Description

This invention relates to a method for processing signals in a channel, and more particularly to processing, in a signal processing channel, sample values corresponding to a digitized and equalized analog signal representative of coded binary data transmitted via the channel.

Background of the Invention

The trend towards high density and high data rates presents heavy demands on the signal detection process in a magnetic recording channel. A peak-detection channel, with analog signal processing of one bit at a time, is now widely used in magnetic recording storage products. Due to increases in density and data rates, the one-bit detection window in the conventional peak detection channel becomes very small, which limits detection reliability. An alternative channel, called partial response maximum likelihood (PRML) channel, is discussed in the following papers:

(1) Kabal et al, "Partial-Response Signaling", IEEE Trans. on Comm., vol. COM-23, NO. 9, Sep. 1975.

(2) Forney, "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference", IEEE Trans. on Info. Theory, vol. IT-18, No. 3, May 1972.

(3) Kobayashi, "Application of Probabilistic Decoding to Digital Magnetic Recording Systems", IBM J. Res. Develop., Jan. 1971.

(4) Nishimura et al, "A Design Method for Optimum Equalization in Magnetic Recording with Partial Response Channel Coding", IEEE Trans. on Mag., vol. MAG-19, No. 5, Sep. 1983.

(5) Forney, "The Viterbit Algorithm", Proc. of the IEEE, vol. 61, No. 3, 3/73.

In the PRML channel, characterized by the polynomial (1-D) (1+D), a notch filter is needed because the frequency response requires a sharp cutoff and the frequency spectrum is very different from that of the channel response in magnetic recording.

IBM Technical Disclosure Bulletin, Vol. 30, No. 9, February 1988, page 65 et seq, discloses a system for decoding (1,k) code data using a sequence of digital voltage samples Vn to Vn+3.

In "A Class of Partial Response Systems for Increasing Storage Density in Magnetic Recording", H.K. Thapar and A.M. patel, IEEE Trans. Magnetics, Vol. MAG-23, No. 5, September 1987 at pp. 3666-3668, an extended partial response maximum likelihood (EPRML) channel is described characterized by the polynomial $(1-D)(1+D)^2$. In this signalling method, the signal spectrum resembles a typical magnetic-recording channel transfer function. It desirably obviates the need for the notch filter. However, in this channel, the Viterbi type computations for maximumlikelihood detection become a limiting factor in terms of decoding speed and cost. Furthermore, both PRML and EPRML channels are very sensitive to misequalization or changes in signal shape due, for example, to component tolerances and to nonlinearities of the magnetic recording process such as caused by pulse asymmetry and the crowding of write transitions on the media.

There is a need for a signal processing method and apparatus wherein (1) a run-length limited (RLL) code of the peak-detection channel is used to control the crowding of write transitions on the media, (2) a sequence detection algorithm is used that avoids the complexity of Viterbi-type computations, and (3) the detection process is tolerant to signal shape variations in order to remove stringent requirements for signal equalization and accommodate nonlinearities of the magnetic recording process.

Summary of the Invention

The present invention is defined in the attached claims.

Toward this end and according to the invention, a method is provided for processing, in a signal processing channel, sample values of an analog signal corresponding to binary data coded with a RLL code, such as the (1,7) code. The coded analog input signal is converted by an analog-to-digital converter (ADC) to a sequence of digital sample values at successive clock times. The signal is equalized, either after the ADC by a digital equalizer or before the ADC with an analog filter or (as illustrated) by a combination of the two, to correspond to a predetermined analog shape. Then a sequence detection algorithm is used to decode the digital sample values into the coded binary data. The decoder is preferably programmable to adjust the thresholds in positive and negative phases of the decoder when anomalies in signal shape cannot be compensated for by the equalizer. If desired, the decoder can be continuously adaptive to adjust to variations in signal shape.

The method and apparatus herein disclosed for processing the sample values desirably:

(1) Uses a channel with a signaling method that eliminates the need for a notch filter and processes (1,7) coded analog signals.

(2) Converts the analog input signal to a sequence of digital sample values that corresponds to a predetermined equalized analog shape.

(3) Avoids Viterbi-type computations yet obtains near-optimum performance by a state dependent look-ahead technique focusing on the minimum-distance and next most likely error events.

(4) Utilizes a programmable decoder that is capable of adjusting to variations in signal shape; and

(5) Flags, during the look-ahead process, a previously made decision that is suspected of being in error.

Brief Description of the Drawings

Fig. 1 is a schematic diagram of a coded digital signal processing channel embodying the invention;

Fig. 2A depicts written transitions of magnitude .2 between +1and -1 levels of maqnetic saturation, and Fig. 2B depicts a read response normalized by a scaling factor to correspond to said written transitions;

Fig. 3 shows write and read waveforms for (1,7) coded data and read sample values;

Fig. 4 is a schematic diagram of a decoder embodying a look-ahead feature and decision boundaries for implementing the invention;

Fig. 5 is a diagram illustrating how the decoder shown in Fig. 4 may be modified to a programmable decoder that adjusts to variations in analog input signal shape.

Desciption of Preferred Embodiment

As illustrated in Fig. 1, the apparatus embodying the invention comprises a signal processing channel including a line 10 via which an analog read signal is fed from a transducer, such as a magnetic or optical read head in a digital storage device. This read signal corresponds to a recorded (1,7) data sequence, which is a binary data sequence coded with the (1,7) RLL code during the write operation. The read signal passes through a preamplifier 11 with automatic gain control (AGC) and a low bandpass filter 12. The output of filter 12 is supplied to a phase-locked clock 13 comprising a phase-locked loop timing recovery circuit including a variable frequency oscillator (VFO). The apparatus, as thus far described, may be identical with that used in the conventional peak-detection channel with (1,7) code.

The channel is preferably of the type referred to as extended partial response maximum likelihood (EPRML) channel which is characterized by the polynomial $(1-D)(1+D)^2$ In EPRML signaling, the signal spectrum resembles a typical magnetic recording channel transfer function, desirably obviating the need for the notch filter required in the normal PRML channel characterized by the polynomial (1-D) (1+D).

More specifically, the filter 12 and equalizer 22 operate to filter the read signal so that the response to a single magnetic transition is a pulse given by the sample values ..., 0, 2, 4, 2, 0 Fig. 2A depicts written transitions of magnitude 2 between +1 and -1 levels of magnetic saturation; and Fig. 2B depicts the read response that is normalized by a scaling factor to correspond thereto.

In the environment herein described, use of the (1,7) RLL code reduces crowding of write transitions on the media to 2/3 that experienced with the PRML or EPRML channel. The (1,7) code acts as a trellis code providing a built-in redundancy with its 2/3 rate. It is very tolerant to nonlinearities in a high-density, high-data-rate magnetic recording channel.

Briefly, according to the invention, the coded analog read signal output from filter 12 is supplied to a delay means 20.Delay means 20 may be a digital chip of the type designated as the PECLDL-28-01 Programmable Logic Delay Line, marketed by Engineered Components Company of San Luis Obispo, California, and which permits selectively adjustable delays in increments as low as a tenth of a nanosecond, depending upon the rate of the clock. Delay means 20 provides delay alignment between the analog signal and clock signal inputs to an analog-to-digital converter (ADC) 21. ADC 21 converts the analog input siqnal into digital sample values at successive clock times. These digital sample values, in some cases (depending on the characteristic of filter 12) may be equalized by an equalizer 22. The digital sample values

are then passed to a decoder 23. Positive and negative threshold constants are-applied via lines 24,25, respectively, to decoder 23. Decoder 23 applies a decoding algorithm to the equalized digitized sample values to provide a coded binary data output in line 26 and, where appropriate, a pointer flag in line 27, at successive clock times under control of clock 13.

Fig. 3 depicts the write and read waveforms and associated sequences of data and sample values in a noise-free environment as they are clocked by clock 13. With the pulse shape as shown in Figs. 2A,2B, the sample values, in accord with the decoder implementation, will range through the discrete set {-4, -2, 0, +2, +4}. Actual samples, however, will include signal anomalies and additive noise, and thus will range through various noninteger values around the integers in the discrete set.

The read clock is derived from read signal 10 using a conventional peak-detection circuit driving a variable frequency oscillator (VFO) in the clock 13. As earlier noted, the ADC 21 under control of clock 13 converts the analog signal to digitized sample values at successive clock times; and the digitized sample values are equalized at 22.

Assume that y_i denotes the digitized sample value corresponding to the i^{th} clock cycle. Then, as illustrated in Fig. 4, the current value of y_i is illustrated as y_0, corresponding to the current clock cycle i=0. Decoder 23 processes these digitized equalized sample values iteratively, receiving one new sample per clock time with a look-ahead of six sample values. Thus, decoder 23 receives y_{i+6} as it processes the previously received samples and decodes y_i into the i^{th} digit of the recorded sequences. It should be noted that the decoding process is state dependent. The state of decoder 23 at the i^{th} clock cycle is represented by a three digit binary number $a_ib_ic_i$ where a_i, b_i and c_i represent the binary logic levels of the write current (as illustrated in Fig. 3) for the last three bit values at the ith clock cycle. This identifies one of the six possible states, namely, 000, 100, 110, 111, 011, and 001. (The states 101 and 010 do not occur with (1,7) coded data employed in an EPRML channel). During each cycle of clock 13, decoder 23 determines one digit of the recorded (1,7) sequence and also identifies the next decoder state for processing of the next cycle.

The Decoding Algorithm

Decoder 23 executes an algorithm using the equations in Tables 1 and 2. In this algorithm, the sample value that is currently being decoded is denoted by y_0, corresponding to the clock cycle i=0. The process is iterative and is driven by read clock 13.

According to a feature of the invention, decoder 23 is "state dependent"; i.e., given the current state $a_0b_0c_0$ and sample values y_0, y_1, y_2, y_3 and y_4, the decoder determines the noise-free sample value $\tilde{y}_0$ (corresponding to y_0) and the resultant next state $a_1b_1c_1$. It also decodes this state into a 1 or 0 (presence or absence of a magnetic transition) in the originally recorded (1,7) data sequence. The actual decision results from the outcome of the tests on the sample values as given by the equations in Tables 1 and 2 as decoder 23 moves from one state to another. Note that the movement from one state to another is highly structured and that the form of the equations is very simple and similar at various states. The states 110, 100 and 000 (Table 1) corresponding to the positive phase of the signal have mirror image symmetry with states 001, 011, and 111 respectively (Table 2) corresponding to the negative phase of the signal. The mirror image symmetry is characterized by a change in the sign for all the sample values in the corresponding places in Tables 1 and 2.

The decisions of decoder 23 are driven by the outcome of three basic checks: namely (i) the baseline check; (ii) the peak-position check; and (iii) the phase check. The outcomes of these checks are represented in the form of Boolean vari-ables (see Fig.4) denoted by A_1, A_2,B_1 and B_2 for the baseline check; and by Q_1 and Q_2 for the peak-position check, where Q_i and Q_2 are functions of more specific checks represented by X_1, X_2, Y_1' Y_2, Z_1 and Z_2; and by the Boolean variable p for the phase check. Each check is a comparison of a specific function of the sample values against a fixed threshold. Each threshold is represented by a corresponding constant that is strongly related to the shape of the readback signal and defined by the right-hand side of the corresponding test equation. In the following test equations, the threshold constants correspond to the nominal pulse shape with sample values .., 0, 2, 4, 2, 0, .. and .., 0, -2, -4, -2, 0, .. as illustrated in Fig. 2B.

(i) Baseline Check (A_1, A_2, B_1 and B_2)

$$A_1=1 \text{ implies } (y_0+y_1)+(y_1+y_2) \leqq 4 \quad (1)$$

$$A_2=1 \text{ implies } (y_0+y_1)+(y_1+y_2) \leqq 6 \quad (2)$$

$$B_1=1 \text{ implies } (y_0+y_1)+(y_1+y_2)-y_3 \leqq 4 \quad (3)$$

$$B_2=1 \text{ implies } (y_0+y_1)+(y_1+y_2)-y_3 \leqq 6 \quad (4)$$

(ii) Peak-Position Check Q_1 and Q_2

The expression

$$\lceil_a^{\ } (y_3 + y_4) \rceil_b^{\ }$$

is defined as follows:

$$\lceil_a^{\ } (y_3 + y_4) \rceil_b^{\ } = \begin{cases} a & \text{if } (y_3 + y_4) \leqq a \\ b & \text{if } (y_3 + y_4) \geqq b \\ (y_2 + y_4) & \text{if } a < (y_3 + y_4) < b \end{cases} \quad (5)$$

Then, the peak-position check represented by Boolean variables Q_1 and Q_2 is expressed as follows:

$$Q_1 = 1 \quad \text{implies} \quad (y_0 + y_1) - (y_2 + y_3) + \lceil_{-4}^{\ } (y_3 + y_4) \rceil_{-2}^{\ } \leqq -4 \quad (6)$$

$$Q_2 = 1 \quad \text{implies} \quad (y_0 + y_1) - (y_2 + y_3) + \lceil_{-4}^{\ } (y_4 + y_3) \rceil_{-2}^{\ } \leqq -2 \quad (7)$$

where

$$Q_1=X_1+Y_1Z_1 \quad (8)$$

$$Q_2=X_2+Y_2Z_2 \quad (9)$$

and

$$X_1=1 \text{ implies } (y_0+y_1)-(y_2+y_3) \leqq -2 \quad (10)$$

$$X_2=1 \text{ implies } (y_0+y_1)-(y_2+y_3) \leqq 0 \quad (11)$$

$$Y_1=1 \text{ implies } (y_0+y_1)-(y_2+y_3) \leqq 0 \quad (12)$$

$$Y_2=1 \text{ implies } (y_0+y_1)-(y_2+y_3) \leqq 2 \quad (13)$$

$$Z_1=1 \text{ implies } (y_0+y_1)-(y_2+y_3)+(y_3+y_4) \leqq -4 \quad (14)$$

$$Z_2=1 \text{ implies } (y_0+y_1)-(y_2+y_3)+(y_3+y_4) \leqq -2 \quad (15)$$

(iii) Phase Check (p)

$$p=1 \text{ implies } (y_1+y_2)+(y_2+y_3) \leqq -7 \quad (16)$$

The above tests correspond to the equations in Table 1 where $c_0=0$ which corresponds to the states in the positive phase of the signal. The same tests are used with a sign change to all the sample values when $c_0=1$ for the equations in Table 2 which corresponds to the states in the negative phase.

The hardware of decoder 23 is shown in Fig. 4. Registers 30-40 hold the respective interim functions of the sample values y_0 through y_4 as shown. During each cycle, the content of each register 30-40 is moved in the direction of the

arrow to the next register as appropriate functions are created through five adder circuits 41-44, 44a. Sign change blocks 45-49 change the sign of the incoming binary number whenever the c_0 bit in a state register 50 has the value "1". The outputs from the respective sign change blocks 45-49 are fed to comparators 51-61 as shown in Fig. 4. The outputs of comparators 51-61 correspond to those shown in Equations 1-4 for the variables A_1, A_2, B_1, and B_2; Equations 5-15 for the variables X_1, X_2, Y_1, Y_2, Z_1 and Z_2, respectively; and Equation 16 for the variable p. The outputs from the comparators 51-61 are values "1" or "0" which together with the state values a_0, b_0, c_0 from state register 50, are fed to the logic block 62 for implementing the logic shown in Table 3.

The logic of block 62 provides the decoded data d_0 corresponding to state $a_0b_0c_0$ and the erasure pointer in accordance with the logic of section 1 of Table 3. Block 62 also creates the state value $a_1b_1c_1$ for the next state in accordance with the logic of section 2 of Table 3. This state value is supplied to state register 50 to become the current state value $a_0b_0c_0$ for the next clock cycle. Block 62 also determines the "expected sample value" (i.e., maximum likely sample value) $\tilde{Y}_0$ in accordance with the logic of section 3 of Table 3.

Programmable or Adaptive Decoding

The decoder 23 structure provides considerable flexibility towards adaptation to the signal and noise environment. The shape of the analog read signal varies with the radius on the surface of a magnetic disk. It is also affected by the head geometry, which often shows up as an unavoidable nonsymmetry of the positive and negative readback pulses. These and other anomalies in the signal shape cannot always be completely compensated for by the filter 12 or equalizer 22, or at least it is not always easy and cost effective to attempt to do so.

Decoder 23 provides means to compensate for this misequalization. For this purpose, the read responses to the positive and negative magnetic transitions are characterized by the pulses with sample values .. 0, α_1, β_1,γ_1, 0, .. and .. 0, $-\alpha_2$,$-\beta_2$, $-\gamma_2$, 0 .., respectively (see Fig. 2B). In other words, α_1, β_1, γ_1, $-\alpha_2$, $-\beta_2$, $-\gamma_2$ defines the signal shape for accommodating nonlinearities and small variations around the nominal sample values which, as illustratively assumed in Fig. 2B and previous description, are given by $\alpha_1=\alpha_2=2$, $\beta_1=\beta_2=4$ and $\gamma_1=\gamma_2=2$

In Table 4, the programmable values of the threshold constants for each test are expressed as a function of the signal shape given by the general parameters α_1, β_1, γ_1, $-\alpha_2$, $-\beta_2$, $-\gamma_2$. In event of nonsymmetry in the shape of positive and negative readback pulses, the threshold constants for the corresponding tests in the two phases of the decoder (in Tables 1 and 2, respectively) will not always be equal.

In Table 4, the threshold constants for the positive phase are indicated with a plus sign, and those for the negative phase are indicated with a minus sign. For example, the test corresponding to Boolean variable A_1 will use the constant $K(A_1+)$ for the states in the positive phase (states with $c_0=0$) and the constant $K(A_1-)$ for the states in the negative phase (states with $C_0=1$).

Thus, in Fig. 4, the comparator circuit corresponding to the Boolean variable A will be supplied with two different values for the constant operand, namely $K(A_1+)$ and $K(A_1-)$, and the output will be gated selectively according to whether the condition is $c_0=0$ or $c_0=1$, as shown in Fig. 5. (The sign change block 45 is then not required.) The comparator circuit for each of the eleven Boolean variables (namely, A_1, A_2, B_1, B_2, X_1, X_2, Y_1, Y_2, Z_1, Z_2 and p) will be similarly modified to receive two different values for the constant operand, and the output will be gated selectively according to whether the condition is $c_0=0$ or $c_0=1$.

According to an important feature of the invention, when anomalies in the signal shape cannot be compensated for by a filter and an equalizer, such as 12 and 22, the programmable decoder 23 provides an easy means to compensate for misequalization by adjusting the thresholds in the two phases of the decoder. This flexibility permits operating decoder 23 at near-optimum performance even in the absence of ideal shape of the readback signal. In disk files, this is especially important because the programmability of the decoder can be used to change threshold constants at different tracks or bands of tracks and obtain near-optimum performance at all tracks.

According to another feature of the invention, programmable decoder 23 may be used with a partially equalized or even unequalized read signal in a recording channel.

The programmability of decoder 23 also enables it to continuously adaptively adjust digital parameters in the decoder. More specifically, the estimated sample values at the output of decoder 23 can be monitored in comparison with the received sample values. When averaged over a preselected number of like sample values, the signal shape can be characterized in terms of the parameters α_1, β_1, γ_1, $-\alpha_2$, $-\beta_2$, $-\gamma_2$. This information can be used to obtain the corresponding threshold constants (Table4) and adaptively adjust the decoder for variations in the signal shape.

Also with this improved decoder structure, the decoder threshold values can be easily verified through an extensive amount off testing, and then be adjusted in accordance with the empirical results to accommodate specific anomalies in the signal.

In particular, the first set of preselected thresholds corresponding to the baseline checks A_1,A_2, B_1, B_2 is used for

controlling the missing bit and extra bit errors caused by media defects-related anomalies in the incoming analog signal; the second set of preselected thresholds corresponding to the peak position checks X_1, X_2, Y_1, Y_2, Z_1, Z_2 is used for controlling peak-shift errors caused by nonlinearities-related anomalies in said signal; and the third set of preselected thresholds corresponding to the phase check p is used for controlling the propagation of errors in the decoding of said signal and creating pointers to such errors. No such flexibility is offered with the conventional Viterbi decoder.

It will now be seen that the improved method and apparatus comprises a decoder for implementing a state dependent look-ahead feature by use of a sequence detection algorithm. Appropriate functions of equalized digital sample values are precomputed for a preselected number of bits ahead of the current bit, then compared against various thresholds in the algorithm, whereupon the digitized sample values are then decoded into (1,7) coded binary data. The decoder is programmable and may be continuously adaptive to adjust digital parameters in the decoder. Finally, the benefits of partial response signalling are obtained without requiring a notch filter and the benefits of Viterbi decoding are achieved without the complexity of the conventional trellis-type Viterbi decoder.

Table 1
Decoding Equations for Positive Phase

CURRENT STATE $a_0\ b_0\ c_0$	CURRENT STATE $\tilde{y}_{-1}$	TEST ON SAMPLE VALUES y_0, y_1, y_2, y_3, y_4	DECODED DATA	NEXT STATE $a_1\ b_1\ c_1$	NEXT STATE $\tilde{y}_0$
1 1 0	-2 or 0	NONE	1	1 0 0	-4
1 0 0	-4	$(y_0+y_1)+(y_1+y_2)\le 4$ & $(y_0+y_1)+(y_1+y_2)-y_3\le 4$ or $(y_0+y_1)-(y_2+y_3)+\lceil_{-4}(y_3+y_4)\rceil_{-2}\le -4$	0	0 0 0	-2
		OTHERWISE *	0	0 0 1	0
0 0 0	0 or -2	$(y_0+y_1)+(y_1+y_3)\le 6$ & $(y_0+y_1)+(y_1+y_2)-y_3\le 6$ or $(y_0+y_1)-(y_2+y_3)+\lceil_{-4}(y_3+y_4)\rceil_{-2}\le -2$	0	0 0 0	0
		OTHERWISE *	0	0 0 1	2
1 0 0 0 0 0	-	* PHASE CHECK: $(y_1+y_2)+(y_2+y_3)\le -7$	ERASURE	1 1 1 1 1 1	?

* Phase check overrides all other tests.
$\lceil \rceil_a^b$ means the enclosed function is bounded below and above by a and b respectively.

Table 2
Decoding Equations for Negative Phase

CURRENT STATE $a_0\ b_0\ c_0$	$\tilde{y}_{-1}$	TEST ON SAMPLE VALUES y_0, y_1, y_2, y_3, y_4	DECODED DATA	NEXT STATE $a_1\ b_1\ c_1$	$\tilde{y}_0$
0 0 1	2 or 0	NONE	1	0 1 1	4
0 1 1	4	$-(y_0+y_1)-(y_1+y_2)\le 4$ & $-(y_0+y_1)-(y_1+y_2)+y_3\le 4$ or $-(y_0+y_1)+(y_2+y_3)+\lceil_{-4}\ -(y_3+y_4)\ \rceil_{-2}\le -4$	0	1 1 1	2
		OTHERWISE *	0	1 1 0	0
1 1 1	0 or 2	$-(y_0+y_1)-(y_1+y_2)\le 6$ & $-(y_0+y_1)-(y_1+y_2)+y_3\le 6$ or $-(y_0+y_1)+(y_2+y_3)+\lceil_{-4}\ -(y_3+y_4)\ \rceil_{-2}\le -2$	0	1 1 1	0
		OTHERWISE *	0	1 1 0	-2
0 1 1 1 1 1	-	* PHASE CHECK: $-(y_1+y_2)-(y_2+y_3)\le -7$	ERASURE	0 0 0 0 0 0	?

* Phase check overrides all other tests.
$\lceil_a\ \rceil_b$ means the enclosed function is bounded below and above by a and b respectively.

Table 3
LOGIC FUNCTIONS for Decoder Output

1. Decoded Data and Pointer: (two output lines)

$d_0 = a_0 b_0 \bar{c}_0 + \bar{a}_0 \bar{b}_0 c_0$ Decoded data

$P = p(b_0 c_0 + \bar{b}_0 \bar{c}_0)$ Erasure pointer

2. Next State: (three output lines)

$a_1 = P\bar{b}_0 + \bar{P}b_0$

$b_1 = P\bar{b}_0 + \bar{P}c_0$

$c_1 = P\bar{b}_0 + \bar{P}(c_0\bar{b}_0 + R_1\bar{a}_0 b_0 + \bar{R}_1 a_0 \bar{b}_0 + R_2 a_0 c_0 + \bar{R}_2 \bar{a}_0 \bar{c}_0)$

where $R_1 = A_1 B_1 + X_1 + Y_1 Z_1$

$R_2 = A_2 B_2 + X_2 + Y_2 Z_2$

3. Detected Sample Value $\tilde{y}_0$: (five output lines)

$\bar{a}_0\bar{b}_0 c_0 = 1$	$\Rightarrow \tilde{y}_0 = 4$
$a_0 b_0 \bar{c}_0 = 1$	$\Rightarrow \tilde{y}_0 = -4$
$R_1\bar{a}_0 b_0 + \bar{R}_2\bar{a}_0\bar{c}_0 = 1$	$\Rightarrow \tilde{y}_0 = 2$
$R_1 a_0\bar{b}_0 + \bar{R}_2 a_0 c_0 = 1$	$\Rightarrow \tilde{y}_0 = -2$
$\bar{R}_1\bar{a}_0 b_0 + \bar{R}_1 a_0\bar{b}_0 + R_2 a_0 c_0 + R_2\bar{a}_0\bar{c} = 1$	$\Rightarrow \tilde{y}_0 = 0$

Table 4

Programmed Values for Thresholds

TEST	PROGRAMMED VALUE OF THRESHOLD		NOMINAL VALUE
	Positive Phase ($c_0 = 0$)	Negative Phase ($c_0 = 1$)	
Base-Line Check			
A1	$K(A1+) = .5(\alpha_1+2\beta_1+\gamma_1)-\gamma_2$	$K(A1-) = .5(\alpha_2+2\beta_2+\gamma_2)-\gamma_1$	4
A2	$K(A2+) = .5(\alpha_1+2\beta_1+\gamma_1)$	$K(A2-) = .5(\alpha_2+2\beta_2+\gamma_2)$	6
B1	$K(B1+) = .5(\alpha_1+2\beta_1+\gamma_1)-\gamma_2$	$K(B2-) = .5(\alpha_2+2\beta_2+\gamma_2)-\gamma_1$	4
B2	$K(B2+) = .5(\alpha_1+2\beta_1+\gamma_1)$	$K(B1-) = .5(\alpha_2+2\beta_2+\gamma_2)$	6
Peak Position Check			
X1	$K(X1+) = \alpha_1-\gamma_1-\gamma_2$	$K(X1-) = \alpha_2-\gamma_2-\gamma_1$	-2
X2	$K(X2+) = \alpha_1-\gamma_1$	$K(X2-) = \alpha_2-\gamma_2$	0
Y1	$K(Y1+) = \alpha_1-\gamma_1+\alpha_2-\gamma_2$	$K(Y1-) = \alpha_2-\gamma_2+\alpha_1-\gamma_1$	0
Y2	$K(Y2+) = \alpha_1-\gamma_1+\alpha_2$	$K(Y2-) = \alpha_2-\gamma_2+\alpha_1$	2
Z1	$K(Z1+) = \alpha_1-.5(\alpha_2+\beta_2+\gamma_2)-\gamma_2$	$K(Z1-) = \alpha_2-.5(\alpha_1+\beta_1+\gamma_2)-\gamma_1$	-4
Z2	$K(Z2+) = \alpha_1-.5(\alpha_2+\beta_2+\gamma_1)$	$K(Z2-) = \alpha_2-.5(\alpha_1+\beta_1+\gamma_2)$	-2
Phase Check			
p	$K(p+) = -K(A2-) \times 7/8$	$K(p-) = -K(A2+) \times 7/8$	-7

Note: $\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2$ **are signal-shape parameters.**

Claims

1. A method of processing, in a signal processing channel, digital values corresponding to a digitized incoming analog signal representative of coded binary data, said method being characterised by:

   using a state-dependent sequence detection algorithm iterated by successive clock cycles to advance from a current state (a_0 b_0 c_0) to a next state (a_1 b_1 c_1),

   precomputing functional expressions (30 to 40) of digital sample values (y_0 to y_4) for a preselected number of values ahead of the current value (y_0);

   comparing preselected ones (36 to 40) of said expressions (30 to 40) against corresponding thresholds (51 to 61) to provide respective binary decision outputs (P to Z2);

   decoding the state values corresponding to the current state (a_0 b_0 c_0) into one bit of coded binary data (d_0);

   using said outputs (P to Z2) in conjunction with state values corresponding to the current state (a_0 b_0 c_0), determining state values for the next state (a_1 b_1 c_1), which become the current state values for the next-iteration of the clock cycle.

* C L M S *

Table 4

Programmed Values for Thresholds

TEST	PROGRAMMED VALUE OF THRESHOLD		NOMINAL VALUE
	Positive Phase ($c_0 = 0$)	Negative Phase ($c_0 = 1$)	
Base-Line Check			
A1	$K(A1+) = .5(\alpha_1+2\beta_1+\gamma_1)-\gamma_2$	$K(A1-) = .5(\alpha_2+2\beta_2+\gamma_2)-\gamma_1$	4
A2	$K(A2+) = .5(\alpha_1+2\beta_1+\gamma_1)$	$K(A2-) = .5(\alpha_2+2\beta_2+\gamma_2)$	6
B1	$K(B1+) = .5(\alpha_1+2\beta_1+\gamma_1)-\gamma_2$	$K(B2-) = .5(\alpha_2+2\beta_2+\gamma_2)-\gamma_1$	4
B2	$K(B2+) = .5(\alpha_1+2\beta_1+\gamma_1)$	$K(B1-) = .5(\alpha_2+2\beta_2+\gamma_2)$	6
Peak Position Check			
X1	$K(X1+) = \alpha_1-\gamma_1-\gamma_2$	$K(X1-) = \alpha_2-\gamma_2-\gamma_1$	-2
X2	$K(X2+) = \alpha_1-\gamma_1$	$K(X2-) = \alpha_2-\gamma_2$	0
Y1	$K(Y1+) = \alpha_1-\gamma_1+\alpha_2-\gamma_2$	$K(Y1-) = \alpha_2-\gamma_2+\alpha_1-\gamma_1$	0
Y2	$K(Y2+) = \alpha_1-\gamma_1+\alpha_2$	$K(Y2-) = \alpha_2-\gamma_2+\alpha_1$	2
Z1	$K(Z1+) = \alpha_1-.5(\alpha_2+\beta_2+\gamma_2)-\gamma_2$	$K(Z1-) = \alpha_2-.5(\alpha_1+\beta_1+\gamma_2)-\gamma_1$	-4
Z2	$K(Z2+) = \alpha_1-.5(\alpha_2+\beta_2+\gamma_1)$	$K(Z2-) = \alpha_2-.5(\alpha_1+\beta_1+\gamma_2)$	-2
Phase Check			
p	$K(p+) = -K(A2-) \times 7/8$	$K(p-) = -K(A2+) \times 7/8$	-7

Note: $\alpha_1, \beta_1, \gamma_1, -\alpha_2, -\beta_2, -\gamma_2$ **are signal-shape parameters.**

Claims

1. A method of processing, in a signal processing channel, digital values corresponding to a digitized incoming analog signal representative of coded binary data, said method being characterised by:

   using a state-dependent sequence detection algorithm iterated by successive clock cycles to advance from a current state (a_0 b_0 c_0) to a next state (a_1 b_1 c_1),

   precomputing functional expressions (30 to 40) of digital sample values (y_0 to y_4) for a preselected number of values ahead of the current value (y_0);

   comparing preselected ones (36 to 40) of said expressions (30 to 40) against corresponding thresholds (51 to 61) to provide respective binary decision outputs (P to Z2);

   decoding the state values corresponding to the current state (a_0 b_0 c_0) into one bit of coded binary data (d_0);

   using said outputs (P to Z2) in conjunction with state values corresponding to the current state (a_0 b_0 c_0), determining state values for the next state (a_1 b_1 c_1), which become the current state values for the next-iteration of the clock cycle.

* C L M S *

2. The method of claim 1, including the step of: generating a programmed value for each threshold (51 to 61).

3. The method of claim 1, including the steps of:

determining an expected sample value ($\tilde{Y}_0$) from the decision outputs and the state values corresponding to the current state (a_0 b_0 c_0); and

modifying the thresholds (51 to 61) according to the expected sample value ($\tilde{Y}_0$)

4. The method of claim 1, including the steps of: generating a programmed value for each threshold 51 to 61; and adaptively modifying the programmed values according to the change in each sample value in the incoming analog signal as compared to a corresponding expected sample value ($\tilde{Y}_0$).

5. The method of claim 1, including the step of:

computing each threshold (51 to 61) from digitized parameters corresponding to the analog signal shape.

6. The method of claim 5, wherein the signal shape is equalized and the parameters of the analog signal shape are .., 0, 2, 4, 2, 0, .. and .., 0, -2, -4, -2, 0, ... normalized by a scaling factor.

7. The method of claim 5, wherein the parameters of the analog signal shape are .., 0 , α_1, β_1, γ_1, 0 , and .., 0, $-\alpha_2$, $-\beta_2$, $-\gamma_2$ 0, ...

8. The method of claim 1, including the step of: employing different thresholds for positive and negative phases of the analog signal shape as necessary to accommodate asymmetry of the signal shape.

9. The method of claim 1, including during the precomputing and comparing steps: precomputing a baseline check and comparing it against a first preselected set (52 to 55) of said thresholds for controlling missing bit and extra-bit errors caused by defects.

10. The method of claim 1, including during the precomputing and comparing steps:

precomputing a peak position check and comparing it against a second preselected (56 to 61) set of said thresholds for controlling peak-shift errors caused by nonlinearities in the signal.

11. The method of claim 1, including during the precomputing and comparing steps:

precomputing a phase check (51) and comparing it against a third preselected set of said thresholds for controlling the propagation of errors and creating pointers to such errors.

12. The method of claim 9, wherein said first set (52 to 55) of thresholds is programmable.

13. The method of claim 10, wherein said second set (56 to 61) of thresholds is programmable.

14. The method of claim 11, wherein said third set (51) of thresholds is programmable.

15. The method of claim 1, including during the precomputing and comparing steps:

precomputing a baseline check and comparing it against a first preselected set (52 to 55) of said thresholds for controlling missing bit and extra-bit errors caused by defects;

precomputing a peak position check and comparing it against a second preselected set (56 to 61) of said thresholds for controlling peak-shift errors caused by nonlinearities in the signal; and

precomputing a phase check and comparing it against a third preselected set (51) of said thresholds for controlling the propagation of errors and creating pointers to such errors, each said preselected set of thresholds

* C L M S *

differing from those of the other sets.

16. The method of claim 1, including the steps of:

generating a programmed value for each threshold (51 to 61); and

programmably modifying values for the respective thresholds for different tracks on a disk file to optimize performance for all tracks.

17. The method of claim 1, wherein the coded binary data is in a run-length-limited code.

18. The method of claim 17, wherein the run-length-limited code is a (1,7).

19. The method of claim 1, including prior to use of the sequence algorithm, the steps of:

converting a partially equalized analog signal representative of coded binary data into a sequence of digital sample values at successive clock times; and

further equalizing the digital sample values to correspond to a predetermined analog shape.

20. The method of claim 1, including prior to use of the sequence algorithm, the steps of:

equalizing the incoming analog signal to correspond to a predetermined analog shape; and

then converting the equalized analog signal representative of coded binary data into a sequence of digital sample values at successive clock times.

Patentansprüche

1. Ein Verfahren zur Verarbeitung digitaler Werte in einem Signalverarbeitungskanal, die einem eingehenden, digitalisierten, analogen Signal entsprechen, das codierte Binärdaten darstellt, wobei das Verfahren gekennzeichnet ist durch:

die Verwendung eines statusabhängigen Sequenzerfassungsalgorithmus, der von aufeinanderfolgenden Taktzyklen wiederholt wird, um von einem aktuellen Status (a_0 b_0 c_0) zum nächsten Status (a_1 b_1 c_1) zu gehen,

die Vorausberechnung funktionaler Ausdrücke (30 bis 40) digitaler Beispielwerte (y_0 bis y_4) für eine festgelegte Zahl von Werten im voraus vor dem aktuellen Wert (y_0),

den Vergleich festgelegter einzelner (36 bis 40) Ausdrücke (30 bis 40) mit entsprechenden Schwellen (51 bis 61), um entsprechende Binärentscheidungsausgänge (P bis Z2) zu erhalten,

das Decodieren der Statuswerte, die dem aktuellen Status (a_0 b_0 c_0) entsprechen, in ein Bit der codierten Binärdaten (d_0),

die Verwendung der Ausgänge (P bis Z2) zusammen mit den Statuswerten, die dem aktuellen Status (a_0 b_0 c_0) entsprechen, das Bestimmen der Statuswerte für den nächsten Status (a_1 b_1 c_1), die zu den aktuellen Statuswerten für die nächste Iteration des Taktzyklusses werden.

2. Das Verfahren nach Anspruch 1 mit folgendem Schritt: dem Erzeugen eines programmierten Wertes für jede Schwelle (51 bis 61).

3. Das Verfahren nach Anspruch 1 mit folgenden Schritten:

der Bestimmung eines erwarteten Beispielwerts ($\tilde{y}_0$) von den Entscheidungsausgängen und den Statuswerten, die dem aktuellen Status (a_0 b_0 c_0) entsprechen, und

* C L M S *

dem Ändern der Schwellen (51 bis 61) gemäß dem erwarteten Beispielwert ($\tilde{Y}_0$).

4. Ein Verfahren nach Anspruch 1 mit folgenden Schritten: dem Erzeugen eines programmierten Wertes für jede Schwelle (51 bis 61), und der adaptiven Modifizierung der programmierten Werte entsprechend der Änderung in jedem Beispielwert des eingehenden analogen Signals laut Vergleich mit einem entsprechenden erwarteten Beispielwert ($\tilde{Y}_0$).

5. Das Verfahren nach Anspruch 1 mit folgendem Schritt: dem Berechnen jeder Schwelle (51 bis 61) von digitalisierten Parametern entsprechend der analogen Signalform.

6. Das Verfahren nach Anspruch 5, wobei die Signalform entzerrt wird und die Parameter der analogen Signalform ... 0, 2, 4, 2, 0... und ... 0, -2, -4, -2, 0 ... sind, die von einem Normalisierungsfaktor normalisiert werden.

7. Das Verfahren nach Anspruch 5, wobei die Parameter der analogen Signalform ...0, α_1, β_1, γ_1, 0 ... und ... 0,-α_2, -β_2, -γ_2, 0 ... sind.

8. Das Verfahren nach Anspruch 1 mit folgendem Schritt: der Verwendung unterschiedlicher Schwellen für positive und negative Phasen der analogen Signalform, um Asymmetrien der Signalform auszugleichen.

9. Das Verfahren nach Anspruch 1, zu dem während der Vorberechnungs- und Vergleichsschritte folgendes hinzukommt: das Vorberechnen einer Impulsbodenprüfung und der Vergleich dieser Prüfung mit einer ersten festgelegten Menge (52 bis 55) der Schwellen, um das fehlende Bit und Zusatzbitfehler infolge von Störungen zu prüfen.

10. Das Verfahren nach Anspruch 1, zu dem während der Vorberechnungs- und Vergleichsschritte folgendes hinzukommt: das Vorberechnen einer Spitzenpositionprüfung und der Vergleich dieser Prüfung mit einer zweiten festgelegten Menge (56 bis 61) der Schwellen, um Spitzenverschiebungsfehler infolge von Nicht-Linearität im Signal zu prüfen.

11. Das Verfahren nach Anspruch 1, zu dem während der Vorberechnungs- und Vergleichsschritte folgendes hinzukommt: das Vorberechnen einer Phasenprüfung (51) und der Vergleich der Prüfung mit einer dritten festgelegten Menge von Schwellen, um die Ausbreitung von Fehlern und das Erzeugen von Zeigern auf die Fehler zu prüfen.

12. Das Verfahren nach Anspruch 9, bei dem die erste Menge (52 bis 55) der Schwellen programmierbar ist.

13. Das Verfahren nach Anspruch 10, bei dem die zweite Menge (56 bis 61) der Schwellen programmierbar ist.

14. Das Verfahren nach Anspruch 11, bei dem die dritte Menge (51) der Schwellen programmierbar ist.

15. Das Verfahren nach Anspruch 1, zu dem während der Vorberechnungs- und Vergleichsschritte folgendes hinzukommt:
das Vorberechnen einer Impulsbodenprüfung und der Vergleich der Prüfung mit einer ersten Menge (52 bis 55) von Schwellen, um das fehlende Bit und Zusatzbitfehler infolge von Störungen zu prüfen;
das Vorberechnen einer Spitzenpositionprüfung und der Vergleich der Prüfung mit einer zweiten festgelegten Menge (56 bis 61) von Schwellen, um Spitzenverschiebungsfehler infolge von Nicht-Linearitäten im Signal zu prüfen; und
das Vorberechnen einer Phasenprüfung und der Vergleich der Prüfung mit einer dritten festgelegten Menge (51) von Schwellen, um die Ausbreitung von Fehlern und das Erzeugen von Zeigern auf die Fehler zu prüfen,
wobei sich jede festgelegte Menge von Schwellen von den anderen Mengen unterscheidet.

16. Das Verfahren nach Anspruch 1 mit folgendem Schritt: dem Erzeugen eines programmierten Wertes für jede Schwelle (51 bis 61), und dem programmierbaren Modifizieren der Werte für die entsprechenden Schwellen für unterschiedliche Spuren auf einer Platte, um die Leistung für alle Spuren zu optimieren.

17. Das Verfahren nach Anspruch 1, bei dem die codierten Binärdaten in einem lauflängenbeschränkten Code sind.

18. Das Verfahren nach Anspruch 17, bei dem der lauflängenbe-schränkte Code ein (1,7)-Code ist.

19. Das Verfahren nach Anspruch 1, zu dem vor der Verwendung des Sequenzalgorithmus noch folgende Schritte

* C L M S *

gehören:
das Umwandeln eines teilweise entzerrten analogen Signals, das die codierten Binärdaten darstellt, in aufeinanderfolgenden Taktzeiten in eine Folge von digitalen Beispielwerten, und
die weitere Entzerrung der digitalen Beispielwerte, damit diese einer festgelegten analogen Form entsprechen.

20. Das Verfahren nach Anspruch 1, zu dem vor der Verwendung des Sequenzalgorithmus noch folgende Schritte gehören: das Entzerren des eingehenden analogen Signals, damit dieses einer festgelegten analogen Form entspricht, und das anschließende Umwandeln des entzerrten analogen Signals, das die codierten Binärdaten darstellt, in aufeinanderfolgenden Taktzeiten in eine Folge von digitalen Beispielwerten.

Revendications

1. Procédé de traitement, dans une voie de traitement de signal, de valeurs numériques correspondant à un signal analogique entrant numérisé représentatif d'une donnée binaire codée, ledit procédé étant caractérisé par les étapes consistant à :

   faire usage d'un algorithme de détection de séquence dépendant d'un état, répété par cycles d'horloge successifs d'itération de façon à avancer à partir d'un état actuel (a_0 b_0 c_0) vers un nouvel état (a_1 b_1 c_1),

   effectuer un précalcul d'expressions fonctionnelles (30 à 40) de valeurs d'échantillonnage numériques (y0 à y4) pour un nombre présélectionné de valeurs précédent la valeur actuelle (y0),

   comparer les expressions présélectionnées (36 à 40) parmi lesdites expressions (30 à 40) par rapport aux seuils correspondants (51 à 61) afin de procurer des sorties de décision binaires respectives (P à Z2),

   décoder les valeurs d'état correspondant à l'état actuel (a_0 b_0 c_0) en un bit de donnée binaire codé (d_0),

   faire usage desdites sorties (P à Z2) en conjonction avec les valeurs d'état correspondant à l'état actuel (a_0 b_0 c_0), en déterminant les valeurs d'état pour le nouvel état (a_1 b_1 c_1) qui deviennent les valeurs d'état actuel pour l'itération suivante de cycle d'horloge.

2. Procédé selon la revendication 1, comprenant l'étape consistant à :

   générer une valeur programmée pour chaque seuil (51 à 61).

3. Procédé selon la revendication 1, comprenant les étapes consistant à :

   déterminer une valeur d'échantillon attendue ($\tilde{\dot{Y}}_0$) à partir des sorties de décision et des valeurs d'état correspondant à l'état actuel (a_0 b_0 c_0),

   modifier les seuils (51 à 61) conformément à la valeur d'échantillon attendue ($\tilde{\dot{Y}}_0$).

4. Procédé selon la revendication 1, comprenant les étapes consistant à :

   générer une valeur programmée pour chaque seuil (51 à 61), et

   modifier de façon adaptative les valeurs programmées conformément àla modification dans chaque valeur d'échantillon dans le signal analogique entrant par comparaison à une valeur d'échantillon attendue correspondante ($\tilde{\dot{Y}}_0$).

5. Procédé selon la revendication 1, comprenant l'étape consistant à :

   calculer chaque seuil (51 à 61) à partir des paramètres numérisés correspondant à la forme du signal analogique.

* C L M S *

6. Procédé selon la revendication 5, dans lequel la forme du signal est corrigée et les paramètres de la forme de signal analogique sont .., 0, 2, 4, 2, 0, .. et .., 0, - 2, - 4, - 2, 0, ... normalisés par un facteur d'échelle.

7. Procédé selon la revendication 5, dans lequel les paramètres de la forme de signal analogique sont .., 0, α_1, β_1, γ_1, 0, .. et .., 0, - α_2, - β_2, -γ_2, 0,..

8. Procédé selon la revendication 1, comprenant l'étape consistant à :

employer différents seuils pour des phases positive et négative de la forme de signal analogique comme il est nécessaire pour s'adapter à l'asymétrie de la forme de signal.

9. Procédé selon la revendication 1, comprenant durant les étapes de précalcul et de comparaison :

le précalcul d'une vérification de base et la comparaison de celle-ci avec un premier ensemble présélectionné (52 à 55) parmi lesdits seuils afin de contrôler les erreurs de bits absents et de bits supplémentaires provoquées par des défauts.

10. Procédé selon la revendication 1, comprenant pendant les étapes de précalcul et de comparaison :

le précalcul d'une vérification de position de pic et la comparaison de celle-ci avec un deuxième ensemble présélectionné (56 à 61) parmi lesdits seuils afin de contrôler les erreurs de décalage de pic provoquées par des non linéarités dans le signal.

11. Procédé selon la revendication 1, comprenant pendant les étapes de précalcul et de comparaison :

le précalcul d'une vérification de phase (51) et la comparaison de celle-ci avec un troisième ensemble présélectionné parmi lesdits seuils afin de contrôler la propagation des erreurs et la création de pointeurs sur de telles erreurs.

12. Procédé selon la revendication 9, dans lequel ledit premier ensemble (52 à 55) de seuils est programmable.

13. Procédé selon la revendication 10, dans lequel ledit second ensemble de seuils (56 à 61) est programmable.

14. Procédé selon la revendication 11, dans lequel ledit troisième ensemble (51) de seuils est programmable.

15. Procédé selon la revendication 1, comprenant pendant les étapes de précalcul et de comparaison :

le précalcul d'une vérification de base et la comparaison de celle-ci avec un premier ensemble présélectionné (52 à 55) parmi lesdits seuils afin de contrôler les erreurs de bits absents et de bits supplémentaires provoquées par des défauts,

le précalcul d'une vérification de position de pic et la comparaison de celle-ci avec un deuxième ensemble présélectionné (56 à 61) parmi lesdits seuils afin de contrôler les erreurs de décalage de pic provoquées par des non linéarités dans le signal, et

le précalcul d'une vérification de phase et la comparaison de celle-ci avec un troisième ensemble présélectionné (51) parmi lesdits seuils afin de contrôler la propagation des erreurs et la création de pointeurs sur de telles erreurs, chacun desdits ensembles présélectionnés de seuils étant différent des autres ensembles.

16. Procédé selon la revendication 1, comprenant les étapes consistant à :

générer une valeur programmée pour chaque seuil (51 à 61), et

modifier de façon programmable les valeurs des seuils respectifs pour des pistes différentes sur un fichier de disque afin d'optimiser les performances pour toutes les pistes.

17. Procédé selon la revendication 1, dans lequel les données binaires codées sont dans un codage par limitation de la longueur de déroulement (codage RLL).

* C L M S *

18. Procédé selon la revendication 17, dans lequel le codage par limitation de la longueur de déroulement (codage RLL) est un codage (1, 7).

19. Procédé selon la revendication 1, comprenant avant l'utilisation de l'algorithme de séquence, les étapes consistant à :

convertir un signal analogique corrigé partiellement représentatif de données binaires codées en une séquence de valeurs d'échantillonnage numériques à des instants d'horloge successifs, et

corriger davantage les valeurs d'échantillonnage numériques afin de correspondre à une forme analogique prédéterminée.

20. Procédé selon la revendication 1, comprenant avant l'utilisation de l'algorithme de séquence, les étapes consistant à :

corriger le signal analogique entrant afin de correspondre à une forme analogique prédéterminée, et

convertir alors le signal analogique corrigé représentatif de données binaires codées en une séquence de valeurs d'échantillonnage numériques à des instants d'horloge successifs.