DEVELOPMENT STANDARD

QIC-3080-MC
Revision C
15 Dec 94

SERIAL RECORDED MAGNETIC TAPE MINICARTRIDGE
FOR INFORMATION INTERCHANGE

Streaming Mode
Transition Density: 45,000 ftpi (1772 ftpmm)
Data Density: 60,000 bpi (2362 bpmm)
RLL 1,7 Encoding
Reed-Solomon ECC
Media: 900 Oe Gamma Ferric Oxide (Preformatted)
Non Read-While-Write

Uncompressed Formatted Capacity:
1.18 GBytes with 295-foot 0.250 in. (6.35 mm) tape (60 tracks)
1.6 GBytes with 400-foot 0.250 in. (6.35 mm) tape (60 tracks)
2 GBytes with 400-foot 0.315 in. (8 mm) tape (77 tracks)

Quarter-Inch
Cartridge
Drive Standards, Inc.

311 East Carrillo Street
Santa Barbara, California 93101
Telephone (805) 963-3853
Fax (805) 962-1541
www.qic.org

(See important notices on the following page)

Important Notices

This document is a development standard adopted by Quarter-Inch Cartridge Drive Standards, Inc. (QIC). This document may be revised several times during the development cycle. It is intended solely as a guide for companies interested in developing products which can be compatible with other products developed using this document. QIC makes no representation or warranty regarding this document, and any company using this document shall do so at its sole risk, including specifically the risks that a product developed will not be compatible with any other product or that any particular performance will not be achieved. QIC shall not be liable for any exemplary, incidental, proximate or consequential damages or expenses arising from the use of this document. This development standard defines only one approach to the product. Other approaches may be available in the industry.

This development standard is an authorized and approved publication of QIC. The underlying information and materials contained herein are the exclusive property of QIC but may be referred to and utilized by the general public for any legitimate purpose, particularly in the design and development of quarter-inch tape cartridge drive subsystems. This development standard may be copied in whole or in part *provided* that no revisions, alterations or changes of any kind are made to the materials contained herein. Only QIC has the right and authority to revise or change the material contained in this development standard, and any revisions by any party other than QIC are totally unauthorized and specifically prohibited.

Compliance with this development standard may require use of one or more features covered by proprietary rights (such as features which are the subject of a patent, patent application, copyright, mask work right or trade secret right). By publication of this development standard, no position is taken by QIC with respect to the validity or infringement of any patent or other proprietary right, whether owned by a Member or Associate of QIC, or otherwise. QIC hereby expressly disclaims any liability for infringement of intellectual property rights of others by virtue of the use of this development standard. QIC has not and does not investigate any notices or allegations of infringement prompted by publication of any QIC development standard, nor does QIC undertake a duty to advise users or potential users of QIC development standards of such notices or allegations. QIC hereby expressly advises all users or potential users of this development standard to investigate and analyze any potential infringement situation, seek the advice of intellectual property counsel, and, if indicated, obtain a license under any applicable intellectual property right or take the necessary steps to avoid infringement of any intellectual property right. QIC expressly disclaims any intent to promote infringement of any intellectual property right by virtue of the evolution, adoption, or publication of any QIC development standard.

REVISION HISTORY

January 30, 1994	Adopted, based on QIC-9383 revision B
March 10, 1994	Revised para 1.1, 4.2, 5.1, 6.2, 6.10, and Fig 6.1 in accordance with QIC-94-29
December 15, 1994	Revised para 9.3.2 and added para 1.2.5 in accordance with QIC-94-85

TABLE OF CONTENTS

FIGURES

TABLES

1. SCOPE AND INTRODUCTION

1.1 SCOPE

This Standard provides a format and recording standard for a streaming magnetic tape in a mini-cartridge to be used for information interchange between information processing systems, communication systems, and associated equipment utilizing a standard code for information interchange, as agreed upon by the interchange parties. The Standard provides the following capacities of formatted data on a single mini-cartridge with error correction codes:

a. With 295ft 0.250 in (6.3 mm) tape, 60 tracks 1.18 GBYTES

b. With 400 ft 0.250 in (6.3 mm) tape, 60 tracks 1.6 GBYTES

c. With 400 ft 0.315 in (8.0 mm) tape, 77 tracks 2.0 GBYTES

This standard refers solely to recording on a magnetic tape mini-cartridge meeting the requirements of QIC 152 Preformatted Mini Cartridge Standard.

1.2 INTRODUCTION

1.2.1 This standard defines the requirements of supporting test methods necessary to ensure interchange at acceptable performance levels. It is distinct from a specification in that it delineates a minimum of restrictions consistent with compatibility in interchange transactions. The standard uses a Reed Solomon error correction code to achieve a corrected bit error rate of at least 10^{-15} when data is recorded in compliance with the requirements of Section 5, given an equivalent defect density of up to 30 DPSI. The 30 equivalent DPSI is divided into 15 media defects per square inch and an equivalent 15 errors due to noise per square inch.

1.2.2 The performance levels contained in this standard represent the minimum acceptable levels of performance for interchange purpose. They therefore represent the performance levels which the interchanged items should meet or surpass during their useful life and thus define end-of-life criteria for interchange purposes. The performance levels in this standard are not intended to be employed or substituted for purchase specification.

1.2.3 Wherever feasible, quantitative performance levels are specified which must be met or exceeded in order to comply with this standard. In all cases, including those in which quantitative limits for requirements falling within the scope of this standard are not stated but are left to agreement between interchange parties, standard test methods and measurement procedures shall be used to determine such quantities.

1.2.4 U.S. engineering units are the original dimensions in this standard. Conversions of toleranced dimensions from customary U.S. engineering units (similar to British Imperial Units) to SI units have been done in this standard according to ANSI/IEEE STD 268-1982 and ISO 370-1975 Method A. Method A should be used for economy unless a requirement for absolute assurance of a fit justifies use of Method B. In the national standards of ISO member nations, additional rounding may be done to produce "preferred" values. These values should lie within or close to the original tolerance ranges.

1.2.5 Except as indicated in 1.2.3 above, interchange parties complying with the applicable standards should be able to achieve compatibility without need for additional exchange of technical information.

This standard makes several references to QIC-121.

However, this does not imply that a drive meeting this standard must be compatible with QIC-121. This standard sets no requirements as to the interface of the drive.

2. DEFINITIONS

For the purpose of this standard, the following definitions apply:

2.2 Bad Block. A block determined to be bad during a read operation.

2.3 Bit. A single digit in the binary system.

2.4 Bit Cell. The physical length of a recorded encoded bit along the track. In this standard, the bit cell length must be measured indirectly, by measuring the length of a minimum Transition Cell.

2.5 Block. A group of 512 consecutive data bytes plus additional control bytes recorded as a unit.

2.6 Block Marker. A group of encoded bits following the preamble and marking the start of each block.

2.7 BOT (Beginning of Tape) Marker. The BOT Marker is a set of two holes punched side by side in the tape. There are four sets of holes provided, the innermost of which is used for identifying the storage position for the cartridge. The additional sets of holes are used to ensure reliability of detection.

Note: In the storage position, all of the permissible recording area of the tape is wound on the supply hub and is protected by at least one layer of tape not used for recording data. Cartridges to be interchanged shall be rewound to the storage position prior to interchange.

2.8 Byte. A group of 8 data bits operated on as a unit.

2.9 Cancel Mark. The Cancel Mark acts as a "negative" File Mark or Set Mark. When a Cancel Mark follows as the first block in the next frame after a File Mark or Set Mark, the drive when reading the tape will logically ignore the Cancel Mark and the File Mark or Set Mark it cancels.

2.11 Control Block. A block designated as a Control block. This Standard does not define the use of control blocks nor the contents of the data area of the control block.

2.12 Control Field. A group of 8 bytes recorded before the data area in each block, containing information about block address, track address and block type.

2.13 CRC (Cyclic Redundancy Check). The CRC is a group of 4 bytes recorded at the end of each block of data for the purpose of error detection.

2.14 Data Block. A block containing user valid data in its data field.

2.15 Data Density. The nominal distribution of recorded data information per unit length of track, usually expressed in bits per inch (bpi) or bits per millimeter (bpmm). In this standard, the data density is higher than the transition density.

2.16 DPSI (Defect Per Square Inch). The DPSI number specifies the number of defects (using a defined read method) per square inch of tape.

2.17 ECC (Error Correction Code). Special drive generated information which may be used to correct bad blocks.

2.18 ECC Block. A block containing drive generated ECC data in its data field and part of control field.

2.19 Encoding. A method where by a group of data bits is translated into a group of recording bits. In this standard, 2, 4 or 8 data bits are translated into 3, 6 or 12 recording bits.

2.20 EOD (End of Data) Marker. The EOD marker is used to mark the end of the data area. The marker consists of a minimum of 64K recordings of a 2-byte postamble pattern.

2.21 EOT (End of Tape) Marker. The EOT Marker is a single hole punched in the tape to indicate that the usable recording area of the tape has been exceeded, and that the physical end of the tape is approaching. There are three EOT holes to ensure reliable detection.

2.22 EW (Early Warning) Marker. The EW Marker is a single hole punched in the tape to indicate the approaching end of the usable recording area in the forward direction.

2.23 File Mark Block. A block designated as a File Mark.

2.24 Filler Block. A block containing no valid information in its data field. The purpose of this block is to complete a frame in the case that the host cannot fill the whole frame with valid data information.

2.25 Flux Transition. A point on the magnetic tape which exhibits maximum free space flux density normal to the tape surface.

2.26 Flux Transition Spacing. A distance on the magnetic tape between flux transitions.

2.27 Frame. A group of 64 blocks forming a complete logical unit.

2.28 GBytes (GB). This standard defines 1 GB to be equal to 10^9 Bytes.

2.29 Identifier Block. A unique block identifying the type of format being recorded.

2.30 KBytes (KB). This standard defines 1 KB to be equal to 1024 bytes.

2.31 LP (Load Point) Marker. The LP Marker is a single hole punched in the tape to indicate the approaching start of the usable recording area in the forward direction.

2.32 Magnetic Tape Cartridge. A cartridge containing magnetic tape wound on two coplanar hubs with an internal drive belt to transport the tape between the hubs.

2.33 MBytes (MB). This standard defines 1 MB to be equal to 10^6 bytes.

2.34 Physical Recording Density. See transition density.

2.35 Postamble. A special sequence of bits recorded at the end of each block.

2.36 Preamble. A special sequence of bits recorded at the beginning of each block.

2.38 Recorded Azimuth. The angular deviation, in minutes of arc, of the recorded mean flux transition line from the line normal to the cartridge reference plane.

2.39 Reference Tape Cartridge. A tape cartridge selected for a given property for calibrating purposes.

2.40 Reserved. Reserved fields are to be written with zeros and ignored by firmware to facilitate future use by QIC.

2.41 RLL (Run Length Limited). A data encoding method where data bits are encoded so that certain constraints are met with regard to the maximum and minimum distances between flux transitions.

2.42 Secondary Reference Tape Cartridge. A tape cartridge intended for routine calibration purposes, the performance of which is known and stated in relation to that of the Reference Tape Cartridge.

2.43 Set Mark Block. A block designated as a Set Mark.

2.44 Signal Amplitude Reference Tape Cartridge. A reference cartridge selected as a standard for signal amplitude and reference field.

2.45 Standard Reference Amplitude. The average peak-to-peak signal amplitude output of the Signal Amplitude Reference Cartridge when it is recorded on an TBD measuring system at the maximum flux density specified in this standard.

2.46 Streaming. A method of recording on magnetic tape that maintains continuous tape motion without the requirement to start and stop within an interblock gap.

2.47 Track. A longitudinal area on the tape along which magnetic signals may be serially recorded.

2.48 Transition Cell. The physical distance between two adjacent flux transition at the maximum recording density.

2.49 Transition Density or Physical Recording Density. The number of recorded flux transitions per unit length of track, usually expressed in flux transitions per inch (ftpi) or flux transitions per millimeter (ftpmm). See also Data Density.

2.50 Underrun. A condition developed when the host transmits or receives data at a rate less than required by the device for streaming operation.

2.51 Vendor Specific. Vendor Specific fields may be used to implement unique features beyond the scope of this document.

3. REFERENCE EDGE

The Reference Edge shall be that edge of the tape which is nearest to the baseplate of the cartridge.

4. TRACK GEOMETRY

4.1 TRACK POSITIONS

Each track is written referenced to servo patterns written between the BOT holes and Load Point on the BOT end of the tape, and between EOT and EW on the EOT end of the Tape. The servo pattern is written across the entire width of the tape. Track zero is the lowest position, approximately 0.005 inches from the reference edge of the tape. Each subsequent track is written on the next position above the previous track.

4.2 TRACK WIDTH

The width of the recorded track shall be 0.00390 ±0.00008 in. (0.099 ±0.002 mm). The read track width is 0.002 ± 0.000040 in. (0.051 ± 0.001 mm).

4.4 QUICK FILE ACCESS

This standard supports Quick File Access (QFA).

All tapes are recorded with 2 partitions. The data (or default) partition is designated 0. It shall be recorded on all tracks except track 0. Partition 1 shall be the directory partition and shall be recorded on track 0 only.

5. RECORDING

5.1 METHOD OF RECORDING.

The recording method shall be the Non Return to Zero Mark (NRZI) method where a ONE is represented by a change in direction of longitudinal magnetization.

The recording current shall be **1.15** x Isat ± 3% where Isat is the current providing 95% of the maximum output at 45,000 ftpi on media at 25°C. The Isat is measured on the non-saturated side of the saturation current curve.

5.2 WRITE EQUALIZATION

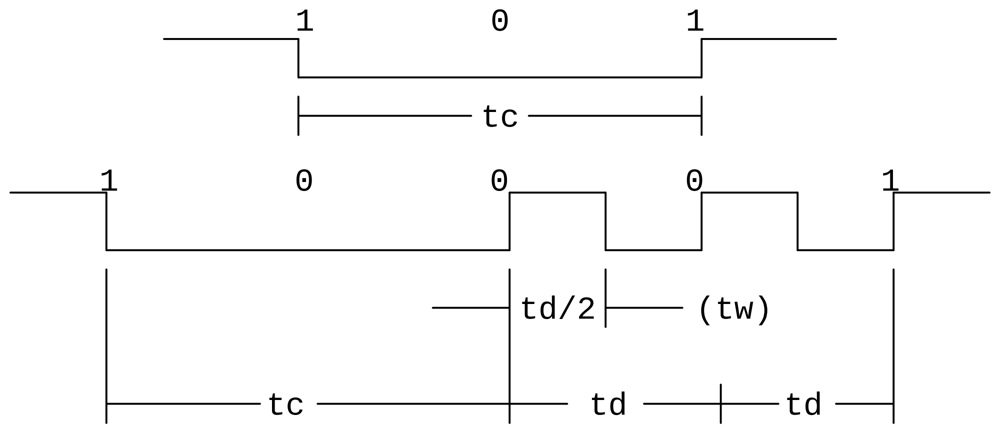
To minimize problems due to the large transition spacing ratio (4:1), write pulse equalization must be used. This section describes the write pulse equalization method required to conform to this standard:

For every "zero" other than the first "zero" following a "one", one or more additional write equalization pulses shall be inserted into the waveform as shown in figures 5.1 and 5.2. The location of the inserted pulse (pulses) with reference to the bit cell shall be exactly as specified in the figures.

The width of the equalization pulse (pulses) shall be 1/4 of the nominal transition period t_c (± .01%) as shown in figure 5.2.

```
                 0     1     0     1     0
                 ------+           +------
                       ¦           ¦
                       ¦           ¦
                       +-----------+
                        |<---tc---->|
              Figure 5.1.1 Pattern ..01010..

              0     1     0     0     1     0
              ------+           +--+  +------
                    ¦           ¦  ¦  ¦
                    ¦           ¦  ¦  ¦
                    +-----------+  +--+
                     |<--tc----->|<-td>¦
             Figure 5.1.2  Pattern ..010010..

           0     1     0     0     0     1     0
           ------+           +--+  +--+  +------
                 ¦           ¦  ¦  ¦  ¦  ¦
                 ¦           ¦  ¦  ¦  ¦  ¦
                 +-----------+  +--+  +--+
                  |<--tc----->|<-td>|< td>|
             Figure 5.1.3  Pattern ..0100010..

        0     1     0     0     0     0     1     0
        ------+           +--+  +--+  +--+  +------
              ¦           ¦  ¦  ¦  ¦  ¦  ¦  ¦
              ¦           ¦  ¦  ¦  ¦  ¦  ¦  ¦
              +-----------+  +--+  +--+  +--+
               |<--tc----->|<-td>|<-td>|<-td>|
            Figure 5.1.4  Pattern ..01000010..

     0     1     0     0     0     0     0     1     0
     ------+           +--+  +--+  +--+  +--+  +------
           ¦           ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦
           ¦           ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦
           +-----------+  +--+  +--+  +--+  +--+
              |<--tc----->|<-td>|<-td>|<-td>|<-td>|
            Figure 5.1.5  Pattern ..010000010..

  0     1     0     0     0     0     0     0     1     0
  ------+           +--+  +--+  +--+  +--+  +--+  +------
        ¦           ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦
        ¦           ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦
        +-----------+  +--+  +--+  +--+  +--+  +--+
         |<--tc>-----|<-td>|<-td>|<-td>|<-td>|<-td>|
           Figure 5.1.6  Pattern ..0100000010..

    1     0     0     0     0     0     0     0     1     0
----+           +--+  +--+  +--+  +--+  +--+  +--+  +------
    ¦           ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦
    ¦           ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦
    +-----------+  +--+  +--+  +--+  +--+  +--+  +--+
     |<--tc----->|<-td>|<-td>|<-td>|<-td>|<-td>|<--d>¦
           Figure 5.1.7  Pattern ..01000000010..
```

Figure 5.1 Write Waveforms and Equalization



td = tc/2
tw = td/2

Maximum variation of tw is ±0.01% from nominal value.

Figure 5.2 Timing Information, Write Equalization

5.3 TRANSITION DENSITIES

The nominal maximum physical recording density or transition density shall be 45,000 ftpi (1772 ftpmm). The nominal transition cell length shall be 22.22 micro-inches (0.564 μm).

With the recording method used in this Standard, seven transition densities may occur on the tape:

45,000 ftpi (1772 ftpmm)
30,000 ftpi (1181 ftpmm)
22,500 ftpi (886 ftpmm)
18,000 ftpi (709 ftpmm)
15,000 ftpi (591 ftpmm)
12,857 ftpi (506 ftpmm)
11,250 ftpi (443 ftpmm)

5.4 AVERAGE TRANSITION CELL LENGTH VARIATIONS

5.4.1 Average Transition Cell Length

The average transition cell length is the sum of the distances between the flux transitions in n transition cells divided by (n-1). The tests referred to below may be made in any continuously recorded pattern, provided the first and the last transition cell in the pattern each contain a flux transition.

5.4.2 Long Term Average Transition Cell Length

The long term average transition cell length is the average bit cell length taken over a minimum of 2,000,000 transition cells. The long term average transition cell length shall be within +/- 3% of the nominal bit cell length of 22.22 μinch (0.564 μm).

5.4.3 Medium Term Average Transition Cell Length

The medium term average transition cell length is the average transition cell length taken over a minimum of 30,000 transition cells and a maximum of 34,000 transition cells. The medium term average transition cell length shall be within +/- 6% of the long term average transition cell length.

5.4.4 Short Term Average Transition Cell Length

The short term average transition cell length is the average transition cell length taken over a minimum of 38 transition cells and a maximum of 50 transition cells. The short term average transition cell length shall be within ± 2% of the medium term average transition cell length.

5.4.5 Instantaneous Flux Transition Spacing

The instantaneous spacing between flux transitions is influenced by the reading and writing process, the pattern recorded (pulse-crowding effect) and other factors. Instantaneous spacings between flux transitions shall satisfy the following conditions:

In a sequence of flux transitions defined by the encoded pattern ..010101000000010101.., the center flux transition of each group of 010101's is called a reference flux transition. The maximum displacement of flux transitions on either side of the reference flux transitions shall not exceed +/- 12.5% of the transition cell length d1 averaged over the six transition cells between the reference flux transitions indicated in the bit pattern in figure 5.4.

```
0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  1  0  1
   +-----+     +-----------------------+     +-----+
   ¦     X     ¦                       ¦     X     ¦
---+     +-----+                       +-----+     +-

   |<-d->|<-d->|                       |<-d->|<-d->|
         |<---------------T----------------->|
```

$$d1 = T/6$$
$$d1(1-0.125) < d < d1(1+0.125)$$

X denotes a reference flux transition.

Figure 5.4 Test Pattern for Instantaneous Flux Transition Spacing Test.

5.5 SIGNAL AMPLITUDE OF A RECORDED CARTRIDGE FOR DATA INTERCHANGE

When performing the tests described below, the output or resultant signal shall be measured on the same pass for both the Standard Amplitude Reference Cartridge and the tape under test. The measurements specified in 5.5.1 through 5.5.3 shall be performed during the first read pass after the write pass. The same equipment shall be used for all measurements. The signal amplitude shall be measured at a point in the read channel where the signal is proportional to the head output.

After writing, the cartridge shall meet the following requirements:

5.5.1 Average Signal Amplitude at Nominal Maximum Density

At the nominal maximum physical recording density of 45,000 ftpi (1772 ftpmm), the Average Peak-to-Peak Signal Amplitude of any track on the interchange tape shall deviate no more than + 25% or - 25% from the Standard Reference Amplitude recorded at 45,000 ftpi (1772 ftpmm) measured at a temperature of 20^{o} C ± 4^{o} C. This averaging shall be made over the central 60 flux transitions of any 64 or more flux transitions recorded at nominal maximum recording density in a block and over at least 600 blocks.

5.5.2 Maximum Signal Amplitude

When interchanged, a tape shall not contain, in the valid information area, any flux transitions where the peak-to-peak signal amplitude is more than twice the Standard Reference Amplitude at 45,000 ftpi (1772 ftpmm) measured at a temperature of 20^{o} C ± 4^{o} C.

5.5.3 Minimum Signal Amplitude

When interchanged, a tape shall not contain, in its valid information area, any flux transitions where the peak-to-peak signal amplitude is less than 50% of the Standard Reference Amplitude at 45,000 ftpi (1772 ftpmm) measured at a temperature of 20^{o} C ± 4^{o} C, exclusive of media defects.

5.6 RECORDED AZIMUTH

On any track the angle that a flux transition across the track makes with a line perpendicular to the cartridge reference plane shall not exceed 9.5 minutes of arc (2.76 mrad).

5.7 ERASURE

Erasure shall be by means of overwrite only.

5.8 OVERWRITE

Overwritten tracks shall not contain any components of previously recorded information whose amplitudes exceeds -26 dB relative to the amplitude of the newly recorded data.

6. USE OF TRACKS

6.1 DATA TRACKS

Each track shall be a data track and shall be written serially, one track at a time.

6.2 TRACK NUMBERING

The tracks are number from 0 to T_{max} where T_{max} is 59 for 0.250 in wide tape and 76 for 0.315 in wide tape. The track nearest the reference edge is track 0. The rest of the tracks are numbered sequentially counting up from track 0.

All odd numbered tracks, and track 0, shall be recorded in the forward direction (the direction from the BOT marker to the EOT marker). All even numbered tracks, except track 0, shall be recorded in the reverse direction (the direction from the EOT marker to the BOT marker).

6.5 MINIMUM/MAXIMUM DISTANCES, FORWARD TRACKS

On all odd numbered tracks (1, 3, ... etc.) and track 0, the beginning of the preamble of the first data frame shall commence a minimum distance of 1 inches (25 mm) and a maximum distance of 2 inches (51 mm) past the LP marker.

The valid data area shall terminate at most a distance of 2 inches (51 mm) and at least a distance of 1 inches (25 mm) before the EW marker, measured from the center of the EW hole.

6.6 MINIMUM/MAXIMUM DISTANCES, REVERSE TRACKS

On all even numbered tracks except track 0 (2, 4, ... etc.) the beginning of the preamble of the first data frame shall commence a minimum distance of 3 inches (76 mm) and a maximum distance of 4 inches (101 mm) past the EW marker.

No data shall be recorded beyond 2 inches (50 mm) past the LP marker.

6.7 TRACK ID FRAME

Following the Servo Pattern Bursts, a Track ID Frame is pre-recorded on each track on both ends of the tape as part of the servo writing process. On the BOT end, the frames are recorded in the forward direction, and end at least 2.5 inches before the LP hole. On the EOT end, the frames are recorded in the reverse direction and end at least 0.125 inches before the EW hole.

There is a Track ID Frame on both ends of every track. This allows the drive to verify which track it is on each time it completes the servo acquisition process. Care must be taken in normal recording operations to leave the Track ID Frames intact when ending each track. Following the Track ID Frames, low frequency preamble pattern is recorded at least 6 inches past the LP or EW hole during the tape format process. Part of this preamble pattern will be over-written the first time data is written on the track.

The first block in each Track ID Frame is given the Physical Block Address of 0. The Physical Block Address increments for each block of the frame. Control Byte 0 for each of the data blocks is set to 1E Hex, indicating an Identifier block. The Track Number is recorded in Control Byte 4. ECC encoding is performed on the frame prior to recording it on the tape. See section 9 for more information on frame construction.

6.9 SUMMARY OF REQUIREMENTS

Table 6.1 and figure 6.1 summarize the requirements in sections 6.1 to 6.8.

	Min	Max	Description
D1	2.5 in. 63.5 mm	- -	End of track ID to LP on all tracks.
D2	1 in. 25 mm	2 in. 51 mm	LP to start of valid data area on all forward tracks.
D3	1 in. 25 mm	2 in. 51 mm	End of data to EW on all forward tracks.
D4	- -	2 in. 51 mm	LP to end of data on all reverse tracks.
D5	3 in. 76 mm	4 in. 102 mm	EW to start of valid data area on all reverse tracks.
D6	.300 in. 7.6 mm	- -	End of track ID to EW on all tracks.

Table 6.1 Summary of Requirements for Use of Tracks.

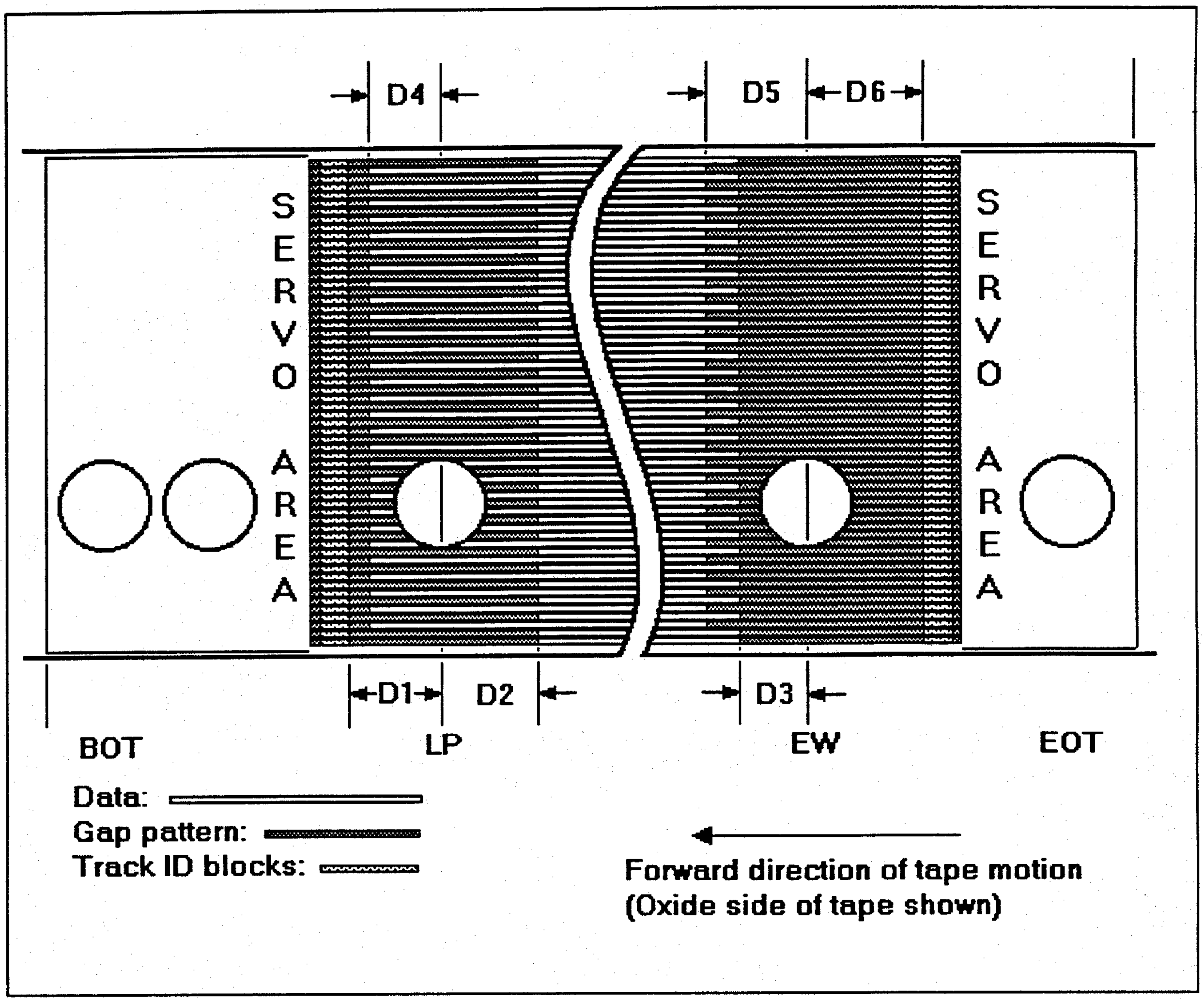


Figure 6.1 Requirements for Use of Tracks.

6.10 TRACK REFERENCE SERVO PATTERN

To allow increased track density, a track servo reference pattern is pre-recorded on the tape. The pattern is recorded in the region between the BOT hole and the LP hole. The same pattern is recorded between the EOT hole and the EW hole. For more information regarding these reference servo patterns, see document QIC 152 "Preformatted Mini Cartridge Standard".

7. BYTE AND CODE REQUIREMENTS

7.1 BYTE LENGTH

The data shall be in eight-bit bytes. The 8 bits in each byte are numbered b0 to b7, b7 being the most significant bit.

8. DATA RANDOMIZING AND ENCODING

Prior to the recording of the data on the tape, the coded characters (see section 7) shall be modified by a special data randomizer circuit (see section 8.1). The randomized information shall then be encoded according to section 8.2 before being recorded on the tape.

Except when otherwise indicated in the description of the tape format, all bytes to be recorded shall be randomized and encoded as described in 8.1 and 8.2.

8.1 DATA RANDOMIZER

In order to reduce problems due to long strings of repetitive data with bad peak shift or amplitude characteristics, a special data randomizer algorithm shall be used on all bytes in the data and control area of each block. This data randomizing process shall take place before the data is encoded according to section 8.2.

Assuming that the data content is converted into a serial stream prior to being encoded, the data stream shall be "randomized" using the following generator polynomial:

$$g(x) = x^{12} + x^{6} + x^{4} + x + 1$$

Figure 8.1 shows one method to achieve this randomizer algorithm. It consists of 12 flip-flops (marked x0 to x11) organized into a twelve bit shift register. Exclusive-or gates are inserted in the data stream between register 0 and 1, between 3 and 4 and between 5 and 6. The other inputs to the exclusive-or gates are taken from the output of shift register 11. The serial data stream to be randomized (or "de-randomized" in the case of a read operation) is exclusive-or'ed with the output of register 0.

The randomizer starts with the most significant bit of byte 7 in the Control area and ends up with the least significant bit of byte 511 in the data area.

```
           +--+ +--+ +--+ +--+ +--+ +--+              +--+
           ¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦      +----+  ¦  ¦
     +---<-¦  +-¦  +-¦  +-¦  +-¦  +-¦  +-<-¦x-or+<-¦  +-<-+
     ¦     ¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦    ¦    ++  ¦  ¦   ¦
     ¦     ¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦ ¦  ¦    +----+¦  ¦  ¦   ¦
     ¦     +--+ +--+ +--+ +--+ +--+ +--+          ¦  +--+   ¦
     ¦      x11  x10  x9    x8   x7   x6          ¦   x5    ¦
     ¦                                            ¦         ¦
     +------>------------------->------------>--+         ¦
     ¦          ¦                         ¦                 ¦
     ¦          ¦                         ¦                 ¦
     ¦          ¦                         ¦                 ¦
     ¦   +--+   ¦         +--+ +--+ +--+  ¦          +--+   ¦
     ¦   ¦  ¦   ¦  +----+ ¦  ¦ ¦  ¦ ¦  ¦  ¦ +----+   ¦  ¦   ¦
     ¦   ¦  ¦   +->¦x-or¦ ¦  ¦ ¦  ¦ ¦  ¦  +-¦x-or¦   ¦  ¦   ¦
     +->-¦  +-->---¦    +->¦  +-¦  +-¦  +->-¦    +->¦  +->+
         ¦  ¦    ¦ +----+ ¦  ¦ ¦  ¦ ¦  ¦    +----+   ¦  ¦
         +--+    ¦        +--+ +--+ +--+             +--+
          x0     ¦         x1    x2   x3              x4
                 ¦        +-----+
                 +--->----¦x-or ¦
Ser.Data In>------->------¦     +----->Ser.Data Out
(Write or Read)           +-----+     (WriteRandomized)
                                  (Read De-random-ized)
```

Figure 8.1 Randomizer Circuit

For every data block all 12 bits in the generator shall be set to 1 prior to the randomizer operation.

> Note that it is only the data in the 512 bytes data area and the control field that are "randomized" prior to encoding and writing. Preambles, postambles, block markers and CRC bytes shall not be randomized.

8.2 DATA ENCODING

Prior to the recording of the data on the tape, the randomized data and control information (plus the non-randomized CRC-bytes) shall be transformed into an encoded bit pattern according to table 8.1 (RLL 1,7 encoding).

For each byte, the most significant two (four) bits shall be encoded first, then the next two (four) and so on.The most significant data bit is always to the left in table 8.1. The most significant encoded bit is also to the left in the table. When recording, the most significant encoded bit in each byte is recorded first. X denotes an encoded bit which is ONE if the preceding encoded bit was a ZERO, but ZERO if the preceding encoded bit was a ONE. This encoding method will give a minimum of one "0" and a maximum of seven "0" 's between two ONE's.

> Note: Table 8.1 lists two exceptions to the general encoding rules. Data bit patterns 11101110 (EEHex) and 10111011 (BBHex) shall be encoded following the rules for special patterns. Note that this is not byte related, but should be treated as a sliding encoding in steps of two or four bits.
> Also note that these special patterns are only used for 11101110 (EEHex) and 10111011 (BBHex) respectively. Other patterns, like 11101111 (EFHex) or 10111010 (BAHex) patterns are encoded in the normal way, using the standard table.
> Examples:

```
1110 1011 1110 0001 --> X01010 010101 001010 100001

1110 1110 1011 1011 --> 010000 001001 010000 001010
| E   E | | B   B |

1001 1011 1011 1010 --> 010100 010000 001010 010010
     | B   B |

  | E    E |
0111 1011 1001 0000 --> X01010 000001 001000 010000

       | B    B |
1011 0110 1110 1101 --> 010101 000010 000001 010100
```

```
+---------------------------------------------------------+
|           Data        Encoded                           |
|           bits        bits                              |
|                                                         |
|           01    ----> X00                               |
|           10    ----> 010                               |
|           11    ----> X01                               |
|                                                         |
|           0001  ----> X00 001                           |
|           0010  ----> X00 000                           |
|           0011  ----> 010 001                           |
|           0000  ----> 010 000                           |
|                                                         |
|                                                         |
|         Special patterns:                               |
|                                                         |
|     EEhex:                                              |
|     1110 1110   ---->    010 000 001 001                |
|                                                         |
|                                                         |
|     BBhex:                                              |
|     1011 1011   ---->    010 000 001 010                |
|                                                         |
|     Normal Preamble, 1 byte repeating:                  |
|             ----> ...010 101 010 101...                 |
|                                                         |
|     Normal Postamble, 1 byte repeating:                 |
|             ----> ...010 101 010 101...                 |
|                                                         |
|     Low Frequency Preamble, 2 Bytes Repeating:          |
|             ----> ...010 000 000 100 000 001 000 000... |
|                                                         |
|     Low Frequency Postamble, 2 Bytes Repeating:         |
|             ----> ...010 000 000 100 000 001 000 000... |
|                                                         |
+---------------------------------------------------------+
```

Table 8.1 ENCODING TABLE, RLL 1,7 CODE

Note:
If the two last bits of the last CRC byte ends with 00, two additional bits 01 shall be added to the 00 bits before encoding. The encoded pattern is then followed by the Normal Postamble as usual.

```
  Example:
Last CRC Byte      Encoded  CRC      Postamble
                        |            |     |        |
   .. 1001 1100    ---> 010100 101000 001 0101010101
                                       |
                        Extra bits -----+
```

Examples:

```
    Data pattern              Encoded pattern
    Byte 1   Byte 0              Byte 1       Byte 0
 |<------>|<------>|       |<----------->|<----------->|

 0011 0111 0010 0001 -->  010001 000101 000000 100001

 1000 0000 0001 1111 -->  010010 000010 000100 101001

 1011 1011 1011 1111 -->  010000 001010 010101 001001

 1001 1011 1011 1001 -->  010100 010000 001010 010100

 1110 1011 1110 1011 -->  X01010 010101 001010 010101

 1111 1011 1011 1101 -->  X01010 000001 001001 001000
```

9. TRACK FORMAT

9.1 GENERAL INFORMATION

Each track is recorded sequentially, starting with track 1, then track 2, and so on. Before recording, data is grouped into blocks and blocks are then grouped into frames with 64 blocks in every frame.

This Standard operates on both a frame and a block basis. Section 9.2 gives a detailed description of frames, while section 9.3 gives a detailed description of blocks.

Two numbering methods are used to number blocks: Physical numbering and Logical numbering.

The Physical numbering relates directly to the recorded block on the tape. Each new block, regardless of its contents, is given a unique physical number. As implied by its name, the physical numbering system therefore relates directly to each recorded block on a track.

Logical numbering does not relate to the blocks physically recorded on the tape, but to the block numbering system used by the host. Very often, a host system operates with logical blocks of a different size from the fixed 512 byte blocks physically recorded on the tape. These host blocks may be either smaller or larger than the physically recorded blocks.

Host blocks may also be either fixed or variable. Fixed host blocks contain the same number of data bytes in every host block, while variable host blocks may contain a different number of data bytes for every host block.

In order to provide the host with a flexible numbering system, this standard records both a physical block number for every block recorded on the tape and a logical block number which may span more than one physical block. These two block numbers are recorded in the control field of every block.

9.2 FRAMES

9.2.1 General Information

On every track data is recorded in 512 data byte blocks. 64 blocks (52 plus 12 ECC blocks) constitute a **frame**. Therefore, each track consists of sequentially recorded frames as shown in figure 9.1.

Frame N	Frame N+1	Frame N+2	Frame N+3	Frame N+4	Frame N+5	Frame N+6

Figure 9.1 General Track Layout.

Each frame contains 64 blocks, 52 are data blocks or information blocks (file marks, cancel marks, set marks, identifier, control or filler blocks), and 12 are ECC blocks.

Frames are numbered indirectly, by using the 18 most significant bits of the Physical Block Address.

The frame operation is controlled by the recording drive and is normally invisible to the host. The purpose for the use of frames is to control the error correction operations.

Frames may be overwritten with new data frames or EOD markers. Append operations may only begin at EOD (End of Data).

Underrun is not allowed in the middle of a frame, regardless of the frame type.

A frame which cannot be completed on one track shall be rewritten in its entirety at the beginning of the following track, i.e. frames will NOT be split around corner turns.

The first frame on track 1 contains only the appropriate Identifier Blocks (plus the twelve ECC blocks).

9.2.2 Frame Layout.

The number of physically recorded blocks within a data frame is always 64. 52 of these blocks are used as Data or Information Blocks and the remaining 12 are ECC (Error Correction Code) blocks.

Each block contains 512 data or information bytes.

The general layout of a frame is shown in figure 9.2.

Data Block 0	Data Block 1	Data Block 2	......	Data Block 51	ECC Block 0	.....	ECC Block 11

Figure 9.2 General Frame Layout.

9.2.3 Frame Types

There are 3 different types of frames:

- o Fixed Data Frame.
- o ID Frame.
- o Track ID Frame.

The Fixed Data Frame contains Data and Information blocks in addition to the normal ECC blocks.

The ID (Identifier) Frame contains only ID blocks and ECC blocks. This frame is only recorded as the first frame on track 1. The ID blocks contain Inquiry, Mode Sense and Diagnostic data as specified in section 9.3.2.

The Track ID Frame is recorded as part of the preformating process of the cartridge. See document QIC 93-84 "Preformated Mini-cartridge Standard".

9.3 BLOCK TYPES

There are 6 different types of blocks:

```
  Data Block
  Identifier Block                      --+
  CancelMark / FileMark / SetMark Block   ¦Information Blocks
  Control Block                           ¦
  Filler Block                          --+
  ECC Block
```

Information in the Frame Control Byte determines the type of block being recorded, except for ECC blocks which are recognized by their block numbers (Last physical block number byte xx11 0100 through xx11 1111). See also table 9.6.

The Data Field of the blocks always contains 512 bytes, although the number of valid data bytes in the block may be less than 512. See sections 9.3.7 and 9.6.

Information about the number of user data bytes available in each data block is recorded in the Frame Control byte of the block.

9.3.1 Data Block

The Data Block contains user data. A full Data Block contains 512 bytes, but data blocks may contain from 1 through 511 valid data bytes depending on the host selected logical block size (see section 9.5).

9.3.2 Identifier Block

The first frame on track 1 is the ID (Identifier) frame. This frame contains 52 IDENTIFIER Blocks (plus the normal 12 ECC blocks). The ID blocks contain specific host, drive and vendor information and are used to specify the recording format and drive type in the INQUIRY DATA field (see Table 9.1).

The Identifier blocks are generally invisible to the host system, but the information in the data area may be transferred to the host by special command sequences.

9.3.2.1 Identifier Block 0

Except for the first 16 bytes, this block shall contain all zero bytes. The layout is shown in figure 9.3.

First 8 Bytes	Next 8 Bytes	496 Bytes
Format ID	Manuf. ID	All Zeroes

Figure 9.3 Layout of Block 0 in ID frame.

The first 8 bytes shall contain the Format ID. These 8 bytes shall contain the ASCII code for the characters TBD as shown in figure 9.4.

The next 8 bytes shall identify the WRITING DRIVE MANUFACTURER in accordance with table J-1 of the QIC-121 VENDOR IDENTIFICATION list.

	BYTES							
	0	1	2	3	4	5	6	7
ASCII char.	Q	I	C	-	3	0	8	0
Hex. Value	51	49	43	2D	33	30	38	30

Figure 9.4 Layout of first 8 bytes of Block 0 in ID frame.

9.3.2.2 Identifier Block 1

The 512 bytes of the data area of Identifier block 1 shall contain Inquiry Data exactly as it would be reported by the drive in response to a SCSI Inquiry command. The Inquiry Data shall at least include the Standard Inquiry Data List, located from byte 0 of the data area up to maximum byte 511.

The format of the Inquiry Data recorded in ID block 1 shall be as specified in table 9.1. Unused bytes (that is unused bytes covered by the Additional Length field) shall be filled with the ASCII code for the blank character.

The Peripheral Device Type shall always be set to 01 hex. The RMB bit shall be set to one. For further information, see

QIC-121.

9.3.2.3 Identifier Block 2

The data area of Identifier block 2 shall contain MODE SENSE data. It shall be set up as if an implicit SCSI MODE SENSE command has been executed where the returned data is moved into the 512 bytes of the data area of Identifier block 2 (and not to the SCSI bus as usual). This data shall be formatted as shown in figure 9.5.

Byte 0	Header List
	Block Descriptor List
	Page Descriptor Lists in ascending order of Page Code

Figure 9.5 Layout of Identifier Block 2.

Byte	Bits							
	7	6	5	4	3	2	1	0
0	Peripheral Qualifier			Peripheral Device Type				
1	RMB	Reserved						
2	ISO Version		ECMA Version			ANSI-Approved Version		
3	AENC	TrmIQP	Reserved		Response Data Format			
4	Additional Length (n-4)							
5	Reserved							
6	Reserved							
7	RelAdr	WBus32	WBus16	Sync	Linked	Res.	CmdQue	SftRe
8 . . 15	(MSB) . . (LSB) Vendor Identification							
16 . . 31	(MSB) . . (LSB) Product Identification							
32 . . 35	(MSB) . . (LSB) Product Revision Level							
36 . . 55	(MSB) . . (LSB) Vendor Specific							
56 . . 95	(MSB) . . (LSB) Reserved							
96 . 511	Vendor-Specific Parameter Bytes							

Table 9.1 Inquiry Data Format

Header List

This is the standard MODE SENSE Header List from QIC-121 and shown in table 9.2. Note that the first part of the Header List (the Mode Sense Data Length) is set to specify the total length of the following sense data including all Page Descriptor Lists. The Medium Type field shall indicate the current cartridge type. The Write Protect (WP) bit shall be set to zero. The Speed field shall be set to zero. The Block Descriptor Length shall be set to 8 to indicate that a single Block Descriptor List follows next. See QIC-121 table 7-61 for further details.

Byte	Bits							
	7	6	5	4	3	2	1	0
0	Mode Sense Data Length							
1	Medium Type							
2	WP	Buffered Mode			Speed			
3	Block Descriptor Length							

Table 9.2 Header List

Block Descriptor List

This is the standard MODE SENSE Block Descriptor List shown in table 9.3. The Density Code shall be set to **TBD** to indicate that the current tape format. See QIC-121 table 9-22 for further details.

Page Descriptor Lists

The various Page Descriptor Lists supported by the device follows in ascending order of page code. They shall be formatted according to table 9.4. See QIC-121 table 7-64 for further details.

Byte	7	6	5	4	3	2	1	0
0	Density Code							
1	(MSB)							
2	Number of Blocks							
3	(LSB)							
4	Reserved							
5	(MSB)							
6	Block Length							
7	(LSB)							

Table 9.3 Block Descriptor List

Byte	7	6	5	4	3	2	1	0
0	PS	Resv.	Page Code					
1	Page Length							
2-n	Mode Parameters							

Table 9.4 Page Descriptor List.

9.3.2.4 Identifier Blocks 3 and 4

The data area of block 3 and 4 are reserved for additional mode data.

9.3.2.5 Identifier Block 5

The data area of Identifier block 5 shall be reserved for Vital Product Data. If Vital Product Data are recorded this block shall at least include the Supported Vital Product Data Page. See figure 9.6. This page shall start at byte 0 in the data area of Identifier block 5 and may be followed by other VPD pages in ascending order of Page Code. Unused bytes shall be filled with blank characters. The format of the Vital Product Data pages shall be as specified in QIC-121 (tables 7-76 and 7-77 and related text).

Byte 0	Summary of Supported Pages (VPD Identifier Ooh)
	Vital Product Data Pages in ascending order of Page Code

Figure 9.6 Layout of Identifier Block 5.

If any Vital Product Data is included the first VPD page shall be the Supported Product Data Page (VPD Page Code 00h). This page holds a list of VPD Page Codes for the VPD pages to follow. The format of this page is specified in table 7-76 in QIC-121.

Vital Product Data Pages

Vital Product Data Pages are appended in ascending order of VPD code. The format of these pages shall be as specified in QIC-121.

9.3.2.6 Identifier Blocks 6 - 15

The data area of blocks 6 through 15 shall be reserved for diagnostic data at the discretion of the drive manufacturer.

9.3.2.7 Identifier Blocks 16 - 51

These blocks are reserved for future use. Each byte in the data area of these blocks shall contain 00Hex.

9.3.3 Filemark Block.

Filemark Blocks are physical blocks written to tape in response to a host "Write Filemarks" command. Filemarks are logical blocks with each Filemark assigned a unique logical block address.

9.3.4 Setmark Block.

Setmark Blocks are physical blocks written to tape in response to a host "Write Setmarks" command. Setmarks are logical blocks with each Setmark assigned a unique logical block address. Setmarks provide a segmentation scheme hierarchically superior to Filemarks.

9.3.5 Cancelmark Block.

Cancelmark Blocks are physical blocks written to tape under firmware control. Cancelmarks are used to "cancel" or negate one or more Filemarks and/or Setmarks when overwritten by host command. The Cancelmark must be the first block in frame immediately following the frame containing the Filemarks and/or Setmarks it cancels.

A Filemark or Setmark may be canceled only if there are no data blocks between it and EOD. A Cancel Mark cancels only the File Mark or Set Mark immediately preceding it.

The logical block number of the Cancelmark shall be the same as the Filemark or Setmark it cancels. The next logical block on tape shall also have the same logical address as the canceled Filemark or Setmark and the Cancel Mark.

9.3.6 Filler Block

A filler block contains no valid information in the data area.

9.3.7 Control Block

The use of Control Blocks is not defined in this Standard. However, all drives compatible with this Standard shall be able to recognize and at least pass over a Control block. The Data Area of the Control block may contain vendor unique information.

NOTE : Filler Blocks and Control Blocks may not be interspersed between physical blocks containing data of any logical block which spans more than one physical block (i.e. filler blocks and control blocks may occur only between logical blocks). Within a frame, filler blocks may never precede a data, filemark, setmark, or cancel mark block.

9.3.8 ECC Block

The ECC Block contains error correction parity bytes which may be used during a subsequent read operation where one or more data blocks cannot be read correctly. The ECC blocks are recognized by the 6 least significant bits of the physical block number (Range 11 0100 through 11 1111).

9.3.9 Fixed and Variable Logical Blocks
(See also section 9.6.)

From a host point of view, logical blocks may be written as fixed or variable. The sequence of commands required to write fixed blocks is different from the command sequence required to write variable blocks. This standard allows the host to mark each logical (host) block as fixed or variable.

The physical recorded blocks on the tape are always fixed in length, containing 512 bytes of data. However, a logical block may contain a number of valid bytes different from 512 (either more or less). This format standard makes it possible to distinguish between fixed and variable logical (host) blocks and between logical blocks containing 512 bytes of data and logical blocks containing either less than 512 or more than 512 bytes of data.

The blocks are numbered in two ways: A physical numbering system numbering every block sequentially, regardless of block type and track number, and a logical numbering system which only numbers valid logical (host) data blocks, filemarks, and setmarks (sequentially). The physical numbering system starts with 0 for the first block on track 0, and then increments this number by one for each new block, regardless of track number and block type. The logical numbering system starts with 0 for the first valid host data block, filemark, or setmark recorded on track 1, and is then incremented by one for each new host data block, filemark, or setmark recorded, regardless of track number. Frames are numbered indirectly, by using the 18 most significant bits of the Physical Block Address.

9.4 BLOCK FORMAT

9.4.1 General Layout

All blocks have the same basic layout as shown in figure 9.7:

Pre-amble	Block Marker	Control Field	Data	CRC	Post-amble

Figure 9.7 Layout of a block

All sections of a block are recorded continuously without any erased gaps between the sections. All blocks within a frame are also always recorded continuously without any erased interblock gaps. See Section 5.8 on overwrite. Frames are also recorded continuously, except during append operations.

During this operation, a short area with erased or damaged recording may occur between the end of the postamble of one frame and the preamble of the next frame due to the write current turn on time. This area shall always be shorter than the length of the recording of one byte of data.

No underrun or append operations are allowed in the middle of a frame write operations. Append operations and underruns are only allowed at frame boundaries.

9.4.2 Preamble

The preamble consists of either the fixed Normal Preamble pattern ..010 101 010.. or a combination of the fixed 2-byte Low Frequency Preamble pattern ..010 000 000 100 000 001 000 000 .. with the fixed Normal pattern. See table 8.1.

The decoding system must be able to distinguish between a preamble and any possible combinations of data patterns.

There are two types of preambles: NORMAL and LONG.

A **Normal Preamble** shall contain a minimum of 13 and a maximum of 30 bytes of the normal preamble pattern (..010 101 010 101..) as described in table 8.1. This preamble shall be recorded at the beginning of every block. It may be preceded by other types of preambles, depending upon the type of block or frame being recorded. To achieve maximum capacity, it is recommended to use the minimum length of 13 bytes whenever possible.

The Normal Preamble (..010 101 010..) shall be used to synchronize the phase locked loop or a similar circuit to the frequency and the phase of the data signal. It shall also be used to measure the average signal amplitude. The Normal Preamble may recorded as the only preamble at the beginning of a block or in a combination with other types of preambles.

The **Long Preamble** shall contain a minimum of 4200 and a maximum of 6000 recordings of the 2-byte Low Frequency Preamble pattern (.. 010 000 000 100 000 001 000 000 ..). It is not recorded alone, but always followed by a Normal Preamble. This preamble shall be recorded at the beginning of the first block on every track.

9.4.3 Block Marker

The Block Marker marks the start of a new frame or block. It contains 24 encoded bits immediately following the Normal Preamble pattern:

010 101 010 000 000 100 000 010

The left bit is the most significant bit, recorded first.

9.4.4 Control Field

The control field consists of 8 bytes, numbered 0 to 7. These bytes contain information pertaining to the block they precede, or to the frame that block resides in. The organization of the control bytes are different for the first 52 blocks of each frame than for the ECC blocks.

Control bytes 0-7 are encoded according to the rules in section 8. Byte 7 is recorded first, followed by byte 6 and so on. The most significant encoded bit in each byte is recorded first.

9.4.4.1 Control Field, Data and Information blocks

Figure 9.8 shows the general layout of the control field for data and information blocks. Control byte 0 (Block Control Byte) specifies the type of block being recorded. Control bytes 1-3 specify the logical block address. Control byte 4 contains the track number and the Early Warning indicator. Control bytes 5-7 specify the physical block address.

Control Bytes 5-7	Control Byte 4	Control Bytes 1-3	Control Byte 0
Physical Block Number	Track Number	Logical Block Address	Block Control Byte

Figure 9.8 Layout of Control Field, first 52 blocks.

Control Byte 0							
7	6	5	4	3	2	1	0
Reserved	Fixed	BOLB	EOLB	Block Type			

Figure 9.9 Layout of Control Byte 0.

Bit 7 in Control Byte 0 are reserved for future use.

Bit 6 in Control Byte 0 specifies whether the logical host block is fixed or variable. Bit 5 is set to 0 for fixed blocks, to 1 for variable blocks.

Bits 5 and 4 in Control Byte 0 deal with Host logical blocks that are larger than the physical blocks size of 512 bytes. This is shown in table 9.5.

Note that a host block may contain a number of data bytes different from the fixed 512 bytes in the physical blocks.

Bits 5 and 4 are used to mark the beginning and the end of logical host blocks. Bit 5 (BOLB) is set to 1 in the first physical block of the host block. In the same way, bit 4 (EOLB) is set to 1 in the last physical block containing data from the same logical host block. Bits 5 and 4 shall be set to 0 for all other physical blocks within the same logical block.

Example: Logical block is 4200 bytes long (Variable Mode Write).
The first physical block will contain the first 512 bytes of this logical block and have bits 6 and 5 both set to 1, bit 4 set to 0. The next 7 physical blocks will have bit 6 set to 1 and bits 5 and 4 set to 0. The next physical block will have bits 6 and 4 set to 1 and bit 5 to 0. This block will contain the last 104 bytes of the logical block, plus 408 padding (filler) bytes.

The four least significant bits of Control Byte 0 are used to indicate the type of block being recorded. The coding of these four bits is shown in table 9.6. There are two different codes to indicate a Full Data Block, 0000 and 1111. The code 1111 is be used to indicate the entire frame is full of data, with no padding or partially filled blocks, and that every logical block in the frame is of equal size.

Control Byte 0 is the only control byte covered by ECC protection.

Control Byte 0	
Bits 5 4	Description
1 0	This physical block marks the beginning of a fixed or a variable logical host block. This block and the next physical block are part of the same logical host block.
0 0	This physical block is neither the first nor the last physical block within the fixed or variable logical host block. This block and the next physical block is part of the same logical block.
0 1	This physical block is the last physical block within the fixed or variable logical host block. It may contain a maximum of 512 valid data bytes.
1 1	This physical block marks both beginning and the end of a fixed or variable logical host block. The host block therefore contains a maximum of 512 valid data bytes.

Table 9.5 Fixed/Variable Logical Host Block Indicator.

Control Byte 0		
Bits 3 2 1 0	Block Type	Comments
0 0 0 0 and 1 1 1 1	Full Data Block	This block contains 512 bytes of valid data. This block may be part of a logical host block either fixed or variable.
0 0 0 1	Data Block with 1 - 255 valid data bytes. End of a logical host block.	This block contains from 1 to 255 bytes of valid data. A host logical block ends with this block. The remaining number of data bytes in the host block is between 1 and 255 (see section 9.6).
0 0 1 0	Data Block with 256-511 valid data bytes. End of a logical host block.	This block contains from 256 to 511 bytes of valid data. A host logical block ends with this block. The remaining number of data bytes in the host block is between 256 and 511 (see section 9.6).
0 0 1 1	Reserved.	
0 1 0 0	Filemark Block.	No valid information in data area.
0 1 0 1	Setmark Block.	No valid information in data area.
0 1 1 0	Cancel Mark Block.	No valid information in data area.
0 1 1 1	Reserved.	
1 0 0 0	Filler Block.	No valid information in data area.
1 0 0 1	Control Block	This optional block may be used for vendor specific operations.
1 0 1 0	Identifier Block	Contains special drive and host data. See text.

Table 9.6 Encoding of Block Type Control Bits.

Control bytes 1-3 are used to specify the Logical Block Address as shown in figure 9.10. Byte 3 is the least significant byte of the logical address, byte 1 is the most significant byte of the logical address. This address specifies the number of logical (host) data blocks that are recorded on the tape. The address starts with 000000Hex and is incremented by one for each new logical data block, filemark, or setmark recorded until a maximum of FFFFFEHex. A logical block may consist of one or more physical blocks. See section 9.6.

Logical block numbering is only used for data blocks, filemarks, setmarks, and cancelmarks. Filler blocks and Control Blocks are recorded with the last known logical block number. Identifier blocks are not given a unique logical block number, but are recorded with Control bytes 1, 2 and 3 always set to FFFFFFHex.

Logical Block Address		
Control Byte 3	Control Byte 2	Control Byte 1
LSB		MSB

Figure 9.10 Logical Block Address

Figure 9.11 shows the layout of Control Byte 4 in the Control Field.

7	6	5	4	3	2	1	0
EW	**TRACK NUMBER**						

Figure 9.11 Layout of Control Byte 4.

Bit 7 in Control Byte 4 shall be set to 1 for all blocks recorded past Pseudo (Logical) Early Warning as defined by the drive. A zero in this field indicates a block recorded prior to the Early Warning for the partition in which the block is recorded.

Bits 0-6 of control byte 4 contain the track address. This track address is the physical track number as specified in section 4.

Control Bytes 5-7 are used to specify the physical block number. This is shown in figure 9.12. The block numbering starts with 000000 Hex and is incremented by one for each new block recorded, regardless of block type, frame type or track number. The block numbering is not reset at the start of a new track.

Physical Block Address		
Control Byte 3	Control Byte 2	Control Byte 1
LSB		MSB

Figure 9.12 Physical Block Address

9.4.4.2 Control Field, ECC blocks

Figure 9.13 shows the general layout of the control field for ECC blocks. Control byte 0 holds the codes to correct errors in the control byte 0 of the first 52 blocks in the frame. Control bytes 1-3 specify the number of times the tape has been recorded from the Beginning of the tape. Control byte 4 contains the track number and the Early Warning indicator. Control bytes 5-7 specify the physical block address.

Control Bytes 5-7	Control Bytes 4	Control Bytes 1-3	Control Byte 0
Physical Block Number	Track number	Number of writes from BOP	ECC for Control Byte 0

Figure 9.13 Layout of Control Field, ECC blocks.

Control Byte 0 of each of the first 52 blocks in each frame is protected by ECC. The correction codes for these bytes are recorded in Control Byte 0 of the ECC blocks.

Control Bytes 1-3 contain a count of how many times the tape has been recorded from Load Point. Each time a write operation is started from the Beginning of Partition zero this count is incremented and the new value is recorded in each frame. This distinguishes left-over frames from a previous write pass from the valid frames of a new write pass. When reading a tape, any frame with a different value in this field from the first frame in partition 0 is not a valid frame. Byte 3 holds the least significant byte of the count, byte 1 holds the most significant byte.

Control Byte 4 in the ECC blocks is identical to Control Byte

4 in each of the first 52 blocks in the frame.

Control Bytes 5-7 contain the Physical Block Address. These bytes are used the same as in the first 52 blocks in the frame.

9.4.5 Data Field

The Data Field contains 512 bytes of data, encoded according to the rules in section 8.1 and 8.2. The contents of the data field depends upon the type of block being recorded:

* Data — All 512 bytes are available for user data.

* Identifier: — Contains drive and/or host valid information in its data field.

* Filler,
* Control: — Contains no valid data in its data field. May contain vendor unique information.

* Filemark,
* Setmark,
* Cancelmark: — Contains no valid data in its data field. May contain vendor unique information.

* ECC Block: — Contains error correction characters generated by the drive.

9.4.6 CRC FIELD

The CRC (Cyclic Redundancy Check) field consists of 4 bytes calculated over the whole data block area and control field area, starting with the most significant bit of byte 7 in the Control Field ending with the least significant bit of byte 511 in the Data Field.

All calculations are done prior to the data randomizing and encoding. All 32 bits in the CRC character shall be set to ONE prior to the start of the CRC calculation. The generating polynomial shall be:

$$x^{32} + x^{28} + x^{26} + x^{19} + x^{17} + x^{10} + x^{6} + x^{2} + 1$$

The four bytes shall be encoded according to the rules in section 8.2 prior to the recording. No randomizing shall take place on the CRC characters. The most significant bits of the most significant byte shall be recorded first. It is a possibility that the two last bits of the last CRC byte both are 0's. In this case 01 shall be appended to the two 0's to enable correct encoding. After decoding, these two extra bits have no significance. See sections 8.2 and 9.4.7.

9.4.7 Postamble

The postamble consists of either the fixed Normal Postamble pattern ..010 101 010 101.. or a combination of this pattern with the Low Frequency Postamble pattern ..010 000 000 100 000 001 000 000 ...See table 8.1.

The decoding system must be able to distinguish between a postamble and any possible combinations of data patterns.

The Postamble is recorded at the end of each block immediately following the CRC bytes.

There are two different types of postambles, NORMAL and ELONGATED.

Additionally, a write operation is terminated by writing a special EOD Marker.

A **Normal Postamble** shall contain a minimum of 1 and a maximum of 2 bytes of the Normal Postamble pattern (..010 101 010 101..) as described in table 8.1. This postamble shall always be recorded at the end of each block. It may be followed by either an Elongated Postamble, or it may be followed by a new preamble.

To achieve maximum capacity, it is recommended to use the minimum length of 1 byte whenever possible.

After writing this postamble, the following preamble or postamble shall be recorded so that there is no phase shift error or transition glitches between the end of the postamble and the beginning of the next preamble/postamble.

If the last two bits of the last CRC byte are 0, the encoding (and decoding) shall add 01 to the CRC bits before encoding. In this case the coding of the last two bits of the CRC character and the postamble will give the following pattern:

```
                          CRC|    | Postamble   |
   .... ..00 Postamble --> X00 001 010 101 010 101
                                 |
                                 |
     Inserted 01 bits encoded --+
```

An **Elongated Postamble** consists of between 6600 and 7200 recordings of the 2-byte Low Frequency Preamble pattern (.. 010 000 000 100 000 001 000 000 ..). This Elongated Postamble will always follow a Normal Postamble.

The **EOD Marker** shall be recorded at the end of every write operation as described in section 9.4.8. The EOD Marker shall consist of a minimum of 64K (65536) recordings of the 2-byte Low Frequency Postamble pattern:

(..010 000 000 100 000 001 000 000 ..)

See figure 9.14 The EOD Marker shall be recorded following the normal postamble.

Table 9.7 shows a summary of the use of preambles and postambles for different types of recording situations.

Recording Type	Preamble	Postamble
Normal data block in a fixed frame	Normal	Normal
Underrun or write termination	Normal	Normal + EOD Marker
Append operation + first block in EOD	Normal	Normal
First block on a track	Long + Normal	Normal
Last block on a track	Normal	Normal

Table 9.7 Summary of Preambles/Postambles Recordings

9.4.8 Underrun

Underrun is only allowed to occur at frame boundaries. When an underrun situation occurs, after the final data block has been written, data shall be terminated with a normal Postamble and an EOD Marker to designate the End of Recorded Area.

Last Block at End of Valid Data	Normal Postamble	EOD Marker 65536 recordings of Low Frequency Postamble

Figure 9.14 End of Data Marker at End of Valid Data Area

If the end of track (LP or EW hole) is detected while writing the EOD marker, the recording drive shall ensure that the current track is terminated such that the last data block on the track has been followed by at least the Normal Postamble and one Elongated Postamble. A complete EOD marker preceded by a Long Preamble shall then be written on the next track (provided that the current track is not the last one).

If End-Of-Partition is encountered the Elongated Postamble which normally occurs at the end of a track shall be sufficient to mark the End Of Recorded Data.

9.4.9 Erase and Append Operations at EOD

When a data frame is appended after an underrun, the recording shall begin at a point between the first 12 to 12150 2-byte recordings of the EOD Marker. The append operation shall start with a Normal Preamble. See figure 9.15.

```
------------------------------|||||-----------
 Last  ¦ Normal    ¦ Residue of|||| Normal    ¦
 Block ¦ Postamble¦ Old EOD   |||| Preamble  ¦
       ¦           ¦ Marker    ||||           ¦
------------------------------|||||-----------
                  |          |    |
                  |<---Te--->|    |
               -->|               |<---To
```

Te = Minimum 12 times 2-byte Low Frequency Elongated Postamble Residue (EOD Marker) not overwritten by the Normal Preamble.

To = Maximum 12150 times 2-byte Low Frequency Elongated Postamble Residue (EOD Marker) not overwritten by the Normal Preamble.

Figure 9.15 Postamble/preamble overlap after append operation.

9.4.10 Erase and Write operation at BOP

Each time a write operation is started from the Beginning of Partition zero, the Use Count from the previous write is incremented. To effectively erase all of the data on a tape, a new ID frame is written at BOT of partition 0 with a higher Use Count than the previously written data. To erase the directory partition, EOD Marker is recorded at BOP. See section 9.4.4.2 for further details.

9.6 FIXED AND VARIABLE BLOCKS

This standard allows for the recording of both fixed and variable host blocks. The physical blocks recorded on the tape contain always 512 data bytes, however, some of these data bytes may not be valid in every block, as provided in Section 9.4.4.

9.6.1 Fixed/Variable Host Block Indicator

If the host specifies that one or more fixed blocks shall be recorded (regardless of block size), bit 6 in Control byte 0 of every physical block used to record the data from these fixed host blocks shall be set to 0. If the host specifies that variable blocks shall be recorded (regardless of block size), bit 6 in Control byte 0 of every physical block used to record the data from these variable host blocks shall be set to 1.

9.6.2 Fixed/Variable Host Blocks, 512 Data Bytes

In this case, the host block size is the same as the size of the physical recorded block. Bit 6 in Control Byte 0 is set to 0 (fixed) or 1 (variable) and bits 0-3 in the same byte is also set to 00Hex. All data bytes in the recorded block are valid. Bits 4 and 5 in Control byte 0 will be set to 1 for every physical block recorded, to indicate the both the start and the end of a new host (logical) block.

9.6.3 Fixed/Variable Host Blocks, <256 Data Bytes

In this case, the host block size is less than the size of the physical recorded block. Bit 6 in Control Byte 0 is set to 0 (fixed) or 1 (variable) and bits 0-3 in the same byte is set to 1Hex. The number of valid data bytes in the block is specified by the Valid Byte Counter, byte 511 of the data area. This byte contains a number from 1 to FFHex (255Dec) depending upon the number of valid data bytes. The valid data bytes are always recorded first in the data field, then come filler bytes (no value specified in this standard) and finally, as the last byte, the Valid Byte Counter. See figure 9.16.

Bits 4 and 5 in Control byte 0 will be set to 1 for every physical block recorded, to indicate the start and end of a new host (logical) block.

```
+-----------------------------------------------------+
¦                Data Field 512 Bytes                 ¦
+-----------------------------------------------------¦
¦   Valid Data      ¦ Filler Bytes      ¦ Valid Byte  ¦
¦   Bytes           ¦ Unspecified       ¦ Counter     ¦
¦                   ¦                   ¦ (Byte 511)  ¦
+-----------------------------------------------------+
```

Figure 9.16 Layout of Data Field, Physical Variable Data Blocks.

9.6.4 Fixed/Variable Host Blocks, 256 - 511 Data Bytes

In this case, the host block size is still less than the size of the physical recorded block. Bit 6 in Control Byte 0 is set to 0 (fixed) or 1 (variable) and bits 0-3 in the same byte are set to 2Hex. The number of valid data bytes in the block is specified as 256 plus the number specified by the Valid Byte Counter, byte 511 of the data area. This byte contains a number from 0 to FFHex (255Dec) depending upon the number of valid data bytes. The valid data bytes are always recorded first in the data field, then come filler bytes (no value specified in this standard) and finally, as the last byte, the Valid Byte Counter. See Figure 9.16.

A host block containing 392 bytes of valid data will therefore be recorded with bits 0-3 in Control Byte 0 set to 02Hex and byte 511 of the data field set to 88Hex (136Dec). This gives us 256 + 136 = 392 valid data bytes in the block (which means data bytes from 000 to 391 in the data area).

Bits 4 and 5 in Control byte 0 will be set to 1 for every physical block recorded, to indicate the start and end of a new host (logical) block.

9.6.5 Fixed/Variable Host Blocks, >512 Data Bytes in Multiples of 512.

In this case, the host block size is greater than the physical block size, but the host blocks are specified as Nx512 where N is 2, 3, etc. Therefore all the physical blocks recorded to cover one host block contain only valid data bytes.

As an example, we will assume a fixed host block of 2048 bytes. This will therefore require 4 physical blocks. Bit 6 in Control Byte 0 is set to 0 in all 4 physical blocks. Bit 5 in the same byte is set to 1 for the first physical block and 0 in the next three blocks. Bit 4 is set to 0 for the first 3 blocks and to 1 for the last one. Bits 0-3 in Control byte 0 are set to 0Hex in all four physical blocks. See Figure 9.17.

Fixed Host Block, 2048 Data Bytes			
Physical Block No. N	Physical Block No. N+1	Physical Block No. N+2	Physical Block No. N+3
Full Data Block Partial Logical Host Block Bits 0-3 in Control Byte 0 is 0Hex Bit 6 = 0 Bit 5 = 1 Bit 4 = 0	Full Data Block Partial Logical Host Block Bits 0-3 in Control Byte 0 is 0Hex Bit 6 = 0 Bit 5 = 0 Bit 4 = 0	Full Data Block Partial Logical Host Block Bits 0-3 in Control Byte 0 is 0Hex Bit 6 = 0 Bit 5 = 0 Bit 4 = 0	Full Data Block End Logical Host Block Bits 0-3 in Control Byte 0 is 0Hex Bit 6 = 0 Bit 5 = 0 Bit 4 = 1

Figure 9.17 Fixed Host Block, 2048 Bytes.

9.6.6 Fixed/Variable Host Blocks, >512 Data Bytes, not a multiple of 512.

In this case, the host block size is greater than the physical block size, but the host blocks are specified as different from Nx512 where N is 2, 3, etc. Therefore the last of the physical blocks recorded to cover one host block contain less than 512 bytes of valid data bytes.

As an example, we will assume a fixed host block of 1027 bytes. This will therefore require 3 physical blocks. Bit 6 in Control Byte 0 is set to 0 (fixed) or 1 (variable) in all 3 physical blocks. Bit 5 in the same byte is set to 1 for the first physical block and 0 in the next two.

Bit 4 is set to 0 for the first 2 blocks and 1 in the last one. Bits 0-3 in Control byte 0 are set to 0Hex in the first two physical blocks and 1Hex for the last one. See Figure 9.18.

Figure 9.19 shows another example, with a fixed block specified as 1417 bytes long.

Fixed Host Block, 1027 Data Bytes		
Physical Block No. N	Physical Block No. N+1	hysical Block No. N+2
Full Data Block Partial Logical Host Block Bits 0-3 in Control Byte 0 is 0Hex Bit 6 = 0 Bit 5 = 1 Bit 4 = 0	Full Data Block Partial Logical Host Block Bits 0-3 in Control Byte 0 is 0Hex Bit 6 = 0 Bit 5 = 0 Bit 4 = 0	Variable Data Block 1 - 255 Data Bytes End Logical Host Block Bits 0-3 in Control Byte 0 is 1Hex Last byte in data field is 03Hex (1027 - 512 - 512 = 3) Bit 6 = 0 Bit 5 = 0 Bit 4 = 1

Figure 9.18 Fixed Host Block, 1027 Bytes.

Fixed Host Block, 1417 Data Bytes		
Physical Block No. N	Physical Block No. N+1	Physical Block No. N+2
Full Data Block Partial Logical Host Block Bits 0-3 in Control Byte 0 is 0Hex Bit 6 = 0 Bit 5 = 1 Bit 4 = 0	Full Data Block Partial Logical Host Block Bits 0-3 in Control Byte 0 is 0Hex Bit 6 = 0 Bit 5 = 0 Bit 4 = 0	Variable Data Block 256 -511 Data Bytes End Logical Host Block Bits 0-3 in Control Byte 0 is 2Hex Last byte in data field is 89Hex (=137Decimal) (1417-512-512-256 = 137) Bit 6 = 0 Bit 5 = 0 Bit 4 = 1

Figure 9.19 Fixed Host Block, 1417 Bytes.

10. ERROR CORRECTION

The ECC blocks at the end of each frame may be used during the data read operation to reconstruct blocks in error. The error correction system is based upon an interleave organization effectively separating each frame into two groups, one containing all the even numbered blocks, the other one containing all the odd numbered blocks. Each group has independent error correction. The group with the even blocks shall be referred to as the even interleave, while the group containing the odd blocks shall be referred to as the odd interleave. For each of these groups, the error correction system makes it possible to correct any combination of s blocks with CRC errors (or pointers) and t blocks with CRC failures, as long as:

$$s + 2t < 7$$

10.1 ERROR CORRECTION MATRIX FORMAT

A frame contains 64 blocks, 52 data blocks and 12 ECC parity blocks. Each block contains 512 data bytes and 1 Control Byte (Control Byte 0) which are covered by ECC control.

The bytes in the ECC frames are considered to be arranged in 32 blocks (rows) by 513 bytes as shown in figure 10.1.

The even parity rows (i.e. blocks 52, 54, 56, 58, 60 and 62) shall be chosen so that each column of the even rows (blocks 0, 2, 4, ..., 48 and 50) of the matrix forms an independent Reed-Solomon codeword of redundancy six, with 8-bit characters, as shown in Figures 10.1. Similarly, the odd parity rows (i.e. 53, 55, 57, 59, 61 and 63) shall be chosen so that each column of the odd rows (blocks 1, 3, 5, ..., 49 and 51) of the matrix forms an independent Reed-Solomon codeword of redundancy six. See figure 10.2. Data shall be written on the tape row by row, starting with row 0, and within each row (i.e. block) the bytes shall be written starting with column 0. This implementation gives a very effective interleaving of the data regarding ECC, although the data itself is recorded in the normal order received from the host. This implementation ensures that the influence of any error spreading over two neighbor blocks is reduced. Figure 10.4 shows the complete ECC frame format with all blocks.

```
       |Control|
       |Byte 0 |<----------Data Bytes------------->|
       |   0      1     2      ...       511   512 |
       +--------------------------------------------------+
     0 ¦       ¦     ¦     ¦           ¦     ¦     ¦ CRC  ¦
       +-------+-----+-----+-----------+-----+-----+------¦  D
     2 ¦       ¦     ¦     ¦           ¦     ¦     ¦ CRC  ¦  A
       +-------+-----+-----+-----------+-----+-----+------¦  T
     4 ¦       ¦     ¦     ¦           ¦     ¦     ¦ CRC  ¦  A
     . +-------+-----+-----+-----------+-----+-----+------¦
     . ¦       ¦     ¦     ¦     .     ¦     ¦     ¦      ¦
     . ¦       ¦     ¦     ¦     .     ¦     ¦     ¦      ¦  R
     . +-------+-----+-----+-----------+-----+-----+------¦  O
    48 ¦       ¦     ¦     ¦           ¦     ¦     ¦ CRC  ¦  W
       +-------+-----+-----+-----------+-----+-----+------¦  S
    50 ¦       ¦     ¦     ¦           ¦     ¦     ¦ CRC  ¦
       ¦=======¦=====¦=====¦===========¦=====¦=====¦=====¦
    52 ¦       ¦     ¦     ¦ ECC0      ¦     ¦     ¦ CRC  ¦  P
       +-------+-----+-----+-----------+-----+-----+------¦  A
    54 ¦       ¦     ¦     ¦ ECC2      ¦     ¦     ¦ CRC  ¦  R
       +-------+-----+-----+-----------+-----+-----+------¦  I
    56 ¦       ¦     ¦     ¦ ECC4      ¦     ¦     ¦ CRC  ¦  T
       +-------+-----+-----+-----------+-----+-----+------¦  Y
    58 ¦       ¦     ¦     ¦ ECC6      ¦     ¦     ¦ CRC  ¦
       +-------+-----+-----+-----------+-----+-----+------¦  R
    60 ¦       ¦     ¦     ¦ ECC8      ¦     ¦     ¦ CRC  ¦  O
       +-------+-----+-----+-----------+-----+-----+------¦  W
    62 ¦       ¦     ¦     ¦ ECC10     ¦     ¦     ¦ CRC  ¦  S
       +--------------------------------------------------+
```

Figure 10.1 ECC Frame Format, Even Blocks

```
    |Control|
    |Byte 0 |<----------Data Bytes------------->|
    |   0      1     2      ...       511   512 |
    +--------------------------------------------------+
  1 |       |     |     |          |     |     | CRC |
    +-------+-----+-----+----------+-----+-----+-----|  D
  3 |       |     |     |          |     |     | CRC |  A
    +-------+-----+-----+----------+-----+-----+-----|  T
  5 |       |     |     |          |     |     | CRC |  A
  . +-------+-----+-----+----------+-----+-----+-----|
  . |       |     |     |    .     |     |     |     |
  . |       |     |     |    .     |     |     |     |  R
  . +-------+-----+-----+----------+-----+-----+-----|  O
 49 |       |     |     |          |     |     | CRC |  W
    +-------+-----+-----+----------+-----+-----+-----|  S
 51 |       |     |     |          |     |     | CRC |
    |=======|=====|=====|==========|=====|=====|=====|
 53 |       |     |     |  ECC1    |     |     | CRC |  P
    +-------+-----+-----+----------+-----+-----+-----|  A
 55 |       |     |     |  ECC3    |     |     | CRC |  R
    +-------+-----+-----+----------+-----+-----+-----|  I
 57 |       |     |     |  ECC5    |     |     | CRC |  T
    +-------+-----+-----+----------+-----+-----+-----|  Y
 59 |       |     |     |  ECC7    |     |     | CRC |
    +-------+-----+-----+----------+-----+-----+-----|  R
 61 |       |     |     |  ECC9    |     |     | CRC |  O
    +-------+-----+-----+----------+-----+-----+-----|  W
 63 |       |     |     |  ECC11   |     |     | CRC |  S
    +--------------------------------------------------+
```

Figure 10.2 ECC Frame Format, Odd Blocks

10.2 FIELD REPRESENTATION

GF(256) is the field consisting of 256 elements. Each field element "a" has the form:

$$a = a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

where each a_i is either 0 or 1. A field element "a" shall be represented by a byte as shown in figure 10.3.

```
  MSB                                                    LSB
+--------------------------------------------------------+
|      |      |      |      |      |      |      |      |
|  a7  |  a6  |  a5  |  a4  |  a3  |  a2  |  a1  |  a0  |
|      |      |      |      |      |      |      |      |
+--------------------------------------------------------+
```

Figure 10.3 Bit Numbering Convention

Field math operations (addition, multiplication, division) are defined to be polynomial math modulo a_n irreducible binary polynomial of degree eight, f(x), where binary addition is the logical exclusive-or operation and binary multiplication is the AND operation. The irreducible polynomial used to generate the field GF(256) shall be:

$$f(x) = x^8 + x^7 + x^2 + x + 1 .$$

```
     |Control|
     |Byte 0 |<----------Data Bytes------------->|
     |   0      1     2      ...       511   512 |
     +-------------------------------------------------+
   0 |       |     |     |           |     |     | CRC |
     +-------+-----+-----+-----------+-----+-----+-----|   D
   1 |       |     |     |           |     |     | CRC |   A
     +-------+-----+-----+-----------+-----+-----+-----|   T
   2 |       |     |     |           |     |     | CRC |   A
   . +-------+-----+-----+-----------+-----+-----+-----|
   . |       |     |     |     .     |     |     |     |
   . |       |     |     |     .     |     |     |     |   R
   . +-------+-----+-----+-----------+-----+-----+-----|   O
  50 |       |     |     |           |     |     | CRC |   W
     +-------+-----+-----+-----------+-----+-----+-----|   S
  51 |       |     |     |           |     |     | CRC |
     |=======|=====|=====|===========|=====|=====|=====|
  52 |       |     |     |  ECC0     |     |     | CRC |   P
     +-------+-----+-----+-----------+-----+-----+-----|   A
  53 |       |     |     |  ECC1     |     |     | CRC |   R
     +-------+-----+-----+-----------+-----+-----+-----|   I
  54 |       |     |     |  ECC2     |     |     | CRC |   T
   . +-------+-----+-----+-----------+-----+-----+-----|   Y
   . |       |     |     |   .       |     |     | CRC |
   . |       |     |     |   .       |     |     |     |   R
   . +-------+-----+-----+-----------+-----+-----+-----|   O
  62 |       |     |     |  ECC10    |     |     | CRC |   W
     +-------+-----+-----+-----------+-----+-----+-----|   S
  63 |       |     |     |  ECC11    |     |     | CRC |
     +-------------------------------------------------+
```

Figure 10.4 ECC Full Frame Format.

10.3 CODE GENERATOR POLYNOMIAL

Let r be a root of f(x); in hex notation, r=2hex. The generator polynomial for the Reed-Solomon code shall be of the form:

$$g(x) = (x+r^0)(x+r^1)(x+r^2)(x+r^3)(x+r^4)(x+r^5)$$

$$= x^6 + 3Fx^5 + 28x^4 + A6x^3 + 12x^2 + 56x + F4$$

where r = 2. This Reed-Solomon code contains the QIC-525 polynomial as a sub-code.

Encoding shall be accomplished as follows. In a full frame, each column contains data bytes d0 to d63, numbered as in figure 10.4. The parity bytes d52 to d63 in each column shall be chosen so that the two polynomials:

$$do(x) = \sum_{i=0}^{31} d_{(62-2i)}x^i$$

and

$$do(x) = \sum_{i=0}^{31} d_{(63-2i)}x^i$$

(Σ = SUM)

are each divisible by g(x), using polynomial division over GF(256).

10.4 EXAMPLES OF CODEWORDS

The following columns of bytes are codewords for the polynomials defined in the preceding sections, using hex notation for the field elements.

Data Blocks Number	Byte number 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	...
1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	...
.	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	...
.	..	..	..	..	..	..(all zeros).					..	..	..	..	..	..	...
42	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	...
43	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	...
44	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	...
45	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	...
46	00	00	00	00	00	00	01	00	01	01	00	01	00	00	00	00	...
47	00	00	00	00	00	00	00	01	01	00	01	01	00	00	00	00	...
48	00	00	00	01	00	01	00	00	00	01	00	01	FF	00	FF	00	...
49	00	00	00	00	01	01	00	00	00	00	01	01	00	FF	FF	00	...
50	01	00	01	00	00	00	00	00	00	01	00	01	00	00	00	00	...
51	00	01	01	00	00	00	00	00	00	00	01	01	00	00	00	00	...

Parity Blocks number	Byte number 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
52	3F	00	3F	6F	00	6F	ED	00	ED	BD	00	BD	0A	00	0A	00	...
53	00	3F	3F	00	6F	6F	00	ED	ED	00	BD	BD	00	0A	0A	00	...
54	28	00	28	A2	00	A2	A9	00	A9	23	00	23	5E	00	5E	00	...
55	00	28	28	00	A2	A2	00	A9	A9	00	23	23	00	5E	5E	00	...
56	A6	00	A6	80	00	80	97	00	97	B1	00	B1	2C	00	2C	00	...
57	00	A6	A6	00	80	80	00	97	97	00	B1	B1	00	2C	2C	00	...
58	12	00	12	D6	00	D6	4C	00	4C	88	00	88	77	00	77	00	...
59	00	12	12	00	D6	D6	00	4C	4C	00	88	88	00	77	77	00	...
60	56	00	56	7E	00	7E	53	00	53	7B	00	7B	33	00	33	00	...
61	00	56	56	00	7E	7E	00	53	53	00	7B	7B	00	33	33	00	...
62	F4	00	F4	E4	00	E4	CD	00	CD	DD	00	DD	C3	00	C3	00	...
63	00	F4	F4	00	E4	E4	00	CD	CD	00	DD	DD	00	C3	C3	00	

Table 10.1 Example of Codewords