MASTER

A Viterbi detector for a class IV partial response magnetic recording system

van der Vleuten, R.J.

Award date:
1989

Eindhoven University of Technology
Faculty of Electrical Engineering
Telecommunications Division (EC)

A Viterbi detector for a class IV partial response magnetic recording system

by

R.J. van der Vleuten

Thesis on a graduation project carried out at
Philips Research Laboratories, Eindhoven.

Period of work : October 1988 - August 1989

Supervisor : Prof.dr.ir. G. Brussaard
Mentors : Dr.ir. K.A. Schouhamer Immink (PRL)
Ir. A.P. Verlijsdonk (EUT)

The faculty of electrical engineering of Eindhoven University of Technology is not liable for the contents of project reports.

Abstract

The performance of a Viterbi detector for a class IV partial response magnetic recording system has been investigated. A hardware implementation of the Viterbi algorithm was made using devices from the FAST-TTL series. A 7-bit analog-to-digital converter was used to digitize the ternary partial response signal.

A 20-Mbit/s experimental digital-video magnetic recording system was used to measure the performance of the Viterbi detector, which was compared with that of a conventional (digital) threshold detector operating on the same samples. Experiments were performed with an optimally adjusted system, with a system having a tracking error, with an incompletely erased tape, and with 4-bit (instead of 7-bit) quantization. The bit error rate was measured and the statistics of burst errors and the number of correct bits between two errors were analyzed.

In all experiments, the Viterbi detector had a better performance than the conventional threshold detector. The performance gain was higher in the experiment with an incompletely erased tape and the experiment with a tracking error, which indicates that the Viterbi detector is robust against interference. Although burst errors did occur, the average burst length was small and most of the errors were single errors. Therefore, partial dropouts seem to be the cause of the majority of the errors. This hypothesis is supported by the analysis of the number of correct bits between errors and the 4-bit quantization experiment, which both indicate that the signal level varies considerably. However, further research into the characteristics of the noise is necessary to confirm the supposition.

A hardware implementation of the Viterbi detector, suitable for bit rates up to 60 Mbit/s, can be made using less than 40 ICs from the FAST-TTL series. This implies that the Viterbi detector can easily be integrated in a single chip.

Table of Contents

1.0 Introduction

Starting with the introduction of the *Compact Cassette* by Philips, in 1963, magnetic recording has become to play an important role in everyday life. Presently, in almost every home an audio cassette recorder or a video recorder can be found. With the advent of the microcomputer, digital magnetic recording entered the consumer market. In the beginning, the computer programs and data were stored on tape, but very soon the floppy disk took over the market.

The trend to switch from analog to digital systems was boosted by the success of the *Compact Disc* (CD), offering a previously unknown quality. The magnetic counterpart of the optical read-only compact disc, the *Digital Audio Tape* (DAT) recorder was introduced.

After the success of digital audio recording, the challenge is digital video recording, eventually leading to a recorder for high-definition television (HDTV). Experimental digital-video magnetic recording systems are being developed at leading laboratories around the world, including the Philips Research Laboratories (Nat Lab) in Eindhoven. The experimental system that has been developed there is based on *class IV partial response.* The essence of this response is that the channel waveform extends over three symbol intervals, thus causing an increase in the number of amplitude levels (three instead of two) at the detector. Partial response can be modeled as a well-defined form of intersymbol interference (ISI).

For detection of the ternary signal, conventionally, a threshold detector is used. Threshold detection, however, is not optimum in this case, since it does not use all the information that is available in the signal. A detector that does take advantage of the correlation between subsequently received signal levels is called a *maximum-likelihood detector*, or also *Viterbi detector* because it is commonly implemented using the well-known Viterbi algorithm. Practical verification of the performance of the detector is necessary, however, since there is no good model for the noise on the magnetic recording channel, and because it is well known that the performance of a partial response system may suffer from other perturbations like amplitude variations of the retrieved signal. The high bit rates used in digital video recording (tens of Mbit/s) form a particular difficulty for the hardware implementation of the detector.

This report describes the implementation of a Viterbi detector for a class IV partial response magnetic recording system and analyses its performance. In the following chapters, the magnetic recording channel will be introduced and partial response signaling for digital magnetic recording will be discussed. Attention will then be focused on class IV partial response and the detection thereof. Starting from the more general maximum-likelihood sequence estimation (MLSE), the Viterbi algorithm will be explained and a highly simplified hardware implementation of the algorithm will be derived. After describing the testing of the detector and introducing the experimental system, the experiments and the obtained results will be discussed. Finally, suggestions for further research will be made.

2.0 The magnetic recording channel

In this chapter, the magnetic recording channel will be introduced and a short survey of the factors that limit its performance will be given.

2.1 The principle of magnetic recording

The magnetic recording process can be modeled as a standard (radio) transmission system (Figure 1).

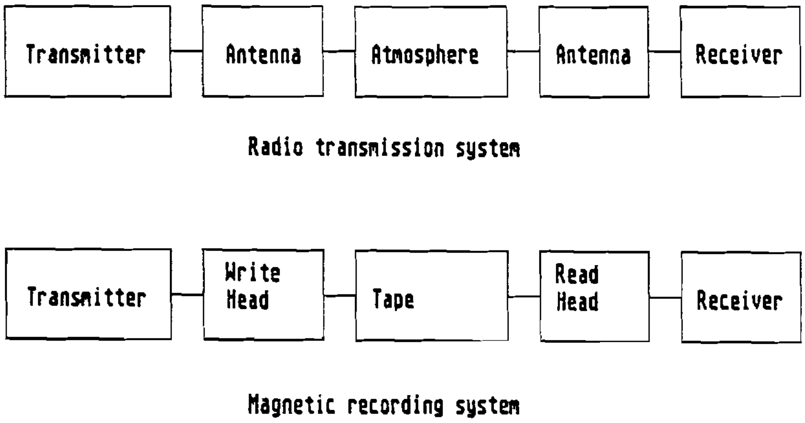


Figure 1. Radio transmission system model

Instead of an antenna, the interface between the signal of the transmitter and the medium is a coil, called the *write head*, which generates a magnetic field. The medium, a tape with a magnetic coating, moves through the field, at a constant speed, and is thus magnetized corresponding to the write-head field. During playback, the interface between medium and receiver is again a coil, called the *read head*, in which a voltage is induced by the field of the magnetized tape.

2.2 Recording limitations

In a magnetic recording system, there are several sources that may impair the retrieved signal. The performance is limited by noise, interference, and distortion

([1]:Chapter 5). In addition to these, dropouts occur. Each of the above-mentioned contributions to impairment will now briefly be discussed.

2.2.1 Noise

In principle, there are three sources of noise on the magnetic recording channel: electronics noise, read-head noise, and (recording) medium noise. The electronics noise is 1/f noise, which means it is only important at very low frequencies and can usually be neglected. The read-head noise is thermal noise, caused by the real part of the read-head impedance. It is approximately additive white Gaussian noise. Recording medium noise, finally, can be additive or multiplicative. If the magnetic particle density in the medium is constant, the noise is additive, otherwise it is multiplicative. Multiplicative noise, called *modulation noise*, is also caused by variations in the head-to-medium velocity and spacing.

In current high-density recording systems, the medium noise and the read-head noise are of comparable magnitude. In order to achieve higher areal densities, the pulse density on the track will be increased and the track width will be decreased. Since both actions reduce the retrieved signal level, and the latter also reduces the medium noise, it is expected that, in future systems, the read-head noise will become dominant.

2.2.2 Interference

Interference is due to the reception of signals other than those intended. It is mainly caused by crosstalk, incomplete erasure, and tracking errors. Crosstalk occurs in multichannel systems, i.e. when two or more heads are used in parallel. Incomplete erasure of a tape, obviously, also causes interference. Finally, interference can be caused by a tracking error that is severe enough for the read head to cover part of an adjacent track.

2.2.3 Distortion

The write process in magnetic recording is highly nonlinear, which causes distortion. More important in digital recording is the fact that, at high pulse densities, nonlinear (not additive) intersymbol interference occurs.

2.2.4 Dropouts

Dropouts are characterized by a degradation of the signal, which, depending on the severity of the dropout, could cause an error. The number of dropouts and their severity mainly depend on the quality of the tape, although a temporary increase in the head-to-medium spacing, e.g. because of a dust particle on the tape, can also cause (partial) dropouts.

2.3 A noise model

Because of the many different sources of impairment, it is not possible to give a sound, generally applicable model of the noise (here, the difference between the ideal signal value and the actual value received from the channel) on the magnetic recording channel. As a consequence, the performance of the detector in a magnetic recording system can not easily be theoretically ascertained.

3.0 Class IV partial response

After a general introduction to partial response signaling, which is well suited for the magnetic recording channel, attention will be focused on class IV partial response. Then, the structure of a class IV partial response magnetic recording system will be described.

3.1 Partial response signaling for digital magnetic recording

In a high-density digital recorder, non-return-to-zero (NRZ) pulses are used to record the information on the tape. While for low pulse densities the magnetic recording channel essentially acts as a differentiator (because of the inductive nature of the read head) producing clearly separated pulses, this approximation is no longer valid for high pulse densities, since interference between adjacent pulses occurs. As an illustration, a typical retrieved signal is displayed in Figure 2, both for a low and a high pulse density.

A linear filter, called an *equalizer*, is used to provide more reliable data detection by compensating for the nonideal channel characteristics. The types of equalizers that are used in digital magnetic recording are ([2]:§4.6.3):

- pulse slimming, which only works well for low pulse densities,
- waveform restoration, which is also called full response,
- derivative, which, in fact, is a form of partial response,
- partial response, which will be explained in the following.

The essence of partial response signaling (also called *partial response channel coding* or *correlative level coding*) is that the channel waveform extends over several symbol intervals, in a well-defined manner, thus causing an increase of the number of amplitude levels at the discrimination point. This can be modeled as a well-defined form of intersymbol interference (ISI). For detection, conventionally, a threshold detector is used. Threshold detection, however, is not optimum, in this case, since it does not use all the information that is available in the signal. A de-

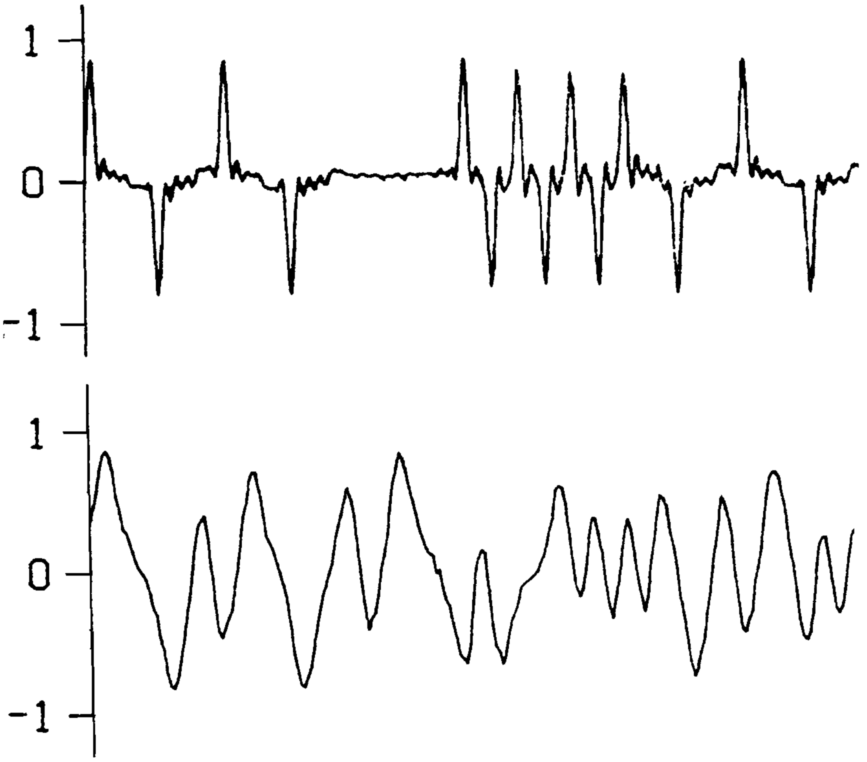


Figure 2. Illustration of intersymbol interference: at low densities (above) the pulses are clearly separated; at high densities (below) intersymbol interference occurs.

tector that does take advantage of the correlation between subsequently received signal levels is called a *maximum-likelihood* detector. Such a detector, commonly implemented using the Viterbi algorithm, has recently become practically feasible.

3.2 Performance comparison of partial responses for magnetic recording

As mentioned before in "A noise model" on page 5, there is no good model for the noise on the magnetic recording channel. Therefore, in order to be able to theoretically compare the performances of the different partial response systems, a simplified model has to be assumed. The results that are obtained using this model only give a qualitative estimation of the performance.

The recording channel is modeled by a Lorentzian step response of the form

$$h(t) = \frac{1}{1 + (2t/pw_{50})^2},$$

where the parameter pw_{50} determines the 50% pulse width. In Figure 3, the step response, $h(t)$, and the pulse response, $h(t) - h(t - T)$ (where T is the bit time) are shown. The latter has been plotted for a normalized density $S = pw_{50}/T = 2$.

The channel noise is assumed to be additive, white, and Gaussian (AWGN). Referring to "Noise" on page 4, this means that only the read-head noise is taken into account. The signal (plus noise) is fed through an equalizer in order to obtain the desired partial response system. The equalizer transfer function is found by dividing the desired spectrum of the system by the spectrum of the Lorentzian pulse. Because of the equalization, the noise at the detector input will be correlated. It is also clear that the more the partial response differs from the Lorentzian pulse response, the more the noise will be enhanced. Or, in other words, the better the partial response matches the response of the channel, the better the performance of the system will be.

In the magnetic recording literature, several partial response systems have been investigated. In [3] and [4], all partial response systems of the form $1 + D^n$ and $1 - D^n$ ($n \in \mathbb{N}$) were investigated, where D is the delay operator corresponding with a delay of one bit time and $1 + D^n$ and $1 - D^n$ are the transfer functions of the time-discrete filters that can be used to generate the respective responses. The performance was calculated for a system disturbed by AWGN, as mentioned earlier, and timing errors. For detection, a sub-optimum threshold detector was assumed. The calculations show that the best responses, in terms of signal-to-noise ratio and insensitivity to timing errors, are class I and class IV partial response [5], also denoted by $1 + D$ and $1 - D^2$, respectively [6]. According to the calculations, at current pulse densities ($S \approx 2$) and even more clearly at higher densities, the $1 + D$ and $1 - D^2$ partial responses, respectively, outperform full-response equal-

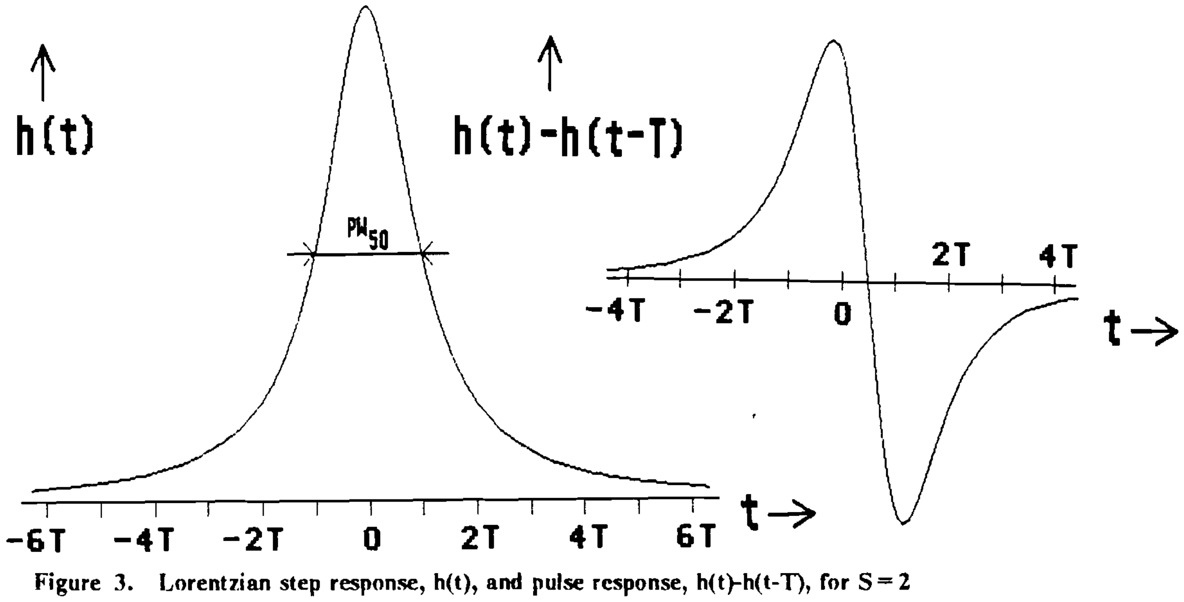


Figure 3. Lorentzian step response, h(t), and pulse response, h(t)-h(t-T), for S = 2

ization and derivative equalization (which is just $1 - D$ partial response), even without optimum (maximum-likelihood) detection. Since the performance is determined, to a large extent, by the enhancement of the noise, the conclusions will qualitatively be the same with Viterbi detection.

In another article [7], the performance of the $(1 - D)(1 + D)^n$ partial response systems ($n \in \mathbb{N}$), disturbed by AWGN only, was investigated. In this case, Viterbi detection was assumed. It was found that at current recording densities the $1 - D^2$ system clearly outperforms the $1 - D$ system, and that the performance of the $(1 - D)(1 + D)^2 = 1 + D - D^2 - D^3$ system is even slightly better (and clearly better for a higher normalized density S) than that of the $1 - D^2$ system.

As an illustration, the transfer functions of the $1 + D$, $1 - D$, $1 - D^2$, and $1 + D - D^2 - D^3$ systems are shown in Figure 4 and listed in Table 1. They are found by substituting D by $\exp(-j\omega T)$ (where $\omega = 2\pi f$), and low-pass filtering [6].

System	$\lvert H(\omega)\rvert$ **for** $\lvert\omega\rvert \leq \pi/T$
$1 + D$	$2T\cos(\frac{\omega T}{2})$
$1 - D$	$2T\sin(\frac{\omega T}{2})$
$(1 - D)(1 + D)$	$2T\sin(\omega T)$
$(1 - D)(1 + D)^2$	$4T\cos(\frac{\omega T}{2})\sin(\omega T)$

Table 1. Transfer functions of several partial response systems

In spite of its theoretical performance, the $1 + D$ response has not been applied in experimental recording systems, mainly because it does not have a spectral null at the zero frequency, whereas the magnetic recording channel has a poor low-frequency response. In the next section, attention will be focused on the $1 - D^2$ response.

3.3 Class IV partial response

The time-discrete filter that can be used to generate a class IV response is shown in Figure 5. The binary {-1,1} input signal is transformed into a ternary {-2,0,2} output signal.

The reason why class IV partial response is so appropriate for the magnetic recording channel may be found in its amplitude spectrum. There is little energy in

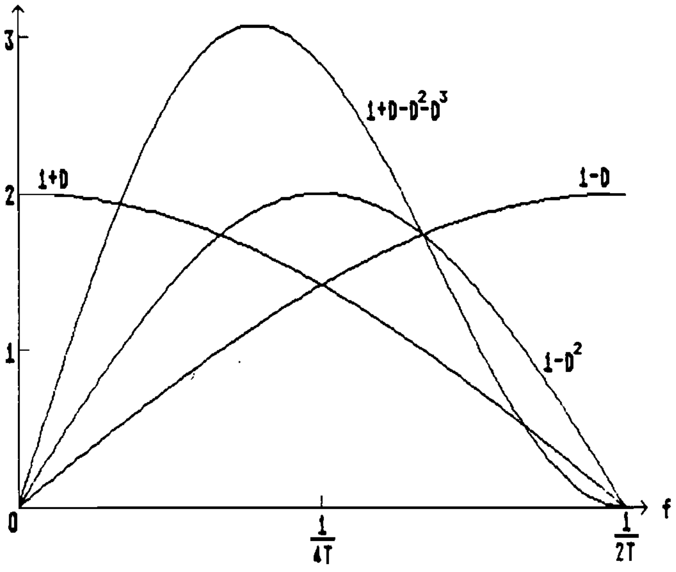


Figure 4. Transfer functions of several partial response systems

the low frequencies where the magnetic recording channel has a poor response, because of the inductive nature of the read head. And, as for any real channel, the bandwidth is limited. The high frequencies are eliminated by the low-pass filter, which cuts off at the Nyquist frequency, $1/2T$. In practice, low-pass filtering will be realized by the combination of channel and equalizer transfer functions.

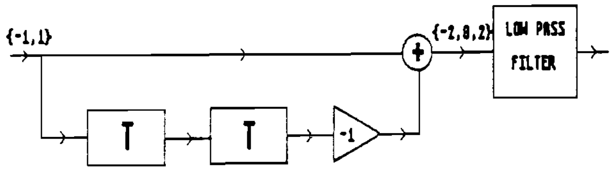


Figure 5. Class IV partial response filter

3.4 *A class IV partial response magnetic recording system*

The block diagram of a class IV partial response magnetic recording system is displayed in Figure 6, where $\{a_k\}$, $\{b_k\}$, and $\{c_k\}$ are binary signals $(a_k, b_k, c_k \in \{0,1\})$. The equalizer corrects the response of the recording channel in such a way that a class IV response is available at its output. The detector converts the ternary signal into a binary one, representing the presence of a pulse (signal level +2 or -2) as a 1 and the absence of a pulse (signal level 0) as a 0. Prior to recording, the data are precoded to prevent error propagation at the detector [8], which will now be explained.

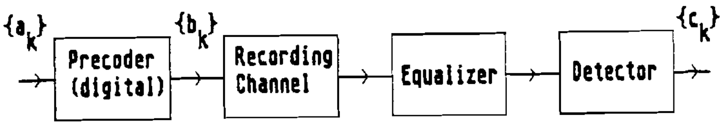


Figure 6. Block diagram of a class IV partial response system

First, consider the system without the precoder. Given an input sequence $\{b_k\}$, the output sequence $\{c_k\}$ is determined by

$$C(D) = G(D)B(D) \bmod 2,$$

where $B(D) \equiv \sum_{k=0}^{\infty} b_k D^k$, $C(D) \equiv \sum_{k=0}^{\infty} c_k D^k$, and $G(D) = 1 - D^2$.

So the transfer function of the system, without the precoder, is $(1 - D^2) \bmod 2$. Therefore, to recover the original input sequence $\{b_k\}$, it is necessary to divide the output sequence by $(1 - D^2) \bmod 2$. The circuit which can be used to this end is shown in Figure 7.

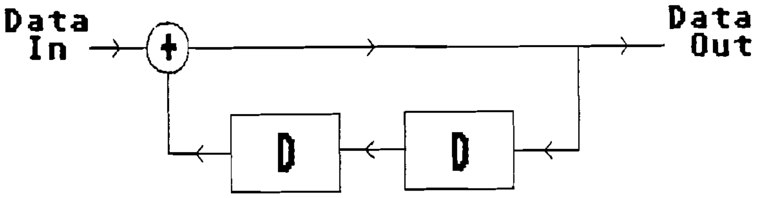


Figure 7. Class IV partial response precoder

The problem is that if a (single) error occurs on the channel, it will propagate throughout the output data pattern. In order to prevent this, the division by $1 - D^2$ is not carried out at the output of the system, but at its input: $a_k = (b_k - b_{k-2}) \bmod 2$. Dividing the input data by the transfer function of the system (modulo 2) is called *precoding.*

4.0 Detection of class IV partial response

The application of threshold detection and Viterbi detection to class IV partial response will be explained. Then, maximum-likelihood sequence estimation will be discussed, after which the Viterbi algorithm will be introduced. The bit error probability will be derived for the case of additive white Gaussian noise.

4.1 *Threshold detection*

To discriminate between the three signal levels of the partial response signal, a threshold detector can be used. Let the signal values be $-2A, 0$, and $+2A$, and the noise zero-mean, then the threshold levels will be $-A$ and $+A$.

The computation of the bit error probability is straightforward. In the absence of noise, the values $-2A, 0$, and $+2A$ are received with probabilities $\frac{1}{4}$, $\frac{1}{2}$, and $\frac{1}{4}$, respectively, so

$$P_e = \frac{1}{4} P(n > A) + \frac{1}{2} P(|n| > A) + \frac{1}{4} P(n < A) = \frac{3}{2} P(n > A),$$

assuming the distribution of the noise n is symmetrical. If the noise is Gaussian, with variance σ^2, P_e can be written as

$$P_e = \frac{3}{2} Q(\frac{A}{\sigma}),$$

where

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy.$$

4.2 *Viterbi detection*

Figure 8 gives a schematized example of an input signal $\{a_i\}$ and the corresponding output signal, $\{y_i\} = \{a_i - a_{i-2}\}$, of the class IV partial response filter (see Figure 5 on page 10). It is not difficult to see that y_i for odd i's is completely

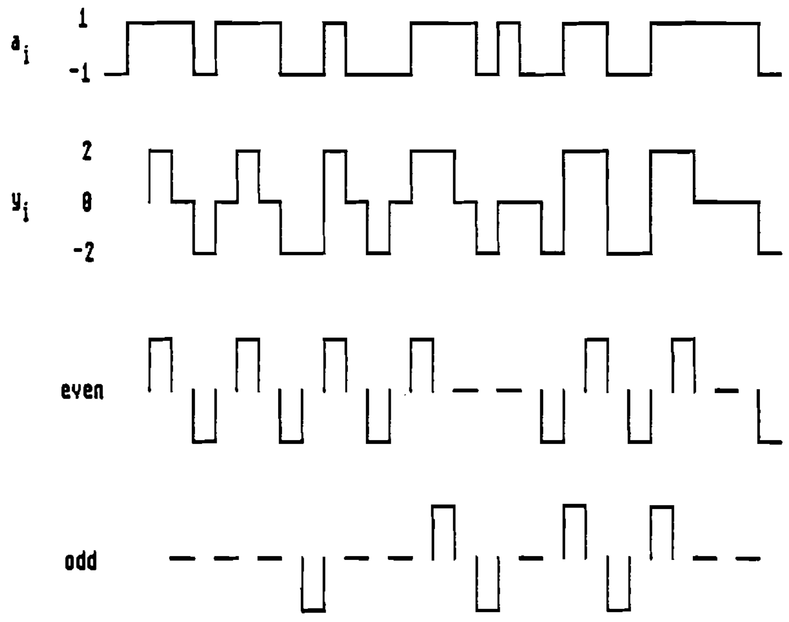


Figure 8. Schematized example of input and output signals

independent of y_i for even i's, which means that $\{y_i\}$ can be partitioned into two streams $\{y_k\} = \{a_k - a_{k-1}\}$, having, essentially, a $1 - D$ response. This, too, is shown in Figure 8. It can easily be verified that $\{y_k\}$ has a certain property: a positive-polarity pulse is always followed by a negative-polarity pulse, and vice-versa. The special property is also reflected in the *trellis diagram* (Figure 9).

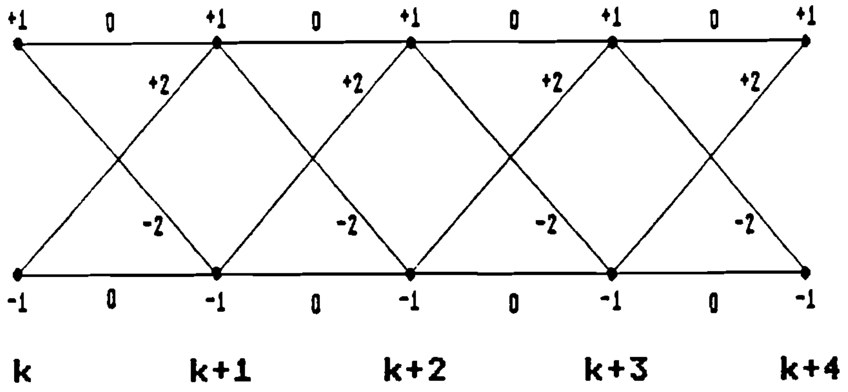


Figure 9. Trellis diagram

All possible output sequences $\{y_k\}$ can be represented as paths through the trellis, of which the nodes represent the state (the last input) of the $1 - D$ encoder and the branch values represent the output signal of the encoder. Horizontally, discrete time is indicated, starting at an arbitrary time k. As an example: the input sequence 1 1 -1 -1 would produce the output sequence 0 -2 0 2.

The detection algorithm is based on the same trellis. For each branch a cost function is defined, which is equal to the squared difference between the ideal value and the value actually received from the noisy channel. Then, the well-known Viterbi algorithm [9] is applied in order to find the cheapest path through the trellis. If the channel noise is additive, white, and Gaussian, the path found is the maximum-likelihood path [10]. The Viterbi algorithm will be explained in the next section.

4.2.1 Maximum-likelihood sequence estimation

In this section the algorithm that finds the most likely path through the trellis of Figure 9 will be derived. The path found, called the *maximum-likelihood* path, is the best estimate of the originally recorded (or transmitted) sequence.

All paths are likely, but some paths are more likely than others: the likelihood of a path depends on the statistics of the channel noise. For now, it is assumed that the noise is uncorrelated and its probability density function is known.

A part of the trellis diagram has been redrawn in Figure 10, where P_m^+ is the probability (or likelihood) of the path ending in state +1 at step m, and P_m^- is the probability of the path ending in state -1 at step m.

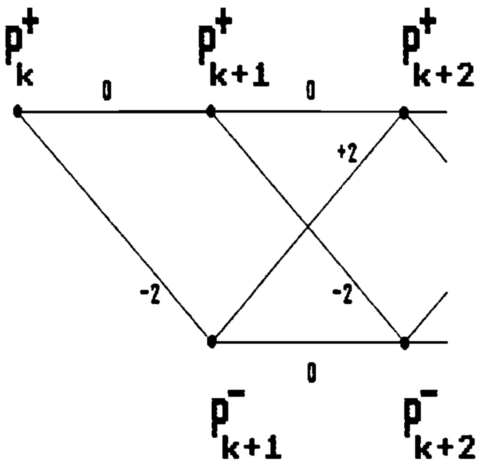


Figure 10. Trellis diagram

Assume, for simplicity, that detection starts with $y_k = a_k - a_{k-1} + n_k$ ($\{a_k\}$ is the binary input signal and $\{n_k\}$ is the channel noise) and that it is sure that the most probable path starts in state +1, i.e. define $P_k^+ = 1$. Now, P_{k+1}^+ and P_{k+1}^- must be calculated. Starting with P_{k+1}^+, given the received value y_{k+1}, the noise, n_{k+1}, is equal to $y_{k+1} - 0 = y_{k+1}$. So

$$P^+_{k+1} = P(n_{k+1} = y_{k+1})P^+_k = P(n_{k+1} = y_{k+1}).$$

Similarly,

$$P^-_{k+1} = P(n_{k+1} = y_{k+1} + 2)P^+_k = P(n_{k+1} = y_{k+1} + 2).$$

At the next step, when y_{k+2} is received, the situation is slightly more complicated, because there are two paths ending in each state. The essence of the algorithm is that only the most likely one of the two paths has to be examined further, and the other path can be discarded. The reason for this is that, because the paths end in the same state at this step, there are no more differences between their extensions at future steps. So the extensions of the most likely path of the two, at this step, will, at all future steps, be more likely than the extensions of the other path. The following formulae are found:

$$P^+_{k+2} = \max\,\{P(n_{k+2} = y_{k+2})P^+_{k+1},\ P(n_{k+2} = y_{k+2} - 2)P^-_{k+1}\}$$

$$P^-_{k+2} = \max\,\{P(n_{k+2} = y_{k+2} + 2)P^+_{k+1},\ P(n_{k+2} = y_{k+2})P^-_{k+1}\}.$$

The procedure is repeated at step $k + 3$ and further.

As an example of practical interest, the case of Gaussian noise with probability density function

$$f_X(x) = \frac{1}{\sqrt{2\pi}\,\sigma}\,e^{-x^2/\sigma^2}$$

will be considered. It is important to realize that finding the maximum of the product of two probabilities is equivalent to finding the maximum of the sum of the logarithms of these probabilities. Since $\log(f_X(x)) = Kx^2$ (so e.g. $P(n_i = y_i + 2) = K(y_i + 2)^2$), the maximum-likelihood path is the path with the least squared difference from the received values y_i. The maximum-likelihood algorithm, in this case, is called the *Viterbi algorithm*.

4.2.2 The performance for additive white Gaussian noise

In the following, a bound on the bit error probability P_e will be given, for the case of additive white Gaussian noise (AWGN).

Using an *error-state diagram*, Viterbi and Omura gave a bound on P_e for the duobinary $(1 + D)$ partial response system [11]. Since the performance of the $1 + D$ system is the same as that of the $1 - D$ system [6] (for AWGN), the same bound may be used. The error-state diagram is shown in Figure 11. Without loss of generality, it is assumed that the all-zero sequence is sent. All sequences different from the all-zero sequence, and therefore all error sequences, can be represented as paths through the graph, of which the branch values correspond with the probability that the branch is taken. I is an abstract variable representing an error (a one) in the output data sequence.

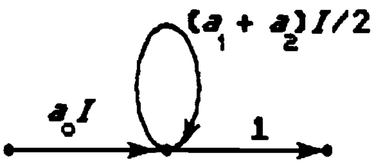


Figure 11. Error-state diagram

The generating function (the transfer function of the graph) is

$$T(a_0, a_1, a_2; I) = \frac{a_0 I}{1 - (a_1 + a_2)I/2}\,.$$

The bound on P_e is obtained by differentiating the generating function with respect to I and setting $I = 1$.

$$P_e < \left.\frac{\partial T}{\partial I}\right|_{I=1} = \frac{a_0}{[1 - (a_1 + a_2)/2]^2}\,.$$

The constants a_0, a_1, and a_2, are defined as follows (for a derivation and explanation the reader is referred to [11]:§4.9):

$$\begin{aligned} a_0 &= e^{-A^2/\sigma^2} \\ a_1 &= e^{-2A^2/\sigma^2} \\ a_2 &= 1. \end{aligned}$$

$-2A, 0$, and $+2A$ are the retrieved signal levels in the absence of noise, and σ^2 is the variance of the noise. Thus, the following result is obtained:

$$P_e < \frac{4e^{-A^2/\sigma^2}}{(1 - e^{-2A^2/\sigma^2})^2}\,.$$

Viterbi and Omura did not take into account the effect of precoding the data, but their analysis is easily modified to do so. The resulting error-state diagram is shown in Figure 12 and the bound becomes

$$P_e < \frac{4e^{-A^2/\sigma^2}}{1 - e^{-2A^2/\sigma^2}}\,.$$

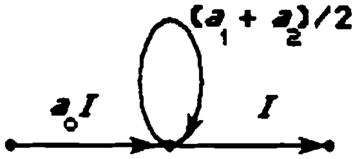


Figure 12. Modified error-state diagram: the effect of precoding the data has been included

Asymptotically, for high signal-to-noise ratios, the bound on the bit error probability can be approximated (both with or without precoding) by

$$P_e < 4\mathrm{e}^{-A^2/\sigma^2}.$$

Comparing this expression with the one obtained by Kobayashi [12],

$$P_e = 4Q(\frac{\sqrt{2}\,A}{\sigma}),$$

one sees they are different. This is explained by the fact that Viterbi and Omura, in their derivation, used the bound $Q(x) < \mathrm{e}^{-x^2/2}$, instead of the more accurate (and asymptotically exact) bound

$$Q(x) < \frac{1}{\sqrt{2\pi}\,x}\mathrm{e}^{-x^2/2}.$$

A comparison with "Threshold detection" on page 12 shows that the asymptotic performance is 3 dB better than that of a conventional threshold detector.

5.0 The implementation of the Viterbi detector

A highly simplified version of the Viterbi algorithm for the $1 - D$ detector will be derived in this chapter. The algorithm, together with the design constraints, leads to the hardware implementation.

5.1 The derivation of the simplified algorithm

Starting from the Viterbi algorithm as it was explained in "Maximum-likelihood sequence estimation" on page 14, a highly simplified version of the algorithm will be derived for the special case of the $1 - D$ detector.

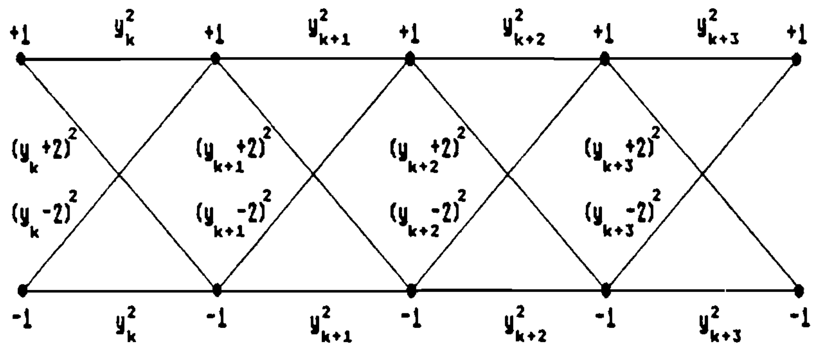


Figure 13. Trellis diagram: the branch values represent the cost function

The trellis diagram is shown again in Figure 13, the branch values representing the cost function, i.e. the squared difference between the received value and the presumed value. If the accumulated squared difference (the total cost) of the most likely path ending in state +1 or -1 at step k is denoted by S_k^+ and S_k^- , respectively, the following formulae apply:

$$S_k^+ = \min\,\{S_{k-1}^+ + y_k^2, S_{k-1}^- + (y_k - 2)^2\}$$

$$S_k^- = \min\,\{S_{k-1}^+ + (y_k + 2)^2, S_{k-1}^- + y_k^2\}.$$

In order to find the most likely (cheapest) path, only the difference between S_k^+ and S_k^- is relevant:

$$S_k^+ - S_k^- = \min\{S_{k-1}^+, S_{k-1}^- - 4y_k + 4\} - \min\{S_{k-1}^+ + 4y_k + 4, S_{k-1}^-\},$$

or, defining $\Delta S_k = S_k^+ - S_k^-$,

$$\begin{aligned}\Delta S_k &= -\Delta S_{k-1} + \min\{\Delta S_{k-1}, -4y_k + 4\} - \min\{+4y_k + 4, -\Delta S_{k-1}\}\\ &= -\Delta S_{k-1} - 8y_k + \min\{\Delta S_{k-1} + 4y_k, 4\} + \max\{\Delta S_{k-1} + 4y_k, -4\}.\end{aligned}$$

There are three different expressions for ΔS_k, depending on $\Delta S_{k-1} + 4y_k$:

$$\Delta S_k = \begin{cases} 4 - 4y_k & \Delta S_{k-1} + 4y_k > 4 \\ \Delta S_{k-1} & -4 < \Delta S_{k-1} + 4y_k < 4 \\ -4 - 4y_k & \Delta S_{k-1} + 4y_k < -4 \end{cases}$$

Following Wood and Petersen [13], ΔS_k is substituted by $4\beta - 4y_p$ and all expressions are divided by 4, resulting in the following assignment statement:

$$\beta - y_p := \begin{cases} 1 - y_k & y_k - y_p > 1 - \beta \\ \beta - y_p & -1 - \beta < y_k - y_p < 1 - \beta \\ -1 - y_k & y_k - y_p < -1 - \beta \end{cases}$$

By observing the expressions, it becomes clear that the algorithm can be implemented using two variables, $\beta \in \{-1, +1\}$ and y_p. The essential loop of the algorithm is: get the new sample y_k ; if $y_k - y_p$ outside the range $<0, -2\beta>$, then make y_p equal to y_k, and β equal to 1 if $y_k > y_p$, or -1 if $y_k < y_p$.

Interpreting the algorithm, one could say that it attempts to find the positions (on the discrete-time axis) of the pulses (signal level +2 or -2) on the channel. Whenever a candidate-pulse is found, its amplitude (y_p) and position (p) are stored. The final decision, whether a pulse did occur at that position, is postponed until more information is available, i.e. until the next candidate-pulse is received.

It is easy to see that the detector can be implemented using a random-access memory, in which the maximum-likelihood path is stored. The size of the memory should exceed the maximum distance between two pulses, which theoretically could be infinite, but the probability that such a situation occurs in reality is zero.

As a formal description of the algorithm for the $1 - D$ detector, a Pascal implementation is shown in Figure 14.

```
PROGRAM Viterbi_detector;

CONST MemSize = 28; TheEnd = False;

VAR RAM : ARRAY[0..MemSize] OF -1..1;
    p,k : 0..MemSize;
    Yp,Yk : Real;
    B, Data : -1..1;
    Update : Boolean;

BEGIN B:=1; Yp:=2; p:=0;
      FOR k:=0 TO MemSize DO RAM[k]:=-1;
      k:=0;
      REPEAT Write(Output,RAM[k]); Read(Input,Yk);
             Update:= (-B*(Yk-Yp) >= 2) OR (-B*(Yk-Yp) <= 0);
             IF Update
             THEN BEGIN Data:=-B*Sgn(Yk-Yp);
                        RAM[p]:=Data;
                        B:=Sgn(Yk-Yp); Yp:=Yk; p:=k
                  END
             ELSE RAM[k]:=-1; {Yk-Yp is inside the range <0,-2B>}
             k:= (k+1) MOD (MemSize+1)
      UNTIL TheEnd
END.
```

Figure 14. Pascal implementation of the simplified Viterbi algorithm

5.2 *The hardware design of the detector*

The hardware design has to satisfy two requirements. The first is that the detector be able to operate at a bit rate of 30 Mbit/s, fulfilling the assignment. Since the data are partitioned into two independent streams, the $1 - D$ detector is required to operate at a speed of 15 Mbit/s. The second requirement is that all circuits be built with parts that are available off the shelf.

The FAST-TTL series meets both requirements. This series is functionally compatible with the other TTL series, but it is faster. The general advantages of using TTL devices are the availability of many standard functions and the need of only a single 5V power supply.

The hardware implementation consists of three parts (see Figure 15): an analog-to-digital converter, a "logic" part, which compares the likelihood of the paths (through the trellis), and a "RAM" part, which stores the most likely path.

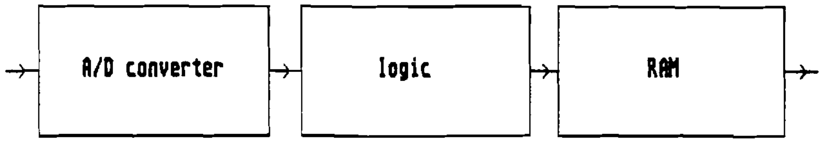


Figure 15. Block diagram of the detector

5.2.1 The design of the "logic" part

The only question left, before the "logic" part can be designed, is how many bits should be used for quantization. The quantization is presumed to be uniform, because this leads to the lowest detector complexity, in this case. Measurements by Wood and Petersen [13] showed that, when using uniform quantization over -2 to +2, 5-bit quantization would result in an acceptable performance degradation and 6-bit quantization in a negligible degradation. Since, however, the complexity of the detector, when implemented with TTL devices, is the same for 5- or 7-bit quantization, the latter was chosen. Referring to the Pascal program in Figure 14, the "logic" part, of which the block diagram is shown in Figure 16, consists of a subtractor, which computes $Y_k - Y_p$, a comparator, which evaluates the result of the subtraction, and a memory for Y_p and B.

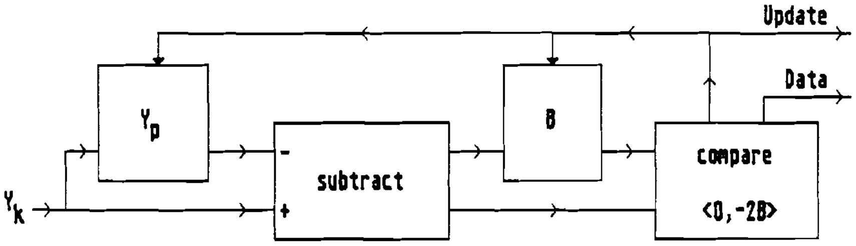


Figure 16. Block diagram of the "logic" part

Because a subtractor is not available as a standard function, an adder is used to compute $Y_k - Y_p$ as $Y_k + \bar{Y}_p + 1$, where $\bar{Y}_p + 1$ is the two's complement of Y_p .

The comparator, which computes Update and Data (defined in the Pascal program in Figure 14), is fairly simple. The values of $Y_k - Y_p$ that are of interest, namely -2, 0, and +2, coincide with changes in the carry and msb of the adder circuit. The comparator, therefore, has only three inputs (Carry, Msb, and B), and two outputs. The logical functions for the outputs are:

$$\text{Update} = \bar{B}(M + \bar{C}) + B(\bar{M} + C), \text{ and Data} = \bar{B}MC + B\bar{M}\bar{C},$$

which can be realized using two exclusive-or gates and three nand gates, as shown in the circuit diagram in "Appendix A. Circuit diagram of the 1-D detector" on page 45.

The memory for Y_p and B, finally, is made with a register that selects data from one of its two inputs, depending on a control signal (in this case Update). One of the inputs is connected to the output of the register, so the register either keeps its old value or selects a new value. The register has both true (Y_p) and complementary ($\bar{Y}_p$) outputs.

5.2.2 The design of the "RAM" part

Because of the required speed, it is not possible to use an ordinary random-access memory (RAM). The RAM is used twice during each clock period: once to read the old data and once to write the new data, as can be seen from the algorithm in Figure 14. The RAM should, therefore, operate at a clock frequency of 30 MHz.

From the algorithm, it can also be seen that the "RAM" part can be realized using a shift register in which the value of a bit, at an arbitrary position, may be changed. Because such a function is not available in one of the standard devices, it was made using programmable devices. The selected device was the PLS105A, which has 14 internal flip-flops, 16 inputs, 8 outputs, a maximum of 48 product terms, and a typical maximum clock frequency of 25 MHz.

The "RAM" part, of which the block diagram is shown in Figure 17, consists of a controller and two shift registers.

The controller maintains a pointer to the bit that could be changed in the shift registers (memory location p, in the Pascal program). The Update and Data signals, generated by the "logic" part, determine whether this bit should actually be changed. Since a 0 is shifted into the registers, the bit only has to be changed if Data = 1. While Update = 0, at each step the pointer has to be increased, so as to still point to the same bit after a shift. If Update becomes 1, then at the next step the pointer is reset to the beginning of the first shift register ($p := k$).

Since for each shift register a PLS105A is used, the length of the RAM is 2x14 = 28. So a *buffer overflow* (the pointer goes beyond the end of the last shift register) occurs if more than 28 subsequent ones or zeros occur in the input bit stream. In that case, the bit at position p can no longer be changed, which could cause an error. If the input bits are random, the probability of the occurrence of a buffer overflow is 2^{-28}.

The programmable devices were programmed using the ABEL™ software package. The advantage of using this package is that names can be used instead of pin

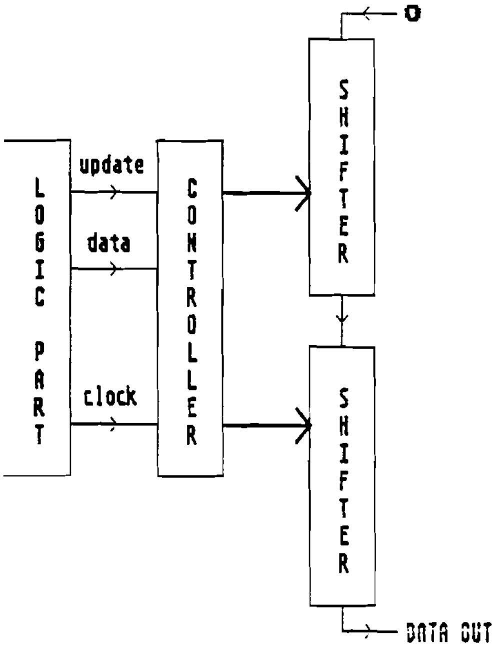


Figure 17. Block diagram of the "RAM" part

numbers; instead of having to make a table of zeros and ones, logical expressions can be specified. Furthermore, software simulation of the device is possible, so design errors may be detected and corrected.

The ABEL™ input file for the controller is given in "Appendix B. Input file for the controller" on page 48. For reasons of flexibility, the controller has been designed such that it can be connected to two or three shift registers. When an overflow occurs, a special output will give a signal as long as the overflow persists. If a bit has to be set to 1 in one of the shift registers, the controller selects a shift register and provides the address (0..13) of the bit concerned.

As can be seen from the circuit diagram of the "RAM" part ("Appendix A. Circuit diagram of the 1-D detector" on page 45), some additional logic has been inserted between the controller and the shift registers. This was necessary because not all functions could be implemented with the 48 product terms available in each shift register. So part of the product terms were generated externally.

The ABEL™ input file for the shift register is given in "Appendix C. Input file for the shift register" on page 51.

5.2.3 The analog-to-digital converter

The remaining missing link between the detector and the "real world" is an *analog-to-digital converter*, which quantizes the continuous retrieved signal. The use of a readily available board, designed for digitizing video signals, proved to be the best solution.

5.3 *Improvement of the existing design*

After the "RAM" part had already been built, a simpler and faster hardware implementation was discovered. A block diagram of the improved design is shown in Figure 18. The implementation consists of two parallel shift registers, each of which, controlled by the Update and Data signals, either shifts its own contents or copies the contents of the other one. When Update is active, the contents of one shift register are copied to the other one; the direction of the copy is determined by Data. In Figure 18, 0 is shifted into one of the registers, and Update into the other one. Instead of Update, 1 could be shifted into the other register, giving the same result (provided a buffer overflow does not occur), because only when Update is active could a bit be set to 1 in the maximum-likelihood path. If desired, a buffer overflow may be detected at the outputs of the shift registers by checking for a difference between the two stored paths.

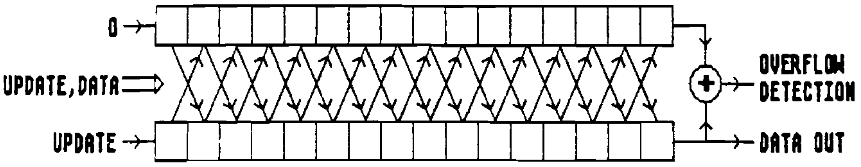


Figure 18. Block diagram of the improved design of the "RAM" part

Another possibility to improve the design would be the use of the PLUS405A, which has recently become available. This IC is the successor of the PLS105A that was used in the design of the detector, and has 16 internal flip-flops, 16 inputs, 8 outputs, a maximum of 64 product terms, and a typical maximum clock frequency of 60 MHz. Because of the larger logic array of the PLUS405A, there is no longer a need for additional logic in the "RAM" part of the detector (see "The design of the "RAM" part" on page 22). The "logic" part can be made with a single PLUS405A, because of the additional internal flip-flops of this IC.

5.4 *The maximum speed of operation*

Currently, the speed of operation of the detector is limited by the "RAM" part, of which the maximum clock frequency is a little over 20 MHz. Therefore, the maximum speed of the $1 - D^2$ detector, consisting of two interleaved $1 - D$ detectors, is about 40 Mbit/s. With the improved "RAM" part, the speed of the "logic" part will become the limiting factor. Its maximum clock frequency, obtained by adding the typical propagation-delay times in the critical loop, is about 30 MHz, corresponding with a speed of 60 Mbit/s for the complete detector.

By using the PLUS405A, as mentioned in the previous section, an even higher speed becomes practically feasible. Since this IC has a maximum clock frequency of 60 MHz, a detector operating at 120 Mbit/s becomes practically feasible.

5.5 The complexity of the hardware

The $1 - D^2$ detector, including the control logic which interleaves the two $1 - D$ detectors (but excluding the A/D converter), consists of 40 FAST-TTL devices and 6 PLS105As. By using the improved design for the "RAM" part, and with some puzzling using all gates of each device and more complex standard devices, the detector could be built with less than 40 standard devices from the FAST-TTL series. Alternatively, a high-speed detector could be built using 8 PLUS405As (and a few additional standard devices).

6.0 The testing of the detector

Each time a unit has been built, it has to be tested to verify that it functions correctly and satisfies the requirements. Subsequently, the testing of the digital part of the $1 - D$ detector, the complete $1 - D$ detector, and the $1 - D^2$ detector will be described.

6.1 Testing the digital part of the detector

First, the correct operation of the digital part of the detector was tested, using an interface to a microcomputer. Via the interface, digital 8-bit numbers could be put on the detector's bus, which is described in "Appendix D. Description of the detector's bus" on page 53. Seven bits were used to send the "samples" to the detector and the eighth bit was used for the clock signal. Through the interface, the "Data Out" and "Overflow" signals, generated by the "RAM" part, and the "Update" and "Data" signals, generated by the "logic" part, could be read back. A digital-to-analog converter was connected to the bus too, so the signal could be monitored on an oscilloscope. A Turbo-Pascal program was written to generate a random bit sequence and simulate a channel with additive white Gaussian noise (AWGN). The samples were sent to the detector, and the resulting output sequence was compared with the generated sequence. The program also simulated a threshold detector. The bit error rates as a function of the signal-to-noise ratio compared well with measurements performed by Wood and Petersen [13], and simulations performed by Kobayashi [12].

6.2 Testing the detector including the A/D converter

The next step was to include the A/D converter in the test. The block diagram of the test set-up is shown in Figure 19. In the test, a $1 - D$ partial response channel with AWGN was simulated. A digital threshold detector, which simply computes the exclusive-or function of the two most significant bits of the sampled value, was also implemented. Thus the difference between the Viterbi detector and the threshold detector could directly be monitored. The circuit diagram of the threshold detector and that of the interface between the A/D converter and the

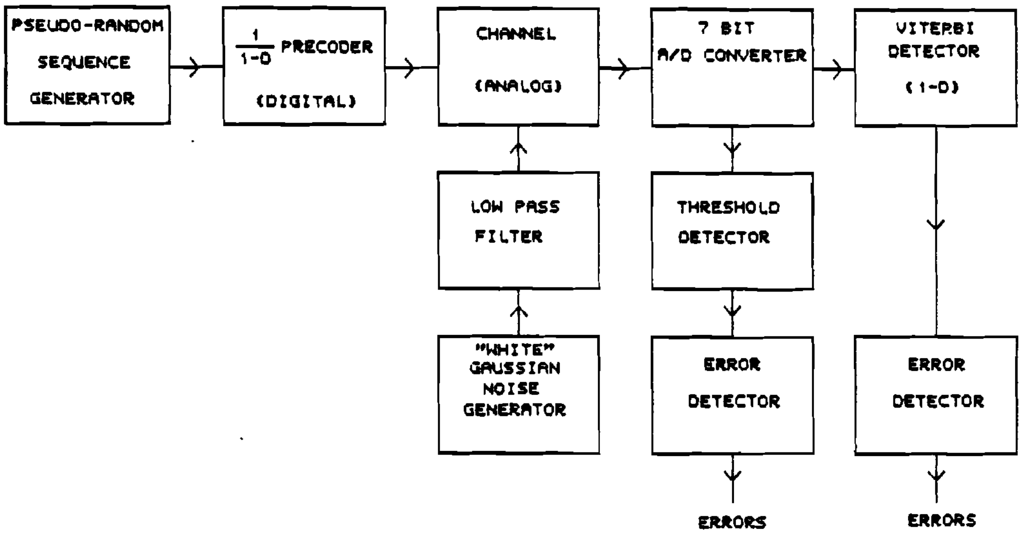


Figure 19. Block diagram of the test set-up

detector's bus are shown in "Appendix E. Circuit diagram of the 1-D interface and the threshold detector" on page 54.

The precoder is essentially the same as the one in Figure 7 on page 11, but it has only one delay element (its circuit diagram is shown in "Appendix F. Circuit diagram of the 1-D precoder and channel" on page 55). In a first design, an operational amplifier was used to simulate the $1 - D$ channel with additive noise, but at high frequencies its performance was not satisfactory. The circuit diagram of the transistor circuit that was finally designed, is shown in "Appendix F. Circuit diagram of the 1-D precoder and channel" on page 55.

Measurement of the bit error rate is done through the use of a pseudo-random sequence (PRS) generator, together with an error detector, as will be described in the next section. The advantage of this method is that synchronization of the input and output data streams is not necessary, because the error detector automatically locks to the bit stream. By dividing the error rate by the bit rate, using a timer/counter (not shown in Figure 19), the bit error probability can directly be read from the display.

6.2.1 Measurement of the bit error rate using a pseudo-random sequence

The pseudo-random sequence is generated by a linear feedback shift register (Figure 20). At power-up, or after a reset, the register is initialized by the all-zero detector, the output of which is one if all elements of the shift register are zero. In mathematical terms, the circuit divides the input signal, i.e. a one followed by all zeros, by the generator polynomial, $1 + x^3 + x^{31}$ (compare with the precoder in "A class IV partial response magnetic recording system" on page 10). The output sequence can therefore be represented by $1/(1 + x^3 + x^{31}) \bmod 2$. The length of the sequence (the number of bits before the sequence repeats itself) is $2^{31} - 1$.

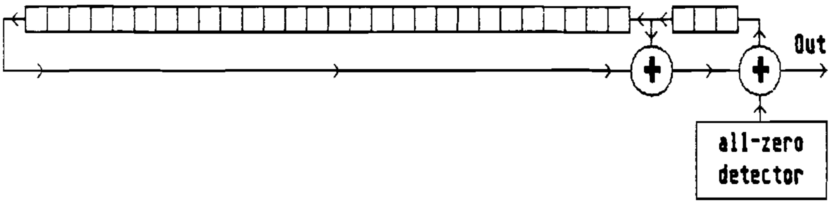


Figure 20. Pseudo-random sequence generator

The error detector, shown in Figure 21, performs the inverse function. Mathematically, it multiplies the input sequence by the generator polynomial. The problem in counting the number of errors is that a single error in the input stream of the error detector causes, in this case, three errors at its output, since the register has three taps. Therefore, a multiple-error cancellation circuit, which anticipates and compensates for the second and third error, has been included. Basically, this circuit is the same as the PRS generator: it divides the data stream by the generator polynomial. One additional circuit is necessary, however. Suppose a noise sequence on the channel produces more than 31 subsequent zeros at the input of the error detector. As can easily be seen from Figure 21, the first register of the error detector will then be completely filled with zeros. If the ensuing bits are received without errors, the shift register will send only zeros to the multiple-error cancellation circuit, which, however, does not contain all zeros, but proceeds in generating a pseudo-random sequence! A circuit has been added to prevent this; after having received 64 zeros from the shift register, it resets the multiple-error cancellation circuit to the all-zero state.

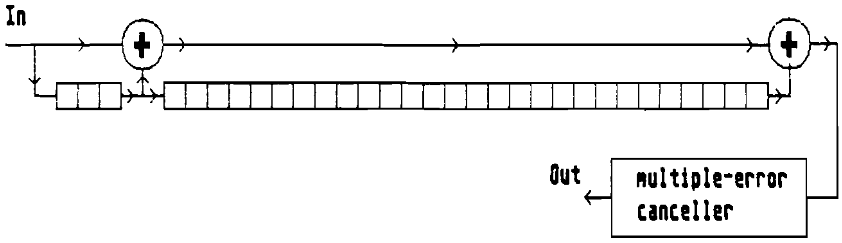


Figure 21. Error detector

At this point, a comment must be made on noise sequences: a sequence of more than 31 zeros at the input of the error detector could very well occur in a magnetic recording system, caused e.g. by a severe dropout. Without modification, even a 1000-bit dropout would cause less than 64 errors at the output of the error detector. Fortunately, the solution is simple: before the pseudo-random sequence is written on the tape, it is inverted. Of course, the retrieved bit stream is also inverted prior to error detection. Thus, a dropout causes a sequence of ones at the input of the error detector, which does not give any problems.

6.2.2 Experiments and results

First, the speed of the detector was tested. It operated correctly at 20 Mbit/s, well over the required 15 Mbit/s. A higher speed could not be tested because of the speed limitations of the PRS generator and the error detector.

The AWGN performance was measured at 5 Mbit/s, because the available low-pass filter had a cut-off frequency of 2.5 MHz. Both the signal and noise power were measured at the input of the A/D converter, using a 'true-rms' meter. The signal power is defined as the average signal power at the detection instants. The measured curves of the bit error rate as a function of the signal-to-noise ratio are shown in Figure 22. They agree well with measurements by Wood and Petersen [13], and simulations by Kobayashi [12].

6.3 Testing the class IV partial response Viterbi detector

In order to construct the $1 - D^2$ detector, first a second $1 - D$ detector had to be built (and tested). Then, the test set-up was modified, and finally the control logic to interleave the two detectors was built. The modified circuit diagrams of the precoder and channel are shown in "Appendix H. Circuit diagram of the class IV precoder and channel" on page 57, and the circuit diagram of the control logic is shown in "Appendix G. Circuit diagram of the class IV control logic" on page 56.

The $1 - D^2$ detector functioned correctly at a speed of 20 Mbit/s, the transfer rate of the experimental recording system to be introduced in "Description of the experimental system" on page 31.

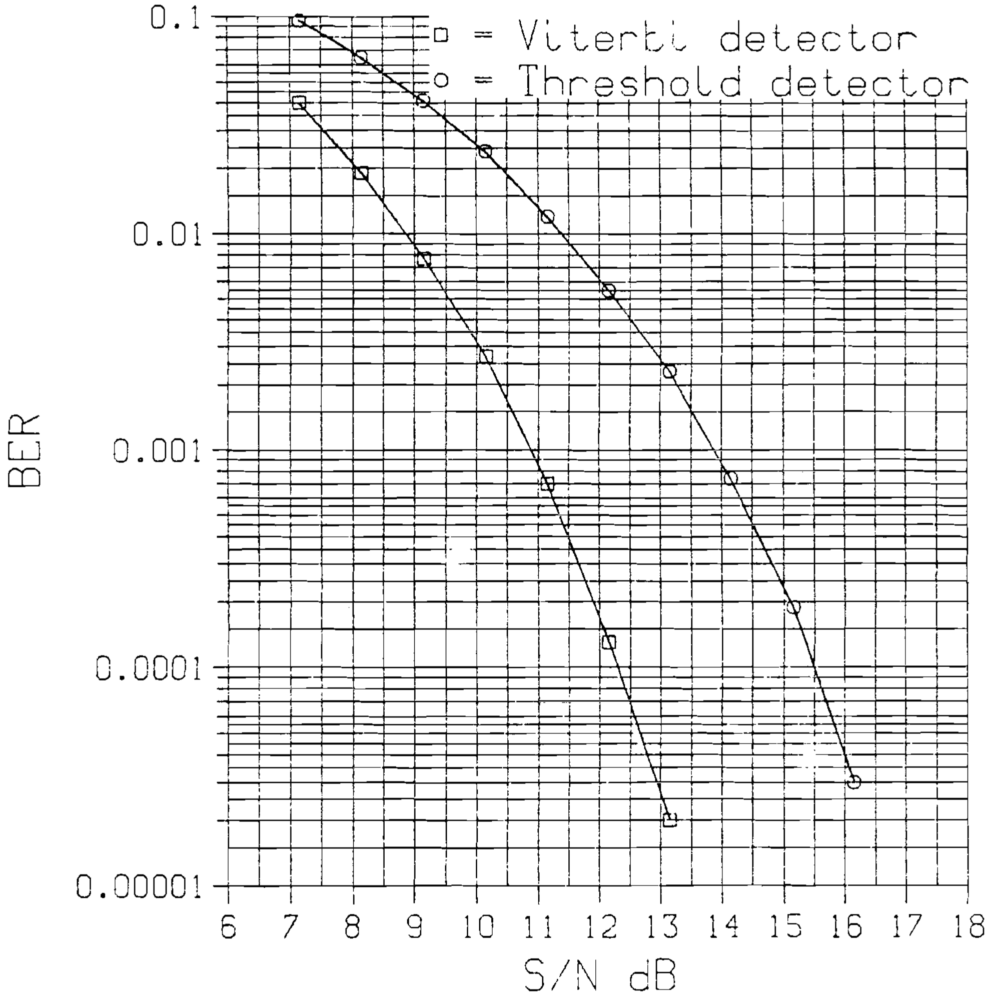


Figure 22. Measured AWGN performance

7.0 Experiments and results

This chapter begins with a description of the experimental system, followed by a description of the interface between the system and a mainframe, used to analyse the error statistics. Finally, the experiments will be discussed.

7.1 Description of the experimental system

The system used in the experiments is a *helical-scan* system ([14]:Chapter 2). It will briefly be described in the following.

The Mini Helical-scan system (MH) is an experimental digital-video recording system based on small-sized recording mechanics. Multispeed playback mode is realized by tilting the scanner axis. The recording bit rate is 20 Mbit/s. The MH system uses a small-diameter drum with a small wrap angle, which leads to small overall dimensions of the mechanics and simple tape threading. Another feature is the short track length, which enables the application of very narrow tracks without the need of a dynamic track following system.

The practical realization is as follows. On the basis of a drum diameter of 30 mm, a wrap angle of 96°, and a tape width of 8 mm, an experimental MH deck with a good accessibility of the scanner has been constructed. The tape path has been optimized for a smooth and accurate running of the tape, which together with the short track length of 24 mm leads to a track linearity within +/- 1 μm. To achieve a continuous signal, four heads are placed on the rotating drum, at 90° angles; azimuth recording is used to minimize crosstalk. The connection of the four rotating heads to the stationary part of the scanner is realized through the use of a fourfold co-axial transformer. The main parameters of the experimental MH system are listed in Table 2.

Drum diameter	30 mm
Speed of revolution	75 rps
Head speed	7 m/s
Wrap angle	96°
Effective wrap angle	90°
Tape width	8 mm
Tape	metal powder
Track width	26 μm
Track pitch	28 μm
Track length	23.5 mm
Track angle	15.3°
Tilt of scanner disc	+/- 0.25°
Recording bit rate	20 Mbit/s
Recording wavelength	0.7 μm
One field in	6 tracks
Tape speed	32 mm/s
Playing time	1 hour (8mm cassette)

Table 2. Main parameters of the experimental MH system

The block diagram of the recording channel is given in Figure 23.

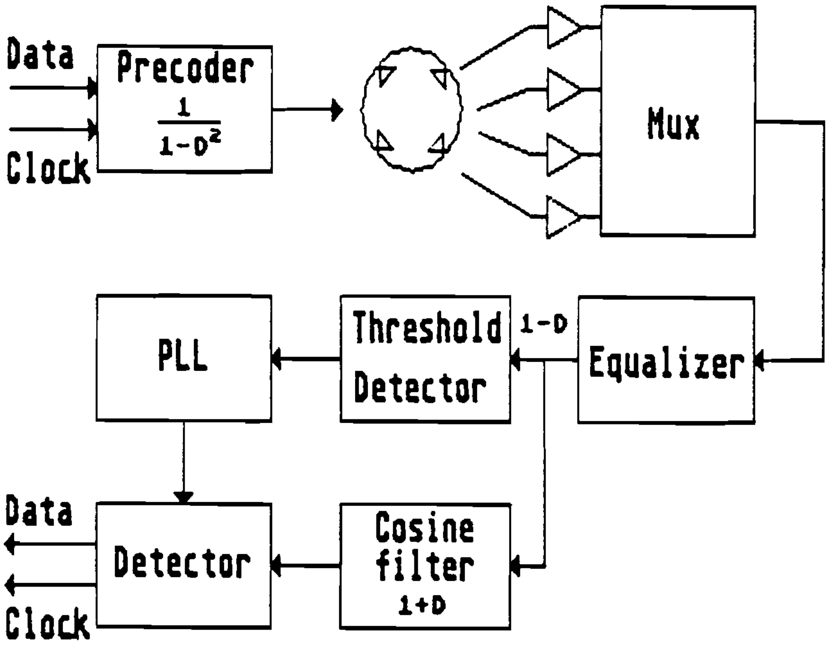


Figure 23. Block diagram of the recording channel

Prior to recording, the data are precoded (see "A class IV partial response magnetic recording system" on page 10). During playback, the outputs of the four preamplifiers are multiplexed to obtain a continuous signal. Only one equalizer is used for all four heads. As a consequence, the differences between the heads must be small for an optimum channel performance. The equalizer corrects the phase and frequency responses of the recording channel in such a way that a $1 - D$ response is available at the output. Then pulse detection is done, and the result forwarded to a PLL for clock recovery. For data detection a cosine filter $(1 + D)$ is inserted after the equalizer, to enhance the signal-to-noise ratio. Thus a class IV partial response system is obtained.

The block diagram of the detector is shown in Figure 24. Both the conventional threshold detector and the Viterbi detector operate on the same data, which allows simultaneous comparison of their performances. The $1 - D^2$ Viterbi detector consists of two independent interleaved $1 - D$ Viterbi detectors.

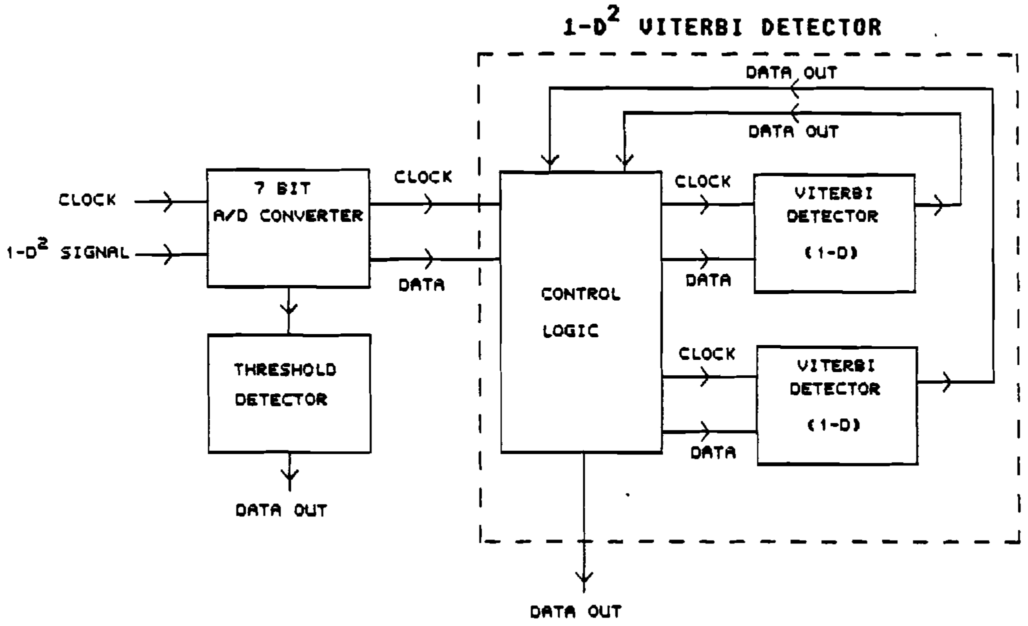


Figure 24. Block diagram of the detector

7.2 The interface between the system and a mainframe

The performance of a detector for the magnetic recording channel is measured in terms of the bit error rate (BER). Errors can be detected by using a pseudo-random sequence, as described in "Measurement of the bit error rate using a pseudo-random sequence" on page 27. Although the BER could have been monitored using a timer/counter, it was decided to build an interface between the system and a mainframe. Not only can the BER be measured more accurately in this way, but the interface also enables the analysis of the error statistics.

The interface consists of two parts: the hardware and the software. The hardware is a 32 Mbit high-speed memory. The data, coming from the error detectors for the threshold detector and the Viterbi detector, is read into the memory at 20 Mbit/s. From this, it follows that one measurement corresponds with 16 Mbit of data and a time of 0.8 s. Therefore, in order to obtain enough data, several measurements have to be done.

The software part of the interface is a data compression program, running on a microcomputer connected to the high-speed memory, which decomposes the data into blocks of zeros and blocks of ones and stores the lengths of the blocks. The data are stored in hexadecimal format, preventing conversion problems when transferring data between different computer systems. The first bit of each entry in the data file indicates whether a block contains zeros or ones. That bit is followed by the length of the block. Typically, four hexadecimal digits are used to store a block of zeros and only one digit is used to store a block of ones. These numbers are written once, at the start of the data file. Two data files can easily be concatenated by appending the second file to the first one (after removing the first line of the second file). Since most of the data are zeros (a zero represents a correct bit; a one an error), the compression reduces the length of the data file by several orders of magnitude. After compression, the data file is transferred from the microcomputer to a mainframe.

7.3 The bit error rate

Two experiments were performed. In the first experiment, the system was optimally adjusted to achieve the smallest possible BER. For the threshold detector a BER of 1.9×10^{-4} was measured. The Viterbi detector achieved a reduction of this BER by a factor of 2.9, resulting in a BER of 6.6×10^{-5}. In the second experiment, a tracking error was introduced, which increased the BER to 1.9×10^{-3} for the threshold detector and 2.0×10^{-4} for the Viterbi detector; a reduction by a factor of 9.3.

The first experiment consisted of 42 measurements; the second of 11. The number of bits in the experiments and the number of errors for the threshold detector and the Viterbi detector are shown in Table 3.

	First experiment	**Second experiment**
Number of bits	704643072	184549376
Number of errors with threshold detection	136313	345032
Number of errors with Viterbi detection	46407	37079

Table 3. Number of bits and errors measured in the experiments

7.4 Analysis of the error statistics

Although the BER does provide a measure for the performance of the detector, under several conditions, it does not reveal how or why it works. In an attempt to better understand the operation of the detector, the error statistics were investigated.

7.4.1 Burst errors

One of the characteristics examined was the occurrence of error bursts, where a burst is defined as a sequence of correctly and incorrectly detected bits, having at most N correct bits between two incorrect ones. Taking $N = 10$ and comparing the two experiments, it could be observed that in the first experiment 65%, and in the second experiment 83% of the errors made by the threshold detector were single errors. The percentage of single errors was about the same for other values of N.

Further results are shown in Table 4. It can be seen that in the second experiment the average burst length was lower than in the first experiment (The length of a burst equals the number of bits in the burst.). From the average burst length excluding single errors (bursts of length 1), it can be seen that short bursts dominate in both experiments. The fact that there are very few long bursts means that the performance is determined by partial dropouts.

	First experiment	Second experiment
Number of bits	704643072	184549376
Number of bursts	96569	309614
Average burst length	2.09	1.64
Average burst length excluding single errors	14	10

Table 4. Results of the burst error analysis

7.4.2 The number of correct bits between two errors

A second characteristic examined was the number of correct bits between two errors, which is directly related with the BER: the average number of correct bits between two incorrect ones equals 1/BER – 1. Counting the frequency of the occurrence of each number of correct bits and normalizing the area under the curve renders the probability density function (PDF). The idea behind the analysis was to find a way to graphically display the difference between burst errors and random errors: between two errors in a burst there will be few correct bits, whereas two random errors will have a large number of correct bits between them.

As an example, the case of uncorrelated noise and a constant signal-to-noise ratio will be considered. In that case, the threshold detector can be modeled by the binary symmetric channel (BSC). The errors at its output have a binomial distribution and the number of correct bits between errors is geometrically distributed. The binomial distribution gives the probability of k errors in n bits:

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k},$$

where p is the bit error probability. The geometrical distribution gives the probability of k correct bits up to the first error:

$$P(X = k) = p(1 - p)^k.$$

For large n, the binomial distribution may be approximated by the Poisson distribution,

$$P(X = k) = \mathrm{e}^{-k} \frac{\lambda^k}{k!},$$

with average $\lambda = np$. The equivalent approximation for the geometrical distribution is the exponential distribution,

$$P(X = k) = \lambda \mathrm{e}^{-\lambda k},$$

with average $1/\lambda$. Therefore, the PDF of the number of correct bits between errors, plotted on a logarithmic scale, is a straight line (see Figure 25), of which the slope depends linearly on the BER (for all practical BERs).

The PDF of the errors at the output of the Viterbi detector (also shown in Figure 25) can be approximated by two straight lines. The errors do not occur independently, but in *error events*. The number of bits between two error events is geometrically distributed, which explains one of the straight lines. As can be seen from the error-state diagram in Figure 12 on page 17, an error event causes two errors and the probability of the occurrence of the second error of an event, after the first one has occurred, is 1/2 at each step (at high signal-to-noise ratios, when $(a_1 + a_2)/2 \approx 1/2$). Therefore, the number of bits between the two errors of an event is also geometrically distributed, which explains the second line.

The PDFs that were measured are shown in Figure 26 for the first experiment, and Figure 27 for the second experiment. A single line for the threshold detector, and two lines for the Viterbi detector can roughly be identified. The curves are not straight, which means that the signal level varies considerably, since each different

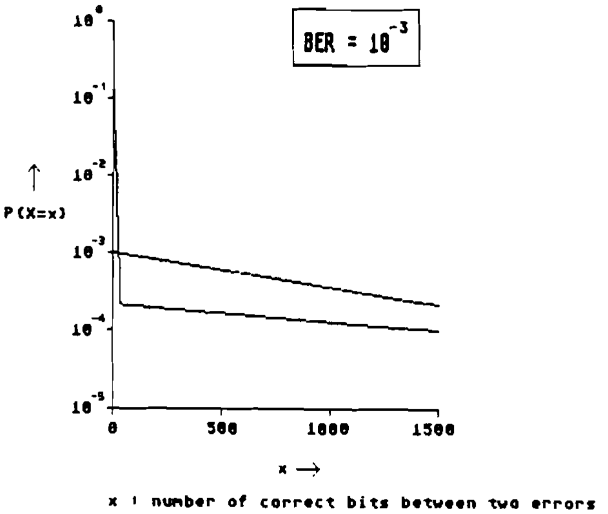


Figure 25. Example of the PDFs of the number of correct bits between two errors: the PDF for the threshold detector is a straight line; the PDF for the Viterbi detector can be approximated by two straight lines.

slope of the curve corresponds to a different signal level. It can be observed that, in the first experiment, there are two lines in the curve for the threshold detector too, which means that in this experiment error bursts, during which the BER equals 1/2, occurred relatively more often than in the second experiment, where the second line is not visible. This observation agrees with the analysis of the error bursts.

In conclusion, the analysis does show the difference between burst errors and random errors. It does not provide a quantitative explanation of the performance of the Viterbi detector, because of the lack of a suitable noise model that incorporates the variable signal-to-noise ratio.

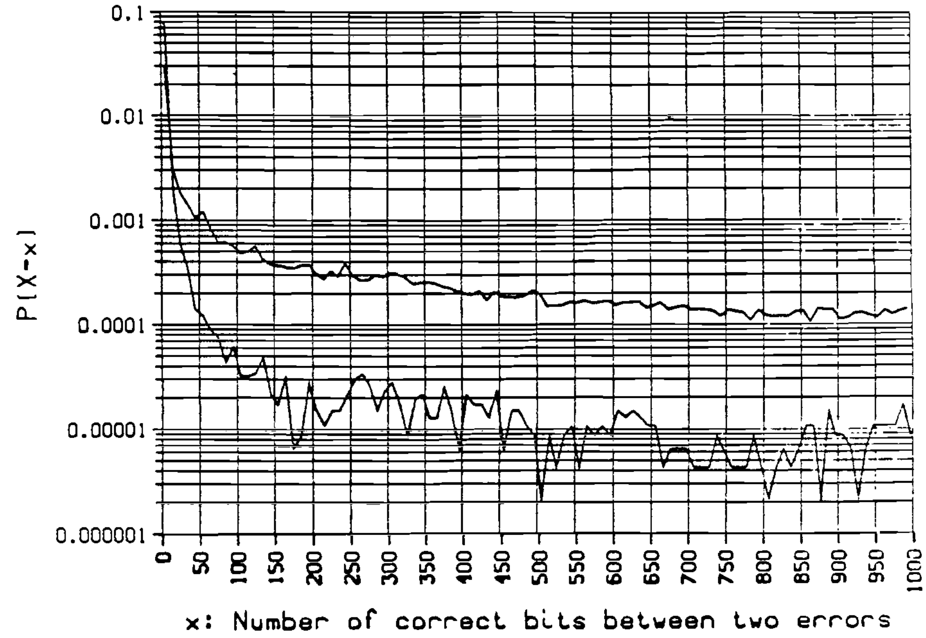


Figure 26. PDFs of the number of correct bits between two errors (first experiment): The upper curve was measured for the threshold detector, the lower one for the Viterbi detector.

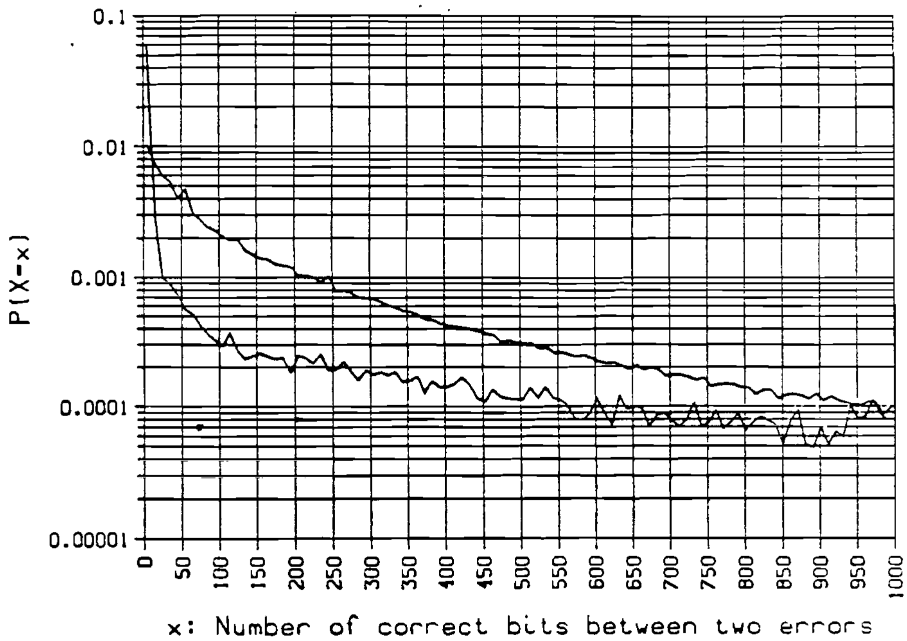


Figure 27. PDFs of the number of correct bits between two errors (second experiment): The upper curve was measured for the threshold detector, the lower one for the Viterbi detector.

7.5 Miscellaneous experiments

Two other qualitative experiments were performed. The BER was monitored on a timer/counter.

In one experiment the system was optimally adjusted, but the data had been recorded on an incompletely erased tape. The performance gain of the Viterbi detector (over the threshold detector) was higher than in the case of a well-erased tape. Since the performance was also higher in case of a tracking error, the Viterbi detector, apparently, is robust against interference (see "Interference" on page 4).

In the other experiment, the system was again optimally adjusted, but the detector was modified to use 4-bit quantization, instead of 7-bit quantization. As was explained in "The design of the ″logic″ part" on page 21, the choice for 7-bit quantization was founded on measurements by Wood and Petersen [13]. These measurements showed that the BER would double because of the 4-bit modification (in case of AWGN at the input of the detector). In the experiment, however, no performance degradation was observed, meaning that the dominant noise was not Gaussian noise from the read-head. Apparently, the additional quantization noise was negligible, compared with the amplitude variations of the retrieved signal. Other quantizations were not examined, because of the inaccuracy of monitoring the BER on the timer/counter and because quantitative results would only apply to the mini helical-scan system.

8.0 Conclusions and suggestions for further research

The Viterbi detector for a class IV partial response magnetic recording system has a better performance than the conventional threshold detector. Furthermore, a hardware implementation of the Viterbi detector, suitable for bit rates up to 60 Mbit/s, can be made using less than 40 ICs from the FAST-TTL series.

The experiments indicate that the Viterbi detector is robust against interference, since it has a higher performance gain in case of incomplete erasure or a tracking error. Its robustness against interference could well be the main reason why the Viterbi detector has a better performance than the threshold detector.

Although burst errors occur, the average burst length is small and the bursts are not the dominant source of errors: the majority of the errors are single errors. Therefore, partial dropouts seem to be the cause of most of the errors. This hypothesis is supported by the analysis of the number of correct bits between errors and the 4-bit quantization experiment, which both indicate that the signal level varies considerably.

Since there is still no good model for the noise on the magnetic recording channel, further research into its characteristics is necessary. A theoretical prediction (or a computer simulation) of the performance of a detector is impossible without a noise model. The noise characteristics could be determined by measuring the noise amplitude, defined as the difference between the ideal signal value and the actual value received from the channel. The measurement could e.g. be done directly after the A/D converter, using special-purpose hardware.

Once a good model of the noise is available (or a good characterization, obtained by measurements), computer simulation of the performance of a detector becomes possible. Also, a better detector, that takes into account the information that is available on the noise, can be developed (The Viterbi detector is, in this case, not the maximum-likelihood detector.). Furthermore, the information about the noise variance can be used to determine the number of bits necessary for quantization.

Whether or not the investigation of the noise characteristics confirms that partial dropouts are the main cause of errors, the performance of the detector during partial dropouts may be improved by making it adaptive to the amplitude level of the retrieved signal. Another advantage of an adaptive detector, compared with the existing design, is that the amplitude level no longer has to be adjusted manually, allowing the detector to be applied in a consumer-type video recorder.

There are two possibilities to realize an adaptive detector, the first of which is to use an adaptive amplifier to ensure that the average signal level at the input of the A/D converter is automatically adjusted. The design of such a circuit is well known. The second possibility, which might be easier to implement and will achieve the same performance as the first one (presuming the additional quantization noise is negligible), is a modification of the detector: an additional circuit computes the average level of the (rectified) partial response signal and accordingly adjusts the comparator level (see "The design of the "logic" part" on page 21). The important advantage of the second solution is that the additional circuit is completely digital, so it can easily be integrated with the existing design.

All previously mentioned improvements are applicable to the class IV partial response Viterbi detector. In "Performance comparison of partial responses for magnetic recording" on page 7, it was mentioned that the better the partial response matches the response of the channel, the better the performance of the system will be. A logical step, therefore, would be to search for systems that are appropriate to the magnetic recording channel. The search could be based on the measured characteristics of the channel, or on a channel model.

The best candidate for further research is the $(1 - D)(1 + D)^2 = 1 + D - D^2 - D^3$ partial response system. As was mentioned in "Performance comparison of partial responses for magnetic recording" on page 7, calculations of the performance of this system, based on a Lorentzian step response with AWGN, indicate that its performance should be slightly better than that of the $1 - D^2$ system. In an experimental system, however, the better performance of the $1 + D - D^2 - D^3$ system can not be guaranteed. The system might be more sensitive to amplitude variations of the retrieved signal, because it has five amplitude levels at the discrimination point. On the other hand, the $1 + D - D^2 - D^3$ system has less power at the high frequencies of the (minimum bandwidth) spectrum (see Figure 4 on page 10), so it might be less sensitive to partial dropouts (which mainly cause a loss of these high frequencies).

A point of practical interest is that the sensitivity of the $1 + D - D^2 - D^3$ system to timing errors is virtually the same as that of the $1 - D^2$ system [6]. Another practical observation is the fact that a $1 - D^2$ system can be upgraded to a $1 + D - D^2 - D^3$ system by simply changing the detector circuit (assuming the additional quantization noise is negligible): the new detector circuit incorporates a Viterbi detector for the $1 + D - D^2 - D^3$ response, and an additional digital

$1 + D$ filter. As for the class IV detector, for this system, too, a highly simplified implementation of the Viterbi algorithm may be derived.

Before proceeding with a final suggestion, it must be remarked that the approach that is normally taken to improve the performance of a channel — channel coding — is not effective, in this case. An analysis by Immink [15], on the basis of a Lorentzian step response with AWGN, showed that the codes designed for the $1 - D$ or $1 - D^2$ partial response channels do not promise a performance gain in practical recording systems. The main reason for this is that the $1 - D$ and $1 - D^2$ responses differ too much from the response of the channel.

As an alternative, however, the $1 + D$ system should be mentioned. This response has not yet been used in experimental recording systems, mainly because it does not have a spectral null at the zero frequency (see "Performance comparison of partial responses for magnetic recording" on page 7). It would be interesting to investigate the performance of the $1 + D$ system combined with a dc-free code. The higher performance of the system might compensate for the additional bandwidth needed.

9.0 Acknowledgements

I would like to thank Mr. Brussaard, Mr. Schouhamer Immink, and Mr. Verlijsdonk for letting me carry out my final project at Philips Research Laboratories.

Mr. Schouhamer Immink deserves special gratitude for encouraging me to write and co-authoring the paper I presented at the Benelux information theory symposium [16]. I could not have done that without his support.

Then, I would like to thank Mr. Rijckaert for his description of the mini helical-scan system and his invaluable support of the experiments.

Last but certainly not least, I would like to thank the members of the Magnetic Recording group who accepted me as their guest and helped me in many ways.

References

[1] C.D. Mee and E.D. Daniel, editors, *Magnetic Recording, Volume I: Technology,* McGraw-Hill Book Company, New York, 1987.

[2] C.D. Mee and E.D. Daniel, editors, *Magnetic Recording, Volume II: Computer Data Storage,* McGraw-Hill Book Company, New York, 1988.

[3] H. Osawa, Y. Okamoto, and S. Tazaki, 'Performance Comparison of Partial-Response Systems in Digital VTR', Electron. & Commun. Jpn. Part 1, vol. 68, pp. 75-83, July 1985.

[4] H. Osawa, S. Tazaki, and S. Ando, 'Performance Analysis of Partial-Response Systems for Non-Return-to-Zero Recording', IEEE Trans. Magn., vol. MAG-22, pp. 253-258, July 1986.

[5] E.R. Kretzmer, 'Generalization of a Technique for Binary Data Communication', IEEE Trans. Commun. Technol., vol. COM-14, pp. 67-68, Feb. 1966.

[6] P. Kabal and S. Pasupathy, 'Partial-Response Signaling', IEEE Trans. Commun., vol. COM-23, pp. 921-934, Sept. 1975.

[7] H.K. Thapar and A.M. Patel, 'A Class of Partial Response Systems for Increasing Storage Density in Magnetic Recording', IEEE Trans. Magn., vol. MAG-23, pp. 3666-3669, Sept. 1987.

[8] H. Kobayashi and D.T. Tang, 'Application of Partial Response Channel Coding to Magnetic Recording Systems', IBM J. Res. Develop., vol. 14, pp. 368-375, July 1970.

[9] G.D. Forney, Jr., 'The Viterbi Algorithm', Proc. IEEE, vol. 61, pp. 268-278, Mar. 1973.

[10] G.D. Forney, 'Maximum-likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference', IEEE Trans. Inform. Theory, vol. IT-18, pp. 363-378, May 1972.

[11] A.J. Viterbi and J.K. Omura, *Principles of Digital Communication and Coding,* McGraw-Hill, New-York, 1979.

[12] H. Kobayashi, 'Application of Probabilistic Decoding to Digital Magnetic Recording Systems', IBM J. Res. Develop., vol. 15, pp. 64-74, Jan. 1971.

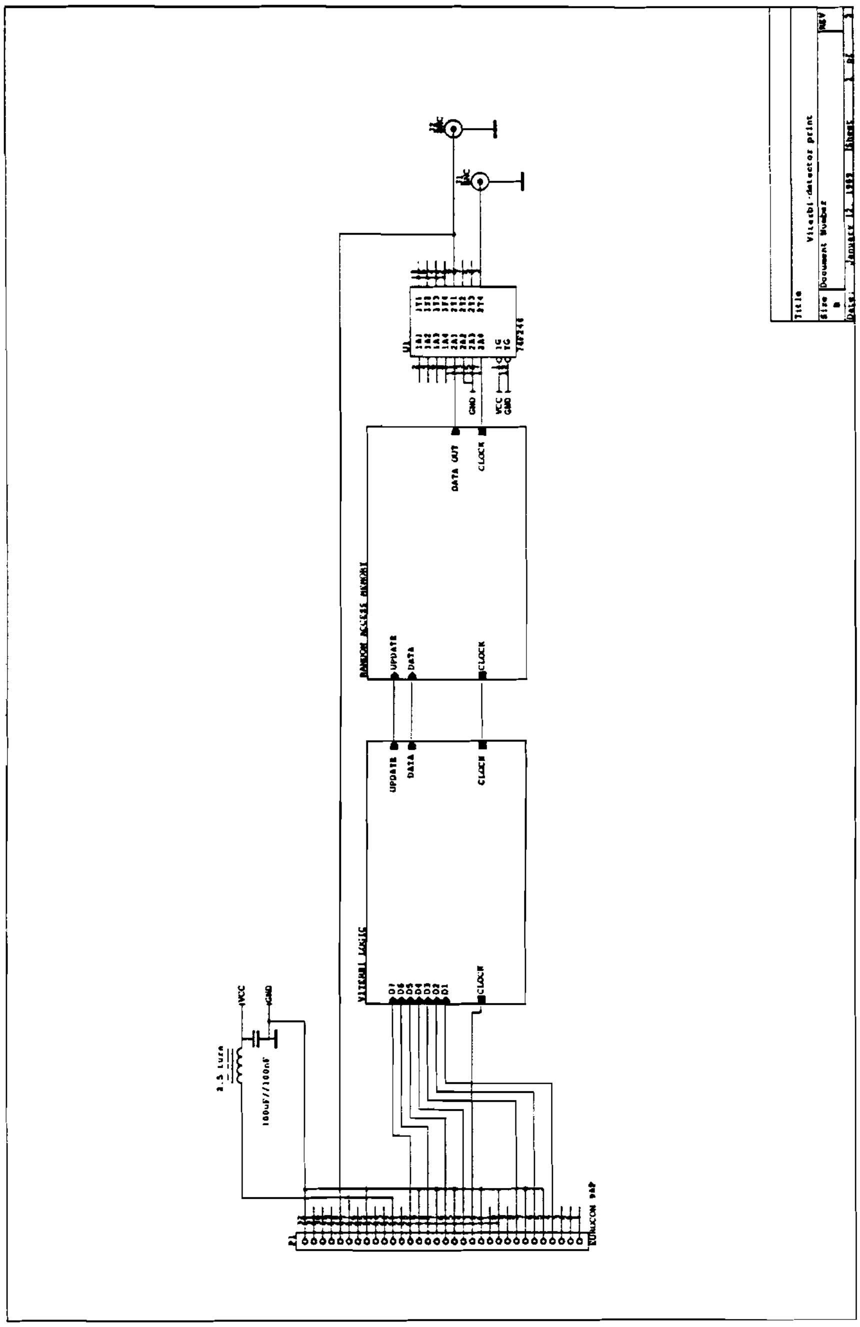
[13] R.W. Wood and D.A. Petersen, 'Viterbi Detection of Class IV Partial Response on a Magnetic Recording Channel', IEEE Trans. Commun., vol. COM-34, pp. 454-461, May 1986.

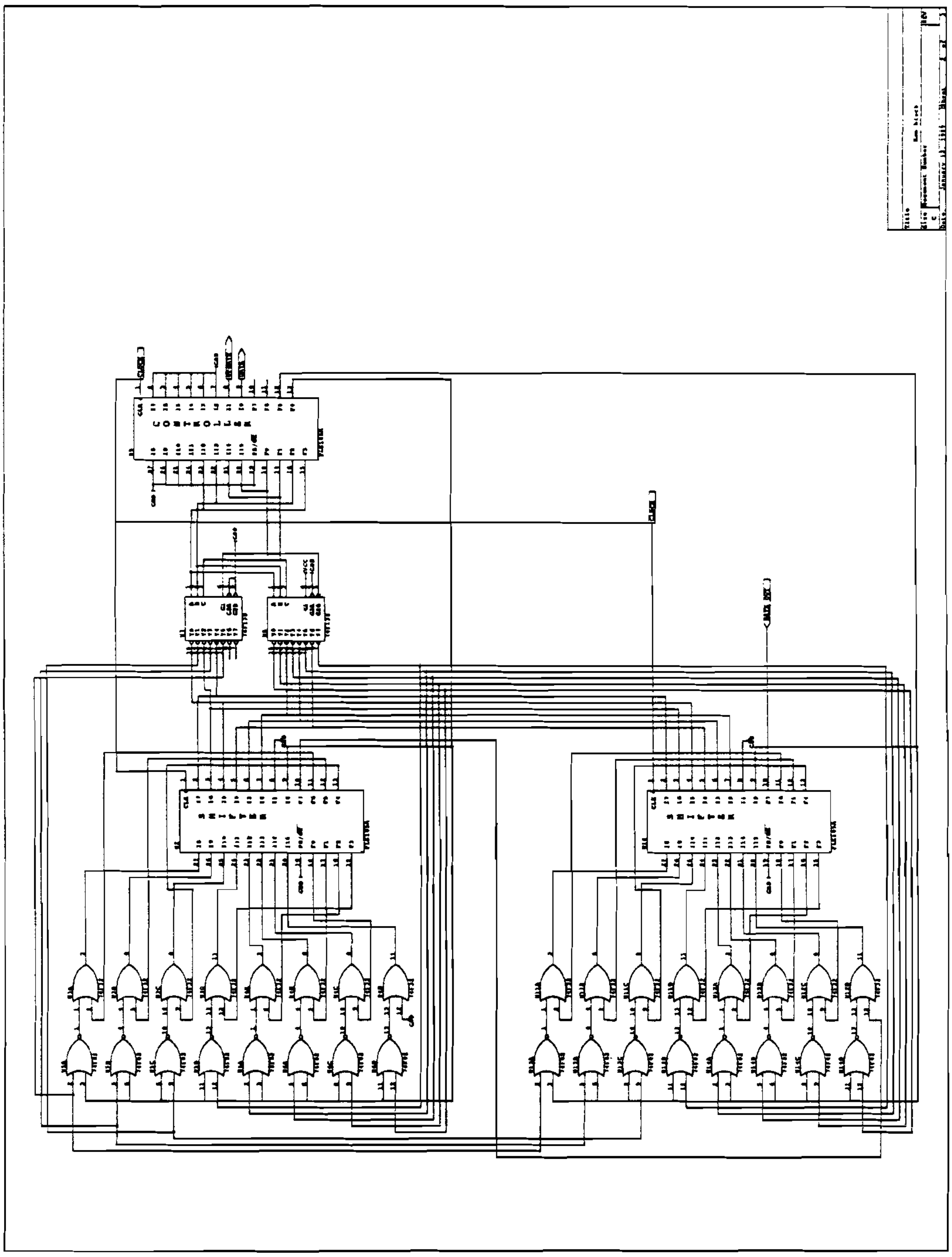
[14] C.D. Mee and E.D. Daniel, editors, *Magnetic Recording, Volume III: Video, Audio, and Instrumentation Recording,* McGraw-Hill Book Company, New York, 1988.

[15] K.A.S. Immink, 'Coding Techniques for the Noisy Magnetic Recording Channel: A State-of-the-Art Report', IEEE Trans. Commun., vol. COM-37, pp. 413-419, May 1989.

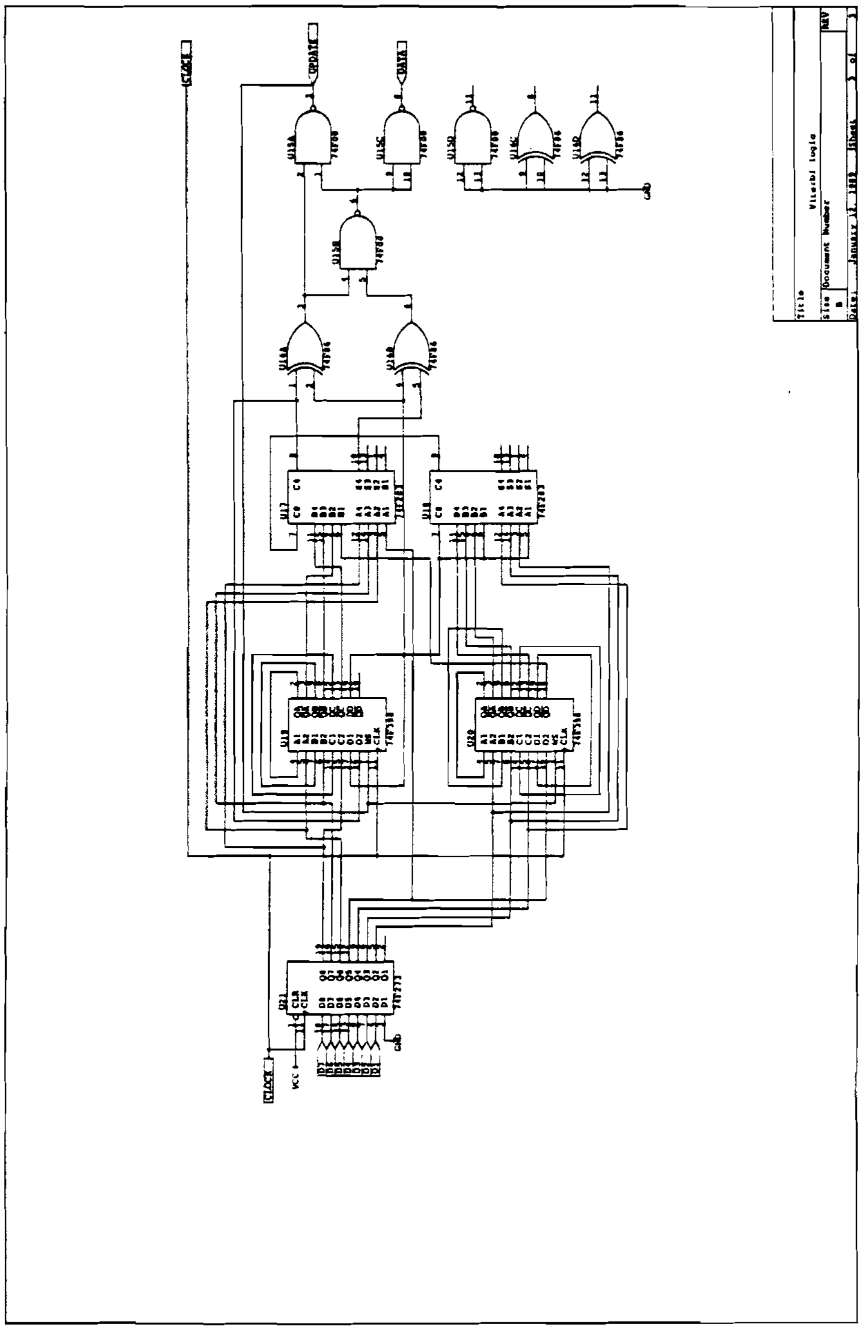
[16] R.J. van der Vleuten and K.A. Schouhamer Immink, 'A Maximum-likelihood Detector for a Class IV Partial Response Magnetic Recording System', Proc. of 10th Symp. on Inform. Theory in the Benelux, Houthalen, Belgium, pp. 117-123, May 25-26, 1989.

Appendix A. Circuit diagram of the 1-D detector





Appendix A. Circuit diagram of the 1-D detector



Appendix A. Circuit diagram of the 1-D detector

Appendix B. Input file for the controller

```
MODULE CONTROLLER;

FLAG '-R1';

TITLE 'RAM CONTROLLER BY RENE VAN DER VLEUTEN *** 15-11-1988';

CTRL DEVICE 'F105';

CLK     PIN     1;
PR      PIN     19;
C = .C.;

"POINTER"
Q3,Q2,Q1,Q0     PIN     18,17,16,15;

"FEEDBACK"
IQ3,IQ2,IQ1,IQ0 PIN     20,21,22,23;

"2 OR 3 RAMS CONNECTED"
LONG    PIN     24;

"SELECT CHIP"
SEL2,SEL1,SEL0  PIN     11,12,13;

"INPUTS"
UPDATE,DATA     PIN     8,9;

"DELAYED UPDATE"
DELUP   NODE    37;

"CHIP SELECT STATUS"
CSS1,CSS0       NODE    38,39;

"OVERFLOW"
OVF     NODE    40;

"EXTRA OUTPUT"
EXTRA   PIN     10;

Q = [Q3,Q2,Q1,Q0];
IQ = [IQ3,IQ2,IQ1,IQ0];
CSS = [CSS1,CSS0];
SEL = [SEL2,SEL1,SEL0];

EQUATIONS

Q0:= !IQ0&!DELUP;
Q0.R= IQ0#DELUP;

Q1:= !IQ1&IQ0&!(IQ==13)&!DELUP;
Q1.R= IQ1&IQ0#DELUP;

Q2:= !IQ2&IQ1&IQ0&!DELUP;
Q2.R= (IQ2&IQ1&IQ0#(IQ==13))#DELUP;

Q3:= IQ2&IQ1&IQ0&!DELUP;
Q3.R= (IQ==13)#DELUP;

DELUP:= UPDATE;
DELUP.R= !UPDATE;

CSS0:= !DELUP&!CSS0&(IQ==13);
CSS0.R= DELUP#CSS0&(IQ==13)#(IQ==15);

CSS1:= !DELUP&CSS0&(IQ==13);
CSS1.R= DELUP#(IQ==15);
```

```
SEL0:= !(((CSS==0)&!(IQ==13)#DELUP)&DATA&!OVF&UPDATE);
SEL0.R= ((CSS==0)&!(IQ==13)#DELUP)&DATA&!OVF&UPDATE;

SEL1:= DELUP;
SEL1.R= ((CSS==1)&!(IQ==13)&!DELUP#(CSS==0)&(IQ==13)&!DELUP)&DATA&!OVF&UPDATE;

SEL2:= DELUP;
SEL2.R= ((CSS==2)&!(IQ==13)#(CSS==1)&(IQ==13))&!DELUP&DATA&!OVF&UPDATE;

OVF:= !LONG&(CSS==1)&(IQ==12)&!UPDATE#LONG&(CSS==2)&(IQ==12)&!UPDATE;
OVF.R = UPDATE#(IQ==15);

EXTRA:= !LONG&(CSS==1)&(IQ==12)&!UPDATE#LONG&(CSS==2)&(IQ==12)&!UPDATE;
EXTRA.R= UPDATE#(IQ==15);

TEST_VECTORS ([CLK,PR,IQ,LONG,UPDATE,DATA] -> [SEL,Q])

[C,1,0,0,0,0] -> [7,15];
[C,0,15,0,0,1] -> [7,0];
[C,0,0,0,0,1] -> [7,1]; "UPDATE ON POWER ON"
[C,0,1,0,0,1] -> [7,2]; "COUNT : 2"
[C,0,2,0,0,1] -> [7,3]; "3"
[C,0,3,0,0,1] -> [7,4]; "4"
[C,0,4,0,0,1] -> [7,5]; "5"
[C,0,5,0,0,1] -> [7,6]; "6"
[C,0,6,0,0,1] -> [7,7]; "7"
[C,0,7,0,0,1] -> [7,8]; "8"
[C,0,8,0,0,1] -> [7,9]; "9"
[C,0,9,0,0,1] -> [7,10]; "10"
[C,0,10,0,0,1] -> [7,11]; "11"
[C,0,11,0,0,1] -> [7,12]; "12"
[C,0,12,0,0,1] -> [7,13]; "13"
[C,0,13,0,0,1] -> [7,0]; "0"
[C,0,0,0,0,1] -> [7,1]; "1"
[C,0,1,0,0,1] -> [7,2]; "2"
[C,0,2,0,1,1] -> [5,3]; "UPDATE"

[C,0,3,0,0,1] -> [7,0];
[C,0,2,0,0,1] -> [7,1];
[C,0,1,0,0,1] -> [7,2]; "COUNT : 2"
[C,0,2,0,0,1] -> [7,3]; "3"
[C,0,3,0,0,1] -> [7,4]; "4"
[C,0,4,0,0,1] -> [7,5]; "5"
[C,0,5,0,0,1] -> [7,6]; "6"
[C,0,6,0,0,1] -> [7,7]; "7"
[C,0,7,0,0,1] -> [7,8]; "8"
[C,0,8,0,0,1] -> [7,9]; "9"
[C,0,9,0,0,1] -> [7,10]; "10"
[C,0,10,0,0,1] -> [7,11]; "11"
[C,0,11,0,0,1] -> [7,12]; "12"
[C,0,12,0,0,1] -> [7,13]; "13"
[C,0,13,0,0,1] -> [7,0]; "0"
[C,0,0,0,0,1] -> [7,1]; "1"
[C,0,1,0,0,1] -> [7,2]; "COUNT : 2"
[C,0,2,0,0,1] -> [7,3]; "3"
[C,0,3,0,0,1] -> [7,4]; "4"
[C,0,4,0,0,1] -> [7,5]; "5"
[C,0,5,0,0,1] -> [7,6]; "6"
[C,0,6,0,0,1] -> [7,7]; "7"
[C,0,7,0,0,1] -> [7,8]; "8"
[C,0,8,0,0,1] -> [7,9]; "9"
[C,0,9,0,0,1] -> [7,10]; "10"
[C,0,10,0,0,1] -> [7,11]; "11"
[C,0,11,0,0,1] -> [7,12]; "12"
[C,0,12,0,0,1] -> [7,13]; "13"
```

```
[C,0,13,0,1,1] -> [7,0]; "OVERFLOW"

[C,1,0,0,0,0] -> [7,15];
[C,0,15,0,0,1] -> [7,0];
[C,0,0,0,1,0] -> [7,1];
[C,0,1,0,0,0] -> [7,0];
[C,0,0,0,0,0] -> [7,1];
[C,0,1,0,0,0] -> [7,2];
[C,0,2,0,1,1] -> [6,3];
[C,0,3,0,0,0] -> [7,0];
[C,0,0,0,1,1] -> [6,1];
[C,0,1,0,0,0] -> [7,0];
[C,0,0,0,0,0] -> [7,1];
[C,0,1,0,0,0] -> [7,2];
[C,0,2,0,1,1] -> [6,3];
[C,0,3,0,1,1] -> [6,0];
[C,0,0,0,1,1] -> [6,0];
[C,0,0,0,1,1] -> [6,0];
[C,0,0,0,1,1] -> [6,0];
[C,0,0,0,0,0] -> [7,0];
[C,0,0,0,1,1] -> [6,1];
[C,0,1,0,0,0] -> [7,0];
[C,0,0,0,0,0] -> [7,1];
[C,0,1,0,0,0] -> [7,2];
[C,0,2,0,1,1] -> [6,3];
[C,0,3,0,1,1] -> [6,0];
[C,0,0,0,1,1] -> [6,0];


TEST_VECTORS([CLK,PR,IQ,LONG,UPDATE,DATA] -> EXTRA)

[C,0,15,0,0,1] -> 0;
[C,0,1,0,0,1] -> 0; "COUNT : 2"
[C,0,2,0,0,1] -> 0; "3"
[C,0,3,0,0,1] -> 0; "4"
[C,0,4,0,0,1] -> 0; "5"
[C,0,5,0,0,1] -> 0; "6"
[C,0,6,0,0,1] -> 0; "7"
[C,0,7,0,0,1] -> 0; "8"
[C,0,8,0,0,1] -> 0; "9"
[C,0,9,0,0,1] -> 0; "10"
[C,0,10,0,0,1] -> 0; "11"
[C,0,11,0,0,1] -> 0; "12"
[C,0,12,0,0,1] -> 0; "13"
[C,0,13,0,0,1] -> 0; "0"
[C,0,0,0,0,1] -> 0; "1"
[C,0,1,0,0,1] -> 0; "COUNT : 2"
[C,0,2,0,0,1] -> 0; "3"
[C,0,3,0,0,1] -> 0; "4"
[C,0,4,0,0,1] -> 0; "5"
[C,0,5,0,0,1] -> 0; "6"
[C,0,6,0,0,1] -> 0; "7"
[C,0,7,0,0,1] -> 0; "8"
[C,0,8,0,0,1] -> 0; "9"
[C,0,9,0,0,1] -> 0; "10"
[C,0,10,0,0,1] -> 0; "11"
[C,0,11,0,0,1] -> 0; "12"
[C,0,12,0,0,1] -> 1; "13"
[C,0,13,0,0,1] -> 1; "OVERFLOW"
[C,0,0,0,1,1] -> 0; "NO ACTION ON OVERFLOW"

END CONTROLLER;
```

Appendix C. Input file for the shift register

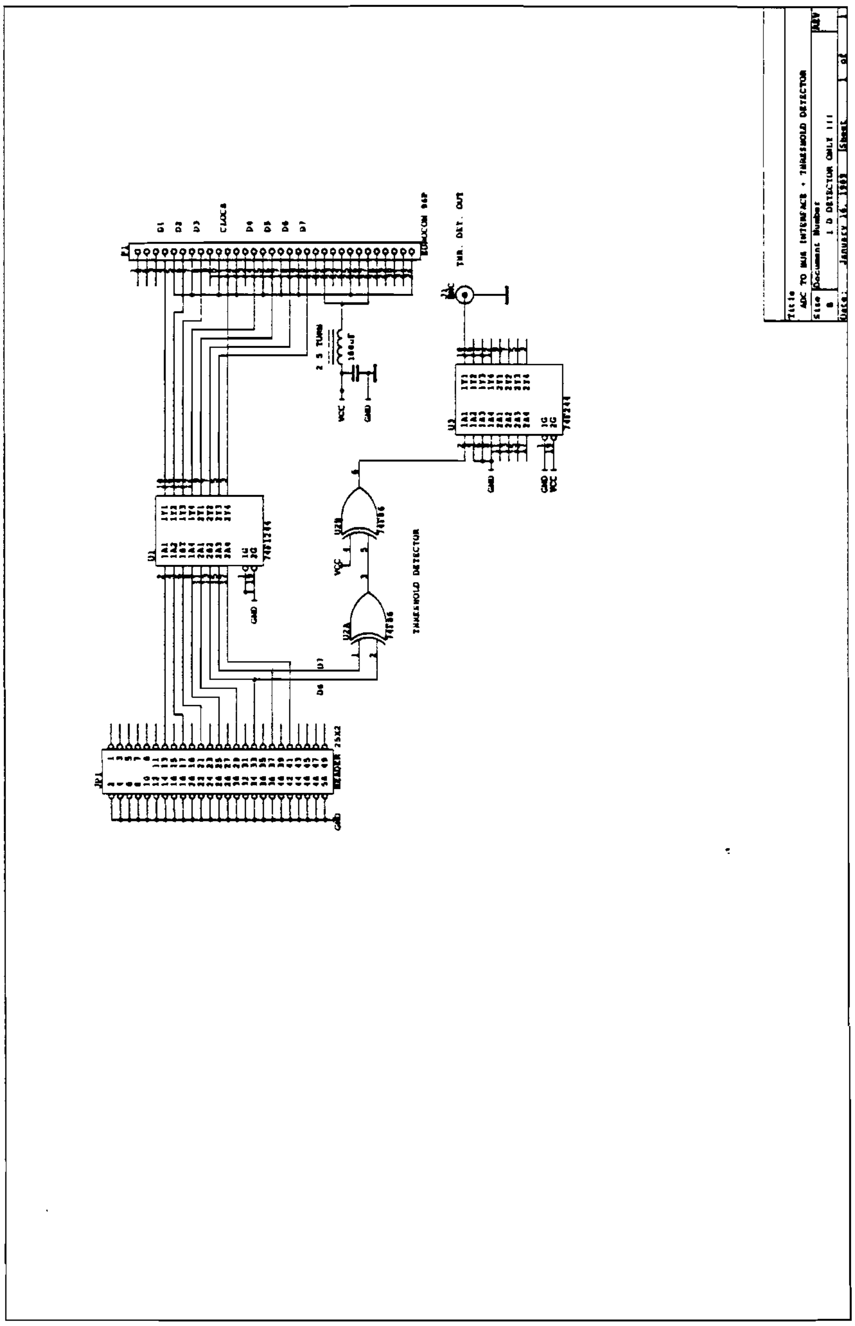
```
MODULE SHIFTER00;

FLAG '-R1';

TITLE 'RAM SHIFTER BY RENE VAN DER VLEUTEN *** 18-10-1988'

RS00 DEVICE 'F105';

CLK      PIN      1;
C = .C.;

"EXTERNAL OUTPUTS"
Q0,Q2,Q4,Q6,Q8,Q10,Q12,Q13       PIN      18,17,16,15,13,12,11,10;

"INTERNAL OUTPUTS"
Q1,Q3,Q5,Q7,Q9,Q11               NODE     37,38,39,40,41,42;

"EXTERNAL FEEDBACK"
S0,S1,S3,S5,S7,S9,S11,S13        PIN      20,21,22,23,24,25,26,27;

"ADDRESS"
A2,A4,A6,A8,A10,A12      PIN      7,6,5,4,3,2;

"SELECT"
SEL      PIN      9;

S = [S13,S11,S9,S7,S5,S3,S1,S0];
A = [A12,A10,A8,A6,A4,A2];
Q = [Q13,Q12,Q11,Q10,Q9,Q8,Q7,Q6,Q5,Q4,Q3,Q2,Q1,Q0];

EQUATIONS

Q0 := S0;
Q0.R = !S0;

Q1 := S1;
Q1.R = !S1;

Q2 := Q1#(!A2&!SEL);
Q2.R = !(Q1#(!A2&!SEL));

Q3 := S3;
Q3.R = !S3;

Q4 := Q3#(!A4&!SEL);
Q4.R = !(Q3#(!A4&!SEL));

Q5 := S5;
Q5.R = !S5;

Q6 := Q5#(!A6&!SEL);
Q6.R = !(Q5#(!A6&!SEL));

Q7 := S7;
Q7.R = !S7;

Q8 := Q7#(!A8&!SEL);
Q8.R = !(Q7#(!A8&!SEL));

Q9 := S9;
Q9.R = !S9;

Q10 := Q9#(!A10&!SEL);
Q10.R = !(Q9#(!A10&!SEL));

Q11 := S11;
```

```
Q11.R = !S11;

Q12 := Q11#(!A12&!SEL);
Q12.R = !(Q11#(!A12&!SEL));

Q13 := S13;
Q13.R = !S13;

TEST_VECTORS ( [CLK,!SEL,!A,S] -> Q)

[C,0,0,0] -> ^H1554; "POWER-ON PRESET"
[C,0,0,1] -> 1; "SET Q0"
[C,0,0,2] -> 2; "SET Q1"
[C,0,1,0] -> 4; "SET Q2"
[C,0,0,4] -> 8; "SET Q3"
[C,0,2,0] -> 16; "SET Q4"
[C,0,0,8] -> 32; "SET Q5"
[C,0,4,0] -> 64; "SET Q6"
[C,0,0,16] -> 128; "SET Q7"
[C,0,8,0] -> 256; "SET Q8"
[C,0,0,32] -> 512; "SET Q9"
[C,0,16,0] -> 1024; "SET Q10"
[C,0,0,64] -> 2048; "SET Q11"
[C,0,32,0] -> 4096; "SET Q12"
[C,0,0,128] -> ^H2000; "SET Q13"
[C,0,0,0] -> 0; "ALL 0"

[C,1,1,0] -> 4; "SET Q2"
[C,1,2,0] -> 16; "SET Q4"
[C,1,4,0] -> 64; "SET Q6"
[C,1,8,0] -> 256; "SET Q8"
[C,1,16,0] -> 1024; "SET Q10"
[C,1,32,0] -> 4096; "SET Q12"
[C,1,0,0] -> 0; "ALL 0"

[C,0,0,1] -> 1; "SET Q0"
[C,0,0,2] -> 2; "SET Q1"
[C,0,1,0] -> 4; "SET Q2"
[C,0,0,4] -> 8; "SET Q3"

[C,1,1,0] -> 20;
[C,0,2,0] -> 0; "SET Q4"
[C,0,0,8] -> 32; "SET Q5"
[C,0,4,0] -> 64; "SET Q6"
[C,0,0,16] -> 128; "SET Q7"
[C,0,8,0] -> 256; "SET Q8"
[C,0,0,32] -> 512; "SET Q9"
[C,0,16,0] -> 1024; "SET Q10"
[C,0,0,64] -> 2048; "SET Q11"
[C,0,32,0] -> 4096; "SET Q12"
[C,0,0,128] -> ^H2000; "SET Q13"
[C,0,0,0] -> 0; "ALL 0"

END SHIFTER00
```

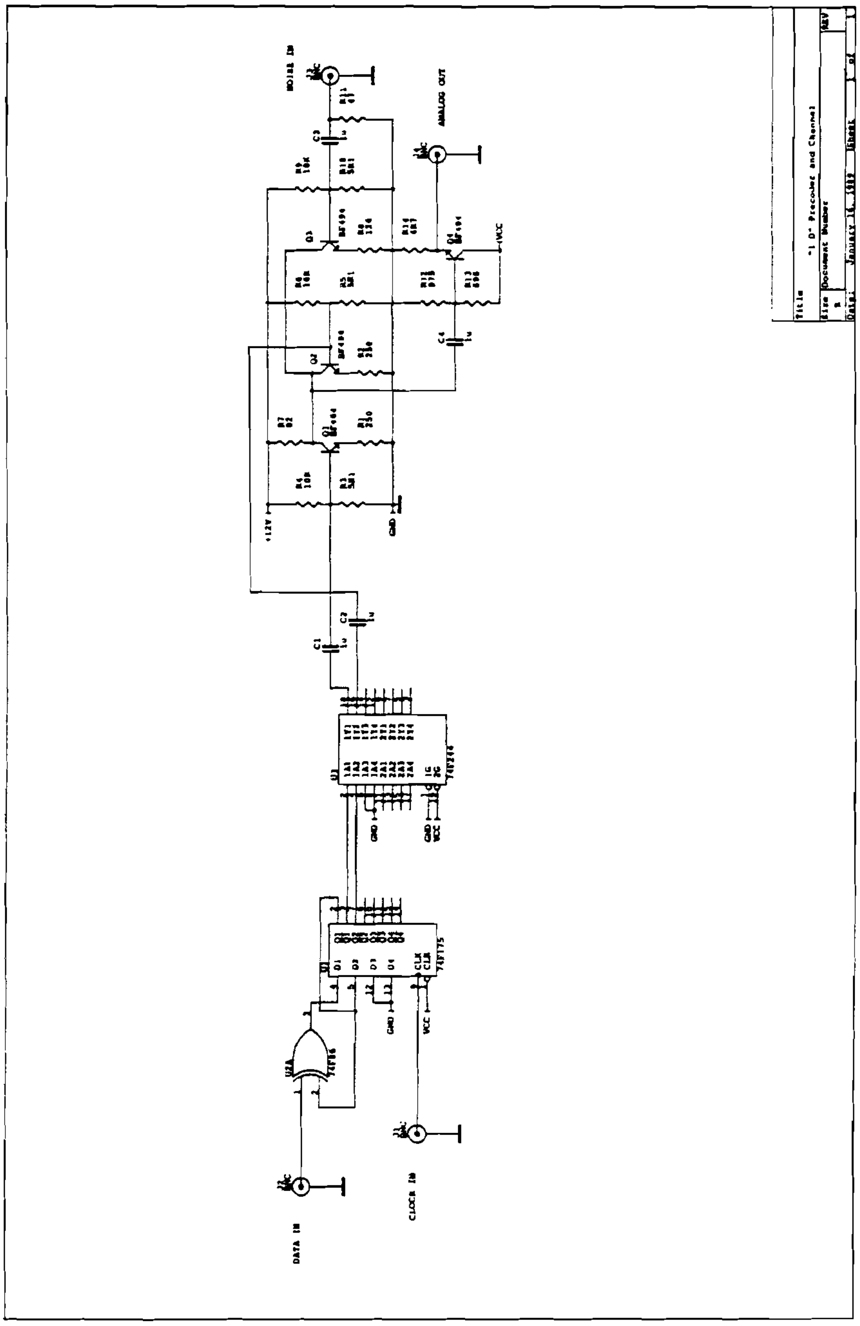
Appendix D. Description of the detector's bus

Pin number	Name/Function
1	-
2	-
3	-
4	D1
5	GND
6	D2
7	GND
8	D3
9	-
10	GND
11	CLOCK1
12	GND
13	CLOCK2
14	D4
15	GND
16	D5
17	GND
18	D6
19	GND
20	D7
21	GND
22	VCC
23	GND
24	Q1
25	GND
26	OVF
27	VCC
28	Q2
29	-
30	-
31	-
32	GND

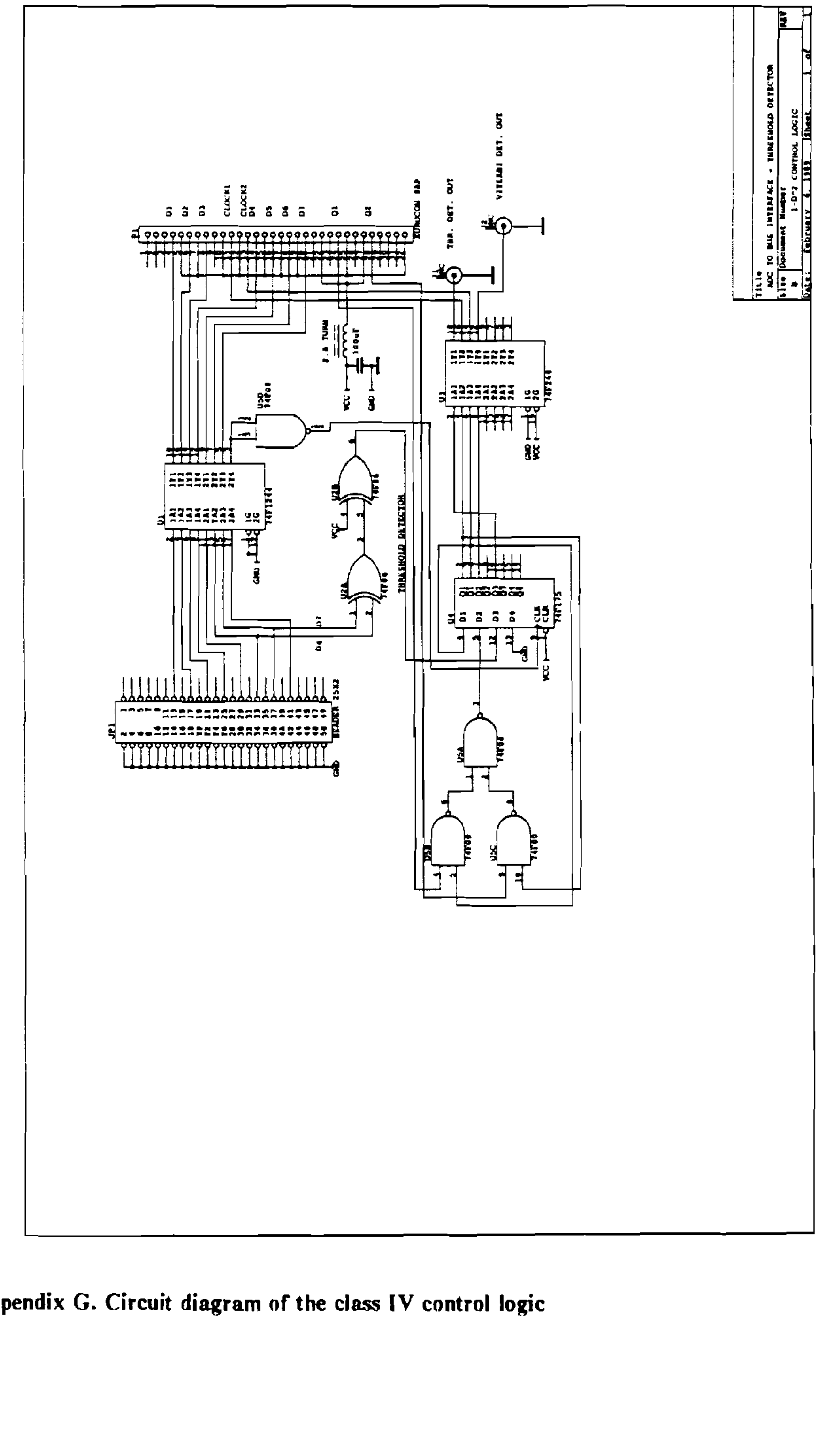
Appendix E. Circuit diagram of the 1-D interface and the threshold detector



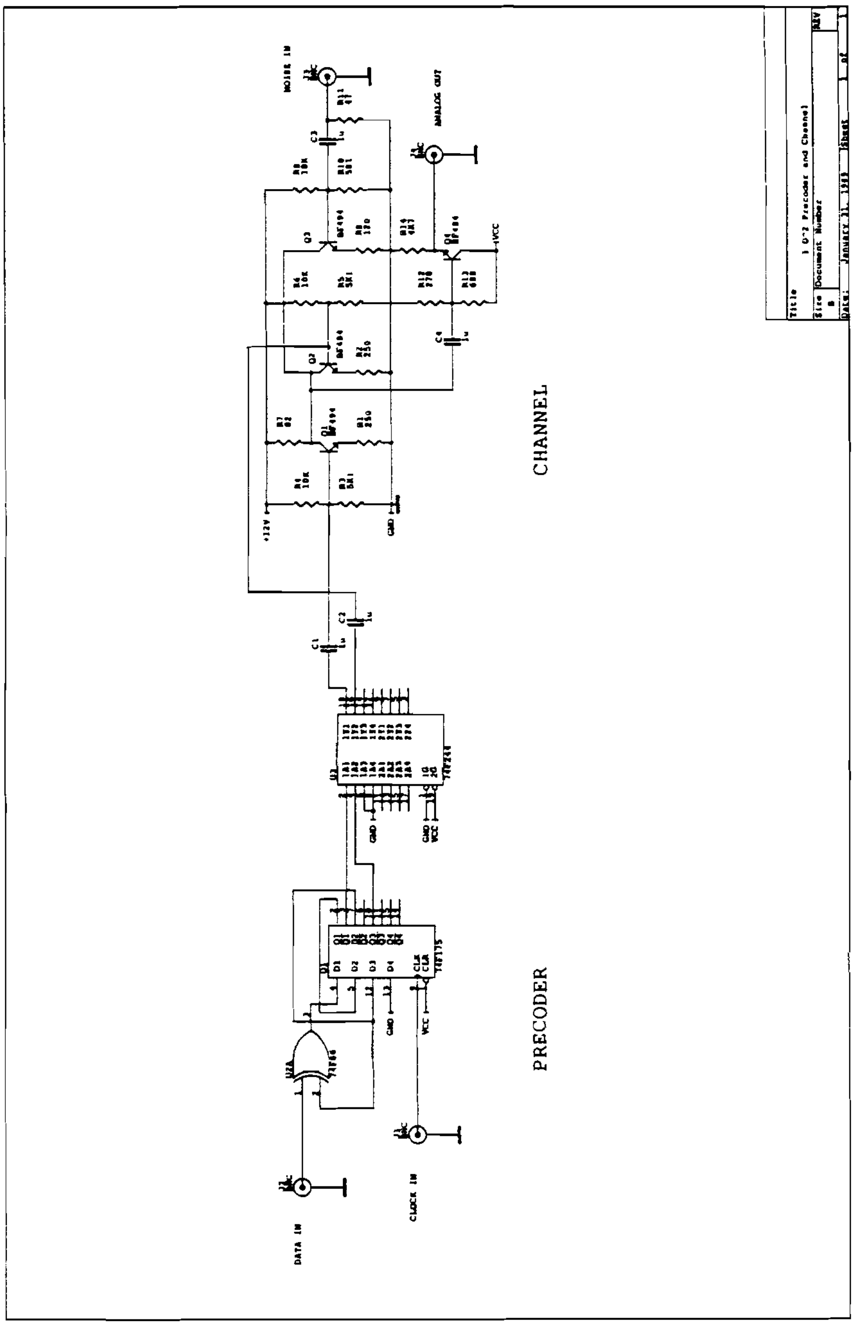
Appendix F. Circuit diagram of the 1-D precoder and channel



Appendix G. Circuit diagram of the class IV control logic



Appendix H. Circuit diagram of the class IV precoder and channel


CHANNEL
PRECODER
NOISE IN
ANALOG OUT
DATA IN
CLOCK IN
+12V
GND
VCC
74F175
74F244
74F86
C1
C2
C3
C4
Q1
Q2
Q3
Q4
Title
Size
Document Number
REV
Date:
Sheet