NXP Semiconductors
Application Note

Document Number: AN5131
Rev. 1, 02/2019

Using the Built-in Self-Test (BIST) on the MPC5777M

Contents

1. Introduction

The MPC5777M device is targeted at automotive powertrain controller and chassis control applications including Diesel, Gasoline and Hybrid combustion systems. These applications may require that the device be considered functionally safe. With this in mind the MPC5777M has been designed to achieve ISO26262 ASIL-D Compliance.

The ISO26262 standard defines functional safety for automotive equipment. A requirement of the standard is to detect the accumulation of latent defects. To meet this requirement the MPC5777M has the ability to execute Built-In Self-Test (BIST) procedures. The BIST is intended to identify latent system faults within the MCU. The BIST can be performed on the device's embedded memories and logic.

Additionally, NXP has created its "SafeAssure" Functional Safety program to reduce the development effort required by customers to meet ISO26262. As part of this program NXP provides an MPC5777M safety manual to advice users on how to configure the MPC5777M to obtain ISO26262 ASIL D compliance.

2. Objective

This application note provides an introduction to BIST on the MPC5777M and explains how to configure and use the BIST features of the MPC5777M. After reading this applications note the user should:

- Understand the BIST features that are available on the MPC5777M
- Understand the difference between MBIST and LBIST
- Understand the difference between online and offline BIST
- Be able to develop application strategies for deploying online and offline BIST testing
- Understand what is included in each BIST partition on the MPC5777M
- Understand when offline BIST takes place
- Be able to develop a specified configuration for offline BIST testing and understand how to use DCF clients to achieve this
- Be able to invoke the desired online BIST sequence using the STCU2

The application note also provides four example online and offline Self-Test configurations that can be directly implemented by the user. The examples are provided in the software package that accompanies this document and are also explained in detail. NXP guarantees the functionality of these four configurations and highly recommends users utilize these in their application.

To aid in understanding of this document and software package the reader should obtain the MPC5777M reference and Safety manuals from the NXP website.

3. 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.

3.1. MBIST and LBIST

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 SRAM and peripheral memories on the MCU, such as SRAM memory contained in the peripheral modules such as FlexRay or MCAN. For MBIST testing purposes each of the memories is segmented into individual MBIST *partitions*. The segmentation of the memories is discussed in Chapter 7 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 user recognizable logic modules (CPU, XBAR, MCAN etc.). LBIST must be configured to test partitions sequentially.

The LBIST and MBIST execution is managed by the Self-Test Control Unit (STCU2).

3.2. Off-Line and Online BIST overview

It is anticipated that the MPC5777M automotive user application would require two standard configurations for BIST testing.

- **Vehicle start-up** – test as much as possible within the vehicle start-up time constraints – known as offline.
- **Vehicle Shutdown or diagnostic testing** – maximum test coverage, no time constraints when vehicle powered down– known as online

For off-line testing the BIST tests can be configured to execute every time the MCU boots or gets a destructive reset. This procedure is performed while the MCU is powered and held in reset. In this mode, user configurable Device Configuration Format (DCF) Records that are stored in the one-time programmable UTEST memory are loaded at start-up by the System Status and Configuration Module (SSCM) module into the STCU2 to configure the self-test procedure. When the BIST executes successfully in offline mode the device exits reset and the application software is executed. Configuring the device for offline testing is discussed in section 9-Offline BIST procedure to configure by DCF.

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.

The intended usage of on-line BIST is to execute a test of memory and modules that are critical to the start-up of the application. This helps to minimize the startup time of the MCU. Executing a full BIST at start-up will exceed the needs of many users. 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.

4. Self-Test Control Unit

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, figure 1 shows the STCU structure and connections. To control offline BIST testing the STCU2 operates in conjunction with the System Status and Configuration Module (SSCM) module which has the ability load the Self-test parameters from flash memory automatically during the boot phase. The SSCM interface is able only to write the configuration parameters and start the Self-Test execution once after the STCU2 global reset has been applied.

To configure online BIST testing via software an IPS interface allows access by the device CPU(s) to the STCU2 registers. Using this interface software can configure the STCU2 registers for execution of the online tests, or check the results of the offline tests. The configuration flow for offline and online testing is shown in figure 2.

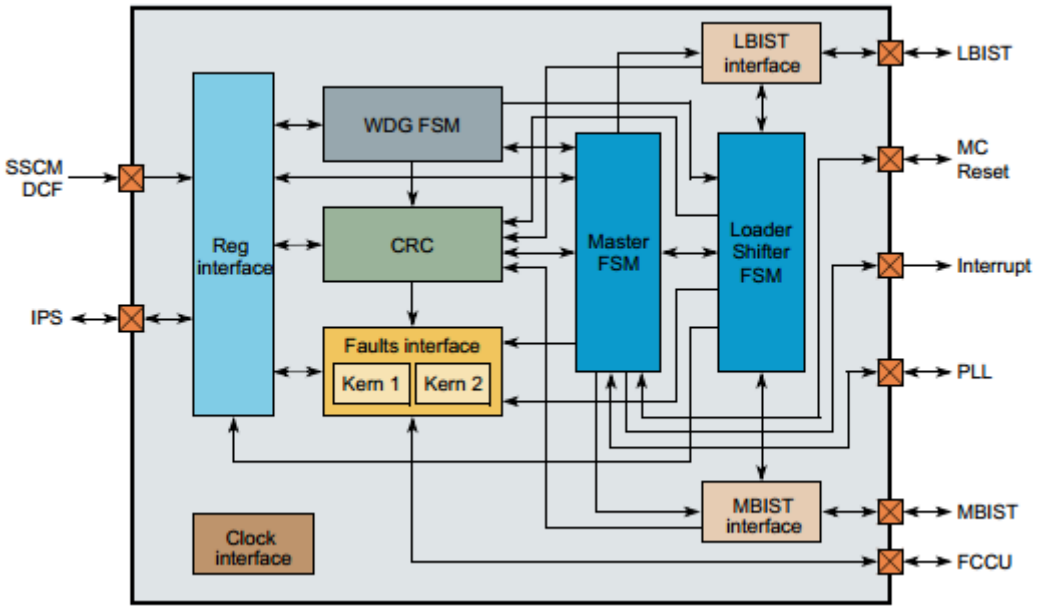


Figure 1. STCU2 block diagram

The STCU2 module includes the following sub-modules:

- Reg interface: This module includes the registers, the security key logic, the IPS and SSCM DCF bus interface
- CRC: This module is the core of the Self-Checking architecture of the STCU2. It samples a set of selected internal signals when STCU2 is running.
- Fault interface: This module collects the error conditions related to L/MBIST execution and STCU2 internal failures and, depending on the Unrecoverable or Recoverable (UF/RF) configuration of each one of them, sets the global STCU2 UF or RF flag. It also manages the UF/RF fault lines and the Set/Clear injection mechanism provided by the Fault Collection and Control Unit (FCCU). To improve the intrinsic safety of this critical logic, the generation logic is duplicated.
- Clock Block: This module manages the internal and the L/MBIST TCK clock pre-scalar, the internal Clock Gating Power Saving and the wake-up clock feature.
- WDG FSM: This module includes two different Watchdogs. The first one is hard coded and is used to auto lock the STCU2 access forcing a reset condition on the double security key registers while the second one has a double functionality. After a reset event initializes the STCU2, it is used as Hard Coded Watchdog time-out while during L/MBIST run it is used as programmable Watchdog timer to check the L/MBIST have been completed in the assigned to L/MBIST time slot.
- STCU2 Master FSM: This module includes the main FSM of the STCU2 used to coordinate and schedule all the operations performed during the Self-Test Sequence.
- STCU2 Load Shifter FSM: This module includes the common shifter register and the related state machine used to program the L/MBIST registers and read back the data to be checked at the end of each Test operation.

- LBIST interface: This module includes the interface between the device's LBIST engines and the STCU2 controllers.
- MBIST interface: This module includes the interface between the MBIST controller and STCU2 controllers.

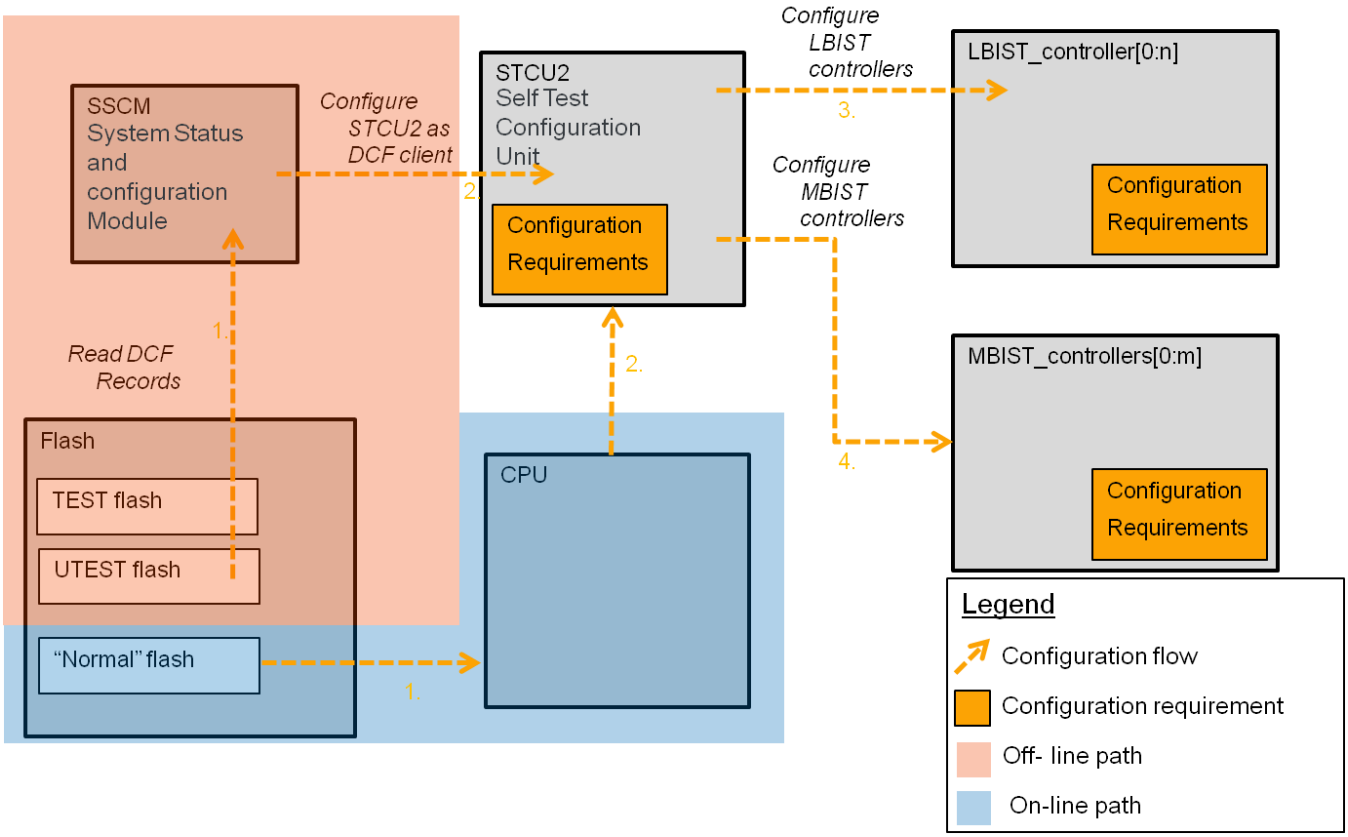


Figure 2. Configuration flow for BIST testing

5. MBIST and LBIST testing and partitions

This section details the partitioning of the memory and logic on the MPC5777M. This information will allow users to identify the partitions that they want to run for their chosen Self-Test requirements.

This section will also provide information on the levels of test coverage that can be achieved for each LBIST partition and how this is determined. The options for testing of the MBIST partitions are also introduced.

5.1. LBIST testing

Logic Built-In Self-Test (LBIST) is implemented by 10 LBIST controllers which operate independently on each LBIST partition. This independence is needed to meet safety requirements regarding the independence and diversity of replicated IP and also helps to avoid exceeding power limits. Each of the LBIST controllers is connected independently to the STCU2 as shown in figure 2. The STCU2 must be

configured to run all LBIST controllers sequentially. This sequence is defined in the STCU2 by the user configuration.

5.2. LBIST partitions

The MPC5777M implements 10 different LBIST partitions. The partitions are numbered from 0 to 9. Each LBIST partition is contained in a separate physical partition with its own control logic. The percentage of the logic gates that will be tested during the BIST, generally known as the test coverage level, is determined by the number of patterns that are run on each partition. As the number of gates varies in each partition the number of patterns required to achieve a determined level of coverage will also vary. The test coverage achieved for the examples in the application note is provided with each individual example. Reducing the number of patterns run on a partition reduces the time taken to complete the BIST for that partition but also in turn it would reduce the level of coverage level. The pattern count for each partition is configured within the STCU2.

The self-test of each logic partition is deemed to be successful by checking the resulting Multiple Input Signature Register (MISR) value of the LBIST against a 64-bit expected MISR value. 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 unless there is a fault in the tested logic. The STCU2 provides registers for the result and expected MISR. The expected signatures for the MPC5777M are provided in the "MPC5777M_0N50N_LBIST_STCU_CONFIGRATION" spreadsheet that can be found in the attachments tab of this document. It is important to note that the expected MISR values and other LBIST configuration of the device may change with any modification to the internal logic design of the device. The expected values provided with this application note apply only to the 0N50N maskset of the MPC5777M device.

LBIST Partitions

Table 1 shown below lists the distribution of the IP blocks among the LBIST partitions:

Table 1. LBIST partitions

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 logic, RAM Overlay, AMU
4 – Peripheral Shell Masters	eDMA, DMACHMUX, DMA TCD RAM, FEC, FlexRay, INTC, FCCU

LBIST Partition	Module Name
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

NOTE: Please note that when testing partition 3 that incorporates the flash memory and flash controller the flash must be in an idle state.

Modules that do not implement any safety function or that are essential components for the self-test execution are excluded from LBIST partitions. These are provided in table 2 below.

Table 2. Hardware not covered by LBIST

Functional Group	Module
System	MC_CGM, MC_ME, MC_PCU, MC_RGM, SSCM, SIUL2, IOMUX, STCU2, PLL'S
Clocking	XOSC, IRCOSC
Power	PMC, ADC Bandgap reference, TSENS
Debug	JTAC, LFAST, SIPI, DCI,JTAGM

5.3. MBIST testing

The MBIST is executed by a single MBIST controller that is programmed via the STCU2. The MBIST engine controls the self-test of multiple partitions. The MPC5777M memory is split into 78 different MBIST partitions which are numbered 0 – 77.

The MBIST controller has three types of tests it can apply when running MBIST that allow for different coverage levels. The Full Test Mode tests memory using all algorithms including the open PMOS algorithm. The Reduced Test Mode tests memory using all algorithms except the open PMOS algorithm. The Auto Test Mode uses a smaller set of algorithms which has lower coverage.

The Auto Test mode is designed to quickly test the RAM with good fault coverage, so this mode is recommended for the off-line BIST as it has an optimum balance of test time versus fault coverage. The Full Test Mode is comparable to the tests used in the NXP factory to test the RAM. Since NXP has already

tested the parts prior to delivery, the primary concern for the user should be to detect latent defects. The Full Test is recommended for the on-line BIST since time constraints are typically not as critical in the user's on-line use case. The particular fault coverage of each MBIST test mode is shown in figure 3.

Figure 3. MBIST test coverage options

		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

5.4. MBIST partitions

Table 3 below lists the memory partitions and the MBIST controller numbering.

Table 3. MBIST partitions

Block	Function	Size	STCU2 MBIST PARTITION	Start Address	End Address
Safety Core	IMEM	16K	0	0x50000000	0x50003FFF
	DMEM	64K	2	0x50800000	0x5080FFFF
			1		
	ICACHE	16K	4	N/A	
			3		

Block	Function	Size	STCU2 MBIST PARTITION	Start Address	End Address
	DCACHE	4K	6		
			5		
	ITAG		8		
	DTAG		7		
Comp Core	IMEM	16K	9	0x51000000	0x51003FFFF
	DMEM	64K	11		
			10	0x51800000	0x5180FFFF
	ICACHE	16K	13	N/A	
			12		
	DCACHE	4K	15		
			14		
	ITAG		17		
	DTAG		16		
Overlay	Overlay	16K	19	0x0D000000	0x0D003FFFF
			18		
IOP Core	IMEM	16K	21	0x52000000	0x52003FFF
			20		
	DMEM	64K	23	0x52800000	0x5280FFFF
			22		
	ICACHE	8K	27	N/A	
			26		
			25		
			24		
	ITAG		28		
HSM	PRAM	40K	29	N/A	
			31		
			30		
	Icache	4K	33		
			32		
	ITAG		36		
			35		
	C3 MPAES		34		
DMA	DMA		38	N/A	
			37		
FlexRAY	DRAM		40	N/A	
			39		
	LRAM		44		
			43		

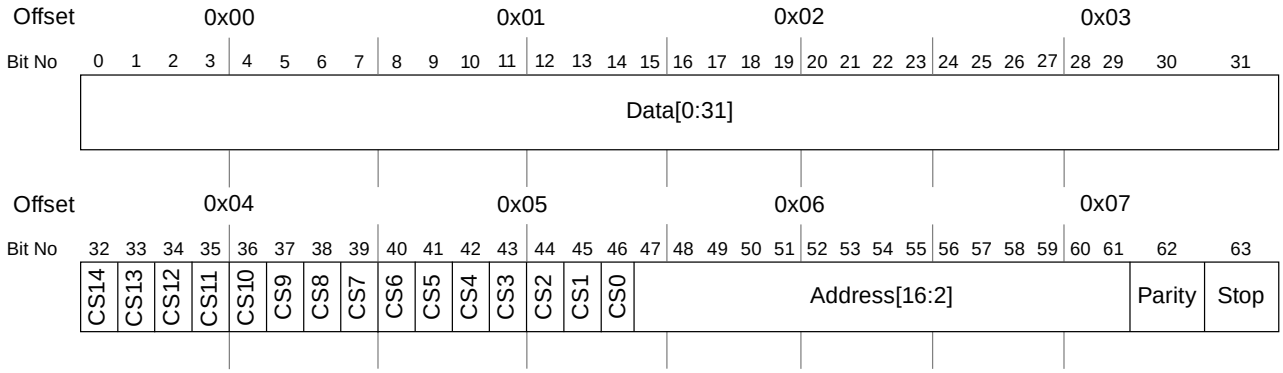
Block	Function	Size	STCU2 MBIST PARTITION	Start Address	End Address
			42		
			41		
TTCAN	TTCAN		45	N/A	
GTM	FIFO		47		
			46	N/A	
	MCS RAM0		50		
			49		
			48		
			53		
			52		
			51		
	MCS RAM1		56		
			55		
			54		
			59		
			58		
			57		
	DPLL RAM1a		61		
	DPLL RAM1b		60		
	DPLL RAM2		62		
System RAM	System RAM	404K	65	0x40010000	0x40065000
			64		
			63		
			68		
			67		
			66		
			69	0x40000000	0x4000FFFF
NAR	NAR		73		
			72	N/A	
			71		
			70		
Reserved	Reserved	32K	74	N/A	
FEC	FIFO (RIF)		75		
	MIB		76		
BAR	BAR (In-Situ)	8K	77		

6. DCF Records, Clients and Configuration

The offline BIST tests are configured using Device Configuration Format (DCF) records that automatically configure the STCU2 to enable and run BIST at start-up or following a destructive reset. This section will initially explain how DCF records operate, before describing how the BIST tasks execute during the MPC5777M Reset sequence.

The DCF is a mechanism to automatically configure specific registers during system boot and to set up an initial configuration for the device after reset or start up. The term DCF client is used to describe a module whose registers can be written by DCF record, e.g. the STCU2 is a DCF client. DCF records are stored in both TEST and UTEST flash. During the boot sequence of the device the SSCM automatically loads the DCF records to the DCF clients.

A DCF record is a 64-bit wide data field made up of the 32 bit data that is to be written to DCF clients, address information and check bits as illustrated in Figure 4.



Field	Name	Description
0–31	Data[0:31]	32 bits of data that is to be written to the DCF Client
32–46	CS*n*	Chip Select *n*. **NOTE:** Only assert one chip per DCF record to select the target module for the DCF client. All other Chip Selects should be negated. 0 Chip Select is asserted 1 Chip Select is negated
47–61	Address[16:2]	Address of the DCF client within the selected module. **NOTE:** Address decoding for DCF clients may not match the standard software address map decoding. Details of DCF Client addresses are defined in each module chapter
62	Parity	Parity Bit for the DCF Record. **NOTE:** This bit is NOT implemented for DCF Client written from UTEST
63	Stop	Stop bit. This bit indicates the end the list of DCF Records. **NOTE:** The Erased state of flash is 0xFFFF_FFFF_FFFF_FFFF. Therefore the list ends with the first unprogrammed double word. This location can be programmed with a new record to extend the list. 0 Not the end of the list. 1 End of the list.

Figure 4. DCF record structure

The DCF records provide an interface that can be used to initialize registers within the selected modules (DCF clients) during system boot. For offline BIST the DCF records are used to configure the registers of the STCU2.

TEST DCF records are developed by the factory and are used mainly to program registers involved in trimming trip points for voltage comparators, adjusting analog to digital voltage supplies, trimming oscillator frequencies, and enabling RAM repair. These TEST DCF records can also be used to write any DCF client. The TEST DCF Records are programmed into TEST flash during production and cannot be modified by the USER. TEST flash is not visible to the user.

UTEST DCF records are programmed into UTEST Flash. Some UTEST DCF records may written by the factory and programmed during production testing, but it is intended that the majority will be created by the user and programmed at the same time application code is programmed into the flash memory. User-supplied UTEST DCF records start at the next location in the UTEST memory map following the UTEST DCF records programmed into the flash memory during factory production. Some of the tasks that the user-defined DCF records can be used for include:

Defining which tests the STCU2 will run during the boot sequence

Configuring the external oscillator

Enabling the lockstep safety core

Configuring the response to low voltage/high voltage detection

Assigning memory blocks as One Time Programmable (OTP),

Assigning flash blocks to be associated with specific password groups

Assigning flash memory blocks to specific tamper detect regions.

The UTEST area is One Time Programmable OTP and as such programmed regions cannot be erased. The DCF records that configure the STCU2 to perform offline BIST are located in the UTEST. Care should be taken by the user to ensure that the configuration created by the customer DCFs is valid, as the OTP nature of the UTEST means that errors cannot be undone.

DCF clients are 32-bit wide hardware registers inside a module that receive and store the data from a DCF record. This stored data is used to initialize registers and to configure features. The STCU2 registers are DCF clients. DCF clients have a default value before any DCF Records and written; and may have special writing constraints; such as ‘Write Once’ or only allow bits to be written from ‘1’ to ‘0’ or vice versa. DCF clients need not implement all 32 bits.

DCF clients may be designated ‘TEST DCF record only’. This means that only DCF Records stored in TEST flash can write to the DCF Client. Please refer to the reference manual for a list of DCF clients and detailed descriptions of the various attributes of the DCF clients.

In the UTEST flash memory the following structure for the DCF records must be present:

The first record must be a start record:

0x00 0:31	0x04 32:63
0x05AA55AF	0x00000000

DCF records containing configuration data must immediately follow the start record with no blank records between -an un-programmed record is interpreted as a stop record and no DCF records following that record are processed.

The end of the configuration records are indicated by the presence of a stop record.

There must never be an un-programmed record in the DCF data structure, as it is interpreted as a stop record and subsequent records are ignored. This allows one to program the records in several sessions, each time appending new records at the end of the list, as shown in Figure 5.

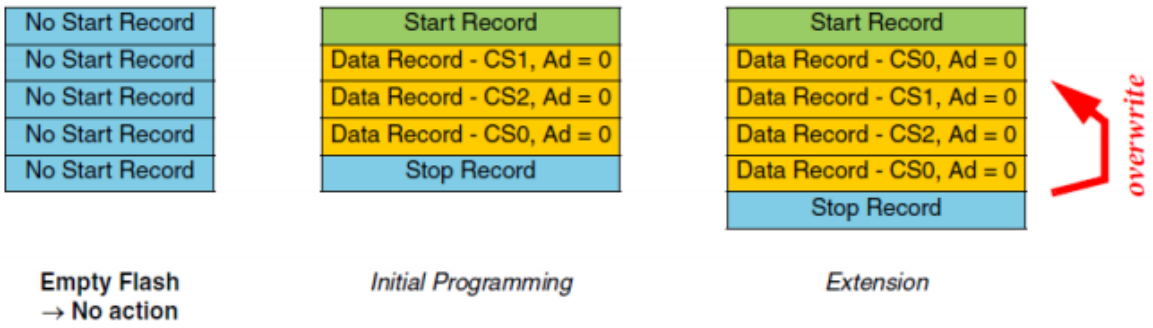


Figure 5. Programming DCF records

It is possible to have more than one DCF record that writes to the same DCF client. In this case, the later record usually overrides a DCF client value set by a previous record. However, not all DCF clients allow overwrites, this depends on the DCF client implementation. Please refer to device's reference manual for details.

There are example UTEST flash programming scripts that configure the STCU2 via DCF provided in the software package accompanying this application note and descriptions are also included in the attachments tab of this document.

6.1. The MPC5777M Reset Sequence and Offline BIST

When the MCU is powered correctly and exits the Power-Up phase the Reset Generation Module (MC_RGM) takes control and manages the reset sequence. The MC_RGM provides a register interface and registers to monitor and control the reset sequence. The reset sequencer is a state machine that controls the different phases (PHASE0, PHASE1, PHASE2, PHASE3, and IDLE) of reset and controls the reset signals generated in the system. Figure 6 shows the reset sequence states.

There are a total of ten different states in the reset sequence. After a power-on reset (PORST) or a destructive reset, the reset sequence starts at PHASE0. The following three phases are used for temporization and setup, flash initialization and configuration. The configuration for the BIST is loaded to the STCU2 by the SSCM module during PHASE3[DEST] using DCF records. Other DCF records are also processed at this time.

After the PHASE3[DEST] completes, the state machine enters the IDLE (DEST) phase where the configured BIST tests are executed. At the end of the offline BIST a functional reset must is triggered, this needs to be specified in the STCU2 configuration. After a functional reset the device proceeds at PHASE1[FUNC]. It is also possible to disable the BIST if it is not required by using a DCF record. In this case the device directly proceeds to phase IDLE[FUNC]. Two active-low reset signals are associated with the internal reset circuitry:

- PORST: Power-on reset is released when the device leaves the POWERUP phase. The signal has a strong pull-down when the device is in the POWERUP state, and a weak pull-down when it is not in this phase, i.e. it has to be pulled up externally in order to bring the device out of reset.
- ESR0: Is released in phase IDLE[FUNC], therefore will remain asserted when the BIST is running.

If a valid DCF STCU2 configuration is present in the UTEST the BIST will be executed at power on and after a destructive reset. The destructive reset could be caused via hardware by asserting the PORST pin externally. The BIST will not execute after a functional reset.

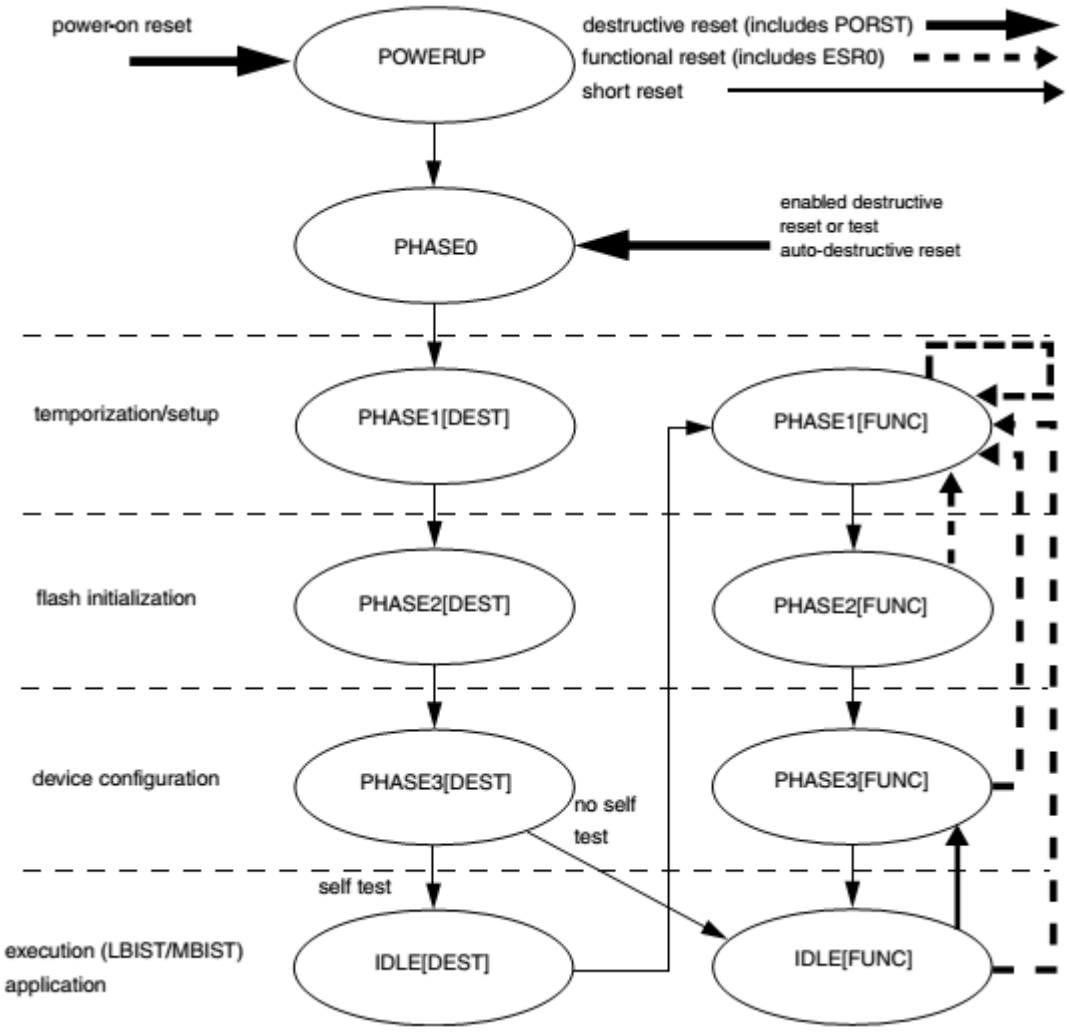


Figure 6. Reset Generation module Reset Sequence

7. STCU2 Unlock and watchdog configuration tasks

Regardless of whether the STCU2 registers are programmed by DCF records during offline testing or whether they are programmed by the core during online testing, the STCU unlock and watchdog mechanisms are the same. The mechanisms are described in this section.

7.1. STCU2 Register write access keys and hardware watchdog timer

The access on the STCU2 registers during the self-test configuration phase is protected by a key mechanism to prevent any unwanted access. To initially access the STCU 2 registers security key one (0xD3FEA98B) must be written to the STCU security code register (STCU2_SKC). This must be immediately followed by writing security key 2 (0x2C015674) to STCU2_SKC. One the STCU2 register access has been unlocked; there is a hardware watchdog timer that limits the time that the STCU2 registers can be accessed after the initial double key sequence.
This feature is intended to prevent unintended access by software during runtime. To prevent the access to the STCU2 registers from being locked, Key 2 must be written to the STCU_SKC register every 30 instructions.

7.2. BIST execution watchdog

The STCU2 has a watchdog timer that is used to ensure that the BIST tests complete in the allocated time span. If the selected LBISTs or MBISTs are not yet completed during the off-line Self-Test assigned time, the current LBISTs or MBISTs execution is interrupted and a failure is flagged into STCU2_ERR_STAT.WDTO and STCU2_MBEL/M/H or STCU2_LBE registers while in case of On-Line Self-Test into STCU2_ERR_STAT.WDTOSW and STCU2_MBELSW/MSW/HSW or STCU2_LBESW.

8. Offline BIST idiosyncrasies

8.1. Offline BIST clocking options

Offline BIST can either be clocked at 16MHz using the onboard Internal Reference Clock (IRC) or run at a maximum of 50MHz by configuring PLL0. For offline BIST the PLL0 is configured by STCU2 registers. The PLL can be selected independently for MBIST and LBIST by using the MBPLLEN and LBPLLEN bit fields in the STCU_RUN register determine whether the PLL0 or IRC is used for the offline BIST. The STCU_PLL_CFG register is used to control the frequency of the tests. The STCU2 will configure the PLL and wait on it to lock before proceeding with the BIST.

8.2. STCU_RUN Register setting in offline mode

The user must avoid setting the RUN bit in the STCU2_RUN register twice via DCF.

8.3. Bypass mode

If the bypass field in the STCU2_RUN register (STCU_RUN[BYP]) is set to 1 then the off-line self test is completely bypassed. If the user writes a DCF record into the UTEST that configures the STCU2 to Bypass the offline Self-Test, the online BIST cannot be re-enabled with a subsequent DCF record. Offline BIST will be permanently disabled.

8.4. Offline Un-testable partitions

LBIST partitions 2, 4 and 5 are not functional during offline BIST with the MPC5777M 050N maskset. The offline BIST examples provided with this application note do not test partitions 2, 4 and 5. These partitions should instead be tested using the online BIST sequence.

9. Offline BIST procedure to configure by DCF

This section details how to use the DCF record to configure the STCU2 to execute the offline BIST tests. Each write to the STCU2 registers requires a DCF record entry in the UTEST area. There are two working examples provided in the software package that accompanies this application note. The examples are also in the attachments tab of this application note. The descriptions in table 4 below can be used in conjunction with the examples to fully understand the procedure. NXP guarantees the functionality of the two configurations and highly recommends users utilize these in their application.

Table 4. Offline BIST programming procedure

Step – DCF Entries		Relevant Register	Notes
DCF Setup	Create a DCF Start Record Entry	First erased location after UTEST Start Address: UTEST START ADDR: 0x400300 DCF Start Record: Data:0x05AA55AF00000000	A valid DCF start record must be written to the UTEST area (This might already be in place if amending the existing DCF config for STCU2 to existing config)
Unlock STCU2	Unlock the STCU2 registers for offline self-test.	Security Key Code register STCU2_SKC = 0xD3FEA98B STCU2_SKC = 0x2C015674	Write offline Key1/Key2 sequence to the Security Key Code register. IMPORTANT: Key 2 must be periodically written to the STCU2 to keep the STCU2 unlocked for configuration

Step – DCF Entries		Relevant Register	Notes
Configure FCCU fault reaction	Configure the FCCU reaction if a MBIST of LBIST fault is found during BIST	LBIST/MBIST Unrecoverable Fault mapping STCU2_LBUFM STCU2_MBUFM	Specify if a fault in each logic or memory partition is recoverable or unrecoverable. 0= Recoverable (default) 1= Unrecoverable
Configure Internal STCU2 fault reaction	Setup how the STCU2 should behave when it detects an BIST fault	Error fault model handling STCU2_ERR_FM	Specify if STCU2 configuration or engine faults generate a recoverable or unrecoverable fault 0 = Recoverable 1 = Unrecoverable
Configure MBIST	Write the control register to configure each MBIST partition that is to be run	STCU2_MB_CTRLn	CSM: Define if the next MBIST will be run concurrently or sequentially with this partition PTR: Provide the number of the MBIST or LBIST partition that is to be run after this one. Write 0x7F if this is the final MBIST partition to be run.
Configure LBIST	Write the control register to configure each MBIST partition that is to be run	STCU2_LB_CTRLn	CSM: Set to 0 for sequential operation of LBIST module PTR: Provide the number of the LBIST partition that is to be run after this one. Write 0x7F if this is the final LBIST partition to be run. PRPGEN: Not used SHS: Shift speed, Configure as shown in attachments to configure Shift Speed to 50MHz SCEN_OFF & SCEN_ON: Set to 5 cycles for this MCU PFT: Always lave as 0 to apply the flush test patterns CWS: Always set to 0x7 on this MCU.

Step – DCF Entries		Relevant Register	Notes
	Set the number of patterns to be run for each LBIST partition that is to be executed	STCU2_LB_PCSn	The number of patterns to be run. Determines the level of coverage that will achieved. PCS value is provided by NXP and is detailed in attachments
	Set the expected MISR values for the LBIST	STCU2_MISRELn STCU2_MISREHn	Provides the expected MISR result for this partition. Will be compared against the actual result in STCU_MISRRLn and STCU_MISRRHn The expected MISR are provided by NXP in the attachments
Set watch dog time	Configure watch dog time out	STCU2_WDG	The value provided in this register provides a time allocation that the BIST must complete in, or a watchdog error will be raised
Configure STCU2 Parameters	Write the STCU2 Configuration register	STCU2_CFG	PTR: Set this field to the first BIST partition that should be executed. If both LBIST and MBIST will be run the MBIST must be executed first. LB_DELAY: Not used WRP: Optionally lock specific registers in the STCU2 after Self-Test CRCEN: Optionally enable the CRC check PMOSEN: Enable in depth memory testing, includes address decode logic MBU: Set to enable simpler Multi Bit upset test for RAM (faster) CLK_CFG: Clock divider for STCU2 – Use examples provided

Step – DCF Entries		Relevant Register	Notes
Configure PLL	Write the PLL settings	STCU2_PLL_CFG	Write the PLL settings to configure the speed of the PLL. Max 50MHz. PLL Input clock is the 16MHz IRC PLL FREQ = (16MHz*PLLLDF)/(PLLIDF*PLLODF)
Configure Run register	Write the Run register to start the Self-Test and select clock options	STCU2_RUN	Set the Run bit. The RUN bit although set now will not take effect until all DCF records have been processed. The MBPLLEN and LBPLLEN fields determine if the PLL should be used for the MBIST and/or LBIST. If not selected then the 16MHz IRC will be used.

10. Online BIST procedure

This section will explain the procedures that the user software must take during runtime to configure online BIST to execute. It is expected that the online BIST tests will be used during shutdown of the MCU as this will likely be the only timeframe in MCU operation where the required device re-configuration and MBIST execution time can be accommodated. Two online BIST examples are provided in the software package. NXP guarantees the functionality of the two configurations and highly recommends users utilize these in their application.

10.1. Preparing the MCU for Online Self-test

Prior to running the online self-test the user code must ensure that the application task is no longer dependent on the MCU. Communications and control should undergo a controlled shut down. Only one core should be used to configure the self-test procedure. The cores that will not be used for this task should be halted via a mode entry change, including the HSM. This can be a single mode entry change that is used to disable the cores and also configure the clocks as described in Clock configuration for Online BIST testing. All peripherals should be programmed as disabled.

10.2. Clock configuration for Online BIST testing

The online testing requires that PLL1 is operating at a maximum output of 200MHz to ensure successful testing of all LBIST partitions. For most users this will be a reduction from the 300MHz used when running the system normally. The PLL frequency should be changed and a mode entry executed to ensure that the update takes place. System clock dividers should also be configured to match the new configuration. The PLL and system clock change sequence from 300MHz to 200MHz are shown in the online BIST example in the supplied software package. This sequence also demonstrates how to disable the peripheral modules. BIST operation at 200MHz ensures that the expected results, as configured in the

MISR register for each partition will match those provided within this applications note across all test conditions. Within the STCU2 the 200MHz PLL frequency will be divided down to 50 MHz for the LBIST, while the MBIST runs at full PLL frequency. Alternatively the 16MHz IRC can be set as the system clock and all system clock dividers can be configured as divide by 1. Both online BIST examples provided in this applications note will show BIST at speed using the PLL.

10.3. On-Line Self-Test STCU2 configuration

In order to configure the self-test to successfully run when the MCU is operational user software must configure the system and the STCU2. The steps of this procedure are described in table 5 below. The process steps are similar to that of the offline test procedure; however the order of the configuration steps is now critical as the configuration is loaded in real-time to the STCU2. Also some of the registers used are replicated for online and offline mode. The appropriate registers must be used for the applicable mode, in this case online mode. The sequences given in table 5 below can be compared to the online BIST examples present in the MAIN.c file of the accompanying software project.

Table 5. On-line self test example sequence

Section in Main.c	Task	Relevant registers in STCU2	Notes
A	Check for offline self-test completion	STCU2.MBEL STCU2.MBEM STCU2.MBEH STCU2.MBSL STCU2.MBSM STCU2.MBSH	This is a code trap to determine if an offline BIST sequence has completed, and to store the results for later analysis. Application code in the field would likely do this straight after boot.
A	Check for Online LBIST completion	STCU2.LBSSW STCU2.LBESW	Code to detect if an online LBIST sequence has completed and subsequently caused a functional RESET. The user should store and handle results as applicable
B	Initialize device	N/A	Standard device initialization function is called
C	Configure device for BIST	User dependent	Ensure that the application task is no longer dependent on the MCU. Communications and control should undergo a controlled shut down. Set PLL1 output to a maximum of 200MHz, with the clock divider for the PBRIDGE configured to

Section in Main.c	Task	Relevant registers in STCU2	Notes
			provide 50MHz since the PBRIDGE clocks the STCU2 modules
D	Unlock the STCU2 registers for online Self-Test.	STCU2_SKU = 0x753F924E STCU2_SKU = 0x8AC06DB1	Write online Key1/Key2 sequence to the Security Key Code register. IMPORTANT: Key 2 must be written again after every 30 instructions to the STCU2 to keep the STCU2 unlocked.
E	Write the control register to configure each MBIST partition that is to be run	STCU2_MB_CTRLn STCU2_SKC	CSM: Define if the next MBIST will be run concurrently or sequentially with this partition PTR: Provide the number of the MBIST or LBIST partition that is to be run after this one. Write 0x7F if this is the final MBIST partition to be run. The Security Key 2 is periodically written to the STCU2 to prevent timeout while programming the STCU2 registers
F	Configure the FCCU reaction if an MBIST found during BIST	STCU2_MBUFM	Specify if a fault in each partition is recoverable or unrecoverable.
G	Write the control register to configure each LBIST partition that is to be run	STCU2_LB_CTRLn	CSM: Set to 0 to configure LBIST Module to run sequentially. PTR: Provide the number of the LBIST partition that is to be run after this one. Write 0x7F is this is the final LBIST partition to be run. PRPGEN: Not used SHS: Shift speed, Configure as shown in appendixes to Shift Speed to 50MHz SCEN_OFF & SCEN_ON: Set to 5 cycles for this MCU PFT: Always lave as 0 to apply the flush test patterns CWS: Always set to 0x7 on this MCU.
G	Set the number of patterns to be run	STCU2_LB_PCSn	The number of patterns to be run. Determines the level of coverage that will achieved.

Section in Main.c	Task	Relevant registers in STCU2	Notes
	for Each LBIST partition that is to be executed		The PCS value is provided by NXP and is detailed in the attachment MPC5777M_0N50N_LBIST_STCU_CONFIGRATION.xls
G	Set the expected MISR for the LBIST	STCU2_LB_MISRELSWn STCU2_LB_MISREHSWn	Provides the expected MISR result for this partition. Will be compared against the actual result in STCU_LB_MISRRLSWn and STCU_LB_MISRRHSWn The expected MISR are provided by NXP in the MPC5777M_0N50N_LBIST_STCU_CONFIGRATION.xls
H	Configure the FCCU reaction if a MBIST of LBIST fault is found during BIST	STCU2_LBUFM	Specify if a fault in each partition is recoverable or unrecoverable.
H	Setup how the STCU2 should behave when it detects an BIST error	STCU2_ERR_FM	Covers errors that are not related to an individual partition and defines if they are recoverable or unrecoverable
H	Set Global function reset to occur at the end of the LBIST	STCU2_LBRMSW	Set all required fields to reset the MCU after the LBIST has completed its final partition. Testing and resetting of individual partitions without a whole MCU reset is not supported on the MPC5777M.
I	Configure watch dog time out	STCU2_WDG	The value provided in this register provides a time out period that will be allowed to occur if there is no BIST activity or progress. After this time the MCU will be reset.

Section in Main.c	Task	Relevant registers in STCU2	Notes
I	Write the STCU2 Configuration register	STCU2_CFG	PTR: Set this field to the first LBIST partition that should be executed. LB_DELAY: Unused WRP: Optionally lock specific registers in the STCU2 after offline Self-Test to prevent unintended activation of BIST CRCEN: Optionally enable the CRC check PMOSEN: If MBU bitfield is set to 0, then this bitfield determines whether Reduced or full MBIST algorithms are carried out MBU: Set to optionally enable simpler Multi Bit upset test for RAM (faster) CLK_CFG: Clock divider for STCU2 – Use examples provided
J	Execute the online LBIST test	STCU2_RUNSW	This register is used to configure whether the BIST tests are run using the onchip PLL or IRC, to enable interrupts at the end of BIST, and to set the RUNSW bit which executes the online MBIST test.

11. Handling BIST faults

11.1. Fault handling overview

A full explanation and example of the fault handling mechanisms provided on the MPC5777M is beyond the scope of this application note. However the intention of this section is to identify the mechanisms that the user must employ when dealing with the BIST results. The user should also consult the MPC5777M safety manual for recommendations.

In general the device should be configured such that if there is an LBIST failure, or MBIST detects uncorrectable failures, the STCU2 will cause a destructive reset, causing execution of the self-test again. This is to ensure that a self-test, which fails only due to a transient error, will not block device usage. If several self-tests fail in a row, the destructive reset escalation will activate and hold the MCU in reset. Full details of the reset escalation mechanism and configuration are provided in the Reset Generation Module chapter of the MPC5777M reference manual.

The user must configure the Fault Configuration and Control Unit (FCCU) to correctly handle faults identified by BIST testing. The FCCU offers a hardware mechanism to aggregate error notifications and

a configurable means to bring the device to a safe state. No CPU intervention is required for collection and control operation. Error indications are passed from the individual hardware components to the FCCU where the appropriate action is decided (according to the FCCU configuration). To configure the FCCU to deal with faults originating from the STCU2 the user should configure the relevant FCCU Channel inputs as detailed in table 6 below.

Table 6. MPC5777M FCCU BIST related inputs

FCCU Channel	Failure	Failure Description	Default reaction configuration after Power on Reset	Recommended recovery mechanism
6	STCU2_UF	STCU2 unrecoverable fault indication during self-test. It is simultaneously reported to the MC_RGM which will initiate a destructive reset in case of an offline self-test. In the STCU2, self-test failures can be configured as recoverable or unrecoverable faults which would then lead to assertion of the corresponding fault line to the FCCU. Fault is cleared by clearing STCU2_ERR_STAT[UFSF] followed by clearing the FCCU channel status, FCCU_RF_S*n*[RFS*m*].	Destructive reset by MC_RGM directly, during offline self-test	POR, destructive reset. During an offline test, assertion of this fault will automatically cause a destructive reset request by MC_RGM
7	STCU2_RF	STCU2 recoverable fault indication during self-test. In the STCU2, self-test failures can be configured as recoverable or unrecoverable faults which would then lead to assertion of the corresponding fault line to the FCCU. Fault is cleared by clearing. STCU2_ERR_STAT[RFSF] followed by clearing the FCCU channel status, FCCU_RF_S*n*[RFS*m*].	No reaction	
8	STCU2_LMBIST_USR_ERR	Activation of LBIST or MBIST control during application mode. Fault is cleared by clearing the FCCU channel status, FCCU_RF_S*n*[RFS*m*]. The status will clear if the fault is not persistent.		Long Functional Reset to reset STCU2, POR, Destructive Reset

If MBIST detects correctable failures, user software must decide whether to continue or halt execution. The MBIST may detect and report two (or more) Single Bit Errors (SBEs) occurring in multiple test passes instead of one Multiple Bit Error (MBE).

User Software should determine if two or more errors reported by the MBIST as SBEs combine to create an uncorrectable error by examining the entries in the System RAM Memory Management Unit (MEMU) error reporting tables. The System RAM MEMU error reporting tables are used by the STCU2 to report errors found during MBIST. The instance is populated by the STCU2 with the failing correctable and uncorrectable addresses identified during MBIST. If several entries exist for the same address with different bit numbers, this data word actually has an MBE instead of the several SBEs discovered by the MBIST. Full configuration details for the MEMU module can be found in the MPC5777M reference manual.

11.2. Offline BIST

After start-up and before the safety application starts, application software shall confirm that all offline LBISTs and MBISTs finished successfully, and no critical failure is flagged. The critical failures may include LBIST failures, MBIST MBEs, MBIST SBEs exceeding the maximum tolerated number (<= 8 due to MEMU buffer size) and self-test failures.

In the event that a critical error has been detected as defined in the STCU2 fault reaction registers it is possible that the MCU will be held in reset by the FCCU and user code will not execute. In this case the user must ensure the FCCU is configured to communicate the status information via the error out pins or have an alternative means of external monitoring in place.

After the offline or online BIST has executed the user software should check the status of the BIST before continuing. In this situation it is highly likely that the requirement will be to save the status of the BIST and shut down.

Table 7 below shows the procedure for checking the results of the self-test:

Table 7. Example procedure to check BIST results

Step		Relevant Register	Notes
Check if BIST has RUN	Read the RGM module to determine if BIST reset has occurred	MC_RGM_FES	Check the ST_Done bit field. Will be set if self-test has run This can be checked along with other flags in the destructive reset register to determine what type of reset caused the self-test to run
Check if BIST was successful	Read STCU2 Error Status Registers	STCU2_ERR_STAT STCU2_ERR_FM	Check if there were any errors when running the BIST.

Step		Relevant Register	Notes
			These fields can also be used to debug when developing the STCU2 configuration.
Check what Partitions completed	Read the completion status of each BIST partition	STCU2_MBEL STCU2_MBEM STCU2_MBEH STCU2_LBE	If the error status register indicated that there was a recoverable or unrecoverable faults the MBIST and LBIST end registers can be read to check if the BIST partition completed.
Check what Partitions passed/failed	Read the success status of each BIST partition	STCU2_MBSL STCU2_MBSM STCU2_MBSH STCU2_LBS	If the partitions completed testing these MBIST and LBIST status registers can be used to check which, if any partitions failed

12. Example configurations

This section explains the configuration of the four BIST examples provided in the application note software examples and provided in the attachments. Two online mode examples and two offline mode examples are provided. The software and DCF configuration required to execute each of these examples is provided along with this applications note.

Table 8. Example BIST configurations for MPC5777M

Self-Test Level	Application Mode	Clock	STCU Config	Execution 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	Autotest
Short	KEY ON (STARTUP)	PLL0 50MHz	Loaded from Flash by SSCM	15ms (9.7ms LB, 6.3ms MB)	90% of 3 partitions	Autotest
Medium	KEY OFF (SHUTDOWN)	PLL1 Full Freq	IOP using IPS I/F	48ms (29ms LB, 19ms MB)	90% of all partitions	Full w/o PMOS open
Long (DEBUG)	Board-level Diagnostics	PLL1 Full Freq	NEXUS JTAG I/F	680ms (660ms LB, 20ms MB)	94% of all partitions	Full Set

12.1. Example 1 ultra short offline BIST

12.1.1. Overview of the configuration

This configuration is designed to be run at start-up with a minimal execution time. It uses 80% coverage of the LBIST partitions to offer a shorter execution time. In this example only the LBIST partitions and MBIST partitions that are considered critical to the boot of the application are configured to be executed. The DCF configuration files for this example are included with comments in the attachments file MPC5777M_050N_DCF_BIST_ULTRA_SHORT.dcf, and the Flash programming scripts are provided in the supplied software package

12.1.2. Configuration

MBIST: 77 partitions are executed with the simplified Multi –bit upset algorithm. The 77 partitions are split into two groups. To provide the best current consumption versus execution time compromise, the partitions within the groups are tested concurrently, but the two groups are tested sequentially.

LBIST: Partitions 0, 1 and 3 are tested sequentially with 80% coverage.

Current and Time Profile

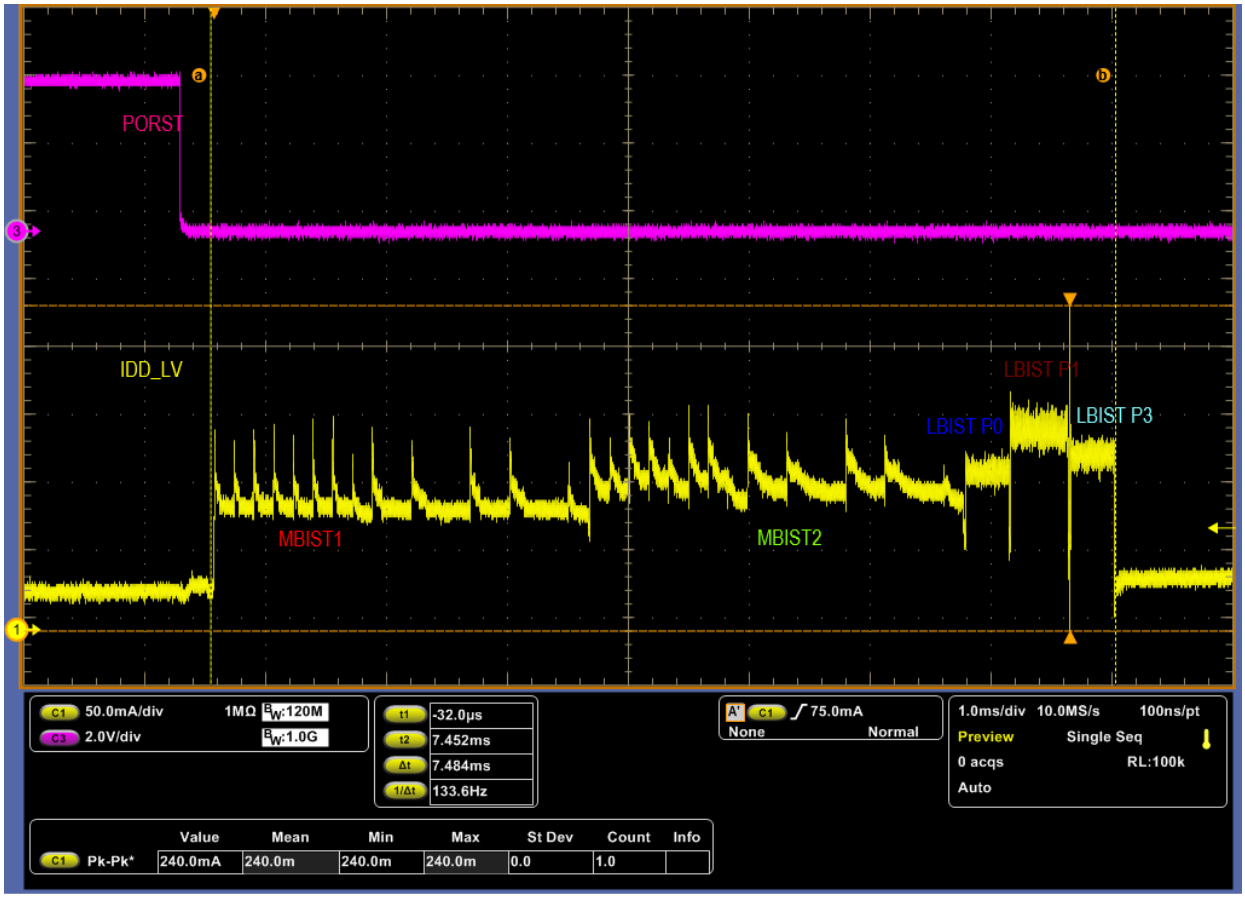


Figure 7. Ultra short offline BIST current profile

12.2. Example 2 short offline BIST

12.2.1. Overview of the configuration

This configuration is designed to be run at start-up with a minimal execution time. It gives 90% coverage of the LBIST partitions. In this example only the LBIST partitions that are considered critical to the boot of the application are configured to be executed, and all MBIST partitions are executed. The DCF configuration for this example is detailed in the attachment MPC5777M_0N50N_DCF_BIST_SHORT.dcf, and the Flash programming scripts are provided in the supplied software package

12.2.2. Configuration

MBIST: 77 partitions are executed with the simplified Multi –bit upset algorithm. The 77 partitions are split into two groups. To provide the best current consumption versus execution time compromise, the partitions within the groups are tested concurrently, but the two groups are tested sequentially.

LBIST: Partitions 0, 1 and 3 are tested sequentially with 90% coverage

Current and Time Profile

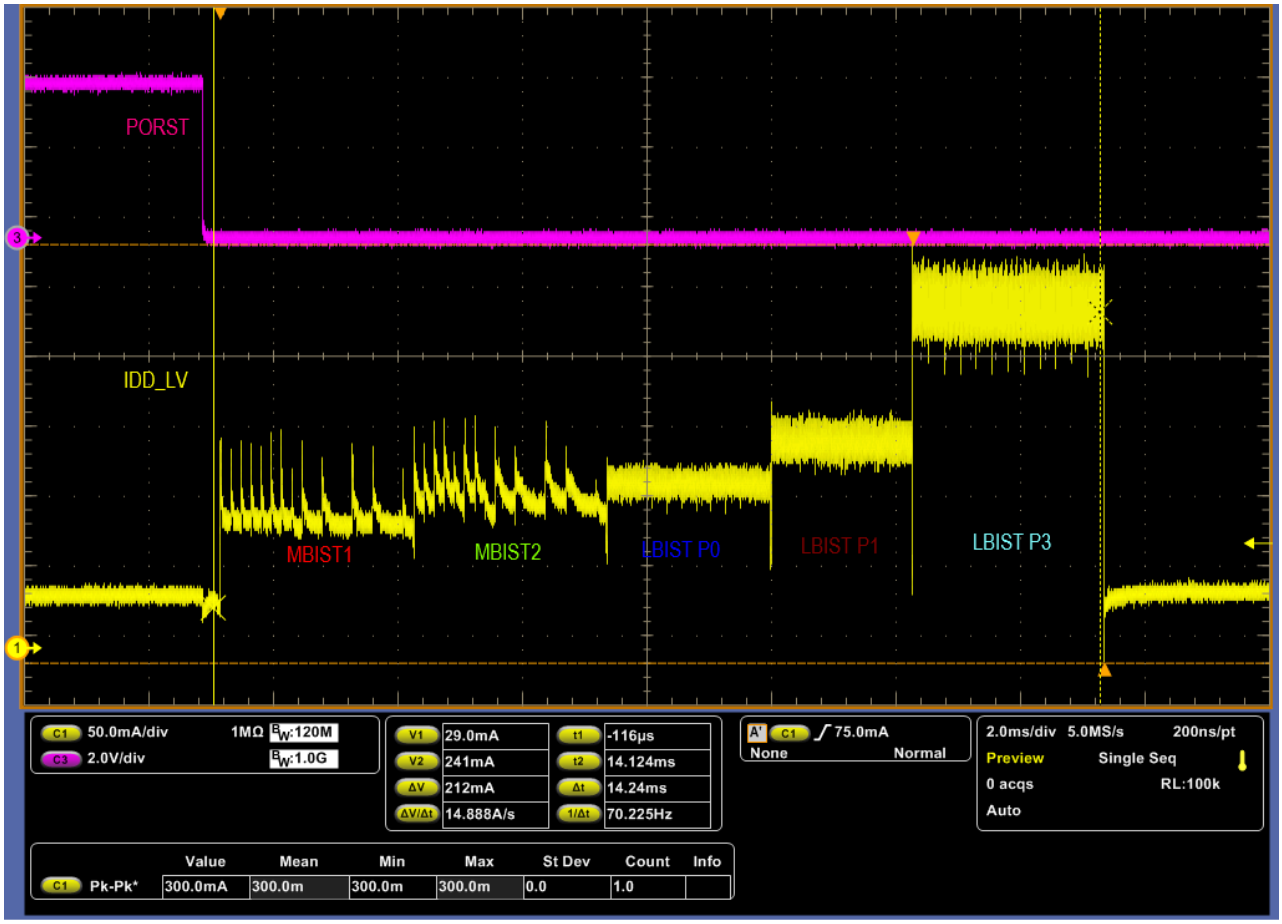


Figure 8. Short Offline BIST current and time profile

12.3. Example 3 Medium – KEY OFF

12.3.1. Overview of the configuration

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 configuration is in the Main.c of the software supplied with this application note and in the attachments tab.

12.3.2. Device configuration

Prior to executing this BIST routine the user application should ensure that the application tasks are completed. A mode change sequence that reconfigures the clocks to the correct configuration is completed prior to the BIST. 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.

Current and Time Profile

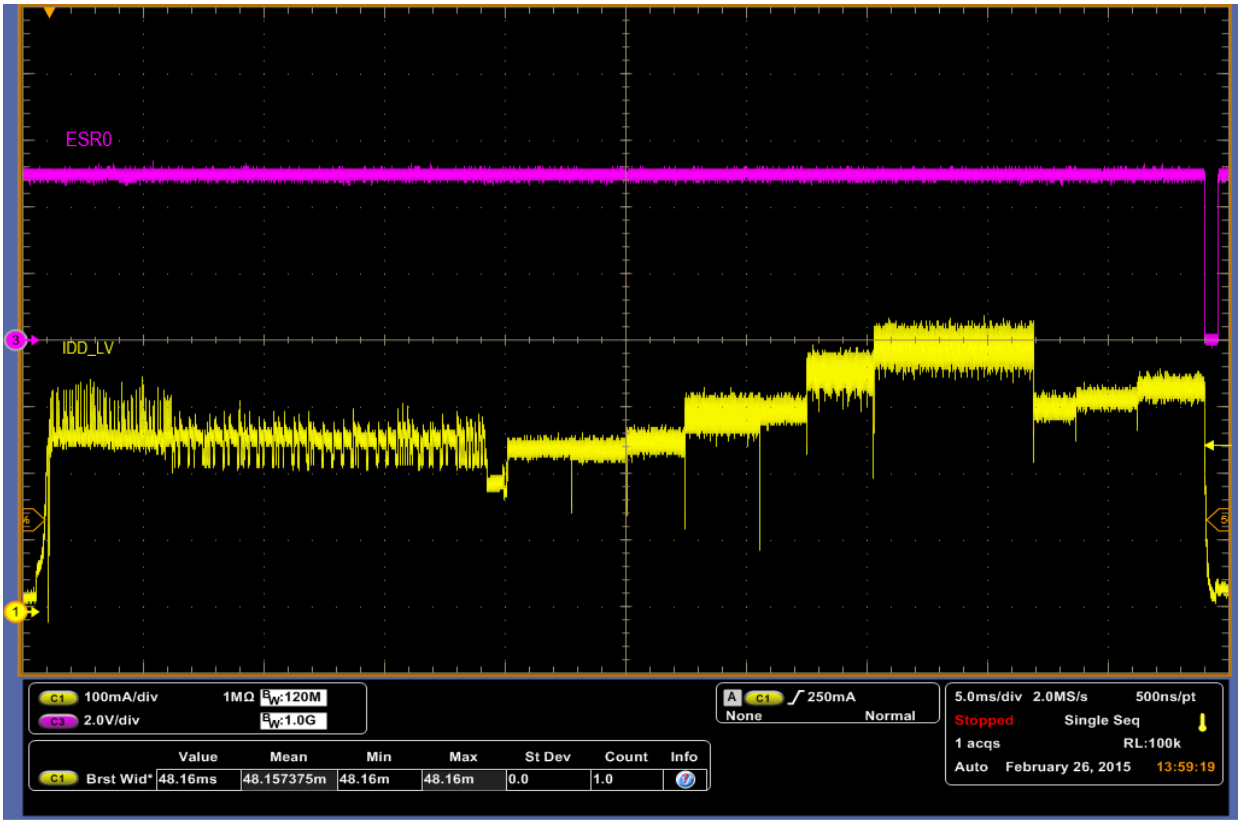


Figure 9. Key-off Online BIST Current and time profile

12.4. Example 4 Long diagnostic example

12.4.1. Overview of the configuration

This is an example configuration for a diagnostic BIST. This BIST routine is intended to be run for diagnostic purposes in ECU fault finding. In this configuration all LBIST and MBIST partitions are tested. This configuration is provided as C code in the Main.c of the software supplied with this application note and also in the attachments tab.

12.4.2. Device configuration

Prior to executing this BIST routine the user application should ensure that the application tasks are completed. A mode change sequence that reconfigures the clocks to the correct configuration is completed prior to the BIST. 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 94% coverage. All MBIST partitions are tested with the full MBIST algorithm and with the PMOSEN test which provides additional coverage of decoders

Current and Time Profile

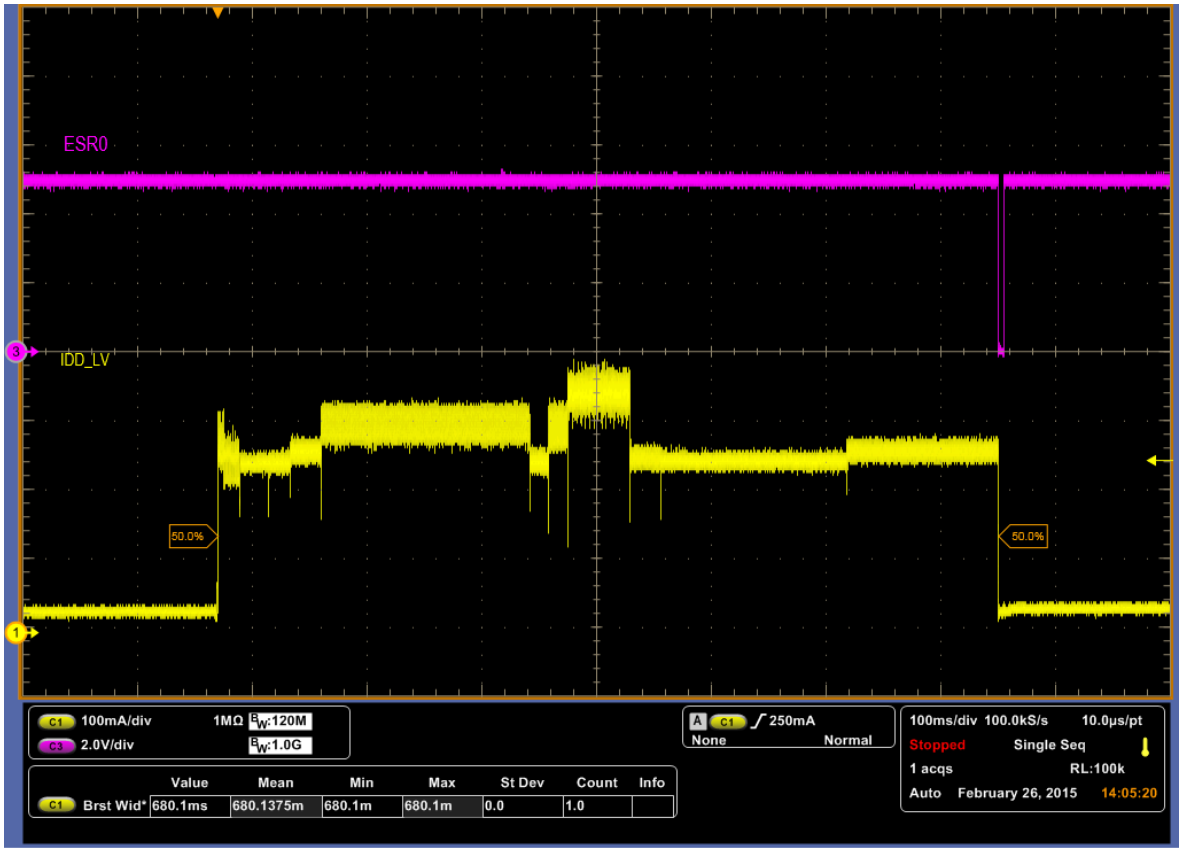


Figure 10. Long Diagnostic Online BIST current and time profile

13. Using the Software Package

The application note is supplied with a software pack designed to help the user to quickly implement BIST procedures on the MPC5777M. The software package contains a project for the Greenhills MULTI Integrated Development Environment, in conjunction with debugger scripts developed for the Lauterbach TRACE32 tool. If the user has different compiler or debug tools the user can extract the code files and flash programming scripts to use with the development tools of their choice.

13.1. Package overview

The compressed project has the structure shown in figure 11 below. The root folder level contains the scripts to configure and execute the Lauterbach TRACE32 debugger. Within the root folder the folder MPC5777M_GHS_SC_BIST contains the GHS MULTI IDE project, and the folder MPC5777M_LTB_DCF_RECORD contains the scripts to program the BIST DCF records into the device UTEST Flash

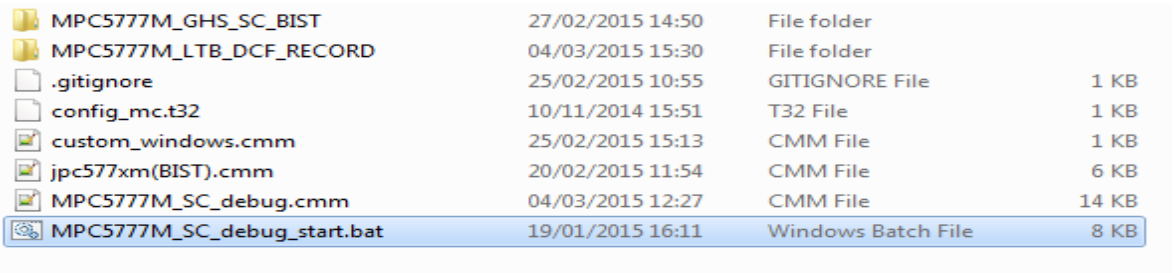


Figure 11. Contents of software package

13.2. Executing the software

To launch the software package click the MPC5777M_SC_debug_start.bat file. This will launch the TRACE32 debugger with a specific configuration for this application note. The first screen that will appear is shown in figure 12. This has the dialog options –Flash and DCF. The Flash option programs the online BIST STCU2 sequence into the device code flash, and the DCF option opens a further dialog to program the offline BIST for SHORT or Ultra-short sequence into the device UTEST flash. It is initially recommended to select the Flash option to program in the online BIST code, and then continue to configure the DCF records.

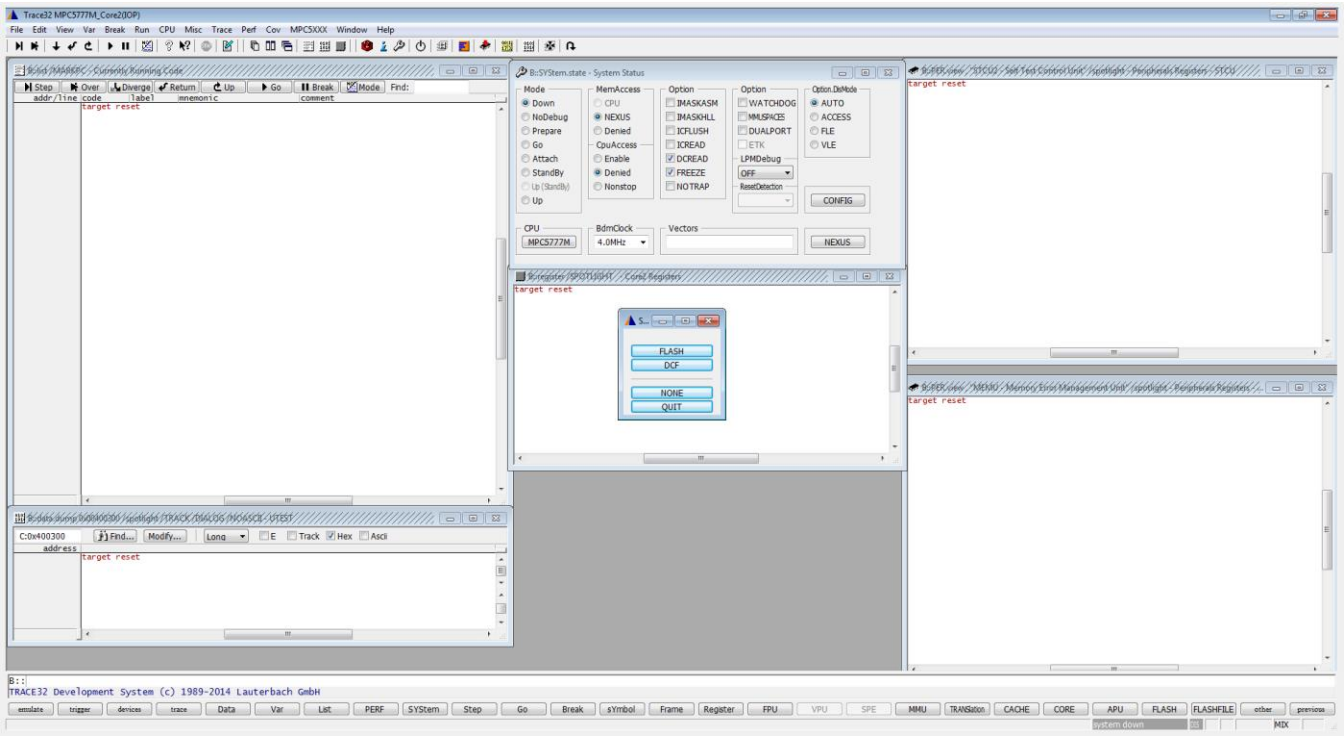


Figure 12. Initial dialog when launching debugger script

13.3. Debugger interface after flashing

If the user selects Flash the online BIST code will be flashed into the device and the device will then execute until it reaches the main function. The debugger will be preconfigured to display the STCU2 and MEMU peripheral windows to allow observation of the STCU2 configurations and test results. There is also a memory dump area showing the UTEST field of the memory map so that the user can observe the DCF records present in the UTEST. This is shown in figure 13.

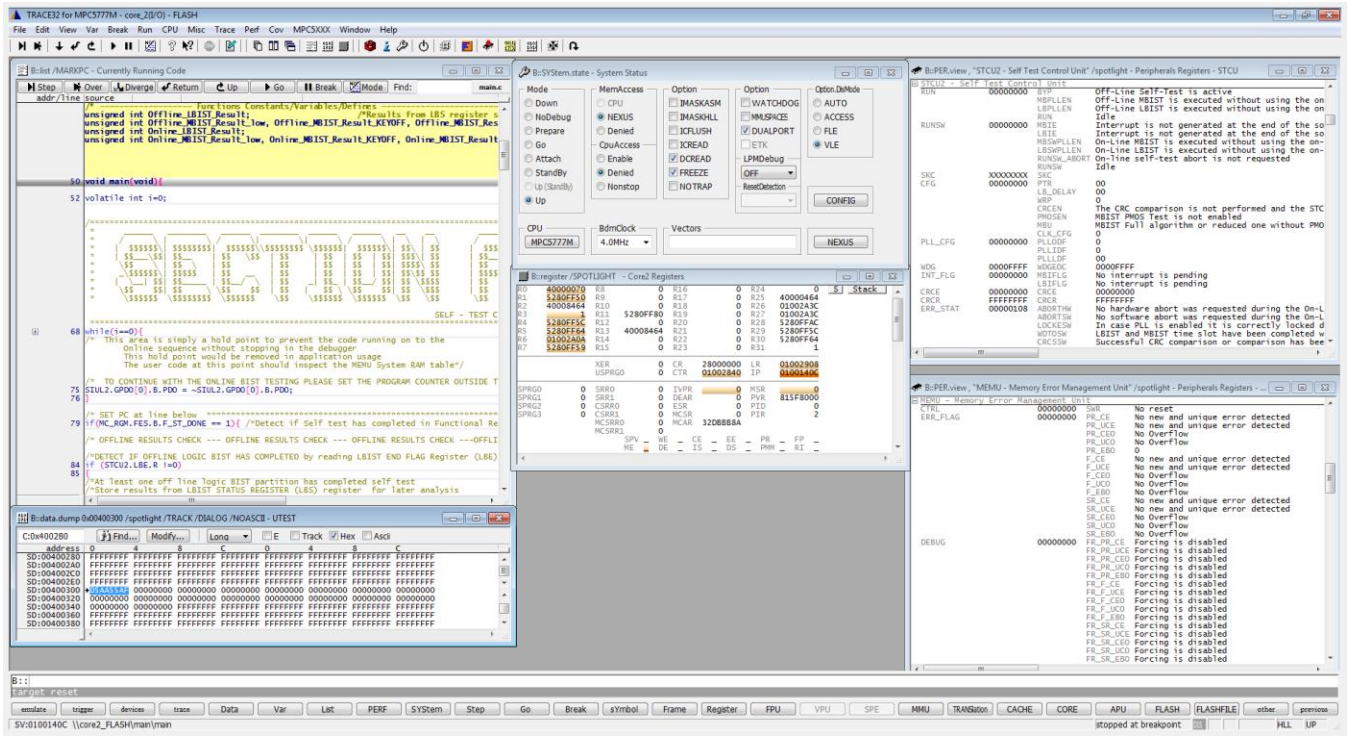


Figure 13. Debugger status after Flashing online BIST code

13.4. Programming the DCF records into UTEST

The user should select the DCF record programming dialog icon as shown in figure 14 item E.

Figure 14. Custom Debugger Icons pertinent to this application note

A – Restart Debug session – Restarts entire debug session, Resets device and debugger and returns to the original dialog box

B - Reload Start Script – Reloads the original Flash/DCF dialog box and window positions, no effect on device.

C – Compile GHS project – Shortcut to the GHS Multi IDE executable to allow project compilation from within debugger

D – Update code Flash and load symbols

E- DCF record programming dialog – Opens dialog to select which offline BIST test DCF record configuration to load

F – Show UTEST – Creates a memory dump window containing the UTEST DCF region

G – Attach/Break – Used to attach the debugger to the device after it has completed BIST

After selecting the option E, the user is presented with a warning message. The UTEST area is OTP and programming in the STCU2 configurations cannot be undone. Should the user wish to continue then the desired offline BIST STCU2 configuration example should be selected for programming into the UTEST area. The dialog will then indicate whether the DCF programming has been successful.

13.5. Executing BIST tests and reading results

After the user has completed programming the offline BIST DCF records, a power cycle or destructive reset should be carried out on the hardware. This will result in the execution of the offline BIST test. After completion of the offline BIST tests the device will load and execute the online BIST code that has been programmed into the device code Flash at the launch of the script. The start of the main function has a software trap to hold the device in a loop where the user can examine the offline test results. The user should attach the debugger to the device by pressing the Attach/break icon in the debugger (item G from figure 14). The will cause the debugger to stop the device in the trap loop.

The debugger will now be in the state as shown in figure 15. The execution status of the BIST tests is shown in the STCU2 window, and the user can also inspect the MEMU fields for potential errors.

To progress with the online BIST tests the user should manually exit the while loop by moving the device program counter to the MC_MODE_INIT function that is immediately after the while loop. Selecting “Go” in the debugger now will execute the online BIST tests. At the end of the online BIST tests the device is configured to perform a functional Reset, which will occur and the device will then execute the offline BIST at reboot and will again be held at the start loop. The user should then use the attach/break button to attach the debugger to the device again. The user can now read the online LBIST results and offline MBIST and LBIST results. If the user wish to inspect the online MBIST results they should place a breakpoint immediately after the online MBIST code sequence, and execute to that breakpoint.

It is important that the user does not single step through, or break within, the online BIST test sequences as this will cause execution errors. The user should also be aware that opening other debugger windows to memory areas within the device can potentially cause errors to appear in the MEMU error reporting tables. The user should not confuse these with genuine STCU2 BIST error reports.

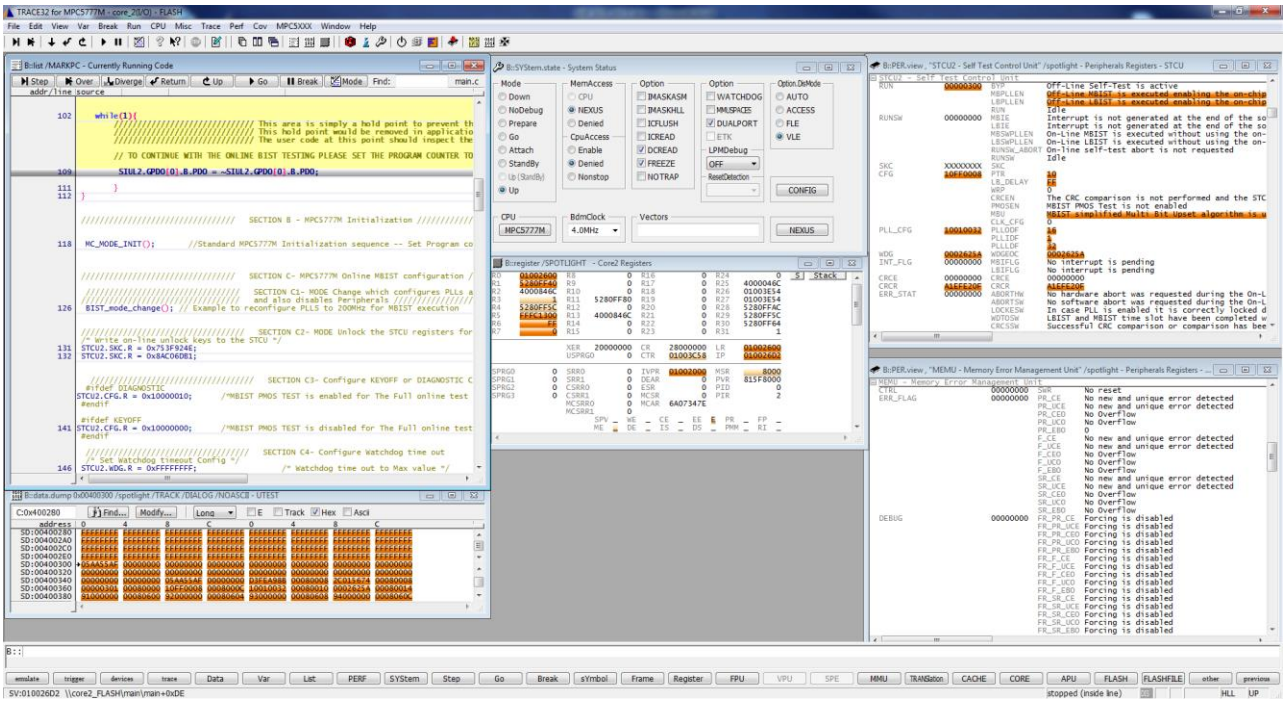


Figure 15. Debugger status after BIST execution

14. Revision history

Revision number	Date	Substantive changes
0	10/2015	Initial release

1	02/2019	• Editorial updates throughout the document.
		• Changed Freescale Semiconductors to NXP Semiconductors throughout the document.
		• In Objective added statement “Added statement “NXP guarantees the functionality of these four configurations and highly recommends users utilize these in their application” in
		  ○ Objective
		  ○ Offline BIST Procedure to Configure by DCF
		  ○ Online BIST Procedure
		• In MBIST and LBIST changed “LBIST can be configured to test partitions sequentially or in parallel …“ to “LBIST must be configured to test partitions sequentially”.

		• In LBIST Testing
		  ○ Changed "The STCU2 can be configured to either run all LBIST controllers in parallel or sequentially." to "The STCU2 must be configured to run all LBIST controllers sequentially."
		  ○ Removed "A combination of parallel and sequential testing can also be used."
		• Grammatical updates in MBIST Testing.
		• In Offline BIST Procedure to Configure by DCF
		  ○ In Table 4: Offline BIST programming Procedure
		  ▪ In Row "Configure LBIST"
		  • In Notes, changed "CSM: Define if the next LBIST will be run concurrently or sequentially with this partition" to "CSM: Set to 0 for sequential operation of LBIST module"
		  • In notes, added modifier "to configure Shift speed to 50 MHz"
		  ▪ In Row "Configure STCU2 Parameters"
		  • In Notes, changed "LB_DELAY: If LBIST partitions … as the BIST starts" to "LB_DELAY: Not used"
		  ▪ In Row "Configure PLL"
		  • In Notes, corrected PLL FREQ equation
		• In On-Line Self-Test STCU2 Configuration
		  ○ In Table 5: On-line self test example sequence
		  ▪ In Row "G"
		  • In Notes, changed "CSM: Define if the next LBIST will be run concurrently or sequentially with this partition" to "CSM: Set to 0 to configure LBIST Module to run sequentially"
		  • In Notes, added modifier "to configure Shift speed to 50MHz"

		▪ In Row “I” • In Notes, changed “LB_DELAY: If LBIST partitions … as the BIST starts” to “LB_DELAY: Unused” • In Notes added “CLK_CFG: Clock divider for STCU2 – Use examples provided” • In Example configurations for Table 8 changed o LBIST Coverage for Ultra-Short and Short, 3 partitions are tested, not 4 as previously stated.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2019 NXP B.V.

Document Number AN5131
Revision 1, 02/2019