3GPP TSG-RAN WG1#47*bis* **R1-070463**
January 15—19, 2007

Source: **Ericsson**
Title: **Contention-Free Interleavers vs. Contention-Avoidance Decoding Solutions**
Agenda Item: **6.4**
Document for: **Discussion and Decision**

1. Introduction

Contention-free condition was originally devised by the ASIC development community to find good interleavers for efficient high-speed parallel turbo decoder implementations [1][2][3][4]. This condition has since been widely adopted by coding researchers and hardware designers because it is informative and decisive: interleavers that satisfy this condition require lower implementation complexity than those that do not. In this contribution, we compare the properties and parallel decoding solutions of contention-free quadratic permutation polynomial (QPP) interleavers [5][6][7] and contention-making Rel6 prunable prime interleavers (PIL) [8].

A QPP interleaver is defined by a simple algebraic expression. Given any integer factorization of the interleaver length $K=MW$, a QPP interleaver can be logically reformatted as an $M\times W$ array. By its maximally contention-free nature, this rectangular representation forms a blueprint for configuring an M-th order contention-free decoder as illustrated in Figure 1. In addition to these preferred parallelization orders (i.e., factors of K), a QPP interleaver can in fact be decoded with any parallelization order as illustrated in Figure 2 because it is a contention-free interleaver.

At its native length $K=RC$, a Rel6 PIL is hierarchically defined as a composition of one inter-row permutation (of length R) and R intra-row permutations (of length C each). Using this native or any other rectangular array representation of the Rel6 PIL, contention-free parallel decoding cannot be configured because it is a contention-making interleaver. Instead, contention-avoidance methods such as starting address staggering (illustrated in Figure 3 and Figure 4) must be used [9][10].

Comparing Figures 1—4, it is clearly seen that the whole purpose of staggering the starting addresses of decoding processors is to avoid memory access contention. The cost of this contention-avoidance solution is analyzed in Section 4. For all parallelization orders M=1, 2, …, 20, the hardware complexity of QPP-based turbo decoders is found to be lower than that of the Rel6 PIL-based decoders. For parallelization orders that are factors of the interleaver length, the hardware cost savings of QPP-based turbo decoders are particularly significant (up to 33%).

2. Contention-Free QPP Interleavers

As contention-free interleavers, parallel contention-free decoding of any order can be supported by the QPP interleavers.

- When the parallelization order is a factor of the interleaver length, all decoding processors behave identically and finish decoding at the same time.
- When the parallelization order is not a factor of the interleaver length, starting points of the decoding processors can be placed freely without staggering consideration.

Hardware implementation advantages of contention-free interleavers come from the fact that the starting points of parallel decoding processors need not be staggered.

2.1. QPP interleavers are contention-free

Given an integer factorization of the interleaver length $K=MW$, a QPP interleaver can be reformatted as an $M\times W$ array. As a contention-free interleaver, this array corresponds to an M-th order parallel decoding where all decoding processors behave identically and finish decoding in W clock ticks. As shown in Figure 1, a K=320 QPP interleaver can be decoded contention-free by 1, 2, 4, 8 or 16 processors. Similar rectangular reformatting shows this QPP interleaver can be decoded contention-free by 5, 10, 16, 20 or 32 decoders.

2.2. QPP interleavers can be decoded with any parallelization orders

For a parallelization order that is not a factor of the interleaver length, contention-free decoding is also supported by a QPP interleaver. For instance, Figure 2 illustrates an M=11-th order parallel decoding method for the K=320 QPP interleaver. This figure demonstrates several strengths of a contention-free interleaver:

- Any parallel decoding order can be supported by a contention-free interleaver.
- The decoding method illustrated here is fundamentally different than the stagger contention-avoidance method used for Rel6 PIL. For QPP interleavers, there is no need to carefully stagger the starting points of the parallel decoding processors.
- In fact, it is advantageous to align as many processors as possible. For instance, by treating the interleaver as a logical 10×32 array, all decoder starting addresses can be aligned.

0	0	60	120	180	240	300	40	100	160	220	280	20	80	140	200	260
1	59	119	179	239	299	39	99	159	219	279	19	79	139	199	259	319
2	198	258	318	58	118	178	238	298	38	98	158	218	278	18	78	138
3	97	157	217	277	17	77	137	197	257	317	57	117	177	237	297	37
4	76	136	196	256	316	56	116	176	236	296	36	96	156	216	276	16
5	135	195	255	315	55	115	175	235	295	35	95	155	215	275	15	75
6	274	14	74	134	194	254	314	54	114	174	234	294	34	94	154	214
7	173	233	293	33	93	153	213	273	13	73	133	193	253	313	53	113
8	152	212	272	12	72	132	192	252	312	52	112	172	232	292	32	92
9	211	271	11	71	131	191	251	311	51	111	171	231	291	31	91	151
10	30	90	150	210	270	10	70	130	190	250	310	50	110	170	230	290
11	249	309	49	109	169	229	289	29	89	149	209	269	9	69	129	189
12	228	288	28	88	148	208	268	8	68	128	188	248	308	48	108	168
13	287	27	87	147	207	267	7	67	127	187	247	307	47	107	167	227
14	106	166	226	286	26	86	146	206	266	6	66	126	186	246	306	46
15	5	65	125	185	245	305	45	105	165	225	285	25	85	145	205	265
16	304	44	104	164	224	284	24	84	144	204	264	4	64	124	184	244
17	43	103	163	223	283	23	83	143	203	263	3	63	123	183	243	303
18	182	242	302	42	102	162	222	282	22	82	142	202	262	2	62	122
19	81	141	201	261	1	61	121	181	241	301	41	101	161	221	281	21

Figure 1 Contention-free decoding of a K=320 QPP interleaver when the parallelization order (M=16) is a factor of the interleaver length.

0	0	60	120	180	240	300	40	100	160	220	280	20	80	140	200	260
1	59	119	179	239	299	39	99	159	219	279	19	79	139	199	259	319
2	198	258	318	58	118	178	238	298	38	98	158	218	278	18	78	138
3	97	157	217	277	17	77	137	197	257	317	57	117	177	237	297	37
4	76	136	196	256	316	56	116	176	236	296	36	96	156	216	276	16
5	135	195	255	315	55	115	175	235	295	35	95	155	215	275	15	75
6	274	14	74	134	194	254	314	54	114	174	234	294	34	94	154	214
7	173	233	293	33	93	153	213	273	13	73	133	193	253	313	53	113
8	152	212	272	12	72	132	192	252	312	52	112	172	232	292	32	92
9	211	271	11	71	131	191	251	311	51	111	171	231	291	31	91	151
10	30	90	150	210	270	10	70	130	190	250	310	50	110	170	230	290
11	249	309	49	109	169	229	289	29	89	149	209	269	9	69	129	189
12	228	288	28	88	148	208	268	8	68	128	188	248	308	48	108	168
13	287	27	87	147	207	267	7	67	127	187	247	307	47	107	167	227
14	106	166	226	286	26	86	146	206	266	6	66	126	186	246	306	46
15	5	65	125	185	245	305	45	105	165	225	285	25	85	145	205	265
16	304	44	104	164	224	284	24	84	144	204	264	4	64	124	184	244
17	43	103	163	223	283	23	83	143	203	263	3	63	123	183	243	303
18	182	242	302	42	102	162	222	282	22	82	142	202	262	2	62	122
19	81	141	201	261	1	61	121	181	241	301	41	101	161	221	281	21

Figure 2 Contention-free decoding of a K=320 QPP interleaver when the parallelization order (M=11) is not a factor of the interleaver length.

3. Contention-Making Rel6 PIL Interleavers

As contention-making interleavers, parallel decoding of Rel6 PIL interleavers requires high-cost contention-avoidance solutions based decoder starting address staggering.

At its native length (before pruning) $K=RC$, a Rel6 PIL is hierarchically defined as a composition of one inter-row permutation (of length R) and R intra-row permutations (of length C each). Let M denote a parallelization order that is a factor of the interleaver length K. Using the native ($K=RC$) or any other ($K=MW$) rectangular

array representation of the Rel6 PIL, parallel decoding of order M can not be configured such that all decoding processors behave identically and finish decoding in the same number of clock ticks.

For instance, the K=320 Rel6 PIL is natively defined as a 20×16 array as shown in Figure 3. This representation, however, offers no help in devising an efficient 16-th order contention-free decoder such as that shown in Figure 1. Instead, a contention-avoidance solution with carefully staggered starting points as illustrated in Figure 3 must be used. By comparing Figure 1 and Figure 3, it is clear to see the processor starting points are staggered in order to avoid contention.

0	304	306	312	313	316	308	318	314	319	317	311	310	307	315	305	309
1	144	154	145	148	147	153	151	146	159	149	158	155	156	150	152	157
2	224	230	238	226	227	234	232	235	239	233	225	237	236	229	231	228
3	64	75	71	74	76	66	65	70	79	68	72	69	67	77	78	73
4	0	2	8	9	12	4	14	10	15	13	7	6	3	11	1	5
5	32	41	46	45	35	37	40	36	47	38	33	34	44	42	39	43
6	80	90	81	84	83	89	87	82	95	85	94	91	92	86	88	93
7	112	123	119	122	124	114	113	118	127	116	120	117	115	125	126	121
8	192	197	193	203	195	198	199	205	207	202	206	196	204	201	200	194
9	288	292	295	293	300	301	289	297	303	299	296	298	291	290	302	294
10	160	173	168	166	172	171	174	165	175	162	167	169	163	164	161	170
11	128	134	142	130	131	138	136	139	143	137	129	141	140	133	135	132
12	208	213	209	219	211	214	215	221	223	218	222	212	220	217	216	210
13	272	276	279	277	284	285	273	281	287	283	280	282	275	274	286	278
14	48	54	62	50	51	58	56	59	63	57	49	61	60	53	55	52
15	16	27	23	26	28	18	17	22	31	20	24	21	19	29	30	25
16	256	265	270	269	259	261	264	260	271	262	257	258	268	266	263	267
17	96	106	97	100	99	105	103	98	111	101	110	107	108	102	104	109
18	240	253	248	246	252	251	254	245	255	242	247	249	243	244	241	250
19	176	181	177	187	179	182	183	189	191	186	190	180	188	185	184	178

Figure 3 Contention-avoidance solution for decoding a K=320 Rel6 PIL interleaver when the parallelization order (M=16 case) is a factor of the interleaver length.

0	304	306	312	313	316	308	318	314	319	317	311	310	307	315	305	309
1	144	154	145	148	147	153	151	146	159	149	158	155	156	150	152	157
2	224	230	238	226	227	234	232	235	239	233	225	237	236	229	231	228
3	64	75	71	74	76	66	65	70	79	68	72	69	67	77	78	73
4	0	2	8	9	12	4	14	10	15	13	7	6	3	11	1	5
5	32	41	46	45	35	37	40	36	47	38	33	34	44	42	39	43
6	80	90	81	84	83	89	87	82	95	85	94	91	92	86	88	93
7	112	123	119	122	124	114	113	118	127	116	120	117	115	125	126	121
8	192	197	193	203	195	198	199	205	207	202	206	196	204	201	200	194
9	288	292	295	293	300	301	289	297	303	299	296	298	291	290	302	294
10	160	173	168	166	172	171	174	165	175	162	167	169	163	164	161	170
11	128	134	142	130	131	138	136	139	143	137	129	141	140	133	135	132
12	208	213	209	219	211	214	215	221	223	218	222	212	220	217	216	210
13	272	276	279	277	284	285	273	281	287	283	280	282	275	274	286	278
14	48	54	62	50	51	58	56	59	63	57	49	61	60	53	55	52
15	16	27	23	26	28	18	17	22	31	20	24	21	19	29	30	25
16	256	265	270	269	259	261	264	260	271	262	257	258	268	266	263	267
17	96	106	97	100	99	105	103	98	111	101	110	107	108	102	104	109
18	240	253	248	246	252	251	254	245	255	242	247	249	243	244	241	250
19	176	181	177	187	179	182	183	189	191	186	190	180	188	185	184	178

Figure 4 Contention-avoidance solution for decoding a K=320 Rel6 PIL interleaver when the parallelization order (M=11 case) is not a factor of the interleaver length.

Similarly, when the parallelization order M is not a factor of the interleaver length, even more delicate staggering of the processors is needed. For the example K=320 and M=11, a decoding processor should normally process a window of size 30 (ceil(320/11)). This would, however, result in access contention among the decoding processors. Instead, the window size needs to be modified in order to introduce additional staggering to avoid contention as shown in Figure 4. Again, by comparing Figure 2 and Figure 4, it is clear to see that contention avoiding staggering is needed for Rel6 PIL interleavers because they are not contention-free interleavers.

4. Hardware Complexity Comparison

The total counts of processing logics and area-equivalent extrinsic buffer sizes of Rel6 PIL and QPP interleavers are compared in this section. For all parallelization orders M=1, 2,…, 20, the hardware complexity of QPP-based turbo decoders is found to be lower than that of the Rel6 PIL-based ones. In particular, for parallelization orders that are factors of the interleaver length, the hardware complexity of QPP interleavers is significantly lower (up to 33% smaller).

4.1. Comparison methodology

A parallel turbo decoder consists internally of M processors, B shared buffer(s) of extrinsic information and multiplexing/routing circuits among the M processors and the B extrinsic buffer(s).

- A decoding processor further consists of processing logics and internal buffers for windowed metrics. From our experience, logics required for a fully parallelized Max-Log-MAP algorithm take around 15Kgates. The buffer size for windowed metrics depends on proprietary choices of window size, temporary bit width and so forth. This buffer size is not included in the comparison for the lack of a universally agreeable benchmark. (Considering that the window size can be far smaller than 512 [11], it is difficult to understand how a decoding processor, including its windowed metric buffer, would require 200K gates as cited in [12].)
- Based on maximum information segment size K_{max}=5120, 35Kbits (assuming 7-bit extrinsic information) is needed for a single extrinsic information buffer. Using an IBM 65nm process example [14], the collective area occupied by 20 buffers holding the same amount of bits would be twice that of a single buffer. For B buffers, we can calculate an approximate area-equivalent buffer size with $35\times1.0372^{(B-1)}$ Kbits. For instance, the area-equivalent buffer sizes are 35K, 39K, 40K, 45K, 49K, 60K and 70K bits for B=1, 4, 5, 8, 10, 16 and 20, respectively.
- From our experience with Rel6 PIL, the gates required for producing R, C, p and v are around 3K and those for calculating the actual PIL addresses are around 7K.
- The computation of QPP addresses requires no more than 2K gates.

4.1.1 Hardware complexity of QPP-based turbo decoders

For K_{max}=5120, parallelization orders M=1, 2, 4, 5, 8, 10, 16, 20 and 32 are factors of the interleaver length. For these preferred parallelization orders, a decoder consists of M decoding processors, one unified extrinsic buffer, and one QPP address generator. Such a decoder would hence require (2+15M) Kgates for logics and 35Kbits of extrinsic buffer.

For parallelization order M that are not factors of K_{max}=5120, a naïve implementation is to split the extrinsic information into B buffers, where B is the smallest factor of K_{max} that is greater than M. For instance, B=16 buffers would be needed for an M=11 decoder. This naïve implementation would require M decoding processors, B extrinsic buffers, and M QPP address generators. The sizes of these naïve implementations represent worst-case upper bounds: 17M Kgates for logics and $35\times1.0372^{(B-1)}$ Kbits of extrinsic buffers.

As shown in Figure 2, a smarter approach would be to align the starting decoding processors. Therefore, instead of requiring M=11 QPP address generators and B=16 extrinsic buffers, only 2 QPP address generators and 2 extrinsic buffers suffice. For K_{max}=5120, a similar hardware reduction method can be applied: aligning all decoder starting addresses by treating it as 160×32 array. That is, it is possible to again use a unified memory buffer and a single QPP interleaver generator to substantially reduce the ASIC hardware sizes. The size of this optimized implementation requires around (2+15M) Kgates for logics and 35 Kbits of extrinsic buffers.

4.1.2 Hardware complexity of Rel6 PIL-based turbo decoders

For parallelization orders M=2, 2, 3, 4 and 5, a decoder consists of M decoding processors, 5 extrinsic buffers, and M PIL address generators. Such a decoder would require (3+22M) Kgates for logics and 40Kbits-equivalent of extrinsic buffers.

For parallelization orders M=6, 7, 8, 9 and 10, a decoder consists of M decoding processors, 10 extrinsic buffers, and M PIL address generators. Such a decoder would require (3+22M) Kgates for logics and 49Kbits-equivalent of extrinsic buffers.

For parallelization orders M=11, 12, …, and 20, a decoder consists of M decoding processors, 20 extrinsic buffers, and M PIL address generators. Such a decoder would require (3+22M) Kgates for logics and 70Kbits-equivalent of extrinsic buffers.

The above computation assumes the structure proposed in [9][10] where the extrinsic memory buffers are organized to allowed very simple accesses in natural orders for the 1st constituent decoders but interleaved orders for the 2nd constituent decoders. In our calculation, no access complexity is assumed for the 1st constituent decoders. A second approach [13] has been recently proposed to organize the extrinsic memory buffers such that simpler accesses are given to the 2nd constituent decoders but the 1st constituent decoders are then forced to access the buffers in interleaved orders. It is not clear how much, if at all, logics savings this hardware moving-around can accomplish since each 1st constituent decoder needs a copy of the intra-row permutation address generator and each 2nd constituent decoder needs some column-wise access circuits. It is, however, clear that the number of separate memory buffers remains the same.

4.2. Hardware implementation advantages of contention-free QPP interleavers

Table 1 Hardware complexity of PIL and QPP based turbo decoders.

Parallelization order *M*	Logics [Kgates]		Area-equivalent extrinsic buffer size [Kbits]	
	Rel6 PIL	QPP	Rel6 PIL	QPP
1	25	**17**	35	**35**
2	47	**32**	40	**35**
3	69	47	40	35
4	91	**62**	40	**35**
5	113	**77**	40	**35**
6	135	92	49	35
7	157	107	49	35
8	179	**122**	49	**35**
9	207	137	49	35
10	223	**152**	49	**35**
11	245	167	70	35
12	267	182	70	35
13	289	197	70	35
14	311	212	70	35
15	333	227	70	35
16	355	**242**	70	**35**
17	377	257	70	35
18	399	272	70	35
19	421	287	70	35
20	443	**302**	70	**35**
32	X	**482**	X	**35**

The total gate counts of processing logics and area-equivalent extrinsic buffer sizes of turbo decoders based on Rel6 PIL or QPP interleavers are compared in Table 1. For all parallelization orders M=1, 2,…, 20, the hardware complexity of QPP-based turbo decoders is always lower than that of the Rel6 PIL-based decoders. For parallelization orders that are factors of the interleaver length, the savings in logics and extrinsic memory are up to 32% and 50%, respectively. For similar hardware costs, higher parallelization orders can be achieved with QPP interleavers.

To further strengthen the value of contention-free condition, the hardware complexity of QPP interleavers based on naive implementation upper bounds can still be found to be lower than that of PIL. These upper bounds have been discussed on RAN1 Reflector forum.

5. Conclusion

Rel6 PIL interleavers are contention-making interleavers. Contention-avoidance decoding of Rel6 PIL interleavers requires higher logic gate counts and larger memory areas.

As maximally contention-free interleavers, parallel contention-free decoding of any order can be supported by the QPP interleavers. For any parallel decoding order, the turbo decoder hardware complexity for QPP interleavers is lower than that for Rel6 PIL interleavers. The savings in hardware implementation costs are highest (up to 33%) when the designed parallelization order is a factor of the interleaver length.

6. References

[1] A. Giulietti, L. Van der Perre, and M. Strum, "Parallel turbo coding interleavers: Avoiding collisions in accesses to storage elements," *Electron. Lett*., vol. 38, no. 5, pp. 232-234, Feb. 2002.

[2] T. K. Blankenship, B. Classon, and V. Desai, "High-throughput turbo decoding techniques for 4G," in *Proc. Int. Conf. 3G Wireless and Beyond*, San Francisco, CA, pp. 137-142, May 2002.

[3] J. Kwak and K. Lee, "Design of dividable interleaver for parallel decoding in turbo codes," *Electron. Lett*., vol. 38, Iss. 22, pp. 1362 1364, Oct. 2002.

[4] J. Kwak, S. M. Park, S.-S. Yoon and K. Lee, "Implementation of a parallel turbo decoder with dividable interleaver," *Proc. of ISCAS '03*, pp. II-65 - II-68, May 2003.

[5] O. Y. Takeshita, "On maximum contention-free interleavers and permutation polynomials over integer rings," *IEEE Trans. Inform. Theory*, vol. 52, no. 3, pp. 1249—1253, Mar. 2006.

[6] R1-063137, Ericsson, "Quadratic permutation polynomial interleavers for LTE turbo coding," 3GPP TSG RAN WG1 #47, Riga, Latvia, Nov. 06 – 10, 2006.

[7] Ericsson, "Quadratic permutation polynomial designs for LTE turbo coding," 3GPP TSG RAN WG1#47*bis*, Sorrento, Italy, Jan. 15–19, 2007.

[8] 3GPP TS 25.212 v6, "Multiplexing and Channel Coding (FDD) (Release 6)."

[9] R1-063265, Samsung and Nortel, "Parallel decoding method for the current 3GPP turbo interleaver," 3GPP TSG RAN WG1 #47, Riga, Latvia, Nov. 06 – 10, 2006.

[10] R1-063357, Panasonic, "Interleaver for LTE turbo code," 3GPP TSG RAN WG1 #47, Riga, Latvia, Nov. 06 – 10, 2006.

[11] A. J. Viterbi, "An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes," *IEEE J. Select. Areas Commun*., vol. 16, pp. 260-264, Feb. 1998.

[12] Samsung, Panasonic and ETRI, "On the contention-free property," 3GPP TSG RAN1 Reflector correspondence, Dec. 5, 2006.

[13] LG Electronics, "Simplified implementation of parallel TC decoder using Rel.6 PIL," 3GPP TSG RAN1 Reflector correspondence, Jan. 8, 2007.

[14] IBM SRAM1PFBS or SRAM2TFBS, configurable at IBM Customer Connect (https://www.ibm.com/technologyconnect/index.jsp).