VLSI DESIGN OF DUAL-MODE VITERBI/TURBO DECODER FOR 3GPP

Kai Huang, Fan-Min Li, Pei-Ling Shen and An-Yeu Wu

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

ABSTRACT

In this paper, a prototype design of a dual-mode Viterbi/turbo decoder for 3rd generation wireless communication systems is proposed. By merging some similar modules in both the Viterbi decoder and the log-MAP turbo code decoder, we built one dual-mode decoder with both of these two functions. When the decoder operates in the turbo mode, early-termination control of the iteration process can reduce the power consumption without influencing the decoding accuracy. Besides, in order to conform to the CDMA2000 standard, our decoder can also perform as a reconfigurable Viterbi decoder. That is, our design meets the requirement of the multi generator polynomial convolutional code specification. The design provides an integrated FEC kernel for modern communication systems.

1. INTRODUCTION

Turbo coding was introduced in 1993 by Berrou, Glavieux, and Thitimajashima [2], and it is well known for its extremely superior decoding accuracy. More precisely speaking, the performance of turbo code is closer to the Shannon limit than any other convolutional code today. Due to the outstanding decoding ability, turbo coding got rapid development within just a few years and become standardized. As a result, the 3G mobile wireless communication system standards [3], like CDMA2000 [4] and WCDMA, adopted turbo coding as the channel coding scheme.

In this paper, we intend to provide a total solution for channel coding in 3G systems. Generally, the voice and data streams in these system use different types of coding schemes, such as convolutional code and turbo code. Traditionally, the corresponding Viterbi and turbo code 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. By analyzing the concepts and the architectures of the Viterbi and the turbo code decoders, some circuits sharing techniques are applied to merge the main functions into one decoder. Thus our design performs dual functions and the chip area is only a slightly larger than the original turbo decoder with the same time. In addition, the Log-MAP algorithm [5], which has better *bit error rate* (*BER*) performance than the *soft output Viterbi algorithm* (*SOVA*) [6] and the Max-Log-MAP algorithm, is adopted to decode the data from the turbo encoder. Besides, early-termination with *cyclic redundancy check* (*CRC*) can be adopted for power saving intention. In the Viterbi mode, our reconfigurable design can be applied to specifications with different generator polynomials.

This work was supported by the MediaTek Inc., under NTU-MTK wireless research project.

2. REVIEW OF LOG-MAP ALGORITHM

We will briefly describe the result of the Log-MAP Algorithm. First, the Max-Log-MAP algorithm simplifies the MAP algorithm [1] by transferring these equations into the log arithmetic domain and then using the approximation

$$\ln\left(\sum_i e^{x_i}\right) \approx \max_i \left(x_i\right). \tag{1}$$

Then, with $A_k(s)$, $B_k(s)$ and $\Gamma_k(`s,s)$ defined and rewritten as follows:

$$A_k(s) \approx \max_{`s}\left(A_{k-1}(`s) + \Gamma_k(`s,s)\right) \tag{2}$$

with the same rule

$$B_{k-1}(s) \approx \max_{`s}(B_k(`s) + \Gamma_k(s,`s)) \tag{3}$$

and

$$\Gamma_k(`s,s) = C + \frac{1}{2}u_k L(u_k) + \frac{L_c}{2}\sum_{i=1}^{n} y_{k_i} x_{k_i} \quad . \tag{4}$$

$u_k L(u_k)$ is the *a priori* LLR term and the correlation term $\sum_{i=1}^{n} y_{k_i} x_{k_i}$ is weighted by the channel reliability value L_c, where n is the number of each codeword. Finally, we can write for the *a posteriori Log Likelihood Ratio* (*LLR*) as

$$L(u_k|y) \approx \max_{(`s,s)\in Bt^1}(A_{k-1}(`s)+\Gamma_k(`s,s)+B_k(s)) - \max_{(`s,s)\in Bt^0}(A_{k-1}(`s)+\Gamma_k(`s,s)+B_k(s)) \quad (5)$$

Bt^1 and Bt^0 are individual defined as transitions caused by $u_k = 1$ and $u_k = 0$. Because of the approximation we applied, the Max-Log-MAP algorithm is suboptimal and the problem can be fixed by using the Jacobian logarithm[5]:

$$\ln\left(e^{\delta_1}+e^{\delta_2}\right) = \max(\delta_1,\delta_2) + f_c\left(|\delta_1-\delta_2|\right), \quad (6)$$

where $f_c\left(|\delta_1-\delta_2|\right)$ is a correction function. We apply this rule to the Log-MAP algorithm by compensating for one correction term.

3. PROPOSED ARCHITECTURE OF THE DUAL-MODE VITERBI/TURBO DECODER

3.1. Overall Architecture

The overall architecture of the proposed dual-mode Viterbi/turbo code system is shown in Fig. 1. The component decoder has two modes and some of the modules are shared. In Viterbi mode, β and *LLR* processors are turned off. Data input goes from *Branch Metric Unit* (*BMU*) processor to *Add Compare Select Unit* (*ACSU*) processor, and then goes out after tracing back. When in turbo mode, the decoder works as a MAP decoder and only *Trace Back Unit* (*TBU*) is turned off.

Fig. 2 shows the block diagram of the MAP/Viterbi decoder, including control, logic and memory modules. The word length of each bus is marked according to the fixed-point analysis. The operating principle of our decoder in turbo mode is based on the sliding windows method consumes half the memory size and half the latency compare to the direct flow.

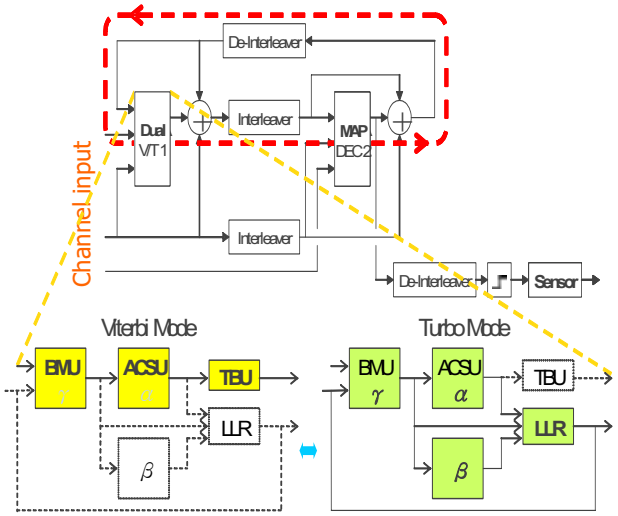

Fig. 1. Block diagram of dual-mode Viterbi/turbo code system.

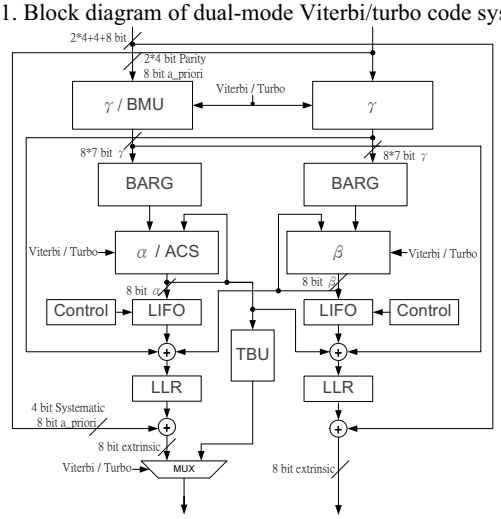

Fig. 2. Block diagram of dual mode Viterbi/turbo decoder (DUAL V/T 1)

3.2. Viterbi Mode

For Viterbi decoding, the decoder uses about half of the architecture, as shown in Fig. 3. The BMU computes the branch metrics and fetch the value to the ACS module via the *BMU to ACS Routing Generator* (*BARG*). In the ideal case, we can achieve a reconfigurable decoder according to the encoder type, including different code rate and encoder structures. In Viterbi mode, data type like voice does not need very high data rate, so we can use hardware reuse or folding techniques to implement Viterbi decoder with high constraint length if we do not accept extra hardware consumption in our dual-mode design comparing to the one-mode turbo decoder. TBU uses trace back method instead of register exchange method because of the power concern and the lower data rate acquirement.

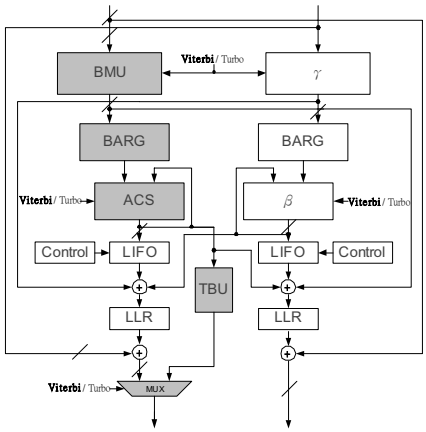

Fig. 3. Block diagram of MAP/Viterbi decoder (Viterbi mode).

3.3. Turbo Mode

For turbo decoding, the decoder uses all the modules except the TBU, as shown in Fig. 4. In our chip, a sliding window Log-MAP decoder is adopted as the SISO decoder, and the window size is 24.

The two γ modules (the module performs BMU in Viterbi mode) compute the γ values. For an 8 states trellis diagram, 16 γ values are generated from one input symbol. The γ values need not to be stored but passed to α or β module (the module performs ACS in Viterbi mode) via BARG. The feedback loop from the output to the input of the α and β modules indicate the recursive calculation, and we store these α and β values at each state of each stage in RAM.

We keep the previous values of α and β in memory because the computation of these values in a trellis diagram takes different latency time, except at the middle point of a trellis sequence. To reduce the total latency time and the power consumption, one method is to store the values of α and β. Then they are calculated and look up these values from memory. Besides, if we execute these steps from both head and tail simultaneously, the storing action can be stopped when the computation crosses the middle point and the LLR values can be generated immediately by one looked up value, one real time value, and γ. Finally, sum up the LLR value, the systematic bit and the a priori information together and we will get the extrinsic information

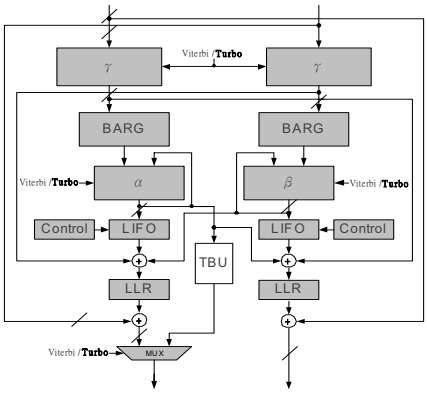

Fig. 4. Block diagram of MAP/Viterbi decoder (Turbo mode).

4. VLSI DESIGN OF DUAL MODE MAP/VITERBI DECODER

4.1. γ / BMU Architecture

The γ / BMU module shown in Fig. 5 is the first computing unit in a decoder. All the branches in a trellis diagram need to generate one value, which represents the Hamming distance between the received information and the code on each branch. When we move to the Log-MAP algorithm, one more term is added to the original branch metrics to give a more conscientious and more careful formula

$$\begin{aligned}\Gamma_k(`s,s) &\equiv \ln\left(\gamma_k(`s,s)\right) \\ &= -\ln\left(e^{-u_k L(u_k)}+1\right)+2\sum y_{kl}x_{kl}\end{aligned} \tag{7}$$

where u_k means the sign of the branch and $L(u_k)$ is the a priori information. By increasing iterations, a-priori information $L(u_k)$ will be more accurate. The code rate of the turbo code specification in CDMA2000 is 1/2, thus there exists four situations, 00, 01, 10 and 11 for computing branch metric in the trellis. But in turbo coder, because the sign of branches are also taken into consideration, eight situations then are generated [7].

When in the Viterbi mode, the hardware only executes the traditional hamming distance computing action and only four outputs are required.

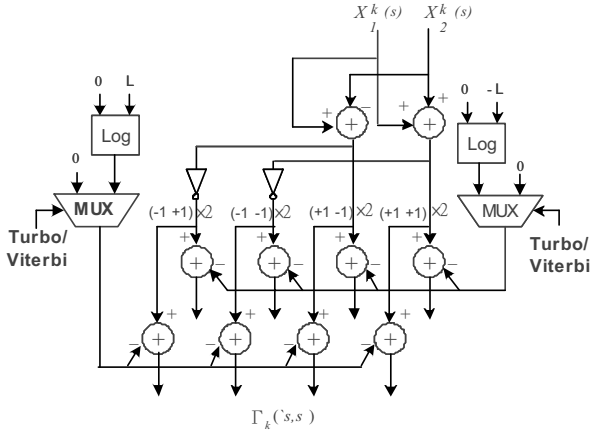

Fig. 5. Dual-mode γ / BMU architecture.

4.2. α / ACS and β / ACS Architecture

The computation of the α and β is demonstrated below. This process is similar to the ACS in Viterbi decoder. In order to simplify the circuit and to reduce the power consumption, the Log-MAP algorithm transfers the original equation to the log domain. The inaccuracy will be compensated by the look up table. At the same time, we find that the former part, $\max_{`s}\left(A_{k-1}(`s)+\Gamma_k(`s,s)\right)$, exactly performs the ACS action in the Viterbi decoder. Thus, we can achieve the dual-mode α / ACS and β / ACS by using a switch to change circuit function between the two modes.

The module in Fig. 6 is the dual-mode α and β computation module, which is composed of one adder, one substracter, one inverter, three multiplexers, the look up table circuit and the comparison circuit. Because the input of the α / ACS and β / ACS modules are quantized as seven bits integer plus one decimal fraction, so we reserve only two elements of the compensation table. The

two first items are then quantized as 1 and 0.5, and can be easily implemented in the circuit without a ROM base look up table [8].

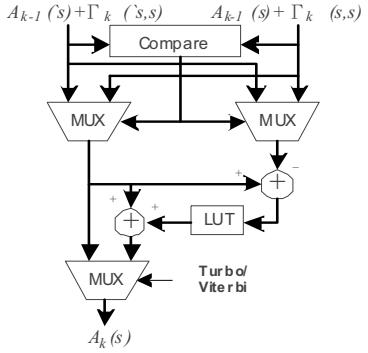

Fig. 6. Dual-mode α / ACS and β / ACS architecture.

4.3. Dual-Mode BARG Architecture

The decoding principle of the convolutoinal code is based on the trellis diagram according to the convolutional encoder. Different encoder structures certainly map to trellises with diverse sizes and truth tables. According to the CDMA2000 standard, the constraint length of Viterbi encoder is 9, different from the turbo encoder's constraint length, 4. Besides, the convolutional code in the standard has several suits of generator polynomials. Since our goal is to build a dual-mode decoder, synthesizing a circuit meeting different specifications becomes a challenge to be overcome. Thus, our dual-mode system in fact includes one turbo decoder and one reconfigurable Viterbi decoder. Not like the previous described γ / BMU, α / ACS and β / ACS modules, which are only the computing units with two modes, the BARG module is the key component that can be programmed to link the whole system to execute decoding process according to the encoder specification [9].

5. CONCLUSIONS

A practical design of a dual-mode convolutional/turbo code decoder for CDMA 2000 is proposed and successfully implemented. The chip summary is list in Tab. 1. The method we use here to combine the Viterbi and the Log-MAP decoder is based on the similarities of the innate characters between these two algorithms. Although this work is developed for the CDMA2000 standard, the basic principle of dual-mode and multi specifications for channel coding design can also be easily adopted to other advanced communication system standards. We compare our work to two similar designs in Tab.2. First, a unified turbo / Viterbi channel decoder for 3GPP mobile wireless is proposed by Lucent Inc. and Bell Lab. Second, a programmable turbo decoder for multiple 3G wireless standards is proposed by KAIST.

Tab. 1 CHIP summary

Technology	Artisan 0.25um 1P5M
Gate count of logic cell	92,300
Supply voltage	2.5v
Convolutional code size	$k = 6$
Maximum operating frequency	80MHz
Maximum data rate (Viterbi mode)	40Mb/s
Maximum turbo decoder data rate	80Mb/s
Maximum turbo code decoding rate	6.67Mb/s (6 iterations)
Die size	2.8mm x 2.8 mm

Tab. 2 comparisons

	Ours	Lucent [10]	KAIST [11]
Technology	0.25 μm	0.18 μm	0.25 μm
Viterbi embedded?	Yes (32 states), programmable	Yes (256 states), programmable	No
Gate count	96,000	140,000	34,400
Voltage	2.5	1.8	2.5
Max. Freq.	80MHz	110MHz	135MHz
Max. Throughput	6.67Mb/s (6 iterations)	4.27Mb/s (6 iterations)	5.48Mb/s (6 iterations)

6. REFERENCES

[1] L. Bahl, J, Cocke, F. Jelinck, 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.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. ICC*, pp. 1064–1070, May 1993.

[3] 3GPP TSG-RAN Working Group 1 (2001, Sept.) *Physical channels and mapping of transport channels (FDD) (Release 4)*, TS25.211 v4.2.0.

[4] P. Robertson, "Illuminating the structure of parallel concatenated recursive systematic (TURBO) codes," in *Proc. GLOBECOM '94* (San Francisco, CA, Nov. 1994), vol. 3, pp. 1298-1303.

[5] P. Robertson, E.Villebrun and P. Hocher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," *IEEE Int. Conf.* Jun 1995, vol.2, pp1009-1013.
Woodard, J.P.; Hanzo, L. Vehicular Technology "Comparative study of turbo decoding techniques: an overview", *IEEE Trans. on , Vol. 49* Issue: 6 , Nov. 2000, pp. 2208 –2233.

[6] G. D. Forney, "Convolutional Codes II: Maximum Likelihood Decoding," *IEEE Trans. on Information and Control*, Vol. 25, July 1974, pp. 222-226.

[7] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. on Information Theory*, Vol. IT-13, April 1967, pp. 260-269.

[8] Kai Huang, "VLSI Design of Dual-Mode Viterbi/Turbo Decoder for 3rd GPP Systems ," *NTU thesis*, 2003.

[9] Pin-Shun Huang, "Design and implementation of a reconfigurable Viterbi decoder," *NCU thesis*, 2001.

[10] M. Bickerstaff et al, "A unified turbo/viterbi channel decoder for 3GPP mobile wireless in 0.18 μm CMOS," *Solid-State Circuits Conference. Digest of Technical Papers. ISSCC. IEEE International* , vol. 1, 2002, pp. 124-451.

[11] M. C. Shin and I. C. Park, "A programmable turbo decoder for multiple 3G wireless standards," *Solid-State Circuits Conference. Digest of Technical Papers. ISSCC. IEEE International*, 2003.