

E3 – Understanding PSOC™ Edge memory usage

Customer training workshop (CTW)

March 2026

Public

Prerequisites

- Recommended training prerequisites
  - Introduction to PSOC™ Edge and ModusToolbox™
    - For an introduction to PSOC™ MCUs, including a getting started guide to ModusToolbox™, visit:
      - https://www.infineon.com/product-information/psocdeveloper
    - For PSOC™ Edge trainings, from beginner tutorials to advanced trainings, visit:
      - PSOC™ Edge E84 Training Collection

- Development tools for all labs (see training manual for links)
  - ModusToolbox™ v3.7
  - Eclipse IDE for ModusToolbox™ v2025.8.0
  - Edge Protect Security Suite v1.6
  - ModusToolbox™ programming tools v1.7.0
  - KIT_PSE84_EVAL_EPC2 BSP v1.0.0
  - Terminal emulator
  - KIT_PSE84_EVAL



Copyright © Infineon Technologies AG 2026. All rights reserved.

Table of contents

1 PSOC™ Edge memory architecture 4

2 PSOC™ Edge memory technologies 6

3 Lab 1: Assigning code and data to custom linker sections 19

4 Lab 2: Assigning a source file to a linker section 21

5 PSOC™ Edge bus architecture and memory access 23

6 Lab 3: Placing and executing code from RRAM 28

7 Lab 4: Placing and executing code from SRAM 30

8 PSOC™ Edge Memory tradeoffs 32

9 Lab 5: Low-power memory considerations 36

Copyright © Infineon Technologies AG 2026. All rights reserved.

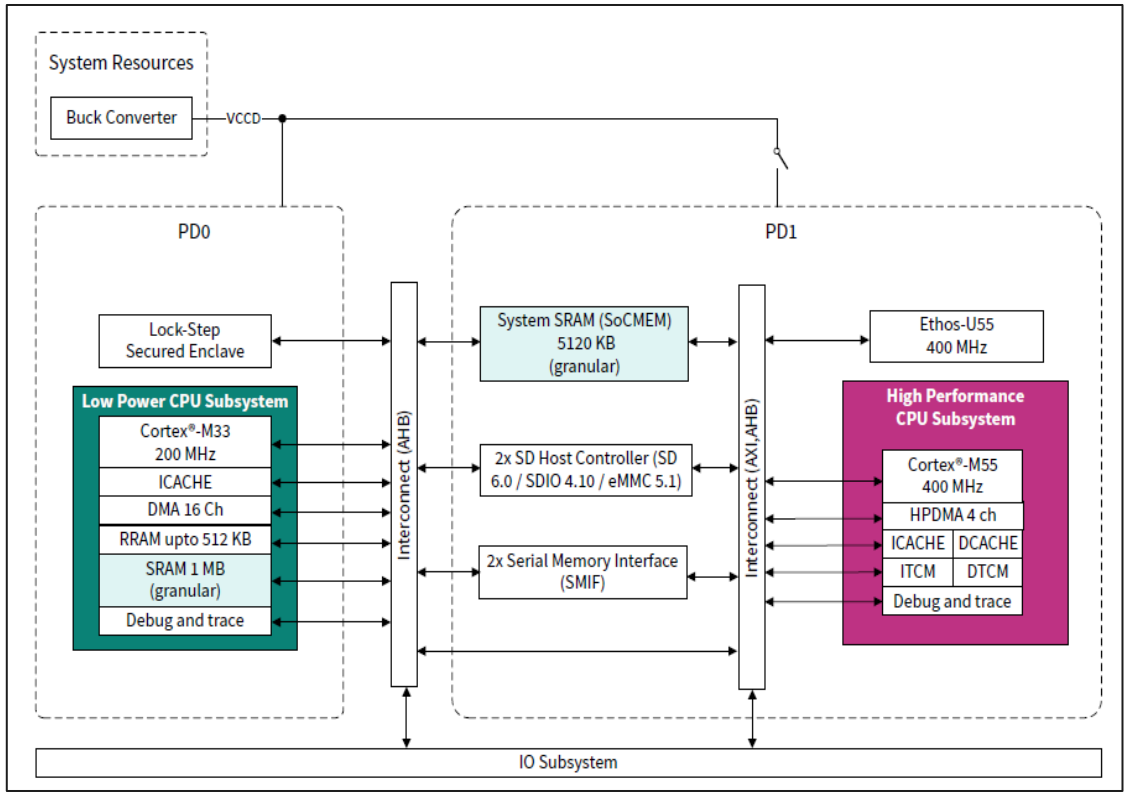
Table of contents

1	**PSOC™ Edge memory architecture**	**4**
2	PSOC™ Edge memory technologies	6
3	Lab 1: Assigning code and data to custom linker sections	19
4	Lab 2: Assigning a source file to a linker section	21
5	PSOC™ Edge bus architecture and memory access	23
6	Lab 3: Placing and executing code from RRAM	28
7	Lab 4: Placing and executing code from SRAM	30
8	PSOC™ Edge Memory tradeoffs	32
9	Lab 5: Low-power memory considerations	36

Copyright © Infineon Technologies AG 2026. All rights reserved.

PSOC™ Edge memory architecture

- The PSOC™ Edge features two power domains
  - PD0: Always-on low-power domain
  - PD1: High-performance domain
- Power source and operating voltage
  - Both PD0 and PD1 are powered by the internal buck output and operate at the same voltage level
- Independent control and flexibility
  - PD1 can be independently switched off without affecting PD0
  - PD1 can be reactivated as necessary for performance needs
- Power conservation
  - When high-performance peripherals are not needed, PD1 is turned off to save power



Copyright © Infineon Technologies AG 2026. All rights reserved.

Table of contents

1 PSOC™ Edge memory architecture 4

2 **PSOC™ Edge memory technologies** **6**

3 Lab 1: Assigning code and data to custom linker sections 19

4 Lab 2: Assigning a source file to a linker section 21

5 PSOC™ Edge bus architecture and memory access 23

6 Lab 3: Placing and executing code from RRAM 28

7 Lab 4: Placing and executing code from SRAM 30

8 PSOC™ Edge Memory tradeoffs 32

9 Lab 5: Low-power memory considerations 36

Copyright © Infineon Technologies AG 2026. All rights reserved.

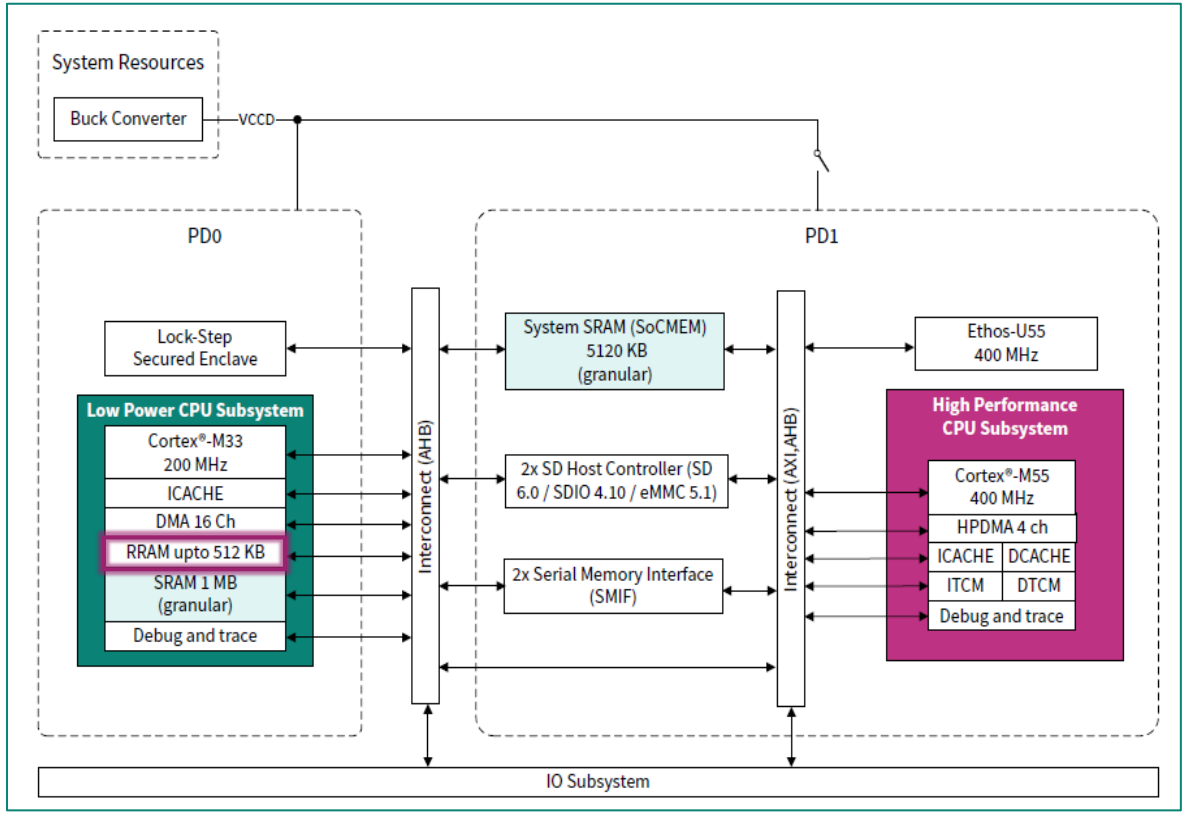
RRAM (NVM)

RRAM

- Up to 512 KB non memory storing data with changeable resistances
- Functions like slow RAM with data retention capabilities
- Ideal for storing CPU instructions/data when power is off
- Can emulate electrically erasable programmable read-only memory and one-time programmable (OTP) regions

OTP

- 5.5 KB OTP memory available for storage of security assets

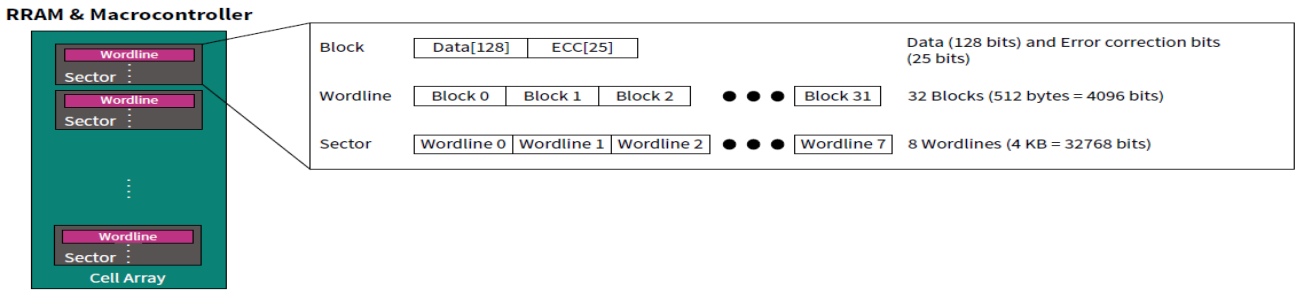


Copyright © Infineon Technologies AG 2026. All rights reserved.

RRAM (NVM) (contd.)

Granularity and organization

– Minimum read/write unit is 128-bits + 25-bits ECC



Endurance and retention

– Minimum 100,000 program/erase cycles with a 20-year retention under -40°C to +85°C ambient conditions

– Minimum 10,000 program/erase cycles with a 10-year retention under -40°C to +105°C ambient conditions

Wait states

– RRAM reads controlled by an internal 160 MHz clock, producing data within 30 ns analog read time

Voltage (V)	Frequency (MHz)	Wait states (WS)
0.9 (HP mode)	200	6
0.8 (LP mode)	80	3
0.7 (ULP mode)	50	2

Copyright © Infineon Technologies AG 2026. All rights reserved.

RRAM (NVM) (contd.)

MAIN NVM

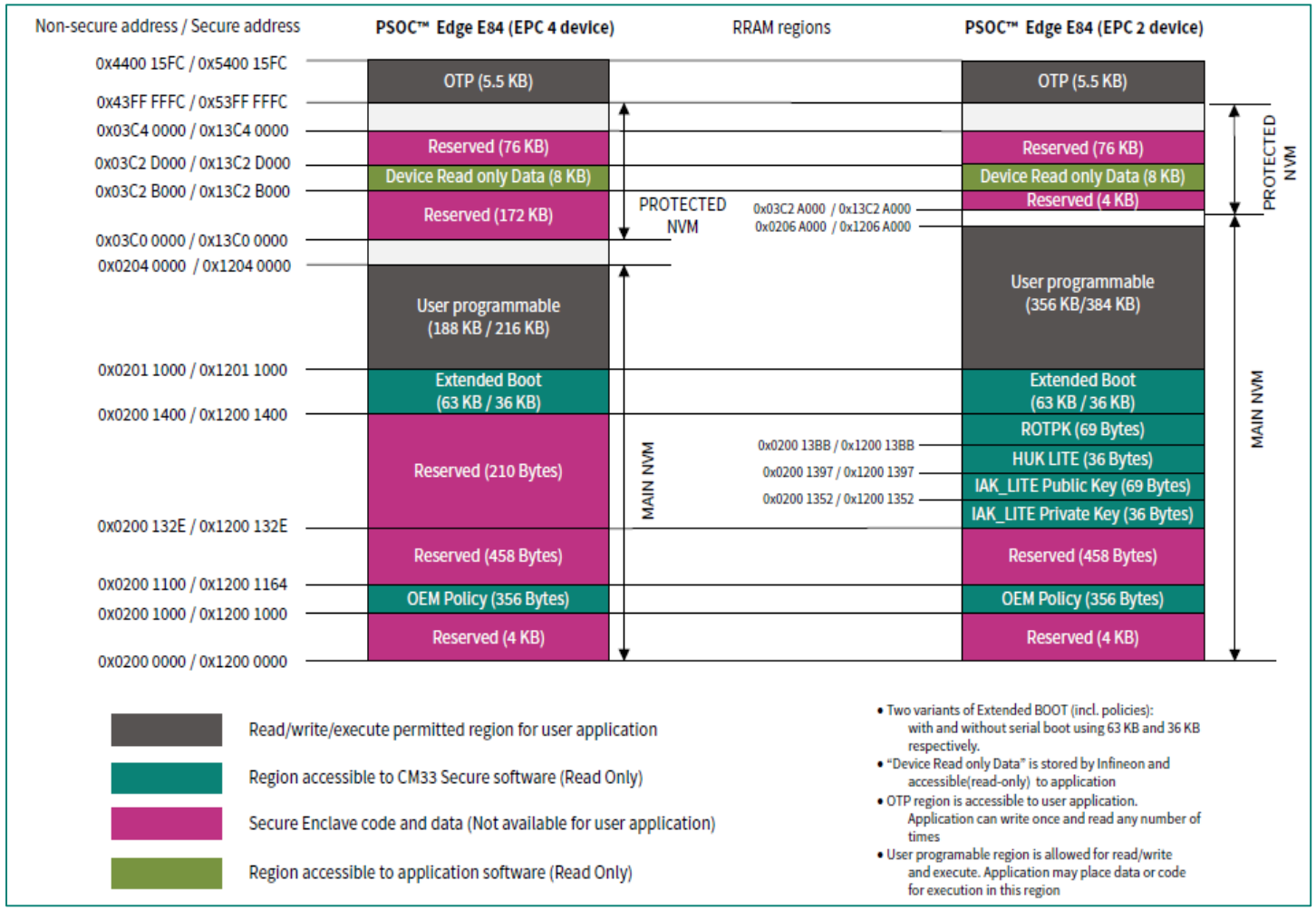
The main non- memory region

PROTECTED NVM

Region accessible only by the secure enclave

OTP

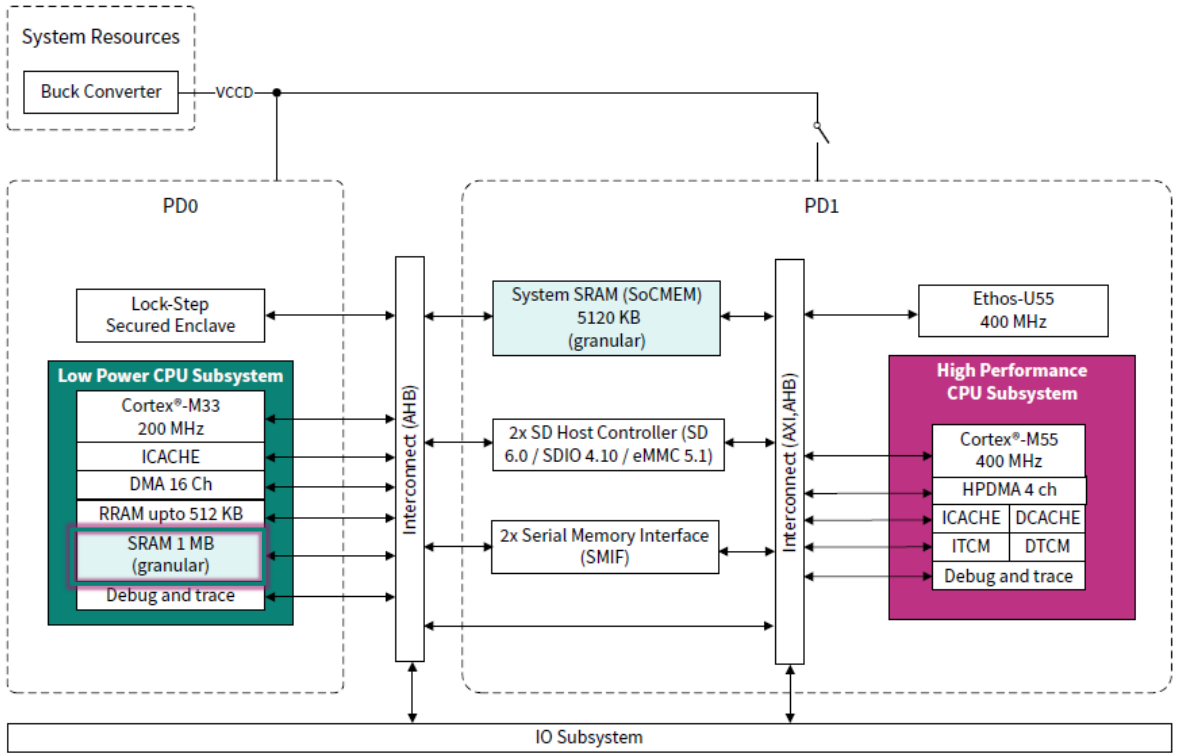
One-time programmable memory region



Copyright © Infineon Technologies AG 2026. All rights reserved.

SRAM

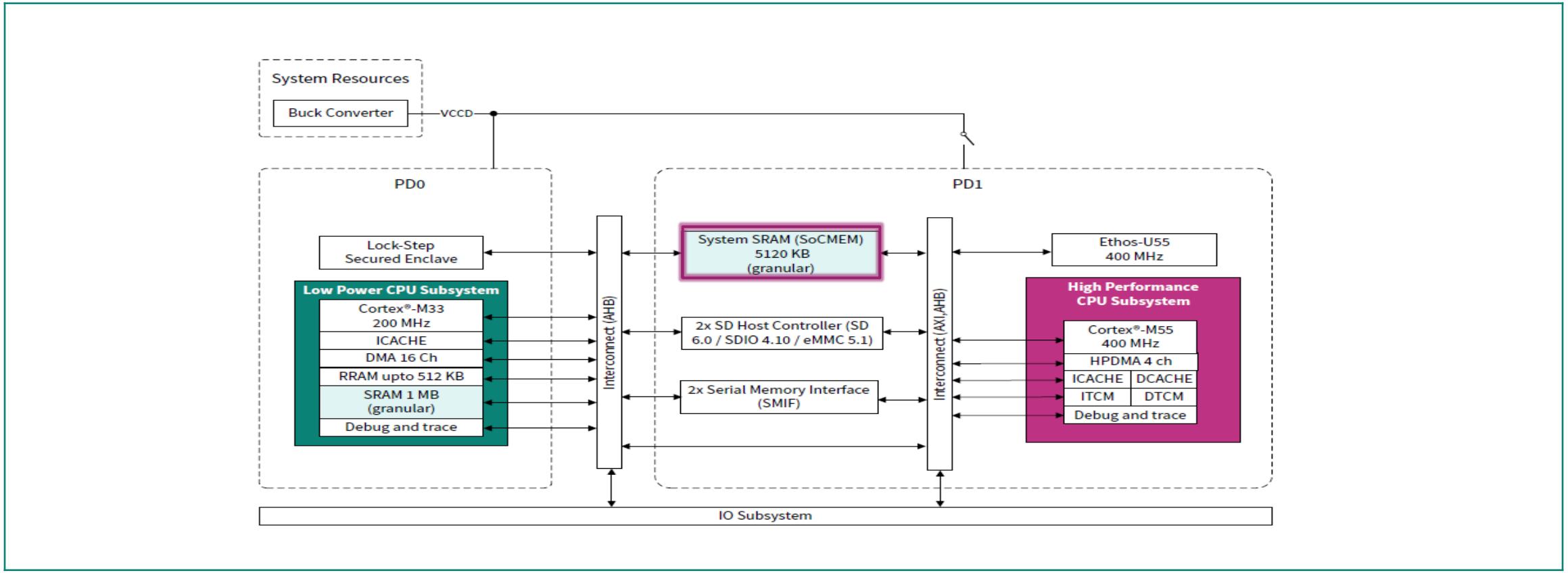
- 1 MB available in the low-power CPU subsystem
- Arrays are partitioned into sectors, resulting in a retention granularity of 64 KB to minimize current draw in the System Deep Sleep power mode
- SRAM doesn't have wait states



Copyright © Infineon Technologies AG 2026. All rights reserved.

System SRAM (SoCMEM)

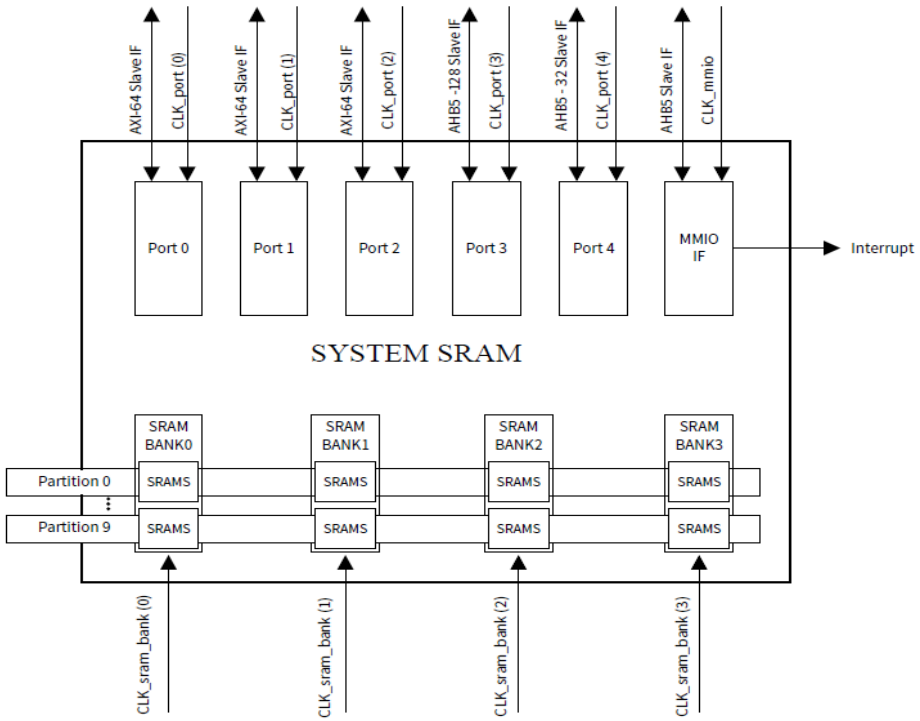
- Multi-ported on-chip memory controller with multiple banks to increase bandwidth
- Features multiple ports for direct bus connections, ensuring efficient bus utilization
- Directly accessible by both interconnect buses



Copyright © Infineon Technologies AG 2026. All rights reserved.

System SRAM (SoCMEM)

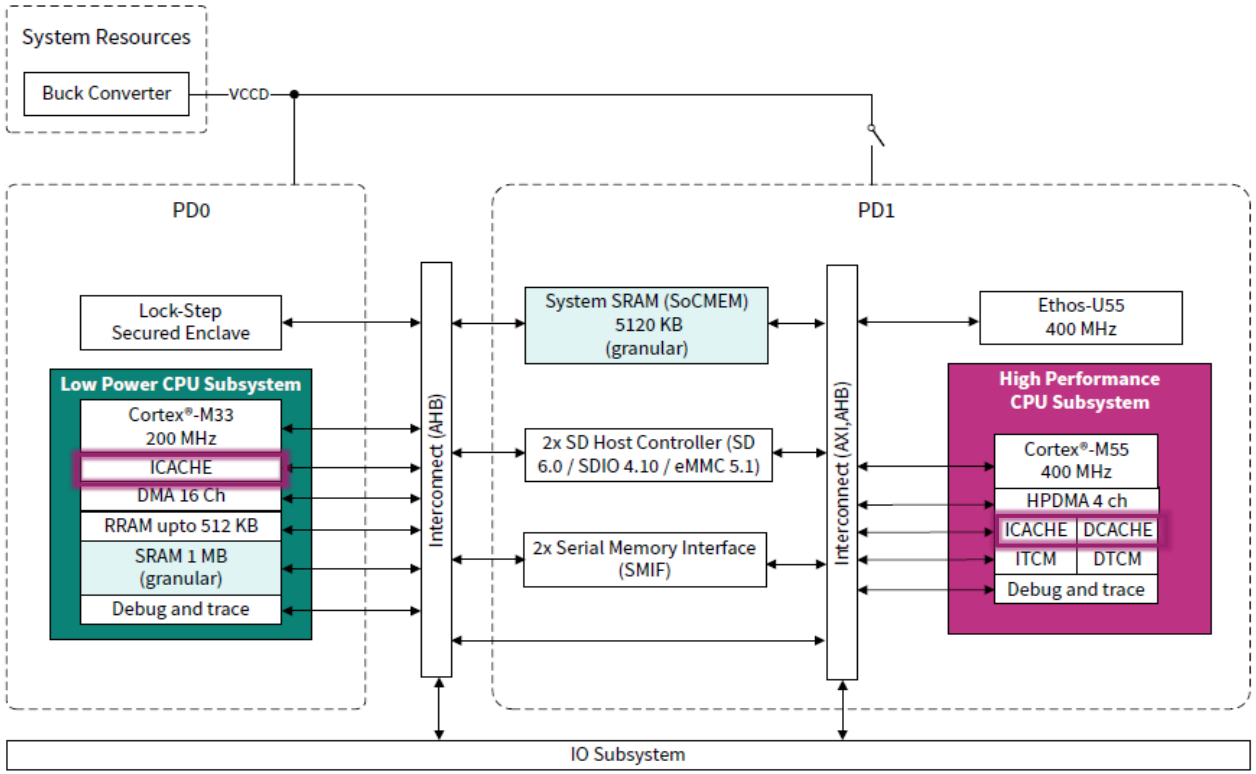
- Organized into 10 partitions of 512 KB, with four banks each, totaling 5120 KB
- 3 types of ports:
  - 3x64-bit AXI ports
  - 1x128-bit AHB5 port
  - 1x32-bit AHB5 port



Copyright © Infineon Technologies AG 2026. All rights reserved.

Cache memories

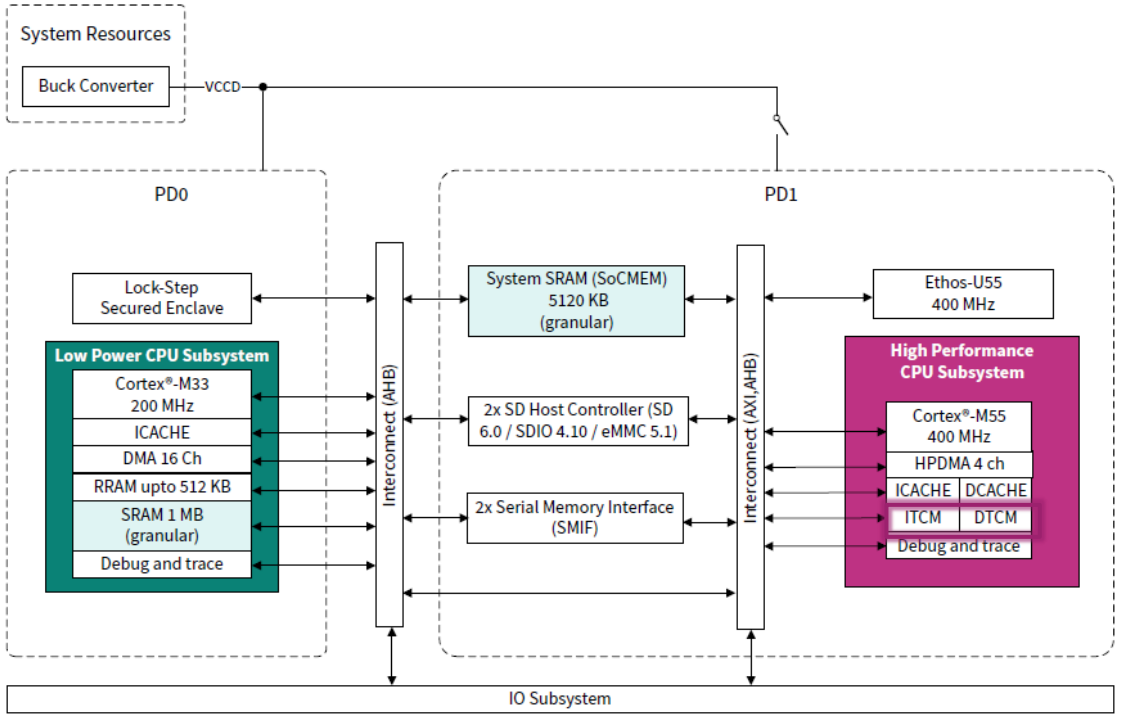
- High-speed, memory for quick access to frequently used instructions and data
- Divided into instruction cache (ICACHE) and data cache (DCACHE)
- The Cortex®-M33 and Cortex®-M55 CPUs have 16 KB and 32 KB instruction caches, respectively
- Data cache is only available for the CM55; its size is 32 KB



Copyright © Infineon Technologies AG 2026. All rights reserved.

Tightly coupled memories for Cortex®-M55 (CM55)

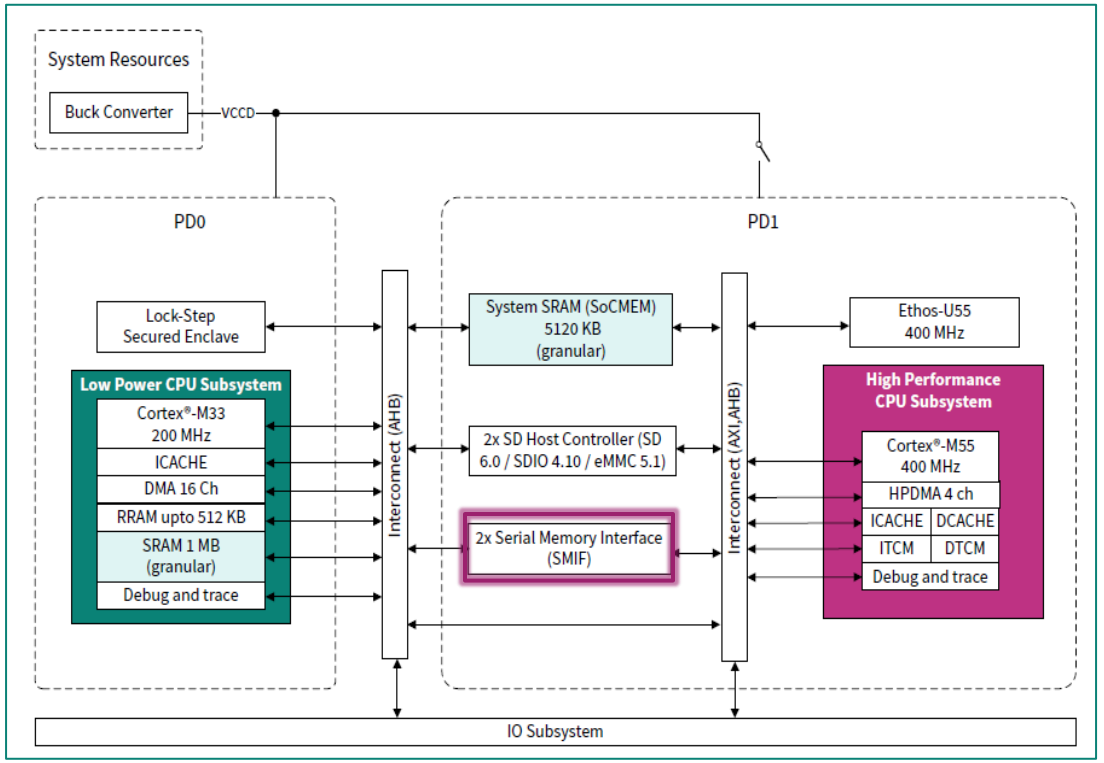
- Integral to the CM55 processor for rapid access to program instructions and data
- Instruction tightly coupled memory (ITCM) and data tightly coupled memory (DTCM) both have 256 KB
- Positioned close to the CPU core to reduce latency
- DTCM for data that demands quick, deterministic access
- ITCM for instructions requiring immediate access without wait states



Copyright © Infineon Technologies AG 2026. All rights reserved.

Serial memory interface (SMIF)

- SMIF interfaces
  - PSOC™ Edge has up to two SMIFs: SMIF0 and SMIF1
  - Availability is package-dependent
- Memory capacity
  - Each interface can access up to 1 GB of address space, or 64 MB in XIP mode
- External device support
  - Compatible with various SPI and HYPERBUS™ memory devices
  - Supports SDR and DDR modes
- Transfer modes
  - XIP for direct memory address execution
  - MMIO for flexibility in device operations
- Security
  - Features on-the-fly encryption and decryption



Copyright © Infineon Technologies AG 2026. All rights reserved.

Serial memory interface – XIP and MMIO

MMIO access to external QSPI flash in SMIF0

```
/* Erase before write */
printf("\r\n1. Erasing %u bytes from offset address 0x%"PRIx32"\r\n",
       sectorSize, ext_mem_address);
result = cy_serial_flash_qspi_erase(ext_mem_address, sectorSize);
check_status("Erasing memory failed", result);

/* Read memory and store in RX buffer */
printf("\r\n2.Reading after Erase\r\n");
result = cy_serial_flash_qspi_read(ext_mem_address, PACKET_SIZE, rx_buf);
check_status("Reading memory failed", result);

/* Write the content of the TX buffer to the memory */
printf("\r\n3. Writing data to offset address 0x%"PRIx32"\r\n",
       ext_mem_address);
result = cy_serial_flash_qspi_write(ext_mem_address, PACKET_SIZE, tx_buf);
check_status("Writing to memory failed", result);
```

XIP access to external OSPI PSRAM in SMIF1

```
printf("\r\nWriting data to XIP memory\n\r");

for (int i = 0; i < RX_BUFFER_SIZE/4; i++)
{
    *write_ptr = test_data;
    write_ptr++;
}

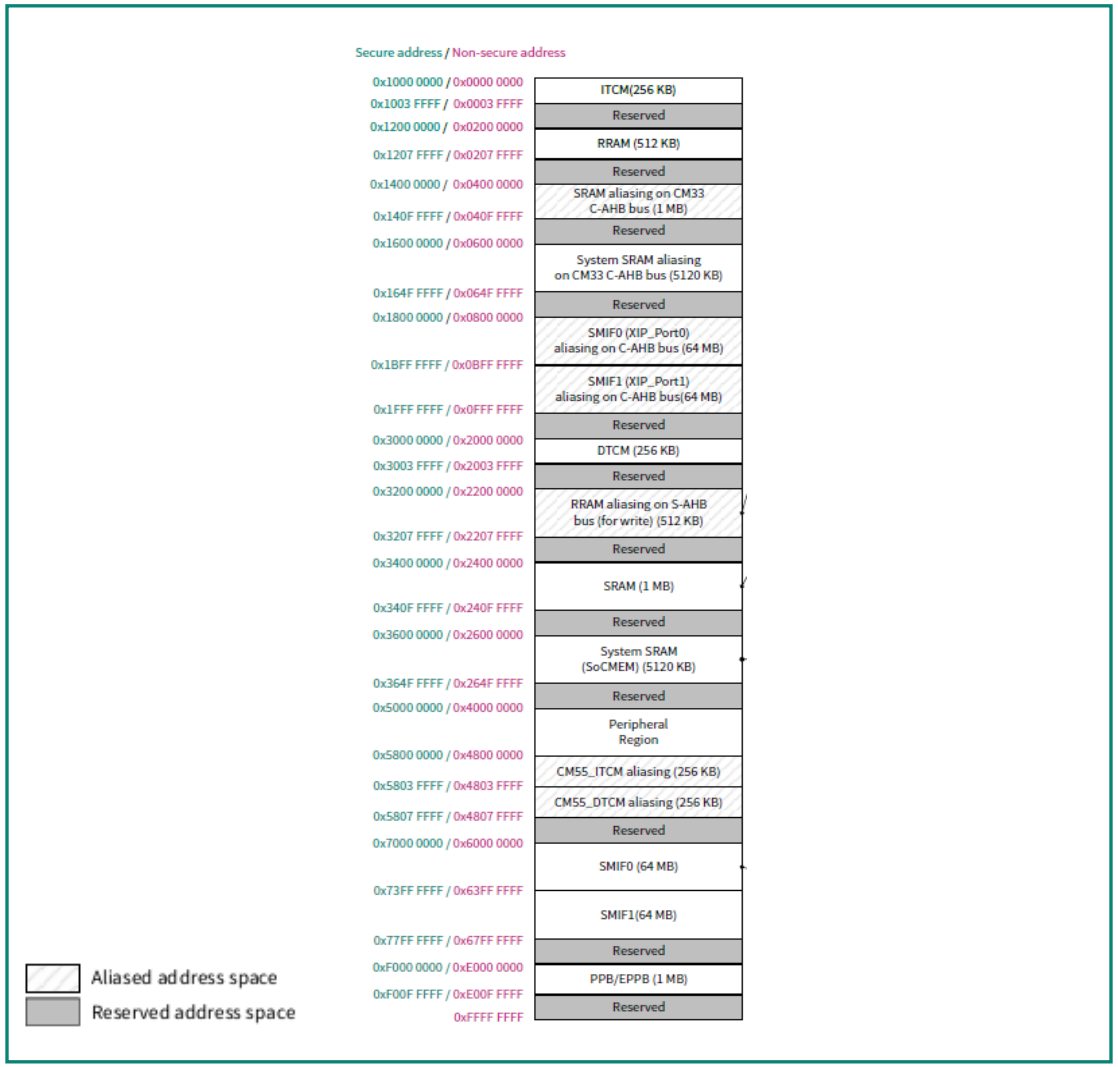
printf("Reading from XIP memory");

/* Print the data in array*/
for (uint32_t index = 0; index < size; index++)
{
    printf("0x%02X ", rx_ptr[index]);
}
```


Copyright © Infineon Technologies AG 2026. All rights reserved.

Default memory map

Description	Global default address		Aliasing	
	Non secure	Secure	Non secure	Secure
RRAM (NVM)	0x0200 0000	0x1200 0000	0x2200 0000	0x3200 0000
SRAM	0x2400 0000	0x3400 0000	0x0400 0000	0x1400 0000
System SRAM (SoCMEM)	0x2600 0000	0x3600 0000	0x0600 0000	0x1600 0000
CM55 ITCM	0x0000 0000	0x1000 0000	Reserved	Reserved
CM55 DTCM	0x2000 0000	0x3000 0000	Reserved	Reserved
SMIF0	0x6000 0000	0x7000 0000	0x0800 0000	0x1800 0000
SMIF1	0x6400 0000	0x7400 0000	Reserved	Reserved

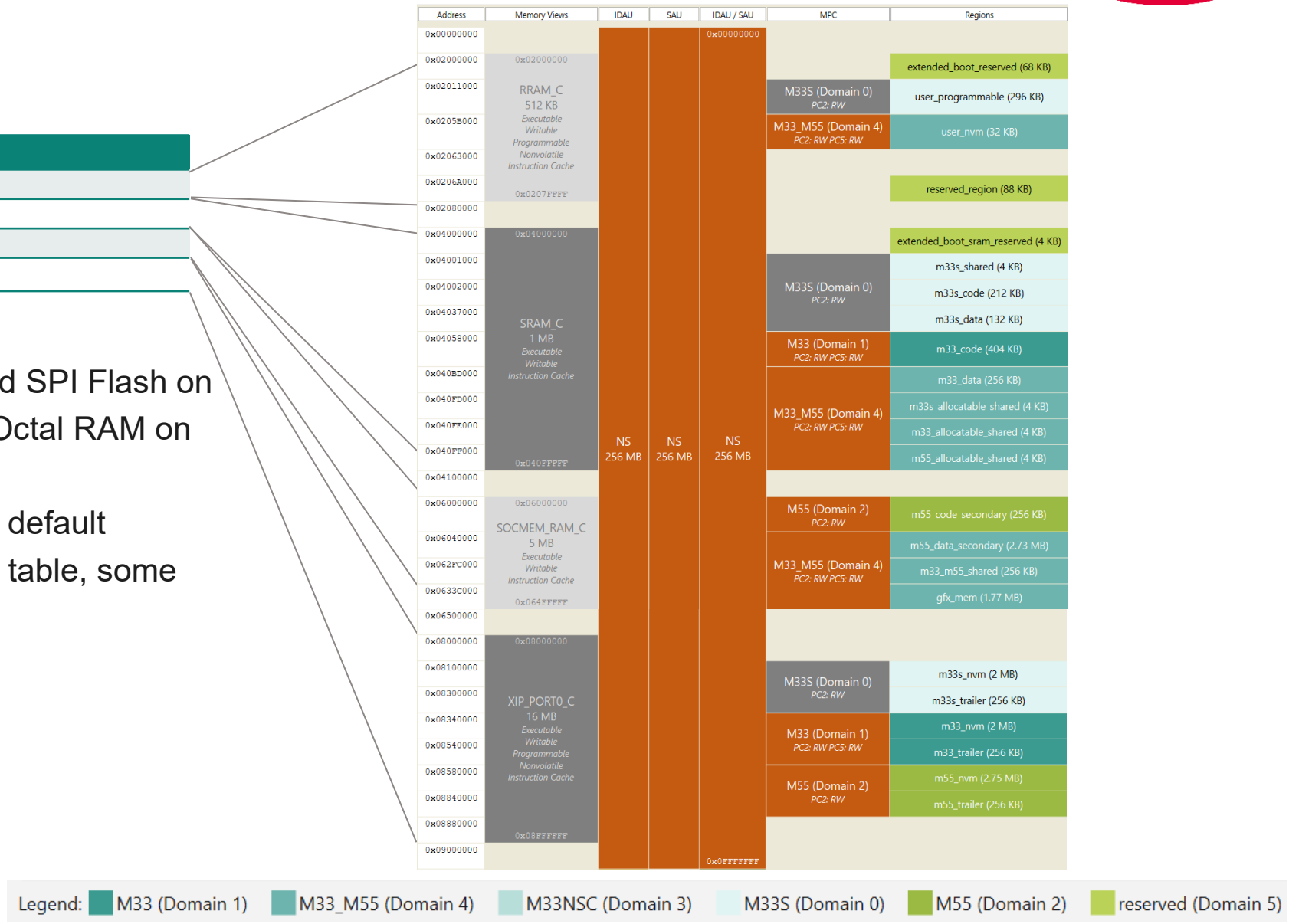


Copyright © Infineon Technologies AG 2026. All rights reserved.

Default memory map

Description	Size
RRAM (NVM)	512 KB
SRAM	1024 KB
System SRAM (SoCMEM)	5012 KB
SMIF0	64 MB

- The PSOC™ Edge EVK has a Quad SPI Flash on SMIF0 enabled by default, and an Octal RAM on SMIF0 not enabled by default
- DTCM is utilized for CM55 stack by default
- ITCM is used to place CM55 vector table, some drivers and lib source files



Copyright © Infineon Technologies AG 2026. All rights reserved.

Table of contents

1 PSOC™ Edge memory architecture 4

2 PSOC™ Edge memory technologies 6

3 **Lab 1: Assigning code and data to custom linker sections** **19**

4 Lab 2: Assigning a source file to a linker section 21

5 PSOC™ Edge bus architecture and memory access 23

6 Lab 3: Placing and executing code from RRAM 28

7 Lab 4: Placing and executing code from SRAM 30

8 PSOC™ Edge Memory tradeoffs 32

9 Lab 5: Low-power memory considerations 36

Copyright © Infineon Technologies AG 2026. All rights reserved.

Lab 1: Assigning code and data to custom linker sections

What will you learn ?

- How to assign functions and variables to specific memory sections
- How to leverage macros to simplify code and data placement
- How to create custom section using ModusToolbox™ Device Configurator

Table of contents

1 PSOC™ Edge memory architecture 4

2 PSOC™ Edge memory technologies 6

3 Lab 1: Assigning code and data to custom linker sections 19

4 **Lab 2: Assigning a source file to a linker section** **21**

5 PSOC™ Edge bus architecture and memory access 23

6 Lab 3: Placing and executing code from RRAM 28

7 Lab 4: Placing and executing code from SRAM 30

8 PSOC™ Edge Memory tradeoffs 32

9 Lab 5: Low-power memory considerations 36

Copyright © Infineon Technologies AG 2026. All rights reserved.

Lab 2: Assigning a source file to a linker section

What will you learn ?

- How to assign files to a memory region
- How to leverage ITCM for CM55 core

Table of contents

1 PSOC™ Edge memory architecture 4

2 PSOC™ Edge memory technologies 6

3 Lab 1: Assigning code and data to custom linker sections 19

4 Lab 2: Assigning a source file to a linker section 21

5 **PSOC™ Edge bus architecture and memory access** **23**

6 Lab 3: Placing and executing code from RRAM 28

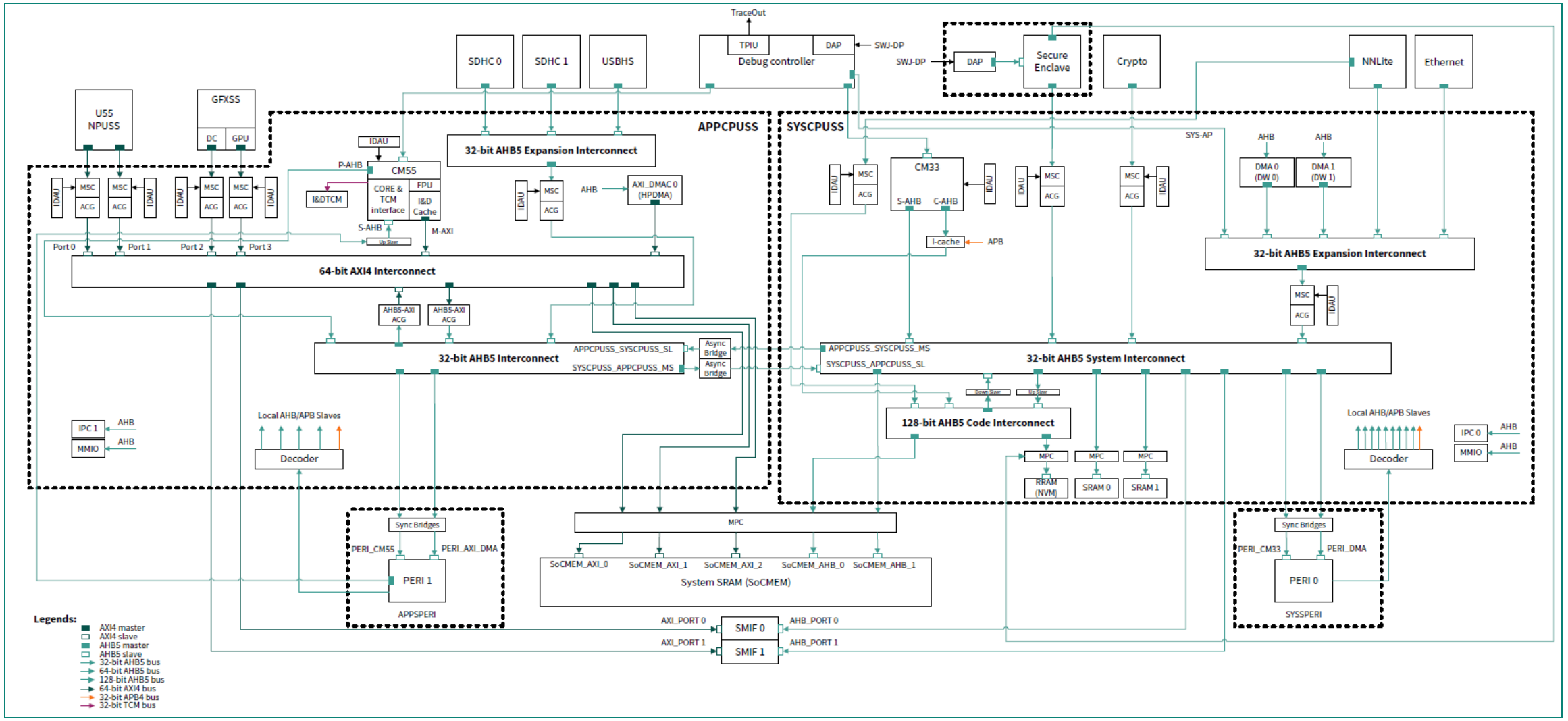
7 Lab 4: Placing and executing code from SRAM 30

8 PSOC™ Edge Memory tradeoffs 32

9 Lab 5: Low-power memory considerations 36

Copyright © Infineon Technologies AG 2026. All rights reserved.

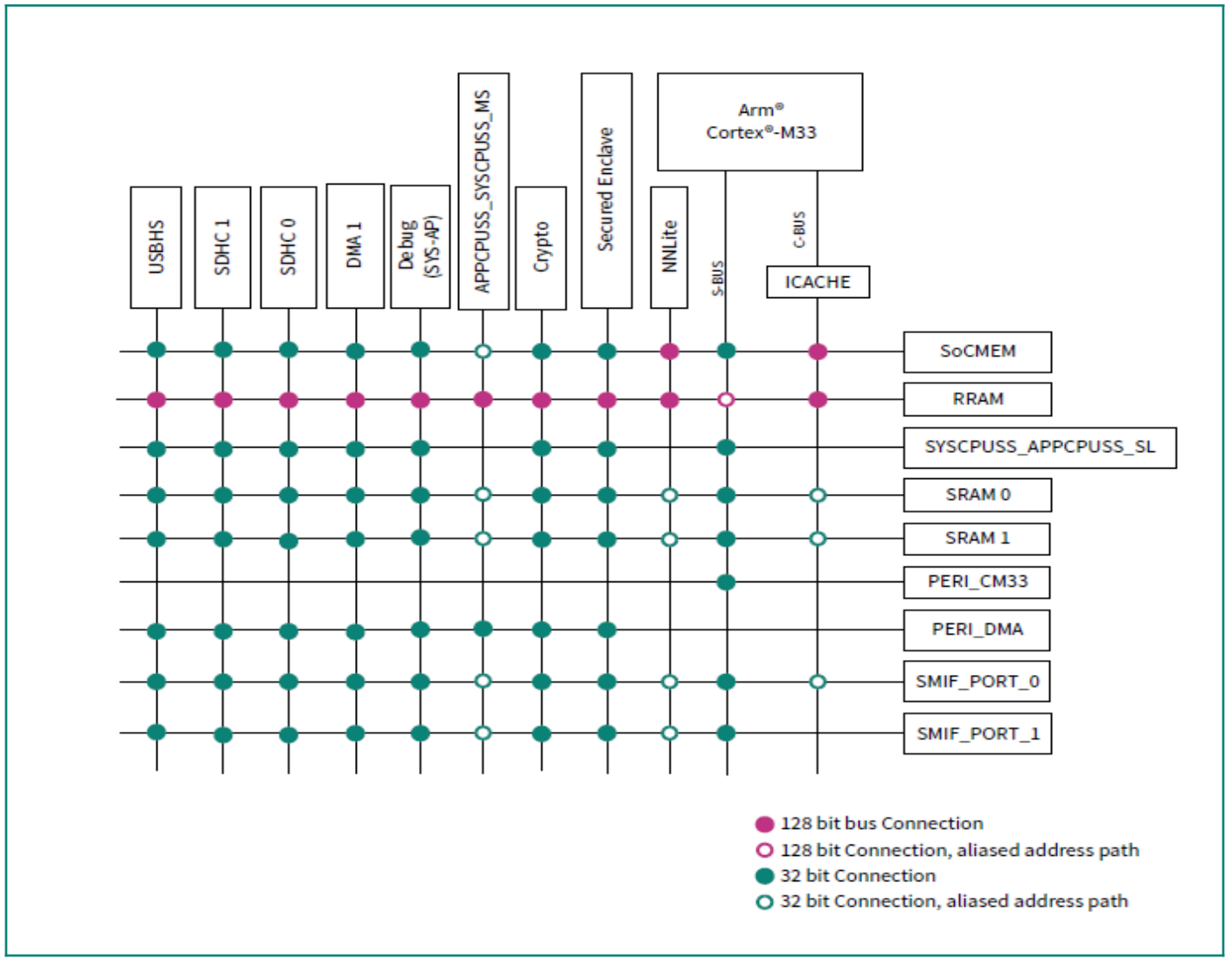
Bus architecture and memory access



Copyright © Infineon Technologies AG 2026. All rights reserved.

SYSCPUSS AHB5 interconnect

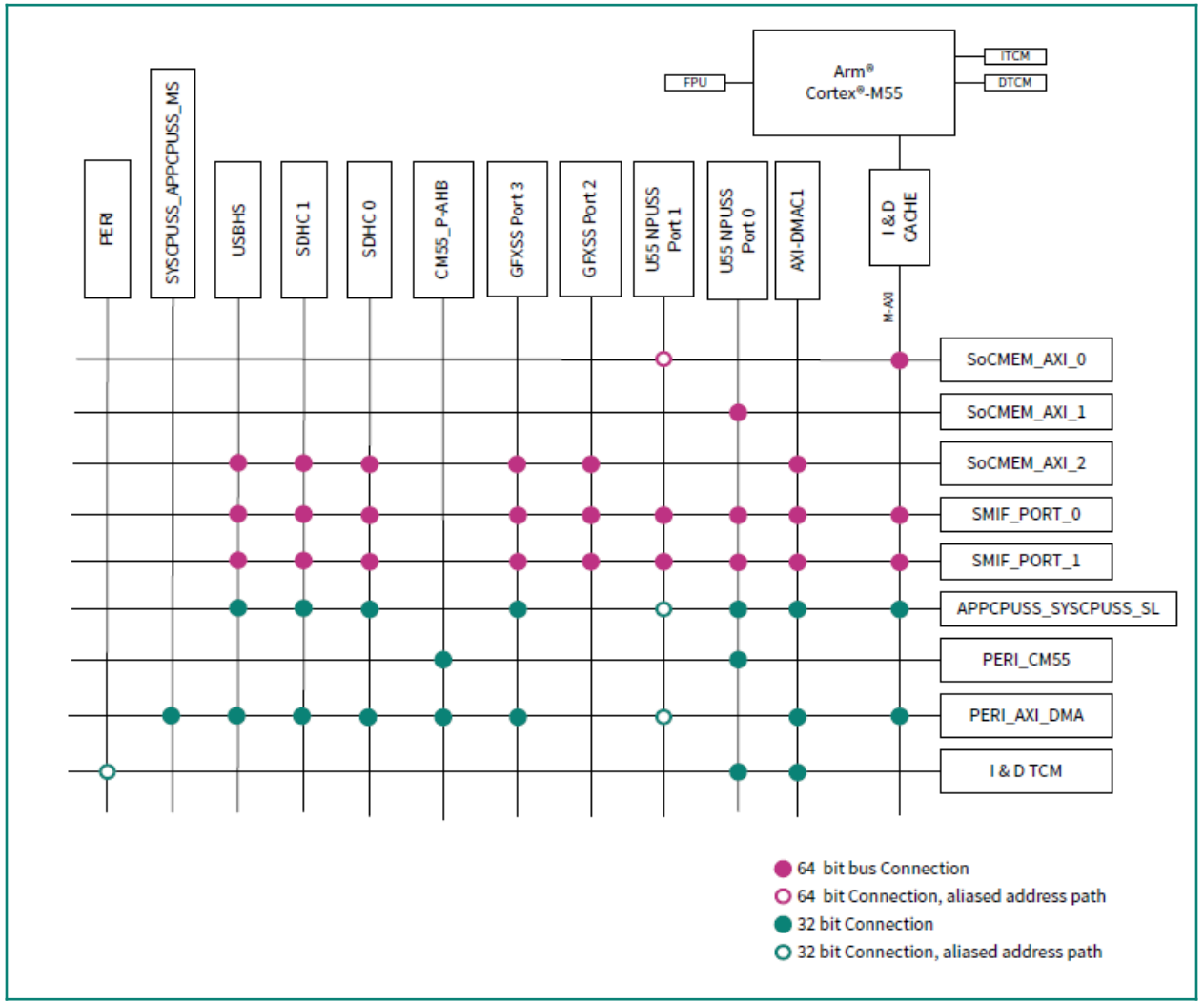
- SYSCPUSS bus system
  - Dual-width AHB5 bus matrix (128/32-bit)
  - Supports internal/external masters and slaves
- Internal masters
  - Cortex®-M33 (C-AHB with ICACHE)
  - Cortex®-M33 (S-AHB)
  - Neural network lite (NNLite)
  - Secured Enclave, CRYPTO modules
- External masters via bridges
  - Bridge to APPCPUSS
  - DMA controller, Dual SDHC, USBHS
- Internal slaves
  - System RAM / SoCMEM (128/32-bit)
  - RRAM controller, SRAM
- Peripheral & External slaves
  - Bridge from APPCPUSS
  - Peripheral and DMA to peripheral interconnects
  - Dual SMIF ports



Copyright © Infineon Technologies AG 2026. All rights reserved.

APPCPUSS AXI4/AHB5 interconnects

- APPCPUSS bus architecture
  - Employs 64-bit AXI4 for memory/devices
  - 32-bit multilayer AHB5 for internal/external bridging
- AXI4/AHB5 masters
  - Cortex®-M55 (M-AXI with I&DCACHE)
  - AXI DMA controller
  - U55 NPUSS (2 ports)
  - Graphics subsystem (2 ports)
  - Cortex®-M55 (P-AHB)
  - Dual SDHC controllers
  - USBHS controller
  - Bridge interface to SYSCPUSS (M55 via AHB5)
- AXI4/AHB5 slaves
  - System RAM / SoCMEM (Three AXI4 ports)
  - SMIF (two ports)
  - Bridge interface to SYSCPUSS, PERI interconnects
  - AXI DMA to PERI interconnect, CM55 ITCM and DTCM



Copyright © Infineon Technologies AG 2026. All rights reserved.

Delays through network inter-connect (NIC)

CM33

Memory resource	Outbound clock cycles	Access time	Inbound clock cycles	Comment
RRAM	3 AHB clocks	Ceil (30 ns / AHB Tcyc)	2 AHB clocks	128-bit C-AHB. I-cache refill. 32-bit access is similar
SRAM	-	-	1 AHB clock	No wait states
System SRAM	3 AHB clocks	6 System SRAM clocks	2 AHB clocks	32-bit access and 128-bit access (parallel bank read)

CM55

Memory resource	Outbound clock cycles	Access time	Inbound clock cycles	Comment
RRAM	3 AXI + 5 AHB cycles	2 x Ceil (30 ns / AHB Tcyc)	4 AHB + 4 AXI cycles	64-bit word read takes 2x cycles
SRAM	3 AXI + 3 AHB clocks	4 AHB cycles (2 accesses)	7 AXI clocks + 3 AHB clocks	64-bit word read takes 2x cycles
System SRAM	3 AXI clocks	6 system SRAM clocks	2 AXI clocks	The first word access takes 6 System SRAM clocks, and each subsequent access requires one system SRAM clock cycle

Copyright © Infineon Technologies AG 2026. All rights reserved.

Table of contents

1 PSOC™ Edge memory architecture 4

2 PSOC™ Edge memory technologies 6

3 Lab 1: Assigning code and data to custom linker sections 19

4 Lab 2: Assigning a source file to a linker section 21

5 PSOC™ Edge bus architecture and memory access 23

6 **Lab 3: Placing and executing code from RRAM** **28**

7 Lab 4: Placing and executing code from SRAM 30

8 PSOC™ Edge Memory tradeoffs 32

9 Lab 5: Low-power memory considerations 36

Copyright © Infineon Technologies AG 2026. All rights reserved.

Lab 3: Placing and executing code from RRAM

What will you learn ?

– How to place and execute all application code from RRAM

Table of contents

1 PSOC™ Edge memory architecture 4

2 PSOC™ Edge memory technologies 6

3 Lab 1: Assigning code and data to custom linker sections 19

4 Lab 2: Assigning a source file to a linker section 21

5 PSOC™ Edge bus architecture and memory access 23

6 Lab 3: Placing and executing code from RRAM 28

7 **Lab 4: Placing and executing code from SRAM** **30**

8 PSOC™ Edge Memory tradeoffs 32

9 Lab 5: Low-power memory considerations 36

Copyright © Infineon Technologies AG 2026. All rights reserved.

Lab 4: Placing and executing code from SRAM

What will you learn ?

- How to use the Edge Protect Bootloader to launch images from external flash
- How to use the Edge Protect Bootloader to load application images into SRAM

Table of contents

1 PSOC™ Edge memory architecture 4

2 PSOC™ Edge memory technologies 6

3 Lab 1: Assigning code and data to custom linker sections 19

4 Lab 2: Assigning a source file to a linker section 21

5 PSOC™ Edge bus architecture and memory access 23

6 Lab 3: Placing and executing code from RRAM 28

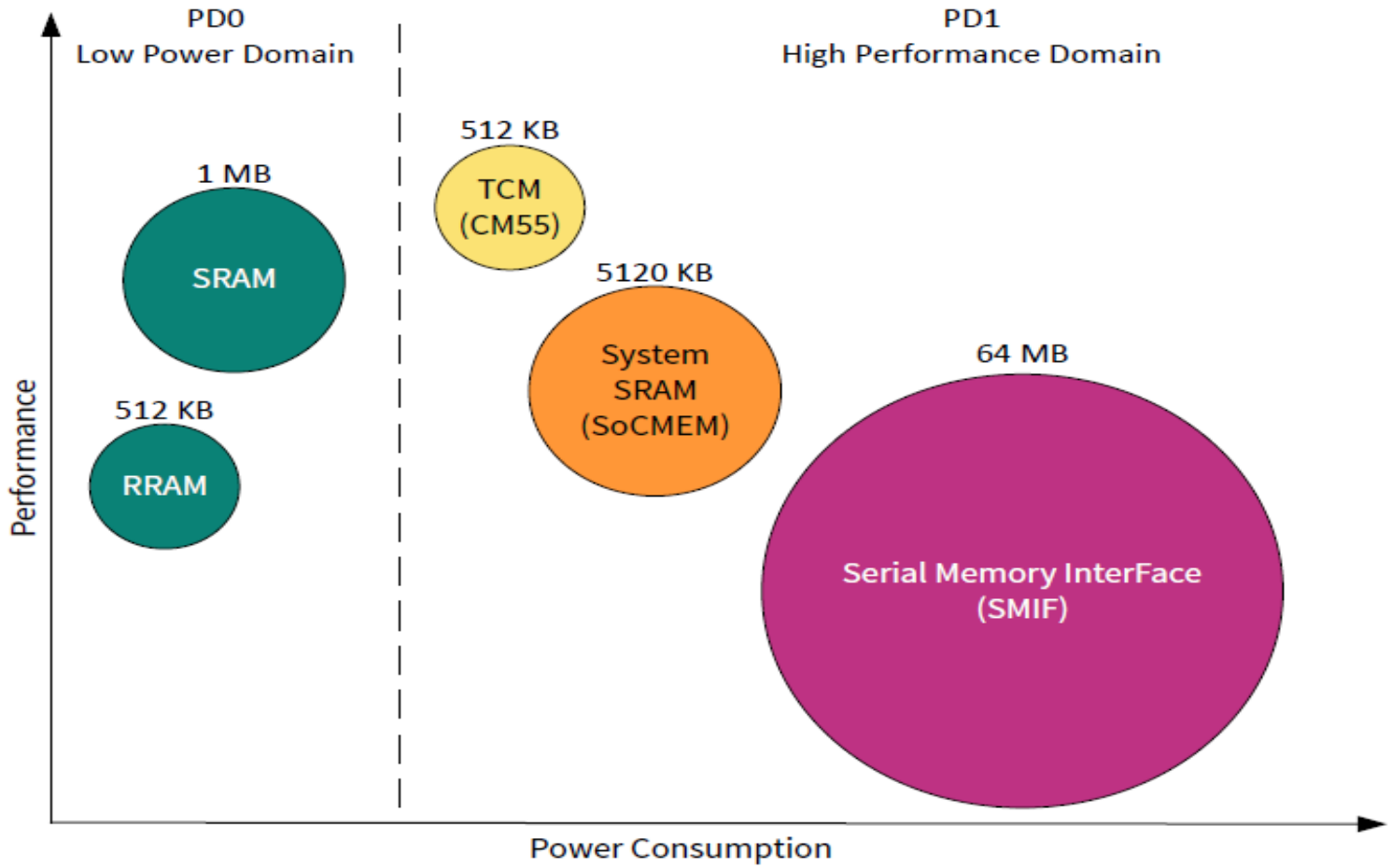
7 Lab 4: Placing and executing code from SRAM 30

8 **PSOC™ Edge Memory tradeoffs** **32**

9 Lab 5: Low-power memory considerations 36

Copyright © Infineon Technologies AG 2026. All rights reserved.

Power and performance impact for accessing memories



Copyright © Infineon Technologies AG 2026. All rights reserved.

Power optimization techniques

1. Turn off system SRAM when not required and execute from RRAM, SRAM, and ITCM
2. Disable the external flash if the application can be completely executed from internal memory
3. Disable high-performance domain (PD1) if CM55, and corresponding peripherals and memories are not required
4. Retain SRAM in blocks of 64 KB for applications not utilising 1 MB
5. Optimise core voltage and frequencies
6. Use DMA for memory transactions while the CPU is put to sleep

Copyright © Infineon Technologies AG 2026. All rights reserved.

Performance optimization techniques

1. Utilize ITCM and DTCM to achieve faster, predictable and deterministic data access using the CM55
2. CM55 can utilize data cache for small repetitive operations
3. Utilize U55 coprocessor to accelerate machine learning applications
4. Utilize Helium extension to accelerate DSP operations
5. When using external memory, utilize SPI protocols with higher speed like Quad over Dual, or Octal over Quad, and higher data rate transfers, such as DDR over SDR
6. Apply compiler optimizations to improve performance or reduce resource usage

Copyright © Infineon Technologies AG 2026. All rights reserved.

Table of contents

1 PSOC™ Edge memory architecture 4

2 PSOC™ Edge memory technologies 6

3 Lab 1: Assigning code and data to custom linker sections 19

4 Lab 2: Assigning a source file to a linker section 21

5 PSOC™ Edge bus architecture and memory access 23

6 Lab 3: Placing and executing code from RRAM 28

7 Lab 4: Placing and executing code from SRAM 30

8 PSOC™ Edge Memory tradeoffs 32

9 **Lab 5: Low-power memory considerations** **36**

Copyright © Infineon Technologies AG 2026. All rights reserved.

Lab 5: Low-power memory considerations

What will you learn ?

- How to adjust SoCMEM retention in system deep sleep
- How to enable and disable SRAM partitions

Revision history

Document revision	Date	Description of change
**	2026-03-09	Initial release

Copyright © Infineon Technologies AG 2026. All rights reserved.


Infineon