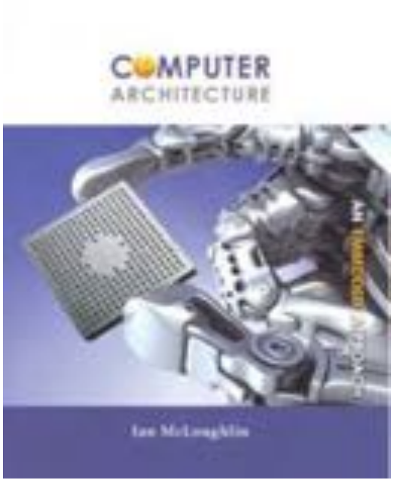
Computer Peripherals

School of Computer Engineering
Nanyang Technological University
Singapore

These notes are part of a 3rd year undergraduate course called "Computer Peripherals", taught at Nanyang Technological University School of Computer Engineering in Singapore, and developed by Associate Professor Kwoh Chee Keong. The course covered various topics relevant to modern computers (at that time), such as displays, buses, printers, keyboards, storage devices etc... The course is no longer running, but these notes have been provided courtesy of him although the material has been compiled from various sources and various people. I do not claim any copyright or ownership of this work; third parties downloading the material agree to not assert any copyright on the material. If you use this for any commercial purpose, I hope you would remember where you found it.

Further reading is suggested at the end of each chapter, however you are recommended to consider a much more modern alternative reference text as follows:



Computer Architecture: ***an embedded approach***
Ian McLoughlin
McGraw-Hill 2011

Chapter 7. Magnetic Recording Fundamentals

Computer memories are divided into two types. Main or working memories including RAM's, ROM's and other semiconductor devices which are directly addressable by the CPU, and mass storage devices. Mass storage memories generally are not directly addressable by the CPU, have a slower access method, are non-volatile and often removable off-line and have a much lower cost per bit. Mass storage devices may be classified into magnetic hard disks and floppy disks, several types of magnetic tape drives, and various kinds of optical disk drives. In additional, there are the many varieties of card storage devices like magnetic cards, smart cards, ROM, RAM and flash memory cards, but these will not be discussed here. Except for the CD-ROM in which data is represented by physical indentations or pits, practically all other write-able mass storage devices make use of the magnetic or magneto-optical (M-O) characteristics of the recording media to store the required information. Thus we shall first consider some basic principles used in magnetic recording and the performance of the digital recording channel.

7.1 Magnetic Material Characteristics

When a piece of magnetic material is moved past a magnetic field, usually created by an electromagnet, it becomes magnetised. Similarly, when the magnetised material is moved past an unenergised coil, it induces a voltage across the coil. This is just Faraday's law of electromagnetic induction, which relates the voltage induced to the magnetic field strength. A number of parameters are used to characterise magnetic materials:

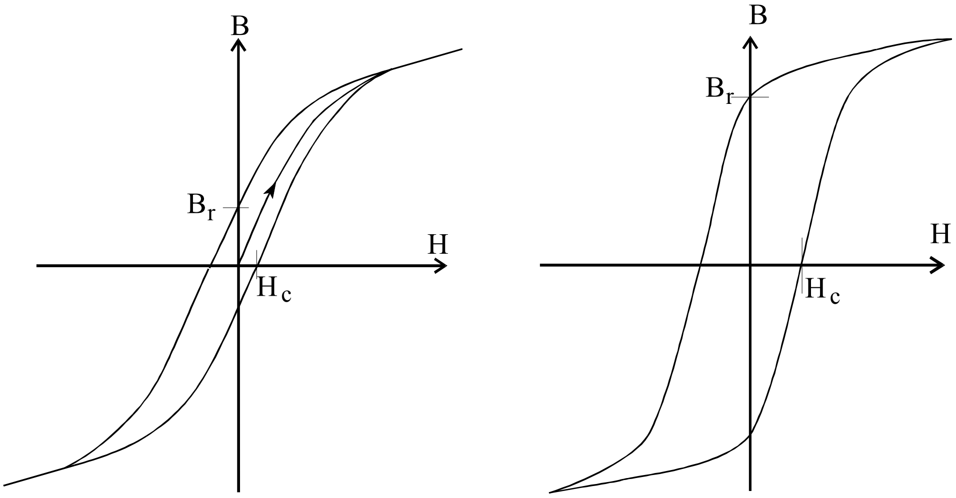
(a) *Coercivity* H_c is the measurement the level of difficulty to magnetise the material. For storage media, a high coercivity is desired in order that information stored will be preserved in the presence of stray magnetic fields that may be present. High coercivity also implies that a strong magnetic field is needed to record information onto it. Magnets with high coercivity are called *hard* magnets.

(b) *Remanence* B_r is the amount of magnetisation that remains after the magnetic field is removed. Soft iron used for electromagnets are chosen to have low remanence so that it will respond efficiently to the applied electromagnetic field.

(c) *Magnetic domains* are small regions in the magnetic media which may be magnetised independently of adjacent regions, so that adjacent domains can have opposite polarities. The size or granularity of these domains have an important bearing on the density of information that can be stored.

(d) *Flux reversal* occurs when a change in polarity is encountered while moving from one domain to the next. The storage density of the media is measured by the flux reversal per inch (frpi) or the flux change per inch (fcpi).

Figure 0-1 shows the B-H characterisation curves for soft and hard magnetic materials.



SOFT MAGNET
$H_c < 10$ Oe

HARD MAGNET
$H_c > 100$ Oe

Figure 0-1. B-H magnetisation curves

7.2 Read/Write head

The "guts" of a magnetic recording system are: the write head, the magnetic medium, and the read head. (The write head could be the same as the read head and usually has been the case for disk drives.) The write head is driven by a current source that carries the information to be stored. The write head radiates flux, which changes the state of magnetization of the magnetic medium immediately under the head. Actually, since the head is moving with respect to the magnetic medium, any point on the magnetic medium retains the state of magnetization corresponding to the last flux it experienced from the write head as the head moves away from that point.

On a rigid disk, the disk moves in a circular motion under the head. Information is stored on the disk in concentric tracks, the width of a track roughly being governed by the size of the write head. The density of recording per sq inch (known as areal density) is the product of the number of tracks per inch (tpi) and the linear density of information along a track measured in bits per inch (bpi). Typical numbers for today's high end (i.e., expensive) rigid disk drives are: 3,000 tpi and 30,000 bpi.

The current into the write head induces a magnetization pattern on the track immediately below the write head. When a track is to be read, a read head is positioned over the track. Then, the magnetization pattern "frozen" on that track radiates flux that is sensed, or "read," by the read head. The read head produces a voltage that is symptomatic of the magnetization on the track being read. There are primarily two types of read head: inductive heads which contain coils of very fine wire and which produce a voltage proportional to the time derivative of the flux that passes through its coils, and magneto-resistive (MR) heads which produce a voltage directly proportional to the flux sensed by the head. MR heads produce larger read voltages than inductive heads, but have a limited dynamic range for linear operation. Only inductive heads have been used for writing, to this date.

Consider the familiar audio cassette tape recorder in Figure 0-2. The recording/playback head is made of easily magnetised ferrite material and has a small air-gap γ at the point where it comes into contact with the recording tape, $\delta = 0$. When energised, the coil winding on the structure is used to create a strong and concentrated magnetic field on the recording media as it moves along with a velocity v. During the playback mode, this coil detects the induced voltages.

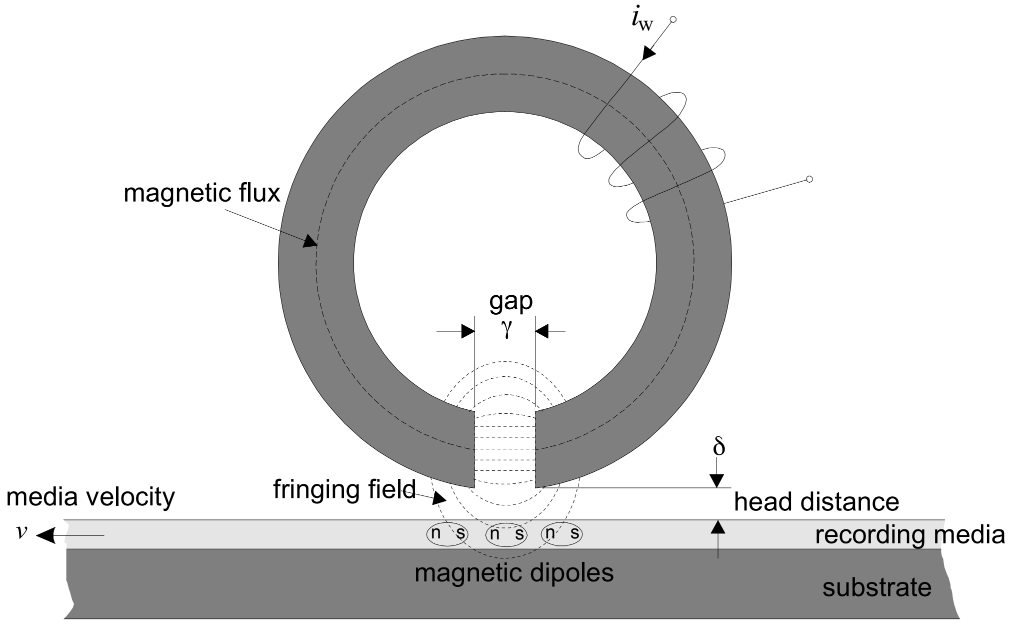


Figure 0-2 Basic Ring read/write head.

The recording media in this case is a length of MYLAR (plastic) tape coated with a powdered ferric oxide compound which is magnetisable and has high remanence. This layer of magnetic material in the unmagnetised state may be conceived as made up of dipoles, tiny magnets with N-S poles randomly positioned. Under the influence of the external magnetic field, these dipoles will align their N-S poles in line with the applied field thus becoming magnetised. Upon removal of the applied field some of these dipoles remain aligned.

By either increasing the rate v the media is moved across the head, or by decreasing the granularity of the magnetic material, (i.e. making the tiny magnets smaller), we can record faster changes in the applied magnetic field, that is, the frequency response is increased. For digital data, the density of the stored information increase with decrease in the granularity of the magnetic media.

With a weak field, only a small number of the dipoles retain their alignment. As the field gets stronger, more and more of them will remain aligned, that is, the stored magnetic field increases. For audio (analogue) recording, the variation in the audio signal levels are recorded in this linear region of the magnetic behaviour. A saturation level is reached when increases in the applied field does not result in a corresponding increase in the stored magnetic field. Digital recording generally operate in the saturation region.

7.3 The Digital Read/Write Channel

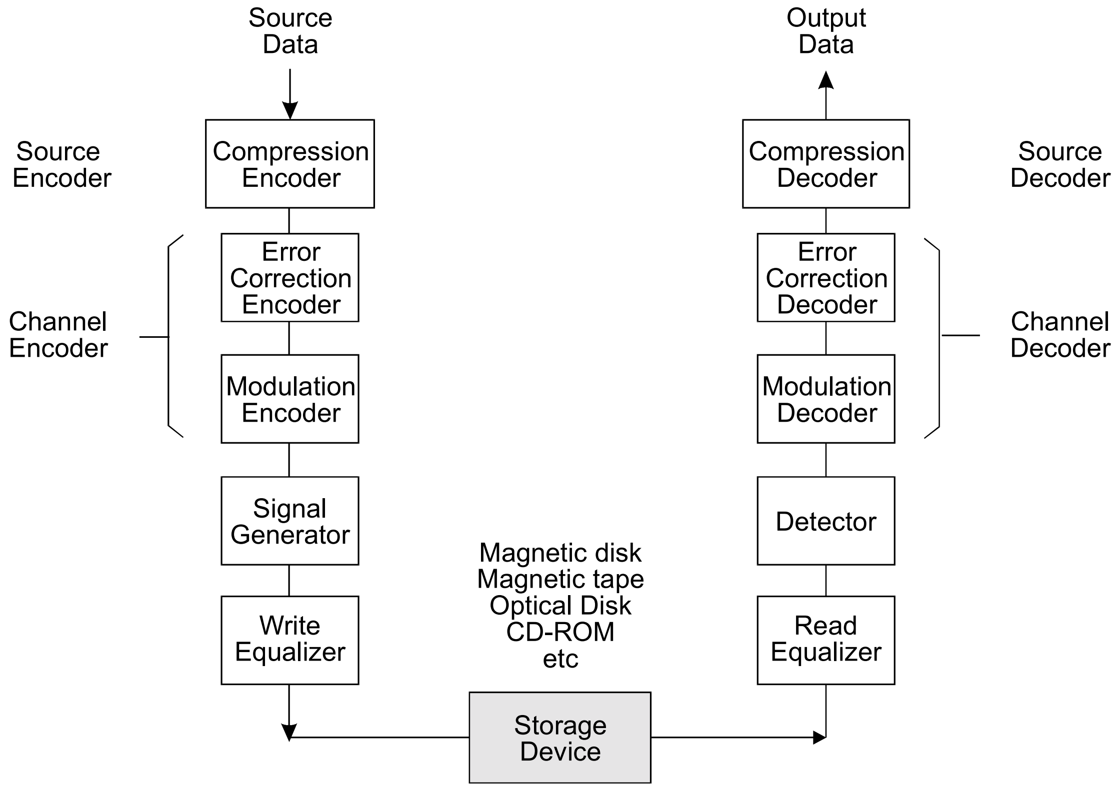


Figure 0-3 The Digital Recording channel

Digital recording systems may be considered as communications channels. There is an input signal and an output signal which is a transformed and noisy version of the input. Quoting from a paper by Berlekamp "Communication links transmit information from here to there, Computer memories transmit information from now to then." Figure 0-3 show the overall block diagram of the digital read/write channel. The source data to be recorded or saved is prepared by the CPU, which also provides the storage device with information concerning the address of the storage locations. Compression and other source data preprocessing takes place before the data passes into the channel encoder which adds error-correcting bits and converts the data stream into a form suitable for recording. This signal is passed through the equalisation filters, and amplified as the write current for the recording head, creating the pattern of magnetic fluxes reversals on the storage medium.

To recover the stored data at the output, a reverse sequence takes place. Starting at the storage medium, the flux reversals are sensed by the magnetic head. The pulses are demodulated, equalised, decoded and finally presented at the output as the read data.

In digital recording, the magnetic medium is saturated and flux reversals are used to represent the digital information. Figure 0-4 shows the relationship between the write signal, flux changes and subsequent induced current in the read mode.

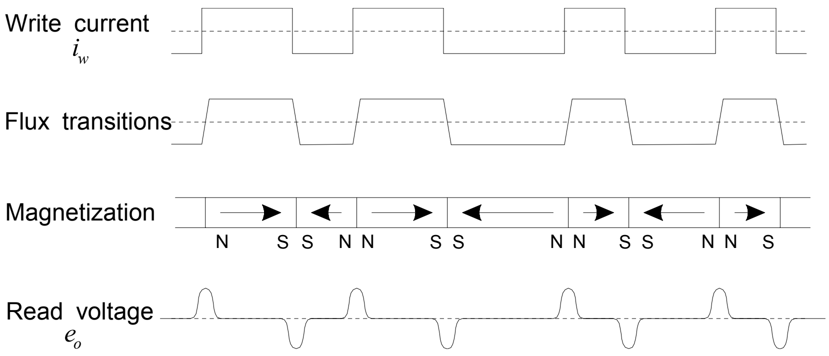


Figure 0-4. Magnetic write and read waveforms.

7.3.1 Write Process

During the *write* process, a write current i_w is passed through the coil. Since the current needs a finite time to build up and the media is moving under the head, the result is a magnetic transition with a finite rise-time as shown in Figure 0-4. We can represent the change in magnetic flux mathematically as

$$M(x-x_0)$$

Disks and tapes employ the longitudinal recording method illustrated above in which the flux lines of the magnetic field are oriented in the direction of the motion of the media. Although higher recording densities are attained using the vertical or perpendicular recording method in which the flux lines are perpendicular to the surface of the media this method is more difficult and correspondingly more expensive to implement. As shown above, the direction of the write current determines the polarity of the magnetisation of the recording medium. When the current reverses, it creates a flux reversal.

7.3.2 Read Process

During the *read* process, the magnetisation on the recording surface is detected by the head and some of the magnetic flux is diverted through the coil, producing an induced voltage e_o which is proportional to the rate of change of flux as seen also in Figure 0-4. Mathematically we can represent this read pulse as

$$e_0 = CV \frac{\partial}{\partial x} \int_{-\infty}^{\infty} D(x)\, M(x - x_0)\, dx$$

where D is a the efficiency of the read head. As it is, these detected current pulses are hard to distinguish from noise pulses found in magnetic media and techniques are required to properly encode and decode the data for magnetic recording purposes. Under ideal conditions, the peak of the read pulse indicates the position of the flux transition. *Peak detection* of this signal is implemented by first converting the signal to positive pulses with full wave rectification. These pulses are differentiated and detected with a zero crossing detector as shown in Figure 0-5.

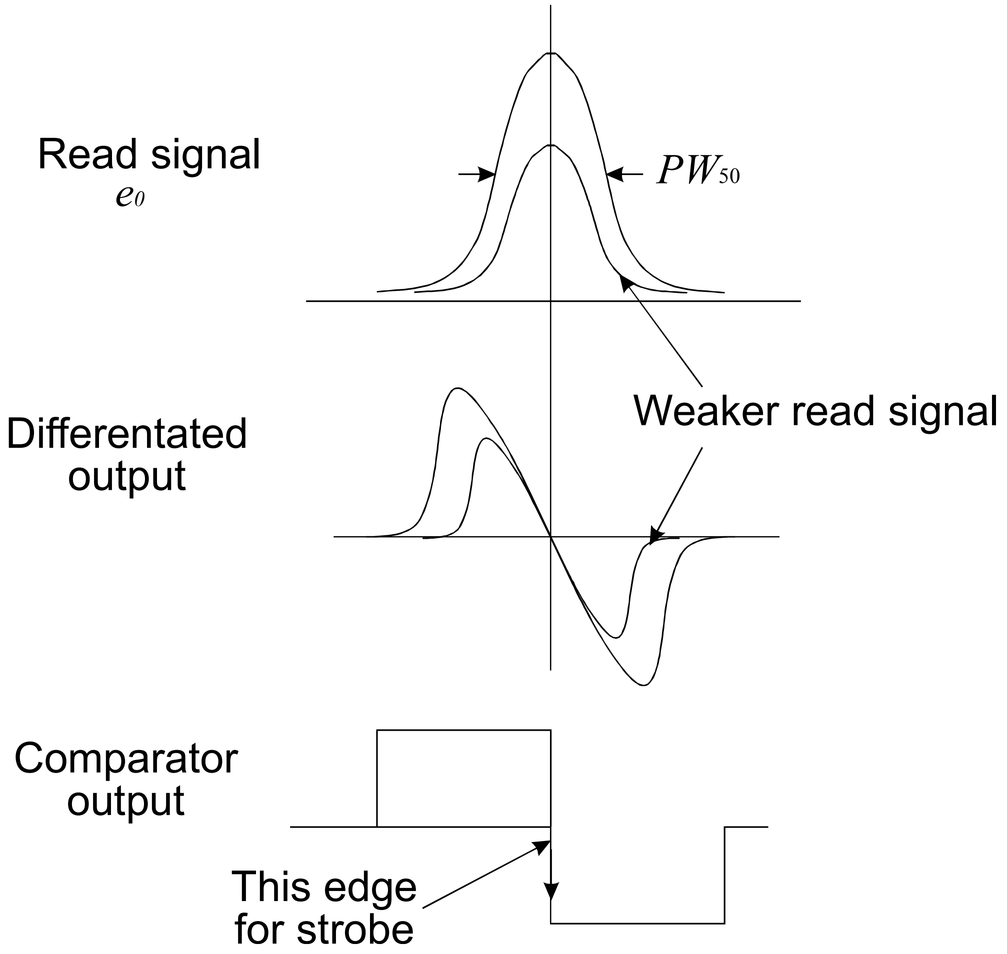
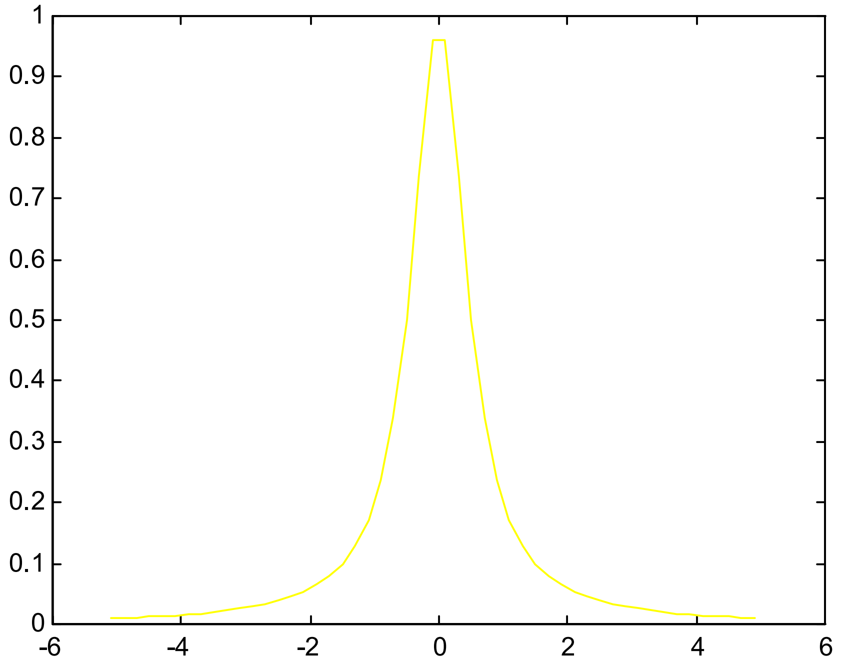


Figure 0-5. Model of read signal used with peak detection.

Mathematical models of the read pulse are used in the design of the demodulation and equalisation circuits, and an important parameter used in describing the pulse is the width of the pulse at 50% amplitude, PW_{50}. The Lorentzian model is the most common mathematical model for an isolated transition response. The Lorentzian pulse shape can be expressed as:

$$h(t) = \frac{A}{1 + (2t / PW_{50})^2}$$



7.4 Peak Detection Systems

One major problem affecting the read signal is the effects of noise. It is common to assume that the noise in the system is additive and Gaussian. Usually, it is also assumed that the noise consists of two components: a Gaussian white noise component due to the electronics read (i.e., receiver) side, and a Gaussian coloured component due to the medium. The spectral characteristics of this coloured noise are essentially the same as would be obtained from passing white noise through the linear transfer function characterizing the system. More complicated models for the noise exist, but we will not go into the discussion.

The design of the modulation, coding and signal processing in past magnetic recording products has been driven by the detector chosen to detect the transitions in the channel input waveform. This detector, called a peak detector has the advantage of being both robust and extremely simple to implement. However, by its very nature, it works best at low linear densities. A block diagram of a typical peak detector is shown in Figure 0-6: Block diagram of a peak detector.

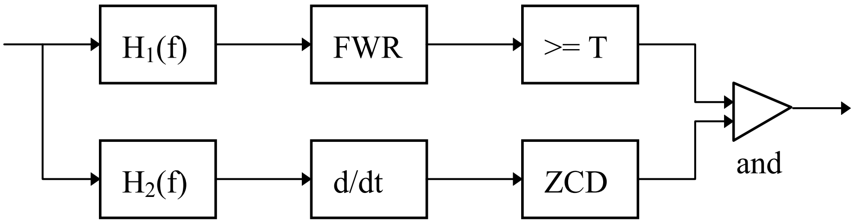


Figure 0-6: Block diagram of a peak detector

The are two path through the detector. The first path is used to qualify a peak, i.e., to ensure that the peak has sufficient amplitude. This path consists of a linear filter, $H_l(f)$, a full wave rectifier (FWR), and a threshold testing circuit. The bottom path is used to locate the peak by differentiating the signal after the linear filter $H_2(t)$ which remove most of the noise and then passing the differentiated signal through a zero crossing detector (ZCD).The system only accepts a peak if the amplitude was large enough to the qualification test.

7.4.1 Data and Clock Recovery

Once a peak is detected by the peak detector, it is thought to be due to a transition in the input waveform. A device called a phase-lock loop (PLL) is used to derive timing from the position of the detected peaks. The PLL produces a clock of period T_b seconds by which to identify channel bit intervals (sometimes called "bit cells"). Then, if an output pulse is located in a bit interval, that bit interval is said to contain a transition.

The output of the peak detector is used as an input to the PLL, and the output clock produced by the PLL is constantly being adjusted so that the average peak position is centered with respect to the edges of the bit interval.

The data and clock information, stored by the schemes described later in the following section, must be recovered and separated during the read operation. A phase-locked loop (PLL) as shown in Figure 0-7 is used for this purpose.

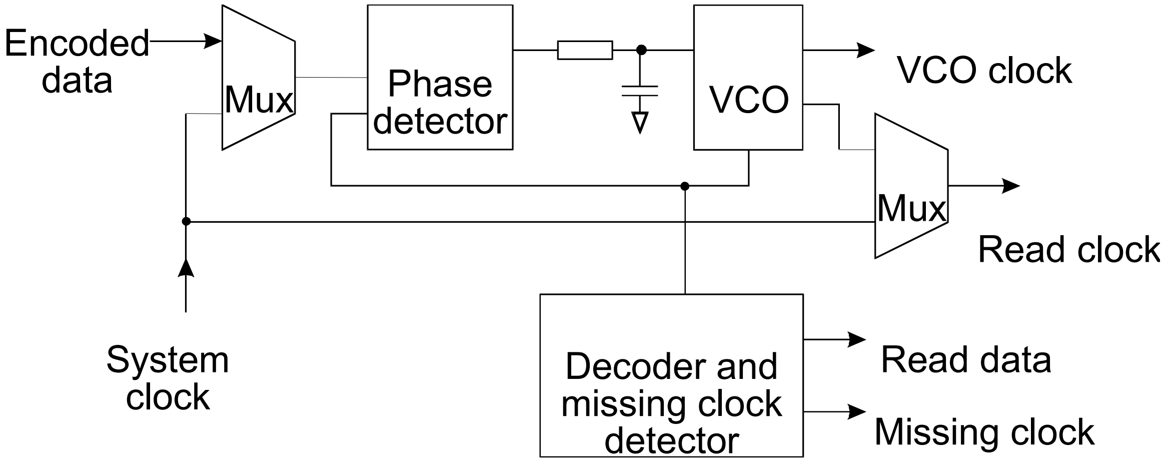


Figure 0-7. Phase-locked loop (PLL).

The phase difference between the incoming signal frequency and the frequency of the voltage-controlled oscillator (VCO) is averaged out by the low-pass filter, forming a DC voltage to control the VCO. The output of the VCO is divided down to a frequency equal to the incoming signal frequency and fed into the phase detector. The VCO and the incoming signal are thus locked together. Any variation in frequency of the incoming signal results in a change in the phase difference which produces a error voltage to the VCO, causing it to track the frequency of the incoming signal.

The VCO, which is synchronised to the encoded data stream read back from the drive, is used to generate two windows, one for the clock and the other for the data pulses. Sync bytes, written as part of the record format are used to identify the two sets of pulses.

7.4.2 Bit Shifting[TC1]

During read back, the signals are very weak and the magnetic head may be considered to be operating linearly. Each read pulse can be considered independently for analysis. As the recording density increases, these read pulses are crowded together, giving rise to a source of problem known as peak shift or bit shift. Referring to Figure 8.1,[TC2] it will be seen that at each flux transition, the polarity of the domain changes setting two like poles adjacent to each other. As the superposition of the two read pulse, the result is that the bits will be stored in a position shifted from the nominal bit cell position. The actual amount of shift depends on the bit pattern and has to be compensated for.

We can look at the problem by considering a sequence of two read pulses. When the waveforms of the two signals are superimposed on each we notice two effects occurring. Firstly, the amplitude of the signals is reduced. More significantly, the position of the peaks are shifted away from each other.

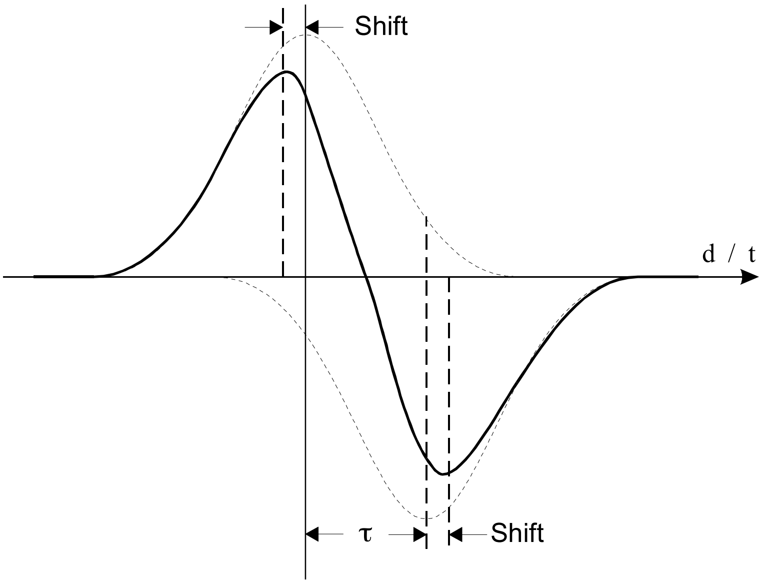


Figure 8.0-8. Two adjacent bits superimposed resulting in bit shift.

If one examines the waveform produced by the linear superposition of two Lorentzian pulses (of opposite sign) separated by αPW_{50} seconds, one finds that this waveform will contain two peaks separated by βPW_{50} seconds, where $\beta > \alpha$. The parameters α and β are related by the formula

$$\beta = \sqrt{\frac{\alpha^2 - 1 + 2\sqrt{1 + \alpha^2 + \alpha^4}}{3}}$$

which for small α becomes

$$\beta \approx \frac{1 + \alpha^2}{\sqrt{3}}$$

For α much greater than 1, β is approximately equal to α, but as α approaches zero, β approaches a fixed, limiting distance given by the value. Thus, the peaks will be centered in their bit interval only at low densities.

The usual method of eliminating errors from bit shift is by *write pre-compensation*. In this technique, the amount of shift is determined from the bit pattern and a deliberate shift in the opposite direction is introduced during recording to compensate for it. The amount of shift is a function of the pattern as well as the proximity of the neighbouring pulses. By assuming a mathematical model of the read pulse waveform, these values can be calculated. These compensation values are stored in a lookup table which would list all possible bit patterns, for say a group of eight bits, and the shift required. 'Early' and 'Late' write compensation is usually incorporated into the controller circuitry. These errors produced by the interference between neighbouring data bits is known as *intersymbol interference* (ISI).

7.5 Data Encoding

There are many ways of encoding the data as a function of flux reversals. The simplest would be to represent each 1 with a pulse leaving the signal low for each 0. This is the return-to-zero (RZ) code as the level always drop back to the zero state. If the level for the 1 is held high for the whole bit period, we have the special case of the non-return-to-zero (NRZ). In the early days of magnetic recording many codes were introduced. Although many of these are now obsolete, new codes are still being proposed mathematically in the search for improved performance. The design of new codes take into account two main factors:

(i) The various parameters like density, immunity to noise and timing inaccuracies, complexity of code etc. have to be balanced against each other, and the design is a compromise of trade-offs depending on the specific application and environment.

(ii) Many codes are protected by patent and sub-optimal solutions are proposed to avoid license fees.

7.5.1 Common Code Definitions

The following section, together with Fig 0-9, defines some of the various codes used in digital recording. Generally, the write waveforms are used in the representation. The two levels of saturation are depicted vertically and the horizontal scale represents the distance moved by the medium under the head, or equivalently, time. The horizontal unit used is the bit cell width and is defined the period between two clock transitions. Within the bit cell, the level can be "up" or "down", and there may or may not be a transition, called the *data transition.* Various combinations of clock and data transitions are possible.

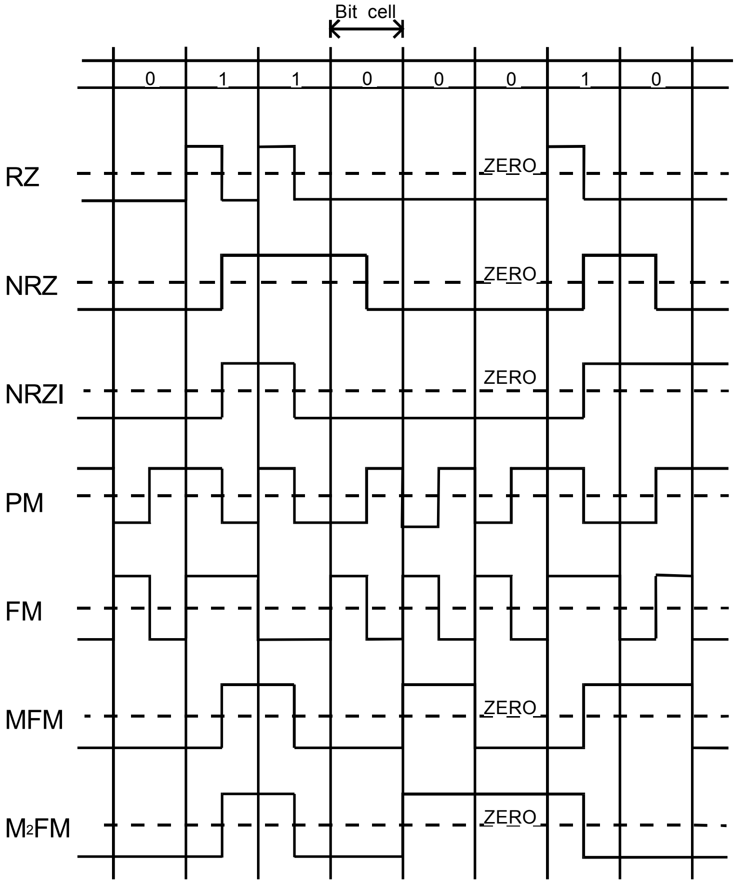


Fig 0-9. Recording codes.

(a) *Return to Zero* (RZ)

One saturation level is designated as the zero level.

0 : there is no flux transition within the whole bit cell

1 : is represented by the occurrence of a downwards transition within the bit cell. As the level is always at the zero level, an upwards transition is required at the beginning of the bit cell.

- RZ is functionally the same as Return to Saturation (RS). Transitions occur at both cell boundaries and mid-cell.

(b) *Non-Return to Zero* (NRZ), *Non-Return to Zero-Level* (NRZ-L)

0 : Down level (baseline)

1 : Up level

- The Non-return to zero (NRZ) technique of recording is the simplest and most efficient, but it is not self-clocking, and we cannot obtain the bit cell information from it. Flux reversals occur only at mid-cells (or cell boundaries in some implementation).

(c) *Non-Return to Zero, Invert* (NRZI) or *Non-Return to Zero, Mark* (NRZ-M)

0 : No transition anywhere, magnetic state remains the same as in the previous bit cell, whether up or down.

1 : A transition occurs in the middle of the bit cell. The direction of the transition is not important.

- A long sequence of zeroes will result in no transitions during that period and the loss of clock synchronization. 9-track parallel tape drives can use NRZI as the track for the parity bit will always have a transition when a long sequence of (00)h is written.

(d) *Phase Modulation* (PM), *Bi-Phase, Transition* (Biϕ-T) or *Manchester* code.

0: Up-going transition in mid-cell.

1: Down-going transition in mid-cell.

- Clock transitions, at the cell boundaries, are used to enable the proper direction for the data transition. For a series of 1's or 0's, an addition transition has to take place at the cell boundary in order that the correct direction of the transition can take place at the cell centre. As each bit could require up to two transitions to encode, this is not an efficient technique. There is always a transition at mid-cell, and this can be used for the clocking purposes. Phase encoding (PE) is for 9-track tapes and is a simple self-clocking encoding technique.

(e) *Frequency Modulation* (FM) or *Bi-Phase, Mark* (Biϕ-S).

0: Transition at mid-cell, direction is not important.

1: No transition at mid-cell, level is not important.

- A clock transition is always present at the cell boundaries.

- Frequency modulation (FM) is similar to PE in efficiency but does not use the direction of the flux transition. Except for the clock transitions, FM is similar to NRZI, with the representations for 1 and 0 interchanged. FM is the encoding method used for single-density floppy disks.

(f) *Modified Frequency Modulation* (MFM) or *Mille*r code.

0: No transition.

1: Transition at mid-cell, exactly as in NRZI.

- Since a sequence of zero bits will not have any transitions, introduce a flux transition at cell boundaries to separate two adjacent 0 bits. In this way there will be at least one flux transition in any two bit cell period, making it self-clocking. Modified FM (MFM) was introduced by IBM to increase the density of recording by removing most of the flux transitions at bit cell boundaries in the FM technique, thus reducing the fcpi to an average of one. MFM is the standard encoding technique for double-density floppy disks and the earlier Winchester hard disks. It should be observed that MFM is actually a modification of NRZI and not FM coding.

(g) *Modified MFM* (M^2FM).

- Modified MFM (M^2FM) was introduced by Shugart Associates, one of the earliest manufacturers of floppy disk drives. It eliminates the flux transition at the bit cell boundary if it is preceded by a cell boundary flux transition in the previous cell. However IBM continued to use the MFM method, which became the de facto industry standard.

7.5.2 Run-Length-Limited Encoding (RLL)

As mentioned earlier, intersymbol interference places an upper limit on the density of the flux transitions that a particular head-medium system can achieve. Bit shift effects can be compensated with an early write signal performed on the code prior to the write process. During the read process we can pass the detected pulses through a equalization filter to "slim" them, thus reducing the effects of pulse superposition.

Another way of improving the recording density makes use of run-length limited encoding schemes. This constrained sequences result in a two-level write waveform for which the minimum and maximum intervals between transitions are fixed.

Run-length limited are designated as:

$$\mathrm{RLL}(d, k, m, r)$$

and are often shortened to

$$\mathrm{RLL}(d, k)$$

where

d = Minimum number of consecutive zeroes allowed (including clock).

k = Maximum number of consecutive zeroes allowed (including clock).

m = Minimum number of data bits to be encoded.

n = Number of code bits (including clock) for each of the m data bits.

r = Number of different word lengths in a variable length code.

DR = Density Ratio. Data bits per flux reversal.

$$DR = \frac{\text{bit per inch (BPI)}}{\text{flux changes per inch (FCPI)}} = (d+1) \text{ x } \left(\frac{m}{n}\right)$$

FR = Frequecy Ratio.

$$FR = \frac{\text{maximum time between transitions}}{\text{minimum time between transitions}} = \frac{k+1}{d+1}$$

w = Detection window expressed as a percentage of a data bit cell

$$w = \left(\frac{m}{n}\right) \times 100$$

Table 1 gives the mathematical calculations performed for a number of popular *d, k* constrained codes.

Code name	Where used	*d*	*k*	*m*	*n*	*r*	*DR*	*FR*	*w*
NRZI	Early disks, tapes	0	∞	1	1	1	1.0	∞	100%
FM	Floppy disks	0	1	1	2	1	0.5	2.0	50%
GCR	Tape	0	2	4	5	1	0.8	3.0	80%
MFM	Disks, IBM 3330	1	3	1	2	1	1.0	2.0	50%
RLL(2,7)	Disks	2	7	2	4	3	1.5	2.67	50%
RLL(1,7)	QIC Tape	1	7	2	3	1	1.33	4.0	64%
EFM	Compact Disks	2	10	8	17	1	1.41	3.67	47%

Table 1 Performance of some RLL codes.

We shall look at some popular RLL codes below:

7.5.3 *Group-coded recording* (GCR) or RLL(0,2)

GCR is another self-clocking modification of NRZI. It will be seen that we can translate each 4-bit group into a 5-bit group with the help of Table 2.[TC3] It will be noted that the 5-bit code are defined such that the resultant data stream will not contain more than two consecutive zeroes. The GCR-translated data can be recorded using NRZI without introducing any additional flux transitions and is thus very efficient. GCR is the recording method used in high density tape drives such as the IBM 3420. The penalty to be paid is in the complexity of the encoding/decoding logic.

Data nibble	GCR code	Data nibble	GCR code
0000	11001	1000	11010
0001	11011	1001	01001
0010	10010	1010	01010
0011	10011	1011	01011
0100	11101	1100	11110
0101	10101	1101	01101
0110	10110	1110	01110
0111	10111	1111	01111

Table 2 GCR encoding table.

7.5.4 Run-length-limited 1,7

The rate 2/3 (1,7) code is arguably the most popular *(d,k)* code in use today. Several variations of this code exist. A simple and elegant description, due to Jacoby, begins with the encoding table in Table 3.

Data nibble	RLL (1,7) code	Data nibble	RLL (1,7) code
00	101	00 00	101 000
01	100	00 01	100 000
10	001	10 00	001 000
11	010	10 01	010 000

Table 3 RLL(1,7) encoding table.

When the encoder is ready to encode a pair of user data bits, it "looks ahead" to the next pair of user data bits to see if it exist in the right side of Table 3 for both pairs. If the combined word is not found, use the left side for encoding.

As illustrated for a rate *m/n* code, the decoding of a code word of length *n* depends on the contents of a decoder window that contains the code word in question, as well as a fixed number of past and future code words ("look back" and "look-ahead"). In the case of the (1,7) code, the decoder decodes the current three-bit code word by looking ahead at the next two upcoming code words. In this way, a single incorrectly detected code symbol can propagate into a burst of at most six user bits (in fact, the burst length does not exceed five user bits).

7.5.5 Run-length-limited 2,7

Another data encoding used together with NRZI modulation encoding in several disk drive products is a rate 1/2 (2,7) code. One encoding and decoding tree for such a code is given in Figure 0-10. RLL 2,7 Coding Tree.

It is easily seen that the code rate is 1/2 (every code word contains exactly twice as many binary digits as the information sequence it represents) and that any concatenation of the variable length code words satisfies the (2,7) constraint (each 1 in every code word is followed by at least two 0s, and no code word begins with more than four 0s or ends in more than three

0s). It can also be verified that every information sequence can be decomposed uniquely into a sequence of variable length strings in the tree. In addition, the decoding errors due to a single code symbol in error cannot affect more than four user bits.

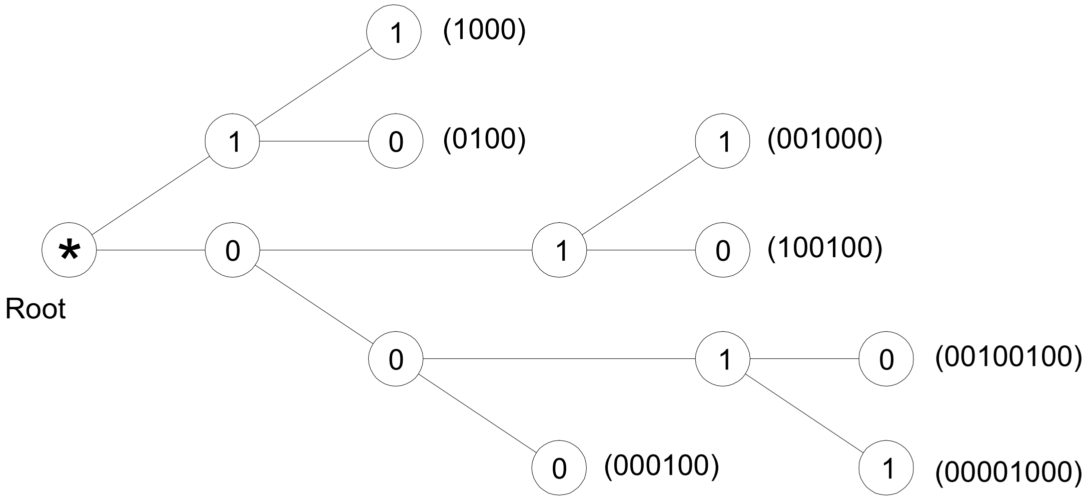


Figure 0-10. RLL 2,7 Coding Tree.

The coding tree for the RLL 2,7 code is given Figure 8.2 and Figure 8.3 shows the write waveforms. Although each data bit is encoded into 2 channel bits, a 50% improvement in data density is obtained when compared to MFM coding when we only consider the size of the magnetic domain (fcpi).

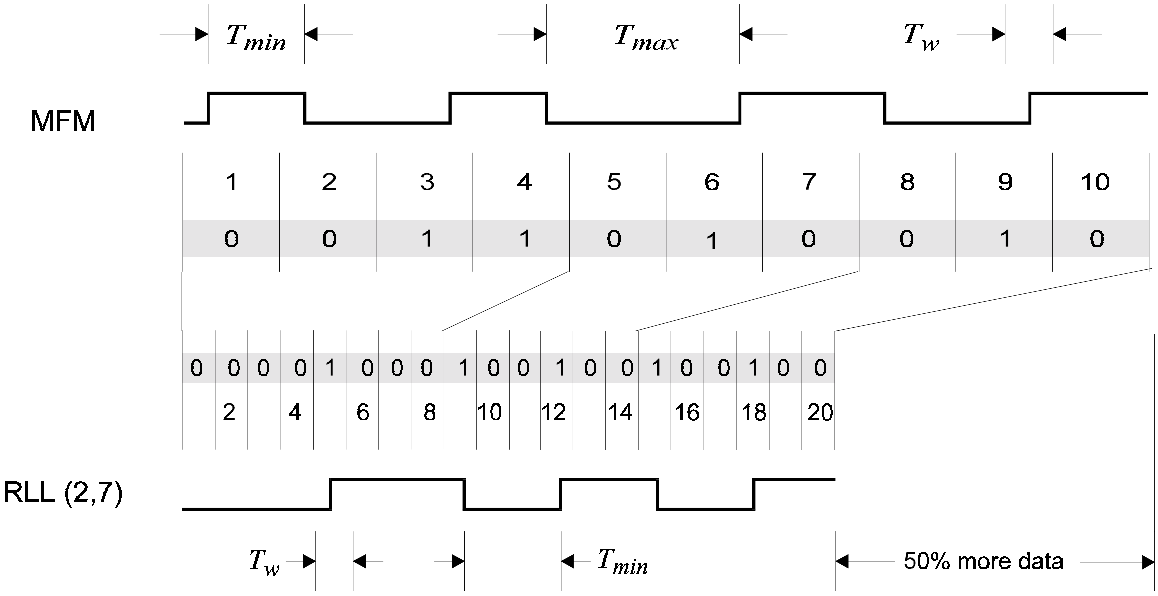


Figure 0-11. Run length limited coding (2,7).

Prior to a few years ago, no general theory existed for the design of data encoding codes (such as *(d,k)* codes) with minimum code word length, finite-state encoders, and sliding-block decoder. Now, there is systematic technique for code construction. The method, called the sliding-block code algorithm, allows for the design of a practical, efficient *(d,k)* code for any choice of the parameters *d* and *k*.

7.6 Error Checking Techniques

Errors can enter the system at each stage of the digital recording channel. Error sources may be classified as noise sources, which arises from random processes and interference sources, which is deterministic. In digital recording, the most important noise sources come from the heads, electronics and media. Interference sources of errors include electromagnetic interference, intersymbol interference, cross-talk, and incomplete overwrite of previously recorded data.

External interference have to be controlled by careful design and shielding. One of the major sources of intersymbol interference is bit shifting when the recording density is very high. This has been discussed earlier.

Head noise is generated randomly in the magnetic head as in any other electrically resistive component. *Electronics noise* is generated in the components of the electronics circuit by the resistive components and active devices such as transistors and integrated devices. These noise sources are temperature dependent and are especially significant in the low level analogue signal processing portions of the circuit.

Medium noise is caused by variations in the physical properties of the recording material. In the case of magnetic recording, this could arise from impurities in the magnetic material, uneven thickness of the coating, variations in the particle size, physical scratches on the surface and stretching of the substrate. Similar physical defects occur in optical media. Other sources of errors come from poor tracking, and mechanical jitter and shock.

During the write process, encoder errors produce permanent (*hard*) errors in the recorded data. Permanent errors may be introduced by the media, where the magnetic properties of the material may be poor, causing bit dropout.

When the data are being read, noise, unstable clocks, mechanical and electronic jitter all contribute to the problem of data recovery. Other factors include crosstalk, inadequate erasure of previously written data, tracking error, etc. Sometimes the data can be recovered by making several attempts at reading. If eventually the data is recovered, these are known as *soft* errors.

In general, careful design of the low level preamplifiers for the read heads will keep under control the head and electronics noise, so that media noise is the primary contribution to data errors. As recording density increases, hard and soft errors created by dirt and other contamination in magnetic disk drives also increase. Optical disks currently have a recording density an order of magnitude greater than magnetic disks making the manufacture of defect free media beyond the current state-of-the-art. Errors will occur, and the only solution is to have a strategy to manage these errors so that data integrity is not compromised. Random single bit errors can cause a number of bits of decoded data to be in error. In some cases a string of errors may occur together; these are called burst errors.

Error detection and correction (EDAC) techniques have been extensively studied and we will only briefly mention some of them here. When more complex codes are used, care must be exercised in the design of codes such that the propagation of errors are controlled to a known extent.

7.6.1 Block codes

Similar to data transmission applications, block codes are used to protect the stored data from noise and random errors. Starting with k data symbols (bits), we can add $(n - k)$ parity symbols to construct a code word with block length n. The code is designed in such a way that up to t corrupted symbols within the block can be detected and corrected. The (n, k) Hamming code is an example of bit-organised codes, whereas the (n, k) Reed-Solomon (RS) codes are used with byte-organised data.

The Hamming single-correction code is expressed as

$$2^{(n - k)} = n + 1$$

When a single parity bit is added to the data word, the number of possible codes is doubled. The parity rule requires that only half of these are valid codes, the other half being invalid as they contain single bit errors. The error words differ from the valid codes by one bit, giving a Hamming distance of 1. By increasing the number of parity bits used, we can increase the Hamming distance between valid codes. Assuming that when an error occurs, small errors with fewer error bits are more likely than large multi-bit errors, then correction may be implemented by choosing the valid code that is closest.

The (7,4) Hamming code represents the 4-bit data word by a seven-bit code word using 3 parity bits. From Table 4. we see that each code word differs from every other code word by a *distance* of 3 bits. When a single-bit error occurs during the read process, then if will differ from the correct code word by 1 bit and from every other code word by 2 bits. Thus we can recover the correct code word using the (7,4) Hamming code when there is a maximum of 1 error bit. The code efficiency k / n, i.e., the ratio of the data bits k to the block length n, is 0.57. When $(n - k) = 8$, $n = 63$, and the code efficient for this larger block length becomes (63 - 8)/63, or about 90%.

Hamming codes are relatively simple but have limited applications, for example in ECC-memories. More widely used in data storage as well as data communications are the more complex Reed-Solomon codes. RS codes are discussed in the standard texts on data communications and information theory.

Data	Code

0000	0000000
1000	1000110
0100	0100101
1100	1100011
0010	0010011
1010	1010101
0110	0110110
1110	1110000
0001	0001111
1001	1001001
0101	0101010
1101	1101100
0011	0011100
1011	1011010
0111	0111001
1111	1111111

Table 4 Hamming (7,4) Error correcting code..

7.6.2 Cyclic redundancy check (CRC) characters

Written together with the block of data, usually 128, 256, 512, or 1024 bytes in length is a 16-bit CRC character. On read back, the CRC is recalculated from the read back data and compared against the written value. Failure to verify the CRC check leads to an error routine, which normally initiates a retry. A *soft* error is one which is temporary and gives a successful read after one or more retries and is probably caused by some dust on the disk surface or by electrical noise. Errors are considered *hard* or permanent when after the specified number of retries have taken place, there is no change to the result. For disk drives, the track is marked and added to the table of bad tracks.

CRC characters are generated using shift-registers and exclusive-OR gates that produce an unique 16-bit number for each block of data. Usually the CRC-CCITT polynomial $x^{16}+x^{12}+x^5+1$ is used although other versions are available. Figure 8.4 shows a circuit for implementing CRC calculation and the algorithm is shown below.

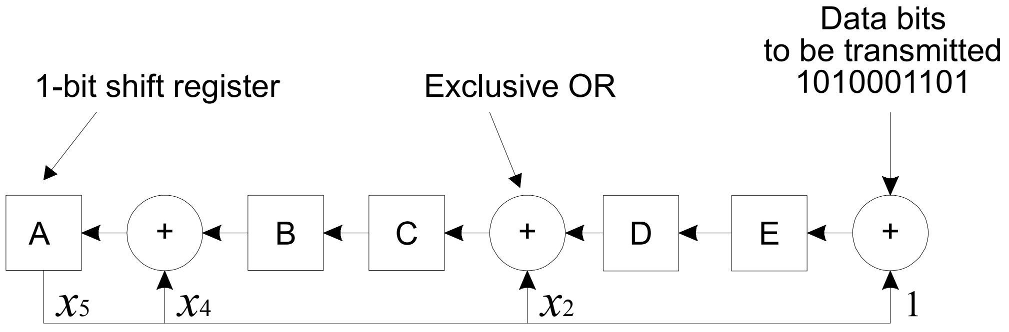


Figure 0-12. Shift register implementation of division by polynomial $x^5+x^4+x^2+1$.

Let us assume the following:

a 10-bit message M = 1010001101

a 6-bit polynomial P = 110101
the 5-bit check character R = to be calculated.

```
                    1101010110 ← Q
P  →  110101  )101000110100000 ← 2^nM
               110101
                111011
                110101
                  111010
                  110101
                    111110
                    110101
                      101100
                      110101
                       1100100
                       110101
                         01110 ← R
```

The remainder R is added to 2^nM to give T = `101000110101110`. At the read channel, T is divided by P, and if the remainder is zero, it is assumed that there have been no errors. All errors except those divisible by P will be detected.

7.6.3 Interleaving

The bit-error rate (BER) is the parameter used to specify the probability of errors in storage devices. BER may be defined as the number of error bits divided by the total number of bits. Magnetic storage devices typically have BER of 10^{-8}. On the other hand, optical storage devices like CD-ROM and WORM work with much higher recording densities, and current manufacturing technology have difficulty achieving BER of better than 10^{-6} consistently. Often a physical defect occurs that is large enough to cause a sequence of many bits to be in error. *Burst errors* average several bits in length but can extend to several hundreds of bits.

The most powerful Reed-Solomon (RS) error correcting code (ECC) can correct only 8 bytes per code word which is insufficient to handle the error bursts of 100 bits or more that are encountered in optical disks. But if, prior to the recording, several code words are combined by interleaving, very long burst errors can be reduced to a number of shorter burst which can be corrected by the ECC.

Rather than bits, RS codes are based on *symbols*, typically an 8-bit byte. If a RS is designed to correct an error of 1 byte in a code length of eight bytes, the burst error correction length is only 1 bit, as a two-bit error occurring at the boundary of the byte symbols would result in a 2-byte error which is uncorrectable. However, by interleaving three code words before storage, burst lengths of up to 17 bits could be managed as shown in Figure 8.5. Three words are encoded and stored in memory by row, but are recorded serially by column. If a burst error of 3 bytes should occur, the error will be distributed over the three code words and during decoding on a row by row basis, the data can be recovered.

RS code with an interleave factor of 3 like the scheme above is used in the IBM 3370 disk storage device. They are also used in optical, high capacity tape and disk storage devices.

1	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0

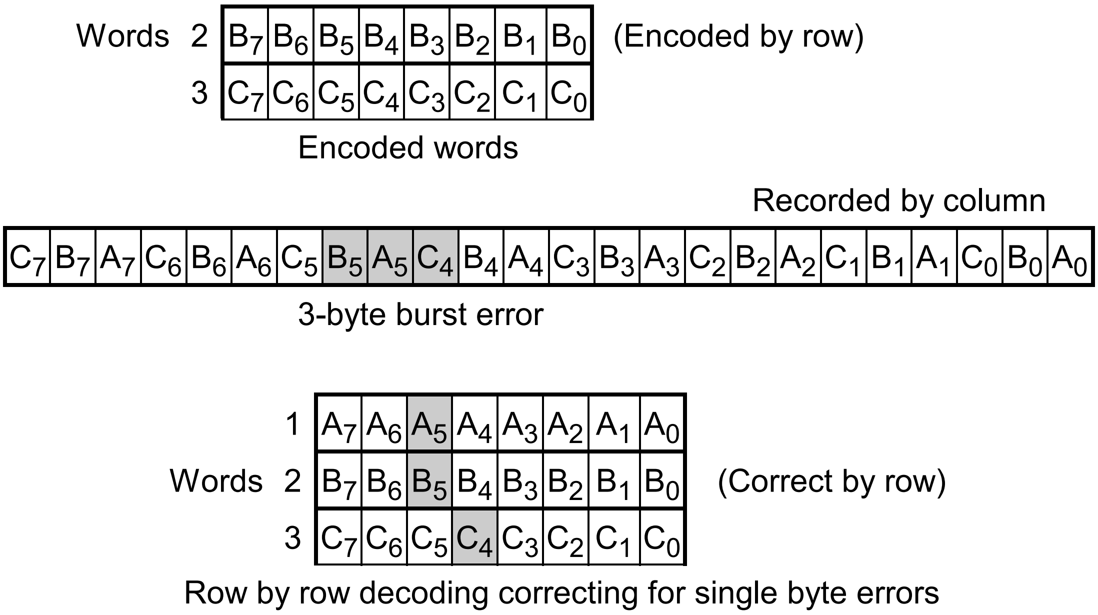


Figure 0-13. Effect of interleaving on burst errors.

7.7 Summary

Digital magnetic recording is a unique application of magnetic recording technology, and in these sections we can only cover briefly the range of engineering disciplines involved. The areas of data encoding and error correction draws heavily on information theory and data communications. For a more comprehensive treatment of the subject the reader is referred to the following:

Digital Magnetic Recording, 2nd Ed., A.S. Hoagland and J.E. Monson, John Wiley, N.Y. 1991.

Coding for Digital Recording, J. Watkinson, focal Press, London, 1990.

Optical Recording, A.B. Marchant, Addison Wesley, Mass, 1990.