Error Detection and Correction Method For Simultaneous Testing In Memories

1Shalini.G.V, 2V.Logeswari

1 PG scholar, Department of ECE, Nandha Engineering College, Erode, India.
2Assistant Professor, Department of ECE, Nandha Engineering College, Erode, India.

Abstract:

ECC techniques have been widely used to correct transient errors and improve the reliability of memories. To prevent MCUs from causing data corruption, more complex error correction codes are widely used to protect memory. Simultaneous testing of data bit and check bit arrays are complicated while transient errors are occur in memories. Scaling technology has raised soft errors to become one of the major sources for processor crashing in many systems in the nano scale era. Soft errors caused by charged particles are dangerous primarily in high atmospheric, where heavy alpha particles are available. A new SEC-DAEC code is proposed for simultaneous testing of the most general memory fault models in both data bit and check bit arrays of memories. Simultaneous testing of data bit and check bit arrays eliminates the test time and hardware overheads required for separate check bit array tests. In order to test data bit and check bit arrays simultaneously, the proposed SEC-DAEC code generates the identical data background patterns for data bit and check bit arrays. The SEC-DAEC code can be implemented by Xilinx and Modelsim.

Keywords: Testing Of Data Bit And Check Bit, SEC-DAEC code.


1 Introduction

ECC is a method used to detect and correct errors introduced during data storage or transmission. Certain kinds of RAM chips inside a computer implement this technique to correct data errors and are known as ECC Memory. ECC Memory chips are predominantly used in servers rather than in client computers. Since servers typically contain several Gigabytes of RAM and are in operation 24 hours a day, the likelihood of errors cropping up in their memory chips is comparatively high, hence they require ECC Memory. Memory errors are of two types, hard and soft. Fabrication defects in the memory chip cause hard errors, which cannot be corrected once they start appearing. On the other hand, electrical disturbances predominantly cause soft errors.

1. Ram Parity Ram

Parity checking is the storing of a redundant parity bit representing the parity (odd or even) of a small amount of computer data (typically one byte) stored in random access memory, and the subsequent comparison of the stored and the computed parity to detect whether a data error has occurred. The parity bit was originally stored in additional individual memory chips; with the introduction of plug-in DIMM, SIMM, etc. modules, they became available in non-parity and parity (with an extra bit per byte, storing 9 bits for every 8 bits of actual data) versions. 1.1 MEMORY ERRORS: In earlier times faulty memory was relatively common, and parity errors, very noticeable to the user, not infrequent. Since then errors have become less visible as simple parity RAM has fallen out of use; either they are invisible as they are not detected, or they are corrected invisibly with ECC RAM. Modern RAM is believed, with much justification, to be reliable, and error-detecting RAM has largely fallen out of use for non-critical applications. Most machines in the twenty-first century do not support parity or ECC, with consequent risk of data corruption; this has become acceptable as a consequence of the increased reliability of memory. Some machines that support parity or ECC allow checking to be enabled or disabled in the BIOS, permitting cheaper non-parity RAM to be used. If parity RAM is used the chipset will usually use it to implement error correction, rather than halting the machine on a single-bit parity error.

1.2 Error Correction:

A single-bit error that would be ignored by a system with no error-checking, would halt a machine with parity checking, or would be invisibly corrected by ECC: a single bit is stuck at 1 due to a faulty chip, or becomes changed to 1 due to background or cosmic radiation; a spreadsheet storing numbers in ASCII format is loaded, and the number “8” is stored in the byte which contains the stuck bit as its eighth bit; then another change is made to the spreadsheet and it is stored. However, the “8” (00111000 binary) has become a “9” (00111001). But in this method we can able to

ISSN (Online): 2348 – 3539.
determine only single bit error and it will increase more
time constraints. In SEC-DAEC code brings significant
decreases in the time required for memory array tests in
varies data bits per word. It can able to detect and correct
the single bit error and multiple bit error.

2. Hamming Codes With Additional Parity (Sec-
Ded)

Extended Hamming code having one extra parity bit is
often used. Extended Hamming codes achieve a Hamming
distance of 4, which allows the decoder to distinguish
between when at most one 1-bit error occurs and when any
2-bit errors occur. This extended Hamming code is popular
in computer memory systems, where it is known as SECDED. Particularly popular is the (72, 64) code.

II. Proposed System

Testing Of Data Bit And Check Bit

The SEC-DED-DAEC codes may be an attractive
alternative to bit interleaving in providing greater flexibility
for optimizing the memory layout. Furthermore the SEC-
DED-DAEC code can be used in conjunction with bit
interleaving and this method can efficiently deal with
adjacent multi-bit errors.

Method For Simultaneous Testing

In order to test data bit and check bit arrays
simultaneously for the same fault model, the data patterns
and the read and write operation sequence for the tests
should be identical for the data bit and check bit arrays. In
addition, the test responses for check bit arrays should be
evaluated together with data bit arrays. Because the
identical data background (DB) patterns cannot be
generated using just any ECC, an appropriate ECC is
required in order to generate identical DB patterns for data
bit and check bit arrays. If certain regularity can be found in
DBs for data bit arrays, the identical DB patterns for data bit
and check bit arrays can be generated to conform to that
regularity Figure 1 shows the block diagram of the ECC
processing circuit for simultaneous testing. The ECC
processing circuit generally consists of the following four
units: 1) a check bit generator, 2) a syndrome generator, 3) an
error locator, and 4) a corrector. In order to detect
occurrences of errors, an error detector is additionally
required. It is assumed that the identical DB patterns for
data bit and check bit arrays are already gen-erated using
the check bit generator based on the appropriate ECC. In order
to evaluate the occurrences of errors, two test responses are
used. The first test response is the corrected DB in which
errors are corrected by the ECC techniques. The second test
response is the error detection signal which indicates the
existence of errors regardless of the number of erroneous
bits and is the OR-sum of the syndromes.

<table>
<thead>
<tr>
<th>Number of erroneous bits</th>
<th>Error detection signal</th>
<th>Corrected DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>No error</td>
<td>0</td>
<td>= Inputted DB</td>
</tr>
<tr>
<td>Single bit</td>
<td>1</td>
<td>= Inputted DB</td>
</tr>
<tr>
<td>Adjacent double bit</td>
<td>1</td>
<td>= Inputted DB</td>
</tr>
<tr>
<td>Non-adjacent double bit</td>
<td>1</td>
<td>≠ Inputted DB</td>
</tr>
<tr>
<td>Triple or more bit</td>
<td>0 or 1</td>
<td>≠ Inputted DB</td>
</tr>
</tbody>
</table>

Table 1 Error detection signal and corrected DB
according to the number of erroneous bits

Table 1 tabulates the error detection signal and corrected
DB according to the number of erroneous bits. For the cases
of no error, a single bit error, or an adjacent double bit error
in the ECC word, the corrected DB is identical for the
inputted DB when the SEC-DAEC code is used. However,
the corrected DB is not identical for the inputted DB when a
non-adjacent double or more bit error occurs in the ECC
word. When there is no error in the ECC word, the
syndromes become zero vectors and the error detection
signal becomes zero. When a single or double bit error
occurs in the ECC word, the syndromes become non-zero
vectors and the error detection signal becomes one.
However, some syndromes generated by triple or more bit
errors in the ECC word can be a zero vector and the error
detection signal becomes zero. Consequently, it is unknown
whether the error detection signal becomes zero or one
when triple or more bit errors occur in the ECC word. The
occurrences of errors in the data bit and check bit arrays can
be evaluated using a combination of the error detection
signal and corrected DB. If the error detection signal is zero
and the corrected DB is the same as the inputted DB, there
is no error.

Conventional Sec-Ded Code

The single-error-correction, double-error-detection, and
double-adjacent-error-correction (SEC-DED-DAEC) codes
have previously been presented to correct adjacent double
bit errors. The required number of check bits for the SEC-
DED-DAEC codes is the same as that for the SEC-DED
codes. In addition, the area and timing overheads for encoder and decoder of the SEC-DED-DAEC codes are similar to those of the SEC-DED code. EXTENDED HAMMING CODE A \((2^r^2 + r, 2^r^2)\) extended Hamming code is designed by adding an overall check bit in a word for the \((2^r^2 + r-1, 2^r^2)\) Hamming code.

That is, the H-matrix of the extended Hamming code is formed by adding a row with all 1’s, and also by adding a 1-weight column with upper r-1 0’s to the H-matrix of the Hamming code.

\[
\begin{align*}
c_0 &= d_{1} + d_{4} + d_{5} + d_{8} + d_{10} + d_{12} + d_{14} \\
c_1 &= d_{0} + d_{2} + d_{4} + d_{6} + d_{7} + d_{11} + d_{13} \\
c_2 &= d_{1} + d_{3} + d_{7} + d_{9} + d_{14} \\
c_3 &= d_{0} + d_{1} + d_{2} + d_{4} + d_{10} + d_{13} + d_{15} \\
c_4 &= d_{2} + d_{5} + d_{6} + d_{9} + d_{12} \\
c_5 &= d_{0} + d_{3} + d_{5} + d_{7} + d_{8} + d_{11} + d_{15} \ldots \ldots \ldots (6.1)
\end{align*}
\]

A 4-bit sequence is repeated in 16-bit DBs during memory test, the relations of data bits are as follows:

\[
\begin{align*}
d_0 &= d_4 = d_8 = d_{12} \\
d_1 &= d_5 = d_9 = d_{13} \\
d_2 &= d_6 = d_{10} = d_{14} \\
d_3 &= d_7 = d_{11} = d_{15} \ldots \ldots \ldots \ldots (6.2)
\end{align*}
\]

Because the modulo-2 sum of any even number of the same value is zero, check bits are generated using the following relations:

\[
\begin{align*}
c_0 &= d_4 + d_8 + d_{12} = d_0 \\
c_1 &= d_{13} = d_1 \\
c_2 &= d_{14} = d_2 \\
c_3 &= d_{15} = d_3 \\
c_4 &= d_{12} = d_0 \\
c_5 &= d_5 = d_1 \ldots \ldots \ldots \ldots, (6.3)
\end{align*}
\]

The six check bits become the same as six lower data bits due to the \((22,16)\) proposed SEC-DAEC code when a 4-bit sequence is repeated in DBs. Consequently, the six check bits can successfully serve as the DBs for single-cell faults, inter word coupling fault and intra word coupling fault.

**IV Simulation Result**

![Figure 2 Simulation result of error locator](image)

**Figure 2 Simulation result of error locator**

![Figure 3 Simulation result of error corrector](image)

**Figure 3 Simulation result of error corrector**

The Table 2 represents the previous method parity generator and checker used to find and correct the soft error. This method can only detect the single bit error and accessing of macro cells is 69%. Single bit error and it will increase more time constraints. In SEC-DAEC code brings significant decreases in the time required for memory array tests in varies data bits per word. It can able to detect and correct the single bit error and multiple bit error.

**Table 2 Comparison Table Between Existing Method And Proposed Method**

<table>
<thead>
<tr>
<th>Fault reduction</th>
<th>Single cell upset using parity</th>
<th>Single cell and multiple cell upset using H-matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macro cells Used</td>
<td>49/72 (69%)</td>
<td>78/144 (55%)</td>
</tr>
<tr>
<td>Perms Used</td>
<td>107/580 (30%)</td>
<td>0/720 (0%)</td>
</tr>
<tr>
<td>Registers Used</td>
<td>23/72 (32%)</td>
<td>0/144 (0%)</td>
</tr>
<tr>
<td>Pins Used</td>
<td>37/52 (72%)</td>
<td>78/0 (1%)</td>
</tr>
<tr>
<td>Function Block Inputs Used</td>
<td>62/216 (29%)</td>
<td>0/432 (0%)</td>
</tr>
</tbody>
</table>

In proposed method the addition of parity bits used in SEC-DEAC code .This code can correct sing bit error and multiple bit error and it reduces the accessing of macro cells is 55%. The overall bits computation is eliminated by check bit pre-computation during the write operation of memory despite using the error locator and double error detection method, coinciding with those of the extended Hamming code.

**V. Conclusion**

The SEC-DAEC code generates the identical DB patterns for data bit and check bit arrays. The test time and hardware overheads required for separate check bit array tests are reduced by simultaneous testing using the proposed SEC-DAEC codes. Moreover, the reduction in the number of 1’s in the H-matrix of the proposed SEC-DAEC reduces hardware overhead. As a benefit for using the proposed SEC-DAEC code in memories, the cost of testing highly reliable memories with single bit error and adjacent double
bit error correcting capabilities can be made more practical. The overall bits computation is eliminated by check bit pre-computation during the write operation of memory despite using the error locator and double error detection method, coinciding with those of the extended Hamming code. It is able to test the most general memory fault models such as single-cell faults, inter word CFs, and intra word CFs in both data bit and check bit arrays simultaneously.

VI. Future Work

The proposed SEC-DED code brings trying to decreases in the power consumption of the ECC processing circuits for 32, 64, and 128 data bits in a word, respectively. Furthermore, the probabilities of miscorrected triple error and undetected quadruple error in the proposed SEC-DED codes match those of the odd-weight-column codes. By implementing the ECC processing circuits based on the proposed SEC-DED code with check bit pre-computation, the costs of employing SEC-DED code in memories can be made more feasible.

References