Simulation of system backend dielectric reliability

Chang-Chih Chen, Muhammad Bashir, Linda Milor*, Dae Hyun Kim, Sung Kyu Lim

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Abstract

Backend dielectric breakdown degrades the reliability of circuits. A methodology to estimate chip lifetime due to backend dielectric breakdown is presented. It incorporates failures due to parallel tracks, the width effect, field enhancement due to line ends, and variation in activity and temperature. Different workloads are considered as well, in order to evaluate aging effects in microprocessors running real-world applications with realistic use conditions.

1. Introduction

Each technology generation reduces the interconnect dimensions without always reducing the supply voltage in proportion. This results in higher electric fields within the backend dielectric. At the same time, as the dielectric constant ($k$) decreases to reduce parasitics, as prescribed by the International Technology Roadmap for Semiconductors, the porosity of materials must increase, at the possible cost of increasing the vulnerability of materials to breakdown. These factors combine to increase the risk of failure of chips due to backend dielectric breakdown in the newer technology nodes.

To better understand the impact of the backend dielectric on design, this work builds an interface between data collected by reliability physicists and designers by linking test structure data to chip-level lifetime estimates. It is a common assumption that the vulnerable area for backend dielectric breakdown for a full chip is the area between minimum spaced lines [1]. Our work shows that it is necessary to take into account all areas with different line spaces.

In addition to the vulnerable area, we demonstrate that lifetime depends on linewidth, even when the line space is constant, due to aspect ratio dependent etch (ARDE), and also on irregular geometries due to aspects of advanced lithography. Others have demonstrated dependencies on the presence of vias [2] and line edge roughness [3].

Since backend dielectric breakdown is activity and temperature dependent, the proposed framework determines the detailed thermal profile of the system under study, as well as the electrical stress of each dielectric segment in the system.

This work not only accounts for activity and temperature, but also accounts for the fact that systems are not in operation at all times. Realistic use conditions include operation modes, standby, and periods of time when the system is turned off, as illustrated in Fig. 1. This paper takes these use scenarios into account.

In this paper, first, our methodology to estimate lifetime, based on data collected from test structures, is summarized in the next section. Section 3 discusses our test structures and the vulnerable area. Section 4 presents the test data and analysis of the impact of layout geometries on lifetime. Section 5 gives the overview of our system-level aging assessment framework. The methodology to determine model parameters through FPGA emulation is described. In Section 6, we study the lifetimes for the systems from our simulator and present a comparison based on our results. This paper is concluded in Section 7 with a summary.

2. Backend dielectric breakdown models and full chip lifetime estimation

The most important reliability concerns for interconnects are electromigration, stress-induced voiding, and time-dependent dielectric breakdown (TDDB) of the backend dielectric. Our purpose is to consider time-dependent backend dielectric breakdown.

2.1. TDBD models

Models of backend TDDB are of the form [5–8]:

$$\ln TF = A - \gamma E^m,$$  \hspace{1cm} (1)

where $A$ is a constant that depends on the material properties of the dielectric, $\gamma$ is the field acceleration factor, $m$ is 1 for the $E$ model and $1/2$ for the $\sqrt{E}$ model, and $TF$ is the time-to-failure. The
activation energy, \( E \), is a function of temperature \([9]\). Eq. (1) provides a correction between the electric field during use conditions and during accelerated stress tests.

Test structures are stressed with DC stress, while chip dielectrics undergo AC stress. If two signals are randomly switching with a 50% duty cycle, the dielectric between the signals is stressed 50% of the time. Fig. 2 shows scaling to use conditions for 45 nm technology, with a supply voltage of 0.8 V under alternating pulsed stress. The selection of the field acceleration factor strongly impacts lifetime at different line spacings. Time-to-failure is also a function of temperature, modeled with an Arrhenius relationship \([7]\):

\[
\ln TF = B - E_a / T,
\]

where \( B \) is a constant and \( E_a \) is an activation energy that depends on electric field. Eq. (2) provides a correction between chip operating conditions and accelerated stress conditions. There is a concern that stressing at high temperatures can activate failure modes that are not present during use conditions. Hence, stressing at high electric fields is preferred in comparison with testing at high temperatures. Our tests were conducted at 150 °C.

### 2.2. Chip/system lifetime models

It should be noted that circuits wearout for a variety of reasons, both related to devices and interconnect. All of these wearout mechanisms happen simultaneously. It is common to describe reliability mechanisms with a Weibull distribution:

\[
P(\text{TF}) = 1 - \exp \left( -\frac{\text{TF}}{\eta}\right)^\beta,
\]

having two parameters: the characteristic lifetime, \( \eta \), and shape parameter, \( \beta \). The characteristic lifetime is the time-to-failure at the 63% probability point, when 63% of the population has failed, and the shape parameter describes the dispersion of the failure rate population. Typically, the shape parameter is close to one. Given a collection of \( n \) independent wearout mechanisms modeled with Weibull distributions, having parameters \( \eta_i, i = 1, \ldots, n \), and \( \beta_i = i = 1, \ldots, n \), then the characteristic lifetime of the system, \( \eta_{\text{chip}} \), i.e., the time when 63% of the population has failed from any mechanism, is the solution of \([10–12]\):

\[
1 = \sum_{i=1}^{n} \left( \frac{\eta_i}{\eta_{\text{chip}}} \right)^{\beta_i}
\]

Similarly \([11]\),

\[
\beta_{\text{chip}} = \sum_{i=1}^{n} \beta_i \left( \frac{\eta_i}{\eta_{\text{chip}}} \right)^{\beta_i}.
\]

The components in Eqs. (4) and (5) could be different wearout mechanisms, different layers of a chip, different geometries within a layer, or different geometries within a layer at different temperatures. Hence, all a reliability simulator has to do is to (a) determine the characteristic lifetimes and shape parameters for all of the underlying wearout mechanisms and geometries, after all components are scaled for temperature and to use conditions with Eqs. (1) and (2), and (b) apply Eqs. (4) and (5) to solve for \( \eta_{\text{chip}} \) and \( \beta_{\text{chip}} \).

Eq. (4) provides the lifetime of the system when 63% have failed, the Weibull characteristic lifetime. If all the components of a system fail according to a Weibull distribution, then for an arbitrary probability of failure, \( P \), the time-to-failure, \( \text{TF} \), is the solution of the following:

\[
-\ln(1-P) = \sum_{i=1}^{n} \left( \frac{\text{TF}}{\eta_i} \right)^{\beta_i}.
\]

Similarly, (5) provides the slope of the Weibull curve at the x-intercept (63% failure). The slope at other probabilities of failure may be different.

### 3. Vulnerable dielectric area and test structures

The simulator operates by determining the vulnerable area of the chip layout and the corresponding test structure. The vulnerable area is defined as the area of a block of dielectric between the two copper lines separated by linespace \( S \) for length \( L \) and having an area \( S_L \). The feature that is extracted from layouts is the vulnerable length between two lines \( L \) associated with a linespace \( S \), which is a function of the widths of the two adjacent lines, \( W_{Li} \), and \( W_{Ri} \), illustrated in Fig. 3(a). A given layout is analyzed by determining the pairs \( (S/W_{Li}, W_{bi}) \) for each layer for all linespaces surrounded by the linewidths \( W_i \) and \( W_{bi} \).

Test structures that vary area, linespace and linewidth have been implemented \([11,13,14]\). Fig. 3(b) shows a top view of a comb test structures used in this study. The test structure in Fig. 3(b) is used to determine the lifetime of the dielectric between parallel tracks with a specific line spacing, \( S \). This test structure has a fixed linespace, \( S \), and vulnerable length, \( L \). The vulnerable area is \( S_L \). To test the lifetime of such a feature, a voltage difference is applied between the two combs. The current between the combs is monitored to determine the time-to-failure. The data set from several samples is fit with a Weibull distribution to estimate \( \eta \) and \( \beta \).

Because the features on a chip differ from a test structure layout, area scaling must be performed to adjust the lifetime to take into account the difference in vulnerable area between the chip and the test structure. To do this, let \( L_i \) and \( L_f \) be vulnerable lengths of the test structure and chip, i.e., the length of the lines that run in parallel in the test structure and chip, respectively, with the same line space, \( S \). \( \eta_i \) is determined by stressing a test structure with vulnerable linespace \( S \) of length \( L_i \). Then, the corresponding
characteristic lifetime for that feature in the chip is
\[
\eta_i = \eta_1(L_i/L_1)^{1/\beta}
\]  
(7)

Test structures that have several irregular features have been designed in order to determine any impact of field enhancement. Fragments of the test structures are shown in Fig. 4. PTT emphasizes the electric field between parallel routing tracks that end at the same point. TLa and TLb emphasize the electric field between line ends and perpendicular lines. TLb includes additional fringing fields, since the line ends are more widely spaced. TTa and TTb emphasize electric fields between line ends. In TTa, the line ends abut, and in TTb the line ends are in parallel tracks. TLa, TLb, TTa, and TTb have 528 line ends each. The separation between line ends is the same for all test structures.

To account for irregular features, the counts of the features are extracted from the layout. Each adds additional parameters, \(\eta_{PTT}, \beta_{PTT}, \eta_{TLa/b}, \beta_{TLa/b}, \eta_{TTa}, \beta_{TTa}, \eta_{TTb}, \beta_{TTb}\) to (4) and (5). These parameters depend on the number of minimally spaced line ends in each category of the layout. Let us consider the computation of \(\eta_{TLa/b}\) for the sake of illustration. Let us suppose the test structure has \(N_{test}\) minimally spaced line ends, from which \(\eta_{test}\) and \(\beta_{TLa/b}\) are computed. Then, for a layout with \(N_{chip}\) similar line ends, by area scaling:
\[
\eta_{TLa/b} = \eta_{test}(N_{test}/N_{chip})^{1/\beta_{TLa/b}}.
\]  
(8)

4. Test results

Test results indicate a strong impact of area, shown in Fig. 5. Die-to-die linewidth variation creates curvature in failure rate distributions [14]. This curvature does not impact \(\eta\). Hence, first, \(\eta\) is extracted and then used to determine \(\beta\) by area scaling [15]. Specifically, \(\beta\) is determined by finding the best fit of the slope for the ordered pairs, \([\ln \eta_{NX} - \ln \eta_{1X}, \ln (1/N)]\), where \(N\) is the area ratio, i.e. two if the area is 2X larger than the reference. Once \(\eta_{1X}\) and \(\beta\) are known, the failure rate distribution is known for all areas. For instance, \(\eta_{NX} = \eta_{1X} + \ln (1/N)/\beta\).

Lifetime is also impacted by the linewidth on each side of the dielectric segment. Fig. 6(a) shows the failure rate distributions for the test structures with 1X, 3X, and 5X linewidths, with fixed linespace.

These test results show a strong impact of linewidth, when the linespace remains constant. Note that the test structures in Fig. 6(a) simultaneously vary density and linewidth. To isolate the cause of variation, our test structure set also includes a test structure that varies linewidth independently of density [11,13]. It was found that linewidth rather than density determines the lifetime. The most plausible explanation is aspect-ratio-dependent-etching [16,17], where narrow trenches suffer from greater lateral etch near the critical CMP interface.

SEM data was used to determine the difference between the actual linewidth, \(W_d\), and the drawn linewidth, \(W_d,\) i.e. \(\Delta W = W_d - W_d\). This translated into a shift in linespace, i.e. \(S_d = S_d - \Delta W\), where \(S_d\) is the actual linespace and \(S_d\) is the drawn linespace. Linespaces with larger positive values of \(\Delta W\) breakdown faster, since \(E = V/S_d\). SEM data were used to determine \(\Delta W\), by fitting measured data through regression, as illustrated in Fig. 6(b).

The line end features in Fig. 4 are also found to have a significant impact on lifetime. The data collected from the test structures is presented in Fig. 7. An area scaled version of a standard comb test structure is included for comparison. It can be seen that all test structures (PTT, TLa, TLb, TTa, and TTb) result in a significantly reduced lifetime in comparison with the reference test structure. The data also indicate that TLa and TLb fail at the same rate, indicating that fringing fields are not significant. The data from these two test structures can be merged to determine a single model. TLa has an improved lifetime, in comparison with TLa/b. Most likely this is due to line-end pull-back, since TTa experiences twice the pull-back. No reference curve is included for comparison of TTb because TTb has no vulnerable length.

A model was extracted for PTT, TLa/b, TLa, and TTb. The model for TTa and TTb was found with the standard method, involving fitting a linear function to the data to find \(\eta_{PTT}, \beta_{PTT}, \eta_{TTa}, \beta_{TTa}\). Extraction of the model for TLa/b and PTT is more complex since these structures combine both line ends and vulnerable length. To find the model for...
5. Aging assessment framework

5.1. Vulnerable area and vulnerable feature extraction

The layout extraction tool was developed using standard object oriented programming languages and is shown in Fig. 8.

Vulnerable area and features are extracted by comparing pairs of lines in a layout. Since tens of millions of lines exist in each metal layer in a layout, it is necessary to find the adjacent lines that border a vulnerable area or form a critical feature quickly. Therefore, vulnerable area and features are extracted as follows.

First, lines are read from a layout, sorted by the bucket sort algorithm, and stored in two separate data variables, LineDataX and LineDataY. The lines in LineDataX (and LineDataY) are sorted in ascending order of the x-coordinate (y-coordinate) of the bottom left corner of the line. If two lines have the same x-coordinate (or y-coordinate), they are sorted in the ascending order of the y-coordinate (or x-coordinate) of the bottom left corner of the line. The lines with the same x-coordinate (or y-coordinate) are placed in the same bucket.

Then, the extraction process starts by comparing the first ($L_1$) and the second ($L_2$) lines in the first bucket of LineDataY. Since each metal layer has a preferred routing direction (horizontal or vertical), the preferred routing direction is assumed to be horizontal in this description. If the y-coordinates of the two lines in the same bucket are the same, they can form TTa or TLa/b, depending on the distance between them and the direction (horizontal or vertical) of $L_2$. In order to form TTa or TLa/b, the spacing between $L_1$ and $L_2$ must be the minimum distance. If both $L_1$ and $L_2$ are horizontal, they can form a TTa feature. If $L_1$ and $L_2$ are perpendicular, they can form TLa/b. To find TLa/b, LineDataX is searched based on the x-coordinate of the bottom right corner of $L_1$ to find any $L_3$ that can form TLa/b with $L_1$. Besides $L_2$, $L_3$ cannot form any critical features with other lines in the same bucket, because $L_2$ lies between $L_1$ and the other lines in the bucket.

TTb and PTT are extracted by comparing two lines in adjacent buckets in LineDataY. (Lines in different buckets have different y-coordinates.) TTb and PTT are checked by comparing $L_1$ and any $L_2$ in the next bucket. If a TTb or PTT is found, a flag for the corresponding edge of $L_1$ is set. By setting the flag, counting an extra TTb or PTT formed by $L_1$ and any other line, $L_3$, is avoided.
Algorithm 1: Layout extraction flow

Input: The maximum line spacing $S_{max}$ and a layout $L$
Output: Tables of vulnerable lengths (VulnerableLengthTable) and new features (TLa, TTB, TTA, PTT)

for each metal layer $m$ do
  LineDataX($m$) ← ReadLines($L$); // BucketSort
  LineDataY($m$) ← ReadLines($L$); // BucketSort
  $TTa$($m$) ← 0; $TTb$($m$) ← 0; PTT($m$) ← 0; $TLab$($m$) ← 0;
  $c ← 1$;
  $n ← 2$;
  while $c < N_{soc}$ do // $N_{soc}$ # lines in LineDataY
    $L_1 ← LineDataY (m,c)$; // $c$-th line
    $L_2 ← LineDataY (m,n)$; // $n$-th line
    if Spacing($L_1,L_2$) $\leq S_{soc}$ then
      $TTa (m) ← CheckTTa (L_1,L_2)$; // check TTa between $L_1$ and $L_2$
      $TTb (m) ← CheckTTb (L_1,L_2)$; // check TTB between $L_1$ and $L_2$
      $TTa (m) ← CheckTTa (L_1,L_2)$; // check TTa between $L_1$ and $L_2$
      $TTb (m) ← CheckTTb (L_1,L_2)$; // check TTB between $L_1$ and $L_2$
    end
    $n ← Adjust (c, n)$;
    $L_2 ← LineDataY (m,n)$;
    if Spacing($L_1,L_2$) $\leq S_{soc}$ then
      $TTb (m) ← CheckTTb (L_1,L_2)$; // check TTB between $L_1$ and $L_2$
      $TTa (m) ← CheckTTa (L_1,L_2)$; // check TTa between $L_1$ and $L_2$
      $TTb (m) ← CheckTTb (L_1,L_2)$; // check TTB between $L_1$ and $L_2$
      $TTa (m) ← CheckTTa (L_1,L_2)$; // check TTa between $L_1$ and $L_2$
    end
    $n ← Adjust (c, n)$;
    $L_2 ← LineDataY (m,n)$;
    if Spacing($L_1,L_2$) $\leq S_{soc}$ then
      $PTT (m) ← CheckPTT (L_1,L_2)$; // check PTT between $L_1$ and $L_2$
      $TTb (m) ← CheckTTb (L_1,L_2)$; // check TTB between $L_1$ and $L_2$
      $TTa (m) ← CheckTTa (L_1,L_2)$; // check TTa between $L_1$ and $L_2$
      VulnerableLengthTable($m$) ← VulnerableLength ($L_1,L_2$);
      $LineDataY (m) ← Split (L_1,L_2)$;
      $n ← Adjust (c, n)$;
    end
    $n ← Adjust (c, n)$;
  end
  $c ← c + 1$;
end

Fig. 8. Algorithm for layout pattern extraction.

since there may be other lines, $L_x$, that have the same x-coordinate as $L_2$ and that are within the minimum distance. TLa/b can also be found by comparing lines in adjacent buckets, if $L_2$ is perpendicular to $L_1$.

After extracting irregular features formed by $L_1$ and its adjacent lines, the algorithm searches for a vulnerable length associated with $L_1$. To form a vulnerable length, $L_2$ must be in a different bucket. $L_2$ is the first line in the sorted list where the area between the x-coordinates overlap. If the vertical spacing is less than or equal to the maximum line spacing, a vulnerable area surrounded by these two lines exists. The linespace is computed, and the vulnerable length is added to the vulnerable length table for the corresponding linespace. Then $L_1$ is split into one or two new lines; they are inserted into LineDataY, and $L_1$ is removed from LineDataY.

Fig. 9 shows an example with four line segments, $S_1$, $S_2$, $S_3$, and $S_4$. They are ordered according to their bottom left coordinate. The algorithm starts with the first line segment, $S_1$. First the irregular features are checked. In checks for irregular features, the algorithm looks for specific geometries that are within a minimum distance from $S_1$. Hence, for $S_1$, PTT, TTA, TTb, and TLa/b are checked. The TTA geometry only appears if the segments are on the same track.

A TLa relation between $S_1,1$ and $S_2$ is shown in Fig. 9(e). It is not included in the TTB count, because the left side of $S_1,1$ was generated during the split operation. On the other hand, the right side of $S_1,1$ is the right side of the original line, $S_1$, and there exists a TTB relation between $S_1,1$ and $S_2$ as shown in Fig. 9(f). Hence, the TTB count is incremented. $S_1,1$ does not have a minimum distance with any other line segment. It does not have a vulnerable area either. Therefore, the algorithm proceeds to the next line $S_2$. $L_1$ is set to $S_2$ by the index adjustment function. The algorithm checks for irregular geometries that are separated from $S_2$ by the minimum distance. It finds a TLa/b relation between $S_2$ and $S_3$, as shown in Fig. 9(g). It also finds a TTA relation between $S_2$ and $S_4$ in Fig. 9(h). The TLa/b and TTA counts are incremented.

The runtime for the simulator is the sum of the time taken to extract features from the layout and a constant time to evaluate Eqs. (4) and (5). Complexity of feature extraction is $O(n)$, where $n$ is the number of features, since bucket-sort is used. Complexity of extracting statistics from features is also $O(n)$, because the bucket is scanned from the bottom most element, and the maximum number of features within a fixed distance from an element is constant. Lifetime is estimated in constant time.

5.2. Extraction of the stress and thermal profiles

Because backend dielectric breakdown is activity and temperature dependent, our methodology includes determining the temperature and stress for each dielectric segment while running benchmarks. A framework for acquisition of spatial and temporal thermal/electrical stress of the system was constructed. Fig. 10 summarizes the electrical and thermal profile acquisition flow.

For activity tracking, the hardware RTL/netlist was synthesized for emulation on an FPGA, and counters were placed at the I/O ports, which track both the state probabilities and the toggle rates of the ports during application runtime, as illustrated in Fig. 11. A standard set of benchmarks was used as the applications for analysis [19].

The I/O activities and the gate-level netlist were then used for activity propagation to each net in the design, for a complete stress/transition probability profile of the internal nodes of a circuit under study. This provides the probability of a transition occurring at any node and the probability at each state, i.e., the probability at logic “1”. It is this probability at logic “1” and logic “0” that is needed to compute the probability that each dielectric segment is under stress. The probabilities of dielectric stress of each dielectric segment is determined by

$$a = \alpha_1 (1-\alpha_2) + \alpha_2 (1-\alpha_1)$$

(9)

where $\alpha_1$ and $\alpha_2$ are the probabilities that each net in the pair of nets that border the dielectric segment are at logic “1”. If $\eta_{dc}$ is the characteristic lifetime under dc stress, then the characteristic lifetime under ac stress, $\eta_{ac}$, is

$$\eta_{ac} = \eta_{dc}/\alpha.$$  

(10)

Note that it is sufficient to find the total fraction of time under stress for each and every dielectric segment because backend dielectric breakdown does not show recovery. Eq. (9) has been verified by comparing the exact stress durations of random-selected vulnerable dielectric segments from an example system layout with the ones calculated. The result, as illustrated in Fig. 12, shows the percent errors are less than 15% for more than 80% of the selected samples. Stress profiles for an example microprocessor are shown in Figs. 13 and 14.

The netlist is also used for layout generation. The RC information from the layout, together with the net activity, is used for extraction of the power profile and the consequent thermal profile, through the power simulator [20] and the thermal simulator [21]. An example of temperature profile is shown in Fig. 15.
Note that computational complexity in accounting for operating conditions is dominated by the complexity to propagate activities within a circuit block, which is $O(n)$, where $n$ is the number of gates in the system. Hence, the overall complexity of the simulator is $O(n)$.

### 5.3. Workload dependent aging analysis

Eqs. (4) and (5) only provide a failure rate distribution for one mode of operation. We need to be able to combine multiple modes to provide a lifetime under use conditions. Let $\xi_{\text{active}}$ be the fraction of time in active mode. Let $\xi_{\text{standby}}$ be the fraction of time in standby mode. And, let $\xi_{\text{off}} = 1 - \xi_{\text{active}} - \xi_{\text{standby}}$ be the fraction of time in the off state. Let the active mode Weibull parameters be $\eta_{\text{active}}$ and $\beta_{\text{active}}$. Similarly, the standby mode Weibull parameters are $\eta_{\text{standby}}$ and $\beta_{\text{standby}}$.

The impact of multiple operation modes is a change in the failure rate per unit time. Let $h(t) = \left(\frac{\beta_{\text{active}}}{\eta_{\text{active}}}\right)^{\frac{T_F}{\eta_{\text{active}}}} - 1 + \xi_{\text{standby}} \left(\frac{T_F}{\eta_{\text{standby}}}\right)^{\frac{1}{\beta_{\text{standby}}}}$ be the number of failures per unit time, divided by the number of remaining units for a Weibull distribution. Therefore, for multiple modes of operation,

$$h(t) = \xi_{\text{active}} \left(\frac{\beta_{\text{active}}}{\eta_{\text{active}}}\right)^{\frac{T_F}{\eta_{\text{active}}}} + \xi_{\text{standby}} \left(\frac{T_F}{\eta_{\text{standby}}}\right)^{\frac{1}{\beta_{\text{standby}}}}.$$  

(11)
The cumulative probability of failure is \( P = 1 - e^{-\int h(t) dt} \). Hence

\[
P(\text{TF}) = 1 - \exp \left( - \zeta_{\text{active}} \left( \frac{TF}{\eta_{\text{active}}} \right)^{\beta_{\text{active}}} - \zeta_{\text{standby}} \left( \frac{TF}{\eta_{\text{standby}}} \right)^{\beta_{\text{standby}}} \right).
\]

The characteristic lifetime corresponds to \( P = 1 - e^{-1} \). Therefore, the overall characteristic lifetime, \( \eta_{\text{use}} \), is the solution of

\[
1 = \zeta_{\text{active}} \left( \frac{\eta_{\text{use}}}{\eta_{\text{active}}} \right)^{\beta_{\text{active}}} + \zeta_{\text{standby}} \left( \frac{\eta_{\text{use}}}{\eta_{\text{standby}}} \right)^{\beta_{\text{standby}}}.
\]

If \( \beta \) is constant, then there is closed form solution:

\[
\eta_{\text{use}} = \left( \frac{\zeta_{\text{active}}}{\beta_{\text{active}}} + \frac{\zeta_{\text{standby}}}{\beta_{\text{standby}}} \right)^{-1/\beta}.
\]

6. Lifetime estimations for the systems

6.1. Case study 1: LEON3 microprocessor

The well-known open-source LEON3 IP core processor with superscalar abilities [22] was studied. The microprocessor logic units consist of a 32-bit general purpose integer unit (IU), a 32-bit multiplier (MUL), a 32-bit divider (DIV) and a memory management unit (MMU). Storage blocks include a window-based register file unit (RF), separate data (D-Cache) and instruction (I-Cache) caches and cache tag storage units (Dtags and Itags). The microprocessor includes around 240 k gates.

By weighting the lifetimes of operation, standby, and off mode in accordance with Fig. 1, we have estimated the lifetime of each unit within the microprocessor and analyzed the lifetime for each metal layer in the design technology used under different use scenarios, as shown in Figs. 16 and 17.

The lifetime of the microprocessor is clearly limited by the Metal 1 layer. Higher metal layers are associated with increased metal spacing, resulting in an increase in time-to-failure. The data-cache and the instruction-cache are the lifetime limiting units in the microprocessor. On-line reconfiguration, through redundancy allocation, was not considered here, but could improve the lifetime of these units. Among the combinational blocks, lifetime is limited...
6.2. Case study 2: 32-bit RISC microprocessor

Besides LEON3, a 32-bit RISC microprocessor [23] which includes around 73 k gates was also studied.

Fig. 18 shows the estimated lifetime for each metal layer of the RISC microprocessor. Similar to the results for LEON3, the lifetime of the microprocessor is closely limited by the Metal 1 layer. As we move up in the metal layer stack, the metal spacing increases, resulting in an increased time-to-failure. Regarding the use scenarios, gaming has the worst lifetime result, while office work has the best result.

7. Conclusion

A methodology has been proposed to assess backend dielectric reliability of full chips. It takes into account all vulnerable dielectric areas with all line spaces in a layout. It also takes into account linewidth and many critical geometries, as well as the temperature and electrical stress profiles under realistic use conditions.

Acknowledgment

The authors would like to thank AMD for providing the wafers for this study and Changsoo Hong and Sohrab Aftabjahani for designing and laying out the test structures. The authors would also like to thank the National Science Foundation under Award Number 0901576 for financial support.

References