CYPRESS EMBEDDED IN TOMORROW™

AN98555

High Reliability Programming Methodology for Floating Gate Flash

AN98555 discusses Pseudo Single Bit Charge Loss (pSBCL) related flash failures, real customer implementation expectations, experiences, and questions.

1 Abstract

Cypress ships Automotive Grade Floating Gate Flash Memory devices to many market segments through out the world. Automotive, Aerospace, and some other markets typically make necessary investments to improve system reliability. Along these same lines this document recommends steps to be taken to mitigate Pseudo Single Bit Charge Loss (pSBCL) related flash failures. This application note discusses pSBCL failure analysis data, real customer implementation expectations, experiences, and questions.

The goal of this application note is to provide information which supports the following items:

- Clarify the differences between Single Bit Charge Loss (SBCL) and Pseudo Single Bit Charge Loss (pSBCL).
- Show pSBCL is an application induced failure and not a device related failure.
- The occurrence of Pseudo Signal Bit Charge Loss (pSBCL) is a rare phenomenon resulting in a very low ppm failure rate. Given this failure is a low ppm issue typically implies that customers with strong quality goals to eliminate the re-occurrence of all identified failures will pursue the recommended corrective action.

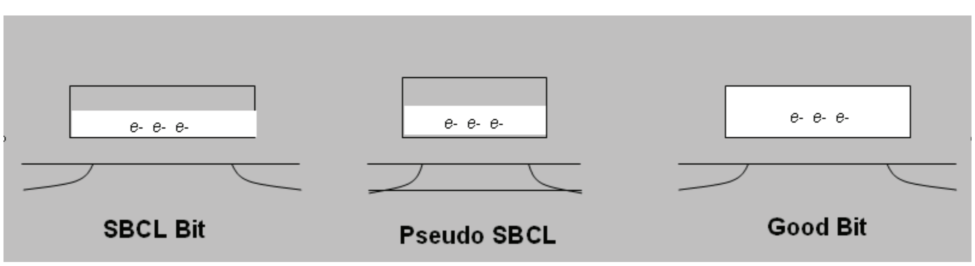
2 Pseudo_SBCL Definition and Prevention

2.1 Defining pSBCL

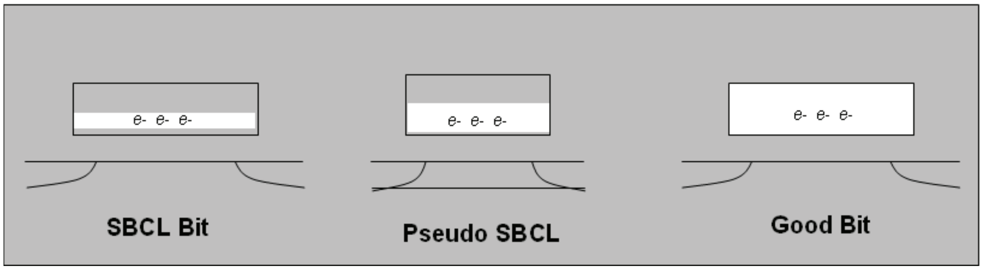
A Single Bit Charge Loss (SBCL) cell is a defective memory cell which exhibits electron leakage. The electron leakage from the floating gate can be accelerated with appropriate voltage or high temperature stress tests. The subject SBCL cell does not have good data retention which is equal to defective bit. The Cypress CCAR uses standard test processes to characterize and identify defective SBCL cells.

A pSBCL cell is a physically good memory bit which had not been fully programmed or had been marginally programmed. The cell has no physical anomalies and does not exhibit charge leakage from the floating gate under normal conditions or when subjected to voltage or high temperature stresses. The subject pSBCL cell has good data retention which is equal to good bit. The following characterizes the differences in the behavior and identification of SBCL and pSBCL cells.

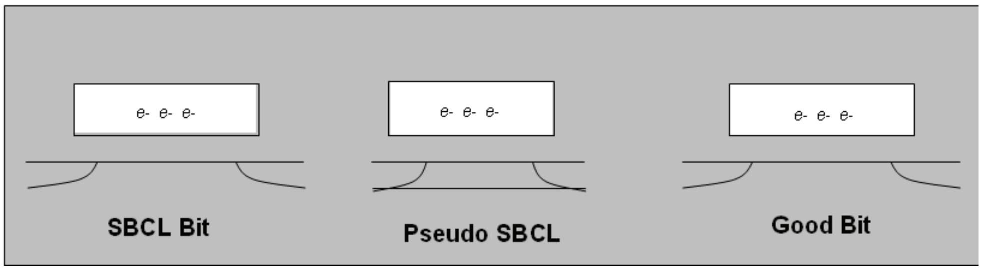
- Initial Analysis of true SBCL and pSBCL appear very similar; the subject reported SBCL and pSBCL are compared to a known good programmed cell.



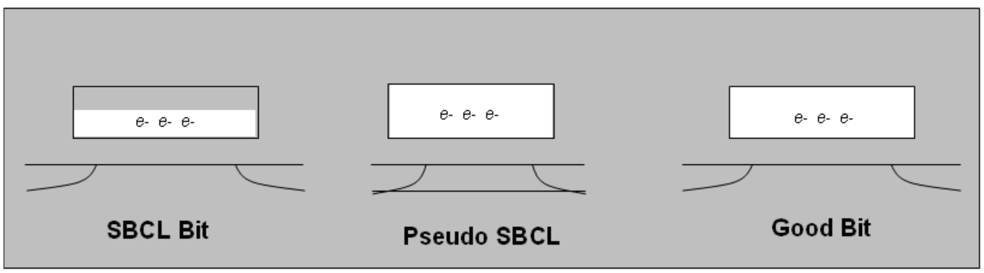
- Note after standard Electrical and Thermal Stress Testing there are differences in the cell charge levels. See below.



- SBCL Cell, pSBCL Cell, and Good Cell all reprogram the same.



After standard Electrical and Thermal Stress Testing the true SBCL cell loses charge while a pSBCL and a known good cell do not have charge loss.



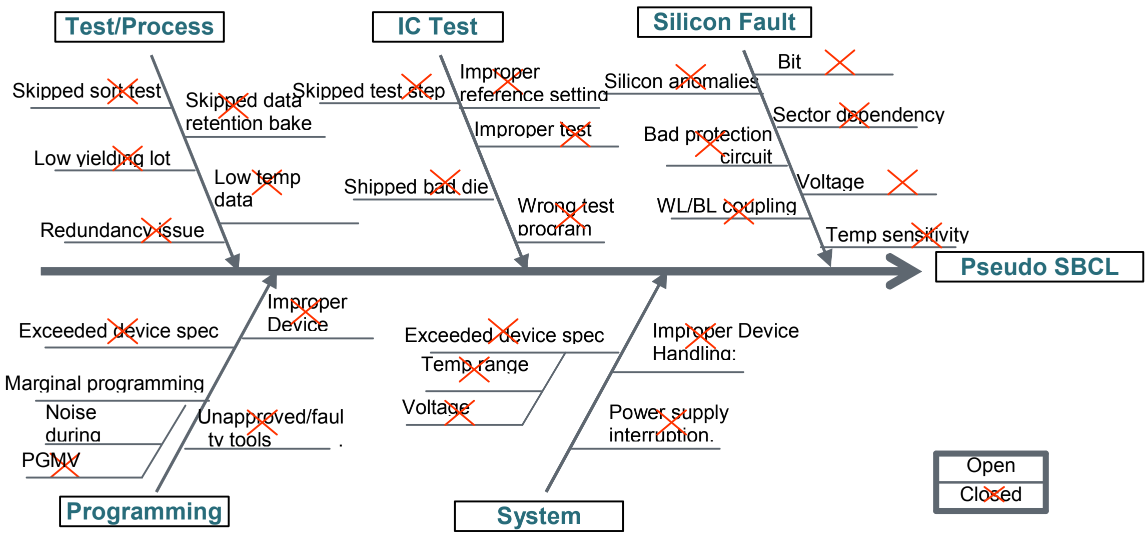
2.2 Cause and Prevention of pSBCL

The following sections provide explanations of the possible causes and prevention of pSBCL.

2.2.1 Possible Causes of pSBCL: Fishbone Analysis

Figure 1 shows the major areas that have been investigated to determine the possible cause(s) of pSBCL.

Figure 1. Pseudo SBCL Fishbone



The following is a high-level explanation for each leg of the Fishbone Diagram:

2.2.1.1 Test/Process

- Skipped SORT: Lot history confirms all pSBCL units were sorted.
- Low yield: Lot yields for all pSBCL failures are normal.
- Redundancy:
  - pSBCL failures are seen on devices w/o redundancy also.
  - Failures are not necessarily seen on redundant columns
- Skipped DRB: Lot history confirms all wafers received DRB.
- Low Temp DRB: No CL seen with 25°C and 50°C bake.

2.2.1.2 IC Test

- Skipped test step: Lot history confirms no tests were skipped.
- Wrong program: All programs have SBCL screens.
- Improper references: All references on pSBCL failures are within limits.
- Improper test limits: All test limits were confirmed to meet specs.
- Shipped bad die:
  - All returned pSBCL units pass production programs.
  - Test lot unit quantities properly tallied (all units accounted for).

2.2.1.3 Silicon Fault

- Silicon anomalies: No physical anomaly seen during DA at the failing address.
- Bad protection Circuit: No defects seen on ESD protection circuitry.
- WL/BL Coupling: No read margin issues upon reprogramming.

- Sector Dependency: Failures seen in various sectors.
- Voltage/Temp sensitivity: Units passed across V_{CC} and temperature upon reprogramming.
- Bit Interaction: No failures noticed with CKBD and Master data pattern DRB.

2.2.1.4 Programming

- Exceeded Spec.: Programming times on pSBCL failures meet the spec and did not show any abnormality.
- PGMV Defect: Failing location was reprogrammed successfully in its original condition, indicating PGMV successfully detected the bit was not fully programmed.
- Improper Handling: No evidence of EOS.
- Unapproved/Faulty Tools: Cypress's test equipment are calibrated and certified routinely compliant to TS16949.

2.2.1.5 System

- Exceeded Device Spec:
  - Temp/Voltage: Customer confirmed system environment complies to data sheet temp and voltage range.
- Improper Handling: No evidence of EOS and package/lead integrity has been verified.
- Power supply:
  - No evidence of EOS in the units.
  - Transients still a possibility.

The results of this analysis process showed that the marginally programmed cell was due to presence of noise or transients during the programming operations. The following sections provide additional background and explanations which support these findings.

2.2.2 Additional Background Information: Flash Programming Operation Details

During a Flash programming operation the voltage threshold (V_t) used in Embedded Programming is set at a higher level over the V_t used in Read operation. The Programming V_t is set in this manner to build in additional read margin and guarantee long term data retention. The tighter Program Verify V_t is used to detect any marginally program bits and the Flash State machine will automatically apply the required program pulse(s) to fully charge the subject bit.

2.2.3 Flip Bit Theory

The following highlights the Flip Bit Theory as a possible cause of marginally programmed bit, also known as “soft-programmed cell”. The presence of system noise or transients during programming operation can lead to “under-programming” or uncharged cell. High noise levels could corrupt or limit the programming V_t from reaching the required levels to fully program a cell. When the user performs a “Read Verify” a marginally programmed bit can be read as “0” or a “1”.

In the case where a marginally programmed bit is read as a “0” passing the customer verification testing the units become a probably candidate for a later pSBCL failure.

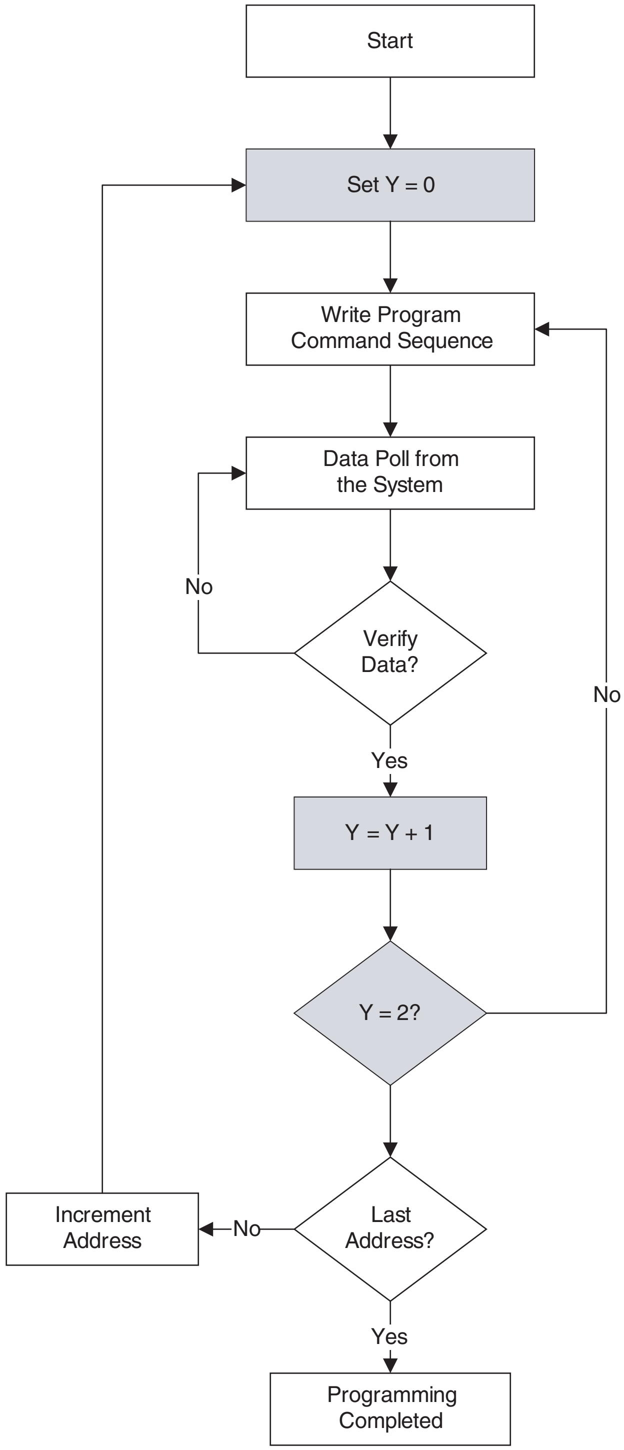
Investigations were made to re-create soft programmed cells in the test lab. Analysis showed the occurrence of real pSBCL was very low ppm value. Given the rare occurrence of soft programming in real application it was understood the probability to re-create soft programming in the lab would be very low. The lab results were as expected that no soft program cells were created.

2.2.4 Preventing pSBCL Flip Bits

Instituting an additional Program Verify routine immediately after the programming operation can insure the all bits were programmed to the targeted threshold voltage.

The second Program Verify function can be accomplished by modifying the Flash Software drivers to replace typical “Read Verify” after “Programming Completed” with more stringent “Program Verify” as shown in Figure 2. By re-executing the “Write Program Command”, “Program Verify” is leveraged which provides insurance against marginal programmed bits that could lead to a flipped bit or Pseudo-SBCL event. This change results in minimal overhead for the Flash programming process with any marginally programmed bit(s) being detected and programmed automatically. This process ensures quality of programmed bits in a very efficient manner.

Figure 2. Double Programming Algorithm



3 Program-Program Verify Implementation

pSBCL is not a device related failure but an application induced failure that is a rare phenomenon resulting in a very low ppm failure rate. Given this low occurrence of pSBCL, there are many customers who are not willing to implement the recommended corrective action "Program-Program Verify". The target customers are those who have strong quality goals to eliminate re-occurrence of an identified failure mode.

3.1 Psuedo_SBCL Failure Rates and Corrective Action Effectiveness

At the time of this writing Cypress had observed that multiple customers in the Automotive Market who have implemented double programming. One customer has a very demanding application and end customer. The application was a Transmission Control Unit (TCU) with application temperature requirements up to 145°C. The OEM's end customer place high demands on the OEM to eliminate all identified failures.

Cypress CCAR performed database analysis on the returns from the subject TCU program to understand the effectiveness of implementing "Program-Program Verify" to mitigate pSBCL failures. The CCAR database analysis was partitioned into two segments. Below are the pSBCL failure rates on this particular program prior to implementing "Program-Program Verify" and the analysis showing the new failure rate after implementing "Program-Program Verify".

- Standard single pass programming: ~6.7 ppm (Production: Two Years=> 750K units)
- "Program-Program Verify" Algorithm: 0 ppm (Production 12 months => 600K units)

This customer has realized a reduction in pSBCL failures by implementing "Program-Program Verify" the recommended best practice/corrective action.

3.2 Typical Customer Expectation and Questions

Each customer and potentially each program at a customer which implements "Program-Program Verify" will have unique requirements. The following is intended to highlight specific tasks that were required prior to or during the implementation of the "Program-Program Verify" algorithm. It is key to clarify for the Customer the differences between pSBCL and SBCL and the new "Program-Program Verify" algorithm. (See *Pseudo_SBCL Definition and Prevention* on page 1.)

Below is a collection of items the customer might request during "Double Programming Implementation."

- Program-Program Verify Overhead
  - Customer will request that Cypress provides information concerning the additional over head associated with Program-Program Verify algorithm implementation. There are estimates that the additional over head is approximately 20%, obviously the customer must run the new algorithms on their products to obtain a concise answer.
- Where are Code Updates required
  - Once implementation begins the customers' program might require the software algorithms to be updated in any or all of the following areas:
    - Module Code Updates.
    - Software Tools used update code.
    - Flash Programmer Algorithm Updates. Cypress personnel must work with Programmer Mfg to update the subject algorithm(s).
- Verifying Program-Program Verify Works
  - Some customer(s) will require that Cypress provide verification that the new software algorithm does fully charge a marginally programmed cell:

  Below is one methodology that was used to generate marginally programmed cells:

1. Identified "Programmed bits" in each sector of the customer pattern and recorded the BP address locations.
2. Programmed a "dummy" unit at those BP address locations and read the unit on the Nextest tester.
3. Recorded the Nextest address locations. (Different than BP addresses.)
4. The customer pattern was modified so that the identified bits could be programmed into a "Weakly" or marginally programmed state.

5. Each identified cell was pulsed on the Nextest tester to put the units into a weakly programmed state.
6. The IDS currents at those locations were then recorded for the respective sectors on each device.
7. The subject unit was subjected to Program-Program Verify on the customer module.
8. Cypress verified that charge level of the previously identified cells were now fully programmed.

3.3 Sample Customer Questions

Question 1

The flow chart in Figure 2 has gray blocks added in addition to the flow chart provided in a standard data sheet. The gray blocks are to add a second “Program Verify” step. Why doesn't the first “Program Verify” in the flow catch the weakly charged bit if the threshold is set higher during a “Program Verify”?

Cypress Reply

If there is noise or transient present during the first Program Verify the Program Reference Level can be corrupted or shifted by the subject noise or transient. Under these conditions the subject Cell Charge Level is verified with using an invalid Program Reference Level resulting in a marginally programmed cell.

Reference Section 3.2:

Possible cause for marginal programmed bit, also known as “soft-programmed cell”

Presence of system noise or transients during programming operation can lead to “under-programming”

High noise levels could limit the programming voltage from reaching the required levels to strongly program cells

Reading a marginal program bit can be a “0” or a “1”

Follow up to Question 1

Is this is a probabilistic phenomenon?

Cypress Reply

Yes

Question 2

Based on what is being said here; it is theoretically possible that even implementing the Program-Program Verify algorithm could result in us having the same issue, however the probably is very low (e.g. <<< 1 ppm).

Cypress Reply

Yes

Document History Page

Document Title: AN98555 - High Reliability Programming Methodology for Floating Gate Flash Document Number: 001-98555				
Rev.	ECN No.	Orig. of Change	Submission Date	Description of Change
**	–	–	09/15/2010	Initial version
*A	4958986	MSWI	10/12/2015	Updated in Cypress template
*B	5872216	AESATMP8	09/05/2017	Updated logo and Copyright.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Video | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709