GCR Increases

data recording rates and reliability

Group-coded recording (GCR) of data onto magnetic tape has been around for a number of years. IBM first announced it in 1973. Even though GCR increased data recording rates and reliability, most manufacturers originally stayed out of the GCR race. They felt that improved tape rate and reliability would prove unnecessary, because other storage media (such as disk) might push tape aside. They were wrong: the GCR tape market, $3.5 billion four years ago, should reach three times that in 1981-82. Today, over 70% of large computer systems installed in the U.S. utilize GCR-equipped tape subsystems.

Why this dramatic growth? The reason: tape remains the least expensive medium for data storage — approximately one-tenth the cost of disk. It is the only legal electronic means of archiving data, and is the most interchangeable medium.

Designing faster tape drives did not appear to be a solution for improving tape-subsystem performance, because electromechanical limitations make speeds above 250 ips impractical. The alternative (putting more data on the same amount of tape by increasing fluz density) seemed more practical. However, increased density called for improved methods for detecting and correcting errors in the data.

by Mike Newton

Non-return to zero (NRZI) recording, the earliest method, writes 800 bits of information per inch of tape. Magnetically, NRZI relies on a flux change in the oxide coating of the tape to indicate a '1' bit of information, while the absence of a flux change signifies a '0'. Because NRZI requires one flux change for one bit of information, the format provides 100% recording efficiency. Unfortunately, mechanical skew considerations limit NRZI density to the 800 bpi range, and the format cannot correct errors.

Phase encoding (PE), which uses a self-clocking approach, was next developed to overcome the deficiencies of NRZI. Like NRZI, PE records nine tracks on the tape. However, it records the '1' and '0' bits via a flux change — the difference between the two bits depends on direction of flux change. PE suffers from the drawback of needing at least one flux change per bit. Two like bits in succession require a second flux change at the boundary between the cell bits. This flux allows write current to return to a state from which it can write a flux change in the same direction as flux change of the preceding bit of information. Thus, PE can require as many as two flux changes per bit of information that give it a recording efficiency of 50%.

At least one flux change occurs per bit cell in PE. The technique benefits from the fact that each of the nine tracks can have its own clocked detection circuit. This circuit will have a variable-frequency clock running in a phase-locked loop with data on the track. With a clocking circuit monitoring each track, added hardware can sense flux

Mike Newton is from Storage Technology Corp.

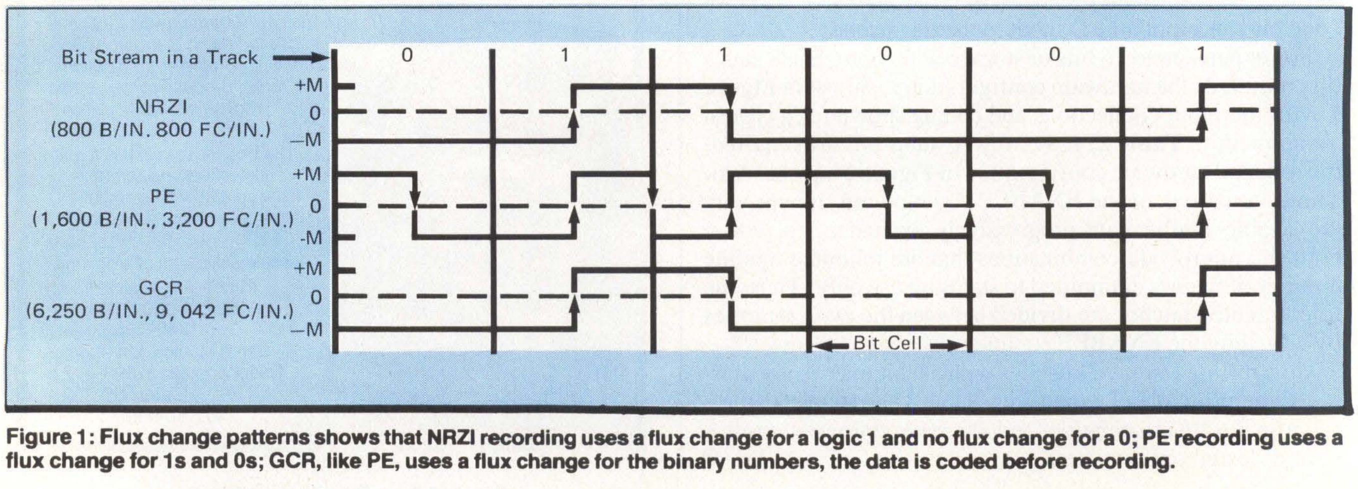

Figure 1: Flux change patterns shows that NRZI recording uses a flux change for a logic 1 and no flux change for a 0; PE recording uses a flux change for 1s and 0s; GCR, like PE, uses a flux change for the binary numbers, the data is coded before recording.

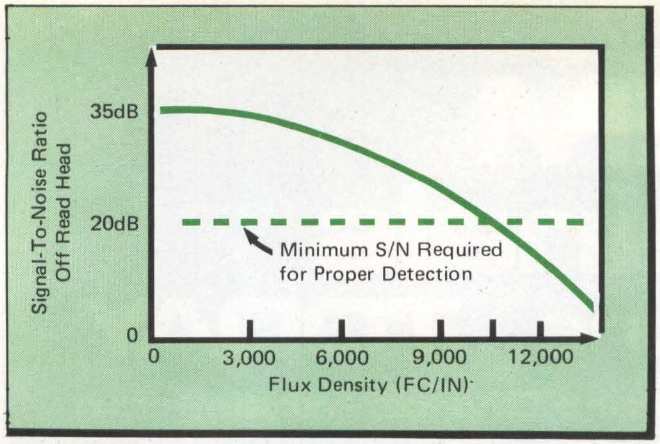

Figure 2: Signal-to-noise ratio at the read head depends upon the closeness of the flux change on the tape. GCR, which operates at a flux density of 9000 flux changes/inch (fc/in.), provides an S/N above the minimum needed for reliable information.

change amplitude and position within each bit cell. Detection of either of these two conditions occurring outside normal paramaters indicates erroneous data within a specific track. Byte-by-byte parity checking can further pinpoint the incorrect bit or bits within that track and correct errors on the fly. However, in any given data record, only single-track error correction is possible via these safeguards.

Since the independent clock of each of the nine tracks virtually frees PE from skewing problems experienced with NRZI, it allows PE recording at 1600-bpi density. It is possible to use PE at higher densities, but the increased frequency of flux changes begins to cause an unacceptable signal-to-noise ratio. Also, dual-track error correction — not provided for by PE — becomes important at higher densities.

what is GCR?

Very simply, GCR is a recording format that takes advantage of the efficiency of NRZI while implementing the clocking procedures of PE. Aided by a sophisticated approach to error correction, GCR has resulted in a 6250-bpi recording density, plus a dual-track error detection and correction capability. The 6250 density was selected, because 6250 bpi processed at 200 ips equals 1.25 Mbytes/sec., which is a data rate compatible with many of today's high-performance computer channels.

how does it work?

Like its predecessors, GCR records on nine tracks. Basically, it uses the NRZI convention: a flux change represents a '1' bit of information, while the absence of a flux change indicates a '0' bit. However, it requires a modification. Originally, a NRZI recorder utilized no clocking system, and it could write a string of '0' bits, represented by a series of bit cells with no flux change. Incorporating a clocking system into the GCR approach eliminated this capability, since periodic flux transitions are necessary to ensure the synchronization of the clocks on the nine tracks. Essentially, GCR can write no more than two '0' bits consecutively.

This is where the term "group-coded" comes in. Before the GCR system records the information, it collects the data in an eight-byte buffer. The tape controller adds seven data bytes and an eighth byte for error checking and correcting information. This function produces an eight-byte data group. Although the information comes in broadside (byte-serial) from the CPU, it is coded bit-serially in subgroups of four bytes.

For clarity, let's number the bits in each byte, #1 through #9. The translator receives data bits four at a time, and converts the four #1 bits into a five-bit code character, the four #2 bits into a five-bit code character, and so on down through the four #9 bits. The resulting nine five-bit characters are then recorded onto tape. Thus, each eight-byte data group is recorded as a ten-byte storage group. For retrieval, the process is reversed. The four-to-five translation scheme was devised to provide a five-bit storage subgroup that, for each four-bit data subgroup, contains no more than two consecutive '0' bits and no more than one '0' bit on either end. This scheme insures that no recorded track will ever contain more than two consecutive bit cells without a flux transition.

error detection and correction

PE error-correction system provides single-track error correction. The more extensive GCR approach utilizes a system of cyclic codes recorded with the data. GCR uses the same hardware as PE — amplitude sensing (noting the fluctuations in flux-change amplitude) and phase error (checking the exactness of the flux location within the bit cell) — to give the system its multiple-bit error-correction ability.

The error-checking and correcting (ECC) byte of information (attached to the data subgroup in the buffer prior to encoding is generated by a polynomial as described in ANSI spec ×3.54). This polynomial is based on the 56 bits of information that make up the data group. Thus, a direct mathematical relationship exists between construction of the ECC byte and all data from which it is generated. When the data is read back, control hardware reworks the polynomial generation and checks the resulting error-correction byte against the ECC byte originally written. If a discrepancy occurs, it can be used to calculate exact position of the erroneous bit or bits within the data group. Although this technique can correct only single-bit errors, amplitude sensing and phase error checking can correct multiple-bit errors.

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

Figure 3: Group coding starts with the data entering the controller from the CPU in byte-serial order. The data bytes are stacked in the controller buffer. An error-correcting code (ECC) byte is generated by, and added to, every 7 data bytes to make an 8-byte data group. The translator converts the data into storage code. This process is bit-serial in which each 4-bit data subgroup is assigned a 5-bit storage code. See Figure 4 for a more detailed explanation.

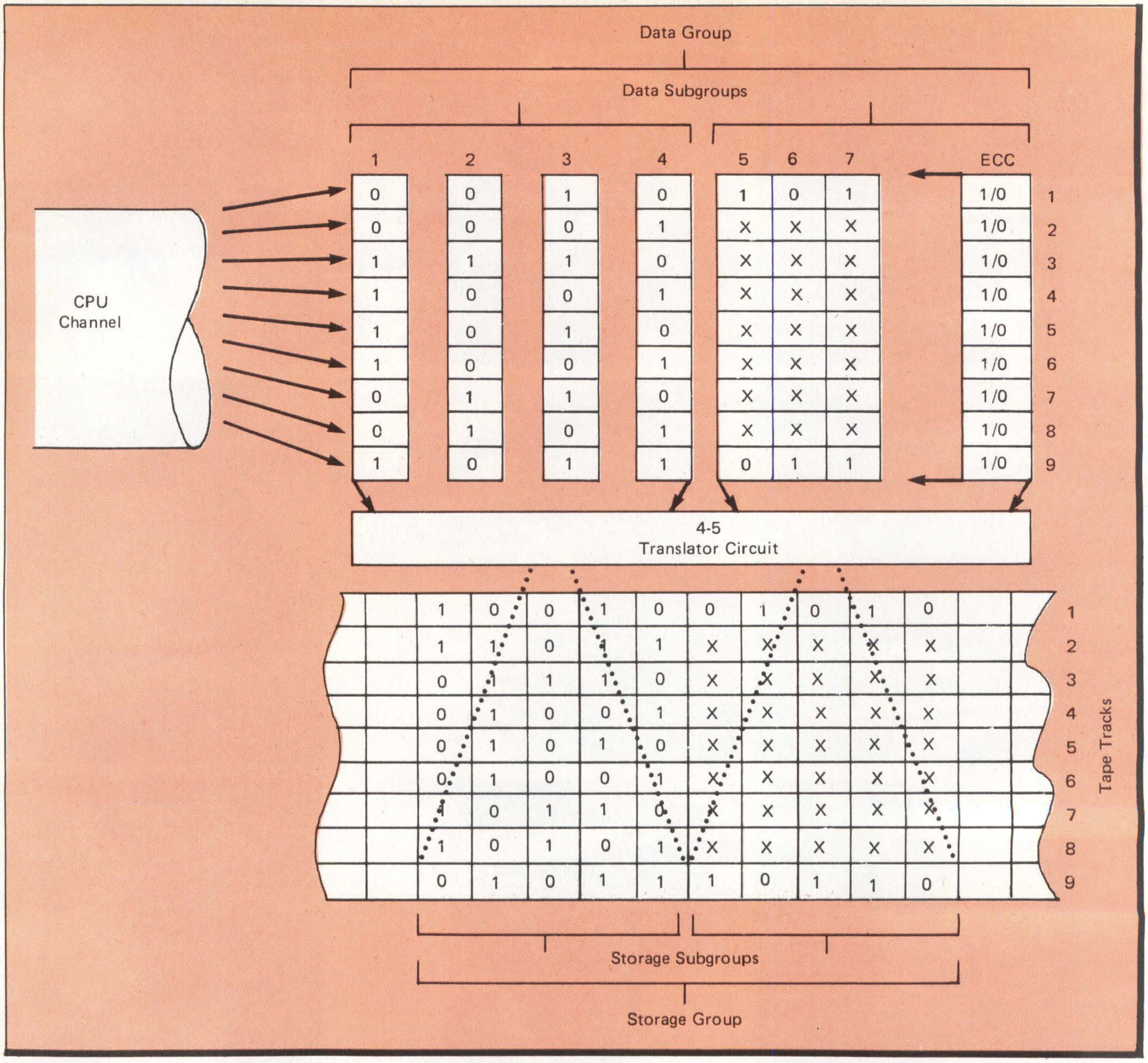

Figure 4: The four-to-five GCR translation scheme insures that, when data is recorded on the tape, no more than two successive zeros appear on any track. This scheme makes synchronization of the read head more reliable.

Two additional error check characters, incorporated into each storage record, supplement the GCR error detection and correction systems. A polynomial, defined by ANSI specs, generates the AuxCRC (auxiliary cyclic redundancy character). A different polynomial, based on all the data, plus the pad bytes (inserted after the residual customer data bytes to fill out the last storage group) and the AuxCRC itself generates. The AuxCRC and the CRC are generated in a manner similar to the ECC, and provide a final check to insure the integrity of the data after it is read. These characters do not locate or correct errors, but simply indicate that an error or errors exist. Whenever that happens, the computer is automatically alerted to switch to retry procedures.

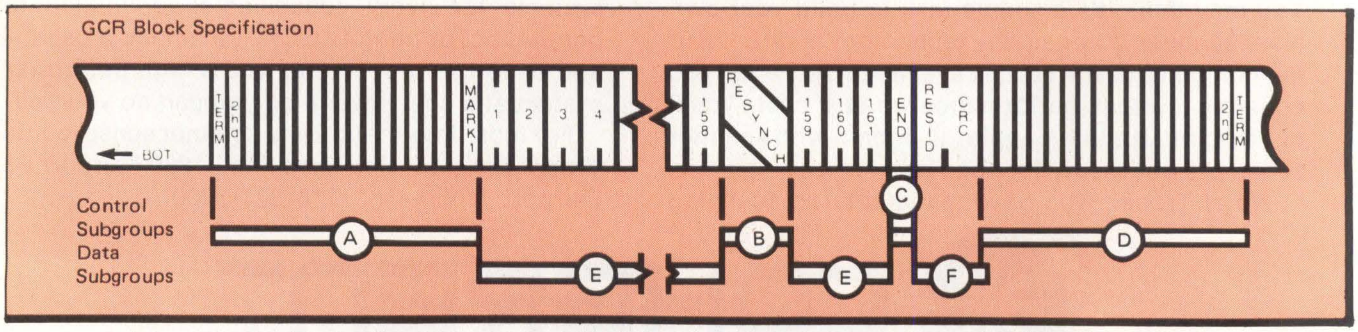

Figure 5: GCR group consists of preamble (A), resync burst (B), end mark (C), postamble (D), storage groups (E), and residual and GCR groups. (F).

resynchronization

Amplitude sensing, used with GCR to allow multiple-bit, double-track error correction, works the same way with GCR as it does with PE recording. When an erroneous bit is sensed, the dead-track register causes faulty track shut down. This action could create a real problem in GCR, since long data blocks are desirable to optimize the advantages of GCR's increased density and to avoid devoting too much tape space to interblock gaps. It is obviously unsatisfactory for one or perhaps two tracks to shut down for the duration of a record of that length. Consequently, the GCR format provides an opportunity to reset the dead-track register. It also resynchronizes the read detection circuits during recording by writing a resync burst in all tracks after every 158 data groups, provided that at least one more data group remains to be written. The resync burst consists of a Mark 1 subgroup, ten '1'-bit bytes, and a Mark 2 subgroup. The burst is unique and automatically triggers resynchronization of the nine read detection circuits. Because error probability relates directly to record length, and since resynchronization occurs after every 1,106 data bytes, the inclusion of this precaution reduces the probability of an uncorrectable error in an 8000-byte data block by a factor of almost seven.

the GCR data block

So far, we've discussed data groups, AuxCRCs, CRCs and resync bursts. Next we will discuss the other components that make up a GCR data block and their assembly on tape.

The Preamble. Just as PE does, the GCR record begins with a block of information designed to alert the amplitude sensors of the beginning of a block, and to synchronize the read-detection clock for each tape track. The preamble consists of a term subgroup and a secondary subgroup. These subgroups announce the beginning of a data block, followed by 14 subgroups made up entirely of '1' bits which allow the read-detection circuits to synchronize. A Mark 1 subgroup ends the preamble.

Storage Groups. Groups of coded data are then written on the tape. Each group consists of 10 bytes. When decoded, the 10 storage bytes are reconverted back into seven bytes of data and the ECC byte.

Resync Burst. After each 158 storage groups (data), a written resynch burst resets the dead-tracking circuits and brings the read-detection circuits back into sync.

End Mark. When the system has finished writing all the storage groups (data), it writes End Mark on all tracks.

Residual Group. Each group contains (prior to coding) eight bytes of information (7 data + ECC). Therefore, if after the system has coded and written all data groups, fewer than seven bytes of customer data remain, they are written into the residual group. The system then fills the group up with pad bytes of zeros, plus an AuxCRC byte and an ECC byte.

CRC Data Group. Next, the system writes a CRC group containing the CRC bytes and the residual count byte. The count byte describes the number of actual data bytes vs. pad bytes in the preceding residual group. Five or 6 CRC bytes (all identical) plus an ECC byte make up the CRC group prior to conversion by coding into a 10-byte storage group.

The Postamble. A mirror image of the preamble, the postamble which finishes the record contains a Mark 2 subgroup (the reverse of a Mark 1), 14 subgroups made up entirely of '1' bits, a secondary character and a term character.

We've seen how GCR provides nearly error-free storage of information at low cost. Keep these fundamentals in mind when selecting tape storage subsystems. **D**