Philips tech. Rev. 40, 166-172, 1982, No. 6

Error correction and concealment in the Compact Disc system

H. Hoeve, J. Timmermans and L. B. Vries

Introduction

When analog signals such as audio signals are transmitted and recorded via an intervening system such as a gramophone record it is difficult to properly correct signal errors that have occurred in the path between the audio source and the receiving end. With suitably coded digital signals, however, a practical means of error correction does exist. We shall demonstrate this with the following example [1].

Suppose that a message of 12 binary units (bits) has to be transmitted (a stream of digital information can always be divided into groups of a particular size for transmission). The 12 bits x_{ij} are arranged as follows in a matrix, in which all x_{ij} can only have the value 0 or 1:

$$\begin{matrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{matrix}$$

To discover at the receiving end whether the message read there contains an error, and, if so, what the error is, one extra bit (called a 'parity bit') is added to each row and column: x_{15}, x_{25}, x_{35} and x_{41}, x_{42}, x_{43}, x_{44} respectively. These parity bits provide a check on the correctness of the message received. The values assigned to them are such that x_{i5} $(i = 1, 2, 3)$ makes the number of ones in row i even, for example, while x_{4j} $(j = 1, 2, 3, 4)$ makes the number of ones in column j even. Next, a further parity bit (x_{45}) is added that has a value such that the number of ones in the block is made even. This results in the following matrix of four rows and five columns:

$$\begin{matrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} \\ x_{31} & x_{32} & x_{33} & x_{34} & x_{35} \\ x_{41} & x_{42} & x_{43} & x_{44} & x_{45} \end{matrix}$$

It is easy to verify that the number of ones in the last row is also even, and so is the number of ones in the last column. If now a bit, say x_{23}, is incorrectly read at the receiving end, then the number of ones in the second row and the number of ones in the third column will no longer be even, and once this has been ascertained, a 0 at position x_{23} can be changed into a 1, or vice versa, thus correcting the error.

So as to be able in this way to correct one error in 12 information bits, it is necessary to send a total of 20 bits instead of 12: the 'code word' of $n = 20$ bits consists of $k = 12$ information bits and $n - k = 8$ parity bits. The (n,k) code used here, a (20,12) code, makes it possible to correct single errors and also, as can easily be verified, to detect various multiple bit errors.

The 'rate' of an error-correcting code is taken to be the ratio of the number of information bits to the total number of bits per code word: k/n. The (20,12) code does not have a high rate, because it requires a relatively large number of parity bits. For the Compact Disc this would entail a considerable reduction in the playing time.

The theory of error-correcting codes [2] gives design methods that entail a minimal addition of parity bits when certain correction criteria are satisfied. An important concept in this theory is the 'distance' and in particular the 'minimum distance' d_m between two code words of n bits. Distance here is taken to be the number of places in which the bits of the two code words differ from each other. In the above example the minimum distance d_m is equal to 4: if one single bit of the k information bits changes, then the two parity bits of the associated row and column change at the same time, as does the one at the bottom right-hand corner, x_{45}, so that the entire code word has changed at four places. Theory tells us that to correct all the combinations of t errors occurring within one word, the minimum distance must be at least $2t + 1$. To correct single errors, therefore, the minimum distance need be no greater than three. Examples of this are the single-error-correcting Hamming codes [4].

Ir H. Hoeve and J. Timmermans are with the Philips Audio Division, Eindhoven; Ir L. B. Vries is with Philips Research Laboratories, Eindhoven.

The statement that a code word x, which is received as a different word z because of t errors, can be restored to its original form if the minimum distance is $2t + 1$, can be seen from *fig. 1.* A decoder provided with a list in which all the code words are stored can compare z with each of these code words and thus recover the correct code word unambiguously.

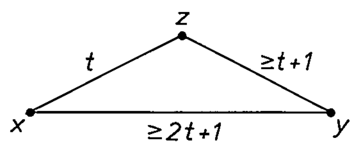


Fig. 1. The original transmitted code word x is received as z owing to t bit errors. Any code word y differing from x lies at a distance $\geqslant 2t + 1$ from x. To cause z to change into y it is necessary to change at least $t + 1$ bits. It follows that x is the only code word that has a distance t from z.

On the theory of block codes

In the foregoing we have shown with a simple example that it is possible to correct errors. Error-correcting systems do have their limitations, of course. To make this clear we shall consider how error-correcting codes should be designed to guarantee a specific measure of correction, with as few extra bits as possible added to the digital information to be transmitted. It will help if we first say something about the theory of block codes.

So that known and efficient error-correcting codes can be applied, groups of bits are formed by adding together a fixed number s of consecutive bits; these groups are called symbols. With these symbols we now set to work in the same way as with the bits in the foregoing: the information symbols are grouped together to form blocks with a length of k symbols. For error-correction we now add parity symbols to expand each block of k information symbols into a code word of n symbols. The $n-k$ parity symbols to be added are calculated from the k information symbols, and this is done in such a way as to make the error correction as effective as possible. Thus, of the very large number of possibly different words of n symbols only a small fraction, i.e. $2^{(k-n)s}$, become code words (see *fig. 2*). For a given encoding system both n and k are fixed.

As already mentioned in the article on modulation in the Compact Disc system [3], the start of each word is marked by a synchronization symbol. (A word marked by a synchronization symbol is called a 'frame'.) The error-correcting system therefore knows when a new word begins, and the only errors it has to deal with are errors that occur in the transmission of data.

There are two kinds of errors: those that are distributed at random among the individual bits, the random errors, and errors that occur in groups that may cover a whole symbol or a number of adjacent symbols; these are called 'bursts' of errors. They can occur on a disc as a result of dirt or scratches, which interfere with the read-out of a number of adjacent pits and lands.

The best code for correcting random errors is the one that, for given values of n and k, is able to correct the largest number of independent errors within one code word. In the detection and correction of errors the symbols have to undergo a wide variety of operations. Large k-values (as with the Compact Disc) require extremely complex computing hardware. Practice has shown that the only acceptable solution to this problem is to choose a convenient code. And the only usable codes that enter into consideration, so far as we know at present, are the 'linear codes'.

A code is linear if it obeys the following rule:

If $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ are code words, then their sum $x + y = (x_1 + y_1, \ldots, x_n + y_n)$ is also a code word.

In this sum the symbol $x_i + y_i$ is produced — irrespective of the number of bits s per symbol — by a modulo-2 bit addition. The special feature of the linear code is thus that each sum of code words yields another code word, i.e. a word of n symbols, which also belongs to the small fraction of symbol combinations permitted in the code.

It is this linearity feature that makes it possible to cut down considerably on the extent of the decoding equipment. The *Reed-Solomon codes* [2] are examples of such a linear code. They are also extremely efficient,

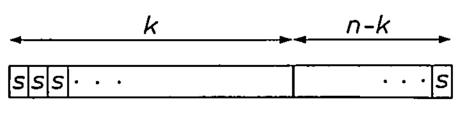


Fig. 2. A code word of length n consists of an information block of k symbols and a parity block of $n-k$ symbols; each symbol comprises s bits. The number of possible words of n symbols is 2^{ns}. The parity bits are fixed for each combination of the ks information bits in accordance with established encoding rules. The number of code words is thus 2^{ks}. It follows that the fraction $2^{(k-n)s}$ of the number of possible words consists of code words.

[1] This example is taken from S. Lin, An introduction to error-correcting codes, Prentice-Hall, Englewood Cliffs 1970.

[2] See for example F. J. MacWilliams and N. J. A. Sloane, The theory of error-correcting codes, North-Holland, Amsterdam 1978.

[3] See J. P. J. Heemskerk and K. A. Schouhamer Immink, Compact Disc: system aspects and modulation, this issue, p. 157.

since for every $s > 1$ and $n \leqslant 2^s - 1$ there exists a Reed-Solomon code with

$$d_m = n - k + 1.$$

Together with the general condition $d_m \geqslant 2t + 1$ mentioned earlier, which the minimum distance must satisfy for the correction of t errors, this yields $n - k \geqslant 2t$. Put in another way: to correct t symbol errors it is sufficient to add $2t$ parity symbols. (By 'distance' between two words we mean here the number of positions in which there are different symbols in the two words; it does not matter how many corresponding bits differ from each other within the corresponding symbols.)

In practice a less cumbersome algorithm will generally be used for error correction than the comparison with the aid of a list of all the code words, as described on page 167. We shall not consider the details of the algorithm here. We shall, however, try to give some idea of the manner in which error bursts are tackled with block codes. To do this we must introduce the concept of 'erasure'.

The position (i) of a particular symbol (x_i) in a transmitted code word (x) is called an erasure position if a decoder-independent device signals that the value of x_i is not reliable. This value is then erased, and in the decoding procedure the correct value has to be calculated. The decoding is now simpler and quicker because the positions at which errors can occur are known. (We assume for the moment that no errors occur outside the erasure positions.) The advantage of correcting by means of the erasures is expressed quantitatively by the following proposition:

If a code has a minimum distance d_m, then $d_m - 1$ erasures can be reconstituted.

Since the number of errors that can be corrected without erasure information is $\frac{1}{2}(d_m - 1)$ at most, the advantage of correcting by means of erasures is clear.

The proposition that for a minimum distance d_m, the number of erasures that can be reconstituted is $d_m - 1$, can be proved as follows:

Let x be the code word transmitted and z the word received. Let z be subject to a maximum of $d_m - 1$ erasures, so that it differs from x at a maximum of $d_m - 1$ positions, i.e. for the distance $d(x,z)$ between the code words z and x we have $d(x,z) \leqslant d_m - 1$. We now replace the symbols at all the erasure positions in z by other symbols, which gives a number of words that we denote by $\tilde{z}$; we try all the possible substitutions $\tilde{z}$. These words also differ from x at a maximum of $d_m - 1$ positions, i.e. $d(x,\tilde{z}) \leqslant d_m - 1$. Since all the code words different from x have a distance to x that is greater than or equal to the minimum distance d_m, the words $\tilde{z}$ include no other code word except x itself. It is therefore only necessary to find out which of the finite number of words $\tilde{z}$ represents a code word.

In the Compact Disc system the value of the analog signal to be reproduced is converted at every sampling instant into a binary number of 16 bits per audio channel. For error correction the digital information to be transmitted is divided into groups of eight bits, so that in each sampling operation four information symbols (consisting of audio bits) are generated. In fact, eight parity symbols are added to each block of 24 audio symbols [4]. The calculation of the parity symbols will not be dealt with here.

Cross-Interleaved Reed-Solomon Code

The error-correcting code used in the Compact Disc system employs not one but two Reed-Solomon codes (C_1, C_2), which are interleaved 'crosswise' (Cross-Interleaved Reed-Solomon Code, CIRC). For code C_1 we have: $n_1 = 32$, $k_1 = 28$, $s = 8$, and for C_2: $n_2 = 28$, $k_2 = 24$, $s = 8$. The rate of the CIRC we use is $(k_1/n_1)(k_2/n_2) = 3/4$.

For both C_1 and C_2 we have $2t = n - k = 4$, so that for each the minimum distance d_m is equal to $2t + 1 = 5$. This makes it possible to directly correct a maximum of two ($= t$) errors in one code word or to make a maximum of four ($= d_m - 1$) erasure corrections. A combination of both correction methods can also be used.

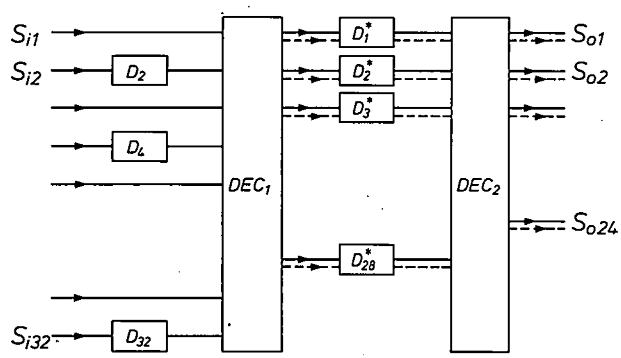


Fig. 3. Schematic representation of the decoding circuit for CIRC. The 32 symbols ($s_{i1}, \ldots, S_{i32}$) of a frame (24 audio symbols and 8 parity symbols) are applied in parallel to the 32 inputs. The delay lines D_{2i} ($i = 1, \ldots, 16$) have a delay equal to the duration of one symbol, so that the information of the 'even' symbols of a frame is cross-interleaved with that of the 'odd' symbols of the next frame. The decoder DEC_1 is designed in accordance with the encoding rules for a Reed-Solomon code with $n_1 = 32$, $k_1 = 28$, $s = 8$. It corrects one error, and if multiple errors occur passes them on unchanged, attaching to all 28 symbols an erasure flag, sent via the dashed lines. Owing to the different lengths of the delay lines D_j^* ($j = 1, \ldots, 28$), errors that occur in one word at the output of DEC_1 are 'spread' over a number of words at the input of DEC_2. This has the effect of reducing the number of errors per DEC_2 word. The decoder DEC_2 is designed in accordance with the encoding rules for a Reed-Solomon code with $n_2 = 28$, $k_2 = 24$, $s = 8$. It can correct a maximum of four errors by means of the erasure-positions method. If there are more than four errors per word, 24 symbol values are passed on unchanged, and the associated positions are given an erasure flag via the dashed lines. $S_{o1}, \ldots, S_{o24}$ outgoing symbols.

Decoding circuit

The error-correction circuit [5] is shown schematically in *fig. 3*; *fig. 4* is a photograph of the actual IC. The circuit consists of two decoders, *DEC*, and a number of delay lines, *D* and *D**. The input signal is a sequence of frames [6].

The 32 symbols of a frame are applied in parallel to the 32 inputs. In passing through the delay lines $D_2, D_4, \ldots, D_{32}$, each of length equal to the duration of one symbol, the even symbols of a frame with the odd symbols of the next frame form the words that are fed to the decoder DEC_1. (The symbols of the frames are 'cross-interleaved'. In fact they are 'de-interleaved', because the 'interleaving' [4] has already taken place, before the information was recorded on the disc.) If there are no errors in the transmission path, the decoder DEC_1 will receive code words that correspond to the encoding rules for C_1, and it will pass on 28 symbols unchanged. DEC_1 is designed for correcting one error. If it receives a word with a double or triple error, that event is detected with certainty; all the symbols of the received word are passed on unchanged, and all 28 positions are provided with an erasure flag. The same happens in principle for events from 4 to 32 errors, but here there is a small probability ($\approx 2^{-19}$) that this detection will fail. We shall return to this probability later.

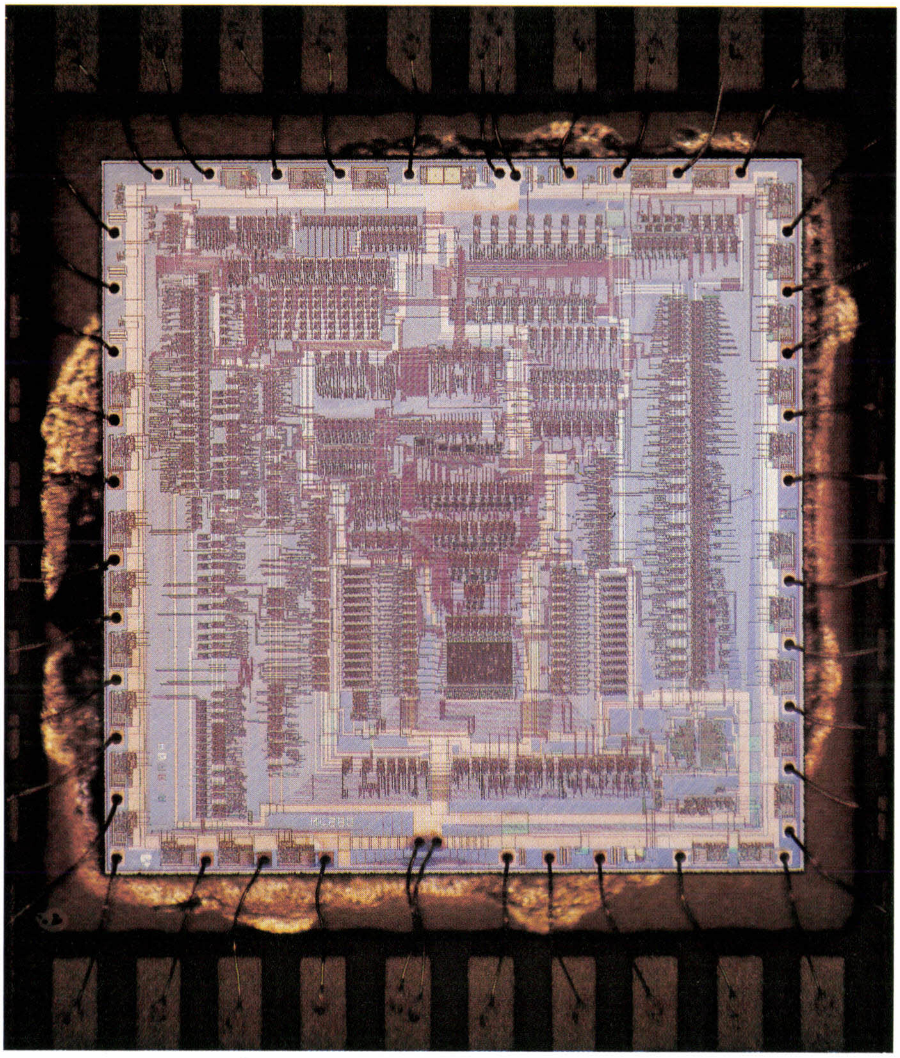


Fig. 4. The integrated circuit for error detection and correction is fabricated in n-channel MOS silicon-gate technology. It has an area of 45 mm^2 and contains about 12 000 gates.

The symbols arrive via the delay lines $D_1^*, \ldots, D_{28}^*$, which differ from each other in length, at the input of

[4] The calculation and addition of the parity symbols take place in the encoding circuit *PAR COD* in fig. 2 of the article of note [3]. Delay lines are used for interleaving the audio and parity symbols.

[5] This circuit corresponds to the *ERCO* chip in fig. 3 of the article by M. G. Carasso, J. B. H. Peek and J. P. Sinjou, this issue, p. 151.

[6] In fig. 9 of [3] a frame of this kind is represented by the bit stream B_2, from which the C&D symbol has already been removed.

DEC_2 in different words. If there are no errors present, DEC_2 will receive words that correspond to the encoding rules for C_2, and it will pass on 24 audio symbols unchanged. DEC_2 can correct up to four errors, by means of erasure decoding. (In the current Compact Disc system full use is not made of this facility: DEC_2 is arranged in such a way that only two errors are corrected.) If DEC_2 receives a word containing five or more errors with given erasure positions, it will pass on 24 symbols unchanged, but provided with an erasure flag at the appropriate positions; this flag has in fact already been assigned by DEC_1. A value for the erroneous samples can still be calculated with the aid of a linear interpolation.

As already mentioned, DEC_1 has been designed to allow the correction of single errors, and the detection of double and triple errors. The probability that DEC_1 will not detect quadruple or higher multiple errors is only about 2^{-19}. It may seem strange that the possibility of correcting two random errors is not utilized: in fact it would considerably increase the chance of DEC_1 failing to detect quadruple or higher multiple errors.

The probability P of quadruple or higher multiple errors passing DEC_1 without being detected can be approximated by the expression

$$P = \frac{1 + n_1(2^s - 1)}{2^{s(n_1 - k_1)}} \approx 2^{-19}.$$

The numerator contains the number of error patterns with one error or none. (The factor $(2^s - 1)$ is the number of possibilities for one bit error per symbol; such a symbol can occur at n_1 positions. The value 1 is added because zero errors can be achieved in exactly one way.) This complete expression is to be related to the number of possibilities for filling in the parity: $2^{s(n_1-k_1)}$). For proof of this equation the reader is referred to the literature [7].

When a disc is used for the recording and read-out of digital signals there are few random errors; most errors then occur as bursts. This is because the dimensions of a pit are small in relation to the most common mechanical imperfections such as dirt and scratches. It is therefore very important that multiple errors of this type cannot pass DEC_1 without being indicated with a high degree of certainty.

Since the bursts are 'spread out' over several words at the input of DEC_2, the number of errors per word hardly ever exceeds the limit value $d_m - 1 = 4$. In this way most error bursts are fully corrected.

Specifications of CIRC

In assessing the quality of our CIRC decoder for Compact Disc applications its ability to correct both error bursts and random errors is of great importance.

The quality characteristics for the correction of bursts are the maximum fully correctable burst length and the maximum interpolation length. The first is determined by the design of the CIRC decoder and in our case amounts to about 4000 data bits, corresponding to a track length on the disc of 2.5 mm. The maximum interpolation length is the maximum burst length at which all erroneous symbols that leave the decoder uncorrected can still be corrected by linear interpolation between adjacent sample values. This 'length' is about 12 000 data bits; see the next section.

Random errors can also introduce multiple errors within one code word now and again; we shall return to this presently. The greater the relative number of errors ('bit error rate', BER) at the receiving end, the greater is the probability of uncorrectable errors. A measure for the performance of this system is the number of sample values that have to be reconstituted by interpolation for a given BER value per unit time. This number of sample values per unit time is called the sample interpolation rate. The lower this rate is at a given BER value, the better the quality of the system for random-error correction.

Table I. Specifications of CIRC.

Aspect	Specification
Maximum *completely* correctable burst length	≈ 4000 data bits (i.e. ≈ 2.5 mm track length on the disc)
Maximum interpolatable burst length in the *worst* case	≈ 12 300 data bits (i.e. ≈ 7.7 mm track length)
Sample interpolation rate	One sample every 10 hours at BER = 10^{-4}; 1000 samples per minute at BER = 10^{-3}
Undetected error samples (clicks)	Less than one every 750 hours at BER = 10^{-3}; negligible at BER $\leqslant 10^{-4}$
Code rate	3/4
Structure of decoder	One special LSI chip plus one random-access memory (RAM) for 2048 words of 8 bits
Usefulness for future developments	Decoding circuit can also be used for a four-channel version (quadraphonic reproduction)

An objective assessment of the quality of the error-correcting system also requires an indication of the number of errors that pass through unsignalled and are therefore not corrected by the system. These unsignalled and uncorrected errors may produce a clearly audible 'click' in the reproduction.

The main features of the CIRC system are summarized in *Table I*. Details of the calculation relating to the quality can be found in the literature [7].

Concealment of residual errors

The purpose of error concealment is to make the errors that have been detected but not corrected by the CIRC decoder virtually inaudible. Depending on the magnitude of the error to be concealed, this is done by interpolation or by muting the audio signal [8].

Two consecutive 8 bit symbols delivered by the decoder together form a 16 bit sample value. Since a sample value in the case of a detected error carries an erasure flag, the concealment mechanism 'knows' whether a particular value is reliable or not. A reliable sample value undergoes no further processing, but an unreliable one is replaced by a new value obtained by a linear interpolation between the (reliable) immediate neighbours. Sharp 'clicks' are thus avoided; all that happens is a short-lived slight increase in the distortion of the audio signal. With alternate correct and wrong sample values, the bandwidth of the audio signal is halved during the interpolation (10 kHz).

If the decoder delivers a sequence of wrong sample values, a linear interpolation does not help. In that case the concealment mechanism deduces from the configuration of the erasure flags that the signal has to be muted. This is done by rapidly turning the gain down and up again electronically, a procedure that starts 32 sampling intervals before the next erroneous sample values arrive. To achieve this the reliable values are first sent through a delay line with a length of 32 sampling intervals, while the unreliable values are processed immediately. The gain is kept at zero for the duration of the error and then turned up again in 32 sampling intervals. The gain variation follows a cosine curve (from 0 to 180° and from 180 to 360°) to avoid the occurrence of higher-frequency components. This also means that there are no clicks when the audio signal is muted, as in switching the player on and off, during an interval in playing or during the search procedure.

Maximum burst-interpolation length

Two associated 16 bit sample values, one from the left and one from the right audio channel, together form a sample set. If these sets were fed to the concealment circuit in the correct sequence, it would not be possible to interpolate more than one set from their reliable neighbouring sets. This would mean that in the case of an error longer than the maximum correctable burst length signal muting would very soon have to be applied.

By interleaving the sample sets it becomes possible to interpolate new sets for a given length of consecutive erroneous sets. This is done by alternating groups of 'even' sample sets with groups of 'odd' sets. Such a group, odd or even, can be interpolated from its neighbouring group or groups. The maximum burst-interpolation length is thus equal to the length of such a group. In our system we have grouped the twelve 16 bit sample values of a frame in the way shown in *fig. 5*. The odd and even groups are separated by the parity values Q. Since these are not necessary for the reconstitution of the original signal and may therefore permissibly be unreliable, they increase the interpolation length. The maximum length with this grouping is certainly seven or even eight sample values, for some error patterns.

The delay lines corresponding to D_i* (see fig. 3) in the encoder [4] have placed eight frames between two successive sample values, after interleaving. The maximum burst length that can always be interpolated is therefore 56 frames. This presupposes, of course, that we are working with sample values consisting of two immediately consecutive symbols; the distance between all successive symbols is four frames, however. This is also the work of the delay lines D_i*.

The delay lines corresponding to D_i (again fig. 3) in the encoder [4] now ensure, however, that this distance is alternatively three and five frames, after interleaving. The distance of five frames is responsible for a decrease in the maximum interpolation length from 56 to 51 frames. We have tacitly assumed here that the burst also comes within a block of eight frames. If we discount this assumption, there is still a reduction of a length of 1 frame − 2 symbols. The maximum burst length that can be interpolated with certainty has now become 50 frames + 2 symbols.

So far we have taken no account of random errors that can be interpolated; this is the subject of the next and final section. At this point we shall simply mention the effect of the interpolation of such errors on the maximum interpolation length.

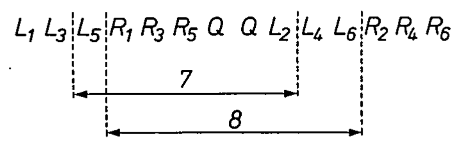


Fig. 5. Grouping of the sample values within a frame; L_i values for the left channel, R_i values for the right channel. For each sequence of seven unreliable values, new values can be calculated with certainty from reliable neighbours (e.g. if L_5 to L_2 are unreliable, the new values are interpolated from R_6 of the preceding frame and from the reliable values of the above frame). Given a favourable situation, new values can in fact be derived for eight consecutive values (e.g. values for R_1 up to L_6 from R_6 of the preceding frame, the reliable values of the above frame and L_1 of the succeeding frame).

[7] L. M. H. E. Driessen and L. B. Vries, Performance calculations of the Compact Disc error correcting code on a memoryless channel, in: 4th Int. Conf. on Video and data recording, Southampton 1982 (IERE Conf. Proc. No. 54), pp. 385-395.

[8] Error concealment takes place in the *CIM* chip in fig. 3 of the article of note [5].

To achieve good results in the treatment of random errors, the symbols are finally sent through a further set of delay lines Δ_i with a length of two frames. These delay lines, which serve purely and simply for 'restoring' uncorrected random errors, cause in their turn a reduction of the interpolation length by two frames. The final maximum burst length that is guaranteed capable of interpolation is thus 48 frames + 2 symbols, which corresponds to 12 304 bits.

Interpolation of random errors

If the symbols S_{0i} (fig. 3) after the decoder DEC_2 were already in the correct sequence, a pattern of errors might arise that would rule out any possibility of interpolation, even though there were no long error bursts. This would happen if DEC_1 failed to detect an error but DEC_2 had detected it, resulting in the rejection of the entire frame at the output of DEC_2. As described on page 170, however, the chance of DEC_1 failing is very small.

Since we prefer not to have to mute the audio signal, the concealment network contains a set of delay lines Δ_i, with a length of two frames, which ensure that the symbols of a single or double completely rejected frame from DEC_2 can still be interpolated from the reliable adjacent frames (see *fig. 6*). The probability that three completely rejected frames will occur within the interpolatable length determined by Δ_i is negligible.

After the symbols have passed through the delay lines Δ_i, they are in the correct sequence. Most of the errors have been corrected and the signal is ready for the digital-to-analog conversion [9].

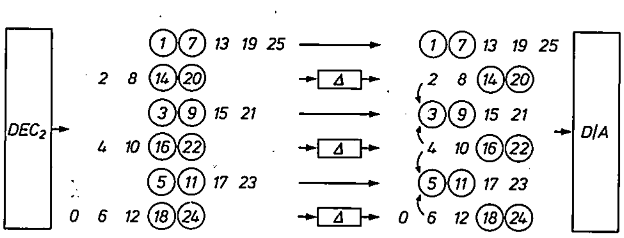
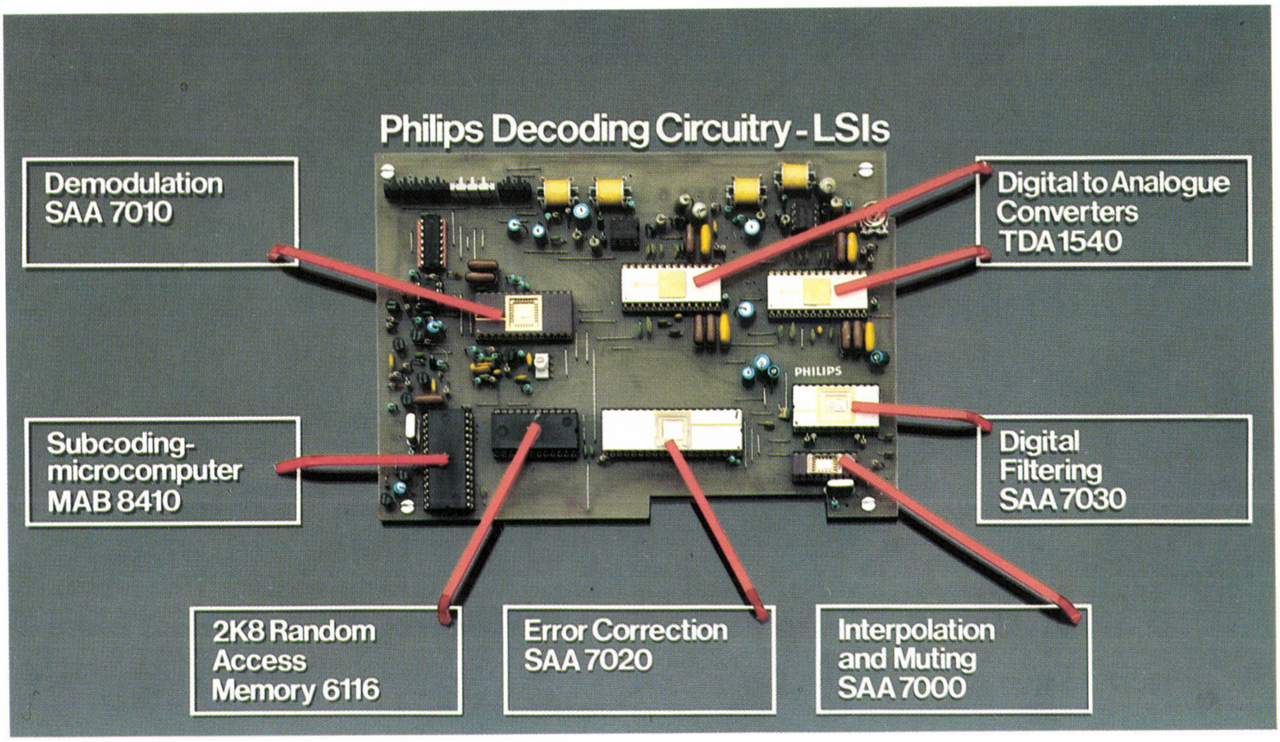


Fig. 6. The effect of the delay lines Δ_i with a length equal to the duration of two frames on the signal from the decoder DEC_2. Each number represents a sample set, and a circle around a number is an erasure flag. A frame, consisting of 24 symbols or 6 sample sets, is represented by a complete column. The succession of frames on the left in the figure (sample sets that are irrelevant in the present context have been omitted) comes direct from DEC_2 and comprises a pattern of random errors, causing the total rejection of two consecutive frames (1, 14, 3, ... 11, 24). It can be seen, however, that the chosen grouping enables a new value from reliable neighbours to be interpolated for each unreliable sample set, e.g. a value for 5 from 4 and 6. After passing through the delay lines Δ_i with a length equal to the duration of two frames, the sample sets are applied in the correct sequence to the D/A converter. If a frame in the succession of frames on the right in the figure were to be completely rejected, no interpolation would be possible.

[9] See D. Goedhart, R. J. van de Plassche and E. F. Stikvoort, Digital-to-analog conversion in playing a Compact Disc, this issue, p. 174.

Summary. After an example showing how errors in a digital signal can be corrected, the article deals with the theory of block codes. The treatment of random errors and error bursts is discussed. Error correction in the Compact Disc system uses a Cross-Interleaved Reed-Solomon Code (CIRC), which is a combination of a (32,28) and a (28,24) code. One of the two decoders in the CIRC decoding circuit corrects single errors, the other corrects double errors. The residual errors are interpolated linearly to a length of up to 12 000 bits, and longer errors are muted. The interpolation and the signal muting take place in a separate chip, whose configuration is briefly discussed.



The most important printed circuit in the player contains a number of ICs in VLSI technology for reconstituting the analog audio signal from the digital signal read from the Compact Disc. The chips for error correction and digital-to-analog conversion are discussed in this issue.