

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
6 March 2003 (06.03.2003)

PCT

(10) International Publication Number  
**WO 03/019793 A2**

(51) International Patent Classification<sup>7</sup>: **H03M 13/41**

(21) International Application Number: PCT/EP02/09443

(22) International Filing Date: 23 August 2002 (23.08.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
01120383.3 25 August 2001 (25.08.2001) EP

(71) Applicant (for all designated States except US): **THOMSON LICENSING S.A.** [FR/FR]; 46 Quai A. le Gallo, F-92100 Boulogne-Billancourt (FR).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **KRAVTCHEIKO, Alexander** [DE/DE]; Deutenbergring 146, 78056 Villingen-Schwenningen (DE).

(74) Agent: **THIES, Stephan**; DEUTSCHE THOMSON-BRANDT GMBH, European Patent Operations, Karl-Wiechert-Allee 74, 30625 Hannover (DE).

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

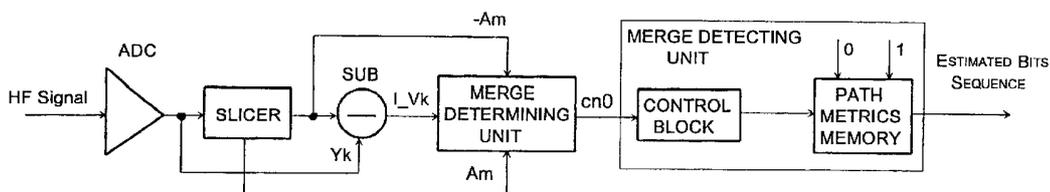
(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: MAXIMUM LIKELIHOOD DETECTOR



(57) Abstract: The present invention relates to a method and an apparatus to transform a signal received from a channel into a binary code sequence, and more particularly to a signal processing apparatus, which can perform the maximum likelihood detection of the reproduced data from an optical disk. A new algorithm is used, which has shown to be both, more precise than the known solution and independent of signal amplitudes. This procedure has the advantage that no selective amplification is necessary for Maximum Likelihood Detection. The general idea of the invention is described as follows: - known algorithms only work properly if (value >1), but not if the (value <1) - according to the invention: in case that (value <1), invert conditions are required - this will lead to correct results for value ><1 and sometimes to a wrong "first" result, which has no critical influence.



WO 03/019793 A2

**Maximum likelihood detector**Field of the invention

5 The present invention relates to the field of signal processing, and in particular, to a maximum likelihood detector for Viterbi detection.

Background and prior art

10

Communication of voice and data signals is often accomplished by converting analog signals to digital signals. These digital signals are then transmitted from a transmitting device to a receiving device, converted back to analog, 15 if necessary, and communicated to a user. This digital transmission is often performed through analog channels. Digital information is transmitted in the form of a "symbol" representing a digital value. In some cases, adjacent symbols can overlap, resulting in a phenomenon known as intersymbol interference. This interference can corrupt a digital 20 transmission, leading to errors in the receipt of the digital information.

The same problem occurs when digital data is read from a recording medium, such as an optical disc. 25

In a recording channel, a method for decoding the binary symbol sequence that is outputted from the channel in its corrupted form is required. Maximum-likelihood sequence estimation (MLSE) decoding has been employed in the past as an 30 effective tool in pulse detectors for receiving and decoding digital transmissions that suffer from intersymbol interference.

35 Using partial response signaling allows a better handling of intersymbol interference and allows a more efficient utilization

zation of the bandwidth of a given channel. In partial response systems, a controlled amount of intersymbol interference can be allowed. The partial response system is described by the polynomials  $1+D$ ,  $1-D$  and  $(1-D^2)$ , also called  
5 duobinary, dicode, and class-IV, respectively.

Class IV partial response waveforms are formed by the subtraction of binary waveforms two bit intervals apart. This process boosts midband frequencies making the system more  
10 immune to noise and distortion at both high and low frequencies. This is especially useful in a magnetic recording channel where, using a conventional inductive head, there is little signal at low frequencies and spacing losses can cause large attenuation at high frequencies.

15

Because class IV partial response signaling for digital detection is especially suited for the magnetic recording channel, sampled amplitude detection can be applied for magnetic recording. To minimize the propagation of data errors,  
20 the signal is turned into a sequence of binary numbers. Procedures for determining the maximum likelihood sequence in the presence of noise can then be applied. With sequence detection, sequences of bits are detected and processed to minimize error.

25

Maximum likelihood sequence estimation, in particular, the Viterbi algorithm, is used in improving the detection of symbol (pulse) sequences in the presence of noise and intersymbol interference. MLSE is described by G.D. Forney in  
30 "The Viterbi Algorithm," Proceedings of the IEEE, Vol. 61, No. 3, March, 1973, pp. 268-278, and by R. W. Wood, etc., in "Viterbi Detection of Class IV Partial Response on a Magnetic Recording Channel," IEEE transactions on communications, Vol. COM-34, No. 5, May, 1986, pp. 454-461.

35

One of the earliest references (Kobayashi, "Application of

Probabilistic Decoding to Digital Magnetic Recording Systems," IBM journal of Research and Development, Vol. 15, No. 1, January 1971, pp. 64-74) to the use of the Viterbi algorithm on the intersymbol interference problem was related to  
5 Class IV Partial Response on the Magnetic Recording Channel. In essence, the algorithm provides an iterative method of determining the "best" route along the branches of a trellis.

10 If, for each branch, a "metric" is calculated which corresponds to the logarithm of the probability of that branch, then the Viterbi algorithm can be used to determine the path which accumulates the highest log probability, i.e., the maximum likelihood sequence. In essence, given a received  
15 sequence  $(a_n)$  where  $n$  is an integer time index, choose from among all possible transmitted sequences  $(b_n)$  the one which is most likely to cause  $(a_n)$  to be received, i.e., choose  $(b_n)$  to maximize  $P((a_n)|(b_n))$ .

20 With the Viterbi algorithm, data is not decoded as soon as it is received. Instead, a sequence of data, having a predetermined coding depth following the digit to be decoded, is first collected. Then, by computing the path metrics, a limited number of possible messages are selected, each extending  
25 throughout the decoding depth far beyond the digit presently to be decoded, with one such survivor sequence ending in each of the data states.

The correlation between each survivor sequence and the data  
30 actually received is computed for the entire decoding depth under consideration. The highest correlated of the survivor sequences is selected to be the sole survivor sequence. The earliest of the received digital digits within the decoding depth is then permanently decoded under the temporary as-  
35 sumption that the sole survivor sequence is the correct sequence.

The MLSE problem is similar to the problem of finding the shortest route through a certain graph. The Viterbi algorithm arises as a natural recursive solution. This algorithm is often associated with a state diagram which can be illustrated by a trellis. In a two-state trellis, each node represents a distinct state at a given time, and each branch represents a transition to some new state at the next instant of time.

10

Partial response class IV waveforms can be considered as two independent interleaved duobinary (1-D) sequences and each sequence can be decoded independently. In his paper entitled "Optimal Reception for Binary Partial Response Channels", in The Bell System Technical Journal, volume 51, number 2, pp. 493-505, published in February 1972, M. J. Ferguson proposes a simplified method for Viterbi detection for binary partial response channels (e.g. 1-D).

20

Straightforward Viterbi detection needs to keep track of both probabilities going into the two states, and the ranges of the probabilities are not bounded. Ferguson's method only needs to keep track of the difference between the two probabilities and the value of the difference is well-bounded.

25

Two prior art examples of applying the Viterbi algorithm for decoding magnetic recording channel outputs are U.S. Pat. No. 4,644,564 by Dolivo et al. and U.S. Pat. No. 4,087,787 by Acampora. Dolivo et al. disclose a method that uses two survivor sequences and the difference metric between the two metrics is processed. In Dolivo et al., as each sample is received, the prior art patent recursively determines the new pair of survivor sequences and the new difference metric. Dolivo et al. is implemented in a digital format that suffers in terms of speed and requires more electronics to implement. Furthermore, resetting of the voltage signals

30  
35

does not occur automatically as is desired.

US patent no. 5,917,859 shows a Viterbi detector for use in a partial-response maximum-likelihood (PRML) channel. A sampled data Viterbi detector compares a sampled analog input signal with two threshold signals. The binary outputs of the comparing means are then provided to a survival sequence register, as well as being used to formulate new threshold signals for the subsequent input sample. The hardware implements Ferguson's method for calculating sequence metrics by representing the accumulated metric difference as two threshold signals.

Another method and hardware architecture for implementing this technique have been described in "A PRML DETECTOR FOR A DVDR SYSTEM", IEEE Transactions On Consumer Electronics, Vol. 45, No. 2, MAY 1999. This method requires the recalculation of the threshold level "1" depending on the signal amplitude, the automatic gain control (AGC) and adaptive equalizer.

A known solution of the Viterbi detector (VD) applied to target (1+D) partial response has been described in "Digital Baseband Transmission and Recording" (Solutions Manual to Accompany, pp. 61) by J. W. M. Bergmans, 1998.

This solution is:

1. Difference metrics

The VD has two states  $s_-$  and  $s_+$  with their associated path metrics  $\lambda_{k+}$  and  $\lambda_{k-}$ . The corresponding trellis diagram is shown in Fig. 1.

There are 4 branches with metrics  $\beta_{k++}$ ,  $\beta_{k-+}$ ,  $\beta_{k--}$ ,  $\beta_{k+-}$ . The surviving path to  $s_{k+1}$  is determined by comparing the metrics of the completing paths via  $s_{k-}$  and  $s_{k+}$ , i.e.

$$\lambda_{k+1-} = \min (\lambda_{k-} + \beta_{k-}, \lambda_{k+} + \beta_{k+}) \quad (1)$$

A similar comparison occurs for state  $s_{k+1+}$

5

$$\lambda_{k+1+} = \min (\lambda_{k-} + \beta_{k+}, \lambda_{k+} + \beta_{k++}). \quad (2)$$

What is of interest in these comparisons, is the largest metric. Thus, only the difference between metrics is of concern. In order to exploit this fact, the difference path metric was defined

10

$$\text{del}k = (\lambda_{k-}) - (\lambda_{k+}), \quad (3)$$

15 along with metric increments

$$Q_{k+1-} = (\lambda_{k+1-}) - (\lambda_{k+}) \quad (4)$$

and

$$Q_{k+1+} = (\lambda_{k+1+}) - (\lambda_{k+}) \quad (5)$$

20 It is easy to express (1) and (2) in terms of these quantities upon subtracting  $\lambda_{k+}$  from the left and right hand sides. The result is

$$Q_{k+1-} = \min (\text{del}k + \beta_{k-}, \beta_{k+}) \quad (6)$$

$$25 \quad Q_{k+1+} = \min (\text{del}k + \beta_{k+}, \beta_{k++}) \quad (7)$$

As expected, the absolute values of  $\lambda_{k+}$  and  $\lambda_{k-}$  will not be taken into consideration. Furthermore, subtraction of both minima yields the new difference path metric:

30

$$Q_{k+1-} - Q_{k+1+} = (\lambda_{k+1-} - \lambda_{k+}) - (\lambda_{k+1+} - \lambda_{k+}) = \text{del}k+1 \quad (8)$$

In summary, based on the old difference path metric  $\text{del}k$  and

branch metrics  $\beta_{k++}, \dots, \beta_{k--}$ , survivors for time  $k+1$  can be determined, according to (6) and (7).

These comparisons further yield minima  $Q_{k+1-}$  and  $Q_{k+1+}$  ... ,  
 5 whose difference determines the new difference path metric  $\delta_{k+1}$ .

Thus, the entire detection process is cast in terms of a single difference metric, as opposed to two metrics  $\lambda_{k+}$  and  
 10  $\lambda_{k-}$  in the standard VD.

Consequently, difference path metrics can be used for any two-state VD.

15 A simplification is possible for the (1+D) partial response channel by exploiting the simple structure of partial response branch metrics.

This will lead to Ferguson's algorithm.

20 VD has input

$$Z_k = X_k + \gamma_k,$$

where

$$X_k = a_k + a_{k-1}$$

and

25  $\gamma_k$  is noise.

In the trellis diagram of Fig. 2 the transitions between the states are noted with their corresponding values of noiseless channel output  $x$ .

30 For a (1+D) channel there are the following branch metrics

$$\beta_{k--} = [Z_k + 2]^2,$$

$$\beta_{k-+} = \beta_{k+-} = Z_k^2$$

and

$$\beta_{k++} = [Z_k - 2]^2.$$

There are three possible path extensions:

5 1. Negative merge (m-):

This occurs when

$$\text{del}_k + \beta_{k-+} < \text{del}_{k++} \text{ and } \text{del}_k + \beta_{k--} < \beta_{k+-},$$

i. e. when

$$\text{del}_k < -4z_{k+4} \text{ and } \text{del}_k < -4z_{k-4}.$$

10

The latter condition is the strongest.

The new difference metric (8) amounts to

$$\begin{aligned} \text{del}_{k+1} &= (\lambda_{k+1-}) - (\lambda_{k+1+}) \\ &= (\lambda_{k-} + \beta_{k--}) - (\lambda_{k-} + \beta_{k+-}) \\ &= 4z_{k+4} \end{aligned} \quad (9)$$

15

2. Cross-over (mx):

This occurs when

$$20 \quad \text{del}_k + \beta_{k-+} < \beta_{k++} \text{ and } \text{del}_k + \beta_{k--} > \beta_{k+-},$$

i. e. when

$$\text{del}_k < -4z_{k+4} \text{ and } \text{del}_k > -4z_{k-4}.$$

The new difference metric is

$$\begin{aligned} 25 \quad \text{del}_{k+1} &= (\lambda_{k+1-}) - (\lambda_{k+1+}) \\ &= (\lambda_{k+} + \beta_{k+-}) - (\lambda_{k-} + \beta_{k+-}) \\ &= \lambda_{k+} - \lambda_{k-} \\ &= -\text{del}_k \end{aligned} \quad (10)$$

30

3. Positive merge (m+):

This occurs when

$$\text{delk} + \beta_{\kappa-+} > \beta_{\kappa++} \text{ and } \text{delk} + \beta_{\kappa--} > \beta_{\kappa+-}$$

i. e. when

$$5 \quad \text{delk} > -4z_{\kappa+4} \text{ and } \text{delk} > -4z_{\kappa-4}.$$

The former condition is strongest.

The new difference metric (8) amounts to

$$10 \quad \begin{aligned} \text{delk}+1 &= (\lambda_{\kappa+1-}) - (\lambda_{\kappa+1+}) \\ &= (\lambda_{\kappa+} + \beta_{\kappa+-}) - (\lambda_{\kappa+} + \beta_{\kappa++}) \\ &= 4z_{\kappa-4}. \end{aligned} \quad (11)$$

It is obviously convenient to record and update

$$\text{delk}' = \text{delk}/4.$$

15

The required actions are summarized in the algorithm:

If( $\text{delk}' < -z_{\kappa-1}$ )

$$\text{then } \text{delk}+1' = z_{\kappa+1} \quad (\text{m-})$$

20 else if( $(\text{delk}' < -z_{\kappa+1})$  and  $(\text{delk}' > -z_{\kappa-1})$ )

$$\text{then } \text{delk}+1' = -\text{delk}' \quad (\text{mx})$$

else if( $\text{delk}' > -z_{\kappa+1}$ )

$$\text{then } \text{delk}+1' = z_{\kappa-1} \quad (\text{m+})$$

25

We will consider a few examples by using this algorithm.

Example 1

5 The following input sequence ( $z_k$ ) is to be detected:

-1, 7, 10, 10, 7, 1, -6, -6, -1, 7.

The original conditions:

a) "1" is equal to 5 (threshold),

b) del0=0;

10

step1:	$-z_{k-1}=1-5=-4$ $-z_{k+1}=1+5=6$	$\underline{0 > -4}$ $\underline{0 < 6}$	del1' = -del0' = 0	(mx)
step2:	$-z_{k-1}=-7-5=-12$ $-z_{k+1}=-7+5=-2$	$0 < -12$ $\underline{0 > -2}$	del2' = $z_{k-1}=7-5=2$	(m+)
step3:	$-z_{k-1}=-10-5=-15$ $-z_{k+1}=-10+5=-5$	$2 < -15$ $\underline{2 > -5}$	del3' = $z_{k-1}=10-5=5$	(m+)
step4:	$-z_{k-1}=-10-5=-15$ $-z_{k+1}=-10+5=-5$	$5 < -15$ $\underline{5 > -5}$	del4' = $z_{k-1}=10-5=5$	(m+)
step5:	$-z_{k-1}=-7-5=-12$ $-z_{k+1}=-7+5=-2$	$5 < -12$ $\underline{5 > -2}$	del5' = $z_{k-1}=7-5=2$	(m+)
step6:	$-z_{k-1}=-1-5=-6$ $-z_{k+1}=-1+5=4$	$\underline{2 > -6}$ $\underline{2 < 4}$	del6' = -del5' = -2	(mx)
step7:	$-z_{k-1}=6-5=1$ $-z_{k+1}=6+5=11$	$\underline{-2 < 1}$ $-2 > 11$	del7' = $z_{k+1}=-6+5=-1$	(m-)
step8:	$-z_{k-1}=6-5=1$ $-z_{k+1}=6+5=11$	$\underline{-1 < 1}$ $-1 > 11$	del8' = $z_{k+1}=-6+5=-1$	(m-)
step9:	$-z_{k-1}=1-5=-4$ $-z_{k+1}=1+5=6$	$\underline{-1 > -4}$ $\underline{-1 < 6}$	del9' = -del8' = 1	(mx)
step10:	$-z_{k-1}=-7-5=-12$ $-z_{k+1}=-7+5=-2$	$1 < -12$ $\underline{1 > -2}$	del10' = $z_{k-1}=7-1=6$	(m+)

Table 1

**Example 2**

The following input sequence ( $z_k$ ) is to be detected:

-1, 5, 8, 8, 5, 1, -4, -4, -1, 5.

5

The original conditions:

a) "1" is equal to 5 (threshold),

b)  $del_0=0$ ;

step1:	$-z_{k-1}=1-5=-4$ $-z_{k+1}=1+5=6$	$\underline{0 > -4}$ $\underline{0 < 6}$	$del_1' = -del_0' = 0$	(mx)
step2:	$-z_{k-1}=-5-5=-10$ $-z_{k+1}=-5+5=0$	$0 < -10$ $0 > 0$		no m-, m+, mx conditions, detection error.
step3:	$-z_{k-1}=-8-5=-12$ $-z_{k+1}=-8+5=-3$	$0 < -12$ $\underline{0 > -3}$	$del_3' = z_{k-1} = 8 - 5 = 3$	(m+)
step4:	$-z_{k-1}=-8-5=-12$ $-z_{k+1}=-8+5=-3$	$3 < -12$ $\underline{3 > -3}$	$del_4' = z_{k-1} = 8 - 5 = 3$	(m+)
step5:	$-z_{k-1}=-5-5=-10$ $-z_{k+1}=-5+5=0$	$3 < -10$ $\underline{3 > 0}$	$del_5' = z_{k-1} = 5 - 5 = 0$	(m+)
step6:	$-z_{k-1}=-1-5=-6$ $-z_{k+1}=-1+5=4$	$\underline{0 > -6}$ $\underline{0 < 4}$	$del_6' = -del_5' = 0$	(mx)
step7:	$-z_{k-1}=4-5=-1$ $-z_{k+1}=4+5=9$	$\underline{0 > -1}$ $\underline{0 < 9}$	$del_7' = -del_6' = 0$	(mx) wrong calculation of merge, must to be m-, detection error.
step8:	$-z_{k-1}=4-5=-1$ $-z_{k+1}=4+5=9$	$\underline{0 > -11}$ $\underline{0 < 9}$	$del_8' = -del_7' = 0$	(mx) wrong calculation of merge, must to be m-, detection error.
step9:	$-z_{k-1}=1-5=-4$ $-z_{k+1}=1+5=6$	$\underline{0 > -4}$ $\underline{0 < 6}$	$del_9' = -del_8' = 0$	(mx)

step10:	$-z_{k-1} = -5 - 5 = -10$ $-z_{k+1} = -5 + 5 = 0$	$0 < -12$ $0 > 0$	no m-, mx, m+ conditions, de- tection error.
---------	------------------------------------------------------	----------------------	----------------------------------------------------

Table 2

The analysis of the example 1 and the example 2 shows the following:

- 5 1. The algorithm calculates a merges right, if input values  $|z_k| > "1"$  (see Fig. 3).
2. The algorithm calculates a merges wrong or does the detection error (failure of detection), if input values  $|z_k| \leq "1"$  (see Fig. 4).

10

For correct calculations of a merges "1" level must be decreased (recalculation of the threshold level "1").

If the "1" level is put to 3 ("1"=3) (in example 2) the algorithm will compute correct merges.

15 Generally, we can say that the algorithm is depending on the signal amplitude. For example, the 3T pattern in the input signal (DVD or CD) has the little amplitudes and the "1" level must be always adjusted to 3T pattern amplitudes. Disadvantage of this algorithm is the dependence on the threshold level "1".

20

In order to avoid the reduction of the detection performance the "1" level must be adjusted to a little amplitude in the signal.

25 It is therefore an object of the present invention to provide an improved method and apparatus for maximum likelihood detection.

Summary of the invention

The present invention provides for an improved method and apparatus for maximum likelihood detection as set forth in the respective independent claims. Preferred embodiments of the invention are given in the dependent claims.

In essence, the invention enables to calculate the merges independently from the threshold level "1" in signal. This is especially useful for viterby detection applied to target partial response (1+D).

The present invention is particularly advantageous in that it enables to avoid the requirement of using the automatic gain control (AGC), an adaptive equalizer or the recalculation of the threshold level "1" depending on a signal amplitude.

Preferred applications of the invention include audio, video and in particular CD and DVD applications to improve the detection of data.

The general idea of the invention is to develop a method and an apparatus of the VD applied to target partial response (1+D), especially for the merge calculation unit, that calculates the merges independent of the threshold level "1" of the signal.

According to the invention a new algorithm is used, which has shown to be both, more precise than the known solution and independent of signal amplitudes.

This procedure has the advantage that no selective amplifi-

cation is necessary for Maximum Likelihood Detection.

The general idea of the invention is described as follows:

- known algorithms only work properly if (value >1), but not  
5 if the (value <1)
- according to the invention: in case that (value <1), invert conditions are required
- afterwards: invert result
  - this will lead to:  
10
    - correct results for value  $\geq 1$
    - sometimes a wrong "first" result, which has no critical influence.

#### Brief description of the drawings

15

In the following preferred embodiments of the invention will be described in greater detail by making reference to the drawings in which:

- 20 FIG 1 is a trellis diagram of different matrix,
- FIG 2 is a trellis diagram of the transitions between the states with their corresponding values of noiseless channel output,
- 25 FIG 3 is illustrative of the correct calculation of a merges, if input values  $|z_k| > "1"$  ,
- FIG 4 is illustrative of the calculation of a merges,  
30 if input values  $|z_k| \leq "1"$  ,

FIG 5 is a block diagram of a ML detector apparatus,

FIG 6 is a block diagram of a merge determining unit  
5 (MDU),

FIG 7 is a block diagram of a merge detecting unit.

#### Detailed specification

10

In the following a preferred embodiment of a method for calculation of merges independently from the threshold level "1" in signal is described in more detail:

15 The above equation (3) is modified as follows:

$$-\text{del}k = (\lambda_{k+}) - (\lambda_{k-}) \quad (12).$$

As a result from equations (4-7) and (8), the following  
20 equations are derived:

$$Q_{k+1-} = (\lambda_{k+1-}) - (\lambda_{k-}) \quad (13)$$

$$Q_{k+1+} = (\lambda_{k+1+}) - (\lambda_{k-}), \quad (14)$$

25  $Q_{k+1-} = \min (\beta_{k--}, -\text{del}k + \beta_{k+-}) \quad (15)$

$$Q_{k+1+} = \min (\beta_{k-+}, -\text{del}k + \beta_{k++}) \quad (16)$$

$$\begin{aligned} Q_{k+1+} - Q_{k+1-} &= (\lambda_{k+1+} - \lambda_{k-}) - (\lambda_{k+1-} - \lambda_{k-}) \\ &= -\text{del}k+1 \end{aligned} \quad (17).$$

30

The following relationships and equations are deducted from equations 12-17.

1. Negative merge (m-):

5 This occurs when

$$-\text{delk} > 4z_{k+4} \text{ and } -\text{delk} > 4z_{k-4}.$$

The former condition is the strongest.

10

The new difference metric (17) amounts to

$$-\text{delk}+1 = -4z_{k-4} \quad (18)$$

15 2. Cross-over (mx):

This occurs, when

$$-\text{delk} < 4z_{k-4} \text{ and } -\text{delk} > 4z_{k+4}.$$

20 The new difference metric is

$$-\text{delk}+1 = \text{delk} \quad (19)$$

3. Positive merge (m+):

25 This occurs when

$$-\text{delk} > -4z_{k-4} \text{ and } -\text{delk} > 4z_{k-4}.$$

The latter condition is the strongest.

30

The new difference metric (17) amounts to

$$-\text{delk}+1 = -4z_{k+4} \quad (20)$$

35 According to the invention it is proposed to record and to update

$$\text{delk}' = \text{delk}/4;$$

The required actions are summarized in the algorithm:

```

5
if((delk' < -zk+1) and (delk' > -zk-1))
    then delk+1' = -delk'           (mx)   (21)
else if(delk' > -zk-1)
    then delk+1' = zk+1           (m-)
10 else if(delk' < -zk+1)
    then delk+1' = zk-1           (m+)

```

The input sequences  $z_k$  from examples 1 and 2 are used to test the new algorithm.

15

The following input sequence ( $z_k$ ) is to detect:  
 -1, 7, 10, 10, 7, 1, -6, -6, -1, 7.

The original conditions:

- 20 a) "1" is equal to 5 (threshold),  
 b) del0=0;

step1:	$-z_k-1=1-5=-4 \quad 0 > -4$ $-z_k+1=1+5=6 \quad 0 < 6$	$del1' = -del0' = 0$	(mx)
step2:	$-z_k-1=-7-5=-12 \quad 0 > -12$ $-z_k+1=-7+5=-2 \quad 0 < -2$	$del2' = z_k+1=7+5=12$	(m-)
step3:	$-z_k-1=-10-5=-15 \quad 9 > -15$ $-z_k+1=-10+5=-5 \quad 9 < -5$	$del3' = z_k+1=10+5=15$	(m-)
step4:	$-z_k-1=-10-5=-15 \quad 15 > -15$ $-z_k+1=-10+5=-5 \quad 15 < -5$	$del4' = z_k+1=10+5=15$	(m-)
step5:	$-z_k-1=-7-5=-12 \quad 15 > -12$ $-z_k+1=-7+5=-2 \quad 15 < -2$	$del5' = z_k+1=7+5=12$	(m-)
step6:	$-z_k-1=-1-5=-6 \quad 12 > -6$ $-z_k+1=-1+5=4 \quad 12 < 4$	$del6' = z_k+1=1+5=6$	(m-)
step7:	$-z_k-1=6-5=1 \quad 6 > 1$ $-z_k+1=6+5=11 \quad 6 < 11$	$del7' = -del6' = -6$	(mx)
step8:	$-z_k-1=6-5=1 \quad -6 > 1$ $-z_k+1=6+5=11 \quad -6 < 11$	$del8' = z_k-1=-6-5=-11$	(m+)
step9:	$-z_k-1=1-5=-4 \quad -11 > -4$ $-z_k+1=1+5=6 \quad -11 < 6$	$del9' = z_k-1=-1-5=-6$	(m+)
step10:	$-z_k-1=-7-5=-12 \quad -6 > -12$ $-z_k+1=-7+5=-2 \quad -6 < -2$	$del10' = -del9' = 6$	(mx)

Table 3

5 Example 2

The following input sequence ( $z_k$ ) is to be detected:

-1, 5, 8, 8, 5, 1, -4, -4, -1, 5.

10 The original conditions:

- a) "1" is equal to 5 (threshold),
- b)  $del0=0$ ;

step1:	$-z_k-1=1-5=-4 \quad 0 > -4$ $-z_k+1=1+5=6 \quad 0 < 6$	$del1' = -del0' = 0$	(mx)
step2:	$-z_k-1=-5-5=-10 \quad 0 > -10$ $-z_k+1=-5+5=0 \quad 0 < 0$	$del2' = z_k+1=5+5=10$	(m-)
step3:	$-z_k-1=-8-5=-12 \quad 10 > -12$ $-z_k+1=-8+5=-3 \quad 10 < -3$	$del3' = z_k+1=8+5=13$	(m-)
step4:	$-z_k-1=-8-5=-12 \quad 13 > -12$ $-z_k+1=-8+5=-3 \quad 13 < -3$	$del4' = z_k+1=8+5=13$	(m-)
step5:	$-z_k-1=-5-5=-10 \quad 13 > -10$ $-z_k+1=-5+5=0 \quad 13 < 0$	$del5' = z_k+1=5+5=10$	(m-)
step6:	$-z_k-1=-1-5=-6 \quad 10 > -6$ $-z_k+1=-1+5=4 \quad 10 < 4$	$del6' = z_k+1=1+5=6$	(m-)
step7:	$-z_k-1=4-5=-1 \quad 6 > -1$ $-z_k+1=4+5=9 \quad 6 < 9$	$del7' = -dl6' = -6$	(mx)
step8:	$-z_k-1=4-5=-1 \quad -6 > -1$ $-z_k+1=4+5=9 \quad -6 < 9$	$del8' = -4-5=-9$	(m+)
step9:	$-z_k-1=1-5=-4 \quad -9 > -4$ $-z_k+1=1+5=6 \quad -9 < 6$	$del9' = -1-5=-6$	(m+)
step10:	$-z_k-1=-5-5=-10 \quad -6 > -10$ $-z_k+1=-5+5=0 \quad -6 < 0$	$del10' = -del9' = 6$	(mx)

Table 4

5 The Detector is divided into four blocks:

1. Slicer
2. Subtractor;
3. Merge Determining Unit;
- 10 4. Merge Detecting Unit.

1. Slicer

The slicer calculates the average value  $A_m$  of a sampled HF signal  $Y_k$ .

## 2. Subtractor

The subtractor calculates the value  $I_{Vk}$ ,  $I_{Vk}=Y_k -A_m$ .

## 5 3. Merge Determining Unit

The merge determining unit (MDU) judges the merges of the sampled data. The values  $A_m$ ,  $-A_m$ ,  $I_V$  are used to compute the merges of the sampling data.

10 The following algorithm is used to calculate the merges (under the conditions (21)).

```

BmA=-I_Vk-Am;
BpA=-I_Vk+Am;
15 If((del_k<BpA) and (dl_k>BmA)) {
    cn0=1; // mx merge
    del_k+1= -del_k;
}
else if(del_k>BmA) {
20 cn0=2; // m- merge
    del_k+1=I_V+Am;
}
else if(del_k<BpA) {
    cn0=0; // m+ merge
25 del_k+1=I_V-Am;
}
del_k=del_k+1;

```

This algorithm is implemented in the apparatus.

30

This block consists of two circuits for the performance of the negation operation  $N_1$ ,  $N_2$ ,  $ADD_1$ ,  $ADD_2$ ,  $ADD_3$ ,  $ADD_4$  adders, two multiplexers  $MUX_1$ ,  $MUX_2$ , two registers, three comparers  $COM_1$ ,  $COM_2$ ,  $COM_3$ , control block and encoder.

35

The  $ADD_1$ -adder is used to calculate the sum of  $-A_m$  and  $I_{Vk}$

values.

The ADD2-adder is used to calculate the sum of Am and I\_Vk values.

5

The ADD3-adder is used to calculate the sum  
 $BmA = (-Am \text{ and } -I\_Vk)$ .

The ADD4-adder is used to calculate the sum

10  $BpA = (Am \text{ and } -I\_Vk)$ .

The comparator COM1 checks the condition  $dl\_k > BmA$ .  
 If  $dl\_l > BmA$ , then generate the signal (m-).

15 The comparator COM2 checks the condition  $dl\_k < BpA$ .  
 If  $dl\_l < BpA$  then generate the signal (m+).

The comparator COM3 checks the condition  
 $dl\_k > BmA$  and  $dl\_k > BpA$ .

If this condition is fulfilled then the signal (mx) is being  
 20 generated.

The Control Block controls the MUX1 multiplexer and MUX2 multiplexer. The Control Block realizes the combination logic, described in Table 5:

25

m-	m+	mx	SEL M1	SEL M2	Comments
1	0	0	1	1	REG1=ADD2 OUT
0	1	0	0	1	REG1=ADD1 OUT
0	0	1	-	0	REG1=-REG2

Table 5

The encoder encodes the ML Decoder status according to next  
 30 rules:

If  $((mx=1) \text{ AND } (m- =0) \text{ AND } (m+ =0))$  then  $cn0=1$ .

If  $((mx=0) \text{ AND } (m- =1) \text{ AND } (m+ =0))$  then  $cn0=2$ .

If  $((mx=0) \text{ AND } (m- =0) \text{ AND } (m+ =1))$  then  $cn0=0$ .

The process in the MDU can be explained by the following algorithm.

```

5  /* Start */
   ADD1_OUT=-Am+I_V;
   ADD2_OUT=Am+I_V;
   ADD3_OUT=-Am+(-I_V);
   ADD4_OUT=Am+(-I_V);
10  If((REG2<ADD4_OUT) and (REG2>ADD3_OUT)) {
   REG1=-REG2;
   REG2=REG1;
   cn0=1; // mx merge
   }
15  else if(REG2>ADD3_OUT)
   {
   REG1=ADD2_OUT;
   REG2=REG1;
   cn0=2; // m- merge
20  }
   else if(REG2<ADD4_OUT)
   {
   REG1=ADD1_OUT;
   REG2=REG1;
25  cn0=0; // m+ merge
   }

```

#### 4. Merge Detecting Unit

Fig. 7 shows the merge detecting unit (MDTU). The MDTU consist of the control block and the cross-coupled shift registers (the path metrics memory).

The control block controls cross-coupled shift registers.

Depending on the cn0 value from MDU, the control block generates the following signals :

1. LD\_SP - load parallel the register SH\_SP from the register SH\_SM;
- 2.SHT\_SP - shift the contents of the register SH\_SP on the right;
- 5 3. LD\_SM - load parallel the register SH\_SM from the register SH\_SP;
4. SHT\_SM - shift the contents of the register SH\_SM on the right;
5. SHT\_PM - shift the contents of the register SH\_SM and
- 10 SH\_SP on the right at the same time;

Depending on the cn0 value from MDU, the control block generates different signal combinations.

- 15 The process in the MDU can be explained by the following algorithm.

```

if (cn0==1) {
    for(j=0; j<14; ++j) {
20         sh_sp[14-j]=sh_sp[13-j];    // shift sh_sp and sh_sm
           sh_sm[14-j]=sh_sm[13-j]; // at the same time
    }
    sh_sp[0]=1;  sh_sm[0]=0;
}
25 if(cn0==2) {
    for(j=0; j<14; ++j) {
        sh_sp[j+1]=sh_sm[j] ; // load sh_sp from sh_sm
    }
    for(j=0; j<14; ++j) {
30         sh_sm[14-j]= sh_sm[13-j]; //shift sh_sm
    }
    sh_sp[0]=0; sh_sm[0]=0;
}
if(cn0==0) {
35 for(j=0; j<14; ++j) {
    sh_sm[j+1]=sh_sp[j]; // load sh_sm from sh_sp

```

```
    }  
    for(j=0; j<14; ++j) {  
        sh_sp[14-j] sh_sp[13-j]; // shift sh_sp  
    }  
5  sh_sm[0]=1; sh_sp[0]=1;  
    }
```

The estimated bit sequence is output from sh\_sp[14] register.

10

The new algorithm calculates a correct merge, if the following conditions for input values

$$| z_k | > "1"$$

15 and

$$| z_k | < "1"$$

are fulfilled.

This means that the algorithm calculates a merge independently of a threshold level "1" and of a signal amplitude.

This solution allows avoiding the use of the automatic gain control (AGC) or the adaptive equalizer or the recalculation of the threshold level "1" depending on a signal amplitude.

25 The general idea of the invention can be applied also to audio, video CD, DVD application in an acquisition part to improve the detection of data.

30

Claims

- 5 1. A method for Viterbi detection of a partial response, the method comprising the steps of :
- if (( $\text{delk}' < -z_{k+1}$ ) and ( $\text{delk}' > -z_{k-1}$ )) then  
 $\text{delk}+1' = -\text{delk}'$  (mx) to perform a cross-over,
  - 10 - else if ( $\text{delk}' > -z_{k-1}$ ) then  $\text{delk}+1' = z_k + 1$   
(m-) to perform a negative merge,
  - else if ( $\text{delk}' < -z_{k+1}$ ) then  $\text{delk}+1' = z_k - 1$   
15 (m+) to perform a positive merge,
- where
- $\text{delk}' = \text{delk}/4,$
- 20  $\text{delk} = (\lambda_{k-}) - (\lambda_{k+}),$
- and whereby  $z_k = x_k + \gamma_k$  is input into a Viterbi decoder,  
where  $x_k = a_k + a_{k-1}$ , and  $\gamma_k$  is noise.
2. A computer programm product comprising program means for  
25 performing a method of claim 1.
3. A maximum likelihood detector for Viterbi detection of a  
partial response, the maximum likelihood detector com-  
prising means for performing a method in accordance with  
30 claim 1.
4. The maximum likelihood detector of claim 3 comprising
- a slicer for calculating an average value ( $A_m$ ) of  
35 a sampled high frequency signal ( $Y_k$ ),

- a subtractor subtracting the high frequency signal from an output signal of the slicer,
  - 5       - a merge determining unit for determining a merge based on the high frequency signal, a slicer output signal and an output signal of the subtractor, and
  - 10       - a merge detecting unit for generating a bit sequence from an output signal of the merge determining unit wherein, in case that an input value is below a threshold value, the merge detection and / or determination conditions and the final
  - 15       result are inverted.
5. The maximum likelihood detector of claim 4, the merge detecting unit comprising cross-coupled shift registers to realize a path matrix memory and a control block to
- 20       control the cross-coupled shift registers.
6. A reader for an optical disc, such as a CD or DVD player, comprising a maximum likelihood detector in accordance with anyone of the preceding claims 3, 4 or 5.

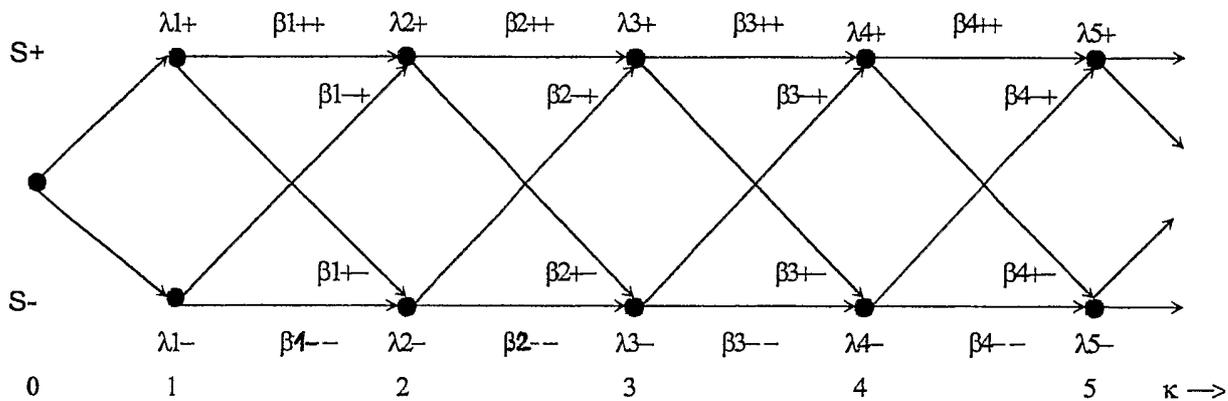


Fig. 1

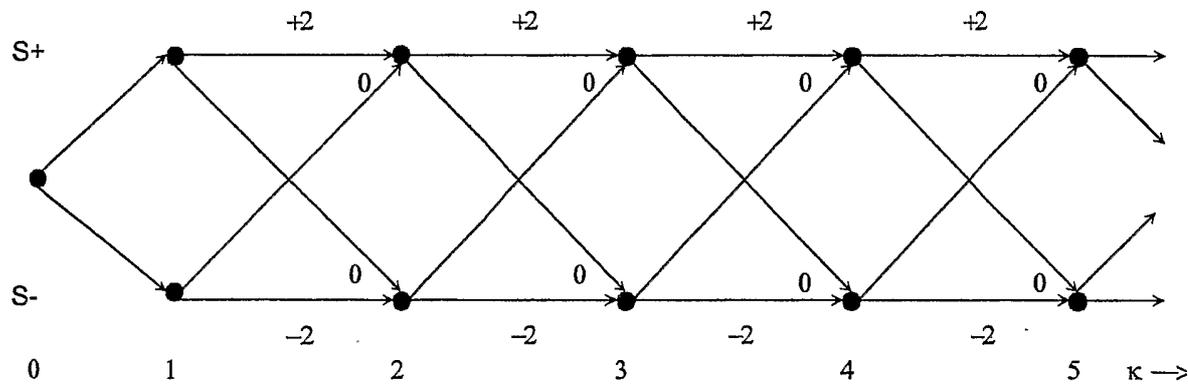


Fig. 2

2 / 5

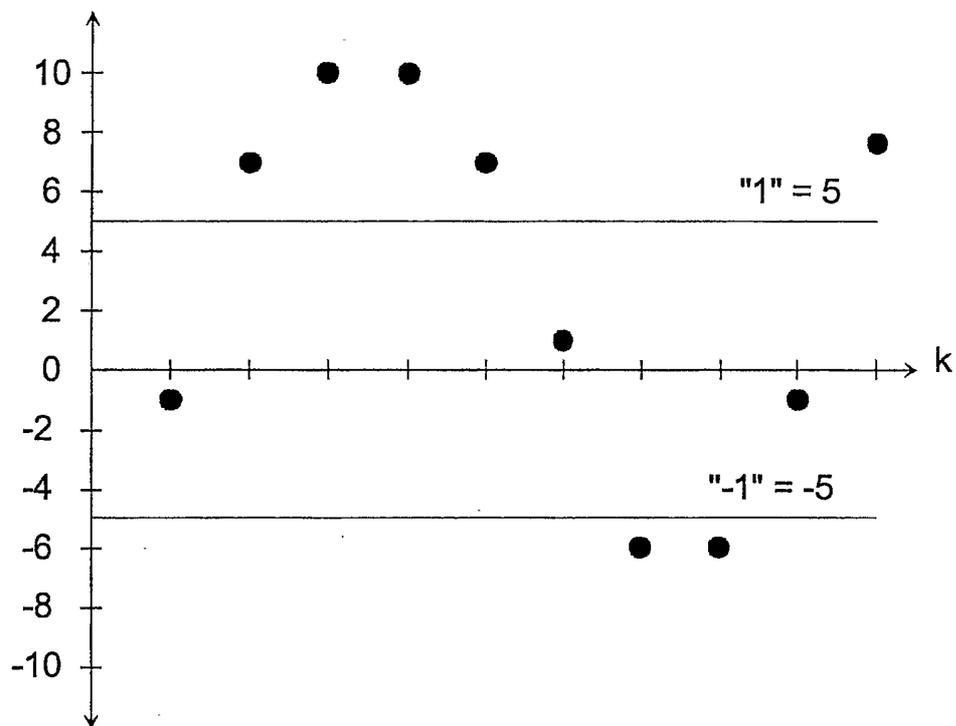


Fig. 3

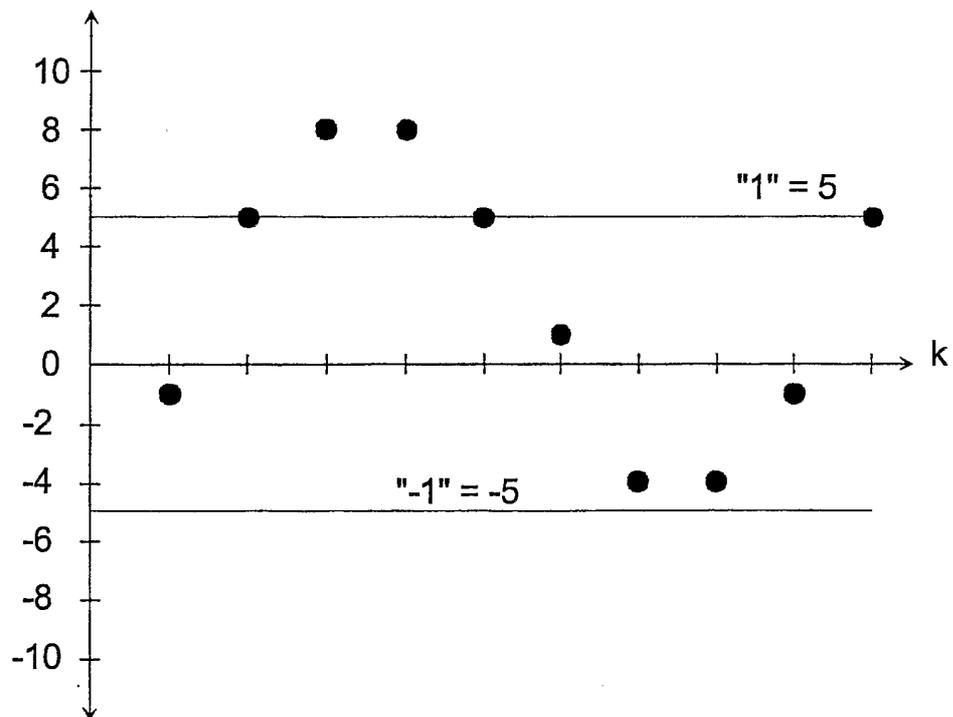


Fig. 4

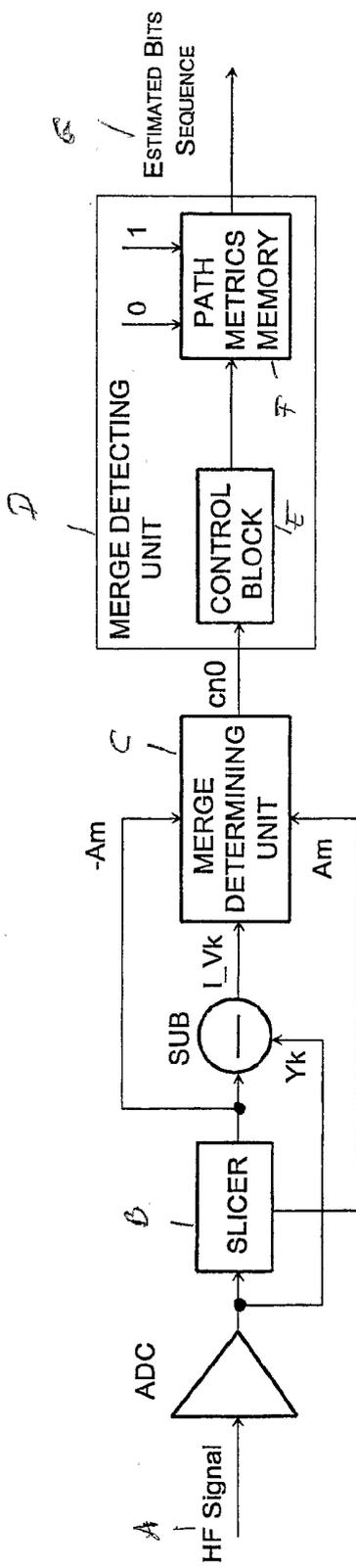


Fig. 5

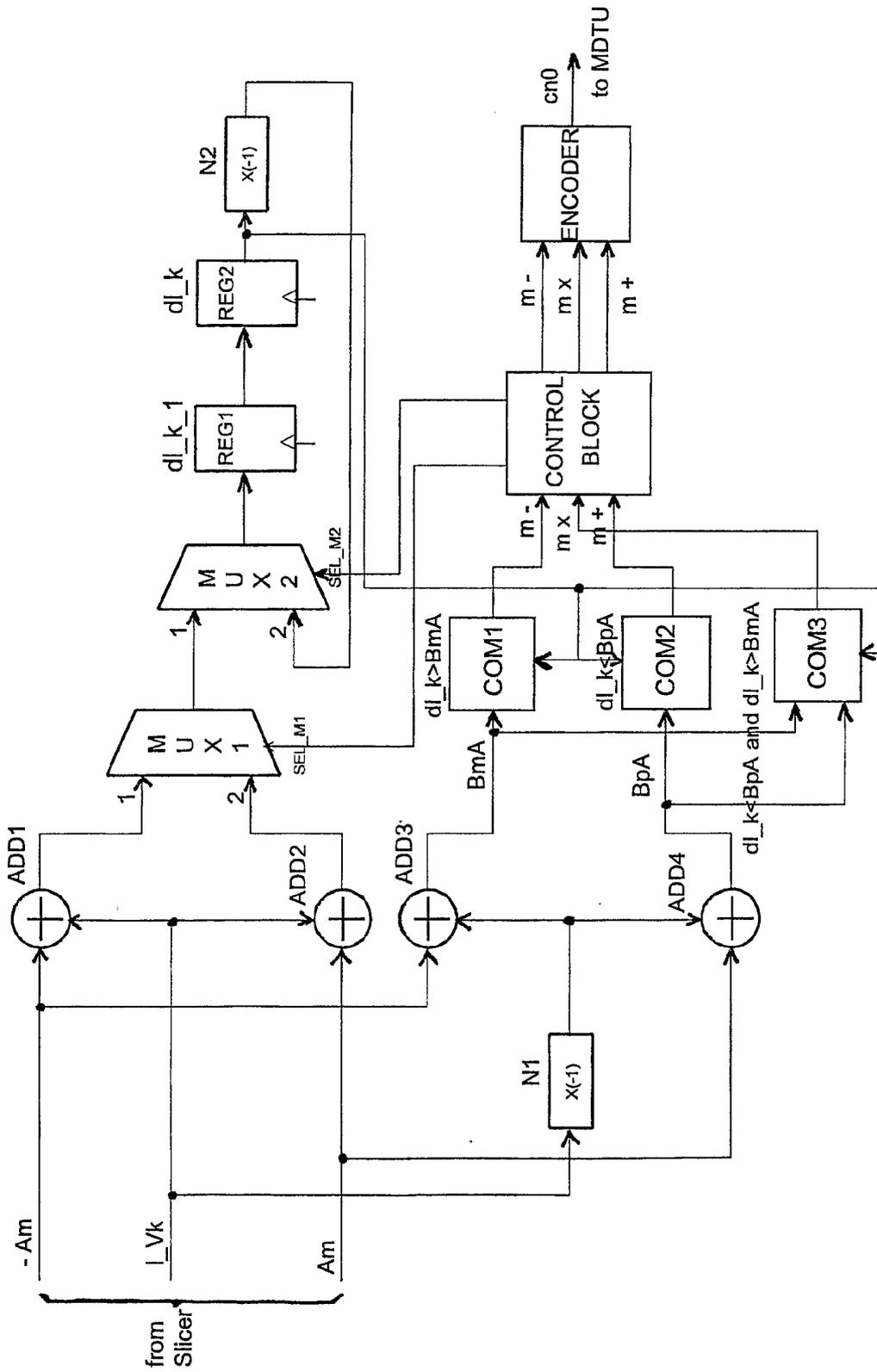


Fig. 6

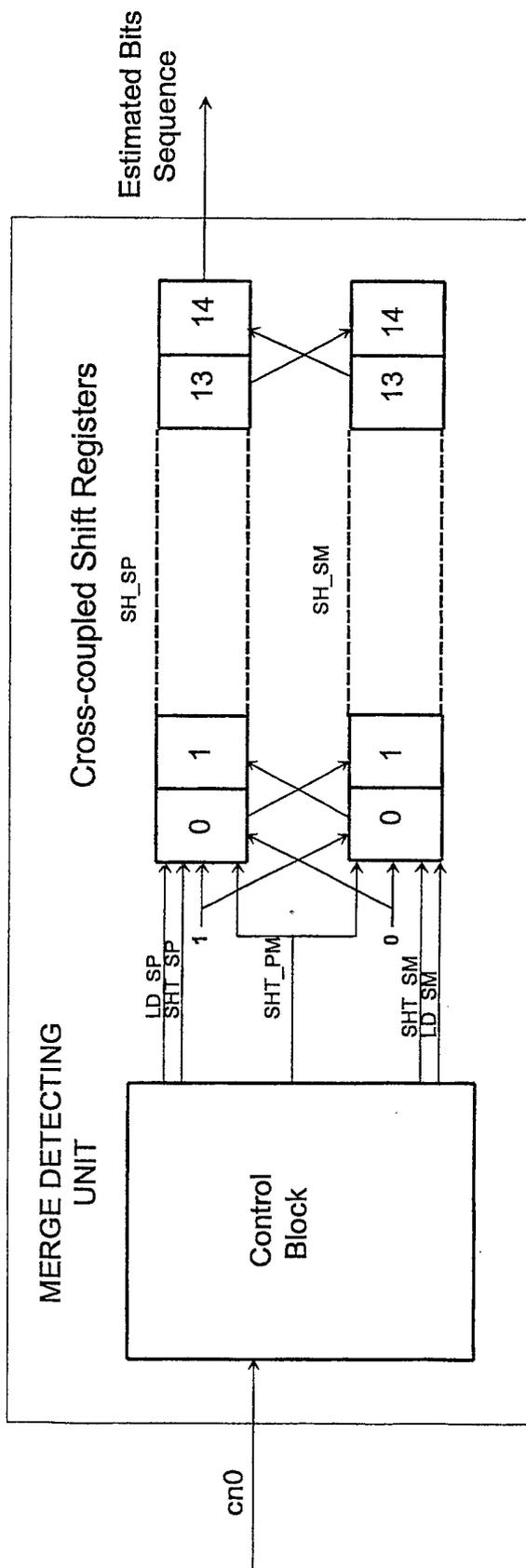


Fig. 7