

Interleaved Convolutional Code and Its Viterbi Decoder Architecture

Jun Jin Kong

Department of Electrical and Computer Engineering, University of Minnesota, 200 Union Street, Minneapolis, MN 55455, USA
Email: jjkong@ece.umn.edu

Keshab K. Parhi

Department of Electrical and Computer Engineering, University of Minnesota, 200 Union Street, Minneapolis, MN 55455, USA
Email: parhi@ece.umn.edu

Received 4 February 2003 and in revised form 17 June 2003

We propose an area-efficient high-speed interleaved Viterbi decoder architecture, which is based on the state-parallel architecture with register exchange path memory structure, for interleaved convolutional code. The state-parallel architecture uses as many add-compare-select (ACS) units as the number of trellis states. By replacing each delay (or storage) element in state metrics memory (or path metrics memory) and path memory (or survival memory) with I delays, interleaved Viterbi decoder is obtained where I is the interleaving degree. The decoding speed of this decoder architecture is as fast as the operating clock speed. The latency of proposed interleaved Viterbi decoder is “decoding depth (DD) \times interleaving degree (I) + extra delays (A),” which increases linearly with the interleaving degree I .

Keywords and phrases: interleaved convolutional code, interleaved Viterbi decoder, burst-error correction, random-error correction, interleaving.

1. INTRODUCTION

It is well known that burst-error is a serious problem especially in storage and wireless mobile communication systems. In order to cope with burst-error, interleaving, denoted here as channel interleaving, with random-error correcting code, is generally used. Interleaving randomizes error bursts by spreading the erroneous bits with introducing a very long delay time, which is intolerable in some applications.

A burst-error correcting Viterbi algorithm, which combines maximum likelihood decoding algorithm with a burst detection scheme, instead of using channel interleaving, was proposed in [1] and extended to the Q²PSK in [2]. This adaptive Viterbi algorithm (AVA) outperforms interleaving strategies in the presence of very long bursts. However, when many short error bursts are present, AVA is inferior to interleaving scheme. An interleaved convolutional code also can be used for burst-error correction [3]. A modified Viterbi algorithm (MVA) [4], which is based on the multitrellis decomposition [5], was presented for interleaved convolutional code. The MVA introduces a much smaller delay time and much lower memory requirements than channel interleaving techniques with convolutional code. However, the implementation of MVA in [4], which uses as many delay elements as decoding depth (DD) \times interleaving degree (I) for

each code word component, is not area efficient. Some applications of interleaved convolutional code for asynchronous transfer mode (ATM) networks [6] and image communication systems [7, 8, 9] have been proposed.

In this paper, an area-efficient high-speed interleaved Viterbi decoder architecture, which has state-parallel architecture with register exchange path memory structure, is proposed. This paper is an expanded version of [10]. A brief introduction of the interleaved convolutional code is given in Section 2. A proposed interleaved Viterbi decoding algorithm and its architecture for interleaved convolutional code are shown in Section 3.

2. INTERLEAVED CONVOLUTIONAL CODE (ICC)

Interleaved convolutional code with extra delay (A), which further randomizes the error bursts, can be used for burst-error correction as shown in Figure 1. In this coding scheme, the channel interleaving is not used. The performance of this interleaved convolutional coding scheme depends on the interleaving degree and the extra delay. Interleaved convolutional code with interleaving degree I is obtained by replacing each delay (or storage) element in generator polynomials with I delays. In Figure 1, MUX and DE-MUX represent multiplexer and demultiplexer, respectively.

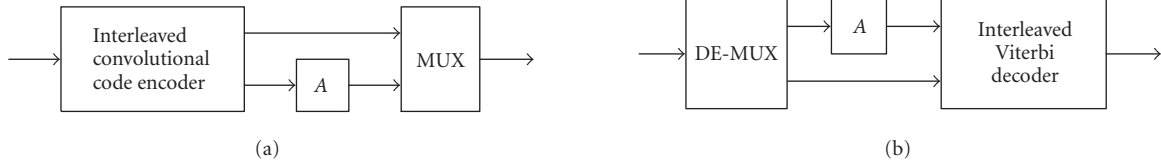
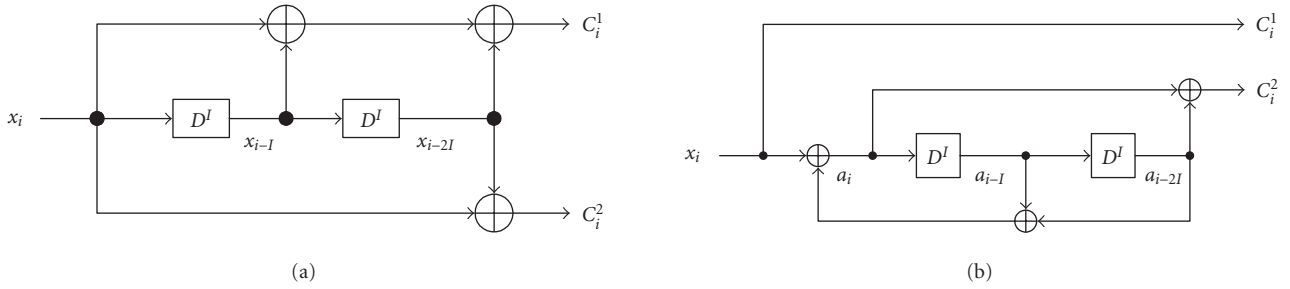


FIGURE 1: Interleaved convolutional code block diagram: (a) encoder (b) decoder.

FIGURE 2: Interleaved convolutional code encoder with interleaving degree I for $m = 2$, $R = k/n = 1/2$, and (a) $G = (7, 5)_8$ NRNSC code and (b) $G = (1, 5/7)_8$ RSC code.

An interleaved convolutional code can be obtained from a nonrecursive nonsystematic convolutional (NRNSC) code or a recursive systematic convolutional (RSC) code as shown in Figure 2. In order to illustrate the algorithm, we will consider an $(n, k, m) = (2, 1, 2)$ binary convolutional code with the following generator polynomials (G):

(1) NRNSC:

$$G(D) = (g^1(D), g^2(D)) = (7, 5)_8 = (1 + D + D^2, 1 + D^2), \quad (1a)$$

(2) RSC:

$$\begin{aligned} G(D) &= (g^1(D), g^2(D)) = (1, 5/7)_8 \\ &= (1, (1 + D^2)/(1 + D + D^2)). \end{aligned} \quad (1b)$$

For these codes, the generator polynomials of interleaved convolutional code with interleaving degree I become

(1) interleaved NRNSC:

$$G(D^I) = (g^1(D^I), g^2(D^I)) = (1 + D^I + D^{2I}, 1 + D^{2I}), \quad (2a)$$

(2) interleaved RSC:

$$G(D^I) = (g^1(D^I), g^2(D^I)) = (1, (1 + D^{2I})/(1 + D^I + D^{2I})) \quad (2b)$$

which yield $(2, 1, 2I)$ interleaved convolutional code. From equations (1) and (2), we can see that each delay element (D) in generator polynomials is replaced by I delays as shown in Figure 2.

The encoding and decoding processes will be explained in z -transform domain. In this representation, each delay element D of generator polynomials is replaced by z^{-1} .

A binary information sequence to be encoded is represented as

$$X(z) = \sum_{k=0}^{\infty} a_k z^{-k}, \quad a_k \in \{0, 1\}, \quad (3)$$

where a_k is a coefficient of information sequence and it has the values 0 or 1 since the binary system is considered. For an $(n, 1, mI)$ interleaved convolutional code, the generator polynomials with interleaving degree I are

$$G^i(z^I) = \sum_{j=0}^m g_j^i z^{-jI}, \quad (4)$$

where g_j^i is a coefficient of the generator polynomials and $g_j^i \in \{0, 1\}$, $g_0^i = g_m^i = 1$, and $i = 1, 2, \dots, n$. For this interleaved convolutional code encoder, codeword (encoder output) sequences are generated as follows:

$$C^i(z^I) = G^i(z^I)X(z) = \sum_{j=0}^m g_j^i z^{-jI} \cdot \sum_{k=0}^{\infty} a_k z^{-k}. \quad (5)$$

Generator polynomials, for the case of $n = 2$, $m = 2$, and $I = 2$, and with $g(D) = (7, 5)_8$ for original convolutional code, are

$$\begin{aligned} G^1(z^2) &= 1 + z^{-2} + z^{-4}, \\ G^2(z^2) &= 1 + z^{-4}. \end{aligned} \quad (6)$$

Codeword (encoder output) sequences of this encoder are

$$\begin{aligned} C^1(z^2) &= G^1(z^2)X(z) = C_0^1(z^2) + C_1^1(z^2), \\ C^2(z^2) &= G^2(z^2)X(z) = C_0^2(z^2) + C_1^2(z^2), \end{aligned} \quad (7)$$

where

$$\begin{aligned}
C_0^1(z^2) &= a_0 + z^{-2}\{a_0 + a_2\} + z^{-4}\{a_0 + a_2 + a_4\} \\
&\quad + z^{-6}\{a_2 + a_4 + a_6\} + z^{-8}\{a_4 + a_6 + a_8\} + \dots, \\
C_0^2(z^2) &= a_0 + a_2z^{-2} + z^{-4}\{a_0 + a_4\} + z^{-6}\{a_2 + a_6\} \\
&\quad + z^{-8}\{a_4 + a_8\} + \dots, \\
C_1^1(z^2) &= a_1z^{-1} + z^{-3}\{a_1 + a_3\} + z^{-5}\{a_1 + a_3 + a_5\} \\
&\quad + z^{-7}\{a_3 + a_5 + a_7\} + z^{-9}\{a_5 + a_7 + a_9\} + \dots, \\
C_1^2(z^2) &= a_1z^{-1} + a_3z^{-3} + z^{-5}\{a_1 + a_5\} + z^{-7}\{a_3 + a_7\} \\
&\quad + z^{-9}\{a_5 + a_9\} + \dots.
\end{aligned} \tag{8}$$

Two independent codeword sequences are obtained by interleaving with degree 2: $(C_0^1(z^2), C_0^2(z^2))$ and $(C_1^1(z^2), C_1^2(z^2))$. They are transmitted alternatively. Extra delays are used for one codeword sequence to add more interleaving effect. In this case, the decoder also requires extra delays to adjust timing of received sequences as shown in Figure 1.

3. INTERLEAVED VITERBI DECODING

Viterbi decoding algorithm consists of branch metrics calculation, add-compare-select (ACS) operation, and estimation of the information sequence from the survival path information. Hamming distance (hard decision) or Euclidean distance (soft decision) between the received data and the possible codeword sequences are computed in the branch metrics calculation unit. Those branch metrics are accumulated and the most likely path (survival path) is selected by the ACS unit. For a binary convolutional code with the code rate (R) is k/n , the number of possible codeword is 2^n . From the survival path information, the decoded data sequence is obtained.

Interleaved Viterbi decoding algorithm is based on the decomposed trellis diagram. The trellis diagram of an (n, k, mI) interleaved convolutional code can be decomposed to $I \times (n, k, m)$ trellis diagrams. Figure 3 shows the decomposed trellis diagram of $(2, 1, 2 \times 2)$ NRNSC. As we can see in Figure 3, each decomposed (n, k, m) trellis diagram is identical.

A received sequence, which may be corrupted by errors, can be represented as

$$R^i(z) = \sum_{k=0}^{\infty} r_k^i z^{-k}, \quad i = 1, 2, \dots, n. \tag{9}$$

From these sequences, branch metrics can be calculated as

$$\begin{aligned}
\Lambda^p(z) &= \text{bm} \{(r_0^1, r_0^2, \dots, r_0^n), (c^1, c^2, \dots, c^n)\} \\
&\quad + \text{bm} \{(r_1^1, r_1^2, \dots, r_1^n), (c^1, c^2, \dots, c^n)\} z^{-1} + \dots \\
&= \sum_{k=0}^{\infty} \lambda_k^p z^{-k}, \quad p = 0, \dots, 2^n - 1 \text{ for } R = \frac{k}{n},
\end{aligned} \tag{10}$$

where $\text{bm}\{(r_k^1, r_k^2, \dots, r_k^n), (c^1, c^2, \dots, c^n)\}$ and (c^1, c^2, \dots, c^n)

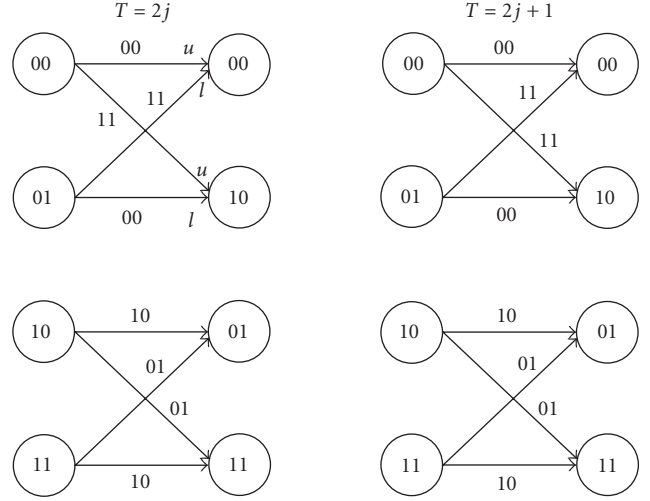


FIGURE 3: Decomposed trellis diagram for $(2, 1, 2 \times 2)$ interleaved convolutional code in Figure 2a.

represent the branch metrics and the possible codeword, respectively. Using this branch metrics, ACS operations can be executed as

$$z^{-(k+1)I-j} \gamma^s(z) = Z^{-kI-j} \min \{\lambda_k^u + \gamma^u, \lambda_k^l + \gamma^l\}, \tag{11}$$

where λ_k^s and γ^s represent branch metrics and state metrics (or path metrics or accumulated state metrics), respectively; s stands for trellis state, which varies from 0 to $2^m - 1$; $k = 0, 1, 2, \dots, \infty$; and $j = 0, 1, 2, \dots, I - 1$. The superscripts u and l in (11) mean, respectively, upper and lower branches that merged into a trellis state (see Figure 3). The survivor path information (referred to as path select signal, PS) of this ACS operation is as follows:

$$\text{PS}^s(z) = \sum_{k=0}^{\infty} p_k^s z^{-k}, \quad s = 0, \dots, 2^m - 1, \quad p_k^s \in \{0, 1\}, \tag{12}$$

where p_k^s is 0 when the upper branch is selected and p_k^s is 1 when the lower branch is selected for a trellis state s . From (11), (12), and Figure 3, we know that I delays (or storage elements) are needed to guarantee proper ACS operations.

Viterbi decoder consists of branch metrics calculator (BMC), ACS units, state metrics memory (referred to as SMM or path metrics memory), and path memory (PM or survival path memory). For fast decoding applications, state-parallel architecture, which uses as many ACS units as the number of trellis states, is generally used with register exchange path memory structure. The BMC computes Euclidean or Hamming distance between the received data and codeword sequences. Generally, Euclidean distance is used to get better coding gain. The ACS unit selects most probable path by comparing the accumulated branch metrics. The accumulated branch metrics, which are resulted from ACS operation, are stored in SMM and the selected path information (PS) is stored in PM.

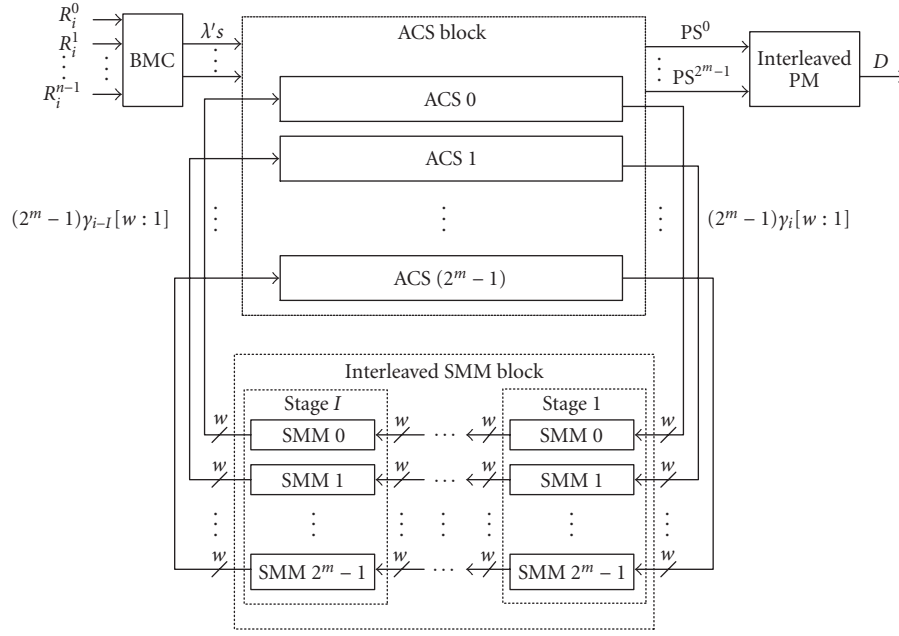


FIGURE 4: A proposed Viterbi decoder architecture for an $(n, 1, mI)$ interleaved convolutional code.

In case of $n = 2$, $m = 2$, and $I = 2$, received information sequences are represented as

$$R^i(z) = \sum_{k=0}^{\infty} r_k^i z^{-k}, \quad i = 1, 2. \quad (13)$$

The BMC computes branch metrics as follows:

$$\begin{aligned} \Lambda^0(z) &= \sum_{k=0}^{\infty} \lambda_k^0 z^{-k} = \sum_{k=0}^{\infty} \text{bm} \{(r_k^1, r_k^2), (0, 0)\} z^{-k}, \\ \Lambda^1(z) &= \sum_{k=0}^{\infty} \lambda_k^1 z^{-k} = \sum_{k=0}^{\infty} \text{bm} \{(r_k^1, r_k^2), (0, 1)\} z^{-k}, \\ \Lambda^2(z) &= \sum_{k=0}^{\infty} \lambda_k^2 z^{-k} = \sum_{k=0}^{\infty} \text{bm} \{(r_k^1, r_k^2), (1, 0)\} z^{-k}, \\ \Lambda^3(z) &= \sum_{k=0}^{\infty} \lambda_k^3 z^{-k} = \sum_{k=0}^{\infty} \text{bm} \{(r_k^1, r_k^2), (1, 1)\} z^{-k}, \end{aligned} \quad (14)$$

where $\Lambda^0(z)$, $\Lambda^1(z)$, $\Lambda^2(z)$, and $\Lambda^3(z)$ represent branch metrics between the received symbols and the possible code words $(0, 0)$, $(0, 1)$, $(1, 0)$, and $(1, 1)$, respectively. These branch metrics are used in ACS computation. The ACS unit adds branch metrics (λ) and previous state metrics (y), and then selects minimum state metrics from two incoming branches (see Figure 3) as follows:

$$z^{-2(k+1)-j} \gamma^s(z) = z^{-2k-j} \min \{\lambda_k^u + \gamma^u, \lambda_k^l + \gamma^l\}, \quad (15)$$

where $k = 0, 1, 2, \dots, \infty$ and $j = 0, 1$. The survivor path information is

$$\text{PS}^s(z) = \sum_{k=0}^{\infty} p_k^s z^{-k}, \quad s = 0, \dots, 3. \quad (16)$$

The selected state metrics are stored in the SMM as a new state metrics. For $m = 2$, which means that the number of trellis states are $2^m = 2^2 = 4$, we need four PS signals: $\text{PS}^0(z)$, $\text{PS}^1(z)$, $\text{PS}^2(z)$, and $\text{PS}^3(z)$. These PS signals go into the PM.

Since the number of trellis states of interleaved convolutional code, which is 2^{mI} for the (n, k, mI) interleaved convolutional code with interleaving degree is I , for large interleaving degree or encoder constraint length ($K = m + 1$) is very large, a straightforward state-parallel implementation of the Viterbi algorithm for this code requires huge hardware resources. For $(2, 1, 2 \times 4)$ interleaved convolutional code, the number of trellis states is 256, which is the same as for $K = 9$. Therefore, area-efficient high-speed Viterbi decoder architecture for interleaved convolutional code is needed.

By substituting I delays for each delay (or storage) element in SMM and path memory cell (PMC) of PM, an area-efficient high-speed interleaved Viterbi decoder architecture for interleaved convolutional code is obtained. In this architecture, we can get the throughput rate of the Viterbi decoder as high as the operating clock speed. Since the decoding latency of the state-parallel Viterbi decoder with register exchange path memory structure is the same as the decoding depth, the decoding latency of the interleaved Viterbi decoder is increased by $I \times \text{DD}$. Therefore, the decoding latency of proposed architecture is the decoding depth multiplied by the interleaving degree, that is, decoding latency = $\text{DD} \times I$. Since interleaved convolutional coding scheme uses extra delay (A), its overall decoding latency becomes $\text{DD} \times I + A$.

A proposed state-parallel Viterbi decoder architecture for interleaved $(n, 1, mI)$ convolutional code is shown in Figure 4.

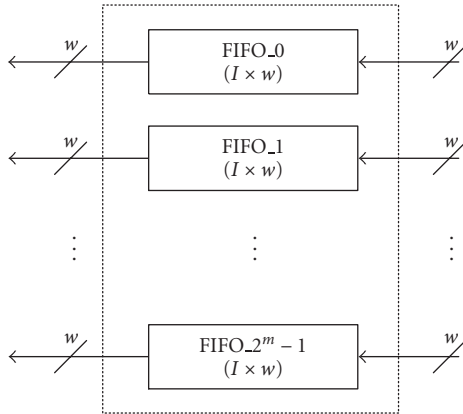


FIGURE 5: Interleaved SMM architecture using FIFO.

If the decoding speed is not critical, state-serial architecture, which uses less ACS units than the number of trellis states without changing SMM and PM structures, can be used. But it needs a control unit for proper connection between ACS units and SMM and PM. The BMC and the ACS unit architectures of the proposed Viterbi decoder are identical with that of the original Viterbi decoder architecture.

In general, random access memory (RAM) and D -type flip-flop are used as an SMM for the state-serial and state-parallel noninterleaved Viterbi decoders, respectively. For both cases, its size becomes interleaving degree (I) \times number of trellis states (2^m) \times state metrics width (w). Figure 5 shows an alternative SMM architecture, which uses first-in first-out (FIFO) memory.

Interleaved PM and interleaved PMC (IPMC) architectures for proposed interleaved Viterbi decoder are shown in Figure 6.

The basic architecture of this interleaved PM is exactly the same as the architecture of the original register exchange PM architecture. However, it uses modified PM cell architecture that consists of one multiplexer and I storage elements as shown in Figure 6b. D -type flip-flop is generally used for storage element in register exchange PM structure. Due to the extra delay elements in IPMC, the estimated information sequence can be properly recovered from the PM. Also by virtue of its simple structure, placement and routing of path memory cells are easier than that of a straightforward implementation. Reduction of power consumption is also expected in this proposed Viterbi decoder architecture. The PS^0 , PS^1 , PS^2 , and PS^3 are used as select signals for the first, second, third, and fourth row of IPMC in PM, respectively. The connection of IPMC in PM is exactly the same as the trellis diagram. The path select signals can be used as inputs of the IPMC for the first column in PM. When the DD is large enough, that is, $DD \geq 4K$, the outputs of the IPMC at the last column in PM have the same values with very high probability. Therefore, some IPMC in PM can be

removed with ignorable performance degradation as shown in Figure 6a.

The Viterbi decoder for interleaved convolutional code also can be implemented in I -parallel manner. It consists of I -parallel Viterbi decoder components. Each Viterbi decoder component is used for decoding each interleaved data sequence.

In Table 1, the complexity, latency, and throughput rate of this proposed Viterbi decoder architecture are compared with a straightforward implementation.

From Table 1, we can see that the hardware complexity of the proposed Viterbi decoder architecture is much smaller than that of the straightforward implementation for the same throughput rate. For $I = 2$ and $m = 2$, we can achieve hardware reduction of 75% for ACS, 50% for SMM, and 50% for PM, approximately. Furthermore, the connections of proposed architecture are reduced. The proposed interleaved Viterbi decoder architecture saves areas for the ACS units and PM. Since the IPMC uses less number of multiplexers, the size of IPMC is smaller than that of ($I \times$ PMC) as shown in Figure 6b.

However, the latency of this proposed architecture, which is linearly increased with the interleaving degree, is the largest among three different implementations.

4. CONCLUSION

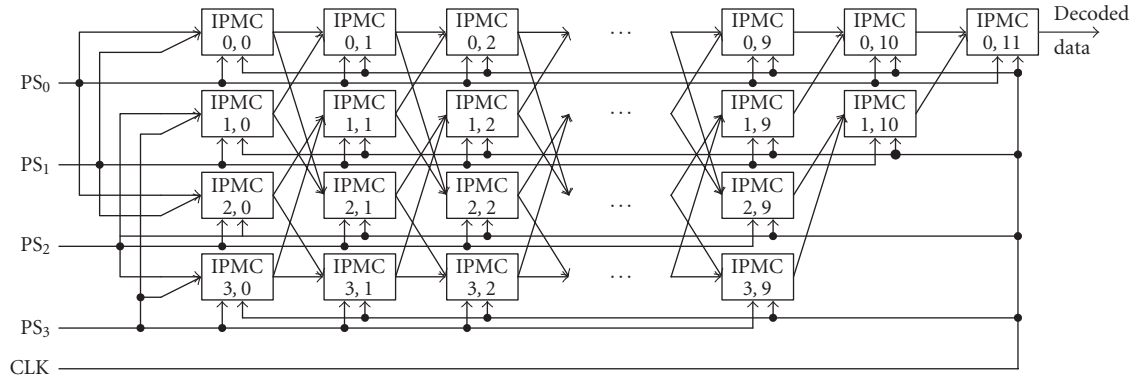
An area-efficient high-speed Viterbi decoder architecture is proposed to decode $(n, 1, mI)$ interleaved convolutional code. By replacing each delay (or storage) element in state metrics memory and path memory with I delays, interleaved Viterbi decoder is obtained. More hardware complexity reduction can be achieved with higher interleaving degree. It means that this proposed architecture is more area efficient for interleaved Viterbi decoder with higher interleaving degree.

However, it is inevitable that the latency of this proposed architecture is increased as the interleaving degree is increased. The latency of proposed interleaved Viterbi decoder itself is “decoding depth (DD) \times interleaving degree (I),” which is linearly increased with the interleaving degree. Since interleaved convolutional coding scheme uses extra delay (A), its actual decoding latency becomes $DD \times I + A$.

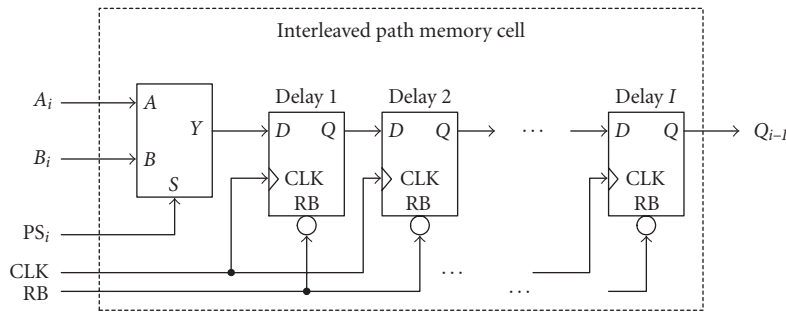
The performance of this interleaved convolutional coding scheme depends on the interleaving degree and the size of extra delay.

ACKNOWLEDGMENTS

The authors would like to thank anonymous reviewers for their comments and suggestions. The first author also appreciates Samsung Advanced Institute of Technology for financial support of his study. This research was supported by National Science Foundation (NSF) under Grant CCR-9988262. This paper was presented in part at the 36th Annual Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, Calif, November 2002.



(a)



(b)

FIGURE 6: (a) Interleaved PM for DD is 12 (b) IPMC architectures for $(n, k, 2I)$.

TABLE 1: Complexity, latency, and throughput rate of interleaved Viterbi decoder for an $(n, 1, mI)$ interleaved convolutional code.

Parameters	Proposed	Straightforward	I -parallel
No. of trellis states	2^m	2^{mI}	$I \times 2^m$
No. of ACS	2^m	2^{mI}	$I \times 2^m$
Size of SMM	$I \times (\text{bit widths of SMM} \times 2^m)$	Bit widths of SMM $\times 2^{mI}$	$I \times (\text{bit widths of SMM} \times 2^m)$
Size of PM	$DD \times 2^m \times \text{IPMC}$ $\approx DD \times 2^m \times I \times \text{PMC}$	$DD \times 2^{mI} \times \text{PMC}$	$I \times DD \times 2^m \times \text{PMC}$
Latency	$DD \times I + A$	DD	DD
Throughput rate	Same as ACS clock speed		

REFERENCES

[1] C. B. Schlegel and M. A. Herro, "A burst-error-correcting Viterbi algorithm," *IEEE Trans. Communications*, vol. 38, no. 3, pp. 285–291, 1990.

[2] D. J. van Wyk and L. P. Linde, "Application of a burst-error-correction Viterbi algorithm to Q²PSK on Rician fading channels," in *Proc. IEEE 4th Africon Conference in Africa (AFRICON '96)*, vol. 2, pp. 576–581, Stellenbosch, South Africa, September 1996.

[3] S. Lin and D. J. Costello Jr, *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1983.

[4] N. Benvenuto, L. Bettella, and R. Marchesani, "Performance of the Viterbi algorithm for interleaved convolutional codes," *IEEE Trans. Vehicular Technology*, vol. 47, no. 3, pp. 919–923, 1998.

[5] N. Benvenuto and R. Marchesani, "The Viterbi algorithm for sparse channels," *IEEE Trans. Communications*, vol. 44, no. 3, pp. 287–289, 1996.

[6] K. Y. Wong and K. Takahashi, "Cell loss recovery using interleaved convolutional code for ATM networks," *IEE Electronic Letters*, vol. 36, no. 13, pp. 1126–1127, 2000.

[7] D.-F. Yuan, Z.-W. Li, A.-F. Sui, and J.-J. Luo, "Performance of interleaved (2,1,7) convolutional codes in mobile image communication system," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC '00)*, vol. 2, pp. 634–637, Chicago, Ill, USA, September 2000.

[8] L.-F. Peng, D.-F. Yuan, Z.-W. Li, and D.-F. Guo, "Application of wavelet noise cancellation and interleaving techniques in

image transmission system over mobile fast-fading channels,” in *Proc. IEEE Wireless Communications and Networking Conference (WCNC '00)*, vol. 3, pp. 1492–1495, Chicago, Ill, USA, September 2000.

- [9] L.-F. Peng, D.-F. Yuan, Z.-W. Li, and D.-F. Guo, “Performance of the combination of interleaving and wavelet noise cancellation in mobile image transmission system,” in *IEEE Proc. 11th IEEE International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC '00)*, vol. 1, pp. 237–240, London, UK, September 2000.
- [10] J. J. Kong and K. K. Parhi, “Viterbi decoder architecture for interleaved convolutional code,” in *Proc. Conference Record of the 36th Asilomar Conference on Signal, Systems, and Computers (Asilomar '02)*, vol. 2, pp. 1934–1937, Pacific Grove, Calif, USA, November 2002.

Jun Jin Kong received the B.S. and M.S. degrees from the Department of Electronics Engineering in Hanyang University, Korea, in 1986 and 1988, respectively. Since 1989, he has worked in Samsung Advanced Institute of Technology and Central R&D Center in Samsung Electronics Co., Ltd., Korea. He has developed application-specific integrated circuits (ASIC) for communication systems and storage systems. He is an inventor of several patents on error correcting code decoders. His research in error correcting codes addresses effective decoding algorithms and VLSI architectures. He served as a steering committee member of the Coding and Information Society in Korean Institute of Communication Sciences. He is currently pursuing for Ph.D. degree at the Department of Electrical and Computer Engineering in the University of Minnesota, Minneapolis, USA as the recipient of the Samsung Electronics Corporation, Ltd. scholarship. His current research interests include quantum error-correcting codes and their decoder architectures.



Keshab K. Parhi is a Distinguished McKnight University Professor in the Department of Electrical and Computer Engineering at the University of Minnesota, Minneapolis. He was a Visiting Professor at Delft University and at Lund University, a Visiting Researcher at NEC Corporation, Japan (as a Fellow of the National Science Foundation of Japan), and a Technical Director of DSP Systems at Broadcom Corporation in its Office of CTO. Dr. Parhi's research interests have spanned the areas of VLSI architectures for digital signal and image processing, adaptive digital filters and equalizers, error control coders, cryptography architectures, high-level architecture transformations and synthesis, low-power digital systems, and computer arithmetic. He has published over 350 papers in these areas, authored the widely used text book *VLSI Digital Signal Processing Systems* (Wiley, 1999), and coedited the reference book *Digital Signal Processing for Multimedia Digital Signal Processing Systems* (Wiley, 1999). He has received numerous best paper awards including the most recent 2001 IEEE WRG Baker Prize Paper Award. He is a Fellow of IEEE and the recipient of a Golden Jubilee Medal from the IEEE Circuits and Systems Society in 1999. He is the recipient of the 2003 IEEE Kiyo Tomiyasu Technical Field Award.

