Bad Page Detector for NAND Flash Memory

Yi Liu, Si Wu, and Paul H. Siegel

Center for Memory and Recording Research, UCSD
Overview

1. Bad Page Detector
2. Time Dependent Neural Network
3. Long-Short Term Memory
4. Experimental Result
5. Conclusion
Overview

1. Bad Page Detector
2. Time Dependent Neural Network
3. Long-Short Term Memory
4. Experimental Result
5. Conclusion
NAND Flash Memory

- NAND flash page: basic program and read unit. Page varies significantly.
- Test board: 1X nm TLC flash memory.
- P/E cycling experiment: program pseudo-random data to TLC, for every 100 P/E cycles, record the page bit error counts.
- Test range: 4,000 - 10,000 P/E cycles.
NAND Flash Memory

- If the bit error counts exceed the capacity of error correction codes, data is lost, we say this page is a “bad” page.
- Bad page detector: predict whether the page is a bad page in the future.
**Definition of Bad Pages**

**Definition**

In a flash memory device, a certain physical page is called a \((T, T_{\text{offset}}, N)\)-bad page if the number of bit errors at some P/E cycles \( \leq T + T_{\text{offset}} \) is not less than \( N \).
System Overview

- A bad page detector for 1X nm TLC flash memories.
- Bit error counts measured on 12855 pages (4285 wordlines).
- Measurements: every 100 P/E cycles from 4K to 10K P/E cycles.
- A sliding window bad page detector that detects \((T, 500, 1200)\)-bad page.
- At \(T_4 (N_4)\), the page is labeled as a bad page, because its bit error count surpasses \(N\) for the first time at \(T_9 (N_9)\)

Figure: Sliding window bad page detector
System Overview

- A bad page detector for 1X nm TLC flash memories.
- Bit error counts measured on 12855 pages (4285 wordlines).
- Measurements: every 100 P/E cycles from 4K to 10K P/E cycles.
- A sliding window bad page detector that detects \((T, 500, 1200)\)-bad page.
- At \(T_4\) \((N_4)\), the page is labeled as a bad page, because its bit error count surpasses \(N\) for the first time at \(T_9\) \((N_9)\).

![Bad Page Detector Diagram]

**Figure:** Sliding window bad page detector
System Overview

- A bad page detector for 1X nm TLC flash memories.
- Bit error counts measured on 12855 pages (4285 wordlines).
- Measurements: every 100 P/E cycles from 4K to 10K P/E cycles.
- A sliding window bad page detector that detects \((T, 500, 1200)\)-bad page.
- At \(T_4\) \((N_4)\), the page is labeled as a bad page, because its bit error count surpasses \(N\) for the first time at \(T_9\) \((N_9)\).

Figure: Sliding window bad page detector
Training Features

- A sliding window bad page detector that detects \((T, 500, 1200)\)-bad page.
- Training features: five consecutive BECs (from P/E cycles \(T - 400\) to \(T\)).
- At \(T_4\), the page is labeled as bad page, because its bit error count surpasses \(N\) for the first time at \(T_9\).
- If a detector uses training features from \(N_{-1}\) to \(N_3\) and classifies this page as a bad page, we say this page is a \(-1\) bad page.
- **Negative** bad page is a waste of storage capacity.
- A \(+5\) bad pages indicates the detector failed to detect a bad page.
- **Q—bad page**: early, ontime, late prediction.
Training Features

- A sliding window bad page detector that detects $(T, 500, 1200)$-bad page.
- Training features: five consecutive BECs (from P/E cycles $T - 400$ to $T$).
- At $T_4$, the page is labeled as bad page, because its bit error count surpasses $N$ for the first time at $T_9$.
- If a detector uses training features from $N_{-1}$ to $N_3$ and classifies this page as a bad page, we say this page is a -1 bad page.
- Negative bad page is a waste of storage capacity.
- A +5 bad pages indicates the detector failed to detect a bad page.
- Q–bad page: early, ontime, late prediction.

\[
\begin{align*}
N_{-1} & N_0 N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 \ldots
\end{align*}
\]

\[ -1 \]
Training Features

- A sliding window bad page detector that detects \( (T, 500, 1200) \)-bad page.
- Training features: five consecutive BECs (from P/E cycles \( T - 400 \) to \( T \)).
- At \( T_4 \), the page is labeled as bad page, because its bit error count surpasses \( N \) for the first time at \( T_9 \).
- If a detector uses training features from \( N_{-1} \) to \( N_3 \) and classifies this page as a bad page, we say this page is a \(-1\) bad page.
- Negative bad page is a waste of storage capacity.
- A \(+5\) bad pages indicates the detector failed to detect a bad page.
- \( Q \)-bad page: early, ontime, late prediction.

\[
N_{-1} N_0 \boxed{N_1 N_2 N_3 N_4 N_5} N_6 N_7 N_8 N_9 \ldots + 1
\]
Training Features

▶ A sliding window bad page detector that detects \((T, 500, 1200)\)-bad page.
▶ Training features: five consecutive BECs (from P/E cycles \(T-400\) to \(T\)).
▶ At \(T_4\), the page is labeled as bad page, because its bit error count surpasses \(N\) for the first time at \(T_9\).
▶ If a detector uses training features from \(N_{-1}\) to \(N_3\) and classifies this page as a bad page, we say this page is a -1 bad page.
▶ Negative bad page is a waste of storage capacity.
▶ A +5 bad pages indicates the detector failed to detect a bad page.
▶ \(Q\)-bad page: early, ontime, late prediction.

\[
N_{-1} N_0 N_1 N_2 N_3 \boxed{N_4 N_5 N_6 N_7 N_8} N_9 \ldots + 4
\]
Training Features

- A sliding window bad page detector that detects \((T, 500, 1200)\)-bad page.
- Training features: five consecutive BECs (from P/E cycles \(T - 400\) to \(T\)).
- At \(T_4\), the page is labeled as bad page, because its bit error count surpasses \(N\) for the first time at \(T_9\).
- If a detector uses training features from \(N_{-1}\) to \(N_3\) and classifies this page as a bad page, we say this page is a -1 bad page.
- Negative bad page is a waste of storage capacity.
- A +5 bad pages indicates the detector failed to detect a bad page.
- \(Q\)-bad page: early, ontime, late prediction.

\[
N_{-1} N_0 N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 \ldots + 5
\]
Examples of $Q$-Bad Pages

- An example of *early* bad page, storage capacity is wasted:
Examples of $Q$-Bad Pages

▶ An example of ontime bad page, successfully detected:

![Graph showing bit error count vs. P/E cycle count]
Examples of Q-Bad Pages

- An example of late bad page, failed to be detected as a bad page:

![Graph showing bit error count vs. P/E cycle count]
Overview

1. Bad Page Detector

2. Time Dependent Neural Network

3. Long-Short Term Memory

4. Experimental Result

5. Conclusion
Given a input vector $x = [N_0, N_1, N_2, N_3, N_4]$ at P/E cycle $t$, a dense layer in
a feed-forward neural network do the following calculation

$$y = Ax + B. \quad (1)$$

$A \in \mathbb{R}^{5 \times 5}$ is a matrix and $B \in \mathbb{R}^5$ is a vector, weights are determined during
the training process.
Time Dependent Neural Network

- Given a input vector $\mathbf{x} = [N_0, N_1, N_2, N_3, N_4]$ at P/E cycle $t$, a dense layer in a feed-forward neural network do the following calculation

$$y = A\mathbf{x} + B.$$  \hfill (1)

- $A \in \mathbb{R}^{5 \times 5}$ is a matrix and $B \in \mathbb{R}^5$ is a vector, weights are determined during the training process.

- Time dependent dense layer (TDDL): $A \Rightarrow A(t)$, $B \Rightarrow B(t)$

$$y = A(t)\mathbf{x} + B(t).$$  \hfill (2)
Time Dependent Neural Network

- Given a input vector \( x = [N_0, N_1, N_2, N_3, N_4] \) at P/E cycle \( t \), a dense layer in a feed-forward neural network do the following calculation

  \[
  y = Ax + B. \tag{1}
  \]

- \( A \in \mathbb{R}^{5 \times 5} \) is a matrix and \( B \in \mathbb{R}^5 \) is a vector, weights are determined during the training process.

- Time dependent dense layer (TDDL): \( A \Rightarrow A(t), B \Rightarrow B(t) \)

  \[
  y = A(t)x + B(t). \tag{2}
  \]

- The output of the network is a non-negative pair \( [P_1, P_2] \) \( (P_1 + P_2 = 1) \). Given a threshold \( P_{Th} \) (by default, set \( P_{Th} = 0.5 \)), when \( P_1 > P_{Th} \), we say the page is a bad page. We can control the network’s behavior by changing \( P_{Th} \).
Time Dependent Neural Network

Time Dependent Dense Layer

- Time dependent dense layer (TDDL): $y = A(t)x + B(t)$
- Approximate $A(t)$ and $B(t)$ by polynomial functions

\[
A(t) = \{a_{ij}(t)\} = \begin{bmatrix} a_{11}(t) & a_{12}(t) & \ldots & a_{15}(t) \\ \vdots & \vdots & \ddots & \vdots \\ a_{51}(t) & a_{52}(t) & \ldots & a_{55}(t) \end{bmatrix} \quad a_{ij}(t) = \sum_{k=0}^{L} w_{ij}^k t^k \quad (3)
\]

\[
A(t) = \{a_{ij}(t)\} = \left\{ \sum_{k=0}^{L} w_{ij}^k t^k \right\} = \sum_{k=0}^{L} \{w_{ij}^k\} t^k = \sum_{k=0}^{L} W_k t^k. \quad (4)
\]
Time Dependent Neural Network

- Time dependent dense layer (TDDL): \( y = A(t)x + B(t) \)
- Approximate \( A(t) \) and \( B(t) \) by polynomial functions

\[
A(t) = \{a_{ij}(t)\} = \begin{bmatrix}
a_{11}(t) & a_{12}(t) & \cdots & a_{15}(t) \\
\vdots & \ddots & \vdots \\
a_{51}(t) & a_{52}(t) & \cdots & a_{55}(t)
\end{bmatrix}
\]

\[
a_{ij}(t) = \sum_{k=0}^{L} w_{ij}^k t^k \quad (3)
\]

\[
A(t) = \{a_{ij}(t)\} = \left\{ \sum_{k=0}^{L} w_{kj}^i t^k \right\} = \sum_{k=0}^{L} \{w_{kj}^i\} t^k = \sum_{k=0}^{L} W_k t^k. \quad (4)
\]

- A degree-\( L \) TDDL can be realized by connecting \( L + 1 \) dense layers in parallel.
- Example: Degree-3 TDDL (cubic TDDL)

\[
a_{ij}(t) = w_{3}^j t^3 + w_{2}^j t^2 + w_{1}^j t + w_{0}^j \quad (5)
\]

- Connect 4 (3+1) dense layers in parallel, each multiplied by \( t^i \)
Overview

1. Bad Page Detector
2. Time Dependent Neural Network
3. Long-Short Term Memory
4. Experimental Result
5. Conclusion
Input: $X = [N_0, N_1, N_2, N_3, N_4]$

Output: estimate of bit error count $\hat{N}_9$

Given a threshold $N_{Th}$ (by default, set $N_{Th} = 1200$), if $\hat{N}_9 > N_{Th}$, we say the page is a bad page.

Output of TDNN: $[P_1, P_2]$, indicating good or bad page
Overview

1. Bad Page Detector
2. Time Dependent Neural Network
3. Long-Short Term Memory
4. Experimental Result
5. Conclusion
Performance: $Q$–bad Page Revisited

- $Q$–bad page: early, ontime, late prediction

\[ N_{-1} N_0 N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 \ldots 0 \]

- Early prediction is a waste of storage capacity.
- A late (+5) bad page indicates the detector failed to predict a bad page.
- Label pages into three groups: Group I (early prediction), Group II, Group III (late prediction)
- The goal of the bad page detector is to reduce the number of late prediction (Group III) while also controlling the number of wasted P/E cycles (Group I).
Histogram of Q-Bad Pages

- Bit error counts measured on 12855 pages (4285 wordlines).
- Y axis: the number of pages that are a $x$-bad page in the dataset.
- Change the ratio of each group by adjusting $P_{Th} (N_{th})$.

**Figure:** Cubic TDNN performance with input size 5 ($P_{Th} = 0.5$)
Experimental Result

Histogram of Q-Bad Pages

- Bit error counts measured on 12855 pages (4285 wordlines).
- Y axis: the number of pages that are a $x$—bad page in the dataset.
- Change the ratio of each groups by adjusting $P_{Th}$ ($N_{th}$).

Figure: Cubic TDNN performance with input size 5 ($P_{Th} = 0.2$)
Experimental Result

Histogram of Q-Bad Pages

- Bit error counts measured on 12855 pages (4285 wordlines).
- Y axis: the number of pages that are a $x-$bad page in the dataset.
- Change the ratio of each groups by adjusting $P_{Th}(N_{th})$.

Figure: LSTM performance with input size 5 ($N_{Th} = 1200$)
Experiment Result

Histogram of Q-Bad Pages

- Bit error counts measured on 12855 pages (4285 wordlines).
- Y axis: the number of pages that are a $x$-bad page in the dataset.
- Change the ratio of each groups by adjusting $P_{Th}$ ($N_{th}$).

**Figure:** LSTM performance with input size 5 ($N_{Th} = 1163$)
Experimental Results

(a) Cubic TDNN, length 5
(b) Cubic TDNN, length 7
(c) Cubic TDNN, length 3
(d) Linear TDNN, input 5
(e) Constant TDNN, input 5
(f) Constant TDNN, input 5
(g) SVM, input 5
## Experimental Results

<table>
<thead>
<tr>
<th>Network</th>
<th>Input length</th>
<th>Degree</th>
<th>Polynomial function</th>
<th>#mispredicted pages</th>
<th>wasted P/E cycles($\times 100$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cubic TDNN</td>
<td>5</td>
<td>3</td>
<td>$f(t) = at^3 + bt^3 + ct + d$</td>
<td>375</td>
<td>15982</td>
</tr>
<tr>
<td>Cubic TDNN</td>
<td>7</td>
<td>3</td>
<td>$f(t) = at^3 + bt^3 + ct + d$</td>
<td>378</td>
<td>16012</td>
</tr>
<tr>
<td>Cubic TDNN</td>
<td>3</td>
<td>3</td>
<td>$f(t) = at^3 + bt^3 + ct + d$</td>
<td>378</td>
<td>17436</td>
</tr>
<tr>
<td>Linear TDNN</td>
<td>5</td>
<td>1</td>
<td>$f(t) = at + b(1 - t)$</td>
<td>375</td>
<td>17196</td>
</tr>
<tr>
<td>Constant TDNN</td>
<td>5</td>
<td>0</td>
<td>$f(t) = a$</td>
<td>376</td>
<td>19064</td>
</tr>
<tr>
<td>Constant TDNN</td>
<td>5</td>
<td>0</td>
<td>$f(t) = a$</td>
<td>358</td>
<td>19519</td>
</tr>
<tr>
<td>SVM</td>
<td>5</td>
<td>0</td>
<td>$f(t) = a$</td>
<td>358</td>
<td>19761</td>
</tr>
<tr>
<td>Cubic TDNN</td>
<td>5</td>
<td>3</td>
<td>$f(t) = at^3 + bt^3 + ct + d$</td>
<td>72</td>
<td>37830</td>
</tr>
<tr>
<td>LSTM</td>
<td>5</td>
<td></td>
<td>$N_{Th} = 1200$</td>
<td>1449</td>
<td>3888</td>
</tr>
<tr>
<td>LSTM</td>
<td>5</td>
<td></td>
<td>$N_{Th} = 1163$</td>
<td>376</td>
<td>16084</td>
</tr>
</tbody>
</table>

- TDNN with higher degree has better performance.
- TDNNs with input size 5 and 7 outperform TDNN with input size 3.
- TDNN and LSTM have similar performance.
- About 97% of the bad pages can be detected, the average wasted P/E cycles is 124.
Functions in TDNN

Figure: Functions in the cubic TDNN with input length 5.

- Cubic TDNN with input length 5
- Functions in the first TDDL

\[ w_{11}(t) = 0.236t^3 - 0.020t^2 - 0.361t - 0.032. \]
Overview

1. Bad Page Detector
2. Time Dependent Neural Network
3. Long-Short Term Memory
4. Experimental Result
5. Conclusion
A bad page detector predicts whether a page will become a “bad” page in the near future based on its current and previous BEC information.

Two machine learning algorithms, based on time dependent neural network and long short-term memory architecture, are used to design the detector.

Time dependent dense layer can be realized by connecting dense layers in parallel.

The detector can predict more than 97% of the bad pages.

Trade-off between early prediction and late prediction.