Collision-Free Interleaver Composed of a Latin Square for Parallel-Architecture Turbo Codes

Dae-Son Kim, Hyun-Young Oh, and Hong-Yeop Song, *Member, IEEE*

***Abstract*— In parallel-architecture turbo codes, the constituent interleavers must avoid memory collision. This paper proposes a collision-free interleaver structure composed of a Latin square (LS) and pre-designed interleavers. Our proposed interleavers can be easily optimized for various information block sizes and for various degrees of parallelism. Their performances were evaluated by computer simulation.**

***Index Terms*— Turbo codes, interleaver, parallel architecture, collision-free, Latin square.**

I. Introduction

THESE days, parallel-architecture turbo codes is one of the hottest topics in the field of channel coding [1]-[5]. In the parallel architecture, a block is divided into several sub-blocks having independent processors, and those sub-blocks can be encoded and decoded simultaneously. No tail bits are required with circular tail-biting encoding. In parallel decoding of turbo codes, each processor is a soft-in soft-out (SISO) module. If more than one processor tries to access the same memory bank to read data symbols according to the constituent interleaver, a collision occurs, and so access cannot be accomplished on time and, in turn, additional delays are incurred [1]. For the purpose of avoiding collisions, many collision free interleavers have been proposed [2]-[4].

The collision-free interleavers proposed in [2], [3] and [4] entail a complex optimizing process. In communication systems, various block sizes generally are required to be supported, and the constituent turbo code interleavers need to be defined for all possible block sizes. This paper proposes a collision-free interleaver structure that can be optimized easily for various information block sizes.

II. Partial Literature Reviews

In this section, we review the three collision-free interleavers proposed in [2], [3], and [4].

A 2D interleaver constituted of two permutations, temporal and spatial permutation, was proposed in [2]. The temporal permutation permutes data symbols in each sub-block, and the spatial permutation permutes data symbols among the sub-blocks. Let the number of data symbols in a block be K, the number of sub-blocks or the degree of parallelism be L, and the number of symbols in each sub-block be M, where $M = K/L$. The symbol index $k \in \{0, 1, \cdots, K-1\}$ can be represented by the 2D array structure with temporal index t and spatial index s, where $k = s \cdot M + t$, $s \in \{0, 1, \cdots, L-1\}$ and $t \in \{0, 1, \cdots, M-1\}$. The temporal permutation is denoted by $\Pi_T(t, s)$ and the spatial permutation by $\Pi_S(t, s)$. Then, the collision-free 2D interleaver is defined as

$$\Pi(k) = \Pi(t, s) = \Pi_S(t, s) \cdot M + \Pi_T(t, s). \quad (1)$$

To avoid collisions in parallel-architecture turbo codes, $\Pi_S(t, s)$ must satisfy the condition that for every $t \in \{0, 1, \cdots, M-1\}$, $\Pi_S(t, s)_{s=0,1,\cdots,L-1}$ is in a one-to-one correspondence with sub-blocks $0, 1, \cdots, L-1$.

The almost regular permutation (ARP) is based on the relative prime interleaver [3]. Periodic fluctuation patterns are added according to

$$\Pi(k) = (P \cdot k + L \cdot (\alpha(k) \cdot P + \beta(k)) + \gamma) \pmod K, \quad (2)$$

where P is relatively prime with K, L is the degree of parallelism, $\alpha(k)$ and $\beta(k)$ are the positive integer sequences of period L for $0 \le k \le K-1$, and γ is an initial offset. Generally $\alpha(k)$ is 0 or 1 and $\beta(k)$ is 0 to 8. The ARP has been used with turbo codes in the standards, including IEEE 802.16, DVB-RCS, and DVB-RCT.

The quadratic permutation polynomial (QPP) interleaver is based on an algebraic construction [4]. Takeshita proved that the QPP interleaver is maximum-collision-free, which means that an interleaver is collision-free for all sub-block sizes M dividing the block length K. The QPP interleaver is defined as

$$\Pi(k) = f_1 \cdot k + f_2 \cdot k^2 \pmod K, \quad (3)$$

where f_1 and f_2 are non-negative integers [4].

III. Proposed Interleaver

We define a collision-free interleaver by rewriting the spatial permutation in matrix form as

$$\Pi(k) = \Pi(s \cdot M + t) = u_{ts} \cdot M + \Pi_T(t). \quad (4)$$

Here, the M by L matrix $\mathbf{U} = \{u_{ts}\}$ indicates the mapping among the sub-blocks. To avoid collisions, each row vector of $\mathbf{U}$ must be the permutation of sub-blocks, $0, 1, \cdots, L-1$. We use a pre-structured interleaver as the temporal permutation $\Pi_T(t)$. The optimizing process determines the mapping matrix $\mathbf{U}$, that is, finds M permutations of $\{0, 1, \cdots, L-1\}$.

Spreading the patterns of the symbols in the same sub-block before and after permutations, affects the performance. If two symbols in the same sub-block remain in the same sub-block after permutations, those symbols can form a cycle making a low-weight codeword or restricting the propagation

Manuscript received August 27, 2007. The associate editor coordinating the review of this letter and approving it for publication was D. Tuninetti. Support for this work was provided in part by Samsung Electronics under the project on 4G wireless communication systems.

D.-S. Kim and H.-Y. Song are with the Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea.

H.-Y. Oh is with Samsung Electronics Co. Suwon, Korea (e-mail: ds.kim@yonsei.ac.kr).

Digital Object Identifier 10.1109/LCOMM.2008.071423.

(a) Good Spreading (b) Bad Spreading

Fig. 1. The comparison of two-tuple pattern distribution.

of messages in the iterative decoding process. Thus, we must permute symbols in the same sub-block to different sub-blocks as much as possible. The L by L LS, $\mathbf{u}_L$, is the L by L square matrix over an alphabet of size L, where every row and every column are the permutation of L symbols [7]. We define the $\mathbf{U}$ matrix according to the form of the columnwise repetition of an L by L LS. We call the $\mathbf{u}_L$-structured $\mathbf{U}$ matrix the *Latin square interleaver*, which satisfies the requirement that the distribution of sub-block indices in each column vector is rendered uniform by the repeating feature.

For short- or medium-size blocks, generally, parallelism of 4 degrees can be considered. To reduce optimizing complexity, we simply consider the reduced LS form, that is, the first row of LS fixed by $(0, 1, 2, 3)$ [7]. We need only to investigate 24 cases, not the 576 cases corresponding to all of the possible cases of 4 by 4 LS. Furthermore, we can reduce this number further by selecting only the good cases.

Each column vector of a matrix implies the sub-block permutation pattern of one decoding block, and the distribution of the patterns in each column vector directly influences the performance. So, we consider the criterion that the distribution of consecutive two-tuple patterns of the column vector of $\mathbf{U}$ is observed, for example $(0, y)$ along (circular) columns of $\mathbf{u}_4$ where $y \in \{1, 2, 3\}$. Figure 1(a) shows $(0, 2)$, $(0, 3)$ once and $(0, 1)$ twice, but Fig. 1(b) shows only the patterns $(0, 1)$ and $(0, 3)$ twice. Therefore, the pattern in Fig. 1(a) is expected to give a better performance than in Fig. 1(b), since it is closer to the uniform distribution than is the other.

According to this criterion, we can divide 24 cases into 2 groups, good and bad, where each group has 12 cases. A comparison of the performances of these two groups is illustrated in Fig. 2. We use 3GPP interleavers of size 160 as the temporal permutation. The simulation environment is 3GPP standard turbo codes of information block size 640 with 4 parallelisms. The constituent convolutional codes are given by the generator matrix $[1, (1+D+D^3)/(1+D^2+D^3)]$. The code rate is $1/3$. The decoding algorithm is max log-MAP and the maximum iteration number is 8 with the Genie stopping rule, that is, the iterations are stopped when there are no errors in decoded information bits. The frame error rate (FER) curves of the good group and the bad group are clearly distinguished by the curve of the 3GPP interleaver of size 640 in Fig. 2.

For medium or long block length, parallelism of more than 4 degrees can be considered, for example 8 or 12. But there are at least $2.3 \cdot 10^{10}$ cases of 8 by 8 reduced LS [6]. So, we cannot investigate all of the cases for optimization. Instead, we will extend l by l LS to L by L LS where $L = n \cdot l$, $n \in \{2, 3, 4, \cdots\}$. The extended L by L LS is Eq. (5).

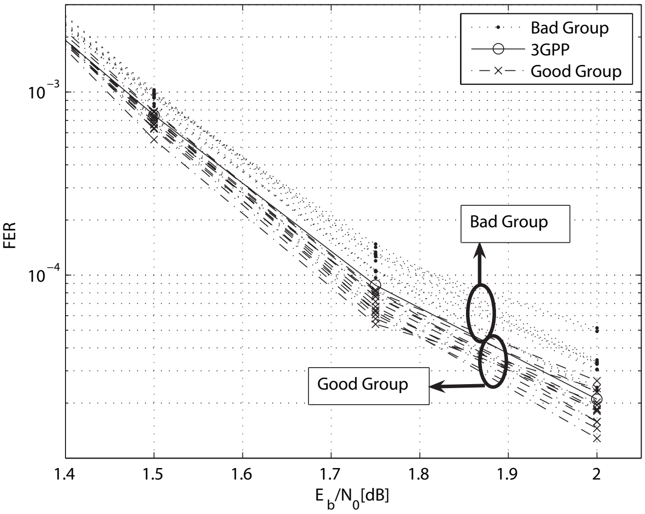

Fig. 2. Comparison of good-gruop and bad-group FER performances.

$$
\begin{aligned}
\mathbf{u}_L = \quad & (\mathbf{u}_l)^{(n)} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1l} \\ a_{21} & a_{22} & \cdots & a_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ a_{l1} & a_{l2} & \cdots & a_{ll} \end{pmatrix}^{(n)} = \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_l \end{pmatrix}^{(n)} \\
= \quad & \begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_1^{(1)} & \cdots & \mathbf{a}_1^{(n-1)} \\ \mathbf{a}_2^{(n-1)} & \mathbf{a}_2 & \cdots & \mathbf{a}_2^{(n-2)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_1^{(1)} & \mathbf{a}_1^{(2)} & \cdots & \mathbf{a}_1 \\ \mathbf{a}_2 & \mathbf{a}_2^{(1)} & \cdots & \mathbf{a}_2^{(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_1^{(n-1)} & \mathbf{a}_1 & \cdots & \mathbf{a}_1^{(n-2)} \\ \mathbf{a}_2^{(n-2)} & \mathbf{a}_2^{(n-1)} & \cdots & \mathbf{a}_2^{(n-3)} \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix}, \qquad (5)
\end{aligned}
$$

where $(\mathbf{u}_l)^{(n)} = \mathbf{u}_{l \cdot n}$, $\mathbf{a}_i = (a_{i1}, a_{i2}, \cdots, a_{il})$, $\mathbf{a}_i^{(k)} = (a_{i1}^{(k)}, a_{i2}^{(k)}, \cdots, a_{il}^{(k)})$, and $a_{i,j}^{(k)} = a_{i,j} + l \cdot k$. The $\mathbf{u}_L$-structured $\mathbf{U}$ is the *extended LS interleaver* from $\mathbf{u}_l$. $\mathbf{u}_8$ can be constructed from $\mathbf{u}_4$ and is also a reduced LS. We investigate only 24 cases of $\mathbf{u}_8$ since there are 24 cases of 4 by 4 reduced LS.

If we do not concern the optimization process, we can add the irregularity to the LS interleaver. We generate M permutations randomly as the row vectors of U with the constraint that σ by L matrix is a LS and $\sigma \leq L$. We call it by the *semi-LS interleaver*.

IV. Simulation Results and Discussion

In this section, we compare the performance of the proposed LS interleaver with the ARP and QPP interleavers. First, for the 320 and 640 block sizes, parallelism of 4 degrees is considered in comparing the LS interleaver with that of the ARP. The ARP for 3GPP2 is proposed in [5]. The interleaver parameters are:

- P=197, L=4, α=(0,0,1,1), β=(0,2,5,3), γ=3 for K=320,
- P=201, L=4, α=(0,0,1,1), β=(0,6,3,1), γ=3 for K=640.

In the proposed interleaver, we use the 3GPP interleaver of sizes 80 and 160, respectively, as a temporal permutation. The

TABLE I

COMPARISON OF COMPLEX OPTIMIZATION PROCESS

Int. type	$K = 320, L = 4$	$K = 640, L = 4$	$K = 1024, L = 8$
LS	12	12	24
ARP	40960	81920	1073741824
QPP	15200	55360	261632

LS used by the LS interleavers is given as

$$\mathbf{u}_{4,1} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 3 & 2 & 0 & 1 \\ 2 & 3 & 1 & 0 \end{pmatrix}, \ \mathbf{u}_{4,2} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \\ 1 & 2 & 3 & 0 \end{pmatrix}. \quad (6)$$

Among the 12 candidates, $\mathbf{u}_{4,1}$ shows the best performance for $K = 320$ and $\mathbf{u}_{4,2}$ for $K = 640$. The other simulation environments are the same as those discussed in Section III. Figure 3 shows the FER versus E_b/N_0. The proposed LS interleaver shows almost the same performance as the ARP for $K = 320$. For $K = 640$, the ARP shows a slightly better performance in the waterfall region but the error floor occurs early. We cannot find a better semi-LS interleaver than the best LS interleaver but they show almost same performance. The proposed LS interleaver shows a good performance for high signal to noise ration (SNR).

For the 1024 block sizes, parallelism of 8 degrees is considered, and we compared the LS interleaver with the QPP interleaver. We also use the 3GPP interleaver of size 128 as a temporal permutation.

$$\mathbf{u}_8 = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 3 & 0 & 1 & 2 \\ 2 & 3 & 0 & 1 \end{pmatrix}^{(2)}. \quad (7)$$

$\mathbf{u}_8$ shows the best performance among the 24 candidates. The QPP interleaver parameters are $f_1 = 31$ and $f_2 = 64$. Figure 4 shows a similar result for the $K = 640$ case. The proposed LS interleaver has a better performance for high SNR, almost without error floor.

Table I shows a comparison of the complexity of the optimization process for the different interleavers. In the ARP, we suppose that α and γ are fixed, that only $\beta(k)$ for $k = 0 \pmod{L}$ is zero, and that the other $\beta(k)$s are from 1 to 8, so there are $|P| \cdot 8^{L-1}$ cases to be investigated [3], where $|P|$ is the cardinality of P. In the QPP interleaver, there are $|f_1| \cdot |f_2|$ cases [4]. Our proposed LS interleaver has a greatly reduced complexity for optimization and shows good performance. Note that one can use any pre-structured interleaver as a temporal interleaver in the design of the proposed interleaver.

V. CONCLUSIONS

We propose a collision-free interleaver for parallel-architecture turbo codes. Using a given pre-structured interleaver, one can make interleavers of various block sizes by defining a mapping matrix $\mathbf{U}$. When LS structure is used as the mapping matrix, the optimizing process is much less complex than that for the ARP and QPP interleavers. In the case of 4 (resp. 8) parallelisms, only 12 (resp. 24) cases are investigated,

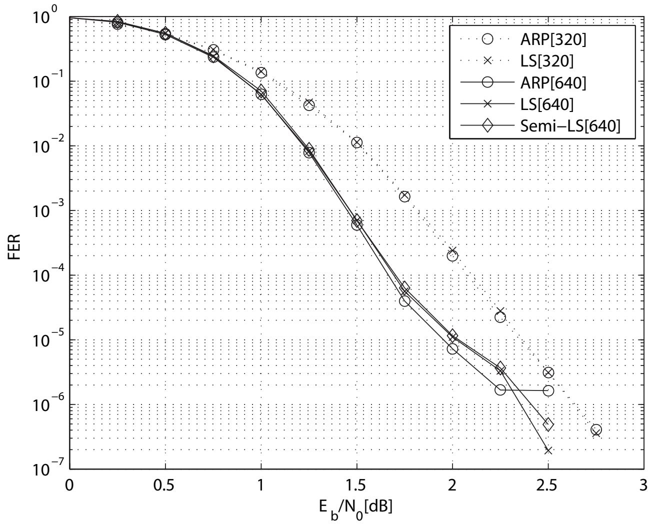

Fig. 3. Comparison of LS and ARP interleavers.

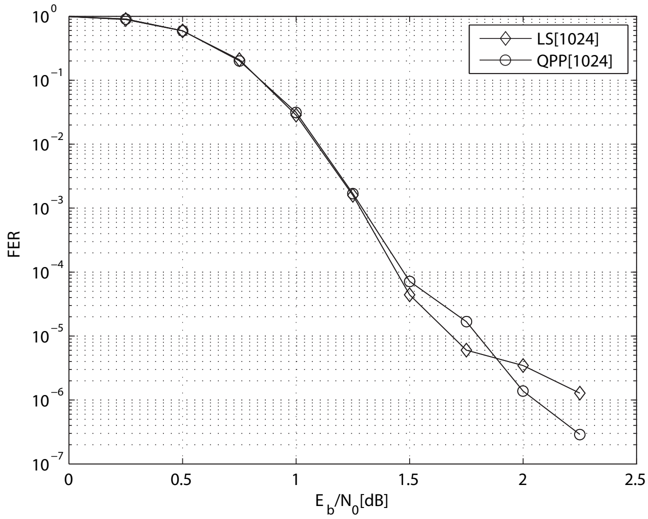

Fig. 4. Comparison of LS and QPP interleavers.

regardless of block size. Moreover, the proposed interleaver shows almost the same performance as those of the ARP and QPP interleavers.

REFERENCES

[1] A. Tarable, S. Benedetto, and G. Montorsi, "Mapping interleaving laws to parallel turbo and LDPC decoder architectures," *IEEE Trans. Inform. Theory*, vol. 50, no. 9, pp. 2002–2009, Sept. 2004.

[2] D. Gnaedig, E. Boutillon, M. Jézéquel, V. Gaudet, and P. Gulak, "On multiple slice turbo codes," in *Proc. 3rd Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp. 343–346, Sept. 2003.

[3] C. Berrou, S. Kerouedan, Y. Saouter, C. Douillard, and M. Jézéquel, "Designing good permutations for turbo codes: towards a single model," in *Proc. Int. Conf. Commun.*, Paris, France, pp. 341–345, June 2004.

[4] O. Y. Takeshita, "On maximum contention-free interleavers and permutation polynomials over integer rings," *IEEE Trans. Inform. Theory*, vol. 52, no. 3, pp. 1249–1253, Mar. 2006.

[5] 3GPP TSG RAN WG1-43, "Enhancement of Rel. 6 Turbo Code," Nov. 2005.

[6] J. M. Hall, *Combinatorial Theory*, $2^n d$ ed. John Wiley & Sons, 1986.

[7] C. J. Colbourn and J. H. Dinitz, *The CRC Handbook of Combinatorial Designs*, $2^n d$ edition. CRC Press, pp. 97-110, 1996.