AN6284

Application note

Avoiding false positives during online LBIST execution for SPC58xHx

Introduction

This document details how to run the LBIST for SPC58xHx in online mode using software and properly wait for its execution.

Users can employ the SPC58xHx MCU for safety-critical applications with ASIL-D requirements. The SPC58xHx safety concept mandates executing the LBIST at least once per trip time to detect potential multipoint failures in the device.

This document focuses on the SPC58xHx, but the described strategy for executing online LBIST applies to all SPC58 devices with LBIST online (SPC58xEx, SPC58xNx, and SPC58xGx).

1 LBIST in online mode

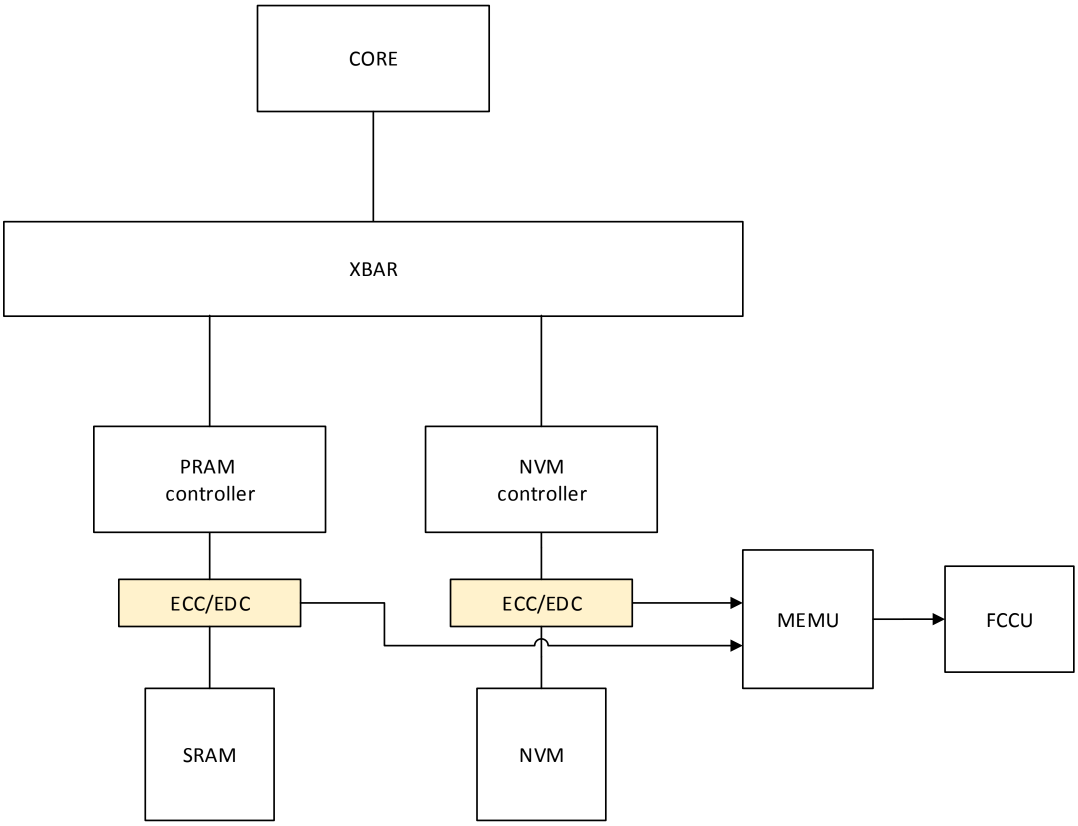
The LBIST checks the integrity of several modules in the SPC58xHx, including the ECC/EDC logic connected to both volatile and nonvolatile memories.

The Figure 1 shows a simplified connection schema between one core and these memories. Whenever a core or another initiator accesses these memories, data passes through the ECC/EDC logic check, ensuring the integrity of the read data.

Integrators must note that during the LBIST execution, no core should access the memories protected by the ECC/EDC logic. If this happens, the ECC/EDC may signal an unwanted ECC error fault (false positive). This could cause the system to react with a core exception and an FCCU response, depending on the FCCU configuration. Under this circumstance, users may observe fake" UCE entries in MEMU.

Consequently, users must forbid any access to these memories while the LBIST is running in online mode.

Figure 1. Simplified schema of the connection between core and memories



1.1 Steps to safely run LBIST in online mode

There are multiple options to prevent a core from accessing memories shown in the Figure 1. The most reliable method is to halt the core's execution using a wait instruction until an interrupt is received. This instruction puts the core in a low-power state, waiting for an interrupt, or an external event.

Hereafter the recommended steps to run the LBIST in online mode:

1. Configure the LBIST via the STCU registers
2. Set the STCU to trigger a global reset when the LBIST execution ends
3. Stop software execution of all cores
4. Disable interrupts of all cores
5. Start the LBIST execution
6. Issue a wait instruction

After the last step, the core stops executing any operations and waits for an interrupt or external event. The reset triggered by the STCU upon LBIST completion is the external event that wakes up the core.

Interrupts from other peripherals may occur during the LBIST execution. To prevent these interrupts from waking up the core prematurely, disable all interrupts before starting the LBIST (step 4).

1.2 Reference code

The code below shows a reference code that implements the steps highlighted in the previous section.

```
void run_LBIST(void) {

    STCU2.SKC.R = 0x753F924E; //key1
    STCU2.SKC.R = 0x8AC06DB1; //key2

    /*start LBIST */
    STCU2.CFG.R = 0x005A0001;
    STCU2.WDG.R = 0xFFFFFFF0;

    STCU2.LB[0].CTRL.R = 0xFF014400;
    STCU2.LB[0].PCS.R = 0x1388;

    STCU2.LB[0].MISRELSW.R = 0x21332EBB;
    STCU2.LB[0].MISREHSW.R = 0x2C349496;

    /* The STCU trigger a functional reset at the end of LBIST run */
    STCU2.LBRMSW.R = 0x00000001;

    StopOtherCores();

    DisableAllInterrupts();

    /* Trigger the LBIST execution */
    STCU2.RUNSW.R = 0x00000001;

    asm(“wait”);

    /* The core never executes the instruction below due to the wait instruction */
    while (STCU2.RUNSW.B.RUNSW == 1);
}
```

2 Safety consideration

False positives may occur when an initiator accesses the memories in the Figure 1 while the LBIST runs online.

This behavior depends on the timing relationship between memory access and ECC/EDC logic testing. A small clock skew affecting the memory access path and ECC/EDC testing logic can cause false positives. As a result, this behavior may appear more frequently in some samples than others.

Estimating the probability of occurrence is not feasible because it depends on multiple parameters, including the application software running in the core and the accuracy of external and internal oscillators.

It is worth noting that this behavior does not affect the functional safety of a fail-safe system. The ECC/EDC logic still detects random failures with the diagnostic coverage estimated in the FMEDA. However, this behavior does impact system availability. A false positive can move the system to a safe state, stopping the provision of safety services.

3 Conclusion

If the software does not run the online LBIST correctly, the system's ECC/EDC logic may trigger one or more ECC error false positives. This detection can cause the system to respond with an exception or an FCCU reaction.

To prevent this event, the software should prevent a core from accessing memories shown in the Figure 1 after starting the LBIST execution. The most reliable method is to halt the core's execution using a wait instruction.

This document includes reference code to run the LBIST in online mode, avoiding false positives from the ECC/EDC logic.

Revision history

Table 1. Document revision history

Date	Revision	Changes
26-Mar-2025	1	Initial release.

Contents