

Agenda item:

Source: Samsung

Title: New Optimal Coding for extended TFCI with almost no Complexity increase (rev 2)

Document for: Proposal

1. Abstract

TFCI code is very important because decoder depends on the rate information which was carried by TFCI bits. So, TFCI decoding fails, whole decoding also fails. Therefore using a good code for TFCI is very important. However, during studies on Harmonization impact on TFCI, we found some problems. The current coding scheme for extended TFCI is not optimal at all, so there is considerable performance degradation. Based on this observation, Samsung proposed the new Extended TFCI coding scheme having an optimal minimum distance and almost no complexity increase in [1], because they reuse inverse hadamard transform (IHT). Moreover, this code can be generated by natural and simple extension of the current (32,6) TFCI code.

In the last TSG-RAN WG1 meeting, some additional simulation results in TDD mode and Compressed mode were requested. So this contribution add these additional performance results for a comparison between the proposed scheme and the current scheme on TDD mode and Compressed mode. And we also include more analysis on the complexity.

It turns out that TFCI is more critical in both TDD and Compressed modes because situation is much worse than normal, and the new proposed coding scheme gives us an excellent performance at all times. Especially in TDD, there was a concern on additional repetition only for Extended TFCI [3],[4], but this proposed code can remove that concern, so we can have a more flexible operation in TDD.

2. Current Coding Scheme for TFCI

In this section, the current coding scheme is described.

For 6bit TFCI case, the current coding scheme is as following .

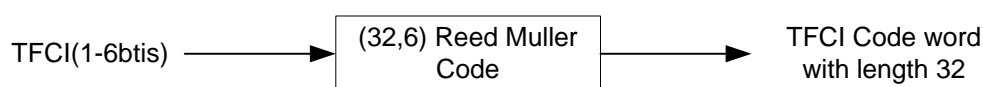


Fig1. The current coding scheme for 6 bit TFCI case

Mapping of the TFCI bits to the code words is described in the following table.

Information bits	Code word
000000	$C_{32,1}$
000001	$\overline{C}_{32,1}$
000010	$C_{32,2}$
.....
111101	$\overline{C}_{32,31}$
111110	$C_{32,32}$
111111	$\overline{C}_{32,32}$

Table 1. mapping of 6bit TFCI bits into codewords

For 7-10bit TFCI case, the current coding scheme is as following .

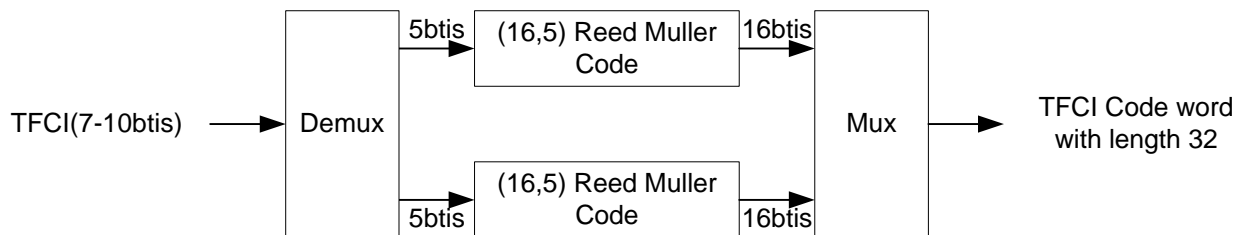


Fig2. The current coding scheme for 7-10bit TFCI case

Mapping of the TFCI bits to the code words is described in the following table.

Information bits	Code word
00000	$C_{16,1}$
00001	$\overline{C}_{16,1}$
00010	$C_{16,2}$
.....
11101	$\overline{C}_{16,15}$
11110	$C_{16,16}$
11111	$\overline{C}_{16,16}$

Table 2. mapping of 7-10TFCI bits into Code words

3. Problems of the current coding scheme for extended TFCI

As we see in figure in this section, we see a big performance difference between current (32,6) and (16,5) Reed Muller code. There is more than 0.8.dB performance difference in the operating point in AWGN. 0.8dB difference in AWGN is a big difference, and this difference must be bigger in fading environment. We should start to think if we can avoid this situation.

If we look at the paper [1], we can see the minimum dsitance of the code of length 32. Those values are summarized in the following table1.

	6 TFCI bits	7 TFCI bits	8 TFCI bits	9 TFCI bits	10 TFCI bits
Optimal minimum distance	16	14	13	12	12

Table 3. Optimal bound of the minimum distance for the code of length 32

Considering the table above, we can see that the current (32,6) Reed Muller code achieves the optimality because the minimum distance of the current (32,6) Reed Muller code is 16. However, the current coding scheme for (32,7-10) code is not optimal at all because the minimum distance of the current (16,5) Reed Muller code is 8. Therefore, we can expect a considerable performance degradation comparing with the optimal code. Furthermore, concatenating short code is a bad idea in terms of the performance.

Therefore, we should try to find a new code which can improve the performance for Extended TFCI, and does not require much complexity increase at the same time.

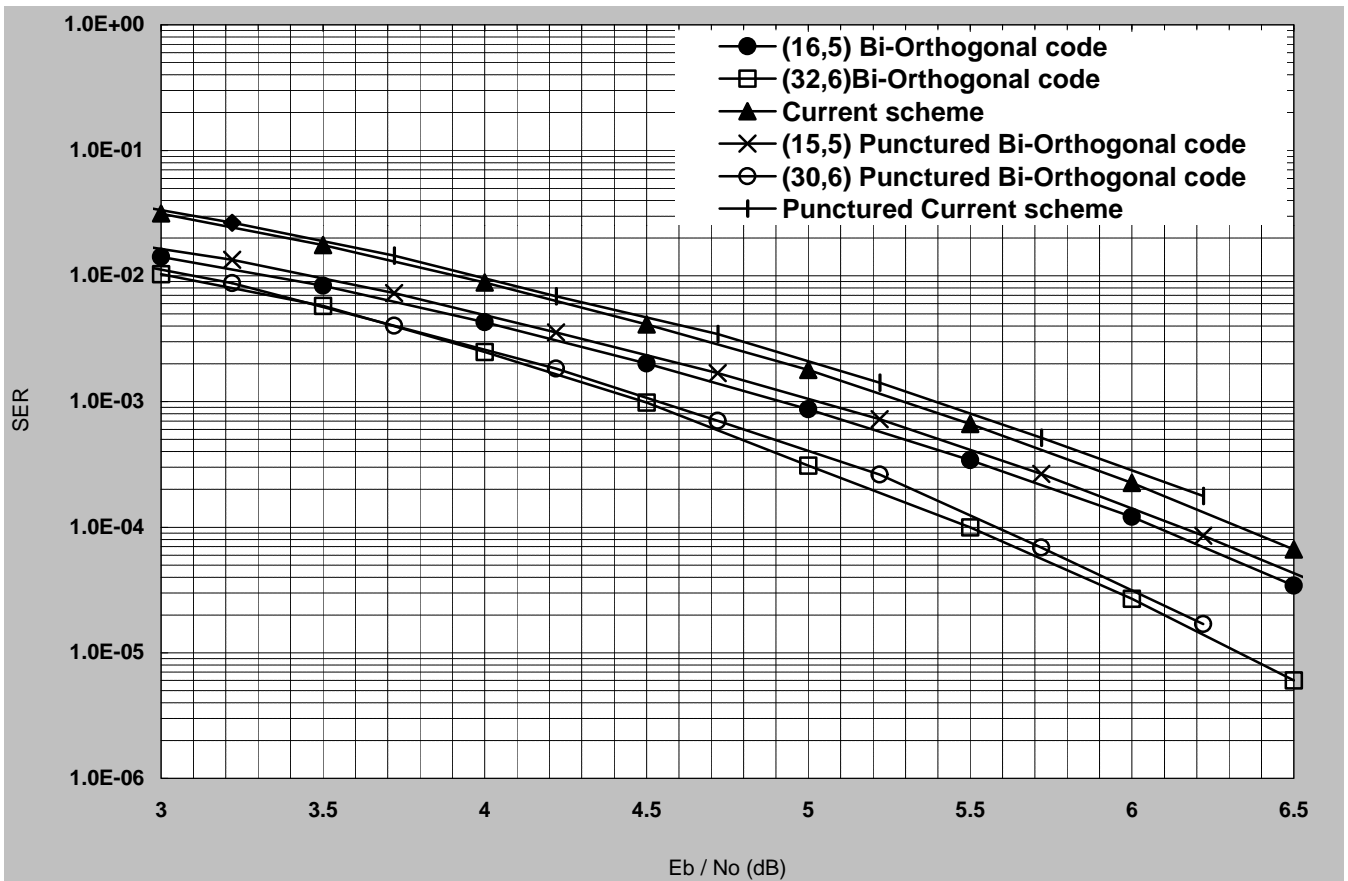


Fig 3 .SER performance of TFCI in AWGN

4. Proposed New TFCI Coding scheme

Based on Problems described in the previous section, we hope to have a better code than the current coding. Surprisingly, we can find a optimal code in the sense of a natural extension of the current (32,6) First Order Reed Muller code. Using sub-code of Second order Reed Muller code, we can achieve that goal. Second order Reed Muller code consists of codewords by adding some masks to the current First Order Reed Muller code.

Then, the number of mask is decided by the number of TFCI bits exceeding 6bit. In n bit TFCI case, $2^{n-6} - 1$ masks is used for coding. For example, if we have 8bits for TFCI, then 3 masks are needed. To cover 10bit TFCI, 15 masks are needed. We can think the current (32,6) Reed Muller has a mask which consists of all zeros.

For 6-10bit TFCI case, the proposed coding scheme is as following .

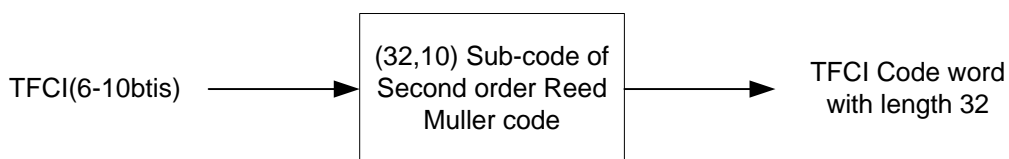


Fig 4. The Proposed coding scheme

Mapping of the TFCI bits to the code words is described in the following table.

Information bits	Code word	Information bits	Code word	Information bits	Code word
000000000	$C_{32,1}$	000100000	$M_1 + C_{32,1}$
000000001	$\overline{C}_{32,1}$	000100001	$M_1 + \overline{C}_{32,1}$	111100000	$M_{15} + C_{32,1}$
000000010	$C_{32,2}$	000100010	$M_1 + C_{32,2}$	111100001	$M_{15} + \overline{C}_{32,1}$
.....
0000111101	$\overline{C}_{32,31}$	0001111101	$M_1 + \overline{C}_{32,31}$	1111111101	$M_{15} + \overline{C}_{32,31}$
0000111110	$C_{32,32}$	0001111110	$M_1 + C_{32,32}$	1111111110	$M_{15} + C_{32,32}$
0000111111	$\overline{C}_{32,32}$	0001111111	$M_1 + \overline{C}_{32,32}$	1111111111	$M_{15} + \overline{C}_{32,32}$

Table 4. Mapping of 6-10bit TFCI bits into codeword corresponding to the proposed scheme

We see that the new coding scheme for extended TFCI is made by just adding 15 masking functions to the current (32,6) Reed Muller code. The minimum distance of the new scheme is 12 as the optimal bound, and can be made as an extension of the current (32,6) Reed Muller code.

5. Encoder Structure

In this section, encoder structure of the new proposed scheme is described. In fact, all Walsh code with length 32 is a vector space of dimension 5. So, there are 5 basis for vector space, for example, $W_{32,2}$, $W_{32,3}$, $W_{32,5}$, $W_{32,9}$, $W_{32,17}$. The current encoding structure is shown in the box of the figure below. The other is the addition because of the extension from (32,6) First Order Reed Muller code. As you see, additional complexity for this extension in the encoder is very minor, and we can also expect a simple decoding procedure because of this natural extension.

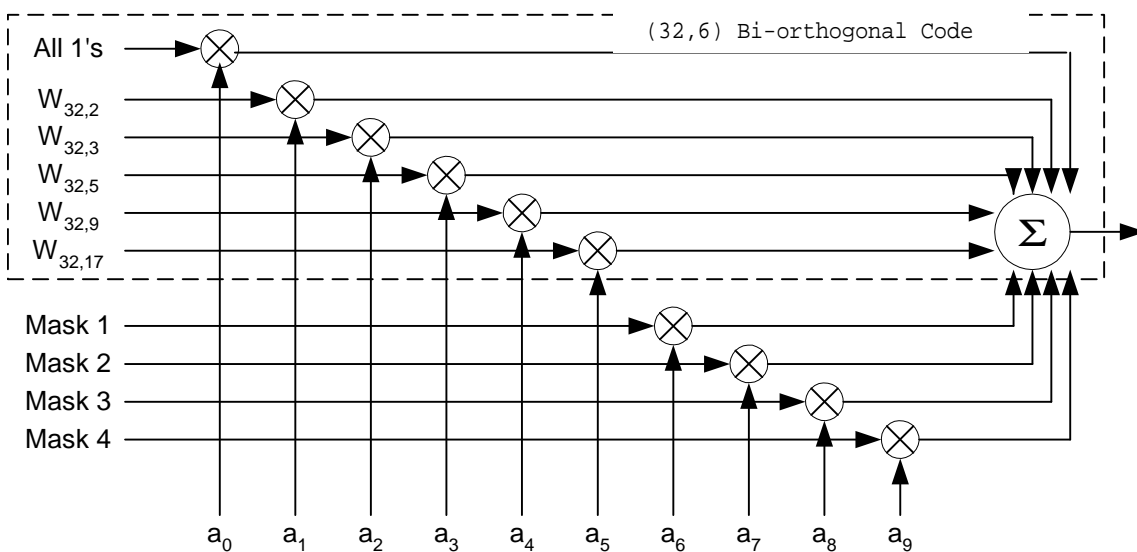


Fig 5 . Encoder structure

Actually, this basis is very meaningful, because we can make an encoder simply by linear operation. For (32,6) code case, if code index n is transferred into binary form $(a_0a_1a_2a_3 \dots a_5)_2$, where $a_i=0,1$, then code becomes $a_5 * W_{32,17} + a_4 * W_{32,9} + a_3 * W_{32,5} + a_2 * W_{32,3} + a_1 * W_{32,2} + a_0 * \text{all } 1\text{'s}$. Moreover, we select a mask set $\{0, M_1, M_2, M_3, \dots, M_{15}\}$ in a vector space with dimension 4, and we can choose a basis, M_1, M_2, M_3, M_4 . Then, codeword set is a vector space of dimension 10 with $W_{32,2}, W_{32,3}, W_{32,5}, W_{32,9}, W_{32,17}, M_1, M_2, M_3, M_4$, and all 1's vector as basis. So, using the basis, the very simple encoder can be implemented, and the structure is as follows.

$$M1 = 00101000011000111111000001110111$$

$$M2 = 00000001110011010110110111000111$$

$$M3 = 00001010111110010001101100101011$$

$$M4 = 00011100001101110010111101010001$$

For example, when input TFCI bits $(0010010001)_2$, output coded symbol is

$$\begin{aligned} & a_9 * \text{Mask}_4 + a_8 * \text{Mask}_3 + a_7 * \text{Mask}_2 + a_6 * \text{Mask}_1 + a_5 * W_{32,17} + a_4 * W_{32,9} + a_3 * W_{32,5} + a_2 * W_{32,3} + a_1 * W_{32,2} + a_0 * \text{all } 1\text{'s} \\ &= (00000001110011010110110111000111) + (00000000111111110000000111111111) + \\ & \quad (11111111111111111111111111111111) \\ &= 1111110110011011001001011000111 \end{aligned}$$

Selecting masks is very important for good codeword. Actually, masks can be derived from Gold Code

6. Weight Distribution

Weight distribution and the minimum distance are the most important factor to determine the performance of the Linear Block code. In this section, we compare the weight distribution and minimum distance of the current coding scheme with that of proposed coding scheme in detail.

In TFCI 6bit case, The minimum distance is 16 and the weight distribution is shown in the following table.

Codeword Weight	Occurence
0	1
16	62
32	1

Table 5. The weight distribution of the current (32,6) bi-orthogonal code

The current scheme and the proposed scheme for Extended TFCI have the different minimum distance and weight distribution, and those values are shown in the table below. The difference of the minimum distance between the current scheme and the proposed scheme is 4. In Coding Theory, the difference 4 of the minimum distance in code length 32, is very high. We expect a considerable performance difference. The performance comparison between the current and proposed scheme is shown in next sections.

Current scheme		Proposed scheme	
Codeword Weight	Occurence	Codeword Weight	Occurence
0	1	0	1
8	60	12	240
16	902	16	542
24	60	20	240
32	1	32	1

Table 6. Comparison of weight distribution of the current & proposed scheme for 7-10bit TFCI case

7. Performance

In this section, we show the comparison of the performance for 3 coding schemes by the simulation in AWGN channel and fading channels. The compared 3 coding schemes are as follows.

1. New proposed sub-code of Second order Reed Muller code
2. Current coding scheme ((16,5) x 2) for the extended TFCI bit
3. Single (16,5) Bi-orthogonal Code

7.1 Basic Performance

In AWGN, new proposed Sub-code of Second order Reed Muller code has about more than 0.6dB as a coding gain compared with the current scheme. And, in fading channel new proposed Sub-code of Second order Reed Muller code has more than 3.0dB coding gain compared with the current scheme. This is the tremendous gain to consider the change of the current Extended TFCI coding scheme. This new code gives us a great coding gain, and does not require much complexity increase at the same time because of natural extension of the current code. The complexity will be explained in next section.

7.1.1 AWGN Channel

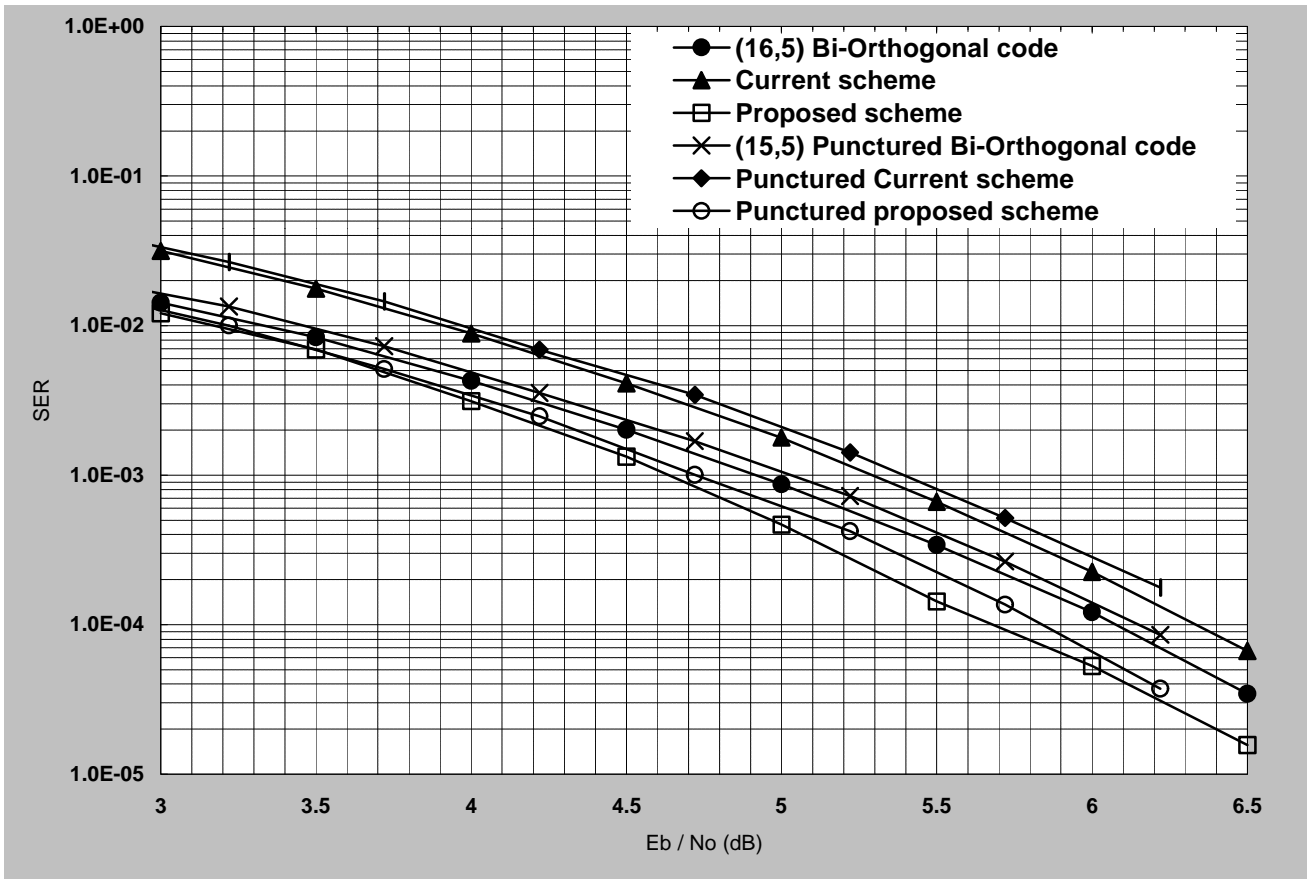


Fig 6. Performance curve in AWGN channel

7.1.2 Fading Channel

- JTC Model
- Vehicular Speed : 30Km
- ideal estimation

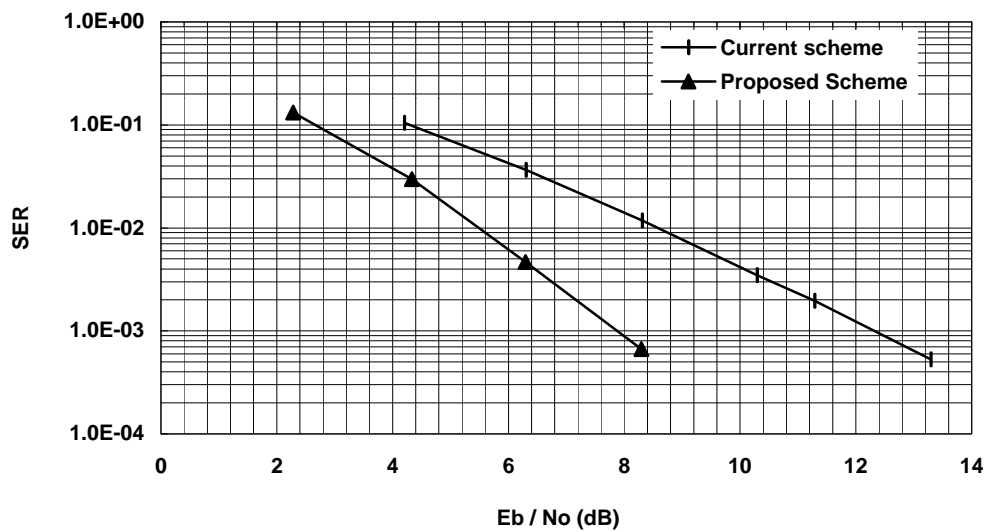


Fig 7. Performance curve in 1-path fading channel with power control and vehicular speed 30km/h

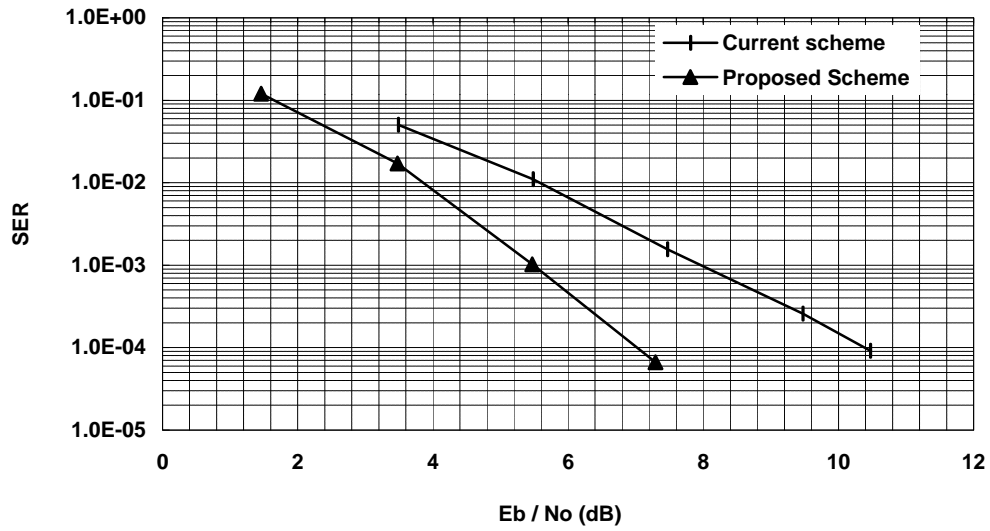


Fig 8. Performance curve in 2-path fading channel with power control and vehicular speed 30km/h.

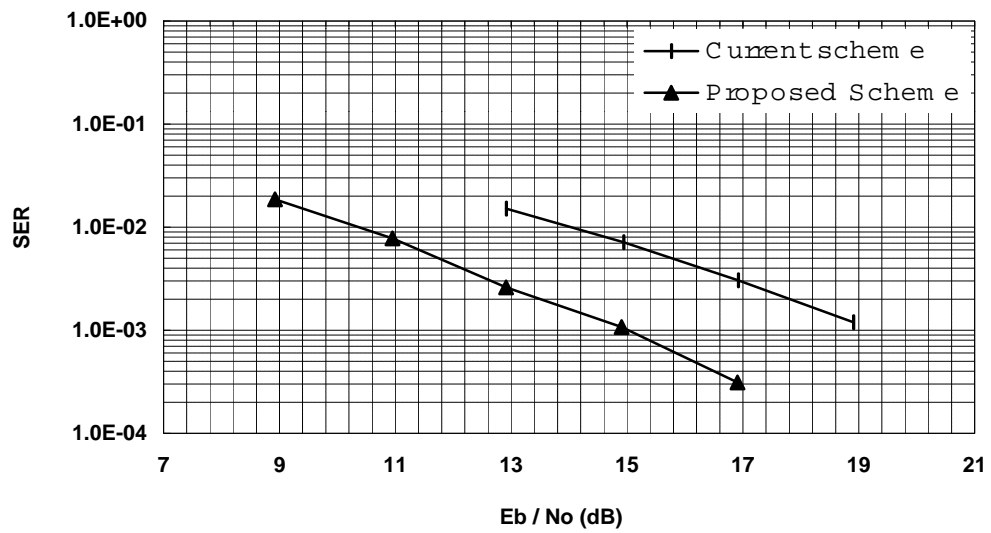


Fig 9. Performance curve in 1-path fading channel with no power control and vehicular speed 30km/h.

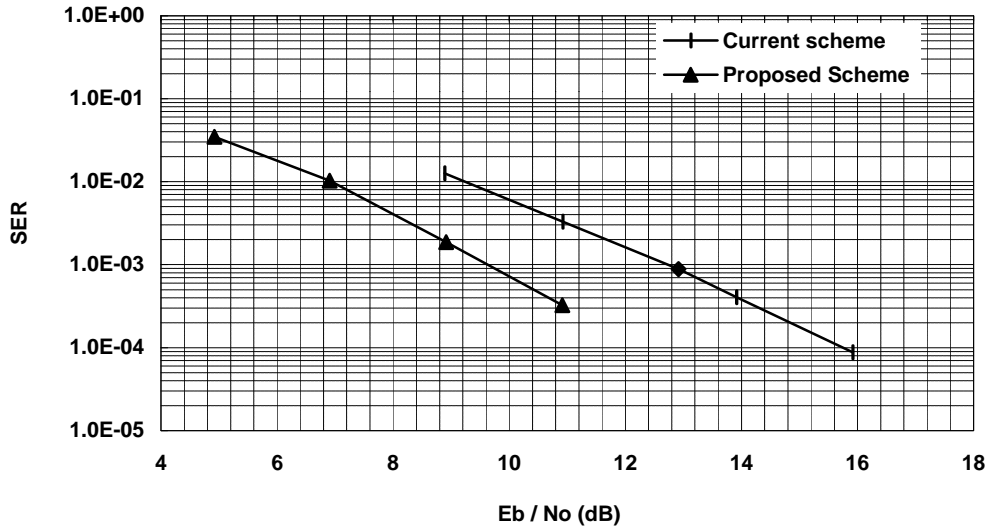


Fig 10. Performance curve in 2-path fading channel with no power control and vehicular speed 30km/h.

7.2 Consideration on 7-9 bits TFCI

In the current specification, there is an algorithm to enhance the performance when we only have from 7 to 9 TFCI bits. So we need to investigate these cases too.

We know that the gain comes from the reduction of signal constellation. Therefore, that algorithm also can be applied to new proposed scheme, and both schemes have some performance improvement. But we are not sure if they will achieve the same amount of performance improvement. If there is the performance difference in these cases, then the case of 7 bit TFCI will be the most significant case. So we run some simulation for 7bit TFCI.

Following figure shows the performance difference between the current scheme and new proposed scheme when we only have 7 bit TFCI. We can see that both schemes achieve the performance improvement comparing with 10 bit TFCI case, but the amount of difference is kept almost the same. Therefore, we can say there is the same gain even when we have 7 –9 bits TFCI.

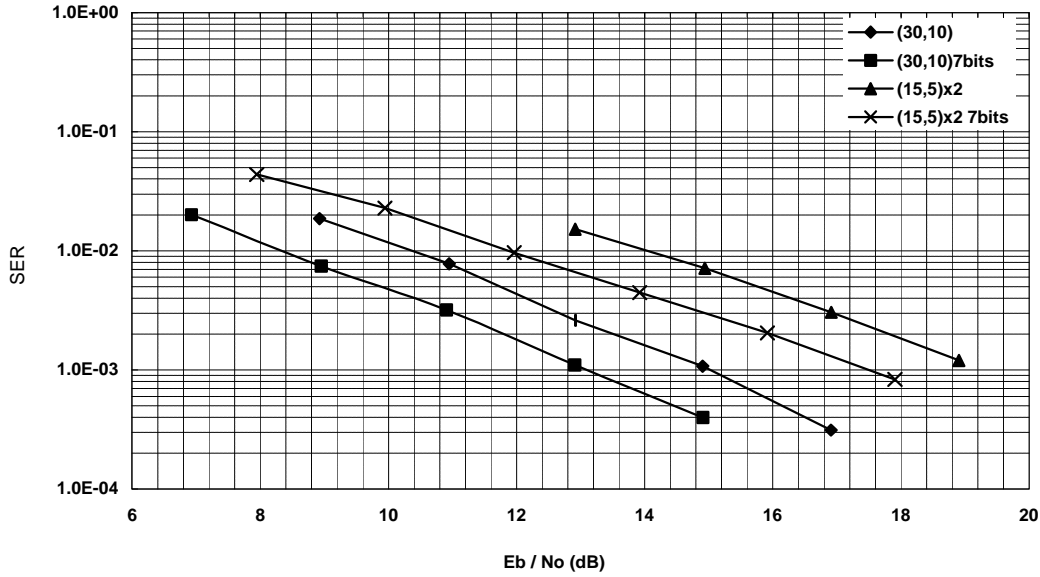


Fig 11. Performance in 1-path fading channel with no power control and vehicular speed 30km/h.

7.3 Performance on TDD Mode

In this section, simulation results for TDD mode are shown. In TDD mode, all TFCI 's are transmitted in one slot. So there is a considerable performance degradation comparing with FDD, because the influence from fading is more serious than that in FDD mode. Therefore, it is more necessary to use a good code in TDD mode than FDD. In [3] and [4], it was pointed out that the current coding scheme for Extended TFCI, (16,5)x2, has the worse performance than the other coding schemes, (16, 5), (32, 6), and it may have additional repetition rather than other codes. However, with this new proposal, we can avoid the concern because the performance of the proposed code is even better than that of (16,5) code. Consequently, the proposed coding scheme provides very natural and stable operation in TDD. Since most of results are simulated in 120km/h in [3], we also perform the simulation in that environment.

7.3.1 Simulation Results for TDD mode

In this section, we show some simulation results for TDD mode. The simulation results of the current scheme and proposed scheme are shown in Figure 12 when the vehicular speed is 120km/h. As we can see in the results, proposed scheme has much better performance than the current scheme. The performance difference between the proposed and the current one is about 5 dB at SER=1e-3.

In [3], the proponent simulated various coding schemes in TDD mode, such as (32,6), (16,5)*2, (16,5) and so on. It was pointed out that (16,5)*2 has the worst performance and the performance difference is large compared with other coding schemes, and they recommended that additional repetition may need for the current (16,5)*2, Extended TFCI. However, the code repetition is not necessary if we use the proposed coding scheme because the performance of the proposed scheme is even better than (16,5) code as you can see in the Figure12. This might give us more flexible and reliable operation.

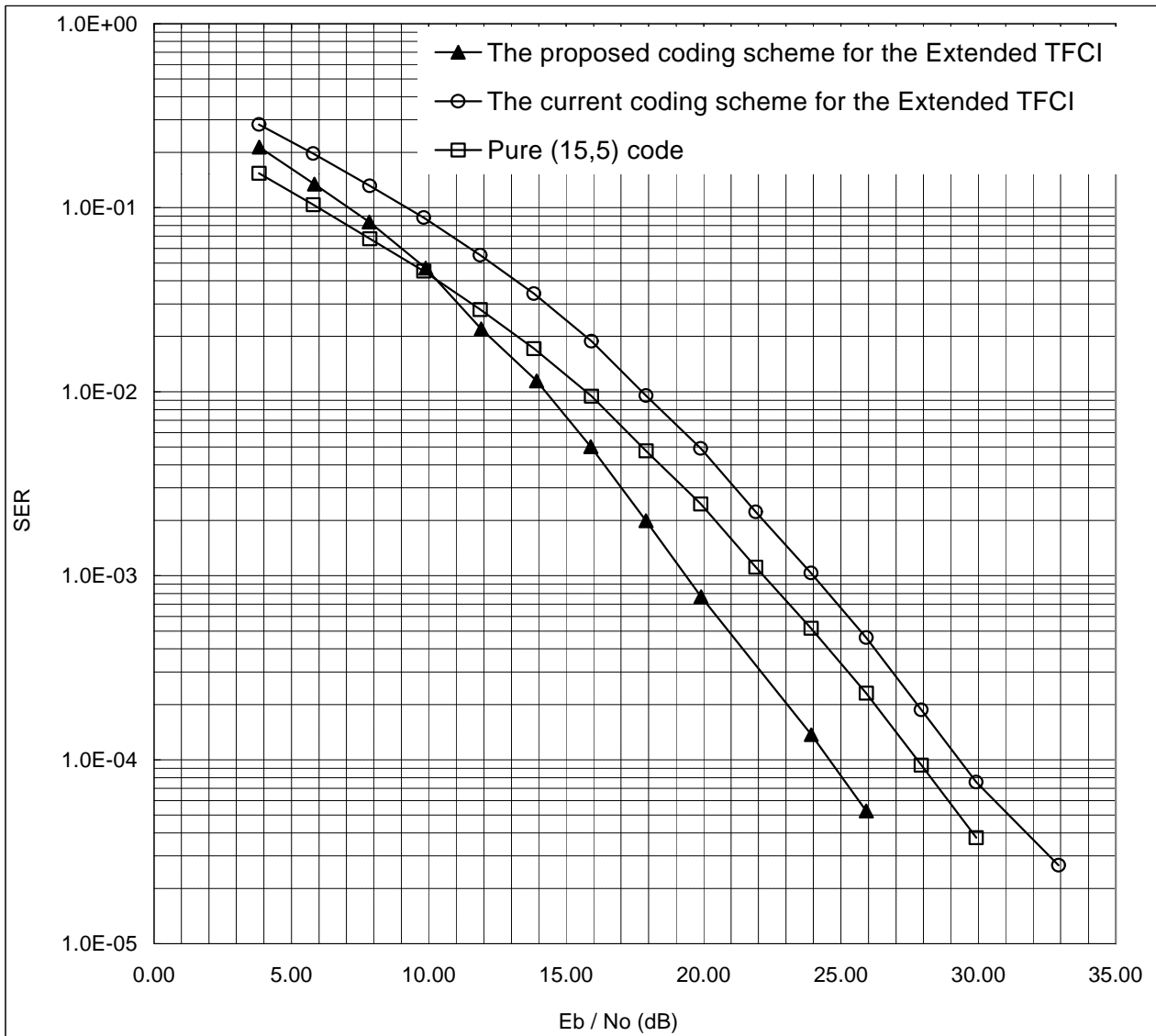


Fig 12. The performance comparison in fading channel with vehicular speed 120km/h

7.4 Performance on Compressed Mode

In this section, some simulation results for Compressed mode are shown. In Compressed mode, some coded symbols are punctured. Note that, in the above section, the performance difference between the proposed scheme and the current scheme is already high even without puncturing because of the difference of minimum distance, i.e. 12 v.s. 8. In fact, the performance difference is getting larger when we puncture more from the original code. The idea is simple and following. 10 v.s. 6 has a more performance difference than 12 v.s. 8 in terms of minimum distance. We will see this in the following simulation section

7.4.1 Simulation Results for compressed mode

In this section, we show some simulation results for Compressed mode, specially, with 8 symbol puncturing case. The simulation results of current and proposed scheme are obtained in fading channel environment with vehicular speed 3, 30,120km/h. As we can see in the results, proposed scheme provides much better performance than the current scheme.

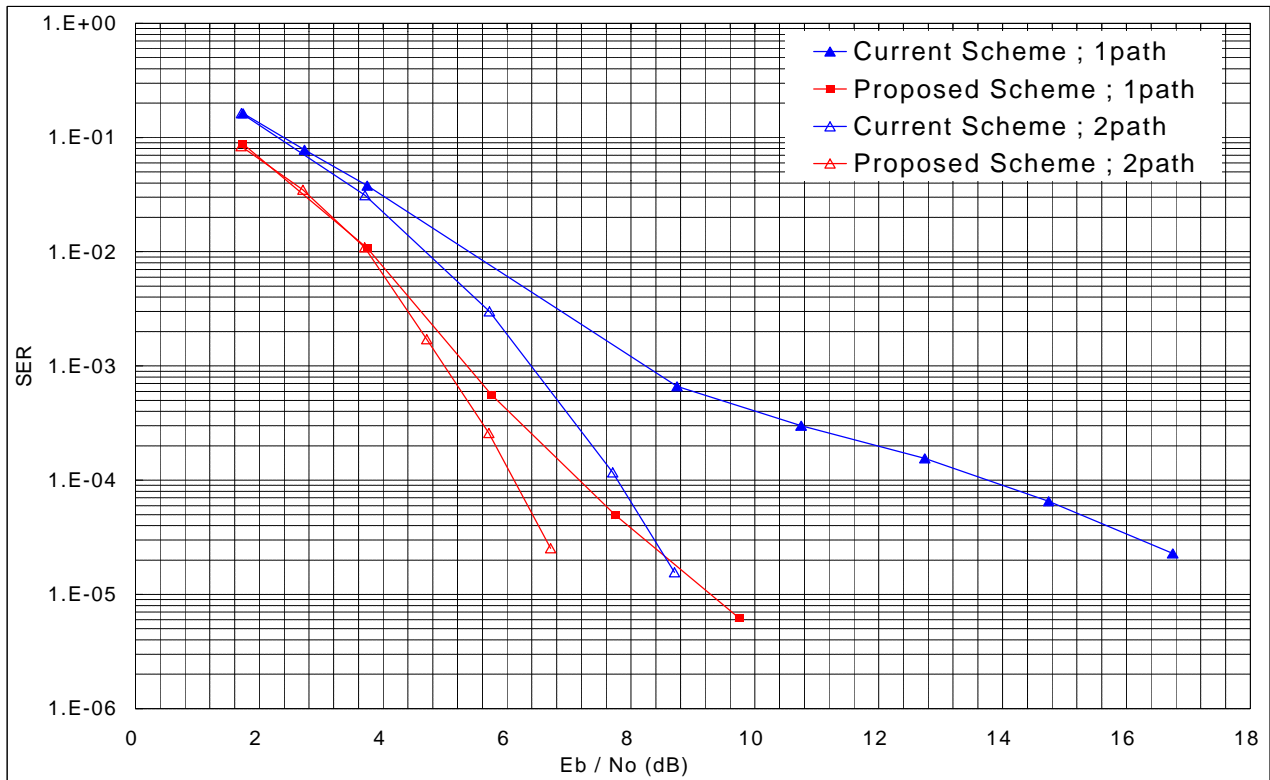


Fig 13. The performance comparison in fading channel with vehicular speed 3km/h and with power control

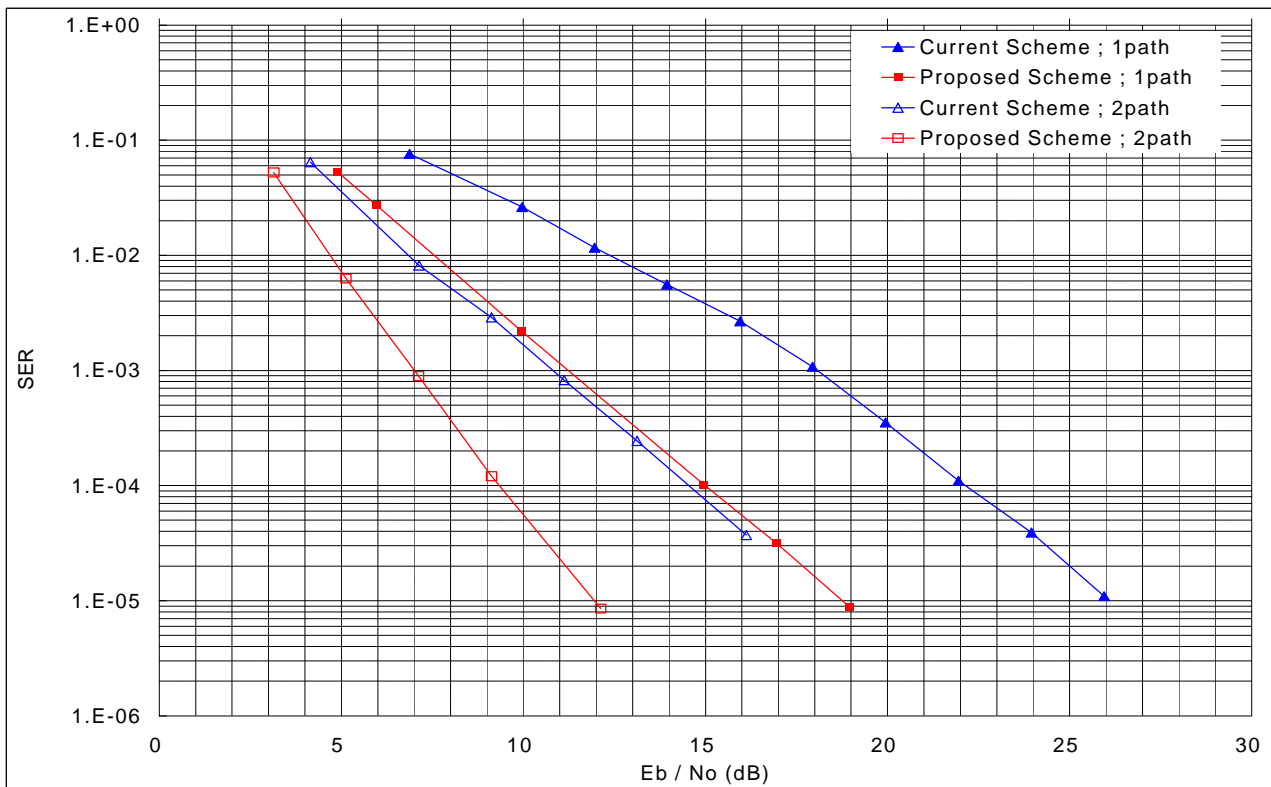


Fig 14. The performance comparison in fading channel with vehicular speed 30km/h and with power control

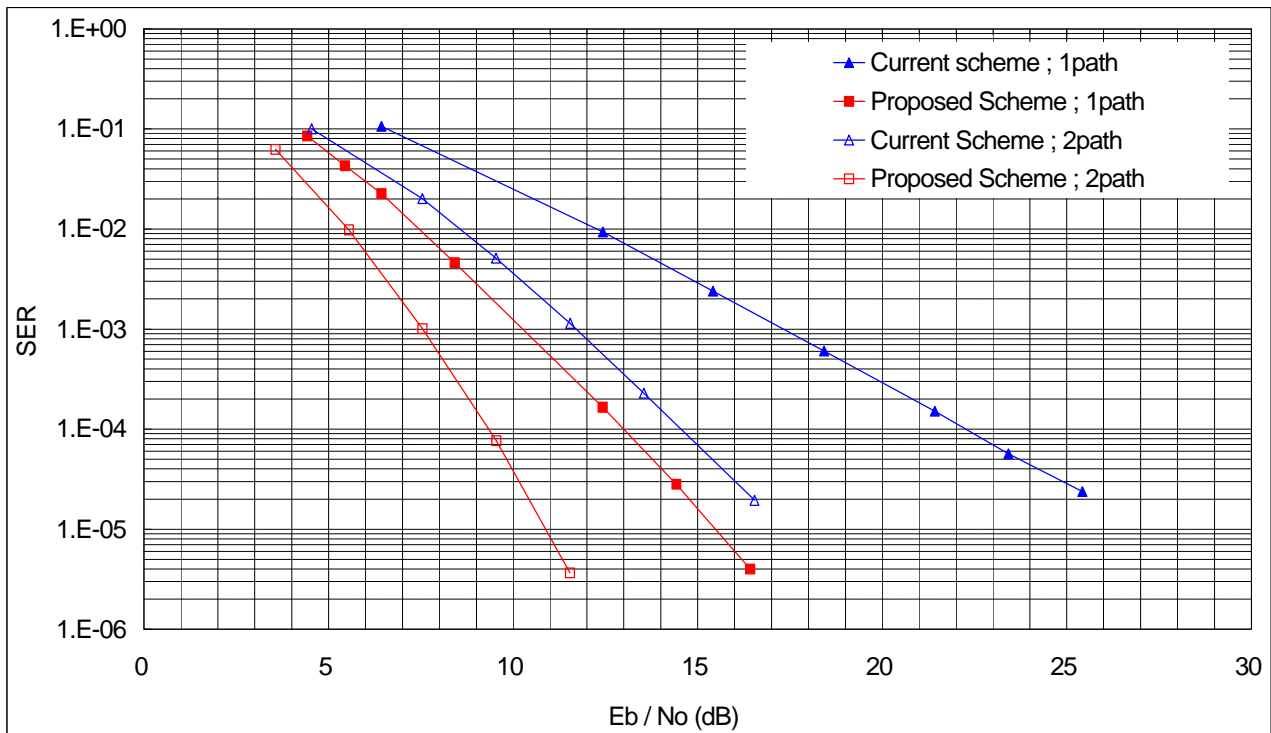


Fig 15. The performance comparison in fading channel with vehicular speed 120km/h and with power control

8. Complexity Issue

In this section, the decoder structure corresponding to the proposed scheme and the decoder structure corresponding to the current scheme are described.

The proposed coding scheme is performed by multiplying 16 possible masking functions with the codeword of the (32,6) First Order Reed Muller code and the decoder structure corresponding to the proposed scheme is as the following Fig16

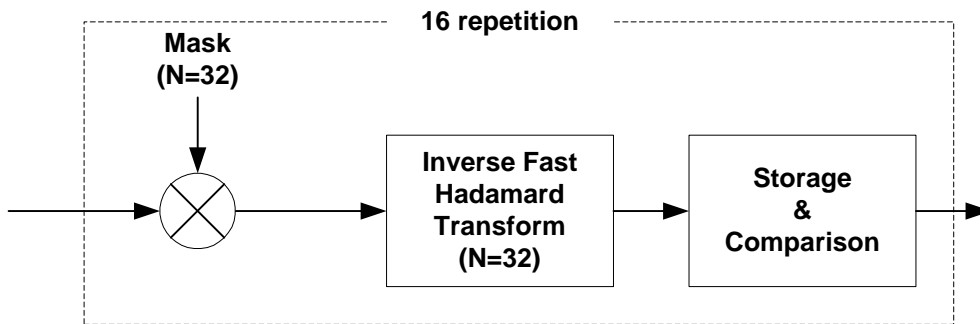


Fig16. The decoder structure corresponding to the current scheme

We can compare this with the decoder of the current (32,6) First Order Reed Muller code. In the figure, when we receive the signal, we multiply all possible masks with the received signal, and perform the fast Hadamard transform 16 times or less iteratively. There are two ways for implementation. One is for hardware, and the other is software.

First, let's consider the case of hardware. When we implement this by hardware, we can simply re-use most of the core block (IHT block), so there is almost no hardware complexity increase, but the decoding delay is increased about 16 times. But this increase of decoding time is not a matter comparing with other processes for demodulation. If we consider the current Hadamard Transform, then it needs about 64 cycles to finish to decode 32-length Hadamard. So for the decoding of the new proposed scheme, it needs about 64×16 cycles. If we have a hardware operating with 70 MHz clock, then the decoding delay for the current scheme is about $0.9 \mu\text{s}$ and for the new coding is about $14.6 \mu\text{s}$. This decoding time delay is no problem at all for the real-time implementation.

When we implement this by software, most of the operation is governed by the IHT operation. To do IHT for length 32, we need 32×5 additions for (32,6) code, and we need $16 \times 32 \times 5$ additions to decode length 10 Extended TFCL proposed by Samsung. It needs 16 times a cycle to decode the proposed scheme, but this is not a big matter either comparing with other demodulation processes.

With this proposal, there is an obvious complexity benefit comparing with other optimal codes because we can use IHT. Furthermore, we can reuse the IHT block because of the natural extension. This gives us great flexibility and benefit to implement.

9. Conclusion

During studies on Extended TFCI, we found some performance problem. The current coding scheme is not optimal at all, and too far from optimal. Considering the importance of TFCI because of big influence on link performance, we need to have a good code. In this proposal, we proposed new optimal TFCI coding which achieves a big performance improvement (0.6 dB in AWGN and more than 3 dB in Fading). It also give us reliable and flexible operation in both TDD and FDD. We don't have to worry about additional repetition in Extended TFCI with the proposed code. It also turns out that this new code can be generated by natural and simple extension of the current (32,6) First order reed Muller code, so there is no significant complexity increase to decode. So we strongly recommend to change the current Extended TFCI coding scheme to new proposed one. There is no reason not to use this optimal code which does not require much complexity increase.

10. Reference

- [1] : "An Updated Table of Minimum-Distance Bounds for Binary Linear Codes" – A.E. Brouwer and Tom Verhoeff, IEEE Transactions on Information Theory, VOL. 39, N). 2, March 1993
- [2] TSGR1#6(99)970, " Harmonization impact on TFCI and New Optimal Coding for extended TFCI with almost no Complexity increase (rev 1)" - Samsung
- [3] TSGR1#6(99)865, "Simulation results of TFCI coding performance for TDD" - Siemens
- [4] TSGR1#6(99)1005, "Report from AD Hoc #1 : TDD" – Ad Hoc #1

[Contact point]

Jaeyoel Kim kimjy@telecom.samsung.co.kr

Hyeonwoo Lee woojaa@samsung.co.kr

Heewon Kang hkang@metro.telecom.samsung.co.kr

TEL : +82-342-779-6626

TEL : +82-342-779-6699