Design of a Power-Reduction Viterbi Decoder for WLAN Applications

Chien-Ching Lin, Yen-Hsu Shih, Hsie-Chia Chang, and Chen-Yi Lee, *Member, IEEE*

***Abstract*—In this paper, a 64-state four-bit soft-decision Viterbi decoder with power saving mechanism for high speed wireless local area network applications is presented. Based on path merging and prediction techniques, a survivor memory unit with hierarchical memory design is proposed to reduce memory access operations. It is found that more than 70% memory access can be reduced by taking advantage of locality. Moreover, a low complexity compare–select–add unit is also presented, leading to save 15% area and 14.3% power dissipation as compared to conventional add–compare–select design. A test chip has been designed and implemented in 0.18-μm standard CMOS process. The test results show that 30 $\sim$ 40% power dissipation can be reduced, and the power efficiency reaches 0.75 mW per Mb/s at 6 Mb/s and 1.26 mW per Mb/s at 54 Mb/s as specified in IEEE 802.11a.**

***Index Terms*—Add–compare–select, path merging, path prediction, survivor memory, Viterbi decoder.**

I. Introduction

Modern digital communication systems, especially wireless local area network (WLAN) systems, are required to transmit information at high data rates. This results in increased system complexity and power dissipation issues in circuit implementation. Furthermore, to enhance overall system performance, an efficient error-control code is often employed. Convolutional codes that have been widely exploited in communication systems provide a superior error correction capacity while keeping a reasonable coding complexity. Viterbi algorithm is the optimal solution for decoding convolutional codes [1], [2] with the modest computing resource. However, as the requirement of data rate increases in wireless applications, the power consumption becomes an obvious design issue in system-level integrated circuit. In this paper, exploring the system level behavior, redundant operations can be removed to achieve a better system architecture in terms of power dissipation and complexity.

The Viterbi decoder contains three main units [3]: transition metric unit (TMU), add–compare–select unit (ACSU), and survivor memory unit (SMU). As illustrated in Fig. 1, TMU calculates the transition metrics from the input data. The ACSU recursively accumulates transition metrics (TM) as path metrics (PM), and makes decisions to select the most likely state transition sequence. Finally, the SMU traces the decisions to extract this sequence. However, the nonlinear and recursive nature limits the maximum achievable throughput rate. The most common solution to develop a high throughput Viterbi decoder is fully parallel approach, where ACSUs are assigned to each state with high radix structure [4]. However, as the constraint length rises, the hardware complexity increases exponentially, and so does the power consumption. The first problem is the large number of ACS operations, where 2^{K-1} ACS operations are required for each iteration, and K is the constraint length. The second problem is that implementing a high speed traceback unit is more difficult than that based on the register exchange method [5] because of the limited bandwidth in the embedded memory. However, traceback approach provides a power efficient solution as the constraint length becomes larger. The traceback algorithm, k-pointer algorithm, for survivor memory management has been proposed in [6] and [7]. It divides the memory into banks and accesses them concurrently to achieve the demanded data bandwidth. The other approach is one-pointer algorithm which requires higher memory access rates. Both methods will consume much power due to a large amount of memory access operations.

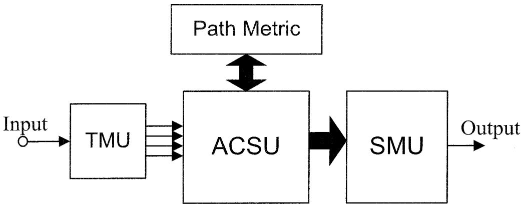


Fig. 1. Block diagram of Viterbi decoder.

The truncation length determines the size of survivor memory and also the decoding latency. It has been found that truncation lengths of four to five times the constraint lengths is sufficient to ensure negligible performance degradation [8], [9]; the traced path will remerge to the correct path within the truncation length with a high probability. To preserve the functionality and performance, truncation length, as a function of code rates (R) and channel capacity, is conventionally set to maximum. This will lead to many redundant operations during traceback, for the operating condition is not always the worst.

The proposed design targets the WLAN system specified in IEEE 802.11a [10]. Based on orthogonal frequency division multiplexing (OFDM) and forward error correction (FEC) coding, the system is able to transmit data with data rates up to 54 Mb/s. Both phase-shift keying (PSK) modulation and quadrature amplitude modulation (QAM) are included to provide various data rates listed in Table I. The FEC coding

Manuscript received February 25, 2004; revised May 27, 2004, and August 23, 2004. This work was supported by the National Science Council under Grant NSC 91-2218-E-009-010 and by MOEA of Taiwan, R.O.C., under Grant 91-EC-17-A-03-S1-005. This paper was recommended by Associate Editor Z. Wang.

The authors are with the Department of Electronics Engineering, National Chiao Tung University, Hsinchu 30050, Taiwan, R.O.C. (e-mail: cclin@si2lab.org).

Digital Object Identifier 10.1109/TCSI.2005.849106

TABLE I
TRANSMISSION MODES OF IEEE 802.11a WLAN

Data rate (Mb/s)	Modulation	Coding rate (R)	Design SNR for FEC (dB)
6	BPSK	1/2	$1 \sim 2$
9	BPSK	3/4	$3 \sim 4$
12	QPSK	1/2	$4 \sim 5$
18	QPSK	3/4	$6 \sim 7$
24	16-QAM	1/2	$9 \sim 10$
36	16-QAM	3/4	$13 \sim 14$
48	64-QAM	2/3	$17 \sim 18$
54	64-QAM	3/4	$18 \sim 19$

employs rate 1/2 convolutional code and derives higher rates from it by puncturing. The design signal-to-noise ratios (SNRs) for FEC are the targets to achieve a packet error rate (PER) of 10% in additive white Gaussian noise (AWGN) channel.

This paper proposes a modified traceback scheme [11] that reduces memory access based on the path merging property [2], which will be discussed in Section II. Section III introduces the path prediction algorithm that makes path merging work better in SMU. Section IV presents the proposed architecture including the soft-decision based TMU, the compare-select-add (CSA) structure, and the hierarchical memory based survivor memory design. The chip implementation and test results are described in Section V, and the conclusion is given in Section VI.

II. TRACEBACK WITH DYNAMIC TRUNCATION LENGTH

Due to practical storage constraints, the survivor memory for each state should be limited to a finite length, as shown in Fig. 2(a). With a truncation length of T, the decoder is required to output data on the branch at depth t by $t+T$ [9]. If all surviving paths have a common node at time t, the unique branch is chosen. Otherwise, the branch corresponding to the best metric value at depth $t+T$ is the choice. This truncation technique will result in an additional error if an incorrect path diverges from the correct path at depth t, and remains unmerged from it before depth $t+T$. Therefore, T must be chosen, so that truncation error is comparable to or less than maximum-likelihood decoding (MLD). In [8], the criterion of how to select a proper truncation length T is shown to be

$$\frac{T}{K} \geq \frac{E(R)}{E_c(R)} \tag{1}$$

where $E(R)$ is the block coding exponent, and $E_c(R)$ is the convolutional coding exponent as defined in [8].

To derive a more intuitive formulation of (1), the very noisy channel will be applied and assumed to be discrete and memoryless [8]. Let X be the set of symbols which can be transmitted on a given channel, and let Y be the set of symbols can be received. The conditional probability of receiving $y \in Y$, given that $x \in X$ is transmitted, can be expressed by

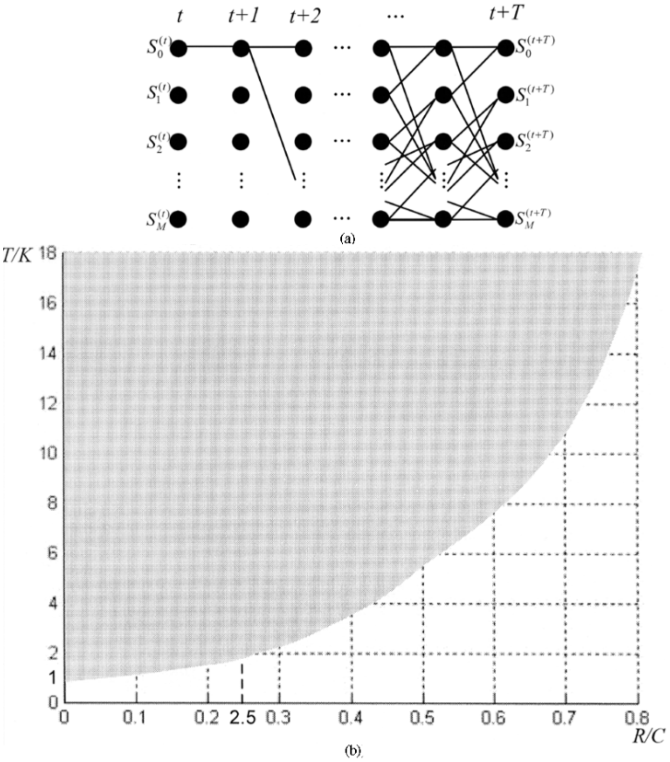


Fig. 2. (a) M-state trellis diagram truncated to T time instances, and $M = 2^{K-1}$. (b) Truncation length versus coding rate.

$$p(y \mid x) = p(y)(1 + \epsilon_{xy}) \tag{2}$$

where $|\epsilon_{xy}| \ll 1$ for all x and y, and $\sum_{y \in Y} p(y)\epsilon_{xy} = 0$ for all x. The condition in (1) can then be reduced to

$$\frac{T}{K} \geq \begin{cases} \frac{1}{1-\frac{2R}{C}}, & 0 \leq R < \frac{C}{4} \\ \frac{1}{2(1-\sqrt{\frac{R}{C}})^2}, & \frac{C}{4} \leq R \leq \frac{C}{2} \\ \frac{1+\sqrt{\frac{R}{C}}}{1-\sqrt{\frac{R}{C}}}, & \frac{C}{2} < R < C. \end{cases} \tag{3}$$

For a different coding rate R, the truncation length ranges from K to infinite as R approaches channel capacity C. Of different R/C ratios, Fig. 2(b) shows the valid truncation length in gray region. And in real applications, truncation length T must vary with respect to R and C.

Fig. 2(a) illustrates the path merging or unification property [2], [6] which all survivor sequences will converge to the same state with a high probability after tracing back T time instances that depends on coding rates and channel conditions. For a fixed T, SMU will trace the same path that had been traced recently as the path remerges to the correct one. This implies that SMU tends to reuse data which have been used in previous traceback operations. Considering the data locality in survivor memory, a dynamic traceback mechanism can be implemented to reduce power consumption caused by a great number of memory access and large memory word width.

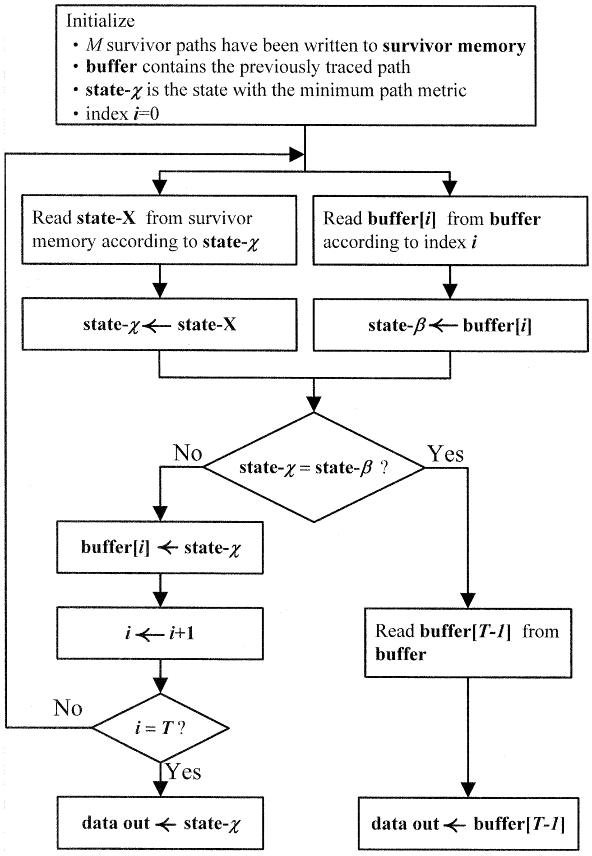

Initialize
• M survivor paths have been written to **survivor memory**
• **buffer** contains the previously traced path
• **state-χ** is the state with the minimum path metric
• index i=0
Read **state-X** from survivor memory according to **state-χ**
Read **buffer[i]** from **buffer** according to index i
state-χ ← state-X
state-β ← buffer[i]
state-χ = state-β ?
No
Yes
buffer[i] ← state-χ
i ← i+1
i = T ?
No
Yes
Read **buffer[T-1]** from **buffer**
data out ← state-χ
data out ← buffer[T-1]

Fig. 3. Path merging algorithm.

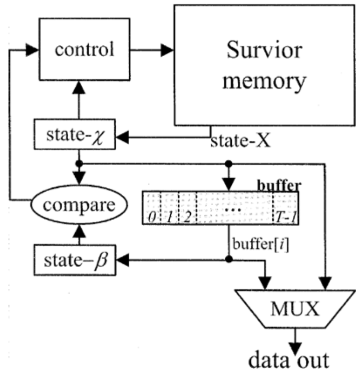


Fig. 4. Proposed memory management of SMU.

TABLE II
DISTRIBUTION OF PATH CONVERGENCE

SNR [†](dB)	2	3	4	5
1 time instance	90.82%	95.16%	97.58%	98.91%
2 time instances	93.33%	97.02%	98.81%	99.60%
3 time instances	94.03%	97.47%	99.05%	99.70%

[†] QPSK modulation and AWGN channel

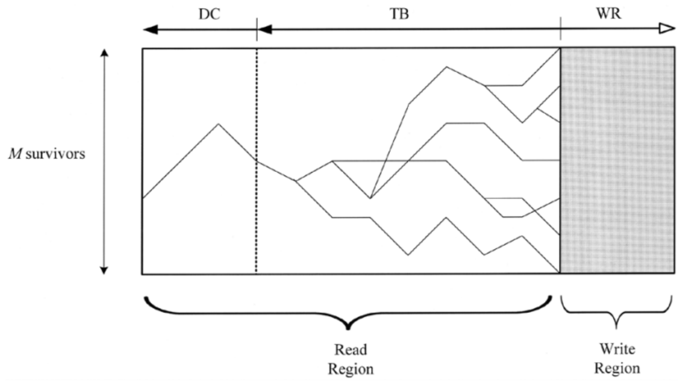


Fig. 5. Basic operations of SMU.

The modified traceback algorithm, path merging algorithm, can be summarized in Fig. 3 according to the cache based SMU design in Fig. 4. Initially, the survivor memory contains M survivor sequences with M equals to $2^{(K-1)}$. Since the correct path cannot be exactly known in receiver systems, the contents of buffer will be the path which is last traced. During tracing, state-χ is recursively updated and compared with the buffer element state-β. While state-χ is different from state-β, the contents of buffer should be revised to the new data state-χ. On the contrary, when the traced path merges to the previous one, SMU can stop further tracing operations because the buffer has contained the same state sequence.

Simulation results, shown in Table II, bring out the effects of the modified traceback algorithm in IEEE 802.11a [10] WLAN platform. These results represent the length required by the traced path to merge to the last one, and over 94% of paths will merge after tracing three time instances in different channel conditions. Therefore, many redundant operations can be eliminated through the proposed SMU design.

III. Path Prediction

In practical implementation, many algorithms are available for the memory management of SMU, and a generalized description has been given in [7], where k-pointer even and odd algorithms were presented. Fig. 5 shows the basic operations of SMU that are divided into three parts: writing new data (WR), traceback read (TB), and decode read (DC). The path merging algorithm can be successfully applied to TB operations, for the contents of buffer are iteratively updated if needed. However, in WR operation where new survivors from ACSU are written into memory, there is nothing to update the buffer, and nothing can be read from it during TB operation. Therefore, the path prediction algorithm accompanied with WR is proposed and shown in Fig. 6 where WR is assumed to process τ time instances. While ACSU proceeds each new time instance in the code trellis, the predicted state with the minimum PM is also found and verified that a valid transition exists. Once the sequence of states encounters an invalid transition, the prediction process should be stopped to avoid improper path merging during TB operation. In terms of different SNRs, the simulation results in Fig. 7 represent the accuracy of prediction, which is defined as the percentage of total predictable states in WR region.

Combined with the path merging algorithm, the updating procedure of buffer can be concluded in Fig. 8. The predicted state

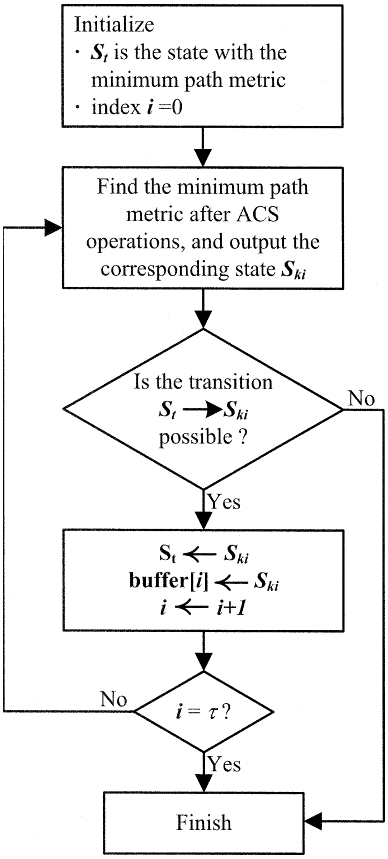

Initialize
· S_t is the state with the minimum path metric
· index i =0
Find the minimum path metric after ACS operations, and output the corresponding state S_{ki}
Is the transition $S_t \rightarrow S_{ki}$ possible ?
No
Yes
$S_t \leftarrow S_{ki}$
buffer[i] $\leftarrow S_{ki}$
$i \leftarrow i+1$
No
$i = \tau$?
Yes
Finish

Fig. 6. Path prediction algorithm.

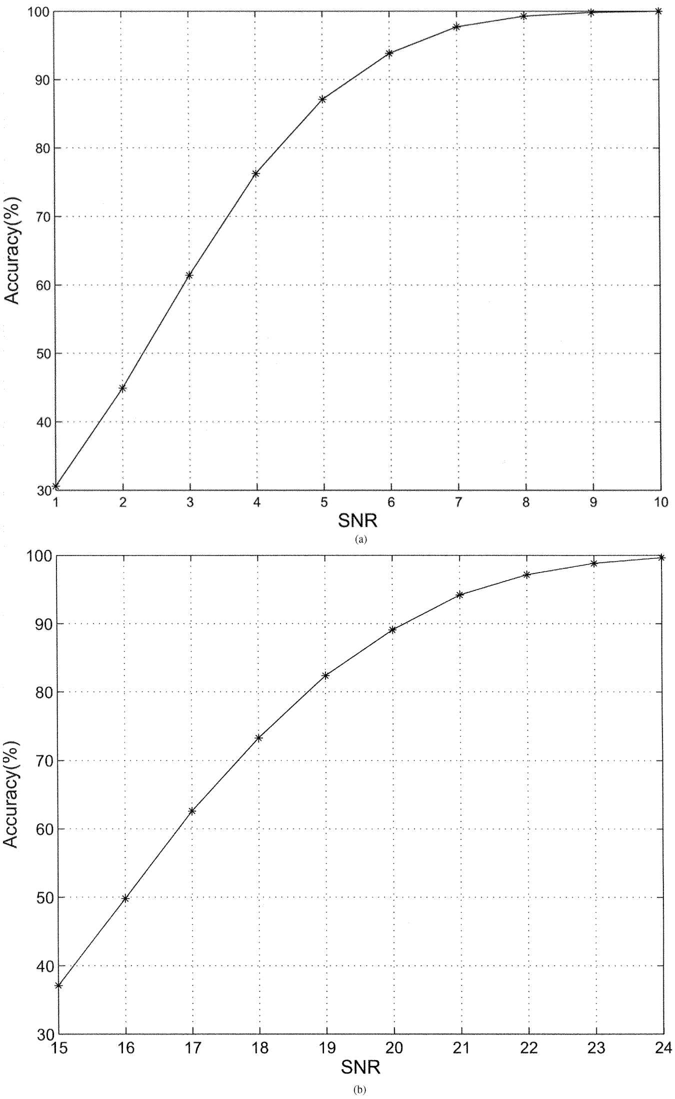


Fig. 7. Prediction accuracy in AWGN channel. (a) QPSK and $R = 1/2$. (b) 64-QAM and $R = 3/4$.

sequence is stored during WR operation while new state sequence is rewritten during TB operation until the traced path merges to that in the buffer. Even if incorrect states are predicted, they will be corrected in TB region, leading to no performance loss by applying the path prediction algorithm.

IV. Architecture Design

Fig. 9 shows the architecture of the proposed design. The de-puncture unit can support various coding rates defined in current applications. The TMU calculates the distance between the received symbols and codewords on the branches. The 64 parallel simplified ACS units perform comparison among candidate paths to determine survivors and compute the corresponding path metrics. The SMU based on the k-pointer even algorithm with $k = 3$ is constructed by 6-bank memory architecture [7]. And each memory bank equips a state buffer which retains the state sequence that would probably be reused. The memory management unit (MMU) governs the operations of SMU, including the path merging and path prediction operations. The comparator (CMP) computes the minimum PM and outputs the corresponding state to prediction unit as well as TB unit. The prediction unit will generate a possible state sequence to increase buffer reuse efficiency.

A. TMU Design

Because of the finite precision in practical implementation, quantizing channel symbols will increase the SNR to achieve the desired bit error rate (BER). A soft decision Viterbi decoder has better error correction capability, where its hardware complexity increases linearly with the number of the quantization bits in the demodulated symbols. The objective is to minimize the quantization loss in terms of hardware cost. The quantization level and stepsize vary with modulation types and channel conditions. For different quantization levels and modulation types, Table III summarizes the performance improvement in terms of SNR over the hard decision decoding. All of the quantization schemes are set to be uniform quantization and optimal stepsize [12]. While considering the indoor multipath channel, Rayleigh fading with a root mean square (rms) delay spread of 50 ns is considered. Simulation results of two extreme cases are also shown in Fig. 10. And there are slight improvements from 8-level to 16-level in BPSK case and from 16-level to 32-level in 64-QAM case. Therefore, to achieve a good compromise between performance and complexity, 16-level soft decision will be our choice.

B. ACS Design

The required ACS operations per trellis stage are equal to $2^{(K-1)}$. For different applications and design constraints, the

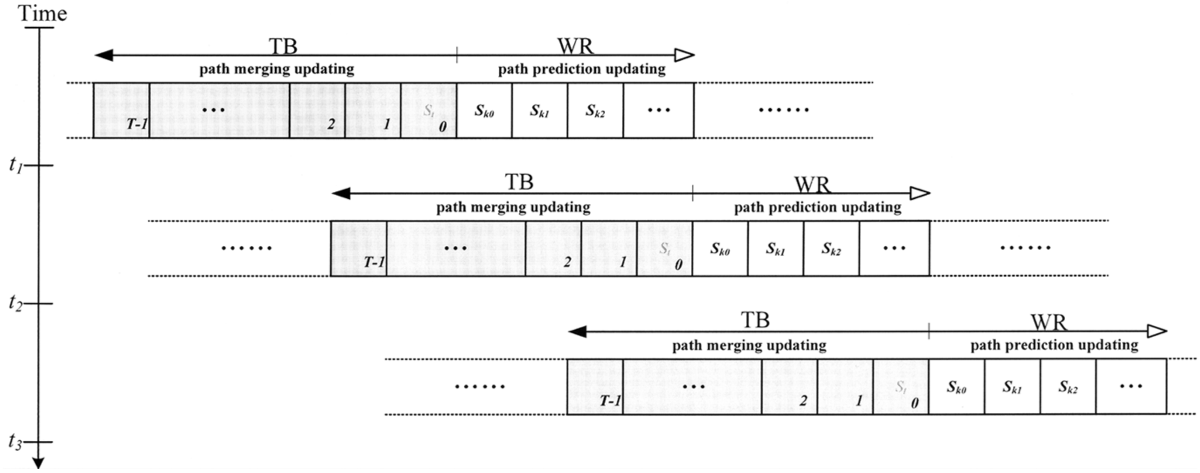


Fig. 8. Buffer updating operations.

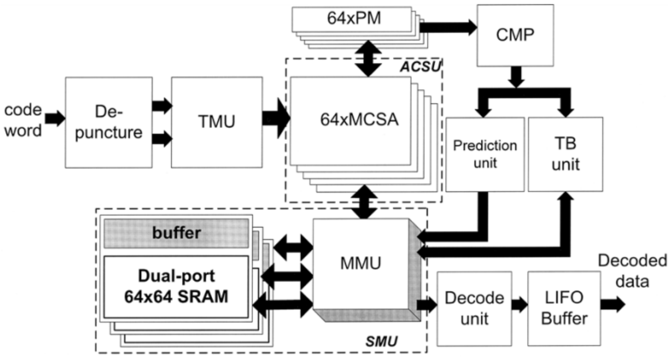


Fig. 9. Block diagram of the proposed Viterbi decoder.

TABLE III
IMPROVEMENT OF SOFT-DECISION VITERBI DECODER COMPARED TO HARD DECISION

	8-level	16-level	32-level
BPSK	1.7dB	1.9 dB	2dB
QPSK	1.7dB	1.9 dB	2dB
16-QAM	2.3dB	2.87 dB	2.96dB
64-QAM	2.2dB	2.6 dB	2.75dB

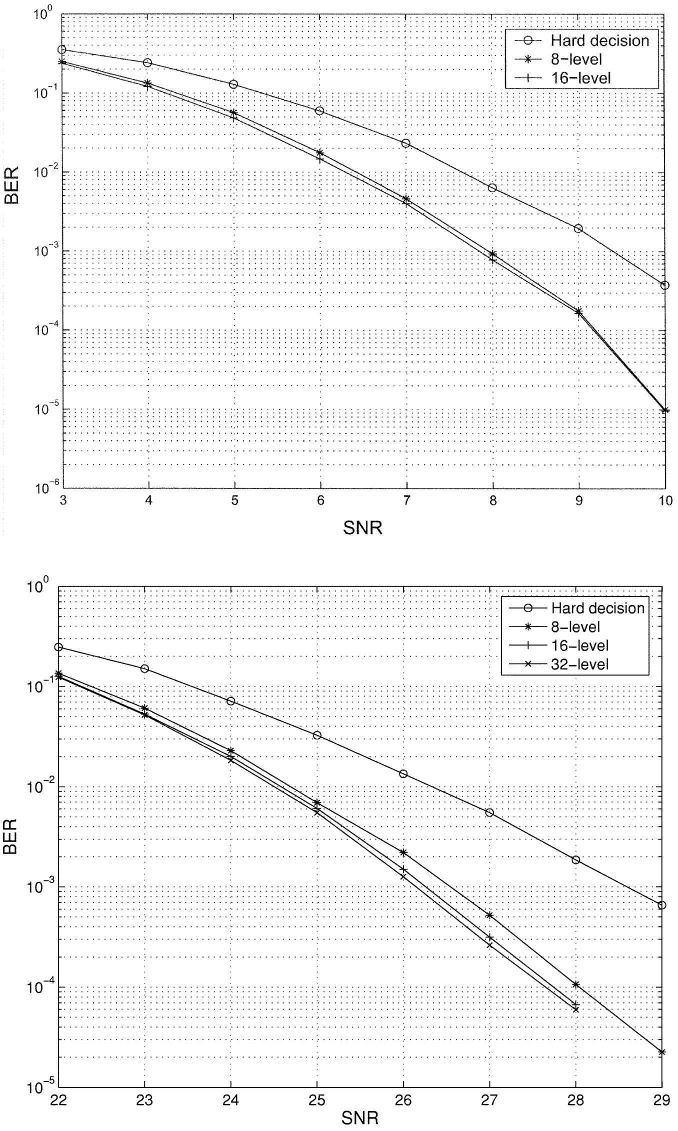
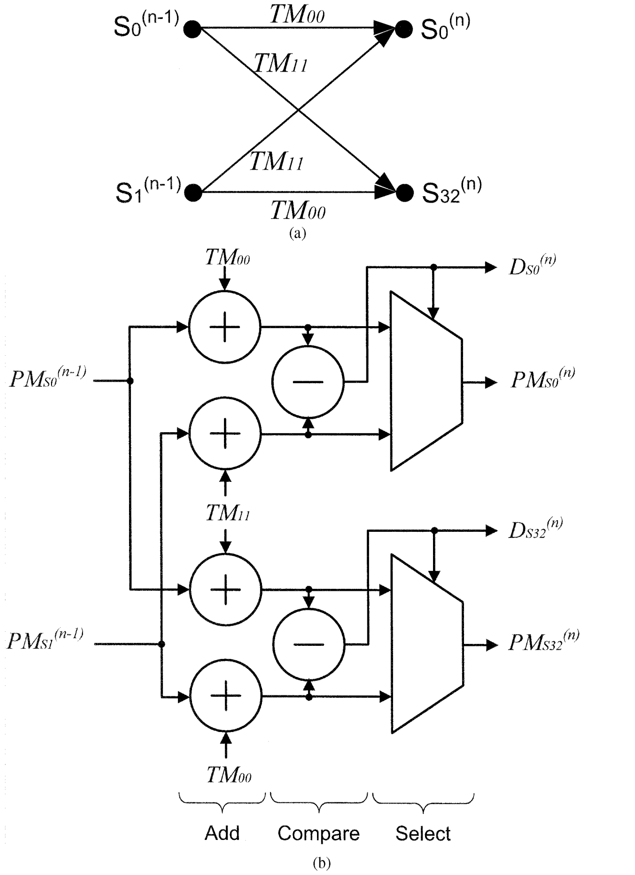


Fig. 10. Simulation results of the soft decision Viterbi decoder in multipath channel. (a) BPSK and $R = 1/2$. (b) 64-QAM and $R = 3/4$.

implementation approach ranges from fully parallel computing array to sharing the computational resources through multiplexing. As for the WLAN category, the fully parallel approach is preferred because the demanded data rate may reach decades of Mb/s or even hundreds of Mb/s. The maximum achievable throughput rate is limited owing to the nonlinear and recursive property. Thus, high radix ACS structure had been explored to achieve a high throughput decoder [4]. As a rule, the radix-4 architecture is used to provide double throughput with a corresponding complexity increase [13], [14]. A further improvement is the parallel architecture [15] where path metrics and decisions are calculated concurrently. The speedup is achieved with an expense of carry-save-adders.



(a)

(b)

Fig. 11. State transition and corresponding ACS structure. (a) Trellis diagram. (b) ACS structure.

ACS operations dominate a large computing power in conventional Viterbi decoder, and many techniques have been introduced to improve either the critical path or the power consumption. The transformations of ACS unit in [16]–[18] result in a CSA structure, leading to lower complexity of computations. The drawback of longer delay path will result in slower decoding speed. Another construction of CSA in [19] reaches a higher speed of computation, but requires twice more adders and path metric memory. A modified CSA (MCSA) is presented to improve the data-path delay in original CSA architectures while preserving lower complexity.

Fig. 11(a) shows the subset of the 64-state trellis with generator polynomial $g_0 = 133_8$ and $g_1 = 171_8$, and the corresponding ACS structure is shown in Fig. 11(b). The path metric updating can be expressed by

$$PM_{S0}^{(n)} = \min\left\{PM_{S0}^{(n-1)} + TM_{00},\ PM_{S1}^{(n-1)} + TM_{11}\right\} \quad (4)$$

$$PM_{S32}^{(n)} = \min\left\{PM_{S0}^{(n-1)} + TM_{11},\ PM_{S1}^{(n-1)} + TM_{00}\right\}. \quad (5)$$

The selection of minimum path metric is accomplished by the decisions $D_{S0}^{(n)}$ and $D_{S32}^{(n)}$.

$$D_{S0}^{(n)} = \operatorname{sign}\left\{PM_{S0}^{(n-1)} + TM_{00} - PM_{S1}^{(n-1)} - TM_{11}\right\} \quad (6)$$

$$D_{S32}^{(n)} = \operatorname{sign}\left\{PM_{S0}^{(n-1)} + TM_{11} - PM_{S1}^{(n-1)} - TM_{00}\right\}. \quad (7)$$

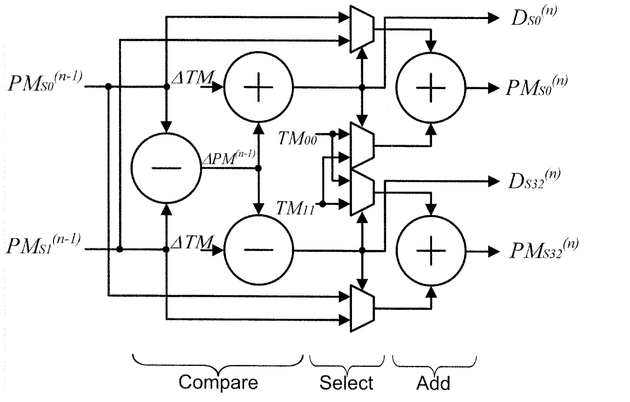


Fig. 12. CSA architecture.

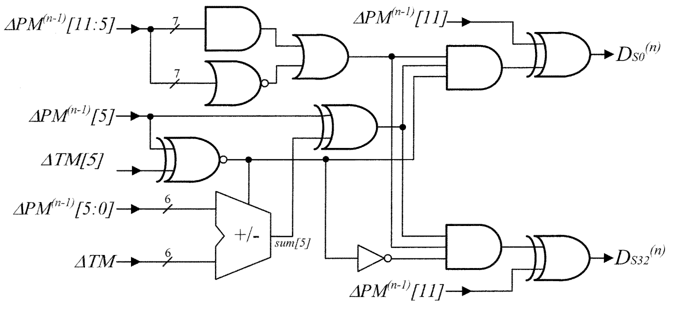


Fig. 13. Modified structure of comparator sub-unit.

Through arrangement, (6) and (7) will be reduced to

$$D_{S0}^{(n)} = \operatorname{sign}\left\{\Delta PM^{(n-1)} + \Delta TM\right\} \quad (8)$$

$$D_{S32}^{(n)} = \operatorname{sign}\left\{\Delta PM^{(n-1)} - \Delta TM\right\} \quad (9)$$

where $\Delta PM^{(n-1)} = PM_{S0}^{(n-1)} - PM_{S1}^{(n-1)}$ and $\Delta TM = TM_{00} - TM_{11}$. The identical terms in (8) and (9) indicate that the computations of $D_{S0}^{(n)}$ and $D_{S32}^{(n)}$ can share the same arithmetic unit. The reconstructed CSA unit is shown in Fig. 12. The calculation of ΔTM can be shared by many CSA units. And the resource sharing of ΔPM for $D_{S0}^{(n)}$ and $D_{S32}^{(n)}$ reduces one subtractor as compared to the conventional ACS. Furthermore, a modified comparator architecture is proposed in Fig. 13 to improve CSA performance. The lookahead approach is explored to speed up compare operations. As shown in (8) and (9), the signs of $\Delta PM^{(n-1)} + \Delta TM$ and $\Delta PM^{(n-1)} - \Delta TM$ are expected results and can be obtained by comparing the magnitudes of $\Delta PM^{(n-1)}$ and ΔTM, and by checking their difference in signs. Since $\Delta PM^{(n-1)}$ is a 12-bit number, and ΔTM is a 6-bit number, the comparison can be simplified to a 6-bit adder/subtractor as well as a carry propagation circuit. As a result, the circuit in Fig. 13 implements both (8) and (9) with a lower critical delay and complexity. Table IV summarizes the gate count and power dissipation of different ACS architectures, including the parallel ACS structure in [15]. In the modified CSA scheme, the area reduction from the conventional ACS structure is about 15%, and an average 14.3% power saving is observed through simulation.

TABLE IV
COMPARISON OF DIFFERENT TYPES OF 64-STATE ACS UNIT

	Conventional ACS	Parallel ACS[3]	CSA	MCSA
Gate count [1]	17.3k	32k	19.6k	14.7k
Critical path report [1]	6.21ns	5.75ns	8.28ns	8.06ns
Power dissipation [2]	28mW	31mW	43mW	24mW

[1] Synthesis constraints: 8.5ns timing constraint, zero target area, high mapping effort, and the worst operating condition.

[2] Simulated with 100MHz frequency and 1.8V supply.

[3] The parallel architecture is proposed in [15].

C. SMU Design

In terms of power consumption, memory traceback method that avoids moving data in the survivor memory is preferred. Compared with the register-exchange approach, traceback architecture has a limited memory bandwidth in nature, and thus limits the decoding speed. Therefore, there have been many memory management algorithms and architectures [6], [7], [20] proposed to improve data rates. However, the algorithms and architectures in the literature suffer a great deal of memory access and large memory word widths, leading to large power consumption in the survivor memory unit. As shown in (3), the truncation length is a function of coding rate and channel capacity. It is not appropriate to design the Viterbi decoder works in the worst cases. And for a fixed truncation length, the controller tends to trace the path which the controller traced recently. The locality of reference can be exploited to reduce memory access. With the above-mentioned features, the buffer based memory architecture is proposed. For each traceback process, the state transition sequence can be saved in a small buffer. At the next traceback, while the traced path is merging to the previous one, the data in buffer after the merged point must be the desired state sequence, and the memory read operation can be avoided. Furthermore, before traceback process, the initial state is chosen to have the minimum PM to accelerate the path merging process.

Based on the 3-pointer even algorithm proposed in [7], the survivor memory is divided into 6 banks, each of size $T/2$ words, and is constructed by three dual-port SRAMs. Combined with the buffer memory, the buffer efficiency will decrease to 66% as the buffer is empty during the first traceback. Thus, a path prediction algorithm discussed in Section III is applied to increase the buffer reuse efficiency. As shown in Fig. 8, the buffer is pre-written during WR operation and revised during TB operation.

Since the traced path is continuous in trellis, the buffer can be implemented to store the single continuous path, leading to save area. And the overhead that implements the buffer and additional control circuit is less than 6%. The maximum truncation length is set to 64, and each bank is sized to 32 × 64 bits. Each bank contains a 38 bits buffer that reduces a lot of memory read operations. The reduction of memory access in 54 Mb/s mode and AWGN channel is also shown in Fig. 14. In conventional SMU, the number of memory access is constant. On the other hand, in the proposed SMU the number of memory read operations degrades as the channel gets better. If the SNR is grater than 17 dB, the average memory access will be dominated by writing operations.

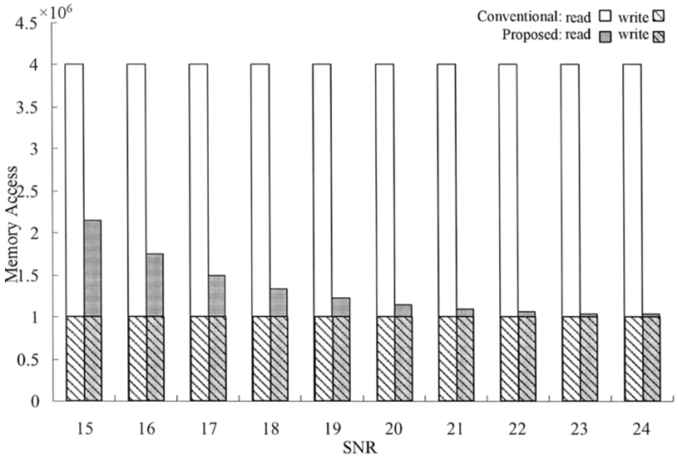


Fig. 14. Comparison of memory access operations.

V. CHIP IMPLEMENTATION AND TEST RESULTS

The chip is implemented by cell-based design flow and fabricated in 0.18-μm 1P6M CMOS process. The used cell is 1.8-volt SAGE-X standard cell library from Artisan. After synthesis, the gate level design contains about 49 k gates. And the post-layout simulation with 1.8 V supply at 25 °C shows that the proposed Viterbi decoder chip can work above 100-MHz clock rate, which will provide 75 Mb/s decoding rate at $R = 3/4$. The chip shown in Fig. 15 has been tested, and Table V summarizes the chip measurement results. Fig. 16 shows the measured power consumption in terms of different data rates listed in Table I. The conventional data in Fig. 16 is obtained from this chip with the path merging and path prediction functions being turned off. The more detailed information is illustrated in Fig. 17 where 54 Mb/s data rate is measured at $\mathrm{SNR} = 18$ dB. The increased power dissipation in clock tree and CMP unit is due to the additional buffer and the computation of the minimum PM. The power consumption which varies with the channel condition has

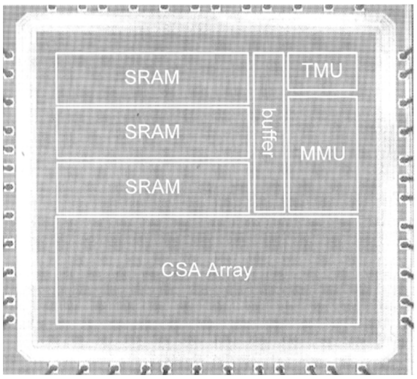


Fig. 15. Micro-photo of the Viterbi decoder test chip.

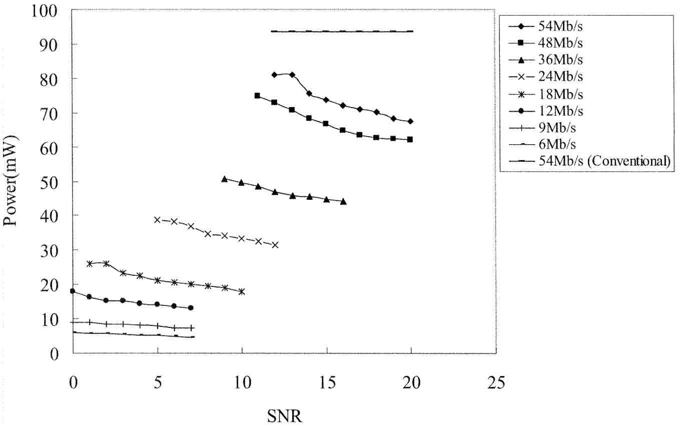


Fig. 16. Measured power consumption at different data rates defined in IEEE 802.11a.

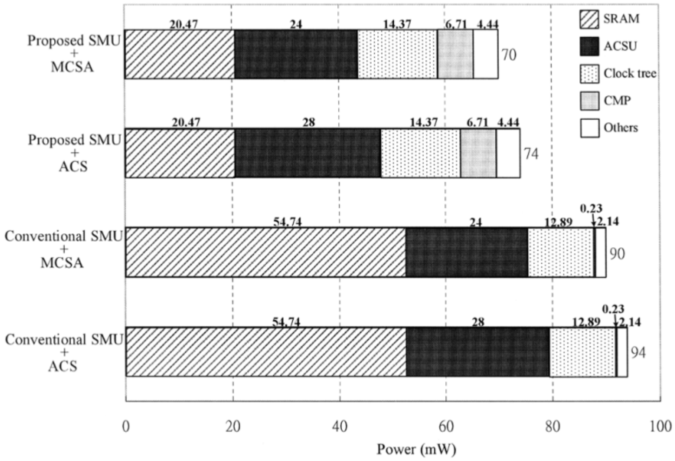


Fig. 17. Difference in power consumption before and after applying the proposed algorithm.

a $30\% \sim 40\%$ reduction as compared to conventional designs. And as shown in Table VI, the power efficiency at different rates reaches about 1 mW per Mb/s.

VI. Conclusion

In this paper, we have proposed a high performance and power reduction Viterbi decoder for WLAN applications. The modified CSA architecture lowers down hardware complexity as well as power dissipation. With the aid of path merging and prediction features, the memory access reduces more than 70% on the average. And the power consumption decreases due to the reduced memory access operations. The proposed design not only considers the error correction capacity, but also provides a high-speed and power-efficient solution.

TABLE V
Summary of Chip Implementation

Chip size	$2.37mm \times 2.37mm$
Core size	$1.75mm \times 1.75mm$
Gate count	49k
Embedded SRAM	12kbit
Supply voltage	1.64V ~ 1.98V
Operating frequency	100MHz
Average Power	68mW at 72MHz (54Mb/s and SNR=19dB)

TABLE VI
Power Efficiency at Different Data Rates

Data rate (Mb/s)	Target SNR (dB)	Power efficiency (mW per Mb/s)
6	1	0.96
9	3	0.92
12	4	1.2
18	6	1.15
24	9	1.43
36	13	1.28
48	17	1.32
54	18	1.30

Acknowledgment

The authors would like to thank Y. C. Tseng for layout assistance and C. C. Wu for the contribution in the early stage of this work. The authors also thank Chip Implementation Center for fabrication of the test chip. In addition, they want to thank the anonymous reviewers for many valuable comments to improve this paper's quality.

REFERENCES

[1] A. J. Viterbi, "Error bounds for convolutional codes and asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 2, pp. 260–269, Apr. 1967.

[2] J. G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.

[3] G. Fettweis and H. Meyr, "A 100 MBit/s Viterbi decoder chip: Novel architecture and its relization," in *Proc. IEEE Int. Conf. Communications (ICC)*, vol. 2, Aug. 1990, pp. 463–467.

[4] P. J. Black and T. H. Meng, "A 140-Mb/s, 32-state, radix-4, Viterbi decoder," *IEEE J. Solid-State Circuits*, vol. 27, no. 12, pp. 1877–1885, Dec. 1992.

[5] C. M. Rader, "Memory management in a Viterbi decoder," *IEEE Trans. Commun.*, vol. 29, no. 9, pp. 1399–1401, Sep. 1981.

[6] R. Cypher and C. B. Shung, "Generalized trace back techniques for survivor memory management in the Viterbi algorithm," in *Proc. GLOBECOM*, vol. 2, San Diego, CA, Dec. 1990, pp. 1318–1322.

[7] G. Feygin and P. Gulak, "Architectural tradeoffs for survivor sequence memory management in Viterbi decoders," *IEEE Trans. Commun.*, vol. 41, no. 3, pp. 425–429, Mar. 1993.

[8] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1979.

[9] J. G. D. Forney, "Convolutional codes II: Maximum-likelihood decoding," *Inf. Contr.*, vol. 25, pp. 222–266, Jul. 1974.

[10] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11a, 1999.

[11] C. C. Lin, C. C. Wu, and C. Y. Lee, "A low power and high speed Viterbi decoder chip for WLAN applications," in *Proc. 29th Eur. Solid State Circuits Conf. (ESSCIRC)*, Sep. 2003, pp. 723–726.

[12] I. M. Onyszchuk, K. M. Cheung, and O. Collins, "Quantization loss in convolutional decoding," *IEEE Trans. Commun.*, vol. 41, no. 2, pp. 261–265, Feb. 1993.

[13] A. K. Yeung and J. M. Rabaey, "A 210 Mb/s radix-4 bit-level pipelined Viterbi decoder," in *Dig. Tech. Papers IEEE Int. Solid-State Circuit Conf. (ISSCC)*, Feb. 1995, pp. 88–89.

[14] T. Gemmeke, M. Gansen, and T. G. Noll, "Implementation of scalable power and area efficient high-throughput Viterbi decoder," *IEEE J. Solid-State Circuits*, vol. 37, no. 7, pp. 941–948, Jul. 2002.

[15] S. Sridharan and L. R. Carley, "A 110 Mhz 350 mW 0.6-μm CMOS 16-state generalized-target Viterbi detector for disk drive read channels," *IEEE J. Solid-State Circuits*, vol. 35, no. 4, pp. 362–370, Mar. 2000.

[16] G. Fettweis, R. Karabed, P. H. Siegel, and H. K. Thapar, "Reduced-complexity Viterbi detector architectures for partial response signalling," in *Proc. GLOBECOM*, vol. 1, Singapore, Nov. 1995, pp. 559–563.

[17] K. Page and P. M. Chau, "Improved architectures for the add-compare-select operation in long constraint length Viterbi decoding," *IEEE J. Solid-State Circuits*, vol. 33, no. 1, pp. 151–155, Jan. 1998.

[18] C. Tsui, R. S.-K. Cheng, and C. Ling, "Low power ACS unit design for the Viterbi decoder," in *Prpc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, vol. 1, May 1999, pp. 137–140.

[19] I. Lee and J. L. Sonntag, "A new architecture for the fast Viterbi algorithm," *IEEE Trans. Commun.*, vol. 51, no. 10, pp. 1624–1628, Oct. 2003.

[20] D. Garrett and M. Stan, "A low power architecture of the soft-output Viterbi algorithm," in *Proc. ACM ISLPED98*, Aug. 1998, pp. 262–267.

Chien-Ching Lin received the B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 2001. He is currently working toward the Ph.D. degree at National Chiao Tung University.

His research interests include coding theory, very large-scale integration (VLSI) architectures, and integrated circuit design for communications and signal processing.

Yen-Hsu Shih received the B.S. degree and M.S. degree in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 2002 and 2004, respectively.

He is now with Realtek Semiconductor Corp., working as an Engineer. His research interests are in the area of algorithm and circuit design for channel coding.

Hsie-Chia Chang received the B.S.and M.S., and the Ph.D. degrees in electronics engineeringfrom National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1995 , 1997, and 2002, respectively.

From 2002 to 2003, he was with OSP/DE1 in MediaTek Corp., working in the area of decoding architectures for Combo single chip. In February 2003, he joined the faculty of the Electronics Engineering Department, National Chiao Tung University. His research interests include algorithms and circuit architectures in signal processing, especially for error control codes and encryption/decryption systems. Recently, he also commits himself to intra-chip communication and multivalued logic design.

Chen-Yi Lee (S'89–M'90) received the B.S. degree from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1982, and the M.S. and Ph.D. degrees from Katholieke University Leuven (KUL), Belgium, in 1986 and 1990, respectively, all in electrical engineering.

From 1986 to 1990, he was with IMEC/VSDM, working in the area of architecture synthesis for DSP. In February 1991, he joined the faculty of the Electronics Engineering Department, National Chiao Tung University, where he is currently a Professor and Department Chair. His research interests mainly include very large-scale integration (VLSI) algorithms and architectures for high-throughput digital signal processing applications. He is also active in various aspects of high-speed networking, system-on-chip design technology, very low power designs, and multimedia signal processing.

Dr. Lee served as the Director of Chip Implementation Center (CIC), an organization for IC design promotion in Taiwan. He was the former IEEE CAS Taipei Chapter Chair. He is now the microelectronics program coordinator of Engineering Division under National Science Council of Taiwan.