TABLE 9.5 5B6B Sub-block Encoding

Input Character	Binary Value HGF	Encoded Value fghj Positive Disparity	Encoded Value fghj Negative Disparity
D/Kxx.0	000	1011	0100
Dxx.1	001	1001	
Kxx.1	001	0110	1001
Dxx.2	010	0101	
Kxx.2	010	1010	0101
D/Kxx.3	011	1100	0011
D/Kxx.4	100	1101	0010
Dxx.5	101	1010	
Kxx.5	101	0101	1010
Dxx.6	110	0110	
Kxx.6	110	1001	0110
Dxx.7	111	1110 (0111)	0001 (1000)
Kxx.7	111	0111	1000

ous character to the current 5B6B code. If the 5B6B code is neutral, CRD´ will reflect the disparity of the generated 5B6B code. Otherwise, the lookup table will attempt to choose a code that balances out the CRD. If the character maps to a neutral code, the CRD will be passed through. The 3B4B table not only performs a simple mapping of the Y sub-block, but it also handles the alternate encoding of the special cases mentioned previously. The final CRD from this table is stored for use in the next character encoding.

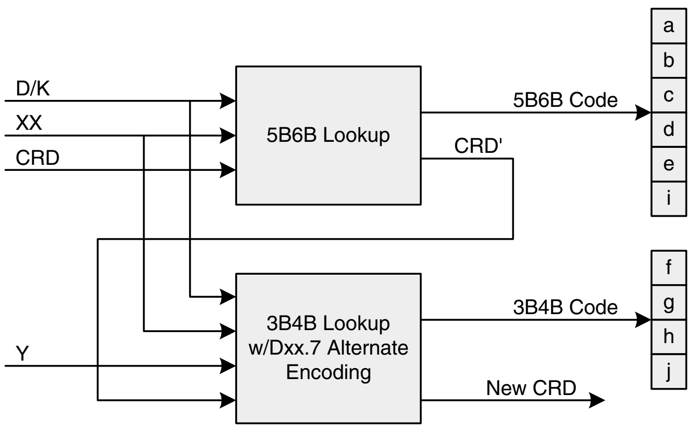

FIGURE 9.8 8B10B encoding logic.

The 8B10B decoding process requires similar lookup tables that perform the reverse operation. Decoding should also deal with CRD errors and invalid characters that indicate a bit error on the data link. However, not all bit errors on the data link will result in CRD errors or invalid characters.

9.6 ERROR DETECTION

Error detection and recovery are key requirements of data communications systems, because undesired results can occur if corrupted data is handled as if it were correct. While you might not mind an extra 0 being added to your bank account balance, you would certainly not want a 0 accidentally removed due to a data error! Transducer circuitry seeks to achieve the lowest *bit error rate* (BER) possible, but it will never be 0. Bit error rates of 10^{-10} to 10^{-12} are commonly achievable in wired (copper and fiber optic) data links. When these links carry data at 1 to 10 Gbps, errors will statistically occur every few seconds or minutes. These statistics make bit errors infrequent but recurring events that must be handled appropriately.

It is easier to detect an error than it is to correct one. Certain coding schemes, including 8B10B, provide some inherent bit error detection capability. If a bit error causes the detection of an invalid 8B10B code, or one with the wrong disparity, the receiver can detect the error. However, 8B10B coding cannot be relied upon to detect all errors, because not all errors will result in an invalid code word. Some single-bit or multibit errors will result in a different yet valid code word.

For channels with relatively low BER (e.g., high-quality wired data links), coding with a low-overhead scheme such as 8B10B is sufficient, and the responsibility for error detection and recovery can be passed up to the upper protocol levels. If errors are rare, the protocol-handling logic (both hardware and software) will not have to spend much time recovering from errors. A typical action when an error is detected at the protocol level is to request the retransmission of the affected frame. Such retransmission is expensive in terms of relative time and effort but is usually insignificant overall because of the low error rate. This situation changes when a channel has a higher BER.

A high-quality wireless data link may exhibit a BER of 10^{-6}, making errors much more frequent than in a high-quality wired channel. If all errors were handled by retransmitting frames, overall system throughput would suffer, because a much higher proportion of traffic would be devoted to error recovery. In these situations, it is worth utilizing a coding scheme with higher overhead that is capable of not only detecting errors but correcting them on the fly as well. Such schemes are called *forward error correction* (FEC). FEC codes calculate additional bits to be sent along with each data unit that have a degree of redundancy built into them so that if one or more bits get changed, a logical transformation can detect the mismatch and determine the correct values. It stands to reason that the overhead of FEC increases with the desire to correct increased numbers of bit errors within a single coded data unit. More FEC bits are required to correct two simultaneous bit errors than a single bit error. The decision on how complex a coding should be used is based on the channel's BER and on the penalty for passing error recovery functions to the protocol level. These characteristics are analyzed mathematically by considering the *coding gain* of a particular FEC code. An FEC code can be considered as its own channel that, rather than causing bit errors, resolves them. When the FEC channel is placed together with the real channel, the FEC coding gain effectively reduces the BER of the overall channel. FEC and its implementation are complex topics that are covered in specialized texts.

Regardless of whether a communications channel implements FEC, the data that is passed to higher protocol layers is subject to a net nonzero BER. The responsibility for handling these remaining errors lies at the data link layer and above. Network frame formats usually contain one or more error detection fields. These fields generally fall into one of two categories: *checksum* and *cyclic redundancy check* (CRC).