SR5 E1 line–How to use the memory error management unit 2 (MEMU2)

Introduction

This document describes the use of memory error management unit 2 (MEMU2) embedded in the SR5 E1 line devices.

These devices target automotive applications and offer a cost-effective solution. The MEMU2 module stores the details of the ECC (error correction code) errors detected in the memories of the device and reports the event.

Note: It is called MEMU2 because the version embedded in the SR5 E1 line devices implements features that are not present in the previous version of the module called MEMU embedded in the previous generation of automotive microcontrollers.

Refer to the reference manual [1] for further details on each module (see Reference documents). This application note applies to the devices listed in the following table:

Table 1. Device summary

Series	Part number
SR5E1x	SR5E1E3, SR5E1E5, SR5E1E7

1 Memory error management unit 2 (MEMU2)

1.1 Overview

The memory error management unit 2 (MEMU2), as part of the safety architecture of SR5 E1 line devices, is responsible for collecting and reporting error events associated with faults detected by MBISTs (memory built-in self-test) and ECC (error correction code) logic.

The input errors reported to the MEMU2 are forwarded to the fault collection and control unit (FCCU), either directly or via the collective error manager (CEM), using a filtering mechanism.

This document provides examples of how to use the MEMU2 and highlights its dependencies on other modules present in the SR5 E1 line device.

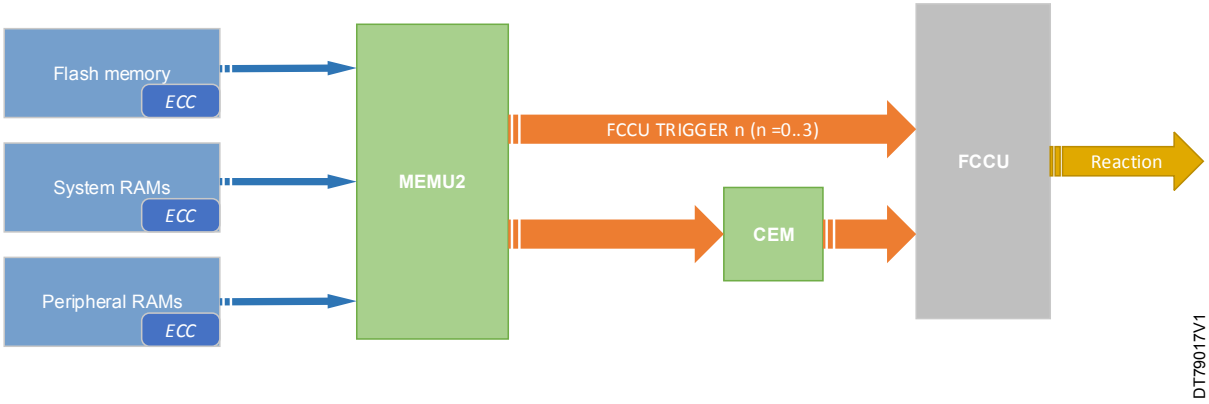
1.2 MEMU2 system connections

The input signals to the MEMU2 are divided into three categories, which gather the ECC errors happening upon the access to each of the respective memory types:

1. System-accessible RAM
2. Peripheral local RAM
3. Nonvolatile memory (NVM)

The MEMU2 records and reports correctable and uncorrectable errors.

Figure 1. MEMU2 connections



After MEMU2 collects the input faults, the enabled ones are forwarded directly to the FCCU or via the CEM. The FCCU receives a set of input triggers, each indicating a certain type of fault, and it is user-programmable in terms of system reaction.

The CEM module gathers multiple errors and sends an aggregated fault trigger to the FCCU, reducing the number of individual fault triggers received by the FCCU.

1.3 MEMU2 module features

The MEMU2 has the following features:

1. Support for ECC error reporting.
2. Logging of unique errors into MEMU2's reporting tables, accessible by the CPU via a memory-mapped register interface, store of errors in correctable/non-correctable sections if not already logged and error indication to the FCCU.
3. Overflow handling during error assertion and report status accordingly.
4. Allow the MEMU2's reporting tables to be updated via the register interface, enabling software to add or remove error entries.
5. Region filtering for ECC errors, allowing error flags to be routed to different FCCU channels depending on the address of the error and the type of reaction required.

1.4 MEMU2 error sources and reporting table

Faults detected by MBISTs and ECC logic are reported to MEMU2 and stored in the corresponding reporting table based on the type of memory where it was found and the type of error.

Table 2 shows the actual number of input sources for each error type.

Table 2. MEMU2 unique input sources

Instance	Value
Number of system RAM unique error sources	22
Number of peripheral RAM unique error sources	9
Number of flash memory unique error sources	1

The MEMU2 records and reports the following categories of errors in a specific table:

1. Correctable error:
   a. Single bit errors are detected for system RAM, peripheral, and local RAM access.
   b. Single and double bit errors detected for NVM access (see Note).
2. Uncorrectable error:
   a. Double bit errors detected for system RAM, peripheral, and local RAM access.
   b. Triple bit errors for NVM access (see Note).

Note:

1. *The ECC function on SR5 E1 line devices provides:*
   – *SEC/DED (single bit error correction and double bit error detection) for core, local memory, peripheral RAM.*
   – *SEC/DED (single bit error correction, double bit error detection) for CODE flash memory.*
   – *DEC/TED (double bit error correction, triple bit error detection) for DATA flash memory.*
2. *Data NVM ECC error forwarding is not implemented on SR5 E1 line devices, so only code NVM errors are reported to MEMU2.*

Each entry in a reporting table corresponds to a unique error event. The number of entries supported by each table depends on the error source and it is device-dependent. Table 3 gathers the details about the SR5 E1 line devices implementation.

Table 3. MEMU2 entry details in the ECC reporting tables

Memories MEMU2 instance	Entry details	Number of entries	Register interface(1)	Overflow register
NVM	Single bit correctable error reporting table	32	NVM_OUT_TRIG_CTRL(2) NVM_SB_CERR_STSn NVM_SB_CERR_ADDR*n* NVM_SB_CERR_TBL_FILL_STAT	NVM OFLW [0]
	Double bit correctable error reporting table	1	NVM_OUT_TRIG_CTRL(2) NVM_DB_CERR_STSn NVM_DB_CERR_ADDR*n* NVM_DB_CERR_TBL_FILL_STAT	
	Uncorrectable error reporting table	1	NVM_OUT_TRIG_CTRL(2) NVM_UNCERR_ADDR*n* NVM_UNCERR_STS*n* NVM_UNCERR_TBL_FILL_STAT	
System RAM	Single bit correctable error reporting table	10	SYS_RAM_OUT_TRIG_CTRL(2) SYS_RAM_CERR_STS*n*	SYS_RAM_OFLW [0] SYS_RAM_OFLW [1]

Memories MEMU2 instance	Entry details	Number of entries	Register interface[1]	Overflow register
System RAM			SYS_RAM_CERR_ADDR*n* SYS_RAM_CERR_TBL_FILL_STAT	SYS_RAM_OFLW [0] SYS_RAM_OFLW [1]
	Uncorrectable error reporting table	1	SYS_RAM_OUT_TRIG_CTRL[2] SYS_RAM_UNCERR_STSn SYS_RAM_UNCERR_ADDRn SYS_RAM_UNCERR_TBL_FILL_STAT	
Peripheral RAM	Single bit correctable error reporting table	2	PERIPH_RAM_OUT_TRIG_CTRL[2] PERIPH_RAM_CERR_STS*n* PERIPH_RAM_CERR_ADDRn PERIPH_RAM_CERR_TBL_FILL_STAT	PERIPH_RAM_OFLW [0] PERIPH_RAM_OFLW [1]
	Uncorrectable error reporting table	1	PERIPH_RAM_OUT_TRIG_CTRL[2] PERIPH_RAM_UNCERR_STS*n* PERIPH_RAM_UNCERR_ADDRn PERIPH_RAM_UNCERR_TBL_FILL_STAT	

1. *n = 0 ... number of entries -1.*
2. *Check the connections between MEMU2 and FCCU via fake fault injection by software.*

The CTLR (Control register) and ERR_FLAG (error flag register) are common for the three MEMU2 instances.

If a reporting table has been filled and a new error occurs, then a fault trigger to the FCCU is activated, indicating table overflow. The MEMU2 overflow error is reported to the FCCU Channel #33 through CEM_10 instance.

When there is an overflow in the synchronous FIFO, a FIFO_overflow flag is reported to:

- The FCCU channel #34 via CEM_11 instance for system RAM.
- The FCCU channel #35 via CEM_12 instance for peripheral RAM.
- The FCCU channel #36 for NVM.

1.5 MEMU2 ECC filtered reaction

The ECC filtering scheme is designed to partition the application memory into different safety levels, ranging from full ASIL-D to QM.

This requires that, upon an ECC failure, the reaction is handled differently depending on:

- Where the failure occurred: identifying the specific memory region.
- Type of failure: determining whether it is a single bit error or an uncorrectable bit error

1.5.1 ECC fault descriptor

The ECC filtering mechanism uses an architecture based on fault descriptors.

The user software programs the fault-descriptor registers in the MEMU2 register interface, so to identify a certain memory region and the fault type (correctable single bit, or uncorrectable multibits).

Each ECC fault descriptor holds three 32-bit registers:

- Start address
- End address
- Control

Table 4. Descriptor control register

Bit	Description
EN	Used to enable the ECC fault descriptor. If not enabled, it is simply ignored during sequential access to the various ECC fault descriptors.
T32	A 33rd address bit, which is appended as the MSB bit to the 32-bit address reported to the MEMU2. The usage of this bit is to minimize the number of ECC fault descriptors needed for memories mapped into nonconsecutive addresses.
SBC	Single bit correctable fault: Determines whether single bit correctable ECC errors must be considered (if set to 1) or ignored (if set to 0).
DBC	Double bit correctable fault (not applicable to SR5E1 line devices): Determines whether double bit correctable ECC errors must be considered (if set to 1) or ignored (if set to 0). This bit is instantiated only for MEMU2 ECC input fault channels connected to the NVM.
UC	Uncorrectable fault: Determines whether uncorrectable ECC errors must be considered (if set to 1) or ignored (if set to 0).
FCCU_TRG	Determines which FCCU trigger number is activated. The programmed value corresponds to the output trigger number driven by the MEMU2 to the FCCU. If this field is set to "all-1", no FCCU trigger is generated. This case is needed to latch ECC errors in some memory regions where no specific treatment is required.

For each of the three instances of MEMU2 (system RAM, peripheral RAM and NVM), there are dedicated descriptors. The following table shows the number of descriptors for each area and the detailed registers.

Table 5. MEMU2 descriptors details

Memories	Descriptors register detail	Number of descriptors/registers
NVM	NVM_ECC_FD_CTRL NVM_ECC_FD_START NVM_ECC_FD_END	4 (descriptors)
	NVM_RAM_T32 (1)	1 (register)
System Ram	SYS_RAM_ECC_FD_CTRL SYS_RAM_ECC_FD_START SYS_RAM_ECC_FD_END	4 (descriptors)
	SYS_RAM_T32n (1)	1 (register)
Peripheral RAM	PERIPH_RAM_ECC_FD_CTRL PERIPH_RAM_ECC_FD_START PERIPH_RAM_ECC_FD_END	4 (descriptors)
	PERIPH_RAM_T32n (1)	2 (register)

1. *These registers are used to enable a 33rd address bit as the MSB of the address used for ECC fault filtering.*

Note: By default, if the user software does not program the fault-descriptor registers, the first descriptor (n = 0) of each MEMU2 instance is enabled to take all ECC errors (SB, DB, UC) and to use the FCCU trigger number 0. The whole device memory (0x00000000-- 0xFFFFFFFF) is selected as the safety region.

2 MEMU2 module functional description

MEMU2 processes the errors based on whether they are correctable or uncorrectable (from either ECC or MBIST logic), independently of the source generating these errors.

The MEMU2 has multiple instances, see Table 3, one each for:

- System RAM ECC and MBIST
- Peripheral SRAM ECC
- NVM ECC

Each instance has two types reporting tables: correctable errors are reported in one table while the uncorrectable are reported in the other.

The following description applies to each instance. The type of error is determined and stored in the appropriate reporting table (correctable or uncorrectable). Each entry in the reporting table is associated with a "valid" flag, which is set once an error is stored in the table.

If new entry is unique, it is stored in the reporting table; otherwise, it is discarded.

To assess the uniqueness, only the entries in the reporting table with the VALID flag set to 1 and from the same error source (ECC or MBIST) as the incoming error are considered.

The fault comparison is based on the following criteria:

- Fault address (32 bits)
- Syndrome (16 bits for NVM, 8 bits otherwise)
- Fault type (1 bit to signal a correctable error, 1 bit to signal an uncorrectable error). NVM errors provide two separate bits for single and double correctable errors.

In the case of uncorrectable errors, no syndrome or bad-bit information is present, neither are the errors categorized into ECC or MBIST when stored in the reporting table. Therefore, for the uniqueness check of uncorrectable errors, only address comparison is performed. If the address of the incoming error matches the address of a valid error in the uncorrectable reporting table, the error is discarded.

If a table is full and a new error occurs, a fault trigger to the FCCU is activated, indicating table overflow. Each table has its dedicated fault to indicate an overflow. The user software can write known error addresses into the reporting table to prevent those error addresses from being reported to the FCCU if they are accessed later during operation.

The MEMU2 signals a fault to the FCCU if the fault is new and a valid region is found.

2.1 Error reporting for code NVM

This section describes error reporting for the code NVM.

When an ECC error occurs, the error is detected and, if enabled in the NVM platform interface registers, forwarded to the MEMU2.

Note: Data NVM ECC error forwarding is not implemented on SR5 E1 line devices, so only code NVM errors are reported to MEMU2.

Figure 2 shows the error reporting between NVM, MEMU2, CEM, and FCCU.

User software configures which FCCU channel the hardware routes an ECC fault to, as shown in Table 6. If the user does not program the fault descriptors, the fault indication uses the FCCU trigger number 0, and it reaches the FCCU channel #29.

If a FIFO error overflow (FLASH_OVRFLOW_FIFO_0) occurs, the error is captured and directly reported to the FCCU through channel #36.

If a MEMU2 FIFO error overflow occurs, the error is captured in the NVM_OFLWn register of MEMU2 and reported to CEM_10. When enabled, the error is also forwarded to the FCCU channel #33.

An overflow condition, encountered when attempting to make a new “unique” entry in an error reporting table, when all entries of this table are valid, is indicated by asserting the flags of ERR_FLAG register in the MEMU2: F_UCEO, F_SCEO.

Figure 2. **Error reporting for the NVM**

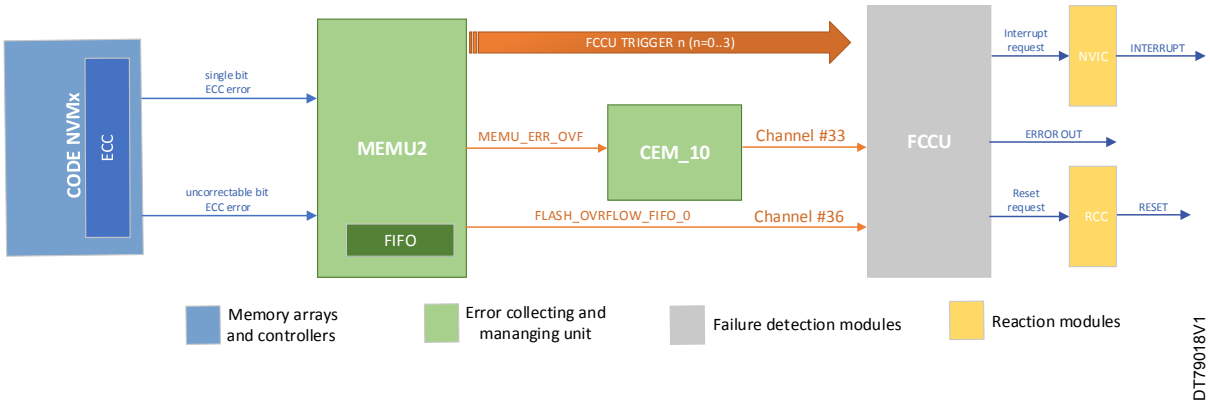


Table 6. **NVM ECC fault descriptor trigger value**

NVM_ECC_FD[n]. CTRL		FCCU channel#
EN	FCCU_TRG	
1, n = 0 0, 1 ≤ n < 4 (default configuration after power-on reset)	0x0: NVM_TRIG_0 is asserted	29
	0x1: NVM_TRIG_1 is asserted	30
	0x2: NVM_TRIG_2 is asserted	31
	0x3: NVM_TRIG_3 is asserted	32
	0xF: no FCCU trigger is asserted	N.A.

2.2 Error reporting for system RAM

This section describes the error reporting for the system RAM.

When an ECC error occurs, the detected error is forwarded to the MEMU2, and, if enabled, forwarded to the FCCU. See Figure 3 for the link between system RAM, MEMU2, CEM, and FCCU.

Users can configure which FCCU channel the hardware routes an ECC fault to, as shown in Table 7.

When user software does not program the fault descriptors, the fault indication uses the FCCU trigger number 0, and it reaches the FCCU channel #21.

If a FIFO error overflow (SYS_RAM_OVRFLOW_FIF0_x) occurs, the error is captured by CEM_11 and reported to the FCCU through channel #34.

If a MEMU FIFO error overflow occurs, the error is captured in the SYS_RAM_OFLW*n* register and reported to CEM_10 and, if enabled, forwarded to FCCU channel #33. When one bit between SR_CEO, or SR_UCEO in the ERR_FLAG register is asserted, it can indicate the occurrence of the following condition: overflow in correctable or uncorrectable reporting tables occurs, depending on the flag set.

Overflow occurs when the relevant table is full, all entries of this table are valid, and a new “unique” entry arrives.

Figure 3. Error reporting for system RAM

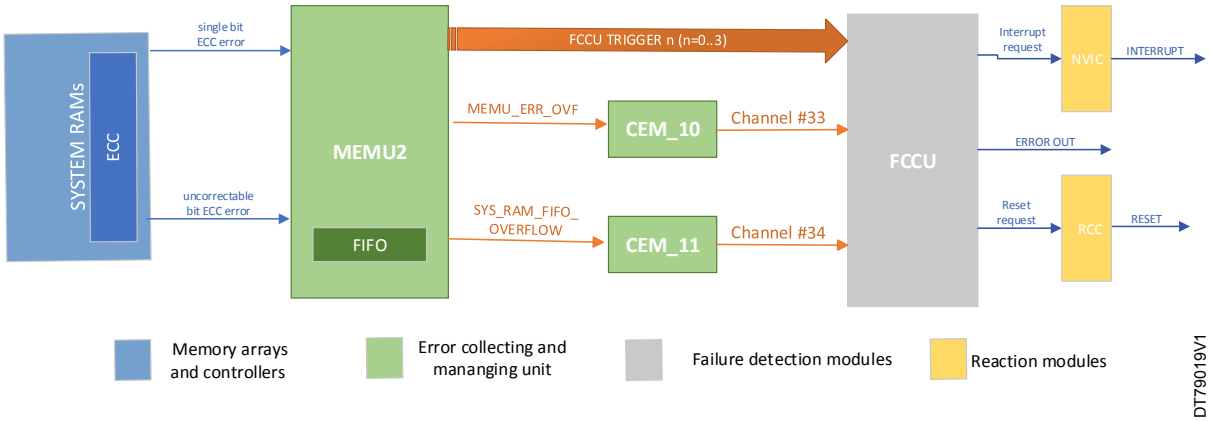


Table 7. System RAM ECC fault descriptor trigger value

SYS_RAM_ECC_FD_CTRL[n]. FCCU_TRG		FCCU channel#
EN	FCCU_TRG	
1, n = 0 0, 1 ≤ n <4 (default configuration after power-on reset)	0x0: SYS_RAM_TRIG_0 is asserted	21
	0x1: SYS_RAM _TRIG_1 is asserted	22
	0x2: SYS_RAM _TRIG_2 is asserted	23
	0x3: SYS_RAM _TRIG_3 is asserted	24
	0xF: no FCCU trigger is asserted	N.A.

2.3 Error reporting for peripheral RAM

This section is dedicated to error reporting for the peripheral RAM.

When a correctable or uncorrectable bit error occurs, the error is captured by MEMU2 and, if enabled, forwarded to the FCCU. See Figure 4 for details of the link between peripheral RAM, MEMU2, CEM, and FCCU.

If MEMU2 receives an error before the software programs the fault-descriptor registers, it is routed to FCCU channel #25. If you want to set another FCCU channel, refer to Table 8.

If a FIFO error overflow (PERIPH_RAM_OVRFLOW_FIF0_x) occurs, the error is captured by CEM_12 and reported to the FCCU through channel #35.

If a FIFO error overflow occurs, the error is captured in the PERIPH_RAM_OFLWn register and reported to CEM_10 and, if enabled, forwarded to FCCU channel #33. When one bit between PR_CEO, or PR_UCEO in the ERR_FLAG register is asserted, it can indicate the occurrence of the following condition: overflow in correctable or uncorrectable reporting tables occurs, depending on the flag set.

Overflow occurs when the relevant table is full, all entries of this table are valid, and a new “unique” entry arrives.

Figure 4. Error reporting for peripheral RAMs

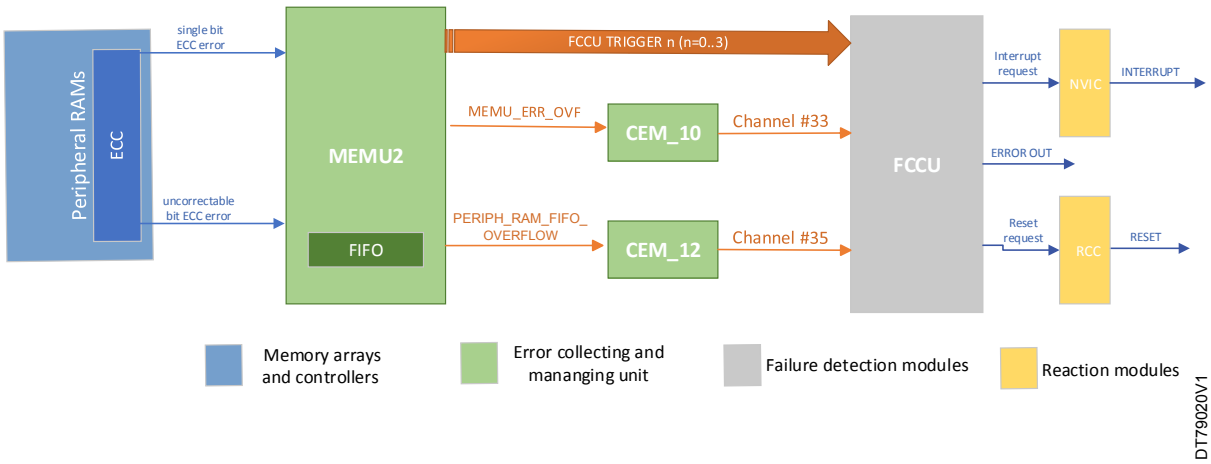


Table 8. Peripheral ECC fault descriptor trigger value

PERIPH_RAM_ECC_FD_CTRL[n]. FCCU_TRG		FCCU channel#
EN	FCCU_TRG	
1, n = 0 0, 1≤ n < 4 (default configuration after power-on reset)	0x0: PERIPH_RAM_TRIG_0 is asserted	25
	0x1: PERIPH _RAM _TRIG_1 is asserted	26
	0x2: PERIPH _RAM _TRIG_2 is asserted	27
	0x3: PERIPH _RAM _TRIG_3 is asserted	28
	0xF: no FCCU trigger is asserted	N.A.

Note:
- *For CEM details, refer to reference manual [1].*
- *For the injection procedure to trigger ECC and overflow, and the interrupt/reaction, refer to [2].*

2.4 Application information

The primary function of the MEMU2 module is to collect and report correctable and uncorrectable ECC errors. An example is explained for each memory link to MEMU2.

Note: Refer to the safety manual [6] of the SR5 E1 device for the list of safety requirements related to the MEMU2. These guidelines align with the SR5 E1 safety concept to ensure the proper use of the MCU in a safety related application.

3 Hardware and software implementation

In this section, some practical examples are explained.

Test conditions: only one ECC descriptor (fault-descriptor 0) for each of three MEMU2 instances using the default values.

Note: User software must configure the number of memory regions, and their limits (start address and end address) based on the safety levels of the application.

The focus is on the reaction of the MEMU2 module, not on how to inject the error or the FCCU settings or its reaction.

This document uses the SR5E1E7 and its EVBE7000P evaluation board. See the SR5Ex reference manual [1] and EVBE7000P user manual [5] for further details. The user can utilize these concepts for all other devices belonging to the SR5E1 family.

3.1 Testing the MEMU2 with error injection in code NVM

The SR5 E1 line devices feature a mechanism to inject safety relevant errors and to test the reaction path. For each read-while-read partition (RWR_0 and RWR_1), dedicated slots permit to directly trigger the MEMU2 NVM tables: correctable or uncorrectable error.

Table 9. NVM error injection memory mapping

Injected error	RWR0 by NVMC1	RWR1 by NVMC2
Customer single bit correction area	0x1FF80280	0x1FFBC280
Customer uncorrectable bit detection area	0x1FF802C0	0x1FFBC2C0

The focus is on the reaction of MEMU2 for the flash memory controller NVMC1, as the results for NVMC2 are similar.

For simplicity, the default values are used (this step will not be mentioned below).

In the following examples, the ECC fault descriptor 0 is supposed to be enabled and configured to map the whole device memory (default configuration).

In the below table there is the MEMU2 descriptor configuration.

Table 10. MEMU2 descriptor configuration

MEMU2 NVM descriptor 0	Value
NVM_ECC_FD_CTRL [0]. EN	1
NVM_ECC_FD_START [0]	0x0
NVM_ECC_FD_END [0]	0xFFFF_FFFF

3.1.1 Single bit error reaction

Single bit corrections of NVM modules are always active while the logging of these errors is disabled by default.

The user software can enable the reporting of ECC single bit correction through the NVM interface by setting UT0.SBCE1 bit, resulting in the MCR.SBC1 flags and related address to be captured in ADR.

Only the errors from code NVM module may be instantiated to MEMU2 NVM tables and then directed to the FCCU module.

The triggering of these errors to FCCU requires the enabling detection of single bit correctable fault and the selecting of the FCCU trigger number to be activated.

Table 11. NVM SBC configuration

IP interface	Register/Bit field	Value	Description
NVM	UTO	0xF9F99999	Write access to UTO. (Set UTE in UTO: Enable user test)
	UTO.SBCE1	1	This is used as an enable for interrupt signals created by the embedded flash memory. ECC corrections that occur when SBCE1 is cleared are not logged.
MEMU2	NVM_ECC_FD_CTRL [0]. SBC	1	Single bit ECC errors must be considered
	NVM_ECC_FD_CTRL [0]. FCCU_TRG	0	It is possible to change this value by changing the FCCU channel triggering; for more details, see Table 6

Once configured the NVM interface and the MEMU2 tables, as showed in Table 11, the user software can read the UTEST content at address 0x1FF80280, to test the reaction path of a single bit correction error from RWR_0.

Table 12. NVM SBC detection, IPs status

IP interface	Register/Bit field	Value	Description
NVM	MCR.SBC1	1	SBC1 provides information on previous reads.
	ADR	-	The address register provides the first address at which an ECC correction occurs
MEMU2	NVM_SB_CERR_STS [0]. VLD	1	the entry in the reporting table is valid
	ERR_FLAG. F_SCE	1	Indicates that a new and unique NVM ECC single bit correctable error was detected.
	NVM_SB_CERR_ADDR [0]	0x1FF80280	Indicates the address on which the error is detected.

If all these conditions described in Table 12 are met, the test is executed successfully.
Refer to [2] about the reporting to FCCU interface.

3.1.2 Uncorrectable bit error reaction

To capture uncorrectable errors, you need to enable uncorrectable ECC errors and configure the NVM ECC trigger by following the steps shown in the table below.

Table 13. NVM UB configuration

IP interface	Register/Bit field	Value	Description
MEMU2	NVM_ECC_FD_CTRL [0].UC = 1	1	Uncorrectable ECC errors must be considered
	NVM_ECC_FD_CTRL [0]. FCCU_TRG	0	It is possible to change this value by changing the FCCU channel triggering; for more details, see Table 6

Once configured the NVM interface and the MEMU2 tables, as showed in Table 13, the user software can read the UTEST content at address 0x1FF802C0 to test the reaction path of uncorrectable error from RWR_0.

Table 14. **NVM UB detection, IPs status**

IP interface	Register/Bit field	Value	Description
NVM	MCR.EER	1	ECC uncorrectable event error EER provides information on previous reads.
	ADR	-	The address register provides the first address at which an ECC uncorrectable error occurs
MEMU2	NVM_UNCERR _STS [0].VLD	1	The entry in reporting table is valid
	ERR_FLAG.F_UCE	1	Indicates that a new and unique NVM ECC uncorrectable error is detected.
	NVM_ NVM_UNCERR_ADDR [0]	0x1FF802C0	Indicates the address on which the error is detected.

If all these conditions are verified, the test is executed successfully.

Note: For DATA NVM, the NVM interface is used to capture potential errors.

3.2 Testing the MEMU2 with error injection in system RAM

To test the capturing of errors by the MEMU2 module from system RAM, the ITCM is the memory under test.

For the injection procedure, refer to "Section 3.1 Testing ITCM ECC logic" of [3].

For simplicity, the default values are used (this step will not be mentioned below).

In the following examples, the ECC fault descriptor 0 is supposed enabled and configured to map the whole device memory (default configuration).

In the below table there is the MEMU2 descriptor configuration.

Table 15. **MEMU2 descriptor configuration**

MEMU2 SYS RAM descriptor 0	Value
SYS_RAM_ECC_FD_CTRL [0]. EN	1
SYS_RAM_ECC_FD_START [0]	0x0
SYS_RAM_ECC_FD_END [0]	0xFFFFFFFF

3.2.1 Single bit error reaction

Inject a single bit error from system RAM ITCM address 0x00000008, following the procedure described in [3].

Table 16. **System RAM SBC configuration**

IP interface	Register/Bit field	Value	Description
MEMU2	SYS_RAM_ECC_FD_CTRL [0]. SBC	1	Single bit ECC errors must be considered
	SYS_RAM _ECC_FD_CTRL [0]. FCCU_TRG	0	It is possible to change this value by changing the FCCU channel triggering; for more details, see Table 9

Once configured the MEMU2 tables, as shown in Table 16, the user software can read the system RAM ITCM content at address 0x00000008, to test the reaction path of single bit correction error from system RAM.

Table 17. **System RAM SBC detection**

IP interface	Register/Bit field	Value	Description
MEMU2	SYS_RAM_CERR_STS [0]. VLD	1	The entry in the reporting table is valid
	MEMU2.ERR_FLAG. SR_CE	1	Indicates that a new and unique NVM ECC single bit correctable error was detected.
	SYS_RAM_CERR_ADDR [0]	0x5A000008(1)	Indicates the address on which the error is detected.

1. *Equivalent address to 0x00000008 in Core1 ITCM (indirect mode access) range. See reference manual [1] for details.*

If all these conditions described in Table 17 are met, the test is executed successfully.
Refer to [2] about the reporting to FCCU interface.

3.2.2 Uncorrectable bit error reaction

Inject uncorrectable bit error from system RAM ITCM address 0x00000008, following the procedure described in [3].

Table 18. System UB configuration

IP interface	Register/Bit field	Value	Description
MEMU2	SYS_RAM_ECC_FD_CTRL [0]. UC	1	Uncorrectable ECC errors must be considered
	SYS_RAM_ECC_FD_CTRL [0]. FCCU_TRG	0	It is possible to change this value by changing the FCCU channel triggering; for more details, see Table 6

Once configured the MEMU2 tables, the user software can read the RAM ITCM content at address 0x00000008, to test the reaction path of uncorrectable error from system RAM.

Table 19. System UB detection

IP interface	Register/Bit field	Value	Description
MEMU2	SYS_RAM_UNCERR_STS [0]. VLD	1	The entry in reporting table is valid
	ERR_FLAG.SR_UCE	1	Indicates that a new and unique NVM ECC uncorrectable error is detected.
	SYS_RAM_UNCERR_ADDR [0]	0x5A000008(1)	Indicates the address on which the error is detected.

1. *Equivalent address to 0x00000008 in Core1 ITCM (indirect mode access) range. See reference manual [1] for details.*

If all these conditions are verified, the test is executed successfully.

3.3 Testing the MEMU2 with error injection in peripheral RAM

To test the capturing of errors by the MEMU2 module from peripheral RAM, the CAN RAM is the memory under test.

For the injection procedure, refer to "Section 3.2 Testing CAN RAM ECC logic" of [3].

For simplicity, the default values are used (this step will not be mentioned below).

In the following examples, the ECC fault descriptor 0 is supposed enabled and configured to map the whole device memory (default configuration).

In the table below, the MEMU2 descriptor configuration is provided:

Table 20. MEMU2 descriptor configuration

MEMU2 PERIPH RAM descriptor 0	Value
MEMU2. PERIPH_RAM_ECC_FD_CTRL [0]. EN	1
MEMU2. PERIPH_RAM_ECC_FD_START [0]	0x0
MEMU2. PERIPH_RAM_ECC_FD_END [0]	0xFFFFFFFF

3.3.1 Single bit error reaction

Inject a single bit error at CAN RAM address 0x4200C000, following the procedure described in the [3].

Table 21. Peripheral RAM SBC configuration

IP interface	Register/Bit field	Value	Description
MEMU2	PERIPH_RAM_ECC_FD_CTRL [0]. SBC	1	Single bit ECC errors must be considered
	PERIPH_RAM_ECC_FD_CTRL [0]. FCCU_TRG	0	It is possible to change this value by changing the FCCU channel triggering; for more details, see Table 10

Once configured the MEMU2 tables, the user software can read the CAN RAM content at address 0x4200C000, to test the reaction path of single bit correction error from peripheral RAM.

Table 22. Peripheral RAM SBC detection

IP interface	Register/Bit field	Value	Description
MEMU2	PERIPH_RAM_ECC_FD_CTRL [0]. VLD	1	The entry in the reporting table is valid
	ERR_FLAG. PR_CE	1	Indicates that a new and unique NVM ECC single bit correctable error was detected.
	SYS_RAM_CERR_ADDR [0]	0x4200C000	Indicates the address on which the error is detected.

If all these conditions described in Table 22 are met, the test is executed successfully.
Refer to [2] about the reporting to FCCU interface.

3.3.2 Uncorrectable bit error reaction

Inject an uncorrectable bit error at CAN RAM address 0x4200C000, following the procedure described in [3].

Table 23. Peripheral UB configuration

IP interface	Register/Bit field	Value	Description
MEMU2	PERIPH_RAM_ECC_FD_CTRL [0].UC	1	Uncorrectable ECC errors must be considered
	PERIPH_RAM_ECC_FD_CTRL [0]. FCCU_TRG	0	It is possible to change this value by changing the FCCU channel triggering; for more details, see Table 6

Once configured the MEMU2 tables, the user software can read the CAN RAM content at address 0x4200C000, to test the reaction path of uncorrectable error from peripheral RAM.

Table 24. Peripheral UB detection

IP interface	Register	Value	Description
MEMU2	PERIPH _RAM_UNCERR_STS [0].VLD = 1	1	The entry in the reporting table is valid
	ERR_FLAG.PR_UCE = 1	1	Indicates that a new and unique NVM ECC uncorrectable error is detected.
	PERIPH _RAM_UNCERR_ADDR [0]	0x4200C000	Indicates the address on which the error is detected.

If all these conditions are verified, the test is executed successfully.

Note:

1. *The bits within the MEM_ID field of xxxx_xxx__CERR_STSn are not intended for user access and must be considered RESERVED.*
2. – *From a functional point of view, if the user wants to clear an error entry in the reporting table, it is only required to clear the VLD bit. In this case, the row of the table will be reused to capture a new error (bad bit/address).*
   – *From a functional safety point of view, if the user wants to clear an error entry in the reporting table, the software must set all the bits of the reporting table entry to invalidate the entry as a countermeasure in case of a failure affecting the VLD bit (for more details refer to safety manual [6]).*

4 Conclusion

By following the guidelines outlined in this document, users can configure and monitor ECC errors effectively using the MEMU2 module. This ensures robust error detection and correction in automotive applications. For further details on each module, refer to reference manual [1]. Additionally, a reference code is available to assist with the implementation and validation process.

Appendix A Acronyms, abbreviations, and reference documents

Acronyms and abbreviations

API	Application programming interface
MEMU2	Memory error management unit 2
FCCU	Fault collection and control unit
ECC	Error correction code
CEM	Combined error manager
NVM	Non-volatile memory
RWR	Read-While-Read partition

Reference documents

[1] Reference manual *SR5E1x 32-bit Arm® Cortex®-M7 architecture microcontroller for electrical vehicle applications* (RM0483)

[2] Application note *SR5 E1 line – FCCU fault sources and reaction* (AN6042)

[3] Application note *SR5 E1–IMA usage* (AN6157)

[4] Errata sheet *SR5 E1 line–Devices errata JTAG_ID = 0x0012_2041 and 0x1012_2041* (ES0597)

[5] User manual *SR5E1-EVBE7000P rev. C evaluation board* (UM3104)

[6] User manual *SR5 E1 line–Safety manual* (UM3421)

Revision history

Table 25. **Document revision history**

Date	Revision	Changes
07-Jul-2025	1	Initial release.

Contents

List of tables

List of figures

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2025 STMicroelectronics – All rights reserved