AN4480

PIC32CM JH00/JH01 High-Level Usage of MBIST and ECC with Fault Injection

Introduction

The PIC32CM JH00/JH01 family of devices support Hamming Error Code Correction (ECC) logic with fault injection on Flash, Data Flash, User Row, Software Calibration Area, Serial Number, and SRAM memory. The PIC32CM JH00/JH01 also embeds an SRAM Memory Built-In Self-Test (SMBIST).

The Hamming Error Code Correction (ECC) is a functional safety feature allowing *Single-Error Correction (SEC)* and *Dual-Error Detection (DED)* for the NVMCTRL and SRAM memory sections. This enables the reduction of error for devices that are under external constraints, which could flip bits overtime.

The SMBIST module can be used to test the internal SRAM memory and its associated ECC bits to detect potential defects, as part of a global functional safety scheme. This document describes the principle and operations of the ECC and SMBIST features.

To illustrate the theory and test the features, the following two MPLAB® Code Configurator (MCC) for MPLAB Harmony v3 code examples will be provided along with this document:

- NVMCTRL ECC with Fault Injection: Allows testing of the ECC with Fault Injection for NVMCTRL by writing a word, injecting one or two errors, and observing the results.
- SRAM MBIST and ECC with Fault Injection: Allows testing of the ECC with Fault Injection for SRAM by writing a word, injecting one or two errors, and observing the results. The SRAM is also tested with the SMBIST feature at boot time.

1. Prerequisites

The following hardware and software are required to illustrate the MPLAB Harmony v3-based examples as shown in Applications Overview.

Hardware Requirements:

- One PIC32CM JH01 Curiosity Pro Board
- One Micro-USB cable

Software Requirements:

- MPLAB X IDE latest version
- MPLAB Code Configurator (MCC) latest version
  - The `csp` package
  - The `csp_apps_pic32cm_jh00_jh01` package

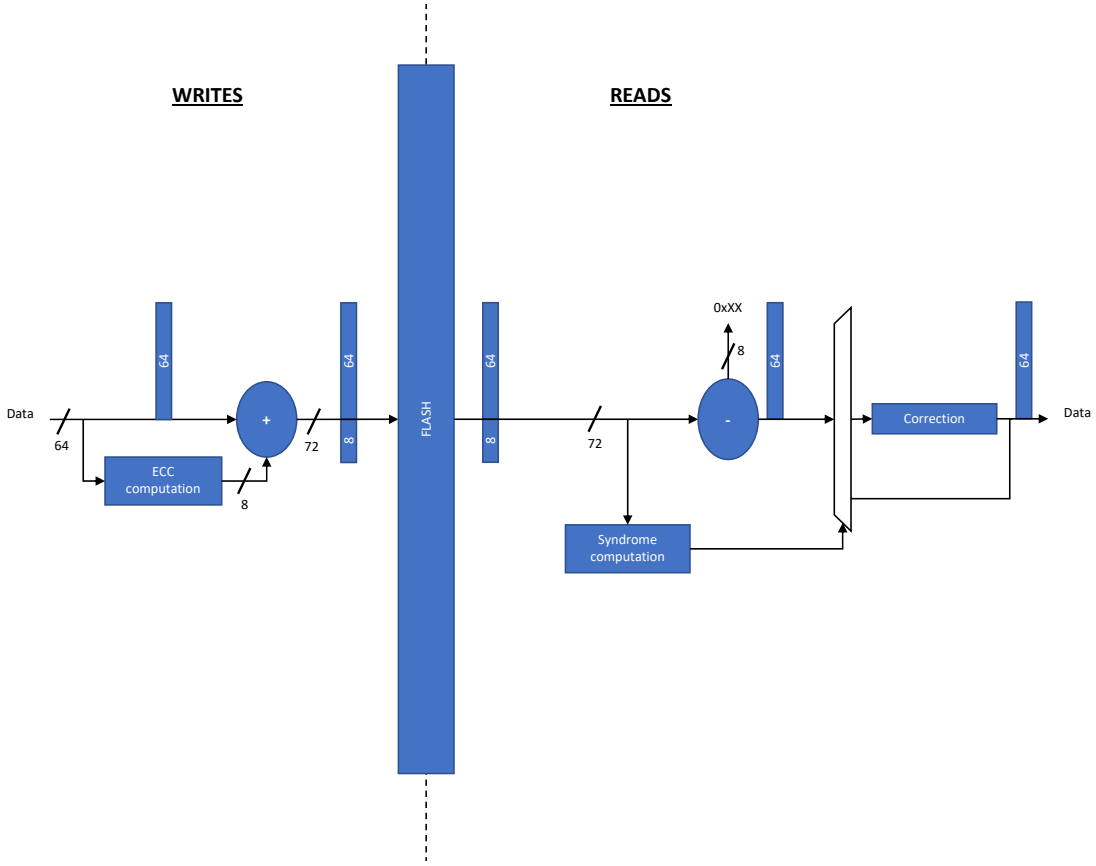
2. Hamming Error Correction Code (ECC)

2.1 Non-Volatile Memory Controller (NVMCTRL)

2.1.1 ECC Principle and Operations

The NVMCTRL embeds the ECC, which is a functional safety feature providing single-error correction (SEC) and double-error detection (DED). It applies to the main Flash, Data Flash, User Row, Software Calibration Area, and Serial Number memory panels. The ECC logic is shown in the following figure:

Figure 2-1. NVMCTRL ECC Logic (Address Bus and Page Buffer are not Represented)



Process Upon Writes: When a 64-bit section of the page buffer is written, the 8-bit ECC is computed on the 64-bit data. Once the page buffer has been written, it can be written in the memory array manually or automatically depending on the CTRLB.MANW value. Each 8-bit ECC will automatically be stored along with the related 64-bit data.

Process Upon Reads: To perform a 8, 16, or 32-bit read in the memory, an internal 72-bit read is performed (64-bit section along with the 8 related ECC bits). If the ECC feature is disabled, the 8, 16, or 32-bit read word is provided with no correction. If the ECC feature is enabled, the ECC syndrome is computed on the 72-bit read in memory and informs whether no error, one, or two errors were detected. In addition to this information, the ECC syndrome informs which one of the 72-bits is defective if a SEC error is detected

The following three cases may occur:

1. **No error is detected:** the 8,16, or 32-bit read word is provided as is.
2. **One error is found:**
   – The error is corrected on the fly, and the corrected byte, half-word, or word is provided seamlessly (the corrected word is written in cache, but is not written back in memory).
   – The ECCCTRL.SECCNT bit field is decremented (until reaching 0).
   – If ECCCTRL.SECCNT = 0x0, INTFLAG.SERR is set, and an interrupt occurs if INTENSET.SERR is set to 0x1.
3. **Two errors are found:**
   – INTFLAG.DERR and INTFLAG.SERR are set to 0x1, and an interrupt occurs if INTENSET.DERR is set to 0x1
   – The Host which tried to read the faulty word receives a bus client error and the corrupted word

Note: A bit error in the address field cannot be detected nor corrected.

Setting the ECCCTRL.SECCNT to any non-zero value allows for the identification of a larger amount of SEC errors being corrected. The counter will decrement upon each SEC error until reaching zero. Then the INTFLAG.SERR will be set, and an interrupt issued (if INTENSET.SERR = 0x1). The counter must then be reloaded manually.

2.1.2 ECC Testing with Fault Injection or Fault Capture

For testing purposes in a functional safety mindset, the user can enable Fault Injection (for injecting on purpose errors for testing) or Fault Capture (for capturing details on the found errors).

The following list highlights bits to configure for ECC with Fault Injection/Capture:

- The bit(s) to be flipped in FFLTPTR. The associated FFLTPTR.FLTxPTR values are shown below:

Table 2-1. FLTxPTR Values

Associated FFLTPTR.FLTxPTR Value	DATA BITS [0:63]	ECC Parity BITS [0:7]
0x0	--	0
0x1	--	1
0x2	--	2
0x3	0	--
0x4	--	3
0x5 to 0x7	1 to 3	--
0x8	--	4
0x9 to 0xF	4 to 10	--
0x10	--	5
0x11 to 0x1F	11 to 25	--
0x20	--	6
0x21 to 0x3F	26 to 56	--
0x40	--	7
0x41 to 0x47	57 to 63	--

- The address where the fault will be injected at in FFLTADR.
- The fault injection type in FFLTCTRL.FLTMD. Refer to the *PIC32CM JH00/JH01 Family Data Sheet* for additional information on fault injection mode for ECC in the NVMCTRL.

If fault injection or fault capture is enabled in FLTCTRL.FLTMD, the computed 8-bit ECC on write will be visible in FFLTPAR.SECOUT for each 64 bits double word written in the page buffer (FFLTPAR.SECIN will always read 0 on writes).

The characteristics of the error captured for SEC or DED errors are logged in the following error capture registers:

- The address of the faulty word in FFLTCAP.FLTADR.
- The parity bits computed on the read/write data in FFLTPAR.
  - For Writes, SECOUT is computed for each 64 bits data written in the page buffer and SECIN is always 0.
  - For Reads, SECOUT is computed on the 64 data bits available after the ECC bits subtraction from 72-bit ECC vector and before the ECC correction logic. SECIN corresponds to the 8 ECC bits extracted from the 72-bit ECC vector.
- If a SEC is detected, the FFLTSYN, FFLTCAP and FFLTPAR registers will be populated with the information related to the SEC error, and locked until either INTFLAG.SERR and INTFLAG.FLTCAP are cleared and a new SEC error is logged, or a DED error occurs. Only then the registers will be updated.
- FFLTSYN.SECSYN indicates which of the 72-bit has been corrupted if an SEC is detected. The following table shows the SECSYN value versus faulty bit location:

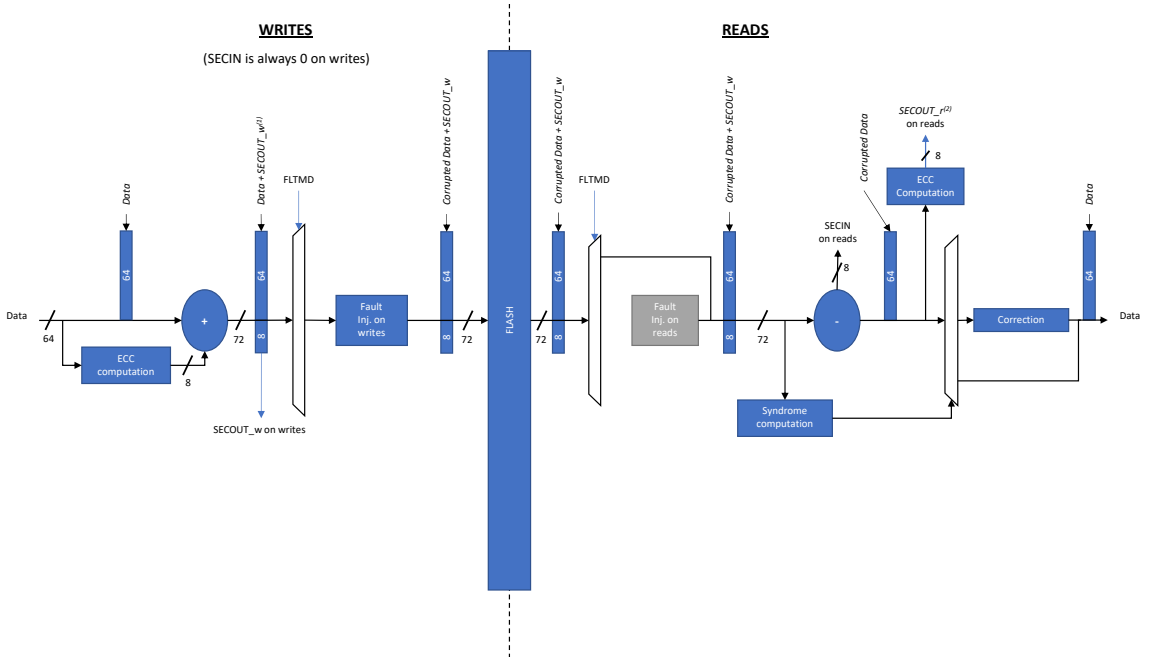
Table 2-2. Syndrome Versus Faulty Bit Location

SECSYN Value	Faulty Bit	SECSYN Value	Faulty Bit	SECSYN Value	Faulty Bit
0x23	D[0]	0x19	D[24]	0xC8	D[48]
0x43	D[1]	0x1A	D[25]	0xD0	D[49]
0x83	D[2]	0x1C	D[26]	0xE0	D[50]
0x3D	D[3]	0xE9	D[27]	0x4F	D[51]
0x45	D[4]	0x2A	D[28]	0x51	D[52]
0x85	D[5]	0x2C	D[29]	0x61	D[53]
0x89	D[6]	0x4C	D[30]	0x62	D[54]
0x49	D[7]	0x4A	D[31]	0x52	D[55]
0x46	D[8]	0x32	D[32]	0x91	D[56]
0x86	D[9]	0x34	D[33]	0xA1	D[57]
0x07	D[10]	0x38	D[34]	0xC1	D[58]
0x7A	D[11]	0xD3	D[35]	0x9E	D[59]
0x8A	D[12]	0x54	D[36]	0xA2	D[60]
0x0B	D[13]	0x58	D[37]	0xC2	D[61]
0x13	D[14]	0x98	D[38]	0xC4	D[62]
0x92	D[15]	0x94	D[39]	0xA4	D[63]
0x8C	D[16]	0x64	D[40]	0x01	ECC[0]
0x0D	D[17]	0x68	D[41]	0x02	ECC[1]
0x0E	D[18]	0x70	D[42]	0x04	ECC[2]
0xF4	D[19]	0xA7	D[43]	0x08	ECC[3]
0x15	D[20]	0xA8	D[44]	0x10	ECC[4]
0x16	D[21]	0xB0	D[45]	0x20	ECC[5]
0x26	D[22]	0x31	D[46]	0x40	ECC[6]
0x25	D[23]	0x29	D[47]	0x80	ECC[7]

- If a DED is detected, the FFLTSYN, FFLTCAP and FFLTPAR registers will be populated with the information related to the DED error and locked until INTFLAG.SERR, INTFLAG.DERR and INTFLAG.FLTCAP are cleared and a new SEC or DED error is logged. Only then the registers will be updated.

The following figure illustrates the fault injection on write and capture logic:

Figure 2-2. NVMCTRL ECC with Fault Injection on Writes



Notes:

1. SECOUT_w refers to the SECOUT value computed on write.
2. SECOUT_r refers to the SECOUT value computed on read.

For additional information on ECC Testing with Fault Injection, refer to the NVMCTRL section of the *PIC32CM JH00/ JH01 Family Data Sheet*.

2.2 SRAM Controller (MCRAMC)

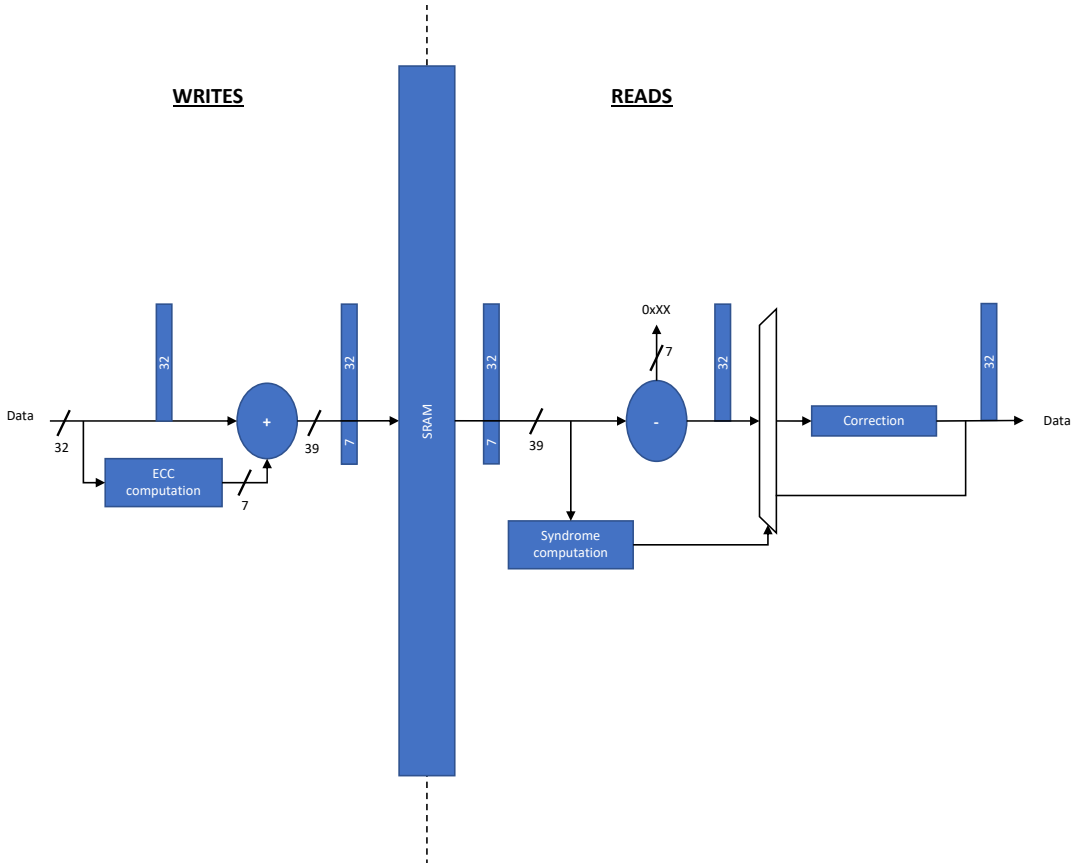
2.2.1 ECC Principle and Operations

The MCRAMC embeds the ECC feature, which improves the global functional safety. The ECC feature on the MCRAMC is enabled by default after reset. It can be disabled/enabled through the CTRLA.ENABLE bit.

Note: The ECC feature can only be disabled for ECC decoding upon reads. The ECC encoding upon writes is always enabled.

The ECC logic on the SRAM provides single-error correction (SEC), and double-error detection (DED), and adds 7 ECC bits (not accessible for the user) for each 32-bit word as shown in the following figure:

Figure 2-3. MCRAMC ECC Logic (Address Bus)



Process Upon Writes

Upon any 8, 16, or 32-bit write in the memory, the seven ECC bits are computed and stored along with the data (the 8 or 16-bit writes are composed of an atomic read of 32 bits, modify, and write 32 bits).

⚠ CAUTION After a reset, the SRAM content (data and ECC) is random and the ECC feature is enabled by default. Any 32-bit write will initialize the data and the related ECC bits. However, 8/16-bit writes will probably trigger an SEC or DED due to the atomic read. Consequently, the SRAM content must be initialized properly before use.

For additional information, refer to the *PIC32CM JH00/JH01 Family Data Sheet*.

Process Upon Reads

Upon any 8, 16 or /32-bit read in the memory, if the ECC feature is disabled, then the SEC or DED errors are not corrected or detected. If the ECC feature is enabled, the ECC syndrome is computed on the related 39-bits (32 data bits + 7 ECC bits).

If an SEC is found, it is corrected on the fly. The corrected data is sent back to the requester and written back in the memory. The INTSTA.SERR Status bit is set and an interrupt is triggered if the interrupt is enabled (INTENSET.SERR = 0x1)..

If a DED is found, a bus error response is issued, typically leading to a synchronous abort exception. This stops the bus Host access sequence precisely at the faulty address. The INTSTA.DERR Status bits is set and an interrupt is triggered if the interrupt is enabled (INTENSET.DERR = 0x1).

The characteristics of the error captured for the SEC or DED errors are logged in following error capture registers:

- The address of the faulty word in ERRCADR.
- The parity bits computed on the read data in ERRCPAR.
- The syndrome in ERRCSYN.ERCSYN (only for SEC error).
- The error type (SEC or DED) in ERRCSYN.ERR1 or ERRCSYN.ERR2.

The following table shows the correspondence between the syndrome value and the faulty bit.

Table 2-3. Syndrome versus Faulty Bit Location

ERCSYN Value	Faulty Bit	ERCSYN Value	Faulty Bit	ERCSYN Value	Faulty Bit
0x61	D[0]	0x2A	D[13]	0x1C	D[26]
0x51	D[1]	0x23	D[14]	0x4C	D[27]
0x19	D[2]	0x1A	D[15]	0x38	D[28]
0x45	D[3]	0x2C	D[16]	0xE	D[29]
0x43	D[4]	0x64	D[17]	0xD	D[30]
0x31	D[5]	0x26	D[18]	0x49	D[31]
0x29	D[6]	0x25	D[19]	0x1	ECC[0]
0x13	D[7]	0x34	D[20]	0x2	ECC[1]
0x62	D[8]	0x16	D[21]	0x4	ECC[2]
0x52	D[9]	0x15	D[22]	0x8	ECC[3]
0x4A	D[10]	0x54	D[23]	0x10	ECC[4]
0x46	D[11]	0xB	D[24]	0x20	ECC[5]
0x32	D[12]	0x58	D[25]	0x40	ECC[6]

2.2.2 ECC Testing with Fault Injection

For testing purpose, single-bit or double-bit faults can be injected during writes at a specific address. The list below highlights the bits to be configured for ECC with Fault Injection testing:

- The address where fault will be injected in FLTADR.
- The bits to be flipped in FLTPTR. The associated FLTPTR.FLTxPTR values are shown in the following table:

Table 2-4. FLTxPTR Values

Associated FFLTPTR.FLTxxPTR Value	DATA Bits [0:31]	ECC Parity Bits [0:6]
0x0 to 0x20	0 to 31	—
0x21 to 0x27	—	0 to 6

- The fault injection mode in FLTCTRL.FLTMD.

Note: The bits are enable-protected. They can only be written when FLTCTRL.FLTEN = 0x0. Make sure to disable this bit before attempting any write on these bit fields to avoid a bus error.

After fault injection, the following constraints must be observed during the whole ECC testing process:

- After changing the configuration, a dummy read access to FLTCTRL is required prior to performing any access to the SRAM.
- When both ECC decoding and fault injection are enabled, no single-bit fault SRAM word must be read at the SRAM fault injection address, because memory correction write-back will inject a fault again.
- When both ECC decoding and fault injection are enabled, a double-bit fault SRAM word at the SRAM fault injection address must be overwritten only with 32-bit access.
- When fault injection is enabled, the data bits to be flipped, as programmed in FLTPTR, must always be part of the bytes modified by the write access to the SRAM fault injection address. A simple way to ensure this is to restrict the write accesses to the SRAM fault injection address to be 32-bit wide only.

3. SRAM Memory Built-In Self-Test (SMBIST)

The SMBIST module can be used to test the internal SRAM memory and its associated ECC logic to detect potential defects, as part of a global functional safety scheme. The module will run a pass or fail test based on the `SMarchCHKBvcd` algorithm. The SRAM content is flushed during the process, hence it must be launched at boot time before any variables, and stacks are stored in the SRAM memory.

No information is provided regarding the number and the location of failing bits. The user is responsible for taking actions and correct the situation, because even if any number of single bit defects will be corrected by the ECC logic, a unique double bit defect will generate a bus error if the location is later accessed by the application software.

The SRAM memory is divided into two partitions: partition 1 and partition 2 for the odd and even addresses respectively.

The following steps outline the procedure for performing the SMBIST testing:

1. Enable the module clock `CLK_SMBIST_APB` in the Main Clock Controller.
   Note: The `CLK_SMBIST_APB` clock module is enabled by default after reset.
2. Run the following instructions from the Flash, ensuring no access to the SRAM:
   a. Clear the STATUS.FAIL and STATUS.DONE bits by writing 0x1 to each of them.
   b. Start the test on both SRAM partitions by writing 0x1 to the CTRL.SMBIST1 and CTRL.SMBIST2 bits.
   c. Wait for STATUS.DONE = 0x1.
   d. Read the STATUS.FAIL bit. The bit is set if the SMBIST test is failed. This means one or more bits is defective in any partition.

Notes:

1. The SMBIST can continue to operate in any Sleep mode where the selected source clock is running.
2. When the CPU is halted in Debug mode, this peripheral will continue normal operation.

4. Applications Overview

4.1 NVMCTRL ECC with Fault Injection Testing

To illustrate the ECC with fault injection feature for the NVMCTRL, an application software is provided along with this document. The goal is to highlight the ECC process upon writes for SEC or DED errors.

Note: For the SEC use case both Flash and Data Flash will be written.

For each use case, the ECC for NVMCTRL is configured as follows:

- ECCCTRL.SECCNT = 0x0
- ECC feature is enabled for Flash and Data Flash memories
- SERR and DERR interrupts are enabled (INTENSET.SERR = INTENSET.DERR = 0x1)
- FFLTPTR.FLT1PTR = 0x3 and FFLTPTR.FLT2PTR = 0x5
- FFLTADR.FLTADR = 0x9000 for Flash memory read/write operations and 0x401000 for Data Flash memory read/write operations
- FLTCTRL.FLTMD = 0x6 (Single-Fault Injection upon Writes) for SEC errors or 0x7 (Double-Fault Injection upon Writes) for DED errors.
- FLCTRL.FLTEN = 0x1 (Fault Injection is enabled)

4.1.1 Opening the NVMCTRL ECC with Fault Injection Testing Example

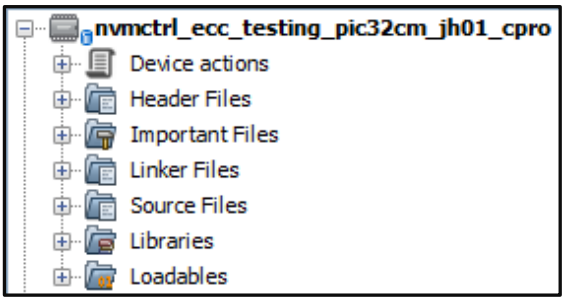
The code example is available in the MPLAB Harmony v3 framework repository.

Note: The user must ensure the MPLAB Harmony v3 Framework is installed. This folder can be downloaded on a computer from the MPLAB Code Configurator (MCC) Content Manager. If MPLAB Harmony v3 framework is not installed while opening an MCC project, the required packages will be downloaded. The default folder path to download is C:/Users/HarmonyFramework.

Follow these steps to open the software project:

1. Open MPLAB X IDE.
2. Select *Toolbar* > *File* > *Open Project (Ctrl* + *Shift* + *O)*.
3. Open the `nvmctrl_ecc_testing` example:
   a. Navigate to *C:\<Harmony3_Framework_Path>\csp_apps_pic32cm_jh00_jh01\apps\nvmctrl*.
   b. Once opened, the `nvmctrl_ecc_testing` example will be available in the MPLAB X IDE Project Tree:

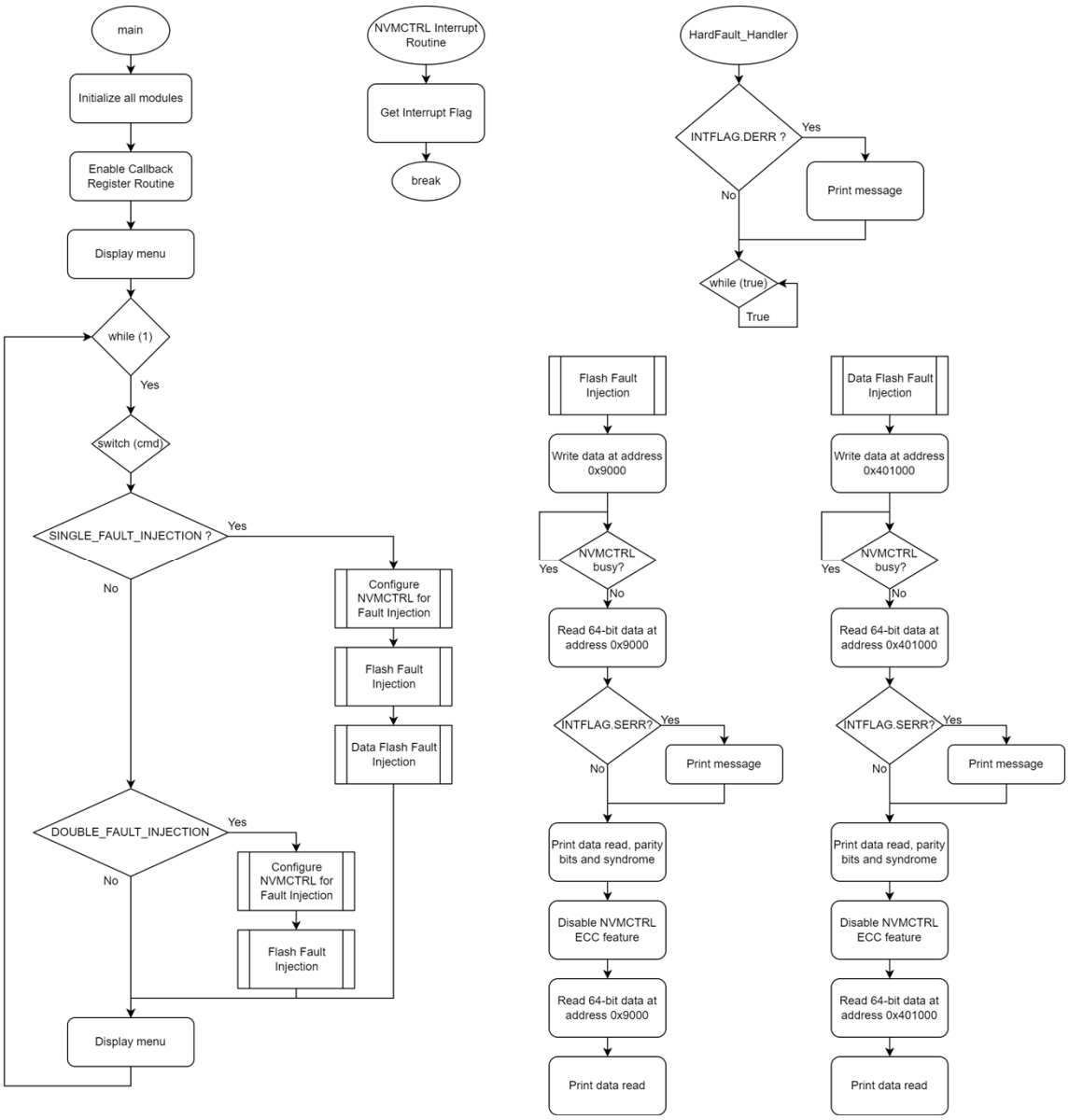
Figure 4-1. NVMCTRL ECC with Fault Injection Testing: Project Tree



4.1.2 Application Flowcharts

The NVMCTRL ECC with Fault Injection Testing example shows the configuration required for the NVMCTRL to perform the ECC with Fault Injection to test the SEC or DED errors process. The application starts with the NVMCTRL.ECCCTRL register configuration then enables the SERR and DERR interrupts and displays a menu to choose whether a single or double fault will be injected as shown in the figure below:

Figure 4-2. NVMCTRL ECC with Fault Injection Testing: Application Flowchart



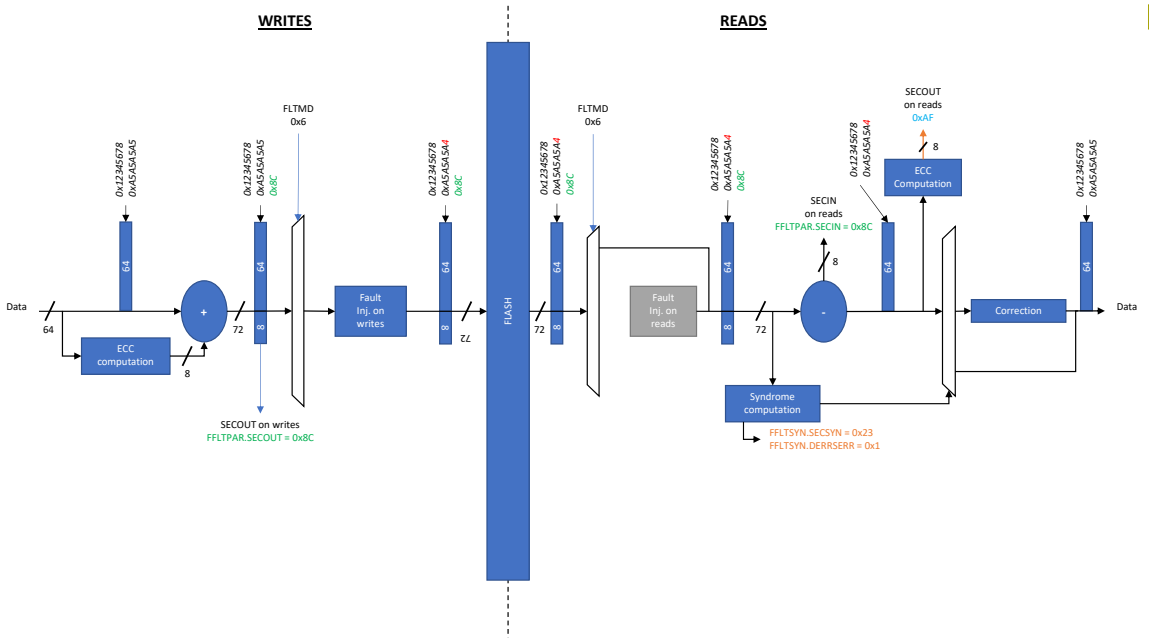
When `SINGLE_FAULT_INJECTION` is selected, the application configures the NVMCTRL fault injection registers to perform single-fault injection on the double-word *0x12345678A5A5A5A5* written in the Flash and Data Flash memories. Once configured, the application writes the data in the Flash at address 0x9000. It reads back the data that is corrected on-the-fly and displayed on terminal, along with the SECIN, SECOUT, and syndrome values. Then, the ECC feature is disabled, the data is read again and displayed on the console. Because the ECC feature is disabled, the data is no more corrected on-the-fly. The same process is done for the Data Flash at address 0x401000 onwards. A message is displayed on console each time when a SERR interrupt is triggered. The following figure shows the message displayed on the console when `SINGLE_FAULT_INJECTION` is selected

Figure 4-3. NVMCTRL ECC with Fault Injection: Single-Fault Injection Display

```
************************************************
*** NVMCTRL ECC with Fault Injection Testing ***
************************************************

Select the test to launch:
a) Single Fault Injection on Writes
b) Double Fault Injection on Writes
Enter your choice:

Single Fault Injection for Flash memory on writes
Single Fault Detected !
Value Read at address 0x9000: 0xA5A5A5A5 is corrected on the fly
Value Read at address 0x9004: 0x12345678 is corrected on the fly
The computed SECIN is 0x8C
The new computed SECOUT is 0xAF
The syndrome is 0x23

Physical value at address 0x9000: 0xA5A5A5A4
Physical value at address 0x9004: 0x12345678

Single Fault Injection for Data Flash memory on writes
Single Fault Detected !
Value Read at address 0x401000: 0xA5A5A5A5 is corrected on the fly
Value Read at address 0x401004: 0x12345678 is corrected on the fly
The computed SECIN is 0x8C
The new computed SECOUT is 0xAF
The syndrome is 0x23

Physical value at address 0x401000: 0xA5A5A5A4
Physical value at address 0x401004: 0x12345678

Please RESET the board as ECC feature cannot be re-enabled.
```

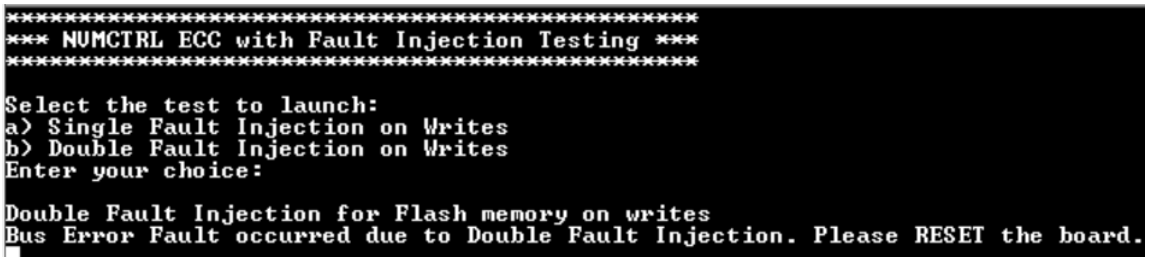
The following figure illustrates the internal operations performed when `SINGLE_FAULT_INJECTION` is selected:

Figure 4-4. NVMCTRL ECC with Fault Injection on Writes: Single-Fault Injection Logic Example



When DOUBLE_FAULT_INJECTION is selected, the application configures the NVMCTRL fault injection registers to perform double fault injection on the Flash memory. Once configured, the application writes the data in the Flash at address 0x9000 and reads back the data. Due to double error detection, the application jumps in HardFault_Handler and prints the message to Reset the board. The following figure shows the message displayed on the console when DOUBLE_FAULT_INJECTION is selected:

Figure 4-5. NVMCTRL ECC with Fault Injection on Writes: Double-Fault Injection Display



4.2 SRAM MBIST and MCRAMC ECC with Fault Injection Testing

To illustrate the SMBIST and SRAM ECC logic, an example is provided along with this document. The goal is to highlight the SMBIST process and ECC steps for the SRAM.

The following configuration is applied for the SRAM MBIST and ECC with Fault Injection Testing example:

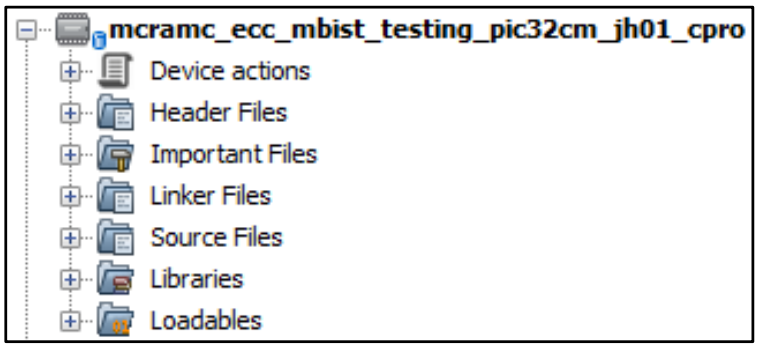
- PAC Write Protection for SMBIST is disabled
- SMBIST test is launched on the SRAM's partitions one and two:
  - SMBIST: CTRL.SMBISTP1 = 0x1
  - SMBIST: CTRL.SMBISTP2 = 0x1
- SERR and DERR interrupts are enabled (INTENSET.SERREN = INTENSET.DERREN = 0x1)
- FLTPTR.FLT1PTR = 0x0 and FLTPTR.FLT2PTR = 0x1
- FFLTADR.FLTADR = 0x4000 for SRAM memory write operation (fault injection will be applied to address 0x20004000)
- FLTCTRL.FLTMD = 0x1 (Single-Fault Injection) for SEC errors or 0x2 (Double-Fault Injection) for DED errors
- FLCTRL.FLTEN = 0x1 (Fault Injection is enabled)

4.2.1 Opening the SRAM MBIST and ECC with Fault Injection Testing Example

The code example is available in the MPLAB Harmony v3 framework repository. Follow these steps to open the software project:

1. Open MPLAB X IDE.
2. From the Toolbar, select *File* > *Open Project (Ctrl + Shift + O)*.
3. Open the `sram_ecc_mbist_testing` example:
   a. Navigate and open: *C:\<Harmony3_Framework_Path>\csp_apps_pic32cm_jh00_jh01\apps\sram*.
   b. The `sram_ecc_mbist_testing` example will be displayed under the MPLAB X IDE project tree:

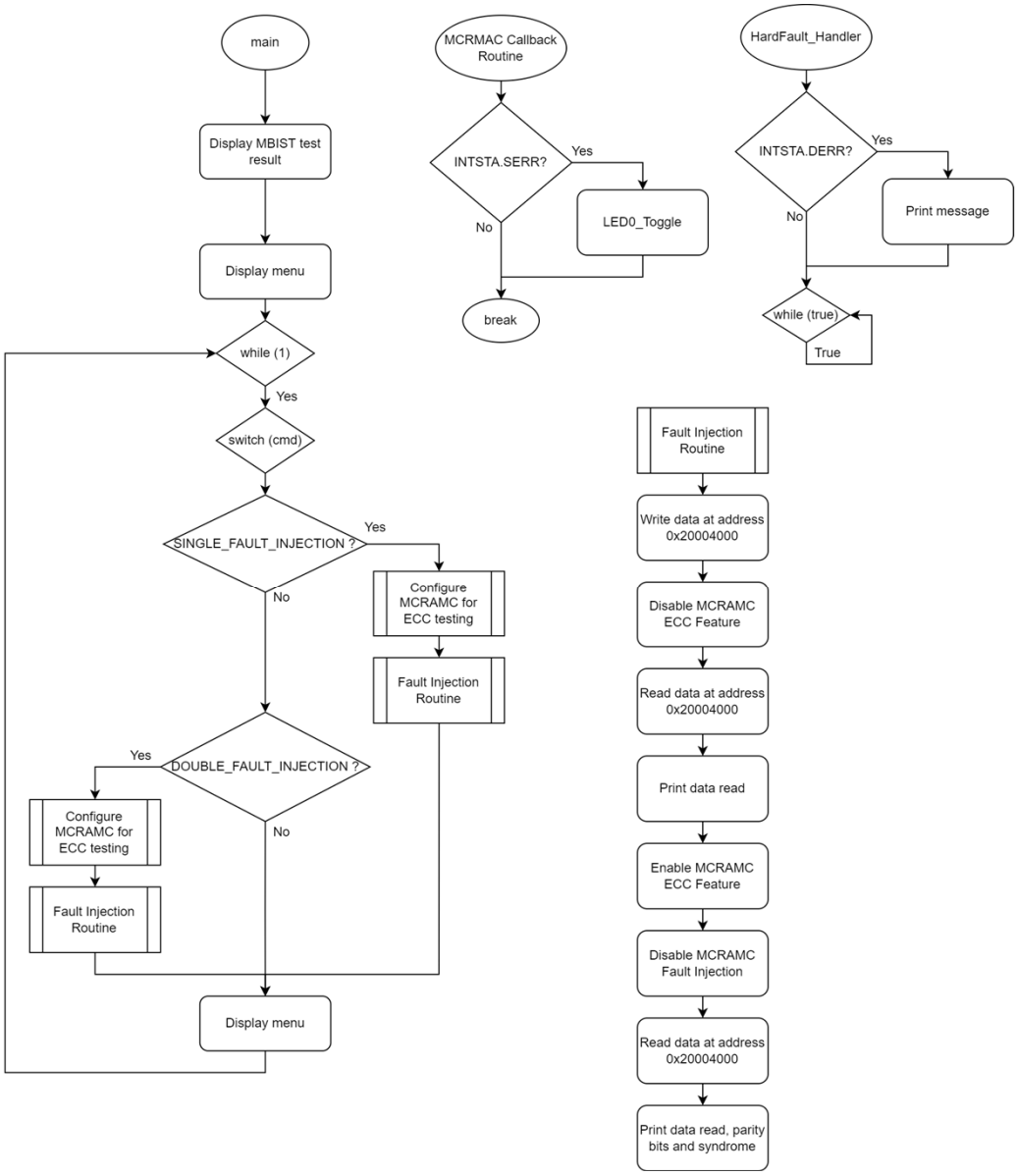
Figure 4-6. MCRAMC ECC with Fault Injection on Writes: Project Tree



4.2.2 Application Flowcharts

This application shows the ECC with Fault Injection on the MCRAMC for testing, and the SMBIST process to validate the SRAM. The application will start by displaying the SMBIST test result obtained after the SMBIST performed during device startup (in Reset Handler). The application enables the SERREN, DERREN interrupts, and displays the menu to choose whether a single-fault or double-fault will be injected in the configured SRAM address. The application flowchart is illustrated below:

Figure 4-7. SMBIST and MCRAMC ECC with Fault Injection on Writes: Application Flowchart



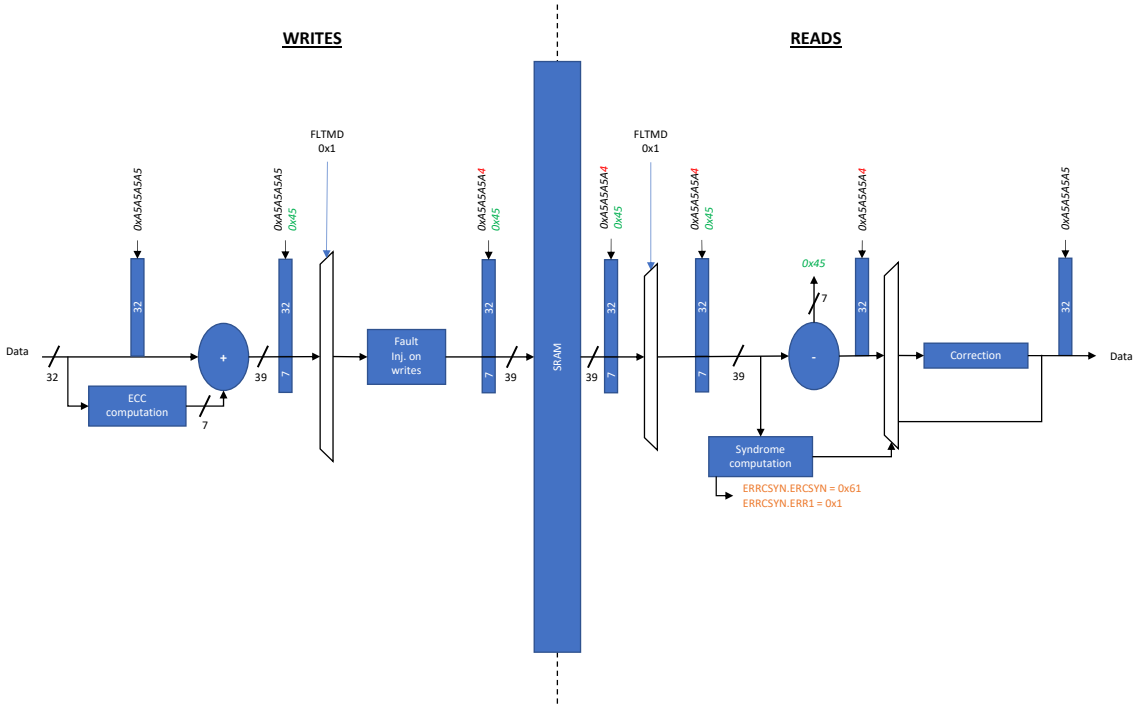
When SINGLE_FAULT_INJECTION is selected, the application configures the MCRAMC fault injection registers to perform a single-fault injection on the word *0xA5A5A5A*. Once configured, the application writes the data in the SRAM at address 0x20004000, then the MCRAMC ECC feature is disabled. The application reads the data at address 0x20004000, which is not corrected on-the-fly and is printed afterward. The ECC feature is then enabled, and fault injection is disabled (for not injecting the error again when the corrected word is automatically written back in memory). The application again reads the data at address 0x20004000, which is this time corrected on the fly (and written back in memory) and printed along with the parity bits and syndrome. The LED0 toggles each time a SERR interrupt is triggered. The following figure shows the message displayed on the console when SINGLE_FAULT_INJECTION is selected:

Figure 4-8. SMBIST and MCRAMC ECC with Fault Injection: Single-Fault Injection Display

```
*********************************************
*** MCRAMC ECC with Fault Injection Testing ***
*********************************************

MBIST Test on SRAM succeeded.

Select the test to launch:
a) Single Fault Injection
b) Double Fault Injection
Enter your choice:

Single Fault Injection for SRAM memory
Injecting Fault at address 0x20004000
ECC Decoding Disabled ..
Value Read from SRAM at address 0x20004000 is 0xA5A5A5A4

ECC Decoding enabled ..
Value Read from SRAM at address 0x20004000 is 0xA5A5A5A5, corrected on the fly
The parity bits are 0x45
The syndrome is 0x61
```

The following figure illustrates the internal operations performed when SINGLE_DOUBLE_INJECTION is selected:

Figure 4-9. SMBIST and MCRAMC ECC with Fault Injection on Writes: Single-Fault Injection Logic Example



When DOUBLE_FAULT_INJECTION is selected, the application configures the MCRAMC fault injection registers to perform double-fault injection on the SRAM memory. The following figure shows the message displayed on the console when DOUBLE_FAULT_INJECTION is selected:

Figure 4-10. SMBIST and MCRAMC ECC with Fault Injection on Writes: Double-Fault Injection Display

Note: Due to double-fault injection, a bus fault will occur when the ECC feature is enabled and the data is read, leading the application to jump in the HardFault_Handler and then display a message on the console. Then press the RESET button.

5. Conclusion

The ECC functional safety feature for the NVMCTRL and MCRAMC corrects single errors seamlessly and detects double errors, which can occur in an environment under external constraints, such as gamma rays and noise.

For additional safety, the PIC32CM JH00/JH01 family of devices also embeds the MBIST on the SRAM to test potential SRAM defects. This document highlights the logic behind these features and how to test them using Fault Injection (or Fault Capture) for the ECC in NVMCTRL and MCRAMC related memories, and the SMBIST for SRAM through code examples.

6. References

The following documents are used as reference. For additional information, visit the Microchip Website.

- PIC32CM JH00/JH01 Family Data Sheet *(DS60001632)*
- PIC32CM JH00/JH01 Family Silicon Errata and Data Sheet Clarifications *(DS80001000)*

7. Revision History

Revision B - 06/2022
The following updates were made in this revision:

- Removed the NDA Confidential verbiage, and made minor formatting updates
- Updated the FLTPTR.FLT1PTR and FLTPTR.FLT2PTR values in SRAM MBIST and MCRAMC ECC with Fault Injection Testing
- Updated verbiage for disabling the ECC feature in Application Flowcharts
- Updated the following figures in the document:
  - Figure 2-1
  - Figure 2-2
  - Figure 2-3
  - Figure 4-2
  - Figure 4-3
  - Figure 4-4
  - Figure 4-6
  - Figure 4-7
  - Figure 4-8
  - Figure 4-9
  - Figure 4-10

Revision A - 04/2022
This is the original release of the document.

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable". Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

Trademarks

ISBN: 978-1-6683-0696-3

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820