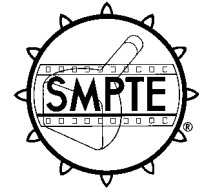


AVA Documents
Document Sharing Hub

SMPTE STANDARD

for Television Digital Component Recording — 19-mm Type D-1 — Helical Data and Control Records



Page 1 of 37 pages

1 Scope

This standard specifies the content, format, and recording method of the data blocks forming the helical records on the tape containing video, audio, and associated data in 19-mm type D-1 television digital component recording. In addition, clause 6 of this document specifies the content, format, and recording method of the longitudinal record containing tracking information for the scanning head associated with the helical records. Track dimensions and locations are specified in ANSI/SMPTE 224M.

The standard applies to recorders operating in the 525-line television system with a frame frequency of 29.97 Hz nominal and in accord with ITU-R BT.601. One video channel and four independent audio channels are recorded. Audio channels operate in accord with ANSI S4.40 at a 48-kHz sampling frequency.

Figure 1 shows a block diagram of the processes involved in the recorder.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this

standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

ANSI S4.40-1992, Digital Audio Engineering — Serial Transmission Format for Two-Channel Linearly Represented Digital Audio Data

ANSI/SMPTE 224M-1996, Television Digital Component Recording — 19-mm Type D-1 — Tape Record

ITU-R BT. 601-5, Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-Screen 16:9 Aspect Ratios

3 Helical record content, format, synchronization, and recording method

3.1 Introduction

The helical track defined mechanically in ANSI/SMPTE 224M is recorded with the digital data from the video channel and the four audio channels. Data is arranged in six sectors per track as shown in figure 2. Two sectors are employed for video data and four sectors each containing data from one of the four audio channels. Details of sector assignment are shown in clauses 4 and 5 of this document. Each sector is divided into the following elements:

The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights.

By publication of this standard, no position is taken with respect to the validity of this claim or of any patent rights in connection therewith. The patent holder has, however, filed a statement of

willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. Details may be obtained from the publisher.

No representation or warranty is made or implied that this is the only license that may be required to avoid infringement in the use of this standard.

CAUTION NOTICE: This Standard may be revised or withdrawn at any time. The procedures of the Standard Developer require that action be taken to reaffirm, revise, or withdraw this standard no later than five years from the date of publication. Purchasers of standards may receive current information on all standards by calling or writing the Standard Developer.

Printed in USA.

- Preamble containing a clock run-up sequence, sync pattern, and identification pattern;
- Sync blocks containing sync pattern and an identification pattern followed by a fixed length data block with error control;
- Postamble containing channel sync pattern and an identification pattern.

Details of the elements are shown in figure 3. The space between sectors may be unrecorded or filled with the clock run-up sequence (CC)_H. This space is used to accommodate sector timing errors and to allow editing.

A portion of the guard-space at the beginning of the track may contain run-up sequence data pattern (CC)_H of a length up to 100 bytes.

3.2 Labelling convention for audio and video data

3.2.1 The least significant bit is written on the left and is the first recorded to tape.

3.2.2 The lowest numbered byte is shown at left/top and is the first encountered in the input data stream.

3.2.3 Byte values are expressed in hexadecimal notation unless otherwise noted.

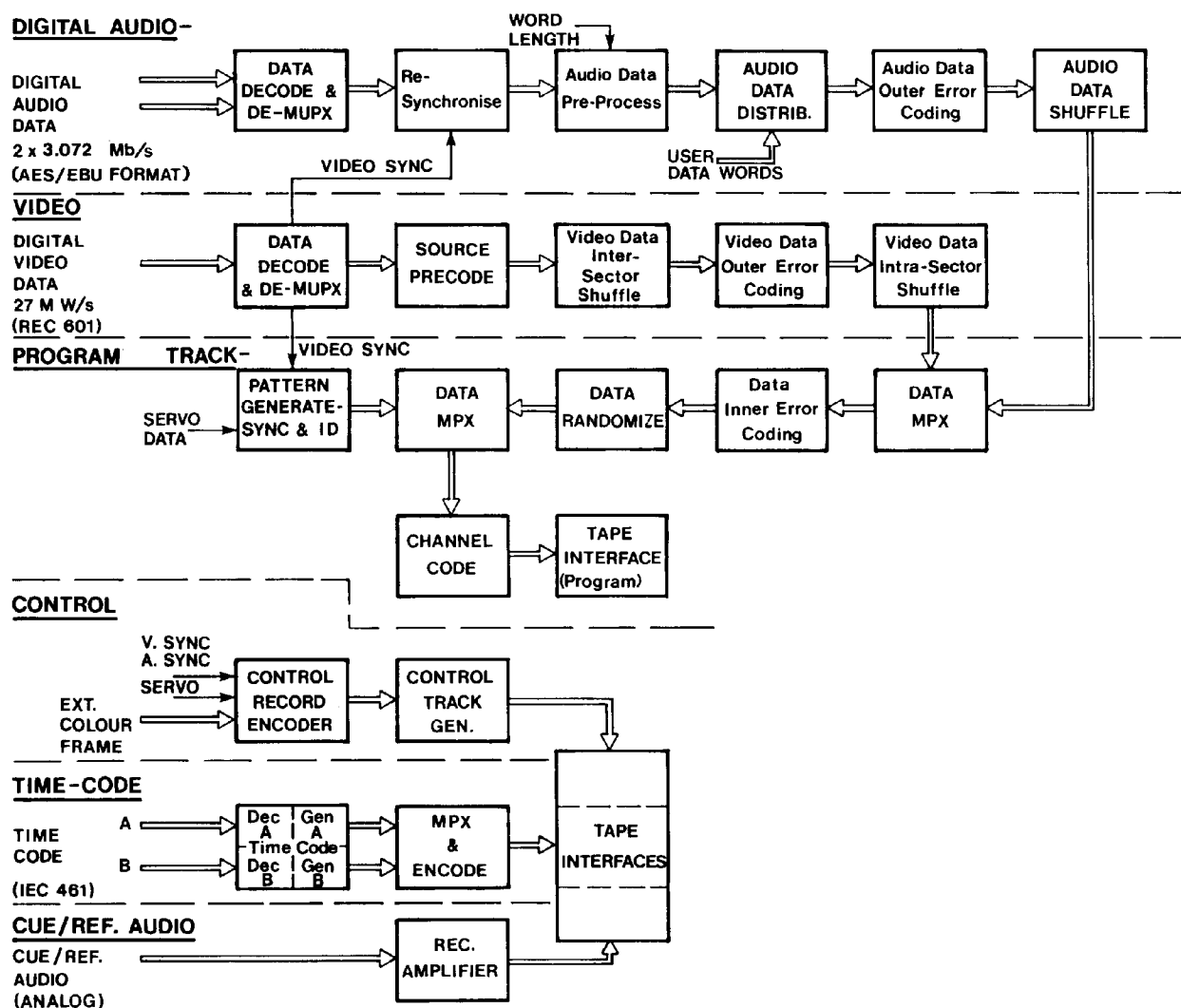
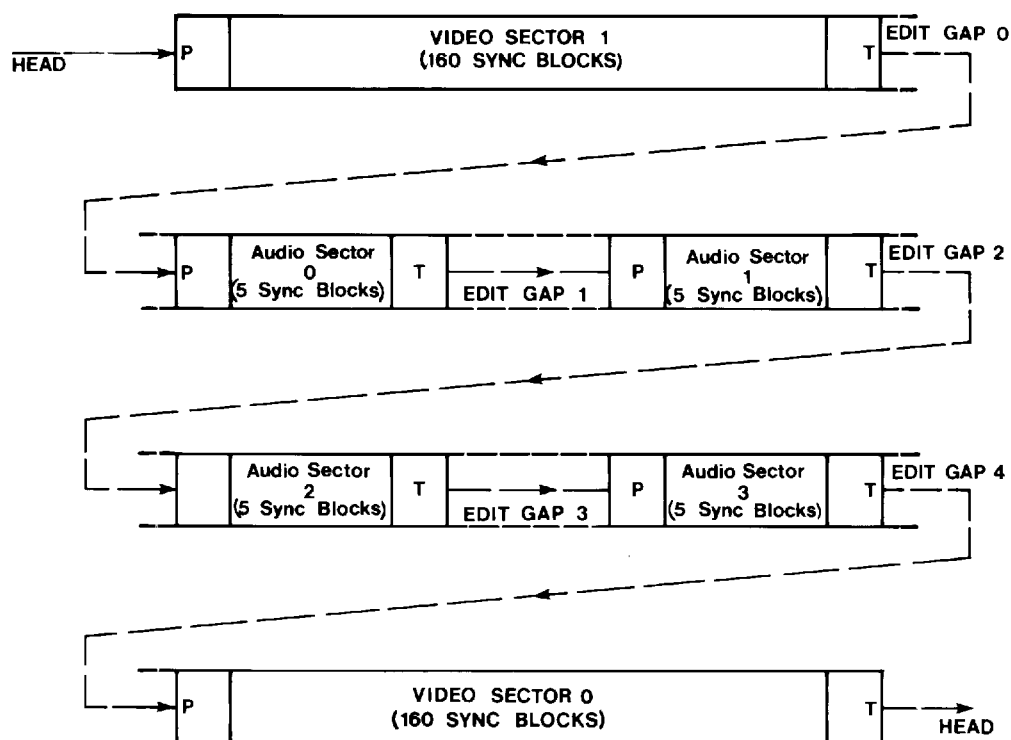


Figure 1 – Digital recorder — Record path processing



P = PRE-AMBLE (30 BYTES)

T = POST-AMBLE (6 BYTES)

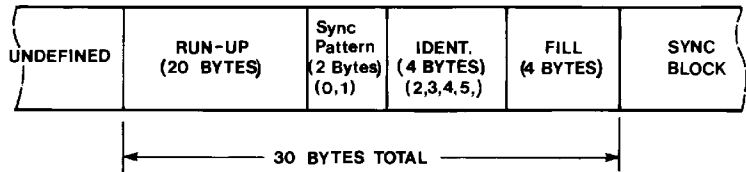
SYNC BLOCK (134 BYTES)

Figure 2 Sector arrangement on helical track

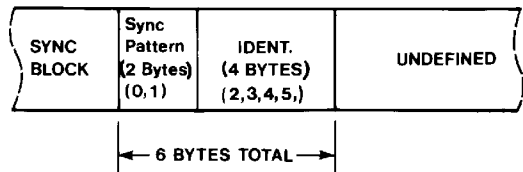
Table 1 -- Helical track sector length

Sector name	Length mm	Size	
		S. blocks	Bytes
V1	77.71	160	21476
A0	2.55	5	706
A1	2.55	5	706
A2	2.55	5	706
A3	2.55	5	706
V0	77.71	160	21476
Edit gap 0-4	5×0.84	—	5×232
Total	169.83 ¹⁾	—	46936
¹⁾ 169.83 is derived from specified value $\frac{170.0 \text{ mm}}{1.001}$			

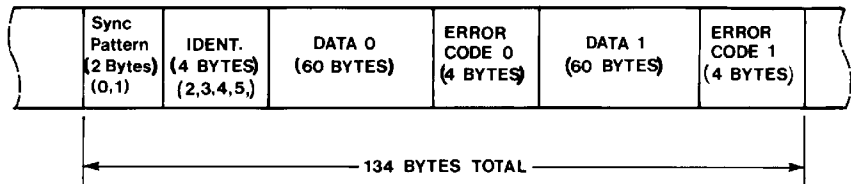
(a) **PREAMBLE**



(b) **POSTAMBLE**



(c) **SYNC BLOCK**



(d) **SECTOR**

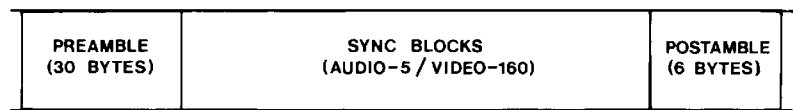


Figure 3 – Sector components

3.3 Sector details

3.3.1 Sync block

Details of the sync block are shown in figure 3(c). All sync blocks consist of 134 bytes consisting of sync pattern (2 bytes) and identification pattern (4 bytes including error coding) followed by 128 data bytes.

3.3.2 Sync pattern

(a) Length: 16 bits (2 bytes)

(b) Pattern: 30 F5 (in hexadecimal notation)

	LSB						MSB	
Byte 0	0	0	0	0	1	1	0	0
	LSB							MSB
Byte 1	1	0	1	0	1	1	1	1

(c) Protection: None

(d) Randomization: None

3.3.3 Identification pattern

(a) Length: 32 bits (4 bytes)

(b) Arrangement:

Byte 2 — sync block ID — see figure 4

Byte 3 — sync block ID — see figure 4

Byte 4 — segment ID — see figures 4 and 9

Byte 5 — sector ID — see figures 4 and 9

(c) Protection: 4 to 8 mapping as in table 2

(d) Randomization: None

The values of the sync block ID are shown in figure 4.

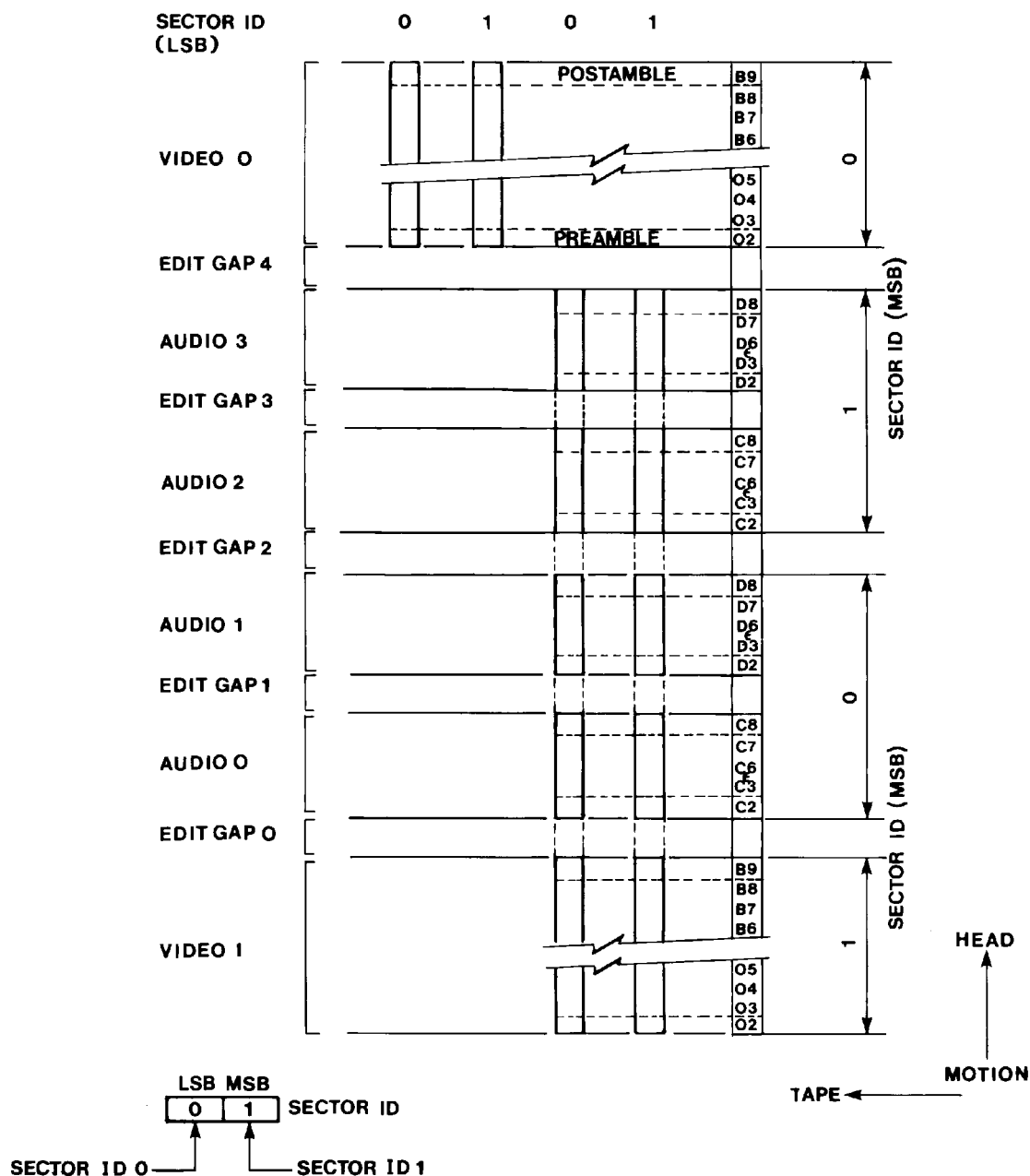


Figure 4 – Values of sync block identification and sector identification codes

Table 2 -- 4- to 8-bit mapping

Input	Output	Input	Output
0	1B	8	96
1	2E	9	A3
2	35	A	B8
3	47	B	CA
4	5C	C	D1
5	69	D	E4
6	72	E	Illegal
7	8D	F	
NOTE -- Values expressed are hexadecimal.			

3.3.3.1 Identification pattern — Byte 2

Mapped from word 0 (4 bits) where word 0 is the right-hand character identified in figure 4 (see table 2 and figure 5).

3.3.3.2 Identification pattern — Byte 3

Mapped from word 1 (4 bits) where word 1 is the left-hand character identified in figure 4 (see table 2 and figure 6).

3.3.3.3 Identification pattern — Byte 4

Mapped from word 2 (4 bits) by table 2 (see figure 7).

3.3.3.4 Identification pattern — Byte 5

Mapped from word 3 (4 bits) by table 2 (see figure 8).

Figure 5 -- Identification pattern — Byte 2

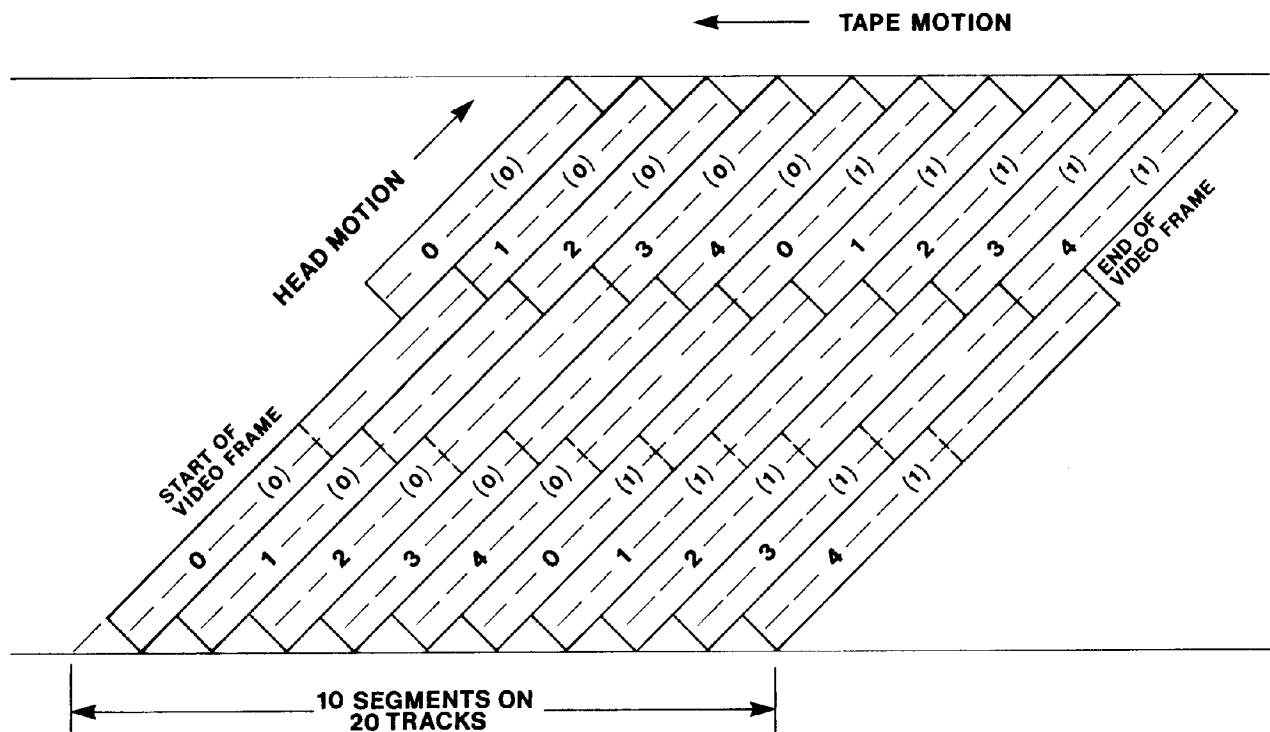
Figure 7 -- Identification pattern — Byte 4

Figure 6 -- Identification pattern — Byte 3

Figure 8 -- Identification pattern — Byte 5

Notes to figures 5, 6, 7, and 8

- 1 Sync block ID is an 8-bit word formed from two 4-bit words, each lying in the range 0 – D_H, uniquely identifying each sync block within one sector. Figure 4 specifies these values.
- 2 Field ID lies in the range 0 – 3 with the origin aligned with the field sequence identification pulse doublet mark (see clause 6). The values of field ID are shown in figure 9. These values must be continuous, even after editing operations.
- 3 Segment ID lies in the range 0 – 4 as shown in figure 9.



NOTES

- 1 Segment numbers lie in the range 0–4 (unbracketed).
- 2 Field numbers lie in the range 0–3 (bracketed).
- 3 Fields 0–1 shown; Fields 2–3 are similar.

Figure 9 – Segment and field numbers

3.3.4 Data field – Sync block

This block construction is used for all audio and video data and the associated error correction data.

(a) Length: 2 inner code blocks, each of 60 data bytes plus 4 inner error-code check bytes. (Outer error-code check bytes are considered as data.)

(b) Arrangement: See figure 3(c).

(c) Protection (inner code):

Type: Reed-Solomon

Galois field: GF(256)

Field generator polynomial: $x^8 + x^4 + x^3 + x^2 + x^0$ (x^i are place keeping variables in GF(2), the binary field).

Order of use: Left-most term is most significant, "oldest" in time computationally, and first written to tape.

Code generator polynomial (in GF(256)): $G(x) = (x + \alpha^0)(x + \alpha^1)(x + \alpha^2)(x + \alpha^3)$, where α^1 is given by 02H in GF(256).

Check characters: K_3, K_2, K_1, K_0 in $K_3x^3 + K_2x^2 + K_1x^1 + K_0x^0$ obtained as the remainder after dividing $x^4 \times D(x)$ by $G(x)$ where $D(x) = B_{59}x^{59} + B_{58}x^{58} + \dots + B_1x^1 + B_0x^0$.

Equation of full code: $B_{59}x^{63} + B_{58}x^{62} + \dots + B_0x^4 + K_3x^3 + \dots + K_0x^0$.

An example of three possible patterns is shown in table 3, where pattern 1 is the impulse function where the values in the check locations represent the expansion of the code generator polynomial.

Table 3 -- Sync block data field patterns

Symbol position	Data symbols D(x)							Check symbols					
	0	1	2	3	4	5	6	58	59	60	61	62	63
Pattern 1	00	00	00	00	00	00	00	00	01	0F	36	78	40
Pattern 2	00	01	02	03	04	05	06	3A	3B	85	24	A9	08
Pattern 3	CC	CC	CC	CC	CC	CC	CC	CC	CC	B6	D4	B6	D4
Symbol identify	B ₅₉	B ₅₈	B ₅₇	B ₅₆	B ₅₅	B ₅₄	B ₅₃	B ₁	B ₀	K ₃	K ₂	K ₁	K ₀

(d) Interleaving: Not used.

(e) Randomization: All data and error correction check characters are randomized before being recorded. (Sync, identification, and fill patterns are not randomized). The randomizing is equivalent to performing the EXOR operation between the serial data stream, and the serial stream generated by the polynomial function $x^8 + x^4 + x^3 + x^2 + x^0$ (in GF(2)). The first term is the most significant and the first to enter the division computation.

In order that successive sync blocks are randomized with different sequences, the polynomial generator noted above is pre-set to 80_H (see note) to read for byte 0 of the sync block locations having ID values as follows:

03, 08, 0D, 14, 19, 20, 25, 2A, 31, 36, 3B, 42, 47, 4C, 53, 58, 5D, 64, 69, 70, 75, 7A, 81, 86, 8B, 92, 97, 9C, A3, A8, AD, B4, C3, D3.

NOTE – This will generate a byte sequence beginning 80, 38, D2, 81, 49, etc. Although the sync and identification patterns are not randomized, the polynomial generator continues to cycle during this period.

3.3.5 Sector preamble

All sectors commence with the preamble sequence.

(a) Length: 30 bytes

(b) Arrangement: See figure 3(a)

Run-up: 20 bytes minimum of CC_H (for clock reference)

Sync pattern: 2 bytes (see 3.3.2)

Identification pattern: 4 bytes (see 3.3.3)

Fill: 4 bytes of CC_H

(c) Protection: None (ID data directly recorded)

(d) Randomization: None

(e) Interleaving: None

3.3.6 Sector postamble

All sectors terminate with the postamble sequence.

(a) Length: 6 bytes

(b) Arrangement: See figure 3(b)

Sync pattern: 2 bytes (see 3.3.2)

Identification pattern: 4 bytes (see 3.3.3)

(c) Protection: None

(d) Randomization: None

(e) Interleaving: None

3.4 Edit gaps

The space, of 232-byte (0.84 mm) nominal length, between sectors may be left unwritten. The area may be unwritten or filled with (CC)_H.

3.5 Channel code

The NRZ data stream shall be recorded directly without further coding.

3.6 Magnetization

During the time interval of a recorded data 1, the polarity of data flux shall be such that the north pole of the magnetic domain shall point in the direction of head motion. Similarly, during the time interval of a recorded data 0, the polarity of data flux shall be such as to cause the south pole of the magnetic domain to point in the direction of head motion. Magnetization shall bring the tape to saturation.

4 Video processing

4.1 Recorded data

Information received during the digital horizontal blanking interval is not recorded on tape. The appropriate blanking data are recreated for output during playback.

4.1.1 Recorded lines

The last 250 lines from each television field are recorded. These comprise lines 14 through 263 inclusive from field 1 and lines 276 through 525 inclusive from field 2. Lines 21 to 263 inclusive and 283 to 525 inclusive contain video data.

4.1.2 Digital active line

A total of 1440 bytes, are recorded: 720 luminance bytes and 360 bytes for each of the two color-difference components. These are taken from bytes 0 through 1439 following the 4-byte start of active video (SAV) timing reference signals.

4.1.3 Source precoding

The input video data stream is precoded by a one-for-one mapping of each source data byte as defined in

table 4. The inverse mapping for the regeneration of the original video source data bytes is defined in table 5. Data in lines 14-20 and 276-282 inclusive are not precoded.

4.2 Pixel labelling

There are 250 recorded lines per field, with 720 pixels per line. They can be considered as an array of 250 rows by 720 columns, in which each pixel is identified by a pair of integers (i, j), where i identifies the row and is numbered 0 to 249 from top to bottom, and j identifies the column and is numbered 0 to 719 from left to right. Columns with even j are associated with a luminance value Y_{ij} and two cosited color-difference values, CB_{ij} and CR_{ij} , where CB and CR designate scaled B-Y and R-Y components, respectively. The 4:2:2 video data sequence for line i is written as follows:

$$\begin{aligned} &CB_{i,0} \ Y_{i,0} \ CR_{i,0} \ Y_{i,1} \dots \\ &CB_{i,k} \ Y_{i,k} \ CR_{i,k} \ Y_{i,k+1} \dots \\ &CB_{i,718} \ Y_{i,718} \ CR_{i,718} \ Y_{i,719} \\ &0 \leq i \leq 249 \\ &0 \leq j \leq 719 \\ &\text{and } k = 2(\text{int}(j/2)) \end{aligned}$$

Table 4 – Source video mapping

Input		LS word (4 bits)															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
MS word (4 bits)	0	00	80	40	20	10	08	04	02	01	C0	A0	90	88	84	82	81
	1	60	50	48	44	42	41	30	28	24	22	21	18	14	12	11	0C
	2	0A	09	06	05	03	E0	D0	C8	C4	C2	C1	B0	A8	A4	A2	A1
	3	98	94	92	91	8C	8A	89	86	85	83	70	68	64	62	61	58
	4	54	52	51	4C	4A	49	46	45	43	38	34	32	31	2C	2A	29
	5	26	25	23	1C	1A	19	16	15	13	0E	0D	0B	07	F0	E8	E4
	6	E2	E1	D8	D4	D2	D1	CC	CA	C9	C6	C5	C3	B8	B4	B2	B1
	7	AC	AA	A9	A6	A5	A3	9C	9A	99	96	95	93	8E	8D	8B	87
	8	78	74	72	71	6C	6A	69	66	65	63	5C	5A	59	56	55	53
	9	4E	4D	4B	47	3C	3A	39	36	35	33	2E	2D	2B	27	1E	1D
	A	1B	17	0F	F8	F4	F2	F1	EC	EA	E9	E6	E5	E3	DC	DA	D9
	B	D6	D5	D3	CE	CD	CB	C7	BC	BA	B9	B6	B5	B3	AE	AD	AB
	C	A7	9E	9D	9B	97	8F	7C	7A	79	76	75	73	6E	6D	6B	67
	D	5E	5D	5B	57	4F	3E	3D	3B	37	2F	1F	FC	FA	F9	F6	F5
	E	F3	EE	ED	EB	E7	DE	DD	DB	D7	CF	BE	BD	BB	B7	AF	9F
	F	7E	7D	7B	77	6F	5F	3F	FE	FD	FB	F7	EF	DF	BF	7F	FF

Table 5 – Inverse video mapping

Input		LS word (4 bits)															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
MS word (4 bits)	0	00	08	07	24	06	23	22	5C	05	21	20	5B	1F	5A	59	A2
	1	04	1E	1D	58	1C	57	56	A1	1B	55	54	A0	53	9F	9E	DA
	2	03	1A	19	52	18	51	50	9D	17	4F	4E	9C	4D	9B	9A	D9
	3	16	4C	4B	99	4A	98	97	D8	49	96	95	D7	94	D6	D5	F6
	4	02	15	14	48	13	47	46	93	12	45	44	92	43	91	90	D4
	5	11	42	41	8F	40	8E	8D	D3	3F	8C	8B	D2	8A	D1	D0	F5
	6	10	3E	3D	89	3C	88	87	CF	3B	86	85	CE	84	CD	CC	F4
	7	3A	83	82	CB	81	CA	C9	F3	80	C8	C7	F2	C6	F1	F0	FE
	8	01	0F	0E	39	0D	38	37	7F	0C	36	35	7E	34	7D	7C	C5
	9	0B	33	32	7B	31	7A	79	C4	30	78	77	C3	76	C2	C1	EF
	A	0A	2F	2E	75	2D	74	73	C0	2C	72	71	BF	70	BE	BD	EE
	B	2B	6F	6E	BC	6D	BB	BA	ED	6C	B9	B8	EC	B7	EB	EA	FD
	C	09	2A	29	6B	28	6A	69	B6	27	68	67	B5	66	B4	B3	E9
	D	26	65	64	B2	63	B1	B0	EB	62	AF	AE	E7	AD	E6	E5	FC
	E	25	61	60	AC	5F	AB	AA	E4	5E	A9	A8	E3	A7	E2	E1	FB
	F	5D	A6	A5	E0	A4	DF	DE	FA	A3	DD	DC	F9	D8	F8	F7	FF

4.3 Intersector distribution

Consider the pixels in a field to be numbered according to 4.2.

Let m designate the number of a given line within a segment then

$$m = i \bmod 50$$

Let r designate the sector number within a segment, $0 \leq r \leq 3$.

The pixels within each segment are evenly distributed between the four corresponding sectors as shown in figure 9 and by the following equations:

(i) for the luminance (Y) component,

$$r_y = 2((f+g+j) \bmod 2) + \text{int}(((j+2(m \bmod 2)) \bmod 4)/2)$$

(ii) and for the color-difference components (CB and CR), $r_c = 2((f+g+\text{int}(j/2)) \bmod 2) + \text{int}(((\text{int}(j/2)+2(m \bmod 2)) \bmod 4)/2)$

when g designates the segment in which a given line i falls: $g = \text{int}(i/50)$

f = least significant part of the field ID for the 525 system only.

(The function $\text{int}(x)$ yields the integer part of (x) .)

This results in 180 luminance pixels and 90 pairs of color-difference pixels per line in each sector of a segment.

The distribution of pixels in each sector is further described in table 6.

4.4 Intrasector shuffling

The intrasector shuffling sequence during the record process shall be described in terms of two successive shuffling processes:

- An intraline shuffle which shuffles video and ancillary words within a single line prior to outer error coding;
- A sector array shuffle which shuffles data and error correction code words within the sector, prior to being written to tape.

The sector array dimensions are 32 rows by 600 columns. Each column corresponds to one outer code block and contains 30 video data bytes plus two outer correction check bytes. The sector array is further divided into 10 contiguous subarrays, each having dimensions of 32 rows by 60 columns. The 60 data bytes within a single subarray row correspond to one inner code block on tape.

4.4.1 Intraline shuffle

Let the horizontal pixel index, j , be normalized to the range (0 –179) following the intersector distribution described in 4.3.

For luminance component,
 $jy' = \text{int}(jy/4)$

For the color-difference components (CB and CR),
 $jc' = 2 \text{ int}(jc/8)$

where j' indicates a normalized index.

Then the sector data sequence for a given line contains 360 bytes as shown in table 7.

Table 6 – Intersector shuffling for odd and even lines

For $(f + g) \bmod 2 = 0$										
Even line numbers ($m \bmod 2 = 0$)	$j =$	0 1	2 3	4 5	6 7	8 9	10 11	12 13	14 15	16...
	$r_y =$	0 2	1 3	0 2	1 3	0 2	1 3	0 2	1 3	0
	$r_c =$	0	2	1	3	0	2	1	3	0
Odd line numbers ($m \bmod 2 = 1$)	$j =$	0 1	2 3	4 5	6 7	8 9	10 11	12 13	14 15	16...
	$r_y =$	1 3	0 2	1 3	0 2	1 3	0 2	1 3	0 2	1
	$r_c =$	1	3	0	2	1	3	0	2	1
For $(f + g) \bmod 2 = 1$										
Even line numbers ($m \bmod 2 = 0$)	$j =$	0 1	2 3	4 5	6 7	8 9	10 11	12 13	14 15	16...
	$r_y =$	2 0	3 1	2 0	3 1	2 0	3 1	2 0	3 1	2
	$r_c =$	2	0	3	1	2	0	3	1	2
Odd line numbers ($m \bmod 2 = 1$)	$j =$	0 1	2 3	4 5	6 7	8 9	10 11	12 13	14 15	16...
	$r_y =$	3 1	2 0	3 1	2 0	3 1	2 0	3 1	2 0	3
	$r_c =$	3	1	2	0	3	1	2	0	3

Table 7 – Sector data sequence

k:	0	1	2	3	4	5	6	7	356	357	358	359
byte:	CB ₀	Y ₀	CR ₀	Y ₁	CB ₂	Y ₂	CR ₂	Y ₃	CB ₁₇₈	Y ₁₇₈	CR ₁₇₈	Y ₁₇₉