Triple-Mode MAP/VA Timing Analysis for Unified Convolutional/Turbo Decoder Design

Fan-Min Li, Pei-Ling Shen, An-Yeu (Andy) Wu

Graduate Institute of Electronics Engineering, and
Department of Electrical Engineering, National Taiwan University
Taipei, 106, Taiwan, R.O.C

ABSTRACT

To satisfy the advanced FEC standard that performs both convolutional code and turbo code, a unified convolutional/turbo decoder is needed. In this paper, both timing of Viterbi and MAP algorithm are analyzed. We introduce three techniques, including interleaving, pointer, and parallel schemes, which can be used in timing charts to reduce memory or increase throughput. In the recent works, several timing charts of VA or MAP are presented, but there is no combined timing analysis of both algorithms. In this paper, we propose two types of triple-mode MAP/VA timing charts by complementing idle time of each other.

1. INTRODUCTION

The Viterbi algorithm (VA) is a maximum-likelihood decoding method, which minimizes the probability of word error for convolutional codes [1]. In digital communication systems, the VA is widely used to detect sequential error-control codes and to detect symbols. In the recent years, a new class of convolutional codes called turbo codes was introduced by Berrou, Glavieux, and Thitimajashima [2], and it is well known for its extremely superior decoding accuracy. The decoder consists of two component soft-output decoders and operators by iterative fashion. The component soft-output algorithm prescribed in the original turbo code paper [2] is usually known as the *maximum a-posteriori probability* (MAP) algorithm [2].[4].

Due to the outstanding decoding ability, turbo coding gets rapid development within just a few years and become standardized [5], like CDMA2000 and WCDMA. Generally, the voice and data streams in these systems use different types of coding schemes, such as convolutional code and turbo code. Traditionally, the corresponding Viterbi and turbo decoder are built separately. But here, in order to save chip area and make the design simple and efficient, we propose a unified solution by integrating the two decoders. In the recent works, several timing charts of VA or MAP are presented [6] [7] [8], but there is no combined timing analysis of both algorithms. Therefore, we propose two types of triple-mode MAP/VA timing

charts by complementing idle time of each other. We define triple mode below: (a) MAP/VA concurrent mode. (b) VA single mode. (c) MAP single mode. In addition, we introduce three techniques, including interleaving, pointer and parallel schemes, which can be used in timing charts to reduce memory or increase throughput. Then, by analyzing the concepts and the architectures of the Viterbi and the turbo decoders, some circuits sharing techniques can be applied to merge the main functions into one unified decoder.

2. TIMING ANALYSIS OF VITERBI DECODING

2.1. Viterbi decoding

In the VA operation, it needs to store the values of decision information and fetch while tracing back. Due to reduce the storage space, sliding-window decoding is adopted in Fig. 1. At the first time unit (Area I), forward-path-metric recursion begins with equal probability at the head of the first sub block and stores the decision information. When the operation at the second time unit (Area II) reaches the tail of the second sub block, we can choose any state to trace back at the third time unit (Area III) to find the converge state at the head of second sub block. Then, in fourth time unit (Area IV), the valid tracing-back starts at the head of the first sub block and decodes correct information bits. The sliding-window method just reuse three sub block memory and make almost the same performance as global recursion.

Fig. 1 Timing of Viterbi decoding, Type B.

We define two parameters V and S : the valid (V) decoding region and shift (S) from present operating to next operating. The unit L means the time of decoding one sub block or the sliding window length. In the Fig. 1, the valid region equals L ($V=L$), and the shift equals L ($S=L$).

The shade area means what time and location the decision information must be kept. Observing the cross line, we can find that at each time constant the decoder needs three sub-block Memory ($M=3L$), one forward *Path Metric* (PM) recursive unit ($Npm=1$) and two *Trace-Back* (TB) units ($Ntb=2$). Finally, V divided by S equals throughput ($T=V/S=1$).

Several researches about timing of VA also been studied. We arrange and plot graphic represent Fig. 2. The memory can be reduced in both Type A and Type C.

Fig. 2 Timing of Viterbi decoding: (a) Type A. (b) Type C.

2.2. Timing analysis of Viterbi algorithm

Now, we introduce three techniques to reduce memory or increase throughput.

A. Interleaving Technique

For reducing the memory of Type B in Fig. 1, we use more than one PM recursive unit and the original big triangular area in Type B, which means the decision bits stored in the memory, can be interleaving into two small ($If=4$) triangular areas in Type D. The parameter forward interleaving (If) indicates the degree of interleaving, and the memory can be reduced approximate $1/If$ but the PM recursive unit can be increased approximate If .

Fig. 3 Timing of Viterbi decoding with forward interleaving technique, Type D ($If=4$).

B. Pointer Technique

In the above discussion, we know the interleaving technique can reduce memory but increase PM recursive units. Observing the Type D in Fig. 3, we find that by increasing the degree of interleaving, more and more PM

recursive units are used, and just the triangular areas are used. Therefore, if we can just run the triangular sections, the hardware or power can be reduced. But the problem is where the initial probability of PM recursive units is from. For the Type E in Fig. 4, one PM recursive unit pre-computes the initial probability, stores in registers and pass for computing of other PM recursive units in triangular sections. The parameter forward pointer (Pf) indicates the degree of pointer, and the memory can be reduced approximate $1/Pf$ but the hardware is approximate the same.

C. Parallel Technique

The throughput (T) is decided by valid region (V) and Shift (S),

$$T=V/S. \quad (1)$$

Therefore, the throughput of Type B in Fig. 1 equals one. Thus, for increasing throughput, we can increase valid region or decrease shift (Type F) in Fig. 5.

Fig. 4 Timing of Viterbi decoding with forward pointer technique, Type E ($Pf=4$).

Fig. 5 Timing of Viterbi decoding with parallel technique, Type F ($V=2L, S=L$).

D. Combined Approach

For high throughput and low memory, we can combine interleaving, pointer and parallel techniques in Type G in Fig. 6.

Fig. 6 Timing of Viterbi decoding with parallel and forward pointer techniques, Type G ($V=2L, S=L, Pf=12$).

We summarize the information of all VA timing charts in Table 1.

Table 1 Summary of VA timing charts.

Type	Parameters						Techniques			
	V	S	Npm	Ntb	M	T	F.I.	F.P.	P.	
A	L	2L	1	1	2L	0.5	-	-	-	
B	L	L	1	2	3L	1	-	-	-	
C	0.5L	0.5L	1	3	2.5L	1	-	-	-	
D	L	L	2	2	2L	1	If=2	-	-	
	L	L	4	2	L	1	If=4	-	-	
	L	L	8	2	0.5L	1	If=8	-	-	
E	L	L	3	2	L+5R	1	-	Pf=4	-	
	L	L	3	2	0.5L+11R	1	-	Pf=8	-	
F	2L	L	3	3	5L+1R	2	-	-	V	
	3L	L	4	4	9L	3	-	-	V	
	L	0.5L	4	4	5L	2	-	-	V	
	1.5L	0.5L	5	5	6.5L+1R	3	-	-	V	
G	2L	L	6	3	0.75L+3R	2	-	Pf=12	V	
	3L	L	8	4	L+4R	3	-	Pf=16	V	
	1L	0.5L	7	4	L+3R	2	-	Pf=8	V	
	1.5L	0.5L	9	5	L+5R	3	-	Pf=10	V	

3. TIMING ANALYSIS OF MAP DECODING

3.1. MAP decoding

In the Log-MAP operation, it needs to store the values of forward-alpha recursion and fetch while backward-beta recursion. Due to the high cost of large memory and latency, sliding-window decoding is adopted. A long frame is divided into several sub blocks which length equals about five times constrain length. For easily speaking, we define one time unit is the time of forward or backward accumulation in one sub block. In the first time unit (Area I) of Fig. 7(a), backward-beta recursion begins with equal probability at the tail of the second sub block. When the operation reaches the head of the second sub block, the accumulated beta values are converged and used to be the initial beta values for valid backward-beta recursion in the second time unit (Area II). The valid beta values of the second time unit must be stored in the memory. Then, in third time unit (Area III), the forward-alpha recursion starts the head of the first sub block and combines the valid beta values from the memory to produce LLR values. The shade area means what time and location the valid beta values must be kept. We can find that at each time constant the decoder needs one sub block memory, one forward recursive unit (RUA) and two backward recursive units (RUB). With the same hardware as Fig. 7(a), the stored values in memory can be halves alpha and beta values in Fig. 7(b), or all alpha values in Fig. 7(c).

3.2. Timing analysis of MAP algorithm

In the recent works, several researches about timing of MAP have been studied [6][10]. We arrange and plot graphic represents as follows.

A. Interleaving Technique

The same as VA, for reducing the memory of Type B ($D=2L$) in Fig. 7(a), we use more than one beta recursive unit (RUB) and the original big triangular area in Type B ($D=2L$), which means the backward recursive probability

stored in the memory, can be interleaving into two small ($Ib=2$) triangular areas in Type C in Fig. 9. The parameter backward interleaving (Ib) indicates the degree of interleaving, and the memory can be reduced approximate $1/Ib$ but the beta recursive unit can be increased approximate Ib .

Fig. 7 Timing of MAP decoding: (a) Graphic represent, Type B ($D=2L$). (b) Type B ($D=L$). (c) Type B ($D=0$).

Fig. 8. Timing of MAP decoding, Type A ($D=2L$).

Fig. 9. Timing of MAP decoding with backward interleaving technique: (a) Type C ($Ib=2, D=L$). (b) Type C ($Ib=4, D=L/2$).

Fig. 10. Timing of MAP decoding with forward interleaving technique: (a) Type D ($If=2, D=L$). (b) Type D ($If=4, D=L/2$).

Besides, we also can use forward interleaving technique to reduce memory in Type D in Fig. 10. The parameter forward interleaving (I_f) indicates the degree of interleaving, and the memory can be reduced approximate $1/I_f$ but the forward recursive unit can be increased approximate I_f . Moreover, backward and forward interleaving can be combined to use in Type E in Fig. 11.

Fig. 11. Timing of MAP decoding with forward and backward interleaving tech. ($I_f=2$ and $I_b=2$): (a) Type E1. (b) Type E2.

B. Pointer Technique

To improve interleaving technique, the pointer method is also used in MAP timing charts. In Type F in Fig. 12(a), the parameter backward pointer P_b indicates the degree of pointer, and the memory can be reduced approximate $1/P_b$ but the hardware is approximates the same. The forward pointer also can be used in Type G in Fig. 12(b). The parameter forward pointer P_f indicates the degree of pointer, and the memory can be reduced approximate $1/P_f$ but the hardware approximates the same. Moreover, backward and forward pointer can be combined to use in Type H in Fig. 13.

Fig. 12. Timing of MAP decoding with pointer technique: (a) Type F ($P_b=4$, $D=L/2$). (b) Type G ($P_f=4$, $D=L/2$).

Fig. 13. Timing of MAP decoding with forward and backward pointer techniques ($P_f=2$ and $P_b=2$) (a) Type H1 (b) Type H2.

C. Parallel Technique

The same as VA, the throughput (T) is also decided by valid region (V) and Shift (S). For increasing throughput, we can increase valid region (Type I in Fig. 14(a)) or decrease shift (Type J in Fig. 14(b)).

D. Combined Approach

For high throughput and low memory, we can combine backward interleaving with parallel techniques in Type L in Fig. 15(a), combine forward interleaving with parallel techniques in Type M in Fig. 15(b), combine backward and forward interleaving with parallel techniques in Type N in Fig. 15(e). Again, we can combine backward pointer with parallel techniques in Type O in Fig. 15(c), combine forward pointer with parallel techniques in Type P in Fig. 15(d), combine backward and forward pointer with parallel techniques in Type Q in Fig. 15(f). Table 2 shows the summary of the MAP timing charts.

Fig. 14. Timing of MAP decoding with parallel technique: (a) Type I ($V=2L$, $S=L$, $D=4L$). (b) Type J ($V=L$, $S=L/2$, $D=2L$).

Fig. 15. Timing of MAP decoding with interleaving, pointer and parallel techniques ($V=2L$ and $S=L$): (a) Type L ($I_b=2$). (b) Type M ($I_f=2$). (c) Type O ($P_b=4$). (d) Type P ($P_f=4$). (e) Type N ($I_f=2$ and $I_b=2$). (f) Type Q ($P_f=2$, $P_b=2$).

Table 2 Summary of MAP timing charts.

Type	Parameters							Techniques				
	V	S	D	Na	Nb	M	T	I.F.	I.B.	P.F.	P.B.	P.
A	L	2L	2L	1	1	L	0.5	-	-	-	-	-
	L	2L	L	1	1	L	0.5	-	-	-	-	-
	L	2L	0	1	1	L	0.5	-	-	-	-	-
B	L	L	2L	1	2	L	1	-	-	-	-	-
	L	L	L	1	2	L	1	-	-	-	-	-
	L	L	0	1	2	L	1	-	-	-	-	-
C	L	L	L	1	3	L/2	1	-	Ib=2	-	-	-
	L	L	L	1	3	L/2	1	-	Ib=2	-	-	-
	L	L	L/2	1	3	L/2	1	-	Ib=2	-	-	-
	L	L	0	1	3	L/2	1	-	Ib=2	-	-	-
	L	L	L/2	1	5	L/4	1	-	Ib=4	-	-	-
	L	L	L/4	1	5	L/4	1	-	Ib=4	-	-	-
D	L	L	0	1	5	L/4	1	-	Ib=4	-	-	-
	L	L	L	2	2	L/2	1	If=2	-	-	-	-
	L	L	L/2	2	2	L/2	1	If=2	-	-	-	-
	L	L	0	2	2	L/2	1	If=2	-	-	-	-
	L	L	L/2	4	2	L/4	1	If=4	-	-	-	-
	L	L	L/4	4	2	L/4	1	If=4	-	-	-	-
E1	L	L	0	4	2	L/4	1	If=4	-	-	-	-
E2	L	L	L/2	2	3	L/4	1	If=2	Ib=2	-	-	-
F	L	L	L/2	1	3	L/4+3R	1	-	-	-	Pb=4	-
	L	L	L/4	1	3	L/4+3R	1	-	-	-	Pb=4	-
	L	L	0	1	3	L/4+3R	1	-	-	-	Pb=4	-
G	L	L	L/2	2	2	L/4+3R	1	-	-	Pf=4	-	-
	L	L	L/4	2	2	L/4+3R	1	-	-	Pf=4	-	-
	L	L	0	2	2	L/4+3R	1	-	-	Pf=4	-	-
H1	L	L	L/2	2	3	L/4	1	-	-	Pf=2	Pb=2	-
H2	L	L	L/2	2	3	L/4	1	-	-	Pf=2	Pb=2	-
I	2L	L	4L	3	3	4L	2	-	-	-	-	V
	2L	L	2L	3	3	2L	2	-	-	-	-	V
	2L	L	0	3	3	4L	2	-	-	-	-	V
J	L	L/2	2L	4	4	2L	2	-	-	-	-	V
	L	L/2	L	4	4	L	2	-	-	-	-	V
	L	L/2	0	4	4	2L	2	-	-	-	-	V
K	L	L/4	2L	8	8	4L	4	-	-	-	-	V
	L	L/4	3L/2	8	8	5L/2	4	-	-	-	-	V
	L	L/4	L	8	8	2L	4	-	-	-	-	V
	L	L/4	L/2	8	8	5L/2	4	-	-	-	-	V
	L	L/4	0	8	8	4L	4	-	-	-	-	V
L	L	L/2	2L	3	4	2L	2	-	Ib=2	-	-	V
	L	L/2	L	3	5	L	2	-	Ib=4	-	-	V
	L	L/2	L/2	3	9	L/2	2	-	Ib=8	-	-	V
M	2L	L	0	3	3	2L+1R	2	If=2	-	-	-	V
	2L	L	0	3	5	L+1R	2	If=4	-	-	-	V
	2L	L	0	3	8	L/2	2	If=8	-	-	-	V
N	2L	L	(L, 0)	3	5	L	2	If=2	Ib=2	-	-	V
	2L	L	(L/2, 0)	5	6	L/2+1R	2	If=4	Ib=4	-	-	V
O	2L	L	L	3	4	L	2	-	-	-	Pb=4	V
	2L	L	L/2	3	6	L/2+6R	2	-	Ib=2	-	Pb=8	V
	2L	L	L/2	3	5	L/2+14R	2	-	-	-	Pb=8	V
P	2L	L	L	4	3	L+2R	2	-	-	Pf=4	-	V
	2L	L	L/2	5	3	L/2+6R	2	If=2	-	Pf=8	-	V
	2L	L	L/2	5	3	L/2+13R	2	-	-	Pf=8	-	V
Q	2L	L	L	3	4	L+1R	2	-	-	Pf=2	Pb=2	V
	2L	L	L/2	4	4	L/2+5R	2	-	-	Pf=4	Pb=4	V

4. TIMING ANALYSIS OF TRIPLE-MODE MAP/VA DECODING

We have detailed discussion of MAP and VA timing charts. Now we want to find a suitable timing chart that can combine both different algorithms. Observing Type A timing chart of VA, we find that the PM recursive unit is always idle in half time and the alpha recursive unit in Type A timing chart of MAP is the same situation. Therefore, we try to combine timing charts of MAP and VA in Fig. 16 that can decode at the same time. The principle of combination is to use the idle time of each

other, thus, the decoder can run in MAP mode, VA mode or even MAP/VA mode.

Fig. 16 Timing of triple mode decoding, Type A of VA and Type A (D=0L) of MAP.

We propose two types of triple-mode VA/MAP timing charts that first mode is VA mode, second mode is MAP mode and third mode is MAP/VA mode. In Type I, the MAP/VA mode in Fig. 16 is combined Type A of VA and Type A (D=0L) of MAP. In the VA mode in Fig. 17(a), we use the interleaving technique ($I_f=2$) by the original sharing hardware of forward recursive unit. In the MAP mode in Fig. 17(b), we also use the pointer technique ($P_f=4$) by the original sharing hardware.

Fig. 17 Timing of triple mode decoding (Type I): (a) VA mode-a ($I_f=2$). (b). MAP mode ($P_f=4$).

Moreover, we try to translate the backward recursive unit of MAP part into forward recursive unit by exchanging the input and output ports of trellis wires. Thus, when running in VA mode, we can use pointer technique in Fig. 18 by the backward recursive unit of MAP part, which is idle.

Fig. 18 VA mode of Type I: (a) VA mode-b ($I_f=2$ and $P_f=2$). (b) VA mode-c ($I_f=2$ and $P_f=4$).

In Type II, we use pointer technique to reduce memory by additional forward recursive unit in Fig. 19. Specially, when running in MAP mode, it can operate like original MAP part in Fig. 19(a) or enhance throughput in Fig. 19(c).

Fig. 19 Timing of triple mode decoding (Type II): (a) MAP/VA mode. (b) VA mode ($I_f=2$ and $P_f=2$). (c) MAP mode

Table 3 Summary of MAP/VA timing charts

Type	Mode	Parameters								Techniques			
		V	S	D	Na	Nb	Npm	Ntb	M	T	I.F	P.F	
I	1	VA ^a b c	L	2L	0L	-	-	1	1	1L+1R	0.5	If=2	-
			L	2L	0L	-	-	2	1	L/2+2R	0.5	If=2	Pf=2
			L	2L	0L	-	-	2	1	L/4+4R	0.5	If=2	Pf=4
	2	MAP	L	2L	0L	1	1	-	-	L/4+3R	0.5	-	Pf=4
	3	VA	L	2L	0L	-	-	1 share	1	2L+1R	0.5	-	-
MAP		L	2L	0L	1 share	1	-	-	1L	0.5	-	-	
II	1	VA	L	2L	0L	-	-	2	1	L/2+2R	0.5	If=2	Pf=2
	2	MAP	L	L	0L	1	2	-	-	1L	1	-	-
	3	VA	L	2l	0L	-	-	2 share	1	1L+1R	0.5	If=2	-
		MAP	L	2l	0L	2 share	1	-	-	L/4+3R	0.5	-	Pf=4

5. CONCLUSIONS

In this paper, we present three techniques, Interleaving, Pointer, and Parallel schemes, to reduce memory or increase throughput weather in Viterbi algorithm decoding or MAP decoding. Then, we present several timing charts of two algorithms and do some comparisons. Finally, we propose two types of triple-mode MAP/VA timing charts, thus it can be used to design a unified convolutional/turbo decoder for modern communication, such as 3GPP handset.

6. REFERENCES

- [1] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260-269, Apr. 1967.
- [2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. IEEE Global Telecommunications Conf. 1993*, Houston, TX, Nov. 29-Dec. 2, 1993, pp. 809-813.
- [3] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284-287, March 1974.
- [4] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. Int. Conf. Comm.*, June 1995, pp. 1009-1013.
- [5] <http://www.3gpp.org/>
- [6] E. Boutillon, W. J. Gross and P. G. Gulak, "VLSI Architectures for the MAP algorithm," *IEEE Trans. Comm.* Vol. 51, No. 2, pp. 175-185, Feb. 2003.
- [7] C. Schurgers, F. Catthoor and M. Engels, "Memory Optimization of MAP Turbo Decoder Algorithm," *IEEE Trans. VLSI*, Vol. 9, No. 2, pp.305-312, Apr. 2001.
- [8] Z. Wang, Z. Chi and K. K. Parhi, "Area -Efficient High-Speed Decoding Schemes for Turbo Decoders," *IEEE Trans. VLSI*, Vol. 10, No. 6, pp. 902-912, Dec. 2002.
- [9] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. Info. Theory*, Vol. IT-13, pp. 260-269, April 1967.
- [10] Z. Wang, Z. Chi, K. K. Parhi, "Area-Efficient High-Speed Decoding Schemes for Turbo Decoders," *IEEE Trans. VLSI systems*, Dec. 2002.
- [11] K. K. Parhi and T. Nishitani, "Digital Signal Processing for Multimedia Systems." Marcel Dekker, New York.