

# Bit-Level Extrinsic Information Exchange Method for Double-Binary Turbo Codes

Ji-Hoon Kim, *Student Member, IEEE*, and In-Cheol Park, *Senior Member, IEEE*

**Abstract**—Nonbinary turbo codes have many advantages over single-binary turbo codes, but their decoder implementations require much more memory, particularly for storing symbolic extrinsic information to be exchanged between two soft-input–soft-output (SISO) decoders. To reduce the memory size required for double-binary turbo decoding, this paper presents a new method to convert symbolic extrinsic information to bit-level information and vice versa. By exchanging bit-level extrinsic information, the number of extrinsic information values to be exchanged in double-binary turbo decoding is reduced to the same amount as that in single-binary turbo decoding. A double-binary turbo decoder is designed for the WiMAX standard to verify the proposed method, which reduces the total memory size by 20%.

**Index Terms**—Double-binary turbo codes, extrinsic information, maximum a posteriori algorithm, turbo decoding.

## I. INTRODUCTION

THE TURBO codes introduced in 1993 are among the most powerful forward error correction codes and provide near-optimal performance approaching the Shannon limit [1]. Recently, nonbinary turbo codes have received a great deal of attention and are adopted in several mobile radio systems such as DVB-RCS and IEEE 802.16 standard (WiMAX) [2], as they can offer many advantages over single-binary turbo codes [3].

Compared with classical single-binary turbo codes, the size of extrinsic information memory becomes large since the number of extrinsic information values to be exchanged between two soft-input–soft-output (SISO) decoders increases significantly in nonbinary turbo decoding. Although nonuniform quantization adopted in single-binary turbo decoding can be applied to reduce the extrinsic memory size [4], the large memory size incurred by the increased number of extrinsic information values is still a major obstacle in implementing a nonbinary turbo decoder [5].

In order to reduce the memory size needed in double-binary turbo decoding, this paper presents a method to convert symbolic extrinsic information to bit-level information and vice versa. By converting symbol-level extrinsic information values into bit-level values, the number of extrinsic information values to be exchanged can be reduced. To verify the proposed

method, a double-binary turbo decoder is implemented for the WiMAX standard.

## II. EXTRINSIC INFORMATION IN DOUBLE-BINARY TURBO CODES

Focusing on nonbinary turbo decoding, we describe in this section the conventional symbol-level extrinsic information and implementation issues.

### A. Symbol-Level Extrinsic Information in Double-Binary Turbo Codes

In an  $m$ -ary turbo code where a symbol is represented in  $m$  bits, the number of possible symbol-level extrinsic information values is  $2^m - 1$  [6]. Since the value of  $m$  is two for double-binary turbo codes, three symbol-level extrinsic information values are defined as follows:

$$L_e^z = \ln \frac{p(u_k = z)}{p(u_k = 00)} \Rightarrow p(u_k = z) = p(u_k = 00) \cdot \exp[L_e^z] \quad (1)$$

where  $z$  belongs to  $\phi = \{01, 10, 11\}$ ,  $u_k$  is the input symbol consisting of two bits, and  $p(\cdot)$  means the probability. The extrinsic information is exchanged iteratively between two SISO decoders during the whole decoding process. As indicated in (1), the extrinsic information in double-binary turbo codes is defined as the ratio of two input symbols, each of which consists of two bits. To store the increased number of extrinsic information values, a large memory is needed in implementing a nonbinary turbo decoder.

### B. Memory Requirement in Double-Binary Turbo Decoder

A typical turbo decoder is based on the time-multiplex architecture that contains only one SISO decoder, one interleaver, and one extrinsic memory [7]. For the SISO decoder, the sliding-window technique, is widely used to reduce the memory needed to store metric values [5]. To avoid the complex dummy metric calculation required in the sliding-window technique, we can adopt the border memory in nonbinary turbo decoding [5]. However, the extrinsic information memory cannot be reduced even if the sliding-window technique is employed, and the size is determined by the largest frame size that is 2400 pairs in the WiMAX standard [2].

The experimental environment for the WiMAX turbo code is indicated in Table I, where  $(q, f)$  denotes a quantization scheme that uses  $q$  bits in total and  $f$  bits to represent the fractional part. Taking into account the quantization scheme indicated in Table I, the memory size required for a double-binary SISO decoder is summarized in Table II, and that for storing extrinsic information values is tabulated in Table III. It is crucial to reduce the extrinsic information memory, as the extrinsic information

Manuscript received October 07, 2007; revised January 12, 2008 and June 05, 2008. Current version published January 16, 2009. This work was supported by the IC Design Education Center (IDEC). This paper was recommended by Associate Editor G. M. Maggio.

The authors are with the School of Electrical Engineering and Computer Science, KAIST, Daejeon 305-701, Korea (e-mail: sckid@eeinfo.kaist.ac.kr; icpark@ee.kaist.ac.kr).

Digital Object Identifier 10.1109/TCSII.2008.2008523

TABLE I  
SIMULATION ENVIRONMENT

<b>SISO Algorithm</b>	Max-log-MAP
<b>Window Size</b>	32
<b>Quantization</b>	Received input: (4, 2) Branch Metric & State Metrics : (10, 2) Extrinsic Information : (8, 2) LLR value : (11, 2)

TABLE II  
MEMORY CONFIGURATION FOR ONE SISO DECODER

<b>Forward Metric Memory</b>	2 banks, 32*(10*7) bits/bank 4480 bits
<b>Branch Metric Memory</b>	2 banks, 32*(10*8) bits/bank 5120 bits
<b>Border Metric Memory</b>	[(2400/32)-1]*(10*7) bits 5180 bits

TABLE III  
MEMORY CONFIGURATION FOR THE EXTRINSIC INFORMATION

	$L_e^{01}$	$L_e^{10}$	$L_e^{11}$
<b>Extrinsic Info. Memory</b>	2 banks, 8*2400bits/bank 38400 bits	2 banks, 8*2400bits/bank 38400 bits	2 banks, 8*2400bits/bank 38400 bits
<b>Total</b>	115200 bits		

memory is much bigger than the memory required in SISO decoding, as indicated in Tables II and III.

In higher order nonbinary turbo decoders such as triple-binary ( $m = 3$ ) and quaternary ( $m = 4$ ) turbo decoders [6], [8], the extrinsic information memory becomes much bigger, since the number of extrinsic information values to be stored is proportional to  $2^m - 1$ .

### III. PROPOSED BIT-LEVEL EXTRINSIC INFORMATION EXCHANGE

Here, we propose a new bit-level extrinsic information exchange method that can reduce the number of extrinsic values to be exchanged between two SISO decoders, as shown in Fig. 1. Specifically, we present two conversions—symbol-to-bit conversion and bit-to-symbol conversion of extrinsic information. With simple conversions, the number of values to be exchanged can be reduced from the number of possible symbols,  $2^m - 1$ , to the number of bits in a symbol,  $m$ , without inducing any modifications to conventional symbol-based double-binary SISO decoders. Therefore, the proposed method is effective in reducing the size of extrinsic information memory regardless of quantization.

#### A. Bit-Level Extrinsic Information for Double-Binary Turbo Codes

Like the symbol-level extrinsic information in (1), two bit-level extrinsic information values are defined as follows:

$$L_{be}^A = \ln \frac{p(A=1)}{p(A=0)} \Rightarrow p(A=1) = p(A=0) \cdot \exp[L_{be}^A]$$

$$L_{be}^B = \ln \frac{p(B=1)}{p(B=0)} \Rightarrow p(B=1) = p(B=0) \cdot \exp[L_{be}^B] \quad (2)$$

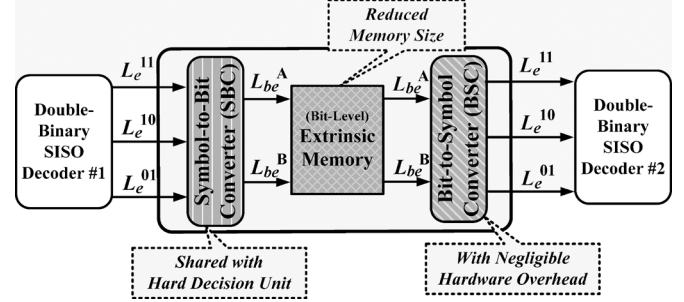


Fig. 1. Block diagram of the proposed bit-level extrinsic information exchange.

where the input symbol  $u_k$  consists of a pair of two bits,  $A$  and  $B$ , i.e.,  $u_k = AB$ . The bit-level probabilities in (2) can be derived from the symbol-level probabilities in (1), as described in the following:

$$p(A=0) = p(u_k = 00) + p(u_k = 01) \quad (3a)$$

$$p(A=1) = p(u_k = 10) + p(u_k = 11) \quad (3b)$$

$$p(B=0) = p(u_k = 00) + p(u_k = 10) \quad (3c)$$

$$p(B=1) = p(u_k = 01) + p(u_k = 11). \quad (3d)$$

Considering the basic property of the probabilities

$$p(A=0) + p(A=1) = p(A=0) \cdot (1 + \exp[L_{be}^A]) = 1 \quad (4a)$$

$$p(B=0) + p(B=1) = p(B=0) \cdot (1 + \exp[L_{be}^B]) = 1. \quad (4b)$$

Similarly, for the symbol-level probabilities

$$p(u_k = 00) + p(u_k = 01) + p(u_k = 10) + p(u_k = 11) = p(u_k = 00) \cdot (1 + \exp[L_e^{01}] + \exp[L_e^{10}] + \exp[L_e^{11}]) = 1. \quad (5)$$

From the aforementioned properties,  $p(u_k = 00)$  can be expressed as follows:

$$p(u_k = 00) = \frac{1}{(1 + \exp[L_e^{01}]) \cdot (1 + \exp[L_{be}^A])} = \frac{1}{(1 + \exp[L_e^{10}]) \cdot (1 + \exp[L_{be}^B])}. \quad (6)$$

Additionally, the following equation can be derived from (3):

$$p(u_k = 11) - p(u_k = 00) = p(B=1) - p(A=0) = p(A=1) - p(B=0). \quad (7)$$

Using (2), (4), and (7), we can derive

$$\exp[L_e^{11}] = \frac{1}{p(u_k = 00)} \cdot \left( \frac{\exp[L_{be}^A]}{1 + \exp[L_{be}^A]} - \frac{1}{1 + \exp[L_{be}^B]} \right) + 1 = \frac{1}{p(u_k = 00)} \cdot \left( \frac{\exp[L_{be}^B]}{1 + \exp[L_{be}^B]} - \frac{1}{1 + \exp[L_{be}^A]} \right) + 1. \quad (8)$$

By applying (6) to (8)

$$\begin{aligned} \exp [L_e^{11}] &= \exp [L_{be}^A] \cdot (1 + \exp [L_e^{01}]) - \exp [L_e^{10}] \\ &= \exp [L_{be}^B] \cdot (1 + \exp [L_e^{10}]) - \exp [L_e^{01}]. \end{aligned} \quad (9)$$

Since  $p(A = 0) \cdot p(B = 1) - p(A = 1) \cdot p(B = 0) = p(u_k = 01) - p(u_k = 10)$ , we can obtain the following relations from (6):

$$\begin{aligned} \exp [L_e^{01}] &= \frac{\exp [L_{be}^B] - \exp [L_{be}^A]}{1 + \exp [L_{be}^A]} \\ &\quad + \frac{1 + \exp [L_{be}^B]}{1 + \exp [L_{be}^A]} \cdot \exp [L_e^{10}] \end{aligned} \quad (10a)$$

$$\begin{aligned} \exp [L_e^{10}] &= \frac{\exp [L_{be}^A] - \exp [L_{be}^B]}{1 + \exp [L_{be}^B]} \\ &\quad + \frac{1 + \exp [L_{be}^A]}{1 + \exp [L_{be}^B]} \cdot \exp [L_e^{01}]. \end{aligned} \quad (10b)$$

By considering both (9) and (10), we can obtain the relations among the symbol-level extrinsic information values with the bit-level extrinsic information values

$$\begin{aligned} \exp [L_e^{11}] &= \frac{\exp [L_{be}^A] \cdot (1 + \exp [L_{be}^B])}{1 + \exp [L_{be}^A]} \\ &\quad + \frac{\exp [L_{be}^A + L_{be}^B] - 1}{1 + \exp [L_{be}^A]} \cdot \exp [L_e^{10}] \\ &= \frac{\exp [L_{be}^B] \cdot (1 + \exp [L_{be}^A])}{1 + \exp [L_{be}^B]} \\ &\quad + \frac{\exp [L_{be}^A + L_{be}^B] - 1}{1 + \exp [L_{be}^B]} \cdot \exp [L_e^{01}]. \end{aligned} \quad (11)$$

Based on the relations discussed earlier, we can derive the symbol-to-bit conversion and bit-to-symbol conversion of extrinsic information.

### B. Symbol-to-Bit Conversion of Extrinsic Information

From (1)–(3), we can derive two equations which relate the bit-level probabilities to the extrinsic information values, as shown in the following:

$$\begin{aligned} p(A = 1) &= p(u_k = 10) + p(u_k = 11) \\ &= p(A = 0) \cdot \exp [L_{be}^A] \\ &= [p(u_k = 00) + p(u_k = 01)] \cdot \exp [L_{be}^A]. \end{aligned} \quad (12)$$

Similarly,  $p(B = 1) = [p(u_k = 00) + p(u_k = 10)] \cdot \exp [L_{be}^B]$ . As a consequence, two bit-level extrinsic information values can be obtained from (12) as follows:

$$\begin{aligned} L_{be}^A &= \ln \left( \frac{p(u_k = 10) + p(u_k = 11)}{p(u_k = 00) + p(u_k = 01)} \right) \\ &= \ln \left( \frac{\exp [L_e^{10}] + \exp [L_e^{11}]}{1 + \exp [L_e^{01}]} \right) \\ &\simeq \max (L_e^{10}, L_e^{11}) - \max (0, L_e^{01}) \end{aligned} \quad (13)$$

As expressed above, two bit-level extrinsic information values can be obtained from three symbol-level extrinsic information values. Therefore, the number of values to be stored in the memory can be reduced from three to two.

### C. Bit-to-Symbol Conversion of Extrinsic Information

To keep the compatibility with the conventional symbol-based SISO decoder, we should retrieve the symbol-level extrinsic information values from the bit-level extrinsic information values, as shown in Fig. 1. Using the relations derived previously, the proposed bit-to-symbol conversion of extrinsic information can be classified into four cases by taking into account the sign values of the two bit-level extrinsic information values.

1) *Case I:  $L_{be}^A \geq 0$  and  $L_{be}^B \geq 0$ :* From (8)–(11), we can relate  $L_e^{11}$  with the two bit-level extrinsic information values as follows:

$$\begin{aligned} \exp [L_e^{11}] &= \frac{1}{p(u_k = 00)} \\ &\quad \cdot \left( \frac{\exp [L_{be}^A]}{1 + \exp [L_{be}^A]} - \frac{1}{1 + \exp [L_{be}^B]} \right) + 1 \\ &\simeq \frac{1}{p(A = 0) \cdot p(B = 0)} \\ &\quad \cdot \left( \frac{\exp [L_{be}^A]}{1 + \exp [L_{be}^A]} - \frac{1}{1 + \exp [L_{be}^B]} \right) + 1 \\ &= (1 + \exp [L_{be}^A]) (1 + \exp [L_{be}^B]) \\ &\quad \cdot \left( \frac{\exp [L_{be}^A]}{1 + \exp [L_{be}^A]} - \frac{1}{1 + \exp [L_{be}^B]} \right) + 1 \\ &= \exp [L_{be}^A] + \exp [L_{be}^B] \end{aligned} \quad (14)$$

where  $p(u_k = 00)$  is approximated to  $p(A = 0) \cdot p(B = 0)$ . From (14),  $L_e^{11}$  can be determined as follows:

$$L_e^{11} \simeq \ln (\exp [L_{be}^A] + \exp [L_{be}^B]) \simeq \max (L_{be}^A, L_{be}^B). \quad (15)$$

Additionally, we can use (11) to approximate  $L_e^{10}$  as follows:

$$\begin{aligned} L_e^{10} &= \ln \left( \frac{(1 + \exp [L_{be}^A]) \cdot \exp [L_e^{11}]}{\exp [L_{be}^A + L_{be}^B] - 1} \right. \\ &\quad \left. - \frac{\exp [L_{be}^A] \cdot (\exp [L_{be}^B] + 1)}{\exp [L_{be}^A + L_{be}^B] - 1} \right) \\ &\simeq \ln (\exp [-L_{be}^B] \cdot \exp [L_e^{11}] - 1) \\ &\simeq \ln (\exp [L_e^{11} - L_{be}^B]) = L_e^{11} - L_{be}^B \\ &\simeq \max (L_{be}^A, L_{be}^B) - L_{be}^B. \end{aligned} \quad (16)$$

Similarly,  $L_e^{01} \simeq L_e^{11} - L_{be}^A \simeq \max (L_{be}^A, L_{be}^B) - L_{be}^A$ .

2) *Case II:  $L_{be}^A \geq 0$  and  $L_{be}^B < 0$* : Using the relations between  $L_e^{01}$  and  $L_e^{10}$  in (10a) and (10b), we can approximate  $L_e^{01}$  and  $L_e^{10}$

$$\begin{aligned} L_e^{10} &= \ln \left( \frac{\exp[L_{be}^A] - \exp[L_{be}^B]}{1 + \exp[L_{be}^B]} \right. \\ &\quad \left. + \frac{1 + \exp[L_{be}^A]}{1 + \exp[L_{be}^B]} \cdot \exp[L_e^{01}] \right) \\ &\simeq \ln \left( \frac{\exp[L_{be}^A] - \exp[L_{be}^B]}{1 + \exp[L_{be}^B]} \right) \\ &\quad (\because p(u_k = 01) \ll p(u_k = 00)) \\ &\simeq L_{be}^A \quad (\because L_{be}^A \geq 0 \&\& L_{be}^B < 0) \end{aligned} \quad (17)$$

$$\begin{aligned} L_e^{01} &= \ln \left( \frac{\exp[L_{be}^B] - \exp[L_{be}^A]}{1 + \exp[L_{be}^A]} \right. \\ &\quad \left. + \frac{1 + \exp[L_{be}^B]}{1 + \exp[L_{be}^A]} \cdot \exp[L_e^{10}] \right) \\ &\simeq \ln(-1 + \exp[-L_{be}^A + L_e^{10}]) \simeq -L_{be}^A + L_e^{10} \\ &\simeq 0 \quad (\because \text{from (17), } L_e^{10} \simeq L_{be}^A). \end{aligned} \quad (18)$$

According to (9),  $L_e^{11}$  can be expressed as follows:

$$\begin{aligned} L_e^{11} &= \ln(\exp[L_{be}^B] \cdot (1 + \exp[L_e^{10}]) - \exp[L_e^{01}]) \\ &\simeq L_{be}^B + L_e^{10} \\ &\simeq L_{be}^A + L_{be}^B \quad (\because \text{from (17), } L_e^{10} \simeq L_{be}^A). \end{aligned} \quad (19)$$

3) *Case III:  $L_{be}^A < 0$  and  $L_{be}^B \geq 0$* : In this case, symbol-level extrinsic information values can be retrieved similarly as discussed in Case II. Therefore

$$L_e^{01} \simeq L_{be}^B \quad L_e^{10} \simeq 0 \quad L_e^{11} \simeq L_{be}^A + L_{be}^B. \quad (20)$$

4) *Case IV:  $L_{be}^A < 0$  and  $L_{be}^B < 0$* : Since the two bit-level extrinsic information values are less than zero in this case, we can roughly approximate  $p(u_k = 00)$  as follows:

$$\begin{aligned} p(u_k = 00) &\simeq p(A = 0) \cdot p(B = 0) \\ &\simeq \frac{1}{1 + \exp[L_{be}^A] + \exp[L_{be}^B]} \end{aligned} \quad (21)$$

According to (8) and (21)

$$\begin{aligned} L_e^{11} &\simeq \ln \left( (1 + \exp[L_{be}^A] + \exp[L_{be}^B]) \right. \\ &\quad \left. \cdot \left[ \frac{\exp[L_{be}^A]}{1 + \exp[L_{be}^A]} - \frac{1}{1 + \exp[L_{be}^B]} \right] + 1 \right) \\ &\simeq \ln \left( \frac{\exp[L_{be}^A + L_{be}^B]}{\exp[L_{be}^A] + \exp[L_{be}^B]} \right) \\ &\simeq L_{be}^A + L_{be}^B - \max(L_{be}^A, L_{be}^B). \end{aligned} \quad (22)$$

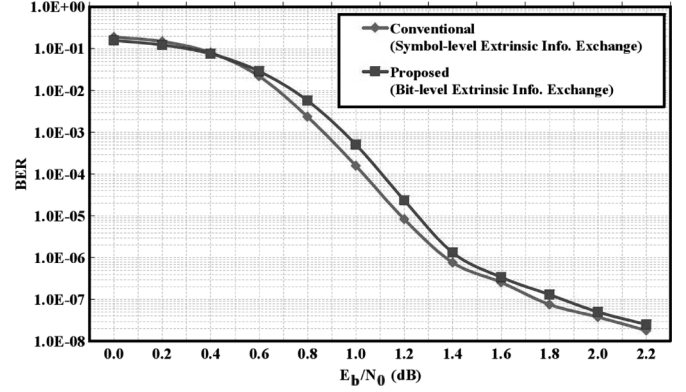


Fig. 2. Comparison of BER performances of eight iterations for 1920-bit frame.

We can approximate (11) by considering  $L_{be}^A < 0$  and  $L_{be}^B < 0$

$$\begin{aligned} \exp[L_e^{11}] &\simeq \exp[L_{be}^A] \cdot (1 + \exp[L_{be}^B]) \\ &\quad + (\exp[L_{be}^A + L_{be}^B] - 1) \cdot \exp[L_e^{10}] \\ &\simeq \exp[L_{be}^B] \cdot (1 + \exp[L_{be}^A]) \\ &\quad + (\exp[L_{be}^A + L_{be}^B] - 1) \cdot \exp[L_e^{01}] \\ &\simeq \exp[L_{be}^A] - \exp[L_e^{10}] \simeq \exp[L_{be}^B] - \exp[L_e^{01}] \\ &\simeq 0 \quad (\because p(u_k = 11) \ll p(u_k = 00)). \end{aligned} \quad (23)$$

Therefore,  $L_e^{01} \simeq L_{be}^B$  and  $L_e^{10} \simeq L_{be}^A$ .

As described previously, symbol-level extrinsic information values can be retrieved from bit-level extrinsic information values by using simple operations such as addition and maximum. Similarly, symbol-to-bit conversion can be derived for higher order nonbinary turbo codes by expressing the bit-level probabilities with the symbol-level probabilities. After the symbol-level extrinsic information is expressed with the bit-level extrinsic information, the bit-to-symbol conversion can be obtained by applying appropriate approximations according to the sign values of the bit-level extrinsic information.

#### IV. EXPERIMENTAL RESULTS

The proposed conversion method is applied to decode the turbo code specified in the WiMAX standard. With the experimental environment denoted in Table I, Fig. 2 shows the BER performance of a turbo decoder employing the proposed conversion method and compares with that of the conventional turbo decoder [5] using the symbol-level extrinsic information directly. The BER performances are measured for a coding rate of 1/2. Although some approximations are applied in deriving the proposed conversions, they lead to only a slight degradation of the signal-to-noise ratio, about less than 0.1 dB, since the extrinsic information does not need to be exact in decoding [4]. Consequently, we can reduce the number of extrinsic information values to be exchanged without inducing a considerable loss of error-correcting capability.

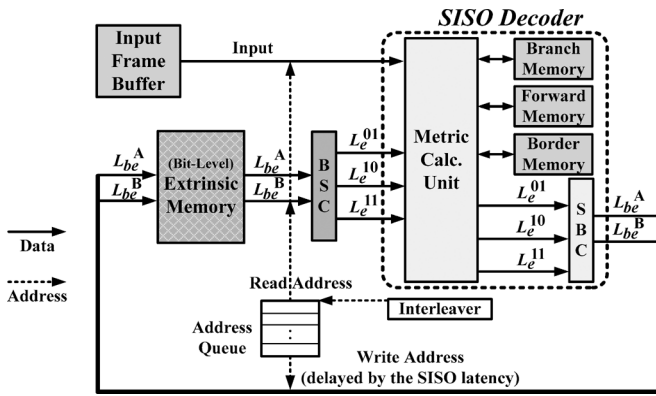


Fig. 3. Block diagram of the proposed double-binary turbo decoder.

TABLE IV  
SINGLE-PORT SRAM SIZE REQUIRED FOR THE TURBO DECODER

	Conventional [6]	Proposed
Input Frame Memory	57600 bits	57600 bits
Forward Metric Memory	4480 bits/SISO	4480 bits/SISO
Branch Metric Memory	5120 bits/SISO	5120 bits/SISO
Border Metric Memory	2*5180 bits = 10360 bits	2*5180 bits = 10360 bits
Extrinsic Info. Memory	115200 bits	76800 bits
Total	SISO x 1 192760 bits (100%)	154360 bits (80.0%)
	SISO x 4 221560 bits (100%)	183160 bits (82.7%)

### A. Hardware Implementation

With the quantization scheme indicated in Table I, a turbo decoder associated with the proposed conversion was designed in Verilog HDL and synthesized with a 0.18- $\mu\text{m}$  CMOS standard-cell library and compiled SRAM memories. The turbo decoder is based on the time-multiplex architecture and employs a dedicated hardware interleaver [5], as shown in Fig. 3. The memory size required for the turbo decoder is summarized in Table IV. Since a separate border memory is needed for each SISO decoding, two border memories are integrated in the decoder implementation. By adopting the proposed bit-level extrinsic information exchange method, the memory size required for the extrinsic information is reduced to two-thirds of the conventional method, as denoted in Table IV. When several SISO decoders are adopted to achieve a higher throughput, the size of the state metric memory should be increased, but the proposed conversion is still effective in reducing the total memory size, as the extrinsic information memory is much larger than the state metric memory, as indicated in Table IV. The proposed symbol-to-bit conversion can be performed in the hard-decision unit that makes hard-decided values at the end of the decoding process [9], since the relations expressed in (13) are the same as those needed to calculate the log-likelihood ratio values. Therefore, the only hardware overhead caused by the proposed conversion method is the bit-to-symbol converter (BSC) shown in

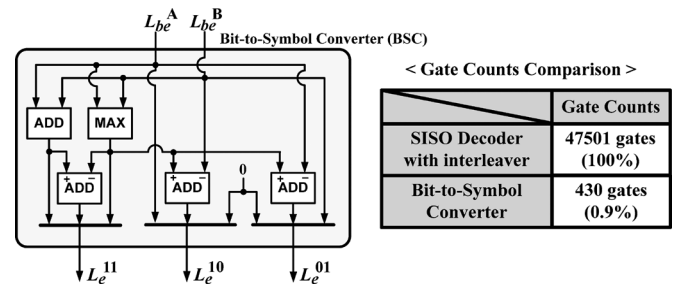


Fig. 4. Block diagram and complexity of the proposed BSC.

Fig. 4. As denoted in Fig. 4, the complexity of the proposed BSC is negligible compared to that of the total SISO decoder including the dedicated hardware interleaver and the hard-decision unit.

Since the amount of values per symbol required for  $m$ -ary extrinsic information is  $2^m - 1$ , the memory saving resulting from the proposed bit-level extrinsic information exchange increases as  $m$  increases, although no results on performance loss have been reported yet for  $m$  larger than two.

### V. CONCLUSION

We have presented a simple bit-level extrinsic information exchange method for double-binary turbo codes. The proposed method can reduce the number of extrinsic information values to be exchanged between two SISO decoders. Since the size of the extrinsic information memory is significant, the proposed method is effective in reducing the total memory size required in double-binary turbo decoders. To verify the proposed method, a double-binary turbo decoder is designed for the WiMAX turbo code. By applying the proposed conversion method, the total memory size is reduced by 20%.

### REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. Int. Conf. Commun.*, May 1993, pp. 1064–1070.
- [2] *Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems—Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands*, IEEE Std 802.16e/D5-2004, Nov. 2004.
- [3] C. Douillard and C. Berrou, "Turbo codes with rate-  $m/(m+1)$  constituent convolutional codes," *IEEE Trans. Commun.*, vol. 53, no. 10, pp. 1630–1638, Oct. 2005.
- [4] J. Vogt, J. Ertel, and A. Finger, "Reducing bit width of extrinsic memory in turbo decoder realizations," *Electron. Lett.*, vol. 36, no. 20, pp. 1714–1716, Sep. 2000.
- [5] J. H. Kim and I. C. Park, "Double-binary circular turbo decoding based on border metric encoding," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 55, no. 1, pp. 79–83, Jan. 2008.
- [6] Y. Gao and M. R. Soleymani, "Triple-binary circular recursive systematic convolutional turbo codes," in *Proc. 5th Int. Symp. Wireless Pers. Multimedia Commun.*, 2002, vol. 3, pp. 951–955.
- [7] M. C. Shin and I. C. Park, "SIMD processor-based turbo decoder supporting multiple third-generation wireless standards," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 7, pp. 801–810, Jul. 2007.
- [8] C. Berrou and M. Jezequel, "Non-binary convolutional codes for turbo coding," *Electron. Lett.*, vol. 35, no. 1, pp. 39–40, Jan. 1999.
- [9] S. J. Lee, N. R. Shanbhag, and A. C. Singer, "Area-efficient high-throughput MAP decoder architectures," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 8, pp. 921–933, Aug. 2005.