Collision-free Interleavers using Latin Squares for Parallel Decoding of Turbo Codes

April. 25. 2007

Hyun-Young Oh, **Dae-Son Kim**, Joon-Sung Kim and Hong-Yeop Song

Coding and Information Theory Lab
YONSEI University
Seoul, Korea

Outline

- Motives
- Parallel Archtecture of Turbo Codes
- Collision-free Interleavers
- Proposed Collision-free Interleavers
- Concluding Remarks

Motives

- **Turbo Codes**
  - Outstanding error-correcting capability
  - High decoding latency
  - Parallel architecture
    - Memory *collision-free*
- **Interleavers for parallel architecture**
  - 2D interleaver, ARP (almost regular permutation)

    ➔*Complex* optimizing process

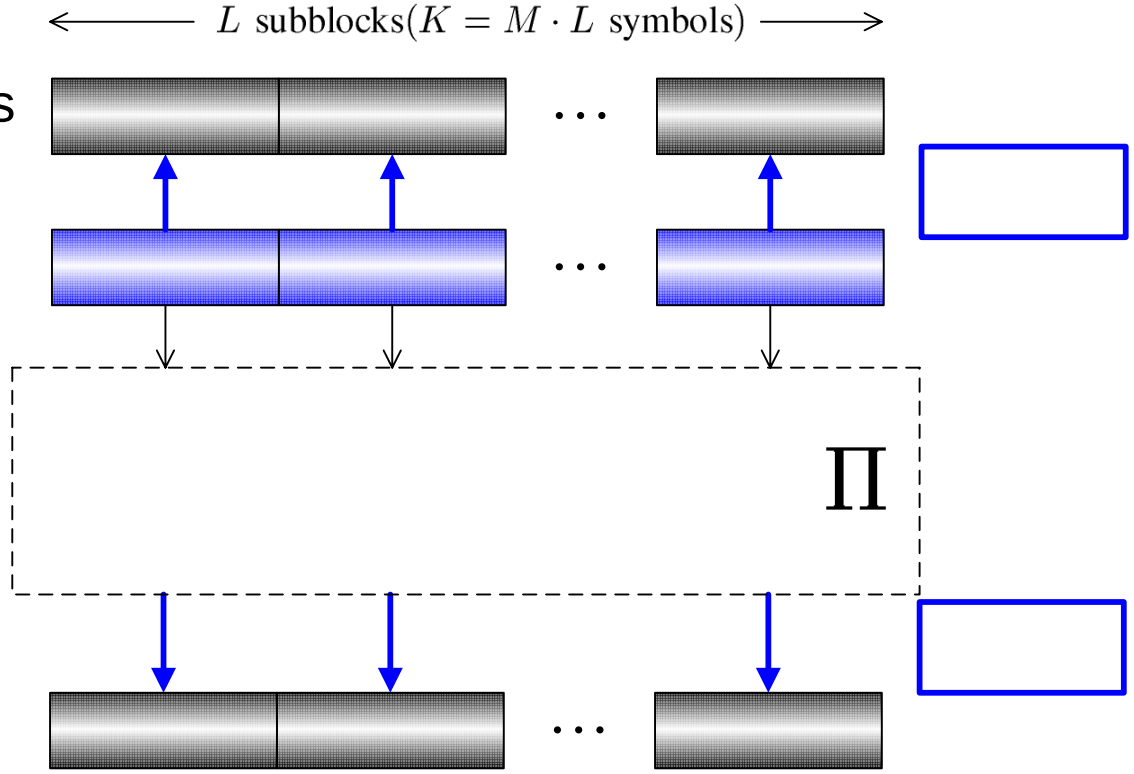
    ➔*Hard* to optimize over various block size

➔Need new collision-free interleaver which can be *easily* optimized over *various* block size

Parallel Architecture of Turbo Codes

Parallel Architecture

- L subblocks
  - → L SISO modules
  - → *L-parallelism*
- Tail-biting encoding
  - → No tail-bits
- Roughly L times reduction of delay

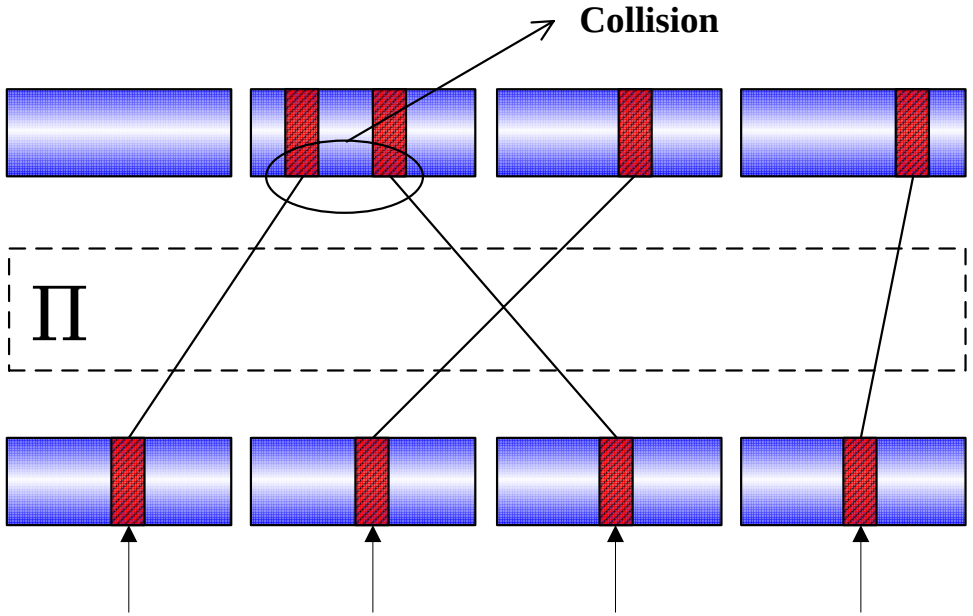


Parallel architecture

Parallel Architecture of Turbo Codes

Collision Problem

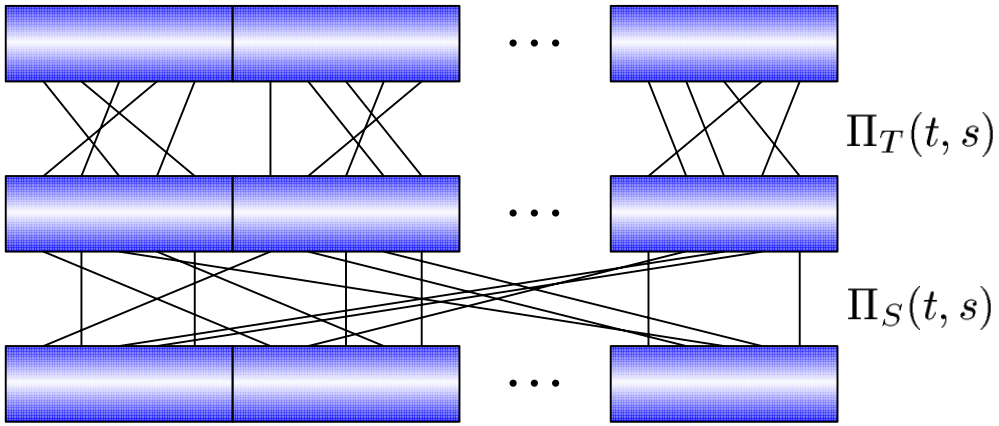
- When more than one module try to access the *same* memory bank → *Additional delay*



Memory collision

Collision-free Interleavers

2D Interleaver



2-dimensional array structure

$$k = s \cdot M + t$$

$$s \in \{0, 1, \cdots, L-1\}$$

$$t \in \{0, 1, \cdots, M-1\}$$

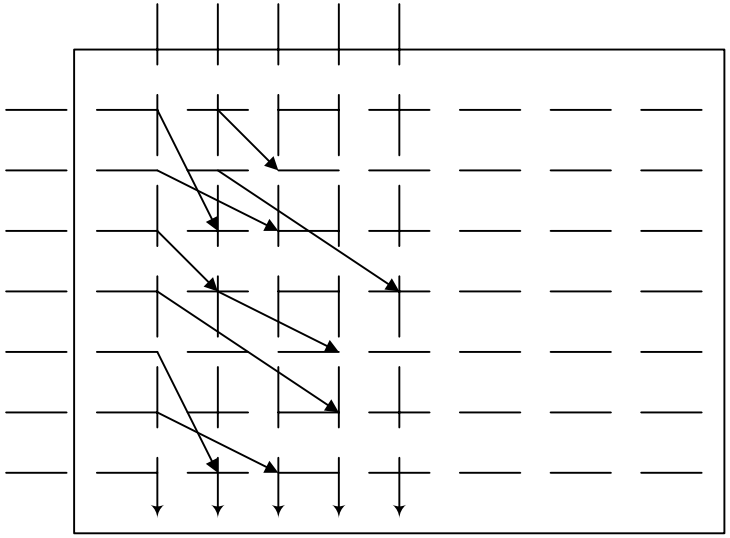
2D interleaver

Information **Information** **Information**

- $\Pi(k) = \Pi(t,s) = \Pi_S(t,s) \cdot M + \Pi_T(t,s)$
- *Easy construction* using any kind of $\Pi_S(t,s)$ and $\Pi_T(t,s)$

Collision-free Interleavers

- **ARP (almost regular permutation)**



Fluctuation patterns

P : Relative prime with K

$\alpha(k),\ \beta(k)$: Integer sequence of period C

γ : Initial offset

writin

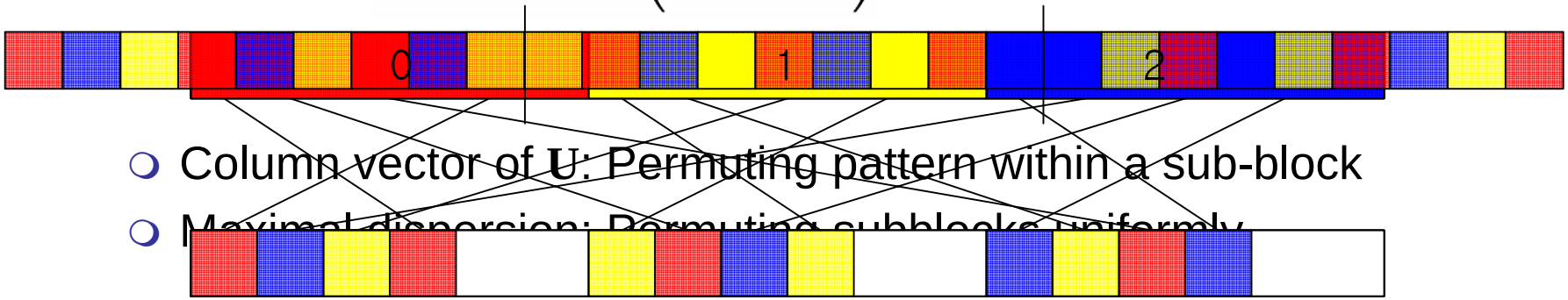
- $\Pi(k) = (P \cdot k + C \cdot (\alpha(k) \cdot P + \beta(k)) + \gamma) \pmod{K}$
- *Complex* optimizing process
- 0.55dB improvement against 3GPP at FER 10^{-5}, block size 640

Proposed Collision-free Interleavers

- **Latin square structure**
  - $\Pi(k) = \Pi(s \cdot M + t) = u_{ts} \cdot M + \Pi_T(t)$ → 2D structure
  - Example) $K = 27,\ L = 3,\ M = 9,\ \Pi_T = (7, 3, 5, 0, 4, 1, 8, 2, 6)$

$$(27 \text{ by } 3)\ \mathbf{U} = \begin{pmatrix} 0 & 1 & 2 \\ 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 1 & 2 \\ \vdots & \vdots & \vdots \end{pmatrix}$$

Column-wise repetition of the Latin square (*Latin square structure*)



  - Column vector of **U**: Permuting pattern within a sub-block
  - Maximal dispersion: Permuting subblocks uniformly

$2 \times 9 + 7 \times 27 + 0 = 0 \quad 1 \times 9 + 3 \times 27 + 9 = 9 \quad 2 \times 9 + 5 \times 27 + 18 = 18$

Proposed Collision-free Interleavers

- **Optimizing process (*4-parallelism*)**
  - $\Pi_T(t)$: 3GPP standard interleaver (defined for *40-5114*)
  - $\mathbf{U} = \{u_{ts}\}$: 576 cases → 24 cases → *12 cases*
    - Let the first row of the Latin square by $(0, 1, 2, 3)$
    - Pick out good candidates among 24 cases

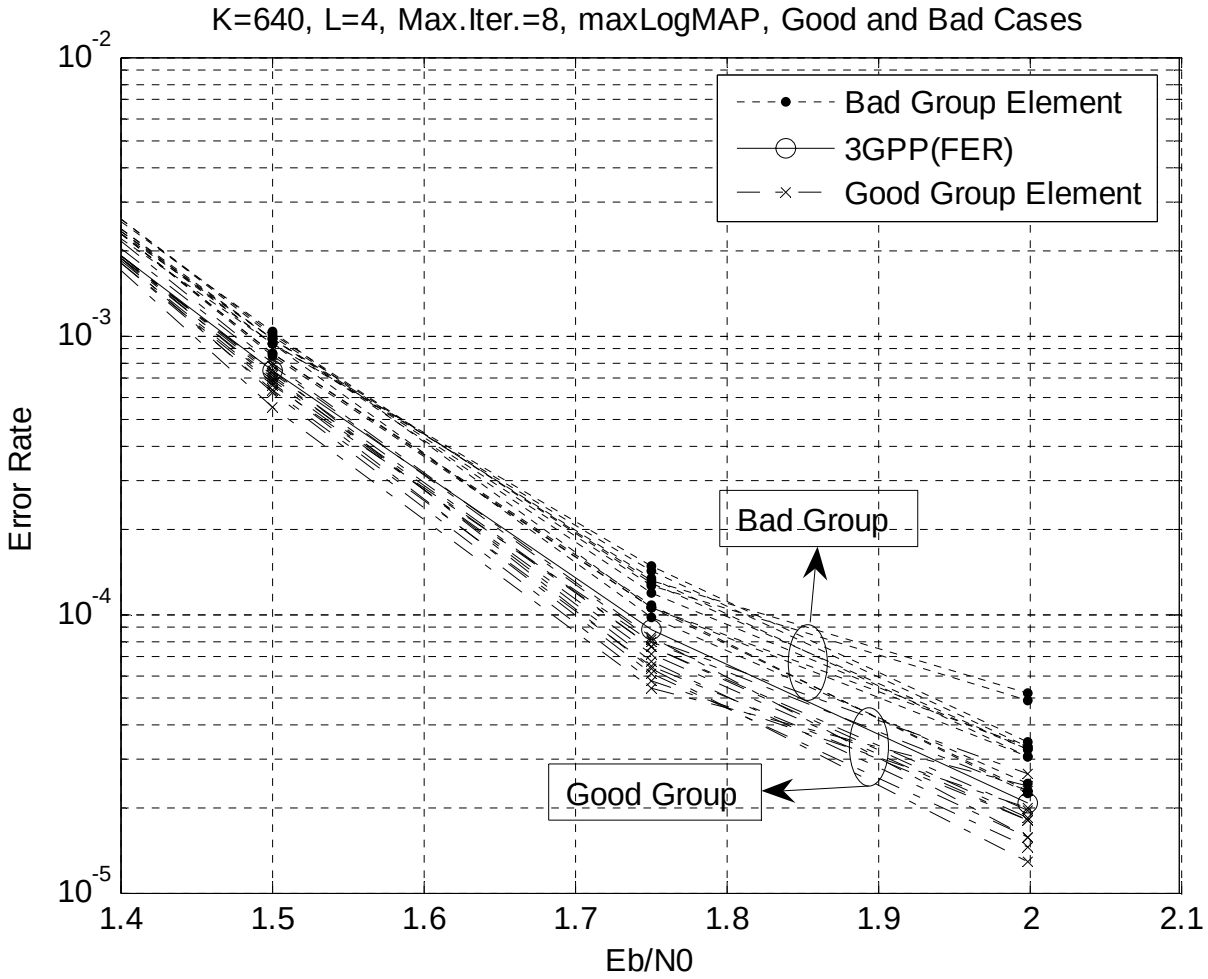
$$\begin{pmatrix} \mathbf{0} & 1 & 2 & \mathbf{3} \\ \mathbf{1} & \mathbf{0} & 3 & 2 \\ 2 & \mathbf{3} & \mathbf{0} & 1 \\ 3 & 2 & \mathbf{1} & \mathbf{0} \end{pmatrix}$$

(a) Bad Spreading

$$\begin{pmatrix} \mathbf{0} & 1 & \mathbf{2} & 3 \\ \mathbf{1} & \mathbf{0} & 3 & 2 \\ 2 & \mathbf{3} & 1 & \mathbf{0} \\ 3 & 2 & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

(b) Good Spreading

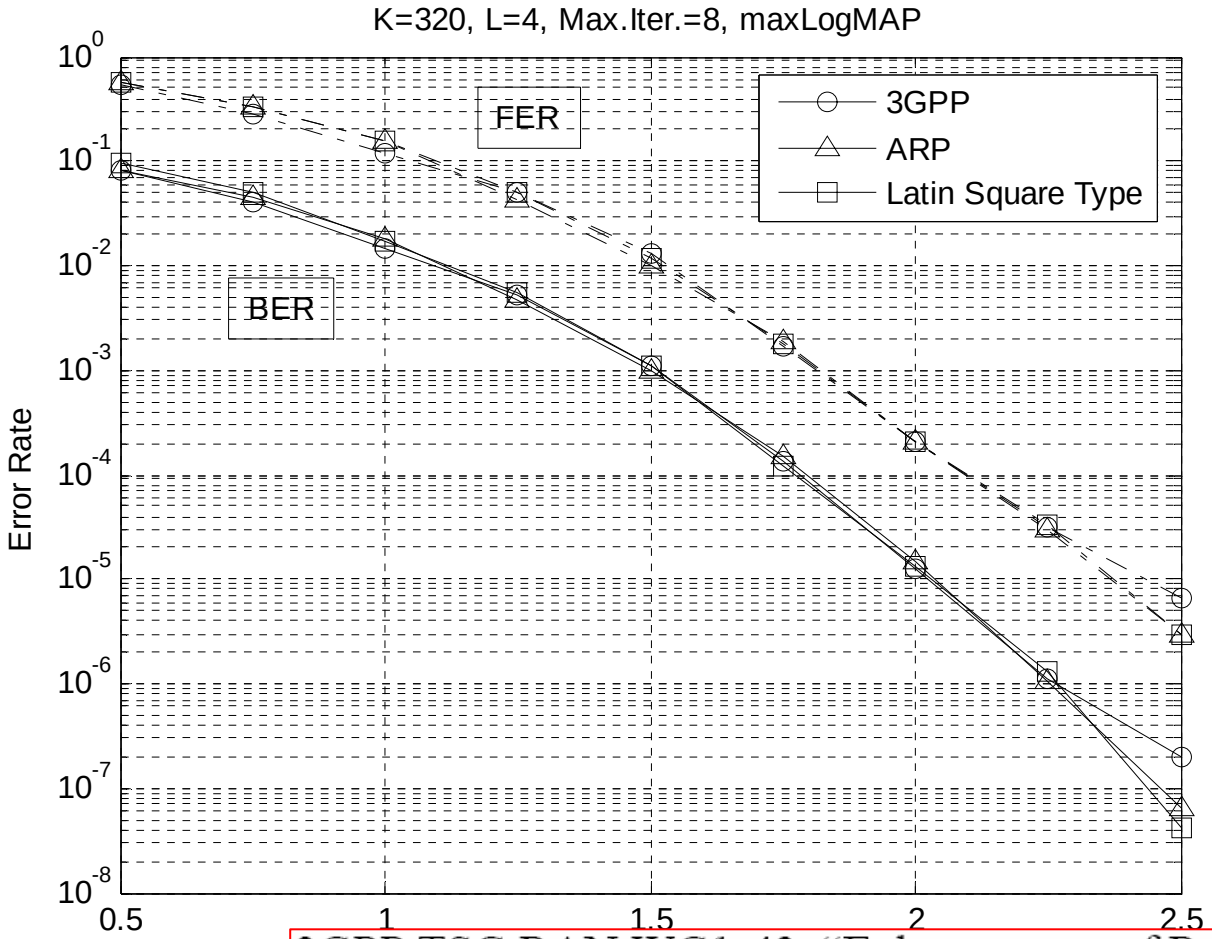
Proposed Collision-free Interleavers



Distingushed definitely

Comparison between good and bad group

Proposed Collision-free Interleavers



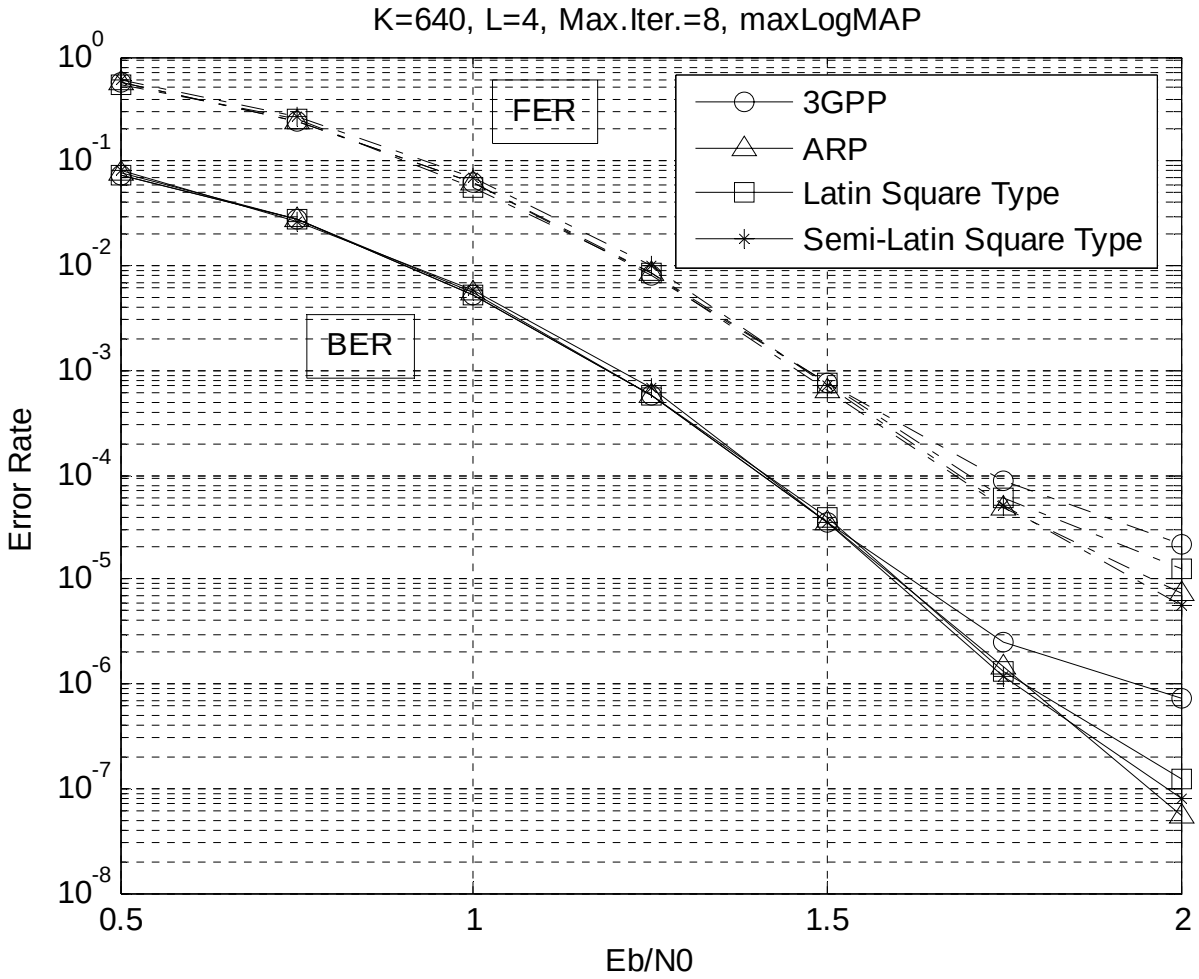
Latin square structure

$$\mathbf{U}_1 = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 3 & 2 & 0 & 1 \\ 2 & 3 & 1 & 0 \end{pmatrix}$$

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

Comparison of BER, FER at block size 320

Proposed Collision-free Interleavers



Comparison of BER, FER at block size 640

Latin square structure

$$\mathbf{U}_2 = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 2 & 3 & 0 & 1 \\ 1 & 2 & 3 & 0 \\ 3 & 0 & 1 & 2 \end{pmatrix}$$

Semi-Latin square structure

$$\mathbf{U}_3 = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 2 & 3 \\ 2 & 3 & 0 & 1 \\ 1 & 2 & 3 & 0 \\ 0 & 1 & 2 & 3 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

Concluding Remarks

- ***Conventional*** **collision-free interleavers**
  → ***complex*** **optimizing process**

- **The proposed collision-free interleavers**
  → ***easy*** **to optimize at** ***various*** **block size**

- **Any kind interleaver can be apply to the parallel architecture of turbo code using Latin square interleaver.**