## SERESSA 2022

$5^{\text {th }}$ to $9^{\text {th }}$ of December at CERN, Geneva

## Analyzing data extracted from radiation tests in advanced SRAMs

Juan A. Clemente, Universidad Complutense de Madrid (UCM)



## Agenda

1. Introduction and motivation
2. Extraction of simple events (SBUs) / multiple events (MCUs / MBUs)
3. Analysis of "false" Multiple Cell Upsets (MCUs) by accumulation

- Birthday statistics
- Correction of experimental data

4. Analysis of "false" Multiple Bit Upsets (MBUs) by accumulation

- Error Correcting Codes (ECC)
- Accumulation of events and ECC reliability


## 5. Conclusions

## 1-Introduction and motivation

## Introduction

$\square$ Accelerated radiation tests on SRAMs are a common way of estimating the sensitivity of a device in harsh conditions.
$\square$ OK, we test a device against radiation and... what do we get?


$\square$ What are these: Single Bit Events (SBUs), Multiple Cell Upsets (MCUs)...?

## Motivation

$\square$ Technology miniaturization (Moore's law) leads to more cell density.

- Increase of the SER/device.
- Also, increase of the \% of the MCU SER contribution.
- >+900\% MCU SER contribution between 180-nm and 22-nm nodes.
$\square$ MCU understimations lead to wrong estimations of the total SER.
$\square$ A correct (or at least, accurate) MCU extraction

A. Neale, M. Jonkman and M. Sachdev, Adjacent-MBU-Tolerant SEC-DED-TAEC yAED Codes for Embedded SRAMs, in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 62, no. 4, pp. 387-391, April 2015.


## 2-Extraction of simple events (SBUs) / multiple events (MCUs / MBUs)

## Definition of "bit interleaving"

## Bit interleaving

Manufacturing technique that physically separates bits belonging to the same word, so they are distant enough and they cannot be affected by the same particle.2 types of n-bit multiple events:

- Multiple Bit Upsets (MBUs): $n$ bits in the same word are flipped by the same particle. Difficult to recover by standard Error Correcting Codes (ECCs).
- Multiple Cell Upsets (MCUs): 1 bit is affected in n words. Each single error is easy to recover (just 1 bit per word).

© Kitz 2006


## MCU/MBU extraction with unscrambling

## Unscrambling

Information about the internal organization of the memory, provided by the manufacturer, who makes possible to establish a relationship between "logical addresses" and the physical positions of those bits in the XY layout of the memory.


## MCU/MBU extraction without unscrambling

## "Statistical" MBU/MCU extraction techniques

When unscrambling is not available, many authors have proposed techniques that identify MCUs by detecting statistical anomalies in the set of observed bitflips. For instance, XOR'ed values between addresses more abundant than they should be in a theoretical scenario where no MCUs can occur.

F. J. Franco et al., Statistical Deviations from the Theoretical only-SBU Model to Estimate MCU rates in SRAMs, in IEEE Transactions on Nuclear Science (TNS), vol. 64, no. 8, pp. 2152-2160, July 2017

## How to analyze data correctly?

1. Initialize the memory with a known pattern (i.e., $0 \times 55$ ).
2. Expose the memory under the radiation beam for a given time.
3. Read the memory contents to search errors provoked by radiation.
4. Group errors by multiplicity:

- Single Bit Upsets (SBUs): 1 particle $\rightarrow 1$ error
- Multiple Cell Upsets (MCUs): 1 particle $\rightarrow$ several errors in different data words.
- Multiple Bit Upsets (MBUs): 1 particle $\rightarrow$ several errors in the same data word.

5. Give a metric for the SBU/MCU sensitivity:
" "Cross section" (б): Probability of a single particle (proton, neutron, heavy ion...) to provoke an error in a memory bit.

$$
\sigma=\frac{\text { Number of events }}{\text { Particle fluence } \cdot \text { Memory size (bits) }}\left\{\begin{array}{c}
\sigma_{S B U}=\frac{\text { Number of SBUs }}{\text { Particle fluence } \cdot \text { Memory size (bits) }} \\
\sigma_{M C U-2 b i t}=\frac{\text { Number of } 2-\text { bit MCUs }}{\text { Particle fluence } \cdot \text { Memory size (bits) }} \\
\text { etc ... }
\end{array}\right.
$$

# 3-Analysis of "false" MCUs by accumulation 

Birthday statistics

## Correction of experimental data

## Accumulation of "false" MCUs in a radiation-ground experiment



SRAM columns

## Estimation of false MCU rates

## Idea of the "Birthday paradox".

$\square$ How many people we need to put in the same group so the probability of finding, at least 2 people with the same birthday, is greater than $50 \%$ ?

- Only 23 people.
- https://keisan.casio.com/exec/system/1223738282
-Probability with same birthdays

n

$$
P_{\text {coincidence }}=1-\frac{365 \cdot 364 \cdot 363 \cdot \ldots \cdot(365-n+1)}{365^{n}}
$$

Z. E. Schnabel, The estimation of the total fish population of a lake, in American Mathematical Monthly, vol. 45, no. 6, pp. 348-352, June-July 1938.

James K. Polk (Nov. 2, 1795)



One coincidence for the US presidents happened for the $\mathbf{2 8}^{\text {th }}$ president (W. G. Harding)

## More on birthday statistics

$\square$ How many people we need to put in the same group so the probability of finding, at least 2 people whose birthdays are $k$ days apart is greater than 50\%?

- Much less: for $k=1$ day, only 14 people.



## More on birthday statistics

$\square$ The previous idea can be used for analyzing bitflips observed in a memory.


## Birthday statistics

1. Which is the probability of finding, at least, 2 people whose birthdays are $k$ days apart in a group of $n$ people?
2. Which is the probability of finding, at least, 2 bitflips that are $k$ bitcells apart in a memory with $n$ bitflips?
$\square$ Birthday statistics can be used for analyzing probability of occurrence of close bitflips (MBUs and MCUs) falsely attributed to the same particle.
3. In a group of n people, it 's not that unlikely to find 2 birthdays being placed, at least, k days apart
4. In a set of n bitflips, it's not that unlikely to find 2 affected addresses being placed, at least, $k$ bitcells apart
$\square$ In other words, it's not that unlikely to find multiple events by accumulation (false MCUs).

## Estimation of the number of 2-bit "false MCUs"

## Manhattan Distance (MD)



Infinity Norm
Distance (IND)

| 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 3 | 2 | 2 | 2 | 2 | 2 | 3 |
| 3 | 2 | 1 | 1 | 1 | 2 | 3 |
| 3 | 2 | 1 |  | 1 | 2 | 3 |
| 3 | 2 | 1 | 1 | 1 | 2 | 3 |
| 3 | 2 | 2 | 2 | 2 | 2 | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 |

Area of influence $=48$ cells

Threshold Distance
(TD)

| $n=$ Size of the memory (bits) | IND $=$ Infinity Norm Distance |
| :--- | :---: |
| $p=$ Number of bitflips | TD $=$ Threshold Distance |
| $M D=$ Manhattan Distance |  |

F. J. Franco et al., Influence of Randomness during the Interpretation of Results from Single-Event Experiments on SRAMs, in IEEE Transactions on Device and Materials Reliability (TDMR), vol. 19, no. 1, pp. 104-111, March 2019.
F. J. Franco et al., Inherent Uncertainty in the Determination of Multiple Event Cross Sections in Radiation Tests, in IEEE Transactions on Nuclear Science (TNS), vol. 67, no. 7, pp. 1547-1554, July 2020.

3-bit MCUs can also be estimated, but equations are way more complex and out of the scope of this discussion.

## Correction of experimental data

$\square$ Example. For a 16-Mbit memory and 2400 bitflips, NF_MCUs_2bit = 4. Does this mean that any time we find 2400 bitflips in a 16-Mbit memory, 4 false 2-bit MCUs will occur for sure?

- NO. NF_MCUs_2bit is a false 2-bit MCU rate.
$\square$ Such false MBUs/MCUs are "rare events" and their stochastic occurrence can be modeled with the Poisson distribution.Let $\lambda$ be such an event rate:


$k=$ number of observed false events


## Correction of experimental data

$\square$ Alternative 1. Let an experiment be:

- Memory size $=1 \mathrm{Mb}$ ( $2^{20}$ bits)
- Criteria: MD (threshold value = 3)
- $p=592$ bitflips
- $N_{F_{-} M C U S_{-2 b i t}}=4$ false 2-bit MCUs
- $N_{\text {observed_MCUs_2bit }}=5$ observed 2-bit MCUs
$\square$ What are those 5 MCUs? false, true...?
$\square$ Let's find a value of $k\left(k_{0}\right)$ such that $\operatorname{CDF}\left(k_{0}\right)>99 \%$

- $\mathrm{k}_{0}=9$. There is $99 \%$ probability that between 0 and 9 false 2-bit MCUs occur in that experiment.
- 5 false MCUs are perfectly within that range, hence we can consider them as false.


## Correction of experimental data

$\square$ Alternative 2. Let another experiment be:

- Memory size $=16 \mathrm{Mb}$ ( $2^{24}$ bits)
- Criteria: MD (threshold value = 3)
- $p=2439$ bitflips
- $N_{F_{-} M C U U_{-} 2 b i t}=4$ false 2-bit MCUs
- $N_{\text {observed_MCUs_2bit }}=11$ observed 2-bit MCUsThe following methodology can be followed:
- Confidence margins are calculated around $N_{\text {events }}=11$.
- A good approach is: $=\frac{1}{2} \chi^{2}\left(\frac{\alpha}{2}, 2 N_{\text {events }}\right)<N_{\text {events }}<\frac{1}{2} \chi^{2}\left(1-\frac{\alpha}{2}, 2\left(N_{\text {events }}+1\right)\right)$
- With $95 \%$ confidence, $N_{\text {events }}=\left[N_{\text {events_Low, }}, N_{\text {events_HIGH }}\right]=[5.49,19.68]$
- The following correction is made:
- $\left[N_{\text {events_LOW }}-N_{\text {observed_MCUs_2bit }}, N_{\text {events_HIGH }}-N_{\text {observed_MCUs_2bit }}\right]$
- In this case, $[5.49-4,19.68-4]=[1.49,15.68]$
- We can say that, in that experiment, there is $95 \%$ probability that, between 1.49 and 15.68 actual 2-bit MCUs occurred.


# 4-Analysis of "false" MBUs by accumulation 

Error Correcting Codes (ECC)
Accumulation of events and ECC reliability

## Accumulation of "false MBUs"



## Accumulation of "false MBUs" - ECC



## Error Correcting Codes (ECC)

$\checkmark$ Mechanism to add redundancy to the memory contents.
$\checkmark$ An M-bit word contains $\mathrm{N}=\mathrm{M}+\mathrm{K}$ bits.
$\checkmark$ The " f " module generates the K redundancy bits.
$\checkmark$ The "Comparator" reports if there has been an error in the word (ERR signal)
$\checkmark$ The "Corrector" corrects the DOUT, but it does not correct the fault in the memory module.
$\checkmark$ ECCs are sensitive to accumulated errors.

## Types of ECC

Single Error Correction - Double Error Detection (SEC-DED)


Double Error Correction - Triple Error Detection (DEC-TED)


Triple Error Correction (TEC)


Double Adjacent Error Correction (DAEC)


Single Nibble Correction - Double Nibble Detection (SNC-DND)


## Estimation of false MBU rates

$\square$ This is relevant for studying the efficiency of Error Correction Codes (ECC).
$\square$ For instance, in a block of $n$ bits, a SEC-DED code will be effective only if 2 bitflips do not occur in the same word:


## Estimation of MBU rates

$\square$ The most accurate estimation ever made in the literature:

- Estimated number of false 2-bit MBUs:

$$
\boldsymbol{N}_{\boldsymbol{F} \boldsymbol{M}}(\mathbf{2}) \approx \frac{W-1}{W}\left(N_{H}(2)+6 \cdot \frac{W-2}{W^{2}} \cdot N_{H}(4)\right)
$$


$\square$ Where $N_{H}(k)$ is the estimated number of addresses being hit $k$ times:

$$
N_{H}(k)=\binom{m}{k} \cdot\left(L_{A}\right)^{1-k} \cdot\left(1-\frac{1}{L_{A}}\right)^{m-k} \quad \begin{aligned}
& W=\text { Data width per address (bits) } \\
& m=\text { Number of bitflips } \\
& L_{A}=\text { Total number of data addresses }
\end{aligned}
$$

## Estimation of false MBU rates

$\square$ Similarly:

- Estimated number of false 3-bit MBUs:

$$
\boldsymbol{N}_{\boldsymbol{F} \boldsymbol{M}}(\mathbf{3}) \approx \frac{(W-1) \cdot(W-2)}{W^{2}} \cdot\left(N_{H}(3)+10 \cdot \frac{W-3}{W^{2}} \cdot N_{H}(5)\right)
$$

$$
\begin{aligned}
& W=\text { Data width per address (bits) } \\
& N_{H}(k)=\text { same as previous slide }
\end{aligned}
$$

- Estimated number of false 4-bit MBUs:

$$
\boldsymbol{N}_{\boldsymbol{F} \boldsymbol{M}}(\mathbf{4}) \approx \frac{(W-1) \cdot(W-2) \cdot(W-3)}{W^{3}} \cdot\left(N_{H}(4)+5 \cdot \frac{3 W-8}{W^{2}} \cdot N_{H}(6)\right)
$$

- The same can be done for 5-bit, ... n-bit MBUs.


## Estimation of MBU rates - ECC reliability

$\square$ Single Error Correction - Double Error Detection (SEC-DED) is sensitive against any MBU of any multiplicity.
$\square$ Therefore, the probability of failure of SEC-DED is the cumulated probability of seeing any MBU of any multiplicity.

The expected number (or rate) of MBUs with any multiplicity $i$ is: $\quad N_{F M}^{2+}=\sum_{i=2}^{W} N_{F M}(i)$

$$
\begin{aligned}
P_{\text {failure_SEC_DED }} & =\sum_{k=1}^{\infty} \frac{e^{-\lambda \cdot \lambda^{k}}}{k!} \\
& =1-e^{-\lambda} \\
& =\mathbf{1}-\boldsymbol{e}^{-N_{F M}^{2+}}
\end{aligned}
$$

- Example (8Kx8-bit memory). As the number of accumulated bitflips increase, the probability of breaking SEC-DED increases too.

If an event rate is $\lambda$, the probability of seeing exactly $k$ events is: $\frac{e^{-\lambda \cdot \lambda^{k}}}{k!}$ (Poisson distribution):


## Estimation of MBU rates - ECC reliability

$\square$ Similarly, DEC-TED is sensitive against >2-bit MBUs

- It tolerates MBUs with multiplicity 2.

$$
\text { For DEC-TED, } \lambda=N_{F M}^{3+}=\sum_{i=3}^{W} N_{F M}(i)
$$



- And similarly: $P_{\text {failure_DEC_TED }}=1-e^{-\lambda}=\mathbf{1}-\boldsymbol{e}^{-N_{F M}^{3+}}$
$\square$ Similar calculations can be proposed for events that "break" other ECC types: DAEC, SNC-DND, etc.


## Estimation of MBU rates - ECC reliability

$\square$ The number of accumulated bitflips needed keep different ECC techniques under certain reliability can also be calculated.

Reliability of Different ECC Techniques against MBUs Provoked by SBU Accumulation, for a 256-MB Memory

|  | Total number of bitflips ( $m$ ) |  |  |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | SEC-DED |  |  |  | DEC-TED |  |  |  | SNC-DND |  | TEC |
| ECC reliab. | $(22,16)$ | $(39,32)$ | $(72,64)$ | $(137,128)$ | $(27,16)$ | $(45,32)$ | $(79,64)$ | $(145,128)$ | $(144,128)$ | $(152,128)$ | $(64,45)$ |
| 99\% | 1682 | 1178 | 828 | 584 | 106813 | 66250 | 41328 | 25886 | 590 | 598 | 147488 |
| 99.9\% | 531 | 372 | 262 | 185 | 49505 | 30705 | 19154 | 11997 | 187 | 189 | 99477 |
| 99.99\% | 169 | 118 | 83 | 59 | 22975 | 14250 | 8890 | 5568 | 60 | 61 | 66480 |
| 99.999\% | 54 | 38 | 27 | 19 | 10664 | 6615 | 4127 | 2585 | 20 | 20 | 43581 |
| 99.9999\% | 18 | 13 | 9 | 7 | 4951 | 3071 | 1916 | 1201 | 7 | 7 | 27638 |
| 99.99999\% | 6 | 5 | 4 | 3 | 2299 | 1426 | 890 | 558 | 3 | 3 | 16772 |
| 99.999999\% | 3 | 2 | 2 | 2 | 1068 | 663 | 414 | 260 | 2 | 2 | 9786 |

J.A. Clemente, M. Rezaei and F. J. Franco, Reliability of Error Correction Codes Against Multiple Events by Accumulation, in IEEE Transactions on Nuclear Science (TNS), vol. 69, no. 2, pp. 169-180, February 2022.

## 5-Conclusions

Modern memories implement techniques such as bit interleaving and Error Correcting Codes (ECC) to increase reliability.
$\square$ Devices are increasingly sensitive to multiple events, therefore a correct SBU/MCU extraction and classification is very important.
$\square$ In radiation-ground experiments, analyzing data correctly involves:

- Classifying events by multiplicity.
" Estimating the "false MCU" rates.
$\square$ In the real world, SBUs coincidentally affecting bitcells in the same word can break the ECC.
- Chances are not that low!! (remember "birthday statistics").
$\square$ Equations have been given to estimate false "multiple event rates".
- False MCUs: Provide accurate results in tests.
- False MBUs: Estimate the reliability of ECC techniques.


## Thanks for your attention!

