BUILD A 40-MHz DIGITAL FREQUENCY COUNTER

Popular Electronics®

WORLD'S LARGEST-SELLING ELECTRONICS MAGAZINE

JUNE 1977/$1.25

Creative Recording with 4-Channel Tape Recorders

How To Program H-P Calculators For Fun & Games

★ BLACKJACK ★ FOOTBALL
★ TESTING ESP ★ DIVE BOMBER
★ FORECASTING BIORHYTHM

Build a State-of-the-Art Battery Charge Monitor

Analyzing Performance Capabilities of the New 40-Channel CB Transceivers

TEST REPORTS:

Technics "Linear Phase" Speaker System

Lafayette LR-3030 Stereo FM Receiver

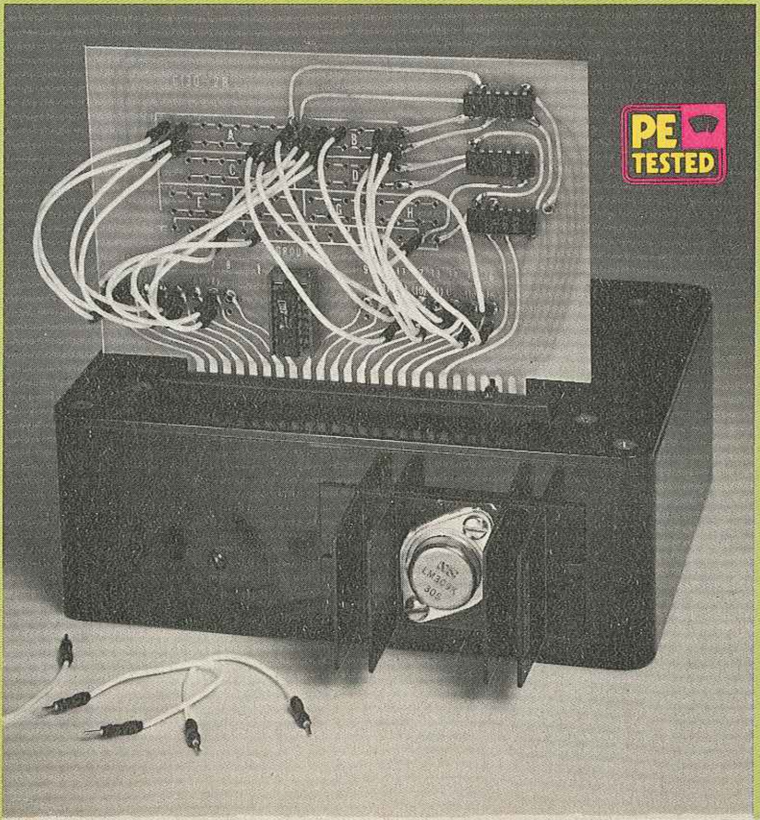
Kris XL-50 40-Channel AM Mobile

16-PAGE BONUS!

SPECIAL FOCUS ON Test Instruments

INCLUDING

HOW TO BUILD A LOW-COST DIGITAL IC TESTER



How Computers Detect and Correct Transmission Errors

Parity checking and redundancy are two of the devices used in error detection.

BY JEROME MAY

ALMOST everyone has had some experience in dealing with a computer mixup. One classic story goes like this: a computer, doing one of the earliest payroll jobs, put a 1 where a zero should have been, and the next week the janitor picked up his paycheck for exactly $1,000,147.38!!

Systems engineers and serious microprocessor hobbyists are aware of the problems that random noise can cause on data lines. Noise can cause the "three" that was sent to show up as a "seven."

Because "error" can be treated as a random event, having equal likelihood of occurring in any given datum, it becomes possible to apply techniques of *information theory, probability,* and *statistics* toward designing systems that are resistant to this type of error.

A simplifying assumption that will be useful is that, in a data word consisting of *n* bits, only *one* bit will be in error. The treatment of multiple-error-detecting systems uses methods similar to those discussed, but the treatment becomes extremely mathematical and complicated and is beyond the scope of an introductory article.

DECIMAL	BINARY	BCD	EXCESS-3	GRAY CODE	DECIMAL
0	0000	0000	0011	0000	0
1	0001	0001	0100	0001	1
2	0010	0010	0101	0011	2
3	0011	0011	0110	0010	3
4	0100	0100	0111	0110	4
5	0101	0101	1000	0111	5
6	0110	0110	1001	0101	6
7	0111	0111	1010	0100	7
8	1000	1000	1011	1100	8
9	1001	1001	1100	1101	9
10	1010	xx	xx	1111	10
11	1011	xx	xx	1110	11
12	1100	xx	xx	1010	12
13	1101	xx	xx	1011	13
14	1110	xx	xx	1001	14
15	1111	xx	xx	1000	15

Fig. 1. Five different 4-bit codes for the decimals 0 through 15.

Redundancy. An error can be detected by redundancy, which is the inclusion of extra information with each data transmission. This extra information helps the receiver to decide if the data it has received has been altered in transmission.

Obviously, the simplest error detecting system, conceptually, would be the transmission of each data unit twice. Thus, if the first transmission does not match the second transmission, the receiver can signal that an error has occurred and to please repeat the data.

"This this has has some some obvious obvious disadvantages disadvantages,, and and the the search search goes goes on on for for better better methods methods.."

Restating the problem, the decimal numbers 0 through 15 have exactly 16 representations in binary, as shown in Fig. 1. Since all possible combinations of 0 and 1 in four positions are used, there is no way to detect datum error because all combinations are equally likely; there is no room for redundancy.

DECIMAL	BINARY	PARITY
0	0000	1
1	0001	0
2	0010	0
3	0011	1
4	0100	0
5	0101	1
6	0110	1
7	0111	0
8	1000	0
9	1001	1
10	1010	1
11	1011	0
12	1100	1
13	1101	0
14	1110	0
15	1111	1

Fig. 2. Binary codes and odd-parity digits for decimal numbers 0 through 15.

Binary-Coded Decimal. Suppose only the decimal digits, 0 through 9, are to be transmitted. From these 10 digits any positive decimal integer can be constructed. The 10 binary representations for these digits are known as *binary-coded decimal*, or BCD.

In BCD the codes for the numbers 10 through 15 decimal are not used and, if they show up at a receiver, they can be detected as "illegal" by checking the received datum with a "legal word" list. By selecting BCD over straight binary, a designer makes it possible to detect one type of error.

Excess-Three Code. By making a simple change to the code, more information—redundancy—can be built right in. The *excess-three* (X-3) code is obtained by adding binary 0011 to the BCD codes for the decimal digits 0 through 9. Figure 1 shows that X-3 does not allow the codes 0000 or 1111. Thus, every legal X-3 word contains at least *one* 0 and 1, providing another bit of information—that the data channel is active and transmitting.

Another property that makes it interesting for error-checking is that X-3 is a *self-complementary* code. That is, if each 0 of a legal X-3 word is changed to a 1, and each 1 to 0, the process generates the 9's complement of the word. This feature makes error-checking in X-3 easier and statistically more reliable than BCD codes. In a BCD error-checking algorithm, for example, a minimum of *six* comparisons and table look-ups must be made before an error can possibly be detected.

BCD is a *weighted positional* code; in which the position of each bit in the data word determines its value. X-3 is not a weighted positional code. By counting 1's and 0's in each position of the X-3 representation of the digits 0 through 9, it can be seen that for a given X-3 word, each position has a 50% probability of being either 0 or 1. This eliminates any statistical bias in the code itself.

In an X-3 error-checking algorithm, with judicious use of the "complement" function (a very fast and very easy operation in most microprocessors), the algorithm can detect a bad code in six comparisons but only *two* memory look-ups. In some cases, the error checking can be done 100% *faster* by use of the X-3 code for data transmission.

As a further bonus, the X-3 code makes keeping track of carries and borrows in addition and subtraction of coded decimal digits significantly simpler than in the straight BCD code.

Gray Code. Turning to a completely different four-bit code, the *Gray code* finds an application in many analog-to-digital data-transmission systems. The Gray code's most significant feature is that, going from one number to another with a difference of only one, only a single bit changes in the Gray code. In a system where the analog signal is expected to change slowly with respect to the sample frequency (say, for example, temperature inside a house is being monitored and encoded), a change in more than one digit position would automatically signal an error to the receiver.

Some information—more redundancy—about the data itself is reflected in the code itself. The odd decimal digits (1, 3, 5 etc.) have Gray-code equivalents that contain an *odd* number of 0's and 1's. This extra information is designed right into the code system itself, taking advantage of its most likely application.

So far, discussion has been restricted to four-bit codes. This has been done keeping in mind the 8-bit data bus structures of microprocessors such as the 8008, 8080, or 6502. With more complicated codes, however, it is possible to add an extra bit to the data word to contain extra information about the data.

Parity. One of the more widely used complicated coding systems is called *parity checking.* Parity checking simply counts the number of 0's or 1's in a data word and assigns a value to an extra *parity bit*, depending on the result (Fig. 2). Thus a seven-bit code such as ASCII (American Standard Code for Information Interchange) might be transmitted in an eight-bit format, with one parity bit.

The *odd parity* system adds a 1 to a data word so that it always has an *odd* number of 1's in it. *Even* parity adds a 1 to cause an even number of 1's. Odd parity has a slight advantage, similar to excess-three, because every code word has at least one 1 or 0 in it, providing a verification of data-channel operation.

Note that with the parity system, if *two* errors occur in the same data word, parity check will not detect the error! But it will detect a three-bit error, or an error in any odd number of different bits.

BIT POSITION	01	02	03	04	05	06	07
DESIGNATION	C_1	C_2	D_3	C_3	D_2	D_1	D_0
VALUE	0	1	1	0	0	1	1

DECIMAL 11 = 1011 BINARY

C_1 = mod 2 sum $(D_0+D_2+D_3)$ = (1+0+1) = 0

C_2 = mod 2 sum $(D_0+D_1+D_3)$ = (1+1+1) = 1

C_3 = mod 2 sum $(D_0+D_1+D_2)$ = (1+1+0) = 0

Fig. 3A. How Hamming-code check digits are generated.

	01	02	03	04	05	06	07
	C_1	C_2	D_3	C_3	D_2	D_1	D_0
RECEIVED VALUE	0	1	1	0	0	1	1

C_1' = mod 2 sum $(C_1+D_0+D_2+D_3)$ = (0+1+0+1) = 0

C_2' = mod 2 sum $(C_2+D_0+D_1+D_3)$ = (1+1+1+1) = 0 (C_3', C_2', C_1') = (0,0,0)

C_3' = mod 2 sum $(C_3+D_0+D_1+D_2)$ = (0+1+1+0) = 0 NO ERROR

Fig. 3B. Check digits show that word was transmitted correctly.

	01	02	03	04	05	06	07
	C_1	C_2	D_3	C_3	D_2	D_1	D_0
RECEIVED VALUE	0	1	1	0	0	0	1

C_1' = mod 2 sum $(C_1+D_0+D_2+D_3)$ = (0+1+0+1) = 0

C_2' = mod 2 sum $(C_2+D_0+D_1+D_3)$ = (1+1+0+1) = 1 C_3', C_2', C_1' = (1,1,0)

C_3' = mod 2 sum $(C_3+D_0+D_1+D_2)$ = (0+1+0+0) = 1 ERROR IN BIT 6

Fig. 3C. The check digits show the error is in bit 6.

	01	02	03	04	05	06	07
	C_1	C_2	D_3	C_3	D_2	D_1	D_0
RECEIVED VALUE	0	0	1	0	0	1	1

C_1' = mod 2 sum $(C_1+D_0+D_2+D_3)$ = (0+1+0+1) = 0

C_2' = mod 2 sum $(C_2+D_0+D_1+D_3)$ = (0+1+1+1) = 1 (C_3', C_2', C_1') = (0,1,0)

C_3' = mod 2 sum $(C_3+D_0+D_1+D_2)$ = (0+1+1+0) = 0 ERROR IN BIT 2

Fig. 3D. The check digits show an error in bit 2, proof that even the error-protect digits are safe from error.

Multiple Errors. So far, the code structures and error-checking systems have provided enough redundancy to provide for some manner of single-error checking. But all fail pretty badly at detecting more than one error per datum.

Is single-error detection enough? Statistics (and the binomial theorem) say that for an error rate of one in 1000 transmitted bits, the odds of having two erroneous bits in a 5-bit word are one in 100,000. So, the single-error approximation seems a realistic, although simplistic, choice.

The amount of redundancy in an error-correcting code must be much higher than in a mere error-detecting code. Mathematicians, using such esoteric items as group theory, vector spaces and cyclic codes, have come up with a whole flock of codes that contain information about themselves, but most of them are best applicable when large bit-strings are processed.

Hamming Code. One of the simplest error-correcting codes is called the *Hamming code*, after its developer. The Hamming system for a four-bit datum, for example, generates three check bits, and the data must be transmitted in a particular manner (that is, the check and data bits interspersed in a particular manner) in order for the decoding and verification to correctly occur.

Suppose the decimal number 11 is to be transmitted. The binary representation is arranged in the indicated positions of the data word, in Fig. 3A and the check digits are generated by following the rules shown. The *modulo 2 sum* is the remainder (0 or 1) after adding a string of binary digits and disregarding any carry operations. The check digits go into the positions shown.

The resulting seven-bit word is now transmitted. The error-checking and results for a word transmitted with no error is shown in Fig. 3B.

Figure 3C shows the seven-bit word received with an error in data digit D1 in bit position 6. Note that, when the verification digits (C_3', C_2', C_1') are arranged as a three-bit binary number, they point right to the erroneous bit—the number 110 in binary is 6 decimal.

Figure 3D shows the seven-bit word received with an error in check digit C_2. The verification digits now form the binary number 010, bit position 2, where the error is!

Again, note that if two errors occur, it is impossible to completely reconstruct the data word with this given scheme. Multiple-error-correction codes do exist, however. ◇

CREATIVE RECORDING WITH 4-CHANNEL TAPE RECORDERS

How to achieve special effects such as echo, reverb, and balancing for stereo level and position.

BY LEONARD FELDMAN

ACCORDING to several audio dealers surveyed recently, one of the hottest items in all of high-fidelity componentry is the 4-channel, open-reel tape deck. Of course, sales of these multi-track machines don't even come close to those of better stereo cassette decks, but when you consider the fact that reasonably good 4-channel decks sell for around $600.00 and up, as opposed to Dolby-equipped cassette machines that can be had for as little as $200.00 or sometimes less, consumer interest in the open-reel format seems unusual.

Add to this the fact that very few recording companies offer even a meager selection of pre-recorded 4-channel programming on open-reel tapes and the sudden interest in these expensive machines becomes even more puzzling. Surely, owners of 4-channel, open-reel decks are not spending that kind of money simply to transcribe their newly acquired CD-4 or matrix 4-channel records onto tape, although of course that is one application for these quadraphonic recorders.

A clue to the most popular usage of these machines was uncovered by further questioning of dealers and by thumbing through some of the recent product offerings from manufacturers who normally concentrate on such conventional products as tuners, amplifiers, and receivers. They are now offering such "odd-ball" products as mixers and portable mixing consoles for consumer use. These include Shure, Teac, Sony, and others. Some of their mixers have six or more input channels and up to four output-channel facilities.

In addition, we found that microphone sales are better than ever at the consumer level, and we don't mean single microphone purchases to replace the original-equipment models supplied with cassette decks. We're talking about good dynamic and condenser microphones that sell from $50.00 and up. These are finding their way into home hi-fi systems in increasing numbers, as are separate Dolby noise-reduction systems, compress-expand systems, and others. From all this sales activity, we concluded that the big 4-channel decks aren't necessarily being used to record or play 4-channel programming at all! They are forming the basis of thousands of "home recording studios," often capable of turning out master tapes that rival some of the products made by professional studios.

"Sel-Sync" Makes the Difference. At least five makes of multi-track tape decks sold to consumers have an important built-in feature that enables users to employ some of the same techniques used in recording popular music. While just about any stereo or 4-channel deck is equipped with three tape heads (erase, record, and playback), the physical position of these heads in relation to tape travel is normally that shown in Fig. 1. The tape passes across the erase head first, where any previously recorded material is erased. Desired new program information is then recorded onto the tape as it passes in front of the record head and, a fraction of a second later, the newly recorded program can be "monitored" by the playback head and the playback preamplifier associated with that head.

This is a fine arrangement for making ordinary stereo or even 4-channel recordings, since it enables the operator to hear his recorded results (either via phones or through his speaker system) just a short time after the recording occurs. If he hears distortion, over-recording, or under-recording, he can correct control settings *almost* instantly. The delay is determined by the distance between the record and playback heads (in inches) divided by the tape speed (in inches-per-second). The faster the tape speed, the shorter the delay.

Suppose, however, that you wanted to record one tape track at a time, adding other tracks later. You might want to record the singing of a "one man quartet"—in which you or a talented friend provide all four harmonizing vocal parts