EURASIP Journal on Applied Signal Processing 2001:3, 181–188

A Class of High-Rate $N/(N+1)$ Low-Complexity Error-Detecting RLL Codes

Anthony G. Bessios[†]

Texas Instruments, Inc., Dallas, TX 75243, USA

Received 3 April 2000 and in revised form 24 August 2001

We present a new methodology for the construction of high-rate channel modulation run-length-limited RLL$(0, k)$ codes. Simple modulation encoders and decoders are constructed, with low error propagation during decoding. They combine partial error detection capability (PED) to boost the performance of a concatenated outer Error Correction Code (ECC) (Blaum, 1991). Moreover, current systems are using low redundancy ECC, and the overall rate is mainly determined by the inner modulation code rate, which critically is to be maintained high. Code rates $R_c = N/(N+1)$, for example, 16/17, 24/25 and higher are achievable, with efficiency exceeding 0.94 and 0.96, respectively. The proposed fixed length block decodable codes, are generalized schemes of the type $N/(N+1)$ $(d = 0, k = [N/2])$ for $N \geq 5$.

Keywords and phrases: run-length-limited constrained codes.

1. INTRODUCTION

New high-rate RLL block codes are proposed in this paper. These are $(d = 0, k)$ codes, where d and k denote, respectively, the minimum and maximum run-length of zeros between ones in an unprecoded channel data stream. There are several RLL codes with or without enhanced error control capabilities [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. The $(d = 0, k/I)$ *RLL* codes use gated-partition logic to achieve high rates such as 8/9 [1] and 16/17 [2], while focusing on the k, I (interleave) constraints. Their error detection capability is limited to the codewords non-supported by the code, as well as to the weak constraint of k, I violations. Furthermore, the block mapping size grows exponentially with the user data word length N. Concatenation of conventional RLL codes with ECC can reduce the effectiveness of the ECC, especially with a sliding block encoder/decoder subject to error propagation [3]. The *single-error correcting RLL* codes combine RLL with single-error correcting capability via an increase of the codewords' minimum distance $d_{\min}$, but with the adverse effect of a lower rate, such as 8/21, 8/28 [3], or 1/3 and 7/17 [4]. In [5], it was proposed that the Hamming subcode block length is kept as large as possible to avoid rate loss for a single-error correcting ECC/RLL code. In [6], redundancy based on appended parity bits is used, or access to channel-side information is possible and sufficiently long codewords are assumed, to construct high rate codes with single error correction capability. Single error detecting systematic RLL codes subject to rate loss are presented in [7], using m parity check bits to produce rates of $R_C = N/(N+1+m)$. Finally, in [8] the *error detecting* modulation codes with 3–4 times larger block length than conventional RLL of the same rate, alleviate the code rate overhead due to the appended parity, but they increase the system's probability of error. In [8], code rate reduction is avoided by choosing the odd or the even sequences only, whichever provides sufficient number of codewords satisfying constraints d, k. Not all rates are feasible for every block length due to insufficient number of available codewords, and the obtained rates are lower than the conventional 8/9, and 16/17 RLL codes.

The new RLL/PED codes utilize both parities. Hence, the number of available codewords is increased sufficiently to obtain RLL codes of the highest code rate $N/(N+1)$ for any block length $N+1$. Code rates as high as possible are required to increase the linear recording density in band-limited systems and avoid a larger bandwidth expansion factor $B_e \sim {R_C}^{-1}$. Low complexity high-rate constrained codes were presented in [9] with smaller constraint k. The new proposed codes are characterized by lower computational complexity independent of the chosen blocklength $N+1$. They have an increased list of error-detection conditions rather than just constraints pertaining to k only as in [1, 2]. The block mapping is designed so that any concatenation of codewords c satisfy constraint k. The memoryless type

[†] The author is currently with Agere Systems, Milpitas, CA 95035, USA; *Email: abessios@agere.com*.

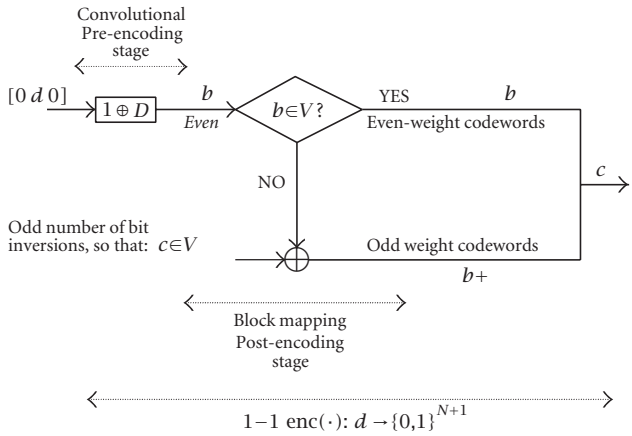

FIGURE 1: The two-stage modulation encoder: (1) Convolutional pre-encoding, (2) Post-encoding: block mapping via conditional bit inversions, where V is the set of all $N+1$-bit sequences any concatenation of which satisfies constraint k.

of encoding/decoding bounds the decoding error propagation within a block's length only. Very low error propagation decoding is supported by the small number of manipulated bits $Q = 1, 2$ per codeword ($Q = 8$ in [9]).

In Section 2, the code construction methodology is presented, with a specific construction example. In Section 3, the system performance is evaluated in relation to its capability for detection of the dominant error events (EE) in high density partial response channels such as EPR4 and E^2 PR4, and analytical results are presented on the decoding error propagation properties. Finally in Section 4, conclusions on the new RLL constrained codes are drawn.

2. CODE CONSTRUCTION

2.1. The algorithm

We consider the set of all polynomials of degree N in GF(2)[x]. Associate the data polynomial $d(x)$ to each dataword $(d_0, d_1, \ldots, d_{N-1})$ so that $d(x) = d_0 + d_1 x + \cdots + d_{N-1}x^{N-1}$. Similarly, denote by $c(x) = c_0 + c_1 x + \cdots + c_N x^N$ and $y(x) = y_0 + y_1 x + \cdots + y_N x^N$, the codeword polynomials of degree N, associated to the recorded $(c_0, c_1, \ldots, c_N)$ and restored $(y_0, y_1, \ldots, y_N)$ codewords, respectively. Encoding of $d(x)$ into codeword $c(x)$, is taking place in two steps (Figure 1).

Component code 1, "pre-encoding":[1,2]

$$b(x) = (1+x)d(x) \quad \text{(overall parity [11])}, \tag{1}$$

where $b(x)$ is the pre-encoded polynomial of degree N and

[1]Binary domain equivalent: $[0\ d\ 0] \xrightarrow{\text{convolve } 1\oplus D} b(N+1\text{-tuples of even weight})$.

[2]Alternatively, the same high rate can be produced if a single parity bit is appended at the end of the N-bit dataword d to produce the $(N+1)$-*bit* b of only odd or only even parity.

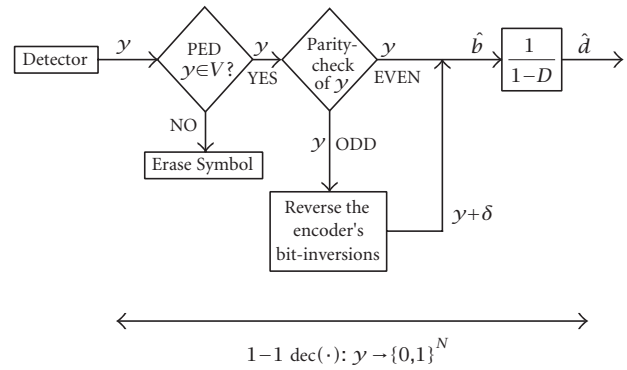

FIGURE 2: The demodulation decoder.

even weight w_b. If constraint k is violated by some concatenations of b, then post-encode:

Component code 2, "post-encoding": add $y(x)$, to invert Q preselected *bits in codeword* b so that all c concatenations form k-constrained sequences

$$c(x) = b(x) + y(x),$$
$$y(x)\begin{cases} = 0 & \text{if all } b \text{ concatenations satisfy } k, \\ \neq 0 & \text{if at least one concatenation of } b \text{ violates } k, \end{cases} \tag{2}$$

where $c(x)$ is the final codeword polynomial of degree N and *even* or *odd* weight w_c, that is, $y(x) = 0$ or $y(x) \neq 0$, respectively.

Set I_Q contains Q predetermined positions (odd number) for bits b_i of the pre-encoded codeword $b(x)$. The set I_Q is a function of q *critical bits* partitioning $b(x)$, and is chosen so that constraint k is satisfied after manipulating Q bits b_i during the post-encoding stage. The post-encoding mapping dimensionality is equal to 2^q, where q out of $N+1$ bits are preselected to determine the *change of an odd number of bits* $Q \geq 1$, so that constraint k is satisfied. The changed parity keeps track of the manipulated codewords c. Finally, the "arbitrary" polynomial $R(x)$ has the remaining terms of $b(x)$ which remain unchanged.

TABLE 1: The block mapping rules.

ξ_i	1	$\bar{b}_i$	0
c_i	$\bar{b}_i$	1	b_i

During the post-encoding stage, two types of conditional mappings are distinguished.

Type I. Odd number of bit inversions $Q = 3, 5, \ldots$ $(Q > 1)$ $I_Q = \{\underbrace{l, m, \ldots, n}_{Q \text{ odd}}\}$

$$\xi_l = \xi_m = \cdots = \xi_n = 1 \longrightarrow y(x) = \underbrace{x^l + x^m + \cdots + x^n}_{Q \text{ terms (odd num.)}},$$
$$\xrightarrow{\text{Tables 1, 2}} c_l = \bar{b}_l,\ c_m = \bar{b}_m,\ c_n = \bar{b}_n. \tag{3}$$

Table 2: The Encoding/Decoding equations.

Encoding	Decoding
$c(x) = \underbrace{(1+x)d(x)}_{b(x)\text{ even weight}} + \underbrace{\sum_{i\in I_Q}\xi_i x^i}_{y(x)}$	$\hat{d}(x) = (1+x)^{-1}\underbrace{\left\{y(x) + (w_y \mod 2)\overbrace{\sum_{i\in I_Q}\zeta_i x^i}^{\delta(x)}\right\}}_{\hat{b}(x)}$
$= \sum_{i=0}^{N} b_i x^i + \sum_{i\in I_Q}\xi_i x^i = \sum_{i\in I_Q}(b_i \oplus \xi_i)x^i + R(x),$	$= (1+x)^{-1}\left\{\sum_{i\in I_Q}\{y_i x^i \oplus (w_y \mod 2)\zeta_i x^i\} + \sum_{i\notin I_Q} y_i x^i\right\}$
where $R(x) = \sum_{i\notin I_Q} b_i x^i$, $I_Q \triangleq f(\underbrace{b_m,\ldots,b_n}_{q})$	$= \begin{cases}(1+x)^{-1}\sum_{i=0}^{N} y_i x^i = (1+x)^{-1}y(x) & \text{if } w_y \text{ even}\\ (1+x)^{-1}\left\{\sum_{i\in I_Q}(y_i \oplus \zeta_i)x^i + \sum_{i\notin I_Q} y_i x^i\right\} & \text{if } w_y \text{ odd}\end{cases}$

Table 3: Key code parameters.[†]

encoding: b_i	$b_{i+(N/2)}$	Encoder $I_Q = f(b_i, b_{i+(N/2)})$	Encoding/Decoding	Codeword
decoding: y_i	$y_{i+(N/2)}$	Decoder $I_Q = f(y_i, y_{i+(N/2)})$	coefficient ξ_i/ζ_i	Parity
0	0	$Q = 3$: $\{i, i+(N/2), i+(3N+4)/4\}$	$1/1$	odd
0	1	$Q = 1$: $\{i-1\}$	$\bar{b}_{i-1}/1$	odd
1	0	$Q = 1$: $\{i+(N/2)+1\}$	$\bar{b}_{i+(N/2)+1}/1$	odd
1	1	$\varnothing$	$0/0$	even

[†]For $i = 0,\ldots,N$, cyclical shifts modulo$(N+1)$ of the k-constraint controlling bits within the codeword's block, generate $N+1$ equivalent codes with respect to rate and constraint k.

Type II. *One* bit is set equal to 1 ($Q = 1$):

$$\xi_i = \bar{b}_i \longrightarrow y(x) = \bar{b}_i x^i \xrightarrow{\text{Tables 1, 2}} c_i = 1. \tag{4}$$

Decoding (see Figure 2). (1) For all odd-weight codewords reverse the inversions incurred during post-encoding:

$$\hat{b}(x) = y(x) + (w_y \bmod 2)\delta(x), \tag{5}$$

where $\hat{b}$ is the estimate of b, w_y is the weight of the received noise-distorted codeword y, and $\delta(x)$ is defined in Table 2.

(2) Divide by $1+x$ to find an estimate of the used word d:

$$\hat{d}(x) = (1+x)^{-1}\hat{b}(x). \tag{6}$$

2.2. Construction example: 2-bit partitioning

$N/(N+1)$ $(0, k = N/2)$ $N = 8, 16, 24, 32, \ldots$

We assume that the *critical bits* $b_i, b_{i+(N/2)}$ $(q = 2)$ partition (index i can be shifted cyclically modulo$(N+1)$) the pre-encoded $b = [{}^{**}b_i{}^{*}\underbrace{\cdots}_{N/2-1}{}^{*}b_{i+N/2}{}^{*}\underbrace{\cdots}_{N/2-2}{}^{*}]$, where * represents a "don't care condition (0 or 1). The samples controlling constraint k are c_{i-1}, c_i, $c_{i+[N/2]}$, $c_{i+[N/2]+1}$, $c_{i+[(3N+4)/4]}$. The 2-bit state $(b_i\ b_{i+(N/2)})$ determines the post-encoding rules and the parity of the generated codeword c. Therefore only four mappings take place independent of the block length N, whereas for the gated logic techniques [1, 2] the mapping lookup-table grows exponentially with N. It is the state $(b_i\ b_{i+(N/2)})$ that an odd number of bit inversions is dependent upon, so that constraint k is satisfied for all possible codeword concatenations. When there is no need for post block mapping without violating the target constraint k (allowed state), then the final codeword c preserves an even weight. The state $(y_i\ y_{i+(N/2)})$ combined with y's parity calculated by the decoder, determine the decoding rules (see Figure 3). There are many alternative choices to the above transformations with the same resulting k, that is, instead of inverting sample $b_{i+(3N+4)/4}$, any other bit can be used within the $\lfloor b_{i+(N/2)+2}\ b_{i+N-1}\rfloor$ range, or instead of setting bit $c_{i+(N/2)+1} = 1$, $c_{i+(N/2)-1}$ could be set to 1, and so forth.

The existing coupling between constraint k and the block length N, has implications on the phase detector updates in the PLL, that is, very large block lengths N yield relatively loose constraint k unable to guarantee sufficiently frequent transitions at the phase detector. However, typical block lengths of $N = 8, 16, 32, \ldots$ yield practical k values. The target value for the constraint k depends on the selected number of critical and manipulated bits q and Q, respectively, as well as their relative positions within the codeword. To produce a tighter constraint k, such as $k = N/3$, a larger number of critical bits $(q > 2)$ and/or manipulated bits $(Q > 3)$ should be used.

For the rate $16/17$ and $q = 2$-bit or 5-bit partitioning, less than 0.07% and 0.01%, respectively of the codewords c have Hamming weight w_c less than 4, and for more than 91% of the codewords c, their Hamming weights w_c range between 7 and 13 (see Figure 4). This type of Hamming weight distribution for the encoded symbols secures that the necessary frequency of transitions is produced to update regularly the phase detector in the timing recovery loop.

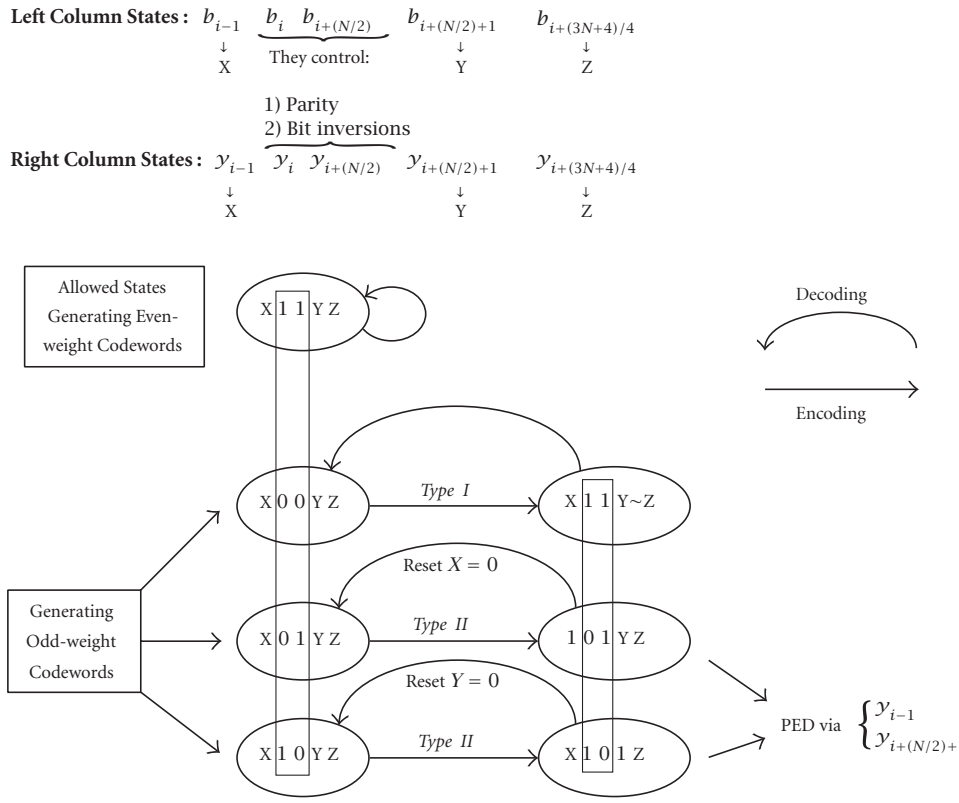

Figure 3: The state-transition diagram describing the conditional bit inversions during encoding/decoding; X, Y, Z are "don't care values".

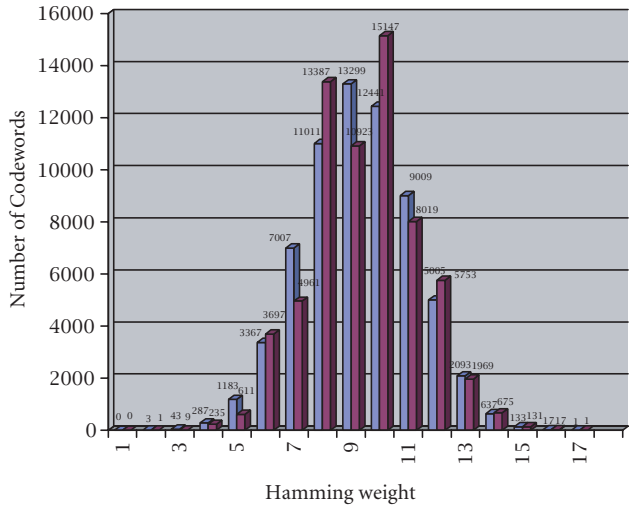

Figure 4: Hamming-weight distribution for the 2-bit (left) and 5-bit (right) partitioning codes.

2.3. Error control (PED)

Observing the signal space-partitioning by the encoding rules, derives the error-detection equations. By complementing the post-encoding rules described by (3), (4), and Table 2, the produced illegal combinations of y_i's form the error detecting conditions. *Type II* mapping performs error detection under the following two three-bit conditions, while *Type I* mapping eliminates the state $b_i = 0$, $b_{i+N/2} = 0$ to form one two-bit error detecting condition. These error detection conditions capture both random and/or burst errors.

Table 4: Error Detection (* denotes a "don't care" condition).

$q = 2$ critical bits		mapping onto the	The derived error detection eqns.			
b_i	$b_{i+N/2}$	final codeword c	y_{i-1}	y_i	$y_{i+N/2}$	$y_{i+N/2+1}$
0	1	Type II: $c_{i-1} = 1$	0	0	1	*
1	0	Type II: $c_{i+N/2+1} = 1$	*	1	0	0
0	0	Type I: $c_i = c_{i+N/2} = 1$	*	0	0	*

2.4. Example code of rate 8/9

Assume the user dataword $d = (01001000)$:

(1) Encoding: $b(x) = (1 + x)d(x) \rightarrow b = (011011000)$. From Table 3, and for $i = 2$: $\xi_2 = \xi_6 = \xi_9 = 1$, $\xi_1 = \bar{b}_1 = 0$, $\xi_7 = \bar{b}_7 = 1$, $b_2 = 1$, $b_6 = 0 \Rightarrow y(x) = x^7 \Rightarrow c(x) = b(x) \oplus y(x) \rightarrow c = (011011010)$ recorded codeworded.

(2) Decoding: $y = c = (011011010)$ has odd parity and $y_2 = 1, y_6 = 0 \Rightarrow \delta(x) = x^7 \Rightarrow \hat{b} = y \oplus \delta = (011011000) \Rightarrow \hat{d}(x) = \hat{b}(x)/(1+x) \rightarrow \hat{d} = (01001000)$ restored codeword.

(3) Error Control: we assume that in the presence of excessive noise levels, the value of bit y_7 is changed from 1 to 0, so that the erroneous codeword $y = (011011000)$

Table 5: The single bit *NRZ* error $e_i^{\pm 1}$; EPR4 16/17(0, 8); $d_{\min}^2 = 4$ EEs.[‡]

	Composite error detecting patterns; $e_i = c_i - y_i$; $1 \le i < 17$
Type I error detection rules:	
$y_i = 0$ & $y_{i+8} = 0$	$\left.\begin{array}{rcr} e_i^1 & \& & y_{i+8} = 0 \\ y_i = 0 & \& & e_{i+8}^1 \end{array}\right\} \rightarrow P_{\text{cmp}} = P_{\text{EE}}/2$
Type II error detection rules:	
$y_{i-1} = y_i = 0$ & $y_{i+8} = 1$	$\left.\begin{array}{rcrcr} e_{i-1}^1 & \& & y_i = 0 & \& & y_{i+8} = 1 \\ y_{i-1} = 0 & \& & y_i = 0 & \& & e_{i+8}^{-1} \\ y_{i-1} = 0 & \& & e_i^1 & \& & y_{i+8} = 1 \end{array}\right\} \rightarrow P_{\text{cmp}} = P_{\text{EE}}/4$
$y_i = 1$ & $y_{i+8} = y_{i+9} = 0$	$\left.\begin{array}{rcrcr} e_i^{-1} & \& & y_{i+8} = 0 & \& & y_{i+9} = 0 \\ y_i = 1 & \& & e_{i+8}^1 & \& & y_{i+9} = 0 \\ y_i = 1 & \& & y_{i+8} = 0 & \& & e_{i+9}^1 \end{array}\right\} \rightarrow P_{\text{cmp}} = P_{\text{EE}}/4$

[‡]All operations on indices are mod $(N+1)$, that is, mod 17 in this case.

is received. From Tables 3 and 4: $\xi_7 = \bar{b}_7 \Rightarrow c_7 = 1$, since $y_7 = 0 \rightarrow y_2 = 1$, $y_6 = 0$, $y_7 = 0 \Rightarrow$ An error is detected. If instead y_2 bit value is changed from 1 to 0, then the received codeword $y = (010011010)$, and from Tables 3 and 4: $\xi_2 = \xi_6 = 1$ and $y_2 = 0 \Rightarrow$ An error is detected.

3. PERFORMANCE EVALUATION

3.1. PED and the sequence detector's dominant error events

All symbol errors non-supported by the code are detectable. This is not the case for channel bit errors fabricating valid codewords c. The number of invalid codewords equals the number of valid codewords therefore for independent bit errors an equal probability (1/2) for detecting or missing an error is being produced. However, in systems using sequence detectors or convolutional decoders (VA) there are dominant error events. Among other random error patterns, some of the sequence detector's dominant minimum distance EEs can be captured. Without loss of generality, we define the *NRZ* error sequence $e = c - y$. Then the *NRZ* errors $e_i = c_i - y_i$ and the corresponding detected bit values y_i are related by: if $e_i = 0 \Rightarrow y_i = 0$, or 1; if $e_i = 1 \Rightarrow y_i = 0$; if $e_i = -1 \Rightarrow y_i = 1$. We assume in an EPR4 system the $d_{\min}^2 = 4$ EE with error sample vector $[1\ 0\ -1\ 1\ 0\ -1]$, produced by the *NRZ* input error vector e_i^{1-11} with first bit error at position i:

$$e_i^{1-11} \triangleq [e_i = 1, e_{i+1} = -1, e_{i+2} = 1], \tag{7}$$

where the superscript denotes the *NRZ* error sequence and the subscript denotes the first bit error position. The detectable error patterns combine $d_{\min}$ EEs and conditions on certain restored bits y_j. Their probability of occurrence $P_{\text{cmp}} = 2^{-m} P_{\text{EE}} < P_{\text{EE}}$ (probability of $d_{\min}$ EE), where $m = 1, 2, \ldots$ is the number of additional bits y_j in the condition. Tables 5 and 6, list the conditions under which the single-bit and the tribit *NRZ* errors are captured to enhance the *hit/miss* error ratio.

The new channel coding schemes become more powerful when cascaded with an outer ECC code [12]. The decoder observes the restored high-rate encoded binary sequences y and based on the encoder's list of constraints, it either decodes a legitimate restored codeword y to the initial data word $\hat{d}$, or it generates erasure byte(s) $y^{(L)}$ after a constraint violation is detected [12]. The location L of the erased symbol $y^{(L)}$ is passed from the demodulation decoder to the outer ECC decoder. Its error correcting capability is doubled by operating on the erased symbols (error symbols of known location) produced by the PED scheme of the new RLL code.

3.2. Low error propagation decoding

The RLL/PED codes are better immuned to demodulator-induced error bursts than conventional RLL [1, 2]. The decoder's feedback operation due to the polynomial division, renders the system prone to error propagation, however bounded by the block's boundary. That feedback effect can be eliminated if during pre-encoding, instead of multiplying with $1 + x$, a single parity bit is appended to form either only even, or only odd pre-encoded codewords b. To further reduce the decoding error propagation, the added parity bit is dropped instead of performing division by $1 + x$.

A channel bit error does not necessarily corrupt the whole codeword, while this is not the case with conventional RLL codes (see Table 7). The decoder's error burst length B (in bytes) depends on 2 properties of the EE: (1) its length L, and (2) its starting point L_1. If the critical bits are y_i, $y_{i+[N/2]}$ and the channel EE starts at a position $L_1 > i + [N/2]$, then the front part of the codeword is still decoded correctly: first byte is saved for rates 16/17, 24/25, first and second bytes for rate 32/33, and so forth. In general, the first $[N/16]$ bytes are uncorrupted.

In general, the tighter (looser) the target constraint k, the larger (smaller) the number of critical and manipulated bits needed q and Q, respectively, and therefore the higher (lower) the decoding error propagation, while the smaller (larger) the probability that a larger fraction of the codeword will be uncorrupted.

Table 6: The tribit-*NRZ* error $e_i^{\pm 1-11}$; EPR4 16/17(0, 8); $d^2_{\min} = 4$ EEs.[#]

	Composite error detecting patterns; $e_i = c_i - y_i$; $1 \le i < 17$
Type I error detection rules	
$y_i = y_{i+8} = 0$	$\left.\begin{array}{ll} e_i^{1-11} & \& \; y_{i+8} = 0 \\ e_{i-2}^{1-11} & \& \; y_{i+8} = 0 \\ e_{i-1}^{-11-1} & \& \; y_{i+8} = 0 \\ y_i = 0 & \& \; e_{i+6}^{1-11} \\ y_i = 0 & \& \; e_{i+8}^{1-11} \\ y_i = 0 & \& \; e_{i+7}^{-11-1} \end{array}\right\} \rightarrow P_{cmp} = P_{\text{EE}}/2$
Type II error detection rules	
$y_{i-1} = y_i = 0 \;\&\; y_{i+8} = 1$	$\left.\begin{array}{lll} e_{i-3}^{1-11} & \& \; y_i = 0 & \& \; y_{i+8} = 1 \\ e_{i-1}^{-11-1} & \& \; y_i = 0 & \& \; y_{i+8} = 1 \\ y_{i-1} = 0 & \& \; e_{i-1}^{-11-1} & \& \; y_{i+8} = 1 \\ y_{i-1} = 0 & \& \; e_i^{1-11} & \& \; y_{i+8} = 1 \\ y_{i-1} = 0 & \& \; y_i = 0 & \& \; e_{i+6}^{1-11} \\ y_{i-1} = 0 & \& \; y_i = 0 & \& \; e_{i+8}^{1-11} \\ y_{i-1} = 0 & \& \; y_i = 0 & \& \; e_{i+7}^{-11-1} \end{array}\right\} \rightarrow P_{\text{cmp}} = P_{\text{EE}}/4$
$y_i = 1 \;\&\; y_{i+8} = y_{i+9} = 0$	$\left.\begin{array}{lll} e_{i-1}^{1-11} & \& \; y_{i+8} = 0 & \& \; y_{i+9} = 0 \\ e_i^{-11-1} & \& \; y_{i+8} = 0 & \& \; y_{i+9} = 0 \\ e_{i-2}^{-11-1} & \& \; y_{i+8} = 0 & \& \; y_{i+9} = 0 \\ y_i = 1 & \& \; e_{i+6}^{1-11} & \& \; y_{i+9} = 0 \\ y_i = 1 & \& \; y_{i+8} = 0 & \& \; e_{i+9}^{1-11} \end{array}\right\} \rightarrow P_{\text{cmp}} = P_{\text{EE}}/4$

[#]Note that the tribit generates the dominant $d_{\min}$ EE for E^2 PR4 systems too.

Table 7: min length EE l, and max length error burst B (*in bytes*).

16/17 l	B	24/25 l	B
New/Conventional		New/Conventional	
1/1	2	1/1	3
6/2	3	8/2	4
10/2	4	14/2	6

Table 8: RLL rate 16/17: Decoder Error Propagation.

[EE Type] (d)	$B_{\min}^{\text{RLL/PED}} - B_{\max}^{\text{RLL/PED}} / B_{\min}^{\text{conv}} - B_{\max}^{\text{conv}}$
[1] (2)	1(94%) – 2(6%) / 2(100%)
[11] (2.4)	1(74%) – 2(26%) / 2(97%) – 4(3%)
[111],[101] (2), [1–11] (3.4)	1(59%) – 2(41%) / 2(91%) – 4(9%)

3.3. Analytical performance evaluation for the RS-RLL/PED concatenation

An analytical upper bound for the BER performance of the RLL/PED and its coding gain over the conventional RLL codes when they are concatenated with an outer ECC code, is developed in this section. This coding gain is attributed to its lower decoding error propagation and its PED capability. EE types listed in Table 8, dominate the assumed PR channel with transfer function $(1 - D^2)(1 + D)$. Therefore, the channel output BER is upper bounded by:

$$\begin{aligned} &\text{channel outp. BER: } p_b \le \sum_d K_d Q(Rd\,\text{snr}), \\ &\qquad\qquad \text{where } R = R_{\text{RLL}} R_{\text{ECC}} = \frac{N-2t}{N+1}, \qquad (8) \\ &\text{RLL decode outp. BER: } \quad p_b^{\text{RLL}} \le \sum_d B_d^{\text{avg}} K_d Q(Rd\,\text{snr}), \end{aligned}$$

where d is the EE distance, K_d is the multiplicity factor, and B_d^{avg} is the average number of byte errors per EE of distance d at the RLL decoded output derived from Table 8.

An ECC decoding failure occurs each time the number of RLL decoded bytes in error per block exceeds its error correcting capability t. Value equal to 1 is assigned to variable B_i for a byte in error and 0 otherwise. Then the ECC failure rate F can be evaluated as follows:

$$\begin{aligned} F &= P\left(\sum_{i=1}^{N} B_i > t\right) = \sum_{i=1}^{N-t} \binom{N}{t+i} \left(P_{\text{byte}}^{\text{err}}\right)^{t+i} \left(P_{\text{byte}}^{\text{corr}}\right)^{N-t-i}, \\ P_{\text{byte}}^{\text{err}} &= P(B_i = 1) = \sum_{i=1}^{k} \binom{k}{i} \left(p_b^{\text{RLL}}\right)^i \left(1 - p_b^{\text{RLL}}\right)^{k-i}, \\ P_{\text{byte}}^{\text{corr}} &= P(B_i = 0) = 1 - P_{\text{byte}}^{\text{err}} = \left(1 - p_b^{\text{RLL}}\right)^k. \end{aligned} \qquad (9)$$

Depending on the application, the two ways of exploiting the RS-RLL/PED gain are summarized in Table 9.

Analytical performance comparisons are shown in Figure 5, between RLL/PED and conventional RLL [1, 2]. At a BER = $1e - 10$ (ECC decoded output), the RLL/PED and the conventional RLL suffer, respectively, 1.5 dB and 0.6 dB loss due to error propagation. Turning on the PED capability

Table 9: ECC-RLL/PED concatenated system performance gain.

Design parameters kept constant	RLL/PED advantage over RLL conventional
ECC Error Correcting capability t (same R, BW expansion, channel BER)	Lower RLL decoded BER (PED + lower B combined), Larger system margin (lower end BER)
End BER	Smaller: overhead (t), BW expansion (lower channel BER), B lower speed electronics (slower channel clock/sampling rate)

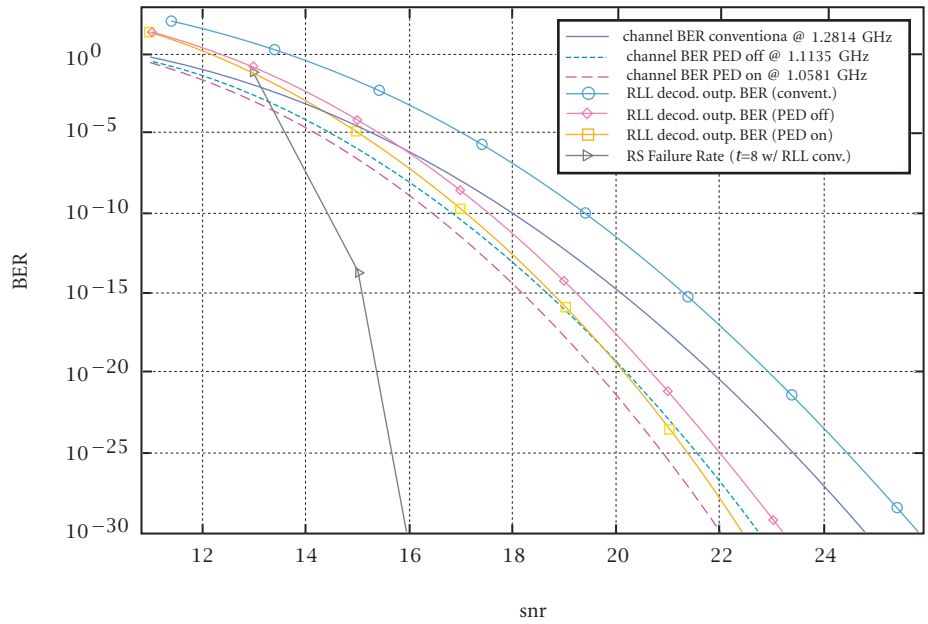

Figure 5: Performance comparison: RLL convent. versus RLL/PED; same *end BER*; different t, R, B.

an additional 0.3 dB is gained, to yield a 2 dB compound gain which can be translated as an 18% clock speed savings (by reducing the outer ECC required overhead).

4. CONCLUSIONS

A construction methodology for a new class of highly efficient, low complexity, RLL/PED constrained codes has been presented. The encoding, decoding, error control equations and constraint k are expressed as simple functions of the information block length N, so that a *fixed* rate-independent endec *architecture* for encoder/decoder/error-control is accomplished, to offer: (1) very high efficiency obtained at a constant very low complexity, (2) rate adjustable codes. These properties render the codes suitable for simple implementation requiring inexpensive endec circuitry. The decoder's *PED capability* indistinguishably captures some of the single drop-in, dropout, bit-shifts as well as error bursts. The creation of composite error control patterns, capable of capturing the sequence detector's $d_{\min}$ EEs, increases the probability of error detection. The new codes obtain superior performance for the concatenation of inner modulation/outer ECC decoders: If the channel errors corrupt any of the q critical bits, its PED capability generates erasure symbols and boosts the ECC performance; if not, then due to the decoder's lower error propagation, most probably one or more bytes per block are still decodable therefore relaxing the ECC redundancy requirement.

ACKNOWLEDGEMENT

This research was done while the author was affiliated with Texas Instruments, Inc., Dallas, TX, USA.

REFERENCES

[1] J. Eggenberger and A. M. Patel, "Method and apparatus for implementing optimal PRML codes," US Patent, no. 4707681, November 1987.

[2] A. M. Patel, "Rate 16/17(0,6/6) code," *IBM Tech. Discl. Bull.*, vol. 38, 1989.

[3] P. Lee and J. K. Wolf, "A general error-correcting code construction for RLL binary channels," *IEEE Trans. Inform. Theory*, vol. 35, no. 6, pp. 1330–1335, 1989.

[4] P. H. Liu and Y. Lin, "A class of (d,k) block codes with single error-correcting capability," *IEEE Trans. on Magnetics*, vol. MAG-33, no. 5, pp. 2758–2760, 1997.

[5] A. Patapoutian and P. V. Kumar, "The (d,k)-subcode of a linear block code," *IEEE Trans. Inform. Theory*, vol. 38, no. 4, pp. 1375–1381, 1992.

[6] H. C. Ferreira and S. Lin, "Error and erasure control (d, k) block codes," *IEEE Trans. Inform. Theory*, vol. 37, no. 5, pp. 1399–1408, 1991.
[7] P. N. Perry, "RLL codes for single error detection in the magnetic recording channel," *IEEE Trans. Inform. Theory*, vol. 41, no. 3, pp. 809–815, 1995.
[8] K. A. S. Immink, "Error detecting runlength-limited sequences," IEE Video, Audio & Data Recording.
[9] K. A. S. Immink and A. van Wijngaarden, "Simple high-rate constrained code," *Electronics Letters*, vol. 32, no. 20, pp. 1877–1877, 1996.
[10] A. G. Bessios, "Construction of low-complexity high-performance $n/n + 1$ deterministic modulation codes with adjustable codeword length and error control capability," in *Proceedings of SPIE*, San Diego, July 1998.
[11] W. W. Peterson and E. J. Weldon, *Error-Correcting Codes*, MIT Press, Cambridge, MA, 2nd edition, 1996.
[12] M. Blaum, "Combining ECC with modulation: performance comparison," *IEEE Trans. Inform. Theory*, vol. 37, no. 3, pp. 945–949, 1991.

Anthony Bessios received the B.S.E.E. from the National Technical University of Athens, Greece, in 1988 and the M.S. and Ph.D. degrees in electrical engineering from Northeastern University, Boston, in 1990 and 1993, respectively. From 1991–1993 he was a visiting research assistant at the University of Southern California with the Signal and Image Processing Institute. In 1994 he was a visiting scientist at the National Technical University of Athens with the Institute of Communications and Computer Systems. In 1996 he joined Texas Instruments, Dallas, TX where he was with the Storage products Department in the Semiconductor Group. In 1998 he joined the Microelectronics R&D group at Bell Labs-Lucent Technologies, in Allentown, PA, as a member of the technical staff. He also held an Adjunct Professor appointment with the EE Department at Lehigh University, in Bethlehem PA. Since July 2000, he is with the ICD/NetCom group at Agere Systems, at Milpitas, CA. His current work involves performance analysis and modeling of equalization, detection, timing recovery and coding techniques for PRML read channels in magnetic recording, and high speed data transmission at the physical layer.