The 2004 IEEE Asia-Pacific Conference on
Circuits and Systems, December 6-9, 2004

# REALIZATION OF A SYSTEMATIC BIT-WISE DECOMPOSITION METRIC

*Chia-Wei Chang*[1], *Po-Ning Chen*[1] *and Yunghsiang S. Han*[2]

sekwan.cm87@nctu.edu.tw, qponing@mail.nctu.edu.tw, yshan@csie.ncnu.edu.tw

[1]Dept. of Communications Engineering
National Chiao-Tung University
Hsin Chu, Taiwan, ROC

[2]Dept. of Computer Science and Info. Eng.
National Chi Nan University
Pu Li, Taiwan, ROC

## ABSTRACT

In this paper, a realization structure for our previously proposed systematic recursive formula for bit-wise decomposition of $M$-ary symbol metric is proposed, which can be applied to reduce the memory space and processing latency of a system where the information sequence is binary-coded and interleaved before $M$-ary modulated. Different from conventional structure where de-interleaver and decoder are separate circuits, our structure de-interleaves and decodes at the same time.

## 1. INTRODUCTION

The state-of-the-art wireless transmission technique of IEEE 802.11a/g [4, 5] incorporated high QAM into OFDM to achieve a high data rate. In order to make a better use of the error correcting capability of the adopted (2,1,6) convolutional code, the standard specified a two-step bit-wise interleaver, where the first step maps adjacent code bits onto non-adjacent sub-carriers, and the second step permutes the code bits alternately onto less and more significant bits of the QAM constellation (cf. Fig. 2).

In order to design a general receiver for use of $M$-ary symbol transmission of coded and interleaved information sequence, we proposed a bit-wise systematic recursive decomposition of $M$-ary symbol metric, which was named *Soft-proposed* in [1]. Interestingly, our recurve formula can be reduced to the bit reliability decomposition proposed by Tosato and Bisaglia with a different first-bit metric function [7], which is abbreviated as *Soft-TB* in [1]. Hence, an implementation of both Soft-proposed and Soft-TB metrics can be simultaneously obtained.

Traditionally, realization of the Viterbi decoder can be divided into three units [3, 8], as shown in Fig. 1. The input data is used in the branch metric unit (BMU) to calculate the branch metrics for each new time step. These metrics are then fed to the add-compare-select unit (ACSU), which accumulates the branch metrics as the path metric (PM) stored in the path metric memory (PMM) according to the ACS-recursion. The survivor memory unit (SMU) processes the decisions which are being made in ACSU, and outputs the estimated path with a latency of at least D, called the survivor depth. Other than the combination design of interleaver and decoder, we will provide a dual-mode (Soft-proposed/Soft-TB) system architecture for BMU .

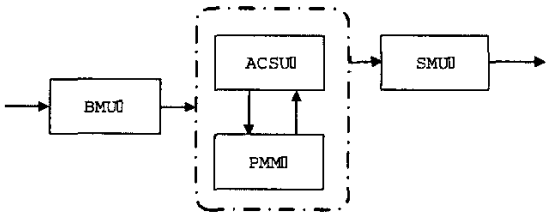


**Fig. 1**. A simplified block diagram of the Viterbi decoder

## 2. BIT-WISE DECOMPOSITION OF $M$-ARY SYMBOL METRIC

In 2003, we have proposed a general soft-output demapper for $M$-ary modulation system [1]. We then examined its performance under the setting of IEEE 802.11a/g [4, 5]. Specifically, for $M^2$-QAM with amplitude spacing $u$, the bit-wise decomposition soft metric is given by:

$$f_1^{(1)}(c,r) = c \cdot \mathrm{sgn}(-r) \cdot |r|,$$

0-7803-8660-4/04/$20.00 ©2004 IEEE

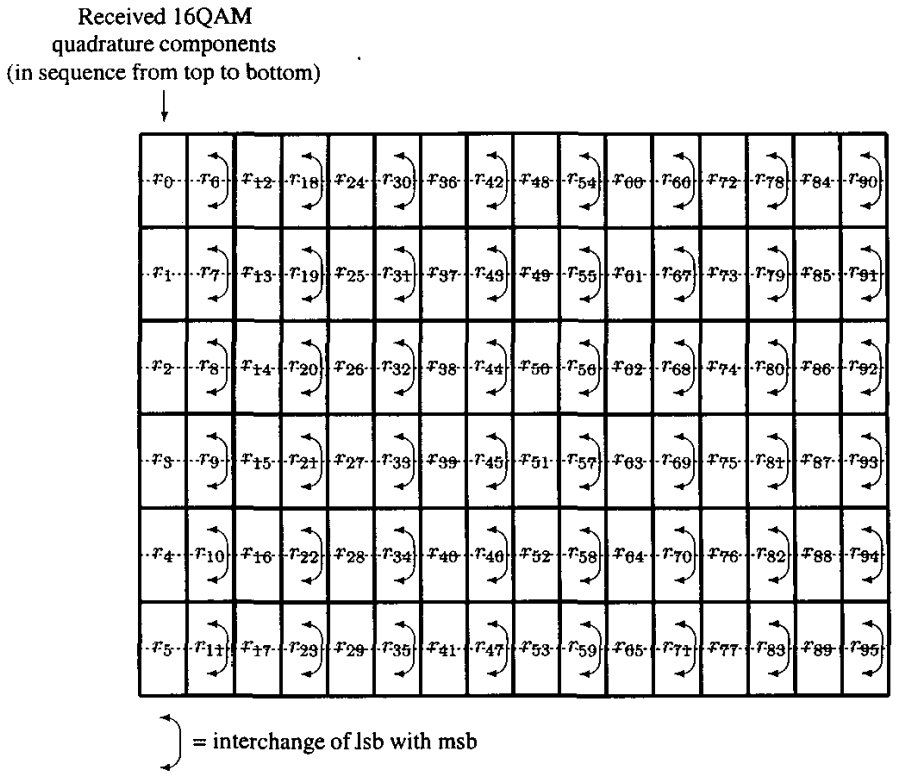


**Fig. 2.** (a) The received 16QAM quadrature components are indicated by a solid-lined upright rectangular; each column consists of 6 quadrature components. The de-interleaver should then exchange (the weights of) the least significant bit and the most significant bit of the quadrature components for those columns indicated.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ | $c_9$ | $c_{10}$ | $c_{11}$ | $c_{12}$ | $c_{13}$ | $c_{14}$ | $c_{15}$ | $c_{16}$ |
| $c_{17}$ | $c_{18}$ | $c_{19}$ | $c_{20}$ | $c_{21}$ | $c_{22}$ | $c_{23}$ | $c_{24}$ | $c_{25}$ | $c_{26}$ | $c_{27}$ | $c_{28}$ | $c_{29}$ | $c_{30}$ | $c_{31}$ | $c_{32}$ |
| $c_{33}$ | $c_{34}$ | $c_{35}$ | $c_{36}$ | $c_{37}$ | $c_{38}$ | $c_{39}$ | $c_{40}$ | $c_{41}$ | $c_{42}$ | $c_{43}$ | $c_{44}$ | $c_{45}$ | $c_{46}$ | $c_{47}$ | $c_{48}$ |
| $c_{49}$ | $c_{50}$ | $c_{51}$ | $c_{52}$ | $c_{53}$ | $c_{54}$ | $c_{55}$ | $c_{56}$ | $c_{57}$ | $c_{58}$ | $c_{59}$ | $c_{60}$ | $c_{61}$ | $c_{62}$ | $c_{63}$ | $c_{64}$ |
| $c_{65}$ | $c_{66}$ | $c_{67}$ | $c_{68}$ | $c_{69}$ | $c_{70}$ | $c_{71}$ | $c_{72}$ | $c_{73}$ | $c_{74}$ | $c_{75}$ | $c_{76}$ | $c_{77}$ | $c_{78}$ | $c_{79}$ | $c_{80}$ |
| $c_{81}$ | $c_{82}$ | $c_{83}$ | $c_{84}$ | $c_{85}$ | $c_{86}$ | $c_{87}$ | $c_{88}$ | $c_{89}$ | $c_{90}$ | $c_{91}$ | $c_{92}$ | $c_{93}$ | $c_{94}$ | $c_{95}$ | $c_{96}$ |
| $c_{97}$ | $c_{98}$ | $c_{99}$ | $c_{100}$ | $c_{101}$ | $c_{102}$ | $c_{103}$ | $c_{104}$ | $c_{105}$ | $c_{106}$ | $c_{107}$ | $c_{108}$ | $c_{109}$ | $c_{110}$ | $c_{111}$ | $c_{112}$ |
| $c_{113}$ | $c_{114}$ | $c_{115}$ | $c_{116}$ | $c_{117}$ | $c_{118}$ | $c_{119}$ | $c_{120}$ | $c_{121}$ | $c_{122}$ | $c_{123}$ | $c_{124}$ | $c_{125}$ | $c_{126}$ | $c_{127}$ | $c_{128}$ |
| $c_{129}$ | $c_{130}$ | $c_{131}$ | $c_{132}$ | $c_{133}$ | $c_{134}$ | $c_{135}$ | $c_{136}$ | $c_{137}$ | $c_{138}$ | $c_{139}$ | $c_{140}$ | $c_{141}$ | $c_{142}$ | $c_{143}$ | $c_{144}$ |
| $c_{145}$ | $c_{146}$ | $c_{147}$ | $c_{148}$ | $c_{149}$ | $c_{150}$ | $c_{151}$ | $c_{152}$ | $c_{153}$ | $c_{154}$ | $c_{155}$ | $c_{156}$ | $c_{157}$ | $c_{158}$ | $c_{159}$ | $c_{160}$ |
| $c_{161}$ | $c_{162}$ | $c_{163}$ | $c_{164}$ | $c_{165}$ | $c_{166}$ | $c_{167}$ | $c_{168}$ | $c_{169}$ | $c_{170}$ | $c_{171}$ | $c_{172}$ | $c_{173}$ | $c_{174}$ | $c_{175}$ | $c_{176}$ |
| $c_{177}$ | $c_{178}$ | $c_{179}$ | $c_{180}$ | $c_{181}$ | $c_{182}$ | $c_{183}$ | $c_{184}$ | $c_{185}$ | $c_{186}$ | $c_{187}$ | $c_{188}$ | $c_{189}$ | $c_{190}$ | $c_{191}$ | $c_{192}$ |

→ Convolutional code bits (from left to right)

**Fig. 2.** (b) After de-interleaving, each 2-bit trellis branch for convolutional decoding is indicated by the flat rectangular in sequences from left to right.

$$f_1^{(m)}(c\ r) = c \cdot \mathrm{sgn}(-r) \sum_{i=-(m-2)}^{(m-2)} (|r+2ui| - |2ui|)$$

for $m \geq 2$, and

$$f_j^{(m)}(c\ r) = f_{j-1}^{(m-1)}\left(c\ (-1)^j \left[2^{m-2}u - |r|\right]\right)$$

for $2 \leq j \leq m$ and $m = \log_2(M) \geq 2$, where $r$ denotes the received QAM quadrature component and $c \in \{0\ 1\}$ is the code bit. A recursive bit-metric decomposition formula is then established. Notably, $m$ is respectively 2 and 3 for 16QAM and 64QAM.

Our recursive formula can be directly applied to the soft bit reliability[1] proposed by Tosato and Bisaglia [7] in 2001 for binary interleaved COFDM with application to HIPERLAN/2 [2] with a different first-bit metric function as:

$$g_1^{(m)}(c\ r) \quad = \quad c(-r) \quad \text{for } m = 1\ 2\ 3$$

This suggests that by varying the first-bit metric, variants of bit-decomposed metrics can be resulted through the recursive formula we established.

## 3. ARCHITECTURE OF BRANCH METRIC UNIT

In this section, we present a BMU architecture that can be used for both Soft-proposed/Soft-TB metrics and $2^{2m}$-QAM modulations for every $2 \leq m \leq 5$. An extension of this architecture to higher QAM (i.e., $m > 5$) can be easily obtained by adding more MTUs. Specifically, an BMU with $(m_{\max} - 2)$ serially connected MTUs can do 16QAM, 64QAM, , up to $2^{2m_{\max}}$-QAM.

Under the setting of IEEE 802.11a/g, an OFDM signal consists of 96 $2^{2m}$-QAM symbols, and each $2^{2m}$-QAM symbol was mapped from $m$ coded and interlaved bits.

The decoding sequences from left to right can be divided into 6 rows, where each row consists of 16 QAM quadrature symbols. For example, the first row in Fig. 2(a) contains $\{r_0\ r_6\ r_{12}\ \ r_{90}\}$, and the symbols in $\{r_1\ r_7\ r_{13}\ \ r_{91}\}$ belong to the second row. An internal control variable, named Index_c, is used to keep the record of which column of QAM quadrature symbol in Fig. 2(a)-block the BMU is currently used. Consequently, Index_c is initially zero for each new 96 QAM symbol block, and is updated according to the rule of Index_c = (Index_c+1) mod 16. Another internal control variable, Sym_bit, is used to adjust the demapped bit number in each QAM symbol according to the interleaver rule. It is equal to $(6 \times m - 1)$ initially for each new 96 QAM symbol block, and is reduced by one when Index_c equals 0. We also record the position of the QAM quadrature component (in the 96 QAM symbol block) that is currently used. As a result, position = $6 \times$Index_c $+ 5 -$ [Sym_bit $m$], where $\lfloor \cdot \rfloor$ denotes the floor function. All the internal control variables are updated whenever a new QAM quadrature symbol enters the BMU. Notably, for $2^{2m}$-QAM demodulation, each QAM quadrature symbol has to be re-accessed $m$ times; for example, the BMU needs to generate $192 = 2 \times 96$ soft bit metrics for 16QAM demodulation in Fig. 2.

To combine the implementation of both our and Tosato/Bisaglia metrics, two metric calculation modes are set as: "Metric-mode = 1" is for Soft-proposed, and "Metric-mode = 0" is for Soft-TB. A forth internal control variable, $q$ = (Index_c+Sym_bit) mod $m$, is used to determine the bit metric function number in a way that either function $f_1^{(q+1)}(\cdot\ \cdot)$ or function $g_1^{(q+1)}(\cdot\ \cdot)$ is used depending on the Metric-mode. The last internal control variable, $n$, is simply $m - 1 - q$. This dual-mode BMU architecture is depicted in Fig. 3.

Our BMU requires two external input signals, which are *(i)* the received QAM quadrature symbol $r$ that are derived from proper position of the 96 symbol block, and *(ii)* $m = \log_2(M)$ if what has been received is $M^2$-QAM symbol (cf. Fig. 4).

Finally, since the soft bit metric equals zero when the branch bit $c$ is zero, the BMU only evaluates bit metric values for $c = 1$. The case for "c=0 is then handled by a check box appended at the output end of the BMU.

[1] In fact, their soft-output demapper has been appeared in a book published in 1997 [6]. In the book, the soft-output demapper is heuristically obtained through a direct derivation, as opposed to the simplified-from-LLR-decision approach taken by Tosato and Bisaglia.

## 4. REFERENCES


[1] C.-W. Chang and P.-N. Chen, "On bit-wise decomposition of $M$-ary symbol metric," *2003 International Conference on Informatics Cybernetics and Systems (ICICS2003)*, Kaohsiung, Taiwan, December 14-16, 2003.

[2] ETSI TS 101 475, *Broadbadn Radio Access*

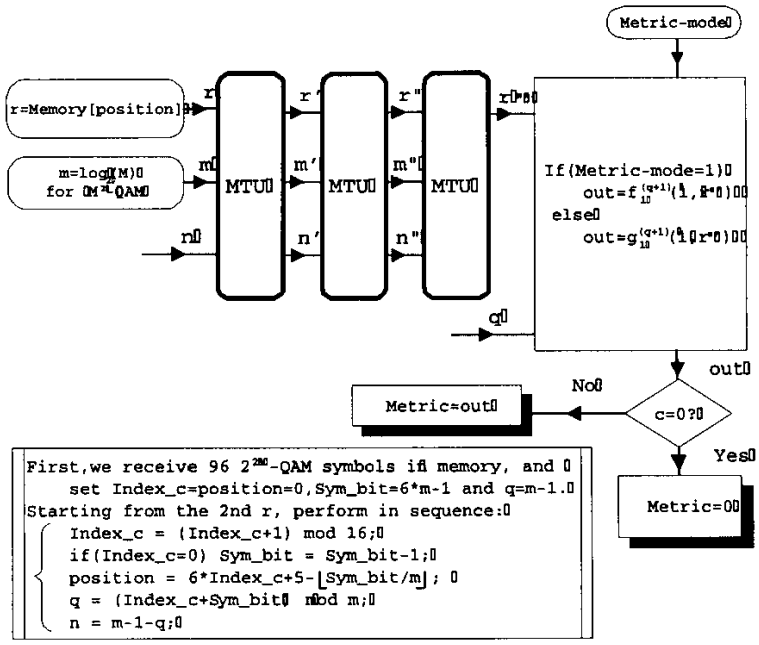


**Fig. 3.** A dual-mode BMU architecture with $(m_{\max} - 2)$ MTU's can perform the bit metric evaluations for $2^{2m}$-QAM, where $2 \leq m \leq m_{\max}$. Here, $m_{\max} = 5$. All the constants in the formulas of Index_c, Sym_bit and position, such as 5 $(= 6 - 1)$, 6 and 16, are set because of the $6 \times 16$ interleaver block used in IEEE 802.11a/g standard.

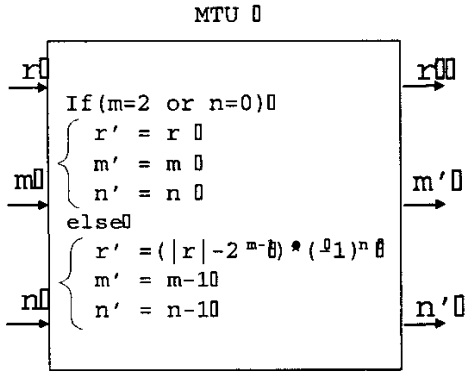


**Fig. 4.** The metric transition unit (MTU) in Fig. 3.

*Networks (BRAN); HIPERLAN Type 2; Physical (PHY) layer*, v1.2.2, 2001.

[3] G. Fettweis, "Algebraic survivor memory management design for Viterbi detectors," *IEEE Trans. Communications*, vol. 43, Iss. 9, pp. 2458 2463, September 1995.

[4] IEEE Std 802.11a-1999, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed Physical Layer in the 5 GHz band*, Sept. 1999.

[5] IEEE Draft 802.11g, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Further Higher Data Rate Extension in the 2.4 GHz band*, Draft 8.2, Apr. 2003.

[6] Richard van N. R. Prasad, *OFDM for Wireless Multimedia Communications*, Artech House, May 1997.

[7] F. Tosato and P. Bisaglia, "Simplified soft-output demapper for binary interleaved COFDM with application to HIPERLAN/2," *HP Labs 2001 Technical Report*, HP-2001-246, 2001.

[8] C.-Y. Tsui, R.S.-K. Cheng and C. Ling, "Low power ACS unit design for the Viterbi decoder," Prof. *of the 1999 IEEE International Symposium on Circuits and Systems (ISCAS'99)*, vol. 1:30, pp. 137 140, June 1999.