TN1279
Technical note

SPC58xEx - End to end ECC

Introduction

The ECC provides a protection against errors in data stored in memories and improves the functional safety of the system avoiding errors caused by reading these wrong data. The main impact of ECC is in volatile memories (RAMs) where miniaturization of technology causes a higher risk of a bit flip.

The Hamming code is one of the most used algorithms used for ECC. It works on parity bits applied on sequence of data.

In order to make the system more robust and safe, an end to end ECC (e2eECC) is introduced which allows the detection of data or address corruption on all the data paths between the memory where the data is stored and any module that uses it in the MCU with a high coverage.

This technical note is an extension for the SPC58xEx/SPC58xGx/SPC58xNx family of an application note written for SPC56 microcontrollers (i.e. ECC management on SPC560x - Application Note).

Contents

1 Comparison of e2eECC vs ECC

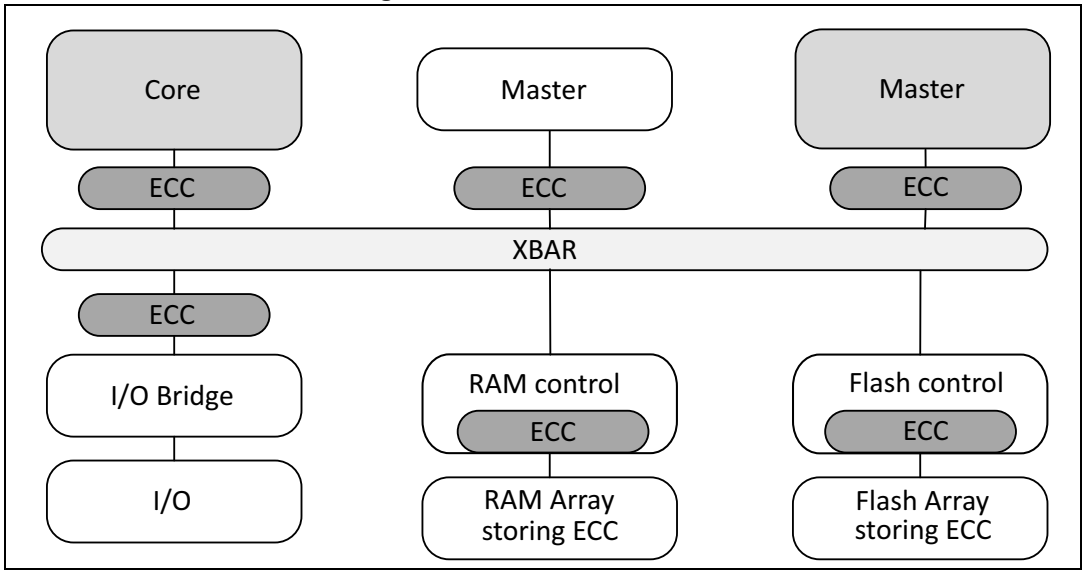
The e2eECC scheme of the volatile memories implements a single-error correction, double-error detection (SECDED) code using the Hsiao odd-weight column criteria, an implementation of the Hamming algorithm with a distance of 4. It is organized with 64 bits of data plus 29 address bits (the upper bits of the 32-bit address field minus the 3 bits which select the byte within the 64-bit data field). This results in a (101, 93) code, 93 is the total number of data bits and 101 is the total number of data bits (93) plus 8 checkbits[a].

The e2eECC generates error protection codes at the source of data generation by XBAR masters (including, but not limited to the Safety Core). It stores the data and error protection codes to the memory when a write operation is initiated by a master. It performs a data integrity check using the previously stored error protection codes when a read operation is requested by a master.

In principle, no ECC logic is necessary at the memories themselves where the protection codes are stored. However, as the storage architecture may use a different ECC schema[b], the SRAM and FLASH controllers may embed additional ECC logic to convert the original ECC bits into the ECC bits saved in the memory and vice versa.

The e2eECC is transparent to the user. As ECC logic is a source of single point faults, the hardware monitors its functionality using replication (e.g. in the cores) or an EDC after ECC mechanism (e.g. in the SRAM or FLASH controller).

Figure 1. Structure of e2eECC

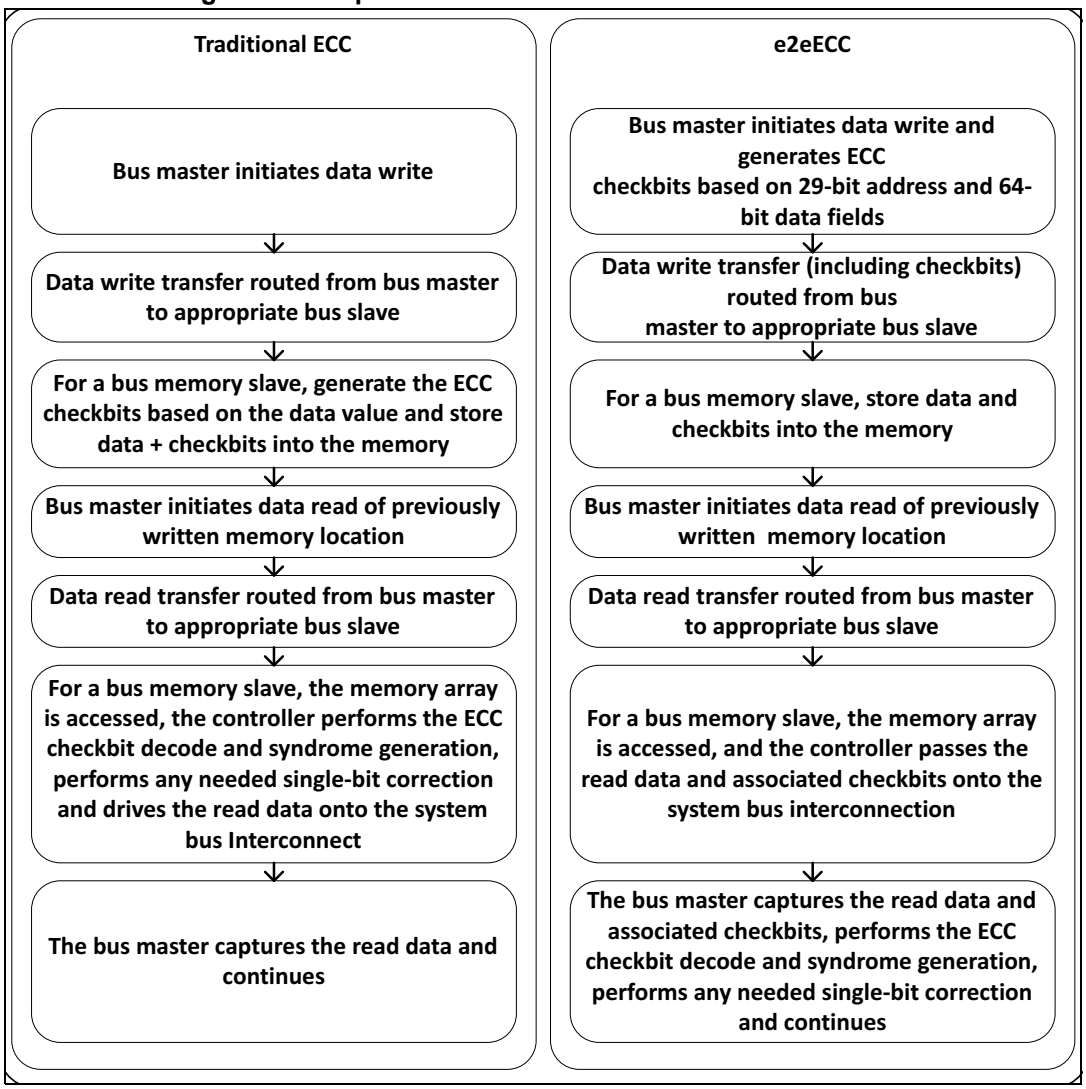


The *Figure 2* compares two different approaches for the management of data errors, the ECC and the e2eECC.

a. Refer to SPC58xEx Reference manual for the H matrix.

b. ECC granularity of the FLASH is 64 bits for data partitions and 128 bits for the code partitions. Memories in peripheral modules, in general, use an ECC without address error protections.

Figure 2. Comparison between e2eECC and common ECC



The main benefit of e2eECC vs ECC is that the e2eECC covers the entire data path. Indeed, it also detects failures occurring in the XBAR bus. The e2eECC is structurally different from regular ECC. It provides error detection from one endpoint to the other endpoint of the data path. ECC is generated locally, transported and stored with data.

In both cases (i.e. e2eECC and local ECC), the hardware assesses the presence of an error during reading accesses. As a consequence, a fault can remain latent in the memory for a certain amount of time until a master reads the faulty location.

Faults in the data path can cause different failures: incorrect address, incorrect write data, incorrect read data or incorrect size of transaction. In order to reach the highest integrity level, additional dedicated safety mechanisms are implemented in the data path and memories: dataless replication, address feedback mechanism and RAM/FLASH array columns muxing[c].

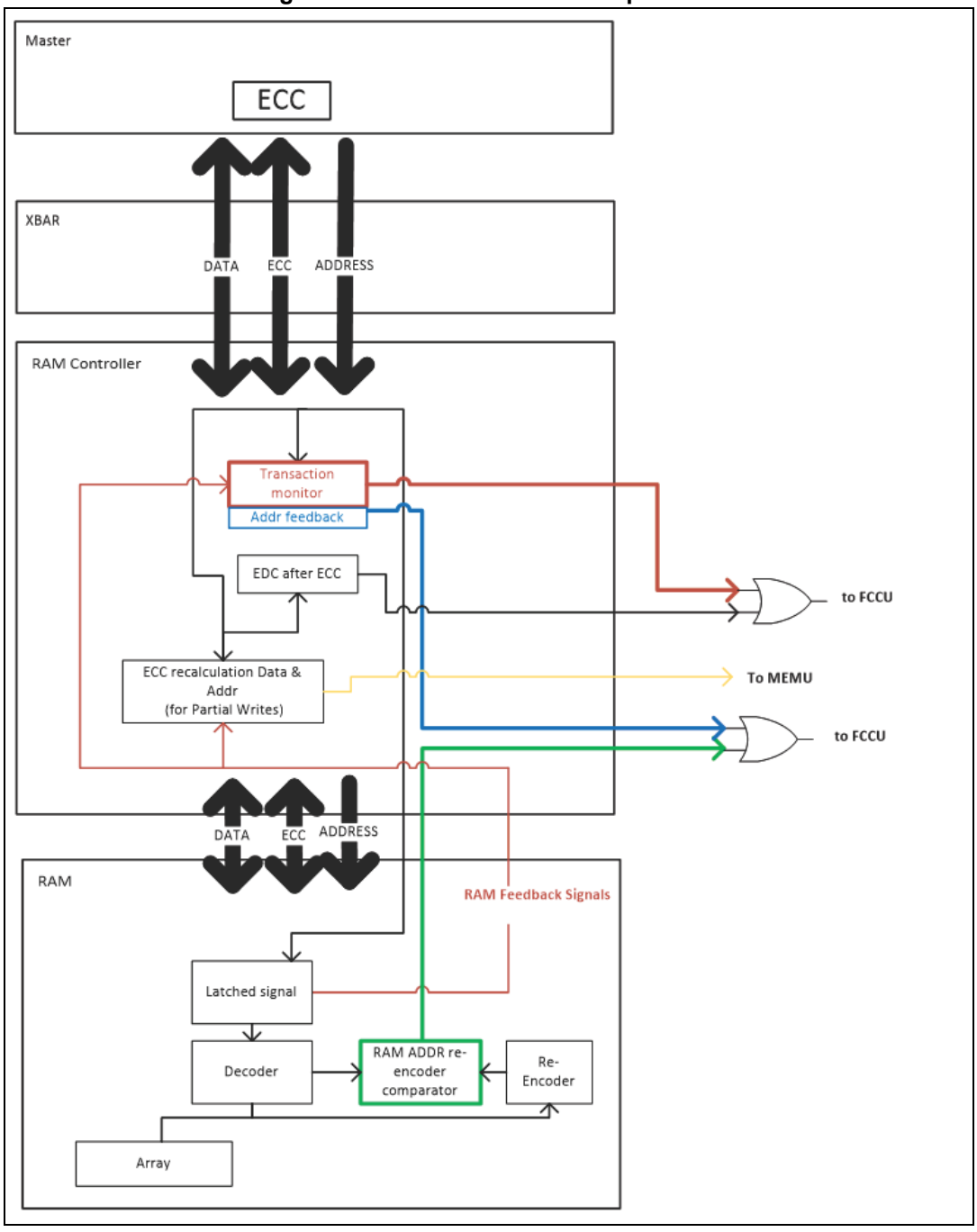
c. Column muxing maps logical data to physical data spread over several words to reduce the probability that a MCU (multi columns upset) transforms in a MBE (multi bit error).

2 E2eECC write and read

The following two sections describe the process of read or write between master and slave of the XBAR bus.

The *Figure 3* shows the different blocks involved during the read/write operations of a XBAR master to the SRAM.

Figure 3. SRAM write and read operation



2.1 System write

A write cycle consists of the following steps:

- Address and data are sent to e2eECC check bit generation logic to generate the check bits according to the encoding. During bus transfer the check bits are driven with the data.
- If there is a partial-width write[d] the slave uses the address, data and check bits to read the data written in the location, regenerate the syndrome by combining the already written data with the new one and then perform the write. The local ECC module detects or corrects errors during the read operation. The error is reported to the MEMU. If there is a full-width write any ECC error will be checked and possibly corrected when accessing the data.
- The slave stores the data and check bits.

2.2 System read

A read cycle consists of the following steps:

- The address to be read is sent to the transaction address queue to calculate a partial syndrome.
- The slave returns the data from the requested address without any ECC checking or corrections. If an error is detected during these phases by the address feedback mechanism, it is reported to the FCCU.
- The e2eECC syndrome checking logic completes the syndrome based on all inputs.
- If an ECC error has occurred it is either detected or corrected. Corrected data is provided to the master. If there is an indication of an address bit error the correction is never done and a machine check exception is generated.

d. It is a partial-width write if the core accesses the memory with a size smaller than the size of a word e.g. if a word in memory is 32-bit width, a partial-width write means that the core accesses the memory with 8-bit or 16-bit access.

3 ECC error injection

Testing or diagnostic activities may require the injection of ECC errors in the memories. The SPC58xEx/SPC58xGx/SPC58xNx MCU provide two mechanisms for accessing and modifying the content of the SRAM memories including the checkbits:

- The built-in CPU e2e ECC test capability provides a path for accessing e2eECC data from the cores.
- An IMA controller module provides registers for selecting, reading, and writing memory data, including the checkbits for subset of the RAM arrays.

Utest FLASH contains an area named “Customer bit correction”. This area is written by ST during production phase with a specific content aimed to generate ECC errors when read.

4 MEMU

The MEMU is responsible for collection and reporting of error events associated with ECC logic used on:

- System RAM[e]
- Peripheral System RAM
- Flash memory[f]

When any of the above events occur, the MEMU receives an error signal together with the following information:

- Type of the error (i.e. either correctable or uncorrectable)
- Memory address where the error occurred
- ECC-syndrome or MBIST bad-bit information

As a consequence, the MEMU performs two actions:

1. records the event in the corresponding error reporting table
2. triggers an error to the related FCCU channel.

Not all ECC units provide a syndrome. If an ECC unit does not provide the syndrome, the MEMU stores 0xFF as the syndrome in the error table.

Figure 4 shows the logical connection between the above components.

Figure 4. ECC event, MEMU and FCCU connections

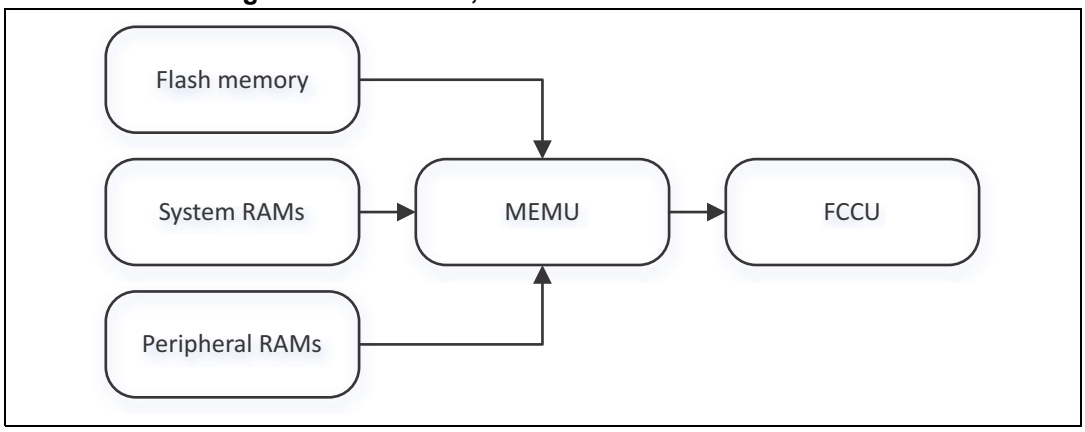


Table 1 shows how many entries for each reporting table the MEMU can store.

The MEMU reporting tables contain enough entries to store the estimated number of correctable and uncorrectable errors that can occur during the lifetime of the device. If there are more requests than available entries, the MEMU set a flag in an overflow register.

e. The system RAMs in this context are those RAMs with “System RAM” memory classification as in the Table “ECC RAM implementations” of the reference manual.

f. The ECC events detected on accesses to the Data Flash blocks are suppressed from being reported to the MEMU.

Table 1. Number of entries inside MEMU

Error Source	# of entries in correctable error report table	# of entries in uncorrectable error report table
Flash	20	1
System RAM	10	1
Peripheral RAM	2	1

5 Handling of errors

5.1 Microcontroller's HW reaction to ECC errors

The microcontroller reacts to the ECC errors in different ways. It depends on which memory it is detected and the type of error detected. The *Table 2* summarizes the behaviour of the devices in the case of different ECC error events during a core access[g].

g. ECC errors reported by ECC logic of the cores.

Table 2. Behavior of devices in case of different ECC errors

MEMORY AREA	ERROR TYPE	HW specific reaction, register's flags	MEMU Error Flag register (ERR_FLAG)	FCCU INDICATION
CODE FLASH	SEC (Single Error Correction)	In FLASH user test mode enabled, setting UT0.SBCE1 = 1 enables to flag the information about any eventual ECC Single Bit Correction in the Flash array (MCR.SBC1 and ADR).	When a new and unique FLASH single bit correctable ECC error is detected the MEMU ERR_FLAG[F_CE] flag is set. **Note:** It depends on MEMU CTRL.FLASH_ERR_SELECT register. The MEMU reporting tables contain a number of entries that are considered enough to store permanent single or double errors that can occur during the lifetime of the device. The overflow of correctable errors reporting tables indicates a large number of errors (permanent or transient) and needs to be treated accordingly by the application. If the MEMU FLASH table overflows, an additional signal is sent to the FCCU.	FCCU channel #27 – MEMU_FLS_CE FCCU channel #29 – MEMU_FLS_OV
	DEC (Double Error Correction)	In FLASH user test mode enabled, setting UT0.SBCE = 1 enables to flag the information about any eventual ECC Double Bit Correction in the Flash array (MCR.SBC and ADR).	When a new and unique FLASH double bit correctable ECC error is detected the MEMU ERR_FLAG[F_CE] flag is set. **Note:** It depends on MEMU CTRL.FLASH_ERR_SELECT register. The MEMU reporting tables contain a number of entries that are considered enough to store permanent single or double errors that can occur during the lifetime of the device. The overflow of correctable errors reporting tables indicates a large number of errors (permanent or transient) and needs to be treated accordingly by the application. If the MEMU FLASH table overflows, an additional signal is sent to the FCCU.	FCCU channel #27 – MEMU_FLS_CE FCCU channel #29 – MEMU_FLS_OV

Table 2. Behavior of devices in case of different ECC errors (continued)

MEMORY AREA	ERROR TYPE	HW specific reaction, register's flags	MEMU Error Flag register (ERR_FLAG)	FCCU INDICATION
CODE FLASH	TED (Triple Error Detection)	In case of an uncorrectable ECC error the FLASH array MCR[EER] bit is set. In case of uncorrectable error during a core access, the bus master jumps to a Machine check exception (triggers corresponding FCCU channel).	When a new FLASH uncorrectable ECC error is detected the MEMU ERR_FLAG[F_UCE] flag is set. There is only one entry available in the MEMU reporting table for UCE. The overflow of uncorrectable errors reporting table needs to be treated accordingly by the application. If the MEMU FLASH table overflows, an additional signal is sent to the FCCU.	FCCU channel #28 – MEMU_FLS_UCE FCCU channel #29 – MEMU_FLS_OV FCCU channel #82, #83, #84 – Core exception
	MULTI BIT ERROR (4 bits or more)	The ECC logic can react to multibit errors in an unpredictable way, the error can be erroneously signaled as single, double bit corrected, triple bit error detected or even as not an error. The HW will also signal/log the error. Multibit error can lead to EDC after ECC indication into the FLASH array MCR[EEE] flag. In case of false detection as uncorrectable error during a core access, the bus master jumps to a Machine check exception and linked FCCU channels are triggered.	The MEMU will log the ECC error according to the indication from the ECC logic.	FCCU channel #27 – MEMU_FLS_CE FCCU channel #28 – MEMU_FLS_UCE FCCU channel #29 – MEMU_FLS_OV FCCU channel #82, #83, #84 – Core exception FCCU channel #58,#64 EDC_ECC_FLASH 0/1
DATA FLASH	SEC (Single Error Correction)	In FLASH user test mode enabled, setting UT0.SBCE1 = 1 enables to flag the information about any eventual ECC Single Bit Correction in the Flash array (MCR.SBC1 and ADR).	No Reporting	No indication in FCCU
	DEC (Double Error Correction)	In FLASH user test mode enabled, setting UT0.SBCE = 1 enables to flag the information about any eventual ECC Double Bit Correction in the Flash array (MCR.SBC and ADR).	No Reporting	No indication in FCCU

Table 2. Behavior of devices in case of different ECC errors (continued)

MEMORY AREA	ERROR TYPE	HW specific reaction, register's flags	MEMU Error Flag register (ERR_FLAG)	FCCU INDICATION
DATA FLASH	TED (Triple Error Detection)	In case of an uncorrectable ECC error the FLASH array MCR[EER] bit is set. In case of uncorrectable error during a core access, a fixed, illegal opcode value is returned to the requesting master along with the associated ECC checkbits.	No Reporting	No indication in FCCU
	MULTI BIT ERROR (4 bits or more)	The ECC logic can react to multibit errors in an unpredictable way, the error can be erroneously signaled as single, double bit corrected, triple bit error detected or even as not an error. The HW will also signal/log the error. Multibit error can lead to EDC after ECC indication into the FLASH array MCR[EEE] flag. In case of false detection as uncorrectable error during a core access, a fixed, illegal opcode value is returned to the requesting master along with the associated ECC checkbits.	No Reporting	FCCU channel #58,#64 EDC_ECC_FLASH 0/1

Table 2. Behavior of devices in case of different ECC errors (continued)

MEMORY AREA	ERROR TYPE	HW specific reaction, register's flags	MEMU Error Flag register (ERR_FLAG)	FCCU INDICATION
System RAM	Correctable Error (SEC)	None	When a new and unique System RAM single bit correctable ECC error is detected the MEMU ERR_FLAG[SR_CE] flag is set. The MEMU reporting tables contain a number of entries that are considered enough to store permanent single bit errors that can occur during the lifetime of the device. The overflow of correctable errors reporting tables indicates a large number of errors (permanent or transient) and needs to be treated accordingly by the application. If the MEMU SRAM table overflows, an additional signal is sent to the FCCU.	FCCU channel #21 – MEMU_RAM_CE FCCU channel #23 – MEMU_RAM_OV
	Uncorrectable Error (DED)	In case of uncorrectable error during a core access, the bus master jumps to a Machine check exception (triggers corresponding FCCU channel).	When a new System RAM uncorrectable ECC error is detected the MEMU ERR_FLAG[SR_UCE] flag is set. There is only one entry available in the MEMU reporting table for UCE. The overflow of uncorrectable errors reporting table needs to be treated accordingly by the application. If the MEMU System RAM table overflows, an additional signal is sent to the FCCU.	FCCU channel #22 – MEMU_RAM_UCE FCCU channel #23 – MEMU_RAM_OV FCCU channel #82, #83, #84 – Core exception

Table 2. Behavior of devices in case of different ECC errors (continued)

MEMORY AREA	ERROR TYPE	HW specific reaction, register's flags	MEMU Error Flag register (ERR_FLAG)	FCCU INDICATION
Peripheral RAM (e.g. CAN)	Correctable Error (SEC)	None	When a new and unique Peripheral RAM single bit correctable ECC error is detected the MEMU ERR_FLAG[PR_CE] flag is set. The MEMU reporting tables contain a number of entries that are considered enough to store permanent single bit errors that can occur during the lifetime of the device. The overflow of correctable errors reporting tables indicates a large number of errors (permanent or transient) and needs to be treated accordingly by the application. If the MEMU SRAM table overflows, an additional signal is sent to the FCCU.	FCCU channel #24 – MEMU_PER_CE FCCU channel #26 – MEMU_PER_OV
	Uncorrectable Error (DED)	In case of uncorrectable error during a core access, the bus master jumps to a Machine check exception (triggers corresponding FCCU channel).	When a new Peripheral RAM uncorrectable ECC error is detected the MEMU ERR_FLAG[PR_UCE] flag is set. There is only one entry available in the MEMU reporting table for UCE. The overflow of uncorrectable errors reporting table needs to be treated accordingly by the application. If the MEMU Peripheral table overflows, an additional signal is sent to the FCCU.	FCCU channel #25 – MEMU_PER_UCE FCCU channel #26 – MEMU_PER_OV FCCU channel #82, #83, #84 – Core exception

5.2 Application's reaction to ECC errors

The application must handle the occurrence of ECC errors based on the system requirements.

From a functional safety point of view, the application must move the system to a safe state[h] in case of uncorrectable errors and optionally, log the indication of correctable ones while continue providing the safety relevant services.

However, a more sophisticated approach can maximize the availability of the service and at the same time, satisfy the safety requirements.

h. The definition of safe state of the application depends on its safety analysis.

In the following sections, it is assumed that the FCCU is configured to go to the alarm state and trigger an interrupt when an ECC event occurs and it is logged in the MEMU. After a timeout[i], if the software has not recovered and has not cleared the related FCCU status register, the FCCU goes to the FAULT state and triggers an external reaction (either via the error out pins or a reset). It is the responsibility of the user to configure the FCCU accordingly. Its is also assumed that the Machine check in the cores are properly handled.

Moreover, the user implements mechanism in the application to detect errors in data (e.g. safety relevant data redundantly stored), and ECC errors in Data FLASH are handled by an EEPROM emulation driver.

In the case of overflow of a reporting table, the MEMU triggers a fault to the FCCU. In such a case, the software must move the system to its safe state.

5.2.1 Correctable errors

The FCCU IRQ handler manages the indication of a new unique correctable ECC error in a RAM or Peripheral location. It logs the details of the error (e.g. address and syndrome) to the not volatile memory for a later usage.

User can consider recurrent errors as known ECC errors. The application can insert these errors in the MEMU tables at each startup to avoid the IRQ handler execution each time a master accesses these locations. The application can continue to provide the service.

In case of correctable ECC error indication in the FLASH, the software must discriminate between real correctable and multi-bit errors[j] introduced by permanent failures in the Flash control logic (e.g. a failure in read pump, read timing, Vref, and so on). This test consists of a read-back pattern that the software must run on demand in case of a correctable error event.

In case of failure of the control logic of the flash, it is expected that most locations issue an ECC error. As a consequence, if the test pattern triggers a correctable error only in a location, it is a real correctable error and the system can keep providing the service.

If the pattern triggers an ECC error in most of the accessed locations, multi-bit errors affect the flash. As a consequence, the system must go to its safe state.

The software must execute a margin read test after a new correction occurs in the Flash memory. This test gives an indication of the status of the flash. If the margin test fails, the system should go to its safe state.

5.2.2 Uncorrectable errors

In case of uncorrectable ECC errors reported by the hardware, the application may asses if the safety service can continue or if it is necessary to move the system to the safe state.

If an uncorrectable error occurs, during a core access, two separate reactions occur: a machine check to the core and an interrupt triggered by the FCCU. Since the machine check has a priority higher than the interrupt, the core executes the handler of the machine check before the one of the interrupt[k].

i. The failure indication time shall be shorter than the FTTI.

j. A multi-bit error means an error in 4 or more bits of the same word.

k. It is assumed that the external interrupts and Recoverable Interrupt Modes are disabled (i.e MSR.EE=0 and MSR.RI =0)

The machine check handler must try to recover the fault. In addition, it must signal to the interrupt handler of the FCCU if this recover succeeds or does not. If it does not, the FCCU goes to its FAULT state and take its reaction.

When an uncorrectable ECC error occurs during a non-CPU transaction no exception triggers to any core. The error goes to the MEMU and then to the FCCU.

Figure 5 summarizes the flow of the uncorrectable ECC errors processing in IVOR1 handler.

Figure 5. Machine check interrupt handler schema

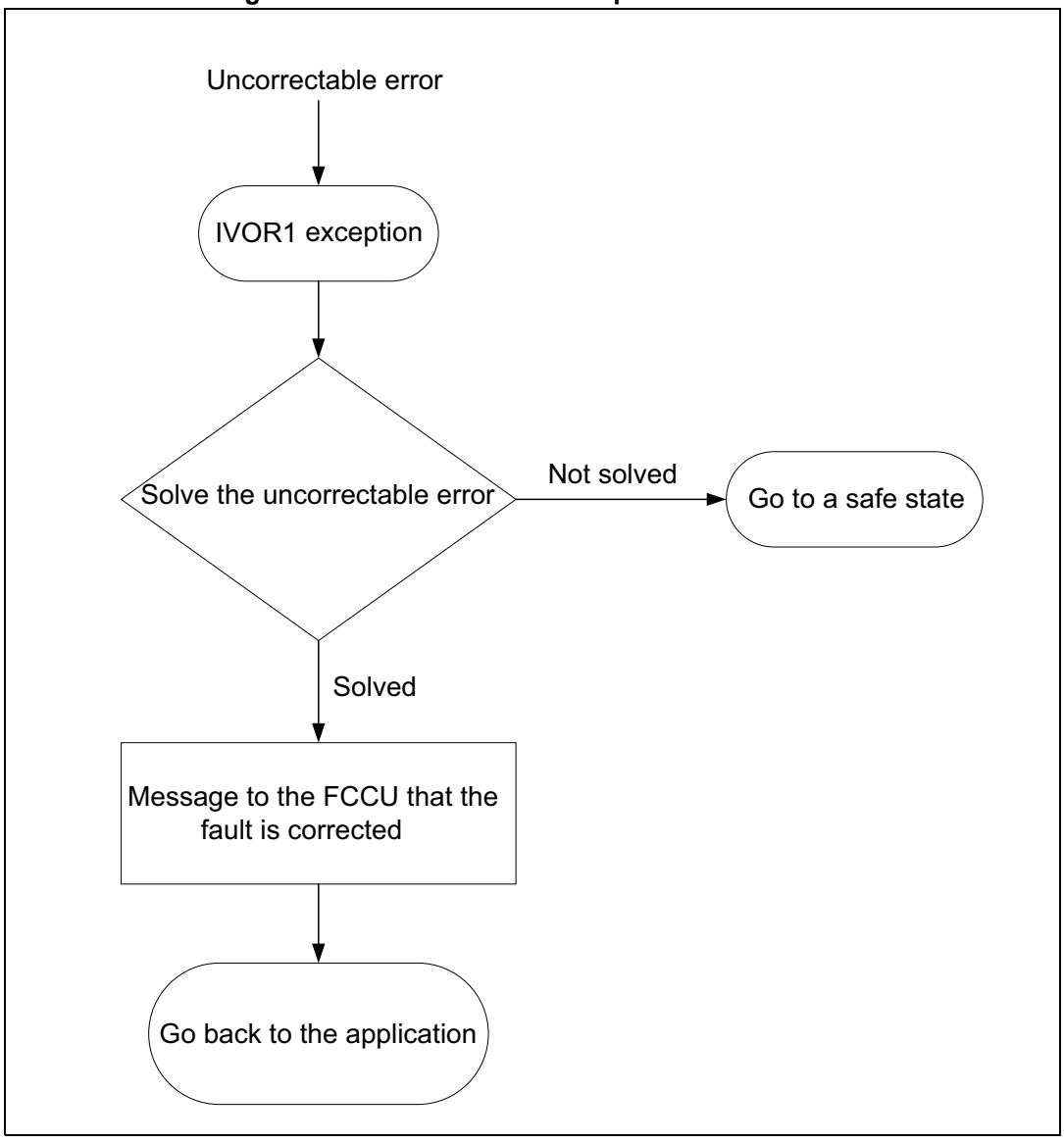
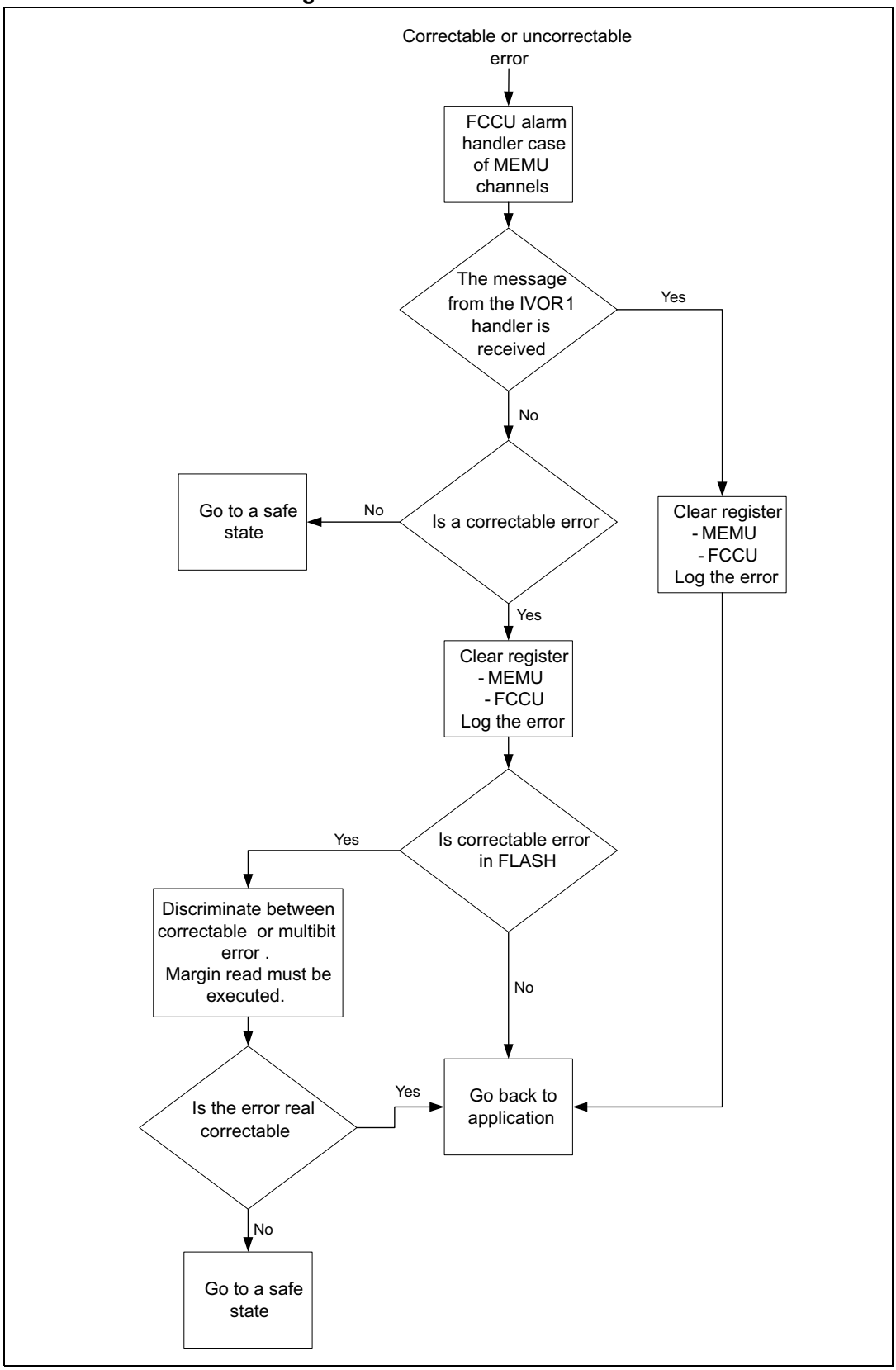


Figure 6. FCCU handler schema



6 Summary

6.1 Error handling

The general recommendation in case of a correctable error is to log it and keep providing the safety-relevant services.

In the case of uncorrectable errors, the safest solution is to move the system into its safe state (e.g. microcontroller in reset) and to stop providing the safety-relevant services. On the other hand, depending on the application, this solution may not be the right trade-off between safety and availability of the system.

As a consequence, the general recommendation of handling uncorrectable errors is trying to recover the system to avoid losing the safety-relevant services. The handler of the machine check can execute the recovery task.

6.2 Evaluation of e2eECC

The e2eECC mechanism provides better robustness in comparison with “normal” ECC. The most significant advantage is that it protects the entire data path of the data against errors.

Appendix A Acronyms and abbreviations

Table 3. Acronyms and abbreviations

Acronym	Definition
DED	Double Error Detection
ECC	Error Correction Code
e2eECC	End-to-end ECC
EDC	Error Detection Code
IMA	Indirect memory access
MBE	Multi Bit Error
MCU	Multiple columns upset
OPCODE	Operation CODE
SBC	Single Bit Correction
SBE	Single Bit Error

Appendix B Document management

Table 4. Reference Documents

Doc name	ID	Title	Version
RM0391	027214	SPC58xEx/SPC58xGx 32-bit Power Architecture® microcontroller for automotive ASILD applications	3
AN4276	024408	ECC management on SPC560x	2

Revision history

Table 5. Document revision history

Date	Revision	Changes
11-Sep-2019	1.0	Initial release.