2010 5th International Conference on Industrial and Information Systems, ICIIS 2010, Jul 29 - Aug 01, 2010, India

Soft Decision Decoding of Davydov-Tombak Codes Using a Parity Check Tree

Prashantha Kumar H, Sripati U, Rajesh Shetty K.
Electronics and Communication Engineering
National Institute of Technology, Karnataka
Surathkal, India.

B. Shankarananda
Mangalore Institute of Technology & Engineering
Moodabidri, India.

***Abstract*—Davydov and Tombak have designed an excellent single error correction-double error detection (SEC-DED) code that appears to be more capable of detecting triple and quadruple errors than the conventional Hamming SEC-DED codes. These codes have been applied to memory subsystems and digital storage devices in order to achieve efficient and reliable data processing and storage. A new approach to soft decision decoding of Davydov-Tombak codes using a parity check tree associated with the Tanner graph is presented. For the AWGN channel, gains in excess of 1.6dB at reasonable bit error rates with respect to conventional hard decision decoding are demonstrated for the (40, 33), (37, 30), (35, 28) and (72, 64) Davydov-Tombak codes.**

Keywords − Davydov-Tombak codes, data storage systems, parity check matrix, tree diagram, Tanner graph, soft decision decoding, horizontal and vertical decoding steps.

I. INTRODUCTION

Error control codes enable a decoder to recover from errors produced by noise in a communication channel. Error control coding (ECC) algorithms have constituted a significant enabler in the telecommunications revolution, the internet, digital recording and space exploration. ECC is nearly ubiquitous in modern, information based society. The last decade has been characterized not only by an exceptional increase in data transmission and storage but also by a rapid development in microelectronics, providing us with both a need for and the possibility of implementing sophisticated algorithms for error control [1].

Every computer memory and data storage systems have adopted some types of error detecting or error correcting codes in order to enhance system reliability. This in turn enables us to create, use and manipulate data as we wish. The reliability levels that are required by storage devices are extremely high since unlike communication systems generally no retransmission is possible. We expect to save our data and be able to retrieve it perfectly at any future time. It is the art of error control and correction that makes this possible. Permanent and temporal faults are the major sources of errors in modern digital storage systems. Power supply breakdown, defective open or short circuits, bridging or open lines, electron-migration etc. cause permanent faults. Permanent fault leads to hard errors; they therefore affect the system functions for a long period of time. Temporal faults can be transient or intermittent. Transient faults occur randomly and externally because of external noise, namely electromagnetic waves and also particles such as α – particles and neutrons. Intermittent faults occur randomly but internally because of unstable or marginally stable hardware, varying hardware or software state as a function of load, or signal coupling (i.e., crosstalk) between adjacent signal lines. Some intermittent faults may be due to glitches which are unpredictable spike noise pulses occurring and propagated especially in large combinational digital circuits. Temporal fault leads to soft errors. Soft errors have a limited duration, meaning they interrupt system functions for a very short time period. Therefore soft errors are also called transient errors. Some reports show that more than 60% of all failures in computer systems are caused by transient or intermittent faults. For example, in dynamic random access memory (DRAM) chips, transient errors result mainly from α – particles emitted by the decay of radio active particles. As they pass through the chip, α – particles create sufficient electron-hole pairs to add charge to the DRAM capacitor cells. These particles have low energy level, and thus have very low probability of causing more than one memory cell to flip when the memory cells are not packed in extreme density. In today’s ultra-high density RAMs, not only DRAMs but also static RAMs, it has been recognized that multiple cosmic ray induced transient errors are a serious problem [2, 3, 4, 5].

Error detection is an essential part of a storage system design. Ideally, error detection will block the propagation of an error during online operations, before it reaches the system interface and causes a system failure. The error is best be detected immediately as it occurs so that its effect can be minimized. Many different error control codes have been studied and developed to correct and/or detect the types of errors mentioned above. Error correcting codes head the list of the most effective and efficient technique used to mask faults, both temporal and permanent. The coding approach involves some redundancy, for example, additional check bits, additional hardware in the form of encoding/decoding logic circuits, and additional decoding time delay. Nevertheless, the coding performance is superior to those competitive techniques, especially in quickly masking of temporal faults [6].

II. CONSTRUCTION OF DAVYDOV-TOMBAK CODES

According to the manner in which redundancy is added to messages, error correcting codes can be divided into two classes: Block and Convolutional. Block codes implement a one-to-one mapping of a set of K information symbols on to a set of N code symbols. We call this code is an (N, K) linear block code. The $R = N - K$ symbols in a codeword are a function of the information symbols, and provide redundancy that can be used for error correction and/or detection purposes. The minimum distance d_{min} of a block code $\boldsymbol{C}$ is the smallest Hamming distance between any two codewords in the code. The Davydov-Tombak code [7, 8] has $d_{min} = 4$. Since a nonzero N-tuple of weight 3 or less is not a codeword, any set of three columns of the parity check matrix **H** should be linearly independent. The **H** matrix of this code has the following form:

$$\mathbf{H} = \begin{bmatrix} \mathbf{B}_0 & \mathbf{B}_1 & \mathbf{B}_2 & \ldots\ldots & \mathbf{B}_{D-1} \\ \mathbf{Q} & \mathbf{Q} & \mathbf{Q} & \ldots\ldots & \mathbf{Q} \end{bmatrix}.$$

Here **Q** consists of the following 4×5 matrix, and $\mathbf{B}_i = [b_i , b_i , b_i ,\ldots\ldots, b_i]$, i = 0, 1, 2,……, D −1, where $D = 2^{R-4}$, is an $(R - 4)\times 5$ matrix consisting of identical columns b_i, where b_i is the binary representation of i:

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

The maximum codeword length (in bits) of the code is $N = 5{\cdot}2^{R-4}$ for $R \geq 5$. For example, the parity check matrix of the (40, 33) code with $R = 7$ is

$$\mathbf{H} = \left[\begin{matrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&1&1&1&1&1&0&0&0&0&0&1&1&1&1&1 \\ 1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1 \\ 0&1&0&0&1&0&1&0&0&1&0&1&0&0&1&0&1&0&0&1 \\ 0&0&1&0&1&0&0&1&0&1&0&0&1&0&1&0&0&1&0&1 \\ 0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1 \end{matrix}\right.$$

$$\left.\begin{matrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&1&1&1&1&1&0&0&0&0&0&1&1&1&1&1 \\ 1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1 \\ 0&1&0&0&1&0&1&0&0&1&0&1&0&0&1&0&1&0&0&1 \\ 0&0&1&0&1&0&0&1&0&1&0&0&1&0&1&0&0&1&0&1 \\ 0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1 \end{matrix}\right].$$

The shortening algorithm of the code can obtain the excellent code with any code parameters. The matrix **H** is shortened by j columns, $j \leq 8$, where the columns of **H** are deleted in the following order:

$$\begin{bmatrix} b_\delta \\ \hline q_v \end{bmatrix}, \begin{bmatrix} b_\sigma \\ \hline q_v \end{bmatrix}, \begin{bmatrix} b_\rho \\ \hline q_v \end{bmatrix}, \begin{bmatrix} b_\alpha \\ \hline q_v \end{bmatrix}, \begin{bmatrix} b_\gamma \\ \hline q_v \end{bmatrix},$$

where q_v is a column of matrix **Q** corresponding to the binary representation of v, and columns b_δ, b_σ, b_ρ, b_α, b_γ are distinct.

Let $R = 7, j = 5, \delta = 3, \sigma = 4, \rho = 5, \alpha = 6$, and $\gamma = 7$. Then the parity check matrix **H** of the (35, 28) code has the following form:

$$\mathbf{H} = \left[\begin{matrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&1&1&1&1&1&0&0&0&0&0&1&1&1&1 \\ 1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&0&0&0&1 \\ 0&1&0&0&1&0&1&0&0&1&0&1&0&0&1&1&0&0&1 \\ 0&0&1&0&1&0&0&1&0&1&0&0&1&0&1&0&1&0&1 \\ 0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&1&1 \end{matrix}\right.$$

$$\left.\begin{matrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \\ 1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&0 \\ 0&0&0&1&0&1&0&1&0&1&0&1&0&1&1&0 \\ 0&1&0&1&0&0&0&1&0&0&1&1&0&0&1&0 \\ 0&0&1&1&0&0&1&1&0&0&0&1&0&0&0&1 \end{matrix}\right].$$

Table I. shows some of the possible Davydov-Tombak codes for $R = 7$ and 8.

TABLE I. Parameters of Davydov-Tombak Codes

$j \leq 8$	$R =$ 7		$R =$ 8	
	N	K	N	K
0	40	33	80	72
1	39	32	79	71
2	38	31	78	70
3	37	30	77	69
4	36	29	76	68
5	35	28	75	67
6	–		74	66
7	–		73	65
8	–		72	64

All matrices obtained have regular structure. Therefore, these matrices are suitable for VLSI implementation. In computer applications, these codes are encoded and decoded in parallel manner. In encoding, the message bits enter the encoding circuit in parallel, and the parity check bits are formed simultaneously [8]. In decoding, the received bits enter the decoding circuit in parallel, the syndrome bits are formed simultaneously, and the received bits are corrected in parallel. Single error correction is accomplished by the table-lookup decoding. This is called hard decision decoding (HDD) or algebraic decoding.

III. DECODING USING A PARITY CHECK TREE

We have observed that Davydov-Tombak code has a very sparse parity check matrix. A matrix is said to be sparse if fewer than half of the elements are nonzero [9]. Associated with a parity check matrix **H** is a graph called the Tanner graph containing two set of nodes [10]. The first set consists of *N* nodes which represent the *N* bits of a codeword; nodes in this set are called *bit* nodes. The second set consists of *M* nodes, called *check* nodes, representing the parity constraints. The graph has an edge between j^{th} bit node and the i^{th} check node if and only if j^{th} bit is involved in the i^{th} check, that is, if $\mathbf{H}_{ij} = 1$. Thus the Tanner graph is a graphical depiction of the parity check matrix. Fig. 1 illustrates the graph for (35, 28) Davydov-Tombak code.

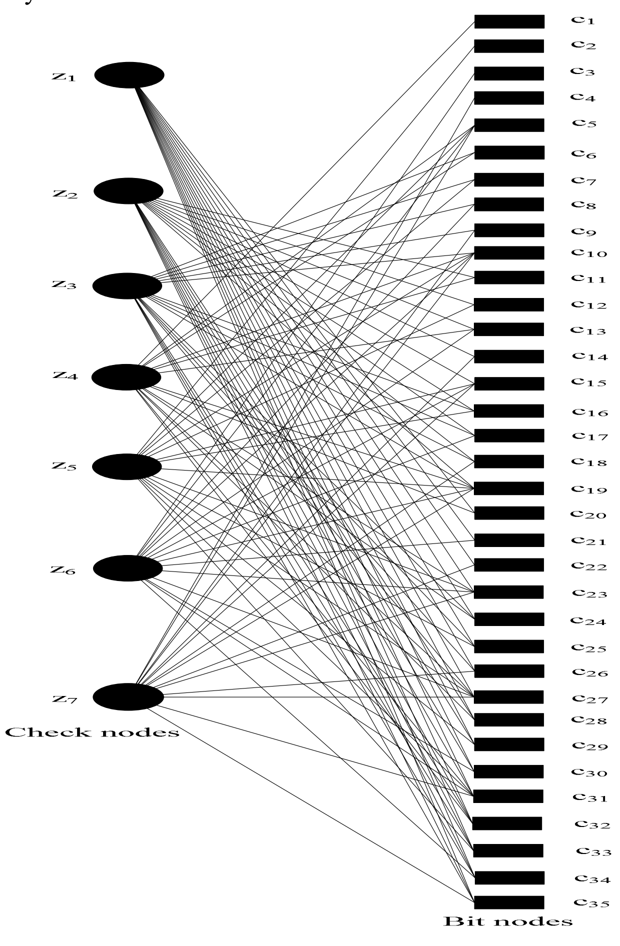

Figure 1. Tanner graph of a (35, 28) Davydov-Tombak code

A graph such as this, consisting of two distinct sets of nodes and having edges only between the nodes in different sets, is called a bipartite graph. The Tanner graph is used to develop insight into the decoding algorithm. The soft decision decoding (SDD) algorithm to decode Davydov-Tombak codes using a parity check tree associated with the Tanner graph is explained below.

For each code bit c_n, compute the checks for those checks that are influenced by c_n. To do this, we propagate probabilities through the Tanner graph, there by accumulating the evidence that the checks provide about the bits. Suppose that c_n is in error and that other bits influencing its checks are also in error. Arrange the Tanner graph with c_n as a root. In Fig. 2, suppose the bits in the star mark are in error. The bits that connect to the checks connected to the root node are said to be in level 1. The bits that connect to the checks from the first level are said to be in level 2. We can establish many such levels. Then, decode by proceeding from the leaves of the tree (right most part of the figure). By the time decoding on c_n is reached, other erroneous bits may have been corrected. Thus bits and checks which are not directly connected to c_n still influence c_n [11].

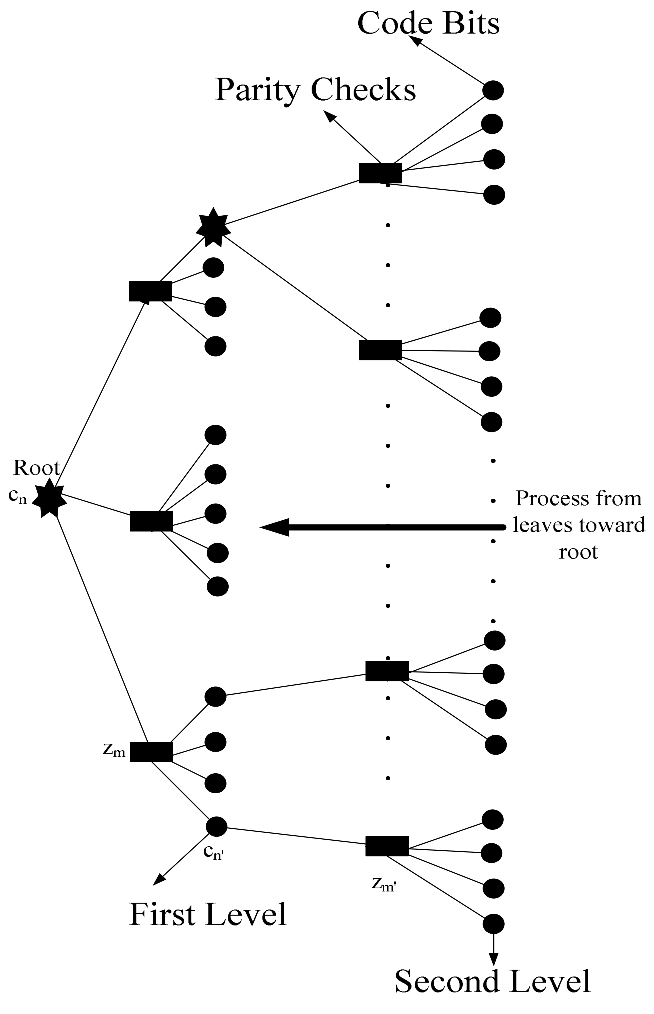

Figure 2. Two level parity check tree associated with the Tanner graph

In this section, mathematical description of a soft decision decoding algorithm for Davydov-Tombak codes by traversing through parity check tree is presented. The following notation is convenient in describing the algorithm [12]. Let $h_{i,j}$ denote the entry of **H** in the i^{th} row and j^{th} column. Let

$$L(m) = \{l : h_{m,l} = 1\} \quad (1)$$

denote the set of code positions that participate in the m^{th} parity check equation, and let

$$M(l) = \{m : h_{m,l} = 1\} \quad (2)$$

denote the set of check positions in which code position l participates. The algorithm iteratively computes two types of conditional probabilities:

q^x_{ml} , the probability that the l^{th} bit of codeword **c** has the value x, given the information obtained via the check nodes other than check node m.

r^x_{ml} , the probability that a check node m is satisfied when bit l is fixed to a value x and the other bits are independent with probabilities $q^x_{ml'}, l' \in L(m) \setminus l$.

In the following, BPSK transmission over an AWGN channel is assumed. Modulated symbols $m(c_i) = (-1)^{c_i}\sqrt{E_s}$ are transmitted over an AWGN channel and received as $r_i = m(c_i) + w_i$, where w_i is a Gaussian distributed random variable with zero mean and variance $N_0 / 2$, $1 \le i \le N$.

Initialization:

For $l \in \{1, 2, 3, \ldots, N\}$, initialize the a priori probabilities of the code nodes

$$p_l^1 = \frac{1}{1 + \exp(r_l \frac{4}{N_0})} \quad (3)$$

and $p_l^0 = 1 - p_l^1$. For every (l, m) such that $h_{m,l} = 1$,

$$q^0_{m,l} = p_l^0 ; \; q^1_{m,l} = p_l^1. \quad (4)$$

Horizontal Step: Updating r^x_{ml}

For each l, m compute

$$\delta r_{m,l} = \prod_{l' \in L(m)\setminus l} (q^0_{m,l'} - q^1_{m,l'}), \quad (5)$$

and

$$r^0_{m,l} = (1 + \delta r_{m,l})/2; \; r^1_{m,l} = (1 - \delta r_{m,l})/2 \quad (6)$$

Vertical Step: Updating q^x_{ml}

For each l, m compute

$$q^0_{m,l} = p^0_l \prod_{m' \in M(l)\setminus m} r^0_{m',l} ; \; q^1_{m,l} = p^1_l \prod_{m' \in M(l)\setminus m} r^1_{m',l} \quad (7)$$

and normalize, with $\alpha = \frac{1}{(q^0_{m,l} + q^1_{m,l})}$,

$$q^0_{m,l} = \alpha q^0_{m,l} ; \quad q^1_{m,l} = \alpha q^1_{m,l} . \quad (8)$$

For each l, compute the a posteriori probabilities

$$q^0_l = p^0_l \prod_{m \in M(l)} r^0_{m,l} ; \; q^1_l = p^1_l \prod_{m \in M(l)} r^1_{m,l} , \quad (9)$$

and normalize, with $\alpha = \frac{1}{(q^0_l + q^1_l)}$,

$$q^0_l = \alpha q^0_l ; \quad q^1_l = \alpha q^1_l . \quad (10)$$

Final Step:

Make a tentative decision: Set, $\hat{c}_n = 1$, if $q^0_l > 0.5$,

else set $\hat{c}_n = 0$. If $H\hat{\mathbf{c}} = 0$, *Stop*.

Since the vast majority of the errors in the byte organized semiconductor memory systems are independent random bit errors, which may be caused by α − particles, cell failures, or external noises, which is equivalent to the AWGN channel model. We have simulated (40, 33), (37, 30), (35, 28) and (72, 64) Davydov-Tombak codes under AWGN channel conditions. Soft decision decoding using a parity check tree is compared with contemporary hard decision decoding. Results show almost 1.6 to 2dB improvement at a bit error rate (BER) of 10^{-6}. This gain is very significant and tells that although the minimum distance of the code is 4, the code was able to decode beyond the minimum distance and correct more than one error.

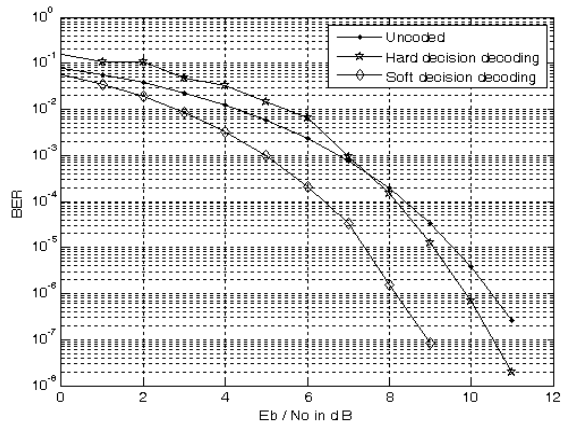

Figure 3. Performance for the (40, 33) Davydov-Tombak code over AWGN with HDD and SDD

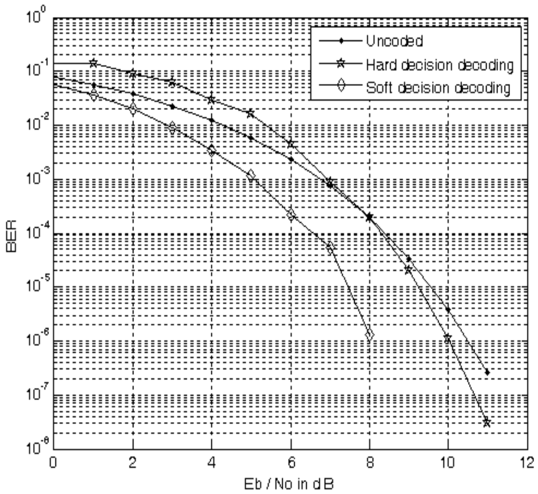

Figure 4. Performance for the (37, 30) Davydov-Tombak code over AWGN with HDD and SDD

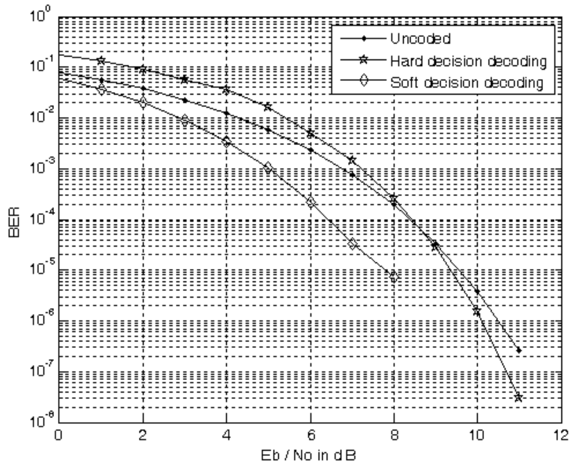

Figure 5. Performance for the (35, 28) Davydov-Tombak code over AWGN with HDD and SDD

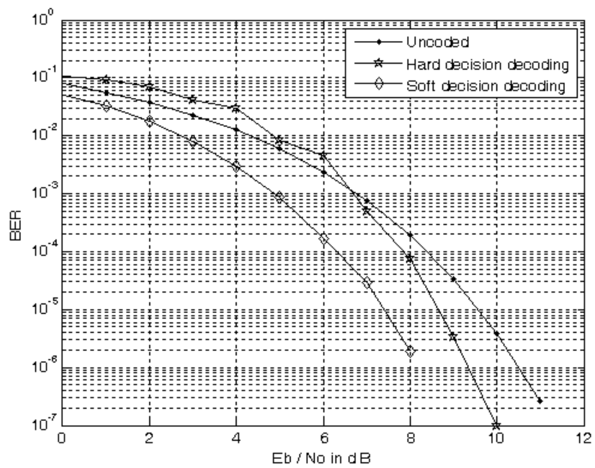

Figure 6. Performance for the (72, 64) Davydov-Tombak code over AWGN with HDD and SDD

IV. CONCLUSION

A moderately simple and efficient soft decision decoding scheme for Davydov-Tombak codes which employs parity check tree representation has been proposed. It is noteworthy that soft decoder always detect the occurrence of decoding errors. The decoding complexity seems to be on the higher side. As technology progresses, power consumption of a decoder can be made very low. However, it is well known, the use of soft decision decoding on channels perturbed by Gaussian noise improves the performance by 1.5 to 3dB over the use of hard decision decoding. Thus, a typical application for the soft decision decoding algorithm of Davydov-Tombak codes could be in mass memories and data entry systems. Multiple hard and soft error problems can be solved by employing this decoding method. In wireless applications, 3dB coding gain over hard decision decoding means that increase data throughput by a factor of 2 or increase range by 40% or reduce transmitter power by a factor of 2. Which translates into smaller transmit antennas or, alternatively, smaller receive antennas for the same transmission power.

REFERENCES

[1] Daniel Costello, David Forney, "Channel Coding: The road to channel capacity", Proceedings of the IEEE | Vol 95, No. 6, June 2007.

[2] J. Yamada, "Selector line merged built in ECC technique for DRAMs", IEEE Transaction on Solid State Circuits, pp. 868-873, October 1987.

[3] L.W. Massengill, "Cosmic and terrestrial single-event radiation effects in dynamic random access memories", IEEE Transaction on Nuclear Science, pp. 576-593, April 1996.

[4] P. Calvel, P. Lamothe, and C. Barillot, "Space radiation evaluation of 16 Mbit DRAMs for mass memory applications", IEEE Transaction on Nuclear Science, Vol. 41, pp. 2267-2271, December 1994.

[5] J.F. Ziegler, M.E. Nelson, J.D. Shell, R.J. Peterson, C.J. Gelderloos, H.P. Mohlfeld, and C.J. Montrose, "Cosmic ray soft error rates of 16-Mb DRAM memory chips", IEEE Journal on Solid State Circuits, pp. 246-252, February 1998.

[6] J.C.Lo and E.Fujiwara,"Transient behavior of the encoding/decoding circuits of error control codes", Proc. IEEE Int. Symposium on Defect and Fault Tolerance in VLSI Systems, October 2005.

[7] Alexander A. Davydov and Leonid M. Tombak, "An alternative to the class of SEC-DED codes in semiconductor memory", IEEE Transactions on Information Theory, Vol. 37, N0. 3, May 1991.

[8] Eiji Fujiwara, Code Design for Dependable Systems: Theory and Practical Applications, John Wiley & Sons, Ltd., 2006.

[9] D.J.C. Mackay, "Good error-correcting codes based on very sparse matrices", IEEE Transaction on Information Theory, Vol. 45, pp. 399-431, March 1999.

[10] Shu Lin and Daniel Costello, Error Control Coding, Second Edition, Pearson Education, Inc., 2004.

[11] Todd K. Moon, Error Correction Coding: Mathematical Methods and Algorithms, Wiley Interscience, 2006.

[12] Robert H. Morelos-Zaragoza, The Art of Error Correcting Coding, Second Edition, John Wiley & Sons, Ltd., 2006.