

Agenda Item: 6.4
Source: Broadcom
Title: Formulaic Collision-free Memory Accessing for Parallel Turbo decoding with ARP Interleave

Document for: Discussion/Decision

1 Introduction

Turbo coding was suggested for 3GPP LTE channel coding. For this coding system, algebraic interleave, almost regular permutation (ARP) [3], is considered as one of the candidates. On the other hand, due to the high data throughput and large block size for LTE parallel decoding of turbo code becomes necessary. In parallel decoding, the decoding algorithm is windowed by dividing the entire block into some subblocks and performing shorter recursions inside the subblock, all assigned to distinct decoding processors working in parallel. Since every processor uses the same algorithm, they all access the memory at the same time instants. This may cause collisions, that is two (or more) accesses at the same time are attempted to the same memory bank, see Fig.1. The collisions will weaken the efficiency of the decoding implementation.

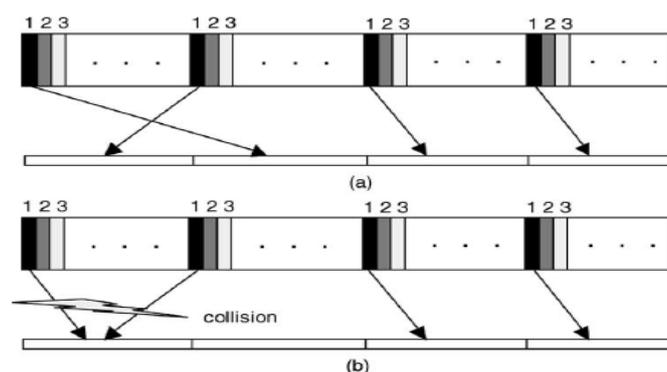


Fig.1 [1] (a) Absence of collisions. (b) A collision happens

In [1], it is shown that there always exists collision-free memory accessing for parallel decoding of any degree and for any interleave. However, in general, the method given in [1] is somehow ad-hoc and is not implementation friendly, especially when a big amount of different block sizes (e.g. the coding system for LTE) have to be supported. In this document, we first reviewed three published formulaic collision-free memory accessing methods [2-4], which work only when the number of processors used is in a

small restrictive set. We then propose a new scheme that allows for more flexible parallelization.

2 Parallel Turbo decoding and collision-free memory mapping

To carry out a parallel decoding with P parallel processors, one has to partition the interleave L of turbo code to P subblocks. Since turbo code uses convolutional encoder as its constituent encoder, the consecutive symbols are connected through the states and therefore the subblock has to contain consecutive information bits, Fig.2. We call the subblock a window and its size the *window size*.

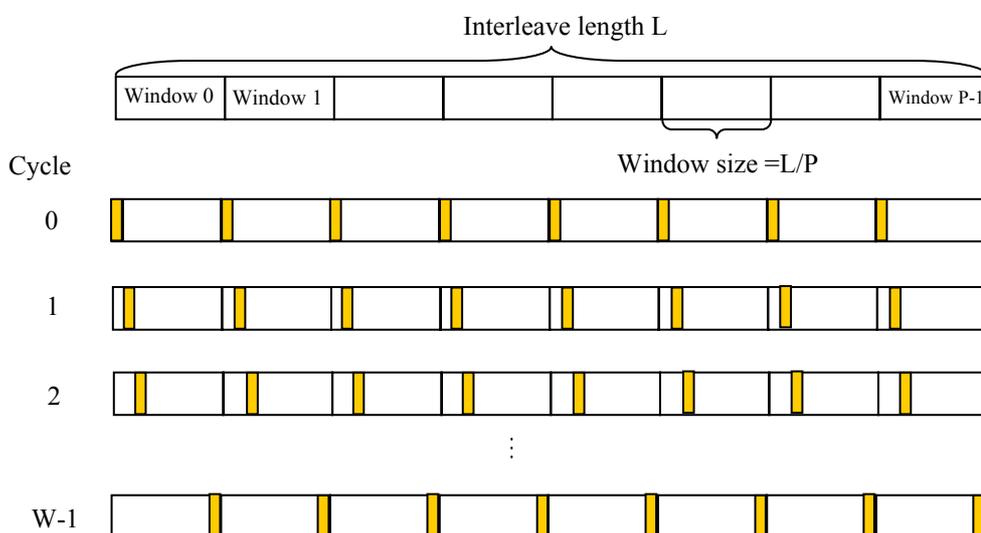


Figure 2. Parallel decoding

Let index set of the information sequence be $I = \{0, 1, \dots, L-1\}$ and the index set of the interleaved information sequence be $\pi(I) = \{\pi(0), \pi(1), \dots, \pi(L-1)\}$. Then the index sets of the P windows for I are

$$\{0, 1, \dots, W-1\}, \{W, W+1, \dots, 2W-1\}, \dots, \{(P-1)W, \dots, PW-1\}$$

see Fig.3.

	Cycle 0	Cycle 1	Cycle 2	...	Cycle t	...	Cycle W-1
Processor 0	0	1	2	...	t	...	W-1
Processor 1	W	W+1	W+2	...	W+t	...	2W-1
...							
Processor s	sW	sW+1	sW+2	...	sW+t	...	(s+1)W-1
...							
Processor P-1	(P-1)W	(P-1)W+1	(P-1)W+2	...	(P-1)W+t	...	PW-1

Window size W, # Parallel processor=P, Interleave size PW

Figure 3. Partition positions

The set of the elements in every column of Figure 3 is defined to be

$$E_i = \{i, W + i, \dots, sW + i, \dots, (P-1)W + i\}, \quad (\text{EQ-1})$$

i.e. the set of the all i-th elements in all P windows. In fact, at cycle i, the P proposes will perform decoding in parallel on the bits of the indices in E_i .

On the interleave side, the index sets of the P windows for $\pi(I)$ are

$$\{\pi(0), \pi(1), \dots, \pi(W-1)\}, \{\pi(W), \pi(W+1), \dots, \pi(2W-1)\}, \dots, \{\pi((P-1)W), \dots, \pi(PW-1)\}$$

	Cycle 0	Cycle 1	Cycle 2	...	Cycle t	...	Cycle W-1
Processor 0	$\pi(0)$	$\pi(1)$	$\pi(2)$...	$\pi(t)$...	$\pi(W-1)$
Processor 1	$\pi(W)$	$\pi(W+1)$	$\pi(W+2)$...	$\pi(W+t)$...	$\pi(2W-1)$
...							
Processor s	$\pi(sW)$	$\pi(sW+1)$	$\pi(sW+2)$...	$\pi(sW+t)$...	$\pi((s+1)W-1)$
...							
Processor P-1	$\pi((P-1)W)$	$\pi((P-1)W+1)$	$\pi((P-1)W+2)$...	$\pi((P-1)W+t)$...	$\pi(PW-1)$

Window size W, # Parallel processor=P, Block size PW

Figure 4. Partition position for interleaved information

Similarly, we define

$$\hat{E}_i = \{\pi(i), \pi(W+i), \dots, \pi(sW+i), \dots, \pi((P-1)W+i)\} \quad (\text{EQ-2})$$

Then a map \mathcal{M} defined from the index set I and $\pi(I)$ to the set $Z_p = \{0, 1, \dots, P-1\}$ is called a *collision-free mapping* [1] for parallel decoding with parallel degree P if it satisfies the following condition:

$$j, j' \in E_i \Rightarrow \mathcal{M}(j) \neq \mathcal{M}(j') \quad \text{and} \quad j, j' \in \hat{E}_i \Rightarrow \mathcal{M}(j) \neq \mathcal{M}(j') \quad (\text{EQ-3})$$

for every $j, j' \in \{0, \dots, L-1\}, j \neq j'$ and all possible i , in other words, the decoded values belonging to the index set at cycle i should be stored into different memory banks.

In [1], it is proved that for any given interleave there always exists a collision-free mapping for parallel decoding of any parallel decoding processor. However, to a communication system which need support channel coding of many different block sizes, the method to generate such a mapping given [1] is hardly implementable.

In [2], one formulaic collision-free mapping is defined, which maps $\pi(jW+i)$ to $\lfloor \pi(jW+i)/W \rfloor$ and $\pi^{-1}(jW+i)$ to $\lfloor \pi^{-1}(jW+i)/W \rfloor$, where W is the window size. An interleave is called the contention-free in [2] if such a mapping used in this interleave will give collision-free memory accessing. Since this mapping uses division, we call it *division mapping*. In fact, the mapping can be represented by

$$\mathcal{M}_{DIV,W} : i \mapsto \lfloor i/W \rfloor \quad (\text{EQ-4})$$

An interleave which is contention-free using this division mapping for all possible window size that divides the interleave length is called a maximum contention-free interleave in [4]. However, as we will show in the following sections, the division mapping is not the only formulaic collision-free mapping and furthermore many interleave may not be maximum contention-free interleave when restrict the formulaic mapping to the division mapping.

3 Almost regular permutation (ARP) and its properties

An almost regular permutation (ARP) of size $L = CW$ (i.e. C is a divider of L) introduced in [2] is defined by

$$i = \pi(j) = jP_0 + \theta + A(j \bmod C)P_0 + B(j \bmod C) \pmod L \quad (\text{EQ-5})$$

where P_0 is relative prime to L , θ is a constant and $A(x)$ and $B(x)$ are integer function defined on $\{0,1,\dots,C-1\}$. To insure the function defined the function is a permutation (i.e. one to one and on to), in [2] $A(x)$ and $B(x)$ are further restricted to

$$A(i)P_0 + B(i) = C[\alpha(i)P_0 + \beta(i)], i = 0, \dots, C-1$$

where α and β are integer functions. In this document, we call C the **dithering period** (or period in short) of the ARP.

Example 1 Let $L = 24, C = 4$ and $P_0 = 7$. Define

$$A(j) = \begin{cases} 0 & j = 0 \\ 0 & j = 1 \\ 1 & j = 2 \\ 1 & j = 3 \end{cases}, \quad B(j) = \begin{cases} 0 & j = 0 \\ 1 & j = 1 \\ 1 & j = 2 \\ 1 & j = 3 \end{cases} \quad \text{then}$$

$$\pi(j) = jP_0 + 4[A(j \bmod C)P_0 + B(j \bmod C)] \bmod L = \begin{cases} jP_0 \bmod L & \text{if } j = 0 \bmod 4 \\ (jP_0 + 4) \bmod L & \text{if } j = 1 \bmod 4 \\ (jP_0 + 4P_0 + 4) \bmod L & \text{if } j = 2 \bmod 4 \\ (jP_0 + 4P_0 + 4) \bmod L & \text{if } j = 3 \bmod 4 \end{cases}$$

is an ARP of size 24 such that $(\pi(0), \pi(1), \pi(2), \dots, \pi(22), \pi(23)) = (0, 11, 22, 5, 4, 15, 2, 9, 8, 19, 6, 13, 12, 23, 10, 17, 16, 3, 14, 21, 20, 7, 18, 1)$.

□

Let π be an ARP interleave of size L with period C . Then we have the following properties.

Proposition 1 $\pi(x_0) = \pi(x_1) \bmod C$ if and only if $x_0 = x_1 \bmod C$.

Proposition 2 π is also an ARP interleave with period $\tilde{C} = mC$ provided $\tilde{C} \mid L$.

Proposition 3 If π is an ARP interleave then its inverse π^{-1} is also ARP.

4 Formulaic collision-free memory accessing mapping

class I: Restricted parallel decoding processor number

Consider an ARP interleave of size L . Then the period C of the ARP is a factor of L . We can write $L=CM$. The memory accessing mapping discussed in this section restricts the number of parallel processors to be a divisor of the interleave length.

4.1 Division mapping

Let number of parallel processors P be a factor of $M=L/C$. Then the division mapping (EQ-4) [3] with window size $W=L/P$ is collision-free. This method is used in Motorola's proposal [6]. We state this conclusion as a theorem below.

Theorem 1 Let π be an ARP of size L with period C and P and W defined as above. If the window size is a multiple of C , then $\mathcal{M}_{DIV,W} : x \rightarrow \lfloor x/W \rfloor$ is collision-free. \square

However, if C is not a factor of window size (equivalently P is not a factor of M). Then the mapping $\mathcal{M}_{DIV,W}$ may not be collision-free, see the following example.

Example 2 Consider the ARP of size $L=24$ from Example 1. We have $C=4$. Take $P=4$ we have $W=6$ which is not a multiple of C , i.e. not satisfy the condition given in Theorem 1. With these parameters we have the following.

Cycle i	E_i	$\mathcal{M}_{DIV,6}(E_i)$	\hat{E}_i	$\mathcal{M}_{DIV,6}(\hat{E}_i)$
0	0,6,12,18	0,1,2,3	0,2,12,14	0,2
1	1,7,13,19	0,1,2,3	11,19,23,21	1,3
2	2,8,14,20	0,1,2,3	22,8,10,20	1,3
3	3,9,15,21	0,1,2,3	5,19,17,7	0,1,2,3
4	4,10,16,22	0,1,2,3	4,6,16,18	0,1,2,3
5	5,11,17,23	0,1,2,3	15,13,3,1	0,2

$\mathcal{M}_{DIV,6}$ puts 4 outputs from the four parallel decoding processors to 2 memory banks.

This causes the collision. \square

Proposition 4 If the period C of an ARP is a prime number, then $\mathcal{M}_{DIV,W}$ with any window size that dividing the interleave length L is a collision-free. \square

4.2 Modulation mapping

In [2], another memory mapping is introduced for ARP, i.e.

$$\mathcal{M}_{MOD,C} : i \mapsto i \bmod C$$

where C is a period of the ARP and the number of parallel processor $P=C$. We call this a modulation mapping. By Proposition 2, we can extend this mapping to the following

$$\mathcal{M}_{MOD,\tilde{C}} : i \mapsto i \bmod \tilde{C} \quad (\text{EQ-6})$$

where $\tilde{C} = mC$ and $\tilde{C} | L$, i.e. the number of parallel processors is $P = \tilde{C}$.

Moreover, the window size is $W = L/\tilde{C}$.

The following theorem shows a sufficient condition of collision-free mapping for modulation mapping.

Theorem 2 Suppose the number of parallel decoding processors $P = \tilde{C}$ and $L = PW$, i.e. W is the window size. If $\gcd(W, \tilde{C}) = 1$ then $\mathcal{M}_{MOD,\tilde{C}}$ is collision-free. \square

If the condition $\gcd(W, \tilde{C}) = 1$ does not hold, the following counter-example shows that $\mathcal{M}_{MOD,\tilde{C}}$ may not be collision-free.

Example 3 Consider the ARP in Example 1, we have the interleave size $L=24$ and the period $C=4$. Take $P=C=4$ we have the window size $W=6$. With $\mathcal{M}_{MOD,4}$, we have

Cycle i	E_i	$\mathcal{M}_{MOD,4}(E_i)$	\hat{E}_i	$\mathcal{M}_{MOD,4}(\hat{E}_i)$
0	0,6,12,18	0, 2	0,2,12,14	0,2
1	1,7,13,19	1,3	11,19,23,21	1,3
2	2,8,14,20	0,2	22,8,10,20	0,2
3	3,9,15,21	1,3	5,19,17,7	1,3
4	4,10,16,22	0,2	4,6,16,18	0,2
5	5,11,17,23	1,3	15,13,3,1	1,3

$\mathcal{M}_{MOD,4}$ maps 4 outputs from the 4 parallel decoding processors to 2 memory banks,

which causes a collision.

4.3 Combined mapping

Consider an ARP interleave of size L and period C . We have shown in the previous sub-sections that when number of parallel processor $P = \tilde{C} = mC$ provided $\tilde{C} | L$ both division mapping $\mathcal{M}_{DIV,W}$ ($W = L/\tilde{C}$) and modulation mapping $\mathcal{M}_{MOD,\tilde{C}}$ may not be collision-free. In [6] we proposed a formulaic mapping that guarantees the collision-free property for all possible number of parallel processor $P = \tilde{C} = mC$ provided $\tilde{C} | L$. Since this method uses both division and modulation we call it combined method which is defined as follows

$$\mathcal{M}_{COM,\tilde{C},W} : x \mapsto \left(x + \left\lfloor \frac{x}{qW} \right\rfloor \right) \bmod \tilde{C} \quad (\text{EQ-7})$$

where q is the smallest positive integer such that $qW = 0 \bmod \tilde{C}$ and $W = L/\tilde{C}$ is the window size.

Theorem 3 Let the number of parallel decoding processors $P = \tilde{C} = mC$ and the interleave length $L = PW$, where W is the window size. Then $\mathcal{M}_{COM,\tilde{C},W}$ is always collision-free.

Example 4 Consider the ARP from Example 1 with $L = 24, C = 4$. When take the number of parallel processor $P=4$, i.e. the window size $W=6$, both $\mathcal{M}_{DIV,6}$ and $\mathcal{M}_{MOD,6}$ are failed to be collision-free, see Examples 2 and 3. As defined in the combined mapping we have $q = 2$ with $qW = 12$. Then

Cycle i	E_i	Mod 4	$\left\lfloor \frac{x}{12} \right\rfloor$	$\mathcal{M}_{COM,4,6}(E_i)$
0	0,6,12,18	0,2,0,2	0, 0,1,1	0,2,1,3
1	1,7,13,19	1,3,1,3	0,0,1,1	1,3,2,0
2	2,8,14,20	2,0,2,0	0,0,1,1	2,0,3,1
3	3,9,15,21	3,1,3,1	0,0,1,1	3,1,0,2
4	4,10,16,22	0,2,0,2	0,0,1,1	0,2,1,3
5	5,11,17,23	1,3,1,3	0,0,1,1	1,3,2,0

On the other hand,

Cycle i	\hat{E}_i	Mod 4	$\lfloor \frac{x}{12} \rfloor$	$\mathcal{M}_{COM,4,6}(\hat{E}_i)$
0	0,2,12,14	0,2,0,2	0, 0,1,1	0,2,1,3
1	11,19,23,21	3,1,3,1	0,0,1,1	3,1,0,2
2	22,8,10,20	2,0,2,0	1,0,0,1	3,0,2,1
3	5,19,17,7	1,3,1,3	0,1,1,0	1,0,2,3
4	4,6,16,18	0,2,0,2	0,0,1,1	0,2,1,3
5	15,13,3,1	3,1,3,1	1,1,0,0	0,2,3,1

Therefore, $\mathcal{M}_{COM,4,6}$ is collision-free. □

4.4 Memory arrangement

With division mapping $\mathcal{M}_{DIV,W}$, the output data from the processors can be stored in the memories according to the natural order. With other two mappings $\mathcal{M}_{MOD,P}$ and $\mathcal{M}_{COM,P,W}$, the data are put in the memories in such a way that they are accessed sequentially in the interleaved phase.

5 Formulaic collision-free memory accessing mappings class II: Flexible parallel-processor number

In the previous section, the number of parallel processors is limited to be a factor of the interleave length. For example, take the interleave size $L = 4272 = 8 * 6 * 89$ and let the dithering period of ARP be $C=8$. Then

- Using $\mathcal{M}_{DIV,W}$, the number of parallel decoding processors P must be a factor of $L/8=6*89$, that is 6, 89 or 534;
- Using $\mathcal{M}_{MOD,\tilde{C}}$, the number of parallel decoding processors P must be 16 or 48
- Using $\mathcal{M}_{COM,\tilde{C},W}$, the number of parallel decoding processors P must be 8, 16, 24, 48 or 712.

However, when the number of the parallel decoding processors $P=2, 3, 4, 5, 7, 9, 10, 11, 12, 13, 14, 15, 17, 18, 19$ or 20 , all the three mappings cannot be used. In the following we propose a flexible but still formulaic memory mapping that will accommodate all the above mentioned P for this example.

Consider an ARP of size L and period C . By Proposition 2, all $\tilde{C} = mC$ provided $\tilde{C} | L$ is also a period. Given any number of parallel processors P such that $P \leq \tilde{C}$. Take a window size W be a smallest number satisfy

$$\text{a) } W \geq L/P \quad \text{and b) } \gcd(W, \tilde{C}) = 1. \quad (\text{EQ-8})$$

Let $L' = P * W$. We can now consider parallel decoding on an ‘‘virtual’’ interleave length L' . Using Figure 4 (or 5 we notify that some cycle with either $sW + t > L$ (or $\pi(sW + t) > L$) becomes dummy cycle, i.e. the corresponded processor in that cycle will do nothing. The modulation mapping once again can be used as a collision-free mapping thanks to the condition b) $\gcd(W, \tilde{C}) = 1$. Since the window size W plays an important role we re-denote the modulation map to be

$$\mathcal{M}_{MOD, \tilde{C}, W} : i \mapsto i \bmod \tilde{C}.$$

We now have the following theorem.

Theorem 4 Let the number of parallel decoding processors P be any number satisfy $P \leq \tilde{C}$ for any $\tilde{C} = mC$ provided $\tilde{C} | L$. When the window size W satisfies conditions in (EQ-8), then $\mathcal{M}_{MOD, \tilde{C}, W}$ is a collision-free memory accessing mapping.

Example 5 Consider the ARP of size $L=24$ and period $C=4$ form Example 1. Let $P=5$, we have $\tilde{C} = 8$ and $W = 5 > L/5$. Then, $L' = P * W = 25$. Using $\mathcal{M}_{MOD, 8, 5}$ we have

Cycle i	E_i	$\mathcal{M}_{MOD, 8, 5}(E_i)$	\hat{E}_i	$\mathcal{M}_{MOD, 8, 5}(\hat{E}_i)$
0	0,5,10,15,20	0,5,2,7,4	0,15,6,17,20	0,7,6,1,4
1	1,6,11,16,21	1,6,3,0,5	11,2,13,16,7	3,2,5,0,7
2	2,7,12,17,22	2,7,4,16	22,9,12,3,18	6,1,4,3,2
3	3,8,13,18,23	3,0,5,2,7	5,8,23,14,1	5,0,7,6,1
4	4,9,14,19	4,1,6,3	4,19,10,21	4,3,2,5

Let us go back to the example given at the beginning of this section, i.e. the interleave length $L = 4272 = 8 * 6 * 89$ and period of the ARP $C = 8$. Then using this new mapping method we have

P (#parallel processors)	\tilde{C} Period	W (window size)	L' (virtual size)
2	8	2137	4274
3	8	1425	4275
4	8	1069	4276
5	8	855	4275
6	8	713	4278
7	8	611	4277
8*	8	535	4280
9	16	475	4275
10	16	429	4290
11	16	389	4279
12	16	357	4284
13	16	329	4277
14	16	307	4298
15	16	285	4275
16	16	267	4272
17	24	253	4301
18	24	239	4302
19	24	227	4313
20	24	215	4300

*) when $P=8$ $\mathcal{M}_{COM,8,534}$ can also be used.

6 Conclusion

In this document we first studied the three existing formulaic collision-free mapping for parallel turbo decoding using ARP as it's interleave and we show the pros and cons of these interleaves. The rule of using all these mapping is that the number of parallel decoding processors has to be a divisor of the interleave length. To overcome this hurdle we proposed a more flexible but still formulaic collision-free mapping which allows the number of parallel decoding processor to be almost any number.

7 References

- [1] A. Tarable, S. Benedetto and G. Montorsi “Mapping Interleaving Laws to Parallel Turbo and LDPC Decoder Architectures,” *IEEE Trans. on Information Theory*, Vol. 50, No. 9, pp.2002-2009, Sept. 2004

- [2] C. Berrou, Y. Saouter, C. Douillard, S. Kerouédan, and M. Jézéquel, “Designing good permutations for turbo codes: towards a single model,” *2004 IEEE International Conference on Communications (ICC)*, Vol.: 1, pp: 341- 345, 20-24 June 2004.

- [3] A. Nimbalkar, T. E. Fuja, D. J. Costello, Jr. T. K. Blankenship and B. Classon, “Contention-Free Interleavers,” *IEEE ISIT 2004*, Chicago, USA, June 27 – July 2, 2004

- [4] Oscar Y. Takeshita, “On Maximum Contention-Free Interleavers and Permutation Polynomials Over Integer Rings,” *IEEE Trans. on Information Theory*, Vol. 52, No. 3, pp.1249-1253, March. 2006

- [5] Motorola, “A Contention-free Interleaver design for LTE Turbo Codes,” 3GPP TSG RAN WG1 #47, R1-063061, Riga, Latvia, 06 – 10 November 2006.

- [6] Broadcom, “A contention-free memory mapping for ARP interleaved turbo codes of arbitrary Sizes,” 3GPP TSG RAN WG1 #47, R1-063243, Riga, Latvia, 06 – 10 November 2006.