Traditionally memory cells were the only circuitry susceptible to transient faults. The supporting circuitries around the memory were assumed to be fault free. Due to the increase in soft error rate logic circuits encoder and decoder circuitry around the memory block have become susceptible to soft errors as well and must be protected. The main key novel development is the fault-secure detector (FSD) error-correcting code (ECC) definition and associated circuitry that can detect errors in the received encoded vector despite experiencing multiple transient faults in its circuitry. The structure of the detector is general enough that it can be used for any ECC that follows our FSD-ECC definition. We prove that two known classes of Low-Density Parity-Check Codes have the FSD-ECC property: Euclidean Geometry and Projective Geometry codes. We identify a specific FSD-LDPC code that can tolerate up to 33 errors in each memory word or supporting logic that requires only 30% area overhead for memory blocks of 10 Kbits or larger. Larger codes can achieve even higher reliability and lower area overhead. We quantify the importance of protecting encoder and decoder (corrector) circuitry and illustrate a scenario where the system failure rate (FIT) is dominated by the failure rate of the encoder and decoder.

**Keywords:** Fault tolerant, FSD-ECC, FSD-LDPC, Encoder, Decoder

**INTRODUCTION**

Memory systems are protected against transient upsets of data bits using ECCs. Hamming codes are often used in today’s memory systems to correct single error and detect double errors in any memory word. In these memory architectures, only errors in the memory words are tolerated and there is no preparation to tolerate errors in the supporting logic (i.e. encoder and corrector). However combinational logic has already started showing susceptibility to soft errors, and therefore the encoder and decoder (corrector) units will no longer be immune from the transient faults. Furthermore, memory system designed with nanotechnology devices are expected to experience even higher transient fault
rate [3] [5]; therefore, protecting the memory system support logic implemented with nanotechnology devices is even more important. Here we proposed a fault tolerant memory system that tolerates multiple errors in each memory word as well as multiple errors in the encoder and corrector units. We illustrate using Euclidean Geometry codes and Projective Geometry codes to design the above fault-tolerant memory system, due to their well-suited characteristics for this application, which include:

1) Memory applications require low latency encoders and decoders.

2) These codes allow us to design a fault tolerant error-detector unit that detects any error in the received code-vector despite having faults in the detector circuitry in other words we can design a fault-secure detector for these codes. We use the fault secure detector unit to check the output vector of the encoder and corrector circuitry, and if there is any error in the output of either of these units, that unit has to redo the operation to generate the correct output vector. Using this detect-and-repeat technique we can correct potential transient errors in the encoder or corrector output and provide fault-tolerant memory system with fault-tolerant supporting circuitry. In the rest of this paper we first show

Section 2 : A brief over- view of ECC’s and memory system

Section 3 : An overview of our fault tolerant memory system

Section 4: Presents a stronger ECC definition that can have fault secure detector capability.

Section 5 : Reviews Euclidean geometry and prove that these codes are fault secure.

Section 6 : The technique and an efficient implementation of an one step majority logic corrector is demonstrated and the reliable systematic encoder is presented

Section 7 : Conclusion and future scope.

OVERVIEW OF ERROR CORRECTION CODE

This section provides a brief introduction on ECC’s. Let $I=(i_0, i_1, ..., i_{k-1})$ be k-bit information vector that will be encoded into n bit code word $C=(c_0, c_1, ..., c_{n-1})$ for linear codes the encoding operation essentially per- forms the following vector matrix multiplication.

$$C = I \times G$$

where $G$ is a $k \times n$ generator matrix. The checking or detecting operation is the following vector matrix multiplication

$$S = C \times H^T$$

where $H$ is an $(n-k) \times n$ parity check matrix and the $(n-k)$bit vector. $S$ is called syndrome vector. A syndrome vector is zero if $c$ is a valid codeword and non zero if $c$ is an erroneous codeword. A code word is a systematic code, if any codeword consist of the original k-bit information vector followed by n-k parity it’s with this definition; the generator matrix of a systematic code must have the following structure

$$G = [I : X]$$

where $I$ is a $k \times k$ identity matrix and $X$ is a $k \times (n-k)$ matrix that generates the parity bits.

The advantage of using systematic codes is that there is no need for a decoder circuitry to extract the information bits. The information bits are simply available in the first k bits of any encoded vector. A code is said to cyclic code if
for correct codeword, the code- word is then stored in the memory.

An n-bit codeword c, which encodes a k-bit information vector i is generated by multiplying the k-bit information vector with k × n bit generator matrix G, i.e. c = i · G.

Corrector
During memory access operation, the stored codeword will be accessed from the memory unit. Codeword’s are susceptible to transient faults while they are stored in the memory. There for a corrector unit is designed to correct potential errors in the retrieved code words. In our de- sign, all the memory words pass through corrector and any potential error in the memory words will be correct- ed. Similar to the encoder unit, a fault secure detector monitors the operation of the corrector unit.

Memory Block
Data bits stay in memory for a number of cycles and during this period, each memory bit can be upset by a transient fault with certain probability. Therefore, transient errors accumulate in the memory words over time. In order to avoid accumulation of too many errors in any memory word that surpasses that code correction capability, the system must perform memory scrubbing. Memory scrubbing is the process of periodically reading memory words from the memory, correcting any potential errors and writing them back into the memory. To perform the periodic scrubbing operation, the normal memory access operation is stopped and the memory performs the scrub operation.

ECC WITH FAULT TOLERANT ENCODER AND DECODER
Single-error fault-tolerant decoder and encoder circuits for Reed-Solomon codes have been suggested in (Daniele Rossi et al., 2008). In this work the encoder is protected with parity-prediction and parity checker. The decoder is protected by adding a code checker (detector) block and a hamming distance counter block to count the number of error bits at the output of the decoder. If the code checker detects a non-codeword, then the error in the decoder is detected. If the code checker detects a codeword but the hamming-distance counter indicate a non-zero error, then an error is also detected. Here we propose a multiple-error fault tolerant decoder and encoder that is general enough for any decoder and encoder implementation and for any kind of ECC that satisfies the restricted ECC definition. The restricted ECC definition which guarantees a fault-secure detector capable ECC (FSD-ECC) is as follows: Let C be an ECC with minimum distance d. C is FSD-ECC if it can detect any combination of overall d - 1 or fewer errors in the received codeword and in the detector circuitry.

Theorem I: Let C be an ECC, with minimum distance d. C is FSD-ECC if any error vector of weight e d" d - 1, has syndrome vector of weight at least d - e. Note: The following proof depends on the fact that any single error in the detector circuitry can corrupt at most one output (one syndrome bit). This can be easily satisfied for any type of circuitry by implementing the circuit in such a way that no logic element is shared among multiple output bits, therefore any single error in the circuit corrupt at most one output (one syndrome bit).

Proof: The core of a detector circuitry is a multiplier that implements the vector-matrix multiply of the received vector and the parity-check matrix to generate the syndrome vector. Now if e errors strike the received code-vector
the syndrome weight of the error pattern is at least $d - e$ from the assumption. Furthermore, the maximum number of tolerable errors in the whole system is $d - 1$ and $e$ errors already exist in the encoded vector, therefore the maximum number of errors that can strike in the detector circuitry is $d - 1 - e$. From the above note, this many errors can corrupt at most $d - 1 - e$ syndrome bit, which in worst case leaves at least one non-zero syndrome bit and therefore detects the errors. Q.E.D

The difference between FSD-ECC and normal ECC is the demand on syndrome weight: i.e., a normal ECC demands non-zero syndrome weight while FSD-ECC demands $\geq d - e$.

**EUCLIDEAN GEOMETRY CODES AND FSD-ECC**

The following example illustrated the explanation about the proposed algorithm. Let $(21, 11)$ be a Euclidean Geometry code, a triple error occurs then decoded into another valid code word can trigger words that can cause miscorrection and silent data corruption (SDC). The proposed decoder reduces the SDC by keeping the Euclidean Geometry code $(21, 11)$ as same as one shown in Figure 1. But it takes four equations to check at least to perform a one bit correction. There are two important observations to get the exact knowledge about the proposed design. In first design, a circuit is used instead of the majority of the inputs to a circuit which requires a minimum of four, then three miscorrection is that if the error does not occur. The second modified this decoder is used, the error bit of a double-fault again equation may affect the approval of only one bit can be rest time to heal another error is corrected. Therefore, the value of one of the four remaining equations will check and some will be corrected.

Conventionally, there will be $2(t + 1)$ equations to check for a Euclidean Geometry code which can correct $t$ errors. Then, instead of ones that are at least $(t + 2)$ a majority (a little correction) that requires no miscorrection there with $(t + 1)$ errors and errors are corrected at the same time $t$ will provide the warranty. As described earlier, when there are $(t + 1)$ errors, the increment in the majority level to perform a correction avoids the performing a miscorrection for decoding.

**ONE STEP MAJORITY LOGIC CORRECTOR**

One step majority logic correction is a fast and relatively compact error correcting technique. The core of the one step majority logic corrector is generating $\gamma$ parity check sum from the

<table>
<thead>
<tr>
<th>Parameter</th>
<th>EG-LDPC</th>
<th>PG-LDPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information bits (k)</td>
<td>$2^{2S} - 3^c$</td>
<td>$2^{2S} + 2^s - 3^c$</td>
</tr>
<tr>
<td>Length (n)</td>
<td>$2^{2S} - 1$</td>
<td>$2^{2S} + 2^s + 1$</td>
</tr>
<tr>
<td>Minimum distance (d)</td>
<td>$2^{2S} + 1$</td>
<td>$2^{2S} + 2$</td>
</tr>
<tr>
<td>Dimensions of the Parity-Check matrix</td>
<td>n x n</td>
<td>n x n</td>
</tr>
<tr>
<td>Row weight of the Parity-Check matrix ($\rho$)</td>
<td>$2^{2S} + 1$</td>
<td>$2^{2S} + 1$</td>
</tr>
<tr>
<td>Column weight of the Parity-Check matrix ($\lambda$)</td>
<td>$2^{2S} + 1$</td>
<td>$2^{2S} + 1$</td>
</tr>
</tbody>
</table>
appropriate rows of the parity check matrix. There are few ECC's known to be one step majority correctable, including type–I 2 dimensional EG-LDPC. These codes, the one step majority logic corrector corrects up to $\gamma/2$ error, bits in the received encoded vector, by computing a parity check sum of $n$ code bits each. Each parity sum is implemented with $n$ input XOR function. The majority value of the parity check sums are then evaluated with $a$ input majority gate. If the majority value is 1 then the code bit under consideration holds an erroneous value and has to be inverted. For cyclic codes including EG, a single serial majority corrector circuit can be used. For all the code bits, where the received encoded vector is cyclic shifted and fed into the XOR gates to correct each code bit.

**SIMULATION RESULTS AND CONCLUSION**

In this paper, we have developed a memory system that can tolerate and correct errors not only in the storage unit but also in the supporting circuitry. We used Euclidean geometry codes. We prove that these codes are FSD- ECC. Using these FSD we design a fault tolerant encoder and corrector, where the fault secure detector monitors the operation.
Figure 2: Simulation Results of FSED with Error at Output Side

Figure 3: Simulation Results of FSED without Error at Output Side
REFERENCES


