Chapter 5
BASIC PRINCIPLES OF OPERATION

This chapter describes the operation of the Quantum Fireball TM 1.0/1.2/1.7/2.1/2.5/3.2/3.8AT hard disk drives' functional subsystems. It is intended as a guide to the operation of the drive, rather than a detailed theory of operation.

5.1 QUANTUM FIREBALL TM DRIVE MECHANISM

This section describes the drive mechanism. Section 5.2 describes the drive electronics. The Quantum Fireball TM 1.0/1.2/1.7/2.1/2.5/3.2/3.8AT hard disk drives consist of a mechanical assembly and a PCB as shown in Figure 5-1. The drawing is illustrated with two disks, showing the Quantum Fireball TM 2110AT drive configuration. The Quantum Fireball TM 1080AT hard disk drive contains only one hard disk, and the 3840AT contains three hard disks.

The head/disk assembly (HDA) contains the mechanical subassemblies of the drive, which are sealed under a metal cover. The HDA consists of the following components:

- Base casting
- DC motor assembly
- Disk stack assembly
- Headstack assembly
- Rotary positioner assembly
- Automatic actuator lock
- Air filter

The drive is assembled in a Class-100 clean room.

CAUTION: To ensure that the air in the HDA remains free of contamination, never remove or adjust its cover and seals. Tampering with the HDA will void your warranty.

The Quantum Fireball TM drives are a one, two, or three disk product family. The Quantum Fireball TM 1080AT and 1280AT hard disk drive contains one magnetic disk and two read/write heads. Quantum Fireball TM 1700AT hard disk drive contains two magnetic disks and three read/write heads. Quantum Fireball TM 2110AT and 2550 AT drive contains two magnetic disks and four read/write heads. Quantum Fireball TM 3200AT drive contains three magnetic disks and five read/write heads, and the Quantum Fireball TM 3280AT drive contains three magnetic disks and six read/write heads.

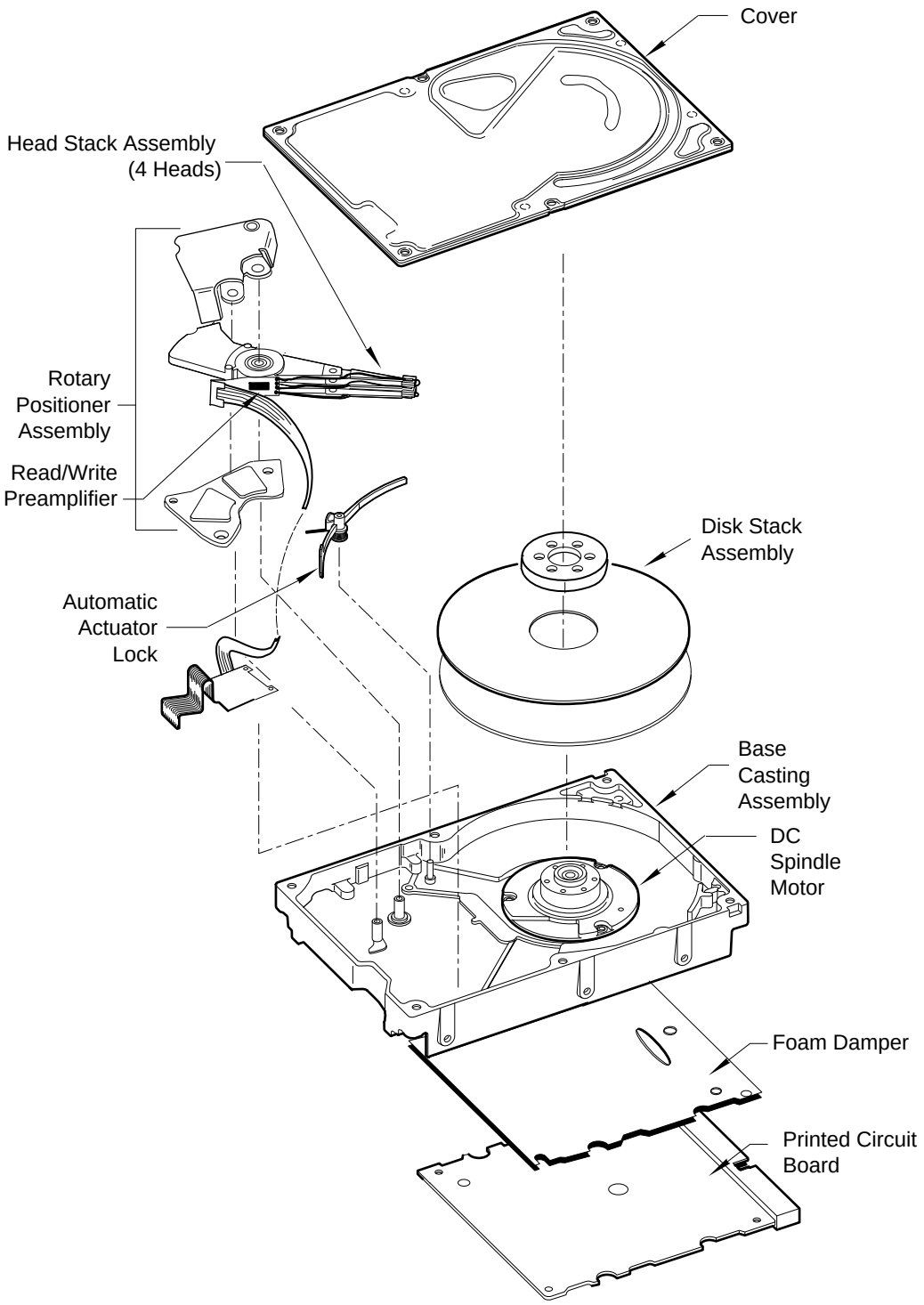

Figure 5-1 *Quantum Fireball TM (Two-Disk) Drive Exploded View*

5.1.1 Base Casting Assembly

A single-piece, aluminum-alloy base casting provides a mounting surface for the drive mechanism and PCB. The base casting also acts as the flange for the DC motor assembly. To provide a contamination-free environment for the HDA, a gasket provides a seal between the base casting, and the metal cover that encloses the drive mechanism.

5.1.2 DC Motor Assembly

Integral with the base casting, the DC motor assembly is a fixed-shaft, brushless DC spindle motor that drives the counter-clockwise rotation of the disks.

5.1.3 Disk Stack Assemblies

The disk stack assembly in the Quantum Fireball TM 1080AT hard disk drive consists of one disk secured by a disk clamp. The Quantum Fireball TM 2110/3200AT hard disk drives contain two and three disks. The aluminum-alloy disks have a sputtered thin-film magnetic coating.

A carbon overcoat lubricates the disk surface. This prevents head and media wear due to head contact with the disk surface during head takeoff and landing. Head contact with the disk surface occurs only in the landing zone outside of the data area, when the disk is not rotating at full speed. The landing zone is located at the inner diameter of the disk, beyond the last cylinder of the data area.

The Quantum Fireball TM 1.0/1.2/1.7/2.1/2.5/3.2/3.8AT hard disk drives have 6,816 tracks per recording surface. Of these tracks, 6 are used for system data, leaving 6,810 for data. The data tracks are divided into 15 recording zones. The drive uses multiple zone recording, where each data track contains between 104 and 210 sectors for the 1080 MB per disk format, and 122 and 232 sectors for the 1280 MB per disk format depending on the recording zone. The sectors per track allocation for each zone is provided in Table 5-1.

Table 5-1 *Cylinder Contents*

		NO. OF TRACKS		SECTORS/TRACK		DATA RATE (Mbit/s)	
CYLINDER CONTENTS	**ZONE**[1]	**1080 MB Format**	**1280 MB Format**	**1080 MB Format**	**1280 MB Format**	**1080 MB Format**	**1280 MB Format**
System Data	System	6	6	135	135	54.04	54.04
User Data	0	454	454	210	232	84.26	92.86
	1	454	454	204	229	81.88	91.69
	2	454	454	198	225	79.26	90.35
	3	454	454	187	225	75.29	89.16
	4	454	454	180	214	72.40	85.75
	5	454	454	180	205	70.41	82.14
	6	454	454	166	195	66.79	77.86
	7	454	454	157	185	63.19	74.40
	8	454	454	150	180	60.24	71.37
	9	454	454	141	170	57.10	68.24
	10	454	454	131	162	53.11	65.16
	11	454	454	124	153	50.59	61.74
	12	454	454	120	142	48.77	57.37
	13	454	454	112	135	45.71	53.68
	14	454	454	104	122	42.56	49.50
Sum/ Average	—	6816	6816	158	185	63.44	74.08

1. For user data, zone 14 is the innermost zone and zone 0 is the outermost zone.

5.1.4 Headstack Assembly

The headstack assembly consists of read/write heads, an E-block and coil joined together by insertion molding to form an E-block/coil subassembly, bearings, and a flex circuit. Read/write heads mounted to spring-steel flexures are swage mounted onto the rotary positioner assembly arms. The E-block is a single piece, die-cast design.

The flex circuit exits the HDA between the base casting and the cover. A cover gasket seals the gap. The flex circuit connects the headstack assembly to the PCB. The flex circuit contains a read preamplifier/write driver IC.

5.1.5 Rotary Positioner Assembly

The rotary positioner, or rotary voice-coil actuator, is a Quantum-proprietary design that consists of an upper permanent magnet plate and lower flux plate bolted to the base casting, a rotary single-phase coil molded around the headstack mounting hub, and a bearing shaft. The single bi-polar magnet consists of two alternating poles and is bonded to the magnet plate. A resilient crash stop prevents the heads from being driven into the spindle or off the disk surface

Current from the power amplifier induces a magnetic field in the voice coil. Fluctuations in the field around the permanent magnet cause the voice coil to move. The movement of the voice coil positions the heads over the requested cylinder.

5.1.6 Automatic Actuator Lock

To ensure data integrity and prevent damage during shipment, the drive uses a dedicated landing zone and Quantum's patented Airlock®. The Airlock holds the headstack in the landing zone whenever the disks are not rotating. It consists of an air vane mounted near the perimeter of the disk stack, and a locking arm that restrains the actuator arm assembly.

When DC power is applied to the motor and the disk stack rotates, the rotation generates an airflow on the surface of the disk. As the flow of air across the air vane increases with disk rotation, the locking arm pivots away from the actuator arm, enabling the headstack to move out of the landing zone. When DC power is removed from the motor, an electronic return mechanism automatically pulls the actuator into the landing zone, where the Airlock holds it in place.

5.1.7 Air Filtration

The Quantum Fireball TM 1.0/1.2/1.7/2.1/2.5/3.2/3.8AT hard disk drives are Winchester-type drives. The heads fly very close to the media surface. Therefore, it is essential that the air circulating within the drive be kept free of particles. Quantum assembles the drive in a Class-100 purified air environment, then seals the drive with a metal cover. When the drive is in use, the rotation of the disks forces the air inside of the drive through an internal filter.

The highest air pressure within the HDA is at the outer perimeter of the disks. A constant stream of air flows through a 0.3-micron circulation filter positioned in the base casting. As illustrated in Figure 5-2, air flows through the circulation filter in the direction of disk rotation. This design provides a continuous flow of filtered air when the disks rotate.

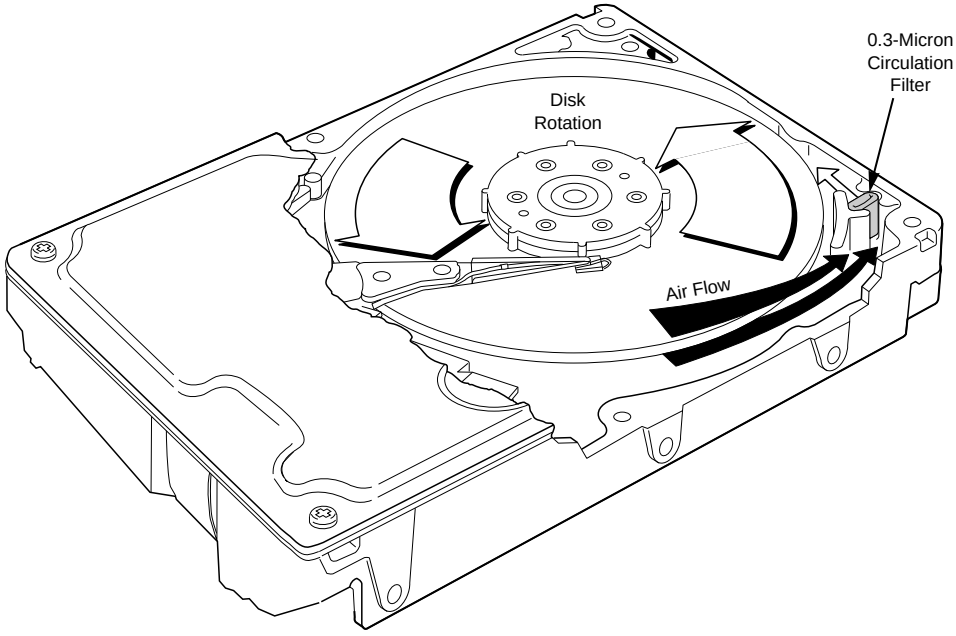

Figure 5-2 *HDA Air Filtration*

5.2 DRIVE ELECTRONICS

Advanced circuit design, the use of miniature surface-mounted components, and proprietary VLSI integrated circuits enable the drive electronics, including the IDE bus interface, to reside on a single printed circuit board assembly (PCBA).

Note: The components are mounted on only one side of the PCB.

Figure 5-3 contains a simplified block diagram of the Quantum Fireball TM 1.0/1.2/1.7/2.1/2.5/3.2/3.8AT hard disk drive electronics.

The only electrical component not on the PCBA is the PreAmplifier and Write Driver IC. It is on the flex circuit (inside of the sealed HDA). Mounting the preamplifier as close as possible to the read/write heads improves the signal-to-noise ratio. The flex circuit (including the PreAmplifier and Write Driver IC) provides the electrical connection between the PCB, the rotary positioner assembly, and read/write heads.

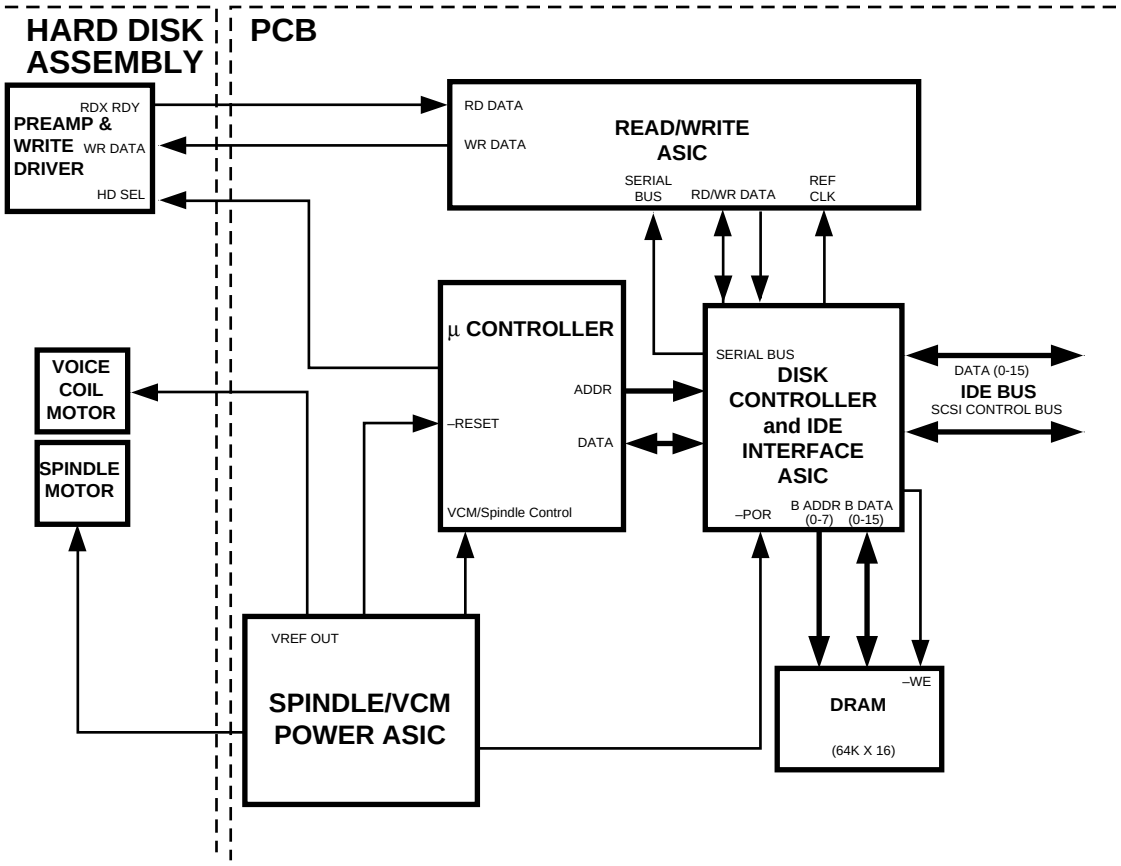

Figure 5-3 *Quantum Fireball TM 1.0/1.2/1.7/2.1/2.5/3.2/3.8AT Hard Disk Drive Block Diagram*

5.2.1 µController

The µController provides local processor services to the drive electronics under program control. The µController manages the resources of the Disk Controller and IDE Interface ASIC (DCIIA), the Read/Write ASIC, and the Spindle/VCM Driver. In addition, it controls the head selection process.

An internal 32Kbyte ROM contained within the µController provides program code that the µController executes to complete a drive spinup and recalibration procedure, after which the µController reads additional control code from the disk (diskware), and stores it in the buffer DRAM.

5.2.2 DCIIA

The DCIIA (Disk Controller and IDE Interface ASIC) shown in Figure 5-4 provides control functions to the drive under the direction of the µController.

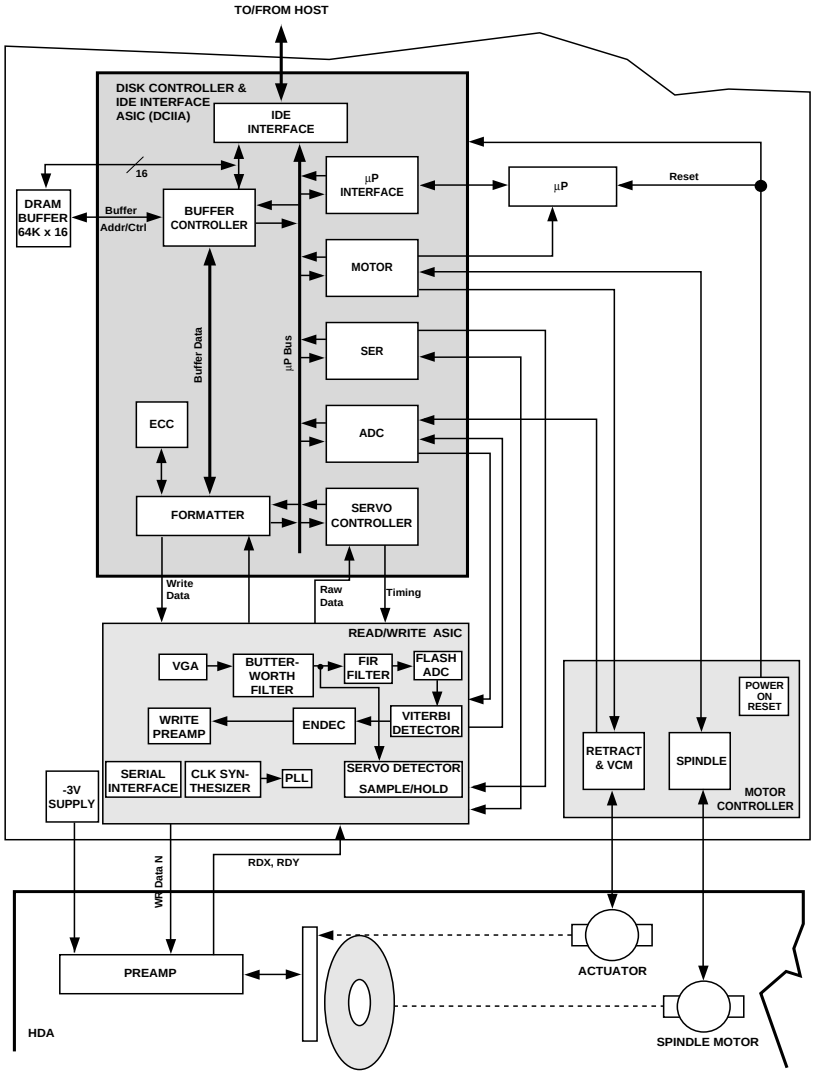

Figure 5-4 *DCIIA Block Diagram*

The DCIIA is a proprietary ASIC developed by Quantum. The DCIIA is comprised of eight functional modules (described below):

- 8-bit A/D Converter
- Error Correction Control
- Motor Control
- Sequencer (Formatter)
- Analog Phase Lock Loop
- Buffer Controller
- µController Interface
- Servo Controller, including PWM
- Serial Interface
- IDE Interface Controller

5.2.2.1 A/D Converter

The Analog to Digital converter (A/D) receives multiplexed burst analog inputs from the Read/Write ASIC. The A/D is used to sample the demodulated position information (burst inputs), and to convert it to a digital signal which the Servo Controller uses to position the HDA actuator.

5.2.2.2 Error Correction Control

The Error Correction Control block utilizes a Reed-Solomon encoder/decoder circuit that is used for disk read/write operations. It uses a total of 28 redundancy bytes organized as 24 ECC (Error Correction Code) bytes and four cross-check bytes. The ECC uses eight bits per symbol and four interleaves. This allows triple-burst error correction of at least 65, and as many as 96 bits in error.

5.2.2.3 Sequencer (Formatter)

The sequencer controls the operation of the read and write channel portions of the DCIIA. To initiate a disk operation, the µController loads a set of commands into the WCS (writable control store) register. Loading and manipulating the WCS is done through the µController Interface registers.

The sequencer also directly drives the read and write gates (RG, WG) of the Read/Write ASIC and the R/W Preamplifier, as well as passing write data to the Precompensator circuit in the Read/Write ASIC.

5.2.2.4 Buffer Controller

The buffer controller supports a 128 Kbyte buffer, which is organized as 64 K x 16 bits. The 16-bit width implementation provides a 60 MB/s maximum buffer bandwidth, which allows programmable maximum disk channel bandwidth. This increased bandwidth allows the µController to have direct access to the buffer, eliminating the need for a separate µController RAM IC.

The buffer controller supports both drive, and host address rollover and reloading to allow for buffer segmentation. Drive and host addresses may be separately loaded for automated read/write functions.

The Buffer Controller operates under the direction of the µController.

5.2.2.5 µController Interface

The µController Interface provides the means for the µController to read and write data to the DCIIA modules to control their operation, or supply them with needed information. It consists of both physical and logical components.

The physical component of the interface is comprised of the 16-bit MAD (Multiplexed Address/Data) bus, four additional address lines, read and write strobe, an address latch enable (ALE) signal, and a wait control line.

The logical component of the interface is comprised of internal control and data registers accessible to the µController. By writing and reading these registers, the µController loads the Sequencer, controls and configures the Buffer controller, and passes coded servo information to the Servo Controller.

5.2.2.6 Servo Controller

The Servo Controller contains a 13-bit Digital to Analog converter (D/A), in the form of a Pulse Width Modulator (PWM). The PWM signal is output to the Actuator Driver to control the motion of the actuator. The Servo Controller also decodes raw data from the disk to extract the current position information. The position information is read by the µController and is used to generate the actuator control signal that is sent to the PWM. The actuator driver is an analog power amplifier circuit external to the DCIIA. The Servo Controller operates under the direction of the µController.

5.2.2.7 Serial Interface

The Serial Interface provides a high speed Read/Write interface path to the Read/Write ASIC under the direction of the µController. Allows 10, 20, and 40 MHz operating modes.

5.2.2.8 IDE Interface Controller

The IDE Interface Controller portion of the DCIIA provides data handling, bus control, and transfer management services for the IDE interface. Configuration and control of the interface is accomplished by the µController across the MAD bus. Data transfer operations are controlled by the DCIIA Buffer Controller module.

5.2.2.9 Motor Controller

The Motor Controller block of the DCIIA in conjunction with the µP and the required firmware provides all the necessary functionality for Motor Spindle Commutation, Speed Control, as well as Actuator Voice Coil controls.

The Motor Controller consists of three basic functional blocks, the Motor Commutation logic, Motor Power Mode (PWM) logic, and the Voice Coil PWM Logic. Mode outputs are provided to support analog mode control for both the Spindle and the VCM.

5.2.3 Read/Write ASIC

The Read/Write ASIC shown in Figure 5-5 provides write data precompensation and read channel processing functions for the drive. The Read/Write ASIC receives the RD GATE signal, reference oscillator, serial programming, and servo burst and sample gates from the DCIIA. The Read/Write ASIC sends decoded read data and the read reference clock, and receives write data from the DCIIA. This a highly integrated circuit which is completely under digital control from the DCIIA.

The Read/Write ASIC comprises 11 main functional modules (described below):

- Pre-Compensator
- Variable Gain Amplifier (VGA)
- Butterworth Filter
- FIR Filter
- Flash A/D Converter
- Viterbi Detector
- ENDEC
- Servo Detector and Sample/Hold
- Clock Synthesizer
- PLL
- Serial Interface

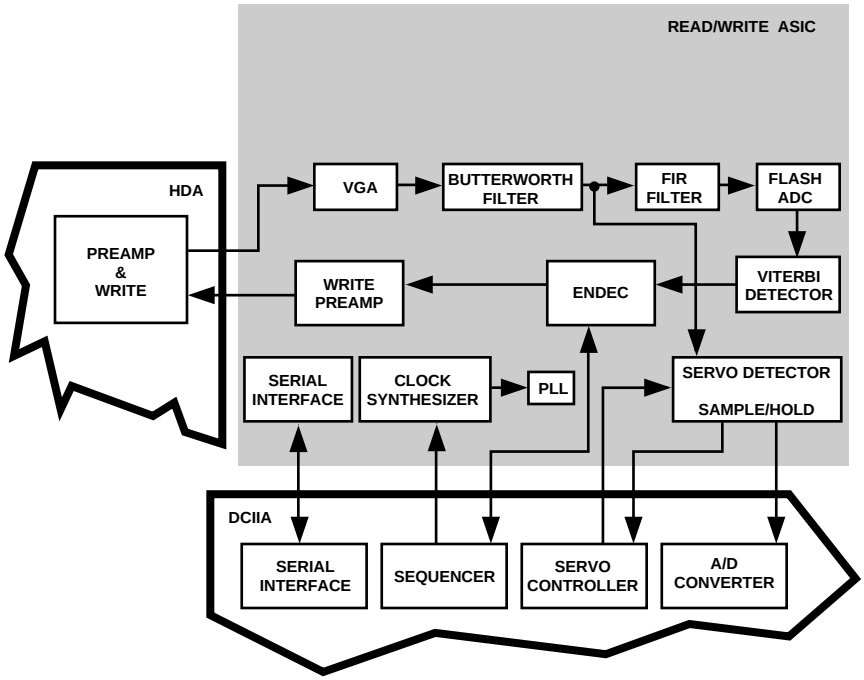

Figure 5-5 *Read/Write ASIC Block Diagram*

5.2.3.1 Pre-Compensator

The pre-compensator introduces pre-compensation to the write data received from the sequencer module in the DCIIA. The pre-compensated data is then passed to the R/W Pre-Amplifier and written to the disk. Pre-compensation reduces the write interference from adjacent write bit.

5.2.3.2 Variable Gain Amplifier (VGA)

Digital and analog controlled AGC function with input attenuator for extended range.

5.2.3.3 Butterworth Filter

Continuous time data filter which can be programmed for each zone rate.

5.2.3.4 FIR (Finite Impulse Response) Filter

Digitally controlled and programmable filter for partial response signal conditioning.

5.2.3.5 Flash A/D Converter

Provides very high speed digitization of the processed read signal.

5.2.3.6 Viterbi Detector

Decodes ADC result into binary bit stream.

5.2.3.7 ENDEC

Provides 16/17 code conversion to NRZ. Includes preamble and sync mark generation and detection.

5.2.3.8 Servo Detector and Sample/Hold

Peak detection with weighted averaging and multiple sample and hold of servo bursts.

5.2.3.9 Clock Synthesizer

Provides programmable frequencies for each zone data rate.

5.2.3.10 PLL

Provides digital read clock recovery.

5.2.3.11 Serial Interface

High speed interface for digital control of all internal blocks.

5.2.4 PreAmplifier and Write Driver

The PreAmplifier And Write Driver provides write driver and read pre-amplifier functions, and R/W head selection. The write driver receives precompensated write data from the PreCompensator module in the Read/Write ASIC. The write driver then sends this data to the heads in the form of a corresponding alternating current. The read pre-amplifier amplifies the low-amplitude differential voltages generated by the R/W heads, and transmits them to the VGA module in the Read/Write ASIC. Head select is determined by the µController.

5.3 SERVO SYSTEM

5.3.1 General Description

The servo system controls the positioning of the read/write heads, and holds the read/write heads on track during read/write operations. The servo system also compensates for thermal offsets between heads on different surfaces, and any shock and vibration the drive is subjected to.

The Quantum Fireball TM 1.0/1.2/1.7/2.1/2.5/3.2/3.8AT hard disk drives use a high performance embedded sectored servo system. Positioning information is radially encoded in evenly-spaced servo bursts on each track. These servo burst wedges provide radial position information for each data head. Because the drive uses multiple zone recording, where each zone has a different bit density, split data fields are necessary to optimally utilize the non-servo area of the disk. The split data fields are achieved by special processing through the DCIIA, and their presence is transparent to the host system. The servo area remains phase coherent across the surface of the disk, even though the disk is divided into various data zones. The main advantage of the sectored servo systems is that the data heads are also servo heads, which means that sectored servo systems eliminate the problems of static and dynamic offsets between heads on different surfaces.

The Quantum Fireball TM series of hard disk drives' servo system is also classified as a digital servo because track following compensation is done in firmware. The bump detect, on-track, velocity profiles, and other "housekeeping" tasks are also done in firmware.

The state of the servo system determines how the position information is derived. During low velocity seeking, the position signal is the convolution of the track number and A/C or B/D burst values, and has a 1/1000th of a track pitch resolution—about 0.1%. During high velocity seeking, the track pitch resolution is equal to 1 track. While track following, the A/C and B/D bursts are used for position information, and the resolution is at least 1/1000th of a track pitch—about 0.1%.

5.3.2 Servo Burst and Track Information

Positional information is encoded on all tracks on all data surfaces. All data heads are also servo heads. The areas with servo/position information are called wedge areas. These wedge areas are evenly spaced radially around the disk, like spokes on a wheel. There are 90 wedge areas per track. Since the disk rotation is 75.0 revolutions/second, the position information is updated at 6750 Hz (90 x 75.0). This is also known as the sampling frequency fs. The sample period, Ts, is $1/f$s = 148.15 µs. Every wedge area consists of four separate fields: (1) Automatic Gain Control (AGC)/Sync field, (2) Servo Address Mark (SAM) field, (3) Track number and (4) Burst area. Since a Phase Lock Loop (PLL) is not used in the servo wedge area, time discrimination is used. Timing for all four fields is generated from the same crystal reference.

5.4 READ AND WRITE OPERATIONS

The following paragraphs provide descriptions of the read channel, write channel, and IDE interface control operations.

5.4.1 The Read Channel

The drive has one read/write head for each data surface (two for Quantum Fireball TM 1080AT and 1280AT drives; four for Quantum Fireball TM 1700AT, 2110AT, and 2550AT drives; five for Quantum Fireball TM 3200AT drives; and six for Quantum Fireball TM 3840AT drives). The signal path for the read channel begins at the read/write heads. As the magnetic flux transitions recorded on a disk pass under a head, they generate low-amplitude, differential output voltages. These read signals pass from the read/write head to the flex circuit's read preamplifier, which amplifies the signal. To ensure a high signal to noise ratio, preamplification occurs on the flex circuit because of its proximity to the heads.

The flex circuit transmits the preamplified signal from the HDA to the drive PCB. On the PCB, the Read/Write ASIC further processes the read signal to reduce ambiguities, for example, drop-ins, drop-outs, and ISI (Inter-symbol Interferences). In addition, it converts the signal from the serial encoded head data to a synchronized data stream, with its accompanying clock. The Read/Write ASIC then sends the resynchronized and decoded data output to Quantum's proprietary Disk Controller and IDE Interface ASIC (DCIIA).

The DCIIA manages the flow of data between the Read/Write ASIC and its IDE Interface Controller. It also controls data access for the external RAM buffer. The DCIIA format provides a serial bit stream. This NRZ (Non-Return to Zero) serial data is converted to an 8-bit byte. The Sequencer module identifies the data as belonging to the target sector. Data is presented to the host in a 16-bit word.

After a full sector is read, the DCIIA checks to see if the firmware needs to apply ECC on-the-fly, or single-, double-, or triple-burst correction to the data. The buffer controller section of the DCIIA stores the data in the Cache and transmits the data to the IDE Interface Controller module, which transmits the data to the IDE bus.

5.4.2 The Write Channel

For the write channel, the signal path follows the reverse order of that for the read channel. The host presents a 16-bit word of data, by means of the IDE bus, to the DCIIA IDE Interface Controller. The Buffer Controller section of the DCIIA stores the data in the cache. Because data can be presented to the drive at a rate that exceeds the rate at which the drive can write data to a disk, data is stored temporarily in the cache. Thus, the host can present data to the drive at a rate that is independent of the rate at which the drive can write data to the disk.

Upon correct identification of the target address, the data is shifted to the Sequencer where an error correcting code is generated and appended. The Sequencer then converts the bytes of data to a serial bit stream. The DCIIA transmits the data to the Read/Write ASIC, where the data is encoded and precompensated to reduce intersymbol interference. The data is then transmitted to the Write Driver by means of the write data lines.

The drive's DCIIA switches the Preamplifier Write Driver IC to write mode and selects a head. Once the Write driver receives a write gate signal, it transmits current reversals to the head, which induces magnetic transitions on the disk.

5.4.3 Interface Control

The interface with the host system is through a 40-pin IDE interface connector. The DCIIA IDE Interface Controller module implements the IDE interface logic. Operating under the drive's µprocessor control, the DCIIA receives and transmits words of data over the IDE bus.

The DCIIA Buffer Controller writes data to, or reads data from the Cache over 16 data lines. Under µController direction, the DCIIA controls the transfer of data and handles the addressing of the Cache. The internal data transfer rate to and from the Cache is 32 MB/s. This high transfer rate allows the DCIIA to communicate over the IDE interface at a PIO data transfer rate of 6.67 MB/s without using IORDY, up to 16.67 MB/s with PIO using IORDY, or a DMA transfer rate of up to 16.67 MB/s while it simultaneously controls disk-to-RAM transfers, and microcontroller access to control code stored in the buffer RAM.

5.4.4 ID-Less Format

The Quantum Fireball TM series of hard disk drives utilize an ID-Less format. The ID-Less Format has several advantages over the traditional 'ID After Wedge' or 'ID Before Sector' methods of tracking the location of the actuator. For example, the lack of an ID field written on the disk gains approximately 4% of the overall track 'real-estate', thus increasing the total capacity. Secondly, since no ID's have to be read or corrected in case of an error, overall throughput is increased. In ID-Less formatting, the ID of each sector is not written on the disk after the servo wedge. Instead, it is stored in the buffer RAM and called the Descriptor. Each sector has an associated descriptor which contains the following basic information. The servo wedge number after which the sector is located, the sector start time after the wedge, and when to skip over the next wedge. The descriptor does not have defect information. The defect map is also stored in the buffer RAM but in a separate location. The formatter section of the DCIIA will access both, the descriptor and the defect lists through a request to the buffer block of the DCIIA. Only the user data and the ECC information is actually written to the disk.

Table 5-2 *ID-Less Controller Descriptor Format*

BYTE	BIT							
	7	6	5	4	3	2	1	0
0	PARITY	WEDGE NUMBER						
1					MSB	SECTOR MARK		
2	SECTOR MARK							LSB
3	BRK COUNT 2				00H RESERVED			

Note: For a split sector, the descriptor will comprise of three bytes. For a triple sector, the descriptor will comprise of four bytes.

5.4.4.1 Descriptor Byte Functions

Parity
This is the odd parity bit for the 3-byte or 4-byte descriptor. If the 'parena' bit in the CONFIG register of the DCIIA is set, and the formatter detects a parity error in the descriptor, then the State Machine will interrupt the µP. If the parity bit is not set the formatter will ignore any parity errors.

Wedge Number
This is the wedge number of this sector. It is compared to the internal wedge counter of the formatter to determine if this is the right wedge for this sector.

Brk Count 1
This value is used in conjunction with the pre-wedge signal from the DCIIA's servo register. It provides the exact location where to split the sector and skip over the wedge.

Sector Mark
The Sector Mark is the starting time of this sector. This 12-bit value is sent to the DCIIA's servo register, and compared with the MSB of its Sector Timer to determine the start of the sector.

Brk Count 2
Same as Brk Count 1. This value is used to determine where to split the second segment in a triple split sector.

5.5 FIRMWARE FEATURES

This section describes the following firmware features:

- Disk caching
- Track and cylinder skewing
- Error detection and correction
- Defect management

5.5.1 Disk Caching

The Quantum Fireball TM 1.0/1.2/1.7/2.1/2.5/3.2/3.8AT hard disk drives incorporate DisCache, a 76 K disk cache, to enhance drive performance. This integrated feature is user-programmable and can significantly improve system throughput. Read and write caching can be enabled or disabled by using the Set Configuration command.

5.5.1.1 Adaptive Caching

The cache buffer for the Quantum Fireball TM series of hard disk drives feature adaptive segmentation for more efficient use of the buffer's RAM. With this feature, the buffer space used for read and write operations is dynamically allocated. The cache can be flexibly divided into several segments under program control. Each segment contains one cache entry.

A cache entry consists of the requested read data plus its corresponding prefetch data. Adaptive segmentation allows the drive to make optimum use of the buffer. The amount of stored data can be increased.

5.5.1.2 Read Cache

DisCache anticipates host-system requests for data and stores that data for faster access. When the host requests a particular segment of data, the caching feature uses a prefetch strategy to "look ahead," and automatically store the subsequent data from the disk into high-speed RAM. If the host requests this subsequent data, the RAM is accessed rather than the disk.

Since typically 50 percent or more of all disk requests are sequential, there is a high probability that subsequent data requested will be in the cache. This cached data can be retrieved in microseconds rather than milliseconds. As a result, DisCache can provide substantial time savings during at least half of all disk requests. In these instances, DisCache could save most of the disk transaction time by eliminating the seek and rotational latency delays that dominate the typical disk transaction. For example, in a 1K data transfer, these delays make up to 90 percent of the elapsed time.

DisCache works by continuing to fill its cache memory with adjacent data after transferring data requested by the host. Unlike a noncaching controller, Quantum's disk controller continues a read operation after the requested data has been transferred to the host system. This read operation terminates after a programmed amount of subsequent data has been read into the cache segment.

The cache memory consists of a 76 K DRAM buffer allocated to hold the data, which can be directly accessed by the host by means of the READ and WRITE commands. The memory functions as a group of segments (ring buffers) with rollover points at the end of each segment (buffer). The unit of data stored is the logical block (that is, a multiple of the 512 byte sector). Therefore, all accesses to the cache memory must be in multiples of the sector size.The following commands force emptying of the cache:

- WRITE BUFFER
- SET FEATURES
- DRIVE FAILURE PREDICTION
- WRITE CONFIGURATION
- DOWNLOAD
- BUFFER RAM TEST

5.5.1.3 Write Cache

When a write command is executed with write caching enabled, the drive stores the data to be written in a DRAM cache buffer, and immediately sends a GOOD STATUS message to the host before the data is actually written to the disk. The host is then free to move on to other tasks, such as preparing data for the next data transfer, without having to wait for the drive to seek to the appropriate track, or rotate to the specified sector.

While the host is preparing data for the next transfer, the drive immediately writes the cached data to the disk, usually completing the operation in less than 20 ms after issuing GOOD STATUS. With WriteCache, a single-block, random write, for example, requires only about 3 ms of host time. Without WriteCache, the same operation would occupy the host for about 20 ms.

WriteCache allows data to be transferred in a continuous flow to the drive, rather than as individual blocks of data separated by disk access delays. This is achieved by taking advantage of the ability to write blocks of data sequentially on a disk that is formatted with a 1:1 interleave. This means that as the last byte of data is transferred out of the write cache and the head passes over the next sector of the disk, the first byte of the of the next block of data is ready to be transferred, thus there is no interruption or delay in the data transfer process.

The WriteCache algorithm fills the cache buffer with new data from the host while simultaneously transferring data to the disk that the host previously stored in the cache.

5.5.1.4 Performance Benefits

In a drive without DisCache, there is a delay during sequential reads because of the rotational latency even if the disk actuator already is positioned at the desired cylinder. DisCache eliminates this rotational latency, time (6.67 ms on average) when requested data resides in the cache.

Moreover, the disk must often service requests from multiple processes in a multitasking or multiuser environment. In these instances, while each process might request data sequentially, the disk drive must share time among all these processes. In most disk drives, the heads must move from one location to another. With DisCache, even if another process interrupts, the drive continues to access the data sequentially from its

high-speed memory. In handling multiple processes, DisCache achieves its most impressive performance gains, saving both seek and latency time when desired data resides in the cache.

The cache can be flexibly divided into several segments under program control. Each segment contains one cache entry. A cache entry consists of the requested read data plus its corresponding prefetch data.

The requested read data takes up a certain amount of space in the cache segment. Hence, the corresponding prefetch data can essentially occupy the rest of the space within the segment. The other factors determining prefetch size are the maximum and minimum prefetch. The drive's prefetch algorithm dynamically controls the actual prefetch value based on the current demand, with the consideration of overhead to subsequent commands.

5.5.2 Track and Cylinder Skewing

Track and cylinder skewing in the Quantum Fireball TM 1.0/1.2/1.7/2.1/2.5/3.2/3.8AT drives minimize latency time and thus increases data throughput.

5.5.2.1 Track Skewing

Track skewing reduces the latency time that results when the drive must switch read/write heads to access sequential data. A track skew is employed such that the next logical sector of data to be accessed will be under the read/write head once the head switch is made, and the data is ready to be accessed. Thus, when sequential data is on the same cylinder but on a different disk surface, a head switch is needed but not a seek. Since the sequential head-switch time is well defined on the Quantum Fireball TM series of hard disk drives, the sector addresses can be optimally positioned across track boundaries to minimize the latency time during a head switch. See Table 5-3.

5.5.2.2 Cylinder Skewing

Cylinder skewing is also used to minimize the latency time associated with a single-cylinder seek. The next logical sector of data that crosses a cylinder boundary is positioned on the drive such that after a single-cylinder seek is performed, and when the drive is ready to continue accessing data, the sector to be accessed is positioned directly under the read/write head. Therefore, the cylinder skew takes place between the last sector of data on the last head of a cylinder, and the first sector of data on the first head of the next cylinder. Since single-cylinder seeks are well defined on the Quantum Fireball TM 1.0/1.2/1.7/2.1/2.5/3.2/3.8AT drives, the sector addresses can be optimally positioned across cylinder boundaries to minimize the latency time associated with a single-cylinder seek. See Table 5-3.

5.5.2.3 Skewing with ID-less

In the ID-less environment, the drive's track and cylinder skewing will be based in unit of wedges instead of the traditional sectors. The DCIIA controller contains a "Wedge Skew Register" to assist in the task of skewing, where the skew offset must now be calculated with every read/write operation. The firmware will program the skew offset into this register every time the drive goes to a new track. The DCIIA will then add this value to the wedge number in the sector descriptor, effectively relocating the "first" sector of the track away from the index. For example, if without skew, sector 0 is to be found following wedge 0, then if the skew register is set to 10, sector 0 will be found following wedge 10.

Since the wedge-to-wedge time is constant over the entire disk, a single set of track and cylinder skew off-sets will fulfill the requirement for all recording zones.

5.5.2.4 Skew Offsets

Table 5-3 *Skews Offsets*

	SWITCH TIME	WEDGE OFFSET
Track Skew	3 ms	21
Cylinder Skew	4 ms	28

Note: Wedge-to-wedge time of 147.85 µs is used. Worst case spindle variation (-0.2%) is used while calculating to provide a safety margin.

Wedge offsets are rounded to the closest whole number.

5.5.2.5 Runtime Calculation

Since the wedge-to-wedge time is constant over the entire disk, a single set of track and cylinder skew offsets will fulfill the requirement for all recording zones. The formula used to compute the wedge skew for a given cylinder and head is:

Wedge skew = [C* ((# of heads – 1) * TS + CS) + H * TS] MOD 90

Where: C = Cylinder number
H = Head number
TS = Track Skew Offset
CS Cylinder Skew Offset
(wedges/track = 90)

5.5.3 Error Detection and Correction

As disk drive areal densities increase, obtaining extremely low error rates requires a new generation of sophisticated error correction codes. Quantum Fireball TM 1.0/1.2/1.7/2.1/2.5/3.2/3.8AT hard disk drive series implement 224-bit triple-burst Reed-Solomon error correction techniques to reduce the uncorrectable read error rate to less than one bit in 1×10^{14} bits read.

When errors occur, an automatic retry, a double-burst, and a more rigorous triple-burst correction algorithm enable the correction of any sector with three bursts of four incorrect bytes each, or up to twelve multiple random one-byte burst errors. In addition to these advanced error correction capabilities, the drive uses an additional cross-checking code and algorithm, to double check the main ECC correction. This greatly reduces the probability of a miscorrection.

5.5.3.1 Background Information on Error Correction Code and ECC On-the-Fly

A sector on the Quantum Fireball TM 1.0/1.2/1.7/2.1/2.5/3.2/3.8AT drive is comprised of 512 bytes of user data, followed by four cross-checking (XC) bytes (32 bits), followed by 24 ECC check bytes (192 bits). The four cross-checking bytes are used to double check the main ECC correction and reduce the probability of miscorrection. Errors of up to 64 bits within one sector can be corrected "on-the-fly," in real time as they occur, allowing a high degree of data integrity with no impact on the drive's performance.

The drive does not need to re-read a sector on the next disk revolution, or apply ECC for those errors that are corrected on-the-fly. Errors corrected in this manner are invisible to the host system.

When errors cannot be corrected on-the-fly, an automatic retry, and a more rigorous triple-burst error correction algorithm enables the correction of any sector with three bursts of four incorrect bytes each (up to 12 contiguous bytes), or up to 12 multiple random one-byte burst errors. In addition to this error correction capability, the drive's implementation of an additional cross-checking code and algorithm double checks the main ECC correction, and greatly decreases the likelihood of miscorrection.

The 24 ECC check bytes shown in Figure 5-6 are used to detect and correct errors. The cross-checking and ECC data is computed and appended to the user data when the sector is first written.

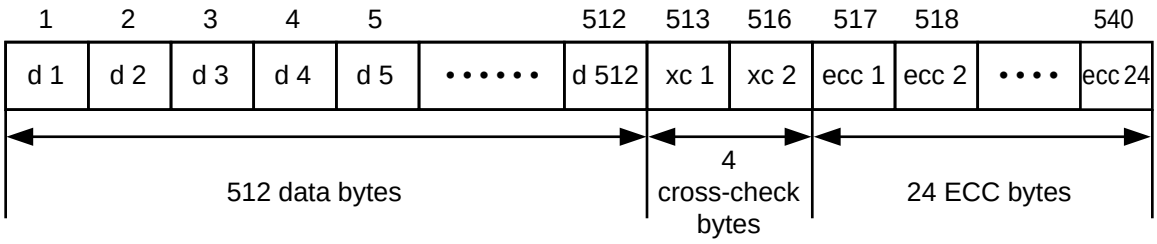

Figure 5-6 *Sector Data Field with ECC Check Bytes*

To obtain the ECC check byte values, each byte (including cross-checking and ECC bytes) within the sector is interleaved into one of three groups, where the first byte is in interleave 1, the second byte is in interleave 2, the third byte is in interleave 3, the fourth byte is in interleave 1, the fifth byte is in interleave 2, and so on, as shown in Figure 5-7.

Interleave 1 ⇨	d1	d5	• • • •	d508	d512	xc4	ecc4	ecc8	ecc12	ecc16	ecc20	ecc24
Interleave 2 ⇨	d2	d6	• • • •	d509	xc1	ecc1	ecc5	ecc9	ecc13	ecc17	ecc21	
Interleave 3 ⇨	d3	d7	• • • •	d510	xc2	ecc2	ecc6	ecc10	ecc14	ecc18	ecc22	
Interleave 4 ⇨	d4	d8	• • • •	d511	xc3	ecc3	ecc7	ecc11	ecc15	ecc19	ecc23	

Figure 5-7 *Byte Interleaving*

Note: ECC interleaving is not the same as the sector interleaving that is done on the disk.

Each of the four interleaves is encoded with six ECC bytes, resulting in the 24 ECC bytes at the end of the sector. The four cross checking bytes are derived from all 512 data bytes. The combination of the interleaving, and the nature of the ECC formulas enable the drive to know where the error occurs.

Because the ECC check bytes follow the cross checking bytes, errors found within the cross-checking bytes can be corrected. Due to the power and sophistication of the code, errors found within the ECC check bytes can also be corrected.

Each time a sector of data is read, the Quantum Fireball TM 1.0/1.2/1.7/2.1/2.5/3.2/3.8AT drives will generate a new set of ECC check bytes and cross-checking bytes from the user data. These new check bytes are compared to the ones originally written to the disk. The difference between the newly computed and original check bytes is reflected in a set of 24 *syndromes* and four cross checking syndromes, which correspond to the number of check bytes. If all the syndrome values equal zero, the data was read with no errors, and the sector is transferred to the host system. If any of the syndromes do not equal zero, an error has occurred. The type of correction the drive applies depends on the nature and the extent of the error.

High speed on-the-fly error correction saves several milliseconds on each single-, or double- burst error, because there is no need to wait for a disk revolution to bring the sector under the head for re-reading.

Correction of Single-, or Double-Burst Errors On-the-Fly

Single-burst errors may have up to four erroneous bytes (32 bits) within a sector, provided that each of the four bytes occur in a different interleave.

The Quantum Fireball TM 1.0/1.2/1.7/2.1/2.5/3.2/3.8AT drives have the capability to correct double-burst errors on-the-fly as well. Double-burst errors can be simply viewed as two spans of errors within one sector. More specifically, correctable double-burst errors must have two or fewer erroneous bytes per interleave.

The drive's Reed-Solomon ECC corrects double-burst errors up to 64 bits long, (provided that the error consists of two or fewer bytes residing in each of the interleaves).

Double-Burst Error Examples

In the example shown in Figure 5-8 C, the 58-bit error is uncorrectable since it occupies more than two erroneous bytes per interleave.

The other two 64-bit errors, shown in Figure 5-8 A and B, are correctable because no more than two error bytes of the entire error reside in any one of the interleaves.

Note: Any 57-bit error burst can be corrected on-the-fly using double-burst error correction because no more than two bytes can occupy each interleave.

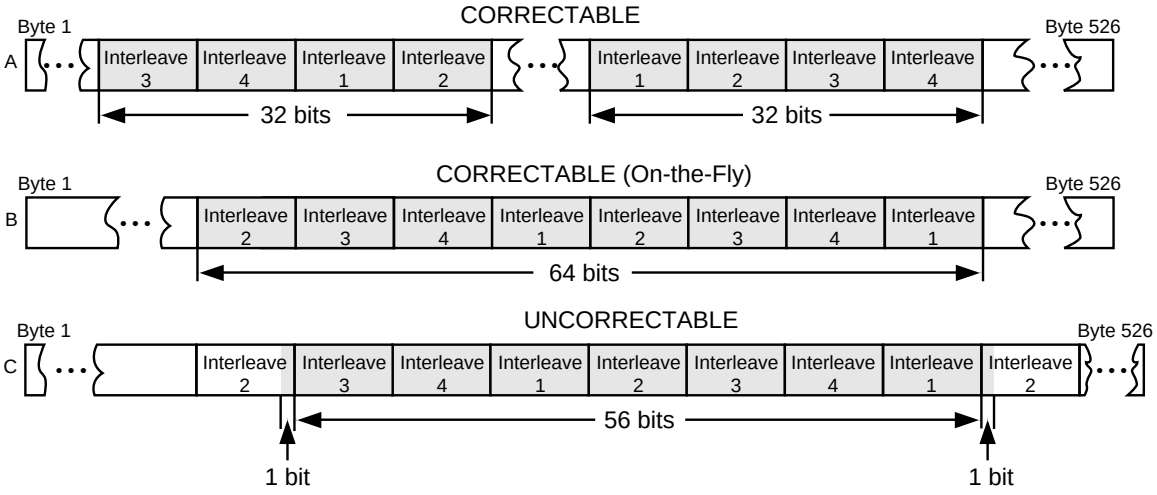

Figure 5-8 *Correctable and Uncorrectable Double-Burst Errors*

Correction of Triple-Burst Errors

Through sophisticated algorithms, Quantum Fireball TM 1.0/1.2/1.7/2.1/2.5/3.2/3.8AT drives have the capability to correct triple-burst errors, even though the probability of their occurrence is low. Triple-burst errors can be simply viewed as three spans of errors within one sector. More specifically, correctable triple-burst errors must have three or fewer erroneous bytes per interleave, and will not be corrected on-the-fly.

The drive's Reed-Solomon ECC corrects triple-burst errors up to 96 bits long, (provided that the error consists of three or fewer bytes residing in each of the interleaves).

If the triple-burst correction is successful, the data from the sector can be written to a spare sector, and the logical address will be mapped to the new physical location.

Triple-Burst Error Examples

In the example shown in Figure 5-9 C, the 90-bit error is uncorrectable since it occupies more than three erroneous bytes per interleave.

The other two 96-bit errors, shown in Figure 5-9 A and B, are correctable because no more than three error bytes of the entire error reside in any one of the interleaves.

Note: Any 89-bit error burst can be corrected using triple-burst error correction because no more than three bytes can occupy each interleave.

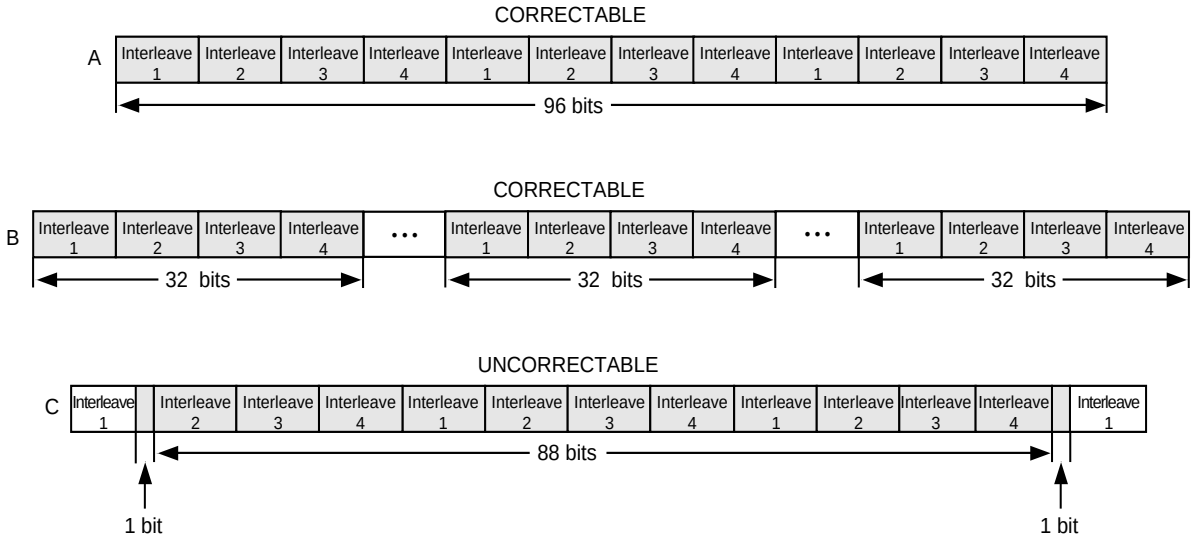

Figure 5-9 *Correctable and Uncorrectable Triple-Burst Errors*

Multiple Random Burst Errors

The drive's ECC can correct up to 96 bits of multiple random errors, provided that the incorrect bytes follow the guidelines for correctable triple-burst errors. Up to 64 bits of multiple random errors can be corrected on-the-fly, provided that the incorrect bytes follow the guidelines for correctable double-burst errors. Up to 24 bits of multiple random errors can be corrected on-the-fly if two bytes per interleave contains an error. If more than three bytes in any one interleave are in error, the sector cannot be corrected. Figure 5-10 shows an example of a correctable random burst error consisting of 12 bytes (96 bits). This random burst error is correctable because no more than three bytes within each interleave are in error.

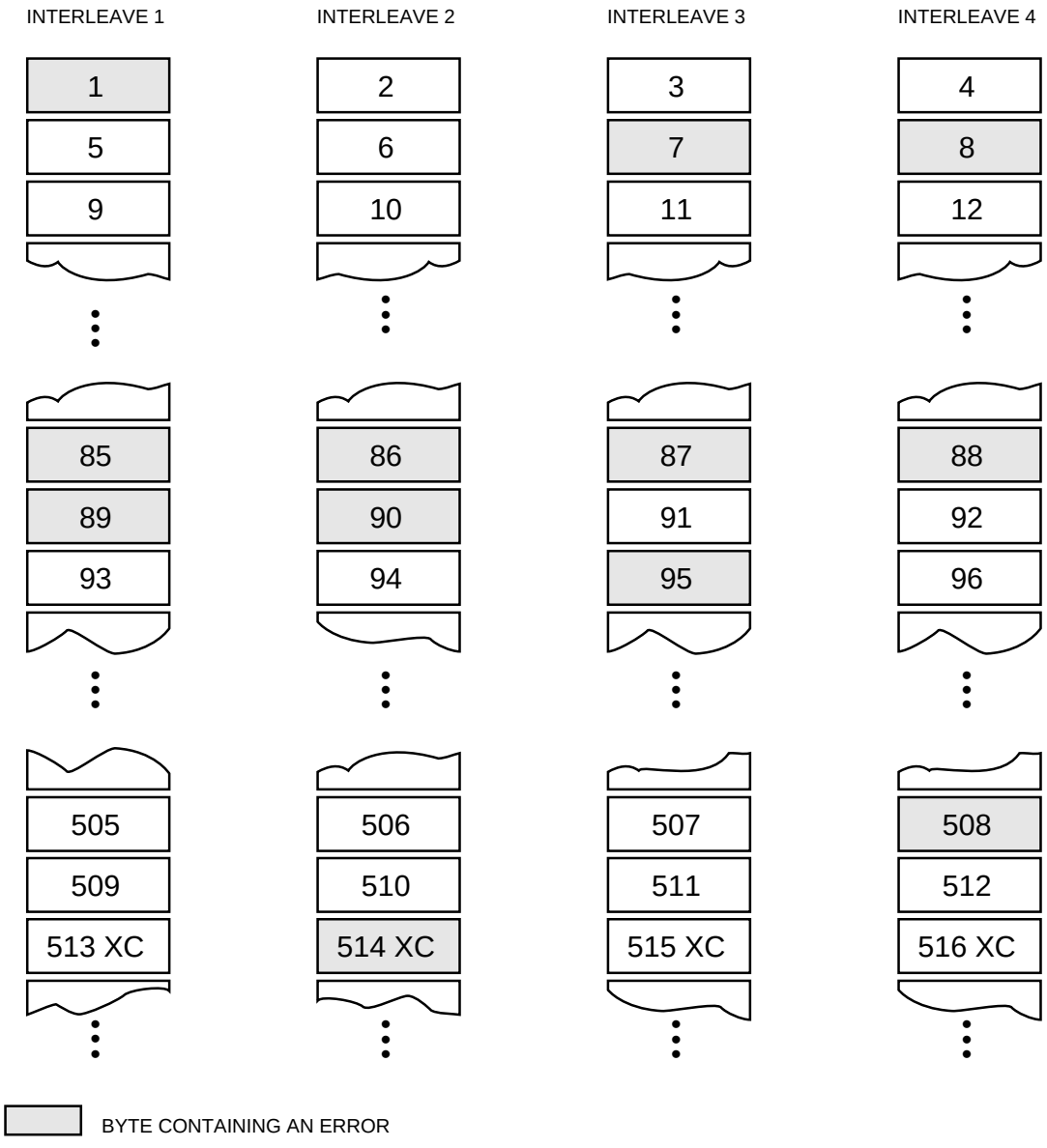

Figure 5-10 *Nine Correctable Random Burst Errors*

5.5.3.2 ECC Error Handling

When a data error occurs, the Quantum Fireball TM 1.0/1.2/1.7/2.1/2.5/3.2/3.8AT hard disk drives check to see if the error is correctable on-the-fly. This process takes about 200 µs. If the error is correctable on-the-fly, the error is corrected and the data is transferred to the host system.

If the data is not correctable on-the-fly, the sector is re-read in an attempt to read the data correctly without applying the triple-burst ECC correction. Before invoking the complex triple-burst ECC algorithm, the drive will always try to recover from an error by attempting to re-read the data correctly. This strategy prevents invoking correction on non-repeatable errors. Each time a sector in error is re-read a set of ECC syndromes is computed. If all of the syndrome values equal zero, the data was read with no errors, and the sector is transferred to the host system. If any of the syndrome values do not equal zero, an error has occurred, the syndrome values are retained, and another re-read is invoked.

Note: Non-repeatable errors are usually related to the signal to noise ratio of the system. They are not due to media defects.

When the sets of syndromes from two consecutive re-reads are the same, a stable syndrome has been achieved. This event may be significant depending on whether the automatic read reallocation or early correction features have been enabled. If the early correction feature has been enabled and a stable syndrome has been achieved, triple-burst ECC correction is applied, and the appropriate message is transferred to the host system (e.g., corrected data, etc.).

Note: These features can be enabled or disabled through the ATA Set Configuration command. The EEC bit enables early ECC triple-burst correction if a stable syndrome has been achieved before all of the re-reads have been exhausted. The ARR bit enables the automatic reallocation of defective sectors.

If the automatic read reallocation feature is enabled, the drive, when encountering triple-burst errors, will attempt to re-read up to 8 times the retry count set in the AT Configuration bytes.

Note: The Quantum Fireball TM series of drives are shipped from the factory with the automatic read reallocation feature enabled so that any new defective sectors can be easily and automatically reallocated for the average AT end user.

5.5.4 Defect Management

The Quantum Fireball TM 1.0/1.2/1.7/2.1/2.5/3.2/3.8AT drives allocate two sectors per cylinder as spares. In the factory, the media is scanned for defects. If a sector on a cylinder is found to be defective, the address of the sector is added to the drive's defect list. Sectors located physically subsequent to the defective sector are assigned logical block addresses such that a sequential ordering of logical blocks is maintained. This inline sparing technique is employed in an attempt to eliminate slow data transfer that would result from a single defective sector on a cylinder.

If more than two sectors are found defective on a cylinder, the above inline sparing technique is applied to the first two sectors only. The remaining defective sectors are replaced with the nearest available spare sectors on nearby cylinders. Such an assignment of additional replacement sectors from nearby sectors, rather than having a central pool of spare sectors is an attempt to minimize the motion of the actuator and head that otherwise would be needed to find a replacement sector. The result is minimal reduction of data throughput.

Defects that occur in the field are known as *grown* defects. If such a defective sector is found in the field, the sector is reallocated according to the same algorithm used at the factory for those sectors that are found defective *after* the first defective sector on a cylinder; that is, inline sparing is not performed on these grown defects. Instead, the sector is reallocated to an available spare sector on a nearby cylinder.

Sectors are considered to contain grown defects if the triple-burst ECC algorithm must be applied to recover the data. If this algorithm is successful, the corrected data is stored in the newly allocated sector. If the algorithm is not successful, the erroneous data is stored in the newly allocated sector, and a flag is set in the data ID field that causes the drive to report an ECC error each time the sector is read. This condition remains until the sector is rewritten.