Chapter 4
Modern Hard Disk Drive Systems: Fundamentals and Future Trends

Tong Zhang, George Mathew, Hao Zhong, and Rino Micheloni

Abstract The objective of this chapter is to provide the readers with a basic technical overview of modern hard disk drives. As a mainstream data storage media, magnetic recording hard disk drives have been enjoying a steady increase in areal recording density for more than four decades, with a rate at least as dramatic as what has been predicted by Moore's Law for the increase of transistor density in integrated circuits. Thanks to continuous advances in recording materials, read and write heads, and read channel signal processing and error correction coding, the areal density increase in hard disk drives appears to be well on track and is steadily moving toward achieving the milestone of 1 Tb/in^2. This chapter will provide an overview of modern hard disk drive systems with the focus on recording channel modeling and advanced signal processing and coding techniques being used in current design practice.

Keywords Hard disk drive · Magnetic recording · Read channel

Hard Disk Drive Systems

Arguably, the two most important technologies that enable today's information age are integrated circuits and magnetic recording. With more than 100 years of non-stopping development [11], magnetic recording has enabled a multibillion dollar industry. Today, magnetic recording systems, particularly hard disk drives, are being used virtually everywhere and play a crucial role in numerous computing and communication applications. During the evolution of hard disk drives, the most noticeable metric is probably *areal storage density*. Since IBM introduced the first

T. Zhang (✉)
Electrical, Computer and Systems Engineering Department, Rensselaer Polytechnic Institute, NY, USA
e-mail: tzhang@ecse.rpi.edu

G. Campardo et al. (eds.), *Memory Mass Storage*,
DOI 10.1007/978-3-642-14752-4_4, © Springer-Verlag Berlin Heidelberg 2011

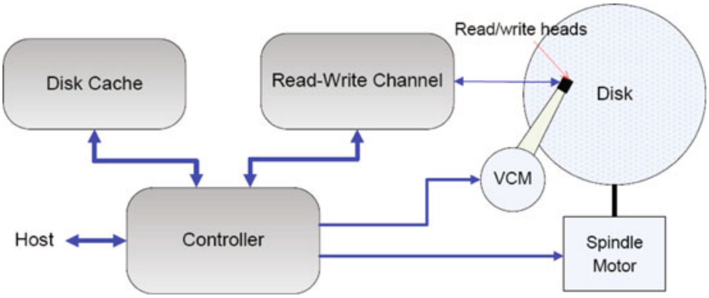


Fig. 4.1 Illustration of a hard disk drive structure

commercial hard disk drive with an area storage density of 2 Kb/in^2 in 1957, the areal storage density compound annual growth rate (CAGR) has been 30% for the first 35 years, jumped to 60% with the introduction of magnetoresistive (MR) heads in 1992, and further increased to around 100% with the introduction of giant magnetoresistive (GMR) heads in the late 1990s. In 2008, hard disk drives with an areal storage density of 610 Gb/in^2 were successfully demonstrated. Such areal storage density growth is enabled by continuous improvements across a variety of technological aspects, mainly including (i) thinner magnetic media with better properties, (ii) better and smaller write and read heads with smaller spacing between heads and media, (iii) more powerful magnetic recording read channel with sophisticated signal processing and coding, and (iv) more accurate head positioning servo.

Figure 4.1 illustrates a simplified hard disk drive structure. The read/write heads and magnetic recording disk are certainly the two most important components and largely determine the overall hard disk drive system performance. The motor controller drives both the spindle motor that rotates the disk and the voice coil motor (VCM) that rotates the actuator on which the read/write heads are mounted. The read–write channel carries out sophisticated digital and analog data processing, geared to the specific characteristics of recording media and read/write heads, to write data to and read data from the disk. The hard disk controller handles a variety of control and management functions such as interface between the hard disk drive and the host computer, disk cache management, and error recovery and fault management, and the disk cache is used as a buffer between host and physical magnetic recording media to improve the overall hard disk drive system performance. This section aims to provide a brief overview of major components in modern hard disk drives.

Hard Disk Drive Recording Media

A recording media is a rotating disk with ferromagnetic surface plane that, in current practice, is a uniform magnetic film in which each bit is stored across a few hundred magnetic grains that are magnetically isolated. As illustrated in Fig. 4.2, the media magnetic anisotropy can be either oriented in the recording medium plane, referred

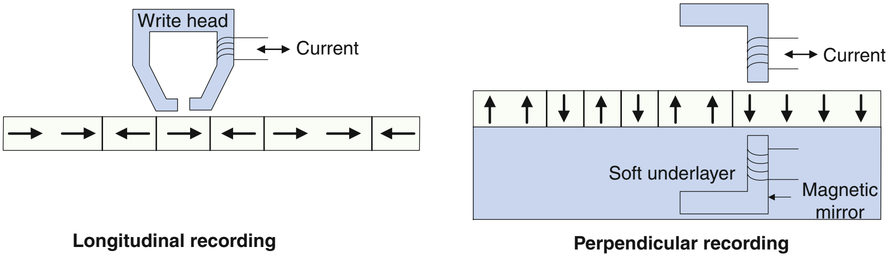


Fig. 4.2 Illustration of longitudinal recording and perpendicular recording

to as longitudinal recording, or aligned perpendicular to the recording medium plane, referred to as perpendicular recording [27].

The essential objective of magnetic recording media design is to maximize the areal density (i.e., minimize the footprint of each bit on the surface of storage media) and meanwhile ensure sufficient magnetization reliability. Magnetization reliability depends on the product of anisotropy constant K_u and volume V of the magnetic grain, called anisotropy energy K_uV. A larger anisotropy energy K_uV means a higher energy barrier for switching the magnetization direction. As we reduce the grain volume V to increase the areal storage density, the anisotropy energy will accordingly reduce. If it becomes even comparable to the thermal energy k_BT, where k_B is the Boltzmann constant and T is the temperature, the magnetization of grains may not be able to guarantee its stability over a long period such as 10 years. In general, hard disk drives should maintain a large ratio (e.g., 60 and larger) between anisotropy energy K_uV and thermal energy k_BT. In theory, perpendicular recording could achieve a higher storage density than its longitudinal counterpart for two main reasons: (i) By aligning the magnetic anisotropy perpendicular to the media surface, we could reduce the footprint of each bit while maintaining a sufficiently large magnetic region with the use of a thicker film; (ii) Aided with the soft magnetic underlayer (SUL) in perpendicular recording as illustrated in Fig. 4.2, the write head field gradient can be stronger, which makes it possible to employ storage material with a higher anisotropy constant K_u. Moreover, as the areal density increases, the demagnetization field of the magnetic grains, which is in opposition to the magnetization of the grains, will decrease. Another advantage of perpendicular recording is that under the same areal density, it is able to produce a larger playback signal compared with longitudinal recording, leading to better signal-to-noise ratio (SNR).

In spite of the above theoretical advantages of perpendicular recording, practical realization of suitable perpendicular recording media is much more complicated than its longitudinal counterpart [9]. As a result, longitudinal recording has been dominating the commercial hard disk drives until recently when the industry began to ship perpendicular recording hard disk drives in 2005. Since then, perpendicular recording has gained an ever increasing momentum, e.g., Hitachi demonstrated 610 Gb/in^2 perpendicular recording in 2008. Today, perpendicular recording is widely considered as a viable technology to approach 1 Tb/in^2 storage density, and the entire industry is quickly switching from longitudinal recording to perpendicular recording.

Currently, several techniques are being actively researched to further enhance the potential of perpendicular recording, among which two promising candidates are heat-assisted magnetic recording [57] and bit-patterned media [32]. In heat-assisted magnetic recording, magnetic properties of recording media monotonically depend on the temperature, and the magnetization can be more easily switched under a higher temperature. The write process exploits this effect to facilitate the data recording by heating up the recording media to a sufficiently high temperature while the write head field is applied and then rapidly cooling down the heated region to an ambient temperature.

In bit-patterned media, the conventional uniform magnetic thin film is replaced by a uniform array of isolated single-domain magnetic islands, each island stores one bit. Since the volume of each island can be much larger than the magnetic grains used in conventional recording media, bit-patterned media can achieve a much better thermal stability. Moreover, by pre-defining the position of each bit, bit-patterned media is not subject to bit position/boundary randomness as in conventional technology. To achieve a sufficiently high-areal storage density, these isolated single-domain magnetic islands should be very closely packed and each island should have a small footprint, e.g., see Fig. 4.3. One of the most critical challenge of bit-patterned media is how to economically realize such a fine-grained patterning, for which several advanced lithography technologies such as nano-imprint or X-ray lithography and self-assembly technologies are currently being actively explored. Accurate synchronization of the write data to the array of magnetic islands is another challenge.

Regardless of the specific recording technology, data on each disk is always organized in a hierarchical structure. First, the disk is partitioned into concentric circles, each circle is called a *track*, and then each track is further partitioned into a certain number of *sectors*. Sector is the basic indivisible storage unit in hard disk drives. Today, each sector contains 512-byte user data, while the industry now is in a transition to a 4096-byte user data sector format. Since the disk rotates at a constant speed, if all the tracks use the same maximum possible recording density, read and write heads will have to read from and write to different tracks at different speeds. Given the hundreds of thousands of tracks on one disk, such an ideal scheme will

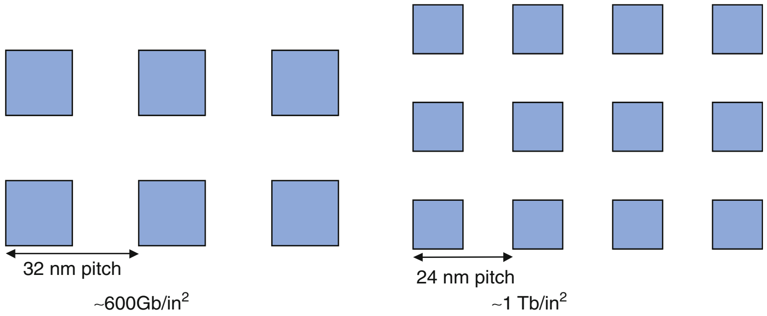


Fig. 4.3 Illustration of the required island pitch to achieve 600 Gb/in^2 and 1 Tb/in^2 storage density

make read and write circuit prohibitively complex. One simple solution used in the early days is that enforce all the tracks have the exactly same number of sectors; hence read and write circuits always operate under the same speed regardless the position of heads on the disk. This nevertheless resulted in a significant loss of storage capacity. As a compromise, a technique called zone-bit recording is typically used today. The basic idea is to partition the disk surface into a few concentric zones, and all the tracks within the same zone have the same number of sectors.

Magnetic Recording Heads

Magnetic recording write heads are responsible to write channel signals on the recording medium, and read heads are responsible to sense and convert the information on the medium to an electrical signal. Although the same inductive transducer was used to both write and read during the early days, modern hard disk drives use separate inductive transducer write heads and flux-sensing read heads. An inductive write head contains a core of magnetically soft material around which a coil of wire is wrapped. As illustrated in Fig. 4.2, in longitudinal recording, the ring-shaped core has a very short gap and, when a current passes through the coil, the fringing flux of the gap is used to write the media, leading to the magnetization oriented in the recording medium plane; while in perpendicular recording, the medium is located in the gap formed by the write head and the underlying SUL, leading to the magnetization perpendicular to the recording medium plane.

Both longitudinal and perpendicular recording use the same basic read head design that utilizes the MR or GMR effect (i.e., the MR or GMR sensor electrical resistance changes as a function of externally applied magnetic field) to sense magnetization transition recorded on the media. By steering a constant current through the MR or GMR sensor, the resistance change is converted to a voltage signal that is fed to the read channel. In both longitudinal and perpendicular recording, read head aims to sense the change of magnetization between adjacent magnetization regions. Let $g(t)$ represent the isolated transition response, i.e., the signal pulse due to a single change in magnetization direction; the noiseless readback waveform can be expressed as

$$z(t) = \sum_{k} (b_k - b_{k-1})\, g\,(t - kT),$$

where b_k represents the bits recorded and T is the spacing of a single bit. In longitudinal recording, the isolated transition response $g(t)$ is typically modeled as a simple single-parameter Lorentzian pulse:

$$g(t) = \frac{1}{1 + \left(\dfrac{2t}{\mathrm{PW50}}\right)^2},$$

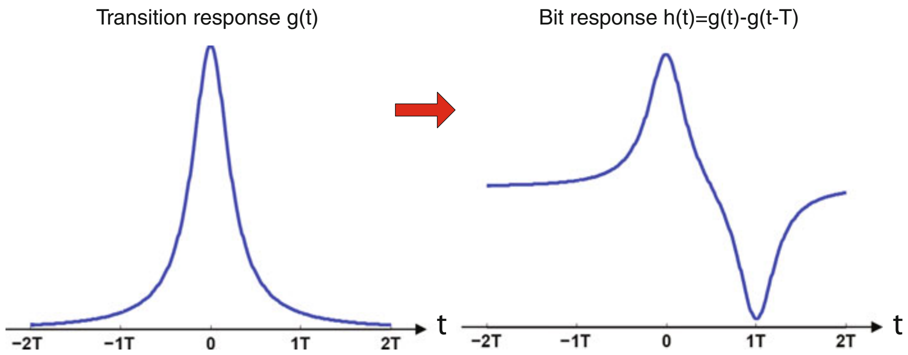


Fig. 4.4 Transition response and the corresponding bit response shapes in longitudinal recording

where the parameter PW50 represents the pulse width at the 50% of the maximum amplitude. In perpendicular recording, the isolated transition response $g(t)$ can be approximated as

$$g(t) = \mathrm{erf}\left(\frac{\sqrt{\ln 16} \cdot t}{\mathrm{PW50}}\right),$$

where $\mathrm{erf}(\cdot)$ is the error function defined as $\mathrm{erf}(x) = \left(2/\sqrt{\pi}\right)\int_0^x \mathrm{e}^{-z^2}\mathrm{d}z$ and PW50 is the pulse width of the derivative of $g(t)$ at the 50% of the maximum amplitude. Equivalently, the noiseless readback waveform can also be expressed in terms of the bit response $h(t) = g(t)–g(t–T)$ as

$$z(t) = \sum_k b_k h\left(t - kT\right).$$

Figures 4.4 and 4.5 illustrate the corresponding waveforms of *g(t)* and *h(t)* in the longitudinal recording and perpendicular recording, respectively.

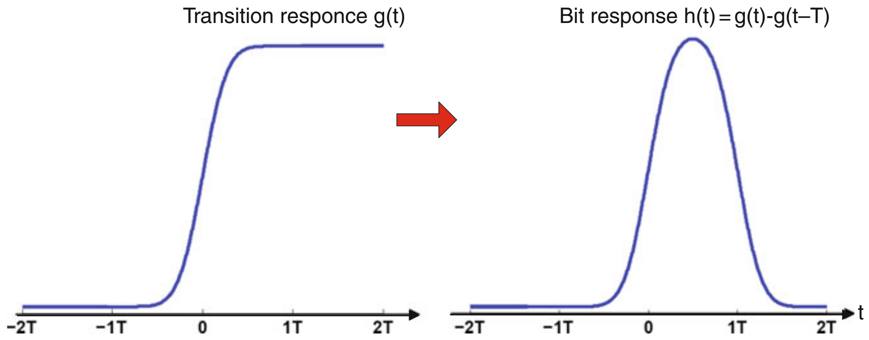


Fig. 4.5 Transition response and the corresponding bit response shapes in perpendicular recording

It is clear from the above figures that the bit response is (much) longer than the spacing between two adjacent bits, T. This means the signal sensed by the read head at each bit position depends on both the current bit and the bits before/after the current bit, which is referred to as inter-symbol interference (ISI). Typically, the significance of ISI is represented by the normalized density D, which is defined as the ratio between PW50 and T, i.e., $D = \mathrm{PW50}/T$.

Finally, we note that one critical parameter affecting the write/read performance, and hence areal storage density is the spacing between the heads and media, referred to as *head fly height*. To sustain the continuous growth of storage density, the head fly height has been consistently shrinking and is only a few nanometers in modern hard disk drives [12]. To maintain such a small head fly height without direct friction between heads and disk, air bearing is used to make the heads ride hydrodynamically on a cushion of air over the disk surface. Moreover, a very thin layer of polymeric lubricant layer is deposited on the disk surface to protect the magnetic material from potential scratches.

Read–Write Channel

As the interface between the read/write heads and hard disk drive controller, read–write channel performs necessary data processing, geared to the characteristics of recording media and heads, to ensure data storage integrity in the presence of inevitable mechanical, magnetic, and electronic inaccuracy and noises. Figure 4.6 shows a simplified structure of a write channel, which mainly carries out modulation encoding, write pre-compensation, and write current driving. Modulation encoding imposes a certain constraints into the bit stream to be recorded in order to facilitate the timing recovery and signal detection during the read process. The most well-known constraints are the run length constraints, which is realized by run length limited (RLL) coding. With two parameters d and k, an RLL(d, k) code enforces that any two 1 s are separated by at least d and not more than k 0 s. Since a 1 is recorded on the hard disk as a magnetization transition, the d-constraint could limit the significance of ISI, and the k-constraint aims to provide sufficiently frequent transitions to improve timing recovery and automatic gain control during the read process. Besides run length constraints, modulation encoding may also enforce some other constraints in order to improve signal detection performance. For example, read channel employs PRML (partial response maximum likelihood) signal detection that uses a Viterbi detector. The detection errors of a Viterbi detector can be dominated by several different error patterns. Hence, certain constraints

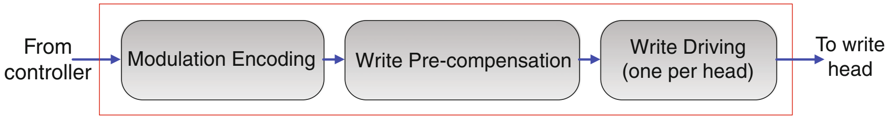


Fig. 4.6 Illustration of write channel structure

can be imposed into the bit stream, which can help to capture the occurrence of such dominant error events during the read process. In this case, the simplest constraint is the parity check constraint. Over the years, a significant amount of research efforts have been devoted to developing various modulation coding techniques for hard disk drives, and interested readers are referred to [24] for detailed discussions.

The purpose of write pre-compensation is to help mitigate nonlinear ISI. If two magnetization transitions are very close, the demagnetizing field from the previous transition will cause the current transition to be shifted during writing. The amount of shift depends on the pattern of transitions immediately preceding the current transition. The basic idea of write pre-compensation is to intentionally introduce a time shift of the write transition during the write process to compensate the effect of the nonlinear transition shift due to previous transitions. Finally, the write current is delivered to the write head through a write current driver in write channel.

Compared with write channel, read channel has a much more complicated structure. The sensed readback signal is amplified by the preamplifier, and then passes the automatic gain control (AGC) circuit to adjust the peak value and a continuous time low-pass anti-aliasing filter. The subsequent ADC (analog-to-digital converter) and timing recovery circuits sample the readback signal and send the digitized signal to the following signal processing and coding functions. Read channel will be discussed in more detail in sections "Modeling of Magnetic Recording Channels" and "Signal Detection and Decoding for Magnetic Recording Channel."

Controller

As the brain of hard disk drives, the controller ensures the entire hard disk drive functions and responds to the host computer correctly. The major functions handled by the hard disk controller include (1) interfacing with the host computer, (2) motor control, (3) managing cache memory, and (4) error correction coding and defect management.

Host Interface

The controller interfaces with the host computer, which can be a server, personal computer, or microcontroller in a consumer product, through a standard data communication protocol. Over the years, a variety of industry standard protocols have been developed, such as serial and parallel ATA and SCSI [26], which defines physical signal transfers, required registers, and command sets. ATA and SCSI have different costs vs. performance trade-off and hence are being used in different systems. With a lower cost, ATA is primarily used in low-end small-scale systems with only few devices to be connected, while SCSI is typically used in high-end larger-scale computing systems because of its flexibility and superior performance in a multitasking and/or multi-user environment. Early versions of ATA and SCSI are parallel with an 8-bit or 16-bit data bus, which has been recently replaced by their serial counterparts that can reduce the cable-bulk and cost and meanwhile increase data transfer rate (e.g., up to 6 Gbit/s data transfer rate can be realized by serial ATA and SCSI).

To reduce design cost and improve flexibility, a hard disk drive controller may support multiple interface protocols. Moreover, it is not uncommon for a controller to embed certain special purpose hardware to handle a small set of critical interface protocol commands, in order to ensure system performance and reduce the load of main processor in the controller.

Motor Control

As illustrated in Fig. 4.1, the controller controls two motors, spindle motor and VCM. The spindle motor spins the disk at a constant rotational speed, while the VCM aims to move the read/write heads to the desired location on the disk very precisely. The controller controls the operation of these two motors through a feedback loop by constantly monitoring the current status/position of the disk and heads based on which it generates control signals to the motors. To facilitate accurate and fast head positioning, periodic servo fields are inserted on the disk surface, as illustrated in Fig. 4.7.

Those servo fields are written on the disk surface once when the disk is manufactured; hence during the run-time, the controller must ensure the servo fields will never be accidentally overwritten by the write head. Since the servo fields occupy the disk surface that would be otherwise available to record user data, there is a design trade-off on how many servo fields should be embedded. In most hard disk drives, the servo fields typically occupy 5–10% of the disk surface.

Cache Memory Management

Modern hard disk drives contain a disk cache DRAM (dynamic random access memory), typically ranging from 8 to 64 MB, as a buffer to streamline the data transfer between the host and the hard disk. The appropriate use of cache can greatly improve the hard disk drive response time and help reduce hard disk drive energy

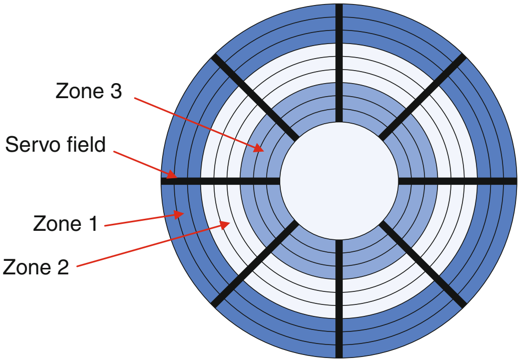


Fig. 4.7 Illustration of a disk with eight servo fields along each track and three zones

consumption. The disk cache can serve for both write and read I/O requests. In the context of write caching, the write data from the host are not immediately recorded on the disk; instead the data are transferred to the disk cache. Since DRAM has a much less write latency than hard disk, the controller can respond to the host to acknowledge the write much earlier, thus allowing the host to quickly continue its own operations without a long wait. However, before being actually recorded on the disk, the cached write data are *dirty* and will result in data storage integrity failure if power is somehow lost. When the write cache becomes (almost) full, a part or all of the data will be flushed to the hard disk, preferably during the idle period of hard disk to minimize the impact on the hard disk drive performance. Besides the obvious advantage of accelerating write acknowledge, the use of write cache can bring the following two advantages: (i) Given a relatively large amount of data to be written, the hard disk drive controller has a high flexibility to schedule the order of the write operations to minimize the hard disk drive write energy consumption and/or latency. By exploiting the proximity among all the data to be written, an appropriate write order scheduling can effectively reduce the overall disk/head rotation time and energy consumption. Over the years, many scheduling algorithms have been developed and their effectiveness has been well demonstrated (e.g., see [8, 17, 66]). (ii) For applications that tend to frequently update the same data, these multiple writes can be filtered out by the write cache, leading to a reduced total number of actual disk recording operations.

Read cache holds a very small portion of the data stored on the disk, which may be soon read by the host with a relatively high probability. Hence, for each host read command, the controller always checks whether the requested data already reside in the disk cache; if yes (i.e., a cache hit), then the controller simply fetches the data from the cache instead of executing the actual disk read operation. Clearly, if the cache hit rate is high enough, read cache can largely improve the overall hard disk drive system performance. To improve the read cache hit rate, good prefetching algorithms should be used for effectively exploiting the temporal and spatial locality presented in the host data access (e.g., see [53, 60]).

In today's hard disk drives, a unified DRAM is used to serve both write caching and read caching, in which the percentages of memory dedicated to read caching and write caching are dynamically configurable. This could make the disk cache dynamically adaptable to the run-time work-loads with different read and write characteristics. Finally, we note that although disk cache predominantly uses DRAM in current practice, there have been much recent discussions on using non-volatile memory, particularly NAND flash memory, to replace DRAM as disk cache in hard disk drives. The potential advantages include write caching without data integrity issue, faster host boot process by pre-loading critical operating system files in the non-volatile disk cache, and reduced overall hard disk drive power consumption.

Error Correction and Defect Management

All the hard disk drives employ error-correcting codes (ECCs) to ensure their storage integrity. The key of ECC is to appropriately introduce a certain amount of redundant information, which can be exploited to detect and correct errors in the retrieved data. Over the past several decades, a very rich family of various ECCs and their efficient encoding/decoding have been developed [6, 40], and ECC has been playing a critical role in numerous data storage and communication systems. Reed–Solomon (RS) codes [65] are used as ECCs in most of today's hard disk drives. RS codes are a class of non-binary linear block codes defined over Galois Fields (GFs). With an underlying GF(2^m), an (n, k) RS code (i.e., the codeword contains n m-bit symbols out of which k symbols carry the user data and the other $n–k$ symbols are redundant data) satisfies $n \leq 2^m - 1$ and $n - k \geq 2t$, where t is the number of erroneous symbols that can be corrected. The objective of RS decoding is to identify the locations of the erroneous symbols and recover their correct values. Because of the use of Viterbi detection in the read channel and bursty nature of recording media defects, the errors in the output of read channel tend to be bursty. The popularity of RS codes in hard disk drives mainly attributes to their non-binary nature and hence good burst error correction capability (i.e., since each symbol contains m bits, an RS code can correct up to $m \cdot t$-bit burst errors). Moreover, there are efficient decoding algorithms for RS codes, e.g., the well-known Berlekamp–Massey and Euclidean algorithms [6, 40], which makes it possible to satisfy the decoding speed requirement at reasonable silicon and energy cost. Furthermore, it is possible that the locations of some erroneous symbols can be identified beforehand by the read channel. Such errors with known locations are called *erasures*. Let t and s denote the number of errors and erasures, decoding of an (n, k) RS code will succeed as long as $n–k \geq 2t–s$. Although the error correction capability of RS codes can be directly improved by increasing the amount of coding redundancy (i.e., decreasing the code rate k/n), this will nevertheless reduce the disk surface used to store real user data. Hence, the code rate has to be carefully selected so that the overall effective storage density can be maximized while satisfying the desired storage reliability. As a result, hard disk drives typically use RS codes with relatively high code rates, e.g., 9/10 and higher.

As the areal storage density continues to grow, the raw bit error rate (BER) after the read channel tends to increase, which in turn demands the use of more powerful ECCs. One simple solution is to use a longer RS code, i.e., its codeword length is longer. In general, to achieve the same target decoding failure rate, the longer the codeword length is, the higher code rate can be used. This has motivated the industry to switch from the traditional 512-byte user data per sector to 4096-byte user data per sector. However, it should be pointed out that the decoder implementation complexity and decoding latency will accordingly increase as we increase the codeword length. Encouraged by the success of iterative codes such as Turbo codes [4] and LDPC (low-density parity-check) codes [16] in wireless communications, researchers have been heavily investigating on applying such iterative codes to hard disk drives, which may complement with or even completely replace RS codes. However, due to the lack of accurate analytical methods, it remains a challenge to predict the error-correcting performance of such iterative codes under the very low decoding failure rates as demanded by hard disk drives (e.g., 10^{-10} and below). Moreover, decoders of these iterative codes require the read channel signal detector to deliver soft output, which may largely increase the read channel implementation complexity and power consumption.

To further assure the data storage integrity, a cyclic redundancy checksum (CRC) is also generated and recorded together with each sector of user data. This can be used to identify those undetected ECC decoding failures. In case of ECC decoding failures detected by either ECC itself or CRC check, the controller will issue a retry command to reread the same sector. Such reread may be tried several times, and various read head and read channel parameters may be adjusted. Due to the obviously big impact on the system performance, hard disk drives demand a very low retry rate of about 10^{-14}.

No matter how powerful the ECC is, there are always some sectors that contain too many defects to be successfully corrected by the ECC. Such sectors are called defective sectors. Those too many defects may be due to several possible causes, such as disk surface scratches, insufficient magnetic coating material at some spots, and deterioration of magnetic materials. Moreover, regardless to their causes, defects can be either primary defects, which are detected during the hard disk manufacturing, or grown defects, which develop during the lifetime of the hard disk drives.

Once a defective sector is identified, the controller should manipulate the logical to physical address mapping in such a way that this defective sector will never be used to store data. In case of defective sectors detected during manufacturing, the controller simply removes all those defective sectors from the physical address space and makes the non-defective sectors to cover a continuous physical address space. However, in case of defective sectors detected in the field, the controller has to relocate the physical address of each defective sector to another spare sector.

Modeling of Magnetic Recording Channels

Read channel is a critical block in hard disk drives. It is responsible for performing a variety of operations including writing the data bits onto the magnetic medium and recovering the stored data from the medium during readback. The main modules in a read channel chip are [see Fig. 4.8] encoders for protecting the user data bits against distortions in the recording channel, a write precompensation module for minimizing media-induced non-linear distortions during the write process, a loop module for compensating the readback signal for timing/gain/offset errors, a filtering and equalization module for minimizing noise and equalizing the playback signal to a desired shape, a detection module for recovering the stored data bits from the equalized signal, and decoders for decoding the detected data into the original user data bits. To get the best performance from a hard disk drive, these modules in the read channel should be optimally configured according to the specific characteristics of the recording channel consisting of the recording medium and read–write heads. This necessitates the need for accurate modeling of the recording channel so that the resulting channel model encompasses all the key characteristics of heads and media from a signal processing perspective.

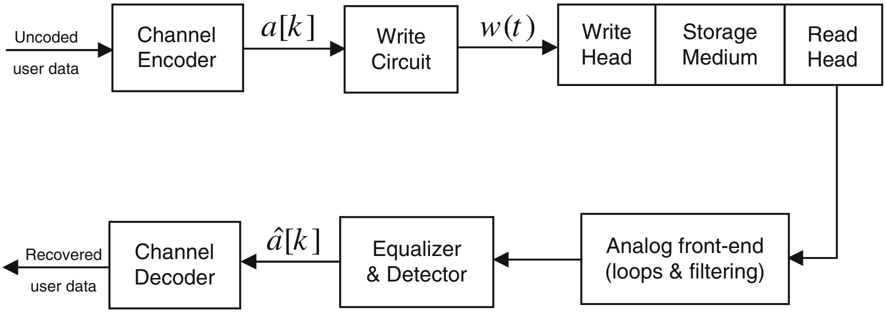


Fig. 4.8 Block schematic of a hard disk drive system

While technological innovations in heads and media are mainly responsible for the growth in storage capacity in hard disk drives, the role of advanced coding and signal processing techniques realized through read channels is critical not only for supporting the advanced heads–media but also as a cost-efficient means for enhancing storage capacity. As conventional magnetic recording technology used in hard disk drives is approaching physical limits and as alternative magnetic recording technologies are being explored, the role of advanced read channel techniques becomes even more critical in maintaining the growth in storage capacity. This underscores the utmost importance of accurate modeling of the recoding channel, and this section deals with this topic.

Key Characteristics of Recording Channel

The various characteristics of recording channel can be classified into categories such as linear vs. non-linear, deterministic vs. stochastic, data-dependent vs. data-independent (here, "data" refer to the data bits written on the medium), stationary vs. non-stationary, with memory vs. without memory, and write-path vs. read-path. We shall briefly list the key channel characteristics here and we will elaborate on these in subsequent sections.

- Bit response: This is the response of the recording channel to a single bit at its input. Duration of this response indicates the memory of the channel. Impulse response and step response are equivalent representation of the recording channel in place of bit response. Ideal readback signal (i.e., noiseless and distortionless) is obtained by convolving the encoded data bits with bit response.
- Electronics noise: This noise is generated by electronic devices, such as preamplifier that conditions the readback signal before passing it to the read channel. Shot noise, thermal noise, and flicker noise constitute the main components of electronics noise [63]. It is data independent and is often modeled as a white noise, recognizing the dominance of shot/thermal noises since flicker noise is

important only at very low frequencies. The most simplified model of recording channel consists of a signal term due to bit response and an additive white noise term due to electronics noise.

- Head noise: There are two types of head noise: Barkhausen noise and Johnson noise [63]. The former arises from fluctuations of magnetic domain walls of the head core material and the latter arises from resistive dissipation in the head. These are also data-independent noises.
- Medium noise: This noise is generated by the recording medium and has several components. The main components are transition noise, DC noise, and modulation noise [5, 68]. Transition noise occurs due to fluctuation in the grain magnetization at the written transition boundaries, and it manifests in the form of position-jitter and width-variation in the written transition, resulting in jitter noise and width-variation noise, respectively. Clearly, transition noise is dependent on the written data and is also non-stationary. DC noise occurs due to randomness in grain sizes and grain center locations, and it is a data-independent stationary noise. Modulation noise occurs due to magnetic fluctuations taking place between magnetic transitions and is hence a non-stationary data-dependent noise. Texture noise is an important form of modulation noise and is associated with the mechanical texture of the disk substrate.
- Write-related non-linear distortions: There are several non-linear distortions that occur during the writing process. The important distortions are non-linear transition shift (NLTS), partial erasure (PE), transition broadening (TB), and overwrite [48, 51, 62]. NLTS and TB refer to shifts and broadening, respectively, of written transitions due to the influence of the demagnetizing field from previous transitions on the head field writing the current transition. Bandwidth limitations of the write-path also contribute to NLTS [3]. PE refers to partial erasing of a transition and the resulting loss in signal amplitude when closely spaced transitions are written. Overwrite or hard transition shift (HTS) refers to shift in written transitions due to the influence of old information residing on the medium. These distortions are clearly data dependent and approximately deterministic in nature.
- Read-related non-linear distortions: Two major non-linear distortions during readback are that due to the non-linear transfer function of the magneto-resistive (MR) read-head and that due to the presence of asperities on the medium [59, 72]. When MR head comes in contact with the medium because of the presence of an asperity, a resistance transient is induced in the head resulting in a sudden change in the readback signal and is referred to as MR thermal asperity (TA). The non-linearity of the transfer function of a given MR head is clearly a data-independent deterministic distortion. The occurrence and intensity of TA distortions are data independent and random in nature, due to randomness associated with the location and origin of asperities.
- Media defects: The presence of defects on the medium causes the readback signal to be attenuated (known as drop-outs), and the degree/profile of attenuation and duration of this depend on the severity of the defect [61]. The occurrence and intensity of drop-outs are data independent and random in nature, due to randomness associated with the location and origin of defects.

In the next few sections, we will elaborate on each of the above characteristic. Our emphasis will be on providing simple and sufficiently accurate models for representing each of these in numerical simulation of the recording channel. Before we get into the details, we shall first describe some general principles that would help to clarify the similarity between recording channels and communication channels and lay the foundation for the discussion in subsequent sections.

Channel Modeling Preliminaries

Figure 4.8 shows the essential blocks of a hard disk drive from a signal processing perspective. Here, $a[k]$ denotes the channel encoded data bits in $\{-1,+1\}$ format, $w(t)$ denotes the current waveform supplied to the write head, and $\hat{a}[k]$ denotes the data bits recovered/detected from the readback signal. The signal path from the input of "write head" to the output of "read-head" is what is referred to as "recording channel" and this is what we need to model for simulation and design of read channels.

Hard disk drives make use of the so-called saturation recording for storing binary data bits on magnetic medium. That is, bit "+1" or "-1" is stored by magnetizing a selected small region on the medium in one direction or in the other. This is done by supplying current pulses of appropriate polarity to the inductive write-head so that the resulting head field will be in the correct direction to magnetize the medium. Thus, there is a one-to-one correspondence between the magnetization pattern on the medium and the sequence of data bits. Upon readback, a MR read-head senses the magnetization stored on the medium, converts that into a voltage signal, and gives it to the preamplifier. The ideal readback signal can be mathematically represented as [48]

$$r_{\mathrm{s}}(t) = \frac{\mathrm{d}w(t)}{\mathrm{d}t} \otimes h_{\mathrm{s}}(t) \qquad \text{with} \qquad w(t) = \sum_k a[k] s\,(t - kT), \tag{4.1}$$

where $h_s(t)$ denotes a readback system response which is related to the head-medium parameters, $w(t)$ denotes the current supplied to the write-head, T denotes the duration of one bit, and $s(t)$ denotes an ideal unit-amplitude rectangular current pulse of duration T located at $t = 0$. Using the expression for the write current waveform, we can obtain

$$r_{\mathrm{s}}(t) = \sum_k a[k] h_{\mathrm{b}}\,(t - kT), \tag{4.2}$$

where

$$h_{\mathrm{b}}(t) = h_{\mathrm{s}}(t) - h_{\mathrm{s}}\,(t - kT)\,. \tag{4.3}$$

Here, $h_{\mathrm{b}}(t)$ is known as "bit response" of the recording channel since it is the read-head output for a single bit "+1" at the input of write circuit. It is also called "pulse response" since a single bit at write circuit input corresponds to pulse $s(t)$ of one bit duration at the input of write-head. Correspondingly, $h_{\mathrm{s}}(t)$ is known as "step response" of the channel since the pulse $s(t)$ is obtained by subtracting a unit step input shifted by T from the unshifted one. Finally, "impulse response" $h_{\mathrm{i}}(t)$ of the channel, which is the readback response when a Dirac delta function is applied at write-head input, is given by the time-derivative of step response as

$$h_{\mathrm{i}}(t) = \frac{\mathrm{d}h_{\mathrm{s}}(t)}{\mathrm{d}t}. \tag{4.4}$$

Observe from (4.2) that the magnetic recording channel resembles a binary pulse amplitude modulated (PAM) base-band communication channel.

Linear Channel Models

The description of a noiseless linear recording channel is complete if its bit response, step response, or impulse response are specified. All of these three responses are equivalent. Commonly used models for step response and corresponding impulse response in perpendicular recording are [18, 61, 69]

$$h_{\mathrm{s}1}(t) = A_0 \mathrm{erf}\left(\frac{2\sqrt{\ln 2}}{\mathrm{PW}_{50}}t\right), \quad h_{\mathrm{i}1}(t) = A_0 \frac{4\sqrt{\ln 2}}{\sqrt{\pi}\mathrm{PW}_{50}} \exp\left(-\frac{4\ln 2}{\mathrm{PW}_{50}^2}t\right), \tag{4.5}$$

$$h_{\mathrm{s}2}(t) = A_0 \tanh\left(\frac{\ln 3}{T_{50}}t\right), \qquad h_{\mathrm{i}2}(t) = A_0 \frac{\ln 3}{T_{50}} \mathrm{sech}^2\left(\frac{\ln 3}{T_{50}}t\right), \tag{4.6}$$

$$h_{\mathrm{s}3}(t) = A_0 \tan^{-1}\left(\frac{2}{T_{50}}t\right), \qquad h_{i3}(t) = A_0 \frac{2 \cdot T_{50}}{T_{50}^2 + 4t^2}, \tag{4.7}$$

where T_{50} denotes the time required for the step response to rise from –50% to + 50% of the saturation amplitude and PW_{50} denotes the pulse-width of the impulse response at 50% of its peak amplitude. The ratios T_{50}/T and PW_{50}/T are used as measures of the normalized linear bit density from a signal processing perspective. The higher these numbers are, the larger will be the memory of the channel and ISI caused by the channel, and vice versa. Figure 4.9 shows the step response and the resulting bit response for the error function model given in (2.5) for PW_{50}/T equal to 1.5 and 3.0. Observe that as PW_{50}/T increases, the length of ISI increases and amplitude of the bit response decreases.

Head Noise and Electronics Noise

The Barkhausen-type head noise arises in both inductive and MR heads. As mentioned already, it is caused by large changes in domain wall structure in the thin films in response to magnetic fields or mechanical stresses. This noise can be minimized

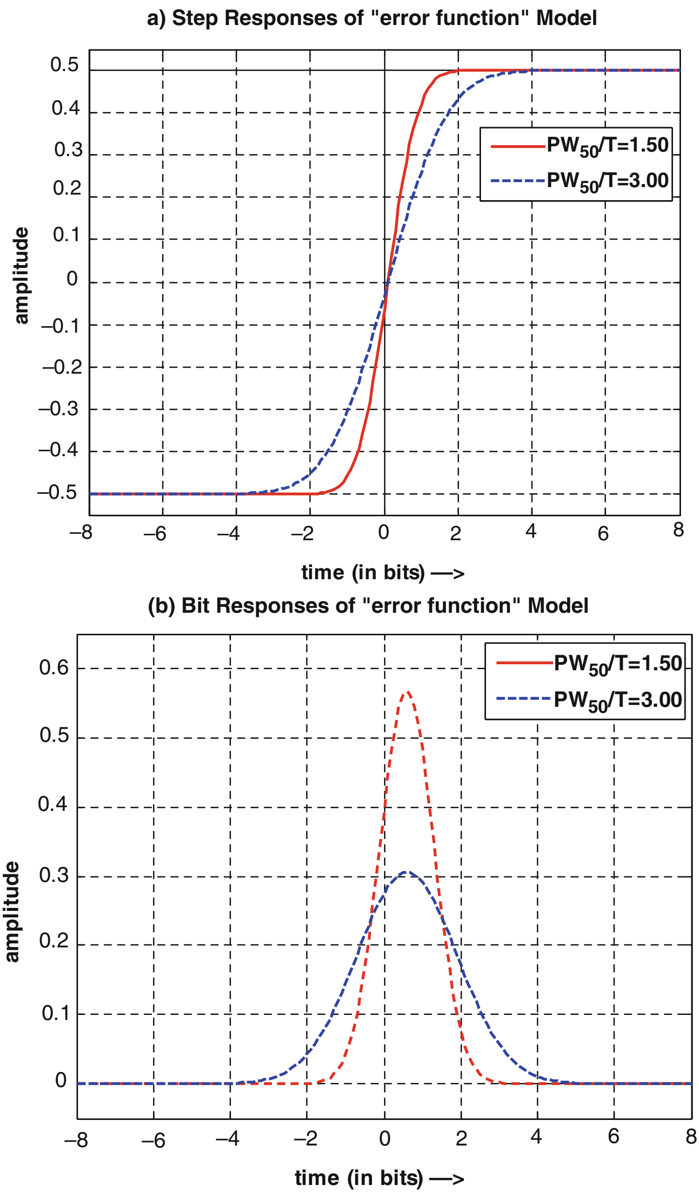


Fig. 4.9 Error function model for perpendicular recording channel with normalized linear bit density 1.5 and 3.0. (**a**) Step response and (**b**) bit response

by careful micro-magnetic design of the head [47]. The source for Johnson (or thermal)-type head noise is the resistive component of the head impedance. The resistive part dissipates energy and generates noise [47]. The root mean square (rms) value of the resulting Johnson noise is given by

$$V_{\mathrm{rms}} = \sqrt{4k_{\mathrm{b}}T_0\mathrm{R}e(Z)\Delta f}, \tag{4.8}$$

where k_b denotes Boltzmann's constant, T_0 denotes the absolute temperature, $\mathrm{R}e(Z)$ denotes the real part (i.e., resistive component) of the head impedance, and Δf denotes bandwidth. In MR heads, Johnson noise is determined almost solely by the electrical resistance of the MR sensor.

Electronics noise, which is a random signal characteristic of all electronic devices, is caused by random fluctuations in time of the electric charge carriers. As mentioned already, the main components of electronics noise in magnetic recording are thermal (or, Johnson) noise and shot noise. In a given device, thermal noise can be described by an equation similar to (4.8), while shot noise is related to the average current through the device. Over the frequency range of interest in practical systems, the combined thermal and shot noise can be modeled by an additive white Gaussian process $\varepsilon_e(t)$ with double-sided power spectral density by $N_e\big/2$. The resulting model of linear recording channel with head/electronics noise can be given as

$$r_{\mathrm{l}}(t) = \sum_k a[k]h_b\left(t - kT\right) + \varepsilon_e(t). \tag{4.9}$$

The flicker noise component, which arises due to device imperfections, has power spectral density that is proportional to $1\big/f^{\alpha}$ with $0.5 < \alpha < 2$.

Medium Noise

As highlighted earlier, the main components of media-generated noise are transition noise, DC noise, and texture noise. Since transition noise can be modeled using position-jitter and width-variation in written transitions [71], readback signal with transition noise can be modeled as

$$r(t) = \sum_k b[k]h_s\left(t - kT + \tau[k], W + \Delta W[k]\right), \tag{4.10}$$

where $b[k] = a[k] - a[k-1]$, with $b[k] \in \{-2, 0, +2\}$, denotes the sequence of written transitions; W stands for PW_{50} or T_{50}; and $\tau[k]$ and $\Delta W[k]$ denote the position-jitter and width-variation, respectively, in the transition at kth bit instant. When the width-variation and position-jitter are sufficiently small, an approximate expression for transition noise can be obtained from (4.10) using first-order Taylor series approximation of $h_{\mathrm{s}}\left(t - kT + \tau[k], W + \Delta W[k]\right)$ resulting in [48]

$$n_{\mathrm{m}}(t) = \sum_k b[k]\left\{\tau[k]h_{\mathrm{i}}\left(t - kT\right) + \Delta W[k]h_{\mathrm{w}}\left(t - kT\right)\right\} \tag{4.11}$$

with

$$h_{\mathrm{i}}\,(t) = \frac{\partial h_{\mathrm{s}}(t)}{\partial t} \qquad \text{and} \qquad h_{\mathrm{w}}\,(t) = \frac{\partial h_{\mathrm{s}}(t)}{\partial W}, \tag{4.12}$$

where $h_{\mathrm{i}}\,(t)$ and $h_{\mathrm{w}}\,(t)$ are the impulse response and width response, respectively, of the recording channel. Extension of this model using second-order Taylor approximation is straightforward [48]. It is easy to observe from Eqs. (4.10), (4.11), and (4.12) that (i) transition noise depends on the written data sequence, (ii) it is strong in areas where there are more transitions and vice-versa, and (iii) it can be characterized by specifying the distributions of position-jitter $\tau[k]$ and width-variation $\Delta W[k]$ and step response $h_{\mathrm{s}}\,(t)$. It is common to model $\tau[k]$ and $\Delta W[k]$ as mutually independent Gaussian random variables that are independent of the data sequence $b[k]$. At very high linear recording densities, the distribution of $\tau[k]$ may need to be modified to support high jitter percentages with restricted distribution tails.

DC noise is modeled as a data-independent stationary colored noise with power spectral density given by that of channel impulse response [29]. The resulting expression for DC noise can be given by

$$n_{\mathrm{d}}(t) = \int_{-\infty}^{\infty} \varepsilon_{\mathrm{d}}(u) h_{\mathrm{i}}\,(t-u)\,\mathrm{d}u, \tag{4.13}$$

where $\varepsilon_{\mathrm{d}}(t)$ is a Gaussian white noise process with double-sided power spectral density $N_{\varepsilon}\big/2$. Using (4.9), (4.10), (4.11), (4.12), and (4.13), the model of linear recording channel with head/electronics noise and media noise can be given as

$$r(t) = \sum_k b[k] h_{\mathrm{s}}\,(t - kT + \tau[k], W + \Delta W[k]) + \varepsilon_e(t) + n_{\mathrm{d}}(t), \tag{4.14}$$

$$\approx \sum_k b[k] h_{\mathrm{s}}\,(t - kT, W) + n_{\mathrm{m}}(t) + \varepsilon_{\mathrm{e}}(t) + n_{\mathrm{d}}(t). \tag{4.15}$$

Write-Related Non-linear Distortions

As recording density increases, nearby magnetic transitions begin to interact and this leads to breaking down the linear nature of the channel. As introduced briefly earlier, the important non-linear distortions are NLTS, PE, TB, and HTS. A model of the noiseless readback signal incorporating these distortions can be given as [37, 52]

$$\begin{aligned} r_{\mathrm{n}}(t) = \sum_k \tilde{b}[k] h_{\mathrm{s}} \left(t - kT + \frac{\Delta_0}{4} b[k]\,(1 - a[k]) + \frac{\Delta_1}{4} b[k] b[k-1] \right) \\ + \frac{\Delta_b}{4} \sum_k \tilde{b}[k] \cdot b[k] b[k-1] h_{\mathrm{s}}''\,(t - kT), \end{aligned} \tag{4.16}$$

where $\tilde{b}[k]$ is a modified form of $b[k]$ due to PE given by

$$\left.\begin{aligned}\tilde{b}[k] &= \gamma^2 b[k] \text{ if } & b[k+1] \neq 0 \text{ and } & b[k-1] \neq 0\\ &= \gamma b[k] \text{ if } & b[k+1] \neq 0 \text{ or } & b[k-1] \neq 0\\ &= b[k] \text{ if } & b[k+1] = 0 \text{ and } & b[k-1] = 0.\end{aligned}\right\}. \tag{4.17}$$

Here, $0 < \gamma < 1$ denotes amplitude loss in transition due to PE, Δ_0 denotes the shift in current transition due to pre-existing magnetization in the direction of $a[k] = +1$ (i.e., HTS), Δ_1 denotes the shift in current transition due to a transition one-bit earlier (i.e., NLTS), and Δ_b models the broadening of current transition due to NLTS arising from immediately preceding transition. For the sake of simplicity, (4.16) considers the NLTS from the transition one-bit earlier only. Generalizing this to the case of patterns of transitions preceding the current transition is straightforward [39]. When transition shifts are small, (4.16) can be simplified using first-order Taylor series as

$$\begin{aligned} r_n(t) \approx \sum_k \tilde{b}[k] h_s(t-kT) + \frac{\Delta_0}{4} \sum_k \tilde{b}[k] b[k] (1-a[k])\, h_i(t-kT)\\ + \frac{1}{4} \sum_k \tilde{b}[k] b[k] b[k-1] \left\{\Delta_1 h_i(t-kT) + \Delta_b h_s''(t-kT)\right\}. \end{aligned} \tag{4.18}$$

Readback Non-linear Distortions

MR non-linearity and MR thermal asperity (TA) are the two major readback non-linear distortions. Non-linear transfer function of MR head manifests asymmetry and saturation [41, 72]. Asymmetry arises due to not choosing the operating point of the MR head in the center of the linear region, and saturation arises when input signal amplitude exceeds the linear region in both directions. If $r(t)$ denotes the output of ideal linear MR head, then the output of non-linear MR head can be modeled as

$$\tilde{r}(t) = r(t) + \alpha r^2(t) + \beta r^3(t), \tag{4.19}$$

where α and β denote the amount of signal asymmetry and stripe saturation, respectively.

MR TA caused by contact between MR head and a raised defect on the medium manifests in the readback signal as a change in baseline that is characterized by fast rise time, large peak amplitude, and approximately exponential decay [13, 59]. The resulting model for the baseline shift caused by TA can be given by

$$\left.\begin{aligned} c(t) &= A_0\, t/T_r & \text{for } 0 \leq t \leq T_r\\ &= A_0 \exp\left(-(t-T_r)/T_d\right) & \text{for } T_r < t \leq T_f \end{aligned}\right\}, \tag{4.20}$$

where A_0 denotes the peak TA amplitude, T_r and T_d denote the rise time and decay time-constant, respectively, and T_f denotes the duration of TA. This model for baseline shift is added to the normal readback signal to create readback signal with TA.

The peak amplitude in TA signal is proportional to the maximum average temperature rise in the stripe, and the decay time constant is related to the read-gap thickness. While this type of contact-based MR TAs constitutes the majority of occurrences of MR TAs, there are also non-contact-type MR TAs [59]. Manifestation or signature of non-contact MR TAs in readback signal is quite different from that of contact MR TAs.

Media Defects

Most commonly encountered media defects are characterized by loss of amplitude and are known as drop-outs. These are caused mainly by various types of defects on the medium and/or loose particles in the head-medium interface. The most common model used for modeling drop-outs modulates the amplitude of the readback signal based on certain profile [58, 61]. Parameters of the model include the depth of amplitude loss and duration of drop-out. The loss of amplitude may be modeled as sudden drop in amplitude or a continuous variation in amplitude over the defect length.

Generalized Channel Models

The models that we have considered in the above sections are closely tied to the physical mechanism underlying each of the distortion. There have also been other models proposed in the literature which are developed with the objective of capturing multiple distortions with a single generalized model. The data-dependent auto-regressive (DDAR) model [31] and Volterra series model [22] are typical examples for generalized modeling. While these models are very appealing from a signal processing perspective, it is difficult to relate the model parameters directly to physical mechanisms.

The DDAR model is very extensively used in magnetic recording. It is a concise parametric representation of the recording channel and provides comprehensive modeling coverage. The reason for its wide acceptance can be attributed to the several advantages it offers. It is able to model linear/non-linear deterministic distortions and stochastic noises with only a few parameters. Since there are only a few parameters, estimation of model parameters can be done rather fast with good accuracy. Most importantly, the form of the DDAR model points to the form of data detector that is optimum for such models [30]. We recall here the causal version of this model, while the non-causal version is available in [30]. Based on this model, the bit-rate sampled output of the channel is given by

$$z[k] = y\left(a_{k-I}^{k}\right) + v[k], \tag{4.21}$$

$$v[k] = \sum_{i=1}^{L} g_i\left(a_{k-I}^{k}\right) \cdot v[k-i] + w\left(a_{k-I}^{k}\right), \tag{4.22}$$

where $y\left(a_{k-I}^{k}\right)$ models the noiseless output of the channel which depends on the $I+1$ data bits $a_{k-I}^{k}=\{a[k-I],\ a[k-I+1],\ \cdots,\ a[k]\}$, I is the data memory length, and $v[k]$ models the additive noise component. The signal component $y\left(a_{k-I}^{k}\right)$ is constructed as a look-up table to incorporate both linear and non-linear channel distortions. The noise component $v[k]$ is the output of a data-dependent auto-regressive (AR) filter whose parameters, i.e., coefficients $g_i\left(a_{k-I}^{k}\right)$, $i=1,2,\ldots,L$, and standard deviation $\sigma_w\left(a_{k-I}^{k}\right)$ of uncorrelated Gaussian excitation $w[k]$, are dependent on the data bits a_{k-I}^{k}. Here, L denotes the Markov memory length of the noise model. Clearly, the noise $v[k]$ is both data-dependent and correlated, and hence non-stationary. The parameters of this DDAR model, i.e., $\left\{y\left(a_{k-I}^{k}\right),\ g_1\left(a_{k-I}^{k}\right),\ g_2\left(a_{k-I}^{k}\right),\ \cdots,\ g_L\left(a_{k-I}^{k}\right),\ \sigma_w\left(a_{k-I}^{k}\right)\right\}$ for all possible a_{k-I}^{k}, can be estimated using data-dependent mean and covariances and solving the Yule–Walker equations [31].

The Volterra series model [22] models the non-linear portion of the channel output as the sum of the outputs of non-linear kernels, where each kernel is driven by a product of selected data bits. The shape of the kernels and associated bit-products at their inputs depend on the types of non-linear distortions being modeled. As an example, considering non-linear distortions that result in non-liner combination of $\{a[k],\ a[k-1],\ a[k-2],\ a[k-3]\}$ at k th instant, the resulting Volterra model for the non-linear distortion components in channel output can be given by

$$\tilde{r}_n(t)=\sum_k\left\{\begin{array}{l} a[k]a[k-1]h_{2,1}(t-kT)+\ a[k]a[k-2]h_{2,2}(t-kT) \\ +\ a[k]a[k-3]h_{2,3}(t-kT) \\ +\ a[k]a[k-1]a[k-2]h_{3,1}(t-kT) \\ +\ a[k]a[k-1]a[k-3]h_{3,2}(t-kT) \\ +\ a[k]a[k-2]a[k-3]h_{3,3}(t-kT) \\ +\ a[k]a[k-1]a[k-2]a[k-3]h_{4,1}(t-kT) \end{array}\right\}, \tag{4.23}$$

where $h_{\mathrm{i,j}}(t)$ denotes the kernels associated with the various bit-products. These kernels can be estimated from measured data, e.g., by an adaptive training approach or through systematic cross-correlation approaches. It is straightforward to see that by adding the linear component and stochastic noise components to $\tilde{r}_{\mathrm{n}}(t)$ in (4.23), one can obtain a complete model for the channel output.

Signal-to-Noise Ratio

Signal-to-noise ratio (SNR) is the fundamental metric used for assessing the effectiveness or advantage of coding and/or signal processing approaches. The purpose of SNR is to determine the amount of noise required to achieve a specified level of error rate performance for a given signal strength. Consequently, to ensure fair comparison, it is important that the SNR definition is independent of the code rate of the channel code and recording density. In magnetic recording systems, this requirement is complicated by the fact that media noise is dependent on the written data bits.

A SNR definition that is widely used in magnetic recording [49, 69] is as follows:

$$\mathrm{SNR} = \frac{E_{\mathrm{i}}}{N_0 + M_0}, \qquad N_{\mathrm{mix}} = \frac{M_0}{N_0 + M_0}. \tag{4.24}$$

where E_{i} is the energy of the channel impulse response, $N_0/2$ is the power spectral density of electronics noise (white Gaussian), $M_0/2$ is the average energy of the media noise associated with each transition, and N_{mix} denotes the ratio of medium noise power to the total in-band noise power. Clearly, the quantities $\{E_{\mathrm{i}},\ N_0,\ M_0\}$ are independent of the code rate and linear density. For the first-order model for transition noise given in (4.11), the expression for M_0 can be obtained as

$$M_0 = 2\sigma_\tau^2 \int_{-\infty}^{\infty} h_{\mathrm{i}}^2(t)\mathrm{d}t + 2\sigma_{\mathrm{w}}^2 \int_{-\infty}^{\infty} h_{\mathrm{w}}^2(t)\mathrm{d}t, \tag{4.25}$$

where σ_τ^2 and σ_{w}^2 are the variances of write-jitter $\tau[k]$ and width-variation $\Delta W[k]$, respectively. In the presence of DC noise, the denominators in (4.24) should be replaced by $N_0 + M_0 + D_0$ where D_0 denotes the DC noise power normalized by a reference bandwidth that is independent of code rate and linear density. To make the media noise specification complete, it is also necessary to define the ratio of D_0 to $N_0 + M_0 + D_0$.

Equalization for Magnetic Recording Channels

In this section, we briefly discuss the role of equalization in magnetic recording channels and the various strategies used for equalization. Figure 4.10 shows the schematic of a recording channel followed by equalizer and detector. Data bits $a[k] \in \{-1, +1\}$ written to the medium result in a readback signal $r(t)$. The read-head output $r(t)$ is conditioned using preamplifier and the analog front-end in read channel. The conditioned signal $x(t)$ is sampled at bit-rate $1/T$ to result in $x[k] = x(kT + t_0)$ where T denotes the duration of one bit and t_0 denotes the sampling phase. The sampled output $x[k]$ is filtered using an equalizer and the equalized sequence $y[k]$ is acted upon by a detector to recover the written data bits. One could also sample $x(t)$ at a rate higher than the bit-rate and accordingly use an equalizer with fractionally spaced taps to ensure that aliasing caused by sampling

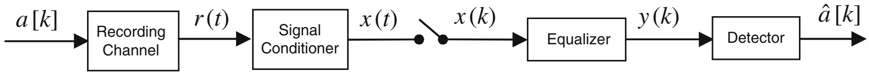


Fig. 4.10 Recording channel with equalizer and detector

does not affect equalizer performance. In this section, we assume that $x(t)$ is sufficiently band-limited so that bit-rate sampling results in no or negligible aliasing in $x[k]$, and hence the equalizer performance can be considered to be independent of the sampling phase t_0. Therefore, without loss of generality, we set $t_0 = 0$ in this section.

It is clear from the previous section that the head output $r(t)$ contains linear ISI, non-linear distortions, media noise, and electronics noise. We assume that mitigation approaches such as write precompensation and asymmetry correction have been used so that non-linear distortion can be assumed negligible in $x(t)$. The main purpose of equalizer is to shape the signal component in $x[k]$ into a form suitable for detection of the data bits by detector, while minimizing the noise as much as possible. In other words, the function of equalizer is closely tied to the technique used for detecting the data bits. In modern read channels, equalizer is implemented as a T-spaced finite impulse response (FIR) filter with $N_w + 1$ coefficients given by w_i for $i = 0, 1, 2, \ldots, N_w$. In the rest of this section, we will briefly discuss the various equalization strategies and the different techniques used for designing the equalizer.

Equalization Strategies

There are mainly three types of equalization strategies: full response (FR) equalization, partial response (PR) equalization, and decision feedback (DF) equalization. Assuming that $x[k]$ is free from non-linear distortions, we can express it as

$$x[k] = h_0 a[k] + \sum_{i \neq 0} h_i a[k-i] + \tilde{\eta}[k], \tag{4.26}$$

where h_0 and h_i for $i \neq 0$ denote the linear ISI in unequalized channel and $\tilde{\eta}[k]$ denotes the noise part. The equalizer output $y[k]$ can be expressed as

$$y[k] = g_0 a[k] + \sum_{i \neq 0} g_i a[k-i] + \eta[k],$$

where g_0 and g_i for $i \neq 0$ denote equalized linear ISI and $\eta[k]$ denotes the corresponding noise part.

The objective of FR equalization is to "fully" cancel the ISI caused by the finite bandwidth of the recording channel, i.e., to make $g_i = 0$ for $i \neq 0$ so that the equalizer output consists of only a scalar multiple of the current data bit $a[k]$ and noise. Thus, the equalized channel (i.e., the cascade of recording channel, signal conditioner and equalizer) behaves like an ideal infinite bandwidth channel with flat magnitude response. Consequently, the detector can be a simple threshold detector to discern the sign of $y[k]$ since the mean of $\eta[k]$ is 0. However, this equalization strategy is not useful for magnetic recording except at very low linear densities. This is because, in frequency domain, the FR equalizer behaves as inverse of the channel transfer function. This results in excessive noise enhancement at medium and

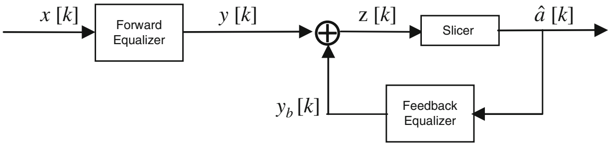


Fig. 4.11 Decision-feedback equalizer

high densities since the magnetic recording channel becomes more and more band limited as density increases [54].

The objective of DF equalization is also to fully cancel the ISI, but this is implemented using "decision-feedback" so as to achieve FR equalization without suffering from undesirable noise enhancement. Figure 4.11 shows the schematic of DF equalizer. The equalization task is divided between forward and feedback equalizers. Forward equalizer is designed to cancel non-causal ISI, i.e., $g_i = 0$ for all $i < 0$, in its output $y[k]$. Thus, forward equalizer output can be expressed as

$$y[k] = g_0 a[k] + \sum_{i>0} g_i a[k-i] + \eta[k]. \tag{4.27}$$

Since the second term on the RHS corresponds to ISI from past bits (i.e., causal ISI), the feedback equalizer is configured to compute and cancel this ISI using detector decisions, assuming correct decisions. The detector input and output are given by

$$z[k] = y[k] - y_b[k] \qquad \text{and} \qquad \hat{a}[k] = \text{sgn}\,\{z[k]\}\,, \tag{4.28}$$

where $y_b[k] = \sum_{i>0} g_i \hat{a}[k-i]$. Design of finite-length DF equalizers is addressed in [46]. The advantage of DF equalization over FR equalization arises from the fact that cancellation of non-causal ISI can be accomplished with no or minimum noise enhancement [1]. The main disadvantage of DF equalization is the possibility of error propagation, i.e., the presence of decision errors at detector output results in incorrect cancellation of causal ISI at detector input which leads to the possibility of more decision errors. Error propagation is one of the main reasons why DF equalization is not used in magnetic recording applications. Another reason for its unpopularity in magnetic recording is the limitation in slicer-detector to support sophisticated coding and detection methods. Nevertheless, there have been a few extensions to enhance its detection capability [2, 25].

PR equalization addresses the noise enhancement problem without using decision feedback. This is done by requiring the equalizer to shape the channel response into a specified response called "PR target" whose duration spans multiple bits, which is unlike FR equalization where the underlying PR target spans only one bit. Since PR equalization allows for certain amount of ISI (in the form of PR target) to remain at equalizer output, noise enhancement can be minimized by choosing the PR target to be spectrally similar to the unequalized channel response [54].

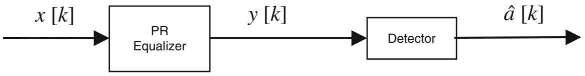


Fig. 4.12 Partial response (PR) equalizer with detector

Knowledge of the PR target is used in the detector to provide near-optimum detection performance. Viterbi detector or its advanced versions are used for data detection [30]. Let $\{g_0, g_1, \ldots, g_{N_g}\}$ be the PR target. The output of PR equalizer can be given by (see Fig. 4.12)

$$y[k] = \sum_{i=0}^{N_g} g_i a[k-i] + \eta[k], \tag{4.29}$$

where $\eta[k]$ denotes noise and residual ISI, where residual ISI is the difference between unequalized channel and PR target. Since the hardware complexity of the Viterbi-like detectors is exponential in the number of coefficients in PR target, the choice of PR target is governed by the trade-off between spectral match of PR target to unequalized channel and hardware complexity of detector.

Approaches for Equalizer Design

Since PR equalization is the approach used in all magnetic recording systems currently, we restrict our discussion on equalizer design to that for designing PR equalizer. There are mainly two approaches for designing equalizers: zero-forcing (ZF) approach and mean square (MS) approach. Whereas the ZF approach aims to minimize residual ISI, the objective of MS approach is to minimize the mean square value of the sum of residual ISI and noise. The MS approach is the most commonly used in disk drives.

The error at PR equalizer output for a given PR target can be given by

$$e[k] = y[k] - d[k] = e_{\text{isi}}[k] + e_{\text{nse}}[k], \tag{4.30}$$

where

$$y[k] = \sum_{i=0}^{N_w} w_i x[k-i] = \boldsymbol{w}^T \boldsymbol{x}[k], \qquad d[k] = \sum_{i=0}^{N_g} g_i a[k-i] = \boldsymbol{g}^T \boldsymbol{a}[k], \tag{4.31}$$

$d[k]$ denotes the desired output of PR target; $e_{\text{isi}}[k]$ and $e_{\text{nse}}[k]$ denote residual ISI and noise, respectively, in the error $e[k]$; $\boldsymbol{w} = [w_{0,}\ w_1, \cdots, w_{N_w}]^T$ denotes the equalizer coefficients, $x[k] = [\boldsymbol{x}[k],\ x[k-1], \cdots, x[k-N_w]]^T$ and $\boldsymbol{a}[k] = [a[k],\ a[k-1], \cdots, a[k-N_g]]^T$ denote the inputs of equalizer and PR target filters, respectively, at kth time instant, and superscript T denotes matrix transpose. ZF equalization amounts to, e.g., minimizing $E[e_{\text{isi}}^2[k]]$ and MS approach amounts to minimizing $E[e^2[k]]$ where $E[\cdot]$ denotes statistical expectation operator. Clearly, second-order statistics of equalizer input and data bits are sufficient to determine the optimum equalizers under ZF and MS criteria. For example, the optimum MS equalizer is given by

$$\boldsymbol{w}_{\text{opt}} = \boldsymbol{R}_{xx}^{-1}\boldsymbol{r}_{xd}, \tag{4.32}$$

where $\boldsymbol{R}_{xx} = E\left[\boldsymbol{x}[k]\boldsymbol{x}^T[k]\right]$ and $\boldsymbol{r}_{xd} = E\left[d[k]\boldsymbol{x}[k]\right]$ are the correlation matrix of equalizer input and cross-correlation vector between equalizer input and target output, respectively.

While PR equalization is usually exercised with pre-specified PR targets [18, 29, 54], the design of good PR targets is very important to guarantee near-optimum detection performance since the choice of PR target determines the noise characteristics at detector input and the type of dominant error events in detection. As a result, an important problem in equalization is joint design of equalizer and PR target. To prevent trivial solution (i.e., zero values for equalizer and target), joint optimization has to be done under some constraints imposed on equalizer and/or target [36, 50]. We shall illustrate this using the so-called monic constraint on target (i.e., $g_0 = 1$). For example, the optimum equalizer and monic-constrained target under MS criterion are given by

$$\boldsymbol{w}_{\text{opt}} = \boldsymbol{R}_{xx}^{-1}\boldsymbol{R}_{xa}\,\boldsymbol{g}_{\text{opt}} \quad \text{and} \quad \boldsymbol{g}_{\text{opt}} = \frac{1}{\boldsymbol{u}^T\boldsymbol{R}_o^{-1}\boldsymbol{u}}\boldsymbol{R}_o^{-1}\boldsymbol{u}, \tag{4.33}$$

where $\boldsymbol{R}_o = \boldsymbol{R}_a - \boldsymbol{R}_{xa}^T\boldsymbol{R}_x^{-1}\boldsymbol{R}_{xa}$ and $\boldsymbol{u} = [1, 0, \cdots, 0]^T$.

In disk drives, design of equalizer and target using closed-form expressions of the type, (4.33) are not a recommended option in view of computational complexity as well as the necessity to keep track of the changing characteristics of the recording channel across heads and media. Therefore, it is very common to use adaptive approaches [55] for designing equalizer. The adaptive version of MS approach is obtained by implementing the well-known gradient descent approach for minimizing the squared error $e^2[k]$. The resulting algorithm is traditionally known as LMS (least mean square) algorithm. As an example, the LMS adaptive algorithm for designing equalizer and monic-constrained target can be given by [14]

$$\left.\begin{aligned} e[k] &= \boldsymbol{w}^T[k-1]\boldsymbol{x}[k] - \boldsymbol{g}^T[k-1]\boldsymbol{a}[k] \\ \boldsymbol{w}[k] &= \boldsymbol{w}[k-1] - \mu_w e[k]\boldsymbol{x}[k] \\ \boldsymbol{g}[k] &= \boldsymbol{g}[k-1] + \mu_g e[k]\boldsymbol{a}[k] \\ g_0[k] &= 1 \end{aligned}\right\}, \tag{4.34}$$

where $\boldsymbol{w}[k]$ and $\boldsymbol{g}[k]$ denote the coefficient vectors of equalizer and target, respectively, at kth time instant, and μ_w and μ_g are positive scalars that control the adaptation rate.

It should be kept in mind that the ZF and MS approaches for equalizer design do not guarantee that detection performance will be optimized under all conditions.

For example, equalizer and target that result in minimum mean square error do not necessarily result in minimum bit error rate unless the noise at Viterbi detector input is white Gaussian and residual ISI is 0. Nevertheless, practical systems resort to MS and ZF approaches since the approaches that minimize error rate are usually quite expensive to implement while the solutions of MS and ZF approaches are usually near the optimum. We would like to close this section by pointing out certain references that present error rate minimizing approaches for equalizer design. Reference [38] presents an analytical approach for designing optimum PR target and equalizer by maximizing the detection SNR of Viterbi detector – which is equivalent to minimizing bit error rate. Reference [56] presents an adaptive algorithm that minimizes bit error rate for designing equalizer.

Signal Detection and Decoding for Magnetic Recording Channel

As pointed out earlier, ECCs have played an important role in achieving reliable data transmission. A good overview of various error correction codes can be found in [40]. In hard disk drive, the Reed–Solomon (RS) coding has been used for decades. Recently, the industry is replacing RS codes with low-density parity-check (LDPC) codes and iterative detection/decoding which can achieve more coding gain. ECC provides error correction capability at the cost of adding redundancy into information bits. This leads to additional reduction of magnetic recording density. Given a constant rotation speed, the sampling rate 1/T is accordingly increased due to more bits are compacted in the unit length of track. As a result, the energy decreases in the pulse response because the isolated pulses p(t) and −p(t) get closer and cancel each other more. Moreover, the increase in sampling rate expands the signal bandwidth which introduces more noise energy. So, the overall SNR is decreased. By adopting the LDPC codes into read channel, the error correction capability can be dramatically improved in terms of sector failure rate, as shown in Fig. 4.13.

In this section, we mainly focussed on soft in soft out (SISO) channel detection, soft LDPC decoding, and iterative detection and decoding.

Channel Detector

The read channel is typically equalized as target which can be described in terms of the partial response polynomial:

$$H(D) = h_0 + h_1 \cdot D + h_2 \cdot D^2 + \cdot + h_v \cdot D^v,$$

where D indicates one sample delay. The target coefficients h_i are often integers to simplify the implementation of the read channel.

Given the channel response polynomial, the channel can be treated as rate a trellis code with 2^v states. So, a trellis decoder can be employed to detect the

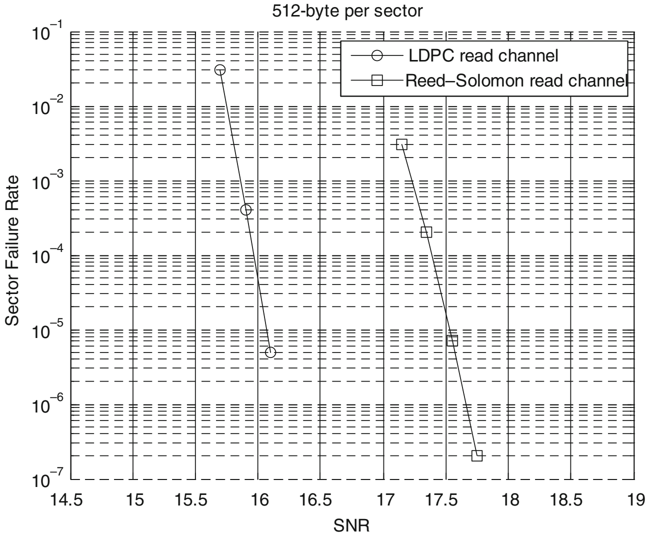


Fig. 4.13 Error correction comparison between RS and LDPC

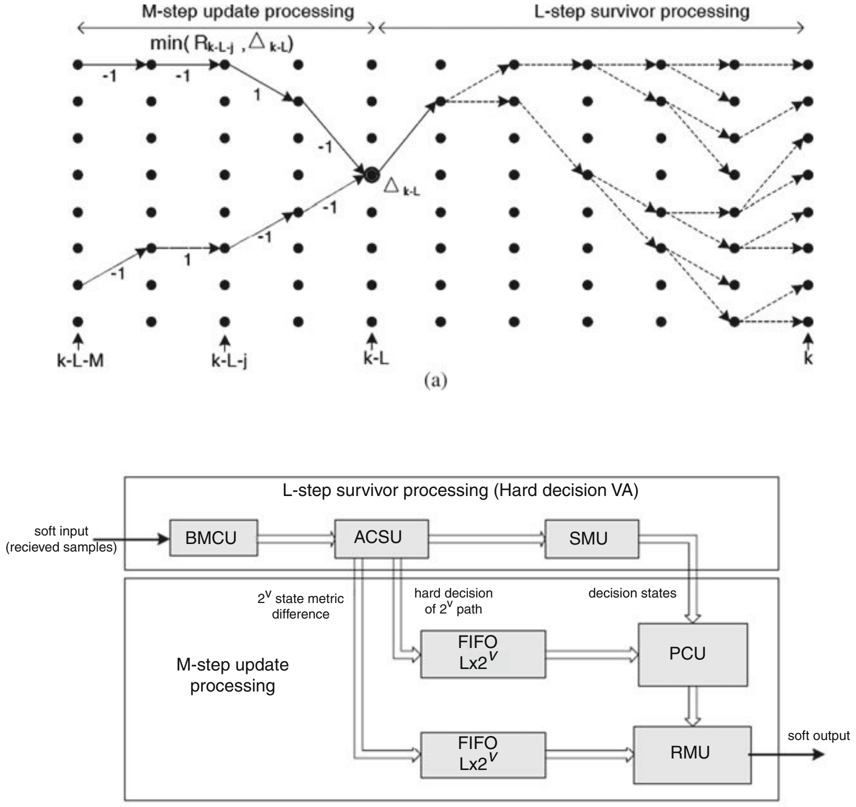
recorded sequence. Hard decision Viterbi detector has been used since early 1990s. The detected bit sequence is passed to a Reed–Solomon decoder for further error correction. However, to exploit the powerful error correction capability of LDPC decoder, soft output information as a measure of a posteriori probability (APP) is required from the channel detector. As a result, detection algorithms which provide soft output are required.

There are two types of soft output detectors. The first one is maximum a posteriori (MAP) detector that implements the Bahl–Cocke–Jelinek–Raviv (BCJR) algorithm [7], which is often referred as Max-Log-MAP algorithm when it is approximated and implemented in the log domain. The other one is maximum likelihood sequence detector that employ soft output Viterbi algorithm (SOVA) [28, 70]. The former one provides better soft output quality at the cost of higher computation complexity. Moreover, we note that the SOVA can be modified [15] so that it can perform equivalently well as Max-Log-MAP algorithm.

SOVA Detector

Conventional Viterbi algorithm takes soft quantized information as input and outputs hard decision sequence. It has been recognized that performance can be improved if the reliability of each decision bit can be obtained for further use by outer code. As a matter of fact, soft output Viterbi algorithm has been known for long time, though it was not used in real applications due to the high computational complexity. With the progress on reduction of algorithm complexity and advance of VLSI technology, soft output detectors/decoders are now playing an important role in channel coding, equalization, and signal detection. In this section, the VLSI architecture of high-speed SOVA is presented.

The process of SOVA as shown in Fig. 4.14a can be divided into two stages: survivor processing and update processing. The survivor processing is similar to conventional hard decision VA. First, the branch metric of each possible state transition is computed by branch metric calculation unit (BMCU). Then, the branch metrics are added to state metrics, which denote of weighting of the state transition sequence. This is performed by Add-Compare-Select unit (ACSU), which also compares the accumulated state metrics of the paths entering into same state and selects the one with minimum metric value. At last, the most likely output sequence is found out through the survivor memory unit (SMU). The update processing aims to evaluate the difference between the survivor path and competing path and update the reliability of the decision. The diagram of SOVA architecture is shown in Fig. 4.14b. The BMCU, ACSU, and SMU form a VA detector, which is employed to determine the survivor states and the output at depth L. The hard decision and metric difference of each state are computed by ACSU and stored in FIFO as a delay line. The path comparison unit (PCU) tracks the hard decision through the final survivor (decision)



(a)

(b)

Fig. 4.14 SOVA detector

path and the competing path. If the hard decision is different, the reliability measure unit (RMU) updates the reliability value R_{K-L-j} by min($\mathrm{R}_{\mathrm{K-L-j}}$, $\Delta_{\mathrm{K-L}}$), where $\Delta_{\mathrm{K-L}}$ is the metric difference between the decision path and competing path.

Log-Max-MAP Detector

The Max-Log-Map detector is usually implemented via sliding window approach [7]. To further improve the throughput, we can use multiple sliding windows sub-detectors that operate in parallel, leading to a parallel sliding window detector.

The number of sub-detectors depends on the desired trade-off between silicon area and throughput. Each sub-detector has a structure as shown in Fig. 4.15. Each sub-detector consists of one branch metric unit (BMU), one forward recursion unit (FRU), two backward recursion units (BRU), one log-likelihood-ratio (LLR) calculation unit, one memory bank Gamma Memory consisting of 4 single-port memory blocks to store the branch metrics computed by BMU, and one memory bank Alpha Memory consisting of 2 single-port memory blocks to store the forward state metrics computed by FRU. The operation can be briefly described as follows:

1. Branch Metrics Calculation: Upon receiving the symbol y_k with additive noise, BMU calculates the branch metrics of the transition from state *s'* to state *s* as:

$$\gamma_k(s', s) = L(y_k|u_k) + L(u_k),$$

   where u_k is the transmitted symbol, y_k is received sample, and L is log-likelihood ratio.
2. Forward Recursion: FRU computes the forward state metrics α_k as

$$\alpha_k(s) = \max_{s'}(\alpha_{k-1}(s') + \gamma_k(s', s)).$$

3. Backward Recursion and Soft output Calculation: With the available branch metrics and forward state metrics, the BRUs compute backward state metrics β_{k-1} as

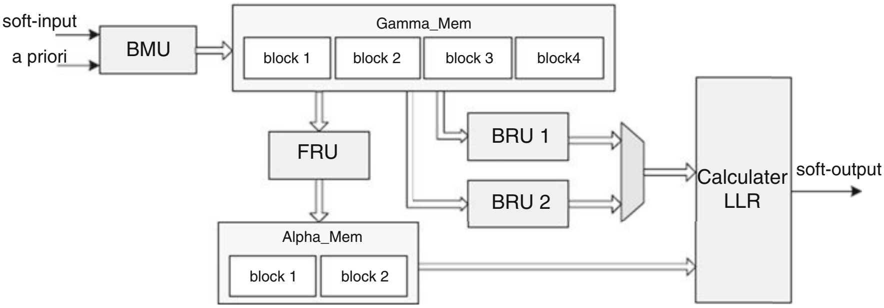


Fig. 4.15 Sliding window max-log-map detector

$$\beta_{k-1}(s') = \max_{s}(\beta_k(s) + \gamma_k(s', s)).$$

When all the needed α, β, and γ are ready, the LLR calculation unit works out the a posteriori LLR of the information bit u_k as

$$L(u_k|y) = \max_{\substack{(s',\, s)\\ u_k=+1}} (\alpha_{k-1}(s') + \gamma_k(s',\, s) + \beta_k(s)) - \max_{\substack{(s', s)\\ u_k=-1}} (\alpha_{k-1}(s') + \gamma_k(s', s) + \beta_k(s)).$$

The operation sequences of each component are illustrated in Fig. 4.16. For the purpose of simplicity, the operation of LLR calculation unit is merged into BRUs and the operation of BRUs is divided into two stages: BRU_{1a} and BRU_{2a} do not generate LLR output, and BRU_{1b} and BRU_{2b} generate LLR output. In Fig. 4.16, the vertical axis represents time, and the horizontal axis represents the memory access address. From $t = 0$, in every L clock cycles, BMU calculates L of γ values to store them in the corresponding block of Gamma Memory (each block has L words). From $t = 2L$, FRU performs forward recursions to calculate L of α values in every L clock cycles and store them in Alpha Memory. At the same time BRU_1 begins backward recursions to accumulate backward state metrics for path mergence. From $t = 3L$, BRU_1 continues backward recursion and soft output is calculated every clock cycle, while BRU_2 starts backward recursion to accumulate βs in parallel. Then BRU_1 and BRU_2 alternate every L clock cycles to generate for soft-out calculation and accumulate for path mergence. The detection latency is $3L$ clock cycles.

LDPC Codes and Decoding

Invented by Gallager [16] in 1962, LDPC codes were largely neglected for several decades due to their high computation complexity. Inspired by remarkable

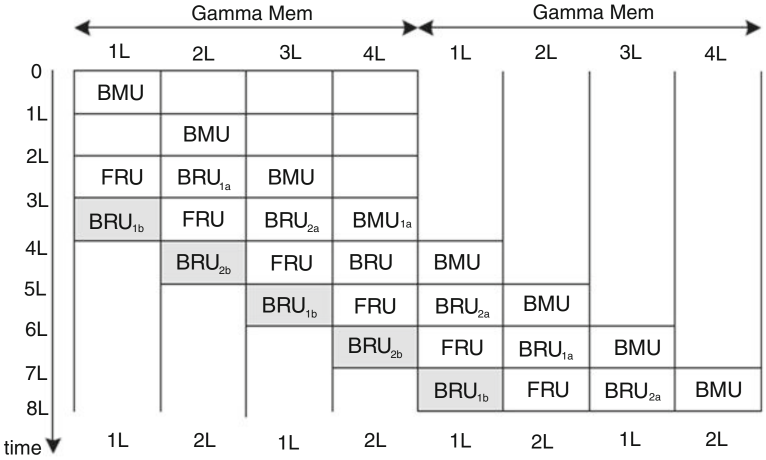


Fig. 4.16 Scheduling for sliding window max-log-map detector

success of Turbo codes in early 1990s, MacKay and Neal [44] and Wiberg [64] re-discovered LDPC codes in 1996. Since then, LDPC codes have been attracting tremendous research interest because of their excellent error-correcting performance and highly parallel decoding scheme. Today, many industry standards begin to adopt LDPC codes, such as digital video broadcasting (DVB-S2) for satellite video, IEEE 802.3 for 10G Ethernet, IEEE 802.16 for Wireless Metropolitan Area Network (WiMAX), IEEE 802.11 for Wireless Local Area Network (WiFi), IEEE 802.12 for Wireless Personal Area Network (WPAN), and IEEE 802.20 for mobile Broadband Wireless Access Networks (MBWA).

An LDPC code is defined as the null space of a parity check matrix H. The name comes from the characteristic of H in which the number of 1 s is much less than the number of 0s. There are two common representations of LDPC codes. Like all linear block codes they can be described via matrices. The other one is a graphical representation in which each row of H is represented by a check node; each column is represented by a variable node; and each "1" element is represented by an edge connecting between the corresponding check and variable nodes. For example, an M × N sparse parity check matrix H can be represented by a bipartite graph consisting of M check nodes in one set, N variable nodes in the other set and L edges connecting between the two sets of nodes, where L is the total number of 1s in H. The bipartite graph representation greatly facilitates the development and illustration of the decoding algorithm. If the number of 1s in both the rows and the columns of the parity check matrix is same, i.e., all the variable nodes have the same degree and all the check node also have the same degree, the LDPC codes are called regular LDPC codes. On the contrary, LDPC codes which do not have a constant number of non-zero entries in the rows or in the columns of parity check matrix are called irregular LDPC codes. Following the notation in [42], they are specified by the weight distribution of rows $\rho(x)$ and the weight distribution of column $\lambda(x)$:

$$\rho(x) = \sum_{i=2}^{d_c} \rho_i x^{i-1},$$

$$\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1},$$

where ρ_i and λ_i denote the proportion of rows and columns with weight i; while d_c and d_v are the maximum row weight and column weight, respectively. Given the parity check matrix $H_{\mathrm{M}\times\mathrm{N}}$, the rate R of LDPC codes is defined by $R > R_{\mathrm{d}} = (N - M)/N$. $R_{\mathrm{d}} = R$ if the parity check matrix has full rank.

Decoding Algorithms

Generally, the LDPC decoding algorithms can be categorized into two classes: hard decision decoding and soft decision decoding. The latter schemes provide superior performance at the cost of higher complexity. The commonly known hard decision algorithms include majority-logic (MLG) decoding, bit-flipping decoding, and weighted BF or MLG decoding, while the sum-product algorithm (SPA) and min-sum algorithm are the two most popular soft decision decoding schemes. Since the hard decision decoding is relatively simple and the decoding performance is inferior to soft decision, we focus only on the soft decision decoding. The readers are referred to [34, 43] for various hard decision decoding algorithms.

The soft decision decoding of LDPC codes is also called iterative message passing or belief-passing (BP) decoding [35, 43] that directly matches the code bipartite graph as illustrated in Fig. 4.17: After each variable node is initialized with the input channel message, the decoding messages are iteratively computed by all the variable nodes and check nodes and exchanged through the edges between the neighboring nodes. When the decoding is performed in log domain, it is called iterative Log-BP decoding algorithm.

It is well known that various soft decision LDPC decoding algorithms may fall into two categories, including Sum-Product algorithm (SPA) and Min-Sum algorithm. Depending on the decoder scheduling, they can be further categorized as non-layered and layered decoding algorithm [20, 23, 45]. Min-Sum algorithm is commonly used because its check node processing approximation may potentially lead to significant silicon area savings from two perspectives: (i) the logic complexity may be reduced due to the elimination of the function log[tanh(/2)] that is typically implemented as lookup-table (LUT) in hardware and (ii) more importantly, the size of memory for decoding message storage may be reduced due to the possible compact representation of check-to-variable messages. However, this memory saving potential has not been fully exploited by existing high-speed partially parallel Min-Sum decoders, although such potential has been pointed out in some serial Min-Sum decoding schemes [19, 67]. Moreover, for partially parallel decoder design, the conventional Min-Sum algorithm formulation results in explicit implementation of a sorter in each check node processing unit, which will make the potential of logic silicon area saving quickly diminish and result in an essential speed bottleneck as the code rate (or, more specifically, the row weight of parity check matrix) increases.

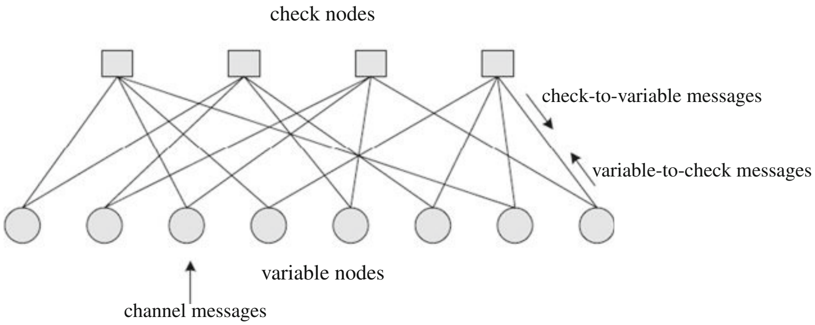


Fig. 4.17 Illustration of LDPC code bipartite graph

Before presenting the decoding algorithm, some definitions are introduced as follows: Let H denotes the $M \times N$ parity check matrix and $H_{\mathrm{m,n}}$ denote the entry of H at the position (m, n). The set of bits n that participate in parity check m is defined as $N(m) = \{n : H_{\mathrm{m,n}} = 1\}$, and the set of parity check m in which bit n participates is defined as $M(n) = \{m : H_{m,\mathrm{n}} = 1\}$. The set $\mathrm{N}(m)$ with bit n excluded is denoted as $N(m)\backslash n$, and the set $M(n)$ with check m excluded as $M(n)\backslash m$. The channel message, variable-to-check message, check-to-variable message, and posterior log-likelihood ration (LLR) are denoted as γ_n, $\alpha^i_{m,n}$, $\beta^i_{m,n}$, and λ^i_n, respectively, where the superscript i is iteration index.

The main difference between Sum-Product algorithm and Min-Sum algorithm lies in the check node processing, i.e., the check node processing in SPA is realized as

$$\beta_{m,n} = \Phi\left(\sum_{n'\in \mathrm{N}(m)\backslash n} \Phi(|\alpha_{m,n'}|)\right) \prod_{n'\in \mathrm{N}(m)\backslash n} \mathrm{sign}(\alpha_{m,n'}),$$

where$\Phi(x) \equiv -\log[\tanh(x/2)]$. The check node process in Min-Sum algorithm is approximated as

$$\beta_{m,n} = \min_{n'\in \mathrm{N}(m)\backslash n} (\alpha_{m,n'}) \prod_{n'\in \mathrm{N}(m)\backslash n} \mathrm{sign}(\alpha_{m,n'}).$$

Therefore, the function $\Phi(x)$, which is typically implemented as look-up table (LUT) in hardware, is eliminated in Min-Sum algorithm. The conventional Min-Sum algorithm formulation is described as follows:

```
Intialization: $\alpha^0_{m,n} = \gamma_n$;
for $i$= 0 to $i_{max}$ or convergence to codeword do
    forall check nodes $c_m$, $m \in$ {1,. . .,M} do
        $mag\left(\beta^i_{m,n}\right) = \min_{n'\in N(m)\backslash n}\left\{\left|\alpha^{i-1}_{m,n'}\right|\right\}$;
        $sign\left(\beta^i_{m,n}\right) = \prod_{n'\in N(m)\backslash n} sign\left(\alpha^{i-1}_{m,n'}\right)$;
    end
    forall variable nodes $v_n$, $n \in \{1, \ldots, N\}$ do
        $\lambda^i_n = \gamma_n + \sum_{m\in M(n)} \beta^i_{m,n}$;
        $\alpha^i_{m,n} = \lambda^i_n - \beta^i_{m,n}$;
    end
end
Output the decoded bits as sign $\left(\lambda^i_n\right)$
```

Due to the check node processing approximation, the check-to-variable messages from each check node only have two different magnitudes (i.e., the minimum and the second minimum ones among the magnitudes of all the variable-to-check messages entering into this check node), no matter how large the check node degree is. Meanwhile, the check node processing approximation eliminates the function$\Phi(x)$. Intuitively, these two features may be leveraged to reduce the storage and logic silicon area. However, a direct realization of partially parallel decoders based on the above conventional Min-Sum algorithm formulation may not be able to effectively materialize such silicon area saving potential for following two main reasons:

1. In spite of much less check-to-variable messages storage requirement, the total number of distinct variable-to-check messages always equals to the total number of 1s in the parity check matrix. A direct realization of partially parallel decoders may have to provide explicit storage for these variable-to-check messages, leading to the same (or similar) storage requirement as in its SPA decoder counterpart.
2. The direct realization of partially parallel decoders tends to implement parallel-input parallel-output check node processing units that use a sorter to search the two minimum ones among all the incoming variable-to-check messages. As code rate increases, the silicon area overhead incurred by sorters will quickly increase.

To solve the above two issues, a transformed Min-Sum algorithm is described in the following. Although it is mathematically equivalent to the original Min-Sum algorithm, its formulation and execution order make it straightforward to realize silicon area savings at VLSI architecture level. In particular, this algorithm transformation has two key features, including (1) the check node processing and variable node processing are interleaved in such a way that each newly updated variable-to-check message may be directly absorbed by check node processing units without being intermediately stored and (2) the check node processing is sequentialized so that the explicit implementation of a sorter is eliminated. To generate the outgoing check-to-variable messages from each check node (i.e., $\{|\beta_{m,n}|, n \in \mathrm{N}(m)\}$), the sequentialized check node processing n|, n 2 N(m)}), only needs to keep track of the two minimum magnitudes, i.e., $\min1_m$ and $\min2_m$ where $\min1_m \leq \min2_m$, among the input variable-to-check messages, the sign $s_{m,n}$ of each input variable-to-check message and $S_m = \prod s_{m,n}$, and the variable node index I_m representing which variable node provides the message with the minimum magnitude.

```
for $i$= 0 to $i_{max}$ or convergence to codeword do
    forall variable nodes $v_n$, $n \in \{1,\ldots,N\}$ do
        if $i$= 0 then
            $\alpha^i_{m,n} = \gamma_n$;
        else
            $\beta^i_{m,n} = \begin{cases} S^i_m \cdot s^i_{m,n} \cdot min1^i_m & n \neq I^i_m \\ S^i_m \cdot s^i_{m,n} \cdot min2^i_m & \text{otherwise} \end{cases}$
            $\lambda^i_n = \gamma_n + \sum_{m \in M(n)} \beta^i_{m,n}$;
            $\alpha^i_{m,n} = \lambda^i_n - \beta^i_{m,n}$;
        end
```

```
            Initialize $min1_m^{i+1} = min2_m^{i+1} = +\infty, S_m^{i+1} = 1$;
            forall check nodes $c_m$, $m \in M(n)$ do
                if $\left|\alpha_{m,n}^i\right| < min1_m^{i+1}$ then
                    $min1_m^{i+1} = \left|\alpha_{m,n}^i\right|$;
                    $I_m^{i+1} = n$;
                else if $\left|\alpha_{m,n}^i\right| < min2_m^{i+1}$ then
                    $min2_m^{i+1} = \left|\alpha_{m,n}^i\right|$;
                $s_{m,n}^{i+1} = sign\left(\alpha_{m,n}^i\right)$;
                $S_m^{i+1} = S_m^{i+1} \cdot s_{m,n}^{i+1}$;
            end
        end
    end
    Output the decoded bits as sign $\left(\lambda_n^i\right)$
```

Additionally, we note that the Min-Sum algorithm is generally less sensitive to quantization errors compared to SPA and hence may enable the use of smaller finite word-length. For example, it has been shown in [10] that a 4-bit quantization of decoding messages may be sufficient to avoid error floor, although it may result in about 0.2 dB performance loss compared to 6-bit quantization.

QC-LDPC Code and Decoder Architecture

For efficient hardware realization, the LDPC code is typically constructed in quasi-cyclic (QC) form. The parity check matrix of a QC-LDPC code consists of an $m_b \times n_b$ array of $p \times p$ square circulant matrices. So the $m_b{\cdot}p\times\, n_b\, p$ parity check matrix can be represented by a bipartite graph as shown in Fig. 4.18, where each group of consecutive p rows (or columns) is represented by a set of p check (or variable) nodes. Notice that the cyclic permutation blocks realize message passing between adjacent variable and check node groups through simple cyclic permutations based on the quasi-cyclic parity check matrix structure.

In each decoding iteration, the decoding scheduling directly follows the above transformed formulation. This can be illustrated in Fig. 4.19 and explained as follows. One out of the total n_b variable node sets is processed at one time and all the check nodes are processed in a serial manner interleaved with the variable node processing. In Fig. 4.19a the first set of variable nodes is processed and feeds variable-to-check messages to all the check nodes for partial check node processing. Then the decoding moves to the second step, as shown in Fig. 4.19b, where the second set of variable nodes is processed and feeds variable-to-check messages to all the check nodes for further partial check node processing. Once all the variable nodes are processed within present iteration, as shown in Fig. 4.19c, all the check nodes also receive all the input variable-to-check messages and finish the check node processing for present iteration. Then all the check-to-variable messages will be fed to the variable nodes for the next iteration. Figure 4.19 clearly illustrates the two desirable features of this proposed Min-Sum algorithm transformation, i.e., the

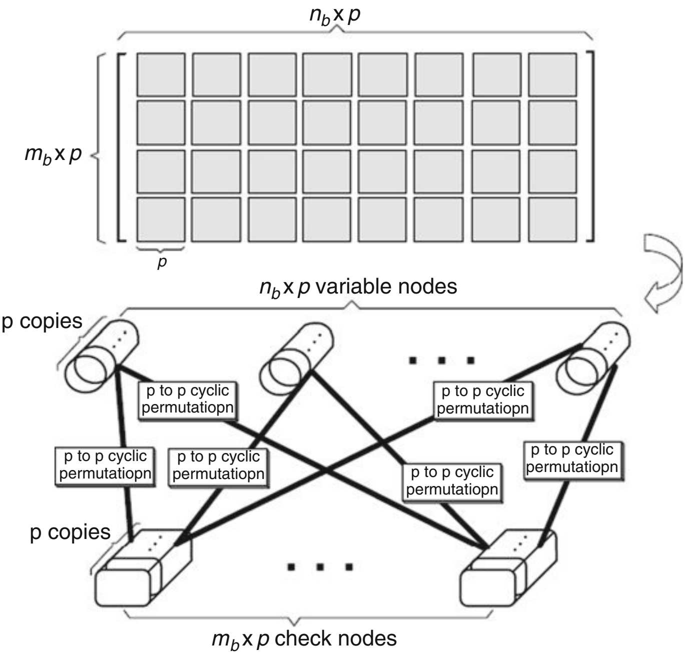

n_b x p
m_b x p
p
n_b x p variable nodes
p copies
p to p cyclic permutatiopn
p to p cyclic permutatiopn
p to p cyclic permutatiopn
p to p cyclic permutatiopn
p to p cyclic permutatiopn
p to p cyclic permutatiopn
p copies
m_b x p check nodes

Fig. 4.18 Quasi-cyclic LDPC and graph mapping

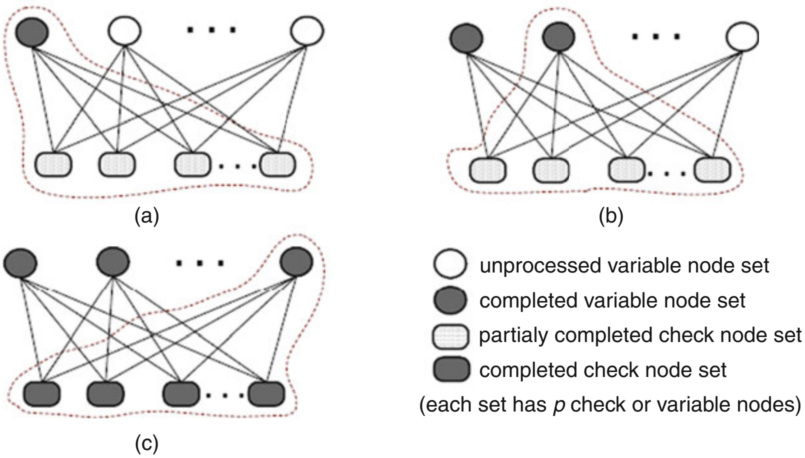

(a)
(b)
(c)
unprocessed variable node set
completed variable node set
partialy completed check node set
completed check node set
(each set has p check or variable nodes)

Fig. 4.19 Illustration of decoding scheduling

obviation of explicit storage of variable-to-check decoding messages and concurrent operation of check node processing and variable node processing.

Figure 4.20 shows the corresponding partially parallel QC-LDPC decoder architecture. It contains p variable node processing units (VNUs) that process one set of

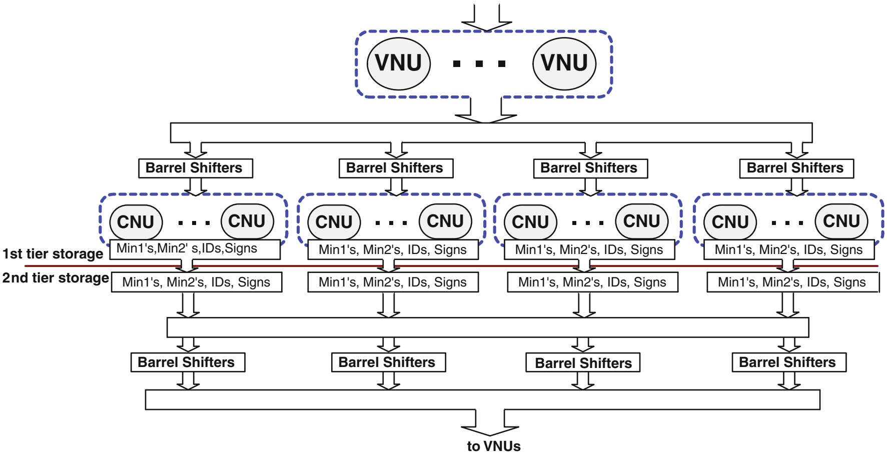


Fig. 4.20 Corresponding partially parallel QC-LDPC decoder architecture

p variable nodes in each clock cycle, and $m \cdot p$ check node processing units (CNUs) that process all the $m \cdot p$ check nodes in n clock cycles. The variable-to-check messages computed in each clock cycle are directly routed to the $m \cdot p$ CNUs via the barrel shifters which are configured according to the structures of circulant matrices. Each CNU serially updates min1_m, min2_m, $s_{m,n}$, S_m, and I_m that are kept in the first set storage. After n clock cycles, all the check-to-variable messages are computed and then passed to the second set storage that provide input to all the VNUs. Note that the storage for the signs in the two tiers can be shared using first in first out buffers (FIFOs). The reverse routing from CNUs to VNUs is also realized by barrel shifters. The data path is naturally pipelined so that all the VNUs and CNUs can work concurrently. All the variable-to-check messages are immediately absorbed by CNUs and hence do not require storage, while only two minimum magnitudes, one location index, and signs need to be stored to generate the check-to-variable messages from each check node.

For layered decoding algorithm, only p check node processing units need to be implemented; however it needs m times of cycles to complete all layers for a full iteration. Typically, the layered decoding converges faster, so the number of iterations can be reduced.

Iterative Detection/Decoding

Following the Turbo principle [21, 33], an iterative soft detection and LDPC decoding system has the block diagram as shown in Fig. 4.21. The soft-in-soft-out (SISO) channel detector takes received sample and generates reliability or log-likelihood ratio (LLR) of binary decision. Taking max-log-map detector as an example, it maximizes the APP $p(u_k \mid y)$ given the observed sequence y, i.e.,

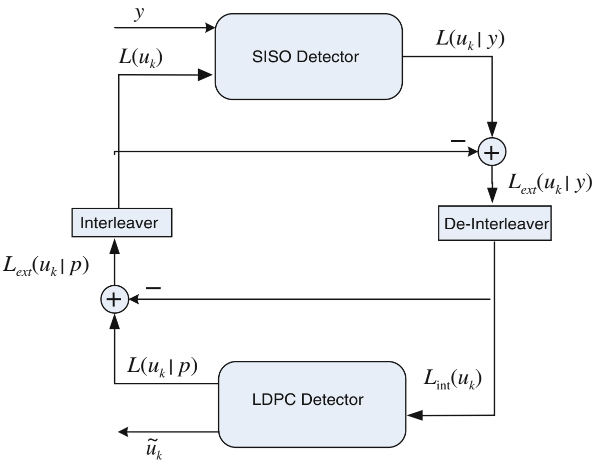


Fig. 4.21 Iterative detection and decoding

$$\hat{u}_k = \underset{a\in\{0,1\}}{\arg\max}\, P(u_k = a \mid y),$$

and calculate the log-likelihood ratio, i.e.,

$$L(u_k \mid y) = \ln \frac{p(u_k = 0 \mid y)}{p(u_k = 1 \mid y)}.$$

The probability $P(u_k = a \mid y)$, where $a \in \{0, 1\}$, can be written as

$$P(u_k = a \mid y) = \sum_{\forall u: u_k = a} p(u \mid y) = \sum_{\forall u: u_k = a} \frac{p(y \mid u)p(u)}{y}.$$

The probability $P(\boldsymbol{u})$ is the a priori probability of sequence $\mathbf{u}$, which can be used to derive knowledge about the source producing the bit u_k. Assuming the bits in $\boldsymbol{u}$ are independent, i.e., the probability $P(\boldsymbol{u})$ factors as $\prod_{k=1}^{K} p(u_k)$, the LLR can be reformulated as follows:

$$\begin{aligned} L(u_k \mid y) &= \ln \frac{\sum_{\forall u: u_k = 0} p(y \mid u) \prod_{i=1}^{K} p(u_k)}{\sum_{\forall u: u_k = 1} p(y \mid u) \prod_{i=1}^{K} p(u_k)} \\ &= \ln \frac{\sum_{\forall u: u_k = 0} p(y \mid u) \prod_{i=1:i\neq k}^{K} p(u_k)}{\sum_{\forall u: u_k = 1} p(y \mid u) \prod_{i=1:i\neq k}^{K} p(u_k)} + L(u_k) \\ &= L_{\text{ext}}(u_k \mid y) + L(u_k) \end{aligned}.$$

The term $L_{\text{ext}}(u_k \mid y)$is the extrinsic information about u_k contained in $\boldsymbol{y}$, while the $L(u_k)$ is the a priori information about u_k. The extrinsic information is key point in iterative detection and decoding, which is also called Turbo equalization.

The extrinsic information $L_{\text{ext}}(u_k \mid y)$ is passed through a de-interleaver and fed to LDPC decoder as intrinsic message $L_{\text{int}}(u_k)$. The LDPC decoder performs message passing decoding and generates soft output $L(u_k \mid p)$ and hard decision $\tilde{u}_k$. By subtracting intrinsic message from the soft output, the extrinsic message $L_{\text{ext}}(u_k \mid p)$ is obtained. The extrinsic information means that the soft output is not the current best estimate of the reliability of the decision, but is instead the new information of current best estimate excluding the intrinsic soft-input. The extrinsic messages $L_{\text{ext}}(u_k \mid p)$ go through interleaver and pass to the detector as prior information $L(u_k)$ for next iteration.

Typically the iteration between the detector and decoder is called global iteration, while the iteration in LDPC decoding is called local iteration. The number of local and global iterations is determined by the trade-off between error correction performance, hardware implementation complexity and throughput. Usually, more iteration can help error correction performance at cost of hardware resource.

Conclusions

This chapter briefly discusses the basics of modern hard disk drives and overall system organization. As the hard disk drive areal storage density is being pushing toward the 1 Tb/in^2 with the introduction of new technologies such as perpendicular recording and patterned media, powerful read channel designs are becoming increasingly crucial. Hence, this chapter further elaborates on the magnetic recording read channels including modeling and signal processing and coding. In spite of recent significant progress of solid-state drives based on NAND flash memory, areal density scaling and cost reduction of hard disk drives will surely continue with a rate at least same as solid-state drives and will remain the mainstream mass data storage technology for the foreseeable future.

References

1. Belfiore C, Park J (1979) Decision feedback equalization. Proc IEEE 67:1143–1156
2. Bergmans J, Voorman J, Wong-Lam H (1997) Dual decision feedback equalizer. IEEE Trans Commun 45:514–518
3. Bergmans J, Voorman J, Wong-Lam H (1999) Structure and adjustment of a novel write precompensation scheme. IEEE Trans Magn 35:2053–2059
4. Berrou C, Glavieux A, Thitimajshima P (1993) Near shannon limit error-correcting coding and decoding: Turbo-codes. In: Proceedings of ICC'93, Geneva, Switzerland, pp 1064–1070
5. Bertram H, Takeo A, Jin Z, Fu C (2006) Transition shape analysis from medium noise measurements. IEEE Trans Magn 42:1620–1625
6. Blahut RE (1984) Theory and practice of error control codes. Addison Wesley, New York, NY
7. Boutillon E, Gross WJ, Gulak PG (2003) VLSI architectures for the MAP algorithm. IEEE Trans Commun 51:175–185

8. Bruno J, Brustoloni J, Gabber E, Ozden B, Silberschatz A (1999) Disk scheduling with quality of service guarantees. In: Proceedings of IEEE International Conference on Multimedia Computing and Systems, Florence, Italy, pp 400–405
9. Cain W, Payne A, Baldwinson M, Hempstead R (1996) Challenges in the practical implementation of perpendicular magnetic recording. IEEE Trans Magn 32:97–102
10. Chen J, Dholakia A, Eleftheriou E, Fossorier M, Hu XY (2005) Reduced-complexity decoding of LDPC codes. IEEE Trans Commun 53:1288–1299
11. Daniel E, Mee C, Clark M (1999) Magnetic recording: the first 100 years. IEEE Press
12. Elerath J (2007) Hard disk drives: the good, the bad and the ugly! ACM Queue 5(6):28–37
13. Erden M, Kurtas E (2004) Thermal asperity detection and cancellation in perpendicular magnetic recording systems. IEEE Trans Magn 40:1732–1737
14. Farhang-Boroujeny B (1999) Adaptive filters: theory and applications, chap. 6. Wiley, New York, NY
15. Fossorier M, Burkert F, Lin S, Hagenauer J (1998) On the equivalence between SOVA and Max-Log-MAP decodings. IEEE Commun Lett 2(5):137–139
16. Gallager RG (1962) Low-density parity-check codes. IRE Trans Inf Theory IT-8:21–28
17. Geist R, Daniel S (1987) A continuum of disk scheduling algorithms. ACM Trans Comput Syst 5:77–92
18. Gopalaswamy S, McEwen P (2001) Read channel issues in perpendicular magnetic recording. IEEE Trans Magn 37:1929–1931
19. Guilloud F, Boutillon E, Danger J (2003) l-min decoding algorithm of regular and irregular codes. In: Proceedings of the 3nd international symposium on turbo codes and related topics, Brest, France, pp 451–454
20. Gunnam K, Choi G, Yeary MB: A parallel VLSI architecture for layered decoding for array LDPC codes. In: Proceedings of International Conference on VLSI Design, Andhra Pradesh, India, pp 738–743
21. Hagenauer J (1997) The Turbo principle: tutorial introduction and state of the art. In: Proceedings of the international symposium on turbo codes, Brest, France, pp 1–11
22. Herrman R (1990) Volterra modeling of digital magnetic saturation recording. IEEE Trans Magn 26:2125–2127
23. Hocevar D (2004) A reduced complexity decoder architecture via layered decoding of LDPC codes IEEE workshop on signal processing systems (SIPS), Texas, USA, pp 107–112
24. Immink KS, Siegel P, Wolf J (1998) Codes for digital recorders. IEEE Trans Inf Theory 44:2260–2299
25. Indukumar K, Lee Y, Mathew G (1999) Performance comparison of modified multilevel decision feedback equalization detectors. IEEE Trans Magn 35:594–604
26. Information Technology Standards. I.C.: http://www.incits.org/ (2010)
27. Iwasaki S, Nakamura Y (1977) An analysis for the magnetization mode for high density magnetic recording. IEEE Trans Magn 13:1272–1277
28. Joeressen O, Meyr H (1995) A 40 Mb/s soft-output Viterbi decoder. IEEE J Solid State Circuits 30:812–818
29. Kaitsu I, Inamura R, Toda J, Morita T (2006) Ultra high density perpendicular magnetic recording technologies. Fujitsu Sci Tech J 42:122–130
30. Kavcic A, Moura J (2000) The Viterbi algorithm and Markov noise memory. IEEE Trans Inf Theory 46:291–301
31. Kavcic A, Patapoutian A (1999) A signal-dependent autoregressive channel model. IEEE Trans Magn 35:2316–2318
32. Kikitsu A, Kamata Y, Sakurai M, Naito K (2007) Recent progress of patterned media. IEEE Trans Magn 43:3685–3688
33. Koetter R, Singer AC, Tuchler M (2004) Turbo equalization. IEEE Signal Process Mag 21: 67–80
34. Kou Y, Lin S, Fossorier MPC (2001) Low-density parity-check codes based on finite geometries: a rediscovery and new results. IEEE Trans Inf Theory 47:2711–2736

35. Kschischang FR, Frey BJ, Loeliger HA (2001) Factor graphs and the sum-product algorithm. IEEE Trans Inf Theory 47:498–519
36. Lee I, Cioffi J (1997) Equalized maximum likelihood receiver with a unit energy constraint. IEEE Trans Magnetics 33:855–862
37. Lee I, Yamauchi T, Cioffi J (1994) Modified maximum likelihood sequence estimation in a simple partial erasure model. In: Proceedings Of IEEE Internal Conference on Communications (ICC), LA, USA, pp 245–249
38. Li C, Mathew G (2006) Analytical solution for optimum partial-response target in viterbi-based receivers. IEEE Trans Commun 54:1715–1719
39. Lim F, Kavcic A (2005) Optimal precompensation for partial erasure and nonlinear transition shift in magnetic recording using dynamic programming. In: Proceedings of IEEE International Conference on Global Telecommunications (GLOBECOM), Missouri, USA, pp 58–62
40. Lin S, Costello DJ (2004) Error control coding: fundamentals and applications. 2nd edn. Prentice Hall
41. Liu B, Lee Y, Mutoh H, Garg H (1997) The effect of MR head nonlinerity on MDFE and PRML performance. IEEE Trans Magn 33:2764–2766
42. Luby M, Mitzenmacher M, Shokrollahi A, Spielman D (1998) Analysis of low density codes and improved designs using irregular graphs. In: Proceedings of 30th Annual ACM Symposium Theory of Computing, Texas, USA, pp 249–258
43. MacKay DJC (1999): Good error-correcting codes based on very sparse matrices. IEEE Trans Inf Theory 45:399–431
44. MacKay DJC, Neal RM (1996) Near Shannon limit performance of low density parity check codes. Electron Lett 32:1645–1646
45. Mansour M, Shanbhag N (2006) A 640-Mb/s 2048-bit programmable LDPC decoder chip. IEEE J Solid State Circuits 41:684–698
46. Mathew G, Farhang-Boroujeny B, Wood R (1997) Design of multilevel decision feedback equalizers. IEEE Trans Magn 33:4528–4542
47. Mee C, Daniel E (1996) Magnetic recording technology, vol 3, chap. 8. McGraw Hill, New York
48. Moon J (1998) The role of SP in data-storage. IEEE Signal Process Mag 15:54–72
49. Moon J (2000) Signal-to-noise ratio definition for magnetic recording channels with transition noise. IEEE Trans Magn 36:3881–3883
50. Moon J, Zeng W (1995) Equalization for maximum likelihood detectors. IEEE Trans Magn 31:1083–1088
51. Moon J, Zhu J (1991) Nonlinear effects of transition broadening. IEEE Trans Magn 27: 4831–4833
52. Palmer D, Ziperovich P, Wood R, Howell T (1987) Identification of nonlinear write effects using pseudo-random sequences. IEEE Trans Magn 23:2377–2379
53. Patterson R, Gibson G, Ginting E, Stodolsky D, Zelenka J (1995) Informed prefetching and caching. In: Proceedings of ACM symposium on operating systems principles, Colorado, USA, pp 79–95
54. Proakis J (2005) Partial response equalization with application to high density magnetic recording channels. In: Vasic B, Kurtas EM (eds) Coding and signal processing fro magnetic recording systems. CRC Press, 8:1–23
55. Qureshi S (1985) Adaptive equalization. Proc IEEE 73:1349–1387
56. Riani J, van Beneden S, Bergmans J, Immink A (2007) Near minimum-BER equalizer adaptation for PRML systems. IEEE Trans Commun 55:2316–2327
57. Rottmayer R, Batra S, Buechel D, Challener W, Hohlfeld J, Kubota Y, Li L, Lu B, Mihalcea C, Mountfield K, Pelhos K, Peng C, Rausch T, Seigler M, Weller D, Yang X (2006) Heat-assisted magnetic recording. IEEE Trans Magn 42:2417–2421
58. Sarigoz F, Song H, Kumar B, Bain J (2001) Dropout-tolerant read channels. IEEE J Selected Areas Commun 19:744–755

59. Sawatzky E (1998) Thermal asperities: MR heads face new dangers. Data Storage Mag 5:49–54
60. Smith A (1985) Disk cachemiss ratio analysis and design considerations. ACM Trans Comput Syst 3:161–203
61. Tan W, Cruz J (2005) Detection of media defects in perpendicular magnetic recording channels. IEEE Trans Magn 41:2956–2958
62. Taratorin A, Klaassen K (2006) Media saturation and overwrite in perpendicular recording. IEEE Trans Magn 42:157–162
63. Wang S, Taratorin A (1999) Magnetic information storage technology, chap. 9. Academic, New York, NY
64. Wiberg N (1996) Codes and decoding on general graphs. PhD dissertation, Linkoping University, Sweden
65. Wicker SB, Bhargava VK (1994) Reed-Solomon codes and their applications. IEEE Press, New York, NY
66. Worthington B, Ganger G, Patt Y (1994) Scheduling algorithms for modern disk drives. In: Proceedings of ACM SIGMETRICS conference on Measurement and modeling of computer systems, Tennessee, USA, pp 241–251
67. Wu Z, Burd G (2005) Equation based LDPC decoder for inter symbol interference channels. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp 757–760
68. Xing X, Bertram H (1999) Error rate analysis of partial response channels in the presence of texture noise. IEEE Trans Magn 35:2070–2079
69. Yang X, Kurtas E (2005) Signal and noise generation for magnetic recording channel simulations. In: Vasic B, Kurtas EM (eds) Coding and signal processing fro magnetic recording systems, CRC Press, Boca Raton, FL, 6:1–20
70. Yeo E, Augsburger S, Davis W, Nikolic B (2003) A 500 Mb/s soft-output Viterbi decoder. IEEE J Solid-State Circuits 38:1234–1241
71. Yuan S, Bertram H (1992) Statistical data analysis of magnetic recording noise mechanisms. IEEE Trans Magn 28:84–92
72. Ziperovich P (1991) Performance degradation of PRML channels due to nonlinear distortions. IEEE Trans Magn 27:4825–4827