March 2013
UILU-ENG-13-2202
CRHC-13-02

# RELIABILITY MODELS FOR DOUBLE CHIPKILL DETECT/CORRECT MEMORY SYSTEMS

**Xun Jian, Sean Blanchard, Nathan Debardeleben, Vilas Sridharan, and Rakesh Kumar**

*Coordinated Science Laboratory*
*1308 West Main Street, Urbana, IL 61801*
*University of Illinois at Urbana-Champaign*

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB NO. 0704-0188*



| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | March 2013 | |

*4. TITLE AND SUBTITLE*
Reliability Models for Double Chipkill Detect/Correct Memory Systems

5. FUNDING NUMBERS
DE-FC02-06ER25750 (Department of Energy)

6. AUTHOR(S)
Xun Jian, Sean Blanchard, Nathan Debardeleben, Vilas Sridharan, Rakesh Kumar

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Coordinated Science Laboratory, 1308 W. Main St., Urbana, IL, 61801-2307

8. PERFORMING ORGANIZATION REPORT NUMBER
UILU-ENG-13-2202 (CRHC-13-02)

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
This work was performed in part at the Ultrascale Systems Research Center (USRC) at Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545, supported by the U.S. Department of Energy, 1000 Independence Ave. SW, Washington, D.C., 20585

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
LA-UR-13-21186 (Los Alamos National Laboratory)

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Chipkill correct is an advanced type of error correction used in memory subsystems. Existing analytical approaches for modeling the reliability of memory subsystems with chipkill correct are limited to those with chipkill correct solutions that can only guarantee correction of errors in a single DRAM device. However, chipkill correct solutions capable of guaranteeing the detection and even correction of errors in up to two DRAM devices have become common in existing HPC systems. Analytical reliability models are needed for such memory subsystems. This paper proposes analytical models for the reliability of double chipkill detect and/or correct. Validation against Monte Carlo simulations shows that the outputs of our analytical models are within 3.9% of Monte Carlo simulations, on average.

14. SUBJECT TERMS
ECC; Memory; Chipkill correct; Modeling; Reliability

15. NUMBER OF PAGES
6

16. PRICE CODE

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | UL |

# Reliability Models for Double Chipkill Detect/Correct Memory Systems

Xun Jian
UIUC
stevensonjian@gmail.com

Sean Blanchard
Los Alamos National Laboratory
seanb@lanl.gov

Nathan Debardeleben
Los Alamos National Laboratory
ndebard@lanl.gov

Vilas Sridharan
RAS Architecture
Advanced Micro Devices, Inc.
vilas.sridharan@amd.com

Rakesh Kumar
UIUC
rakeshk@illinois.edu

***Abstract*—Chipkill correct is an advanced type of error correction used in memory subsystems. Existing analytical approaches for modeling the reliability of memory subsystems with chipkill correct are limited to those with chipkill correct solutions that can only guarantee correction of errors in a single DRAM device. However, chipkill correct solutions capable of guaranteeing the detection and even correction of errors in up to two DRAM devices have become common in existing HPC systems. Analytical reliability models are needed for such memory subsystems. This paper proposes analytical models for the reliability of double chipkill detect and/or correct. Validation against Monte Carlo simulations shows that the output of our analytical models are within 3.9% of Monte Carlo simulations, on average.**

## I. Introduction

Chipkill correct is an advanced type of error correction in memory that provides high reliability [12]. A recent large scale field study of DRAM errors showed that chipkill correct reduces the detectable uncorrectable error (DUE) rate of memory by 42X compared to SECDED [1]. As a result, chipkill correct has become popular among HPC systems. As the memory capacity of computing systems continues to increase, the demand for strong error correction such as chipkill correct will continue to increase as well. As such, it becomes useful for memory system designers to be able to predict the reliability of their systems for a particular strength of chipkill correct.

Existing analytical reliability models for systems with chipkill correct are limited to chipkill correct solutions that can only correct a single error per word [2], [3]. However, chipkill correct solutions capable of detecting or even correcting up to two errors in the same word have become standard in existing HPC systems. For example, the memory subsystem of the Jaguar supercomputer can detect up to two errors in the same word [1]. The memory subsystem of the IBM Blue Gene/L supercomputer is protected by Double Chip Sparing, a form of double chipkill correct which can correct up to two errors in the same word as long as the second error does not appear before the first error is detected [14], [6]. As far as we are aware, analytical reliability models do not exist for memory subsystems with these levels of protection.

In this paper, we propose reliability models for chipkill correct solutions that can detect and/or correct up to two errors per word. We develop models for both simultaneous double error detection and/or correction as well as Double Chip Sparing. Unlike previous models for chipkill correct, our models can also be used to study the effect of memory scrubbing on memory subsystems with chipkill correct by distinguishing between permanent faults and transient faults. We validate our models with Monte Carlo simulations using the fault rates from a field study of DRAM faults[1]. The memory error probability calculated using the analytical models are, on average, within 3.9% of those obtained from the Monte Carlo simulations.

## II. Background

### A. Memory Organization

A conventional memory subsystem consists of one or more *memory channels*; a memory channel usually serves memory requests independent of other memory channels. A memory channel, in turn, consists of one or more *ranks*; a rank is a group of *devices* within a channel that serve a memory request together. Devices that belong to different ranks but share the same bits in the I/O bus of the memory channel form a *lane*. Meanwhile, each rank consists of multiple *banks*. A bank is a logical entity that consists of a group of *subbanks*, one from each device in the rank. A subbank, in turn, consists of rows and columns of *symbols*, which are groups of adjacent bits.

### B. DRAM Fault Modes

Sridharan et al. [1] report that multiple symbols in the same lane, device, subbank, column, and row can become faulty at the same time due to the fact that these structures are controlled by their respective device drivers which can malfunction. The authors also show that these device-level

This work was largely performed at the Ultrascale Systems Research Center (URSC) at Los Alamos National Laboratory, supported by the U.S. Department of Energy DE-FC02-06ER25750. The publication has been assigned the LANL identifier LA-UR-13-21186.

faults can be either transient or permanent. Transient faults can be repaired by periodic memory scrubbing. Memory scrubbing repairs transient faults in the memory channel by reading every word and then writing back the corrected word. However, scrubbing cannot fix permanent faults since the memory location is permanently damaged in a permanent fault. Our model in Section IV considers these types of faults. Finally, similar to [1], we define *fault* as a physical malfunction in a DRAM device and *error* as the manifestation of the fault when a word affected by the fault is accessed.

### C. Commercial Chipkill Correct Solutions

Conventional error-correcting codes (ECCs) work by adding *check bits* to groups of data bits in memory. These check bits provide redundant information that allow detection and/or correction of some set of data bits. Each group of data and check bits is referred to as a *codeword*. A special class of ECCs called linear block codes divide a codeword into multiple *symbols*. By storing every symbol of a codeword in a device in a different lane, commercial chipkill correct solutions ensure that a fault that develops in a device only affects at most a single symbol per codeword. Double chipkill detect and/or correct codes, which are the focus of this paper, ensure that any two symbol errors can be detected and/or corrected, respectively. As a result, an undetected or uncorrected error will occur only if three DRAM devices in different lanes each develop a fault that overlaps in a single codeword. For the rest of this paper, a *memory error* refers to such three-device errors, which cannot be corrected by double chipkill correct and Double Chip Sparing, and cannot be detected by double chipkill detect.

## III. Related Work

Although a large body of work in literature model memory reliability (e.g., [7], [8], [9], [10], [11]), many of these studies focus on SECDED, which is a weaker form of ECC that targets random transient faults. SECDED does not provide protection against many device-level faults, which are the targeted fault modes of chipkill correct. A small number of studies such as [2] and [3] also consider device-level faults; however the models in these studies are limited to chipkill correct solutions capable of correcting only a single bad symbol per codeword. In addition, these models do not differentiate between transient and permanent faults, both of which are shown to occur frequently in the field [1], [13].

## IV. Proposed Models

In this section, we propose models to calculate the probability of developing memory error in a memory subsystem with double chipkill detect and/or correct and Double Chip Sparing. In particular, our models calculate the probability of developing fault combinations that result in application errors when memory locations affected by such fault combinations is accessed. We do not model the actual probability of application error due to faults in memory, since this depends heavily on application memory access patterns.

### A. Assumptions

To keep our models tractable, we make several assumptions about the behavior of DRAM faults. First, we assume faults in one lane are independent from faults in other lanes. Second, we assume an exponential fault distribution with a constant fault rate, which is supported by recent field studies [1]. Third, we assume an ideal memory scrubber such that no transient fault persists across memory scrubs.

In our reliability model for double chipkill detect and/or correct (but not Double Chip Sparing), we also neglect memory errors where two or more of the three bad symbols are due to transient faults. The rationale behind this simplification is that since transient faults are repaired after each memory scrub, the probability that two or more of the three faults affecting a codeword are transient is much smaller than the probability that two or more of the three faults are permanent. As such, this probability can be neglected with small impact on the overall probability of encountering memory errors. Note, however, that our Monte Carlo experiments in Section V used to validate the analytical model do consider these memory errors.

Finally, we assume that at least one memory error always occurs when the same memory structures (e.g., device, subbank etc) in 3 or more lanes of the same channel are faulty. For example, we assume that when the same subbank (e.g., the first subbank) in 3 devices in the same rank are faulty, at least one memory error occurs.

This assumption is based on the observation that when the same memory structure in 3 different devices of the same rank are faulty, having no memory error requires that *every* symbol location do not contain matching bad symbols across all 3 devices; the greater the total number of symbols in the memory structure, the smaller the probability of the event above. Since memory structures tend to contain a large number of symbols, the probability of having at least one memory error when the same memory structures in 3 different devices are faulty is large. Assuming that the affected symbols in a memory structure are randomly distributed across the entire memory structure, the probability of having at least 1 memory error when the same memory structure in 3 different devices in the rank are faulty can be approximated by $1-(1-x^3)^y$. $x$ represents the fraction of the total symbols in a memory

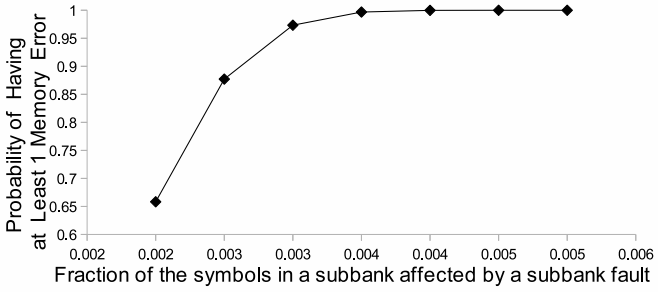


Fig. 1. The probability of having at least one memory error when the same subbank in 3 different devices in the same rank are affected by subbank fault.

structure affected by the fault; as such, $x^3$ represents the probability that the same symbol in all 3 devices are faulty. Meanwhile, $y$ is the total number of symbols in a subbank; as such, $(1-x^3)^y$ represents the probability that no symbol location out of all $y$ symbol locations are faulty across all 3 devices. Let's consider, for example, a subbank with a total of $y = 2^{15} \cdot 2^{11} = 2^{26}$ symbols. Figure 1 shows the calculated probability of having at least one memory error versus the fraction of the symbols in a subbank affected by a single subbank fault, when the same subbank in 3 different devices in the same rank are faulty. The figure shows that the probability is nearly 1 when the fraction of symbols in a subbank that is affected by a subbank fault is greater than 0.3%.

### *B. Reliability Model for Double Chipkill Detect and/or Correct*

Our analytical reliability model differentiates between transient and permanent faults. Since transient faults are periodically removed but permanent faults are not, the model must consider the time of occurrence of the faults. We used an iterative approach to enumerate all the possible combinations of the time of occurrence of the faults that do not lead to memory errors. Our analytical model also considers different types of device-level faults (e.g., lane fault, device fault, subbank fault etc.). As such, it must also consider the location of each fault to determine whether it overlaps with other faults to affect a common set of codewords. As the types of faults to be considered increase, the number of their possible spatial combinations that do not result in memory errors grows rapidly. We rely on a recursive method to enumerate these spatial combinations of different types of faults.

Our method relies on recursively describing the different types of faults and the set of codewords that they affect. We define a *fault region*, or simply *region*, as the set of codewords affected by a particular type of fault. Conversely, we refer to the fault for which a fault region is defined as the *spanning fault* of the region. We say that a fault is *inside* a region if it affects only a subset of codewords in the region. The regions corresponding to the faults that affect a proper subset of codewords of a region are referred to as the *sub-regions* of the region. The largest sub-region of a region is called the *immediate sub-region* of the region. Conversely, a region is also called the *super-region* of its sub-regions. From here on, we denote a region by $x$, a sub-region of $x$ by $x^-$, a super-region of $x$ by $x^+$, and an immediate sub-region of $x$ by $x-1$.

TABLE I
REGIONS, SUB-REGIONS, IMMEDIATE SUB-REGIONS, SUPER-REGIONS, AND FAULTS INSIDE A REGION FOR AN EXAMPLE SCENARIO.

| $x$ | spanning fault of $x$ | fault inside $x$ | $x^-$ | $x-1$ | $x^+$ |
|---|---|---|---|---|---|
| channel | lane fault | lane, device, subbank fault | rank, bank | rank | NA |
| rank | device fault | device, subbank fault | bank | bank | channel |
| bank | subbank fault | subbank fault | NA | NA | channel, rank |

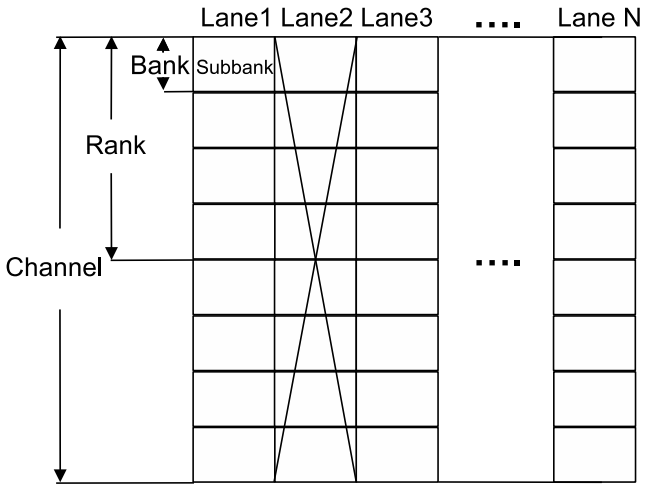


Fig. 2. Regions and sub-regions. The cross represents a lane fault.

Let's consider a simple example scenario where the only types of faults to be considered are the lane fault, device fault, and subbank fault. Table I lists the regions corresponding to these faults as well as the other terms defined above.

Figure 2 graphically illustrates regions and sub-regions for the scenario in Table I. Since each region is a set of codewords, the figure shows that a region spans across $N$ lanes, where $N$ is the number of symbols in each codeword. The figure shows an example where a channel contains 2 immediate sub-regions (2 ranks) and each rank contains 4 immediate sub-regions (4 banks). The figure also shows a lane fault (represented by a cross). It serves to illustrate that each fault is confined to a single lane.

We say that $x$ is *reliable* if it contains zero codewords with memory error. Note from previously that a memory error occurs when a codeword is affected by 3 or more faults in 3 or more DRAM devices. Graphically, $x$ is reliable as long as there does not exist a horizontal slice of $x$ spanning across all $N$ lanes that intersects with three or more faults inside $x$. We observe that $x$ is reliable if for all scrub intervals,

1) $x$ contains no spanning faults of its own and every $x-1$ is reliable, or
2) $x$ contains a single spanning fault of its own and there are no two faults that overlap with each other inside any $x^-$, or
3) $x$ contains exactly two spanning faults of its own and no $x^-$ contains any faults.

To analytically describe the probability that a region is reliable, we need to define some additional variables. We let $p^+$ represent the sum of the number of permanent spanning faults (*PSF*) of every $x^+$ (e.g., in Figure 2, $p^+ = 1$ when $x$ represents either a rank or a bank). Let $k$ represent the scrub interval when the last spanning fault (*SF*) of all $x^+$s occurs. Finally, let *O* be a true/false value representing whether the last SF of all $x^+$s is permanent ($O = T$) or transient ($O = F$). In the special cases where there are no $x^+$s (e.g., when $x$ represents the topmost region) or where every $x^+$ has neither

PSF nor TSF, O takes on the value of F. We let $e(x,t|p^+,k,O)$ denote the event that $x$ is reliable over timespan $t$ given that the sum of the number of PSFs of every $x^+$ is $p^+$, the last SF of all $x^+$s occurs in interval $k$, and the Boolean condition of whether the last SF of all $x^+$s is permanent is $O$. We chose to let the first scrub interval be interval 1, not interval 0; we let $k$ equal 0 for the special case where there are no $x^+$s or when no $x^+$ contains any SF at all.

Under these definitions, the probability that an entire memory channel with double chipkill detect and/or correct remains reliable over timespan $t$ is:

$$R(t) = P(e(x,t|0,0,F)) \tag{1}$$

where $x$ above represents a channel and $P(e(x,t|0,k,F))$ is the probability of $e(x,t|0,k,F)$.

To describe $e(x,t|p^+,k,O)$, we define some additional variables. We let $p$ represent the total number of PSFs of $x$. We let $j$ represent the scrub interval when the last PSF of $x$ occurs; similarly, we let $l$ and $L$ represent the interval when the last transient spanning fault (TSF) of and last transient fault inside $x$ occur, respectively. $e(x,t|0,k,F)$ is true if

1) $p = 2$, $L < j$ (e.g., there are no transient faults in $x$ during and after the scrub interval when the last PSF of $x$ occurs), and there are no other permanent faults inside $x$ for all scrub intervals, or
2) $p = 1$, $max(k,l) < j$ (e.g., there is no TSF of $x$ and no TSF of any $x^+$ during or after the scrub interval when the last PSF of $x$ occurs), and correspondingly, $e(x-1,t|1,j,T)$ for every $x-1$, or
3) $p = 1$, $max(k,l) >= j$ (e.g., the last TSF of $x$ or the last TSF of all $x^+$s occurs during or after the scrub interval when the last PSF of $x$ occurs), and correspondingly, $e(x-1,t|1,max(k,l),F)$ for every $x-1$, or
4) $p = 0$, $e(x-1,t|0,max(k,l),F)$ for every $x-1$.

$P(e(x,t|0,k,F))$ is, therefore, equal to the sum of $P_1(e(x,t|0,k,F))$, $P_2(e(x,t|0,k,F))$, $P_3(e(x,t|0,k,F))$, and $P_4(e(x,t|0,k,F))$, which correspond to the probabilities of the four exclusive sets of conditions listed above.

$P_1(e(x,t|0,k,t))$ equals to the sum of the individual probabilities of all fault combinations in both time and space that satisfy condition 1) of $e(x,t|0,k,t)$. By using the formula for the exponential fault distribution $R(t) = e^{-\lambda \cdot t}$, where $\lambda$ is the fault rate, and by letting $s$ be the duration of a scrub interval, $\omega_x$ be the incidence rate of the PSF of $x$ in a single lane, $\Omega_x$ be the combined incidence rate of all permanent faults inside $x$ in a single lane, $\phi_x$ be the incidence rate of the TSF of $x$ in a single lane, and $\Phi_x$ be the combined incidence rate of all transient faults inside $x$ in a single lane,

$$\begin{aligned} &P_1(e(x,t|0,k,F)) \\ &= e^{-\Omega_x(N-2)t} \sum_{i=1}^{t/s} \{[N \cdot e^{-\omega_x \cdot s \cdot (i-1)}(1-e^{-\omega_x \cdot s}) \cdot \\ &\quad \sum_{j=max(i+1,k+1)}^{t/s} (N-1)e^{-\omega_x \cdot s \cdot (j-1)}(1-e^{-\omega_x \cdot s}) \cdot \\ &\quad e^{-\Phi_x(N-2)(t-s\cdot(j-1))}]\} + \underline{\sum_{i=1}^{t/s} \binom{N}{2} (e^{-\omega_x \cdot s \cdot (i-1)})^2 \cdot} \\ &\quad \underline{(1-e^{-\omega_x \cdot s})^2 \cdot e^{-\Phi_x(N-2)(t-s\cdot(i-1))}} \end{aligned} \tag{2}$$

The $e^{-(N-2)\cdot\Omega_x \cdot t}$ term represents the fact that except for the two lanes with the two PSFs of $x$, none of the remaining $N-2$ lanes of $x$ can contain any permanent fault. The nested summations account for all combinations of the two PSFs of $x$ where the two PSFs appear in different scrub intervals. The underlined summation accounts for the special case where both PSFs of $x$ occur in the same scrub interval (interval $i$) so that only a single summation is needed. Let's consider the terms under the nested summation. $N$ stands for the fact that the first PSF of $x$ has $N$ lanes to choose from. The subsequent $e^{-\omega_x \cdot s \cdot (i-1)}(1-e^{-\omega_x \cdot s})$ term specifies that the first PSF of $x$ occurs during the $i^{th}$ scrub interval. Similarly, $N-1$ means that the second PSF of $x$ has $N-1$ lanes to choose from and the subsequent $e^{-\omega_x \cdot s(j-1)}(1-e^{-\omega_x \cdot s})$ term specifies that the second PSF of $x$ occurs during the $j^{th}$ interval. The $e^{-\Phi_x(N-2)(t-s\cdot(j-1))}$ term that follows specifies that there are no transient faults in $x$ during and after interval $j$.

The remaining probabilities, $P_2(e(x,t|0,k,F))$, $P_3(e(x,t|0,k,F))$, and $P_4(e(x,t|0,k,F))$ can be similarly translated from their textual descriptions into mathematical expressions; they are listed in the Appendix.

Meanwhile, $e(x,t|1,k,T)$ is true if

1) $p = 1$, $L < max(j,k)$ (e.g., the last transient fault inside $x$ occurs before the scrub interval when both the PSF of $x$ and the PSF of a $x^+$ have both occured), and there are no other permanent faults inside $x$ in all scrub intervals, or
2) $p = 0$ and $e(x-1,t|1,max(k,l),k > l)$ for every $x-1$.

$P(e(x,t|1,k,T))$, is, therefore, equal to the sum of $P_1(e(x,t|1,k,T))$ and $P_2(e(x,t|1,k,T))$, which are the probabilities of the two exclusive sets of conditions listed above.

Finally, $e(x-1,t|1,k,F)$ is true if

1) $p = 1$, $max(k,L) < j$ (e.g., there is no transient fault in $x$ and no TSF of any $x^+$ during or after the scrub interval when the PSF of $x$ occurs), and there are no other permanent faults inside $x$, or
2) $p = 0$ and $e(x-1,t|1,max(k,l),F)$ for every $x-1$.

$P(e(x,t|1,k,F)$ is, therefore, equal to the sum of $P_1(e(x,t|1,k,F)$ and $P_2(e(x,t|1,k,F)$, the probabilities of the two exclusive sets of conditions listed above.

$P_1(e(x,t|1,k,T))$, $P_2(e(x,t|1,k,T))$, $P_1(e(x,t|1,k,F)$, and $P_2(e(x,t|1,k,F)$ are provided in the Appendix.

### C. Double Chip Sparing

Double Chip Sparing can correct up to two bad symbols in a codeword as long as the second bad symbol does not occur before the first bad symbol has been detected. To model Double Chip Sparing without having to take into account application access patterns, we say that Double Chip Sparing can correct up to two bad symbols per codeword as long as the two bad symbols do not occur in the same scrub interval.

To estimate the reliability of Double Chip Sparing using the reliability model in Section IV, one simply needs to subtract from the output of the model the probability of the event of exactly two faults affecting the same codeword in a single interval. We observe that the probability of having exactly two faults affecting the same codeword in a single scrub interval can be calculated as the probability of having two or fewer faults affecting the same codeword in a single scrub interval minus the probability of having one or fewer fault affecting the same codeword in a single scrub interval.

The probability of having one or fewer faults affecting the same codeword in a single scrub interval can be calculated by

$$r_1(x) = e^{-N(\phi_x+\omega_x)\cdot s} \cdot r_1(x-1)^{n_x} + N \cdot e^{-(N-1)(\Phi_x+\Omega_x)s}(1-e^{-(\phi_x+\omega_x)s}) \quad (3)$$

where $n_x$ represents the total number of $x-1$s in $x$.

Meanwhile, the probability of having two or fewer faults affecting the same codeword in a single scrub interval can be calculated by

$$r_2(x) = e^{-N(\phi_x+\omega_x)s} \cdot r_2(x-1)^{n_x} + N \cdot e^{-(N-1)(\phi_x+\omega_x)s}(1-e^{-(\phi_x+\omega_x)s}) r_1(x-1)^{n_x} + \binom{N}{2} e^{-(N-2)(\Phi_x+\Omega_x)s}(1-e^{-(\phi_x+\Phi_x)s})^2 \quad (4)$$

The probability of having scrub intervals with exactly two faults affecting the same codeword in at least one scrub interval over timespan $t$ is, therefore, $r_2(x)^{(t/s)} - r_1(x)^{(t/s)}$. The overall reliability of Double Chip Sparing is, therefore,

$$R_{sparing}(t) = R(t) - (r_2(x)^{(t/s)} - r_1(x)^{(t/s)}) \quad (5)$$

## V. Validation

To evaluate our analytical model, we used an example memory channel consisting of 2 ranks, where each rank consists of 8 banks, and each bank contains 512 columns of 16B words (for a total of 8KB per memory row). Assuming that there are 36 symbols per codeword to provide double chipkill detect and/or correct in existing commercial chipkill correct solutions [4], we let each channel consist of 36 lanes.

The evaluation model considers the multi-rank, multi-bank, bank, column, and row faults from [1] and uses also the corresponding fault rates from [1]. Although [1] reports different types of multi-bank faults (e.g., some multi-bank faults affect 2 subbanks, others affect 4 subbanks etc), to keep the corresponding Monte Carlo simulations (described in the next paragraph) used to validate the models tractable, we model all types of multi-bank faults as faults that affect all 8 subbanks in a device. Note that the proposed analytical models are also capable of modeling some of these more detailed types of faults (e.g., dual-bank faults or quad-bank faults, which affect 2 or 4 adjacent subbanks in a device, respectively). Similarly, we chose to consider all types of multi-rank faults as lane faults, whereby every device in a lane is affected. Correspondingly, the fault regions in the evaluation model are the channel (region 4), rank (region 3), bank (region 2), word column and word row (both modeled as region 1). A scrub interval of once every 24 hours was used. To ensure that our analytical model works well for different fault rates, we scaled the fault rates reported in [1] by factors of 2X and 4X.

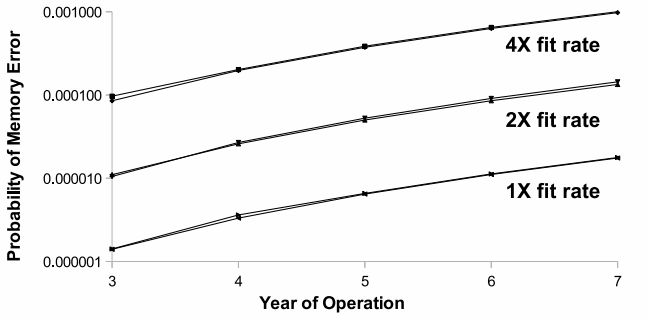


Fig. 3. Probability of having one or more memory errors in a channel with double chipkill detect and/or correct as calculated using the analytical model and as obtained by Monte Carlo simulations.

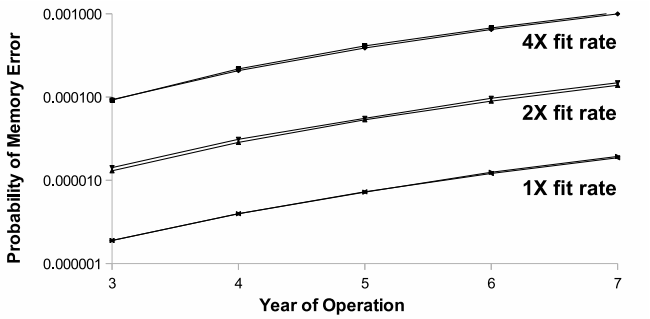


Fig. 4. Probability of having one or more DUEs in a memory channel by the $n^{th}$ year as calculated using the analytical model for Double Chip Sparing and as obtained by Monte Carlo simulations.

We used over 250 million Monte Carlo experiments to validate the results obtained using the evaluation model. The Monte Carlo experiments take the same inputs as the evaluation model. Each Monte Carlo experiment simulates a memory channel; it ends either when the first memory error occurs in the channel or after a total of 7 simulated years.

Figure 3 shows that the results of the evaluation model closely match those of the Monte Carlo simulations for different device fault rates. The curves for the Monte Carlo simulations almost lie exactly on top of those of the analytical model. The average difference between the calculated and simulated values for each data point is only 3.6% with a standard deviation of 3.9%.

Figure 4 shows that the results of the evaluation models for Double Chip Sparing also closely match those of the Monte Carlo simulations for Double Chip Sparing. The average difference between the calculated and simulated values for each data point is 3.9% with a standard deviation of 3.2%. On average across all the calculated data points, the probability of having an uncorrectable error is 10.3% higher for Double Chip Sparing than for chipkill correct solutions with simultaneous double symbol correction; the average is 9.6% higher for data points obtained via Monte Carlo simulations.

## VI. Conclusion

In this paper, we propose novel analytical models for the reliability of simultaneous double chipkill detect and/or correct and Double Chip Sparing. We validated our analytical models against Monte Carlo simulations and showed that the mean and standard deviation of the differences between the model and Monte Carlo simulations are within 3.9%.

## References


[1] V. Sridharan and D. Liberty, "A Study of DRAM Failures in the Field," *Super Computing*, 2012.

[2] M. Blaum, R. Goodman, and R. McEliece, "The Reliability of Single Error Protected Computer Memories," *IEEE Transaction on Computers*, 1988.

[3] W. F. Mikhail, R. W. Bartoldus, and R. A. Rutledge, "The Reliability of Memory With Single-Error Correction," *IEEE Transaction on Computers*, 1982.

[4] H. Ahn et al, "Future Scaling of Processor-Memory Interfaces", *Super Computing*, 2009.

[5] M. Ohmacht, R. A. Bergamaschi, and S. Bhattacharya, "Blue Gene/l Compute Chip: Memory and Ethernet Subsystem," *IBM Journal of Research & Developement*, 2005.

[6] HP, "RAS Features of the Mission-Critical Converged Infrastructure," 2010.

[7] S. Mukherjee, J. Emer, T. Fossum, and S.Reinhardt, "Cache Scrubbing in Microprocessors: Myth or Necessity?" *PRDC*, 2004

[8] A.M.Saleh, J.J.Serrano, and J.H.Patel, "Reliability of Scrubbing Recovery Techniques for Memory Systems," *IEEE Transactions on Reliability*, 1990.

[9] L. Schiano, M. Ottavi, and F. Lombardi. "Markov Models of Fault-Tolerant Memory Systems under SEU," *Memory Technology, Design and Testing*, 2004.

[10] P. Reviriego, "Reliability Analysis of Memories Suffering Multiple Bit Upsets," *IEEE Transactions on Device and Materials Reliability*, 2007

[11] J.A. Maestro, "Reliablity of Single-Error Correction Protected Memories," *IEEE Transactions on Reliability*, 2009

[12] T.J. Dell, "A White Paper on the Benefits of Chipkill Correct ECC for PC Server Main Memory," *IBM Microelectronics Division*, 1997.

[13] B. Schroeder, E. Pinheiro, and W.D. Webber, *"DRAM Errors in the Wild: a Large-Scale Field Study,"* SIGMETRICS, 2009

[14] A.A. Huwang, L.A. Stefanovici, and B. Schroeder, "Cosmic Rays Don't Strike Twice: Understanding the Nature of DRAM Errors and the Implications for System Design," *SIGARCH*, 2012


## Appendix

$$P_2(e(x,t|0,k,F)) = e^{-(N-1)\cdot\omega_x\cdot t}\sum_{j=k+1}^{t/s} N\cdot e^{-\omega_x\cdot s\cdot(j-1)}(1-e^{-\omega_x\cdot s})\cdot e^{-\phi_x(N-1)(t-s\cdot(j-1))}P(e(x-1|1,j,T))^{n_x} \quad (6)$$

$$P_3(e(x,t|0,k,F)) = e^{-(N-1)\cdot\omega_x\cdot t}[e^{-\phi_x\cdot N\cdot t}\cdot P(e(x-1,t|1,k,F))^{n_x} + \sum_{l=1}^{t/s}(1-e^{-\phi_x(N-1)s})e^{-\phi_x(N-1)(t-s\cdot l)}\cdot P(e(x-1,t|1,max(k,l),F))^{n_x}]\cdot \sum_{j=1}^{max(k,l)} N(1-e^{-\omega_x\cdot s})e^{-\omega_x\cdot(j-1)\cdot s} \quad (7)$$

$$P_4(e(x,t|0,k,F)) = e^{-N\cdot\omega_x\cdot t}[e^{-\phi_x\cdot N\cdot t}\cdot P(e(x-1,t|0,k,F))^{n_x} + \sum_{l=1}^{t/s}(1-e^{-\phi_x\cdot N\cdot s})e^{-N\cdot\phi_x\cdot(t-s\cdot l)}\cdot P(e(x-1,t|0,max(k,l),F))^{n_x}] \quad (8)$$

$$P_1(e(x,t|1,k,T)) = e^{-\Omega_x(N-2)t}\sum_{j=1}^{t/s}(N-1)e^{-\omega_x\cdot s\cdot(j-1)}\cdot(1-e^{-\omega_x\cdot s})e^{-\Phi_x(N-2)(t-s\cdot(max(j,k)-1))} \quad (9)$$

$$P_2(e(x,t|1,k,T)) = e^{-\omega_x(N-1)t}\{e^{-\phi_x(N-1)t}\cdot P(e(x-1,t|1,k,T))^{n_x} + [\sum_{l=1}^{k-1} P(e(x-1,t|1,k,T))^{n_x} + \sum_{l=k}^{t/s} P(e(x-1,t|1,l,F))^{n_x}]\cdot(1-e^{-\phi_x(N-1)s})e^{-\phi_x(N-1)(t-s\cdot l)}\} \quad (10)$$

$$P_1(e(x,t|1,k,F)) = e^{-\Omega_x(N-2)t}\sum_{j=k+1}^{t/s}(N-1)e^{-\omega_x\cdot s\cdot(j-1)}\cdot(1-e^{-\omega_x\cdot s})e^{-\Phi_x(N-2)(t-s\cdot(j-1))} \quad (11)$$

$$P_2(e(x,t|1,k,F)) = e^{-\omega_x(N-1)t}[e^{-\phi_x(N-1)t}\cdot\cdot P(e(x-1,t|1,k,F))^{n_x} + \sum_{l=1}^{t/s}(1-e^{-\phi_x(N-1)s})e^{-\phi_x(N-1)(t-s\cdot l)}\cdot P(e(x-1,t|1,max(k,l),F))^{n_x}] \quad (12)$$

where $n_x$ represents the total number of $x-1$s in $x$. When $x$ does not contain any sub-regions, $P(e(x-1,t|0,k,F)) = P(e(x-1,t|1,k,T)) = P(e(x-1,t|1,k,F)) = 1$ .