SpinRite

“What's Under The Hood”

by

Steve Gibson

What the SpinRite Owner's Guide Doesn't Say

Rather than loading the owner's guide down with lots of “this is what you just bought and let us tell you how wonderful it is” propaganda, we fought to keep it concise, non- intimidating, and solution directed. Many of SpinRite 3.1's initial users have reported that they love the guide's brevity. But it explains nothing about how and what SpinRite does. So we created this “under the hood” technical brief to explain SpinRite . . . what it is, what it does, and how it works.

A Word About Patents and SpinRite

We are, like many software developers, philosophically opposed to the U.S. patent system's too-liberal granting of software intellectual property rights.

We feel that what is being called an “invention” is often little more than clever engineering, and as a result innovation is being stifled rather than promoted. However, as a small software publisher in a land of software giants, we would be foolish to give up the same protection for our software that everyone else is attaining. In a climate which condones (and increasingly expects) “reverse engineering,” securing patent protection could easily separate the survivors from the dead benefactors. Since many of the new and completely unique technologies that were developed for SpinRite 3 easily qualify as “inventions” within the current definition, we are actively pursuing all means of securing rights in the intellectual property we have created. Much of what is disclosed within these pages is the subject of extensive ongoing patent acquisition.

Content Summary

Appendix

A totally new SpinRite

SpinRite II was aging in an industry that takes no prisoners. There had been so much evolution since those days of mis-interleaved, 17-sector, MFM drives, that we knew it was time for some serious change. The tremendous success of IDE hard disk drives with their standardized Western Digital AT-compatible hardware, meant that for the first time we could interface ***directly*** with the controller and drive hardware to completely circumvent the limitations of the system's motherboard BIOS. SpinRite's users had long been asking for features that could not be readily shoehorned into the framework of SpinRite II, and I had been fiddling around with a number of intriguing ideas relating to recovering data from completely unreadable sectors.

The stage was set for a whole new SpinRite

So we made the hard decision to take everything we'd learned from SpinRite II, but scrap its code and start anew with three goals: We wanted to incorporate years of field experience and user suggestions into an entirely new product. We wanted to develop and package significantly more powerful technology for data recovery, drive analysis, and long-term mass storage maintenance. (Adding IDE and diskette compatibility in the process.) And we wanted to make SpinRite 3 even easier to use than SpinRite II had been. If you've been familiar with prior versions of SpinRite, you'll immediately recognize that SpinRite 3 is a complete rewrite of the original.

In case you don't already know . . . What is SpinRite?

SpinRite is a stand-alone DOS application that specializes in the recovery of marginally or completely unreadable hard and floppy disk data, and in the lifetime maintenance of PC mass storage devices. It earned its stripes many years ago by introducing the concept of non-destructive low-level reformatting and sector interleave optimization. Since then its capabilities have continued to broaden until it has become the premiere tool for disk data recovery and magnetic mass storage drive maintenance. Written in assembly language, SpinRite still performs as well on a clunky old 4.77 megahertz PC/XT as on a screaming 333 megahertz Pentium II.

Major New Features

• A full super-set of prior SpinRite functions

SpinRite's interleave optimization process is bypassed when a drive is already capable of transferring one track of data in a single revolution (because we can't improve upon that), otherwise SpinRite will determine the optimal sector interleave for the system's various hard disks and will reset their interleave to achieve maximum performance. SpinRite 3.1 includes all of the prior version non-destructive low-level reformatting functionality.

• BIOS bypassing, hardware level interaction with standard hard disk systems

SpinRite completely bypasses the system's motherboard BIOS software when used on any standard hard disk system. By interacting directly with the system's hardware, SpinRite leverages all of the system's extended hardware capabilities such as data caching, extended error recovery, and internal defect management, to provide enhanced performance, and to query the drive's hardware for detailed behavior diagnosis.

• Dramatically extended data recovery capabilities

SpinRite's new DynaStat data recovery technology has proven surprisingly effective at recovering unreadable data wherever it occurs on any drive, in any sector, in any file, anywhere within any DOS hard or floppy disk partition. The DynaStat system's statistical analysis capability frequently determines a sector's correct data even when the data could never be read correctly from the mass storage medium.

- **“Flux Synthesis” surface analysis breakthrough**

When mass storage systems used simple single density (FM) and double-density (MFM) data encoding, worthwhile surface analysis test patterns were universally known. However the proliferation of proprietary RLL encoding used in contemporary drives has rendered these well-known test patterns useless. SpinRite's new “flux synthesis” technology incorporates functional models of every existing proprietary RLL encoder to generate coherent families of surface analysis test data, on demand, which are specific to the individual drive under test. This allows far more sensitive defect analysis testing to be achieved in one-fourth the time.

- **Performance benchmarking milestones**

Traditional mass storage performance benchmarks attempt to characterize drives by measuring their data throughput and average seek times. These measurements miss the mark because they fail to incorporate an awareness of local drive caching, intrinsic sector translation, and the significant real performance impact of variable sector counts and cylinder data density. SpinRite incorporates a whole new measure of drive performance, Sector Access Velocity (SAV), that compensates for these factors by measuring the drive's “seek to the data” performance which is a measurement in megabytes per second. The resulting SAV value closely tracks the drive's perceived performance.

- **Completely reworked user-interface with many convenience features**

In an effort to make SpinRite 3.1 even easier to use than SpinRite II was, and to support the many new features of 3.1, the user-interface and the fundamental operating approach have been completely redesigned. SpinRite's user-option settings are now “sticky,” and its drive “fingerprinting” technology retains SpinRite's detailed analysis of drive's physical characteristics to allow SpinRite to be used without any delay. A true multithreaded user interface allows SpinRite to operate in the background while the user browses among seven information screens in the foreground.

- **Compatibility with compressed partitions**

SpinRite 3.1 operates transparently with DoubleSpace, Stacker, and SuperStor partition compression, and is “aware” and compatible with all other known compressors.

- **Operation on floppy diskettes**

If floppy diskettes weren't so universally compatible and inexpensive, we would never tolerate their low degree of reliability. But they are cheap and easy, so we do. SpinRite 3.1 addresses this problem by deliberately extending its operation to include diskettes. SpinRite's new DynaStat technology has proven amazingly effective on floppy disks.

Data Recovery

• How can SpinRite read “unreadable” data?

Although it's comforting to think of disks as “digital” and stuffed with perfect little 0's and 1's, drives are much more “analog” than we'd like to believe. We've all had the experience of having a floppy disk hit a spot it can't read until we open the door, reinsert the diskette, and press “Retry” a few times. In doing so, we're hoping that we can “get it past” the problem area. Quite often we can. Similarly, hard disk drives don't quit and declare a sector to be in error until the drive has tried to read the sector many times. If a sector didn't read, perhaps it will the next time around. Since it often will, it's clear that magnetic mass storage differs a great deal from RAM memory. It is these differences that SpinRite exploits.

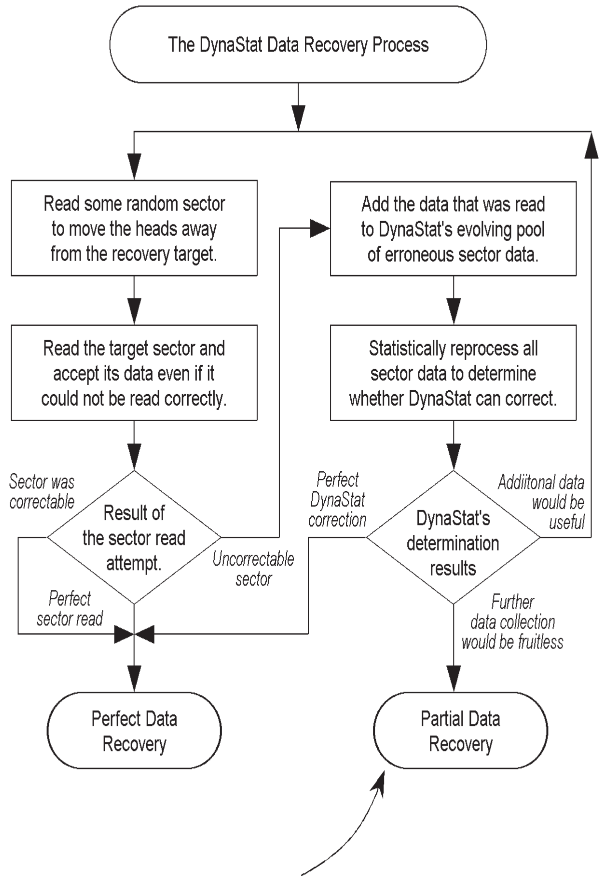
Even when a sector won't ***ever*** read correctly, there's still hope. The data being read from a marginally readable sector changes from one reading to the next, and useful, if not correct, information is contained within each of these differing readings. We have found that a careful statistical analysis of the results of multiple incorrect readings can be used to pinpoint a sector's trouble, and to reconstruct the original information the drive has been “trying” to read. This is the key behind SpinRite's DynaStat data recovery system.

• DynaStat Data Recovery

At the first sign of trouble reading from a sector, whether or not we're ultimately going to get a perfect reading from it, the DynaStat system kicks in. It begins analyzing the nature and extent of the problem, collecting every bit of information possible. DynaStat's recovery methodology incorporates several complementary strategies: The first is simply extensive retries. As we've seen, just trying harder often results in just one good read . . . which is all we need. The recovered data won't then be returned to the same sector, after we've retrieved it, unless we verify that it's truly a safe place to restore the data.

During this exhaustive rereading, DynaStat employs its second recovery strategy of deliberately wiggling the drive's heads. By successively approaching the troubled sector from different distances and directions, the heads arrive at the sector's track at different velocities, which in turn produce small but significant displacements in the head's resting position. This allows DynaStat to compensate for the long-term alignment drift that occurs in non-servo based drives, and the positioner hysterysis that occurs in servo-based designs. Thus the drive's heads are given every opportunity to land in the best possible location to correctly read the sector. This approach is also extremely effective at recovering data from misaligned diskettes – which SpinRite 3.1 is proving to be extremely effective upon.

DynaStat's exhaustive, head-wiggling re-reading is almost always able to coerce one good or correctable read from a recalcitrant sector. But when the sector just will not read, DynaStat's third, core, recovery strategy is brought into play: The mass of data collected during its many re-reading attempts is statistically analyzed in an attempt to calculate the sector's original contents. At the very least, the amount of data lost is significantly minimized by this process, and more often than not the sector's data is correctly calculated and completely restored.



• What good is partial data recovery?

Contrary to casual belief, recovering only most of the data from a sector can be a ***tremendous*** benefit for data recovery. SpinRite is able to at least recover most of a sector's data even in the worst situations. For example, if that sector were a chunk of a partition's file allocation table, a few lost bytes would probably damage the structure of just one file, but losing the entire sector would confuse 256 clusters and all of the files containing them. If a sector of the root directory or any sub-directory were completely lost, all of the directory's files and sub-directories would be lost, but if the loss were contained within just a few bytes, one directory entry would be hurt, but everything else in the directory and its sub-directories would be saved.

• What about the user's files?

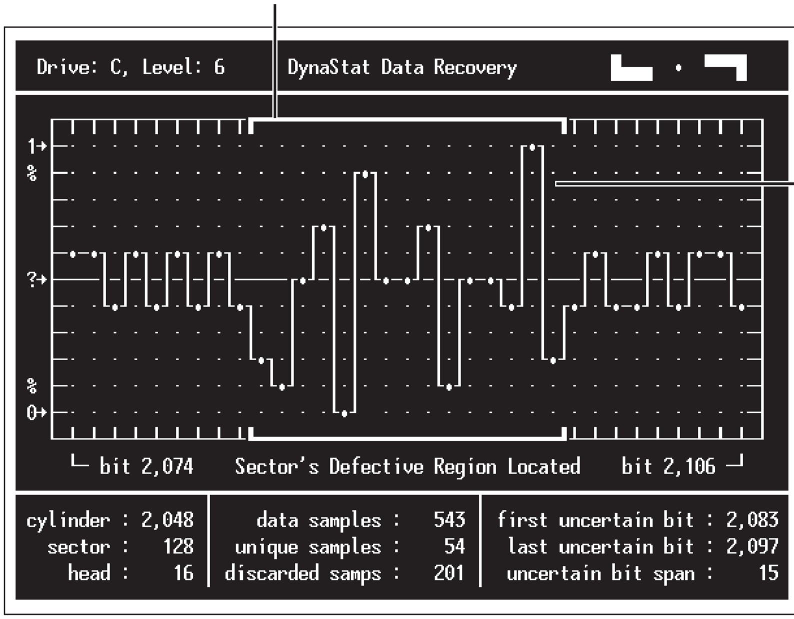
Partial data recovery is even useful when the sector is not part of the DOS file system. For example, recovering only a portion of an error in a large database often allows the balance of the database's data to be recovered rather than rendering the entire file useless since the database application would then be able to read past the error. Since DynaStat so drastically minimizes a sector's data loss, rather than simply "dropping" a damaged sector from further consideration, it's even possible for ***executable*** files to be used with care. SpinRite's users have reported that most functions of non-compressed executables can still be used after partial SpinRite recovery. Although such use is never preferred, and SpinRite provides explicit, ample warnings about using altered executable files, there are times when an executable file is completely irreplaceable and accepting some alteration is preferable to losing the file's entire functionality. In any case SpinRite will recover everything it can from the drive and advises its user about what it's achieved.

DynaStat takes advantage of the fact that hard disk drives transfer data even when the sector's data is known to be erroneous. By accepting this data anyway, and analyzing it carefully, SpinRite's DynaStat technology can do a far better job of recovering lost data.

When DynaStat locates the defective sector's region of data uncertainty, the "calipers"appear to graphically deliniate the uncertain region.

The current DOS drive device and SpinRite operating level are shown at all times. SpinRite can change the operating level itself when a region requires deeper analysis.

The "Spinner" operates while SpinRite's multi-threading user-interface is in use.



This scale depicts the percentage of time individual data bits have been read as a "1"and "0." Since the region outside the calipers would always be 100% 1's or 0's, its display is clamped to highlight the central area.

The region inside the calipers contains the first and last bits which have changed during DynaStat's successive re-reading of the sector's data. The size and location of this region is subject to change during DynaStat's operation as additional sector data is accumulated.

An "incremental statistical" algorithm processes each new sector of data to produce the analysis used by the preceeding flow chart.

During DynaStat's operation the damaged sector's logical cylinder, head, and sector address is continually shown.

DynaStat's incremental statistical analysis can decide that some sector data read from the drive does not contribute to a meaningful reconstitution of the target sector's data. Such sectors are discarded to make additional space available for meaningful data.

Sectors can be damaged in a way that causes the drive's flux-reversal data decoders to have difficulty "relocking" after passing the sector's damaged region. Since DynaStat's calipers cannot show large uncertain regions, the first and last uncertain bits, and the uncertain bit span, are shown below.

After coercing all possible data from a drive, SpinRite then determines whether the drive's storage surfaces underneath the recovered data are capable of safely storing and retrieving whatever data the system might choose to place there. We need verify the drive's fundamental storage integrity. It's time to test the surfaces.

Surface Analysis

• Where have all the defects gone?

Hard drive manufacturers have never been able to produce totally defect-free magnetic storage platters. Variations in surface terrain, coating thickness, and material composition create minuscule variations in the surface's magnetic properties that affect stored data. In the days of MFM and RLL drives, the separation of the drive from the controller forced drives to publicly “confess” and actively publish their known surface defects. But today's tight drive/controller integration in modern IDE and SCSI drives allows these surface defects to be completely masked and hidden. Although users understandably rejoice in the illusion of completely “defect free” drives, the surface's ever-present defects are actually being hidden beneath a layer of on-the-fly sector relocation. As a result, effective, periodic, surface defect management is every bit as critical today as ever in the past.

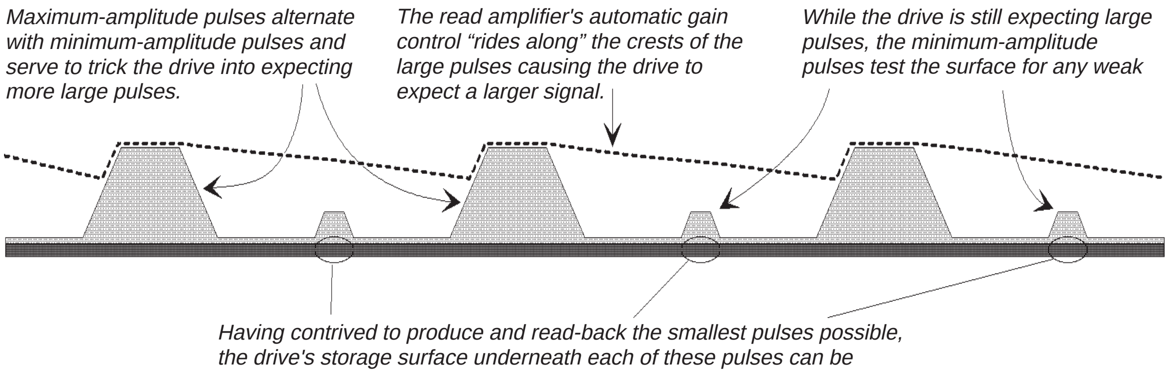
The fact that a drive's manufacturing defects were hidden by the factory should not inspire users with a false sense of the drive's perfection. These drives can die just as surely as drives always have. Popular personal computer publications have described IDE drives as ***disposable***. They've said “... you can't tell when they're dying and you can't fix them after they have.” This misconception is understandable since IDE drives are indeed ***different*** from their predecessors, and their trouble signs and requirements have changed, but properly re-engineered analysis and maintenance software tools can provide modern IDE and SCSI drive owners with the same degree of early warning, loss prevention, and data recovery capabilities as prior drive technologies enjoyed. In fact, SpinRite 3.1 empowers owners of these newer technology drives and older drives with significantly greater data recovery and prevention capabilities than has ever been possible.

• The search for “weak bit” spots.

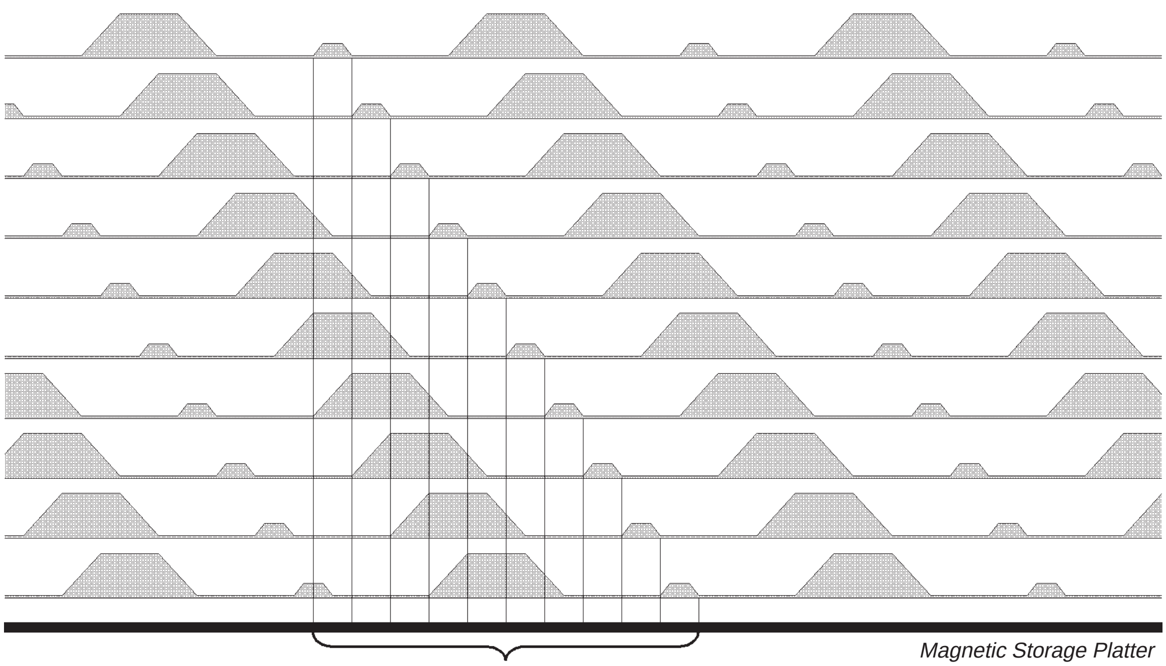
Since magnetic mass storage devices are not completely defect free, the best aid for the long-term maintenance of reliable data storage is the early detection and elimination of inevitable surface defects. These defects, which are caused by surface scratches, abrasions, pits, or thin magnetic material plating, reduce the strength of the recorded signal when it is being read back. Defects have also been shown to develop or “grow” due to a gradual evolution of the drive's storage surfaces. To achieve the highest possible storage reliability, any locations that can be shown to affect the integrity of recorded data should be immediately removed from the operating system's use.

The strategy used by SpinRite 3.1 to detect these regions is currently unique in the industry: A special data sequence is custom-designed and recorded onto the drive, then carefully read back with the drive's internal “error correction” protocols momentarily held in check. The specially crafted data sequence plays a fundamental role in the detection of weak spots by sliding a signal that alternates between maximum and minimum amplitude along the drive's entire surface.

The maximum-amplitude portion of the signal tricks the drive into lowering the “gain” of its read amplifier. Since any signal “clipping” that would result from the amplifier's gain being turned up too high must be avoided at all cost, the drive's AGC (automatic gain control) circuitry quickly responds to any large signal amplitude by lowering the amplifier's gain. This large amplitude signal is immediately followed by a small pulse of the lowest possible strength. Since the amplifier's gain has been cranked down by its encounter with the largest possible pulse, the small signal pulse is made even smaller. If there's anything at all weak or uncertain about the location underneath the tiny pulse, a deliberately detectable read error will result and SpinRite will have found a new defect in the surface!



Since only the locations lying directly underneath the minimum-amplitude pulses are really being tested, a minimum-amplitude pulse must be successively placed over every possible flux-bit location of the drive. In other words, the special alternating large/small amplitude signal must be “slid along” the surface of the drive to search for all possible defects.



The sliding “family” of small pulses thoroughly scour the magnetic storage surface to detect any regions that might be unsafe for data storage.

With this understanding under your belt, SpinRite's Surface Analysis user-interface screen probably makes much more sense:

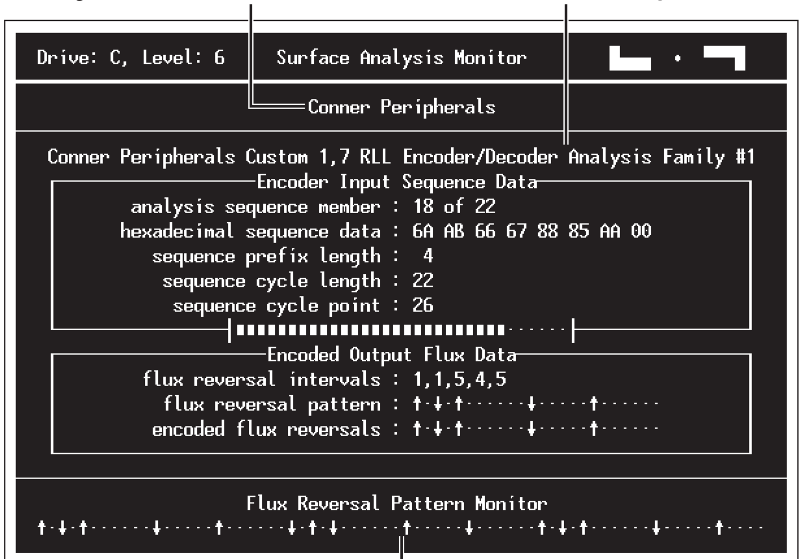
The appendix of this booklet contains additional details about the "FluxSynthesis" technology incorporated into SpinRite. This core technology allows SpinRite to deliberately craft the data sequences shown below for the generation of maximum and minimum amplitude data

The total time required for SpinRite's surface analysis can be dramatically shortened when the drive's manufacturer is recognized since specific knowledge of the drive's flux-encoder can be utilized.

As explained further in the appendix, accurate flux-synthesis depends upon knowledge of the relationship between the user's data and the drive's generated flux-reversals. SpinRite contains internal models describing every flux encoder in use today.

The screen's "Encoder Input Sequence Data" monitor shows the total number of input sequences used for the specific encoder, the current hexadecimal sequence data being applied to the encoder, the sequence's non-repeated bit-length (used to phase align the flux-pattern), and the total overall sequence bit- length.

This specification completely governs the stream of data flowing to the target sector.



The screen's "Encoded Output Flux Data" monitor shows the flux reversal patterns generated by the drive's flux encoder when presented with the Input Sequence Data.

It shows the inter-flux reversal intervals, the resultant non-phase shifted pattern of flux reversals, and the actual flux reversal pattern after SpinRite's deliberate phase-alignment.

The "Flux Reversal Pattern Monitor" shows the series of phase-shifting flux-reversal test sequences being recorded on the drive as it's happening. As explained further in the appendix, the isolated flux reversals result in two maximum-amplitude pulses, then the closely-spaced flux-reversal "triplet" squelches the amplitude of the pulse in the middle, causing it to be read back at the minimum possible strength.

The "Flux Synthesis" appendix thoroughly explains the relationship between large and small pulses and the patterns of flux reversals shown on the screen above.

Performance Benchmarking

Although the subject of storage sub-system performance benchmarking admittedly lies outside the realm of data recovery and long-term drive maintenance, SpinRite has traditionally incorporated an accurate and usable assessment of storage system performance. Maintaining the useful accuracy of this benchmark in the face of the rapid evolution of modern drive technologies required a complete rethinking of traditional benchmarking approaches. SpinRite 3.1 now incorporates a ground-breaking new measurement of drive “access” performance as well as separate measurements for “from-the-buffer” and “from-the-media” transfer rates.

- **Sector Access Velocity (SAV)**

Perhaps the biggest news is SpinRite's accurate new assessment of drive access (or data seeking) performance known as “Sector Access Velocity” or SAV. Traditional seek measurements have attempted to characterize a drive's “seeking” performance by measuring the average time required to reposition the drive's read/write heads over any of the drive's physical data tracks. For reasons explained below, the usefulness of these measurements were always of dubious value, but they were all we had at the time. Unfortunately, some drive manufacturers responded to the industry's affection for this measurement by “cheating” the benchmarks and either “short circuiting” seek operations altogether (simply ignoring seek requests) or returning a premature “drive seek complete” status. Both cheating methods return artificially low seek performance figures and so fail to accurately characterize a drive's real-world performance. This raw seeking measurement also fails to compensate drives with high spindle speeds since it fails to consider the reduced rotational latencies of higher performance drives.

Even when a true measure of a drive's average seek performance was available, its real value was still unclear since it is the drive's cylinder density that determines how far a drive's heads must move. For example, a drive with two heads and only 34 sectors per track contains 68 sectors per cylinder and so must move its heads through 74 cylinders in order to seek across 5,000 data sectors. But a drive with 96 sectors per track and 8 physical heads stores 768 sectors on each cylinder and so only needs to move through 7 cylinders when seeking across the same 5,000 sectors. In other words, the second drive with a higher “cylinder density” needed to move its heads less than 10% as far as the first drive with a lower density.

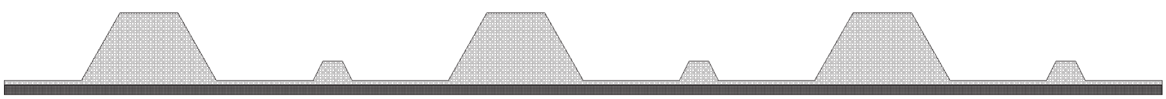
What's needed is a universal, foolproof, uncheatable measurement of a drive's data access performance that automatically takes the drive's data storage densities into consideration. SpinRite incorporates just such a test by reporting the rate of the drive's head motion across the drive in units of megabytes per second. By thinking in terms of the amount of data being crossed over when seeking to a randomly chosen sector, SpinRite's Sector Access Velocity, or SAV, provides performance numbers that accurately track the perceived performance of the measured drive and allow any drives to be compared fairly.

– Appendix –
Inside SpinRite's Flux Synthesizer

The following material is provided to satisfy any shreds of curiosity which might have somehow survived the foregoing discussions. Although an understanding of these principles was required for SpinRite's development, the preceding discussions should have already provided ample background for understanding SpinRite's operation and benefits.

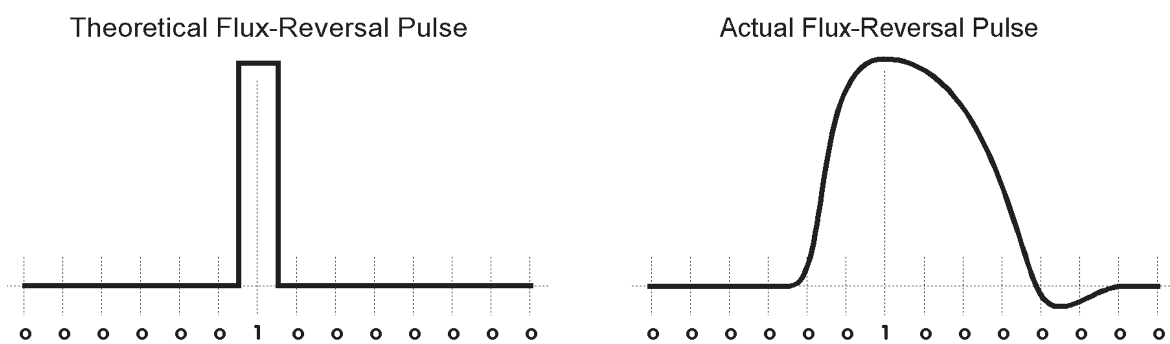
• Generating Large and Small flux reversals

As you'll remember from this diagram . . .



. . . the key to accurate defect detection lies in somehow managing to generate a series of flux reversals of alternating strength. The maximum strength flux reversals trick the drive's internal automatic gain control into expecting a large signal, and the small flux reversals provides a means for detecting any diminished capacity on the storage surface underneath. So the thousand dollar question is: How can a software-only product like SpinRite possibly ***control*** the recorded strength of a drive's flux reversals?

The first part of the secret lies in the fact that flux reversals are not clean and perfect little pulses, they're actually big ugly lumps that ***interact*** with their neighboring flux reversals in complex ways:



As you can see from the diagram above, whereas a theoretically perfect flux reversal pulse would be exactly one pulse-period wide, the pulse generated by an actual flux reversal has a much slower “attack” and “decay” time, and even dips a bit below the original signal level with behavior known as “*overshoot*.”

So the trick to the deliberate manipulation of flux reversal pulse amplitudes is to understand how these large actual flux reversals combine into a resultant signal. Here's the trick in a nutshell:

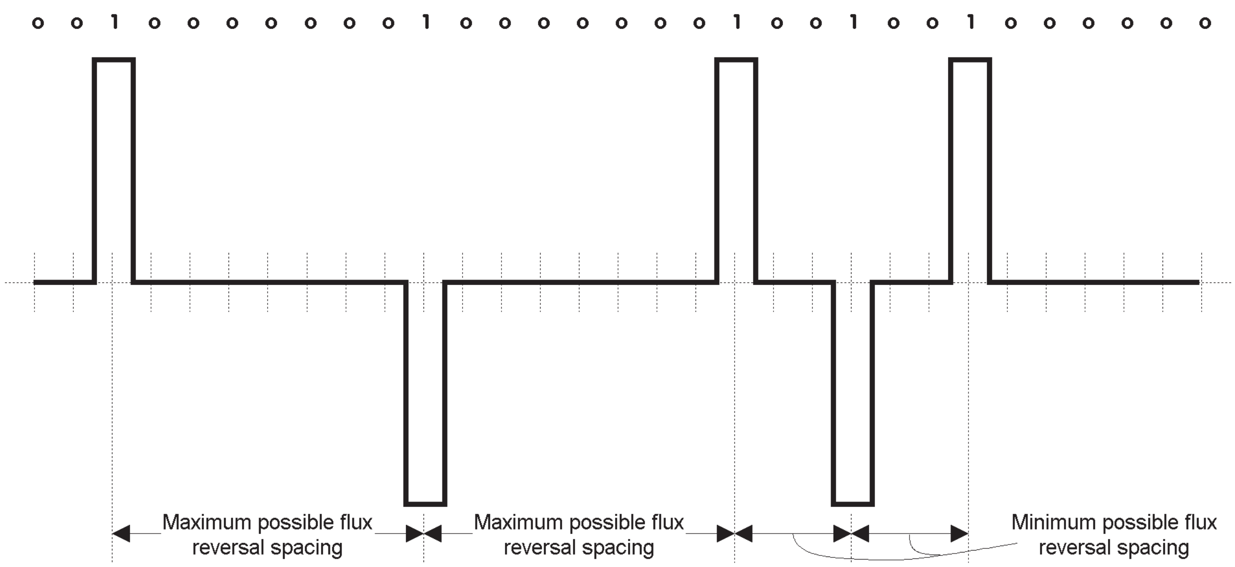
Rules for Combining Flux-Reversals

1. A single, isolated, flux reversal generates the greatest possible strength signal, ... and ...

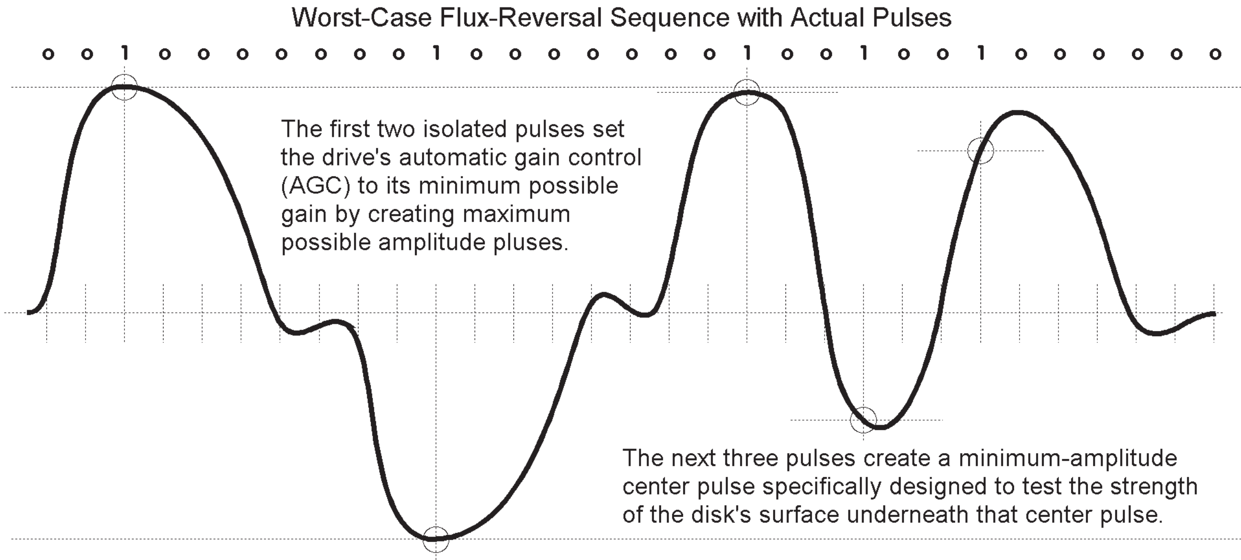
2. A “triplet” of three flux-reversals occurring as close to each other as possible, generate a minimum-strength flux reversal in the center.

These rules dictate that two isolated flux reversal pulses, each separated from its neighbors (and with each pulse going in opposite directions) would generate a maximum strength signal, followed by three flux reversal pulses occurring as close together as theoretically possible to generate a minimum-strength central pulse. A theoretical depiction of this looks like:

Worst-Case Flux-Reversal Sequence with Theoretically Perfect Pulses



However, the actual signals generated by real flux reversals interact with each other to produce a very different result:



As you can see, the full-strength amplitude of the two isolated flux reversals is preserved, however the three closely-spaced pulses interact highly to produce a greatly reduced strength center pulse. This is the key to the first part of SpinRite's super-sensitive magnetic media surface analysis technology.

• Placing flux reversals wherever we want them

We've seen that SpinRite must be able to accurately cause a drive to lay down specific flux reversal patterns wherever it requires, and to "slide" these sequences of pulses along the drive's surface in order to test every possible "bit storage cell" on the drive's surface. In order for SpinRite to specify the flux reversal sequences it desires, rather than merely the data it wishes to record, it must understand the relationship between the data and the flux reversals for the drive being tested. This understanding is used to "reverse engineer" the data from the flux reversal sequences.

For this to really make sense, let's digress for a moment into a brief examination of the way data and flux reversals are related:

• Magnetic Data Recording (a brief tutorial)

A physical property of nature known as ***electromagnetic induction*** bridges the gap between electricity and magnetism. A flowing electric current generates a companion magnetic field, and a changing magnetic field in the presence of an electrical conductor generates an electric current. When data is being read from a disk, a weak electric current is induced to flow through a drive's read/write head whenever there's a direction change in the magnetic field "***flux***" underneath the head. Since changes in the magnetic field's "***flux***" are what is sensed by the read head, "***flux reversal events***" are what is recorded to store data on the drive. This means that a computer system's information must be converted into a pattern of flux reversal events in order for it to be recorded. This data encoding process serves to ensure that the storage system will remain “locked onto” the flux reversal information when it is read back from the drive.

The earliest encoding technology used for this purpose was ***Frequency Modulation*** recording, or FM:

Frequency Modulation Encoding

Input Data:	0 1 1 0 0 1 1 1 0	5 flux reversals
Clock Bits:	1 1 1 1 1 1 1 1 1	9 flux reversals
Encoded Data:	1 0 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 0	14 flux reversals

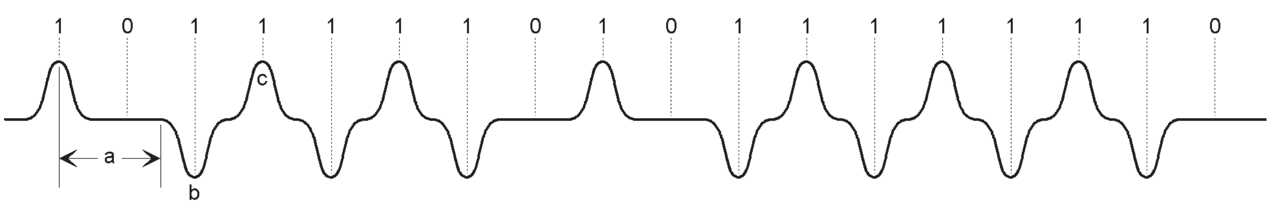
FM Encoding Table

Input Data	Encoded Flux Reversals
0	1 0
1	1 1

Notice in the preceding diagram, that a “clock” flux reversal is placed in front of every “data bit” flux reversal. When reading the data back from the drive, the drive's “data separator” or “flux decoder” uses the interspersed clock pulses to locate the one and zero data bits and to strip the clock information – which is useful only while the data is stored on the magnetic media – back out of the data. Since the clock and data flux reversals are completely mixed together, the drive's data separator needs some way to determine which are the clock bit reversals and which are the data bit reversals.

Decoding an FM-encoded flux-reversal waveform

Flux reversals appear as an alternating polarity signal containing both data and clocking information. The data separator's job is to determine which pulses represent data ans which represent the extra clocking information



When an interval is encountered during which no pulse occurs (see “a” above) the data separator “knows” that the next pulse must be a clock pulse (“b”). With the known clock pulse as a reference, the data pulses (“c”) can then be easily picked out and retained while the clock pulses are ignored.

• What's wrong with this simple FM recording?

The physics of electromagnetic induction places a finite lower bound on the minimum spacing between successive flux reversals. In other words, they need some space in-between. This inter-flux spacing requirement limits the number of flux reversals that will fit onto one track of a disk. Thus, from the standpoint of cramming as much data as we possibly can onto a drive, flux reversals form our precious and limited resource. Frequency Modulation recording was convenient, simple, and very reliable on early PC's, but so many flux reversals were spent on clock bits compared to data that FM turned out to be an inefficient means for storing data. So let's see about modifying our approach . . .

• Modified Frequency Modulation ... MFM recording

If we're clever with our design of a magnetic flux encoding method, it's possible to completely eliminate the requirement of data clock bits. By eliminating all of the clock bits from the frequency modulation approach, ***Modified*** Frequency Modulation (MFM), also known as ***double-density*** recording, doubles our efficiency of flux reversal usage, and thus doubles the amount of data we can store around a single track of a disk. The Encoded Data line in the chart on page 13 showed that FM encoding consumed 14 flux reversals in order to encode the “011001110” data pattern. By comparison, a better scheme known as MFM encoding requires 7:

Modified Frequency Modulation Encoding

Input Data:	0 1 1 0 0 1 1 1 0	
Encoded Data:	0 1 0 0 1 0 0 1 0 1 0 0 1 0 1 0 0 1	7 flux reversals

MFM Encoding Table

Input Data	Encoded Flux Reversals
0	0 1
1	x 0

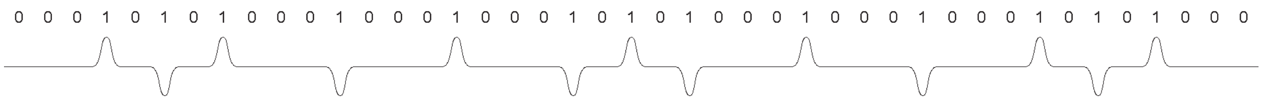
Note: the "x" on the chart indicates using the opposite of the immediately preceeding encoded output bit.

MFM allows the same number of flux reversals to represent twice the encoded data ... that's why we call it “Double Density” recording.

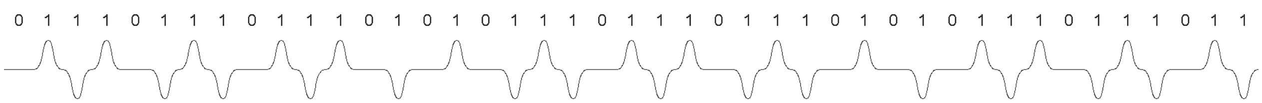
As it turns out, FM and MFM encoding are just two members of a mathematically infinite family of possible data-to-flux reversal encoding schemes. The next step takes us into the domain of ***RLL encoding*** where the traditional fixed FM and MFM encoding/decoding rules no longer apply. IBM has their own (patented) RLL encoding scheme which they license to those willing to pay. Conner Peripherals uses their own design, as does Maxtor, Quantum and Seagate. Because the use of an RLL encoding scheme delivers another 50% economy in flux reversals, allowing a corresponding 50% increase in data storage, everyone has developed or licensed a means for providing it.

• Okay, now to the point of all this!

Since the drive's encoder/decoder determines the relationship between the user's data and the drive's magnetic flux reversals, we must know ***WHICH ENCODER/DECODER*** is being used by the drive in order to determine what data to send to the drive for surface analysis. For example, one data sequence for analyzing the surfaces of an MFM drive is a repeating hexadecimal `51h`. This has a bit sequence of "`0101 0001`." When this sequence is encoded into flux reversals by an MFM encoder, the result will be:



This flux reversal sequence exhibits the characteristic of isolated flux reversals alternating with flux triplets which, as we've seen, is ideal for surface defect detection. However, the same `51h` data, encoded by an FM encoder, yields the following flux reversal pattern:



In contrast to the first sequence of flux reversals, this FM translation of the same 51h data is a mess of uncoordinated and interacting flux reversals. It would result in an overall low-amplitude signal (since all neighbors would be damping each other) which the drive's automatic gain control would adapt to by turning up the drive's read amplifier gain. The result would be that the weak signals over serious defective areas would be amplified also, and therefore completely missed.

• The final piece of the puzzle

Back in the days when recording technology was based upon simple FM or MFM encoding, simple and well known "worst-case data patterns" could be used to search a drive for surface defects. But as we've seen before, that approach will no longer function ***at all*** in a world with a vast array of different RLL data-to-flux reversal encoder/decoders. Surface analysis data which would produce a useful flux reversal pattern through one encoder will be completely useless when passed through another.

SpinRite 3.1 contains mathematical description simulation models for every flux encoder being employed in hard disk drives today (and even for some which are still in the labs getting ready for tomorrow!). After identifying the drive's manufacturer and model number, SpinRite utilizes the corresponding encoder's mathematical model to derive a family of test data which is specifically tailored to produce these optimal flux reversal sequences for testing the surface of the drive. Each successive sequence of test data results in single-bit shifted flux reversal phasing which thoroughly scrubs the entire surface of the drive.

Some moderately sophisticated artificial-intelligence technology was used to recursively goal-seek the optimum sequence of flux reversals, within the constraints and limitations of the drive's data-to-flux encoder, then work back up through the drive's decoder model to deduce the original input data which will produce these optimal surface analysis flux reversal patterns.

The resulting customized sequences of data patterns are specifically designed for each drive and technology. SpinRite 3.1 thereby performs a far better job of surface analysis in a shorter time than has ever been possible before. All this technology works, and has been incorporated into – and buried inside of – SpinRite 3.1. Although the results are all that really matters, and they speak well for themselves, we wanted to satisfy any curiosity you might have had.

Thanks ***very much*** for your interest in

what's under the hood of SpinRite!