A Memory-Efficient Architecture for Low Latency Viterbi Decoders

Yun-Ching Tang[1,2], Do-Chen Hu[1], Weiyi Wei[1], Wen-Chung Lin[1], Hongchin Lin[1]

[1]*Department of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan*
[2]*Department of Electronics Engineering, Hsiuping Institute of Technology, Taichung, Taiwan*
hclin@dragon.nchu.edu.tw

Abstract—**A memory-efficient Viterbi decoder (VD) named modified state exchange (MSE) is proposed using pre-trace back technique to obtain the decoded data by blocks. Since the architecture of MSE can record the "survival state number," which can also be the resulted decoded data, no decision bit is required during trace back and decoding. Therefore, the power and chip area of the survivor memory unit in the MSE method are smaller than those of the existing trace back approaches. The VD using MSE approach for (2, 1, 6) convolutional code was designed using TSMC 0.18μm 1P6M CMOS technology. The core area is 0.69mm^2 with power consumption of 58mW at 100MHz.**

I. INTRODUCTION

In wireless communications, transmission rate and data correction are always the top priorities to consider. However, there are various styles of disturbance and noises that will make the received signals difficult to be recovered to the correct transmitted data, so many algorithms have been proposed to alleviate these problems, for example, error correction codes. In the IEEE 802.11a/g/n, the convolutional encoder is used in the transmitter, and the Viterbi decoder (VD) [1] is employed in the receiver to enhance transmission quality.

Fig. 1 shows the four functional blocks of VD, including branch metric unit (BMU), add-compare-select unit (ACSU), feed-back unit (FBU) and survivor memory unit (SMU). The BMU calculates branch metric of each branch according to maximum likelihood of the received data. The ACSU makes the sum of branch and path metrics, then compares and selects the survivor path metric and the decision bit. The FBU stores the survivor path metric for ACSU to be used in the next cycle. The SMU produces the decoded data based on the decision bit and the survivor path metric.

The SMU marked by boldfaced letters in Fig. 1 significantly influences latency, power and chip area in a VD. It is usually categorized into register exchange (RE) [2], [3] and trace back (TB) [4] approaches. The hardware complexity of RE compared to that of TB may be lower, but the power consumption is usually much higher. On the contrary, power consumption of TB is lower. However, larger memory and register requirement as well as higher latency are its drawbacks. An improved TB with pointer registers named pre-trace back was proposed to reduce the memory and latency [5], but it requires additional pointer registers and still needs decision bits to decode data. In general, the RE technique is only good for trellises with small state numbers, whereas the TB approach is more appropriate for the trellis with large state numbers.

The proposed architecture, which was first reported in the dissertation [6], improves the SMU by tracing the state number which can also be the decoded data. Thus, no decision bits are required during trace back. That makes "block" trace back possible and dramatically reduces the clock cycles during tracing back. In the meanwhile the latency becomes shorter, and the usage of memory/registers is less. The similar concept named state exchange (SE) was also proposed later [7] without minimum path metric unit (MPMU). To differentiate the architectures, the proposed one is named modified state exchange (MSE).

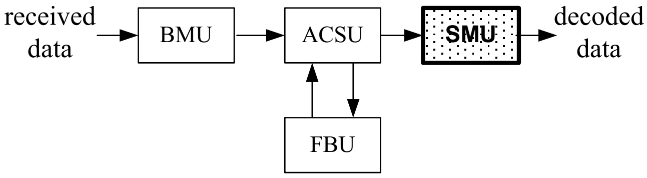


Figure 1 Functional blocks of VD

II. CONVENTIONAL DESIGN OF SMU

To facilitate the presentation of SMU architectures, the convolutional code of (n, k, m) = (2, 1, 6) in the IEEE 802.11a/g/n is employed in this paper. That means there are 2 outputs, 1 input and 6 shift registers in the encoder with the generator polynomials of G_1 = 1011011 and G_2 = 1111001. According to literatures [8], the correct data bits can be extracted if the decoded data sequence is accumulated to the decoding length T, which is approximately $4m$ to $6m$. Our simulation with adding additive white Gaussian Noise (AWGN) indicates $T = 6m$ with 32-level soft decision inputs for 16 symbols Quadrature Amplitude Modulation (16QAM) is very close to the theoretical limits.

The decoding length T also determines the amount of memory/registers in the SMU. Since the total number of states is $N = 2^m = 64$, the minimum memory would be T×N bits. The following briefly describes the operations of conventional TB approaches.

The conventional TB method stores the decision bits in the SMU directly. The survivor path of trellis is determined by these decision bits which correlate to the states. Fig. 2 illustrates the operation of SMU in the 3-pointer even TB VD [4]. The 3 pointers, 2 TB's (trace back) and 1 DC (decoding), work at the same period in the T/2 clock cycles.

978-1-4244-2782-6/09/$25.00 ©2009 IEEE

Because there are N = 64 states with T = 36, each bank has the memory size of 18 bits × 64 (T/2×N). When each writing operation (WR) starts, the previous bank releases a tracing pointer for TB. Since it traces back one bit per clock cycle and keeps WR continuously, the latency is much longer than the RE approach. After the TB path length reaches the decoding length T, one more bank is used to trace the states and generate the decoded data labeled as DC.

Fig. 3 shows the SMU architecture in the 3-pointer even TB VD. It requires six 18×64 memory banks and three TB units and obtains the decoded data bit by bit.

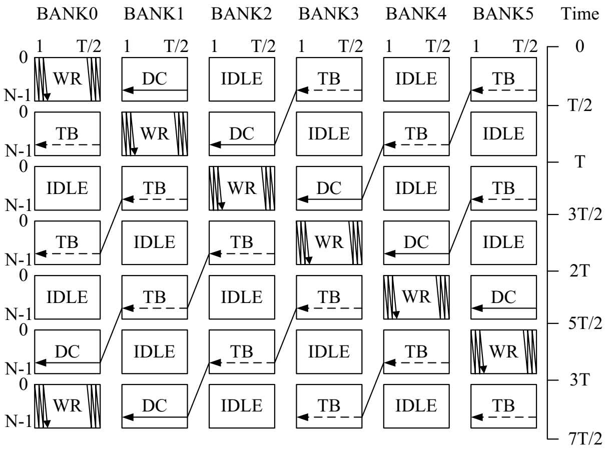


Figure 2 SMU operation of 3-pointer even TB VD

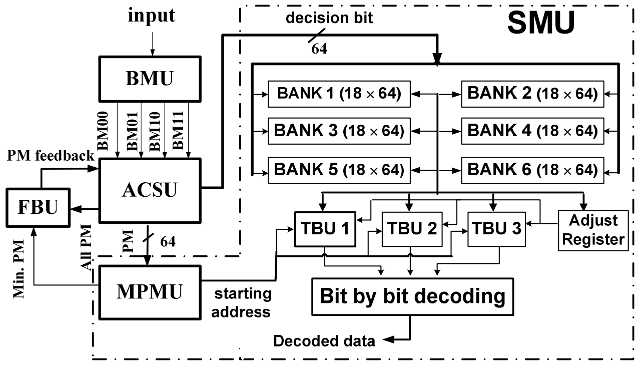


Figure 3 SMU architecture in 3-pointer even TB VD

The pre-trace back approach [5] employs pointer registers to store the states, so the TB process is completed very fast and DC of Bank 0 is performed at time 3T/2 to 2T in Fig. 2. The memory banks are reduced to 4 and the latency is 2T. However, it requires additional 2 sets of pointer registers and additional power consumption. Note that if the length of memory bank is increased to 2T as the example in Ref. [5], the memory banks may be reduced to 3 and only 1 set of pointer register is required. However, memory requirement is larger and latency is longer.

III. THE PROPOSED SMU - STATE EXCHANGE

The proposed SMU like pre-trace back stores the states in the registers as the pointers. The difference is the binary state assignment could be mapped to the decoded data without decision bits and extra circuits.

To explain the operation, if the 64 states are denoted as s0, s1, … , s63, and the trellis starts from s0 with the sequential data (1 0 0 1 1 1, 1 0 1 1 0 0, 1 0 0 1 1 0 …), the survivor path will be s0 => s32 => s16 => s8 => s36 => s50 => **s57** => s60 => s30 => s47 => s55 => s27 => **s13** => s38 => s19 => s9 => s36 => s50 => **s25**. Fig. 4(a) shows state exchange process of the first bank. The state number 0 is transferred to address 57 corresponding to s57 after 6 clocks. The state number 57 will be stored at address 60 (s60) of the second bank in the 7th clock as shown in Fig. 4(b). After another five clocks, "**57**" is transferred to address 13. Fig. 4(c) shows the similar operation after 6 more clocks in the third bank. Thus, "**13**" is stored in address **25**. Note that we only take one survivor path as an example for illustration.

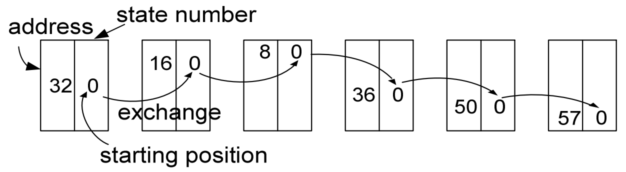


Figure 4(a) State exchange of the first bank

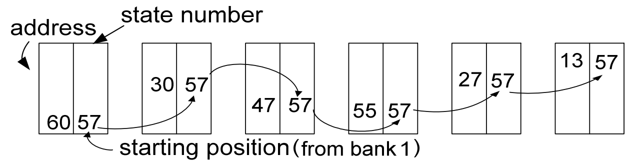


Figure 4(b) State exchange of the second bank

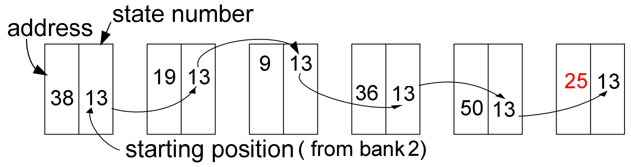


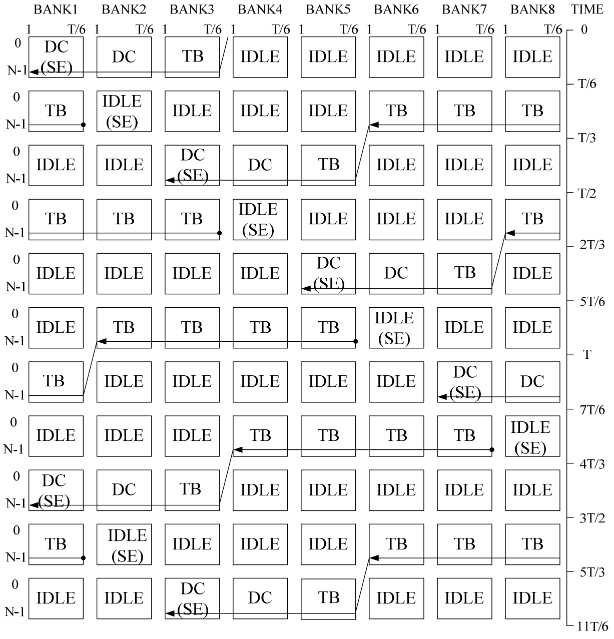
Figure 4(c) State exchange of the third bank

If the tracing back and decoding process are started now, the smallest path metric is determined first, and assumed to be state **25**, the state number stored in address 25 of bank 3 is identified as **13**. Next, the state number in address 13 of bank 2 is **57**. The tracing back process is similar to the pre-trace back, the pointers make trace back fast. Due to the specific binary state assignment, it can be observed that the boldfaced state numbers 57, 13, and 25 are expressed in binary formats, the decoded data can be obtained by reversing the binary order:

57 => (1 1 1 0 0 1) bit reversed => (1 0 0 1 1 1)
13 => (0 0 1 1 0 1) bit reversed => (1 0 1 1 0 0)
25 => (0 1 1 0 0 1) bit reversed => (1 0 0 1 1 0)

In order to maintain the decoding performance like the conventional Viterbi algorithm, the MSE method also needs to accumulate the survivor path longer than the decoding length T before trace back. Since m = 6, the size of each memory bank is m×2^m = 6×64. Due to the decoding length T = 36, 8 memory banks were employed to store the state numbers. A set of 6 × 64 registers was used for state exchange. It is similar to the role of pointer registers in pre-trace back approach [5]. The operations of these 8 memory banks along the time are illustrated in Fig. 5.

The first trace back begins after the survivor path is accumulated for 7 banks. In Fig. 5, five TB's represent six pointer trace back activities, which means the decoding length is $T/6 \times 6 = T$. Due to $m = 6$, six clocks are considered as a group. The first two clocks are used to find the minimum path metric of the 64 states. The other four do pointer TB like to find the state number in the 4 memory banks. In the mean time, state exchange is performed using registers. At the end of the six clocks, the states stored in the registers are transferred to Bank 8. In other words, Bank 8 is idle during these six clocks. The TB operations are continued for another three clock cycles and the extracted state numbers are 12 decoded bits, which are labeled as DC in Fig. 5. These operations are repeated for every 12 clocks until all the data sequences are completely decoded.



BANK1	BANK2	BANK3	BANK4	BANK5	BANK6	BANK7	BANK8
DC (SE)	DC	TB	IDLE	IDLE	IDLE	IDLE	IDLE
TB	IDLE (SE)	IDLE	IDLE	IDLE	TB	TB	TB
IDLE	IDLE	DC (SE)	DC	TB	IDLE	IDLE	IDLE
TB	TB	TB	IDLE (SE)	IDLE	IDLE	IDLE	TB
IDLE	IDLE	IDLE	IDLE	DC (SE)	DC	TB	IDLE
IDLE	TB	TB	TB	TB	IDLE (SE)	IDLE	IDLE
TB	IDLE	IDLE	IDLE	IDLE	IDLE	DC (SE)	DC
IDLE	IDLE	IDLE	TB	TB	TB	TB	IDLE (SE)
DC (SE)	DC	TB	IDLE	IDLE	IDLE	IDLE	IDLE
TB	IDLE (SE)	IDLE	IDLE	IDLE	TB	TB	TB
IDLE	IDLE	DC (SE)	DC	TB	IDLE	IDLE	IDLE

Figure 5 SMU Operation of proposed MSE VD

Fig. 6 shows the SMU architecture in the proposed MSE VD. The register is used to store state numbers for state exchange. The eight memory banks keep the state numbers which are acted as pointers. Before trace back, the minimum path metric is determined first through MPMU. Then, trace back can be done easily by using pointers. The contents of pointers are also the decoded data during "block" decoding. Thus, no decision bits are required. In contrast to the complicated SMU architecture in Fig. 3, the proposed SMU only needs 6×64 registers and $6 \times 64 \times 8$ memory as well as a simple block decoding process. It is worth noting that the SE method [7] did trace back without MPMU, but that would require longer decoding length before trace back. Another advantage of proposed MSE is using SRAM instead of registers to store states for those memory blocks not performing state exchange. That further reduces the power and area.

In summary, the proposed "modified state exchange" (MSE) is to make the state number equal to the decoded data bits. With the state number stored in the memory, the fast trace back process like pre-trace back can be achieved by directly extracting the state number which also represents the decoded data. Therefore, no extra circuits are required for decoding, unlike the conventional TB or pre-trace back method requiring decision bits for decoding data.

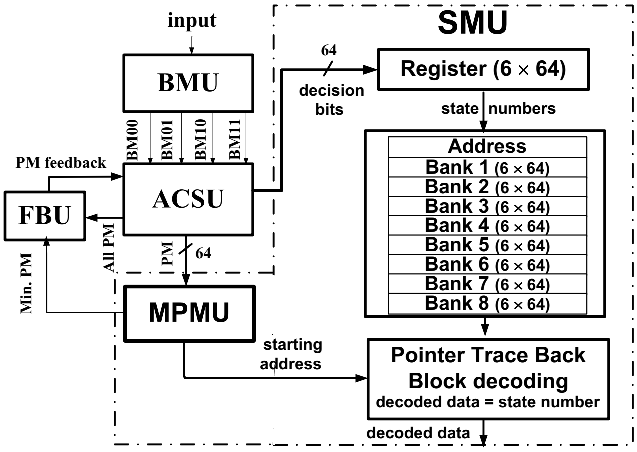


Figure 6 SMU architecture in the proposed MSE VD

IV. PERFORMANCE EVALUATION

To verify the error correcting capability of the proposed MSE algorithm is comparable to the theoretical conventional Viterbi algorithm (VA), bit error rate (BER) simulations with AWGN were performed. Fig. 7 shows all the curves are very close for BPSK, including floating point, fixed point with 32 levels simulations, and FPGA emulation.

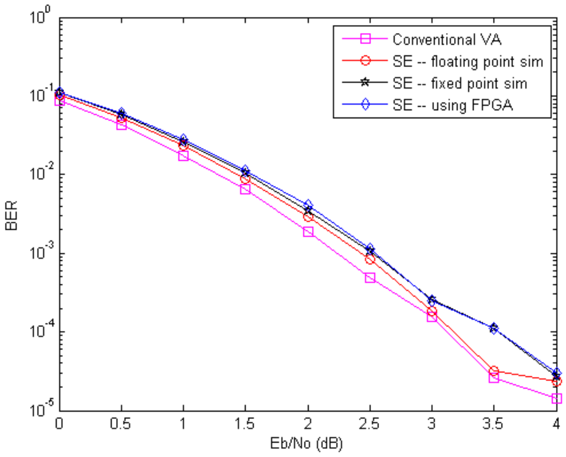


Figure 7 Comparison of BER to the conventional VA

The memory capacity and the latency of various SMU's for (2, 1, 6) VD's with the same decoding length of $T = 36$ are compared in TABLE I [5], [9]. The area of memory unit using SRAM-type circuits is much smaller than that of exchange unit using registers. Note that the 3-pointer even and odd types as well as pre-trace back are TB architectures. These trace back methods need additional trace back unit (TBU) or extra pointer registers, and decision bits for decoding, so the power, area and latency are higher or larger than those of the proposed approach. Some different MRE's [9] were tried to reduce latency of the TB approach. However, the larger exchange unit consumes more power.

We designed two versions of SMU architectures for 100MHz operation using standard cells in 0.18μm CMOS technology, including 3-pointer even TB, and the proposed MSE. Table II compares the area of memory including registers, TBU and SMU. For the area of SMU, MSE is about 23% smaller than TB, which is a significant improvement. Power consumptions of SMU's after place and route are compared in Table III. The power of SMU for MSE is 25% less than that for TB.

TABLE I Comparison of memory capacity and latency

	3-pointer even	3-pointer odd	Register exchange	MRE (3 banks)	MRE (5 banks)	Pre-TB (4 banks)	MSE (8 banks)
memory unit	64×18×6 (6912)	64×18×5 (5760)		64×18×3 (3456)	64×9×5 (2880)	64×18×4 (4608)	64×6×8 (3072)
exchange unit			64×36 (2304)	64×6×2 (768)	64×6×4 (1536)	64×6×2 (768)	64×6 (384)
Latency	3T	3T	T	2T	3T/2	2T	3T/2

TABLE II Comparison of SMU area

Core area (μm^2)	Trace back	MSE	reduction
Memory	214k	173k	-19.16%
TBU	24k	1k	-58.33%
SMU total	238k	183k	-23.11%

TABLE III Comparison of SMU power

Power(mW) @100MHz	Trace back	MSE	reduction
Memory	13.6	11.38	-16.32%
TBU	1.98	0.7	-64.65%
SMU total	15.58	12.08	-22.46%

Several VD's for IEEE 802.11a/g using trace back technique have been implemented in recent years [10]-[12]. The comparison to our proposed MSE method is given in TABLE IV. One of them adopts hard decision. The other use soft decision with 3 to 5 bits. However, our simulation indicates 5 bits to achieve the closest performance to theoretical Viterbi algorithm. The normalized area is normalized with respect to CMOS 0.18μm technology. The overall performance, like area, power and latency, the proposed MSE architecture is superior to the others.

TABLE IV COMPARISON OF VITERBI DECODER CHIPS

Design	technology	states	type	Power (mW)	Core area (mm^2)	Normalized area
[10]	CMOS 0.25 μm	64	soft 3-bit	90 (100MHz)	2.16	1.12
[11]	CMOS 0.18 μm	64	hard	39 (100MHz)	2	2
[12]	CMOS 0.18 μm	64	soft 4-bit	68 (72MHz)	3.06	3.06
Proposed MSE	CMOS 0.18 μm	64	Soft 5-bit	58 (100MHz)	0.69	0.69

V. CONCLUSIONS

The proposed "modified state exchange" algorithm simplifies the trace back processes with "block decoding" capability to achieve low power, low latency and low memory requirement. The latency of MSE is reduced from 3T of the conventional TB to 3T/2. The power and chip area are significantly smaller than those of TB, simultaneously. The VD using MSE architecture for the (2, 1, 6) convolutional code was designed using TSMC 0.18μm 1P6M CMOS technology. The core area is 0.69mm^2 with power consumption of 58mW at 100MHz.

ACKNOWLEDGMENT

The authors would like to acknowledge Chip Implementation Center (CIC) of National Applied Research Laboratories (NARL) of Taiwan for the support in chip design. This work was supported by National Science Council of Taiwan (NSC 96-2221-E-005-091 and NSC 96-2220-E-005-005).

REFERENCES

[1] I. Kang and A. N. Willson Jr, "Low-power Viterbi decoder for CDMA mobile terminals," *IEEE J. Solid-State Circuits*, vol.33, pp. 473-482, 1998.

[2] D.A.F. Ei-Dib, M.I. Elmasry, "Low-power Register-exchange Viterbi Decoder for High-speed Wireless Communications," *IEEE Int. Symp. Circuits and Systems* (*ICSAS*), vol. 5, pp. 737-740, 2002.

[3] D.A.F. Ei-Dib, M.I. Elmasry, "Modified Register-exchange Viterbi Decoder for Low-power Wireless Communications," *IEEE Trans. Circuits and Systems I*, vol. 51, pp. 371-378, 2004.

[4] T. K. Truong, M.T. Shih, I. S. Reed, E. H. Satorius, "A VLSI design for a trace-back Viterbi decoder," *IEEE Trans. Commun.*, vol. 40, pp. 616-624, 1992.

[5] Y. Gang, A. T. Erdogan, and T. Arslan, "An Efficient Pre-Traceback Architecture for the Viterbi Decoder Targeting Wireless Communication Applications", *IEEE Trans. on Circuits and Systems I*, vol. 53, pp. 1918-1927, 2006.

[6] Do-Chen Hu, "Design of Low Latency Trace Back Viterbi Decoder Using State Exchange," M. S. Dissertation, National Chung-Hsing University, July 2007.

[7] C.-Y. Chu, Y.-C. Huang and A.Y. Wu, "Power Efficient Low Latency Survior Memory Architecture for Viterbi Decoder," IEEE *International Symposium on VLSI Design, Automation, and Test*, pp. 228-231, Hsinchu, Taiwan, April 2008.

[8] G. C. Clark and J. B. Cain, *Error-Corection Coding for Digital Communications*, New York: Plenum, pp. 227-264, 1981.

[9] J. S. Han, T. J. Kim, C. Lee, "High performance Viterbi decoder using modified register exchange methods," *IEEE Int. Symp. Circuits and Systems* (*ISCAS*), vol. 3, pp. 553-556, May 2004.

[10] Y. X. You, J. X. Wang, F. C. Lai, Y. Z. Ye, "VLSI Design and Implementation of High-Speed Viterbi Decoder," *IEEE Commun. Circuits and systems and west sino Expositions*, vol. 1, pp 64-48, July 2002.

[11] M. Kawokgy, C.A.T. Salama, "Low-power asynchronous Viterbi decoder for wireless applications," *IEEE Int. Symp. Low power Electronics and Design* (*ISLPED*), pp. 286-289, Aug. 2004.

[12] C. C. Lin, Y. H. Shih, H. C. Chang, C.Y. Lee, "Design of a Power –Reduction Viterbi Decoder of WLAN Application," *IEEE Trans. Circuit and Systems I*, vol. 52, pp. 1148-1156, 2005.