

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

SAMSUNG ELECTRONICS CO., LTD.  
Petitioner

v.

CALIFORNIA INSTITUTE OF TECHNOLOGY  
Patent Owner

---

Patent No. 7,421,032

---

**DECLARATION OF MATTHEW C. VALENTI, PH.D., P.E.  
IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW  
OF U.S. PATENT NO. 7,421,032**

## TABLE OF CONTENTS

I.	INTRODUCTION .....	1
II.	BACKGROUND AND QUALIFICATIONS .....	2
III.	MATERIALS REVIEWED .....	7
IV.	PERSON OF ORDINARY SKILL IN THE ART AND THE TIME OF THE ALLEGED INVENTION .....	9
V.	TECHNICAL BACKGROUND .....	10
A.	Error Control Coding .....	10
B.	Matrix Multiplication and Modulo-2 Arithmetic .....	12
VI.	THE '032 PATENT .....	15
VII.	CLAIM CONSTRUCTION .....	18
VIII.	OVERVIEW OF THE PRIOR ART .....	19
A.	<i>Kobayashi</i> (Ex. 1005) .....	19
B.	<i>McEliece</i> (Ex. 1006) .....	23
IX.	THE PRIOR ART DISCLOSES AND/OR SUGGESTS THE RECITED FEATURES OF CLAIMS 1-8 and 10-22 OF THE '032 PATENT .....	28
A.	<i>Kobayashi</i> Discloses the Features of Claims 1, 3-5, 7-8, 11-12, and 14-16 .....	28
1.	Claim 1 .....	28
2.	Claim 3 .....	44
3.	Claim 4 .....	44
4.	Claim 5 .....	45
5.	Claim 7 .....	48
6.	Claim 8 .....	49

Declaration of Matthew C. Valenti, Ph.D., P.E.  
U.S. Patent No. 7,421,032

7.	Claim 11 .....	49
8.	Claim 12 .....	56
9.	Claim 14 .....	56
10.	Claim 15 .....	58
11.	Claim 16 .....	59
B.	<i>Kobayashi</i> Discloses and/or Suggests the Features of Claims 2, 6-7, 10, 13, and 17 .....	60
1.	Claim 2 .....	60
2.	Claim 6 .....	63
3.	Claim 7 .....	71
4.	Claim 10 .....	73
5.	Claim 13 .....	75
6.	Claim 17 .....	76
C.	<i>Kobayashi</i> in View of <i>McEliece</i> Discloses and/or Suggests the Features of Claims 6, 13, and 18-22 .....	79
1.	Claim 6 .....	79
2.	Claim 13 .....	91
3.	Claim 18 .....	92
4.	Claim 19 .....	100
5.	Claim 20 .....	101
6.	Claim 21 .....	101
7.	Claim 22 .....	104
X.	CONCLUSION .....	105

I, Matthew C. Valenti, Ph.D., P.E., declare as follows:

## **I. INTRODUCTION**

1. I have been retained by Samsung Electronics Co., Ltd. (“Petitioner”) as an independent expert consultant in this proceeding before the United States Patent and Trademark Office (“PTO”) against California Institute of Technology (“Patent Owner”) regarding U.S. Patent No. 7,421,032 (“the ’032 patent”) (Ex. 1001).<sup>1</sup> I have been asked to consider whether certain references disclose or suggest the features recited in claims 1-8 and 10-22 (“the challenged claims”) of the ’032 patent. My opinions are set forth below.

2. I am being compensated at a rate of \$525/hour for my work in this proceeding. My compensation is in no way contingent on the nature of my findings, the presentation of my findings in testimony, or the outcome of this or any other proceeding. I have no other interest in this proceeding.

---

<sup>1</sup> Where appropriate, I refer to exhibits that I understand are to be attached to the petition for *inter partes* review of the ’032 patent.

## **II. BACKGROUND AND QUALIFICATIONS**

3. I presently serve as a Professor in the Lane Department of Computer Science and Electrical Engineering at West Virginia University. All of my opinions stated in this declaration are based on my own personal knowledge and professional judgment. In forming my opinions, I have relied on my knowledge and experience in signal processing, communications, and coding referenced in this declaration.

4. I am over 18 years of age and, if I am called upon to do so, I would be competent to testify as to the matters set forth herein. I understand that a copy of my current curriculum vitae (CV), which details my education and professional and academic experience, is being submitted by Petitioner as Exhibit 1003. The following provides an overview of some of my experience that is relevant to the matters set forth in this declaration.

5. I received my Ph.D. in Electrical Engineering from Virginia Polytechnic Institute & State University (Virginia Tech) in 1999. I also received a MS in Electrical Engineering from Johns Hopkins University in 1995, and a BS in Electrical Engineering from Virginia Tech in 1992.

6. Since receiving my Ph.D., I have been teaching electrical engineering at West Virginia University. I was an Assistant Professor between August 1999 and 2005, an Associate Professor between 2005 and 2010, and a Professor from 2010 to

present. In recent years, I have spent time as Interim Chair (July 2019 to June 2020) and Raymond J. Lane Department Chair (July 2020 to June 2021) of the Lane Department of Computer Science and Electrical Engineering.

7. As described in my CV, I am a licensed Professional Engineer and have more than 30 years of industry and academic experience, including extensive experience in the areas of signal processing, communications, and coding. For example, in addition to my academic experience, I worked as an electronics engineer at the U.S. Naval Research Laboratory between May 1992 and August 1995, where I developed systems and algorithms for the processing and analysis of signals such as those received over antenna arrays.

8. As also described in my CV, I am a member of the Institute of Electrical and Electronic Engineers (IEEE), which is the world's largest technical professional organization dedicated to advancing technology for the benefit of humanity, and in 2018, I was elevated to the rank of *IEEE Fellow*, which is an honor reserved for the top one-tenth of one percent of IEEE members annually.

9. At West Virginia University, I have taught courses in communications, signal processing, probability theory, coding theory, systems theory, wireless networking, and digital signal processing. My coding theory course—EE 567—is a graduate-level course taught every-other year covering the topics of linear codes,

generator and parity-check matrices, Hamming codes, convolutional codes, turbo codes, LDPC codes, interleaving, Tanner graphs, and iterative decoding. As a professor, I have been the main advisor to over 50 Master's and Doctoral students.

10. I am the author of several book chapters covering the area of channel coding, including a chapter entitled "The Interplay Between Modulation and Channel Coding" in the book *Transmission Techniques for Digital Communications* (Elsevier, 2016), a chapter entitled "Turbo and LDPC Codes for Digital Video Broadcasting" in the book *Turbo Code Applications: A Journey from a Paper to Realization* (Springer, 2005), and a chapter entitled "Turbo Codes" in the book *Handbook of RF and Wireless Technologies* (Newnes Press, 2004).

11. I have authored, and coauthored, more than 150 papers in the areas of signal processing, communications, and coding. Many of these papers involve the design of error-correcting codes. For instance, in the paper "Constellation Shaping for Bit-Interleaved LDPC Coded APSK," published in the October 2012 issue of *IEEE Transactions on Communications*, I designed a low-density parity check (LDPC) code to be used with a modulation format commonly utilized by digital satellite broadcasting systems. In another paper "Coherent continuous-phase frequency-shift keying: Parameter optimization and code design," published in the April 2009 issue of *IEEE Transactions on Wireless Communications*, I designed a

low-density generator matrix (LDGM) code to be used with another modulation format.

12. I have received numerous awards for my work from the Benjamin M. Statler College of Engineering and Mineral Resources, which houses the Lane Department of Computer Science and Electrical Engineering at West Virginia University. I was recognized as an Outstanding Researcher by the College of Engineering and Mineral Resources in 2001, 2002, and 2009; an Outstanding Advisor in 2005 and 2013; and an Outstanding Teacher in 2002, 2004, 2007, 2010, and 2013. I also received the West Virginia University Foundation Outstanding Teaching Award in 2013. In 2019, I received the IEEE Military Communications Conference (MILCOM) Award for Sustained Technical Achievement.

13. As a Fellow of the IEEE, I am actively involved with the Communications Society and its organization of several conferences and journals. For instance, I was the Technical Program Co-Chair for the *2021 IEEE International Conference on Communications (ICC)*, a flagship conference organized by the IEEE Communications Society which attracted nearly 2,000 technical paper submissions from top researchers around the world.



Declaration of Matthew C. Valenti, Ph.D., P.E.

U.S. Patent No. 7,421,032

14. I am not an attorney and offer no legal opinions, but in the course of my work, I have had experience studying and analyzing patents and patent claims from the perspective of a person skilled in the art.

### III. MATERIALS REVIEWED

15. The opinions contained in this declaration are based on the documents I reviewed, my professional judgment, as well as my education, experience, and knowledge regarding signal processing, communications, and coding.

16. In forming my opinions expressed in this declaration, I reviewed the following materials and information:

Ex. 1001	U.S. Patent No. 7,421,032
Ex. 1004	Prosecution History of U.S. Patent No. 7,421,032
Ex. 1005	U.S. Patent No. 6,029,264 to Kobayashi et al. ("Kobayashi")
Ex. 1006	McEliece <i>et al.</i> , "Turbo Decoding as an Instance of Pearl's 'Belief Propagation' Algorithm," <i>IEEE Journal On Selected Areas in Communication</i> , Vol. 16, No. 2 (February 1998). ("McEliece")
Ex. 1007	MacKay, "A Free Energy Minimization Framework for Inference Problems in Modulo 2 Arithmetic," <i>Fast Software Encryption</i> , B. Preneel, Ed. Berlin, Germany: Springer-Verlag Lecture Notes in Computer Science, Vol. 1008 (1995). ("MacKay")
Ex. 1009	Rorabaugh, <i>Error Coding Cookbook: Practical C/C++ Routines and Recipes for Error Detection and Correction</i> (1996). ("Rorabaugh")
Ex. 1010	Lin & Costello, <i>Error Control Coding: Fundamentals and Applications</i> (1983). ("Lin/Costello")
Ex. 1011	Cheng, "On the Construction of Efficient Multilevel Coded Modulations," <i>Proceedings 1997 IEEE International Symposium on Information Theory</i> (July 1997). ("Cheng I")
Ex. 1012	Cheng, "Iterative Decoding," Ph.D. dissertation, California Institute of Technology, Pasadena, CA (March 1997). ("Cheng II")

17. I also reviewed any other materials I refer to in this declaration in support of my opinions.

18. My opinions contained in this declaration are based on the documents I reviewed and my knowledge and professional judgment. My opinions have also been guided by my appreciation of how a person of ordinary skill in the art would have understood the state of the art, the prior art, and the claims and the specification of the '032 patent at the time of the alleged invention.

19. I have been asked to consider that the time of the alleged invention of the '032 patent is around May 18, 2000, which I understand is the filing date of Provisional Application No. 60/205,095, to which the '032 patent claims priority. (Ex. 1001, 1 (Related U.S. Application Data section).) My opinions reflect how one of ordinary skill in the art (which I describe below) would have understood the '032 patent, the prior art to the patent, and the state of the art at the time of the alleged invention as I was asked to consider, noted above.

20. Based on my experience and expertise, it is my opinion that the prior art discloses and/or suggests all the features recited in challenged claims 1-8 and 10-22 of the '032 patent, as I discuss in detail below.

#### **IV. PERSON OF ORDINARY SKILL IN THE ART AND THE TIME OF THE ALLEGED INVENTION**

21. Based on my knowledge and experience, I understand what a person of ordinary skill in the art would have known at the time of the alleged invention, which I discussed above as being around May 18, 2000. My opinions herein are, where appropriate, based on my understandings as to a person of ordinary skill in the art at that time. In my opinion, based on the materials and information I have reviewed, and based on my experience in the technical areas relevant to the '032 patent, a person of ordinary skill in the art at the time of the alleged invention of the '032 patent would have had a Ph.D. in mathematics, electrical or computer engineering, or computer science with an emphasis in signal processing, communications, or coding, or a master's degree in the above areas with at least three years of work experience in the field at the time of the alleged invention. Additional education would compensate for less experience, and vice versa. I apply this understanding in my analysis herein.

22. All of my opinions in this declaration are from the perspective of one of ordinary skill in the art, during the relevant timeframe (*e.g.*, the time of the alleged invention), which I discussed above as being around May 18, 2000. During this timeframe, I possessed at least the qualifications of a person of ordinary skill in the art, as defined above.

## V. TECHNICAL BACKGROUND

23. In this section, I discuss the state of the art with respect to certain technologies relevant to the subject matter of the '032 patent.

### A. Error Control Coding

24. An *error control code*, also called a *channel code*, adds controlled redundancy to a data transmission for the purpose of allowing the receiver of the transmission to detect and/or correct errors that will inevitably occur during transmission. (Ex. 1009, 41; Ex. 1010, 1-3.) *Linear block codes*, which are a common type of channel code and are relevant to the technology at issue in this proceeding, take blocks of  $k$  data bits and map them to codewords of length  $n$ , where  $n > k$ . (Ex. 1009, 41.) The difference between  $n$  and  $k$  is the amount of redundancy introduced by the code, *i.e.*, the number of additional/redundant bits that would need to be transmitted due to using the code. By using this redundancy, the decoder at the receiver will be able to detect that errors have occurred during the transmission, wherein the maximum number of detectable errors is related to the amount of redundancy. With enough redundancy, the decoder will be able to locate the errors, which will allow them to be corrected. In general, the more redundancy there is in the codeword, *i.e.*, the larger the value of  $(n - k)$ , the more bit errors the decoder will be able to detect and/or correct. (*Id.*, 42.)

25. The *rate*  $R$  of a code is the ratio of  $k$  to  $n$ ; *i.e.*,  $R = k/n$ . Since  $n > k$ , it follows that  $R < 1$ . (Ex. 1010, 4.) The rate is another way to describe how much redundancy is added by the code.

26. Codes can be *systematic* or *nonsystematic*. If the length- $n$  codeword can be partitioned into two parts with one part consisting of the  $k$  data bits and the other part consisting of the  $(n - k)$  redundant bits, then the code is said to be a *systematic* code. (Ex. 1009, 45; Ex. 1010, 54.) For systematic codes, the data bits are sometimes called the *systematic bits* and the redundant bits called the *parity bits*. Whether to place the parity bits before or after the systematic bits does not change the error detecting and correcting capabilities of the code and such ordering is at the discretion of the system designer. Indeed, different authors order the two parts of a systematic code differently. For instance, in Ex. 1009 (“*Rorabaugh*”), the systematic bits come before the parity bits, while in Ex. 1010 (“*Lin/Costello*”), the parity bits come before the systematic bits. (Ex. 1009, 45; Ex. 1010, 54.) If a code’s codewords cannot be so partitioned, then the code is said to be a *nonsystematic* code. (Ex. 1009, 45.)

27. The codewords of a linear block code may be generated through a matrix multiplication operation. (Ex. 1009, 44.) Let  $\mathbf{u}$  be a row-vector containing the  $k$  information bits (*i.e.*, the *message*) and  $\mathbf{v}$  be a row-vector containing the  $n$  code

bits (*i.e.*, the *codeword*). The codewords are related to the messages by a k-by-n matrix  $\mathbf{G}$  called the *generator matrix*. The format of  $\mathbf{u}$ ,  $\mathbf{v}$ , and  $\mathbf{G}$  is shown below:

$$\begin{aligned}\mathbf{u} &= [u_0 \ u_1 \ u_2 \ \dots \ u_{k-1}], \\ \mathbf{v} &= [v_0 \ v_1 \ v_2 \ \dots \ v_{n-1}], \\ \mathbf{G} &= \begin{bmatrix} g_{00} & g_{01} & g_{02} & \dots & g_{0,n-1} \\ g_{10} & g_{11} & g_{12} & \dots & g_{1,n-1} \\ \vdots & & & \ddots & \vdots \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \dots & g_{k-1,n-1} \end{bmatrix}\end{aligned}$$

*Id.* Notice in the above that indexing starts at 0 rather than 1. The codeword  $\mathbf{v}$  is found by multiplying the message  $\mathbf{u}$  by the generator matrix  $\mathbf{G}$  as follows:

$$\mathbf{v} = \mathbf{u} \cdot \mathbf{G}$$

where the ‘ $\cdot$ ’ symbol is here used to denote a matrix multiplication. *Id.*

## B. Matrix Multiplication and Modulo-2 Arithmetic

28. Matrix multiplications, such as the one used to encode a codeword from its corresponding message, are commonly used in the fields of coding, signal processing, and communications. When the operation involves the multiplication of a row vector by a matrix, then it is typically performed by taking the *inner product* of the vector with each of the columns of the matrix. For instance, turning back to our example of  $\mathbf{v} = \mathbf{u} \cdot \mathbf{G}$ , where  $\mathbf{u}$  is a length-k row vector (1-by-k matrix) and  $\mathbf{G}$  is a k-by-n matrix, then the  $i^{\text{th}}$  component of  $\mathbf{v}$  is found by taking the inner product of

$\mathbf{u}$  with the  $i^{\text{th}}$  column of  $\mathbf{G}$ . To understand how to compute the inner product, first note that  $\mathbf{u}$  and the  $i^{\text{th}}$  column of  $\mathbf{G}$  both have  $k$  elements. To perform the inner product, the same indexed elements of  $\mathbf{u}$  and the  $i^{\text{th}}$  column of  $\mathbf{G}$  are multiplied, and then these  $k$  products are added. This operation is best described by the following equation:

$$v_i = \sum_{j=0}^{k-1} u_j g_{ji}$$

(Ex. 1009, 45.) The above operation is performed for each of the  $n$  elements of  $\mathbf{v}$ .

29. Oftentimes, the linear block code is said to be *binary*. For binary codes, all the coefficients in the matrix multiplication operation are either 0 or 1. The message is binary, containing just 0's and 1's; the codeword is binary; and the  $\mathbf{G}$  matrix also contains only 0's and 1's. For such codes, the addition implied by the summation in the above equation is not ordinary addition, but rather, is *modulo-2* (mod-2) addition. Under the rules of modulo-2 addition,  $0+0=0$ ,  $1+0=1$ , and  $0+1=1$  as in ordinary addition; however, under modulo-2 addition,  $1+1 = 0$  unlike in ordinary addition. (*Id.*, 6-9.) Modulo-2 addition can be easily implemented in digital hardware by using an exclusive-or (XOR) gate. (*Id.*, 7.)

30. The concepts of matrix multiplication and modulo-2 arithmetic are best described by an example. Let's consider a case where  $\mathbf{u}$  and  $\mathbf{G}$  are as follows:



$$\mathbf{u} = [1 \quad 0 \quad 1]$$

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

31. The first element of  $\mathbf{v}$  would be found by taking the inner product of  $\mathbf{u}$  with the first column of  $\mathbf{G}$ . This involves an element-by-element multiplication, followed by a summation of the partial products:

$$v_0 = u_0g_{00} + u_1g_{10} + u_2g_{20} = (1 \times 1) + (0 \times 0) + (1 \times 0) = 1 + 0 + 0 = 1$$

32. The second element of  $\mathbf{v}$  is then found by taking the inner product of  $\mathbf{u}$  with the second column of  $\mathbf{G}$  as follows:

$$v_1 = u_0g_{01} + u_1g_{11} + u_2g_{21} = (1 \times 1) + (0 \times 0) + (1 \times 1) = 1 + 0 + 1 = 0$$

33. Note in the above that the rules of modulo-2 arithmetic had to be invoked:  $1+1 = 0$ .

34. The third and fourth elements of  $\mathbf{v}$  are found in similar fashion:

$$v_2 = u_0g_{02} + u_1g_{12} + u_2g_{22} = (1 \times 0) + (0 \times 1) + (1 \times 1) = 0 + 0 + 1 = 1$$

$$v_3 = u_0g_{03} + u_1g_{13} + u_2g_{23} = (1 \times 1) + (0 \times 1) + (1 \times 1) = 1 + 0 + 1 = 0$$

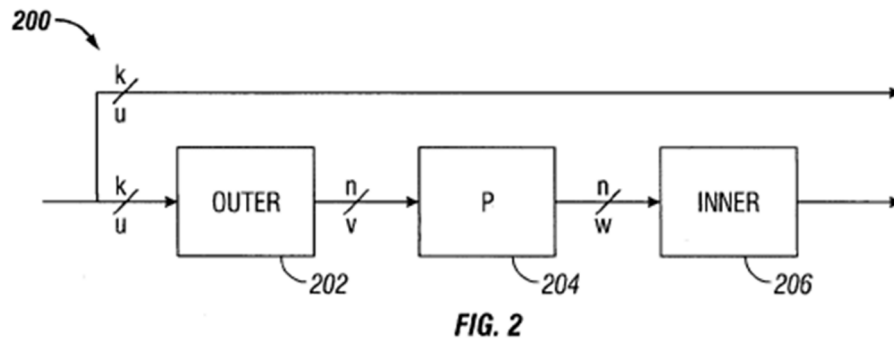
35. It thus follows that the codeword  $\mathbf{v}$  is:

$$\mathbf{v} = [1 \quad 0 \quad 1 \quad 0]$$

## VI. THE '032 PATENT

36. The '032 patent relates to “serial concatenation of interleaved convolutional codes forming turbo-like codes.” (Ex. 1001, Title.) For example, the '032 patent describes a “serial concatenated coder” that “includes an outer coder and an inner coder,” where the “outer coder irregularly repeats bits in a data block according to a degree profile and scrambles the repeated bits,” which are then “input to an inner coder, which has a rate substantially close to one.” (*Id.*, Abstract.)

37. The specification of the '032 patent discloses an exemplary embodiment of the alleged invention by way of Figure 2. (*Id.*, 2:35.)



(*Id.*, FIG. 2.) The '032 patent explains that “coder 200 may include an outer coder 202, an interleaver 204, and inner coder 206.” (*Id.*, 2:36-37.) “The outer coder 202 receives the uncoded data,” which “may be partitioned into blocks of fixed size, say  $k$  bits.” (*Id.*, 2:43-44.) The '032 patent explains that “[t]he outer coder may be an

( $n, k$ ) binary linear block coder, where  $n > k$ ,” such that “[t]he coder accepts as input a block  $u$  of  $k$  data bits and produces an output block  $v$  of  $n$  data bits.” (*Id.*, 2:44-47.) The outer coder irregularly “repeats the  $k$  bits in a block” such that “different bits in the block may be repeated a different number of times.” (*Id.*, 2:50-60.) The ’032 patent explains that “[t]he bits output from the outer coder 202 are scrambled” by interleaver 204 “before they are input to the inner coder 206.” (*Id.*, 3:25-29.) “The inner coder 206 may be a linear rate-1 coder,” specifically “an accumulator, which produces outputs that are the modulo two (mod-2) partial sums of its inputs.” (*Id.*, 2:61-3:1.) Together, “[t]he serial concatenation of the interleaved irregular repeat code and the accumulate code produces an irregular repeat and accumulate (IRA) code.” (*Id.*, 3:30-32.)

38. The ’032 patent further explains that “[a]n IRA code . . . may be represented as a set of parity checks,” which in turn “may be represented in a bipartite graph, called the Tanner graph, of the code.” (*Id.*, 3:32-35.) The specification of the ’032 patent additionally discloses that “[b]elief propagation’ on the Tanner Graph realization may be used to decode IRA codes,” where “the belief propagation decoding technique allows the messages passed on an edge to represent posterior densities on the bit associated with the variable node.” (*Id.*, 5:20-24.)

39. However, as I discuss in more detail below, all of the features recited in the challenged claims were already known and disclosed in the prior art. (*See my discussions below in Section IX.*)

## **VII. CLAIM CONSTRUCTION**

40. I understand that claim terms are typically given their ordinary and customary meanings, as would have been understood by a person of ordinary skill in the art at the time of the alleged invention, which as I explained above I have been asked to assume is around May 18, 2000. In considering the meaning of the claims, however, I understand that one must consider the language of the claims, the specification, and the prosecution history of record. I have been asked to consider the claim terms under their plain meanings and thus I have considered the claims, specification and prosecution history for the '032 patent in doing so in support of my opinions concerning the '032 patent and the prior art discussed herein. (Ex. 1001; Ex. 1004.)

41. I have also been asked to consider the following meaning of the claim term “repeat” in claims 1(c), 5(b), 7(b), 13(b)-(c), and 17 of the '032 patent: “generation of additional bits, where generation can include, for example, duplication or reuse of bits.” I have been informed that this is a construction that a court has affirmed in another litigation involving the same patent. I have also been informed that the court found in the other litigation that passing an input information bit through an AND gate when the other input is a “1” bit comprises “repeating” the information bit. I have thus considered this construction in my analysis below.

## VIII. OVERVIEW OF THE PRIOR ART

### A. *Kobayashi* (Ex. 1005)

42. U.S. Patent No. 6,029,264 (“*Kobayashi*”) is titled “System and Method for Error Correcting a Received Data Stream in a Concatenated System,” and my understanding is that *Kobayashi* issued February 22, 2000, from U.S. Patent Application No. 08/840,383, which was filed on April 28, 1997. (Ex. 1005, Cover.) *Kobayashi* discloses a concatenated encoding and decoding system with transmitter and receiver portions, shown at the top and bottom, respectively, of Figure 8 below.

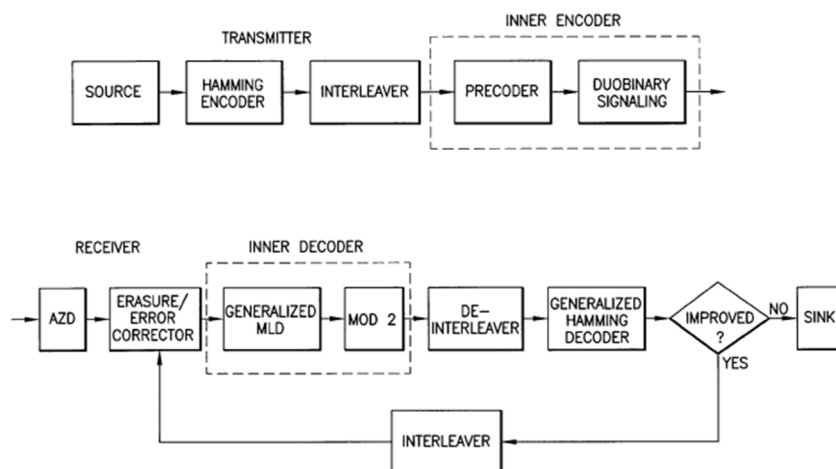


FIG.8

(*Id.*, FIG. 8.)

43. *Kobayashi*’s transmitter is for “a simple packet transmission system in which there are 28 information bits in a packet, an example of which is given by the stream:  $I_1=(0001001000110100010101100000)$ .” (*Id.*, 7:46-49.)

44. After obtaining the 28 information bits for transmission, *Kobayashi's* transmitter then performs several encoding steps. “Rather than encoding the entire packet at once, it is first segmented into blocks of  $k=4$  bits, and each block is then encoded to a codeword of length  $n=7$ , by using a  $(7, 4)$  Hamming code,” which has the following systematic parity-check and generator matrices:

$$\begin{array}{c} \text{and} \\ \mathbf{H} = \begin{array}{ccccccc} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array} \\ \\ \mathbf{G} = \begin{array}{ccccccc} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{array} \end{array}$$

(*Id.*, 7:50-65.) The Hamming code output is the following sequence:  $I_2=(0001101, 0010111, 0011010, 0100011, 0101110, 0110100, 0000000)$ . (*Id.*, 7:66-8:2.)

45. Next, the *Kobayashi* transmitter “perform[s] a permutation action,” using “a  $7 \times 7$  block interleaver . . . [to] store the above 49 bits row-wise in the following array structure.”

$$\pi = \begin{array}{ccccccc} 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

(*Id.*, 8:3-15.) “[T]he permutation output is obtained by reading out the above array, column by column as follows:  $I_3=(0000000, 0001110, 0110010, 1010100, 1100110, 0111100, 1101000)$ .” (*Id.*, 8:16-20.)

46. The final encoding step is performed by a precoder. “The precoder output is obtained by taking the modulo-2 sum of the current input and the previous output (where ‘modulo-2 summation’ can be implemented by Exclusive OR:  $0+0=0$ ,  $0+1=1$ ,  $1+0=1$ ,  $1+1=0$ ).” (*Id.*, 8:21-24). The resulting encoded sequence from the precoder is “ $I_4=(0000000, 0001011, 1011100, 1100111, 0111011, 1010111, 011000[0])$ .” (*Id.*, 8:25-27.)<sup>2</sup>

47. While duobinary signaling is depicted in Figure 8 as part of the “inner encoder,” *Kobayashi* discloses that duobinary signaling is simply a transmission technique for *Kobayashi*’s system. (*Id.*, 7:30-31 (“The precoder introduces a simple transformation prior to the transmission by duobinary signaling.”), 7:43-45

---

<sup>2</sup> In my opinion, a person of ordinary skill in the art would have understood based on the overall disclosure of *Kobayashi* and the context in the relevant portions that the  $I_4$  sequence contains a typographical error and is missing the 49<sup>th</sup> bit. A person of ordinary skill in the art would have understood this missing 49<sup>th</sup> bit to be a “0” bit.



(“Duobinary signaling illustrated in this example is a simplest case of partial-response channel coding referred to in the Background of the Art.”); *see also id.*, 2:21-25 (“Partial-response channel coding is well recognized as a bandwidth-efficient transmission technique . . .”).)

48. *Kobayashi*’s receiver receives a duobinary sequence, passes it through an ambiguity zone detector (AZD), and begins an iterative decoding method that “attempts to resolve as many erasures/errors as possible” on each iteration. (*Id.*, 8:33-67.) The decoding method applies the following components on each iteration: an inner decoder composed of a generalized maximum likelihood decoder and mod-2 decoder; an inverse permutation (7x7 de-interleaver); and a generalized Hamming decoder. (*Id.*, 8:67-9:67.) A permutation action is performed at the end of each iteration; the output is compared to the AZD output; and the comparison is used to correct errors/erasures in the original AZD output sequence. (*Id.*, 10:1-36.) Additional iterations are performed until “the iterative decoder has successfully recovered the original information sequence of length 28 bits.” (*Id.*, 10:37-11:17.)

**B. *McEliece* (Ex. 1006)**

49. I understand that the article titled “Turbo Decoding as an Instance of Pearl’s ‘Belief Propagation’ Algorithm” (“*McEliece*”) is an article published in the *IEEE Journal on Selected Areas in Communications* in February 1998, which is a publication of the IEEE Communications Society. (Ex. 1006, Cover.)

50. *McEliece* discloses a decoding algorithm called “Pearl’s belief propagation algorithm,” which is an algorithm that “solves the probabilistic inference problem in many important special cases.” (*Id.*, 144.) *McEliece* describes the probabilistic inference problem as follows:

[L]et  $\mathbf{X} = \{X_1, X_2, \dots, X_N\}$ <sup>5</sup> be a set of  $N$  discrete variables, where  $X_i$  assumes values in the finite alphabet  $A_i$ . The joint density function

$$p(\mathbf{x}) = p(x_1, x_2, \dots, x_N) \\ \stackrel{\text{def}}{=} \Pr\{X_1 = x_1, \dots, X_N = x_N\}$$

is then a mapping from  $A_1 \times \dots \times A_N$  into the set of real numbers  $R$ . We assume that the marginal densities  $p(x_i) \stackrel{\text{def}}{=} \Pr\{X_i = x_i\}$  are also known. The marginal density function  $p(x_i)$  represents our *a priori* “belief” about the random variable  $X_i$ . Now, suppose that one or more of these random variables is measured or “observed.” This means that there is a subset  $J \subseteq \{1, 2, \dots, N\}$  (the evidence set) such that, for all  $j \in J$ , the random variable  $X_j$  is

known to have a particular value, say  $a_j$ . The *evidence* is then defined to be the event

$$\mathcal{E} = \{X_j = a_j : j \in J\}.$$

The fundamental *probabilistic inference problem* is to compute the *updated beliefs*, i.e., the *a posteriori* or conditional probabilities  $p(X_i|\mathcal{E})$ , for all  $i \notin J$ .

(*Id.* (emphasis in original).)

51. *McEliece* discloses that a “‘Bayesian belief network’ approach . . . to this inference problem is to exploit any ‘partial independencies’ which may exist among the  $X_i$ ’s to simplify belief updating.” (*Id.*) These “‘partial independencies can be described by a *directed acyclic graph*, or DAG,” which is “a finite, directed graph, in which there are no directed cycles.” (*Id.* (emphasis in original).) Suppose “ $G$  is a DAG, and . . .  $X$  is a set of random variables in one-to-one correspondence with the vertices of  $G$ .” (*Id.* (emphasis in original).) Additionally, suppose that if  $G$  has a “‘directed edge  $a \rightarrow b$ , then  $a$  will be called a ‘parent’ of  $b$ , and  $b$  will be called a ‘child’ of  $a$ .” (*Id.*) Then a “DAG, together with the associated random variables  $X$ , is called a *Bayesian belief network*, or *Bayesian network* for short.” (*Id.*, 145 (emphasis in original).)

52. *McEliece* discloses that “Bayesian networks can sometimes lead to considerable simplifications of the probabilistic inference problem,” such as “Pearl’s

*belief propagation* algorithm.” (*Id.* (emphasis in original).) *McEliece* further discloses that “Pearl’s belief propagation algorithm is a decentralized ‘message-passing’ algorithm, in which there is a processor associated with each vertex of  $G$ ” and “[e]ach processor can communicate only with its parents and children.” (*Id.*, 146.) The general principle underlying Pearl’s belief propagation algorithm is that “[w]hen a processor is activated, it ‘reads’ the messages received from each of its parents and children, updates its belief based on these messages, and then sends new messages back to its parents and children.” (*Id.*)

53. *McEliece* discloses advantageous uses of Pearl’s belief propagation algorithm, which can be “used to derive effective iterative decoding algorithms for a number of . . . error-control systems, including . . . serially concatenated codes.” (*Id.*, 140.) *McEliece* describes these belief propagation (“BP”) decoding algorithms as “remarkably effective” and notes that “belief propagation provides an attractive general method for devising low-complexity iterative decoding algorithms for hybrid coded systems.” (*Id.*) In particular, *McEliece* discloses that “decoding algorithms [for serially concatenated codes] can be derived routinely from a BP viewpoint, using the network of Fig. 12.” (*Id.*, 149.)

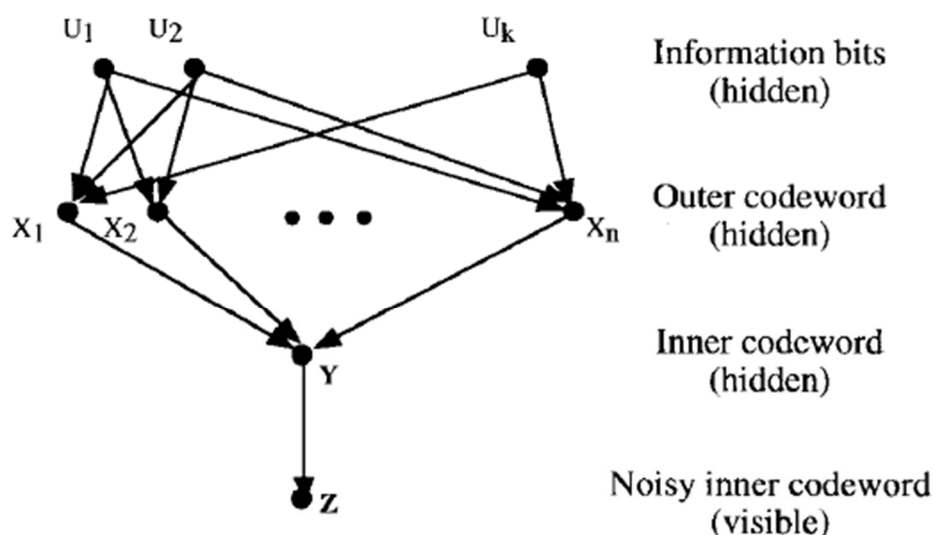


Fig. 12. Belief network for decoding a pair of serially concatenated codes.

(*Id.*, 150 (FIG. 12).)

54. *McEliece* further discloses that Pearl's belief propagation algorithm can be "used to derive effective iterative decoding algorithms for . . . the recently introduced low-density generator matrix codes." (*Id.*, 140.) In particular, *McEliece* discloses using "BP decoding on certain systematic linear block codes with low-density *generator matrices*." (*Id.*, 149 (emphasis in original).) *McEliece* discusses the benefits of using low-density generator matrix codes along with a belief propagation decoder, noting that "[t]he decoding algorithm[']s . . . results were quite good, especially at high rates." (Ex. 1006, 149.) Moreover, "these same ideas [were used] to construct a class of block codes which yield some remarkably efficient multilevel coded modulations." (*Id.*)

55. Figure 11 below “shows the belief network for low-density generator matrix codes.” (*Id.*)

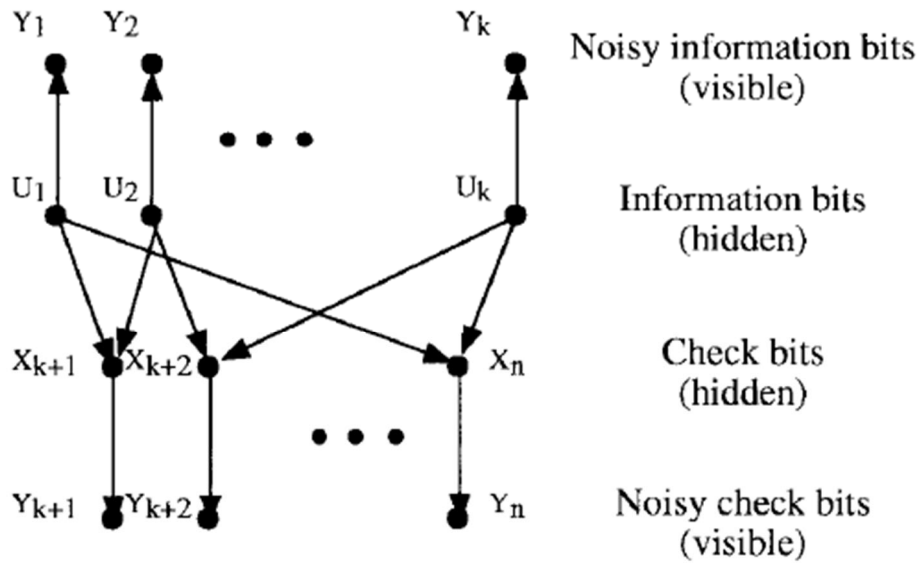


Fig. 11. Belief network for decoding systematic, low-density generator matrix codes.

(*Id.*, 149 (FIG. 11).)

**IX. THE PRIOR ART DISCLOSES AND/OR SUGGESTS THE RECITED FEATURES OF CLAIMS 1-8 AND 10-22 OF THE '032 PATENT**

**A. *Kobayashi* Discloses the Features of Claims 1, 3-5, 7-8, 11-12, and 14-16**

**1. Claim 1**

**a) A method comprising:**

56. I understand that “[a] method comprising:” is the preamble of claim 1. I have been asked to assume that the preamble is limiting. In my opinion, *Kobayashi* discloses the limitations of the preamble. For example, *Kobayashi* discloses a concatenated system with both transmitter and receiver portions, wherein the transmitter side receives message bits from a source and uses several encoders that perform an encoding “method” as claimed. (Ex. 1005, FIG. 8, 5:25-27, 7:5-8:34 (describing the method in the context of Figure 8); see Sections IX.A.1(b)-(d); see also Section V.A (background on error control coding).)

57. In particular, *Kobayashi* discloses that the method (as shown in Figure 8’s transmitter below) comprises receiving message bits from a source via a packet transmission system; encoding the sequence of message bits using the Hamming encoder, interleaver, and precoder components to generate a sequence of parity bits; and making the encoded sequence available for transmission to the receiver via duobinary signaling. (Ex. 1005, 7:5-8:34.)

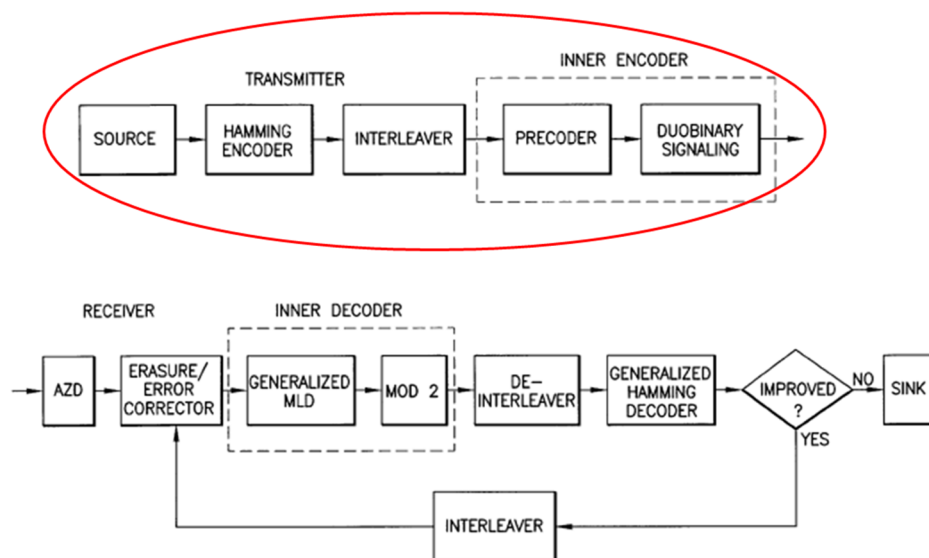


FIG.8

(Ex. 1005, FIG. 8 (annotated); *see also id.*, 5:17-24.)

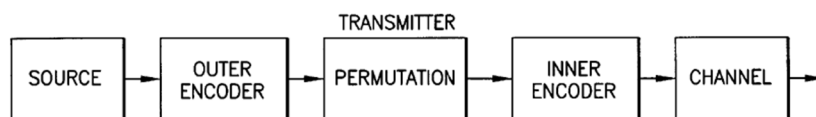


FIG.7A

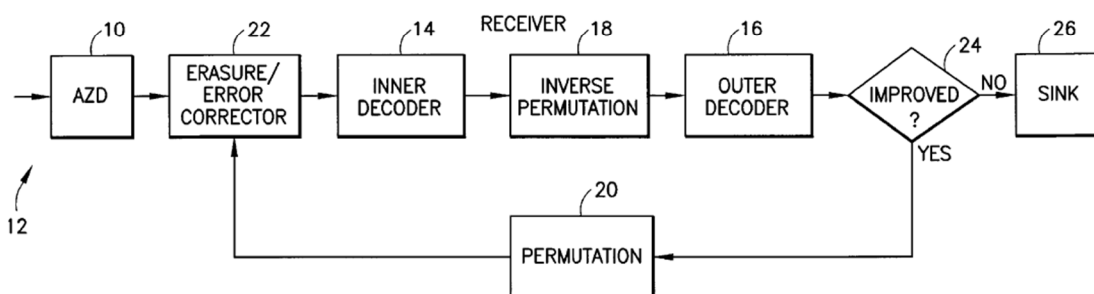


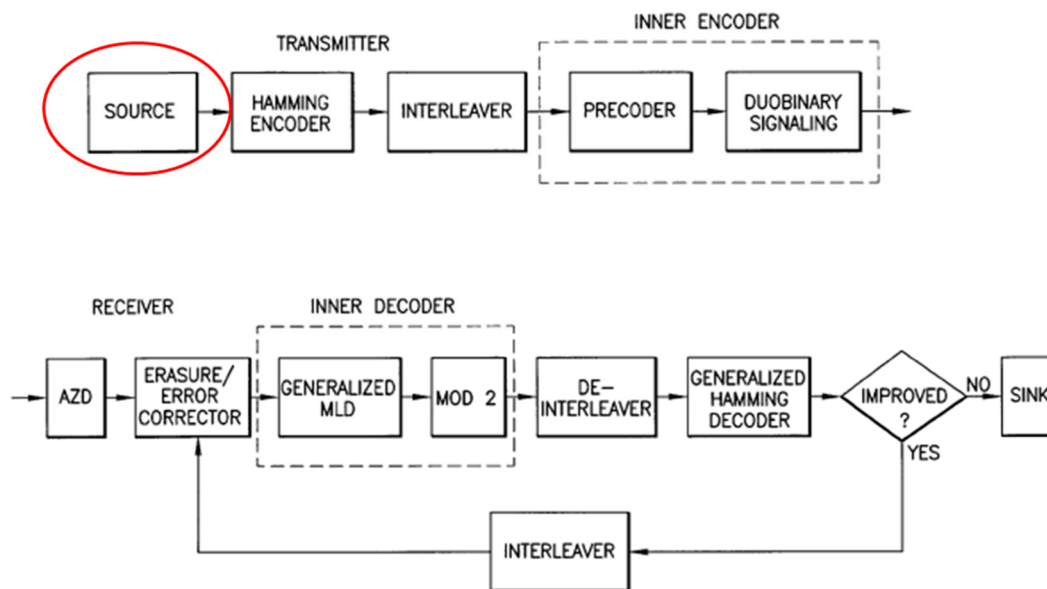
FIG.7B

(*Id.*, FIGS. 7A, 7B (showing generalized versions of the transmitter and receiver sides of Figure 8).)



**b) receiving a collection of message bits having a first sequence in a source data stream;**

58. In my opinion, *Kobayashi* discloses these limitations. For example, *Kobayashi* discloses that the concatenated system's transmitter receives data via "a simple packet transmission system in which there are **28 information bits in a packet**, an example of which is given by **the stream:  $I_1=(0001001000110100010101100000)$** ." (Ex. 1005, 7:46-49 (emphasis added).) *Kobayashi* further depicts the source of the data stream in Figure 8 below.



**FIG.8**

(Ex. 1005, FIG. 8 (annotated).) Thus, in my opinion, *Kobayashi* discloses "receiving a collection of message bits having a first sequence in a source data stream" because it discloses receiving the 28-message-bit sequence  $I_1$  from a source data stream.

- c) **generating a sequence of parity bits, wherein each parity bit “ $x_j$ ” in the sequence is in accordance with the formula  $x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}$  where “ $x_{j-1}$ ” is the value of a parity bit “j-1,” and “ $\sum_{i=1}^a v_{(j-1)a+i}$ ” is the value of a sum of “a” randomly chosen irregular repeats of the message bits; and<sup>3</sup>**

59. In my opinion, *Kobayashi* discloses these limitations. *Kobayashi* discloses that the received first sequence of message bits  $I_1$  is encoded in several steps, as shown below in Figure 8.

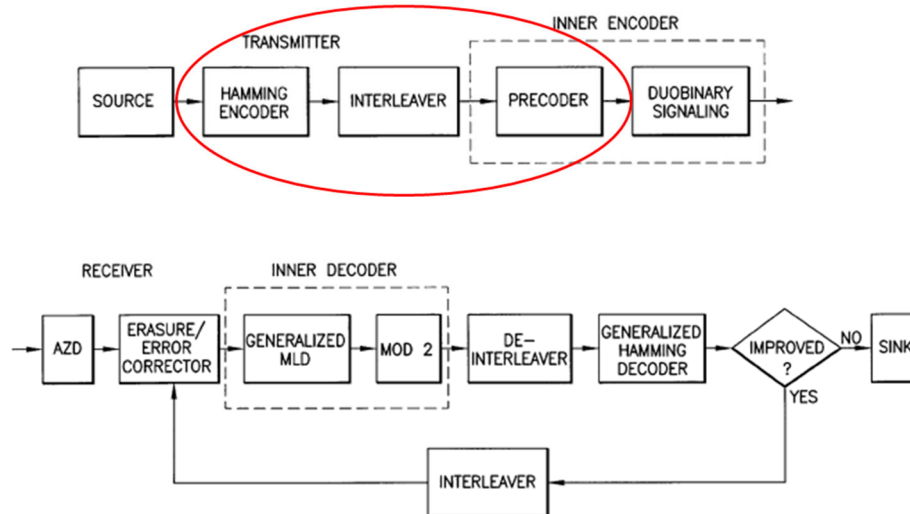


FIG.8

<sup>3</sup> I understand that this claim language reflects updates made in the Certificate of Correction appended to the '032 patent. (See Ex. 1001.) My analysis in this declaration reflects this updated language.

(Ex. 1005, FIG. 8 (annotated).)<sup>4</sup>

60. *Kobayashi* discloses that the 28-bit sequence  $I_1$  is first segmented into seven sub-blocks, and a Hamming code is applied to each sub-block.

Consider a simple packet transmission system in which there are 28 information bits in a packet, an example of which is given by the stream:

$$I_1 = (0001001000110100010101100000)$$

**Rather than encoding the entire packet at once, it is first segmented into blocks of  $k=4$  bits, and each block is then encoded to a codeword of length  $n=7$ , by using a (7, 4) Hamming code. Its parity-check and generator matrices are given in systematic form by:**

		1	0	1	1	1	0	0
H =		1	1	1	0	0	1	0
		0	1	1	1	0	0	1
and								
G =		1	0	0	0	1	1	0
		0	1	0	0	0	1	1
		0	0	1	0	1	1	1
		0	0	0	1	1	0	1

---

<sup>4</sup> As I have discussed further for claim 1(d), *see* Section IX.A.1(d), while duobinary signaling is depicted in Figure 8 as part of the “inner encoder,” *Kobayashi* discloses that duobinary signaling is simply a transmission technique for transmitting the sequence of parity bits to the decoder.

Then **the Hamming encoder output is the following 49 bits** (commas are placed between code words for clarity):  
 **$I_2=(0001101, 0010111, 0011010, 0100011, 0101110, 0110100, 0000000)$**

(Ex. 1005, 7:46-8:2 (emphasis added); *see also* Sections V.A (background on error control coding), V.B (background on matrix multiplication).)

61. *Kobayashi*'s method then uses a "7x7 block interleaver" to "perform a permutation action . . . which will store the above 49 bits [of  $I_2$ ] row-wise in the following array structure."

	0	0	0	1	1	0	1
	0	0	1	0	1	1	1
	0	0	1	1	0	1	0
$\pi =$	0	1	0	0	0	1	1
	0	1	0	1	1	1	0
	0	1	1	0	1	0	0
	0	0	0	0	0	0	0

(*Id.*, 8:3-15.) *Kobayashi* discloses that the "permutation output is obtained by "reading out the above array column by column as follows:  $I_3=(0000000, 0001110, 0110010, 1010100, 1100110, 0111100, 1101000)$ ." (*Id.*, 8:16-20.)

62. *Kobayashi* discloses that the sequence  $I_3$  is the input to the precoder, which performs the following operations. (*Id.*, 8:18-27.)

The precoder output is obtained by taking the modulo-2 sum of the current input and the previous output (where “modulo-2 summation” can be implemented by Exclusive OR:  $0+0=0$ ,  $0+1=1$ ,  $1+0=1$ ,  $1+1=0$ ).

(*Id.*, 8:21-24.)

The precoder introduces a simple transformation prior to the transmission by duobinary signaling. Its purpose is to prevent a possible error propagation in the decoded output. The precoder maps the input binary sequence into another binary sequence, based on the following rule: when the current input is 0, the output should remain in the previous value; and when the input is 1, the output changes its value from the previous one, i.e. either 0 to 1 or from 1 to 0.

(*Id.*, 7:30-37; *see also* Section V.B (background on modulo-2 arithmetic).) The resulting encoded sequence from the precoder is “ $I_4=(0000000, 0001011, 1011100, 1100111, 0111011, 1010111, 011000[0])$ .” (Ex. 1005, 8:25-27.)<sup>5</sup>

---

<sup>5</sup> In my opinion, a person of ordinary skill in the art would have understood based on the overall disclosure of *Kobayashi* and the context in the relevant portions that the  $I_4$  sequence contains a typographical error and is missing the 49<sup>th</sup> bit. A person

63. In my opinion this generated sequence  $I_4$  is a “sequence of parity bits” because the precoder can be represented as a (49, 49) nonsystematic linear block code. (See Section V.A (background on error control coding and nonsystematic linear block codes).) In particular, the precoder operation is equivalent to multiplying the 1x49 vector  $I_3$  by a 49x49 generator matrix  $\mathbf{G}_A$  with “1”s both along and above the main diagonal and “0”s below the main diagonal, such as shown below, and obtaining 1x49 vector  $I_4$  (*i.e.*, the sequence of parity bits) as a result. (See Ex. 1009, 44 (“the encoding operation [of a linear block code] is represented mathematically as  $\mathbf{v} = \mathbf{u} \cdot \mathbf{G}$ , where  $\mathbf{v}$  is a vector of the encoded data bits,  $\mathbf{u}$  is a vector of  $k$  information bits, and  $\mathbf{G}$  is the *generator matrix*” (emphasis in original)), 45 (describing a nonsystematic code as one that does not contain a  $k \times k$  identity matrix); see also Sections V.A (background on error control coding and linear block codes), V.B (background on matrix multiplication).)

$$\mathbf{G}_A = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

---

of ordinary skill in the art would have understood this missing 49<sup>th</sup> bit to be a “0” bit.

For such a nonsystematic linear block code, the bits of the codeword are parity bits. (See Ex. 1009, 45 (describing a nonsystematic code as having parity digits, and its corresponding generator matrix as being a matrix of parity-check coefficients); see also Section V.A (background on error control coding and nonsystematic codes).)

64. It is also my opinion that “each parity bit ‘ $x_j$ ’ in the sequence”  $I_4$  is represented by “the formula  $x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}$  where ‘ $x_{j-1}$ ’ is the value of a parity bit ‘ $j-1$ ,’ and ‘ $\sum_{i=1}^a v_{(j-1)a+i}$ ’ is the value of a sum of ‘ $a$ ’ randomly chosen irregular repeats of the message bits.” As I described above, in the precoder step, each parity bit of  $I_4$  is generated by taking the mod-2 sum (which is implemented using an exclusive-OR ‘XOR’ operation) of the previous parity bit of  $I_4$  and the current bit of  $I_3$ . (See Section V.B (background on modulo-2 arithmetic).) That is, the  $j$ th bit of  $I_4$  (“ $x_j$ ”) equals the mod-2 sum of the “ $j-1$ ”th bit of  $I_4$  and the  $j$ th bit of  $I_3$ :  $x_j = x_{j-1} + [jth\ bit\ of\ I_3]$ .

65. In my opinion, the  $j$ th bit of  $I_3$  is equal to  $\sum_{i=1}^a v_{(j-1)a+i}$ , i.e., “a sum of ‘ $a$ ’ randomly chosen irregular repeats of the message bits.” As I described above, in the first encoding step, each 4-bit block of  $I_1$  is multiplied by generator matrix  $\mathbf{G}$ , and this process comprises irregular repetition of the message bits under the claim construction of “repeat” that I have been asked to consider. (See Section VII.) As I previously discussed in Section VII, I have been informed that a court in another

litigation involving the same patent found that passing an input message bit through an AND gate when the other input is a “1” bit comprises “repeating” the message bit. (Section VII.) Multiplying a binary message bit by a “1” bit is equivalent to passing the message bit through an AND gate with a “1” bit, and thus in my opinion, under this construction of “repeat,” multiplying an message bit by a “1” bit comprises “repeating” the message bit. (*See, e.g.*, Ex. 1009, 7-8 (disclosing that binary/modulo-2 multiplication is equivalent to a bitwise AND operation).) It is my opinion that under this construction, any type of linear code using a non-zero generator matrix will “repeat” input bits because the process of multiplying a vector of message bits by the generator matrix will necessarily involve multiplying input bits by “1” bits.

66. For example, as I described above, *Kobayashi*’s Hamming encoder multiplies each 4-bit block of  $I_1$  by the  $4 \times 7$  generator matrix  $G$ , resulting in seven 7-bit codewords. In my opinion, this encoding step performs *repetition* of each and every message bit because the process of multiplying each 4-bit block by generator matrix  $G$  involves multiplying each input bits by at least one “1” bit (*i.e.*, repeating the input bits) and then summing the repeated bits to generate the codeword.

67. It is also my opinion that the message bits are repeated irregularly such that message bits are repeated a different number of times. The example below



shows the first 4-bit block of sequence  $I_1$  being multiplied by 4x7 generator matrix  $G$ , and I have used  $c_1$  through  $c_4$  to represent the first four message bits of  $I_1$ . (See Sections V.A (background on error control coding and linear block codes), V.B (background on matrix multiplication).) As I have shown, the first, second, and fourth bits of the block are “repeated” (multiplied by a “1” bit) three times because the first, second, and fourth rows of generator matrix  $G$  each have three “1”s (in blue). However, the third bit of the sub-block is repeated *four* times because the third row of generator matrix  $G$  has four “1”s (in yellow). The 7-bit codeword is comprised of seven sums of “a” irregular repeats of the message bits, where  $a=1$  (first four bits)<sup>6</sup> and  $a=3$  (last three bits).

$$[c_1 \quad c_2 \quad c_3 \quad c_4] \begin{bmatrix} \text{1} & 0 & 0 & 0 & \text{1} & \text{1} & 0 \\ 0 & \text{1} & 0 & 0 & 0 & \text{1} & \text{1} \\ 0 & 0 & \text{1} & 0 & \text{1} & \text{1} & \text{1} \\ 0 & 0 & 0 & \text{1} & \text{1} & 0 & \text{1} \end{bmatrix} =$$

---

<sup>6</sup> The specification admits “a” can be 1. Ex. 1005 at 4:42-45 (“IRA codes with  $a=1$ ”).

$$\begin{bmatrix} c_1(\text{1}) + c_2(0) + c_3(0) + c_4(0) \\ c_1(0) + c_2(\text{1}) + c_3(0) + c_4(0) \\ c_1(0) + c_2(0) + c_3(\text{1}) + c_4(0) \\ c_1(0) + c_2(0) + c_3(0) + c_4(\text{1}) \\ c_1(\text{1}) + c_2(0) + c_3(\text{1}) + c_4(\text{1}) \\ c_1(\text{1}) + c_2(\text{1}) + c_3(\text{1}) + c_4(0) \\ c_1(0) + c_2(\text{1}) + c_3(\text{1}) + c_4(\text{1}) \end{bmatrix}^T = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_1 + c_3 + c_4 \\ c_1 + c_2 + c_3 \\ c_2 + c_3 + c_4 \end{bmatrix}^T$$

68. When the full 28-bit sequence  $I_1$  is encoded via the Hamming encoder, the 3<sup>rd</sup>, 7<sup>th</sup>, 11<sup>th</sup>, 15<sup>th</sup>, 19<sup>th</sup>, 23<sup>rd</sup>, and 27<sup>th</sup> bits are repeated four times, while the other 21 bits are repeated three times. Each of the 49 bits of sequence  $I_2$  is thus “a sum of ‘a’ . . . irregular repeats of the message bits,” where  $a=1$  (no highlighting)<sup>7</sup> or  $a=3$  (green highlighting), as shown below.

$$I_1 = (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{16}, c_{17}, c_{18}, c_{19}, c_{20}, c_{21}, c_{22}, c_{23}, c_{24}, c_{25}, c_{26}, c_{27}, c_{28})$$

$$I_2 = (c_1, c_2, c_3, c_4, c_1 + c_3 + c_4, c_1 + c_2 + c_3, c_2 + c_3 + c_4, c_5, c_6, c_7, c_8, c_5 + c_7 + c_8, c_5 + c_6 + c_7, c_6 + c_7 + c_8, c_9, c_{10}, c_{11}, c_{12}, c_9 + c_{11} + c_{12}, c_9 + c_{10} + c_{11}, c_{10} + c_{11} + c_{12}, c_{13}, c_{14}, c_{15}, c_{16}, c_{13} + c_{15} + c_{16}, c_{13} + c_{14} + c_{15}, c_{14} + c_{15} + c_{16}, c_{17}, c_{18}, c_{19}, c_{20}, c_{17} + c_{19} + c_{20}, c_{17} + c_{18} + c_{19}, c_{18} + c_{19} + c_{20}, c_{21}, c_{22}, c_{23}, c_{24}, c_{21} + c_{23} + c_{24}, c_{21} + c_{22} + c_{23}, c_{22} + c_{23} + c_{24}, c_{25}, c_{26}, c_{27}, c_{28}, c_{25} + c_{27} + c_{28}, c_{25} + c_{26} + c_{27}, c_{26} + c_{27} + c_{28})$$

---

<sup>7</sup> The specification states “a” can be 1. (Ex. 1001 at 4:42-45 (“ . . . IRA codes with  $a=1$  . . .”).)

69. The interleaver  $\pi$  permutes the sequence  $I_2$  to result in sequence  $I_3$ , as shown below. The interleaver does not otherwise alter the bits of  $I_2$ , and thus each bit of  $I_3$  is still “a sum of ‘a’ . . . irregular repeats of the message bits,” where  $a=1$  (no highlighting) or  $a=3$  (green highlighting), as shown below.

$$\pi = \begin{bmatrix} c_1 & c_2 & c_3 & c_4 & c_1 + c_3 + c_4 & c_1 + c_2 + c_3 & c_2 + c_3 + c_4 \\ c_5 & c_6 & c_7 & c_8 & c_5 + c_7 + c_8 & c_5 + c_6 + c_7 & c_6 + c_7 + c_8 \\ c_9 & c_{10} & c_{11} & c_{12} & c_9 + c_{11} + c_{12} & c_9 + c_{10} + c_{11} & c_{10} + c_{11} + c_{12} \\ c_{13} & c_{14} & c_{15} & c_{16} & c_{13} + c_{15} + c_{16} & c_{13} + c_{14} + c_{15} & c_{14} + c_{15} + c_{16} \\ c_{17} & c_{18} & c_{19} & c_{20} & c_{17} + c_{19} + c_{20} & c_{17} + c_{18} + c_{19} & c_{18} + c_{19} + c_{20} \\ c_{21} & c_{22} & c_{23} & c_{24} & c_{21} + c_{23} + c_{24} & c_{21} + c_{22} + c_{23} & c_{22} + c_{23} + c_{24} \\ c_{25} & c_{26} & c_{27} & c_{28} & c_{25} + c_{27} + c_{28} & c_{25} + c_{26} + c_{27} & c_{26} + c_{27} + c_{28} \end{bmatrix}$$

$$I_3 = (c_1, c_5, c_9, c_{13}, c_{17}, c_{21}, c_{25}, c_2, c_6, c_{10}, c_{14}, c_{18}, c_{22}, c_{26}, c_3, c_7, c_{11}, c_{15}, c_{19}, c_{23}, c_{27}, c_4, c_8, c_{12}, c_{16}, c_{20}, c_{24}, c_{28}, c_1 + c_3 + c_4, c_5 + c_7 + c_8, c_9 + c_{11} + c_{12}, c_{13} + c_{15} + c_{16}, c_{17} + c_{19} + c_{20}, c_{21} + c_{23} + c_{24}, c_{25} + c_{27} + c_{28}, c_1 + c_2 + c_3, c_5 + c_6 + c_7, c_9 + c_{10} + c_{11}, c_{13} + c_{14} + c_{15}, c_{17} + c_{18} + c_{19}, c_{21} + c_{22} + c_{23}, c_{25} + c_{26} + c_{27}, c_2 + c_3 + c_4, c_6 + c_7 + c_8, c_{10} + c_{11} + c_{12}, c_{14} + c_{15} + c_{16}, c_{18} + c_{19} + c_{20}, c_{22} + c_{23} + c_{24}, c_{26} + c_{27} + c_{28})$$

70. In my opinion, each bit in sequence  $I_3$  is also a sum of  $a=1$  or  $a=3$  “randomly chosen irregular repeats of the message bits.” A person of ordinary skill in the art would have understood that while claim 1(c) uses the term “randomly chosen irregular repeats,” the pattern of irregular repeats would need to be known to both the encoder and decoder in order for the code to be used in an actual communication system. If the encoder were to choose repeats on a truly random basis, the decoder would not know how the bits were repeated and would be unable to achieve its goal of decoding the encoded data stream. A person of ordinary skill

in the art would have therefore understood “randomly chosen irregular repeats” to encompass irregularly repeated bits chosen according to a pseudo-random or arbitrary pattern, and would have recognized that an interleaver distributes bits in a pseudo-random or arbitrary sequence. (See Ex. 1001, 3:25-29 ('032 patent similarly disclosing interleaver 204 as performing a “pseudo-random permutation” of the irregularly repeated message bits).) When discussing Figure 3, the specification describes the “permutation block 310,” which is labeled “random permutation” in Fig. 3,” as having connections indicated by “arbitrary permutation of the  $r_a$  edges joining information nodes 302 and check nodes 304. (Ex. 1001, 3:52-55, Fig. 3 (“random permutation”).) The specification states these random permutations correspond to the scrambling performed by the interleaver 204. *Id.* (“These connections correspond to the scrambling performed by the interleaver 204.”).) It is my opinion that each bit in sequence  $I_3$  is therefore “a sum of ‘a’ randomly chosen irregular repeats of the message bits” because the message bits are irregularly repeated and permuted according to an arbitrary pattern, *i.e.*, the pattern implied by the combination of *Kobayashi*’s Hamming code and interleaver.

71. Thus, in my opinion, every bit of  $I_3$  is equal to  $\sum_{i=1}^a v_{(j-1)a+i}$ , *i.e.*, “the value of a sum of ‘a’ randomly chosen irregular repeats of the message bits.” When the full “sequence of parity bits”  $I_4$  is generated by using the  $I_3$  sequence as input to

the precoder operation, “each parity bit ‘ $x_j$ ’ in the sequence is in accordance with the formula  $x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}$ ,” for  $a=1$  and  $a=3$ , as I have shown below.

Assuming  $x_0 = 0$ ,

$$x_1 = x_0 + c_1$$

$\vdots$

$$x_{28} = x_{27} + c_{28}$$

$$x_{29} = x_{28} + (c_1 + c_3 + c_4)$$

$\vdots$

$$x_{49} = x_{48} + (c_{26} + c_{27} + c_{28})$$

72. Thus, it is my opinion that *Kobayashi* discloses “generating a sequence of parity bits, wherein each parity bit ‘ $x_j$ ’ in the sequence is in accordance with the formula  $x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}$  where ‘ $x_{j-1}$ ’ is the value of a parity bit ‘ $j-1$ ,’ and ‘ $\sum_{i=1}^a v_{(j-1)a+i}$ ’ is the value of a sum of ‘ $a$ ’ randomly chosen irregular repeats of the message bits.”

**d) making the sequence of parity bits available for transmission in a transmission data stream.**

73. In my opinion, *Kobayashi* discloses these limitations. For example, *Kobayashi* discloses that the encoded sequence of parity bits  $I_4$  is made available for transmission, and transmitted via duobinary signaling, to a decoder that receives the transmission data stream.

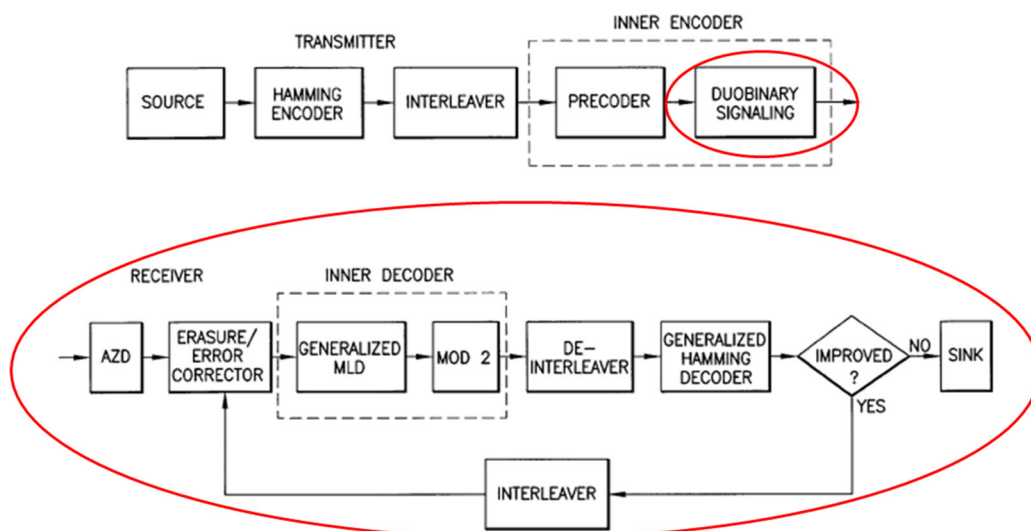


FIG.8

(Ex. 1005, FIG. 8 (annotated to show transmission via duobinary signaling and the decoder).) While duobinary signaling is depicted in Figure 8 above as part of the “inner encoder,” *Kobayashi* describes duobinary signaling as the transmission technique for transmitting the sequence of parity bits  $I_4$  to the decoder. (*Id.*, 8:25-32 (describing the “duobinary sequence which might be observed at the channel output” after encoded data stream  $I_4$  is transmitted and received), 7:30-31 (“The precoder introduces a simple transformation prior to the transmission by duobinary signaling.”), 7:43-45 (“Duobinary signaling illustrated in this example is a simplest case of partial-response channel coding referred to in the Background of the Art.”), 2:21-25 (“Partial-response channel coding is well recognized as a bandwidth-efficient transmission technique . . .”).) *Kobayashi* discloses that the decoder

receives the transmission data stream in the form of a duobinary sequence “ $I_5=(0000000, 0001112, 2112210, 1210122, 1122112, 2111122, 1121000)$ ” after it is transmitted to the decoder via duobinary signaling, but “[b]ecause of channel noise or interference, a received (and sampled) sequence will deviate from the sequence  $I_5$ .” (*Id.*, 8:25-34.)

74. Accordingly, it is my opinion that *Kobayashi* discloses “making the sequence of parity bits available for transmission in a transmission data stream.”

## 2. Claim 3

- a) **The method of claim 1, wherein the sequence of parity bits is generated is in accordance with “a” varying for different parity bits.**

75. In my opinion, *Kobayashi* discloses these limitations. As I discussed for claim 1(c), the sequence of parity bits is generated in accordance with  $a=1$  and  $a=3$ —that is, “a” varies for different parity bits. (Section IX.A.1(c).)

## 3. Claim 4

- a) **The method of claim 1, wherein generating the sequence of parity bits comprises performing recursive modulo two addition operations on the random sequence of bits.**

76. In my opinion, *Kobayashi* discloses these limitations. As I discussed for claim 1(c), *Kobayashi* discloses that the precoder generates the sequence of parity bits  $I_4$  by performing modulo-2 addition operations on the random sequence

of bits in  $I_3$  and previous parity bits in  $I_4$ . (Section IX.A.1(c).) The precoder operation is also a recursive operation because it defines each parity bit  $x_j$  in terms of preceding bit  $x_{j-1}$ . Thus, in my opinion, *Kobayashi* discloses “generating the sequence of parity bits comprises performing recursive modulo two addition operations on the random sequence of bits.”

#### 4. Claim 5

- a) The method of claim 1, wherein generating the sequence of parity bits comprises:
- b) generating a random sequence of bits that repeats each of the message bits one or more times with the repeats of the message bits being distributed in a random sequence, wherein different fractions of the message bits are each repeated a different number of times and the number of repeats for each message bit is irregular; and

77. In my opinion, *Kobayashi* discloses these limitations. As I discussed for claim 1(c), the *Kobayashi* method “generat[es] the sequence of parity bits”  $I_4$  by generating at the conclusion of the interleaver step the permuted sequence  $I_3$ , which as discussed is a “random sequence of bits that repeats each of the message bits one or more times with the repeats of the message bits being distributed in a random sequence.” (Section IX.A.1(c).) As I also discussed for claim 1(c), prior to the interleaving step, the Hamming encoder produces a sequence  $I_2$  of repeated bits, in which the 3<sup>rd</sup>, 7<sup>th</sup>, 11<sup>th</sup>, 15<sup>th</sup>, 19<sup>th</sup>, 23<sup>rd</sup>, and 27<sup>th</sup> message bits of  $I_1$  are repeated four



times, while the other 21 message bits of  $I_1$  are repeated three times. (Section IX.A.1(c).) Thus, it is my opinion that “the number of repeats for each message bit is irregular” because  $1/4$  of the message bits are repeated four times and  $3/4$  of the message bits are repeated three times (*i.e.*, “different fractions of the message bits are each repeated a different number of times”).

**c) XOR summing in linear sequential fashion a predecessor parity bit and “a” bits of the random sequence of bits.**

78. In my opinion, *Kobayashi* discloses these limitations. As I discussed for claim 1(c), *Kobayashi* discloses that the precoder operation applies a sequential operation (“sequential fashion”) that takes a mod-2 sum of the preceding parity bit of  $I_4$  (“a predecessor parity bit”) and a current bit of  $I_3$ , which is the sum of “‘a’ bits of the random sequence of bits.” (Section IX.A.1(c).) As I further discussed, *Kobayashi* discloses that this precoder operation is implemented using an exclusive OR operation (“XOR summing”). (Section IX.A.1(c); *see also* Section V.B (background on modulo-2 arithmetic).) Thus, *Kobayashi* discloses “XOR summing in . . . sequential fashion a predecessor parity bit and ‘a’ bits of the random sequence of bits.”

79. In my opinion, this precoder operation also XOR sums in “linear [] fashion” because it is a linear operation. A person of ordinary skill in the art would

have understood that a coder, such as *Kobayashi*'s precoder, operates in "linear [] fashion" if it is a linear transformation, *i.e.*, if its operation can be represented as a linear block code. (See Section V.A (background on error control coding and linear block codes).) As I explained for claim 1(c), the precoder's operation—which involves XOR summing as described immediately above—can be represented as a (49, 49) linear block code and is thus a linear transformation. (Section IX.A.1(c).) In particular, the precoder's operation is numerically equivalent to a matrix multiplication operation in which the 1x49 vector  $I_3$  is multiplied by a 49x49 generator matrix  $\mathbf{G}_A$  below with "1"s both along and above the main diagonal and "0"s below the main diagonal, resulting in 1x49 vector  $I_4$ . (Section IX.A.1(c); see also Ex. 1009, 44 ("the encoding operation [of a linear block code] is represented mathematically as  $\mathbf{v} = \mathbf{u} \cdot \mathbf{G}$ , where  $\mathbf{v}$  is a vector of the encoded data bits,  $\mathbf{u}$  is a vector of  $k$  information bits, and  $\mathbf{G}$  is the *generator matrix*" (emphasis in original)); see also Section V.B (background on matrix multiplication).) Thus, in my opinion, the precoder's XOR summing operation occurs in a "linear [] fashion."

$$\mathbf{G}_A = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & 1 & 1 & \dots & 1 \\ 0 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

80. It is my opinion that *Kobayashi* therefore discloses “XOR summing in linear sequential fashion a predecessor parity bit and ‘a’ bits of the random sequence of bits.”

**5. Claim 7**

- a) **The method of claim 5, wherein generating the random sequence of bits comprises:**
- b) **producing a block of data bits, wherein different message bits are each repeated a different number of times in a sequence that matches the first sequence; and**

81. In my opinion, *Kobayashi* discloses these limitations. As I discussed for claim 1(c), the first encoding step (Hamming code) involves repetition of each message bit in the first sequence to produce a block of data bits  $I_2$ , such that all message bits are repeated, and different message bits are each repeated a different number of times. (Section IX.A.1(c).) Thus, it is my opinion that *Kobayashi* discloses “producing a block of data bits, wherein different message bits are each repeated a different number of times in a sequence that matches the first sequence.”

- c) **randomly permuting the different bits to generate the random sequence.**

82. In my opinion, *Kobayashi* discloses these limitations. As I discussed for claims 1(c) and 5(b), *Kobayashi*’s interleaver performs a permutation action as part of the method to generate a random sequence of bits that repeats each one of the

message bits one or more times. (Sections IX.A.1(c), IX.A.4(b).) As I further discussed, a person of ordinary skill in the art would have understood random permutations of bits, as claimed, to encompass pseudo-random or arbitrary permutations of bits, such as performed by *Kobayashi*'s interleaver. (Sections IX.A.1(c), IX.A.4(b).) Thus, it is my opinion that *Kobayashi* discloses "randomly permuting the different bits to generate the random sequence."

**6. Claim 8**

**a) The method of claim 1, further comprising transmitting the sequence of parity bits.**

83. In my opinion, *Kobayashi* discloses these limitations for the same reasons as I explained for claim 1(d). As I explained, the *Kobayashi* method both makes the sequence of parity bits available for transmission, and actually transmits the sequence of parity bits, to the decoder. (Section IX.A.1(d).)

**7. Claim 11**

**a) A device comprising:**

84. I understand that "[a] device comprising:" is the preamble of claim 11. I have been asked to assume that the preamble is limiting. In my opinion, *Kobayashi* discloses the limitations of the preamble. For example, *Kobayashi* discloses a concatenated system with both transmitter and receiver portions, wherein the transmitter is "[a] device," as claimed, which receives message bits from a source

and encodes the bits using several encoders. (Ex. 1005, FIG. 8, 5:25-27, 7:5-8:34 (describing the transmitter device in the context of Figure 8); *see also* Section V.A (background on error control coding).)

85. In particular, *Kobayashi* discloses that the device (as shown in Figure 8 below) receives message bits from a source via a packet transmission system; encodes the sequence of message bits using the Hamming encoder, interleaver, and precoder components to generate a sequence of parity bits; and makes the encoded sequence available for transmission to the receiver via duobinary signaling. (Ex. 1005, 7:5-8:34.)

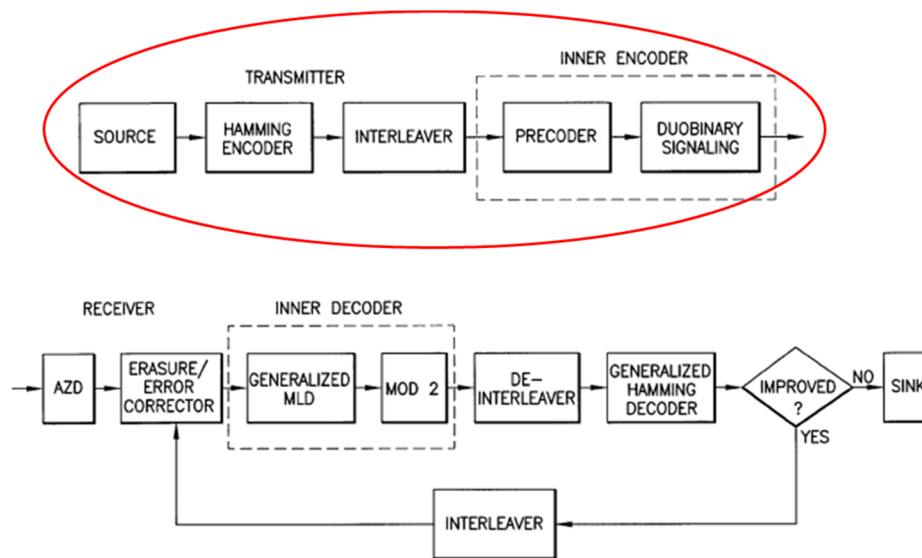


FIG.8

(Ex. 1005, FIG. 8 (annotated); *see also id.*, 5:17-24.)

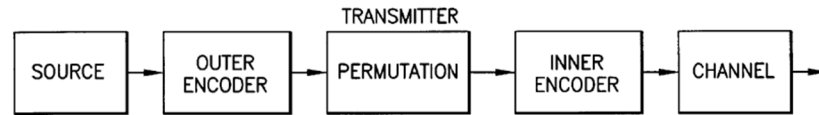


FIG.7A

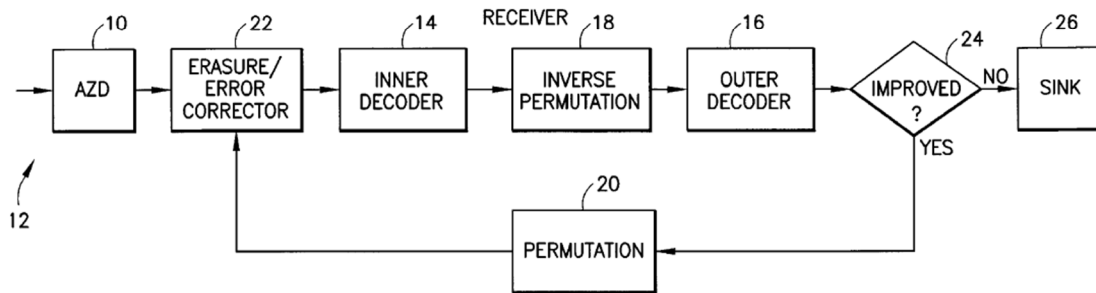


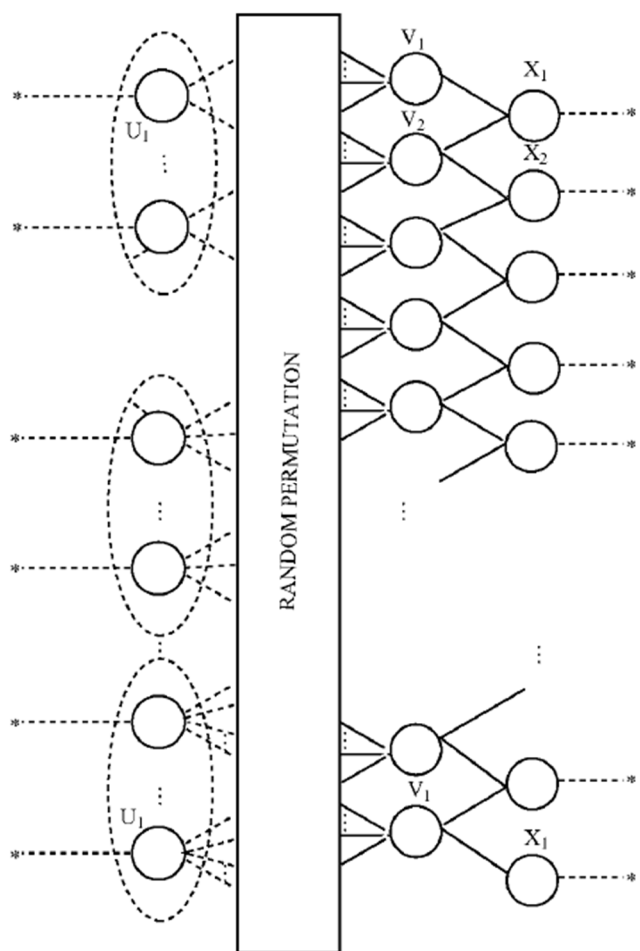
FIG.7B

(*Id.*, FIGS. 7A, 7B (showing generalized versions of the transmitter and receiver sides of Figure 8).)

- b) **an encoder configured to receive a collection of message bits and encode the message bits to generate a collection of parity bits in accordance with the following Tanner graph:<sup>8</sup>**

---

<sup>8</sup> I understand that the Certificate of Correction appended to the '032 patent states that the bottom  $U_1$  of the Tanner graph should read  $U_k$ , the bottom  $V_1$  should read  $V_r$ , and the bottom  $X_1$  should read  $X_r$ . (*See* Ex. 1001.) My analysis in this declaration reflects this corrected graph.



86. In my opinion, *Kobayashi* discloses these limitations for the same reasons as I discussed above for claim 1. The '032 Patent states that the Tanner graph shown in Figure 3 represents a binary linear block code with the recursive formula:  $x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}$ . (Ex. 1001 at 3:34-4:15.) The Tanner graph of Figure 3 is the same as the Tanner graph appearing in Claim 11 and the equation in column 4, lines 13-15, shown above, is the same as the formula appearing in claim 11(c). Thus, a code that implements the formula of claim 1 also “generate[s] a collection of parity

bits in accordance with the [ ] Tanner graph” of claim 11, as required. As discussed for claim 1(c), *Kobayashi* discloses that the transmitter has an encoding mechanism consisting of a Hamming code, interleaver, and precoder (together, “an encoder” as claimed). (Section IX.A.1(c).) As I discussed for claims 1(b) and 1(c), *Kobayashi*’s encoder is “configured to receive a collection of message bits” in the form of 28-bit sequence  $I_1$ , and “encode the message bits to generate a collection of parity bits” in the form of 49-bit sequence  $I_4$ . (Sections IX.A.1(b)-(c).)

87. In my opinion, the encoder also operates “in accordance with the [above] Tanner graph” because *Kobayashi*’s Hamming code and interleaver, along with *Kobayashi*’s precoder, together correspond to the above Tanner graph, as I describe below. *Kobayashi* encoder is an irregular repeat accumulate encoder for the reasons I discussed above for claim 1.

88. As I discussed for claim 1(c), the encoder first applies a (7, 4) Hamming code to each of seven 4-bit blocks of 28-bit sequence  $I_1$ , generating a 49-bit Hamming code output  $I_2$ . A person of ordinary skill in the art would have recognized that this encoding repeats the message bits such that subsets of message bits are repeated different numbers of times, as I explained in Section IX.A.1(c), above. A person of ordinary skill in the art would have understood this irregular repetition



corresponds to the dashed lines on the left side of the Tanner graph extending from the message bits ( $U_j$ ) to the permutation box in the middle of the Tanner graph.

89. As I discussed for claim 1(c), the encoder then applies a 7x7 interleaver, transforming the 49-bit sequence  $I_2$  to the 49-bit sequence  $I_3$ . (See Section IX.A.1(c).) In my opinion, a person of ordinary skill in the art would have also recognized that, together, the Hamming encoder (discussed in the previous paragraph) and interleaver apply a random permutation action that randomly connects the irregularly repeated message bits to enforce constraints that determine the parity bits ( $I_4$ ). This corresponds to the random permutation box and its connections with check nodes ( $V_j$ ) in the middle of the Tanner graph. My opinion is supported by the fact that the '032 Patent describes this portion of the Tanner graph as corresponding to the aforementioned permutation of Kobayashi. The '032 Patent states:

Each check node 304 is **connected** to exactly “a” information nodes 302. In FIG. 3,  $a=3$ . These **connections** can be made in many ways, as indicated by the arbitrary permutation of the  $ra$  edges joining information nodes 302 and check nodes 304 in permutation block 310. **These**

**connections correspond to the scrambling performed  
by the interleaver 204.**

Ex. 1001 at 3:49-55 (discussing the embodiment of Figure 3).

90. The '032 Patent states the Tanner graph represents both a “non-systematic” and “systematic” IRA code so a POSITA would have understood the Tanner graph of claim 11 covers both types of codes. (Ex. 1001, 4:19-25.)

91. As I also discussed for claim 1(c), the encoder then accumulates the bits of 49-bit sequence  $I_3$ , resulting in 49-bit sequence  $I_4$ . (See Section IX.A.1(c).) As I discussed for claim 1(c), this accumulation operation is achieved through the operation  $x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}$  where  $\sum_{i=1}^a v_{(j-1)a+i}$  is the output of check node  $V_{j-i}$  because it is the sum of a randomly chosen irregular repeats of the message/information bits (See Section IX.A.1(c).). Thus, the accumulation operation corresponds to the right-hand side of the Tanner graph, where parity bits ( $I_4$ ) are determined as a function of both message/information bits and other parity bits in the same way shown by the configuration of nodes and edges of the right side Tanner graph.

92. Thus, in my opinion, *Kobayashi* discloses “an encoder configured to receive a collection of message bits and encode the message bits to generate a collection of parity bits in accordance with the . . . Tanner graph” above because the

generator matrix representing *Kobayashi*'s Hamming code and interleaver, along with *Kobayashi*'s precoder, together correspond to the above Tanner graph and comprise an "an encoder configured to receive a collection of message bits and encode the message bits to generate a collection of parity bits."

## 8. Claim 12

- a) **The device of claim 11, wherein the encoder is configured to generate the collection of parity bits as if a number of inputs into nodes  $v_i$  was not constant.**

93. In my opinion, *Kobayashi* discloses these limitations. As I discussed for claim 11(b), "the encoder is configured to generate the collection of parity bits" for "a number of inputs into nodes  $v_i$  [that is] not constant" because the encoder operates in accordance with a Tanner graph in which the number of inputs equals either one ( $a=1$ ) or three ( $a=3$ ). (Section IX.A.7(b).)

## 9. Claim 14

- a) **The device of claim 12, wherein the accumulator comprises a recursive convolutional coder.<sup>9</sup>**

94. In my opinion, *Kobayashi* discloses these limitations. As I discussed for claims 11 and 12, *Kobayashi*'s encoder includes a precoder that performs

---

<sup>9</sup> Neither claim 12, nor claim 11 from which claim 12 depends, mentions an "accumulator." Nevertheless, my analysis demonstrates how *Kobayashi* discloses

operations on encoded sequence  $I_3$  by taking the modulo-2 partial sum of the current input of  $I_3$  and the immediately previous output of  $I_4$  (*i.e.*, the modulo-2 partial sum of all previous inputs up to the current input). (See Section IX.A.1(d); *see also* Section V.B (background on modulo-2 arithmetic).) In my opinion, the precoder is an “accumulator” that accumulates the bits in sequence  $I_3$  because the precoder’s accumulation operation is consistent with that disclosed by the specification of the ’032 patent. The ’032 patent describes an accumulator as follows:

In an embodiment, the inner coder 206 is an **accumulator, which produces outputs that are the modulo two (mod-2) partial sums of its inputs**. The accumulator may be a truncated rate-1 **recursive convolutional coder with the transfer function  $1/(1+D)$** . Such an accumulator may be considered a block coder whose input block  $[x_1, \dots, x_n]$  and output block  $[y_1, \dots, y_n]$  are related by the formula

$$\begin{aligned}y_1 &= x_1 \\y_2 &= x_1 \oplus x_2 \\y_3 &= x_1 \oplus x_2 \oplus x_3 \\y_n &= x_1 \oplus x_2 \oplus x_3 \oplus \dots \oplus x_n\end{aligned}$$

---

that the “device of claim 12” includes an “accumulator [that] comprises a recursive convolutional coder.”

where “ $\oplus$ ” denotes mod-2, or exclusive-OR (XOR),  
addition.

(Ex. 1001, 2:66-3:24 (emphasis added).)

95. As shown above, the accumulator of the '032 patent is described as a “truncated rate-1 *recursive convolutional coder* with the transfer function  $1/(1+D)$ .” (*Id.*, 3:1-3 (emphasis added).) Thus, it is my opinion that *Kobayashi*'s precoder is an “accumulator [that] comprises a recursive convolutional coder” as claimed because the precoder of the device of claim 12 operates in the same manner as the accumulator described in the '032 patent.

#### 10. Claim 15

- a) **The device of claim 14, wherein the recursive convolutional coder comprises a truncated rate-1 recursive convolutional coder.**

96. In my opinion, *Kobayashi* discloses these limitations for the same reasons as I explained for claim 14. As I described for claim 14, *Kobayashi*'s precoder is an “accumulator [that] comprises a recursive convolutional coder” because it operates in the same manner as the accumulator disclosed by the specification of the '032 patent, which describes its accumulator as a “*truncated rate-1 recursive convolutional coder* with the transfer function  $1/(1+D)$ .” (Section IX.A.9; Ex. 1001, 3:1-3 (emphasis added).) Thus, it is my opinion that *Kobayashi*'s

precoder/accumulator is also a “recursive convolutional coder [that] comprises a truncated rate-1 recursive convolutional coder.”

## 11. Claim 16

- a) **The device of claim 14, wherein the recursive convolutional coder has a transfer function of  $1/(1+D)$ .**

97. In my opinion, *Kobayashi* discloses these limitations for the same reasons as I explained for claim 14. As I described for claim 14, *Kobayashi*’s precoder is an “accumulator [that] comprises a recursive convolutional coder” because it operates in the same manner as the accumulator disclosed by the specification of the ’032 patent, which describes its accumulator as a “truncated rate-1 *recursive convolutional coder with the transfer function  $1/(1+D)$* .” (Section IX.A.9; Ex. 1001, 3:1-3 (emphasis added).) Thus, it is my opinion that *Kobayashi*’s precoder/accumulator is also a “recursive convolutional coder [that] has a transfer function of  $1/(1+D)$ .”

**B. *Kobayashi* Discloses and/or Suggests the Features of Claims 2, 6-7, 10, 13, and 17**

**1. Claim 2**

- a) The method of claim 1, wherein the sequence of parity bits is generated is in accordance with “a” being constant.**

98. In my opinion, *Kobayashi* discloses or suggests these limitations. As I have discussed, *Kobayashi* discloses that the overall encoding mechanism is a nonsystematic code that generates 49 parity bits, where  $a=1$  and  $a=3$ . (See Section IX.A.1(c).) However, in my opinion, a person of ordinary skill in the art would have been motivated to implement the *Kobayashi* method using a *systematic* encoding mechanism, in which the 28 message bits appear directly in the codeword along with the 21 parity bits following the 28 message bits. (Ex. 1009, 45 (describing a “systematic code” as one in which the codeword “can be partitioned into two parts, with one part consisting of the  $k$ -bit information word and the other part consisting of the  $n - k$  parity digits”); see also Section V.A (background on error control coding and systematic codes).)

99. In my opinion, a person of ordinary skill in the art would have recognized that the *Kobayashi* method could be easily modified to use a systematic encoding mechanism by applying the precoder operation *only* over the last 21 bits of  $I_3$  (highlighted in green below), and replacing the first 28 message bits of  $I_3$

(highlighted in yellow below) with the 28 message bit sequence  $I_1$  (in order,  $c_1$  through  $c_{28}$ ).

(unmodified)  $I_3 = (c_1, c_5, c_9, c_{13}, c_{17}, c_{21}, c_{25}, c_2, c_6, c_{10}, c_{14}, c_{18}, c_{22}, c_{26}, c_3, c_7, c_{11}, c_{15}, c_{19}, c_{23}, c_{27}, c_4, c_8, c_{12}, c_{16}, c_{20}, c_{24}, c_{28}, c_1 + c_3 + c_4, c_5 + c_7 + c_8, c_9 + c_{11} + c_{12}, c_{13} + c_{15} + c_{16}, c_{17} + c_{19} + c_{20}, c_{21} + c_{23} + c_{24}, c_{25} + c_{27} + c_{28}, c_1 + c_2 + c_3, c_5 + c_6 + c_7, c_9 + c_{10} + c_{11}, c_{13} + c_{14} + c_{15}, c_{17} + c_{18} + c_{19}, c_{21} + c_{22} + c_{23}, c_{25} + c_{26} + c_{27}, c_2 + c_3 + c_4, c_6 + c_7 + c_8, c_{10} + c_{11} + c_{12}, c_{14} + c_{15} + c_{16}, c_{18} + c_{19} + c_{20}, c_{22} + c_{23} + c_{24}, c_{26} + c_{27} + c_{28})$

(modified)  $I_4 = (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{16}, c_{17}, c_{18}, c_{19}, c_{20}, c_{21}, c_{22}, c_{23}, c_{24}, c_{25}, c_{26}, c_{27}, c_{28}, [21 \text{ parity bits representing the precoder output applied over the last 21 bits of } I_3])$

100. With such a modification, the “sequence of parity bits” that is generated by the modified *Kobayashi* method would thus be comprised of the last 21 bits of  $I_4$ , rather than the entire sequence of  $I_4$ . (See Section IX.A.1(c); see also Section V.A (background on error control coding and parity bits of systematic codes).) As I previously described for claim 1(c),  $a=3$  for this entire 21-bit sequence (*i.e.*, “ $a$ ” is constant). (Section IX.A.1(c).) Further, the modified “sequence of parity bits” would still be available for transmission because it is part of sequence  $I_4$ , which as discussed for claim 1(d), is transmitted to the decoder. (Section IX.A.1(d).)

101. In my opinion, a person of ordinary skill in the art would have recognized that there are only two types of codes—systematic and non-systematic—and thus would have been motivated to pursue a modification with a systematic code. Moreover, a person of ordinary skill in the art would have recognized the benefits of



a systematic code. In particular, because the parity bits are merely appended to the sequence of message bits, a systematic code has the advantage that the receiver does not need to decode the original input stream if the bits are received correctly. (*See* Ex. 1010, 54 (“A desirable property for a linear block code to possess is the *systematic structure* of the code words . . . , where a code word is divided into two parts, the message part and the redundant checking part.” (emphasis in original)).)

102. In my opinion, such a modification would have been within the skills and capabilities of a person of ordinary skill in the art. A person of ordinary skill in the art would have also had a reasonable expectation of success in achieving such a modification because it would have involved nothing more than an application of a known element (systematic code) to a known device (transmitter with concatenated encoders) to yield predictable results (provide an encoded output that includes not only parity bits, but also all of the message bits). Thus, in my opinion, *Kobayashi* discloses or suggests that “the sequence of parity bits is generated is in accordance with ‘a’ being constant.”

**2. Claim 6**

- a) **The method of claim 5, wherein generating the random sequence of bits comprises coding the collection of message bits using a low-density generator matrix (LDGM) coder.**

103. In my opinion, *Kobayashi* discloses or suggests these limitations. As I previously discussed for claim 5(b), *Kobayashi* discloses “generating the random sequence of bits” via a (7, 4) Hamming code and an interleaver. (Section IX.A.4(b).) *Kobayashi* discloses that the (7, 4) Hamming code could be replaced with a different encoder—for example, with a different (n, k) Hamming code, or with a Reed-Solomon or BCH code.

It is easiest to explain the invention by way of an example. A concatenated system of the type shown in FIG. 3 is shown in further detail in FIG. 8 which illustrates both the transmission and reception sides. **As the outer code, a (7, 4) Hamming code is used** and the inner code is duobinary signaling with a precoder. **An (n, k) Hamming code is a single error correcting code**, which can correct any single error that may exist in a block of bits, consisting of message bits, and parity-check bits (see e.g., Lin/Costello or Peterson/Weldon for details on Hamming codes).

(Ex. 1005, 7:6-15 (emphasis added).)

In the above description, it has been assumed that the information source is binary data. The invention can also be applied to a non-binary system. **For instance, a Reed-Solomon code can be used as an error correcting code instead of Hamming or BCH codes.**

(*Id.*, 11:18-22 (emphasis added).) Moreover, as shown above, *Kobayashi* incorporates other references such as the *Lin/Costello* text for further details on those modified implementations. (*Id.*, 7:6-15.)

104. It is my opinion that a person of ordinary skill in the art would have therefore been motivated to consult the incorporated *Lin/Costello* reference for further guidance on additional implementations. A person of ordinary skill in the art would have understood that an  $(n, k)$  Hamming code can be implemented for “any positive integer  $m \geq 3$ ,” where the code length is  $n = 2^m - 1$ , the number of information bits is  $k = 2^m - m - 1$ .

Hamming codes are the first class of linear codes devised for error correction [6]. These codes and their variations have been widely used for error control in digital communication and data storage systems.

**For any positive integer  $m \geq 3$ , there exists a Hamming code with the following parameters:**

**Code length:**  $n = 2^m - 1$

**Number of information symbols:**  $k = 2^m - m - 1$

Number of parity-check symbols:  $n - k = m$

Error-correcting capability:  $t = 1 (d_{\min} = 3).$

(Ex. 1010, 79 (emphasis added); *see also* Ex. 1005, 1:65-67 (providing a citation to the *Lin/Costello* text (*i.e.*, Ex. 1010).)

105. In my opinion, a person of ordinary skill in the art would have understood from the disclosures of *Kobayashi* and its incorporated *Lin/Costello* reference that instead of using a (7, 4) Hamming code (*i.e.*,  $m = 3$ ), it would have been possible to use a (255, 247) Hamming code (*i.e.*,  $m = 8$ ). A person of ordinary skill in the art would have had the skills to achieve such a configuration, and similarly, had a reasonable expectation that such an implementation would have successfully performed as intended. In my opinion, such a modification would have been a mere application of a known element ((255, 247) Hamming code) to a known device (*Kobayashi*'s transmitter that uses a (7, 4) Hamming code and interleaver to generate a random sequence of bits) to yield predictable results (generate a random sequence of bits using a (255, 247) Hamming code and interleaver). A person of ordinary skill in the art would have recognized a benefit of making this modification to be an increase in the code rate, from a rate of 4/7 for the (7, 4) Hamming code to a rate of 247/255 for the (255, 247) Hamming code, and that such an increase in code

rate translates to better spectral efficiency; *i.e.*, relatively more of the transmission corresponds to data and less of the transmission must be used on the overhead of transmitting parity bits. (See Section V.A (background on error control coding and code rate).)

106. Further, it is my opinion that a person of ordinary skill in the art would have understood that for this modified *Kobayashi* method, the systematic generator matrix  $\mathbf{G}$  for a (255, 247) Hamming code is a 247x255 matrix comprised of two submatrices: a 247x247 identity matrix, and a 247x8 matrix  $\mathbf{Q}^T$  in which the 247 rows are the 247 nonzero  $m$ -tuples (eight-tuples) of weight 2 or more.

The parity-check matrix  $\mathbf{H}$  of this [Hamming] code consists of all the nonzero  $m$ -tuples as its columns. In systematic form, the columns of  $\mathbf{H}$  are arranged in the following form:

$$\mathbf{H} = [\mathbf{I}_m \quad \mathbf{Q}],$$

where  $\mathbf{I}_m$  is an  $m \times m$  identity matrix and the submatrix  $\mathbf{Q}$  consists of  $2^m - m - 1$  columns which are the  $m$ -tuples of weight 2 or more.

...

The columns of  $\mathbf{Q}$  may be arranged in any order without affecting the distance property and weight distribution of the code. In systematic form, the generator matrix of the code is

$$\mathbf{G} = [\mathbf{Q}^T \quad \mathbf{I}_{2^m - m - 1}],$$

where  $\mathbf{Q}^T$  is the transpose of  $\mathbf{Q}$  and  $\mathbf{I}_{2^m - m - 1}$  is an  $(2^m - m - 1) \times (2^m - m - 1)$  identity matrix.

(Ex. 1010, 79-80.) This is because a person of ordinary skill in the art would have understood from the above disclosures of the incorporated *Lin/Costello* reference that the transpose of  $\mathbf{Q}$  (i.e.,  $\mathbf{Q}^T$ ) consists of  $2^m - m - 1$  rows (rather than columns) which are the  $m$ -tuples of weight 2 or more.<sup>10</sup>

107. A person of ordinary skill in the art would have recognized that this 247x255 generator matrix has a density of approximately 2%. The 247x247 identity matrix has 247 “1”s along the diagonal. The submatrix  $\mathbf{Q}^T$  is a 247x8 matrix whose

---

<sup>10</sup> I note here a trivial difference in how *Kobayashi* and *Lin/Costello* describe systematic Hamming codes. In *Kobayashi*, the identity matrix  $\mathbf{I}$  is on the left side of the generator matrix  $\mathbf{G}$ , while in *Lin/Costello* it is on the right side. This is an artifact of *Lin/Costello* placing the systematic bits *after* the parity bits, while *Kobayashi*, like Ex. 1009 (“*Rorabaugh*”), places the systematic bits *before* the parity bits. As I have noted in the Technical Background section, this is a design choice that does not impact the error detecting and correcting capabilities of the code. (See Section V.A (background on error control coding).)

rows are the  ${}_8C_2 = 28$  eight-tuples of weight 2;  ${}_8C_3 = 56$  eight-tuples of weight 3;  ${}_8C_4 = 70$  eight-tuples of weight 4;  ${}_8C_5 = 56$  eight-tuples of weight 5;  ${}_8C_6 = 28$  eight-tuples of weight 6;  ${}_8C_7 = 8$  eight-tuples of weight 7; and  ${}_8C_8 = 1$  eight-tuple of weight 8.<sup>11</sup> That is, submatrix  $\mathbf{Q}^T$  has  $28*2 + 56*3 + 70*4 + 56*5 + 28*6 + 8*7 + 1*8 = 1,016$  “1”s, and thus density =  $[(247 + 1,016) / (247*255)] * 100\% = 2\%$ . In my opinion, a person of ordinary skill in the art would have understood 2% to be low-density, and would have recognized that a Hamming code is a “generator matrix . . . coder.”

108. In my opinion, and as further described below, a person of ordinary skill in the art would have also understood that with such a modification, the sequence of parity bits  $I_4$  would have still been generated in accordance with claims 1(c) and 5(b). (See Sections IX.A.1(c), IX.A.4(b).) In particular, the modified *Kobayashi* method would still operate according to the equation of claim 1(c), but with message bits irregularly repeated a different number of times, and with a different value of “a.” (Section IX.A.1(c).)

109. As a person of ordinary skill in the art would have recognized, the number of “1”s in each row of the LDGM (*i.e.*, one “1” for the identity matrix plus

---

<sup>11</sup> I use the notation  ${}_nC_r = n! / (r! (n-r)!)$  to refer to the number of ways to choose  $r$  elements from a set of  $n$  distinct objects, assuming order does not matter.

the number of “1”s in each row of the last eight columns) determines how many times each message bit is repeated. Thus, as discussed above, 28 message bits would be repeated 3 times; 56 bits would be repeated 4 times; 70 bits would be repeated 5 times; 56 bits would be repeated 6 times; 28 bits would be repeated 7 times; 8 bits would be repeated 8 times; and 1 bit would be repeated 9 times. Such a modification would have therefore still had “irregular repeats of the message bits,” *see* Section IX.A.1(c), and would have still “repeat[ed] each of the message bits one or more times . . . wherein different fractions of the message bits are each repeated a different number of times and the number of repeats for each message bit is irregular,” *see* Section IX.A.4(b).

110. A person of ordinary skill in the art would have understood that the number of “1”s in each column of the LDGM determines the value of “a” for the equation in claim 1(c). (Section IX.A.1(c).) In particular, “a” would equal 1 for the first 247 columns of the generator matrix that comprise an identity matrix. The last eight columns of the (255, 247) Hamming code generator matrix each have the same number of “1”s, and thus “a” would be the same value for all eight columns. As I described above, the last eight columns of the generator matrix  $\mathbf{G}$  comprise a  $247 \times 8$  submatrix  $\mathbf{Q}^T$  whose rows are the  ${}_8C_2 = 28$  eight-tuples of weight 2;  ${}_8C_3 = 56$  eight-tuples of weight 3;  ${}_8C_4 = 70$  eight-tuples of weight 4;  ${}_8C_5 = 56$  eight-tuples of weight



5;  ${}_8C_6 = 28$  eight-tuples of weight 6;  ${}_8C_7 = 8$  eight-tuples of weight 7; and  ${}_8C_8 = 1$  eight-tuple of weight 8. It follows that  $\mathbf{Q}^T$  has 1,016 “1”s over 8 columns, *i.e.*, each column of  $\mathbf{Q}^T$  has exactly 127 “1”s in it, and accordingly, “a” equals 127 for the last 8 columns of the generator matrix. Thus, it is my opinion that the modified *Kobayashi* method would have still met the equation of claim 1(c), but for  $a=1$  and  $a=127$ . (Section IX.A.1(c).)

111. Accordingly, a person of ordinary skill in the art would have understood that the modified *Kobayashi* method would have still met the equation of claim 1(c), but for  $a=1$  and  $a=127$ . (Section IX.A.1(c).)

112. Furthermore, for claim 5(b), it is my opinion that the modified *Kobayashi* method would have still “generat[ed] a random sequence of bits . . . with the repeats of the message bits being distributed in a random sequence” because the interleaver operation would still be applied. (Section IX.A.4(b).)

113. For the reasons I have discussed above, it is my opinion that *Kobayashi* therefore discloses or suggests “generating the random sequence of bits comprises coding the collection of message bits using a low-density generator matrix (LDGM) coder.”

**3. Claim 7**

- a) The method of claim 5, wherein generating the random sequence of bits comprises:**
- b) producing a block of data bits, wherein different message bits are each repeated a different number of times in a sequence that matches the first sequence; and**

114. In my opinion, *Kobayashi* discloses or suggests these limitations for the same reasons I previously explained for claims 7(a)-(b). (Section IX.A.5(a)-(b).) I have been asked to consider a scenario in which *Kobayashi* alone does not disclose that “different message bits are each repeated . . . *in a sequence that matches the first sequence.*” Under that scenario, it is my opinion that a person of ordinary skill in the art would have been motivated to implement the (7, 4) Hamming code, which performs the irregular repetition, using known methods for matrix multiplication to multiply the blocks of sequence  $I_1$  by generator matrix  $G$ .

115. For example, the below code fragment is an example of a matrix multiplication method known at the time of the alleged invention for implementing an (n, k) linear block code, such as *Kobayashi*’s (7, 4) Hamming code. (Ex. 1006, 44-45.)

```

/*****
for( i=0; i<n; i++){
    v[i] =0;
    for( j=0; j<k; j++ ) {
        v[i] = v[i] ^ (u[j] & g[j][i]);
    }
}
/* end of fragment */
*****/

```

(*Id.*, 45.) The C code fragment reflects an “encoding operation” that can be “represented mathematically as  $\mathbf{v} = \mathbf{u} \cdot \mathbf{G}$ , where  $\mathbf{v}$  is a vector of the encoded bits,  $\mathbf{u}$  is a vector of  $k$  information bits, and  $\mathbf{G}$  is the *generator matrix*,” and where  $\mathbf{v}$  is an  $n$ -bit vector,  $\mathbf{u}$  is a  $k$ -bit vector, and  $\mathbf{G}$  is a  $k \times n$  generator matrix. (*Id.*, 44 (emphasis in original); see also Sections V.A (background on error control coding), V.B (background on matrix multiplication).)

116. In my opinion, a person of ordinary skill in the art would have recognized that the inner loop of the fragment increments  $j$  sequentially, and thus iterates sequentially over the sequence of message bits in order to generate the repeated bits. Thus, a person of ordinary skill in the art would have understood that when implementing the *Kobayashi* method’s Hamming code using a similar known matrix multiplication method, the “different message bits are each repeated a different number of times in a sequence that matches the first sequence” because the

repeated bits are generated in an order corresponding to the order of the first sequence.

117. Such a modification would have been within the skills and capabilities of a person of ordinary skill in the art. Furthermore, it is my opinion that a person of ordinary skill in the art would have had a reasonable expectation of success in achieving such a modification because it would have involved nothing more than an application of a known method (matrix multiplication methods for linear block codes) to a known device (*Kobayashi*'s encoder that uses a linear block code (Hamming code)) to yield predictable results (provide an encoder that uses known matrix multiplication methods for linear block codes to implement the Hamming code).

**c) randomly permuting the different bits to generate the random sequence.**

118. In my opinion, *Kobayashi* discloses these limitations for the same reasons I previously explained for claim 7(c). (Section IX.A.5(c).)

**4. Claim 10**

**a) The method of claim 8, wherein transmitting the sequence of parity bits comprises transmitting the sequence of parity bits as part of a systematic code.**

119. In my opinion, *Kobayashi* discloses or suggests these limitations for the same reasons as I explained for claim 2 (*see* Section IX.B.1) and claims 1(d) and 8

(see Sections IX.A.1(d), IX.A.6). As I explained for claim 2, a person of ordinary skill in the art would have been motivated to modify the *Kobayashi* method so that the sequence of parity bits is generated as part of a systematic code. (Section IX.B.1.) As another example, the information bits can simply be prepended to Kobayashi's existing 49 bit codeword ( $I_4$ ), as was most commonly done in the prior art. (Ex. 1009, 45; Ex. 1012, 20, 55; Ex. 1005, 7:53-63 (showing systematic Hamming encoder with prepended information bits); Ex. 1002, ¶104.) A POSITA would have understood this could have been accomplished in any number of ways, including by simply storing the input bits into the memory used to store the first portion of the now systematic codeword or by simply prepending an identity matrix to the code's generator matrix. Such a modification would not require a modification to how the parity bits are encoded, as described above. A POSITA would have understood this could have been accomplished in any number of ways, including by simply storing the input bits into the first portion of memory used to store the codeword. As I explained for claims 1(d) and 8, the *Kobayashi* method comprises transmitting the sequence of parity bits as part of a code. (Sections IX.A.1(d), IX.A.6.)

**5. Claim 13**

- a) The device of claim 11, wherein the encoder comprises:**
- b) a low-density generator matrix (LDGM) coder configured to perform an irregular repeat on message bits having a first sequence in a source data stream to output a random sequence of repeats of the message bits; and**

120. In my opinion, *Kobayashi* discloses or suggests these limitations. As I discussed for claim 6 (addressing the modified *Kobayashi* method), a person of ordinary skill in the art would have been motivated to implement the *Kobayashi* transmitter using a “low-density generator matrix (LDGM) coder to perform an irregular repeat on [the] message bits” of sequence  $I_1$  and “output a random sequence of repeats of the message bits.” (Section IX.B.2.) As I further discussed for claim 1(b), the “message bits hav[e] a first sequence in a source data stream.” (Section IX.A.1(b).)

- c) an accumulator configured to XOR sum in linear sequential fashion a predecessor parity bit and “a” bits of the random sequence of repeats of the message bits.**

121. In my opinion, *Kobayashi* discloses or suggests these limitations. As I previously discussed for claim 5(c), the *Kobayashi* transmitter includes a precoder that “XOR sum[s] in linear sequential fashion a predecessor parity bit and ‘a’ bits of the random sequence of repeats of the message bits.” (Section IX.A.4(c).) As I

further discussed for claim 14, *Kobayashi*'s precoder is an "accumulator" because it operates consistent with the encoder the '032 patent describes as an accumulator.

(Section IX.A.9.)

**6. Claim 17**

- a) **The device of claim 12, further comprising a second accumulator configured to determine a second sequence of parity bits that defines a second condition that constrains the random sequence of repeats of the message bits.<sup>12</sup>**

122. In my opinion, *Kobayashi* discloses or suggests these limitations. *Kobayashi* discloses that "[t]he invention can be extended to a concatenated system with three or more encoders." (Ex. 1005, 12:4-5.) In my opinion, a person of ordinary skill in the art would have been motivated to add a second accumulator (in addition to the precoder, *i.e.*, the first accumulator, *see* Section IX.A.9) to the *Kobayashi* transmitter as a third encoder. A person of ordinary skill in the art would have appreciated the benefits of such an implementation because *Kobayashi*

---

<sup>12</sup> As I have explained, *see* n.7, neither claim 12, nor claim 11 from which claim 12 depends, mentions a first "accumulator." Nevertheless, I explained for claim 14 how *Kobayashi* discloses that the "device of claim 12" includes a precoder which comprises a first "accumulator." (*See* Section IX.A.9.)

discloses that concatenating multiple error correcting codes may allow a receiver to “achieve a higher performance” in identifying errors in the received sequence. (Ex. 1005, 1:26-48.) In my opinion, a person of ordinary skill in the art would have been motivated to add another accumulator to the *Kobayashi* transmitter because *Kobayashi* provides examples of only two types of encoders: a Hamming encoder, and an accumulator (the precoder), and thus there would be only a finite number of solutions to try when implementing a third encoder. (*See id.*, 7:46-8:27 (describing the Hamming code and accumulator).) Moreover, a POSITA would have been motivated to select the precoder as the additional encoder because it provides “simple transformation” that “prevent[s] possible error propagation.” *See id.* 7:30-45. A POSITA would have understood that an additional accumulator would provide thus the benefit of additional error correction performance without significantly increasing the complexity of the encoder or decoder.

123. In my opinion, a person of ordinary skill in the art would have had the skills to achieve the configuration, and similarly, had a reasonable expectation that such an implementation would have successfully performed as intended, particularly because *Kobayashi* discloses that such a modification would have been not only possible, but also beneficial because it would have rendered receiver performance benefits, as I have described above. Moreover, such a modification would have been



Declaration of Matthew C. Valenti, Ph.D., P.E.

U.S. Patent No. 7,421,032

a mere application of a known element (accumulator) to a known device (*Kobayashi's* transmitter with a Hamming code and accumulator) to yield predictable results (provide a concatenated system with an outer encoder comprising a Hamming code and an inner encoder comprising two accumulators).

**C. *Kobayashi* in View of *McEliece* Discloses and/or Suggests the Features of Claims 6, 13, and 18-22**

**1. Claim 6**

- a) The method of claim 5, wherein generating the random sequence of bits comprises coding the collection of message bits using a low-density generator matrix (LDGM) coder.**

124. In my opinion, *Kobayashi* in view of *McEliece* discloses or suggests these limitations. As I previously discussed for claim 5(b), *Kobayashi* discloses “generating the random sequence of bits” via a (7, 4) Hamming code and an interleaver. (See Section IX.A.4(b); see also Sections V.A (background on error control coding), V.B (background on matrix multiplication).) It is my opinion that a person of ordinary skill in the art would have further been motivated to modify *Kobayashi* in view of *McEliece* to implement this encoding step using a *low-density* generator matrix transformation.

125. As I previously explained for claim 1(a), *Kobayashi*’s concatenated system includes a receiver side. (Section IX.A.1(a).) *Kobayashi* discloses that its receiver is a decoder (shown below) which decodes the output of the second encoding operation (*i.e.*, the precoder). (See Ex. 1005, 6:16-7:4 (describing the general decoding technique), 7:28-11:17 (describing the Figure 8 implementation of the general decoding technique).)

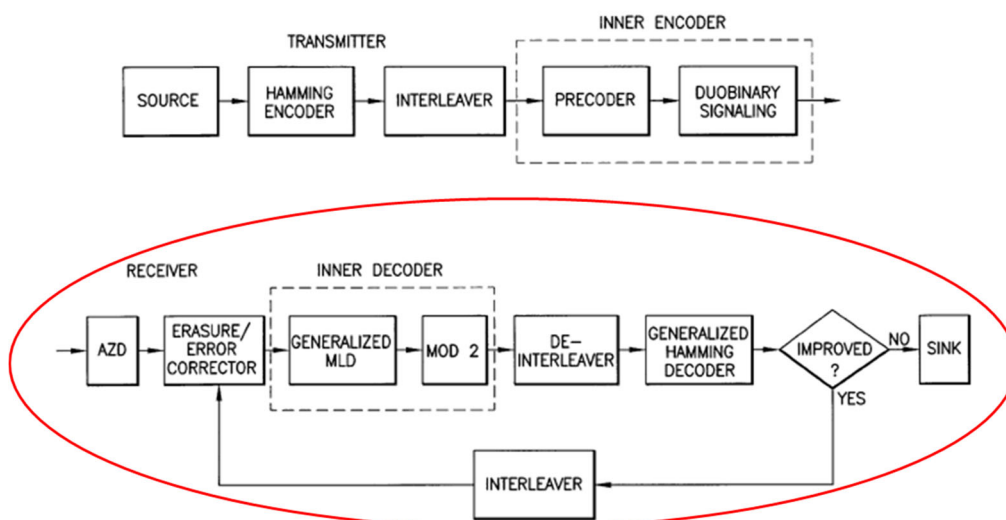


FIG.8

(*Id.*, FIG. 8 (annotated).)

126. *McEliece*, however, discloses a decoding algorithm called “Pearl’s belief propagation algorithm,” which is an algorithm that “solves the probabilistic inference problem in many important special cases.” (Ex. 1006, 144.) *McEliece* describes the probabilistic inference problem as follows:

[L]et  $X = \{X_1, X_2, \dots, X_N\}^5$  be a set of  $N$  discrete variables, where  $X_i$  assumes values in the finite alphabet  $A_i$ . The joint density function

$$p(x) = p(x_1, x_2, \dots, x_N) \\ \stackrel{\text{def}}{=} \Pr\{X_1 = x_1, \dots, X_N = x_N\}$$

is then a mapping from  $A_1 \times \dots \times A_N$  into the set of real numbers  $R$ . We assume that the marginal densities  $p(x_i) \stackrel{\text{def}}{=} \Pr\{X_i = x_i\}$  are also known. The marginal

density function  $p(x_i)$  represents our *a priori* “belief” about the random variable  $X_i$ . Now, suppose that one or more of these random variables is measured or “observed.” This means that there is a subset  $J \subseteq \{1, 2, \dots, N\}$  (the evidence set) such that, for all  $j \in J$ , the random variable  $X_j$  is known to have a particular value, say  $a_j$ . The *evidence* is then defined to be the event

$$\mathcal{E} = \{X_j = a_j : j \in J\}.$$

The fundamental *probabilistic inference problem* is to compute the *updated beliefs*, i.e., the *a posteriori* or conditional probabilities  $p(X_i|\mathcal{E})$ , for all  $i \notin J$ .

(*Id.* (emphasis in original).)

127. *McEliece* discloses that a “‘Bayesian belief network’ approach . . . to the inference problem is to exploit any ‘partial independencies’ which may exist among the  $X_i$ ’s to simplify belief updating.” (*Id.*) These “partial independencies can be described by a *directed acyclic graph*, or DAG,” which is “a finite, directed graph, in which there are no directed cycles.” (*Id.* (emphasis in original).) Suppose “ $G$  is a DAG, and . . .  $X$  is a set of random variables in one-to-one correspondence with the vertices of  $G$ .” (*Id.* (emphasis in original).) Additionally, suppose that if  $G$  has a “directed edge  $a \rightarrow b$ , then  $a$  will be called a ‘parent’ of  $b$ , and  $b$  will be called a ‘child’ of  $a$ .” (*Id.*) Then a “DAG, together with the associated random variables  $X$ ,

is called a *Bayesian belief network*, or *Bayesian network* for short.” (*Id.*, 145 (emphasis in original).)

128. *McEliece* discloses that “Bayesian networks can sometimes lead to considerable simplifications of the probabilistic inference problem,” such as “Pearl’s *belief propagation* algorithm.” (*Id.* (emphasis in original).) *McEliece* further discloses that “Pearl’s belief propagation algorithm is a decentralized ‘message-passing’ algorithm, in which there is a processor associated with each vertex of  $G$ ” and “[e]ach processor can communicate only with its parents and children.” (*Id.*, 146.) The general principle underlying Pearl’s belief propagation algorithm is that “[w]hen a processor is activated, it ‘reads’ the messages received from each of its parents and children, updates its belief based on these messages, and then sends new messages back to its parents and children.” (*Id.*) *McEliece* discloses that processors may also update their beliefs in parallel. (*Id.*, 148.)

129. *McEliece* discloses advantageous uses of Pearl’s belief propagation algorithm, which can be “used to derive effective iterative decoding algorithms for a number of . . . error-control systems, including . . . serially concatenated codes.” (*Id.*, 140.) *McEliece* describes these belief propagation (“BP”) decoding algorithms as “remarkably effective” and notes that “belief propagation provides an attractive general method for devising low-complexity iterative decoding algorithms for

hybrid coded systems.” (*Id.*, 140.) In particular, *McEliece* discloses that “decoding algorithms [for serially concatenated codes] can be derived routinely from a BP viewpoint, using the network of Fig. 12.” (*Id.*, 149.)

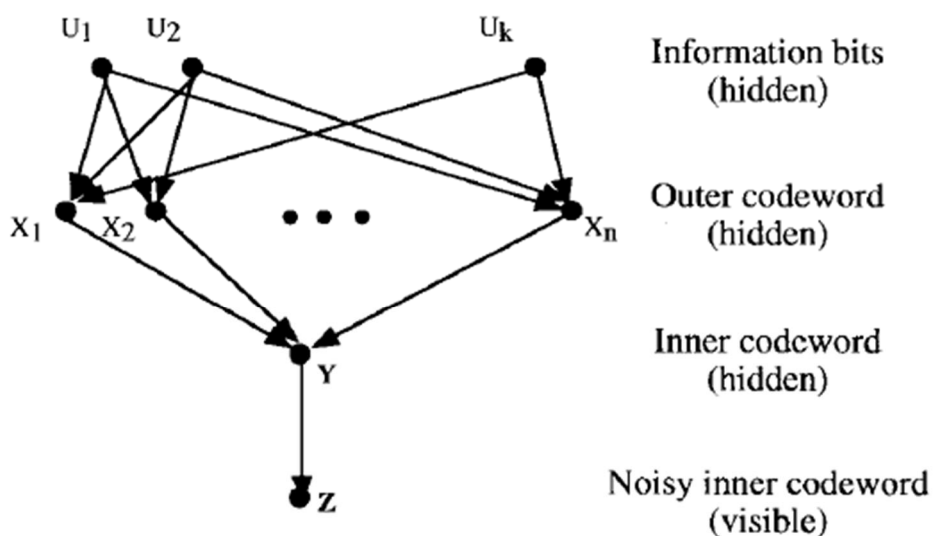


Fig. 12. Belief network for decoding a pair of serially concatenated codes.

(*Id.*, 150 (FIG. 12).)

130. In my opinion, a person of ordinary skill in the art would have therefore been motivated to modify the *Kobayashi* serially concatenated system to include a decoding technique similar to that disclosed by *McEliece*, rather than *Kobayashi*'s decoding technique, so as to take advantage of the benefits of belief propagation algorithms. Such a modification would have been within the skills and capabilities of a person of ordinary skill in the art, and given that *McEliece* discloses that such a modification would have been both possible and beneficial, a person of ordinary

skill in the art would have had a reasonable expectation of success in implementing it. Indeed, such a modification would have merely involved an application of a known element (belief propagation decoder) to a known device (*Kobayashi's* serially concatenated system with a decoder) to yield predictable results (provide a serially concatenated system with a belief propagation decoder).

131. *McEliece* further discloses that Pearl's belief propagation algorithm can be "used to derive effective iterative decoding algorithms for . . . the recently introduced low-density generator matrix codes." (*Id.*, 140.) In particular, *McEliece* discloses using "BP decoding on certain systematic linear block codes with low-density *generator matrices*," and discloses a small number of options for the low-density generator matrix, including classes of block codes constructed by *MacKay* and *Cheng*. (*Id.*, 149 (emphasis in original); *see also, e.g.*, Ex. 1007 ("*MacKay*"); Ex. 1011 ("*Cheng I*"), Ex. 1012 ("*Cheng II*").)<sup>13</sup> In my opinion, because *McEliece* discloses only a handful of options for the low-density generator matrix, a person of

---

<sup>13</sup> I refer to "*Cheng*" as the collective disclosures of *Cheng I* and *Cheng II* cited in *McEliece* to discuss the state of the art. *See* n.1. I cite to *Cheng I* and/or *Cheng II* for support of my discussion, as applicable.

ordinary skill in the art would have had reason to pursue a modification with any one of them, such as, for example, *MacKay*'s low-density generator matrix code.

132. The *MacKay* paper derives an algorithm for inference problems in modulo-2 arithmetic and applies it to decoding systematic linear block codes with low-density generator matrices. (Ex. 1007, 179, 192.) In particular, *MacKay* discloses the use of a random sparse generator matrix  $\mathbf{A}$  where "[t]he first  $N$  lines of  $\mathbf{A}$  were set to the identity matrix, and of the remaining bits, a fraction  $f_A$  were randomly set to 1." (*Id.*, 185-86.) *MacKay* discloses that this low-density generator matrix "can be viewed as defining a systematic error correcting code in which the signal  $s$  is transmitted, followed by  $(M - N)$  sparse parity checks." (*Id.*, 185.) *MacKay* demonstrates results for vector length  $N$  set to 50; number of measurements  $M$  set to 500, 1000, and 2000; density  $f_A$  varying from 0.05 to 0.50; and varying noise levels. (*Id.*, 185-86.) A POSITA would have been motivated to combine *Kobayashi* with *McEliece* because both relate to the design and implementation of linear, error-correcting codes (Ex. 1005, 1:6-11, 7:6-8:27; Ex. 1006, 3, 23.) A POSITA would have been motivated by the benefits of implementing a low-density generator matrix codes. As an example, due to their sparse matrices, LDGM have low-complexity encoders, especially when the desired coding rate is lower than  $1/2$ . This reduction of of complexity allows for implementations with fewer logic gates or software



operations than other codes available at the time of the alleged invention. Fewer logic gates reduces the cost of the semiconductor chips that implement the encoder. LDGMs are able to be encoded in linear time. (Ex. 1011 at 23.) LDGMs were known to have good error correction performance—within 1 dB of channel capacity. (Ex. 1011 at 23.) As discussed by Cheng, a POSITA would have known LDGMs had similar results as alternative high performance prior art codes, LDPC and turbo coding, but with the potential for lower complexity. *Id.* LDGMs can also be decoded in a distributed parallel method, which can decrease decoding time and increase efficiency, making them “more suitable for practical applications.” (Ex. 1011 at 23.)

133. *McEliece* discusses the benefits of using low-density generator matrix codes along with a belief propagation decoder, noting that “[t]he decoding algorithm[’s] . . . results were quite good, especially at high rates.” (Ex. 1006, 149.) Moreover, “these same ideas [were used] to construct a class of block codes which yield some remarkably efficient multilevel coded modulations.” (*Id.*)

134. Figure 11 below “shows the belief network for low-density generator matrix codes.” (*Id.*)

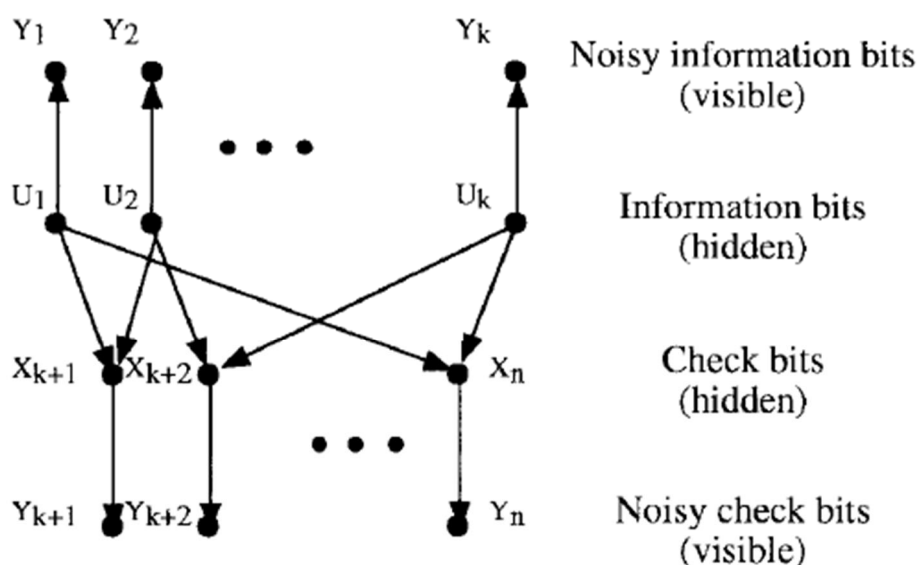


Fig. 11. Belief network for decoding systematic, low-density generator matrix codes.

(*Id.*, 149 (FIG. 11).)

135. In my opinion, a person of ordinary skill in the art would have recognized that the Hamming encoder of the *Kobayashi* method is a systematic linear block code with a generator matrix, and thus would have been motivated to further modify the *Kobayashi-McEliece* method to use a systematic linear block code with a *low-density* generator matrix instead. Such a modification would have enabled a person of ordinary skill in the art to take advantage of the benefits of LDGMs (discussed above in paragraph 132 and those disclosed by *McEliece*), and to further improve the concatenated system. This modification would have also been within the skills and capabilities of a person of ordinary skill in the art, and given

that *McEliece* discloses that such a modification would have been both possible and beneficial, for the reasons I have discussed above, a person of ordinary skill in the art would have had a reasonable expectation of success in implementing it. Indeed, such a modification because it would have merely involved an application of a known element (systematic linear block code with a low-density generator matrix) to a known device (*Kobayashi*'s method that uses a systematic linear block code (Hamming encoder) with a generator matrix as a first encoding step) to yield predictable results (*Kobayashi-McEliece*'s method that has a first encoding step comprised of a systematic linear block code with a low-density generator matrix).

136. In my opinion, a person of ordinary skill in the art would have understood that the *Kobayashi-McEliece* combination would have still generated a sequence of parity bits  $I_4$  in accordance with claims 1(c) and 5(b). (See Sections IX.A.1(c), IX.A.4(b).) In particular, the modified *Kobayashi* method would still operate according to the equation of claim 1(c), but with message bits irregularly repeated a different number of times, and with a different value of "a." (Section IX.A.1(c).)

137. A person of ordinary skill in the art would have understood that the *Kobayashi-McEliece* combination would have still had "irregular repeats of the message bits," see Section IX.A.1(c), and would have still "repeat[ed] each of the

message bits one or more times . . . wherein different fractions of the message bits are each repeated a different number of times and the number of repeats for each message bit is irregular,” *see* Section IX.A.4(b). This is because the low-density generator matrix disclosed by *MacKay* (and incorporated in *McEliece*) is randomly created, and thus the repetition of bits of data block  $I_1$  would still be irregular.

138. The number of times each message bit is repeated is ordinarily determined by the number of “1”s in each row when the output vector  $\mathbf{v}$  is determined from message bit vector  $\mathbf{u}$  and generator matrix  $\mathbf{G}$  such that  $\mathbf{v} = \mathbf{u} \cdot \mathbf{G}$ . However, *MacKay* describes determining  $\mathbf{v}$  such that  $\mathbf{v} = \mathbf{G} \cdot \mathbf{u}$ . (*See, e.g.*, Ex. 1007, 180 (depicting a (7, 4) Hamming code generator matrix that is a transpose of the generator matrix disclosed by *Kobayashi*), 185 (describing setting the first  $N$  rows, rather than columns, to the identity matrix).) Thus, a person of ordinary skill in the art would have understood that for *MacKay*’s low-density generator matrix, the number of times each message bit is repeated is determined by the number of “1”s in each *column* of the  $(M - N) \times N$  sub-matrix below the identity matrix that occupies the first  $N$  lines of generator matrix  $\mathbf{A}$ .<sup>14</sup> The probability that every single column

---

<sup>14</sup> In *MacKay* (Ex. 1007),  $N$  is the number of message bits and  $M$  is the number of code bits.

of the sub-matrix has the same number of “1”s is effectively zero for such large values of  $(M - N)$ . (See ¶ 132 (discussing how *MacKay* sets  $N$  to 50 and  $M$  to 500, 1000, and 2000).) Moreover, each column has at least one “1” (because of the identity submatrix in *MacKay*’s generator matrix), and thus each message bit is repeated at least once. Therefore, it is my opinion that the *Kobayashi-McEliece* combination would have still had “irregular repeats of the message bits,” see Section IX.A.1(c), and would have still “repeat[ed] each of the message bits one or more times . . . wherein different fractions of the message bits are each repeated a different number of times and the number of repeats for each message bit is irregular,” see Section IX.A.4(b).

139. A person of ordinary skill in the art would have further understood that the *Kobayashi-McEliece* combination would have still met the equation of claim 1(c), but because the low-density generator matrix disclosed by *McEliece* and *MacKay* is randomly created, the value of “ $a$ ” would vary. A person of ordinary skill in the art would have recognized that the value of “ $a$ ” would have ordinarily been determined by the number of “1”s in each column when the output vector  $\mathbf{v}$  is determined from information bit vector  $\mathbf{u}$  and generator matrix  $\mathbf{G}$  such that  $\mathbf{v} = \mathbf{u} \cdot \mathbf{G}$ . However, as I described earlier, *MacKay* describes determining  $\mathbf{v}$  such that  $\mathbf{v} = \mathbf{G} \cdot \mathbf{u}$ . A person of ordinary skill in the art would have thus understood that for *MacKay*’s

low-density generator matrix, the value of “a” is determined by the number of “1”s in each *row* of the  $(M - N) \times N$  sub-matrix below the identity matrix that occupies the first  $N$  lines of generator matrix **A**. The probability that every single row of the sub-matrix has the same number of “1”s (*i.e.*, has the same value of “a”) is effectively zero for such large values of  $(M - N)$ . (See ¶ 132 (discussing how *MacKay* sets  $N$  to 50 and  $M$  to 500, 1000, and 2000).) Thus, the *Kobayashi-McEliece* combination would have still met the equation of claim 1(c), but the value of “a” would vary.

140. Furthermore, for claim 5(b), it is my opinion that the *Kobayashi-McEliece* combination would have still “generat[ed] a random sequence of bits . . . with the repeats of the message bits being distributed in a random sequence” because the interleaver operation would still be applied. (Section IX.A.4(b).)

141. Thus, for the reasons I have described above, it is my opinion that the *Kobayashi-McEliece* combination discloses or suggests “generating the random sequence of bits comprises coding the collection of message bits using a low-density generator matrix (LDGM) coder.”

## **2. Claim 13**

- a) The device of claim 11, wherein the encoder comprises:**
- b) a low-density generator matrix (LDGM) coder**

**configured to perform an irregular repeat on message bits having a first sequence in a source data stream to output a random sequence of repeats of the message bits; and**

142. In my opinion, the *Kobayashi-McEliece* combination discloses or suggests these limitations. As I discussed above for claim 6, in my opinion, a person of ordinary skill would have been motivated to implement the *Kobayashi* transmitter using a “low-density generator matrix (LDGM) coder,” such as that disclosed by *McEliece*, “to perform an irregular repeat on [the] message bits” of sequence  $I_1$  and “output a random sequence of repeats of the message bits.” (Section IX.C.1.) As I also discussed for claim 1(b), the “message bits hav[e] a first sequence in a source data stream.” (Section IX.A.1(b).)

**c) an accumulator configured to XOR sum in linear sequential fashion a predecessor parity bit and “a” bits of the random sequence of repeats of the message bits.**

143. In my opinion, the *Kobayashi-McEliece* combination discloses or suggests these limitations for the same reasons as I previously explained for claim 13(c) (addressing the modified *Kobayashi* method). (Section IX.B.5(c).)

### **3. Claim 18**

**a) A device comprising:**

144. I understand that “[a] device comprising:” is the preamble of claim 18. I have been asked to assume that the preamble is limiting. In my opinion, *Kobayashi*

discloses the limitations of the preamble. For example, *Kobayashi* discloses a concatenated system with both transmitter and receiver portions, wherein the receiver is “[a] device,” as claimed, which performs the iterative decoding technique shown in Figure 8 below. (Ex. 1005, 6:16-7:4 (describing the general iterative decoding technique), 7:28-11:17 (describing the Figure 8 implementation of the general iterative decoding technique); *see also* Section V.A (background on error control coding).)

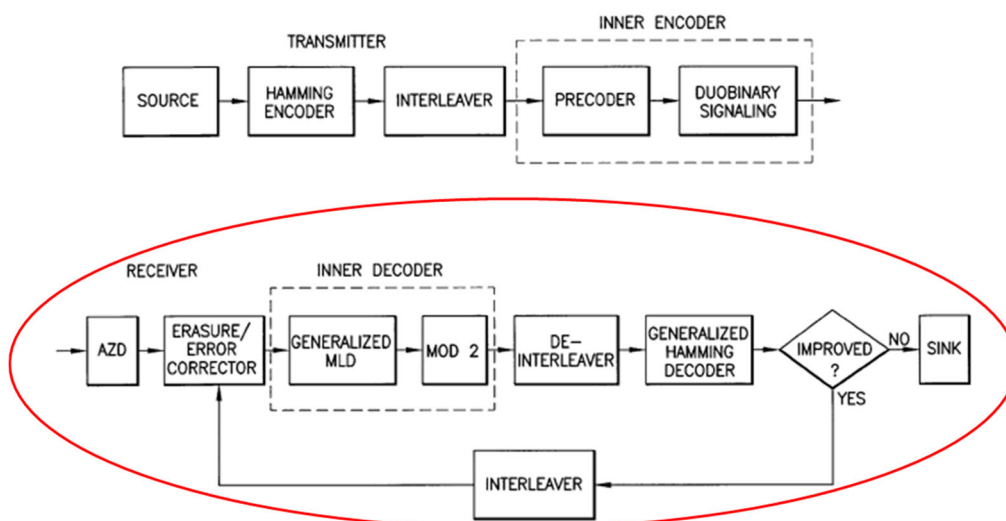


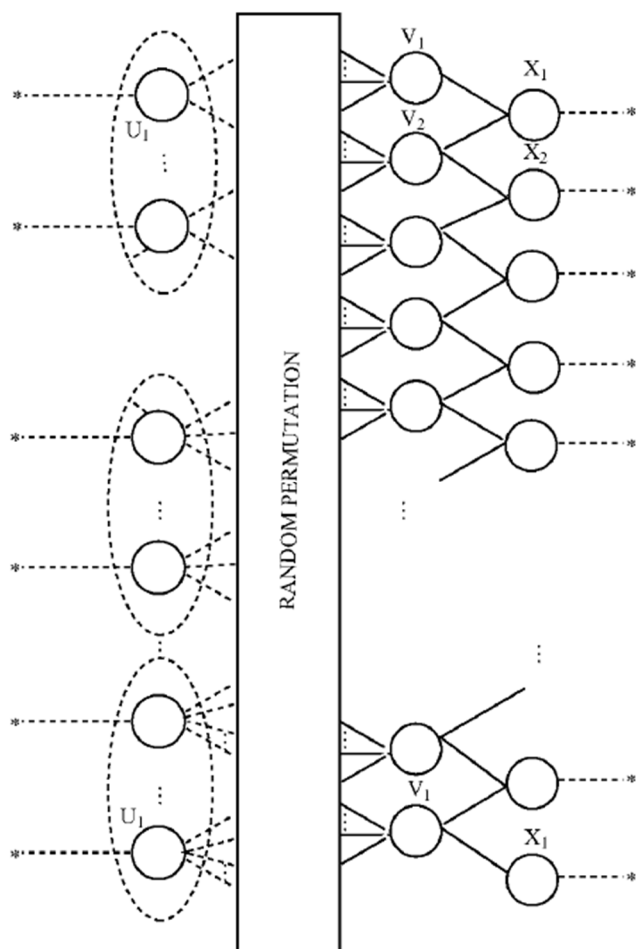
FIG.8

(Ex. 1005, FIG. 8 (annotated).)

- b) a message passing decoder configured to decode a received data stream that includes a collection of parity bits, the message passing decoder comprising two or more check/variable nodes operating in parallel to receive messages from neighboring check/variable nodes and send updated messages to



**the neighboring variable/check nodes, wherein the message passing decoder is configured to decode the received data stream that has been encoded in accordance with the following Tanner graph:<sup>15</sup>**



---

<sup>15</sup> I understand that the Certificate of Correction appended to the '032 patent states that the bottom  $U_1$  of the Tanner graph should read  $U_k$ , the bottom  $V_1$  should read  $V_r$ , and the bottom  $X_1$  should read  $X_r$ . (See Ex. 1001.) My analysis in this declaration reflects this corrected graph.

145. In my opinion, the *Kobayashi-McEliece* combination discloses or suggests these limitations. As I discussed for claim 6 (addressing the *Kobayashi-McEliece* combination), a person of ordinary skill in the art would have been motivated to modify the *Kobayashi* coding system so that the *Kobayashi* decoder discussed for claims 6 and 18(a) is replaced with a belief propagation decoder similar to that disclosed by *McEliece*. (See Section IX.C.1, ¶¶ 124-130 (discussing how a person of ordinary skill in the art would have been motivated to modify the *Kobayashi* decoder in view of *McEliece*<sup>16</sup>); see also Section IX.C.3(a).)

146. As I discussed for claim 6, the decoder of the *Kobayashi-McEliece* system is a “message passing decoder” because it uses Pearl’s belief propagation algorithm, which is described as a decentralized, message-passing algorithm. (Section IX.C.1.) The *Kobayashi-McEliece* decoder is also “configured to decode a received data stream that includes a collection of parity bits” because it is configured to decode the data stream sent by *Kobayashi*’s transmitter, which as I previously discussed for claims 1(d) and 8, transmits a stream to the receiver device that

---

<sup>16</sup> While I discuss in Section IX.C.1 a second modification to *Kobayashi* in view of *McEliece* to implement a low-density generator matrix coder (see Section IX.C.1, ¶¶ 131-141), such a modification is not necessary for claim 18(b).

includes a collection of parity bits. (Section IX.C.1; *see also* Sections IX.A.1(d), IX.A.6.)

147. As I discussed for claim 6 (addressing the *Kobayashi-McEliece* combination), the “message passing decoder compris[es] two or more check/variable nodes operating in parallel to receive messages from neighboring check/variable nodes and send updated messages to the neighboring variable/check nodes” because *McEliece*’s belief propagation decoder may have multiple nodes/processors operating in parallel to receive messages from neighboring nodes (parents and children), update the belief based on the messages, and send new messages to neighboring nodes (parents and children). (Section IX.C.1.)

148. In my opinion, a person of ordinary skill in the art would have recognized that the *Kobayashi-McEliece* decoder is also “configured to decode the received data stream that has been encoded in accordance with the . . . Tanner graph” above because it is configured to decode the data stream sent by *Kobayashi*’s transmitter, which as I previously discussed for claim 11(b), encodes the message bits in accordance with the same Tanner graph. (Section IX.C.1; *see also* Section IX.A.7(b).)

149. I have been asked to consider a scenario in which the *Kobayashi-McEliece* decoder, with the modifications I have already described, is not

“configured to decode the received data stream that has been encoded in accordance with the . . . Tanner graph.” Under that scenario, it is my opinion that a person of ordinary skill in the art would have been further motivated to implement the *Kobayashi-McEliece* combination such that its decoder decodes based on a Tanner graph representation of the encoder output, which is received by the decoder.

150. *McEliece* discloses that its belief propagation decoding algorithm for serially concatenated codes can decode an encoded data stream based on the belief network shown in Figure 12, reproduced below. (Ex. 1006, 149.)

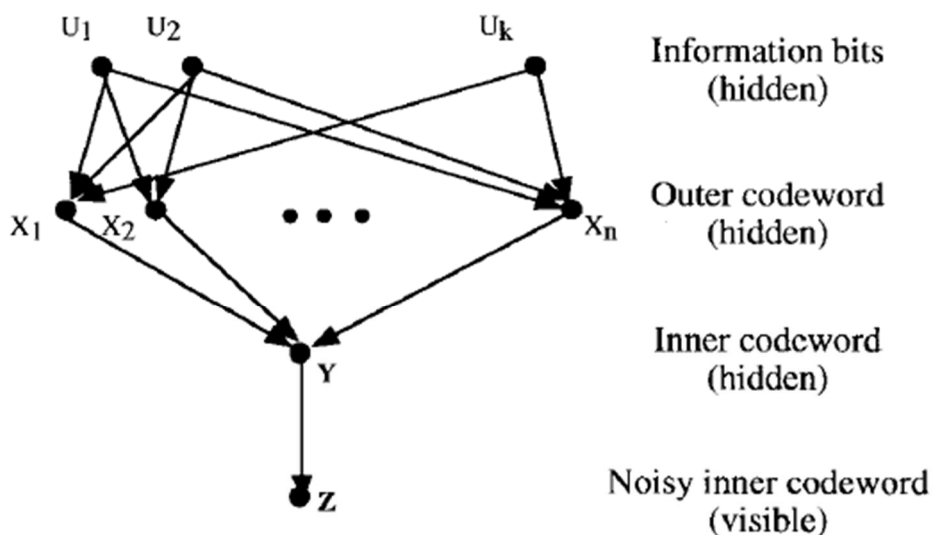


Fig. 12. Belief network for decoding a pair of serially concatenated codes.

(*Id.*, 150 (FIG. 12).)

151. *McEliece* discloses that its other belief network diagrams are comparable to Tanner graph representations. (Ex. 1006, 145 n.9 (“Our Fig. 7 [belief

network] should be compared to Wiberg [67, Fig. 2.5], which describes the ‘Tanner graph’ of a turbo code.”.) *McEliece* also discloses that, in the context of several variations on the belief propagation algorithm, both belief networks and Tanner graphs can be successfully used to yield these algorithm variations.

If we were to apply Pearl’s algorithm to the belief diagram of Fig. 13, we would obtain an iterative decoding algorithm for the tail-biting code. . . . Wiberg [67] has applied his algorithm to the Tanner graph of a tail-biting code with good success, and functionally, these two approaches yield virtually identical algorithms.

(*Id.*, 150.)

In this connection, we should note that Wiberg [67] has observed that his algorithm, when applied to a Tanner graph similar to Fig. 13 (less the tail-biting edge), also implies the BCJR algorithm.

(*Id.*, 150 n.10.)

[T]he BCJR algorithm . . . can be viewed as a kind of belief propagation.

(*Id.*, 150.)

152. In my opinion, a person of ordinary skill in the art would have been motivated to modify the *Kobayashi-McEliece* coding system’s decoder such that it

decodes based on a Tanner graph representation, rather than a belief network representation, of Figure 12. Decoding based on a Tanner graph representation was well known to a person of ordinary skill in the art at the time of the invention, and *McEliece* discloses decoding using Tanner graph representations as a viable option for use with belief propagation decoding. In my opinion, a person of ordinary skill in the art would have also had the knowledge and skills to successfully implement such a modification. Moreover, a person of ordinary skill in the art would have had a reasonable expectation of success in achieving such a modification because it would have merely involved an application of a known method (belief propagation decoding based on a Tanner graph representation) to a known device (the *Kobayashi-McEliece* coding system's belief propagation decoder) to yield predictable results (provide a belief propagation decoder that decodes based on a Tanner graph representation).

153. Thus, it is my opinion that the *Kobayashi-McEliece* combination discloses or suggests “a message passing decoder configured to decode a received data stream that includes a collection of parity bits, the message passing decoder comprising two or more check/variable nodes operating in parallel to receive messages from neighboring check/variable nodes and send updated messages to the neighboring variable/check nodes, wherein the message passing decoder is

configured to decode the received data stream that has been encoded in accordance with the . . . Tanner graph” depicted above.

#### **4. Claim 19**

- a) The device of claim 18, wherein the message passing decoder is configured to decode the received data stream that includes the message bits.**

154. In my opinion, the *Kobayashi-McEliece* combination discloses these limitations. As I discussed for claims 6 and 18(b) (addressing the *Kobayashi-McEliece* combination), a person of ordinary skill in the art would have been motivated to implement the *Kobayashi-McEliece* device such that the “message passing decoder is configured to decode the received data stream” from *Kobayashi*’s transmitter. (Sections IX.C.1, IX.C.3(b).) In my opinion, a person of ordinary skill in the art would have further been motivated to implement *Kobayashi*’s transmitter (and thus the *Kobayashi-McEliece* decoder) such that the “received data stream . . . includes the message bits” for the same reasons as I explained for claim 2. (Section IX.B.1.) As I discussed for claim 2, a person of ordinary skill in the art would have been motivated to implement the *Kobayashi* transmitter using a systematic code. (Section IX.B.1.) Thus, in my opinion, the *Kobayashi-McEliece* combination discloses or suggests that “the message passing decoder is configured to decode the received data stream that includes the message bits.”

**5. Claim 20**

- a) The device of claim 18, wherein the message passing decoder is configured to decode the received data stream as if a number of inputs into nodes  $v_i$  was not constant.**

155. In my opinion, the *Kobayashi-McEliece* combination discloses these limitations. As I discussed for claims 6 and 18(b) (addressing the *Kobayashi-McEliece* combination), it is my opinion that a person of ordinary skill in the art would have been motivated to implement the *Kobayashi-McEliece* device such that the “message passing decoder is configured to decode the received data stream” from *Kobayashi*’s transmitter. (Sections IX.C.1, IX.C.3(b).) As I further discussed for claim 12, the *Kobayashi* transmitter is configured to encode message bits “as if a number of inputs into nodes  $v_i$  was not constant.” (Section IX.A.8.) Thus, in my opinion, the *Kobayashi-McEliece* combination discloses or suggests that “the message passing decoder is configured to decode the received data stream as if a number of inputs into nodes  $v_i$  was not constant.”

**6. Claim 21**

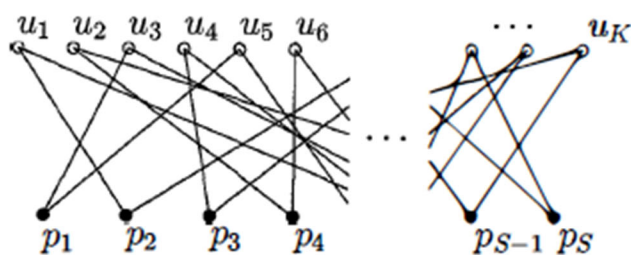
- a) The device of claim 18, wherein the message passing decoder is configured to decode in linear time at rates that approach a capacity of a channel.**

156. In my opinion, the *Kobayashi-McEliece* combination discloses or suggests these limitations. As I discussed for claim 13 (addressing the *Kobayashi-*



*McEliece* combination), a person of ordinary skill in the art would have been motivated to implement a second modification to *Kobayashi* in view of *McEliece* to implement a low-density generator matrix coder. (Section IX.C.1.) As further discussed, *McEliece* discloses a small number of options for the low-density generator matrix, including classes of block codes constructed by *MacKay* (Ex. 1007) and *Cheng* (Ex. 1011, Ex. 1012), and thus in my opinion, a person of ordinary skill in the art would have had good reason to pursue a modification with any of the solutions. (Section IX.C.1.) Thus, a person of ordinary skill in the art would have had good reason to pursue a modification with *Cheng*'s low-density generator matrix code. (See Ex. 1006, 149 (citing Ex. 1011 (*Cheng I*), Ex. 1012 (*Cheng II*)).)

157. *Cheng* discloses a class of low-density generator matrix codes whose corresponding Bayesian belief networks, such as shown below, may be decoded using a belief propagation algorithm. (Ex. 1012, 66.)



**Figure 5.2** A random regular bipartite belief network.

(*Id.*, 66 (FIG. 5.2).) *Cheng* discloses that the random regular bipartite graph has “two partite sets as  $K$  and  $S$ ,” which are called “*systematic* nodes and *parity* nodes,” and the vertices are labeled as “ $u_1, u_2, \dots, u_k$  and  $p_1, p_2, \dots, p_s$ , respectively, as shown in Fig. 5.2.” (*Id.*, 67.) *Cheng* describes “ $G = [I \mid P]$ ” as the “corresponding generator matrix of the new code, where  $I$  is a  $K \times K$  identity matrix and  $P$  is a  $K \times S$  matrix.” (*Id.*) Because “[t]he entry  $P_{ij}$  of the matrix  $P$  will be 1 if and only if there is an edge connecting  $u_i$  and  $p_j$ , and 0 otherwise,” the “class of codes is termed low-density generator matrix (LDGM) codes since  $G$  is sparse by this construction.” (*Id.*) *Cheng* discloses that its class of LDGM codes encompasses both regular and irregular belief networks. (*Id.*, 67.) Since any given code is either a regular code or an irregular code and Kobayashi is an irregular code, a POSITA would have been motivated to try the irregular type disclosed by *Cheng*.

158. As I discussed for claim 18(b), a person of ordinary skill in the art would have been motivated to modify the *Kobayashi-McEliece* coding system’s decoder such that it decodes based on a Tanner graph representation, rather than a belief network representation, of an encoding mechanism. (Section IX.C.3(b).) Because *McEliece* and *Cheng* disclose belief network representations of LDGM codes, a person of ordinary skill in the art would have therefore recognized that the *Kobayashi-McEliece* decoder, when implemented using *Cheng*’s LDGM (as taught

by *McEliece*) and the belief propagation decoder disclosed by *McEliece* (see Section IX.C.1) would still meet the limitations of claim 18 for the reasons I previously discussed for claim 18(b). (Section IX.C.3(b).)

159. *Cheng* further discloses that its LDGM codes “can be encoded and decoded with complexity linear in the code length,” and that “[t]he performances of these codes . . . are [within] 1 dB or so of the channel capacity.” (Ex. 1011, 522; see also Ex. 1012, 65 (disclosing that *Cheng*’s class of LDGM codes “can be encoded and decoded in linear time” and “the performance of these new systems . . . approach the capacity of the Gaussian channel”).) In my opinion, a person of ordinary skill in the art would have therefore understood that the *Kobayashi-McEliece* message passing decoder is also “configured to decode in linear time at rates that approach a capacity of a channel.”

160. Thus, in my opinion, the *Kobayashi-McEliece* combination discloses or suggests that “the message passing decoder is configured to decode in linear time at rates that approach a capacity of a channel.”

## **7. Claim 22**

### **a) The device of claim 18, wherein the message passing decoder comprises a belief propagation decoder.**

161. In my opinion, the *Kobayashi-McEliece* combination discloses or suggests these limitations. As I discussed for claim 18(b), the message passing

Declaration of Matthew C. Valenti, Ph.D., P.E.

U.S. Patent No. 7,421,032

decoder of the *Kobayashi-McEliece* combination is a belief propagation decoder.

(Section IX.C.3(b).)

## **X. CONCLUSION**

162. I declare that all statements made herein of my knowledge are true, and that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: October 31, 2022

By: Matthew C. Valenti

Matthew C. Valenti, Ph.D., P.E.