FTF FREESCALE TECHNOLOGY FORUM 2015

Hands-On Workshop: Configuring and Optimizing Built-In Self Tests with MPC5777M and MPC57XX

FTF-ACC-F1258

Andrew Turner | Automotive MCU Applications Engineer

J U N E . 2 0 1 5

freescale™

External Use

Session Introduction

- Built-In Self-Testing (BIST) is a mechanism provided on Freescale MPC57XX devices to detect the accumulation of latent faults, a requirement of the ISO26262 standard
- This session demonstrates how to configure the BIST tests on MPC5777M using the Self-Test Control Unit
- The safety coverage aspects of the BIST tests will be briefly explained
- This session will work though examples of BIST execution during runtime (on-line) and boot (off-line)
- The examples will detail how to configure Memory BIST (MBIST) and digital Logic BIST (LBIST)

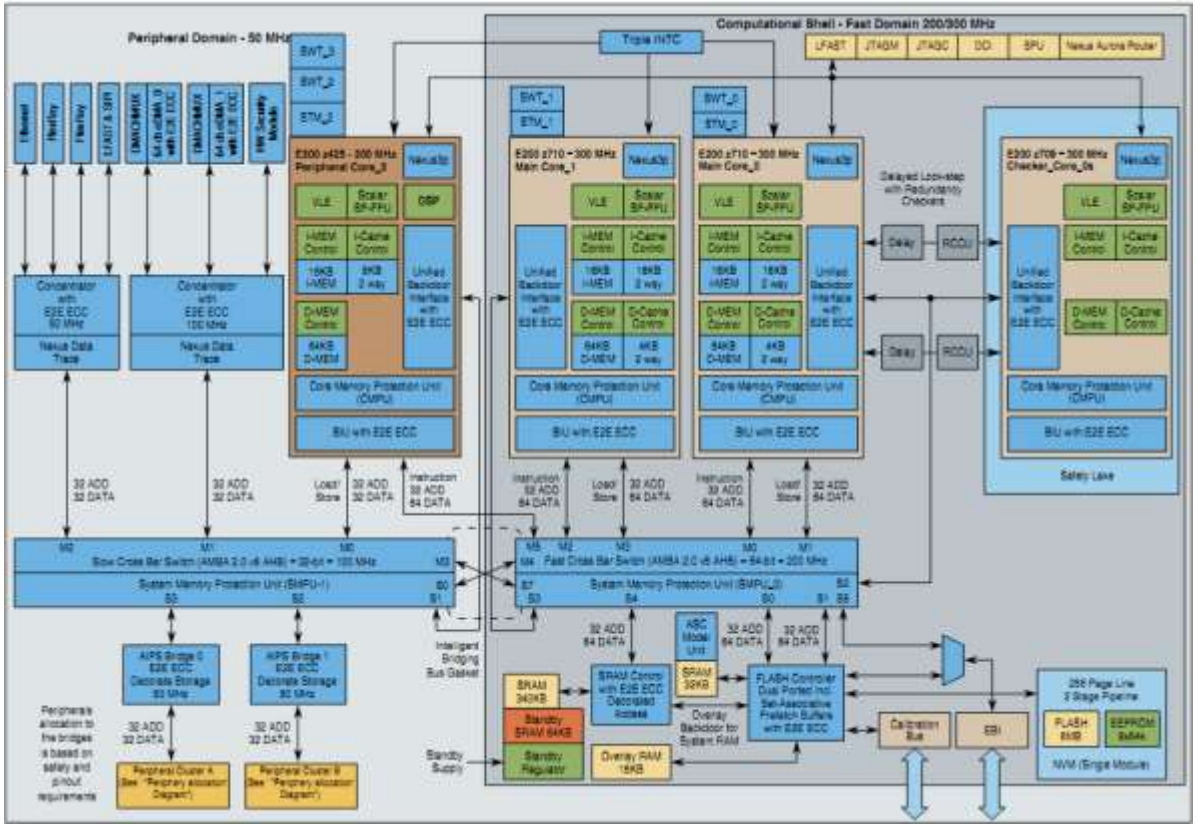
Agenda

- MPC5777M Overview and Architecture
- BIST Overview
- BIST and Functional Safety
- STCU2 Description
- Lab Examples and Tools Introduction
- Walk-Through of Examples
- Summary

MPC5777M Overview

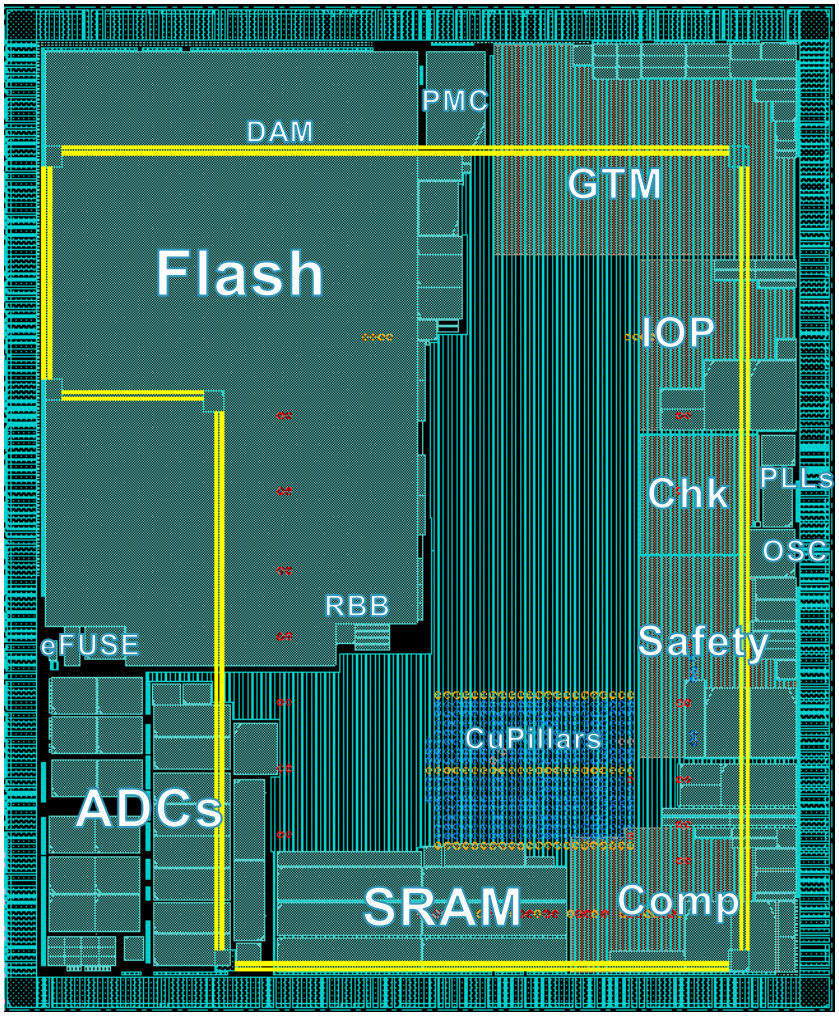
Key Functional Characteristics

- Two independent 300 MHz Power Architecture z7 computational cores
  - Single 300 MHz Power Architecture z7 core in delayed lockstep for ASIL-D safety
- Single I/O 200 MHz Power Architecture z4 core
- Hardware Security Module (HSM)
- Dual PLL with freq modulation
- eDMA controller – 128 channels
- 8M Flash with ECC
- 596k total SRAM with ECC
  - 404k of system RAM (incls. 64k standby)
  - 192k of tightly coupled data RAM
- 10 ΣΔ Converters 16-bit
- 12 SAR converters 12-bit
- Ethernet (MII/RMII)
- DSPI – 8 channels (3 supporting µSec ch.)
- LINFlex - 6 channels (3 supporting µSec ch.)
- MCAN / TTCAN – 4x modules/1x module
- 2 x PSI5 Controllers
- GTM 104– Generic Timer Module
- External Bus Interface (EBI)
- 2 x FlexRay controllers
- **Built In Self Test capability**



BIST Overview

Overview of Built-In Self-Test



- The term Built-In Self-Test (BIST) is used to describe the on-chip hardware mechanisms that can be used to detect latent faults within the MCU
- The BIST allows the MCU to conduct periodic self-tests to identify faults.
- The results of these self-tests can then be used by the MCU to handle the faults and ensure that the device remains in a safe state
- On MPC5777M the BIST provides the ability to meet the latent fault detection requirements defined by the ISO26262 ASIL-D functional safety standard

MBIST and LBIST Overview

- Two different types of BIST are implemented on the MPC5777M: Memory Built-In Self-Test (MBIST) for memory and Logic Built-In Self-Test (LBIST) for digital logic. MBIST is implemented for each of the RAM and peripheral memories on the MPC5777M.
- For MBIST testing purposes each of the memories is segmented into individual MBIST partitions. Each memory is broken down into multiple partitions providing flexibility to test selected address ranges only.
- LBIST tests operate on the digital logic of the device and use scan test techniques to provide high coverage defect detection. The logic is divided up into multiple partitions, with each partition containing multiple user recognizable logic modules (CPU, XBAR, MCAN etc).

LBIST & MBIST for functional safety

- **LBIST and MBIST** are **safety mechanisms** to detect permanent die faults during start-up (at time 0) or within the Latent-FTTI (at shutdown for example).

- The failure rate of permanent die faults is very low
  - Much less than 10 FIT for the MCU according to IEC TR 62380
  - Whereas transient faults are much greater than 1000 FIT for the MCU.
  - The majority of MCU safety mechanisms are to address transient faults during application runtime – e2eECC, Lock-Step

- The LBIST and MBIST tests on MPC5777M do not distinguish between Safe Faults, Single-Point Faults or Latent Faults, they focus on finding faults in the digital logic or SRAM logic, no matter what their categorization.

- From ISO 26262 FMEDA perspective, the LBIST is used to detect **Latent Faults**.

- LBIST contributes to the Latent Fault Metric calculated in the FMEDA and is required by **ISO 26262 to be >90%** for the MCU **to achieve ASIL D** in the context of the safety system.

Application Context for LBIST & MBIST

- Expected that each product will have 2 standard configurations for LBIST & MBIST
  - **Start-up** – test as much as possible within the start-up time constraints – OFF-LINE
  - **Shutdown / diagnostic** – maximize test coverage, no time constraints – ON-LINE

- **Start-up**
  - During start-up the execution time is critical.
  - Ideally LBIST & MBIST execution time needs to be in the range of 10 ms – 30 ms dependent on application

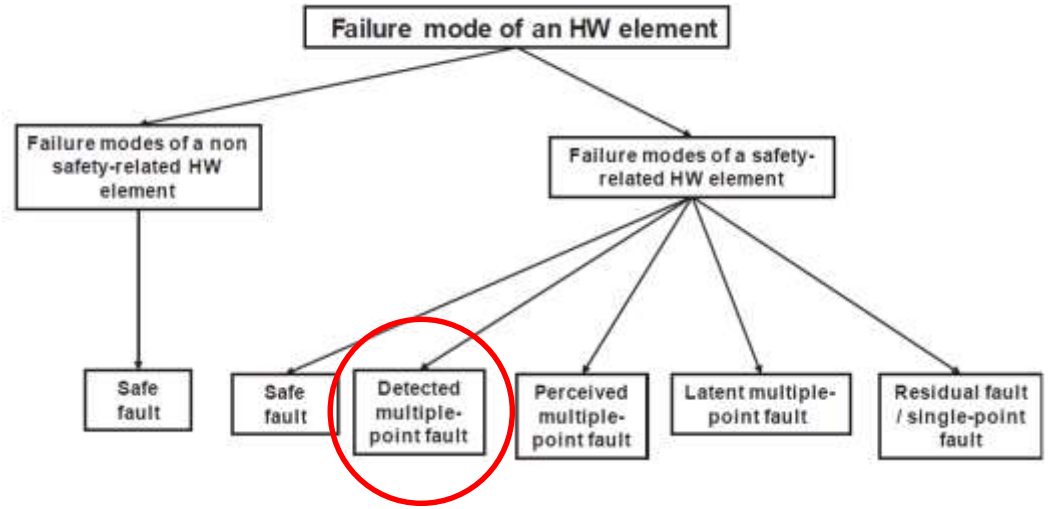
  - In order to achieve the time of 10 – 30 ms, power and test coverage are tailored
    - More power consumed, test time is shorter
    - More test coverage, test time is longer
  - Power budget is part of MCU spec, so application independent and can be optimized by Freescale through standard LBIST & MBIST configuration
  - Test coverage is dependent on ASIL, ranging from 60 – 90%

ISO 26262 – Latent Fault Metric (LFM)

- The requirements from ISO 26262-5 for Latent Fault Metric are as shown in the table below
- Freescale use this as guideline for LBIST / MBIST target fault coverage
- The FMEDA uses the LBIST and MBIST as Latent Fault Safety Mechanisms in order to quantify if the ISO 26262 target metrics are met for the MCU

Table 5 — Possible source for the derivation of the target "latent-fault metric" value

	ASIL B	ASIL C	ASIL D
Latent-fault metric	≥60 %	≥80 %	≥90 %



LBIST & MBIST Latent Fault Coverage

ISO 26262:2011 5 Annex B figure B.1

LBIST Standard Configurations for MPC57XX Products

- **Start-up**
  - ASIL C & D: If time permits, 90% stuck-at coverage, otherwise reduce test to most critical partitions only as defined by safety concept & module classification (maintain 90% stuck-at coverage on partitions tested)
  - ASIL B: 60% stuck-at coverage

- **Shutdown / diagnostic**
  - ASIL C & D: 90% stuck-at coverage & 60% transition-delay coverage
  - ASIL B: 60% stuck-at coverage

- *Note: Not all ASIL B products require LBIST & MBIST*

MBIST Standard Configurations for MPC57XX Products

- Three types of MBIST March tests are available, with different coverage and execution time
  - *Auto Test* (18N)
  - *Reduced RunBIST*
  - *Full RunBIST*
- Start-up
  - ASIL C & D: If time permits, Full RunBIST, otherwise Auto Test (18N)
  - ASIL B: Auto Test (18N)
- Shutdown / diagnostic
  - ASIL C & D: Full RunBIST
  - ASIL B: Auto Test (18N)

		Mode bit setting		
STCU_CFG.MBU		0	0	1
STCU_CFG.PMOSEN		1	0	X
		Diagnostic Coverage of MBIST mode		
Fault Model	**MBU?**	**RunBIST Mode (full test)**	**Reduced RunBIST Mode**	**Auto Test Mode (18N algo)**
Stuck at faults		high	high	high
Stuck open faults		high	high	Low
Transition faults		high	high	high
Write destructive faults		high	high	Low
Read destructive fault		high	high	high
Deceptive read destructive fault		high	high	Low
Read deceptive coupling fault		high	high	Low
Deceptive read disturb faults triggered by several read		high	high	Low
Data retention fault		high	high	Low
Coupling faults		high	high	Low
Realistic linked coupling faults		high	high	Low
Weak pull-up PMOS transistor of bitcell		high	Low	Low
SNPSF (static)		high	high	Partial
PNPSF (passive) & ANPSF (active) partially covered		Partial	Partial (*)	Partial (**)
Address Decoder faults	MBU	high	high	high
Address decoder Activation	MBU	high	high	high
Address decoder Deactivation	MBU	high	Low	Low
Slow Sense amplifier faults	MBU	high	high	high
Slow Write drivers faults	MBU	high	high	high
Slow precharge circuit faults	MBU	high	high	high
Bit line imbalance faults	MBU	high	high	high
Coupling Faults between Global bitlines with local bitlines		high	high	Low
CSN and Mask pins test		high	high	Low
IO coupling faults		high	high	Low

Application Usage

MPC57XX Start-up/Off-Line BIST implementation

- Off-line, or start-up BISTS are configured to execute every time the MCU boots or performs a destructive reset.
- This procedure is performed while the MCU is powered and held in reset.
- The configuration at start up is accomplished by user programmable Device Configuration Format (DCF) Records that are stored in the one-time programmable Flash UTEST memory
- If programmed to do so, the DCF records automatically configure the self test parameters and the test is executed

On-Line (Shut-down/Diagnostic) BIST Overview

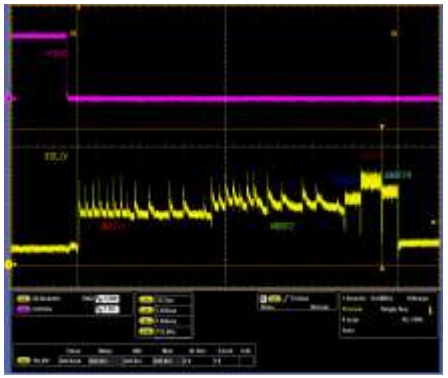
- In addition to being able to run at start-up under control of the STCU2, the MCU allows software to write to the STCU2 during runtime to configure and trigger the execution of MBIST or LBIST. This is known as online testing

- Online testing is mainly intended for a full BIST of the MCU, typically performed prior to shutdown of the ECU, when execution time is not as critical. The online mode can also be used for failure diagnostics and quality control within a manufacturing environment

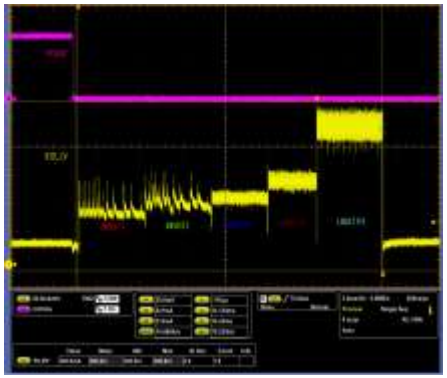
Example Configurations for MPC5777M

Self Test Level	Application Mode	Clock	STCU2 Config.	Allocated Time	LBIST coverage	MBIST Algo,
Ultra-Short	KEY ON (STARTUP)	PLL0 50MHz	Loaded from Flash by SSCM	7.5ms (1.2ms LB, 6.3ms MB)	80% of 3 partitions (1ms/partition)	Autotest (4 ms)
Short	KEY ON (STARTUP)	PLL0 50MHz	Loaded from Flash by SSCM	15ms (8.7ms LB, 6.3ms MB)	90% of 3partition s (6ms/partition)	Autotest (4 ms)
Medium	KEY OFF (SHUTDOWN)	PLL1 Full Freq	IOP using IPS I/F	48ms (29ms LB, 19ms MB)	90% of all partitions (6ms/partition)	Full w/o PMOS open
Diagnostic	Board-level Diagnostics	PLL0 Full Freq	IOP using IPS I/F or NEXUS JTAG I/F	680ms (660ms LB, 20ms MB)	94% of all partitions	Full set

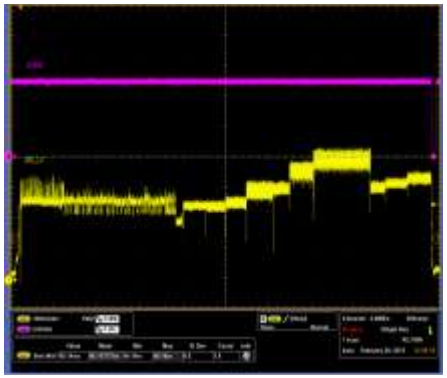
Ultra-Short



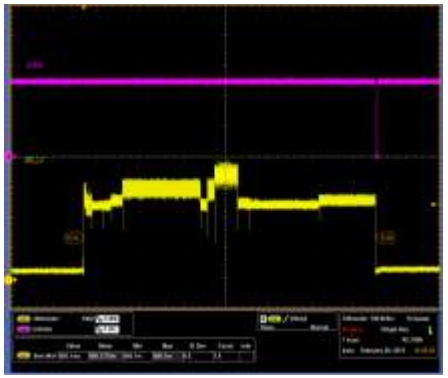
Short



Medium



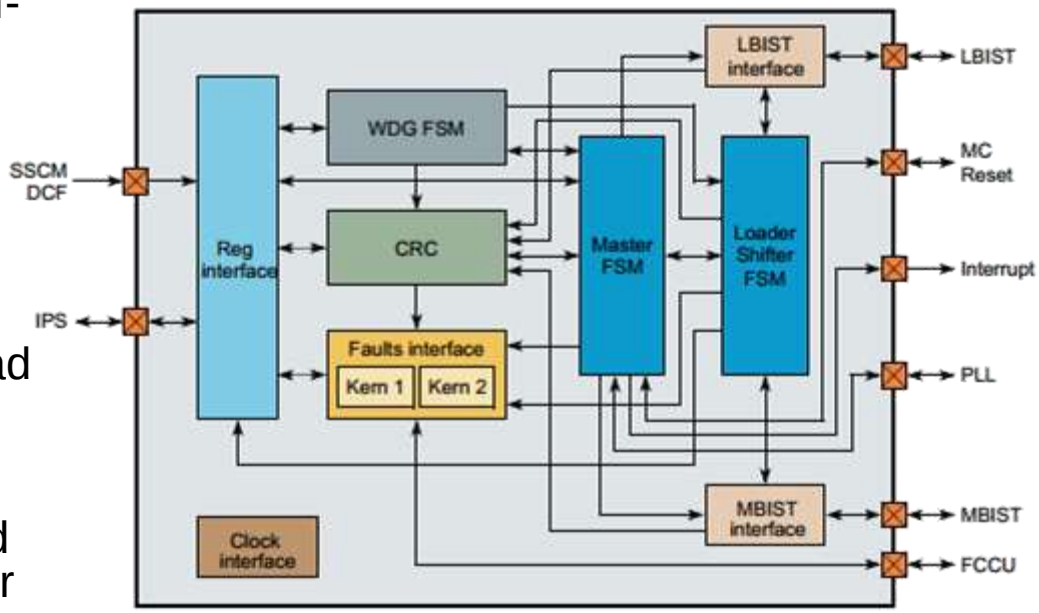
Diagnostic



BIST Hardware

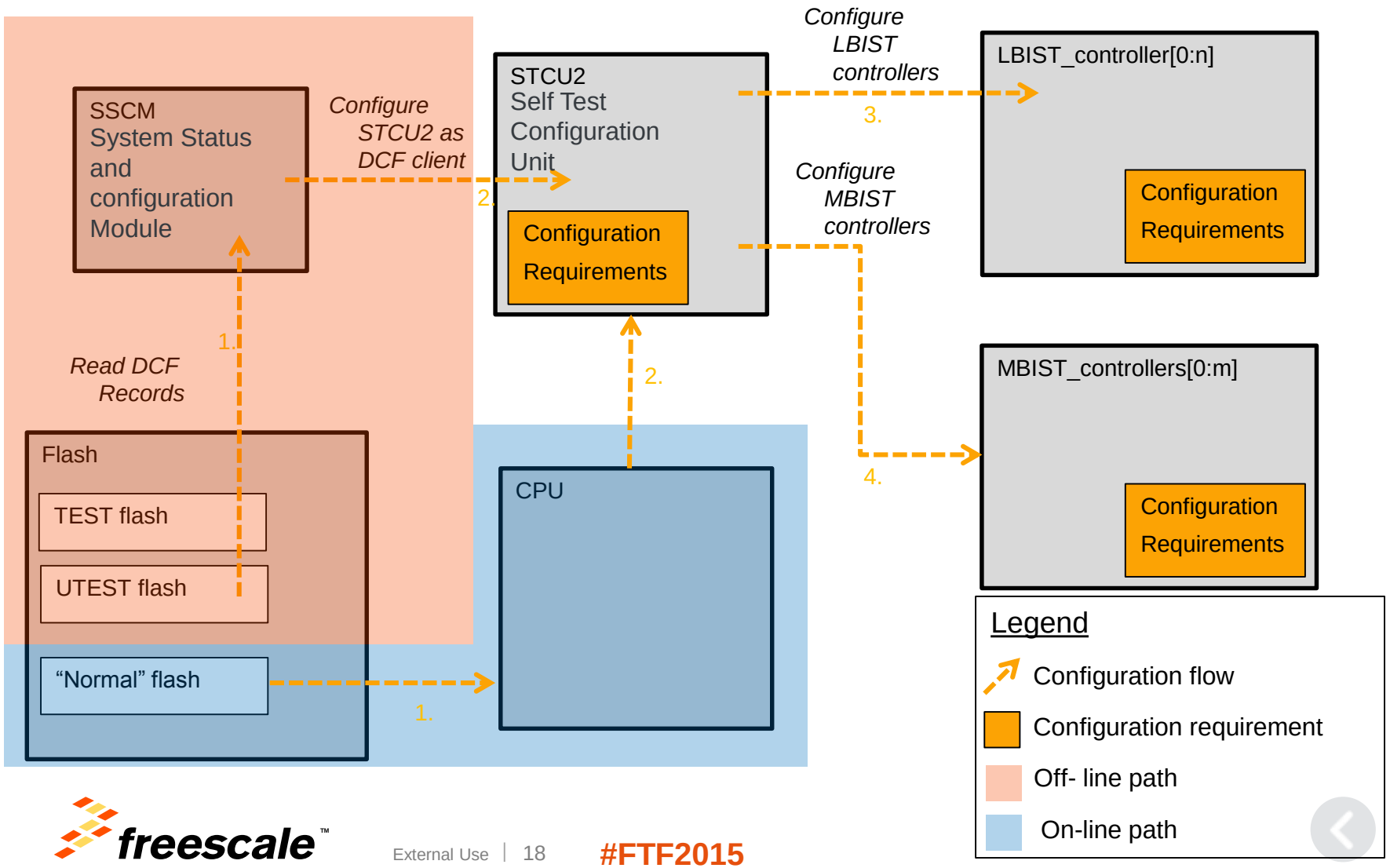
Self Test Control Unit summary

- The STCU2 is a programmable hardware module that controls the self-test sequence applied both during the offline and/or online conditions. It is able to manage by hardware the device's LBIST and MBIST blocks
- Off-line: The STCU2 uses the System Status and Configuration Module (SSCM) module which has a Device Configuration Format (DCF) bus to load the Self-test parameters from flash memory during the Off-Line Self-Test phase. This interface is able only to write the configuration parameters and start the Self-Test execution once after the STCU2 global reset has been applied
- On-line: The STCU2 register access by software is granted by a bus interface with the purpose to check the results of the Off-Line Self-Test but also to load/check the execution of the On-Line Self-Test



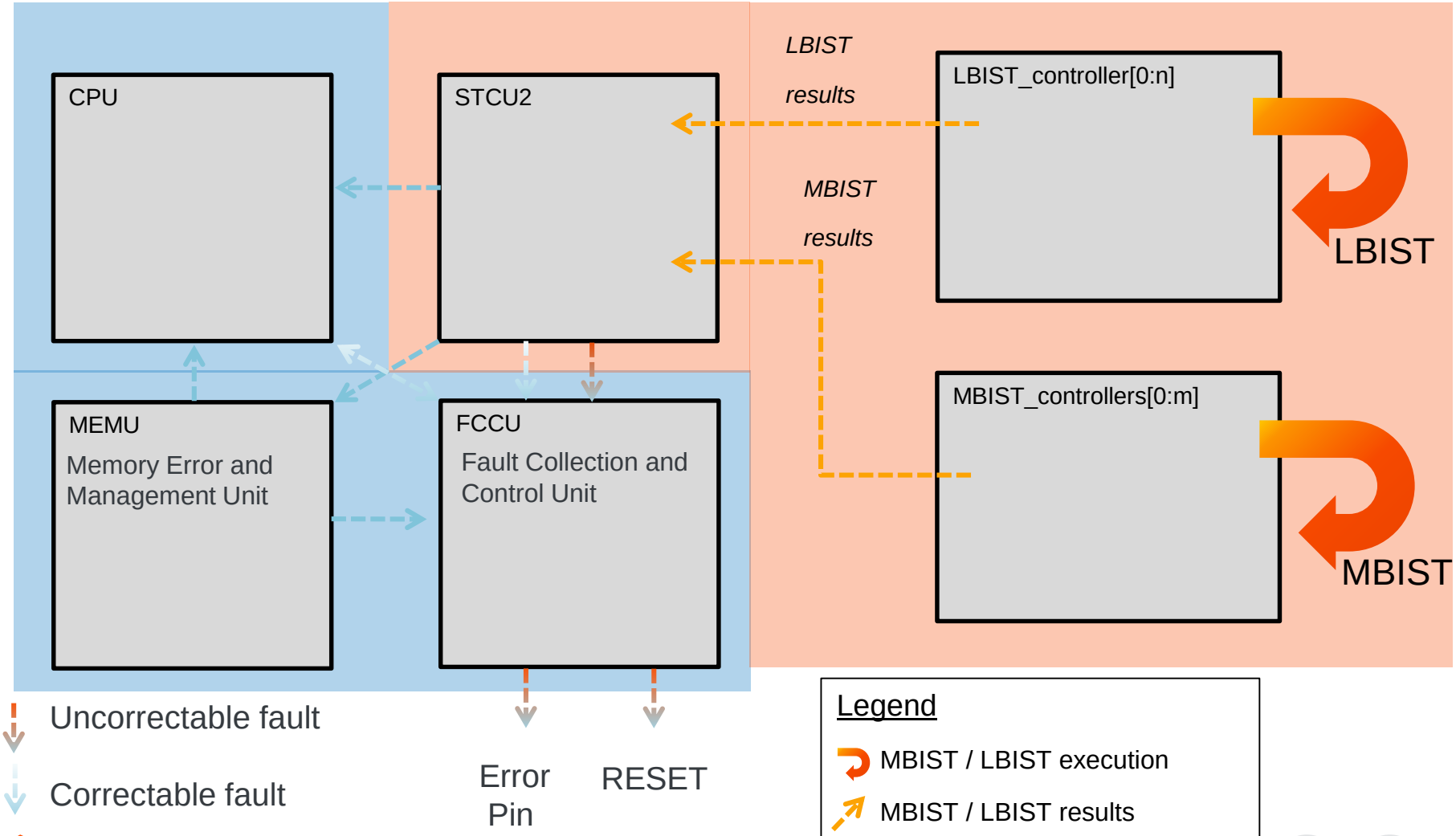
BIST configuration flow

STCU2: Self Test Configuration Unit
SSCM: System Status and Configuration Module



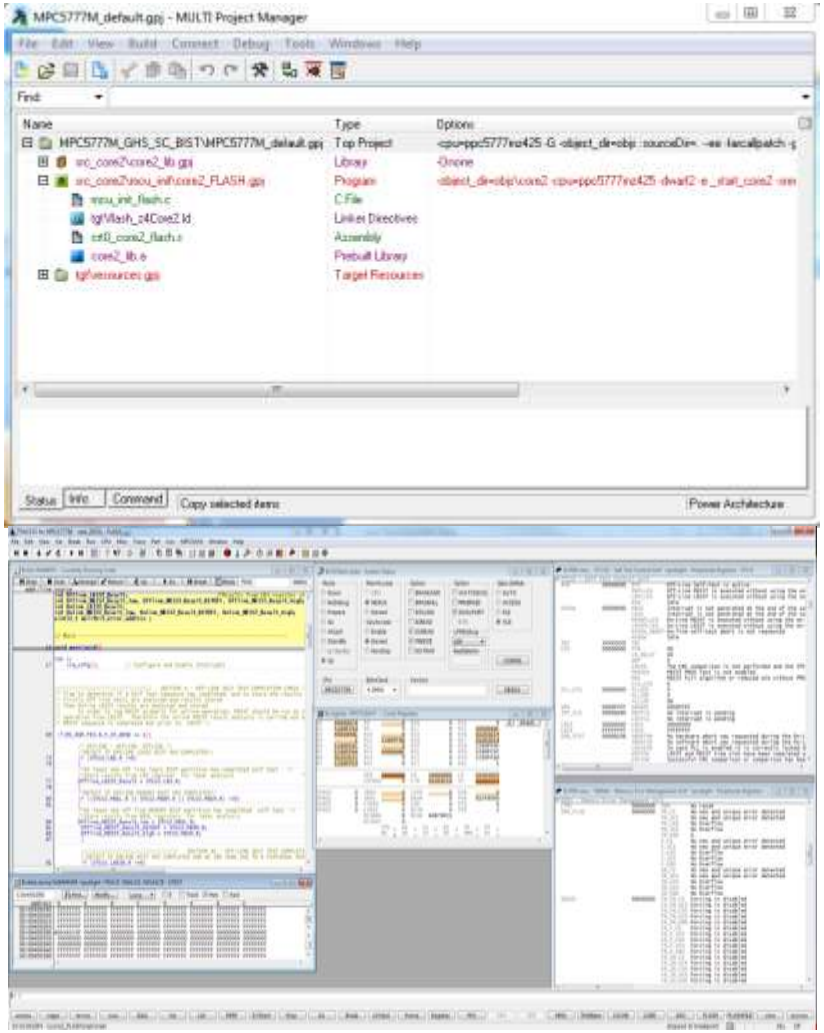
execution and results

FCCU:Fault Collection and Control Unit
MEMU:Memory Error Management Unit



Lab Exercise Project

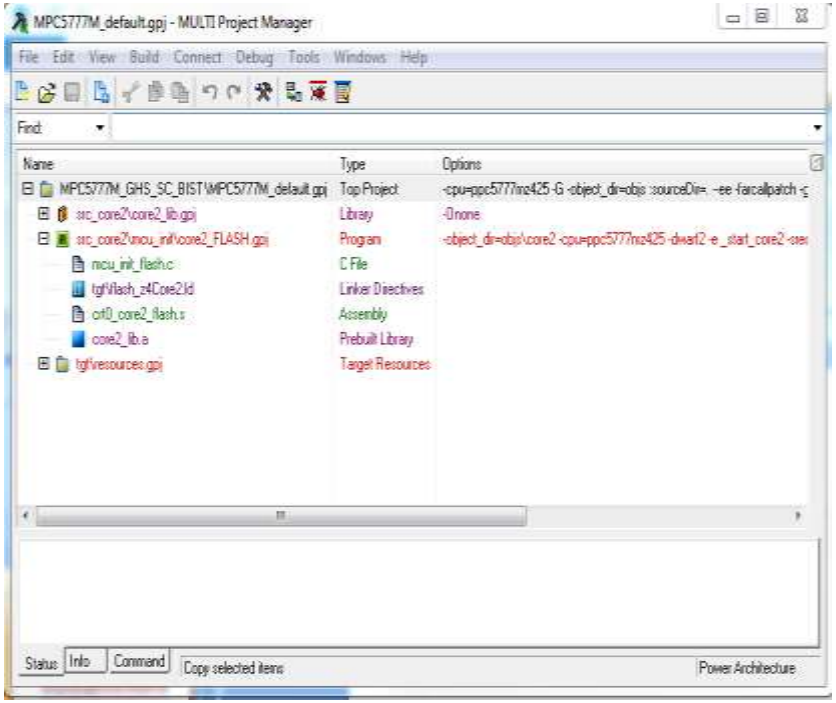
Lab Exercises



- The provided project contains 2 off-line and 2 online BIST configurations
- We will step through the operation of one online BIST, then work through a lab question to create a new On-line BIST configuration
- The project uses the Green Hills software MULTI IDE to compile and assemble the code, and the Lauterbach TRACE 32 debugger to run and step through the code

Project Structure

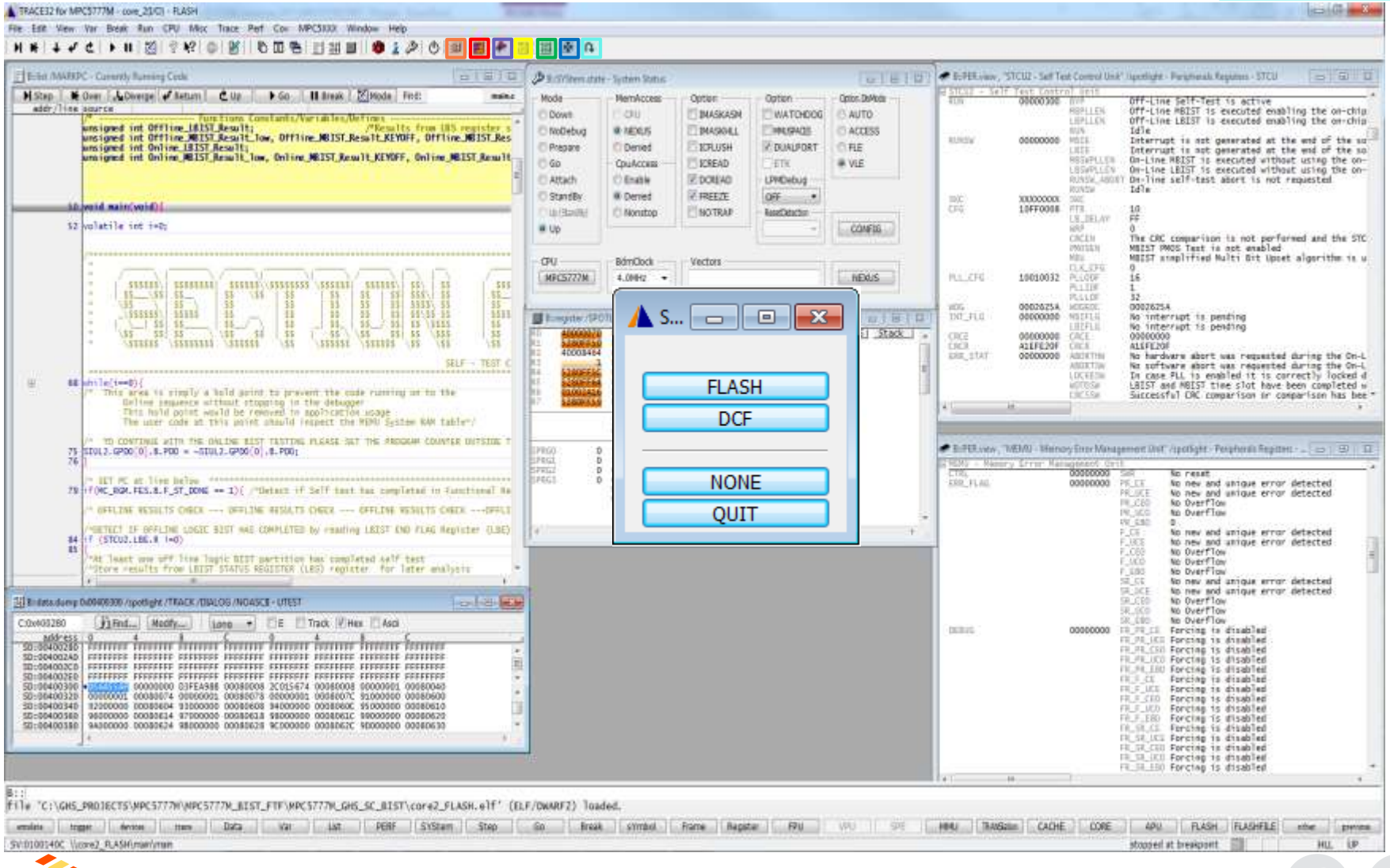
- DOCS
- MPC5777M_GHS_SC_BIST
- MPC5777M_LTB_DCF_RECORD
- config_mc.t32
- custom_windows.cmm
- DCF.cmm
- jpc577xm(BIST).cmm
- MPC5777M_SC_debug.cmm
- MPC5777M_SC_debug_start.bat



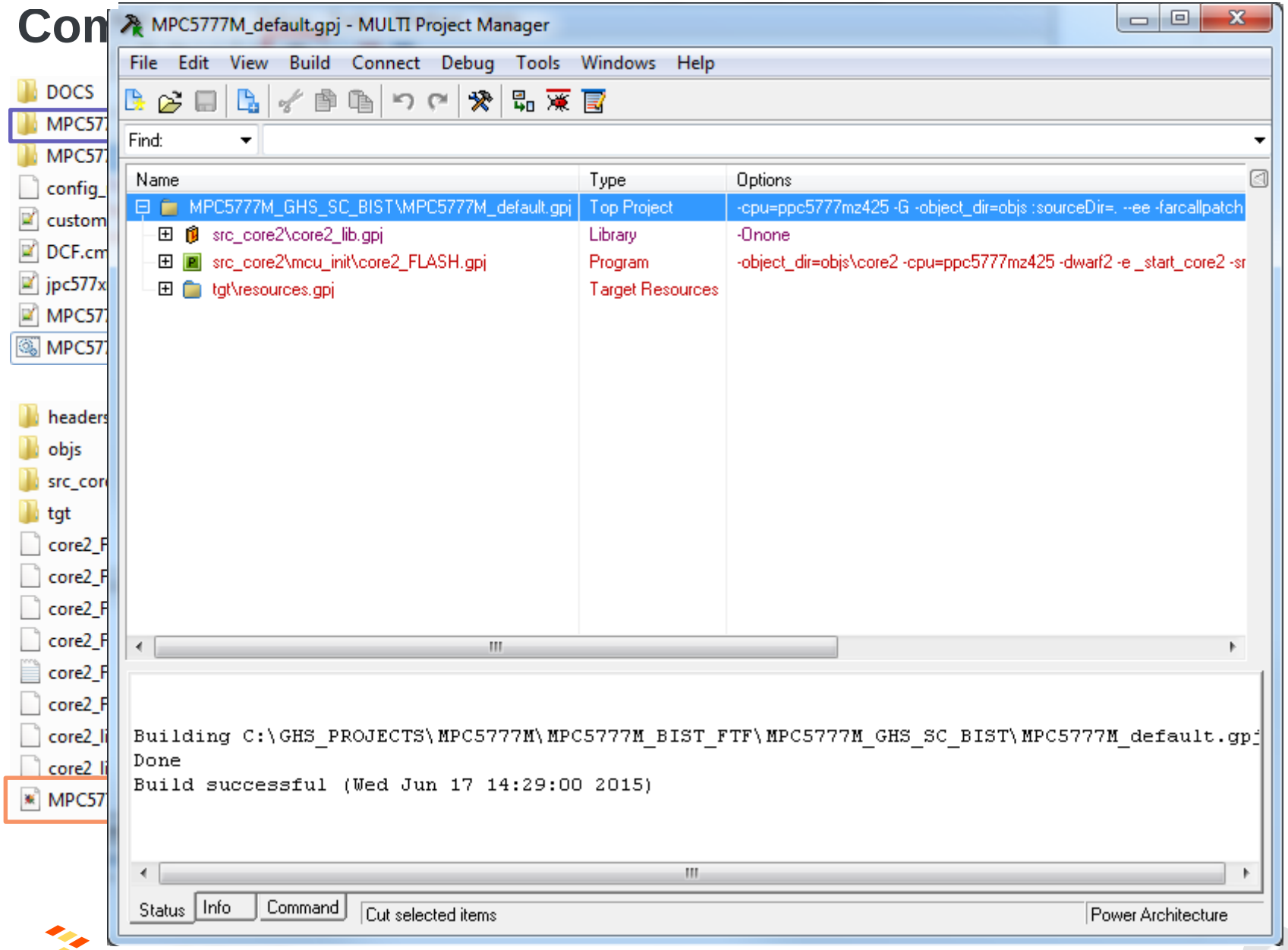
- Batch file launches custom Lauterbach TRACE scripts
- MPC5777M_GHS_SC_BIST contains the Green Hills MULTI project
- MPC5777M_LTB_DCF_RECORDS contains the flash programming scripts to configure offline BIST
- Docs contains the reference manual, Safety manual, and BIST pertinent documents we will access during the demo

TRACE 32 Customized Debugging Window

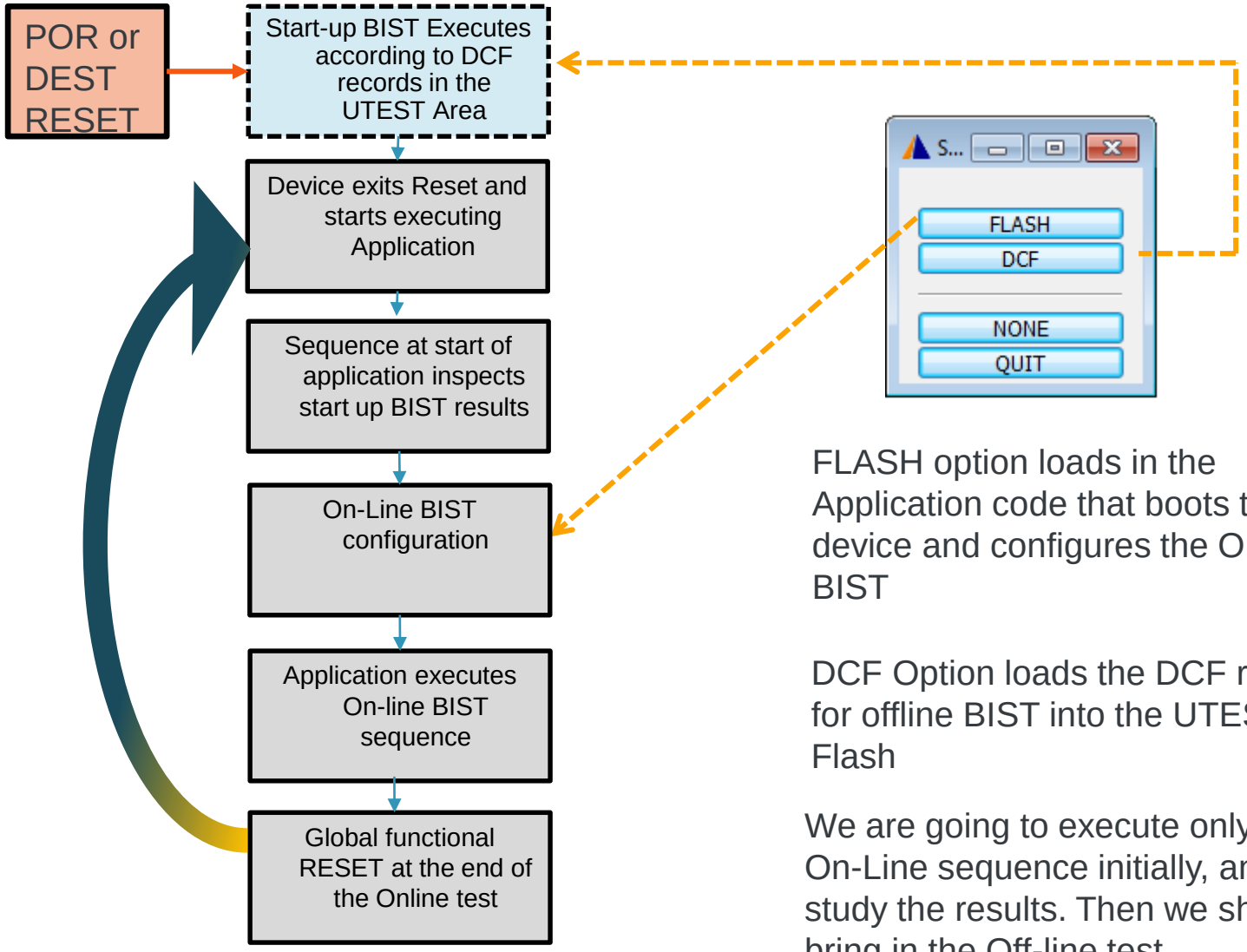
Re-Start Session | Build GHS Project | Update Flash and Load Symbols | DCF Record programming dialog | Show UTEST | Attach then break | Move PC out of start loop



Con


MPC5777M_default.gpj - MULTI Project Manager
File Edit View Build Connect Debug Tools Windows Help
Find:
Name
Type
Options
MPC5777M_GHS_SC_BIST\MPC5777M_default.gpj
Top Project
-cpu=ppc5777mz425 -G -object_dir=objs :sourceDir=. --ee -farcallpatch
src_core2\core2_lib.gpj
Library
-Onone
src_core2\mcu_init\core2_FLASH.gpj
Program
-object_dir=objs\core2 -cpu=ppc5777mz425 -dwarf2 -e _start_core2 -sr
tgt\resources.gpj
Target Resources
Building C:\GHS_PROJECTS\MPC5777M\MPC5777M_BIST_FTF\MPC5777M_GHS_SC_BIST\MPC5777M_default.gpj
Done
Build successful (Wed Jun 17 14:29:00 2015)
Status
Info
Command
Cut selected items
Power Architecture

Self-test sequence in the demonstration project

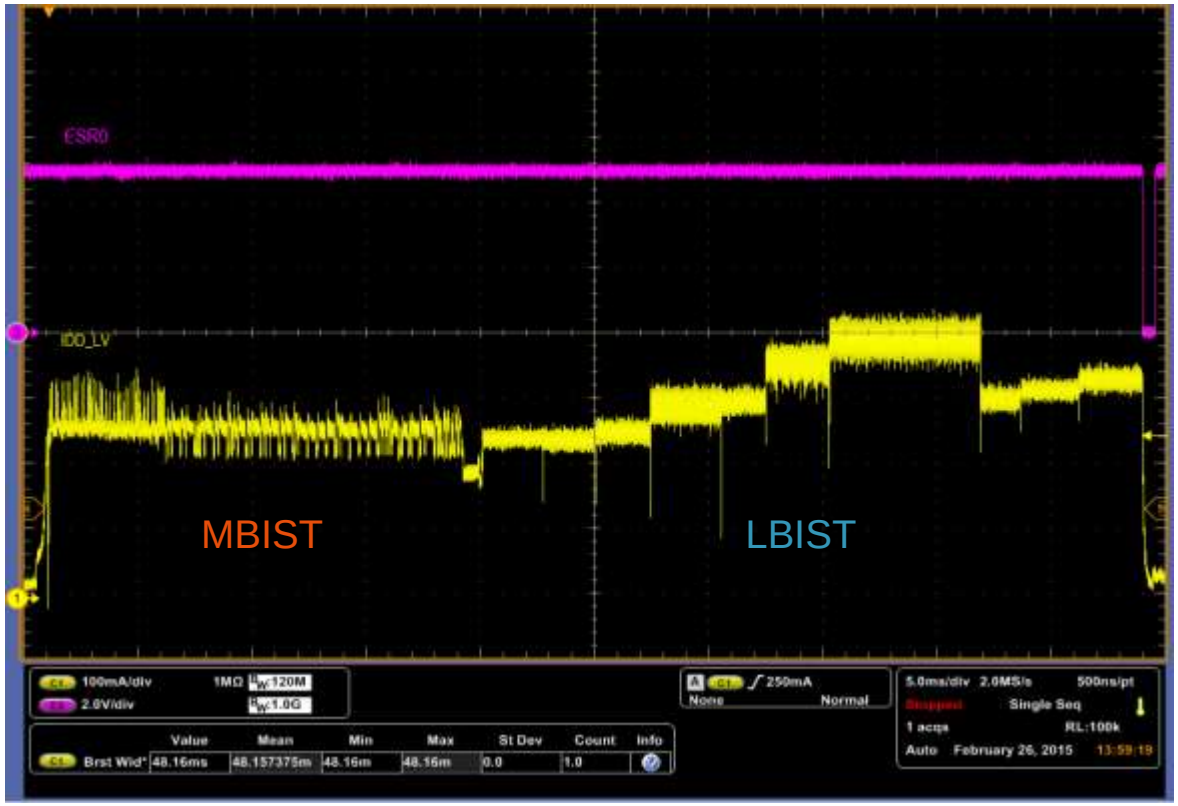


FLASH option loads in the Application code that boots the device and configures the On-line BIST

DCF Option loads the DCF records for offline BIST into the UTEST Flash

We are going to execute only the On-Line sequence initially, and study the results. Then we shall bring in the Off-line test

On-Line BIST “Key Off Example” step through



- This is an example configuration for a Key-off or MCU shutdown BIST. This BIST routine is intended to be run at the end of the application prior to shutting down the ECU. In this configuration all LBIST and MBIST partitions are tested as it is anticipated that there will be sufficient time to accommodate this at application Key-off
- This mode change stops all cores other than core 2 (which is the boot core on MPC5777M), configures the systems clock dividers and configures the PLL to support the BIST execution. The MBIST is configured to run at 200MHz and the LBIST is configured to execute at 50MHz. These are the maximum frequencies allowable.
- In this test all LBIST partitions are tested with 90% coverage. All MBIST partitions are tested with the full MBIST algorithm but without the PMOSEN test which provides additional coverage of decoders.

On-Line BIST Step Through Sequence

- The code is broken down into sections A through G
- A : Self test Completion check
- B: Device Init
- C: BIST Mode change
- D: STCU2 Unlock
- E: MBIST CONFIGURATION
- F: MBIST ERROR configuration
- G: LBIST Configuration
- H: LBIST Error handling
- I: Final config and Watchdog setting
- J: Start test

Section A – Checking for Results of Previous Tests

```c
/****************************************************************************************************
 *
 *     ______   _________   ______  _________  ______   ______   __    __        ______
 *    /      \ |         \ /      \|         \|      \ /      \ |  \  |  \      /      \
 *   |  $$$$$$\| $$$$$$$$$|  $$$$$$\\$$$$$$$$$ \$$$$$$|  $$$$$$\| $$\ | $$     |  $$$$$$\
 *   | $$___\$$| $$__    | $$   \$$  | $$      | $$  | $$  | $$| $$$\| $$     | $$__| $$
 *    \$$    \ | $$  \   | $$        | $$      | $$  | $$  | $$| $$$$\ $$     | $$    $$
 *    _\$$$$$$\| $$$$$   | $$   __   | $$      | $$  | $$  | $$| $$\$$ $$     | $$$$$$$$
 *   |  \__| $$| $$_____ | $$__/  \  | $$     _| $$_ | $$__/ $$| $$ \$$$$     | $$  | $$
 *    \$$    $$| $$     \ \$$    $$  | $$    |   $$ \ \$$    $$| $$  \$$$     | $$  | $$
 *     \$$$$$$  \$$$$$$$$  \$$$$$$    \$$     \$$$$$$  \$$$$$$  \$$   \$$      \$$   \$$

                                                                  SELF - TEST COMPLETION CHECK
 ****************************************************************************************************/
68  while(i==0){
    /*  This area is simply a hold point to prevent the code running on to the
            Online sequence without stopping in the debugger
            This hold point would be removed in application usage
            The user code at this point should inspect the MEMU System RAM table*/

    /*  TO CONTINUE WITH THE ONLINE BIST TESTING PLEASE SET THE PROGRAM COUNTER OUTSIDE THE WHILE LOOP      */
75  SIUL2.GPDO[0].B.PDO = ~SIUL2.GPDO[0].B.PDO;
76  }

    /* SET PC at line below  **********************************************************************/
79  if(MC_RGM.FES.B.F_ST_DONE == 1){ /*Detect if Self test has completed in Functional Reset Status register  */

    /* OFFLINE RESULTS CHECK --- OFFLINE RESULTS CHECK --- OFFLINE RESULTS CHECK ---OFFLINE RESULTS CHECK --- */

    /*DETECT IF OFFLINE LOGIC BIST HAS COMPLETED by reading LBIST END FLAG Register (LBE)        */
84  if (STCU2.LBE.R !=0)
85  {
    /*At least one off line logic BIST partition has completed self test
    /*Store results from LBIST STATUS REGISTER (LBS) register  for later analysis              */
88  Offline_LBIST_Result = STCU2.LBS.R;
89  }
    /*DETECT IF OFFLINE MEMORY BIST HAS COMPLETED - Read Offline MBIST End Flags (MBEx) registers*/
91  if ((STCU2.MBEL.R || STCU2.MBEM.R || STCU2.MBEH.R) !=0)
92  {
    /*At least one off line MEMORY BIST partition has completed  self test  */
    /*Store results from Offline MBIST status registers (MBSx) registers  for later analysis*/
95  Offline_MBIST_Result_low = STCU2.MBSL.R;
96  Offline_MBIST_Result_KEYOFF = STCU2.MBSM.R;
97  Offline_MBIST_Result_high = STCU2.MBSH.R;
98  }


    /* ONLINE RESULTS CHECK --- ONLINE RESULTS CHECK --- ONLINE RESULTS CHECK ---ONLINE RESULTS CHECK --- */
    /* Read On-Line LBIST END Flag register (LBESW)to determine if we have completed On-line LBIST */
103 if (STCU2.LBESW.R !=0)
104 {
    /*At least one On line logic BIST partition has completed self test*/
    /*Store results from LBSSW register  for later analysis*/
107 Online_LBIST_Result = STCU2.LBSSW.R;
108 }
    /* Read On-Line MBIST END Flag registers (MBExSW) to determine if we have completed On-line MBIST */
110 if (STCU2.MBELSW.R!=0 || STCU2.MBEMSW.R!=0 || STCU2.MBEHSW.R!=0)
111 {
    /*At least one On line Memory BIST partition has completed self test*/
    /*Store results from LBSSW register  for later analysis*/
114 Online_MBIST_Result_low = STCU2.MBSLSW.R;
115 Online_MBIST_Result_KEYOFF = STCU2.MBSMSW.R;
116 Online_MBIST_Result_high = STCU2.MBSHSW.R;

118 }
```

- Sequence is intended to hold the program
- STCU2 registers are tested to check for completion of any BIST tests and the results are stored for later analysis
- Set the program counter to line 79 (using the “out of loop” button” to move the program out of the while hold loop.

Section B and Section C

- Section B is the standard MPC5777M device initialization sequence – All clocks configured, PLL's enabled, Progressive clock switching configured, Peripherals and other cores configured
- Section C reconfigures the MPC5777M to the highest frequency for Online BIST(200MHz)
- Section C also disables the auxiliary clocks and disables the peripherals

Section D Unlock Sequence

```
/*****************************************************************************************************
 *     $$$$$$\  $$$$$$$$\  $$$$$$\ $$$$$$$$\ $$$$$$\  $$$$$$\  $$\   $$\       $$$$$$$\
 *    $$  __$$\ $$  _____|$$  __$$\\__$$  __|\_$$  _|$$  __$$\ $$$\  $$ |      $$  __$$\
 *    $$ /  \__|$$ |      $$ /  \__|  $$ |     $$ |  $$ /  $$ |$$$$\ $$ |      $$ |  $$ |
 *    \$$$$$$\  $$$$$\    $$ |        $$ |     $$ |  $$ |  $$ |$$ $$\$$ |      $$ |  $$ |
 *     \____$$\ $$  __|   $$ |        $$ |     $$ |  $$ |  $$ |$$ \$$$$ |      $$ |  $$ |
 *    $$\   $$ |$$ |      $$ |  $$\   $$ |     $$ |  $$ |  $$ |$$ |\$$$ |      $$ |  $$ |
 *    \$$$$$$  |$$$$$$$$\ \$$$$$$  |  $$ |   $$$$$$\  $$$$$$  |$$ | \$$ |      $$$$$$$  |
 *     \______/ \________| \______/   \__|   \______| \______/ \__|  \__|      \_______/
 *
                                  Unlock STCU using keys
 *****************************************************************************************************/

/* Write on-line unlock keys to the STCU */
177 STCU2.SKC.R = 0x753F924E;
178 STCU2.SKC.R = 0x8AC06DB1;
```

```
/* Write key 2 to service the watchdog */
219 STCU2.SKC.R = 0x8AC06DB1;
```

- There are three different watchdogs that should not be confused!!!!
- 1) Initialization watchdog – Hardcoded time-out that detects faults in the initialization phase preventing the self-test running or being bypassed. Nothing required from user side
- 2) Execution of BIST tests- If the tests do not complete within user defined time the watchdog will time out and Reset the device if configured to do so
- 3) Register write access timeout – Access to the STCU2 registers during the configuration phase is unlocked via a set of keys to prevent unintentional access. After a predetermined amount of cycles the access rights will time out unless the second part of the key is written
- The STCU2_SKC register implements the security key code mechanism needed to begin the write sequence to the other STCU2 registers.
- Depending on the off/on-line test step, the two keys will be different. Byte write operation is not allowed since the full key has to be recognized as one unit.

Section E Configuring MBIST partitions

```
/*****************************************************************************
*    $$$$$$\  $$$$$$$$\  $$$$$$\  $$$$$$$$\ $$$$$$\  $$$$$$\  $$\   $$\     $$$$$$$$\
*   $$  __$$\ $$  _____|$$  __$$\ \__$$  __|\_$$  _|$$  __$$\ $$$\  $$ |    $$  _____|
*   $$ /  \__|$$ |      $$ /  $$ |   $$ |     $$ |  $$ /  $$ |$$$$\ $$ |    $$ |
*   \$$$$$$\  $$$$$\    $$$$$$$$ |   $$ |     $$ |  $$ |  $$ |$$ $$\$$ |    $$$$$\
*    \____$$\ $$  __|   $$  __$$ |   $$ |     $$ |  $$ |  $$ |$$ \$$$$ |    $$  __|
*   $$\   $$ |$$ |      $$ |  $$ |   $$ |     $$ |  $$ |  $$ |$$ |\$$$ |    $$ |
*   \$$$$$$  |$$$$$$$$\ $$ |  $$ |   $$ |   $$$$$$\  $$$$$$  |$$ | \$$ |    $$$$$$$$\
*    \______/ \________|\__|  \__|   \__|   \______| \______/ \__|  \__|    \________|
*
*****************************************************************************/

/* MBIST Config - MBIST Will run first */

/* Configure MBIST Control registers */
STCU2.MB_CTRL[0].R = 0x91000000;        /* MBIST CTRL00 Run concurrently, next in sequence is MBIST 1 */
STCU2.MB_CTRL[1].R = 0x92000000;        /* MBIST CTRL01 Run concurrently, next in sequence is MBIST 2 */
STCU2.MB_CTRL[2].R = 0x93000000;        /* MBIST CTRL02 Run concurrently, next in sequence is MBIST 3 */
STCU2.MB_CTRL[3].R = 0x94000000;        /* MBIST CTRL03 Run concurrently, next in sequence is MBIST 4 */
STCU2.MB_CTRL[4].R = 0x95000000;        /* MBIST CTRL04 Run concurrently, next in sequence is MBIST 5 */
STCU2.MB_CTRL[5].R = 0x96000000;        /* MBIST CTRL05 Run concurrently, next in sequence is MBIST 6 */
STCU2.MB_CTRL[6].R = 0x97000000;        /* MBIST CTRL06 Run concurrently, next in sequence is MBIST 7 */
STCU2.MB_CTRL[7].R = 0x98000000;        /* MBIST CTRL07 Run concurrently, next in sequence is MBIST 8 */
STCU2.MB_CTRL[8].R = 0x99000000;        /* MBIST CTRL08 Run concurrently, next in sequence is MBIST 9 */
STCU2.MB_CTRL[9].R = 0x9A000000;        /* MBIST CTRL09 Run concurrently, next in sequence is MBIST 10 */
STCU2.MB_CTRL[10].R = 0x9B000000;       /* MBIST CTRL10 Run concurrently, next in sequence is MBIST 11 */
STCU2.MB_CTRL[11].R = 0x9C000000;       /* MBIST CTRL11 Run concurrently, next in sequence is MBIST 12 */
STCU2.MB_CTRL[12].R = 0x9D000000;       /* MBIST CTRL12 Run concurrently, next in sequence is MBIST 13 */
STCU2.MB_CTRL[13].R = 0x9E000000;       /* MBIST CTRL13 Run concurrently, next in sequence is MBIST 14 */
STCU2.MB_CTRL[14].R = 0x9F000000;       /* MBIST CTRL14 Run concurrently, next in sequence is MBIST 15 */
STCU2.MB_CTRL[15].R = 0xA0000000;       /* MBIST CTRL15 Run concurrently, next in sequence is MBIST 16 */
STCU2.MB_CTRL[16].R = 0xA1000000;       /* MBIST CTRL16 Run concurrently, next in sequence is MBIST 17 */
STCU2.MB_CTRL[17].R = 0xA2000000;       /* MBIST CTRL17 Run concurrently, next in sequence is MBIST 18 */
STCU2.MB_CTRL[18].R = 0xA3000000;       /* MBIST CTRL18 Run concurrently, next in sequence is MBIST 19 */
STCU2.MB_CTRL[19].R = 0xA4000000;       /* MBIST CTRL19 Run concurrently, next in sequence is MBIST 20 */
STCU2.MB_CTRL[20].R = 0xA5000000;       /* MBIST CTRL20 Run concurrently, next in sequence is MBIST 21 */

/* Write key 2 to service the watchdog */
STCU2.SKC.R = 0x8AC06DB1;
```

The STCU2_MB_CTRL register defines the control setting of MBIST controller.

The R/W fields in this register are readable at any time. You can write to these fields when either of the following conditions is true:

- Off-line Self-Test phase is active
- On-line Self-Test phase is active and STCU2_CFG[WRP] = 0

The offset of this register is a function of NMCUT .

Address: 0h base + 600h offset + (4d × i), where i=0d to 77d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	CSM	PTR							0							
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0															
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- The reset value is chip-specific; see the chapter that describes how modules are configured and connected.x = Undefined at reset.

- CSM determines whether the partitions are tested concurrently or sequentially
- PTR defines the logical pointer to the next LBIST or MBIST partition
- 0h to (LBIST-1) = LBIST
- 10h to (MBIST-1) = MBIST
- 7Fh = pointer to Nil – CSM must be set to sequential in this instance
- To reduce current consumption our example runs two sequential blocks of concurrent partitions

Please open MBIST_Partitions.xlsx in the docs folder!

Section F MBIST Fault Mapping Configuration

```
/*****************************************************************************************************************
 *     $$$$$$\  $$$$$$$$\  $$$$$$\ $$$$$$$$\ $$$$$$\  $$$$$$\  $$\   $$\       $$$$$$$\
 *    $$  __$$\ $$  _____|$$  __$$\\__$$  __|\_$$  _|$$  __$$\ $$$\  $$ |      $$  _____|
 *    $$ /  \__|$$ |      $$ /  \__|  $$ |     $$ |  $$ /  $$ |$$$$\ $$ |      $$ |
 *    \$$$$$$\  $$$$$\    $$ |        $$ |     $$ |  $$ |  $$ |$$ $$\$$ |      $$$$$\
 *     \____$$\ $$  __|   $$ |        $$ |     $$ |  $$ |  $$ |$$ \$$$$ |      $$  __|
 *    $$\   $$ |$$ |      $$ |  $$\   $$ |     $$ |  $$ |  $$ |$$ |\$$$ |      $$ |
 *    \$$$$$$  |$$$$$$$$\ \$$$$$$  |  $$ |   $$$$$$\  $$$$$$  |$$ | \$$ |      $$ |
 *     \______/ \________| \______/   \__|   \______| \______/ \__|  \__|      \__|
 *
 *                          Configure MBIST recoverable/unrecoverable fault handling
 *****************************************************************************************************************/


307 STCU2.MBUFML.R = 0x0;          /* Configure all faults as recoverable in this example */
308 STCU2.MBUFMM.R = 0x0;          /* Configure all faults as recoverable in this example */
309 STCU2.MBUFMH.R = 0x0;          /* Configure all faults as recoverable in this example */
```

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	MBUFM31	MBUFM30	MBUFM29	MBUFM28	MBUFM27	MBUFM26	MBUFM25	MBUFM24	MBUFM23	MBUFM22	MBUFM21	MBUFM20	MBUFM19	MBUFM18	MBUFM17	MBUFM16
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	MBUFM15	MBUFM14	MBUFM13	MBUFM12	MBUFM11	MBUFM10	MBUFM9	MBUFM8	MBUFM7	MBUFM6	MBUFM5	MBUFM4	MBUFM3	MBUFM2	MBUFM1	MBUFM0
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- The reset value is chip-specific; see the chapter that describes how modules are configured and connected.x = Undefined at reset.

STCU2_MBUFML field descriptions

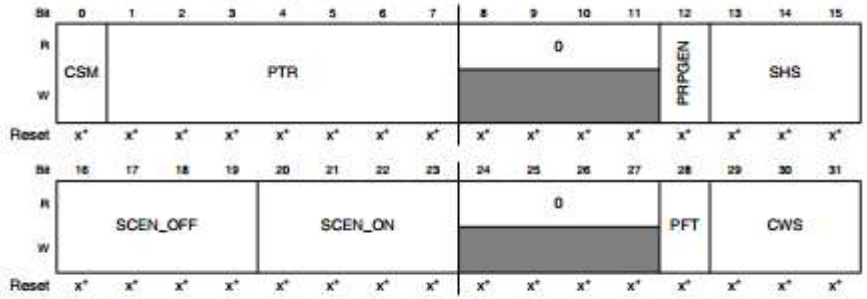
Field	Description
0 MBUFM31	MBIST Unrecoverable Fault Mapping 0 Recoverable Fault mapping 1 Unrecoverable Fault mapping
1	MBIST Unrecoverable Fault Mapping

- The MBUFML/M/H registers allow the user to configure whether a fault in a particular MBIST partition is recoverable or unrecoverable.
- This consequently determines whether the STCU2 recoverable or STCU2 unrecoverable flag is set in the Fault control and Configuration Unit (FCCU)

Section G Configuring Online LBIST Partitions

```
/*****************************************************************************************************************
 *     $$$$$$\  $$$$$$$$\  $$$$$$\ $$$$$$$$\ $$$$$$\  $$$$$$\  $$\   $$\        $$$$$$\
 *    $$  __$$\ $$  _____|$$  __$$\\__$$  __|\_$$  _|$$  __$$\ $$$\  $$ |      $$  __$$\
 *    $$ /  \__|$$ |      $$ /  \__|  $$ |     $$ |  $$ /  $$ |$$$$\ $$ |      $$ /  \__|
 *    \$$$$$$\  $$$$$\    $$ |        $$ |     $$ |  $$ |  $$ |$$ $$\$$ |      $$ |$$$$\
 *     \____$$\ $$  __|   $$ |        $$ |     $$ |  $$ |  $$ |$$ \$$$$ |      $$ |\_$$ |
 *    $$\   $$ |$$ |      $$ |  $$\   $$ |     $$ |  $$ |  $$ |$$ |\$$$ |      $$ |  $$ |
 *    \$$$$$$  |$$$$$$$$\ \$$$$$$  |  $$ |   $$$$$$\  $$$$$$  |$$ | \$$ |      \$$$$$$  |
 *     \______/ \________| \______/   \__|   \______| \______/ \__| \__|       \______/
 *
 *                                                          On-line LBIST partition configuration
 *****************************************************************************************************************/


/* Configure Online LBIST partition coverage                                    */
/* Configure LBIST for 90% coverage using KEYOFF #define                 */
/*                          OR                                                  */
/*Configure LBIST for 94% coverage using DIAGNOSTIC #define     */



/* LBISTO HSM Config */
#ifdef KEYOFF
337 STCU2.LB[0].LB_CTRL.R = 0x01035507;       /* LBIST CTRL0 Run concurrently, next in sequence is LBIST 1 */
338 STCU2.LB[0].LB_PCS.R = 0x00000580;        /* LBIST 0 pattern count */
339 STCU2.LB[0].LB_MISRELSW.R = 0xC600B837;  /* LBIST MISREL Expected Low */
340 STCU2.LB[0].LB_MISREHSW.R = 0x9CC36E3F;  /* LBIST MISREH Expected High */
#endif

#ifdef DIAGNOSTIC
STCU2.LB[0].LB_CTRL.R = 0x01035507;       /* LBIST CTRL0 Run concurrently, next in sequence is LBIST 1 */
STCU2.LB[0].LB_PCS.R = 0x00003380;        /* LBIST 0 pattern count */
STCU2.LB[0].LB_MISRELSW.R = 0xE29FCE36;   /* LBIST MISREL Expected Low */
STCU2.LB[0].LB_MISREHSW.R = 0x50EA7D46;   /* LBIST MISREH Expected High */
#endif
```

Address: 0h base + 100h offset + (64d × i), where i=0d to 9d



* Notes:
- The reset value is chip-specific; see the chapter that describes how modules are configured and connected.x = Undefined at reset.

- LB_CTRL register is similar to the MBIST_CTRL register as it configures concurrent /sequential partition mode, and sets the next partition to be tested.
- LB_CTRL also contains fields for LBIST shift speed, SCAN ENABLE OFF delay, SCAN ENABLE ON delay, Past Flush test, Capture window size and PPGEN.
- The values for the LB_CTRL fields are provided my Freescale, in the Excel table accompanying this exercise.

Section G Configuring Online LBIST partitions cont'd

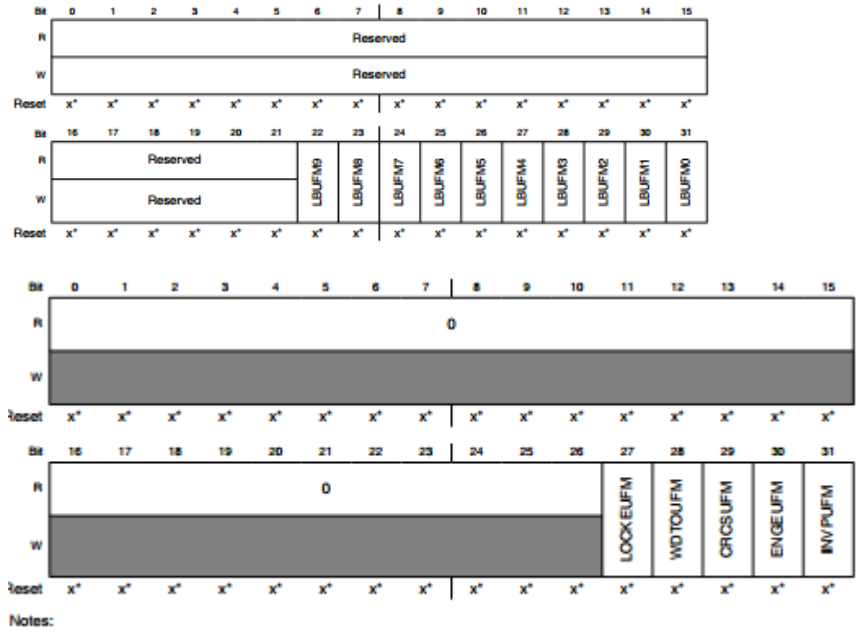
Partition	STCU REG	KEYON (OFFLINE) Mode ULTRA-SHORT	KEYON (OFFLINE) Mode SHORT	KEYOFF (ONLINE) Mode	DEBUG Mode
Misc. STCU Configuration Settigns	STCU_PLL_CFG	08010019	08010019	-	-
	STCU_VD0	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
	STCU_LBPMSV	000003FF	000003FF	000003FF	000003FF
partition 0	CONFPASF	81.84%	90.05%	90.05%	94.00%
	SHIFT SPEED(MHz)	50	50	50	50
	CAPTURE SPEED	SLOW SPEED	SLOW SPEED	FUNC. SPEED	FUNC. SPEED
	TEST TIME (ms)	0.37	2.70	2.37	22.15
	STCU_LB_CTRL_0	00865507	00865507	00035507	00035507
	STCU_LB_PCS_0	000000C0	00000580	00000580	00003380
	STCU_LB_PRPGL_0	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
	STCU_LB_PRPGH_0	-	-	-	-
	STCU_LB_MISREL_0	EDAD9DED	8BCA2E09		
	STCU_LB_MISREH_0	DBE388A1	A9B8733C		
	STCU_LB_MISRELSV_0			C600B837	E29FCE36
	STCU_LB_MISREHSV_0			9CC36E3F	50EA7D46

```
#ifdef KEYOFF
STCU2.LB[0].LB_CTRL.R = 0x01035507;       /* LBIST CTRL0 Run concurrently, next in sequence is LBIST 1 */
STCU2.LB[0].LB_PCS.R = 0x00000580;        /* LBIST 0 pattern count */
STCU2.LB[0].LB_MISRELSW.R = 0xC600B837;   /* LBIST MISREL Expected Low */
STCU2.LB[0].LB_MISREHSW.R = 0x9CC36E3F;   /* LBIST MISREH Expected High */
#endif

#ifdef DIAGNOSTIC
STCU2.LB[0].LB_CTRL.R = 0x01035507;       /* LBIST CTRL0 Run concurrently, next in sequence is LBIST 1 */
STCU2.LB[0].LB_PCS.R = 0x00003380;        /* LBIST 0 pattern count */
STCU2.LB[0].LB_MISRELSW.R = 0xE29FCE36;   /* LBIST MISREL Expected Low */
STCU2.LB[0].LB_MISREHSW.R = 0x50EA7D46;   /* LBIST MISREH Expected High */
#endif
```

- The Excel document contains the pertinent settings for each of the LBIST partitions for Offline and Online mode
- PCS register = Pattern counter stop
- MISRELSW and MISREHSW registers are populated with the expected MISR values that will be returned by the LBIST unit.
- MISR – In BIST it is not possible to store all the circuit outputs, but the circuit output can be compressed to form a signature that can later be compared to the known "golden" signature. The Multiple – Input Signature Register is a type of Linear feedback Signature register. Each state of the MISR relies on the previous states rather than just the current state, so the MISR will always generate the same correct output sequence from the same input sequence
- The expected MISR signatures are provided by Freescale
- All 10 partitions are configured.

Section H LBIST fault handling and Global Reset

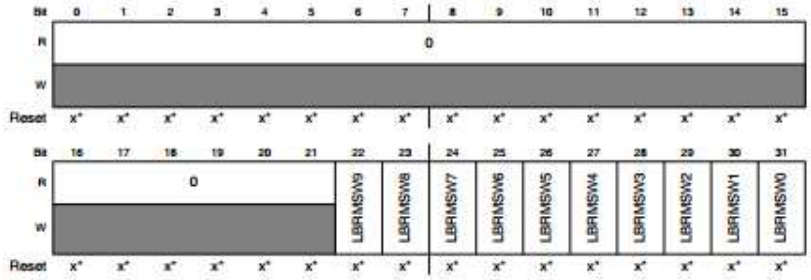
```
/*****************************************************************************************************************
 *     $$$$$$\  $$$$$$$$\  $$$$$$\ $$$$$$$$\ $$$$$$\  $$$$$$\  $$\    $$\       $$\   $$\
 *    $$  __$$\ $$  _____|$$  __$$\\__$$  __|\_$$  _|$$  __$$\ $$$\   $$ |      $$ |  $$ |
 *    $$ /  \__|$$ |      $$ /  \__|  $$ |     $$ |  $$ /  $$ |$$$$\  $$ |      $$ |  $$ |
 *    \$$$$$$\  $$$$$\    $$ |        $$ |     $$ |  $$ |  $$ |$$ $$\ $$ |      $$$$$$$$ |
 *     \____$$\ $$  __|   $$ |        $$ |     $$ |  $$ |  $$ |$$ \$$$$ |      $$  __$$ |
 *    $$\   $$ |$$ |      $$ |  $$\   $$ |     $$ |  $$ |  $$ |$$ |\$$$ |      $$ |  $$ |
 *    \$$$$$$  |$$$$$$$$\ \$$$$$$  |  $$ |   $$$$$$\  $$$$$$  |$$ | \$$ |      $$ |  $$ |
 *     \______/ \________| \______/   \__|   \______| \______/ \__|  \__|      \__|  \__|
 *
 *                              Configure LBIST STCU2 Fault handling
 *****************************************************************************************************************/
/* Configure LBIST recoverable/unrecoverable fault handling  */
STCU2.LBUFM.R = 0x0;                    /* Configure all faults as recoverable in this example */
/*  Configure LBIST error recoverable/unrecoverable fault handling  */
STCU2.ERR_FM.R = 0x0;                   /* Configure all faults as recoverable in this example */
/*  Configure LBIST RESET management register  */
/* Set Global function Reset to occur at end of LBIST 9 */
STCU2.LBRMSW.R = 0x200;
```



- The LBUFM register is used to configure whether each LBIST partition generates an un-recoverable fault .
- The Error FM configuration determines whether the other STCU2 faults generate a recoverable or unrecoverable fault

Section H Global Reset

```
/*****************************************************************************************************************
 *     $$$$$$\  $$$$$$$$\  $$$$$$\ $$$$$$$$\ $$$$$$\  $$$$$$\  $$\    $$\       $$\   $$\
 *    $$  __$$\ $$  _____|$$  __$$\\__$$  __|\_$$  _|$$  __$$\ $$$\   $$ |      $$ |  $$ |
 *    $$ /  \__|$$ |      $$ /  \__|  $$ |     $$ |  $$ /  $$ |$$$$\  $$ |      $$ |  $$ |
 *    \$$$$$$\  $$$$$\    $$ |        $$ |     $$ |  $$ |  $$ |$$ $$\ $$ |      $$$$$$$$ |
 *     \____$$\ $$  __|   $$ |        $$ |     $$ |  $$ |  $$ |$$ \$$$$ |      $$  __$$ |
 *    $$\   $$ |$$ |      $$ |  $$\   $$ |     $$ |  $$ |  $$ |$$ |\$$$ |      $$ |  $$ |
 *    \$$$$$$  |$$$$$$$$\ \$$$$$$  |  $$ |   $$$$$$\  $$$$$$  |$$ | \$$ |      $$ |  $$ |
 *     \______/ \________| \______/   \__|   \______| \______/ \__|  \__|      \__|  \__|
 *
 *                              Configure LBIST STCU2 Fault handling
 *****************************************************************************************************************/
/* Configure LBIST recoverable/unrecoverable fault handling  */
STCU2.LBUFM.R = 0x0;                    /* Configure all faults as recoverable in this example */
/*  Configure LBIST error recoverable/unrecoverable fault handling  */
STCU2.ERR_FM.R = 0x0;                   /* Configure all faults as recoverable in this example */
/*  Configure LBIST RESET management register  */
/* Set Global function Reset to occur at end of LBIST 9 */
STCU2.LBRMSW.R = 0x200;
```



* Notes:
- The reset value is chip-specific; see the chapter that describes how modules are configured and connected.x = Undefined at reset.

STCU2_LBRMSW field descriptions

Field	Description
0–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 LBRMSW9	On-Line LBIST Reset Management **NOTE:** In case one of the selected LBIST has this bit set to '1', then only the Global functional reset will be pulsed. 0 Reserved 1 Global functional reset is pulsed at the end of LBIST run

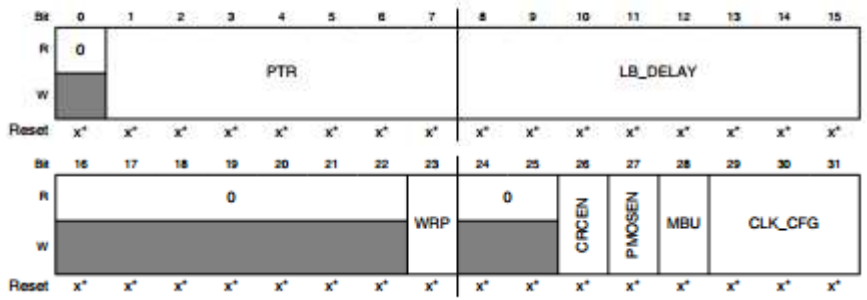
- In our example as LBIST partition 9 is the last partition to be tested, we configure LBMSRW such that a Global functional Reset is generated when LBIST 9 completes its test.

Section I Configure Watchdog and MBIST PMOS Test

```
 *                          Configure Watchdog Time out and  PMOS MBIST test
 ***************************************************************************************************************************/

/* Set Watchdog timeout Config */
523 STCU2.WDG.R = 0xFFFFFFFF;                  /* Watchdog time out to Max value */

/* Configure pointer to point to first MBIST partition and configure PMOST Test                                          */
#ifdef DIAGNOSTIC
STCU2.CFG.R = 0x10000010;       /*MBIST PMOS TEST is enabled for The Full online test sequence*/
#endif

#ifdef KEYOFF
531 STCU2.CFG.R = 0x10000000;       /*MBIST PMOS TEST is disabled for The Full online test sequence*/
#endif
```

MBIST Type	STCU2_CFG[PMOSEN]	STCU2_CFG[MBU]	MBIST Algorithm	Internal Nodes/ Circuits Additionally Covered by Test	Usecase	Test Cycle
Full Test	1	0	Test memory using all built in algorithms. Open PMOS algorithm included	Address Decoder and Bitcell as well as resistive defects in the address decoder	Used by FSL production test. Recommended to use in the field for MBIST online self-test.	About 3 to 4x more than Auto Test
Reduced Test	0	0	Test memory using all built in algorithms except the open PMOS algorithm.	Address Decoder and Bitcell	Recommended to use in the field for MBIST online self-test if the Full Test takes too long.	About 10% less than Full Test
Auto Test	x	1	A smaller set of algorithms which target latent defect mechanisms.	Latent defects such as NBTI of the PMOS transistors in the bitcells	Recommended to use for MBIST offline self-test as an optimum balance of test time vs. fault coverage.	About 3 to 4x faster than Full Test

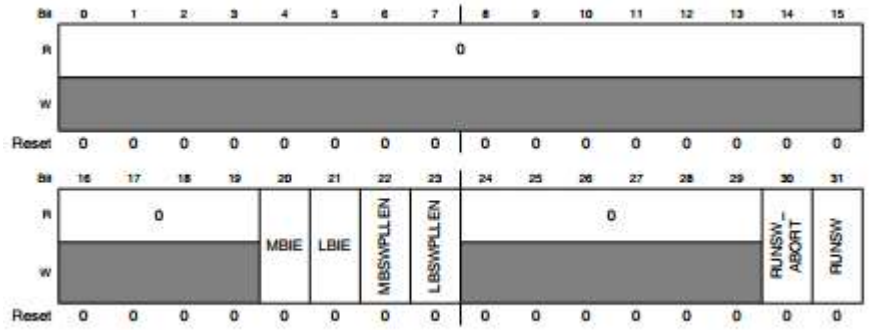
Address: 0h base + Ch offset = Ch



- For our example we have configured the watchdog timeout value to be the maximum. This watchdog value defines the timeout of the execution run
- The CFG PTR file is configured to point to the first MBIST partition in our example.
- The PMOSEN field is set to provide either the full or reduced test (Auto is used by the off-line BIST)

Section J Start the Test

```
/*****************************************************************************************************************
 *     $$$$$$\  $$$$$$$$\  $$$$$$\ $$$$$$$$\ $$$$$$\  $$$$$$\  $$\   $$\          $$$$$\
 *    $$  __$$\ $$  _____|$$  __$$\\__$$  __|\_$$  _|$$  __$$\ $$$\  $$ |         \__$$ |
 *    $$ /  \__|$$ |      $$ /  \__|  $$ |     $$ |  $$ /  $$ |$$$$\ $$ |            $$ |
 *    \$$$$$$\  $$$$$\    $$ |        $$ |     $$ |  $$ |  $$ |$$ $$\$$ |            $$ |
 *     \____$$\ $$  __|   $$ |        $$ |     $$ |  $$ |  $$ |$$ \$$$$ |      $$\   $$ |
 *    $$\   $$ |$$ |      $$ |  $$\   $$ |     $$ |  $$ |  $$ |$$ |\$$$ |      $$ |  $$ |
 *    \$$$$$$  |$$$$$$$$\ \$$$$$$  |  $$ |   $$$$$$\  $$$$$$  |$$ | \$$ |      \$$$$$$  |
 *     \______/ \________| \______/   \__|   \______| \______/ \__| \__|       \______/
 *
 *                                                  Execute Online BIST
 *****************************************************************************************************************/


/* Start the BIST sequence */
STCU2.RUNSW.R = 0x00000301;

while(STCU2.RUNSW.B.RUNSW == 1);
```

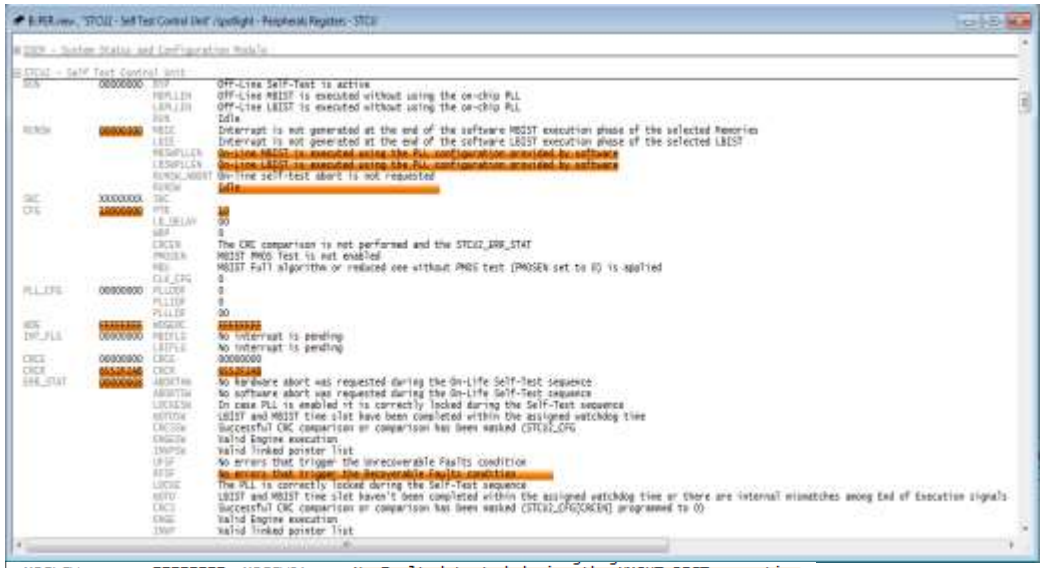


STCU2_RUNSW field descriptions

22 MBSWPLLEN	On-Line MBIST with PLL Enabled 0 On-Line MBIST is executed without using the on-chip PLL. 1 On-Line MBIST is executed using the PLL configuration provided by software. STCU2 does not take the PLL control but monitors the PLL lock signal to check if PLL is working correctly.

- The MBIST and LBIST tests are configured to be executed using the on-chip PLL
- The RUNSW bit field is set to start the Online –self test procedure

Result Checking



MBSLSW	FFFFFFFF	MBSSW31	No Fault detected during the NMCUT BIST execution
		MBSSW30	No Fault detected during the NMCUT BIST execution
		MBSSW29	No Fault detected during the NMCUT BIST execution
		MBSSW28	No Fault detected during the NMCUT BIST execution
		MBSSW27	No Fault detected during the NMCUT BIST execution
		MBSSW26	No Fault detected during the NMCUT BIST execution
		MBSSW25	No Fault detected during the NMCUT BIST execution
		MBSSW24	No Fault detected during the NMCUT BIST execution
		MBSSW23	No Fault detected during the NMCUT BIST execution
		MBSSW22	No Fault detected during the NMCUT BIST execution
		MBSSW21	No Fault detected during the NMCUT BIST execution
		MBSSW20	No Fault detected during the NMCUT BIST execution
		MBSSW19	No Fault detected during the NMCUT BIST execution
		MBSSW18	No Fault detected during the NMCUT BIST execution
		MBSSW17	No Fault detected during the NMCUT BIST execution
		MBSSW16	No Fault detected during the NMCUT BIST execution
		MBSSW15	No Fault detected during the NMCUT BIST execution
		MBSSW14	No Fault detected during the NMCUT BIST execution
		MBSSW13	No Fault detected during the NMCUT BIST execution
		MBSSW12	No Fault detected during the NMCUT BIST execution
		MBSSW11	No Fault detected during the NMCUT BIST execution
		MBSSW10	No Fault detected during the NMCUT BIST execution
		MBSSW9	No Fault detected during the NMCUT BIST execution
		MBSSW8	No Fault detected during the NMCUT BIST execution
		MBSSW7	No Fault detected during the NMCUT BIST execution
		MBSSW6	No Fault detected during the NMCUT BIST execution
		MBSSW5	No Fault detected during the NMCUT BIST execution
		MBSSW4	No Fault detected during the NMCUT BIST execution
		MBSSW3	No Fault detected during the NMCUT BIST execution
		MBSSW2	No Fault detected during the NMCUT BIST execution
		MBSSW1	No Fault detected during the NMCUT BIST execution
		MBSSW0	No Fault detected during the NMCUT BIST execution
MBSMSW	FFFFFFFF	MBSSW63	No Fault detected during the NMCUT BIST execution
		MBSSW62	No Fault detected during the NMCUT BIST execution

- Press the attach then break button
- Observation of the STCU2 registers will display the results of the BIST tests
- ERR_STAT displays the status flags for internal fault conditions that occurred during configuration or Self-Testing
- LBSSW Register shows the Successfully executed LBIST tests
- LBESW shows which LBIST tests have completed
- MBS[L/M/H]SW display the off-line MBIST results. MBE[L/M/H]SW show the completion status

MISR Matching

B::PER.view , "STCU2 - Self Test Control Unit" /spotlight - Peripherals Registers - STCU

		MBUFM64	Recoverable Fault mapping
LB_CTRL0	01035507	CSM	Sequential mode
		PTR	01
		PRPGEN	Default LBIST value of the PRPG is used during LBIST run
		SHS	Shift at 1/4 rate (bist_clk)
		SCEN_OFF	5 delay cycles
		SCEN_ON	5 delay cycles
		PFT	Apply Flush Test Patterns
		CWS	controller waits 7 shift cycles for capture to finish
LB_PCS0	00003380	PCS	00003380
LB_PRPGL0	FFFFFFFF	PRPGx	FFFFFFFF
LB_PRPGH0	00000000	PRPGx	00000000
LB_MISREL0	8BCA2E09	MISREx	8BCA2E09
LB_MISREH0	A9B0733C	MISREx	A9B0733C
LB_MISRRL0	8BCA2E09	MISRRx	8BCA2E09
LB_MISRRH0	A9B0733C	MISRRx	A9B0733C
LB_MISRELSW0	E29FCE36	MISRESWx	E29FCE36
LB_MISREHSW0	50EA7D46	MISRESWx	50EA7D46
LB_MISRRLSW0	E29FCE36	MISRRSWx	E29FCE36
LB_MISRRHSW0	50EA7D46	MISRRSWx	50EA7D46
LB_CTRL1	02035507	CSM	Sequential mode
		PTR	02
		PRPGEN	Default LBIST value of the PRPG is used during LBIST run
		SHS	Shift at 1/4 rate (bist_clk)
		SCEN_OFF	5 delay cycles
		SCEN_ON	5 delay cycles
		PFT	Apply Flush Test Patterns
		CWS	controller waits 7 shift cycles for capture to finish
LB_PCS1	00002780	PCS	00002780
LB_PRPGL1	FFFFFFFF	PRPGx	FFFFFFFF
LB_PRPGH1	00000000	PRPGx	00000000
LB_MISREL1	3E7EB360	MISREx	3E7EB360
LB_MISREH1	93067DD8	MISREx	93067DD8
LB_MISRRL1	3E7EB360	MISRRx	3E7EB360
LB_MISRRH1	93067DD8	MISRRx	93067DD8
LB_MISRELSW1	91150D56	MISRESWx	91150D56
LB_MISREHSW1	B8D517BE	MISRESWx	B8D517BE
LB_MISRRLSW1	91150D56	MISRRSWx	91150D56
LB_MISRRHSW1	B8D517BE	MISRRSWx	B8D517BE
LB_CTRL2	03035507	CSM	Sequential mode
		PTR	03
		PRPGEN	Default LBIST value of the PRPG is used during LBIST run
		SHS	Shift at 1/4 rate (bist_clk)
		SCEN_OFF	5 delay cycles
		SCEN_ON	5 delay cycles
		PFT	Apply Flush Test Patterns
		CWS	controller waits 7 shift cycles for capture to finish
LB_PCS2	00003700	PCS	00003700

- Scroll down to the LB_CTRL registers. We can see here that the returned MISR values match the expected MISR values

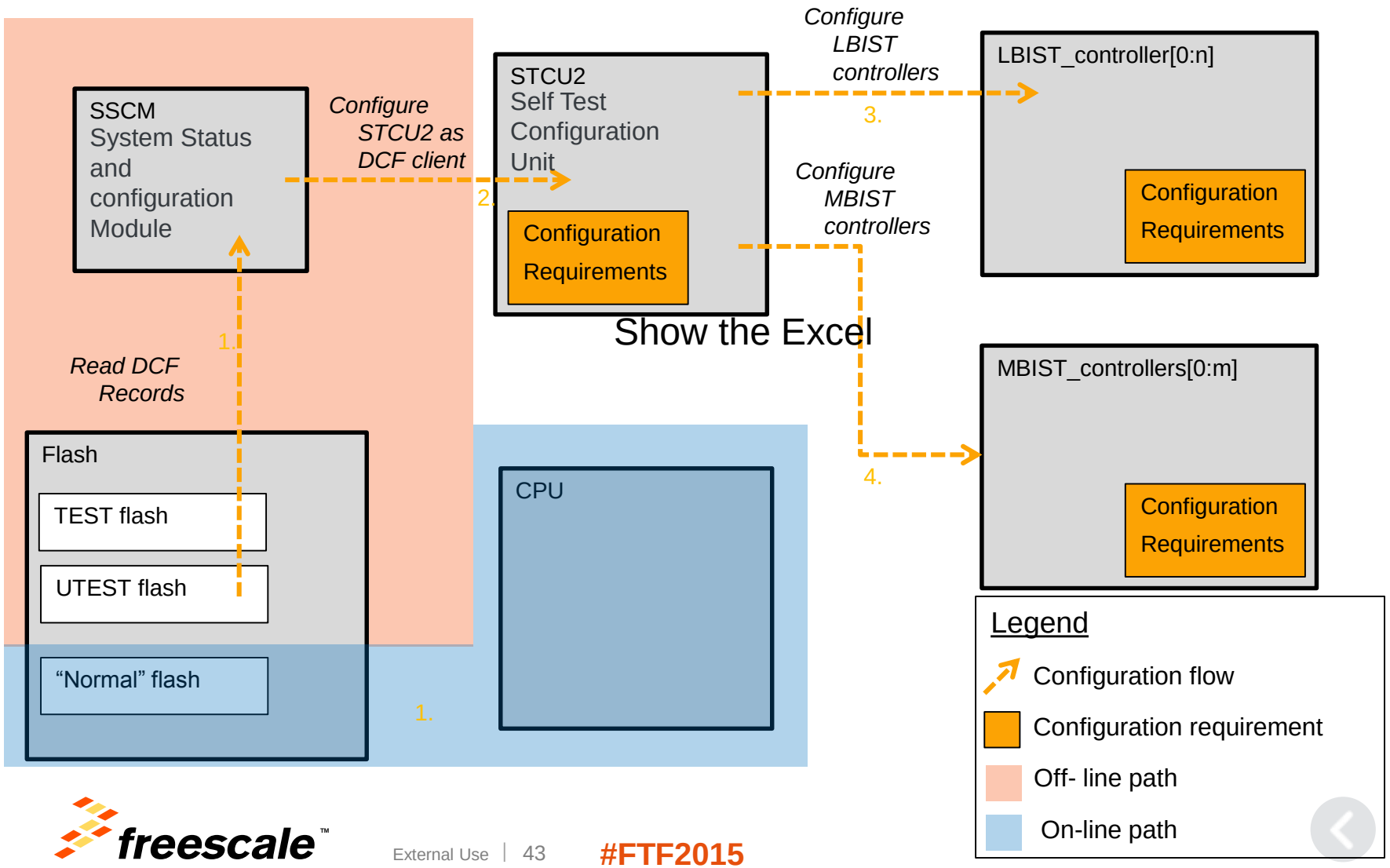
Exercise

- Change Test to full diagnostic sequence using pre-processor directive
- Set MBIST sequence to be fully concurrent. See MBIST CTRL Register on RM page 4564)
- Change the LB_MISRELSW, and LB_MISREHSW values to be 0x0 for LBIST Partition 9 – Lines 485 and 486 of main.c
- Recompile
- Execute test

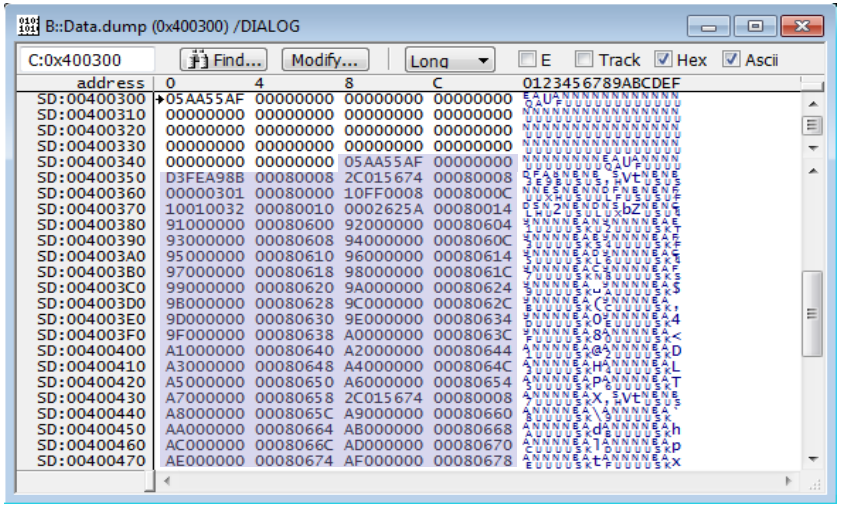
Off-line/Start-up Example

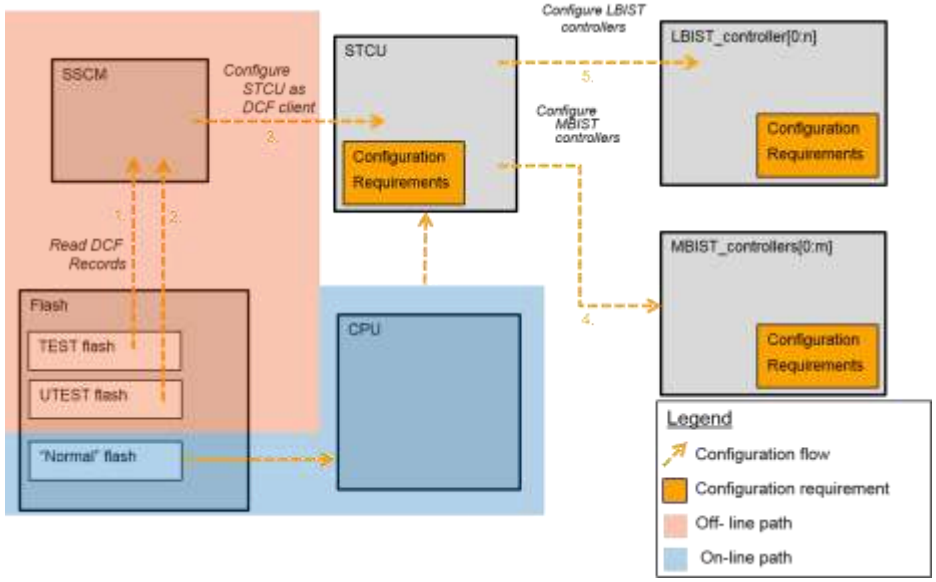
BIST
configuration flow

STCU2: Self Test Configuration Unit
SSCM: System Status and Configuration Module



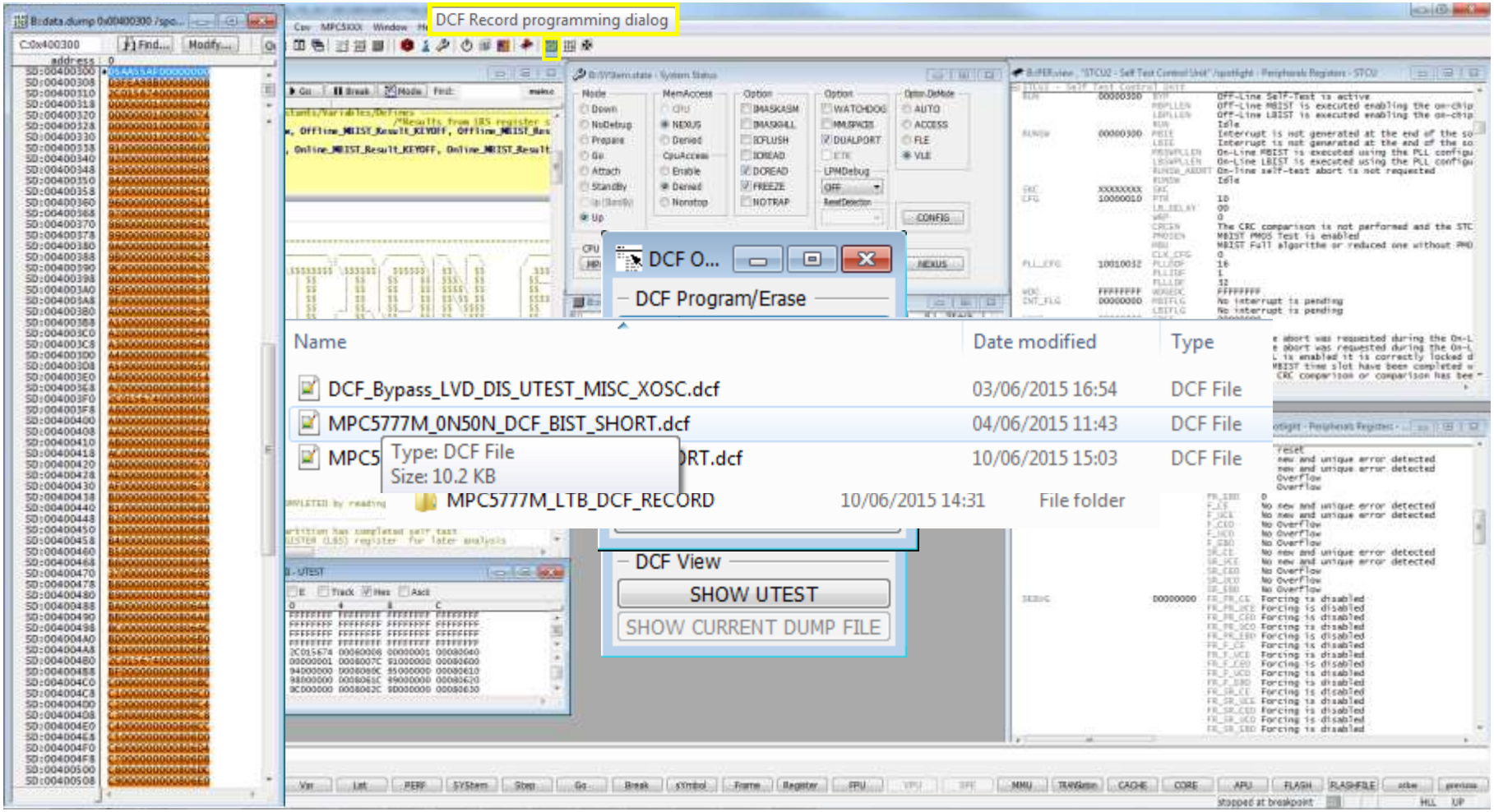
STCU2 DCF Configuration for Off-Line BIST



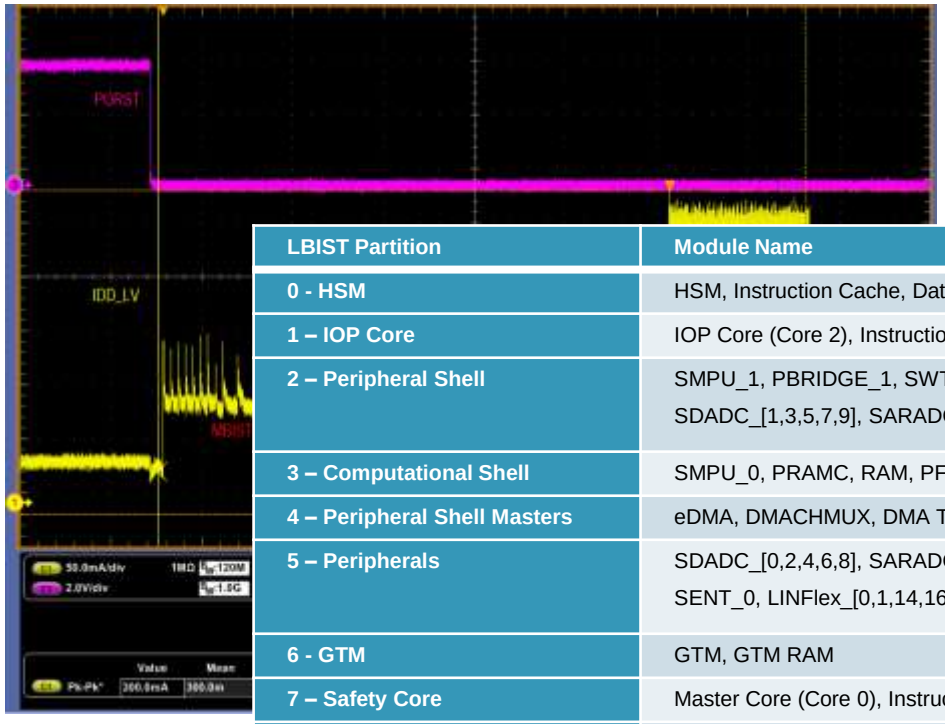


- The Device Configuration Format (DCF) is a mechanism to configure specific registers during system boot and to set up an initial configuration for the device after reset or start up
- In order to configure the self-test to run at start-up a DCF configuration must be written and programmed into the UTEST memory. The DCF records will be processed by the SSCM at start-up and written to the STCU2. The STCU2 will then execute the configured self-test
- A DCF record is 64-bit wide data field in TEST or UTEST Flash that contains 32-bit data that is written to DCF clients, address information and check bits
- A DCF client is a module whose registers can be written by DCF record
- Each write to the STCU2 has a corresponding DCF record entry

Programming Off-line STCU2 Configuration to UTEST



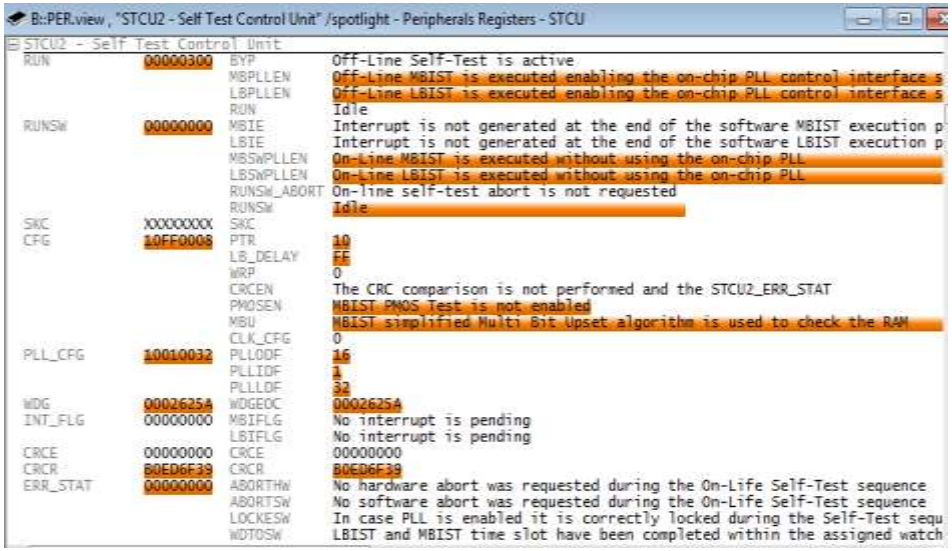
Off-Line DCF Short Example



LBIST Partition	Module Name
0 - HSM	HSM, Instruction Cache, Data Cache
1 – IOP Core	IOP Core (Core 2), Instruction Cache, STM_2, SWT_2
2 – Peripheral Shell	SMPU_1, PBRIDGE_1, SWT_3, LINFLEX[2 - 15], SENT_1, PSI5_1, IIC_1, DSPI_[2,3,5], SDADC_[1,3,5,7,9], SARADC_[1-3, 5-10], ALL CMU's
3 – Computational Shell	SMPU_0, PRAMC, RAM, PFLASHC, Flash Memory, RAM Overlay
4 – Peripheral Shell Masters	eDMA, DMACHMUX, DMA TCD RAM, FEC, FlexRay, INTC, FCCU
5 – Peripherals	SDADC_[0,2,4,6,8], SARADC_[0,4,B], BAR, CRC_0, DSPI_[0,1,4,6,12], IIC_0, PSI5_0, SENT_0, LINFlex_[0,1,14,16], CAN, PBRIDGE_0, MEMU, WKPU, NAR, SPU, PIT
6 - GTM	GTM, GTM RAM
7 – Safety Core	Master Core (Core 0), Instruction Cache, Data cache, STM_0, SWT_0
8 – Checker Core	Checker core (Core 0 - Checker), RCCU and delay logic
9 – Computational Core	Computational Core (Core 1), Instruction cache, Data cache, STM_1, SWT_1

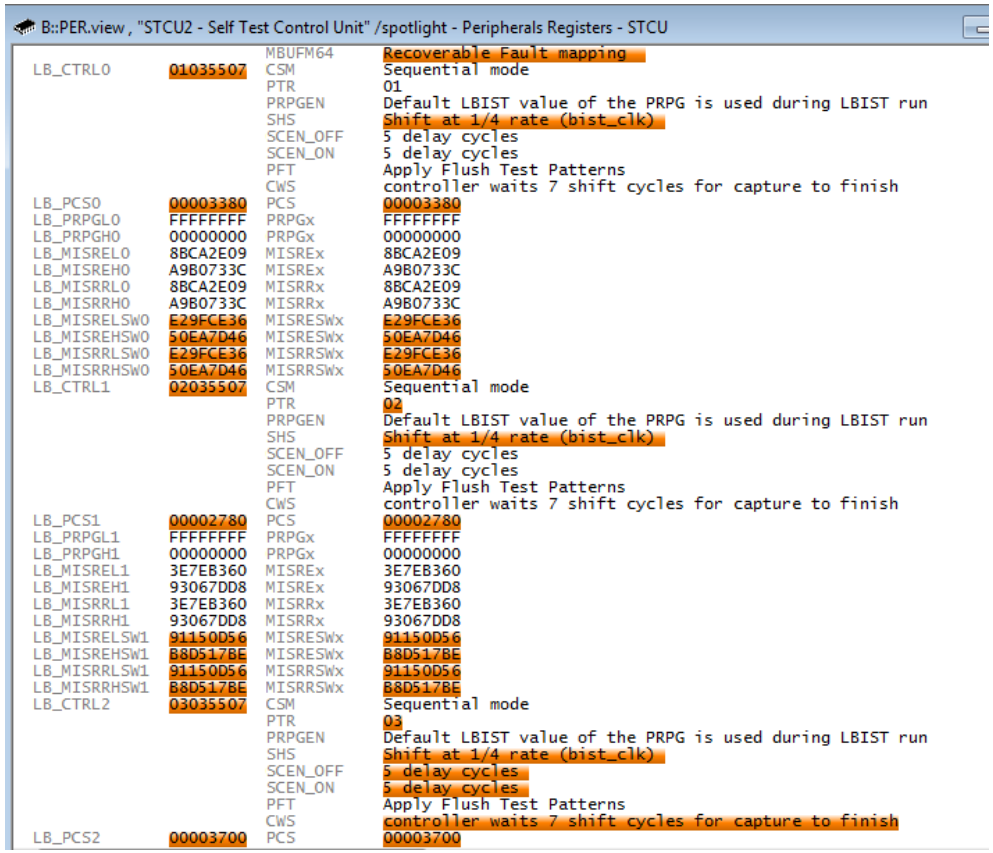
- This configuration is designed to be run at start-up with a minimal execution time. It uses 90% coverage of the LBIST partitions. In this example only the 3 LBIST partitions that are
ot of the
be
titions are
ecuted with
e 77
roups. To
sumption
romise,
ps are
two groups are tested sequentially
- LBIST: Partitions 0, 1 and 3 are tested sequentially with 90% coverage

Offline Result Checking



- Press the attach then break button 
- Observation of the STCU2 registers will display the results of the BIST tests
- ERR_STAT displays the status flags for internal fault conditions that occurred during configuration or Self-Testing
- LBS Register shows the Successfully executed LBIST tests
- LBE shows which LBIST tests have completed
- MBSL/M/H display the off-line MBIST results. MBEL/M/H the completion status

MISR Matching



- Scroll down to the LB_CTRL registers. We can see here that the returned MISR values match the expected MISR values
- Not required to check this in application

Summary

- Built-In Self-Testing (BIST) is a mechanism provided on Freescale MPC57XX devices in order to detect the accumulation of latent faults, a requirement of the ISO26262 standard which defines functional safety for automotive equipment.
- MPC5777M can be configured to carry out Self Test of both memory and logic at either start up or during run time.
- Freescale provides example configurations with Specific ASIL coverage for MPC5777M.
- Other MPC5777x devices will be provided with a standard offline configuration programmed into the UTEST memory, and an application note detailing the configurations available for On-line BIST

www.Freescale.com