Data integrity check in SEMPER™ NOR flash

About this document

Scope and purpose

This application note describes the data integrity check feature in SEMPER™ NOR flash memories, and provides guidelines and suggestions to implement host application software.

Table of contents

About this document 1

Table of contents 2

1 Introduction 3

2 CRC algorithm 4

3 Implementation 5

4 Use cases 6

4.1 Verifying program data 6

4.2 Verifying stored data 6

5 Conclusion 7

References 8

Revision history 9

Disclaimer 10

1 Introduction

Electronic components are essential in the modern world. When it comes to selecting a reliable digital memory storage, NOR flash memory is often the choice. Infineon has been providing reliable NOR flash memory solutions for many years to many applications in different areas including automotive, industrial, networking, and consumer space. As the technology improves over time, the demand for flash memory to provide not only a reliable storage, but also a fail-safe one becomes higher and higher. Automotive industry is leading the way of defining functional safety standards, such as ISO 26262. The goal of ISO 26262 is to provide a unifying safety standard for all automotive electronic systems.

With the focus on functional safety, Infineon recently launched SEMPER™ NOR Flash family that is designed to meet the functional safety standard from the design kick-off to mass production time. Furthermore, SEMPER™ Flash provides various features that allow users to detect, analyze, and recover from potential failures. These features include detection of initialization and configuration register corruptions, Error Correction Code (ECC) for memory storage, interface bit error checking, and a data integrity checking function that will be discussed in this application note.

The reliability of an application depends heavily on the correctness of the program memory. Any unintended alteration of the program memory will inevitably lead to a malfunction of an application. Such alterations may be caused by many reasons, for example, software defects, exposure to extreme temperatures, or an intentional attack from a hacker. To ensure the functional safety of the application, it is necessary to check the data integrity before using for an application. One of the quick ways to verify data integrity is using a Cyclic Redundancy Check (CRC) algorithm. This application note describes the basics of the CRC algorithm and some use cases on how to use it for verifying data integrity.

2 CRC algorithm

The basic idea of CRC algorithm is simply to treat the original data as an enormous binary number, divide it by another carefully chosen fixed binary number, and then store the remainder from this division along with the original data. In a transmission line example, the remainder is transmitted along with the original data to the receiver. The receiver will verify the original data integrity by doing the same division and comparing the remainder with the original one. If the remainder values are different, it means the data integrity has been compromised, either in the original data portion, or in the remainder portion.

In CRC world, this divisor for the division calculation is depicted by a polynomial, specifically called Generator Polynomial. Mathematically, not all polynomials are good divisors in CRC algorithms. Scientific researches have shown that some polynomials are better suited than others in terms of higher chances of detecting possible errors. Some of the better suited polynomials were chosen by the CCITT (now called ITU-T, International Telecommunications Union) and are known as the CRC-CCITT. There are different CRC-CCITT polynomials available that differ in their size and consequently in the error burst lengths that can be detected. For example, the CRC-CCITT (CRC-16) corresponds to polynomial $x^{16}+x^{12}+x^5+1$, which represents the value `0x1021` (0001000000100001b) in binary notation (x^{16} ignored for 16 bit).

Table 1 shows some of the most commonly used polynomials for 16- and 32-bit CRCs.

Table 1 International Standard CRC Polynomials

Name	Generator polynomial	Notation
CRC-CCITT	$G(x) = X^{16} + X^{12} + X^5 + 1$	0001000000100001b
CRC-16	$G(x) = X^{16} + X^{15} + X^2 + 1$	1000000000000101b
CRC-32	$G(x) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$	00000100110000010001110110110111b

These common CRC algorithms can detect the following type of errors:

- All single-bit and double-bit errors;
- All odd number of errors, if the constraint length (length of redundant bits added) is sufficient;
- Most burst errors

Semper flash family implements a hardware CRC-32 engine that uses one of these common polynomials. The Generator Polynomial in Semper flash is:

$G(x) = X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$.

It can be represented by this binary string: 00011110110111000110111101000001b, or hex 0x1EDC6F41, without the X^{32} term.

The remainder from a CRC calculation is often called CRC checksum. As mentioned earlier, the remainder is often kept with the original data to verify their integrity in the future or at the receiving end of the transmission.

3 Implementation

SEMPER™ Flash family implements a hardware-accelerated CRC calculation engine. It can calculate the CRC checksum over a user-defined address range in the memory array.

To calculate the CRC checksum over a certain memory range, the host can issue a specific CRC command. The command loads the beginning and the end addresses into CRC Start and End Address registers, identifying the beginning and end of the address range. Upon receiving the command, the hardware CRC engine inside the SEMPER™ Flash device will start an embedded operation to calculate CRC checksum. The device will be in the busy state during the calculation time. The host can check the status register to verify the completion of the CRC calculation. Once the operation is completed, the CRC checksum is stored in the Data Integrity CRC Register, and can be read out to the host with the Read Any Register command.

If the block of data is bigger than 1 KB, the hardware CRC engine can calculate the CRC checksum very quickly; the rate is about 65 MB/sec. That means for a 512 Mb device, calculating the CRC checksum over the entire memory space will only take about 1 second.

4 Use cases

Here are a couple examples that illustrate how users can benefit from the CRC engine inside SEMPER™ Flash.

4.1 Verifying program data

To ensure that the data is programmed to the flash correctly, most programmers will compare the data written to the flash with the original data. This is reasonable when the length of the programmed data is not too long. If it is, it will take a long time to verify data by comparison.

SEMPER™ Flash's CRC engine provides a fast and convenient way to verify the data programmed to the flash device. Do the following to verify the data:

1. Calculate the CRC checksum for the original data. The software for CRC calculation can be found in the referenced document. Retain the calculated checksum on the host
2. Program the data to the flash using the maximum page size that the flash device allows. You don't need to verify the data after each program completion
3. After all data is programmed, issue the CRC command with the start and end addresses to the flash and receive back the CRC checksum

Compare the CRC checksum obtained at Step 1 with the checksum from the SEMPER™ Flash device. If they are the same, that means all data have been programmed correctly.

4.2 Verifying stored data

To prevent corruption of critical data caused by malicious hackers or by accident, it is a good idea to verify data integrity before using the data. The following is an example of how to use CRC to verify data integrity.

1. Program the data to the flash device, verify that all data is programmed correctly. (Use Case A shows how to verify data integrity.)
2. Store the original data CRC checksum in the OTP region in the flash device. OTP is a one-time programmable region where no one can erase the data after programming. If the checksum is stored in the OTP region, it cannot be replaced by a new checksum value which a hacker may attempt to do
3. When required to verify data integrity, read the original checksum from the OTP region. Issue the CRC command to get the new CRC checksum value for the stored data
4. Compare these two checksum values. If they are the same, that means the data is still intact

5 Conclusion

SEMPER™ NOR Flash family provides an internal hardware CRC engine that can be used to efficiently verify the data integrity.

References

[1] Infineon Technologies AG: *SEMPER™ Flash*; available online

[2] Infineon Technologies AG: *CRC-32 Parameters Used in S26KL-S/S26KS-S HYPERFLASH™ Family - KBA223179*; available online

Revision history

Document version	Date of release	Description of changes
**	2018-12-18	Initial release
*A	2021-03-17	Updated to Infineon template
*B	2026-08-31	Template update; no content update.

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2026-08-31

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2026 Infineon Technologies AG

All Rights Reserved.

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
IFX-gkg1765268114046

Important notice

Products which may also include samples and may be comprised of hardware or software or both ("Product(s)") are sold or provided and delivered by Infineon Technologies AG and its affiliates ("Infineon") subject to the terms and conditions of the frame supply contract or other written agreement(s) executed by a customer and Infineon or, in the absence of the foregoing, the applicable Sales Conditions of Infineon. General terms and conditions of a customer or deviations from applicable Sales Conditions of Infineon shall only be binding for Infineon if and to the extent Infineon has given its express written consent.

For the avoidance of doubt, Infineon disclaims all warranties of non-infringement of third-party rights and implied warranties such as warranties of fitness for a specific use/purpose or merchantability. Infineon shall not be responsible for any information with respect to samples, the application or customer's specific use of any Product or for any examples or typical values given in this document.

If this document is marked as 'Preliminary', 'Target', 'Draft' or 'Proposal', Infineon reserves the right to change all information given in this document at any time without notice. Subject to the development and release of a Product for series supply by Infineon, the technical specifications of the Product are set forth in the relevant datasheet provided by Infineon without such marking.

The data contained in this document is exclusively intended for technically qualified and skilled customer representatives. It is the responsibility of the customer to evaluate the suitability of the Product for the intended application and the customer's specific use and to verify all relevant technical data contained in this document in the intended application and the customer's specific use. The customer is responsible for properly designing, programming, and testing the functionality and safety of the intended application, as well as complying with any legal requirements related to its use.

Unless otherwise explicitly approved by Infineon, Products may not be used in any application where a failure of the Products or any consequences of the use thereof can reasonably be expected to result in personal injury.

However, the foregoing shall not prevent the customer from using any Product in such fields of use that Infineon has explicitly designed and sold it for, provided that the overall responsibility for the application lies with the customer. Infineon expressly reserves the right to use its content for commercial text and data mining (TDM) according to applicable laws, e.g. Section 44b of the German Copyright Act (UrhG). If the Product includes security features:

Because no computing device can be absolutely secure, and despite security measures implemented in the Product, Infineon does not guarantee that the Product will be free from intrusion, data theft or loss, or other breaches ("Security Breaches"), and Infineon shall have no liability arising out of any Security Breaches.

If this document includes or references software:

The software is owned by Infineon under the intellectual property laws and treaties of the United States, Germany, and other countries worldwide. All rights reserved. Therefore, you may use the software only as provided in the software license agreement accompanying the software.

If no software license agreement applies, Infineon hereby grants you a personal, non-exclusive, non-transferable license (without the right to sublicense) under its intellectual property rights in the software (a) for software provided in source code form, to modify and reproduce the software solely for use with Infineon hardware products, only internally within your organization, and (b) to distribute the software in binary code form externally to end users, solely for use on Infineon hardware products. Any other use, reproduction, modification, translation, or compilation of the software is prohibited.

For further information on the Product, technology, delivery terms and conditions, and prices, please contact your nearest Infineon office or visit www.infineon.com.