

PSOC™ control C3: CPU subsystem

Customer training workshop (CTW)

February 2026

public

Table of contents

1 Target products 3

2 CPU subsystem 5

3 Register summary 13

4 Interrupts 25

5 Memories 40

6 Clock 55

7 Lab: Getting started with an empty application 65

8 Revision history 67

Copyright © Infineon Technologies AG 2026. All rights reserved.

Table of contents

1 **Target products** **3**

2 CPU subsystem 5

3 Register summary 13

4 Interrupts 25

5 Memories 40

6 Clock 55

7 Lab: Getting started with an empty application 65

8 Revision history 67

Copyright © Infineon Technologies AG 2026. All rights reserved.

Target products

– Target product list for this training material:

Family category	Series	Code flash memory size
PSOC™ Control C3 Entry Line	PSC3P2xD	Up to 256 KB
PSOC™ Control C3 Entry Line	PSC3M3xD	Up to 256 KB
PSOC™ Control C3 Main Line	PSC3P5xD	Up to 256 KB
PSOC™ Control C3 Main Line	PSC3M5xD	Up to 256 KB

Copyright © Infineon Technologies AG 2026. All rights reserved.

Table of contents

1 Target products 3

2 **CPU subsystem** **5**

3 Register summary 13

4 Interrupts 25

5 Memories 40

6 Clock 55

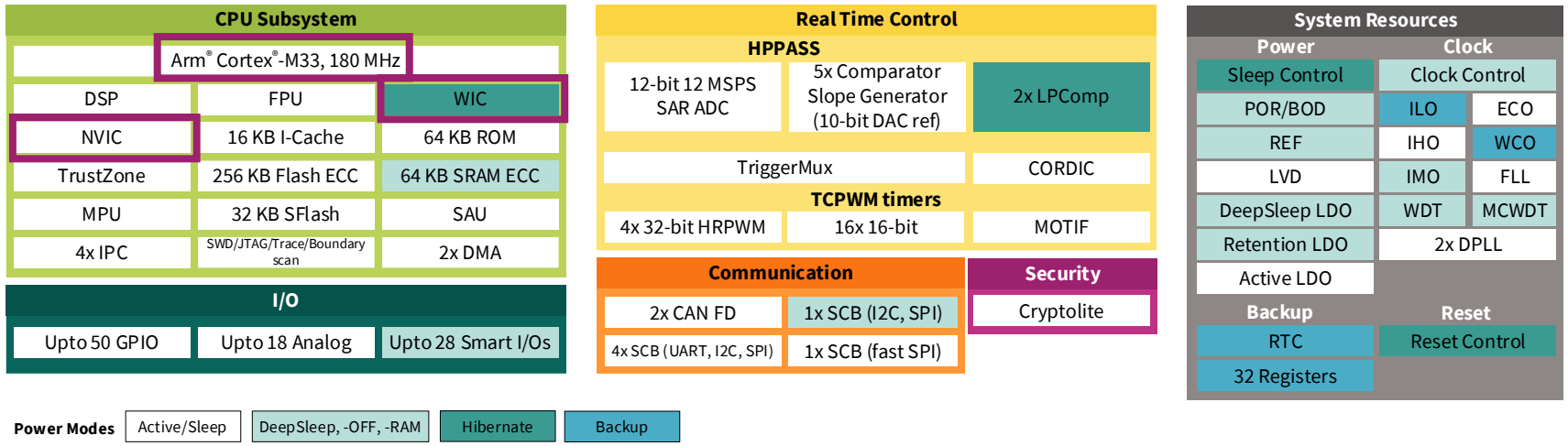
7 Lab: Getting started with an empty application 65

8 Revision history 67

Copyright © Infineon Technologies AG 2026. All rights reserved.

CPU subsystem

Overview



Features

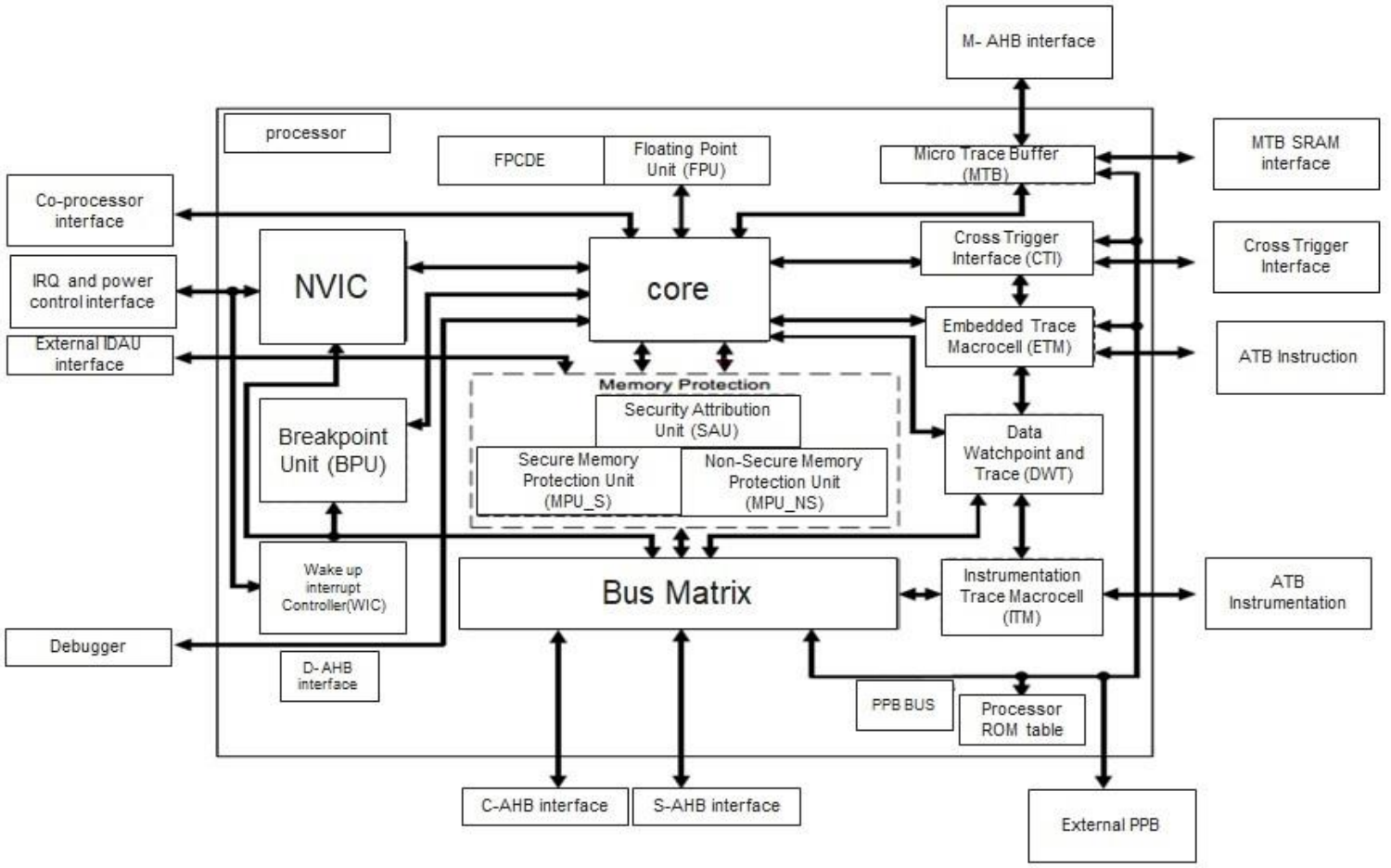
- Sleep modes with integrated wait for interrupt (WFI) and wait for event (WFE), plus sleep on exit capability
- TrustZone® Arm® v8-M Security Extension supporting Secure and Non-secure states, including the secure arbitration unit (SAU) and a Memory protection unit (MPU)
- Secure programming environment (SPE) for handling of secure and non-secure application SW
- Built-in instruction cache

Benefit

- Industry standard core with sophisticated low-power capabilities and wake up mechanisms resulting in less interrupts and faster response times.
- All optional enhanced security features integrated in the MCU core
- Improved real-time performance by reducing instruction access latency for time-critical blocks of code

Copyright © Infineon Technologies AG 2026. All rights reserved.

Architecture



Copyright © Infineon Technologies AG 2026. All rights reserved.

Architecture (contd.)

- Core: Based on v8-M architecture, includes support for TrustZone® technology
- NVIC and WIC:
  - NVIC supports efficient and flexible interrupt management with low-latency interrupt handling and prioritisation
  - WIC is designed to process wake-up from low power mode in response to interrupts
- Memory protection unit (MPU), which allows for the definition of memory regions with different access permissions, enhancing security by preventing unauthorised access to critical memory areas
- Floating-point unit (FPU), single precision FPU to accelerate mathematical computations in applications that require higher numerical precision
- Bus matrix: Provides flexible and efficient interconnection between the core, memory, and peripherals

Copyright © Infineon Technologies AG 2026. All rights reserved.

Mode transition

CPU supports two operating modes and two privilege levels:

- Operating mode
  - Thread mode executes application software
  - Handler mode handles exceptions
- Privilege levels
  - Unprivileged: Software has limited access to MSR/MRS instructions (uses CPS instructions), system timer, NVIC, system control block, and memory/peripherals
  - Privileged: Software can use all instructions and has access to all resources

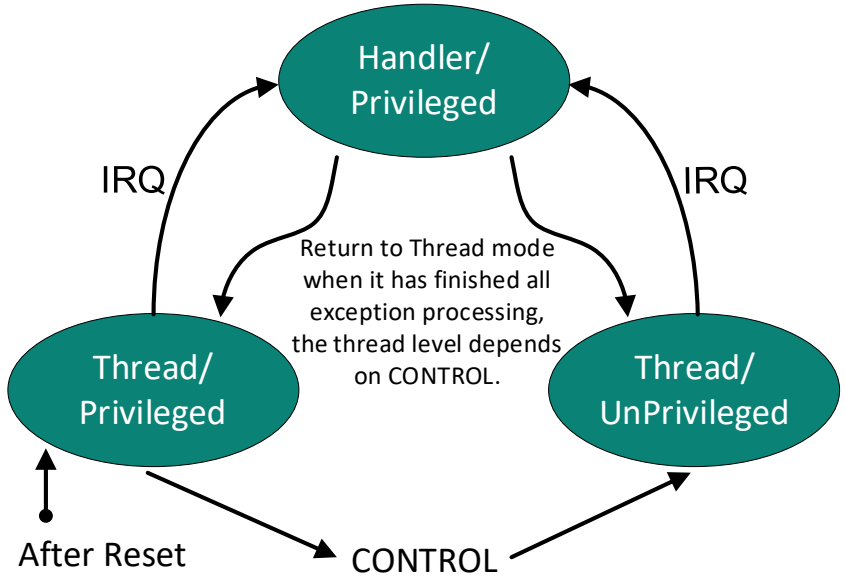
Hint bar

- See the reference manual, operating modes and privilege levels (section 2.3), for additional details
- Arm® reference material at: infocenter.arm.com
- Move to the system coprocessor register from the Arm® register (MSR)
- Move the contents of the program status register to a general-purpose register (MRS)
- Change processor state (CPS)

Copyright © Infineon Technologies AG 2026. All rights reserved.

Mode transition (contd.)

In handler mode, SW is always privileged.



- Transition from Thread/privileged to Thread/unprivileged controlled by the CONTROL register
  - The CONTROL register is a CPU-specific register. It defines privileged/unprivileged, stack pointer, and FPU extension
- Transition from Thread/unprivileged to Thread/privileged via SVC
  - Supervisor calls are used to request privileged operations

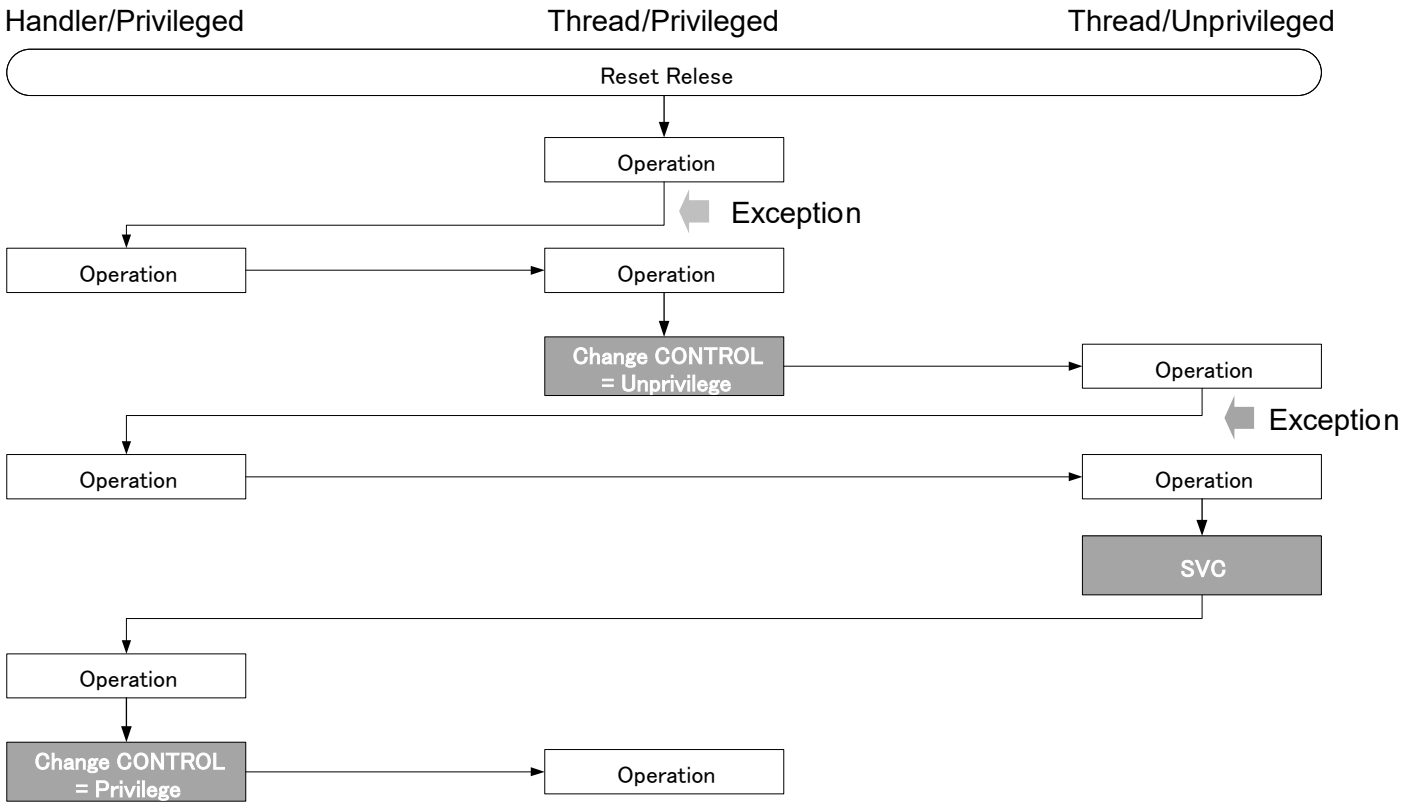
Hint bar

- See the reference manual, operating modes and privilege levels (section 2.3), for additional details
- Arm® reference material at: infocenter.arm.com
- Move to the system coprocessor register from the Arm® register (MSR)
- Move the contents of program status register to a general-purpose register (MRS)
- Change processor state (CPS)

Copyright © Infineon Technologies AG 2026. All rights reserved.

Operation modes transition

Use case: Changing privileged/unprivileged mode



Hint bar

- Arm® provides additional reference material on their webpage at: infocenter.arm.com
- The required privilege level for core register access is discussed in the following slides

Copyright © Infineon Technologies AG 2026. All rights reserved.

Table of contents

1 Target products 3

2 CPU subsystem 5

3 **Register summary** **13**

4 Interrupts 25

5 Memories 40

6 Clock 55

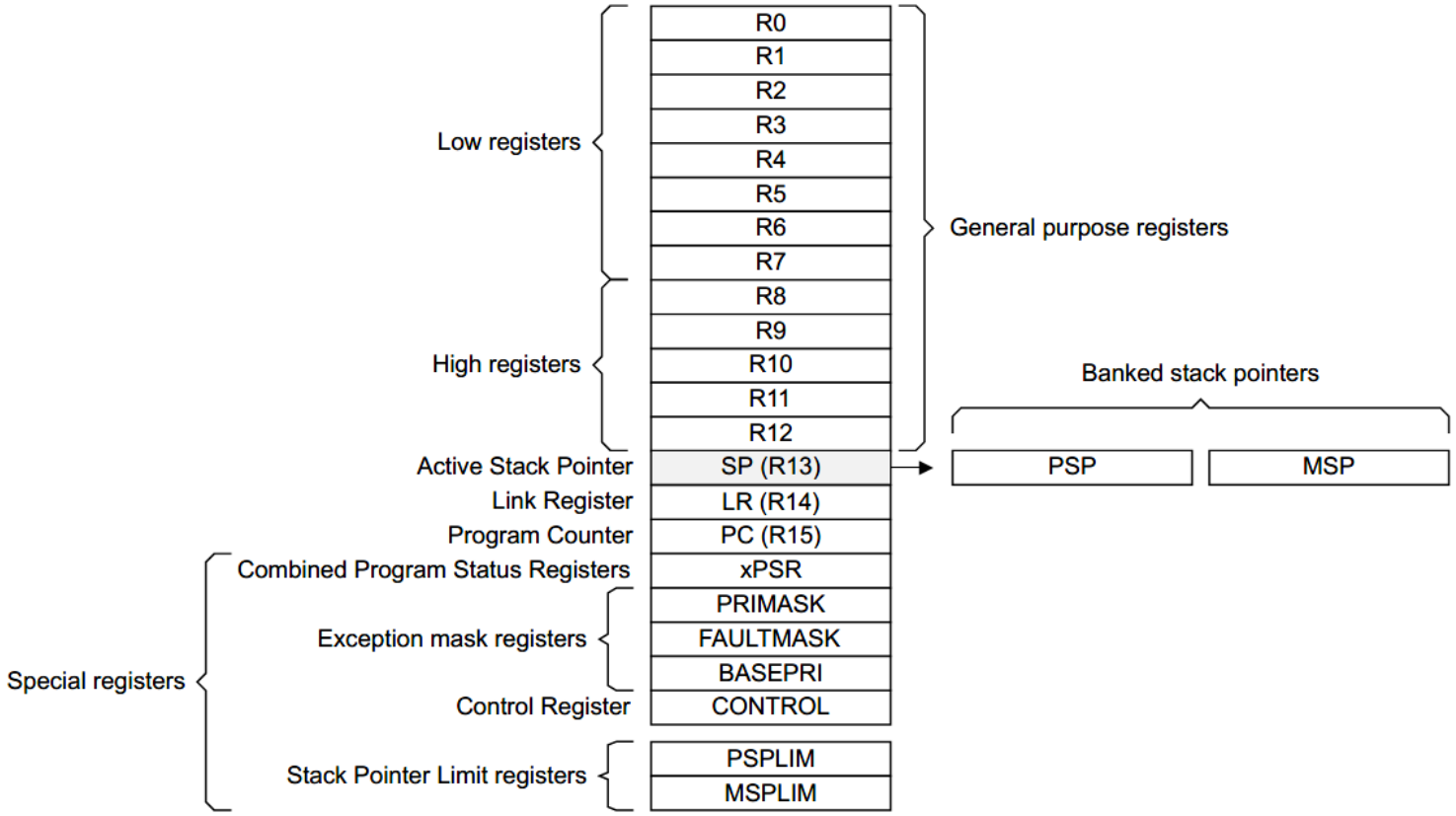
7 Lab: Getting started with an empty application 65

8 Revision history 67

Copyright © Infineon Technologies AG 2026. All rights reserved.

Register summary

Arm® Cortex® -M33 core registers



Hint bar

- See the architecture reference manual, Registers summary (section 2.5), for additional details
- Note that the following slides describe the core register details. Click here to proceed further

Copyright © Infineon Technologies AG 2026. All rights reserved.

Register details (1/8)

– The CM33 CPU has the following 32-bit core registers

Name	Access type	Required privilege	Reset value	Description
R0-R12	RW	Privileged/unprivileged	unknown	R0-R12 are general purpose registers for data operations
MSP (R13) (Main stack pointer)	RW	Privileged/unprivileged	Soft reset to the value retrieved by the reset handler	Control register indicates the SP to use (MSP or PSP). Secure and Non-secure state of SP is available when security extension is included
PSP (R13) (Process stack pointer)	RW	Privileged/unprivileged	unknown	

Hint bar

See the architecture reference manual, Registers summary (section 2.5), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Register details (2/8)

– MSPLIM and PSPLIM

Name	Access type	Required privilege	Reset value	Description
MSPLIM	RW	Privileged	0x00000000	The stack limit registers limit the extent to which the MSP and PSP registers can descend respectively. Secure and Non-secure state of SPLIM is available when security ext. is included
PSPLIM	RW	Privileged	0x00000000	

Hint bar

See the architecture reference manual, Registers summary (section 2.5), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Register details (3/8)

– LR and PC

Name	Access type	Required privilege	Reset value	Description
LR (R14)	RW	Privileged/unprivileged	unknown	The link register (LR) is register R14. It stores the return information for subroutines, function calls, and exceptions
PC (R15)	RW	Privileged/unprivileged	Soft reset to the value retrieved by the reset handler	The program counter (PC) is register R15. It contains the current program address

Hint bar

See the architecture reference manual, Registers summary (section 2.5), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Register details (4/8)

– xPSR

Name	Access type	Required privilege	Reset value	Description
xPSR	RW	Privileged/unprivileged	Bit[24] is the T-bit and is loaded from bit[0] of the reset vector. All other bits are reset to ‘0’	The Program Status Register (PSR) combines: Application Program Status register (APSR), Interrupt Program Status Register (IPSR), Execution Program Status Register (EPSR). These registers provide different views of the PSR

Hint bar

See the architecture reference manual, Registers summary (section 2.5), for additional details

Copyright © Infineon Technologies AG 2026. All rights reserved.

Register details (5/8)

– APSR and IPSR

Name	Access type	Required privilege	Reset value	Description
APSR	RW	Privileged/unprivileged	unknown	The APSR contains the current state of the condition flags from previous instruction executions
IPSR	RO	Privileged	0x00000000	The IPSR contains the exception number of the current ISR

Hint bar

See the architecture reference manual, Registers summary (section 2.5), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Register details (6/8)

– EPSR

Name	Access type	Required privilege	Reset value	Description
EPSR	RO	Privileged	Bit[24] is the T-bit and is loaded from bit[0] of the reset vector. All other bits are reset to ‘0’	The EPSR contains the thumb state bit and the execution state bits for the If-then (IT) instruction, and interruptible-continuable instruction (ICI) field for an interrupted load multiple or store multiple instruction

– Use case for ICI field: When multiple cycle instruction for example load-multiple LDM is interrupted by an exception, the ICI bits help the processor keep track of the state of that instruction so it can be resumed correctly after the interrupt is handled
– The IT field is used to manage the execution of up to four instruction conditionally

Hint bar

See the architecture reference manual, Registers summary (section 2.5), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Register details (7/8)

Name	Access type	Required privilege	Reset value	Description
PRIMASK	RW	Privileged	0x00000000	The PRIMASK Register prevents activation of exceptions with configurable priority. Secure and Non-secure state of PRIMASK is available when security ext. is included
BASEPRI	RW	Privileged	0x00000000	The BASEPRI Register defines the minimum priority for exception processing

Hint bar

See the architecture reference manual, Registers summary (section 2.5), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Register details (8/8)

Name	Access type	Required privilege	Reset value	Description
FAULTMASK	RW	Privileged	0x00000000	The FAULTMASK Register prevents activation of all exceptions except for Non-maskable interrupt (NMI) and optionally secure HardFault
CONTROL	RW	Privileged	0x00000000	The CONTROL Register controls the stack used, and optionally the privilege level, when the processor is in the Thread mode

Hint bar

- See the architecture reference manual, Registers summary (section 2.5), for additional details
- Secure and Non-secure state of FAULTMASK and CONTROL is available when Arm® V8-M security extension is included

Copyright © Infineon Technologies AG 2026. All rights reserved.

Wait state lookup tables

– The following table gives number of wait states required for memory access operation based on memory type in PSOC™ Control C3

Parameter	Symbol	Value (min)	Value (typ)	Value (max)	Test condition
Wait states	$M33_{cache}$	–	–	0	–
Wait states from CM33	$M33_{SRAM}$	–	0	3(With ECC enabled)	–
Wait states from flash via C-bus (cached)	$M33_{Flash}$	0 (on cache hit) 3 (on cache miss)	–	9	Depending on cache hit, pre-fetch and opcode
Wait states from flash via S-bus	$M33_{Flash}$	–	–	9	–

Hint bar

– See the architecture reference manual, Wait state lookup tables (section 2.2.2.1), for additional details

– This table is applicable for all power modes (ULP/LP/OD)

Copyright © Infineon Technologies AG 2026. All rights reserved.

Table of contents

1 Target products 3

2 CPU subsystem 5

3 Register summary 13

4 **Interrupts** **25**

5 Memories 40

6 Clock 55

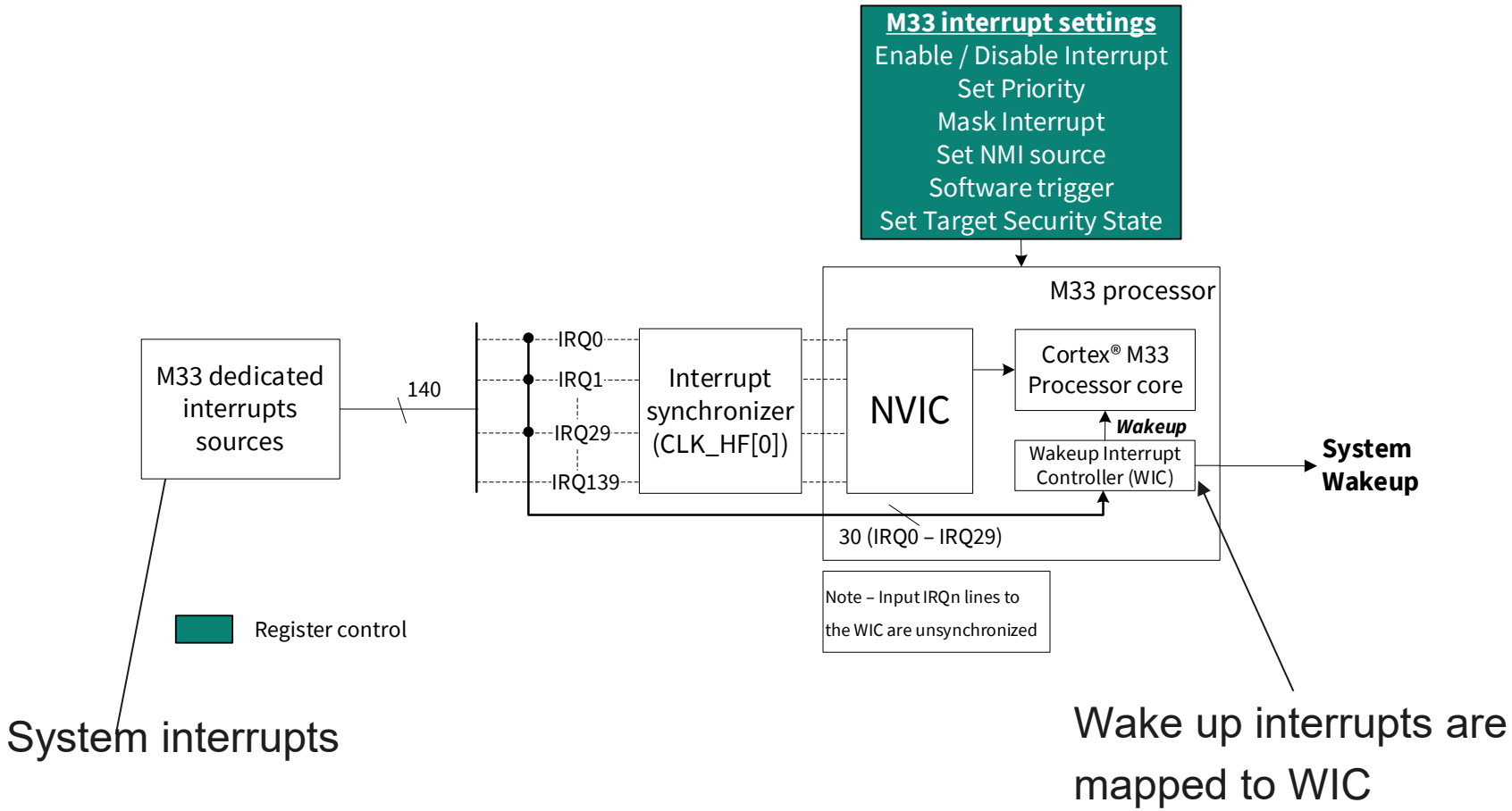
7 Lab: Getting started with an empty application 65

8 Revision history 67

Copyright © Infineon Technologies AG 2026. All rights reserved.

Interrupts

Architecture- Block diagram



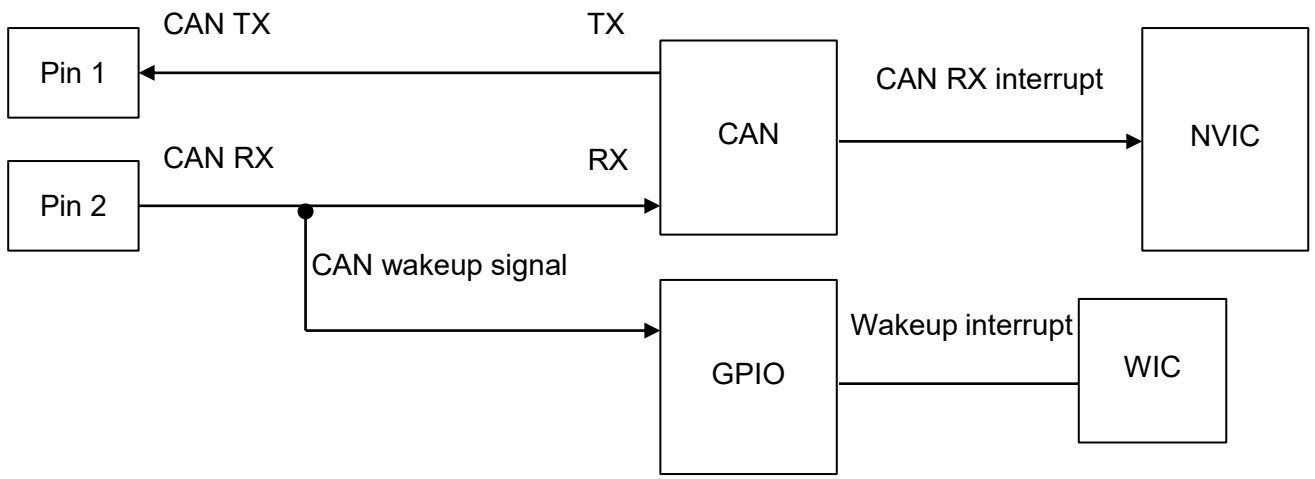
Hint bar

See the architecture reference manual, Interrupt architecture (section 7.2), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Interrupt sources

- System interrupts are routed to NVIC
  - Originate from peripherals
  - Include wakeup interrupts
- Wakeup interrupts are routed to WIC
  - Wakes CPU up from Deep Sleep mode
- For example, GPIO event can be used as a CAN wakeup interrupt

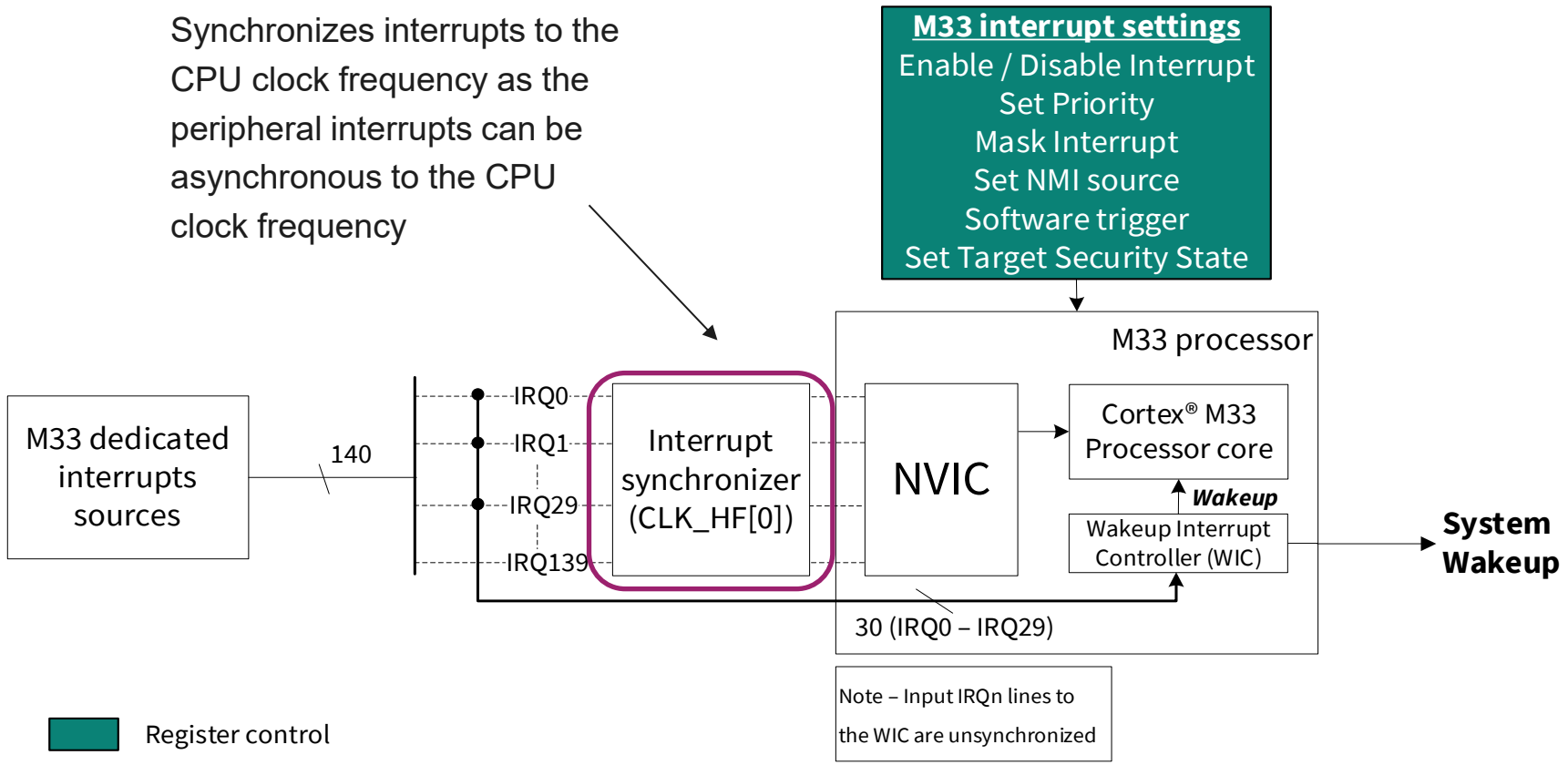


Hint bar

See the architecture reference manual, Interrupt architecture (section 7.2), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Interrupt synchronizer

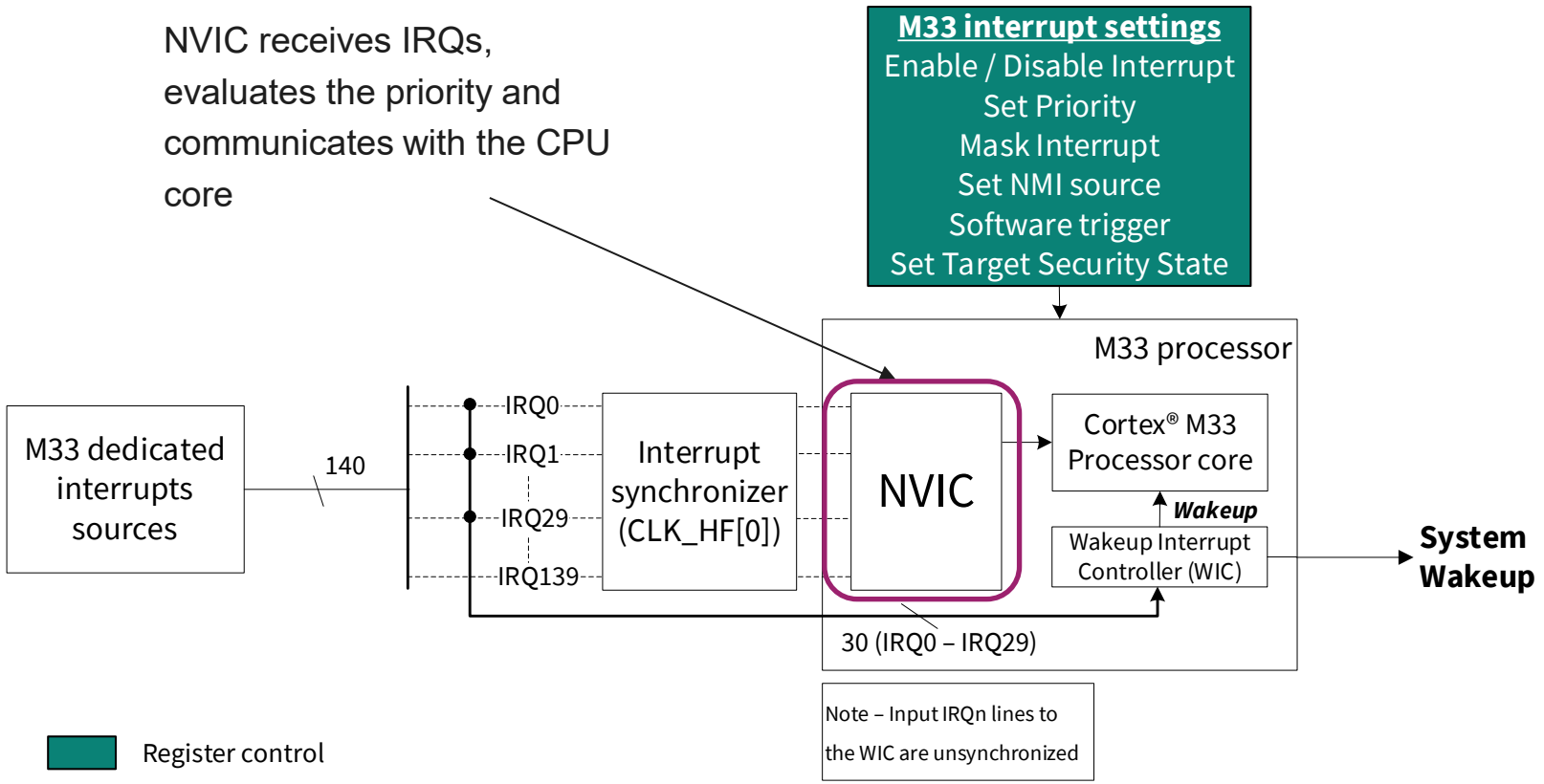


Hint bar

See the architecture reference manual, Interrupt architecture (section 7.2), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Nested vector interrupt controller (NVIC)



Hint bar

See the architecture reference manual, Interrupt architecture (section 7.2), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Exception vector table

– The exception vector tables store the entry point addresses for all exception handlers in the Cortex®-M33 core

Exception #	Exception	Exception priority	Vector address
–	Initial stack pointer value	Not Applicable (N/A)	Start_Address = 0x0000 , Start_Address =CM33_S_VECTOR_TABLE _BASE or CM33's VTOR_NS
1	Reset	-4	Start_Address + 0x0004
2	non-maskable interrupt	-2	Start_Address + 0x0008
3	HardFault	(-1 or -3)	Start_Address + 0x000C
4	Memory Management	Configurable (0-7)	Start_Address + 0x0010
5	Bus Fault	Configurable (0-7)	Start_Address + 0x0014
6	Usage Fault	Configurable (0-7)	Start_Address + 0x0018

Hint bar

See the architecture reference manual, Exception vector table (section 7.3.3), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Exception vector table (contd.)

Exception #	Exception	Exception priority	Vector address
7	Secure Fault	Configurable (0-7)	Start_Address + 0x001C
8-10	Reserved	-4	…
11	Supervisory Call	-2	Start_Address + 0x002C
12	Debug Monitor	(-1 or -3)	Start_Address + 0x0030
13	Reserved	Configurable (0-7)	…
14	Pending Supervisory	Configurable (0-7)	Start_Address + 0x0038
15	Sys Tick Timer	Configurable (0-7)	Start_Address + 0x003C
16	External interrupt (IRQ0)	Configurable (0-7)	Start_Address + 0x0040
…	…	…	…
155	External interrupt IRQ(139)	Configurable (0-7)	Start_Address + 0x022C

Hint bar

- See the architecture reference manual, Exception vector table (section 7.3.3), for additional details
- See the architecture reference manual, Interrupt sources (section 7.5), for additional details

Copyright © Infineon Technologies AG 2026. All rights reserved.

Enabling and disabling interrupts

- The NVICs of CM33 core provide registers to individually enable and disable the CPU interrupts in software
- CM33 interrupts are enabled and disabled using the **ISER** and **ICER**
- If an interrupt is not enabled, the NVIC will not process the interrupt requests on that interrupt line
- CM33 provide additional registers to control the activation of exceptions/interrupts based on their priority
  - PRIMASK: Prevent activation of exceptions having configurable priority
  - FAULTMASK: Prevent activation of all exceptions other than NMI
  - BASEPRI: Prevent activation of exceptions having the same or lower priority than the BASEPRI
- API to enable interrupt
  - NVIC_EnableIRQ(IRQ_number)

Hint bar

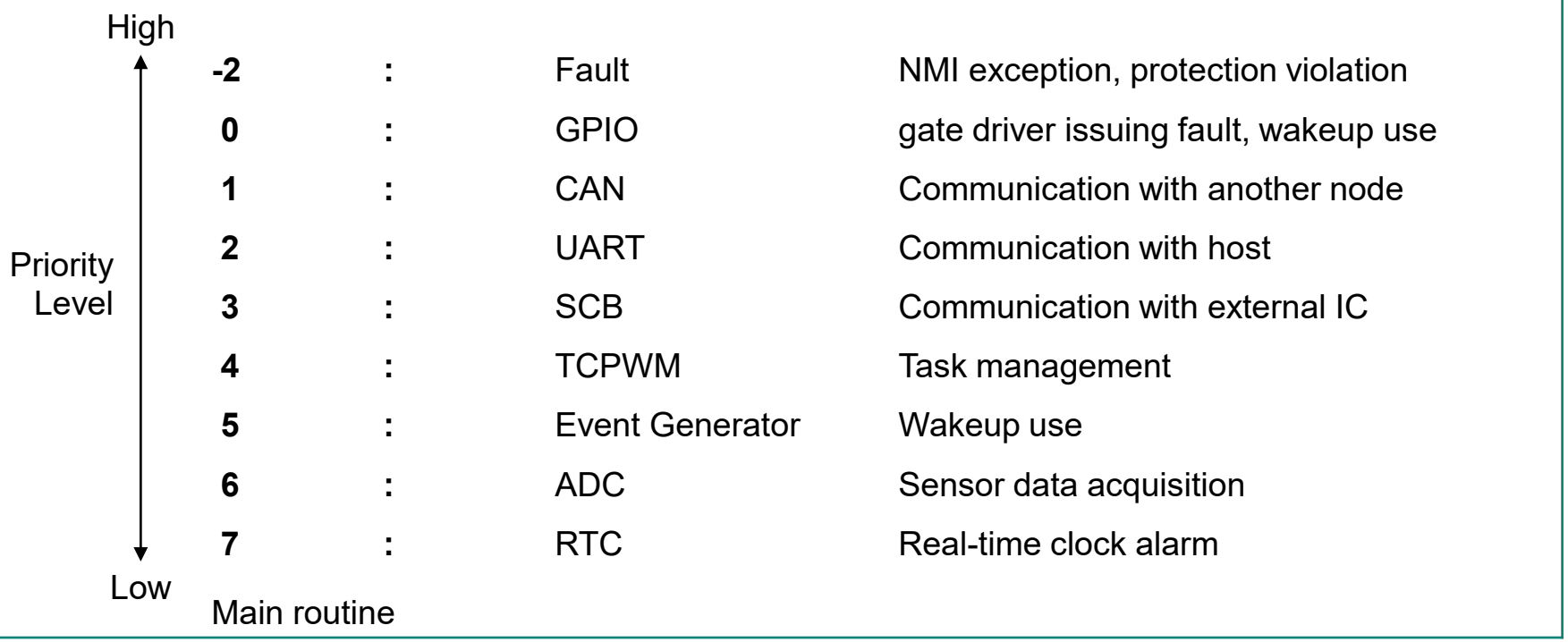
See the architecture reference manual, Enabling and disabling interrupts (section 7.8), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

CPU interrupt priority

– The priority of each interrupt can be configured to eight levels for Cortex® M33 where a lower number represents a higher priority

Example of interrupt priority levels setting in a pseudo-application



– API to set priority to an interrupt: `NVIC_SetPriority(irq_number, Priority)`

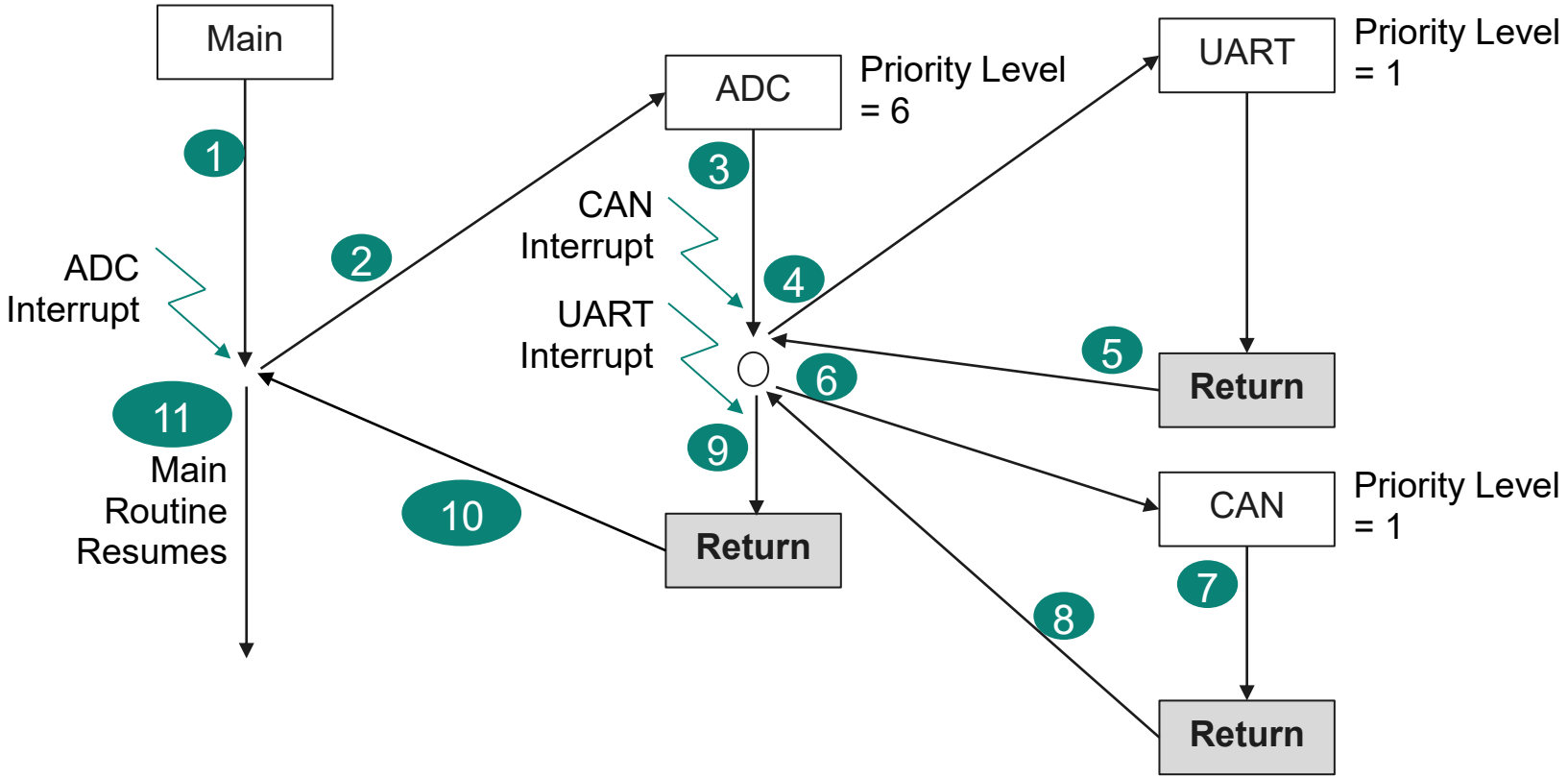
Hint bar

See the architecture reference manual, Interrupt/exception priority (section 7.7), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Nested interrupts

- Depending on the interrupt priority level, nested interrupts are possible
- Use case for nested interrupts



Hint bar

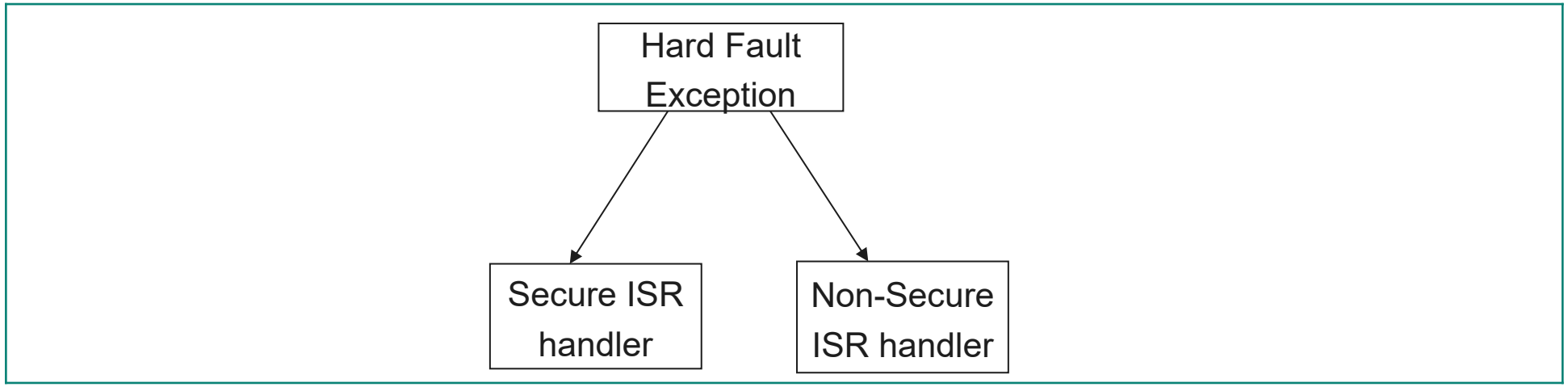
See the architecture reference manual, Interrupt/exception priority (section 7.7), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Interrupt/exception target security state and banking

- For the Cortex® -M33 CPU, exceptions and interrupts can target the secure or non-secure ISR handler
- If an exception is said to be banked, this means that the exception handler and priority of the exception are duplicated and have a secure and non-secure version
- If an exception is said to not be banked, this means the exception can be configured to only target the secure or non-secure handler

Banked exception



- When Hard Fault is triggered by secure application the Secure ISR handler handles the Hardfault exception

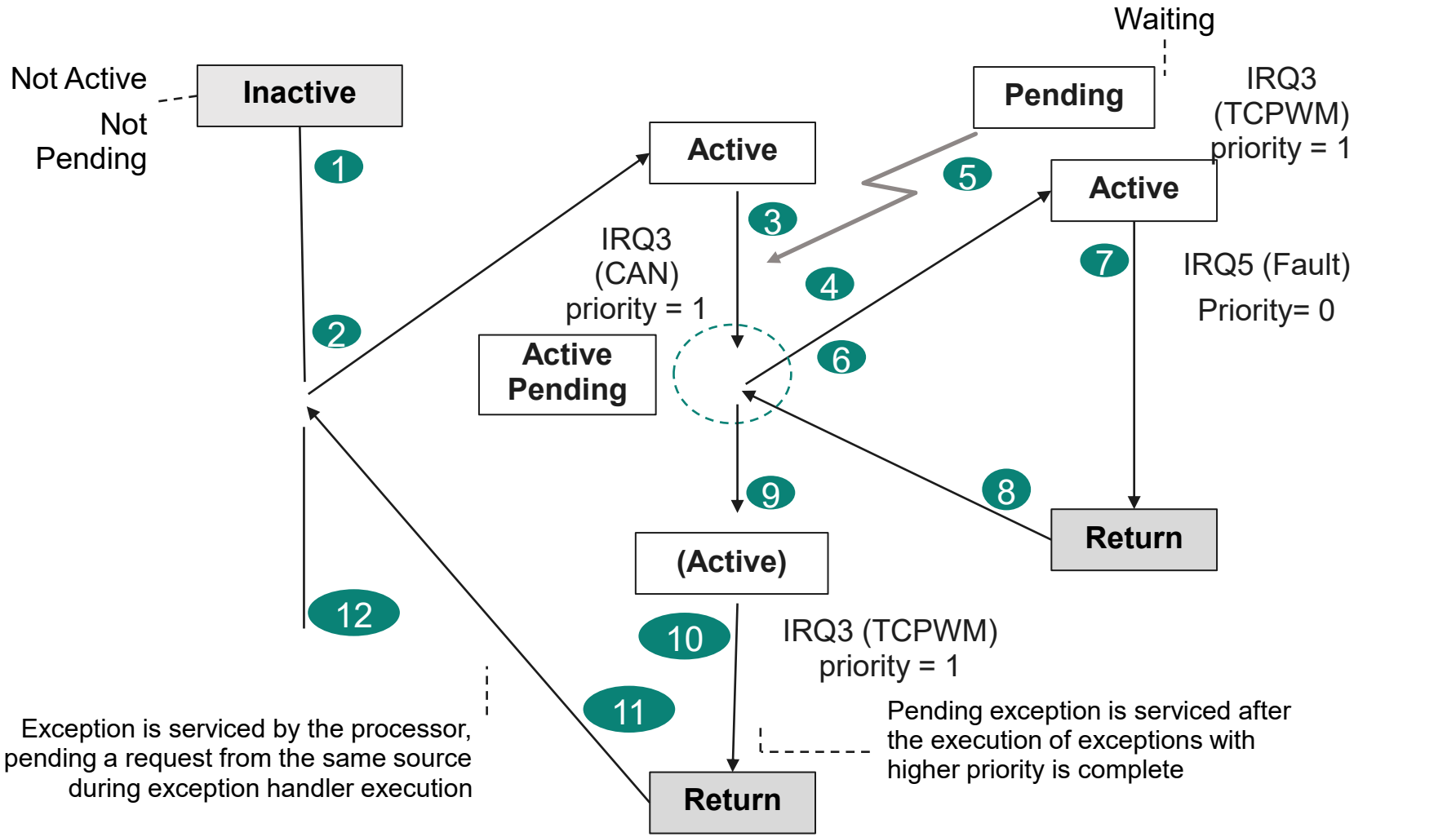
Hint bar

See the architecture reference manual, Interrupt/exception target security state and banking (section 7.6), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Exception states

– Each exception can be in one of the following states:



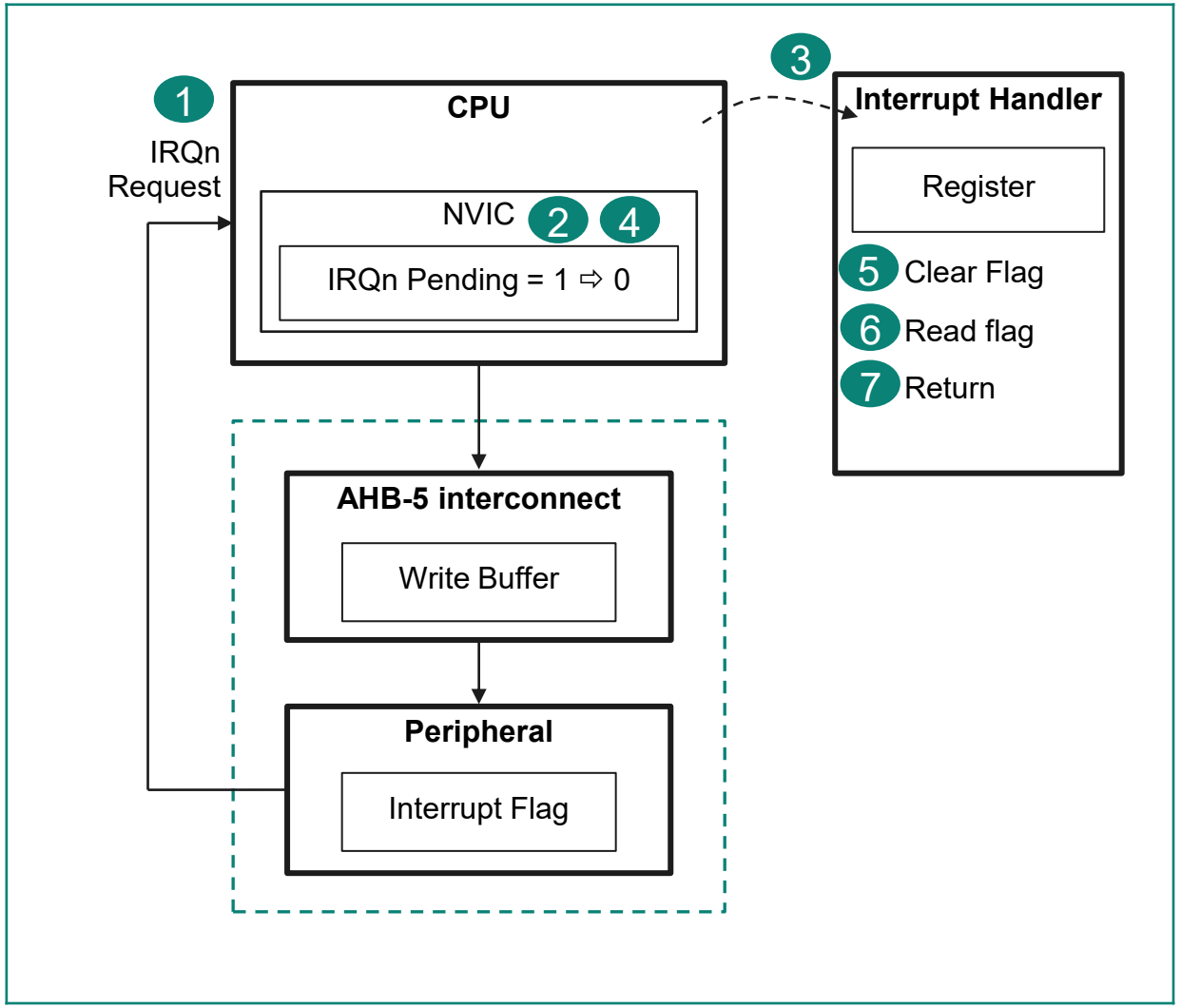
Hint bar

See the architecture reference manual, Interrupt/exception target security state and banking (section 7.6), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Interrupt handler processing (1/2)

- Sequence of a normal interrupt request handling
  1. Interrupt request asserts IRQn.
  2. When the IRQn request can be accepted, the NVIC sets the pending status.
  3. Jumps to the interrupt handler.
  4. Entering the interrupt handler clears the pending status.
  5. Clears the interrupt flag of the Peripheral Register.
  6. Reads the interrupt flag of the Peripheral Register to drain the Write Buffer.
  7. Return



Copyright © Infineon Technologies AG 2026. All rights reserved.

Interrupt handler processing (2/2)

- The CPU interrupt handler uses the SYSTEM_INT_IDX field to index a system interrupt lookup table and jump to the system interrupt handler
- Read after write (RAW) is important in the interrupt handler processing to ensure completion of the write buffer

Hint bar

See the architecture reference manual, Interrupt/exception handling (section 7.3.1), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.
Copyright © Infineon Technologies AG 2022. All rights reserved.

Table of contents

1 Target products 3

2 CPU subsystem 5

3 Register summary 13

4 Interrupts 25

5 **Memories** **40**

6 Clock 55

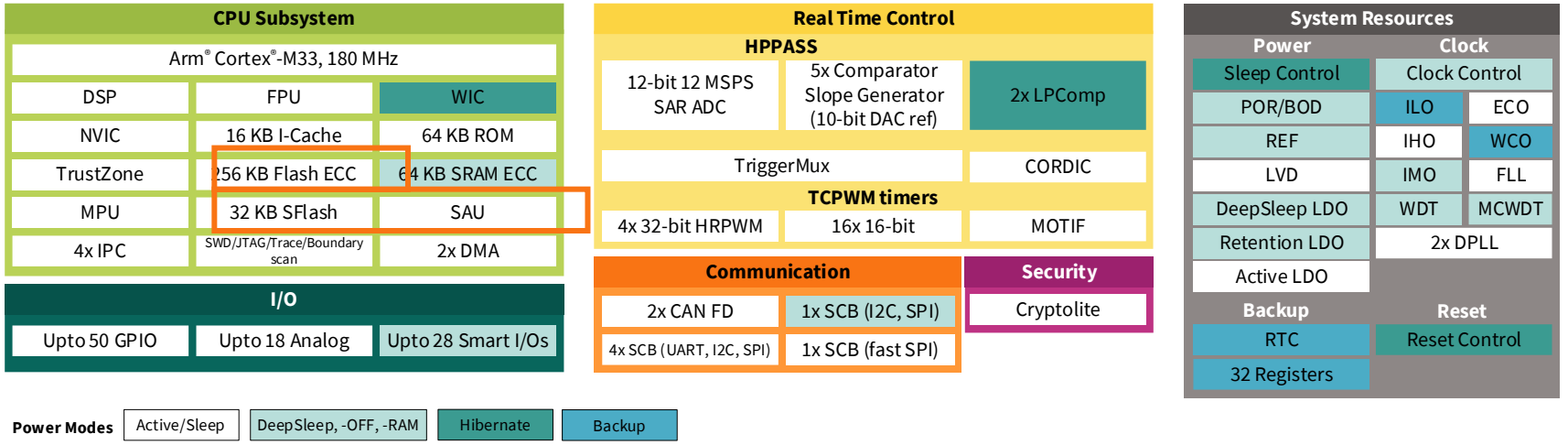
7 Lab: Getting started with an empty application 65

8 Revision history 67

Copyright © Infineon Technologies AG 2026. All rights reserved.

Memories

Memory: Flash/SRAM/EEPROM



Features

- Flash memory size of up to 256 KB, SRAM of 64 KB, and 16 KB I-Cache
- 512-byte row size; minimum programmable unit
- 9-bit ECC for single error correction and dual error detection (SECDED)
- 7-bit SECDED ECC support for 32bit data word of SRAM
- Supports the Read While Write feature for Flash
- Flash/SRAM memory protection using the memory protection controller (MPC)
- Supports a programmable number of wait states
- Flash retention 10 - 20 years based on different temperature profiles
- Flash endurance of minimum 100k program cycles

Benefit

- Support ultra-low power operation
- Flash supports single and dual bank modes for easy firmware upgrade
- Allow reading from one sector while programming the other
- On-chip flash module has ECC support
- Configurable Flash partitioning and protection
- Full 64 KB SRAM available in Deep sleep mode
- Zero wait states for flash while executing instruction via I-Cache

Copyright © Infineon Technologies AG 2026. All rights reserved.

Memories- Interconnections

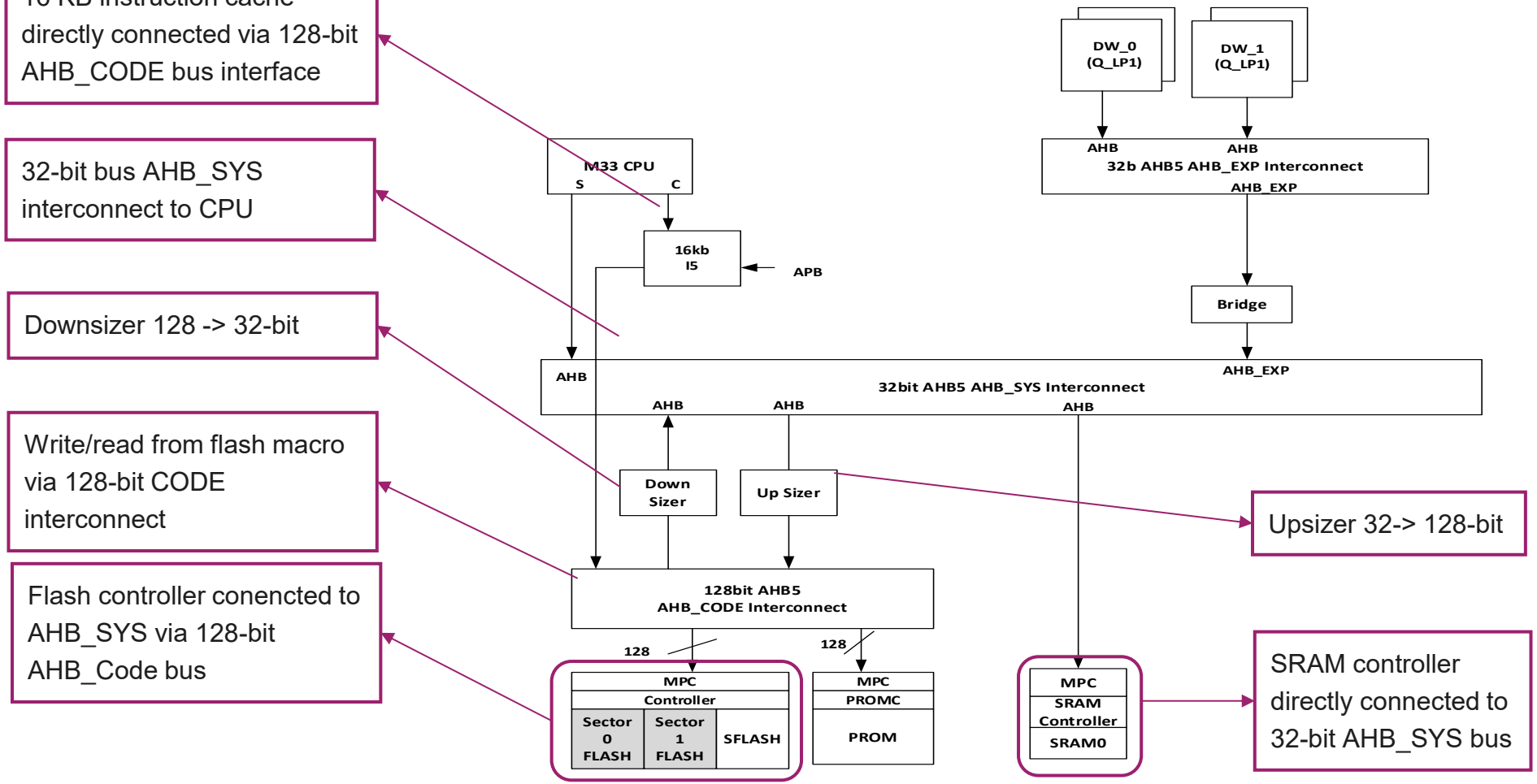
16 KB instruction cache directly connected via 128-bit AHB_CODE bus interface

32-bit bus AHB_SYS interconnect to CPU

Downsizer 128 -> 32-bit

Write/read from flash macro via 128-bit CODE interconnect

Flash controller conencted to AHB_SYS via 128-bit AHB_Code bus



Upsizer 32-> 128-bit

SRAM controller directly connected to 32-bit AHB_SYS bus

Hint bar

See the architecture reference manual, Flash block diagram (section 5.2.1), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Flash controller - Block diagram

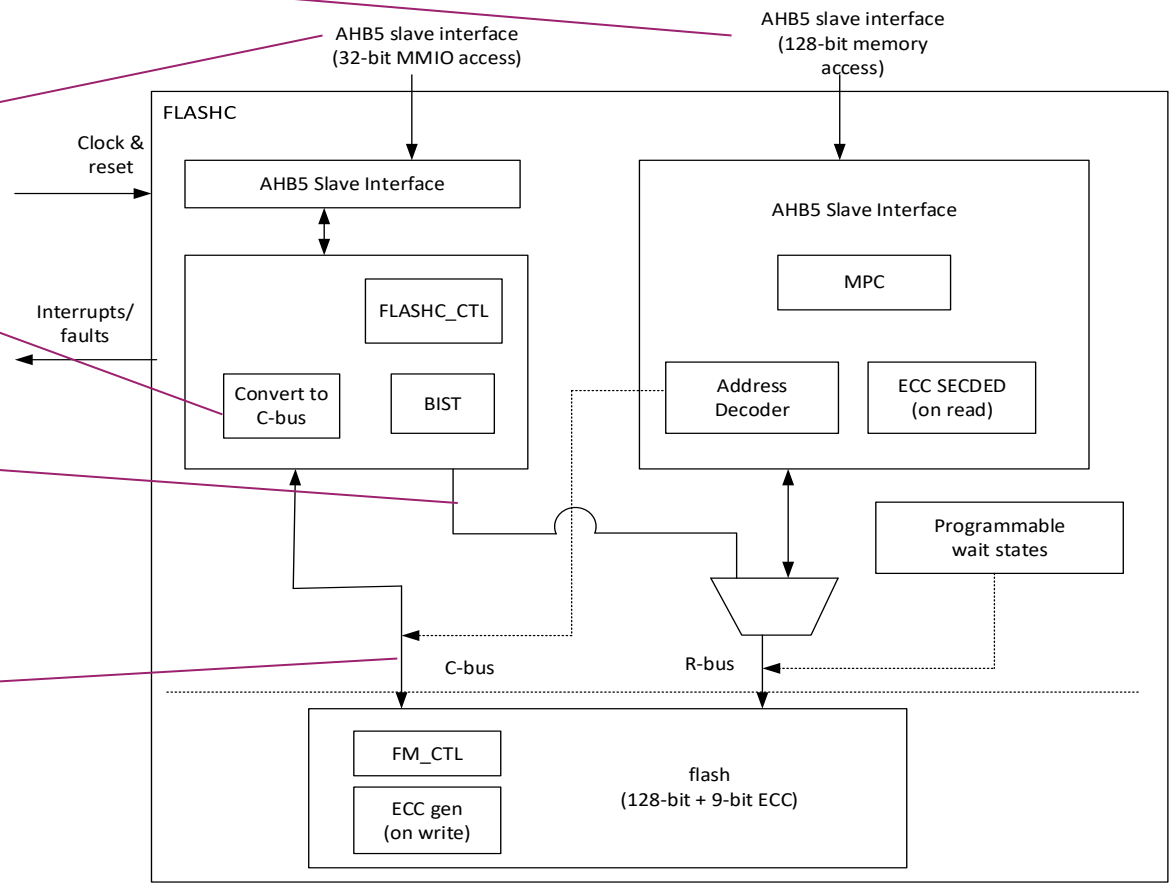
Read and write with 128-bit AHB5 host interface

32-bit MMIO interface for peripheral interconnect with PPC protection

Converts MMIO transaction into C bus

Converts AHB5 transaction into R bus for data read

Flash controller support 3 modes of operation
- full R bus ,
- full C bus
- RWW



Hint bar

See the architecture reference manual, Flash controller (section 5.3), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

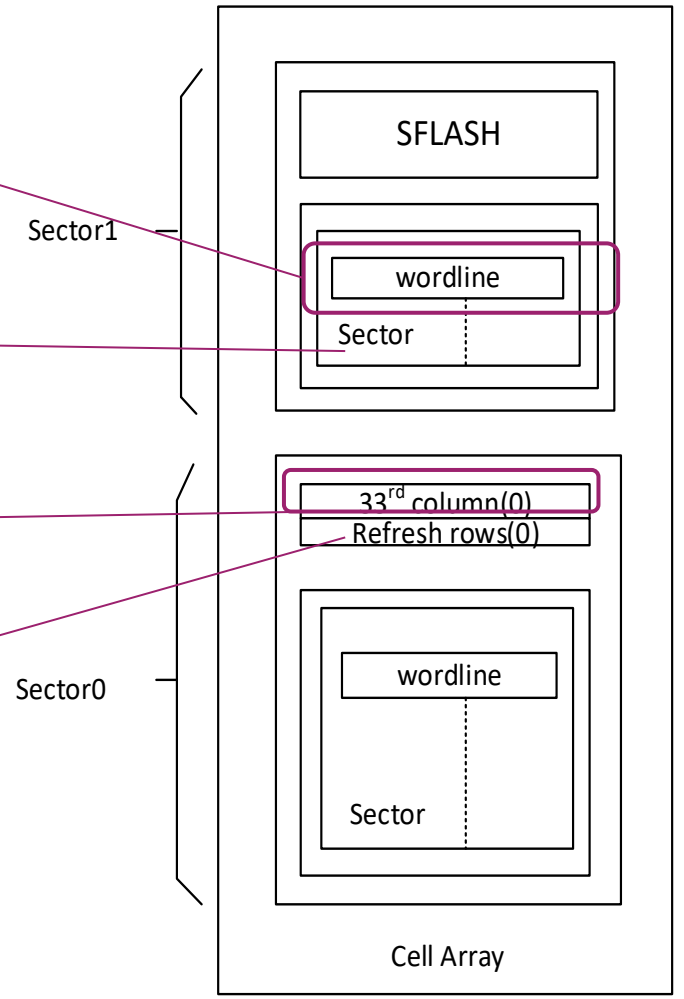
Flash geometry

- Wordline or Rows can be programmed/erased entirely
- Each wordline size is 512 bytes
- It composed of 32 units of Read word
- Read word is 128 data bits +9 bits of ECC

- Sector: It is composed of 256 rows
- Each sector is 128 KB in size

- 33rd Column is used to initiate a refresh operation

- Refresh rows are additional pages in each sector only to support refresh operations
- Refresh operation involves erasing and destroying original data before reprogramming it



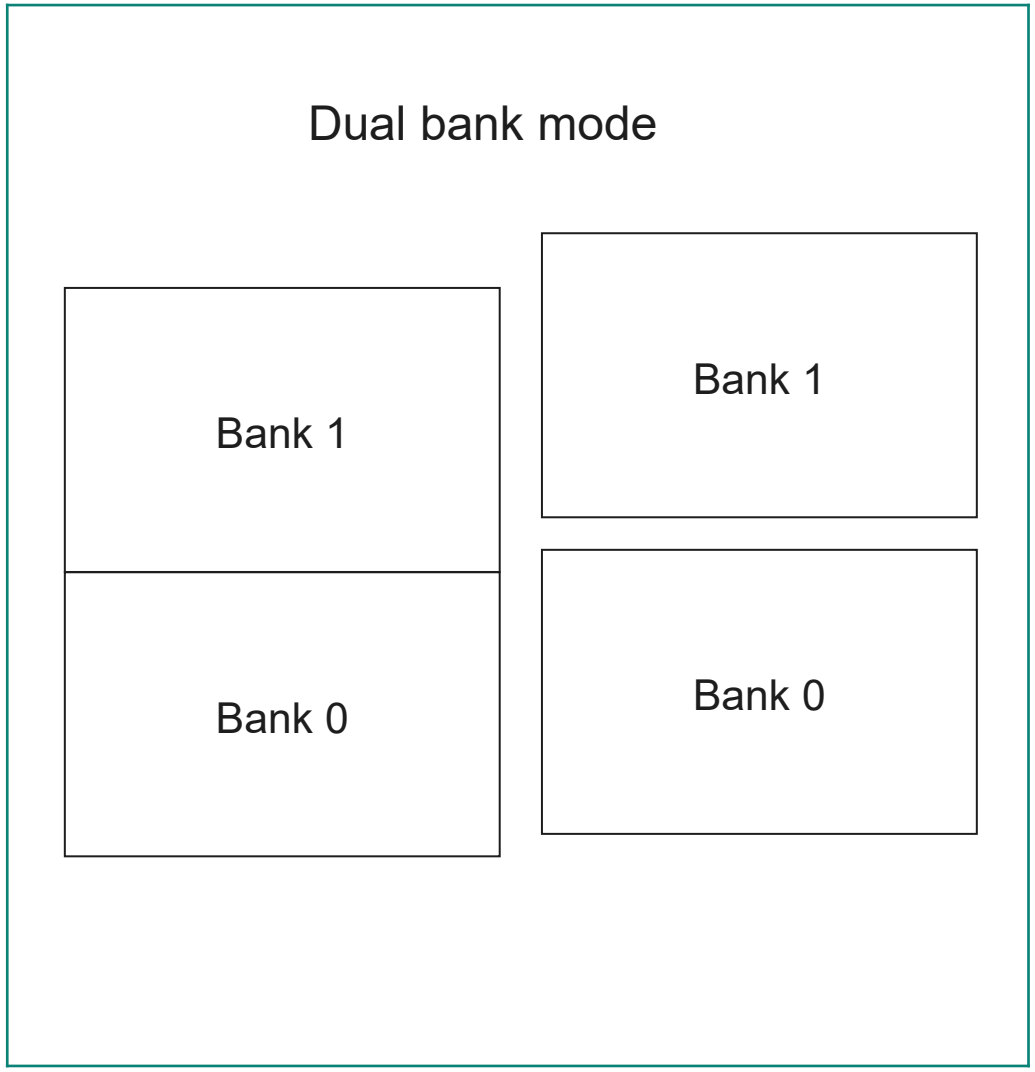
Hint bar

See the architecture reference manual, Flash geometry (section 5.2.2), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Flash banking modes

- Flash memory controller has dual bank feature
  - Main Bank
  - Alternate Bank
- In single bank mode, the entire main flash regions are mapped as a single contiguous address region
- In dual bank mode,
  - Main flash region is split into two halves
  - Each half is presented as a separate address region.
  - These halves can be swapped to support same location firmware upgrades



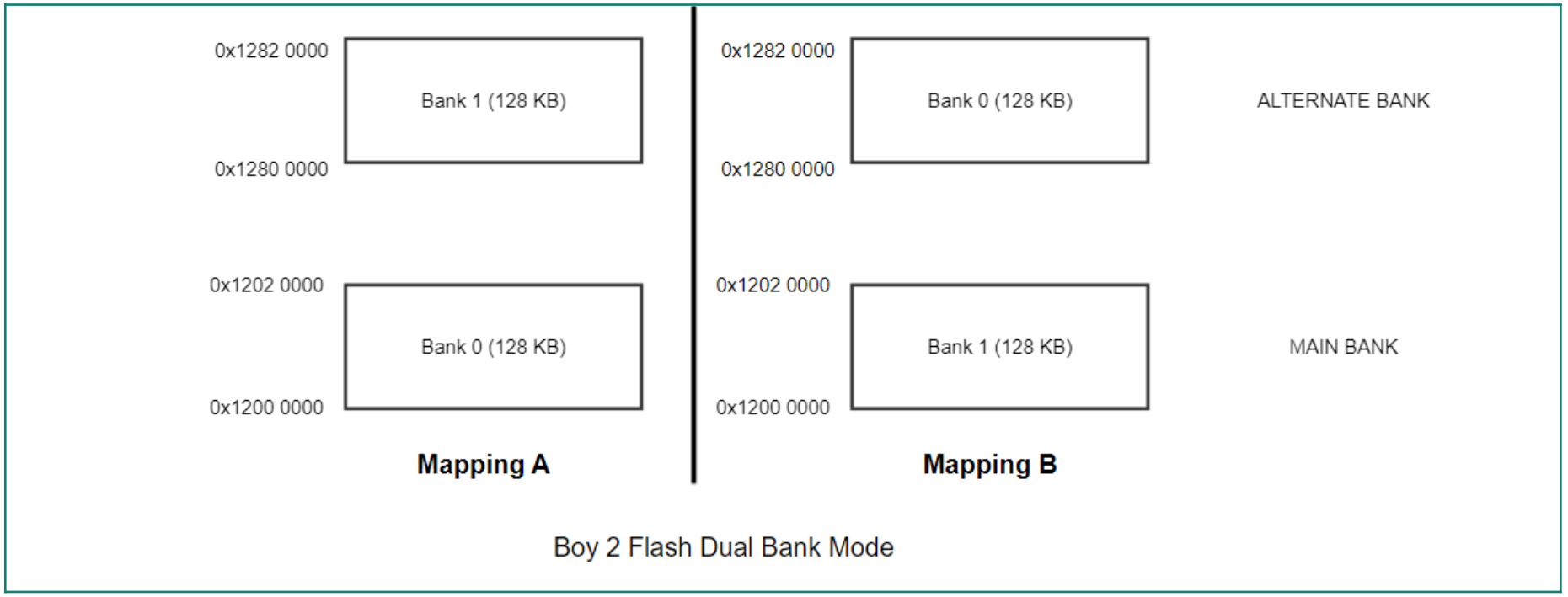
Hint bar

See the architecture reference manual, Bank modes and address mapping (section 5.2.3), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Flash banking modes- Dual bank mapping

- The main bank and alternate bank halves can be swapped to support same-location firmware upgrades
- Mapping A will present the first half in the lower region and the second half in the upper region
- Mapping B will present the first half in the upper region and the second half in the lower region
- Code performing the switch should be in physical sector 0



Boy 2 Flash Dual Bank Mode

Hint bar

See the architecture reference manual, Bank modes and address mapping (section 5.2.3), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Flash dual bank- Remap functionality

- Software can control the remap functionality, which enables writing during flash execution in both bank modes
  - The higher and lower physical banks are swapped in the main flash region

Use case

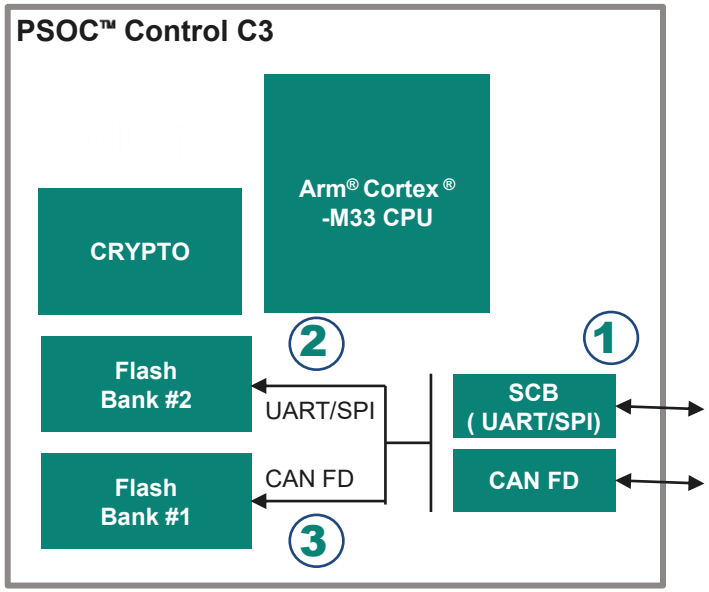
- The CPU executes from a current software image in the lower bank while the higher bank is programmed with a new software image
- When the CPU reboots, the ROM boot code changes the REMAP field such that the CPU executed from the new image is on the higher bank

Hint bar

See the architecture reference manual, Bank modes and address mapping (section 5.2.3), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Flash dual bank- Remap functionality (contd.)



1. Software update data is received via wired communication interface.
2. Updated data can be stored in the internal Flash Bank #1. The control system sends a command to switch to the new software version. The new software version is activated.
3. The previous version is still available in the internal Flash Bank #0 and can be reused as a roll-back solution.

Hint bar

See the architecture reference manual, Bank modes and address mapping (section 5.2.3), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Flash read while write (RWW) support

- RWW is supported by PSOC™ control C3, which means that the application can read or execute from one sector while there is an ongoing write or erase operation in another sector
- During RWW, the following operations can be performed on the write sector:
  - Row erase or program
  - Sector erase or pre-program during the RWW
- The following are not available:
  - Data read in the Write Sector
  - Bulk erase or pre-program
  - Test modes

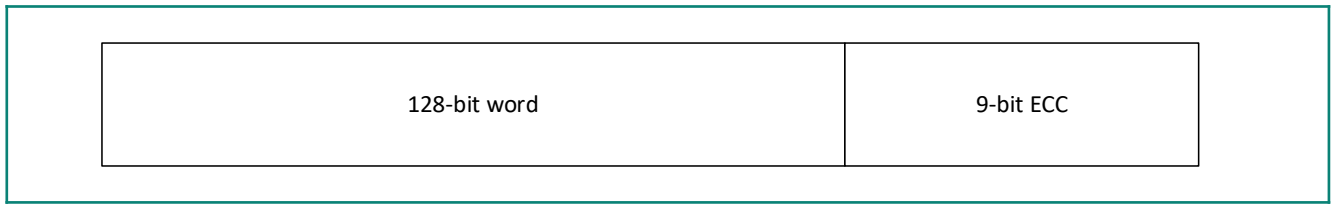
Hint bar

See the architecture reference manual, Read While Write support (section 5.4), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Flash error correcting code (ECC)

Address



- The FM uses an additional 9 bits for every word of 128 bits to perform single error correction (SEC) and dual error detection (DED)
- Due to ECC, the internal read width is extended from 128 bits to 137 bits
- Fault report structure
  - Both correctable and non-correctable ECC errors are reported to the fault structure
- Error Injection
  - It is possible to generate ECC error by providing an error injection address and error injection data register
  - Use Case
    - As initial diagnosis of ECC before running the application
    - To test the error recovery routines

Hint bar

See the architecture reference manual, Bank modes and address mapping (section 5.2.3), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

SRAM controller

- The SRAM controller has an AHB5 interface that connects to the AHB5 infrastructure
- Supports 8, 16, and 32-bit access
- Provides exclusive access support using exclusive access monitor (EAM) from Arm® SIE-200
- SRAM memory protection using the memory protection controller (MPC)
- SRAM power mode control using Arm® power policy unit (PPU)
- The SRAM controller supports the error correction code (ECC) for the SRAM
- The SRAM controller differentiates between the following three types of AHB5 transfers
  - AHB5 read transfers
    - AHB5 reads transfers, and a read transfer is translated into SRAM read access
  - 32-bit AHB5 write transfers
    - A 32-bit AHB5 write transfer is translated into SRAM write access
  - 8-bit and 16-bit AHB5 write transfers (also referred to as partial AHB5 write transfers)
    - Partial writes (8-bits and 16-bits) are supported through read, modify, and write by applying the associated mask during write

Hint bar

See the architecture reference manual, Architecture (section 6.2), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

SRAM ECC

- The 7-bit SECDED parity covers the 32-bit data word as a single entity; there is no dedicated parity for the four 8-bit bytes within the data
- Therefore, any partial 8-bit or 16-bit AHB5 write bus transfer requires a SRAM read access and a SRAM write access
- When a non-recoverable error, that is an error of more than 1-bit is detected, the data word that is used as the result for an AHB5 bus transfer is incorrect, but NO AHB5 bus error is generated
- Fault report structure
  - Both correctable and non-correctable ECC errors are reported to the CPUSS fault structure
- Error injection
  - The Error injection mechanism is supported by 32bit data word write into an address defined in ECC_MATCH

Hint bar

See the reference manual, Error correcting code (ECC) (section 6.3), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

EEPROM memory

Function

– A type of non-volatile memory used to store small amounts of data that must be saved when power is removed

Note: Any EEPROM emulation memory should be allocated in sector 0.

Common use

– Configuration settings: Storing user preferences and system configuration
– Calibration data: Keeping calibration constant for sensor and analog components
– Firmware storage: Holding bootloaders or small firmware updates
– Data logging: Recording events, usage data, or logs that need to persist between power cycles

Hint bar

See the architecture reference manual, Flash geometry (section 5.2.2), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Table of contents

1 Target products 3

2 CPU subsystem 5

3 Register summary 13

4 Interrupts 25

5 Memories 40

6 **Clock** **55**

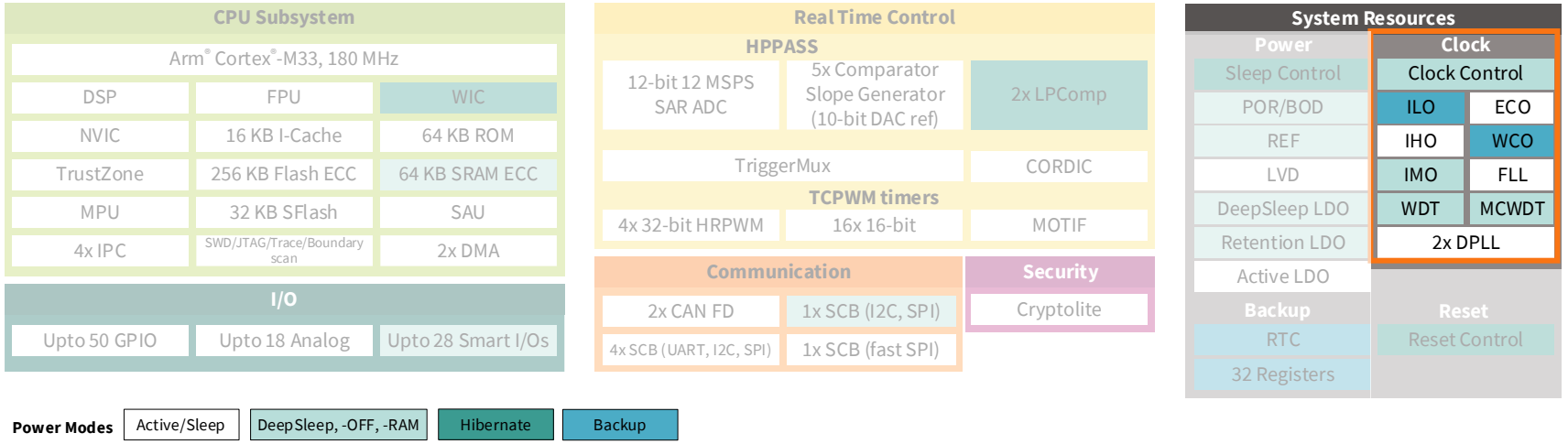
7 Lab: Getting started with an empty application 65

8 Revision history 67

Copyright © Infineon Technologies AG 2026. All rights reserved.

Clock

System Resources: Clocks



Features

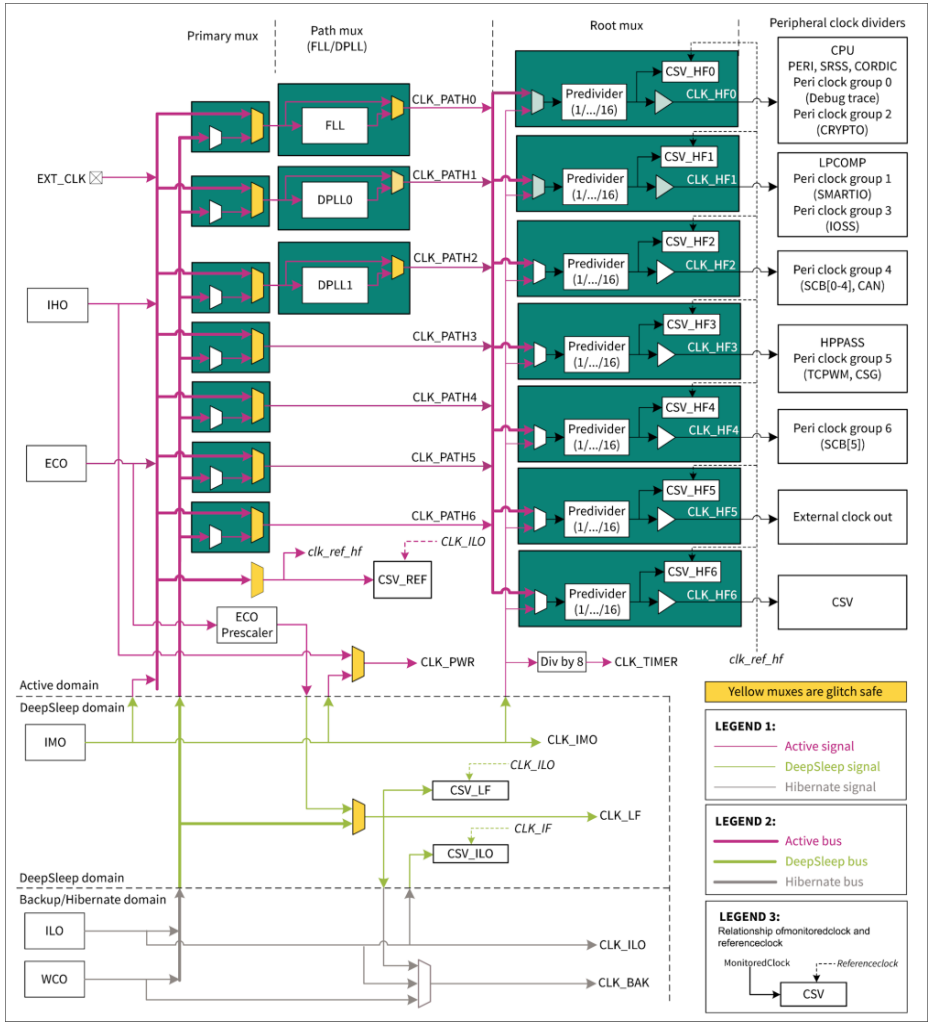
- Three internal clock sources: ILO, IHO, IMO
- Three external clock sources: EXT_CLK, ECO, WCO
- One frequency lock loop (FLL) with 24 to100 MHz output range
- Two digital phase-locked loops (DPLL0 and DPLL1) with 25 MHz to 250 MHz output range

Benefit

- 8-MHz ±2% internal main oscillator (IMO)
- 48-MHz ±1% internal high-speed oscillator (IHO)
- 32-kHz ±10% internal low-speed oscillator (ILO)
- Up to 80 MHz external clock (EXT_CLK) from an I/O
- 4 to 35 MHz external crystal oscillator (ECO)
- 32.768 kHz external watch crystal oscillator (WCO)

Copyright © Infineon Technologies AG 2026. All rights reserved.

Clock tree



Hint bar

See the architecture reference manual, Clocking system (chapter 19), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Clock tree (contd.)

- Seven clock paths
  - CLK_PATH0 … CLK_PATH6
- Seven high-frequency root clocks (CLK_HF)
  - CLK_HF0 … CLK_HF6
- Low-frequency clock (CLK_LF) options
  - ILO, WCO, and ECO prescaler
- Peripheral clock dividers
  - Thirteen 8-bit integer clock dividers
  - Five 16-bit integer clock dividers
  - Four 16.5-bit fractional clock dividers
  - Two 24.5-bit fractional clock dividers
- Each clock source has a clock supervisor (CSV)
- Timer clock: fixed 1 MHz derived from IMO
- Low-frequency clock, backup clock, and power clock

Hint bar

See the architecture reference manual, Clocking system (chapter 19), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Clock source summary

Clock	Frequency (range)	Power modes
IMO	8 MHz ±2%	Active/Sleep/Deep Sleep
IHO	48 MHz ±1%	Active/Sleep
ILO	32 kHz ±10%	Active/Sleep/Deep Sleep/ Hibernate and Backup
EXT_CLK	0 – 80 MHz	Active/Sleep
ECO	4 - 35 MHz	Active/Sleep
WCO	32.768 kHz	Active/Sleep/Deep Sleep/ Hibernate and Backup

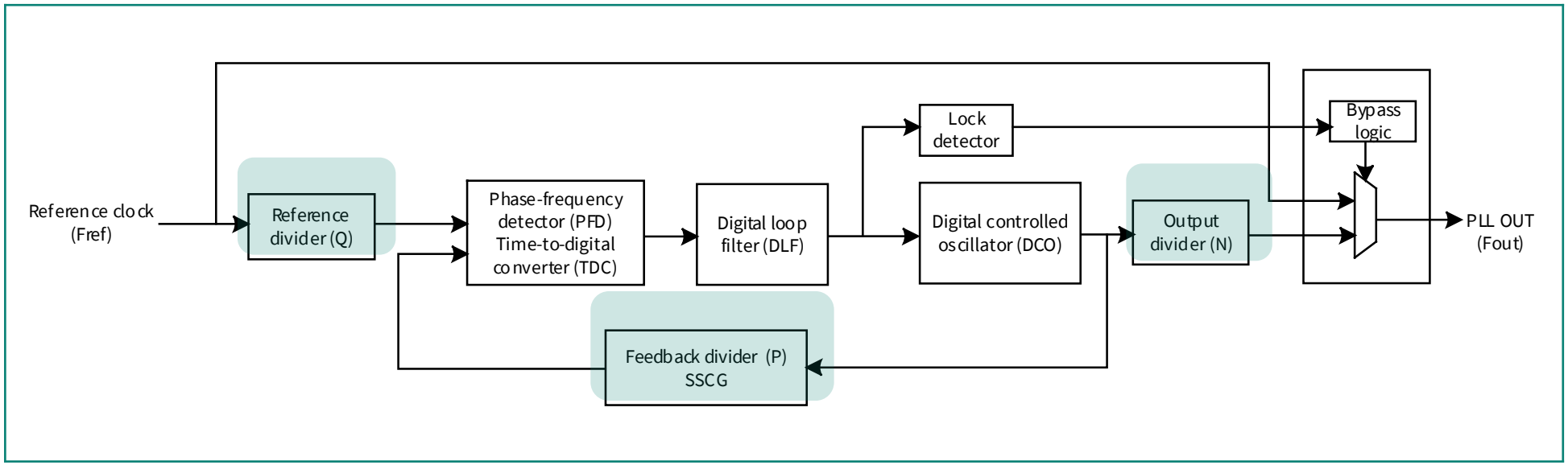
Hint bar

See the architecture reference manual, Clock sources (section 19.3), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Digital phase- locked loop (DPLL)

- PSOC™ Control C3 MCU has two DPLLs with 25 - 250 MHz output range
- Input range of 4 to 64 MHz
- Supports the following operation modes
  - Integer operation mode: $F_{out} = F_{ref} \times (P/Q/N)$
  - Fractional operation mode: $F_{out} = (F_{ref}/Q) \times (P + FRAC_DIV)/N$, the frequency resolution is $^{1}/_{2^{24}}$
    For example, Non-standard frequency 40.456 MHz generated from F_{ref} = 8 MHz, P = 40, Q = 1, N = 8, FRAC_DIV = 7650410/16777216. $F_{out} = (8 \times (40 + 7650410/16777216)) / 8 = 40.456MHz$
- Available in the Active and Sleep modes



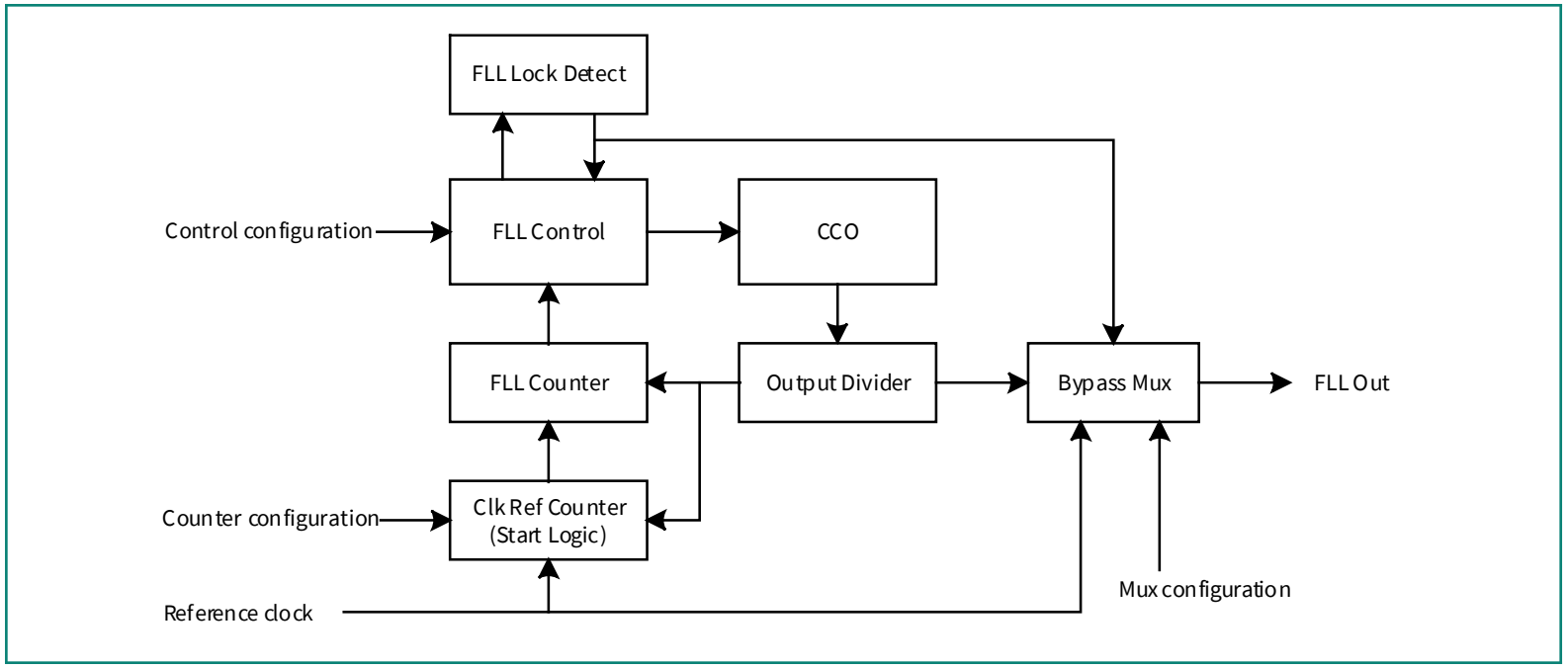
Hint bar

See the architecture reference manual, Digital phase-locked loop (DPLL) (section 19.4.1), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Frequency lock loop (FLL)

- PSOC™ Control C3 MCU has one FLL. FLL takes an input and multiplies the frequency
- Consumes less current, better duty cycle than the DPLL



- Output Mux selects FLL input, FLL output, or automatically switches between input and output based on FLL lock
- Available in the Active and Sleep modes

Hint bar

See the architecture reference manual, Frequency lock loop (FLL) (section 19.4.2), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Clock supervision (CSV)

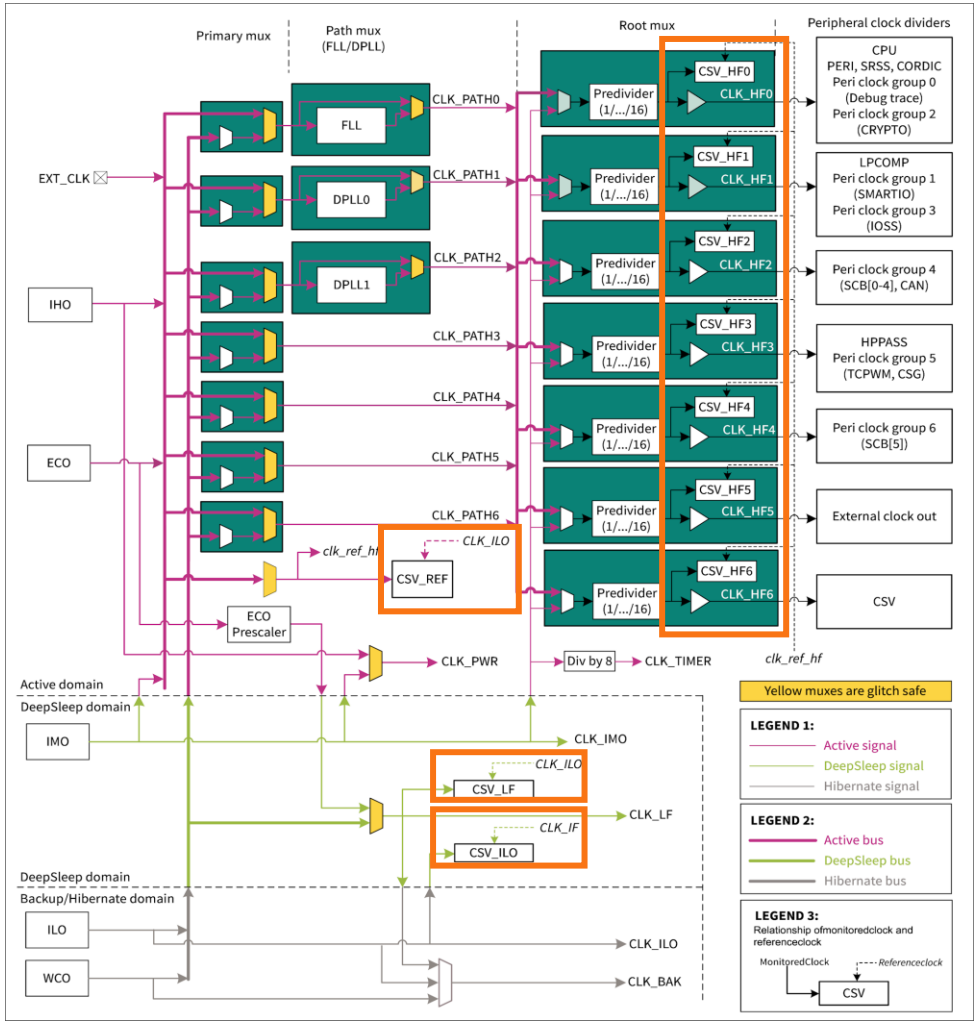
- PSOC™ Control C3 MCU has CSV_HFs, CSV_REF, CSV_LF, and CSV_ILO
  - All CLK_HFs have the CSV. All CLK_HF CSVs use the same reference clock (CLK_REF_HF)
  - ILO is supervised both by CLK_LF and CLK_REF_HF; CLK_LF is needed for Deep Sleep supervision and CLK_REF_HF is needed for accuracy while Active
  - CSV_REF monitor CLK_REF_HF with CLK_ILO
  - All CSV_HFs and CSV_REF are in the Active domain
  - The Deep Sleep domain has two CSVs (CSV_LF and CSV_ILO)

Hint bar

See the architecture reference manual, Clock supervision (CSV) (section 19.9), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Clock supervision (CSV) (contd.)



Hint bar

See the aritecture reference manual, Clock supervision (CSV) (section 19.9), for additional details.

Copyright © Infineon Technologies AG 2026. All rights reserved.

Table of contents

1 Target products 3

2 CPU subsystem 5

3 Register summary 13

4 Interrupts 25

5 Memories 40

6 Clock 55

7 **Lab: Getting started with an empty application** **65**

8 Revision history 67

Copyright © Infineon Technologies AG 2026. All rights reserved.

Lab: Getting started with an empty application

Table of contents

1 Target products 3

2 CPU subsystem 5

3 Register summary 13

4 Interrupts 25

5 Memories 40

6 Clock 55

7 Lab: Getting started with an empty application 65

8 **Revision history** **67**

Copyright © Infineon Technologies AG 2026. All rights reserved.

Revision history

Document revision	Date	Description of change
**	2026-02-06	Initial release

Copyright © Infineon Technologies AG 2026. All rights reserved.


Infineon