**RAN WG1 meeting** **R1#10-00-139**
**Place :** **Beijin**
**Date** **18-21 January 00**
**Title :** **Discussion on De-shuffling in place of 1st IL in the context of Compressed Mode by puncturing.**
**Source :** **Mitsubishi Electric**
**Paper for :** **Discussion**

## Introduction

In [6] Nortel has proposed a description of compressed mode by puncturing in which RM pattern algorithm would be modified, and some dummy bits that we call here prune bits (bit p) would be introduced.
We think that the description made by Nortel is a bit unclear, and we consider an alternative and very similar method, where the prune bits are inserted in the 1st IL.

We also make a summary of the possible policies for the de-shuffling in the 1st IL with corresponding advantages and drawbacks.
We would like to stress that whatever the policy followed a crucial requirement is that the behaviour in normal mode is unchanged, and a desirable requirement is that the same ILing algorithms be used both in normal and compressed mode, except for a different set of parameters.

## References

[1] R1-99 G03 Text proposal to TS 25.212 for Implementation of compressed mode in the IL & MUX chain, source Mitsubishi Electric (MCRD)
[2] R1-99 J03 Means for compressed mode by puncturing in downlink, source Nokia
[3] R1-99 J04 CR019 25.212, Rate matching and multiplexing for compressed mode with puncturing (Method A), source Nokia
[4] R1-99539 Shuffle multiplexing definition and complexity, source Mitsubishi Electric
[5] R1-99672 Optimal Shuffling Multiplexing of coded QOS blocks, source Nortel Networks
[6] R1-00121 Compressed Mode by Puncturing source Nortel
[7] R1-00122 Proposal to take more time to study and standardise compressed mode method A2, source Mitsubishi

## Notations

$x \mapsto \mathrm{BR}_F(x)$ denotes the Bit Reversal function on $\log_2(F)$ bits. In other words $\mathrm{BR}_F$ is a permutation of the set $\{0, 1, \ldots, F-1\}$ with the following examples :

| F | $(\mathrm{BR}_F(0), \mathrm{BR}_F(1), \ldots, \mathrm{BR}_F(F-1))$ |
|---|---|
| 1 | (0) |
| 2 | (0,1) |
| 4 | (0,2,1,3) |
| 8 | (0,4,2,6,1,5,3,7) |

## General on modifying Nortel's proposal by enhancing "1st ILing"

We had the feeling that in the new RM pattern algorithm proposed by Nortel the prune bits insertion decision is not connected to the repeat / puncture decision. We would like Nortel to confirm this.
So, this means that the prune bits (bits p) insertion and removal are steps independent from the RM, and these steps could be new boxes added into the flow of the CCTrCH. However, it seems more appropriate, as in Nortel's proposal, to put these two new steps into existing boxes that can accomodate them nicely.

We think that the prune bit insertion would be more clearly understood if it was done during the 1st IL. Putting this in the RM pattern determination algorithm really makes it unclear. In affecting the CCTrCH in the same way in the end, it would be simpler to make a slight modification of the 1st IL description, that is conceptually easier to understand, rather than this thing in the RM pattern determination.

We also think that the proposal would be simpler if the prune bit removal was done at the 1st IL output and not afterwards. We don't see the benefit of removing the prune bits later, except the somewhat political argument that this allow to keep the radio frame segmentation as an equal segmentation. In fact, when you see the impact on the whole chain, making an unequal segmentation is not more complex than removing prune bits later.
Moreover, removing the prune bits as early as at the 1st IL output is more technology neutral: this way you don't need to have real prune bit indicators stored in memory and accessed, but, instead, you can make the 1st IL with an address generator that generates the read and write addresses with some comparer that make you skip the prune bits. This kind of implementation is very common and convenient when pruning an IL, and it has the advantage of fewer accesses to data memory.

In a nutshell we say that the Nortel proposal would :

- Gain in clarity if the prune bit insertion was not done during the RM step but during the 1st IL step.
- Gain in simplicity of implementation if the prune bit removal was done at the output of 1st IL, thus having no longer any prune bit handling in the sequel.

Finally, we keep the same advantage of Nortel's proposal, that the algorithms in use are the same both in CM and in normal mode, just the parameters change.

### *Insertion and removal of prune bits in the 1st IL*

For TrCH *i* we have :
$Q_i$ bits. to be interleaved
$Q_i = H_i^0 + H_i^1 + \cdots + H_i^{F_i - 1}$
where $H_i^c$ denotes the size of radio frame segmentation segment number c (numbering from 0).

We propose the following description :
$R_i$ and $C_i$ are the respective row number and column number of the 1st IL. They are defined as:
$C_i = F_i$

$$R_i = \max_{0 \le c < F_i} H_i^c$$

So we have the following description for 1st IL :

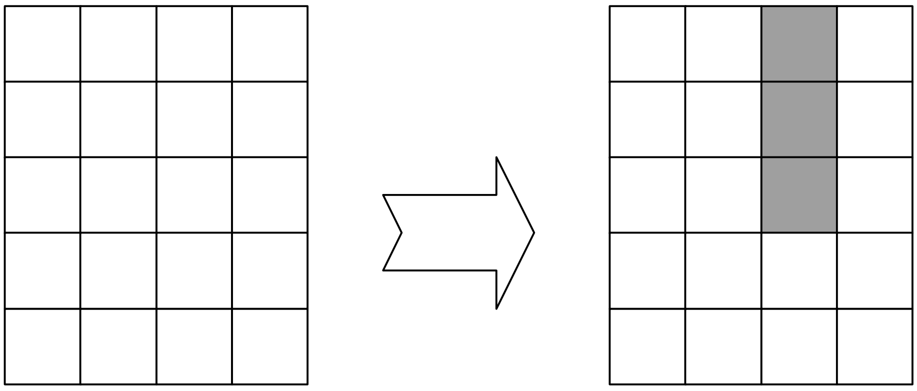
```
-- colomn-wise bit counter initialisation
col = 0
while col < $F_i$ do
    cbi[col] = 0
end do

-- write in
n = 0
while n < $R_i \cdot C_i$ do
    col = $n$ mod $F_i$
    if cbi[col] < $H_i^{\mathrm{BR}_{F_i}(col)}$ do
        $x_{i,n} = h_{i,m}$
        m = m+1
        cbi[col] = cbi[col]+1
    else
        $x_{i,n} = p$
    endif
    n = n +1
end do

-- column permutation
n = 0
while n < $R_i \cdot C_i$ do
    m = $\left\lfloor \frac{n}{F_i} \right\rfloor + \mathrm{BR}_{F_i}(n \bmod F_i)$
    $y_{i,n} = x_{i,m}$
    n = n +1
end do

-- read out
n = 0
m = 0
while n < $R_i \cdot C_i$ do
    $z_{i,m} = y_{i,m}$
    n = n +1
    m = (m +$F_i$) mod $R_i \cdot C_i$
end do

-- pruning
n = 0
m = 0
```

```
while $n < R_i \cdot C_i$ do
    if $z_{i,n} != p$ then
        $q_{i,m} = z_{i,n}$
        $m = m + 1$
    end if
    $n = n + 1$
end do
```

This is can be graphically described in the following example :

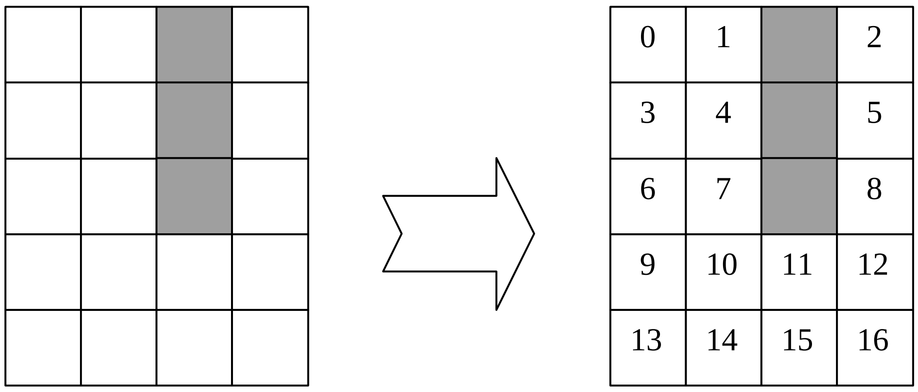
- $Q_i = 17$
- $F_i = 4$
- $H_i^0 = H_i^2 = H_i^3 = 5$ and $H_i^1 = 2$

1st step is to bar some of the square in the 5×4 matrix. Barred squares are shown hatched. For c going from 0 to $F_i$-1. The first $R_i - H_i^{\mathrm{BR}_{F_i}(c)}$ square of column *c* are barred.

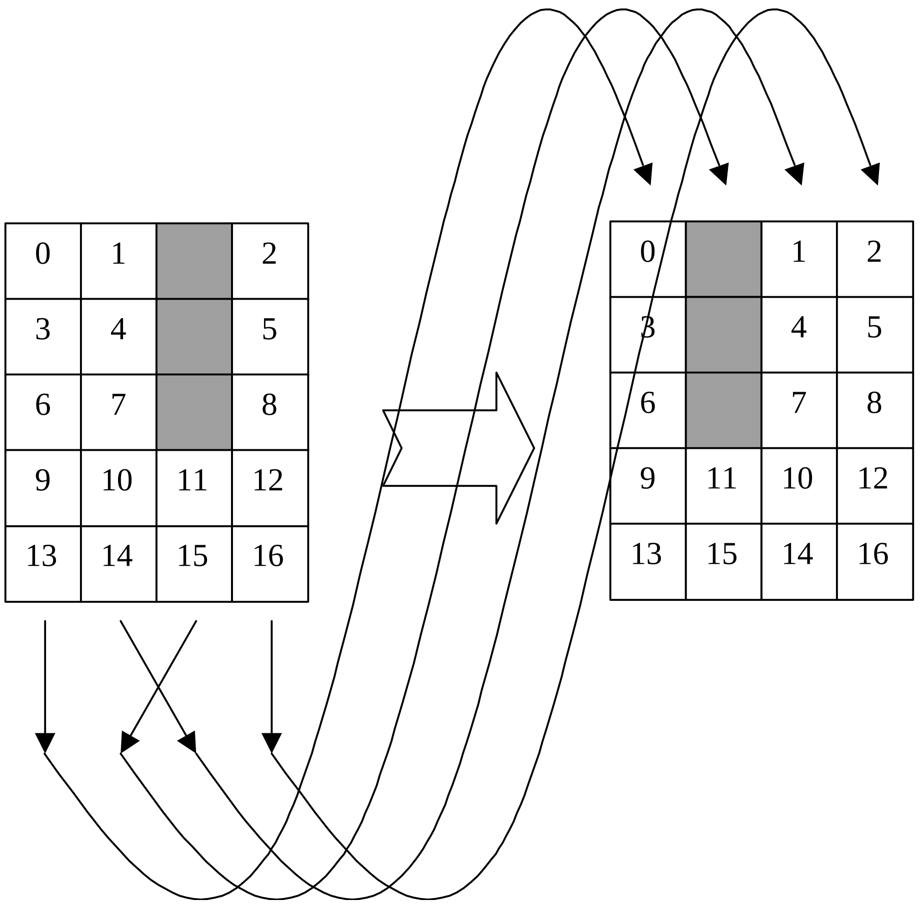


Following step is to fill in the matrix along the rows with skipping barred squares :

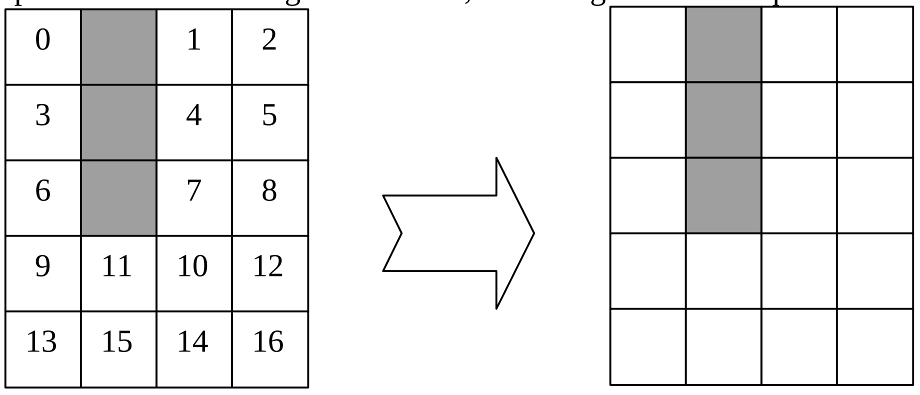
Bits 0, 1, 2, …, 16 are input to the 1st IL



Following step is the well established column permutation:

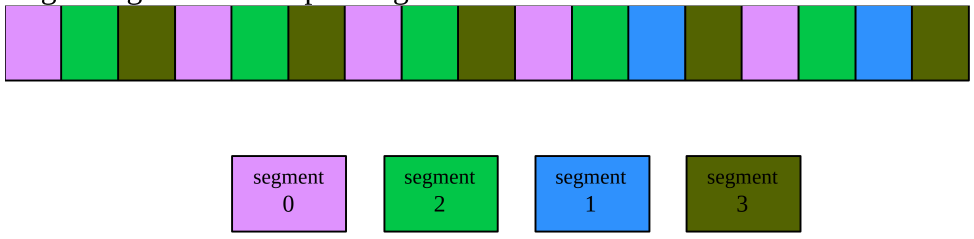


The final step is the read-out along the columns, excluding the barred squares :



Bits 0 3 6 9 13 11 15 1 4 7 10 14 3 5 8 12 16 are output by the 1st IL

The shuffling of segments corresponding to this method is shown below



The figures shows that the segment corresponding to the compressed frame (segment 1) is not perfectly shuffled. This might lead to some degradation compared to other shuffling algorithms, if for instance the compressed frame is badly affected because of TPC recovery.

## Summary of different de-shuffling policies.

Here we make a summary of the different shuffling policies. It is to be noted that whatever the policy, the same algorithm can be used both in normal mode and in compressed mode, and **the behaviour in normal mode is unchanged**, so if the network does not implement CM method A2, then there is no impact.

| policy | variations | advantage | drawback |
|---|---|---|---|
| 1st IL unchanged | - | simple | can cause some degradation of the ILing as mentioned in [7] |
| 1st IL with matrix column having forbidden elements (prune bits) | position of prune bit removal | allows to keep equal segmentation when the prune bit removal is done after radio frame segmentation | shuffling is not optimal, but it seems to be enough to solve the problem of [7]. |
| 1st IL with de-shuffling algorithm of [5] or [4] | choice of the shuflfing algortihm | shuffling is perfect. | complexity compared to use of prune bits needs to be *very carefully* evaluated before going into this. |

## Conclusion

In this paper we have shown that Nortel's proposal [6] can be described more simply, and can be improved by removing the prune bits earlier.
Also we have made a comparison of the different de-shuffling policies.