

A Method for Recovering Data From Failing Floppy Disks with a Practical Example

Dr. Frederick B. Cohen, Ph.D. and Charles M. Preston

Fred Cohen & Associates and California Sciences Institute

Abstract: As floppy disks and other similar media age, they have a tendency to lose data because of the reduction in retention of electromagnetic fields over time associated with various environmental degradation factors. In attempting to read these disks, the coding techniques used to write them can be exploited along with the proposed fault mechanisms to demonstrate that repetitions of read attempts under proper conditions yield valid data subject to specific error modes. In many cases those errors can be partially or completely reversed by analysis of read results. A practical example involving a case of substantial legal value is used as an example of the application of these methods.

Keywords: weak bits, floppy disks, digital forensics, coding, field density loss, the birthday problem, error inversion

1. Background

This paper is about a method for recovering data from floppy disks that are failing due to “weak bits”. It describes a repetitive read technique that has successfully recovered data from failing floppy disks in forensic cases and describes analysis of the results of such repetitive reads in terms of yielding forensically sound data values. These techniques are not new or particularly unique; however, they are not widely published and analysis necessary to support their use in legal matters has not been found elsewhere.

The particular matter used as an example in this analysis involved a floppy disk that was more than 15 years old and that contained the only copy of a binary assembled version of a software program that was subject to intellectual property claims of sufficient value to warrant recovery beyond the means normally used by commercial recovery firms. After attempts to read the disk by such firms had failed, the disk was placed in the custody of the authors to attempt to use more rigorous and possibly destructive methods, subject to court approval.

A number of widely used techniques for reading hard-to-read floppy disks are in common use, including reading only relevant sectors from a disk where other sectors fail to read properly, and altering drive alignment so as to better align the heads with the tracks as originally written. In the example case the key content was contained in the hard-to-read sectors of the disk and custom head alignment only marginally altered the recovery characteristics on the disk in question. A floppy drive can also be modified or constructed to read analog signals and allow the thresholds of detection to be altered, the signals from the read heads can be additionally amplified, rates of rotation can be increased to increase induced currents, and other similar methods can also be attempted, however these introduce other problems, cost a fair amount more, consume more time, and are more complex to properly explain in terms of assurance that they produce valid data and don't end up turning noise into detected signal.

Other techniques include analog reads with digital storage scopes, use of epoxies with suspended fine ferris material that attach to the media and are visible under a microscope, the use of magnetic force scanning tunneling microscopy, and other similar exotic methods, some of which are destructive, and all of which are more expensive or risky to the data because of handling or processes used.

2. The technique used in this case

Before moving to the exotic methods, another alternative is to try repeated sector by sector reads of the disk, with failed read sectors repeated till valid reads are completed, and assembly of the sectors into a complete image of the disk by appending the data from sectors together to form a single complete image. This technique has several advantages in that (1) it only uses the designed features of the floppy disk drives and thus requires very little in the way of explanation or analysis to be considered credible, (2) it is relatively low cost and takes relatively little time to perform, and (3) it uses the built-in coding analysis methods and phased lock loops of the floppy drive to decode the current changes resulting from orientations of charges in areas on the disk, and thus eliminates all of the difficulties in explaining how to deal with coding errors, side band signals, additional introduced errors, and other similar problems associated with building special purpose hardware for the same function.

The specific program used in this case was executed from a bootable White Glove Linux CD which was kept with the evidence after processing to assure that the process could be precisely repeated if necessary. The shell script code executed in that environment is as follows:

```
for i in `count 0 1439`; do
    dd conv=noerror bs=512 count=1 skip=$i if=/dev/fd0 > noerr/$i.out
done
```

Within the White Glove environment, the “count” command used in this syntax counts from the first value (0) to the second value (1439) by increments of one. For each count value, the “dd” command is then executed with the “noerr” conversion option that specifies that, on error retries are to be attempted an unlimited number of times. The block size is set to 512 (the normal block size for such a floppy disk) and a count of 1 block per execution is used. This is done after skipping the count number of blocks from the beginning of the media, in this case the floppy disk “/dev/fd0”, with the output stored in a file named including the block number, in this case “noerr/[count].out” where [count] is the block number and noerr is the directory used to store all of the blocks. On each read attempt, a file is created, but unless the file read succeeds with a valid checksum, the file is overwritten on the next attempt.

The reason it is beneficial to read a sector at a time is that a single error in a read produces a failure for the whole read. If a single sector takes 20 attempts on average to succeed, than on average, reading 2 sectors would take 400 attempts, and so forth. Since reading less than one whole sector does not produce different hardware execution, this approach minimizes the number of reads and reduces unnecessary wear and tear on the Evidence Disk while still reading repeatedly until a match between the CRC code and the data is attained.

This process was applied to the Evidence Disk and produced different numbers of retry cycles on different sectors. On sectors that read without error consistently, there were no retry cycles. On the previously unreadable sectors, the number of retry cycles required ranged from one to more than 70 with many in the range of 20 to 30. Each sector was stored individually in a file of 512 bytes on a hard disk as it was read, and stored with a filename associated with the sector number as indicated above. For block number ranging from 0 to 1439, the total is 1440 blocks of 512 bytes each, or 737260 bytes of data, the entire readable contents of a 720K floppy disk.

The individual files representing the blocks of the Evidence Disk are then either independently examinable or may be assembled together into a single file representing the entire content of the original floppy disk and mounted using a loopback mounting interface or written to a fresh floppy disk which can then be used to read the data as if it were the original evidence disk. In the specific case used as an example here, the assembly was done using the following program in the same environment described earlier:

```
for i in `count 0 1439`; do dd seek=$i if=noerr/$i.out of=noerrdd.out ; done
```

In this case the blocks are written into the file at the appropriate location in the same way as they were read from the Evidence Disk in the first place. Multiple copies were made of the recovered disk for use by all parties in the matter at hand. Having now read the disk and provided usable forensic duplicates, it remains necessary to show that the method is forensically valid and sound.

3. The nature of 'weak' bits and floppy disk failure modes

Floppy disks tend to degrade in various ways over time and under various environmental conditions such as temperature, humidity, and so forth. In some cases this results in the presence of so-called “weak” bits on the media. Weak bits are degraded electromagnetic orientation alignments or charge densities on the media that are sufficiently degraded from normal so as to reduce voltage and current swings under normal read conditions to levels that are too small to reliably trigger transitions in the hardware detectors that are used to interpret them. This is a normally the result of reduced charge density or misorientation in the electromagnetic media, but can, presumably, also be caused by physical shifting of magnetic materials under temperature changes, the growth of organisms that move the media, friction that rubs off portions of the coatings used to retain charges, or other similar things. The hardware-level coding used in floppy disks is usually Modified Frequency Modulation (MFM)[1]. This uses timed flux density transitions to encode bits. The way floppy drives store data is shown in Figure 1. The magnetic write/read head causes magnetic particles to align in one of two orientations along cylinders around the circumference of concentric circles at different distances from the center of the platter. Because the sizes of these circles differ with radius, the timing of transitions from one orientation to the other that are used to detect changes that are the encoded form of bits also varies with radius and therefore lead-in transitions are required to set up an oscillator to synchronize this change detection.

The mechanisms used to read from floppy disks is described in [2] and applies a read head, amplifier, pulse generator, phased lock loop, demodulator, and additional hardware to produce a controller that is usable by a computer at the bus level. This is shown in Figure 2.

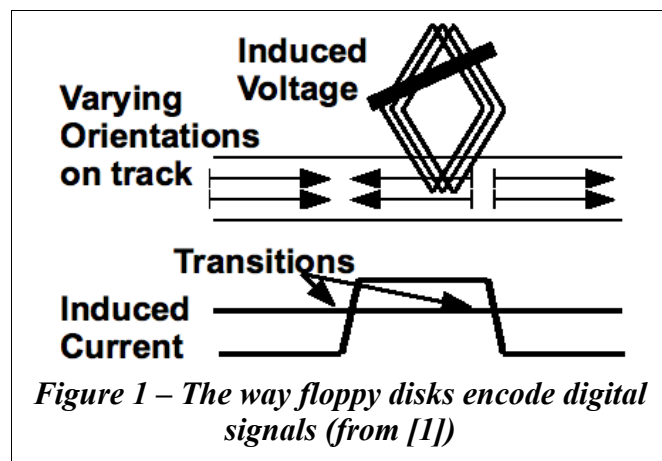


Figure 1 – The way floppy disks encode digital signals (from [1])

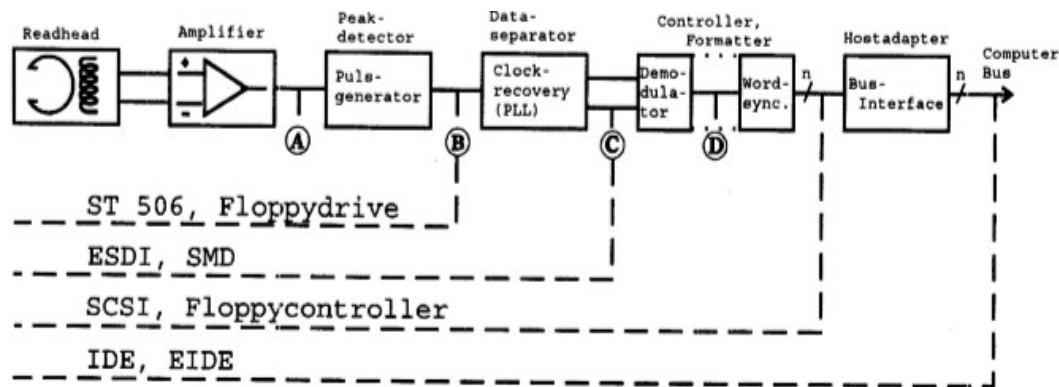


Figure 2 – Floppy electronic components (from [2])

Figure 3 shows the signals that appear at different locations in Figure 2 [2] and helps to provide clarity surrounding the effect of reduced signal levels in the media. As the analog signal degrades (A) peak pulses disappear (B), resulting in loss of MFM transitions and resulting demodulated data (D) or phased lock loop desynchronization.

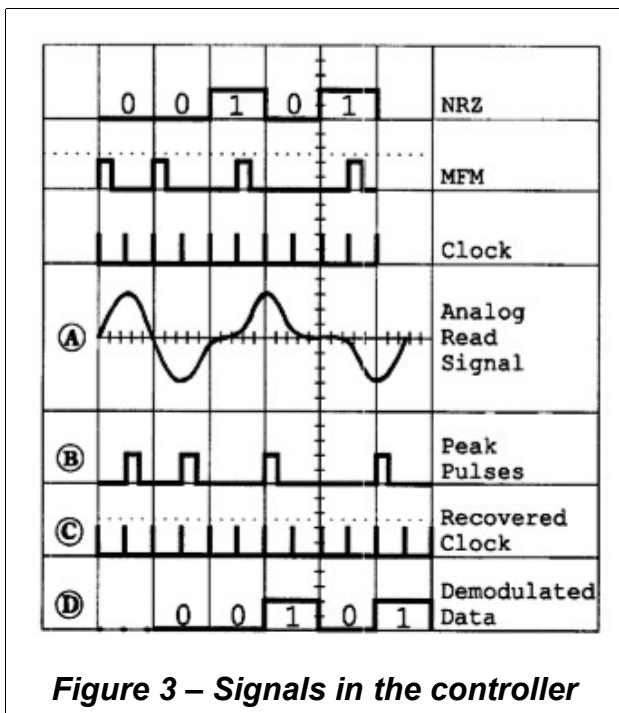


Figure 3 – Signals in the controller

To read from a floppy drive, changes in field density produce induced current in the read head that, regardless of the current field direction, is seen as a transition (T), while the lack of a change at a timing signal produces a non-transition (N). The MFM coding uses a “No transition, Transition” (NT) sequence to indicate a “1”, a TN to indicate a '0' preceded by a '0', and an NN to indicate a '0' preceded by a '1'. If a transition is not detected because of the loss of electromagnetic flux density, it can turn an NT into an NN or a TN into an NN, but it cannot turn an NN into either an NT or a TN. Pairs of bits always involve a transition. In particular, a '11' will produce NTNT, a '00' will produce either a TNTN (if there was another zero preceding it) or a NNTN (if there was a '1' preceding it), a '10' will

always produce an NTNN, and a '01' will produce either a TNNT (if it was preceded by a '0'), or an NNNT (if it was preceded by a '1').

If no transitions are detected, the controller will normally indicate an error condition and a CRC code that is coded at the end of every 512 bit block of data will be irrelevant. So weak bits will either produce controller errors indicative of the inability to observe transitions at all, or weak transitions will result in the change of a T to an N. They cannot turn the lack of a transition into a transition. As a result, 7 out of 11 possible field reductions turn into invalid codings that should be detected by the drive controller as invalid data. Of the remaining 4 errors that could produce valid data, 3 require that the

previous bit be a 1 or they too will produce invalid data in the controller. All the possible changes are shown in Table 1. In this table, data values represented by *T* and *NT* sequences are enclosed in brackets (e.g., [11]) and required preceding bits are indicated prior to the bracketed pairs (e.g., 1[00]) where appropriate.

Data	Originally	Can turn into	Result
[11]	NTNT	NNNT	1[01]
	NTNT	NTNN	[10]
	NTNT	NNNN	invalid
0[00]	TNTN	NNTN	1[00]
	TNTN	TNNN	invalid
	TNTN	NNNN	invalid
1[00]	NNTN	NNNN	invalid
[10]	NTNN	NNNN	invalid
0[01]	TNNT	NNNT	1[01]
	TNNT	TNNN	invalid
1[01]	NNNT	NNNN	invalid

Table 1 – Code space changes from flux density reductions

None of these errors can produce a transition of the coded data from a '0' value to a '1' value. So no weak bit error can ever turn a '0' into a '1', it can only turn a '1' into a '0' or produce an invalid code space output. Additional consistency checks could potentially detect errors such as the transition of 0[00] into 1[00] but the previous '1' bit could not be the result of a weak bit (or its coding would be a '0' to '1' transition that weak bits cannot produce in its position). This then eliminates the otherwise possible error turning 0[01] into 1[01] and 0[00] into 1[00], leaving only the transition of 1[11] into 1[01] or [11] into [10] as results from reduced electromagnetic flux density in transitions. If the previous bit was not a 1 [NT] or the reduction in flux density reduced the T to an N, then the 1[01] error is also impossible.

Unfortunately, the design of floppy controllers does not always have to produce error outputs for non-existent transition. That depends on the specifics of the controller electronics design. As identified in [1], the finite state machine for producing output bits based on current state, previous bit value, and transitions is shown in Figure 4. Note the lack of state transitions identified for the 0/11 and 0/01 cases and the 1/10 and 1/11 cases. These are not defined for the code space because they are unused values in valid encodings. They are typically designed as “Don't Care” (DC) values which leaves the designer free to optimize the electronics by ignoring the outputs that in theory cannot happen. In practice, a weak transition could produce a change from 0/10 to 0/11, however, the controller would be in state A and this is only identified as a transition for state B. That would happen is arbitrary depending on the design choice. The incomplete specification of error states leaves this in question.

Fortunately, there is also a cyclic redundancy check (CRC) code used in floppy disks that can compensate for many such errors. This will be analyzed in detail later, however, these unspecified states in the FSM description leads to potential sources of error depending on the particular floppy drive electronics design.

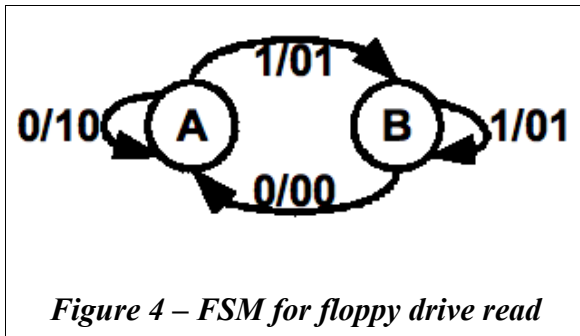


Figure 4 – FSM for floppy drive read

This analysis is based on the assumption that a weakened field density in the locality of a bit cannot trigger a transition and that is worth discussing further. Normally, in order for a transition to be detected by a floppy disk controller, the electromagnetic field density in one region has to be oriented in one direction, while the adjacent region has to be oriented in the opposite direction. Which

direction is 01 and which is 10 combined with the direction of the movement of the disk in the drive dictates whether the drive head gets a positive or negative impulse, but these are not differentiated by the controller – both are considered transitions. If a transition from the maximum field density to zero field density were to trigger a transition, floppy disks would be very unreliable because regions near tracks are commonly not used and any minor movement in the head could cause such a transition. In addition, the design of such devices is such that the positive and negative field densities are used to assure sound triggering. A half-level density change should not trigger a transition on most floppy disk drives. For that reason, even a full field density area next to a zero field density area should not trigger a transition, and thus the weakening of electromagnetic strength on the disk should not create transitions where none existed. Of course the physical phenomena associated with weak bits are analog at this level of granularity. The size of a region of storage on a 720K floppy disk is on the order of 1/8000th of an inch circumference. Because of this relatively high density, most common physical phenomena is unlikely to reduce the field density of one region to near zero while retaining the density of the area right next to it at full strength. Perhaps a scratch could cause this to happen, but in the case of a scratch, the damage would be permanent and would likely produce the same level of transition on each use.

An electromagnetic field such as is produced by a magnet passing near the disk or a temperature condition, or even a biological phenomena is highly unlikely to produce such a dramatic edge condition. There is a strong tendency for these phenomena to produce regions with decreasing effects as a function of distance, and this produces a slow transition in field density resulting in a change in field strength with distance that will not normally produce a transition in the floppy disk controller. As a result it appears to be a sound assumption that no transitions will be created by reduction in electromagnetic field density associated with weak bits, and only the loss of transitions is likely to occur from these physical phenomena.

4. CRC code analysis and potential error rates

In addition to the MFM coding, floppy disks also use a cyclic redundancy check (CRC) code to encode a value at the end of each sector written. This is highly likely to be inconsistent when specific classes of errors occur to portions of the sector. Specifically, they are readily able to detect single bit flips, multiple bit flips in close proximity, and many other combinations of bit flips. According to [3]:

“Any bit error term $E(x)$ which is an exact multiple of $P(x)$ will not be detected. This is the case, in particular, for the two bit error 10000001, where the two bad

bits are 7-bits apart. Note that 10000001 =(1011)(1101)(11). The allowable separation between two bad bits is related to the choice of P(x). In general, bit errors and bursts up to N-bits long will be detected for a prime P(x) of order N. For arbitrary bit errors longer than N-bits, the odds are one in 2^N that a totally false bit pattern will nonetheless lead to a zero remainder. In essence, 100% detection is assured for all errors E(x) not an exact multiple of P(x). For a 16-bit CRC, this means:

- 100% detection of single-bit errors;
- 100% detection of all adjacent double-bit errors;
- 100% detection of any errors spanning up to 16-bits;
- 100% detection of all two-bit errors not separated by exactly $2^{16}-1$ bits (this means all two bit errors in practice!);
- For arbitrary multiple errors spanning more than 16 bits, at worst 1 in 2^{16} failures, which is nonetheless over 99.995% detection rate.

If we assume that the CRC is intact, the impact of this coding on the available error modes from weak bits is such that the degradation mechanism would have to produce reduced flux densities exactly 32 transition distances from each other in order for the CRC code to fail to detect pairs of errors. Reductions in flux densities producing lost transitions in adjacent bits or other sequences of less than 32 transition areas (representing 16 bits of data) are 100% detected by CRC codes unless they range over large areas, in which case they would produce invalid codes in the MFM decoding mechanism in the controller. Thus, the physical phenomena identified with producing weak bits is very unlikely to produce a condition in which a correct match between data from a sector and the CRC code is attained, no MFM coding error is produced, and yet an alteration from the loss of a transition occurs.

This implies that if weak bits are the cause of an error and a successful read of the data with matching CRC code is completed, it is highly likely that the data recovered accurately reflects the data last written to that sector. While we do not know how to produce a precise calculation of the resulting probability, it is certainly less than the probability of errors associated with either the MFM or CRC codes alone. That is, there is no known synergistic effect that can cause one to correct an error produced by the other.

The method used tends to support the contention that the disk failures were caused by weak bits. Specifically, if another mechanism was in effect, such as alignment errors or mechanical defects in the original writer, then the realignment process would have yielded better or worse data instead of nearly identical error behaviors. If bits were not written at all or if a typical contemporaneous weak bit writing mechanism were used, the levels would not likely vary across such a wide range of rereads. The fact that different numbers of rereads were needed at different locations on the disk tends to indicate that the failure mechanism produced errors distributed over a range of loss of electromagnetic field such as that seen in overheating from poor storage, infection of the media with fungi or similar biological effects, or loss of data with time as is seen in many floppy disks, all of which take place over time as opposed to from instantaneous phenomena. These are precisely the sorts of errors that the CRC codes were designed to detect. No further examination of the media in this matter has been done at this time to identify the specific mechanism of failure.

There is, however, a lingering question that is worth addressing; the potential that repeated reads with reduced flux density in the sector and the CRC code could produce

result after result that would eventually lead to a valid CRC code and no MFM errors. This would then lead to false sector data accepted as legitimate. This particular scenario, because it involves weak bits, is somewhat less complicated to analyze than a scenario in which random changes are made, because the changes associated with weak bits tend to be all in one direction, thus eliminating transitions, and thus turning '1's into '0's. The likelihood of lost transitions causing detections is at least 17/22 for each transition based on the analysis of Table 1 (the number of invalid transitions vs. the number of rows). Because of the nature of the CRC code, errors that can go undetected also must be in quantities larger than 16 bits and distributed across the sector data area, or as combinations of the sector data area and the CRC area with probability that would appear to be no higher than 1 in 2^{16} . Since the CRC and MFM methods are not correlated in any way we are aware of, a reasonable assumption is that the likelihood of both failing to detect a change from reduced electromagnetic density is no greater than 1 in $2^{16} * (5/22)^{16}$, which is less than 1 in 10^{15} . The odds of coming across such an erroneous recovery is clearly low enough that for retries on the order of hundreds, there is almost no chance that false recovery could take place.

However, this analysis ignores the retry numbers from the actual recovery. In the specific matter under examination, many read errors were corrected after only a relatively small number of rereads, ranging in distribution from 1 to 15 retries with a few samples having more. Strangely, several of the sectors only read after about 80 retries, and none took significantly more than 80. The floppy drive does three retries per reported retry, so that actual number of attempted reads in these cases was about 240. Exact figures are unavailable because of court orders, it is impossible to repeat the examination for the same reason, and since the disk in question was not generated by us, we have no way to create another equivalent disk. It is somewhat disturbing that many sectors had on the order of 80 retries and those who got the recovered disk indicated that portions of the recovered blocks appeared corrupt. It would be beneficial to gain a better understanding of this.

In reviewing the situation, it was noted that the birthday paradox [4] shows that, in a room with 23 people having randomly distributed birthdays, the probability is just about $\frac{1}{2}$ that two will have the same birthday, even though there are 365 days in a year. Further analysis has shown that the 50% probability of matching birthdays (or random values within the range of a finite space) occurs when the number of samples is something like 1.1 times the square root of the size of the sample space as the sample space grows large. If the birthday problem is relevant, then for a 16 bit CRC, the square root of 65536 is 256. So as the number of reads reaches 281, the probability of a birthday collision is about 50%. But this problem is slightly different from the birthday paradox in that we are not selecting CRC values without replacement out of the total set. Furthermore, plots of the Birthday Paradox, which has no known closed form solution, show that the probability changes more or less linearly around the square root, so it would be unexpected that there would be a peak near the square root rather than a linearly increasing number of matches. It appears that some other phenomena is occurring and we do not know what it is.

The Birthday Paradox is not apparently the cause of the uneven distribution of recoveries, and furthermore, the results on CRC codes are not necessarily generalizable to weak bit failures which produce less than random results. If the floppy drive does not have detection of coding errors at the level of transitions and the don't care states of the finite state machine that decodes the content do not produce errors, there are other potential weakness of this technique and possible sources of error.

5. Correcting otherwise uncorrected errors

Given that some subset of blocks read successfully after about 80 retries are suspect because we don't know why the distribution is as it is, the errors that can be produced by weak bits are still limited, and this leads to a variety of helpful results. Two specific lines of approach are obvious. One is to do the reread process repeatedly and match the results from multiple runs to determine if there is consistency of some portion of the bits decoded across multiple runs. The other is to use a more in-depth analysis to try to determine which bits could be altered in this situation. By doing both, presumably consistent results would be very illuminating. Unfortunately, as floppy disks degrade, repeated reads cause them to degrade further because the read head literally rides on top of the floppy disk as the disk turns, creating friction that produces mechanical wear. This is problematic on the only copy of original evidence.

Analysis of the errors that can be produced by weak bits with consistent CRC codes has not been undertaken at this time, however, analysis of reconstitution of original content has been investigated to a limited extent.

Since only 1-0 transitions can occur and only in particular places within bit sequences. In particular, a 1-0 transition can only occur when a 11 turns into either a 10 or 01, shown as 11-[10/01]. Patterns appearing on decoded disk content cannot all result from lost transitions. Therefore, the candidates for lost transition changes can be substantially limited and specific bits can be definitively determined to not have resulted from loss of flux density. An approach to revealing which bits could and could not have been altered by such faults is to examine all possible 11-[10/01] transitions in each reread block and identify those that form valid parts of the code space both before and after transitions are lost. An observed 11 or 00 cannot come from such a change, so all pairs of 1s and 0s can be eliminated from the analysis, reducing the number of possible faults on a random content block by 50%.

In cases where languages are known and there is redundancy in those languages, the nature of the languages can be used to produce substantial improvements. The typical content of English, for example, is on the order of 2.3 bits per byte. [5] In theory, this means that if 4 bits per byte are potentially corrupted and each of the two remaining pairs could only have been produced by one of two codings, all of the original text should be recoverable. For example if the original text was "This" in ASCII, there are only a few resulting outputs that can stem from missing transitions. Note that all intra-byte pairings include a 0 because ASCII is 7-bit and thus the initial 0 bit stops and 1-0 transitions from crossing the byte boundaries.

```
This =8 124 150 151 163 (the initial '0' bit stops intra-word effects)
1248 = 01 010 1002 No valid weak bit errors
1508 = 01 101 0002 -> 00 101 0002 = 508 = '(' = T(is
1518 = 01 101 0012 No valid weak bit errors
1638 = 01 110 0112 -> 00 110 0112 = 638 = '3' = Thi3
1638 = 01 110 0112 -> 01 010 0112 = 1238 = 'S' = ThiS
1638 = 01 110 0112 -> 01 100 0112 = 1438 = 'c' = Thic
1638 = 01 110 0112 -> 01 110 0102 = 1628 = 'r' = Thir
1638 = 01 110 0112 -> 01 110 0012 = 1618 = 'q' = Thiq
```

There are also double bit errors that can result from weak bits in this situation, and they produce similar new valid codes, resulting in additional codes for “s” only. There are also other valid codes that can produce these same values from different lost transitions. For example, 161₈ (01 110 001₂) can be produced by 11 110 001 and a wide range of other values that involve turning 0s into 1s.

The process for inverting these faults involves generating the set of all possible source bytes and then eliminating those that do not fit within the language in use. For example, 'Thiq' can be inverted to create quite a few different characters to replace the 'q', but the only valid ones in English would have to be valid words starting in 'Thi', which include only “Thin” and “This”. The code for “n” is 110₈ or '01 001 000₂', which cannot result in 161₈ through any combination of missed transitions. Similarly, “Thir”, “Thic”, “ThiS”, and “Thi3” cannot be generated from “Thin” but can be generated from “This” with only 1-0 transitions. Extending this to the word as a whole, “T” and “i” cannot be altered by 1-0 failures from missed transitions, and other sources of '((50₈) that fit into the English word T?i? where the second '?' must be transformable into any one of the identified values are again limited.

As a final note, this technique does not directly apply to RLL coded disk drives, which became predominant among hard disk drives starting in the 1990s, and a different coding analysis is required for recovery of weak bits on those media. Similar problems with degraded CD-ROM drives also requires yet another coding analysis.

6. Summary, conclusions, and further work

It appears that the multiple read technique is effective in that it produces meaningful results in a reasonable amount of time with relatively low damage to original evidence and a high likelihood of accurate results. It appears to be accurate in retrieving data that is otherwise unreadable because of reduction in magnetic field to levels relatively close to original levels such as occur in natural disk degradation with time. Because this technique is based on normal floppy disks reads by standard unmodified equipment, it is less likely to be challenged and easier to implement than more complex and expensive techniques involving some sort of electromagnetic examination of the media or modified electronics.

This method also has some disadvantages in that the repeated reads can cause added wear and tear on the original evidence which may be fragile, it is likely to suffer from increased numbers of reads over time if the failure mechanisms are worsened by the repeated uses, and it does not reveal the specific mechanism of failure even if it produces reasonable results. It is also possible that large numbers of reads will not produce a valid result and that the process will have to be manually terminated and restarted at the following sector, leading to less complete recovery, and of course involving human intervention. If very similar numbers of reads are required for repeated attempts or across multiple locations on the disk, it appears to be potentially indicative of a copy protection or similar scheme, but this has not been tested or validated at this time though our efforts.

There is the possibility that the repeated reads could produce invalid data that happens to match the CRC code on the sector without creating invalid MFM codes in the controller, leading to false results. This may increase to nearly 50% as the number of reads increases to around 240 but we don't know the mechanism yet.

In cases where numbers of rereads are on the order of hundreds, or in other cases where reads can be completed with questioned data, the limits of 11-[10/01] transitions

can be used to restrict the possible original values and the languages used can further limit the possible original content, ultimately leading to clear resolution of the original content in many cases. The same methods apply to all MFM coded media, and while this does not apply to modern hard disks, it applies to floppy disks even today and historic hard drives.

Further work to automate the decoding and analysis processes would be useful and might be applied to other codings for other disk types as well. Further analysis of multiple errors in CRC codes is also needed to clarify the cause for high probability of success around 80 rereads. Unfortunately, doing repeated experiments is problematic because of the inability to readily repeat fault producing conditions in a timely manner. In the particular case the results were accepted by the court and the parties to the matter.

Finally, similar coding analysis of RLL codings used in most modern hard drives does not yield the same properties of always turning 1s to 0s on reduced flux density situation. As such, a far more complex analysis is required for decoding degraded RLL coded drives. Similar analysis of CD-ROMs and other media will also yield differing results and this analysis should be undertaken in order to generate definitive methods of analysis of these media under similar circumstances.

References:

- [1] Paul H. Suegel, "Recording Codes for Digital Magnetic Storage", IEEE Trans Magnetics Mag-21#5, September, 1985.
- [2] Hans-Joachim Leimkuller , "Computer Evidence : Analysis and Recovery of Magnetic Storage Media" 0-7803-2627-X/95 IEEE 1995.
- [3] Roger L. Freeman, "Practical Data Communications", John Wiley & Sons Inc, 1995 ISBN 0-471-31021-2 (cited in <http://www.ee.unb.ca/tervo/ee4253/crc.htm>).
- [4] Donald Knuth, "The Art of Computer Programming - Sorting and Searching", pp 506-7, Addison Wesley, 1973 which cites an unpublished work of H. Davenport.
- [5] Claude Shannon, "A Mathematical Theory of Communications", Bell Systems Technical Journal. 3, no. 27, July 1948.