IETE Journal of Research

ISSN: 0377-2063 (Print) 0974-780X (Online) Journal homepage: https://www.tandfonline.com/loi/tijr20

Enhancing the Error-correcting Capability of Imai-Kamiyanagi Codes for Data Storage Systems by Adopting Iterative Decoding using a Parity Check Tree

H. Prashantha Kumar, U. Sripati, K. Rajesh Shetty & B. Shankarananda

To cite this article: H. Prashantha Kumar, U. Sripati, K. Rajesh Shetty & B. Shankarananda (2012) Enhancing the Error-correcting Capability of Imai-Kamiyanagi Codes for Data Storage Systems by Adopting Iterative Decoding using a Parity Check Tree, IETE Journal of Research, 58:4, 272-278, DOI: 10.4103/0377-2063.102305

To link to this article: https://doi.org/10.4103/0377-2063.102305

Enhancing the Error-correcting Capability of Imai-Kamiyanagi Codes for Data Storage Systems by Adopting Iterative Decoding using a Parity Check Tree

H. Prashantha Kumar, U. Sripati, K. Rajesh Shetty and B. Shankarananda[1]

Department of Electronics and Communication Engineering, National Institute of Technology Karnataka, [1] Department of Electronics and Communication, Gopalan College of Engineering and Management, Bangalore, Karnataka, India

ABSTRACT

A novel low-complexity, soft decision technique which allows the decoding of distance–5 double error-correcting Imai-Kamiyanagi codes by using a parity check tree associated with the Tanner graph is proposed. These codes have been applied to memory subsystems and digital storage devices in order to achieve efficient and reliable data processing and storage. For the AWGN channel, gains in excess of 1.5 dB at reasonable bit error rates with respect to conventional hard decision decoding are demonstrated for the (46, 32), (81, 64), and (148, 128) shortened Imai-Kamiyanagi codes.

Keywords:
Imai-Kamiyanagi codes, Iterative decoding, Memory architecture, Parity check tree, Tanner graph.

1. INTRODUCTION

Channel coding is commonly used to achieve reliable transmission of information. Error control codes enable a decoder to recover from errors produced by noise in a communication channel/storage device. Codes ensure higher noise tolerance at the receiver by adding redundancy into the user data to achieve better separation of data sequences. Error control coding (ECC) algorithms have contributed enormously to the communication revolution. It is the development of ECC algorithms and techniques have made the internet and digital recording ubiquitous facilities which can be enjoyed by all people. In addition, they have also contributed significantly to deep space exploration and the explosion in human knowledge about the various planets and satellites in the solar system. The last fifteen years have been characterized by research work which has enabled exceptional increase in data transmission rates and storage capabilities. Increased use of cellular wireless communication in an environment plagued by spectrum shortage has resulted in the development of several innovative and novel ideas to secure information integrity and enhance channel capacity. There has also been rapid development in microelectronics. Both of these developments often going hand in hand has provided us with both a need for and the possibility of implementing sophisticated algorithms for error control [1].

The advent of the internet age has produced enormous demand for increases in storage capacity and storage density. Physical/Media improvements along with sophisticated signal processing and coding techniques have played a critical role in the constant augmentation of storage/communication channel capacities. Every computer memory and data storage systems have adopted some types of error-detecting or error-correcting codes in order to enhance system reliability. 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 [2]. Permanent and temporal faults are the major sources of errors in modern digital storage systems. Power supply break down, 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 radioactive 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 [3-6].

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. Thus, the use of error control algorithms can protect the integrity of user information against errors caused by aging, wear out due to repeated read and write operations, electromagnetic waves, and manufacturing defects. Many different error control codes have been studied and developed to correct and/or detect the types of errors mentioned above. 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 [7,8].

2. CONSTRUCTION OF IMAI-KAMIYANAGI 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 $\mathbf{C}$ is the smallest Hamming distance between any two codewords in the code. Imai and Kamiyanagi added an extended structure to the double error-correcting (DEC) BCH codes. The Imai-Kamiyanagi code has $d_{\min}=5$. Hence, the hard decision decoding (HDD) of these codes correct any two bit random errors in a block of size N. Although the Imai-Kamiyanagi code has slightly larger check-bit length, the parity check matrix of the code is very sparse (i.e., a small portion of the entries being one, all others being zero). This is why the decoding hardware has less complexity and decoding can be made faster than DEC-BCH code [8]. BCH codes have a minimum or smaller number of check bits, but have complex decoding hardware and a longer decoding time. Today's high-speed computer systems have adopted the information-bit lengths $K=32$, 64, or 128 bits. In Table 1, we have given a comparison of pertinent parameters of shortened BCH and Imai-Kamiyanagi codes for data blocks of size 32, 64, and 128 bits.

The parity check matrix H of general Imai-Kamiyanagi code is given by [8],

$$\mathrm{H}=\begin{bmatrix}\mathrm{H}_1'\\ \mathrm{H}_2'\\ \mathrm{H}_3'\end{bmatrix}. \tag{1}$$

where, H_1', H_2' and H_3' are defined as

$$\mathrm{H}_1'=\left[\mathrm{A}\otimes\mathrm{H}_1 \;\; 0_{m\times r_1}\right],\quad \mathrm{H}_2'=\left[\mathrm{B}\otimes 1_n \;\; \mathrm{I}_{r_1}\right],$$
$$\mathrm{H}_3'=\left[1_{n_0}\otimes\mathrm{H}_3 \;\; 0_{m\times r_1}\right]. \tag{2}$$

Here, symbol $\otimes$ depicts the Kronecker product. The meaning of other notations is defined here:
1_n: all-1 n-tuple row vector, and $n=2^m-1$, $m\geq 2$
1_{n_0}: all -1 n_0-tuple row vector,
0_n: all-0 n-tuple row vector,
I_{r_1}: $r_1\times r_1$ identity matrix,
$0_{m\times r_1}$: $m\times r_1$ zero matrix,
A: $r_0\times n_0$ matrix defined over Galois Field F_2. The rightmost r_0 column vectors are linearly independent.
B: $r_1\times n_0$ matrix over F_2. This has r_1 linearly independent column vectors.

Table 1: Comparison of shortened BCH and Imai-Kamiyanagi code parameters

Codes	Code length, N	Data length, K	Check bits, $N-K$	$d_{\min}$
	44	32	12	
BCH	78	64	14	5
(Shortened)	144	128	16	
	46	32	14	
Imai -Kamiyanagi	81	64	17	5
(Shortened)	148	128	20	

$H_1 = \begin{bmatrix} 1 & \alpha & \alpha^2 & ..... & \alpha^{n-1} \end{bmatrix}$,
$H_3 = \begin{bmatrix} 1 & \alpha^3 & \alpha^6 & ..... & \alpha^{3(n-1)} \end{bmatrix}$. Here, α be a primitive element in F_{2^m}.

$N = n_0 n + r_1$,
$K = n_0 n - m(r_0 + 1)$,
$R = N - K = m(r_0 + 1) + r_1$.

Example 1: Design of (47, 33) Imai-Kamiyanagi Code

Let $m = 4$, $n = 2^m - 1 = 15$, $n_0 = 3$, $r_0 = 2$, $r_1 = 2$,

$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$, and $B = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

Hence,

$$H_1' = \begin{bmatrix} H_1 & H_1 & 0_{m\times n_1} & 0_{m\times 2} \\ H_1 & 0_{m\times n_1} & H_1 & 0_{m\times 2} \end{bmatrix}, H_2' = \begin{bmatrix} 0_n & 1_n & 0_n & 1 & 0 \\ 0_n & 0_n & 1_n & 0 & 1 \end{bmatrix}, \text{and}$$
$$H_3' = \begin{bmatrix} H_3 & H_3 & H_3 & 0_{m\times 2} \end{bmatrix}.$$

Then, from (1) and (2)

$$H = \begin{bmatrix} H_1 & H_1 & 0_{m\times n} & 0_{m\times 2} \\ H_1 & 0_{m\times n} & H_1 & 0_{m\times 2} \\ 0_n & 1_n & 0_n & 10 \\ 0_n & 0_n & 1_n & 01 \\ H_3 & H_3 & H_3 & 0_{m\times 2} \end{bmatrix}. \quad (3)$$

So, code length $N = n_0 n + r_1 = 47$ and information bit length $K = n_0 n - m(r_0 + 1) = 33$. Hence (47, 33), linear Imai-Kamiyanagi DEC code is obtained.

The parity check matrix of the above code with $d_{\min} = 5$, and $R = 14$ is calculated as

In this parity check matrix, total number of ones is 250 and number of zeros is 408. Thus, the H matrix may be classified as being sparse. In the same way, we can derive parity check matrix for (95, 78) and (191, 171) Imai-Kamiyanagi codes defined over F_{2^5} and F_{2^6}, respectively. All parity check matrices obtained have full rank (the rows are linearly independent). Once a parity check matrix H is constructed, Gaussian elimination and re-ordering of columns transform the original H into a matrix in systematic form, from which the generator matrix can be easily constructed. Deleting suitable rows and columns of generator matrix G yields shortened (46, 32), (81, 64), and (148, 128) Imai-Kamiyanagi codes.

In data storage systems, data are processed in parallel streams. In encoding, the message bits enter the encoder circuit in parallel, and the parity check bits are formed simultaneously [9]. In decoding, the received bits enter the decoder circuit in parallel, the syndrome bits are formed simultaneously, and the received bits are corrected in parallel. This is called HDD or algebraic decoding.

3. ITERATIVE DECODING USING A PARITY CHECK TREE

We have observed that Imai-Kamiyanagi code has a very sparse parity check matrix. A matrix is said to be sparse if fewer than half of the elements are nonzero [10]. Such codes possess a low-density parity check matrix. Associated with a parity check matrix H is a graph called the Tanner graph containing two set of nodes. 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 $H_{ij} = 1$. Thus, the Tanner graph is a graphical depiction of the parity check matrix [11]. Figure 1 illustrates the graph for (47, 33) Imai-Kamiyanagi code.

$$H = \begin{bmatrix}
1&0&0&0&1&0&0&1&1&0&1&0&1&1&1&1&0&0&0&1&0&0&1&1&0&1&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&1&1&0&1&0&1&1&1&1&0&0&0&1&0&0&1&1&0&1&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&1&1&0&1&0&1&1&1&1&0&0&0&1&0&0&1&1&0&1&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&1&1&0&1&0&1&1&1&1&0&0&0&1&0&0&1&1&0&1&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&1&0&0&1&1&0&1&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&1&1&0&1&0&1&1&1&0&0\\
0&1&0&0&1&1&0&1&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&1&0&1&0&1&1&1&1&0&0&0&0\\
0&0&1&0&0&1&1&0&1&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&1&0&1&0&1&1&1&1&0&0&0&0\\
0&0&0&1&0&0&1&1&0&1&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&1&0&1&0&1&1&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1\\
1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&0&0\\
0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0&1&1&0&0\\
0&0&1&0&1&0&0&1&0&1&0&0&1&0&1&0&0&1&0&1&0&0&1&0&1&0&0&1&0&1&0&0&1&0&1&0&0&1&0&1&0&0&1&0&1&0&0\\
0&1&1&1&1&0&1&1&1&1&0&1&1&1&1&0&1&1&1&1&0&1&1&1&0&1&1&1&1&0&1&1&1&1&0&1&1&1&1&0&1&1&1&1&0&0&0
\end{bmatrix}_{14\times 47}$$

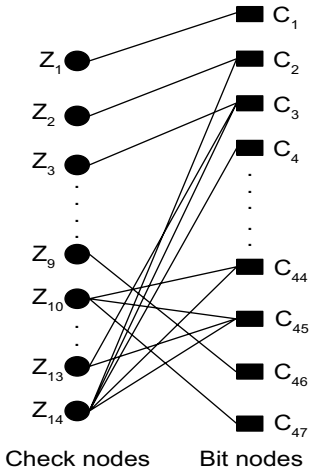

Figure 1: Tanner graph of a (47, 33) Imai-Kamiyanagi code.

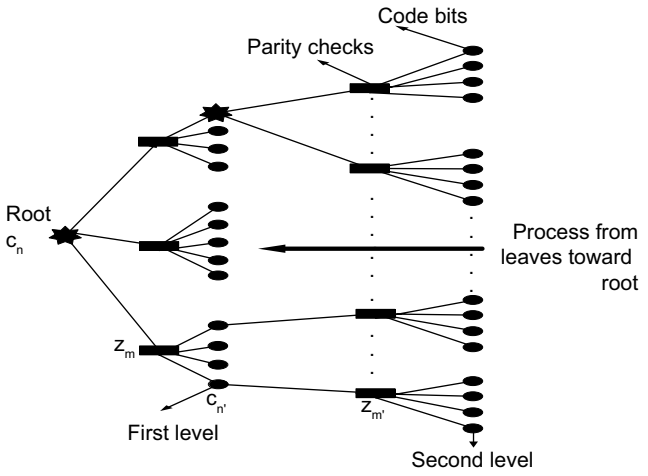

Figure 2: Two-level parity check tree associated with the Tanner graph.

The leitmotiv here is soft decoding of Imai-Kamiyanagi codes described through graphical structures. 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 or Tanner graph. The Tanner graph is used to develop insight into the decoding algorithm. The iterative soft decoding algorithm to decode Imai-Kamiyanagi 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, thereby 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. Let arrange the Tanner graph with c_n as a root shown in Figure 2. This rearrangement of Tanner graph is called a parity check tree. The situation is particularly simple if the graph is a tree, i.e., it is connected, but if any one edge is removed, the graph is no longer connected, in particular a tree has no circuits (A closed path containing at least one edge is called a circuit). Thus, for each edge, we can talk about the sub graph on either side of it. Assume 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 [12].

4. MATHEMATICAL DESCRIPTION OF DECODING ALGORITHM

In this section, mathematical description of iterative decoding algorithm for DEC Imai-Kamiyanagi codes by traversing through parity check tree is presented. The following notation is convenient in describing the algorithm [13-16]. Let $h_{i,j}$ denote the entry of H in the i^{th} row and j^{th} column. Then,

$$L(m) = \{l : h_{m,l} = 1\} \tag{4}$$

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

$$M(l) = \{m : h_{m,l} = 1\} \tag{5}$$

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

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

$r^x_{m,l}$, 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 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 $N_0/2$, $1 \le i \le N$.

Initializaion:

For $l \in \{1, 2, 3, ..., N\}$, initialize the apriori probabilities of the code nodes

$$p^1_l = \frac{1}{1 + \exp(r_l \frac{4}{N_0})} \tag{6}$$

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

$$q_{m,l}^0 = p_l^0; \quad q_{m,l}^1 = p_l^1. \tag{7}$$

Horizontal Step: Updating r_{ml}^x

For each l, m compute

$$\delta r_{m,l} = \prod_{l' \in L(m) \setminus l} (q_{m,l'}^0 - q_{m,l'}^1), \tag{8}$$

and

$$r_{m,l}^0 = (1 + \delta r_{m,l})/2; \quad r_{m,l}^1 = (1 - \delta r_{m,l})/2 \tag{9}$$

Vertical Step: Updating q_{ml}^x

For each l, m compute

$$q_{m,l}^0 = p_l^0 \prod_{m' \in M(l) \setminus m} r_{m',l}^0; q_{m,l}^1 = p_l^1 \prod_{m' \in M(l) \setminus m} r_{m',l}^1 \tag{10}$$

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

$$q_{m,l}^0 = \alpha q_{m,l}^0; \quad q_{m,l}^1 = \alpha q_{m,l}^1. \tag{11}$$

For each l, compute the a posteriori probabilities

$$q_l^0 = p_l^0 \prod_{m \in M(l)} r_{m,l}^0; \quad q_l^1 = p_l^1 \prod_{m \in M(l)} r_{m,l}^1, \tag{12}$$

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

$$q_l^0 = \alpha q_l^0; \quad q_l^1 = \alpha q_l^1. \tag{13}$$

Final Step:

Make a tentative decision: Set $\hat{c}_n = 1$, if $q_l^0 > 0.5$, else set $\hat{c}_n = 0$. If $H\hat{c} = 0$, *Stop*.

The majority of the errors in the byte-organized

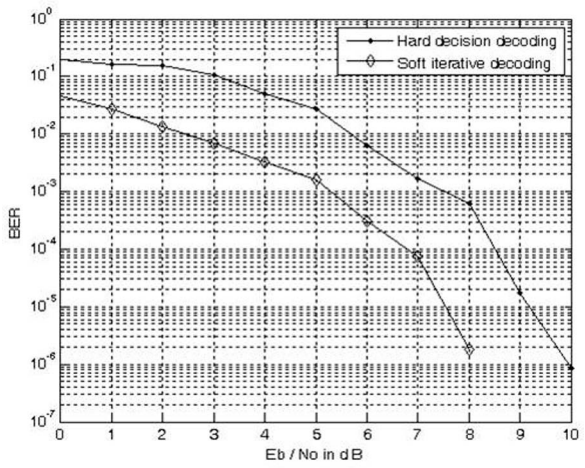

Figure 3: Performance for the (46, 32) Imai-Kamiyanagi code over AWGN with HDD and SDD.

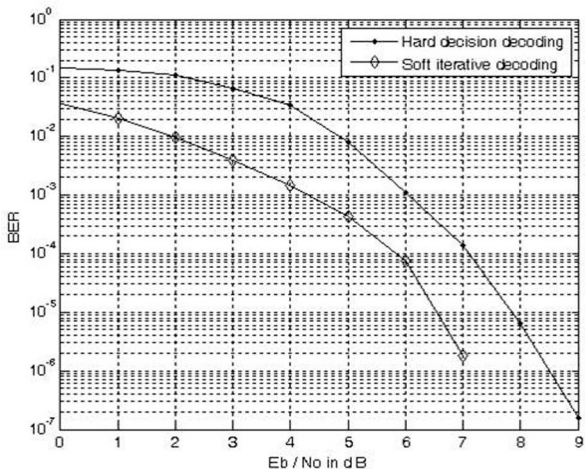

Figure 4: Performance for the (81, 64) Imai-Kamiyanagi code over AWGN with HDD and SDD.

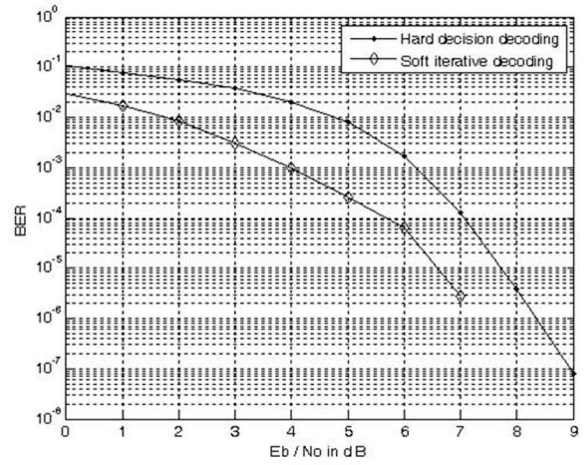

Figure 5: Performance for the (148, 128) Imai-Kamiyanagi code over AWGN with HDD and SDD.

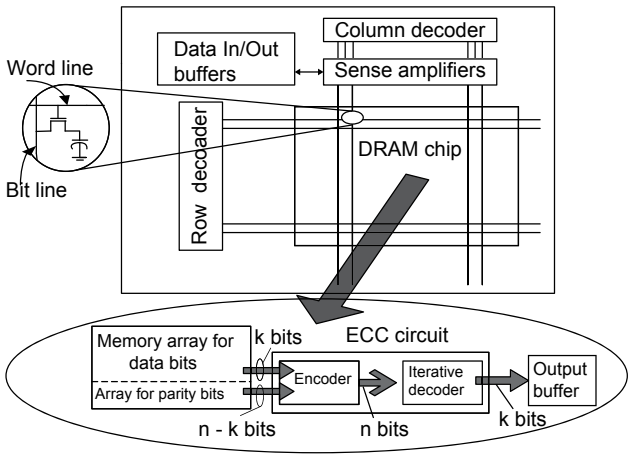

Figure 6: DRAM cell architecture with Imai-Kamiyanagi coder-decoder.

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 (46, 32), (81, 64), and (148, 128)-shortened Imai-Kamiyanagi codes under AWGN channel conditions. Soft iterative decoding using a parity check tree is compared with contemporary HDD. Results [Figures 3-5] show almost 1.5 to 2.0 dB improvement at a bit error rate of 10^{-6}. Code exhibits 3 dB gain in lower SNR region. This gain is very significant and tells that although the error correcting capability of the code is 2 bit, the code was able to decode beyond the minimum distance and correct more than two bit errors.

The latest semiconductor DRAM chips have a multi-bank architecture. Each bank is constructed with highly independent subdivided memory arrays called memory subarrays [17]. The entire output of DRAM chip is called a block and block size may be 32, 64, or 128 bits. Figure 6 shows the architecture of DRAM chip with proposed soft iterative Imai-Kamiyanagi coder-decoder.

5. CONCLUSION

A moderately simple and efficient soft decision-decoding scheme for Imai-Kamiyanagi code which employs parity check tree representation has been proposed. It is noteworthy that soft iterative 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 that the use of soft decision decoding on channels perturbed by Gaussian noise improves the performance by 1.5 to 3 dB over the use of HDD. Thus, a typical application for the soft iterative decoding algorithm of Imai-Kamiyanagi codes could be in mass memories, computer sub memory systems, and data entry systems. Multiple hard and soft error problems can be solved by employing this decoding method. In wireless applications, 3 dB coding gain over un-coded transmission means that data throughput can be increased by a factor of 2 for a fixed SNR. Equivalently, transmitter power can be reduced by a factor of 2 if increase in throughput is not desired. This translates into smaller transmit antennas or, alternatively, smaller receive antennas for the same transmission power. Therefore, collectively we can say that coding gain increases the system performance or reduces cost or both.

REFERENCES

1. D Costello, and D Forney, "Channel Coding: The road to channel capacity", *Proceedings of the IEEE* | Vol 95, pp. 1150-77, June 2007.
2. Z Wu, "Coding and Iterative Detection for Magnetic Recording Channels", 1st Edn, Norwell, Massachusetts: Springer; 2000.
3. J Yamada, "Selector line merged built in ECC technique for DRAMs", *IEEE Transaction on Solid State Circuits,* pp. 868-73, Oct. 1987.
4. L W Massengill, "Cosmic and terrestrial single-event radiation effects in dynamic random access memories", *IEEE Transaction on Nuclear Science,* pp. 576-93, Apr. 1996.
5. 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. 2267-71, Dec. 1994.
6. J F Ziegler, M E Nelson, J D Shell, R J Peterson, C J Gelderloos, and H P Mohlfeld, *et al*, "Cosmic ray soft error rates of 16-Mb DRAM memory chips", *IEEE Journal on Solid State Circuits,* pp. 246-52, Feb. 1998.
7. 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,* Oct. 2005.
8. E Fujiwara, "Code Design for Dependable Systems: Theory and Practical Applications", Hoboken, New Jersey: John Wiley & Sons, Ltd., 2006.
9. B Vasic, and E M Kurtas, "Coding and Signal Processing for Magnetic Recording Systems", 1st Edn, New York, USA: CRC Press; 2004.
10. D J Mackay, "Good error-correcting codes based on very sparse matrices", *IEEE Transaction on Information Theory,* Vol. 45, pp. 399-431, Mar. 1999.
11. W Ryan, and S Lin, "Channel Codes: Classical and Modern", 1st Edn, New York, USA: Cambridge University Press; 2009.
12. T K Moon, "Error Correction Coding: Mathematical Methods and Algorithms", Hoboken, New Jersey: Wiley Interscience; 2006.
13. R H Morelos-Zaragoza, "The Art of Error Correcting Coding", Second Edn, Hoboken, New Jersey: John Wiley & Sons, Ltd; 2006.
14. F R Kschischang, and B J Frey, "Iterative decoding of compound codes by probability propagation in graphical models", *IEEE Journal on Selected Areas in Communications,* 1998.
15. F R Kschischang, B J Frey, and H A Loeliger, "Factor graphs and the sum product algorithm", *IEEE Transactions on Information Theory,* Vol. 47, no. 2, pp. 498-519, Feb. 2001.
16. H Xiao, and A H Banihashemi, "Graph-based message-passing schedules for decoding LDPC codes", *IEEE Transactions on Communications,* Vol. 52, no. 12, pp. 2098-105, Dec. 2004.
17. H Sun, J Liu, R Anigundi, N Zheng, J Q Lu, and K Rose, *et al*, "Design of 3D DRAM and its application in 3D integrated multi-core computing systems", *IEEE Design and Test of Computers,* pp. 36-47, Sept. 2009.

AUTHORS

H. Prashantha Kumar received his B.E. degree in Electronics and Communication Engineering from MCE Hassan in 1998 and received his M.Tech degree in Digital Electronics and Communication from MIT Manipal in 2001. Currently, he is a Ph.D student in the Department of Electronics and Communication Engineering, National Institute of Technology Karnataka, Surathkal, India.

E-mail: hprashanthakumar@gmail.com

U. Sripati received his Ph.D degree from IISC Bangalore in Electrical Communication Engineering Department in 2004. He is currently Associate Professor in Electronics and Communication Engineering at National Institute of Technology Karnataka, Surathkal, India. His major research interests include error control coding, space–time communication, wireless sensor networks, next generation mobile communication, and free space optics.

E-mail: sripati_acharya@yahoo.co.in

K. Rajesh Shetty received his B.E. degree in Electronics and Communication Engineering from NMAMIT Nitte in 1990 and received his M.Tech degree in Digital Electronics and Advanced Communication from NITK Surathkal in 2004. Currently, he is a Ph.D student in the department of Electronics and Communication Engineering, National Institute of Technology Karnataka, Surathkal, India.

E-mail: krshetty_nitte@yahoo.co.in

B. Shankarananda received his Ph.D degree from IIT Bombay in Electrical Engineering Department. He is currently Principal of Gopalan College of Engineering and Management, Bangalore, India. His major research interests include digital communication, optical networks, and error control coding.

E-mail: bsnanda2000@yahoo.co.uk

DOI: 10.4103/0377-2063.102305; Paper No JR 267_11; Copyright © 2012 by the IETE