REAL TIME MICROPROCESSOR TECHNIQUES FOR A DIGITAL MULTITRACK TAPE RECORDER.

Terence Donnelly B.Sc. M.Sc.

Submitted to the Council for National Academic Awards in Partial Fulfilment for the Degree of Doctor of Philosophy.

Sponsoring Establishment:
Plymouth Polytechnic
Department of Electrical and Electronic Engineering.

Collaborating Establishment:
British Broadcasting Corporation.

January 1989
DECLARATION.

I declare that this thesis is the result of my investigation only, and is not submitted in candidature for the award of any other degree. During the research programme I was not registered for the award of any other C.N.A.A. or University degree.

21/12/1988

J. Deane

ADVANCED STUDIES.

During the research programme I undertook a course of advanced studies. These included attendance at several conferences and colloquia. Papers were presented at: the Sixth International Conference on Video and Data Recording, Brighton, 1986; Euromicro '87 Conference, Portsmouth, 1987; Institute of Acoustics Reproduced Sound 3 Conference, Windermere 1987.
ABSTRACT.

REAL - TIME MICROPROCESSOR TECHNIQUES FOR A DIGITAL MULTITRACK TAPE RECORDER.

by

Terence Donnelly

Transport properties of a standard compact-cassette tape system are measured and software techniques devised to configure a low-cost, direct digital recording system.

Tape-velocity variation is typically ±10% of standard speed over tape lengths of 5 μm, with occasional variations of ±40%. Static tape-skew can result due to axial movement of the tape reel when it spools. Dynamic tape skew occurs and is primarily caused by tape-edge curvature with a constant contribution due to the transport mechanism. Spectral skew components range from 0.32 Hz to 8 Hz with magnitude normally within one 10 kbit/sec-bit cell. The pinch roller works against the friction of the tape guides to cause tape deformation. Average values of tape deformation are: 0.67 μm, 0.85 μm and 1.08 μm for C60, C90 and C120 tape respectively.

Parallel, software encoding/decoding algorithms have been developed for several channel codes. Adaptive software methods permit track data rates up to 3.33 k bits/sec in a multitrack system using a single microcomputer. For a 4-track system, raw error rates vary from 10^-7 at 500 bits/sec/track to 10^-5 at 3.33 kbits/sec/track. Adaptive software reduces skew-induced errors by 50%. A skew-correction technique has been developed and implemented on an 8-track system at a track data rate of 10 k bits/sec.

Real-time error correction gives a theoretical corrected error rate of 10^-11 for a raw error rate of 10^-7. Multiple track errors can cause mis-correction and interleaving is advised. Software algorithms have been devised for Reed-Solomon code. With a more powerful microprocessor this code may be combined with the above techniques in a layered error-correction scheme.

The software techniques developed may be applied to N tracks with an N-bit computer. Recording density may be increased by using thin-film, multitrack heads and a faster computer.
ACKNOWLEDGEMENTS.

I would like to thank all the people who have been associated with this project during the past few years, in particular:

Professor D. J. Mapps of the Department of Electrical and Electronic Engineering, Plymouth Polytechnic, for his advice and guidance.

Mr. R. Wilson, now of Leeds Polytechnic, for his support and encouragement.

Mr. F. A. Bellis, of the B. B. C, for his constructive advice and useful observations.

The technical staff of the Department of Electrical & Electronic Engineering, Plymouth Polytechnic, for their practical help.

And, not least of all, to my wife Anne to whom I dedicate this work.
# TABLE OF CONTENTS

DECLARATION ................................................................. ii
ABSTRACT ................................................................. iii
ACKNOWLEDGEMENTS ....................................................... iv

INTRODUCTION ............................................................. 1

1. BACKGROUND TO THE INVESTIGATION.

1.1. OVERVIEW OF MODERN RECORDING METHODS ....................... 5

1.1.1. Magnetic Recording .............................................. 5
1.1.2. Optical Recording .............................................. 8
1.1.3. Solid State Recording ......................................... 14

1.2. APPLICATIONS OF DIGITAL TAPE SYSTEMS ......................... 16

1.2.1. Entertainment .................................................. 16
1.2.2. Instrumentation ............................................... 19
1.2.3. Computer Storage .............................................. 20

1.3. COMPACT CASSETTE ................................................ 21

1.3.1. Compact-Cassette Format .................................... 21
1.3.2. Tape Coating .................................................. 24
1.3.3. The Compact-Cassette Channel ............................... 25
1.3.4. Frequency Response .......................................... 26

1.4. COMPACT-CASSETTE STORAGE SYSTEMS ........................... 27

1.4.1. Analogue Recording of Digital Signals ....................... 29
1.4.2. Linear, Direct Recording of Digital Signals ................. 31
1.4.3. Saturated, Direct Recording of Digital Signals ........... 35
1.4.4. Multitrack Recording Systems ................................ 37

1.5. MECHANICAL PERFORMANCE OF COMPACT-CASSETTE TAPE
    TRANSPORT .......................................................... 39

1.5.1. Tape - Velocity Variation .................................... 40
1.5.2. Tape - Azimuth Variation ................................... 40

1.6. ERROR CORRECTION ................................................ 41

1.6.1. Simple Parity ................................................ 42
1.6.2. B.C.H. Code .................................................. 43
1.6.3. Intelligent Methods .......................................... 45
3.3. ANALYSIS OF MOVING PARITY ERROR CORRECTION. ..........111

3.3.1. Efficiency of Moving Parity Code. ..........111
3.3.2. Theoretical Deduction of Moving Parity Error Correction. ..........114

3.4. SOFTWARE IMPLEMENTATION of REED - SOLOMON CODE. ..........116

3.4.1. Reed - Solomon Encoder. .............121
3.4.2. Reed - Solomon Decoder. ..........123

3.5. REFERENCES FOR CHAPTER 3. .................126

4. EXPERIMENTAL APPARATUS AND TECHNIQUES.

4.1. INTRODUCTION. ....................................131

4.2. APPARATUS. ........................................132

4.2.1. Compact - Cassette Recording System 1. ..132
4.2.2. Tape Transport. ................................134
4.2.3. Solenoid Control. ..............................134
4.2.4. Write Amplifier. ...............................135
4.2.5. Read Amplifier. .................................137
4.2.6. Microcomputer. .................................140
4.2.7. Test Equipment. .................................140
4.2.8. Software. ......................................142

4.3. MEASUREMENTS. ....................................142

4.3.1. Record Head Characteristics. ..............143
4.3.2. 4 - Track Inductive Head. .................144
4.3.3. 8 - Track Inductive Head. .................144
4.3.4. 18 - Track, Thin - Film Head. ............146
4.3.5. Tape - Velocity Variation. .................147
4.3.6. Theory of Tape Velocity Measurement. ......147
4.3.7. Tape - Velocity Measuring Apparatus. ......149
4.3.8. Tape Skew. .................................151
4.3.9. Skew Angle. .................................151
4.3.10. Theory of Skew Measurement. ............152
4.3.11. Skew - Angle Measuring Apparatus. ......154
4.3.12. Analysis of Measuring Technique. .........155
4.3.13. Tape Deformation. .........................158
4.3.14. Theory of Tape - Deformation Measurement. .........159
4.3.15. Tape - Deformation Measuring Apparatus. ....159
4.3.16. Analysis of Measuring Technique. ........162

4.4. COMPACT - CASSETTE RECORDING SYSTEM 2. ..............163

4.4.1. Hardware. .................................163
4.4.2. Write Amplifier. ..........................164
4.4.3. Read Amplifier. ..........................165
4.4.4. Software. ................................166
4.4.5. Biphase - L Code. ......................166
4.4.6. Biphase - L Recording Software. .......168
4.4.7. Biphase - L Playback Software. .........169
4.4.8. Miller Code. ............................173
4.4.9. Miller Code Recording Software. .......174
4.4.10. Miller Code Playback Software. ........177
4.4.11. 2/3 Rate 0,2 Code. ....................178
4.4.12. Code Description. .....................179
4.4.13. 2/3 Rate 0,2 Code Recording Software. 180
4.4.14. 2/3 Rate 0,2 Code Playback Software. 180
4.4.15. Evaluation of Recording System 2. ....181
4.4.16. Distribution of Track Errors. ...........181
4.4.17. Error Burst Length. ....................181

4.5. COMPACT - CASSETTE RECORDING SYSTEM 3. ..........182

4.5.1. ISS 2/3 Software Algorithm. ..............184
4.5.2. Encoding Algorithm. ......................185
4.5.3. Decoding Algorithm. ......................187
4.5.4. Codebit Synchronisation And Detection. ..189
4.5.5. Clock Regeneration. ......................190
4.5.6. Clock Switch. ...........................190
4.5.7. Clockswitch Synchronisation. ............192

4.6. ERROR CORRECTION. ........................195

4.6.1. Moving Parity Error Correction. ..........195

4.7. REFERENCES FOR CHAPTER 4. ...............199

5. RESULTS.

5.1. INTRODUCTION. ............................202
5.2. TAPE - HEAD RESPONSE. ....................203
5.3. TAPE VELOCITY VARIATION. .................209
5.4. TAPE - AZIMUTH VARIATION. .................209
5.4.1. Measurement of Tape Azimuth. ............210
5.4.2. Spectral Analysis of Tape Azimuth. .....211
5.5. TAPE DEFORMATION ..........................211
5.6. ERROR CHARACTERISTICS OF RECORDING SYSTEM 1. ..222
5.6.1. Error Burst Length. .....................222
INTRODUCTION.

The demand for high density, low-cost, information-storage products can be attributed to the development of the small computer and the ease and accuracy with which analogue signals can be processed and stored if first converted to digital form. The widespread use and application of microcomputers with ever increasing operating speeds has not been possible without storage systems capable of accommodating the vast quantities of data microcomputers can generate. Not only do these systems compute data at extremely high rates they also process data bits simultaneously in parallel. Digital techniques are now applied extensively in the analogue field. Audio signals are digitised and stored on optical disc or magnetic tape. Magnetic tape is also used to store digitised analogue signals in data-logging applications and digital video tape recorders are under development.

The vast majority of digital storage systems utilise magnetic recording techniques. The market for these products has been estimated to be worth 35 billion dollars in 1986 and growing at a rate of 20% per year [1].

The two principal magnetic media are the hard disc and the tape. Currently, Winchester discs are capable of storing up to 5 Gbytes of data at rates up to 3 Mbytes/sec. These devices require a highly sophisticated mechanised drive and read/write head assembly to accomplish these high storage capabilities. Strict environmental conditions must be maintained to exclude possible
contaminants such as dust and smoke which, with a flying height measured in thousands of angstrom units, could cause the head to crash.

Although hard disc offers a higher data throughput than magnetic tape their areal storage density is some two orders less. Helical scan and multitrack formats utilise the full area of tape in rotary head and stationary head recorders to bring about a higher areal storage density for tape systems compared to disc. Also, magnetically recorded tape has a higher archival life than magnetic disc. The foregoing characteristics of magnetic tape make it suitable for disc backup in addition to being used as a stand alone storage device.

In increasing the areal storage density of a magnetic tape system an increase in the complexity of the tape / head interface is usually necessary. This often results in a higher degree of mechanical tape - transport sophistication and a reduction in robustness. Attendant with these "improvements" is an increase in system cost and volume. As recording density increases so too does error rate. Tape flaws, poor signal - to - noise ratio and increased time jitter, serve to increase the number of errors as a proportion of error - free data.

Digital tape recorders either incorporate microprocessor technology or are used in connection with a microcomputer system. In seeking to improve the performance of such a recorder an alternative strategy to increasing its mechanical complexity is to utilise the decision - making properties of the micro -
computer. In applying such a device intelligent methods can be used to compensate for the mechanical vagaries of the tape transport. A further advantage of this approach is that, for a stationary - head multitrack recorder, each track can be simultaneously processed - one bit of the microprocessor to each track.

This permits the microcomputer to be used to fulfill some of the recording / reproduce functions such as encoder / decoder, write amplifier, clock regenerator, etc. normally implemented in hardwired electronics.

The above approach has been adopted in the work reported on in this thesis. Microprocessor techniques have been applied to implement a low - cost, compact - cassette digital recording system. This follows measurements made on the tape - transport characteristics of such a system. The first chapter reviews the area of recording and assesses the relative performance of existing, low - cost, tape recording systems. In chapter 2 the characteristics of magnetic recording systems are examined. The theory of error correction and recording codes follows with some analysis: - consideration here is given to real - time, software implementation. Chapter 4 covers the design of hardware and software used in the configuration of a number of recording systems after which the data generated by such systems is presented. Finally, the results of the research programme are discussed with conclusions.
REFERENCES FOR INTRODUCTION.

CHAPTER 1.

BACKGROUND TO THE INVESTIGATION.

1.1. OVERVIEW OF MODERN RECORDING METHODS.

Various techniques have been developed for recording information which can be represented by an electrical signal. Due to the advantages of digital representation and the demand created by the widespread application of low-cost digital computers these techniques are almost exclusively in the digital domain.

A brief review of modern recording methods is given. These fall into three broad categories: magnetic, optical and solid state.

1.1.1. Magnetic Recording.

Advantages of magnetic recording include reversibility and non-volatility. Media in disk or tape form is inexpensive, robust and portable. However, the precision with which the media can be transported across the read/write head is important in determining the density at which data may be stored. For this reason, tape transport mechanisms and disc drives are often mechanically sophisticated and thus represent a major cost item in a magnetic recording system.

The vast majority of magnetic recording devices utilise longitudinal recording. A magnetic layer is passed across the fringing field formed by the pole pieces of a ring-type record head. This results in magnetisation in the direction of the media travel, (figure 1.1.).

A limiting factor associated with longitudinal recording is the
Fig. 1.1. Illustrating longitudinal recording.

Fig. 1.2. Illustrating perpendicular recording.
demagnetising field generated by the recording patterns. As the magnetisation reversal region is decreased the demagnetising field increases setting an upper limit to the packing density. Demagnetising effects may be reduced by increasing the coercivity of the magnetic coating and/or reducing its depth. Improvements in these parameters are limited, however, because of pole-tip saturation and the need to maintain a sufficient magnetising-layer thickness to allow for wear. Densities greater than 100,000 fci are now possible with longitudinal recording.

The above packing density may be increased if the recording is made perpendicular to the direction of media travel, (figure 1.2.). This is possible using a cobalt-chromium (CoCr) recording film deposited on a permalloy underlayer. CoCr has a perpendicular oriented crystalline anisotropy which permits media recording through the full depth of the magnetic layer. Further, the magnetic field between two elements marking a transition reinforce each other thus reducing the transition zone. Demagnetising effects reduce with increasing packing density and, theoretically, the transition zone is zero as the recording density approaches infinity. The transition zone is, however, limited by the size of the grains of CoCr which, at 50nm diameter gives a possible recording density of 500,000 fci.

Bauer [1] has recorded data on C90 compact cassette using perpendicular recording techniques. Data are recorded at 19.2k bits/sec. to give an overall storage capacity of 20 Mbytes.

As yet the promise of perpendicular recording techniques remains

*flux changes per inch.*
to be realised on an economic scale. High - density systems have been demonstrated however,[2] and development continues.

1.1.2. Optical Recording.
With optical techniques it is possible to produce storage devices which are

(a) for reading only

(b) write once read many times (WORM)

(c) rewritable.

The read only method is exemplified by the compact disk [3]. Data are stored in the form of indentations (pits) impressed into the surface of a disk. These form a spiral track onto which is focussed an AlGaAs laser of wavelength 780nm. The depth of the pits equals one quarter of the wavelength of the laser beam. The interference between the reflected and incident beam is used to differentiate between the pits and the interval between pits known as the land. Minimum pit length is 1µm giving a lineal recording density of 25,400 bits/inch. Because of the high resolution of the laser spot, areal density is very high. The spiral track of pits is located below the surface of the disk giving a system which is tolerant of surface contamination and minor damage. The "read" only disks are produced (recorded) under clean room conditions. By comparison the WORM disk is recorded on site.

Several techniques are used to encode the data on a WORM disc, four of these are illustrated in figure 1.3., [4]. Essentially the reflective qualities of the disk surface are changed by
Fig. 1.3. Four methods of encoding a WORM disc.
the local heating effect of a laser beam. These include decreasing or increasing surface reflectivity by respectively burning a hole or changing the surface texture of a tellurium alloy layer on a polymer substrate. Other methods involve the alloying together of two surface-coated films to give an area with different optical properties to the untreated surface and the formation of shallow bubbles in the platinum coating on a polycarbonate substrate. Data densities are the same as for the "read only" method and the recording process is once only and permanent.

WORM systems are currently available but as yet they are costly when compared to magnetic storage systems.

Considerable interest is currently being generated with the announcement of optical tape [5]. This new, write-once medium is a dye polymer coated onto a flexible polyester-based substrate. Packing densities are quoted as 600 Gbytes per 10.5 inch tape reel with 1 terabyte tape reels under development.

Although still at the experimental stage it is possible to record and read data many times using optical methods. Two techniques are under development: phase change and magneto-optic.

If heated to the molten state and allowed to cool rapidly the crystalline state of certain alloys changes to the amorphous state. This permits the writing of amorphous marks using a pulsed laser beam. These marks have different optical characteristics to the unaffected, crystalline areas. Erasure is effected using an oblong laser spot to return the amorphous marks to the
crystalline state - the oblong spot heating the designated mark for some 10 µs compared to 100ns during recording.

As the name implies magneto-optic recording involves the application of both magnetic and optical methods. The coercivity of a thin film is reduced locally by laser heating. This permits the magnetisation of the heated area to be switched by a general magnetic field. Readout is accomplished by detecting the Kerr rotation in the polarisation of a light beam as it is reflected from the recorded areas of the disk.

![Diagram](image)

**Fig.1.4.** Illustrating the variation of coercive force $H_c$, magnetisation $M_s$ and Kerr rotation angle $\Theta_k$ with temperature for an amorphous rare earth transition metal alloy.

*Figure 1.4. illustrates the variations of coercive force, magnetisation and Kerr rotation angle, as a function of*
temperature for an amorphous rare earth transition metal alloy. These are ferrimagnetic materials and are used in the magneto-optic recording process. As the temperature of the material is increased the magnetisation, \( M_s \) falls until, at the compensation temperature for the material, \( M_s \) reaches zero. At \( T_{comp} \) the coercive force, being inversely proportional to \( M_s \), tends towards infinity. With increasing temperature above \( T_{comp} \) the coercive force falls to reach zero at the Curie temperature.

To record data a spot on the film is heated by a laser beam (figure 1.5.). After an initial rise in coercive force it falls until it reaches a value below that of an applied external field. The direction of the field determines the direction of magnetisation of the film after the laser spot is turned off. After recording, the film comprises areas of magnetisation the directions of which represent logical 1 or logical 0. These are detected on readout by determining the difference in polarization of a light beam as it reflects from the areas of the film having opposite magnetic directions. This Kerr effect gives a difference in the polarization of reflected light of the order of 0.25 to 0.5 degrees.

Magneto optic recording material such as terbium gadolinium iron are used. Compensation temperature for this type of material is in the region of 150°C. Recorded bit diameters are of the order of 1 \( \mu \)m thus giving a system with a potentially high recording density.
Fig. 1.5. Illustrating magneto-optic recording.
1.1.3. Solid State Recording.

Although it is possible to utilise conventional CMOS semiconductor RAM with battery backup in a high-density recording system, at the time of writing, the high costs involved limit this form of storage to all but the specialist of applications [6]. Advantages of solid-state recording devices include durability and reliability under harsh environmental conditions, they are also maintenance free. Such a device is the magnetic bubble memory [7] which, unlike semiconductor memory, is non-volatile. Magnetic "bubbles" in the form of magnetic domains are generated in a magnetic garnet film and passed around a track of permalloy overlay elements by a circulating magnetic field. The "bit" size is typically 4μm x 4μm permitting a 4Mbit device to be packaged in an area 4.25cm x 3.75cm.

A limit on the packing density of bubble memories is the use of permalloy tracks — a number of factors are involved. The stray magnetic field from the bubbles increases as bubble size decreases. This stray field magnetises the permalloy tracks and this has to be overcome by increasing the drive field. Further, higher densities require smaller track electrodes which in turn require increased drive field to overcome the weaker attractive forces of the electrodes. Suzuki [8] details the use of ion implantation techniques to overcome this limitation. The tracks are formed by ion implantation into a garnet film. The easy direction of magnetisation of this film is perpendicular to the plane of the film. The easy direction of magnetisation of the implanted area is in-plane causing charged walls to form in the
thickness of the implanted track. These charged walls attract the magnetic bubbles and pass them along the track when the in-plane drive field is rotated. 16 Mbit hybrid bubble memory devices using both permalloy and implanted tracks are under development with 64 Mbit chips predicted.

At the experimental stage is a development of the bubble memory: the Bloch line memory [9]. Here binary data are stored in Bloch-line pairs which are formed in the wall of the magnetic bubble. The width of the Bloch lines is about one fourth the diameter of a bubble. Also, since the depth of the wall is much less than the bubble size the potential for storing data at ultra high density is very great. Konishi [9] estimated a possible storage density of 1 Gbit/square cm using 0.5μm bubble material.

Of the various recording technologies optical devices currently offer the highest areal packing density. However, true read/write systems are still at the development stage and system costs will be high compared to magnetic recording. The density of optical devices is approached by the predicted capability of Bloch line memories, however, at the time of writing these are at the experimental stage.

Cost-effective read/write storage is exemplified by magnetic recording devices, in particular tape systems which offer an areal density several times that of rigid disk. By comparison tape systems are inexpensive—particularly media costs—which are significantly lower than rigid disk. The only serious disadvantage to tape systems is the long access time compared to
magnetic and optical disk. There are, however, a number of applications where the inherent timebase of the tape system is of benefit and in disk back-up situations access time is irrelevant. Consequently the demand for digital tape systems is high [10].

1.2. APPLICATIONS OF DIGITAL TAPE SYSTEMS.

The market for digital tape systems covers the spectrum of storage applications from the entertainment field, video and audio, through industrial and military applications to computer memory requirements. In computing and entertainment, tape recording is used both at the high volume, consumer level and at the high performance, professional level. There is also a substantial market for data-acquisition applications, particularly in respect of inexpensive and portable equipment. Each area of application: entertainment, instrumentation and computer storage is considered.

1.2.1. Entertainment.

There are a number of digital tape recording standards in the audio entertainment area: rotary-head digital audio tape recorder (R-DAT) [11] and stationary-head digital audio tape recorder (S-DAT) [12] in the consumer field and (DASH) [13], digital audio stationary head, in the professional area. The high tape/tape head velocity of R-DAT plus use of the helical scan technique, permits the high density requirements of digital audio to be accommodated on a tape cassette measuring 7.2 cm x 5.2 cm. Two, 16-bit samples are stored at a rate of 44.1 kHz giving an areal storage density of 176 kbits/square mm. The raw error rate is $10^{-3}$. 

16
to $10^{-4}$ necessitating a comprehensive error-correction strategy.
The R-DAT track format is shown in figure 1.6.

Fig.1.6. R-DAT track format.

Azimuth recording is used to accommodate head mis-alignment. The high tape/tape - head velocity ensures adequate bandwidth whilst the low tape velocity gives playing times of up to 2 hours. This long playing time is achieved by using metal powder tape of overall thickness 13 $\mu$m, 3 $\mu$m of which is the magnetic coating.

In the S-DAT format (figure 1.7.) 20 data tracks plus 2 auxiliary tracks occupy half the width of tape 3.81mm. wide, a
similar track set occupies the other half of the tape permitting reverse play. Data-track width is 65μm wide requiring thin-film record heads. The use of Magnetoresistive (MR) read heads permits the slow tape speed necessary to accommodate a play time of up to 180 minutes. Error correction is effected using Reed-Solomon code in a co-ordinate format. As of the time of writing S-DAT products have yet to become available on the open market.
R-DAT has been developed as a consumer product and although its fidelity of reproduction approaches professional acceptability, its mechanical shortcomings limit its use in this area. For professional applications, tape splicing/editing requirements, plus the need for flexibility and greater dynamic range than consumer products, usually dictate the use of stationary-head, open-reel recorders such as the two-channel "Prodigi" format. This recorder is used extensively in the broadcast and recording studio areas [14]. Data are recorded at either 44.1, 48 or 96 k.samples/sec. on 8 tracks with either 16 or 20 bit quantisation. Dash-format products are also used in the professional field.

The data rate for recording video signals digitally (DVTR) is of the order of 150 Mbits/sec. and much higher for high definition television (HDTV). To achieve this and still maintain a reasonable playing time, slant-azimuth recording techniques are adopted with multiple heads used in the helical scan mode. To date the technical problems of DVTR places it in the development stage, however a number of systems have been demonstrated and formats proposed [15] [16] [17].

1.2.2. Instrumentation.

Tape recorders utilising both rotary and stationary heads are used extensively in applications where the requirement is to log data. Rotary-head recorders are now being used in flight-borne recording. Considerations here are for low volume and ruggedised operation. Transverse scanning of the tape may be used to fulfill these requirements [18]. Open-reel recorders are also used in
military and aerospace applications where the high system costs are justified by the very high performance specification.

Terrestrial applications of tape recorders are too numerous and diverse to list [19] [20]. Open reel and closed reel stationary head systems are common. These are invariably multitrack/multi-channel instruments. At the lower end of the market tape cartridges are used. These offer convenience and portability in addition to low cost. Compact cassette also offers these advantages. Compact cassette instrumentation tape recorders are usually custom developed for a specific application. These are examined in more detail in section 1.4.

1.2.3. Computer Storage.
At the professional end of the computer market 1/2 inch tape in open reel format is used to record data at up to 6250 bits/inch. Sophisticated tape transports are employed with vacuum columns buffering the tape reels from the tape head to permit high tape acceleration and low access time. The trend towards disk-based computer systems at both the high and low ends of the market has stimulated the need for tape back-up. A common technique is to use a single-track stationary head and record a number of tracks in serpentine fashion. At the end of each tape pass the tape head is moved transversally across the tape. Tape speed may be adjusted for either a high data rate or a long recording period. Track data rate at high tape speed is 48kbits/sec, giving a storage capacity of 5.8 Mbits/track. At the time of writing the R-DAT format is being developed as a computer storage device with a potential storage capacity of 2 Gbytes [21].
Computer-generated digital data can also be stored on compact cassette. This form of secondary storage was widely used with the first generation of personal computers and although largely superseded by disk it still forms the bulk-storage requirements of many operational systems.

1.3. COMPACT CASSETTE.
Although the compact cassette tape system was originally conceived as a consumer product it can also be found in the professional recording area. Because the tape is protected inside a cassette, problems due to tape breakages and tape threading are eliminated. The format is robust, simple to use and economical to produce. Since its introduction it has gained a high degree of acceptance not least of all because of its ability to reproduce analogue signals of high quality.

The above features of the compact-cassette tape system also make it an attractive medium for recording digital data. However, the system requirements for digital tape recorders are more stringent than those of analogue recorders.

1.3.1. Compact - Cassette Format.
The compact - cassette tape format figure 1.8. was developed by the Philips company and accepted as an international standard in 1963 [22]. The three most popular tape lengths are C60, C90 and C120 where the numbers relate to the total playing time in minutes (on the assumption that the tape direction is reversed). The standard tape speed is 4.75 cms/sec. The size of the tape reel in each case is the same, therefore the thickness of the
Fig. 1.8. Compact-cassette tapes.
tape varies to accommodate the different playing times. The thickness of the polyester backing for C120, C90 and C60 tapes is 6μm, 7μm and 13μm respectively. Both the C90 and C60 tapes have a magnetic coating of 5μm whilst that of C120 tape is 3μm thick. During production the tensile strength of the tape is increased by stretching. Even so, there remains a significant difference in the transport properties of the tapes with the C60 tape exhibiting better performance compared to the others—especially the C120.

The cassette-track geometry is shown in figure 1.9. With a track

<table>
<thead>
<tr>
<th></th>
<th>max</th>
<th>min</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0.66mm</td>
<td>0.56mm</td>
</tr>
<tr>
<td>b</td>
<td>2.00mm</td>
<td>1.80mm</td>
</tr>
<tr>
<td>c</td>
<td>2.70mm</td>
<td>2.40mm</td>
</tr>
<tr>
<td>d</td>
<td>3.81mm</td>
<td>3.66mm</td>
</tr>
</tbody>
</table>

Fig.1.9. Compact cassette track dimensions.
width of 0.66 mm a signal-to-noise ratio of approximately 58 dB is possible. The original concept was to accommodate four tracks across the width of the tape. This number may be exceeded, however. Crucial factors to be considered when increasing the track density are the reduction by 3 dB in signal-to-noise ratio when track width is halved, intertrack crosstalk and the problems associated with sensing all the tracks simultaneously.

The cassette is usually constructed from plastic. Metal-housed tapes are also available but at a higher cost. The tape reels are held in position by the sides of the casing which form bushes at the centres of the two reels. These bushes permit a high degree of radial movement of the reels within the casing. Axial movement of the reels is restricted by the sides of the casing and by plastic shims positioned each side of each reel. A single pressure pad is located on the inside of the tape opposite the tape-head position. Since the tape drive force is low it is essential that the tape/casing friction is minimal.

1.3.2. Tape Coating.
Apart from their playing times cassette tapes are categorised in terms of the coercivity of their magnetic coating. The magnetic coatings range from the low coercivity gamma ferric oxide, Fe$_2$O$_3$ to chromium dioxide, CrO$_2$ and latterly the high coercivity metal particle tapes. The high coercivity tapes exhibit higher output and improved frequency response. CrO$_2$ tape, for instance gives a SNR improvement of 10-12 dB, compared to Fe$_2$O$_3$. However, CrO$_2$ tape is more expensive and its abrasiveness is greater than Fe$_2$O$_3$ tape.
The doping of oxides to improve magnetic tape characteristics has resulted in a range of tape coatings. Moir[23] has reported that, on the basis of their bias requirements, tapes can be divided into five classes. These range from ferric oxide coatings, requiring the minimum bias through to metallic tapes which require the maximum. This range also spans the coercivity of available tapes. Although bias is not used when saturated recording techniques are employed, the bias requirements do give an indication of the type of compact-cassette tape available.

Metal evaporated tapes (MET) promise to offer improved performance compared to conventional roller-coated tapes [24]. One advantage of MET is that because the magnetic layer is much thinner than conventional tapes a greater tape length may be accommodated in the compact-cassette format. However, MET is not compatible with existing tape recorders and its incorporation into the compact cassette format will depend on the provision of tape decks with the appropriate head design and bias arrangements.

1.3.3. The Compact-Cassette Channel.

The compact-cassette tape is a low-cost, high-quality product which has yet to be fully exploited as a high-density, direct digital recording medium. Mallinson has forecast that the limit on areal density of magnetic recording systems will depend on head signal-to-noise ratio and that densities of the order of several hundred megabits per square inch are possible [25]. This view is corroborated by Lemke who predicts future magnetic recording densities of up to 300 megabits per square inch [26].
Before these theoretical targets may be approached, however, a great number of mechanical problems would need to be solved to realise a viable system.

1.3.4. Frequency Response

The analogue bandwidth of the compact cassette system is a function of tape coating, tape speed and read-head gap length. A typical response for an inductive head is shown in figure 2.8. In analogue cassette recorders the frequency response of the read-back channel is equalised by passing the replayed signal through a filter with an inverse response, figure 2.14. The minimum of the filter characteristic depends on the type of tape used; for ferric tapes it is 1350 Hz. and for chrome tapes it is 2250 Hz. Because the human ear is insensitive to phase difference between the frequency components of a complex signal the non-linear phase response of the system implied by the bandlimited response of the channel, is not equalised in tape systems intended for analogue recording.

The operational bandwidth of a tape recorder may suffer degradation as a consequence of two factors - tape/head separation and tape azimuth misalignment. If tape/head separation increases during play the amplitude of the readback signal will fall; this is classified as a dropout should signal loss be 75% (12 dB) or greater [27]. The relationship between signal loss and tape/head separation is given by,

\[ \text{signal loss} = 54.6 \, \frac{d}{\lambda} \, \text{dB} \]

where \( d \) is the tape/head separation and \( \lambda \) is the wavelength of the recorded signal. This separation may be caused by tape-
surface asperities or tape debris/dust accumulating on the head surface.

If the longitudinal axis of the tape is not perpendicular to the read-head gap the effective length of the gap is increased. This tape azimuth misalignment increases the "gap loss" and the response of the system is reduced. The amplitude level of the signal also falls, the loss being given by the expression,

$$20 \log \left( \frac{\sin \left( \frac{\pi \omega \tan \alpha}{\lambda} \right)}{\frac{\pi \omega \tan \alpha}{\lambda}} \right) \text{dB.} \quad \ldots \ldots \quad (1.2).$$

Where $\omega$ is the angular frequency of the recorded signal, $\lambda$ is the wavelength of the recorded signal and $\alpha$ is the azimuth angle.

1.4. COMPACT - CASSETTE STORAGE SYSTEMS.

Although primarily designed to store analogue audio signals, the compact cassette can be used to store digital signals. The techniques employed fall into one of three categories: the unmodified audio compact-cassette recorder may be utilised where the digital data are stored as analogue signals; the same recorder may be used with direct-digital recording codes (in this case it may be necessary to use some form of signal equalisation); the audio amplifier and bias arrangements of the standard cassette player may be dispensed with and data applied directly to the record head via a write amplifier designed to handle the direct recording codes involved. In each of the above cases encoding and decoding circuitry must be provided.
The first generation of home personal computers utilised the compact-cassette tape recorder as a secondary store. They incorporated the necessary encoding/decoding circuitry through which data were recorded onto a low-cost, portable cassette player. This provided an inexpensive back-up facility for the computer's random access memory (RAM). Also, commercial software was made available on compact cassette tape. The recent trend in secondary storage for home computers is, however, away from cassette and towards disc and many current home computers are not equipped to interface to compact cassette tape.

Compact cassette recording systems are also used in the field of data logging. Here, the qualities of the format, such as low-cost, robustness and convenience, are ideal for on-site instrumentation application.

In the medical field, development of off-line computer diagnostic procedures [28] has generated the need to store vast quantities of data. Magnetic recording systems are used extensively to fulfill this need [29]. Storing signals such as from electrocardiograms (ECG), blood pressure, and heart rate.

The reliability and portability of compact cassette systems have been exploited in remote data-logging applications. Time-dependent data such as electrical power load variations [30] and road traffic flow can be logged [31] using either remote control or timed recording.

A number of attempts have been made to utilise the standard cassette recorder as a direct digital storage device [32-35].
In the main these have been either experimental or specialised applications. The compact cassette format is however, used in some commercial applications. One such application is as a tape streamer. Tapes used in this application are designed to operate at higher speeds than the standard compact cassette and are usually more expensive.

The three methods of recording digital data using compact-cassette equipment will now be considered in more detail.

1.4.1. Analogue Recording of Digital Signals.

With this technique digital data are encoded in the form of linear, analogue signals. Biased recording is used and the amplifier channel of the recorder is utilised. Using frequency shift keying (FSK), figure 1.10, digital data are recorded onto a single track. A common format is the Kansas City standard [36].

```
1 0 1 1 0
```

Fig. 1.10. Frequency shift keying (FSK) modulation.

With this format a logical 0 is recorded as 4 sinusoidal cycles at 1200 Hz with 8 cycles at 2400 Hz representing logical 1. This
gives a data rate of 300 bits/sec and a capacity of 2 x 540 k.bts for a C60 tape.

Data are encoded in Kansas City format by switching the frequency of an oscillator with the two logic levels which are to be encoded. Since the amplifier channel of the cassette tape recorder is used the output level of the oscillator must be set to a level sufficient to withstand subsequent amplification.

The decoding circuit comprises a frequency discriminator followed by a comparator.

The encoding/decoding circuits are simple and are usually situated within the personal computer (PC) with which this form of recording is used.

Error detection is employed which takes the form of inserting a cyclic redundancy check character (CRCC) at the end of a block of data from which the CRCC is calculated. On replay the PC associated with the recording system computes the same CRCC from the data block. This is compared to the recorded checksum and a match signals acceptance of the data block. Failure to match the CRCC requires the tape to be rewound and replayed, usually after readjusting the amplifier level of the recorder.

The Kansas City format was designed for use with home computers and as such is robust and inexpensive to implement. Bit synchronisation is not required and the two signalling frequencies are sufficiently far apart to give a tape speed tolerance of 30%. The system is also reasonably immune to tape
dropouts. In the event of loss of input the frequency discriminator "free wheels" and so the loss of a few cycles of signal can be accommodated. One problem associated with the system is that, unless care is taken during the recording process, the record amplifier overloads and the ensuing distortion causes errors which are detected on subsequent playback. The encoding/decoding circuitry is specific to this application and therefore has to be provided by the PC manufacturer in addition to the usual serial/parallel, input/output interface chips. Also, tapes must be erased before recording.

Various versions of the above frequency shift keying (FSK) recording method give data rates up to 2400 Baud. In one popular microcomputer logical '0' is represented by one cycle of a 1200 Hz. sinusoid and logical '1' by a frequency of 2400 Hz. [37], giving a data rate of 1200 Baud.

1.4.2. Linear, Direct Recording of Digital Signals.

There are a number of examples of utilising the unmodified compact cassette analogue recorder to record digital data directly. In one application the digital waveform of data in ASCII form is first differentiated before application to the standard recorder [38]. Peak detection is used on playback with the tape replayed at a higher tape speed. Data are recorded on a single track using a two-track head. The second track is used to record a clock signal giving a system which is independent of tape velocity. This system achieves an error rate of 1 in $5 \times 10^5$ at 750 bits per second.
Smith and Zorkoczy [39] have used an audio cassette recorder as part of an Open University radio text project. Computer software is broadcast via VHF transmission to the homes of students where it is recorded on a low-cost audio cassette recorder. The recording code Biphase-M is used and a data rate of 2400 Baud is achieved. The signal is encoded onto a single track by a simple hardware encoder. Decoding is accomplished using a phase-locked loop synchronised to the data. This provides a clock which is counted between transitions of the playback signal. The number of clock pulses thus recorded determines the value of the decoded signal. The linear channel of the recorder is used with its inherent frequency equalisation. Additionally, phase equalisation is applied to the replay signal to increase the opening of the replay-signal eye pattern. Although no quantitative data on the error performance is given the authors report excellent performance of the system using a range of tapes.

The above system offers a moderately high data rate at a low cost. The use of a more expensive "hi-fi" tape recorder together with a higher sophistication of hardware can yield a system with higher data rate.

Ewins [31] has used a high-quality cassette tape recorder as the basis for an instrumentation-standard data logger. The linear record/playback amplifier of the recorder has been retained enhanced by tape-speed control circuitry. Miller code is used to store data at a total data rate of 22,733 k.bits/sec. Data are recorded on two tracks using the standard 2-track head of the stereo cassette recorder.
In the simple system of Smith and Zorkoczy error detection methods are not applied. Ewins does employ simple parity error detection but in both cases a quantitative assessment of the error performance of the systems is not given—this is unfortunate. The performance criteria of recording systems vary but as a general rule the objective is to maximise the data rate (and thus the longitudinal packing density) and to minimise the errors which are observed on playback. An approximate figure of merit for such recording systems would be the data rate/error rate ratio. This, of course is only a guide to the effectiveness of a system. Other factors would have to be taken into account, not least of which would be the complexity of the implementation.

Although still retaining the use of a high-quality cassette tape recorder Kageyama et al [40] have proposed the application of partial response signalling techniques in recording digital data. A data compression technique is used to store up to 300 still—colour picture frames on a C90 compact cassette tape. The application of \((1,0,-1)\) class IV partial response signalling permits data to be stored on a single track. With data recorded in blocks of 136 bits the block error rate is less than \(10^{-3}\). This error rate is achieved by applying error correction techniques. A 16-bit cyclic redundancy character (CRC) is applied to each block of 132 bits and interleaving to a depth of 4 blocks is used. In addition a parity word is appended to every 9 blocks to realise a coordinate error—correction scheme.

Using the figure of merit criterion mentioned above the data rate/error rate ratio is \(2.1 \times 10^7\). The nett error rate is 33.
relatively high, however this is consistent with the high data rate achieved. The process of reproducing visual images from digital data is relatively error tolerant and the pursuit of a high data rate at the expense of error rate is justified in this case. This figure of merit is higher than that obtained by Ewins [31] which at 1.5 x 10⁷ is two orders of magnitude lower. However, the sophisticated signalling technique adopted by Kageyama and the complexity of its implementation must also be considered when comparing the two systems.

The recording and reproduction of still-picture data is an emerging field. Kihara [41] has reported on the prospects of an electronic still-picture camera and recording system and Konica [42] have demonstrated the operation of a "35mm" still camera with integral 2 inch floppy disc capable of storing 50 still colour pictures. Also under development is a WORM compact disc system capable of storing 800 still, colour pictures [43].

Although disc technology is used to store the images, the author envisages a demand for low-cost, tape back-up once the technology becomes available as a consumer product.

The advantages of adapting the conventional audio cassette tape recorder to record digital data include inherent frequency compensation and user familiarity. However, since the recorder was initially designed for recording analogue signals this approach could lead to a sub-optimum system. A number of attempts have been made to develop a high-density direct recording system.
using the compact - cassette format without the attendant "hi-fi" electronics.

1.4.3. Saturated, Direct Recording of Digital Signals.
Early work on the use of a compact - cassette system as a direct recording device in its own right is due to Weiler [33]. In a comprehensive investigation Weiler assessed performance by measuring bit error rate against data rate for the three recording codes, ENRZ, Miller and Miller squared. A read/write head with a gap width of 3 \( \mu \text{m} \) was used and data were recorded on a single track. It is not clear what the track width used was, the implication is, however, that the head spanned the full width of the tape. The results are shown in figure 1.11. The three coding techniques show equal performance up to 9kbits/sec, after which there is a rapid deterioration in bit error rate with Miller squared code and ENRZ code outperforming Miller. The data rate is about one half that obtained by Ewins and Kageyama, the figure of merit is, however, at 1.1 \( \times 10^4 \) some 50 times better than that of Kageyama.

Weiler uses a transversal filter as a pulse slimmer. The slimmed pulses are then sliced in a circuit which employs decision feedback to equalise the low frequency content of the signal. The decoding circuitry is clocked from a phase-locked loop synchronised to the replayed data.

The recording systems discussed so far give varying degrees of performance at a cost of given complexity. One feature they all share is that in each case a single track is used to record data;
Fig. 1.1. Bit error rate against data rate for Miller, Miller squared and ENRZ codes. Due to Weiler [33].
no attempt is made to utilise the potentially high areal density which tape offers by employing multitrack techniques.

1.4.4. Multitrack Recording Systems.
Digital data are normally generated in parallel. To store these data in single-track format requires that multiplexing techniques be employed. Similarly, the data must be de-multiplexed when they are played back. To utilise fully the potentially high areal density offered by tape a multitrack format can be used. This requires the operation of $n$ channels in an $n$-track system, however, in some cases these $n$ channels may be handled simultaneously by computer.

Sakamoto et al. [44] have applied a multitrack format to compact-cassette tape. A total of 24 tracks are arranged across the width of the tape, figure 1.12. To provide a reversible format 12

![Diagram of compact-cassette track format](image)

Fig.1.12. Compact-cassette track format due to Sakamoto et al [45].
"forward" tracks are arranged in the guard space of 12 "reverse" tracks. The spacing between tracks is 35 μm giving a track width of 120 μm. A single, ferrite inductive head is used and the data are recorded on metal-evaporated tape. The base-layer thickness of the tape is the same as normal C90 tape. The magnetising layer thickness is, however, much less and although the tape speed is double that of a normal cassette system, the playing time of the tape is one hour.

This prototype system is intended for the recording of "hi-fi" digital audio data. A single-track recording rate of 70.56k bits/sec. is claimed using four-to-eight modulation (FEM) [45] and signal equalisation. No error rate performance is given to complement this very high data rate.

A feature of this recorder is the extensive tape-guidance system adopted, figure 1.13. Four "precision" tape guides are arranged

---

Fig.1.13. Modified tape guidance system for Sakamoto's cassette recorder.
each side of the head. A dual capstan moves the tape across the
head, a further four guides are located within the tape housing.
This latter arrangement implies the modification of the normal
compact cassette to suit the extensively re-modelled tape guide.

The function of the tape guide is to maintain a constant tape
tension and to reduce the effects of tape skew. The authors
report that skew effects are held to within ±10 - 15 μm by the
guide. At the frequency of data recorded FEM gives a minimum
wavelength of 2.1μm suggesting a bit - cell displacement across
the tape of 5 to 7 bits due to skew. Although no mention is made
of skew correction it must be assumed that some means of tape-
skew correction is employed.

1.5. MECHANICAL PERFORMANCE OF COMPACT CASSETTE TAPE TRANSPORT.
The compact cassette tape has been developed as a low - cost,
robust item. This philosophy is also reflected in the design of
the tape transport mechanism on which the tape is played.

Cassette tape transport mechanisms are designed for analogue-
encoded tapes and range from the basic, portable player to the
more expensive deck found in "hi - fi" equipment. However, even
the mechanisms at the top end of the market are basic when
compared to the professional reel - to - reel machines designed
for broadcasting, instrumentation and computer storage.

Problems associated with transporting the tape uniformly across
the write/read head include tape - velocity variations, tape
azimuth variations and maintaining a good contact with the head.
Although to a great extent the above problems are tape dependent
careful design of the transport mechanism can reduce their effect.

1.5.1. Tape - Velocity Variation.
The quoted tolerance on the 4.75 cm./sec. velocity of the compact cassette tape transport ranges from $\pm 0.5\%$ to $\pm 2\%$ depending on the quality of the deck. These values are aimed for through the application of speed control to the motor driving the capstan. It is, however, the relative tape/tape-head velocity which needs to be constant and this will only be so if the constant speed of the motor is transmitted to the tape/tape-head interface by constant tape tension. Actual velocity variations therefore depend on the tape used. In practice this velocity variation often falls outside the quoted tolerance [46] causing problems when digital data are recorded. These problems are compounded by static friction (stiction) which is an inherent problem with slow-moving in-contact bodies. This latter velocity characteristic is in large part responsible for the jitter in the readback signal as the tape accelerates / decelerates across the read head.

1.5.2. Tape - Azimuth Variation.
The basic nature of the tape guidance of the compact cassette tape transport coupled with the less-than-ideal tape-edge straightness causes tape azimuth variation. This results in a reduction in the high frequency content of the reproduced signal. Additionally, if the system is multitrack and digital data are recorded broadside across the tape, mis-detection may result as
data bits on the outside tracks are time displaced as they pass their respective read heads.

Azimuth variation may be compensated for by dynamically rotating the read head by an amount equal to the azimuth angle [47]. The head is fixed to a piezoelectric element which is pivoted about an axis in line with the head gap. The amplitude of the readback signal is monitored and controls the pivotal angle of the head via the piezoelectric element. The whole device is small enough to fit inside the read head and it has been used on a high-end production compact cassette tape deck. Other methods of azimuth compensation have also been proposed which are similar [48 - 50].

The effects of azimuth variation on multitrack, digital recordings may also be reduced by clocking the time-displaced data off each track and reintroducing the correct timebase offline (chapter 2). This method is employed in stationary-head, multitrack recorders and although it is effective it does require a battery of switchable data buffers. Perhaps the main disadvantage is the need to format the tape with control data to provide switching signals.

1.6. ERROR CORRECTION
Attempts to increase recording density are accompanied by a reduction in SNR and/or detection window time, with a consequential adverse effect on error rate. To combat this, error-correction schemes are often applied. These displace data by the introduction of redundant information into the recorded data stream - usually to an extent proportional to the error -
correcting power of the scheme. Error correction schemes inevitably increase the complexity of a recording system and their implementation involves a trade-off between the factors mentioned above. This section covers error correction methods suitable for tape systems.

1.6.1. Simple Parity

Error correction schemes which use simple parity are suitable for those applications where the requirement is for minimum delay in the correction of replayed data. Co-ordinate error correction schemes which use cyclic redundancy check characters to flag a track failure and simple parity checks across the tape to identify the failed track are simple to implement. Correction time is minimised in view of the simple parity equations which need to be solved to identify and correct replayed data. Some multitrack failures may not be corrected by this technique, in these cases it is often possible to signal track failure. Problems associated with simple-parity error correction include the need to position check characters "in-line" with the data at regular intervals. Not only does this displace data but also the detection circuitry must differentiate between data and parity bits on the same track. This latter restriction is overcome by rotating the co-ordinate axes of the parity elements by 45 degrees and locating them both on dedicated tracks. In doing this the CRCC, which normally covers long strings of data, can be replaced by a simple parity bit which is adequate for the shorter data string occupying a 45 degree line across the tape. This further decreases the time required to process the
corrected data. The above cross-parity error correction scheme has been applied by Patel and is treated in section 3.2.1.

The incorporation of interleaving to disperse error bursts significantly enhances the error correcting power of simple parity systems. However, although interleaving reduces burst errors to random errors, isolated errors which occur adjacent to the bursts will adversely affect performance as they could be de-interleaved with the bursts error to give multiple errors in one codeword. This effect is nullified by the use of cross interleaving techniques [51]. Cross interleaving is illustrated in figure 1.14. Parity check characters are generated for each codeword before and after interleaving, P and Q respectively. P and Q cover data organised in rows and diagonally respectively. Multiple errors occurring in rows are treated as simple diagonal errors and vice-versa.

Cross interleaving is utilised in the DASH tape recorder format and it is also applied to the compact disc where the Reed-Solomon code is used to generate the parity check characters.

1.6.2. B.C.H. Code

The more sophisticated error codes such as the Bose, Chaudhuri-Hocquenghem codes offer lower redundancy and greater error correcting power. These advantages, however, must be contrasted against the complexity of their implementation. Fast decoding may be achieved with the use of hardwired decoding circuitry but multitrack recorders could require either multiple circuitry or the use of multiplexing techniques. The flexibility of the

A microprocessor has been used to simplify hardware complexity [52] but program execution time is a limiting factor. Shayan et al [53] have reduced the software decoding time for BCH codes by implementation of a fast Chien search algorithm. This reduces the search time by 50% to improve the overall decoding time by 4/3.
1.6.3. Intelligent Methods.

The data-packing overhead incurred by the inclusion of error detection/correction ranges up to 100% and more, depending on the application. In such schemes not only are data displaced by parity bits but the cost of implementing parity-generating circuitry must also be borne. However, the detection of errors in reproduced data is possible through the intelligent interpretation of the replayed data sequence. In these schemes the packing density overhead for error detection is zero.

In certain, run-length limited channel codes the size of the intervals between transitions is fixed by the encoding algorithm. Variation of interval size from that determined by the encoding rules can then be used to flag errors. In Miller code, intervals of two, three and four clock periods occur between transitions (this assumes a double frequency clock). Csengery [54] has noted that the number of "three" intervals between any two "four" intervals must be even, figure 1.15. Also, a "two" interval changed

![Miller-code sequence showing even number of "three" intervals between two "four" intervals (double frequency clock assumed).](image)

45
to a "three" interval by error, or vice-versa, will add or subtract one "three" interval from the correct number. In noting the occurrence of interval size on playback a parity of "threes" can be implemented. Csengery has used the above Miller-code characteristic to detect track errors in a multi-track tape recorder. One track is used to record a conventional parity bit covering the data across the tracks to form a coordinate error-correcting system. An improvement in error rate of greater than an order of magnitude is claimed for a data-packing overhead of 10%.

1.7. PURPOSE OF THE INVESTIGATION.

In terms of areal density, tape systems are currently the most cost effective for recording digital information. However, because of the need to transport the tape over the read/write head with precision, a high proportion of the cost is in the mechanical components. Costs may also be held high by the requirement to provide each track of a multitrack recorder with the appropriate encode/decode, error correction and clock electronics.

Previous attempts to utilise a downgraded tape transport and tape, such as the compact-cassette system, result in a proportionate downgrading in system performance. If system performance is maintained it may well be at a cost of increased support-electronics complexity and/or improved tape-transport mechanics.

It may be possible to compensate for the mechanical vagaries of
a downgraded system by employing intelligent techniques. This would require the application of programmable electronics which, in addition to providing the required flexibility of response, could also assume the functions associated with hardwired circuitry, such as, channel encoding/decoding, error correction, data synchronisation etc. Such a system would be inexpensive, robust and find application across the range of uses considered in this chapter.

The purpose of this investigation is to apply intelligent software techniques to low-cost, compact-cassette mechanical components with the aim of producing a high-density digital recording system. Implicit in this aim is a comprehensive knowledge of tape-transport characteristics which have not previously been determined in terms of recording digital data directly.
1.8. REFERENCES FOR CHAPTER 1.


CHAPTER 2.

CHARACTERISTICS AND ANALYSIS OF MAGNETIC RECORDING SYSTEMS.

2.1. THE RECORDING CHANNEL.

A block diagram of a basic digital data recording channel is shown in figure 2.1. In essence the objective is to economically communicate information from source to sink, over a noisy channel with as low an error rate as possible. In the context of magnetic recording the term noise embraces the gaussian noise introduced by the tape, tape head and interface electronics. Due regard must also be paid to time limitations of jitter and timebase variation.

Practical implementations of the above system vary according to application. Many basic systems do not incorporate error control or equalisation whereas channel encoding is an essential element. The specific requirement also dictates the complexity of each element employed since this can often be traded off against performance.

This chapter is concerned with the fundamental theory of each element in the recording channel. The first section considers the recording channel from an overall viewpoint.

2.2. CHANNEL CAPACITY.

The maximum rate, \( R \), at which information can reliably be transmitted through a channel is limited by the channel capacity, \( C \). For \( R < C \) information may be transmitted with an arbitrarily low error rate. The actual error rate is governed by the
Fig. 2.1. Basic digital-data recording system.
quality of the channel and the ingenuity of the coding scheme employed. Channel capacity was determined by Shannon [1] as,

$$C = B \log_2 \left[ 1 + \frac{P}{N_0 B} \right] \text{ bits per sec.} \quad \cdots \cdots \cdots \cdots \cdots \cdots (2.1)$$

where, $B =$ the channel bandwidth in Hz.

$P =$ the received signal power in watts.

$N_0 =$ noise spectral density.

$\frac{P}{N_0 B} =$ signal-to-noise ratio (SNR).

This shows that the bandwidth of a channel and the power SNR may be traded off against each other in achieving the desired capacity. Although equation (2.1) indicates that channel capacity increases with bandwidth it is in fact limited by the signal-to-noise ratio.

At channel capacity the bit period is $\frac{1}{C}$ secs. Thus, the energy per bit,

$$E_b = \frac{P}{C}$$

Equation (2.1) now becomes,

$$\frac{C}{B} = \log_2 \left[ 1 + \frac{E_b C}{N_0 B} \right] \text{ bits/sec.} \quad \cdots \cdots \cdots \cdots \cdots \cdots (2.2)$$

Two conditions pertain:

(a) The power limited case when $C < B$.

(b) The bandlimited case when $C > B$.

These are plotted in figures 2.2 and 2.3 respectively.

The "Shannon limit" gives the minimum value of the received energy per bit - to noise ratio theoretically possible to ensure reliable data communication. When only a limited bandwidth is available the required energy is greater, for the same data rate, and the two may be traded off against each other.

Although figure 2.2 gives the theoretical limits of possible
Fig. 2.2. $\frac{E_b}{N_0}$ against $C/B$ for $C < B$.

Fig. 2.3. $\frac{E_b}{N_0}$ against $C/B$ for $C > B$. 
communication systems the engineering challenge is to design viable and economical systems to come as close as possible to these limits.

2.2.1. Capacity of Multitrack Tape Recorder.

Equation (2.1) may be applied to a multitrack tape system. Consider a tape recorder with \( M \) tracks. The data capacity is increased by \( M \) but since halving the track width reduces the signal by \( 6\text{dB} \) and the noise by \( 3\text{dB} \), the effective SNR is reduced by a factor \( \sqrt{M} \), therefore,

\[
C_m = M B \log_2 \left( 1 + \text{SNR}/\sqrt{M} \right) \text{ bits/sec.} \ldots \ldots \ldots (2.3).
\]

Taking the power-limited case (figure 2.2). The "Shannon limit" is reached at a SNR of \(-1.6\text{dB}\). Using Hallinon's \([2]\) read-head noise figure of \(20\text{dB} \) as the dominant parameter in a recording system,

\[20\text{dB} / \sqrt{M} = -1.6\text{dB}, \text{ from which } M = 21,000!\]

This figure is well outside that considered practical with conventional magnetic recording methods.

Taking the more realistic case of a 4-track compact cassette recorder with a SNR of \(30\text{dB} \) and bandwidth \(10\text{kHz} \), the channel capacity \(C_m\) is,

\[
C_m = 4 \times 10 \times 10^3 \log_2 \left( 1 + 1000 / 2 \right)
\]

\[
C_m = 359 \times 10^3 \text{ bits/sec.}
\]

This gives a notional track recording rate of approximately 90 kbits/sec.

From the preceding section both the SNR and the bandwidth of a multitrack magnetic tape recorder are adequate to support a
theoretically high data rate. However, to record on a magnetic medium requires the use of a run-length-limited (RLL) code and this too impacts on the channel capacity. This aspect of channel capacity is treated in section 3.1.1.

2.3. THE RECORDING PROCESS.
A brief review of the fundamental process of recording digital data on magnetic tape is given. Also, mechanical aspects of tape transportation are considered.

2.3.1. The Write Process.
Maxwell's equations for magnetic media yield the following two equations,
\[ \oint \mathbf{H} \cdot d\mathbf{l} = I_{\text{total}} \quad \ldots \ldots \ldots \quad (2.4) \]
\[ \phi = \int \mathbf{B} \cdot d\mathbf{A} \quad \ldots \ldots \ldots \quad (2.5) \]
where the symbols have the usual meanings. In the ring type inductive head the high permeability magnetic material forms a small, non-magnetic gap across which the tape passes.

The fringing field close to the head is described by Karlquist [3] as,
\[ H_x = \frac{H_0}{\pi} \left[ \tan^{-1} \frac{g/2 + x}{y} + \tan^{-1} \frac{g/2 - x}{y} \right] \quad \ldots \ldots \quad (2.6) \]
\[ H_y = \frac{H_0}{2\pi} \ln \left[ \frac{(g/2 - x)^2 + y^2}{(g/2 + x)^2 + y^2} \right] \quad \ldots \ldots \quad (2.7) \]

For distances \( d > g \) a simpler form of equation may be deduced:
Applying equation (2.4) around the flux path, ignoring head-core reluctance, the deep gap field is,
\[ H_0 \approx \frac{NI}{g} \quad \ldots \ldots \quad (2.8) \]
where \( N \) = number of turns on the head,
\( I \) = current,
\( g \) = gap length.

Also, applying equation (2.4) through the fringing field of radius \( r \),
\[
H = \frac{NI}{\pi r} \quad \ldots \ldots \quad (2.9)
\]
From equations (2.8) and (2.9) the longitudinal, \( x \) and vertical, \( y \) components of the field may be deduced:
\[
H_x = Ho g y/\pi (x^2 + y^2) \quad \ldots \ldots \quad (2.10),
\]
\[
H_y = -Ho g x/\pi (x^2 + y^2) \quad \ldots \ldots \quad (2.11).
\]
\( H_x \) and \( H_y \) may be combined to give
\[
H = Ho g/\pi \sqrt{x^2 + y^2} \quad (2.12).
\]
Figure 2.4 shows the fringing field across the head gap. The tape is recorded as it leaves the field after the trailing edge of the head gap. The first quadrant of the field is called the record zone.

At high bit densities the bit length approaches half the record zone length, consequently, a recording loss ensues as the media particles of different coercivity are variously affected as they pass through the remanence gradient of the record zone, \( L \). As the tape/head separation increases, \( L \) increases and this results in an increased cancellation loss. Bertram [4] has shown that for high-density media this cancellation results in a record loss of,
\[
-44 \frac{d}{\lambda} \text{ dB} \quad \ldots \ldots \quad (2.13)
\]
where \( d \) = tape/head separation.
\[
\lambda = \text{twice the bit length.}
\]

In practice the demagnetisation field of the media elongates the
Fig. 2.4. Recording zone of record head.
record zone and effectively increases $L$. However, $L$ may be decreased by reducing the record-head gap $g$.

In saturation recording $H_x$ must switch the tape coating and therefore exceed the coercivity of the tape. In achieving this the recording - head material must not saturate. Since, to avoid stray demagnetisation, the coercive force of the tape cannot be too low, considerable care is required in tape-head design. As can be seen from equation (2.12) the strength of the field in contact with the tape can be increased by increasing the gap $g$. In a write - only head this is an option since the recording pattern on the tape is determined as the tape leaves the trailing edge of the record gap.

2.3.2. The Read Process.

The record head may also be used in the readback process. The voltage induced in the head winding is given by,

$$v = -N \frac{d\phi}{dt} \ldots \ldots \ldots (2.14).$$

The readback voltage is thus phase shifted relative to the recording current.

To achieve a high packing density the recorded flux transitions should be placed as close together as possible. Attempts to do this lead to peak shift or pulse crowding, figure 2.5. Excessive peak shift impairs the detection process since the detector assumes that magnetic transitions occur at the point where they were recorded. Peak shift also affects the amplitude of the playback signal. Dudson and Davies [5] have shown the relationship between packing density and output amplitude, figure
2.6. The packing density is shown in terms of the width of the isolated playback pulse at 50% of its amplitude (P50). In a study of peak shift, Chi [6] measured the effect of pulse crowding on the shift in peak position and the ensuing reduction in amplitude. A doublet test pattern was used and the equalised
results are shown in figure 2.7. Whilst the fall off in amplitude of the two peaks is the same the extent of the peak shift is different for both transitions.

Pulse-slimming techniques have been advocated to reduce P50 and so increase packing density; these are considered in section 2.5.1.

From equation (2.14), \( v = -j\omega N \) volts. If the medium passes an ideal tape head with velocity \( S \) and the wavelength of the recorded signal is \( \lambda \) then the frequency of the playback signal is, \( f = S/\lambda \), therefore,
Fig. 2.7 Effect of pulse crowding on readback pulse amplitude and position.
\[ v = -j k s N \phi \] where \( k = \frac{2\pi}{\lambda} \) \hspace{1cm} (2.15).

If \( H_f \) is the field entering the head then,
\[ \phi = \mu_0 H_f \frac{W}{jk} \] \hspace{1cm} (2.16),

where the flux area is given by the width of the track, \( W \), times \( \lambda / 2\pi \).

From (2.15) and (2.16),
\[ v = -s N W \mu_0 H_f \] \hspace{1cm} (2.17).

The significance of equation (2.17) for multitrack systems is the fall in replay signal amplitude of 6dB each time the track width is halved. Further, since track noise is random this reduces as the square root of track width, i.e., 3dB, hence the signal-to-noise ratio deteriorates by 3dB each time the track width is halved.

The frequency response of the replayed signal is given by,
\[ v(k) = K s k e \left( \frac{1 - e^{-k}}{k} \right) e^{-kd} \sin \left( \frac{1.11 \text{ kg/2}}{1.11 \text{ kg/2}} \right) \] \hspace{1cm} (2.18).

Where, \( \delta \) = thickness of magnetisation zone,
\( d \) = tape head separation,
\( K \) = constant
\( k \) = wavelength number, \( \frac{2\pi}{\lambda} \),
\( s \) = velocity and
\( g \) = gap length.

The effect of the various components of equation (2.18) is shown in the response curve of figure 2.8. From the null at d.c. the response increases by 6dB per octave. This rise is counteracted by the thickness-loss term as \( k \) exceeds 1.

Thickness loss is brought about by the different vector magnetisations of the tape-coating lamina. This is caused by the
Amplitude
Id
B l
Head -differentiation
gain.
6 dB per octave.
(ks term)

(spacing loss term $e^{-kd}$
attenuates the response
with increasing frequency.

thickness loss
term $\frac{1-e^{-k\delta}}{k\delta}$
attenuates the response
with increasing frequency.

gap loss term $\frac{\sin(1\cdot11\cdot11)}{111\cdot11}$
contributes to attenuation.

Fig. 2.8. Frequency response of reproduce head.
fringing field, figure 2.4. The lamina furthest away from the head receiving a magnetisation orthogonal to the nearest lamina with areas in between making the transition between these extremes. Because of thickness loss the read head only senses the magnetisation in the tape coating near the surface. Estimates of this effect [7] show that 90% of output reproduction comes from a tape-coating penetration of $\lambda/3$.

In a practical system the magnetisation thickness may be made less by reducing the recording current; this increases the frequency at which thickness loss takes effect. The spacing loss makes some contribution to the reproduction loss since, because of tape-surface roughness and imperfections in the head profile, there is always a finite, but small, tape-head separation. The gap loss follows a $\sin x/x$ law which rapidly reduces the response to zero as the wavelength becomes approximately equal to the gap length.

Unlike the record process the gap length of the readback head must be as short as possible. The foregoing to place the first zero in the frequency response as high as possible.

The foregoing theory applies to conventional ring inductance heads. Reproduce heads fabricated from magneto-resistive (MR) material may also be used [8 - 15].

One advantage of MR heads is that their response is media/head-velocity independent. Also, their sensitivity is much higher than conventional inductance heads. MR heads do, however, require biasing and head wear can be a problem.
2.4. MECHANICAL CONSIDERATIONS.

The aspects of magnetic tape recording so far considered assume ideal tape/head contact. In practice, the transportation of the tape across the head is far from ideal and aspects of this non-ideal behaviour will now be considered.

2.4.1. Tape - Velocity.

The frequency, $f$, of the reproduced signal is related to tape velocity, $s$, through the equation,

$$f = \frac{s}{\lambda},$$

where $\lambda$ is wavelength.

At a given frequency the length of tape occupied by one wavelength depends on the tape velocity. Should the velocity profile of the tape replay not be identical to that on record then the reproduced frequency will be different. The velocity of compact-cassette tape is standardised at 4.75 cm./sec. The tolerance on this speed is $\pm 0.5\%$ for professional recorders and $\pm 2\%$ for domestic players [16]. These close tolerances are usually achieved through the application of speed control circuitry. Whilst the above velocity specification applies to the motor driving the tape it is the velocity of the tape across the head gap which determines performance.

The effect of tape-velocity tolerances of magnitude $0.5\% - 2\%$ on digital recording would be negligible. Because of the low velocity of the tape, however, stiction is a problem. This causes flutter the effect of which is to introduce jitter into the playback signal. Jitter reduces the timing window during which the playback pulses may be sampled, figure 2.9.
2.4.2. Tape/Head Separation.

Equation (2.18) contains a spacing loss component which allows for the reduction in head output due to the distance between the tape and the head gap. The spacing-loss term gives an attenuation of:

$$20 \log_{10} e^{-kd} = 54.6 \text{ d/B dB}.$$  

A head separation of some 22% of the recorded wavelength is all that is required to produce a dropout (-12dB loss). This translates into a tape/head separation of 2 μm for a bit rate of 10 kb/sec at compact cassette speed. The surface smoothness of audio tape is in the range 0.25 to 0.12 μm. [17] and the worst-case contribution to signal loss from this source is <1 dB at 10 k.b/sec.

At high bit densities account must also be taken of the record loss which is also a function of tape/head separation (equation 2.13). This combines with the spacing-loss component to give a total signal attenuation of 94.6 d/B dB. The tape/head separation

---

Fig. 2.9. Reduced timing window due to jitter tj.
now required to bring about amplitude loss of dropout proportions falls to 12.7% of the recorded wavelength.

The major cause of spacing loss is the build up of oxide and debris on the tape head. This cause of signal loss has significant implications for the detection method employed. Its effect may be reduced through the use of automatic gain control (agc) in the readback amplifier, although agc degrades the signal-to-noise ratio.

2.4.3. Tape - Azimuth.

In the gap-loss component of equation (2.18) a constant gap loss is assumed. In practice an azimuth, or skew, angle occurs between the normal to the longitudinal direction of the tape and the head gap. The result is to effectively increase the gap length (figure 2.10) and consequently lower the first zero in the

![Diagram of skew angle and effective head gap](image)

Fig. 2.10. Skew angle increases effective head gap for track width w.
replay response. The gap-loss component now becomes,

\[
\sin \left[ \frac{1.11 \times 2\pi}{2\lambda} \left( W \tan \alpha \right) \right] \quad \text{which results in an attenuation of approximately,}
\]

\[
20 \log \sin \left( \frac{\pi W \tan \alpha}{\lambda} \right) \quad \text{dB. .... (2.20).}
\]

This loss component is shown in figure 2.11 for a compact-cassette, 4-track format (track width 0.66 mm.) at a bit rate of 10 kb/sec. The dropout level is reached at a tape/head mis-alignment of 0.65 degrees.

Fig. 2.11. Azimuth loss and bit-cell displacement against azimuth angle for compact-cassette tape at 10 kb/s.
Tape azimuth loss is a function of track width - the narrower the track width the less is the effect. It also reduces as the tape velocity is increased since this gives a higher wavelength for a given frequency. In addition to the adverse effect of tape/head alignment on amplitude response is its effect on multitrack digital recordings. This is illustrated in figure 2.12. A tape azimuth angle of 1° causes a track-space displacement of 66.5 μm across the tape. The implication of this for a recorded data rate of 10 k.bits/sec/track is that a bit-cell displacement of 14 bits will occur between the outside tracks. The results shown assume a constant tape-azimuth angle across the width of the tape.

![Fig. 2.12 Effect of tape skew on digital recording.](image)
Two components of tape azimuth can be identified - static skew and dynamic skew. Compact - cassette tape heads are fitted with a single tape guide. This, coupled with possible axial movement of the tape reels, permits the tape to lie across the tape head with a skew angle $\neq 0$, static skew. In addition to this static skew the skew angle also varies under dynamic conditions. A major contribution to dynamic skew is the tape itself. During manufacture magnetically coated sheets are slit into tape. During this process a wavy edge can be imparted to the tape and as a result the tape "weaves" through the tape guide as it is transported causing skew - angle variation. A curvature of up to 3 mm in 1 metre of tape length is possible [17].

The effect of tape - skew variation on the coherent detection of digitally recorded signals may be removed by employing data buffers at the outputs of each track, figure 2.13. The data from

![Diagram of deskew circuitry for multitrack tape recorder](image)

Fig.2.13. Deskew circuitry for multitrack tape recorder.
each of the tracks are clocked independently into their respective buffer. When all buffers 'A' are full the track data are clocked into the 'B' buffers whilst the data from the 'A' buffers are clocked out uniformly. The length of the buffers is equal to the maximum bit-cell displacement due to skew.

2.5. CHANNEL EQUALISATION.

The mis-match between the frequency response of the recording channel and the frequency spectrum of the signal to be recorded imposes a limit on packing density. As the recording rate is increased intersymbol interference (ISI) and peak-shift effects reduce the opening of the detected eye pattern to inhibit data detection. These effects may be reduced by modifying the frequency and phase response of the recording channel to accommodate the spectral components of the recording code.

Channel equalisation may be effected during the replay process, post equalisation, or during the recording process, pre-equalisation.

2.5.1. Post Equalisation.

Frequency equalisation may be applied by placing the recording channel in series with a filter having a response which is the inverse of the channel, figure 2.14. In addition the phase response of the channel should be equalised proportional to frequency.

Selective frequency equalisation may also be applied. With an inductive head a d.c. null occurs. If the recording code to be employed is not d.c.free it may be necessary to compensate for
Fig. 2.14. Frequency equalisation of recording channel.

The null. The basis of d.c. equalisation is to re-introduce the d.c. content on playback.

Conventional d.c. restoration techniques can be applied but the variable mark space ratios of the playback waveform, coupled with amplitude variation, causes base-line wander and so impairs the detection process. Huber [18] addresses this problem by d.c. restoring both the positive and negative waveform peaks and summing the two signals. This fixes the base-line of the signal thus permitting peak detection. Equalisation of the d.c. null using decision feedback has also been proposed [19].

ISI is caused by the spectral response of recorded pulses overlapping into the time zone of adjacent pulses. The amplitude of these interfering pulse "tails" will reduce rapidly if the
channel is equalised to a raised cosine response. Kamayama et al [20] has applied raised cosine equalisation to realise an improvement in packing density of 30% compared to the unequalised channel.

The theory of superposition permits the recorded pulses to be slimmed during the replay process and so increase the packing density. This may be accomplished by adding a differentiated component of the replayed waveform to itself. Schneider [21] has proposed a pulse slimming technique which permits the leading and trailing edge of a replayed pulse to be slimmed independently. Results show pulse-slimming ratios of 3:1 with possible reductions of up to 10:1 with cascaded sections.

One advantage of pulse slimming is that both frequency and phase are equalised in one operation. A disadvantage is that each time the differential of the signal is taken the signal noise doubles, thus the process degrades signal-to-noise ratio.

The theoretical effect of pulse slimming on system performance has been investigated by Mackintosh [22] who reports improvements in packing density of up to 20%. Balanced against this improvement one must weigh the degradation in SNR and the increase in system complexity. This latter point warrants serious consideration where a multitrack format is proposed.

2.5.2. Pre-Equalisation.

Equalisation is also possible by modifying the spectral components of the recorded signal to match the response of the channel.
The interfering effect of adjacent pulses serves to shift the peaks of the recorded magnetic transitions relative to the position of the signal transition which caused it (figure 2.5). The extent of peak shift depends on the packing density of the recorded signal and the immediate pattern of ones and zeroes formed by the particular recording code employed. Peak shift is undesirable since it impairs the detection process. Whilst post-equalisation reduces peak shift it may also be reduced by transition shifting [23].

The effect of transition shifting is shown in figure 2.15. If two
adjacent signal transitions are likely to cause peak shifting when recorded the individual pulses are actually recorded with a reduced time interval between their transitions. When detected on playback the effect of peakshift will have moved the transitions back to their normal position.

A major problem when transition shifting is employed is deciding which two adjacent signal transitions merit pre-compression. This would depend on the recording code adopted and in some cases on the transition pattern of the signals to be encoded.

As in the case of post- equalisation the benefits of this method are limited. Also, because pre - shifting transitions effectively moves the signal to a higher frequency one effect of transition shifting is to reduce the amplitude of the playback signal [24].

The reduction in amplitude caused by transition shifting is capitalised upon by Schneider [25] to produce a flat response of the replay signal. By adding higher frequency write pulses at strategic points in the write current waveform, pre-equalisation is effected. The resultant equalisation of the replayed waveform produces a signal whose amplitude is independent of flux-transition density.

In terms of utilising intelligent devices to parallel encode data in a multitrack format pre - equalisation is an attractive proposition.

As indicated in the previous section the effects of intersymbol interference (ISI) limits the recording density of a magnetic
recording system. However, if the spectral response of the data channel is accurately defined the ensuing ISI due to a high recording rate may be predicted and thus allowed for in detection. This technique makes use of the partial response of the system at high data rates.

Kretzmer[26] has classified a number of partial response systems in terms of the number of received signal levels. Of particular interest is class IV. The transfer function of the class IV partial response channel is very similar to the bandpass characteristic of the standard recording channel.

Class IV partial response offers a number of advantages. Signalling at the Nyquist rate is possible and the constraints placed on the level transitions endow it with an inherent error-detection capability. A number of researchers have proposed class IV partial response recording systems which seek to exploit one or more of its advantages [27 - 29]. On the debit side implementation is complex and certain aspects of detection performance are inferior to other methods [30].
2.6. REFERENCES FOR CHAPTER 2.


[27] Coleman D. et al., "High data rate magnetic recording on a single channel", IERE Proc. No 59, Fifth International Conference on Video and Data Recording, Southampton, April 1984.


CHAPTER 3.

THEORY & ANALYSIS of ERROR CORRECTION & RECORDING CODES.

3.1. RECORDING CODES.

The use of a modulation, or channel, code is beneficial for two reasons: to match the spectra of the raw digital data to the response of the recording channel and to increase the packing density above that which would obtain without the use of a channel code. Selection criteria for a recording code include ease of implementation and the ability to decode efficiently.

The basic characteristics of binary recording codes and their construction will be covered. A review of codes and coding techniques will also be given. Finally, a brief overview of M-ary codes is given.

3.1.1. Parameters of Recording Code.

A number of basic code parameters are first explained.

Code rate: When \( m \) information symbols are encoded into \( n \) code symbols the code rate is \( n/m \). Code rate \(< 1\).

Run length: This relates to the distance between transitions in the coded sequence. Two parameters, \( d \) and \( k \) are defined:

\( d \) = The minimum number of "zeroes" between adjacent ones. (The highest transition density).

\( k \) = The maximum number of "zeroes" between adjacent ones. (The lowest transition density).

In this definition of \( d \) and \( k \) ones are interpreted as changes in signal level and zeroes as no change.
A low value of \( d \) is desirable to ensure synchronism between the code and the data clock during decoding. However, a low value of \( d \) restricts packing density. \( k \) should be limited to reduce the low-frequency components in the spectral response of the code.

Digital sum variation (DSV): The digital sum variation is the running integral of the area beneath the coded sequence assuming the binary levels to be \(+1\). If the DSV of a code is bounded the code is d.c. free.

Capacity: In his paper Shannon [1] showed that capacity in units of user bits per channel bit is given by,

\[
C = \log_2 x \quad \text{where } x \text{ is the largest root of the equation,}
\]

\[
F(x) = x^{k+1} - x^{k-d} - x + 1 = 0. \quad \ldots \ldots \ldots \ldots (3.1)
\]

\( d, k \) are the code parameters defined above. Equation 3.1 is derived from a consideration of the finite-state timing diagram (FSTD) of the code. This is done in appendix A.

In this form \( C < 1 \) and the Shannon capacity can be determined in terms of the code parameters \( d \) and \( k \). A code is deemed efficient if its rate is close to \( C \). Consequently,

code efficiency = code rate/capacity.

In practice code rate is chosen as a rational number \( m/n < C \).

The capacity of a code for various values of \( d, k \) is shown in figure 3.1.

3.1.2. Code Construction.

A method of run length limited code construction using finite state machine graphing methods is due to Franaszek [2]. Adler et al. [3] have applied mathematical techniques to develop a
structured method of generating run-length limited codes. This is known as a sliding block code algorithm and it can be used to generate efficient codes with limited error propagation. The sliding block code algorithm incorporates the FSM graphing methods of Franaszek. These ideas are demonstrated in Siegel's paper on recording codes [4].

Consider a d,k constraint of 1,3. The capacity for this code is 0.5515, therefore choose a rate of 1/2. A finite state transition diagram (FSTD) is constructed having k+1 states, figure 3.2. The paths through the FSTD correspond to binary sequences satisfying the 1,3 constraint. From the FSTD a finite state machine (FSM) graph is developed where each transition between states is labelled with a two-bit codeword, figure 3.3.
Fig. 3.2. FSTD for code with constraint 1,3.

Fig. 3.3. FSM graph of FSTD in figure 3.2.
Figure 3.3. is now inspected and states with less than 2 outgoing edges are eliminated. In this case D is eliminated to yield figure 3.4.

Fig. 3.4. FSM graph with D state eliminated.

The final stage is the merging of any states which have identical outgoing labels, i.e., states B and C in figure 3.4. This results in the FSM graph of figure 3.5.

Fig. 3.5. FSM graph for code with d,k constraint 1,3.
The binary value appended to each codeword is the information bit represented by that codeword.

The state diagram of figure 3.5. gives the encoding rules for the code MFM, or Miller code. The waveform for MFM is deduced by following the paths through figure 3.5. and ascribing a change in level to a logical 1 and no change to zero, figure 3.6. The above procedure may be followed to construct efficient d,k codes where \( m/n < C \).

![Miller code waveform](image)

**Fig.3.6. Miller code waveform.**

The above method of code construction applies to a memoryless encoder. If the encoder can store information before it is encoded successive data bits may be inspected before encoding. This look-ahead technique permits coding decisions to be made based on present and future data thus overcoming the codeword-length restriction encountered in the method described.

Of fundamental importance is the ability to identify each codebit during playback. This process involves identifying the
time period of each codebit. A clock operating at codebit rate is used and this must be synchronised to the code. The ability of a code to synchronise a replay clock is quantified in terms of the $d$ value of the $d,k$ code. This should not be so high as to inhibit clock synchronisation. In terms of the frequency domain: the spectral response of a recording code should include a strong component at either the bit rate or a multiple of it.

Because of the importance of bit synchronisation a key parameter of a channel code is the detection or sample window. This relates to the timing interval during which the codebit may be sampled to determine its level. As an example the detection window for Miller code is shown in figure 3.6. to be one half of a bit cell. In practice this will be reduced considerably by timebase jitter.

3.1.3. Review of Channel Codes For Recording.

Through the utilisation of channel codes the packing density of digital tape recorders used in computer storage has roughly doubled each decade since the 1950s [5]. In the relatively new field of digital-audio tape recording, where system requirements are not as stringent, channel codes have permitted packing densities to rise to five times that of computer systems since the advent of the first digital-audio tape recorder constructed by the BBC in 1974 [6].

A brief review, with critique, now follows of digital channel codes used in tape recording.

A code adopted when digital tape recording began to emerge is
Non-Return to Zero (NRZ). The two most important versions of this family are NRZ-L (L signifies level) and NRZ-M (M signifies mark), figure 3.7. These codes have a rate of unity and \( d,k = 0, \infty \), consequently their efficiency is 1. The detection window is equal to one bit-cell, \( T \) and a clock frequency of \( 1/T \) is necessary to decode. Two major disadvantages of the codes is that they are not run length limited nor are they self clocking. This latter disadvantage implies the requirement for a timing reference to be recorded alongside the encoded data. Enhanced versions of NRZ (E-NRZ) have been developed to overcome these disadvantages. One technique is to randomise the data before encoding and so render the code run length limited.

Perhaps the earliest of self-clocking codes are the Biphasic group of codes. These are variously known as Frequency Modulation
(FM) or Manchester codes. Two versions are in use:— Biphase-L and Biphase-M. Biphase-L is illustrated in figure 3.8. This is a run-length limited code with code rate 1/2 and a d,k constraint of 0,1 which gives a capacity of 0.6942 and efficiency 72%. Biphase-L has excellent clock synchronising characteristics and is d.c. free. Its disadvantages include a detection window of one half of one bit cell and a high transition density which limits the potential packing density of the code.

The Biphase family of codes have been modified to produce Modified Frequency Modulation (MFM), also known as Miller code after its inventor[7]. This code is classed as a double-density code since the transition density is one half that of the Biphase codes. The FSM graph and sample waveform for MFM code are shown in figure 3.5 and 3.6. As already stated its d,k value is 1,3 which permits clock synchronisation, although not with the same ease as Biphase code. The code rate of MFM is 1/2 giving an efficiency of 90.6%. The detection window of MFM is one half of one bit cell and this code requires a detection clock rate which is 2 x the bit rate. MFM code is not entirely d.c. free.
The move towards d.c. free codes for magnetic recording began in 1975 with the publication of Patel's paper: "Zero Modulation Encoding in Magnetic Recording" [8]. This was followed by Miller's modification to his namesake's code which resulted in Miller Squared code [9]. The basic technique is to group the code sequences into consecutive ones bounded by ones and bounded by zeroes. If the DSV of these sequences is zero they are encoded in straight Miller code. Should the DSV of a sequence not be zero its transition positions are changed to give a zero DSV.

A family of codes developed specifically for digital audio applications are the High density modulation codes (HDM) [5]. In particular HDM-1 code has been developed for use in stationary-head tape recorders. HDM-1 code has rate 1/2, and d,k = 2,8. This run-length constraint is achieved through the application of a complicated algorithm requiring look-ahead and look-back capability. The code is not d.c. free and the detection window is half of one bit cell.

The final code to be considered in this brief overview is a group code which employs a look-ahead technique. ISS 2/3 code is a 2/3 rate code invented by Cohn, Jacoby and Bates [10] who were with the company ISS Sperry/Univac at the time. With this code two data bits are encoded into three code bits, figure 3.9. The codebit ones represent a signal transition. No two adjacent code bits are ones; thus ISS 2/3 code offers a greater potential recording density when compared to Miller Squared code. In the event of data sequences occurring which would give adjacent code ones the code bits are modified to prevent this, figure 3.10.
Fig. 3.9. Encoding table for ISS 2/3 code.

<table>
<thead>
<tr>
<th>Data</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>101</td>
</tr>
<tr>
<td>01</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>001</td>
</tr>
<tr>
<td>11</td>
<td>010</td>
</tr>
</tbody>
</table>

Fig. 3.10. Replacement code for data sequences which would otherwise code into two adjacent transitions, ISS 2/3 code.

<table>
<thead>
<tr>
<th>Data sequence</th>
<th>Replacement code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>101000</td>
</tr>
<tr>
<td>0001</td>
<td>100000</td>
</tr>
<tr>
<td>1000</td>
<td>001000</td>
</tr>
<tr>
<td>1001</td>
<td>010000</td>
</tr>
</tbody>
</table>

With $d,k = 1,7$ the code efficiency is 98%, the detection window is 0.66 of one bit cell and the clock rate is 1.5 times bit rate. ISS 2/3 code is not d.c. free and it is not self clocking.

A summary of the characteristics of the codes discussed above is given in figure 3.11.

A number of attempts have been made to compare digital recording codes [11 - 14]. Mackintosh [12] concludes that there is little to choose between all popular codes as regards maximum packing.
<table>
<thead>
<tr>
<th>Code</th>
<th>Rate</th>
<th>Capacity</th>
<th>Efficiency</th>
<th>d,k constraint</th>
<th>Detection window</th>
<th>dc free?</th>
</tr>
</thead>
<tbody>
<tr>
<td>NRZ-L</td>
<td>1</td>
<td>1</td>
<td>100%</td>
<td>0,∞</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>NRZ-M</td>
<td>1</td>
<td>1</td>
<td>100%</td>
<td>0,∞</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>E-NRZ</td>
<td>7/8</td>
<td>1</td>
<td>87.5%</td>
<td>0.13</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>Biphasel</td>
<td>1/2</td>
<td>0.6942</td>
<td>72%</td>
<td>0.1</td>
<td>T/2</td>
<td>Yes</td>
</tr>
<tr>
<td>Biphasem</td>
<td>1/2</td>
<td>0.6942</td>
<td>72%</td>
<td>0.1</td>
<td>T/2</td>
<td>Yes</td>
</tr>
<tr>
<td>Miller</td>
<td>1/2</td>
<td>0.5515</td>
<td>90.6%</td>
<td>1.3</td>
<td>T/2</td>
<td>No</td>
</tr>
<tr>
<td>Miller²</td>
<td>1/2</td>
<td>0.6509</td>
<td>76.8%</td>
<td>1.5</td>
<td>T/2</td>
<td>Yes</td>
</tr>
<tr>
<td>ZM</td>
<td>1/2</td>
<td>0.5515</td>
<td>90.6%</td>
<td>1.3</td>
<td>T/2</td>
<td>Yes</td>
</tr>
<tr>
<td>ISS 2/3</td>
<td>2/3</td>
<td>0.6793</td>
<td>98%</td>
<td>1.7</td>
<td>0.66T</td>
<td>No</td>
</tr>
<tr>
<td>HDM-1</td>
<td>1/2</td>
<td>0.6266</td>
<td>79.8%</td>
<td>2.8</td>
<td>T/2</td>
<td>No</td>
</tr>
</tbody>
</table>

Fig.3.11. Summary of recording code parameters.
density whilst according to Kiwimagi [13] there is no single best choice for all situations. It is indeed difficult to place codes in league order when considering them in isolation. No small consideration is the cost and complexity of implementation and since the objective is high data rate, error-free recording, compatible equalisation and error-detection/correction techniques must be considered.

3.1.4. M-ary Codes.

Increased packing density is possible through the use of multi-state coding methods. If M states are recorded the packing density may be increased by a factor of M/2 compared to binary coding. The degrees of recording freedom are position, polarity and amplitude. M-ary coding schemes for magnetic recording utilise one or more of these conditions with binary, being a class of M-ary codes, using the position of flux transitions to encode data.

Data may be represented by one of a number of magnetisation levels on tape. Mackintosh and Jorgensen have conducted a theoretical investigation into the use of multi-level encoding [15]. They examined a 2-ary recording system with fixed transition density and signal-to-noise ratio (SNR). As the transition density falls the SNR improves. This improvement in SNR can be utilised by increasing the number of encoding levels to a point where the SNR is again the same. This procedure yields a value for improved packing density against SNR. For any improvement over a conventional system the required SNR was found to be outside that of practical tape recording equipment.
Also, in view of tape-separation losses, and the loss in benefits of saturation methods, the use of multi-level encoding for reliable data recording is very dubious.

M-ary code implementation using saturation recording has been proposed. Jacoby [16] has published details of a ternary "Three-Position Modulation" (3PM) coding scheme. This uses the peak shift between two closely-spaced flux transitions (doublet) as the third signalling element. The three elements are 0, 1 and 2 represented by the absence and presence of magnetic flux and a doublet respectively. These are illustrated in figure 3.12. From

```
1 0 2 0 0 1 0 0 2
+sat---
-sat
```

Fig.3.12. Elements of Ternary 3PM Recording code.

Chi's results on peak shift [17], figure 2.7., the crossover position for a recorded doublet is well defined. The detection of the replay waveform is determined by identifying the position at which the peak of a singlet pulse or the crossover point of a doublet pulse occurs. Packing densities double that of MFM code are possible using ternary 3PM and the detection window is 66%.
of one" bit" cell. The recording of consecutive ones and twos is not permitted and coding algorithms to avoid this must be applied.

Chi has utilised a change in the polarity sequence of two recorded doublets in a ternary recording scheme [18]. In "Controlled Return to AC" (CRA) code a high frequency AC erasure pattern is continuously recorded only to be broken for short durations to mark either one or two doublets, figure 3.13. The

![Fig. 3.13. Two doublets from CRA code.](image)

elegance of this scheme is that true saturation methods are not sacrificed for the additional signalling element. The code is d.c. free and its overwrite properties are excellent. It offers a 50% increase in packing density compared to MFM code with a detection window of 50% of one CRA digit. One disadvantage is that since the flux change is from +θ or -θ to zero the amplitude of the readback signal is 6dB down compared to a conventional recording code.
Encoding/decoding methods for M-ary codes are often complex and this usually offsets the improvements in packing density. However, data synchronising techniques developed in Chapter 4 may be applied to identify each CRA cell. In a later modification [17] Chi incorporated an additional dimension by varying the duration of recorded doublets and thus exploited the change in amplitude in readback signal between doublets of two different durations. This gives a density ratio improvement of 130% over MFM code although system complexity is considerably increased.

The sliding block coding algorithm of Adler et.al. [3] may also be applied to M-ary codes. Dixon et.al. [19] have used this algorithm to deduce capacities for a number of codes including the 3PM and CRA codes described above. Capacity values of 0.747 and 1.5458 for 3PM and CRA yield efficiencies of 89.2% and 97% respectively for the two codes.

3.2. ERROR CORRECTION CODES.
Although Shannon predicts that for code rate \( R \leq \) capacity \( C \) data may be transmitted at an arbitrarily low error rate it is often expedient to maximise \( R \) at the expense of data integrity and to use error-correction techniques to reduce the resultant errors. This section deals with error-correction methods suitable for application to digital data tape recording. It opens with an examination of spatial techniques which utilise simple parity after which cyclic codes are examined.

3.2.1. Spatial Error Detection/Correction.
In order to detect/correct errors in a received codeword it is
necessary to add redundant data to the information to be transmitted. The power of an error detection/correction code is usually proportional to the extent of this redundancy. In multi-track tape recorders advantage may be taken of the spatial distribution of the data bits on the tape and so reduce the redundancy requirement.

Data are arranged to configure a co-ordinate system, figure 3.14.

Fig. 3.14. Co-ordinate error correction.

Solution of equations $P_{vt}$ and $P_{th}$ identify the bit in error which subsequently may be corrected: only single errors may be corrected. Multiple errors may be detected although some error patterns will be missed. This latter disadvantage may be overcome by replacing the horizontal parity bit by a polynomial cyclic redundancy check character (CRCC). This has been applied in some multitrack recorders [6] [20] where the vertical parity bits are recorded on a single parity track and the CRCC is placed on each track at regular intervals. This technique works well since
when errors occur they are likely to be confined to one track. A disadvantage of this method is that track data are interrupted by the CRCC and additional synchronising patterns need to be incorporated on each track to discriminate between data and parity bits.

If uninterrupted track data is required the parity bits may be placed on dedicated tracks with the parity bits covering data lying at ±45 degrees across the tape. Patel [21] has used this technique with two additional "vertical" parity tracks in a cross parity error-correction scheme, figure 3.15. This can correct up to two tracks in error out of fourteen data tracks. Data lying at other than ±45 degrees across the tape may also be covered by parity tracks [22].

Fig. 3.15. Cross parity error correction.
3.2.2. Interleaving.

Isolated errors are easier to correct than burst errors. However, this latter error pattern is common in tape systems where the build up of debris on the head can wipe out a number of consecutive track bits. One technique of converting burst errors into isolated errors is to interleave codewords before recording. Assume a block of m,n - tuple codewords each with burst error - correcting capability b:

\[ v_l = d_{l1} + d_{l2} + ... + d_{ln} \]

\[ v_2 = d_{21} + d_{22} + ... + d_{2n} \]

\[ \vdots \]

\[ v_m = d_{m1} + d_{m2} + ... + d_{mn} \]

Record the codewords interleaved to a degree m:

\[ v = d_{11} + d_{21} + ... + d_{mn} + d_{2n} + ... + d_{mn} \]  \hspace{1cm} (3.3)

A burst error of length mb will only affect b bits of a codeword therefore the error - correction ability of the interleaved code is increased by m, the degree of interleaving.

It should be noted that a degree of interleaving is inherent in the process of recording data broadside across the tape, the depth of interleaving being equal to the number of tracks.

3.2.3. Block Codes.

For an n, m binary code there are \( 2^m \) data words of length m. These are encoded into a block of \( 2^n \) codewords of length n. Any decoder must assign the appropriate data word to any of the \( 2^n \) codewords. Since for large m this would require an extensive decoder the virtues of linearity and periodicity are used to simplify decoder design. A linear code is one where the sum of any two
codewords is a codeword. If a logical shift of a codeword results in another codeword the code is cyclic.

In the treatment of linear, cyclic codes, codewords may be represented by a polynomial in $X$ the order of which represents the weight of the codebit. The coefficients of the polynomial are either zero or one, e.g.

$$M(X) = 1 + X^2 + X^3 + X^5 \quad 101101$$

Codes are classified in terms of their generating polynomial $G(X)$. Let $M(X)$ represent the data to be encoded, then

$$\frac{M(X)}{G(X)} = Q(X) + \frac{R(X)}{G(X)} \quad \text{...... (3.4).}$$

where $Q(X) = \text{quotient}$, $R(X) = \text{remainder}$, therefore,

$$M(X) = Q(X) G(X) + R(X) \quad \text{and}$$

$$M(X) + R(X) = Q(X) G(X) \quad \text{modulo 2} \quad \text{...... (3.5).}$$

The data plus the remainder are completely divisible by $G(X)$. $R(X)$ is the polynomial cyclic redundancy check character (CRCC), the "parity" bits. At the decoder $M(X) + R(X)$ is further divided by $G(X)$ to give a zero remainder. If errors have been incurred during transmission the remainder will be non-zero.

In order to identify data bits on inspection equation (3.5) may be put in systematic form. This involves shifting the data $n - m$ places before the remainder is appended, i.e.

$$M(X) X^{n-m} + R(X) = Q(X) G(X) \quad \text{...... (3.6).}$$

The error correction ability of a linear block code can be demonstrated as follows, let $r(X)$ be a received codeword,
dividing $r(X)$ by $G(X)$:

$$r(X) = Q(X) G(X) + S(X) \quad \ldots \ldots \ldots (3.7)$$

where the syndrome $S(X)$ is the remainder.

Let $v(X)$ be a transmitted codeword and let $e(X)$ be an error pattern. The received vector:

$$r(X) = v(X) + e(X) \quad \ldots \ldots \ldots (3.8)$$

from (3.6) and (3.7),

$$e(X) = Q(X) G(X) + S(X) + v(X)$$

$v(X)$ is a multiple of $G(X)$, i.e. $v(X) = b(X) G(X)$

therefore, $e(X) = Q(X) G(X) + S(X) + b(X) G(X)$

$$e(X) = [Q(X) + b(X)] G(X) + S(X) \quad \ldots \ldots (3.9).$$

Equation (3.9) shows that the syndrome is equal to the remainder resulting from dividing the error pattern by $G(X)$, thus the syndrome may be used to identify the error pattern.

The generating polynomial for an $n,m$ code is a factor of $x^n + 1$ [23 - 24] and any factor of $x^n + 1$ with degree $n - m$ generates an $n,m$ cyclic code. These codes have minimum distance $d_s \leq n - m$ and are thus able to correct $t$ errors where,

$$t \leq n - m - 1$$

The encoding/decoding circuitry required for cyclic codes comprises a number of shift - register stages and exclusive OR gates. Consider the generating polynomial $G(X) = 1 + X + X^3$. This generates a 7,4 code which requires an encoder with 7 - 4 stages, figure 3.16. The effect of clocking $M(X)$ through the encoder is to divide it by $G(X)$. Whilst this is in progress $M(X)$ is being
recorded. When this is complete the switch is thrown to B and the R(X) thus generated is clocked out of the encoder into the record channel. The decoder follows the same form as figure 3.16.

After decoding, the syndrome in the shift register is unique and identifies one of the seven bit positions in error which can then be corrected. This may be accomplished using a look-up table containing the corrections corresponding to each syndrome. Although fast, this decoding method is cumbersome and for long codes the memory requirement is extensive. An alternative is to use a Meggitt decoder [25]; this utilises the cyclic nature of the code. A non-zero syndrome, when shifted in the decoding shift register, sequences through the n-1 syndromes of the code each of which relates to the bit positions of the codeword. The number of shifts required to give the syndrome of the most significant bit of the codeword identifies the bit position in error. Because of
the cyclic shift process involved this method is slower than the
look-up table decoding method.

3.2.4. BCH Codes.
The Bose, Chaudhuri and Hocquenghem (BCH) codes are a class of
random error correcting cyclic codes [25]. The generating
polynomial of a t error correcting BCH code of length $2^m - 1$ is
given by the lowest degree polynomial over Galois Field 2
(GF(2)) which has $\alpha^1, \alpha^2, \ldots, \alpha^{2t}$ and their conjugates as its
roots, where $\alpha$ is a primitive element in the extension field
GF (2$^m$). The generating polynomial $G(X)$ is,

$$ G(X) = \text{LCM} \{ \varphi_1(X), \varphi_2(X), \ldots, \varphi_{2t}(X) \} \ldots \ldots \ldots (3.10), $$

where LCM represents the lowest common multiple.

Now $\alpha^i = (\alpha^j)^{2^e}$, where $j$ is an odd number, $i$ is an even integer
and $e \geq 1$. Therefore, $\alpha^j$ is the conjugate of $\alpha^i$ and $\varphi_i(X) = \varphi_j(X)$.
Thus, the minimal polynomials with even powers may be eliminated
from the generating polynomial and equation 3.10 becomes,

$$ G(X) = \text{LCM} \{ \varphi_1(X), \varphi_3(X), \ldots, \varphi_{2t-1}(X) \} \ldots \ldots \ldots (3.11). $$

The generating polynomial of a BCH code may be determined as
follows,

(a) Choose the length of the code, i.e. 15. Therefore $15 = 2^m - 1$
    and $m = 4$.

(b) Choose error-correcting ability, i.e. $t = 2$.

(c) Refer to literature for the appropriate primitive polynomial,
    i.e. $1 + X + X^4$.

(d) List roots of $G(X)$ up to the required number, i.e. $\alpha, \alpha^3, \alpha^5$ etc.
(e) Find the minimal polynomial for each of the above roots.

e.g. for \( \alpha \):

conjugate roots of \( \alpha = (\alpha^2), (\alpha^3), (\alpha^7), (\alpha^8), (\alpha^9) \)

but \( \alpha^6 = \alpha \) therefore the minimal polynomial for the above four roots is,

\[
\phi_1 = (X + \alpha) (X + \alpha^2) (X + \alpha^4) (X + \alpha^8)
= X^4 + X^3 (\alpha + \alpha + \alpha^3 + \alpha^5) + X^2 (\alpha + \alpha + \alpha^3 + \alpha^4 + \alpha^5 + \alpha^6) + X (\alpha + \alpha + \alpha^3 + \alpha^4 + \alpha^5 + \alpha^6 + \alpha^7) + \alpha^8
\]

\[
\phi_1 = X^4 + X^3 (0) + X^2 (0) + X (1) + 1
\]

\[
\phi_1 = X^4 + X + 1.
\]

The same procedure for \( \alpha^3 \) and its conjugates yields,

\[
\phi_3 = X^4 + X^3 + X^2 + X + 1.
\]

The generating polynomial is, therefore,

\[
G(X) = \phi_1(X) \phi_3(X)
= (1 + X + X^4) (1 + X + X^2 + X^3 + X^4)
G(X) = 1 + X^4 + X^6 + X^7 + X^8.
\]

3.2.5. Decoding Process.

Each codeword is a factor of \( G(X) \) and therefore also has \( \alpha', \alpha^2, \ldots, \alpha^{2t} \) as roots. The decoding process involves substituting these roots into each codeword, \( v \) and verifying that \( v(\alpha^i) = 0 \).

For \( v(\alpha^i) \neq 0 \) the syndromes are used to determine the error positions. This process generates \( 2t \) (\( 2t/2 \) for odd powers of \( G(X) \) roots) syndromes. Since the syndromes are evaluated by substitution of the roots \( \alpha^i \) of \( G(X) \) they take the form,

\[
S_i = e_{i_1} \alpha^{i_1} + e_{i_2} \alpha^{i_2} + e_{i_3} \alpha^{i_3} \ldots + e_{i_{2t}} \alpha^{i_{2t}} \ldots (3.12)
\]

where \( e_{i_{2t}} \) is the magnitude of the \( 2t \)th error. For binary codes \( e = 1 \) and re-writing (3.12) the syndromes are,
Attempts at achieving a viable BCH decoding algorithm have centred on the solution of the set of equations (3.13). This is done by defining an error - location polynomial the coefficients of which relate to the syndrome components. The error - location polynomial, \( \sigma (X) \) has roots at \( x_1^{-1}, x_2^{-1}, x_3^{-1} \) etc. Once the roots of \( \sigma (X) \) have been determined the bit position of the error is found and this may be corrected.

The error - location polynomial is defined as,

\[
\sigma(X) = (1 + x_1 X) (1 + x_2 X) \ldots \ldots (1 + x_r X)
\]

where the roots of (3.14) are the inverses of the error location numbers and,

\[
\begin{align*}
\sigma_0 &= 1 \\
\sigma_x &= x_1 + x_2 + x_3 + \ldots \ldots + x_r \\
\sigma_2 &= x_1 x_2 + x_2 x_3 + \ldots \ldots + x_{r-1} x_r \\
\sigma_3 &= x_1 x_2 x_3 + \ldots \ldots \ldots x_r
\end{align*}
\]

From equations (3.13) and (3.14) the syndrome components are related to the coefficients of the error - location polynomial as follows,

\[
\begin{align*}
S_1 &= \sigma_x \\
S_2 &= \sigma_x S_1 + 2 \sigma_2 \\
S_3 &= \sigma_x S_2 + \sigma_2 S_1 + 3 \sigma_3 \ldots \ldots (3.15)
\end{align*}
\]
\[
S_{V+1} + \sigma_1 S_{V} + \ldots + \sigma_{V-1} S_2 + \sigma_1 S_1 = 0, \text{and} \ldots \ldots (3.16)
\]
\[
\sigma_1 S_{V} + \sigma_1 S_{V-1} + \ldots + \sigma_{V} S_1 = -S_{V+1}, \ldots (3.17)
\]
Solution of equations (3.15) give the coefficients of the error-location polynomial from which the roots may be found.

In the Peterson [27] and Gorenstein and Zierler [28] decoding algorithm equation 3.17 is placed in matrix form:

\[
\begin{bmatrix}
S_1 & S_2 & S_3 & \ldots & S_{V-1} & S_V \\
S_2 & S_3 & S_4 & \ldots & S_{V} & S_{V+1} \\
S_3 & S_4 & S_5 & \ldots & S_{V+1} & S_{V+2} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
S_V & S_{V+1} & S_{V+2} & \ldots & S_{2V-2} & S_{2V-1} \\
\end{bmatrix}
\begin{bmatrix}
\sigma_V \\
\sigma_{V-1} \\
\sigma_{V-2} \\
\vdots \\
\sigma_1 \\
\end{bmatrix}
= \begin{bmatrix}
-S_{V+1} \\
-S_{V+2} \\
-S_{V+3} \\
\vdots \\
-S_{2V} \\
\end{bmatrix} \ldots (3.18)
\]

\[
[M] [\sigma^v] = [S]
\]
It can be shown [29] that for some value \( u > v \) \( \det [M] = 0 \), if \( u = v \) \( \det [M] \neq 0 \).

In the decoding process \( v \) is set equal to \( t \) and \( \det [M_t] \) is evaluated. If \( \det [M] = 0 \) \( v \) is set equal to \( t-1 \) and \( \det [M_{t-1}] \) is evaluated. This process continues until \( \det [M_{t-1}] \neq 0 \) at which point the error location polynomial is evaluated from equation (3.18).

The error-location polynomial may also be deduced using Berlekamp's iterative algorithm [30]. The process involves finding a minimum degree polynomial \( \sigma^v(X) \) whose coefficients satisfy the equations of (3.15). A minimum degree polynomial satisfying the first equation of (3.15) is formed, \( \sigma^1(X) \). If the coefficients of \( \sigma^1(X) \) satisfy the second equation in 3.15, then let \( \sigma^2(X) = \sigma^1(X) \), if not, a correction term is added to \( \sigma^1(X) \) to form \( \sigma^2(X) \) such that \( \sigma^2(X) \) has minimum degree and satisfies the

108
first two of equations (3.15).

The above process continues until $\alpha^{-2t} (X)$ is arrived at. For binary codes it is only necessary to complete $t$ iterations to arrive at the error - location polynomial.

Once the error - location polynomial has been found its roots are determined. This is usually undertaken by instituting a Chien search [31]. This involves substituting $\alpha_i$ into $\alpha^{-i} (X)$. Since $\alpha^n = 1$, $\alpha^{-i} = \alpha^{-i} - n$, therefore, if $\alpha^{-i}$ is a root of $\alpha^{-i} (X)$, $\alpha^{-i}$ is an error-location number and the received digit $r_{n-i}$ is in error. The high order bits of the received codevector are treated first. To decode $r_{n-i}$ the inverse of $\alpha^{-i} (\alpha^{-i})$ is substituted into $\alpha^{-i} (X)$. If $\alpha^{-i} = 0$, $\alpha^{-i}$ is a root and bit position $n-i$ is in error, if $\alpha^{-i} \neq 0$ there is no error at this bit position.

3.2.6. Reed - Solomon Code.
A subclass of the BCH codes is the Reed - Solomon (RS) code [32]. The RS code is a non-binary code with code symbols from GF ($q$).
A $t$ error-correcting RS code has the following parameters.
Block length: $n = q - 1$.
Number of parity check bits: $n - m = 2t$.
Minimum distance: $d_m = 2t + 1$.
Generating polynomial: $G (X) = (X + \alpha^1) (X + \alpha^2) \ldots (X + \alpha^{2t})$. (3.19).

The RS code has a $d_m$ of one more than the number of parity - check digits and for this reason it is a maximum distance code.
With allowance for the handling of symbols rather than bits, the encoding process for RS code is the same as for the binary BCH codes. Decoding is also the same with a fourth step necessary to determine the magnitude of the error.

From equation 3.12,

\[ S_1 = e_{i1} \alpha^{i_1} + e_{i2} \alpha^{i_2} + \ldots \ldots e_{iv} \alpha^{i_v} \]
\[ S_1 = Y_1 X_1 + Y_2 X_2 + \ldots \ldots Y_v X_v \]

where \( Y_v = e_{iv} \) = the error magnitude.

\( X_v = \alpha^{i_v} \) = the error position.

Therefore,

\[ S_1 = Y_1 X_1 + Y_2 X_2 + \ldots \ldots Y_v X_v \]
\[ S_2 = Y_1 X_1^2 + Y_2 X_2^2 + \ldots \ldots Y_v X_v^2 \]
\[ \ldots \]
\[ S_{2t} = Y_1 X_1^{2t} + Y_2 X_2^{2t} + \ldots \ldots Y_v X_v^{2t} \ldots (3.20). \]

The error positions, \( X_v \) are determined from the error - location polynomial and a Chien search. Once the error positions have been identified the error magnitude may be determined from equations 3.20.

E.g., for two errors, \( X_1 X_2 \), denote the error positions with corresponding error magnitudes \( Y_1 Y_2 \),

therefore,

\[ S_1 = Y_1 X_1 + Y_2 X_2 \ldots \ldots (3.21) \]
\[ S_2 = Y_1 (X_1)^2 + Y_2 (X_2)^2 \ldots \ldots (3.22) \]

from which,

\[ Y_1 = \frac{S_2 + S_1 X_2}{(X_1 X_1 + X_1 X_2)} \ldots \ldots (3.23) \]
and,

\[
y_2 = \frac{S_2 + S_1 x_1}{x_1 x_2 + x_2 x_2} \quad \ldots \ldots \quad (3.24)
\]

3.3. ANALYSIS OF MOVING - PARITY ERROR CORRECTION.

The data channel of a digital magnetic recording system is subject to burst errors. The error - burst length may be reduced to manageable proportions by interleaving the recorded codewords. The use of interleaving permits simpler, random error - correcting codes to be employed. The additional decoding overhead incurred in employing interleaving is minimal and, with micro - computer techniques, is relatively simple to implement. The effectiveness of interleaving can, however, be thwarted by the occurrence of random errors. In the de - interleaving process the components of the error burst could be positioned close to random errors thus creating a further error burst which would be uncorrectable. This possibility is reduced if crossinterleaving methods are used [33], although this further complicates the error - correction procedure. An alternative approach is to employ a burst error - correcting code [34 - 39]. This reduces, or may eliminate, the need for interleaving. However, system complexity may well outweigh any disadvantages attached to the application of both interleaving and random error - correcting codes.

3.3.1. Efficiency of Moving Parity Code.

The error burst length incurred with the compact - cassette system depends on the recorded data rate, figure 5.31. The error logging system, section 4.4.17, interprets an error burst as the
number of consecutive failures of the replayed data between two successful readings. For the purpose of analysis the error burst length is redefined as the number of bits \( b \), between two error bits preceded and followed by a guard space of error-free bits, \( g \). The error burst must not contain error-free bits with consecutive length \( g \), e.g. if 0 and 1 represent correct and incorrect bits respectively and the guard space is 6 an error burst of length 13 would be,

\[
\begin{array}{cccccccccccc}
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
\]

Consider the area of the tape covered by the parity strips, figure 3.17.

Assuming single-track errors, all errors with burst length \( b < \ell \) and guard space \( g > L \) will be corrected.

The efficiency of the error-correcting code may be determined:

From figure 3.17, \( L = (\ell - 2) + 2N \) \( \ldots (3.25) \),

where \( N \) is the number of tracks.

This is equal to the number of parity bits, \( (n - m) \), required to cover the data bits \( L (N - 1) = m \), therefore \( L = (n - m) \).

Now, to satisfy the Reiger bound [40] for an optimum burst error-correcting code:

\[
\frac{(n - m)}{2} = \text{Correctable burst length}.
\]

The maximum burst length which is correctable = \( \ell - 1 \).

Therefore, \( \frac{(n - m)}{2} = \ell - 1 \) for optimum code \( \ldots (3.26) \).

Substituting equation 3.25,
Fig. 3.17 Area of tape showing parity strips and parity bits, n-k.
\[
\frac{L}{2} = \frac{l - 2 + 2N}{2} = \frac{l}{2} - 1
\]

Therefore \( L = 2N \) for optimum code.

Thus, if length \( L \) is equivalent to twice the number of tracks the code is optimum and its efficiency \([40]\),

\[
= \frac{2b}{(n - m)} = \frac{2 (L - 1)}{2 (L - 1)} = 1.
\]

The moving-parity error correcting code can only be considered optimum on a track failure basis. However, burst errors only occur along a track and not across them.

Equation 3.26 satisfies the condition for a maximum distance code. However, since any \( t \) errors in \( n \) cannot be corrected, this code cannot be claimed to have this property.

3.3.2. Theoretical Deduction of Moving Parity Error Correction.

The theoretical error-correcting performance of the code will now be deduced. To produce a realistic estimate of performance the worst case (highest probability of error) will be considered.

Let the probability of incurring a single error be \( p \).

The probability of incurring two single-bit errors is higher than incurring three, or more, single-bit errors.

From the binomial probability distribution the probability that \( r \) errors occur in a sample size of \( n - r \) is given by,

\[
C_r^p q^{(n-r)}
\]

where \( C_r = \frac{n!}{r! (n - r)!} \).
\( q = \text{probability of no errors} = (1 - p) \).

Consider the area bounded by the tape edges and the two parity strips, figure 3.17. The number of bit-cells within this boundary, including the bit-cells crossed by the two parity strips is,

\[
B = \begin{cases} \text{Nt} = \text{N-1} & \text{Nt} = 0 \\ \ell + 2\text{Nt} \end{cases}
\]

where \( \text{Nt} = \text{track number}, \)
\( N = \text{total number of tracks}. \)

The probability of two errors occurring within, or on, this boundary is, therefore,

\[
P(2) = \frac{B!}{2!(\ell - \text{2})!} p^2 (1 - p) (1 - 10^{-7})^{\theta - 2} \quad 3.27.
\]

The measured probability of a single error depends on the recorded data rate and the recording code used and varies from approximately \( 10^{-7} \) to \( 10^{-4} \), figure 5.35.

Taking \( p = 10^{-7} \) and \( \ell = 8 \),

\[
P(2) = \frac{44!}{2!(42)!} 10^{-14} (1 - 10^{-7})^{42}
\]

therefore,

\[
P(2) = \frac{44 \times 43}{2} 10^{-14}
\]

\[
P(2) = 946 \times 10^{-14} \approx 10^{-11}
\]

Now the total number of 2-error combinations within the prescribed area,

\[
= \frac{n!}{r!(n - r)!} = \frac{44!}{2!(42)!} = 946.
\]

115
Of the 2-error combinations the number occurring on a single track, and therefore correctable,

\[
N_t = \begin{cases} 
N - 1 & \text{if } D! \\
\frac{D!}{2! (D - 2)!} & \text{if } N_t = 0
\end{cases}
\]

where \( D = (L - 2N_t - 1) \).

\( N_t \) = track number.

\( N \) = total number of tracks.

This evaluates to 190 for \( L = 14 \) and \( N = 4 \).

The total number of incorrectable, double errors is, therefore,

\[ 946 - 190 = 756. \]

Total probability of incorrectable double errors.

\[ = \frac{756 \times 10^{-11}}{946} = 0.8 \times 10^{-11} \]

i.e. better than one in \( 10^{11} \).

3.4. SOFTWARE IMPLEMENTATION OF REED - SOLOMON CODE.

The Reed - Solomon code is an efficient, burst-error correcting code. Its implementation, however, is complex and for operation in real time, at realistic data throughput, dedicated hardware is usually employed. Software implementation is possible but forms an additional overhead for the microcomputer on top of the decoding, synchronising and sampling functions already undertaken. To incorporate the RS code into the recording systems described in this thesis would require efficient coding/decoding software algorithms. The essence of such algorithms would be the manipulation of finite field elements with a microcomputer designed to process numbers in an infinite field. To date RS code has not been applied but the progress made towards this aim will now be discussed.
Subtraction in a finite field is the same as addition and this is easily implemented as the exclusive OR of the two finite-field elements to be summed. The result is automatically reduced by the field-generating polynomial and thus falls within the field.

Multiplication and division by a primitive element, $\alpha$, in the finite field may be accomplished by the software simulation of a linear feedback shift register [41]. For instance, for a field generated by the polynomial,

$$p(X) = 1 + X^2 + X^3 + X^6 + X^9$$

the linear feedback shift register takes the form shown in figure 3.18. The element to be multiplied is loaded into the register and for each clock pulse the contents are multiplied by $\alpha$. Division is accomplished by forming the inverse of the divisor and multiplying. In the software simulation the multiplication operation requires the execution of four logical operations:

- **Shift to right.**
  
  $$b_2 = \text{Exclusive OR of } b_0 \text{ and } b_3.$$  
  $$b_3 = \text{Exclusive OR of } b_0 \text{ and } b_1.$$  
  $$b_4 = \text{Exclusive OR of } b_0 \text{ and } b_4.$$  

Whilst the above method is suitable for multiplying by $\alpha$, the multiplication of two arbitrary elements from the finite field is more complex and may be achieved either through the appropriate hardware configuration or its software simulation [42]. A simple software algorithm for multiplication and division of finite field elements has been developed. By way of
Fig. 3.18. GF Multiplier for field generated by $1 + X^2 + X^3 + X^4 + X^8$
example consider the finite field generated by the polynomial, 
\[ p(X) = 1 + X + X^4 \]. figure 3.19.

<table>
<thead>
<tr>
<th>Power</th>
<th>Polynomial</th>
<th>4-tuple</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1000</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>( \alpha )</td>
<td>0100</td>
</tr>
<tr>
<td>( \alpha^2 )</td>
<td>( \alpha^2 )</td>
<td>0010</td>
</tr>
<tr>
<td>( \alpha^3 )</td>
<td>( \alpha^3 )</td>
<td>0001</td>
</tr>
<tr>
<td>( \alpha^4 )</td>
<td>1 + ( \alpha )</td>
<td>1100</td>
</tr>
<tr>
<td>( \alpha^5 )</td>
<td>( \alpha + \alpha^2 )</td>
<td>0110</td>
</tr>
<tr>
<td>( \alpha^6 )</td>
<td>( \alpha^2 + \alpha^3 )</td>
<td>0011</td>
</tr>
<tr>
<td>( \alpha^7 )</td>
<td>1 + ( \alpha ) + ( \alpha^3 )</td>
<td>1101</td>
</tr>
<tr>
<td>( \alpha^8 )</td>
<td>1 + ( \alpha^2 )</td>
<td>1010</td>
</tr>
<tr>
<td>( \alpha^9 )</td>
<td>( \alpha + \alpha^3 )</td>
<td>0101</td>
</tr>
<tr>
<td>( \alpha^{10} )</td>
<td>1 + ( \alpha + \alpha^2 )</td>
<td>1110</td>
</tr>
<tr>
<td>( \alpha^{11} )</td>
<td>( \alpha + \alpha^2 + \alpha^3 )</td>
<td>0111</td>
</tr>
<tr>
<td>( \alpha^{12} )</td>
<td>1 + ( \alpha + \alpha^2 + \alpha^3 )</td>
<td>1111</td>
</tr>
<tr>
<td>( \alpha^{13} )</td>
<td>1 + ( \alpha^2 + \alpha^3 )</td>
<td>1011</td>
</tr>
<tr>
<td>( \alpha^{14} )</td>
<td>1 + ( \alpha^3 )</td>
<td>1001</td>
</tr>
</tbody>
</table>

Fig.3.19. Finite field generated by 
\[ p(x) = 1 + x + x^4 \]

Inspection of figure 3.19 shows that the weighting values of the 4-tuple representation are \( \alpha^0, \alpha^1, \alpha^2 \) and \( \alpha^3 \), reading from left to right. Also, for the first four elements in the field, starting at 1, shifting the 4-tuple representation one position to the right is equivalent to multiplying by \( \alpha \). From the fifth element onwards the field wraps around but a postulated overflow
would have a weighting value of \( \alpha^4 \). Multiplication by \( \alpha \) is
effected as follows.

Shift 4 - tuple representation one position to the right.

If " overflow " = 0, end of process.

If " overflow " = 1, XOR 1100 (i.e. \( \alpha^4 \)).

e.g. \((\alpha) \ (\alpha^2 + \alpha^3)\)

4 - tuple representation of \((\alpha^2 + \alpha^3)\) = 0011
Shift right:

\[
\begin{array}{c}
\begin{array}{c}
0001 \\
1100
\end{array}
\end{array}
\]
(overflow)

1101 = \( \alpha^3 + \alpha^4 \) = \( 1 + \alpha + \alpha^3 \).

The above algorithm may be employed in multiplying two arbitrary
finite - field elements.

e.g. \((1 + \alpha^2 + \alpha^3) \ (1 + \alpha^2)\).

4 - tuple representation of \((1 + \alpha^2 + \alpha^3)\) = 1011 .......(a)
4 - tuple representation of \((1 + \alpha^2)\) = 1010 .......(b)

Multiply (a) by \( \alpha^0 \): 1011, bit 3 of (b) = 1, carry forward 1011

Multiply (a) by \( \alpha^1 \): 0101
\[
\begin{array}{c}
\begin{array}{c}
0001 \\
1100
\end{array}
\end{array}
\]
(overflow), bit 2 of (b) = 0, no carry forward.

Multiply (a) by \( \alpha^2 \): 0100
\[
\begin{array}{c}
\begin{array}{c}
0000 \\
1100
\end{array}
\end{array}
\]
1000, bit 1 of (b) = 1, carry forward 1000

Multiply (a) by \( \alpha^3 \): 0100
\[
\begin{array}{c}
\begin{array}{c}
0000 \\
1100
\end{array}
\end{array}
\]
1000, bit 0 of (b) = 0, no carry forward.

XOR of partial results = 1011 + 1000 = 0011.

\((1 + \alpha^2 + \alpha^3) \ (1 + \alpha^2)\) = \( 1 + \alpha^2 + \alpha^2 + \alpha^4 + \alpha^3 + \alpha^5 \)

= \( \alpha^2 + \alpha^3 \) \( \rightarrow \) 0011 . C.E.D.
A similar algorithm may be used in the division of two finite-field elements. In this case successive division by $\alpha$ is possible by shifting left and modulo - 2 adding (1001) for each "underflow".

The above algorithms require $3n$ computer instructions to multiply, or divide, two arbitrary $n$-tuple field elements. This compares favourably with $5n$ instructions quoted by Lin [42]. The speed of execution of programmed finite-field arithmetic is much slower than for a hardware configured unit. However, for long codewords it may be the only viable solution.

3.4.1. Reed Solomon Encoder.

The encoding process is less complex than decoding and so can be executed in a shorter time. It involves the software simulation of the appropriate encoding circuitry. Consider the double, error-correcting RS (15, 11) code over GF ($2^4$) generated by, 

$$p (X) = 1 + X + X^4.$$ 

The code generating polynomial, $g (X)$ is,

$$g (X) = (x + \alpha) \ (x + \alpha^2) \ (x + \alpha^3) \ (x + \alpha^4)$$

therefore,

$$g (X) = X^4 + X^3 (1 + \alpha^1 + \alpha^3) + X^2 (\alpha^1 + \alpha^3) + X (\alpha^2)$$

$$+ \alpha + \alpha^2).$$

The encoding circuitry, suitable for a 4-track system, is shown in figure 3.20. The data are divided by $g (X)$ as they are being recorded. Following this the four parity words are clocked out of the latches onto the tape. The encoding process requires approximately $3n \times 4$ computer instructions ($n = 4$), which must be added to the channel - encoding execution time.
\[ p(x) = 1 + x + x^2 \]

**Fig. 3.20.** (15,11) Reed Solomon encoder.
3.4.2. Reed Solomon Decoder.

There are four steps in the decoding of a RS code:

(a) Compute the syndromes.

(b) Generate the error - location polynomial.

(c) Identify the error positions by finding the roots of the error location polynomial.

(d) Calculation of error magnitude.

It is possible to determine the elements of the syndromes whilst the data are being read back from the tape. The syndromes are generated by substituting the roots of the generating polynomial in the received codevector. In the general case, for an error free, all ones codevector,

\[
S_1 = \alpha^0 + \alpha^1 + \alpha^2 \ldots \ldots \alpha^{n-1} \\
S_2 = (\alpha^0)^2 + (\alpha^1)^2 + (\alpha^2)^2 \ldots \ldots (\alpha^{n-1})^2 \\
\vdots \\
S_{2t} = (\alpha^0)^{2t} + (\alpha^1)^{2t} + (\alpha^2)^{2t} \ldots \ldots (\alpha^{n-1})^{2t}
\]

where \( n \) = code length.

\( t \) = error - correcting ability of the code.

E.g. for a double, error - correcting code with \( n = 15 \),

\[
S_1 = 1 + \alpha + \alpha^2 + \alpha^3 \ldots \ldots \alpha^{12} + \alpha^{13} + \alpha^{14} \\
S_2 = 1 + \alpha^2 + \alpha^4 + \alpha^6 \ldots \ldots \alpha^{24} + \alpha^{26} + \alpha^{28} \\
S_3 = 1 + \alpha^3 + \alpha^6 + \alpha^9 \ldots \ldots \alpha^{36} + \alpha^{39} + \alpha^{42} \\
S_4 = 1 + \alpha^4 + \alpha^8 + \alpha^{12} \ldots \ldots \alpha^{68} + \alpha^{52} + \alpha^{56}
\]

Reducing, modulo \( p(X) \), the above elements gives,

\[
S_1' = 1 + \alpha^1 + \alpha^4 + \alpha^7 \ldots \ldots \alpha^{12} + \alpha^{13} + \alpha^{14} \\
S_2' = 1 + \alpha^2 + \alpha^4 + \alpha^6 \ldots \ldots \alpha^9 + \alpha^{12} + \alpha^{13}
\]

123
Assuming the S1 syndrome has been generated then, the 14th element of syndrome $S_2' = S_{2/4}' = S_{1/4}' / \alpha$

the 14th element of syndrome $S_3' = S_{3/4}' = S_{1/4}' / \alpha \alpha / \alpha$

the 14th element of syndrome $S_4' = S_{4/4}' = S_{1/4}' / \alpha \alpha \alpha / \alpha$

In general, for the jth element of the ith syndrome, $S_{i,j}$

$$S_{i,j} = S_{(i-1),j} / \alpha^{(n-j)}$$

Also, if the syndrome $S_4$ is computed first,

$$S_{i,j} = S_{(i+1),j} / \alpha^{(n-j)}$$

As the codevector is being received one element from each of the 2t syndromes may be computed using the relationships of 3.28 or 3.29. Thus, a running sum of the syndromes may be kept resulting in the full computation of each syndrome element after the final codeword of the codevector has been received.

The second step in the decoding process involves 2t iterations of Berlekamp's algorithm [43], where t is the number of errors the code can correct. This process results in the error-location polynomial, $\sigma^{-}(x).$ Each iteration involves the computation of two equations:

$$\sigma^{-}^{(\mu \rho)}(x) = \sigma^{(\mu)}(x) + \frac{d \mu}{d \rho} x^{(\mu-1)}(x) \sigma^{-}(x) \ldots \ldots 3.30$$

and,

$$\dot{c} (\mu + 1) = S (\mu + 2) + \sigma_{i}^{(\mu+1)} S_{\mu+1} + \ldots \ldots 3.31$$
where \( \mu \) = iteration number.

\[ p = \text{previous value of } \mu \text{ such that } dp \neq 0. \]

After the appropriate number of iterations equation 3.30 is the error - location polynomial.

The execution time for this second stage depends on the errors incurred but the number of arithmetic instructions required has been estimated at \( 2t^2 \) additions and \( 2t^2 \) multiplications [42].

The roots of the error - location polynomial, \( \sigma^-(x) \), identify the error positions in the received codeword. These are found using a Chien search [44]. This involves substituting \( \alpha^0, \alpha^1, \ldots, \alpha^{n-1} \) in turn, into the error - location polynomial and checking for a zero solution. The reciprocals of the roots give the error positions.

A means of implementing a Chien search is to maintain a number of \( n \) - tuple vectors relating to the coefficients of the error - location polynomial. The variables are then replaced, in turn, by the above values and the error - location polynomial evaluated. The procedure involves \( n \) calculations for an \( n,m \) code. If the \( r \)th calculation is associated with the \( (n - r) \)th root the need to explicitly find the reciprocal of each root is removed.

The final RS decoding step is to determine the error magnitudes. These are calculated using the appropriate equations deduced in section 3.2.6. Once the error magnitudes have been determined they are modulo 2 added to the received codeword at the error positions to effect correction.
3.5. REFERENCES FOR CHAPTER 3.


129


CHAPTER 4.

EXPERIMENTAL APPARATUS AND TECHNIQUES.

4.1. INTRODUCTION.

In this chapter details of the hardware/software techniques used in the investigation are presented.

Three direct-digital tape recording systems have been designed and implemented.

System 1 comprises a solenoid-controlled compact-cassette tape transport with 8-track head interfaced to a microcomputer through read/write circuitry. This system was used to measure tape transport properties.

System 2 is similar to system 1 but a 4-track read/write head is employed and the interface circuitry is simplified by utilising the I/O ports of the microcomputer. System 2 represents a low-cost implementation of a medium density recording system.

System 3 is system 1 with additional deskew electronic circuitry employed in the replay process. The additional circuitry permits recording and detection at high data rates.

Combined hardware/software methods have been devised to measure tape-transport properties, such as, velocity, skew and tape deformation. Also presented is the software implementation of channel codes and error-correction techniques.
4.2. APPARATUS.

The basic items of equipment used in the investigation consist of a compact-cassette tape transport interfaced to a microcomputer to form the first direct digital recording system. The interface elements comprise read/write amplifiers plus tape-transport circuitry. Descriptions of these are given in this section. In support of this apparatus a number of computer programs were written. These are referred to in section 4.2.8 but are explained fully in later sections in the context of their application.

4.2.1. Compact Cassette Recording System 1.
A block diagram of the system is given in figure 4.1. with a photograph in figure 4.2. The interface circuitry comprises tape-deck control, write amplifier and read amplifier. The additional functions normally associated with a direct digital tape recorder, such as, encoder/decoder, clock regeneration etc., are implemented in software.

![Fig. 4.1. Compact cassette recording system 1.](image-url)
Fig. 4.2. Complete recording system.
4.2.2. Tape Transport.

The tape transport, figure 4.2, is a medium-quality, solenoid-controlled mechanism which accommodates proprietary compact-cassette tapes. A single motor with integral speed control drives the tape via a single pinch roller. Two solenoids control fast forward and rewind. A third solenoid may be optionally engaged in a search mode of operation. The play mode is entered by operating both the fast forward and the rewind solenoids. These engage a mechanism which utilises the stored energy of a flywheel to wind the head plate into position. Because of this, a software delay is used to make certain that the motor and flywheel are up to operating speed before the recording process begins. Tape head accessibility is facilitated by the open construction of the tape transport.

4.2.3. Solenoid Control.

The solenoids are controlled from a time delay switch designed for this purpose [1], figure 4.3. The "pull in" current of the

![Solenoid control switch diagram](image)

Fig.4.3. Solenoid control switch.
solenoids is twice the holding current. This requires a current switch with a time delay sufficient for the solenoid to operate. Two VMOS power field-effect transistors (FET) form the switch and the delay is provided by the time constant of the CR network. The delay is such that V2 will gradually turn off thus giving a "soft" switch between currents. This reduces the back emf across the solenoid and thus the problem of interfering switch transients is reduced.

4.2.4. Write Amplifier.

The design of the read/write amplifiers has been conditioned by the requirement to provide a multi-track capability at a low cost. Consideration has also been given to the need to interface these items to a microcomputer.

Two write amplifier designs have been developed, fig. 4.4 and fig. 4.5. A third design was also used and this will be dealt with in section 4.4.

Figure 4.4 shows the basic design. Open-collector logic gates drive current differentially through the record coil of the head. The value of the collector load resistors is chosen to give the required record current. This design is suitable for low to medium data rates. The bandwidth of the write amplifier is limited by the time constant of the head inductance and its series resistance, which includes the collector load resistor—this limits the upper recording frequency. The bandwidth may be improved by replacing the passive collector load resistors with active loads, fig. 4.5. The transistors act as constant current sources and the rise time of the record current is reduced.
Fig. 4.4. Write amplifier with passive loads.

Fig. 4.5. Write amplifier with active loads.
compared to the circuit of figure 4.4. A further refinement of the active-load circuit is to connect "speed-up" capacitors as shown. This reduces the turn on time of the transistors giving a further decrease in record-current rise time.

4.2.5. Read Amplifier.

The replay signals from the tape head are sliced to logic levels before application to the microcomputer.

The main read amplifier design is illustrated in figure 4.6. The reproduced signal is sliced to convert the peaks into pulses. These are gated with a coincident pulse derived from the differential of the signal [2].

The signals at each stage of the amplifier are shown graphically in 4.7. A pre-amplifier converts the high output impedance of the read head to a low impedance to drive the line connecting the tape transport to the main read amplifier. Following an initial stage of amplification the signal passes through a low-pass filter. A further stage of amplification follows after which the signal path divides. Along the first path the signal is sliced at a positive and negative level and the two outputs are applied to two NAND gates. Along the second path the signal is shifted by 90 degrees to convert the peaks of the signal to zero crossing points. These points are detected by a comparator which converts them to signal transitions. The signal-to-noise ratio at the comparator input is degraded by 6dB due to the differentiator. To prevent the noise signal generating a false output, threshold hysteresis is employed. The edges are converted
Fig. 4.6. Read amplifier.
Fig. 4.7 Read amplifier waveforms.
to pulses which are applied to the second input of the NAND gates. The output of the NAND gates are used to control a set/reset flip-flop which delivers a pulse train corresponding to the direction of the recorded magnetic flux.

The double-detection action of the circuit reduces the possibility of false triggering of the flip-flop due to points of inflection of the differentiated signal. The slice levels are adjusted to allow for the variation in signal amplitude with recording frequency. Although other, more complex, detection techniques have been proposed [3] the circuit adopted offers a reasonable compromise between performance and complexity.

4.2.6. Microcomputer.

In a commercial product the microcomputer would be mounted on a single board along with the necessary support circuits. For the purposes of the investigation a microprocessor development system was used. This unit is a Research Machines 380Z-D system which supports development of software for the Zilog Z80 microprocessor, figure 4.2. The 380Z-D system includes 512 K bytes of RAM and two, 500K-byte floppy-disc drives plus the usual software development tools. It also supports the high-level languages Basic and Pascal. An important feature is the facility to patch assembly language into a program written in high-level language. This permits the speed advantage of assembly language to be complemented by high-level functions.

4.2.7. Test Equipment.

Many of the measuring techniques used throughout the
investigation were software based and the 380Z - D was fully utilised as an item of test equipment. In addition, more powerful computing facilities were available and these were also used. For the measurement of tape velocity, skew and tape deformation, specialised circuitry was designed and used in conjunction with the 380Z - D. Consideration of these methods is deferred until section 4.3. The computer was also used to simulate the operation of a multitrack tape recorder and to examine the performance of a spatial parity system and it is these applications that will briefly be considered under this heading.

To develop the software for the decoding and synchronisation of a channel code in a multitrack format a signal source is required. All the uncertainties associated with an actual multitrack tape system may be removed if it is replaced by a computer generating the code in the required manner. Not only does this give repeatability but measured amounts of error variable may be added in order to assess the performance of the software under development. In this context a second 380Z - D development system was used. This provided "n track" outputs of encoded data whose make up and timing could conveniently be modified as required.

Darwood has proposed the use of a "moving parity" error detection system for a multitrack tape system [4-5]. An error-correction technique based on a similar principle has also been investigated by Prusinkiewicz and Budkowski [6]. These systems rely on the inter - track spatial distribution of data for their operation. A parity pattern is chosen and the parity of data
covered by this pattern is recorded on a parity track. On replay the parity check is re-applied and a syndrome is generated which is used to detect and/or correct errors. A program was written to simulate the application of this technique. This permits any parity pattern to be selected for which the computer generates syndromes for all possible error distributions. These syndromes are analysed by the program for error-pattern matching and for repeatability. Details of this technique are given in section 4.6.

4.2.8. Software.
As previously explained programmable electronics has been used extensively during the course of the investigation, consequently software forms a large proportion of the "experimental apparatus". It also forms an important element of the recording system of figure 4.1. Software is also employed to fulfill a number of measurement, display and diagnostic requirements. Rather than explain these software techniques under the heading of "apparatus" details of each program will be given later in this chapter when each application is considered.

4.3. MEASUREMENTS.
Three record heads were used during the investigation, a four-track inductive head, an eight-track inductive head and an eighteen-track, magneto-resistive (MR) head. The four-track head is a proprietary item of the type often found in auto-reverse audio cassette players. The second inductive head is a prototype device and was constructed as a "one off" by a local company. The MR head is an experimental, thin film, head and was fabricated
by a separate research group within the department of Electrical and Electronic Engineering, Plymouth Polytechnic. Prior to using these devices a number of measurements were made to assess their suitability for direct digital recording.

The tape transport properties of the record deck and tapes were characterised by measuring three parameters: tape velocity variation; tape skew and tape deformation. This required the formulation of appropriate measuring algorithms and the design and construction of specialised circuitry. Details are given of the measurement techniques adopted together with the associated circuitry.

4.3.1. Record Head Characteristics.
A photograph of the three tape heads used in the investigation is shown in figure 4.8.

![Image of three tape heads](image_url)

**Fig.4.8. Compact-cassette tape heads.**
4.3.2. 4 - Track Inductive Head.

The characteristics of the four -track head are listed in figure 4.9. These are manufacturers' figures and were taken at face value - no additional data was sought. The value of recording current used for this head was half the value obtained experimentally for the eight - track head.

### 4 - Track head specification.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Impedance</td>
<td>1 kHz., 100μA</td>
</tr>
<tr>
<td>P.B. sensitivity</td>
<td>315 Hz.</td>
</tr>
<tr>
<td>P.B. freq' resp'</td>
<td>10 kHz./315 Hz.</td>
</tr>
<tr>
<td></td>
<td>14 kHz./315 Hz.</td>
</tr>
<tr>
<td>Rec' current</td>
<td>1 kHz.-10dB below saturation level.</td>
</tr>
<tr>
<td>Crosstalk</td>
<td>1 2ch., 3 4ch.</td>
</tr>
<tr>
<td>(1kHz.-30dBm).</td>
<td>2 3ch.</td>
</tr>
<tr>
<td></td>
<td>1 3ch., 2 4ch</td>
</tr>
<tr>
<td>Difference in channel azimuth</td>
<td>10 kHz.</td>
</tr>
</tbody>
</table>

Fig. 4.9. Characteristics of 4 - track head.

4.3.3. 8 - Track Inductive Head.

A diagram of the eight -track inductive head is shown in figure 4.10. Although the head was designed to record on compact cassette tape, its physical dimensions are much larger than proprietary designs. Because of the limitation of space in such a design the use of internal, inter-gap shielding had been dispensed with. To ascertain the correct value of record current a series of recordings were made at different currents. This was achieved by returning the collector loads of the write amplifier (fig.4.4) to a variable voltage source and recording
Track width: 0.25. All dimensions in mm.
Track pitch: 0.46.
dc resistance: 25Ω.
1kHz inductance: 85mH.

Fig. 4.10. Dimensions of eight-track inductive head.
the value FF hex at 10k bits/sec./track in Miller code with the collector voltage as a parameter. A number of tapes were encoded and the amplitude of the playback signal was plotted against write current, figures 5.4 and 5.5.

To quantify the extent of inter-gap crosstalk in the eight-track head, implied by the absence of internal shielding, recordings were made on seven of the eight tracks using an erased tape. The replayed outputs of the recorded tracks were then compared against the signals picked up by the unrecorded tracks, figure 5.6.

The above measurements were used to determine a value of recording current of 1.5mA.

4.3.4. 18-Track, Thin-Film Head.
The experimental 18-track, thin-film, head has a single-turn write winding and an MR read sensing element for each track.
Track width: 150 µm.
Intertrack distance: 50 µm.
Sensing element: 81/19 Ni/Fi.
Sensing element thickness: 400 Å + 20 Å.
Sensing element resistance: 120 Ω.
Write winding: single turn, thin film.
Write gap: 0.3 µm.

The use of flux guides permits the sensing element to be located 50 µm below the tape surface of the head. This protects the element from oxidisation and also damage due to tape contact wear.
4.3.5. Tape - Velocity Variation.

The tape velocity characteristic of a tape recorder depends on both the tape - transport and the tape. This is particularly so in low tape - velocity systems where mechanical stiction is a factor. In a digital recording system, signals are encoded in both space and time. These are subject to variation on playback. The time variation depends on how accurately the tape velocity at recording can be reproduced on playback. It is the relative value of tape velocity between record and replay which is important and to which the coherent detection of digitally - encoded tapes is susceptible.

4.3.6. Theory of Tape Velocity Measurement.

The tape velocity variation was evaluated by recording a sequence of pulses whose time interval is determined by the crystal - controlled system clock of the microprocessor. On playback this time interval is compared to the recorded value to give a measure of the difference in tape velocity between record and playback. The theory of the measurement technique is as follows;

Let \( l \) = the length of tape between pulses recorded with interval \( T \) secs.

Record tape velocity \( v = 1/T \) metres/sec. ........ (4.1)

Let \( t \) = the time between pulses as detected on playback.

Playback tape velocity \( v' = 1/t \) metres/sec. ........ (4.2)

from equations (4.1) and (4.2),

\[
\begin{align*}
vT &= v't \\
v &= v't/T
\end{align*}
\]
\[ v - v' = v' t / T - v' \]
\[ v' - v = v' - v' t / T \]
\[ v' - v = \Delta v = v' ( T - t ) / T \quad \ldots \quad (4.3) \]

Equation (4.3) expresses the difference between the replay tape velocity and the record tape velocity in terms of the replay tape velocity. This equation was implemented with the aid of the circuit in figure 4.11.

**Fig.4.11. Tape-velocity measuring apparatus.**
4.3.7. Tape Velocity Measuring Apparatus.

The system of figure 4.1, fitted with the four-track head, was used to record a sequence of pulses on a single track. The frequency of these pulses was determined by a programmed timing loop which was part of the recording software. The timing loop of the record programme also formed part of the replay software. This controls the accumulation of clock pulses into register one of figure 4.11. At the end of this process register one contains a value which is proportional to the time interval between the pulses on the tape as they were recorded.

The number of clock pulses accumulated by register one = \( fT \).

Where \( f \) is the frequency of the clock.

Let this value be \( N_1 \), therefore \( T = N_1/f \) .......(4.4)

This value is transferred to register two which is counted down by the same clock source for a period determined by the recorded pulses as they are played back. Should the velocity of the tape on playback be identical to the velocity when the recording took place the contents of register two after countdown will be zero. A negative value would indicate a slowing down of the tape whereas a positive value indicates a higher velocity on playback compared to recording. The actual value in register two after countdown is a measure of the magnitude of tape velocity difference between record and playback.

The number of clock pulses remaining in register two after countdown = \( f(T - t) \).

Let this value be \( N_2 \), therefore \( (T - t) = N_2/f \) .......(4.5)

Substituting equations (4.4) and (4.5) in equation (4.3):
This equation is independent of the clock frequency $f$ and also the system clock which generates the software timing loop (assuming the latter remains constant for both the record and replay process).

Choice of the length of the registers one and two determine the resolution of the measuring technique:

Capacity of register one $= 2^n - 1$ counts.

Resolution of register two $= 1$ count.

Resolution of technique $= v' \frac{1}{2^n - 1}$ metres/sec.

An eight-bit register was chosen and this gives a resolution of, $v' \frac{1}{255}$ i.e. < 0.5% of replay velocity.

For the measurement system to give as near an instantaneous indication as possible the interval between recorded pulses, $T$, must be short. Also, register one must not overflow. The relationship between these variables is,

$$f = \frac{2^n - 1}{T}.$$ 

A value for $T$ of 0.5 mS; gives a tape length of 24$\mu$m (at standard cassette speed) over which velocity measurements are repeated. This value of $T$ gives a maximum value for $f$ of 510 kHz. A frequency $f$ of 500 kHz. was used with a recorded pulse time-interval $T$ of 0.4 mS. This gives an approximate weighting value of 0.5% to each count in register two.

Register control pulses are provided from data recorded on the tracks that are not used for timing purposes. As register two is counting down register one is accumulating another value, thus the process is continuous.
The accuracy of the measuring technique depends on the stability of the system clock of the microprocessor. It is also subject to the usual error of ± one count in registers one and two. The estimated level of accuracy, therefore, is

\[ 0.5\% < \text{accuracy} < 1.0\% \]

4.3.8. Tape Skew.

Although the tape-skew behavior of high-cost computer tape systems has been studied [7], to the authors' knowledge, no similar study has been made with compact cassette systems as the subject. Consequently, the solutions to tape skew which have been proposed, such as single edge guidance [8] and phase comparison, apply to open reel tape systems and not to closed reel systems such as the compact cassette. Skew angle variation imposes one of the limits on the upper recording frequency at which digital data may successfully be read back from a multitrack tape. Therefore, to assess the potential of a digital multitrack format for compact cassette it was necessary to examine its skew behavior.

A precise technique of measuring the skew performance of a closed reel, stationary-head tape system was devised. This involved the development of a measuring algorithm and its implementation using hardware and software. Details are given in this section.

4.3.9. Skew Angle.

Skew angle is the angle between the normal to the longitudinal axis of the tape and the axis of the sensing head gaps (fig.4.12). It comprises two components — static skew and
dynamic skew. As with tape velocity it is not the absolute value of skew angle which is important but the variation which occurs between record and playback. Skew angle variation results in the data across the tape not being presented to the sensing head gaps in an identical manner to that laid down when the data were initially recorded. Consequently, the outer head gaps could be simultaneously sensing data displaced by several bit cells compared to when they were recorded. The extent of the displacement depends on the frequency of the recorded signal since as this increases the distance on tape occupied by one bit decreases.

![Diagram of tape passing read head with skew angle](image)

Fig. 4.12. Tape passing read head with skew angle $\alpha$.

4.3.10. Theory of Skew Measurement.

Consider a tape with marker pulses recorded on the outer tracks at a constant interval $k$ (fig. 4.13.). Let $T_a$ and $T_b$ represent the time intervals between the marker pulses on opposite tracks appearing under their read heads on playback. From the diagram of figure 4.13, the displacement $x$ may be determined:
Fig. 4.13 Marker pulses on outer tracks of tape from which bit-cell displacement may be calculated.

\[ \cos \alpha = \frac{(T_a - T_b)}{2x} \quad \ldots \ldots \ldots \ldots (4.7), \]

also,

\[ \cos \alpha = \left[ T_b + \frac{(T_a - T_b)}{2} \right]/k \quad \ldots \ldots (4.8), \]

where \( \alpha \) = skew angle.

\( x \) = displacement.

\( k \) = separation of marker pulses.

Thus,

\[ \frac{(T_a - T_b)}{2x} = \frac{(T_a + T_b)}{2k} \]

and,

\[ x = k(T_a - T_b)/(T_a + T_b) \quad \ldots \ldots (4.9). \]

The units of \( k \) determine the units of \( x \) and may be distance or time.

The above technique measures the effects of skew-angle variation, however, from the knowledge of the distance between the tracks the skew angle may be determined if required.
4.3.11. Skew - Angle Measuring Apparatus.

The microcomputer development system was utilised to calculate, and display, bit - cell displacement \( x \), equation (4.9). Two timers were constructed and interfaced to the microcomputer, fig.4.14.

Each timer accumulates a value proportional to the time interval \( T_a \) and \( T_b \) as defined in figure 4.13. The timer - control logic generates enable pulses which correspond to the correct sequence of pulses on the outside tracks of the pre - recorded tape, fig. 4.13. The timers are reset to zero by initialise pulses recorded on an inside track and positioned to appear before each set of timer control pulses. The output of the timers is connected to the input ports of the microcomputer.
A software polling technique is used to read in the timer outputs. Each timer value is read twice and a comparison made, only when each comparison is valid does the software accept the two values. These are then used to compute a value for x which is subsequently displayed in graphical form on the VDU screen after which the process repeats. The spacing of the marker pulses, k, was chosen such that the bit - cell displacement x was measured in terms of the percentage of one bit cell at 10 k. bits/sec.


On the assumption that k in equation (4.9) remains constant the theoretical worst - case fractional error in the measurement of skew is,

\[
a/Ta + \frac{a}{Tb} + \text{slicing error in read amplifier} + \text{computational error} \ldots (4.10).
\]

Numbers are handled by the microcomputer with a precision better than six significant decimal places.

The marker pulses from each of the outside tracks are passed through a pair of read amplifiers. If the slice levels of these amplifiers differ then the time length of their output pulses will differ even though the time interval between each set of input transitions is the same. However, timers A and B are started and stopped by read - amplifier output pulses which have been derived from opposite tracks, thus any error due to this difference will accrue to both timers thus reducing the potential error from this source. Also, any tape - velocity variation will
affect both timers with the same result on the error contribution.

Eight-bit registers are used for timers A and B giving a maximum count of \(2^8 - 1\). On the basis that the timers are susceptible to an error of \(\pm 1\) count the error contribution from this source is,

\[
\frac{1}{255} + \frac{1}{255}.
\]

To prevent overflow, limit the maximum value to 200 counts, the estimated error becomes,

\[
\frac{1}{200} + \frac{1}{200} = \frac{1}{100}.
\]

This is the dominant error source of equation (4.10). In practice it is likely to be less than this 1% since the calculation involves the ratio of two variables.

The program which calculates the tape skew assumes a constant value of \(k\). However, although the marker pulses have been recorded at a constant interval \(k\), this value is a function of tape velocity on playback. If necessary this value may be measured dynamically and used in the calculation.

To verify the measurement technique, skew angles in the range \(+0.3^\circ > \alpha > -0.3^\circ\) were introduced using a rotary micrometer, fig. 4.15. This represents a bit - cell displacement of four units at 10 k.bits/sec. The micrometer consists of a circular scale of radius 21 cm. graduated in degrees. The tape transport is placed inside the circular scale with the head azimuth adjust screw at its centre. Connected to the screwdriver, used to adjust the head azimuth, is a pointer which aligns with the circular scale. From a knowledge of the thread pitch of the
azimuth adjust screw and the distance between the mounting points of the head, together with readings from the circular scale, a precise value of skew angle may be deduced.

A computer program was written which measured and indicated static skew by measuring and averaging skew angle variation. From this the skew angle for a number of settings of the micrometer were calculated. The relationship between the mechanically introduced values and the computed values is shown in figure 4.16 which shows a high degree of correlation between the ideal and the actual results.
4.3.13. Tape Deformation.

The skew measurement technique described above measures the tape skew with reference to marker pulses positioned on the outer tracks of the multitrack tape. The azimuth error between data on the outer tracks is not necessarily the maximum. It has been shown that with the multitrack, open-reel tapes used in computer data recording the skew angle does not remain constant across the width of the tape [7]. In order to determine the skew angle distribution for the compact cassette system a series of measurements were taken using an item of equipment designed to measure the phase difference between pulses recorded on eight
tracks. This revealed that the skew was not always linearly distributed across the width of the tape. In these cases the tape exhibited a deformation along its centre tracks. Details of the measuring technique and the equipment used are given in this section.

An "in-phase" square waveform was recorded on each track of a multitrack tape. On replay, the phase difference between detected signals from adjacent tracks was measured. The relative values of these measurements gives an indication of the skew-angle distribution across the width of the tape.

4.3.15. Tape - Deformation Measuring Apparatus.
The recording system of figure 4.1 was used to record a 10 kbits/sec, in-phase square waveform on eight tracks. Associated with each track is a seven-bit counter timer each of which is clocked from a common source (fig. 4.17). The phase difference between the detected signals of two adjacent tracks is used to control the timing signal to each counter. The polarity of the phase difference is recorded as an eighth counter bit. When all eight counter timers have accumulated their values a flag is set and the outputs of the timers are multiplexed into the microcomputer under software control.

The computer program controlling the above process displays the data and computes a numerical value for the tape deformation. The display format is illustrated in figure 4.18. The horizontal axis is calibrated in time and the vertical axis represents the width...
Fig. 4.17. Tape deformation measuring equipment.
of the tape. A centre-zero format is used to allow for "leading and lagging" values of skew. Track one is taken as the reference track and the time of occurrence of a pulse transition on this track is positioned at the top centre of the screen. The time difference between pulse transitions on other tracks and transitions on track one are displayed at their appropriate vertical position, these points are joined to give a visual indication of the profile of the skew across the tape.

The deformation of the tape is defined as the worst-case difference, in time or space, between the point at which a track transition occurs and the point at which it would have occurred had the tape skew been linearly distributed across the tape.
figure 4.18. This value was measured by the computer and successive readings were averaged and displayed.

4.3.16. Analysis of Measuring Technique.
A frequency of 10kHz was chosen for the squarewave signal which was encoded onto the tape. This gives a sample interval of $1/2 \times 10^{-4}$ secs. However, this is effectively increased by the time taken for the computer to read each value and display the results. The clock source is a 1 MHz crystal-controlled oscillator which gives a resolution of 1 μS. Discounting computational error the only sources of error are the tolerance on the crystal oscillator frequency, the errors introduced by the difference in the slicing levels of the eight read amplifiers and the usual error of ± one count. The error due to the oscillator tolerance is small compared to the others and may, therefore, be neglected.

To assess the effect of differences in the slicing levels of the read amplifiers the input to the amplifiers were disconnected and replaced by a 10k Hz sinusoidal signal from a generator. This was simultaneously applied to all the inputs of the read amplifiers to simulate an "in - phase condition". The dispersion of the displayed readings about the vertical line of the read gaps gave an indication of the difference between the slice level of the track-one read amplifier, the reference track, and the other tracks. This value did not exceed 5% of the full scale value. This full scale value is determined by the length of the counter timers which is seven bits, i.e. 128. One bit in 128 is less than 1% and so the accuracy of the measuring technique is conservatively stated as ± 6%.
4.4. COMPACT - CASSETTE RECORDING SYSTEM 2.

The use of a microcomputer to implement system functions reduces the need for conventional hardware at the expense of more complex software. However, software development is a "one off" cost and although a microcomputer is required these are commonplace and widely used in many applications in the form of the personal computer (PC). Also, virtually all the requirements for digital-data storage include an associated computer. The utilisation of the computing power of a PC would, therefore, offer a cost effective means of configuring a data recording system.

This section is concerned with the design and development of a simple tape backup recording system for a personal computer (PC). Software techniques have been developed which reduce the conventional hardware requirement to a minimum. Particular attention has been paid to realising a practical system. A patent application for this low-cost system has been applied for and a paper based on the techniques used has been presented to the Euromicro '87 conference at Portsmouth, U.K. [9].

4.4.1. Hardware.

Figure 4.19 illustrates the system. The tape transport is fitted with a proprietary four-track head and may be the solenoid-controlled tape deck described in section 4.2.1. or a manually controlled deck. The essential difference between this system and the system 1, described earlier, is the write amplifier.
4.4.2. Write Amplifier.

The write amplifier is formed from the parallel input/output (PIO) integrated circuit within the PC, figure 4.20. Data to be

Fig. 4.20. Write amplifier.
recorded are outputted by the PC to the PIO which is programmed in the output mode. The direction of the recorded flux is determined by the complementary values of the logic levels sent to the bit positions of the PIO to which each end of a recording-head winding is connected. The value of the resistor \( R_r \) determines the magnitude of the recording current. If the PIO has been fabricated using tri-state logic it may be programmed to the third state thus isolating the record/replay head. This permits the usual mechanical switching of the common head between the record and read circuitry to be dispensed with.

4.4.3. Read Amplifier.

Data from the read amplifier are read into the PC via the second port of the dual PIO. For convenience the read amplifier used in this system is the one illustrated in figure 4.6. However, to maintain the low-cost principle a simpler design may be adopted, figure 4.21. For the four-track system under

![Fig. 4.21. Simplified read amplifier.](image)
consideration this design may be implemented using two quad integrated circuits and a number of passive components. This simple circuit may be powered from the auxiliary power supply of the PC.

4.4.4. Software.

There are several functions the software must fulfill:

(a) Encode data and time its recording.
(b) Synchronise to the data on replay.
(c) Sample the data and decode on replay.

The above must be accomplished under conditions of tape azimuth and velocity variations. An additional, diagnostic function, is to measure the performance of the system.

Numerous codes have been devised for the recording of data on magnetic media [10-14]. The effectiveness of the system and the complexity of the software hinges on the choice of a suitable recording code. The replay process is the most demanding and in effect dictates the choice.

Three codes have been used, Biphase - L, Miller and a 2/3 rate 0.2 code developed for this application. Each will be considered in this section along with a description of the diagnostic technique used.


Biphase - L, or Frequency Modulation (FM) code, belongs to the Manchester group of codes and was one of the first self-clocking codes to be developed for direct digital recording [15]. The
coding rule is illustrated in figure 4.22. The code has a d,k value of 0,1 and a code rate of 1/2. Whilst the low k value facilitates data recovery, the low value of d implies a potential for the code to generate intersymbol interference should the recording frequency be placed too high. The code is run length limited and the detection window is one half of one bit cell.

As implied above the spectral response of the code shows a broad maximum at the bit rate, (fig.4.23). Also, the d.c. component is zero giving a digital sum variation (DSV) which converges to zero independently of the data content. Biphase-L code has largely been superseded by more channel-efficient recording codes. However, it has a number of features which make it suitable for software implementation in this application. Reference to figure 4.22 shows that the signal level always changes at bit-cell centre. Also, the ratio of time intervals between successive transitions can be used to identify the bit -
cell centre or the bit-cell boundary. If the ratio of time intervals between three successive signal transitions is 2:1 the third transition always occurs at the bit-cell boundary, similarly a 1:2 ratio signals the bit-cell centre.

4.4.6. Biphase-L Recording Software.

The parallel bit structure of the PC permits each track of the tape to be recorded simultaneously, one bit of the processor word per track. The internal counter timer circuit (CTC) of the PC is used to time the recording of the complement of each data bit for half a cycle followed by the true values for the remainder of the period.
4.4.7. Biphase - L Playback Software.

The playback process is far more demanding of processor time than the record operation. The playback software must:
(a) Identify each bit-cell centre or boundary.
(b) Remain synchronised to the data throughout replay.
(c) Simultaneously sample the four data tracks and decode the data.

Bit-cell identification is achieved by timing the interval between successive transitions of the recovered signal and computing the ratio between successive intervals. As mentioned above, the third transition of a 1:2 ratio occurs at the bit-cell centre whilst that of a 2:1 ratio marks the bit-cell boundary. 1:1 ratios mark either the bit-cell boundary or centre and are ignored by the software.

The software polls the output of the read amplifier and detects the point where one of the four outputs change. The counter timer circuit within the PC is then used to measure the transition intervals. To speed up the computation of ratios the shorter subtraction and addition instructions of the PC are used in preference to the division instruction. If the result of subtracting two successive intervals is positive, the ratio is either 2:1 or 1:1. Similarly, a negative value would indicate a ratio of either 1:2 or 1:1. The ambiguity is resolved by a further subtraction of one half of the second interval in the first case or by adding one half of the first interval when the initial result is negative, e.g.

169
Thus, maximum discrimination between wanted and unwanted ratios is achieved. Once bit-cell identification has been achieved the playback software continuously monitors the ratios of successive intervals to maintain synchronism throughout replay. The use of ratios in this way gives a system which is independent of tape velocity.

To decode the data, the outputs of the four tracks are simultaneously sampled at the three quarters point of each bit cell. The instant at which the software samples is determined from a knowledge of the duration of the current bit cell. This is stored and updated when either a 2:1 or a 1:2 ratio is measured. Before the sample point is updated the value of the measured bit interval is compared to that previously measured by the software. If it is within acceptable limits it overwrites the previously measured value. If not, it is ignored and the previously measured value continues to be used. The program thus behaves as an intelligent phase-locked loop by only synchronising to the data deemed to be acceptable. As in the case of synchronisation the determination of the sample point is independent of tape velocity.

In the playback software described above the sample point is determined using the information from a single track. The outputs
of the four tracks will, however, be "out of phase" due to tape skew. Typically, the bit cell displacement across the tape is of the order of $\pm 1/4$ bit cell at a track data rate of 2.5k.b/sec. [16]. Thus, when a single, fixed track is used to determine the common sample point, the tape skew variation gives a sample window of $+1/4$ bit cell at 2.5k.b/sec. (fig.4.24). This sample window was doubled by programming the software to respond to the skew dynamics of the tape.

The software synchronises to the signal on the track which changes first - the leading track. Transitions of the signals on each track are detected by polling the signals and performing an exclusive OR operation on successive samples until a change occurs. Since the signals on each track always change at the bit cell centre the leading track may be determined. Once the leading track has been determined the software samples all four tracks as late as possible after the detection of the leading track bit cell centre (fig.4.25). Provided the other tracks have changed within 1/2 bit cell the sample will be valid. Since the tape skew "oscillates" between positive and negative values this sampling arrangement effectively doubles the tolerance to tape skew compared to when the software synchronises to a fixed track.

The rate of tape skew variation is low compared to the recording frequency [17] and so the leading track remains so for several thousand periods of the recorded signal. Even so the software determines the leading track each time a 2:1 or a 1:2 ratio is detected.
Soft~are synchronised to fixed track

Sample timing window: $\pm\frac{1}{4}$ bit cell

Fig. 4.24. Sample timing window is $\pm\frac{1}{4}$ bit cell when software synchronised to a fixed track.

Software synchronised to leading track

Sample timing window: $\pm\frac{1}{2}$ bit cell

Fig. 4.25. Sample timing window is $\pm\frac{1}{2}$ bit cell when software synchronised to the leading track.

Miller code (also known as Modified Frequency Modulation, MFM), like FM, has a code rate of 1/2. However with a d,k value of 1, 3 data may be recorded at twice the data rate of Biphase - L under the same channel conditions. The increased value of k however, does make data recovery more difficult. Like Biphase - L, Miller code is run length limited and the detection window is the same. However, Miller code is not d.c.free. Figure 4.26 illustrates the coding rules - logic ones are encoded as a transition, either positive going or negative going; a transition is placed between pairs of zeros; zeros which occur in isolation are ignored. The frequency spectrum of Miller code is shown in figure 4.27.

Fig. 4.26. Miller code coding rules.
255-long PRBS
1kb/s. data rate.

Fig. 4.27. Spectral response of Miller encoded PRBS.

4.4.9. Miller Code Recording Software.

The recording algorithm for this code is deduced from a consideration of the four possible recording sequences. Transitions can occur at the bit cell limit and bit cell centre, the next bit is the data bit to be recorded.

(a) Next bit = 1, previously recorded bit = 0:
Next cell limit = previous cell centre.
Next cell centre = complement of previous cell limit.

(b) Next bit = 1, previously recorded bit = 1:

![Diagram for case (b)](image)

Next cell limit = previous cell centre.
Next cell centre = complement of previous cell limit.

(c) Next bit = 0, previously recorded bit 0:

![Diagram for case (c)](image)

Next cell limit = complement of previous cell centre.
Next cell centre = previous cell limit.

(d) Next bit = 0, previously recorded bit 1:

![Diagram for case (d)](image)
Next cell centre = previous cell limit.
Next cell limit = previous cell centre.

The above coding rules may be expressed in terms of Boolean functions.

Let, the previous cell - centre logic level = A,
the previous cell - limit logic level = B,
the previous recorded data bit = C,
the data bit to be recorded = D.

Next cell centre = \( \overline{B} \) ( C.D + \overline{C}.D ) + B ( \overline{D}.C ) + B ( \overline{C}.D ).

This reduces to,
Next cell centre = \( \overline{B}.D + B.\overline{D} \) ...........(4.11).

Next cell limit = A ( C.D + \overline{C}.D ) + A ( \overline{D}.C ) + \overline{A} ( \overline{C}.D ).

This reduces to,
Next cell limit = A ( \overline{C}.D ) + \overline{A} ( \overline{C}.D ) ....(4.12).

From equation (4.11) the centre of each bit cell is encoded by performing an exclusive OR function on the data bit to be recorded, D and the previous cell - limit logic level B.

To encode the bit - cell limit, the previous cell - centre logic level, A is exclusively ORed with the AND function of the complements of the previous and present data bits, equation (4.12).

These encoding functions are easily implemented in software. The recording program maintains a "bit mask" of the previous recorded output, either cell limit or cell centre. The data to be encoded is processed in accordance with equations (4.11) and (4.12).
before being written to tape. This now becomes the mask for the data to be encoded.

A counter timer circuit within the computer times the generation of data. The simplicity of the algorithm, coupled with the short execution times of the logic functions involved, permit a high data recording rate to be achieved with a microcomputer of modest speed. The algorithm is fully parallel with one bit of the computer's word handling the data for a single track:– for a computer of word length \( n \), \( n \) data tracks may be simultaneously recorded.

4.4.10. Miller Code Playback Software.

A similar technique to that described in section 4.4.7. was used in the playback software.

The bit-cell centre of the replayed data may be uniquely identified by detecting a bit sequence of 101. This represents an interval between two successive transitions of two bit periods. The software detects a ratio of either 2:1 or 1:2 to establish bit cell centre. The bit cell discrimination is not as high as with Biphase - L code since, with Miller code an interval of 1.5 is commonplace. Also, the sequence of bits which represent either a 2:1 or a 1:2 ratio do not repeat as frequently as those producing the same ratio in Biphase - L code.

Once bit cell identification is made the software remains synchronised by continuously monitoring the transition interval ratios to register the bit-cell centre point when the appropriate ratio is measured.
To decode, the recorded information samples are taken at the one quarter and three quarter points in the bit cell. The data originally encoded is the exclusive OR of these samples. Unlike Biphase - L code, signal transitions do not always occur at the same point within a bit cell, therefore, the replay software cannot adapt to the variable tape skew.

4.4.11. 2/3 Rate 0.2 Code.

The Biphase-L code offers certain attractions when used in this application: signal changes always occur at bit - cell centres; a ratio of transition intervals which can be used to identify bit-cell position frequently occur. The code rate is, however, only 1/2 and in terms of channel capacity the efficiency of the code is 72%. On the other hand Miller code has a higher efficiency, 91%, but its code rate is also 1/2. The lower bandwidth required by Miller code does permit a higher data rate to be achieved when considering a single channel. However, if the Miller encoded tracks are sampled in parallel the detection window is halved if an attempt is made to utilise this greater signalling capability. Figure 4.28 shows the reduced detection window which occurs between adjacent tracks. The bit-cell spacing is in terms of that used in the Biphase - L example given previously.

If the requirement for the code to be d.c. free is waived a code of rate greater than 1/2 and having characteristics suitable for systems 1 and 2 may be designed.

A rate $\frac{2}{3}$ code was devised, Fig. 4.29. A change always occurs at the limit of the first sub cell. The levels during the second two subcells represent the encoded data. The mandatory change in each cell group renders the code run-length limited and provides a means for the decoding software to identify the leading track. Data synchronisation is achieved by detecting transition ratio intervals of either 3:1 or 1:3. This code has a $d,m$ value of 0.2 which gives a code efficiency of 74%.

A code of rate $\frac{3}{4}$ may also be devised using the same coding algorithm. Now, there are four sub cells. The first is reserved for
a data change whilst the remaining three encode data, this increases the code efficiency to 79%. Transition interval ratios of 4:1 and 1:4 identify each group of data bits. In general, for a d,m code data may be identified by detecting transition interval ratios of $(d+1):(m+1)$ and $(m+1):(d+1)$.

4.4.13. 2/3 Rate 0,2 Code Recording Software.
Before encoding each three-bit data sequence it is necessary to look ahead to the second sub cell level to determine the required transition direction. Consequently, the first sub cell is encoded as the inverse of the second sub cell.

The counter timer circuit within the computer is used to determine the duration of each sub cell.

4.4.14. 2/3 Rate 0,2 Code Playback Software.
The method of detecting and decoding the software is the same as that used for Biphase - L and Miller code.
4.4.15. Evaluation of Recording System 2.

Two diagnostic software methods were used to evaluate the performance of the recording system. These measure the error distribution across the tracks and compile a record of the error burst lengths. Both involve a data-gathering, assembly language routine which is patched into a high level graphics program.

4.4.16. Distribution of Track Errors

Previous work on open-reel multitrack digital tape recorders has indicated that errors are largely confined to single tracks [18]. To ascertain the position with the system under test a repeated data sequence was recorded in Biphase-L code using the eight track head. On playback the sequence was checked and any differences between the recorded and the detected sequence were logged by the playback software. At the end of the replay period the errors were displayed on the VDU of the microcomputer as single, double, triple or quadruple track errors.

4.4.17. Error Burst Length.

A number of tapes were recorded with a four-bit pseudo random binary sequence (PRBS). The sequence was generated by software simulation of a pseudo random binary generator (PRBG), figure 4.30. The sequence was recorded at various bit rates in Biphase-L code. The same PRBG was maintained by the playback software. So too was a 256-wide section of memory addressed by a memory pointer. Each byte of this memory field corresponded to burst errors of length 1 to 255. The operation of the software algorithm which measures
and logs the burst errors is best described by a flow chart, figure 4.31. As each four-bit PRBS value is read from tape it is compared to the software generated PRBS. An unfavourable comparison leads to the read value being transferred to the software PRBG and the memory pointer being incremented. After the burst length, plus one, a favourable comparison is made, one is subtracted from the memory pointer and the memory location thus addressed is incremented. At the end of the replay period a histogram of error burst lengths is displayed.

4.5. COMPACT - CASSETTE RECORDING SYSTEM 3.
To compensate for the effect of tape skew a hardware technique is applied which measures the extent of the prevailing skew and generates a skew-correction signal. This signal is used to de-skew the data by introducing the right amount of skew into the
Start

Read value from tape.

Does the value compare to PRBG?

Was the last reading an error?

Subtract one from pointer address.

Increment location addressed by pointer.

Reset address pointer.

Transfer reading to PRBG.

Increment memory address pointer.

Shift PRBG
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 using the eight-track head.

The code used in this process is the ISS 2/3 code developed by Jacoby and Kost[19] for which a parallel encoding/decoding software algorithm has been devised. This de-skew technique was presented to the Sixth International Conference on Video, Audio and Data Recording, University of Sussex, in March 1986 and has been published in the paper "High density data storage on audio compact cassette tape using a low-cost tape transport"[20].

Details of the ISS 2/3 code are given in section 3.1.3. Presented here are details of the ISS 2/3 code parallel encoding/decoding software algorithm, the method of synchronising to the information on tape and the skew-correction hardware design.

4.5.1. ISS 2/3 Software Algorithm.

Control pulses recorded on the outside tracks of the tape are used to estimate the skew value and identify each three-bit group of code bits. The encoding process involves converting two, 6-bit data words into three, 6-bit codewords and recording them, broadside, onto the six inner tracks of the tape. The control pulses must be inserted at the correct points on the outside tracks whilst the data are being recorded.

The decoding process involves identifying each 3-bit codeword and converting it back to 2 data bits. This process must be applied in parallel to six, 3-bit codewords.
4.5.2. Encoding Algorithm.
The encoding 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 fails with certain illegal sequences of data bits. These are the sequences which give two adjacent code ones, figure 4.32. These illegal code sequences are replaced by a code sequence which always terminates with three zeroes.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Data</th>
<th>Illegal code</th>
<th>Replacement code</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0000</td>
<td>101101</td>
<td>101000</td>
</tr>
<tr>
<td>b</td>
<td>0001</td>
<td>101100</td>
<td>100000</td>
</tr>
<tr>
<td>c</td>
<td>1000</td>
<td>001101</td>
<td>001000</td>
</tr>
<tr>
<td>d</td>
<td>1001</td>
<td>001100</td>
<td>010000</td>
</tr>
</tbody>
</table>

Fig.4.32. Illegal code bits and their replacement

The encoder program looks forward over four data words and forms a mask identifying the illegal sequences. These are readily identifiable by consecutive zeroes 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 zeroes 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 figure 4.33 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.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Data</th>
<th>Data modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>0001</td>
<td>01XX</td>
</tr>
<tr>
<td>d</td>
<td>1001</td>
<td>11XX</td>
</tr>
</tbody>
</table>

Fig. 4.33. Data modification required  
(X = don't care.)

The control pulses are encoded onto the outer tracks of the tape by exclusively ORing the composite 8-bit word to be recorded with the mask 10000001 prior to encoding the third codebit. The outer-track control bits are initialized to zero before the recording process thus ensuring a control-pulse frequency of one third the codebit rate.

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 software encoding algorithm permits recording rates in excess of 20 kb/s per track to be achieved using the basic Z80-based microcomputer described in section 4.2.1.

4.5.3. Decoding Algorithm.

The decoding rules for the ISS 2/3 code are shown in figure 4.34. To decode each three-bit group of codewords it is necessary to look ahead over two codewords and look back over one codeword bit. Thus, seven codewords have to be examined for every two data words that are decoded.

<table>
<thead>
<tr>
<th>Previous codeword last bit</th>
<th>Present codeword</th>
<th>Succeeding codeword</th>
<th>Decoded dataword</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>101</td>
<td>NNN</td>
<td>00</td>
</tr>
<tr>
<td>X</td>
<td>100</td>
<td>NNN</td>
<td>01</td>
</tr>
<tr>
<td>X</td>
<td>001</td>
<td>NNN</td>
<td>10</td>
</tr>
<tr>
<td>X</td>
<td>010</td>
<td>NNN</td>
<td>11</td>
</tr>
<tr>
<td>X</td>
<td>101</td>
<td>000</td>
<td>00</td>
</tr>
<tr>
<td>X</td>
<td>100</td>
<td>000</td>
<td>00</td>
</tr>
<tr>
<td>X</td>
<td>001</td>
<td>000</td>
<td>10</td>
</tr>
<tr>
<td>X</td>
<td>010</td>
<td>000</td>
<td>10</td>
</tr>
<tr>
<td>0</td>
<td>000</td>
<td>XXX</td>
<td>01</td>
</tr>
<tr>
<td>1</td>
<td>000</td>
<td>XXX</td>
<td>00</td>
</tr>
</tbody>
</table>

Fig. 4.34. Decoding table (X = don’t care, N = not all zeros).
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 summarised in Figure 4.35. The decoder

<table>
<thead>
<tr>
<th>Present codeword</th>
<th>First decoded data bit</th>
<th>Second decoded data bit</th>
<th>Succeeding codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>010</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>000</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>000</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 4.35. Summary of decoding rules.
program maintains 'non-zeros' masks for each 3-bit code sequence and uses these to generate the appropriate data bits. 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.

4.5.4. Codebit Synchronisation and Detection.
The signals on the six inner tracks, representing the codewords, plus the two outer-track control pulse signals, are sensed and sliced to logical-signal levels by the read amplifier of figure 4.6. The output of each of the six "codeword" read amplifiers is used to synchronise a locally-generated clock, figure 4.36 [21].

![Locally generated clock diagram](image)

Fig. 4.36. Locally generated clock.

The above clock signals control the flow of information from each codebit track to the microcomputer via a clock switch. One clock switch is associated with each of the six codebit tracks and is necessary to allow for tape skew.
4.5.5. Clock Regeneration.

Each track clock is formed from a type 555 timer in the astable mode triggered from each transition of the input. The timing components are selected to give the correct clock frequency with a waveform which is slightly asymmetric. During the absence of input transitions the timer "free-wheels".

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.

4.5.6. Clock Switch.

One of the six clock switches is shown in figure 4.37. The
regenerated clock is switched between two data latches the data inputs of which is the output of the read amplifier. The clock is also switched between two "flag" latches each of which is set when its corresponding data latch has received a codebit. The outputs of the data latches are passed to the microcomputer, along with the five other data latch outputs, via a 6-bit multiplexer under the control of two 6-input AND gates which detect when six bits of a codeword have been loaded into their respective data latches.

The clock switching is controlled by a toggle which operates at codebit rate but 180 degrees in advance of the regenerated clock. Each of the six toggles may be reset independently of each other. On the assumption that tape skew is zero the six toggles will operate in phase and six codebits will be simultaneously clocked off each codebit track into their respective data latch. The associated flag will be set and, whilst the 6-bit codeword is input and processed by the microcomputer, the clock switch will load the next 6-bit codeword into the second data latch and so on. In the event of tape skew the above process will continue but the clock-switch control toggles will move out of phase to an extent depending on the tape skew.

The toggles must be initially synchronised to their respective codebits when tape replay begins. Furthermore, each toggle must remain synchronised and maintain its relationship with the five other units as tape skew varies under conditions of possible signal dropouts and spurious pulses. The clock-switch control toggle must therefore, receive periodic skew correction pulses to "update" their phase in accordance with the prevailing skew.
4.5.7. Clockswitch Synchronisation.

Six skew correction pulses are generated which are phase displaced over a period of time equal to the time difference between the codebits across the tracks due to tape skew. This requires a measurement of tape skew to be made.

Reference to equation (4.9) gives,

\[ x = k \frac{(T_a - T_b)}{(T_a + T_b)} \]  \hspace{1cm} (4.9)

Also,

\[ \cos \alpha = \frac{[T_b + (T_a - T_b)/2]}{k} \] \hspace{1cm} (4.8)

In compact cassette tape systems the skew angle, \( \alpha \), is in the range \(-0.1^\circ < \alpha < +0.1^\circ\) [17]. Therefore, taking \( \cos \alpha \) as 1,

\[ k = T_a - \frac{(T_a - T_b)}{2} \] \hspace{1cm} (4.13)

\[ k = \frac{(T_a + T_b)}{2} \] \hspace{1cm} (4.14)

Substituting equation (4.13) in equation (4.9) gives a measurement of bit cell displacement in terms of time:

\[ x = \frac{(T_a - T_b)}{2} \] \hspace{1cm} (4.15)

The time \( (T_a - T_b) \) may be obtained by exclusively ORing the control pulses on the outer tracks (fig. 4.13).

The skew correction pulses are generated by the circuit of figure 4.38. 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
Fig. 4.38. Skew-correction pulse generator.

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, i.e.

Shift register input frequency = \( \frac{f}{f(T_a-T_b)/2N} \)

= \( \frac{2N}{(T_a-T_b)} \)

= \( \frac{N}{x} \) Hz.

The input pulses to the shift register, and consequently the shift register output pulse sequence, are thus spaced precisely over the time period represented by the bit-cell displacement, whatever value this may take. (fig. 4.39).
Fig. 4.39. Skew correction pulses.

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, therefore 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 skew correction 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.

The control pulses on the outer tracks are also used to set a codebit-group flag every three codebits. This is polled by the decoding software and used in the decoding process.

4.6. ERROR CORRECTION.

Since increasing the data density in a recording system will result in a corresponding increase in error rate a necessary requirement for a viable system is the application of error correction. Numerous error correction techniques for tape systems have been proposed [22-23]. The effectiveness of these methods is limited only by the complexity of their implementation and the reduction in information throughput that may be tolerated as the increasing number of redundant data bits displace the information bits. The optimum error correction scheme must be judged in the light of the application.

A general treatment of error correction is given in chapter 3. Under this heading a simple, moving-parity error correction method will be considered. This has been implemented on the system described in section 4.2. The essence of the scheme is simplicity of implementation. It employs low redundancy to correct single-track errors.

4.6.1. Moving Parity Error Correction.

The error correction technique adopted follows the "moving parity" method of error detection proposed by Darwood [4-5]. Patel and
Blaum have also reported on a similar technique [24-25] which they have adapted to error correction.

The method used is illustrated in figure 4.40. A single track of the tape is dedicated to recording the parity value of data covered by two "parity strips", A and B, located a distance $L$ apart. The data covered by the parity strips lie at an angle across the tape. For the $m$ th "A" parity check the parity value is given by the equation,

$$\sum_{N=0} \sum_{N=n} A_{m-N}(N) = 0$$

where $N =$ track number and $\oplus$ signifies modulo two summation.

Similarly the "B" parity check is given by,

$$\sum_{N=0} \sum_{N=n} B_{m+L+N}(N) = 0$$

The parity value covered by strips A and B is, therefore,

$$\sum_{N=0} \sum_{N=n} A_{m-N}(N) \oplus \sum_{N=0} \sum_{N=n} B_{m+L+N}(N) = 0 \ldots \ldots \ldots (4.16)$$

Consider a single error on track N (X in figure 4.40). This will cause equation (4.16) to fail twice as it crosses the parity strips A and B. These failures occur $P$ bits apart where,

$$P = (m + L + N) - (m - N) = L + 2N$$

The replay software counts the number of bits between parity failures and is able to determine, and correct, the track which is
Fig. 4.40. Moving parity error correction.

\[
m^\text{th} \ A \text{ cross parity} = \sum_{N=0}^{N=n} A_{m-N} (N)
\]

\[
m^\text{th} \ B \text{ cross parity} = \sum_{N=0}^{N=n} B_{m+l+N} (N)
\]

\[
\text{total parity} = \left[ \sum_{N=0}^{N=n} A_{m-N} (N) \oplus \sum_{N=0}^{N=n} B_{m+l+N} (N) \right] = 0
\]

Fig. 4.41. Moving parity error correction.
\[ N = \frac{p - \ell}{2} \]

In the case of a burst error of length \( b < \ell \), equation (4.16) will fail \( b \) times as the burst crosses strip A and a further \( b \) times as it crosses strip B. The length of burst may, therefore, be determined and corrected.

Single track errors of any burst-length combination may be corrected provided the overall error length is \( < \ell \). The value of \( \ell \) may be chosen to accommodate the worst-case length of error combination.
4.7. REFERENCES FOR CHAPTER 4.


5.1. INTRODUCTION

Included in this chapter are results pertaining to the mechanical transport characteristics of the compact-cassette tape-deck and tapes. These were measured using the recording system of figure 4.1, enhanced by circuitry and/or software to enable the specific measurement to be made.

The system performance of a number of recording configurations was also assessed. These assessments invariably utilised the computing power of the microcomputer in figure 4.2, operated in the dual role of system controller and data logger. Error distributions in the X - Y axis of the tape were measured and a number of recording codes were compared. Also included in this chapter are results of the moving parity error-correction scheme used in both the 4-track and 8-track modes.

A number of proprietary compact-cassette tapes were used in the investigation. They are:

<table>
<thead>
<tr>
<th>Tape</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sony metallic, C60.</td>
</tr>
<tr>
<td>2</td>
<td>Memorex HB II, C60.</td>
</tr>
<tr>
<td>3</td>
<td>Scotch AVX 90, C90.</td>
</tr>
<tr>
<td>4</td>
<td>Memorex dB series, C60.</td>
</tr>
<tr>
<td>5</td>
<td>Memorex dB series, C90.</td>
</tr>
<tr>
<td>6</td>
<td>Memorex dB series, C120.</td>
</tr>
<tr>
<td>7</td>
<td>Philips ultra ferro, C60.</td>
</tr>
</tbody>
</table>
In the results which follow tapes are referred to either by name or number.

The chapter opens with the presentation of results on the response of the 4 and 8-track inductive heads used in the investigation.

5.2. TAPE - HEAD RESPONSE.

The frequency response of both inductive heads was measured using the arrangement shown in figure 5.1. The input to the write amplifiers was commoned and n tracks were simultaneously recorded with squarewave signals across a wide frequency range.
Following tape rewind, the track signals were monitored at the outputs of the read-head amplifiers. The results for three tape coatings; $\gamma$-$\text{Fe}_2\text{O}_3$, CrO$_2$ and metallic particles, is shown in figures 5.2 and 5.3 for the 4-track head and the 8-track head respectively. The write current used in the above recordings was set at 1.06 mA for the 4-track head and 2.6 mA for the 8-track head. These were determined to be the optimum values of recording current for each head and for all coercivities of tape used. Whilst the recording current must be

![Graph](image_url)

**Fig. 5.2.** Reproduce voltage against frequency for three tape coatings, (4-track head). Recording current: 1.06 mA.
Fig. 5.3. Reproduce voltage against frequency for three tape coatings, (8-track head). Recording current: 2.6 ma.

high enough to exceed the coercivity of each tape coating a low value is desirable to minimise peak shift on replay [1]. A number of recordings were made using the 8-track head. A fixed pattern of FF. Hex. was encoded onto each track in Miller code at 10k bits/sec per track. During recording the write current was varied, figures 5.4 and 5.5. In each case the level of crosstalk was measured by recording on all but one track.
Fig. 5A. Amplified output of read head against write current.

Memorex C60 tape, FFH at 10 kb/sec. Miller code. Track 5 blank.

3M AVX 90 tape, FFH at 10 kb/sec. Miller code. Track 5 blank.
Fig. 5.5. SNR of read-head output against write current.
To complement the above measurements on crosstalk a blank tape was recorded on one of eight tracks for several minutes, all tracks receiving a period of recording relative to the remaining blank tracks. The effect of the crosstalk is illustrated in figure 5.6. The worst-case crosstalk occurs between tracks 1 and 2 where the pick up from track 2 is 14 dB down on the track 1 signal amplitude and only 6 dB below the level recorded on

---

**Fig. 5.6. Illustrating the extent of crosstalk in the 8-track head.**
track 2 itself. Track 1 pick up is increased further by the crosstalk between tracks 1 and 3. The inner tracks are subject to pick up from adjacent tracks lying on either side. The total pick up on track 5 is only 7dB below the level recorded on this track.

The effect of track crosstalk is to shift the peak of the signal on the affected track thus impairing detection. The extent of the impairment depends on the relative code patterns on the tracks concerned.

5.3. TAPE VELOCITY VARIATION

The specified velocity of the tape of a compact-cassette recorder is 4.75cm/sec ± up to 2%. This, however, is average velocity. Of importance in the recording of digital data is the velocity characteristic over a distance occupied by one data bit—typically 5 μm. The velocity measuring apparatus described in section 4.3.6. measures relative record / playback tape velocity over tape distances of some 25 μm.

A velocity profile over these short distances was measured and the worst-case variation determined. Figure 5.7. shows the typical velocity variation experienced. A variation of ±10% is common, with occasional variations up to ±40%. These variations inevitably add to the jitter and reduce the timing window during which the data are sampled.

5.4. TAPE - AZIMUTH VARIATION

To capitalise on the high areal packing density offered by tape a multitrack format must be adopted. This introduces an
additional dimension of detection orthogonal to the tape direction. The choice of detection process now depends not only on the linear velocity of the tape but also on the azimuth angle formed as the tape skews across the tape head.

5.4.1. Measurement of Tape Azimuth.
A series of readings were taken of tape-azimuth variation against time, using the technique and apparatus detailed in section 4.3.10. In the measurement procedure a number of tapes were bulk erased and recorded successively with the appropriate control pulses. After recording, each tape was rewound using the
fast rewind facility of the tape transport. During readback the real-time azimuth variation of the tape was displayed on the microcomputer's VDU against an internally generated timebase. On demand the displayed data could be stored as a disk-based file for subsequent hard-copy reproduction.

The results of tape azimuth variation are shown in figures 5.8 to 5.15. No adjustment of head azimuth was made between recording and replay. The diagrams illustrate various characteristics of tape azimuth behaviour over long and short time periods, the timebase being a function of the software. The large offset from the zero axis, observed in some of the results, is static skew. This is caused by non-ideal tape spooling during the fast rewinding of the tape and is tape dependent. Also, the magnitude of the dynamic skew variation is a function of the tape type. Whilst visual analysis of these results give an indication of static and dynamic tape-skew variation, further analysis is required to reveal the periodicity of the signal.

5.4.2. Spectral Analysis of Tape Azimuth.

The spectral content of the dynamic azimuth signal was measured by outputting the digitised azimuth value from the microprocessor to a spectrum analyser via a D/A C. These results are shown in figures 5.16. to 5.21. for a number of tapes.

5.5. TAPE DEFORMATION

The tape-azimuth results were measured using information encoded on the outer tracks of the tape. This assumes that the azimuth is constant across the tape width. This was found not always to be the case.
Fig. 5.8. Bit-cell displacement of three tapes against time.

Fig. 5.9. Illustrating the effect of ideal tape spooling on bit-cell displacement.
Fig. 5.10. Illustrating the effect of tape spooling on bit-cell displacement.

Fig. 5.11. Dynamic skew of one tape for three successive "rewind and play" cycles.
Fig. 5.12. Tape skew showing large variation in dynamic skew.

Fig. 5.13. Dynamic skew against time.
Fig. 5.14. Variation in static skew with time.

Fig. 5.15. Skew of one tape for two successive "rewind and play" cycles.
Fig. 5.16. Spectral components of tape-skew variation.

Fig. 5.17. Spectral components of tape-skew variation.
Fig. 5.18. Spectral components of tape-skew variation.

Fig. 5.19. Spectral components of tape-skew variation.
Fig. 5.20 Spectral components of tape-skew variation.

Fig. 5.21 Spectral components of tape-skew variation.
A series of measurements were made which effectively monitored and displayed the phase difference between track signals which were recorded "in-phase". The microcomputer used in the above process also calculated and logged the deformation of the tape as defined in section 4.3.13. A running value of the average of successive tape deformation readings was taken and displayed by the microcomputer. Tapes of three different lengths (and, therefore, thickness) were used: C60, C90 and C120.

The measurements revealed (for the particular tape transport/head combination used) two predominant skew pattern characteristics, figures 5.22. and 5.23. In figure 5.22. the skew angle is evenly distributed across the width of the tape — this allowing for about 5% error in the system due to differences between the eight read amplifiers. In the second skew characteristic (figure 5.23.) the centre tracks of the tape reach their corresponding read gaps before the outer tracks suggesting a deformation of the tape along its centre. The deformation of the tape may vary in magnitude as the "polarity" of the skew angle changes. Figure 5.24 shows the tape deformation when this occurs.

Similar results were obtained with tapes of different lengths, the magnitude of the tape deformation did, however, vary with tape thickness.

The microcomputer was used to read and calculate average tape deformation \( D \), for C60, C90 and C120 tapes. More than 800 samples per tape were taken during a total playing time of seven hours.
Fig. 5.22. 'Linear' distribution of skew angle across the tape.

Fig. 5.23. Tape deformation display format.
Fig. 5.24: "Non-linear" distribution of skew angle across the tape showing reduced tape deformation.

This gave an average tape deformation of 0.67 μm, 0.85 μm and 1.08 μm for C60, C90 and C120 tape respectively. These are plotted in figure 5.25.

Fig. 5.25: Tape deformation against tape thickness.
5.6. ERROR CHARACTERISTICS OF RECORDING SYSTEM 1.

The compact-cassette recording system of figure 4.1. was characterised in terms of its error rate and modes of failure. The eight-track head was used to obtain error burst length information and details relating to the spatial distribution of errors across the tracks. Also, the effectiveness of the moving-parity error correction scheme was assessed.

5.6.1. Error Burst Length.

Using the 8-track head, tapes were encoded with a fixed, repetitive pattern of data in Biphase-L code. On playback the eight tracks were sampled simultaneously by the replay software and decoded. Data synchronisation was achieved through computation of transition interval ratios. As data were detected each reading was compared to the known value and for each successive mismatch the address of an error-logging memory space was incremented. When the data are read successfully the contents of the memory thus addressed is incremented and the memory address reset. On completion of the measuring period the error-logging memory contents are displayed as a histogram of error-burst length.

Results typical of those measured are given in figure 5.26. The general trend is for the frequency of error-burst lengths to decrease with error-burst length.

5.6.2. Spatial Distribution of Errors.

A technique similar to that described above was used to determine track failure mode. As a pre-coded tape was read
Fig.5.26. Typical distribution of errors over 25 hours at 25 k.bits/sec/track.

back the replay software categorised the occurring errors in terms of single, double, treble, etc., track failures. The tracks failing were also logged and, as before, a histogram of failure modes was displayed at the termination of a measuring period. Data were encoded at 1 kbit/s per track using Biphasic - L code. Figure 5.27 shows the results from a typical recording of 20 minutes of a C60 tape with the average of 10 x 20 minute passes illustrated in figure 5.28. Other C60 and C90 tapes show a similar distribution of track errors.
**Fig. 5.27.** Typical distribution of track errors over 20 mins. at 1 k.bits/sec/track (8-track head).

**Fig. 5.28.** Relative distribution of track errors averaged over 3.3 hours. 1 k.bits/sec/track (8-track head).
5.6.3. Moving Parity Error Correction Results.
The moving parity error-correction scheme of section 3.3 was implemented. The error-correcting software overhead precluded error correction in real time at high data rates. However, the software used to detect and log burst errors was modified to readback and store 208, 8-bit samples. These were sandwiched between 2 x 24 memory bytes containing error-free data to form a data buffer. After a buffer is filled the software "scans" it with the two parity strips and corrects errors. Data verification then takes place with errors being logged and categorised as in the error-burst program.

A direct comparison was made between the error rate logged with and without the application of error correction. Due allowance was made for the difference in throughput between the on-line and "off-line" methods used.

Results showing the errors for an average of 30 minutes playing time is shown in figure 5.29. Average results are shown as measured over a four-hour period. The recording rate is 1 k.bits/sec/track and the same, C60 tape, was used throughout. Results obtained over a single 30 minute playing time, with the same C60 tape, are shown in figure 5.30. The simple, moving-parity error-correction scheme clearly reduces the raw error rate. However, since it is a single-track error correction system, dual-track errors adversely affect its performance. This aspect is considered in chapter 6.
Fig. 5.29. Comparison of errors with and without moving parity error correction. 1k bits/sec/track (8-track head).

Fig. 5.30. Comparison of errors with and without moving parity error correction. 1k bits/sec/track (8-track head).
The error - burst measurement technique, described in section 5.6.1., was used to assess the error performance of the second recording system (section 4.4.). In this case the 4-track, record / playback head was used and a 4-bit pseudo-random binary sequence (PRBS) was recorded in Biphase - L code as the test pattern. Two methods of data detection were employed: data were sampled using a fixed track as reference and also an adaptive detection technique was employed (section 4.4.7.). In both cases the software synchronised to the data by continuously measuring and computing the ratio between transition intervals. Recordings were made at data rates ranging from 500 bits/sec/track to 2.5 k.bits/sec/track. To obtain error rates which represent system performance accurately, recordings were repeatedly played over a five hour period. Figure 5.31 shows the typical accumulation of errors with data rate - as the latter increases the error burst length increases, as does the frequency of occurrence. The repeatability of the error measurements was assessed by employing tape 1, the tape with excellent skew characteristics (figure 5.11), encoded at 2.5 k.bits/sec/track. The results are shown in figure 5.32. With errors due to tape skew virtually eliminated, those errors which do occur are almost certainly due to tape dropouts.

The effect of employing two tape guides on the same head was measured, figure 5.33. An additional guide was fixed on the opposite side of the head to the standard guide and with this configuration error readings were taken over a four hour period.
Fig. 5.31. Error burst length over 5 hours for data rates 0.5 to 2.5 k bits/sec/track (4 track head).
Fig. 5.32. Error burst length for ten × fifteen minute passes of tape. 4-track head, 25 kbits/sec/track (tape1).

<table>
<thead>
<tr>
<th>Burst 1st pass</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 5.33. Error burst length over 4 hours at 15 kbits/sec/track with two tape guides and one tape guide (4-track head).
When compared with a similar period of measurement with only one tape guide, the errors incurred were greater.

The difference between using a fixed track as a synchronising reference and the adaptive method of synchronising was measured.

A number of C60 tapes were encoded at data rates 500 bits / sec per track to 2.5 k. bits / sec per track. The replay procedure was as follows. The same tape was replayed a number of times, typically 20, for odd-numbered passes, error-burst length was measured with the software using a fixed track as reference. For even-numbered passes the adaptive software was used.

The results are shown in figure 5.34. Overall error rate is given

![Diagram](image)

**Fig.5.34.** Error rate performance for fixed and variable synchronisation.
against data rate for both detection methods, the burst errors are treated as \( n \) - single bit errors. At the lower data rates isolated errors are due to blemishes in the tape coating and are reproducible at the same point as the tape passes the read head.

The performance of Miller code and the 2/3 rate \( 0.2 \) code, discussed in section 4.4.11, were compared with Biphase - L code. The results of all three are shown in figure 5.35. In each case

![Graph showing data rate against error rate for Miller code and 2/3 rate, 0.2 code and Biphase-L code.](image-url)

Fig 5.35 Data rate against error rate for Miller code and 2/3 rate, 0.2 code and Biphase-L code.
the successive interval ratio method of synchronisation was used. In order to effect a straight comparison the fixed-track detection method was used. The 2/3 rate 0.2 code was decoded by sampling once every data-bit cell. To decode Miller code two samples were taken every bit cell with the decoded data being taken as the exclusive OR function of the two samples.

5.8. SKEW CORRECTION RESULTS.

The data recording rate may be increased further than that achieved above if, in the time consuming detection process, the burden on the software is eased. This is possible through the use of hardware to assume part of the detection process.

The recording system 1 (section 4.2.1.) was enhanced by the addition of a hardware clock for each track and skew correction circuitry. Software was developed to encode and decode data at 10 k.bits/sec per track in ISS 2/3 code. Details of the software and the operation of the skew-correction circuitry are given in section 4.5.

The two outside tracks of the tape were encoded with skew-correction control pulses. These comprise a periodic square waveform simultaneously recorded on the tracks at a frequency of 10 / 4 kHz. These signals were used to identify each 3-bit code group of the ISS 2/3 code and control the operation of the skew-correction circuitry. The encoding software simultaneously recorded the 6 inner tracks with data. These data were the digital values of a "sawtooth" waveform. On replay the decoded data were outputted to a digital-to-analogue
converter (DAC) the output of which was used to assess the error rate.

In developing the ISS 2/3 decoding software a second microcomputer was utilised. This simulated the eight tracks of an encoded tape by generating a "sawtooth" waveform in ISS 2/3 code at the required frequency. The results of using the skew-correction circuitry and the decoding software to decode the computer-generated signals is shown in figure 5.36. The decoded data are outputted by the microcomputer to a DAC. The 64 steps of the sawtooth are clearly seen with no tape dropout or skew-generated errors to mar the reproduction.

Fig. 5.36. Decoded "sawtooth" waveform. Data encoded and generated by a second microprocessor: ISS2/3 code, 10k bits/sec/"track" (6-data"tracks"+2 deskew"tracks").
The results with data read from an encoded tape are shown in figure 5.37. These results are discussed in chapter 6.

The skew - correction pulse generator periodically synchronises the clock switch to ensure that the sequence of the 6 track clocks is in accord with the prevailing tape skew. If these skew - correction pulses are removed synchronisation is lost within a few seconds as a glitch / dropout disrupts the correct clock sequence.

Fig.5.37."Sawtooth" waveform showing natural errors.10k.bts/sec/track(8-track head).

5.9. 18 - TRACK MR HEAD.

The software developed during this investigation has been for application to a multitrack format. Both in the encoding and decoding process the microcomputer programs operate with each tape track being processed simultaneously : one bit of the microprocessor word to each track. When the software of the 8 - bit processor is applied to a 4 - track system the processor is under utilised, also, the masking out of the unused 4 bits
increases the software overhead.

All the software may be transcribed to a 16-bit format without difficulty and it was for use with a 16-bit processor that the 18-track MR head was developed. The two outside tracks would be encoded with skew-correction pulses with the 16 inner tracks carrying the data. Difficulties have been encountered in the development of a practical 18-track, inductive write, MR read head and to date no results involving this device have been generated.
5.10. REFERENCE FOR CHAPTER 5.

CHAPTER 6

DISCUSSION OF RESULTS & CONCLUSIONS.

6.1. INTRODUCTION.

The results detailed in chapter 5 are reviewed and discussed in this section. These results were taken using both the four and eight - track heads. The tape employed ranged from low - coercivity \( \gamma - \text{Fe}_2\text{O}_3 \) tape (250 - 350 oersted) to the high coercivity metallic tape (800 - 1500 Oe).

A bandwidth of 10 kHz is possible with a combination of 4-track head and metallic tape. This gives a theoretical Nyquist signalling rate of 20k bits/sec/track with about half this value for the other tape coatings. The value of recording current adopted was 1.06mA and 2.6mA for the four and eight - track heads respectively.

6.2. Head Crosstalk.

As evidenced by figures 5.4 and 5.5 the performance of the eight - track head is far from ideal. This head suffers from a high degree of crosstalk, figure 5.6. The worst - case crosstalk is 18dB with 15dB being typical, compared to a worst - case cross - talk value of 40dB for the four - track head. Crosstalk has a significant effect on the peak detection process as the zero - cross over point of the reproduced signal is shifted. Van Gestel, et al [1] have identified two components of crosstalk: crosstalk during playback and crosstalk during writing. Playback crosstalk can be reduced by increasing the guardband between tracks. Also, since it is a linear process, the playback signal
from each track may be compensated by subtracting a proportion of the signal from adjacent tracks amounting to the crosstalk incurred by the read head. Crosstalk during writing is a non-linear process and arises as the record current from one write coil acts as a bias field for the crosstalk from adjacent tracks. This may be reduced by cycling the write currents in each write coil of the headstack so that adjacent write coils never carry current at the same time, figure 6.1. The penalty incurred by high levels of crosstalk is an increase in bit error rate. One method of reducing the effect of crosstalk is to adopt a slant azimuth recording format. Adjacent tracks are recorded with different azimuth angles. The effect of this on replay is for each read gap in the headstack to filter out signals from adjacent tracks. An azimuth angle of 15 degrees has been proposed at which crosstalk is reduced to an acceptable level[2].

![Diagram of bit cells and write currents](image)

**Fig.6.1.** Track crosstalk is reduced by ensuring that no two adjacent write coils carry current at the same time.
6.3. TAPE TRANSPORT CHARACTERISTICS.

A low-cost tape transport arrangement such as the compact cassette system is not suited to recording digital data at high densities in a multitrack format. The mechanical vagaries of the transportation system, such as tape velocity and azimuth variation, impose a limitation on the recording rate if not compensated for.

6.3.1. Tape Velocity.

As stated in section 5.3, the tape-velocity variations over distances comparable to those occupied by one data bit are important. The velocity variation shown in figure 5.7. has been transcribed from a recording made on a UV chart recorder with the apparatus of figure 4.11. Although points are plotted at 0.25 sec. intervals the time resolution of the measuring system is 25 μm. This is considered adequate to detect velocity variations over 5 μm, i.e. one bit cell at 10 kbits/sec.

As seen from the sample graph (figure 5.7.) variations of ±5% of one 10 kbits/sec bit cell are typical. Occasionally transient changes of up to ±40% also occur. Variations as large as this must be accommodated by the replay process. A common technique is to use a phase-locked loop (PLL) synchronised by a self-clocking code to regenerate a clock for each track [3]. The time constant of a PLL is, however, long and transient changes in tape velocity are missed. A further shortcoming of adopting this approach is the expense of providing one PLL circuit for each tape track.
The software clock regeneration devised for the compact-cassette transport is tape-velocity independent. The method also synchronises to the replayed data to discriminate between bit-cell centres and bit-cell boundaries. This latter function is continuous throughout the replay time of the tape and depends on the frequency of occurrence of certain data sequences. With Biphase-L code these data sequences are 001, 100, 110 or 011 which occur frequently in random data. The method is applicable to any run-length limited code provided a code sequence yields a unique pair of transition intervals.

To eliminate mis-synchronisation due to extraneous signal transitions the sampling software compares each measured bit-cell interval with that used by the microcomputer to calculate the sample point. If there is a large discrepancy the new interval is rejected and the stored bit-cell interval is not updated. The software thus mimics the action of a phase-locked loop (PLL) with the software limits of comparison equivalent to the PLL time constant. This action ensures that the data continue to be sampled at regular intervals in spite of possible dropouts and additional signal transitions. This latter error mechanism forces additional bit-cells into the data stream which would inhibit potential error correction.

6.3.2. Tape Azimuth.

The effects of tape azimuth variation are to reduce the amplitude of the reproduced signal and to impair the coherent detection of bits recorded simultaneously across the tape. It is convenient to measure tape azimuth variation in terms of bit
cell displacement across the tape. A 100μs bit cell was chosen. This is the time duration of one bit at 10 kbits/sec and corresponds to a distance on tape of 4.75 μm. In terms of azimuth angle this amounts to 4.4 minutes of arc. Computer - tape systems experience skew - angle variation of magnitude 2 to 3 minutes of arc [4]. In such systems tape - head azimuth alignment to an accuracy of 12 seconds of arc is normal [5].

The accuracy of the tape - azimuth measurements was verified by the results obtained in manually introducing a known value of skew and comparing this with the measured value, (section 4.4.12). Both dynamic skew and static skew was observed in each of the tapes tested. Dynamic skew variation varied between tapes from an excellent ±20% of one bit cell to values in excess of 100% of a bit cell length.

Typical results are shown in figure 5.8. Here the skew - angle variations of three C60 tapes from different manufacturers is compared. Each shows a total skew - angle variation within one bit cell at 10 kbits/sec. A common feature of the results is the cyclic variation of skew - angle as the tape is played. This is common to all the tapes tested.

Static skew manifests itself in two distinct forms:
(a) An offset of the skew characteristic at time \( t = 0 \).
(b) Static skew drift.

Skew offset occurs after the tape has been rewound. The drift in static skew occurs over long playing times of the tape. Consequently the static skew of a tape could vary substantially between successive playing periods with rewind in between.
Figure 5.9. illustrates the tape azimuth variation of one tape which (at the time of measurement) spooled ideally. This contrasts with the results of a second tape which shows an offset from zero, figure 5.10. The offset is due to the lateral movement of the tape hub as the tape is rewound (spooled) figure 6.2. The extent of this movement depends on the cassette construction. Some manufacturers insert plastic shims to restrain this movement whilst in some cassettes the casing fulfills this role. To some extent this lateral movement is buffered from the tape head by the two pulleys which separate the tape head from the two reels. In spite of this the lateral movement of the tape hubs manifests itself as static skew. The distance between the two pulleys is 88mm. With a tape skew of one 10 kbits/sec. bit cell the skew-angle is:

\[
\text{arctan } 4.75 \times 10^{-6}/3.7 \times 10^{-3} = 0.0735 \text{ degrees.}
\]

The lateral hub displacement due to this skew-angle is:

\[
88 \times 10^{-3}/2 \tan 0.0735 \text{ metres} = 0.056 \text{ mm.}
\]

Fig.6.2. Lateral movement of tape during spooling causes static skew.
Within the above constraints of lateral movement the static skew can change as the tape is played. This effect is seen in figure 5.14 where the static skew gradually changes before stabilising at a steady value after 10 minutes.

The skew characteristics of a tape are unrelated to its "quality" as measured by cost. The excellent dynamic and reproducible static skew exhibited by tape 1 contrasts with that of tape 13. The first tape is housed in a plastic cassette with a cost of one quarter of tape 13 — the cassette of this tape is formed from a heavy metal frame with plastic sides. Both are metallic tape. Apart from this expensive item the remaining tapes had plastic cassettes and were modestly priced. In view of the similarity in cassette construction and guidance system the skew behaviour of the cassettes is primarily due to the uneveness of the tape width. Apart from an occasional large transient change in skew the maximum variation was invariably within plus or minus one 10 k.bit/sec.bit cell.

6.3.3. Spectra of Tape Azimuth Variation.

The graphical display of tape azimuth variation was down loaded from the microcomputer to a spectrum analyser. Sample results from four tapes are shown in figures 5.16 to 5.21. The relative amplitude of each component varies as the tape is running. Apart from the 2 Hz component, which was common to all tapes, the frequency of the spectra varied from tape to tape. The 2 Hz component was attributed to the tape transport — in particular the pinch roller. At compact-cassette tape speed 2 Hz translates into a length which approximates the circumference of this...
component. The other harmonic components are due either to the tape or the cassette tape guides.

The final process in the production of tape is the slitting of the magnetically coated rolls of base film into tape. This is done using either fixed or rotary cutters which invariably impart a curvature to the edges of the tape. This causes the tape to weave through the tape guides as it is played thus giving rise to tape skew. Jorgensen [4] quotes a figure of tape-edge curvature for computer tape of up to 1/8 inch over 36 inches. It is likely, therefore, that the low frequency components of the tape azimuth spectra are due to this parameter whilst the higher frequency components are due to the cassette tape guides. Taking the 0.32 Hz spectral component, figure 5.20., the "period" of curvature on the tape would be:

\[
4.75 / 0.32 = 14.8 \text{ cm.}
\]

6.3.4. Tape Deformation.

In determining the incremental skew characteristic across the full width of the tape it was necessary to measure the skew angle distribution away from the edges of the tape. This had been assumed to be linear by interpolating between the measured values of tape azimuth on the outer tracks.

Detailed investigations of the skew angle on the inner tracks of the tape revealed two predominant modes of operation. As predicted by the dynamic tape-azimuth measurements, the tape skew between the outer tracks of the tape changes polarity cyclically. In the first mode of operation, figure 5.22.
the skew angle is distributed across the full width of the tape in an orderly fashion. Some scattering of the track readings about the mean value does occur but, allowing for the inherent error in the measuring system, the skew angle distribution is virtually linear.

In the second mode of operation, figure 5.23, the data on the centre tracks reach their respective read-head gaps before those of the outer tracks. This results in a deformation of the tape along its centre and the profile of the tape—skew angle across the tape resembles the bow wave of a ship. The likely cause of this "bow-wave" effect is the friction, offered by the tape guide, to the outside edges of the tape. This works contrary to the action of the pinch roller operating along the tape centre. The deformation characteristic of figure 5.23 illustrates tape operation with a low value of tape skew, the pinch roller and tape-edge-guide forces forming a balanced couple on the tape. In figures 5.22 and 5.24, the balance of the couple on the tape is tipped one way or the other depending on the prevailing tape skew.

The tape deformation was quantified by modifying the display software to compute D, as defined in figures 5.23 and 5.24, and average its value. This was done for C60, C90 and C120 tapes of the same type. The results are shown in figure 5.25 which clearly shows the thinner tapes C120 and C90 being stretched further than the C60 tape which has twice the thickness of the C120 tape. In terms of bit-cell time the effect of tape deformation is shown in figure 6.3. It can be seen that the
average time displacement of the tracks for C120 tape amounts to almost 25% of one bit cell at 10k bits/sec/track compared to 18% and 14% for C90 and C60 tape respectively. These values amount to a corresponding reduction in detection window when data, recorded broadside across the width of these tapes, are sampled simultaneously. Also, the above figures represent average values only. Tape-deformation values in excess of one 10k.bit/sec.bit - cell do occur, paticularly with the thinner tapes. In these cases data recorded at this rate would be mis-detected if simultaneous sampling of each track is used.
6.4. ERROR PERFORMANCE OF RECORDING SYSTEMS.

Results relating to three recording systems have been obtained.

(a) System 1, figure 4.1, in which the 8-track head was employed.

(b) System 2, with 4-track head, section 4.4.

(c) System 3, which is system 1 enhanced by additional circuitry.

A simple, moving parity, error-correction scheme was also applied to system 1 and its effectiveness measured.


Errors incurred in this system are predominantly single-track errors. This mode of failure is typical of multitrack tape systems. Double and triple-track errors also occur but to a much lesser extent. Little can be concluded about the distribution of errors between tracks although it is noted that track 1 of the 8-track head suffers a high level of crosstalk.

The number of errors which do occur during tape replay is a function of recorded data rate. This also applies to the burst-error length distribution. A typical distribution of errors, measured over a long period, is shown in figure 5.26. Here, the burst length is defined as the number of successive error readings between two good readings. The frequency of occurrence of error bursts is seen to fall off with error burst length. The relationship between error frequency and burst length can be seen in figure 6.4. In this graph the inverse of the normalised error frequency is compared with the inverse of the burst length, b, and the inverse exponential of the burst length, both normal-
Fig. 6.4. Normalised occurrence of burst error, $b$, compared with $e^{-b}$ and $kb^{-1}$.

For this example the fall off in burst-length frequency is a closer fit to the $Ke^{-b}$ curve than the $kb^{-1}$ curve. This is in agreement with Meek's results[6]. Error bursts are usually caused by loose tape-coating debris or dust particles between the head and the tape, tape blemishes are also a cause. Because of the finite size of these error-causing mechanisms, error-burst length is inversely proportional to bit-cell length and, therefore, for a given code, data rate.
Notwithstanding the problems of crosstalk associated with the 8-track head, the moving-parity error correction scheme was applied. The results, figures 5.29 and 5.30, are, predictably, poor with a reduction in errors to only 25% below the raw error rate, approximately.

Although the overall error rate was reduced, certain error patterns show an increase. In figure 5.29 all single and double-track errors are reduced but some double-track errors are caused. The same applies to figure 5.30 with the exception of a single-track error increase on track 3. As is common with any error-correction code, errors increase because of mis-correction of error patterns the code was not designed to contend with: so-called pathological error patterns. In this case the code was designed to handle single-track error bursts which occur at least L bits apart. Error patterns not falling within these restrictions may be mis-corrected.

The diagrams 6.5. and 6.6. illustrate the mis-correction which takes place due to double-track errors and certain single-track failure patterns to which the system is insensitive. The error-correcting software must detect the same sequence of parity failures due to errors as they pass both strip A and strip B. This condition is satisfied by any double-track error, however, the correction system assumes two errors on one track and mis-corrects, figure 6.5. The existing double-track error is left uncorrected. The above software, error-correction criterion is also satisfied by the single-track failure pattern shown in figure 6.6. Here, the two errors which lie in-line with the
Fig. 6.5. Miscorrection of data due to double-track error.

Fig. 6.6. Double-track error due to miscorrection of single-track error pattern.
parity strips do not produce a parity failure. The one error which does cause parity failure in strips A and B is different. Consequently, the error-correcting software mis-computes the track and location in error and adds to the list of double-track errors. A similar error pattern of the form $X \times X$ will also produce a similar result.

Notwithstanding its shortcomings the error-correction scheme is simple and for an 8-track system, the redundancy is only 12.5%. The error-correction performance could be improved by interleaving the data to remove the adverse effects of error clustering.

6.4.2. Compact - Cassette Recording System 2.

The results of recording data in Biphase-L, Miller and 2/3 rate 0,2 code with the four-track head is summarised in figure 5.35. The finite instruction time of the decoding software, for Biphase-L and Miller code, limits the recording rate to 2.5k.bits/sec/track. The 2/3 rate 0,2 code is a development of the 1/2 rate Biphase-L code and operates 1.33 times faster thus giving an encoding rate of 3.33k.bits/sec/track for the same computer execution time.

At the low data rates the three codes yield the same error rate ($10^{-7}$). The bit-cell time, at 500 bits/sec/track, is 2000μS and well within the execution time of the program. At these long wavelengths the skew errors are minimal. The errors which do occur are due to tape-coating blemishes and are reproducible at the same points on the tape each time it is played.
As data rate increases, additional errors, due to tape skew occur. Although Miller code operates at half the frequency of Biphase - L, because the data tracks are parallel and sampled simultaneously, the effective detection window is the same, at T/2, where T is the duration of a bit cell. The detection window of the 2/3 rate 0,2 code is also T/2 but since three data bits are encoded for every two Biphase - L data bits the error rate plot is virtually that of Biphase - L displaced to the right. The higher error rate of Miller code is due to the decoding process which requires that each data bit be sampled twice compared to once for the other two codes.

The maximum data rate/error rate ratio is,

\[
\frac{2000}{10^{-7}} = 2 \times 10^{10}
\]

at 500 bits/sec/track. This is about five times lower than that achieved by Weiller (section 1.43). However, at a total data rate of 12k.bits/sec. the figure of merit is the same for both systems. Unlike Weiller's system, pulse slimming is not employed. The incorporation of this process into the simple recording system would give a potential for increase in performance of some 20\%, albeit at an increase in complexity.

The 2/3 rate 0,2 code gives an improved performance when compared to Miller and Biphase - L code. Further improvement is theoretically possible by increasing the code rate to 3/4, 4/5, 5/6 and so on. As the code rate increases the maximum distance between transitions, k, also increases and although the code efficiency increases, the higher k/d ratio makes signal detection
more difficult and code synchronisation would be adversely affected. The digital sum variation (DSV) would also increase.

The spectral content of the codes used in the investigation are compared in figure 6.7. The spectra are of a 255 long PRBS at 1k.bits/sec. As expected the spectra for Biphase-L and 2/3 rate 0,2 codes are similar with the latter having a slight d.c. component. Miller and ISS 2/3 code also have a d.c. component but notably the frequency components of ISS code are lower than Miller. Because the energy of both spectrums is concentrated at lower frequencies the performance of these codes would be enhanced by channel equalisation.

6.4.3. Errors Due To Skew.

At the higher data rates the effects of tape skew make a significant contribution to the error rate. Also, at these higher rates, the execution time of the decoding software becomes significant. This delays the precise point at which data are sampled thus cutting down on the detection window.

Errors due to tape skew are shown in figure 6.8, this diagram also shows the effect of adaptive track synchronisation. At the low data rates the effect of tape skew is negligible and the error rate for single and multitrack sampling is the same. At the higher data rates the two plots diverge as multitrack sampling is subject to additional errors due to tape skew. The single-track error rate also increases as the fixed size of the tape-coating blemishes cover more data bit wavelengths plus additional errors due to program-time limitations. Tape-skew
Fig. 6.7. Comparison of spectral components of the codes used in the investigation.
Fig. 6.8. Errors due to tape skew (crosshatched area) is the difference between 4 times single-track errors and errors incurred when 4 tracks are sampled together.

Induced errors increase the error rate by an order of magnitude at 2.5 k.bits/sec./track.

The effect of synchronising to the leading track is to double the detection window. The result of this is seen in figure 6.8, where skew-related errors are halved.
6.5. ERROR CORRECTION.

The graph of raw bit-error rate against the theoretical (section 3.3.1) corrected bit error rate is shown in figure 6.9. Data rates are indicated at the corresponding raw bit error rates incurred.

The corrected error rate at 500 bits/sec/track is $10^{-11}$ or better than one error in one year of continuous operation. At this data rate the data rate/error rate ratio is,

$$\frac{2000}{10^{-11}} = 2 \times 10^{10}$$

an increase of $10^4$ above the uncorrected value. The results show

![Graph showing the corrected error rate against raw error rate for moving-parity error correction.](image)

**Fig.6.9.** Corrected error rate against raw error rate for moving-parity error correction.
the necessity of minimizing the raw-bit error rate in order to achieve a reasonable value of corrected error rate. A fundamental advantage to this error-correction method is the incremental nature of the codeword. No codeword identification is necessary which makes it possible for the tape to start at any point with error correction being operational after a delay of L bits.

The error correction analysis, in section 3.3, assumes single-track errors only. With multitrack sampling, however, simultaneous multitrack errors occur as tape skew and tape deformation become significant compared to bit-cell length. Also, tape blemishes spanning two, or more, tracks is a cause. These errors are not only uncorrectable but, as previously discussed, cause additional errors. For this reason moving parity error correction would not be suitable at the higher data rates unless forming part of a layered error-correction scheme. An improvement would be to assign two sets of parity strips to alternate tracks. This would correct simultaneous, adjacent-track failures plus outside track correction for a four-track system.

At low data rates simultaneous track errors are infrequent (with the 4-track head) and to date an error rate of better than $10^{-8}$ has been measured.

6.5.1. Proposed Error Correction Strategy.

At 500 bits/sec/track, moving parity error-correction yields an error rate of $1 \times 10^{-11}$. Whilst this is adequate for some data-storage purposes, the data rate is at the low end of the performance specification of recording systems 1 and 2. At the high end, 3.33 kbits/sec/track, the error rate is $1 \times 10^{-5}$. The
data rate may be increased by the application of a faster microcomputer. However, the raw error rate deteriorates by approximately one order of magnitude per octave of track data rate. At a notional data rate of 10 kbits/sec/track, the raw error rate would be approximately 1 in $10^{-4}$ with a corrected rate of 1 in $10^{-5}$. Improvement at this, and higher data rates is possible through the addition of interleaving and Reed-Solomon coding.

6.5.2. Convolutional Interleaving.
The moving parity error-correction code corrects single track errors but mis-corrects multitrack errors. However, these may be dispersed into single-track errors by convolutional interleaving. Before recording, $N$ successive tracks are delayed by $0, D, 2D, \ldots (N-1)D$. At replay the inverse is applied to de-interleave. To accommodate track error bursts of length $b$, the depth of interleaving $D$ should equal $L + b$, where $L$ is the length of the moving parity error-correction zone, figure 3.17.

Convolutional interleaving can be programmed with marginal software overhead. Furthermore, data-block identification is not required, thus the overall redundancy remains unchanged. A minor disadvantage is that data are not fully recovered until after the occurrence of $N (L + b)$ parallel-track samples each time the tape is started. The incorporation of convolutional interleaving permits the moving parity code to accommodate simultaneous multitrack errors of up to burst length $b$. Isolated errors could, however, impair performance if they occur a distance $(L + b)$ apart on different tracks. The problem of isolated track errors being de-interleaved into multitrack
errors may be addressed by the application of a cross interleaved error correction scheme.

6.5.3. Cross Interleaved Moving Parity / RS code.
Error bursts normally occur along tracks, on the x axis, and infrequently across tracks, on the y axis, as a multitrack error. Gross errors, caused by missing or loose tape coating, may also cause a combination of x - y errors. This would be the case as the data rate is increased. The power of the interleaved moving parity code to correct errors along the x axis can be combined with the y - axis - correcting power of the Reed - Solomon code to reduce any impairment caused by isolated errors.

Figure 6.10 illustrates a possible scheme. Before recording, data are encoded in moving parity code. Following interleaving, RS code is applied with the 4 - bit elements of the code recorded across the tape. Errors may occur during record or playback, or both. Immediately after playback, before de - interleaving, the RS code is applied to correct up to two, 4 - bit samples. After de - interleaving the data are passed to the moving parity error - correcting code for correction. The RS code could be the 15, 11 code over GF ( 2^4 ). Within each 15 x 4 - bit code vector the code can correct any two 4 - bit elements.

The above error correction strategy would significantly enhance the correction ability of the moving parity code. However the inclusion, "in line", of the RS code vectors generates the need for data - block markers, so further decreasing data throughput. The RS decoding procedure is also demanding of microcomputer
Fig. 6.10. Cross interleaved error-correction scheme for 4-track tape using RS 15,11 code and moving-parity code.
time and at the higher data rates a multi-processor system would be called for.

6.6. SKEW - CORRECTION.

Because of the limited response time of the microcomputer at an encoding rate of 10 k.bits/sec./track, the discrete measurement of errors was not possible. The error performance of the skew-correction circuitry was gauged, however, by observing the analogue equivalent of the digitised sawtooth waveform which was recorded.

When the computer-generated data are replaced by signals off tape the errors in the decoded sawtooth signal can be observed, figure 5.37. These are identified as large changes in the steps of the sawtooth signal.

The skew-correction circuitry and the ISS 2/3 encoding/decoding software operate faultlessly on data generated by a second microcomputer (figure 5.36.). These data are error free, and more importantly, are generated with zero skew between the tracks.

On the basis of this limited data sample the BER is extremely high, of the order of 1 in 50, however, in view of the problems associated with this 8-track head, this is not entirely unexpected. One successful aspect of the operation of the skew-correction hardware is the ability of the system to maintain synchronisation. Throughout the replay time of the encoded tapes the skew-correction pulse generator (figure 4.38.) maintained the clock switch (figure 4.37.) in sequence. Only when the skew-
correction pulses were inhibited was synchronisation lost.

The additional hardware to deskew and multiplex the data into the microcomputer was necessary to achieve the high data rate of 10 k.bits/sec./track. With a microcomputer of increased power these functions of the hardware could be assumed by software. The increased computing power could be used to synchronise to each data track independently. This approach is being pursued in a complementary investigation which uses an array of Transputers [7]. Each track of a 4-track system is processed independently to eliminate the effects of skew. An attendant problem with this technique is to reform the data from the individual tracks into a deskewed data word after sampling. This requires the inclusion of data-block identification on each track if skew in excess of one bit cell is to be accommodated. The technique is also computing-time intensive, especially for high track numbers.

The ISS 2/3 channel code is an efficient code with a larger detection window than Miller code. The spectral components of a PRBS encoded in ISS 2/3 are similar to Miller-code components, figures 6.7, but the larger amplitude components occur at a low frequency and the high frequency components die away quicker. A disadvantage is that each 3-bit code group must be identified and this would normally require a separate clock track. A possible solution to this shortcoming would be to program a code violation on one (or more) data tracks at multiples of the code group rate. This is the only synchronisation requirement since encoded signal transitions only
occur at the centre of each bit cell and thus cell-centre identification is inherent. This latter characteristic of the code doubles the tolerance on sampling time, when sampling a number of parallel tracks, compared to Biphase-L code and Miller code.

6.7. CONCLUSIONS.

The mechanical vagaries of a low-cost tape transport and tape, such as the compact-cassette format, may be offset by programmable techniques to produce a viable, multitrack, digital tape recorder. System functions, such as channel/error encoding/decoding and data synchronisation may also be implemented in software to maintain the low-cost philosophy. Parallel software algorithms can be developed which permit each data track to be processed by each bit of an n-bit computer.

The tape-transport limitations are velocity variation and tape azimuth (skew) variation aggravated by tape deformation. Tape wander may also prove a limitation at high track densities. Causes are attributed to both tape and tape-transport mechanism.

Over bit-cell distances of 25 μm velocity variations of ±15% are typical with occasional variations of up to ±40%. These variations can be ironed out by a velocity-independent detection procedure which is based on the computation of successive flux-transition interval ratios. This technique not only nullifies the effect of tape-velocity changes but can also be used to identify, and thus synchronise to, bit-cell centres.
and/or boundaries. Implicit in this technique is the ability to sample, off tape, data at an optimum point within a bit cell under conditions of velocity variation. The detection procedure can be applied to any channel code where a unique combination of successive transition intervals identifies a point within the bit cell.

A dynamic change in tape azimuth occurs as the tape passes the record/replay head. This may be superimposed on a static skew value which is due to the axial offset of sections of the tape reel caused by imperfect tape spooling. Dynamic azimuth variation is primarily caused by the curvature of the tape edges. In addition, the non-uniform profile of the pinch roller imparts a regular azimuth component variation with period proportional to its circumference. This latter component is most likely due to the build up of tape oxide on the surface of the pinch roller.

The combined effect of tape curvature and pinch-roller surface irregularity is to produce a cyclical tape-azimuth variation with period of approximately 0.5 sec. The magnitude of this variation is tape dependent and, for the sample of tapes used in the investigation, varies from $\pm 20\%$ to $\pm 120\%$ of one 10 kbit/sec bit-cell. Tape azimuth characteristics appear to be unrelated to cost or cassette design. The tape azimuth angle is not always linearly distributed across the full width of the tape.

The tape-transport capstan should exert uniform pressure across the full width of the tape as it is forced against the pinch roller. Due to roller wear and tape-oxide build up
however, this pressure is unevenly distributed and concentrated along the axial centre of the roller surface. Counteracting this forward force on the tape is the friction of the tape guide which slows down the tape at its edges. The combined result is to deform the tape along its centre to create a "bow-wave" effect. The symmetry of the "bow wave" varies as the tape-edge curvature causes increased friction at one edge of the tape guide with respect to the other. The extent of the tape deformation depends on tape thickness with the thinner C90 and C120 tapes exhibiting deformation proportionately greater than C60 tape. For this latter tape thickness, average deformation amounts to 14% of one, 10 kbits/sec. bit cell with approximately 18% and 23% for C90 and C120 respectively. Tape deformation reduces the sample time window by the above amounts when data are sampled in parallel across tracks.

Tape - azimuth differences between record and playback, plus tape deformation reduce the rate at which data may be coherently sampled in one operation. Below a data rate of 500 bits/sec/track the above effects are negligible. Errors below this data rate, for a 4-track system, are due to tape dropouts and debris build up on the head. Above this rate, tape skew/deformation errors accrue in proportion to data rate. At approximately 3 kbits/sec/track the error rate due to tape azimuth/deformation is one order of magnitude higher than the "natural" error rate. This figure may be halved by the application of an adaptive decoding method which synchronises to the data on the leading track of the tape; this doubles the effective sample time window.
Errors due to the above tape-transport deficiencies may be eliminated by treating each track independently. However, this approach is computing-time intensive, especially for high track densities. The prevailing tape skew can be monitored by interpolating between the outer-track skew difference and computing the sample point for each track accordingly. This reduces the speed demand on the software but, due to tape deformation, errors attributed to this source will not be eliminated. Although tape skew is not evenly distributed across the full width of the tape, tape-deformation profiles show that it is likely to be linearly distributed from tape edge to centre. Skew may, therefore, be measured, and interpolated between, the outside and centre tracks and used to compute skew/deformation-independent sample points.

The data/error rate ratio depends, in part, on the recording code adopted. Whilst the transition density of Miller code is half that of Biphase-L, its double-density recording capability cannot be realised when track data are sampled in parallel since, in this mode of operation, the sample time window is the same for both codes. Also, the decoding algorithm for Miller code requires two parallel data samples for each decoded sample compared to one for Biphase-L code. For the foregoing reasons the error rate with Miller code is higher than with Biphase-L code when sampling tracks in parallel. Biphase-L code may be adapted to give a higher code rate. This sacrifices the d c-free qualities of Biphase-L but characteristics such as a unique transition interval ratio and bit-cell transition
regularity are preserved. Compared to Biphase-L code this modified code offers a higher efficiency and a higher data rate/error rate for a given recording system.

The predominant error pattern is track burst errors. The frequency of occurrence of these error bursts falls off exponentially with burst length. In addition, isolated, single-bit errors occur and, more frequently, short bursts of multi-track errors due to gross tape skew/deformation conditions during the sampling periods. A single-error-correcting scheme used in isolation is inadequate at data rates above 500 bits/sec/track. Above this rate a multilayered error-correcting strategy is necessary. The effectiveness of any error-correcting scheme depends on the raw error rate which should be as low as possible. Moving Parity error-correction is simple to apply and will handle any single-track burst error length in real time, provided it is reasonably isolated. The multitrack errors generated by parallel sampling can be dispersed by convolutional interleaving. The software overhead for this combined correction scheme is minimal, permitting real-time operation at reasonably high data rates. Increased performance may be achieved by applying Moving Parity error-correction to alternate tracks, but this would require two parity tracks. In addition to the real-time capability of the above scheme, a fundamental advantage is its convolutional structure which obviates the need to block the data and thus format the tape. If this latter restriction is acceptable the Reed-Solomon error-correction code may be incorporated in a cross-interleaved
correction scheme. Software algorithms for the processing of finite-field elements have been devised but their use in real-time applications requires computing power far in excess of that utilised in this investigation.

A track density of 3.33 kbits/sec, with reasonable error rate, has been achieved with a microcomputer of modest processing power. This rate can be increased further with the application of a more powerful processor or a multiprocessor task-sharing approach. The application of low-transition density codes, such as the ISS 2/3 code, and pulse slimming techniques will also contribute to increased data rate. To retain low-cost principles, however, the benefits of write equalisation should be explored. A target track density of the order of 10 kbits/sec is not overly ambitious. This can be doubled if tape speed is similarly increased. The emerging MEV tape with thinner coating would permit a doubling of speed whilst still retaining a reasonable playing time, as would the current C120 tape. Skew and deformation characteristics of MEV tape are yet to be determined but the higher incidence of errors due to C120 tape may be acceptable in some applications.

All the techniques developed in this investigation are for parallel operation and may be applied to any track density provided a microcomputer of comparable word length is used. If an adequate signal-to-noise ratio can be maintained, thin-film, MR read/write recording heads can be applied with high track density. The ensuing increase in data density may be utilised to increase sample resolution or, alternatively, data
rate may be further increased. This latter option, combined with data compression, would give a low-cost, digital recording system capable of encoding and reproducing high quality audio.
6.8. REFERENCES FOR CHAPTER 6.


270
APPENDIX A.

Derivation of Channel Code capacity.
APPENDIX A.

Derivation of channel code capacity.

Consider the finite-state-transition diagram (FSTD) of a d,k code, figure A1.

![Fig. A1. Finite state transition diagram (FSTD) for d,k code.](image)

The FSTD has k+1 states. The numbers associated with each branch of the FSTD represent, no change:0, or change:1. Passage between any of the states generates a coding sequence which is constrained in d,k.

From the FSTD a state transition matrix, \(T\), is formed. The elements of \(T\) are \(t_{ij} = x\) for \(x\) branches from state \(i\) to state \(j\), otherwise \(t_{ij} = 0\).

For example consider a code with \(d,k = 1,3\). The FSTD and state transition matrix are shown in figure A2.

The channel capacity in user bits per channel bit, \(C\), is,

\[ C = \log_2 \lambda \]

where \(\lambda\) is the largest positive eigenvalue of the state transition matrix \(T\).
The largest positive eigenvalue of \( T \) is the largest positive root of the characteristic polynomial of \( T \). The characteristic polynomial is found as follows.

Consider the following square matrix,

\[
\begin{bmatrix}
a_{11} x_1 + \cdots + a_{1n} x_n \\
a_{21} x_1 + \cdots + a_{2n} x_n \\
\vdots \\
a_{n1} x_1 + \cdots + a_{nn} x_n
\end{bmatrix}
= \begin{bmatrix}
\lambda x_1 \\
\lambda x_2 \\
\vdots \\
\lambda x_n
\end{bmatrix}
\]

From equation A1:

\[
\begin{bmatrix}
(a_{11} - \lambda) x_1 + a_{12} x_2 + \cdots + a_{1n} x_n \\
a_{21} x_1 + (a_{22} - \lambda) x_2 + \cdots + a_{2n} x_n \\
\vdots \\
a_{n1} x_1 + a_{n2} x_2 + \cdots + (a_{nn} - \lambda) x_n
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]

Form the determinant of the coefficients of A2:

\[
\begin{vmatrix}
a_{11} - \lambda & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} - \lambda & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nn} - \lambda
\end{vmatrix} = 0 \quad \text{......A3}
\]

Solve A3 to form a polynomial of nth degree in \( \lambda \): this is the characteristic polynomial of \( T \).
Taking the state transition matrix of figure A2:

\[
\begin{vmatrix}
-\lambda & 1 & 0 & 0 \\
1 & -\lambda & 1 & 0 \\
1 & 0 & -\lambda & 1 \\
1 & 0 & 0 & -\lambda
\end{vmatrix} = \lambda^4 - \lambda^2 - \lambda - 1 = 0 \quad \ldots \ldots \text{A4}
\]

Equation A4 is the characteristic polynomial of T.

The largest positive root of equation A4 = 1.465.

Therefore,

channel capacity of 1,3 code = \log_2 1.465 = 0.5515.
APPENDIX B.

Published Papers & Articles.
Real-time microprocessor monitoring of skew angle in a compact cassette multitrack magnetic tape system

T. DONELLY, BSc, MSc, CEng, MIEE*
D. J. MAPPS, BEng, PhD, MinstP, CEng, MIEE*
and
R. WILSON, BEng, MSc, CEng, MIEE†

SUMMARY
Tape-skew variation imposes one of the limits on the upper recording frequency at which digital data may be read coherently from a multitrack tape. It can be measured using timers controlled from marker transitions on the outer tracks of the tape. The timers are interfaced to a microprocessor which applies a ratio technique to indicate a bit-cell displacement which is insensitive to tape velocity and circuit variations. Spectral analysis of the skew of C60 compact cassette tapes reveals a cyclic variation of skew angle of about 2 Hz with a total displacement across the tape of within one bit-cell at 10 kb/s.

The skew angle is not always uniformly distributed across the tape due to tape deformation. The computed values of average tape deformation are 0.67 μm, 0.85 μm and 1.08 μm for C60, C90 and C120 tapes respectively.

1 Introduction
Direct digital recording using magnetic tape and stationary-head recorders is used extensively in the computing and instrumentation fields. Here, the reproduction of recorded data must be as near 100% correct as possible. Consequently the tape mechanisms and associated hardware are usually quite sophisticated and the data rate is chosen to be commensurate with the above aim. A further application of direct digital recording and one tolerant to less than error-free reproduction is digital audio.

The advantages of recording audio signals in digital form are well known. Although digital audio tape recorders for professional use are available and others have been reported the only mass-produced digital audio device currently available is the Compact Disc. Magnetic tape, however, has great potential for the recording of digital data at the very high densities digital audio demands. It also offers a read/write facility and a recording medium which, at present, is less costly than the Compact Disc.

The application of thin-film heads has increased the viability of multitrack formats at reduced tape widths and a data format for Digital Audio using Stationary Heads (DASH format) has already been proposed. This is a multitrack format with twenty tracks spanning 0.25 inch tape—16 data tracks bounded by 2 auxiliary tracks each side.

If data could be recorded at a sufficiently high density the compact cassette tape would make an ideal medium for storing digital audio data. Although primarily designed for recording analogue signals, digital data may be recorded directly on cassette tape in a multitrack format. As the data density increases however, many difficulties arise with data detection—skew angle variation is one such difficulty.

Although the skew behaviour of high-cost computer tape systems has been documented, few or no results are available for compact cassettes. Consequently, the solutions to tape skew which have been proposed, such as single-edge guidance and phase comparison, apply to open reel tape systems and not to closed reel systems such as the compact cassette. To assess the suitability of this medium for high density storage requires more information on the skew behaviour of the various types of compact cassette available.

2 Skew Angle
Skew angle is the angle between the normal to the longitudinal axis of the tape and the axis of the sensing head gaps (Fig. 1). Static skew presents few problems but since skew angle varies with tape motion it is dynamic skew which can limit the recording frequency in a simple practical system, i.e. a system without skew correction.
hardware. Skew-angle variation results in the data across the tape not being presented to the sensing head gaps in an identical manner to that laid down when recording. Consequently, the outer head gaps could be simultaneously sensing data displaced by several bit-cells compared to when they were recorded. The extent of the displacement depends on the frequency of the recorded signal since as this increases the distance on tape occupied by one bit decreases.

Multitrack digital tape recorders employ de-skewing circuitry to overcome the problem of dynamic skew. One technique is to assign to each track a number of shift registers into which the track data may be switched. The data enter the shift register in skewed form when on-line to the clocked out at regular intervals in de-skewed form.

Dynamic skew correction by phase comparison of analogue signals recorded on outer tracks can be implemented by controlling an electrically variable delay line. In this method the wavelength of the correction signals must be sufficiently long so as to preclude the possibility of phase differences in excess of 360°. However, for reliable phase measurement the phase difference should be a large fraction of the total wavelength, suggesting the use of short wavelengths. Further, the problems of measuring phase difference accurately increase as the tape width decreases.

Before the skew correction circuits can be effectively designed a knowledge of the nature of skew is required. Also, knowledge of the maximum skew angle likely to be encountered would give a recording frequency below which de-skewing circuitry would be unnecessary.

Although examined here in a digital application skew-angle variation also has adverse effects on analogue recording where losses of high frequency components and incorrect phasing result.

3 Method of Skew Measurement

The tape skew on playback may be measured relative to the tape skew on recording by encoding the outside tracks of the tape with marker transitions at regular intervals (Fig. 2). On readback the time intervals \( T_A \) and \( T_B \), between marker transitions on opposite tracks appearing under the read heads are measured. From this the displacement \( x \) may be determined:

\[
\cos x = \frac{(T_A - T_B)}{2x}
\]

also

\[
\cos x = \frac{(T_A + (T_A - T_B)/2)}{k}
\]

where

\[
x = \text{skew angle}
\]

\[
x = \text{bit-cell displacement}
\]

\[
k = \text{separation of marker transitions.}
\]

Thus

\[
(T_A - T_B)/2x = (T_A + T_B)/2k
\]

and

\[
x = k(T_A - T_B)/(T_A + T_B)
\]

The units of \( k \) determine the units of \( x \) and may be distance or time. Digital pulses may be positioned on the inner tracks to control timers A and B. Analogue signals may also occupy the inner tracks if required.

The above technique measures the effects of skew-angle variation. In the development of a practical high density digital tape recorder this may be of greater interest than the skew angle itself. However, from the knowledge of the maximum skew angle between the tracks the skew angle may be determined if required.

As a means of verifying the measurement technique skew angles in the range \(+0.3° > x > -0.3°\) were introduced using a rotary micrometer technique to mechanically adjust the head azimuth. This represents a bit-cell displacement of 4 units at 10 kbits/s. The skew angle variation from the microcomputer (the dynamic skew) was averaged to give an indication of the static skew only. From this the skew angle was calculated. The relationship between the mechanically introduced values and the computed values is shown in Fig. 3 which shows a high degree of correlation between the ideal and actual results.

4 Experiment

A series of measurements were made using commercially available audio compact cassette tapes. A number of tapes were used which represented a spread of manufacturers as well as different tape coatings and cassette construction.

The tape transport used in the tests was a Hart iere. Vol. 56, No. 2, February 1986
Electronicst SF925F (cost £39). It features solenoid control, a single motor with integral speed control and a pinch roller tape drive. Two similar tape transports were used, one fitted with a standard 4-track head and one with a custom-built 8-track head. No erase head was fitted, the only tape guides being the ones integral to the read/write heads. Each tape was encoded using a software encoder run on a Research Machines 380Z microcomputer (Fig. 4). The same arrangement was used on readback with the addition of the hardware timers A and B. The spacing of the marker transitions is such that the displacement \( x \) is measured in terms of the percentage of one bit-cell at 10 kb/s.

![Fig. 4. Readback data channel. (Timer A measures interval between pulse edges 3 and 2. Timer B measures interval between pulse edges 1 and 4.)](image)

Typical results are shown in Fig. 5. Here the skew-angle variation of three C60 tapes from different manufacturers is compared. Each shows a total skew-angle variation within one bit-cell at 10 kb/s, tape 1 exhibiting a variation of approximately half that of tape 3. The d.c. offset in the mean of the skew variation is due to lateral movement within the tape cassette. The measured offset can be accounted for by variations in guide roller position brought about by tape spooling.

A common feature of the results is the cyclic variation of skew angle as the tape is played. This was common to all tapes tested and was attributed to the pinch roller of the tape transport. A spectral analysis of the results reveals this as a strong 2 Hz component which, at a tape speed of 4.75 cm/s, gives a wavelength approximately equal to the circumference of the pinch roller. The other harmonic components in the spectrum (Fig. 6) varied between the tapes used. The low frequency components are due to the non-parallel edges of the tape imparted during the tape slitting process, and the high frequency components are due to the cassette tape guides.

![Fig. 6. Spectral components of tape-skew variation.](image)

The ratio technique employed in the measurement makes it insensitive to tape-velocity variations and to the frequency of the timer clock. It is also independent of the slice levels of the two read amplifiers. Since the timers A and B are started and stopped by pulses derived from opposite tracks, each will accumulate a number of pulses proportional to the slice-level difference.

This measurement technique measures the skew with reference to transitions positioned on the outer tracks of a multitrack tape. The azimuth error between data on the outer tracks is not necessarily the maximum. It has been shown that with the multitrack tapes used in computer data recording the skew angle does not remain constant across the width of the tape. It was necessary, therefore, to determine experimentally the skew angle characteristics for the cassette tape and tape transport employed.

Using an 8-track head the time difference between the detection of simultaneously recorded transitions was measured. This measurement was made by recording a 10 kb/s 'in-phase' square waveform on each track. On replay, an exclusive-or between tracks gave pulses whose lengths were proportional to the time differences. These pulses were then used to gate a common clock to a number of counters.

The measurement revealed (for the particular tape transport/head combination used) two predominant skew pattern characteristics (Figs 7(a) and 7(b)). In Fig. 7(a) the skew angle is evenly distributed across the width of the tape—this allowing for about 5% error in the system due to differences between the eight read amplifiers. When the skew angle changes polarity (Fig. 7(b)), however, the centre tracks of the tape reach their corresponding read gaps before the outer tracks suggesting a deformation of the tape along its centre.

The microcomputer was used to read and calculate the average deformation \( D \) (Fig. 7(b)) for C60, C90 and C120 tapes. More than 800 samples per tape were taken during a total playing time of 7 hours. This gave an average tape deformation of 0.67 µm, 0.85 µm and 1.08 µm for C60, C90 and C120 tape respectively.

The total skew indicated comprises a component due to the record process and one due to the read operation; these may be additive or subtractive. This is true whether
the tape has been encoded on the same system as that on which it is being read or otherwise. If the tape was encoded using a different system, however, a component of static skew would be evident and this could easily be compensated for.

5 Conclusion
A new method of accurately measuring dynamic skew-angle variations of compact cassette tapes has been given. It is capable of measuring a wide range of skew angles and has proved insensitive to tape-velocity variations and to circuit parameters. A total skew-angle variation giving a displacement across the tape of within one bit-cell at 10 kb/s is achievable using C60 tape and a medium quality recorder.

Due to tape deformation the skew angle is not always evenly distributed across the width of the tape. This deformation becomes more pronounced as the tape thickness decreases.

6 References
2 Lagadec, R., 'Current status in digital audio', paper delivered to the IERE Fifth International Conference on Video and Data Recording, Southampton, April 1984.

Manuscript first received by the Institution on 14th December 1984 and in revised form on 12th July 1985

Paper No. 2220/REC19
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

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 channel2 and the simple implementation of data encoding and recovery methods.

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.4 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.3 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.

Table 1. Encoding table

<table>
<thead>
<tr>
<th>Data</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>101</td>
</tr>
<tr>
<td>01</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>001</td>
</tr>
<tr>
<td>11</td>
<td>010</td>
</tr>
</tbody>
</table>

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 outside, de-skew tracks.
Table 2. Illegal code bits and their replacement

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Data</th>
<th>Illegal code</th>
<th>Replacement code</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0001</td>
<td>010110</td>
<td>101000</td>
</tr>
<tr>
<td>b</td>
<td>0001</td>
<td>001100</td>
<td>010000</td>
</tr>
<tr>
<td>c</td>
<td>1000</td>
<td>001101</td>
<td>001000</td>
</tr>
<tr>
<td>d</td>
<td>1001</td>
<td>001100</td>
<td>010000</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Data</th>
<th>Data modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>0001</td>
<td>01XX</td>
</tr>
<tr>
<td>d</td>
<td>1001</td>
<td>11XX</td>
</tr>
</tbody>
</table>

The encoder program looks forward over four data words and forms a mask identifying the illegal sequences.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Data</th>
<th>Data modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>0001</td>
<td>01XX</td>
</tr>
<tr>
<td>d</td>
<td>1001</td>
<td>11XX</td>
</tr>
</tbody>
</table>

The illegal 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 kbps 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 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.

The illegal sequences are replaced by a code sequence which always terminates with three zeros.

<table>
<thead>
<tr>
<th>Table 4. Decoding table (X = don't care, N = not all zeros)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous code-</td>
</tr>
<tr>
<td>word last bit</td>
</tr>
<tr>
<td>X</td>
</tr>
<tr>
<td>X</td>
</tr>
<tr>
<td>X</td>
</tr>
<tr>
<td>X</td>
</tr>
<tr>
<td>X</td>
</tr>
<tr>
<td>X</td>
</tr>
<tr>
<td>X</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

Table 5. Summary of decoding rules

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. 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
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:

\[ \cos x = \frac{(T_A - T_B)}{2x} \]  

(1)

where \( x \) is the tape skew angle, \( x \) the bit-cell displacement and \( T_A, T_B \) are time intervals. Also

\[ \cos x = \frac{T_A + 1(T_A - T_B)}{k} \]  

(2)

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

\[ \text{bit-cell displacement } x = \frac{(T_A - T_B)}{k} \]

(3)

In compact cassette tape systems the skew angle \( x \) is in the range \(-0.1^\circ < x < +0.1^\circ\). Therefore

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

\[ \approx \frac{T_A + T_B}{2} \]

(4)

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

\[ x = \frac{(T_A - T_B)}{2} \]  

(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. The following de-skewing technique, however, is robust enough to accommodate a bit-cell displacement error of \(\pm 0.5\) 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. 1.** Bit-cell displacement, \( x \), caused by skew angle \( x \).

**Fig. 2.** Pulses derived from control tracks.

**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 \( \Delta = \frac{1}{2}(T_a - T_b) \). During this time interval the circuit of Fig. 4 generates six initialize pulses.

\[
\text{Shift register input frequency} = \frac{f}{T_a - T_b} = \frac{2N}{2N} = \frac{N}{x} \text{ Hz}
\]

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 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 12 kb/s per track using a Zilog Z800 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


Manuscript received by the Institution in final form in June 1986
Paper No. 2291; REC 2/2

Fig. 5. Skew correction pulses.
AN INTELLIGENT MICROPROCESSOR INTERFACE FOR A LOW-COST DIGITAL MAGNETIC TAPE RECORDER

T. Donnelly, D.J. Napps and R. Wilson

Department of Electrical and Electronic Engineering
Plymouth Polytechnic
Drake Circus, Plymouth, Devon, PL4 8AA, U.K.

A direct-digital, compact-cassette, four-track tape recorder has been interfaced to a personal computer with the minimum of components. The functions of data encoding/decoding and synchronisation are implemented in software. Software is also used to reduce the effects of tape velocity/azimuth variation. A recording density of 2.5 kbits/sec per track has been achieved with an error rate of 1 in 10^5.

1. INTRODUCTION

Developments in data storage technology have sought to keep pace with the increasing use of the microprocessor. Hard/soft magnetic discs and optical storage methods offer high density storage with short access time. Tape systems also offer high-density storage and although the access time is longer than disc the cost per bit is considerably lower. Tape systems also offer an inherent timebase in those applications where the requirement is to store data sequentially. Music, data logging and the emerging technology of still-picture storage [1] are examples of areas where the longer access time of a tape system is acceptable. Other applications include disc backup and home-computer use.

Mainframe computers use expensive tape systems which incorporate a speed-controlled, high performance tape-transport mechanism. Also, data encoding/decoding and synchronising electronics are required to handle the direct digital recording codes employed. A parallel-track format is often used to increase the data density. However, this requires that deskew circuitry be employed to eliminate the effects of tape-azimuth variation. Tape azimuth variation occurs because, in operation, the tape passes the headstack with a skewed motion in both the record and playback modes (fig.1). The primary cause is the "wavy" edge of the tape which is usually imparted during the tape slitting process. Poor tape transport also contributes to the problem. Unless corrected, tape azimuth variation leads to incoherent detection of the data which were recorded broadside across the tape.

Compact-cassette tape recorders form an inexpensive and convenient system for recording digital data. However, the problems of tape speed/azimuth variation are more pronounced than in the mainframe systems referred to above. Consequently, recording is usually confined to a single track at fairly low data rates using analogue encoding. This paper shows how an improved-performance compact-cassette tape recorder with multi-track format can be implemented. Direct, digital recording techniques are applied with the functions of encoding, decoding and bit synchronisation being assumed by intelligent software. Software is also used to eliminate the effects of tape-speed variation and to reduce the effects due to tape-azimuth variation.

![Fig.1: Tape passing headstack with skewed motion.](image)

By utilising the computing power of a personal computer (PC) a robust and practical system can be realised with the minimum of additional hardware.

* Formerly at Plymouth Polytechnic, now at Leeds Polytechnic, Calverley Street, Leeds, LS1 3HE, U.K.
2. RECORDING SYSTEMS HARDWARE

Figure 2 shows a schematic diagram of a four-track compact-cassette tape recording system. The combined record/playback head is a four-track head of the type normally found in auto-reverse audio tape recorders. Because saturation recording is used an erase head is not required.

![Diagram of Digital Tape Recorder System](image)

The write amplifier is formed from one half of a dual parallel input output (PIO) integrated circuit within the PC (fig.3). The record current is fixed by suitable choice of resistor Rr and should be sufficient to saturate the tape coating, a value of 2 mA was used.

![Diagram of Write Amplifier](image)

For peak detection the read amplifier comprises a differentiator/amplifier and a comparator together with the necessary passive components. These may be configured using quadripple comparator and amplifier integrated circuits which are interfaced to the PC via the second port of the PIO. The need to switch the record/playback head is obviated by the use of tri-state devices.

3. RECORDING CODE

Numerous codes have been devised for the recording of data on magnetic media [2]. The code adopted for the compact-cassette system is Bi0-L or Manchester code (fig.4). Although largely superceded by the so-called double-density codes Bi0-L has a number of features which make it suitable for software implementation in this application. With Bi0-L code a change in signal level always occurs at the bit-cell centre. Also, the ratio of intervals between successive transitions of the signal can be used to identify either the bit-cell centre or the bit-cell boundary.

![Diagram of Bi0-L Code](image)

4. RECORDING SOFTWARE

The recording of data can be timed by using either a software timer or the internal counter timer circuit (CTC) of the PC. The parallel bit structure of the PC permits each track of the tape to be simultaneously recorded, one bit of the processor word per track. Of the two processes, record/playback, the record process is the simplest: the compliment of the data are transmitted for half a cycle followed by the true value for the remainder of the period.
5. PLAYBACK SOFTWARE

The playback process is far more demanding of processor time than the record operation. The playback software must:

(a) Identify each bit-cell centre or boundary.

(b) Remain synchronised to the data throughout replay.

(c) Calculate the simultaneous sampling point for all tracks and decode the data from Bi~L back to binary code.

The above must be accomplished in real time under conditions of tape azimuth and velocity variations.

Bit cell identification is achieved by timing the interval between transitions of the recovered signal and computing the ratio of successive intervals (fig.4). The third transition of a 1:2 ratio occurs at the bit-cell centre whilst that of a 2:1 ratio marks the bit-cell boundary. 1:1 ratios mark either the bit-cell centre or boundary and are therefore ignored.

To speed up the computation of ratios the subtraction and addition instructions of the processor are used. If the result of subtracting two successive intervals is positive the ratio is either 2:1 or 1:1. Similarly a negative value would indicate a ratio of either 1:2 or 1:1. The ambiguity is resolved by a further subtraction of one half of the second interval in the first case or by adding one half of the first interval when the initial result is negative. The sample point from a single track is used for identification purposes. Once identification is achieved the playback software continuously monitors the ratios of successive intervals to maintain synchronism throughout replay. The use of ratios in this way gives a system which is independent of tape velocity.

To decode, the four data tracks are simultaneously sampled at the centre of the second half of the bit cell. The sample point is determined from a knowledge of the current duration of a bit cell. This is stored and updated when either a 2:1 or a 1:2 ratio is measured. Thus, a precise sample point is obtained which is also independent of tape velocity.

The above sample point is determined using the information from a single, fixed track. The outputs of the four tracks will, however, be "out of phase" due to tape azimuth variations. Previous work in this area [3] has shown that tape-azimuth variations can typically produce a displacement across the tape of up to ± 1/4 bit cell at a track data rate of 2.5 kb/s.

Thus, when a single, fixed track, is used to determine the common sample point of all four tracks the azimuth variation gives a sample window of ± 1/4 bit cell at this frequency (fig.5a). Increasing the recording frequency above 2.5 kb/s per track will result in an increasing number of playback errors due to tape azimuth variation.

To the four data tracks are simultaneously sampled at the centre of the second half of the bit cell. The sample point is determined using the information from a single, fixed track. The outputs of the four tracks will, however, be "out of phase" due to tape azimuth variations. Previous work in this area [3] has shown that tape-azimuth variations can typically produce a displacement across the tape of up to ± 1/4 bit cell at a track data rate of 2.5 kb/s.

The effective sample window may be doubled by synchronising to the signal on the track which changes first, the leading track (fig.5b). Transitions of the signals on all four tracks are detected by polling the track signals and performing an exclusive OR operation on successive samples until a change occurs. Since the signals on each track always change at the bit-cell centres the leading track may be determined. A simultaneous sample of all four tracks is taken as late as possible after the bit-cell centre of the leading track and therefore, provided the other track signals have changed within 1/2 bit cell of the leading track, the sample will be valid. The rate of tape-azimuth variation is relatively low compared to the recording frequency [3]. However, the software polls and checks for the leading track at every bit-cell centre.
6. RESULTS

A Z80-based personal computer with 4 MHz clock was used to encode a number of proprietary audio compact-cassette tapes. These C60 and C90 tapes were encoded with a 4-bit pseudo random binary sequence (PRBS). On playback the data were decoded and checked against a software-generated PRBS. The execution time of the assembly-code programme limited the recording frequency to approximately 3 kb/s per track, although this included error checking and logging overheads. The results are shown in figures 6 and 7. At low data rates the errors incurred are due to tape dropouts caused by tape surface irregularities and the occasional build up of debris on the tape head. As the data rate increases these error mechanisms become more effective as the recorded signal wavelength shortens. However, errors due to tape-azimuth variation predominate at the higher frequencies. There is a clear improvement in performance when the software continuously synchronises to the first track to change compared to when the software synchronises to a fixed track. This is indicative of the wider sampling window.

The errors which do occur are mainly single-bit errors. Figure 7 shows a typical distribution of error bursts taken over three hours at a data rate of 1.5 kb/s per track. Further work on the categorisation of errors in this system has shown that in general the errors are due to single track failures.

Fig.5b. Sample timing window is ±1/2 bit cell when software synchronised to the leading track.

Fig.6. Error rate performance for fixed and variable synchronisation.

Fig.7. Typical distribution of errors over 3 hours at 1.5 kbits/sec./track.
7. CONCLUSIONS

By utilising the computing power of a PC an inexpensive, direct-digital, compact-cassette tape recording system has been configured.

The use of intelligent software permits a parallel-track format with data rates in excess of those found in analogue compact-cassette tape systems. The software has proved to be robust with typical error rates of 1 in $2 \times 10^5$ at 1 k.bits/sec per track and 1 in $8 \times 10^5$ at 1.5 k.bits/sec per track. The hardware required for the interface is minimal.

Typical applications would be as a data store and/or disc backup in a PC and within portable data logging and medical recording equipment. A considerable increase in data rate would be necessary before it could be used to store digitised music. However, digitised speech could be stored with a modest increase in data rate.

Future developments include the implementation of software error-correction and the application of a new code which would increase the data rate of the system described here by 50%. Improved performance will also result by increasing the number of tracks and through the use of a faster PC to reduce the effect of tape-azimuth variation by synchronising to each track.

8. ACKNOWLEDGEMENTS

Grateful thanks is extended to Devon County Council and the SERC for financial support.

9. REFERENCES


A MICROPROCESSOR-BASED CODEC FOR A LOW-COST DIGITAL TAPE RECORDER

T. Donnelly (1), D.J. Mapps (1) and R. Wilson (2)

(1) Plymouth Polytechnic, Devon.
(2) Leeds Polytechnic, Yorks.

I. INTRODUCTION

The widespread application of digital techniques has generated a complimentary demand for cost-effective digital-data recording systems.

Personal computers (PC) are capable of generating vast amounts of data at reasonable cost. These data are usually stored on hard or floppy disc backed up by tape storage systems. In the entertainment field the storage and handling of music in digital form is commonplace. Digital tape systems have been employed at the professional level for some time and in the consumer field the compact disc (CD) is now being followed by the rotary-head, digital-audio tape recorder (R-DAT).

A further area where digital recording techniques are employed is that of data logging. Here, convenience and equipment portability are seen as attractions in addition to low cost.

The virtues of convenience, low-cost and portability are exemplified in the compact-cassette tape system. Although originally designed for analogue recording it may be adapted to record digital data directly. The computing power of a PC may be utilised to perform the function of a CODEC thus removing the requirement for much of the conventional circuitry normally associated with a digital recording system. Through the use of a PC and the adoption of a multitrack format a moderately high-density tape storage system may be realised at a low cost.

This paper reviews the techniques employed to realise a microprocessor-based CODEC by presenting two digital tape recorder systems: a basic recording system and an enhanced system. However, before detailing these techniques two major problems associated with the low-cost format of compact cassette will be considered: tape-velocity variation and tape skew.

Since information to be recorded is encoded in terms of the distance between magnetic flux changes the velocity of the tape must be constant. Low-cost tape decks, such as those used with compact-cassette tapes, suffer from velocity variation. To combat this, speed-control circuitry is often employed. Even so, speed related errors can occur if this problem is not fully accounted for.

A more serious problem is tape-skew variation (fig.1). As the tape moves across the headstack the skew angle varies about zero. Previous work on the measurement of skew [1] has shown that the rate of change of skew angle is of the order of 2 Hz and its magnitude is sufficient to give a displacement across the tape normally occupied by one bit at a recording rate of 10kb/s. The effect of tape skew on a single track is to reduce the effective read gap of

REPRODUCED SOUND 3
the replay head and so reduce the amplitude of the detected signal. The effect on multitrack recordings is that data recorded broadside across the tape may not be sensed in the order in which they were recorded thus causing errors.

Fig. 1 Tape passing headstack with skewed motion.

2. BASIC RECORDING SYSTEM

A basic recording system may be configured as shown in Fig. 2. The 4-track record/playback head is of the type normally found in auto-reverse audio tape recorders, an erase head is not required. Data are recorded by applying complimentary signals to each of the four head coils from the 8-bit peripheral input/output (PIO) chip of the PC.

Fig. 2 Basic recording system.
A resistor should be placed in series with each coil to set the record current to approximately 2 mA. The read amplifier is the only conventional hardware required. It amplifies and slices the low level tape signal before application to a second PIO.

Software is used to encode, detect and decode data. A recording code which has been used with some success is Biø-L code (fig.3). To record, the compliment of the data are transmitted followed by the true value. The data may be timed by using a software timer or the internal counter/timer of the PC may be utilised. The parallel bit structure of the PC permits each track of the tape to be simultaneously recorded, one bit of the processor word per track.

Biø-L code has a number of features which make it suitable for software implementation in this application. With Biø-L code a change in signal level always occurs at the centre of the space occupied by one bit - the bit cell. Also, the ratio of intervals between three successive transitions of the signal can be used to identify either the bit-cell boundary or the bit-cell centre. A 1:2 ratio of two successive intervals marks the bit-cell centre whereas following a 2:1 ratio the bit-cell boundary is reached. This characteristic is used in the decoding process.

The decoding software polls the output of a single track and times the interval between transitions. The ratio of successive intervals is computed and the result indicates the current point within the bit cell. If a 1:1 ratio is encountered it is ignored. Because ratios are involved this method of synchronising to the data is tape-velocity independent.

The process of measuring and computing the ratios is continuous. Each time a 2:1 or a 1:2 ratio is measured the "1" value is stored in the PC and is used by the software to calculate an accurate sample point which is halfway through the second half of the bit cell. This method of determining the sample point is also independent of tape velocity.
A MICROPROCESSOR-BASED CODEC FOR A LOW-COST DIGITAL TAPE RECORDER

Although the sample point is determined by synchronising to a single track all four tracks are simultaneously sampled at this point. The attraction of this technique is the simplicity of the software required. However, skew-generated errors will occur especially at the higher data rates.

A software refinement of the technique is to make the track to which the software synchronises variable. The advantage of this is that the first track signal to change in time (the leading track) may be used for synchronisation and the sample point determined as late as possible within this bit cell. Since the three other track signals will change some time after the leading track signal (due to tape skew) the late sample point will coincide with the three track signals in the second half of their respective bit cells and, if the total skew is within $\pm \frac{1}{2}$ bit cell, the data will be sampled without error. This technique effectively doubles the tape-skew tolerance of the software compared to the simple arrangement described above.

Results showing error rate against data rate are shown in fig. 4. The advantage of synchronising to the leading track is clearly seen. The execution time of the software limited the data rate to 2.5 kb/s/track. However, the software included error-verification procedures and the clock rate of the microprocessor used was only 4 MHz. This data rate is capable of being significantly increased by the application of a faster processor and the elimination of the error-verification software overheads.

Fig. 4. Error rate performance for fixed and variable synchronisation.
3. ENHANCED RECORDING SYSTEM

With the addition of specialised de-skew and clock hardware the performance of the basic system of fig.2 may be improved. In this enhanced recording system an 8-track head is used and the recently developed ISS 2/3 recording code is employed to give a recording rate of 10 kb/s/track. With the ISS 2/3 code two data bits are recorded as three code bits (Table 1). The advantage of this code is that since no adjacent ones are allowed to occur a greater recording density can be achieved. A code sequence which would give two adjacent ones is deemed illegal and is replaced by a code sequence conforming to the rules of the code (Table 2). Although originally developed for recording data serially a parallel software algorithm for the ISS 2/3 code has been devised for this application [3].

The two outside tracks of the tape are encoded with control pulses whilst the six inner tracks are occupied by data. The control pulses serve two functions. They are spaced every three code bits so that the decoding software can identify each code-bit group. The de-skew hardware also uses these control pulses to calculate the tape skew.

<table>
<thead>
<tr>
<th>Data</th>
<th>Code</th>
<th>Data</th>
<th>Illegal code</th>
<th>Replacement code</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>101</td>
<td>0000</td>
<td>101101</td>
<td>101000</td>
</tr>
<tr>
<td>01</td>
<td>100</td>
<td>0001</td>
<td>101100</td>
<td>100000</td>
</tr>
<tr>
<td>10</td>
<td>001</td>
<td>1000</td>
<td>001101</td>
<td>001000</td>
</tr>
<tr>
<td>11</td>
<td>010</td>
<td>1001</td>
<td>001100</td>
<td>010000</td>
</tr>
</tbody>
</table>

Table 1. ISS 2/3 Encoding table

In this system a hardware clock is used for each of the six inner tracks. On replay these clock the code bits off each track into one of two data flip-flops associated with each of the six data tracks (fig.5). The pulses on the two outer tracks are used to measure the magnitude and polarity of the prevailing tape skew. The output of the skew measurement circuit comprises six pulses each of which is used to control the clock input to the data flip-flops. The time over which the sequence of six pulses occurs is equal to the time displacement across the tape due to skew. Thus, the six data bits, which were recorded across the tape, are routed into their respective data flip-flop after which they are multiplexed into the microprocessor. Two sets of registers are used to give an acceptable tolerance to tape skew at this recording rate. This may be increased by employing more than two sets of registers on a proportional basis.

The error rate of this system was assessed by recording a "sawtooth" waveform digitised to six bits resolution. On replay the waveform was decoded and outputted to a digital-to-analogue (DAC) converter where it was viewed with an oscilloscope. Although performance was marred somewhat by an excessive amount of inter-gap crosstalk within the replay head the sawtooth remained in "synch" failing only when the skew measurement circuit was disabled.
A microprocessor-based CODEC has been implemented to produce a low-cost digital tape recorder. Two, direct digital compact cassette recording systems are presented. Data are recorded on proprietary audio compact-cassette tapes.

With the basic system a data rate of 10 kb/s has been achieved with a reasonable error count. This could be significantly increased with the application of a faster processor. An 8-track head may also be used with the same software to double the data rate. The system may be applied as a low-cost data logger, it could also be used to store low-grade digitised audio signals where data compression techniques are employed.

The second system gives an improved performance but specialised hardware is required and the 8-track head used is not a standard item. Once again a faster processor could be applied either to increase the data rate and/or reduce the specialised hardware requirement. A further improvement would be to integrate data and control pulses on the outer tracks and thus improve the data rate by 30%.

Currently under development is the application of an 18-track, thin-film magneto-resistive head which will be used with the software developed for the above systems run on a 16-bit microprocessor.
5. REFERENCES


TIME DELAY SOLENOID SWITCH USING VMOS FETS

This circuit was developed for a solenoid-operated cassette tape-deck which was to be interfaced to a microprocessor. The pull-in current of the solenoid was specified as twice the holding current. This required a current switch with a time delay sufficient for the solenoid to operate. A simple solution was found which used a CR network for the delay and two N-channel enhancement mode VMOS FETs as the switch.

With the output of the buffer at logic 0 the solenoid is off. A high at the output causes both V₁ and V₂ to conduct but the solenoid current is routed through V₂ as D₁ is reverse biased. After a delay caused by the CR network, V₂ ceases to conduct and V₁ takes the solenoid current which is one half of that taken by V₂. R₁ and R₂ are "gate stoppers" to prevent parasitic oscillation and D₂/D₃ are protection diodes.

The level of current switched depends on the FET selected. The delay depends not only on the CR value but also on V₂ gate threshold voltage and the collector supply voltage to the buffer Vₑ. However, if Vₑ is about three times the threshold voltage then the delay is given directly by the product CR. In the application described, a current of 600/300 mA is switched with a delay of 75 ms.

T. Donnelly, Plymouth, Devon.

SYNCHRONISED DATA CLOCK

In many applications a locally generated clock has to be synchronised to incoming digital data. One way of achieving this is to use a 555 timer in the astable mode and to trigger it from each transition of the data (Fig. 1). R₁, R₂, and C are selected to give the correct clock frequency with a waveform which is slightly asymmetric. The transitions of the data are detected by the 74136 open-circuit collector exclusive OR gate. The final stage of the OR gate causes the capacitor C to discharge to the correct threshold level, approximately Vₑ/3. The output of the timer is thus forced high as it resets (Fig. 2). During the absence of data transitions the timer "free-wheels".

The circuit has been successfully employed in the recovery of digital data from magnetic tape.

T. Donnelly, Plymouth, Devon.

ABACUS DESIGN FOCUS COMPETITION

Abacus Electronics, the Newbury-based component and systems distributor will award an additional prize of £100 for what it judges to be the best Design Focus contribution published each month.

The winner of the competition for January is R. A. Worsley with his "Floating SMPS stabiliser".
APPENDIX C.

Program Listings.
; MANTRCK
; PROGRAM TO READ DATA
; IN MANCHESTER CODE AND
; CATALOGUE TRACK ERRORS
; APPROX 1KHZ
; CTC2 USED TO MEASURE RATIOS
; 11/12/1986
; WRITE PORT P108A: 04D, 06C
; READ PORT P108B: 05D, 07C
; WRITE CONTROL P107A: 00D, 02C
; READ CONTROL P107B: 01D, 03C
; LED DISPLAY P109A: 08D, 10C
; CTC1: 00C, CTC2: 00D

0037 = CW=00110111B
000C = CTC1=00H
000D = CTC2=0DH
0008 = LEDS=08H
000A = DEL=0AH

0112 = OUTMX=112H
0118 = BBUFV=118H
0106 = USR=106H
0109 = USRVF=109H
010C = USRVT=10CH
011A = SUBPTR=11AH

0106 ORG USR
0106 C3FC6A JP USRM
0118 ORG BBUFV
0118 016E DEFW BASEND
011A ORG SUBPTR
011A 0069 DEFW CLINK

6900 ORG 6900H

6900 0000 CLINK: DEFW 0
6902 05 DEFB 5
6903 53554252 DEFM 'SUBRU'
6908 0A69 DEFW SUBRT

690A 3E05 SUBRT: LD A, 05H
690C D300 OUT (00H), A ; SET DECK TO PLAY
690E 3E0A DELAY: LD A, DEL
6910 2600 LD H, 00H
6912 2E00 LD L, 00H
6914 24 LOOP: INC H
6915 CA1B69 JP Z, DELA
6916 C31469 JP LOOP
6918 2C DELA: INC L
6919 C2269 JP Z, DELB
691B 3D DELB: DEC A
6922 CA2969 JP Z, ON
6923 C31469 JP LOOP
6926 C937
6929 3E37 ON: LD A, CW ; INIT' CTC
692B D30D OUT (CTC2), A
692D ED5E IM2
692F 210A6B LD HL, TABLE
6932 7C LD A, H
6933 ED47 LD I, A
6935 FD21E66A LD IY, EX1T
6939 FD220C6B LD (TABLE+02H), IY
693D 3E02 LD A, 02H
693F 85 ADD A, L
6940 D303 OUT (03H), A
6942 3E4F LD A, 4FH
6944 D303 OUT (03H), A
6946 3E87 LD A, 87H
6948 D303 OUT (03H), A
694A FB EI
694B D9 EXX
694C 1E80 LD E, 80H
694E 21006C LD HL, BUFF1
6951 3E00 LD A, 00H
6953 77 BACK1: LD (HL), A
6954 2C INC L
6955 C25369 JP NZ, BACK1 ; RESET BUFFER
6958 21006D LD HL, BUFF2
695B 77 BACK2: LD (HL), A
695C 2C INC L ; SET DISP BUFFER TO ZERO
695D C25669 JP NZ, BACK2
6960 21006C LD HL, BUFF1
6963 D9 EXX
6964 DB05 IN A, (05H) ; READ IN 1ST DATA BYTE
6966 47 LD B, A ; SAVE
6967 DB05 POL1: IN A, (05H)
6969 A8 XOR B ; 1ST DATA CHANGE?
696A E608 AND 08H
696C CA6769 JP Z, POL1 ; NO DATA CHANGE
696F 3EFF LD A, OFFH ; 1ST DATA CHANGE
6971 D30D OUT (CTC2), A ; START TIMER
6973 DB05 IN A, (05H)
6975 47 LD B, A
6976 DB05 POL2: IN A, (05H)
6978 A8 XOR B ; WAIT FOR
6979 E608 AND 08H
697B CA7669 JP Z, POL2
697E DB0D IN A, (CTC2) ; GET PERIOD VALUE N-C1
6980 4F LD C, A ; SAVE PERIOD VALUE N-C1

C3
LD A, CW : STOP PERIOD COUNTER

OUT (CTC2), A

LD A, OFFH : START TIMING

OUT (CTC2), A : 2ND PERIOD

SUB C : TRUE VALUE OF 1ST PERIOD

LD C, A : SAVE TRUE VALUE OF 1ST PERIOD

AGAIN: IN A, (05H)

LD B, A

POL3: IN A, (05H) : WAIT FOR 3RD DATA

JP Z, POL3

XOR B : CHANGE

AND 08H

CABE69

JP M, NEGPD : EITHER 1:2 OR 1:1

POSPD: SRL D : EITHER 2:1 OR 1:1

SUB D : 1ST PERIOD-2ND PERIOD

JP M, NEGPD : EITHER 1:2 OR 1:1

SUB D : 1ST PERIOD-2ND PERIOD-2ND PER

SUB D : 1ST PERIOD-2ND PERIOD-2ND PER

RL D : 1:1 TRY AGAIN

LD C, D : AFTER MAKING 2ND PERIOD

JP AGAIN : THE 1ST PERIOD

BDNARY: LD L, D : SAVE 1/4 BIT PERIOD

LD H, L

LD C, D

LD E, 00H : SAMPLE 1 FLAG

LD C, D

JP SAMP

NEGPD: SRL C : EITHER 1:2 OR 1:1

ADD C : 1ST PERIOD-2ND PERIOD+1ST PER

JP M, CENTRE : 1:2 RATIO, CENTRE FOUND

LD C, D : 1:1 RATIO, TRY AGAIN

JP AGAIN : AFTER MAKING 2ND PERIOD THE 1

CENTRE: LD L, C : SAVE 1/4 BIT PERIOD

LD H, L

LD C, D

LD E, 01H : SAMPLE 1 FLAG

IN A, (05H)

LD B, A

IN A, (CTC2) : GET PERIOD COUNTER

NEG

ADD OFFH : TRUE VALUE

CP L : SAMPLE POINT?
JP M, ONCMR ; NO
LD A, L ; UPDATE NEXT
ADD L ; COMPARISON POINT
LD L, A
BIT 0, E ; 1ST OR 2ND HALF OF BIT CELL
JP Z, FIRST
JP 2, FIRST
LD E, 00H ; SECND:
EXX
BIT 7, E ; END OF RUN
JP Z, CAT ; YES
EXX
IN A, (08H) ; GET DATA
CP 00H ; ALL ZEROS?
JP Z, ONCMR ; OK, CONTINUE
CP OFFH ; ALL ONES?
JP Z, ONCMR ; OK, CONTINUE
ERROR: EXX
LD L, A
OUT (08H), A ; ERROR TO LEDS
INC (HL) ; LOG THE ERROR
JP Z, OVER
JP Z, OVER
EXX
JP ONCMR
JP ONCMR
DEC (HL) ; OVER:
JP CAT ; OVERFLO
LD E, 01H ; FIRST:
IN A, (05H) ; GET INPUT
XOR B ; DATA CHANGE?
AND 08H
JP Z, SAMPL ; NO
AND 08H
JP Z, SAMPL
IN A, (CTC2) ; GET PERIOD VALUE N-C2
LD D, A ; SAVE PERIOD VALUE N-C2
LD A, CW ; STOP PERIOD COUNTER
OUT (CTC2), A ; COUNTER
LD A, 0FFH
OUT (CTC2), A ; START TIMING PERIOD AGAIN
SUB D ; TRUE VALUE OF 2ND PERIOD
LD D, A ; SAVE IN D
LD A, 79
SUB D ; 1ST PERIOD-2ND PERIOD
JP M, M1NPD ; EITHER 1:2 OR 1:1
SUB D ; 1ST PERIOD-2ND PERIOD-2ND PER
JP M, M1NPD
JP M, M1NPD
JP M, M1NPD
PLUSPD: SRL D ; EITHER 2:1 OR 1:1
JP P, EDGE ; 2:1 FOUND
LD L, H
RL D ; 1:1, TRY ONCE MORE
LD C, D ; AFTER MAKING THE 2ND
JP SAMP ; AFTER MAKING 2ND
JP M, MIDL ; 1:2 RATIO FOUND
MINPD: SRL C ; EITHER 2:1 OR 1:1
ADD C ; 1ST PERIOD-2ND PERIOD+1ST PER
JP M, MIDL
LD L, H
LD C, D
JP SAMP

LD L, C
LD H, L
LD E, 01H
LD C, D
JP SAMP

LD L, D
LD H, L
LD E, 00H
RL D
LD C, D
JP SAMP

LD L, 00H
STPOST: INC L
JP Z, DISP
LD D, 09H
LD C, 00H
RESET COUNTER

LD L, C
LD C, 00H
RLC L
JP Z, FORW
ALL BITS CHECKED

INC C
JP REPEAT
REPEAT UNTIL ALL

LD A, C
NO. OF 1'S TRACK ERRORS
SAVE IN TEMP STORE

LD A, (HL)
NO. OF 1'S TRACK ERRORS
SAVE IN TEMP STORE

LD E, A
LD A, L
CPL
LD L, A
LD A, L
CPL
ADD D
TOTAL ERRORS
LD B, L
SET ASIDE L
LD D, A
SET ASIDE TOTAL ERRORS

LD A, C
NO. OF 1'S IN DATA
CP 01H
JP Z, ONES
CP 02H
JP Z, TWOS
CP 03H
JP Z, THREES
CP 04H
JP Z, FOURS

LD L, B
TOO MANY ERRORS TO LOG
JP STPOST
6A8C 2E00  
6A8E CDB06A  
6A91 68  
6A92 C3506A  

6A95 2E10  
6A97 CDB06A  
6A9A 68  
6A9B C3506A  

6A9E 2E20  
6AA0 CDB06A  
6AA3 68  
6AA4 C3506A  

6AA7 2E30  
6AA9 CDB06A  
6AAC 68  
6AAD C3506A  

6A00 6A01 24  
6A03 C340  
6A05 C4E26A  
6A06 2C  
6A07 CB48  
6A09 C4E26A  
6A0B CB50  
6A0C C4E26A  
6A0E CB58  
6A10 C4E26A  
6A12 2C  
6A14 CB60  
6A16 C4E26A  
6A18 CB68  
6A1A C4E26A  
6A1C 2C  
6A1E CB70  
6A20 C4E26A  
6A22 2C  
6A24 CB78  
6A26 C4E26A  
6A28 25  
6A2A C7  

6A8C 2E00  
6A8E CDB06A  
6A91 68  
6A92 C3506A  

6A95 2E10  
6A97 CDB06A  
6A9A 68  
6A9B C3506A  

6A9E 2E20  
6AA0 CDB06A  
6AA3 68  
6AA4 C3506A  

6AA7 2E30  
6AA9 CDB06A  
6AAC 68  
6AAD C3506A  

6AB0 24  
6AB1 CB40  
6AB3 C4E26A  
6AB6 2C  
6AB7 CB48  
6AB9 C4E26A  
6ABC 2C  
6ABD CB50  
6ABF C4E26A  
6AC2 2C  
6AC3 CB58  
6AC5 C4E26A  
6AC8 2C  
6AC9 CB60  
6ACB C4E26A  
6ACE 2C  
6ACF CB68  
6AD1 C4E26A  
6AD4 2C  
6AD5 CB70  
6AD7 C4E26A  
6AD9 2C  
6ADB CB78  
6ADD C4E26A  
6AE0 25  
6AE1 C9  

ONES:  
LD L, 00H  
CALL TRKID  
; SET BASE ADDRESS  
LD L, B  
JP STPOST  

TWOS:  
LD L, 10H  
CALL TRKID  
; SET BASE ADDRESS  
LD L, B  
JP STPOST  

THREES:  
LD L, 20H  
CALL TRKID  
; SET BASE ADDRESS  
LD L, B  
JP STPOST  

FOURS:  
LD L, 30H  
CALL TRKID  
; SET BASE ADDRESS  
LD L, B  
JP STPOST  

TRKID:  
INC H  
; BUFF/ DISP/ ADDRESS  
BIT 0, B  
CALL NZ, STOR  
INC L  
BIT 1, B  
CALL NZ, STOR  
INC L  
BIT 2, B  
CALL NZ, STOR  
INC L  
BIT 3, B  
CALL NZ, STOR  
INC L  
BIT 4, B  
CALL NZ, STOR  
INC L  
BIT 5, B  
CALL NZ, STOR  
INC L  
BIT 6, B  
CALL NZ, STOR  
INC L  
BIT 7, B  
CALL NZ, STOR  
DEC H  
RET
6AE2 7A STOR: LD A, D ; GET UPDATE VALUE
6AE3 86 ADD A, (HL) ; ADD TO TALLY
6AE4 77 LD (HL), A ; RE-STOR
6AE5 C9 RET

6AE6 CB7B EXIT: BIT 7, E ; REGISTERS?
6AE8 C2F26A JP NZ, NOCHGE ; NO
6AEB D9 CHANGE: EXX
6AEC 1E40 LD E, 40H
6AEE D9 EXX
6AEF FB EI
6AF0 ED4D RETI
6AF2 1E40 NOCHGE: LD E, 40H
6AF4 FB EI
6AF5 ED4D RETI
6AF7 3E00 DISP: LD A, 00H ; STOP THE DECK
6AF9 D300 OUT (00H), A ; RETURN TO BASIC
6AFB C9 RET

6AFC CD0901 USRM: CALL USRVF
6AFF 21006D LD HL, BUFF2
6B02 19 ADD HL, DE
6B03 46 LD D, (HL)
6B04 3E00 LD A, 00H
6B06 CD0C01 CALL USRVT
6B09 C9 RET

TABLE:
6C00 ORG 6C00H
6D00 ORG 6D00H
6E00 ORG 6E00H
6F00 ORG 6F00H

BUFFEND:
6E00 00 DEFB 0

BASEND:
0000 END

NO ERRORS

C8
ORG 0100H
TABLE=02FOH
DEL=0AH
POINT=0300H
*L

; ISSKSASW

; ISS2/3 ENCODER
; REWIND TAPE AND WRITE
; ISS2/3 ENCODED DATA
; AND DESEWER PULSES
; FROM CHECKLIST WITHOUT CTC
; FREQUENCY DETERMINED
; BY DELAY: RECDEL

; 20/6/1985

; WRITE PORT P108 04D, 06C
; READ PORT P108B 05D, 17C
; WRITE CONTROL P107A 00D, 02C
; READ CONTROL P107B 01D, 03C

; REG B HOLDS NEXT MASK
; REG C HOLDS CURRENT MASK
; REG D IS TEMPORARY STORE
; REG E HOLDS MODIFY MASK
; REG B' HOLDS OLD OUTPUT
; DATA ADDRESS: (HL)

0100 F3
0101 CD8E01
DI
CALL RESET

0104 210003
LIST: LD HL, POINT
0107 3E00
LD A, 00H
0109 77
BACK: LD (HL), A
010A 2C
INC L
010B 3C
INC A
010C C20901
JP NZ, BACK

010F CD9701
CALL INITREG

0112 CDE001
REWIND: CALL STOP
0115 3E08
LD A, 08H
0117 U300
OUT (00H), A
0119 FB
EI

011A 00
WAIT: NOP
011B C31A01
JP WAIT
; WAIT FOR REWIND

011E CDE001
FIN: CALL STOP
0121 0D
STOP THE TAPE
DEC C
0122 CA2701
JP Z, WRITE
0125 F3
DI
0126 76
HALT
; WRITE ENDED

C9
<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0127 E01</td>
<td>WRITE: LD A, 01H</td>
<td>SET</td>
</tr>
<tr>
<td>0129 D300</td>
<td>OUT (00H), A</td>
<td>TO PLAY</td>
</tr>
<tr>
<td>012B CDE401</td>
<td>CALL DELAY</td>
<td></td>
</tr>
<tr>
<td>012E 3E02</td>
<td>LD A, 02H</td>
<td>SET PASS CNTR</td>
</tr>
<tr>
<td>0130 4F</td>
<td>LD C, A</td>
<td>TO&gt;1</td>
</tr>
<tr>
<td>0131 3E07</td>
<td>LD A, 07H</td>
<td>SET DECK</td>
</tr>
<tr>
<td>0133 D300</td>
<td>OUT (00H), A</td>
<td>TO WRITE</td>
</tr>
<tr>
<td>0135 CDD001</td>
<td>REGINIT: CALL REGIS</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0138 2C</td>
<td>FORMASK: INC L</td>
<td>GET 2ND WORD</td>
</tr>
<tr>
<td>0139 7E</td>
<td>LD A, (HL)</td>
<td>STORE</td>
</tr>
<tr>
<td>013A 57</td>
<td>LD D, A</td>
<td></td>
</tr>
<tr>
<td>013B 2C</td>
<td>INC L</td>
<td></td>
</tr>
<tr>
<td>013C 7E</td>
<td>LD A, (HL)</td>
<td>GET 3RD DATA WORD</td>
</tr>
<tr>
<td>013D B2</td>
<td>OR D</td>
<td></td>
</tr>
<tr>
<td>013E 47</td>
<td>LD B, A</td>
<td>SAVE NEXTMASK, ILLEGAL</td>
</tr>
<tr>
<td>013F 79</td>
<td>LD A, C</td>
<td>GET CURNT ILLEG SEQ</td>
</tr>
<tr>
<td>0140 2F</td>
<td>CPL</td>
<td></td>
</tr>
<tr>
<td>0141 B0</td>
<td>OR B</td>
<td>TERMIN CURR ILLEG SEQ</td>
</tr>
<tr>
<td>0142 47</td>
<td>LD B, A</td>
<td>SAVE NEXT MASK ILLEG</td>
</tr>
<tr>
<td>0143 2C</td>
<td>INC L</td>
<td>SEQUENCES IDENTIFIED BY ZEROS</td>
</tr>
<tr>
<td>0144 7E</td>
<td>LD A, (HL)</td>
<td>GET 4TH WORD</td>
</tr>
<tr>
<td>0145 2F</td>
<td>CPL</td>
<td>LAST BIT IDENT BY ZEROS</td>
</tr>
<tr>
<td>0146 B0</td>
<td>OR B</td>
<td>AND ZEROS</td>
</tr>
<tr>
<td>0147 2F</td>
<td>CPL</td>
<td>MODIFY MASK, ZEROS</td>
</tr>
<tr>
<td>0148 5F</td>
<td>LD E, A</td>
<td></td>
</tr>
<tr>
<td>0149 2D</td>
<td>DEC L</td>
<td>SAVE MODIFY MASK</td>
</tr>
<tr>
<td>014A 2D</td>
<td>DEC L</td>
<td></td>
</tr>
<tr>
<td>014B 2D</td>
<td>DEC L</td>
<td></td>
</tr>
<tr>
<td>014C 7E</td>
<td>ENCODE: LD A, (HL)</td>
<td>GET 1ST DATA WORD</td>
</tr>
<tr>
<td>014D 2F</td>
<td>CPL</td>
<td></td>
</tr>
<tr>
<td>014E A1</td>
<td>AND C</td>
<td>SET 3RD WORD OF PREVIOUS ILLEGAL SEQUENCE TO ZERO</td>
</tr>
<tr>
<td>014F 00</td>
<td>NOP</td>
<td></td>
</tr>
<tr>
<td>0150 00</td>
<td>NOP</td>
<td></td>
</tr>
<tr>
<td>0151 CDE001</td>
<td>CALL REC</td>
<td>GET 1ST DATA WORD</td>
</tr>
<tr>
<td>0154 7E</td>
<td>LD A, (HL)</td>
<td>STORE 1ST DATA WORD</td>
</tr>
<tr>
<td>0155 57</td>
<td>LD D, A</td>
<td></td>
</tr>
<tr>
<td>0156 2C</td>
<td>INC L</td>
<td>GET 2ND DATA WORD FOR USE</td>
</tr>
<tr>
<td>0157 7E</td>
<td>LD A, (HL)</td>
<td>MODIFY APPROPRIATE BITS</td>
</tr>
<tr>
<td>0158 B3</td>
<td>OR E</td>
<td>AND WITH 1ST DATA WORD</td>
</tr>
<tr>
<td>0159 A2</td>
<td>AND D</td>
<td></td>
</tr>
<tr>
<td>015A A1</td>
<td>AND C</td>
<td></td>
</tr>
<tr>
<td>015B 1607</td>
<td>LD D, 07H</td>
<td></td>
</tr>
<tr>
<td>015D 15</td>
<td>WAIT2: DEC D</td>
<td></td>
</tr>
<tr>
<td>015E C25D01</td>
<td>JP NZ, WAIT2</td>
<td></td>
</tr>
<tr>
<td>0161 00</td>
<td>NOP</td>
<td></td>
</tr>
<tr>
<td>0162 00</td>
<td>NOP</td>
<td></td>
</tr>
<tr>
<td>0163 CDE001</td>
<td>CALL REC</td>
<td>GET 2ND DATA WORD FOR OUTPUT</td>
</tr>
<tr>
<td>0166 7E</td>
<td>LD A, (HL)</td>
<td>MODIFY APPROPRIATE BITS</td>
</tr>
<tr>
<td>0167 B3</td>
<td>OR E</td>
<td></td>
</tr>
<tr>
<td>0168 2F</td>
<td>CPL</td>
<td></td>
</tr>
<tr>
<td>0169 A1</td>
<td>AND C</td>
<td></td>
</tr>
</tbody>
</table>
016A 1608
016C 15
016D C26C01
0170 00
0171 00
0172 CD8001
0175 48
0176 2C
0177 D9
0178 78
0179 EE81
017B 47
017C D9
017D C33801
0180 E67E
0182 D9
0183 A8
0184 47
0185 D9
0186 D304
0188 08
0189 00
018A 00
018B 00
018C 08
018D C9
018E 219501
0191 E5
0192 E5
0193 E5
0194 E5
0195 ED4D
0197 ED5E
0199 21F002
019C 7C
019D ED47
019F FD211E01
01A3 FD22F202
01A7 3E02
01A9 85
01AA D303
01AC 3E4F
01AE D303
01B0 3E87
01B2 D303
01B4 3E00
01B6 D306
01B8 3E0F
01BA D306
01BC 3E07 DISIN: LD A, 07H ; DIS INT
01BE D306 OUT (06H), A ; FROM PORT 06
01C0 3E00 INIT2: LD A, 00H ; INT
01C2 D302 OUT (02H), A ; VECTOR
01C4 3E0E LD A, 0FH ; SET PORT 02
01C6 D302 OUT (02H), A ; FOR OUTPUT
01C8 3E07 DISINT: LD A, 07H ; DIS INT
01CA D302 OUT (02H), A ; FROM PORT 02
01CC 3E01 PAS: LD A, 01H ; PASS REG INIT
01CE 4F LD C, A
01CF C9 RET

01D0 0600 REGIS: LD B, 00H
01D2 0EFF LD C, 0FFH
01D4 1600 LD D, 00H
01D6 1E00 LD E, 00H
01D8 210003 LD HL, POINT
01DB D9 EXX
01DC 0600 LD B, 00H
01DE D9 EXX
01DF C9 RET

01E0 3E00 STOP: LD A, 00H ; SET DECK
01E2 D300 OUT (00H), A ; TO STOP
01E4 3E0A DELAY: LD A, DEL
01E6 2600 LD H, 00H
01E8 2E00 LD L, 00H
01EA 24 LOOP: INC H
01EB CAF101 JP Z, DELA
01EE C3EA01 JP LOOP
01F1 2C DELA: INC L
01F2 CAF801 JP Z, DELB
01F5 C3EA01 JP LOOP
01F8 3D DELB: DEC A
01F9 CAF01 JP Z, ON
01FA C3EA01 JP LOOP
01FF ED4D ON: RET1

0300 ORG 0300H
0300 DEFS 00FFH
0400 ORG 0400H

0400 210003 STRT: LD HL, POINT
0403 3E00 LD A, 00H
0405 77 LD (HL), A
0406 2C INC L
0407 3E06 LD A, 06H
0409 77 LD (HL), A
040A 2C INC L
040B 3E0C LD A, 0CH
040D 77 LD (HL), A
040E 2C INC L
040F 3E7E LD A, 7EH
0411 77 LD (HL), A
0412 2C INC L

C12
C13
10 OUT 2,0:OUT 2,15:OUT 2,7
20 OUT 6,0:OUT 6,15:OUT 6,7
30 OUT 7,207:OUT 7,255:OUT 7,7
40 OUT 10,0:OUT 10,15:OUT 10,7
50 CALL "SUBRU"
60 FOR I=0 TO 255
70 PRINT USR(I)
80 NEXT I
90 LET D=1
100 FOR I=0 TO 70
110 LET P=USR(I)
120 IF P>D THEN LET D=P
130 NEXT I
140 GRAPH
150 PLOT 9,9,1
160 LINE 17,9,1
170 PLOT 25,9,1
180 LINE 33,9,1
190 PLOT 41,9,1
200 LINE 48,9,1
210 PLOT 57,9,1
220 LINE 65,9,1
230 PLOT 9,5,"SINGLE"
240 PLOT 25,5,"DOUBLE"
250 PLOT 41,5,"TRIPLE"
260 PLOT 57,5,"QUAD"
270 LET B$=STR$(D)
280 PLOT 2,58:B$+9
290 FOR S=0 TO 48 STEP 16
300 FOR Z=(O+S)TO (7+S)
310 LET P=USR(Z)
320 LET X=Z+9
330 LET Y=P*(50/D)+9
340 PLOT X,9,0
350 LINE X,Y,1
360 NEXT Z
370 NEXT S
380 END
10 OUT 2, 0: OUT 2, 15: OUT 2, 7
20 OUT 6, 0: OUT 6, 15: OUT 6, 7
30 OUT 7, 207: OUT 7, 255: OUT 7, 7
40 CALL "SUBRU"
50 FOR I=0 TO 255
60 PRINT USR(I)
70 NEXT I
80 LET D=1
90 FOR I=0 TO 70
100 LET P=USR(I)
110 IF P>D THEN LET D=P
120 NEXT I
130 GRAPH
140 PLOT 9, 9, 1
150 LINE 79, 9, 1
160 PLOT 76, 5, "70"
170 LET B$=STR$(D)
180 PLOT 2, 58, B$
190 FOR Z=0 TO 70
200 LET P=USR(Z)
210 LET X=Z+9
220 LET Y=P*(50/D)+9
230 PLOT X, 9, 0
240 LINE X, Y, 1
250 NEXT Z
260 END
ORG 0100H
DEL=0AH
TABLE=0300H

; ISSUE

: TO READ DATA
; IN ISS FORM, DECODE AND
; OUTPUT TO DAC

21/6/1985
; OUTPUT PORT P108A 04D,06C
; READ DATA P108B 05D,07C
; WRITE CONTROL & FLAG RESET
; (BIT 7) P107A 00D,02C
; READ FLAGS P107B 01D,03C
; DAC PORT P108A 08D,0AC

0100 F3
0101 CD9701
0104 CDD001
0107 3E05
0109 D300
010B CDD601
010E 3EC5
0110 D300
0112 3E05
0114 D300
0116 DB01
0119 D21601
011C 1F
011E 3EC5
0120 DA7601
0124 031601
0123 DB01
0126 D22301
0129 1F
012A DA7601
012D D805
012F D304
0131 6F
0132 3E35
0134 D300
0136 3E05
0138 D300
013A 7D
013B AC
013C 65
013D 2F

C16
AND B FOR 1ST & 2ND CODEWORD
LD C, A
MONIT3: IN A, (01H) LOOK AT FLAG
RRA
JP NC, MONIT3
RRA
RRA
MONIT3: IN A, (01H)
MONIT3: IN A, (01H)
MONIT3: IN A, (01H)
LD C, A
LD A, 05H
OUT (00H), A
LD A, 05H
OUT (00H), A
LD C, A
LD A, 05H
OUT (00H), A
LD A, 05H
OUT (00H), A
LD A, I
GET INPUT
XOR H DECODE CODEWORD
CPL
LD D, A
AND C
CPL
LD C, A
AND 1ST, 2ND&3RD CODEWORDS
CPL
LD C, A
XNZEROS MASK IN C
EXX SWITCH TO PRESENT DATA
A2 AND C IGNORE 000 AS PRESENT SEQUENCE
LD D, A
GET PNZEROS MASK
CPL
EXX SWITCH TO SUCC' DATA
LD E, D AND WITH COMP'OF PREV'CODEWORD
LD E, D SAVE PREV'CODEWORD LAST BIT (C)
EXX SWITCH TO PRES'DATA
LD E, D OR L FORM 2ND DATA WORD
LD E, D AND 01111110B
RRCA
OUT (08H), A DAC OUTPUT
MONIT1: IN A, (01H) LOOK AT FLAGS
RRA
JP NC, MONIT1 FLAG NOT SET
LD A, 1 GET LAST INPUT
LD H, A PUT LAST INPUT IN "PRES" H
LD A, (05H) INPUT 1ST CODEWORD
OUT (04H), A
LD L, A SAVE INPUT
LD A, 05H
OUT (00H), A
LD A, 05H
OUT (00H), A
LD A, 05H
OUT (00H), A
LD A, 05H
OUT (00H), A
LD A, L GET INPUT
XOR H DATA WORD IN A
LD H, L
CPL
LD B, A INPUT COMP' IN B
EXX SWITCH TO PRESENT DATA
LD A, C GET PNZEROS MASK
AND B
EXX SWITCH TO SUCC' DATA
LD E, D AND 01111110B
RRCA
OUT (08H), A DAC OUTPUT
MONIT2
JP MONIT2
"AND" 01111110B
RRCA
OUT (08H), A DAC OUTPUT
MONIT2
JP MONIT2
0197 3ECF
0199 D303
019B 3E03
019D D303
019F 3E07
01A1 D303
01A3 3E00
01A5 D302
01A7 3E0F
01A9 D302
01AB 3E07
01AD D302
01AF 3E0F
01B1 D306
01B3 3E07
01B5 D306
01B7 3E07
01B9 D307
01BB 3E0F
01BD D307
01BF 3E07
01C1 D307
01C3 3E00
01C5 D30A
01C7 3E0F
01C9 D30A
01CB 3E07
01CD D30A
01CF C9
01D0 3E00
01D2 D300
01D4 E04D
01D6 3E0A
01DB 3E00
01DA 2E00
01DC 24
01D0 CAE301
01DE C3DC01
01E3 2C
01E4 CAEA01
01E7 C3DC01
01EA 3D
01EB CAH101
01EE C3DC01
01F1 C9
0000
END

0108 AGAIN 000A DEL 01E3 DELA 01D6 DELAY 01EA DELB
01A3 INIT2 0197 INIT3 01AF INIT6 01B7 INIT7 01C3 INIT9
01DC LOOP 0170 MONIT1 0123 MONIT2 0140 MONIT3 01F1 ON
0107 READ 01D0 STOP 0116 SYNC 0300 TABLE

NO ERRORS

C18
ORG 0100H
TABLE=02FOH
DEL=0AH

;MILLER ENCODER
;REWIND TAPE AND WRITE
;MILLER ENCODED DATA @ 10KHZ
;FROM CHECKLIST

;14/2/1984

;WRITE PORT P108 04D,06C
;READ PORT P108B 05D,17C
;WRITE CONTROL P107A 00D,02C
;READ CONTROL P107E 01D,03C

;MASK STORAGE:REG D
;OUTPUT:REG C
;TEMP STORE/CELL BOUNDARY IDENTIFIER:REG B
;DATA ADDRESS: (HL)

0100 F3
0101 CD6201
0104 CD6B01

DI
CALL RESET
CALL INITREG *

0107 CDB801
010A 3E08
010C D300
010E FB

REWIND: CALL STOP
LD A, 08H ;SET TRANSPORT
OUT (00H).A ;TO REWIND
EI

010F 00
0110 C30F01

WAIT1: NOP
JP WAIT1 ;WAIT FOR REWIND

0113 CDBB01
0116 0D
0117 CA1C01
011A F3
011B 76

FIN: CALL STOP ;STOP THE TAPE
DEC C ;IF INIT REWIND
JP Z; WRITE
DI
HALT ;WRITE ENDED

011C 3E01
011E D300
0120 CDBF01
0123 3E02
0125 4F
0126 3E07
0128 D300

WRITE: LD A, 01H ;SET
OUT (00H).A ;TO PLAY
CALL DELAY

012A CDB101

REGINIT: CALL REGIS

012D 3E95

TIMER: LD A, 95H
012F D30C OUT (OCH), A : LOAD CONTROL WORD
0131 3E0D LD A, ODH
0133 D30C OUT (OCH), A : LOAD IC
0135 FB E1

0136 7E CELCEN: LD A, (HL) : GET MASK1
0137 2D DEC L
0138 C23D01 JP NZ, CONT
013B 2E20 LD L, 32D : RELOAD POINTER
013D 2F CONT: CPL
013E S7 LD D, A : PUT MASK1 COMP IN REG D
013F 2F CPL
0140 A9 XOR C
0141 4F LD C, A : RECORD OUTPUT
0142 0601 LD B, 01H
0144 CB40 DELAY1: BIT 0, B
0146 C24401 JP NZ, DELAY1
0149 C34C01 JP CELCEN

014C 7E CELLMT: LD A, (HL) : GET NEXT MASK1
014D 2F CPL
014E A2 AND D : PARSTASK FORMED
014F A9 XOR C
0150 4F LD C, A : RECORD OUTPUT
0151 0601 LD B, 01H : RECORD CELLMT
0153 CB40 DELAY2: BIT 0, B
0155 C25301 JP NZ, DELAY2
0158 C35601 JP CELCEN

015B D304 INTRUPT: OUT (04H), A : WRITE CODE TO TAPE
015D 0600 LD B, 00H
015F FB E1
0160 ED4D RETI

0162 216901 RESET: LD HL, PRESET
0165 E5 PUSH HL
0166 E5 PUSH HL
0167 E5 PUSH HL
0168 E5 PUSH HL
0169 ED4D PRESET: RETI

016B ED5E INITREG: IM2
016D 21F002 LD HL, TABLE : BASE VECTOR ADDR
0170 7C LD A, H : HIGH BYTE OF AD
0171 ED47 LD I, A : SET INT REG
0173 FD211301 LD IY, INS : ADDR OF HALT
0177 FD22F202 LD (TABLE+02H), IY : SET VECT
017B 3E02 LD A, 02H
017D 85 ADD A, L : LOW BYTE OF TAB

C20
[The image contains a block of text that appears to be a listing of machine code or assembly language, possibly related to programming or computer science. The text is not legible due to distortion or noise.]

**C21**
017E B303 OUT (03H), A INTO PORT 03H
0180 3E4F LD A, 4FH :SET PORT 03
0182 B303 OUT (03H), A :FOR INP HANDSHK
0184 F0215E01 LD 1Y, INTRPT ADDR OF OUTPUT INS
0188 F022F802 LD (TABLE+08H), 1Y, SET VECT
018C 3E08 LD A, 08H
018E 85 ADD A, L :LOW BYTE OF TABLE
0190 B30C OUT (00H), A
0191 3E87 ENP10: LD A, 87H :ENABLE PIO
0193 B303 OUT (03H), A :INTS
0195 3E00 INIT6: LD A, 00H :INT
0197 B306 OUT (06H), A :VECTOR
0199 3E0F LD A, 0FH :SET PORT 06 FOR
019B D306 OUT (06H), A :OUTPUT HANDSHK
019D 3E07 DISIN: LD A, 07H :DIS INT
019F B306 OUT (06H), A :FROM PORT 06
01A1 3E00 INIT2: LD A, 00H :INT
01A3 B302 OUT (02H), A :VECTOR
01A5 3E0F LD A, 0FH :SET PORT 02H
01A7 D302 OUT (02H), A :FOR OUTPUT
01A9 3E07 DISINT: LD A, 07H :DIS INT
01AB B302 OUT (02H), A :FROM PORT 02
01AD 3E01 PAS: LD A, 01H :PASS REG INIT
01AF 4F LD C, A
01B0 C9 RET
01B1 0601 REGIS: LD B, 01H
01B3 0E00 LD C, 00H
01B5 1600 LD D, 00H
01B7 212003 LD HL, POINT+31D
01BA C9 RET
01BC 3E00 STOP: LD A, 00H :SET DECK
01BD D300 OUT (00H), A :TO STOP
01BF 3E0A DELAY: LD A, DEL
01C1 2600 LD H, 00H
01C3 2E00 LD L, 00H
01C5 24 LOOP: INC H
01C6 CACC01 JP Z, DELA
01C8 C3C501 JP LOOP
01C9 2C DELA: INC L
01CB C3D301 JP Z, DELB
01CD C3C501 JP LOOP
01D0 3D DELB: DEC A
01D2 C3D301 JP Z, ON
01D4 C3C501 JP LOOP
01D7 ED4D ON: RET1
0301 ORG 0301H

0301 00  POINT:  00D
0302 01  01D
0303 02  02D
0304 03  03D
0305 04  04D
0306 05  05D
0307 06  06D
0308 07  07D
0309 08  08D
030A 09  09D
030B 0A  10D
030C 0B  11D
030D 0C  12D
030E 0D  13D
030F 0E  14D
0310 0F  15D
0311 10  16D
0312 11  17D
0313 12  18D
0314 13  19D
0315 14  20D
0316 15  21D
0317 16  22D
0318 17  23D
0319 18  24D
031A 19  25D
031B 1A  26D
031C 1B  27D
031D 1C  28D
031E 1D  29D
031F 1E  30D
0320 1F  31D
0000  END

0136 CELCEN  014C CELMT  013D CONT  000A DEL  01CC DELA
01BF DELAY  0144 DELAY1  0153 DELAY2  01D3 DELB  019D DIS1N
01A9 DISINT  0191 ENP10  0113 FIN  01A1 INIT2  0195 INIT6
016B INITRE  015B INTRUP  0165 LOOP  01DA ON  01AD PAS
0301 POINT  0169 PRESET  012A REGIN1  01B1 REGIS  0162 RESET
0107 REWIND  01BB STOP  02FO TABLE  012D TIMER  010F WAIT1
011C WRITE

NO ERRORS

C23
1 REM CALCULATE AND PLOT SKEW INDICATION
2 INPUT "ENTER PULSE SPACING": S:
10 OUT 2, 0: OUT 2, 15: OUT 2, 7
20 OUT 3, 0: OUT 3, 79: OUT 3, 7
30 OUT 6, 0: OUT 6, 15: OUT 6, 7
40 OUT 7, 0: OUT 7, 79: OUT 7, 7
50 OUT 10, 0: OUT 10, 79: OUT 10, 7
60 OUT 11, 0: OUT 11, 79: OUT 11, 7
70 FOR Z=1000 TO 1 STEP-1
80 OUT 0, 5: REM SET TO READ
90 NEXT Z: REM DELAY
95 LET P=0
96 LET N=0
98 GRAPH
99 PLOT 0, 40, 1
100 LINE 2, 40
101 PLOT 0, 50
102 LINE 2, 50
103 PLOT 0, 59
104 LINE 2, 59
105 PLOT 3, 40, "33"
106 PLOT 3, 50, "66"
107 PLOT 3, 59, "100"
108 PLOT 8, 55, "% OF 100US BIT CELL"
110 PLOT 0, 0, 1
120 LINE 0, 59
130 PLOT 0, 30
140 LINE 159, 30
155 LET X=0
160 LET A=INP(8)
170 LET B=INP(9)
180 LET C=INP(8)
190 LET D=INP(9)
200 IF A<C THEN 160
210 IF B<D THEN 160
215 IF A+B=0 THEN 160
220 LET Y=(A-B)/(A+B)*S
225 IF Y>P THEN LET P=Y
226 IF Y<N THEN LET N=Y
227 LET Y=Y*30/100
230 LET Y=Y+30
250 PLOT X, Y, 2
270 LET X=X+1
280 IF X<79 THEN 160
285 PRINT P
286 PRINT N
287 PRINT P-N
290 GOTO 95
300 GOTO 95
310 END
1 REM CALCULATE AND PLOT SKEW INDICATION
2 INPUT "ENTER PULSE SPACING"; S
4 INPUT "OUTPUT FILE NAME"; F$
10 OUT 2,0: OUT 2, 15: OUT 2, 7
20 OUT 3,0: OUT 3, 79: OUT 3, 7
30 OUT 6,0: OUT 6, 15: OUT 6, 7
40 OUT 7,0: OUT 7, 79: OUT 7, 7
50 OUT 10,0: OUT 10, 79: OUT 10, 7
60 OUT 11,0: OUT 11, 79: OUT 11, 7
70 FOR Z=1000 TO 1 STEP-1
80 OUT 0.5: REM SET TO READ
90 NEXT Z: REM DELAY
92 CREATE £10, F$
155 LET X=0
160 LET A=1NP(8)
170 LET B=1NP(8)
180 LET C=1NP(8)
190 LET D=1NP(8)
200 IF A>C THEN 160
210 IF B>D THEN 160
215 IF A+B=0 THEN 160
220 LET Y=(A-B)/(A+B)*S
230 IF Y>150 THEN 160
240 IF Y<-150 THEN 160
250 PRINT £10, X
260 PRINT £10, Y
262 FOR Z=2500 TO 1 STEP-1
264 OUT 0.5
266 NEXT Z
270 LET X=X+10
280 IF X<1000 THEN 160
285 CLOSE £10
310 END
10 OUT 7, 207: OUT 7, 255: OUT 7, 7
20 OUT 3, 207: OUT 3, 255: OUT 3, 7
30 OUT 6, 0: OUT 6, 15: OUT 6, 7
40 OUT 2, 0: OUT 2, 15: OUT 2, 7
50 OUT 4, 255
60 LET NUMB=0
70 LET SUM=0
80 CALL "SUBRU"
90 GRAPH
100 LET E=40
110 LET M=40
120 FOR N=7 TO 1 STEP -1
130 PLOT 40, (20+5*N), 2
140 NEXT N
150 PLOT 40, 59, 2
160 FOR N=7 TO 1 STEP -1
170 LET A=USK(N)
180 IF A AND 1 THEN GOTO 210
190 LET B=B+A
200 GOTO 220
210 LET B=B-A
220 LINE (B), (20+5*N), 2
230 IF B>B40 GOTO 270
240 IF M>B GOTO 260
250 GOTO 270
260 LET M=B: LET T=(E-N)
270 NEXT N
280 PLOT (40+25), 59, 2
290 LINE (B+25), (20+5*N), 2
300 PLOT (B+25), (20), "+25US", 2
310 PLOT (40-25), 59, 2
320 LINE (B-25), (20+5*N), 2
330 PLOT (B-25), (20), "-25US", 2
340 IF M=40 GOTO 80
350 IF B<40 GOTO 80
360 IF (B=40)<0 GOTO 400
370 LET Z=(B-40)*1*5/35
380 LET DF=(M-40-Z)
390 GOTO 420
400 LET Z=(40-B)*1*5/35
410 LET DF=(M-40+Z)
420 LET NUMB=NUMB+1
430 LET SUM=SUM+DF
440 PRINT "NUMB" "NUMB" DF "DF" AV "SUM/NUMB
450 PRINT ""
460 GOTO 80
1 REM CALCULATE LOSS DUE TO SKEW
2 INPUT "ENTER PULSE SPACING"; S
3 INPUT "TRACK WIDTH"; W
4 INPUT "SIGNAL FREQUENCY"; F
10 OUT 2, 0: OUT 2, 15: OUT 2, 7
20 OUT 3, 0: OUT 3, 15: OUT 3, 7
30 OUT 6, 0: OUT 6, 15: OUT 6, 7
40 OUT 7, 0: OUT 7, 15: OUT 7, 7
50 OUT 10, 0: OUT 10, 15: OUT 10, 7
60 OUT 11, 0: OUT 11, 15: OUT 11, 7
70 FOR Z=1000 TO 1 STEP-1
80 OUT 0, 5: REM SET TO READ
90 NEXT Z: REM DELAY
95 INPUT "ZERO DB REF"; Q
98 LET E=0
100 LET Z=0
102 LET M=INP(5)
104 LET Z=Z+M
106 LET E=E+1
108 IF E<>8 THEN 102
110 LET Z=Z/8
112 LET E=0
114 LET N=0
160 LET A=INP(8)
170 LET B=INP(9)
172 LET M=INP(5)
174 LET N=Q+M
176 LET E=E+1
178 IF E<>8 THEN 172
179 LET N=N/8
180 LET C=INP(8)
190 LET D=INP(9)
200 IF A>C THEN 112
210 IF B>D THEN 112
215 IF A+B=0 THEN 112
220 LET V=(A-B)/(A+B)*8
250 LET G=Y/2.06
260 LET H=G*3.142*W/F/0.0475
265 IF H=0 THEN 160
270 LET J=SIN(H)
280 LET K=J/H
290 LET L=LOG(K)
300 LET G=L*0.4343*20
310 LET P=20*0.4343*LOG(Z/N)
320 PRINT G"DB"; P"DB"
330 GOTO 112
ORG 0100H
TIME=100; SAMPLE PERIOD: 450MICROSEC
PASS=7D; MULTIPLIER1
PASS2=128H; MULTIPLIER2
DEL=0AH
TABLE=01F0H
*L

;SKEWPAT
;REWIND TAPE AND WRITE
;SKEW/VELOCITY CHECK PATTERN
;27/4/1983

;WRITE PORT P108 04D-06C
;READ PORT P108B 05D-07C
;WRITE CONTROL P107A 00D-02C
;READ CONTROL P107B 01D-03C

0100 F3
0101 CD7601
0104 CD7F01
D1
CALL RESET
CALL INITREG

0107 CD8E01
010A 3E08
010C D300
010E E8
RENEW: CALL STOP
LD A, 08H : SET TRANSPORT
OUT (004H), A : TO REWIND
E1

010F 00
0110 C30F01
WAIT: NOP
JP WAIT1 : WAIT FOR REWIND

0113 CD4E01
0116 00
0117 CA1001
011A F3
011B 76
FIN: CALL STOP
DEC C : STOP THE TAPE
 JP Z: WRITE
D1
HALT : WRITE ENDED

011C 3E01
011E D300
0120 CDC201
0123 3E02
0125 4F
0126 3E07
0128 D300
012A E8
WRITE: LD A, 01H : SET
OUT (004H), A : TO PLAY
CALL DELAY
LD A, 02H : SET PASS CNTK
LD C, A : 101H
LD A, 07H : SET DECK
OUT (004H), A : TO WRITE
E1

012B 3E02
012D D304
012F E0801
0132 3E00
0134 3E04
0136 CD5F01
0138 3E06
013B 3E04
013D CD7801
PATN: LD A, 02H
OUT (044H), A : INIT PULSE
CALL CLOK1
LD A, 09H
OUT (044H), A
CALL CLOK2
LD A, 09H
OUT (044H), A : START COUNT PULSE
CALL CLOK1
C28
0140 3E00  LD A, 060H
0142 0304  OUT (04H),A
0144 CD5F01 CALL CLOCK2
0147 3E00  LD A, 060H
0149 D304  OUT (04H),A
STOP COUNT PULSE
014B CD5801 CALL CLOCK1
014E 3E00  LD A, 060H
0150 D304  OUT (04H),A
0152 CD5C01 CALL CLOCK3
0155 3E00  JP PAIN
0158 3E0A  CLOCK1: LD A, TIME
015A 3D    CALL PRE100
015B C25A01 BACK1: JP NZ, BACK1
015E C9    RET
015F 0607  CLOCK2: LD B, PASS
0161 3E0A  MULTIPLIER
0163 3D    LOOP: LD A, TIME
0164 C26301 CLOCK PERIOD
0167 05    BACK2: JP NZ, BACK2
0168 C26101 DEC B
016B C9    JP NZ, LOOP
016C 0800  RET
016E CD5F01 CLOCK3: LD C, PASS
0171 0D    MULTIPLIER
0172 C26E01 LOOPY: CALL CLOCK2
0175 C9    DEC C
0176 217D01 JP NZ, LOOPY
0179 E5    RET
017A E5    RESET: LD HL, PRESET
017B E5    PUSH HL
017C E5    PUSH HL
PUSH HL
017D EB4D  PRESET: RET1
017F ED5E  INITREG: IM2
0181 21F001 LD HL, TABLE
0184 7C    BASE VECT ADDR
0185 ED47  LD A, H
0187 FD21501 HIGH BYTE OF AD
018B FD22F01 LD B: LD HL, A
018F 3E02  SET INT REG
0191 85    LD (TABLE+00H), B
0192 U303   SET Vect
0194 3E48  ADD A, L
0195 U303   OUT (03H), A
0196 U303   INTO PORT 03H
0198 3E49  LD A, 0FH
0199 U303   SET PORT 03
019A 3E49  OUT (03H), A
019B U303   SHOW INT HANDSHK
019C U303  ENP10: LD A, 87H
019E U306   ENABLE P10
01A0 3E4F  OUT (03H), A
01A1 3E4F  PINT
01A3 3E4F  INTH
01A4 3E07  INT
01A5 3E1A  INT
0148 3E00 INIT2: LD A, OOH SET REG INIT
014A 3E02 OUT (0A0H): A WRITE INTO PORT 0A
014C 3E04 LD A, OOH READ PORT 00H
0150 3E06 OUT (020H): A WRITE OUTPUT
0152 3E07 DISINT: LD A, 07H IF INT ENABLE
0154 3E02 OUT (02H): A READ FROM PORT 02
0156 3E01 PAS: LD A, 01H PASS REG INIT
0158 4F LD C, A
015D C9 RET
0161 0D DEC C IF INIT READ
0165 CA1C01 JP Z, WRITE IF JP TO WRITE
016F F3 DI ENABLEABLE INT
0170 76 HALT WRITE ENDED

01BE 3E00 STOP: LD A, 00H SET DECK
01C0 B300 OUT (000H): A WRITE STOP
01C2 3E0A DELAY: LD A, DEL
01C4 2600 LD H, 00H
01C6 2E00 LD L, 00H
01C8 24 LOOP: INC H
01C9 CACF01 JP Z, DELA
01CC C3801 JP LOOP
01CF 2C DELA: INC L
01D0 CAC601 JP Z, DELB
01D3 C3801 JP LOOP
01D6 3D DELB: DEC A
01D7 CADD01 JP Z, ON
01DA C3801 JP LOOP
01DD ED4D ON: RETI
0000 END

015A BACK1 0163 BACK2 0158 CLOCK1 015F CLOCK2 0160 CLOCKS
000A DEL 01CF DELA 0102 DELAY 0106 DELB 0144 DISINT
01E0 DISINT 0192 ENF10 0113 FIN 01A8 INIT2 019C INIT6
017F INIT1 01C8 LOOP 0161 LOOPS 016E LOOPY 011D ON
0184 PASS 0007 PASS 0080 PASS2 0128 PATM 017D PRESET
017E RESEI 0107 REWIND 01BE STOP 011D TABLE 00DA TIME
010F WAIT 011C WRITE

NO ERRORS
;MENU
;PROGRAM TO WRITE IN
;MANCHESTER CODE FROM
;DATA LIST: 2KHZ DATA
;RATE.
;CTC USED TO DETERMINE
;DATA RATE

;20/11/86

;WRITE PORT P108A: 04D, 06C
;READ PORT P108B: 05D, 07C
;WRITE CONTROL P107A: 00D, 02C
;READ CONTROL P107B: 01D, 03C
;CTC=0CH

0000 C =
0200 =
0400 =
01FO =
000A =
CTC=0CH
LIST=0200H
TIM=64D
TABLE=01FOH
DEL=0AH

0100
ORG 0100H

0100 FB
0101 CD2001
0104 3E00
0106 210002
0109 77
0108 3C
0108 2C
010C C20001
010E 3E01
0111 D500
0113 CD8401
0116 FB
0117 3E40
0118 U20C
0118 210002
011E 0601
0120 2E
0121 2F
0122 CD40
0124 C22001
0127 2F
0128 0601
012B CD40
012C C22001
012F 2C
0130 0601
0132 C22001

PLAY:
LD A,01H
OUT (00H);A
TO PLAY
CALL DELAY

0116 FB
0117 3E40
0118 U20C
0118 210002
011E 0601
0120 2E
0121 2F
0122 CD40
0124 C22001
0127 2F
0128 0601
012B CD40
012C C22001
012F 2C
0130 0601
0132 C22001

REC:
LD A,(HL)
CALL DELAY
CPL
;COMPLEMENT OF DATA

WAIT1:
BIT 0; B
JP N2, WAIT1
CPL
;TRUE VALUE OF DATA

WAIT2:
BIT 0; B
JP N2, WAIT2
INC L
LD B, 01H
JP REC

C31
; PROGRAM TO READ PRBS DATA
; IN MANCHESTER CODE AND
; RECORD ERROR BURSTS
; SYNCH TO TRACK WITH LEADING CHANGING
; 4 TRACK
; APPROX. 2KHZ
; CTC2 USED TO MEASURE RATIOS
; 27/2/1987

; WRITE PORT P109A: 040, 06C
; READ PORT P108B: 05D, 07C
; WRITE CONTROL P107B: 00D, 02C
; READ CONTROL P107B: 01D, 03C
; LED DISPLAY P109A: 08D, 10C

; CTC1: 0C, CTC2: 0D

0017 = CH=00010111D
000C = CTC1=0C1H
000D = CTC2=0DH
0008 = LDIS=08H
000A = DEL=0AH

0112 = OUTMY=112H
0118 = DERV=118H
0106 = USR=106H
0108 = USRP=108H
010C = USRVT=10CH
011A = SUBPTR=11AH

0106 = ORG USR:
0106 02406B = JP USRM

0118 = ORG DEVRM
0118 016D = DERW BASEMD

011A = ORG SUBPR
011A 0069 = DERW CLINK

6900 = ORG 6900H

690C 0000 = CLINK: DERW 0
6902 05 = DERW 5
6902 5564252 = DERW SUBPR
6908 0647 =
EDGE2: SRL D
       LD L.D
       LD E: 00H
       JL D
       LD D:
       LD D.L
       JP ROLDFR
       JR RE-SYNC ON NEW TRACK

FUTHA1: BIT 7,E
       JP CHANG
       DEC L
       INC (HL)
       JR 2: FIN
       INC ERRORS LOGGED

FUTHA2: BIT 7,E
       JP CHANG2
       DEC L
       INC (HL)
       JR 2: FIN
       INC ERRORS LOGGED

FIN: LD A: 00H
     OUT (00H),A
     RET

EXIT: EY AF,AF
       BIT 6,E
       ALT REGISTERS
       JP NZ:NOSWICH
       LD E: 11110000B
       BIT 5 EXY SWICH.BIT 4 EXIT

NOSWICH: DI
       BIT 5,E
       ALT REG?
       JP Z: SWICHNO
       LD E: 11010000B
       EXY
       EY AF,AF
       EI
       RET1

SWICHNO: LD E: 11010000B
       EY AF,AF
       EI
       RET1

USRNS: CALL USRVF
       LD HL,QUE
       ADD HL,DE
       LD B: (HL)
       LD A: 00H
       CALL USRV1
       RET
TABLE:

ORG 6000H

BUFF:

ORG 6000H

BUFEND:

ORG 6000H

DEFB 0

BASEND:

END

6801 AGAIN 684F BACK 6D01 BASEND 0118 BBUFV 6906 BNDARY
6D00 BUFEND 6C00 BUFF 6A5D CHANGE1 6A53 CHANGE2 6900 CLINK
000C CTC1 0000 CTC2 0017 CW 000A DEL 6818 DELA
690E DELAY 6922 DELB 6AED EDGE2 6A53 ERROR1 6AAA ERROR2
68E2 EVENT 6889 EVENPS 6A3F EVMP 6D24 EXIT 681F FIN
690D PRI2:1 6A8C FIRST 6A32 FIRST1 6884 FORT 6988 FOUR
6A18 FOUR4 6A5D FUTHA1 6B0E FUTHA2 0008 LEDS 6914 LOOP
6A55 MIDL2 6A0A MIN2 6B2D MOSW1C 6850 ODDP 6A97 ODDS
6A3D ODP 6929 ON 69EA ONSCR1 6A5F ONSCR1 6AB3 ONSCR2
697E ONE 6A0E ONE1 0112 OUTMY 6A0D PLUS2 695F POL1
69BD POL2 6A4A POL3 6A1A POLE1 6A74 POLE2 69C8 POLF1K
6A1D POL11 6A77 POLL2 68BD POSPD 6A4A POS1 6AA1 POSMD
6858 START 011A SUBPTR 690A SUBRT 163A SWICHN 654E TABLE
6879 THREE 6A08 THREE3 6983 TWO 6A13 TWO2 0106 USR
6B40 USRM 0109 USRUP 010C USRT

NO ERRORS
ERRMANTRCK

PROGRAM TO READ DATA
IN MANCHESTER CODE,
CORRECT ERRORS AND
CATALOGUE TRACK ERRORS
(MANTRCK WITH ERROR CORRECTION)
APPROX 1KHZ

CTC2 USED TO MEASURE RATIOS

8/1/1987

WRITE PORT P108A: 04D, 06C
READ PORT P108B: 05D, 07C
WRITE CONTROL P107A: 00D, 02C
READ CONTROL P107B: 01D, 03C
LED DISPLAY P109A: 08D, 10C

CTC1: OC, CTC2: OD

0037 =
000C =
000D =
0008 =
000A =
0008 =

0112 =
0118 =
0106 =
0109 =
010C =
011A =

0106 =
0106 C3606C

0118 =
0118 0170

011A =
011A 0069

6900 =

6900 0000
6902 05
6903 53554252
6908 0A69

0112 =
0118 =
0106 =
0109 =
010C =
011A =

0106 =
0106 C3606C

0118 =
0118 0170

011A =
011A 0069

6900 =

6900 0000
6902 05
6903 53554252
6908 0A69

ORG USR
JP USRM

ORG BBUFF
DEFW BASEND

ORG SUBPTR
DEFW CLINK

ORG 6900H

CLINK:
DEFW 0
DEFB 5
DEFM "SUBRT"
SUBRT: LD A, 05H ; SET DECK TO PLAY
OUT (00H), A

DELAY: LD A, DEL
LD H, 00H
LD L, 00H

LOOP: INC H
JP Z, DELA
JP LOOP

DELA: INC L
JP Z, DELB
JP LOOP

DELB: DEC A
JP Z, ON
JP LOOP

ON: IM2
LD HL, TABLE
LD A, H
LD I, A
LD IY, EXIT
LD (TABLE+02H), IY
LD A, 02H
ADD A, L
OUT (03H), A
OUT (03H), A
OUT (03H), A
EI

EXX: LD E, 80H
LD HL, BUFF1
LD A, 00H

BACK1: LD (HL), A
INC L
JP NZ, BACK1
RESET BUFFER
LD HL, BUFF2

BACK2: LD (HL), A
INC L
SET DISP BUFFER TO ZERO
JP NZ, BACK2

C42
INITCTC: LD A, CW
OUT (CTC2), A
IN A, (05H)
LD B, A
AND 08H
JP Z, POL1
LD A, OFFH
OUT (CTC2), A
1ST DATA CHANGE?

POL1: IN A, (05H)
XOR B
AND 08H
JP Z, POL2
NO DATA CHANGE
LD A, OFFH
1ST DATA CHANGE
OUT (CTC2), A
START RATIO TIMER

POL2: IN A, (05H)
XOR B
AND 08H
JP Z, POL3
GET PERIOD VALUE N-C1
LD C, A
SAVE PERIOD VALUE N-C1
LD A, CW
STOP PERIOD COUNTER
OUT (CTC2), A
START TIMING

POL3: IN A, (05H)
XOR B
AND 08H
JP Z, POL4
GET PERIOD VALUE N-C2
LD D, A
SAVE PERIOD VALUE N-C2
LD A, CW
STOP PERIOD COUNTER
OUT (CTC2), A
START TIMING

AGAIN: IN A, (05H)
LD B, A

WAIT FOR 3RD DATA
XOR B
CHANGE

GET PERIOD VALUE N-C3
LD C, A
SAVE PERIOD VALUE N-C3
LD A, CW
STOP PERIOD COUNTER
OUT (CTC2), A
START TIMING

SUB C
TRUE VALUE OF 1ST PERIOD
LD C, A
SAVE TRUE VALUE OF 1ST PERIOD
OUT (CTC2), A
2ND PERIOD
SUB D
TRUE VALUE OF 2ND PERIOD
LD D, A
SAVE IN D
LD A, C
1ST PERIOD-2ND PERIOD
SUB D
1ST PERIOD-2ND PERIOD+1ST

JP M, NEGPD
EITHER 1:2 OR 1:1
POSPD: SRL D
EITHER 2:1 OR 1:1
SUB D
1ST PERIOD-2ND PERIOD+2ND
JP P, BNDARY
2:1 FOUND
RL D
1:1 TRY AGAIN
LD C, D
AFTER MAKING 2ND PERIOD
LD E, OFFH
THE 1ST PERIOD
JP AGAIN
BNDARY: LD L, D
SAVE 1/4 BIT PERIOD
LD H, L
RL D
LD C, D
LD E, 00H
JP SAMP
SAMPLE 1 FLAG
NEGPD: SRL C
EITHER 1:2 OR 1:1
ADD C
1ST PERIOD-2ND PERIOD+1ST
JP M, CENTRE
1:2 RATIO, CENTRE FOUND
LD C, D
1:1 RATIO, TRY AGAIN
JP AGAIN
AFTER MAKING 2ND PERIOD TH
CENTRE: LD L, C
SAVE 1/4 BIT PERIOD
LD H, L
LD C, D
LD E, 01H
SAMPLE 1 FLAG

C43
SAMP: IN A. (O3H)  
LD B: A  
SAMPL: IN A. (CTC2)  
NEG  
ADD OFFH  
CP L  
GET PERIOD COUNTER  
TRUE VALUE  
SAMPLE POINT?  
NO  
UPDATE NEXT  
COMPARISON POINT  
BIT O, E  
1ST OR 2ND HALF OF BIT CELL  
JP Z, FIRST  
SECOND:  
LD E, OOH  
EXX  
LD A: L  
CP 18H  
1ST TIME THRO'G?  
JP Z, YES  
BUFFER FULL?  
CP OEH  
JP Z, BFULL  
CONTINUE  
IN A. (O5H)  
LD (HL): A  
PLACE DATA IN  
INC L  
BUFF 3  
GNI:  
EXX  
JP ONCMR  
YES:  
IN A. (O5H)  
CP OOH  
IS IT ALL ZEROS?  
JP NZ, GNI  
NO, GET NEXT INPUT  
LD (HL): A  
YES, STORE DATA  
INC L  
AND CONTINUE  
EXX  
JP ONCMR  
FIRST:  
LD E, O1H  
ONCMR: IN A. (O5H)  
GET INPUT  
XOR B  
DATA CHANGE?  
AND OSH  
NO  
JP Z, SAMPL  
GET PERIOD VALUE N-C2  
LD D: A  
SAVE PERIOD VALUE N-C2  
IN A. (CTC2)  
LD A: CW  
STOP PERIOD COUNTER  
OUT (CTC2): A  
COUNTER  
LD A, OFFH  
START TIMING PERIOD AGAIN  
OUT (CTC2): A  
START TIMING PERIOD AGAIN  
SUB D  
TRUE VALUE OF 2ND PERIOD  
SUB D  
SAVE IN D  
LD D: A  
LD A: C  
1ST PERIOD-2ND PERIOD  
SUB D  
1ST PERIOD-2ND PERIOD  
JP M, MINPD  
EITHER 1: 2 OR 1:1  
PLUSPD: SRL D  
EITHER 2: 1 OR 1:1  
SUB D  
1ST PERIOD-2ND PERIOD-2ND F  
JP P, EDGE  
2:1 FOUND  
LD L: H  
RL D  
1:1, TRY ONCE MORE  
LD C: D  
AFTER MAKING THE 2ND  
JP SAMP  
AFTER MAKING 2ND
MINPD: SRL C
ADD C
JP M, MIDL
LD L, H
LD C, D
JP SAMP

MIDL: LD L, C
LD H, L
LD E, 01H
LD C, D
JP SAMP

EDGE: LD L, D
LD H, L
LD E, 00H
RL D
LD C, D
JP SAMP

BFULL: LD HL, BUFF3
JP ALONG

REPASS: INC L
JP Z, STERR

ALONG: LD A, LENGTH
NEG
LD D, A
TRAK COUNTER

PAR1POS: CALL PCHECK
CHECK PARITY
BIT O, A
1ST ERROR?
INC L
JP Z, STERR
INC L
JP PAR1POS

ERR2POS: INC D
TRACK COUNTER

NDONE: INC L
JP Z, STERR
CALL PCHECK
SLA B
SHIFT +ERROR COUNTER
BIT O, A
ERROR?
JP Z, ERR2POS

ERRPOS: SET O, B
YES, SHIFIT IN 1
ERRPOS: BIT 7, B
8 CONSECUTIVE SHIFTS?
JP NZ, PAR1NEG
YES
INC D
NO
JP NDONE

PAR1NEG: INC D
INC L
JP Z, STERR
CALL PCHECK
BIT O, A
ERROR?
JP Z, PAR1NEG
NO
INC D
FINAL INC
ERRNEG: LD C, 01H : YES, SHIFT 1 INTO -ERROR COUNTER
NEND: INC L
JP Z, STERR
CALL PCHECK
SLA C : SHIFT -ERROR COUNTER
BIT 0, A : ERROR?
JP Z, ERR8NEG : NO
JP Z, ERR8NEG : NO
ERR8NEG: SET O, C : YES, SHIFT IN 1
BIT 7, C : 8 CONSECUTIVE SHIFTS?
JP Z, NEND : NO
ECHK: LD A, B : YES
CP C : +&- ERRORS THE SAME?
JP NZ, REPASS : NO, ABORT

EXX
LD E, 08H
EXX
LD E, 01H : SPATIAL FORMAT REG
SRL D : DIVIDE BY 2
LD A, D
EXX
LD B, A : SAVE TRACK COUNTER
EXX
DEC D : CONVERT TRACK
JP Z, ENDCON : TO SPATIAL FORMAT
SLA C
JP CONV
ENDCON: EXX
LD A, B
EXX
LD D, A : GET TRACK COUNTER
LD A, L
ADD 01H : HL POINTS TO
SUB D : 1ST LOCATION IN ERROR
LD L, A
BITCOR: RLC B : EMPTY ERROR COUNTER
CALL C, ERA : ERROR, GO CORRECT
INC L
EXX
DEC E
EXX
JP Z, ENDCOR
JP BITCOR : REPEAT
ERA: LD A, (HL) : GET ERROR BYTE
XOR C : CORRECT ERROR
LD (HL), A : RE-STORE CORRECTED BYTE
RET

ENDCOR: LD A, L
SUB 08H
ADD D : REPOSITION HL POINTER
LD L, A
JP REPASS
CALL NZ, PRTY
INC L
LD A, (HL)
BIT 6, A
CALL NZ, PRTY
INC L
LD A, (HL)
BIT 7, A
CALL NZ, PRTY
LD A, L
LD L, A
LD A, (HL)
SUB 10H
SUB LENGTH-1
LD L, A
EXX
LD A, B
GET PARITY STATUS
EXX
RET
LD HL, BUFF3+18H
LD B, L
SAVE ADDRESS
LD A, (HL)
GET DATA BYTE
CP 00H
ALL ZEROS?
JP Z, OK
CP OFFH
ALL ZEROS
JP Z, OK
LD HL, BUFF1
LD L, A
LOG THE ERROR
INC (HL)
JP Z, OVER
INC B
JP Z, ND
END OF PROCESS
LD HL, BUFF3
LD L, B
NEXT LOCATION IN HL
LD 100H
DEC (HL)
DEC (HL)
JP ERST
CONTINUE
DEC (HL)
JP CAT
OVERFLOW
BIT 7, E
END OF RUN?
JP Z, CAT
YES
LD HL, BUFF3+18H
EXX
NO
LD (100H
DEC (HL)
JP INITCTC
CONTINUE
LD HL, BUFF1
INC L
LD D, 09H
BIT POSITION COUNTER
LD C, 00H
RESET COUNTER
DEC D
REPEAT:
DEC D
JP Z, FORW
ALL BITS CHECKED
RLC L
JP INC, REPEAT
INC C
JP REPEAT
18 COUNTER
REPEAT UNTIL ALL
C48
FORW: LD A, (HL) : BIT POSITIONS CHECKED
LD E, A : NO. OF 1'S TRACK ERRORS
LD A, L : SAVE IN TEMP STORE
CPL
LD L, A : SET NO. OF 0'S TRACK ERRORS
LD D, A : SAVE IN TEMP STORE
ADD D : TOTAL ERRORS
LD B, L : SET ASIDE L
LD A, E : SET ASIDE TOTAL ERRORS
ADD D : TOO MANY ERRORS TO LOG
JP STPOST
LD L, 00H : SET BASE ADDRESS
CALL TRKID
LD L, B
LD L, 10H : SET BASE ADDRESS
CALL TRKID
LD L, B
LD L, 20H : SET BASE ADDRESS
CALL TRKID
LD L, B
LD L, 30H : SET BASE ADDRESS
CALL TRKID
LD L, B
INC H : BUFFER DISPLAY ADDRESS
BIT 0, B
CALL NZ, STOR
INC L
INC L
BIT 1, B
CALL NZ, STOR
INC L
INC L
BIT 2, B
CALL NZ, STOR
INC L
BIT 3, B
CALL NZ, STOR

EXIT: BIT 7, E ; REGISTER E?

CHANGE: EXX
LD E, 40H
EXX
EI
NOCHGE: LD E, 40H
EI
RETI

DISP: LD A, 00H
OUT (60H), A ; STOP THE DECK
RET ; RETURN TO BASIC

USRH: CALL USRHF
LD HL, BUFF2
ADD HL, DE
LD E, (HL)
LD A, 00H
CALL USRVI
RET

ORG 6C80H
TABLE:

ORG 6D00H
BUFF1:
ORG 6E00H
BUFF2:
ORG 6F00H

C50
BUFF3:  ORG 2000H

BUFFEND:

BUFFEND:

BASENU:

BASE:

NO ERRORS
ORG 0100H
DEL=0AH
TABLE=0300H

;ISSDEC

; TO READ DATA
; IN ISS FORM, DECODE AND
; OUTPUT

; 20/6/1985
; OUTPUT PORT P108A 04D, 06C
; READ DATA P108B 05D, 07C
; WRITE CONTROL & FLAG RESET
; (BIT 7) P107A 00D, 03C
; READ FLAGS P107B 01D, 03C

0100 F3
0101 CD7401
0104 CDA101

0107 3E05 READ: LD A, 05H SET TO
0109 D300 OUT (00H), A READ
010B CDA701 AGAIN: CALL DELAY

010E C35301 JP MONIT1

0111 DB01 MONIT2: IN A, (01H) : LOOK AT FLAG
0113 1F RRA
0114 D21101 JP NC, MONIT2 : FLAG NOT SET
0117 1F RRA
0118 DA5901 JP C, MONIT1+6 : START OF 3-BIT GROUP
011B DB05 IN A, (05H) : INPUT 2ND CODEWORD
011D 6F LD L, A : SAVE INPUT
011E AC XOR H : DECODE CODEWORD
011F 65 LD H, L
0120 2F CPL
0121 A0 AND B : FOR 1ST & 2ND CODEWORD
0122 4F LD C, A
0123 3E85 LD A, 85H
0125 D300 OUT (00H), A
0127 3E05 LD A, 05H : RESET FLAGS
0129 D300 OUT (00H), A
012B DB01 MONIT3: IN A, (01H) : LOOK AT FLAG
012D 1F RRA
012E D22B01 JP NC, MONIT3
0131 1F RRA
0132 DA5901 JP C, MONIT1+6 : START OF 3-BIT GROUP
0135 DB05 IN A, (05H) : INPUT 3RD CODEWORD
0137 ED47 LD I, A : SAVE INPUT
0139 AC XOR H : DECODE CODEWORD
013A 2F CPL

C52
013B 57  LD D, A
013C A1  AND C
013D 2F  CPL
013E 4F  LD C, A
013F D9  EXX
0140 A2  AND D
0141 6F  LD L, A
0142 79  LD A, C
0143 2F  CPL
0144 D9  EXX
0145 A9  AND E
0146 5A  LD E, D
0147 D9  EXX
0148 B5  OR L
0149 U804  OUT (04H), A
014B 3E85  LD A, 85H
014D D300  OUT (00H), A
014F 3E05  LD A, 05H
0151 D300  OUT (00H), A
0153 DB01  MONIT1: IN A, (01H)
0155 1F  RRA
0156 U25301  JP NC, MONIT1
0159 ED57  LD A, 1
015B 67  OUT (04H), A
015C DB05  IN A, (05H)
015E 6F  LD L, A
015F AC  XOR H
0160 65  LD H, L
0161 2F  CPL
0162 47  LD B, A
0163 U9  EXX
0164 79  LD A, C
0165 A0  AND B
0166 U9  EXX
0167 D304  OUT (04H), A
0169 3E85  LD A, 85H
016B D300  OUT (00H), A
016D 3E05  LD A, 05H
016F D300  OUT (00H), A
0171 C31101  JP MONIT2
0174 3ECE  IN1T3: LD A, 0CFH
0176 U303  OUT (03H), A
0178 3E03  LD A, 03H
017A U303  OUT (03H), A
017C 3E07  LD A, 07H
017E D303  OUT (03H), A
0180 3E00  IN1T2: LD A, 00H
0182 D302  OUT (02H), A
0184 3E0F  LD A, 0FH
0186 D302  OUT (02H), A
0188 3E07  LD A, 07H
018A D302  OUT (02H), A

C53
018C 3E0F INIT6: LD A, 0FH
018E 3D06 OUT (06H), A
0190 3E07 LD A, 07H
0192 3D06 OUT (06H), A
0194 3ECF INIT7: LD A, 11001111B ;BITMODE
0196 3D07 OUT (07H), A
0198 3E0F LD A, 0FFH
019A 3D07 OUT (07H), A
019C 3E07 LD A, 00000111B
019E 3D07 OUT (07H), A
01A0 C9 RET
01A1 3E00 STOP: LD A, 00H
01A3 D300 OUT (00H), A
01A5 ED4D RETI
01A7 3E0A DELAY: LD A, DEL
01A9 2600 LD H, 00H
01AB 2E00 LD L, 00H
01AD 24 LOOP: INC H
01AE CAB401 JP Z, DELA
01B1 C3AD01 JP LOOP
01B4 2C DELA: INC L
01B5 CABB01 JP Z, DELB
01B8 C3AD01 JP LOOP
01BB 3D DELB: DEC A
01BC CAC201 JP Z, ON
01BF C3AD01 JP LOOP
01C2 C9 ON: RET

END

01DB AGAIN 000A DEL 01B4 DELA 01A7 DELAY 01B8 DELB
0180 INIT2 0174 INIT3 018C INIT6 0194 INIT7 01AD LOOP
0153 MONIT1 0111 MONIT2 012B MONIT3 01C2 ON 0107 READ
01A1 STOP 0300 TABLE

NO ERRORS

C54
ORG 0100H
TABLE=02FOH
DEL=0AH
POINT=0300H
*L

;ISS2/3 ENCODER
;RECEIVE TAPE AND WRITE
;ISS2/3 ENCODED DATA @ 10KBS
;FROM CHECKLIST

;26/2/1985

;WRITE PORT P108 04D,06C
;READ PORT P108B 05D,17C
;WRITE CONTROL P107A 00D,02C
;READ CONTROL P107B 01D,03C

;REG B HOLDS NEXT MASK
;REG C HOLDS CURRENT MASK
;REG D IS TEMPORARY STORE
;REG E HOLDS MODIFY MASK
;REG B' HOLDS OLD OUTPUT
;DATA ADDRESS: (HL)

0100 FB
0101 CBA301
DI
CALL RESET

0104 210003
;LIST:  
0107 3E3C
0109 77
010A 2C
010B 3E24
010D 77
010E 2C
010F 3E18
0111 77
0112 2C
0113 3EC3
0115 77
0116 2C
0117 3EFF
0119 77
011A 2C
011B 3E24
011D 77
011E 2C
011F 3EDB
0121 77
0122 2C
0123 3E00
0125 77
0126 2C
0127 C20501
JP NZ,LIST+01H
CALL INITREG

REWIND: CALL STOP
LD A, 08H ; SET TRANSPORT
OUT (00H), A ; TO REWIND
EI

WAIT: NOP
JP WAIT ; WAIT FOR REWIND

FIN: CALL STOP ; STOP THE TAPE
DEC C ; IF INIT REWIND
JP Z, WRITE
DI
HALT ; WRITE ENDED

WRITE: LD A, 01H ; SET
OUT (00H), A ; TO PLAY
CALL DELAY
LD A, 02H ; SET PASS CNTR
OUT (00H), A
EI

REGINIT: CALL REGIS

TIMER: LD A, 95H ; LOAD CONTROL WORD
OUT (00H), A
LD A, 12H
OUT (00H), A
EI

FORMASK: INC L
LD A, (HL) ; GET 2ND WORD
LD D, A ; STORE
INC L
LD A, (HL) ; GET 3RD DATA WORD
OR U
LD B, A ; SAVE NEXTMASK, ILLEGAL
LD A, C ; GET CURRT ILLEG SEQ
CPL
OR B ; TERMIN CURR ILLEG SEQ
LD B, A ; SAVE NEXT MASK ILLEG SEQUENCES IDENTIFIED BY ZERO

INC L
LD A, (HL) ; GET 4TH WORD
CPL ; LAST BIT IDENT BY ZEROS
LD A, (HL) ; GET 3RD DATA WORD
OR B ; AND ZEROS
CPL ; MODIFY MASK, ZEROS
LD E, A ; SAVE MODIFY MASK
DEC L
DEC L
DEC L
DEC L

ENCODE: LD A, (HL) ; GET 1ST DATA WORD
CPL ; SET 3RD WORD OF PREVIOUS
AND C
LD D, 00H ; ILLEGAL SEQUENCE TO ZERO
0175 CB42   WAIT1: BIT 0, D
0177 CA7501  JP Z, WAIT1
017A 7E   LD A, (HL) ; GET 1ST DATA WORD
017B 57   LD D, A ; STORE 1ST DATA WORD
017C 2C   INC L
017D 7E   LD A, (HL) ; GET 2ND DATA WORD FOR USE
017E B3   OR E ; MODIFY APPROPRIATE BITS
017F A2   AND D ; AND WITH 1ST DATA WORD
0180 A1
0181 1600
0183 CB42   WAIT2: BIT 0, D
0185 CA8301  JP Z, WAIT2
0188 7E   LD A, (HL) ; GET 2ND DATA WORD FOR OUTPUT
0189 B3   OR E ; MODIFY APPROPRIATE BITS
018A 2F   CPL
018B A1
018C 1600
018E CB42   WAIT3: BIT 0, D
0190 CA8E01  JP Z, WAIT3
0193 48   LD C, B ; NEXT MASK
0194 2C   INC L
0195 CE35C01  JP FORMASK
0198 D9
0199 A8   XOR B ; XOR WITH OLD OUTPUT
019A 47   LD B, A ; STORE-BECOMES OLD OUTPUT
019B D9   EXX
019C DD04   OUT (04H), A
019E 1601   LD D, 01H
01A0 FB   EI
01A1 ED4D   RET1
01A3 21AA01
01A6 E5   PUSH HL
01A7 E5   PUSH HL
01A8 E5   PUSH HL
01A9 E5   PUSH HL
01AA ED4D   PRESET: RETI
01AC ED5E
01AE 21FO02
01B1 7C   LD A, H ; HIGH BYTE OF AD
01B2 ED47   LD L, A ; SET INT REG
01B4 FD213901   LD IY, FIN ; ADDR OF HALT
01B8 FD22F202   LD (TABLE+02H), IY ; SET VECT
01BC DE02   LD A, 02H
01BE 85   ADD A, L ; LOW BYTE OF TAB
01BF D303   OUT (03H), A ; INTO PORT 03H
01C1 3E4F   LD A, 4FH ; SET PORT 03
01C3 D303   OUT (03H), A ; FOR INP HANDSHK
01C5 FD219801   LD IY, INTRUP ; ADDR OF OUTPUT INS
01C9 FD22F802   LD (TABLE+05H), IY ; SET VECT
01CD DE08   LD A, 08H
01CE 85   ADD A, L ; LOW BYTE OF TABLE
01D0 D30C   OUT (0CH), A
01D2 3E87   ENPIO: LD A, 87H ; ENABLE P10
01D4 D303   OUT (03H), A ; INTS
01D6 3E00   INIT6: LD A, 00H ; INT
01D8 D306   OUT (06H), A ; VECTOR
01DA D30F   LD A, 0FH ; SET PORT 06 FOR
01DC D306   OUT (06H), A ; OUTPUT HANDSHK
01DE 3E07   DISIN: LD A, 07H ; DIS INT
01E0 D306   OUT (06H), A ; FROM PORT 06

C57
INIT:  LD A, 00H ; SET INT
        OUT (02H), A ; VECTOR
DISINT: LD A, 07H ; DIS INT
         OUT (02H), A ; FROM PORT 02
PAS:  LD A, 01H ; PASS REG INIT
       LD C, A
REGIS: LD B, 00H
       LD C, OFFH
       LD D, 00H
       LD E, 00H
       LD HL, POINT
       EXX
       LD B, 00H
       EXX
       LD B, 00H ; SET DECK
       OUT (00H), A ; TO STOP
STOP:  LD A, 00H
       OUT (02H), A
DELAY: LD A, DEL
       LD H, 00H
       LD L, 00H
       INC H
       JP Z, DELA
       JP LOOP
DELA:  INC L
       JP Z, DELB
       JP LOOP
       DEC A
       JP Z, ON
       JP LOOP
       ON: RETI
REGINI: DEF 00FFH
REGIS:  DEF 0000H
END
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSO</td>
<td>Fetch Save Offset</td>
</tr>
<tr>
<td>RA</td>
<td>Read Accumulator</td>
</tr>
<tr>
<td>SBO</td>
<td>Store to Buffer Offset</td>
</tr>
<tr>
<td>MTGT</td>
<td>Move to General Register</td>
</tr>
<tr>
<td>CLI</td>
<td>Clear Interrupt Flag</td>
</tr>
<tr>
<td>M11</td>
<td>Move to Immediate 11-bit</td>
</tr>
<tr>
<td>CM</td>
<td>Compare</td>
</tr>
<tr>
<td>M18</td>
<td>Move to Immediate 18-bit</td>
</tr>
<tr>
<td>M24</td>
<td>Move to Immediate 24-bit</td>
</tr>
</tbody>
</table>

**Example Usage**

```assembly
ADD A, B
SUB A, C
ADD A, #100
```

**Flags**

- Z: Zero Flag
- N: Negative Flag
- V: Overflow Flag
- C: Carry Flag

**Registers**

- A: Accumulator
- B: General Register
- C: General Register
- D: General Register
- E: General Register
- F: General Register
- G: General Register
- H: General Register
- L: General Register

**Note:**
- M11, M18, and M24 are instructions for moving data to immediate values.
- CLI is used to clear the interrupt flag.
- ADD and SUB instructions are used for arithmetic operations with the accumulator (A) and another register (B or C).
- ADD #100 is an example of using an immediate value (100) to add to the accumulator.
Get Period Counter

Sample

Sample

Start

Start

Pos/Pos

Save in B

Save

Save and Period

Start Timing

Stop Period Counter

Save Period Value N-C

Get Period Counter

Change

Wait for ZP Data

Save True Value of N-C

True Value of N-C

Start Timing

Stop Period Counter

Get Period Counter

XOR A
01E8 6C  LD L, H
01E9 4A  LD C, D
01E7 C3 E01 JP SAMP

01EA 6A  TWOPLUS: LD L, D
01EB 45  LD H, L
01EC 48  LD C, B
01ED 1E02 LD E, 02H
01EE C3 E01 JP SAMP

01F2 6A  ONEMIN: LD L, C
01F3 45  LD H, L
01F4 4A  LD C, D
01F5 1E01 LD E, 01H
01F7 C3 E01 JP SAMP

01FA 1E00 CHECK: LD E, 00H
01FB 0E00 LD C, 00H
01FC 211003 LD HL, LIST+10H
0201 7E  LD A, (HL)
0202 47  LD B, A
0203 0B20  SHIFT: SLA B
0204 E60C AND CCH
0205 8A802 JP PE, EVENP
0206 CB0C  ODDP: SET O, B
0207 78  EVENP: LD A, B
0208 E60F AND 0FH
0209 47  LD B, A
020A 2C  INC L
020B 0A0F01 JP 7 START
0214 7E  LD A, (HL)
0215 88  CP B
0216 022B02 JP NZ, ERROR
0219 CB41  NOERR: BIT A, C
021B 0A0302 JP 2, SHIFT
021E 1D  JP 2, SHIFT
021F 0A08  DEC E
0221 88  ADD E
0222 D308 OUT (O8H), A
0224 1E00  LD E, 00H
0225 0E00  LD C, 00H
0228 C3002 JP SHIF
022B 1C  ERROR: INC E
022C 0E01 LD C, 01H
022E 47  LD B, A
022F 030202 JP SHIF

0232 2E0A  DELAY: LD A, DEL
0234 2E00  LD H, 00H
0236 2E00  LD L, 00H
0238 2A  LOOP: INC H
0239 C3 E02 JP 7, DELA

[62]
024E 3E00 PINIT: LD A, 00H
0250 D902 OUT (02H), A
0252 3E05 LD A, 00H
0254 D902 OUT (02H), A
0256 3E07 LD A, 07H
0258 D902 OUT (02H), A
025A 3E0F LD A, 0CFH
025C D903 OUT (03H), A
025E 3EFF LD A, 0FFH
0260 D903 OUT (03H), A
0262 3E07 LD A, 07H
0264 D903 OUT (03H), A
0266 3E00 LD A, 00H
0268 D904 OUT (04H), A
026A 3E0F LD A, 0FFH
026C D904 OUT (04H), A
026E 3E07 LD A, 07H
0270 D904 OUT (04H), A
0272 3E0F LD A, 0FFH
0274 D907 OUT (07H), A
0276 3EFF LD A, 0FFH
0278 D907 OUT (07H), A
027A 3E07 LD A, 07H
027C D907 OUT (07H), A
027E 3E00 LD A, 00H
0280 D904 OUT (04H), A
0282 3E0F LD A, 0FFH
0284 D904 OUT (04H), A
0286 3E07 LD A, 07H
0288 D904 OUT (04H), A
028A CF REt

0000 END

0135 AGAIN
01FA CHECK
000C CTC1
000D CTC2
0077 CIV
0006 DEL
023E DELA
022F DELB
0227 DELD
0228 DELB
022A ERROR
010C EVAE
01AC FIRST
0127 FORD
0008 LEDS 0000 LIST
0200 LOOP
010E MIMP
014F NEGO
0219 NOERRA
020A 000F
010A ON
0102 ONCHB
01F1 ONEMI
0179 ONEPO
018F OVER
0248 PGM
0109 PLAY
0101 PLUSPD 
0118 POLI
012A POL2
0149 POL3
015C POSTPD
017E SAMP
0191 SAMPL
019E SECO
0202 START
01EA TWWFL 
0167 WWOPS 0199 ZEROTH
PROGRAM TO WRITE IN
ENHANCED MANCHESTER CODE FROM
PRBS DATA LIST
1KB/S DATA RATE

4 TRACK

CTC USED TO DETERMINE DATA RATE

30/8/1988

WRITE PORT RJ08A:040:06C
READ PORT R108B:050:07C
WRITE CONTROL R108A:000:02C
READ CONTROL R107B:01B:05C
CTC: OCH

ORG 0100H

FD
0101 C08401
0102 3801
0103 310002
0104 011501

DI
CALL R101H
LD A,01H
LD HL, LIST+10H
JP MESEN

FLIST: LD (HL+1) A
FILL UP LIST
LD B, A
SAVE DATA
SLA B
SHIFT 0 INTO BIT 0
AND ACH
SELECT PARITY BITS
JP PE, EVENP

NDDP: SET D, B
ODD PARITY, SET BIT 0 TO 1

EVENP: LD A, B
AND ACH

INC L
RECORDED

JP MY, FLIST

NEWSEL: LD A, 00H

INC L

LD (HL+1) A
IN

LD (HL+1) A
IN
0126 3EFF  LD A, OFFH
0126 77  LD (HL), A
0126 2C  INC L
0126 77  LD (HL), A
0126 2C  INC L
0126 77  LD (HL), A
0126 2C  INC L
0126 3E00  LD A, 00H
0126 77  LD (HL), A
0126 2C  INC L
0126 77  LD (HL), A
0126 2C  INC L
0126 3E00  LD A, OFFH
0126 77  LD (HL), A
0126 2C  INC L
0126 77  LD (HL), A
0126 2C  INC L
0126 3E00  LD A, 00H
0126 77  LD (HL), A
0126 2C  INC L
0126 77  LD (HL), A
0126 E011  JP NZ, NEUSEC
0127 00  NOP

0128 3E01  PLAY:  LD A, 01H  ; SET DECK
0128 0000  OUT (COM1), A  ; START PLAY
0128 C000  CALL DELAY

0128 FF  EI
0129 3E00  LD A, 11H
0129 0000  OUT (COM1), A  ; START TIMER
0129 210003  LD HL, LIST+10H
0129 0401  LD B, 01H

012A 7E  REC:  LD A, (HL)
012A 7E  CPI  ; COMPLEMENT OF DATA
012A CB40  WAIT0:  BIT A, B
012A 25E001  JR NZ, WAIT0
012A 0401  LD B, 01H
012A 7E  LD A, (HL)
012A CB40  WAIT1:  BIT A, B
012A 25E001  JR NZ, WAIT1
012A 0401  LD B, 01H
012A 7E  INC L
012A CB40  WAIT2:  BIT A, B
012A 25E001  JR NZ, WAIT2
012A 0401  LD B, 01H
012A 2C  INC L
012A CB40  WAIT3:  BIT A, B
012A 25E001  JR NZ, WAIT3
012A 0401  LD B, 01H
012A 2C  INC L

012C 0400  WRITE:  OUT (COM1), A  ; WRITE POINT RECORD DATA
012C 0400  LD B, NOW
012C FF  EI
012D FF  RET1

C65
0184 3E00   PICINIT: LD A, 00H
0186 D302   OUT (02H), A   ; INITIALISE WRITE CONTROL
0188 3E0F   LD A, 0FH   ; TO OUTPUT
018A D302   OUT (02H), A
018C 3E07   LD A, 07H
0190 D302   OUT (02H), A
0192 D303   OUT (03H), A   ; INITIALISE READ CONTROL
0194 3E0F   LD A, 0FFH   ; TO BIT MODE INPUT
0196 D303   OUT (03H), A
0198 3E07   LD A, 07H
019A D303   OUT (03H), A
019C 3E00   LD A, 00H   ; INITIALISE WRITE
019E D306   OUT (06H), A   ; PORT TO OUTPUT
01A0 3E0F   LD A, 0FFH
01A2 D306   OUT (06H), A
01A4 3E07   LD A, 07H
01A6 D306   OUT (06H), A
01A8 3E0F   LD A, 0FFH   ; INITIALISE READ PORT
01AC 3E0F   LD A, 0FFH   ; TO BIT MODE INPUT
01AE D307   OUT (07H), A
01B0 3E07   LD A, 07H
01B2 D307   OUT (07H), A
01B4 ED5E   IM2
01B6 21F001   LD HL, TAB
01B8 7C   LD A, H
01BA ED47   LD 1, A
01BC ED217D01   LD 1Y, WRITE  
01C0 ED2E0001   LD (TAB), 1Y
01C4 7D   LD A, L
01C6 D30C   OUT (CTC), A   ; LO BYTE OF INTERRUPT VECTOR
01C8 3E95   LD A, 10010101B
01CA D30C   OUT (CTC), A   ; CTC CONTROL WORD
01CB C9   RET

01CC 3E0A   DELAY: LD A, DEL
01CE 2600   LD H, 00H
01D0 2600   LD L, 00H
01D2 24   LOOP: INC H
01D3 06801   JP Z, DELA
01D6 06201   JP LOOP
01D9 26   DELA: INC L
01DA 06801   JP Z, DELB
01DD 06201   JP LOOP
01E0 30   DELB: DEC A
01E1 06801   JP Z, ON
01E4 06201   JP LOOP
01E7 C9   OM: RET

0000   END

C66

NO ERRORS
: MILPRSR
: PROGRAM TO READ PRBS DATA
: IN MILLER CODE AND
: TO CHECK DATA
: CTC EXTERNAL CLOCK
: A TRACK
: CTC2 USED TO MEASURE RATIOS
: 16/8/1989

: WRITE PORT P108A: 04D, 06C
: READ PORT P108B: 05D, 07C
: WRITE CONTROL P107A: 00D, 02C
: READ CONTROL P107B: 01D, 03C
: LED DISPLAY P109A: 05D, 10C
: CTC1: 0C, CTC2: 0D

0027 = CH=0110111B
000C = CTC1=0CH
0000 = CTC2=00H
0008 = LEDG=00H
0000 = LIST=0000H
0004 = DEX=04H

0100 ORG 0100H
0100 D3 CALL PIDINIT
0104 D10 CALL PEXIT
0107 2E00 LD A, 00H
0109 2300 OUT (08H), A
010A 2E01 PLAY: LD A, 01H
010C 0300 OUT (00H), A
010E C6402 CALL DELAY
0112 U9
0113 210004 START: EXY
0116 0500 LD HL, LIST
0118 08 EXY
0119 2E77 LD A, 0H
011B 0300 OUT (CTC1), A
011D 2D05 IN A, (CTC1)
011F 40 LD R, A
0121 2000 POL1: IN A, (CTC1)
0122 69 XOR B
0125 4A 00 AND B
0128 C9001 OR A, POL1
012B 3EE6 LD A, OFFR

C67
012A DB05
012C DB05
012E 47
012F DB05
0131 68
0132 E602
0134 C2FO1
0137 DB0C
0139 4F
013A 3E77
013C D30C
013E 3EBF
0140 D30C
0142 91
0143 4F
0144 DB05
0146 47
0147 DB05
0149 48
014A E602
014C C44701
014F DB0C
0151 57
0152 3E77
0154 D30C
0156 3EBF
0158 DB0C
015A 92
015B 79
015D 92
015E F83201
0161 47
0162 7A
0163 C44F
0165 C44801
0168 C404
016A E40C
016C 57
016D 78
016E 42
016F C82A
0171 92
0172 C82A
0174 92
0175 F27B01
0178 C51201
017B 44
017C 48
017D 48
017E 1601
0180 C84101

OUT (CTC1), A ; START RATIO TIMER
IN A, (05H)
LD B, A
IN A, (05H)
YOR B ; WAIT FOR
AND 02H
JP 2, POL2
IN A, (CTC1) ; GET PERIOD COUNTER
LD C, A ; SAVE PERIOD VALUE N-1
LD A, CU ; STOP PERIOD COUNTER
OUT (CTC1), A ; START TIMING
LD D, OFFH ; 2ND PERIOD
OUT (CTC1), A ; TRUE VALUE OF 1ST PERIOD
LD D, A ; SAVE TRUE VALUE OF 1ST PERIOD
AGAIN: IN A, (05H) ; WAIT FOR 3RD DATA
LD B, A ; CHANGE
YOR B
AND 02H
JP 2, POL2
IN A, (CTC1) ; GET PERIOD COUNTER
LD D, A ; SAVE PERIOD VALUE N-2
LD A, CU ; STOP PERIOD COUNTER
OUT (CTC1), A ; START TIMING
LD A, OFFH ; 1ST PERIOD
OUT (CTC1), A ; TRUE VALUE OF 2ND PERIOD
LD D, A ; SAVE IN D
LD A, C
SUB D ; 1ST PERIOD-2ND PERIOD
SUB D
JP M, NEGPD
ROSEP: LD B, A ; SAVE PARTIAL RESULT
LD A, D ; ROUND OFF
BIT 1, A
JP 2, NOADD
JR Z, NOADD
ADD 04H
ADD1111100H
LD D, A
LD A, B ; GET PARTIAL RESULT
LD B, D ; SAVE D
SRL D
SUB D ; SUB 1/2 2ND PERIOD
SRL D
SUB D ; SUB 3/4 2ND PERIOD
JP R, CENTROS
JP STARTY ; TRY AGAIN
CENTROS: LD L, D
LD H, L
LD C, D ; REPLACE D
0183 47          NEQRD:    LD B, A    :SAVE PARTIAL RESULT
0184 79          LD A, C    :ROUND OFF
0185 CB4F         BIT 1, A
0187 CASC01       JR Z, NOSUM
0189 C604         ADD 04H
018C E4FC         NOSUM:    AND 11111100B
018E 4F          LD C, A
018F 78          LD A, B
0190 CB29         SRL C
0192 81          ADD C    :ADD 1/2 BIT CELL
0193 CB29         SRL C
0195 81          ADD C    :ADD 3/4 1ST PERIOD
0196 FASC01       JR M, CENTNEG
0199 C31201       TRY AGAIN
019C 49          CENTNEG:    LD L, C
019D 45          LD H, L
019E 4A          LD C, D
019F 1E01        LD E, 01H
01A1 D805        SAMP:     IN A, (05H)
01A3 47          LD B, A
01A4 D80C        SAMPL:    IN A, (C1C1) :GET PERIOD COUNTER
01A6 ED44        NEG
01A7 C6FF        ADD OFFH
01AA B0          CP L     :SAMPLE POINT?
01AB D0401       JR M, ONCMRP
01AE 70          LD A, L    :UPDATE NEXT
01AF 84          ADD H     :COMPARISON POINT
01B0 84
01B1 4F          LU L, A
01B2 CB43        OVER:     BIT 0, E    :FIRST OR 2ND SAMPLE
01B4 CA9B01      JR Z, 1ST
01B7 D805        SECOND:   IN A, (05H) :GET DATA
01B9 1E00        LD E, 00H   :SAMPLE FLAG
01BD 09
01BE CB41        BIT 0, C
01BF CA701       JR Z, READ
01C1 48          YOR B    :DECODE
01C2 77          LD (HL), A
01C3 2C          INC L
01C4 CA5502      JR Z, L.1
01C7 09          NOREAD:   EYY
01C8 C30401      JR ONCMRP
01C8 DB05        FIRST:    IN A, (05H) :GET DATA
01CD 1E01        LD E, 01H   :SAMPLE FLAG
01CE 09
01D0 0E01        LD C, 01H
01D2 47          LD B, A    :SAVE DATA
01D3 08

C69
0104 0805  ONICMR: IN A, (05H) ; GET DATA
0106 68  XOR B ; DATA CHANGE?
0107 E402  AND 02H
0109 CAAA01  JP Z: SAMPL ; NO
010C D80C  IN A, (CTC1) ; GET PERIOD COUNTER
010E 57  LD D, A ; SAVE
0110 2E77  LD A, CW ; STOP PERIOD COUNTER
0114 D30C  OUT (CTC1), A
0118 3F88  LD A, OFFH
011C D30C  OUT (CTC1), A ; START TIMING
0120 92  SUB D ; TRUE VALUE OF 2ND PERIOD
0124 57  LD D, A ; SAVE IN D
0126 79  LD A, C
0128 92  SUB D ; 1ST PERIOD-2ND PERIOD
012C FA002  JP M: MINPD
012E 47  PLUSPD: LD B, A ; SAVE PARTIAL RESULT
0130 7A  LD A, D ; ROUND OFF
0134 CB4F  BIT 1, A
0138 CB701  JP Z, ADDNOT
013C 0404  ADD 04H
013E E4FC  ADDNOT: AND 11111100B
0141 57  LD D, A ; GET PARTIAL RESULT
0143 78  LD A, B
0145 42  LD B, D
0147 CB38  SRL D
0149 92  SUB D ; SUB 1/2 PERIOD
014B CB38  SRL D
014D 92  SUB D ; SUB 3/4 PERIOD
014F FA2502  JP P: MIDPLUS
0153 6C  LD L, H
0155 48  LD C, B ; D BECOMES D
0157 C3A101  JP SAMP
015B 47  MINPD: LD B, A ; SAVE PARTIAL RESULT
015D 79  LD A, C ; ROUND OFF
0160 CB4F  BIT 1, A
0164 C9102  JP Z, SUMNOT
0168 0404  ADD 04H
016A E4FC  SUMNOT: AND 11111100B
016D 4F  LD C, A
016F 78  LD A, B
0171 CB39  SRL C
0173 81  ADD C ; ADD 1/2 BIT CELL
0175 CB39  SRL C
0177 81  ADD C ; ADD 1/4 PERIOD
0179 FA2D02  JP M: MIDMIN
017D 6C  LD L, H
017F 4A  LD C, D
0181 C3A101  JP SAMP

C70
<table>
<thead>
<tr>
<th>Address</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0225 40</td>
<td>LD H.L</td>
<td>MIDPLUS: LD L.D</td>
</tr>
<tr>
<td>0226 45</td>
<td>LD H.L</td>
<td></td>
</tr>
<tr>
<td>0227 48</td>
<td>LD C.B</td>
<td></td>
</tr>
<tr>
<td>0228 4E01</td>
<td>LD E.01H</td>
<td></td>
</tr>
<tr>
<td>0229 C8A101</td>
<td>JP SAMP</td>
<td></td>
</tr>
<tr>
<td>022D 4F</td>
<td>LD L.C</td>
<td>MIDMIN: LD L.C</td>
</tr>
<tr>
<td>022E 45</td>
<td>LD H.L</td>
<td></td>
</tr>
<tr>
<td>022F 40</td>
<td>LD C.D</td>
<td></td>
</tr>
<tr>
<td>0230 1E01</td>
<td>LD E.01H</td>
<td></td>
</tr>
<tr>
<td>0232 C8A101</td>
<td>JP SAMP</td>
<td></td>
</tr>
<tr>
<td>0235 210004</td>
<td>LOGERR: LD HL, LIST</td>
<td></td>
</tr>
<tr>
<td>0238 1600</td>
<td>LD D.00H</td>
<td></td>
</tr>
<tr>
<td>0239 7E</td>
<td>LD A, (HL)</td>
<td>MEYA: CP 00H</td>
</tr>
<tr>
<td>023B FE00</td>
<td>JR Z, MEY1</td>
<td></td>
</tr>
<tr>
<td>023C C4A102</td>
<td>INC D</td>
<td></td>
</tr>
<tr>
<td>0240 1A</td>
<td>INC L</td>
<td>MEY1: INC L</td>
</tr>
<tr>
<td>0241 2C</td>
<td>JR Z, MEY7</td>
<td>MEY2: INC L</td>
</tr>
<tr>
<td>0242 C47C02</td>
<td>JR Z, MEY7</td>
<td></td>
</tr>
<tr>
<td>0245 7E</td>
<td>LD A, (HL)</td>
<td></td>
</tr>
<tr>
<td>0246 FE00</td>
<td>JR Z, MEY2</td>
<td></td>
</tr>
<tr>
<td>0248 C4A002</td>
<td>INC D</td>
<td></td>
</tr>
<tr>
<td>024B 1A</td>
<td>INC L</td>
<td>MEY3: INC L</td>
</tr>
<tr>
<td>024C 2C</td>
<td>JR Z, MEY7</td>
<td>MEY4: INC L</td>
</tr>
<tr>
<td>024D C47C02</td>
<td>JR Z, MEY7</td>
<td></td>
</tr>
<tr>
<td>0250 7E</td>
<td>LD A, (HL)</td>
<td></td>
</tr>
<tr>
<td>0251 FE00</td>
<td>JR Z, MEY4</td>
<td></td>
</tr>
<tr>
<td>0252 C45702</td>
<td>INC D</td>
<td></td>
</tr>
<tr>
<td>0256 1A</td>
<td>INC L</td>
<td>MEY5: INC L</td>
</tr>
<tr>
<td>0257 2C</td>
<td>JR Z, MEY5</td>
<td>MEY6: INC L</td>
</tr>
<tr>
<td>0258 C47C02</td>
<td>JR Z, MEY7</td>
<td></td>
</tr>
<tr>
<td>025B 7E</td>
<td>LD A, (HL)</td>
<td></td>
</tr>
<tr>
<td>025C FE00</td>
<td>JR Z, MEY4</td>
<td></td>
</tr>
<tr>
<td>025E C46202</td>
<td>INC D</td>
<td></td>
</tr>
<tr>
<td>0261 1A</td>
<td>INC L</td>
<td>MEY7: JR NZ, MEY6</td>
</tr>
<tr>
<td>0262 2C</td>
<td>JR Z, MEY7</td>
<td></td>
</tr>
<tr>
<td>0263 C47C02</td>
<td>JR Z, MEY7</td>
<td></td>
</tr>
<tr>
<td>0266 7E</td>
<td>LD A, (HL)</td>
<td></td>
</tr>
<tr>
<td>0267 FE00</td>
<td>JR NZ, MEY0</td>
<td></td>
</tr>
<tr>
<td>0268 C46102</td>
<td>INC D</td>
<td></td>
</tr>
<tr>
<td>026C 1A</td>
<td>INC L</td>
<td>MEY8: JR Z, START</td>
</tr>
<tr>
<td>026D 2C</td>
<td>JR Z, MEY7</td>
<td></td>
</tr>
<tr>
<td>026E C47C02</td>
<td>JR Z, MEY7</td>
<td></td>
</tr>
<tr>
<td>0271 7E</td>
<td>LD A, (HL)</td>
<td></td>
</tr>
<tr>
<td>0272 FE00</td>
<td>JR Z, MEY6</td>
<td></td>
</tr>
<tr>
<td>0274 C47802</td>
<td>INC D</td>
<td></td>
</tr>
<tr>
<td>0277 1A</td>
<td>INC L</td>
<td>MEY9: JR NZ, MEY0</td>
</tr>
<tr>
<td>0278 2C</td>
<td>JR Z, MEY7</td>
<td></td>
</tr>
<tr>
<td>0279 C42802</td>
<td>INC D</td>
<td></td>
</tr>
<tr>
<td>027C 7A</td>
<td>JR Z, START</td>
<td></td>
</tr>
<tr>
<td>027D FE00</td>
<td>IN A, (08H)</td>
<td></td>
</tr>
<tr>
<td>027E C41201</td>
<td>INC A</td>
<td></td>
</tr>
<tr>
<td>0282 D008</td>
<td>OUT (084H), A</td>
<td></td>
</tr>
<tr>
<td>0284 3C</td>
<td>JR START</td>
<td></td>
</tr>
<tr>
<td>0285 0008</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0287 C01201</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
028A 1600  CHECK:  LD D, 00H
028C 1600  LD E, 00H
028E 210004  LD HL, LIST
0291 7E  DATA1:  LD A, (HL)  ; GET 1ST DATA
0292 47  LD B, A
0292 78  REPT:  LD A, B
0294 CB20  SLA B
0294 E407  AND 0CH
0298 E8D02  JP PE, EVENP
029B CB00  ADD:  SET D, B
029D 78  EVENP:  LD A, B
029E 640F  AND 0FH
02A0 47  LD B, A
02A1 2C  INC L
02A2 C9C02  JP Z, UPDATE
02A5 7E  LD A, (HL)
02A6 B9  CP B
02A7 C2502  JP NZ, ERROR
02A8 CB43  BIT 0, E  ; ERROR LOGGED?
02A9 C9302  JP Z, REPT
02AA 15  DEC D
02A9 1600  LD E, 00H
02AB C9302  JP REPT
02AC 14  ERROR:  INC D
02AD 1601  LD E, 01H
02AA 47  LD B, A  ; REPLACE PREV
02A9 C9302  JP REPT
02BA DB08  UPDATE:  IN A, (O8H)  ; GET LEDS
02BE E2  ADD D
02BF D308  OUT (O8H), A
02C0 C31201  JP START

02C4 2E04  DELAY:  LD A, DEL
02C6 2E00  LD D, 00H
02C8 2E00  LD L, 00H
02CA 2A  LOOP:  INC H
02CB C9D02  JP Z, DELA
02CE C3C02  JP LOOP
02D1 2C  DELA:  INC L
02D2 C9802  JP Z, DELB
02D5 C3C02  JP LOOP
02D8 3D  DELB:  DEC A
02DA C9E02  JP Z, OM
02DB C3C02  JP LOOP
02DF 09  OM:  RET

02E0 3E00  P1INIT:  LD A, 00H  ; INITIALISE WRITE CONTROL
02E2 D302  OUT (O2H), A  ; TO OUTPUT
02E4 3E00  LD A, 00H
02E4 3E07  OUT (O2H), A
02E8 3E00  LD A, 07H
02EA D302  OUT (O2H), A

C72
02EC 3ECF  LD A, 00FH  : INITIALISE READ CONTROL
02ED 3D03  OUT (03H), A  : TO BIT MODE INPUT
02EF 3EFF  LD A, 00FH  : INITIALISE WRITE
02E2 3D03  OUT (03H), A  : PORT TO OUTPUT
02E4 3E07  LD A, 07H
02E4 3D03  OUT (03H), A
02E8 3E00  LD A, 00H  : INITIALISE READ PORT
02E8 3D07  OUT (07H), A  : TO BIT MODE INPUT
02EA 3E04  LD A, 00H  : INITIALISE LED
02EA 3D07  OUT (07H), A  : PORT TO OUTPUT
02EE 3E07  LD A, 07H
02EE 3D07  OUT (07H), A
0304 3E00  LD A, 00H
0306 3E04  OUT (04H), A  : INITIALISE LED
0306 3D07  LD A, 07H  : PORT TO OUTPUT
030C 3D07  OUT (07H), A
0310 3E00  RET
0312 3D04  OUT (04H), A
0314 3E00  OUT (00H)
0316 3D04  OUT (04H), A
0318 3E07  LD A, 07H
031A 3D04  OUT (04H), A
031C 3F  REGINIT: EYX
031E 210004  LD HL, LIST
0321 0E00  LD C, 00H
0323 09  EYX
0324 09  RET

0000   END

01F7 ADDMNT 0144 AONIN 019C CENTINE 0176 CENTRO 028A CHECK
0000  CTC1 0000 CTC2 0072 CW 0281 DATAT 000A UEL
0201 DELA 0204 DELAY 0208 DELP 0285 ERROR 0290 EVENP
019F FIRST 012C FORM 0008 LEDS 0400 LIST 0286 LOGERR
0204 LOOP 022D MIDMIN 0225 MIDPLU 020A MINPD 0183 NEGPU
022A NEYO 0241 NEY1 024C NEY2 0257 NEY3 0262 NEY4
0264 NEYS 0278 NEY6 027C NEY7 016A NOADD 0107 NOREAD
015C NOSUM 029B ODDP 0204 ON 0104 ODMEM 01B2 OVER
020A PLOINT 0100 PLAY 01EE PLUSPD 0120 POLI 012F POL2
0167 POLY 0161 POSPD 0310 REGINI 0292 REPI 0141 SAPM
016A SAMPL 0167 SECON 0112 START 0212 SUMMN 029C UPDATE

NO ERRORS

C73