UDC 621.394.625.15
Indexing Terms: Tape recorders, multitrack, magnetic, Data storage

High density data storage on audio compact cassette tape using a low-cost tape transport

T. DONNELLY, BSc, MSc*

Professor D. J. MAPPS, BEng, PhD*

and

R. WILSON, BTech, MSc†

Based on a paper presented at the IERE Conference on Video, Audio and Data Recording held at the University of Sussex in March 1986

* *Department of Electrical and Electronic Engineering, Plymouth Polytechnic, Drake Circus, Plymouth, Devon PL4 8AA*
† *Department of Electronic and Communication Engineering, Leeds Polytechnic, Calverley Street, Leeds LS1 3HE*

SUMMARY
A parallel encode/decode software algorithm has been designed for the recently developed ISS 2/3 code. In addition a low-cost solution has been applied to the problem of tape–head azimuth variation (tape skew). The above enables the recording and playback of direct digital data at 60 kb/s on normal audio compact cassette tape using an 8-track format.

The combination of a low-cost tape transport and a standard audio cassette tape produces a high degree of tape skew. This is overcome by encoding control pulses on the outer tracks which are used to introduce the same amount of data skew into the clock signals used to clock the data off the tracks. The self-clocking ISS 2/3 code facilitates clock regeneration which, together with simple de-skew circuitry and software encoding/decoding permits the realization of a low-cost bulk storage system. An increase in the data rate to 100 kb/s is possible through the use of equalization and by incorporating data onto the outer, de-skew tracks.

1 Introduction

New data recording codes are constantly being developed in order to realize the full potential offered by magnetic media for high density storage.[1] Desirable characteristics of such codes include the matching of the code's frequency spectrum to that of the recording channel[2] and the simple implementation of data encoding and recovery methods.

Table 1. Encoding table

Data	Code
00	101
01	100
10	001
11	010

A recent addition to the existing range of recording codes is the ISS 2/3 code.[3] This 2/3 rate code looks ahead over 4 bits and encodes 3 code bits for each group of 2 data bits (Table 1). The ISS 2/3 code permits a greater recording density when compared with existing codes such as MFM and 3PM.[3] Also, the code may be encoded/decoded using basic circuitry.[3] A 'look-up' method is employed with the data/code forming the input address to a ROM which contains the required code/data. Although viable for high frequency serial operation this technique would prove costly and cumbersome if expanded into parallel form. At the lower data rates found in fixed-head tape systems a microprocessor may be employed as an encoder/decoder. This software approach not only reduces the complexity of data encoding/decoding but also permits parallel implementation, thus increasing further the potential of the ISS 2/3 code to record at higher bit densities.

Increasing the recording density in a parallel-format tape recorder, however, highlights a further problem. As the recording rate in a fixed-head multitrack tape recorder increases the problem of tape–head azimuth variation (tape skew) becomes more severe. This is especially so with inexpensive tapes and tape decks such as those used in compact cassette systems. Instrumentation and data recording tape systems use sophisticated tape-deck mechanisms and complex de-skew circuitry to combat the effects of tape skew.[4] Such solutions would prove too costly to be applied to compact cassette tape recording, the philosophy of which is essentially low cost. The tape skew variation in compact cassette tapes is found to be of a low frequency, cyclic nature.[5] This permits its measurement to be used in a method of generating a skew-correction signal. This signal is used to de-skew the data by introducing the right amount of skew into the clock signals which clock the data off the tape tracks. The measurement of tape skew involves the encoding of control pulses on the outside tracks of the tape. These pulses are also used in decoding the ISS 2/3 code by identifying each group of 3 code bits.

In this paper an ISS 2/3 parallel encoding/decoding software algorithm is described. A low-cost method of reducing the effects of tape skew in a compact cassette tape recorder is also given. Both have been applied to an 8-track compact cassette tape recorder by recording digital data directly onto the inner 6 tracks and control signals onto the outside tracks. These control signals are also used to identify each 3-bit group of the ISS 2/3 code.

2 Parallel Encoding of ISS 2/3 Code

Full details of the ISS 2/3 code are given in the paper by Jacoby and Kost.[3] The basic conversion of data bits into code bits is shown in the encoding table of Table 1 where the coded 'ones' represent a signal transition (either +ve or −ve).

Information is to be recorded 'broadside' onto the tape in a 6-bit parallel format. The encoding process involves converting two, 6-bit data words into three, 6-bit codewords. The software algorithm used to convert the data bits into code bits is

1st code bit: Complement first data bit
2nd code bit: AND first and second data bits
3rd code bit: Complement second data bit.

The above algorithm will fail under certain illegal sequences of data bits. These are the sequences which give two adjacent code ones (Table 2). The illegal code sequences are replaced by a code sequence which always terminates with three zeros.

Table 2. Illegal code bits and their replacement

Sequence	Data	Illegal code	Replacement code
a	0000	101101	101000
b	0001	101100	100000
c	1000	001101	001000
d	1001	001100	010000

Table 3. Data modification required (X = don't care)

Sequence	Data	Data modification
b	0001	01XX
d	1001	11XX

The encoder program looks forward over four data words and forms a mask identifying the illegal sequences. These are readily identifiable by consecutive zeros in positions two and three of each illegal data word. The complement of the mask is ANDed with the second group of three codewords before recording. This terminates the illegal sequences with three zeros whilst the remaining, legal sequences are unaffected.

In the case of illegal sequences 'a' and 'c', the first three code bits will be generated correctly by the algorithm. Sequences 'b' and 'd' have to be modified as shown in Table 3 before the algorithm will generate the correct code. The illegal sequences which are to be modified are identified by a one in their fourth position.

This masking technique allows legal and illegal sequences of data to be processed in a parallel operation. Furthermore, the encoding algorithm applies a number of simple logical operations on the data. These logical operations can be found on most small microprocessors and since they are non-memory reference instructions they are implemented in the minimum of computer time. Generating the code immediately from the data also obviates the need for storing look-up tables of data. This method of encoding data permits recording rates in excess of 20 kb/s per track to be achieved using a Zilog Z80 microprocessor with a 4 MHz system clock.

To decode the data each group of three code bits must be identified. This is done by recording control pulses on the outer tracks of the tape for every three code bits. These control pulses will also be used to de-skew the data on the inner tracks.

3 Parallel Decoding of ISS 2/3 Code

The ISS 2/3 decoding table is given in Table 4. It can be seen that seven codewords have to be examined for every two data words that are decoded.

The decoding process involves identifying present and succeeding 3-bit code groups which are non-zero. This is accomplished by logically ORing each 3-bit code group. With the exception of the case when the present codeword is zero, the first decoded data bit is the complement of the first code bit. If the succeeding codeword is not zero the second data bit is the complement of the third code bit: if it is zero the second data bit is also zero. When the present codeword is zero the first decoded data bit is zero and the second decoded data bit is the complement of the last bit of the previous codeword. The decoding process is summarized in Table 5. The decoder program generates 'non-zeros' masks for each 3-bit code sequence and uses these to generate the appropriate data bits.

The transitions of the ISS 2/3 code occur at mid-bit cell only. This permits the read data clock to run at bit rate and not twice bit rate as with MFM code. The transitions, marking a logical one, are detected by the decoding software by exclusively ORing the code bits, as they are read by the microprocessor, with the previous reading. The decoding software also includes provision for flag status checking and resetting. These software overheads increase the decoding time slightly to give a maximum decoding rate of 13 kb/s per track.

As with the encoder the application of masking and logical operator techniques permits full parallel decoding of data whatever the mixture of legal and illegal sequences.

Table 4. Decoding table (X = don't care, N = not all zeros)

Previous codeword last bit	Present codeword	Succeeding codeword	Decoded dataword
X	101	NNN	00
X	100	NNN	01
X	001	NNN	10
X	010	NNN	11
X	101	000	00
X	100	000	00
X	001	000	10
X	010	000	10
0	000	XXX	01
1	000	XXX	00

Table 5. Summary of decoding rules

Present codeword	First decoded data bit complement of first code bit	Succeeding codeword not all zeros— Second decoded data bit complement of third code bit	Succeeding codeword all zeros— Second decoded data bit always zero
101	0	0	0
100	0	1	0
001	1	0	0
010	1	1	0
Present codeword all zeros	**Previous codeword last bit**	**First decoded data bit always zero**	**Second decoded data bit complement of previous codeword last bit**
000	0	0	1
000	1	0	0

4 Tape Skew

Because of tape skew the code bits appearing under each read gap of the multitrack read head may not all belong to the same group of six code bits as when they were recorded. For coherent detection of each codeword some means must be found of correcting for tape skew. The technique adopted is to measure the effect of tape skew and use this result to generate a correcting signal.

4.1 Tape Skew Measurement

A method of measuring the effect of tape skew has been determined.[5] This involves the positioning of control pulses on the outer tracks of the tape. On playback these pulses control timers from which either the skew angle or

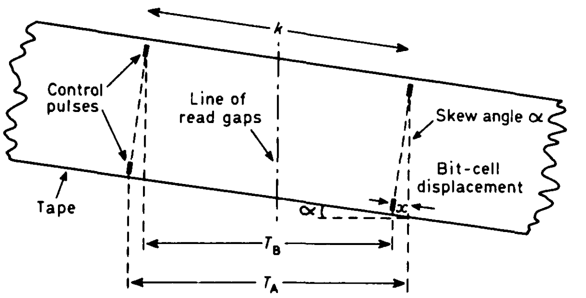

Fig. 1. Bit-cell displacement, x, caused by skew angle α.

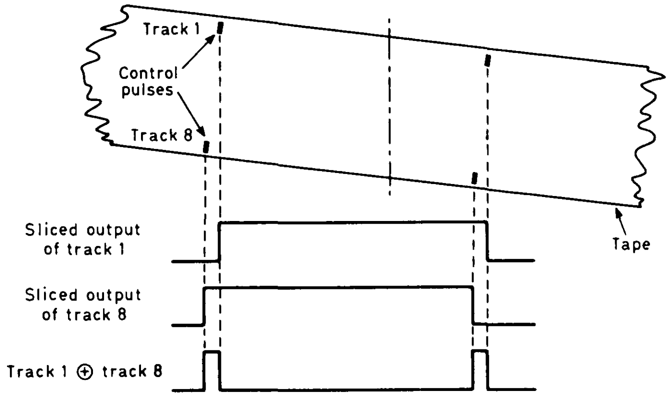

Fig. 2. Pulses derived from control tracks.

the bit-cell displacement may be calculated. A simplification of this measurement method gives a pulse whose length is proportional to the maximum extent by which a code bit is displaced from the read gap due to tape skew.

Reference to Fig. 1 shows that,

$$\cos\alpha = \frac{(T_A - T_B)}{2x} \quad (1)$$

where α is the tape skew angle, x the bit-cell displacement and T_A, T_B are time intervals, Also

$$\cos\alpha = \frac{T_B + \frac{1}{2}(T_A - T_B)}{k} \quad (2)$$

where k is the distance between control pulses. From equations (1) and (2)

$$\text{bit-cell displacement } x = \frac{(T_A - T_B)}{(T_A + T_B)}k \quad (3)$$

In compact cassette tape systems the skew angle α is in the range $-0{\cdot}1° < \alpha < +0{\cdot}1°$. Therefore

$$k \approx T_A - \frac{(T_A - T_B)}{2}$$

$$\approx \frac{(T_A + T_B)}{2} \quad (4)$$

Substituting equation (4) into equation (3) gives a measurement of bit cell displacement in terms of time:

$$x = \frac{(T_A - T_B)}{2} \quad (5)$$

The time $(T_A - T_B)$ may be obtained by exclusively ORing the control pulses on the outer tracks (Fig. 2).

Using the exclusive-OR of the control pulses to generate the signal $(T_A - T_B)$ gives a result which is independent of read-amplifier slice-level variations and/or signal amplitude. If one read amplifier slices its control pulse input at a different level, one of the two pulses of the $(T_A - T_B)$ signal will increase in length but the other will decrease proportionately.

The above measurement technique does not account for the possibility of the bit-cell displacement being maximum on one of the inner tracks. This is likely to occur, especially with C90 and C120 tapes.[5] The following de-skewing technique, however, is robust enough to accommodate a bit-cell displacement error of $\pm 1/2$ bit cell at 10 kb/s.

4.2 De-skewing of Code Bits

The code bits of each track are detected by the circuit shown in Fig. 3. The clock, which is synchronized to the code bits on the tape, is switched between two data latches. The switching is controlled from a toggle operating at the bit rate of the code. As each code bit is latched an associated flag is set. When the corresponding

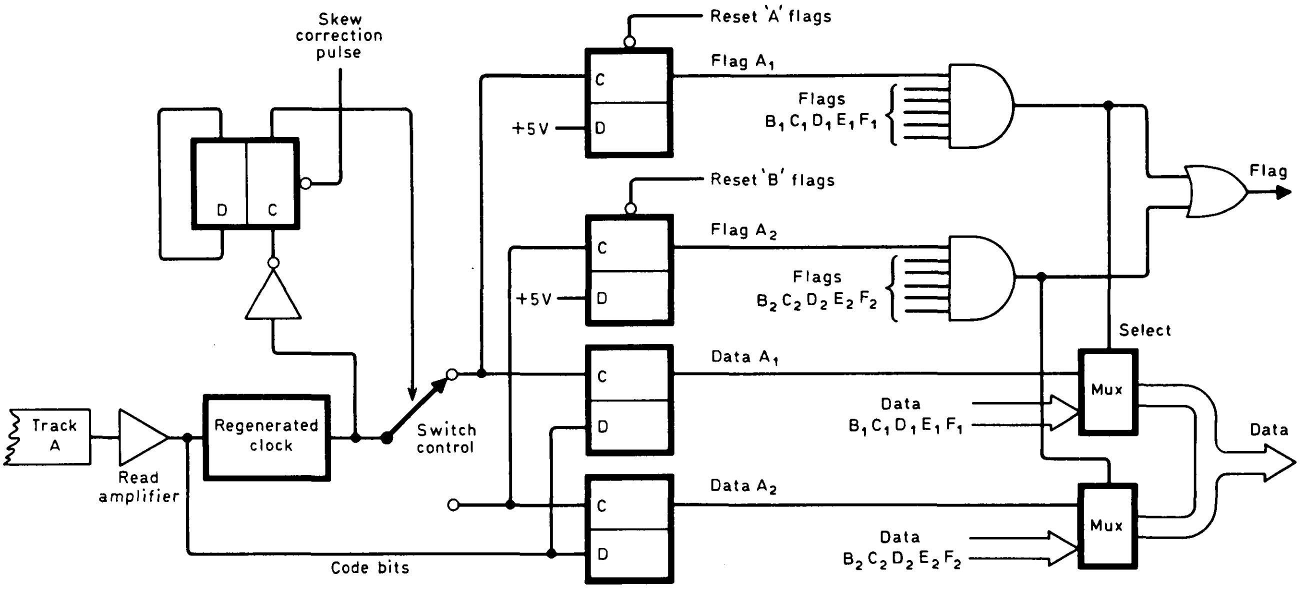

Fig. 3. Clock switch.

flag from each track is set the associated 6-bit codeword is read by the microprocessor via a multiplexer.

Assuming that all six clock switches are selecting their respective code bits and that the maximum bit-cell displacement does not exceed ± 2 bit cells then, because each track is independently clocked, the 6-bit codeword will be available at the output of the associated latches in de-skewed form. Should the envisaged bit-cell displacement exceed ± 2 bit cells then the number of data latches per track could be increased and the clock control toggle replaced by a counter.

In a practical system, tape dropouts and glitches will cause loss of synchronism between the six tracks and the code output of the latches will be garbled. Also, it must be possible to stop and restart the tape at any point. The clock control switches of each track must, therefore, be continually synchronized in line with the current state of the tape skew. This is accomplished by generating skew correction pulses which align each clock switch to its respective code bit. The correct order in which the skew correction pulses are generated is determined by the tape skew measurement $x = \frac{1}{2}(T_A - T_B)$. During this time interval the circuit of Fig. 4 generates six initialize pulses.

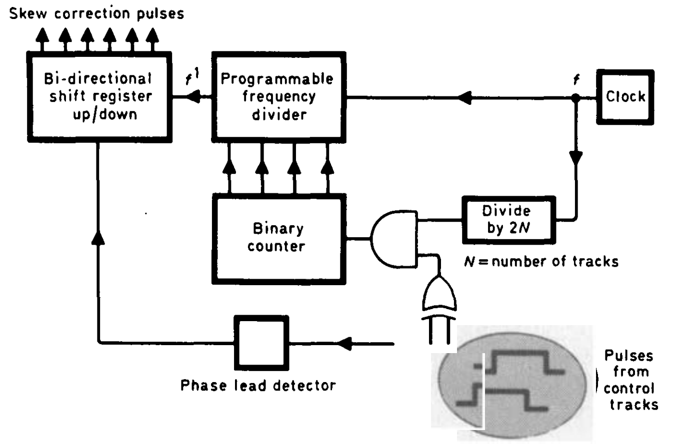

Fig. 4. Skew-correction pulse generator.

During the time interval, which is proportional to bit-cell displacement $(T_A - T_B)$, the binary counter accumulates a value of $f(T_A - T_B)/2N$ counts, where a clock frequency of f Hz is counted for a period of $(T_A - T_B)$ seconds via a divider of $2N$. N is the number of pulses required.

The accumulated count is loaded into a programmable frequency divider which is then successively counted down until each of the shift register outputs has been asserted. The frequency of the pulses fed to the shift register is given by the input frequency of the programmable frequency divider divided by its contents:

Shift register input frequency =

$$\frac{f}{f\dfrac{(T_A - T_B)}{2N}} = \frac{2N}{(T_A - T_B)} = \frac{N}{x}\,\text{Hz}$$

The input clock pulses to the shift register are thus spaced precisely over the time period represented by the bit-cell displacement, whatever value this may take (Fig. 5).

The leading control track determines the direction of the shift register, thus allowing for both positive and negative tape skew. The flux transitions on this track also identify the centre of the bit cells; thus, each shift register output pulse coincides with the centre of its corresponding bit cell—all six pulses identifying the appropriate bit of the 6-bit codeword that was recorded. Essentially the de-skew

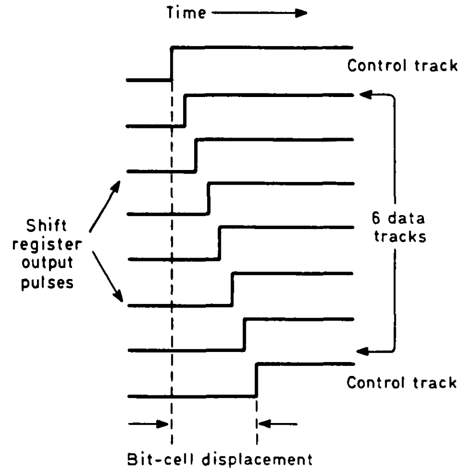

Fig. 5. Skew correction pulses.

circuitry samples the value of the tape skew and initializes each clock switch accordingly. This is repeated at intervals determined by the spacing of the control pulses on the outer tracks.

5 Conclusion

A software encoding/decoding method has been presented which permits parallel operation of the new ISS 2/3 code. This makes possible the recording and subsequent decoding of data in a multitrack format up to a frequency of 13 kb/s per track using a Zilog Z80 microprocessor with a 4 MHz system clock.

The above encoding/decoding method may be used to record digital data directly onto compact cassette tape run on an inexpensive tape deck. The tape skew which results when such a combination of tape and tape deck are employed is compensated for by using simple skew-correction circuitry. This compensates for a tape skew of up to ± 2 bit-cells at 10 kb/s between the outer tracks. Tape skew in excess of this value may be accommodated by increasing the number of stages in the de-skew hardware.

Data at a rate of 10 kb/s per track has been recorded on the six inner tracks of an 8-track compact cassette tape using the above code and skew correction method. At this level of recording density, data-channel equalization becomes necessary to enhance the data detection process; work is continuing in this direction. Also, channel crosstalk within the custom-built 8-track head impairs data detection.

Improvements in head design and the introduction of channel equalization would increase the recording density. In addition, consideration may also be given to recording data on the outer tracks of the tape with the incorporation of skew-correction control pulses as code violations.

6 References

1 Mallinson, J. C., 'The next decade in magnetic recording', *IEEE Trans. on Magnetics*, **MAG-21**, no. 3, pp. 1217–20, May 1985.
2 Mallinson, J. C. and Miller, J. W., 'On optimal codes for digital magnetic recording', IERE Conference on Video and Data Recording, Birmingham, July 1976, pp. 161–9 (IERE Conf. Publication No. 35)
3 Jacoby, G. V. and Kost, R., 'Binary two-thirds rate code with full word look ahead', *IEEE Trans. on Magnetics*, **MAG-20**, no. 5, pp. 709–14, September 1984.
4 EMI, 'Modern Instrumentation Tape Recording: An Engineering Handbook', p. 65 (EMI Technology, 1978).
5 Donnelly, T., Mapps, D. J. and Wilson, R., 'Real-time microprocessor monitoring of skew angle in a multitrack magnetic tape system', *J. Instn Electronic & Radio Engineers*, **56**, no. 2, pp. 49–52, February 1986.

Manuscript received by the Institution in final form in June 1986
Paper No. 2295/REC22