Is Your Memory Protected?

Uncovering Hidden Vulnerabilities in Automotive MPUs

Nimrod Stoler & David Lazar

PLAXIDITYX
GO EVERYWHERE

Who we are

Nimrod Stoler
Security Researcher @ PlaxidityX

David Lazar
Embedded Research Team Lead @ PlaxidityX

PlaxidityX

Cybersecurity Solutions for the Automotive Industry:

- Comprehensive cybersecurity products and services
- Automotive intrusion detection systems
- Anti-vehicle theft solutions
- Vulnerability management
- DevSecOps

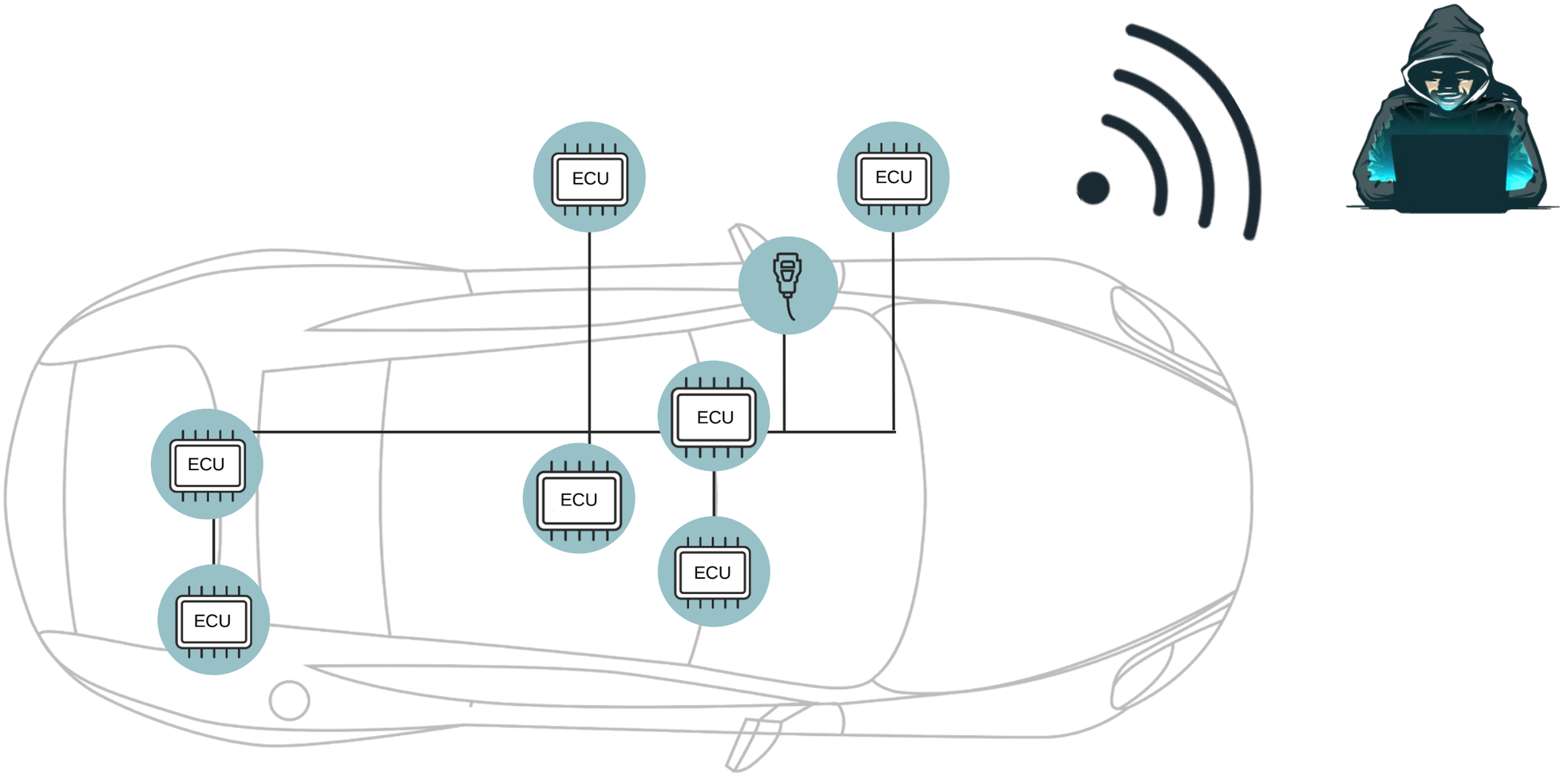
Embedded Research Team

Specializations:

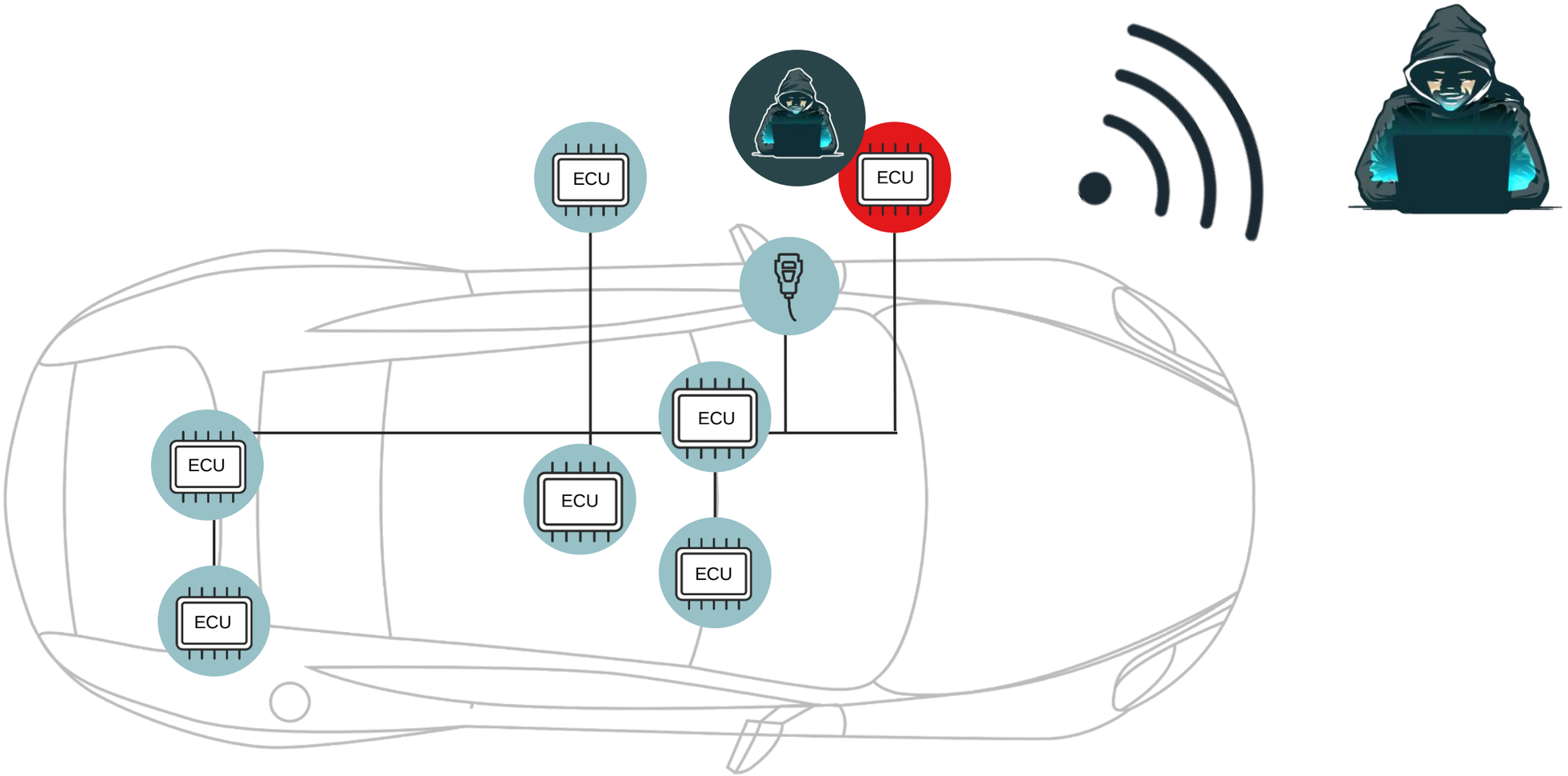
- Automotive vulnerability research & Penetration testing
- Embedded systems & Hardware research
- Reverse engineering
- Fuzzing

Highlights:

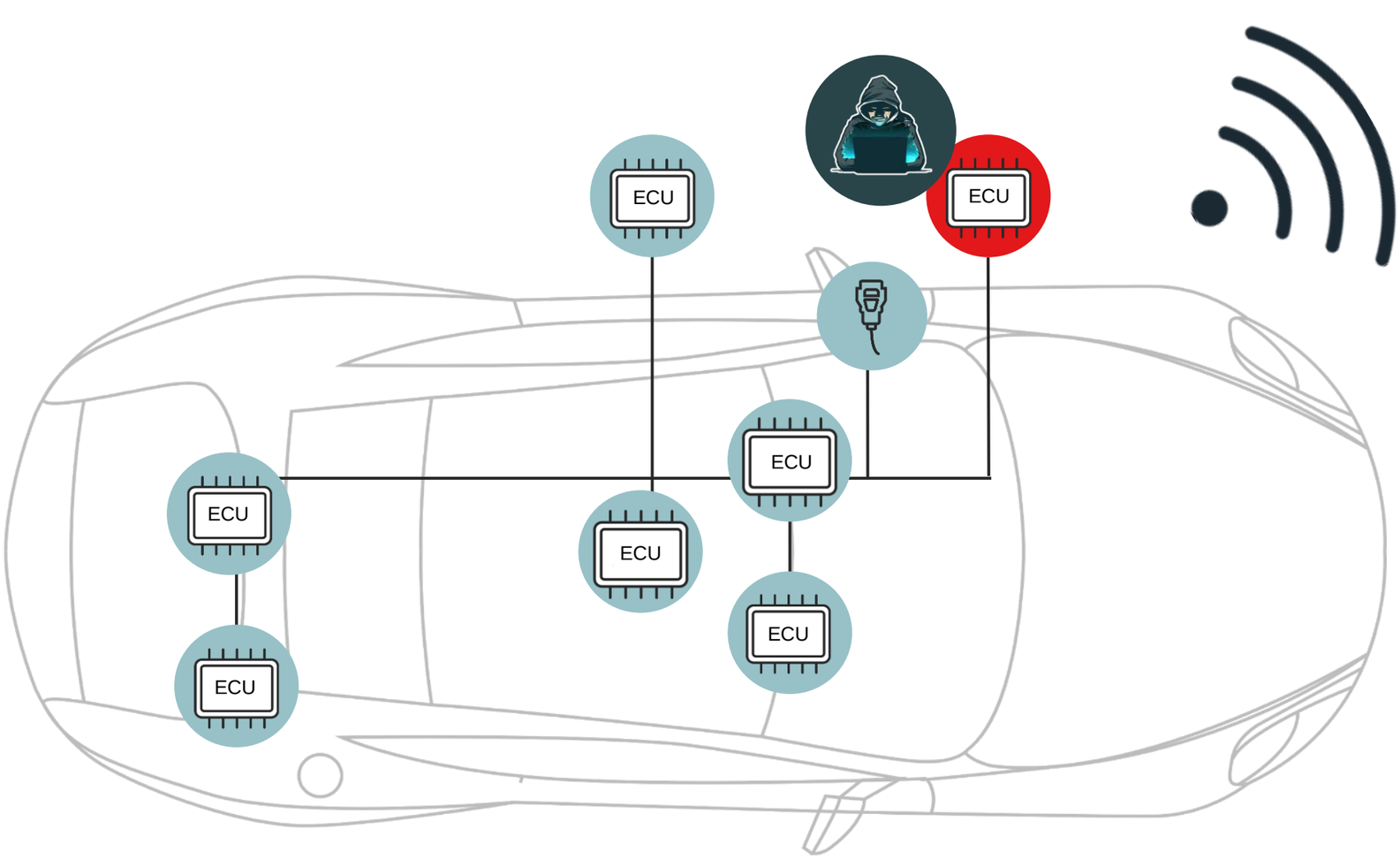
- Over 250,000 hours of combined experience in automotive cybersecurity
- In-depth knowledge of the vehicle cybersecurity lifecycle
- Expertise in automotive architectures, protocols, and standards
- Proficient in UNR 155 & 156, ISO 21434, and related incident and vulnerability management and treatment



ECU – Electronic Control Unit



ECU – Electronic Control Unit



- X Execute code from stack
- X Send messages to ECUs
- X Read sensitive data

ECU – Electronic Control Unit

MPU

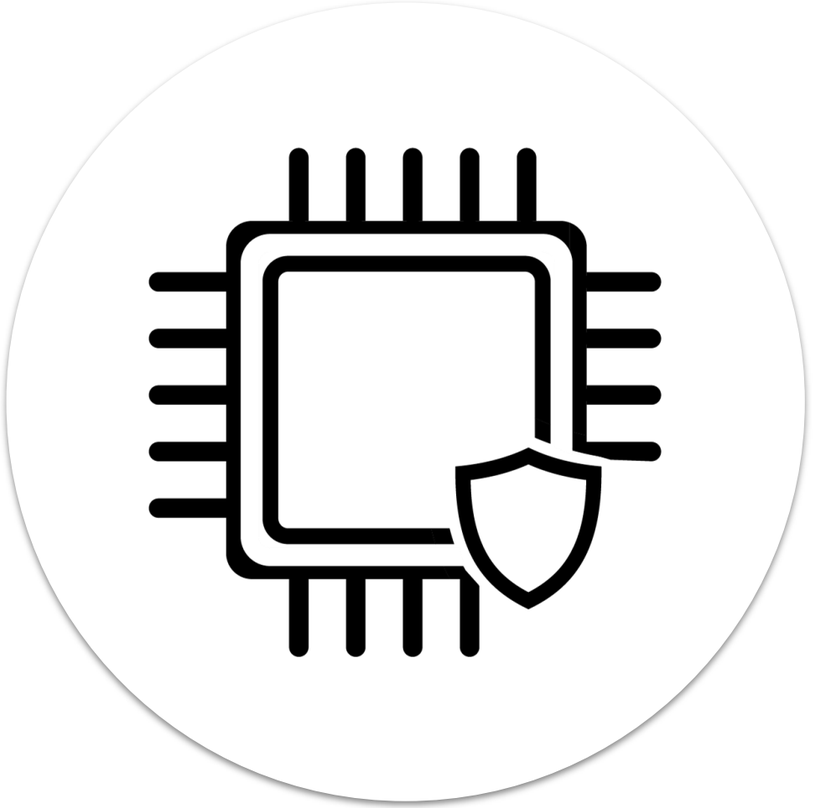
Memory **P**rotection **U**nit

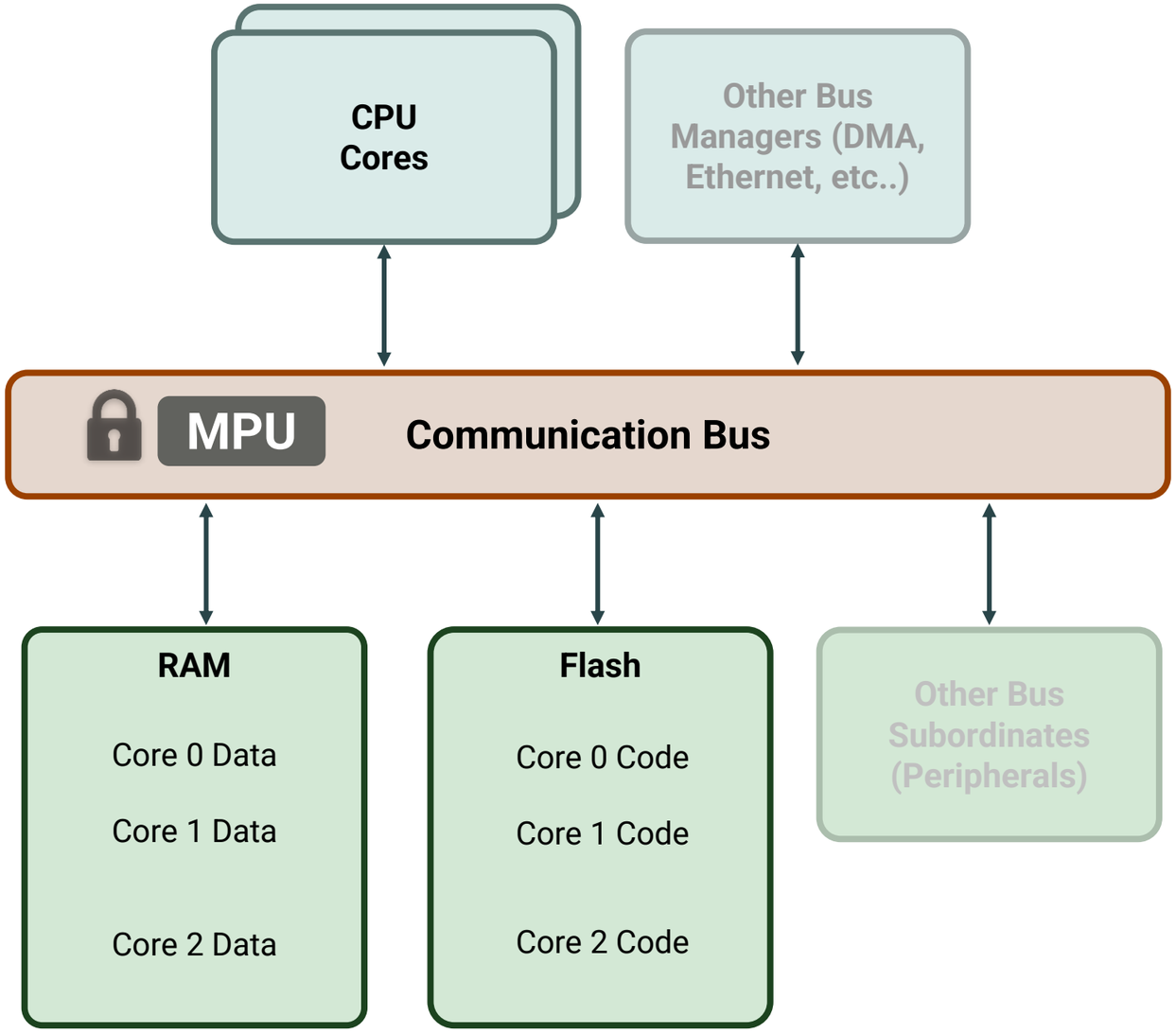
Agenda

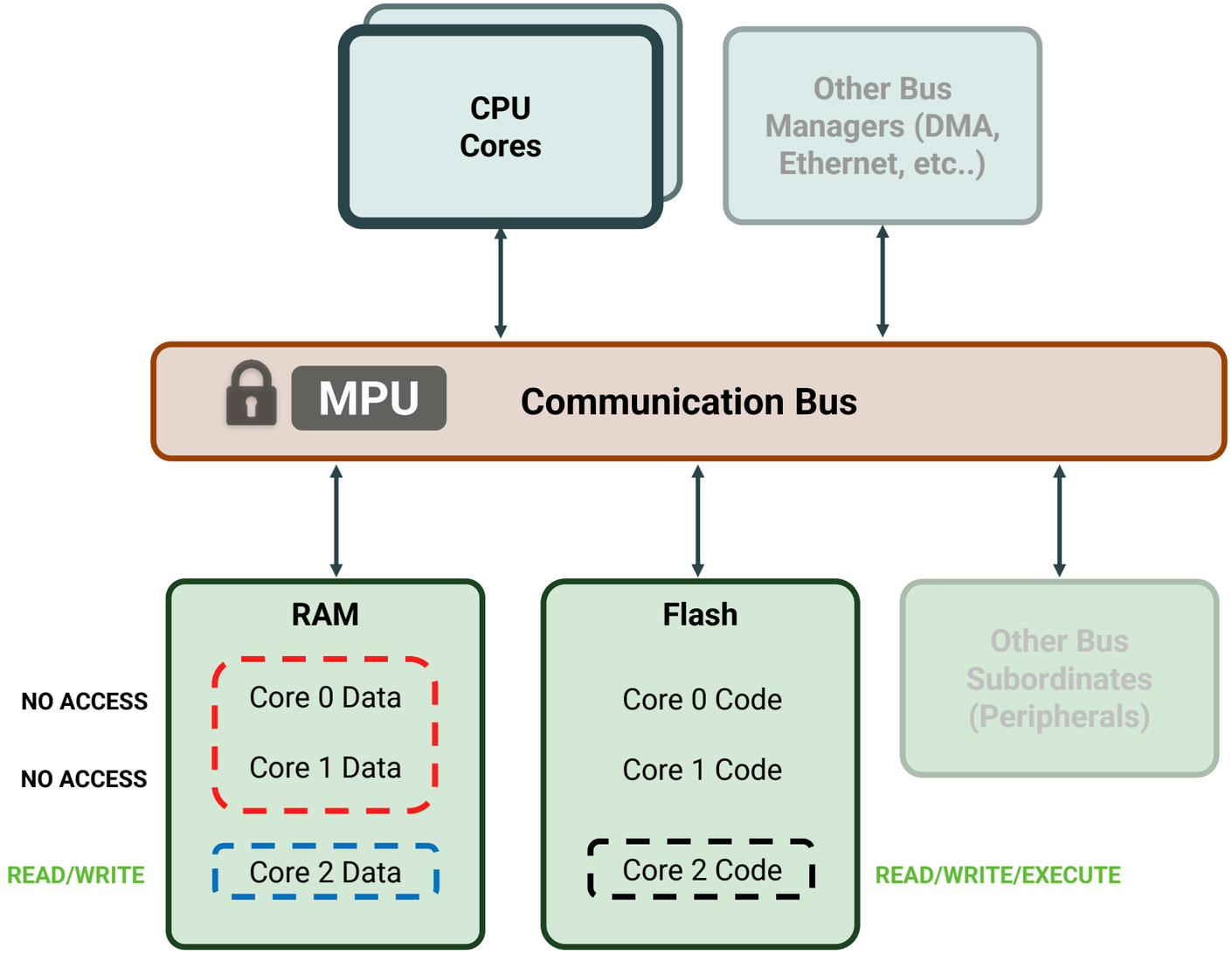
- MPU Introduction & Functionality
- Analysis of the vulnerabilities & Demo
- Disclosure Processes with vendors
- Mitigations and Concluding remarks

What's an MPU?

- **M**emory **P**rotection **U**nit
- Programmable hardware unit that acts as a gatekeeper of memory
- **Divides** memory into **regions**
- For each region, MPUs set:
  - Memory **access permissions** and
  - Memory **attributes**
- MPUs oversee access control to shared memory resources.
  Goal is to **reduce attack surface**







MPU Regions
Region 1: No Access
Region 2: Read/Write
Region 3: Read/Write/Execute

Read Core 1 Data

CPU Cores

Other Bus Managers (DMA, Ethernet, etc..)

MPU

Communication Bus

RAM

NO ACCESS

Core 0 Data

NO ACCESS

Core 1 Data

READ/WRITE

Core 2 Data

Flash

Core 0 Code

Core 1 Code

Core 2 Code

READ/WRITE/EXECUTE

Other Bus Subordinates (Peripherals)

MPU Regions

Region 1: No Access

Region 2: Read/Write

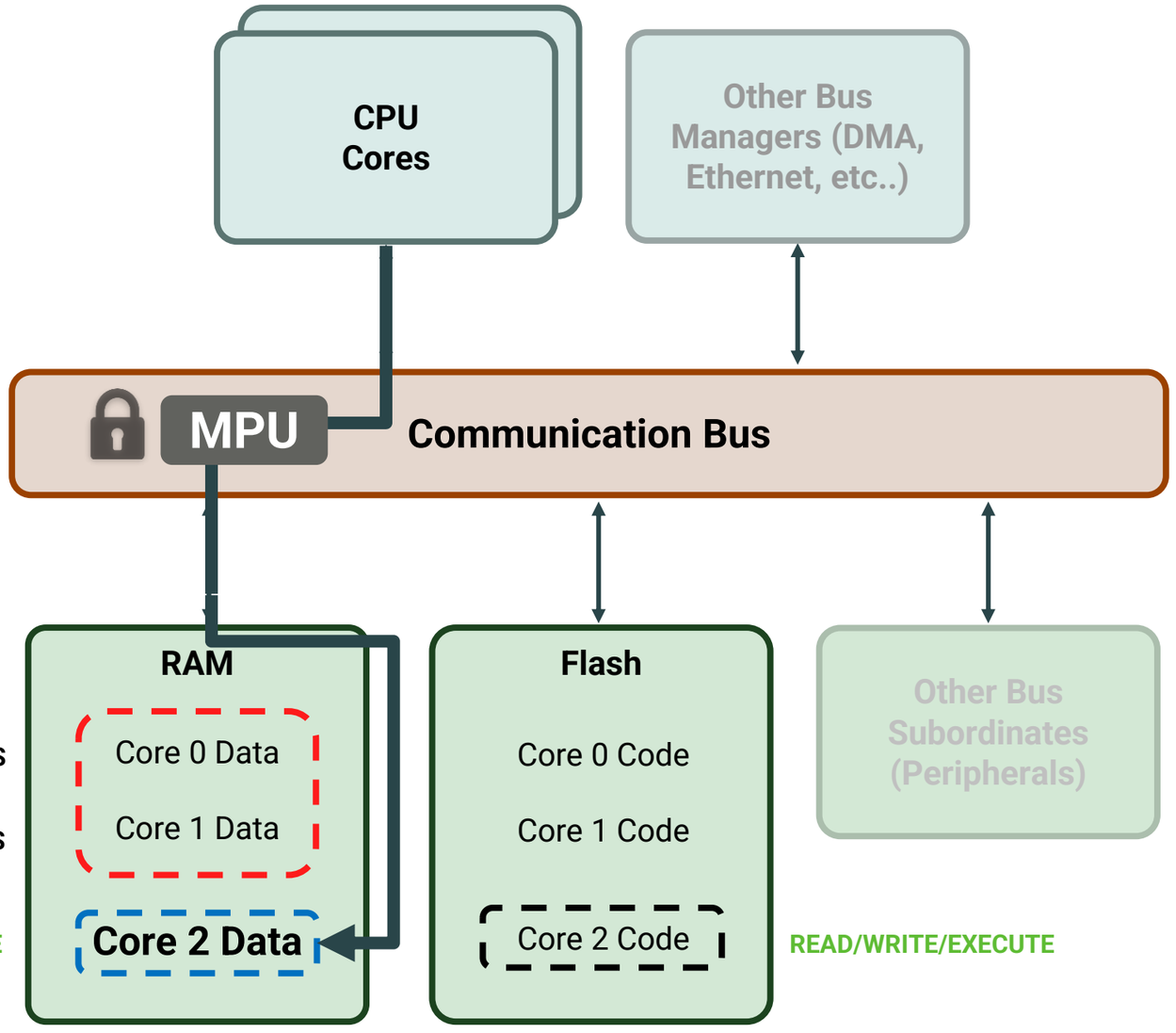
Region 3: Read/Write/Execute

Read Core 2 Data

NO ACCESS

NO ACCESS

READ/WRITE



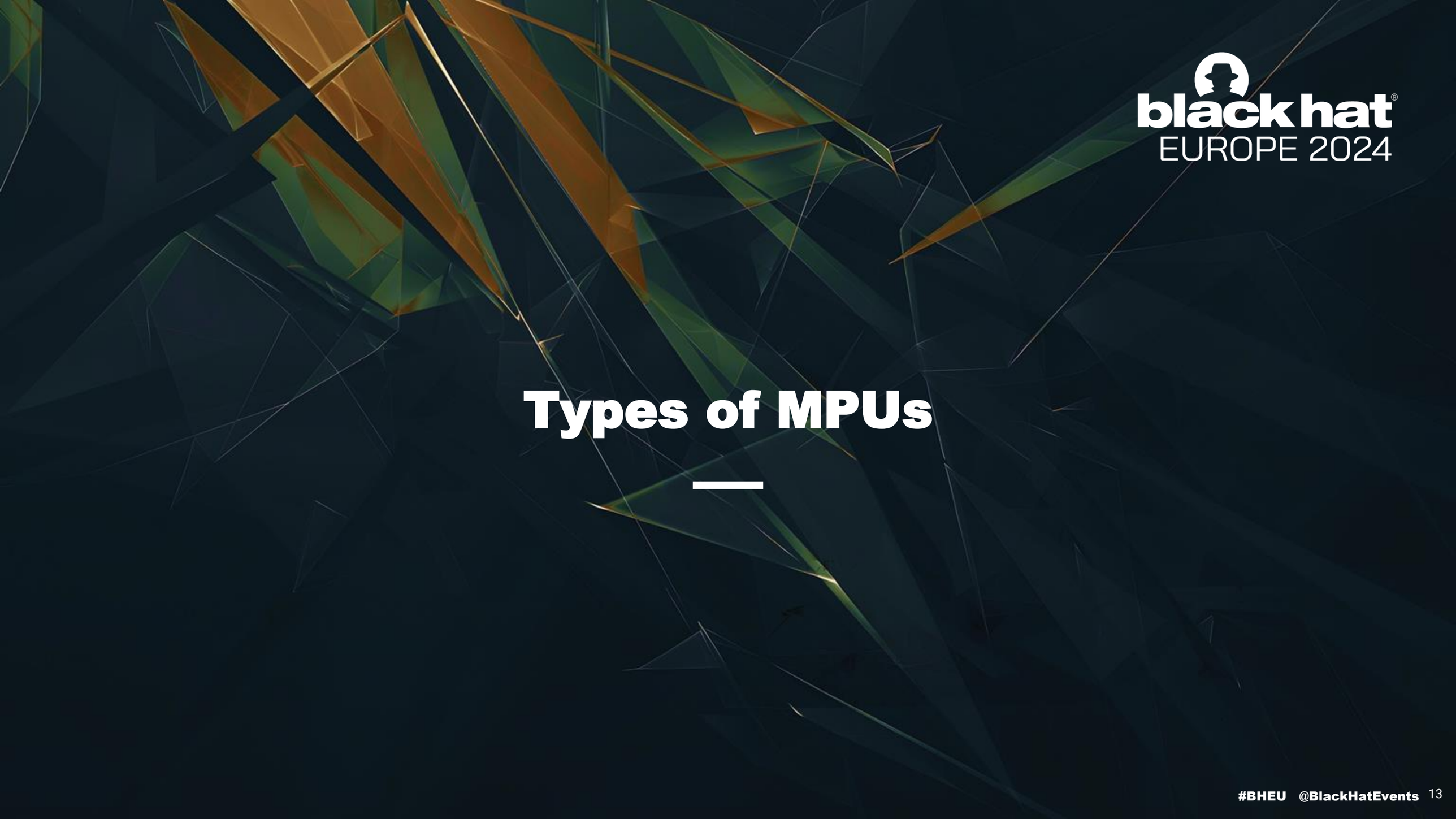
READ/WRITE/EXECUTE

MPU Regions

Region 1: No Access

Region 2: Read/Write

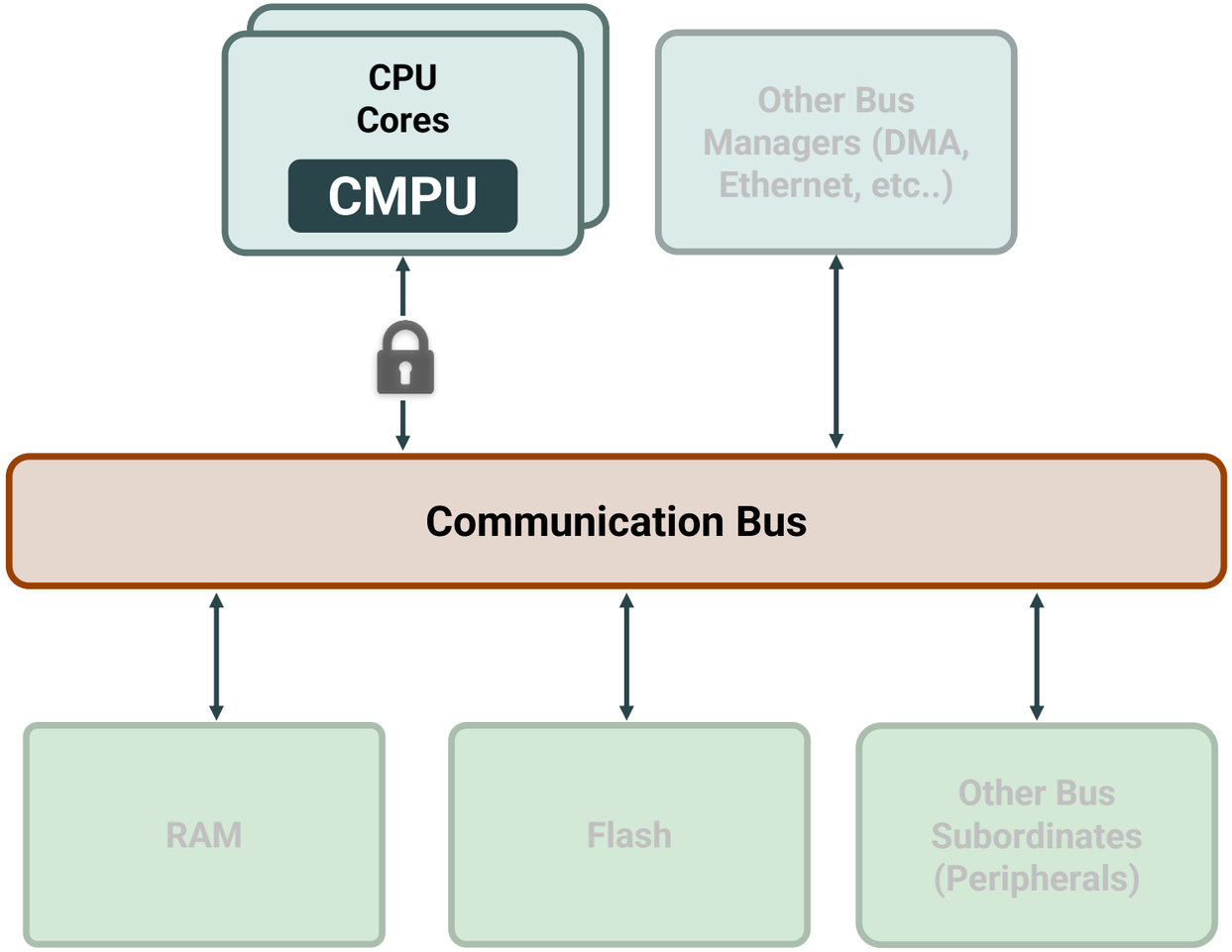
Region 3: Read/Write/Execute



Types of MPUs

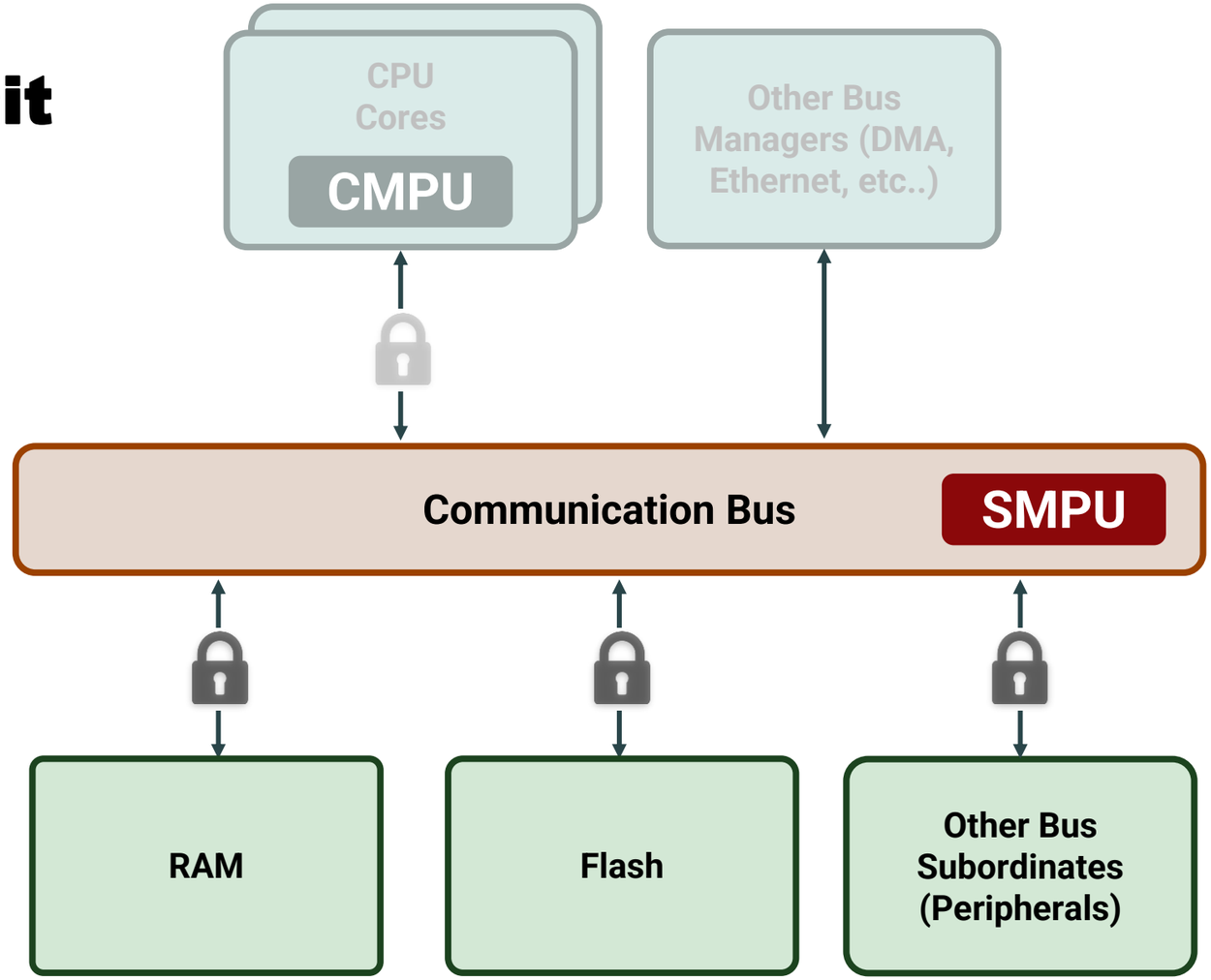
Core Memory Protection Unit

- Also called CMPU or CPU MPU
- **Integrated** into each of the cores
- Controls memory transactions **originating** from each of the cores
- Important attribute is the **“execute” flag**, critical to mitigate buffer overflows

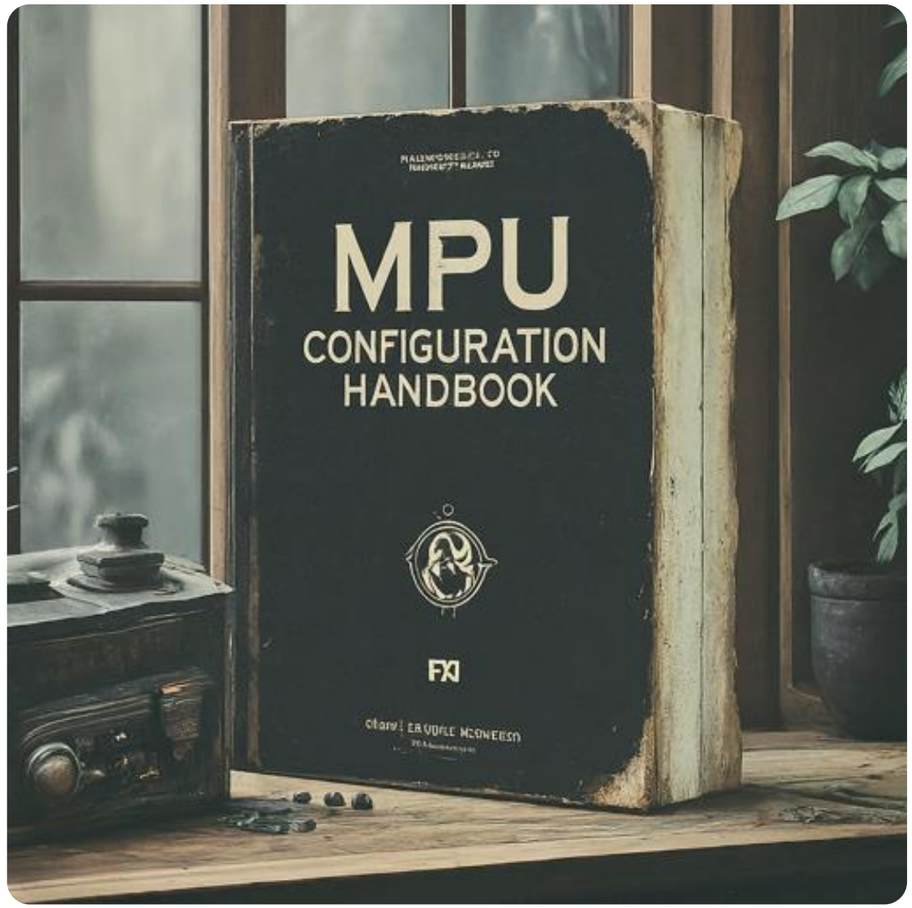


System Memory Protection Unit

- Positioned between memory transaction sources and common memories
- It is **source aware**
- Offers **system wide protection**
- Manages a list of allowed sources, or bus-masters with their attributes



How to configure PowerPC MPUs?



Source: Google’s Gemini

5 / 452 — 136%

452 PAGES

Table of Contents

Book E:
Enhanced PowerPC™ Architecture

Version 1.0

May 7, 2002

1 / 390 — 100%

390 PAGES

UM0434

e200z3 PowerPC core
Reference manual

Introduction

The primary objective of this user's manual is to describe the functionality of the e200z3 embedded microprocessor core for software and hardware developers. This book is intended as a companion to the *EREF: A Programmer's Reference Manual for Freescale Book E Processors* (hereafter referred to as *EREF*).

Book E is a PowerPC™ architecture definition for embedded processors that ensures binary compatibility with the user-instruction set architecture (UISA) portion of the PowerPC architecture as it was jointly developed by Apple, IBM, and Motorola (referred to as the AIM architecture).

This document distinguishes among the three levels of the architectural and implementation definition, as follows:

- The Book E architecture—Book E defines a set of user-level instructions and registers that are drawn from the user instruction set architecture (UISA) portion of the AIM definition PowerPC architecture. Book E also includes numerous supervisor-level registers and instructions as they were defined in the AIM version of the PowerPC architecture for the virtual environment architecture (VEA) and the operating environment architecture (OEA).

  Because the operating system resources (such as the MMU and interrupts) defined by Book E differ greatly from those defined by the AIM architecture, Book E introduces many new registers and instructions.
- Freescale Book E implementation standards (EIS)—In many cases, the Book E architecture definition provides a general framework, leaving specific details up to the implementation. To ensure consistency among its Book E implementations, Freescale has defined implementation standards that provide an additional layer of architecture between Book E and the actual devices.

Table of contents

4 cores 200... 1 / 147 — 100% + | ⟳

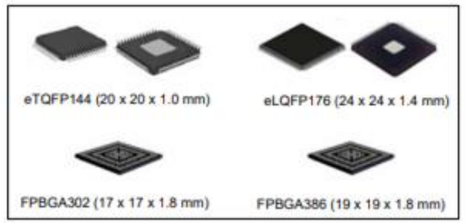
147 PAGES

life.augmented

SPC58EHx, SPC58NHx

SPC58 H Line - 32 bit Power Architecture automotive MCU Triple z4 cores 200 MHz, 10 MBytes Flash, HSM, ASIL-D

Datasheet - production data



Features

- AEC-Q100 qualified
- High performance e200z4 triple core:
  - 32-bit Power Architecture technology CPU
  - Core frequency as high as 200 MHz
  - Variable Length Encoding (VLE)
  - Floating Point, End-to-End Error Correction
- 10496 KB (10240 KB code Flash + 256 KB data Flash) on-chip Flash memory:
  - Supports read during program and erase operations, and multiple blocks allowing EEPROM emulation
- Comprehensive new generation ASIL-D safety concept:
  - ASIL-D of ISO 26262
  - One CPU channel in lockstep
  - Logic BIST
  - FCCU for collection and reaction to failure notifications
  - Memory BIST
  - Cyclic redundancy check (CRC) unit
  - Memory Error Management Unit (MEMU) for collection and reporting of error events in memories
- Crossbar switch architecture for concurrent access to peripherals, Flash, or RAM from multiple bus masters with end-to-end ECC
- Body cross triggering unit (BCTU):
  - Triggers ADC conversions from any eMIOS channel
  - Triggers ADC conversions from up to 2 dedicated PIT_RTIs
- Enhanced modular IO subsystem (eMIOS):
  - up to 96 timed IO channels with 16-bit

Contents

1 / 1439 | − 100% + | ⛶ ⟲

1,439 PAGES

life.augmented

RM0070

Reference manual

SPC564Bxx, SPC56ECxx

32-bit MCU family built on the embedded Power Architecture®

Introduction

The SPC564Bxx and SPC56ECxx is a family of Power Architecture® based microcontrollers that target automotive vehicle body and gateway applications such as:

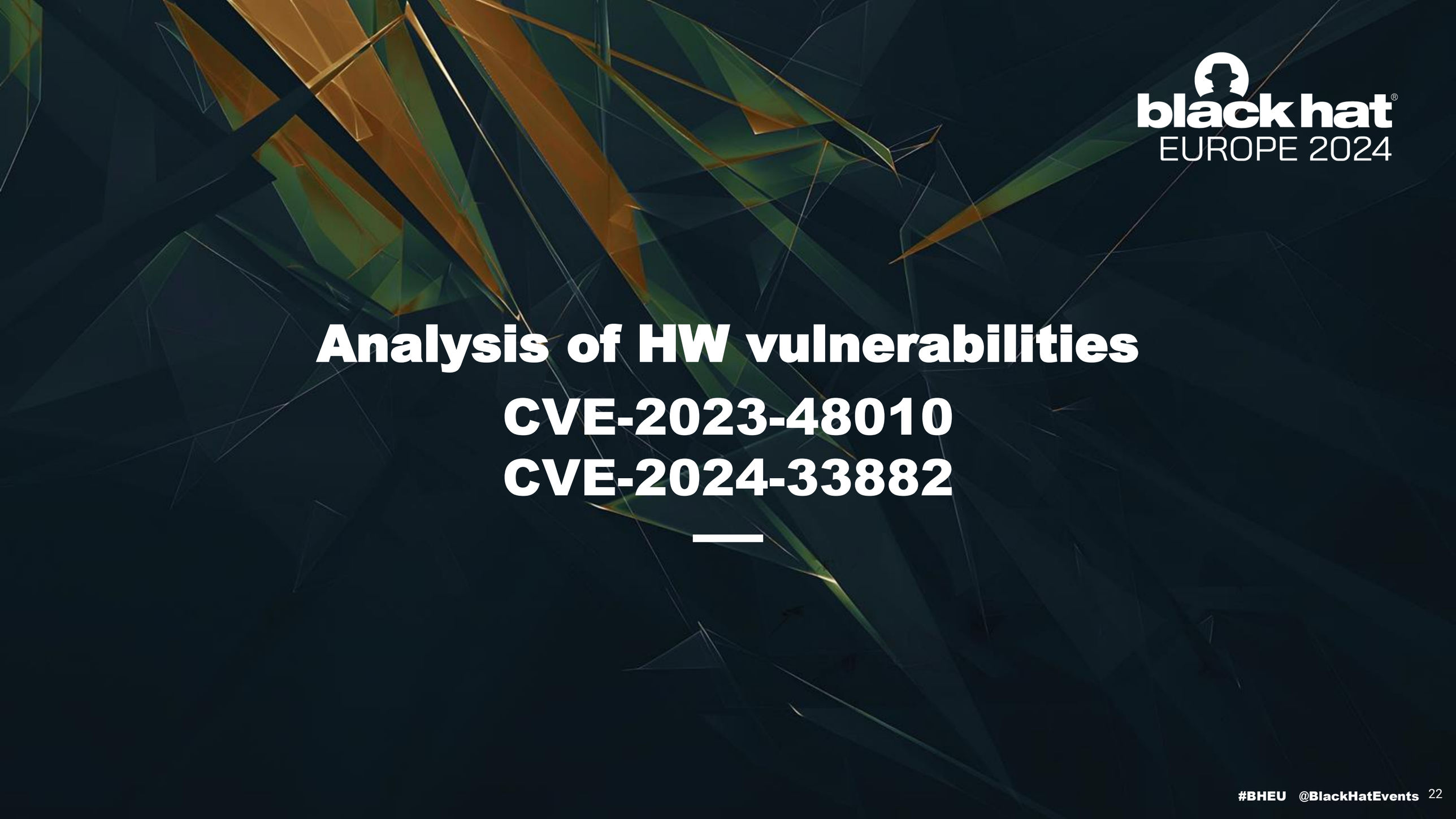
- Central body controller
- Smart junction boxes
- Front modules
- High end gateway
- Combined Body controller and gateway

The SPC564Bxx and SPC56ECxx family expands the range of the SPC560B/C microcontroller family. It provides the scalability needed to implement platform approaches and delivers the performance required by increasingly sophisticated software architectures.

Contents

2428 pages

1 bit



Analysis of HW vulnerabilities

CVE-2023-48010

CVE-2024-33882

- **STMicroelectronic PowerPC Automotive & Industrial MCU**

Source: https://www.st.com/resource/en/brochure/brspc58c.pdf

- STMicroelectronic PowerPC Automotive & Industrial MCU
- **Used in different automotive applications**

life.augmented

SPC58 C-line automotive MCUs for car body and security applications

APPLICATIONS

- Body Control Module
- Smart Junction Box
- Gateway
- Comfort Module
- Door Module
- Seat Module
- Lighting
- Battery Management
- Control Panel

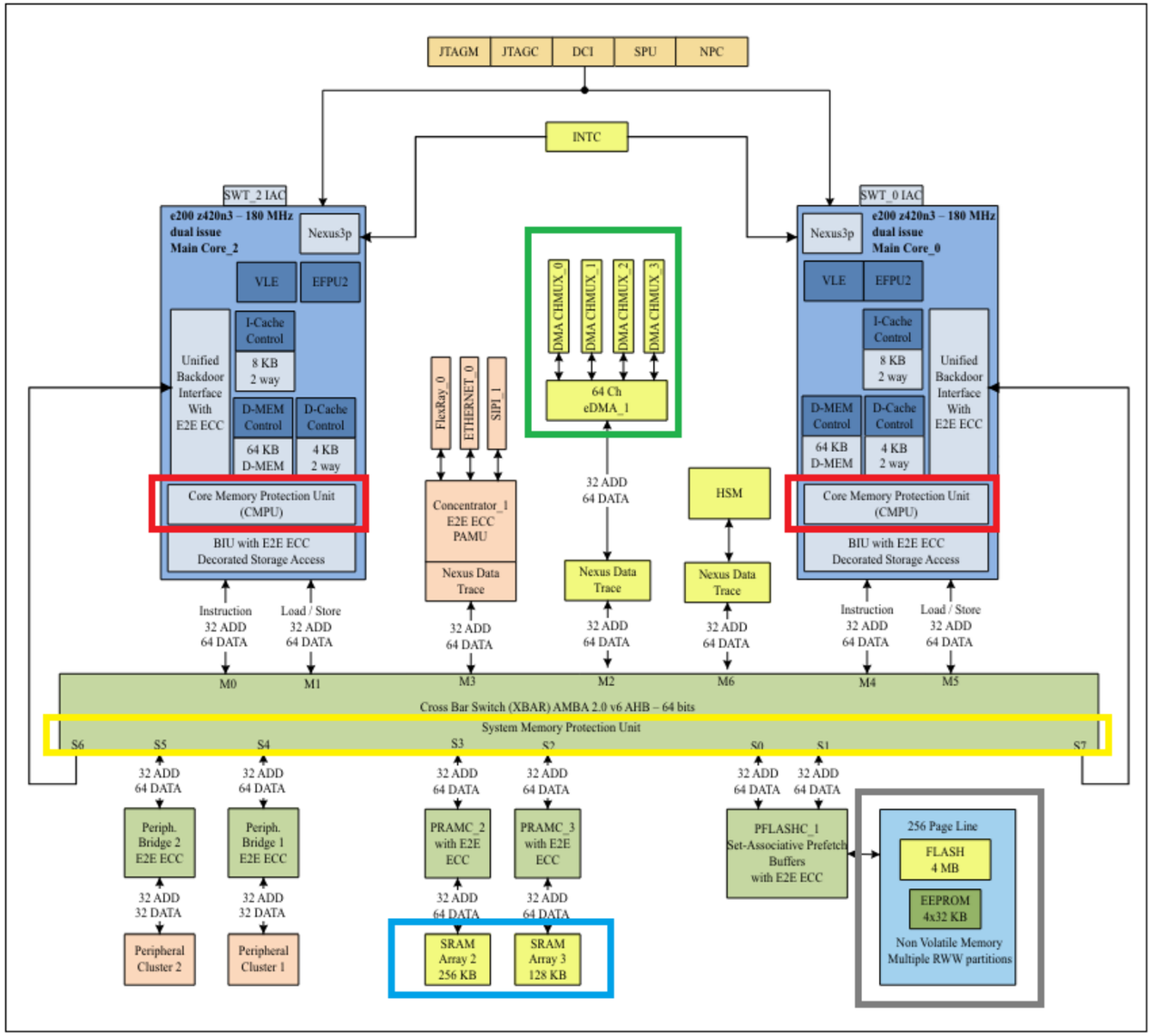
Source: https://www.st.com/resource/en/brochure/brspc58c.pdf

- STMicroelectronic PowerPC Automotive & Industrial MCU
- Used in different automotive applications
- **Affected STM Parts:**
  - All SPC58 devices
  - SR5E1
  - SPC574K (K2)
  - SPC572L (Lavaredo)
  - SPC574Sx (Sphaero)

Source: https://www.st.com/resource/en/brochure/brspc58c.pdf

STMicroelectronics SPC58xx Power PC

Figure 1. Block diagram



Source: SPC584C Datasheet, DS11620 Rev 8, p. 9

STMicroelectronics SPC58xx Power PC

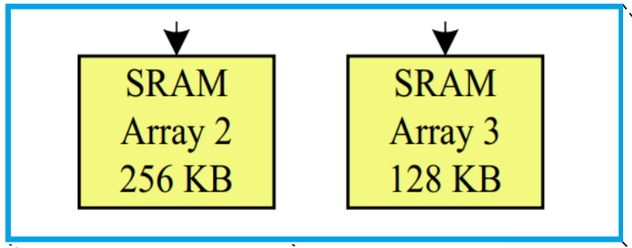
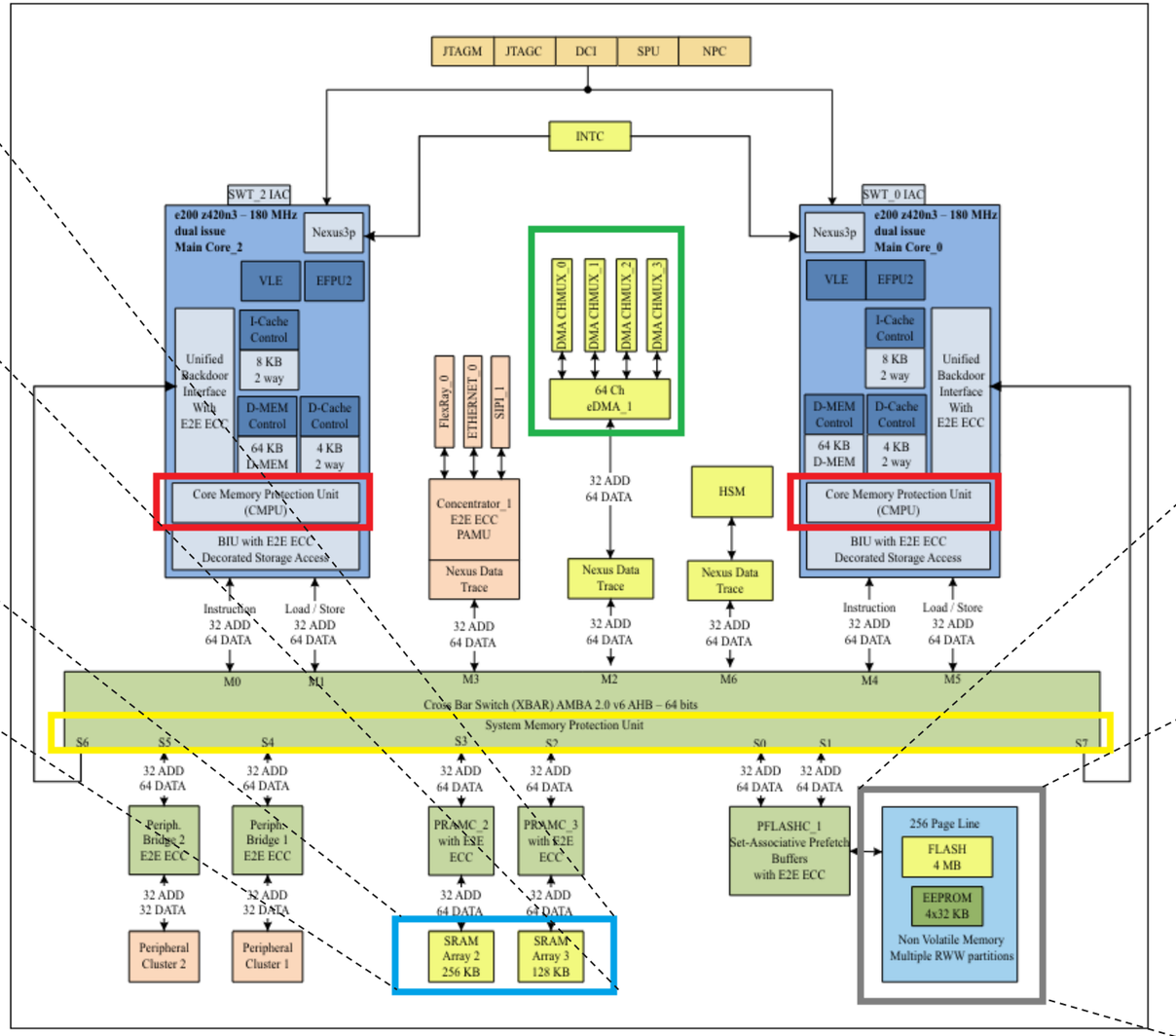
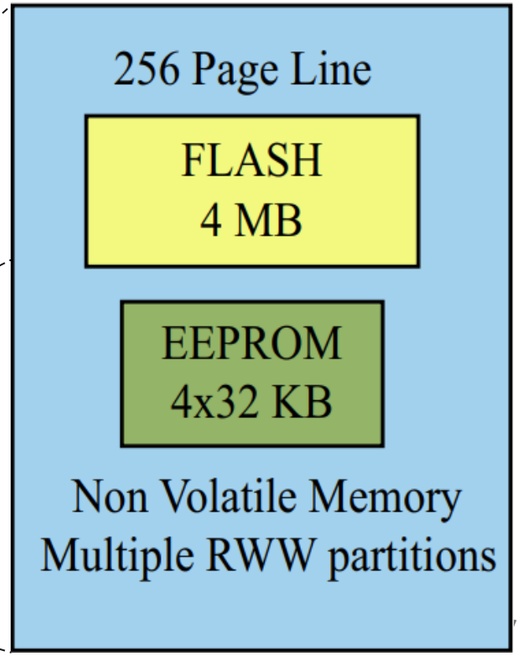


Figure 1. Block diagram





Source: SPC584C Datasheet, DS11620 Rev 8, p. 9

STMicroelectronics SPC58xx Power PC

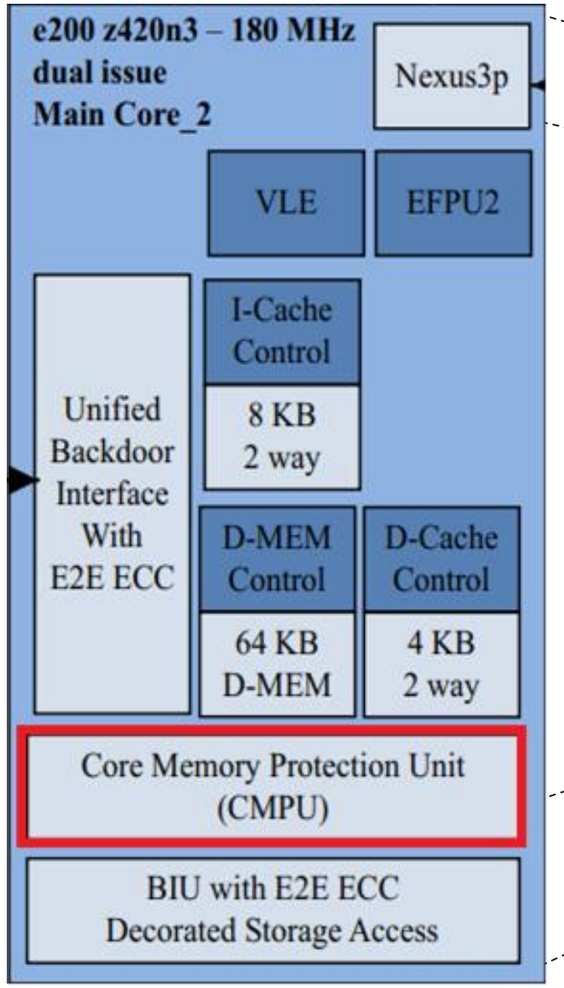
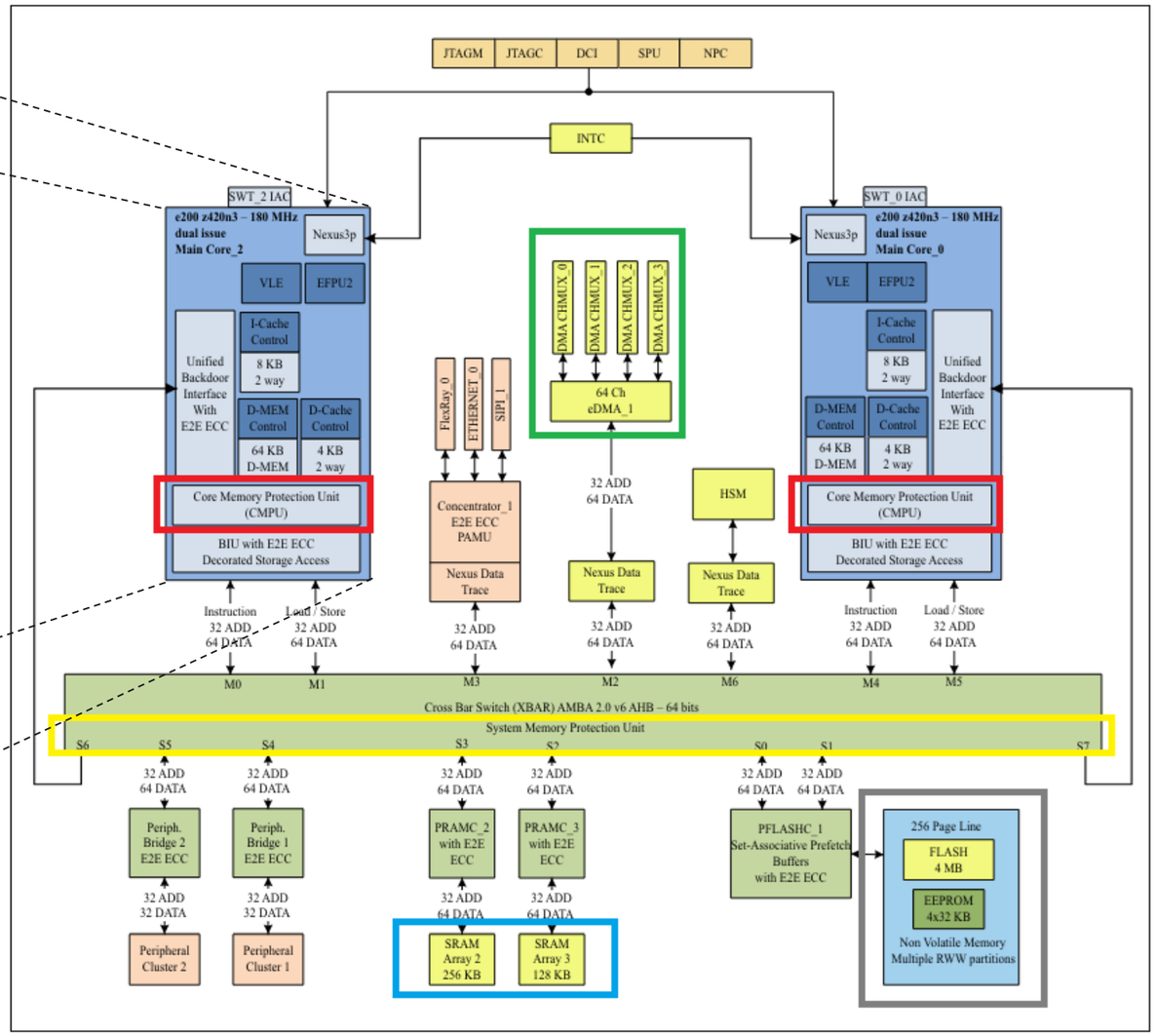
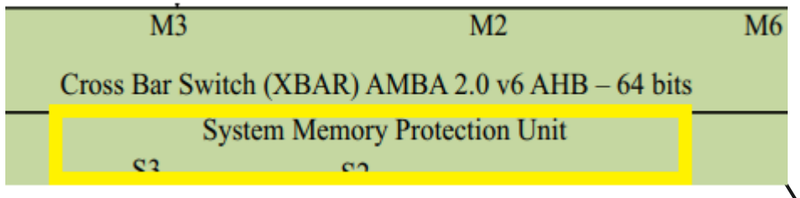


Figure 1. Block diagram



Source: SPC584C Datasheet, DS11620 Rev 8, p. 9

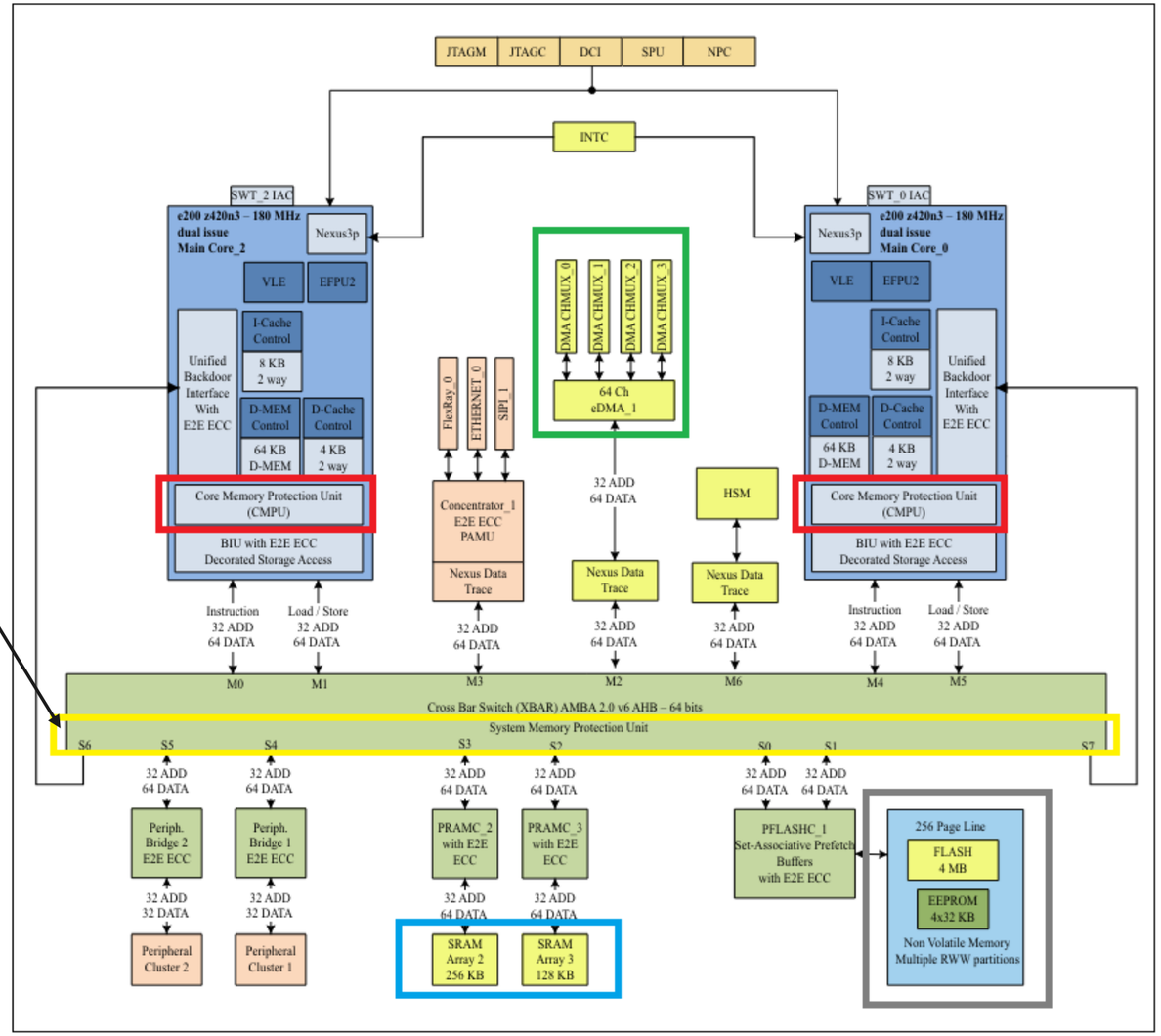
STMicroelectronics SPC58xx Power PC



Per Each SMPU Region:

- Start address
- End address
- List of allowed **Sources** and their attributes

Figure 1. Block diagram



Source: SPC584C Datasheet, DS11620 Rev 8, p. 9

SMPU Bus-Masters

7.3.3.5 AHB Master ID Assignments

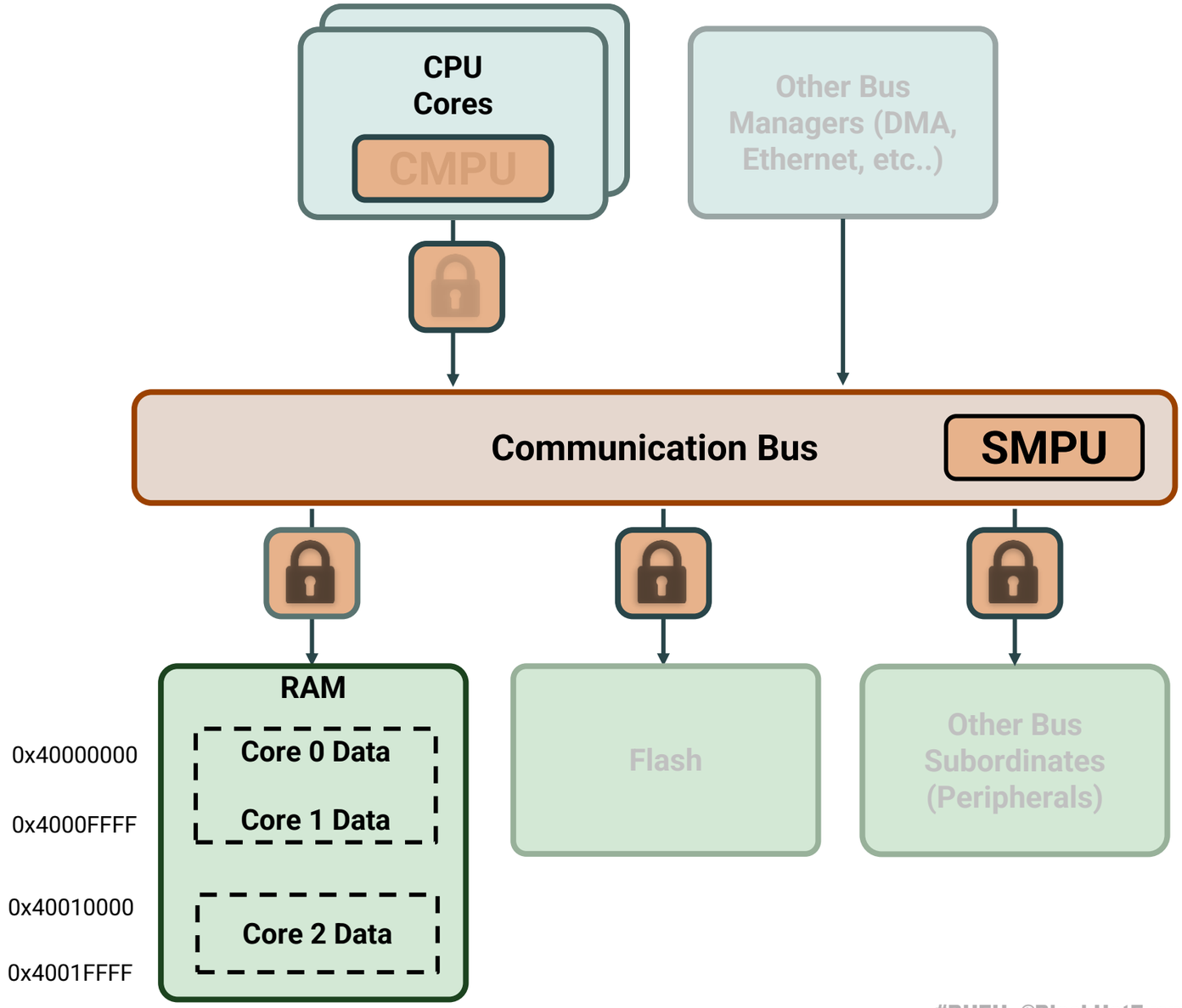
Table 34. AHB Master ID Assignments

SMPU logical bus master number	Bus master
0	Core0 (z42560n3_0)
1	Core1 (z4256n3_1)
2	Core2 (z4256n3_2)
3	eDMA_0
4	Ethernet_0
5	FlexRay_0
6	SDMMC
8	Core0 Debug
9	Core1 Debug
10	Core2 Debug
11	eDMA_1
12	—
13	Ethernet_1
14	HSM
15	—

Source: RM0452-spc58-line Rev.4 p. 136

Configuration Steps

1. Define the region descriptors:
   - Set start address & end address for each region
   - List allowed bus masters and their access attributes
2. Lock the RGDs using RO field
3. Enable the SMPU by setting the GVLD bit
4. SMPU is now enabled and locked

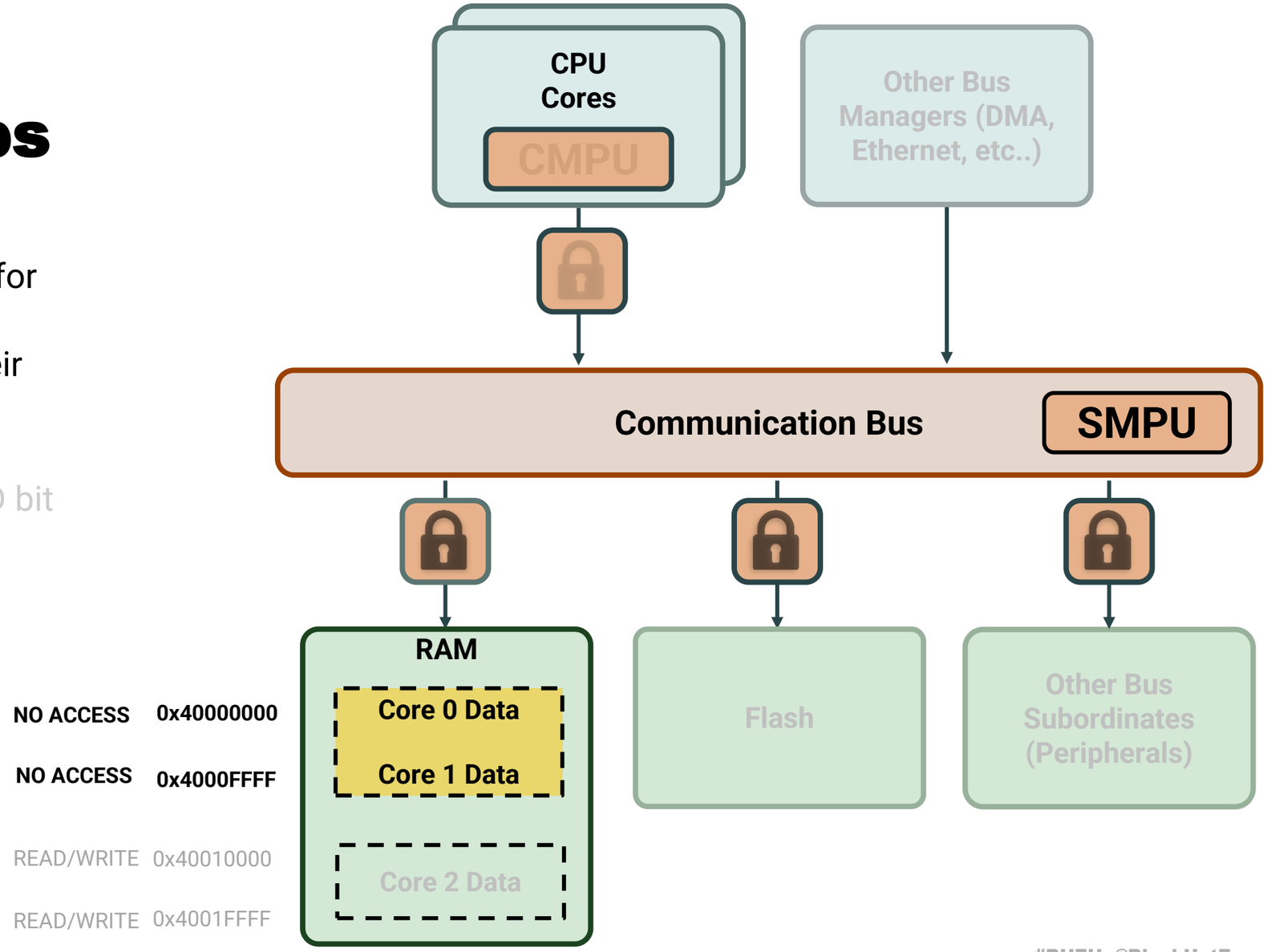


Configuration Steps

1. Define the region descriptors:
   - Set start address & end address for each region
   - List allowed bus masters and their access attributes
2. Lock the RGDs using RO field
3. Enable the SMPU by setting the GVLD bit
4. SMPU is now enabled and locked

SMPU Region 1
[0x40000000-0x4000FFFF]
Bus masters: CPU Core 2 – No access

SMPU Region 2
[0x40010000-0x4001FFFF]
Bus masters: CPU Core 2 – R/W

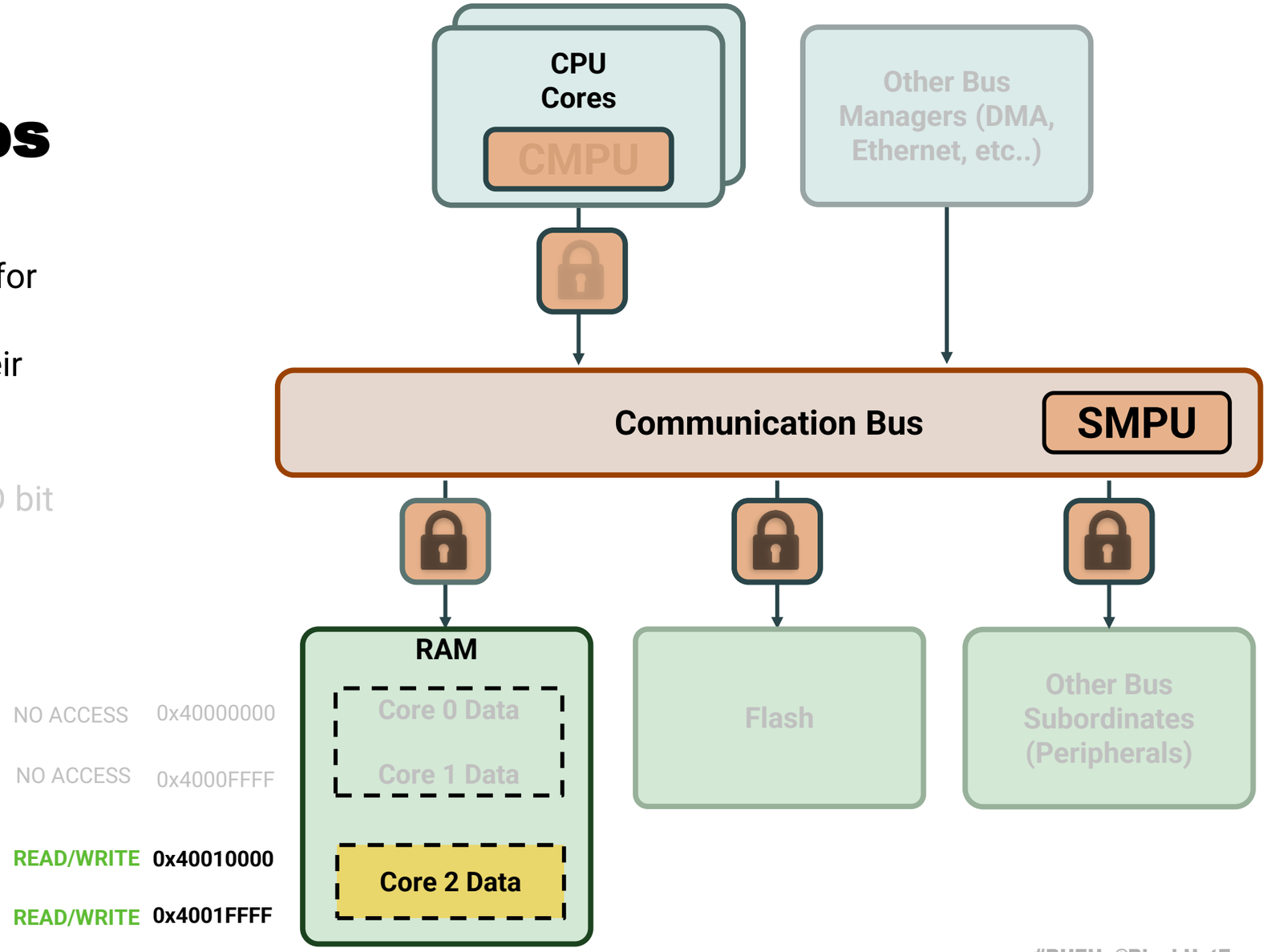


Configuration Steps

1. Define the region descriptors:
   - Set start address & end address for each region
   - List allowed bus masters and their access attributes
2. Lock the RGDs using RO field
3. Enable the SMPU by setting the GVLD bit
4. SMPU is now enabled and locked

SMPU Region 1
[0x40000000-0x4000FFFF]
Bus masters: CPU Core 2 – No access

SMPU Region 2
[0x40010000-0x4001FFFF]
Bus masters: CPU Core 2 – R/W



The Control/Error Status Register 0

Table 277. CESR0 field descriptions

Field	Description
0:15 MERR	Master *n* error, where the bus master number matches the bit number Indicates a captured error in EAR*n* and EDR*n*. A bit is set when the hardware detects an error and records the faulting address and attributes. It is cleared by writing '1' to it. If another error is captured at the exact same cycle as the write, the flag remains set. A find-first-one instruction (or equivalent) can detect the presence of a captured error. 0 No error has occurred for bus master *n*. 1 An error has occurred for bus master *n*.
16	Reserved This read-only bit is reserved and always has the value '1'.
28:30 HRL	Hardware revision level Specifies the SMPU's hardware and definition revision level. It can be read by software to determine the functional definition of the module.
31 GVLD	Global Valid (global enable/disable for the SMPU) 0 SMPU is disabled. All accesses from all bus masters are allowed. 1 SMPU is enabled.

Source: RM0452-spc58-line Rev.4 p. 557

31 GVLD	Global Valid (global enable/disable for the SMPU) 0 SMPU is disabled. All accesses from all bus masters are allowed. 1 SMPU is enabled.

Table 284. RGD*n*_WORD3 field descriptions

Field	Description
27 FMT	Region Descriptor Format This bit selects the configuration format (FMT0 or FMT1) for this region descriptor. **Note:** A specific module instance of the SMPU may support only the FMT0 format. If so, the FMT field is read-only with a fixed value of 0 and only RGD_WORD2_FMT0 applies. 0 Use format 0 (RGD_WORD2_FMT0) 1 Use format 1 (RGD_WORD2_FMT1)
28 RO	Read-Only This bit is intended to prevent accidental writes of an SMPU region descriptor. **Note:** Setting RO in an RGD locks all four words of the RGD until a system reset; the valid bit of the RGD and the global valid bit have no effect. 0 The region descriptor can be read or written. 1 Attempted writes to any location in the region descriptor are ignored with an error-free data transfer termination.

Source: RM0452-spc58-line Rev.4 p. 563

Table 284. RGD*n*_WORD3 field descriptions

Field	Description
27 FMT	Region Descriptor Format This bit selects the configuration format (FMT0 or FMT1) for this region descriptor. **Note:** A specific module instance of the SMPU may support only the FMT0 format. If so, the FMT field is read-only with a fixed value of 0 and only RGD_WORD2_FMT0 applies. 0 Use format 0 (RGD_WORD2_FMT0)
28 RO	**Note:** Setting RO in an RGD locks all four words of the RGD until a system reset; the valid bit of the RGD and the global valid bit have no effect. 0 The region descriptor can be read or written. 1 Attempted writes to any location in the region descriptor are ignored with an error-free data transfer termination.

Note: Setting RO in an RGD locks all four words of the RGD until a system reset; the valid bit of the RGD and the global valid bit have no effect.

Source: RM0452-spc58-line Rev.4 p. 563

Table 284. RGD*n*_WORD3 field descriptions

Field	Description
27 FMT	Region Descriptor Format This bit selects the configuration format (FMT0 or FMT1) for this region descriptor. **Note:** A specific module instance of the SMPU may support only the FMT0 format. If so, the FMT field is read-only with a fixed value of 0 and only RGD_WORD2_FMT0 applies. 0 Use format 0 (RGD_WORD2_FMT0)
28 RO	**Note:** Setting RO in an RGD locks all four words of the RGD until a system reset; the valid bit of the RGD and the global valid bit have no effect. 0 The region descriptor can be read or written. 1 Attempted writes to any location in the region descriptor are ignored with an error-free data transfer termination.

Note: Setting RO in an RGD locks all four words of the RGD until a system reset; the valid bit of the RGD and the global valid bit have no effect.

Source: RM0452-spc58-line Rev.4 p. 563

Table 284. RGD*n*_WORD3 field descriptions

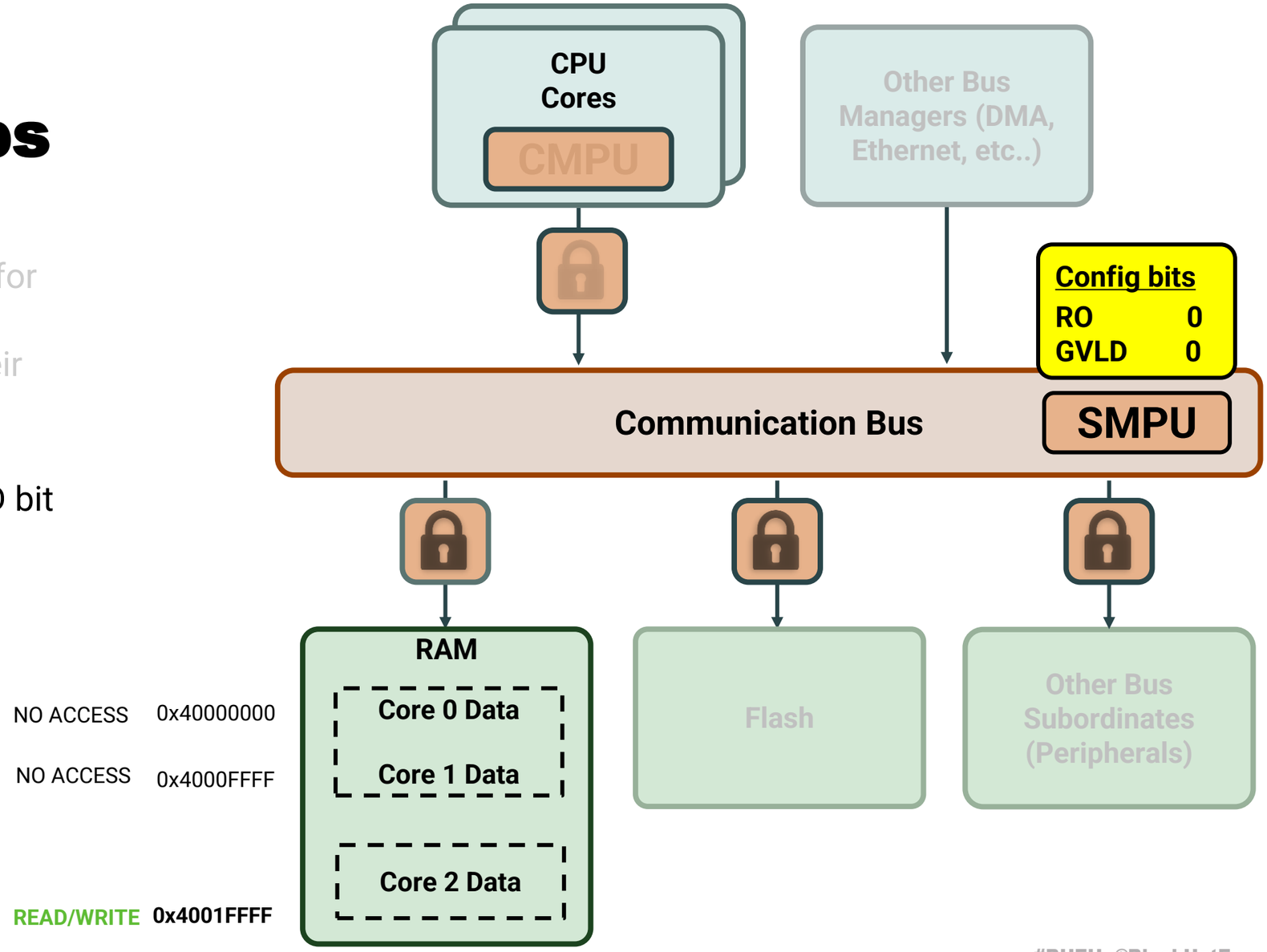
Field	Description
27 FMT	Region Descriptor Format This bit selects the configuration format (FMT0 or FMT1) for this region descriptor. **Note:** A specific module instance of the SMPU may support only the FMT0 format. If so, the FMT field is read-only with a fixed value of 0 and only RGD_WORD2_FMT0 applies. 0 Use format 0 (RGD_WORD2_FMT0)
28 RO	**Note:** Setting RO in an RGD locks all four words of the RGD until a system reset; the valid bit of the RGD and the global valid bit have no effect. 0 The region descriptor can be read or written. 1 Attempted writes to any location in the region descriptor are ignored with an error-free data transfer termination.

Note: Setting RO in an RGD locks all four words of the RGD until a system reset; the valid bit of the RGD and the global valid bit have no effect.

Source: RM0452-spc58-line Rev.4 p. 563

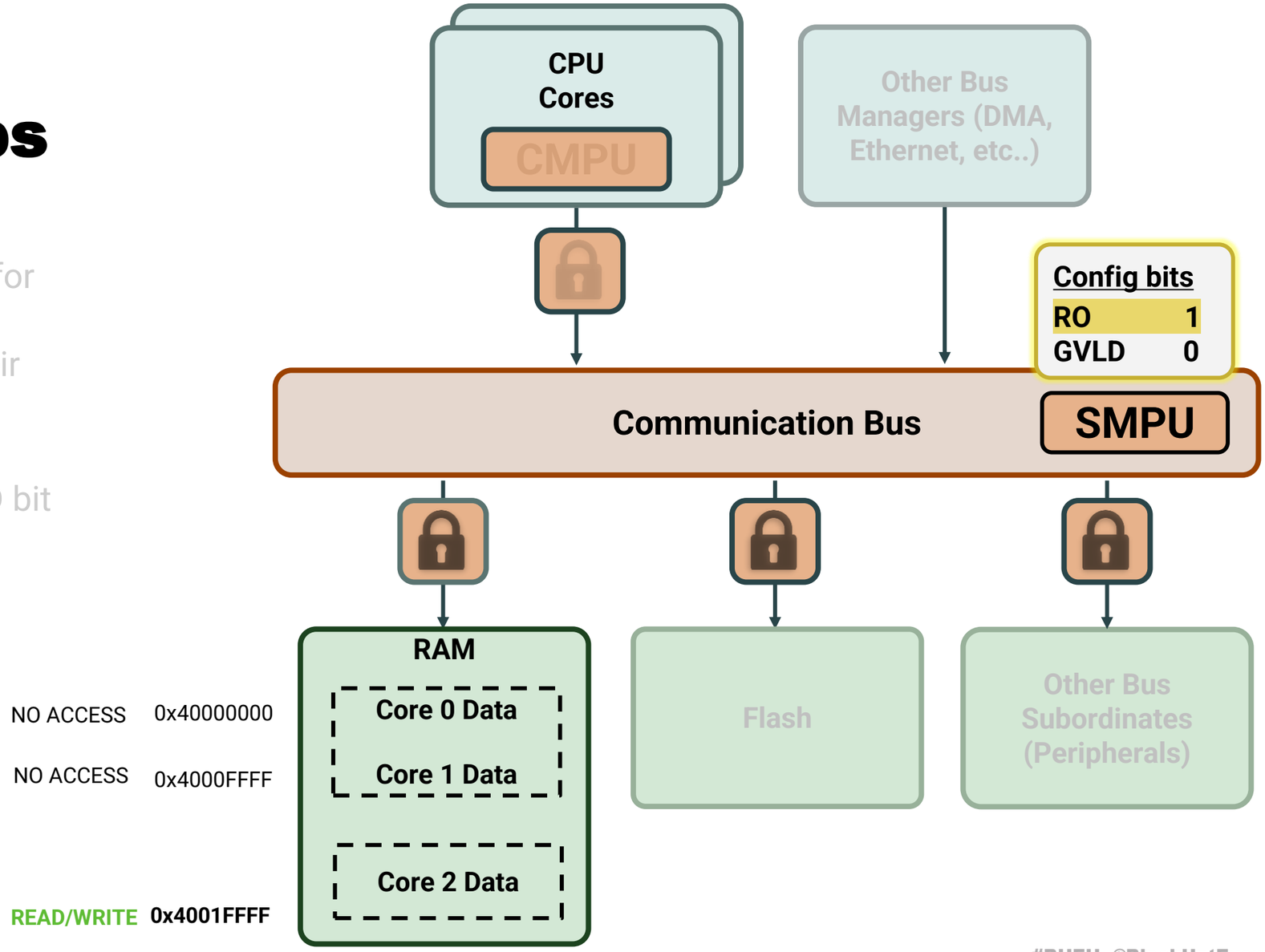
Configuration Steps

1. Define the region descriptors:
   - Set start address & end address for each region
   - List allowed bus masters and their access attributes
2. Lock the RGDs using RO field
3. Enable the SMPU by setting the GVLD bit
4. SMPU is now enabled and locked



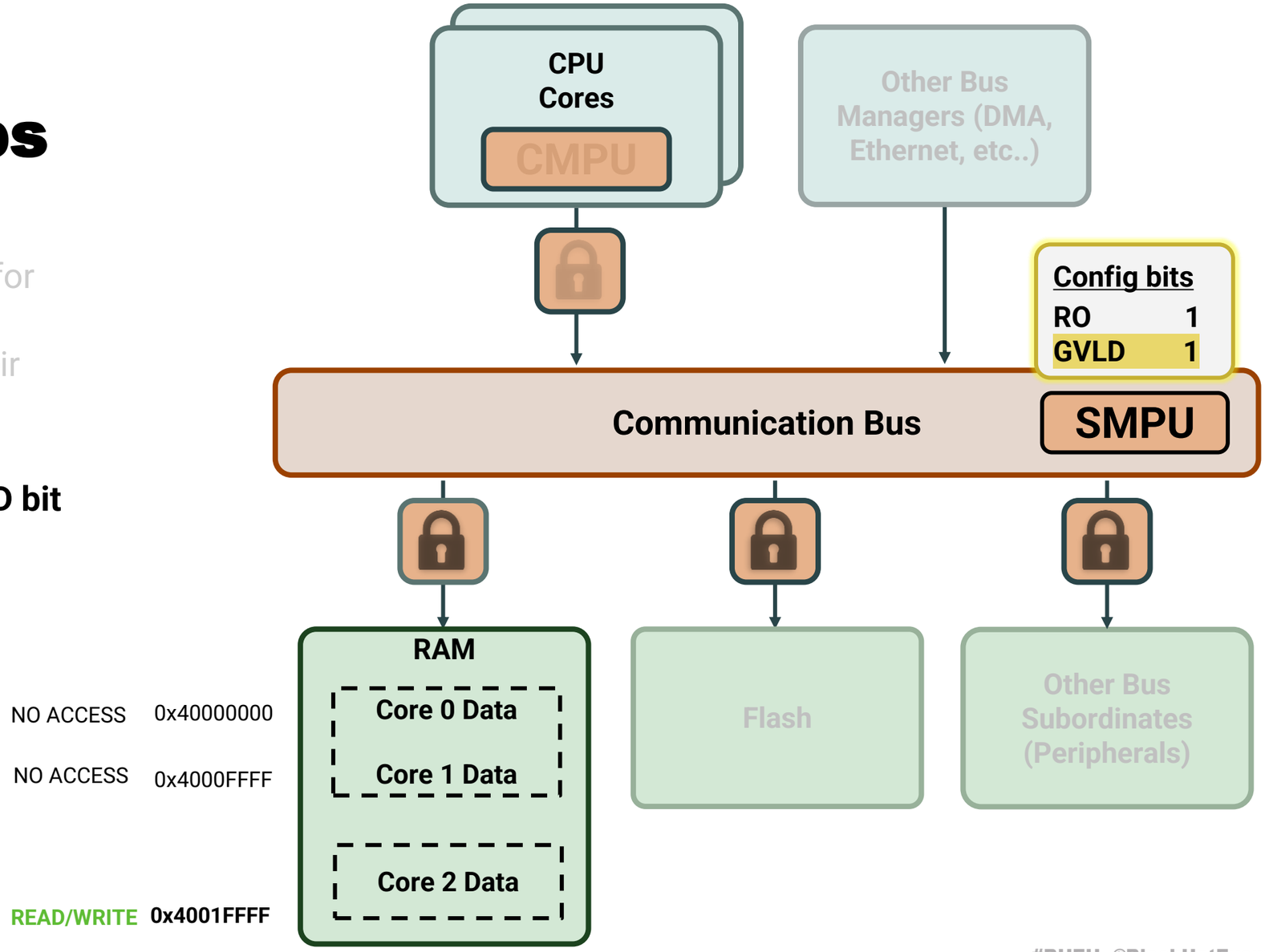
Configuration Steps

1. Define the region descriptors:
   - Set start address & end address for each region
   - List allowed bus masters and their access attributes
2. **Lock the RGDs using RO field**
3. Enable the SMPU by setting the GVLD bit
4. SMPU is now enabled and locked



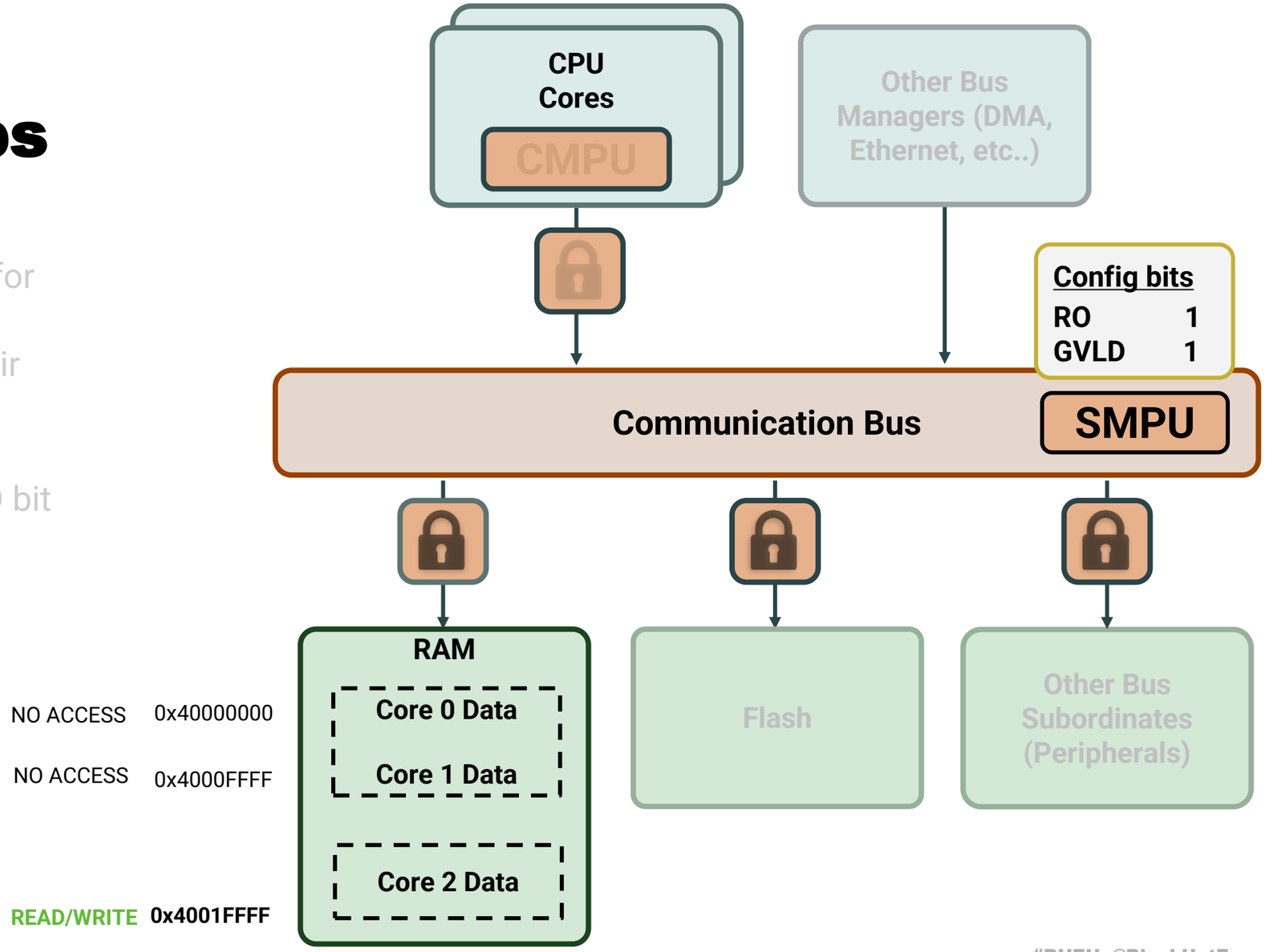
Configuration Steps

1. Define the region descriptors:
   - Set start address & end address for each region
   - List allowed bus masters and their access attributes
2. Lock the RGDs using RO field
3. **Enable the SMPU by setting the GVLD bit**
4. SMPU is now enabled and locked

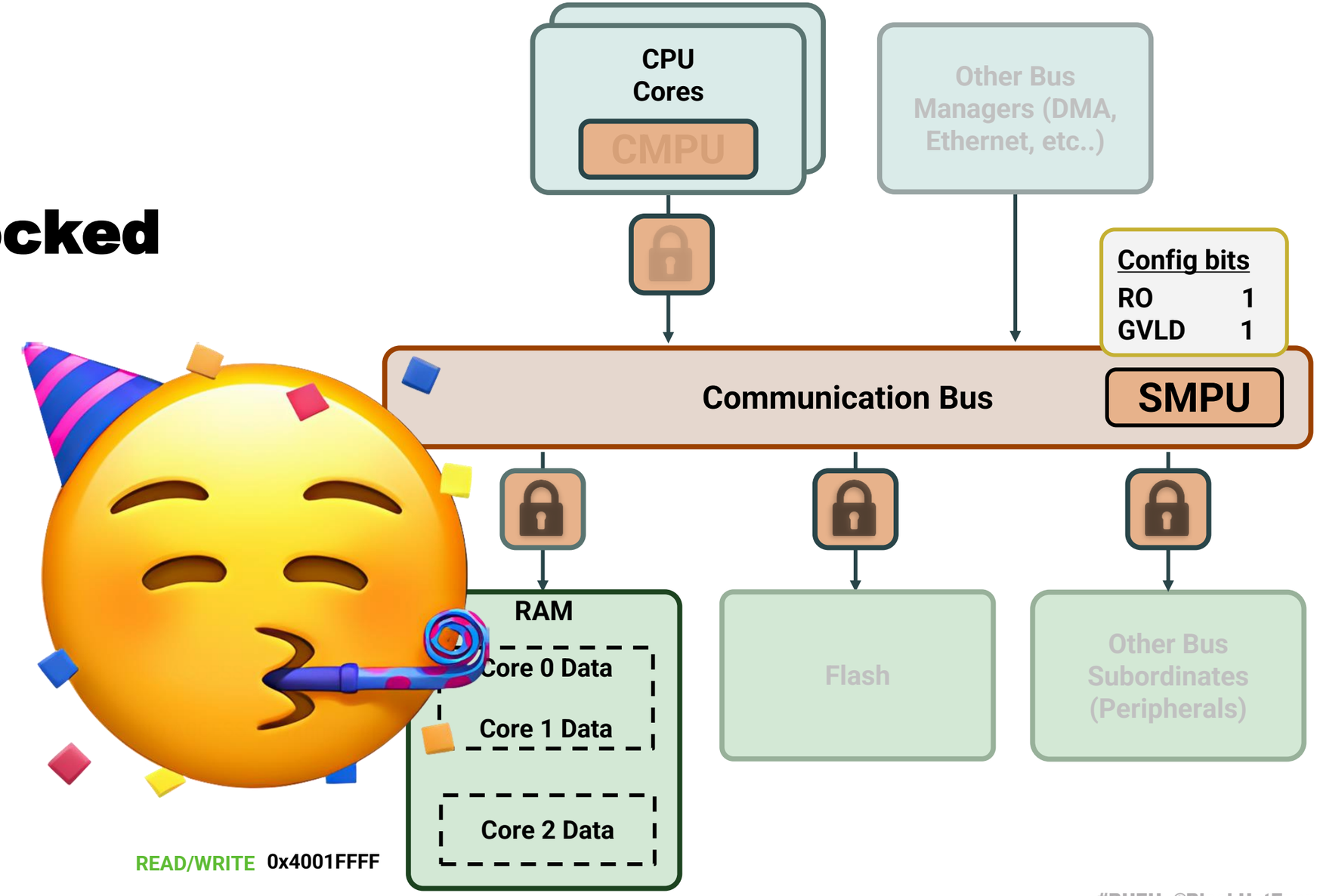


Configuration Steps

1. Define the region descriptors:
   - Set start address & end address for each region
   - List allowed bus masters and their access attributes
2. Lock the RGDs using RO field
3. Enable the SMPU by setting the GVLD bit
4. **SMPU is now enabled and locked**

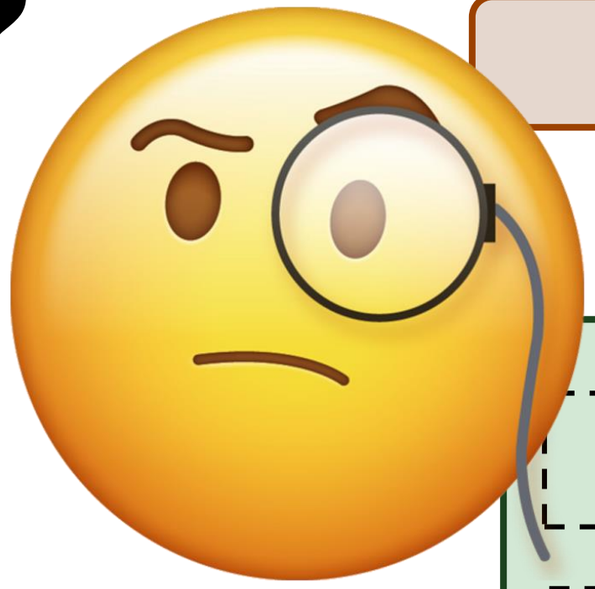


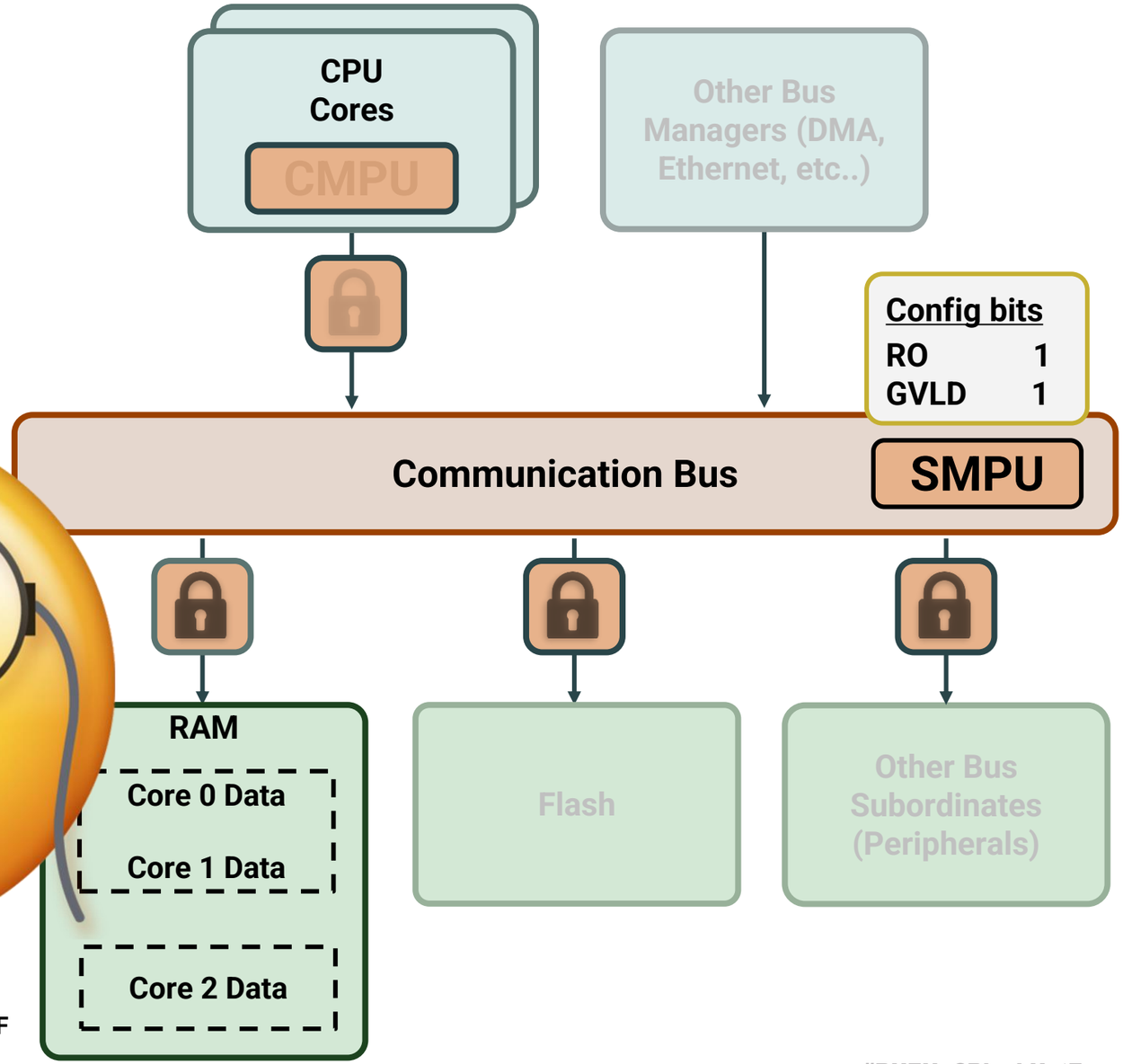
SMPU is now enabled and locked



SMPU is now enabled and locked

Or is it?





SMPU is now enabled and locked

Or is it?

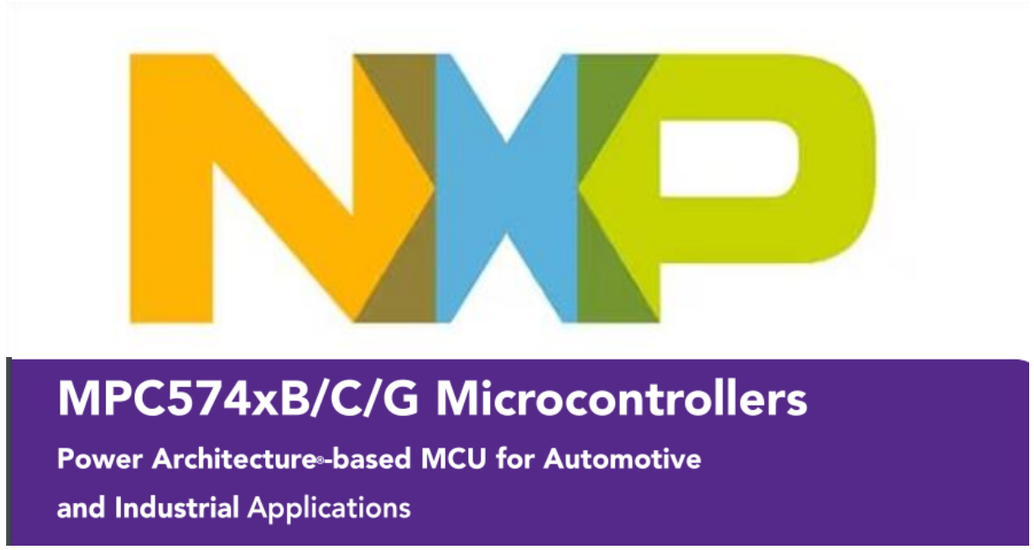
CVE-2023-48010

After the SMPU is **configured**, **enabled** and regions are **locked**, a privileged attacker can flip the GVLD bit, **disabling** the SMPU

Providing **read and write** access to protected areas

Seeking for similar targets

- NXP MPC5xx PowerPC family
- For our test we chose the MPC5748
- Automotive and Industrial Control MCU
- Does NXP MPC5748 share the SMPU issue?



Source: https://www.nxp.com/docs/en/fact-sheet/MPC5748GFS.pdf

NXP MPC5748 PowerPC

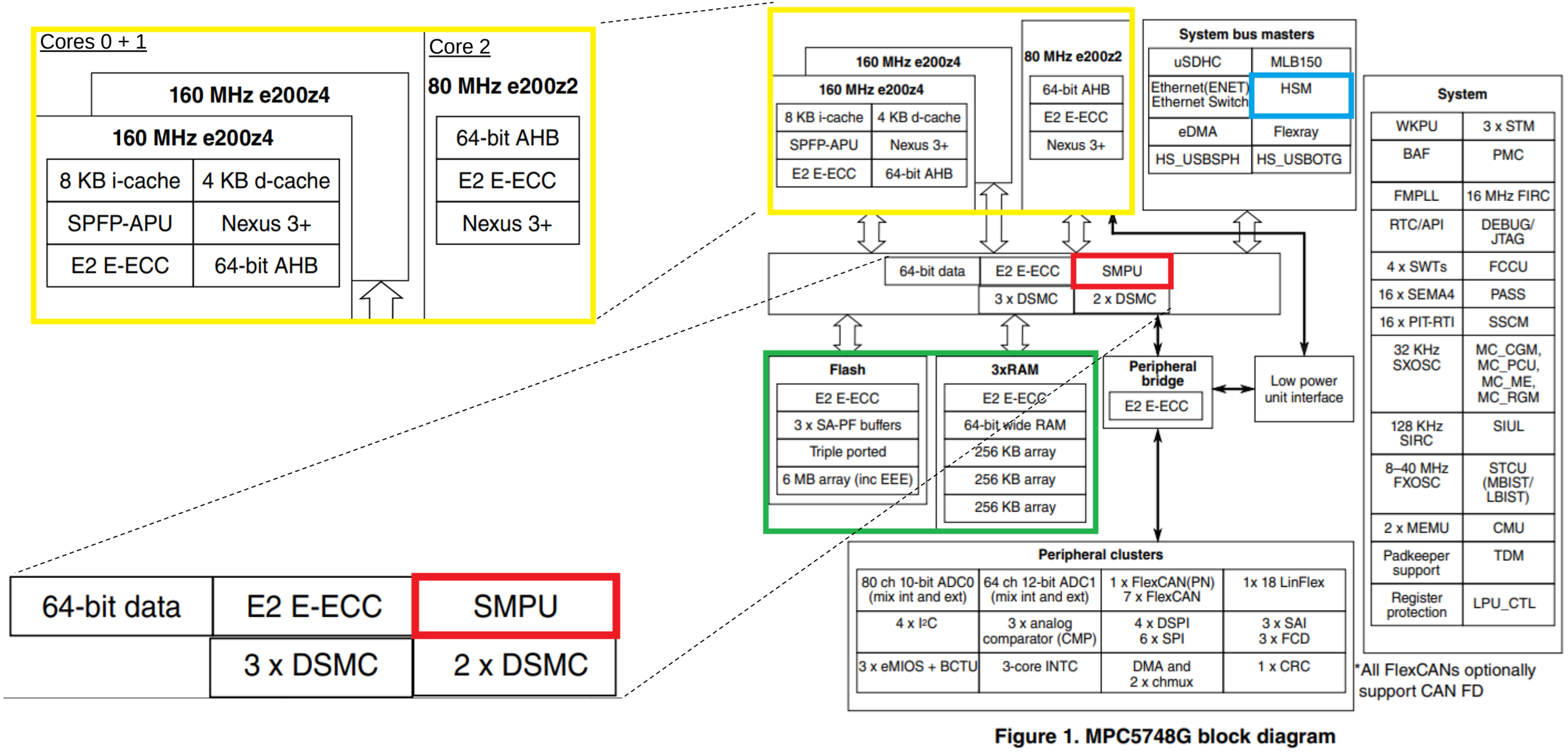


Figure 1. MPC5748G block diagram

Source: MPC5748G Microcontroller Data Sheet, Rev. 6, 11/2018, p. 4

NXP SMPU Global Valid Flag

Memory map/register definition

SMPUx_CES0 field descriptions (continued)

Field	Description
	same cycle as the write, the flag remains set. A find-first-one instruction (or equivalent) can detect the presence of a captured error. 0 No error has occurred for bus master n. 1 An error has occurred for bus master n.
16–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–30 HRL	Hardware revision level Specifies the SMPU's hardware and definition revision level. It can be read by software to determine the functional definition of the module.
31 GVLD	Global Valid (global enable/disable for the SMPU) 0 SMPU is disabled. All accesses from all bus masters are allowed. 1 SMPU is enabled.

Source: MPC5748G Reference Manual, Rev. 7.1 p. 493-494

NXP SMPU Lock Bit and note

SMPUx_RGD*n*_WRD5 field descriptions

Field	Description
0–7 MID	Master ID of RGDn owner If this RGDn is locked through the setting of SMPUx_RGDn_WORD5[LCK], the MID field will reflect the Master ID of the current RGDn owner.
8–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–29 LCK	Region Descriptor Lock This bit allows a master to Lock a region descriptor. Once locked, only that master can modify the Region descriptor's contents. This includes all words of the RGD. **NOTE:** Setting LCK to 1x in an RGD locks all words of the RGD until system reset, the Valid bit of the RGD and the Global Valid bit of the SMPU have no effect. 00 Unlocked 01 Locked by the master wrote this register and LCK bit. Attempted writes by other masters are ignored with an error-free data transfer termination 10 This value is reserved for future enhancements and should not be used. 11 Attempted writes to any location in the region descriptor are ignored with an error-free data transfer termination.
30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
31 VLD	Valid Signals the region descriptor is valid. Any write to RGDn_WORD0–4 clears this bit. 0 Region descriptor is invalid 1 Region descriptor is valid

Source: MPC5748G Reference Manual, Rev. 7.1, p. 505

NXP SMPU Lock Bit and note

SMPUx_RGDn_WRD5 field descriptions

Field	Description
0–7 MID	Master ID of RGDn owner If this RGDn is locked through the setting of SMPUx_RGDn_WORD5[LCK], the MID field will reflect the Master ID of the current RGDn owner.
8–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–29 LCK	Region Descriptor Lock 00 Unlocked 01 Locked by the master wrote this register and LCK bit. Attempted writes by other masters are ignored with an error-free data transfer termination 10 This value is reserved for future enhancements and should not be used. 11 Attempted writes to any location in the region descriptor are ignored with an error-free data transfer termination.
30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
31 VLD	Valid Signals the region descriptor is valid. Any write to RGDn_WORD0–4 clears this bit. 0 Region descriptor is invalid 1 Region descriptor is valid

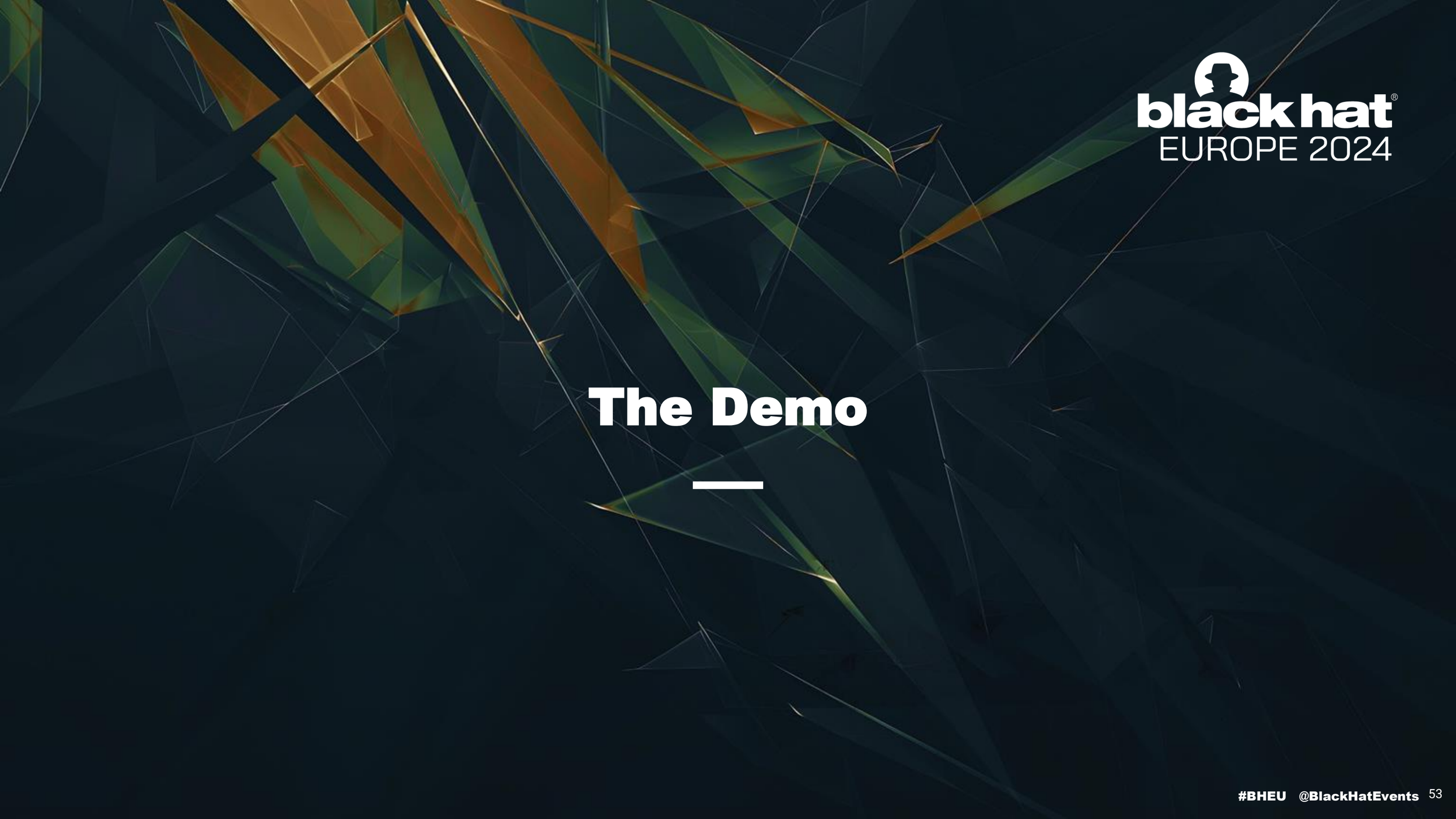
NOTE: Setting LCK to 1x in an RGD locks all words of the RGD until system reset, the Valid bit of the RGD and the Global Valid bit of the SMPU have no effect.

Source: MPC5748G Reference Manual, Rev. 7.1, p. 505

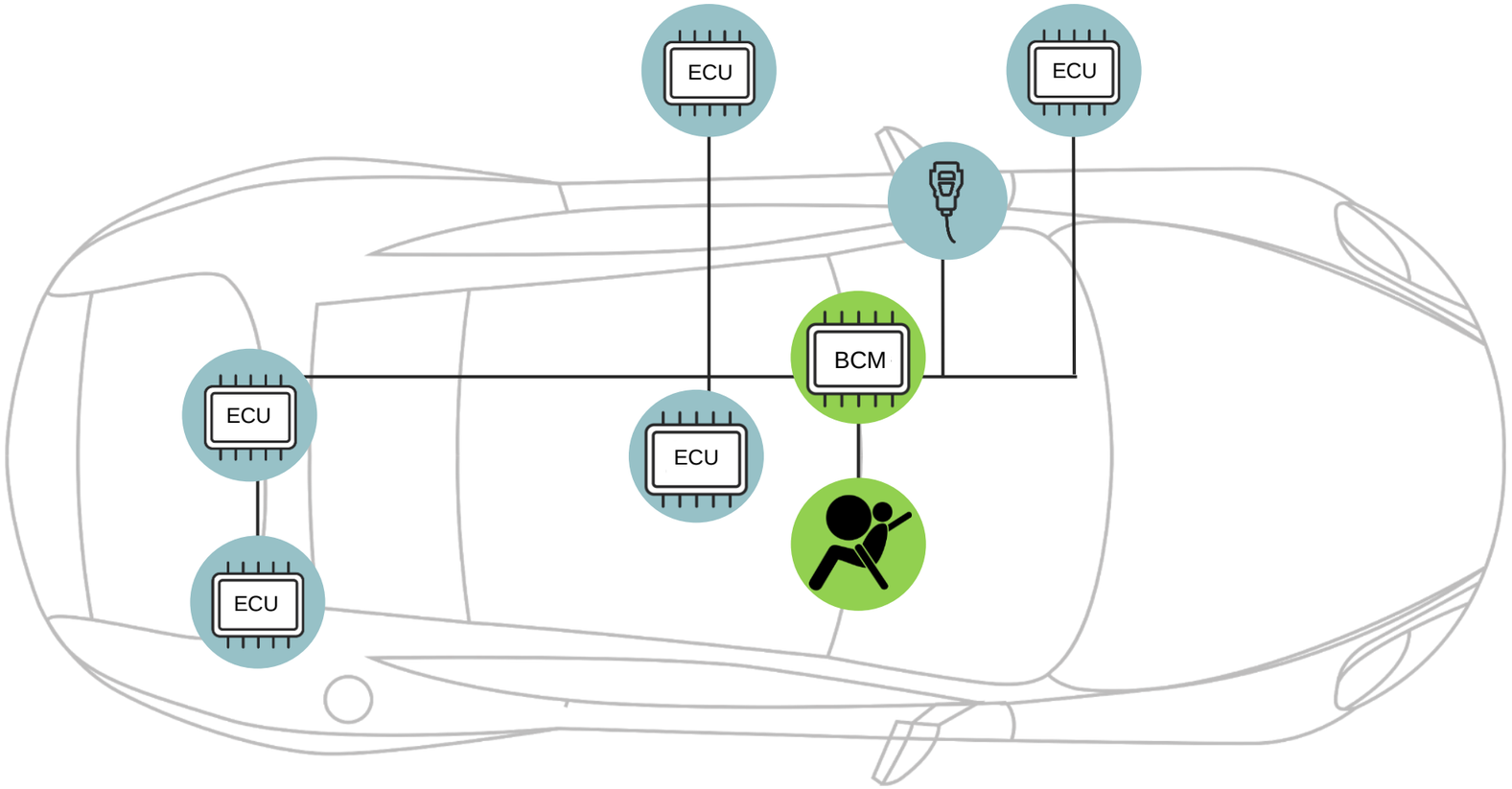
CVE-2024-33882

After the SMPU is **configured**, **enabled** and regions are **locked**, a privileged attacker can flip the GVLD bit, **disabling** the SMPU

Providing read, write and **execute** access to protected areas


black hat
EUROPE 2024
The Demo

The Demo Setup

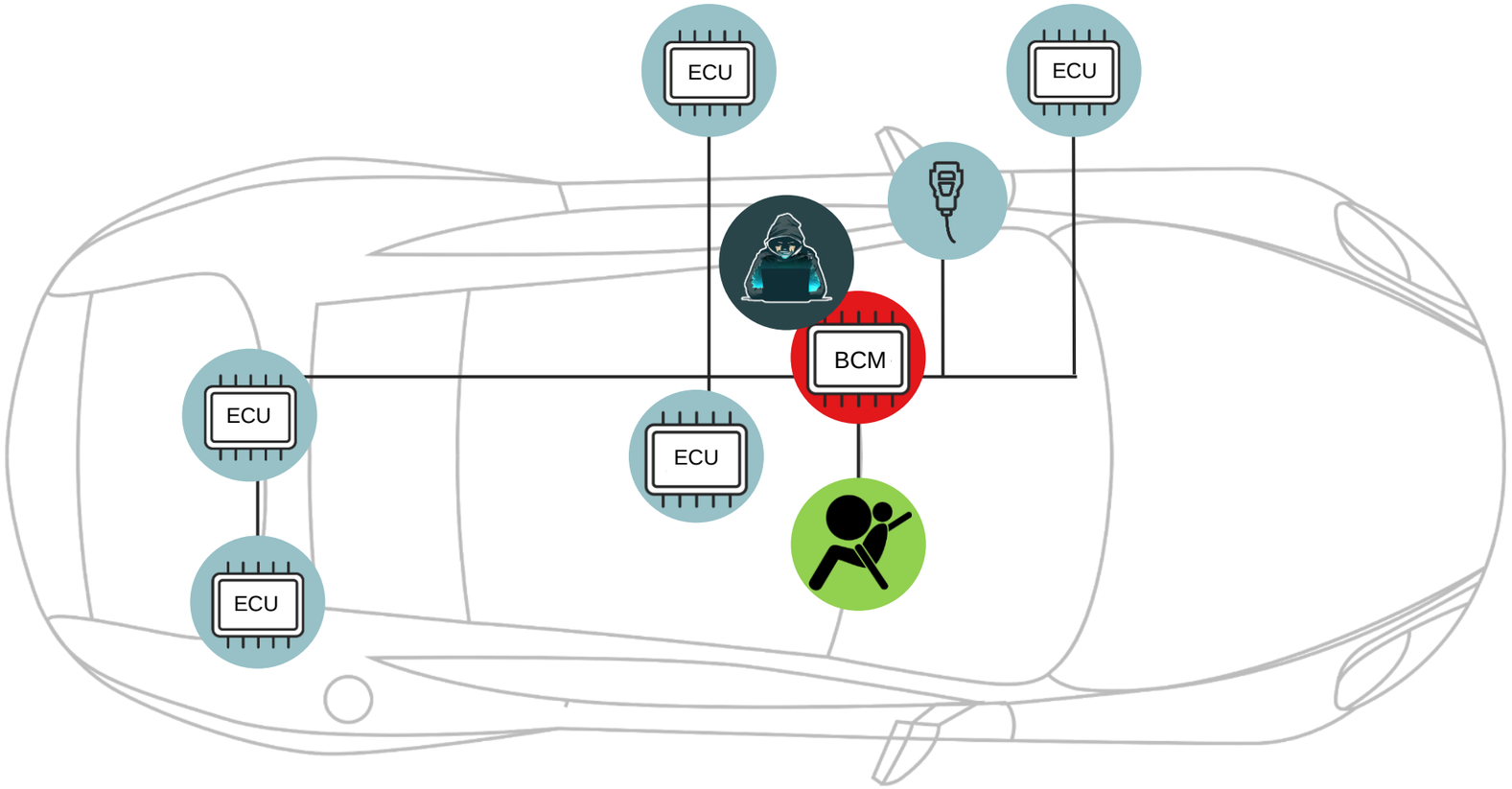


- A demo BCM is connected to the Airbag deployment mechanism and in charge of activating it

ECU – Electronic Control Unit

BCM – Body Control Module

The Demo Setup

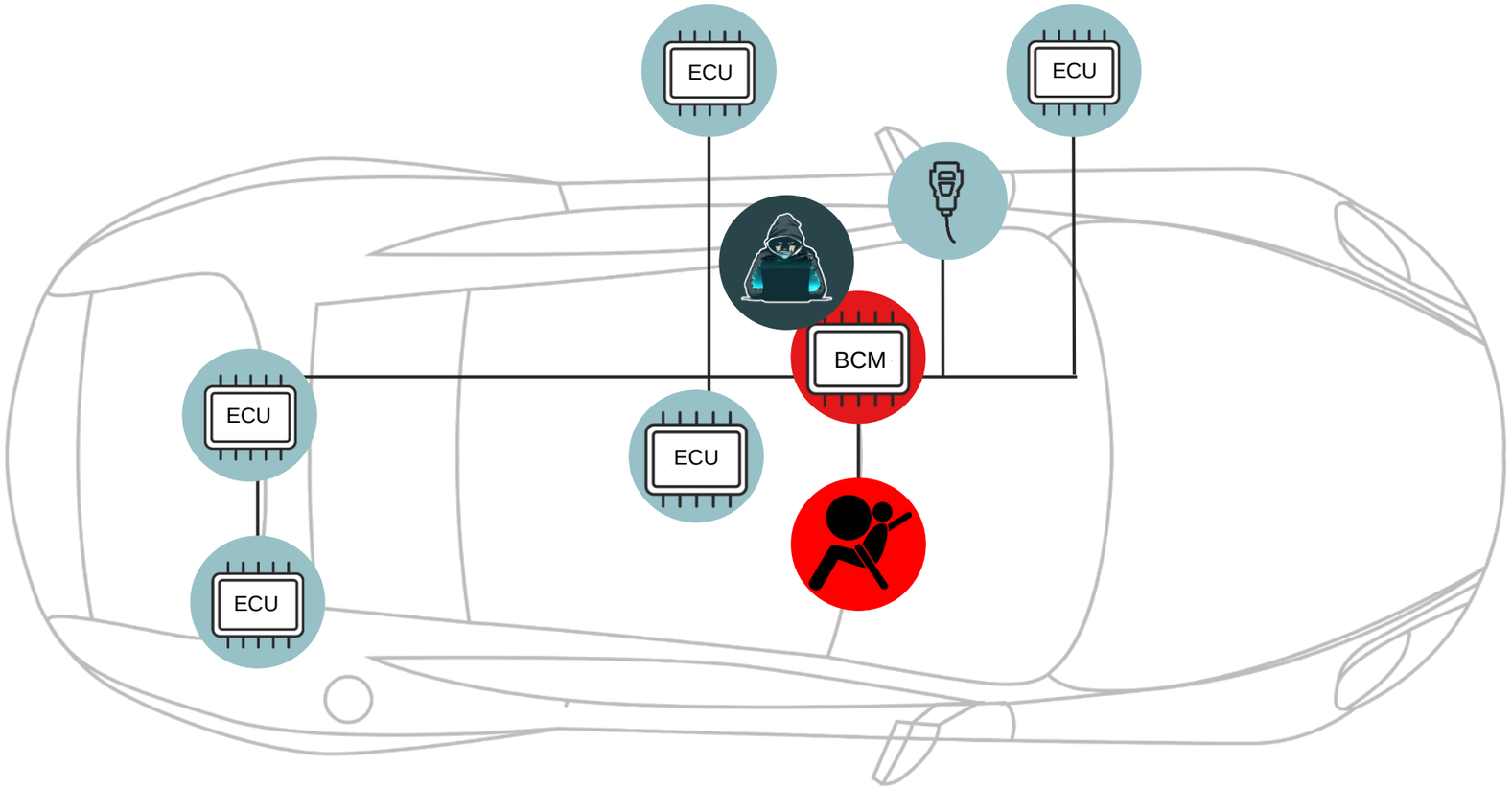


- Attacker is exploiting a stack buffer overflow on the BCM
- The Airbag mechanism is protected by the BCM SMPU
- Stack is non-executable, so only ROP is available

ECU – Electronic Control Unit

BCM – Body Control Module

The Demo Setup

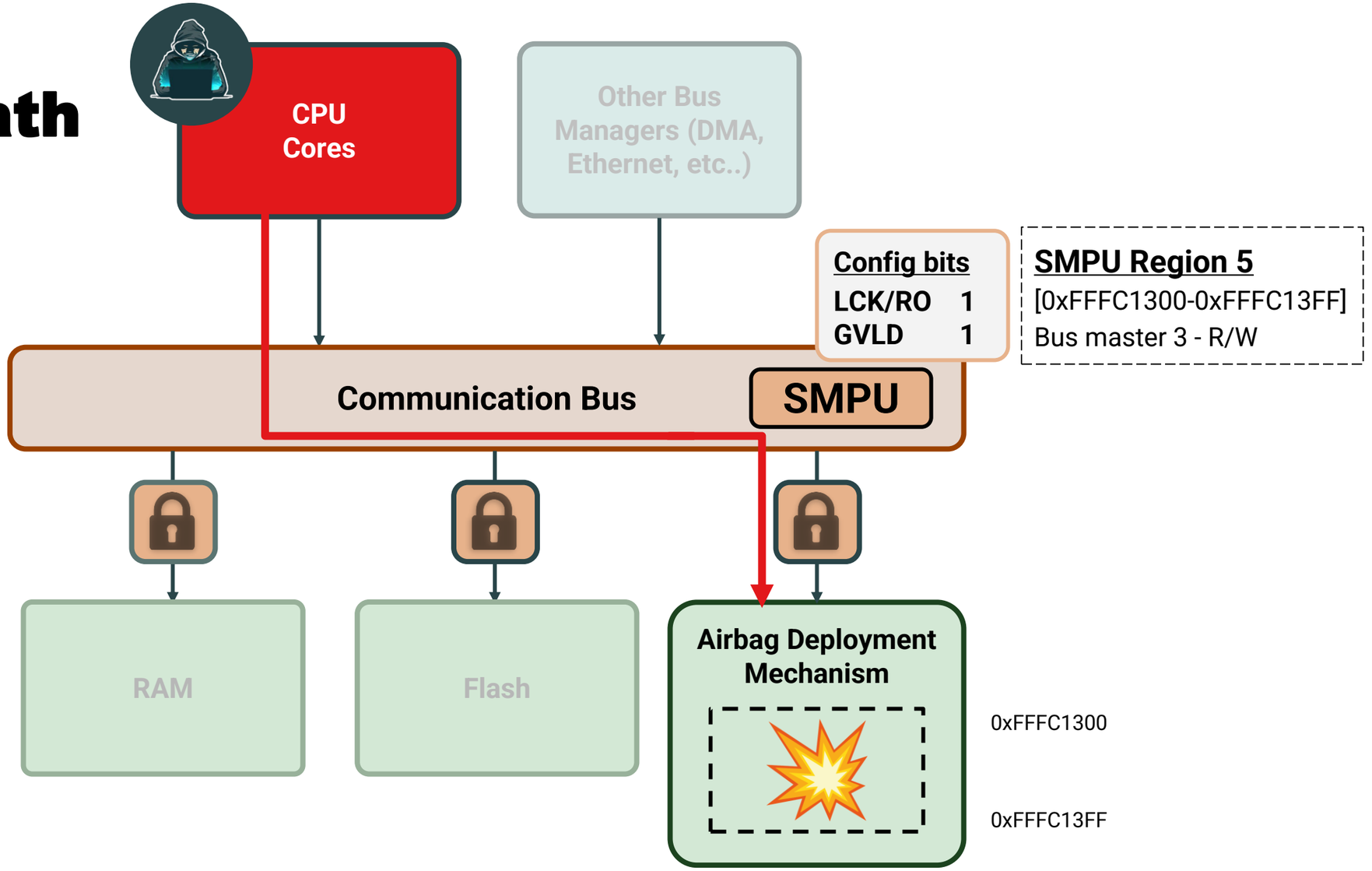


- Attacker's goal is to reach the airbag mechanism's memory and explode the airbags

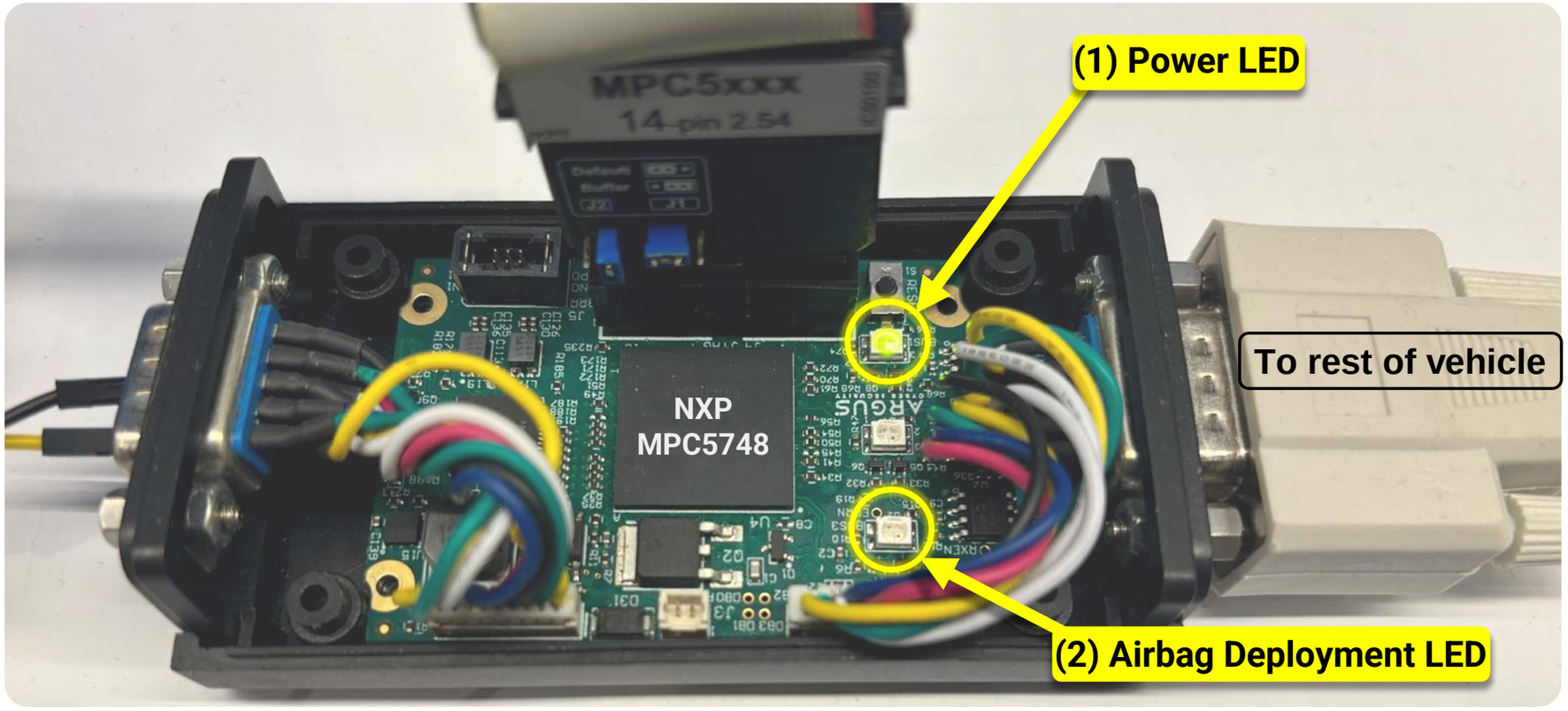
ECU – Electronic Control Unit

BCM – Body Control Module

Demo Attack Path



The Demo BCM


(1) Power LED
To rest of vehicle
(2) Airbag Deployment LED
NXP
MPC5748
MPC5xxx
14-pin 2.54

DEMO 1: SMPU is ON

Attacker has stack buffer overflow over the target BCM

Goal is to trigger Airbag Deployment

X

DEMO 2: Exploit SMPU vulnerability

(1) Attacker disables SMPU by flipping GVLD

(2) Attacker attempts to trigger the Airbags

X

The Responsible Disclosures

Responsible Disclosure with ST

- We contacted ST PSIRT with all information
- ST acknowledged the issue
- **Would release an errata**
- But claimed the SMPU is not a security mechanism

Source: https://www.st.com/content/dam/st-crew/st-logo-blue.svg

STMicroelectronics Response:

> The behavior deviation of SMPU you detected may affect non-secure device domains. However, this domain should not be used for storing security information.
> Secret/security-critical data shall be stored within the HSM sub-system memory, if stored outside, **they need to be encrypted**.
>
> **The SMPU is not a security protection mechanism:** rather, for example, it helps to avoid interference.

[Emphasis added]

Responsible Disclosure with NXP

- We contacted NXP PSIRT with all information
- NXP acknowledged the issue
- **Would release a Documentation Errata**
- Also claimed the SMPU is not a security feature

Source: https://www.nxp.com/docs/en/fact-sheet/MPC5748GFS.pdf

NXP Response:

> **The product's Reference Manual is clear about the SMPU not being a security feature.** The SMPU is **not mentioned in the chapter 'Security Overview'**, nor in the section 'Security Modules', **but rather in the section on 'System Modules'**. The SMPU is also not listed under 'Security' in the 'Feature List' table and the chapter that describes the SMPU does not mention 'security'.

[Emphasis added]

Mitigations and conclusion

Mitigations and Concluding Remarks

- Memory Protection Units are a crucial part of every microcontroller's defense
- MPUs are indispensable in the context of Automotive apps
- Fixing hardware issue is hard!

Mitigations:

- Test important claims made in datasheet
- Use exploit mitigations, such as stack smashing defenses
- In the case of the ST chip, block as much as possible using the CMPUs

Thank you!

PLAXIDITYX

GO EVERYWHERE