TN1234
Technical note

TDM: SW Tamper Region override

Introduction

The aim of this technical note is to deepen a particular feature of the TDM, for example the software overriding of the tamper regions.

TDM module protects the Flash memory from illegal modification of its content. TDM forces the user to leave a signature[a] in a diary, also known as Tamper Detection Region (TDR), before the Flash block(s) can be erased. TDM includes 6 TDR that can be associated with one or more Flash blocks.

Since each TDR has a limited space, for instance 2KB, and the user writes a signature before each Flash erase event, there is the risk to fill the diary.

An override operations permits to overcome the protection provided by TDM itself. The user has 2 choices to perform the override, either by software or by DCF.

For details about how TDM works refer to the RM and to the AN4557 (see *Section A.1: Reference documents*).

The software Tamper Region override feature described in this document applies also on other SC57xx and SPC58xx devices.

a. The user shall choose the values of the signature. Protection provided by the TDM does not depend on the values of the signature.

Contents

List of figures

1 Overview

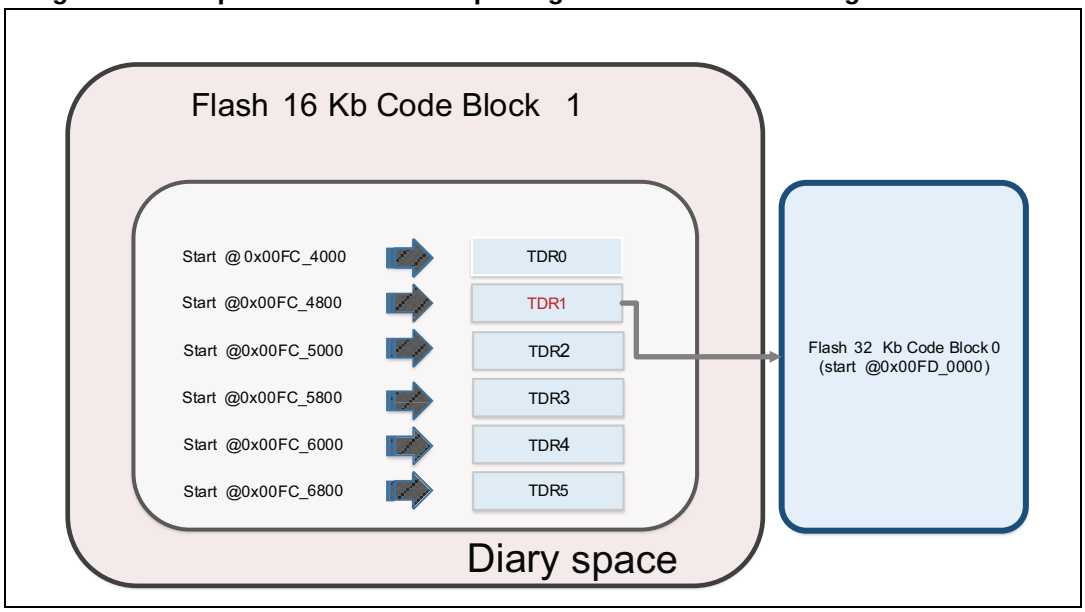
TDM is a security module that forces users to write a signature into a sort of diary, for instance TDR, before they can erase the related Flash blocks.

TDM implements six tamper regions that are saved in the Flash memory. The user configures the start address of this diary by a DCF records, for instance 'Diary Base Address'. The structure of the diary and the included TDR are shown in *Figure 1*.The size of each diary is 2 Kb.

There is no restriction on what signature to program into the diary region. TDM unlocks the erase operation as soon as the user writes "something in the Diary. The aim of TMD is to keep track of erase operations.

Another security module, for instance PASS, protects the erasing and programming by using one or more passwords.

Figure 1. Example about how a temper region is associated to region of the Flash



User can protect Flash sectors from erasing by connecting them with TDR. *Figure 1* shows an example on this link:

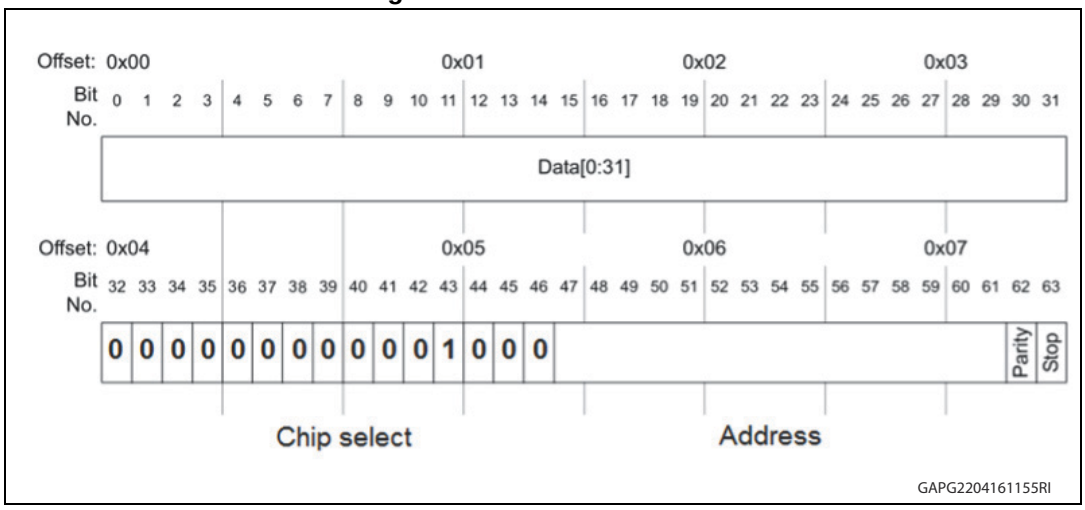
- The diary is located in *Flash 16Kb Code Block 1*:
  - 0x00FC_4000 ÷ 0x00FC_7FFF
  - User can configure this base address accordingly its needs
  - To avoid that an hacker modifies the Diary content, the Diary space has to be configured as OTP
- The TDR1 is linked to *Flash 32 kB Code Block 0*
  - Before erasing the *Flash 32 kB Code Block 0*, the user has to write a signature within tamper detection region 1
- The user does not need to link every Flash block to as TDR. It is important that the user cannot link a Flash block to more than a single TDR.

1.1 DCF for TDM

The user can configure most of the TDM features by programming some DCF records:

- Set Diary Base Address
- enable/disable the override feature[(b)]
- enable/disable the software Tamper Region Override[(c)]
- enable the OTP in the zone of diary (by the DCF ->OTP)[(d)]
- associate the Tamper region ‘x’ to Flash memory ‘y’ by the TDRx_LOCKy DCF record
  - if a Flash block is not connected to any TDR, TDM does not protect it.

Figure 2. DCF for TDM client



User can configure the TDM as shown in *Figure 1* by writing the following DCF records:

- DCF Diary Base Address
  - 0x00FC_4000_0020_0000
- DCF TDR1_LOCK0
  - 0x0040_0000_0020_0060

1.2 How to erase in Flash with the diary enabled

This paragraph summarizes how to execute an erase operation of a Flash block linked to a tamper region. For example how to erase *Flash 32 kB Code Block 0* that the user has linked to the TDR1 as shown in *Figure 1*.

By default, the erase operation of Flash blocks linked to any Tamper Regions is locked. The TDRSR register of the TDM reports the status of this protection:

- TDRSR register = 0x3F

b. If override disabled by DCF records, once a TDR is full the Flash sectors linked to this TDR cannot be erased any more.

c. This DCF record disables the possibility to enable the tamper protection override by software. In this case the user can still enable the override by writing a specific DCF record.

d. This configuration is very important. If the diary is not OTP, a hacker can reset its content.

Once the user links a Flash block to a TDR[e], the erase operation of this Flash block is conditioned by a successfully programming operation within the associated TDR.

Considering the example in *Figure 1*, in order to erase the Flash 32 kB Code Block 0 the user has to:

1. program one or more double words in an „empty “location within TDR1
   - this operation unlocks the erase the Flash 32 kB Code Block 0 and other Flash blocks linked to this TDR
   - TDM_TDRSR = 0 x 3D reports that block linked to TDR1 is not locked anymore.
2. Erase the *Flash 32 kB Code Block 0*

Warning: User shall set the Program/Erase Complete Interrupt Enable (PECIE) bit in the Flash Module Configuration Register (MCR) before doing any Flash block erase operation which includes TDM diary update operation. The PECIE interrupt does not need to be processed by the Interrupt controller. To prevent the interrupt from being processed its priority should be left at the default of 0 within the Interrupt controller.

- Set Flash.MCR.PECIE = 1
- Program the TDR1
- Unset Flash.MCR.PECIE after programing

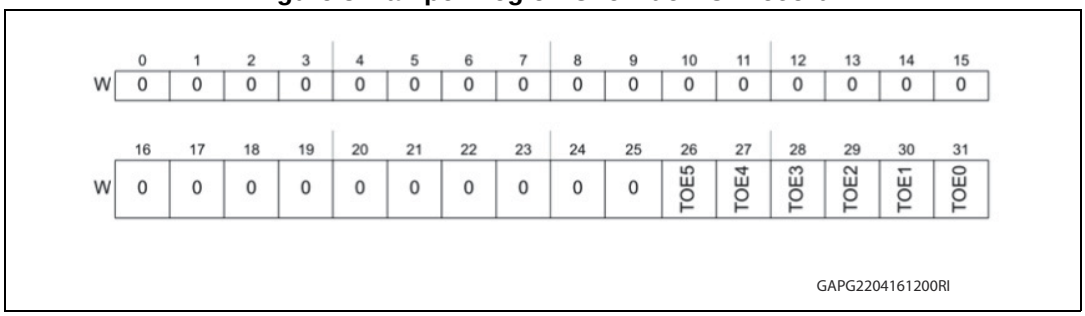
1.3 Different types of Tamper Region Overriding

When the space of a Tamper Region fills up, the Flash blocks, linked to this Tamper Region, couldn’t be erased. The override operation permits to bypass the protection of TDM module. Once the users enable the override, they don’t need to write any signatures before erasing the Flash blocks linked to the TDR.

Two kinds of overriding are possible:

1. by writing the TOEx flag in the “Tamper region Override” DCF record, or

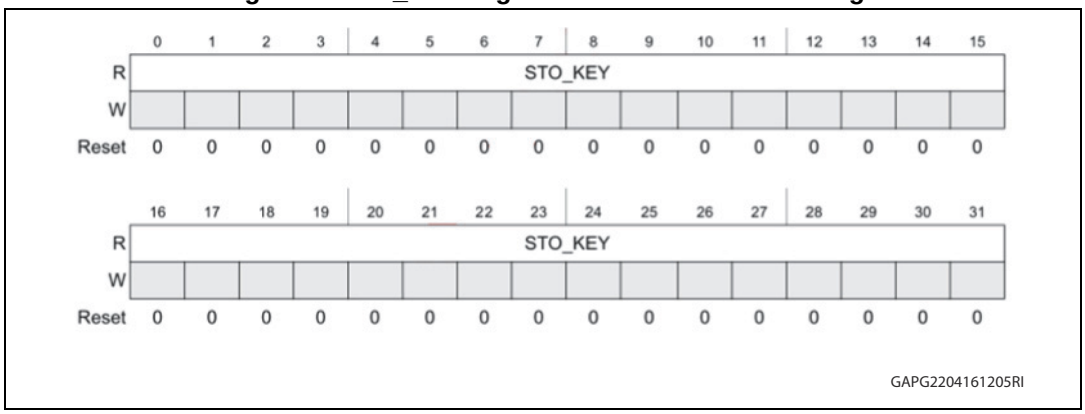
Figure 3. Tamper Region Override DCF record



2. by writing a known signature, for instance 0x55AA5A5A, within the register STO_KEYx

e. By the TDRx_LOCKy DCF registers

Figure 4. STO_KEY register for software overriding



By writing the ‘Software Tamper Region Override Disable’[f] DCF record, the user can decide to remove the possibility to perform the overriding via the STO_KEY register.

f. STO_DISx field of the STO_DIS_DCF(Software Tamper Region Override Disable)

2 Software Tamper Region override

Once the software writes the correct signature, for instance 0x55AA5A5A, into the STO_KEYx register, the Tamper Detect Region “x” is overridden. TDM does not protect anymore the Flash blocks linked to this TDR.

User shall consider that the STO_KEY is a “write only once” register, for instance software can write this register only once per reset cycle.

The BAF software writes this register by its own. It means that application code can’t write directly the STO_KEYx registers.

BAF, however, implements a sort of handshake with the application software. Before the tamper region is full, the application software shall write the correct signature, for instance 0x55AA5A5A, in the first 32 bit of the last double word of the tamper region. After each reset the BAF copies data from this location to the STO_KEYx register. As result the TDM protection is overridden for ever.

Considering the example of *Figure 1*, if TDR1 is the diary region selected, application software shall write the signature in the location 0x00FC4F(g)F8 that contains the first 32 bit of the last double word of the diary.

After each reset, BAF copies data from 0x00FC4FF8 to the STO_KEYx register. If 0x00FC4FF8 contains the correct signature, TDM protection is removed for Flash blocks linked to TDR1.

Tamper region status register (TDRSR) indicates that TDR1 is unlocked(h) and user can erase *Flash 32 kB Code block 0* without leaving a signature within the diary.

To summarize:

- TDM implements a one time programming register to enable the override, for instance STO_KEYx. One time programming refers to once per reset.
- Software writes a known signature to this register to activate the override.
- Application software cannot do it, because after each reset the BAF copies data from the last entry of the tamper region to the STO_KEYx register(i).
- To activate the override, application software shall write the correct signature to the last entry of the tamper region.
- Afterwards, during boot process, BAF copies the correct signature into the STO_KEYx and the TDM protection for Flash blocks linked to the specific tamper region is removed forever.

g. The 0x00FC_4FF8 address is obtained considering:

0x00FC_4000 (start of TDR1) + 0x800 (each TDR is large 2K) – 0x8

h. TDRSR=0x1D.

i. By default the last entry of the tamper region contains all ‘1b’. Then BAF copies 0xFFFF_FFFF into the STO_KEYx register. TDM doesn’t recognize it as correct signature and doesn’t enable the override.

3 Summary

The device implements two possibility of overriding, for instance either by software or by DCF. This TN gives some hints on the software overriding.

User can enable the override by software by writing the last entry of the Tamper Region. Once user overrides the TDM protection, it cannot be enabled anymore.

Appendix A Further information

A.1 Reference documents

- *SPC58NE84x/SPC58xG84x 32-bit Power Architecture® microcontroller for automotive ASILD applications* (RM0391, DocID027214)
- *Introduction to the usage of TDM peripheral SPC57xx devices* (AN4557, DocID026704)

A.2 Acronyms

Table 1. List of acronyms

Acronym	Name
TDM	Power Management Controller Digital Interface
TDR	Tamper Detection Region
BAF	Boot Assist Flash
OTP	One Time Programming

Revision history

Table 2. Document revision history

Date	Revision	Changes
26-Apr-2016	1	Initial release.