

Agenda Item:

Source: NTT DoCoMo and Nortel Networks

Title: Extended Turbo code internal interleaver for smaller block size

Document for: Decision

1. Introduction

In R1#9 and RAN#6 meeting, it was agreed that the extended Turbo code internal interleaver for smaller coding block sizes (i.e. the sizes of less than 320 bits) will be included as an UE option in release-99. In this contribution, we propose an optimal algorithm for such smaller size extension based on the specified algorithm i.e. prime interleaver (PIL) algorithm [1]-[5] and its HWA performance evaluation[6] results are also presented in this contribution.

2. Proposed extension scheme

The proposed extended PIL algorithm can generate additional block size range of [1-bit to 319-bit, the exactly same prime interleaver algorithm used in the current 25.212 (25.222) specification, is employed. The existing algorithm and the parameters for the block size from 320-bit to 5114-bit are not changed at all. The number of the mother interleavers is changed from 134 to 165 according to the input block size extension. Additional 4[6] primes [2, 3,] 5, 7, 11 and 13, and its primitive roots are defined. The extended range of the input block size is divided into three sub-blocks and each sub-block has different row number and also different inter-row permutation patterns as shown in Table 1. Here only one additional inter-row permutation pattern i.e. $*P_D = \{4, 3, 2, 1, 0\}$ for the row number $R = 5$ and $*P_D = \{0\}$ for the row number $R=1$ needs to be defined. It can be expected that the above extension increasing implementation complexity.

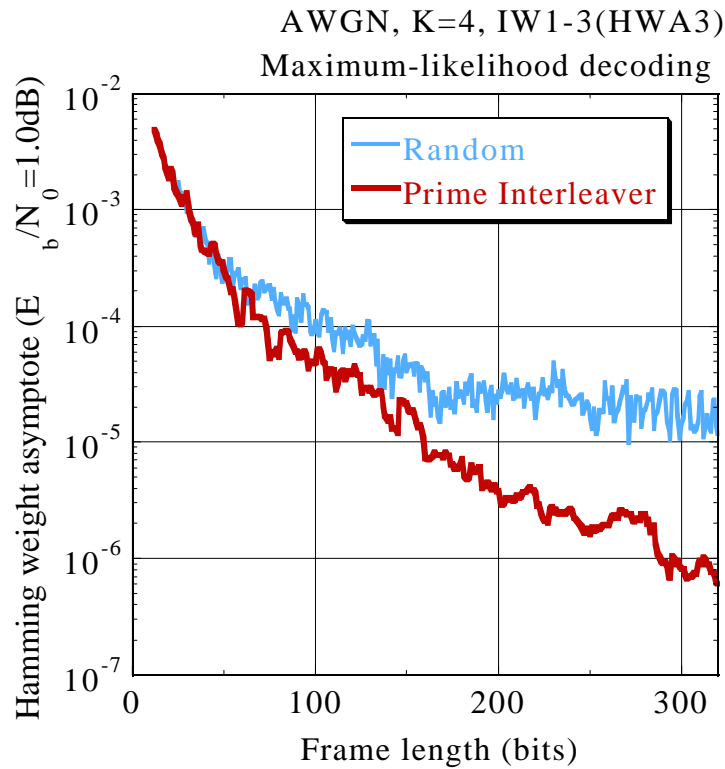
Table 1 Row number and inter-row permutation patterns

Input block size K	Row number R	Inter-row permutation patterns
1 to 24-bit	1	PE*
25 to 159-bit	5	PD*
160 to 200-bit	10	PC
201 320 to 480-bit	20	PA
481 to 530-bit	10	PC
531 to 2280-bit	20	PA
2281 to 2480-bit	20	PB
2481 to 3160-bit	20	PA
3161 to 3210-bit	20	PB
3211 to 5114-bit	20	PA

3. Performance evaluation

Figure 1 shows the result of the HWA performance evaluations[6] for the extended prime interleaver. It can be seen that the extended prime interleaver gives the relative low HWA for the whole bit size range compared to the reference interleaver i.e. the ordinary random interleaver.

Figure 1 HWA performance of the proposed Prime interleaver



4. Revised text proposal for TS 25.212 and 25.222

The revised text for Turbo code internal interleaver description in 25.212 (25.222) is shown below.

===== Beginning of the text =====

4.1.3.2.3 Turbo code internal interleaver

Figure 5 depicts the overall 8 state PCCC Turbo coding scheme including Turbo code internal interleaver. The Turbo code internal interleaver consists of mother interleaver generation and pruning. For arbitrary given block length K, one mother interleaver is selected from the ~~434168~~ mother interleavers set. The generation scheme of mother interleaver is described in section 4.1.3.2.3.1. After the mother interleaver generation, *l*-bits are pruned in order to adjust the mother interleaver to the block length K. The definition of *l* is shown in section 4.1.3.2.3.2.

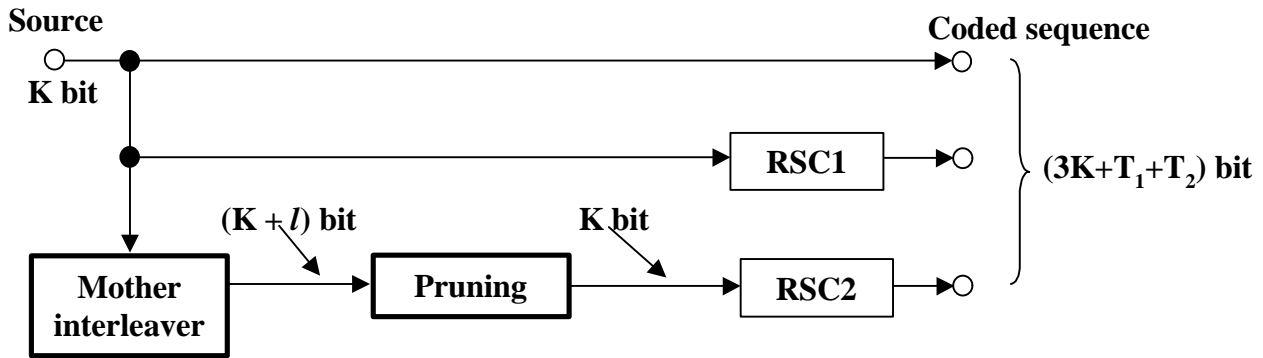


Figure 5: Overall 8 State PCCC Turbo Coding

4.1.3.2.3.1 Mother interleaver generation

The interleaving consists of three stages. In first stage, the input sequence is written into the rectangular matrix row by row. The second stage is intra-row permutation. The third stage is inter-row permutation. The three-stage permutations are described as follows, the input block length is assumed to be K (320-1 to 5114 bits).

First Stage:

(1) Determine a row number R such that

$R = 1$ ($K = 1$ to 24 bits)

$R = 5$ ($K = 25$ to 159 bits)

$R = 10$ ($K = 160$ to 200 bits and 481 to 530 bits; Case-1)

$R = 20$ ($K =$ any other block lengths ~~except 481 to 530 bits; Case-2~~)

(2) Determine a column number C such that

~~if $K = 481$ to 530 Case-1; $C = p = 53$~~

~~else Case-2;~~

(i) find minimum prime p such that,

$$0 \leq (p+1) - K/R,$$

(ii) if $(0 \leq p - K/R)$ then go to (iii),

$$\text{else } C = p+1.$$

(iii) if $(0 \leq p-1 - K/R)$ then $C = p-1$,

$$\text{else } C = p.$$

(3) The input sequence of the interleaver is written into the $R \times C$ rectangular matrix row by row.

Second Stage:

A. If $C = p$

(A-1) Select a primitive root g_0 from table 2.

(A-2) Construct the base sequence $c(i)$ for intra-row permutation as:

$$c(i) = [g_0 \times c(i-1)] \bmod p, i = 1, 2, \dots, (p-2), c(0) = 1.$$

(A-3) Select the minimum prime integer set $\{q_j\}$ ($j=1, 2, \dots, R-1$) such that

$$\text{g.c.d}\{q_j, p-1\} = 1$$

$$q_j > 6$$

$$q_j > q_{(j-1)}$$

where g.c.d. is greatest common divider. And $q_0 = 1$.

(A-4) The set $\{q_j\}$ is permuted to make a new set $\{p_j\}$ such that $p_{P(j)} = q_j, j = 0, 1, \dots, R-1$,

where $P(j)$ is the inter-row permutation pattern defined in the third stage.

(A-5) Perform the j -th ($j = 0, 1, 2, \dots, R-1$) intra-row permutation as:

$$c_j(i) = c([i \times p_j] \bmod (p-1)), \quad i = 0, 1, 2, \dots, (p-2), \text{ and } c_j(p-1) = 0,$$

where $c_j(i)$ is the input bit position of i -th output after the permutation of j -th row.

B. If $C = p+1$

(B-1) Same as case A-1.

(B-2) Same as case A-2.

(B-3) Same as case A-3.

(B-4) Same as case A-4.

(B-5) Perform the j -th ($j = 0, 1, 2, \dots, R-1$) intra-row permutation as:

$$c_j(i) = c([i \times p_j] \bmod (p-1)), \quad i = 0, 1, 2, \dots, (p-2), \text{ and } c_j(p-1) = 0, \text{ and } c_j(p) = p,$$

(B-6) If $(K = C \times R)$ then exchange $c_{R-1}(p)$ with $c_{R-1}(0)$.

where $c_j(i)$ is the input bit position of i -th output after the permutation of j -th row.

C. If $C = p-1$

(C-1) Same as case A-1.

(C-2) Same as case A-2.

(C-3) Same as case A-3.

(C-4) Same as case A-4.

(C-5) Perform the j -th ($j = 0, 1, 2, \dots, R-1$) intra-row permutation as:

$$c_j(i) = c([i \times p_j] \bmod (p-1)) - 1, \quad i = 0, 1, 2, \dots, (p-2),$$

where $c_j(i)$ is the input bit position of i -th output after the permutation of j -th row.

Third Stage:

(1) Perform the inter-row permutation based on the following $P(j)$ ($j=0,1, \dots, R-1$) patterns, where $P(j)$ is the original row position of the j -th permuted row.

$P_A: \{19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11\}$ for $R=20$

$P_B: \{19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 16, 13, 17, 15, 3, 1, 6, 11, 8, 10\}$ for $R=20$

$P_C: \{9, 8, 7, 6, 5, 4, 3, 2, 1, 0\}$ for $R=10$

$P_D: \{4, 3, 2, 1, 0\}$ for $R=5$

$P_E: \{0\}$ for $R=1$

The usage of these patterns is as follows:

Block length K : $P(j)$

1 to 24-bit: P_E

25 to 159-bit: P_D

160 to 200-bit: P_C

~~201-220~~ to 480-bit: P_A

481 to 530-bit: P_C

531 to 2280-bit: P_A

2281 to 2480-bit: P_B

2481 to 3160-bit: P_A

3161 to 3210-bit: P_B

3211 to 5114-bit: P_A

(2) The output of the mother interleaver is the sequence read out column by column from the permuted $R \times C$ matrix.

Table 2: Table of prime p and associated primitive root

<u>p</u>	<u>g_o</u>
<u>2</u>	<u>1</u>
<u>3</u>	<u>2</u>
<u>5</u>	<u>2</u>
<u>7</u>	<u>3</u>
<u>11</u>	<u>2</u>
<u>13</u>	<u>2</u>

p	g _o	p	g _o	p	g _o	p	g _o	p	g _o
17	3	59	2	103	5	157	5	211	2
19	2	61	2	107	2	163	2	223	3
23	5	67	2	109	6	167	5	227	2
29	2	71	7	113	3	173	2	229	6
31	3	73	5	127	3	179	2	233	3
37	2	79	3	131	2	181	2	239	7
41	6	83	2	137	3	191	19	241	7
43	3	89	3	139	2	193	5	251	6
47	5	97	5	149	2	197	2	257	3
53	2	101	2	151	6	199	3		

4.1.3.2.3.2 Definition of number of pruning bits

The output of the mother interleaver is pruned by deleting the l -bits in order to adjust the mother interleaver to the block length K , where the deleted bits are non-existent bits in the input sequence. The pruning bits number l is defined as:

$$l = R \times C - K,$$

where R is the row number and C is the column number defined in section 4.1.3.2.3.1.

===== End of the text =====

References

- [1] "A Study on merge Interleaver for the Turbo codes", NTT DoCoMo, TSGR1#3(99)217
- [2] NTT DoCoMo and Nortel Networks, "Performance Evaluations for Prime InterLeaver (PIL)", TSGR1#4(99)470
- [3] NTT DoCoMo and Nortel Networks, "Text proposal for Turbo code internal interleaver of S1.12, S1.22", TSGR1#4(99)471

- [4] "Prime interleaver complexity analysis", Nortel Networks and NTT DoCoMo, TSGR1#4(99)513
- [5] "A Study on Turbo-interleaver Flexibility", NTT DoCoMo, TSGR1#2(99)095