ON A TAPE FORMAT FOR RELIABLE PCM MULTI-CHANNEL TAPE RECORDERS

1669 (K-1)

Kunimaro Tanaka, Minoru Ozaki,
Toru Inoue, Tetsuya Yamaguchi
Mitsubishi Electric Corporation
Amagasaki, Hyogo, Japan

Presented at the 66th Convention 1980 May 6 through 9 Los Angeles

AES

AN AUDIO ENGINEERING SOCIETY PREPRINT

ON A TAPE FORMAT FOR RELIABLE PCM MULTI-CHANNEL TAPE RECORDERS

Kunimaro Tanaka, Minoru Ozaki,
Toru Inoue, Tetsuya Yamaguchi

Mitsubishi Electric Corporation
Amagasaki, Hyogo, Japan

Abstract

In order to keep the reliability of PCM multi-channel tape recorders high, error-correcting codes have to be strong enough to assure correct functioning of the recorder even in poor conditions. This paper describes the problems when an RSC code (a combination of a Reed-Solomon code and CRC) is applied to the multi-channel tape recorder.

1. Introduction

Recent developments in digital audio engineering technology have been remarkable, and various items of equipment using it have been adopted for practical use in recording and broadcasting. The considerable merits of using PCM tape recorder for multiple (e.g., "ping-pong") dubbing with multi-channel tape recorders have lead to the publication of a number of different methods used by different companies in their tape recorders. One of the most important technical features of a multi-channel tape recorder is the tape format, which tends to determine the stability and the functions that the recorder provides.

Formats so far published can be divided broadly into one-track per channel,[1] two-track per channel,[2,3] and four-track per channel methods.[4] Each of these methods, however, uses an error-correcting code that operates independently for each channel. We shall call this the "separate" method. The method introduced in this article, and which we have called "semi-separate," on the other hand, features the generation of a parity code for multiple channels that is recorded on a special parity track. It uses an unusually powerful combination of Reed-Solomon and CRC codes to give an RSC code for error correction.

2. Recording Format

2-1 The semi-separate method

It is well known that errors in magnetic recording usually occur in bursts of several hundreds of bits along a single track, with the errors occurring independently in the different tracks. In order to cope with this kind of error it is common to apply interleaving over a considerable length of a given track, so that the influence of the burst at playback is dispersed over a much wider area. In contrast with this method, the semi-separate method was devised to cope with the following two additional factors:

(1) It is our experience that the common causes of errors in digital multi-track magnetic recordings, which

are responsible for degradation of recording quality, do not generally affect overall quality, but rather selectively degrade a few, particular tracks. Because of the high degree of bit error rate in these comparatively few tracks, the interleaving technique mentioned above has little effect.

(2) The greater the length of the code word, the greater is the code efficiency.

Fig. 1 shows the code block for the semi-separate method, and (a) is the structure of the sub-block. There are two parity tracks for the eight audio channels. The parity is determined by a Reed-Solomon code over $GF(2^4)$. In the Fig., sync is the frame synchronization, and (n-m) indicates the m-th sample of the n-th channel. The number of tracks required is therefore 30 for 24 audio channels, and 40 for 32 audio channels. Fig. 1b shows how the main block consists of 16 sub-blocks. The first sub-block carries the main sync., which is separately coded.

2-2 The RSC code

A rectangular code like that shown in Fig. 1 has conventionally been either a products code[5] or a concatenated code,[6] but neither of these could be claimed to be ideal for correcting errors in a multi-track magnetic recording, and so we have developed an expanded code (codeX).[7] The new code, which we have called the RSC code, consists of a combination of a Reed-Solomon and CRC code. Turning first to the Reed-Solomon code, the primitive polynomial over $GF(2^4)$ is[8]

$$P(x) = x^4 + x + 1 \tag{1}$$

and the root, α, of $P(x) = 0$ is the primitive element of $GF(2^4)$. A single code word consists of the 4-bit elements for the eight tracks that are aligned vertically after the frame sync and for the two parity tracks. The parity is recorded on the parity track. If we let the information on the i-th track be a_i, a comparison with α_k from Table 1 gives a_9 and a_{10} as

$$a_9 = \sum_{i=1}^{8} a_i \alpha_i \tag{2}$$

$$a_{10} = \sum_{i=1}^{9} a_i \tag{3}$$

The method of calculation of these equations is given in the appendix. The relationship is shown in Fig. 2, where the four bits for the first channel (0010) correspond with a_1, and the four bits for the second channel (1011) correspond with a_2, in that order, giving a_9 as (0101) and a_{10} as (0100).

Again, as shown in Fig. 1, at the very end of each sub-block there is a 16-bit CRCC to detect the tracks containing errors in the sub-block. The generator polynomial for the CRCC is

$$G(x) = x^{16} + x^{12} + x^5 + 1 \tag{4}$$

The information on error tracks from this CRCC and the information from the Reed-Solomon code are combined to give error correction for up to two channels within any one sub-block and error detection for three or more tracks.

3. Punch-in and Punch-out

3-1 Recording method

The fact that with the semi-separate method the individual channels are not completely separated from each other means that different considerations apply to punch-in and punch out from those for the separate method. Fig. 3 shows the tape format when punch-in, punch-out has been performed on the 2nd, 3rd and 4th channels. In the Fig., the hatched portions correspond to the portions that have been newly recorded after punch-in and punch-out. When portion A is re-recorded, portion C of the parity track must also be re-recorded. Portion C is determined from Eq. (2) and (3) for Portions A and B.

The structure adopted to achieve this recording format is shown in Fig. 4. The Fig. illustrates operation for any given channel. As shown in the Fig., a sync. playback head precedes the recording head. For overdubbing, the PCM signal played back by the sync. playback head has its errors corrected in the playback circuit, and the performer listens to this playback while performing. The inputs to channels 2, 3, and 4 are converted to a PCM signal by the A/D converter. This PCM signal is combined with the PCM playback signal from channels 1, 5, 6, 7 and 8, to derive the parity signal for portion C. Only portions A and C are subjected to over-write and re-recorded.

Because the blocks of Fig. 1, as will be explained later, are all interleaved, no noise is generated at punch-in and punch-out, so that signals join smoothly and continuously.

3-2 Reconstruction of played back blocks

Fig. 5 shows channels 1 and 2 after over-dubbing. In the Fig., 1/1 indicates the frame for the first channel of the first sub-block, and 1/2 indicates the frame for the second channel of the first sub-block. If error correction is required for these two frames, they must be combined correctly. If a wrong combination, such as that of 2/1 with 1/2, is made, the correction will generate errors and give rise to noise. Such mis-corrections arise when the relative positions of the recordings for channel 1 and channel 2 vary within frames, as caused by the following factors.

(1) Tape speed: caused by variations in capstan diameter or by slippage between the capstan and the tape.
(2) Head position: misalignment.
(3) Tape stretch: caused by variations in tape tension or changes in ambient temperature.
(4) Time-base errors

With the separate method, as explained in Section 2, in order to cope with the burst errors in the magnetic recording, interleaving is performed along the longitudinal direction of the tape, so that at punch-in and punch-out the problems arise along the length of the tape. With the semi-separate method, on the other hand, they arise across the width of the tape.

However, if blocks are reconstructed in units of the main block shown in Fig. 1b, these problems are solved. If for instance, as in Fig. 5, one main block is formed of 16 sub-blocks, then even if there is difficulty in pairing 2/1 and 2/2, it is possible to combine the set (1/1.........16/1) with (1/2.........16/2) quite unambiguously.

4. Error Correction with the Semi-Separate Method

4-1 Random error and short burst error

4-1-1 Assessment based on numerical calculation

If we let M be the number of bits in one sub-frame, then one sub-block will have $M \times 10$ bits. If, further, U_B is the number of tracks on the tape that actually have errors within a given block, S_B the number of erasures through detection with the CRCC, and e_B the number of errors undetected by the CRC, then decoding by the soft decision method operates as shown in Table 2.

Error detecting performance of CRCC is as following

(1) Two-bit random errors and odd-numbered bit errors are detected with a probability of unity.

(2) Burst errors with a length of up to 16 bits are detected with a probability of unity.

(3) Burst errors with a length of 17 bits are detected with a probability of $1 - 2^{-15}$.

(4) Burst errors with a length of more than 17 bits are detected with a probability of $1 - 2^{-16}$.

Accordingly, the probability P_m of failing to detect an error in a single track is given by the following equation:

$$P_m = 2^{-16} \sum_{b=18}^{M} \sum_{i=2}^{b} P_D(i,b,M) + 2^{-15} \sum_{i=2}^{17} P_D(i,17,M)$$

$$= 2^{-16} \left\{ \sum_{b=17}^{M} \sum_{i=2}^{b} P_D(i,b,M) + \sum_{i=2}^{17} P_D(i,17,M) \right\} \qquad (5)$$

where $P_D(l,m,n)$ is the probability of there being l error bits from a single concentrated burst error of length m bits within a length of n bits $(2 \leq l \leq m \leq n)$.

The probability P_d of detecting an error within a single track is given by the following equation:

$$P_d = 1 - P_c - P_m \qquad (6)$$

where P_c is the probability of there being no errors within n bits, and can be derived as $P(0^n)$ in reference (9).

4-1-2 RSC error concealment and mis-correction

For 4 bit x 10 track (one code word) mis-correction or error concealment, the probability of occurrence can be derived from the probability of detecting an error in a single track, P_d, and the probability of failing to detect an error, P_m, as follows:

(1) The probability of correct decoding $P_{c(RSC)}$ is

$$P_{c(RSC)} = P_c^{10} + 10.P_m.P_c^{9} + {}_{10}C_2\, P_d^{2} P_c^{8} \qquad (7)$$

(2) The probability of error concealment $P_{I(RSC)}$ is

$$P_{I(RSC)} = 10.P_d.9.P_m.P_c^{8} + \sum_{i=3}^{10} {}_{10}C_i\, P_d^{i}.P_c^{10-i} + \sum_{i=2}^{10} {}_{10}C_i\, P_m^{i}\, P_c^{10-i} + \sum_{i=3}^{10} {}_{10}C_i P_d^{i}\{(P_m + P_c)^{10-i} - P_c^{10-i}\} \qquad (8)$$

(3) The probability of mis-correction $P_{E(RSC)}$ is

$$P_{E(RSC)} = 10.P_d\{\sum_{i=2}^{9} {}_9C_i P_m^{i}.P_c^{9-i}\} + {}_{10}C_2.P_d^{2} \sum_{i=1}^{8} {}_8C_i .P_m^{i}.P_c^{9-i} \qquad (9)$$

We assumed a Gilbert model for error generation.[9]

As shown in Fig. 1, the CRCC is located at the very end of each frame, so that increasing the number of samples within each frame gives a desirable increase in coding efficiency. On the other hand, increasing the length causes a corresponding increase in the probability of numerous errors arising within a single block.

Fig. 6, 7 and 8 show the result of examining the effect on mean time between error concealment and mean time between mis-correction of changes in the number of bits, M, within one frame, in the average probability of bit errors, and in the average burst length. From the figures, it appears that coding efficiency is at a maximum when the frame length and the average burst length are approximately the same. Average burst lengths are typically of the order of 10^2. Consideration of limitations associated with the circuit configuration led us to adopt a frame length of 224 bits. Fig. 9 shows the error correcting capability of the format for this frame length.

With an average bit error rate of 10^{-4} and an average burst length of 200 bits, the mean time between error concealment is 2.5 hours and the mean time between mis-corrections is 62.2 years, both representing fully adequate error correcting capability.

4-2 Long burst error

To cope with long burst errors, we interleaved the signal. Fig. 10 shows the form of interleaving adopted. Fig. 10a shows a main block in which the odd number samples have been allocated to a preceding sub-block and the even number samples to the subsequent sub-block. Fig. 10b shows an example where the even number samples have been allocated to a block K main blocks away, and Fig. 10c shows a further development of (b) in which the parity code, rather than being arranged in four-bit elements across the width of the tape, is staggered diagonally across the tape, forming the parity code by collating the diagonally related frames. Fig. 11 shows the error control capability for these forms of interleaving when the sync. playback head and the recording head are separate by 3cm.

When one track fails, correction will be performed even if dropouts occur on another track, ensuring stable operation (as shown in Fig. 11a). Fig. 11b shows the size of tape defect that can be corrected and Fig. 11c the size of tape defect for which error concealment is possible. This figure illustrates the contribution made to stable tape recorder operation by the semi-separate method.

5. Conclusion

We have introduced here a tape format for multi-channel PCM tape recorders featuring a semi-separate method in which a long code word is distributed across the width of the tape. Not only does this give the high error correcting efficiency mentioned above, it also is effective for punch-in and punch-out. Tape-cut splicing and electronic editing are different methods of editing used with PCM tape recorders. Our method enables signals to be processed in a way that corresponds to punch-in and punch-out with smooth signal joins.

The functions and performance of PCM multi-channel tape recorders offer considerable advantages over conventional analogue methods, but they have suffered the disadvantage of high price. However, the speed with which semiconductor device prices are dropping, and the excellent error code efficiency of the method introduced here, offer good prospects of the rapid introduction of PCM multi-channel tape recorders.

Reference

(1) F.A. Bellis, M.R. Brookhart, "An error correcting system for a multichannel digital audio recorder", AES preprint 1298 (M-2) Nov. 1977

(2) Yoshikazu Tsuchiya et al., "A 24 channel stationary-head digital audio recorder" AES preprint 1412 (F-2) Nov. 1978

(3) Edwin W. Engberg,"A digital audio recorder format for professional applications" AES preprint 1413 (F-2) Nov. 1978

(4) H. Matsushima,"A new digital audio recorder for professional applications" AES preprint 1447 (G-7) March 1979

(5) P. Elias, "Error-free coding " IEEE Trans. Information theory, IT-4, pp29 - 37, 1954

(6) D.G. Forney, "Concatenated codes" MIT press, Cambridge Massachusetts, 1966

(7) T. Inoue et al., "A new class of burst error correcting code and it's application to PCM tape recording systems NTC'78 vol. II p20.6.1 - p20.6.5 1978

(8) Shu Lin, "An introduction to error-correcting code" Prentice-Hall

(9) E.N. Gibbert, "Capacity of a burst-noise channel" B.S.T.J. 39 pp1253 1265 Sep., 1960

6. Appendix

The calculation of equations (2) and (3) is not based on the ordinary rules of calculation, but on the rule of modulo 2 over 2^4 Galois field. An explanation of this method of calculation follows.

Addition: the result, z, of adding the number y to the number x ($z = x + y$) is shown in Table A for various values of x and y. For instance, if x = (0101) and y = (0011), Table 1 indicates that $x = \alpha^8$, and $y = \alpha^4$, so that $z = \alpha^8 + \alpha^4$, which from Table A gives $z = \alpha^5$. From Table 1 this gives z = (0110).

Multiplication: In the same way, the result, z, of multiplying the number x by the number y ($z = x.y$) is shown in Table B. For instance, if x = (1111) and y = (0110), Table 1 gives $x = \alpha^{12}$ and $y = \alpha^5$, so that from Table B, $z = \alpha^{12}.\alpha^5$ or α^2. From Table 1, again, this gives z = (0100).

For the calculation of equation (2), if we assume that the 4-bit pattern from track 1 to track 8 is (0010), (1011), (0100), (0001), (1001), (1110), (1000), (0010), then from Table 1, $a_1 = \alpha$, $a_2 = \alpha^7$, $a_3 = \alpha^2$, $a_4 = 1$, $a_5 = \alpha^{14}$, $a_6 = \alpha^{11}$, $a_7 = \alpha^3$, $a_8 = \alpha$. Again, if α_1 to α_8 are (1111), (1110), (1101), (1100), (1011), (1010), (1001), (1000), then $\alpha_1 = \alpha^{12}$, $\alpha_2 = \alpha^{11}$, $\alpha_3 = \alpha^{13}$, $\alpha_4 = \alpha^6$, $\alpha_5 = \alpha^7$, $\alpha_6 = \alpha^9$, $\alpha_7 = \alpha^{14}$ $\alpha_8 = \alpha^3$. The 4-bit pattern of the 9th track, a_9, will be

$$a_9 = \alpha.\alpha^{12} + \alpha^7.\alpha^{11} + \alpha^2.\alpha^{13} + 1.\alpha^6 + \alpha^{14}.\alpha^7 + \alpha^{11}.\alpha^9 + \alpha^3.\alpha^{14} + \alpha.\alpha^3$$

Then, calculating each term from Table B,

$$a_9 = \alpha^{13} + \alpha^3 + 1 + \alpha^6 + \alpha^6 + \alpha^5 + \alpha^2 + \alpha^4.$$

Further, by appropriate substitution in Table A, we obtain

$$a_9 = \alpha^8$$

In other words, the error correction 4-bit pattern on the 9th track is (0101). Finally we calculate the 4-bit pattern for the 10th track from equation (3). This pattern, a_{10}, is given by

$$a_{10} = a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + a_9$$

$$= \alpha + \alpha^7 + \alpha^2 + 1 + \alpha^{14} + \alpha^{11} + \alpha^3 + \alpha + \alpha^8$$

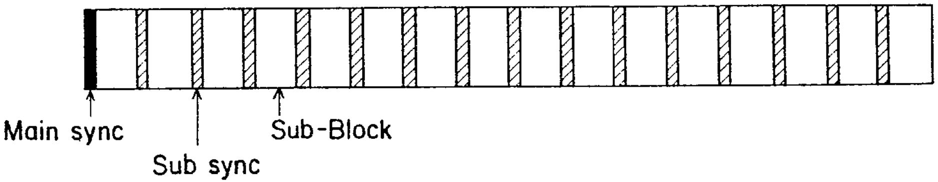
so that from Table A,

$$a_{10} = \alpha^2$$

In other words, the error correction 4-bit pattern on the 10th track is (0100) from Table 1. The 4-bit x 10-track code unit is as shown in Fig. 2.

		Sample ↓												
Channel 1	SYNC	1-1	1-2	1-3	1-4	1-5	1-6	1-7	1-8	1-9	1-10	1-11	1-12	CRC
Channel 2	SYNC	2-1	2-2	2-3	2-4									CRC
Channel 3	SYNC	3-1	3-2											CRC
Channel 4	SYNC	4-1											4-12	CRC
Channel 5	SYNC	5-1											5-12	CRC
Channel 6	SYNC											6-11	6-12	CRC
Channel 7	SYNC										7-10	7-11	7-12	CRC
Channel 8	SYNC								8-8	8-9	8-10	8-11	8-12	CRC
Parity 1	SYNC	$\sum a_i \alpha_i$												CRC
Parity 2	SYNC	$\sum a_i$												CRC

(a) Construction of a Sub-Block.



(b) Construction of a Main-Block

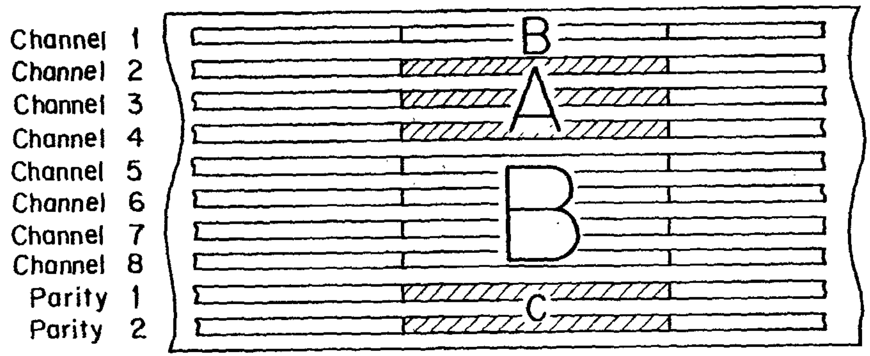
Fig. 1 Construction of a Block

Element of GF (2^4)	Symbol
0	0 0 0 0
1	0 0 0 1
α	0 0 1 0
α^2	0 1 0 0
α^3	1 0 0 0
α^4	0 0 1 1
α^5	0 1 1 0
α^6	1 1 0 0
α^7	1 0 1 1
α^8	0 1 0 1
α^9	1 0 1 0
α^{10}	0 1 1 1
α^{11}	1 1 1 0
α^{12}	1 1 1 1
α^{13}	1 1 0 1
α^{14}	1 0 0 1

Table 1 Correspondence between element and Symbol

0	0	1	0	a_1
1	0	1	1	a_2
0	1	0	0	a_3
0	0	0	1	a_4
1	0	0	1	a_5
1	1	1	0	a_6
1	0	0	0	a_7
0	0	1	0	a_8
0	1	0	1	$a_9 = \sum_{i=1}^{8} a_i \alpha_i$
0	1	0	0	$a_{10} = \sum_{i=1}^{9} a_i$

Fig. 2 Code word of Reed solomon code



Hatched potions correspoud to punch-in.
(B) represents the portion not punched-in.

Fig. 3 Over dubbing

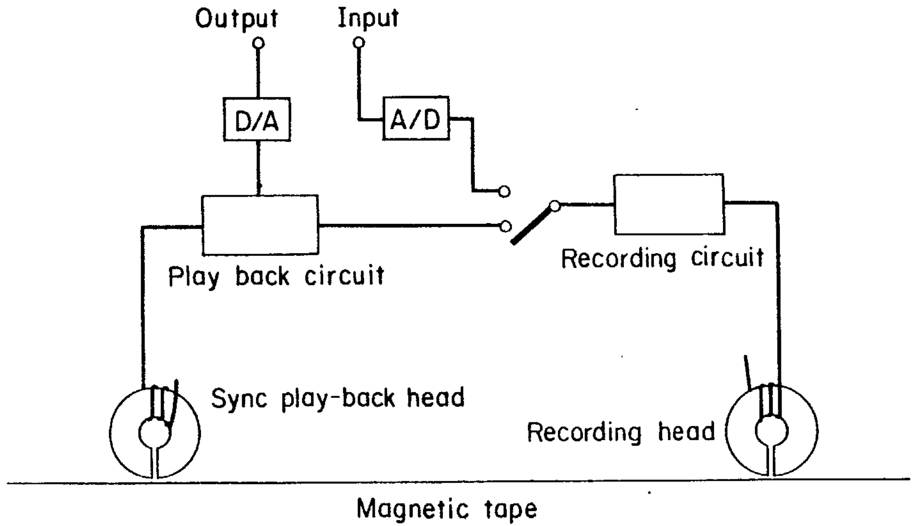


Fig. 4 Circuit diagram for over dubbing

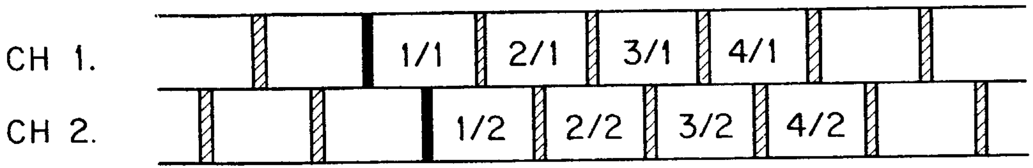


Fig. 5 Recorded pattern of channel 1 and channel 2 after over dubbing.

U_B	S_B	e_B	Results of decoding procedure
0	0	0	Original sound
1	1	0	Original sound
1	0	1	Original sound
2	2	0	Original sound
2	1	1	Compensated sound
2	0	2	Compensated sound
3	3	0	Compensated sound
3	no greater than 2	$U_B - S_B$	Miscorrected sound
⋮	⋮	⋮	⋮
U_B	U_B	⋮	Compensated sound
U_B	no greater than $U_B - 1$	$U_B - S_B$	Miscorrected sound
⋮	⋮	⋮	⋮

Table 2

Results of decoding procedure for codeword of RSC code corresponding to values of S_B and e_B

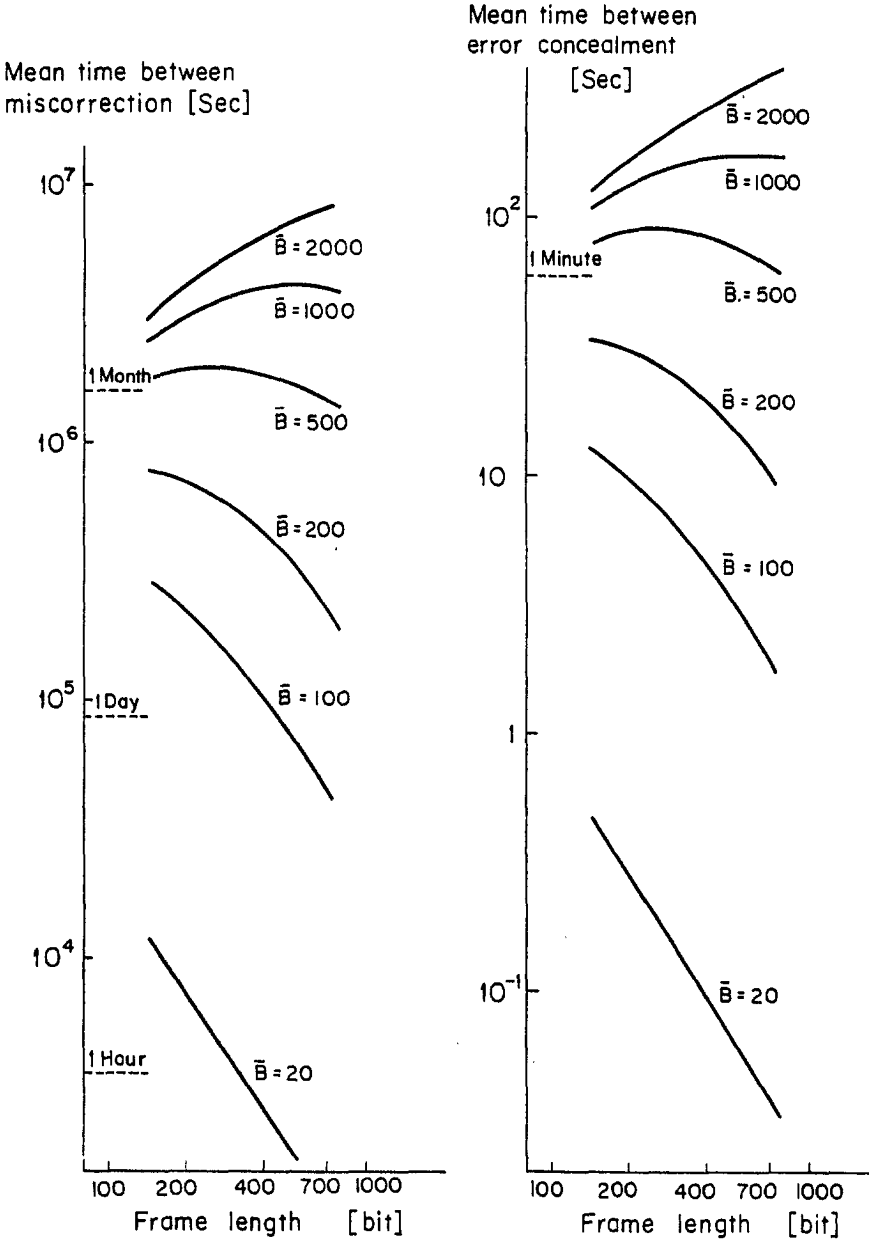


Fig. 6 Frame length vs error correcting capability

(Averag bit error rate 10^{-3})

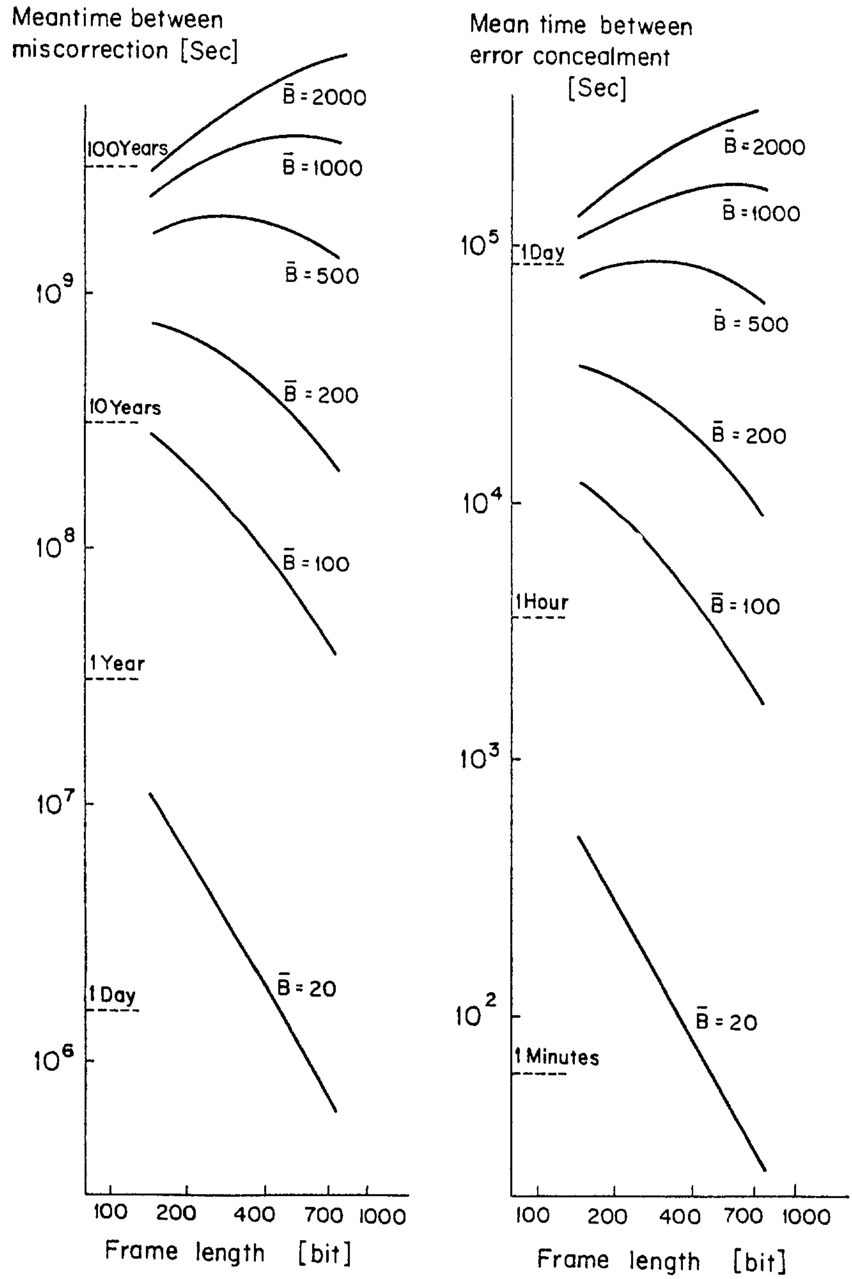


Fig. 7 Frame length vs error correcting capability
(Average bit error rate 10^{-4})

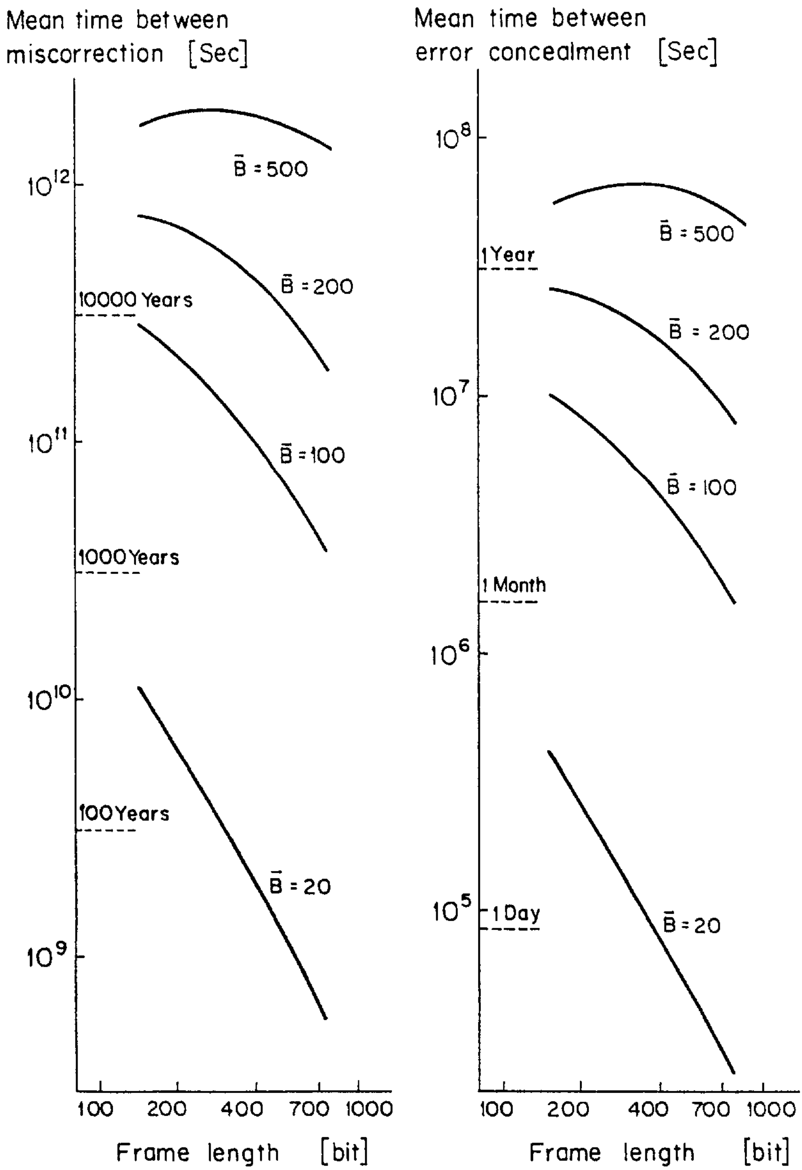


Fig. 8 Frame length vs error correcting capability

(Averag bit error rate 10^{-5})

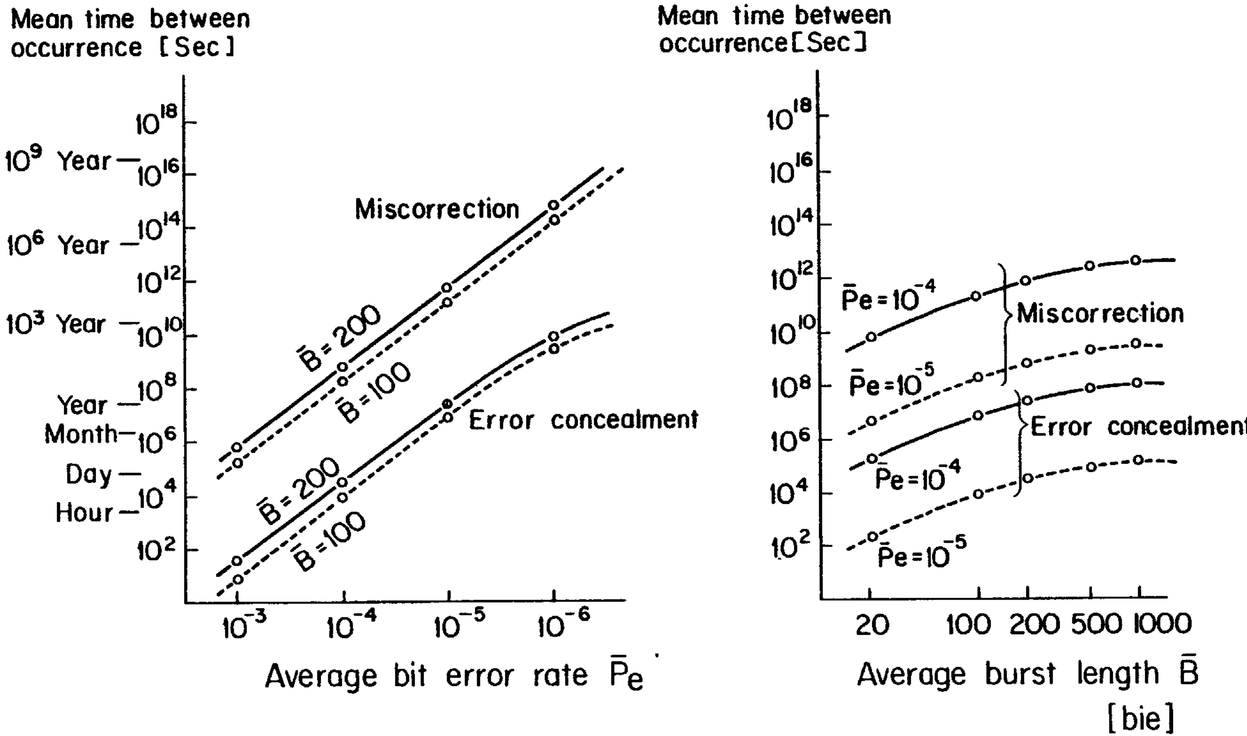
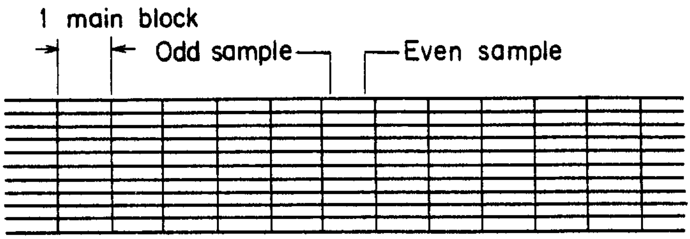
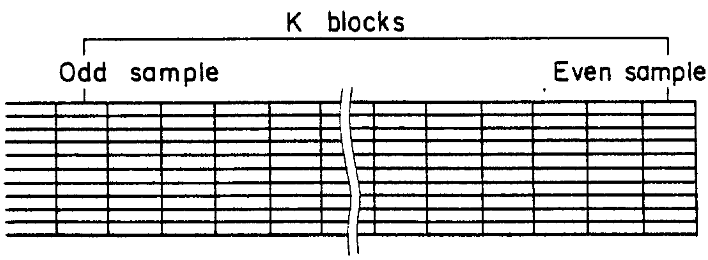


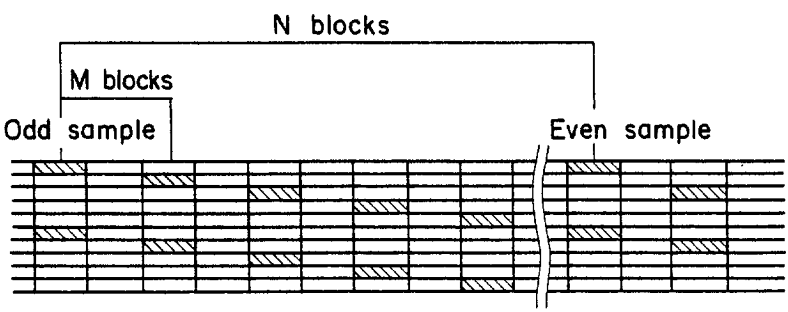
Fig. 9 Error correcting capability of the format



(a) Interleave within a block



(b) Interleave over a few blocks



(c) Diagonal interleave

Fig 10 Interleave of recording information

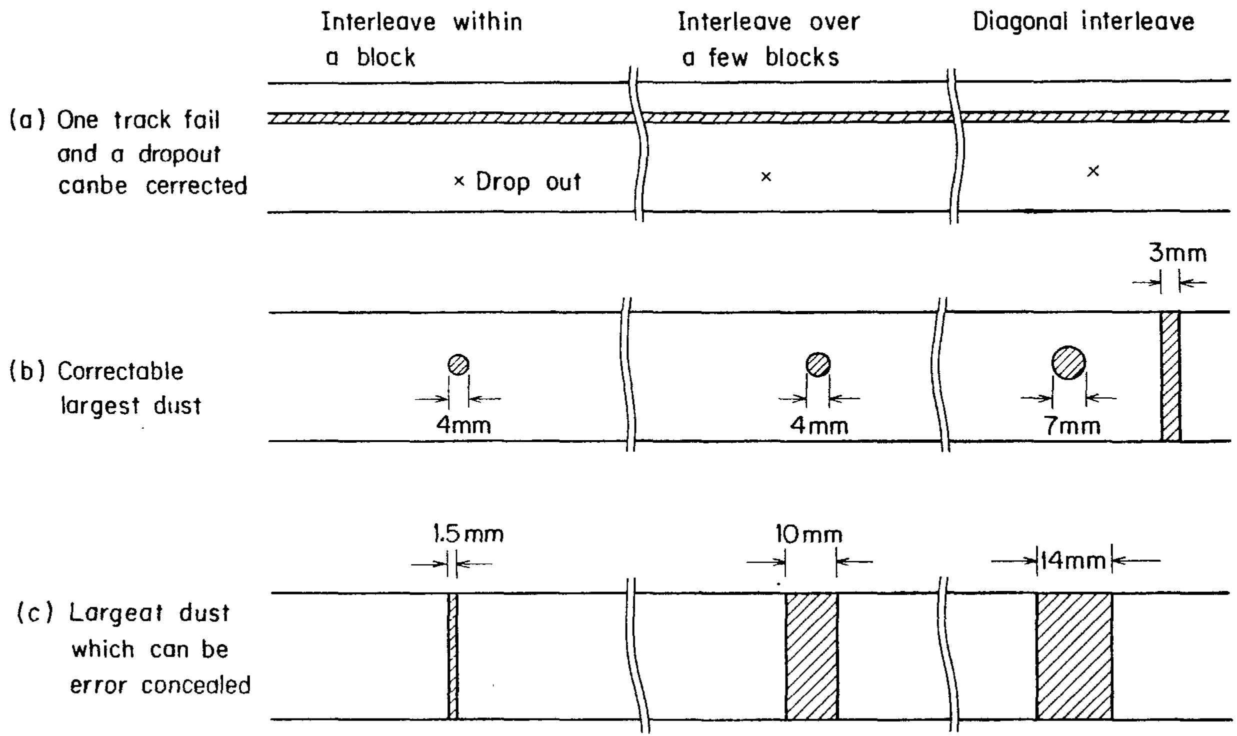


Fig. 11 Error correction of large dust

$z = x + y$

$x \backslash y$	0	1	α	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
0	0	1	α	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
1	1	0	α^4	α^8	α^{14}	α^1	α^{10}	α^{13}	α^9	α^2	α^7	α^5	α^{12}	α^{11}	α^6	α^3
α	α	α^4	0	α^5	α^9	1	α^2	α^{11}	α^{14}	α^{10}	α^3	α^8	α^6	α^{13}	α^{12}	α^{17}
α^2	α^2	α^8	α^5	α	α^6	α^{10}	α	α^3	α^{12}	1	α^{11}	α^4	α^9	α^7	α^{14}	α^{13}
α^3	α^3	α^{14}	α^8	α^6	0	α^7	α^{11}	α^2	α^4	α^{13}	α	α^{12}	α^5	α^{10}	α^8	1
α^4	α^4	α^1	1	α^{10}	α^7	0	α^8	α^{12}	α^3	α^5	α^{14}	α^2	α^{13}	α^6	α^{11}	α^9
α^5	α^5	α^{10}	α^2	α	α^{11}	α^8	0	α^9	α^{13}	α^{14}	α^6	1	α^3	α^{14}	α^7	α^{12}
α^6	α^6	α^{13}	α^{11}	α^3	α^2	α^{12}	α^9	0	α^{10}	α^{14}	α^5	α^7	α	α^4	1	α^8
α^7	α^7	α^9	α^{14}	α^{12}	α^4	α^3	α^{13}	α^{10}	0	α^{11}	1	α^6	α^8	α^2	α^5	α
α^8	α^8	α^2	α^{10}	1	α^{13}	α^5	α^4	α^{14}	α^{11}	0	α^{12}	α	α^7	α^9	α^{14}	α^{13}
α^9	α^9	α^7	α^3	α^{11}	α	α^{14}	α^6	α^5	1	α^{12}	0	α^{13}	α^2	α^8	α^{10}	α^4
α^{10}	α^{10}	α^5	α^8	α^4	α^{12}	α^2	1	α^7	α^6	α	α^{13}	0	α^{14}	α^3	α^9	α^{11}
α^{11}	α^{11}	α^{12}	α^6	α^9	α^5	α^{13}	α^3	α	α^8	α^7	α^2	α^{14}	0	1	α^4	α^{10}
α^{12}	α^{12}	α^{11}	α^{13}	α^7	α^{10}	α^6	α^{14}	α^4	α^2	α^9	α^8	α^3	1	0	α	α^5
α^{13}	α^{13}	α^6	α^{12}	α^{14}	α^8	α^{11}	α^7	1	α^5	α^{14}	α^{10}	α^9	α^4	α	0	α^2
α^{14}	α^{14}	α^3	α^7	α^{13}	1	α^9	α^{12}	α^8	α	α^{13}	α^4	α^{11}	α^{10}	α^5	α^2	0

Table A Defimition of addition

$z = x \cdot y$

$x \backslash y$	0	1	α	α^{2}	α^{3}	α^{4}	α^{5}	α^{6}	α^{7}	α^{8}	α^{9}	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	α	α^{2}	α^{3}	α^{4}	α^{5}	α^{6}	α^{7}	α^{8}	α^{9}	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
α	0	α	α^{2}	α^{3}	α^{4}	α^{5}	α^{6}	α^{7}	α^{8}	α^{9}	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1
α^{2}	0	α^{2}	α^{3}	α^{4}	α^{5}	α^{6}	α^{7}	α^{8}	α^{9}	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α
α^{3}	0	α^{3}	α^{4}	α^{5}	α^{6}	α^{7}	α^{8}	α^{9}	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^{2}
α^{4}	0	α^{4}	α^{5}	α^{6}	α^{7}	α^{8}	α^{9}	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^{2}	α^{3}
α^{5}	0	α^{5}	α^{6}	α^{7}	α^{8}	α^{9}	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^{2}	α^{3}	α^{4}
α^{6}	0	α^{6}	α^{7}	α^{8}	α^{9}	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^{2}	α^{3}	α^{4}	α^{5}
α^{7}	0	α^{7}	α^{8}	α^{9}	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^{2}	α^{3}	α^{4}	α^{5}	α^{6}
α^{8}	0	α^{8}	α^{9}	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^{2}	α^{3}	α^{4}	α^{5}	α^{6}	α^{7}
α^{9}	0	α^{9}	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^{2}	α^{3}	α^{4}	α^{5}	α^{6}	α^{7}	α^{8}
α^{10}	0	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^{2}	α^{3}	α^{4}	α^{5}	α^{6}	α^{7}	α^{8}	α^{9}
α^{11}	0	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^{2}	α^{3}	α^{4}	α^{5}	α^{6}	α^{7}	α^{8}	α^{9}	α^{10}
α^{12}	0	α^{12}	α^{13}	α^{14}	1	α	α^{2}	α^{3}	α^{4}	α^{5}	α^{6}	α^{7}	α^{8}	α^{9}	α^{10}	α^{11}
α^{13}	0	α^{13}	α^{14}	1	α	α^{2}	α^{3}	α^{4}	α^{5}	α^{6}	α^{7}	α^{8}	α^{9}	α^{10}	α^{11}	α^{12}
α^{14}	0	α^{14}	1	α	α^{2}	α^{3}	α^{4}	α^{5}	α^{6}	α^{7}	α^{8}	α^{9}	α^{10}	α^{11}	α^{12}	α^{13}

Table B Definition of multiplication