when CKMODE = 0 (T = CLK period) | 582 |
| Figure 79. | nCS when CKMODE = 1 in SDR mode (T = CLK period) | 582 |
| Figure 80. | nCS when CKMODE = 1 in DDR mode (T = CLK period) | 583 |
| Figure 81. | nCS when CKMODE = 1 with an abort (T = CLK period) | 583 |
| Figure 82. | ADC block diagram | 600 |
| Figure 83. | ADC clock scheme | 603 |
| Figure 84. | ADC1 connectivity | 604 |
| Figure 85. | ADC2 connectivity | 605 |
| Figure 86. | ADC3 connectivity | 606 |
| Figure 87. | ADC4 connectivity | 607 |
| Figure 88. | ADC5 connectivity | 608 |
| Figure 89. | ADC calibration | 611 |
| Figure 90. | Updating the ADC calibration factor | 612 |
| Figure 91. | Mixing single-ended and differential channels | 613 |
| Figure 92. | Enabling / disabling the ADC | 614 |
| Figure 93. | Bulk mode timing diagram | 617 |
| Figure 94. | Analog to digital conversion time | 620 |
| Figure 95. | Stopping ongoing regular conversions | 621 |
| Figure 96. | Stopping ongoing regular and injected conversions | 621 |
| Figure 97. | Triggers sharing between ADC master and ADC slave | 623 |
| Figure 98. | Injected conversion latency | 629 |
| Figure 99. | Example of JSQR queue of context (sequence change) | 632 |
| Figure 100. | Example of JSQR queue of context (trigger change) | 632 |

| | |
|--|-----|
| Figure 101. Example of JSQR queue of context with overflow before conversion | 633 |
| Figure 102. Example of JSQR queue of context with overflow during conversion | 633 |
| Figure 103. Example of JSQR queue of context with empty queue (case JQM = 0) | 634 |
| Figure 104. Example of JSQR queue of context with empty queue (case JQM = 1) | 635 |
| Figure 105. Flushing JSQR queue of context by setting JADSTP = 1 (JQM = 0) Case when JADSTP occurs during an ongoing conversion | 635 |
| Figure 106. Flushing JSQR queue of context by setting JADSTP = 1 (JQM = 0) Case when JADSTP occurs during an ongoing conversion and a new trigger occurs | 636 |
| Figure 107. Flushing JSQR queue of context by setting JADSTP = 1 (JQM = 0) Case when JADSTP occurs outside an ongoing conversion | 636 |
| Figure 108. Flushing JSQR queue of context by setting JADSTP = 1 (JQM = 1) | 637 |
| Figure 109. Flushing JSQR queue of context by setting ADDIS = 1 (JQM = 0) | 637 |
| Figure 110. Flushing JSQR queue of context by setting ADDIS = 1 (JQM = 1) | 638 |
| Figure 111. Single conversions of a sequence, software trigger | 640 |
| Figure 112. Continuous conversion of a sequence, software trigger | 640 |
| Figure 113. Single conversions of a sequence, hardware trigger | 641 |
| Figure 114. Continuous conversions of a sequence, hardware trigger | 641 |
| Figure 115. Right alignment (offset disabled, unsigned value) | 643 |
| Figure 116. Right alignment (offset enabled, signed value) | 644 |
| Figure 117. Left alignment (offset disabled, unsigned value) | 644 |
| Figure 118. Left alignment (offset enabled, signed value) | 645 |
| Figure 119. Example of overrun (OVR) | 646 |
| Figure 120. AUTODLY = 1, regular conversion in continuous mode, software trigger | 649 |
| Figure 121. AUTODLY = 1, regular HW conversions interrupted by injected conversions (DISCEN = 0; JDISCEN = 0) | 650 |
| Figure 122. AUTODLY = 1, regular HW conversions interrupted by injected conversions (DISCEN = 1, JDISCEN = 1) | 651 |
| Figure 123. AUTODLY = 1, regular continuous conversions interrupted by injected conversions | 652 |
| Figure 124. AUTODLY = 1 in auto-injected mode (JAUTO = 1) | 652 |
| Figure 125. Analog watchdog guarded area | 653 |
| Figure 126. ADCy_AWDx_OUT signal generation (on all regular channels) | 655 |
| Figure 127. ADCy_AWDx_OUT signal generation (AWDx flag not cleared by software) | 656 |
| Figure 128. ADCy_AWDx_OUT signal generation (on a single regular channel) | 656 |
| Figure 129. ADCy_AWDx_OUT signal generation (on all injected channels) | 656 |
| Figure 130. 20-bit to 16-bit result truncation | 658 |
| Figure 131. Numerical example with 5-bit shift and rounding | 658 |
| Figure 132. Triggered regular oversampling mode (TROVS bit = 1) | 660 |
| Figure 133. Regular oversampling modes (4x ratio) | 661 |
| Figure 134. Regular and injected oversampling modes used simultaneously | 661 |
| Figure 135. Triggered regular oversampling with injection | 662 |
| Figure 136. Oversampling in auto-injected mode | 662 |
| Figure 137. Dual ADC block diagram ⁽¹⁾ | 665 |
| Figure 138. Injected simultaneous mode on 4 channels: dual ADC mode | 666 |
| Figure 139. Regular simultaneous mode on 16 channels: dual ADC mode | 668 |
| Figure 140. Interleaved mode on 1 channel in continuous conversion mode: dual ADC mode | 669 |
| Figure 141. Interleaved mode on 1 channel in single conversion mode: dual ADC mode | 670 |
| Figure 142. Interleaved conversion with injection | 670 |
| Figure 143. Alternate trigger: injected group of each ADC | 671 |
| Figure 144. Alternate trigger: 4 injected channels (each ADC) in discontinuous mode | 672 |
| Figure 145. Alternate + regular simultaneous | 673 |
| Figure 146. Case of trigger occurring during injected conversion | 673 |

| | |
|--|-----|
| Figure 147. Interleaved single channel CH0 with injected sequence CH11, CH12 | 674 |
| Figure 148. Two Interleaved channels (CH1, CH2) with injected sequence CH11, CH12 - case 1: Master interrupted first | 674 |
| Figure 149. Two Interleaved channels (CH1, CH2) with injected sequence CH11, CH12 - case 2: Slave interrupted first | 674 |
| Figure 150. DMA Requests in regular simultaneous mode when MDMA = 00 | 675 |
| Figure 151. DMA requests in regular simultaneous mode when MDMA = 10 | 676 |
| Figure 152. DMA requests in interleaved mode when MDMA = 10 | 676 |
| Figure 153. Temperature sensor channel block diagram | 678 |
| Figure 154. VBAT channel block diagram | 679 |
| Figure 155. VREFINT channel block diagram | 680 |
| Figure 156. Dual-channel DAC block diagram | 723 |
| Figure 157. Data registers in single DAC channel mode | 730 |
| Figure 158. Data registers in dual DAC channel mode | 731 |
| Figure 159. Timing diagram for conversion with trigger disabled TEN = 0 | 733 |
| Figure 160. DAC LFSR register calculation algorithm | 735 |
| Figure 161. DAC conversion (SW trigger enabled) with LFSR wave generation | 735 |
| Figure 162. DAC triangle wave generation | 736 |
| Figure 163. DAC conversion (SW trigger enabled) with triangle wave generation | 736 |
| Figure 164. DAC sawtooth wave generation (STDIRx = 0) | 737 |
| Figure 165. DAC sawtooth wave generation (STDIRx = 1) | 737 |
| Figure 166. DAC sawtooth STINCTRIG and STRSTTRIG priority (STDIR = 0) | 738 |
| Figure 167. DAC Sample and hold mode phase diagram | 740 |
| Figure 168. Comparator block diagram | 773 |
| Figure 169. Comparator hysteresis | 775 |
| Figure 170. Comparator output blanking | 776 |
| Figure 171. Standalone mode: external gain setting mode | 784 |
| Figure 172. Follower configuration | 785 |
| Figure 173. PGA mode, internal gain setting (x2/x4/x8/x16/x32/x64), inverting input not used | 786 |
| Figure 174. PGA mode, internal gain setting (x2/x4/x8/x16/x32/x64), inverting input used for filtering | 787 |
| Figure 175. PGA mode, non-inverting gain setting (x2/x4/x8/x16/x32/x64) or inverting gain setting (x-1/x-3/x-7/x-15/x-31/x-63) | 788 |
| Figure 176. PGA mode, non-inverting gain setting (x2/x4/x8/x16/x32/x64) or inverting gain setting (x-1/x-3/x-7/x-15/x-31/x-63) with filtering | 789 |
| Figure 177. Example configuration | 789 |
| Figure 178. Timer controlled Multiplexer mode | 792 |
| Figure 179. RNG block diagram | 826 |
| Figure 180. Entropy source model | 827 |
| Figure 181. RNG initialization overview | 830 |
| Figure 182. High-resolution timer overview | 840 |
| Figure 183. Counter and capture register format vs clock prescaling factor | 846 |
| Figure 184. Timer A..F overview | 848 |
| Figure 185. Continuous timer operation | 850 |
| Figure 186. Single-shot timer operation | 850 |
| Figure 187. Timer reset resynchronization (prescaling ratio above 32) | 852 |
| Figure 188. Repetition rate versus HRTIM_REPxR content in continuous mode | 853 |
| Figure 189. Repetition counter behavior in single-shot mode | 853 |
| Figure 190. Compare events action on outputs: set on compare 1, reset on compare 2 | 855 |
| Figure 191. Timer A timing unit capture circuitry | 859 |
| Figure 192. Auto-delayed overview (compare 2 only) | 860 |

| | |
|--|-----|
| Figure 193. Auto-delayed compare | 861 |
| Figure 194. Triggered-half mode example | 863 |
| Figure 195. Push-pull mode block diagram | 864 |
| Figure 196. Push-pull mode example | 865 |
| Figure 197. Push-Pull with deadtime | 866 |
| Figure 198. Complementary outputs with deadtime insertion | 867 |
| Figure 199. Deadtime insertion versus deadtime sign (1 indicates negative deadtime) | 867 |
| Figure 200. Complementary outputs for low pulsewidth ($SDTRx = SDTFx = 0$) | 868 |
| Figure 201. Complementary outputs for low pulsewidth ($SDTRx = SDTFx = 1$) | 868 |
| Figure 202. Complementary outputs for low pulsewidth ($SDTRx = 0, SDTFx = 1$) | 869 |
| Figure 203. Complementary outputs for low pulsewidth ($SDTRx = 1, SDTFx = 0$) | 869 |
| Figure 204. Master timer overview | 870 |
| Figure 205. Basic symmetric waveform in up-down counting mode | 871 |
| Figure 206. Complex symmetric waveform in up-down counting mode | 872 |
| Figure 207. Asymmetric waveform in up-down counting mode | 872 |
| Figure 208. External event management in up-down counting mode | 873 |
| Figure 209. Interleaved up-down counter operation | 873 |
| Figure 210. Interleaved up-down counter operation | 874 |
| Figure 211. Push-pull up-down mode example | 874 |
| Figure 212. Up-down mode with "greater than" comparison | 875 |
| Figure 213. Up-down mode with output set on period event, for $OUTROM[1:0]=10$ | 876 |
| Figure 214. Repetition counter behavior in up-down counting mode | 877 |
| Figure 215. Short distance set/reset management for narrow pulse generation | 879 |
| Figure 216. External event conditioning overview (1 channel represented) | 882 |
| Figure 217. Latency to external events falling edge (counter reset and output set) | 885 |
| Figure 218. Latency to external events (output reset on external event) | 885 |
| Figure 219. Event blanking mode | 886 |
| Figure 220. Event postpone mode | 886 |
| Figure 221. External trigger blanking with edge-sensitive trigger | 888 |
| Figure 222. External trigger blanking, level sensitive triggering | 888 |
| Figure 223. Event windowing mode | 889 |
| Figure 224. External trigger windowing with edge-sensitive trigger | 890 |
| Figure 225. External trigger windowing, level sensitive triggering | 890 |
| Figure 226. External event counter – channel A | 891 |
| Figure 227. External event counter cumulative mode ($EEVxRSTM = 1, EEVxCNT = 2$) | 892 |
| Figure 228. Delayed Idle mode entry | 893 |
| Figure 229. Burst mode and delayed protection priorities ($DIDL = 0$) | 894 |
| Figure 230. Burst mode and delayed protection priorities ($DIDL = 1$) | 895 |
| Figure 231. Balanced Idle protection example | 896 |
| Figure 232. Resynchronized timer update ($TAU=1$ in $HRTIM_TIMBCR$) | 901 |
| Figure 233. Early turn-ON and early turn-OFF behavior in "greater than" PWM mode | 903 |
| Figure 234. Output management overview | 908 |
| Figure 235. HRTIM output states and transitions | 909 |
| Figure 236. Burst mode operation example | 910 |
| Figure 237. Burst mode trigger on external event | 913 |
| Figure 238. Delayed burst mode entry with deadtime enabled and $IDLESx = 1$ | 914 |
| Figure 239. Delayed burst mode entry during deadtime | 915 |
| Figure 240. Burst mode exit when the deadtime generator is enabled | 916 |
| Figure 241. Burst mode emulation example | 918 |
| Figure 242. Carrier frequency signal insertion | 918 |
| Figure 243. HRTIM outputs with Chopper mode enabled | 919 |
| Figure 244. Fault protection circuitry (FAULT1 fully represented, FAULT2..6 partially) | 920 |

| | |
|--|------|
| Figure 245. Fault signal filtering (FLTxF[3:0]= 0010: f _{SAMPLING} = fHRTIM, N = 4) | 921 |
| Figure 246. Fault counter cumulative mode (FLTxRSTM = 1, FLTxCNT[3:0] = 2) | 923 |
| Figure 247. Auxiliary outputs | 925 |
| Figure 248. Auxiliary and main outputs during burst mode (DIDLx = 0) | 926 |
| Figure 249. Deadtime distortion on auxiliary output when exiting burst mode. | 926 |
| Figure 250. Counter behavior in synchronized start mode | 930 |
| Figure 251. ADC trigger selection overview | 931 |
| Figure 252. ADC triggers | 932 |
| Figure 253. ADC trigger post-scaling in up-counting mode | 933 |
| Figure 254. ADC trigger post-scaling in up/down counting mode | 933 |
| Figure 255. Combining several updates on a single hrtim_dac_trgx output | 934 |
| Figure 256. DAC triggers for slope compensation | 937 |
| Figure 257. DAC triggers overview | 938 |
| Figure 258. DMA burst overview | 942 |
| Figure 259. Burst DMA operation flowchart | 943 |
| Figure 260. Registers update following DMA burst transfer | 944 |
| Figure 261. Buck converter topology | 946 |
| Figure 262. Dual Buck converter management | 947 |
| Figure 263. Synchronous rectification depending on output current | 947 |
| Figure 264. Buck with synchronous rectification | 948 |
| Figure 265. 3-phase interleaved buck converter | 948 |
| Figure 266. 3-phase interleaved buck converter control | 949 |
| Figure 267. Transition mode PFC | 950 |
| Figure 268. Transition mode PFC waveforms | 951 |
| Figure 269. Advanced-control timer block diagram | 1078 |
| Figure 270. Counter timing diagram with prescaler division change from 1 to 2 | 1085 |
| Figure 271. Counter timing diagram with prescaler division change from 1 to 4 | 1085 |
| Figure 272. Counter timing diagram, internal clock divided by 1 | 1087 |
| Figure 273. Counter timing diagram, internal clock divided by 2 | 1087 |
| Figure 274. Counter timing diagram, internal clock divided by 4 | 1088 |
| Figure 275. Counter timing diagram, internal clock divided by N | 1088 |
| Figure 276. Counter timing diagram, update event when ARPE=0 (TIMx_ARR not preloaded) | 1089 |
| Figure 277. Counter timing diagram, update event when ARPE=1 (TIMx_ARR preloaded) | 1090 |
| Figure 278. Counter timing diagram, internal clock divided by 1 | 1091 |
| Figure 279. Counter timing diagram, internal clock divided by 2 | 1092 |
| Figure 280. Counter timing diagram, internal clock divided by 4 | 1092 |
| Figure 281. Counter timing diagram, internal clock divided by N | 1093 |
| Figure 282. Counter timing diagram, update event when repetition counter is not used | 1093 |
| Figure 283. Counter timing diagram, internal clock divided by 1, TIMx_ARR = 0x6 | 1095 |
| Figure 284. Counter timing diagram, internal clock divided by 2 | 1095 |
| Figure 285. Counter timing diagram, internal clock divided by 4, TIMx_ARR=0x36 | 1096 |
| Figure 286. Counter timing diagram, internal clock divided by N | 1096 |
| Figure 287. Counter timing diagram, update event with ARPE=1 (counter underflow) | 1097 |
| Figure 288. Counter timing diagram, Update event with ARPE=1 (counter overflow) | 1098 |
| Figure 289. Update rate examples depending on mode and TIMx_RCR register settings | 1099 |
| Figure 290. External trigger input block | 1100 |
| Figure 291. Control circuit in normal mode, internal clock divided by 1 | 1101 |
| Figure 292. tim_t2 external clock connection example | 1101 |
| Figure 293. Control circuit in external clock mode 1 | 1102 |
| Figure 294. External trigger input block | 1103 |

| | |
|--|------|
| Figure 295. Control circuit in external clock mode 2 | 1104 |
| Figure 296. Capture/compare channel (example: channel 1 input stage) | 1104 |
| Figure 297. Capture/compare channel 1 main circuit | 1105 |
| Figure 298. Output stage of capture/compare channel (channel 1, idem ch. 2, 3 and 4) | 1106 |
| Figure 299. Output stage of capture/compare channel (channel 5, idem ch. 6) | 1106 |
| Figure 300. PWM input mode timing | 1109 |
| Figure 301. Output compare mode, toggle on tim_oc1 | 1111 |
| Figure 302. Edge-aligned PWM waveforms (ARR=8) | 1112 |
| Figure 303. Center-aligned PWM waveforms (ARR=8) | 1113 |
| Figure 304. Dithering principle | 1114 |
| Figure 305. Data format and register coding in dithering mode | 1115 |
| Figure 306. PWM resolution vs frequency | 1116 |
| Figure 307. PWM dithering pattern | 1117 |
| Figure 308. Dithering effect on duty cycle in center-aligned PWM mode | 1118 |
| Figure 309. Generation of 2 phase-shifted PWM signals with 50% duty cycle | 1120 |
| Figure 310. Combined PWM mode on channel 1 and 3 | 1121 |
| Figure 311. 3-phase combined PWM signals with multiple trigger pulses per period | 1122 |
| Figure 312. Complementary output with symmetrical dead-time insertion | 1123 |
| Figure 313. Asymmetrical deadtime | 1124 |
| Figure 314. Dead-time waveforms with delay greater than the negative pulse | 1124 |
| Figure 315. Dead-time waveforms with delay greater than the positive pulse | 1124 |
| Figure 316. Break and Break2 circuitry overview | 1127 |
| Figure 317. Various output behavior in response to a break event on tim_brk (OSSI = 1) | 1129 |
| Figure 318. PWM output state following tim_brk and tim_brk2 assertion (OSSI=1) | 1130 |
| Figure 319. PWM output state following tim_brk assertion (OSSI=0) | 1131 |
| Figure 320. Output redirection (tim_brk2 request not represented) | 1132 |
| Figure 321. tim_ocref_clr input selection multiplexer | 1133 |
| Figure 322. Clearing TIMx tim_ocxref | 1134 |
| Figure 323. 6-step generation, COM example (OSSR=1) | 1135 |
| Figure 324. Example of one pulse mode | 1136 |
| Figure 325. Retriggerable one-pulse mode | 1137 |
| Figure 326. Pulse generator circuitry | 1138 |
| Figure 327. Pulse generation on compare event, for edge-aligned and encoder modes | 1139 |
| Figure 328. Extended pulselength in case of concurrent triggers | 1140 |
| Figure 329. Example of counter operation in encoder interface mode | 1142 |
| Figure 330. Example of encoder interface mode with tim_t1fp1 polarity inverted | 1142 |
| Figure 331. Quadrature encoder counting modes | 1143 |
| Figure 332. Direction plus clock encoder mode | 1144 |
| Figure 333. Directional clock encoder mode (CC1P = CC2P = 0) | 1144 |
| Figure 334. Directional clock encoder mode (CC1P = CC2P = 1) | 1145 |
| Figure 335. Index gating options | 1146 |
| Figure 336. Jittered Index signals | 1146 |
| Figure 337. Index generation for IPOS[1:0] = 11 | 1147 |
| Figure 338. Counter reading with index gated on channel A (IPOS[1:0] = 11) | 1148 |
| Figure 339. Counter reading with index ungated (IPOS[1:0] = 00) | 1148 |
| Figure 340. Counter reading with index gated on channel A and B | 1149 |
| Figure 341. Encoder mode behavior in case of narrow index pulse (IPOS[1:0] = 11) | 1150 |
| Figure 342. Counter reset Narrow index pulse (closer view, ARR = 0x07) | 1151 |
| Figure 343. Index behavior in x1 and x2 mode (IPOS[1:0] = 01) | 1152 |
| Figure 344. Directional index sensitivity | 1153 |
| Figure 345. Counter reset as function of FIDX bit setting | 1153 |
| Figure 346. Index behavior in clock + direction mode, IPOS[0] = 1 | 1154 |

| | |
|---|------|
| Figure 347. Index behavior in directional clock mode, IPOS[0] = 1 | 1154 |
| Figure 348. State diagram for quadrature encoded signals | 1155 |
| Figure 349. Up-counting encoder error detection | 1156 |
| Figure 350. Down-counting encode error detection | 1157 |
| Figure 351. Encoder mode change with preload transferred on update (SMSPS = 0) | 1158 |
| Figure 352. Measuring time interval between edges on 3 signals | 1159 |
| Figure 353. Example of Hall sensor interface | 1161 |
| Figure 354. Control circuit in reset mode | 1162 |
| Figure 355. Control circuit in Gated mode | 1163 |
| Figure 356. Control circuit in trigger mode | 1164 |
| Figure 357. Control circuit in external clock mode 2 + trigger mode | 1165 |
| Figure 358. General-purpose timer block diagram | 1223 |
| Figure 359. Counter timing diagram with prescaler division change from 1 to 2 | 1229 |
| Figure 360. Counter timing diagram with prescaler division change from 1 to 4 | 1230 |
| Figure 361. Counter timing diagram, internal clock divided by 1 | 1231 |
| Figure 362. Counter timing diagram, internal clock divided by 2 | 1231 |
| Figure 363. Counter timing diagram, internal clock divided by 4 | 1232 |
| Figure 364. Counter timing diagram, internal clock divided by N | 1232 |
| Figure 365. Counter timing diagram, Update event when ARPE=0 (TIMx_ARR not preloaded) | 1233 |
| Figure 366. Counter timing diagram, Update event when ARPE=1 (TIMx_ARR preloaded) | 1234 |
| Figure 367. Counter timing diagram, internal clock divided by 1 | 1235 |
| Figure 368. Counter timing diagram, internal clock divided by 2 | 1236 |
| Figure 369. Counter timing diagram, internal clock divided by 4 | 1236 |
| Figure 370. Counter timing diagram, internal clock divided by N | 1237 |
| Figure 371. Counter timing diagram, Update event | 1237 |
| Figure 372. Counter timing diagram, internal clock divided by 1, TIMx_ARR=0x8 | 1239 |
| Figure 373. Counter timing diagram, internal clock divided by 2 | 1239 |
| Figure 374. Counter timing diagram, internal clock divided by 4, TIMx_ARR=0x38 | 1240 |
| Figure 375. Counter timing diagram, internal clock divided by N | 1240 |
| Figure 376. Counter timing diagram, Update event with ARPE=1 (counter underflow) | 1241 |
| Figure 377. Counter timing diagram, Update event with ARPE=1 (counter overflow) | 1242 |
| Figure 378. Control circuit in normal mode, internal clock divided by 1 | 1243 |
| Figure 379. tim_t12 external clock connection example | 1243 |
| Figure 380. Control circuit in external clock mode 1 | 1244 |
| Figure 381. External trigger input block | 1245 |
| Figure 382. Control circuit in external clock mode 2 | 1246 |
| Figure 383. Capture/compare channel (example: channel 1 input stage) | 1246 |
| Figure 384. Capture/compare channel 1 main circuit | 1247 |
| Figure 385. Output stage of capture/compare channel (channel 1, idem ch.2, 3 and 4) | 1247 |
| Figure 386. PWM input mode timing | 1250 |
| Figure 387. Output compare mode, toggle on tim_oc1 | 1252 |
| Figure 388. Edge-aligned PWM waveforms (ARR=8) | 1253 |
| Figure 389. Center-aligned PWM waveforms (ARR=8) | 1254 |
| Figure 390. Dithering principle | 1255 |
| Figure 391. Data format and register coding in dithering mode | 1256 |
| Figure 392. PWM resolution vs frequency (16-bit mode) | 1257 |
| Figure 393. PWM resolution vs frequency (32-bit mode) | 1257 |
| Figure 394. PWM dithering pattern | 1258 |
| Figure 395. Dithering effect on duty cycle in center-aligned PWM mode | 1259 |
| Figure 396. Generation of 2 phase-shifted PWM signals with 50% duty cycle | 1261 |
| Figure 397. Combined PWM mode on channels 1 and 3 | 1262 |
| Figure 398. OCREF_CLR input selection multiplexer | 1263 |

| | |
|--|------|
| Figure 399. Clearing TIMx tim_ocxref | 1263 |
| Figure 400. Example of One-pulse mode | 1264 |
| Figure 401. Reriggerable one-pulse mode | 1266 |
| Figure 402. Pulse generator circuitry | 1266 |
| Figure 403. Pulse generation on compare event, for edge-aligned and encoder modes | 1267 |
| Figure 404. Extended pulse width in case of concurrent triggers | 1268 |
| Figure 405. Example of counter operation in encoder interface mode | 1270 |
| Figure 406. Example of encoder interface mode with tim_ti1fp1 polarity inverted | 1270 |
| Figure 407. Quadrature encoder counting modes | 1271 |
| Figure 408. Direction plus clock encoder mode | 1272 |
| Figure 409. Directional clock encoder mode (CC1P = CC2P = 0) | 1273 |
| Figure 410. Directional clock encoder mode (CC1P = CC2P = 1) | 1273 |
| Figure 411. Index gating options | 1275 |
| Figure 412. Jittered Index signals | 1275 |
| Figure 413. Index generation for IPOS[1:0] = 11 | 1276 |
| Figure 414. Counter reading with index gated on channel A (IPOS[1:0] = 11) | 1276 |
| Figure 415. Counter reading with index ungated (IPOS[1:0] = 00) | 1277 |
| Figure 416. Counter reading with index gated on channel A and B | 1277 |
| Figure 417. Encoder mode behavior in case of narrow index pulse (IPOS[1:0] = 11) | 1278 |
| Figure 418. Counter reset Narrow index pulse (closer view, ARR = 0x07) | 1279 |
| Figure 419. Index behavior in x1 and x2 mode (IPOS[1:0] = 01) | 1280 |
| Figure 420. Directional index sensitivity | 1281 |
| Figure 421. Counter reset as function of FIDX bit setting | 1281 |
| Figure 422. Index behavior in clock + direction mode, IPOS[0] = 1 | 1282 |
| Figure 423. Index behavior in directional clock mode, IPOS[0] = 1 | 1282 |
| Figure 424. State diagram for quadrature encoded signals | 1283 |
| Figure 425. Up-counting encoder error detection | 1284 |
| Figure 426. Down-counting encode error detection | 1285 |
| Figure 427. Encoder mode change with preload transferred on update (SMSPS = 0) | 1286 |
| Figure 428. Control circuit in reset mode | 1288 |
| Figure 429. Control circuit in gated mode | 1289 |
| Figure 430. Control circuit in trigger mode | 1289 |
| Figure 431. Control circuit in external clock mode 2 + trigger mode | 1291 |
| Figure 432. Master/Slave timer example | 1291 |
| Figure 433. Master/slave connection example with 1 channel only timers | 1292 |
| Figure 434. Gating TIM_slv with tim_oc1ref of TIM_mstr | 1293 |
| Figure 435. Gating TIM_slv with Enable of TIM_mstr | 1294 |
| Figure 436. Triggering TIM_slv with update of TIM_mstr | 1295 |
| Figure 437. Triggering TIM_slv with Enable of TIM_mstr | 1295 |
| Figure 438. Triggering TIM_mstr and TIM_slv with TIM_mstr tim_ti1 input | 1296 |
| Figure 439. TIM15 block diagram | 1340 |
| Figure 440. TIM16/TIM17 block diagram | 1341 |
| Figure 441. Counter timing diagram with prescaler division change from 1 to 2 | 1346 |
| Figure 442. Counter timing diagram with prescaler division change from 1 to 4 | 1346 |
| Figure 443. Counter timing diagram, internal clock divided by 1 | 1348 |
| Figure 444. Counter timing diagram, internal clock divided by 2 | 1348 |
| Figure 445. Counter timing diagram, internal clock divided by 4 | 1349 |
| Figure 446. Counter timing diagram, internal clock divided by N | 1349 |
| Figure 447. Counter timing diagram, update event when ARPE=0 (TIMx_ARR not preloaded) | 1350 |
| Figure 448. Counter timing diagram, update event when ARPE=1 (TIMx_ARR preloaded) | 1351 |

| | |
|---|------|
| Figure 449. Update rate examples depending on mode and TIMx_RCR register settings | 1352 |
| Figure 450. Control circuit in normal mode, internal clock divided by 1 | 1353 |
| Figure 451. tim_t1 external clock connection example | 1353 |
| Figure 452. Control circuit in external clock mode 1 | 1354 |
| Figure 453. Capture/compare channel (example: channel 1 input stage) | 1355 |
| Figure 454. Capture/compare channel 1 main circuit | 1355 |
| Figure 455. Output stage of capture/compare channel (channel 1) | 1356 |
| Figure 456. Output stage of capture/compare channel (channel 2 for TIM15) | 1356 |
| Figure 457. PWM input mode timing | 1359 |
| Figure 458. Output compare mode, toggle on tim_oc1 | 1361 |
| Figure 459. Edge-aligned PWM waveforms (ARR=8) | 1362 |
| Figure 460. Dithering principle | 1363 |
| Figure 461. Data format and register coding in dithering mode | 1363 |
| Figure 462. PWM resolution vs frequency | 1364 |
| Figure 463. PWM dithering pattern | 1365 |
| Figure 464. Combined PWM mode on channel 1 and 2 | 1367 |
| Figure 465. Complementary output with symmetrical dead-time insertion | 1368 |
| Figure 466. Asymmetrical deadtime | 1369 |
| Figure 467. Dead-time waveforms with delay greater than the negative pulse | 1369 |
| Figure 468. Dead-time waveforms with delay greater than the positive pulse | 1369 |
| Figure 469. Break circuitry overview | 1371 |
| Figure 470. Output behavior in response to a break event on tim_brk | 1373 |
| Figure 471. Output redirection | 1375 |
| Figure 472. tim_ocref_clr input selection multiplexer | 1375 |
| Figure 473. 6-step generation, COM example (OSSR=1) | 1376 |
| Figure 474. Example of one pulse mode | 1377 |
| Figure 475. Retriggerable one pulse mode | 1379 |
| Figure 476. Measuring time interval between edges on 2 signals | 1379 |
| Figure 477. Control circuit in reset mode | 1380 |
| Figure 478. Control circuit in gated mode | 1381 |
| Figure 479. Control circuit in trigger mode | 1382 |
| Figure 480. Basic timer block diagram | 1443 |
| Figure 481. Control circuit in normal mode, internal clock divided by 1 | 1444 |
| Figure 482. Counter timing diagram with prescaler division change from 1 to 2 | 1445 |
| Figure 483. Counter timing diagram with prescaler division change from 1 to 4 | 1446 |
| Figure 484. Counter timing diagram, internal clock divided by 1 | 1447 |
| Figure 485. Counter timing diagram, internal clock divided by 2 | 1447 |
| Figure 486. Counter timing diagram, internal clock divided by 4 | 1448 |
| Figure 487. Counter timing diagram, internal clock divided by N | 1448 |
| Figure 488. Counter timing diagram, update event when ARPE = 0 (TIMx_ARR not preloaded) | 1449 |
| Figure 489. Counter timing diagram, update event when ARPE=1 (TIMx_ARR preloaded) | 1450 |
| Figure 490. Dithering principle | 1451 |
| Figure 491. Data format and register coding in dithering mode | 1451 |
| Figure 492. FCnt resolution vs frequency | 1452 |
| Figure 493. PWM dithering pattern | 1452 |
| Figure 494. Low-power timer block diagram | 1462 |
| Figure 495. Glitch filter timing diagram | 1465 |
| Figure 496. LPTIM output waveform, single counting mode configuration | 1466 |
| Figure 497. LPTIM output waveform, Single counting mode configuration and Set-once mode activated (WAVE bit is set) | 1467 |

| | |
|---|------|
| Figure 498. LPTIM output waveform, Continuous counting mode configuration | 1467 |
| Figure 499. Waveform generation | 1469 |
| Figure 500. Encoder mode counting sequence | 1473 |
| Figure 501. IRTIM internal hardware connections with TIM16 and TIM17 | 1486 |
| Figure 502. AES block diagram | 1488 |
| Figure 503. ECB encryption and decryption principle | 1490 |
| Figure 504. CBC encryption and decryption principle | 1491 |
| Figure 505. CTR encryption and decryption principle | 1492 |
| Figure 506. GCM encryption and authentication principle | 1493 |
| Figure 507. GMAC authentication principle | 1493 |
| Figure 508. CCM encryption and authentication principle | 1494 |
| Figure 509. Encryption key derivation for EC B/CBC decryption (Mode 2) | 1497 |
| Figure 510. Example of suspend mode management | 1498 |
| Figure 511. EC B encryption | 1499 |
| Figure 512. EC B decryption | 1499 |
| Figure 513. CBC encryption | 1500 |
| Figure 514. CBC decryption | 1500 |
| Figure 515. EC B/CBC encryption (Mode 1) | 1501 |
| Figure 516. EC B/CBC decryption (Mode 3) | 1502 |
| Figure 517. Message construction in CTR mode | 1504 |
| Figure 518. CTR encryption | 1505 |
| Figure 519. CTR decryption | 1505 |
| Figure 520. Message construction in GCM | 1507 |
| Figure 521. GCM authenticated encryption | 1508 |
| Figure 522. Message construction in GMAC mode | 1512 |
| Figure 523. GMAC authentication mode | 1512 |
| Figure 524. Message construction in CCM mode | 1513 |
| Figure 525. CCM mode authenticated encryption | 1515 |
| Figure 526. 128-bit block construction with respect to data swap | 1519 |
| Figure 527. DMA transfer of a 128-bit data block during input phase | 1521 |
| Figure 528. DMA transfer of a 128-bit data block during output phase | 1522 |
| Figure 529. RTC block diagram | 1537 |
| Figure 530. TAMP block diagram | 1578 |
| Figure 531. USART block diagram | 1593 |
| Figure 532. Word length programming | 1596 |
| Figure 533. Configurable stop bits | 1598 |
| Figure 534. TC/TXE behavior when transmitting | 1601 |
| Figure 535. Start bit detection when oversampling by 16 or 8 | 1602 |
| Figure 536. usart_ker_ck clock divider block diagram | 1605 |
| Figure 537. Data sampling when oversampling by 16 | 1606 |
| Figure 538. Data sampling when oversampling by 8 | 1607 |
| Figure 539. Mute mode using Idle line detection | 1614 |
| Figure 540. Mute mode using address mark detection | 1615 |
| Figure 541. Break detection in LIN mode (11-bit break length - LBDL bit is set) | 1618 |
| Figure 542. Break detection in LIN mode vs. Framing error detection | 1619 |
| Figure 543. USART example of synchronous master transmission | 1620 |
| Figure 544. USART data clock timing diagram in synchronous master mode (M bits = 00) | 1620 |
| Figure 545. USART data clock timing diagram in synchronous master mode (M bits = 01) | 1621 |
| Figure 546. USART data clock timing diagram in synchronous slave mode (M bits = 00) | 1622 |

| | |
|---|------|
| Figure 547. ISO 7816-3 asynchronous protocol | 1624 |
| Figure 548. Parity error detection using the 1.5 stop bits | 1626 |
| Figure 549. IrDA SIR ENDEC block diagram | 1630 |
| Figure 550. IrDA data modulation (3/16) - Normal mode | 1630 |
| Figure 551. Transmission using DMA | 1632 |
| Figure 552. Reception using DMA | 1633 |
| Figure 553. Hardware flow control between 2 USARTs | 1633 |
| Figure 554. RS232 RTS flow control | 1634 |
| Figure 555. RS232 CTS flow control | 1635 |
| Figure 556. Wakeup event verified (wake up event = address match, FIFO disabled) | 1638 |
| Figure 557. Wakeup event not verified (wake up event = address match, FIFO disabled) | 1638 |
| Figure 558. LPUART block diagram | 1680 |
| Figure 559. LPUART word length programming | 1682 |
| Figure 560. Configurable stop bits | 1684 |
| Figure 561. TC/TXE behavior when transmitting | 1686 |
| Figure 562. lpuart_ker_ck clock divider block diagram | 1689 |
| Figure 563. Mute mode using Idle line detection | 1693 |
| Figure 564. Mute mode using address mark detection | 1694 |
| Figure 565. Transmission using DMA | 1696 |
| Figure 566. Reception using DMA | 1697 |
| Figure 567. Hardware flow control between 2 LPUARTs | 1698 |
| Figure 568. RS232 RTS flow control | 1698 |
| Figure 569. RS232 CTS flow control | 1699 |
| Figure 570. Wakeup event verified (wake up event = address match, FIFO disabled) | 1702 |
| Figure 571. Wakeup event not verified (wake up event = address match, FIFO disabled) | 1702 |
| Figure 572. SPI block diagram | 1732 |
| Figure 573. Full-duplex single master/single slave application | 1733 |
| Figure 574. Half-duplex single master/ single slave application | 1734 |
| Figure 575. Simplex single master/single slave application (master in transmit-only/slave in receive-only mode) | 1735 |
| Figure 576. Master and three independent slaves | 1736 |
| Figure 577. Multi-master application | 1737 |
| Figure 578. Hardware/software slave select management | 1738 |
| Figure 579. Data clock timing diagram | 1739 |
| Figure 580. Data alignment when data length is not equal to 8-bit or 16-bit | 1740 |
| Figure 581. Packing data in FIFO for transmission and reception | 1744 |
| Figure 582. Master full-duplex communication | 1747 |
| Figure 583. Slave full-duplex communication | 1748 |
| Figure 584. Master full-duplex communication with CRC | 1749 |
| Figure 585. Master full-duplex communication in packed mode | 1750 |
| Figure 586. NSSP pulse generation in Motorola SPI master mode | 1753 |
| Figure 587. TI mode transfer | 1754 |
| Figure 588. I ² S block diagram | 1757 |
| Figure 589. I ² S Philips protocol waveforms (16/32-bit full accuracy) | 1759 |
| Figure 590. I ² S Philips standard waveforms (24-bit frame) | 1759 |
| Figure 591. Transmitting 0x8EAA33 | 1760 |
| Figure 592. Receiving 0x8EAA33 | 1760 |
| Figure 593. I ² S Philips standard (16-bit extended to 32-bit packet frame) | 1760 |
| Figure 594. Example of 16-bit data frame extended to 32-bit channel frame | 1760 |

| | |
|--|------|
| Figure 595. MSB Justified 16-bit or 32-bit full-accuracy length | 1761 |
| Figure 596. MSB justified 24-bit frame length | 1761 |
| Figure 597. MSB justified 16-bit extended to 32-bit packet frame | 1762 |
| Figure 598. LSB justified 16-bit or 32-bit full-accuracy | 1762 |
| Figure 599. LSB justified 24-bit frame length | 1762 |
| Figure 600. Operations required to transmit 0x3478AE | 1763 |
| Figure 601. Operations required to receive 0x3478AE | 1763 |
| Figure 602. LSB justified 16-bit extended to 32-bit packet frame | 1763 |
| Figure 603. Example of 16-bit data frame extended to 32-bit channel frame | 1764 |
| Figure 604. PCM standard waveforms (16-bit) | 1764 |
| Figure 605. PCM standard waveforms (16-bit extended to 32-bit packet frame) | 1765 |
| Figure 606. Start sequence in master mode | 1766 |
| Figure 607. Audio sampling frequency definition | 1767 |
| Figure 608. I ² S clock generator architecture | 1767 |
| Figure 609. SAI functional block diagram | 1790 |
| Figure 610. Audio frame | 1793 |
| Figure 611. FS role is start of frame + channel side identification (FSDEF = TRIS = 1) | 1795 |
| Figure 612. FS role is start of frame (FSDEF = 0) | 1796 |
| Figure 613. Slot size configuration with FBOFF = 0 in SAI_xSLOTR | 1797 |
| Figure 614. First bit offset | 1797 |
| Figure 615. Audio block clock generator overview | 1799 |
| Figure 616. PDM typical connection and timing | 1803 |
| Figure 617. Detailed PDM interface block diagram | 1804 |
| Figure 618. Start-up sequence | 1805 |
| Figure 619. SAI_ADR format in TDM, 32-bit slot width | 1806 |
| Figure 620. SAI_ADR format in TDM, 16-bit slot width | 1807 |
| Figure 621. SAI_ADR format in TDM, 8-bit slot width | 1808 |
| Figure 622. AC'97 audio frame | 1811 |
| Figure 623. SPDIF format | 1812 |
| Figure 624. SAI_xDR register ordering | 1813 |
| Figure 625. Data companding hardware in an audio block in the SAI | 1817 |
| Figure 626. Tristate strategy on SD output line on an inactive slot | 1818 |
| Figure 627. Tristate on output data line in a protocol like I2S | 1819 |
| Figure 628. Overrun detection error | 1820 |
| Figure 629. FIFO underrun event | 1820 |
| Figure 630. I2C block diagram | 1857 |
| Figure 631. I2C bus protocol | 1859 |
| Figure 632. Setup and hold timings | 1861 |
| Figure 633. I2C initialization flow | 1864 |
| Figure 634. Data reception | 1865 |
| Figure 635. Data transmission | 1866 |
| Figure 636. Slave initialization flow | 1869 |
| Figure 637. Transfer sequence flow for I2C slave transmitter, NOSTRETCH = 0 | 1871 |
| Figure 638. Transfer sequence flow for I2C slave transmitter, NOSTRETCH = 1 | 1872 |
| Figure 639. Transfer bus diagrams for I2C slave transmitter | 1873 |
| Figure 640. Transfer sequence flow for slave receiver with NOSTRETCH = 0 | 1874 |
| Figure 641. Transfer sequence flow for slave receiver with NOSTRETCH = 1 | 1875 |
| Figure 642. Transfer bus diagrams for I2C slave receiver | 1875 |
| Figure 643. Master clock generation | 1877 |
| Figure 644. Master initialization flow | 1879 |
| Figure 645. 10-bit address read access with HEAD10R = 0 | 1879 |
| Figure 646. 10-bit address read access with HEAD10R = 1 | 1880 |

| | |
|--|------|
| Figure 647. Transfer sequence flow for I ² C master transmitter for N≤255 bytes | 1881 |
| Figure 648. Transfer sequence flow for I ² C master transmitter for N>255 bytes | 1882 |
| Figure 649. Transfer bus diagrams for I ² C master transmitter | 1883 |
| Figure 650. Transfer sequence flow for I ² C master receiver for N≤255 bytes | 1885 |
| Figure 651. Transfer sequence flow for I ² C master receiver for N >255 bytes | 1886 |
| Figure 652. Transfer bus diagrams for I ² C master receiver | 1887 |
| Figure 653. Timeout intervals for t _{LOW:SEXT} , t _{LOW:MEXT} | 1891 |
| Figure 654. Transfer sequence flow for SMBus slave transmitter N bytes + PEC | 1895 |
| Figure 655. Transfer bus diagrams for SMBus slave transmitter (SBC=1) | 1896 |
| Figure 656. Transfer sequence flow for SMBus slave receiver N Bytes + PEC | 1897 |
| Figure 657. Bus transfer diagrams for SMBus slave receiver (SBC=1) | 1898 |
| Figure 658. Bus transfer diagrams for SMBus master transmitter | 1899 |
| Figure 659. Bus transfer diagrams for SMBus master receiver | 1901 |
| Figure 660. Independent watchdog block diagram | 1923 |
| Figure 661. Watchdog block diagram | 1933 |
| Figure 662. Window watchdog timing diagram | 1934 |
| Figure 663. CAN subsystem | 1939 |
| Figure 664. FDCAN block diagram | 1941 |
| Figure 665. Bit timing | 1943 |
| Figure 666. Transceiver delay measurement | 1948 |
| Figure 667. Pin control in Bus monitoring mode | 1949 |
| Figure 668. Pin control in Loop Back mode | 1951 |
| Figure 669. Message RAM configuration | 1952 |
| Figure 670. Standard Message ID filter path | 1955 |
| Figure 671. Extended Message ID filter path | 1956 |
| Figure 672. USB peripheral block diagram | 2004 |
| Figure 673. Packet buffer areas with examples of buffer description table locations | 2008 |
| Figure 674. UCPD block diagram | 2038 |
| Figure 675. Clock division and timing elements | 2039 |
| Figure 676. K-code transmission | 2042 |
| Figure 677. Transmit order for various sizes of data | 2043 |
| Figure 678. Packet format | 2044 |
| Figure 679. Line format of Hard Reset | 2044 |
| Figure 680. Line format of Cable Reset | 2045 |
| Figure 681. BIST test data frame | 2046 |
| Figure 682. BIST Carrier Mode 2 frame | 2046 |
| Figure 683. UCPD BMC transmitter architecture | 2047 |
| Figure 684. UCPD BMC receiver architecture | 2048 |
| Figure 685. Block diagram of STM32 MCU and Cortex®-M4 with FPU-level debug support | 2075 |
| Figure 686. SWJ debug port | 2077 |
| Figure 687. JTAG TAP connections | 2081 |
| Figure 688. TPIU block diagram | 2100 |

1 Documentation conventions

1.1 General information

The STM32G4 Series devices have an Arm® Cortex®-M4 with FPU core.



1.2 List of abbreviations for registers

The following abbreviations^(b) are used in register descriptions:

| | |
|---------------------------------|--|
| read/write (rw) | Software can read and write to this bit. |
| read-only (r) | Software can only read this bit. |
| write-only (w) | Software can only write to this bit. Reading this bit returns the reset value. |
| read/clear write0 (rc_w0) | Software can read as well as clear this bit by writing 0. Writing 1 has no effect on the bit value. |
| read/clear write1 (rc_w1) | Software can read as well as clear this bit by writing 1. Writing 0 has no effect on the bit value. |
| read/clear write (rc_w) | Software can read as well as clear this bit by writing to the register. The value written to this bit is not important. |
| read/clear by read (rc_r) | Software can read this bit. Reading this bit automatically clears it to 0. Writing this bit has no effect on the bit value. |
| read/set by read (rs_r) | Software can read this bit. Reading this bit automatically sets it to 1. Writing this bit has no effect on the bit value. |
| read/set (rs) | Software can read as well as set this bit. Writing 0 has no effect on the bit value. |
| read/write once (rwo) | Software can only write once to this bit and can also read it at any time. Only a reset can return the bit to its reset value. |
| toggle (t) | The software can toggle this bit by writing 1. Writing 0 has no effect. |
| read-only write trigger (rt_w1) | Software can read this bit. Writing 1 triggers an event but has no effect on the bit value. |
| Reserved (Res.) | Reserved bit, must be kept at reset value. |

-
- a. Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.
 - b. This is an exhaustive list of all abbreviations applicable to STMicroelectronics microcontrollers, some of them may not be used in the current document.

1.3 Glossary

This section gives a brief definition of acronyms and abbreviations used in this document:

- **Word**: data of 32-bit length.
- **Half-word**: data of 16-bit length.
- **Byte**: data of 8-bit length.
- **IAP (in-application programming)**: IAP is the ability to re-program the Flash memory of a microcontroller while the user program is running.
- **ICP (in-circuit programming)**: ICP is the ability to program the Flash memory of a microcontroller using the JTAG protocol, the SWD protocol or the bootloader while the device is mounted on the user application board.
- **Option bytes**: product configuration bits stored in the Flash memory.
- **AHB**: advanced high-performance bus.

1.4 Product category definition

[Table 1](#) gives an overview of memory density versus product line.

The present reference manual describes the superset of features for each product category. Refer to [Table 2](#) for the list of features per category.

Table 1. STM32G4 Series memory density

| Memory density | Category 2 | Category 3 | Category 4 |
|----------------|------------------------------|---|------------------------------|
| 128 Kbytes | STM32G431 STM32G441 (AES) | STM32G471 STM32G473 STM32G474 | - |
| 256 Kbytes | - | STM32G471 STM32G473 STM32G474 | - |
| 512 Kbytes | - | STM32G471 STM32G473 STM32G474 STM32G483 (AES) STM32G484 (AES) | STM32G491 STM32G4A1 (AES) |

1.5 Availability of peripherals

[Table 2: Product specific features](#) summarizes the list of peripherals available in all the STM32G4xx products considering the biggest package in every product.

For availability of peripherals and their number across all sales types, refer to the particular device datasheet.

Table 2. Product specific features

| Feature | STM32G474/ STM32G484 | STM32G473/ STM32G483 | STM32G471 | STM32G431/ STM32G441 | STM32G491/ STM32G4A1 |
|----------|--|--|---|---|---|
| Flash | 512/256/128K, Dual bank | 512/256/128K, Dual bank | 512/256/128K, Dual bank | 128/64/32K, single bank | 512/256K, single bank |
| SRAM1 | 80K, parity check on the first 32K | 80K, parity check on the first 32K | 80K, parity check on the first 32K | 16K, parity check on the whole SRAM1 | 80K, parity check on the first 32K |
| SRAM2 | 16K, no parity check | 16K, no parity check | 16K, no parity check | 6K, no parity check | 16K, no parity check |
| CCM SRAM | 32K, parity check on the whole CCM SRAM | 32K, parity check on the whole CCM SRAM | 16K, parity check on the whole CCM SRAM | 10K, parity check on the whole CCM SRAM | 16K, parity check on the whole CCM SRAM |
| CRS | Yes | Yes | Yes | Yes | Yes |
| DMA | 2 DMA controllers: DMA1: 8 channels DMA2: 8 channels | 2 DMA controllers: DMA1: 8 channels DMA2: 8 channels | 2 DMA controllers: DMA1: 8 channels DMA2: 8 channels | 2 DMA controllers: DMA1: 6 channels DMA2: 6 channels | 2 DMA controllers: DMA1: 8 channels DMA2: 8 channels |
| DMAMUX | Yes | Yes | Yes | Yes | Yes |
| Cordic | Yes | Yes | Yes | Yes | Yes |
| FMAC | Yes | Yes | Yes | Yes | Yes |
| RNG | Yes | Yes | Yes | Yes | Yes |
| AES | Yes (Note) | Yes (Note) | No | Yes (Note) | Yes (Note) |
| CRC | Yes | Yes | Yes | Yes | Yes |
| FSMC | Yes | Yes | No | No | No |
| QUADSPI | Yes | Yes | Yes | No | Yes |
| ADC | 5 x ADC: ADC1/2 can be used in dual mode ADC3/4 can be used in dual mode ADC5 usable only in single mode | 5 x ADC: ADC 1/2 can be used in dual mode ADC3/4 can be used in dual mode ADC5 usable only in single mode | 3 x ADC: ADC 1/2 can be used in dual mode ADC3 usable only in single mode | 2 x ADC: ADC 1/2 can be used in dual mode | 3 x ADC: ADC 1/2 can be used in dual mode ADC3 usable only in single mode |
| DAC | 7 DAC ch: DAC1_CH1/ DAC1_CH2/ DAC2_CH1: external DAC3_CH1/ DAC3_CH2/ DAC4_CH1/ DAC4_CH2: internal | 7 DAC ch: DAC1_CH1/ DAC1_CH2/ DAC2_CH1: external DAC3_CH1/ DAC3_CH2/ DAC4_CH1/ DAC4_CH2: internal | 4 DAC ch: DAC1_CH1/ DAC1_CH2: external DAC3_CH1/ DAC3_CH2: internal | 4 DAC ch: DAC1_CH1/ DAC1_CH2: external DAC3_CH1/ DAC3_CH2: internal | 4 DAC ch: DAC1_CH1/ DAC1_CH2: external DAC3_CH1/ DAC3_CH2: internal |
| COMP | 7 (COMP1..7) | 7 (COMP1..7) | 4 (COMP1..4) | 4 (COMP1..4) | 4 (COMP1..4) |
| OPAMP | 6 (OPAMP1..6) | 6 (OPAMP1..6) | 4 (OPAMP1..2,3,6) | 3 (OPAMP1..3) | 4 (OPAMP1..2,3,6) |
| VREFBUF | Yes | Yes | Yes | Yes | Yes |

Table 2. Product specific features (continued)

| Feature | STM32G474/ STM32G484 | STM32G473/ STM32G483 | STM32G471 | STM32G431/ STM32G441 | STM32G491/ STM32G4A1 |
|--|---|--|--|--|--|
| HRTIM1 | Yes | No | No | No | No |
| Advanced control timers (TIM1/TIM8 / TIM20) | TIM1/8/20 | TIM1/8/20 | TIM1/8/20 | TIM1/8 | TIM1/8/20 |
| General purpose timers (TIM2/TIM3 / TIM4/TIM5) | TIM2/3/4/5 | TIM2/3/4/5 | TIM2/3/4 | TIM2/3/4 | TIM2/3/4 |
| General purpose timers (TIM15/TIM16/TIM17) | TIM15/16/17 | TIM15/16/17 | TIM15/16/17 | TIM15/16/17 | TIM15/16/17 |
| Basic timers (TIM6/TIM7) | TIM6/7 | TIM6/7 | TIM6/7 | TIM6/7 | TIM6/7 |
| Low power timer (LPTIM1) | Yes | Yes | Yes | Yes | Yes |
| Infrared interface (IRTIM) | Yes | Yes | Yes | Yes | Yes |
| Independent watchdog (WDG) | Yes | Yes | Yes | Yes | Yes |
| System window watchdog (WWDG) | Yes | Yes | Yes | Yes | Yes |
| RTC and TAMP | Yes | Yes | Yes | Yes | Yes |
| I2C | 4 x I2C (I2C1..4) | 4 x I2C (I2C1..4) | 4 x I2C (I2C1..4) | 3 x I2C (I2C1..3) | 3 x I2C (I2C1..3) |
| USART/UART | 3 x USART (USART1..3) 2 x UART (UART4,5) | 3 x USART (USART1..3) 2 x UART(UART4,5) | 3 x USART (USART1..3) 2 x UART(UART4,5) | 3 x USART (USART1..3) 1 x UART(UART4) | 3 x USART (USART1..3) 2 x UART(UART4,5) |
| LPUART | 1 x LPUART | 1 x LPUART | 1 x LPUART | 1 x LPUART | 1 x LPUART |
| SPI/I2S | 4 x SPI/2 x I2S (SPI1..4 - I2S2,3) | 4 x SPI/2 x I2S (SPI1..4 - I2S2,3) | 4 x SPI/2 x I2S (SPI1..4 - I2S2,3) | 3 x SPI/2 x I2S (SPI1..3 - I2S2,3) | 3 x SPI/2 x I2S (SPI1..3 - I2S2,3) |
| SAI | 1 x SAI | 1 x SAI | 1 x SAI | 1 x SAI | 1 x SAI |

Table 2. Product specific features (continued)

| Feature | STM32G474/ STM32G484 | STM32G473/ STM32G483 | STM32G471 | STM32G431/ STM32G441 | STM32G491/ STM32G4A1 |
|------------|--------------------------|--------------------------|-------------------------|-------------------------|-------------------------|
| FDCAN | 3 x FDCAN (FDCAN1..3) | 3 x FDCAN (FDCAN1..3) | 2 x FDCAN (FDCAN1,2) | 1 x FDCAN (FDCAN1) | 2 x FDCAN (FDCAN1,2) |
| USB device | 1 x USB device | 1 x USB device | 1 x USB device | 1 x USB device | 1 x USB device |
| UCPD1 | 1 x UCPD | 1 x UCPD | 1 x UCPD | 1 x UCPD | 1 x UCPD |

Note: The AES is available only on STM32G483xx, STM32G484xx, STM32G441x and STM32G4A1 devices.

2 System and memory overview

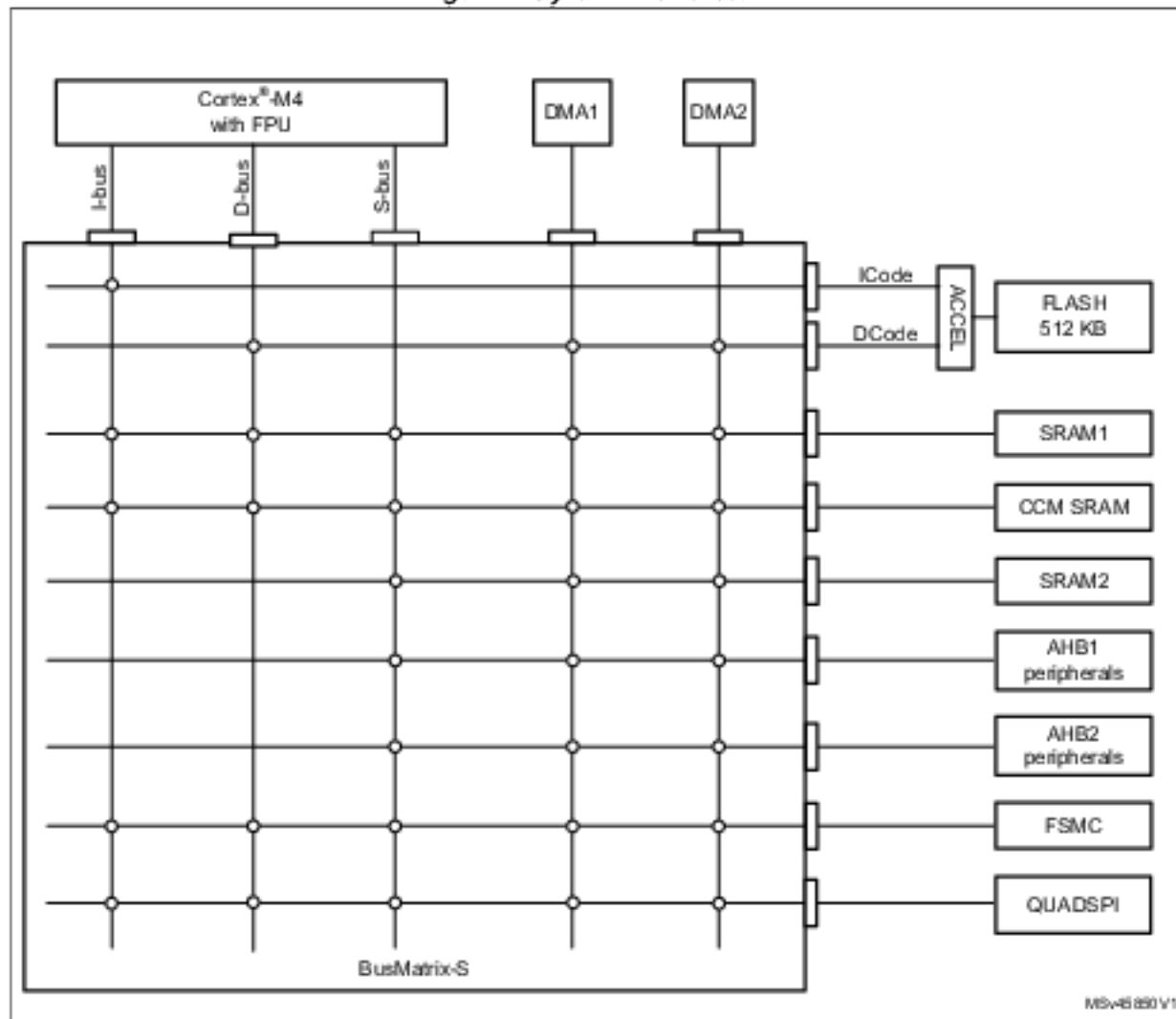
2.1 System architecture

The main system consists of 32-bit multilayer AHB bus matrix that interconnects:

- Up to five masters:
 - Cortex®-M4 with FPU core I-bus
 - Cortex®-M4 with FPU core D-bus
 - Cortex®-M4 with FPU core S-bus
 - DMA1
 - DMA2
- Up to nine slaves:
 - Internal Flash memory on ICode bus
 - Internal Flash memory on DCode bus
 - Internal SRAM1
 - Internal SRAM2
 - Internal CCM SRAM
 - AHB1 peripherals including AHB to APB bridges and APB peripherals (connected to APB1 and APB2)
 - AHB2 peripherals
 - Flexible static memory controller (FSMC)
 - QUAD SPI memory interface (QUADSPI)

The bus matrix provides access from a master to a slave, enabling concurrent access and efficient operation even when several high-speed peripherals work simultaneously. This architecture is shown in [Figure 1](#):

Figure 1. System architecture



2.1.1 I-bus

This bus connects the instruction bus of the Cortex®-M4 with FPU core to the BusMatrix. This bus is used by the core to fetch instructions. The target of this bus is a memory containing code (either internal Flash memory, internal SRAMs or external memories through the FSMC or QUADSPI).

2.1.2 D-bus

This bus connects the data bus of the Cortex®-M4 with FPU core to the BusMatrix. This bus is used by the core for literal load and debug access. The target of this bus is a memory containing code (either internal Flash memory, internal SRAMs or external memories through the FSMC or QUADSPI).

2.1.3 S-bus

This bus connects the system bus of the Cortex®-M4 with FPU core to the BusMatrix. This bus is used by the core to access data located in a peripheral or SRAM area. The targets of this bus are the internal SRAM, the AHB1 peripherals including the APB1 and APB2 peripherals, the AHB2 peripherals and the external memories through the QUADSPI or the FSMC.

The CCM SRAM is also accessible on this bus to allow continuous mapping with SRAM1 and SRAM2.

2.1.4 DMA-bus

This bus connects the AHB master interface of the DMA to the BusMatrix. The targets of this bus are the SRAM1, SRAM2 and CCM SRAM, the AHB1 peripherals including the APB1 and APB2 peripherals, the AHB2 peripherals and the external memories through the QUADSPI or the FSMC.

2.1.5 BusMatrix

The BusMatrix manages the access arbitration between masters. The arbitration uses a Round Robin algorithm. The BusMatrix is composed of up to five masters (CPUAHB, system bus, DCode bus, ICode bus, DMA1, DMA2,) and up to nine slaves (FLASH, SRAM1, SRAM2, CCM SRAM, AHB1 (including APB1 and APB2), AHB2, QUADSPI, and FSMC).

AHB/APB bridges

The two AHB/APB bridges provide full synchronous connections between the AHB and the two APB buses, allowing flexible selection of the peripheral frequency.

Refer to [Section 2.2.2: Memory map and register boundary addresses on page 82](#) for the address mapping of the peripherals connected to this bridge.

After each device reset, all peripheral clocks are disabled (except for the SRAM1/2 and Flash memory interface). Before using a peripheral you have to enable its clock in the RCC_AHBxENR and the RCC_APBxENR registers.

Note: When a 16- or 8-bit access is performed on an APB register, the access is transformed into a 32-bit access: the bridge duplicates the 16- or 8-bit data to feed the 32-bit vector.

2.2 Memory organization

2.2.1 Introduction

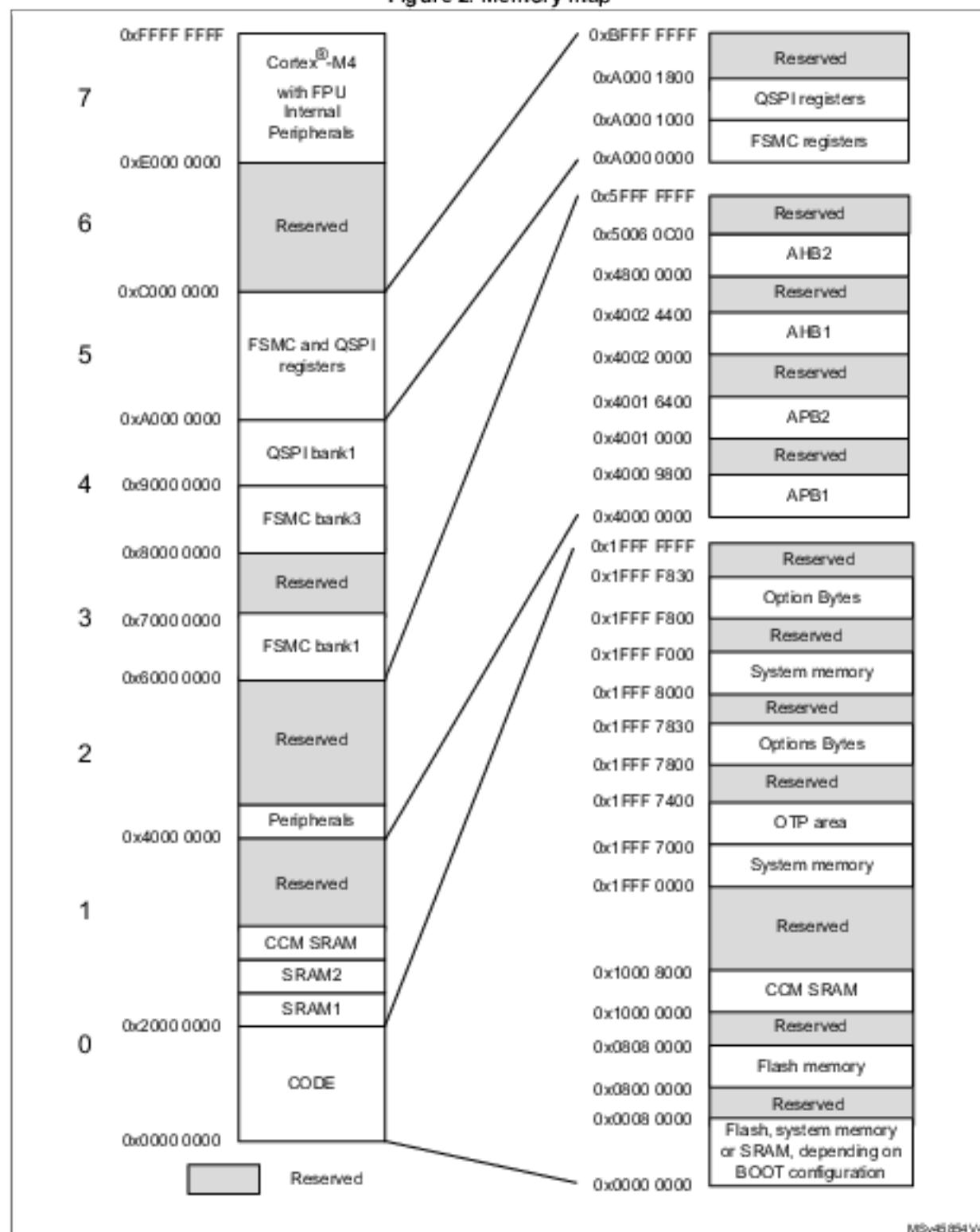
Program memory, data memory, registers and I/O ports are organized within the same linear 4-Gbyte address space.

The bytes are coded in memory in Little Endian format. The lowest numbered byte in a word is considered the word's least significant byte and the highest numbered byte the most significant.

The addressable memory space is divided into eight main blocks, of 512 Mbytes each.

2.2.2 Memory map and register boundary addresses

Figure 2. Memory map



All the memory map areas that are not allocated to on-chip memories and peripherals are considered "Reserved". For the detailed mapping of available memory and register areas, refer to the following table.

The following table gives the boundary addresses of the peripherals available in the devices.

Table 3. STM32G4 Series memory map and peripheral register boundary addresses⁽¹⁾

| Bus | Boundary address | Size (bytes) | Peripheral | Peripheral register map |
|------|---------------------------|--------------|---------------------------|--|
| | 0xA000 1400 - 0xAFFF FFFF | 262 MB | Reserved | - |
| | 0xA000 1000 - 0xA000 13FF | 1 KB | QUADSPI control Registers | Section 20.5.14: QUAD SPI register map |
| | 0xA000 0400 - 0xA000 0FFF | 3 KB | Reserved | - |
| | 0xA000 0000 - 0xA000 03FF | 1 KB | FMC | Section 19.7.8: FMC register map |
| AHB2 | 0x5006 0C00 - 0x5FFF FFFF | 256MB | Reserved | - |
| | 0x5006 0800 - 0x5006 0BFF | 1 KB | RNG | Section 26.7.4: RNG register map |
| | 0x5006 0400 - 0x5006 07FF | 1 KB | Reserved | - |
| | 0x5006 0000 - 0x5006 03FF | 1 KB | AES | Section 34.7.18: AES register map |
| | 0x5000 1800 - 0x5005 FFFF | 377 KB | Reserved | - |
| | 0x5000 1400 - 0x5000 17FF | 1 KB | DAC4 | Section 22.7.24: DAC register map |
| | 0x5000 1000 - 0x5000 13FF | 1 KB | DAC3 | Section 22.7.24: DAC register map |
| | 0x5000 0C00 - 0x5000 0FFF | 1 KB | DAC2 | Section 22.7.24: DAC register map |
| | 0x5000 0800 - 0x5000 0BFF | 1 KB | DAC1 | Section 22.7.24: DAC register map |
| | 0x5000 0400 - 0x5000 07FF | 1 KB | ADC3 - ADC4 - ADC5 | Section 21.9: ADC register map |
| | 0x5000 0000 - 0x5000 03FF | 1 KB | ADC1 - ADC2 | Section 21.9: ADC register map |
| | 0x4800 1C00 - 0x4FFF FFFF | 127 MB | Reserved | - |
| | 0x4800 1800 - 0x4800 1BFF | 1 KB | GPIO G | Section 9.4.12: GPIO register map |
| | 0x4800 1400 - 0x4800 17FF | 1 KB | GPIO F | Section 9.4.12: GPIO register map |
| | 0x4800 1000 - 0x4800 13FF | 1 KB | GPIO E | Section 9.4.12: GPIO register map |
| | 0x4800 0C00 - 0x4800 0FFF | 1 KB | GPIO D | Section 9.4.12: GPIO register map |
| | 0x4800 0800 - 0x4800 0BFF | 1 KB | GPIO C | Section 9.4.12: GPIO register map |
| | 0x4800 0400 - 0x4800 07FF | 1 KB | GPIO B | Section 9.4.12: GPIO register map |
| | 0x4800 0000 - 0x4800 03FF | 1 KB | GPIO A | Section 9.4.12: GPIO register map |

Table 3. STM32G4 Series memory map and peripheral register boundary addresses⁽¹⁾ (continued)

| Bus | Boundary address | Size (bytes) | Peripheral | Peripheral register map |
|------|---------------------------|--------------|-----------------|---|
| AHB1 | 0x4002 3400 - 0x47FF FFFF | 127 MB | Reserved | - |
| | 0x4002 3000 - 0x4002 33FF | 1 KB | CRC | Section 16.4.6: CRC register map |
| | 0x4002 2400 - 0x4002 2FFF | 3 KB | Reserved | - |
| | 0x4002 2000 - 0x4002 23FF | 1 KB | Flash Interface | Section 5.7.14: FLASH register map |
| | 0x4002 1400 - 0x4002 1FFF | 3 KB | FMAC | Section 18.4.9: FMAC register map |
| | 0x4002 1000 - 0x4002 13FF | 1 KB | RCC | Section 7.4.31: RCC register map |
| | 0x4002 0C00 - 0x4002 0FFF | 1 KB | CORDIC | Section 17.4.4: CORDIC register map |
| | 0x4002 0800 - 0x4002 0BFF | 1 KB | DMAMUX | Section 13.6.7: DMAMUX register map |
| | 0x4002 0400 - 0x4002 07FF | 1 KB | DMA 2 | Section 12.6.7: DMA register map |
| | 0x4002 0000 - 0x4002 03FF | 1 KB | DMA 1 | Section 12.6.7: DMA register map |
| APB2 | 0x4001 7800 - 0x4001 FFFF | 2 KB | Reserved | - |
| | 0x4001 6800 - 0x4001 77FF | 3 KB | HRTIM | Section 27.5.82: HRTIM register map |
| | 0x4001 5800 - 0x4001 67FF | 4 KB | Reserved | - |
| | 0x4001 5400 - 0x4001 57FF | 1 KB | SAI1 | Section 40.6.19: SAI register map |
| | 0x4001 5000 - 0x4001 53FF | 1 KB | TIM20 | Section 28.6.31: TIMx register map |
| | 0x4001 4C00 - 0x4001 4FFF | 1 KB | Reserved | - |
| | 0x4001 4800 - 0x4001 4BFF | 1 KB | TIM17 | Section 30.8.22: TIM16/TIM17 register map |
| | 0x4001 4400 - 0x4001 47FF | 1 KB | TIM16 | Section 30.8.22: TIM16/TIM17 register map |
| | 0x4001 4000 - 0x4001 43FF | 1 KB | TIM15 | Section 30.7.23: TIM15 register map |
| | 0x4001 3C00 - 0x4001 3FFF | 1 KB | SPI4 | Section 39.9.10: SPI/I2S register map |
| | 0x4001 3800 - 0x4001 3BFF | 1 KB | USART1 | Section 37.8.15: USART register map |
| | 0x4001 3400 - 0x4001 37FF | 1 KB | TIM8 | Section 28.6.31: TIMx register map |
| | 0x4001 3000 - 0x4001 33FF | 1 KB | SPI1 | Section 39.9.10: SPI/I2S register map |
| | 0x4001 2C00 - 0x4001 2FFF | 1 KB | TIM1 | Section 28.6.31: TIMx register map |
| | 0x4001 0800 - 0x4001 2BFF | 9 KB | Reserved | - |
| | 0x4001 0400 - 0x4001 07FF | 1 KB | EXTI | Section 15.5.13: EXTI register map |
| | 0x4001 0300 - 0x4001 03FF | 1 KB | OPAMP | Section 25.5.13: OPAMP register map |
| | 0x4001 0200 - 0x4001 02FF | | COMP | Section 24.6.2: COMP register map |
| | 0x4001 0030 - 0x4001 01FF | | VREFBUF | Section 23.4.3: VREFBUF register map |
| | 0x4001 0000 - 0x4001 0029 | | SYSCFG | Section 10.2.11: SYSCFG register map |

Table 3. STM32G4 Series memory map and peripheral register boundary addresses⁽¹⁾ (continued)

| Bus | Boundary address | Size (bytes) | Peripheral | Peripheral register map |
|------|---------------------------|--------------|-----------------------|--|
| APB1 | 0x4000 AFFE - 0x4000 FFFF | 23 KB | Reserved | - |
| | 0x4000 AC00 - 0x4000 AFFF | 1 KB | FDCANs Message RAM | Section 44.4.38: FD CAN register map |
| | 0x4000 A800 - 0x4000 ABFF | 1 KB | | |
| | 0x4000 A400 - 0x4000 A7FF | 1 KB | | |
| | 0x4000 A000 - 0x4000 A3FF | 1 KB | UCPD1 | Section 46.7.15: UCPD register map |
| | 0x4000 8800 - 0x4000 9FFF | 6 KB | Reserved | - |
| | 0x4000 8400 - 0x4000 87FF | 1 KB | I2C4 | Section 41.7.12: I2C register map |
| | 0x4000 8000 - 0x4000 83FF | 1 KB | LPUART1 | Section 38.7.13: LPUART register map |
| | 0x4000 7C00 - 0x4000 7FFF | 1 KB | LPTIM1 | Section 32.7.10: LPTIM register map |
| | 0x4000 7800 - 0x4000 7BFF | 1 KB | I2C3 | Section 41.7.12: I2C register map |
| | 0x4000 7400 - 0x4000 77FF | 1 KB | Reserved | - |
| | 0x4000 7000 - 0x4000 73FF | 1 KB | PWR | Section 6.4.23: PWR register map and reset value table |
| | 0x4000 6C00 - 0x4000 6FFF | 1 KB | FDCAN3 | Section 44.4.38: FD CAN register map |
| | 0x4000 6800 - 0x4000 6BFF | 1 KB | FDCAN2 | Section 44.4.38: FD CAN register map |
| | 0x4000 6400 - 0x4000 67FF | 1 KB | FDCAN1 | Section 44.4.38: FD CAN register map |
| | 0x4000 6000 - 0x4000 63FF | 1 KB | USB SRAM 1 Kbyte | - |
| | 0x4000 5C00 - 0x4000 5FFF | 1 KB | USB device FS | Section 45.6.3: USB register map |
| | 0x4000 5800 - 0x4000 5BFF | 1 KB | I2C2 | Section 41.7.12: I2C register map |
| | 0x4000 5400 - 0x4000 57FF | 1 KB | I2C1 | Section 41.7.12: I2C register map |
| | 0x4000 5000 - 0x4000 53FF | 1 KB | UART5 | Section 37.8.15: USART register map |
| | 0x4000 4C00 - 0x4000 4FFF | 1 KB | UART4 | Section 37.8.15: USART register map |
| | 0x4000 4800 - 0x4000 4BFF | 1 KB | USART3 | Section 37.8.15: USART register map |
| | 0x4000 4400 - 0x4000 47FF | 1 KB | USART2 | Section 37.8.15: USART register map |

Table 3. STM32G4 Series memory map and peripheral register boundary addresses⁽¹⁾ (continued)

| Bus | Boundary address | Size (bytes) | Peripheral | Peripheral register map |
|---------------|---------------------------|--------------|---------------------|--|
| APB1 Cont. | 0x4000 4000 - 0x4000 43FF | 1 KB | Reserved | - |
| | 0x4000 3C00 - 0x4000 3FFF | 1 KB | SPI3/I2S3 | Section 39.9.10: SPI/I2S register map |
| | 0x4000 3800 - 0x4000 3BFF | 1 KB | SPI2/I2S2 | Section 39.9.10: SPI/I2S register map |
| | 0x4000 3400 - 0x4000 37FF | 1 KB | Reserved | - |
| | 0x4000 3000 - 0x4000 33FF | 1 KB | WDG | Section 42.4.6: WDG register map |
| | 0x4000 2C00 - 0x4000 2FFF | 1 KB | WWDG | Section 43.5.4: WWWDG register map |
| | 0x4000 2800 - 0x4000 2BFF | 1 KB | RTC & BKP Registers | Section 35.6.21: RTC register map |
| | 0x4000 2400 - 0x4000 27FF | 1 KB | TAMP | Section 36.6.9: TAMP register map |
| | 0x4000 2000 - 0x4000 23FF | 1 KB | CRS | Section 8.7.5: CRS register map |
| | 0x4000 1C00 - 0x4000 1FFF | 1 KB | Reserved | - |
| | 0x4000 1800 - 0x4000 1BFF | 1 KB | Reserved | - |
| | 0x4000 1400 - 0x4000 17FF | 1 KB | TIM7 | Section 29.5.31: TIMx register map Section 31.4.9: TIMx register map |
| | 0x4000 1000 - 0x4000 13FF | 1 KB | TIM6 | Section 31.4.9: TIMx register map |
| | 0x4000 0C00 - 0x4000 0FFF | 1 KB | TIM5 | Section 29.5.31: TIMx register map |
| | 0x4000 0800 - 0x4000 0BFF | 1 KB | TIM4 | Section 29.5.31: TIMx register map |
| | 0x4000 0400 - 0x4000 07FF | 1 KB | TIM3 | Section 29.5.31: TIMx register map |
| | 0x4000 0000 - 0x4000 03FF | 1 KB | TIM2 | Section 29.5.31: TIMx register map |

1. Refer to [Table 1: STM32G4 Series memory density](#), [Table 2: Product specific features](#) and to the device datasheets for the GPIO ports and peripherals available on your device. The memory area corresponding to unavailable GPIO ports or peripherals are reserved (highlighted in gray).

2.3 Bit banding

The Cortex®-M4 with FPU memory map includes two bit-band regions. These regions map each word in an alias region of memory to a bit in a bit-band region of memory. Writing to a word in the alias region has the same effect as a read-modify-write operation on the targeted bit in the bit-band region.

In the STM32G4 Series devices both the peripheral registers and the SRAM are mapped to a bit-band region, so that single bit-band write and read operations are allowed. The operations are only available for Cortex®-M4 with FPU accesses, and not from other bus masters (e.g. DMA).

A mapping formula shows how to reference each word in the alias region to a corresponding bit in the bit-band region. The mapping formula is:

$$\text{bit_word_addr} = \text{bit_band_base} + (\text{byte_offset} \times 32) + (\text{bit_number} \times 4)$$

where:

- *bit_word_addr* is the address of the word in the alias memory region that maps to the targeted bit
- *bit_band_base* is the starting address of the alias region
- *byte_offset* is the number of the byte in the bit-band region that contains the targeted bit
- *bit_number* is the bit position (0-7) of the targeted bit

Example

The following example shows how to map bit 2 of the byte located at SRAM1 address 0x20000300 to the alias region:

$$0x22006008 = 0x22000000 + (0x300 \times 32) + (2 \times 4)$$

Writing to address 0x22006008 has the same effect as a read-modify-write operation on bit 2 of the byte at SRAM1 address 0x20000300.

Reading address 0x22006008 returns the value (0x01 or 0x00) of bit 2 of the byte at SRAM1 address 0x20000300 (0x01: bit set; 0x00: bit reset).

2.4 Embedded SRAM

The STM32G4 Series category 3 devices feature up to 128 Kbytes SRAM:

- 80 Kbytes SRAM1 (mapped at address 0x2000 0000)
- 16 Kbytes SRAM2 (mapped at address 0x2001 4000)
- 32 Kbytes CCM SRAM (mapped at address 0x1000 0000 and end of SRAM2)

The STM32G4 Series category 4 devices feature up to 112 Kbytes SRAM:

- 80 Kbytes SRAM1 (mapped at address 0x2000 0000)
- 16 Kbytes SRAM2 (mapped at address 0x2001 4000)
- 16 Kbytes CCM SRAM (mapped at address 0x1000 0000 and end of SRAM2)

The STM32G4 Series category 2 devices feature up to 32 Kbytes SRAM:

- 16 Kbytes SRAM1 (mapped at address 0x2000 0000)
- 6 Kbytes SRAM2 (mapped at address 0x2000 4000)
- 10 Kbytes CCM SRAM (mapped at address 0x1000 0000 and end of SRAM2)

These SRAM can be accessed as bytes, half-words (16 bits) or full words (32 bits). These memories can be addressed at maximum system clock frequency without wait state and thus by both CPU and DMA.

The CPU can access the SRAM1 through the system bus or through the ICode/DCode buses when boot from SRAM1 is selected or when physical remap is selected

([Section 10.2.1: SYSCFG memory remap register \(SYSCFG_MEMRMP\)](#) in the SYSCFG controller). To get the maximum performance on SRAM1 execution, physical remap should be selected (boot or software selection).

CCM SRAM is mapped at address 0x1000 0000.

Execution can be performed from CCM SRAM with maximum performance without any remap thanks to access through ICode bus.

The CCM SRAM is aliased at address following the end of SRAM2 (0x2000 5800 for category 2 devices, 0x2001 8000 for category 3 devices, 0x2001 8000 for category 4 devices), offering a continuous address space with the SRAM1 and SRAM2.

2.4.1 Parity check

On the Category 3 and Category 4 devices, a parity check is implemented on the first 32 Kbytes of SRAM1 and on the whole CCM SRAM.

On the Category 2 devices, a parity check is implemented on the whole SRAM1 and CCM SRAM.

The user can enable the parity check using the option bit SRAM_PE in the user option byte (refer to [Section 3.4.1: Option bytes description](#)).

The data bus width is 36 bits because 4 bits are available for parity check (1 bit per byte) in order to increase memory robustness, as required for instance by Class B or SIL norms.

The parity bits are computed and stored when writing into the SRAM. Then, they are automatically checked when reading. If one bit fails, an NMI is generated. The same error can also be linked to the BRK_IN Break input of TIM1/TIM8/TIM15/TIM16/TIM17/TIM20, and to hrtim_sys_ft with the SPL control bit in the [Section 10.2.8: SYSCFG configuration register 2 \(SYSCFG_CFGR2\)](#). The SRAM Parity Error flag (SPF) is available in the [Section 10.2.8: SYSCFG configuration register 2 \(SYSCFG_CFGR2\)](#).

Note: When enabling the SRAM parity check, it is advised to initialize by software the whole SRAM memory at the beginning of the code, to avoid getting parity errors when reading non-initialized locations.

2.4.2 CCM SRAM Write protection

The CCM SRAM can be write protected with a page granularity of 1 Kbyte.

Table 4. CCM SRAM organization

| Page number | Start address | End address |
|------------------------|---------------|-------------|
| Page 0 | 0x1000 0000 | 0x1000 03FF |
| Page 1 | 0x1000 0400 | 0x1000 07FF |
| Page 2 | 0x1000 0800 | 0x1000 0BFF |
| Page 3 | 0x1000 0C00 | 0x1000 0FFF |
| Page 4 | 0x1000 1000 | 0x1000 13FF |
| Page 5 | 0x1000 1400 | 0x1000 17FF |
| Page 6 | 0x1000 1800 | 0x1000 1BFF |
| Page 7 | 0x1000 1C00 | 0x1000 1FFF |
| Page 8 | 0x1000 2000 | 0x1000 23FF |
| Page 9 | 0x1000 2400 | 0x1000 27FF |
| Page 10 ⁽¹⁾ | 0x1000 2800 | 0x1000 2BFF |

Table 4. CCM SRAM organization (continued)

| Page number | Start address | End address |
|------------------------|---------------|-------------|
| Page 11 ⁽¹⁾ | 0x1000 2C00 | 0x1000 2FFF |
| Page 12 ⁽¹⁾ | 0x1000 3000 | 0x1000 33FF |
| Page 13 ⁽¹⁾ | 0x1000 3400 | 0x1000 37FF |
| Page 14 ⁽¹⁾ | 0x1000 3800 | 0x1000 3BFF |
| Page 15 ⁽¹⁾ | 0x1000 3C00 | 0x1000 3FFF |
| Page 16 ⁽²⁾ | 0x1000 4000 | 0x1000 43FF |
| Page 17 ⁽²⁾ | 0x1000 4400 | 0x1000 47FF |
| Page 18 ⁽²⁾ | 0x1000 4800 | 0x1000 4BFF |
| Page 19 ⁽²⁾ | 0x1000 4C00 | 0x1000 4FFF |
| Page 20 ⁽²⁾ | 0x1000 5000 | 0x1000 53FF |
| Page 21 ⁽²⁾ | 0x1000 5400 | 0x1000 57FF |
| Page 22 ⁽²⁾ | 0x1000 5800 | 0x1000 5BFF |
| Page 23 ⁽²⁾ | 0x1000 5C00 | 0x1000 5FFF |
| Page 24 ⁽²⁾ | 0x1000 6000 | 0x1000 63FF |
| Page 25 ⁽²⁾ | 0x1000 6400 | 0x1000 67FF |
| Page 26 ⁽²⁾ | 0x1000 6800 | 0x1000 6BFF |
| Page 27 ⁽²⁾ | 0x1000 6C00 | 0x1000 6FFF |
| Page 28 ⁽²⁾ | 0x1000 7000 | 0x1000 73FF |
| Page 29 ⁽²⁾ | 0x1000 7400 | 0x1000 77FF |
| Page 30 ⁽²⁾ | 0x1000 7800 | 0x1000 7BFF |
| Page 31 ⁽²⁾ | 0x1000 7C00 | 0x1000 7FFF |

1. Available on Category 3 and Category 4 devices only.

2. Available on Category 3 devices only.

The write protection can be enabled in [Section 10.2.9: SYSCFG CCM SRAM write protection register \(SYSCFG_SWPR\)](#) in the SYSCFG block. This is a register with write ‘1’ once mechanism, which means by writing ‘1’ on a bit it sets up the write protection for that page of SRAM and it can be removed/cleared by a system reset only.

2.4.3 CCM SRAM read protection

The CCMSRAM is protected with the Read protection (RDP). Refer to [Section 3.5.1: Read protection \(RDP\)](#) for more details.

2.4.4 CCM SRAM erase

The CCMSRAM can be erased with a system reset using the option bit CCMSRAM_RST in the user option byte (refer to [Section 3.4.1: Option bytes description](#)).

The CCM SRAM erase can also be requested by software by setting the bit CCMSR in the [Section 10.2.7: SYSCFG CCM SRAM control and status register \(SYSCFG_SCSR\)](#).

2.5 Flash memory overview

The Flash memory is composed of two distinct physical areas:

- The main Flash memory block. It contains the application program and user data if necessary.
- The information block. It is composed of three parts:
 - Option bytes for hardware and memory protection user configuration.
 - System memory that contains the ST proprietary code.
 - OTP (one-time programmable) area

The Flash interface implements instruction access and data access based on the AHB protocol. It also implements the logic necessary to carry out the Flash memory operations (program/erase) controlled through the Flash registers. Refer to [Section 3: Embedded Flash memory \(FLASH\) for category 3 devices](#), [Section 4: Embedded Flash memory \(FLASH\) for category 4 devices](#) and [Section 5: Embedded Flash memory \(FLASH\) for category 2 devices](#) for more details.

2.6 Boot configuration

2.6.1 Boot configuration

Three different boot modes can be selected through the BOOT0 pin or the nBOOT0 bit into the FLASH_OPTR register (if the nSWBOOT0 bit is cleared into the FLASH_OPTR register), and nBOOT1 bit in FLASH_OPTR register, as shown in the following table.

Table 5. Boot modes

| BOOT_LOCK | nBOOT1 FLASH_OPTR[23] | nBOOT0 FLASH_OPTR[27] | BOOT0 pin PB8 | nSWBOOT0 FLASH_OPTR[26] | Boot Memory Space Alias |
|-----------|--------------------------|--------------------------|------------------|----------------------------|--|
| 1 | X | X | X | X | Main Flash memory |
| 0 | X | X | 0 | 1 | Main Flash memory is selected as boot area |
| 0 | X | 1 | X | 0 | Main Flash memory is selected as boot area |
| 0 | 0 | X | 1 | 1 | Embedded SRAM1 is selected as boot area |
| 0 | 0 | 0 | X | 0 | Embedded SRAM1 is selected as boot area |
| 0 | 1 | X | 1 | 1 | System memory is selected as boot area |
| 0 | 1 | 0 | X | 0 | System memory is selected as boot area |

The values on both BOOT0 pin (coming from the pin or the option bit) and nBOOT1 bit are latched on the 4th edge of the internal startup clock source after reset release. It is up to the user to set nBOOT1 and BOOT0 to select the required boot mode.

The BOOT0 pin or user option bit (depending on the nSWBOOT0 bit value in the FLASH_OPTR register), and nBOOT1 bit are also re-sampled when exiting from Standby mode. Consequently, they must be kept in the required Boot mode configuration in Standby mode. After this startup delay has elapsed, the CPU fetches the top-of-stack value from address 0x0000 0000, then starts code execution from the boot memory at 0x0000 0004.

Depending on the selected boot mode, main Flash memory, system memory or SRAM1 is accessible as follows:

- Boot from main Flash memory: the main Flash memory is aliased in the boot memory space (0x0000 0000), but still accessible from its original memory space (0x0800 0000). In other words, the Flash memory contents can be accessed starting from address 0x0000 0000 or 0x0800 0000.
- Boot from system memory: the system memory is aliased in the boot memory space (0x0000 0000), but still accessible from its original memory space (0x1FFF 0000).
- Boot from the embedded SRAM1: the SRAM1 is aliased in the boot memory space (0x0000 0000), but it is still accessible from its original memory space (0x2000 0000).

PB8/BOOT0 GPIO is configured in:

- Input mode during the complete reset phase if the option bit nSWBOOT0 is set into the FLASH_OPTR register and then switches automatically in analog mode after reset is released (BOOT0 pin).
- Input mode from the reset phase to the completion of the option byte loading if the bit nSWBOOT0 is cleared into the FLASH_OPTR register (BOOT0 value coming from the option bit). It switches then automatically to the analog mode even if the reset phase is not complete.

Note:

When the device boots from SRAM, in the application initialization code, you have to relocate the vector table in SRAM using the NVIC exception table and the offset register.

When booting from the main Flash memory, the application software can either boot from bank 1 or from bank 2 (only for category 3 devices). By default, boot from bank 1 is selected. To select boot from Flash memory bank 2, set the BFB2 bit in the user option bytes. When this bit is set and the boot pins are in the boot from main Flash memory configuration, the device boots from system memory, and the boot loader jumps to execute the user application programmed in Flash memory bank 2. For further details, please refer to AN2606.

See [Table 13: Access status versus protection level and execution modes](#) for bootloader function for different RDP levels

Forcing boot from user Flash memory

Regardless the boot configuration it is possible to force booting from a unique entry point in main Flash memory. See [Section Embedded Flash memory \(FLASH\)](#).

Physical remap

Once the boot pins mode is selected, the application software can modify the memory accessible in the code area (in this way the code can be executed through the ICode bus in place of the System bus). This modification is performed by programming the

[Section 10.2.1: SYSCFG memory remap register \(SYSCFG_MEMRMP\)](#) in the SYSCFG controller.

The following memories can thus be remapped:

- Main Flash memory

- System memory
- Embedded SRAM1
- FSMC bank 1 (NOR/PSRAM 1 and 2)
- QUADSPI memory

Table 6. Memory mapping versus boot mode/physical remap⁽¹⁾

| Addresses | Boot/remap in main Flash memory | Boot/remap in embedded SRAM 1 | Boot/remap in system memory | Remap in FSMC | Remap in QUADSPI |
|--------------------------------------|----------------------------------|----------------------------------|----------------------------------|---|----------------------------------|
| 0x2000 0000 - 0x2002 3FFF | SRAM1 | SRAM1 | SRAM1 | SRAM1 | SRAM1 |
| 0x1FFF 7000 - 0x1FFF FFFF | System memory/OTP/ Options bytes | System memory/OTP/ Options bytes |
| 0x1000 8000 - 0x1FFE FFFF | Reserved | Reserved | Reserved | Reserved | Reserved |
| 0x1000 0000 - 0x1000 7FFF | CCM SRAM | CCM SRAM | CCM SRAM | CCM SRAM | CCM SRAM |
| 0x0808 0000 - 0x0FFF FFFF | Reserved | Reserved | Reserved | Reserved | Reserved |
| 0x0800 0000 - 0x0807 FFFF | Flash memory | Flash memory | Flash memory | Flash memory | Flash memory |
| 0x0400 0000 - 0x07FF FFFF | Reserved | Reserved | Reserved | FSMC bank 1 NOR/ PSRAM 2 (128 MB) Aliased | QUADSPI bank (128 MB) Aliased |
| 0x0010 0000 - 0x03FF FFFF | Reserved | Reserved | Reserved | FSMC bank 1 NOR/ PSRAM 1 (128 MB) Aliased | QUADSPI bank (128 MB) Aliased |
| 0x0000 0000 - 0x0007 FFFF (2) (3) | Flash Aliased | SRAM1 Aliased | System memory (28 KB) Aliased | FSMC bank 1 NOR/ PSRAM 1 (128 MB) Aliased | QUADSPI Aliased |

1. Reserved memory area highlighted in gray in the table.
2. When the FSMC is mapped at address 0x0000 0000, only the first two regions of bank 1 memory controller (bank 1 NOR/PSRAM 1 and NOR/PSRAM 2) can be remapped. When the FSMC is remapped at address 0x0000 0000, only 128 MB are remapped. In remap mode, the CPU can access the external memory via ICode bus instead of system bus, which boosts up the performance.
3. Even when aliased in the boot memory space, the related memory is still accessible at its original memory space.

Embedded boot loader

The embedded boot loader is located in the system memory, programmed by ST during production. Refer to AN2606 STM32 microcontroller system memory boot mode.

3 Embedded Flash memory (FLASH) for category 3 devices

3.1 Introduction

The Flash memory interface manages CPU AHB ICode and DCode accesses to the Flash memory. It implements the erase and program Flash memory operations and the read and write protection mechanisms.

The Flash memory interface accelerates code execution with a system of instruction prefetch and cache lines.

3.2 FLASH main features

- Up to 512 Kbyte of Flash memory with dual bank architecture supporting read-while-write capability (RWW).
- Flash memory read operations with two data width modes supported:
 - Single bank mode DBANK=0: read access of 128 bits
 - Dual bank mode DBANK=1: read access of 64 bits
- Page erase, bank erase and mass erase (both banks)

Flash memory interface features:

- Flash memory read operations
- Flash memory program/erase operations
- Read protection activated by option (RDP)
- 4 Write protection areas (2 per bank when DBANK=1 and 4 for full memory when DBANK=0)
- 2 proprietary code read protection areas (1 per bank when DBANK=1, 2 for all memory when DBANK=0)
- 2 Securable memory areas defined by option (1 per bank when DBANK = 1, 1 for all memory when DBANK = 0).
- Prefetch on ICODE
- Instruction Cache: 32 cache lines of 4 x 64 or 2 x 128 bits on ICode (1 KB RAM)
- Data Cache: 8 cache lines of 4 x 64 bits or 2 x 128 on DCode (256 B RAM)
- Error Code Correction ECC: 8 bits per 64-bit double-word
 - DBANK=1: 8 + 64 = 72 bits, 2 bits detection, 1 bit correction
 - DBANK=0: (8+64) + (8+64) = 144 bits, 2 bits detection, 1 bit correction
- Option byte loader
- Low-power mode

3.3 FLASH functional description

3.3.1 Flash memory organization

The Flash memory has the following main features:

- Capacity up to 512 Kbytes, in single bank mode (read width of 128 bits) or in dual bank mode (read width of 64 bits)
- Supports dual boot mode thanks to the BFB2 option bit (only in dual bank mode)
- Dual bank mode when DBANK bit is set:
 - 512 KB organized in 2 banks for main memory
 - Page size of 2 Kbyte
 - 72 bits wide data read (64 bits plus 8 ECC bits)
 - Bank and Mass erase
- Single bank mode when DBANK is reset:
 - 512 KB organized in one single bank for main memory
 - Page size of 4 Kbyte
 - 144 bits wide data read (128 bits plus 2x8 ECC bits)
 - Mass erase

The Flash memory is organized as follows:

- A main memory block organized depending on the dual bank configuration bit:
 - When dual bank is enabled (DBANK bit set), the Flash is divided in 2 banks of 256 KB, and each bank is organized as follows:
 - The main memory block containing 128 pages of 2 Kbyte
 - Each page is composed of 8 rows of 256 bytes
 - When dual bank is disabled (DBANK bit reset), the main memory block is organized as one single bank of 512 KB as follows:
 - The main memory block containing 128 pages of 4 Kbyte
 - Each page is composed of 8 rows of 512 bytes
- An Information block containing:
 - System memory from which the device boots in System memory boot mode. The area is reserved for use by STMicroelectronics and contains the bootloader that is used to reprogram the Flash memory through one of the following interfaces: USART, SPI, I2C, FDCAN, USB. It is programmed by STMicroelectronics when the device is manufactured, and protected against spurious write/erase operations. For further details, please refer to the AN2806 available from www.st.com.
 - 1 Kbyte (128 double word) OTP (one-time programmable) bytes for user data. The OTP area is available in Bank 1 only. The OTP data cannot be erased and can be written only once. If only one bit is at 0, the entire double word cannot be written anymore, even with the value 0x0000 0000 0000 0000.
 - Option bytes for user configuration.

The memory organization is based on a main area and an information block as shown in [Table 28](#).

Table 7. Flash module - 512/256/128 KB dual bank organization (64 bits read width)

| Flash area | Flash memory addresses | Size (bytes) | Name |
|---------------------------------|---------------------------|---------------------------|---------------|
| Main memory (512/256/128 KB) | 0x0800 0000 - 0x0800 07FF | 2 K | Page 0 |
| | 0x0800 0800 - 0x0800 0FFF | 2 K | Page 1 |
| | 0x0800 1000 - 0x0800 17FF | 2 K | Page 2 |
| | 0x0800 1800 - 0x0800 1FFF | 2 K | Page 3 |
| | - | - | - |
| | - | - | - |
| | - | - | - |
| | - | - | - |
| | 0x0803 F800 - 0x0803 FFFF | 2 K | Page 127 |
| | 0x0804 0000 - 0x0804 07FF | 2 K | Page 0 |
| | 0x0804 0800 - 0x0804 0FFF | 2 K | Page 1 |
| | 0x0804 1000 - 0x0804 17FF | 2 K | Page 2 |
| | 0x0804 1800 - 0x0804 1FFF | 2 K | Page 3 |
| | - | - | - |
| | - | - | - |
| | - | - | - |
| | - | - | - |
| | 0x0807 F800 - 0x0807 FFFF | 2 K | Page 127 |
| Information block | Bank 1 | 0x1FFF 0000 - 0x1FFF 6FFF | 28 K |
| | Bank 2 | 0x1FFF 8000 - 0x1FFF EFFF | 28 K |
| | Bank 1 | 0x1FFF 7000 - 0x1FFF 73FF | 1 K |
| | Bank 1 | 0x1FFF 7800 - 0x1FFF 782F | 48 |
| | Bank 2 | 0x1FFF F800 - 0x1FFF F82F | 48 |
| | | | System memory |
| | | | OTP area |
| | | | Option bytes |

1. For 256 KB devices: from page 0 to page 63
 For 128 KB devices: from page 0 to page 31

Table 8. Flash module - 512/256/128 KB single bank organization (128 bits read width)

| Flash area | Flash memory addresses | Size (bytes) | Name |
|--|---------------------------|---------------------------|----------|
| Main memory ⁽¹⁾ (512/256/128 KB) | 0x0800 0000 - 0x0800 0FFF | 4 K | Page 0 |
| | 0x0800 1000 - 0x0800 1FFF | 4 K | Page 1 |
| | 0x0800 2000 - 0x0800 2FFF | 4 K | Page 2 |
| | - | - | - |
| | - | - | - |
| | - | - | - |
| | - | - | - |
| | - | - | - |
| | - | - | - |
| | - | - | - |
| | 0x0807 F000 - 0x0807 FFFF | 4 K | Page 127 |
| Information block | Bank 1 | 0x1FFF 0000 - 0x1FFF 6FFF | 28 K |
| | Bank 2 | 0x1FFF 8000 - 0x1FFF EFFF | 28 K |
| | Bank 1 | 0x1FFF 7000 - 0x1FFF 73FF | 1 K |
| | Bank 1 | 0x1FFF 7800 - 0x1FFF 782F | 48 |
| | Bank 2 | 0x1FFF F800 - 0x1FFF F82F | 48 |

1. For 256 KB devices: from page 0 to page 63

For 128KB devices: from page 0 to page 31

3.3.2 Error code correction (ECC)

Dual bank mode (DBANK=1, 64-bits data width)

Data in Flash memory are 72-bits words: 8 bits are added per double word (64 bits). The ECC mechanism supports:

- One error detection and correction
- Two errors detection

When one error is detected and corrected, the flag ECCC (ECC correction) is set in [Flash ECC register \(FLASH_ECCR\)](#). If ECCCIE is set, an interrupt is generated.

When two errors are detected, a flag ECCD (ECC detection) is set in FLASH_ECCR register. In this case, a NMI is generated.

When an ECC error is detected, the address of the failing double word and its associated bank are saved in ADDR_ECC[20:0] and BK_ECC in the FLASH_ECCR register. ADDR_ECC[2:0] are always cleared.

When ECCC or ECCD is set, ADDR_ECC and BK_ECC are not updated if a new ECC error occurs. FLASH_ECCR is updated only when ECC flags are cleared.

Single bank mode (DBANK=0, 128-bits data width)

Data in Flash memory are 144-bits words: 8 bits are added per each double word. The ECC mechanism supports:

- One error detection and correction
- Two errors detection per 64 double words

The user must first check the SYSF_ECC bit, and if it is set, the user must refer to the DBANK=1 programming model (because system Flash is always on 2 banks). If the bit is not set, the user must refer to the following programming model:

Each double word (bits 63:0 and bits 127:64) has ECC.

When one error is detected in 64 LSB bits (bits 63:0) and corrected, a flag ECCC (ECC correction) is set in the FLASH_ECCR register.

When one error is detected in 64 MSB bits (bits 127:64) and corrected, a flag ECCC2 (ECC2 correction) is set in the FLASH_ECCR register.

If the ECCCIE is set, an interrupt is generated. The user has to read ECCC and ECCC2 to see which part of the 128-bits data has been corrected (either 63:0, 127:64 or both).

When two errors are detected in 64 LSB bits, a flag ECCD (ECC detection) is set in the FLASH_ECCR register.

When two errors are detected in 64 MSB bits (bits 127:64), a flag ECCD2 (ECC2 detection) is set in the FLASH_ECCR register.

In this case, a NMI is generated. The user has to read ECCD and ECCD2 to see which part of the 128-bits data has error detection (either 63:0, 127:64 or both).

When an ECC error is detected, the address of the failing the 2 times double word is saved into ADDR_ECC[20:0] in FLASH_ECCR. ADDR_ECC [20:0] contains an address of a 2 times double word.

The ADDR_ECC[3:0] are always cleared. BK_ECC is not used in this mode.

When ECCC/ECCC2 or ECCD/ECCD2 is/are set, if a new ECC error occurs, the ADDR_ECC is not updated. The FLASH_ECCR is updated only if the ECC flags (ECCC/ECCC2/ECCD/ECCD2) are cleared.

Note: For a virgin data: 0xFFFFFFFFFFFF FFFF, one error is detected and corrected but two errors detection is not supported.

When an ECC error is reported, a new read at the failing address may not generate an ECC error if the data is still present in the current buffer, even if ECCC and ECCD are cleared.

3.3.3 Read access latency

To correctly read data from Flash memory, the number of wait states (LATENCY) must be correctly programmed in the [Flash access control register \(FLASH_ACR\)](#) according to the frequency of the CPU clock (HCLK) and the internal voltage range of the device V_{CORE}. Refer to [Section 6.1.5: Dynamic voltage scaling management](#). [Table 9](#) shows the correspondence between wait states and CPU clock frequency.

Table 9. Number of wait states according to CPU clock (HCLK) frequency

| Wait states (WS) (LATENCY) | HCLK (MHz) | | |
|-------------------------------|---|--|---------------------------|
| | V _{CORE} Range 1 boost mode | V _{CORE} Range 1 normal mode | V _{CORE} Range 2 |
| 0 WS (1 CPU cycles) | ≤ 34 | ≤ 30 | ≤ 12 |
| 1 WS (2 CPU cycles) | ≤ 68 | ≤ 60 | ≤ 24 |
| 2 WS (3 CPU cycles) | ≤ 102 | ≤ 90 | ≤ 26 |
| 3 WS (4 CPU cycles) | ≤ 136 | ≤ 120 | - |
| 4 WS (5 CPU cycles) | ≤ 170 | ≤ 150 | - |

After reset, the CPU clock frequency is 16 MHz and 1 wait state (WS) is configured in the FLASH_ACR register.

When changing the CPU frequency, the following software sequences must be applied in order to tune the number of wait states needed to access the Flash memory:

Increasing the CPU frequency:

1. Program the new number of wait states to the LATENCY bits in the *Flash access control register (FLASH_ACR)*.
2. Check that the new number of wait states is taken into account to access the Flash memory by reading the FLASH_ACR register.
3. Analyze the change of CPU frequency change caused either by:
 - changing clocksource defined by SW bits in RCC_CFGR register
 - or by CPU clock prescaler defined by HPRE bits in RCC_CFGR

If some of above two steps decreases the CPU frequency, firstly perform this step and then the rest. Otherwise modify The CPU clock source by writing the SW bits in the RCC_CFGR register and then (if needed) modify the CPU clock prescaler by writing the HPRE bits in RCC_CFGR.

4. Check that the new CPU clock source or/and the new CPU clock prescaler value is/are taken into account by reading the clock source status (SWS bits) or/and the AHB prescaler value (HPRE bits), respectively, in the RCC_CFGR register.

Decreasing the CPU frequency:

1. Modify the CPU clocksource by writing the SW bits in the RCC_CFGR register.
2. If needed, modify the CPU clock prescaler by writing the HPRE bits in RCC_CFGR.
3. Analyze the change of CPU frequency change caused either by:
 - changing clock source defined by SW bits in RCC_CFGR register
 - or by CPU clock prescaler defined by HPRE bits in RCC_CFGR

If some of above two steps increases the CPU frequency, firstly perform another step and then this step. Otherwise modify The CPU clock source by writing the SW bits in the

RCC_CFGR register and then (if needed) modify the CPU clock prescaler by writing the HPRE bits in RCC_CFGR.

4. Check that the new CPU clock source or/and the new CPU clock prescaler value is/are taken into account by reading the clock source status (SWS bits) or/and the AHB prescaler value (HPRE bits), respectively, in the RCC_CFGR register.
5. Program the new number of wait states to the LATENCY bits in *Flash access control register (FLASH_ACR)*.
6. Check that the new number of wait states is used to access the Flash memory by reading the FLASH_AC R register.

3.3.4 Adaptive real-time memory accelerator (ART Accelerator)

The proprietary Adaptive real-time (ART) memory accelerator is optimized for STM32 industry-standard Arm® Cortex®-M4 with FPU processors. It balances the inherent performance advantage of the Arm® Cortex®-M4 with FPU over Flash memory technologies, which normally requires the processor to wait for the Flash memory at higher operating frequencies.

To release the processor full performance, the accelerator implements an instruction prefetch queue and branch cache which increases program execution speed from the 64-bit Flash memory. Based on CoreMark benchmark, the performance achieved thanks to the ART accelerator is equivalent to 0 wait state program execution from Flash memory at a CPU frequency up to 170 MHz.

Instruction prefetch

The Cortex®-M4 fetches the instruction over the ICode bus and the literal pool (constant/data) over the DCode bus. The prefetch block aims at increasing the efficiency of ICode bus accesses.

In case of Single bank mode (DBANK option bit is reset), each Flash memory read operation provides 128 bits from either four instructions of 32 bits or eight instructions of 16 bits depending on the launched program. This 128-bits current instruction line is saved in a current buffer, and in case of sequential code, at least four CPU cycles are needed to execute the previous read instruction line.

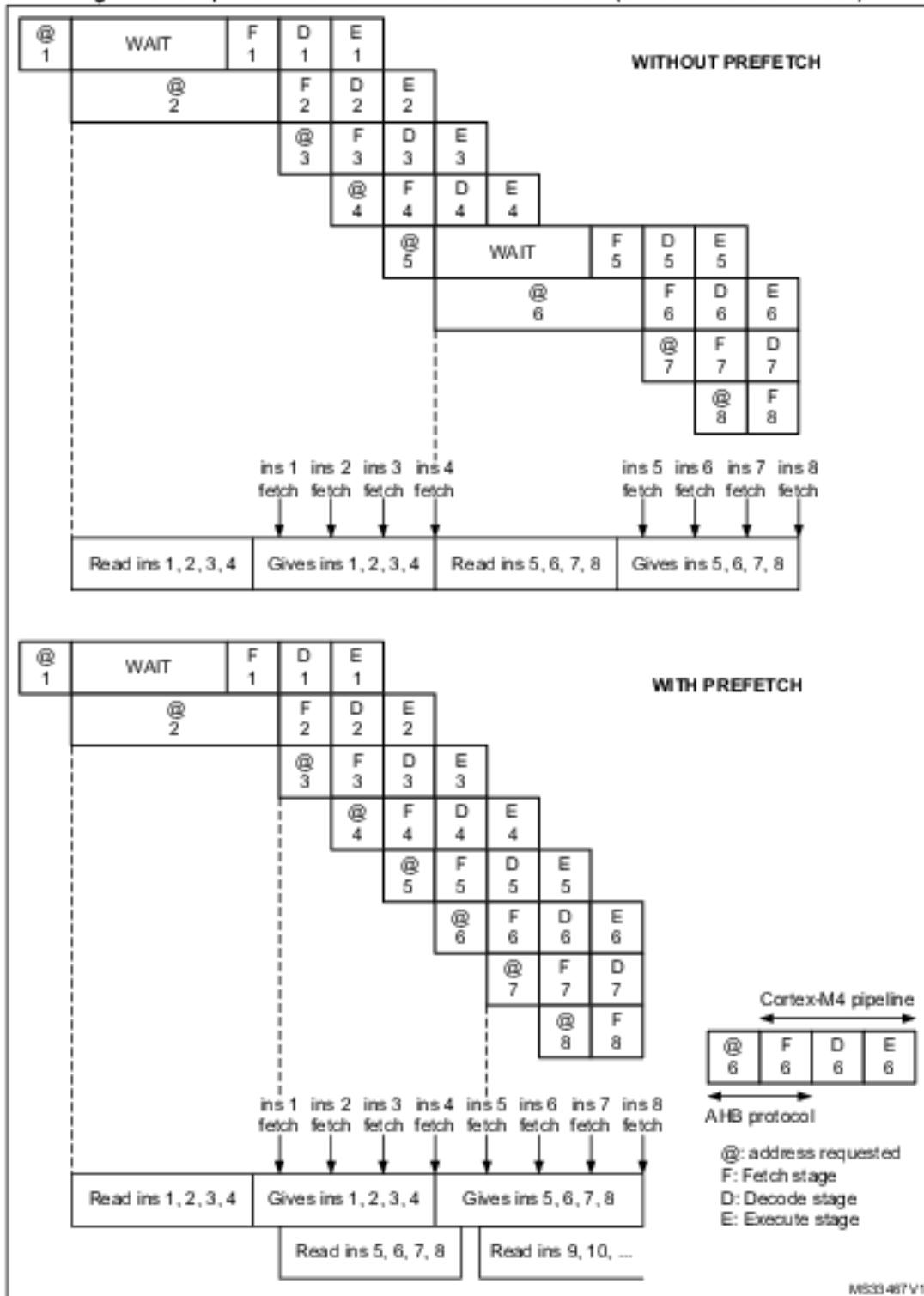
When in dual bank mode (DBANK option bit is set), each Flash memory read operation provides 64 bits from either two instructions of 32 bits or four instructions of 16 bits depending on the launched program. This 64-bits current instruction line is saved in a current buffer, and in case of sequential code, at least two CPU cycles are needed to execute the previous read instruction line.

Prefetch on the ICode bus can be used to read the next sequential instruction line from the Flash memory while the current instruction line is being requested by the CPU.

Prefetch is enabled by setting the PRFTEN bit in the *Flash access control register (FLASH_ACR)*. This feature is useful if at least one wait state is needed to access the Flash memory.

Figure 9 shows the execution of sequential 16-bit instructions with and without prefetch when 3 WS are needed to access the Flash memory.

Figure 3. Sequential 16-bit instructions execution (64-bit read data width)



When the code is not sequential (branch), the instruction may not be present in the currently used instruction line or in the prefetched instruction line. In this case (miss), the penalty in terms of number of cycles is at least equal to the number of wait states.

If a loop is present in the current buffer, no new flash access is performed.

Instruction cache memory (I-Cache)

To limit the time lost due to jumps, it is possible to retain 32 lines of 4 x 64 bits in dual bank mode or 32 lines of 2 x 128 bits in single bank mode in an instruction cache memory. This feature can be enabled by setting the instruction cache enable (ICEN) bit in the [Flash access control register \(FLASH_ACR\)](#). Each time a miss occurs (requested data not present in the currently used instruction line, in the prefetched instruction line or in the instruction cache memory), the line read is copied into the instruction cache memory. If some data contained in the instruction cache memory are requested by the CPU, they are provided without inserting any delay. Once all the instruction cache memory lines have been filled, the LRU (least recently used) policy is used to determine the line to replace in the instruction memory cache. This feature is particularly useful in case of code containing loops.

The Instruction cache memory is enable after system reset.

Data cache memory (D-Cache)

Literal pools are fetched from Flash memory through the DCode bus during the execution stage of the CPU pipeline. Each DCode bus read access fetches 64 or 128 bits which are saved in a current buffer. The CPU pipeline is consequently stalled until the requested literal pool is provided. To limit the time lost due to literal pools, accesses through the AHB databus DCode have priority over accesses through the AHB instruction bus ICode.

If some literal pools are frequently used, the data cache memory can be enabled by setting the data cache enable (DCEN) bit in the [Flash access control register \(FLASH_ACR\)](#). This feature works like the instruction cache memory, but the retained data size is limited to 8 rows of 4*64 bits in dual bank mode and to 8 rows of 2*128 bits in single bank mode.

The Data cache memory is enable after system reset.

Note: *The D-Cache is active only when data is requested by the CPU (not by DMA1 and DMA2). Data in option bytes block are not cacheable.*

3.3.5 Flash program and erase operations

The STM32G4 Series embedded Flash memory can be programmed using in-circuit programming or in-application programming.

The **in-circuit programming (ICP)** method is used to update the entire contents of the Flash memory, using the JTAG, SWD protocol or the boot loader to load the user application into the microcontroller. ICP offers quick and efficient design iterations and eliminates unnecessary package handling or socketing of devices.

In contrast to the ICP method, **in-application programming (IAP)** can use any communication interface supported by the microcontroller (I/Os, USB, CAN, UART, I²C, SPI, etc.) to download programming data into memory. IAP allows the user to re-program the Flash memory while the application is running. Nevertheless, part of the application has to have been previously programmed in the Flash memory using ICP.

The contents of the Flash memory are not guaranteed if a device reset occurs during a Flash memory operation.

An on-going Flash memory operation does not block the CPU as long as the CPU does not access the same Flash memory bank. Code or data fetches are possible on one bank while

a write/erase operation is performed to the other bank (refer to [Section 3.3.8: Read-while-write \(RWW\) available only in dual bank mode \(DBANK=1\)](#)).

The Flash erase and programming is only possible in the voltage scaling range 1. The VOS[1:0] bits in the PWR_CR1 must be programmed to 01b.

On the contrary, during a program/erase operation to the Flash memory, any attempt to read the same Flash memory bank stalls the bus. The read operation proceeds correctly once the program/erase operation has completed.

Unlocking the Flash memory

After reset, write is not allowed in the [Flash control register \(FLASH_CR\)](#) to protect the Flash memory against possible unwanted operations due, for example, to electric disturbances. The following sequence is used to unlock this register:

1. Write KEY1 = 0x45670123 in the [Flash key register \(FLASH_KEYR\)](#)
2. Write KEY2 = 0xCDEF89AB in the FLASH_KEYR register.

Any wrong sequence locks up the FLASH_CR register until the next system reset. In the case of a wrong key sequence, a bus error is detected and a Hard Fault interrupt is generated.

The FLASH_CR register can be locked again by software by setting the LOCK bit in the FLASH_CR register.

Note: *The FLASH_CR register cannot be written when the BSY bit in the Flash status register (FLASH_SR) is set. Any attempt to write to it with the BSY bit set causes the AHB bus to stall until the BSY bit is cleared.*

3.3.6 Flash main memory erase sequences

The Flash memory erase operation can be performed at page level, bank level or on the whole Flash memory (Mass Erase). Mass Erase does not affect the Information block (system flash, OTP and option bytes).

Page erase

To erase a page, follow the procedure below:

1. Check that no Flash memory operation is ongoing by checking the BSY bit in the [Flash status register \(FLASH_SR\)](#).
2. Check and clear all error programming flags due to a previous programming. If not, PGERR is set.
3. In dual bank mode (DBANK option bit is set), set the PER bit and select the page to erase (PNB) with the associated bank (BKER) in the Flash control register (FLASH_CR). In single bank mode (DBANK option bit is reset), set the PER bit and select the page to erase (PNB). The BKER bit in the Flash control register (FLASH_CR) must be kept cleared.
4. Set the STRT bit in the FLASH_CR register.
5. Wait for the BSY bit to be cleared in the FLASH_SR register.

Note: The internal oscillator HSI16 (16 MHz) is enabled automatically when STRT bit is set, and disabled automatically when STRT bit is cleared, except if the HSI16 is previously enabled with HS1ON in RCC_CR register.

If the page erase is part of write-protected area (by WRP or PCR OP), WRPPER is set and the page erase request is aborted.

Bank 1, Bank 2 Mass erase (available only in dual bank mode when DBANK=1)

To perform a bank Mass Erase, follow the procedure below:

1. Check that no Flash memory operation is ongoing by checking the BSY bit in the FLASH_SR register.
2. Check and clear all error programming flags due to a previous programming. If not, PGSER is set.
3. Set the MER1 bit or MER2 (depending on the bank) in the [Flash control register \(FLASH_CR\)](#). Both banks can be selected in the same operation, in that case it corresponds to a mass erase.
4. Set the STRT bit in the FLASH_CR register.
5. Wait for the BSY bit to be cleared in the [Flash status register \(FLASH_SR\)](#).

Mass erase

To perform a Mass erase, follow the procedure below:

1. Check that no Flash memory operation is ongoing by checking the BSY bit in the FLASH_SR register.
2. Check and clear all error programming flags due to a previous programming. If not, PGSER is set.
3. Set the MER1 bit and MER2 in the Flash control register (FLASH_CR).
4. Set the STRT bit in the FLASH_CR register.
5. Wait for the BSY bit to be cleared in the Flash status register (FLASH_SR).

Note: The internal oscillator HSI16 (16 MHz) is enabled automatically when STRT bit is set, and disabled automatically when STRT bit is cleared, except if the HSI16 is previously enabled with HS1ON in RCC_CR register.

When DBANK=0, if only the MERA or the MERB bit is set, PGSER is set and no erase operation is performed.

If the bank to erase or if one of the banks to erase contains a write-protected area (by WRP or PCR OP), WRPPER is set and the mass erase request is aborted (for both banks if both are selected).

3.3.7 Flash main memory programming sequences

The Flash memory is programmed 72 bits at a time (64 bits + 8 bits ECC).

Programming in a previously programmed address is not allowed except if the data to write is full zero, and any attempt sets PROGERR flag in the [Flash status register \(FLASH_SR\)](#).

It is only possible to program double word (2 x 32-bit data).

- Any attempt to write byte or half-word sets SIZERR flag in the FLASH_SR register.
- Any attempt to write a double word which is not aligned with a double word address sets PGAEERR flag in the FLASH_SR register.

Standard programming

The Flash memory programming sequence in standard mode is as follows:

1. Check that no Flash main memory operation is ongoing by checking the BSY bit in the *Flash status register (FLASH_SR)*.
2. Check and clear all error programming flags due to a previous programming. If not, PGERR is set.
3. Set the PG bit in the *Flash control register (FLASH_CR)*.
4. Perform the data write operation at the desired memory address, inside main memory block or OTP area. Only double word can be programmed.
 - Write a first word in an address aligned with double word
 - Write the second word
5. Wait until the BSY bit is cleared in the FLASH_SR register.
6. Check that EOP flag is set in the FLASH_SR register (meaning that the programming operation has succeed), and clear it by software.
7. Clear the PG bit in the FLASH_SR register if there no more programming request anymore.

Note: When the flash interface has received a good sequence (a double word), programming is automatically launched and BSY bit is set. The internal oscillator HSI16 (16 MHz) is enabled automatically when PG bit is set, and disabled automatically when PG bit is cleared, except if the HSI16 is previously enabled with HS1ON in RCC_CR register.

If the user needs to program only one word, double word must be completed with the erase value 0xFFFF FFFF to launch automatically the programming.

ECC is calculated from the double word to program.

Fast programming for a row (64 double words if DBANK=1) or for half row (64 double words if DBANK=0)

This mode allows to program a row (64 double words if DBANK=1) or half row (64 double words if DBANK=0), and to reduce the page programming time by eliminating the need for verifying the flash locations before they are programmed and to avoid rising and falling time of high voltage for each double word. During fast programming, the CPU clock frequency (HCLK) must be at least 8 MHz.

Only the main memory can be programmed in Fast programming mode.

The Flash main memory programming sequence in standard mode is as follows:

1. In single bank mode (DBANK=0), perform a mass erase. If not, PGSERR is set. The Fast programming can be performed only if the code is executed from RAM or from bootloader. In dual bank mode (DBANK=1), perform a mass erase of the bank to program. If not, PGSERR is set.
2. Check that no Flash main memory operation is ongoing by checking the BSY bit in the *Flash status register (FLASH_SR)*.
3. Check and clear all error programming flag due to a previous programming.
4. Set the FSTPG bit in *Flash control register (FLASH_CR)*.
5. Write the 64 double words to program a row or half row. Only double words can be programmed:
 - Write a first word in an address aligned with double word
 - Write the second word.
6. Wait until the BSY bit is cleared in the FLASH_SR register.
7. Check that EOP flag is set in the FLASH_SR register (meaning that the programming operation has succeed), and clear it by software.
8. Clear the FSTPG bit in the FLASH_SR register if there's no more programming request anymore.

Note:

If the flash is attempted to be written in Fast programming mode while a read operation is ongoing in the same bank, the programming is aborted without any system notification (no error flag is set).

When the Flash interface has received the first double word, programming is automatically launched. The BSY bit is set when the high voltage is applied for the first double word, and it is cleared when the last double word has been programmed or in case of error. The internal oscillator HSI16 (16 MHz) is enabled automatically when FSTPG bit is set, and disabled automatically when FSTPG bit is cleared, except if the HSI16 is previously enabled with HSION in RCC_CR register.

The 64 double word must be written successively. The high voltage is kept on the flash for all the programming. Maximum time between two double words write requests is the time programming (around 2 x 25us). If a second double word arrives after this time programming, fast programming is interrupted and MISSERR is set.

High voltage mustn't exceed 8ms for a full row between 2 erases. This is guaranteed by the sequence of 64 double words successively written with a clock system greater or equal to 8MHz. An internal time-out counter counts 7ms when Fast programming is set and stops the programming when time-out is over. In this case the FASTERR bit is set.

If an error occurs, high voltage is stopped and next double word to programmed is not programmed. Anyway, all previous double words have been properly programmed.

Programming errors

Several kind of errors are detected. In case of error, the Flash operation (programming or erasing) is aborted.

- **PROGERR:** Programming Error

In standard programming: PROGERR is set if the word to write is not previously erased (except if the value to program is full zero).

- **SIZERR:** Size Programming Error

In standard programming or in fast programming: only double word can be programmed and only 32-bit data can be written. SIZERR is set if a byte or an half-word is written.

- **PGAERR:** Alignment Programming error

PGAERR is set if one of the following conditions occurs:

- In standard programming: the first word to be programmed is not aligned with a double word address, or the second word doesn't belong to the same double word address.
- In fast programming: the data to program doesn't belong to the same row than the previous programmed double words, or the address to program is not greater than the previous one.

- **PGSERR:** Programming Sequence Error

PGSERR is set if one of the following conditions occurs:

- In the standard programming sequence or the fast programming sequence: a data is written when PG and FSTPG are cleared.
- In the standard programming sequence or the fast programming sequence: MER1, MER2, and PER are not cleared when PG or FSTPG is set.
- In the fast programming sequence: the Mass erase is not performed before setting FSTPG bit.
- In the mass erase sequence: PG, FSTPG, and PER are not cleared when MER1 or MER2 is set.
- In the page erase sequence: PG, FSTPG, MER1 and MER2 are not cleared when PER is set.
- PGSERR is set also if PROGERR, SIZERR, PGAERR, WRPERR, MISSERR, FASTERR or PGSERR is set due to a previous programming error.
- When DBANK=0, in the case that only either MER1 or MER2 is set, PGSERR is set (bank mass erase is not allowed).

- **WRPERR:** Write Protection Error

WRPERR is set if one of the following conditions occurs:

- Attempt to program or erase in a write protected area (WRP) or in a PCROP area or in a Securable memory area.
- Attempt to perform a bank erase when one page or more is protected by WRP or PCROP.
- The debug features are connected or the boot is executed from SRAM or from System flash when the read protection (RDP) is set to Level 1.
- Attempt to modify the option bytes when the read protection (RDP) is set to Level 2.

- **MISSERR:** Fast Programming Data Miss Error

In fast programming: all the data must be written successively. MISSERR is set if the previous data programming is finished and the next data to program is not written yet.

- **FASTERR:** Fast Programming Error

In fast programming: FASTERR is set if one of the following conditions occurs:

- When FSTPG bit is set for more than 7 ms which generates a time-out detection.
- When the fast programming has been interrupted by a MISSERR, PGAERR, WRPERR or SIZERR.

If an error occurs during a program or erase operation, one of the following error flags is set in the FLASH_SR register:

PROGERR, SIZERR, PGAERR, PGSERR, MISSERR (Program error flags),

WRPERR (Protection error flag)

In this case, if the error interrupt enable bit ERRIE is set in the *Flash status register (FLASH_SR)*, an interrupt is generated and the operation error flag OPERR is set in the FLASH_SR register.

Note: *If several successive errors are detected (for example, in case of DMA transfer to the Flash memory), the error flags cannot be cleared until the end of the successive write requests.*

Programming and caches

If a Flash memory write access concerns some data in the data cache, the Flash write access modifies the data in the Flash memory and the data in the cache.

If an erase operation in Flash memory also concerns data in the data or instruction cache, you have to make sure that these data are rewritten before they are accessed during code execution. If this cannot be done safely, it is recommended to flush the caches by setting the DCRST and ICRST bits in the *Flash access control register (FLASH_ACR)*.

Note: *The I/D cache should be flushed only when it is disabled (IDCEN = 0).*

3.3.8 Read-while-write (RWW) available only in dual bank mode (DBANK=1)

The dual bank mode is available only when the DBANK option bit is set, allowing read-while-write operations. This feature allows to perform a read operation from one bank while an erase or program operation is performed to the other bank.

Note: *Write-while-write operations are not allowed. As an example, It is not possible to perform an erase operation on one bank while programming the other one.*

Read from bank 1 while page erasing in bank 2 (or vice versa)

While executing a program code from bank 1, it is possible to perform a page erase operation on bank 2 (and vice versa). Follow the procedure below:

1. Check that no Flash memory operation is ongoing by checking the BSY bit in the *Flash status register (FLASH_SR)* (BSY is active when erase/program operation is ongoing in bank 1 or bank 2).
2. Set PER bit, PSB to select the page and BKER to select the bank in the *Flash control register (FLASH_CR)*.
3. Set the STRT bit in the FLASH_CR register.
4. Wait for the BSY bit to be cleared (or use the EOP interrupt).

Read from bank 1 while mass erasing bank 2 (or vice versa)

While executing a program code from bank 1, it is possible to perform a mass erase operation on bank 2 (and vice versa). Follow the procedure below:

1. Check that no Flash memory operation is ongoing by checking the BSY bit in the *Flash status register (FLASH_SR)* (BSY is active when erase/program operation is on going in bank 1 or bank 2).
2. Set MER1 or MER2 to in the *Flash control register (FLASH_CR)*.
3. Set the STRT bit in the FLASH_CR register.
4. Wait for the BSY bit to be cleared (or use the EOP interrupt).

Read from bank 1 while programming bank 2 (or vice versa)

While executing a program code from bank 1, it is possible to perform a program operation on the bank 2. (and vice versa). Follow the procedure below:

1. Check that no Flash memory operation is ongoing by checking the BSY bit in the *Flash status register (FLASH_SR)* (BSY is active when erase/program operation is on going on bank 1 or bank 2).
2. Set the PG bit in the *Flash control register (FLASH_CR)*.
3. Perform the data write operations at the desired address memory inside the main memory block or OTP area.
4. Wait for the BSY bit to be cleared (or use the EOP interrupt).

3.4 FLASH option bytes

3.4.1 Option bytes description

The option bytes are configured by the end user depending on the application requirements. As a configuration example, the watchdog may be selected in hardware or software mode (refer to [Section 5.4.2: Option bytes programming](#)).

A double word is split up as follows in the option bytes:

Table 10. Option byte format

| 63-24 | 23-16 | 15-8 | 7-0 | 31-24 | 23-16 | 15-8 | 7-0 |
|----------------------------|----------------------------|----------------------------|----------------------------|---------------|---------------|---------------|---------------|
| Complemented option byte 3 | Complemented option byte 2 | Complemented option byte 1 | Complemented option byte 0 | Option byte 3 | Option byte 2 | Option byte 1 | Option byte 0 |

The organization of these bytes inside the information block is as shown in [Table 31: Option byte organization](#).

The option bytes can be read from the memory locations listed in [Table 31: Option byte organization](#) or from the Option byte registers:

- *Flash option register (FLASH_OPTR)*
- *Flash PCR OP1 Start address register (FLASH_PCROP1SR)*
- *Flash PCR OP1 End address register (FLASH_PCROP1ER)*
- *Flash WRP area A address register (FLASH_WRP1AR)*
- *Flash WRP area B address register (FLASH_WRP1BR)*
- *Flash PCR OP2 Start address register (FLASH_PCROP2SR)*
- *Flash PCR OP2 End address register (FLASH_PCROP2ER)*
- *Flash Bank 2 WRP area A address register (FLASH_WRP2AR)*
- *Flash Bank 2 WRP area B address register (FLASH_WRP2BR)*.

Table 11. Option byte organization

| BANK | Address | [63:56] | [55:48] | [47:40] | [39:32] | [31:24] | [23:16] | [15:8] | [7:0] |
|--------|----------|-------------------------|------------------------|---------------------------------|-------------------------|-------------------------|------------------------|--------------------------------|-------------------------|
| Bank 1 | 1FFF7800 | USER.OPT | | | RDP | USER.OPT | | | RDP |
| | 1FFF7808 | Unused | | Unused and PCROP1_STRT[14:0] | | Unused | | Unused and PCROP1_END[14:0] | |
| | 1FFF7810 | PCROP_RDP and Unused | | Unused and PCROP1_STRT[14:0] | | PCROP_RDP and Unused | | Unused and PCROP1_END[14:0] | |
| | 1FFF7818 | Unused | WRP1A _END [6:0] | Unused | WRP1A _STRT [6:0] | Unused | WRP1A _END [6:0] | Unused | WRP1A _STRT [6:0] |
| | 1FFF7820 | Unused | WRP2A _END [6:0] | Unused | WRP2A _STRT [6:0] | Unused | WRP2A _END [6:0] | Unused | WRP2A _STRT [6:0] |
| | 1FFF7828 | Unused | BOOT LOCK | Unused | SEC SIZE1 | Unused | BOOT LOCK | Unused | SEC SIZE1 |

Table 11. Option byte organization (continued)

| BANK | Address | [63:56] | [55:48] | [47:40] | [39:32] | [31:24] | [23:16] | [15:8] | [7:0] |
|--------|---------|---------|------------------------|---------|---------------------------------|---------|------------------------|--------|---------------------------------|
| Bank 2 | 1FFF800 | Unused | | | | | | | |
| | 1FFF808 | Unused | | | Unused and PCROP2_STRT[14:0] | Unused | | | Unused and PCROP2_STRT[14:0] |
| | 1FFF810 | Unused | | | Unused and PCROP2_END[14:0] | Unused | | | Unused and PCROP2_END[14:0] |
| | 1FFF818 | Unused | WRP2B _END [6:0] | Unused | WRP2B _STRT [6:0] | Unused | WRP2B _END [6:0] | Unused | WRP2B _STRT [6:0] |
| | 1FFF820 | Unused | WRP2B _END [6:0] | Unused | WRP2B _STRT [6:0] | Unused | WRP2B _END [6:0] | Unused | WRP2B _STRT [6:0] |
| | 1FFF828 | Unused | | | SEC _SIZE2 | Unused | | | SEC _SIZE2 |

User and read protection option bytes

Flash memory address: 0x1FFF 7800

ST production value: 0xFFEF F8AA

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|-----------|----------------|-----------|----------|-------------|---------|--------|----------|------|------|---------|------------|-----------|---------|----|
| Res. | IRH_IN | NRST_MODE[1:0] | nBOOT0 | nSWBOOT0 | CCM9RAM_RST | SRAM_PE | nBOOT1 | DBANK | Res. | BFB2 | WWDG_SW | WWGD_STDBY | WWGD_STOP | WWGD_SW | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | nRST_SHDW | nRST_STDBY | nRST_STOP | Res. | BORLEV[2:0] | | | RDP[7:0] | | | | | | | |
| | r | r | r | | r | r | r | r | r | r | r | r | r | r | r |

Bit 31 Reserved, must be kept at reset value.

Bit 30 **IRH_IN**: Internal reset holder for PG10

- 0: IRH disabled
- 1: IRH enabled

Bits 29:28 **NRST_MODE[1:0]**: PG10 pad mode

- 00: Reset Input/Output
- 01: Reset Input only
- 10: GPIO
- 11: Reset Input/Output

Bit 27 **nBOOT0**: nBOOT0 option bit

- 0: nBOOT0 = 0
- 1: nBOOT0 = 1

Bit 26 **nSWBOOT0**: Software BOOT0

- 0: BOOT0 taken from the option bit nBOOT0
- 1: BOOT0 taken from PB8/BOOT0 pin

- Bit 25 **CCMSRAM_RST**: CCM SRAM erase when system reset
0: CCM SRAM erased when a system reset occurs
1: CCM SRAM is not erased when a system reset occurs
- Bit 24 **SRAM_PE**: SRAM1 and CCM SRAM parity check enable
0: SRAM1 and CCM SRAM parity check enable
1: SRAM1 and CCM SRAM parity check disable
- Bit 23 **nBOOT1**: Boot configuration
Together with the BOOT0 pin, this bit selects boot mode from the Flash main memory, SRAM1 or the System memory. Refer to [Section 2.6: Boot configuration](#).
- Bit 22 **DBANK**:
0: Single bank mode with 128 bits data read width
1: Dual bank mode with 64 bits data
This bit can be written only when PC ROP1/2 is disabled.
- Bit 21 Reserved, must be kept at reset value.
- Bit 20 **BFB2**: Dual-bank boot
0: Dual-bank boot disable
1: Dual-bank boot enable
- Bit 19 **WWDG_SW**: Window watchdog selection
0: Hardware window watchdog
1: Software window watchdog
- Bit 18 **IWDG_STDBY**: Independent watchdog counter freeze in Standby mode
0: Independent watchdog counter is frozen in Standby mode
1: Independent watchdog counter is running in Standby mode
- Bit 17 **IWDG_STOP**: Independent watchdog counter freeze in Stop mode
0: Independent watchdog counter is frozen in Stop mode
1: Independent watchdog counter is running in Stop mode
- Bit 16 **IWDG_SW**: Independent watchdog selection
0: Hardware independent watchdog
1: Software independent watchdog
- Bit 15 Reserved, must be kept at reset value.
- Bit 14 **nRST_SHDW**:
0: Reset generated when entering the Shutdown mode
1: No reset generated when entering the Shutdown mode
- Bit 13 **nRST_STDBY**:
0: Reset generated when entering the Standby mode
1: No reset generated when entering the Standby mode
- Bit 12 **nRST_STOP**:
0: Reset generated when entering the Stop mode
1: No reset generated when entering the Stop mode

Bit 11 Reserved, must be kept at reset value.

Bits 10:8 BOR_LEV: BOR reset Level

These bits contain the VDD supply level threshold that activates/releases the reset.

000: BOR Level 0. Reset level threshold is around 1.7 V

001: BOR Level 1. Reset level threshold is around 2.0 V

010: BOR Level 2. Reset level threshold is around 2.2 V

011: BOR Level 3. Reset level threshold is around 2.5 V

100: BOR Level 4. Reset level threshold is around 2.8 V

Bits 7:0 RDP: Read protection level

0xAA: Level 0, read protection not active

0xCC: Level 2, chip read protection active

Others: Level 1, memories read protection active

PCROP1 Start address option bytes

Flash memory address: 0x1FFF 7808

Reset value: 0xFFFF FFFF (ST production value)

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|-------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | B | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | PCROP1_STRT[14:0] | | | | | | | | | | | | | | |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

Bits 31:15 Reserved, must be kept at reset value.

Bits 14:0 PCROP1_STRT[14:0]: PCROP area start offset

DBANK=1

PCROP1_STRT contains the first double-word of the PCROP area for bank1.

DBANK=0

PCROP1_STRT contains the first 2xdouble-word of the PCROP area for all memory.

PCROP1 End address option bytes

Flash memory address: 0x1FFF 7810

Reset value: 0x00FF 0000 (ST production value)

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-----------|------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| PCROP_RDP | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| r | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | B | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | PCROP1_END[14:0] | | | | | | | | | | | | | | |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

Bit 31 **PCROP_RDP**: PC ROP area preserved when RDP level decreased

This bit is set only. It is reset after a full mass erase due to a change of RDP from Level 1 to Level 0.

0: PC ROP area is not erased when the RDP level is decreased from Level 1 to Level 0.

1: PC ROP area is erased when the RDP level is decreased from Level 1 to Level 0 (full mass erase).

Bits 30:15 Reserved, must be kept at reset value.

Bits 14:0 **PCROP1_END[14:0]**: Bank 1 PCROP area end offset

DBANK=1

PCROP1_END contains the last double-word of the bank 1 PCROP area.

DBANK=0

PCROP1_END contains the last 2x double-word PC ROP area for all memory.

WRP1 Area A address option bytes

Flash memory address: 0x1FFF 7818

Reset value: 0xFF00 FFFF (ST production value)

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----------------|----|----|----|----|----|--|--|--|--|--|--|--|--|
| Res | WRP1A_END[6:0] | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | |
| Res | WRP1A_STRT[6:0] | | | | | | | | | | | | | |
| | | | | | | | | | | r | r | r | r | r | r | | | | | | | | |

Bits 31:23 Reserved, must be kept at reset value.

Bits 22:16 **WRP1A_END[6:0]**: WRP first area "A" end offset

DBANK=1

WRP1A_END contains the last page of WRP first area in bank1.

DBANK=0

WRP1A_END contains the last page of WRP first area for all memory.

Bits 15:7 Reserved, must be kept at reset value.

Bits 6:0 **WRP1A_STRT[6:0]**: WRP first area "A" start offset

DBANK=1

WRP1A_STRT contains the first page of WRP first area for bank1.

DBANK=0

WRP1A_STRT contains the first page of WRP first area for all memory.

WRP2 Area A address option bytes

Flash memory address: 0x1FFF7820

Reset value: 0xFF00 FFFF (ST production value)

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-----------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | WRP2A_END[6:0] |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | WRP2A_STRT[6:0] |
| | | | | | | | | | | | | | | | |

Bits 31:23 Reserved, must be kept at reset value.

Bits 22:16 **WRP2A_END[6:0]**: WRP second area "B" end offset

DBANK=1

WRP2A_END contains the last page of the WRP second area for bank1.

DBANK=0

WRP2A_END contains the last page of the WPR second area for all memory.

Bits 15:7 Reserved, must be kept at reset value.

Bits 6:0 **WRP2A_STRT[6:0]**: WRP second area start offset

DBANK=1

WRP2A_STRT contains the last page of the WRP second area for bank1.

DBANK=0

WRP2A_STRT contains the last page of the WPR second area for all memory.

Securable memory area Bank 1 option bytes

Flash memory address: 0x1FFF7828

Reset value: 0xFF00 FF00 (ST production value)

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|----------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. |
| | | | | | | | | | | | | | | | r |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | SEC_SIZE1[7:0] |
| | | | | | | | | | | | | | | | |

Bits 31:17 Reserved, must be kept at reset value.

Bit 16 **BOOT_LOCK**: used to force boot from user area

0: Boot based on the padoption bit configuration

1: Boot forced from Main Flash memory

Bits 15:8 Reserved, must be kept at reset value.

Bits 7:0 **SEC_SIZE1[7:0]**: Securable memory area size

Contains the number of Securable Flash memory pages

PCROP2 Start address option bytes

Flash memory address: 0x1FFFF808

Reset value: 0xFFFF FFFF (ST production value)

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|-------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | PCROP2_STRT[14:0] | | | | | | | | | | | | | | |
| | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

Bits 31:15 Reserved, must be kept at reset value.

Bits 14:0 PCROP2_STRT[14:0]: PCROP area start offset

DBANK=1

PCROP2_STRT contains the first double-word of the PCROP area for bank 2.

DBANK=0

PCROP2_STRT contains the first double-word PCROP area for all memory.

PCROP2 End address option bytes

Flash memory address: 0x1FFF F810

Reset value: 0x00FF 0000 (ST production value)

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | PCROP2_END[14:0] | | | | | | | | | | | | | | |
| | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

Bits 31:15 Reserved, must be kept at reset value.

Bits 14:0 PCROP2_END[14:0]: PCROP area end offset

DBANK=1

PCROP2_END contains the last double-word of the PCROP area for bank 2.

DBANK=0

PCROP2_END contains the last 2xdouble-word of the PCROP area for all the memory.

WRP1 Area B address option bytes

Flash memory address: 0x1FFF F818

Reset value: 0xFF00 FFFF (ST production value)

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-----------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | WRP1B_END[6:0] |
| | | | | | | | | | | | r | r | r | r | r |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | WRP1B_STRT[6:0] |
| | | | | | | | | | | | r | r | r | r | r |

Bits 31:23 Reserved, must be kept at reset value.

Bits 22:16 **WRP1B_END[6:0]**: WRP first area "B" end offset

DBANK=1

WRP1B_END contains the last page of the WRP first area for bank2.

DBANK=0

WRP1B_END contains the last page of the WRP third area for all memory.

Bits 15:7 Reserved, must be kept at reset value.

Bits 6:0 **WRP1B_STRT[6:0]**: WRP first area "B" start offset

DBANK=1

WRP1B_STRT contains the first page of the WRP first area for bank2.

DBANK=0

WRP1B_STRT contains the first page of the WRP third area for all memory.

WRP2 Area B address option bytes

Flash memory address: 0x1FFF F820

Reset value: 0xFF00 FFFF (ST production value)

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. |
| | | | | | | | | | | | r | r | r | r | r |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. |
| | | | | | | | | | | | r | r | r | r | r |

Bits 31:23 Reserved, must be kept at reset value.

- Bits 22:16 **WRP2B_END[6:0]**: WRP second area "B" end offset
 DBANK=1
 WRP2B_END contains the last page of the WRP second area for bank2.
 DBANK=0
 WRP2B_END contains the last page of the WRP fourth area for all memory.
- Bits 15:7 Reserved, must be kept at reset value.
- Bits 6:0 **WRP2B_STRT[6:0]**: WRP second area "B" start offset
 DBANK=1
 WRP2B_STRT contains the first page of the WRP second area for bank2.
 DBANK=0
 WRP2B_STRT contains the first page of the WRP second area for all memory.

Securable memory area Bank 2 option bytes

Flash memory address: 0x1FFF F828

Reset value: 0xFF00 FF00 (ST production value)

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|----------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | B | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | r | r | r | r | r | r | r | r |
| SEC_SIZE2[7:0] | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

Bits 31:8 Reserved, must be kept at reset value.

Bits 7:0 **SEC_SIZE2[7:0]**: Securable memory area size contains the number of Securable Flash memory pages.

3.4.2 Option bytes programming

After reset, the options related bits in the [Flash control register \(FLASH_CR\)](#) are write-protected. To run any operation on the option bytes page, the option lock bit OPTLOCK in the [Flash control register \(FLASH_CR\)](#) must be cleared. The following sequence is used to unlock this register:

1. Unlock the FLASH_CR with the LOCK clearing sequence (refer to [Unlocking the Flash memory](#)).
2. Write OPTKEY1 = 0x08192A3B in the [Flash option key register \(FLASH_OPTKEYR\)](#).
3. Write OPTKEY2 = 0x4C5D6E7F in the FLASH_OPTKEYR register.

The user options can be protected against unwanted erase/program operations by setting the OPTLOCK bit by software.

Note: If LOCK is set by software, OPTLOCK is automatically set to.

Modifying user options

The option bytes are programmed differently from a main memory user address. It is not possible to modify independently user options of bank 1 or bank 2. The users Options of the bank 1 are modified first.

To modify the user options value, follow the procedure below:

1. Check that no Flash memory operation is on going by checking the BSY bit in the *Flash status register (FLASH_SR)*.
2. Clear OPTLOCK option lock bit with the clearing sequence described above.
3. Write the desired options value in the options registers: *Flash option register (FLASH_OPTR)*, *Flash PCROP1 Start address register (FLASH_PCR0P1SR)*, *Flash PCROP1 End address register (FLASH_PCR0P1ER)*, *Flash WRP area A address register (FLASH_WRP1AR)*, *Flash WRP area B address register (FLASH_WRP1BR)*, *Flash PCR0P2 Start address register (FLASH_PCR0P2SR)*, *Flash PCR0P2 End address register (FLASH_PCR0P2ER)*, *Flash Bank 2 WRP area A address register (FLASH_WRP2AR)*, *Flash Bank 2 WRP area B address register (FLASH_WRP2BR)*.
4. Set the Options Start bit OPTSTRT in the *Flash control register (FLASH_CR)*.
5. Wait for the BSY bit to be cleared.

Note:

Any modification of the value of one option is automatically performed by erasing both user option bytes pages first (bank 1 and bank 2) and then programming all the option bytes with the values contained in the flash option registers.

Option byte loading

After the BSY bit is cleared, all new options are updated into the Flash but they are not applied to the system. They have effect on the system when they are loaded. Option bytes loading (OBL) is performed in two cases:

- when OBL_LAUNCH bit is set in the *Flash control register (FLASH_CR)*.
- after a power reset (BOR reset or exit from Standby/Shutdown modes).

Option byte loader performs a read of the options block and stores the data into internal option registers. These internal registers configure the system and cannot be read with software. Setting OBL_LAUNCH generates a reset so the option byte loading is performed under system reset.

Each option bit has also its complement in the same double word. During option loading, a verification of the option bit and its complement allows to check the loading has correctly taken place.

During option byte loading, the options are read by double word with ECC. If the word and its complement are matching, the option word/byte is copied into the option register.

If the comparison between the word and its complement fails, a status bit OPTVERR is set. Mismatch values are forced into the option registers:

- For USR/OPT option, the value of mismatch is all options at '1', except for BOR_lev which is "000" (lowest threshold)
- For WRP option, the value of mismatch is the default value "No protection"
- For RDP option, the value of mismatch is the default value "Level 1"
- For PCROP, the value of mismatch is "all memory protected"

On system reset rising, internal option registers are copied into option registers which can be read and written by software (FLASH_OPTR, FLASH_PCR0P1/2SR,

FLASH_PCRP1/2ER, FLASH_WRP1/2AR, FLASH_WRP1/2BR). These registers are also used to modify options. If these registers are not modified by user, they reflects the options states of the system. See [Section : Modifying user options](#) for more details.

Activating dual bank mode (switching from DBANK=0 to DBANK=1)

When switching from one Flash mode to another (for example from single to dual bank) it is recommended to execute the code from the SRAM or use the bootloader. To avoid reading corrupted data from the Flash when the memory organization is changed, any access (either CPU or DMAs) to Flash memory should be avoided before reprogramming.

- Disable Instruction/data caches and/or prefetch if they are enabled (reset PRFTEN and ICEN/DCEN bits in the FLASH_ACR register).
- Flush instruction and data cache by setting the DC RST/ICRST bits in the FLASH_ACR register.
- Set the DBANK option bit and clear all the WRP write protection (follow user option modification and option bytes loader procedure).
 - Once OBL is done with DBANK=1, perform a mass erase.
 - Start a new programming of code in 64 bits mode with DBANK=1 memory mapping
 - Set the new WRP/PCROP with DBANK=1 scheme if needed.
 - Set PRFTEN and ICEN/DCEN if needed.

The new software is ready to be run using the bank configuration.

De-activating dual bank mode (switching from DBANK=1 to DBANK=0)

When switching from one Flash mode to another (for example from single to dual bank) it is recommended to execute the code from the SRAM or use the bootloader. To avoid reading corrupted data from the Flash when the memory organization is changed, any access (either CPU or DMAs) to Flash memory should be avoided before reprogramming.

- Disable Instruction/data caches and/or prefetch if they are enabled (reset PRFTEN and ICEN/DCEN bits in the FLASH_ACR register).
- Flush instruction and data cache by setting the DC RST/ICRST bits in the FLASH_ACR register.
- Clear the DBANK option bit and all WRP write protection (follow user option modification and option bytes loader procedure).
 - Once OBL is done with DBANK=0, perform a mass erase.
 - Start a new programming of code in 128 bits mode with DBANK=0 memory mapping
 - Set the new WRP/PCROP with DBANK=0 scheme if needed. Set PRFTEN and ICEN/DCEN if needed.

The new software is ready to be run using the bank configuration.

3.5 FLASH memory protection

The Flash main memory can be protected against external accesses with the Read protection (RDP). The pages of the Flash memory can also be protected against unwanted write due to loss of program counter contexts. The write-protection (WRP) granularity is one page (2 KByte). Apart of the flash memory can also be protected against read and write from third parties (PCROP). The PCROP granularity is double word (64-bit).

3.5.1 Read protection (RDP)

The read protection is activated by setting the RDP option byte and then, by applying a system reset to reload the new RDP option byte. The read protection protects to the Flash main memory, the option bytes, the backup registers (RTC_BKPxR in the RTC) and the CCM SRAM.

Note: *If the read protection is set while the debugger is still connected through JTAG/SWD, apply a POR (power-on reset) instead of a system reset.*

There are three levels of read protection from no protection (level 0) to maximum protection or no debug (level 2).

The Flash memory is protected when the RDP option byte and its complement contain the pair of values shown in [Table 32](#).

Table 12. Flash memory read protection status

| RDP byte value | RDP complement value | Read protection level |
|-------------------------------|--|----------------------------|
| 0xAA | 0x55 | Level 0 (production value) |
| Any value except 0xAA or 0xCC | Any value (not necessarily complementary) except 0x55 and 0x33 | Level 1 |
| 0xCC | 0x33 | Level 2 |

The System memory area is read accessible whatever the protection level. It is never accessible for program/erase operation.

Level 0: no protection

Read, program and erase operations into the Flash main memory area are possible. The option bytes, the CCM SRAM and the backup registers are also accessible by all operations.

Level 1: Read protection

This is the default protection level when RDP option byte is erased. It is defined as well when RDP value is at any value different from 0xAA and 0xCC, or even if the complement is not correct.

- **User mode:** Code executing in user mode (**Boot Flash**) can access Flash main memory, option bytes, CCM SRAM and backup registers with all operations.
- **Debug, boot RAM and boot loader modes:** In debug mode or when code is running from boot RAM or boot loader, the Flash main memory, the backup registers (RTC_BKPxR in the RTC) and the CCM SRAM are totally inaccessible. In these modes, a read or write access to the Flash generates a bus error and a Hard Fault interrupt.

Caution: In case the Level 1 is configured and no PCROP area is defined, it is mandatory to set PCROP_RDP bit to 1 (full mass erase when the RDP level is decreased from Level 1 to Level 0). In case the Level 1 is configured and a PCROP area is defined, if user code needs to be protected by RDP but not by PCROP, it must not be placed in a page containing a PCROP area.

Level 2: No debug

In this level, the protection level 1 is guaranteed. In addition, the Cortex®-M4 debug port, the boot from RAM (boot RAM mode) and the boot from System memory (boot loader mode) are no more available. In user execution mode (boot FLASH mode), all operations are allowed on the Flash Main memory. On the contrary, only read operations can be performed on the option bytes.

Option bytes cannot be programmed nor erased. Thus, the level 2 cannot be removed at all: it is an irreversible operation. When attempting to modify the options bytes, the protection error flag WRPERR is set in the Flash_SR register and an interrupt can be generated.

Note: *The debug feature is also disabled under reset.*

STMicroelectronics is notable to perform analysis on defective parts on which the level 2 protection has been set.

Changing the Read protection level

It is easy to move from level 0 to level 1 by changing the value of the RDP byte to any value (except 0xCC). By programming the 0xCC value in the RDP byte, it is possible to go to level 2 either directly from level 0 or from level 1. Once in level 2, it is no more possible to modify the Read protection level.

When the RDP is reprogrammed to the value 0xAA to move from Level 1 to Level 0, a mass erase of the Flash main memory is performed if PCROP_RDP is set in the [Flash PCR OP1 End address register \(FLASH_PCROP1ER\)](#). The backup registers (RTC_BKPxR in the RTC) and the CCM SRAM are also erased. The user options except PCROP protection are set to their previous values copied from FLASH_OPTR, FLASH_WRPxyR (x=1, 2 and y=A or B). PCROP is disable. The OTP area is not affected by mass erase and remains unchanged.

If the bit PCROP_RDP is cleared in the FLASH_PCROP1ER, the full mass erase is replaced by a partial mass erase that is successive page erases in the bank where PCROP is active, except for the pages protected by PCROP. This is done in order to keep the PCROP code. If PCROP is active for both banks, both banks are erased by page erases.

Only when both banks are erased, options are re-programmed with their previous values. This is also true for FLASH_PCR0PxSR and FLASH_PCR0PxER registers ($x=1,2$).

Note: Full mass erase or partial mass erase is performed only when Level 1 is active and Level 0 requested. When the protection level is increased ($0 \rightarrow 1$, $1 \rightarrow 2$, $0 \rightarrow 2$) there is no mass erase.

To validate the protection level change, the option bytes must be reloaded through the OBL_LAUNCH bit in Flash control register.

Figure 4. Changing the read protection (RDP) level

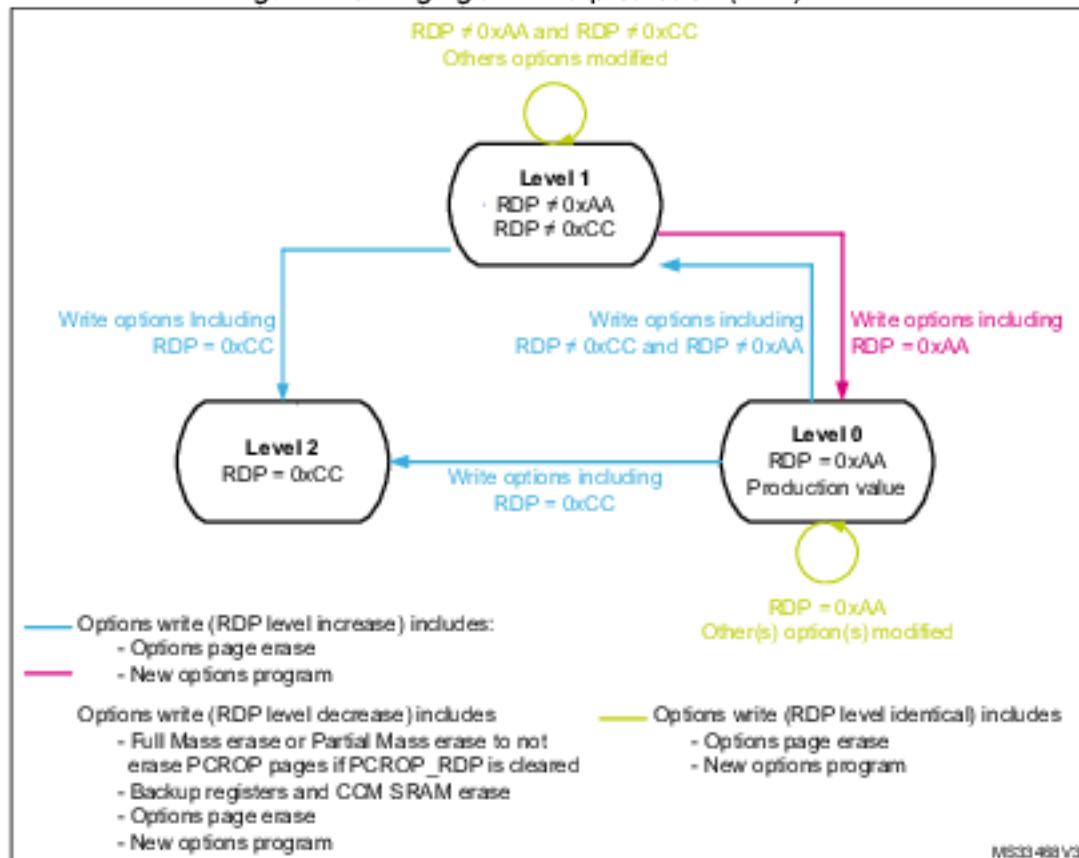


Table 13. Access status versus protection level and execution modes

| Area | Protection level | User execution (BootFromFlash) | | | Debug / BootFromRam / BootFromLoader ⁽¹⁾ | | |
|------------------------------|------------------|--------------------------------|--------------------|-------|---|--------------------|-------------------|
| | | Read | Write | Erase | Read | Write | Erase |
| Flash main memory | 1 | Yes | Yes | Yes | No | No | No ⁽³⁾ |
| | 2 | Yes | Yes | Yes | N/A | N/A | N/A |
| System memory ⁽²⁾ | 1 | Yes | No | No | Yes | No | No |
| | 2 | Yes | No | No | N/A | N/A | N/A |
| Option bytes | 1 | Yes | Yes ⁽³⁾ | Yes | Yes | Yes ⁽³⁾ | Yes |
| | 2 | Yes | No | No | N/A | N/A | N/A |

Table 13. Access status versus protection level and execution modes (continued)

| Area | Protection level | User execution (BootFromFlash) | | | Debug / BootFromRam/ BootFromLoader ⁽¹⁾ | | |
|------------------|------------------|--------------------------------|--------------------|-------|---|-------|-------------------|
| | | Read | Write | Erase | Read | Write | Erase |
| OTP | 1 | Yes | Yes ⁽⁴⁾ | N/A | No | No | N/A |
| | 2 | Yes | Yes ⁽⁴⁾ | N/A | N/A | N/A | N/A |
| Backup registers | 1 | Yes | Yes | N/A | No | No | No ⁽⁵⁾ |
| | 2 | Yes | Yes | N/A | N/A | N/A | N/A |
| CCM SRAM | 1 | Yes | Yes | N/A | No | No | No ⁽⁵⁾ |
| | 2 | Yes | Yes | N/A | N/A | N/A | N/A |

1. When the protection level 2 is active, the Debug port, the boot from RAM and the boot from system memory are disabled.
2. The system memory is only read-accessible, whatever the protection level (0, 1 or 2) and execution mode.
3. The Flash main memory is erased when the RDP option byte is programmed with all level protections disabled (0xAA).
4. OTP can only be written once.
5. The backup registers are erased when RDP changes from level 1 to level 0.
6. The CCM SRAM is erased when RDP changes from level 1 to level 0.

3.5.2 Proprietary code readout protection (PCROP)

A part of the flash memory can be protected against read and write from third parties. The protected area is execute-only: it can only be reached by the STM32 CPU, as an instruction code, while all other accesses (DMA, debug and CPU data read, write and erase) are strictly prohibited. Depending of the DBANK mode, it allows either to specify one PCROP zone per bank in dual bank mode or to specify two PCROP zones for all memory. An additional option bit (PCROP_RDP) allows to select if the PCROP area is erased or not when the RDP protection is changed from Level 1 to Level 0 (refer to [Changing the Read protection level](#)).

Each PCROP area is defined by a start page offset and an end page offset related to the physical Flash bank base address. These offsets are defined in the PCROP address registers [Flash PCROP1 Start address register \(FLASH_PCROP1SR\)](#), [Flash PCROP1 End address register \(FLASH_PCROP1ER\)](#), [Flash PCROP2 Start address register \(FLASH_PCROP2SR\)](#), [Flash PCROP2 End address register \(FLASH_PCROP2ER\)](#).

In single bank mode (DBANK=0):

- The PCROPx ($x = 1, 2$) area is defined from the address: base address + [PCROPx_STRT x 16] (included) to the address: base address + [(PCROPx_END+1) x 16] (excluded). The minimum PCROP area size is two 2 x double-words (256 bits)

In dual bank mode (DBANK=1)

- The PCROPx ($x = 1, 2$) area is defined from the address: bank "x" base address + [PCROPx_STRT x 0x8] (included) to the address: bank "x" base address + [(PCROPx_END+1) x 0x8] (excluded). The minimum PCROP area size is two double-words (128 bits).

For example, to protect by PCROP from the address 0x0806 2F80 (included) to the address 0x0807 0004 (included):

- if boot in flash is done in Bank 1, FLASH_PCROP1SR and FLASH_PCROP1ER registers must be programmed with:
 - PCROP1_STRT = 0xC5F0.
 - PCROP1_END = 0xE000.
- If the two banks are swapped, the protection must apply to bank 2, and FLASH_PCROP2SR and FLASH_PCROP2ER register must be programmed with:
 - PCROP2_STRT = 0xC5F0.
 - PCROP2_END = 0xE000.

Any read access performed through the D-bus to a PCROP protected area triggers the RDERR flag error.

Any PCROP protected address is also write protected and any write access to one of these addresses triggers WRPERR.

Any PCROP area is also erase protected. Consequently, any erase to a page in this zone is impossible (including the page containing the start address and the end address of this zone). Moreover, a software mass erase cannot be performed if one zone is PCROP protected.

For previous example, due to erase by page, all pages from page 0x62 to 0x70 are protected in case of page erase. (All addresses from 0x0806 2000 to 0x0807 0FFF can't be erased).

Deactivation of PCROP can only occurs when the RDP is changing from level 1 to level 0. If the user options modification tries to clear PCROP or to decrease the PCROP area, the options programming is launched but PCROP area stays unchanged. On the contrary, it is possible to increase the PCROP area.

When option bit PCROP_RDP is cleared, when the RDP is changing from level 1 to level 0, Full Mass Erase is replaced by Partial Mass Erase in order to keep the PCROP area (refer to [Changing the Read protection level](#)). In this case, PCROP1/2_STRT and PCROP1/2_END are also not erased.

Note: *It is recommended to align PCROP area with page granularity when using PCROP_RDP, or to leave free the rest of the page where PCROP zone starts or ends.*

Table 14. PCROP protection⁽¹⁾

| PCROPx registers values (x = 1,2) | PCROP protection area |
|---|--|
| PCROPx_offset strt > PCROPx_offset_end | No PCROP area. |
| PCROPx_offset strt < PCROPx_offset_end | The area between PCROPx_offset strt and PCROPx_offset_end is protected. It is possible to write: <ul style="list-style-type: none"> – PCROPx_offset strt with a lower value – PCROPx_offset_end with a higher value. |

- When DBANK=1, the minimum PCROP area size is 2xdouble words: PCROPx_offset_start and PCROPx_offset_end.

When DBANK=0, the minimum PCROP area size is 2x(2xdouble words): PCROPx_offset_start and PCROPx_offset_end.

When DBANK=1, it is the user's responsibility to make sure no overlapping occurs on the PCROP zones.

3.5.3 Write protection (WRP)

The user area in Flash memory can be protected against unwanted write operations.

Depending on the DBANK option bit configuration, it allows either to specify:

- In single bank mode (DBANK=0): four write-protected (WRP) areas can be defined in each bank, with page size (4 KByte) granularity.
- In dual bank mode (DBANK=1): two write-protected (WRP) areas can be defined in each bank, with page (2 KByte) granularity.

Each area is defined by a start page offset and an end page offset related to the physical Flash bank base address. These offsets are defined in the WRP address registers: *Flash WRP area A address register (FLASH_WRP1AR)*, *Flash WRP area B address register (FLASH_WRP1BR)*, *Flash Bank 2 WRP area A address register (FLASH_WRP2AR)*, *Flash Bank 2 WRP area B address register (FLASH_WRP2BR)*.

Dual bank mode (DBANK=1)

The bank "x" WRP "y" area ($x=1,2$ and $y=A,B$) is defined from the address: *Bank "x" Base address + [WRPx_y_START x 0x800]* (included) to the address: *Bank "x" Base address + [(WRPx_y_END+1) x 0x800]* (excluded).

Single Bank mode (DBANK=0)

The WRPx "y" area ($x=1,2$ and $y=A,B$) is defined from the address: *Base address + [WRPx_y_START x 0x1000]* (included) to the address: *Base address + [(WRPx_y_END+1) x 0x1000]* (excluded).

For example, to protect by WRP from the address 0x0806 2800 (included) to the address 0x0807 07FF (included):

- if boot in flash is done in Bank 1, FLASH_WRP1AR register must be programmed with:
 - WRP1A_START = 0x62.
 - WRP1A_END = 0x70.
 WRP1B_START and WRP1B_END in FLASH_WRP1BR can be used instead (area "B" in Bank 1).
- If the two banks are swapped, the protection must apply to bank 2, and FLASH_WRP2AR register must be programmed with:
 - WRP2A_START = 0x62.
 - WRP2A_END = 0x70.
 - WRP2A_START = 0xC5.
 - WRP2A_END = 0xE0.
 WRP2B_START and WRP2B_END in FLASH_WRP2BR can be used instead (area "B" in Bank 2).

When WRP is active, it cannot be erased or programmed. Consequently, a software mass erase cannot be performed if one area is write-protected.

If an erase/program operation to a write-protected part of the Flash memory is attempted, the write protection error flag (WRPERR) is set in the FLASH_SR register. This flag is also set for any write access to:

- OTP area
- part of the Flash memory that can never be written like the ICP
- PCROP area.

Note: When the memory read protection level is selected (RDP level = 1), it is not possible to program or erase Flash memory if the CPU debug features are connected (JTAG or single wire) or boot code is being executed from RAM or System flash, even if WRP is not activated.

Note: To validate the WRP options, the option bytes must be reloaded through the OBL_LAUNCH bit in Flash control register.

Note: When DBANK=0, it is the user's responsibility to make sure that no overlapping occurs on the WRP zone.

Table 15. WRP protection

| WRP registers values (x=1/2 y= A/B) | WRP protection area |
|---|---|
| WRPx _y _STRT = WRPx _y _END | Page WRP _{x_y} is protected. |
| WRPx _y _STRT > WRPx _y _END | No WRP area. |
| WRPx _y _STRT < WRPx _y _END | The pages from WRP _{x_y} _STRT to WRP _{x_y} _END are protected. |

3.5.4 Securable memory area

The Securable memory area defines an area of code which can be executed only once at boot, and never again unless a new reset occurs.

The main purpose of the Securable memory area is to protect a specific part of Flash memory against undesired access. This is a mean to isolate first stage firmware boot together with its sensitive assets (keys) from the rest of the application code. This allows implementing root of trust solution such as secure boot. Securable memory area is located in the Main Flash memory. It is dedicated to executing trusted code. When not secured, the Securable memory behaves like the remainder of Main Flash memory. When secured (the SEC_PROT1 (or SEC_PROT2) bit of the FLASH_CR register set), any attempt to program or erase in a secure memory area generates a write protection error (WRPERR flag is set) and any attempt to read from it generates a read error (RDERR flag is set).

The size of the Securable memory area is defined by the SEC_SIZE1[7:0] (or SEC_SIZE2[7:0]) bitfield of the FLASH_SEC1R (or FLASH_SEC2R) register. It can be modified only in RDP Level 0. Its content is erased upon changing from RDP Level 1 to Level 0, even if it overlaps with PCROP pages.

The securable memory area is defined:

In case of dual bank configuration (DBANK=1):

from the address: Bank1 base address (included) to the address: Bank1 base address + (0x800 * SEC_SIZE1) (excluded) and from the address: Bank2 base address (included) to the address: Bank2 base address + (0x800 * SEC_SIZE2) (excluded).

In case of single bank configuration DBANK=0:

from the address: Bank base address (included) to the address: Bank base address + (0x1000 * SEC_SIZE1) (excluded).

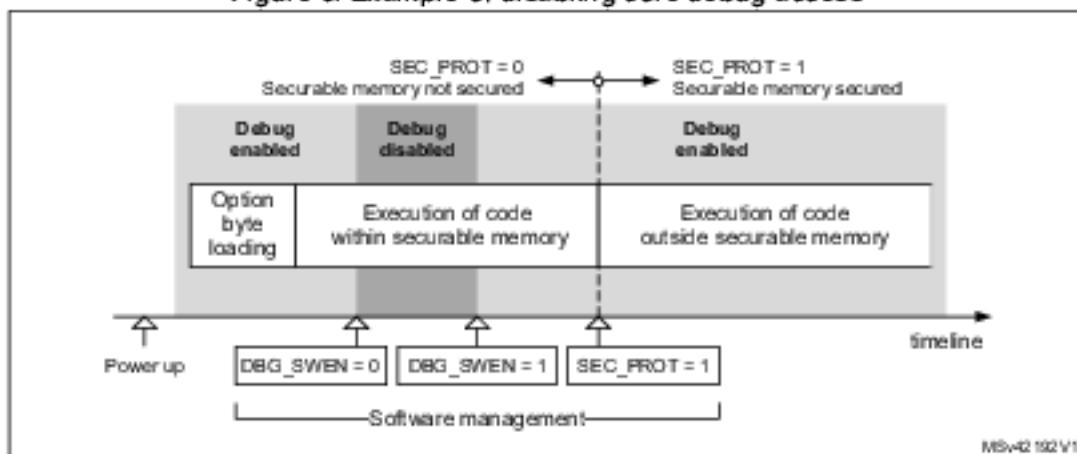
3.5.5 Disabling core debug access

For executing sensitive code or manipulating sensitive data in Securable memory area, the debug access to the core can temporarily be disabled.

In RDP level 2, the debugger is disabled by hardware, but in other RDP levels, the debugger can be disabled by software using the bit DBG_SWEN in the FLASH_ACR register.

Figure 11 gives an example of managing DBG_SWEN and SEC PROT bits.

Figure 5. Example of disabling core debug access



3.5.6 Forcing boot from Flash memory

To increase the security and establish a chain of trust, the BOOT_LOCK option bit of the FLASH_SEC1R/FLASH_SEC2R register allows forcing the system to boot from the Main Flash memory regardless the other boot options. It is always possible to set the BOOT_LOCK bit. However, it is possible to reset it only when:

- RDP is set to Level 0, or
- RDP is set to Level 1, while Level 0 is requested and a full mass-erase is performed.

3.6 FLASH interrupts

Table 16. Flash interrupt request

| Interrupt event | Event flag | Event flag/interrupt clearing method | Interrupt enable control bit |
|------------------|----------------------|--------------------------------------|------------------------------|
| End of operation | EOP ⁽¹⁾ | Write EOP=1 | EOPIE |
| Operation error | OPERR ⁽²⁾ | Write OPERR=1 | ERRIE |
| Read error | RDERR | Write RDERR=1 | RDERRIE |
| ECC correction | ECCC | Write ECCC=1 | ECCCIE |

1. EOP is set only if EOPIE is set.

2. OPERR is set only if ERRIE is set.

3.7 FLASH registers

3.7.1 Flash access control register (FLASH_ACR)

Address offset: 0x00

Reset value: 0x0004 0601

Access: no wait state, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|----------|--------|-------|-------|------|------|--------|------|------|------|------|------|------|------|--------------|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | SLEEP_PD | RUN_PD | DCRST | ICRST | OCEN | ICEN | PRFTEN | Res. | LATENCY[3:0] |
| rw | rw | rw | rw | rw | rw | rw | rw | | | | | rw | rw | rw | rw |

Bits 31:19 Reserved, must be kept at reset value.

Bit 18 **DBG_SWEN**: Debug software enable

SW may use this bit to enable/disable the debugger.

0: Debugger disabled

1: Debugger enabled

Bits 17:15 Reserved, must be kept at reset value.

Bit 14 **SLEEP_PD**: Flash Power-down mode during Sleep or Low-power sleep mode

This bit determines whether the flash memory is in Power-down mode or Idle mode when the device is in Sleep or Low-power sleep mode.

0: Flash in Idle mode during Sleep and Low-power sleep modes

1: Flash in Power-down mode during Sleep and Low-power sleep modes

Caution: The flash must not be put in power-down while a program or an erase operation is on-going.

Bit 13 **RUN_PD**: Flash Power-down mode during Run or Low-power run mode

This bit is write-protected with FLASH_PDKEYR.

This bit determines whether the flash memory is in Power-down mode or Idle mode when the device is in Run or Low-power run mode. The flash memory can be put in power-down mode only when the code is executed from RAM. The Flash must not be accessed when RUN_PD is set.

0: Flash in Idle mode

1: Flash in Power-down mode

Caution: The flash must not be put in power-down while a program or an erase operation is on-going.

Bit 12 **DCRST**: Data cache reset

0: Data cache is not reset

1: Data cache is reset

This bit can be written only when the data cache is disabled.

- Bit 11 **ICRST**: Instruction cache reset
 0: Instruction cache is not reset
 1: Instruction cache is reset
 This bit can be written only when the instruction cache is disabled.
- Bit 10 **DCEN**: Data cache enable
 0: Data cache is disabled
 1: Data cache is enabled
- Bit 9 **ICEN**: Instruction cache enable
 0: Instruction cache is disabled
 1: Instruction cache is enabled
- Bit 8 **PRFTEN**: Prefetch enable
 0: Prefetch disabled
 1: Prefetch enabled
- Bits 7:4 Reserved, must be kept at reset value.
- Bits 3:0 **LATENCY[3:0]**: Latency
 These bits represent the ratio of the SYSCLK (system clock) period to the Flash access time.
 0000: Zero wait state
 0001: One wait state
 0010: Two wait states
 0011: Three wait states
 0100: Four wait states
 ...1111: Fifteen wait states

3.7.2 Flash Power-down key register (FLASH_PDKEYR)

Address offset: 0x04

Reset value: 0x0000 0000

Access: no wait state, word access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| PDKEYR[31:16] | | | | | | | | | | | | | | | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PDKEYR[15:0] | | | | | | | | | | | | | | | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

Bits 31:0 **PDKEYR[31:0]**: Power-down in Run mode Flash key

The following values must be written consecutively to unlock the RUN_PD bit in FLASH_ACR:

PDKEY1: 0x04 15 2637

PDKEY2: 0xFAFB FCFD

3.7.3 Flash key register (FLASH_KEYR)

Address offset: 0x08

Reset value: 0x0000 0000

Access: no wait state, word access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| KEYR[31:16] | | | | | | | | | | | | | | | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| KEYR[15:0] | | | | | | | | | | | | | | | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

Bits 31:0 KEYR[31:0]: Flash key

The following values must be written consecutively to unlock the FLASH_CTR register allowing flash programming/erasing operations:

KEY1: 0x4567 0123

KEY2: 0xCDEF 89AB

3.7.4 Flash option key register (FLASH_OPTKEYR)

Address offset: 0x0C

Reset value: 0x0000 0000

Access: no wait state, word access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| OPTKEYR[31:16] | | | | | | | | | | | | | | | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OPTKEYR[15:0] | | | | | | | | | | | | | | | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

Bits 31:0 OPTKEYR[31:0]: Option byte key

The following values must be written consecutively to unlock the FLASH_OPTR register allowing option byte programming/erasing operations:

KEY1: 0x0819 2A3B

KEY2: 0x4C5D 6E7F

3.7.5 Flash status register (FLASH_SR)

Address offset: 0x10

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access

| | | | | | | | | | | | | | | | | |
|-------------|-----------|------|------|------|------|-------------|-------------|------------|------------|------------|------------|-------------|------|------------|-------|--|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | |
| Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | BSY | |
| | | | | | | | | | | | | | | | r | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| OPTV ERR | RD ERR | Res. | Res. | Res. | Res. | FAST ERR | MISS ERR | PGS ERR | SIZ ERR | PGA ERR | WRP ERR | PROG ERR | Res. | OIP ERR | EOP | |
| re_w1 | rc_w1 | | | | | re_w1 | rc_w1 | re_w1 | rc_w1 | re_w1 | rc_w1 | re_w1 | | re_w1 | re_w1 | |

Bits 31:17 Reserved, must be kept at reset value.

Bit 16 BSY: Busy

This indicates that a Flash operation is in progress. This is set on the beginning of a Flash operation and reset when the operation finishes or when an error occurs.

Bit 15 OPTVERR: Option validity error

Set by hardware when the options read may not be the one configured by the user. If option haven't been properly loaded, OPTVERR is set again after each system reset.

Cleared by writing 1.

Bit 14 RDERR: PCROP read error

Set by hardware when an address to be read through the D-bus belongs to a read protected area of the flash (PCROP protection). An interrupt is generated if RDERRIE is set in FLASH_C.R.

Cleared by writing 1.

Bits 13:10 Reserved, must be kept at reset value.

Bit 9 FASTERR: Fast programming error

Set by hardware when a fast programming sequence (activated by FSTPG) is interrupted due to an error (alignment, size, write protection or data miss). The corresponding status bit (PGAERR, SIZERR, WRPERR or MISSERR) is set at the same time.

Cleared by writing 1.

Bit 8 MISSERR: Fast programming data miss error

In Fast programming mode, 32 double words must be sent to flash successfully, and the new data must be sent to the flash logic control before the current data is fully programmed. MISSERR is set by hardware when the new data is not present in time.

Cleared by writing 1.

Bit 7 PGSERR: Programming sequence error

Set by hardware when a write access to the Flash memory is performed by the code while PG or FSTPG have not been set previously. Set also by hardware when PROGERR, SIZERR, PGAERR, WRPERR, MISSERR or FASTERR is set due to a previous programming error.

Set also when trying to perform bank erase when DBANK=0.

Cleared by writing 1.

Bit 6 SIZERR: Size error

Set by hardware when the size of the access is a byte or half-word during a program or a fast program sequence. Only double word programming is allowed (consequently: word access).

Cleared by writing '1'.

Bit 5 PGAERR: Programming alignment error

Set by hardware when the data to program cannot be contained in the same 64-bit Flash memory row in case of standard programming, or if there is a change of page during fast programming.

Cleared by writing '1'.

Bit 4 WRPERR: Write protection error

Set by hardware when an address to be erased/programmed belongs to a write-protected part (by WRP, PCROP or RDP level 1) of the Flash memory.

Cleared by writing '1'.

Bit 3 PROGERR: Programming error

Set by hardware when a double-word address to be programmed contains a value different from '0xFFFF FFFF FFFF FFFF' before programming, except if the data to write is '0x0000 0000 0000 0000'.

Cleared by writing '1'.

Bit 2 Reserved, must be kept at reset value.**Bit 1 OPERR:** Operation error

Set by hardware when a Flash memory operation (program / erase) completes unsuccessfully.

This bit is set only if error interrupts are enabled (ERRIE = 1).

Cleared by writing '1'.

Bit 0 EOP: End of operation

Set by hardware when one or more Flash memory operation (programming / erase) has been completed successfully.

This bit is set only if the end of operation interrupts are enabled (EOPIE = 1).

Cleared by writing '1'.

3.7.6 Flash control register (FLASH_CR)

Address offset: 0x14

Reset value: 0xC000 0000

Access: no wait state when no Flash memory operation is on going, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | |
|------|----------|---------------------------|---------------------------|----------------------------|------|----------|--------|------|------|------|------|------|-------|----------|-------|----|
| LOCK | OPT LOCK | SEC ₂ PROT2 | SEC ₁ PROT1 | OBL ₂ LAUNCH | RD | ERR IE | EOP IE | Res. | Res. | Res. | Res. | Res. | FSTPG | OPT STRT | START | |
| | | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| MER2 | Res. | Res. | Res. | BKER | Res. | PNB[6:0] | | | | | | | | MER1 | PER | PG |
| | | | | | | | | | | | | | | | | |

Bit 31 LOCK: FLASH_CR Lock

This bit is set only. When set, the FLASH_CR register is locked. It is cleared by hardware after detecting the unlock sequence.

In case of an unsuccessful unlock operation, this bit remains set until the next system reset.

Bit 30 OPTLOCK: Options Lock

This bit is set only. When set, all bits concerning user option in FLASH_CR register and so option page are locked. This bit is cleared by hardware after detecting the unlock sequence. The LOCK bit must be cleared before doing the unlock sequence for OPTLOCK bit.

In case of an unsuccessful unlock operation, this bit remains set until the next reset.

Bit 29 SEC_PROT2: Securable memory area protection bit for bank 2.

This bit is set to lock the access to the Securable area in bank 2. It is set by software when exiting the Securable area, and can only be written once. In case DBANK=0, this bit is useless.

Bit 28 SEC_PROT1: Securable memory area protection bit for bank 1.

This bit is set to lock the access to the Securable memory area for bank 1 (or when DBANK=0). It is set by software when exiting the Securable area, and can only be written once.

Bit 27 OBL_LAUNCH: Force the option byte loading

When set to 1, this bit forces the option byte reloading. This bit is cleared only when the option byte loading is complete. It cannot be written if OPTLOCK is set.

0: Option byte loading complete

1: Option byte loading requested

Bit 26 RDERRIE: PCROP read error interrupt enable

This bit enables the interrupt generation when the RDERR bit in the FLASH_SR is set to 1.

0: PCROP read error interrupt disabled

1: PCROP read error interrupt enabled

Bit 25 ERRIE: Error interrupt enable

This bit enables the interrupt generation when the OPERR bit in the FLASH_SR is set to 1.

0: OPERR error interrupt disabled

1: OPERR error interrupt enabled

Bit 24 EOPIE: End of operation interrupt enable

This bit enables the interrupt generation when the EOP bit in the FLASH_SR is set to 1.

0: EOP interrupt disabled

1: EOP interrupt enabled

Bits 23:19 Reserved, must be kept at reset value.

Bit 18 FSTPG: Fast programming

0: Fast programming disabled

1: Fast programming enabled

Bit 17 OPTSTRT: Options modification start

This bit triggers an options operation when set.

This bit is set only by software, and is cleared when the BSY bit is cleared in FLASH_SR.

Bit 16 START: Start

This bit triggers an erase operation when set. If MER1, MER2 and PER bits are reset and the STRT bit is set, an unpredictable behavior may occur without generating any error flag. This condition should be forbidden.

This bit is set only by software, and is cleared when the BSY bit is cleared in FLASH_SR.

Bit 15 MER2: Bank 2 Mass erase

This bit triggers the bank 2 mass erase (all bank 2 user pages) when set.

Bits 14:12 Reserved, must be kept at reset value.

Bit 11 BKER: Bank erase

DBANK=1

0: Bank 1 is selected for page erase

1: Bank 2 is selected for page erase

DBANK=0

Reserved, must be at reset value.

Bit 10 Reserved, must be kept at reset value.

Bits 9:3 PNB[6:0]: Page number selection

These bits select the page to erase:

00000000: page 0

00000001: page 1

...

11111111: page 255

Bit 2 MER1: Bank 1 Mass erase

This bit triggers the bank 1 mass erase (all bank 1 user pages) when set.

Bit 1 PER: Page erase

0: page erase disabled

1: page erase enabled

Bit 0 PG: Programming

0: Flash programming disabled

1: Flash programming enabled

3.7.7 Flash ECC register (FLASH_ECCR)

Address offset: 0x18

Reset value: 0x0000 0000

Access: no wait state when no Flash memory operation is ongoing, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|----------------|-------|-------|-------|------|------|------|------------|------|----------|--------|------|------|-----------------|----|----|
| ECC0 | ECCC | ECC02 | ECC02 | Res. | Res. | Res. | ECCC IE | Res. | SYSF_ECC | BK_ECC | Res. | Res. | ADDR_ECC[18:16] | | |
| rc_w1 | rc_w1 | rc_w1 | rc_w1 | | | | rw | | r | r | | | r | r | r |
| ADDR_ECC[15:0] | | | | | | | | | | | | | | | |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

Bit 31 ECCD: ECC detection

DBANK=1

Set by hardware when two ECC errors have been detected (only if ECCC/ECCC2/ECCD/ ECC D2 are previously cleared). When this bit is set, a NMI is generated.

Cleared by writing 1.

DBANK=0

Set by hardware when two ECC errors have been detected on 64-bits LSB (bits 63:0) (only if ECCC/ECCC2/ECCD/ ECC D2 are previously cleared). When this bit is set, a NMI is generated.

Cleared by writing 1.

Bit 30 ECCC: ECC correction

DBANK=1

Set by hardware when one ECC error has been detected and corrected (only if ECCC/ECCC2/ECCD/ECCD2 are previously cleared). An interrupt is generated if ECCCIE is set.

Cleared by writing 1.

DBANK=0

Set by hardware when one ECC error has been detected and corrected on 64-bits LSB (bits 63:0) (only if ECCC/ECCC2/ECCD/ ECC D2 are previously cleared).

Cleared by writing 1.

Bit 29 ECCD2: ECC2 detection

DBANK=0

Set by hardware when two ECC errors have been detected on 64-bits MSB (bits 127:64). This bit is set (only if ECCC/ECCC2/ECCD/ECCD2 are previously cleared). When this bit is set, a NMI is generated.

Cleared by writing 1.

DBANK=1

Reserved, must be kept at reset value.

Bit 28 ECCC2: ECC correction

DBANK=0

Set by hardware when one ECC error has been detected and corrected on 64-bits MSB (bits 127:64). This bit is set (only if ECCC/ECCC2/ECCD/ECCD2 are previously cleared). An interrupt is generated if ECCCIE is set.

Cleared by writing 1.

DBANK=1

Reserved, must be kept at reset value.

Bits 27:25 Reserved, must be kept at reset value.

Bit 24 ECCCIE: ECC correction interrupt enable

0: ECCC interrupt disabled

1: ECCC interrupt enabled.

DBANK=0

This bit enables the interrupt generation when the ECCC or ECCC2 bits in the FLASH_ECCR register are set.

DBANK=1

This bit enables the interrupt generation when the ECCC bit in the FLASH_ECCR register is set.

Bit 23 Reserved, must be kept at reset value.

Bit 22 **SYSF_ECC**: System Flash ECC fail

This bit indicates that the ECC error correction or double ECC error detection is located in the System Flash.

Bit 21 **BK_ECC**: ECC fail bank

DBANK=1

This bit indicates which bank is concerned by the ECC error correction or by the double ECC error detection.

0: bank 1

1: bank 2

DBANK=0

If SYSF_ECC is 1, it indicates which bank is concerned by the ECC error

If SYSF_ECC is 0, reserved, must be kept cleared.

Bits 20:19 Reserved, must be kept at reset value.

Bits 18:0 **ADDR_ECC[18:0]**: ECC fail address

DBANK=0

This bit indicates which address in the Flash memory is concerned by the ECC error correction or by the double ECC error detection.

DBANK=1

This bit indicates which address in the bank is concerned by the ECC error correction or by the double ECC error detection.

3.7.8 Flash option register (FLASH_OPTR)

Address offset: 0x20

Reset value: 0xFXXX XXXX. Register bits are loaded with values from Flash memory at OBL.

Access: no wait state when no Flash memory operation is ongoing, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|-----------|-----------------|-----------|--------|-------------|--------------|---------|----------|-------|------|------|---------|------------|-----------|---------|
| Res. | IRHEN | NRST_MODE [1:0] | | nBOOT0 | nSW_BOOT0 | CCM_SRAM_RST | SRAM_PE | nBOOT1 | DBANK | Res. | BFB2 | WWDG_SW | IWDG_STDBY | IWDG_STOP | IWDG_SW |
| | rw | rw | rw | rw | rw | rw | rw | rw | rw | | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | nRST_SHDW | nRST_STDBY | nRST_STOP | Res. | BORLEV[2:0] | | | RDP[7:0] | | | | | | | |
| | rw | rw | rw | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Bit 31 Reserved, must be kept at reset value.

Bit 30 **IRHEN**: Internal reset holder enable bit

0: Internal resets are propagated as simple pulse on NRST pin

1: Internal resets drives NRST pin low until it is seen as low level

Bits 29:28 **NRST_MODE[1:0]**:

00: Reserved

01: Reset Input only: a low level on the NRST pin generates system reset, internal RESET not propagated to the NSRT pin

10: GPIO: standard GPIO pad functionality, only internal RESET possible

11: Bidirectional reset: NRST pin configured in reset input/output mode (legacy mode)

Bit 27 **nBOOT0**: nBOOT0 option bit

0: nBOOT0 = 0

1: nBOOT0 = 1

Bit 26 **nSWBOOT0**: Software BOOT0

0: BOOT0 taken from the option bit nBOOT0

1: BOOT0 taken from PB8/BOOT0 pin

Bit 25 **CCMSRAM_RST**: CCM SRAM Erase when system reset

0: CCM SRAM erased when a system reset occurs

1: CCM SRAM is not erased when a system reset occurs

Bit 24 **SRAM_PE**: SRAM1 and CCM SRAM parity check enable

0: SRAM1 and CCM SRAM parity check enable

1: SRAM1 and CCM SRAM parity check disable

Bit 23 **nBOOT1**: Boot configuration

Together with the BOOT0 pin, this bit selects boot mode from the Flash main memory, SRAM1 or the System memory. Refer to [Section 2.6: Boot configuration](#).

Bit 22 **DBANK**:

0: Single bank mode with 128 bits data read width

1: Dual bank mode with 64 bits data

This bit can only be written when PCROP/A/B is disabled.

Bit 21 Reserved, must be kept at reset value.

Bit 20 **BFB2**: Dual-bank boot

0: Dual-bank boot disable

1: Dual-bank boot enable

Bit 19 **WWDG_SW**: Window watchdog selection

0: Hardware window watchdog

1: Software window watchdog

Bit 18 **WDG_STDBY**: Independent watchdog counter freeze in Standby mode

0: Independent watchdog counter is frozen in Standby mode

1: Independent watchdog counter is running in Standby mode

Bit 17 **WDG_STOP**: Independent watchdog counter freeze in Stop mode

0: Independent watchdog counter is frozen in Stop mode

1: Independent watchdog counter is running in Stop mode

Bit 16 **WDG_SW**: Independent watchdog selection

- 0: Hardware Independent watchdog
- 1: Software Independent watchdog

Bit 15 Reserved, must be at reset value.

Bit 14 **nRST_SHDW**

- 0: Reset generated when entering the Shutdown mode
- 1: No reset generated when entering the Shutdown mode

Bit 13 **nRST_STDBY**

- 0: Reset generated when entering the Standby mode
- 1: No reset generated when entering the Standby mode

Bit 12 **nRST_STOP**

- 0: Reset generated when entering the Stop mode
- 1: No reset generated when entering the Stop mode

Bit 11 Reserved, must be at reset value.

Bits 10:8 **BORLEV**: BOR reset Level

These bits contain the VDD supply level threshold that activates/releases the reset.

- 000: BOR Level 0. Reset level threshold is around 1.7 V
- 001: BOR Level 1. Reset level threshold is around 2.0 V
- 010: BOR Level 2. Reset level threshold is around 2.2 V
- 011: BOR Level 3. Reset level threshold is around 2.5 V
- 100: BOR Level 4. Reset level threshold is around 2.8 V

Bits 7:0 **RDP**: Read protection level

- 0xAA: Level 0, read protection not active
- 0xCC: Level 2, chip read protection active
- Others: Level 1, memories read protection active

Note: Take care about PCROP_RDP configuration in Level 1. Refer to [Section : Level 1: Read protection](#) for more details.

3.7.9 Flash PCROP1 Start address register (FLASH_PCROP1SR)

Address offset: 0x24

Reset value: 0xFFFF XXXX

Register bits are loaded with values from Flash memory at OBL.

Access: no wait state when no Flash memory operation is ongoing, word access.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-----|-------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res | PCROP1_STRT[14:0] | | | | | | | | | | | | | | |
| IW | IW | IW | IW | IW | IW | IW | IW | IW | IW | IW | IW | IW | IW | IW | IW |

Bits 31:15 Reserved, must be kept at reset value.

Bits 14:0 PCROP1_STRT[14:0]: PCROP area start offset

DBANK=1

PCROP1_STRT contains the first double-word of the PCROP area for bank1.

DBANK=0

PCROP1_STRT contains the first 2xdouble-word of the PCROP area for all memory.

3.7.10 Flash PCROP1 End address register (FLASH_PCROP1ER)

Address offset: 0x28

Reset value: 0x0000 XXXX

Register bits are loaded with values from Flash memory at OBL.

Access: no wait state when no Flash memory operation is on going, word, half-word access.
PCROP_RDP bit can be accessed with byte access.

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| PCROP_RDP | Res. |
| # | | | | | | | | | | | | | | | | |
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PCROP1_END[14:0] | | | | | | | | | | | | | | | | |
| | rw |

Bit 31 PCROP_RDP: PCROP area preserved when RDP level decreased

This bit is set only. It is reset after a full mass erase due to a change of RDP from Level 1 to Level 0.

0: PCROP area is not erased when the RDP level is decreased from Level 1 to Level 0.

1: PCROP area is erased when the RDP level is decreased from Level 1 to Level 0 (full mass erase).

Bits 30:15 Reserved, must be kept at reset value.

Bits 14:0 PCROP1_END[14:0]: Bank 1 PCROP area end offset

DBANK=1

PCROP1_END contains the last double-word of the bank 1 PCROP area.

DBANK=0

PCROP1_END contains the last 2x double-word of the first PCROP area in all memory.

3.7.11 Flash Bank 1 WRP area A address register (FLASH_WRP1AR)

Address offset: 0x2C

Reset value: 0x00XX 00XX

Register bits are loaded with values from Flash memory at OBL.

Access: no wait state when no Flash memory operation is on going, word, half-word and byte access

Bits 31:23 Reserved, must be kept at reset value.

Bits 22:16 WRP1A-END[6:0]: WRP first area "A" end offset

DBANK=1

WRP1A-END contains the last page of WRP first area in bank1.

DBANK=0

WRP1A-END contains the last page of WRP first area for all memory.

Bits 15:7 Reserved, must be kept at reset value.

Bits 6:0 WRP1A_STRT[6:0]: WRP first area "A" start offset

DBANK= 1

WRP1A_STRT contains the first page of WRP first area for bank1.

DBANK=0

WRP1A_STRT contains the first page of WRP first area for all memory.

3.7.12 Flash Bank 1 WRP area B address register (FLASH_WRP1BR)

Address offset: 0x30

Reset value: 0x00XX 00XX

Register bits are loaded with values from Flash memory at OBL.

Access: no wait state when no Flash memory operation is ongoing, word, half-word and byte access

Bits 31:23 Reserved, must be kept at reset value.

Bits 22:16 **WRP1B_END[6:0]**: WRP second area "B" end offset
 DBANK=1
 WRP1B_END contains the last page of the WRP second area for bank1.
 DBANK=0
 WRP1B_END contains the last page of the WPR second area for all memory.

Bits 15:7 Reserved, must be kept at reset value.

Bits 6:0 **WRP1B_STRT[6:0]**: WRP second area "B" start offset
 DBANK=1
 WRP1B_STRT contains the last page of the WRP second area for bank1.
 DBANK=0
 WRP1B_STRT contains the last page of the WPR second area for all memory.

3.7.13 Flash PCROP2 Start address register (FLASH_PCROP2SR)

Address offset: 0x44

Reset value: 0xFFFF XXXX

Access: no wait state when no Flash memory operation is on going, word access.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|-------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | PCROP2_STRT[14:0] | | | | | | | | | | | | | | |
| IW | IW | IW | IW | IW | IW | IW | IW | IW | IW | IW | IW | IW | IW | IW | IW |

Bits 31:15 Reserved, must be kept at reset value.

Bits 14:0 **PCROP2_STRT[14:0]**: PCROP area start offset

DBANK=1
 PCROP2_STRT contains the first double-word of the PCROP area for bank 2.
 DBANK=0
 PCROP2_STRT contains the first double-word PCROP area for all memory.

3.7.14 Flash PCROP2 End address register (FLASH_PCROP2ER)

Address offset: 0x48

Reset value: 0x0000 XXXX

Access: no wait state when no Flash memory operation is on going, word access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | PCROP2_END[14:0] | | | | | | | | | | | | | | |
| IW | IW | IW | IW | IW | IW | IW | IW | IW | IW | IW | IW | IW | IW | IW | IW |

Bits 31:15 Reserved, must be kept at reset value.

Bits 14:0 PCROP2_END[14:0]: PCROP area end offset

DBANK=1

PCROP2_END contains the last double-word of the PCROP area for bank2.

DBANK=0

PCROP2_END contains the last 2xdouble-word of the PCROP area for all the memory.

3.7.15 Flash Bank 2 WRP area A address register (FLASH_WRP2AR)

Address offset: 0x4C

Reset value: 0x00XX 00XX

Access: no waitstate when no Flash memory operation is on going, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----------------|
| Res | WRP2A_END[6:0] |
| | | | | | | | | | | | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res | WRP2A_STRT[6:0] |
| | | | | | | | | | | | rw | rw | rw | rw | rw |

Bits 31:23 Reserved, must be kept at reset value.

Bits 22:16 WRP2A_END[6:0]: WRP first area "A" end offset

DBANK=1

WRP2A_END contains the last page of the WRP first area for bank2.

DBANK=0

WRP2A_END contains the last page of the WRP third area for all memory.

Bits 15:7 Reserved, must be kept at reset value.

Bits 6:0 WRP2A_STRT[6:0]: WRP first area "A" start offset

DBANK=1

WRP2A_STRT contains the first page of the WRP first area for bank2.

DBANK=0

WRP2A_STRT contains the first page of the WRP third area for all memory.

3.7.16 Flash Bank 2 WRP area B address register (FLASH_WRP2BR)

Address offset: 0x50

Reset value: 0x00XX 00XX

Access: no wait state when no Flash memory operation is on going, word, half-word and byte access

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-----------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | WRP2B_END[6:0] |
| | | | | | | | | | | | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | WRP2B_STRT[6:0] |
| | | | | | | | | | | | rw | rw | rw | rw | rw |

Bits 31:23 Reserved, must be kept at reset value.

Bits 22:16 WRP2B_END[6:0]: WRP second area "B" end offset

DBANK=1

WRP2B_END contains the last page of the WRP second area for bank2.

DBANK=0

WRP2B_END contains the last page of the WRP fourth area for all memory.

Bits 15:7 Reserved, must be kept at reset value.

Bits 6:0 WRP2B_STRT[6:0]: WRP second area "B" start offset

DBANK=1

WRP2B_STRT contains the first page of the WRP second area for bank2.

DBANK=0

WRP2B_STRT contains the first page of the WRP second area for all memory.

3.7.17 Flash Securable area bank1 register (FLASH_SEC1R)

Address offset: 0x70

Reset value: 0xFFFF FFXX

Access: no wait state when no Flash memory operation is on going, word, half-word and byte access

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|----------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | BOOT_LOCK |
| | | | | | | | | | | | | | | | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | SEC_SIZE1[7:0] |
| | | | | | | | | | | | rw | rw | rw | rw | rw |

Bits 31:17 Reserved, must be kept at reset value.

Bit 16 **BOOT_LOCK**: used to force boot from user Flash area.

0: Boot based on the pad/option bit configuration

1: Boot forced from Main Flash memory

Bits 15:8 Reserved, must be kept at reset value.

Bits 7:0 **SEC_SIZE1[7:0]**: sets the number of pages used in the bank 1 Securable memory area.

Securable area starts at @ 0x0800 0000 and its size is SEC_SIZE1 * page size.
This field can be changed in level0 only.

Any attempt to modify in level1 silently fails, and does not change register value.

3.7.18 Flash Securable area bank2 register (FLASH_SEC2R)

Address offset: 0x74

Reset value: 0xFFFF FFXX

Access: no wait state when no Flash memory operation is on going, word, half-word and byte access.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-----------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | rw |
| SEC_SIZE2[7:0] | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

Bits 31:8 Reserved, must be kept at reset value.

Bits 7:0 **SEC_SIZE2[7:0]**: sets the number of pages used in the bank 2 Securable memory area.

Securable area starts at @ 0x0804 0000 and its size is SEC_SIZE2 * page size.
When DBANK=0, this field is not usefull.

This field can be changed in level0 only. Any attempt to modify in level1 silently fails, and does not change register value.

3.7.19 FLASH register map

Table 17. Flash interface - register map and reset values

Table 17. Flash interface - register map and reset values (continued)

Refer to [Section 2.2 on page 81](#) for the register boundary addresses.

4 Embedded Flash memory (FLASH) for category 4 devices

4.1 Introduction

The Flash memory interface manages CPU AHB ICode and DCode accesses to the Flash memory. It implements the erase and program Flash memory operations and the read and write protection mechanisms.

The Flash memory interface accelerates code execution with a system of instruction prefetch and cache lines.

4.2 FLASH main features

- Up to 512 Kbyte of Flash memory (single bank).
- Flash memory read operations with 64 bits data width
- Page erase and mass erase

Flash memory interface features:

- Flash memory read operations
- Flash memory program/erase operations
- Read protection activated by option (RDP)
- 2 Write protection areas selected by option
- Proprietary code read protection areas defined by option
- Securable memory areas defined by option
- Prefetch on ICODE
- Instruction Cache: 32 cache lines of 4 x 64 bits on ICode (1 KB RAM)
- Data Cache: 8 cache lines of 4 x 64 bits on DCode (256B RAM)
- Error Code Correction ECC: 8 bits per 64-bit double-word
 - $8 + 64 = 72$ bits, 2 bits detection, 1 bit correction
- Option byte loader
- Low-power mode

4.3 FLASH functional description

4.3.1 Flash memory organization

The Flash memory has the following main features:

- Capacity up to 512 Kbytes (read width of 64-bits)
- 512 KB organized in one single bank for main memory
- Page size of 2 Kbyte
- 72 bits wide data read (64 bits plus 8 ECC bits)
- Page and Mass erase
- Each page is composed of 8 rows of 256 bytes
- An Information block containing:
 - System memory from which the device boots in System memory boot mode. The area is reserved for use by STMicroelectronics and contains the bootloader that is used to reprogram the Flash memory through one of the following interfaces: USART, SPI, I2C, USB. It is programmed by STMicroelectronics when the device is manufactured, and protected against spurious write/erase operations. For further details, please refer to the AN2606 available from www.st.com.
 - 1 Kbyte (128 double word) OTP (one-time programmable) bytes for user data. The OTP data cannot be erased and can be written only once. If only one bit is at 0, the entire double word cannot be written anymore, even with the value 0x0000 0000 0000.
 - Option bytes for user configuration.

The memory organization is based on a main area and an information block as shown in [Table 28](#).

Table 18. Flash module - 256/512 Kbytes organization (64 bits read width)

| Flash area | Flash memory addresses | Size (bytes) | Name |
|--|---------------------------|--------------|---------------|
| Main memory 256/512 Kbytes ⁽¹⁾ | 0x0800 0000 - 0x0800 07FF | 2 K | Page 0 |
| | 0x0800 0800 - 0x0800 0FFF | 2 K | Page 1 |
| | 0x0800 1000 - 0x0800 17FF | 2 K | Page 2 |
| | 0x0800 1800 - 0x0800 1FFF | 2 K | Page 3 |
| | - | - | - |
| | - | - | - |
| | - | - | - |
| | 0x0807 F800 - 0x0807 FFFF | 2 K | Page 255 |
| Information block | 0x1FF 0000 - 0x1FF 6FFF | 28 K | System memory |
| | 0x1FF 7000 - 0x1FF 73FF | 1 K | OTP area |
| | 0x1FF 7800 - 0x1FF 782F | 48 | Option bytes |

1. For 256-Kbyte devices: from page 0 to page 127.

4.3.2 Error code correction (ECC)

Data in Flash memory are 72-bits words: 8 bits are added per double word (64 bits). The ECC mechanism supports:

- One error detection and correction
- Two errors detection

When one error is detected and corrected, the flag ECCC (ECC correction) is set in [Flash ECC register \(FLASH_ECCR\)](#). If ECCCIE is set, an interrupt is generated.

When two errors are detected, a flag ECCD (ECC detection) is set in FLASH_ECCR register. In this case, a NMI is generated.

When an ECC error is detected, the address of the failing double word is saved in ADDR_ECC[20:0] in the FLASH_ECCR register. ADDR_ECC[2:0] are always cleared.

When ECCC or ECCD is set, ADDR_ECC is not updated if a new ECC error occurs. FLASH_ECCR is updated only when ECC flags are cleared.

Note: For a virgin data: 0xFFFF FFFF FFFF FFFF, one error is detected and corrected but two errors detection is not supported.

When an ECC error is reported, a new read at the failing address may not generate an ECC error if the data is still present in the current buffer, even if ECCC and ECCD are cleared.

4.3.3 Read access latency

To correctly read data from Flash memory, the number of wait states (LATENCY) must be correctly programmed in the [Flash access control register \(FLASH_ACR\)](#) according to the frequency of the CPU clock (HCLK) and the internal voltage range of the device V_{CORE} . Refer to [Section 6.1.5: Dynamic voltage scaling management](#). Table 19 shows the correspondence between wait states and CPU clock frequency.

Table 19. Number of wait states according to CPU clock (HCLK) frequency

| Wait states (WS) (LATENCY) | HCLK (MHz) | | |
|-------------------------------|----------------------------------|-----------------------------------|--------------------|
| | V_{CORE} Range 1 boost mode | V_{CORE} Range 1 normal mode | V_{CORE} Range 2 |
| 0 WS (1 CPU cycles) | ≤ 34 | ≤ 30 | ≤ 12 |
| 1 WS (2 CPU cycles) | ≤ 68 | ≤ 60 | ≤ 24 |
| 2 WS (3 CPU cycles) | ≤ 102 | ≤ 90 | ≤ 26 |
| 3 WS (4 CPU cycles) | ≤ 136 | ≤ 120 | - |
| 4 WS (5 CPU cycles) | ≤ 170 | ≤ 150 | - |

After reset, the CPU clock frequency is 16 MHz and 1 wait state (WS) is configured in the FLASH_ACR register.

When changing the CPU frequency, the following software sequences must be applied in order to tune the number of wait states needed to access the Flash memory:

Increasing the CPU frequency:

1. Program the new number of wait states to the LATENCY bits in the *Flash access control register (FLASH_ACR)*.
 2. Check that the new number of wait states is taken into account to access the Flash memory by reading the FLASH_ACR register.
 3. Analyze the change of CPU frequency change caused either by:
 - changing clocksource defined by SW bits in RCC_CFGR register
 - or by CPU clock prescaler defined by HPRE bits in RCC_CFGR
- If some of above two steps decreases the CPU frequency, firstly perform this step and then the rest. Otherwise modify The CPU clock source by writing the SW bits in the RCC_CFGR register and then (if needed) modify the CPU clock prescaler by writing the HPRE bits in RCC_CFGR.
4. Check that the new CPU clock source or/and the new CPU clock prescaler value is/are taken into account by reading the clock source status (SWS bits) or/and the AHB prescaler value (HPRE bits), respectively, in the RCC_CFGR register.

Decreasing the CPU frequency:

1. Modify the CPU clocksource by writing the SW bits in the RCC_CFGR register.
2. If needed, modify the CPU clock prescaler by writing the HPRE bits in RCC_CFGR.
3. Analyze the change of CPU frequency change caused either by:
 - changing clocksource defined by SW bits in RCC_CFGR register
 - or by CPU clock prescaler defined by HPRE bits in RCC_CFGR

If some of above two steps increases the CPU frequency, firstly perform another step and then this step. Otherwise modify The CPU clock source by writing the SW bits in the RCC_CFGR register and then (if needed) modify the CPU clock prescaler by writing the HPRE bits in RCC_CFGR.

4. Check that the new CPU clock source or/and the new CPU clock prescaler value is/are taken into account by reading the clock source status (SWS bits) or/and the AHB prescaler value (HPRE bits), respectively, in the RCC_CFGR register.
5. Program the new number of wait states to the LATENCY bits in *Flash access control register (FLASH_ACR)*.
6. Check that the new number of wait states is used to access the Flash memory by reading the FLASH_ACR register.

4.3.4 Adaptive real-time memory accelerator (ART Accelerator)

The proprietary Adaptive real-time (ART) memory accelerator is optimized for STM32 industry-standard Arm® Cortex®-M4 with FPU processors. It balances the inherent performance advantage of the Arm® Cortex®-M4 with FPU over Flash memory technologies, which normally requires the processor to wait for the Flash memory at higher operating frequencies.

To release the processor full performance, the accelerator implements an instruction prefetch queue and branch cache which increases program execution speed from the 64-bit Flash memory. Based on CoreMark benchmark, the performance achieved thanks to the ART accelerator is equivalent to 0 wait state program execution from Flash memory at a CPU frequency up to 170 MHz.

Instruction prefetch

The Cortex®-M4 fetches the instruction over the ICode bus and the literal pool (constant/data) over the DCode bus. The prefetch block aims at increasing the efficiency of ICode bus accesses.

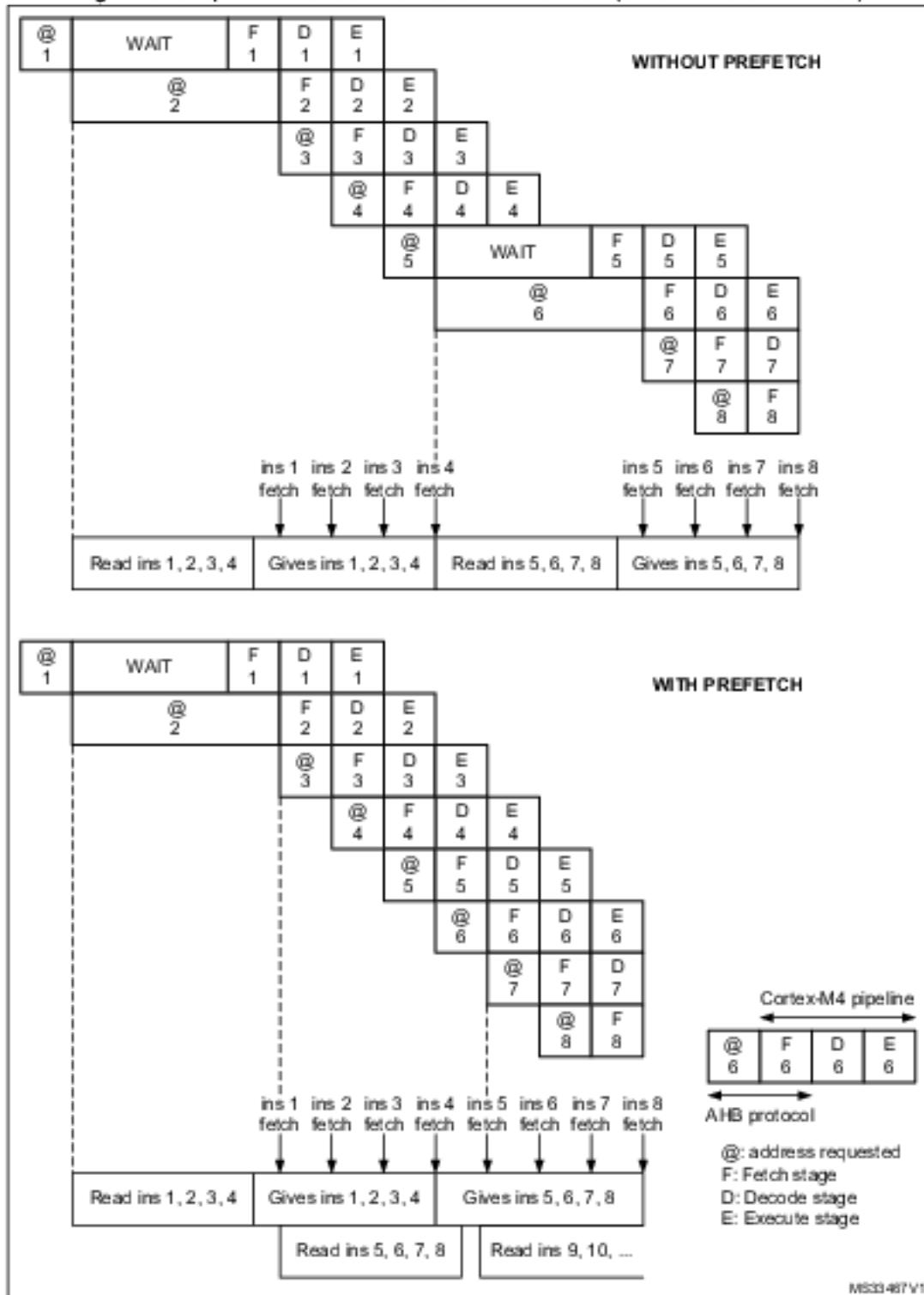
Each Flash memory read operation provides 64 bits from either two instructions of 32 bits or four instructions of 16 bits depending on the launched program. This 64-bits current instruction line is saved in a current buffer, and in case of sequential code, at least two CPU cycles are needed to execute the previous read instruction line.

Prefetch on the ICode bus can be used to read the next sequential instruction line from the Flash memory while the current instruction line is being requested by the CPU.

Prefetch is enabled by setting the PRFTEN bit in the [*Flash access control register \(FLASH_ACR\)*](#). This feature is useful if at least one wait state is needed to access the Flash memory.

[*Figure 9*](#) shows the execution of sequential 16-bit instructions with and without prefetch when 3 WS are needed to access the Flash memory.

Figure 6. Sequential 16-bit instructions execution (64-bit read data width)



When the code is not sequential (branch), the instruction may not be present in the currently used instruction line or in the prefetched instruction line. In this case (miss), the penalty in terms of number of cycles is at least equal to the number of wait states.

If a loop is present in the current buffer, no new flash access is performed.

Instruction cache memory (I-Cache)

To limit the time lost due to jumps, it is possible to retain 32 lines of 4 x 64 bits in an instruction cache memory. This feature can be enabled by setting the instruction cache enable (ICEN) bit in the *Flash access control register (FLASH_ACR)*. Each time a miss occurs (requested data not present in the currently used instruction line, in the prefetched instruction line or in the instruction cache memory), the line read is copied into the instruction cache memory. If some data contained in the instruction cache memory are requested by the CPU, they are provided without inserting any delay. Once all the instruction cache memory lines have been filled, the LRU (least recently used) policy is used to determine the line to replace in the instruction memory cache. This feature is particularly useful in case of code containing loops.

The Instruction cache memory is enable after system reset.

Data cache memory (D-Cache)

Literal pools are fetched from Flash memory through the DCode bus during the execution stage of the CPU pipeline. Each DCode bus read access fetches 64 bits which are saved in a current buffer. The CPU pipeline is consequently stalled until the requested literal pool is provided. To limit the time lost due to literal pools, accesses through the AHB databus DCode have priority over accesses through the AHB instruction bus ICode.

If some literal pools are frequently used, the data cache memory can be enabled by setting the data cache enable (DC EN) bit in the *Flash access control register (FLASH_ACR)*. This feature works like the instruction cache memory, but the retained data size is limited to 8 rows of 4*64 bits.

The Data cache memory is enable after system reset.

Note: The D-Cache is active only when data is requested by the CPU (not by DMA1 and DMA2).
Data in option bytes block are not cacheable.

4.3.5 Flash program and erase operations

The STM32G4 Series embedded Flash memory can be programmed using in-circuit programming or in-application programming.

The **in-circuit programming (ICP)** method is used to update the entire contents of the Flash memory, using the JTAG, SWD protocol or the boot loader to load the user application into the microcontroller. ICP offers quick and efficient design iterations and eliminates unnecessary package handling or socketing of devices.

In contrast to the ICP method, **in-application programming (IAP)** can use any communication interface supported by the microcontroller (I/Os, USB, CAN, UART, I²C, SPI, etc.) to download programming data into memory. IAP allows the user to re-program the Flash memory while the application is running. Nevertheless, part of the application has to have been previously programmed in the Flash memory using ICP.

The contents of the Flash memory are not guaranteed if a device reset occurs during a Flash memory operation.

The Flash erase and programming is only possible in the voltage scaling range 1. The VOS[1:0] bits in the PWR_CR1 must be programmed to 01b.

During a program/erase operation to the Flash memory, any attempt to read the Flash memory stalls the bus. The read operation proceeds correctly once the program/erase operation has completed.

Unlocking the Flash memory

After reset, write is not allowed in the *Flash control register (FLASH_CR)* to protect the Flash memory against possible unwanted operations due, for example, to electric disturbances. The following sequence is used to unlock this register:

1. Write KEY1 = 0x45670123 in the *Flash key register (FLASH_KEYR)*
2. Write KEY2 = 0xCDEF89AB in the *FLASH_KEYR* register.

Any wrong sequence locks up the *FLASH_CR* register until the next system reset. In the case of a wrong key sequence, a bus error is detected and a Hard Fault interrupt is generated.

The *FLASH_CR* register can be locked again by software by setting the LOCK bit in the *FLASH_CR* register.

Note: *The FLASH_CR register cannot be written when the BSY bit in the Flash status register (FLASH_SR) is set. Any attempt to write to it with the BSY bit set causes the AHB bus to stall until the BSY bit is cleared.*

4.3.6 Flash main memory erase sequences

The Flash memory erase operation can be performed at page level or on the whole Flash memory (Mass Erase). Mass Erase does not affect the Information block (system flash, OTP and option bytes).

Page erase

To erase a page, follow the procedure below:

1. Check that no Flash memory operation is ongoing by checking the BSY bit in the *Flash status register (FLASH_SR)*.
2. Check and clear all error programming flags due to a previous programming. If not, PGERR is set.
3. Set the PER bit and select the page to erase (PNB).
4. Set the STRT bit in the *FLASH_CR* register.
5. Wait for the BSY bit to be cleared in the *FLASH_SR* register.

Note: *The internal oscillator HSI16 (16 MHz) is enabled automatically when STRT bit is set, and disabled automatically when STRT bit is cleared, except if the HSI16 is previously enabled with HSION in RCC_CR register.*

If the page erase is part of write-protected area (by WRP or PCR OP), WRPPER is set and the page erase request is aborted.

Mass erase

To perform a Mass erase, follow the procedure below:

1. Check that no Flash memory operation is ongoing by checking the BSY bit in the FLASH_SR register.
2. Check and clear all error programming flags due to a previous programming. If not, PGERR is set.
3. Set the MER1 bit in the Flash control register (FLASH_CR).
4. Set the STRT bit in the FLASH_CR register.
5. Wait for the BSY bit to be cleared in the Flash status register (FLASH_SR).

Note: *The internal oscillator HSI16 (16 MHz) is enabled automatically when STRT bit is set, and disabled automatically when STRT bit is cleared, except if the HSI16 is previously enabled with HSION in RCC_CR register.*

If the Flash memory contains a write-protected area (by WRP or PCROP), WRPERR is set and the mass erase request is aborted.

4.3.7 Flash main memory programming sequences

The Flash memory is programmed 72 bits at a time (64 bits + 8 bits ECC).

Programming in a previously programmed address is not allowed except if the data to write is full zero, and any attempt sets PROGERR flag in the *Flash status register (FLASH_SR)*.

It is only possible to program double word (2 x 32-bit data).

- Any attempt to write byte or half-word sets SIZERR flag in the FLASH_SR register.
- Any attempt to write a double word which is not aligned with a double word address sets PGAERR flag in the FLASH_SR register.

Standard programming

The Flash memory programming sequence in standard mode is as follows:

1. Check that no Flash main memory operation is ongoing by checking the BSY bit in the *Flash status register (FLASH_SR)*.
2. Check and clear all error programming flags due to a previous programming. If not, PGERR is set.
3. Set the PG bit in the *Flash control register (FLASH_CR)*.
4. Perform the data write operation at the desired memory address, inside main memory block or OTP area. Only double word can be programmed.
 - Write a first word in an address aligned with double word
 - Write the second word
5. Wait until the BSY bit is cleared in the FLASH_SR register.
6. Check that EOP flag is set in the FLASH_SR register (meaning that the programming operation has succeed), and clear it by software.
7. Clear the PG bit in the FLASH_SR register if there no more programming request anymore.

Note: *When the flash interface has received a good sequence (a double word), programming is automatically launched and BSY bit is set. The internal oscillator HSI16 (16 MHz) is enabled*

automatically when PG bit is set, and disabled automatically when PG bit is cleared, except if the HSI16 is previously enabled with HSION in RCC_CR register.

If the user needs to program only one word, double word must be completed with the erase value 0xFFFF FFFF to launch automatically the programming.

ECC is calculated from the double word to program.

Fast programming

This mode allows to program a row (32 double words), and to reduce the page programming time by eliminating the need for verifying the flash locations before they are programmed and to avoid rising and falling time of high voltage for each double word. During fast programming, the CPU clock frequency (HCLK) must be at least 8 MHz.

Only the main memory can be programmed in Fast programming mode.

The Flash main memory programming sequence in standard mode is as follows:

1. Perform a mass erase. If not, PGSERR is set.
2. Check that no Flash main memory operation is ongoing by checking the BSY bit in the *Flash status register (FLASH_SR)*.
3. Check and clear all error programming flag due to a previous programming.
4. Set the FSTPG bit in *Flash control register (FLASH_CR)*.
5. Write the 32 double words to program a row. Only double words can be programmed:
 - Write a first word in an address aligned with double word
 - Write the second word.
6. Wait until the BSY bit is cleared in the FLASH_SR register.
7. Check that EOP flag is set in the FLASH_SR register (meaning that the programming operation has succeed), and clear it by software.
8. Clear the FSTPG bit in the FLASH_SR register if there no more programming request anymore.

Note:

If the flash is attempted to be written in Fast programming mode while a read operation is on going, the programming is aborted without any system notification (no error flag is set).

When the Flash interface has received the first double word, programming is automatically launched. The BSY bit is set when the high voltage is applied for the first double word, and it is cleared when the last double word has been programmed or in case of error. The internal oscillator HSI16 (16 MHz) is enabled automatically when FSTPG bit is set, and disabled automatically when FSTPG bit is cleared, except if the HSI16 is previously enabled with HSION in RCC_CR register.

The 32 double word must be written successively. The high voltage is kept on the flash for all the programming. Maximum time between two double words write requests is the time programming (around 2 x 25us). If a second double word arrives after this time programming, fast programming is interrupted and MISSERR is set.

High voltage mustn't exceed 8 ms for a full row between 2 erases. This is guaranteed by the sequence of 32 double words successively written with a clock system greater or equal to 8MHz. An internal time-out counter counts 7ms when Fast programming is set and stops the programming when time-out is over. In this case the FASTERR bit is set.

If an error occurs, high voltage is stopped and next double word to programmed is not programmed. Anyway, all previous double words have been properly programmed.

Programming errors

Several kind of errors can be detected. In case of error, the Flash operation (programming or erasing) is aborted.

- **PROGERR:** Programming Error

In standard programming: PROGERR is set if the word to write is not previously erased (except if the value to program is full zero).

- **SIZERR:** Size Programming Error

In standard programming or in fast programming: only double word can be programmed and only 32-bit data can be written. SIZERR is set if a byte or an half-word is written.

- **PGAERR:** Alignment Programming error

PGAERR is set if one of the following conditions occurs:

- In standard programming: the first word to be programmed is not aligned with a double word address, or the second word doesn't belong to the same double word address.
- In fast programming: the data to program doesn't belong to the same row than the previous programmed double words, or the address to program is not greater than the previous one.

- **PGSERR:** Programming Sequence Error

PGSERR is set if one of the following conditions occurs:

- In the standard programming sequence or the fast programming sequence: a data is written when PG and FSTPG are cleared.
- In the standard programming sequence or the fast programming sequence: MER1, and PER are not cleared when PG or FSTPG is set.
- In the fast programming sequence: the Mass erase is not performed before setting FSTPG bit.
- In the mass erase sequence: PG, FSTPG, and PER are not cleared when MER1 is set.
- In the page erase sequence: PG, FSTPG, MER1 are not cleared when PER is set.
- PGSERR is set also if PROGERR, SIZERR, PGAERR, WRPERR, MISSERR, FASTERR or PGSERR is set due to a previous programming error.

- **WRPERR:** Write Protection Error

WRPERR is set if one of the following conditions occurs:

- Attempt to program or erase in a write protected area (WRP) or in a PCROP area or in a Securable memory area.
- Attempt to perform an erase when one page or more is protected by WRP or PCROP.
- The debug features are connected or the boot is executed from SRAM or from System flash when the read protection (RDP) is set to Level 1.
- Attempt to modify the option bytes when the read protection (RDP) is set to Level 2.

- **MISSERR:** Fast Programming Data Miss Error

In fast programming: all the data must be written successively. MISSERR is set if the previous data programmation is finished and the next data to program is not written yet.

- **FASTERR:** Fast Programming Error

In fast programming: FASTERR is set if one of the following conditions occurs:

- When FSTPG bit is set for more than 7 ms which generates a time-out detection.
- When the fast programming has been interrupted by a MISSERR, PGAERR, WRPERR or SIZERR.

If an error occurs during a program or erase operation, one of the following error flags is set in the FLASH_SR register:

PROGERR, SIZERR, PGAERR, PGSER, MISSERR (Program error flags),
WRPERR (Protection error flag)

In this case, if the error interrupt enable bit ERRIE is set in the *Flash status register (FLASH_SR)*, an interrupt is generated and the operation error flag OPERR is set in the FLASH_SR register.

Note: *If several successive errors are detected (for example, in case of DMA transfer to the Flash memory), the error flags cannot be cleared until the end of the successive write requests.*

Programming and caches

If a Flash memory write access concerns some data in the data cache, the Flash write access modifies the data in the Flash memory and the data in the cache.

If an erase operation in Flash memory also concerns data in the data or instruction cache, you have to make sure that these data are rewritten before they are accessed during code execution. If this cannot be done safely, it is recommended to flush the caches by setting the DCRST and ICRST bits in the *Flash access control register (FLASH_ACR)*.

Note: *The I/D cache should be flushed only when it is disabled (IDCEN = 0).*

4.4 FLASH option bytes

4.4.1 Option bytes description

The option bytes are configured by the end user depending on the application requirements. As a configuration example, the watchdog may be selected in hardware or software mode (refer to [Section 5.4.2: Option bytes programming](#)).

A double word is split up as follows in the option bytes:

Table 20. Option byte format

| 63-24 | 23-16 | 15-8 | 7-0 | 31-24 | 23-16 | 15-8 | 7-0 |
|----------------------------|----------------------------|----------------------------|----------------------------|---------------|---------------|---------------|---------------|
| Complemented option byte 3 | Complemented option byte 2 | Complemented option byte 1 | Complemented option byte 0 | Option byte 3 | Option byte 2 | Option byte 1 | Option byte 0 |

The organization of these bytes inside the information block is as shown in [Table 31: Option byte organization](#).

The option bytes can be read from the memory locations listed in [Table 31: Option byte organization](#) or from the Option byte registers:

- [Flash option register \(FLASH_OPTR\)](#)
- [Flash PCROP1 Start address register \(FLASH_PCROP1SR\)](#)
- [Flash PCROP1 End address register \(FLASH_PCROP1ER\)](#)
- [Flash WRP area A address register \(FLASH_WRP1AR\)](#)
- [Flash WRP area B address register \(FLASH_WRP1BR\)](#)

Table 21. Option byte organization

| Address | [63:56] | [55:48] | [47:40] | [39:32] | [31:24] | [23:16] | [15:8] | [7:0] |
|----------|----------------------|----------------|------------------------------|-----------------|---------|----------------------|----------|------------------------------|
| 1FFF7800 | USER OPT | | | RDP | | | USER OPT | |
| 1FFF7808 | Unused | | Unused and PCROP1_STRT[15:0] | | | Unused | | Unused and PCROP1_STRT[15:0] |
| 1FFF7810 | PCROP_RDP and Unused | | Unused and PCROP1_END[15:0] | | | PCROP_RDP and Unused | | Unused and PCROP1_END[15:0] |
| 1FFF7818 | Unused | WRP1A_END[7:0] | Unused | WRP1A_STRT[7:0] | Unused | WRP1A_END[7:0] | Unused | WRP1A_STRT[7:0] |
| 1FFF7820 | Unused | WRP2A_END[7:0] | Unused | WRP2A_STRT[7:0] | Unused | WRP2A_END[7:0] | Unused | WRP2A_STRT[7:0] |
| 1FFF7828 | Unused | BOOT_LOCK | Unused | SEC_SIZE1 | Unused | BOOT_LOCK | Unused | SEC_SIZE1 |

User and read protection option bytes

Flash memory address: 0x1FFF 7800

ST production value: 0xFFEF F8AA

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|-----------|----------------|-----------|--------|-------------|-------------|---------|----------|-----------|------|------|---------|------------|-----------|---------|----|
| Res. | IRH_EN | NRST_MODE[1:0] | | nBOOT0 | nSWBOOT0 | CCMSRAM_RST | SRAM_PE | nBOOT1 | PB4_PUPEN | Res. | Res. | WWDG_SW | IWDG_STDBY | IWDG_STOP | IWDG_SW | |
| | r | r | r | r | r | r | r | r | r | | | r | r | r | r | |
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | nRST_SHDW | nRST_STDBY | nRST_STOP | Res. | BDRLEV[2:0] | | | RDP[7:0] | | | | | | | | |
| | r | r | r | | r | r | r | r | r | r | r | r | r | r | r | |

Bit 31 Reserved, must be kept at reset value.

Bit 30 **IRH_IN**: Internal reset holder for PG10

- 0: IRH disabled
- 1: IRH enabled

Bits 29:28 **NRST_MODE[1:0]**: PG10 pad mode

- 00: Reset Input/Output
- 01: Reset Input only
- 10: GPIO
- 11: Reset Input/Output

Bit 27 **nBOOT0**: nBOOT0 option bit

- 0: nBOOT0 = 0
- 1: nBOOT0 = 1

Bit 26 **nSWBOOT0**: Software BOOT0

- 0: BOOT0 taken from the option bit nBOOT0
- 1: BOOT0 taken from PB8/BOOT0 pin

Bit 25 **CCMSRAM_RST**: CCM SRAM erase when system reset

- 0: CCM SRAM erased when a system reset occurs
- 1: CCM SRAM is not erased when a system reset occurs

Bit 24 **SRAM_PE**: SRAM1 and CCM SRAM parity check enable

- 0: SRAM1 and CCM SRAM parity check enable
- 1: SRAM1 and CCM SRAM parity check disable

Bit 23 **nBOOT1**: Boot configuration

Together with the BOOT0 pin, this bit selects boot mode from the Flash main memory, SRAM1 or the System memory. Refer to [Section 2.6: Boot configuration](#).

Bit 22 **PB4_PUPEN**: PB4 pull-up enable

- 0: USB power delivery dead-battery enabled/NJTRST pull-up deactivated
 - 1: USB power delivery dead-battery disabled/NJTRST pull-up activated
- Note:* Only for Category 4 devices (otherwise Reserved)

Bits 21:20 Reserved, must be kept at reset value.

Bit 19 **WWDG_SW**: Window watchdog selection

- 0: Hardware window watchdog
- 1: Software window watchdog

Bit 18 **IWDG_STDBY**: Independent watchdog counter freeze in Standby mode

- 0: Independent watchdog counter is frozen in Standby mode
- 1: Independent watchdog counter is running in Standby mode

Bit 17 **IWDG_STOP**: Independent watchdog counter freeze in Stop mode

- 0: Independent watchdog counter is frozen in Stop mode
- 1: Independent watchdog counter is running in Stop mode

Bit 16 **IWDG_SW**: Independent watchdog selection

- 0: Hardware Independent watchdog
- 1: Software Independent watchdog

Bit 15 Reserved, must be kept at reset value.

Bit 14 **nRST_SHDW**

- 0: Reset generated when entering the Shutdown mode
- 1: No reset generated when entering the Shutdown mode

Bit 13 **nRST_STDBY**

- 0: Reset generated when entering the Standby mode
- 1: No reset generated when entering the Standby mode

Bit 12 **nRST_STOP**

- 0: Reset generated when entering the Stop mode
- 1: No reset generated when entering the Stop mode

Bit 11 Reserved, must be kept at reset value.

Bits 10:8 **BORLEV**: BOR reset Level

These bits contain the VDD supply level threshold that activates/releases the reset.

- 000: BOR Level 0. Reset level threshold is around 1.7 V
- 001: BOR Level 1. Reset level threshold is around 2.0 V
- 010: BOR Level 2. Reset level threshold is around 2.2 V
- 011: BOR Level 3. Reset level threshold is around 2.5 V
- 100: BOR Level 4. Reset level threshold is around 2.8 V

Bits 7:0 **RDP**: Read protection level

- 0xAA: Level 0, read protection not active
- 0xCC: Level 2, chip read protection active
- Others: Level 1, memories read protection active

PCROP1 Start address option bytes

Flash memory address: 0x1FFF 7808

Reset value: 0xFFFF FFFF (ST production value)

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PCROP1_STRT[15:0] | | | | | | | | | | | | | | | |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **PCROP1_STRT[15:0]**: PCROP area start offset

PCROP1_STRT contains the first double-word of the PCROP area for bank1.

PCROP1 End address option bytes

Flash memory address: 0x1FFF 7810

Reset value: 0x00FF 0000 (ST production value)

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| PCROP _RDP | Res. |
| r | | | | | | | | | | | | | | | |
| PCROP1-END[15:0] | | | | | | | | | | | | | | | |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

Bit 31 **PCROP_RDP**: PCROP area preserved when RDP level decreased

This bit is set only. It is reset after a full mass erase due to a change of RDP from Level 1 to Level 0.

0: PCROP area is not erased when the RDP level is decreased from Level 1 to Level 0.

1: PCROP area is erased when the RDP level is decreased from Level 1 to Level 0 (full mass erase).

Bits 30:16 Reserved, must be kept at reset value.

Bits 15:0 **PCROP1-END[15:0]**: Bank 1 PCROP area end offset

PCROP1-END contains the last double-word of the PCROP area.

WRP1 Area A address option bytes

Flash memory address: 0x1FFF 7818

Reset value: 0xFF00 FFFF (ST production value)

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-----------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| | | | | | | | | | | r | r | r | r | r | r |
| WRP1A-END[7:0] | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | r | r | r | r | r | r |
| WRP1A-STRT[7:0] | | | | | | | | | | | | | | | |
| | | | | | | | | | | r | r | r | r | r | r |

Bits 31:24 Reserved, must be kept at reset value.

Bits 23:16 **WRP1A-END[7:0]**: WRP first area "A" end offset

WRP1A-END contains the last page of WRP first area.

Bits 15:8 Reserved, must be kept at reset value.

Bits 7:0 **WRP1A-STRT[7:0]**: WRP first area "A" start offset

WRP1A-STRT contains the first page of WRP first area.

WRP2 Area A address option bytes

Flash memory address: 0x1FFF7820

Reset value: 0xFF00 FFFF (ST production value)

| | | | | | | | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|-----------------|----|----|----|----|----|----|----|--|--|--|--|--|--|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | | | | |
| Res. | WRP2A_END[7:0] | | | | | | | | | | | | | |
| | | | | | | | | r | r | r | r | r | r | r | r | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | |
| Res. | WRP2A_STRT[7:0] | | | | | | | | | | | | | |
| | | | | | | | | r | r | r | r | r | r | r | r | | | | | | |

Bits 31:24 Reserved, must be kept at reset value.

Bits 23:16 **WRP2A_END[7:0]**: WRP second area "B" end offset

WRP2A_END contains the last page of the WRP second area.

Bits 15:8 Reserved, must be kept at reset value.

Bits 7:0 **WRP2A_STRT[7:0]**: WRP second area start offset

WRP2A_STRT contains the last page of the WRP second area.

Securable memory area option bytes

Flash memory address: 0x1FFF7828

Reset value: 0xFF00FE00 (ST production value)

| | | | | | | | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|----------------|------|------|------|------|------|------|------|-----------|--|--|--|--|--|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | | | | |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | BOOT_LOCK | | | | | |
| | | | | | | | | | | | | | | | | r | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | |
| Res. | SEC_SIZE1[8:0] | | | | | | | | | | | | | |
| | | | | | | | | r | r | r | r | r | r | r | r | | | | | | |

Bits 31:17 Reserved, must be kept at reset value.

Bit 16 **BOOT_LOCK**: used to force boot from user area

0: Boot based on the pad/option bit configuration

1: Boot forced from Main Flash memory

Bits 15:9 Reserved, must be kept at reset value.

Bits 8:0 **SEC_SIZE1[8:0]**: Securable memory area size

Contains the number of Securable Flash memory pages

4.4.2 Option bytes programming

After reset, the options related bits in the [Flash control register \(FLASH_CR\)](#) are write-protected. To run any operation on the option bytes page, the option lock bit OPTLOCK in the [Flash control register \(FLASH_CR\)](#) must be cleared. The following sequence is used to unlock this register:

1. Unlock the FLASH_CR with the LOCK clearing sequence (refer to [Unlocking the Flash memory](#)).
2. Write OPTKEY1 = 0x08192A3B in the [Flash option key register \(FLASH_OPTKEYR\)](#).
3. Write OPTKEY2 = 0x4C5D6E7F in the FLASH_OPTKEYR register.

The user options can be protected against unwanted erase/program operations by setting the OPTLOCK bit by software.

Note: If LOCK is set by software, OPTLOCK is automatically set too.

Modifying user options

The option bytes are programmed differently from a main memory user address. To modify the user options value, follow the procedure below:

1. Check that no Flash memory operation is on going by checking the BSY bit in the [Flash status register \(FLASH_SR\)](#).
2. Clear OPTLOCK option lock bit with the clearing sequence described above.
3. Write the desired options value in the options registers: [Flash option register \(FLASH_OPTR\)](#), [Flash PCROP1 Start address register \(FLASH_PCR0P1SR\)](#), [Flash PCROP1 End address register \(FLASH_PCR0P1ER\)](#), [Flash WRP area A address register \(FLASH_WRP1AR\)](#), [Flash WRP area B address register \(FLASH_WRP1BR\)](#).
4. Set the Options Start bit OPTSTRT in the [Flash control register \(FLASH_CR\)](#).
5. Wait for the BSY bit to be cleared.

Note: Any modification of the value of one option is automatically performed by erasing both user option bytes pages first and then programming all the option bytes with the values contained in the flash option registers.

Option byte loading

After the BSY bit is cleared, all new options are updated into the flash but they are not applied to the system. They have effect on the system when they are loaded. Option bytes loading (OBL) is performed in two cases:

- when OBL_LAUNCH bit is set in the [Flash control register \(FLASH_CR\)](#).
- after a power reset (BOR reset or exit from Standby/Shutdown modes).

Option byte loader performs a read of the options block and stores the data into internal option registers. These internal registers configure the system and cannot be read with software. Setting OBL_LAUNCH generates a reset so the option byte loading is performed under system reset.

Each option bit has also its complement in the same double word. During option loading, a verification of the option bit and its complement allows to check the loading has correctly taken place.

During option byte loading, the options are read by double word with ECC. If the word and its complement are matching, the option word/byte is copied into the option register.

If the comparison between the word and its complement fails, a status bit OPTVERR is set. Mismatch values are forced into the option registers:

- For USR OPT option, the value of mismatch is all options at ‘1’, except for BOR_lev which is “000” (lowest threshold)
- For WRP option, the value of mismatch is the default value “No protection”
- For RDP option, the value of mismatch is the default value “Level 1”
- For PCROP, the value of mismatch is “all memory protected”

On system reset rising, internal option registers are copied into option registers which can be read and written by software (FLASH_OPTR, FLASH_PCROP1SR, FLASH_PCROP1ER, FLASH_WRP1AR, FLASH_WRP1BR). These registers are also used to modify options. If these registers are not modified by user, they reflects the options states of the system. See [Section : Modifying user options](#) for more details.

4.5 FLASH memory protection

The Flash main memory can be protected against external accesses with the Read protection (RDP). The pages of the Flash memory can also be protected against unwanted write due to loss of program counter contexts. The write-protection (WRP) granularity is one page (2 Kbyte). A part of the flash memory can also be protected against read and write from third parties (PCROP). The PCROP granularity is double word (64-bit).

4.5.1 Read protection (RDP)

The read protection is activated by setting the RDP option byte and then, by applying a system reset to reload the new RDP option byte. The read protection protects the Flash main memory, the option bytes, the backup registers (RTC_BKPxR in the RTC) and the CCM SRAM.

Note: If the read protection is set while the debugger is still connected through JTAG/SWD, apply a POR (power-on reset) instead of a system reset.

There are three levels of read protection from no protection (level 0) to maximum protection or no debug (level 2).

The Flash memory is protected when the RDP option byte and its complement contain the pair of values shown in [Table 32](#).

Table 22. Flash memory read protection status

| RDP byte value | RDP complement value | Read protection level |
|-------------------------------|--|----------------------------|
| 0xAA | 0x55 | Level 0 (production value) |
| Any value except 0xAA or 0xCC | Any value (not necessarily complementary) except 0x55 and 0x33 | Level 1 |
| 0xCC | 0x33 | Level 2 |

The System memory area is read accessible whatever the protection level. It is never accessible for program/erase operation.

Level 0: no protection

Read, program and erase operations into the Flash main memory area are possible. The option bytes, the CCM SRAM and the backup registers are also accessible by all operations.

Level 1: Read protection

This is the default protection level when RDP option byte is erased. It is defined as well when RDP value is at any value different from 0xAA and 0xCC, or even if the complement is not correct.

- **User mode:** Code executing in user mode (**Boot Flash**) can access Flash main memory, option bytes, CCM SRAM and backup registers with all operations.
- **Debug, boot RAM and boot loader modes:** In debug mode or when code is running from boot RAM or boot loader, the Flash main memory, the backup registers (RTC_BKPxR in the RTC) and the CCM SRAM are totally inaccessible. In these modes, a read or write access to the Flash generates a bus error and a Hard Fault interrupt.

Caution: In case the Level 1 is configured and no PCROP area is defined, it is mandatory to set PCROP_RDP bit to 1 (full mass erase when the RDP level is decreased from Level 1 to Level 0). In case the Level 1 is configured and a PCROP area is defined, if user code needs to be protected by RDP but not by PCROP, it must not be placed in a page containing a PCROP area.

Level 2: No debug

In this level, the protection level 1 is guaranteed. In addition, the Cortex®-M4 debug port, the boot from RAM (boot RAM mode) and the boot from System memory (boot loader mode) are no more available. In user execution mode (boot FLASH mode), all operations are allowed on the Flash Main memory. On the contrary, only read operations can be performed on the option bytes.

Option bytes cannot be programmed nor erased. Thus, the level 2 cannot be removed at all: it is an irreversible operation. When attempting to modify the options bytes, the protection error flag WRPERR is set in the Flash_SR register and an interrupt can be generated.

Note: *The debug feature is also disabled under reset.*

STMicroelectronics is not able to perform analysis on defective parts on which the level 2 protection has been set.

Changing the Read protection level

It is easy to move from level 0 to level 1 by changing the value of the RDP byte to any value (except 0xCC). By programming the 0xCC value in the RDP byte, it is possible to go to level 2 either directly from level 0 or from level 1. Once in level 2, it is no more possible to modify the Read protection level.

When the RDP is reprogrammed to the value 0xAA to move from Level 1 to Level 0, a mass erase of the Flash main memory is performed if PCROP_RDP is set in the [Flash PCR OP1 End address register \(FLASH_PCROP1ER\)](#). The backup registers (RTC_BKPxR in the RTC) and the CCM SRAM are also erased. The user options except PCROP protection are set to their previous values copied from FLASH_OPTR, FLASH_WRPxyR (x=1 and y=A or B). PCROP is disable. The OTP area is not affected by mass erase and remains unchanged.

If the bit PCROP_RDP is cleared in the FLASH_PCROP1ER, the full mass erase is replaced by a partial mass erase that is successive page erases, except for the pages protected by PCROP. This is done in order to keep the PCROP code. Only when the Flash memory is erased, options are re-programmed with their previous values. This is also true for FLASH_PCROPxSR and FLASH_PCROPxER registers ($x=1$).

Note: *Full Mass Erase or Partial Mass Erase is performed only when Level 1 is active and Level 0 requested. When the protection level is increased (0->1, 1->2, 0->2) there is no mass erase. To validate the protection level change, the option bytes must be reloaded through the OBL_LAUNCH bit in Flash control register.*

Figure 7. Changing the read protection (RDP) level

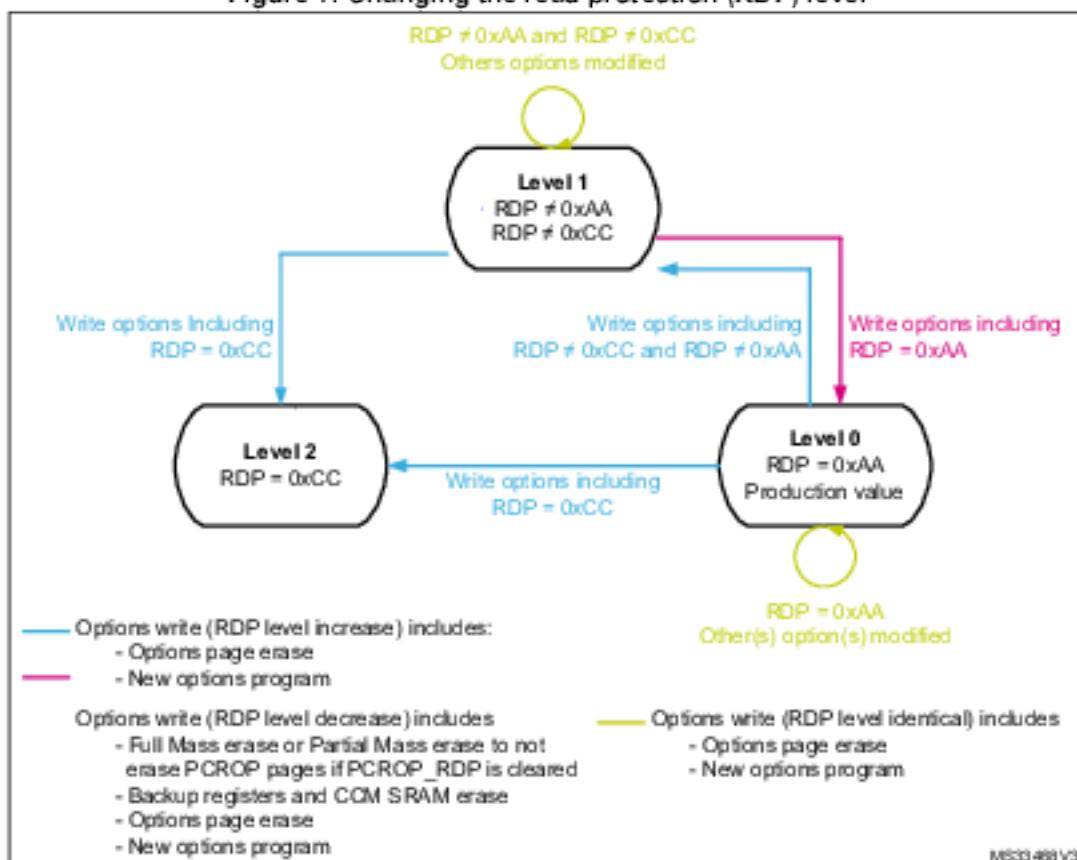


Table 23. Access status versus protection level and execution modes

| Area | Protection level | User execution (BootFromFlash) | | | Debug / BootFromRam / BootFromLoader ⁽¹⁾ | | |
|------------------------------|------------------|--------------------------------|-------|-------|---|-------|-------------------|
| | | Read | Write | Erase | Read | Write | Erase |
| Flash main memory | 1 | Yes | Yes | Yes | No | No | No ⁽³⁾ |
| | 2 | Yes | Yes | Yes | N/A | N/A | N/A |
| System memory ⁽²⁾ | 1 | Yes | No | No | Yes | No | No |
| | 2 | Yes | No | No | N/A | N/A | N/A |

Table 23. Access status versus protection level and execution modes (continued)

| Area | Protection level | User execution (BootFromFlash) | | | Debug / BootFromRam/ BootFromLoader ⁽¹⁾ | | |
|------------------|------------------|--------------------------------|--------------------|-------|---|--------------------|-------------------|
| | | Read | Write | Erase | Read | Write | Erase |
| Option bytes | 1 | Yes | Yes ⁽³⁾ | Yes | Yes | Yes ⁽³⁾ | Yes |
| | 2 | Yes | No | No | N/A | N/A | N/A |
| OTP | 1 | Yes | Yes ⁽⁴⁾ | N/A | No | No | N/A |
| | 2 | Yes | Yes ⁽⁴⁾ | N/A | N/A | N/A | N/A |
| Backup registers | 1 | Yes | Yes | N/A | No | No | No ⁽⁵⁾ |
| | 2 | Yes | Yes | N/A | N/A | N/A | N/A |
| CCM SRAM | 1 | Yes | Yes | N/A | No | No | No ⁽⁶⁾ |
| | 2 | Yes | Yes | N/A | N/A | N/A | N/A |

1. When the protection level 2 is active, the Debug port, the boot from RAM and the boot from system memory are disabled.
2. The system memory is only read-accessible, whatever the protection level (0, 1 or 2) and execution mode.
3. The Flash main memory is erased when the RDP option byte is programmed with all level protections disabled (0xAA).
4. OTP can only be written once.
5. The backup registers are erased when RDP changes from level 1 to level 0.
6. The CCM SRAM is erased when RDP changes from level 1 to level 0.

4.5.2 Proprietary code readout protection (PCROP)

Apart of the flash memory can be protected against read and write from third parties. The protected area is execute-only: it can only be reached by the STM32 CPU, as an instruction code, while all other accesses (DMA, debug and CPU data read, write and erase) are strictly prohibited. The PCROP area has a double word (64-bit) granularity. An additional option bit (PCROP_RDP) allows to select if the PCROP area is erased or not when the RDP protection is changed from Level 1 to Level 0 (refer to [Changing the Read protection level](#)).

Each PCROP area is defined by a start page offset and an end page offset related to the physical Flash address. These offsets are defined in the PCROP address registers [Flash PCROP1 Start address register \(FLASH_PCROP1SR\)](#), [Flash PCROP1 End address register \(FLASH_PCROP1ER\)](#).

- The PCROP_x ($x = 1$) area is defined from the address: Flash memory base address + [PCROP_x_STRT x 0x8] (included) to the address: Flash memory base address + [(PCROP_x_END+1) x 0x8] (excluded). The minimum PCROP area size is two double-words (128 bits).

For example, to protect by PCROP from the address 0x0806 2F80 (included) to the address 0x0807 0004 (included):

- If boot in flash is selected, FLASH_PCROP1SR and FLASH_PCROP1ER registers must be programmed with:
 - PCROP1_STRT = 0xC5F0.
 - PCROP1_END = 0xE000.

Any read access performed through the D-bus to a PCROP protected area triggers RDERR flag error.

Any PCROP protected address is also write protected and any write access to one of these addresses triggers WRPERR.

Any PCROP area is also erase protected. Consequently, any erase to a page in this zone is impossible (including the page containing the start address and the end address of this zone). Moreover, a software mass erase cannot be performed if one zone is PCROP protected.

For previous example, due to erase by page, all pages from page 0xC5 to 0xE0 are protected in case of page erase. (All addresses from 0x0806 2800 to 0x0807 07FF can't be erased).

Deactivation of PCROP can only occurs when the RDP is changing from level 1 to level 0. If the user options modification tries to clear PCROP or to decrease the PCROP area, the options programming is launched but PCROP area stays unchanged. On the contrary, it is possible to increase the PCROP area.

When option bit PCROP_RDP is cleared, when the RDP is changing from level 1 to level 0, Full Mass Erase is replaced by Partial Mass Erase in order to keep the PCROP area (refer to [Changing the Read protection level](#)). In this case, PCROP1_STRT and PCROP1_END are also not erased.

Note: *It is recommended to align PCROP area with page granularity when using PCROP_RDP, or to leave free the rest of the page where PCROP zone starts or ends.*

Table 24. PCROP protection⁽¹⁾

| PCROPx registers values (x = 1) | PCROP protection area |
|---|--|
| PCROPx_offset strt > PCROPx_offset_end | No PCROP area. |
| PCROPx_offset strt < PCROPx_offset_end | The area between PCROPx_offset strt and PCROPx_offset_end is protected. It is possible to write: – PCROPx_offset strt with a lower value – PCROPx_offset_end with a higher value. |

1. The minimum PCROP area size is 2xdouble words: PCROPx_offset strt and PCROPx_offset_end.

4.5.3 Write protection (WRP)

The user area in Flash memory can be protected against unwanted write operations. It allows either to specify:

- Two write-protected (WRP) areas can be defined, with page (2 KByte) granularity.

Each area is defined by a start page offset and an end page offset related to the physical Flash base address. These offsets are defined in the WRP address registers: [Flash WRP area A address register \(FLASH_WRP1AR\)](#), [Flash WRP area B address register \(FLASH_WRP1BR\)](#).

The WRP "y" area (x=1 and y=A,B) is defined from the address: *Flash memory Base address + [WRPx_y_STRT x 0x800] (included)* to the address: *Flash memory Base address + [(WRPx_y_END +1) x 0x800] (excluded)*.

For example, to protect by WRP from the address 0x0801 2800 (included) to the address 0x0801 87FF (included):

- if boot in flash is selected, FLASH_WRP1AR register must be programmed with:
 - WRP1A_STRT = 0x25.
 - WRP1A_END = 0x30.

WRP1B_STRT and WRP1B_END in FLASH_WRP1BR can be used instead (area "B" in Flash memory).

When WRP is active, it cannot be erased or programmed. Consequently, a software mass erase cannot be performed if one area is write-protected.

If an erase/program operation to a write-protected part of the Flash memory is attempted, the write protection error flag (WRPERR) is set in the FLASH_SR register. This flag is also set for any write access to:

- OTP area
- part of the Flash memory that can never be written like the ICP
- PCROP area.

Note: When the memory read protection level is selected (RDP level = 1), it is not possible to program or erase Flash memory if the CPU debug features are connected (JTAG or single wire) or boot code is being executed from RAM or System flash, even if WRP is not activated.

Note: To validate the WRP options, the option bytes must be reloaded through the OBL_LAUNCH bit in Flash control register.

Table 25. WRP protection

| WRP registers values (x=1 y= A/B) | WRP protection area |
|---|---|
| WRPx _y _STRT = WRPx _y _END | Page WRPx _y is protected. |
| WRPx _y _STRT > WRPx _y _END | No WRP area. |
| WRPx _y _STRT < WRPx _y _END | The pages from WRPx _y _STRT to WRPx _y _END are protected. |

4.5.4 Securable memory area

The Securable memory area defines an area of code which can be executed only once at boot, and never again unless a new reset occurs.

The main purpose of the Securable memory area is to protect a specific part of Flash memory against undesired access. This allows implementing software security services such as secure key storage or safe boot. Securable memory area is located in the Main Flash memory. It is dedicated to executing trusted code. When not secured, the Securable memory behaves like the remainder of Main Flash memory. When secured (the SEC PROT1 bit of the FLASH_CR register set), any attempt to program or erase in a secure memory area generates a write protection error (WRPERR flag is set) and any attempt to read from it generates a read error (RDERR flag is set).

The size of the securable memory area is defined by the SEC_SIZE1[8:0] bitfield of the FLASH_SEC register. It can be modified only in RDP Level 0. Its content is erased upon changing from RDP Level 1 to Level 0, even if it overlaps with PCROP pages.

The securable memory area is defined from the address: Bank base address (included) to the address: Bank base address + (0x800 * SEC_SIZE1) (excluded).

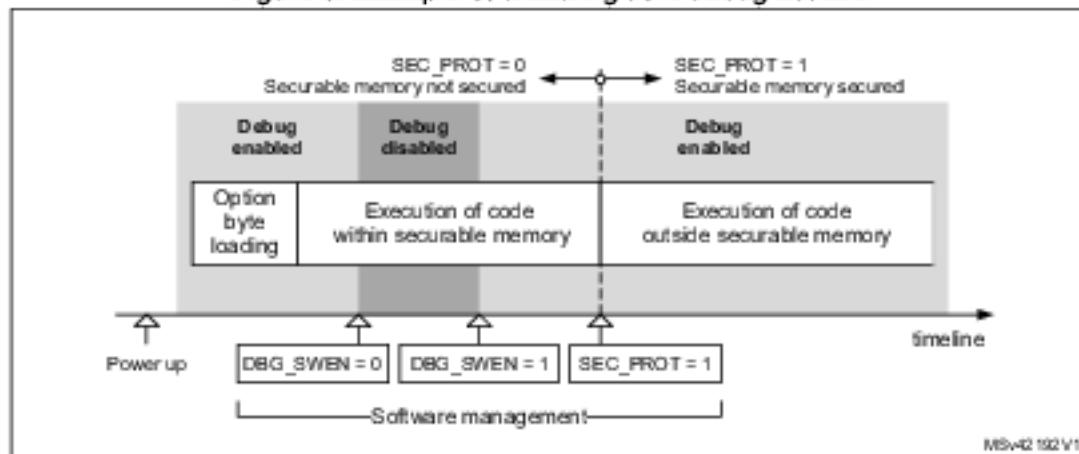
4.5.5 Disabling core debug access

For executing sensitive code or manipulating sensitive data in Securable memory area, the debug access to the core can temporarily be disabled.

In RDP level 2, the debugger is disabled by hardware, but in other RDP levels, the debugger can be disabled by software using the bit DBG_SWEN in the FLASH_ACR register.

Figure 11 gives an example of managing DBG_SWEN and SEC_PROT bits.

Figure 8. Example of disabling core debug access



4.5.6 Forcing boot from Flash memory

To increase the security and establish a chain of trust, the BOOT_LOCK option bit of the FLASH_SEC1R/FLASH_SEC2R register allows forcing the system to boot from the Main Flash memory regardless the other boot options. It is always possible to set the BOOT_LOCK bit. However, it is possible to reset it only when:

- RDP is set to Level 0, or
- RDP is set to Level 1, while Level 0 is requested and a full mass-erase is performed.

4.6 FLASH interrupts

Table 26. Flash interrupt request

| Interrupt event | Event flag | Event flag/interrupt clearing method | Interrupt enable control bit |
|------------------|----------------------|--------------------------------------|------------------------------|
| End of operation | EOP ⁽¹⁾ | Write EOP=1 | EOPIE |
| Operation error | OPERR ⁽²⁾ | Write OPERR=1 | ERRIE |
| Read error | RDERR | Write RDERR=1 | RDERRIE |
| ECC correction | ECCC | Write ECCC=1 | ECCCIE |

1. EOP is set only if EOPIE is set.

2. OPERR is set only if ERRIE is set.

4.7 FLASH registers

4.7.1 Flash access control register (FLASH_ACR)

Address offset: 0x00

Reset value: 0x0004 0601

Access: no wait state, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|----------|--------|-------|-------|------|------|--------|------|------|------|------|------|------|------|--------------|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | SLEEP_PD | RUN_PD | DCRST | ICRST | OCEN | ICEN | PRFTEN | Res. | LATENCY[3:0] |
| rw | rw | rw | rw | rw | rw | rw | rw | | | | | rw | rw | rw | rw |

Bits 31:19 Reserved, must be kept at reset value.

Bit 18 **DBG_SWEN**: Debug software enable

SW may use this bit to enable/disable the debugger.

0: Debugger disabled

1: Debugger enabled

Bits 17:15 Reserved, must be kept at reset value.

Bit 14 **SLEEP_PD**: Flash Power-down mode during Sleep or Low-power sleep mode

This bit determines whether the flash memory is in Power-down mode or Idle mode when the device is in Sleep or Low-power sleep mode.

0: Flash in Idle mode during Sleep and Low-power sleep modes

1: Flash in Power-down mode during Sleep and Low-power sleep modes

Caution: The flash must not be put in power-down while a program or an erase operation is on-going.

Bit 13 **RUN_PD**: Flash Power-down mode during Run or Low-power run mode

This bit is write-protected with FLASH_PDKEYR.

This bit determines whether the flash memory is in Power-down mode or Idle mode when the device is in Run or Low-power run mode. The flash memory can be put in power-down mode only when the code is executed from RAM. The Flash must not be accessed when RUN_PD is set.

0: Flash in Idle mode

1: Flash in Power-down mode

Caution: The flash must not be put in power-down while a program or an erase operation is on-going.

Bit 12 **DCRST**: Data cache reset

0: Data cache is not reset

1: Data cache is reset

This bit can be written only when the data cache is disabled.

| | |
|----------|--|
| Bit 11 | ICRST : Instruction cache reset |
| 0: | Instruction cache is not reset |
| 1: | Instruction cache is reset |
| | This bit can be written only when the instruction cache is disabled. |
| Bit 10 | DCEN : Data cache enable |
| 0: | Data cache is disabled |
| 1: | Data cache is enabled |
| Bit 9 | ICEN : Instruction cache enable |
| 0: | Instruction cache is disabled |
| 1: | Instruction cache is enabled |
| Bit 8 | PRFTEN : Prefetch enable |
| 0: | Prefetch disabled |
| 1: | Prefetch enabled |
| Bits 7:4 | Reserved, must be kept at reset value. |
| Bits 3:0 | LATENCY[3:0] : Latency |
| | These bits represent the ratio of the SYSCLK (system clock) period to the Flash access time. |
| 0000: | Zero wait state |
| 0001: | One wait state |
| 0010: | Two wait states |
| 0011: | Three wait states |
| 0100: | Four wait states |
| ... | 1111: Fifteen wait states |

4.7.2 Flash Power-down key register (FLASH_PDKEYR)

Address offset: 0x04

Reset value: 0x0000 0000

Access: no wait state, word access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| PDKEYR[31:16] | | | | | | | | | | | | | | | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PDKEYR[15:0] | | | | | | | | | | | | | | | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

Bits 31:0 **PDKEYR[31:0]**: Power-down in Run mode Flash key

The following values must be written consecutively to unlock the RUN_PD bit in FLASH_ACR:

PDKEY1: 0x04 15 2637

PDKEY2: 0xFAFB FCFD

4.7.3 Flash key register (FLASH_KEYR)

Address offset: 0x08

Reset value: 0x0000 0000

Access: no wait state, word access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| KEYR[31:16] | | | | | | | | | | | | | | | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| KEYR[15:0] | | | | | | | | | | | | | | | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

Bits 31:0 KEYR[31:0]: Flash key

The following values must be written consecutively to unlock the FLASH_CTR register allowing flash programming/erasing operations:

KEY1: 0x4567 0123

KEY2: 0xCDEF 89AB

4.7.4 Flash option key register (FLASH_OPTKEYR)

Address offset: 0x0C

Reset value: 0x0000 0000

Access: no wait state, word access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| OPTKEYR[31:16] | | | | | | | | | | | | | | | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OPTKEYR[15:0] | | | | | | | | | | | | | | | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

Bits 31:0 OPTKEYR[31:0]: Option byte key

The following values must be written consecutively to unlock the FLASH_OPTR register allowing option byte programming/erasing operations:

KEY1: 0x0819 2A3B

KEY2: 0x4C5D 6E7F

4.7.5 Flash status register (FLASH_SR)

Address offset: 0x10

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access

| | | | | | | | | | | | | | | | | |
|-------------|-----------|------|------|------|------|-------------|-------------|------------|------------|------------|------------|-------------|------|------------|-------|--|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | |
| Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | BSY | |
| | | | | | | | | | | | | | | | r | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| OPTV ERR | RD ERR | Res. | Res. | Res. | Res. | FAST ERR | MISS ERR | PGS ERR | SIZ ERR | PGA ERR | WRP ERR | PROG ERR | Res. | OIP ERR | EOP | |
| re_w1 | rc_w1 | | | | | re_w1 | rc_w1 | re_w1 | rc_w1 | re_w1 | rc_w1 | re_w1 | | re_w1 | re_w1 | |

Bits 31:17 Reserved, must be kept at reset value.

Bit 16 **BSY**: Busy

This indicates that a Flash operation is in progress. This is set on the beginning of a Flash operation and reset when the operation finishes or when an error occurs.

Bit 15 **OPTVERR**: Option validity error

Set by hardware when the options read may not be the one configured by the user. If option haven't been properly loaded, OPTVERR is set again after each system reset.

Cleared by writing 1.

Bit 14 **RDERR**: PCROP read error

Set by hardware when an address to be read through the D-bus belongs to a read protected area of the flash (PCROP protection). An interrupt is generated if RDERRIE is set in FLASH_C.R.

Cleared by writing 1.

Bits 13:10 Reserved, must be kept at reset value.

Bit 9 **FASTERR**: Fast programming error

Set by hardware when a fast programming sequence (activated by FSTPG) is interrupted due to an error (alignment, size, write protection or data miss). The corresponding status bit (PGAERR, SIZERR, WRPERR or MISSERR) is set at the same time.

Cleared by writing 1.

Bit 8 **MISSERR**: Fast programming data miss error

In Fast programming mode, 32 double words must be sent to flash successfully, and the new data must be sent to the flash logic control before the current data is fully programmed. MISSERR is set by hardware when the new data is not present in time.

Cleared by writing 1.

Bit 7 **PGSERR**: Programming sequence error

Set by hardware when a write access to the Flash memory is performed by the code while PG or FSTPG have not been set previously. Set also by hardware when PROGERR, SIZERR, PGAERR, WRPERR, MISSERR or FASTERR is set due to a previous programming error.

Cleared by writing 1.

Bit 6 **SIZERR**: Size error

Set by hardware when the size of the access is a byte or half-word during a program or a fast program sequence. Only double word programming is allowed (consequently: word access).

Cleared by writing 1.

Bit 5 **PGAERR**: Programming alignment error

Set by hardware when the data to program cannot be contained in the same 64-bit Flash memory row in case of standard programming, or if there is a change of page during fast programming.

Cleared by writing 1.

Bit 4 **WRPERR**: Write protection error

Set by hardware when an address to be erased/programmed belongs to a write-protected part (by WRP, PCROP or RDP level 1) of the Flash memory.

Cleared by writing 1.

Bit 3 **PROGERR**: Programming error

Set by hardware when a double-word address to be programmed contains a value different from 0xFFFF FFFF FFFF FFFF before programming, except if the data to write is '0x0000 0000 0000 0000'.

Cleared by writing 1.

Bit 2 Reserved, must be kept at reset value.

Bit 1 **OPERR**: Operation error

Set by hardware when a Flash memory operation (program / erase) completes unsuccessfully.

This bit is set only if error interrupts are enabled (ERRIE = 1).

Cleared by writing '1'.

Bit 0 **EOP**: End of operation

Set by hardware when one or more Flash memory operation (programming / erase) has been completed successfully.

This bit is set only if the end of operation interrupts are enabled (EOPIE = 1).

Cleared by writing 1.

4.7.6 Flash control register (FLASH_CR)

Address offset: 0x14

Reset value: 0xC000 0000

Access: no wait state when no Flash memory operation is ongoing, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|----------|------|-----------|------------|----------|--------|--------|------|------|------|------|------|-------|----------|-------|
| LOCK | OPT LOCK | Res. | SEC PROT1 | OBL LAUNCH | RDERR IE | ERR IE | EOP IE | Res. | Res. | Res. | Res. | Res. | FSTPG | OPT STRT | START |
| 0 | 0 | | 0 | 0_w1 | rw | rw | rw | | | | | | rw | 0 | 0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. Res. Res. Res. Res. PNB[7:0] MER1 PER PG | | | | | | | | | | | | | | | |
| | | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Bit 31 LOCK: FLASH_CR Lock

This bit is set only. When set, the FLASH_CR register is locked. It is cleared by hardware after detecting the unlock sequence.

In case of an unsuccessful unlock operation, this bit remains set until the next system reset.

Bit 30 OPTLOCK: Options Lock

This bit is set only. When set, all bits concerning user option in FLASH_CR register and so option page are locked. This bit is cleared by hardware after detecting the unlock sequence. The LOCK bit must be cleared before doing the unlock sequence for OPTLOCK bit.

In case of an unsuccessful unlock operation, this bit remains set until the next reset.

Bit 29 Reserved, must be kept at reset value.**Bit 28 SEC_PROT1: Securable memory area protection bit.**

This bit is set to lock the access to the Securable memory area. It is set by software when exiting the Securable memory area, and can only be written once.

Bit 27 OBL_LAUNCH: Force the option byte loading

When set to 1, this bit forces the option byte reloading. This bit is cleared only when the option byte loading is complete. It cannot be written if OPTLOCK is set.

0: Option byte loading complete

1: Option byte loading requested

Bit 26 RDERRIE: PCROP read error interrupt enable

This bit enables the interrupt generation when the RDERR bit in the FLASH_SR is set to 1.

0: PCROP read error interrupt disabled

1: PCROP read error interrupt enabled

Bit 25 ERRIE: Error interrupt enable

This bit enables the interrupt generation when the OPERR bit in the FLASH_SR is set to 1.

0: OPERR error interrupt disabled

1: OPERR error interrupt enabled

Bit 24 EOPIE: End of operation interrupt enable

This bit enables the interrupt generation when the EOP bit in the FLASH_SR is set to 1.

0: EOP interrupt disabled

1: EOP interrupt enabled

Bits 23:19 Reserved, must be kept at reset value.**Bit 18 FSTPG: Fast programming**

0: Fast programming disabled

1: Fast programming enabled

Bit 17 OPTSTRT: Options modification start

This bit triggers an options operation when set.

This bit is set only by software, and is cleared when the BSY bit is cleared in FLASH_SR.

Bit 16 START: Start

This bit triggers an erase operation when set. If MER1, MER2 and PER bits are reset and the STRT bit is set, an unpredictable behavior may occur without generating any error flag. This condition should be forbidden.

This bit is set only by software, and is cleared when the BSY bit is cleared in FLASH_SR.

Bits 15:11 Reserved, must be kept at reset value.

Bits 10:3 PNB[7:0]: Page number selection

These bits select the page to erase:

00000000: page 0

00000001: page 1

...

11111111: page 255

Bit 2 MER1: Mass erase

This bit triggers the mass erase (all user pages) when set.

Bit 1 PER: Page erase

0: page erase disabled

1: page erase enabled

Bit 0 PG: Programming

0: Flash programming disabled

1: Flash programming enabled

4.7.7 Flash ECC register (FLASH_ECCR)

Address offset: 0x18

Reset value: 0x0000 0000

Access: no wait state when no Flash memory operation is ongoing, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---------------------------------------|-------|------|------|------|------|------|---------|------|----------|------|------|------|-----------------|----|----|
| ECCD | ECCC | Res. | Res. | Res. | Res. | Res. | ECCC IE | Res. | SYSF_ECC | Res. | Res. | Res. | ADDR_ECC[18:16] | | |
| rc_w1 | rc_w1 | | | | | | rw | | r | | | | r | r | r |
| 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 | | | | | | | | | | | | | | | |
| ADDR_ECC[15:0] | | | | | | | | | | | | | | | |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

Bit 31 ECCD: ECC detection

Set by hardware when two ECC errors have been detected (only if ECCC/ECCD are previously cleared). When this bit is set, a NMI is generated.

Cleared by writing 1.

Bit 30 ECCC: ECC correction

Set by hardware when one ECC error has been detected and corrected (only if ECCC/ECCC2/ECCD/ECCD2 are previously cleared). An interrupt is generated if ECCC IE is set.

Cleared by writing 1.

Bits 29:25 Reserved, must be kept at reset value.

- Bit 24 **ECCCIE**: ECC correction interrupt enable
 0: ECCC interrupt disabled
 1: ECCC interrupt enabled.
 This bit enables the interrupt generation when the ECCC bit in the FLASH_ECCR register is set.
- Bit 23 Reserved, must be kept at reset value.
- Bit 22 **SYSF_ECC**: System Flash ECC fail
 This bit indicates that the ECC error correction or double ECC error detection is located in the System Flash.
- Bits 21:19 Reserved, must be kept at reset value.
- Bits 18:0 **ADDR_ECC[18:0]**: ECC fail address
 This bit indicates which address in the Flash memory is concerned by the ECC error correction or by the double ECC error detection.

4.7.8 Flash option register (FLASH_OPTR)

Address offset: 0x20

Reset value: 0xFXXX XXXX. Register bits are loaded with values from Flash memory at OBL.

Access: no wait state when no Flash memory operation is on going, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-----|-----------|----------------|-----------|--------|-------------|--------------|--------|--------|-----------|-----|-----|---------|------------|-----------|---------|
| Res | IRHEN | NRST_MODE[1:0] | | nBOOT0 | nSWBOOT0 | CCM_SRAM_RST | SRAM_P | nBOOT1 | PB4_PUPEN | Res | Res | WWDG_SW | IWDG_STDBY | IWDG_STOP | IWDG_SW |
| | rw | rw | rw | rw | rw | rw | rw | rw | rw | | | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res | nRST_SHDW | nRST_STDBY | nRST_STOP | Res | BORLEV[2:0] | | | | RDP[7:0] | | | | | | |
| | rw | rw | rw | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Bit 31 Reserved, must be kept at reset value.

Bit 30 **IRHEN**: Internal reset holder enable bit

- 0: Internal resets are propagated as simple pulse on NRST pin
 1: Internal resets drives NRST pin low until it is seen as low level

Bits 29:28 **NRST_MODE[1:0]**:

- 00: Reserved
- 01: Reset Input only: a low level on the NRST pin generates system reset, internal RESET not propagated to the NSRT pin
- 10: GPIO: standard GPIO pad functionality, only internal RESET possible
- 11: Bidirectional reset: NRST pin configured in reset input/output mode (legacy mode)

Bit 27 **nBOOT0**: nBOOT0 option bit

- 0: nBOOT0 = 0
 1: nBOOT0 = 1

- Bit 26 **nSWBOOT0**: Software BOOT0
0: BOOT0 taken from the option bit nBOOT0
1: BOOT0 taken from PB8/BOOT0 pin
- Bit 25 **CCMSRAM_RST**: CCM SRAM Erase when system reset
0: CCM SRAM erased when a system reset occurs
1: CCM SRAM is not erased when a system reset occurs
- Bit 24 **SRAM_PE**: SRAM1 and CCM SRAM parity check enable
0: SRAM1 and CCM SRAM parity check enable
1: SRAM1 and CCM SRAM parity check disable
- Bit 23 **nBOOT1**: Boot configuration
Together with the BOOT0 pin, this bit selects boot mode from the Flash main memory, SRAM1 or the System memory. Refer to [Section 2.6: Boot configuration](#).
- Bit 22 **PB4_PUPEN**: PB4 pull-up enable
0: USB power delivery dead-battery enabled/ NJTRST pull-up deactivated
1: USB power delivery dead-battery disabled/ NJTRST pull-up activated
Note: Only for Category 4 devices (otherwise Reserved)
- Bits 21:20 Reserved, must be reset value.
- Bit 19 **WWDG_SW**: Window watchdog selection
0: Hardware window watchdog
1: Software window watchdog
- Bit 18 **IWDG_STDBY**: Independent watchdog counter freeze in Standby mode
0: Independent watchdog counter is frozen in Standby mode
1: Independent watchdog counter is running in Standby mode
- Bit 17 **IWDG_STOP**: Independent watchdog counter freeze in Stop mode
0: Independent watchdog counter is frozen in Stop mode
1: Independent watchdog counter is running in Stop mode
- Bit 16 **IWDG_SW**: Independent watchdog selection
0: Hardware independent watchdog
1: Software independent watchdog
- Bit 15 Reserved, must be reset value.
- Bit 14 **nRST_SHDW**
0: Reset generated when entering the Shutdown mode
1: No reset generated when entering the Shutdown mode
- Bit 13 **nRST_STDBY**
0: Reset generated when entering the Standby mode
1: No reset generated when entering the Standby mode
- Bit 12 **nRST_STOP**
0: Reset generated when entering the Stop mode
1: No reset generated when entering the Stop mode

Bit 11 Reserved, must be reset value.

Bits 10:8 BOR_lev: BOR reset Level

These bits contain the VDD supply level threshold that activates/releases the reset.

- 000: BOR Level 0. Reset level threshold is around 1.7 V
- 001: BOR Level 1. Reset level threshold is around 2.0 V
- 010: BOR Level 2. Reset level threshold is around 2.2 V
- 011: BOR Level 3. Reset level threshold is around 2.5 V
- 100: BOR Level 4. Reset level threshold is around 2.8 V

Bits 7:0 RDP: Read protection level

0xAA: Level 0, read protection not active

0xCC: Level 2, chip read protection active

Others: Level 1, memories read protection active

Note: Take care about PCR OP_RDP configuration in Level 1. Refer to [Section : Level 1: Read protection](#) for more details.

4.7.9 Flash PCROP1 Start address register (FLASH_PCROP1SR)

Address offset: 0x24

Reset value: 0xFFFF XXXX

Register bits are loaded with values from Flash memory at OBL.

Access: no wait state when no Flash memory operation is ongoing, word access.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| PCROP1_STRT[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 PCROP1_STRT[15:0]: PCROP area start offset

PCROP1_STRT contains the first double-word of the PCROP area.

4.7.10 Flash PCROP1 End address register (FLASH_PCROP1ER)

Address offset: 0x28

Reset value: 0x0000 XXXX

Register bits are loaded with values from Flash memory at OBL.

Access: no wait state when no Flash memory operation is on going, word, half-word access.
PCROP_RDP bit can be accessed with byte access.

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| PCROP_RDP | Res. |
| Si | | | | | | | | | | | | | | | | |
| PCROP1-END[15:0] | | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Bit 31 PCROP_RDP: PCROP area preserved when RDP level decreased

This bit is set only. It is reset after a full mass erase due to a change of RDP from Level 1 to Level 0.

0: PCROP area is not erased when the RDP level is decreased from Level 1 to Level 0.

1: PCROP area is erased when the RDP level is decreased from Level 1 to Level 0 (full mass erase).

Bits 30:16 Reserved, must be kept at reset value.

Bits 15:0 PCROP1-END[15:0]: PCROP area end offset

PCROP1-END contains the last double-word of the PCROP area.

4.7.11 Flash WRP area A address register (FLASH_WRP1AR)

Address offset: 0x2C

Reset value: 0x00XX 00XX

Register bits are loaded with values from Flash memory at OBL.

Access: no wait state when no Flash memory operation is on going, word, half-word and byte access

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|------|------|------|------|------|------|------|------|-----------------|----|----|----|----|----|----|----|
| Res. | WRP1A-END[7:0] | | | | | | | |
| | | | | | | | | | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Res. | WRP1A-STRT[7:0] | | | | | | | |
| | | | | | | | | | rw | rw | rw | rw | rw | rw | rw | rw |

Bits 31:24 Reserved, must be kept at reset value.

Bits 23:16 **WRP1A_END[7:0]**: WRP first area "A" end offset
 WRP1A_END contains the last page of WRP first area.

Bits 15:8 Reserved, must be kept at reset value.

Bits 7:0 **WRP1A_STRT[7:0]**: WRP first area "A" start offset

WRP1A_STRT contains the first page of WRP first area.

4.7.12 Flash WRP area B address register (FLASH_WRP1BR)

Address offset: 0x30

Reset value: 0x00XX 00XX

Register bits are loaded with values from Flash memory at OBL.

Access: no wait state when no Flash memory operation is on going, word, half-word and byte access

| | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res | WRP1B_END[7:0] |
| | | | | | | | | | | | | | | | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res | WRP1B_STRT[7:0] |
| | | | | | | | | | | | | | | | rw |

Bits 31:24 Reserved, must be kept at reset value.

Bits 23:16 **WRP1B_END[7:0]**: WRP second area "B" end offset

WRP1B_END contains the last page of the WRP second area.

Bits 15:8 Reserved, must be kept at reset value.

Bits 7:0 **WRP1B_STRT[7:0]**: WRP second area "B" start offset

WRP1B_STRT contains the first page of the WRP second area.

4.7.13 Flash Securable area register (FLASH_SEC1R)

Address offset: 0x70

Reset value: 0xFFFF FXXX

Access: no wait state when no Flash memory operation is on going, word, half-word and byte access

| | | | | | | | | | | | | | | | |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res |
| | | | | | | | | | | | | | | | BOOT_LOCK |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res |
| | | | | | | | | | | | | | | | rw |
| SEC_SIZE1[8:0] | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | rw |

Bits 31:17 Reserved, must be kept at reset value.

Bit 16 **BOOT_LOCK**: used to force boot from user Flash area

0: Boot based on the pad/option bit configuration

1: Boot forced from Main Flash memory

Bits 15:9 Reserved, must be kept at reset value.

Bits 8:0 **SEC_SIZE1[8:0]**: sets the number of pages used in the Securable area.

Securable area starts at @ 0x0800 0000 and its size is SEC_SIZE1 * page size.

This field can be changed in level0 only.

Any attempt to modify in level1 silently fails, and does not change register value.

4.7.14 FLASH register map

Table 27. Flash interface - register map and reset values

| Offset | Register | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | |
|--------|---------------|-----------|---------|------|------|------|------|------|------|------|------|------|------|------|----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|--------------|------|---|---|---|--|--|
| 0x00 | FLASH_ACR | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | 1 | DBG_SWEN | 18 | | | | | | | | | | | | | | | LATENCY[3:0] | | | | | | |
| | Reset value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 | 0 | 0 | 1 | 0 | | |
| 0x04 | FLASH_PKEYR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Reset value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | |
| 0x08 | FLASH_KEYR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Reset value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | |
| 0x0C | FLASH_OPTKEYR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Reset value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | |
| 0x10 | FLASH_SR | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | | | | | |
| | Reset value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x14 | FLASH_CR | LOC_K | OPTLOCK | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Reset value | 1 | 1 | EDDC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x18 | FLASH_ECCR | EDDC_D | EDDC_C | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Reset value | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x20 | FLASH_OPTR | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | | | | | |
| | Reset value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x24 | FLASH_PCRP1SR | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | | | | | |
| | Reset value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x28 | FLASH_PCRP1ER | PCRDP_RDP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Reset value | x | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x2C | FLASH_WRP1AR | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | | | | | |
| | Reset value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 27. Flash interface - register map and reset values (continued)

| Offset | Register | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----------------|-----|-----|
| 0x30 | FLASH_WRP1B_R | R/W | X | X | X | X | X | X | X | X | R/W | R/W | R/W |
| | Reset value | | | | | | | | | X | X | X | X | X | X | X | X | | | | | | | | X | X | X | X | X | X | X | X | X |
| 0x70 | FLASH_SEC1R | R/W | SIEC_SIZE1[8:0] | | |
| | Reset value | | | | | | | | | | | | | | | | | X | | | | | | | X | X | X | X | X | X | X | X | X |

Refer to [Section 2.2 on page 81](#) for the register boundary addresses.

5 Embedded Flash memory (FLASH) for category 2 devices

5.1 Introduction

The Flash memory interface manages CPU AHB ICode and DCode accesses to the Flash memory. It implements the erase and program Flash memory operations and the read and write protection mechanisms.

The Flash memory interface accelerates code execution with a system of instruction prefetch and cache lines.

5.2 FLASH main features

- Up to 128 Kbyte of Flash memory (single bank).
- Flash memory read operations with 64 bits data width
- Page erase and mass erase

Flash memory interface features:

- Flash memory read operations
- Flash memory program/erase operations
- Read protection activated by option (RDP)
- 2 Write protection areas selected by option
- Proprietary code read protection areas defined by option
- Securable memory areas defined by option
- Prefetch on ICODE
- Instruction Cache: 32 cache lines of 4 x 64 bits on ICode (1 KB RAM)
- Data Cache: 8 cache lines of 4 x 64 bits on DCode (256B RAM)
- Error Code Correction ECC: 8 bits per 64-bit double-word
 - $8 + 64 = 72$ bits, 2 bits detection, 1 bit correction
- Option byte loader
- Low-power mode

5.3 FLASH functional description

5.3.1 Flash memory organization

The Flash memory has the following main features:

- Capacity up to 128 Kbytes (read width of 64-bits)
- 128 KB organized in one single bank for main memory
- Page size of 2 Kbyte
- 72 bits wide data read (64 bits plus 8 ECC bits)
- Page and Mass erase
- Each page is composed of 8 rows of 256 bytes
- An Information block containing:
 - System memory from which the device boots in System memory boot mode. The area is reserved for use by STMicroelectronics and contains the bootloader that is used to reprogram the Flash memory through one of the following interfaces: USART, SPI, I2C, USB. It is programmed by STMicroelectronics when the device is manufactured, and protected against spurious write/erase operations. For further details, please refer to the AN2606 available from www.st.com.
 - 1 Kbyte (128 double word) OTP (one-time programmable) bytes for user data. The OTP data cannot be erased and can be written only once. If only one bit is at 0, the entire double word cannot be written anymore, even with the value 0x0000 0000 0000.
 - Option bytes for user configuration.

The memory organization is based on a main area and an information block as shown in [Table 28](#).

Table 28. Flash module - 64/128 Kbytes organization (64 bits read width)

| Flash area | Flash memory addresses | Size (bytes) | Name |
|---|---------------------------|--------------|---------------|
| Main memory 64/128 Kbytes ⁽¹⁾ | 0x0800 0000 - 0x0800 07FF | 2 K | Page 0 |
| | 0x0800 0800 - 0x0800 0FFF | 2 K | Page 1 |
| | 0x0800 1000 - 0x0800 17FF | 2 K | Page 2 |
| | 0x0800 1800 - 0x0800 1FFF | 2 K | Page 3 |
| | - | - | - |
| | - | - | - |
| | - | - | - |
| Information block | 0x0801 F800 - 0x0801 FFFF | 2 K | Page 63 |
| | 0x1FFF 0000 - 0x1FFF 6FFF | 28 K | System memory |
| | 0x1FFF 7000 - 0x1FFF 73FF | 1 K | OTP area |
| | 0x1FFF 7800 - 0x1FFF 782F | 48 | Option bytes |

1. For 64-Kbyte devices: from page 0 to page 31.

5.3.2 Error code correction (ECC)

Data in Flash memory are 72-bits words: 8 bits are added per double word (64 bits). The ECC mechanism supports:

- One error detection and correction
- Two errors detection

When one error is detected and corrected, the flag ECCC (ECC correction) is set in [Flash ECC register \(FLASH_ECCR\)](#). If ECCCIE is set, an interrupt is generated.

When two errors are detected, a flag ECCD (ECC detection) is set in FLASH_ECCR register. In this case, a NMI is generated.

When an ECC error is detected, the address of the failing double word is saved in ADDR_ECC[20:0] in the FLASH_ECCR register. ADDR_ECC[2:0] are always cleared.

When ECCC or ECCD is set, ADDR_ECC is not updated if a new ECC error occurs. FLASH_ECCR is updated only when ECC flags are cleared.

Note: For a virgin data: 0xFFFF FFFF FFFF FFFF, one error is detected and corrected but two errors detection is not supported.

When an ECC error is reported, a new read at the failing address may not generate an ECC error if the data is still present in the current buffer, even if ECCC and ECCD are cleared.

5.3.3 Read access latency

To correctly read data from Flash memory, the number of wait states (LATENCY) must be correctly programmed in the [Flash access control register \(FLASH_ACR\)](#) according to the frequency of the CPU clock (HCLK) and the internal voltage range of the device V_{CORE} . Refer to [Section 6.1.5: Dynamic voltage scaling management](#). Table 29 shows the correspondence between wait states and CPU clock frequency.

Table 29. Number of wait states according to CPU clock (HCLK) frequency

| Wait states (WS) (LATENCY) | HCLK (MHz) | | |
|-------------------------------|----------------------------------|-----------------------------------|--------------------|
| | V_{CORE} Range 1 boost mode | V_{CORE} Range 1 normal mode | V_{CORE} Range 2 |
| 0 WS (1 CPU cycles) | ≤ 34 | ≤ 30 | ≤ 12 |
| 1 WS (2 CPU cycles) | ≤ 68 | ≤ 60 | ≤ 24 |
| 2 WS (3 CPU cycles) | ≤ 102 | ≤ 90 | ≤ 26 |
| 3 WS (4 CPU cycles) | ≤ 136 | ≤ 120 | - |
| 4 WS (5 CPU cycles) | ≤ 170 | ≤ 150 | - |

After reset, the CPU clock frequency is 16 MHz and 1 wait state (WS) is configured in the FLASH_ACR register.

When changing the CPU frequency, the following software sequences must be applied in order to tune the number of wait states needed to access the Flash memory:

Increasing the CPU frequency:

1. Program the new number of wait states to the LATENCY bits in the *Flash access control register (FLASH_ACR)*.
 2. Check that the new number of wait states is taken into account to access the Flash memory by reading the FLASH_ACR register.
 3. Analyze the change of CPU frequency change caused either by:
 - changing clocksource defined by SW bits in RCC_CFGR register
 - or by CPU clock prescaler defined by HPRE bits in RCC_CFGR
- If some of above two steps decreases the CPU frequency, firstly perform this step and then the rest. Otherwise modify The CPU clock source by writing the SW bits in the RCC_CFGR register and then (if needed) modify the CPU clock prescaler by writing the HPRE bits in RCC_CFGR.
4. Check that the new CPU clock source or/and the new CPU clock prescaler value is/are taken into account by reading the clock source status (SWS bits) or/and the AHB prescaler value (HPRE bits), respectively, in the RCC_CFGR register.

Decreasing the CPU frequency:

1. Modify the CPU clocksource by writing the SW bits in the RCC_CFGR register.
2. If needed, modify the CPU clock prescaler by writing the HPRE bits in RCC_CFGR.
3. Analyze the change of CPU frequency change caused either by:
 - changing clocksource defined by SW bits in RCC_CFGR register
 - or by CPU clock prescaler defined by HPRE bits in RCC_CFGR

If some of above two steps increases the CPU frequency, firstly perform another step and then this step. Otherwise modify The CPU clock source by writing the SW bits in the RCC_CFGR register and then (if needed) modify the CPU clock prescaler by writing the HPRE bits in RCC_CFGR.

4. Check that the new CPU clock source or/and the new CPU clock prescaler value is/are taken into account by reading the clock source status (SWS bits) or/and the AHB prescaler value (HPRE bits), respectively, in the RCC_CFGR register.
5. Program the new number of wait states to the LATENCY bits in *Flash access control register (FLASH_ACR)*.
6. Check that the new number of wait states is used to access the Flash memory by reading the FLASH_ACR register.

5.3.4 Adaptive real-time memory accelerator (ART Accelerator)

The proprietary Adaptive real-time (ART) memory accelerator is optimized for STM32 industry-standard Arm® Cortex®-M4 with FPU processors. It balances the inherent performance advantage of the Arm® Cortex®-M4 with FPU over Flash memory technologies, which normally requires the processor to wait for the Flash memory at higher operating frequencies.

To release the processor full performance, the accelerator implements an instruction prefetch queue and branch cache which increases program execution speed from the 64-bit Flash memory. Based on CoreMark benchmark, the performance achieved thanks to the ART accelerator is equivalent to 0 wait state program execution from Flash memory at a CPU frequency up to 170 MHz.

Instruction prefetch

The Cortex®-M4 fetches the instruction over the ICode bus and the literal pool (constant/data) over the DCode bus. The prefetch block aims at increasing the efficiency of ICode bus accesses.

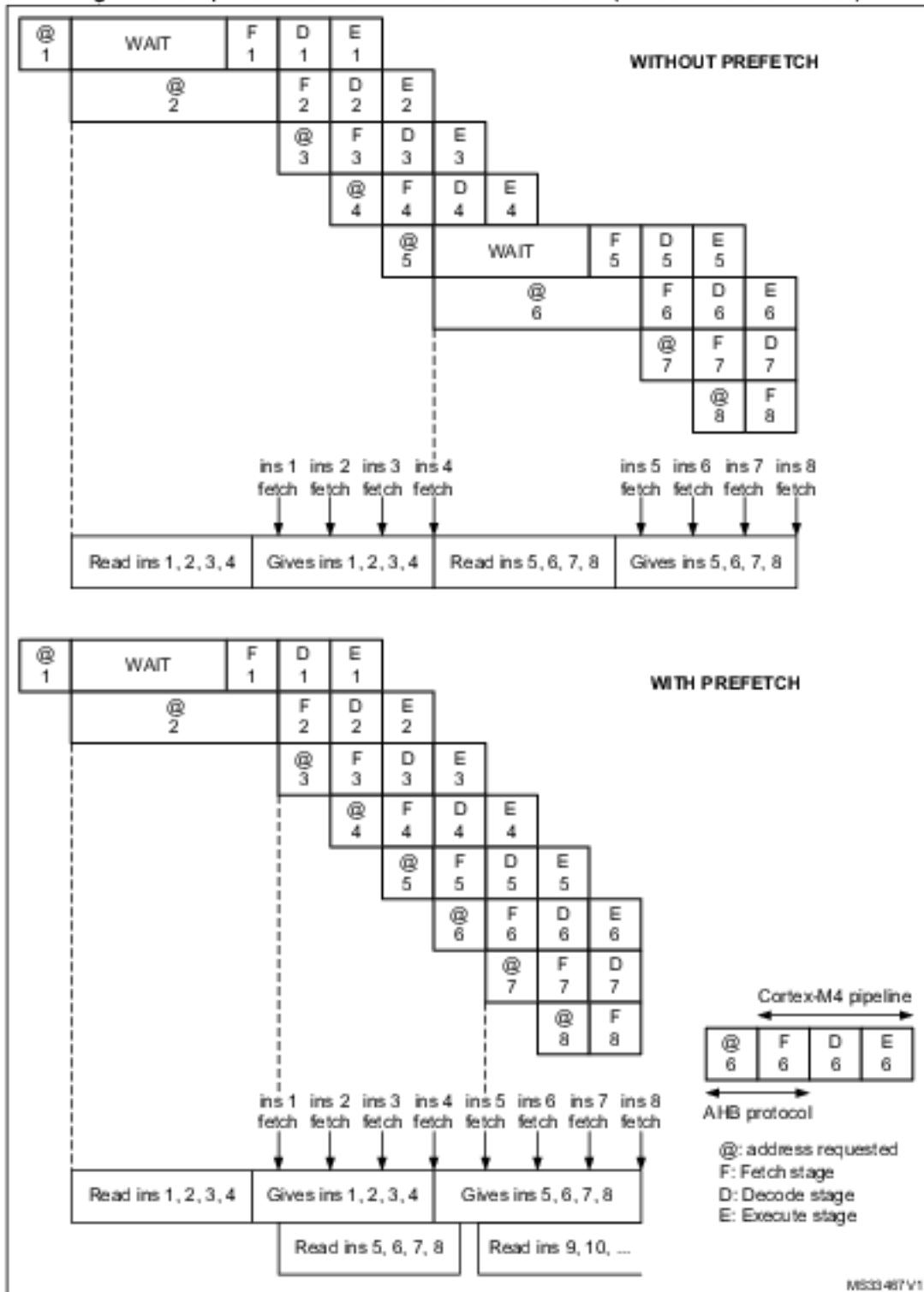
Each Flash memory read operation provides 64 bits from either two instructions of 32 bits or four instructions of 16 bits depending on the launched program. This 64-bits current instruction line is saved in a current buffer, and in case of sequential code, at least two CPU cycles are needed to execute the previous read instruction line.

Prefetch on the ICode bus can be used to read the next sequential instruction line from the Flash memory while the current instruction line is being requested by the CPU.

Prefetch is enabled by setting the PRFTEN bit in the [*Flash access control register \(FLASH_ACR\)*](#). This feature is useful if at least one wait state is needed to access the Flash memory.

[*Figure 9*](#) shows the execution of sequential 16-bit instructions with and without prefetch when 3 WS are needed to access the Flash memory.

Figure 9. Sequential 16-bit instructions execution (64-bit read data width)



When the code is not sequential (branch), the instruction may not be present in the currently used instruction line or in the prefetched instruction line. In this case (miss), the penalty in terms of number of cycles is at least equal to the number of wait states.

If a loop is present in the current buffer, no new flash access is performed.

Instruction cache memory (I-Cache)

To limit the time lost due to jumps, it is possible to retain 32 lines of 4 x 64 bits in an instruction cache memory. This feature can be enabled by setting the instruction cache enable (ICEN) bit in the *Flash access control register (FLASH_ACR)*. Each time a miss occurs (requested data not present in the currently used instruction line, in the prefetched instruction line or in the instruction cache memory), the line read is copied into the instruction cache memory. If some data contained in the instruction cache memory are requested by the CPU, they are provided without inserting any delay. Once all the instruction cache memory lines have been filled, the LRU (least recently used) policy is used to determine the line to replace in the instruction memory cache. This feature is particularly useful in case of code containing loops.

The Instruction cache memory is enable after system reset.

Data cache memory (D-Cache)

Literal pools are fetched from Flash memory through the DCode bus during the execution stage of the CPU pipeline. Each DCode bus read access fetches 64 bits which are saved in a current buffer. The CPU pipeline is consequently stalled until the requested literal pool is provided. To limit the time lost due to literal pools, accesses through the AHB databus DCode have priority over accesses through the AHB instruction bus ICode.

If some literal pools are frequently used, the data cache memory can be enabled by setting the data cache enable (DC EN) bit in the *Flash access control register (FLASH_ACR)*. This feature works like the instruction cache memory, but the retained data size is limited to 8 rows of 4*64 bits.

The Data cache memory is enable after system reset.

Note: The D-Cache is active only when data is requested by the CPU (not by DMA1 and DMA2).
Data in option bytes block are not cacheable.

5.3.5 Flash program and erase operations

The STM32G4 Series embedded Flash memory can be programmed using in-circuit programming or in-application programming.

The **in-circuit programming (ICP)** method is used to update the entire contents of the Flash memory, using the JTAG, SWD protocol or the boot loader to load the user application into the microcontroller. ICP offers quick and efficient design iterations and eliminates unnecessary package handling or socketing of devices.

In contrast to the ICP method, **in-application programming (IAP)** can use any communication interface supported by the microcontroller (I/Os, USB, CAN, UART, I²C, SPI, etc.) to download programming data into memory. IAP allows the user to re-program the Flash memory while the application is running. Nevertheless, part of the application has to have been previously programmed in the Flash memory using ICP.

The contents of the Flash memory are not guaranteed if a device reset occurs during a Flash memory operation.

The Flash erase and programming is only possible in the voltage scaling range 1. The VOS[1:0] bits in the PWR_CR1 must be programmed to 01b.

During a program/erase operation to the Flash memory, any attempt to read the Flash memory stalls the bus. The read operation proceeds correctly once the program/erase operation has completed.

Unlocking the Flash memory

After reset, write is not allowed in the *Flash control register (FLASH_CR)* to protect the Flash memory against possible unwanted operations due, for example, to electric disturbances. The following sequence is used to unlock this register:

1. Write KEY1 = 0x45670123 in the *Flash key register (FLASH_KEYR)*
2. Write KEY2 = 0xCDEF89AB in the FLASH_KEYR register.

Any wrong sequence locks up the FLASH_CR register until the next system reset. In the case of a wrong key sequence, a bus error is detected and a Hard Fault interrupt is generated.

The FLASH_CR register can be locked again by software by setting the LOCK bit in the FLASH_CR register.

Note: *The FLASH_CR register cannot be written when the BSY bit in the Flash status register (FLASH_SR) is set. Any attempt to write to it with the BSY bit set causes the AHB bus to stall until the BSY bit is cleared.*

5.3.6 Flash main memory erase sequences

The Flash memory erase operation can be performed at page level or on the whole Flash memory (Mass Erase). Mass Erase does not affect the Information block (system flash, OTP and option bytes).

Page erase

To erase a page, follow the procedure below:

1. Check that no Flash memory operation is ongoing by checking the BSY bit in the *Flash status register (FLASH_SR)*.
2. Check and clear all error programming flags due to a previous programming. If not, PGSER is set.
3. Set the PER bit and select the page to erase (PNB).
4. Set the STRT bit in the FLASH_CR register.
5. Wait for the BSY bit to be cleared in the FLASH_SR register.

Note: *The internal oscillator HSI16 (16 MHz) is enabled automatically when STRT bit is set, and disabled automatically when STRT bit is cleared, except if the HSI16 is previously enabled with HSION in RCC_CR register.*

If the page erase is part of write-protected area (by WRP or PCR OP), WRPPER is set and the page erase request is aborted.

Mass erase

To perform a Mass erase, follow the procedure below:

1. Check that no Flash memory operation is ongoing by checking the BSY bit in the FLASH_SR register.
2. Check and clear all error programming flags due to a previous programming. If not, PGERR is set.
3. Set the MER1 bit in the Flash control register (FLASH_CR).
4. Set the STRT bit in the FLASH_CR register.
5. Wait for the BSY bit to be cleared in the Flash status register (FLASH_SR).

Note:

The internal oscillator HSI16 (16 MHz) is enabled automatically when STRT bit is set, and disabled automatically when STRT bit is cleared, except if the HSI16 is previously enabled with HSION in RCC_CR register.

If the Flash memory contains a write-protected area (by WRP or PCROP), WRPERR is set and the mass erase request is aborted.

5.3.7 Flash main memory programming sequences

The Flash memory is programmed 72 bits at a time (64 bits + 8 bits ECC).

Programming in a previously programmed address is not allowed except if the data to write is full zero, and any attempt sets PROGERR flag in the *Flash status register (FLASH_SR)*.

It is only possible to program double word (2 x 32-bit data).

- Any attempt to write byte or half-word sets SIZERR flag in the FLASH_SR register.
- Any attempt to write a double word which is not aligned with a double word address sets PGAERR flag in the FLASH_SR register.

Standard programming

The Flash memory programming sequence in standard mode is as follows:

1. Check that no Flash main memory operation is ongoing by checking the BSY bit in the *Flash status register (FLASH_SR)*.
2. Check and clear all error programming flags due to a previous programming. If not, PGERR is set.
3. Set the PG bit in the *Flash control register (FLASH_CR)*.
4. Perform the data write operation at the desired memory address, inside main memory block or OTP area. Only double word can be programmed.
 - Write a first word in an address aligned with double word
 - Write the second word
5. Wait until the BSY bit is cleared in the FLASH_SR register.
6. Check that EOP flag is set in the FLASH_SR register (meaning that the programming operation has succeed), and clear it by software.
7. Clear the PG bit in the FLASH_SR register if there no more programming request anymore.

Note:

When the flash interface has received a good sequence (a double word), programming is automatically launched and BSY bit is set. The internal oscillator HSI16 (16 MHz) is enabled

automatically when PG bit is set, and disabled automatically when PG bit is cleared, except if the HSI16 is previously enabled with HSION in RCC_CR register.

If the user needs to program only one word, double word must be completed with the erase value 0xFFFF FFFF to launch automatically the programming.

ECC is calculated from the double word to program.

Fast programming

This mode allows to program a row (32 double words), and to reduce the page programming time by eliminating the need for verifying the flash locations before they are programmed and to avoid rising and falling time of high voltage for each double word. During fast programming, the CPU clock frequency (HCLK) must be at least 8 MHz.

Only the main memory can be programmed in Fast programming mode.

The Flash main memory programming sequence in standard mode is as follows:

1. Perform a mass erase. If not, PGSERR is set.
2. Check that no Flash main memory operation is ongoing by checking the BSY bit in the *Flash status register (FLASH_SR)*.
3. Check and clear all error programming flag due to a previous programming.
4. Set the FSTPG bit in *Flash control register (FLASH_CR)*.
5. Write the 32 double words to program a row. Only double words can be programmed:
 - Write a first word in an address aligned with double word
 - Write the second word.
6. Wait until the BSY bit is cleared in the FLASH_SR register.
7. Check that EOP flag is set in the FLASH_SR register (meaning that the programming operation has succeed), and clear it by software.
8. Clear the FSTPG bit in the FLASH_SR register if there no more programming request anymore.

Note:

If the flash is attempted to be written in Fast programming mode while a read operation is on going, the programming is aborted without any system notification (no error flag is set).

When the Flash interface has received the first double word, programming is automatically launched. The BSY bit is set when the high voltage is applied for the first double word, and it is cleared when the last double word has been programmed or in case of error. The internal oscillator HSI16 (16 MHz) is enabled automatically when FSTPG bit is set, and disabled automatically when FSTPG bit is cleared, except if the HSI16 is previously enabled with HSION in RCC_CR register.

The 32 double word must be written successively. The high voltage is kept on the flash for all the programming. Maximum time between two double words write requests is the time programming (around 2 x 25us). If a second double word arrives after this time programming, fast programming is interrupted and MISSERR is set.

High voltage mustn't exceed 8 ms for a full row between 2 erases. This is guaranteed by the sequence of 32 double words successively written with a clock system greater or equal to 8MHz. An internal time-out counter counts 7ms when Fast programming is set and stops the programming when time-out is over. In this case the FASTERR bit is set.

If an error occurs, high voltage is stopped and next double word to programmed is not programmed. Anyway, all previous double words have been properly programmed.

Programming errors

Several kind of errors can be detected. In case of error, the Flash operation (programming or erasing) is aborted.

- **PROGERR:** Programming Error

In standard programming: PROGERR is set if the word to write is not previously erased (except if the value to program is full zero).

- **SIZERR:** Size Programming Error

In standard programming or in fast programming: only double word can be programmed and only 32-bit data can be written. SIZERR is set if a byte or an half-word is written.

- **PGAERR:** Alignment Programming error

PGAERR is set if one of the following conditions occurs:

- In standard programming: the first word to be programmed is not aligned with a double word address, or the second word doesn't belong to the same double word address.
- In fast programming: the data to program doesn't belong to the same row than the previous programmed double words, or the address to program is not greater than the previous one.

- **PGSERR:** Programming Sequence Error

PGSERR is set if one of the following conditions occurs:

- In the standard programming sequence or the fast programming sequence: a data is written when PG and FSTPG are cleared.
- In the standard programming sequence or the fast programming sequence: MER1, and PER are not cleared when PG or FSTPG is set.
- In the fast programming sequence: the Mass erase is not performed before setting FSTPG bit.
- In the mass erase sequence: PG, FSTPG, and PER are not cleared when MER1 is set.
- In the page erase sequence: PG, FSTPG, MER1 are not cleared when PER is set.
- PGSERR is set also if PROGERR, SIZERR, PGAERR, WRPERR, MISSERR, FASTERR or PGSERR is set due to a previous programming error.

- **WRPERR:** Write Protection Error

WRPERR is set if one of the following conditions occurs:

- Attempt to program or erase in a write protected area (WRP) or in a PCROP area or in a Securable memory area.
- Attempt to perform an erase when one page or more is protected by WRP or PCROP.
- The debug features are connected or the boot is executed from SRAM or from System flash when the read protection (RDP) is set to Level 1.
- Attempt to modify the option bytes when the read protection (RDP) is set to Level 2.

- **MISSERR:** Fast Programming Data Miss Error

In fast programming: all the data must be written successively. MISSERR is set if the previous data programmation is finished and the next data to program is not written yet.

- **FASTERR:** Fast Programming Error

In fast programming: FASTERR is set if one of the following conditions occurs:

- When FSTPG bit is set for more than 7 ms which generates a time-out detection.
- When the fast programming has been interrupted by a MISSERR, PGAERR, WRPERR or SIZERR.

If an error occurs during a program or erase operation, one of the following error flags is set in the FLASH_SR register:

PROGERR, SIZERR, PGAERR, PGSERR, MISSERR (Program error flags),
WRPERR (Protection error flag)

In this case, if the error interrupt enable bit ERRIE is set in the [Flash status register \(FLASH_SR\)](#), an interrupt is generated and the operation error flag OPERR is set in the FLASH_SR register.

Note: *If several successive errors are detected (for example, in case of DMA transfer to the Flash memory), the error flags cannot be cleared until the end of the successive write requests.*

Programming and caches

If a Flash memory write access concerns some data in the data cache, the Flash write access modifies the data in the Flash memory and the data in the cache.

If an erase operation in Flash memory also concerns data in the data or instruction cache, you have to make sure that these data are rewritten before they are accessed during code execution. If this cannot be done safely, it is recommended to flush the caches by setting the DCRST and ICRST bits in the [Flash access control register \(FLASH_ACR\)](#).

Note: *The I/D cache should be flushed only when it is disabled (IDCEN = 0).*

5.4 FLASH option bytes

5.4.1 Option bytes description

The option bytes are configured by the end user depending on the application requirements. As a configuration example, the watchdog may be selected in hardware or software mode (refer to [Section 5.4.2: Option bytes programming](#)).

A double word is split up as follows in the option bytes:

Table 30. Option byte format

| 63-24 | 23-16 | 15-8 | 7-0 | 31-24 | 23-16 | 15-8 | 7-0 |
|----------------------------|----------------------------|----------------------------|----------------------------|---------------|---------------|---------------|---------------|
| Complemented option byte 3 | Complemented option byte 2 | Complemented option byte 1 | Complemented option byte 0 | Option byte 3 | Option byte 2 | Option byte 1 | Option byte 0 |

The organization of these bytes inside the information block is as shown in [Table 31: Option byte organization](#).

The option bytes can be read from the memory locations listed in [Table 31: Option byte organization](#) or from the Option byte registers:

- [Flash option register \(FLASH_OPTR\)](#)
- [Flash PCROP1 Start address register \(FLASH_PCROP1SR\)](#)
- [Flash PCROP1 End address register \(FLASH_PCROP1ER\)](#)
- [Flash WRP area A address register \(FLASH_WRP1AR\)](#)
- [Flash WRP area B address register \(FLASH_WRP1BR\)](#)

Table 31. Option byte organization

| Address | [63:56] | [55:48] | [47:40] | [39:32] | [31:24] | [23:16] | [15:8] | [7:0] |
|----------|----------------------|-----------------|------------------------------|------------------|---------|----------------------|----------|------------------------------|
| 1FFF7800 | USER OPT | | | RDP | | | USER OPT | |
| 1FFF7808 | Unused | | Unused and PCROP1_STRT[13:0] | | | Unused | | Unused and PCROP1_STRT[13:0] |
| 1FFF7810 | PCROP_RDP and Unused | | Unused and PCROP1_END[13:0] | | | PCROP_RDP and Unused | | Unused and PCROP1_END[13:0] |
| 1FFF7818 | Unused | WRP1A_END [5:0] | Unused | WRP1A_STRT [5:0] | Unused | WRP1A_END [5:0] | Unused | WRP1A_STRT [5:0] |
| 1FFF7820 | Unused | WRP2A_END [5:0] | Unused | WRP2A_STRT [5:0] | Unused | WRP2A_END [5:0] | Unused | WRP2A_STRT [5:0] |
| 1FFF7828 | Unused | BOOT_LOCK | Unused | SEC_SIZE1 | Unused | BOOT_LOCK | Unused | SEC_SIZE1 |

User and read protection option bytes

Flash memory address: 0x1FFF 7800

ST production value: 0xFFFF F8AA

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-----|---------|----------------|-----------|-----------|-------------|-------------|--------|------|----------|------|---------|------------|-----------|---------|----|----|
| Res | IRH_EN | NRST_MODE[1:0] | nBOOT0 | nSWBOOT0 | CCMSRAM_RST | SRAM_PE | nBOOT1 | Res. | Res. | Res. | WWDG_SW | IWDG_STDBY | IWDG_STOP | IWDG_SW | | |
| | r | r | r | r | r | r | r | r | | | r | r | r | r | r | r |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Res | nRST_SW | nRST_STDBY | nRST_STOP | nRST_STOP | Res | BDRLEV[2:0] | | | RDP[7:0] | | | | | | | |
| | r | r | r | | | r | r | r | r | r | r | r | r | r | r | r |

Bit 31 Reserved, must be kept at reset value.

Bit 30 **IRH_IN**: Internal reset holder for PG10

0: IRH disabled

1: IRH enabled

Bits 29:28 **NRST_MODE**: PG10 pad mode

00: Reset Input/Output

01: Reset Input only

10: GPIO

11: Reset Input/Output

Bit 27 **nBOOT0**: nBOOT0 option bit

0: nBOOT0 = 0

1: nBOOT0 = 1

Bit 26 **nSWBOOT0**: Software BOOT0

0: BOOT0 taken from the option bit nBOOT0

1: BOOT0 taken from PB8/BOOT0 pin

Bit 25 **CCMSRAM_RST**: CCM SRAM erase when system reset

0: CCM SRAM erased when a system reset occurs

1: CCM SRAM is not erased when a system reset occurs

Bit 24 **SRAM_PE**: SRAM1 and CCM SRAM parity check enable

0: SRAM1 and CCM SRAM parity check enable

1: SRAM1 and CCM SRAM parity check disable

Bit 23 **nBOOT1**: Boot configuration

Together with the BOOT0 pin, this bit selects boot mode from the Flash main memory, SRAM1 or the System memory. Refer to [Section 2.6: Boot configuration](#).

Bits 22:20 Reserved, must be kept at reset value.

Bit 19 **WWDG_SW**: Window watchdog selection

0: Hardware window watchdog

1: Software window watchdog

Bit 18 **IWDG_STDBY**: Independent watchdog counter freeze in Standby mode

0: Independent watchdog counter is frozen in Standby mode

1: Independent watchdog counter is running in Standby mode

Bit 17 **IWDG_STOP**: Independent watchdog counter freeze in Stop mode

0: Independent watchdog counter is frozen in Stop mode

1: Independent watchdog counter is running in Stop mode

Bit 16 **IWDG_SW**: Independent watchdog selection

0: Hardware independent watchdog

1: Software independent watchdog

Bit 15 Reserved, must be kept at reset value.

Bit 14 **nRST_SHDW**

0: Reset generated when entering the Shutdown mode

1: No reset generated when entering the Shutdown mode

Bit 13 **nRST_STDBY**

0: Reset generated when entering the Standby mode

1: No reset generated when entering the Standby mode

Bit 12 **nRST_STOP**

0: Reset generated when entering the Stop mode

1: No reset generated when entering the Stop mode

Bit 11 Reserved, must be kept at reset value.

Bits 10:8 **BORLEV**: BOR reset Level

These bits contain the VDD supply level threshold that activates/releases the reset.

000: BOR Level 0. Reset level threshold is around 1.7 V

001: BOR Level 1. Reset level threshold is around 2.0 V

010: BOR Level 2. Reset level threshold is around 2.2 V

011: BOR Level 3. Reset level threshold is around 2.5 V

100: BOR Level 4. Reset level threshold is around 2.8 V

Bits 7:0 **RDP**: Read protection level

0xAA: Level 0, read protection not active

0xCC: Level 2, chip read protection active

Others: Level 1, memories read protection active

PCROP1 Start address option bytes

Flash memory address: 0x1FFF 7808

Reset value: 0xFFFF FFFF (ST production value)

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---------------------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PCROP1_STRT[13:0] | | | | | | | | | | | | | | | |
| r r r r r r r r r r r r r r r r | | | | | | | | | | | | | | | |

Bits 31:14 Reserved, must be kept at reset value.

Bits 13:0 **PCROP1_STRT[13:0]**: PCROP area start offset

PCROP1_STRT contains the first double-word of the PCROP area for bank1.

PCROP1 End address option bytes

Flash memory address: 0x1FFF 7810

Reset value: 0x00 FF 0000 (ST production value)

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------|------|------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| PCROP_P_RDP | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| r | | | | | | | | | | | | | | | | |
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | PCROP1-END[13:0] | | | | | | | | | | | | | | |
| | | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

Bit 31 **PCROP_RDP**: PC ROP area preserved when RDP level decreased

This bit is set only. It is reset after a full mass erase due to a change of RDP from Level 1 to Level 0.

0: PC ROP area is not erased when the RDP level is decreased from Level 1 to Level 0.

1: PC ROP area is erased when the RDP level is decreased from Level 1 to Level 0 (full mass erase).

Bits 30:14 Reserved, must be kept at reset value.

Bits 13:0 **PCROP1-END[13:0]**: Bank 1 PCROP area end offset

PCROP1-END contains the last double-word of the PCROP area.

WRP1 Area A address option bytes

Flash memory address: 0x1FFF 7818

Reset value: 0xFF00 FFFF (ST production value)

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|------|------|------|------|------|------|------|------|------|------|------|-----------------|----|----|----|----|
| Res. | WRP1A-END[5:0] | | | | |
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | r | r | r | r |
| Res. | WRP1A-STRT[5:0] | | | | |
| | | | | | | | | | | | | r | r | r | r | r |

Bits 31:22 Reserved, must be kept at reset value.

Bits 21:16 **WRP1A-END[5:0]**: WRP first area "A" end offset

WRP1A-END contains the last page of WRP first area.

Bits 15:6 Reserved, must be kept at reset value.

Bits 5:0 **WRP1A-STRT[5:0]**: WRP first area "A" start offset

WRP1A-STRT contains the first page of WRP first area.

WRP2 Area A address option bytes

Flash memory address: 0x1FFF7820

Reset value: 0xFF00 FFFF (ST production value)

| | | | | | | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-----------------|---|---|---|---|---|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | | | |
| Res. | WRP2A_END[5:0] | | | | | |
| | | | | | | | | | | | | | | | r | r | r | r | r | r |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | |
| Res. | WRP2A_STRT[5:0] | | | | | |
| | | | | | | | | | | | | | | | r | r | r | r | r | r |

Bits 31:22 Reserved, must be kept at reset value.

Bits 21:16 **WRP2A_END[5:0]**: WRP second area "B" end offset

WRP1B_END contains the last page of the WRP second area.

Bits 15:6 Reserved, must be kept at reset value.

Bits 5:0 **WRP2A_STRT[5:0]**: WRP second area start offset

WRP1B_STRT contains the last page of the WRP second area.

Securable memory area option bytes

Flash memory address: 0x1FFF7828

Reset value: 0xFF00FF00 (ST production value)

| | | | | | | | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|----------------|---|---|---|---|---|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | | | | |
| Res. | BOOT_LOCK | | | | | |
| | | | | | | | | | | | | | | | | r | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | |
| Res. | SEC_SIZE1[6:0] | | | | | |
| | | | | | | | | | | | | | | | | r | r | r | r | r | r |

Bits 31:17 Reserved, must be kept at reset value.

Bit 16 **BOOT_LOCK**: used to force boot from user Flash area

0: Boot based on the pad/option bit configuration

1: Boot forced from Main Flash memory

Bits 15:7 Reserved, must be kept at reset value.

Bits 6:0 **SEC_SIZE1[6:0]**: Securable memory area size

Contains the number of Securable Flash memory pages

5.4.2 Option bytes programming

After reset, the options related bits in the [Flash control register \(FLASH_CR\)](#) are write-protected. To run any operation on the option bytes page, the option lock bit OPTLOCK in the [Flash control register \(FLASH_CR\)](#) must be cleared. The following sequence is used to unlock this register:

1. Unlock the FLASH_CR with the LOCK clearing sequence (refer to [Unlocking the Flash memory](#)).
2. Write OPTKEY1 = 0x08192A3B in the [Flash option key register \(FLASH_OPTKEYR\)](#).
3. Write OPTKEY2 = 0x4C5D6E7F in the FLASH_OPTKEYR register.

The user options can be protected against unwanted erase/program operations by setting the OPTLOCK bit by software.

Note: If LOCK is set by software, OPTLOCK is automatically set too.

Modifying user options

The option bytes are programmed differently from a main memory user address. To modify the user options value, follow the procedure below:

1. Check that no Flash memory operation is on going by checking the BSY bit in the [Flash status register \(FLASH_SR\)](#).
2. Clear OPTLOCK option lock bit with the clearing sequence described above.
3. Write the desired options value in the options registers: [Flash option register \(FLASH_OPTR\)](#), [Flash PCROP1 Start address register \(FLASH_PCR0P1SR\)](#), [Flash PCROP1 End address register \(FLASH_PCR0P1ER\)](#), [Flash WRP area A address register \(FLASH_WRP1AR\)](#), [Flash WRP area B address register \(FLASH_WRP1BR\)](#).
4. Set the Options Start bit OPTSTRT in the [Flash control register \(FLASH_CR\)](#).
5. Wait for the BSY bit to be cleared.

Note: Any modification of the value of one option is automatically performed by erasing both user option bytes pages first and then programming all the option bytes with the values contained in the flash option registers.

Option byte loading

After the BSY bit is cleared, all new options are updated into the flash but they are not applied to the system. They have effect on the system when they are loaded. Option bytes loading (OBL) is performed in two cases:

- when OBL_LAUNCH bit is set in the [Flash control register \(FLASH_CR\)](#).
- after a power reset (BOR reset or exit from Standby/Shutdown modes).

Option byte loader performs a read of the options block and stores the data into internal option registers. These internal registers configure the system and cannot be read with software. Setting OBL_LAUNCH generates a reset so the option byte loading is performed under system reset.

Each option bit has also its complement in the same double word. During option loading, a verification of the option bit and its complement allows to check the loading has correctly taken place.

During option byte loading, the options are read by double word with ECC. If the word and its complement are matching, the option word/byte is copied into the option register.

If the comparison between the word and its complement fails, a status bit OPTVERR is set. Mismatch values are forced into the option registers:

- For USR OPT option, the value of mismatch is all options at ‘1’, except for BOR_lev which is “000” (lowest threshold)
- For WRP option, the value of mismatch is the default value “No protection”
- For RDP option, the value of mismatch is the default value “Level 1”
- For PCROP, the value of mismatch is “all memory protected”

On system reset rising, internal option registers are copied into option registers which can be read and written by software (FLASH_OPTR, FLASH_PCROP1SR, FLASH_PCROP1ER, FLASH_WRP1AR, FLASH_WRP1BR). These registers are also used to modify options. If these registers are not modified by user, they reflects the options states of the system. See [Section : Modifying user options](#) for more details.

5.5 FLASH memory protection

The Flash main memory can be protected against external accesses with the Read protection (RDP). The pages of the Flash memory can also be protected against unwanted write due to loss of program counter contexts. The write-protection (WRP) granularity is one page (2 KByte). A part of the flash memory can also be protected against read and write from third parties (PCROP). The PCROP granularity is double word (64-bit).

5.5.1 Read protection (RDP)

The read protection is activated by setting the RDP option byte and then, by applying a system reset to reload the new RDP option byte. The read protection protects the Flash main memory, the option bytes, the backup registers (RTC_BKPxR in the RTC) and the CCM SRAM.

Note: *If the read protection is set while the debugger is still connected through JTAG/SWD, apply a POR (power-on reset) instead of a system reset.*

There are three levels of read protection from no protection (level 0) to maximum protection or no debug (level 2).

The Flash memory is protected when the RDP option byte and its complement contain the pair of values shown in [Table 32](#).

Table 32. Flash memory read protection status

| RDP byte value | RDP complement value | Read protection level |
|-------------------------------|--|----------------------------|
| 0xAA | 0x55 | Level 0 (production value) |
| Any value except 0xAA or 0xCC | Any value (not necessarily complementary) except 0x55 and 0x33 | Level 1 |
| 0xCC | 0x33 | Level 2 |

The System memory area is read accessible whatever the protection level. It is never accessible for program/erase operation.

Level 0: no protection

Read, program and erase operations into the Flash main memory area are possible. The option bytes, the CCM SRAM and the backup registers are also accessible by all operations.

Level 1: Read protection

This is the default protection level when RDP option byte is erased. It is defined as well when RDP value is at any value different from 0xAA and 0xCC, or even if the complement is not correct.

- **User mode:** Code executing in user mode (**Boot Flash**) can access Flash main memory, option bytes, CCM SRAM and backup registers with all operations.
- **Debug, boot RAM and boot loader modes:** In debug mode or when code is running from boot RAM or boot loader, the Flash main memory, the backup registers (RTC_BKPxR in the RTC) and the CCM SRAM are totally inaccessible. In these modes, a read or write access to the Flash generates a bus error and a Hard Fault interrupt.

Caution: In case the Level 1 is configured and no PCROP area is defined, it is mandatory to set PCROP_RDP bit to 1 (full mass erase when the RDP level is decreased from Level 1 to Level 0). In case the Level 1 is configured and a PCROP area is defined, if user code needs to be protected by RDP but not by PCROP, it must not be placed in a page containing a PCROP area.

Level 2: No debug

In this level, the protection level 1 is guaranteed. In addition, the Cortex®-M4 debug port, the boot from RAM (boot RAM mode) and the boot from System memory (boot loader mode) are no more available. In user execution mode (boot FLASH mode), all operations are allowed on the Flash Main memory. On the contrary, only read operations can be performed on the option bytes.

Option bytes cannot be programmed nor erased. Thus, the level 2 cannot be removed at all: it is an irreversible operation. When attempting to modify the options bytes, the protection error flag WRPERR is set in the Flash_SR register and an interrupt can be generated.

Note: *The debug feature is also disabled under reset.*

STMicroelectronics is not able to perform analysis on defective parts on which the level 2 protection has been set.

Changing the Read protection level

It is easy to move from level 0 to level 1 by changing the value of the RDP byte to any value (except 0xCC). By programming the 0xCC value in the RDP byte, it is possible to go to level 2 either directly from level 0 or from level 1. Once in level 2, it is no more possible to modify the Read protection level.

When the RDP is reprogrammed to the value 0xAA to move from Level 1 to Level 0, a mass erase of the Flash main memory is performed if PCROP_RDP is set in the [Flash PCR OP1 End address register \(FLASH_PCROP1ER\)](#). The backup registers (RTC_BKPxR in the RTC) and the CCM SRAM are also erased. The user options except PCROP protection are set to their previous values copied from FLASH_OPTR, FLASH_WRPxyR (x=1 and y=A or B). PCROP is disable. The OTP area is not affected by mass erase and remains unchanged.

If the bit PCROP_RDP is cleared in the FLASH_PCROP1ER, the full mass erase is replaced by a partial mass erase that is successive page erases, except for the pages protected by PCROP. This is done in order to keep the PCROP code. Only when the Flash memory is erased, options are re-programmed with their previous values. This is also true for FLASH_PCROPxSR and FLASH_PCROPxER registers ($x=1$).

Note: *Full Mass Erase or Partial Mass Erase is performed only when Level 1 is active and Level 0 requested. When the protection level is increased (0->1, 1->2, 0->2) there is no mass erase. To validate the protection level change, the option bytes must be reloaded through the OBL_LAUNCH bit in Flash control register.*

Figure 10. Changing the read protection (RDP) level

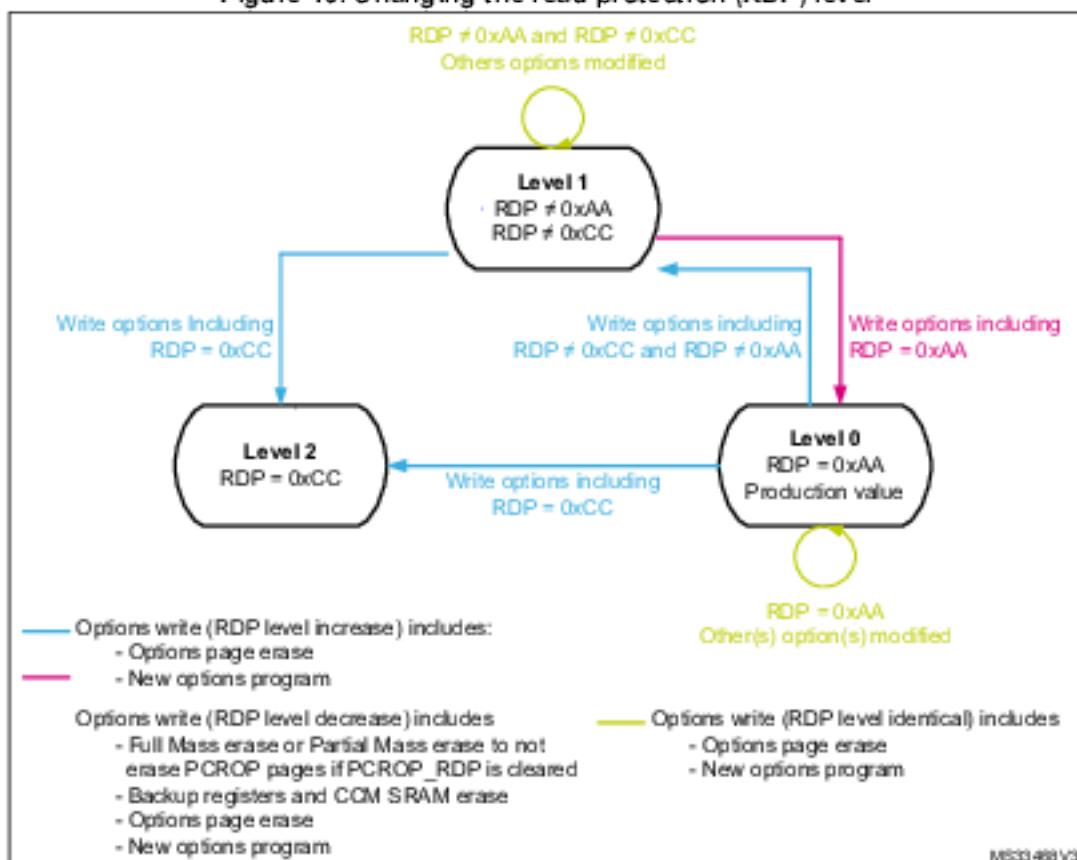


Table 33. Access status versus protection level and execution modes

| Area | Protection level | User execution (BootFromFlash) | | | Debug / BootFromRam / BootFromLoader ⁽¹⁾ | | |
|------------------------------|------------------|--------------------------------|-------|-------|---|-------|-------------------|
| | | Read | Write | Erase | Read | Write | Erase |
| Flash main memory | 1 | Yes | Yes | Yes | No | No | No ⁽³⁾ |
| | 2 | Yes | Yes | Yes | N/A | N/A | N/A |
| System memory ⁽²⁾ | 1 | Yes | No | No | Yes | No | No |
| | 2 | Yes | No | No | N/A | N/A | N/A |

Table 33. Access status versus protection level and execution modes (continued)

| Area | Protection level | User execution (BootFromFlash) | | | Debug / BootFromRam/ BootFromLoader ⁽¹⁾ | | |
|------------------|------------------|--------------------------------|--------------------|-------|---|--------------------|-------------------|
| | | Read | Write | Erase | Read | Write | Erase |
| Option bytes | 1 | Yes | Yes ⁽³⁾ | Yes | Yes | Yes ⁽³⁾ | Yes |
| | 2 | Yes | No | No | N/A | N/A | N/A |
| OTP | 1 | Yes | Yes ⁽⁴⁾ | N/A | No | No | N/A |
| | 2 | Yes | Yes ⁽⁴⁾ | N/A | N/A | N/A | N/A |
| Backup registers | 1 | Yes | Yes | N/A | No | No | No ⁽⁵⁾ |
| | 2 | Yes | Yes | N/A | N/A | N/A | N/A |
| CCM SRAM | 1 | Yes | Yes | N/A | No | No | No ⁽⁶⁾ |
| | 2 | Yes | Yes | N/A | N/A | N/A | N/A |

1. When the protection level 2 is active, the Debug port, the boot from RAM and the boot from system memory are disabled.
2. The system memory is only read-accessible, whatever the protection level (0, 1 or 2) and execution mode.
3. The Flash main memory is erased when the RDP option byte is programmed with all level protections disabled (0xAA).
4. OTP can only be written once.
5. The backup registers are erased when RDP changes from level 1 to level 0.
6. The CCM SRAM is erased when RDP changes from level 1 to level 0.

5.5.2 Proprietary code readout protection (PCROP)

A part of the flash memory can be protected against read and write from third parties. The protected area is execute-only: it can only be reached by the STM32 CPU, as an instruction code, while all other accesses (DMA, debug and CPU data read, write and erase) are strictly prohibited. The PCROP area has a double word (64-bit) granularity. An additional option bit (PCROP_RDP) allows to select if the PCROP area is erased or not when the RDP protection is changed from Level 1 to Level 0 (refer to [Changing the Read protection level](#)).

Each PCROP area is defined by a start page offset and an end page offset related to the physical Flash address. These offsets are defined in the PCROP address registers [Flash PCROP1 Start address register \(FLASH_PCROP1SR\)](#), [Flash PCROP1 End address register \(FLASH_PCROP1ER\)](#).

- The PCROP_x ($x = 1$) area is defined from the address: Flash memory base address + [PCROP_x_STRT x 0x8] (included) to the address: Flash memory base address + [(PCROP_x_END+1) x 0x8] (excluded). The minimum PCROP area size is two double-words (128 bits).

For example, to protect by PCROP from the address 0x0806 2F80 (included) to the address 0x0807 0004 (included):

- If boot in flash is selected, FLASH_PCROP1SR and FLASH_PCROP1ER registers must be programmed with:
 - PCROP1_STRT = 0xC5F0.
 - PCROP1_END = 0xE000.

Any read access performed through the D-bus to a PCROP protected area triggers RDERR flag error.

Any PCROP protected address is also write protected and any write access to one of these addresses triggers WRPERR.

Any PCROP area is also erase protected. Consequently, any erase to a page in this zone is impossible (including the page containing the start address and the end address of this zone). Moreover, a software mass erase cannot be performed if one zone is PCROP protected.

For previous example, due to erase by page, all pages from page 0xC5 to 0xE0 are protected in case of page erase. (All addresses from 0x0806 2800 to 0x0807 07FF can't be erased).

Deactivation of PCROP can only occurs when the RDP is changing from level 1 to level 0. If the user options modification tries to clear PCROP or to decrease the PCROP area, the options programming is launched but PCROP area stays unchanged. On the contrary, it is possible to increase the PCROP area.

When option bit PCROP_RDP is cleared, when the RDP is changing from level 1 to level 0, Full Mass Erase is replaced by Partial Mass Erase in order to keep the PCROP area (refer to [Changing the Read protection level](#)). In this case, PCROP1_STRT and PCROP1_END are also not erased.

Note: *It is recommended to align PCROP area with page granularity when using PCROP_RDP, or to leave free the rest of the page where PCROP zone starts or ends.*

Table 34. PCROP protection⁽¹⁾

| PCROPx registers values (x = 1) | PCROP protection area |
|---|--|
| PCROPx_offset strt > PCROPx_offset_end | No PCROP area. |
| PCROPx_offset strt < PCROPx_offset_end | The area between PCROPx_offset strt and PCROPx_offset_end is protected. It is possible to write: – PCROPx_offset strt with a lower value – PCROPx_offset_end with a higher value. |

1. The minimum PCROP area size is 2xdouble words: PCROPx_offset strt and PCROPx_offset_end.

5.5.3 Write protection (WRP)

The user area in Flash memory can be protected against unwanted write operations. It allows either to specify:

- Two write-protected (WRP) areas can be defined, with page (2 KByte) granularity.

Each area is defined by a start page offset and an end page offset related to the physical Flash base address. These offsets are defined in the WRP address registers: [Flash WRP area A address register \(FLASH_WRP1AR\)](#), [Flash WRP area B address register \(FLASH_WRP1BR\)](#).

The WRP "y" area (x=1 and y=A,B) is defined from the address: *Flash memory Base address + [WRPx_y_STRT x 0x800] (included)* to the address: *Flash memory Base address + [(WRPx_y_END +1) x 0x800] (excluded)*.

For example, to protect by WRP from the address 0x08012800 (included) to the address 0x080187FF (included):

- if boot in flash is selected, FLASH_WRP1AR register must be programmed with:
 - WRP1A_STRT = 0x25.
 - WRP1A_END = 0x30.

WRP1B_STRT and WRP1B_END in FLASH_WRP1BR can be used instead (area "B" in Flash memory).

When WRP is active, it cannot be erased or programmed. Consequently, a software mass erase cannot be performed if one area is write-protected.

If an erase/program operation to a write-protected part of the Flash memory is attempted, the write protection error flag (WRPERR) is set in the FLASH_SR register. This flag is also set for any write access to:

- OTP area
- part of the Flash memory that can never be written like the ICP
- PCROP area.

Note: When the memory read protection level is selected (RDP level = 1), it is not possible to program or erase Flash memory if the CPU debug features are connected (JTAG or single wire) or boot code is being executed from RAM or System flash, even if WRP is not activated.

Note: To validate the WRP options, the option bytes must be reloaded through the OBL_LAUNCH bit in Flash control register.

Table 35. WRP protection

| WRP registers values (x=1 y= A/B) | WRP protection area |
|---|---|
| WRPx _y _STRT = WRPx _y _END | Page WRPx _y is protected. |
| WRPx _y _STRT > WRPx _y _END | No WRP area. |
| WRPx _y _STRT < WRPx _y _END | The pages from WRPx _y _STRT to WRPx _y _END are protected. |

5.5.4 Securable memory area

The Securable memory area defines an area of code which can be executed only once at boot, and never again unless a new reset occurs.

The main purpose of the Securable memory area is to protect a specific part of Flash memory against undesired access. This allows implementing software security services such as secure key storage or safe boot. Securable memory area is located in the Main Flash memory. It is dedicated to executing trusted code. When not secured, the Securable memory behaves like the remainder of Main Flash memory. When secured (the SEC PROT1 bit of the FLASH_CR register set), any attempt to program or erase in a secure memory area generates a write protection error (WRPERR flag is set) and any attempt to read from it generates a read error (RDERR flag is set).

The size of the securable memory area is defined by the SEC_SIZE1[6:0] bitfield of the FLASH_SEC register. It can be modified only in RDP Level 0. Its content is erased upon changing from RDP Level 1 to Level 0, even if it overlaps with PCROP pages.

The securable memory area is defined from the address: Bank base address (included) to the address: Bank base address + (0x800 * SEC_SIZE1) (excluded).

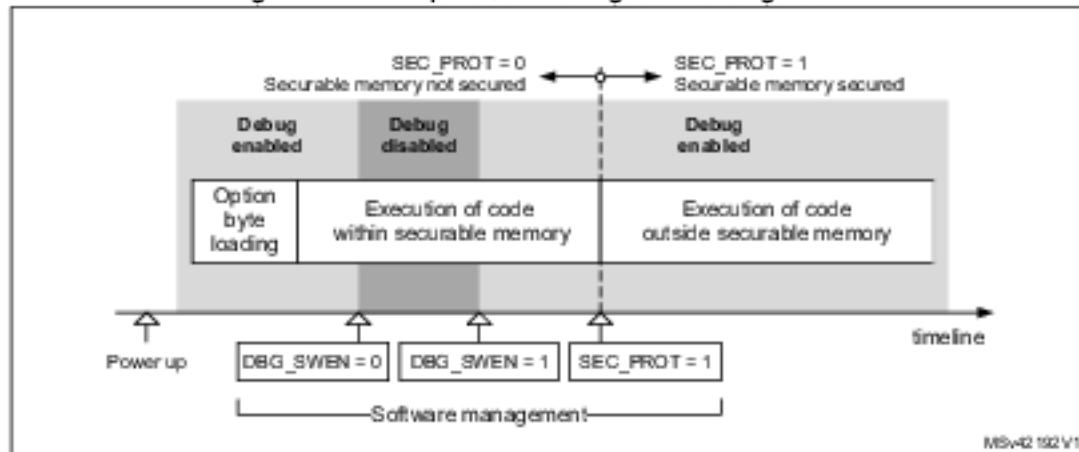
5.5.5 Disabling core debug access

For executing sensitive code or manipulating sensitive data in Securable memory area, the debug access to the core can temporarily be disabled.

In RDP level 2, the debugger is disabled by hardware, but in other RDP levels, the debugger can be disabled by software using the bit DBG_SWEN in the FLASH_ACR register.

Figure 11 gives an example of managing DBG_SWEN and SEC_PROT bits.

Figure 11. Example of disabling core debug access



5.5.6 Forcing boot from Flash memory

To increase the security and establish a chain of trust, the BOOT_LOCK option bit of the FLASH_SEC1R/FLASH_SEC2R register allows forcing the system to boot from the Main Flash memory regardless the other boot options. It is always possible to set the BOOT_LOCK bit. However, it is possible to reset it only when:

- RDP is set to Level 0, or
- RDP is set to Level 1, while Level 0 is requested and a full mass-erase is performed.

5.6 FLASH interrupts

Table 36. Flash interrupt request

| Interrupt event | Event flag | Event flag/interrupt clearing method | Interrupt enable control bit |
|------------------|----------------------|--------------------------------------|------------------------------|
| End of operation | EOP ⁽¹⁾ | Write EOP=1 | EOPIE |
| Operation error | OPERR ⁽²⁾ | Write OPERR=1 | ERRIE |
| Read error | RDERR | Write RDERR=1 | RDERRIE |
| ECC correction | ECCC | Write ECCC=1 | ECCCIE |

1. EOP is set only if EOPIE is set.

2. OPERR is set only if ERRIE is set.

5.7 FLASH registers

5.7.1 Flash access control register (FLASH_ACR)

Address offset: 0x00

Reset value: 0x0004 0601

Access: no wait state, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|----------|--------|-------|-------|------|------|--------|------|------|------|------|------|------|------|--------------|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | SLEEP_PD | RUN_PD | DCRST | ICRST | OCEN | ICEN | PRFTEN | Res. | LATENCY[3:0] |
| rw | rw | rw | rw | rw | rw | rw | rw | | | | | rw | rw | rw | rw |

Bits 31:19 Reserved, must be kept at reset value.

Bit 18 **DBG_SWEN**: Debug software enable

SW may use this bit to enable/disable the debugger.

0: Debugger disabled

1: Debugger enabled

Bits 17:15 Reserved, must be kept at reset value.

Bit 14 **SLEEP_PD**: Flash Power-down mode during Sleep or Low-power sleep mode

This bit determines whether the flash memory is in Power-down mode or Idle mode when the device is in Sleep or Low-power sleep mode.

0: Flash in Idle mode during Sleep and Low-power sleep modes

1: Flash in Power-down mode during Sleep and Low-power sleep modes

Caution: The flash must not be put in power-down while a program or an erase operation is on-going.

Bit 13 **RUN_PD**: Flash Power-down mode during Run or Low-power run mode

This bit is write-protected with FLASH_PDKEYR.

This bit determines whether the flash memory is in Power-down mode or Idle mode when the device is in Run or Low-power run mode. The flash memory can be put in power-down mode only when the code is executed from RAM. The Flash must not be accessed when RUN_PD is set.

0: Flash in Idle mode

1: Flash in Power-down mode

Caution: The flash must not be put in power-down while a program or an erase operation is on-going.

Bit 12 **DCRST**: Data cache reset

0: Data cache is not reset

1: Data cache is reset

This bit can be written only when the data cache is disabled.

- Bit 11 **ICRST**: Instruction cache reset
 0: Instruction cache is not reset
 1: Instruction cache is reset
 This bit can be written only when the instruction cache is disabled.
- Bit 10 **DCEN**: Data cache enable
 0: Data cache is disabled
 1: Data cache is enabled
- Bit 9 **ICEN**: Instruction cache enable
 0: Instruction cache is disabled
 1: Instruction cache is enabled
- Bit 8 **PRFTEN**: Prefetch enable
 0: Prefetch disabled
 1: Prefetch enabled
- Bits 7:4 Reserved, must be kept at reset value.
- Bits 3:0 **LATENCY[3:0]**: Latency
 These bits represent the ratio of the SYSCLK (system clock) period to the Flash access time.
 0000: Zero wait state
 0001: One wait state
 0010: Two wait states
 0011: Three wait states
 0100: Four wait states
 ...1111: Fifteen wait states

5.7.2 Flash Power-down key register (FLASH_PDKEYR)

Address offset: 0x04

Reset value: 0x0000 0000

Access: no wait state, word access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| PDKEYR[31:16] | | | | | | | | | | | | | | | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PDKEYR[15:0] | | | | | | | | | | | | | | | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

Bits 31:0 **PDKEYR[31:0]**: Power-down in Run mode Flash key

The following values must be written consecutively to unlock the RUN_PD bit in FLASH_ACR:

PDKEY1: 0x04 15 2637

PDKEY2: 0xFAFB FCFD

5.7.3 Flash key register (FLASH_KEYR)

Address offset: 0x08

Reset value: 0x0000 0000

Access: no wait state, word access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| KEYR[31:16] | | | | | | | | | | | | | | | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| KEYR[15:0] | | | | | | | | | | | | | | | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

Bits 31:0 KEYR[31:0]: Flash key

The following values must be written consecutively to unlock the FLASH_CTR register allowing flash programming/erasing operations:

KEY1: 0x4567 0123

KEY2: 0xCDEF 89AB

5.7.4 Flash option key register (FLASH_OPTKEYR)

Address offset: 0x0C

Reset value: 0x0000 0000

Access: no wait state, word access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| OPTKEYR[31:16] | | | | | | | | | | | | | | | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OPTKEYR[15:0] | | | | | | | | | | | | | | | |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

Bits 31:0 OPTKEYR[31:0]: Option byte key

The following values must be written consecutively to unlock the FLASH_OPTR register allowing option byte programming/erasing operations:

KEY1: 0x0819 2A3B

KEY2: 0x4C5D 6E7F

5.7.5 Flash status register (FLASH_SR)

Address offset: 0x10

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access

| | | | | | | | | | | | | | | | | |
|-------------|-----------|------|------|------|------|-------------|-------------|------------|------------|------------|------------|-------------|------|------------|-------|-----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | |
| Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | BSY |
| | | | | | | | | | | | | | | | | r |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| OPTV ERR | RD ERR | Res. | Res. | Res. | Res. | FAST ERR | MISS ERR | PGS ERR | SIZ ERR | PGA ERR | WRP ERR | PROG ERR | Res. | OIP ERR | EOP | |
| re_w1 | rc_w1 | | | | | re_w1 | rc_w1 | re_w1 | rc_w1 | re_w1 | rc_w1 | re_w1 | | re_w1 | re_w1 | |

Bits 31:17 Reserved, must be kept at reset value.

Bit 16 **BSY**: Busy

This indicates that a Flash operation is in progress. This is set on the beginning of a Flash operation and reset when the operation finishes or when an error occurs.

Bit 15 **OPTVERR**: Option validity error

Set by hardware when the options read may not be the one configured by the user. If option haven't been properly loaded, OPTVERR is set again after each system reset.

Cleared by writing 1.

Bit 14 **RDERR**: PCROP read error

Set by hardware when an address to be read through the D-bus belongs to a read protected area of the flash (PCROP protection). An interrupt is generated if RDERRIE is set in FLASH_C.R.

Cleared by writing 1.

Bits 13:10 Reserved, must be kept at reset value.

Bit 9 **FASTERR**: Fast programming error

Set by hardware when a fast programming sequence (activated by FSTPG) is interrupted due to an error (alignment, size, write protection or data miss). The corresponding status bit (PGAERR, SIZERR, WRPERR or MISSERR) is set at the same time.

Cleared by writing 1.

Bit 8 **MISSERR**: Fast programming data miss error

In Fast programming mode, 32 double words must be sent to flash successfully, and the new data must be sent to the flash logic control before the current data is fully programmed. MISSERR is set by hardware when the new data is not present in time.

Cleared by writing 1.

Bit 7 **PGSERR**: Programming sequence error

Set by hardware when a write access to the Flash memory is performed by the code while PG or FSTPG have not been set previously. Set also by hardware when PROGERR, SIZERR, PGAERR, WRPERR, MISSERR or FASTERR is set due to a previous programming error.

Cleared by writing 1.

Bit 6 **SIZERR**: Size error

Set by hardware when the size of the access is a byte or half-word during a program or a fast program sequence. Only double word programming is allowed (consequently: word access).

Cleared by writing 1.

Bit 5 **PGAERR**: Programming alignment error

Set by hardware when the data to program cannot be contained in the same 64-bit Flash memory row in case of standard programming, or if there is a change of page during fast programming.

Cleared by writing 1.

Bit 4 **WRPERR**: Write protection error

Set by hardware when an address to be erased/programmed belongs to a write-protected part (by WRP, PCROP or RDP level 1) of the Flash memory.

Cleared by writing 1.

Bit 3 **PROGERR**: Programming error

Set by hardware when a double-word address to be programmed contains a value different from '0xFFFF FFFF FFFF FFFF' before programming, except if the data to write is '0x0000 0000 0000 0000'.

Cleared by writing 1.

Bit 2 Reserved, must be kept at reset value.

Bit 1 **OPERR**: Operation error

Set by hardware when a Flash memory operation (program / erase) completes unsuccessfully.

This bit is set only if error interrupts are enabled (ERRIE = 1).

Cleared by writing '1'.

Bit 0 **EOP**: End of operation

Set by hardware when one or more Flash memory operation (programming / erase) has been completed successfully.

This bit is set only if the end of operation interrupts are enabled (EOPIE = 1).

Cleared by writing 1.

5.7.6 Flash control register (FLASH_CR)

Address offset: 0x14

Reset value: 0xC000 0000

Access: no wait state when no Flash memory operation is on going, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|----------|------|-----------|------------|-------|--------|----------|------|------|------|------|------|-------|----------|------|
| LOCK | OPT LOCK | Res. | SEC PROT1 | OBL LAUNCH | RDERR | ERR IE | EOP IE | Res. | Res. | Res. | Res. | Res. | FSTPG | OPT STRT | STRT |
| 0 | 0 | | 0 | 0_w1 | RW | RW | RW | | | | | | RW | 0 | 0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | Res. | Res. | Res. | Res. | | PNB[6:0] | | | | | | MER1 | PER | PG |
| | | | | | | | RW | RW | RW | RW | RW | RW | RW | RW | RW |

Bit 31 LOCK: FLASH_CR Lock

This bit is set only. When set, the FLASH_CR register is locked. It is cleared by hardware after detecting the unlock sequence.

In case of an unsuccessful unlock operation, this bit remains set until the next system reset.

Bit 30 OPTLOCK: Options Lock

This bit is set only. When set, all bits concerning user option in FLASH_CR register and so option page are locked. This bit is cleared by hardware after detecting the unlock sequence. The LOCK bit must be cleared before doing the unlock sequence for OPTLOCK bit.

In case of an unsuccessful unlock operation, this bit remains set until the next reset.

Bit 29 Reserved, must be kept at reset value.**Bit 28 SEC_PROT1: Securable memory area protection bit.**

This bit is set to lock the access to the Securable memory area. It is set by software when exiting the Securable memory area, and can only be written once.

Bit 27 OBL_LAUNCH: Force the option byte loading

When set to 1, this bit forces the option byte reloading. This bit is cleared only when the option byte loading is complete. It cannot be written if OPTLOCK is set.

0: Option byte loading complete

1: Option byte loading requested

Bit 26 RDERRIE: PCROP read error interrupt enable

This bit enables the interrupt generation when the RDERR bit in the FLASH_SR is set to 1.

0: PCROP read error interrupt disabled

1: PCROP read error interrupt enabled

Bit 25 ERRIE: Error interrupt enable

This bit enables the interrupt generation when the OPERR bit in the FLASH_SR is set to 1.

0: OPERR error interrupt disabled

1: OPERR error interrupt enabled

Bit 24 EOPIE: End of operation interrupt enable

This bit enables the interrupt generation when the EOP bit in the FLASH_SR is set to 1.

0: EOP interrupt disabled

1: EOP interrupt enabled

Bits 23:19 Reserved, must be kept at reset value.**Bit 18 FSTPG: Fast programming**

0: Fast programming disabled

1: Fast programming enabled

Bit 17 OPTSTRT: Options modification start

This bit triggers an options operation when set.

This bit is set only by software, and is cleared when the BSY bit is cleared in FLASH_SR.

Bit 16 START: Start

This bit triggers an erase operation when set. If MER1, MER2 and PER bits are reset and the STRT bit is set, an unpredictable behavior may occur without generating any error flag. This condition should be forbidden.

This bit is set only by software, and is cleared when the BSY bit is cleared in FLASH_SR.

Bits 15:11 Reserved, must be kept at reset value.

Bit 10 Reserved, must be kept at reset v.

Bits 9:3 PNB[6:0]: Page number selection

These bits select the page to erase:

00000000: page 0

00000001: page 1

...

11111111: page 255

Bit 2 MER1: Mass erase

This bit triggers the mass erase (all user pages) when set.

Bit 1 PER: Page erase

0: page erase disabled

1: page erase enabled

Bit 0 PG: Programming

0: Flash programming disabled

1: Flash programming enabled

5.7.7 Flash ECC register (FLASH_ECCR)

Address offset: 0x18

Reset value: 0x0000 0000

Access: no wait state when no Flash memory operation is ongoing, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|----------------|-------|------|------|------|------|------|---------|------|----------|------|------|------|-----------------|----|----|
| ECCD | ECCC | Res. | Res. | Res. | Res. | Res. | ECCC IE | Res. | SYSF_ECC | Res. | Res. | Res. | ADDR_ECC[18:16] | | |
| re_w1 | re_w1 | | | | | | rw | | r | | | | r | r | r |
| ADDR_ECC[15:0] | | | | | | | | | | | | | | | |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

Bit 31 ECCD: ECC detection

Set by hardware when two ECC errors have been detected (only if ECCC/ECCD are previously cleared). When this bit is set, a NMI is generated.

Cleared by writing 1.

Bit 30 ECCC: ECC correction

Set by hardware when one ECC error has been detected and corrected (only if ECCC/ECCC2/ECCD/ECCD2 are previously cleared). An interrupt is generated if ECCCIE is set.

Cleared by writing 1.

Bits 29:25 Reserved, must be kept at reset value.

Bit 24 ECCCIE: ECC correction interrupt enable

0: ECCC interrupt disabled

1: ECCC interrupt enabled.

This bit enables the interrupt generation when the ECCC bit in the FLASH_ECCR register is set.

Bit 23 Reserved, must be kept at reset value.

Bit 22 SYSF_ECC: System Flash ECC fail

This bit indicates that the ECC error correction or double ECC error detection is located in the System Flash.

Bits 21:19 Reserved, must be kept at reset v.

Bits 18:0 ADDR_ECC: ECC fail address

This bit indicates which address in the Flash memory is concerned by the ECC error correction or by the double ECC error detection.

5.7.8 Flash option register (FLASH_OPTR)

Address offset: 0x20

Reset value: 0xFXXX XXXX. Register bits are loaded with values from Flash memory at OBL.

Access: no wait state when no Flash memory operation is on going, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|-----------|-----------------|-----------|---------|-------------|--------------|---------|----------|------|------|------|---------|------------|-----------|---------|
| Res. | IRHEN | NRST_MODE [1:0] | | n BOOTD | nSW BOOTD | CCM SRAM_RST | SRAM_PE | nBOOT1 | Res. | Res. | Res. | WWDG_SW | IWDG_STDBY | IWDG_STOP | IWDG_SW |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | nRST_SHDW | nRST_STDBY | nRST_STOP | Res. | BDRLEV[2:0] | | | RDP[7:0] | | | | | | | |
| | rw | rw | rw | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Bit 31 Reserved, must be kept at reset value.

Bit 30 **IRHEN**: Internal reset holder enable bit

0: Internal resets are propagated as simple pulse on NRST pin

1: Internal resets drives NRST pin low until it is seen as low level

Bits 29:28 **NRST_MODE[1:0]**:

00: Reserved

01: Reset Input only: a low level on the NRST pin generates system reset, internal RESET not propagated to the NSRT pin

10: GPIO: standard GPIO pad functionality, only internal RESET possible

11: Bidirectional reset: NRST pin configured in reset input/output mode (legacy mode)

Bit 27 **nBOOT0**: nBOOT0 option bit

0: nBOOT0 = 0

1: nBOOT0 = 1

Bit 26 **nSWBOOT0**: Software BOOT0

0: BOOT0 taken from the option bit nBOOT0

1: BOOT0 taken from PB8/BOOT0 pin

Bit 25 **CCMSRAM_RST**: CCM SRAM Erase when system reset

0: CCM SRAM erased when a system reset occurs

1: CCM SRAM is not erased when a system reset occurs

Bit 24 **SRAM_PE**: SRAM1 and CCM SRAM parity check enable

0: SRAM1 and CCM SRAM parity check enable

1: SRAM1 and CCM SRAM parity check disable

Bit 23 **nBOOT1**: Boot configuration

Together with the BOOT0 pin, this bit selects boot mode from the Flash main memory, SRAM1 or the System memory. Refer to [Section 2.6: Boot configuration](#).

Bits 22:20 Reserved, must be reset value.

Bit 19 **WWDG_SW**: Window watchdog selection

0: Hardware window watchdog

1: Software window watchdog

Bit 18 **IWDG_STDBY**: Independent watchdog counter freeze in Standby mode

0: Independent watchdog counter is frozen in Standby mode

1: Independent watchdog counter is running in Standby mode

Bit 17 **IWDG_STOP**: Independent watchdog counter freeze in Stop mode

0: Independent watchdog counter is frozen in Stop mode

1: Independent watchdog counter is running in Stop mode

Bit 16 **IWDG_SW**: Independent watchdog selection

0: Hardware independent watchdog

1: Software independent watchdog

Bit 15 Reserved, must be reset value.

Bit 14 **nRST_SHDW**

0: Reset generated when entering the Shutdown mode

1: No reset generated when entering the Shutdown mode

Bit 13 nRST_STDBY

0: Reset generated when entering the Standby mode
1: No reset generated when entering the Standby mode

Bit 12 nRST_STOP

0: Reset generated when entering the Stop mode
1: No reset generated when entering the Stop mode

Bit 11 Reserved, must be reset value.

Bits 10:8 BOR_lev: BOR reset Level

These bits contain the VDD supply level threshold that activates/ deactivates the reset.

000: BOR Level 0. Reset level threshold is around 1.7 V
001: BOR Level 1. Reset level threshold is around 2.0 V
010: BOR Level 2. Reset level threshold is around 2.2 V
011: BOR Level 3. Reset level threshold is around 2.5 V
100: BOR Level 4. Reset level threshold is around 2.8 V

Bits 7:0 RDP: Read protection level

0xAA: Level 0, read protection not active
0xCC: Level 2, chip read protection active
Others: Level 1, memories read protection active

Note: Take care about PCR OP_RDP configuration in Level 1. Refer to [Section : Level 1: Read protection](#) for more details.

5.7.9 Flash PCROP1 Start address register (FLASH_PCROP1SR)

Address offset: 0x24

Reset value: 0xFFFF XXXX

Register bits are loaded with values from Flash memory at OBL.

Access: no wait state when no Flash memory operation is ongoing, word access.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-----|-------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res | PCROP1_STRT[14:0] | | | | | | | | | | | | | | |
| | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Bits 31:15 Reserved, must be kept at reset value.

Bits 14:0 PCROP1_STRT[14:0]: PCROP area start offset

PCROP1_STRT contains the first double-word of the PCROP area.

5.7.10 Flash PCROP1 End address register (FLASH_PCROP1ER)

Address offset: 0x28

Reset value: 0x0000 XXXX

Register bits are loaded with values from Flash memory at OBL.

Access: no wait state when no Flash memory operation is on going, word, half-word access.
PCROP_RDP bit can be accessed with byte access.

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| PCROP_RDP | Res. |
| | | | | | | | | | | | | | | | | |
| PCROP1_END[14:0] | | | | | | | | | | | | | | | | |
| | rw |

Bit 31 PCROP_RDP: PCROP area preserved when RDP level decreased

This bit is set only. It is reset after a full mass erase due to a change of RDP from Level 1 to Level 0.

0: PCROP area is not erased when the RDP level is decreased from Level 1 to Level 0.

1: PCROP area is erased when the RDP level is decreased from Level 1 to Level 0 (full mass erase).

Bits 30:15 Reserved, must be kept at reset value.

Bits 14:0 PCROP1_END[14:0]: PCROP area end offset

PCROP1_END contains the last double-word of the PCROP area.

5.7.11 Flash WRP area A address register (FLASH_WRP1AR)

Address offset: 0x2C

Reset value: 0x00XX 00XX

Register bits are loaded with values from Flash memory at OBL.

Access: no wait state when no Flash memory operation is on going, word, half-word and byte access

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-----------------|------|------|------|------|------|------|------|------|------|------|----------------|----|----|----|----|----|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | WRP1A_END[5:0] | | | | | |
| | | | | | | | | | | | rw | rw | rw | rw | rw | rw |
| WRP1A_STRT[5:0] | | | | | | | | | | | | | | | | |
| | rw | rw | rw | rw | rw | rw |

Bits 31:22 Reserved, must be kept at reset value.

Bits 21:16 **WRP1A_END[5:0]**: WRP first area "A" end offset
 WRP1A_END contains the last page of WRP first area.

Bits 15:6 Reserved, must be kept at reset value.

Bits 5:0 **WRP1A_STRT[5:0]**: WRP first area "A" start offset
 WRP1A_STRT contains the first page of WRP first area.

5.7.12 Flash WRP area B address register (FLASH_WRP1BR)

Address offset: 0x30

Reset value: 0x00XX 00XX

Register bits are loaded with values from Flash memory at OBL.

Access: no wait state when no Flash memory operation is on going, word, half-word and byte access

| | | | | | | | | | | | | | | | |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res |
| | | | | | | | | | | | | | | | |
| WRP1B_END[5:0] | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| WRP1B_STRT[5:0] | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

Bits 31:22 Reserved, must be kept at reset value.

Bits 21:16 **WRP1B_END[5:0]**: WRP second area "B" end offset
 WRP1B_END contains the last page of the WRP second area.

Bits 15:6 Reserved, must be kept at reset value.

Bits 5:0 **WRP1B_STRT[5:0]**: WRP second area "B" start offset
 WRP1B_STRT contains the first page of the WRP second area.

5.7.13 Flash Securable area register (FLASH_SEC1R)

Address offset: 0x70

Reset value: 0xFFFF FFXX

Access: no wait state when no Flash memory operation is on going, word, half-word and byte access

| | | | | | | | | | | | | | | | |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res | Res |
| | | | | | | | | | | | | | | | |
| BOOT_LOCK | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| SEC_SIZE1[8:0] | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

Bits 31:17 Reserved, must be kept at reset value.

Bit 16 **BOOT_LOCK**: used to force boot from user Flash area

0: Boot based on the pad/option bit configuration

1: Boot forced from Main Flash memory

Bits 15:7 Reserved, must be kept at reset value.

Bits 6:0 **SEC_SIZE1[6:0]**: sets the number of pages used in the Securable area.

Securable area starts at @ 0x0800 0000 and its size is SEC_SIZE1 * page size.

This field can be changed in level0 only.

Any attempt to modify in level1 silently fails, and does not change register value.

5.7.14 FLASH register map

Table 37. Flash interface - register map and reset values

| Offset | Register | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | |
|--------|---------------|-----------|----------|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|----------|------|------|----------------|-------------------|---|---|
| 0x00 | FLASH_ACR | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | 1 | DBG_SWEN | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | |
| | Reset value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LATENCY[3:0] | | |
| 0x04 | FLASH_PKEYR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | PKEYR[31:0] | | |
| | Reset value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 0x08 | FLASH_KEYR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | KEYR[31:0] | | |
| | Reset value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 0x0C | FLASH_OPTKEYR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | OPTKEYR[31:0] | | |
| | Reset value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 0x10 | FLASH_SR | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | | |
| | Reset value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x14 | FLASH_CR | LOC_K | OPTLOCK | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | PINB[6:0] | | | |
| | Reset value | 1 | 1 | EDCC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x18 | FLASH_ECCR | EDCD | EDCC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | ADDR_ECC[18:0] | | | |
| | Reset value | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x20 | FLASH_OPTR | WDG_SW | WDG_STBY | WDG_STOP | WDG_SW | WDG_SW | WDG_SW | WDG_SW | WDG_SW | WDG_SW | WDG_SW | WDG_SW | WDG_SW | WDG_SW | WDG_SW | WDG_SW | WDG_SW | WDG_SW | WDG_SW | WDG_SW | RDP[7:0] | | | | | | |
| | Reset value | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| 0x24 | FLASH_PCRP1SR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | PCR0P1_STRT[14:0] | | |
| | Reset value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x28 | FLASH_PCRP1ER | PCR0P_RDP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | PCR0P1_END[14:0] | | |
| | Reset value | x | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x2C | FLASH_WRP1AR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | WRP1A_STRT[6:0] | | |
| | Reset value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | WRP1A_END[6:0] | | |

Table 37. Flash interface - register map and reset values (continued)

| Offset | Register | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|--------|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----------------|---|---|
| 0x30 | FLASH_WRP1B_R | R/W | | | |
| | Reset value | | | | | | | | | | | X | X | X | X | X | X | X | X | | | | | | | | | X | X | X | X | X | X | X |
| 0x70 | FLASH_SEC1R | R/W | SEC_SIZE1[6:0] | | |
| | Reset value | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | X | X | X | X | X |

Refer to [Section 2.2 on page 81](#) for the register boundary addresses.

6 Power control (PWR)

6.1 Power supplies

The STM32G4 Series devices require a 1.71 V to 3.6 V operating supply voltage (V_{DD}). Analog peripherals are supplied through independent power domain V_{DDA} :

- $V_{DD} = 1.71 \text{ V to } 3.6 \text{ V}$

V_{DD} is the external power supply for the I/Os, the internal regulator and the system analog such as reset, power management and internal clocks. It is provided externally through VDD pins.

- $V_{DDA} = 1.62 \text{ V (ADC/COMP) / } 1.71 \text{ V (DAC 1MSPS / DAC 15MSPS) / } 2 \text{ V (OPAMP) / } 2.4 \text{ V (VREFBUF)}$

V_{DDA} is the external analog power supply for A/D converters, D/A converters, voltage reference buffer, operational amplifiers and comparators. The V_{DDA} voltage level is independent from the V_{DD} voltage. V_{DDA} should be preferably connected to V_{DD} when these peripherals are not used.

During power up and power down, the following power sequence is required:

- When V_{DD} is below 1 V, then V_{DDA} supply must remain below $V_{DD} + 300 \text{ mV}$
- When V_{DD} is above 1 V, all power supplies became independent.

During power down phase, V_{DD} can temporarily become lower than other supplies only if the energy provided to the MCU remains below 1 mJ. This allows external decoupling capacitors to be discharged with different time constants during the power down transient phase.

- $V_{BAT} = 1.55 \text{ V to } 3.6 \text{ V}$

V_{BAT} is the power supply for RTC, external clock 32 kHz oscillator and backup registers (through power switch) when V_{DD} is not present. V_{BAT} is internally bonded to VDD for small packages without dedicated pin.

- V_{REF+}, V_{REF-}

V_{REF+} is the input reference voltage for ADCs and DACs. It is also the output of the internal voltage reference buffer when enabled.

When $V_{DDA} < 2 \text{ V}$, V_{REF+} must be equal to V_{DDA} .

When $V_{DDA} \geq 2 \text{ V}$, V_{REF+} must be between 2 V and V_{DDA} .

V_{REF+} can be grounded when ADC and DAC are not active.

The internal voltage reference buffer supports three output voltages, which are configured with VRS bit in the VREFBUF_CSR register:

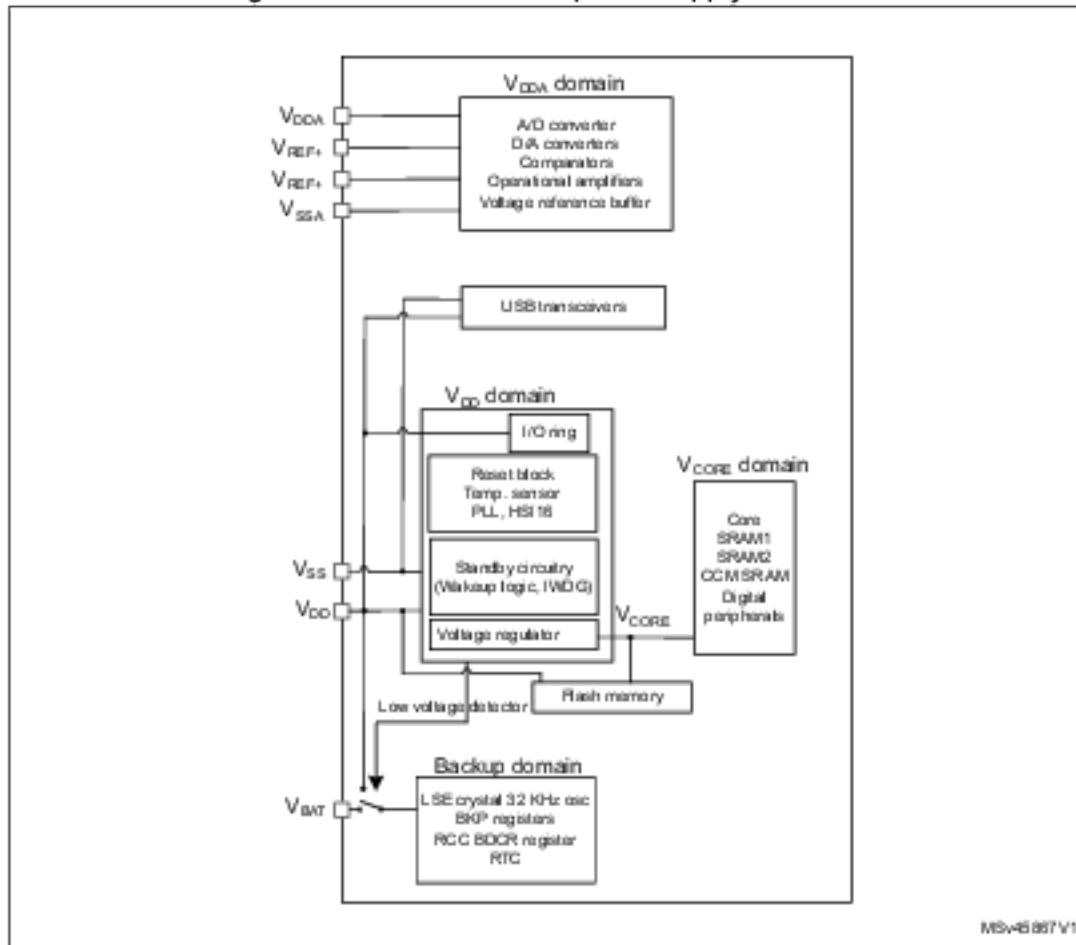
- V_{REF+} around 2.048 V. This requires V_{DDA} equal to or higher than 2.4 V.
- V_{REF+} around 2.5 V. This requires V_{DDA} equal to or higher than 2.8 V.
- V_{REF+} around 2.9 V. This requires V_{DDA} equal to or higher than 3.135 V.

V_{REF+} pin is not available on all packages. When not available on the package, it is bonded to V_{DDA} . When the V_{REF+} is double-bonded with V_{DDA} in a package, the internal voltage reference buffer (VREFBUF) is not available and must be kept disable (refer to related device datasheet for packages pinout description).

V_{REF-} is internally double bonded with V_{SSA} .

An embedded linear voltage regulator is used to supply the internal digital power V_{CORE} . V_{CORE} is the power supply for digital peripherals SRAM1, SRAM2 and CCM SRAM. The Flash is supplied by V_{CORE} and V_{DD} .

Figure 12. STM32G4 Series power supply overview



6.1.1 Independent analog peripherals supply

To improve ADC and DAC conversion accuracy and to extend the supply flexibility, the analog peripherals have an independent power supply which can be separately filtered and shielded from noise on the PCB.

- The analog peripherals voltage supply input is available on a separate V_{DDA} pin.
- An isolated supply ground connection is provided on V_{SSA} pin.

The V_{DDA} supply voltage can be different from V_{DD} . The presence of V_{DDA} must be checked before enabling any of the analog peripherals supplied by V_{DDA} (A/D converter, D/A converter, comparators, operational amplifiers, voltage reference buffer).

The V_{DDA} supply can be monitored by the Peripheral Voltage Monitoring, and compared with thresholds. Refer to [Section 6.2.3: Peripheral Voltage Monitoring \(PVM\)](#) for more details.

When a single supply is used, V_{DDA} can be externally connected to V_{DD} through the external filtering circuit in order to ensure a noise-free V_{DDA} reference voltage.

ADC and DAC reference voltage

To ensure a better accuracy on low-voltage inputs and outputs, the user can connect to V_{REF+} a separate reference voltage lower than V_{DDA} . V_{REF+} is the highest voltage, represented by the full scale value, for an analog input (ADC) or output (DAC) signal.

V_{REF+} can be provided either by an external reference or by an internal buffered voltage reference (VREFBUF).

The internal buffered voltage reference (VREFBUF) is enabled by setting the ENVR bit in the [Section 23.4.1: VREFBUF control and status register \(VREFBUF_CSR\)](#). The internal buffered voltage reference (VREFBUF) is set to 2.048 V, 2.5 V or 2.9 V according the VRS[1:0] bits setting. The internal buffered voltage reference can also provide the voltage to external components through V_{REF+} pin. Refer to the device datasheet and to [Section 23: Voltage reference buffer \(VREFBUF\)](#) for further information.

6.1.2 USB transceivers supply

The USB transceivers are supplied from V_{DD} power supply pin. V_{DD} range for USB usage is from 3.0 V to 3.6 V.

6.1.3 Battery backup domain

To retain the content of the Backup registers and supply the RTC function when V_{DD} is turned off, the V_{BAT} pin can be connected to an optional backup voltage supplied by a battery or by another source.

The V_{BAT} pin powers the RTC unit, the LSE oscillator and the PC13 to PC15 I/Os, allowing the RTC to operate even when the main power supply is turned off. The switch to the V_{BAT} supply is controlled by the power-down reset embedded in the Reset block.

Warning: During $t_{RSTTEMPO}$ (temporization at V_{DD} startup) or after a PDR has been detected, the power switch between V_{BAT} and V_{DD} remains connected to V_{BAT} .
During the startup phase, if V_{DD} is established in less than $t_{RSTTEMPO}$ (refer to the datasheet for the value of $t_{RSTTEMPO}$) and $V_{DD} > V_{BAT} + 0.6$ V, a current may be injected into V_{BAT} through an internal diode connected between V_{DD} and the power switch (V_{BAT}).
If the power supply/battery connected to the V_{BAT} pin cannot support this current injection, it is strongly recommended to connect an external low-drop diode between this power supply and the V_{BAT} pin.

If no external battery is used in the application, it is recommended to connect V_{BAT} externally to V_{DD} with a 100 nF external ceramic decoupling capacitor.

When the backup domain is supplied by V_{DD} (analog switch connected to V_{DD}), the following pins are available:

- PC13, PC14 and PC15, which can be used as GPIO pins
- PC13, PC14 and PC15, which can be configured by RTC or LSE (refer to [Section 3.5.3: Power management](#))

(RTC functional description on page 1537)

- PA0/RTC_TAMP2 and PE6/RTC_TAMP3 when they are configured by the RTC as tamper pins

Note: Due to the fact that the analog switch can transfer only a limited amount of current (3 mA), the use of GPIO PC13 to PC15 in output mode is restricted: the speed has to be limited to 2 MHz with a maximum load of 30 pF and these I/Os must not be used as a current source (e.g. to drive a LED).

When the backup domain is supplied by V_{BAT} (analog switch connected to V_{BAT} because V_{DD} is not present), the following functions are available:

- PC13, PC14 and PC15 can be controlled only by RTC or LSE (refer to [Section 3.5.3: RTC functional description on page 1537](#))
- PA0/RTC_TAMP2 and PE6/RTC_TAMP3 when they are configured by the RTC as tamper pins

Backup domain access

After a system reset, the backup domain (RTC registers and backup registers) is protected against possible unwanted write accesses. To enable access to the backup domain, proceed as follows:

- Enable the power interface clock by setting the PWREN bits in the [Section 7.4.17: APB1 peripheral clock enable register 1 \(RCC_APB1ENR1\)](#)
- Set the DBP bit in the [Power control register 1 \(PWR_CR1\)](#) to enable access to the backup domain
- Select the RTC clock source in the [RTC domain control register \(RCC_BDCR\)](#).
- Enable the RTC clock by setting the RTC EN [15] bit in the [RTC domain control register \(RCC_BDCR\)](#).

VBAT battery charging

When V_{DD} is present, it is possible to charge the external battery on V_{BAT} through an internal resistance.

The V_{BAT} charging is done either through a 5 kOhm resistor or through a 1.5 kOhm resistor depending on the VBR5 bit value in the PWR_CR4 register.

The battery charging is enabled by setting VBE bit in the PWR_CR4 register. It is automatically disabled in VBAT mode.

6.1.4 Voltage regulator

Two embedded linear voltage regulators supply all the digital circuitries, except for the Standby circuitry and the backup domain. The main regulator output voltage (V_{CORE}) can be programmed by software to two different power ranges (Range 1 and Range 2) in order to optimize the consumption depending on the system's maximum operating frequency (refer to [Section 7.2.8: Clock source frequency versus voltage scaling](#) and to [Section 3.3.3: Read access latency](#)).

The voltage regulators are always enabled after a reset. Depending on the application modes, the V_{CORE} supply is provided either by the main regulator (MR) or by the low-power regulator (LPR).

- In Run, Sleep and Stop 0 modes, both regulators are enabled and the main regulator

- (MR) supplies full power to the V_{CORE} domain (core, memories and digital peripherals).
- In low-power run and low-power sleep modes, the main regulator is off and the low-power regulator (LPR) supplies low power to the V_{CORE} domain, preserving the contents of the registers, SRAM1, SRAM2 and CCM SRAM.
- In Stop 1 modes, the main regulator is off and the low-power regulator (LPR) supplies low power to the V_{CORE} domain, preserving the contents of the registers, SRAM1, SRAM2 and CCM SRAM.
- In Standby mode with SRAM2 content preserved (RRS bit is set in the PWR_CR3 register), the main regulator (MR) is off and the low-power regulator (LPR) provides the supply to SRAM2 only. The core, digital peripherals (except Standby circuitry and backup domain) SRAM1 and CCM SRAM are powered off.
- In Standby mode, both regulators are powered off. The contents of the registers, SRAM1, SRAM2 and CCM SRAM is lost except for the Standby circuitry and the backup domain.
- In Shutdown mode, both regulators are powered off. When exiting from Shutdown mode, a power-on reset is generated. Consequently, the contents of the registers, SRAM1, SRAM2 and CCM SRAM is lost, except for the backup domain.

6.1.5 Dynamic voltage scaling management

The dynamic voltage scaling is a power management technique which consists in increasing or decreasing the voltage used for the digital peripherals (V_{CORE}), according to the application performance and power consumption needs.

Dynamic voltage scaling to increase V_{CORE} is known as overvolting. It allows to improve the device performance.

Dynamic voltage scaling to decrease V_{CORE} is known as undervolting. It is performed to save power, particularly in laptop and other mobile devices where the energy comes from a battery and is thus limited.

- Range 1: High-performance range.

In range 1, the main regulator operates in two modes following the R1MODE bit in the PWR_CR5 register:

- Main regulator range 1 normal mode: provides a typical output voltage at 1.2 V. It is used when the system clock frequency is up to 150 MHz. The Flash access time for read access is minimum, write and erase operations are possible.
- Main regulator range 1 boost mode: provides a typical output voltage at 1.28 V. It is used when the system clock frequency is up to 170 MHz. The Flash access time for read access is minimum, write and erase operations are possible. To optimize the power consumption it is recommended to select the range1 boost mode when the system clock frequency is greater than 150 MHz. See [Table 38](#).

Table 38. Range 1 boost mode configuration

| System frequency | $SYSCLK \leq 150$ MHz | $SYSCLK \leq 170$ MHz |
|--------------------------|-----------------------|-----------------------|
| R1MODE bit configuration | 1 | 0 |

- Range 2: Low-power range.

The main regulator provides a typical output voltage at 1.0 V. The system clock frequency can be up to 26 MHz. The Flash access time for a read access is increased as compared to Range 1; write and erase operations are not possible.

Voltage scaling is selected through the VOS bit in the [Section 6.4.1: Power control register 1 \(PWR_CR1\)](#) register.

The sequence to go from Range 1 (Normal/Boost) to Range 2 is:

1. In case of switching from Range 1 boost mode to Range 2, the system clock must be divided by 2 using the AHB prescaler before switching to a lower system frequency for at least 1 μ s and then reconfigure the AHB prescaler.
2. Reduce the system frequency to a value lower than 26 MHz.
3. Adjust number of wait states according new frequency target in Range 2 (LATENCY bits in the FLASH_ACR).
4. Program the VOS bits to "10" in the PWR_CR1 register.

The sequence to go from Range 2 to Range 1 (normal/boost mode) is:

1. Program the VOS bits to "01" in the PWR_CR1 register.
2. Wait until the VOSF flag is cleared in the PWR_SR2 register.
3. Adjust number of wait states according new frequency target in Range 1 (LATENCY bits in the FLASH_ACR).
4. Increase the system frequency by following below procedure:
 - If the system frequency is 26 MHz < SYSCLK ≤ 150 MHz:
 - Select the Range 1 normal mode by setting R1MODE bit in the PWR_CR5 register.
 - Configure and switch to PLL for a new system frequency.
 - If the system frequency is SYSCLK > 150 MHz:
 - The system clock must be divided by 2 using the AHB prescaler before switching to a higher system frequency.
 - Select the Range 1 boost mode by clearing the R1MODE bit in the PWR_CR5 register.
 - Configure and switch to PLL for a new system frequency.
 - Wait for at least 1 μ s and then reconfigure the AHB prescaler to get the needed HCLK clock frequency.

The sequence to switch from Range 1 normal mode to Range 1 boost mode is:

1. The system clock must be divided by 2 using the AHB prescaler before switching to a higher system frequency.
2. Clear the R1MODE bit in the PWR_CR5 register.
3. Adjust the number of wait states according to the new frequency target in range 1 boost mode
4. Configure and switch to new system frequency.
5. Wait for at least 1 μ s and then reconfigure the AHB prescaler to get the needed HCLK clock frequency.

The sequence to switch from Range 1 boost mode to Range 1 normal mode is:

1. Set the R1MODE bit in the PWR_CR5 register.
2. Adjust the number of wait states according new frequency target in Range1 default mode.
3. Configure and switch to new system frequency.

6.2 Power supply supervisor

6.2.1 Power-on reset (POR) / power-down reset (PDR) / brown-out reset (BOR)

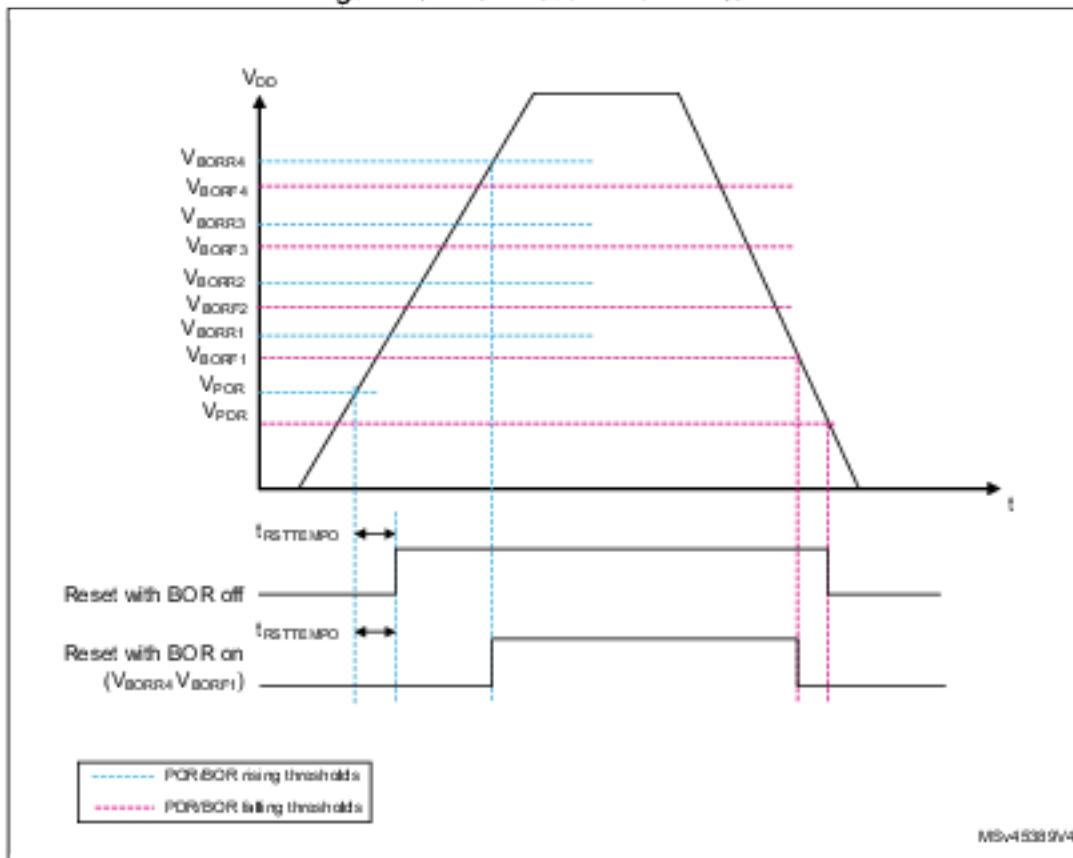
The device has an integrated power-on reset (POR) / power-down reset (PDR), coupled with a brown-out reset (BOR) circuitry. The BOR is active in all power modes except Shutdown mode, and cannot be disabled.

Five BOR thresholds can be selected through option bytes.

During power-on, the BOR keeps the device under reset until the supply voltage V_{DD} reaches the specified V_{BORx} threshold. When V_{DD} drops below the selected threshold, a device reset is generated. When V_{DD} is above the V_{BORx} upper limit, the device reset is released and the system can start.

For more details on the brown-out reset thresholds, refer to the electrical characteristics section in the datasheet.

Figure 13. Brown-out reset waveform



1. The reset temporization $t_{RSTTEMPO}$ is present only for the BOR lowest threshold (V_{BORR0}).

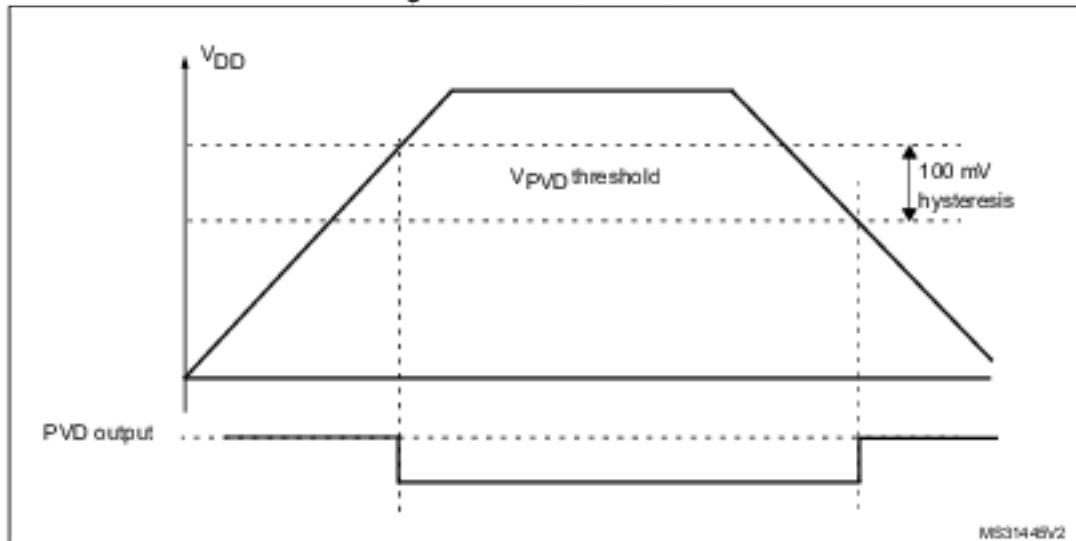
6.2.2 Programmable voltage detector (PVD)

You can use the PVD to monitor the V_{DD} power supply by comparing it to a threshold selected by the PLS[2:0] bits in the [Power control register 2 \(PWR_CR2\)](#).

The PVD is enabled by setting the PVDE bit.

A PVDO flag is available, in the [Power status register 2 \(PWR_SR2\)](#), to indicate if V_{DD} is higher or lower than the PVD threshold. This event is internally connected to the EXTI line16 and can generate an interrupt if enabled through the EXTI registers. The PVD output interrupt can be generated when V_{DD} drops below the PVD threshold and/or when V_{DD} rises above the PVD threshold depending on EXTI line16 rising/falling edge configuration. As an example, the service routine could perform emergency shutdown tasks.

Figure 14. PVD thresholds



6.2.3 Peripheral Voltage Monitoring (PVM)

Only V_{DD} is monitored by default, as it is the only supply required for all system-related functions. The V_{DDA} can be independent from V_{DD} and can be monitored with two peripheral voltage monitoring (PVM).

Each of the PVMx ($x=1, 2$) is a comparator between a fixed threshold V_{PVMx} and the V_{DDA} power supply. PVMOx flags indicate if the independent power supply is higher or lower than the PVMx threshold: PVMOx flag is cleared when the supply voltage is above the PVMx threshold, and is set when the supply voltage is below the PVMx threshold.

Each PVM output is connected to an EXTI line and can generate an interrupt if enabled through the EXTI registers. The PVMx output interrupt is generated when the independent power supply drops below the PVMx threshold and/or when it rises above the PVMx threshold, depending on EXTI line rising/falling edge configuration.

Each PVM can remain active in Stop 0 and Stop 1 modes, and the PVM interrupt can wake up from the Stop mode.

Table 39. PVM features

| PVM | Power supply | PVM threshold | EXTI line |
|------|--------------|----------------------------|-----------|
| PVM1 | V_{DDA} | V_{PVM1} (around 1.65 V) | 40 |
| PVM2 | V_{DDA} | V_{PVM2} (around 1.8 V) | 41 |

The independent analog supply V_{DDA} is not considered as present by default, and a logical and electrical isolation is applied to ignore any information coming from the peripherals supplied by this dedicated supply.

- If V_{DDA} is shorted externally to V_{DD} , the application should assume it is available without enabling any Peripheral Voltage Monitoring.
- If V_{DDA} is independent from V_{DD} , the Peripheral Voltage Monitoring (PVM) can be enabled to confirm whether the supply is present or not.

6.3 Low-power modes

By default, the microcontroller is in Run mode after a system or a power Reset. Several low-power modes are available to save power when the CPU does not need to be kept running, for example when waiting for an external event. It is up to the user to select the mode that gives the best compromise between low-power consumption, short startup time and available wake up sources.

The device features seven low-power modes:

- Sleep mode: CPU clock off, all peripherals including Cortex®-M4 with FPU core peripherals such as NVIC, SysTick, etc. can run and wake up the CPU when an interrupt or an event occurs. Refer to [Section 6.3.4: Sleep mode](#).
- Low-power run mode: This mode is achieved when the CPU clock frequency is reduced below 2 MHz. The code is executed from the SRAM or the Flash memory. The regulator is in low-power mode to minimize the regulator's operating current. Refer to [Section 6.3.2: Low-power run mode \(LP run\)](#).
- Low-power sleep mode: This mode is entered from the Low-power run mode: Cortex®-M4 with FPU is off. Refer to [Section 6.3.5: Low-power sleep mode \(LP sleep\)](#).
- Stop 0 and Stop 1 modes: SRAM and all registers content are retained. All clocks in the V_{CORE} domain are stopped, the PLL, the HSI16 and the HSE are disabled. The LSI and the LSE can be kept running.

The RTC and TAMP can remain active (Stop mode with RTC, Stop mode without RTC). Some peripherals with the wake up capability can enable the HSI16 RC during the Stop mode to detect their wake up condition.

In Stop 0 mode, the main regulator remain ON, which allows the fastest wake up time but with higher consumption. The active peripherals and the wakeup sources are the same as in Stop 1 mode.

The system clock, when exiting from Stop 0 or Stop 1 mode, is the HSI16 clock. If the device is configured to wake up in low-power run mode, the HPRE bits in RCC_CFG register must be configured prior to entering Stop mode to provide a frequency not greater than 2 MHz.

Refer to [Section 6.3.6: Stop 0 mode](#) for details on Stop 0 mode.

- Standby mode: V_{CORE} domain is powered off. However, it is possible to preserve the SRAM contents:
 - Standby mode with SRAM2 retention when the bit RRS is set in PWR_CR3 register. In this case, SRAM2 is supplied by the low-power regulator.
 - Standby mode when the bit RRS is cleared in PWR_CR3 register. In this case the main regulator and the low-power regulator are powered off.

All clocks in the V_{CORE} domain are stopped, the PLL, the HSI16 and the HSE oscillator are disabled. The LSI and the LSE can be kept running.

The RTC can remain active (Standby mode with RTC, Standby mode without RTC).

The system clock, when exiting Standby modes, is the HSI16 oscillator clock.

Refer to [Section 6.3.8: Standby mode](#).

- Shutdown mode: V_{CORE} domain is powered off. All clocks in the V_{CORE} domain are stopped, the PLL, the HSI16, the LSI and the HSE are disabled. The LSE can be kept running. The system clock, when exiting the Shutdown mode, is HSI16 oscillator clock. In this mode, the supply voltage monitoring is disabled and the product behavior is not guaranteed in case of a power voltage drop. Refer to [Section 6.3.9: Shutdown mode](#).

In addition, the power consumption in Run mode can be reduced by one of the following means:

- Slowing down the system clocks
- Gating the clocks to the APB and AHB peripherals when they are unused.

Figure 15. Low-power modes possible transitions

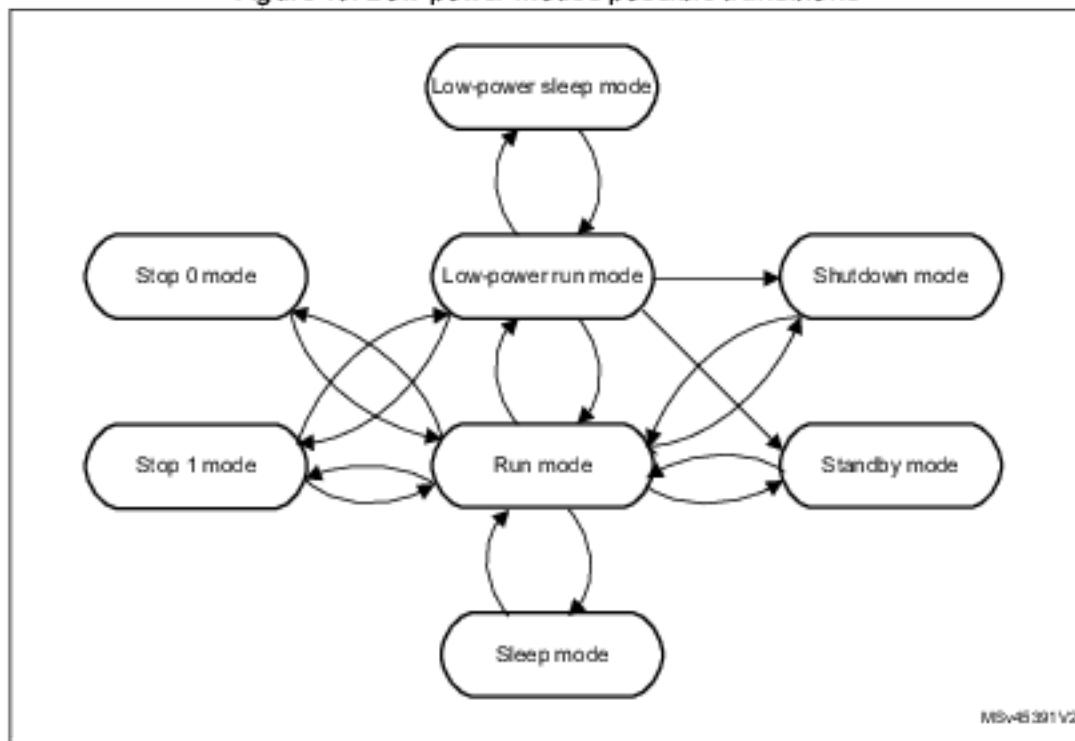


Table 40. Low-power mode summary

| Mode name | Entry | Wakeup source ⁽¹⁾ | Wakeup system clock | Effect on clocks | Voltage regulators | | | |
|---------------------------------------|--|---|--|--|--------------------|-----|--|--|
| | | | | | MR | LPR | | |
| Sleep (Sleep-now or Sleep-on-exit) | WFI or Return from ISR | Any interrupt | Same as before entering Sleep mode | CPU clock OFF no effect on other clocks or analog clock sources | ON | | | |
| | WFE | Wakeup event | | | | | | |
| Low-power run | Set LPR bit | Clear LPR bit | Same as Low-power run clock | None | OFF | | | |
| Low-power sleep | Set LPR bit + WFI or Return from ISR | Any interrupt | Same as before entering Low-power sleep mode | CPU clock OFF no effect on other clocks or analog clock sources | | | | |
| | Set LPR bit + WFE | Wakeup event | | | | | | |
| Stop 0 | LPMS="000" + SLEEPDEEP bit + WFI or Return from ISR or WFE | Any EXTI line (configured in the EXTI registers) Specific peripherals events | HSI16 | All clocks OFF except LSI and LSE | ON | | | |
| Stop 1 | LPMS="001" + SLEEPDEEP bit + WFI or Return from ISR or WFE | | | | | | | |
| Standby with SRAM2 | LPMS="011" + Set RRS bit + SLEEPDEEP bit + WFI or Return from ISR or WFE | WKUP pin edge, RTC event, TAMP event, external reset on NRST pin, IWDG reset | | All clocks OFF except LSE | OFF | | | |
| Standby | LPMS="011" + Clear RRS bit + SLEEPDEEP bit + WFI or Return from ISR or WFE | | | | | | | |
| Shutdown | LPMS="1--" + SLEEPDEEP bit + WFI or Return from ISR or WFE | WKUP pin edge, RTC event, TAMP event, external reset on NRST pin | | All clocks OFF except LSE | | OFF | | |

1. Refer to [Table 41: Functionalities depending on the working mode](#).

Table 41. Functionalities depending on the working mode⁽¹⁾

| Peripheral | Run | Sleep | Low-power run | Low-power sleep | Stop 0/1 | Standby | Shutdown | VBAT |
|-------------------------------------|------------------|------------------|------------------|------------------|------------------|-------------------|------------------|------|
| | | | | | - | Wakeup capability | - | |
| CPU | Y | - | Y | - | - | - | - | - |
| Flash memory | O ⁽²⁾ | O ⁽²⁾ | O ⁽²⁾ | O ⁽²⁾ | - | - | - | - |
| SRAM1 | Y | Y ⁽³⁾ | Y | Y ⁽³⁾ | Y | - | - | - |
| SRAM2 | Y | Y ⁽³⁾ | Y | Y ⁽³⁾ | Y | - | O ⁽⁴⁾ | - |
| CCM SRAM | Y | Y ⁽³⁾ | Y | Y ⁽³⁾ | Y | - | - | - |
| FSMC | O | O | O | O | - | - | - | - |
| QUADSPI | O | O | O | O | - | - | - | - |
| Backup Registers | Y | Y | Y | Y | Y | - | Y | - |
| Brown-out reset (BOR) | Y | Y | Y | Y | Y | Y | Y | - |
| Programmable Voltage Detector (PVD) | O | O | O | O | O | O | - | - |
| Peripheral Voltage Monitor (PVM) | O | O | O | O | O | O | - | - |
| DMA | O | O | O | O | - | - | - | - |
| Oscillator HS16 | O | O | O | O | (5) | - | - | - |
| Oscillator HS48 | O | O | - | - | - | - | - | - |
| High Speed External (HSE) | O | O | O | O | - | - | - | - |
| Low Speed Internal (LSI) | O | O | O | O | O | - | O | - |
| Low Speed External (LSE) | O | O | O | O | O | - | O | - |
| Clock Security System (CSS) | O | O | O | O | - | - | - | - |
| Clock Security System on LSE | O | O | O | O | O | O | O | - |
| RTC / Auto wakeup | O | O | O | O | O | O | O | O |
| Number of RTC Tamper pins | 3 | 3 | 3 | 3 | 3 | O | 3 | O |
| USB | O ^(B) | O ^(B) | - | - | O | - | - | - |
| USARTx (x=1,2,3,4,5) | O | O | O | O | O ⁽⁶⁾ | O ⁽⁶⁾ | - | - |
| Low-power UART (LPUART1) | O | O | O | O | O ⁽⁶⁾ | O ⁽⁶⁾ | - | - |
| I2Cx (x=1,2,3,4) | O | O | O | O | O ⁽⁷⁾ | O ⁽⁷⁾ | - | - |

Table 41. Functionalities depending on the working mode⁽¹⁾ (continued)

| Peripheral | Run | Sleep | Low-power run | Low-power sleep | Stop 0/1 | | Standby | | Shutdown | | VBAT |
|--|------------------|------------------|---------------|-----------------|-------------------|---|-------------------|---|-------------------------------|-------------------------------|------|
| | | | | | Wakeup capability | - | Wakeup capability | - | Wakeup capability | - | |
| SPIx (1,2,3,4) | ○ | ○ | ○ | ○ | - | - | - | - | - | - | - |
| FDCANx (1,2,3) | ○ | ○ | ○ | ○ | - | - | - | - | - | - | - |
| SAI1 | ○ | ○ | ○ | ○ | - | - | - | - | - | - | - |
| ADCx (x=1,2,3,4,5) | ○ | ○ | ○ | ○ | - | - | - | - | - | - | - |
| DACx (x=1,2,3,4) | ○ | ○ | ○ | ○ | ○ | - | - | - | - | - | - |
| VREFBUF | ○ | ○ | ○ | ○ | ○ | - | - | - | - | - | - |
| OPAMPx (x=1,2,3,4,5,6) | ○ | ○ | ○ | ○ | ○ | - | - | - | - | - | - |
| COMPx (x=1,2,3,4,5,6,7) | ○ | ○ | ○ | ○ | ○ | ○ | - | - | - | - | - |
| Temperature sensor | ○ | ○ | ○ | ○ | - | - | - | - | - | - | - |
| Timers (TIMx) | ○ | ○ | ○ | ○ | - | - | - | - | - | - | - |
| High resolution timer 1 (HRTIM1) | ○ | ○ | ○ | ○ | - | - | - | - | - | - | - |
| Low-power timer 1 (LPTIM1) | ○ | ○ | ○ | ○ | ○ | ○ | - | - | - | - | - |
| Independent watchdog (WDG) | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | - | - | - |
| Window watchdog (WWDG) | ○ | ○ | ○ | ○ | - | - | - | - | - | - | - |
| SysTick timer | ○ | ○ | ○ | ○ | - | - | - | - | - | - | - |
| Random number generator (RNG) | ○ ^(B) | ○ ^(B) | - | - | - | - | - | - | - | - | - |
| AES hardware accelerator | ○ | ○ | ○ | ○ | - | - | - | - | - | - | - |
| CRC calculation unit | ○ | ○ | ○ | ○ | - | - | - | - | - | - | - |
| GPIOs | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 5 pins (10) ⁽¹⁾ | 5 pins (10) ⁽¹⁾ | - |
| Filter Mathematical Accelerator (FMAC) | ○ | ○ | ○ | ○ | - | - | - | - | - | - | - |
| CORDIC co-processor (CORDIC) | ○ | ○ | ○ | ○ | - | - | - | - | - | - | - |

1. Legend: Y = Yes (Enable). O = Optional (Disable by default. Can be enabled by software). - = Not available . Wakeup highlighted in gray.

2. The Flash can be configured in power-down mode. By default, it is not in power-down mode.

3. The SRAM clock can be gated on or off.

4. SRAM2 content is preserved when the bit RRS is set in PWR_CR3 register.

5. Some peripherals with wakeup from Stop capability can request HSI16 to be enabled. In this case, HSI16 is woken up by the peripheral, and only feeds the peripheral which requested it. HSI16 is automatically put off when the peripheral does not need it anymore.
6. UART and LPUART reception is functional in Stop mode, and generates a wake up interrupt on Start, address match or received frame event.
7. I2C address detection is functional in Stop mode, and generates a wake up interrupt in case of address match.
8. Voltage scaling Range 1 only.
9. I/Os can be configured with internal pull-up, pull-down or floating in Standby mode.
10. The I/Os with wakeup from Standby/Shutdown capability are: PA0, PC13, PE6, PA2, PC5.
11. I/Os can be configured with internal pull-up, pull-down or floating in Shutdown mode but the configuration is lost when exiting the Shutdown mode.

Debug mode

By default, the debug connection is lost if the application puts the MCU in Stop0, Stop1, Standby or Shutdown mode while the debug features are used. This is due to the fact that the Cortex®-M4 with FPU core is no longer clocked.

However, by setting some configuration bits in the DBGMCU_CR register, the software can be debugged even when using the low-power modes extensively. For more details, refer to [Section 47.16.1: Debug support for low-power modes](#).

6.3.1 Run mode

Slowing down system clocks

In Run mode, the speed of the system clocks (SYSCLK, HCLK, PCLK) can be reduced by programming the prescaler registers. These prescalers can also be used to slow down the peripherals before entering the Sleep mode.

For more details, refer to [Section 7.4.3: Clock configuration register \(RCC_CFGR\)](#).

Peripheral clock gating

In Run mode, the HCLK and PCLK for individual peripherals and memories can be stopped at any time to reduce the power consumption.

To further reduce the power consumption in Sleep mode, the peripheral clocks can be disabled prior to executing the WFI or WFE instructions.

The peripheral clock gating is controlled by the RCC_AHBxENR and RCC_APBxENR registers.

Disabling the peripherals clocks in Sleep mode can be performed automatically by resetting the corresponding bit in the RCC_AHBxSMENR and RCC_APBxSMENR registers.

6.3.2 Low-power run mode (LP run)

To further reduce the consumption when the system is in Run mode, the regulator can be configured in low-power mode. In this mode, the CPU frequency should not exceed 2 MHz.

Please refer to the product datasheet for more details on voltage regulator and peripherals operating conditions.

I/O states in Low-power run mode

In Low-power run mode, all I/O pins keep the same state as in Run mode.

Entering the Low-power run mode

To enter the Low-power run mode, proceed as follows:

1. Optional: Jump into the SRAM and power-down the Flash by setting the RUN_PD bit in the *Flash access control register (FLASH_ACR)*.
2. Decrease the CPU clock frequency below 2 MHz.
3. Force the regulator in low-power mode by setting the LPR bit in the PWR_CR1 register.

Refer to [Table 42: Low-power run](#) on how to enter the Low-power run mode.

Exiting the Low-power run mode

To exit the Low-power run mode, proceed as follows:

1. Force the regulator in main mode by clearing the LPR bit in the PWR_CR1 register.
2. Wait until REGLPF bit is cleared in the PWR_SR2 register.
3. Increase the CPU clock frequency.

Refer to [Table 42: Low-power run](#) on how to exit the Low-power run mode.

Table 42. Low-power run

| Low-power run mode | Description |
|--------------------|--|
| Mode entry | Decrease the CPU clock frequency below 2 MHz LPR = 1 |
| Mode exit | LPR = 0 Wait until REGLPF = 0 Increase the CPU clock frequency |
| Wakeup latency | Regulator wakeup time from low-power mode |

6.3.3 Low power modes

Entering low power mode

Low power modes are entered by the MCU by executing the WFI (Wait For Interrupt), or WFE (Wait for Event) instructions, or when the SLEEPONEXIT bit in the Cortex®-M4 with FPU System Control register is set on Return from ISR.

Entering Low-power mode through WFI or WFE is executed only if no interrupt is pending or no event is pending.

Exiting low power mode

From Sleep modes, and Stop modes the MCU exits low power mode depending on the way the low power mode was entered:

- If the WFI instruction or Return from ISR was used to enter the low power mode, any peripheral interrupt acknowledged by the NVIC can wake up the device.
- If the WFE instruction is used to enter the low power mode, the MCU exits the low power mode as soon as an event occurs. The wakeup event can be generated either

by:

- NVIC IRQ interrupt
 - When SEVONPEND = 0 in the Cortex®-M4 with FPU System Control register. By enabling an interrupt in the peripheral control register and in the NVIC. When the MCU resumes from WFE, the peripheral interrupt pending bit and the NVIC peripheral IRQ channel pending bit (in the NVIC interrupt clear pending register) have to be cleared.

Only NVIC interrupts with sufficient priority wake up and interrupt the MCU.

- When SEVONPEND = 1 in the Cortex®-M4 with FPU System Control register.

By enabling an interrupt in the peripheral control register and optionally in the NVIC. When the MCU resumes from WFE, the peripheral interrupt pending bit and when enabled the NVIC peripheral IRQ channel pending bit (in the NVIC interrupt clear pending register) have to be cleared.

All NVIC interrupts wakeup the MCU, even the disabled ones. Only enabled NVIC interrupts with sufficient priority wake up and interrupt the MCU.

- Event

Configuring a EXTI line in event mode. When the CPU resumes from WFE, it is not necessary to clear the EXTI peripheral interrupt pending bit or the NVIC IRQ channel pending bit as the pending bits corresponding to the event line is not set.

It may be necessary to clear the interrupt flag in the peripheral.

From Standby modes, and Shutdown modes the MCU exit low power mode through an external reset (NRST pin), an IWDG reset, a rising edge on one of the enabled WKUPx pins or a RTC event occurs (see [Figure 529: RTC block diagrams](#)).

After waking up from Standby or Shutdown mode, program execution restarts in the same way as after a Reset (boot pin sampling, option bytes loading, reset vector is fetched, etc.).

6.3.4 Sleep mode

I/O states in Sleep mode

In Sleep mode, all I/O pins keep the same state as in Run mode.

Entering the Sleep mode

The Sleep mode is entered according [Section : Entering low power mode](#), when the SLEEPDEEP bit in the Cortex®-M4 with FPU System Control register is clear.

Refer to [Table 43: Sleep](#) for details on how to enter the Sleep mode.

Exiting the Sleep mode

The Sleep mode is exit according [Section : Exiting low power mode](#).

Refer to [Table 43: Sleep](#) for more details on how to exit the Sleep mode.

Table 43. Sleep

| Sleep-now mode | Description |
|----------------|--|
| Mode entry | WFI (Wait for interrupt) or WFE (Wait for Event) while: – SLEEPDEEP = 0 – No interrupt (for WFI) or event (for WFE) is pending Refer to the Cortex®-M4 with FPU System Control register. |
| | On return from ISR while: – SLEEPDEEP = 0 and – SLEEPONEXIT = 1 – No interrupt is pending Refer to the Cortex®-M4 with FPU System Control register. |
| Mode exit | If WFI or return from ISR was used for entry Interrupt: refer to Table 97: STM32G4 Series vector table If WFE was used for entry and SEVONPEND = 0: Wake up event: refer to Section 15.3.2: Wake up event management If WFE was used for entry and SEVONPEND = 1: Interrupt even when disabled in NVIC: refer to Table 97: STM32G4 Series vector table or Wakeup event: refer to Section 15.3.2: Wake up event management |
| Wakeup latency | None |

6.3.5 Low-power sleep mode (LP sleep)

Please refer to the product datasheet for more details on voltage regulator and peripherals operating conditions.

I/O states in Low-power sleep mode

In Low-power sleep mode, all I/O pins keep the same state as in Run mode.

Entering the Low-power sleep mode

The Low-power sleep mode is entered from low-power run mode according [Section : Entering low power mode](#), when the SLEEPDEEP bit in the Cortex®-M4 with FPU System Control register is clear.

Refer to [Table 44: Low-power sleep](#) for details on how to enter the Low-power sleep mode.

Exiting the Low-power sleep mode

The low-power Sleep mode is exit according [Section : Exiting low power mode](#). When exiting the Low-power sleep mode by issuing an interrupt or an event, the MCU is in Low-power run mode.

Refer to [Table 44: Low-power sleep](#) for details on how to exit the Low-power sleep mode.

Table 44. Low-power sleep

| Low-power sleep-now mode | Description |
|--------------------------|---|
| Mode entry | Low-power sleep mode is entered from the Low-power run mode. WFI (Wait for Interrupt) or WFE (Wait for Event) while: – SLEEPDEEP = 0 – No interrupt (for WFI) or event (for WFE) is pending Refer to the Cortex®-M4 with FPU System Control register. |
| Mode exit | Low-power sleep mode is entered from the Low-power run mode. On return from ISR while: – SLEEPDEEP = 0 and – SLEEPONEXIT = 1 – No interrupt is pending Refer to the Cortex®-M4 with FPU System Control register. |
| Wake-up latency | None |

6.3.6 Stop 0 mode

The Stop 0 mode is based on the Cortex®-M4 with FPU deep sleep mode combined with the peripheral clock gating. The voltage regulator is configured in main regulator mode. In Stop 0 mode, all clocks in the V_{CORE} domain are stopped; the PLL, the HSI16 and the HSE oscillators are disabled. Some peripherals with the wake-up capability (I2Cx (x=1,2,3,4), U(S)ARTx(x=1,2...5) and LPUART) can switch on the HSI16 to receive a frame, and switch off the HSI16 after receiving the frame if it is not a wakeup frame. In this case, the HSI16 clock is propagated only to the peripheral requesting it.

SRAM1, SRAM2, CCM SRAM and register contents are preserved.

The BOR is always available in Stop 0 mode. The consumption is increased when thresholds higher than V_{BOR0} are used.

I/O states in Stop 0 mode

In the Stop 0 mode, all I/O pins keep the same state as in the Run mode.

Entering the Stop 0 mode

The Stop 0 mode is entered according [Section : Entering lowpower mode](#), when the SLEEPDEEP bit in the Cortex®-M4 with FPU System Control register is set.

Refer to [Table 45: Stop 0 mode](#) for details on how to enter the Stop 0 mode.

If Flash memory programming is ongoing, the Stop 0 mode entry is delayed until the memory access is finished.

If an access to the APB domain is ongoing, The Stop 0 mode entry is delayed until the APB access is finished.

In Stop 0 mode, the following features can be selected by programming individual control bits:

- Independent watchdog (IWDG): the IWDG is started by writing to its Key register or by hardware option. Once started, it cannot be stopped except by a Reset. See [Section 4.2.3: IWDG functional description](#).
- real-time clock (RTC): this is configured by the RTCEN bit in the [RTC domain control register \(RCC_BDCR\)](#).
- Internal RC oscillator (LSI): this is configured by the LSION bit in the [Control/status register \(RCC_CSR\)](#).
- External 32.768 kHz oscillator (LSE): this is configured by the LSEON bit in the [RTC domain control register \(RCC_BDCR\)](#).

Several peripherals can be used in Stop 0 mode and can add consumption if they are enabled and clocked by LSI or LSE, or when they request the HSI16 clock: LPTIM1, I2Cx (x=1,2,3,4) U(S)ARTx(x=1,2..5), LPUART.

The DACx (x=1,2,3,4), the OPAMPs and the comparators can be used in Stop 0 mode, the PVM and the PVD as well. If they are not needed, they must be disabled by software to save their power consumptions.

The ADCx (x=1,2,3,4,5), temperature sensor and VREFBUF buffer can consume power during the Stop 0 mode, unless they are disabled before entering this mode.

Exiting the Stop 0 mode

The Stop 0 mode is exit according [Entering low power mode](#).

Refer to [Table 45: Stop 0 mode](#) for details on how to exit Stop 0 mode.

When exiting Stop 0 mode by issuing an interrupt or a wake up event, the HSI16 oscillator is selected as system clock. If the device is configured to wake up in Low-power run mode, the HPRE bits in RCC_CFG register must be configured prior to entering Stop 0 mode to provide a frequency not greater than 2 MHz.

When the voltage regulator operates in low-power mode, an additional startup delay is incurred when waking up from Stop 0 mode with HSI16. By keeping the internal regulator ON during Stop 0 mode, the consumption is higher although the startup time is reduced.

When exiting the Stop 0 mode, the MCU is either in Run mode (Range 1 or Range 2 depending on VOS bit in PWR_CR1) or in Low-power run mode if the bit LPR is set in the Power control register 1 (PWR_CR1).

Table 45. Stop 0 mode

| Stop 0 mode | Description |
|-----------------|--|
| Mode entry | WFI (Wait for Interrupt) or WFE (Wait for Event) while: – SLEEPDEEP bit is set in Cortex®-M4 with FPU System Control register – No interrupt (for WFI) or event (for WFE) is pending – LPMS = "000" in PWR_CR1 |
| | On Return from ISR while: – SLEEPDEEP bit is set in Cortex®-M4 with FPU System Control register – SLEEPONEXIT = 1 – No interrupt is pending – LPMS = "000" in PWR_CR1 |
| | <i>Note:</i> To enter Stop 0 mode, all EXTI Line pending bits (in Pending register 1 (EXTI_PRT1)), and the peripheral flags generating wake-up interrupts must be cleared. Otherwise, the Stop 0 mode entry procedure is ignored and program execution continues. |
| Mode exit | If WFI or Return from ISR was used for entry: Any EXTI Line configured in Interrupt mode (the corresponding EXTI Interrupt vector must be enabled in the NVIC). The interrupt source can be external interrupts or peripherals with wake up capability. Refer to Table 97: STM32G4 Series vector table . If WFE was used for entry and SEVONPEND = 0: Any EXTI Line configured in event mode. Refer to Section 15.3.2: Wakeup event management . If WFE was used for entry and SEVONPEND = 1: Any EXTI Line configured in Interrupt mode (even if the corresponding EXTI Interrupt vector is disabled in the NVIC). The interrupt source can be external interrupts or peripherals with wake up capability. Refer to Table 97: STM32G4 Series vector table . Wake up event: refer to Section 15.3.2: Wakeup event management |
| Wake up latency | Longest wake up time between: HSI16 wake up time and Flash wake up time from Stop 0 mode. |

6.3.7 Stop 1 mode

The Stop 1 mode is the same as Stop 0 mode except that the main regulator is OFF, and only the low-power regulator is ON. Stop 1 mode can be entered from Run mode and from Low-power run mode.

Refer to [Table 46: Stop 1 mode](#) for details on how to enter and exit Stop 1 mode.

Table 46. Stop 1 mode

| Stop 1 mode | Description |
|----------------|--|
| Mode entry | WFI (Wait for interrupt) or WFE (Wait for Event) while: – SLEEPDEEP bit is set in Cortex®-M4 with FPU System Control register – No interrupt (for WFI) or event (for WFE) is pending – LPMS = "001" in PWR_CR1 |
| | On Return from ISR while: – SLEEPDEEP bit is set in Cortex®-M4 with FPU System Control register – SLEEPONEXIT = 1 – No interrupt is pending – LPMS = "001" in PWR_CR1 |
| | Note: To enter Stop 1 mode, all EXTI Line pending bits (in Section 15.5.6: Pending register 1 (EXTI_PR1)), and the peripheral flags generating wakeup interrupts must be cleared. Otherwise, the Stop 1 mode entry procedure is ignored and program execution continues. |
| Mode exit | If WFI or Return from ISR was used for entry Any EXTI Line configured in Interrupt mode (the corresponding EXTI Interrupt vector must be enabled in the NVIC). The interrupt source can be external interrupts or peripherals with wakeup capability. Refer to Table 97: STM32G4 Series vector table . If WFE was used for entry and SEVONPEND = 0: Any EXTI Line configured in event mode. Refer to Section 15.3.2: Wakeup event management . If WFE was used for entry and SEVONPEND = 1: Any EXTI Line configured in Interrupt mode (even if the corresponding EXTI Interrupt vector is disabled in the NVIC). The interrupt source can be external interrupts or peripherals with wakeup capability. Refer to Table 97: STM32G4 Series vector table . Wakeup event: refer to Section 15.3.2: Wakeup event management |
| Wakeup latency | Longest wakeup time between: HSI16 wakeup time and regulator wakeup time from Low-power mode + Flash wakeup time from Stop 1 mode. |

6.3.8 Standby mode

The Standby mode allows to achieve the lowest power consumption with BOR. It is based on the Cortex®-M4 with FPU deepsleep mode, with the voltage regulators disabled (except when SRAM2 content is preserved). The PLL, the HSI16, and the HSE oscillators are also switched off.

SRAM1 and register contents are lost except for registers in the Backup domain and Standby circuitry (see [Figure 12](#)). SRAM2 content can be preserved if the bit RRS is set in the PWR_CR3 register. In this case the Low-power regulator is ON and provides the supply to SRAM2 only.

The BOR is always available in Standby mode. The consumption is increased when thresholds higher than V_{BOR0} are used.

I/O states in Standby mode

In the Standby mode, the I/Os can be configured either with a pull-up (refer to PWR_PUCRx registers ($x=A,B,C,D,E,F,G$)), or with a pull-down (refer to PWR_PDCRx registers ($x=A,B,C,D,E,F,G$)), or can be kept in analog state.

The RTC outputs on PC13 are functional in Standby mode. PC14 and PC15 used for LSE are also functional. 5 wake up pins (WKUPx, $x=1,2\dots 5$) and the 3 RTC tamper pins are available.

Entering Standby mode

The Standby mode is entered according [Section : Entering low power mode](#), when the SLEEPDEEP bit in the Cortex®-M4 with FPU System Control register is set.

Refer to [Table 47: Standby mode](#) for details on how to enter Standby mode.

In Standby mode, the following features can be selected by programming individual control bits:

- Independent watchdog (IWDG): the IWDG is started by writing to its Key register or by hardware option. Once started it cannot be stopped except by a reset. See [Section 42.3: IWDG functional description](#) in [Section 42: Independent watchdog \(MDG\)](#).
- real-time clock (RTC): this is configured by the RTCEN bit in the Backup domain control register (RCC_BDCR)
- Internal RC oscillator (LSI): this is configured by the LSION bit in the Control/status register (RCC_CSR).
- External 32.768 kHz oscillator (LSE): this is configured by the LSEON bit in the Backup domain control register (RCC_BDCR)

Exiting Standby mode

The Standby mode is exit according [Section : Entering low power mode](#). The SBF status flag in the [Power control register 3 \(PWR_CR3\)](#) indicates that the MCU was in Standby mode. All registers are reset after wake up from Standby except for [Power control register 3 \(PWR_CR3\)](#).

Refer to [Table 47: Standby mode](#) for more details on how to exit Standby mode.

Table 47. Standby mode

| Standby mode | Description |
|----------------|--|
| | WFI (Wait for interrupt) or WFE (Wait for Event) while: – SLEEPDEEP bit is set in Cortex®-M4 with FPU System Control register – No interrupt (for WFI) or event (for WFE) is pending – LPMS = "011" in PWR_CR1 – WUFx bits are cleared in power status register 1 (PWR_SR1) |
| Mode entry | On return from ISR while: – SLEEPDEEP bit is set in Cortex®-M4 with FPU System Control register – SLEEPONEXIT = 1 – No interrupt is pending – LPMS = "011" in PWR_CR1 and – WUFx bits are cleared in power status register 1 (PWR_SR1) – The RTC flag corresponding to the chosen wakeup source (RTC Alarm A, RTC Alarm B, RTC wakeup, tamper or timestamp flags) is cleared |
| Mode exit | WKUPx pin edge, RTC event, external Reset in NRST pin, WDG Reset, BOR reset |
| Wakeup latency | Reset phase |

6.3.9 Shutdown mode

The Shutdown mode allows to achieve the lowest power consumption. It is based on the deeplsleep mode, with the voltage regulator disabled. The V_{CORE} domain is consequently powered off. The PLL, the HSI16, the LSI and the HSE oscillators are also switched off.

SRAM1, SRAM2, CCM SRAM and register contents are lost except for registers in the Backup domain. The BOR is not available in Shutdown mode. No power voltage monitoring is possible in this mode, therefore the switch to Backup domain is not supported.

I/O states in Shutdown mode

In the Shutdown mode, the I/Os can be configured either with a pull-up (refer to PWR_PUCRx registers (x=A,B,C,D,E,F,G), or with a pull-down (refer to PWR_PDCRx registers (x=A,B,C,D,E,F,G)), or can be kept in analog state. However this configuration is lost when exiting the Shutdown mode due to the power-on reset.

The RTC outputs on PC13 are functional in Shutdown mode. PC14 and PC15 used for LSE are also functional. 5 wakeup pins (WKUPx, x=1,2...5) and the 3 RTC tamper are available.

Entering Shutdown mode

The Shutdown mode is entered according [Entering low power mode](#), when the SLEEPDEEP bit in the Cortex®-M4 with FPU System Control register is set.

Refer to [Table 48: Shutdown mode](#) for details on how to enter Shutdown mode.

In Shutdown mode, the following features can be selected by programming individual control bits:

- real-time clock (RTC): this is configured by the RTCEN bit in the Backup domain control register (RCC_BDCR). Caution: in case of VDD power-down the RTC content is lost.
- external 32.768 kHz oscillator (LSE): this is configured by the LSEON bit in the Backup domain control register (RCC_BDCR)

Exiting Shutdown mode

The Shutdown mode is exit according [Section : Exiting low power mode](#). A power-on reset occurs when exiting from Shutdown mode. All registers (except for the ones in the Backup domain) are reset after wakeup from Shutdown.

Refer to [Table 48: Shutdown mode](#) for more details on how to exit Shutdown mode.

Table 48. Shutdown mode

| Shutdown mode | Description |
|----------------|---|
| | WFI (Wait for Interrupt) or WFE (Wait for Event) while: – SLEEPDEEP bit is set in Cortex®-M4 with FPU System Control register – No interrupt (for WFI) or event (for WFE) is pending – LPMS = "1XX" in PWR_CR1 – WUFx bits are cleared in powerstatus register 1 (PWR_SR1) |
| Mode entry | On return from ISR while: – SLEEPDEEP bit is set in Cortex®-M4 with FPU System Control register – SLEEPONNEXT = 1 – No interrupt is pending – LPMS = "1XX" in PWR_CR1 and – WUFx bits are cleared in powerstatus register 1 (PWR_SR1) – The RTC flag corresponding to the chosen wakeup source (RTC Alarm A, RTC Alarm B, RTC wakeup, tamper or timestamp flags) is cleared |
| Mode exit | WKUPx pin edge, RTC event, external Reset in NRST pin |
| Wakeup latency | Reset phase |

6.3.10 Auto-wakeup from low-power mode

The RTC can be used to wake up the MCU from low-power mode without depending on an external interrupt (Auto-wakeup mode). The RTC provides a programmable time base for waking up from Stop (0 or 1) or Standby mode at regular intervals. For this purpose, two of the three alternative RTC clock sources can be selected by programming the RTCSEL[1:0] bits in the *RTC domain control register (RCC_BDCR)*:

- Low-power 32.768 kHz external crystal oscillator (LSE OSC)
This clock source provides a precise time base with very low-power consumption.
- Low-power internal RC Oscillator (LSI)
This clock source has the advantage of saving the cost of the 32.768 kHz crystal. This internal RC Oscillator is designed to add minimum power consumption.

To wake up from Stop mode with an RTC alarm event, it is necessary to:

- Configure the EXTI Line 17 to be sensitive to rising edge
- Configure the RTC to generate the RTC alarm

To wake up from Standby mode, there is no need to configure the EXTI Line 17.

To wake up from Stop mode with an RTC wakeup event, it is necessary to:

- Configure the EXTI Line 20 to be sensitive to rising edge
- Configure the RTC to generate the RTC alarm

To wake up from Standby mode, there is no need to configure the EXTI Line 20.

6.4 PWR registers

The peripheral registers can be accessed by half-words (16-bit) or words (32-bit).

6.4.1 Power control register 1 (PWR_CR1)

Address offset: 0x00

Reset value: 0x0000 0200

This register is reset after wake up from Standby mode.

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|----------|------|------|------|------|------|------|------|------|-----------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | LPR | Res. | Res. | Res. | VOS[1:0] | DBP | Res. | LPMS[2:0] | |
| rw | | | | | rw | rw | rw | | | | | | rw | rw | rw |

Bits 31:15 Reserved, must be kept at reset value.

Bit 14 **LPR**: Low-power run

When this bit is set, the regulator is switched from main mode (MR) to low-power mode (LPR).

Bits 13:11 Reserved, must be kept at reset value.

Bits 10:9 **VOS[1:0]**: Voltage scaling range selection

- 00: Cannot be written (forbidden by hardware)
- 01: Range 1
- 10: Range 2
- 11: Cannot be written (forbidden by hardware)

Bit 8 **DBP**: Disable backup domain write protection

In reset state, the RTC and backup registers are protected against parasitic write access. This bit must be set to enable write access to these registers.

- 0: Access to RTC and Backup registers disabled
- 1: Access to RTC and Backup registers enabled

Bits 7:3 Reserved, must be kept at reset value.

Bits 2:0 **LPMS[2:0]**: Low-power mode selection

These bits select the low-power mode entered when CPU enters the deepsleep mode.

- 000: Stop 0 mode
- 001: Stop 1 mode
- 010: Reserved
- 011: Standby mode
- 1xx: Shutdown mode

Note: In Standby mode, SRAM2 can be preserved or not depending on RRS bit configuration in PWR_CR3.

6.4.2 Power control register 2 (PWR_CR2)

Address offset: 0x04

Reset value: 0x0000 0000

This register is reset when exiting Standby mode.

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|-------|-------|------|------|----------|----------|----------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | PVME2 | PVME1 | Res. | Res. | PLS[2:0] | PLS[2:0] | PLS[2:0] | PVDE |
| | | | | | | | | rw | rw | | | rw | rw | rw | rw |

Bits 31:8 Reserved, must be kept at reset value.

Bit 7 **PVME2**: Peripheral voltage monitoring 2 enable: V_{DDA} vs. DAC 1MSPS /DAC 15MSPS min voltage.

- 0: PVM2 (V_{DDA} monitoring vs. 1.8 V threshold) disable.
- 1: PVM2 (V_{DDA} monitoring vs. 1.8 V threshold) enable.

Bit 6 **PVME1**: Peripheral voltage monitoring 1 enable: V_{DDA} vs. ADC/COMP min voltage 1.62V

- 0: PVM1 (V_{DDA} monitoring vs. 1.62V threshold) disable.
- 1: PVM1 (V_{DDA} monitoring vs. 1.62V threshold) enable.

Bits 5:4 Reserved, must be kept at reset value.

Bits 3:1 **PLS[2:0]**: Programmable voltage detector level selection.

These bits select the PVD falling threshold:

- 000: V_{PVD0} PVD threshold 0
- 001: V_{PVD1} PVD threshold 1
- 010: V_{PVD2} PVD threshold 2
- 011: V_{PVD3} PVD threshold 3
- 100: V_{PVD4} PVD threshold 4
- 101: V_{PVD5} PVD threshold 5
- 110: V_{PVD6} PVD threshold 6
- 111: External input analog voltage PVD_IN (compared internally to V_{REFINT})

Note: These bits are write-protected when the PVDL bit is set in the SYSCFG_CFGR2 register. The protection can be reset only by a system reset.

Bit 0 **PVDE**: Programmable voltage detector enable

- 0: Programmable voltage detector disable.
- 1: Programmable voltage detector enable.

Note: This bit is write-protected when the PVDL bit is set in the SYSCFG_CFGR2 register. The protection can be reset only by a system reset.

6.4.3 Power control register 3 (PWR_CR3)

Address offset: 0x08

Reset value: 0x0000 8000

This register is not reset when exiting Standby modes and with the PWRRST bit in the RCC_APB1RSTR1 register.

Access: Additional APB cycles are needed to access this register vs. a standard APB access (3 for a write and 2 for a read).

| | | | | | | | | | | | | | | | |
|------|-------------|-------------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EWUL | UCPD1_DBDIS | UCPD1_STDBY | Res. | Res. | APC | Res. | RRS | Res. | Res. | Res. | EWUP5 | EWUP4 | EWUP3 | EWUP2 | EWUP1 |
| rw | rw | rw | | | rw | | rw | | | | rw | rw | rw | rw | rw |

Bits 31:16 Reserved, must be kept at reset value.

Bit 15 **EWUL:** Enable internal wakeup line

- 0: Internal wakeup line disable.
- 1: Internal wakeup line enable.

Bit 14 **UCPD1_DBDIS:** USB Type-C and Power Delivery Dead Battery disable.

After exiting reset, the USB Type-C "dead battery" behavior is enabled, which may have a pull-down effect on CC1 and CC2 pins. It is recommended to disable it in all cases, either to stop this pull-down or to hand over control to the UCPD1 (which should therefore be initialized before doing the disable).

- 0: Enable USB Type-C dead battery pull-down behavior on UCPD1_CC1 and UCPD1_CC2 pins.
- 1: Disable USB Type-C dead battery pull-down behavior on UCPD1_CC1 and UCPD1_CC2 pins.

Bit 13 **UCPD1_STDBY:** UCPD1_STDBY USB Type-C and Power Delivery standby mode.

- 0: Write '0' immediately after standby exit when using UCPD1, (and before writing any UCPD1 registers).
- 1: Write '1' just before entering standby when using UCPD1.

Bits 12:11 Reserved, must be kept at reset value.

Bit 10 **APC:** Apply pull-up and pull-down configuration

When this bit is set, the I/O pull-up and pull-down configurations defined in the PWR_PUCRx and PWR_PDCRx registers are applied. When this bit is cleared, the PWR_PUCRx and PWR_PDCRx registers are not applied to the IOs.

Bit 9 Reserved, must be kept at reset value.

Bit 8 **RRS:** SRAM2 retention in Standby mode

- 0: SRAM2 is powered off in Standby mode (SRAM2 content is lost).
- 1: SRAM2 is powered by the low-power regulator in Standby mode (SRAM2 content is kept).

Bits 7:5 Reserved, must be kept at reset value.

Bit 4 **EWUP5:** Enable Wakeup pin WKUP5

When this bit is set, the external wakeup pin WKUP5 is enabled and triggers a wakeup from Standby or Shutdown event when a rising or a falling edge occurs. The active edge is configured via the WP5 bit in the PWR_CCR4 register.

Bit 3 **EWUP4:** Enable Wakeup pin WKUP4

When this bit is set, the external wakeup pin WKUP4 is enabled and triggers a wakeup from Standby or Shutdown event when a rising or a falling edge occurs. The active edge is configured via the WP4 bit in the PWR_CCR4 register.

Bit 2 EWUP3: Enable Wakeup pin WKUP3

When this bit is set, the external wakeup pin WKUP3 is enabled and triggers a wakeup from Standby or Shutdown event when a rising or a falling edge occurs. The active edge is configured via the WP3 bit in the PWR_CR4 register.

Bit 1 EWUP2: Enable Wakeup pin WKUP2

When this bit is set, the external wakeup pin WKUP2 is enabled and triggers a wakeup from Standby or Shutdown event when a rising or a falling edge occurs. The active edge is configured via the WP2 bit in the PWR_CR4 register.

Bit 0 EWUP1: Enable Wakeup pin WKUP1

When this bit is set, the external wakeup pin WKUP1 is enabled and triggers a wakeup from Standby or Shutdown event when a rising or a falling edge occurs. The active edge is configured via the WP1 bit in the PWR_CR4 register.

6.4.4 Power control register 4 (PWR_CR4)

Address offset: 0x0C

Reset value: 0x0000 0000

This register is not reset when exiting Standby modes and with the PWRRST bit in the RCC_APB1RSTR1 register.

Access: Additional APB cycles are needed to access this register vs. a standard APB access (3 for a write and 2 for a read).

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | Res. | Res. | Res. | Res. | VBR8 | VBE | Res. | Res. | Res. | WP5 | WP4 | WP3 | WP2 | WP1 |
| | | | | | | rw | rw | | | | rw | rw | rw | rw | rw |

Bits 31:10 Reserved, must be kept at reset value.

Bit 9 VBR8: V_{BAT} battery charging resistor selection

- 0: Charge V_{BAT} through a 5 kOhms resistor
- 1: Charge V_{BAT} through a 1.5 kOhms resistor

Bit 8 VBE: V_{BAT} battery charging enable

- 0: V_{BAT} battery charging disable
- 1: V_{BAT} battery charging enable

Bits 7:5 Reserved, must be kept at reset value.

Bit 4 WP5: Wakeup pin WKUP5 polarity

This bit defines the polarity used for an event detection on external wake-up pin, WKUP5

- 0: Detection on high level (rising edge)
- 1: Detection on low level (falling edge)

Bit 3 WP4: Wakeup pin WKUP4 polarity

This bit defines the polarity used for an event detection on external wake-up pin, WKUP4

- 0: Detection on high level (rising edge)
- 1: Detection on low level (falling edge)

Bit 2 WP3: Wakeup pin WKUP3 polarity

This bit defines the polarity used for an event detection on external wake-up pin, WKUP3

0: Detection on high level (rising edge)

1: Detection on b/w level (falling edge)

Bit 1 WP2: Wakeup pin WKUP2 polarity

This bit defines the polarity used for an event detection on external wake-up pin, WKUP2

0: Detection on high level (rising edge)

1: Detection on b/w level (falling edge)

Bit 0 WP1: Wakeup pin WKUP1 polarity

This bit defines the polarity used for an event detection on external wake-up pin, WKUP1

0: Detection on high level (rising edge)

1: Detection on b/w level (falling edge)

6.4.5 Power status register 1 (PWR_SR1)

Address offset: 0x10

Reset value: 0x0000 0000

This register is not reset when exiting Standby modes and with the PWRRST bit in the RCC_APB1RSTR1 register.

Access: 2 additional APB cycles are needed to read this register vs. a standard APB read.

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WUF1 | Res. | Res. | Res. | Res. | Res. | Res. | SBF | Res. | Res. | Res. | WUF5 | WUF4 | WUF3 | WUF2 | WUF1 |
| r | | | | | | | r | | | | r | r | r | r | r |

Bits 31:16 Reserved, must be kept at reset value.

Bit 15 WUF1: Wakeup flag internal

This bit is set when a wakeup is detected on the internal wakeup line. It is cleared when all internal wakeup sources are cleared.

Bits 14:9 Reserved, must be kept at reset value.

Bit 8 SBF: Standby flag

This bit is set by hardware when the device enters the Standby mode and is cleared by setting the CSBF bit in the PWR_SCR register, or by a power-on reset. It is not cleared by the system reset.

0: The device did not enter the Standby mode

1: The device entered the Standby mode

Bits 7:5 Reserved, must be kept at reset value.

Bit 4 WUF5: Wakeup flag 5

This bit is set when a wakeup event is detected on wakeup pin, WKUP5. It is cleared by writing '1' in the CWUF5 bit of the PWR_SCR register.

Bit 3 WUF4: Wakeup flag 4

This bit is set when a wakeup event is detected on wakeup pin, WKUP4. It is cleared by writing '1' in the CWUF4 bit of the PWR_SCR register.

Bit 2 WUF3: Wakeup flag 3

This bit is set when a wakeup event is detected on wakeup pin, WKUP3. It is cleared by writing '1' in the CWUF3 bit of the PWR_SCR register.

Bit 1 WUF2: Wakeup flag 2

This bit is set when a wakeup event is detected on wakeup pin, WKUP2. It is cleared by writing '1' in the CWUF2 bit of the PWR_SCR register.

Bit 0 WUF1: Wakeup flag 1

This bit is set when a wakeup event is detected on wakeup pin, WKUP1. It is cleared by writing '1' in the CWUF1 bit of the PWR_SCR register.

6.4.6 Power status register 2 (PWR_SR2)

Address offset: 0x14

Reset value: 0x0000 0000

This register is partially reset when exiting Standby/Shutdown modes.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------|-------|------|------|------|------|--------|--------|-----------|------|------|------|------|------|------|------|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PVM02 | PVM01 | Res. | Res. | PVDD | VDSF | REGLPF | REGLPS | FLASH_RDY | Res. |
| r | r | | | r | r | r | r | r | | | | | | | |

Bits 31:16 Reserved, must be kept at reset value.

Bit 15 PVM02: Peripheral voltage monitoring output: V_{DDA} vs. 1.8 V

0: V_{DDA} voltage is above PVM2 thresh hold (around 1.8 V).

1: V_{DDA} voltage is below PVM2 thresh hold (around 1.8 V).

Note: PVM02 is cleared when PVM2 is disabled (PVME2 = 0). After enabling PVM2, the PVM2 output is valid after the PVM2 wakeup time.

Bit 14 PVM01: Peripheral voltage monitoring output: V_{DDA} vs. 1.62 V

0: V_{DDA} voltage is above PVM1 thresh hold (around 1.62 V).

1: V_{DDA} voltage is below PVM1 thresh hold (around 1.62 V).

Note: PVM01 is cleared when PVM1 is disabled (PVME1 = 0). After enabling PVM1, the PVM1 output is valid after the PVM1 wakeup time.

Bits 13:12 Reserved, must be kept at reset value.

Bit 11 PVDO: Programmable voltage detector output

0: V_{DD} is above the selected PVD threshold

1: V_{DD} is below the selected PVD threshold

Bit 10 VOSF: Voltage scaling flag

A delay is required for the internal regulator to be ready after the voltage scaling has been changed. VOSF indicates that the regulator reached the voltage level defined with VOS bits of the PWR_CR1 register.

0: The regulator is ready in the selected voltage range

1: The regulator output voltage is changing to the required voltage level

Bit 9 REGLPF: Low-power regulator flag

This bit is set by hardware when the MCU is in Low-power run mode. When the MCU exits the Low-power run mode, this bit remains at 1 until the regulator is ready in main mode. A polling on this bit must be done before increasing the product frequency.

This bit is cleared by hardware when the regulator is ready.

0: The regulator is ready in main mode (MR)

1: The regulator is in low-power mode (LPR)

Bit 8 REGLPS: Low-power regulator started

This bit provides the information whether the low-power regulator is ready after a power-on reset or a Standby/Shutdown. If the Standby mode is entered while REGLPS bit is still cleared, the wakeup from Standby mode time may be increased.

0: The low-power regulator is not ready

1: The low-power regulator is ready

Bit 7 FLASH_RDY: Flash ready flag

This bit is set by hardware to indicate when the Flash memory is ready to be accessed after wakeup from power-down. To place the Flash memory in power-down, set either FPD_LPRUN, FPD_LPSLP or FPD_STP bits.

0: Flash memory in power-down

1: Flash memory ready to be accessed

Note: If the system boots from SRAM, the user application must wait till FLASH_RDY bit is set prior to jumping to Flash memory.

Bits 6:0 Reserved, must be kept at reset value.

6.4.7 Power status clear register (PWR_SCR)

Address offset: 0x18

Reset value : 0x0000 0000

Access: 3 additional APB cycles are needed to write this register vs. a standard APB write.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|------|------|------|------|------|------|-------|------|------|------|-----------|-----------|-----------|-----------|-----------|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | CSEBF | Res. | Res. | Res. | CWUF 5 | CWUF 4 | CWUF 3 | CWUF 2 | CWUF 1 |
| | | | | | | | w | | | | w | w | w | w | w |

Bits 31:9 Reserved, must be kept at reset value.

Bit 8 CSEBF: Clear standby flag

Setting this bit clears the SBF flag in the PWR_SR1 register.

Bits 7:5 Reserved, must be kept at reset value.

Bit 4 CWUF5: Clearwakeup flag 5

Setting this bit clears the WUF5 flag in the PWR_SR1 register.

Bit 3 CWUF4: Clearwakeup flag 4

Setting this bit clears the WUF4 flag in the PWR_SR1 register.

Bit 2 CWUF3: Clearwakeup flag 3

Setting this bit clears the WUF3 flag in the PWR_SR1 register.

Bit 1 CWUF2: Clearwakeup flag 2

Setting this bit clears the WUF2 flag in the PWR_SR1 register.

Bit 0 CWUF1: Clearwakeup flag 1

Setting this bit clears the WUF1 flag in the PWR_SR1 register.

6.4.8 Power Port A pull-up control register (PWR_PUCRA)

Address offset: 0x20.

Reset value: 0x0000 0000

This register is not reset when exiting Standby modes and with PWRRST bit in the RCC_APB1RSTR1 register.

Access: Additional APB cycles are needed to access this register vs. a standard APB access (3 for a write and 2 for a read).

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PU15 | Res. | PU13 | PU12 | PU11 | PU10 | PU9 | PU8 | PU7 | PU6 | PU5 | PU4 | PU3 | PU2 | PU1 | PU0 |
| rw | | rw |

Bits 31:16 Reserved, must be kept at reset value.

Bit 15 PU15: PortA pull-up bit 15

When set, this bit activates the pull-up on PA[15] when APC bit is set in PWR_CR3 register.
The pull-up is not activated if the corresponding PD15 bit is also set.

Bit 14 Reserved, must be kept at reset value.

Bits 13:0 PUy: PortA pull-up bit y (y = 13 to 0)

When set, this bit activates the pull-up on PA[y] when APC bit is set in PWR_CR3 register.
The pull-up is not activated if the corresponding PDy bit is also set.

6.4.9 Power Port A pull-down control register (PWR_PDCRA)

Address offset: 0x24.

Reset value: 0x0000 0000

This register is not reset when exiting Standby modes and with PWRRST bit in the RCC_APB1RSTR1 register.

Access: Additional APB cycles are needed to access this register vs. a standard APB access (3 for a write and 2 for a read).

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Res. |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | PD14 | Res. | PD12 | PD11 | PD10 | PD9 | PD8 | PD7 | PD6 | PD5 | PD4 | PD3 | PD2 | PD1 | PD0 |
| | rw | | rw |

Bits 31:15 Reserved, must be kept at reset value.

Bit 14 **PD14**: PortA pull-down bit 14

When set, this bit activates the pull-down on PA[14] when APC bit is set in PWR_CR3 register.

Bit 13 Reserved, must be kept at reset value.

Bits 12:0 **PDy**: PortA pull-down bit y (y = 12 to 0)

When set, this bit activates the pull-down on PA[y] when APC bit is set in PWR_CR3 register.

6.4.10 Power Port B pull-up control register (PWR_PUCRB)

Address offset: 0x28.

Reset value: 0x0000 0000

This register is not reset when exiting Standby modes and with PWRRST bit in the RCC_APB1RSTR1 register.

Access: Additional APB cycles are needed to access this register vs. a standard APB access (3 for a write and 2 for a read).

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Res. |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PU15 | PU14 | PU13 | PU12 | PU11 | PU10 | PU9 | PU8 | PU7 | PU6 | PU5 | PU4 | PU3 | PU2 | PU1 | PU0 |
| | rw | | rw |

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **PUs**: Port B pull-up bit y (y = 15 to 0)

When set, this bit activates the pull-up on PB[y] when APC bit is set in PWR_CR3 register.
The pull-up is not activated if the corresponding PDy bit is also set.

6.4.11 Power Port B pull-down control register (PWR_PDCRB)

Address offset: 0x2C.

Reset value: 0x0000 0000

This register is not reset when exiting Standby modes and with PWRRST bit in the RCC_APB1RSTR1 register.

Access: Additional APB cycles are needed to access this register vs. a standard APB access (3 for a write and 2 for a read).

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | B | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PD15 | PD14 | PD13 | PD12 | PD11 | PD10 | PD9 | PD8 | PD7 | PD6 | PD5 | Res. | PD3 | PD2 | PD1 | PD0 |
| rw | | rw | rw | rw | rw |

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:5 PDy: Port B pull-down bit y (y = 15 to 5)

When set, this bit activates the pull-down on PB[y] when APC bit is set in PWR_CR3 register.

Bit 4 Reserved, must be kept at reset value.

Bits 3:0 PDy: Port B pull-down bit y (y = 3 to 0)

When set, this bit activates the pull-down on PB[y] when APC bit is set in PWR_CR3 register.

6.4.12 Power Port C pull-up control register (PWR_PUCRC)

Address offset: 0x30.

Reset value: 0x0000 0000

This register is not reset when exiting Standby modes and with PWRRST bit in the RCC_APB1RSTR1 register.

Access: Additional APB cycles are needed to access this register vs. a standard APB access (3 for a write and 2 for a read).

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | B | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PU15 | PU14 | PU13 | PU12 | PU11 | PU10 | PU9 | PU8 | PU7 | PU6 | PU5 | PU4 | PU3 | PU2 | PU1 | PU0 |
| rw |

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 PUy: Port C pull-up bit y (y = 15 to 0)

When set, this bit activates the pull-up on PC[y] when APC bit is set in PWR_CR3 register.

The pull-up is not activated if the corresponding PDy bit is also set.

6.4.13 Power Port C pull-down control register (PWR_PDCRC)

Address offset: 0x34.

Reset value: 0x0000 0000

This register is not reset when exiting Standby modes and with PWRRST bit in the RCC_APB1RSTR1 register.

Access: Additional APB cycles are needed to access this register vs. a standard APB access (3 for a write and 2 for a read).

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Res. |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | B | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PD15 | PD14 | PD13 | PD12 | PD11 | PD10 | PD9 | PDB | PD7 | PD6 | PD5 | PD4 | PD3 | PD2 | PD1 | PDD |
| rw |

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **PDy**: Port C pull-down bit y (y = 15 to 0)

When set, this bit activates the pull-down on PC[y] when APC bit is set in PWR_CR3 register.

6.4.14 Power Port D pull-up control register (PWR_PUCRD)

Address offset: 0x38.

Reset value: 0x0000 0000

This register is not reset when exiting Standby modes and with PWRRST bit in the RCC_APB1RSTR1 register.

Access: Additional APB cycles are needed to access this register vs. a standard APB access (3 for a write and 2 for a read).

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Res. |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | B | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PU15 | PU14 | PU13 | PU12 | PU11 | PU10 | PU9 | PUB | PUT | PUB | PU5 | PU4 | PU3 | PU2 | PU1 | PUD |
| rw |

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **PUy**: Port D pull-up bit y (y = 15 to 0)

When set, this bit activates the pull-up on PD[y] when APC bit is set in PWR_CR3 register. The pull-up is not activated if the corresponding PDy bit is also set.

6.4.15 Power Port D pull-down control register (PWR_PDCRD)

Address offset: 0x3C.

Reset value: 0x0000 0000

This register is not reset when exiting Standby modes and with PWRRST bit in the RCC_APB1RSTR1 register.

Access: Additional APB cycles are needed to access this register vs. a standard APB access (3 for a write and 2 for a read).

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PD15 | PD14 | PD13 | PD12 | PD11 | PD10 | PD9 | PD8 | PD7 | PD6 | PD5 | PD4 | PD3 | PD2 | PD1 | PD0 |
| RW |

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **PDy**: Port D pull-down bit y (y = 15 to 0)

When set, this bit activates the pull-down on PD[y] when APC bit is set in PWR_CR3 register.

6.4.16 Power Port E pull-up control register (PWR_PUCRE)

Address offset: 0x40.

Reset value: 0x0000 0000

This register is not reset when exiting Standby modes and with PWRRST bit in the RCC_APB1RSTR1 register.

Access: Additional APB cycles are needed to access this register vs. a standard APB access (3 for a write and 2 for a read).

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PU15 | PU14 | PU13 | PU12 | PU11 | PU10 | PU9 | PU8 | PU7 | PU6 | PU5 | PU4 | PU3 | PU2 | PU1 | PU0 |
| RW |

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **PUy**: Port E pull-up bit y (y = 15 to 0)

When set, this bit activates the pull-up on PE[y] when APC bit is set in PWR_CR3 register. The pull-up is not activated if the corresponding PDy bit is also set.

6.4.17 Power Port E pull-down control register (PWR_PDCRE)

Address offset: 0x44.

Reset value: 0x0000 0000

This register is not reset when exiting Standby modes and with PWRRST bit in the RCC_APB1RSTR1 register.

Access: Additional APB cycles are needed to access this register vs. a standard APB access (3 for a write and 2 for a read).

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PD15 | PD14 | PD13 | PD12 | PD11 | PD10 | PD9 | PD8 | PD7 | PD6 | PD5 | PD4 | PD3 | PD2 | PD1 | PD0 |
| RW |

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **PDy**: Port E pull-down bit y (y = 15 to 0)

When set, this bit activates the pull-down on PE[y] when APC bit is set in PWR_CR3 register.

6.4.18 Power Port F pull-up control register (PWR_PUCRF)

Address offset: 0x48.

Reset value: 0x0000 0000

This register is not reset when exiting Standby modes and with PWRRST bit in the RCC_APB1RSTR1 register.

Access: Additional APB cycles are needed to access this register vs. a standard APB access (3 for a write and 2 for a read).

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PU15 | PU14 | PU13 | PU12 | PU11 | PU10 | PU9 | PU8 | PU7 | PU6 | PU5 | PU4 | PU3 | PU2 | PU1 | PU0 |
| RW |

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **PUsy**: Port F pull-up bit y (y = 15 to 0)

When set, this bit activates the pull-up on PF[y] when APC bit is set in PWR_CR3 register. The pull-up is not activated if the corresponding PDy bit is also set.

6.4.19 Power Port F pull-down control register (PWR_PDCRF)

Address offset: 0x4C.

Reset value: 0x0000 0000

This register is not reset when exiting Standby modes and with PWRRST bit in the RCC_APB1RSTR1 register.

Access: Additional APB cycles are needed to access this register vs. a standard APB access (3 for a write and 2 for a read).

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PD15 | PD14 | PD13 | PD12 | PD11 | PD10 | PD9 | PD8 | PD7 | PD6 | PD5 | PD4 | PD3 | PD2 | PD1 | PD0 |
| RW |

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **PDy**: Port F pull-down bit y (y = 15 to 0)

When set, this bit activates the pull-down on PF[y] when APC bit is set in PWR_CR3 register.

6.4.20 Power Port G pull-up control register (PWR_PUCRG)

Address offset: 0x50.

Reset value: 0x0000 0000

This register is not reset when exiting Standby modes and with PWRRST bit in the RCC_APB1RSTR1 register.

Access: Additional APB cycles are needed to access this register vs. a standard APB access (3 for a write and 2 for a read).

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | Res. | Res. | Res. | PU10 | PU9 | PU8 | PU7 | PU6 | PU5 | PU4 | PU3 | PU2 | PU1 | PU0 |
| | | | | | RW |

Bits 31:11 Reserved, must be kept at reset value.

Bits 10:0 **PUy**: Port G pull-up bit y (y = 10 to 0)

When set, this bit activates the pull-up on PG[y] when APC bit is set in PWR_CR3 register. The pull-up is not activated if the corresponding PDy bit is also set.

6.4.21 Power Port G pull-down control register (PWR_PDCRG)

Address offset: 0x54.

Reset value: 0x0000 0000

This register is not reset when exiting Standby modes and with PWRRST bit in the RCC_APB1RSTR1 register.

Access: Additional APB cycles are needed to access this register vs. a standard APB access (3 for a write and 2 for a read).

| | | | | | | | | | | | | | | | |
|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res | Res. | Res. | Res. | Res. | PD10 | PD9 | PD8 | PD7 | PD6 | PD5 | PD4 | PD3 | PD2 | PD1 | PD0 |
| | | | | | rw |

Bits 31:11 Reserved, must be kept at reset value.

Bits 10:0 **PDy**: Port G pull-down bit y (y = 10 to 0)

When set, this bit activates the pull-down on PG[y] when APC bit is set in PWR_CR3 register.

6.4.22 Power control register (PWR_CR5)

Address offset: 0x80.

Reset value: 0x0000 0100

This register is reset only by power-on reset.

| | | | | | | | | | | | | | | | |
|-----|------|------|------|------|------|------|--------|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res | Res. | Res. | Res. | Res. | Res. | Res. | R1MODE | Res. |
| | | | | | | | rw | | | | | | | | |

Bits 31:9 Reserved, must be kept at reset value.

Bit 8 **R1MODE**: Main regular range 1 mode

This bit is only valid for the main regulator in range 1 and has no effect on range 2. It is recommended to reset this bit when the system frequency is greater than 150 MHz. Refer to [Table 38: Range 1 boost mode configuration](#).

0: Main regulator in range 1 boost mode.

1: Main regulator in range 1 normal mode.

Bits 7:0 Reserved, must be kept at reset value.

6.4.23 PWR register map and reset value table

Table 49. PWR register map and reset values

Table 49. PWR register map and reset values (continued)

| Offset | Register | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|--------|-------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 0x3C | PWR_PDCRD | 0x00 |
| | Reset value | 0x00 |
| 0x40 | PWR_PUCRE | 0x00 |
| | Reset value | 0x00 |
| 0x44 | PWR_PDCRE | 0x00 |
| | Reset value | 0x00 |
| 0x48 | PWR_PUCRF | 0x00 |
| | Reset value | 0x00 |
| 0x4C | PWR_PDCRF | 0x00 |
| | Reset value | 0x00 |
| 0x50 | PWR_PUCRG | 0x00 |
| | Reset value | 0x00 |
| 0x54 | PWR_PDCRG | 0x00 |
| | Reset value | 0x00 |
| 0x80 | PWR_CRS | 0x00 |
| | Reset value | 0x00 |

Refer to [Section 2.2 on page 81](#) for the register boundary addresses.

7 Reset and clock control (RCC)

7.1 Reset

There are three types of reset, defined as system reset, power reset and RTC domain reset.

7.1.1 Power reset

A power reset is generated when one of the following events occurs:

1. Power-on reset (POR) or Brown-out reset (BOR).
2. when exiting from Standby mode.
3. when exiting from Shutdown mode.

A Brown-out reset, including power-on or power-down reset (POR/PDR), sets all registers to their reset values except the RTC domain.

When exiting Standby mode, all registers in the V_{CORE} domain are set to their reset value. Registers outside the V_{CORE} domain (RTC, WKUP, IWDG, and Standby/Shutdown modes control) are not impacted.

When exiting Shutdown mode, a Brown-out reset is generated, resetting all registers except those in the RTC domain.

7.1.2 System reset

A system reset sets all registers to their reset values except the reset flags in the clock control/status register (RCC_CSR) and the registers in the RTC domain.

A system reset is generated when one of the following events occurs:

1. A low level on the NRST pin (external reset)
2. Window watchdog event (WWDG reset)
3. Independent watchdog event (IWDG reset)
4. A software reset (SW reset) (see [Software reset](#))
5. Low-power mode security reset (see [Low-power mode security reset](#))
6. Option byte loader reset (see [Option byte loader reset](#))
7. A Brown-out reset

The reset source can be identified by checking the reset flags in the Control/Status register, RCC_CSR (see [Section 7.4.28: Control/status register \(RCC_CSR\)](#)).

NRST pin (external reset)

Through specific option bits (NRST_MODE), the NRST pin is configurable for operating as:

- Reset input/output (default at device delivery)

Any valid reset signal on the pin is propagated to device internal logic and all internal reset sources are externally driven through a pulse generator to this pin. The GPIO functionality (PG10) is not available. The pulse generator guarantees a minimum reset pulse duration of 20 µs for each internal reset source to be output on the NRST pin. An internal reset holder option can be used, if enabled in the option bytes, to ensure that the pin is pulled low until its voltage meets VIL threshold. This function guarantees the detection of internal reset sources by external components when the line faces a

significant capacitive load. In case of an internal reset, the internal pull-up R_{PU} is deactivated in order to save the power consumption through the pull-up resistor. This mode is always active (independently of the option bytes setting) during each device power-on-reset (until option bytes are loaded): power on the device or wake up from Shutdown mode.

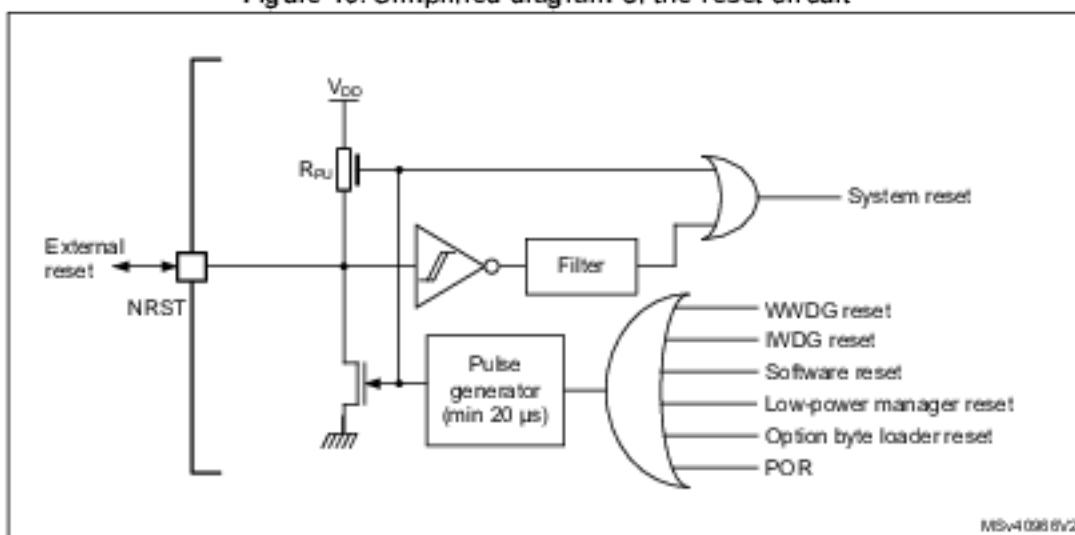
- **Reset input**

In this mode, any valid reset signal on the NRST pin is propagated to device internal logic, but resets generated internally by the device are not visible on the pin. In this configuration, GPIO functionality (PG10) is not available.

- **GPIO**

In this mode, the pin can be used as PG10 standard GPIO. The reset function of the pin is not available. Reset is only possible from device internal reset sources and it is not propagated to the pin.

Figure 16. Simplified diagram of the reset circuit



Software reset

The SYSRESETREQ bit in Cortex®-M4 with FPU Application Interrupt and Reset Control Register must be set to force a software reset on the device (refer to the STM32F3xx/F4xx/L4xx Cortex®-M4 programming manual (PM0214)).

Low-power mode security reset

To prevent that critical applications mistakenly enter a low-power mode, two low-power mode security resets are available. If enabled in option bytes, the resets are generated in the following conditions:

1. Entering Standby mode: this type of reset is enabled by resetting nRST_STDBY bit in User option Bytes. In this case, whenever a Standby mode entry sequence is successfully executed, the device is reset instead of entering Standby mode.
2. Entering Stop mode: this type of reset is enabled by resetting nRST_STOP bit in User option bytes. In this case, whenever a Stop mode entry sequence is successfully executed, the device is reset instead of entering Stop mode.
3. Entering Shutdown mode: this type of reset is enabled by resetting nRST_SHDW bit in User option bytes. In this case, whenever a Shutdown mode entry sequence is successfully executed, the device is reset instead of entering Shutdown mode.

For further information on the User Option Bytes, refer to [Section 3.4.1: Option bytes description](#).

Option byte loader reset

The option byte loader reset is generated when the OBL_LAUNCH bit (bit 27) is set in the FLASH_CR register. This bit is used to launch the option byte loading by software.

7.1.3 RTC domain reset

The RTC domain has two specific resets.

A RTC domain reset is generated when one of the following events occurs:

1. Software reset, triggered by setting the BDRST bit in the [RTC domain control register \(RCC_BDCR\)](#).
2. V_{DD} or V_{BAT} power on, if both supplies have previously been powered off.

A RTC domain reset only affects the LSE oscillator, the RTC, the Backup registers and the RCC RTC domain control register.

7.2 Clocks

Three different clock sources can be used to drive the system clock (SYSCLK):

- HSI16 (high speed internal)16 MHz RC oscillator clock
- HSE oscillator clock, from 4 to 48 MHz
- PLL clock

The HSI16 is used as system clock source after startup from Reset.

The devices have the following additional clock sources:

- 32 kHz low speed internal RC (LSI RC) which drives the independent watchdog and optionally the RTC used for Auto-wakeup from Stop and Standby modes.
- 32.768 kHz low speed external crystal (LSE crystal) which optionally drives the real-time clock (RTCCLK).
- RC 48 MHz internal clock sources (HSI48) to potentially drive the USB FS and the RNG.

Each clock source can be switched on or off independently when it is not used, to optimize power consumption.

Several prescalers can be used to configure the AHB frequency, the APB1 and APB2 domains. The maximum frequency of the AHB, the APB1 and the APB2 domains is 170 MHz.

All the peripheral clocks are derived from their bus clock (HCLK, PCLK1 or PCLK2) except:

- The 48 MHz clock, used for USB device FS, and RNG. This clock is derived (selected by software) from one of the four following sources:
 - PLL "Q" Clock
 - HSI48 internal oscillator

When available, the HSI48 48 MHz clock can be coupled to the clock recovery system allowing adequate clock connection for the USB OTG FS (Crystal less solution).

- The ADCs clock which is derived (selected by software) from one of the following sources:
 - System clock (SYSCLK)
 - PLL "P" clock
- The U(S)ARTs clocks which are derived (selected by software) from one of the four following sources:
 - System clock (SYSCLK)
 - HSI16 clock
 - LSE clock
 - APB1 or APB2 clock (PCLK1 or PCLK2 depending on which APB is mapped the U(S)ART)

The wakeup from Stop mode is supported only when the clock is HSI16 or LSE.

- The I²Cs clocks which are derived (selected by software) from one of the three following sources:
 - System clock (SYSCLK)
 - HSI16 clock
 - APB1 clock (PCLK1)
- The wake up from Stop mode is supported only when the clock is HSI16.
- The SAI1 clock which is derived (selected by software) from one of the following sources:
 - an external clock mapped on I2S_CKIN
 - System clock
 - PLL "Q" clock
 - HSI16 clock
- The QUADSPI kernel clock which is derived (selected by software) from one of the following sources:
 - System clock,
 - PLL "Q" clock
 - HSI16 clock
- The low-power timer (LPTIM1) clock which is derived (selected by software) from one of the five following sources:
 - LSI clock
 - LSE clock
 - HSI16 clock
 - APB1 clock (PCLK1)
 - External clock mapped on LPTIMx_IN1

The functionality in Stop mode (including wakeup) is supported only when the clock is

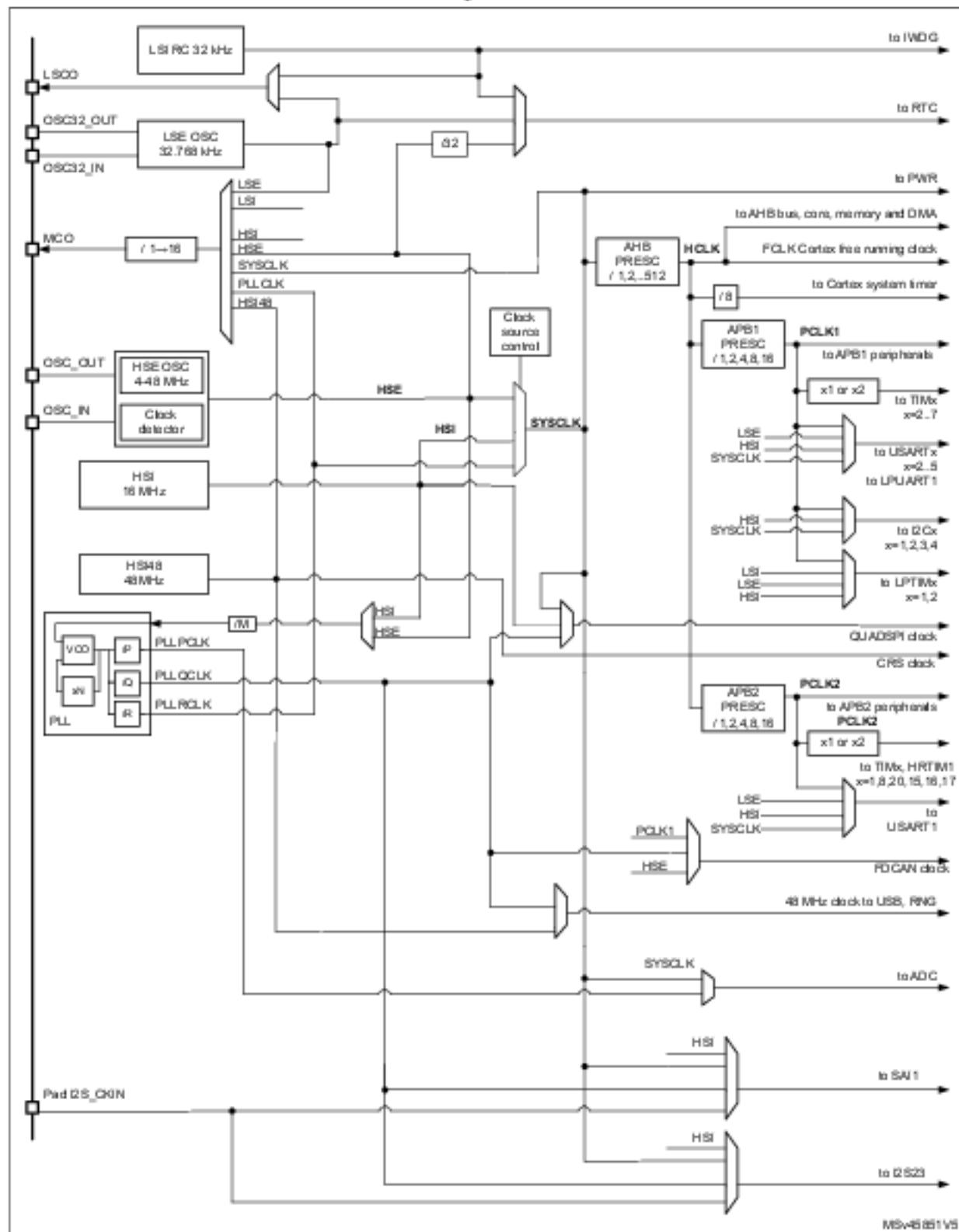
LSI or LSE, or in external clock mode.

- The RTC clock which is derived (selected by software) from one of the three following sources:
 - LSE clock
 - LSI clock
 - HSE clock divided by 32
- The functionality in Stop mode (including wakeup) is supported only when the clock is LSI or LSE.
- The IWDG clock which is always the LSI clock.
- The UCPD1 clock, which is derived from HSI16 clock.
- The FDCAN1 clock, which is derived (selected by software) from one of the two following sources:
 - HSE clock
 - PLL "Q" clock
 - PCLK clock

The RCC feeds the Cortex[®] System Timer (SysTick) external clock with the AHB clock (HCLK) divided by 8. The SysTick can work either with this clock or directly with the Cortex[®] clock (HCLK), configurable in the SysTick Control and Status Register.

FCLK acts as Cortex[®]-M4 with FPU free-running clock. For more details refer to the Cortex[®]-M4 programming manual (PM0214).

Figure 17. Clock tree



1. For full details about the internal and external clock source characteristics, please refer to the "Electrical" section of the device datasheet.

"characteristics" section in your device datasheet.

2. The ADC clock can be derived from the AHB clock of the ADC bus interface, divided by a programmable factor (1, 2 or 4). When the programmable factor is '1', the AHB prescaler must be equal to '1'.

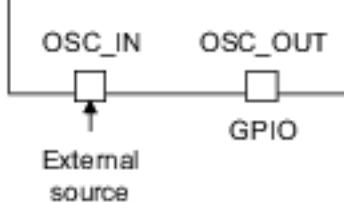
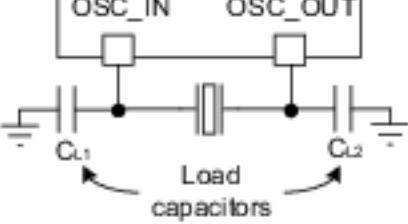
7.2.1 HSE clock

The high speed external clock signal (HSE) can be generated from two possible clock sources:

- HSE external crystal/ceramic resonator
- HSE user external clock

The resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and startup stabilization time. The loading capacitance values must be adjusted according to the selected oscillator.

Figure 18. HSE/ LSE clock sources

| Clock source | Hardware configuration |
|----------------------------|---|
| External clock |  External source |
| Crystal/Ceramic resonators |  Load capacitors |

External crystal/ceramic resonator (HSE crystal)

The 4 to 48 MHz external oscillator has the advantage of producing a very accurate rate on the main clock.

The associated hardware configuration is shown in [Figure 18](#). Refer to the electrical characteristics section of the datasheet for more details.

The HSERDY flag in the [Clock control register \(RCC_CR\)](#) indicates if the HSE oscillator is stable or not. At startup, the clock is not released until this bit is set by hardware. An interrupt can be generated if enabled in the [Clock interrupt enable register \(RCC_CIER\)](#).

The HSE Crystal can be switched on and off using the HSEON bit in the [Clock control register \(RCC_CR\)](#).

External source (HSE bypass)

In this mode, an external clock source must be provided. It can have a frequency of up to 48 MHz. You select this mode by setting the HSEBYP and HSEON bits in the [Clock control register \(RCC_CR\)](#). The external clock signal (square, sinus or triangle) with ~40-60 % duty cycle depending on the frequency (refer to the datasheet) has to drive the OSC_IN pin while the OSC_OUT pin can be used as GPIO. See [Figure 18](#).

7.2.2 HSI16 clock

The HSI16 clock signal is generated from an internal 16 MHz RC Oscillator.

The HSI16 RC oscillator has the advantage of providing a clock source at low cost (no external components). It also has a faster startup time than the HSE crystal oscillator however, even with calibration the frequency is less accurate than an external crystal oscillator or ceramic resonator.

The HSI16 clock can be selected as system clock after wake up from Stop modes (Stop 0, Stop 1). Refer to [Section 7.3: Low-power modes](#). It can also be used as a backup clock source (auxiliary clock) if the HSE crystal oscillator fails. Refer to [Section 7.2.9: Clock security system \(CSS\)](#).

Calibration

RC oscillator frequencies can vary from one chip to another due to manufacturing process variations, this is why each device is factory calibrated by ST for 1 % accuracy at $T_A=25^\circ\text{C}$.

After reset, the factory calibration value is loaded in the HSICAL[7:0] bits in the [Internal clock sources calibration register \(RCC_ICSCR\)](#).

If the application is subject to voltage or temperature variations this may affect the RC oscillator speed. You can trim the HSI16 frequency in the application using the HSITRIM[6:0] in the [Internal clock sources calibration register \(RCC_ICSCR\)](#).

For more details on how to measure the HSI16 frequency variation, refer to [Section 7.2.16: Internal/external clock measurement with TIM5/TIM15/TIM16/TIM17](#).

The HSIRDY flag in the [Clock control register \(RCC_CR\)](#) indicates if the HSI16 oscillator is stable or not. At startup, the HSI16 output clock is not released until this bit is set by hardware.

The HSI16 oscillator can be switched on and off using the HSION bit in the [Clock control register \(RCC_CR\)](#).

The HSI16 signal can also be used as a backup source (Auxiliary clock) if the HSE crystal oscillator fails. Refer to [Section 7.2.9: Clock security system \(CSS\) on page 283](#).

7.2.3 HSI48 clock

The HSI48 clock signal is generated from an internal 48 MHz RC oscillator and can be used directly for USB and for random number generator (RNG).

The internal 48 MHz RC oscillator is mainly dedicated to provide a high precision clock to the USB peripheral by means of a special Clock Recovery System (CRS) circuitry. The CRS can use the LSE or an external signal to automatically and quickly adjust the oscillator frequency on-fly. It is disabled as soon as the system enters Stop or Standby mode. When the CRS is not used, the HSI48 RC oscillator runs on its default frequency which is subject to manufacturing process variations.

The HSI48RDY flag in the Clock recovery RC register (RCC_CRRRCR) indicates whether the HSI48 RC oscillator is stable or not. At startup, the HSI48 RC oscillator output clock is not released until this bit is set by hardware.

The HSI48 can be switched on and off using the HSI48ON bit in the Clock recovery RC register (RCC_CRRRCR).

7.2.4 PLL

The internal PLL can be used to multiply the HSI16 or HSE output clock frequency. The PLL input frequency must be within the range defined in the device datasheet. The selected clock source is divided by a programmable factor PLLM from 1 to 8 to provide a clock frequency in the requested input range. Refer to [Figure 17: Clock tree and PLL configuration register \(RCC_PLLCFGR\)](#).

The PLL configuration (selection of the input clock and multiplication factor) must be done before enabling the PLL. Once the PLL is enabled, these parameters cannot be changed.

To modify the PLL configuration, proceed as follows:

1. Disable the PLL by setting PLLON to 0 in [Clock control register \(RCC_CR\)](#).
2. Wait until PLLRDY is cleared. The PLL is now fully stopped.
3. Change the desired parameter.
4. Enable the PLL again by setting PLLON to 1.
5. Enable the desired PLL outputs by configuring PLLPEN, PLLQEN, PLLREN in [PLL configuration register \(RCC_PLLCFGR\)](#).

An interrupt can be generated when the PLL is ready, if enabled in the [Clock interrupt enable register \(RCC_CIER\)](#).

The PLL output frequency must not exceed 170 MHz.

The enable bit of each PLL output clock (PLLPEN, PLLQEN, PLLREN) can be modified at any time without stopping the corresponding PLL. PLLREN cannot be cleared if PLLCLK is used as system clock.

7.2.5 LSE clock

The LSE crystal is a 32.768 kHz Low Speed External crystal or ceramic resonator. It has the advantage of providing a low-power but highly accurate clock source to the real-time clock peripheral (RTC) for clock/calendar or other timing functions.

The LSE crystal is switched on and off using the LSEON bit in [RTC domain control register \(RCC_BDCR\)](#). The crystal oscillator driving strength can be changed at runtime using the LSEDRV[1:0] bits in the [RTC domain control register \(RCC_BDCR\)](#) to obtain the best compromise between robustness and short start-up time on one side and low-power-consumption on the other side. The LSE drive can be decreased to the lower drive capability (LSEDRV=00) when the LSE is ON. However, once LSEDRV is selected, the drive capability can not be increased if LSEON=1.

The LSERDY flag in the [RTC domain control register \(RCC_BDCR\)](#) indicates whether the LSE crystal is stable or not. At startup, the LSE crystal output clock signal is not released until this bit is set by hardware. An interrupt can be generated if enabled in the [Clock interruptenable register \(RCC_CIER\)](#).

External source (LSE bypass)

In this mode, an external clock source must be provided. It can have a frequency of up to 1 MHz. You select this mode by setting the LSEBYP and LSEON bits in the [AHB1 peripheral clocks enable in Sleep and Stop modes register \(RCC_AHB1SMENR\)](#). The external clock signal (square, sinus or triangle) with ~50 % duty cycle has to drive the OSC32_IN pin while the OSC32_OUT pin can be used as GPIO. See [Figure 18](#).

7.2.6 LSI clock

The LSI RC acts as a low-power clock source that can be kept running in Stop and Standby mode for the independent watchdog (IWDG) and RTC. The clock frequency is 32 kHz. For more details, refer to the electrical characteristics section of the datasheets.

The LSI RC can be switched on and off using the LSION bit in the [Control/status register \(RCC_CSR\)](#).

The LSIRDY flag in the [Control/status register \(RCC_CSR\)](#) indicates if the LSI oscillator is stable or not. At startup, the clock is not released until this bit is set by hardware. An interrupt can be generated if enabled in the [Clock interruptenable register \(RCC_CIER\)](#).

7.2.7 System clock (SYSCLK) selection

Four different clock sources can be used to drive the system clock (SYSCLK):

- HSI16 oscillator
- HSE oscillator
- PLL

The system clock maximum frequency is 170 MHz. After a system reset, the HSI16 oscillator is selected as system clock. When a clock source is used directly or through the PLL as a system clock, it is not possible to stop it.

A switch from one clock source to another occurs only if the target clock source is ready (clock stable after startup delay or PLL locked). If a clock source which is not yet ready is selected, the switch occurs when the clock source becomes ready. Status bits in the [Internal clock sources calibration register \(RCC_ICSCR\)](#) indicate which clock(s) is (are) ready and which clock is currently used as a system clock.

To switch from low speed to high speed or from high speed to low speed system clock, it is recommended to use a transition state with medium speed clock, for at least 1 μ s.

Clock source switching conditions:

- Switching from HSE or HSI16 to PLL with AHB frequency (HCLK) higher than 80 MHz
- Switching from PLL with HCLK higher than 80 MHz to HSE or HSI16

Transition state:

- Set the AHB prescaler HPRE[3:0] bits to divide the system frequency by 2
- Switch system clock to PLL
- Wait for at least 1 μ s and then reconfigure AHB prescaler bits to the needed HCLK frequency

7.2.8 Clock source frequency versus voltage scaling

The following table gives the different clock source frequencies depending on the product voltage range.

Table 50. Clock source frequency

| Product voltage range | Clock frequency | | |
|-----------------------|-----------------|--------|---------|
| | HSI16 | HSE | PLL |
| Range 1 Boost mode | 16 MHz | 48 MHz | 170 MHz |
| Range 1 normal mode | 16 MHz | 48 MHz | 150 MHz |
| Range 2 | 16 MHz | 26 MHz | 26 MHz |

7.2.9 Clock security system (CSS)

Clock Security System can be activated by software. In this case, the clock detector is enabled after the HSE oscillator startup delay, and disabled when this oscillator is stopped.

If a failure is detected on the HSE clock, the HSE oscillator is automatically disabled, a clock failure event is sent to the break input of the advanced-control timers (TIM1/TIM8/TIM20 and TIM15/16/17) and to the hrtim_sys_flt, and an interrupt is generated to inform the software about the failure (Clock Security System Interrupt CSSI), allowing the MCU to perform rescue operations. The CSSI is linked to the Cortex®-M4 with FPU NMI (Non-Maskable Interrupt) exception vector.

Note: Once the CSS is enabled and if the HSE clock fails, the CSS interrupt occurs and a NMI is automatically generated. The NMI is executed indefinitely unless the CSS interrupt pending bit is cleared. As a consequence, in the NMI ISR user must clear the CSS interrupt by setting the CSSC bit in the [Clock interrupt clear register \(RCC_CICR\)](#).

If the HSE oscillator is used directly or indirectly as the system clock (indirectly means: it is used as PLL input clock, and the PLL clock is used as system clock), a detected failure causes a switch of the system clock to the HSI16 oscillator, and the disabling of the HSE oscillator. If the HSE clock (divided or not) is the clock entry of the PLL used as system clock when the failure occurs, the PLL is disabled too.

7.2.10 Clock security system on LSE

A Clock Security System on LSE can be activated by software writing the LSECSSON bit in the [Control/status register \(RCC_CSR\)](#). This bit can be disabled only by a hardware reset or RTC software reset, or after a failure detection on LSE. LSECSSON must be written after

LSE and LSI are enabled (LSEON and LSION enabled) and ready (LSERDY and LSIRDY set by hardware), and after the RTC clock has been selected by RTCSEL.

The CSS on LSE is working in all modes except VBAT. It is working also under system reset (excluding power on reset). If a failure is detected on the external 32 kHz oscillator, the LSE clock is no longer supplied to the RTC but no hardware action is made to the registers. If the HSI16 was in PLL-mode, this mode is disabled.

In Standby mode a wakeup is generated. In other modes an interrupt can be sent to wakeup the software (see [Clock interrupt enable register \(RCC_CIER\)](#), [Clock interrupt flag register \(RCC_CIFR\)](#), [Clock interrupt clear register \(RCC_CICR\)](#)).

The software MUST then disable the LSECSSON bit, stop the defective 32 kHz oscillator (disabling LSEON), and change the RTC clocksource (no clock or LSI or HSE, with RTCSEL), or take any required action to secure the application.

The frequency of LSE oscillator have to be higher than 30 kHz to avoid false positive CSS detection.

7.2.11 ADC clock

The ADC clock is derived from the system clock, or from the PLL "P" output. It can be divided by the following prescalers values: 1,2,4,8,8,10,12,16,32,64,128 or 256 by configuring the ADCx_CCR register. It is asynchronous to the AHB clock. Alternatively, the ADC clock can be derived from the AHB clock of the ADC bus interface, divided by a programmable factor (1, 2 or 4). This programmable factor is configured using the CKMODE bit fields in the ADCx_CCR.

If the programmed factor is '1', the AHB prescaler must be set to '1'.

Refer to the device datasheet for the maximum ADC clock.

7.2.12 RTC clock

The RTCCLK clock source can be either the HSE/32, LSE or LSI clock. It is selected by programming the RTCSEL[1:0] bits in the [RTC domain control register \(RCC_BDCR\)](#). This selection cannot be modified without resetting the RTC domain. The system must always be configured so as to get a PCLK frequency greater than or equal to the RTCCLK frequency for a proper operation of the RTC.

The LSE clock is in the RTC domain, whereas the HSE and LSI clocks are not. Consequently:

- If LSE is selected as RTC clock:
 - The RTC continues to work even if the V_{DD} supply is switched off, provided the V_{BAT} supply is maintained.
- If LSI is selected as the RTC clock:
 - The RTC state is not guaranteed if the V_{DD} supply is powered off.
- If the HSE clock divided by a prescaler is used as the RTC clock:
 - The RTC state is not guaranteed if the V_{DD} supply is powered off or if the internal voltage regulator is powered off (removing power from the V_{CORE} domain).

When the RTC clock is LSE or LSI, the RTC remains clocked and functional under system reset.

7.2.13 Timer clock

The timer clock frequencies are automatically defined by hardware. There are two cases:

1. If the APB prescaler equals 1, the timer clock frequencies are set to the same frequency as that of the APB domain.
2. Otherwise, they are set to twice ($\times 2$) the frequency of the APB domain.

7.2.14 Watchdog clock

If the Independent watchdog (IWDG) is started by either hardware option or software access, the LSI oscillator is forced ON and cannot be disabled. After the LSI oscillator temporization, the clock is provided to the IWDG.

7.2.15 Clock-out capability

- MCO

The microcontroller clock output (MCO) capability allows the clock to be output onto the external MCO pin. One of eight clock signals can be selected as the MCO clock.

- LSI
- LSE
- SYSCLK
- HSI16
- HSI48
- HSE
- PLLCLK

The selection is controlled by the MCOSEL[3:0] bits of the [Clock configuration register \(RCC_CFGR\)](#). The selected clock can be divided with the MCOPRE[2:0] field of the [Clock configuration register \(RCC_CFGR\)](#).

- LSCO

Another output (LSCO) allows a low speed clock to be output onto the external LSCO pin:

- LSI
- LSE

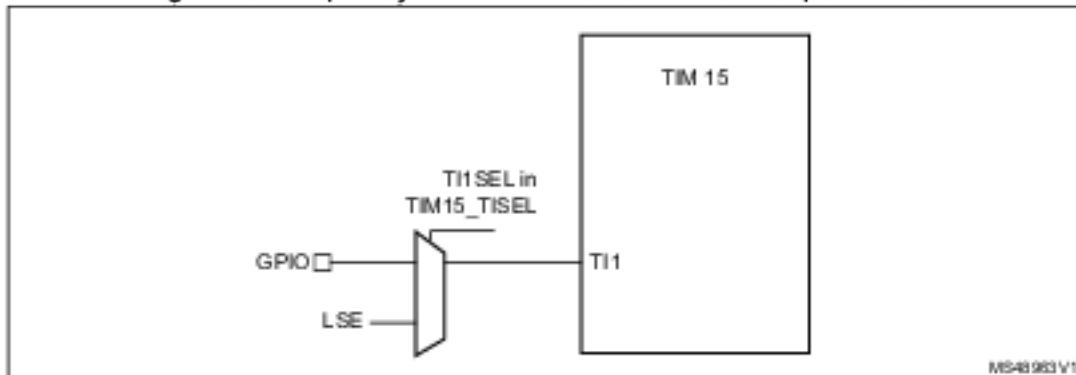
This output remains available in Stop (Stop 0 and Stop 1) and Standby modes. The selection is controlled by the LSCOSEL, and enabled with the LSCOEN in the [RTC domain control register \(RCC_BDCR\)](#).

The MCO clock output requires the corresponding alternate function selected on the MCO pin, the LSCO pin should be left in default POR state.

7.2.16 Internal/external clock measurement with TIM5/TIM15/TIM16/TIM17

It is possible to indirectly measure the frequency of all on-board clock sources by mean of the TIM5, TIM15, TIM16 or TIM17 channel 1 input capture, as represented on [Figure 19](#), [Figure 20](#), [Figure 21](#) and [Figure 22](#).

Figure 19. Frequency measurement with TIM15 in capture mode

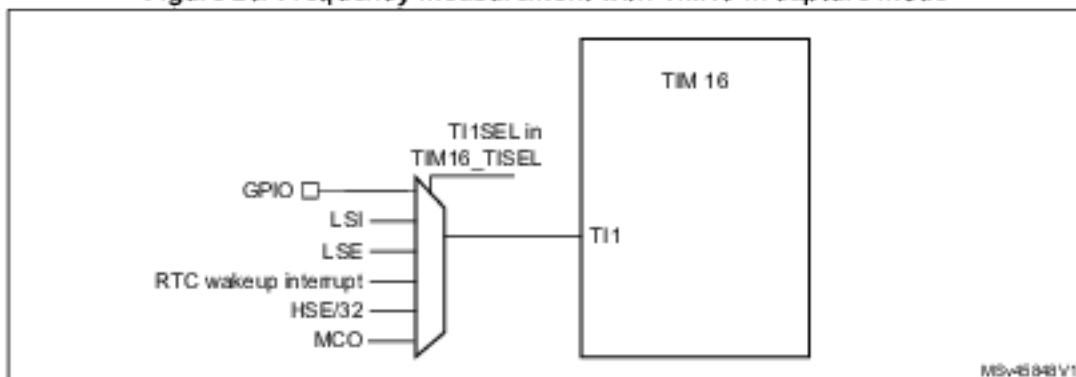


MSA993V1

The input capture channel of the Timer 15 can be a GPIO line or an internal clock of the MCU. The possibilities are the following ones:

- TIM15 Channel1 is connected to the GPIO. Refer to the alternate function mapping in the device datasheets.
- TIM15 Channel1 is connected to the LSE.

Figure 20. Frequency measurement with TIM16 in capture mode

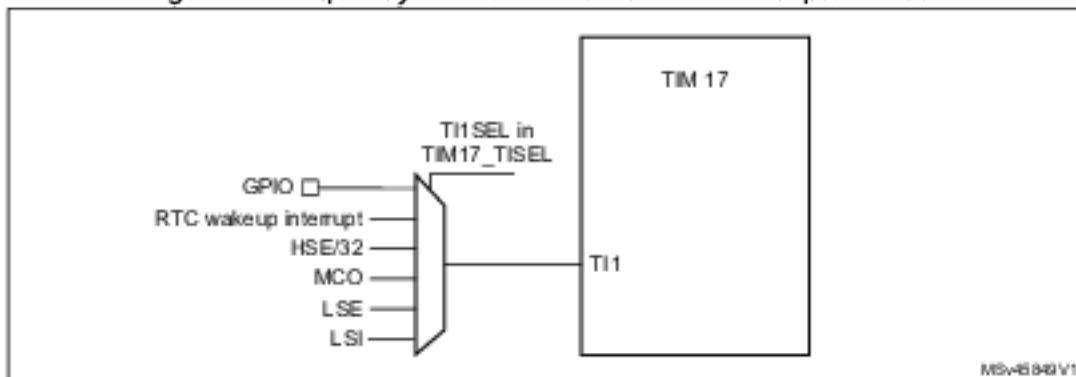


MSA993V1

The input capture channel of the Timer 16 can be a GPIO line or an internal clock of the MCU.

The possibilities are the following ones:

- TIM16 Channel1 is connected to the GPIO. Refer to the alternate function mapping in the device datasheets.
- TIM16 Channel1 is connected to the LSI clock.
- TIM16 Channel1 is connected to the LSE clock.
- TIM16 Channel1 is connected to the RTC wakeup interrupt signal. In this case the RTC interrupt should be enabled.
- TIM16 Channel1 is connected to the HSE/32 clock.
- TIM16 Channel1 is connected to the MCO.

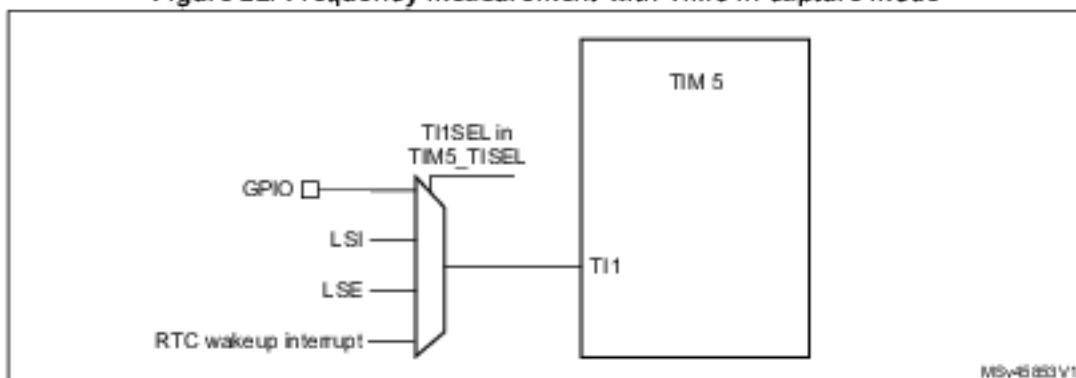
Figure 21. Frequency measurement with TIM17 in capture mode

MSW6849V1

The input capture channel of the Timer 17 can be a GPIO line or an internal clock of the MCU.

The possibilities are the following ones:

- TIM17 Channel1 is connected to the GPIO. Refer to the alternate function mapping in the device datasheets.
- TIM17 Channel1 is connected to the RTC wakeup interrupt. In this case the RTC interrupt should be enabled.
- TIM17 Channel1 is connected to the HSE/32 Clock.
- TIM17 Channel1 is connected to the microcontroller clock output (MCO), this selection is controlled by the **MCOSEL[3:0]** bits of the Clock configuration register (**RCC_C FGR**).
- TIM17 Channel1 is connected to the LSE Clock.
- TIM17 Channel1 is connected to the LSI Clock.

Figure 22. Frequency measurement with TIM5 in capture mode

MSW6849V1

The input capture channel of the Timer 5 can be a GPIO line or an internal clock of the MCU.

The possibilities are the following ones:

- TIM5 Channel1 is connected to the GPIO. Refer to the alternate function mapping in the device datasheets.
- TIM5 Channel1 is connected to the LSI Clock.
- TIM5 Channel1 is connected to the LSE Clock.
- TIM5 Channel1 is connected to the RTC wakeup interrupt signal. In this case the RTC interrupt should be enabled.

Calibration of the HSI16

For TIM15 and TIM16, the primary purpose of connecting the LSE to the channel 1 input capture is to be able to precisely measure the HSI16 system clocks (for this, the HSI16 should be used as the system clock source). The number of HSI16 clock counts between consecutive edges of the LSE signal provides a measure of the internal clock period. Taking advantage of the high precision of LSE crystals (typically a few tens of ppm's), it is possible to determine the internal clock frequency with the same resolution, and trim the source to compensate for manufacturing, process, temperature and/or voltage related frequency deviations.

The HSI16 oscillator has dedicated user-accessible calibration bits for this purpose.

The basic concept consists in providing a relative measurement (e.g. the HSI16/LSE ratio): the precision is therefore closely related to the ratio between the two clock sources. The higher the ratio is, the better the measurement is.

If LSE is not available, HSE/32 is the better option in order to reach the most precise calibration possible.

Calibration of the LSI

The calibration of the LSI follows the same pattern that for the HSI16, but changing the reference clock. It is necessary to connect LSI clock to the channel 1 input capture of the TIM16. Then define the HSE as system clock source, the number of his clock counts between consecutive edges of the LSI signal provides a measure of the internal low speed clock period.

The basic concept consists in providing a relative measurement (e.g. the HSE/LSI ratio): the precision is therefore closely related to the ratio between the two clock sources. The higher the ratio is, the better the measurement is.

7.2.17 Peripheral clock enable register (RCC_AHBxENR, RCC_APBxENRy)

Each peripheral clock can be enabled by the xxxxEN bit of the RCC_AHBxENR, RCC_APBxENRy registers.

When the peripheral clock is not active, the peripheral registers read or write accesses are not supported.

The enable bit has a synchronization mechanism to create a glitch free clock for the peripheral. After the enable bit is set, there is a 2 clock cycles delay before the clock becomes active.

Caution: Just after enabling the clock for a peripheral, software must wait for a delay before accessing the peripheral registers.

7.3 Low-power modes

- AHB and APB peripheral clocks, including DMA clock, can be disabled by software.
- Sleep and Low Power Sleep modes stops the CPU clock. The memory interface clocks (Flash and SRAM1, SRAM2 and CCM SRAM interfaces) can be stopped by software

during sleep mode. The AHB to APB bridge clocks are disabled by hardware during Sleep mode when all the clocks of the peripherals connected to them are disabled.

- Stop modes (Stop 0 and Stop 1) stops all the clocks in the V_{CORE} domain and disables the PLL, the HSI16, and the HSE oscillators.

All U(S)ARTs, LPUARTs and I²Cs have the capability to enable the HSI16 oscillator even when the MCU is in Stop mode (if HSI16 is selected as the clock source for that peripheral).

All U(S)ARTs and LPUARTs can also be driven by the LSE oscillator when the system is in Stop mode (if LSE is selected as clock source for that peripheral) and the LSE oscillator is enabled (LSEON). In that case the LSE remains always ON in Stop mode (they do not have the capability to turn on the LSE oscillator).

- Standby and Shutdown modes stops all the clocks in the V_{CORE} domain and disables the PLL, the HSI16, and the HSE oscillators.

The CPU's deepsleep mode can be overridden for debugging by setting the DBG_STOP or DBG_STANDBY bits in the DBGMCU_CR register.

When leaving the Stop modes (Stop 0, Stop 1 or standby), the system clock is HSI16.

If a Flash memory programming operation is on going, Stop, Standby and Shutdown modes entry is delayed until the Flash memory interface access is finished. If an access to the APB domain is ongoing, Stop, Standby and Shutdown modes entry is delayed until the APB access is finished.

7.4 RCC registers

7.4.1 Clock control register (RCC_CR)

Address offset: 0x00

Reset value: 0x0000 0063

HSEBYP is not affected by reset.

Access: no wait state, word, half-word and byte access

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|---------|-----------|--------|------|------|------|------|-------|--------|--------|-------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | Res. | Res. | Res. | Res. | Res. | PLL RDY | PLLO N | Res. | Res. | Res. | Res. | CSSON | HSEBYP | HSERDY | HSEON |
| | | | | | | r | rw | | | | | is | rw | r | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | Res. | Res. | Res. | HSI RDY | HSI KERON | HSION | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| | | | | | r | rw | rw | | | | | | | | |

Bits 31:26 Reserved, must be kept at reset value.

Bit 25 **PLL RDY**: Main PLL clock ready flag

Set by hardware to indicate that the main PLL is locked.

0: PLL unlocked

1: PLL locked

Bit 24 **PLLO N**: Main PLL enable

Set and cleared by software to enable the main PLL.

Cleared by hardware when entering Stop, Standby or Shutdown mode. This bit cannot be reset if the PLL clock is used as the system clock.

0: PLLOFF

1: PLL ON

Bits 23:20 Reserved, must be kept at reset value.

Bit 19 **CSSON**: Clock security system enable

Set by software to enable the clock security system. When CSSON is set, the clock detector is enabled by hardware when the HSE oscillator is ready, and disabled by hardware if a HSE clock failure is detected. This bit is set only and is cleared by reset.

0: Clock security system OFF (clock detector OFF)

1: Clock security system ON (clock detector ON if the HSE oscillator is stable, OFF if not).

Bit 18 **HSEBYP**: HSE crystal oscillator bypass

Set and cleared by software to bypass the oscillator with an external clock. The external clock must be enabled with the HSEON bit set, to be used by the device. The HSEBYP bit can be written only if the HSE oscillator is disabled.

0: HSE crystal oscillator not bypassed

1: HSE crystal oscillator bypassed with external clock

Bit 17 **HSERDY**: HSE clock ready flag

Set by hardware to indicate that the HSE oscillator is stable.

0: HSE oscillator not ready

1: HSE oscillator ready

Note: Once the HSEON bit is cleared, HSERDY goes low after 6 HSE clock cycles.

Bit 16 **HSEON**: HSE clock enable

Set and cleared by software.

Cleared by hardware to stop the HSE oscillator when entering Stop, Standby or Shutdown mode. This bit cannot be reset if the HSE oscillator is used directly or indirectly as the system clock.

0: HSE oscillator OFF

1: HSE oscillator ON

Bits 15:11 Reserved, must be kept at reset value.

Bit 10 **HSIRDY**: HSI16 clock ready flag

Set by hardware to indicate that HSI16 oscillator is stable. This bit is set only when HSI16 is enabled by software by setting HSION.

0: HSI16 oscillator not ready

1: HSI16 oscillator ready

Note: Once the HSION bit is cleared, HSIRDY goes low after 6 HSI16 clock cycles.

Bit 9 **HSI16ERON**: HSI16 always enable for peripheral kernels.

Set and cleared by software to force HSI16 ON even in Stop modes. The HSI16 can only feed USARTs and PCs peripherals configured with HSI16 as kernel clock. Keeping the HSI16 ON in Stop mode allows to avoid slowing down the communication speed because of the HSI16 startup time. This bit has no effect on HSION value.

0: No effect on HSI16 oscillator.

1: HSI16 oscillator is forced ON even in Stop mode.

Bit 8 **HSION**: HSI16 clock enable

Set and cleared by software.

Cleared by hardware to stop the HSI16 oscillator when entering Stop, Standby or Shutdown mode.

Set by hardware to force the HSI16 oscillator ON when STOPWUCK=1 or HSIASFS = 1 when leaving Stop modes, or in case of failure of the HSE crystal oscillator.

This bit is set by hardware if the HSI16 is used directly or indirectly as system clock.

0: HSI16 oscillator OFF

1: HSI16 oscillator ON

Bits 7:0 Reserved, must be kept at reset value.

7.4.2 Internal clock sources calibration register (RCC_ICSCR)

Address offset: 0x04

Reset value: 0x40XX 00XX

where X is factory-programmed.

Access: no wait state, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | |
|------|--------------|------|------|------|------|------|------|------|-------------|------|------|------|------|------|------|--|
| Res. | HSITRIM[6:0] | | | | | | | | HSICAL[7:0] | | | | | | | |
| | rw | rw | rw | rw | rw | rw | rw | r | r | r | r | r | r | r | r | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | |
| | | | | | | | | | | | | | | | | |

Bit 31 Reserved, must be kept at reset value.

Bits 30:24 HSI16TRIM[6:0]: HSI16 clock trimming

These bits provide an additional user-programmable trimming value that is added to the HSICAL[7:0] bits. It can be programmed to adjust to variations in voltage and temperature that influence the frequency of the HSI16.

The default value is 64, which, when added to the HSICAL value, should trim the HSI16 to 16 MHz ± 1 %.

Bits 23:16 HSICAL[7:0]: HSI16 clock calibration

These bits are initialized at startup with the factory-programmed HSI16 calibration trim value. When HSITRIM is written, HSICAL is updated with the sum of HSITRIM and the factory trim value.

Bits 15:0 Reserved, must be kept at reset value.

7.4.3 Clock configuration register (RCC_CFGR)

Address offset: 0x08

Reset value: 0x0000 0005

Access: 0 ≤ wait state ≤ 2, word, half-word and byte access

1 or 2 wait states inserted only if the access occurs during clock source switch.

From 0 to 15 wait states inserted if the access occurs when the APB or AHB prescalers values update is on going.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-----|-------------|------------|----|-------------|------------|----|----|-----------|-----|-----|-----|----------|---------|-----|-----|
| Res | MCOPRE[2:0] | | | MCOSEL[3:0] | | | | Res | Res | Res | Res | Res | Res | Res | Res |
| | rw | rw | rw | rw | rw | rw | rw | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | B | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res | Res | PPRE2[2:0] | | | PPRE1[2:0] | | | HPRE[3:0] | | | | SWS[1:0] | SW[1:0] | | |
| | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | r | r | rw | rw |

Bit 31 Reserved, must be kept at reset value.

Bits 30:28 MCOPRE[2:0]: Microcontroller clock output prescaler

These bits are set and cleared by software.

It is highly recommended to change this prescaler before MCO output is enabled.

000: MCO is divided by 1

001: MCO is divided by 2

010: MCO is divided by 4

011: MCO is divided by 8

100: MCO is divided by 16

Others: not allowed

Bits 27:24 **MCOSEL[3:0]**: Microcontroller clock output
Set and cleared by software.
0000: MCO output disabled, no clock on MCO
0001: SYSCLK system clock selected
0010: Reserved, must be kept at reset value
0011: HSI16 clock selected
0100: HSE clock selected
0101: Main PLL clock selected
0110: LSI clock selected
0111: LSE clock selected
1000: Internal HS48 clock selected
Others: Reserved

Note: This clock output may have some truncated cycles at startup or during MCO clock source switching.

Bits 23:14 Reserved, must be kept at reset value.

Bits 13:11 **PPRE2[2:0]**: APB2 prescaler

Set and cleared by software to control the division factor of the APB2 clock (PC LK2).
0xx: HCLK not divided
100: HCLK divided by 2
101: HCLK divided by 4
110: HCLK divided by 8
111: HCLK divided by 16

Bits 10:8 **PPRE1[2:0]**: APB1 prescaler

Set and cleared by software to control the division factor of the APB1 clock (PC LK1).
0xx: HCLK not divided
100: HCLK divided by 2
101: HCLK divided by 4
110: HCLK divided by 8
111: HCLK divided by 16

Bits 7:4 **HPRE[3:0]**: AHB prescaler

Set and cleared by software to control the division factor of the AHB clock.

Caution: Depending on the device voltage range, the software has to set correctly these bits to ensure that the system frequency does not exceed the maximum allowed frequency (for more details please refer to [Section 6.1.5: Dynamic voltage scaling management](#)). After a write operation to these bits and before decreasing the voltage range, this register must be read to be sure that the new value has been taken into account.

0xx: SYSCLK not divided
1000: SYSCLK divided by 2
1001: SYSCLK divided by 4
1010: SYSCLK divided by 8
1011: SYSCLK divided by 16
1100: SYSCLK divided by 64
1101: SYSCLK divided by 128
1110: SYSCLK divided by 256
1111: SYSCLK divided by 512

Bits 3:2 **SWS[1:0]**: System clock switch status

Set and cleared by hardware to indicate which clock source is used as system clock.

- 00: Reserved, must be kept at reset value
- 01: HSI16 oscillator used as system clock
- 10: HSE used as system clock
- 11: PLL used as system clock

Bits 1:0 **SW[1:0]**: System clock switch

Set and cleared by software to select system clock source (SYSCLK).

Configured by hardware to force HSI16 oscillator selection when exiting stop and standby modes or in case of failure of the HSE oscillator.

- 00: Reserved, must be kept at reset value
- 01: HSI16 selected as system clock
- 10: HSE selected as system clock
- 11: PLL selected as system clock

7.4.4 PLL configuration register (RCC_PLLCFG R)

Address offset: 0x0C

Reset value: 0x0000 1000

Access: no wait state, word, half-word and byte access

This register is used to configure the PLL clock outputs according to the formulas:

- $f(\text{VCO clock}) = f(\text{PLL clock input}) \times (\text{PLLN} / \text{PLLM})$
- $f(\text{PLL_P}) = f(\text{VCO clock}) / \text{PLLP}$
- $f(\text{PLL_Q}) = f(\text{VCO clock}) / \text{PLLQ}$
- $f(\text{PLL_R}) = f(\text{VCO clock}) / \text{PLLR}$

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|--------------|-----------|----|----|----|-----------|----|------------|------|-----------|----|------------|------|-------------|------|------------|
| PLL DIV[4:0] | | | | | PLLR[1:0] | | PLL REN | Res. | PLLQ[1:0] | | PLL QEN | Res. | Res. | PLLP | PLL PEN |
| rw | rw | rw | rw | rw | rw | rw | rw | | rw | rw | rw | | | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | B | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | PLLN[6:0] | | | | | | PLLM[3:0] | | | | Res. | Res. | PLLSRC[1:0] | | |
| | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | | rw | rw | |

Bits 31:27 **PLLPDIV[4:0]**: Main PLLP division factor

Set and cleared by software to control the PLL "P" frequency. PLL "P" output clock frequency = VCO frequency / PLLPDIV.

00000: PLL "P" clock is controlled by the bit PLLP

00001: Reserved.

00010: PLL "P" clock = VCO / 2

...

11111: PLL "P" clock = VCO / 31

Bits 26:25 **PLL[R][1:0]**: Main PLL division factor for PLL "R" clock (system clock)

Set and cleared by software to control the frequency of the main PLL output clock PLLCLK. This output can be selected as system clock. These bits can be written only if PLL is disabled.

PLL "R" output clock frequency = VCO frequency / PLLR with PLLR = 2, 4, 6, or 8

- 00: PLLR = 2
- 01: PLLR = 4
- 10: PLLR = 6
- 11: PLLR = 8

Caution: The software has to set these bits correctly not to exceed 170 MHz on this domain.

Bit 24 **PLLREN**: PLL "R" clock output enable

Set and reset by software to enable the PLL "R" clock output of the PLL (used as system clock).

This bit cannot be written when PLL "R" clock output of the PLL is used as System Clock. In order to save power, when the PLL "R" clock output of the PLL is not used, the value of PLLREN should be 0.

- 0: PLL "R" clock output disable
- 1: PLL "R" clock output enable

Bit 23 Reserved, must be kept at reset value.

Bits 22:21 **PLLQ[1:0]**: Main PLL division factor for PLL "Q" clock.

Set and cleared by software to control the frequency of the main PLL output clock PLL "Q" clock. This output can be selected for USB, RNG, SAI (48 MHz clock). These bits can be written only if PLL is disabled.

PLL "Q" output clock frequency = VCO frequency / PLLQ with PLLQ = 2, 4, 6, or 8

- 00: PLLQ = 2
- 01: PLLQ = 4
- 10: PLLQ = 6
- 11: PLLQ = 8

Caution: The software has to set these bits correctly not to exceed 170 MHz on this domain.

Bit 20 **PLLQEN**: Main PLL "Q" clock output enable

Set and reset by software to enable the PLL "Q" clock output of the PLL.

In order to save power, when the PLL "Q" clock output of the PLL is not used, the value of PLLQEN should be 0.

- 0: PLL "Q" clock output disable
- 1: PLL "Q" clock output enable

Bits 19:18 Reserved, must be kept at reset value.

Bit 17 **PLLP**: Main PLL division factor for PLL "P" clock.

Set and cleared by software to control the frequency of the main PLL output clock PLL "P" clock. These bits can be written only if PLL is disabled.

When the PLLPDM[4:0] is set to "00000" PLL "P" output clock frequency = VCO frequency / PLLP with PLLP = 7, or 17

- 0: PLLP = 7
- 1: PLLP = 17

Caution: The software has to set these bits correctly not to exceed 170 MHz on this domain.

Bit 16 PLLPEN: Main PLL PLL "P" clock output enable

Set and cleared by software to enable the PLL "P" clock output of the PLL.

In order to save power, when the PLL "P" clock output of the PLL is not used, the value of PLLPEN should be 0.

0: PLL "P" clock output disable

1: PLL "P" clock output enable

Bit 15 Reserved, must be kept at reset value.**Bits 14:8 PLLN[6:0]: Main PLL multiplication factor for VCO**

Set and cleared by software to control the multiplication factor of the VCO. These bits can be written only when the PLL is disabled.

VCO output frequency = VCO input frequency \times PLLN with $8 \leq \text{PLLN} \leq 127$

0000000: PLLN = 0 wrong configuration

0000001: PLLN = 1 wrong configuration

...

0000111: PLLN = 7 wrong configuration

0001000: PLLN = 8

0001001: PLLN = 9

...

1111111: PLLN = 127

Caution: The software has to set correctly these bits to assure that the VCO output frequency is within the range defined in the device datasheet.

Bits 7:4 PLLM[3:0]: Division factor for the main PLL input clock

Set and cleared by software to divide the PLL input clock before the VCO. These bits can be written only when all PLLs are disabled.

VCO input frequency = PLL input clock frequency / PLLM with $1 \leq \text{PLLM} \leq 16$

0000: PLLM = 1

0001: PLLM = 2

0010: PLLM = 3

0011: PLLM = 4

0100: PLLM = 5

0101: PLLM = 6

0110: PLLM = 7

0111: PLLM = 8

1000: PLLSYSM = 9

...

1111: PLLSYSM= 16

Caution: The software has to set these bits correctly to ensure that the VCO input frequency is within the range defined in the device datasheet.

Bits 3:2 Reserved, must be kept at reset value.**Bits 1:0 PLLSRC[1:0]: Main PLL entry clock source**

Set and cleared by software to select PLL clock source. These bits can be written only when PLL is disabled.

In order to save power, when no PLL is used, the value of PLLSRC should be 00.

00: No clock sent to PLL

01: No clock sent to PLL

10: HSI16 clock selected as PLL clock entry

11: HSE clock selected as PLL clock entry

7.4.5 Clock interrupt enable register (RCC_CIER)

Address offset: 0x18

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|----------------|--------------|------|------|------|--------------|--------------|--------------|------|--------------|--------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | Res. | Res. | Res. | HSI48 RDYIE | LSE CSSIE | Res. | Res. | Res. | PLL RDYIE | HSE RDYIE | HSI RDYIE | Res. | LSE RDYIE | LSE RDYIE |
| | | | | | RW | RW | | | | RW | RW | RW | | RW | RW |

Bits 31:11 Reserved, must be kept at reset value.

Bit 10 **HSI48RDYIE**: HSI48 ready interrupt enable

Set and cleared by software to enable/disable interrupt caused by the internal HSI48 oscillator.

0: HSI48 ready interrupt disabled
1: HSI48 ready interrupt enabled

Bit 9 **LSECSSIE**: LSE clock security system interrupt enable

Set and cleared by software to enable/disable interrupt caused by the clock security system on LSE.

0: Clock security interrupt caused by LSE clock failure disabled
1: Clock security interrupt caused by LSE clock failure enabled

Bits 8:6 Reserved, must be kept at reset value.

Bit 5 **PLLRDYIE**: PLL ready interrupt enable

Set and cleared by software to enable/disable interrupt caused by PLL lock.

0: PLL lock interrupt disabled
1: PLL lock interrupt enabled

Bit 4 **HSERDYIE**: HSE ready interrupt enable

Set and cleared by software to enable/disable interrupt caused by the HSE oscillator stabilization.

0: HSE ready interrupt disabled
1: HSE ready interrupt enabled

Bit 3 **HSIRDYIE**: HSI16 ready interrupt enable

Set and cleared by software to enable/disable interrupt caused by the HSI16 oscillator stabilization.

0: HSI16 ready interrupt disabled
1: HSI16 ready interrupt enabled

Bit 2 Reserved, must be kept at reset value.

Bit 1 **LSERDYIE**: LSE ready interrupt enable

Set and cleared by software to enable/disable interrupt caused by the LSE oscillator stabilization.

0: LSE ready interrupt disabled

1: LSE ready interrupt enabled

Bit 0 **LSIRDYIE**: LSI ready interrupt enable

Set and cleared by software to enable/disable interrupt caused by the LSI oscillator stabilization.

0: LSI ready interrupt disabled

1: LSI ready interrupt enabled

7.4.6 Clock interrupt flag register (RCC_CIFR)

Address offset: 0x1C

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|------|------|------|------|---------------|-------------|------|------|------|-------------|-------------|-------------|------|-------------|-------------|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | Res. | Res. | Res. | HSI48 RDYF | LSE CSSF | CSSF | Res. | Res. | PLL RDYF | HSE RDYF | HSI RDYF | Res. | LSE RDYF | LSI RDYF |
| | | | | | r | r | r | | | r | r | r | | r | r |

Bits 31:11 Reserved, must be kept at reset value.

Bit 10 **HSI48RDYF**: HSI48 ready interrupt flag

Set by hardware when the HSI48 clock becomes stable and HSI48RDYIE is set in a response to setting the HSI48ON (refer to [Clock recovery RC register \(RCC_CRRCR\)](#)).

Cleared by software setting the HSI48RDYC bit.

0: No clock ready interrupt caused by the HSI48 oscillator

1: Clock ready interrupt caused by the HSI48 oscillator

Bit 9 **LSECSSF**: LSE Clock security system interrupt flag

Set by hardware when a failure is detected in the LSE oscillator.

Cleared by software setting the LSECSSC bit.

0: No clock security interrupt caused by LSE clock failure

1: Clock security interrupt caused by LSE clock failure

Bit 8 **CSSF**: Clock security system interrupt flag

Set by hardware when a failure is detected in the HSE oscillator.

Cleared by software setting the CSSC bit.

0: No clock security interrupt caused by HSE clock failure

1: Clock security interrupt caused by HSE clock failure

Bits 7:6 Reserved, must be kept at reset value.

Bit 5 PLLRDYF: PLL ready interrupt flag

Set by hardware when the PLL locks and PLLRDYDIE is set.

Cleared by software setting the PLLRDYC bit.

0: No clock ready interrupt caused by PLL lock

1: Clock ready interrupt caused by PLL lock

Bit 4 HSERDYF: HSE ready interrupt flag

Set by hardware when the HSE clock becomes stable and HSERDYDIE is set.

Cleared by software setting the HSERDYC bit.

0: No clock ready interrupt caused by the HSE oscillator

1: Clock ready interrupt caused by the HSE oscillator

Bit 3 HSIRDYF: HSI16 ready interrupt flag

Set by hardware when the HSI16 clock becomes stable and HSIRDYDIE is set in a response to setting the HSION (refer to [Clock control register \(RCC_CR\)](#)). When HSION is not set but the HSI16 oscillator is enabled by the peripheral through a clock request, this bit is not set and no interrupt is generated.

Cleared by software setting the HSIRDYC bit.

0: No clock ready interrupt caused by the HSI16 oscillator

1: Clock ready interrupt caused by the HSI16 oscillator

Bit 2 Reserved, must be kept at reset value.

Bit 1 LSERDYF: LSE ready interrupt flag

Set by hardware when the LSE clock becomes stable and LSERDYDIE is set.

Cleared by software setting the LSERDYC bit.

0: No clock ready interrupt caused by the LSE oscillator

1: Clock ready interrupt caused by the LSE oscillator

Bit 0 LSIRDYF: LSI ready interrupt flag

Set by hardware when the LSI clock becomes stable and LSIRDYDIE is set.

Cleared by software setting the LSIRDYC bit.

0: No clock ready interrupt caused by the LSI oscillator

1: Clock ready interrupt caused by the LSI oscillator

7.4.7 Clock interrupt clear register (RCC_CICR)

Address offset: 0x20

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|---------------|-------------|------|------|------|-------------|-------------|-------------|------|-------------|-------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | Res. | Res. | Res. | HSI48 RDYC | LSE CS8C | CS9C | Res. | Res. | PLL RDYC | HSE RDYC | HSI RDYC | Res. | LSE RDYC | LSI RDYC |
| | | | | | w | w | w | | | w | w | w | | w | w |

Bits 31:11 Reserved, must be kept at reset value.

Bit 10 **HSI48RDYC**: HSI48 oscillator ready interrupt clear
This bit is set by software to clear the HSI48RDYF flag.
0: No effect
1: Clear the HSI48RDYC flag

Bit 9 **LSECSSC**: LSEC clock security system interrupt clear
This bit is set by software to clear the LSECSSF flag.
0: No effect
1: Clear LSECSSF flag

Bit 8 **CSSC**: Clock security system interrupt clear
This bit is set by software to clear the CSSF flag.
0: No effect
1: Clear CSSF flag

Bits 7:6 Reserved, must be kept at reset value.

Bit 5 **PLL RDYC**: PLL ready interrupt clear
This bit is set by software to clear the PLLRDYF flag.
0: No effect
1: Clear PLLRDYF flag

Bit 4 **HSE RDYC**: HSE ready interrupt clear
This bit is set by software to clear the HSERDYF flag.
0: No effect
1: Clear HSERDYF flag

Bit 3 **HSI16 RDYC**: HSI16 ready interrupt clear
This bit is set by software to clear the HSIRDYF flag.
0: No effect
1: Clear HSIRDYF flag

Bit 2 Reserved, must be kept at reset value.

Bit 1 **LSE RDYC**: LSE ready interrupt clear
This bit is set by software to clear the LSERDYF flag.
0: No effect
1: LSERDYF cleared

Bit 0 **LSI RDYC**: LSI ready interrupt clear
This bit is set by software to clear the LSIRDYF flag.
0: No effect
1: LSIRDYF cleared

7.4.8 AHB1 peripheral reset register (RCC_AHB1RSTR)

Address offset: 0x28

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access

| | | | | | | | | | | | | | | | |
|------|------|------|---------|------|------|------|-----------|------|------|------|----------|------------|-------------|----------|----------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | Res. | CRC RST | Res. | Res. | Res. | FLASH RST | Res. | Res. | Res. | FMAC RST | CORDIC RST | DMAMUX1 RST | DMA2 RST | DMA1 RST |
| | | | rw | | | | rw | | | | rw | rw | rw | rw | rw |

Bits 31:13 Reserved, must be kept at reset value.

Bit 12 CRCRST: CRC reset

Set and cleared by software.

0: No effect

1: ResetCRC

Bits 11:9 Reserved, must be kept at reset value.

Bit 8 FLASHRST: Flash memory interface reset

Set and cleared by software. This bit can be activated only when the Flash memory is in powerdown mode.

0: No effect

1: ResetFlash memory interface

Bits 7:5 Reserved, must be kept at reset value.

Bit 4 FMACRST: Set and cleared by software

0: No effect

1: ResetFMAC

Bit 3 CORDICRST: Set and cleared by software

0: No effect

1: ResetCORDIC

Bit 2 DMAMUX1RST: Set and cleared by software.

0: No effect

1: ResetDMAMUX1

Bit 1 DMA2RST: DMA2 reset

Set and cleared by software.

0: No effect

1: ResetDMA2

Bit 0 DMA1RST: DMA1 reset

Set and cleared by software.

0: No effect

1: ResetDMA1

7.4.9 AHB2 peripheral reset register (RCC_AHB2RSTR)

Address offset: 0x2C

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access

| | | | | | | | | | | | | | | | |
|------|---------------|--------------|------|------|------------|------|------------|------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | Res. | Res. | Res. | Res. | RNG RST | Res. | AES RST | Res. | Res. | Res. | Res. | DAC4 RST | DAC3 RST | DAC2 RST | DAC1 RST |
| | | | | | rw | | rw | | | | | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | ADC34S RST | ADC12 RST | Res. | Res. | Res. | Res. | Res. | Res. | GPIOG RST | GPIOF RST | GPIOE RST | GPIOD RST | GPIOC RST | GPIOB RST | GPIOA RST |
| | rw | rw | | | | | | | rw |

Bits 31:27 Reserved, must be kept at reset value.

Bit 26 **RNGRST**: RNG reset

Set and cleared by software.

0: No effect

1: Reset RNG

Bit 25 Reserved, must be kept at reset value.

Bit 24 **AESRST**: AESRST reset

Set and cleared by software.

0: No effect

1: Reset AES

Bits 23:20 Reserved, must be kept at reset value.

Bit 19 **DAC4RST**: DAC4 reset

Set and cleared by software.

0: No effect

1: Reset DAC4

Bit 18 **DAC3RST**: DAC3 reset

Set and cleared by software.

0: No effect

1: Reset DAC3

Bit 17 **DAC2RST**: DAC2 reset

Set and cleared by software.

0: No effect

1: Reset DAC2

Bit 16 **DAC1RST**: DAC1 reset

Set and cleared by software.

0: No effect

1: Reset DAC1

Bit 15 Reserved, must be kept at reset value.

Bit 14 **ADC345RST**: ADC345 reset

Set and cleared by software.

0: No effect

1: Reset ADC345

Bit 13 **ADC12RST**: ADC12 reset

Set and cleared by software.

0: No effect

1: Reset ADC12 interface

Bits 12:7 Reserved, must be kept at reset value.

Bit 6 **GPIOGRST**: IO port G reset

Set and cleared by software.

0: No effect

1: ResetIO port G

Bit 5 **GPIOFRST**: IO port F reset

Set and cleared by software.

0: No effect

1: ResetIO port F

Bit 4 **GPIOERST**: IO port E reset

Set and cleared by software.

0: No effect

1: ResetIO port E

Bit 3 **GPIODRST**: IO port D reset

Set and cleared by software.

0: No effect

1: ResetIO port D

Bit 2 **GPIOCRST**: IO port C reset

Set and cleared by software.

0: No effect

1: ResetIO port C

Bit 1 **GPIOBRST**: IO port B reset

Set and cleared by software.

0: No effect

1: ResetIO port B

Bit 0 **GPIOARST**: IO port A reset

Set and cleared by software.

0: No effect

1: ResetIO port A

7.4.10 AHB3 peripheral reset register (RCC_AHB3RSTR)

Address offset: 0x30

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|------|------|------|------|------|----------|------|------|------|------|------|------|------|------|---------|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | Res. | Res. | Res. | Res. | QSPIIRST | Res. | FMC RST |
| | | | | | | | RW | | | | | | | | RW |

Bits 31:9 Reserved, must be kept at reset value.

Bit 8 **QSPIIRST**: QUADSPI reset

Set and cleared by software.

0: No effect

1: Reset QUADSPI

Bits 7:1 Reserved, must be kept at reset value.

Bit 0 **FMC RST**: Flexible static memory controller reset

Set and cleared by software.

0: No effect

1: Reset FSMC

7.4.11 APB1 peripheral reset register 1 (RCC_APB1RSTR1)

Address offset: 0x38

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------------|----------|------|---------|------|------|-----------|------|---------|----------|----------|-----------|-----------|------------|------------|------|
| LPTIM1 RST | I2C3 RST | Res. | PWR RST | Res. | Res. | FDCAN RST | Res. | USB RST | I2C2 RST | I2C1 RST | UART5 RST | UART4 RST | USART3 RST | USART2 RST | Res. |
| IW | IW | | IW | | | IW | | IW | IW | IW | IW | IW | IW | IW | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SPI3 RST | SPI2 RST | Res. | Res. | Res. | Res. | CRS RST | Res. | Res. | TIM7 RST | TIM6 RST | TIM5 RST | TIM4 RST | TIM3 RST | TIM2 RST | |
| IW | IW | | | | | IW | | | IW | IW | IW | IW | IW | IW | |

- Bit 31 **LPTIM1RST**: Low Power Timer 1 reset
Set and cleared by software.
0: No effect
1: Reset LPTIM1
- Bit 30 **I2C3RST**: I2C3 reset
Set and cleared by software.
0: No effect
1: Reset I2C3 interface
- Bit 29 Reserved, must be kept at reset value.
- Bit 28 **PWRRST**: Power interface reset
Set and cleared by software.
0: No effect
1: Reset PWR
- Bits 27:26 Reserved, must be kept at reset value.
- Bit 25 **FDCANRST**: FDCAN reset
Set and reset by software.
0: No effect
1: Reset the FDCAN
- Bit 24 Reserved, must be kept at reset value.
- Bit 23 **USBRST**: USB device reset
Set and reset by software.
0: No effect
1: Reset USB device
- Bit 22 **I2C2RST**: I2C2 reset
Set and cleared by software.
0: No effect
1: Reset I2C2
- Bit 21 **I2C1RST**: I2C1 reset
Set and cleared by software.
0: No effect
1: Reset I2C1
- Bit 20 **UART5RST**: UART5 reset
Set and cleared by software.
0: No effect
1: Reset UART5
- Bit 19 **UART4RST**: UART4 reset
Set and cleared by software.
0: No effect
1: Reset UART4
- Bit 18 **USART3RST**: USART3 reset
Set and cleared by software.
0: No effect
1: Reset USART3

- Bit 17 **USART2RST**: USART2 reset
Set and cleared by software.
0: No effect
1: Reset USART2
- Bit 16 Reserved, must be kept at reset value.
- Bit 15 **SPI3RST**: SPI3 reset
Set and cleared by software.
0: No effect
1: Reset SPI3
- Bit 14 **SPI2RST**: SPI2 reset
Set and cleared by software.
0: No effect
1: Reset SPI2
- Bits 13:9 Reserved, must be kept at reset value.
- Bit 8 **CRSRST**: CRS reset
Set and cleared by software.
0: No effect
1: Reset CRS
- Bits 7:6 Reserved, must be kept at reset value.
- Bit 5 **TIM7RST**: TIM7 timer reset
Set and cleared by software.
0: No effect
1: Reset TIM7
- Bit 4 **TIM6RST**: TIM6 timer reset
Set and cleared by software.
0: No effect
1: Reset TIM7
- Bit 3 **TIM5RST**: TIM5 timer reset
Set and cleared by software.
0: No effect
1: Reset TIM5
- Bit 2 **TIM4RST**: TIM3 timer reset
Set and cleared by software.
0: No effect
1: Reset TIM3
- Bit 1 **TIM3RST**: TIM3 timer reset
Set and cleared by software.
0: No effect
1: Reset TIM3
- Bit 0 **TIM2RST**: TIM2 timer reset
Set and cleared by software.
0: No effect
1: Reset TIM2

7.4.12 APB1 peripheral reset register 2 (RCC_APB1RSTR2)

Address offset: 0x3C

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|-----------|------|------|------|------|------|----------|--------------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | UCPD1 RST | Res. | Res. | Res. | Res. | Res. | I2C4 RST | LP UART1 RST | |
| | | | | | | | rw | | | | | | rw | rw | |

Bits 31:9 Reserved, must be kept at reset value.

Bit 8 **UCPD1RST**: UCPD1 reset

Set and cleared by software.

0: No effect

1: Reset UCPD1

Bits 7:2 Reserved, must be kept at reset value.

Bit 1 **I2C4RST**: I2C4 reset

Set and cleared by software

0: No effect

1: Reset I2C4

Bit 0 **LPUART1RST**: Low-power UART 1 reset

Set and cleared by software.

0: No effect

1: Reset LPUART1

7.4.13 APB2 peripheral reset register (RCC_APB2RSTR)

Address offset: 0x40

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access

| | | | | | | | | | | | | | | | |
|----------|------------|----------|----------|----------|------------|------|------|------|------|----------|-----------|------|-----------|-----------|-------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | Res. | Res. | Res. | Res. | HRTIM1 RST | Res. | Res. | Res. | Res. | SAI1 RST | TIM20 RST | Res. | TIM17 RST | TIM16 RST | TIM15R ST |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SPI4 RST | USART1 RST | TIMB RST | SPI1 RST | TIM1 RST | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | SYS CFG RST |
| rw | rw | rw | rw | rw | | | | | | | | | | | rw |

Bits 31:27 Reserved, must be kept at reset value.

Bit 26 **HRTIM1RST**: HRTIM1 reset

Set and cleared by software.

0: No effect

1: Reset HRTIM1

Bits 25:22 Reserved, must be kept at reset value.

Bit 21 **SA1RST**: Serial audio interface 1 (SAI1) reset

Set and cleared by software.

0: No effect

1: Reset SAI1

Bit 20 **TIM20RST**: TIM20 reset

Set and cleared by software.

0: No effect

1: Reset TIM20

Bit 19 Reserved, must be kept at reset value.

Bit 18 **TIM17RST**: TIM17 timer reset

Set and cleared by software.

0: No effect

1: Reset TIM17 timer

Bit 17 **TIM16RST**: TIM16 timer reset

Set and cleared by software.

0: No effect

1: Reset TIM16 timer

Bit 16 **TIM15RST**: TIM15 timer reset

Set and cleared by software.

0: No effect

1: Reset TIM15 timer

Bit 15 **SPI4RST**: SPI4 reset

Set and cleared by software.

0: No effect

1: Reset SPI4

Bit 14 **USART1RST**: USART1 reset

Set and cleared by software.

0: No effect

1: Reset USART1

Bit 13 **TIM8RST**: TIM8 timer reset

Set and cleared by software.

0: No effect

1: Reset TIM8 timer

Bit 12 **SPI1RST**: SPI1 reset

Set and cleared by software.

0: No effect

1: Reset SPI1

Bit 11 TIM1RST: TIM1 timer reset
 Set and cleared by software.
 0: No effect
 1: Reset TIM1 timer

Bits 10:1 Reserved, must be kept at reset value.

Bit 0 SYSCFGRST: SYSCFG + COMP + OPAMP + VREFBUF reset
 0: No effect
 1: Reset SYSCFG + COMP + OPAMP + VREFBUF

7.4.14 AHB1 peripheral clock enable register (RCC_AHB1ENR)

Address offset: 0x48

Reset value: 0x0000 0100

Access: no wait state, word, half-word and byte access

Note: When the peripheral clock is not active, the peripheral registers read or write access is not supported.

| | | | | | | | | | | | | | | | |
|------|------|------|-------|------|------|------|---------|------|------|------|--------|----------|-----------|--------|--------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | Res. | CRCEN | Res. | Res. | Res. | FLASHEN | Res. | Res. | Res. | FMACEN | CORDICEN | DMA1UX1EN | DMA2EN | DMA1EN |
| | | | rw | | | | rw | | | | rw | rw | rw | rw | rw |

Bits 31:13 Reserved, must be kept at reset value.

Bit 12 CRCEN: CRC clock enable
 Set and cleared by software.
 0: CRC clock disable
 1: CRC clock enable

Bits 11:9 Reserved, must be kept at reset value.

Bit 8 FLASHEN: Flash memory interface clock enable
 Set and cleared by software. This bit can be disabled only when the Flash is in power down mode.
 0: Flash memory interface clock disable
 1: Flash memory interface clock enable

Bits 7:5 Reserved, must be kept at reset value.

Bit 4 FMACEN: FMAC enable
 Set and reset by software.
 0: FMAC clock disabled
 1: FMAC clock enabled

Bit 3 CORDICEN: CORDIC clock enable
 Set and reset by software.
 0: CORDIC clock disabled
 1: CORDIC clock enabled

Bit 2 **DMAMUX1EN**: DMAMUX1 clock enable

Set and reset by software.

0: DMAMUX1 clock disabled

1: DMAMUX1 clock enabled

Bit 1 **DMA2EN**: DMA2 clock enable

Set and cleared by software.

0: DMA2 clock disable

1: DMA2 clock enable

Bit 0 **DMA1EN**: DMA1 clock enable

Set and cleared by software.

0: DMA1 clock disable

1: DMA1 clock enable

7.4.15 AHB2 peripheral clock enable register (RCC_AHB2ENR)

Address offset: 0x4C

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access

Note: When the peripheral clock is not active, the peripheral registers read or write access is not supported.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|-----------|-----------|------|------|--------|------|--------|------|-----------|----------|----------|----------|----------|----------|----------|
| Res. | Res. | Res. | Res. | Res. | RNG EN | Res. | AES EN | Res. | Res. | Res. | Res. | DAC 4 EN | DAC3 EN | DAC2 EN | DAC1 EN |
| | | | | | rw | | rw | | | | | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | ADC345 EN | ADC 12 EN | Res. | Res. | Res. | Res. | Res. | Res. | GPIO G EN | GPIOF EN | GPIOE EN | GPIOD EN | GPIOC EN | GPIOB EN | GPIOA EN |
| | rw | rw | | | | | | | rw | rw | rw | rw | rw | rw | rw |

Bits 31:27 Reserved, must be kept at reset value.

Bit 26 **RNGEN**: RNG enable

Set and cleared by software.

0: RNG disabled

1: RNG enabled

Bit 25 Reserved, must be kept at reset value.

Bit 24 **AESEN**: AES clock enable

Set and cleared by software.

0: AES clock disabled

1: AES clock enabled

Bits 23:20 Reserved, must be kept at reset value.

Bit 19 **DAC4EN**: DAC4 clock enable

Set and cleared by software.

0: DAC4 clock disabled

1: DAC4 clock enabled

- Bit 18 **DAC3EN**: DAC3 clock enable
Set and cleared by software.
0: DAC3 clock disabled
1: DAC3 clock enabled
- Bit 17 **DAC2EN**: DAC2 clock enable
Set and cleared by software.
0: DAC2 clock disabled
1: DAC2 clock enabled
- Bit 16 **DAC1EN**: DAC1 clock enable
Set and cleared by software.
0: DAC1 clock disabled
1: DAC1 clock enabled
- Bit 15 Reserved, must be kept at reset value.
- Bit 14 **ADC345EN**: ADC345 clock enable
Set and cleared by software
0: ADC 345 clock disabled
1: ADC 345 clock enabled
- Bit 13 **ADC12EN**: ADC12 clock enable
Set and cleared by software.
0: ADC 12 clock disabled
1: ADC 12 clock enabled
- Bits 12:7 Reserved, must be kept at reset value.
- Bit 6 **GPIOGEN**: IO port G clock enable
Set and cleared by software.
0: IO port G clock disabled
1: IO port G clock enabled
- Bit 5 **GPIOFEN**: IO port F clock enable
Set and cleared by software.
0: IO port F clock disabled
1: IO port F clock enabled
- Bit 4 **GPIOEEN**: IO port E clock enable
Set and cleared by software.
0: IO port E clock disabled
1: IO port E clock enabled
- Bit 3 **GIODEN**: IO port D clock enable
Set and cleared by software.
0: IO port D clock disabled
1: IO port D clock enabled

Bit 2 **GPIOCEN**: IO port C clock enable

Set and cleared by software.

0: IO port C clock disabled

1: IO port C clock enabled

Bit 1 **GPIOBEN**: IO port B clock enable

Set and cleared by software.

0: IO port B clock disabled

1: IO port B clock enabled

Bit 0 **GPIOAEN**: IO port A clock enable

Set and cleared by software.

0: IO port A clock disabled

1: IO port A clock enabled

7.4.16 AHB3 peripheral clock enable register(RCC_AHB3ENR)

Address offset: 0x50

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access

Note: When the peripheral clock is not active, the peripheral registers read or write access is not supported.

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|--------|------|------|------|------|------|------|------|--------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | QSPIEN | Res. | FMC EN |
| | | | | | | | rw | | | | | | | | rw |

Bits 31:9 Reserved, must be kept at reset value.

Bit 8 **QSPIEN**: QIADSPI memory interface clock enable

Set and cleared by software.

0: QIADSPI clock disable

1: QIADSPI clock enable

Bits 7:1 Reserved, must be kept at reset value.

Bit 0 **FMCEN**: Flexible static memory controller clock enable

Set and cleared by software.

0: FSMC clock disable

1: FSMC clock enable

7.4.17 APB1 peripheral clock enable register 1 (RCC_APB1ENR1)

Address: 0x58

Reset value: 0x0000 0400

Access: no wait state, word, half-word and byte access

Note: When the peripheral clock is not active, the peripheral registers read or write access is not supported.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|--------------|------------|------|-----------|------------|--------------|-------------|-----------|-----------|------------|------------|-------------|-------------|--------------|--------------|------------|
| LPTIM1 EN | I2C3 EN | Res. | PWR EN | Res. | Res. | FDCAN EN | Res. | USB EN | I2C2 EN | I2C1 EN | UART5 EN | UART4 EN | USART3 EN | USART2 EN | Res. |
| rw | rw | | rw | | | rw | | rw | rw | rw | rw | rw | rw | rw | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SPI3 EN | SPI2 EN | Res. | Res. | WWDG EN | RTCAPB EN | Res. | CRS EN | Res. | Res. | TIM7 EN | TIM6 EN | TIM5 EN | TIM4 EN | TIM3 EN | TIM2 EN |
| rw | rw | | | is | rw | | rw | | | rw | rw | rw | rw | rw | rw |

Bit 31 **LPTIM1EN**: Low power timer 1 clock enable

Set and cleared by software.

0: LPTIM1 clock disabled

1: LPTIM1 clock enabled

Bit 30 **I2C3EN**: I2C3 clock enable

Set and cleared by software.

0: I2C3 clock disabled

1: I2C3 clock enabled

Bit 29 Reserved, must be kept at reset value.

Bit 28 **PWREN**: Power interface clock enable

Set and cleared by software.

0: Power interface clock disabled

1: Power interface clock enabled

Bits 27:26 Reserved, must be kept at reset value.

Bit 25 **FDCANEN**: FDCAN clock enable

Set and cleared by software.

0: FDCAN clock disabled

1: FDCAN clock enabled

Bit 24 Reserved, must be kept at reset value.

Bit 23 **USBEN**: USB device clock enable

Set and cleared by software.

0: USB device clock disabled

1: USB device clock enabled

Bit 22 **I2C2EN**: I2C2 clock enable

Set and cleared by software.

0: I2C2 clock disabled

1: I2C2 clock enabled

- Bit 21 **I2C1EN**: I2C1 clock enable
Set and cleared by software.
0: I2C1 clock disabled
1: I2C1 clock enabled
- Bit 20 **UART5EN**: UART5 clock enable
Set and cleared by software.
0: UART5 clock disabled
1: UART5 clock enabled
- Bit 19 **UART4EN**: UART4 clock enable
Set and cleared by software.
0: UART4 clock disabled
1: UART4 clock enabled
- Bit 18 **USART3EN**: USART3 clock enable
Set and cleared by software.
0: USART3 clock disabled
1: USART3 clock enabled
- Bit 17 **USART2EN**: USART2 clock enable
Set and cleared by software.
0: USART2 clock disabled
1: USART2 clock enabled
- Bit 16 Reserved, must be kept at reset value.
- Bit 15 **SPI3EN**: SPI3 clock enable
Set and cleared by software.
0: SPI3 clock disabled
1: SPI3 clock enabled
- Bit 14 **SPI2EN**: SPI2 clock enable
Set and cleared by software.
0: SPI2 clock disabled
1: SPI2 clock enabled
- Bits 13:12 Reserved, must be kept at reset value.
- Bit 11 **WWDGGEN**: Window watchdog clock enable
Set by software to enable the window watchdog clock. Reset by hardware system reset.
This bit can also be set by hardware if the WWDG_SW option bit is reset.
0: Window watchdog clock disabled
1: Window watchdog clock enabled
- Bit 10 **RTCAPBEN**: RTC APB clock enable
Set and cleared by software
0: RTC APB clock disabled
1: RTC APB clock enabled
- Bit 9 Reserved, must be kept at reset value.
- Bit 8 **CRSEN**: CRS Recovery System clock enable
Set and cleared by software.
0: CRS clock disabled
1: CRS clock enabled
- Bits 7:6 Reserved, must be kept at reset value.

Bit 5 **TIM7EN**: TIM7 timer clock enable

Set and cleared by software.

0: TIM7 clock disabled

1: TIM7 clock enabled

Bit 4 **TIM6EN**: TIM6 timer clock enable

Set and cleared by software.

0: TIM6 clock disabled

1: TIM6 clock enabled

Bit 3 **TIM5EN**: TIM5 timer clock enable

Set and cleared by software.

0: TIM5 clock disabled

1: TIM5 clock enabled

Bit 2 **TIM4EN**: TIM4 timer clock enable

Set and cleared by software.

0: TIM4 clock disabled

1: TIM4 clock enabled

Bit 1 **TIM3EN**: TIM3 timer clock enable

Set and cleared by software.

0: TIM3 clock disabled

1: TIM3 clock enabled

Bit 0 **TIM2EN**: TIM2 timer clock enable

Set and cleared by software.

0: TIM2 clock disabled

1: TIM2 clock enabled

7.4.18 APB1 peripheral clock enable register 2 (RCC_APB1ENR2)

Address offset: 0x5C

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access

Note: When the peripheral clock is not active, the peripheral registers read or write access is not supported.

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|-------------|------|------|------|------|------|------|--------|-------------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | B | 7 | 6 | 5 | 4 | 3 | 2 | 1 | D |
| Res. | UCPD1 EN | Res. | Res. | Res. | Res. | Res. | Res. | I2C4EN | LP UART1 EN |
| | | | | | | | rw | | | | | | | rw | rw |

Bits 31:9 Reserved, must be kept at reset value.

Bit 8 **UCPD1EN**: UCPD1 clock enable

Set and cleared by software.

0: UCPD1 clock disable

1: UCPD1 clock enable

Bits 7:2 Reserved, must be kept at reset value.

Bit 1 **I2C4EN**: I2C4 clock enable

Set and cleared by software

0: I2C4 clock disabled

1: I2C4 clock enabled

Bit 0 **LPUART1EN**: Low power UART 1 clock enable

Set and cleared by software.

0: LPUART1 clock disable

1: LPUART1 clock enable

7.4.19 APB2 peripheral clock enable register (RCC_APB2ENR)

Address: 0x60

Reset value: 0x0000 0000

Access: word, half-word and byte access

Note: When the peripheral clock is not active, the peripheral registers read or write access is not supported.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------------|--------------|------------|------------|------------|--------------|------|------|------|------|------------|-------------|------|-------------|-------------|---------------|
| Res. | Res. | Res. | Res. | Res. | HRTIM1 EN | Res. | Res. | Res. | Res. | SAI1 EN | TIM20 EN | Res. | TIM 17EN | TIM16 EN | TIM15 EN |
| | | | | | rw | | | | | rw | rw | | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SPI4 EN | USART1 EN | TIM8 EN | SPI1 EN | TIM1 EN | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | SYS CFGGEN |
| rw | rw | rw | rw | rw | | | | | | | | | | | rw |

Bits 31:27 Reserved, must be kept at reset value.

Bit 26 **HRTIM1EN**: HRTIM1 clock enable

Set and cleared by software.

0: HRTIM1 clock disabled

1: HRTIM1 clock enabled

Bits 25:22 Reserved, must be kept at reset value.

Bit 21 **SAI1EN**: SAI1 clock enable

Set and cleared by software.

0: SAI1 clock disabled

1: SAI1 clock enabled

Bit 20 **TIM20EN**: TIM20 timer clock enable

Set and cleared by software.

0: TIM20 clock disabled

1: TIM20 clock enabled

Bit 19 Reserved, must be kept at reset value.

Bit 18 **TIM17EN**: TIM17 timer clock enable

Set and cleared by software.

0: TIM17 timer clock disabled

1: TIM17 timer clock enabled

Bit 17 **TIM16EN**: TIM16 timer clock enable

Set and cleared by software.

0: TIM16 timer clock disabled

1: TIM16 timer clock enabled

Bit 16 **TIM15EN**: TIM15 timer clock enable

Set and cleared by software.

0: TIM15 timer clock disabled

1: TIM15 timer clock enabled

Bit 15 **SPI4EN**: SPI4 clock enable

Set and cleared by software.

0: SPI4 clock disabled

1: SPI4 clock enabled

Bit 14 **USART1EN**: USART1 clock enable

Set and cleared by software.

0: USART1 clock disabled

1: USART1 clock enabled

Bit 13 **TIM8EN**: TIM8 timer clock enable

Set and cleared by software.

0: TIM8 timer clock disabled

1: TIM8 timer clock enabled

Bit 12 **SPI1EN**: SPI1 clock enable

Set and cleared by software.

0: SPI1 clock disabled

1: SPI1 clock enabled

Bit 11 **TIM1EN**: TIM1 timer clock enable

Set and cleared by software.

0: TIM1 timer clock disabled

1: TIM1P timer clock enabled

Bits 10:1 Reserved, must be kept at reset value.

Bit 0 **SYSCFGGEN**: SYSCFG + COMP + VREFBUF + OPAMP clock enable

Set and cleared by software.

0: SYSCFG + COMP + VREFBUF + OPAMP clock disabled

1: SYSCFG + COMP + VREFBUF + OPAMP clock enabled

7.4.20 AHB1 peripheral clocks enable in Sleep and Stop modes register (RCC_AHB1SMENR)

Address offset: 0x68

Reset value: 0x0000 130F

Access: no wait state, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|------|------|--------------|------|------|---------------|---------------|------|------|------|---------------|-----------------|-----------------|--------------|--------------|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | Res. | CRC5IM EN | Res. | Res. | SRAM1 SMEN | FLASH SMEN | Res. | Res. | Res. | FMAC5IM EN | CORDIC5IM EN | DMAMUX1 SMEN | DMA2 SMEN | DMA1 SMEN |
| | | | rw | | | rw | rw | | | | rw | rw | rw | rw | rw |

Bits 31:13 Reserved, must be kept at reset value.

Bit 12 CRCSMEN: CRC clocks enable during Sleep and Stop modes

Set and cleared by software.

0: CRC clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1: CRC clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

Bits 11:10 Reserved, must be kept at reset value.

Bit 9 SRAM1SMEN: SRAM1 interface clocks enable during Sleep and Stop modes

Set and cleared by software.

0: SRAM1 interface clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1: SRAM1 interface clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

Bit 8 FLASHSMEN: Flash memory interface clocks enable during Sleep and Stop modes

Set and cleared by software.

0: Flash memory interface clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1: Flash memory interface clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

Bits 7:5 Reserved, must be kept at reset value.

Bit 4 FMACSMEN: FMACSM clock enable.

Set and cleared by software.

0: FMACSM clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1: FMACSM clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

Bit 3 CORDICSMEN: CORDICSM clock enable.

Set and cleared by software.

0: CORDICSM clocks disabled.

1: CORDICSM clocks enabled.

Bit 2 DMAMUX1SMEN: DMAMUX1 clock enable during Sleep and Stop modes.

Set and cleared by software.

0: DMAMUX1 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1: DMAMUX1 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

Bit 1 DMA2SMEN: DMA2 clocks enable during Sleep and Stop modes

Set and cleared by software during Sleep mode.

0: DMA2 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1: DMA2 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

Bit 0 DMA1SMEN: DMA1 clocks enable during Sleep and Stop modes

Set and cleared by software.

0: DMA1 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1: DMA1 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1. This register only configures the clock gating, not the clock source itself. Most of the peripherals are clocked by a single clock (AHB or APB clock), which is always disabled in Stop mode. In this case setting the bit has no effect in Stop mode.

7.4.21 AHB2 peripheral clocks enable in Sleep and Stop modes register (RCC_AHB2SMENR)

Address offset: 0x6C

Reset value: 0x050F 667F

Access: no wait state, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|------|-------------|------------|------|------|------------|---------------|---------|------|------------|------------|------------|------------|------------|------------|------------|
| Res. | Res. | Res. | Res. | Res. | RNG EN | Res. | AESM EN | Res. | Res. | Res. | Res. | DAC4 SMEN | DAC3 SMEN | DAC2 SMEN | DAC1 SMEN |
| | | | | | rw | | rw | | | | | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | ADC345 SMEN | ADC12 SMEN | Res. | Res. | SRAM2 SMEN | CCM3 RAM SMEN | Res. | Res. | GPIOG SMEN | GPIOF SMEN | GPIOE SMEN | GPIOD SMEN | GPIOC SMEN | GPIOB SMEN | GPIOA SMEN |
| | rw | rw | | | rw | rw | | | rw |

Bits 31:27 Reserved, must be kept at reset value.

Bit 26 **RNGEN**: RNG enable

Set and cleared by software.

0: RNG disabled

1: RNG enabled

Bit 25 Reserved, must be kept at reset value.

Bit 24 **AESMEN**: AESM clocks enable

Set and cleared by software.

0: AESM clocks disabled

1: AESM clocks enabled

Bits 23:20 Reserved, must be kept at reset value.

Bit 19 **DAC4SMEN**: DAC4 clock enable

Set and cleared by software.

0: DAC4 clock disabled

1: DAC4 clock enabled during sleep and stop modes

Bit 18 **DAC3SMEN**: DAC3 clock enable

Set and cleared by software.

0: DAC3 clock disabled

1: DAC3 clock enabled during sleep and stop modes

Bit 17 **DAC2SMEN**: DAC2 clock enable

Set and cleared by software.

0: DAC2 clock disabled

1: DAC2 clock enabled during sleep and stop modes

Bit 16 **DAC1SMEN**: DAC1 clock enable

Set and cleared by software.

0: DAC1 clock disabled

1: DAC1 clock enabled during sleep and stop modes

Bit 15 Reserved, must be kept at reset value.

- Bit 14 **ADC345SMEN**: ADC345 clock enable
Set and cleared by software.
0: ADC345 clock disabled
1: ADC345 clock enabled
- Bit 13 **ADC12SMEN**: ADC12 clocks enable during Sleep and Stop modes
Set and cleared by software.
0: ADC12 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: ADC12 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bits 12:11 Reserved, must be kept at reset value.
- Bit 10 **SRAM2SMEN**: SRAM2 interface clocks enable during Sleep and Stop modes
Set and cleared by software.
0: SRAM2 interface clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: SRAM2 interface clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 9 **CCMSRAMSMEN**: CCM SRAM interface clocks enable during Sleep and Stop modes
Set and cleared by software.
0: CCM SRAM interface clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: CCM SRAM interface clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bits 8:7 Reserved, must be kept at reset value.
- Bit 6 **GPIOGSMEN**: IO port G clocks enable during Sleep and Stop modes
Set and cleared by software.
0: IO port G clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: IO port G clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 5 **GPIOFSMEN**: IO port F clocks enable during Sleep and Stop modes
Set and cleared by software.
0: IO port F clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: IO port F clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 4 **GPIOESMEN**: IO port E clocks enable during Sleep and Stop modes
Set and cleared by software.
0: IO port E clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: IO port E clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 3 **GPIODSMEN**: IO port D clocks enable during Sleep and Stop modes
Set and cleared by software.
0: IO port D clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: IO port D clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 2 **GPIOCSMEN**: IO port C clocks enable during Sleep and Stop modes
Set and cleared by software.
0: IO port C clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: IO port C clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 1 **GPIOBSMEN**: IO port B clocks enable during Sleep and Stop modes
Set and cleared by software.
0: IO port B clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: IO port B clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 0 **GPIOASMEN**: IO port A clocks enable during Sleep and Stop modes
Set and cleared by software.
0: IO port A clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: IO port A clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1. This register only configures the clock gating, not the clock source itself. Most of the peripherals are clocked by a single clock (AHB or APB clock), which is always disabled in Stop mode. In this case setting the bit has no effect in Stop mode.

7.4.22 AHB3 peripheral clocks enable in Sleep and Stop modes register (RCC_AHB3SMENR)

Address offset: 0x70

Reset value: 0x0000 0101

Access: no wait state, word, half-word and byte access

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|-----------|------|------|------|------|------|------|------|----------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | Res. | Res. | Res. | Res. | QSPI SMEN | Res. | FMC SMEN | Res. |
| | | | | | | rw | | | | | | | | | rw |

Bits 31:9 Reserved, must be kept at reset value.

Bit 8 **QSPI SMEN**: QUADSPI memory interface clock enable during Sleep and Stop modes
Set and cleared by software.

0: QUADSPI clock disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: QUADSPI clock enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

Bits 7:1 Reserved, must be kept at reset value.

Bit 0 **FMC SMEN**: Flexible static memory controller clocks enable during Sleep and Stop modes
Set and cleared by software.

0: FMC clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: FMC clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1. This register only configures the clock gating, not the clock source itself. Most of the peripherals are clocked by a single clock (AHB or APB clock), which is always disabled in Stop mode. In this case setting the bit has no effect in Stop mode

7.4.23 APB1 peripheral clocks enable in Sleep and Stop modes register 1 (RCC_APB1SMENR1)

Address: 0x78

Reset value: 0xD2FE CD3F

Access: no wait state, word, half-word and byte access

| | | | | | | | | | | | | | | | |
|-------------|-----------|------|----------|-----------|-------------|------------|----------|----------|-----------|-----------|------------|------------|---------------|---------------|-----------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| LPTIM1 SMEN | I2C3 SMEN | Res. | PWR SMEN | Res. | Res. | FDCAN SMEN | Res. | USB SMEN | I2C2 SMEN | I2C1 SMEN | UART5 SMEN | UART4 SMEN | USART 38 SMEN | USART 29 SMEN | Res. |
| rw | rw | | rw | | | rw | | rw | rw | rw | rw | rw | rw | rw | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SPI3 SMEN | SPI2 SMEN | Res. | Res. | VWDG SMEN | RTCAPB SMEN | Res. | CRS SMEN | Res. | Res. | TIM7 SMEN | TIM6 SMEN | TIM5 SMEN | TIM4 SMEN | TIM3 SMEN | TIM2 SMEN |
| rw | rw | | | rw | rw | | rw | | | rw | rw | rw | rw | rw | rw |

- Bit 31 **LPTIM1SMEN**: Low power timer 1 clocks enable during Sleep and Stop modes
Set and cleared by software.
0: LPTIM1 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: LPTIM1 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 30 **I2C3SMEN**: I2C 3 clocks enable during Sleep and Stop modes
Set and cleared by software.
0: I2C3 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: I2C3 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 29 Reserved, must be kept at reset value.
- Bit 28 **PWRSMEN**: Power interface clocks enable during Sleep and Stop modes
Set and cleared by software.
0: Power interface clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: Power interface clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bits 27:26 Reserved, must be kept at reset value.
- Bit 25 **FDCANSMEN**: FDCAN clocks enable during Sleep and Stop modes
Set and cleared by software.
0: FDCAN clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: FDCAN clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 24 Reserved, must be kept at reset value.
- Bit 23 **USBSMEN**: USB device clocks enable during Sleep and Stop modes
Set and cleared by software.
0: USB device clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: USB device clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 22 **I2C2SMEN**: I2C 2 clocks enable during Sleep and Stop modes
Set and cleared by software.
0: I2C2 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: I2C2 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 21 **I2C1SMEN**: I2C 1 clocks enable during Sleep and Stop modes
Set and cleared by software.
0: I2C1 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: I2C1 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 20 **UART5SMEN**: UART5 clocks enable during Sleep and Stop modes
Set and cleared by software.
0: UART5 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: UART5 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 19 **UART4SMEN**: UART4 clocks enable during Sleep and Stop modes
Set and cleared by software.
0: UART4 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: UART4 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 18 **USART3SMEN**: USART3 clocks enable during Sleep and Stop modes
Set and cleared by software.
0: USART3 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: USART3 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

- Bit 17 **USART2SMEN**: USART2 clocks enable during Sleep and Stop modes
 Set and cleared by software.
 0: USART2 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
 1: USART2 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 16 Reserved, must be kept at reset value.
- Bit 15 **SPI3SMEN**: SPI3 clocks enable during Sleep and Stop modes
 Set and cleared by software.
 0: SPI3 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
 1: SPI3 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 14 **SPI2SMEN**: SPI2 clocks enable during Sleep and Stop modes
 Set and cleared by software.
 0: SPI2 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
 1: SPI2 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bits 13:12 Reserved, must be kept at reset value.
- Bit 11 **WWDGSMEN**: Window watchdog clocks enable during Sleep and Stop modes
 Set and cleared by software. This bit is forced to '1' by hardware when the hardware WWDG option is activated.
 0: Window watchdog clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
 1: Window watchdog clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 10 **RTCAPBSMEN**: RTC APB clock enable during Sleep and Stop modes
 Set and cleared by software
 0: RTC APB clock disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
 1: RTC APB clock enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 9 Reserved, must be kept at reset value.
- Bit 8 **CRSSMEN**: CRS timer clocks enable during Sleep and Stop modes
 Set and cleared by software.
 0: CRS clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
 1: CRS clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bits 7:6 Reserved, must be kept at reset value.
- Bit 5 **TIM7SMEN**: TIM7 timer clocks enable during Sleep and Stop modes
 Set and cleared by software.
 0: TIM7 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
 1: TIM7 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 4 **TIM6SMEN**: TIM6 timer clocks enable during Sleep and Stop modes
 Set and cleared by software.
 0: TIM6 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
 1: TIM6 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 3 **TIM5SMEN**: TIM5 timer clocks enable during Sleep and Stop modes
 Set and cleared by software.
 0: TIM5 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
 1: TIM5 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 2 **TIM4SMEN**: TIM4 timer clocks enable during Sleep and Stop modes
 Set and cleared by software.
 0: TIM4 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
 1: TIM4 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

- Bit 1 **TIM3SMEN**: TIM3 timer clocks enable during Sleep and Stop modes
Set and cleared by software.
0: TIM3 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: TIM3 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 0 **TIM2SMEN**: TIM2 timer clocks enable during Sleep and Stop modes
Set and cleared by software.
0: TIM2 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: TIM2 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1. This register only configures the clock gating, not the clock source itself. Most of the peripherals are clocked by a single clock (AHB or APB clock), which is always disabled in Stop mode. In this case setting the bit has no effect in Stop mode.

7.4.24 APB1 peripheral clocks enable in Sleep and Stop modes register 2 (RCC_APB1SMENR2)

Address offset: 0x7C

Reset value: 0x0000 0103

Access: no wait state, word, half-word and byte access

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|-----------|------|------|------|------|------|----------|--------------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | UCPD1SMEN | Res. | Res. | Res. | Res. | Res. | I2C4SMEN | LP UART1SMEN | |
| | | | | | | | rw | | | | | | | rw | rw |

Bits 31:9 Reserved, must be kept at reset value.

- Bit 8 **UCPD1SMEN**: UCPD1 clocks enable during Sleep and Stop modes
Set and cleared by software.
0: UCPD1 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: UCPD1 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

Bits 7:2 Reserved, must be kept at reset value.

- Bit 1 **I2C4SMEN**: I2C4 clocks enable during Sleep and Stop modes
Set and cleared by software.
0: I2C4 clocks disabled by the clock gating⁽²⁾ during Sleep and Stop modes
1: I2C4 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

- Bit 0 **LPUART1SMEN**: Low power UART 1 clocks enable during Sleep and Stop modes
Set and cleared by software.

- 0: LPUART1 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: LPUART1 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1. This register only configures the clock gating, not the clock source itself. Most of the peripherals are clocked by a single clock (AHB or APB clock), which is always disabled in Stop mode. In this case setting the bit has no effect in Stop mode.

2. This register only configures the clock gating, not the clock source itself. Most of the peripherals are clocked by a single clock (AHB or APB clock), which is always disabled in Stop mode. In this case setting the bit has no effect in Stop mode.

7.4.25 APB2 peripheral clocks enable in Sleep and Stop modes register (RCC_APB2SMENR)

Address: 0x80

Reset value: 0x0437 F801

Access: word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|--------------|----------------|--------------|--------------|--------------|----------------|------|------|------|------|--------------|---------------|------|---------------|---------------|--------------------|
| Res. | Res. | Res. | Res. | Res. | HRTIM1 SMEN | Res. | Res. | Res. | Res. | SAI1 SMEN | TIM20 SMEN | Res. | TIM17 SMEN | TIM16 SMEN | TIM15 SMEN |
| | | | | | IW | | | | | IW | IW | | IW | IW | IW |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SPI4 SMEN | USART1 SMEN | TIMB SMEN | SPI1 SMEN | TIM1 SMEN | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | SYS CFG SMEN |
| IW | IW | IW | IW | IW | | | | | | | | | | | IW |

Bits 31:27 Reserved, must be kept at reset value.

Bit 26 **HRTIM1SMEN**: HRTIM1 timer clocks enable during Sleep and Stop modes

Set and cleared by software.

0: HRTIM1 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1: HRTIM1 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

Bits 25:22 Reserved, must be kept at reset value.

Bit 21 **SAI1SMEN**: SAI1 clocks enable during Sleep and Stop modes

Set and cleared by software.

0: SAI1 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1: SAI1 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

Bit 20 **TIM20SMEN**: TIM20 timer clocks enable during Sleep and Stop modes

Set and cleared by software.

0: TIM20 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1: TIM20 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

Bit 19 Reserved, must be kept at reset value.

Bit 18 **TIM17SMEN**: TIM17 timer clocks enable during Sleep and Stop modes

Set and cleared by software.

0: TIM17 timer clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1: TIM17 timer clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

Bit 17 **TIM16SMEN**: TIM16 timer clocks enable during Sleep and Stop modes

Set and cleared by software.

0: TIM16 timer clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1: TIM16 timer clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

Bit 16 **TIM15SMEN**: TIM15 timer clocks enable during Sleep and Stop modes

Set and cleared by software.

0: TIM15 timer clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1: TIM15 timer clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop mode

- Bit 15 **SPI4SMEN**: SPI4 timer clocks enable during Sleep and Stop modes
Set and cleared by software.
0: SPI4 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: SPI4 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop mode
- Bit 14 **USART1SMEN**: USART1 clocks enable during Sleep and Stop modes
Set and cleared by software.
0: USART1 clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: USART1 clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 13 **TIM8SMEN**: TIM8 timer clocks enable during Sleep and Stop modes
Set and cleared by software.
0: TIM8 timer clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: TIM8 timer clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bit 12 **SPI1SMEN**: SPI1 clocks enable during Sleep and Stop modes
Set and cleared by software.
0: SPI1 clocks disabled by the clock gating during⁽¹⁾ Sleep and Stop modes
1: SPI1 clocks enabled by the clock gating during⁽¹⁾ Sleep and Stop modes
- Bit 11 **TIM1SMEN**: TIM1 timer clocks enable during Sleep and Stop modes
Set and cleared by software.
0: TIM1 timer clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: TIM1P timer clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes
- Bits 10:1 Reserved, must be kept at reset value.
- Bit 0 **SYSCFGSMEN**: SYSCFG + COMP + VREFBUF + OPAMP clocks enable during Sleep and Stop modes
Set and cleared by software.
0: SYSCFG + COMP + VREFBUF + OPAMP clocks disabled by the clock gating⁽¹⁾ during Sleep and Stop modes
1: SYSCFG + COMP + VREFBUF + OPAMP clocks enabled by the clock gating⁽¹⁾ during Sleep and Stop modes

1. This register only configures the clock gating, not the clock source itself. Most of the peripherals are clocked by a single clock (AHB or APB clock), which is always disabled in Stop mode. In this case setting the bit has no effect in Stop mode.

7.4.26 Peripherals independent clock configuration register (RCC_CCIPR)

Address: 0x88

Reset value: 0x0000 0000

Access: no wait states, word, half-word and byte access

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---------------|---------------|-----------------|---------------|---------------|---------------|----------------|---------------|----|----|----|----|----|----|----|----|
| ADC34SEL[1:0] | ADC12SEL[1:0] | CLK48SEL[1:0] | FDCANSEL[1:0] | I2C23SEL[1:0] | SA1SEL[1:0] | LPTIM1SEL[1:0] | I2C3SEL[1:0] | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I2C2SEL[1:0] | I2C1SEL[1:0] | LPUART1SEL[1:0] | UART6SEL[1:0] | UART4SEL[1:0] | UART3SEL[1:0] | UART2SEL[1:0] | UART1SEL[1:0] | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Bits 31:30 ADC345SEL[1:0]: ADC3/4/5 clock source selection

These bits are set and cleared by software to select the clock source used by the ADC 345 interface.

- 00: No clock selected
- 01: PLL "P" clock selected as ADC 345 clock
- 10: System clock selected as ADC 3/4/5 clock
- 11: Reserved.

Bits 29:28 ADC12SEL[1:0]: ADC 1/2 clock source selection

These bits are set and cleared by software to select the clock source used by the ADC interface.

- 00: No clock selected
- 01: PLL "P" clock selected as ADC 1/2 clock
- 10: System clock selected as ADC 1/2 clock
- 11: Reserved

Bits 27:26 CLK48SEL[1:0]: 48 MHz clock source selection

These bits are set and cleared by software to select the 48 MHz clock source used by USB device FS and RNG.

- 00: HSI48 clock selected as 48 MHz clock
- 01: Reserved
- 10: PLL "Q" clock (PLL48M1CLK) selected as 48 MHz clock
- 11: Reserved, must be kept at reset value

Bits 25:24 FDCANSEL[1:0]: clock source selection

These bits are set and cleared by software to select the FDCAN clock source.

- 00: HSE clock selected as FDCAN clock
- 01: PLL "Q" clock selected as FDCAN clock
- 10: PCLK clock selected as FDCAN clock
- 11: Reserved, must be kept at reset value.

Bits 23:22 I2S23SEL[1:0]: clock source selection

These bits are set and cleared by software to select the I2S23 clock source.

- 00: System clock selected as I2S23 clock
- 01: PLL "Q" clock selected as I2S23 clock
- 10: Clock provided on I2S_CKIN pin selected as I2S23 clock
- 11: HSI16 clock selected as I2S23 clock.

Bits 21:20 SAI1SEL[1:0]: clock source selection

These bits are set and cleared by software to select the SAI clock source.

- 00: System clock selected as SAI clock
- 01: PLL "Q" clock selected as SAI clock
- 10: Clock provided on I2S_CKIN pin selected as SAI clock
- 11: HSI16 clock selected as SAI clock

Bits 19:18 LPTIM1SEL[1:0]: Low power timer 1 clock source selection

These bits are set and cleared by software to select the LPTIM1 clock source.

- 00: PCLK selected as LPTIM1 clock
- 01: LSI clock selected as LPTIM1 clock
- 10: HSI16 clock selected as LPTIM1 clock
- 11: LSE clock selected as LPTIM1 clock

Bits 17:16 I2C3SEL[1:0]: I2C3 clock source selection

These bits are set and cleared by software to select the I2C3 clock source.

- 00: PCLK selected as I2C3 clock
- 01: System clock (SYSLK) selected as I2C3 clock
- 10: HSI16 clock selected as I2C3 clock
- 11: Reserved

Bits 15:14 I2C2SEL[1:0]: I2C2 clock source selection

These bits are set and cleared by software to select the I2C2 clock source.

- 00: PCLK selected as I2C2 clock
- 01: System clock (SYSLK) selected as I2C2 clock
- 10: HSI16 clock selected as I2C2 clock
- 11: Reserved

Bits 13:12 I2C1SEL[1:0]: I2C1 clock source selection

These bits are set and cleared by software to select the I2C1 clock source.

- 00: PCLK selected as I2C1 clock
- 01: System clock (SYSLK) selected as I2C1 clock
- 10: HSI16 clock selected as I2C1 clock
- 11: Reserved

Bits 11:10 LPUART1SEL[1:0]: LPUART1 clock source selection

These bits are set and cleared by software to select the LPUART1 clock source.

- 00: PCLK selected as LPUART1 clock
- 01: System clock (SYSLK) selected as LPUART1 clock
- 10: HSI16 clock selected as LPUART1 clock
- 11: LSE clock selected as LPUART1 clock

Bits 9:8 UART5SEL[1:0]: UART5 clock source selection

These bits are set and cleared by software to select the UART5 clock source.

- 00: PCLK selected as UART5 clock
- 01: System clock (SYSLK) selected as UART5 clock
- 10: HSI16 clock selected as UART5 clock
- 11: LSE clock selected as UART5 clock

Bits 7:6 UART4SEL[1:0]: UART4 clock source selection

This bit is set and cleared by software to select the UART4 clock source.

- 00: PCLK selected as UART4 clock
- 01: System clock (SYSLK) selected as UART4 clock
- 10: HSI16 clock selected as UART4 clock
- 11: LSE clock selected as UART4 clock

Bits 5:4 USART3SEL[1:0]: USART3 clock source selection

This bit is set and cleared by software to select the USART3 clock source.

00: PCLK selected as USART3 clock

01: System clock (SYSLK) selected as USART3 clock

10: HSI16 clock selected as USART3 clock

11: LSE clock selected as USART3 clock

Bits 3:2 USART2SEL[1:0]: USART2 clock source selection

This bit is set and cleared by software to select the USART2 clock source.

00: PCLK selected as USART2 clock

01: System clock (SYSLK) selected as USART2 clock

10: HSI16 clock selected as USART2 clock

11: LSE clock selected as USART2 clock

Bits 1:0 USART1SEL[1:0]: USART1 clock source selection

This bit is set and cleared by software to select the USART1 clock source.

00: PCLK selected as USART1 clock

01: System clock (SYSLK) selected as USART1 clock

10: HSI16 clock selected as USART1 clock

11: LSE clock selected as USART1 clock

7.4.27 RTC domain control register (RCC_BDCR)

Address offset: 0x90

Reset value: 0x0000 0000

Reset by RTC domain Reset, except LSCOSEL, LSCOEN and BDRST which are reset only by RTC domain power-on reset.

Access: 0 ≤ wait state ≤ 3, word, half-word and byte access

Wait states are inserted in case of successive accesses to this register.

Note: The bits of the [RTC domain control register \(RCC_BDCR\)](#) are outside of the V_{CORE} domain. As a result, after Reset, these bits are write-protected and the DBP bit in the [Section 6.4.1: Power control register 1 \(PWR_CR1\)](#) has to be set before these can be modified. Refer to [Section 6.1.3: Battery backup domain on page 232](#) for further information. These bits (except LSCOSEL, LSCOEN and BDRST) are only reset after a RTC domain Reset (see [Section 7.1.3: RTC domain reset](#)). Any internal or external Reset will not have any effect on these bits.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|--------|------|------|------|------|------|-------------|---------|----------|-----------|-------------|---------|---------|-------|------|-------|
| Res. | Res. | Res. | Res. | Res. | Res. | LSCO SEL | LSCO EN | Res. | Res. | Res. | Res. | Res. | Res. | Res. | BDRST |
| | | | | | | rw | rw | | | | | | | | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RTC EN | Res. | Res. | Res. | Res. | Res. | RTCSEL[1:0] | Res. | LSE C8SD | LSE C8SDN | LSEDRM[1:0] | LSE BYP | LSE RDY | LSEON | | |
| rw | | | | | | rw | rw | r | rw | rw | rw | r | rw | | |

Bits 31:26 Reserved, must be kept at reset value.

Bit 25 **LSCOSEL**: Low speed clock output selection

Set and cleared by software.

0: LSI clock selected

1: LSE clock selected

Bit 24 **LSCOEN**: Low speed clock output enable

Set and cleared by software.

0: Low speed clock output (LSCO) disable

1: Low speed clock output (LSCO) enable

Bits 23:17 Reserved, must be kept at reset value.

Bit 16 **BDRST**: RTC domain software reset

Set and cleared by software.

0: Reset not activated

1: Reset the entire RTC domain

Bit 15 **RTCEN**: RTC clock enable

Set and cleared by software.

0: RTC clock disabled

1: RTC clock enabled

Bits 14:10 Reserved, must be kept at reset value.

Bits 9:8 **RTCSEL[1:0]**: RTC clock source selection

Set by software to select the clock source for the RTC. Once the RTC clock source has been selected, it cannot be changed anymore unless the RTC domain is reset, or unless a failure is detected on LSE (LSECSSD is set). The BDRST bit can be used to reset them.

00: No clock

01: LSE oscillator clock used as RTC clock

10: LSI oscillator clock used as RTC clock

11: HSE oscillator clock divided by 32 used as RTC clock

Bit 7 Reserved, must be kept at reset value.

Bit 6 **LSECSSD**: CSS on LSE failure Detection

Set by hardware to indicate when a failure has been detected by the Clock Security System on the external 32 kHz oscillator (LSE).

0: No failure detected on LSE (32 kHz oscillator)

1: Failure detected on LSE (32 kHz oscillator)

Bit 5 **LSECSSON**: CSS on LSE enable

Set by software to enable the Clock Security System on LSE (32 kHz oscillator).

LSECSSON must be enabled after the LSE oscillator is enabled (LSEON bit enabled) and ready (LSERDY flag set by hardware), and after the RTCSEL bit is selected.

Once enabled this bit cannot be disabled, except after a LSE failure detection (LSECSSD = 1). In that case the software MUST disable the LSECSSON bit.

0: CSS on LSE (32 kHz external oscillator) OFF

1: CSS on LSE (32 kHz external oscillator) ON

Bits 4:3 **LSEDRV[1:0]**: LSE oscillator drive capability

Set by software to modulate the LSE oscillator's drive capability.

00: 'Xtal mode' lower driving capability

01: 'Xtal mode' medium low driving capability

10: 'Xtal mode' medium high driving capability

11: 'Xtal mode' higher driving capability

The oscillator is in Xtal mode when it is not in bypass mode.

Bit 2 **LSEBYP**: LSE oscillator bypass

Set and cleared by software to bypass oscillator in debug mode. This bit can be written only when the external 32 kHz oscillator is disabled (LSEON=0 and LSERDY=0).

0: LSE oscillator not bypassed

1: LSE oscillator bypassed

Bit 1 **LSERDY**: LSE oscillator ready

Set and cleared by hardware to indicate when the external 32 kHz oscillator is stable. After the LSEON bit is cleared, LSERDY goes low after 6 external low-speed oscillator clock cycles.

0: LSE oscillator not ready

1: LSE oscillator ready

Bit 0 **LSEON**: LSE oscillator enable

Set and cleared by software.

0: LSE oscillator OFF

1: LSE oscillator ON

7.4.28 Control/status register (RCC_CSR)

Address: 0x94

Reset value: 0x0C00 0000

Reset by system Reset, except reset flags by power Reset only.

Access: 0 ≤ wait state ≤ 3, word, half-word and byte access

Wait states are inserted in case of successive accesses to this register.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|--------------|--------------|--------------|-------------|-------------|-------------|-------------|------|------|------|------|------|------|------|------------|-----------|
| LPWR RSTF | WWDG RSTF | IWDG RSTF | SFT RSTF | BDR RSTF | PIN RSTF | OBL RSTF | Res. | RMVF | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| r | r | r | r | r | r | r | | rw | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | LSE RDY | LSE ON |
| | | | | | | | | | | | | | | r | rw |

Bit 31 LPWRRSTF: Low-power reset flag

Set by hardware when a reset occurs due to Illegal Stop, Standby or Shutdown mode entry.
Cleared by writing to the RMVF bit.

- 0: No illegal mode reset occurred
- 1: Illegal mode reset occurred

Bit 30 WWDGRSTF: Window watchdog reset flag

Set by hardware when a window watchdog reset occurs.
Cleared by writing to the RMVF bit.

- 0: No window watchdog reset occurred
- 1: Window watchdog reset occurred

Bit 29 IWDGRSTF: Independent window watchdog reset flag

Set by hardware when an independent watchdog reset domain occurs.
Cleared by writing to the RMVF bit.

- 0: No independent watchdog reset occurred
- 1: Independent watchdog reset occurred

Bit 28 SFRSTF: Software reset flag

Set by hardware when a software reset occurs.
Cleared by writing to the RMVF bit.

- 0: No software reset occurred
- 1: Software reset occurred

Bit 27 BORRSTF: BOR flag

Set by hardware when a BOR occurs.
Cleared by writing to the RMVF bit.

- 0: No BOR occurred
- 1: BOR occurred

Bit 26 PINRSTF: Pin reset flag

Set by hardware when a reset from the NRST pin occurs.
Cleared by writing to the RMVF bit.

- 0: No reset from NRST pin occurred
- 1: Reset from NRST pin occurred

Bit 25 OBLRSTF: Option byte loader reset flag

Set by hardware when a reset from the Option Byte loading occurs.
Cleared by writing to the RMVF bit.

- 0: No reset from Option Byte loading occurred
- 1: Reset from Option Byte loading occurred

Bit 24 Reserved, must be kept at reset value.**Bit 23 RMVF:** Remove reset flag

Set by software to clear the reset flags.

- 0: No effect
- 1: Clear the reset flags