TN1239
Technical note

How to change DCF STCU configuration

Introduction

The SPC58 family integrates a dedicated safety module to configure, control and execute self-test operations. This module is called Self-test Control Unit (STCU2).

The user can configure the STCU2 in two ways. The first way uses the standard register access from user application. The second way exploits the Device Configuration Format (DCF) records.

This document explains the correct procedure to update an STCU configuration which has been already set by the specific DCF records in UTEST memory. It focuses only on the second way to configure the STCU.

As prerequisite the reader shall be familiar with the functionality of STCU2, DCF, SSCM and UTEST flash. For any details on these topics refer to the Reference Manual (see *Section A.2: Reference documents*).

Content

1 Programming of STCU2 3

2 How can STCU2 configuration be updated 4

2.1 Using invalidation keys 4

2.1.1 Limitation and prerequisites 5

2.2 Using standard unlock KEY2 5

2.2.1 Limitation and prerequisites 5

3 Conclusion 6

Appendix A Further information 7

A.1 Acronyms 7

A.2 Reference documents 7

Revision history 8

1 Programming of STCU2

The normal flow to program STCU2 starts by providing the unlock keys and afterwards the rest of the configuration (left side of the *Figure 1*).

Unlock keys are static keys. The user can find their values in the STCU chapter of the reference manual.

The STCU loads the configuration only if the user provides the correct UNLOCK keys. If the unlock keys are not correct, the STCU ignores the following configuration.

The user shall pass these values to the STCU by writing them into the UTest sector of the Flash as DCF records as shown on the left side of *Figure 1*.

A problem occurs in the case the user needs to change the STCU configuration. It means that the user has programmed at least a pair of unlock keys into UTEST.

Given that the STCU2 considers only the first valid unlock keys and the following configuration. After the first acceptance of a configuration STCU2 does not accept any other settings even if the provided unlock keys are correct. The SSCM reads the DCF incrementally from lower to higher addresses.

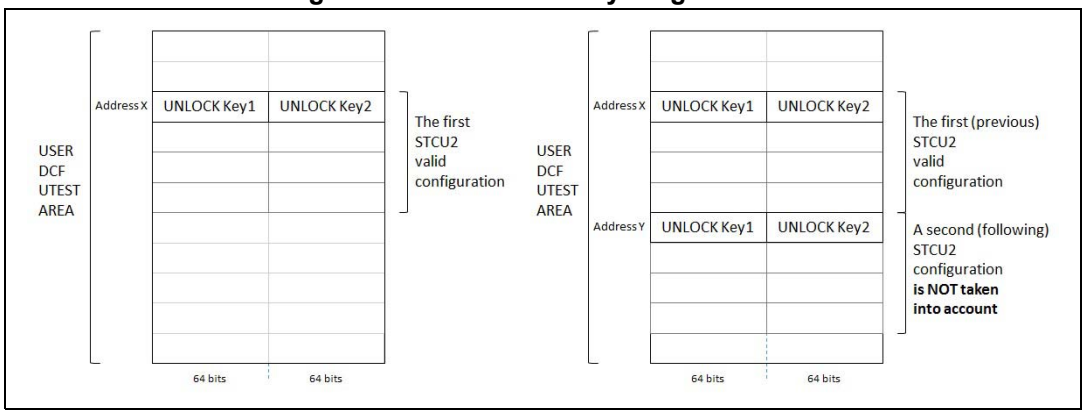
Right side of *Figure 1* shows two different configurations of the STCU. Both configurations start with correct unlock keys:

- First configuration starts at Address X, and
- Second configuration starts at Address Y.

Even if the second configuration starts with correct unlock keys, the STCU ignores it. As a result, the STCU runs the L/MBIST accordingly with the configuration that starts at Address X.

Section 2 explains how to instruct the STCU to load the second configuration and discard the first one.

Figure 1. STCU unlock keys organization



2 How can STCU2 configuration be updated

2.1 Using invalidation keys

As explained in the previous chapter, simply writing a second STCU configuration at different location does not cause any change in the configuration that the STCU loads, i.e. the first one.

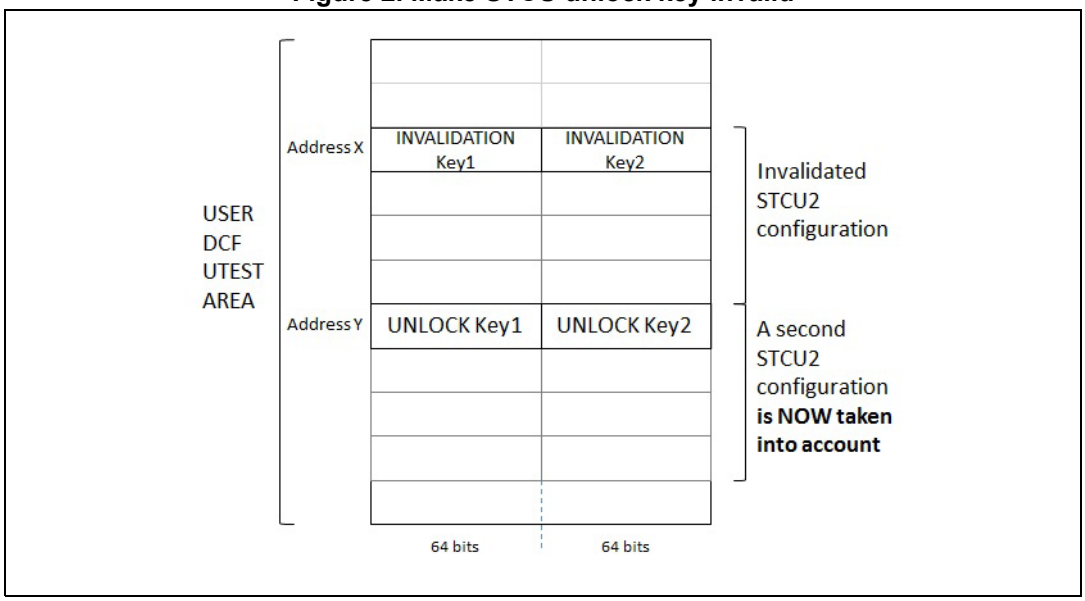
Since the UTest sector is not erasable, if the user wants to change the STCU configuration he must invalidate the first configuration. It means invalidating the first instance of the unlock keys.

As consequence, STCU2 reads unlock keys that are not correct and ignores the following configuration.

With reference to *Figure 2*, if the user invalidates the unlock keys at address X the STCU ignores the first configuration and loads the second one.

If needed, the user can repeat the process by adding a third configuration and invalidating the unlock keys at address Y and so forth.

Figure 2. Make STCU unlock key invalid



Invalidating means overwriting the UNLOCK keys with not correct keys. The user, however, shall do this operation carefully due to the ECC/EDC.

The ECC/EDC protects the UTest content. It means that the UTest contains not only the data, but also some redundant bits. The hardware can correct or detect some errors by comparing the data and the redundant bits. This process is transparent from the standpoint of the user who can't access the redundant bits.

For this reason, if the user overwrites the UNLOCK keys with random data there the risk that the ECC/EDC logic detects one or more not correctable errors. In case of this event the sample doesn't start.

Under those circumstances, the user must use specific values that doesn't cause the ECC/EDC errors. These values are visible in *Table 1*.

Table 1. Physical values of unlock and invalidation keys

	Key 1	Key 2	Complete DCF key 1	Complete DCF key 2
Unlock	0xD3FEA98B	0x2C015674	0xD3FEA98B00080008	0x2C01567400080008
Invalidation	0xD3BCA98B	0x28011674	0xD3BCA98B00080008	0x2801167400080008

2.1.1 Limitation and prerequisites

- If the user invalidates the KEY1 and KEY2, the FCCU signals an SSCM_XFER_FLASH_ERR_MEMORY_ERR
- These INVALIDATION keys are effective only if UNLOCK keys are aligned at 128bits, otherwise the ECC must be computed explicitly for given data
- The user shall use the INVALIDATION keys only in case valid UNLOCK keys are saved in UTEST and a new configuration shall be applied
- INVALIDATION keys must be written at exact positions of UNLOCK keys = overwrite the UNLOCK keys
- These INVALIDATION keys are not effective once the sample became OPP, that is starting from customer delivery
- STCU does not allow to overwrite BYPASS. So key invalidation is required if we have already bypassed STCU and need to re-program it for a valid MBIST/LBIST run

2.2 Using standard unlock KEY2

First of all, this procedure is preferred to *Section 2.1: Using invalidation keys*.

This procedure requires programming of DCF for standard STCU2 unlock KEY2 followed by the new configuration of STCU2. The new configuration can contains only the changes which are required.

This procedure is actually introduced in RM of a SPC58x device in following form (i.e. STCU2 SK Code Register (STCU_SKC) chapter).

To extend the STCU2 register access cycles before the hard-coded WDG timeout expires, only Key2 needs to be applied. The effect of this write operation is to re-initialize the WDG timeout counter. Key1 must not be applied or a a transfer error on the IPS or SSCM bus is asserted depending on the selected source. The STCU2 write access is locked until the correct sequence is applied.

2.2.1 Limitation and prerequisites

This standard unlock KEY2 is not effective when the user what to bypass the STCU

As in this case, the bypass bit is set in a second step (i.e. after a self-test configuration already programmed), the self-test still stays enabled since the internal state machine has already moved ahead due to RUN bit, WDG keeps on running and device would come out of reset after WDG expiry.

3 Conclusion

There are two ways to update the STCU2 configuration.

1. Using invalidation keys

The update of STCU2 configuration is possible even if there are strict rules which the user must consider.

If the user follows these rules this is not risky operation. In addition, change of STCU2configuration is an exceptional state which occurs mainly during developing phase and not in final application.

The application must manage the SSCM_XFER_FLASH_ERR_MEMORY_ERR fault in the FCCU, if the user invalidates the KEY1 and KEY2.

2. Using standard unlock KEY2

This is preferred solution introduced even in RM and brings no specific limitation.

Appendix A Further information

A.1 Acronyms

Table 2. Acronyms

Acronym	Name
STCU2	Self-test control unit
DCF	Device configuration format
SSCM	System status and configuration module

A.2 Reference documents

SPC58NE84C3, SPC58NE84E7, SPC58EG84C3 reference manuals, Rev2, Feb 2016.

Revision history

Table 3. Document revision history

Date	Revision	Changes
08-Sept-2016	1	Initial release
01-Jul-2020	2	Added *Section 2.1: Using invalidation keysSection 2.2: Using standard unlock KEY2* and updated *Section 2.1.1: Limitation and prerequisites*. Added *Section 2.2.1: Limitation and prerequisites*.