

San Diego, USA

February 29 – March 3, 2000

**Agenda item:**

**Source:** Nokia

**Title:** CR 25222-016r1: Clarification on turbo internal interleaver

**Document for:** Decision

---

This CR introduces a clarification on the internal interleaver section of TS 25.222. The functionality is not changed, only presentation is made more consistent with other parts of TS 25.222. After looking at the current internal interleaver description in detail we also found some items missing. Furthermore, TS 25.222 has been changed a lot since June 1999. Thus, we feel that there really is a need to rewrite the current text in order to make it consistent. Our findings are as follows:

- 1) the range of indexes should be from 1 to K inclusive (section 4.2.2.2 in TS 25.222), the current text probably may generate indexes from 0 to K-1, but this is unclear.
- 2) an explicit description for pruning is missing, it is only said how many indexes are pruned but not in which way the pruning is done. This is a consequence of 1), it is difficult to specify pruned bits without explicit indexing.
- 3) the conditions for a minimum prime integer set in A-3, B-3, and C-3 are not tight enough, they do not guarantee a unique selection of  $q_j$ . Hence a new additional requirement for selection has to be included:  $q_j$  is a *least* prime satisfying the selection conditions.

Example:  $K=320$ , so  $R=20$ ,  $C=16$ ,  $p=17$ , and  $p-1=16$ . The correct  $q_j$  is 1,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79. However, the current text allows to take any increasing sequence of prime numbers with  $q_0=1$ ,  $q_1=7$ , so  $q_2=151$ ,  $q_3=199$ , and so on.

Moreover, the current text applies "bad notation", i.e. a symbol has multiple meanings. Also the current text contradicts with use of the global symbols.

For example:

-C is a number, there is a step C and the pattern  $P_C$ .

-there is a step A and the pattern  $P_A$ .

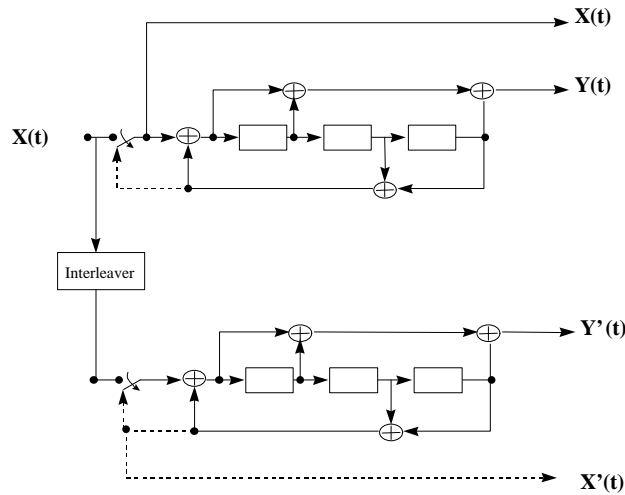
-there is a step B and the pattern  $P_B$ .

The split of the algorithm into three disjoint parts A, B, and C is artificial. Instead, the three are nested of nature.

Finally, the current text does not specify how bits coming from section 4.2.2.2 are interleaved in 4.2.3.2.3

The text in this change request solves these above-mentioned problems.





**Figure 4: Structure of the 8 state PCCC encoder (dotted lines effective for trellis termination only)**

The initial value of the shift registers of the PCCC encoder shall be all zeros.

The output of the PCCC encoder is punctured to produce coded bits corresponding to the desired code rate. For rate 1/3, none of the systematic or parity bits are punctured, and the output sequence is X(0), Y(0), Y'(0), X(1), Y(1), Y'(1), etc.

#### 4.2.3.2.2 Trellis termination for Turbo coding

Trellis termination is performed by taking the tail bits from the shift register feedback after all information bits are encoded. Tail bits are added after the encoding of information bits.

The first three tail bits shall be used to terminate the first constituent encoder (upper switch of figure 4 in lower position) while the second constituent encoder is disabled. The last three tail bits shall be used to terminate the second constituent encoder (lower switch of figure 4 in lower position) while the first constituent encoder is disabled.

The transmitted bits for trellis termination shall then be

$$X(t) Y(t) X(t+1) Y(t+1) X(t+2) Y(t+2) X'(t) Y'(t) X'(t+1) Y'(t+1) X'(t+2) Y'(t+2).$$

#### 4.2.3.2.3 Turbo code internal interleaver

Figure 45 depicts the overall 8 state PCCC Turbo coding scheme including Turbo code internal interleaver. A length of a turbo code internal interleaver is allowed to take any value from 40 to 5114 inclusive assigned according to the rules described in 4.2.2.2. The length is denoted by  $K_i$  for a TrCH  $i$ . Elements of a turbo code internal interleaver are denoted by  $T(k)$ ,  $k = 1, 2, \dots, K_i$ , and each of them stands for the original position of an  $k$ :th interleaved bit. The range of  $T(k)$  is  $1 \leq T(k) \leq K_i$ .

The bits input to the turbo code internal interleaver are denoted by  $o_{ir1}, o_{ir2}, o_{ir3}, \dots, o_{irK_i}$  and the bits after interleaving are denoted by  $x_{ir1}, x_{ir2}, x_{ir3}, \dots, x_{irK_i}$ , where  $i$  is a TrCH number and  $r$  is a code block number (for details see 4.2.2.2). The relationship between the two is defined by:  $x_{irk} = o_{irT(k)}$  for  $k = 1, 2, \dots, K_i$ .

Every interleaving index  $T(k)$  shall satisfy the following stepwise algorithm:

#### 4.2.3.2.3.1 Reference algorithm for turbo interleavers

The following section specific notation is used for the parameters in the algorithm:

$K$	Length of Turbo Code Internal Interleaver for a TrCH
$A$	Number of rows of an $A$ times $B$ matrix
$B$	Number of columns of an $A$ times $B$ matrix
$I$	Prime number
$m$	Primitive root for $I$
$ROP$	Row order pattern
$BR$	Base sequence
$Q$	Minimum prime integer sequence
$MIS$	Minimum row index sequence
$i$	Index in row dimension
$j$	Index in column dimension
$i_0$	Index in row dimension
$z$	Candidate index for Turbo Code Internal Interleaver

1. Assign values for the number of rows  $A$ , the number of columns  $B$ , the prime number  $I$ , and the primitive root  $m$  depending on  $K$ :

If  $480 < K < 531$  then

-  $A = 10$ ;

-  $I = 53$ ;

-  $B = 53$ ;

-  $m = 2$ ;

else

- choose the number of rows  $A$  by

$$A = \begin{cases} 5 & \text{if } 40 \leq K < 160, \\ 10 & \text{if } 160 \leq K < 201, \\ 20 & \text{otherwise.} \end{cases}$$

- find a least prime  $I$  such that  $K \leq A*(I+1)$ ;

- select the number of columns  $B$  by

$$B = \begin{cases} I-1 & \text{if } K \leq A*(I-1), \\ I & \text{if } A*(I-1) < K \leq A*I, \\ I+1 & \text{if } A*I < K \leq A*(I+1). \end{cases}$$

- select  $m$  from Table 2 below on the right side of  $I$ .

endif

2. Select the row order pattern  $ROP$  out of  $Pattern_1$ ,  $Pattern_2$ ,  $Pattern_3$ , and  $Pattern_4$  depending on  $K$ :

$$ROP = \begin{cases} \text{Pattern}_4 & \text{if } 39 < K < 160, \\ \text{Pattern}_3 & \text{if } 159 < K < 201 \text{ or } 480 < K < 531, \\ \text{Pattern}_2 & \text{if } 2280 < K < 2481 \text{ or } 3160 < K < 3211, \\ \text{Pattern}_1 & \text{otherwise.} \end{cases}$$

Pattern<sub>1</sub>: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11}.

Pattern<sub>2</sub>: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 16, 13, 17, 15, 3, 1, 6, 11, 8, 10}.

Pattern<sub>3</sub>: {9, 8, 7, 6, 5, 4, 3, 2, 1, 0}.

Pattern<sub>4</sub>: {4, 3, 2, 1, 0}.

3. Construct the base sequence  $BR(j)$ ,  $j=0,1,2,\dots, I-2$ , by  $BR(0)=1$  and

$$\underline{BR(j) = (m * BR(j-1)) \text{ modulo } I \text{ for } j = 1, 2, \dots, I-2.}$$

4. Selection of the minimum prime integer sequence  $Q(i)$  for  $i=0,1,\dots, A-1$ : assign  $Q(0)=1$  and choose a least prime number  $Q(i)$  such that  $\gcd(Q(i), I-1)=1$ ,  $Q(i)>6$ , and  $Q(i) > Q(i-1)$  for  $i=1,2,\dots, A-1$ . Here  $\gcd(x, y)$  is the greatest common divisor of integers  $x$  and  $y$ .

5. Calculate the minimum row index sequence  $MIS(i)$ ,  $i=0,1,\dots, A-1$ , by

$$\underline{MIS(i) = \begin{cases} ROP(i) * B & \text{if } B = I - 1, \\ ROP(i) * B + 1 & \text{if } B = I \text{ or } B = I + 1. \end{cases}}$$

6. All elements of  $T(k)$  are same as ones obtained from the steps 6.1-6.3:

6.1. Set  $i_0 = 0$  and  $k = 1$ ;

6.1.1 if  $K=A*B$  and  $B=(I+1)$  then  $T(k)=MIS(i_0) + I$ ;  $k = k + 1$ ;  $i_0 = i_0 + 1$ ; endif

6.1.2 for  $j = 0, 1, 2, \dots, I-2$  do

6.1.3 for  $i = i_0, i_0 + 1, i_0 + 2, \dots, A-1$  do

6.1.4  $z = MIS(i) + BR(j * Q(i)) \text{ modulo } (I-1)$ ;

6.1.5 if  $z \notin K$  then  $T(k) = z$ ;  $k = k + 1$ ; else prune  $z$ ; endif

6.1.6 endfor

6.1.7  $i_0 = 0$ ;

6.1.8 endfor

6.2. if  $(I-1) < B$  then

6.2.1 for  $i = 0, 1, 2, \dots, A-1$  do

```

6.2.2 z = MIS( i );
6.2.3 if z ∈ K then T(k) = z; k = k + 1; else prune z; endif
6.2.4 endfor
6.2.5 endif

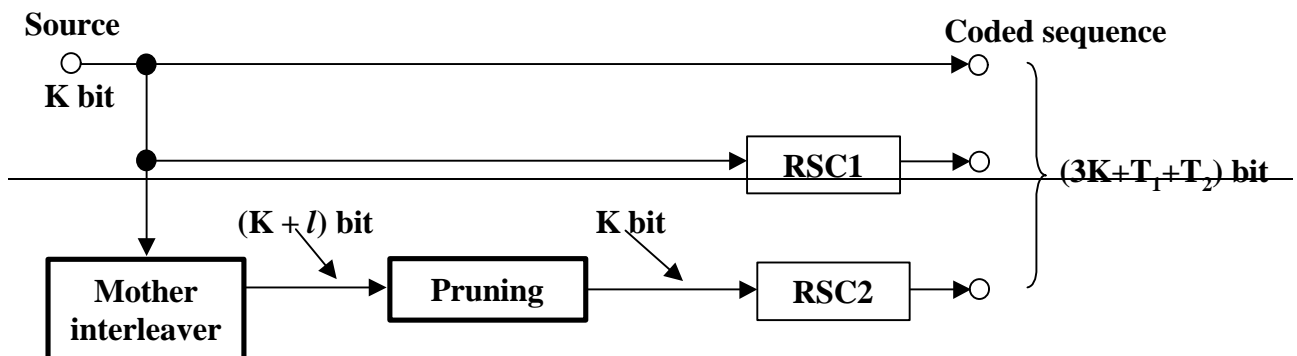
6.3. if I < B then
6.3.1 i0 = 0;
6.3.2 if K = A*B then T(k) = MIS( i0 ) + BR( 0 ); k = k + 1; i0 = i0 + 1; endif
6.3.3 for i = i0, i0 + 1, i0 + 2, ... , A - 1 do
6.3.4 z = MIS( i ) + I;
6.3.5 if z ∈ K then T(k) = z; k = k + 1; else prune z; endif
6.3.6 endfor
6.3.7 endif
    
```

The total number of pruned indexes is  $A*B - K$ .

**Table 2: Table of prime  $l$  and associated primitive root  $m$**

$l$	$m$	$l$	$m$	$l$	$m$	$l$	$m$	$l$	$m$
7	3	47	5	101	2	157	5	223	3
11	2	53	2	103	5	163	2	227	2
13	2	59	2	107	2	167	5	229	6
17	3	61	2	109	6	173	2	233	3
19	2	67	2	113	3	179	2	239	7
23	5	71	7	127	3	181	2	241	7
29	2	73	5	131	2	191	19	251	6
31	3	79	3	137	3	193	5	257	3
37	2	83	2	139	2	197	2		
41	6	89	3	149	2	199	3		
43	3	97	5	151	6	211	2		

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 134 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 to 5114 bits).

##### First Stage:

(1) Determine a row number  $R$  such that

$R=10$  ( $K=481$  to  $530$  bits; Case 1)

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

(2) Determine a column number  $C$  such that

Case 1;  $C=p=53$

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);

— else  $C=p+1$ .

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

— 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:

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

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

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

where  $e_j(i)$  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:

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

where  $e_j(i)$  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  $R=10$

The usage of these patterns is as follows:

Block length  $K$ :  $P(j)$

320 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**

p	g <sub>e</sub>	p	g <sub>e</sub>	p	g <sub>e</sub>	p	g <sub>e</sub>	p	g <sub>e</sub>
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.