Mueller Exhibit 19

View: Next in topic | Previous in topic
Next by same author | Previous by same author
Previous page *(July 1999)* | Back to main 3GPP_TSG_RAN_WG1 page
Join or leave 3GPP_TSG_RAN_WG1
Reply | Post a new message
Search
Log in

Options: Chronologically | Most recent first
Proportional font | Non-proportional font

```
Date:         Sat, 10 Jul 1999 16:21:47 +0900
Reply-To:     [log in to unmask]
Sender:       "3GPP_TSG_RAN_WG1: TSG RAN Working Group 1"
              <[log in to unmask]>
From:         "(Min-goo KIM)" <[log in to unmask]>
Subject:      [AH04] & [AH05] Tdoc 919: "Unified rate matching scheme "
Content-Type: multipart/mixed;
```

```
Dear Ad Hoc 4 and 5 colleagues,

Please find attached the final version of our proposal.


 Tdoc number: TSGR1#6(99)919
 Title:
        "Unified rate matching scheme for turbo/convolutional codes
         and up/down links"
 File name: Tdoc_919.ZIP

I'm sorry for our late submission.
I will bring the paper copies to the meeting.

Ms. Marlene Forina, could you please place these documents on the server?
Thank you in advance.

Best Regards,
Min Goo KIM
Samsung Electronics Co.
E-mail: [log in to unmask]
```

TDOC_919.ZIP [application/zip]

Back to: Top of message | Previous page | Main 3GPP_TSG_RAN_WG1 page

LIST.ETSI.ORG

TSG-RAN Working Group1 meeting #6 **TSGR1#6(99)919**
Espoo, Finland, 13-16, July 1999

Agenda Item:
Source: SAMSUNG Electronics Co.
Title: Unified rate matching scheme for Turbo/ convolutional codes and up/down links
Document for: Discussion & decision

---

1. Introduction

In this document, we propose a new common rate matching scheme for both convolutional codes and turbo codes, and for both up-link and down-link. Recently, LGIC and Fujitsu have proposed rate matching algorithms with different algorithms for convolutional coding and turbo coding, respectively [1]-[5]. Also, Siemens and Nortel have investigated rate matching algorithms for both down-link and up-link respectively[6]-[7]. However the different characteristics of convolutional codes and turbo codes make it difficult to find an optimal solution for all cases. In [6], Siemens proposed a rate matching algorithm for convolutional codes for up-link and down-link, which has been adopted in [8]. In [1]-[5], Fujitsu and LGI C proposed two rate matching algorithms for down-link turbo codes. The performances of two algorithms are similar for turbo coding and it seems difficult to find performance improvement in terms of puncturing patterns in both algorithms. This reason is related to the turbo interleaver characteristics such that the random property of turbo interleaver gives a similar performance regardless of puncturing pattern for parity symbols. In addition, for convolu tional codes both algorithms have shown little performance degradation. In this contribution, we propose a new common rate matching scheme for both convolutional codes and Turbo codes for up/down-links which reuses conventional rate matching algorithm with parameter control.

2. Unified rate matching method

Figure 1 illustrates conventional rate matching algorithm for convolutional codes in down-link. The rate matching algorithm has been proposed by Siemens and adopted for convolutional codes. All symbols from channel encoder are punctured uniformly. However, for turbo codes systematic code symbols should not be punctured and the number of the punctured parity symbols should be distributed evenly. In order to achieve optimal puncturing for turbo codes in down-link with minimal complexity, we propose a new rate matching scheme given in Figure 2. The idea can also be applied in up-link. In Figure 2, all code symbols are classified into three parts such as systematic code symbols (information symbols), the first parity symbols from constituent encoder 1, and the second parity symbols from constituent encoder 2, respectively. Rate matching algorithm blocks (RMB) for each part are the same as conventional algorithm except modifying number of bits per matching block and number of bits per matched block. For R=1/3 turbo codes, number of bits per matching block *Nc* and number of bits per matched block *Ni* are reduced to 1/3 of the total number of bits per matching blocks and matched blocks, respectively. Parameter (a,b) is selected for generating arbitrary puncturing or repetition patterns. Typically, (2,1) is used for conventional rate matching algorithm. If (a,b) for all RMB is fixed to the same value then all patterns for puncturing and repetition are also the same. The definition of parameters is as follows. It is obvious that the proposed algorithm is identical to conventional rate matching algorithm in [8] except parameter control. So, there are no implementation complexity increases.

- Definition of parameters

- *Nc*: Number of bits per matching block (=*Ncs* for convolutional codes)
- *Ni*: Number of bits per matched block (=*Nis* for convolutional codes)
- *Ncs*: Total number of bits per all matching blocks
- *Nis*: Total number of bits per all matched blocks
- *(a,b)*: Parameter for conventional rate matching algorithm (to control pattern generation)

The following shows the definition and roles of parameter (a,b) in conventional rate matching algorithm. Parameter "b" is usually 1 and parameter "a" is only variable for considering hardware or software implementation complexity.

Conventional Rate Matching Algorithm with parameters (a,b) control

Let's denote:

$S_0 = \{d_1, d_2, \ldots, d_{N_C}\}$ = set of N_C data bits

The rate matching rule is as follows:

```
    if puncturing is to be performed

        $y = N_C - N_i$
        $e = (2*S(k) * y + b*Nc) \bmod a*N_C$
                    -- initial error between current and desired puncturing ratio (downlink : S=0)
        $m = 1$                         -- index of current bit
        do while $m <= N_C$
            $e = e - a * y$             -- update error
            if $e <= 0$ then            -- check if bit number m should be punctured
                puncture bit $m$ from set $S_0$
                $e = e + a*N_C$         -- update error
            end if
            $m = m + 1$                 --   next bit
        end do
    else
        $y = N_i - N_C$

        $e = (2*S(k) * y + b*Nc) \bmod a*N_C$
                -- initial error between current and desired puncturing ratio (downlink : S=0)
        $m = 1$                         -- index of current bit
    do while $m <= N_C$
        $e = e - a*y$                  -- update error
        do while $e <= 0$              -- check if bit number m should be repeated
            repeat bit $m$ from set $S_0$
            $e = e + a*N_C$            -- update error
        enddo
        $m = m + 1$                    -- next bit
    end do
end if
```

3. Rate matching for down-link

3.1 Convolutional codes

For convolutional codes, the proposed algorithm is identical to the current optimal rate matching algorithm using parameters below

Nc=Ncs
Ni=Nis
(a,b)=(2,1) : exactly the same as conventional rate matching algorithm in [6,8].

This assures that the proposed algorithm gives optimal performance for convolutional coding.

3.2 Turbo code codes

For turbo codes, all code symbols are classified into three parts such as systematic code symbols (information symbols), the first parity symbols from constituent encoder 1, and the second parity symbols from constituent encoder 2, respectively. Rate matching algorithm blocks (RMB) for each part are the same as conventional algorithm except modifying number of bits per matching block and number of bits per matched block as follows.

- Systematic information symbols part: *Nc=Ncs/3, Ni=Ncs/3, (a,b)=(2,1)*
- Parity symbols part 1: *Nc=Ncs/3, Ni=(Nis- Ncs/3)/2, (a,b) =arbitrary constant*
- Parity symbols part 2: *Nc=Ncs/3, Ni=(Nis- Ncs/3)/2, (a,b) =arbitrary constant*

Parameter "b" is usually 1 and parameter "a" is only variable for considering hardware or software implementation complexity. In order to generate the same puncturing or repetition patterns for RMB2 and RMB3, it is sufficient to make parameter "a" have the same value for RMB2 and RMB3. Otherwise, it is sufficient to make parameter "a" have different value for RMB2 and RMB3, respectively. Therefore, there are no change of conventional rate matching algorithm except these parameters.

Table 1 shows examples of rate matching algorithm (puncturing) according to different parameters (a,b). Typically parameter "b" is 1 and parameter "a" is 2. However, arbitrary constant can be used for (a,b). Case 1, 3, and 5 give systematic puncturing with the same puncturing pattern and Case 2, 4, and 6 give systematic puncturing with different puncturing patterns.

Figure 3 shows implementation of the proposed rate matching scheme using single rate matching block for both convoultional codes and turbo codes in down-link. It is sufficient to use single conventional rate matching block with temporal registers to store (Nc, Ni, e, b, a). These parameters are updated for each RMB blocks. Also, the proposed scheme requires neither high speed clock nor additional memory for buffering. Furthermore a current rate matching algorithm block hardware can be used for the proposed scheme with slight modification for loading parameters. Software implementation is very simple such that it needs only one more subroutine call for rate matching with updated parameters.

Another merit of the proposed scheme is that it dose not produce time delay for rate matching. All symbols form channel encoder are transferred into rate matching block and pass out through rate matching block at the same time, with or without puncturing or repetition.

Table1. Examples of rate matching algorithm variation according to parameters (a,b).

Case	**RMB1**				**RMB2**				**RMB3**				
	Nc	**Ni**	**a**	**b**	**Nc**	**Ni**	**a**	**b**	**Nc**	**Ni**	**a**	**b**	
1	Ncs/3	Ncs/3	NR	NR	Ncs/3	(Nis-Ncs/3)/2	2	1	Ncs/3	(Nis-Ncs/3)/2	2	1	
2	Ncs/3	Ncs/3	NR	NR	Ncs/3	(Nis-Ncs/3)/2	2	1	Ncs/3	(Nis-Ncs/3)/2	5	1	
3	Ncs/3	Ncs/3	NR	NR	Ncs/3	(Nis-Ncs/3)/2	p	1	Ncs/3	(Nis-Ncs/3)/2	p	1	
4	Ncs/3	Ncs/3	NR	NR	Ncs/3	(Nis-Ncs/3)/2	p	1	Ncs/3	(Nis-Ncs/3)/2	q	1	
5	Ncs/3	Ncs/3	NR	NR	Ncs/3	(Nis-Ncs/3)/2	p	q	Ncs/3	(Nis-Ncs/3)/2	p	q	
6	Ncs/3	Ncs/3	NR	NR	Ncs/3	(Nis-Ncs/3)/2	p	q	Ncs/3	(Nis-Ncs/3)/2	r	s	

- *p,q,r,s:* arbitrary constant
- NR: Not related to the rate matching algorithm.
- RMB1, RMB2, RMB3: Rate Matching Block 1, 2, 3.
- RMB1: Rate matching block for Systematic information symbols part (By passed)
- RMB2: Rate matching block for Parity symbol 1
- RMB3: Rate matching block for Parity symbol 2

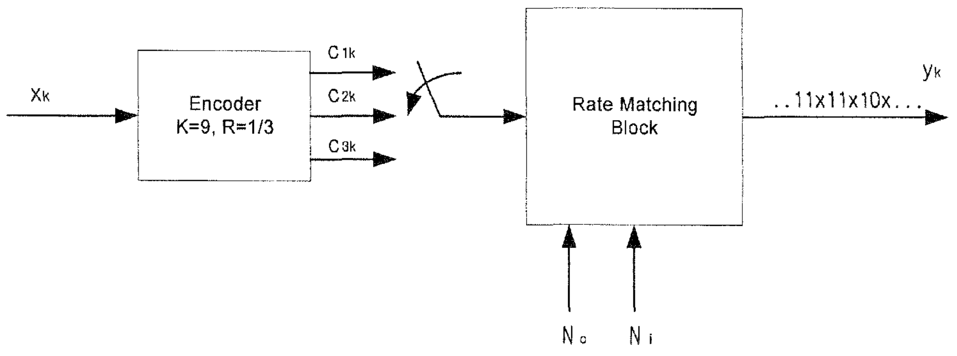


Fig. 1. Conventional rate matching scheme for convolutional codes (Down-link).

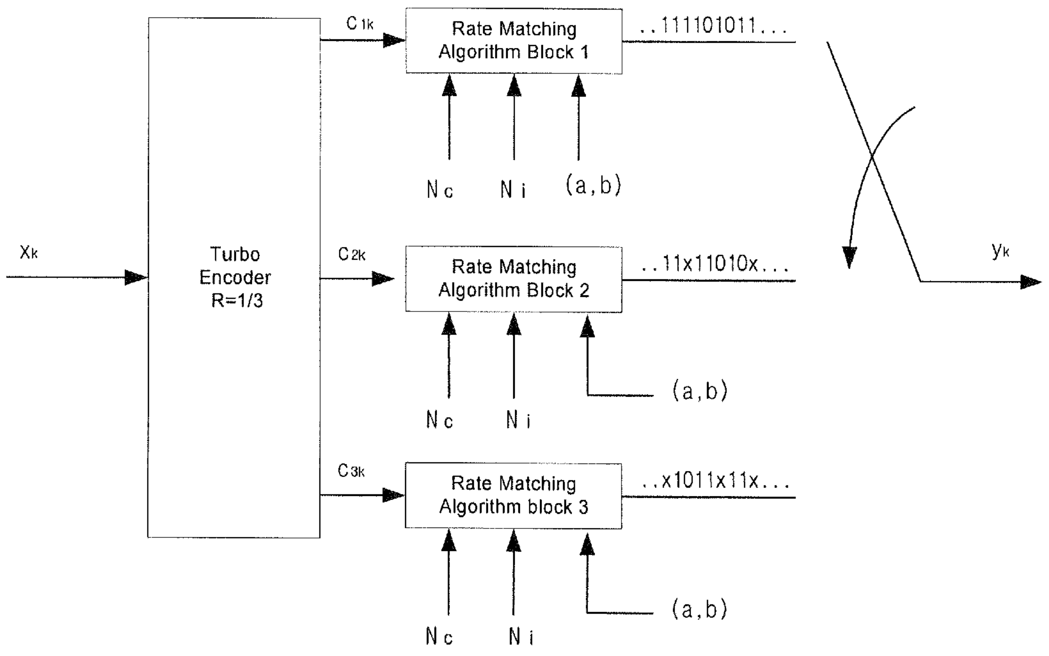

Xk
Turbo Encoder R=1/3
C1k
C2k
C3k
Rate Matching Algorithm Block 1
Rate Matching Algorithm Block 2
Rate Matching Algorithm block 3
..111101011...
..11x11010x...
..x1011x11x...
Nc
Ni
(a,b)
yk

Fig. 2. A conceptual block diagram of the proposed rate matching scheme for both convoultional codes and turbo codes (Down-link).

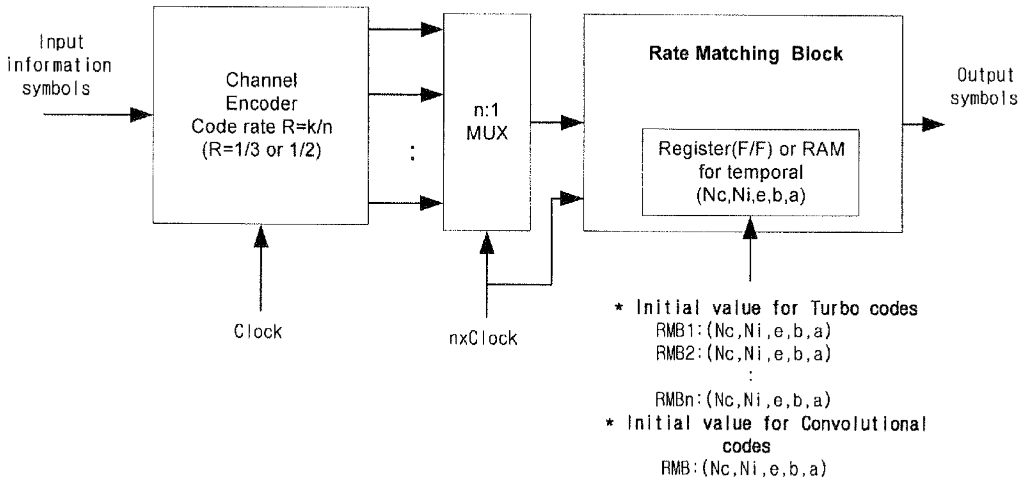

Input information symbols
Channel Encoder Code rate R=k/n (R=1/3 or 1/2)
n:1 MUX
Rate Matching Block
Register(F/F) or RAM for temporal (Nc,Ni,e,b,a)
Output symbols
Clock
nxClock
* Initial value for Turbo codes
RMB1:(Nc,Ni,e,b,a)
RMB2:(Nc,Ni,e,b,a)
RMBn:(Nc,Ni,e,b,a)
* Initial value for Convolutional codes
RMB:(Nc,Ni,e,b,a)

Fig. 3. Implementation of the proposed rate matching scheme using single rate matching block for both convoultional codes and turbo codes (Down-link).

4. Rate matching for up-link

Figure 4 illustrates conventional rate matching algorithm for up-link. Rate matching algorithm operates after radio frame segmentation so it is difficult to separate the flow containing systematic bits and the two flows containing parity bits (in the case of R=1/3). Operation of rate matching block is identical to that of down-link. To solve this problem, we proposed a new rate matching scheme in Figure 5. Note that there is no difference between up-link and down-link except the existence of de-multiplexing in case of up-link. The key idea is that de-multiplexing can separate the three flows containing systematic information, parity bits part 1, and parity bits part 2 one another. This is done due to the good property of MIL 1st interleaver for transmission time interval of 10, 20, 40, 80msec. So, the proposed rate matching algorithm in up-link is exactly the same as in down-link.

4.1 Convolutional codes

For convolutional code, the proposed algorithm is identical to the current optimal rate matching algorithm using parameters below

$N_c=N_{cs}$
$N_i=N_{is}$

This assures that the proposed algorithm gives optimal performance for convolutional coding.

4.2 Turbo code codes

For turbo codes, all code symbols are classified into three parts such as systematic code symbols (information symbols), the first parity symbols from constituent encoder 1, and the second parity symbols from constituent encoder 2. Rate matching algorithm blocks (RMB) for each part are the same as conventional algorithm except modifying number of bits per matching block and number of bits per matched block as follows

- Systematic information symbols part: $N_c=N_{cs}/3$, $N_i=N_{cs}/3$, $(a,b)=(2,1)$
- Parity symbols part 1: $N_c=N_{cs}/3$, $N_i=(N_{is}-N_{cs}/3)/2$, (a,b) =arbitrary constant
- Parity symbols part 2: $N_c=N_{cs}/3$, $N_i=(N_{is}-N_{cs}/3)/2$, (a,b) =arbitrary constant

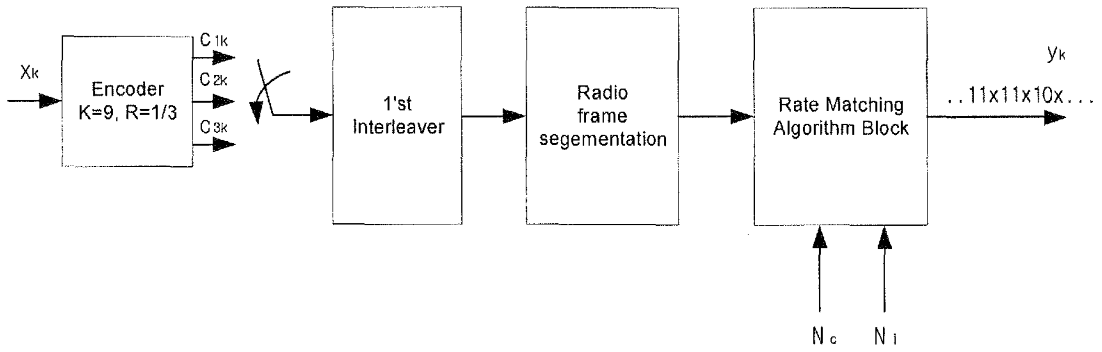
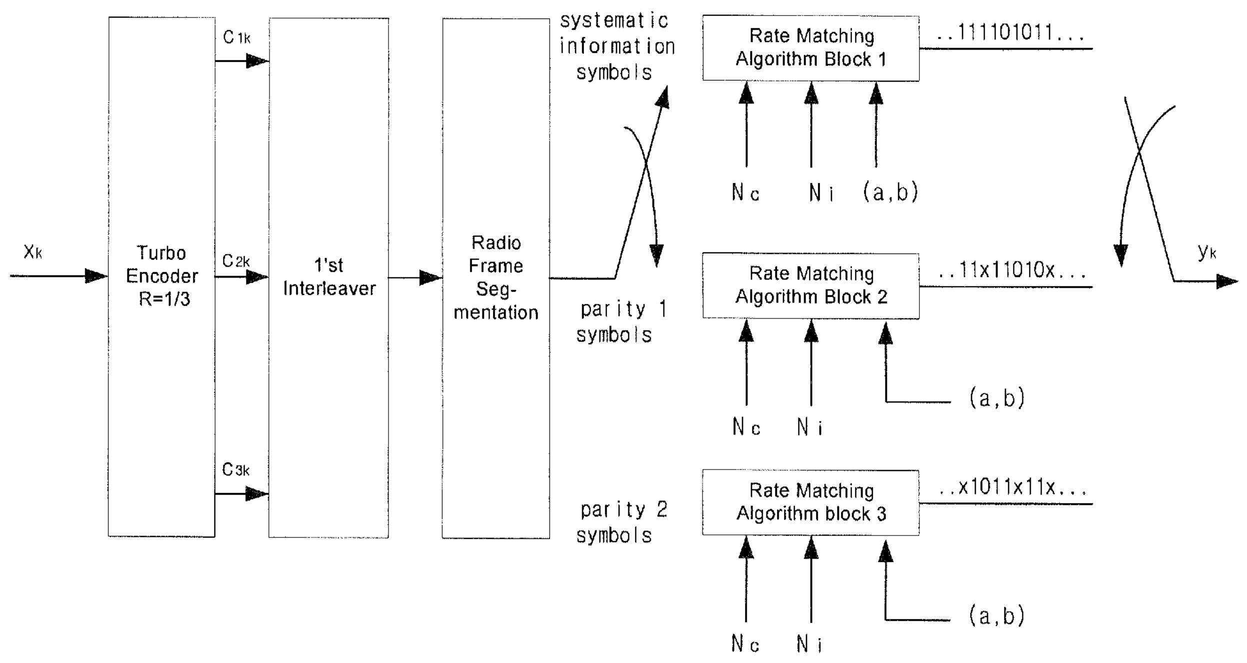


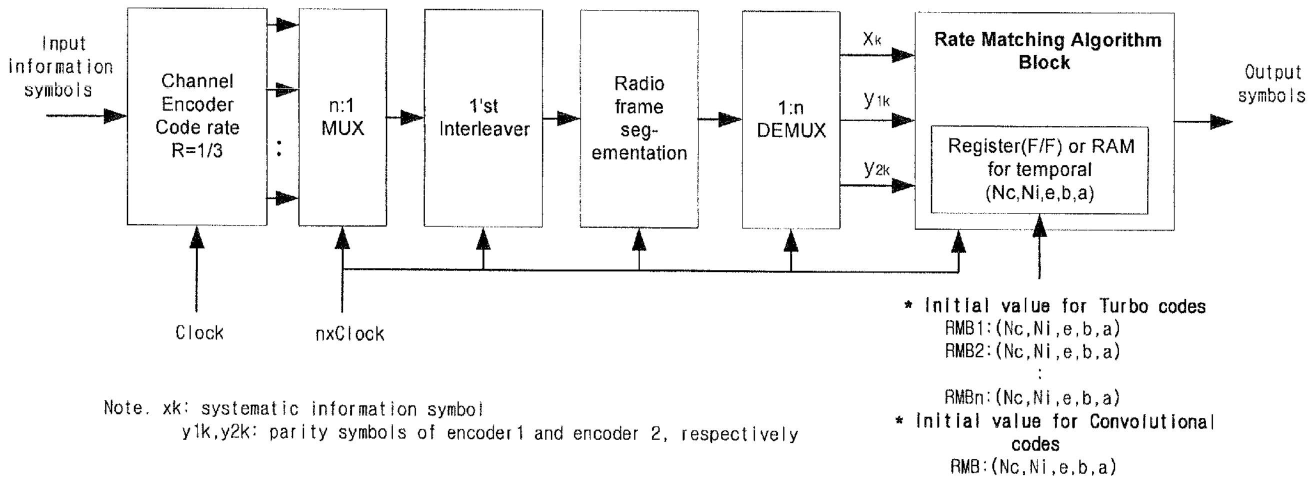
Fig. 4. Conventional rate matching scheme for convolutional codes (Up-link).

APLNDC-WH-A 0000010713



Xk
Turbo Encoder R=1/3
C1k
C2k
C3k
1'st Interleaver
Radio Frame Seg-mentation
systematic information symbols
parity 1 symbols
parity 2 symbols
Rate Matching Algorithm Block 1
Rate Matching Algorithm Block 2
Rate Matching Algorithm block 3
..111101011...
..11x11010x...
..x1011x11x...
Nc Ni (a,b)
yk

Fig. 5. A conceptual block diagram of the proposed rate matching scheme for both convoultional codes and turbo codes (Up-link).



Input information symbols
Channel Encoder Code rate R=1/3
n:1 MUX
1'st Interleaver
Radio frame seg-ementation
1:n DEMUX
Xk
y1k
y2k
Rate Matching Algorithm Block
Register(F/F) or RAM for temporal (Nc,Ni,e,b,a)
Output symbols
Clock
nxClock
Note. xk: systematic information symbol
y1k,y2k: parity symbols of encoder1 and encoder 2, respectively
* Initial value for Turbo codes
RMB1:(Nc,Ni,e,b,a)
RMB2:(Nc,Ni,e,b,a)
RMBn:(Nc,Ni,e,b,a)
* Initial value for Convolutional codes
RMB:(Nc,Ni,e,b,a)

Fig. 6. Implementation of the proposed rate matching scheme using single rate matching block for both convoultional codes and turbo codes (Up-link).

Figure 6 shows implementation of the proposed rate matching scheme using single rate matching block for both convoultional codes and turbo codes in up-link. It is sufficient to use single conventional rate matching block with temporal registers to store (Nc, Ni, e, b, a)

5. Simulation results

For turbo codes, the suggested simulation conditions announced at the 2nd AdHoc meetin g at Cheju are as follows

- Block sizes: 320, 321, 640, 641, 5120, 5119
- Puncturing rates: 5%, 10%, 15%, 20%, and 1/2 code (=33%)
- Decoding algorithm: Log MAP decoder
- Turbo interleaver: PIL
- Number of iterations: 12
- BER: >10E-6
- Number of frame error: greater than 100
- Channel model: AWGN
- Tail bits puncturing
- Algorithms: SEC, Siemens ,LGIC, and Fujitsu

In the following figures, SEC_T1 and SEC_T2 mean the following parameter setting for RMB2 and RMB3, respectively.

	RMB1	RMB2	RMB3
SEC_T1	Not used	(a,b)=(2,1)	(a,b)=(5,4)
SEC_T2	Not used	(a,b)=(2,1)	(a,b)=(3,1)

6. Conclusion

In this contribution, we proposed a unified rate matching algorithm for both turbo codes a nd convolutional codes and both down-link and up-link. The proposed rate matching algori thm does not change the conventional rate matching algorithm proposed by Siemens. It o nly controls parameter for the conventional rate matching algorithm. The conclusions are as follows

- The proposed rate matching algorithm can support both turbo code and convolutional codes and both up-link and down-link.
- The rate matching algorithm is optimal for both convolutional coding and turbo coding.
- For up-link, it has been shown that the output symbols from the 1'st interleaver can b e classified into two parts implicitly such as systematic information symbol part and parity symbols (from constituent encoder 1 or 2 part, respectively.
- For rate matching of turbo codes, LGIC's algorithm and Fujitsu's old algorithm are sp ecial cases of the proposed algorithm.
- Implementation of the proposed rate matching algorithm require neither increase in ha rdware complexity nor software complexity. It can be easily implemented by using convent ional rate matching algorithm block with parameter configuration.
- The proposed rate matching algorithm provides a single structure of rate matching for both down-link and up-link.
- The proposed rate matching algorithm does not require any increase of buffer memor

y in both sides of transmitter and receiver. Also it does not require processing delay for storing output symbols from channel encoder for reordering.

7. References

[1] "Comparison of downlink puncturing algorithms", LGIC, TSGR1#5(99)654(1999-06).
[2] "Puncturing algorithm for turbo code", LGIC, TSGR1#4(99) 338 (1999-04)
[3] "Further results of rate matching algorithm for downlink", Fujitsu,TSGR1#5(99)698(1999-06).
[4] "Tail bits and puncturing of rate 1/2 turbo coding", Fujitsu, TSGR1#5(99)665(1999-06).
[5] "Optimised puncturing schemes for turbo coding", Fujitsu, TSGR1#4(99)388(1999-04).
[6] "Optimised rate matching after interleaving", Siemens, TSGR1#3(99)203.
[7] "Proposal for rate matching for turbo codes", Nortel Networks, TSGR1#4(99)467 (1999-04)
[8] 3GPP TSG RAN WG1 Multiplexing and channel coding (FDD) TS 25.212 v2.0.0 (1999-06)

Contact inform:
Kimmingu@samsung.co.kr
Bjkim@telecom.samsung.co.kr

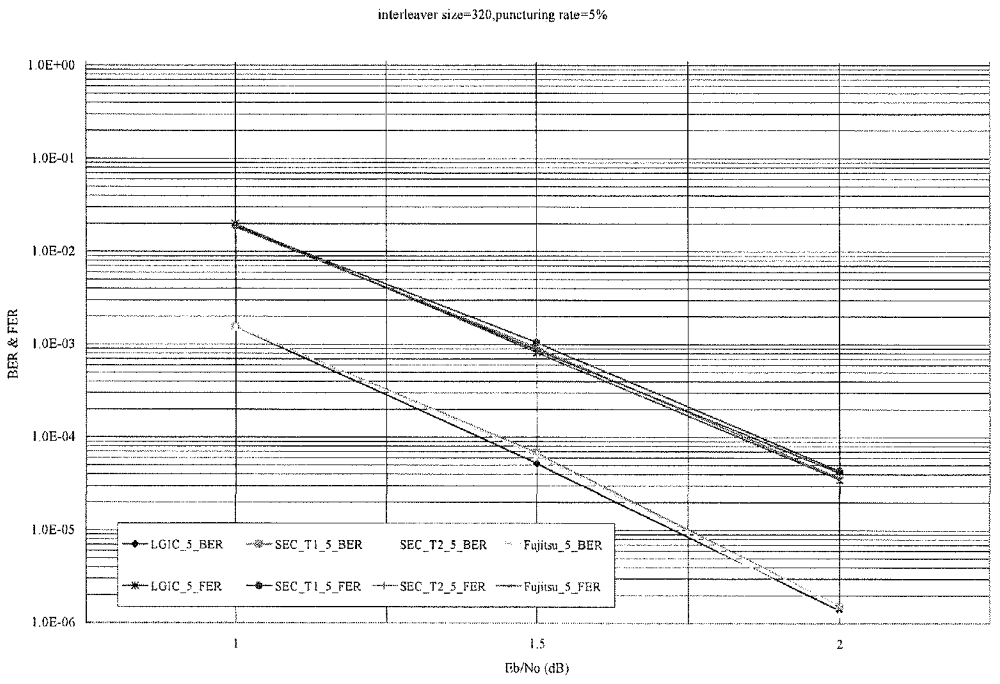


Fig. 7. interleaver size=320,puncturing rate=5%.

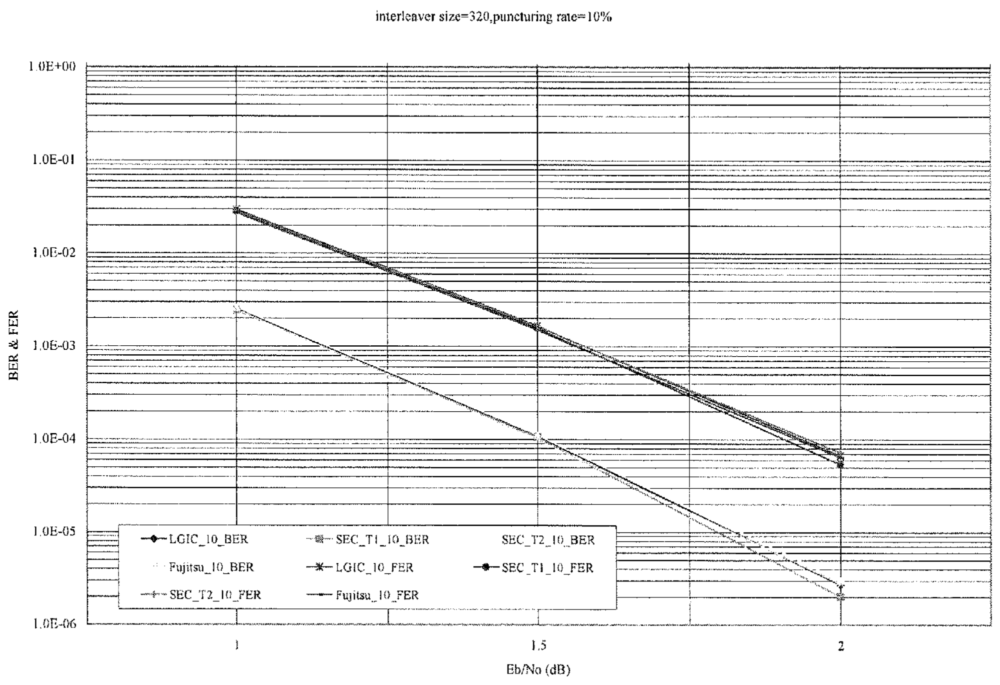


Fig. 8. interleaver size=320,puncturing rate=10%.

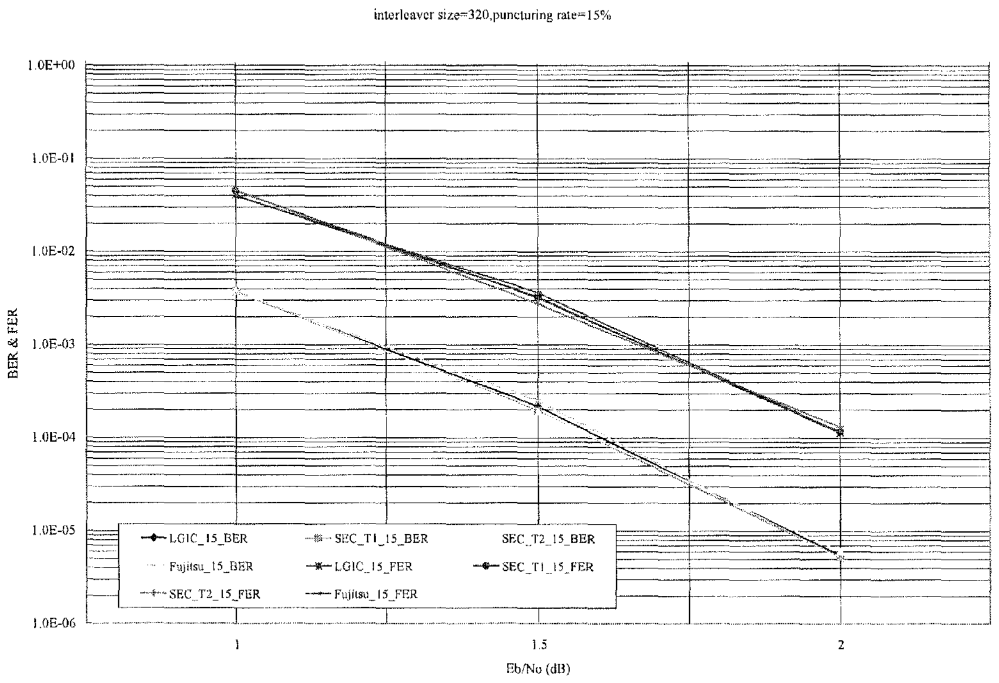


Fig. 9. interleaver size=320,puncturing rate=15%.

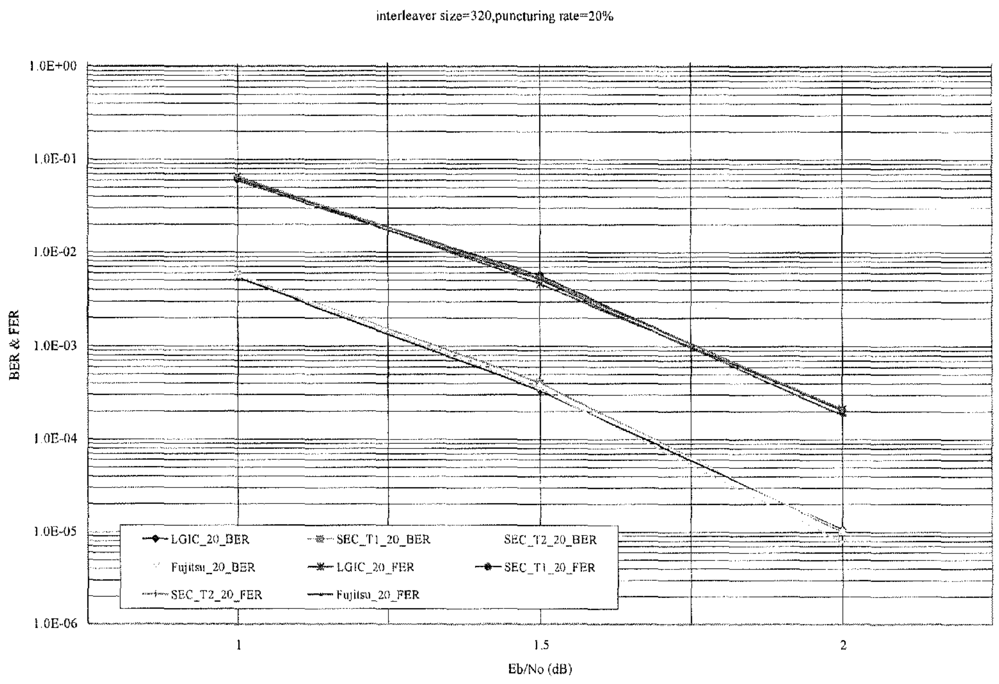


Fig. 10. interleaver size=320,puncturing rate=20%.

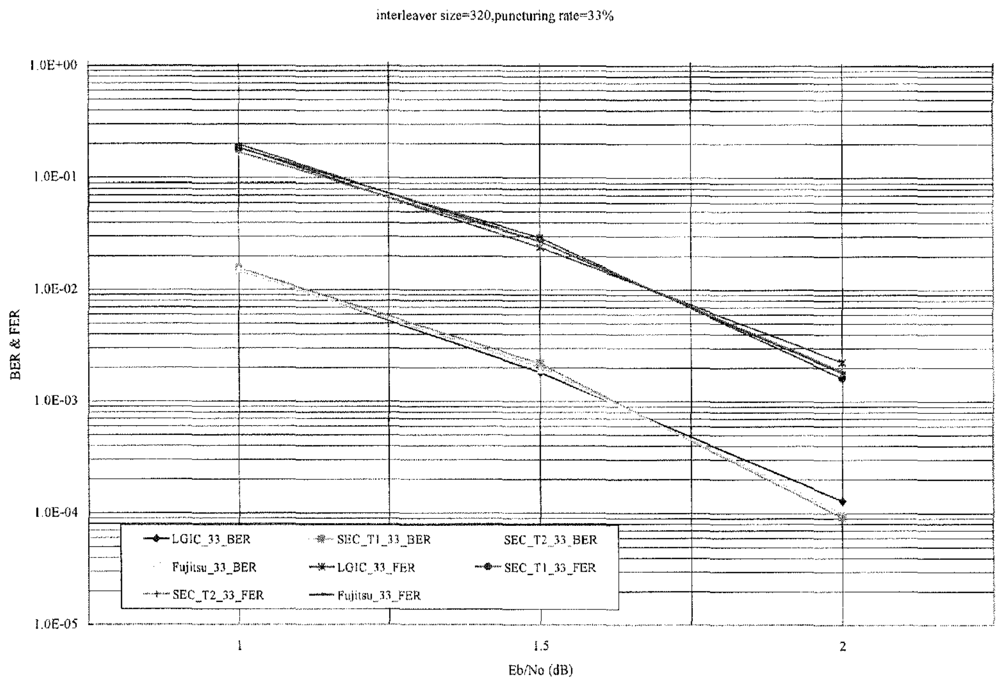


Fig. 11. interleaver size=320,puncturing rate=33%.

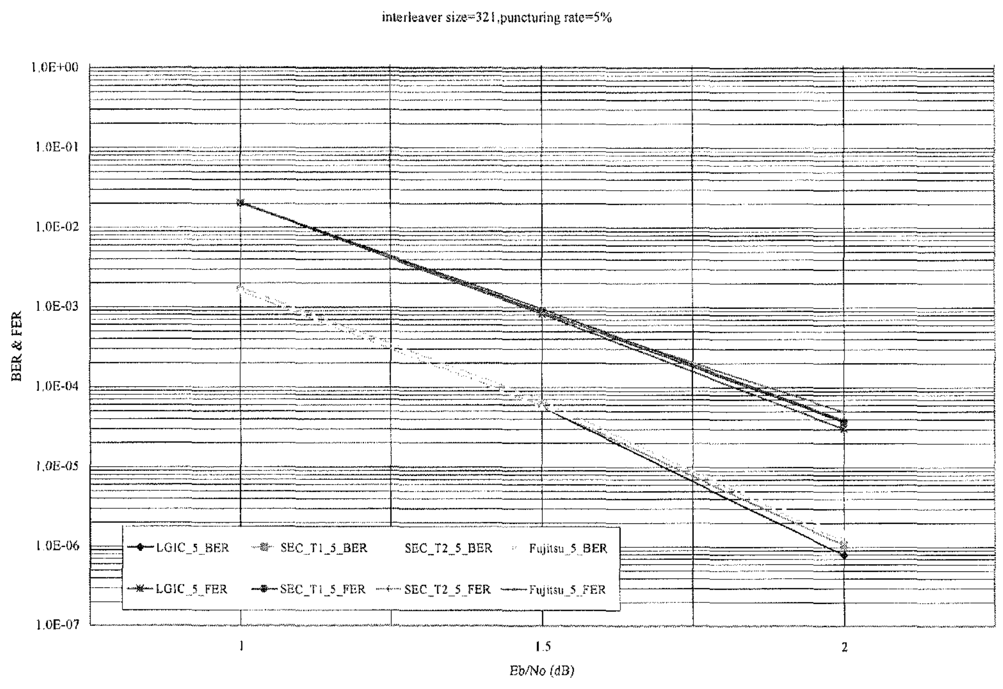


Fig. 12. interleaver size=321,puncturing rate=5%.

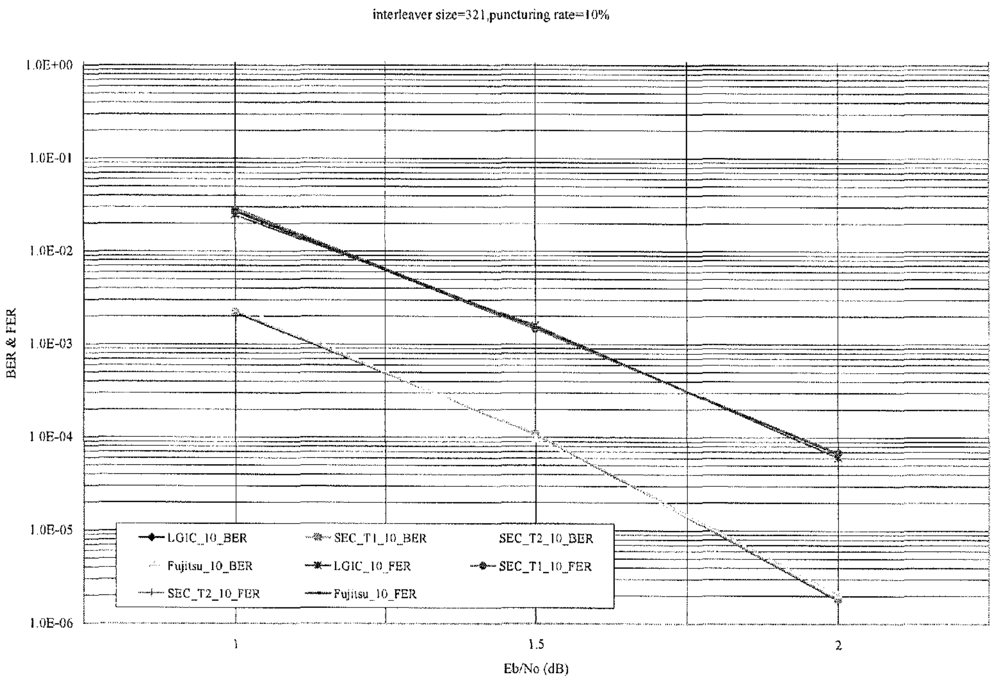


Fig. 13. interleaver size=321,puncturing rate=10%.

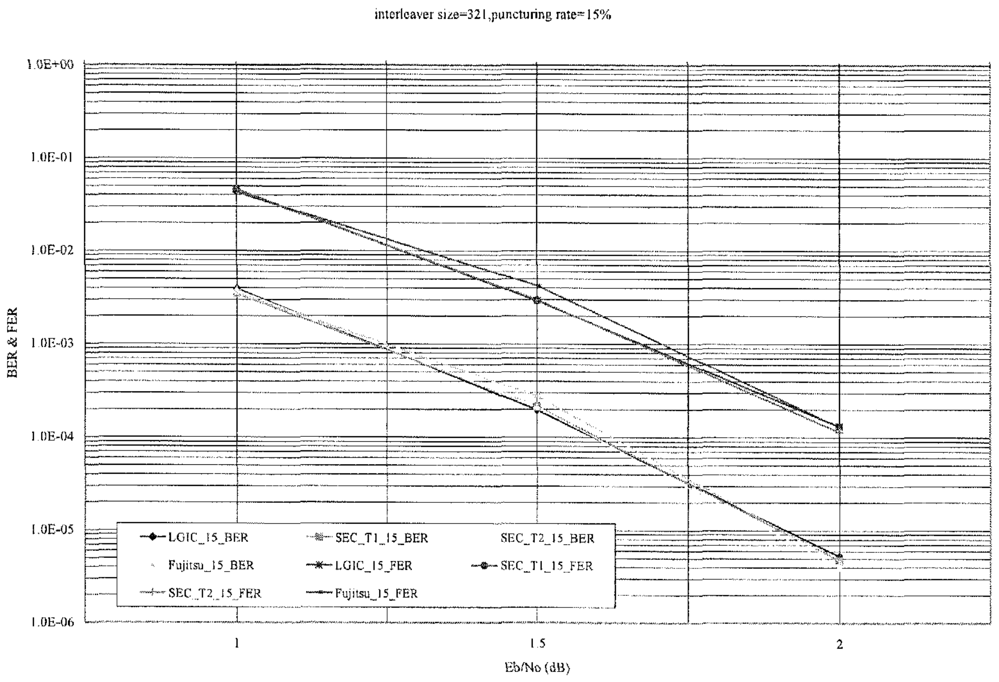


Fig. 14. interleaver size=321,puncturing rate=15%.

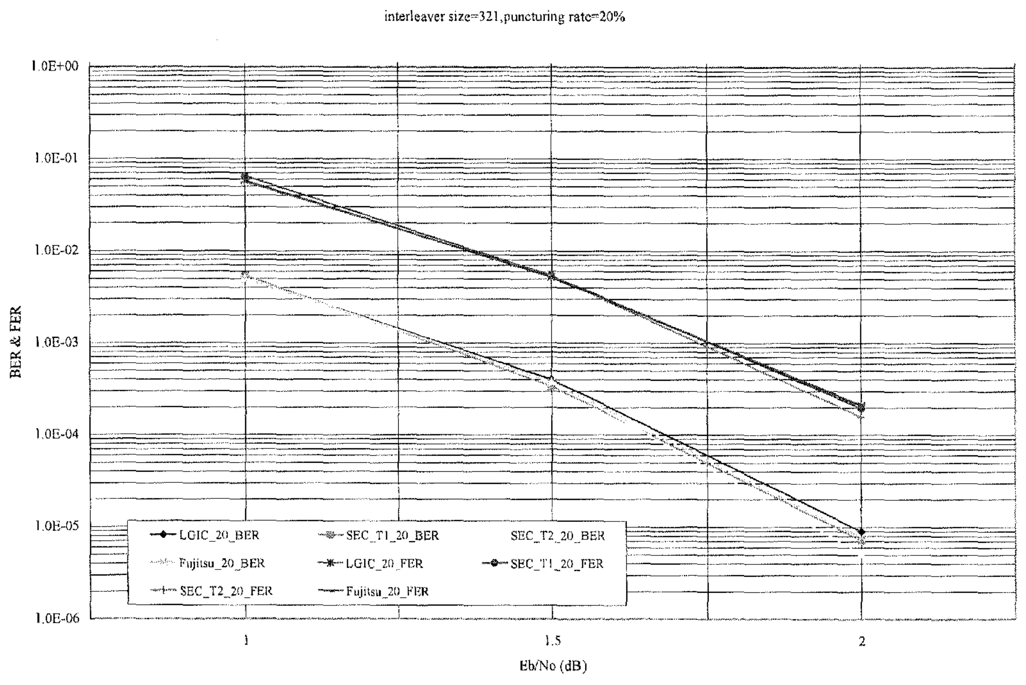


Fig. 15. interleaver size=321,puncturing rate=20%.

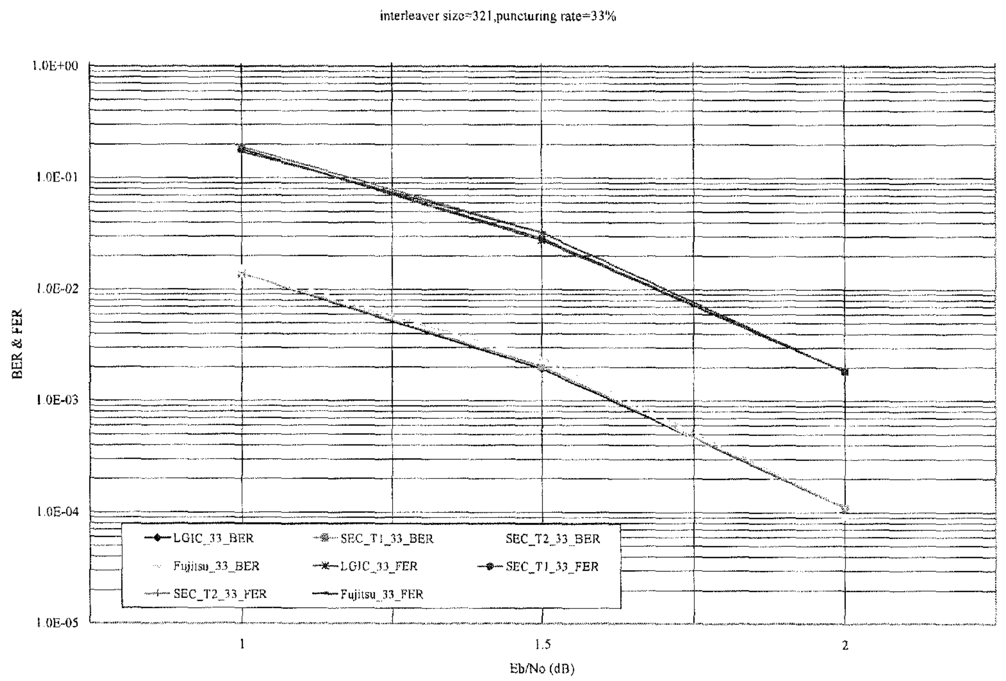


Fig. 16. interleaver size=321,puncturing rate=33%.

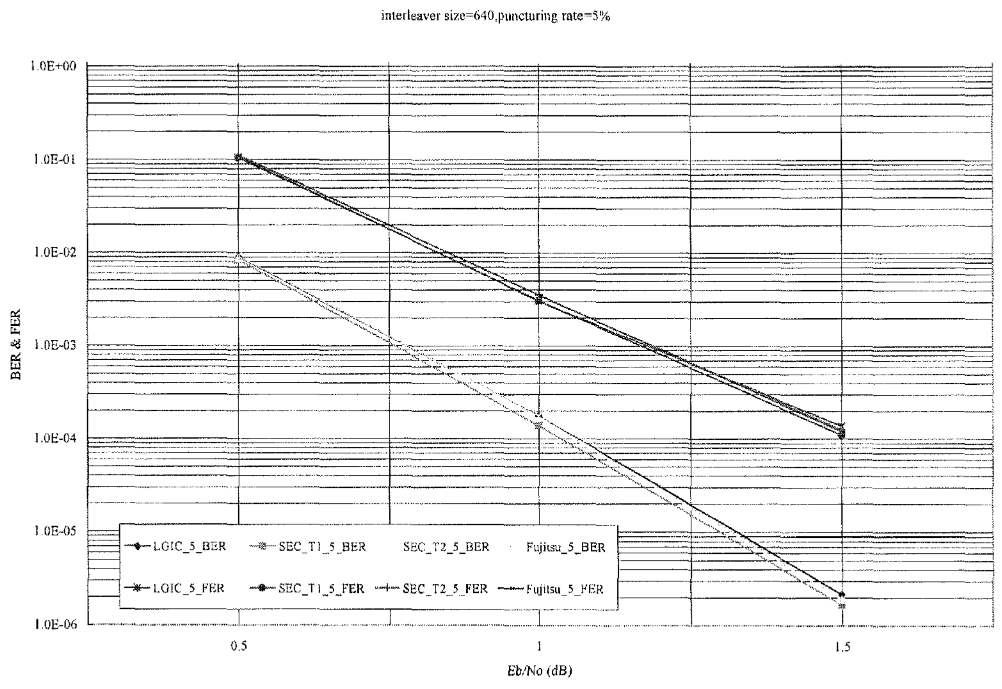


Fig. 17. interleaver size=640,puncturing rate=5%.

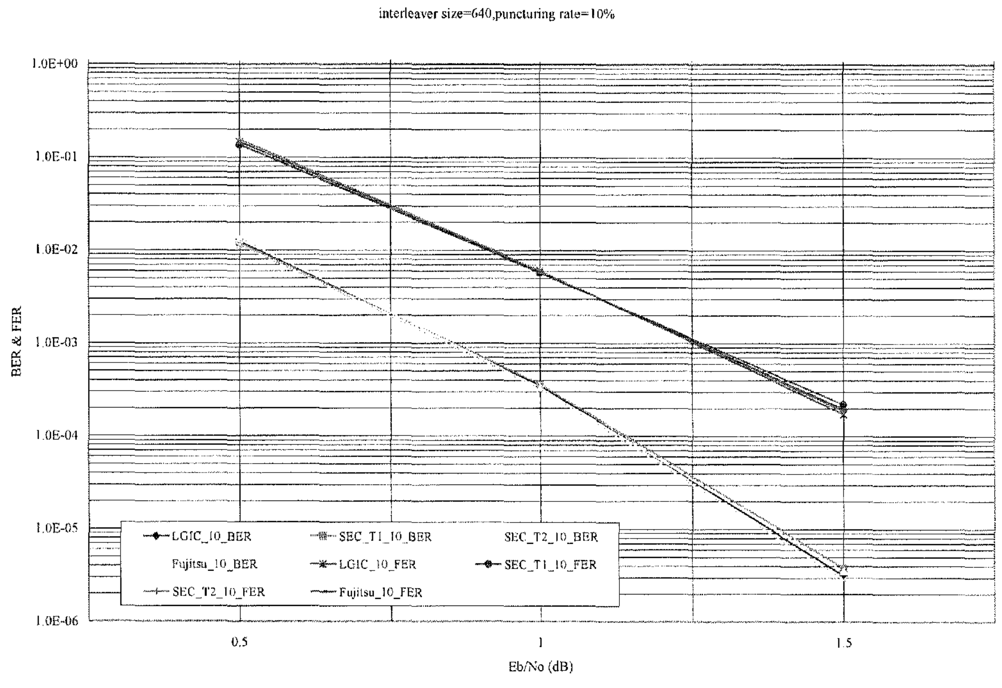


Fig. 18. interleaver size=640,puncturing rate=10%.

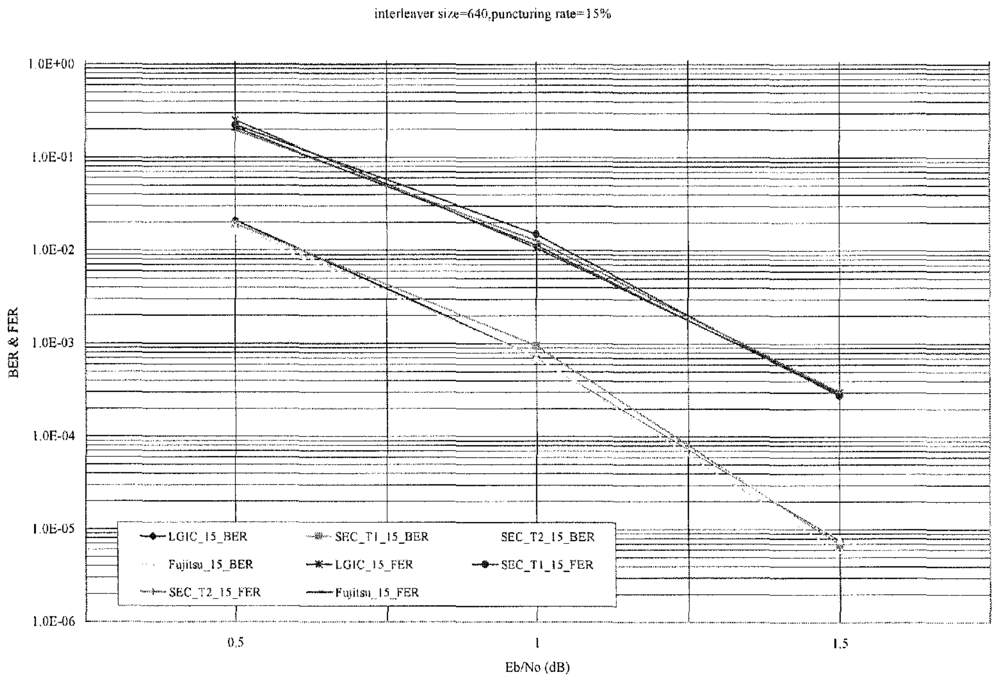


Fig. 19. interleaver size=640,puncturing rate=15%.

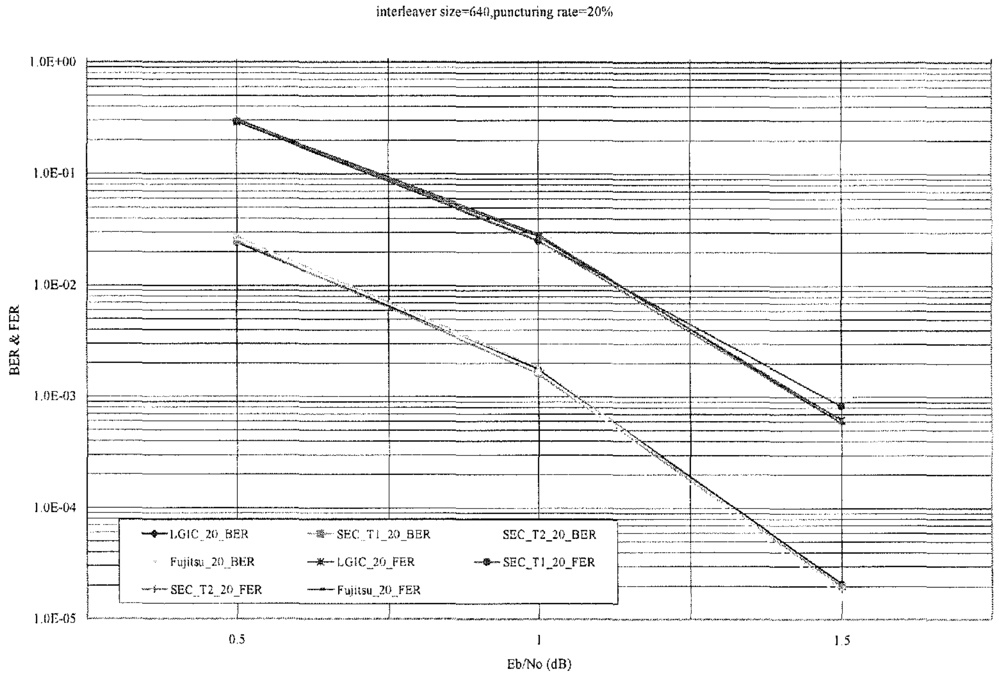


Fig. 20. interleaver size=640,puncturing rate=20%.

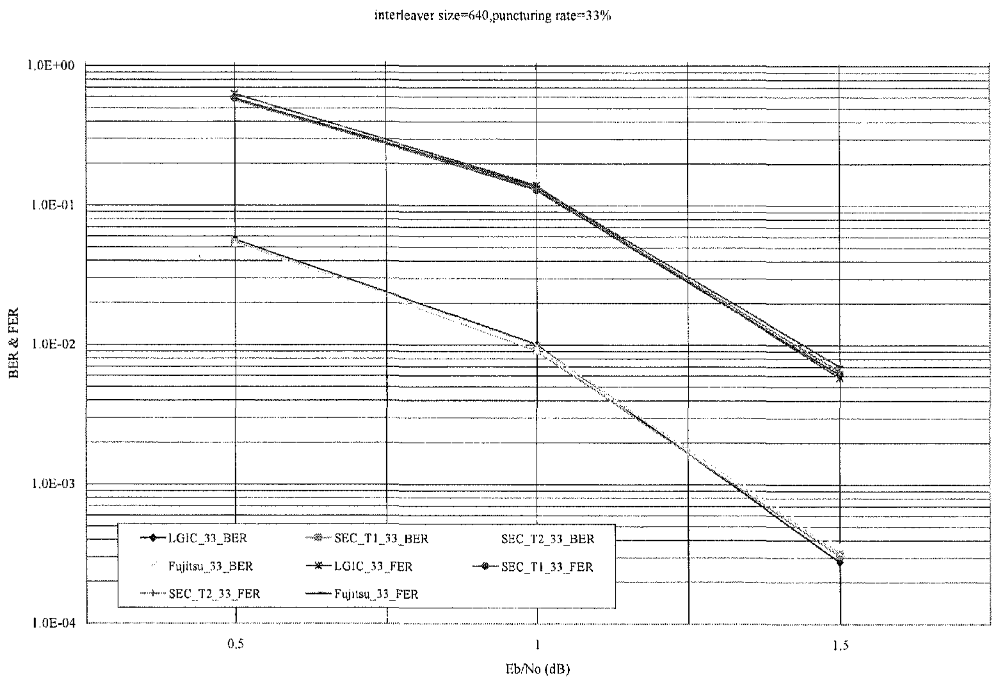


Fig. 21. interleaver size=640,puncturing rate=33%.

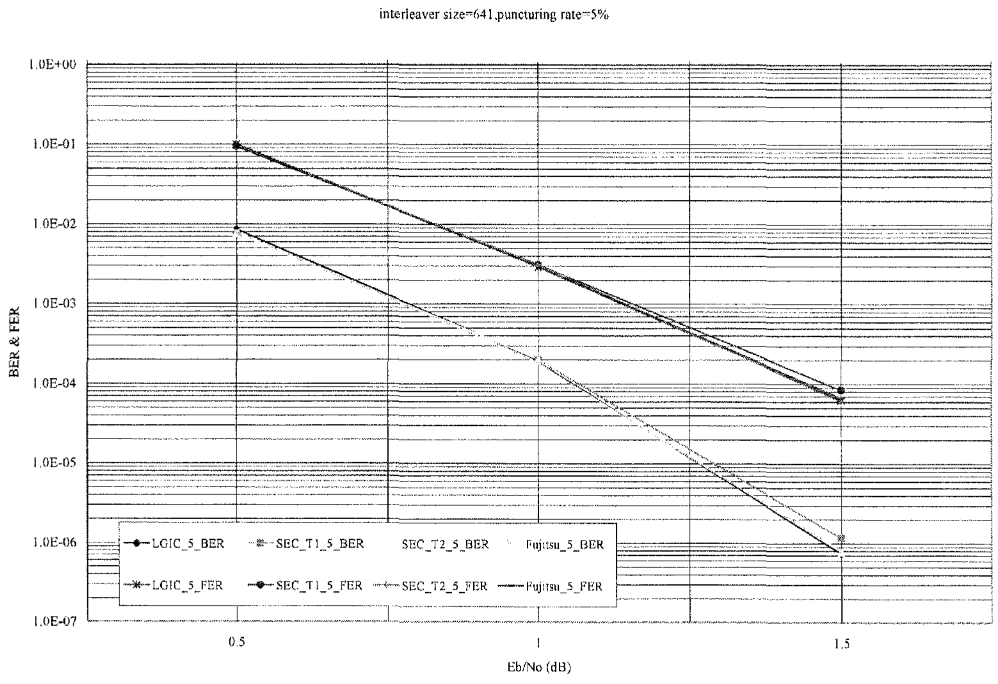


Fig. 22. interleaver size=641,puncturing rate=5%.

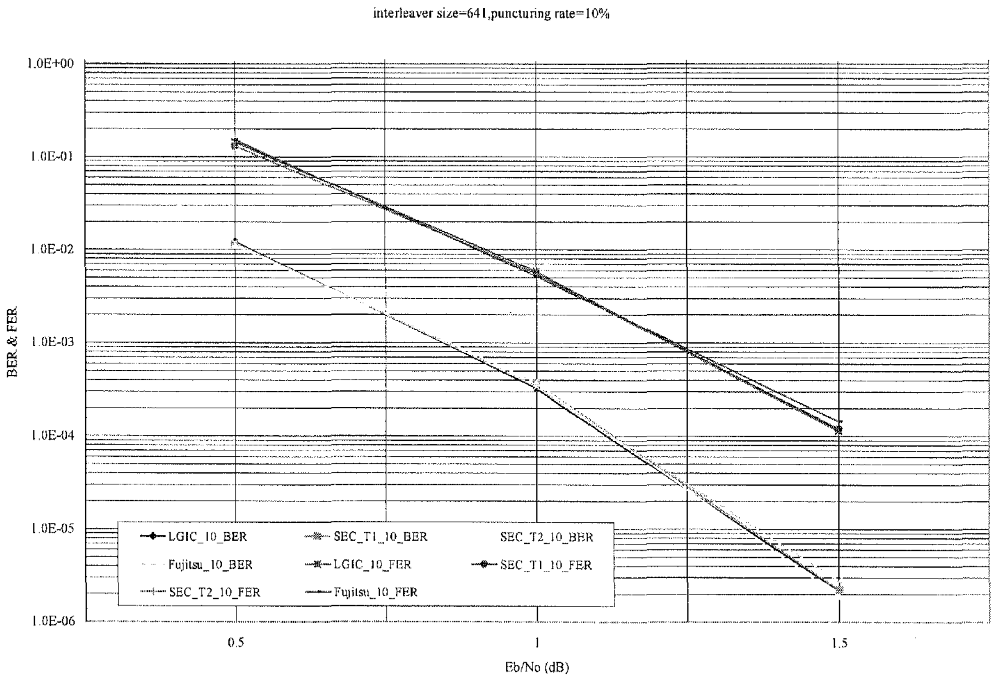


Fig. 23. interleaver size=641,puncturing rate=10%.

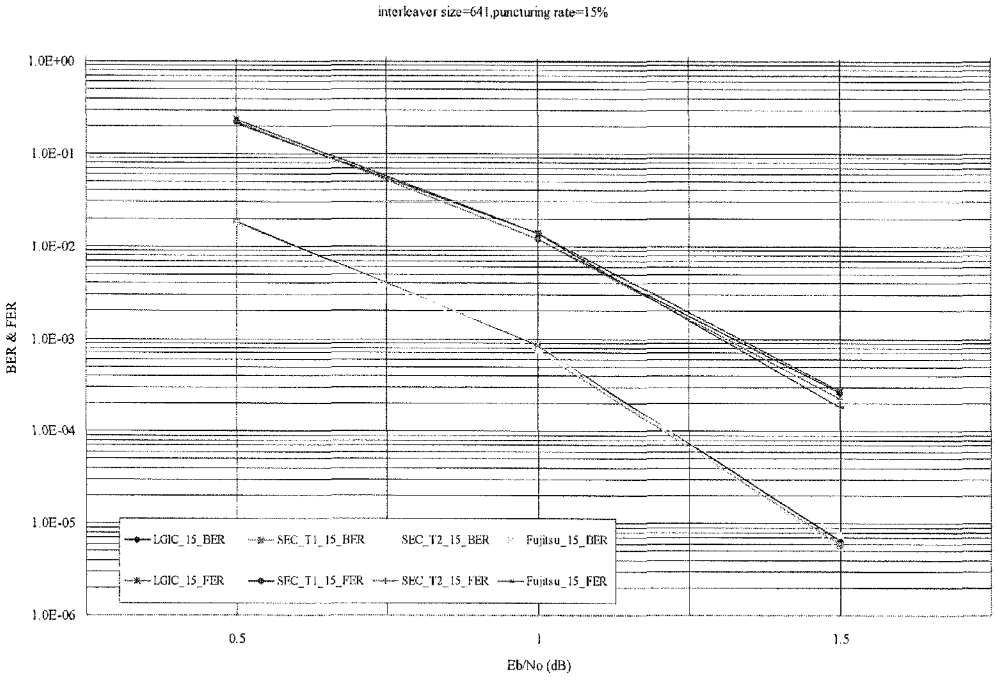


Fig. 24. interleaver size=641,puncturing rate=15%.

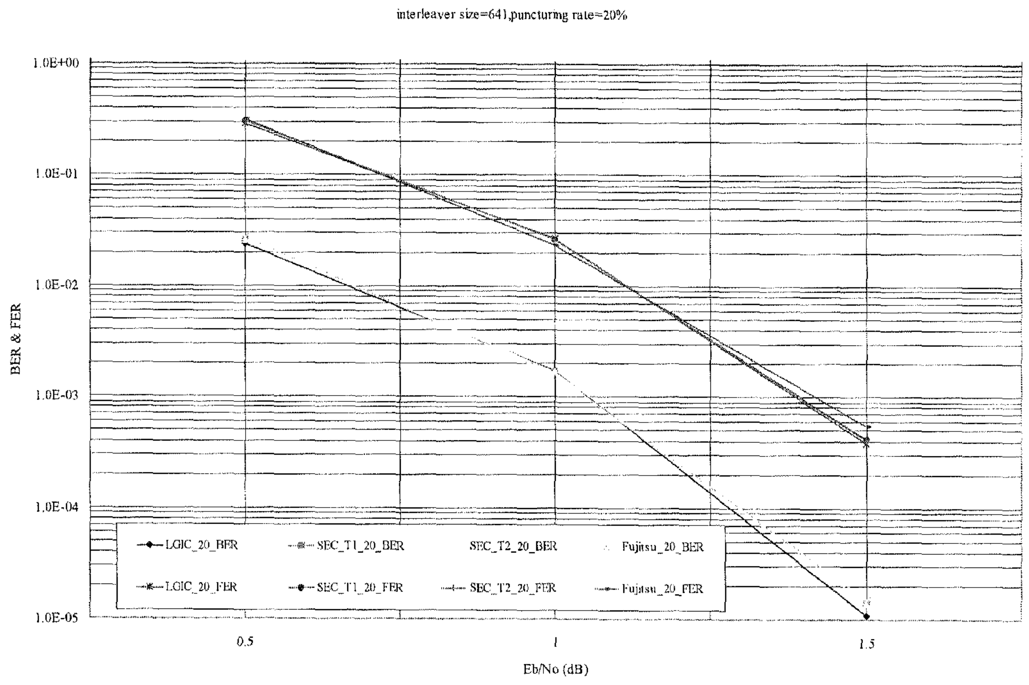


Fig. 25. interleaver size=641,puncturing rate=20%.

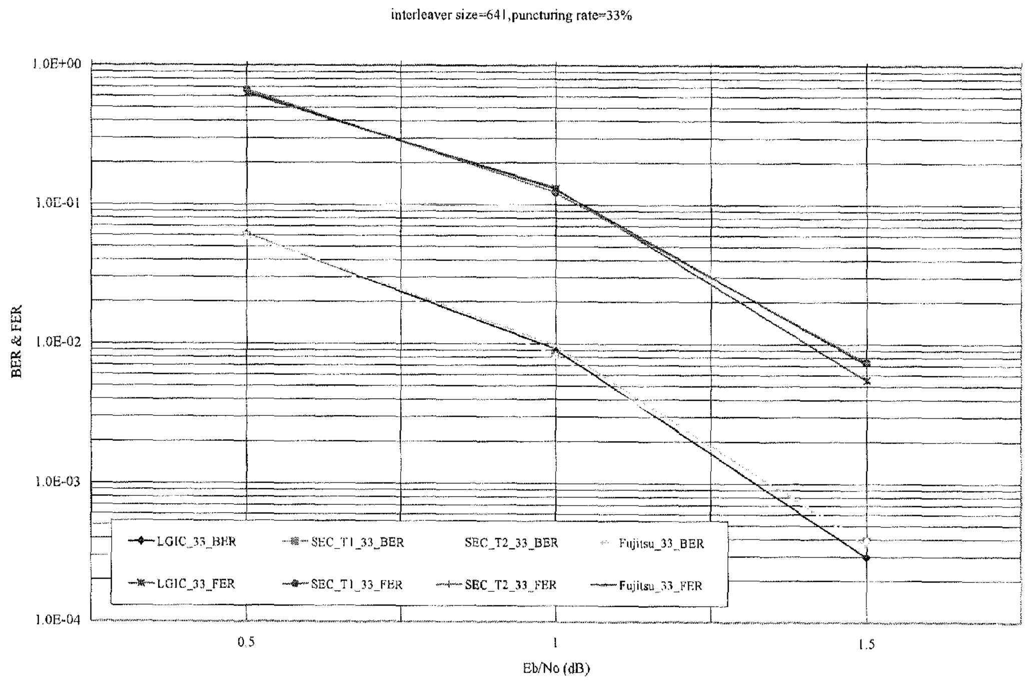


Fig. 26. interleaver size=641,puncturing rate=33%.