MTU_MBIST_1
for KIT_AURIX_TC397_TFT
Memory Built-In Test via MTU

AURIX™ TC3xx Microcontroller Training
V1.0.1

Scope of work

The MBIST is used to assess the state of the memory.

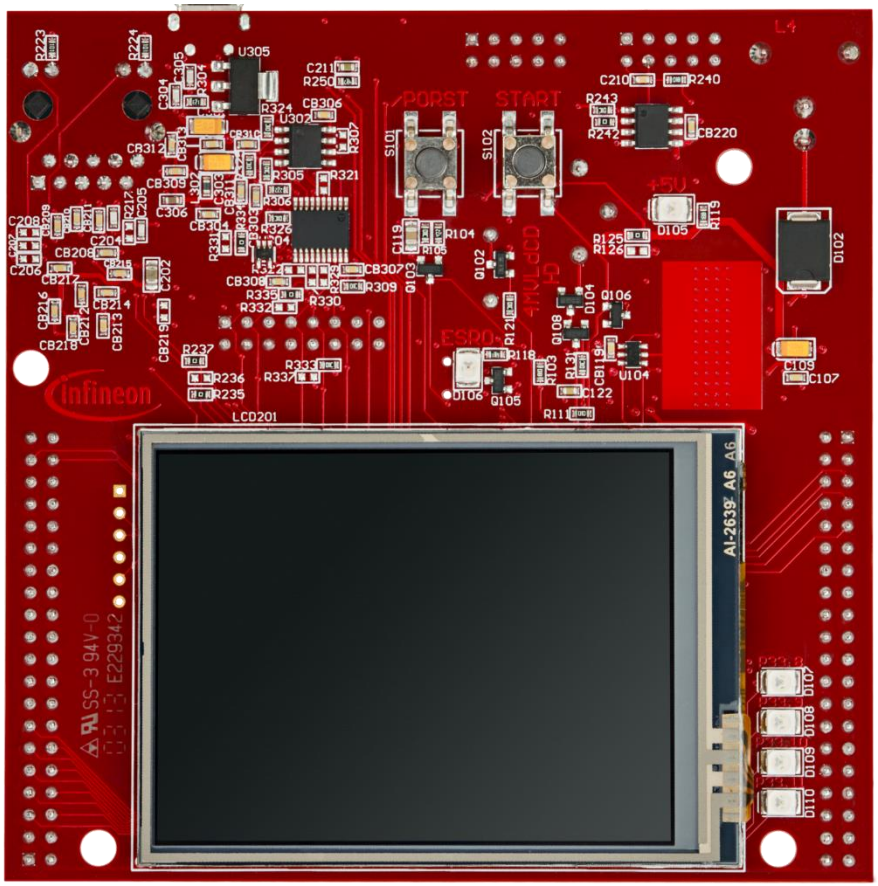
In this training the Memory Test Unit (MTU) is used to initialize and clear the content of an SRAM memory including its ECC code. Additionally, the Non-Destructive Test (NDT) is performed by Memory Built-in-Self-Test (MBIST) to verify the content of the same SRAM memory.
The SRAM of the DMA (DMARAM) is selected to be tested in this example.
In order to test the faulty scenario, this training provides the possibility to inject a single bit error in the memory.
The LEDs on the board are used to signal the correct or faulty behavior of the MBIST.

Introduction

- The Memory Test Unit (MTU) controls and monitors the test, initialization and data integrity checking functions of the various internal memories in the AURIX™ TC3xx family of microcontrollers.
- Each SRAM is provided with digital logic surrounding it, known as SRAM Support Hardware (SSH).
- Each SSH block controls one or more internal memories, providing an unified interface for the control of ECC (Error Correction) and BIST (Built-in-Self-Test).
- The SSH provides direct access to the memories, without involving the CPU.
- The Memory Built-in-Self-Test (MBIST) is an MTU feature, which enables the verification of the integrity of internal SRAMs.

Hardware setup

This code example has been developed for the board KIT_A2G_TC397_5V_TFT.



Implementation

MBIST test implementation:
Execution of the MBIST test is ensured by the ***test_MTU_MBIST()*** function, containing the following steps:

1. MTU module is enabled: ***IfxMtu_enableModule()***
2. A check for UnCorrectable Error (UCE) alarm is performed: ***get_MTU_MBIST_Errors()***
   - If an error is reported*, the UCE alarm status is cleared: ***clear_MTU_MBIST_Errors()***
3. The SRAM to be tested is initialized using MTU: ***IfxMtu_clearSram()***
4. Optionally, an error is injected before test execution (Please refer to next slide)
5. The Non-Destructive Test (NDT) is triggered: ***IfxMtu_runNonDestructiveInversionTest()***
6. After test completion, the ***IfxMtu_runNonDestructiveInversionTest()*** function returns whether the RAM content is correct or an error is detected

› The functions used to enable the MTU, clear the SRAM and run the NDT are provided by the iLLD header ***IfxMtu.h***, while the functions used to get and clear the UCE alarm status can be found in ***MTU_MBIST.c***.

Note: In this training, the DMARAM is tested by calling ***test_MTU_MBIST(IfxMtu_MbistSel_dma)*** in the main function. It is possible to test any other SRAM memory ***X*** by calling ***test_MTU_MBIST(IfxMtu_MbistSel_X)***.

*: After any System Reset: For each and every SSH in the system, the UCE alarm status in the SMU, the ECCD.UCERR (Consequently also SERR) and the FAULTSTS.OPERR[0] are set.

Implementation

Correctable Error Injection:
The error injection option can be enabled by setting the global variable ***g_errorInjection*** to ***TRUE*** in ***MBIST_MTU.c*** file.

Implementation steps:

1. Safety Endinit protection is cleared: ***IfxScuWdt_clearSafetyEndinit()***
2. MBIST controller is enabled: ***IfxMtu_enableMbistShell()***
3. Wait for the end of SRAM initialization, in case it is an auto-initialization memory: ***IfxMtu_isAutoInitRunning()***
4. One memory address (e.g. 0x1F) is read: ***IfxMtu_readSramAddress()***
5. Only one bit should be modified to inject a correctable error. DMARAM is **SECDED** (Single Error Correction, Double Error Detection)
6. The updated memory is written back to the same address (0x1F): ***IfxMtu_writeSramAddress()***
7. MBIST controller is disabled: ***IfxMtu_disableMbistShell()***
8. Safety Endinit protection is restored: ***IfxScuWdt_setSafetyEndinit()***

The functions above are provided by the iLLD headers ***IfxMtu.h*** and ***IfxScuWdt.h***.

Implementation

LED initialization and control:

- Two LEDs are used in this training to report the execution of the test
  - ***LED_PASS*** : LED driven by port 13 pin 0
  - ***LED_FAIL*** : LED driven by port 13 pin 1

- Ports configuration in the right mode to control LEDs:
  - ***IfxPort_setPinMode()***

- LEDs control (LEDs are low-level active)
  - Switch On: ***IfxPort_setPinLow()***
  - Switch Off: ***IfxPort_setPinHigh()***

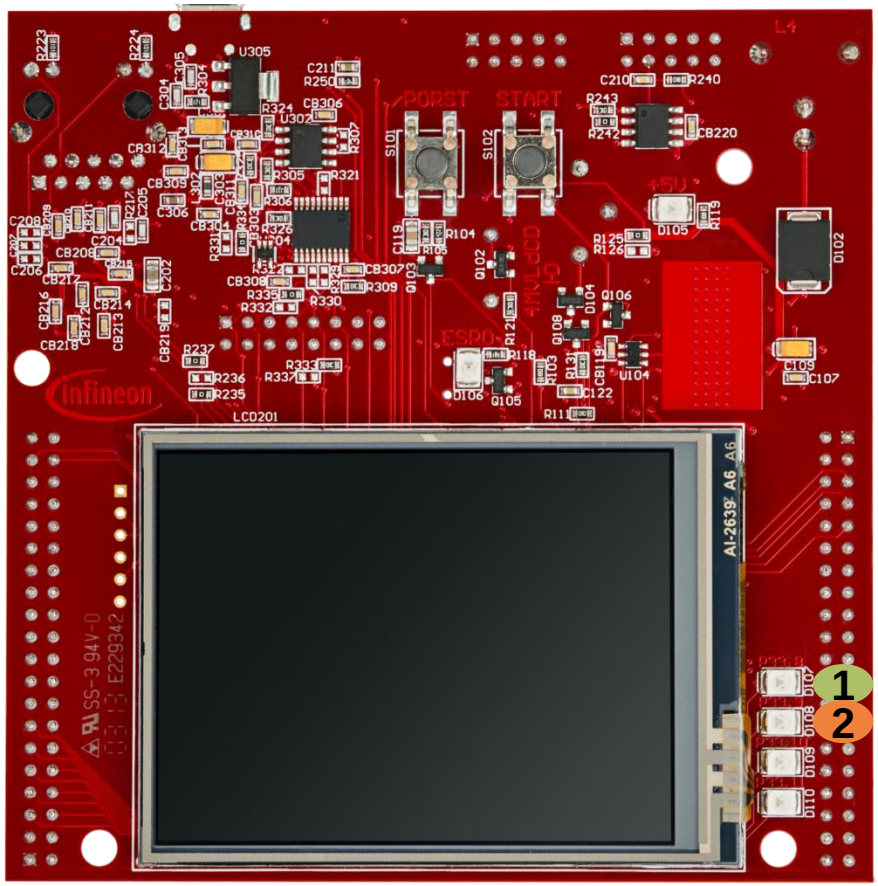
- The functions above are provided by the iLLD header ***IfxPort.h***

Run and Test

After code compilation and flashing the device, check the behavior of ***LED_PASS*** (1) and ***LED_FAIL*** (2):

MBIST of DMARAM		1 **LED PASS**	2 **LED FAIL**
No Error injected	No Error detected		
	Error detected		
Error injected	No Error detected		
	Wrong Error detected		
	Correct Error detected		

LED On LED Off



References

› AURIX™ Development Studio is available online:
› https://www.infineon.com/aurixdevelopmentstudio
› Use the *„Import...“* function to get access to more code examples.

› More code examples can be found on the GIT repository:
› https://github.com/Infineon/AURIX_code_examples

› For additional trainings, visit our webpage:
› https://www.infineon.com/aurix-expert-training

› For questions and support, use the AURIX™ Forum:
› https://www.infineonforums.com/forums/13-Aurix-Forum

Revision history

Revision	Description of change
V1.0.1	Update of version to be in line with the code example’s version
V1.0.0	Initial version

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2022-12
Published by
Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?
Email: erratum@infineon.com

Document reference
MTU_MBIST_1_KIT_TC397_TFT

IMPORTANT NOTICE
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie") .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

WARNINGS
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.