TECH NOTE

Group-Coded Recording Reliably Doubles Diskette Capacity

Pawitter S. Sidhu

Wangco, Incorporated
Orbis Division
Tustin, California

To squeeze more data onto a standard diskette requires sophisticated encoding in order to avoid exceeding the physically attainable limits of standard codes; GCR is the answer

> Ed Note: This is the third of three Tech Notes on the general subject of double-density recording; previous Notes appeared in the September and October issues of *Computer Design.* The Notes, considered together, provide three different solutions to the problem of increasing the capacity and data rate of a flexible disc, and illustrate the tradeoffs that should be considered in choosing one approach over another.—WBR

Diskette drive systems capable of storing more data and transferring them at higher data rates are in increasing demand, and call for improved recording techniques. Higher density recording automatically achieves both increased capacity and increased data rate in a given diskette drive. However, it requires a different magnetic encoding method, because the standard code is already near its density limit. The selected code must both *cope with* and *compensate for* mechanical and recording media imperfections. Group-code recording (GCR) improves the data packing density nearly as far as possible, while offering maximum timing margins, self-clocking, and permitting a unique synchronization pattern of all 1s.

Frequency modulation (FM) is the standard code for diskettes. This technique is a continuance of the recording method for hard discs principally used by IBM. Modified frequency modulation (MFM) or Miller code is one method that has been used for doubling the density. However, MFM results in lower data reliability. After a fresh look at high efficiency codes, particularly nonreturn-to-zero (NRZ), our company selected GCR, a modified NRZ, as the code to double the recording density of floppy discs used on a standard disc drive, instead of modifying an inefficient FM code. The technique is equally applicable to other magnetic recording media where high data density and *reliability* are mandatory.

In a standard FM code, each bit cell boundary is marked by a flux reversal, called a clock. An extra flux reversal for a 1 bit is placed in the middle of the bit cell [Fig. 1(a)]. A 0 bit is indicated by the absence of this extra flux reversal. Since at least one-half of the flux reversals are exclusively for synchronizing or clocking, FM code is limited to a maximum of 50% efficiency. A further disadvantage is the apparent tendency for flux reversals at high density (in any code) to shift away from one another. Because the minimum nominal time between flux reversals in FM is half the duration of one bit cell, maximum shift allowed for reliable data decoding is ±0.25 bit. More simply, FM encoding has a theoretical maximum margin of ±0.25 bit time.

However, hardware implementation of FM code is simple because, for a clock frequency of f, the code limits the system's bandwidth which lies between f and 2f. If the clock frequency is doubled, the bit cell length is halved, and the allowable shift drops to ±0.125 of the single-density bit cell length, taken as a reference. This is unacceptably small —only 19 microinches or half a microsecond on the innermost track, which already carries 6536 flux reversals per inch (frpi) on a standard FM-coded diskette operating at 250K bits/s.

The FM code may be modified to delete the clock bit transitions. In this technique, successive 1s are represented by flux reversals one bit cell apart; 10 and 01 combinations are represented by flux reversals 1.5 bit cells apart; while two or more 0s are represented by flux reversals at intervals of two bit cells [Fig. 1(b)]. Three frequencies are recorded, but reversals can occur either at the center or at the boundary of the bit cell, or at one-half bit-cell intervals. Thus the shift margin is ±0.25 bit time (the same as in FM code). Because the highest frequency is only f, not 2f as in FM, data density of MFM may be doubled or bit times reduced by half without exceeding the bandwidth of an FM system; the data rate can also be doubled. However, reducing the bit cell by half also reduces the shift margin from ±0.25 to ±0.125 bit time, as with FM. This leaves practically no margin for reliability.

In NRZ a 1 is represented by a change in flux level and a 0 is represented by the absence of such a change [Fig. 1(c)]. Successive flux reversals are one bit cell apart, and the theoretical phase margin is ±0.5 bit. Because only one flux transition is required for one bit, the code is highly efficient. However, a long string of recorded 0s produces no read signal, which could result in loss of data synchronization, and requires a system bandwidth from dc to the data rate f. This larger bandwidth reduces the signal to noise ratio.

Achieving Reliability

In GCR, as in NRZ, a 1 is represented by a flux reversal and a 0 by the absence of a flux reversal [Fig. 1(d)]. However, no more than two 0s are recorded in succession, assuring that synchronization, once achieved, will not be lost, and limiting the bandwidth of the system to between 0.33f and f. This bandwidth is much narrower than that of NRZ or FM, and the shift margin is ±0.5 bit. As a result, data density is doubled or bit time halved, relative to FM, for the same bandwidth. Furthermore, reducing bit cell length by one-half reduces the shift margin from ±0.5 to ±0.25 bit time—twice that of FM or MFM codes.

Group encoding avoids recording more than two 0s in succession by translating four bits into five. Four bits can occur in any of 16 combinations from 0000 to 1111. Of the 32 combinations of five bits, those that begin or end with more than one 0 and those that have more than two 0s internally are not used. After eliminating these, only 17 combinations are left; and of these, the combination 11111 is reserved for synchronization. The remaining 16 combinations of five bits are correlated with the 16 possible combinations of four bits (Table 1). When recording, each group of four bits in a sequence of data is translated into a corresponding group of five and recorded in conventional NRZ form. Then, when reading, the reciprocal five to four translation yields the original four bits of data. This

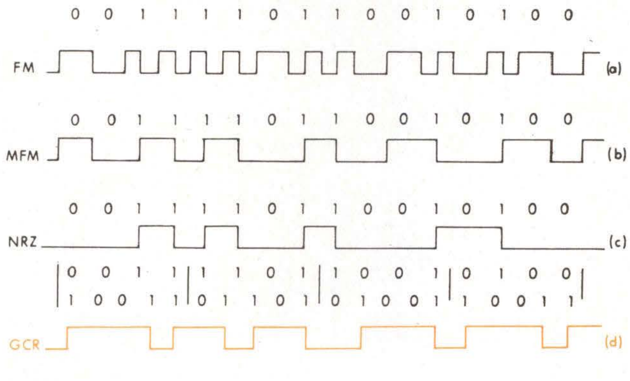

Fig. 1 Various recording codes. Standard code for diskettes is FM (a). Various modifications of FM, such as (b), are in use to increase density, but best approach is a version of NRZ (c) called GCR (d), that eliminates long strings of 0s. All four codes have the same data pattern, but GCR requires a translation of four bits to five as shown

TABLE 1

Translation Four To Five In GCR Code

4-Bit Data	5-Bit Recorded Data
0000	11001
0001	11011
0010	10010
0011	10011
0100	11101
0101	10101
0110	10110
0111	10111
1000	11010
1001	01001
1010	01010
1011	01011
1100	11110
1101	01101
1110	01110
1111	01111

TABLE 2

Comparison of Recording Codes

Code	Data Rate (kbits/s)	Recorded Bit Density (frpi)	Phase Margins	
			Bit Time	μs
FM	250	6536	±0.25	±1.0
MFM	500	6536	±0.125	±0.5
GCR	500	8170	±0.20	±0.8

translation from four bits to five further increases bit density by 25% from 6536 to 8170 frpi, and reduces the bit cell time to 1.6 μs, or 0.4 that of FM—a 20% reduction from the 2.0 μs of the uncoded data. Accordingly, the phase margin is ±0.20 instead of ±0.25.

The distinctions between FM, MFM, and GCR, summarized in Table 2, clearly show the advantages of GCR over MFM in terms of phase margin and data reliability.

Advantages of GCR

In a standard IBM-compatible diskette drive using standard media, data density is doubled simply by using GCR instead of FM. Doubled density automatically increases the data rate to 500K bits/s, while increasing the maximum flux transition density to 8170 frpi. These are increases of 100% and 25% respectively. Any well-designed read/write channel, and most available ferrite heads, can easily handle this 25% increase in flux transition density. In addition, GCR increases data reliability over FM or MFM because its shift margins are 60% larger. Increased phase margins also assure diskette interchangeability when written and read on different drives.

GCR is not without its disadvantages. Chief among these is the extra logic circuitry it requires for group formation and 4-to-5-bit translation. This logic, however, can be assembled inexpensively from standard MSI chips and conceivably could be implemented in software or on a single custom chip.

GCR System

The GCR code is implemented in the Orbis model 86 data encoder on a single printed circuit (PC) board. The data encoder can operate at either the standard rate of 250K bits/s, using FM, or double that rate with GCR. Either way, it accepts data in FM code from an external controller furnished by the user, which translates data from parallel ASCII or whatever code the host computer uses. The controller also distinguishes one record from another by incorporating an address in the recorded data, by keeping track of the angular position of the rotating disc, or by other means.

In standard 250K-bit/s mode, the encoder is logically bypassed, and the data pass directly to and from the controller in FM code. However, in double-capacity mode, the data encoder is a 2-way synchronous translator, communicating with the controller in FM code at 500K bits/s but storing data in GCR at 625K bits/s. Storage density on the innermost track at double capacity is 8170 frpi, corresponding to 6536 bits/in.—compared to 6536 frpi and 3268 bits/in. at single capacity.

Hardware to implement the GCR code has two major sections: a phase-locked oscillator (PLO) to control passage of data through the system, and a 4-to-5-bit data encoder/decoder. The PLO uses two standard integrated circuits (ICs). During either read or write operation, data applied to the PLO inputs bring the output frequency in phase with them. Two counters divide the main oscillator frequency of 5 MHz by 10 and by 8, yielding 500 and 625 kHz respectively (Fig. 2). When data are being read, the PLO follows minor variations in data rate corresponding to diskette speed variations, but averages the rapid variations caused by shifting flux reversals.

All data moving in either direction through the GCR system pass through two 4-bit registers, A and B, and one 5-bit register. The 4-to-5-bit encoder lies between register A and the 5-bit register in the data stream, and the 5-to-4-bit decoder is between the 5-bit register and register B (Fig. 3). Write data, previously coded into FM, are entered serially into register B, then A, at 500 kHz (one bit

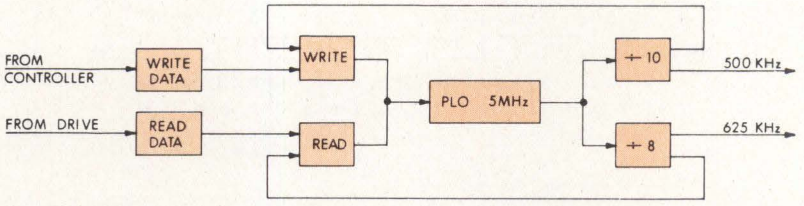

Fig. 2 Phase-locked oscillator. Data passing in either direction synchronize the circuit; two output frequencies control shift registers containing raw or encoded data

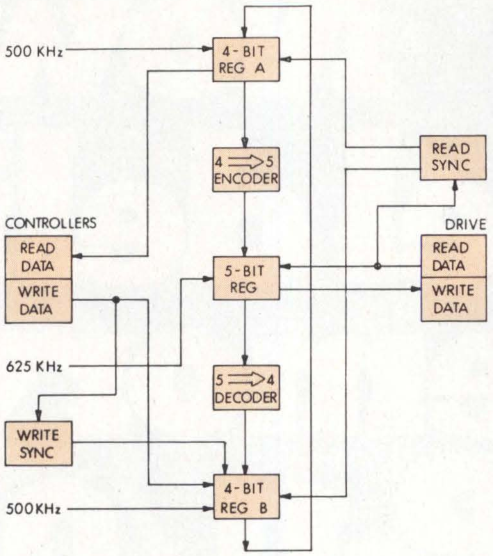

Fig. 3 Data path. Two 4-bit and one 5-bit registers process all data passing to and from the disc, encoding and decoding the GCR format

every 2 μs), translated into the 5-bit code, and strobed by the write sync logic into the 5-bit register at 8-μs intervals. These five bits are serially transferred to the diskette at 625 kHz in NRZ code. During read operation the data pulses, in NRZ code, enter the 5-bit register serially and are decoded to four bits, which are strobed by the read sync logic into register B. From here, again the data pass to register A and then to the host system at 500 kHz.

Every record begins with a string of 1s, forming a preamble and a synchronizing byte, and is generated by sending the output of the 625-kHz clock directly to the recording head. The preamble signals the beginning of a record, and synchronizes PLO into step with the data stream when reading. Data, however, are not encoded until the sync byte has been written. When encoding starts, the write logic begins transferring four data bits through the 4-to-5 encoder every 8 μs as described previously. Another sync byte is written at the end of the record, which can be of fixed or variable length as required by the controller and host computer.

When reading, the logic looks for the unique all 1s pattern, which never occurs in data; having found such a pattern, it then goes into preamble mode until the sync byte is recognized. At that point, decoding begins, and continues until the terminating sync byte is found. The preamble and sync byte, by the way, not only bring the PLO into phase, but also can identify the density of the recorded data—meaning that single- and double-density records can be intermixed on one diskette.

Physically, the encoder/decoder consists of 35 small-scale ICs on a PC board. If some of its functions are incorporated into the user's controller, such as read and write sync logic, translation of host computer logic levels to those of the diskette and vice versa, or using the controller's 625-kHz clock, then the GCR logic can be *reduced* to as little as seven chips unlike decoder for MFM or M^{2}FM (modified, modified frequency modulation).

Summary

Orbis has chosen the GCR technique for *reliably* doubling the recording density on its flexible discs because of its advantages of larger phase margins, distinctive preamble patterns, and the absence of requirements for write pre-emphasis or distorted timing. Its principal disadvantage, compared to MFM, is the extra hardware that it requires. ☐