ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.14, No.5, 1–6

Unified encoder embedded trellis router designs for decoding convolutional and turbo codes

Cheng-Hung Lin[a)] **and Tsung-Ju Hsieh**

Department of Electrical Engineering and Innovation Center for Big Data and Digital Convergence, Yuan Ze University, 135 Yuantung Rd., Jungli, 32003, Taiwan

a) *chlin@saturn.yzu.edu.tw*

Abstract: Communication systems usually adopted different encoding structures of convolutional codes (CCs) and turbo codes (TCs). This Letter proposes a unified encoder embedded trellis router (UEETR) to support various decoding of single-binary/double-binary (SB/DB) CC and TC encoding structures with a small hardware overhead. Furthermore, a low-latency UEETR and a hardware-shared radix-4 UEETR are explored to reduce initial setting time and to support radix-4 SB/DB CC and TC decoding, respectively. In addition, a flexible decoding kernel with a small area cost of proposed UEETR has been implemented by a 90-nm process.

Keywords: convolutional code, turbo code, multiple standards

Classification: Integrated circuits

References

[1] A. J. Viterbi: "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," IEEE Trans. Inf. Theory **13** (1967) 260 (DOI: 10.1109/TIT.1967.1054010).

[2] C. Berrou, *et al.*: "Near Shannon limit error-correcting coding and decoding: Turbo-codes (1)," Proc. IEEE Int. Conf. Commun. (1993) 1064 (DOI: 10.1109/ICC.1993.397441).

[3] C. Berrou and M. Jezequel: "Non-binary convolutional codes for turbo coding," Electron. Lett. **35** (1999) 39 (DOI: 10.1049/el:19990059).

[4] B. M. Hochwald and S. ten Brink: "Achieving near-capacity on a multiple-antenna channel," IEEE Trans. Commun. **51** (2003) 389 (DOI: 10.1109/TCOMM.2003.809789).

[5] F.-M. Li, *et al.*: "Unified convolutional/turbo decoder design using tile-based timing analysis of VA/MAP kernel," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **16** (2008) 1358 (DOI: 10.1109/TVLSI.2008.2000514).

[6] C.-H. Lin, *et al.*: "Area-efficient scalable MAP processor design for high-throughput multistandard convolutional turbo decoding," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **19** (2011) 305 (DOI: 10.1109/TVLSI.2009.2032553).

[7] C.-H. Lin and C.-S. Yu: "Multimode radix-4 SISO kernel design for turbo/LDPC decoding," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **23** (2015) 2256 (DOI: 10.1109/TVLSI.2014.2363777).

[8] C.-H. Lin, *et al.*: "Reconfigurable parallel turbo decoder design for multiple

DOI: 10.1587/elex.14.20170028
Received January 10, 2017
Accepted February 3, 2017
Publicized February 21, 2017
Copyedited March 10, 2017

high-mobility 4G systems," J. Signal Process. Syst. **73** (2013) 109 (DOI: 10.1007/s11265-013-0735-5).
[9] C.-C. Lin, *et al.*: "Design of a power-reduction Viterbi decoder for WLAN applications," IEEE Trans. Circuits Syst. I, Reg. Papers **52** (2005) 1148 (DOI: 10.1109/TCSI.2005.849106).
[10] C. Studer, *et al.*: "Design and implementation of a parallel turbo-decoder ASIC for 3GPP-LTE," IEEE J. Solid-State Circuits **46** (2011) 8 (DOI: 10.1109/JSSC.2010.2075390).

1 Introduction

Two well-known forward error correction (FEC) codes, convolutional code (CC) [1] and turbo code (TC) [2, 3], were widely adopted by communication standards. The prevalent trellis decoding algorithms of the CCs and TCs are Viterbi algorithm (VA) [1] and maximum *a posteriori* algorithm (MAP) [2], respectively. Nowadays, the MAP decoding is widely used to decode the CCs because of rapid growth of iterative detection and decoding [4]. The different communication standards usually adopted different encoding structures (or generator polynomials) of the TCs and CCs. Thus the decoding trellis of MAP and VA requires distinct branch symbols for the different encoding structures. Encoder embedded trellis router (EETR) [5] was proposed to support the radix-2 VA decoding of single-binary (SB) CC composed of a non-recursive systematic convolutional (NSC) encoder with memory element $M = 3$. It also supports the radix-2 MAP decoding of SB TC composed of two $M = 3$ recursive systematic convolutional (RSC) encoders. However, it restrictively supports the radix-2 decoding trellis of VA and MAP decoding.

In this Letter, a unified EETR (UEETR) is proposed to support the alternative SB and double-binary (DB) [3] encoding structures with an arbitrary M. However, a large M prolongs the initial setting time of proposed UEETR. Thus a low-latency UEETR is proposed to reduce the setting time of UEETR. In addition, this Letter demonstrates a hardware-shared radix-4 UEETR (R4UEETR) for the radix-4 SB/DB MAP decoding [6, 7, 8].

2 Proposed UEETR

The EETR has been proposed for $M = 3$ SB RSC encoder. Thus the UEETR for a general DB RSC encoder must be initially derived. Fig. 1 shows a general DB RSC encoder with M memory elements and N parity bits. In Fig. 1, X_A and X_B are the double systematic bits; X_{Pi} is the i_{th} parity bit; g_{ij} is the input signal of encoding structure; S_j is the binary value of the j_{th} memory element; $1 \le i \le N$; $0 \le j \le M-1$; and P is the input value before the first memory element S_0. Based on extending the derivations in [5], we can have

$$\begin{cases} X_A = P \oplus (g_{01} \otimes S_0) \oplus (g_{02} \otimes S_1) \oplus \ldots \oplus (g_{0M} \otimes S_{M-1}), \\ X_{Pi} = X_B \oplus (g_{io} \otimes P) \oplus (g_{i1} \otimes S_0) \oplus \ldots \oplus (g_{iM} \otimes S_{M-1}). \end{cases} \quad (1)$$

Note that "$\oplus$" denotes an XOR operation and "$\otimes$" denotes an AND operation. According to Eq. (1), the UEETR with M (denoted as UEETR-M) is then proposed

DOI: 10.1587/elex.14.20170028
Received January 10, 2017
Accepted February 3, 2017
Publicized February 21, 2017
Copyedited March 10, 2017

to generate the branch symbol (X_A, X_B, X_{Pi}). Fig. 2 shows the proposed UEETR-M that is composed of an $(M+2)$-bit binary counter and $(N+1)$ sets of AND/XOR gates. The UEETR employs the input signals g_{ij} to generate the output signals of branch symbols before the decoding. Compared to the EETR in [5], the counter of UEETR requires an additional memory element X_B to support the DB NSC/RSC encoder. By setting $X_B = 0$, the UEETR supports the SB NSC/RSC encoder and generates the branch symbol (X_A, X_{Pi}).

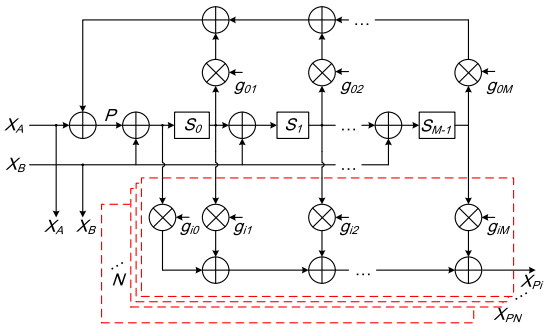

Fig. 1. General DB RSC encoder with M memory elements and N parity bits.

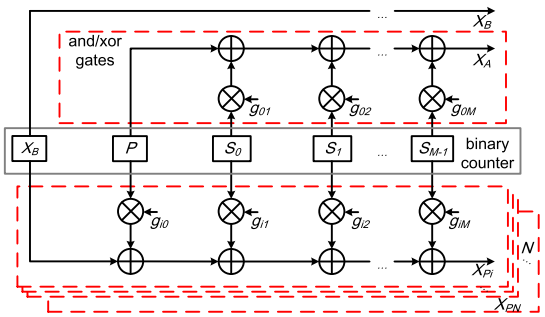

Fig. 2. UEETR-M for decoding the general SB/DB NSC/RSC code.

3 Low-latency UEETR

Before the decoding, the UEETR-M shown in Fig. 2 requires 2^{M+1} and 2^{M+2} cycles to set up the branch symbols of SB radix-2 trellis and DB radix-4 trellis, respectively. The setting time of UEETR-M prolongs the decoding latency when the decoding is switched to a new encoding structure with a large M. Unfolding technique is a straightforward approach to reduce the setting time of the UEETR-M. By using an unfolding parameter f, where $0 \le f \le M+1$, the binary values of f memory elements in the binary counter of UEETR-M are implemented beforehand. The setting time then becomes 2^{-f} of the original one. One output signal then becomes 2^f output signals. Here, we take UEETR-6 for a design example in order to make the CC decoder ($M = 6$) presented in [9] flexible. When $f = 0$, UEETR-6 is original and has eight memory elements in the binary counter. The setting time is 256 cycles. Fig. 3 shows a detailed design of low-latency UEETR-6 with $f = 2$ by setting the binary values of two memory elements S_4 and S_5 beforehand. One output signal X_{Pi} then becomes four output signals, $X_{Pi0} \sim X_{Pi3}$. It reduces the

DOI: 10.1587/elex.14.20170028
Received January 10, 2017
Accepted February 3, 2017
Publicized February 21, 2017
Copyedited March 10, 2017

setting time to 64 cycles. By using the unfolding technique, the four unfolded sets accompany several additional AND/XOR gates. However, some AND gates can be removed because $g_{ij} \otimes 1 = g_{ij}$ and $g_{ij} \otimes 0 = 0$. Some XOR gates can be also removed because $g_{ij} \oplus 0 = g_{ij}$ and $g_{ij} \oplus 1 = (g_{ij})'$. For a small f, these simplifications could decrease the hardware cost of low-latency UEETR.

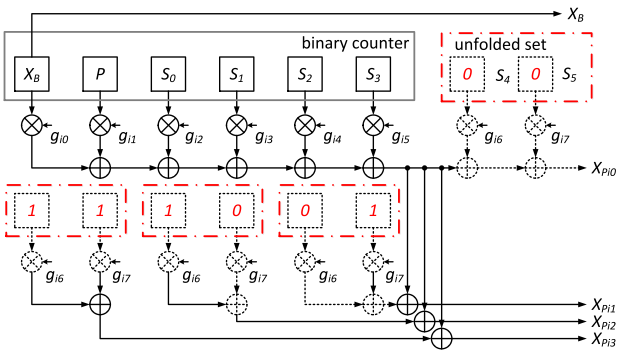

Fig. 3. Low-latency UEETR-6 with $f = 2$.

4 R4UEETR for the radix-4 SB/DB TC decoder

Here, a R4UEETR with $M = 3$ (denoted as R4UEETR-3) is designed for the radix-4 SB/DB MAP decoding. For the radix-4 SB MAP decoding, the R4UEETR-3 generates two time-stage systematic branch bits $(X_{S,0}, X_{S,1})$ and two time-stage branch parity bits $(X_{Y,0}, X_{Y,1})$ based on

$$\begin{cases} X_{S,0} = P_1 \oplus (g_{01} \otimes S_0) \oplus (g_{02} \otimes S_1) \oplus (g_{03} \otimes S_2), \\ X_{S,1} = P_0 \oplus (g_{01} \otimes P_1) \oplus (g_{02} \otimes S_0) \oplus (g_{03} \otimes S_1), \\ X_{Y,0} = (g_{10} \otimes P_1) \oplus (g_{11} \otimes P_1) \oplus (g_{12} \otimes S_0) \oplus (g_{13} \otimes S_1), \\ X_{Y,1} = (g_{10} \otimes P_0) \oplus (g_{11} \otimes P_{k-1}) \oplus (g_{12} \otimes S_0) \oplus (g_{13} \otimes S_1), \end{cases} \tag{2}$$

where P_0 and P_1 are the first two memory elements in the binary counter. For the radix-4 DB MAP decoding, the R4UEETR-3 generates one time-stage systematic branch symbol (X_A, X_B) and one time-stage parity branch symbol (X_{P1}, X_{P2}) based on

$$\begin{cases} X_A = P_1 \oplus (g_{01} \otimes S_0) \oplus (g_{02} \otimes S_1) \oplus (g_{03} \otimes S_2), \\ X_B = P_0, \\ X_{P1} = P_0 \oplus (g_{10} \otimes P_1) \oplus (g_{11} \otimes S_0) \oplus (g_{12} \otimes S_1) \oplus (g_{13} \otimes S_2), \\ X_{P2} = P_1 \oplus (g_{20} \otimes P_0) \oplus (g_{21} \otimes S_0) \oplus (g_{22} \otimes S_1) \oplus (g_{23} \otimes S_2). \end{cases} \tag{3}$$

Fig. 4 shows that the R4UEETR-3 can achieve the hardware-shared AND/XOR gates and output signals from an observation on Eqs. (2) and (3). This design approach can be extended to different M, N, and radix number r. Note that the R4UEETR does not deal with the memory access but generates the branch symbol of radix-4 decoding trellis. The memory access can be referred to in [7, 8] for details.

DOI: 10.1587/elex.14.20170028
Received January 10, 2017
Accepted February 3, 2017
Publicized February 21, 2017
Copyedited March 10, 2017

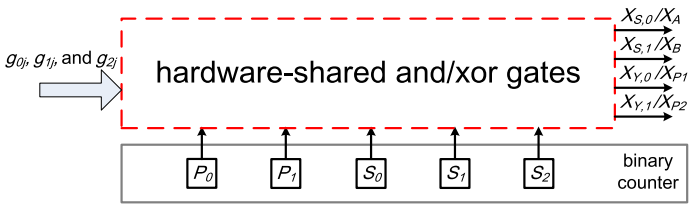

Fig. 4. R4UEETR-3 for the radix-4 trellis decoding.

5 Experimental results

In this Letter, the distinct UEETRs have been implemented and synthesized by using a 90-nm CMOS technology. Table I lists the total gate counts and setting time of the UEETR-6 with distinct f. As f increases, the setting time significantly decreases. The hardware cost slightly decreases for $f \leq 3$ because of the logic simplifications. However, the hardware cost increases rapidly for $f > 3$. If the UEETR-6 with $f = 4$, R4UEETR-3 with $f = 2$, and UEETR-3 with $f = 1$ are assumed to be applied to the SB CC decoder [9], SB/DB TC decoder [8], and SB TC decoder [10] respectively. Table II reveals that the area overheads of UEETRs with the same setting time are less than 2% of the original decoders.

The LTE/WiMAX radix-4 decoding kernel [7] was proposed to decode the WiMAX DB TC, low-density parity check code (LDPCC), and LTE SB TC. When a non-WiMAX matrix permutation is used, this kernel can flexibly decode distinct LDPCCs with a row weight ≤21 by employing radix-4 forward-backward algorithm (FBA). However, it cannot support flexible SB/DB CC and TC decoding with $M \leq 3$ even if a non-LTE/WiMAX interleaving is used. It is because the branch symbols of decoding trellis are fixed to the encoding structures of the LTE/WiMAX TCs. By implementing the R4UEETR-3 with $f = 2$ into the decoding kernel [7], a flexible decoding kernel can be achieved to further decode arbitrary $M \leq 3$ CCs and TCs. In this Letter, a prototyping chip of the flexible decoding kernel has been implemented in core area of 0.48 mm^2 by using the TSMC 90-nm CMOS processes. In Table III, this chip is compared with other distinct decoding kernels [5, 6, 7]. Compared to [7], this chip achieves the flexibility with a 6.7% area overhead. Because the maximum clock frequency of 167 MHz is the same to [7], this chip achieves the same throughputs of 333 Mbps and 800 Mbps for the TCs and LDPCC, respectively. Because of the radix-4 decoding, this chip achieves throughput of 333 Mbps for the CC. Compared to other kernels, this chip supports the standard-free LDPCC, CC, and TC decoding.

Table I. Gate count and setting time evaluation of UEETR-6 with distinct f.

Unfolding parameter f	0	1	2	3	4	5
Total gate count	3921	3906	3904	3915	3961	4066
Setting time (cycle)	256	128	64	32	16	8

DOI: 10.1587/elex.14.20170028
Received January 10, 2017
Accepted February 3, 2017
Publicized February 21, 2017
Copyedited March 10, 2017

Table II. Area overhead of the distinct decoders with the proposed UEETRs.

Decoder type	SB CC [9]	SB/DB TC [8]	SB TC [10]
Decoding algorithm	VA	MAP	MAP
Radix-r	2	4	2
Parallelism	1	8	8
Chip area (mm^2)*	0.77	3.38	1.66
Implemented UEETR type	UEETR-6	R4UEETR-3	UEETR-3
Unfolding parameter f	4	2	1
Setting time (cycle)	8	8	8
UEETR area (μm^2)*	13986	981	920
Area overhead	1.84%	0.03%	0.06%

*Normalized to 90 nm CMOS process

Table III. Chip comparison among distinct decoding kernels.

	This work			TVLSI'15 [7]		TVLSI'08 [5]		TVLSI'11 [6]	
Technology	90 nm			90 nm		180 nm		130 nm	
Core Area (mm^2)	0.48			0.45		2.46		1.28	
Clock Freq. (MHz)	167			167		100		125	
Radix-r	Radix-4			Radix-4		Radix-2		Radix-4	
Standards	Flexible			LTE/ WiMAX		Flexible		LTE/ WiMAX	
Code Type	SB/DB CC	SB/DB TC	LDPCC	SB/DB TC	LDPCC	SB CC	SB TC	SB/DB TC	
Algorithm	MAP	MAP	FBA	MAP	FBA	VA	MAP	MAP	
Throughput (Mbps)	333	333	800	333	800	100	50	500	250

6 Conclusion

In this Letter, the proposed UEETRs provide the decoders with a capability of decoding various SB/DB CC and TC with a small area overhead. In addition, the low-latency UEETR for $f \leq 3$ reduces the setting time with a low hardware cost because of the logic simplifications. To verify the R4UEETR-3, the flexible decoding kernel is implemented in a 0.48 mm^2 chip by using the 90-nm CMOS process. Then, this kernel can decode arbitrary $M \leq 3$ SB/DB CC and TC with a 6.7% area overhead.

Acknowledgments

This work was supported by Ministry of Science and Technology under grants MOST 104-2221-E-155-045 and MOST 103-2221-E-155-070. The authors would like to thank National Chip Implementation Center (CIC) for technical support.

DOI: 10.1587/elex.14.20170028
Received January 10, 2017
Accepted February 3, 2017
Publicized February 21, 2017
Copyedited March 10, 2017