Processor-Based Turbo Interleaver for Multiple Third-Generation Wireless Standards

Myoung-Cheol Shin, *Student Member, IEEE,* and In-Cheol Park, *Senior Member, IEEE*

***Abstract*—A software turbo interleaver running on a SIMD processor is presented for a turbo decoder supporting multiple 3G wireless standards. To hide the timing overhead of interleaver changing, the interleaver generation is split into two parts, preprocessing and incremental on-the-fly generation. Applying the proposed approach, we implemented a W-CDMA and cdma2000 interleaver that generates one interleaved address per cycle and occupies 10% area of the ROM implementation.**

***Index Terms*—cdma2000, SIMD processor, turbo interleaver, W-CDMA.**

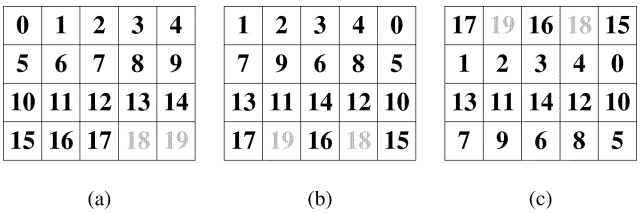

Fig. 1. A prunable interleaver with $N = 18$. (a) Incoming data indexes. (b) Intra-row permutation. (c) Inter-row permutation.

I. Introduction

As TURBO CODES [1], or parallel concatenated convolutional codes, have extremely impressive performances, they entered the field of standardized systems in recent years. One of the most important examples is the channel coding of the third-generation (3G) mobile radio systems such as W-CDMA [2] and cdma2000 [3]. Flexible and programmable turbo decoders are required for 3G communications because 1) global roaming is recommended between different 3G standards, and 2) the frame size may change on a frame-by-frame basis. The turbo decoder consists of soft-input-soft-output (SISO) decoders and interleavers. Flexible and programmable implementation is especially needed for the turbo interleaver, as each 3G standard has a distinct and complicated interleaver while the SISO has virtually the same structure. The most common approach to implement the interleaver is to store the interleaved patterns in a ROM. This approach is not adequate for a turbo decoder supporting multiple wireless standards, as it needs a huge ROM to store all the interleaved patterns.

In this letter, an incremental interleaving algorithm running on a single-instruction-multiple-data (SIMD) processor and special instructions suitable for multiple 3G standards are proposed to provide interleaved data at the speed of the hardware SISO and to change the interleaver structure in a very short time with a small memory.

II. Standardized Turbo Interleavers

The turbo decoders for wireless systems are based on block turbo interleavers. Although the operations and parameters vary depending on the standards, W-CDMA and cdma2000 share the prunable block interleaver structure [4] where the interleaver is implemented by first building a *mother interleaver* of a predefined size and then pruning unnecessary indexes. The mother interleaver writes data in a two-dimensional matrix row by row, permutes them according to intra-row and inter-row permutation rules, and read them out column by column. An example with a frame size $N = 18$ is shown in Fig. 1. The intra-row permutation rule applied to Fig. 1(b) is

$$y_{i,j} = b_i + [(j+1) \times q_i] \bmod 5 \tag{1}$$

where i and j are row and column indexes, respectively, $\mathbf{b} = (b_0, b_1, b_2, b_3) = (0, 5, 10, 15)$, and $\mathbf{q} = (q_0, q_1, q_2, q_3) = (1, 2, 3, 7)$. Fig. 1(c) shows the inter-row permutation result, which will be read out as a sequence of 17, 1, 13, 7, 2, ..., 5, since the elements exceeding the range are pruned. The W-CDMA and cdma2000 turbo interleavers are similar to this example but much more complicated. For its more detailed description, readers are referred to [2], [3].

The entire interleaver should be reconstructed if its size changes by more than the granularity of the mother interleavers. All the 3G mobile communications support variable bit rate that can change the interleaver size on a 10 ms or 20 ms frame basis, and W-CDMA supports separately-coded transport channels. This implies the interleaver structure may change every 10 ms.

III. Turbo Decoder Architecture

As shown in Fig. 2, we assume that the interleaver is contained in the simplest time-multiplex turbo decoder [5] consisting of only one SISO, one interleaver, and one extrinsic LLR (Λ_e) memory. Data are sequentially stored in Λ_e memory as they are always read and written *in-place*. For each iteration, they are accessed in a sequential order for the first SISO decoding and in an interleaved order for the second decoding. As shown with the dotted lines in Fig. 2, the interleaver produces the interleaved addresses to read data in an interleaved order. The address queue saves the addresses so that they can be used again as the write addresses when the SISO produces the results after its latency.

Manuscript received September 9, 2002. The associate editor coordinating the review of this letter and approving it for publication was Prof. K. Narayanan. This work was supported in part by the Korea Science and Engineering Foundation through the MICROS Center, and by the Ministry of Science and Technology and the Ministry of Commerce, Industry and Energy through the Project System IC 2010.

The authors are with the Electrical Engineering and Computer Science Department, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea (e-mail: mcshin@ics.kaist.ac.kr; icpark@ee.kaist.ac.kr).

Digital Object Identifier 10.1109/LCOMM.2003.812176

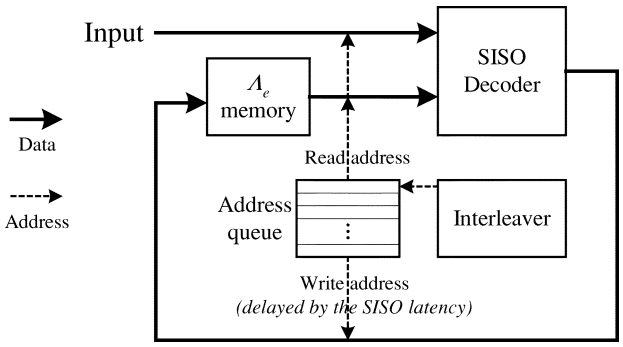

Fig. 2. A block diagram of the time-multiplex turbo decoder.

We propose a turbo decoder implemented with a hardware sliding-window SISO for the computationally intensive but regular SISO decoding and a programmable SIMD processor for the flexible turbo interleaving. In addition to the interleaving, the processor can control the hardware blocks or interface with an external host during the first SISO decoding that does not need interleaved addresses.

To keep pace with the hardware SISO, parallel processing is indispensable for interleaved address generation. SIMD architecture is chosen as it is suitable for the simple and repetitive address calculation. VLIW (very long instruction word) and superscalar architectures have unnecessarily complex control for such simple processing, and their high bandwidth needed for instruction fetching requires much memory area and power.

IV. Preprocessing and On-the-Fly Generation

In 3G wireless systems, the bit rate, and thereby the interleaver size, may change frequently. Generating the whole interleaved address pattern at once requires much time and a large-sized RAM. As a solution to this problem, we split the interleaver generation into two parts: preprocessing for interleaving and incremental on-the-fly address generation. When the bit rate changes, only the preprocessing prepares a small number of seed variables. Whenever the interleaved address sequence is required, the SIMD processor generates it column by column using the variables. This splitting method reduces the timing overhead of interleaver changing and requires only a small memory to save the seed data.

Let us explain the approach using the example of Fig. 1. In order to remove multiplication and modulo operations that take many cycles from (1), we introduce an increment vector $\mathbf{w}$ of $w_i = q_i \bmod 5$ and a cumulative vector $\mathbf{x}_j$ of

$$x_{i,j} = [(j+1) \times q_i] \bmod 5. \tag{2}$$

Then (1) can be rewritten as

$$y_{i,j} = b_i + x_{i,j}, \tag{3}$$

and $\mathbf{x}_j$ can be obtained recursively as

$$\begin{aligned} x_{i,j} &= [(j+1) \times (q_i \bmod 5)] \bmod 5 \\ &= [(j+1) \times w_i] \bmod 5 \\ &= [(jw_i \bmod 5) + w_i] \bmod 5 \\ &= (x_{i,j-1} + w_i) \bmod 5, \end{aligned} \tag{4}$$

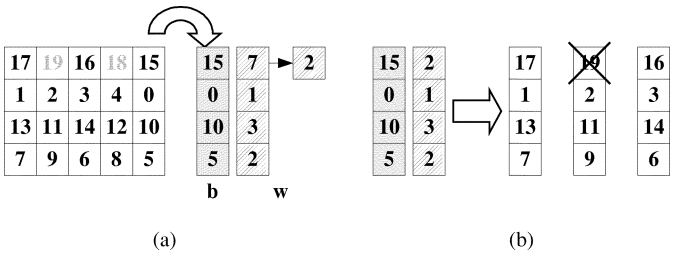

Fig. 3. Proposed algorithm applied to Fig. 1: (a) preprocessing for interleaving and (b) incremental on-the-fly address generation.

```
0  column_counter = C − 1
loop :
1  x = x + w
2  foreach(i = 0, …, R − 1) if (x_i ≥ p − 1)  x_i = x_i − (p − 1)
3  load s(x) from the data memory
4  y = b + s(x)
5  foreach(i = 0, …, R − 1) if (y_i < N)  send y_i to the address queue
6  if ((column_counter − −) ≠ 0) goto loop
```

Fig. 4. Pseudocode of the W-CDMA on-the-fly address generation.

where $j = 1, \ldots, 4$ and $\mathbf{x}_0 = \mathbf{w}$. As $0 \le x_{i,j-1} < 5$ and $0 \le w_i < 5$, $0 \le x_{i,j-1} + w_i < 10$ and thus

$$x_{i,j} = \begin{cases} x_{i,j-1} + w_i - 5 & \text{if } x_{i,j-1} + w_i \ge 5 \\ x_{i,j-1} + w_i & \text{otherwise} \end{cases}, \tag{5}$$

where the multiplication and modulo operations are replaced by cheaper operations.

As Fig. 3(a) shows, in the preprocessing stage, $\mathbf{b}$, $\mathbf{w}$ and $\mathbf{x}_0$ are calculated and stored in vector registers of the SIMD processor in the inter-row permutation order in advance. In the on-the-fly generation shown in Fig. 3(b), the processor updates $\mathbf{x}_j$ according to (5) and calculates the addresses as (3). The calculated addresses are sent to the address queue, if they are smaller than N.

We have applied this technique to W-CDMA [2], cdma2000 [3], and CCSDS standard turbo interleavers. It is not difficult to apply the technique to other standard turbo interleavers. A multi-standard interleaver can be realized by loading several interleaver programs and switching between them.

As a more realistic example, let us briefly describe the W-CDMA turbo interleaver. The number of rows R, the number of columns C, the prime number p, and the permutation function $s(\mathbf{x})$ defined in [2] are obtained in the preprocessing. Finally, for the on-the-fly generation we save the seed variable vectors of length R: address base $\mathbf{b}$, cumulative variable $\mathbf{x}$, increment value $\mathbf{w}$ where $w_i = q_i \bmod (p-1)$. The pseudocode of the on-the-fly address generation for W-CDMA is shown in Fig. 4.

V. SIMD Instructions for Turbo Interleavers

As the interleaved addresses should be generated at a high rate, three customized SIMD instructions, named STOLT (store to output port if less than), SUBGE (subtract if greater or equal),

TABLE I
AREA COMPARISON OF INTERLEAVER IMPLEMENTATIONS

Implementation		Size	Area (mm^2)
ROM implementation	W-CDMA ROM	6.12Mb	25.75
	cdma2000 ROM	613.9kb	2.523
	Total		28.27
Proposed implementation	SIMD processor	21,200 gates	1.727
	Instruction RAM	16kb	0.4075
	Data RAM	16kb	0.4075
	Address queue	2560b	0.1356
	Total		2.678

TABLE II
CYCLE COUNTS FOR THE MOST CRITICAL INTERLEAVER SIZES OF W-CDMA

Interleaver size	SISO decoding	Preproc-essing	On-the-fly generation	Bit/cycle
40	172	317	51	0.7843
41	176	295	58	0.7069
5,040	10,350	3,587	5,295	0.9518
5,114	10,498	3,048	5,375	0.9514

and LOOP, are introduced to replace instruction sequences commonly appearing in the on-the-fly generation. Each of them takes only one cycle to execute, and is equivalent to a sequence of three ordinary instructions.

The function of STOLT is to send the address to the queue only if it is smaller than N, which is needed for the pruning shown in the line 5 of Fig. 4. SUBGE is a conditional instruction corresponding to (5) and the line 2 of Fig. 4, which is useful for block interleavers that commonly use modulo operations. LOOP is adopted from DSP's to reduce the loop overhead by decrementing loop count and branching simultaneously. It corresponds to the line 6 of Fig. 4.

These instructions reduce the length of the on-the-fly generation loop of W-CDMA, cdma2000, and CCSDS to six, five, and four instructions, respectively. Each line of Fig. 4 corresponds to one SIMD instruction. If a SIMD processor has five processing elements (PE's), it can calculate five interleaved addresses for cdma2000 every five cycles and support a SISO that decodes one bit per cycle.

VI. IMPLEMENTATION RESULTS

Based on the proposed SIMD processor, we implemented an entire turbo decoder that decodes both W-CDMA and cdma2000 turbo codes in a 0.25 μm CMOS technology [6]. The SIMD processor has five PE's since the number of rows of W-CDMA block interleaver is a multiple of five.

Table I summarizes the comparison of chip area. The ROM implementation stores all the mother interleaver sequences in ROM's. W-CDMA ROM is much larger than that of cdma2000 because the granularity of the W-CDMA mother interleavers is much finer. W-CDMA has about 150 different mother interleaver patterns, whereas cdma2000 only seven. The proposed processor-based interleaver of our chip occupies 2.678 mm^2 including memory, which is only 10% of the ROM implementation.

Table II shows the performance of the proposed interleaver in terms of the cycle counts for the W-CDMA turbo decoding. For large-sized interleavers for high bit rates, the on-the-fly generation is almost as fast as one address per cycle. In addition, the preprocessing time is shorter than the SISO decoding, meaning that it can be completely hidden within the first SISO decoding.

VII. CONCLUSION

An interleaving algorithm suitable for 3G wireless systems has been proposed. The interleaver generation is split into two parts to reduce the overhead time of interleaver changing. We also proposed a SIMD processor with customized instructions to speed up the algorithm. The experimental results show that the proposed interleaver is fast enough to provide one interleaved address per cycle and hide the timing overhead of interleaver changing in most cases. Foremost, it reduces the chip area to 10% of the conventional ROM implementation.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," *Proc. IEEE ICC '93*, vol. 2, pp. 1064–1070, May 1993.
[2] "Technical Specification Group Radio Access Network; Multiplexing and Channel Coding (FDD)," 3rd Generation Partnership Project, 3GPP TS25.212 v5.1.0, 2002.
[3] "Physical Layer Standard for cdma2000 Spread Spectrum Systems," 3rd Generation Partnership Project 2, 3GPP2 C.S0002-C, v1.0, 2002.
[4] M. Eroz and A. R. Hammons Jr, "On the design of prunable interleavers for turbo codes," *Proc. IEEE 49th VTC*, vol. 2, pp. 1669–1673, July 1999.
[5] J. Vogt, K. Koora, A. Finger, and G. Fettweis, "Comparison of different turbo decoder realization for IMT-2000," *Proc. IEEE GLOBECOM '99*, vol. 5, pp. 2704–2708, Dec. 1999.
[6] M.-C. Shin and I.-C. Park, "A programmable turbo decoder for multiple 3G wireless standards," *ISSCC Dig. Tech. Papers*, pp. 154–155, Feb. 2003.