11 APR 2003

INVERTIBLY MAPPING BINARY SEQUENCES INTO RATE 2/3 (1,K) RUN-LENGTH-LIMITED CODED SEQUENCES WITH MAXIMUM TRANSITION DENSITY CONSTRAINTS

Field of the Invention

This invention relates to anomalous decoding of sequences of binary values such as may be found in magnetic storage recording channels or in optical communications subsystems where the errors, erasures, or faults involve binary subsequences exhibiting high-duty-cycle patterns or selected pattern repetitions. More particularly, the invention relates to methods and means for either preventing indefinite repetition of the patterns or excluding them.

166166

Background of the Invention

It is well known that a binary-valued sequence, such as 1000011011, may be adversely interpreted by decoding devices due to unintended patterns or attributes present in such sequences. For example, a sequence such as 1010101010... exhibits a very high duty cycle. Electronically, a high duty cycle is synonymous with electrical or mechanical elements being repeatedly stressed. This frequently results in attendant thermal and noise increases in component environments resulting in higher error rates, failure rates, and shortened component lives.

The term "duty cycle" for purposes of this specification means the number of binary 1's occurring in a pattern or repetitive subsequence interval. In the 101010 example, the duty cycle is 50 percent since the "1" occurs in every two-bit interval. If the repetitive pattern was of the form of two binary 4-bit words such as 1001,1000, the duty cycle would be 3/8=37.5 percent.

In magnetic recording channels and optical communications there arises the phenomenon of pulse smearing or broadening.. For instance, two binary 4-bit words 0001, 0010 might, as a result of "smearing", appear at the input of the decoder as 0001,1010. In this case, the binary 1 in the 4th-bit position of the first word is electrically or optically "stretched" to appear as if it were two consecutive binary 1's. Such studies or smearing is especially egregious in pulse position modulation (PPM) communications systems or the like. This devolves from the fact that PPM systems are notoriously bandwidth inefficient.

It is known from Adler et al., U.S. Patent 4,413, 251, "Method and Apparatus for Generating a Noiseless Sliding Block Code for a (1,7) Channel with Rate 2/3", issued November 1, 1983, that a finite state

machine (FSM) can convert unconstrained binary-valued sequences into a constrained binary-valued sequence in an invertible manner. Furthermore, Adler teaches that a finite lookahead state-independent machine could perform the decoding. The lookahead capability permits a decoder to resolve n<m bits of a current RLL codeword into m bits of the unconstrained sequence by taking into account a predetermined number of subsequent RLL codewords. Nevertheless, this decoder lookahead feature increases the adverse effect of error or erasure in the RLL codeword.

According to Adler et al., for a given code rate R=m/n of mapping m bits of unconstrained binary sequence into n bits of constrained sequence, the requirement is partially satisfied by deriving an FSM encoder with 2^m branches per state in which the (d, k) constraints are manifested by splitting and merging some of the FSM states in order to obtain a new FSM. The (d, k) constraint means that at least d"0's" and no more than k" 0's" are to be inserted between any pair of consecutive binary "1's". For d<k, d determines the frequency of transitions and whence the intersymbol interference (ISI). In the case of k, it is used for clock resynchronization.

Disclosure of the Invention

According to a first aspect of the invention, there is provided a method for invertibly mapping binary sequences into rate 2/3 (1, k) run-length-limited coded (RLL) sequences with maximum transition density constraints comprising the steps of:

(a) defining and storing in a processor a state transition table of ordered pairs including a next state ($n_1n_2n_3$) and a current RLL-coded tri-bit symbol ($c_1c_2c_3$), each ordered pair in said table being indexed in a first tabular dimension according to its present state and in a second tabular dimension according to a vector ($b_1b_2b_3b_4$) of a present (b_1b_2) and a predetermined number of lookahead (b_3b_4) bit-pairs from the binary sequence, each present bit-pair being a cognizable binary value (00, 01, 10, 11), each of the predetermined number of lookahead bit-pairs being selected from a set consisting of a cognizable binary value and a don't care (xx) value, the state-to-state transitions being constrained such that any counterpart long-run sequence of RLL-coded tri-bits manifests a duty cycle less than 50 percent, the said (1, k) RLL code constraints being selected from a set consisting of (1, 9) and (1, 13), the respective state transition table being selected from:

11 APR 2003

(1, 9) RLL code	Next State/Internal Output N = ($n_1n_2n_3$) / c = ($c_1c_2c_3$) Internal Inputs $b_1b_2b_3b_4$ x = don't care bit value						
Internal State S=$s_1s_2s_3$	00xx	01xx	10xx	1100	1101	1110	1111
000	000/010	001/010	010/010	111/001	111/001	011/010	011/010
001	000/001	001/001	100/010	100/010	100/010	100/010	100/010
010	000/100	001/100	010/100	011/100	011/100	011/100	011/100
011	000/101	001/101	100/101	100/100	100/100	100/100	100/100
100	000/000	001/000	010/000	011/000	011/000	011/000	011/000
111	100/000	100/000	100/000	100/000	100/000	100/000	100/000

or

(1, 13) RLL code	Next State/Internal Output N = ($n_1n_2n_3$) / c = ($c_1c_2c_3$) Internal Inputs $b_1b_2b_3b_4b_5b_6$ x = don't care bit value							
Internal State S=$s_1s_2s_3$	00xxxx	01xxxx	10xxxx	1100xx	1101xx	111011	1110'(11)	1111xx
000	000/010	001/010	010/010	111/001	111/001	111/010	111/010	011/010
001	000/001	001/001	100/010	100/010	100/010	100/010	100/010	100/010
010	000/100	001/100	010/100	011/100	011/100	111/100	011/100	011/100
011	000/101	001/101	100/101	100/100	100/100	100/100	100/100	100/100
100	000/000	001/000	010/000	011/000	011/000	011/000	011/000	011/000
111	100/000	100/000	100/000	100/000	100/000	100/000	100/000	100/000

(b) causing said processor to access the table responsive to a succession of vectors of bit-pairs and extracting therefrom a succession of RLL-coded tri-bits ($c_1c_2c_3$) and a succession of the internal state ($n_1n_2n_3$); and

(c) causing said processor to access the table responsive to a succession of RLL-coded tri-bits ($c_1c_2c_3$) and extracting therefrom a succession of binary sequence bit-pairs.

11 APR 2003

According to a second aspect of the invention, there is provided an apparatus for invertibly mapping binary sequences into rate 2/3 (1, 9) run-length-limited coded (RLL) sequences with maximum transition density constraints comprising:

a finite state machine (FSM) for generating RLL-coded tri-bit symbols ($c_1c_2c_3$) as a first function of a present one of a plurality of FSM internal states ($s_1s_2s_3$) and a vector ($b_1b_2b_3b_4$) of a present (b_1b_2) and a predetermined number of lookahead (b_3b_4) bit-pairs from the binary sequence and for generating a next successor internal machine state ($n_1n_2n_3$) as a second function of the present FSM state and the binary sequence vector, each present bit-pair being a cognizable binary value (00, 01, 10, 11), each of the predetermined number of lookahead bit-pairs being selected from a set consisting of a cognizable binary value and a don't care (xx) value, said first and second functions being defined according to a plurality of Boolean valued relations such that:

$$n_1 = b_1S_3 + b_1b_2\,`b_3\,`s_1\,`s_2 + s_1s_3$$

$$n_2 = b_1\,`s_3$$

$$n_3 = b_2\,`s_3 + `s_1\,`b_1b_2$$

$$c_1 = `s_1s_2$$

$$c_2 = `s_1\,`s_2\,`c_3$$

$$c_3 = `s_1s_3(`b_1 + `b_2) + `s_1\,`s_2\,`s_3b_1b_2\,`b_3b_4.$$

According to a third aspect of the invention, there is provided a computer program directly loadable into the internal memory of a digital computer, comprising software code portions for performing the steps of the method described above.

Brief Description of the Drawings

Fig. 1 shows a constrained channel encoder and decoder positioned with respect to an information storage or transfer medium according to the invention;

Fig. 2 sets out a finite state machine (FSM) for use as an encoder illustrating the output as a function of the present internal state and a set of inputs, and the next internal state as another function of the present internal state and a set of inputs;

Fig. 3 depicts a state diagram (1, 7) rate 2/3 RLL encoder responsive to an unconstrained binary input producing a high-duty-cycle output where a state modification transforms the encoder to a (1, 9) rate 2/3 RLL encoder producing a low-duty-cycle output;

Figs. 4A-4C respectively illustrate a state transition table for a (1, 9) rate 2/3 RLL encoder modified according to the invention, Boolean equations relating to the unconstrained inputs to the constrained outputs, and a high-level logic gate embodiment of the encoder;

Figs. 5A-5C respectively show a decoding table for a (1, 9) rate 2/3 RLL decoder modified according to the invention, Boolean equations

relating constrained inputs to unconstrained outputs, and a high-level logic gate embodiment of the decoder;

Figs. 6A-6C respectively depict Boolean equations relating unconstrained inputs to constrained outputs for a (1,13) rate 2/3 RLL encoder modified according to the invention, a state transition table, and a high-level logic gate embodiment of the encoder; and

Figs. 7A-7C respectively set out a decoding table for a (1,13) rate 2/3 RLL decoder modified according to the invention, Boolean equations relating constrained inputs to unconstrained outputs, and a high-level logic gate embodiment of the decoder.

Detailed Description of the Invention

Referring now to Fig. 1, there is shown a constrained channel encoder 3 positioned before a recording or transmission channel 5 and an information storage or transfer medium 7. A counterpart decoder 11 is positioned between a readback or reception channel 9 and unconstrained binary sequence sink 13. The binary source 1 can be conceived as a random number generator of 1's and 0's having a uniform, Gaussian, or other statistical distribution and exhibiting from time to time repetition patterns having high duty cycles. These cycles range from 50 percent at the highest in the form of 10101010... etc. to sequences with more tolerable duty cycles significantly less than 50 percent. Since the general solution is to apply the unconstrained binary sequence to a (1,k) rate 2/3 RLL encoder, the encoder itself provides a duty cycle minimum.

The constrained binary output strings from encoder 3 is in turn applied to a recording or transmission channel 5 where it is suitably modulated and written out to either a storage or transmission medium 7. At a subsequent time, the constrained and modulated string is applied to a readback or reception channel 9 and converted into a constrained binary string of 1's and 0's. This converted binary string is then applied to a lookahead state-independent decoder 11. The decoder 11 utilizes a predetermined number of subsequent characters (lookahead) and removes the constraints, thereby providing the original unconstrained binary sequence to a sink 13.

Referring now to Fig. 2, there is shown a finite state machine (FSM) for use as an encoder 3. For a specific set of parameters of (d,k)=(1,9) or (1,13), a predetermined number of present input bits (b_1b_2) and future input bits (b_3b_4) are applied ($b_1b_2b_3b_4$) from an input register 21 to an FSM23 over path 22. The FSM23 provides an output to register 25. An output set of binary bits $C=(c_1c_2c_3)$ is derived as a function f of input

bits ($b_1b_2b_3b_4$) and present machine state $S=(s_1s_2s_3)$. Concurrently, the next machine stateN is determined as a functiong of input bits ($b_1b_2b_3b_4$) and present machine state S. Thus, for the next cycle, next machine stateN is written into the present state register 29 and becomes the new present machine state S. In this regard, an FSM can also be conceptualized as a form of directed graph whose nodes represent the FSM machine states S, and whose directed links define the state-to-state transitions for any given input vector while generating a counterpart output vector. The tabular form of the directed graph is termed a "state transition table".

Referring now to Fig. 3, there is depicted a partial-state diagram or directed graph of a (1,7) rate 2/3 RLL encoder responsive to an unconstrained binary input producing a high-duty-cycle output. Nodes 000, 001, 010, 011, and 100 represent the internal states of the (1,7) encoder. Since this is a rate 2/3 encoder, then for an input pattern of two bits at a time of 001100, the encoder output pattern would be three bits at a time of 010101010. Similarly, for the input pattern 001101, the output pattern would be 010101001. If an additional state 111 is added to the state diagram and connected between states 000 and 100 as shown, the encoder is converted to (d,k)=(1,9) rate 2/3 encoder. The additional state provides sufficient computational resource to lookahead over several input symbols and generates an output sequence of lowered duty cycle, namely, 010000000 for input 001100 and 001000000 for input 001101.

Referring now to Figs. 4A-4C, there is respectively shown a state transition table for a (1,9) rate 2/3 RLL encoder modified according to the invention, Boolean equations relating unconstrained binary inputs to highly constrained binary outputs, and a high-level logic gate embodiment of the encoder. In Fig.4A, the encoder is an FSM having six internal states S. While the encoder is rate 2/3, it is designed to be responsive to a present input bit-pair plus one lookahead input bit-pair $b=(b_1b_2b_3b_4)$. Each row is indexed by one of the six internal states S, while each column is indexed by one of seven inputs of two bit-pair combinations. For any given (S,b) table index, the entry is of the form (next state N/Output $C=(c_1c_2c_3)$). Of seven input two-bit-pair input combinations, three of them, namely, 00 xx, 01 xx, and 10 xx, have two "don't care" bit-valued positions. Illustratively, if the (1,9) encoder wherein state 100 in the input was 00 xx, then a transfer would be made from state 100 to state 000 and output C=000 would be generated.

Referring now to Fig. 4B, there is set out Boolean equations defining the next state Boolean function $N=(n_1n_2n_3)=g(S,b)$ and the output Boolean function $C=(c_1c_2c_3)=f(S,b)$. These equations are analytic expressions derived from the state transition table in Fig. 4A. In these

Boolean expressions, the negation of a Boolean variable C is signified by "'C". Thus, the negation of 111 would be exemplified by '(111). It should be noted that the Boolean equations in Fig. 4B completely specify the combinatorial AND/OR logic (415 in figure 4C).

Relatedly, logic for a low-duty-cycle (1,9) rate 2/3 RLL encoder is shown in Fig. 4C. In the logic embodiment, a present input bit-pair (b_1b_2) is applied via latches 407 and 403 to combinatorial logic 415. Similarly, a lookahead input bit-pair (b_3b_4) is applied to logic 415 via latches 405 and 401. The encoded output C is presented by logic 415 to latches 417, 419, and 421. The next internal state N is presented by logic 415 over feedback paths I, II, and III to respective input latches 409, 411, and 413.

Referring now to Figs. 5A-5C, there is respectively shown a decoding table for a (1,9) rate 2/3 RLL decoder modified according to the invention, Boolean equations relating constrained inputs back into unconstrained outputs, and a high-level logic gate embodiment of the decoder. In Fig. 5A, the decoding table while providing for lookahead over several RLL symbols is state independent. That is, it does not constitute an FSM in the sense of an encoder. The RLL binary-coded symbols designate the first three columns from left to right. These include a present symbol ($r_1r_2r_3$) and two lookahead symbols ($r_4r_5r_6$) and ($r_7r_8r_9$). The fourth column is designated as the "Decoded Output". For decoding purposes, the RLL symbol string comprises a sliding block of 3-bit symbols in which decoding of the present symbol occurs as the result of logically combining the present symbol with two lookahead symbols according to the Boolean equations set out in Fig. 5B.

Referring now to Fig. 5C, each RLL-coded symbol comprises a 3-bit byte, which bits are respectively applied to latches 501, 503, and 507. The bits are processed to form three intermediate internal variables z_1, z_2, and z_3 via OR gate 513 and latches 519 and 521. These are applied concurrently as inputs to the combinatorial logic 525. Additionally, the first and third bit values r_1 and r_3 of the present RLL-coded symbol are passed by way of respective latches 509, 517, 515, and 523 as concurred inputs to the logic 525.

Logic 525 is completely defined by the logic equations for the decoded output expressed in Fig. 5B. While the logic equations prescribed a 6-coded bit output ($u_1u_2u_3u_4u_5u_6$), the output variable u_6 is set equal to 0. In the embodiment shown, additional logic is provided to convert decoded output U into the 2-bit bytes of the original unconstrained binary sequence. Thus, a first one of the 2-bit bytes is computed by OR gate 529 actively combining u_4 and a version of u_6 as found in latch 527 and OR

gate 539 actively combining OR gate 529 output as found in latch 531 and the u_2 output from logic 525. A second one of the 2-bit bytes is computed by OR gate 535 actively combining u_1 and a version of u_3 as found in latch 533. The first one of the two bits is presented through latch 541, while the second one of the two bits is presented through latch 537.

Referring now to Figs. 6A-6C, there is respectively depicted the state transition table for a (1,13) rate 2/3 RLL encoder modified according to the invention, Boolean equations relating unconstrained inputs to constrained outputs, and a high-level logic gate embodiment of the encoder. Turning attention to Fig. 6B, there is shown the state transition diagram in the form of an array of ordered pairs of internal states and coded output as indexed by a present internal state S and a compound input ($b_1b_2b_3b_4b_5b_6$). The compound input in the case of a (1,13) encoder is a present coded binary input bit-pair and two of the lookahead coded binary input bit-pairs. If the encoder were subject to a continuous input of 00 xx xx, then it would ultimately transit to state 000 and cycle within the state producing a 010010010010... output for a duty cycle of 1/3. This is still substantially less than one-half associated with the output 010101....

The Boolean equations for the (1,13) encoder shown in Fig. 6A are very similar in gross structure to the equations set out for the (1,9) encoder as shown in Fig. 4B. That is, they are derived from the state transition diagram and are constrained to produce a low duty cycle of output. The equations also articulate an alternative representation for negation through the use of a horizontal bar above the Boolean variable.

Referring now to Fig. 6C, there is shown a logical embodiment of the (1,13) encoder equivalent to that of the logical embodiment of the (1,9) encoder shown in Fig. 4C, with the exception that the (1,13) encoder is depicted in detached contact form. Detached contact form was originally devised to represent relay circuit networks as used, for example, in the telephone industry. Three input bit-pair's d_1d_2 are applied to the latches L, the input bits $b_1..b_6$ together with the internal state are applied to the combinatorial logic in a manner substantially similar to that described in connection with the (1,9) encoder.

Referring now to Figs. 7A-7C, there is respectively set out a decoding table for a (1,13) rate 2/3 RLL decoder according to the invention, Boolean equations relating constrained inputs to unconstrained outputs, and a high-level logic embodiment of the decoder. In Fig. 7A, a state-independent decoding table is set out involving a present RLL symbol and three lookahead RLL symbols in order to effectuate mapping the present RLL symbol into unconstrained binary bit-pair sequences. Relatedly,

Boolean logic equations defining the decoder are set out in Fig. 7B and the detached contact logic embodiment in Fig. 7C, and operating in a manner as previously described in connection with the embodiment set up in Fig. 4C.

CLAIMS

1. A method for invertibly mapping binary sequences into rate 2/3 (l, k) run-length-limited coded (RLL) sequences with maximum transition density constraints comprising the steps of:

(a) defining and storing in a processor a state transition table of ordered pairs including a next state ($n_1n_2n_3$) and a current RLL-coded tri-bit symbol ($c_1c_2c_3$), each ordered pair in said table being indexed in a first tabular dimension according to its present state and in a second tabular dimension according to a vector ($b_1b_2b_3b_4$) of a present (b_1b_2) and a predetermined number of lookahead (b_3b_4) bit-pairs from the binary sequence, each present bit-pair being a cognizable binary value (00, 01, 10, 11), each of the predetermined number of lookahead bit-pairs being selected from a set consisting of a cognizable binary value and a don't care (xx) value, the state-to-state transitions being constrained such that any counterpart long-run sequence of RLL-coded tri-bits manifests a duty cycle less than 50 percent, the said (l, k) RLL code constraints being selected from a set consisting of (1, 9) and (1, 13), the respective state transition table being selected from:

(1, 9) RLL code	Next State/Internal Output N = ($n_1n_2n_3$) / c = ($c_1c_2c_3$) Internal Inputs $b_1b_2b_3b_4$ x = don't care bit value						
Internal State S=$s_1s_2s_3$	00xx	01xx	10xx	1100	1101	1110	1111
000	000/010	001/010	010/010	111/001	111/001	011/010	011/010
001	000/001	001/001	100/010	100/010	100/010	100/010	100/010
010	000/100	001/100	010/100	011/100	011/100	011/100	011/100
011	000/101	001/101	100/101	100/100	100/100	100/100	100/100
100	000/000	001/000	010/000	011/000	011/000	011/000	011/000
111	100/000	100/000	100/000	100/000	100/000	100/000	100/000

or

(1, 13) RLL code	Next State/Internal Output N = ($n_1n_2n_3$) / c = ($c_1c_2c_3$) Internal Inputs $b_1b_2b_3b_4b_5b_6$ x = don't care bit value							
Internal State S=$s_1s_2s_3$	00xxxx	01xxxx	10xxxx	1100xx	1101xx	111011	1110' (11)	1111xx
000	000/010	001/010	010/010	111/001	111/001	111/010	111/010	011/010
001	000/001	001/001	100/010	100/010	100/010	100/010	100/010	100/010
010	000/100	001/100	010/100	011/100	011/100	111/100	011/100	011/100
011	000/101	001/101	100/101	100/100	100/100	100/100	100/100	100/100
100	000/000	001/000	010/000	011/000	011/000	011/000	011/000	011/000
111	100/000	100/000	100/000	100/000	100/000	100/000	100/000	100/000

(b) causing said processor to access the table responsive to a succession of vectors of bit-pairs and extracting therefrom a succession of RLL-coded tri-bits ($c_1c_2c_3$) and a succession of the internal state ($n_1n_2n_3$); and

(c) causing said processor to access the table responsive to a succession of RLL-coded tri-bits ($c_1c_2c_3$) and extracting therefrom a succession of binary sequence bit-pairs.

2. A method as claimed in claim 1, wherein said duty cycle substantially approximates one-third, and further wherein said (1, k) RLL code constraints are ones selected from a set consisting of (1, 9), (1, 10) and (1, 13).

3. A method as claimed in claim 1, wherein said (1, k) RLL code constraints are ones selected from a set consisting of (1, 9) and (1, 13).

4. Apparatus for invertibly mapping binary sequences into rate 2/3 (1, 9) run-length-limited coded (RLL) sequences with maximum transition density constraints comprising:

a finite state machine (FSM) for generating RLL-coded tri-bit symbols ($c_1c_2c_3$) as a first function of a present one of a plurality of FSM internal states ($s_1s_2s_3$) and a vector ($b_1b_2b_3b_4$) of a present (b_1b_2) and a predetermined number of lookahead (b_3b_4) bit-

pairs from the binary sequence and for generating a next successor internal machine state ($n_1n_2n_3$) as a second function of the present FSM state and the binary sequence vector, each present bit-pair being a cognizable binary value (00, 01, 10, 11), each of the predetermined number of lookahead bit-pairs being selected from a set consisting of a cognizable binary value and a don't care (xx) value, said first and second functions being defined according to a plurality of Boolean valued relations such that:

$$n_1 = b_1S_3 + b_1b_2\grave{}b_3\grave{}s_1\grave{}s_2 + s_1s_3$$

$$n_2 = b_1\grave{}s_3$$

$$n_3 = b_2\grave{}s_3 + \grave{}s_1\grave{}b_1b_2$$

$$c_1 = \grave{}s_1s_2$$

$$c_2 = \grave{}s_1\grave{}s_2\grave{}c_3$$

$$c_3 = \grave{}s_1s_3(\grave{}b_1 + \grave{}b_2) + \grave{}s_1\grave{}s_2\grave{}s_3b_1b_2\grave{}b_3b_4.$$

5. A computer program directly loadable into the internal memory of a digital computer, comprising software code portions for performing the steps of any one of claim 1 to claim 3 when said product is run on a computer.