A 50Mbps Double-Binary Turbo Decoder for WiMAX Based on Bit-level Extrinsic Information Exchange

Ji-Hoon Kim and In-Cheol Park

School of EECS, Korea Advanced Institute of Science and Technology (KAIST),

Daejeon, Republic of Korea

jhkim82@gmail.com

Abstract-**A 50Mbps, 2.24mm² double-binary turbo decoder is designed and implemented in 0.13μm CMOS process for the WiMAX standard. To reduce the large extrinsic memory needed in double-binary turbo decoding, the proposed decoder exchanges the bit-level extrinsic information values rather than the traditional symbol-level extrinsic information values, which is achieved by deriving two simple conversions. The proposed turbo decoder, with a low-complexity hardware interleaver generating interleaved addresses for two data flows simultaneously, provides an efficient stopping criterion for double-binary turbo decoding using bit-level extrinsic information as well as huge memory size reduction of 20.6%.**

I. INTRODUCTION

The turbo code introduced in 1993 is one of the most powerful forward error correction channel codes, and provides near optimal performance approaching the Shannon limit [1]. Recently, the double-binary turbo code has received a great attention and adopted in several mobile radio systems such as DVB-RCS and IEEE 802.16 standard (WiMAX) [2], as it can offer many advantages over the single-binary turbo code [3].

Although there are many implementations for classical single-binary turbo decoders [4]-[6], there has been little research dedicated to the hardware implementation of the double-binary turbo decoder [7][8]. Compared to the single-binary turbo code, the double-binary turbo code requires much more memory in decoding. Especially, the size of the extrinsic information memory becomes large, since the number of extrinsic information values to be exchanged between two soft-input soft-output (SISO) decoders increases three times in double-binary turbo decoding. Also, the increased complexity of a SISO decoder makes it prohibitive to implement a high throughput turbo decoder where several SISO decoders are employed for parallel turbo decoding [4].

In this paper, we propose a bit-level extrinsic information-based decoding architecture to reduce the number of extrinsic information values to be exchanged between two SISO decoders, which leads to huge memory reduction in double-binary turbo decoder implementation. In addition, a low complexity SISO decoder structure is also presented. To verify the proposed architecture, a double-binary turbo decoder based on the bit-level extrinsic information exchange is implemented for WiMAX with employing a single SISO decoder and the dedicated double-flow interleaver.

II. DESIGN ISSUES in DOUBLE-BINARY TURBO DECODER

A typical turbo decoder is based on the time-multiplex architecture that contains one or more SISO decoders, an interleaver, and an extrinsic memory [5]. Based on the MAP algorithm [1], the double-binary turbo decoding is well described in [8]. Focusing on double-binary turbo decoding, we describe in this Section the conventional symbol-level extrinsic information and the implementation issues.

A. Increased Hardware Complexity of a SISO decoder

The metric calculation complexity of the non-binary turbo codes is higher than that of the single-binary turbo codes due to the increased number bits to be processed at a time. For the double-binary turbo codes where two bits are processed at a time, the number of branches connected to each trellis state is increased from two to four which leads to three times of hardware complexity than that of the classical single-binary turbo codes as described in [8].

B. Huge Extrinsic Information Memory Requirement

In the double-binary turbo code, three symbol-level extrinsic information values are defined as follows [7][8];

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

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 the two SISO decoders during the whole decoding process. In double-binary turbo decoding, more extrinsic information values are to be exchanged compared to the single-binary turbo decoding that stores only one extrinsic information value. Therefore, a large memory is needed in a double-binary turbo decoder.

Taking into account the quantization scheme indicated in Table I, the memory size required for a double-binary SISO decoder is summarized in Fig. 1. It is crucial to reduce the extrinsic information memory even if several SISO decoders are adopted for parallel decoding, as the extrinsic information memory is much bigger than the memory required in SISO decoding as indicated in Fig. 1.

III. PROPOSED DECODER ARCHITECTURE AND OPERATION

For an efficient turbo decoder implementation, an optimized SISO decoder is necessary especially when several SISO decoders are adopted to achieve high throughput. In addition, by reducing the number of extrinsic information values, overall turbo decoder complexity can be lowered. Focusing on the time-multiplex turbo decoder, we propose a new double-binary turbo decoder architecture that includes a double-flow hardware interleaver.

TABLE I
SIMULATION ENVIRONMENT

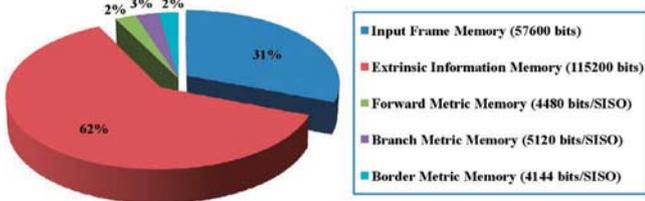| SISO Algorithm	Max-log-MAP
Window Size	32
Quantization (q, f) q bits in total f bits for fraction	Received input: (4, 2) Branch Metric & State Metrics : (10, 2) Extrinsic Information : (8, 2) LLR value : (11, 2)

Fig. 1. Memory requirements in double-binary turbo decoder [8].

Fig. 2. Branch metric memory width comparison

A. Low-Complexity SISO decoder design

As the complexity of the metric calculation increases, the sliding window with border memory is efficient for non-binary turbo decoding, because it can eliminate the need of the complex dummy backward calculation. In the proposed architecture, we adopt the 4-bit border metric encoding presented in [8] to reduce the border memory size. By applying the border metric encoding, since the border memory size is constant regardless of the number of SISO decoders in a turbo decoder, we can remove all complex dummy metric calculation units at the expense of a small border memory and slight performance degradation.

In addition, to lower the complexity of the SISO decoder, we exploit the branch metric recovery scheme. Branch metrics, $\tilde{\gamma}$, are defined in double-binary turbo code as follows.

$$\tilde{\gamma}_k = (x_k^{s_1} y_k^{s_1} + x_k^{s_2} y_k^{s_2} + x_k^{p_1} y_k^{p_1} + x_k^{p_2} y_k^{p_2}) + L_{e,IN}^{(z)} \qquad (2)$$

where z belongs to $\phi = \{01, 10, 11\}$, $\mathbf{x}_k$ and $\mathbf{y}_k$ are transmitted and received codewords, respectively, and we assume the binary phase shift keying modulation. The superscripts p and s denote the parity bits and systematic bits, respectively. In (2), $L_{e,IN}^{(z)}$ is the extrinsic information received from the other SISO decoder. Since there are 16 unique branch metrics in the double-binary turbo code, the branch metric memory size becomes significant, especially multiple SISO decoders are adopted to achieve high throughput. Therefore, we propose a new branch metric recovery scheme which does not store whole branch metric values, but stores only essential values required to recover the branch metrics. From (2), we can obtain following relation;

$$\tilde{\gamma}_k = (y_k^{s_1} + y_k^{s_2}) + x_k^{p_1} y_k^{p_1} + x_k^{p_2} y_k^{p_2} + (L_{e,IN}^{(z)} + L_i^{(z)}) \qquad (3)$$

where $L_i^{(z)}$ is the intrinsic information defined as follows.

$$L_i^{01} = -2y^{s_2}, \qquad L_i^{10} = -2y^{s_1}, \qquad L_i^{11} = -2y^{s_1} - 2y^{s_2} \qquad (4)$$

Therefore, by storing only essential sub-metrics as shown in Fig. 2, we can significantly reduce the memory size required to store branch metrics at the expense of the simple calculation expressed in (3). We can reduce the branch memory size
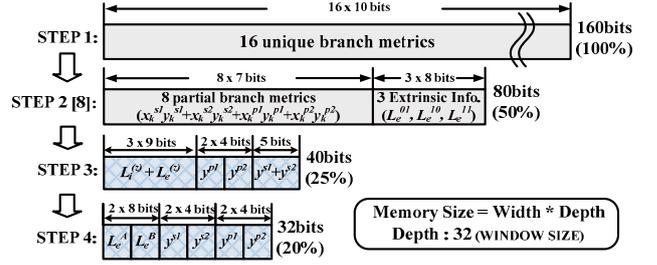
further by keeping proposed the bit-level extrinsic information as shown in Fig. 2.

B. Bit-level Extrinsic Information Exchange

As discussed in Section II, there are three symbol-level extrinsic information values to be exchanged between two SISO decoders. To reduce the size, the extrinsic information can be stored in bit-level. In this subsection, we describe how to derive the bit-level extrinsic information from the symbol-level extrinsic information, and two simple converters as described in Fig. 3.

Two bit-level extrinsic information values are defined as follows.

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

$$L_{be}^B = \ln \frac{p(B=1)}{p(B=0)} \Rightarrow p(B=1) = p(B=0) \cdot \exp[L_{be}^B] \qquad (5)$$

where the input symbol u_k consists of a pair of two bits, A and B, i.e., $u_k = AB$. By expressing the bit-level probabilities with the symbol-level probabilities, the symbol-to-bit conversion can be derived as follows.

$$L_{be}^A = \ln\left(\frac{p(u_k=10)+p(u_k=11)}{p(u_k=00)+p(u_k=01)}\right) \simeq \max\left(L_e^{10}, L_e^{11}\right) - \max\left(0, L_e^{01}\right)$$

$$L_{be}^B = \ln\left(\frac{p(u_k=01)+p(u_k=11)}{p(u_k=00)+p(u_k=10)}\right) \simeq \max\left(L_e^{01}, L_e^{11}\right) - \max\left(0, L_e^{10}\right) \qquad (6)$$

By converting three symbol-level values into two bit-level values as expressed in (6), we can reduce the number of values to be stored in a memory from three to two.

To retrieve the symbol-level extrinsic information values from bit-level values for next SISO decoding, the bit-to-symbol conversion can be obtained by applying appropriate approximations according to the sign values of the bit-level extrinsic information after the symbol-level extrinsic information is expressed with the bit-level extrinsic information as follows.

1) Case-I : $L_{be}^A \geq 0$ and $L_{be}^B \geq 0$

By approximating $p(u_k = 00)$ to $p(A = 0) \cdot p(B = 0)$,

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

$$L_e^{10} \simeq L_e^{11} - L_{be}^B \simeq \max(L_{be}^A, L_{be}^B) - L_{be}^B \qquad (7)$$

$$L_e^{01} \simeq L_e^{11} - L_{be}^A \simeq \max(L_{be}^A, L_{be}^B) - L_{be}^A$$
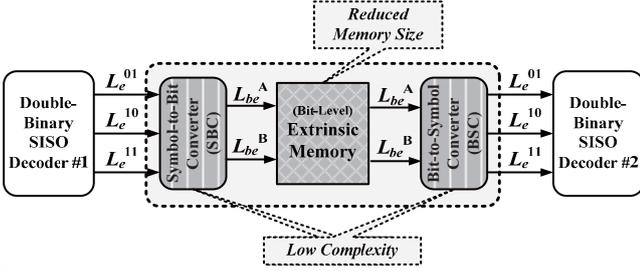
Fig. 3. Block diagram of the proposed bit-level extrinsic information exchange
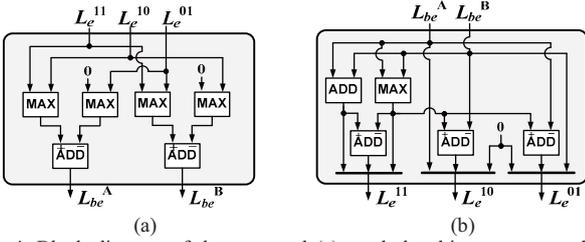

(a)   (b)

Fig. 4. Block diagram of the proposed (a) symbol-to-bit converter and (b) bit-to-symbol converter.

2) *Case-II : $L_{be}{}^A < 0$ and $L_{be}{}^B < 0$*

By approximating $p(u_k = 00)$ to $p(A = 0) \cdot p(B = 0)$,

$$L_e^{11} \simeq \ln\left(\frac{\exp[L_{be}^A + L_{be}^B]}{\exp[L_{be}^A] + \exp[L_{be}^B]}\right) \simeq L_{be}^A + L_{be}^B - \max\left(L_{be}^A, L_{be}^B\right) \quad (8)$$

$$L_e^{10} \simeq L_{be}^A, \qquad L_e^{01} \simeq L_{be}^B$$

3) *Case-III : $L_{be}{}^A \geq 0$ and $L_{be}{}^B < 0$*

By using the property that $p(u_k = 01) \ll p(u_k = 00)$,

$$L_e^{11} \simeq L_{be}^A + L_{be}^B, \qquad L_e^{10} \simeq L_{be}^A, \qquad L_e^{01} \simeq 0 \quad (9)$$

4) *Case-IV : $L_{be}{}^A < 0$ and $L_{be}{}^B \geq 0$*

By using the property that $p(u_k = 10) \ll p(u_k = 00)$,

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

Through (7)-(10), we can retrieve three symbol-level values from two-bit level values stored in a memory, which are required for double-binary SISO decoding.

Both converters, described in (6)-(10), can be implemented with simple operations such as addition and maximum as shown in Fig. 4. Regardless of the bitwidth of the extrinsic information, by decreasing the number of extrinsic information values to be exchanged, the proposed architecture can reduce the extrinsic information memory which is the largest memory in double-binary turbo decoder as shown in Fig. 1.

C. Dedicated Double-Flow Hardware Interleaver

Compared to the traditional memory-based interleaver, a dedicated interleaver can achieve small area as well as low power consumption [5]. Based on the dedicated hardware interleaver presented in [8], we propose the double-flow interleaver which can generate the write addresses as well as read addresses for the extrinsic information memory as shown in Fig. 5, where P_0 and initial values are determined according
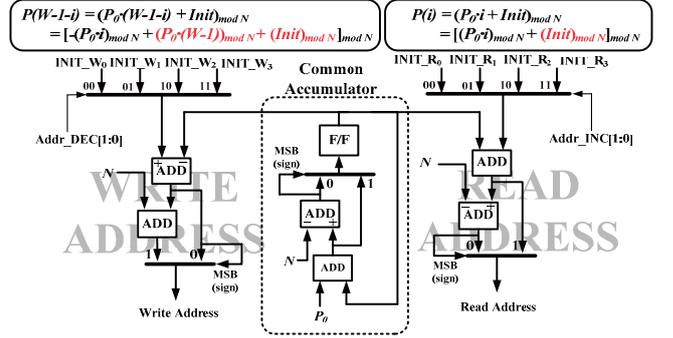

Fig. 5. Double-flow hardware interleaver based on incremental calculation.
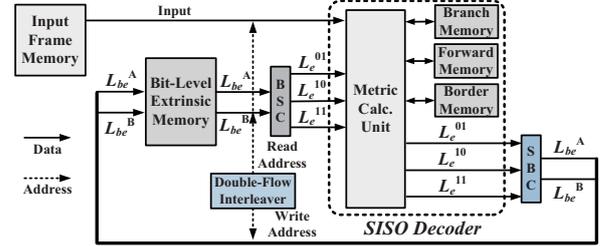

Fig. 6. Block diagram of the proposed double-binary turbo decoder

to the frame length, N [2]. With two initial value sets, two addresses can be generated on-the-fly from one shared accumulator. Since write addresses can be generated on-the-fly, the address queue required to delay read addresses in the time-multiplex architecture [5][8] can be removed. The initial address values can be managed with employing a small look-up table.

IV. Double-Binary Turbo Decoder Design for WiMAX

To verify the proposed architecture, we implemented a double-binary turbo decoder for WiMAX. Design parameters used in the implementation are the same as those in Table I. The proposed time-multiplex turbo decoder consists of a single SISO decoder, two converters, the double-flow dedicated interleaver, and the memory which holds the bit-level extrinsic information as shown in Fig.6.

A. Proposed Stopping Criterion with Bit-level Extrinsic Info.

To avoid unnecessary iterations at a high signal-to-noise ratio (SNR), a simple early stopping criterion is employed where the sign values of incoming bit-level extrinsic information, S_e^A and S_e^B, are compared with hard-decision bits, S_{llr}^A and S_{llr}^B. The effect of the stopping criterion and the hardware schematic is shown in Fig. 7. At the end of SISO decoding, the proposed turbo decoder stops iteration if the following condition is satisfied for all pairs in a frame.

$$\left(S_e^A == S_{llr}^A\right) \&\& \left(S_e^B == S_{llr}^B\right) \quad (11)$$

The proposed stopping criterion can be implemented with low hardware complexity as shown in Fig. 7. If the above condition is satisfied for all pairs in a frame, then STOP indicates 1. However, if there are any pairs which cannot
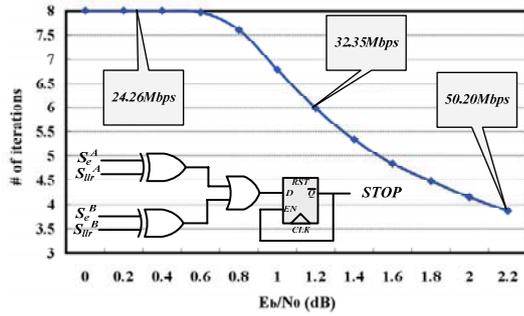
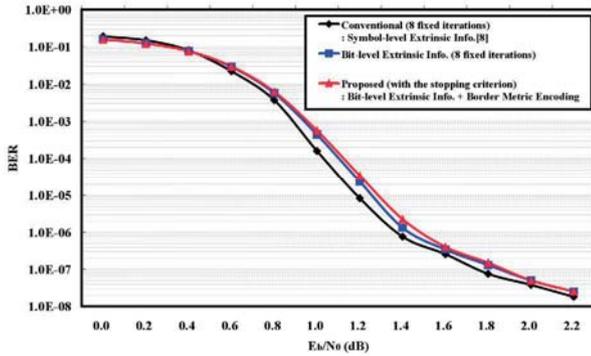Fig. 7. Average number of iterations for the proposed turbo decoder

Fig. 8. Comparison of BER performance for 1920-bit frame.

satisfy the above condition, then the value of STOP stays 0 during the SISO decoding. The flip-flop used in this scheme is reset when the new SISO decoding starts.

B. Double-Binary Turbo Decoder Implementation Results

The BER performance of the proposed turbo decoder is shown in Fig. 8, where we compare with that of a conventional turbo decoder that is based on the symbol-level extrinsic information [8]. The memory size required for the proposed turbo decoder is summarized in Table II. By adopting the proposed bit-level extrinsic information exchange, the memory size required for the extrinsic information is reduced to two-third of the conventional method as denoted in Table II which leads to total memory size reduction of 20.6%. When several SISO decoders are adopted to achieve a higher throughput [4], the size of the state metric memory should be increased, but the proposed architecture is still effective in

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

|		Conventional [8]	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*2072 bits = 4144 bits/SISO	2*2074 bits = 4144 bits/SISO
Extrinsic Info. Memory		115200 bits	76800 bits
Total	SISO x 1	186544 bits (100%)	148144 bits (79.4%)
	SISO x 4	215344 bits (100%)	176944 bits (82.2 %)
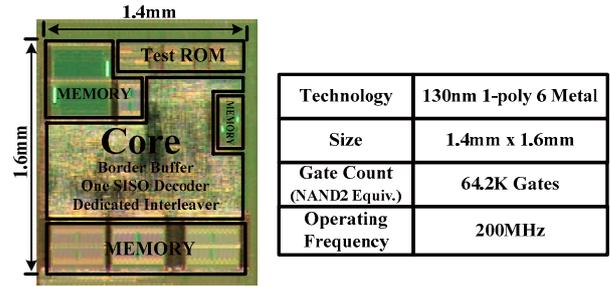
Fig. 9. Die photo of the proposed double-binary turbo decoder chip

reducing the total memory size, as the extrinsic information memory is much larger than the state metric memory as indicated in Table II. Fig. 9 summarizes implementation results. The proposed turbo decoder is implemented with 0.13μm 1-poly 6-metal standard CMOS process. The decoder occupies 2.24mm^2 and takes 4,948 cycles for each iteration to process a 2400-pair (4800 bit) frame. As a result, the proposed decoder provides up to 50Mbps at the frequency of 200MHz.

V. CONCLUSION

This paper has presented a double-binary turbo decoder developed for the WiMAX standard. To reduce the memory size required in double-binary turbo decoding, the proposed decoder exchanges the bit-level extrinsic information instead of the conventional symbol-level extrinsic information. The conversions between the symbol-level information and the bit-level information have been derived with two low-complexity converters. The proposed decoder consisting of a single optimized SISO decoder, a low-complexity double-flow hardware interleaver is implemented in 0.13μm CMOS process and occupies 2.24mm^2. The proposed double-binary decoder can provide up to 50Mbps at the frequency of 200MHz with the proposed simple stopping criterion.

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] C. C. Wong, *et al.,* "A 0.22nJ/b/iter 0.13μm Turbo Decoder Chip Using Inter-Block Permutation Interleaver," in *Proc. IEEE Custom integrated Circuits Conf.*, Sep. 2007, pp. 273–276.

[5] M. C. Shin and I. C. Park, "SIMD Processor-Based Turbo Decoder Supporting Multiple Third-Generation Wireless Standards," in *ISSCC Dig. Tech. Papers*, 2003, pp. 154–155.

[6] C. Benkeser, A. Burg, T. Cupaiuolo, and Q. Huang, "A 58mW 1.2mm^2 HSDPA Turbo Decoder ASIC in 0.13μm CMOS," in *ISSCC Dig. Tech. Papers*, 2008, pp. 264–265.

[7] O. Muller, A. Baghdadi, and M. Jezequel, "ASIP-based multiprocessor SoC design for simple and double binary turbo decoding," in *Proc. DATE 2006*, pp. 1330–1335.

[8] J. H. Kim and I. C. Park, "Double-Binary Circular Turbo Decoding Based on Border Metric Encoding," *IEEE Trans. Circuits Syst. II. Express Briefs*, vol. 55, no. 1, pp. 79–83, Jan. 2008.