Analytical Reliability Model of Die-Stacked DRAM Protected by Error Control Code and TSV Fault Tolerant Coding Technique

Tadayuki Matsumura and Tsuyoshi Tanaka

Central Research Laboratory, Hitachi Ltd.
1-280, Higashi-koigakubo, Kokubunji-shi, Tokyo, Japan
{tadayuki.matsumura.bh, tsuyoshi.tanaka.vz}@hitachi.com

Abstract - **Die-stacked DRAM is a promising innovation to meet the need for high memory bandwidth in HPC systems. HPC systems must also be reliable yet there is no analytical reliability model and it is difficult to evaluate reliability in a time-efficient manner. This paper proposes analytical reliability models for some type of the die-stacked memory configurations. It is shown that through silicon via (TSV) errors can be catastrophic, and an effective coding technique to solve this problem is proposed. The model is validated in simulation experiments. The reliability of future large-scale system is evaluated on the basis of the proposed model.**

I. Introduction

Die-stacked memories such as high bandwidth memory (HBM) [1] and hybrid memory cube (HMC) [2] are a promising innovation to meet the need for high memory bandwidth in future high performance computing (HPC) systems. HPC systems must be reliable, and it is necessary to quantitatively evaluate their reliability at a system design stage. There are two methods for evaluating reliability. The first is to use an analytical reliability model and the second is to use a simulation such as Monte Carlo. However, Monte Carlo simulations of large-scale systems take too much time. This means that analytical reliability models are more preferable for large-scale systems. Moreover, analytical models can provide more insight about reliability problems through their derivation.

Soft errors are the primary concern as far as ensuring the reliability of memory. There are many studies on fault tolerant techniques such as error control codes (ECCs) or bit-interleaving, and current reliability models consider such fault tolerant techniques [3-6].

A recent field study on DRAM errors shows that hard errors (permanent errors) are also a problem in large-scale computer systems [7]. Previous reliability models have considered the effect of such hard errors [4,5]. However, they cannot be applied to die-stacked DRAM because they do not consider inter-die wire, or through silicon via (TSV), faults.

This paper proposes analytical reliability models for some type of die-stacked memory configurations considering hard errors including TSV faults. Moreover, a TSV fault tolerant error control code technique is proposed. The reliability models are validated in Monte Carlo simulations and then used to analyze the reliability of a large-scale system. The results of this reliability evaluation can form the basis of future research on improving system reliability.
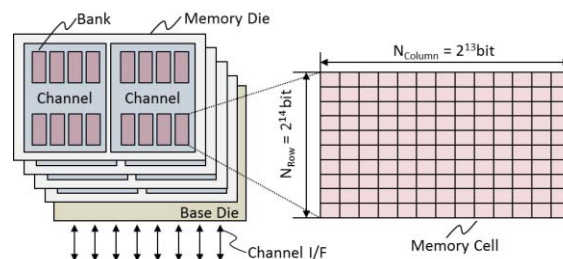

Figure 1: Example of Centralized Channel Architecture.

II. Derivation of Reliability Model for Die-Stacked DRAM

A. Assumptions placed on Memory Configuration

There are several possible architectures for die-stacked memory. In this paper, two memory configurations are considered, the *centralized channel architecture* and the *distributed channel architecture*.

The die has channels in the centralized channel architecture. Each channel is composed of multiple banks composed of a memory cell array. For example, Fig. 1 shows the case of eight channels on four dies. Each channel has eight banks, and each bank is composed of a 2^{14} x 2^{13} bit memory cell array. Although the HBM specification does not place limits on how a physical channel is composed, an HBM will likely have this architecture.

Regarding the distributed channel architecture, a channel is partitioned over multiple dies, and TSVs are shared in a channel. For example, Fig. 2 shows the case of eight channels on four dies, with each channel having eight banks.

As shown in Fig. 3, the channel interface is 144 bits in both architectures. This supposes that the channel is accessed in double 64 bits words with 16 ECC bits. HBM specifies this channel interface. Moreover, as shown in Fig. 4, these 144 bits are divided into multiple partitions in the case of the distributed channel architecture.

B. Assumption placed on Faults and Errors

There are two causes of memory failure, soft errors and hard errors. A soft error is a transient error caused by an alpha particle or neutron hitting a memory cell. A hard error, on the other hand, is a permanent error caused by faults in the hardware at runtime. It is assumed that both types of error
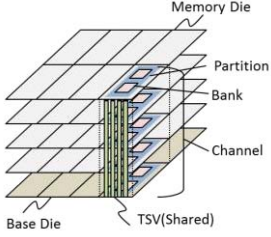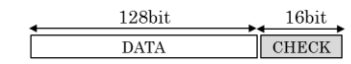
Figure 2: Distributed Channel Architecture.

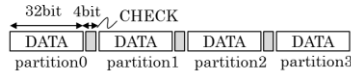**Figure 3:** Channel Interface.

Figure 4: Channel Interface Configuration for Distributed Channel Architecture.
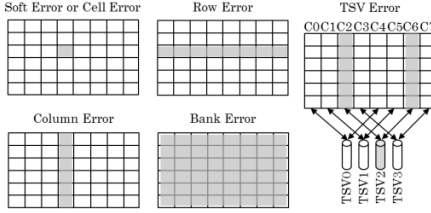

Figure 5: Soft and Hard Errors.

occur at a constant rate. This assumption means that the number of events conforms to a Poisson distribution, and the distribution of event intervals conforms to an exponential distribution.

Recent studies have shown that a neutron hit can cause multi bit upsets (MBUs) and a reliability model considering MBUs has been proposed [6]. However, we will not consider MBUs since they can be treated as multiple single bit upsets in multiple words if the memory interleave technique is appropriately applied. Moreover, it is assumed that more than two temporally accumulated soft errors in a word do not occur. This is because the soft errors can be periodically removed if memory scrubbing is appropriately applied [3].

The six types of error mode are described in Fig. 5. The soft, cell, column, row and bank errors are conventional error modes, and previous reliability models consider them [4,5]. The column, row and bank errors cause multiple bits errors. For example, if a row error occurs, all of the cells in the error row cannot be read or written in reliably.

The TSV error is a novel error mode, and it also causes multiple bit errors. In general, TSVs are shared among several columns. Because of this, if a TSV error occurs, all of the cells that are read or written through the error TSV are not reliable. Consequently, a TSV error can be considered to be a multiple column error as illustrated in Fig. 5.

C. Bit Reordering Technique for TSV error tolerant coding

Currently, the most general error control code (ECC) for DRAM is a SEC-DED (Single Error Correction Double Error Detection) code, which can correct any single bit error and detect any two bit error in a code word. For example, SEC-DED code for 64 bit data can be realized by adding eight redundant check bits [8]. This code is called (72, 64) SEC-DED code. A 144 bit channel interface will likely be used as two (72, 64) code words.
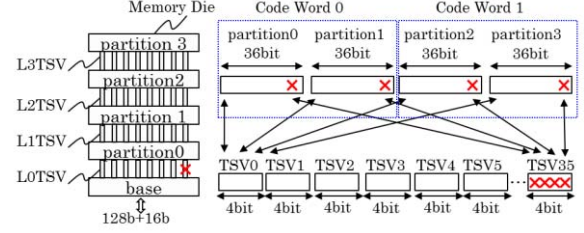

Figure 6: TSV Error Effect and Bit Reordering Technique.

A TSV error can cause a novel error mode in the distributed channel architecture, because the TSV is shared among partitions. The 144 bits for the channel interface is composed of multiple chunks of bits in the distributed channel architecture. For example, four chunks of 36 bits are read from each partition in the case of the four-die stacked memory. In this case, if the TSV at the bottom layer has an error, 1 bit from each partition, totaling 4 bits, is not reliable (Fig. 6). This is a serious reliability problem since SEC-DED code cannot correct this error. Consequently, a TSV error will cause a system failure.

We propose a simple but effective technique named "*bit reordering coding*" for solving this problem. The code word is composed of single 128-bit data bits and a 16-bit check bit, and SbEC-DbED (Single Byte Error Correction and Double Byte Error Detection) code is applied. SbEC-DbED code can correct any 1 byte error and detect any 2 byte errors, where a byte is a chunk of b bits. In the case of $b=4$, S4EC-D4ED code for 128 data bits can be composed by adding 16 check bits [8]. However, if it is naively applied, S4EC-D4ED code may not be effective since the four error bits are distributed throughout the code word as shown in Fig. 6. To effectively apply S4EC-D4ED code, we can reorder the bits on the basis of the TSV used to read/write. The distributed errors can then be reordered into one chunk of 4 bits. Consequently, the TSV error can be corrected using S4EC-D4ED code. Our reliability model for the distributed channel architecture considers both SEC-DED and S4EC-D4ED code cases.

D. Derivation of Reliability Model for Centralized Channel Die-Stacked Memory Architecture

The reliability model for the centralized channel die-stacked memory is derived under the assumptions described in the previous sections. The reliability, R(t), is defined as follows:

Definition 1: Reliability, R(t)
Reliability is the probability that a system will perform a required function under stated conditions for a stated period.

MTTF (Mean Time to Failure) is calculated as

$$\text{MTTF} = \int_0^\infty R(t)\, dt. \qquad (1)$$

The reliability of a channel is independent of those of other channels since failure events in one channel occur independently of fault events in other channels. Therefore, denoting the channel reliability as $R_{channel}(t)$, the reliability of a memory module and a system can be calculated as

$$R_{module}(t) = R_{channel}(t)^{N_{channel}} \qquad (2)$$

$$R_{system}(t) = R_{module}(t)^{N_{module}} \qquad (3)$$

where $N_{channel}$ and N_{module} are the number of channels per die-stacked memory module and the number of memory modules per system, respectively. Accordingly, the system reliability can be calculated by deriving $R_{channel}(t)$.

When SEC-DED code is used, the condition under which a channel failure occurs is that at least one word in the channel has more than two bit errors. Therefore, $R_{channel}(t)$ is the probability of condition 1 below.

Condition 1: The number of errors is zero or one in all of words in a channel during the stated period time.

How the six types of error, i.e., soft errors and cell, column, row, bank and TSV faults (these faults will be referred to as "errors" to simplify the following explanations) factor into the failure occurrence probability will be considered one by one. Each error rate will be described as λ_X, where X is a label denoting cell, column, row, bank, TSV, or soft. For example, λ_{row} means the error rate per row.

Let us consider the effects of the row and bank errors first. Figure 5 shows that these errors clearly cause more than two bit errors in a word. Consequently, if a row or bank error occurs, a channel failure immediately occurs. These types of error are called *catastrophic*. When condition 1 is satisfied, there is no row and bank error in a channel in the stated period of time. Let us derive the probability of an event that there is no row or bank error (E0). Hereafter, *E* expresses an event, and *P(E)* expresses the probability of the event.

Under the assumption of the Poisson distribution, P(E0) can be derived as,

$$P(E0) = r_{bank,row}(t) = e^{-(\lambda'_{bank} + \lambda'_{row})t} \qquad (4)$$

where λ'_{bank} and λ'_{row} are the error rates of bank and row per channel, respectively. Denoting the number of banks and rows as N_{bank} and N_{row}, we have $\lambda'_{bank} = \lambda_{bank} \cdot N_{bank}$ and $\lambda'_{row} = \lambda_{row} \cdot N_{row} \cdot N_{bank}$.

Next, let us consider the effects of cell, column, TSV and soft errors. These errors do not immediately cause failures by themselves because they cause at most one error in each word. However, a combination of these errors can cause a failure. Because of this, these errors must be considered simultaneously. A hierarchical fault tree (HFT) can be used to systematically consider this combinational failure condition. Although HFT is similar to a fault tree, it has a hierarchical structure corresponding to the memory hierarchical structure. A four layer hierarchy is shown in Fig. 7. Each error fits in a hierarchy corresponding to its influence.

The first layer is a *word*. This layer considers only the reliability of a word (r_{word}) relative to cell and soft errors. As described in section II, it was assumed that temporally accumulated two soft errors will not occur in a word. Under this assumption, condition 2 below guarantees that a word is reliable despite cell and soft errors .

Condition 2: There is no cell error (E1), or there is one cell error and no soft error after a cell error occurs (E2).

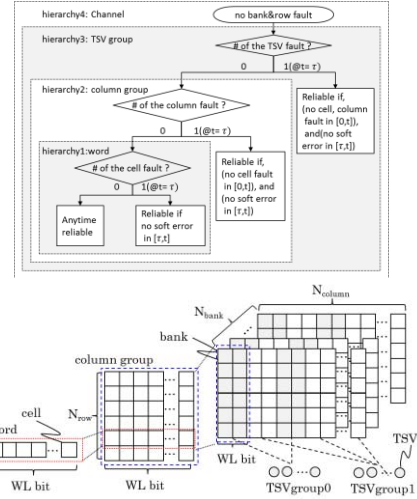**Figure 7:** Hierarchical Fault Tree and Its Corresponding Memory Structure for Centralized Channel Architecture.

An auxiliary function, $r_{aux}(\lambda_{hard}, \lambda_{soft}, t)$ is introduced to derive P(E2). r_{aux} is the sum of the probabilities (P_τ) for [0,t], where P_τ is the probability that one hard error occurs at $t = \tau$, and there is no hard or soft error after τ. r_{aux} is derived as,

$$r_{aux} = \int_0^t \left(\lambda_{hard} \cdot e^{-\lambda_{hard}\tau}\right) e^{-(\lambda_{hard} + \lambda_{soft})(t-\tau)} d\tau. \qquad (5)$$

Equation (5) can be analytically calculated. $r_{word}(t)$ is derived using r_{aux},

$$r_{word}(t) = e^{-\lambda'_{cell}t} + r_{aux}(\lambda'_{cell}, \lambda'_{soft}, t) \qquad (6)$$

where $\lambda'_{cell,soft} = \lambda_{cell,soft} \cdot WL$ (word length), which means the cell and soft error rate per word. In Eq. (6), the first term is P(E1), and the second term is P(E2).

The second layer in HFT is named the *column group (colgrp)*. The effect of column errors is considered at this layer by utilizing $r_{word}(t)$. The column group is a group of words. Columns are shared among all the words in the column group. Accordingly, if a column error occurs, all the words in the same column group have an error.

The column group reliability (r_{colgrp}) for cell, soft and column errors is the probability of condition 3 below.

Condition 3: There is no column error and all of the words in a column group are reliably protected against the cell and soft errors (E3), or there is one column error and no cell error and no soft error after the column error (E4).

Considering the no column error case and one column error case exclusively is essential. P(E3) and P(E4) can be easily derived by utilizing the word hierarchy results, Eq. (6) and r_{aux}, respectively. Namely, r_{colgrp} can be derived as,

$$r_{colgrp} = e^{-\lambda'_{colum}t} \cdot r_{word}(t)^{N_{row}}$$
$$+ e^{-\lambda''_{cell}t} \cdot r_{aux}(\lambda'_{column}, \lambda''_{soft}, t) \qquad (7)$$

where λ'_{column} and $\lambda''_{cell,soft}$ are the column, cell and soft error rates per column group. Denoting the number of rows per column group as N_{row}, we have $\lambda'_{column} = \lambda_{column} \cdot WL$, and $\lambda''_{cell,soft} = \lambda_{cell,soft} \cdot WL \cdot N_{row}$.

The third layer in HFT is named the *TSV group*. TSV group is a group of column groups sharing a TSV. The number of TSV groups corresponds to the channel width (CW) in words. For example, if the channel width is 144 bits and the word length is 64 bits, the CW is 2. The reliability (r_{tsvgrp}) for cell, soft, column and TSV errors is easily derived by noting the hierarchical similarity between the TSV group and the column group. Condition 4 for deriving r_{TSVgrp} is similar to condition 3, as follows.

Condition 4: There is no TSV error and all of the column groups in a TSV group are reliably protected against cell, soft and column errors (E5), or there is no cell and column error, and one TSV error, and no soft error after the TSV error (E6).

From condition 4, r_{tsvgrp} is derived as

$$r_{tsvgrp} = e^{-\lambda'_{TSV}t} \cdot r_{colgrp}(t)^{N_{colgrp}} + e^{-(\lambda'''_{cell}+\lambda''_{column})t} \cdot r_{aux}(\lambda'_{TSV}, \lambda'''_{soft}, t), \quad (8)$$

where N_{colgrp} is the number of column groups in a TSV group, and $\lambda'_{TSV}, \lambda''_{column}$ and $\lambda_{cell,soft}$ are the TSV, column, cell and soft error rates per TSV group. Similar to the discussion of the column group layer, these error rates per TSV group can be derived by multiplying each error rate by the number of components in a TSV group.

Finally, the channel reliability for cell, column, TSV, row, bank errors and soft error is

$$R_{channel}(t) = r_{bank,row} \cdot \left(r_{tsvgrp}\right)^{N_{tsvgrp}} \quad (9)$$

where N_{tsvgrp} is the number of TSV groups in a channel.

E. Reliability Model Derivation of Distributed Channel Die-Stacked Memory Architecture

The reliability models for the distributed channel die-stacked memory architecture protected by SEC-DED or S4EC-D4ED code can be derived in a similar manner to that of the central channel architecture.

First, we derive the reliability model for the SEC-DED code protection case. The reliability model for the distributed channel architecture depends on the number of stacked dies since TSVs are shared among these dies. In this paper, the configuration described in Fig. 6 is assumed to simplify the model derivation process. However, the derivation process can be systematically extended to other configurations. In this configuration, there is a stack of memory dies the first word is constructed from the bits of lower layer partitions (partitions 0 and 1), and the second word is constructed from the bits of partitions 2 and 3.

Figure 8 shows the HFT and the corresponding memory structure for this configuration. As described in Fig. 8, there is no difference between bank and row errors; i.e., both of these errors are catastrophic in this configuration. Therefore, Eq. (4) can be applied for the row and bank error case.

Like bank and row errors, TSV errors can cause immediate failures. If the TSV is between the base die and the lowest memory die (described as L0TSV in Fig. 6), the bits from each partition will have an error. Therefore, if the L0TSV has an error, it causes 2 bits error in each word and it leads a
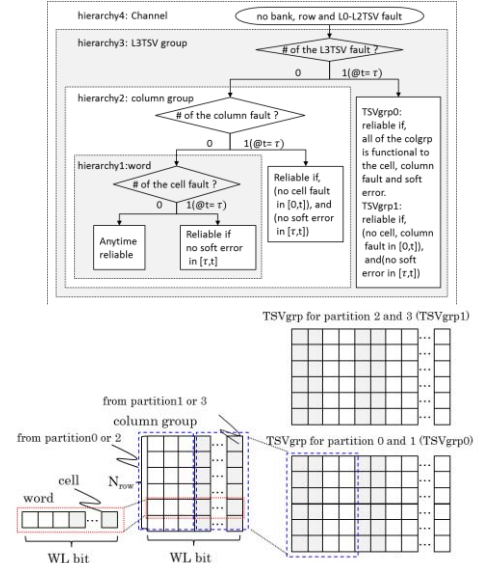

Figure 8: Hierarchical Fault Tree and Its Corresponding Memory Structure for Distributed Channel Architecture.

failure. Similarly, a TSV between the lowest memory die and the 2nd memory die (L1TSV) and between the 2nd memory die and 3rd memory die (L2TSV) causes more than 2 bit errors at least in a word. Therefore, these errors are also catastrophic. Because of this, the effect of the TSV errors in L0TSV, L1TSV and L2TSV can be represented using Eq. 4. Denoting the number of TSVs per partition as N_{TSV}, the reliability for these catastrophic errors can be expressed as,

$$r_{catastrophic}(t) = e^{-(\lambda'_{bank}+\lambda'_{row}+3\lambda_{TSV} \cdot N_{TSV})t} . \quad (10)$$

The reliability against non-catastrophic errors such as the cell, soft, column and L3TSV (TSVs between 3rd die and 4th die) errors will be discussed using the HFT in Fig. 8. Although there is a difference in how to physically construct a word, cell and soft errors do not have different effects from those of the centralized channel architecture. Therefore, Eq. (6) can be applied to the word layer. Similarly, there is no difference in the column group layer in Fig. 8. Therefore, Eq. (7) can also be applied to the column group layer.

However, the TSV group, especially the L3TSV group, is different from the centralized channel architecture case. If the TSV of the L3TSV has an error, only a word constructed from the bits of partition 2 and 3 (word1 in Fig. 6) will have an error. In contrast, a word constructed from the bits of partition 0 and 1 (word0 in Fig. 6) is not affected by an L3TSV error. The TSV group of column groups composed of word0s is denoted as TSVgrp0. Similarly, the group of column groups composed of word1s is denoted as TSVgrp1. The reliabilities of TSVgrp0 and TSVgrp1 against L3TSV errors are different.

In the case of no L3TSV error, the reliabilities of TSVgrp0 and TSVgrp1 are the same. We will denote the reliability of this case as $r_a(t)$. $r_a(t)$ is the probability that all of the column groups in each TSV group are reliable against cell, soft and column errors. Therefore, denoting the number of column groups per TSV group as N_{colgrp}, $r_a(t)$ can be expressed as,

$$r_a(t) = e^{-\lambda'_{tsv}t} \cdot \left(r_{colgrp}(t)\right)^{2N_{colgrp}}, \qquad (11)$$

where λ'_{TSV} is the TSV error rate in L3TSV.

We will denote the reliability in the case of an L3TSV error as $r_b(t)$. $r_b(t)$ is the probability of condition 5.

> ***Condition 5:*** All of the column groups in TSVgrp0 are reliable against cell and column errors, and there are no cell, column or soft errors after the L3TSV error in TSVgrp1.

From condition 5, $r_b(t)$ can be derived as,

$$r_b(t) = \left(r_{colgrp}\right)^{N_{colgrp}} \cdot e^{-(\lambda''''_{cell}+\lambda''''_{column})t} \cdot r_{aux}(\lambda'_{TSV}, \lambda''''_{soft}, t) \qquad (12)$$

where $\lambda''''_{cell,soft}$ and λ''''_{column} are the cell, soft and column error rates per TSV group. Therefore, the reliability against L3TSV errors ($r_{L3TSV}(t)$) can be derived as

$$r_{L3TSV}(t) = r_a(t) + r_b(t). \qquad (13)$$

Accordingly, the reliability of a channel for the channel distributed architecture protected by SEC-DED code is

$$R'_{channel}(t) = r_{catastorophic}(t) \cdot r_{L3TSV}(t). \qquad (14)$$

Next, let us consider the reliability of the S4EC-D4ED code case. The model derivation process for this configuration is almost the same as that of the centralized channel architecture. It is clear that the bank and row errors have the same effect. Therefore, Eq. (4) can be applied. Similarly, Eq. (6) and Eq. (7) can be applied as the word layer and column group layer since the cell, soft and column errors have the same effects.

Moreover, the effect of TSV errors is also the same as in the SEC-DED code case. This is because the condition here is the same as condition 4. Therefore, the reliability in the case of S4ED-D4ED code is given by Eq. (9) with slight modifications to make N_{TSVgrp}, WL or λ correspond to the distributed memory architecture.

III. Model Validation

The reliability models derived in the previous sections were validated by comparing them with Monte Carlo simulation results. However, as the derivation of the reliability model is motivated by the difficulty of simulating a large-scale system, it is difficult to compare results for such a system. Because of this, the model was validated for a small channel. Moreover, although the three types of model were constructed, only the centralized channel architecture case was validated since the essence of the model derivation process is almost the same in all models.

The memory size parameters are listed in Table 1. Three error rate patterns were tested. The parameters for them are also described in Table 1. Pattern 1 is the base case. In this validation, the absolute values of the error rate parameters are not so important. Patterns 2 and 3 are for the lower row and bank errors. Moreover, the TSV error rate is lower and the cell error rate is higher in pattern 2 than in pattern 1. In pattern 3, the soft error rate is lower and the TSV error rate is higher than in pattern 2. The Monte Carlo simulation was iterated 10,000 times.

Table 1: Parameter List for Model Validation Simulation.

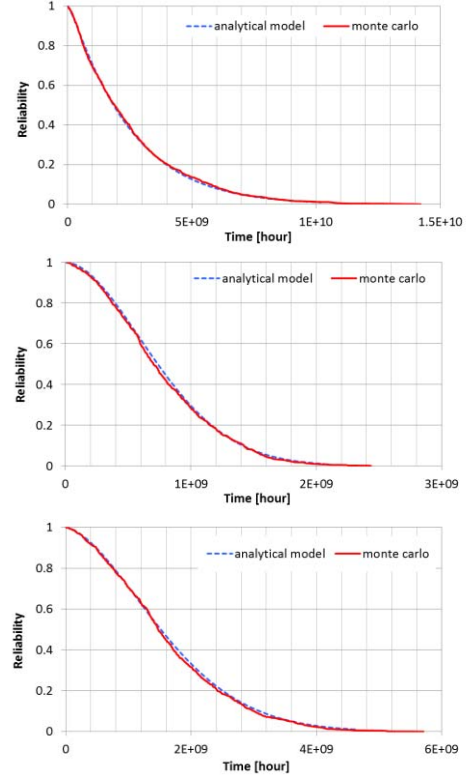| pattern	1	2	3
memory size	$N_{bank} = 8$, $N_{row} = 64$, $N_{column} = 288(256\ data + 32\ check)$		
λ_{cell} [Fit/cell]	$1.7 \cdot 10^{-8}$	$1.7 \cdot 10^{-4}$	$1.7 \cdot 10^{-4}$
λ_{column} [Fit/column]	$8.5 \cdot 10^{-5}$	$8.5 \cdot 10^{-5}$	$8.5 \cdot 10^{-5}$
λ_{row} [Fit/row]	$6.3 \cdot 10^{-5}$	$6.3 \cdot 10^{-6}$	$6.3 \cdot 10^{-6}$
λ_{bank} [Fit/bank]	$1.25 \cdot 10^{-3}$	$1.25 \cdot 10^{-4}$	$1.25 \cdot 10^{-4}$
λ_{TSV} [Fit/TSV]	$1.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-4}$	$5.0 \cdot 10^{-4}$
λ_{soft} [Fit/cell]	$1.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-5}$

Figure 9: Model Validation Results, Pattern 1 (Top), Pattern2 (Middle), and Pattern3 (bottom).

Figure 9 shows the results of the validation. The derived reliability model fits all of the simulation patterns. The results for pattern 1 show that the reliability is almost that same as a simple exponential model, $e^{-\lambda \cdot t}$. This can be understood from Eq. (4). When the row and bank error rates are higher, the reliability is dominated by the catastrophic errors expressed by a simple exponential model. On the other hand, the reliabilities are not a simple exponential model in patterns 2 and 3. Our model also fits these non-trivial cases well. These results indicate that the derived reliability model is valid and sufficient.

IV. Model Used In System Reliability Evaluation

The MTTF for a large-scale system was evaluated on the basis of the derived model. Three memory configurations are assumed. The first is the centralized channel architecture (MemCfg1), whereas the second and third are distributed

Table 2: Memory Size Parameters.

$N_{channel}$	N_{bank}	N_{row}	N_{column}	Total
8	8	2^{14}	2^{13}	1GB

Table 3: Error Rate [Fit] Parameters.

| λ_{cell}	λ_{column}	λ_{row}	λ_{bank}	λ_{TSV}	λ_{soft}
$1.7 \cdot 10^{-8}$	$8.5 \cdot 10^{-5}$	$6.3 \cdot 10^{-5}$	1.25	$1.4 \cdot 10^{-2}$	$1.0 \cdot 10^{-3}$

channel architectures with SEC-DED code (MemCfg2-SECDED) or S4EC-D4ED code (MemCfg2-S4ECD4ED).

1GB die-stacked memory module is assumed. The memory size parameters are listed in Table 2. The error rates assumed in this evaluation are listed in Table 3. The soft error rate was assumed to be 1,000 Fit/Mb [9]. The hard error rates were chosen on the basis of the previous work [7]. However, as far as we know, there is no report on the TSV error rate. Therefore, we assumed 0.014 Fit/TSV as an example. This means that the TSV error rate per channel is roughly 5 fit in the case of MemCfg1. In the case of MemCfg2, the number of TSVs is less than that of MemCfg1 due to sharing the TSVs. Because of this, the TSV error rate per channel is also less in MemCfg2 case.

The results of the MTTF evaluation are shown in Fig. 10. Here, MTTF is less than 100 hours when the system has more than 100,000 memory modules. This points to a serious reliability problem for the next generation of HPC systems because the next generation of HPC systems will have more than 100,000 memory modules. The sensitivity analysis results for the MemCfg1 are also shown in Fig. 11. Here, the bank and row errors have the strongest effect on MTTF. This is also the reason why MTTF is almost inversely proportional to the number of memory modules in Fig. 10. These results show that fault- or error-tolerant techniques for reducing the impact of these catastrophic errors will be essential to improving the reliability of future large-scale computer systems.

V. Conclusion

This paper mainly made three contributions. The first is the derivation of analytical reliability models for die-stacked DRAMs. Our models can be applied two types of die-stacked memory architecture, i.e., centralized channel and distributed channel architectures. Moreover, it was shown that a TSV error can be catastrophic in the case of a distributed channel architecture. The second contribution is the error control code technique to solve this problem. Our model considers this technique. Moreover, it can be extended to other architectures by systematically using the basic idea of hierarchical fault tree analysis. Finally, the reliability of future large-scale systems was evaluated using the derived model. One course of future study would be the development of error-tolerant techniques based on the detailed reliability analysis using the derived reliability model.
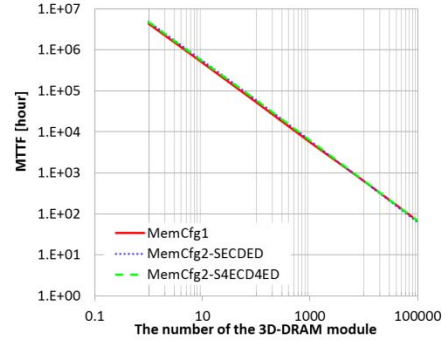
Figure 10: MTTF Evaluation Results
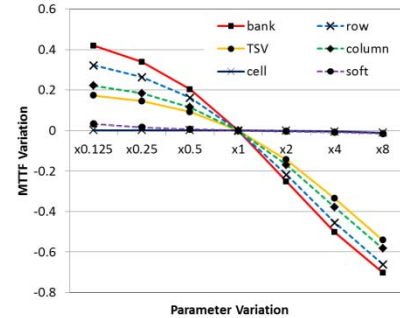
Figure 11: Sensitivity Analysis Result for MemCfg1

References

[1] JEDEC Standard, "High Bandwidth Memory (HBM) DRAM", JESD235, 2013.

[2] Hybrid Memory Cube Consortium, "Hybrid Memory Cube Specification 1.0", 2013.

[3] A. M. Saleh, et al., "Reliability of scrubbing recovery techniques for memory systems", IEEE Trans. Rel., vol. 39, no. 1, pp. 114–122 1990.

[4] M. Blaum, et al., "The reliability of single-error protected computer memories", IEEE Trans. Comput., vol. 37, no. 1, pp. 114–119 1988.

[5] X. Jian, et al.,"Analyzing reliability of memory subsystem with double chipkill detect/correct", in the IEEE Pacific Rim Int. symp. on dependable computing (PRDC), 2013.

[6] P. Reviriego, et al., "Reliability analysis of memories suffering multiple bit upsets", IEEE Trans. Device Mater. Rel., vol. 7, no. 4, pp. 592–601 2007

[7] V. Sridharan and D. Liberty, "A study of dram failures in the field," in Int. Conf. for High Performance Computing, Networking, Storage and Analysis (SC), pp. 1–11, 2012.

[8] E. Fujiwara, "Code design for dependable systems: theory and practical application", Wiley-Interscience, 2006.

[9] Tezzaron Semiconductor. Soft errors in electronic memory. *White paper*, 2004.