Technical note

Built-in self-tests for SPC570Sx

Introduction

The purpose of this document is to describe how to configure and use built-in self-tests (BIST) in SPC570Sx device. Typical usage of the BIST is to detect latent failures before the user code is executed.

There are two different BIST tests available: LBIST (logic BIST) and MBIST (memory BIST).

1 Built-In self-tests

The purpose of this document is to describe how to configure and use built-in self-tests (BIST) in SPC570Sx devices. Typical usage of the BIST is to detect latent failures before the user code is executed. There are two different BIST tests available: LBIST (logic BIST) and MBIST (memory BIST).

A peripheral responsible for the BISTs execution is self-test control unit (STCU2). The STCU2 can execute BISTs during the microcontroller reset phase. Configuration of the STCU is performed via DCF records, which are stored at a specific UTEST address in a test Flash area–once time programmable OTP. User is allowed to program new DCF records, but there is no way to erase the test Flash area. Execution of the DCF records is performed by a system status and configuration module (SSCM) during the reset phase.

1.1 Reset sequence and BIST execution

The hardware reset stage is handled by the RGM module through a state machine. The RGM executes the device initialization through a sequence of phases. At the end of the hardware reset stage, the device starts the software boot stage. The software boot stage is actually the very first code executed by the processor. This code, called BAF, is programmed by ST in a specific Flash location and cannot be modified.

Based on DCF records stored in the user test Flash area, the STCU2 performs BISTs after the destructive reset phase 3.

The destructive reset phase 3 is entered after the reset destructive phase 2 is completed. During this phase, the SSCM module uploads the system configuration and dispatches to the proper clients. During this phase, the CPU reset vector is written into the mode entry (ME) module and the trimming for the analog IPs (such as VR, LVD/HVD, temperature sensor, ADC, I/O, oscillator) are applied. In case no BIST is to be executed, the device can be kept into the phase 3 DEST by keeping the PORST. This can be used for debugging purposes by allowing external test/development equipment to connect through JTAG to the internal debugger.

Figure 1. Reset sequence

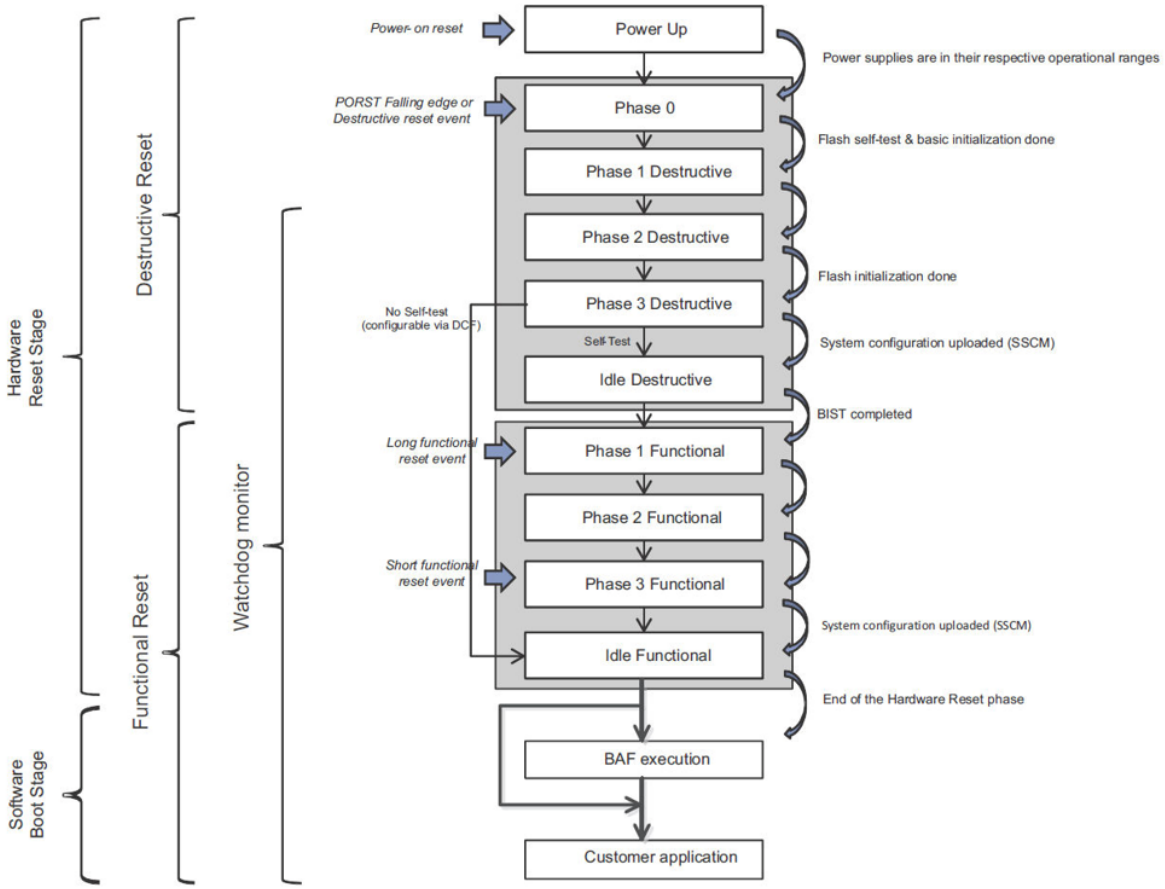
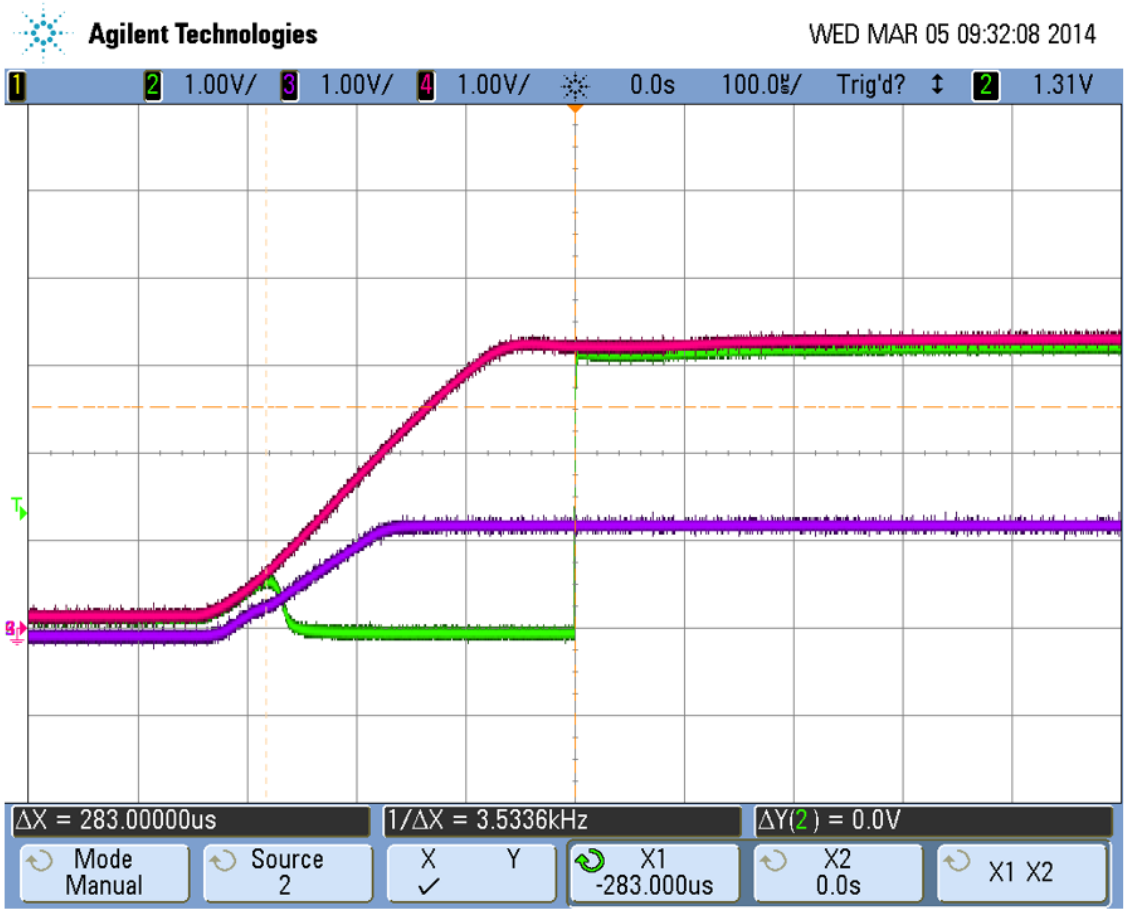


Figure 2. **PORST pin behavior**



Probes:

A2 – green PORST signal (pin 44 of eTQFP64 package)

A3 – violet 1.2 V power supply

A4 – red 3.3 V power supply

Example of power-on PORST pin behavior is shown in Figure 2. PORST pin behavior.

PORST reset signal is released during the reset phase 0. The figure is valid regardless the BISTs are executed after power-on reset or not executed at all.

1.2 System status and configuration module (SSCM)

The SSCM module is used during the boot phase to retrieve a set of system and blocks configurations and dispatch them to the proper modules. The SSCM reads device configuration data from the test Flash - it interprets the data as a series of "write" commands for the connected "client" modules.

Note: STCU2 is one of the clients.

The configuration is stored into an area of the test Flash and organized into a 64-bits record, called DCF (device configuration format). Each DCF contains a client identifier, an address (which identifies the data type within a specific client), and 32-bits data.

The SSCM, during the reset phase 3 reads all the DCF records and dispatches them to the various system modules.

1.2.1 DCF record description

The DCF record structure is reported in the figure below and the DCF record description in Table 1. DCF record description.

Figure 3. DCF record structure

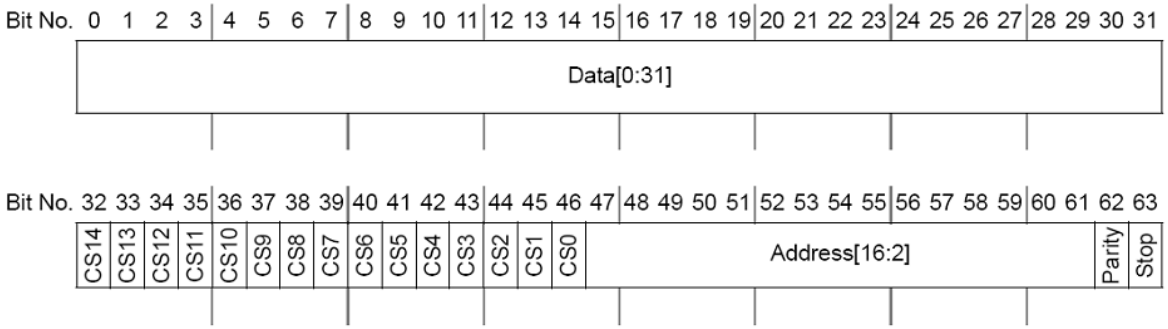


Table 1. DCF record description

Field	Description
0:31 Data[0:31]	32-bit of data that is to be written to the DCF client
32:46 CSn	Chip select n. One chip select is asserted ("1") per DCF record to select the target module for the DCF client. All other chip selects should be negated ("0").
47:61 Address[16:2]	Address of the DCF client within the selected module (1)
62 Parity	Parity bit for the DCF record (2)
63 Stop	Stop bit. This bit indicated the end of the list od DCF records. (3) 0 not the end of the list 1 end of the list

1. *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.*
2. *This bit is NOT implemented for DCF client written from UTEST*
3. *The erased state of the Flash memory 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.*

1.2.2 DCF records location

The DCF records are stored in the user test Flash area (UTEST) starting from address 0x00400300 up to 0x00400FFF.

Table 2. Series of DCF records in UTEST Flash memory

ADDR offset	DATA			
0x00	0x05AA55AF			
0x04	0x00000000			STOP=0
0x08	WDATA[31:0]			
0x0C	CD[14:0]	ADDR[16:2]	PRTY	STOP=0
0x10	WDATA[31:0]			
0x14	CD[14:0]	ADDR[16:2]	PRTY	STOP=0
...	...			
8n-1 + 0x0	Reserved			
8n-1 + 0x4	Reserved			1
8n + 0x0	-			
8n + 0x4				

Caution:

- There must never be an unprogrammed record in the DCF data structure, as it is interpreted as a stop record (bit 63 is set to 1) and subsequent records are ignored. This allows to program the records in several sessions, each time appending new records at the end of the list.
- There is not possible to erase once programmed record. UTEST area is OTP (once time programmable – not erasable.) But there is the possibility to modify DCF records in some cases (see Section 1.4.7 STCU setup invalidation).
- It is possible for more than one DCF Record to write 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 overwriting; this depends on the DCF client implementation.

1.3 Self test control unit (STCU2)

1.3.1 Self-test control unit (STCU2) introduction

Self-test control unit (STCU2) is a comprehensive programmable hardware module that controls the self-test sequence applied both during the off-line and/or on-line conditions. The hardware can manage the logic built-in self-test (LBIST) and the SRAM or ROM built-in self-test (MBIST) blocks of the device. The STCU2 includes the SSCM DCF bus to load the self-test parameters (LBIST or MBIST scheduling activity, LBIST setup, unrecoverable faults (UF) or recoverable faults (RF) management, CRC and MISR expected values, PLL management, and so on) from flash memory during the off-line self-test phase. This interface is only able to write the configuration parameters and start the self-test execution once the STCU2 global reset has been applied. Register access by software is granted by an IPS interface to check the results of the off-line self-test and to load or check the execution of the on-line self-test.

Note: SPC570Sx has implemented off-line self-test only executed during a destructive reset phase.

The STCU2 includes:

- A programmable watchdog timer (WDG) to check the LBIST and MBIST self-test operations have completed within the assigned time slot or the STCU2 RUN or BYPASS bits have been programmed before watchdog time-out.
- Direct PLL control during the off-line self-test sequence.
- Two sets of registers to separately collect the LBIST or MBIST results.

To increase the security of this module, a different security key sequence is also required during self-test to unlock write access to the STCU2 registers.

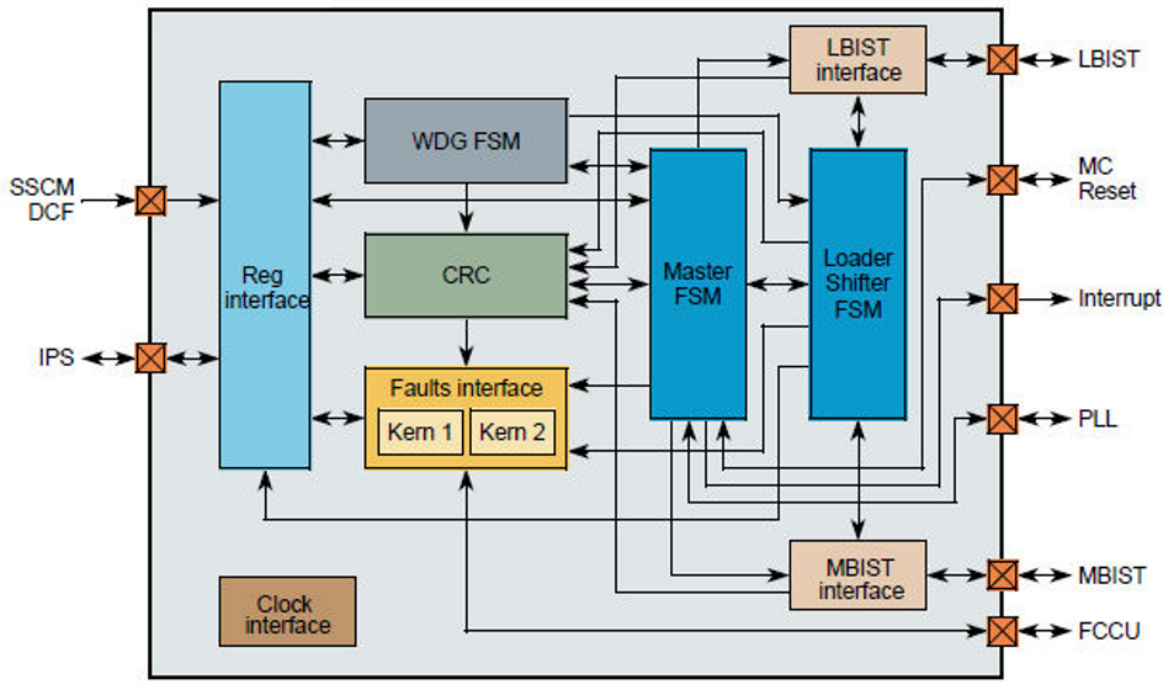
For more details, see Figure 4. STCU2 block diagram.

1.3.2 BIST watchdog

STCU includes two different watchdogs:

- A hard-coded watchdog used to Auto Lock the STCU2 access, forcing a reset condition on the double security key registers.
- A dual function watchdog:
  - After a reset event initializes the STCU2, it is used as a hard-coded watchdog time-out.
  - During LBIST or MBIST runs, it is used as a programmable Watchdog timer to check the LBIST or MBIST have completed in the designated time slot.

Figure 4. STCU2 block diagram



1.3.3 BIST clocking

An internal IRC oscillator 16 MHz or PLL could be used for LBIST tests execution. (LBIST tests execution maximal frequency is 20 MHz, when PLL is used.)

There is possible to enable PLL at up to 80 MHz during MBIST test execution to speed up the testing process.

STCU is able to control the PLL0 during the off-line self-test operations depending on the status of the STCU_RUN[MBPLLEN]. By this flags, off-line MBIST executed enabling the on-chip PLL control interface selecting the parameters defined into STCU_PLL_CFG register (PLLODF, PLLIDF and PLLDF).

Following the formula of PLL0 frequency output:

$$fPLLout = \frac{fVCO}{PLLODF*2}$$

Where:

$$fVCO = fPLLin*(PLLDF*2) = \left(\frac{fRC}{PLLIDF}\right)*\left(PLLDF*2\right)$$

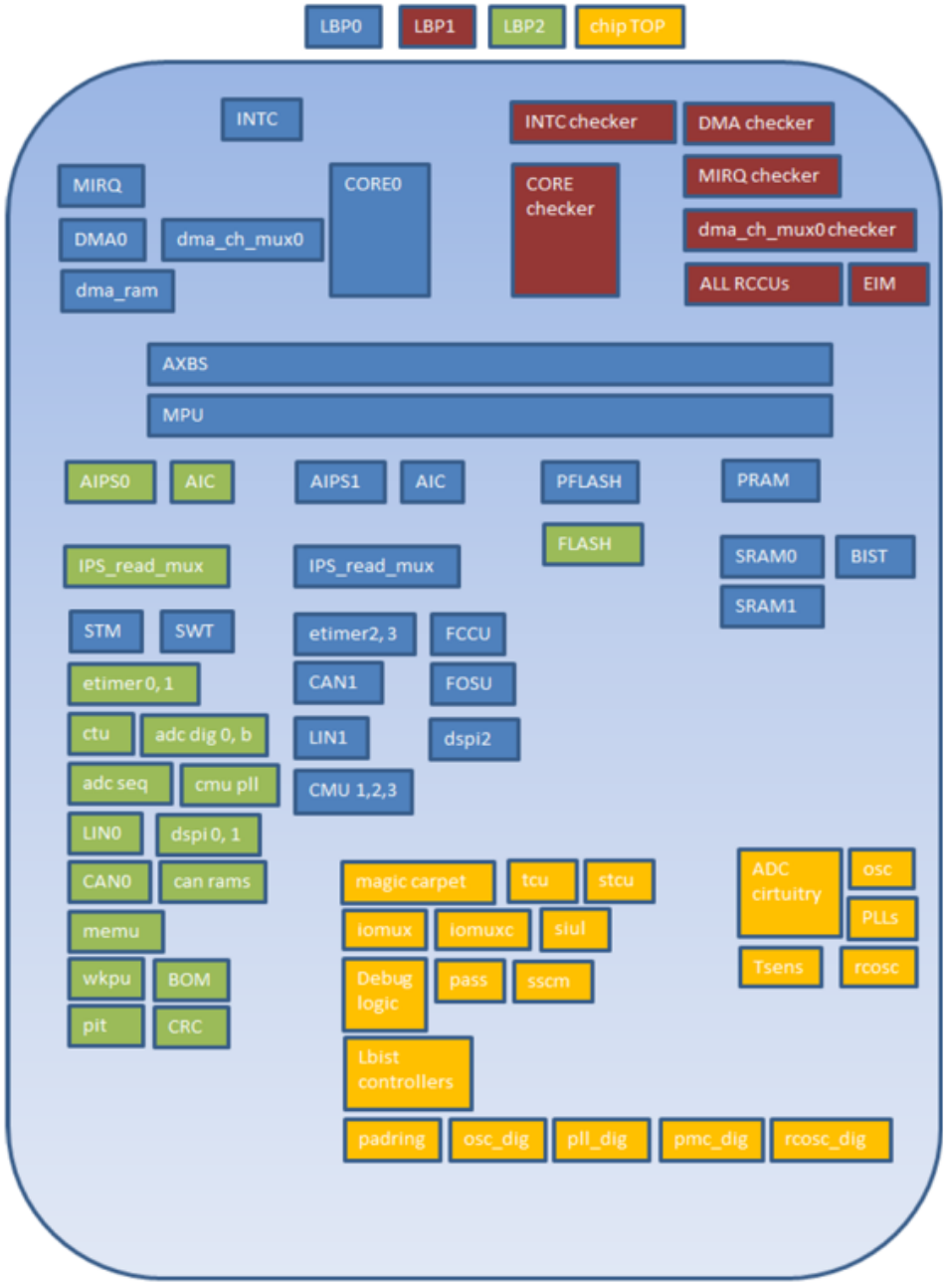
f RC is the frequency of the internal oscillator that is 16 MHz.

Caution: there is necessary to take in account the RC OSC accuracy (typically +-8% see datasheet) for PLL setup not to overcome maximal-allowed clocking frequency.

1.3.4 STCU LBIST description

The LBIST (logic build in self test) contains 3 partitions (lakes of tests):

Figure 5. LBIST structure



Note: LBIST coverage achieved on monitors is 90% and on each partition is >70% using 2000 patterns.

1.3.5 BIST MBIST description

The MBIST (memory build in self test) contains 9 parts:

- MB0 ram_sram_1
- MB1 ram_sram_0
- MB2 ram_dma
- MB3 ram_can_rxim_1
- MB4 ram_can_rxim_0
- MB5 ram_can_mb_1
- MB6 ram_can_mb_0
- MB7 ram_mpflash
- MB8 bam_rom

1.4 BIST setup examples

It is possible to prepare different setup of BIST test. This chapter describes some examples of the following configurations:

- LBIST only
- MBIST only
- LBIST and MBIST
- STCU bypass

1.4.1 MBIST example setup

Table 3. MBIST example setup

Nr.	DCF record	STCU register	Note
1	D3FEA98B 00080008	STCU_KEY1	Key 1/2 to unlock STCU access
2	2C015674 00080008	STCU_KEY2	Key 2/2 to unlock STCU access
3	10000001 0008000C	STCU_CFG	Pointer set for MBIST0, MBIST reduced algorithm (PMOSEN bit set to '0'), PMOSEN and MBU can be played to choose different algorithms and MBIST coverage, WRP bit set to '0' to allow IPS access.
4	0C02003C 00080010	STCU_PLL_CFG	PLL programmed for 40MHz with input frequency as RC(16 MHz+-8%), PLL output freq = (Input_freq*LDF)/(ODF*IDF) = (Input_freq* 0x3C)/(0x0C*02) = 40MHz, according to errata DAN-0043391
5	0003D090 00080014	STCU_WDG	Watchdog timeout setup
6	00000000 00080074	STCU_MBUFML	Programming '1' to the respective MBIST in this register generates an unrecoverable fault in the case of MBIST failure. If the fault is permanent, then the device will never recover from this programming. So it is safe to program '0' in this register and later STCU status can be read by software.
7	00000000 00080028	STCU_ERR_FM	Programming '1' to any corresponding bit of this register can make the device unrecoverable if the fault is permanent.
8	91000000 00080600	MBIST_CTRL0	Next pointer set to MBIST1, CSM BIST set to '1' to indicate parallel running of MBIST0 and MBIST1
9	92000000 00080604	MBIST_CTRL1	Next pointer set to MBIST2, CSM BIST set to '1' to indicate parallel running of MBIST1 and MBIST2
10	93000000 00080608	MBIST_CTRL2	Next pointer set to MBIST3, CSM BIST set to '1' to indicate parallel running of MBIST2 and MBIST3
11	94000000 0008060C	MBIST_CTRL3	Next pointer set to MBIST4, CSM BIST set to '1' to indicate parallel running of MBIST3 and MBIST4

Nr.	DCF record	STCU register	Note
12	95000000 00080610	MBIST_CTRL4	Next pointer set to MBIST5, CSM BIST set to '1' to indicate parallel running of MBIST4 and MBIST5
13	96000000 00080614	MBIST_CTRL5	Next pointer set to MBIST6, CSM BIST set to '1' to indicate parallel running of MBIST5 and MBIST6
14	97000000 00080618	MBIST_CTRL6	Next pointer set to MBIST7, CSM BIST set to '1' to indicate parallel running of MBIST6 and MBIST7
15	98000000 0008061C	MBIST_CTRL7	Next pointer set to MBIST8, CSM BIST set to '1' to indicate parallel running of MBIST7 and MBIST8
16	7F000000 00080620	MBIST_CTRL8	Next pointer set to null pointer to indicate end of self test
17	00000201 00080000	STCU_RUN	Self test triggered with MBIST run on PLL frequency

1.4.2 MBIST evaluation

The MBIST test result is available in STCU registers for evaluation in user application.

ERR_STAT register content to be equal to zero as global no error status.

STCU_MBEx registers contain bit field with finished MBISTs. Content shall match with executed MBISTs based on DCF records. STCU_MBSx contain bit field with finished MBISTs without error. Content of corresponding registers to be equal STCU_MBEx = STCU_MBSx.

Table 4. MBIST results

```
STCU_ERR_STAT = 0x0
STCU_MBEL = 0x1FF
STCU_MBSL = 0x1FF
STCU_MBEM = 0x0
STCU_MBSM = 0x0
STCU_MBEH = 0x0
STCU_MBSH = 0x0
```

1.4.3 LBIST example setup

Table 5. LBIST example setup

Nr.	DCF record	STCU register	Note
1	D3FEA98B 00080008	STCU_KEY1	Key 1/2 to unlock STCU access
2	2C015674 00080008	STCU_KEY2	Key 2/2 to unlock STCU access
3	005A0001 0008000C	STCU_CFG	Pointer set for LBIST partition0, WRP bit set to '0' to allow IPS access. Refer to STCU doc for more details.
4	18020037 00080010	STCU_PLL_CFG	PLL programmed for 18.3MHz with input frequency as RC(16 MHz+-8%), PLL output freq = (Input_freq*LDF)/ (ODF*IDF) = (Input_freq* 0x37)/(0x18*0x02)
5	0001E848 00080014	STCU_WDG	Watchdog timeout setup
6	00000000 00080040	STCU_LBUFM	Programming '1' to the respective LBIST in this register generates an unrecoverable fault in the case of LBIST failure. If the fault is permanent, then the device will never recover from this programming. So it is safe to program '0' in this register and later STCU status can be read by software.
7	81001100 00080100	STCU_LB_CTRL0	Next pointer to LBIST partition1, parallel mode, CSM bit (31st) can be played with to run
8	000007D0 00080104	STCU_LB_PCS0	Number of patterns -2000 for partition0

Nr.	DCF record	STCU register	Note
9	9CCC9484 00080110	STCU_LB_MISREL0	MISR lower 32 bits for 2000th pattern for partition0. It changes according to the number of patterns to be executed.
10	36565224 00080114	STCU_LB_MISREH0	MISR upper 32 bits for 2000th pattern for partition0. It changes according to the number of patterns to be executed.
11	82001100 00080140	STCU_LB_CTRL1	Next pointer to LBIST partition2, parallel mode, CSM bit(31st) can be played with to run
12	000007D0 00080144	STCU_LB_PCS1	Number of patterns: 2000 for partition1
13	CFC6629E 00080150	STCU_LB_MISREL1	MISR lower 32 bits for 2000th pattern for partition1. It changes according to the number of patterns to be executed.
14	10E9A429 00080154	STCU_LB_MISREH1	MISR upper 32 bits for 2000th pattern for partition1. It changes according to the number of patterns to be executed.
15	7F001100 00080180	STCU_LB_CTRL2	Next pointer to null to indicate end of self-test
16	000007D0 00080184	STCU_LB_PCS2	Number of patterns: 2000 for partition2
17	A582B986 00080190	STCU_LB_MISREL2	MISR lower 32 bits for 2000th pattern for partition2. It changes according to the number of patterns to be executed.
18	D654207A 00080194	STCU_LB_MISREH2	MISR upper 32 bits for 2000th pattern for partition2. It changes according to the number of patterns to be executed.
19	00000101 00080000	STCU_RUN	STCU triggered with LBIST programmed to run on PLL freq.

1.4.4 LBIST evaluation

The LBIST test result is available in STCU registers for evaluation in user application.

ERR_STAT register content to be equal to zero as global no error status.

STCU_LBE register contains bit field with finished LBISTs. Content shall match with executed LBISTs based on DCF records. STCU_LBS register contains bit field with finished LBISTs without error. Content of the registers to be equal STCU_LBE = STCU_LBS.

Content of corresponding expected STCU_MISREx and read STCU_MISRRx register values shall match for all executed LBISTs.

Table 6. LBIST results

```
STCU_ERR_STAT = 0x0
STCU_LBE = 0x7
STCU_LBS = 0x7
LBIST0
STCU_LB_MISREL0 = 0x9CCC9484
STCU_LB_MISRRL0 = 0x9CCC9484
STCU_LB_MISREH0 = 0x36565224
STCU_LB_MISRRH0 = 0x36565224
LBIST1
STCU_LB_MISREL1 = 0x0CFC6629E
STCU_LB_MISRRL1 = 0x0CFC6629E
STCU_LB_MISREH1 = 0x10E9A429
STCU_LB_MISRRH1 = 0x10E9A429
LBIST2
STCU_LB_MISREL2 = 0x0A582B986
STCU_LB_MISRRL2 = 0x0A582B986
STCU_LB_MISREH2 = 0x0D654207A
STCU_LB_MISRRH2 = 0x0D654207A
```

1.4.5 Both LBIST and MBIST example setup

Table 7. LBIST and MBIST example setup

Nr.	DCF record	STCU register	note
1	D3FEA98B 00080008	STCU_KEY1	Key 1/2 to unlock STCU access
2	2C015674 00080008	STCU_KEY2	Key 2/2 to unlock STCU access
3	005A0001 0008000C	STCU_CFG	Pointer set for LBIST partition0, MBIST reduced algo (PMOSEN, MBU bits set to '0'), PMOSEN and MBU can be played with to choose different algo and MBIST coverage, WRP bit set to '0' to allow IPS access, please refer to STCU doc for more details
4	0C02003C 00080010	STCU_PLL_CFG	PLL programmed for 40MHz with input frequency as RC(16 MHz+-8%), PLL output freq = (Input_freq*LDF)/(ODF*IDF), = (Input_freq* 0x3C)/(0x0C*02) = 40MHz, according to errata DAN-0043391
5	0003D090 00080014	STCU_WDG	Watchdog timeout setup
6	01001100 00080100	STCU_LB_CTRL0	Next pointer to LBIST partition1, sequential mode, CSM bit(31st) can be played with to run in serial mode
7	000007D0 00080104	STCU_LB_PCS0	Number of patterns -2000 for partition0
8	9CCC9484 00080110	STCU_LB_MISREL0	MISR lower 32 bits for 2000th pattern for partition0. Will change according to the number of patterns to be executed.
9	36565224 00080114	STCU_LB_MISREH0	MISR upper 32 bits for 2000th pattern for partition0. Will change according to the number of patterns to be executed.
10	02001100 00080140	STCU_LB_CTRL1	Next pointer to LBIST partition2, sequential mode, CSM bit(31st) can be played with to run in serial mode
11	000007D0 00080144	STCU_LB_PCS1	Number of patterns -2000 for partition1
12	CFC6629E 00080150	STCU_LB_MISREL1	MISR lower 32 bits for 2000th pattern for partition1. Will change according to the number of patterns to be executed.
13	10E9A429 00080154	STCU_LB_MISREH1	MISR upper 32 bits for 2000th pattern for partition1. Will change according to the number of patterns to be executed.
14	10001100 00080180	STCU_LB_CTRL2	Next pointer to MBIST0 to indicate MBIST run after LBIST
15	000007D0 00080184	STCU_LB_PCS2	Number of patterns -2000 for partition2
16	A582B986 00080190	STCU_LB_MISREL2	MISR lower 32 bits for 2000th pattern for partition2. Will change according to the number of patterns to be executed.
17	D654207A 00080194	STCU_LB_MISREH2	MISR upper 32 bits for 2000th pattern for partition2. Will change according to the number of patterns to be executed.
18	91000000 00080600	MBIST_CTRL0	Next pointer set to MBIST1, CSM BIST set to '1' to indicate parallel running of MBIST0 and MBIST1
19	92000000 00080604	MBIST_CTRL1	Next pointer set to MBIST2, CSM BIST set to '1' to indicate parallel running of MBIST1 and MBIST2
20	93000000 00080608	MBIST_CTRL2	Next pointer set to MBIST3, CSM BIST set to '1' to indicate parallel running of MBIST2 and MBIST3
21	94000000 0008060C	MBIST_CTRL3	Next pointer set to MBIST4, CSM BIST set to '1' to indicate parallel running of MBIST3 and MBIST4
22	95000000 00080610	MBIST_CTRL4	Next pointer set to MBIST5, CSM BIST set to '1' to indicate parallel running of MBIST4 and MBIST5
23	96000000 00080614	MBIST_CTRL5	Next pointer set to MBIST6, CSM BIST set to '1' to indicate parallel running of MBIST5 and MBIST6
24	97000000 00080618	MBIST_CTRL6	Next pointer set to MBIST7, CSM BIST set to '1' to indicate parallel running of MBIST6 and MBIST7

Nr.	DCF record	STCU register	note
25	98000000 0008061C	MBIST_CTRL7	Next pointer set to MBIST7, CSM BIST set to '1' to indicate parallel running of MBIST7 and MBIST8
26	7F000000 00080620	MBIST_CTRL8	Next pointer set to null pointer to indicate end of selftest
27	00000201 00080000	STCU_RUN	Selftest triggered, MBIST run with PLL, LBIST with RC

1.4.6 STCU bypass

By default there is set STCU_RUN.BYP = 0 after reset. It causes, that during reset phase, the STCU waits (until hardcoded WDG Time-out) for BIST setup and STCU_RUN.RUN = 1 to execute the BIST tests.

Figure 6. STCU bypass example – memory dump

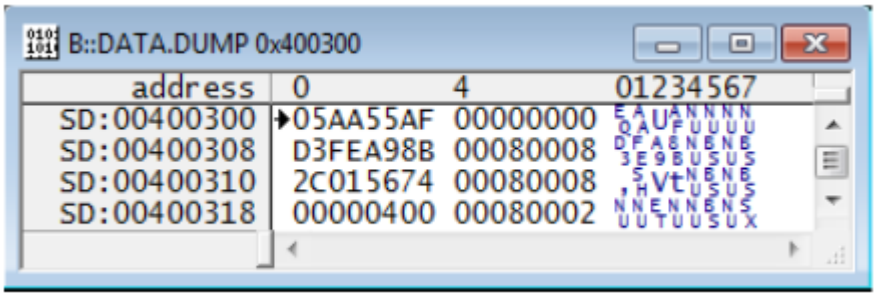


Figure 7. STCU bypass example – description

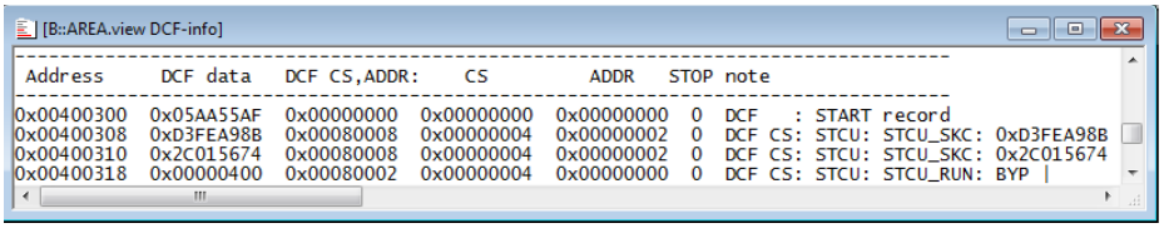


Table 8. STCU bypass structure

Nr.	DCF record	STCU register	Note
1	05AA55AF 00000000	DCF start record	DCF start record. The start record must be placed at the beginning of the DCF area in UTEST flash memory to indicate to the device that the following data records must be processed
2	D3FEA98B 00080008	STCU_KEY1	Key 1/2 to unlock STCU access
3	2C015674 00080008	STCU_KEY2	Key 2/2 to unlock STCU access
4	00000400 00080000	STCU_RUN	00000400 is the setting of STCU_RUN register. In particular it is set the STCU_RUN[BYP] bit. It means that off-line self-test is completely by-passed and the access to STCU2 is locked until the STCU_SCK register is written according to the request access mode

1.4.7 STCU setup invalidation

Considering the default configuration shown in the previous paragraph, STCU is bypassed. If the STCU is bypassed it does not accept any extra programming.

Since a final user can't erase the UTEST Flash context, the only possibility to invalidate the bypass STCU configuration consist in altering some STCU Double Words already written in the UTEST. The double words cannot be erased but they can be reprogrammed only in the positions where '1' appear, but ECC to be taken into account. Eight additional bits for every 64-bits are implemented to to have the possibility of single bit correction and double bit detection. See RM for more details.

Moreover, there is the constraint not to introduce ECC errors in the UTest by overwriting its content without care. It is necessary to reprogram both Double Words (@0x00400308 and @0x00400310) in order to invalidate the STCU bypassed and, at the same time, doesn't inject ECC errors.

The new words, which have been specifically chosen not to trigger any ECC error due to the over-write, are: 0xD3FE0889_00000000 and 0x2C015644_00000000.

To recap the re-writing regards the following values:

- 0xD3FE0889_00000000at the address 0x0040_0308 (to overwrite original value: D3FEA98B_0008008)
- 0x2C015644_00000000 at the address 0x0040_0310 (to overwrite original value: 2C015674_0008008)

These values invalid the bypass bit of STCU_RUN because they are "wrong" keys (due to the ECC error injection it was not possible to find a suitable value to over-write directly the STCU_RUN value.) and, very important, doesn't trigger any ECC error. With such invalidated DCF records, the STCU doesn't receive any STCU KEY valid sequence and therefore the bypass command does not take effect.

Once the DCF records (writing keys into STCU_SKC register) were invalidated, all DCF records following the invalid keys are ignored by the STCU. It allows user to prepare and write a new set of DCF records at the end of the original DCF record area – starting with correct keys.

Note: The STCU setup invalidation can be used not only for bypass skipping, but also to replace current DCF records with new ones or let them invalid only. Thanks to invalidating the last key1 and key2, it is possible proceed with a new STCU setup by adding new set of DCF record staring with valid keys.

WARNING: if ECC errors are injected in the UTEST sector, for example by re-writing the same location with not specifically defined data, there is the risk that the sample can't exit the boot phase. In this case the sample becomes useless.

Code below is an exception of a Lauterbach script used for debugging the off-line self-test. It describes how to overwrite the two STCU keys related to the default configuration of the UTest.

Table 9. **STCU setup invalidation Lauterbach script excerpt**

```
    ;Lock0 ->TSLock enable UTEST memory
    PER.S ANC:0xFFFE0010 %LONG 0x3FFFFFFF

    ;SET_DCF_STCU_STCU_SCK 0xD3FE0889 - key1 overwrite
    GOSUB program_word 0x00400308 0xD3FE088900000000

    ;00 - SET_DCF_STCU_STCU_SCK 0x2C015644 - key2 overwrite
GOSUB program_word 0x00400310 0x2C01564400000000

    ;Lock0 ->TSUnLock enable UTEST memory
    ;PER.S ANC:0xFFFE0010 %LONG 0x0xBFFFFFFF;

program_word:
    entry &address &data

    ;MCR->PGM =1 enable program memory
    PER.S ANC:0xFFFE0000 %LONG 0x610

    D.S EA:(&address) %BE %QUAD (&data)

    ;MCR->EHV=1 program memory
    PER.S ANC:0xFFFE0000 %LONG 0x611

    ;MCR->EHV=0 program memory
    PER.S ANC:0xFFFE0000 %LONG 0x610

    ;MCR->PGM =0
    PER.S ANC:0xFFFE0000 %LONG 0x600

    RETURN
```

1.5 BIST relevant errata

A few errata items are relevant to BIST for SPC570Sx devices.

Device identification for SPC570Sx cut 2.1:

- JTAG_ID = 0x110A_2041
- MIDR1 register: MAJOR_MASK[3:0]: 4’b0001, MINOR_MASK[3:0]: 4’b0001

Table 10. **BIST relevant errata items**

ID	Module	Title	Description
PS1566	LBIST	LBIST: ADC is not converting correctly after LBIST RUN.	When BIST runs the reset value of some TEST registers are getting changed. Due to this ADC values (for both ADC_0 and ADC_B) are lower than the expected value.
DAN-0043391	STCU2	STCU2: PLL configuration needs to be changed for MBIST run	In device's RM, STCU_PLL_CFG (PLL configuration) value is wrongly provided for configuring PLL at 73.3MHz for Offline MBIST run. A too high PLL frequency might lead to the corruption of the logic of Peripheral Clock Divider (AIPS0), which wakes up – after Self Test sequence completion – with a wrong setting (dividing AIPS0 clock by 1 instead of 2). Recommended value for PLL frequency for Offline MBIST run is 40MHz: maximum allowed value is 60MHz.
DAN-0045775	BIST	BIST: Unexpected ESR0 signal in case of executing FULL BIST	The issue is present during offline self-test. PAD[47] when configured as ESR0 through DCF, shows different behavior at the end of the functional reset. 1) running the MBIST Only, ESR0 is deasserted at the end of functional reset

ID	Module	Title	Description
			2) running offline self-test (with LBIST included->FULL BIST), the ESR0 is deasserted at the end of destructive phase (just after BIST execution) and then asserted at the beginning of the next functional reset
ERR008056	LBIST	LBIST: Flash must be idle during LBIST	Logic Built-In Self-Test (LBIST) should only be performed on the flash partition while the flash is in the idle state. LBIST operations must not be initiated for the LBIST partition including flash while flash program or erase operations are in progress. This information was not included in the user documentation.

1.5.1 BIST errata PS1566

In case of LBIST is executed, then errata PS15566 workaround to be applied in user SW to avoid improper ADC conversions.

Table 11. Errata PS15566

ID	Workaround
PS1566	After LBIST run and before ADC initialization, restore RESET value of non-user TEST registers as default by SW. Below the listing of the SW statements to execute to restore default configuration: * ((vuint32_t *) 0xFFE38000) = 0x03020100; * ((vuint32_t *) 0xFFE38004) = 0x07060504; * ((vuint32_t *) 0xFFE38008) = 0x0B0A0908; * ((vuint32_t *) 0xFFE3800C) = 0x0F0E0D0C; * ((vuint32_t *) 0xFFE38010) = 0x13121110; * ((vuint32_t *) 0xFFE38014) = 0x17161514; * ((vuint32_t *) 0xFFE38018) = 0x1B1A1918; * ((vuint32_t *) 0xFFE3801C) = 0x1F1E1D1C; * ((vuint32_t *) 0xFFE38020) = 0x23222120; * ((vuint32_t *) 0xFFE38024) = 0x27262524; * ((vuint32_t *) 0xFFE38028) = 0x2B2A2928; * ((vuint32_t *) 0xFFE3802C) = 0x2F2E2D2C; * ((vuint32_t *) 0xFFE38030) = 0x33323130; * ((vuint32_t *) 0xFFE38034) = 0x37363534; * ((vuint32_t *) 0xFFE38038) = 0x3B3A3938; * ((vuint32_t *) 0xFFE3803C) = 0x3F3E3D3C; * ((vuint32_t *) 0xFFE04000) = 0x03020100; * ((vuint32_t *) 0xFFE04004) = 0x07060504; * ((vuint32_t *) 0xFFE04008) = 0x0B0A0908; * ((vuint32_t *) 0xFFE0400C) = 0x0F0E0D0C; * ((vuint32_t *) 0xFFE04010) = 0x13121110; * ((vuint32_t *) 0xFFE04014) = 0x17161514; * ((vuint32_t *) 0xFFE04018) = 0x1B1A1918; * ((vuint32_t *) 0xFFE0401C) = 0x1F1E1D1C; * ((vuint32_t *) 0xFFE04020) = 0x23222120; * ((vuint32_t *) 0xFFE04024) = 0x27262524; * ((vuint32_t *) 0xFFE04028) = 0x2B2A2928;

ID	Workaround
	* ((vuint32_t *) 0xFFE0402C) = 0x2F2E2D2C; * ((vuint32_t *) 0xFFE04030) = 0x33323130; * ((vuint32_t *) 0xFFE04034) = 0x37363534; * ((vuint32_t *) 0xFFE04038) = 0x3B3A3938; * ((vuint32_t *) 0xFFE0403C) = 0x3F3E3D3C;

1.5.2 BIST errata DAN-0043391

Table 12. Errata DAN-0043391

ID	Workaround
DAN-0043391	STCU_PLL_CFG configuration has to be changed from 0x06020038 00080010 to 0x0C02003C 00080010 to configure PLL at 40MHz or lower. The duration of MBIST run increases consequently, according to the selected algorithm. Total Offline Self Test sequence contains anyway also LBIST, which is not affected by PLL frequency, since it is run using on-chip RC oscillator. NOTE: Running Offline MBIST at a lower speed with respect to the maximum speed does not affect ASIL D targets for the hardware architectural metrics of the device. From Safety Analysis (FMEDA) point of view, Error Correction Code (ECC) is the primary Safety Mechanism against memory faults: on-field MBIST (Offline or Online) must be run only to prevent accumulation of faults, which would negatively impact diagnostic coverage of the primary safety mechanism (ECC). ECC does not cover multiple bit faults, it just corrects single bit errors and detects double bit errors.

The errata workaround is already applied in this document and all examples.

1.5.3 BIST errata DAN-0045775

Table 13. Errata DAN-0045775

ID	Workaround
DAN-0045775	Application has to ignore the spurious ESR0 toggling during case 2 (LBIST included->FULL BIST)

1.5.4 BIST errata ERR008056

Table 14. Errata ERR008056

ID	Workaround
ERR008056	Flash must be in an idle state for LBIST. Ensure that flash is not performing program, erase, or flash user accessible test mode operations when LBIST is initiated.

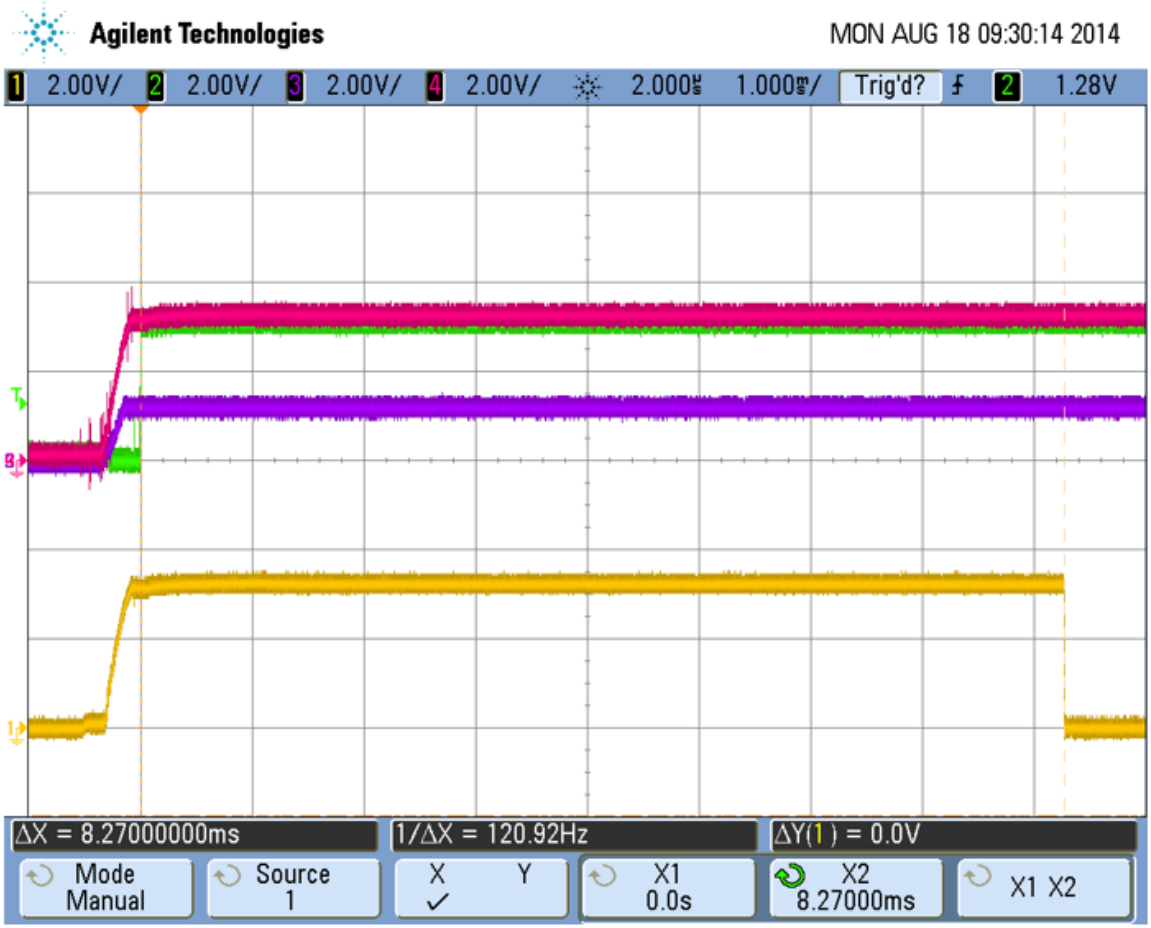
1.6 BIST performance

Practical measurements are performed with BIST setups mentioned in Section 1.4 BIST setup examples using standard evaluation board and oscilloscope for time measurement. An application is flashed, which sets PA[0] to output mode (causes port pin transition level High -> Low) in the very first startup instructions. Starting point for time measurement here is PORST transition to high (see Figure 2. PORST pin behavior), which is better measurable than power on voltage ramp-up.

Table 15. **BIST related practical measurement results**

#	Setup	Time [ms]	Note:
1	Original	8.3	no added DCF records (see Figure 8. Original device measurement result – no added DCF records)
2	LBIST only (parallel mode)	10.0	PLL 18.3MHz
3	MBIST only (parallel mode)	20.5	PLL 40MHz
4	LBIST and MBIST (sequential and parallel modes)	48.4	RC 16MHz + PLL40MHz
5	STCU bypass	0.5	

Figure 8. **Original device measurement result – no added DCF records**



Probes:
A1 – yellow PA[0] port pin (pin 2 of eTQFP64 package)
A2 – green PORST signal (pin 44 of eTQFP64 package)
A3 – violet 1.2 V power supply
A4 – red 3.3 V power supply

2 Conclusion

The document describes how to configure and use built-in self-tests (BIST) in SPC570Sx device. Typical usage of the BIST is to detect latent failures before the user code is executed. Two different BISTs can be used: LBIST (logic BIST) and MBIST (memory BIST). There are available only off-line BISTs, which can be executed after power-on reset, if the BISTs have been configured properly in advance in UTEST Flash by DCF records.

It shows how to modify BISTs setup practically even the DCF records are already stored in nonerasable UTEST Flash.

It provides three typical BIST configurations together with practical performance measurements.

Revision history

Table 16. **Document revision history**

Date	Revision	Changes
05-May-2022	1	Initial release.

Contents

List of tables

List of figures