4.3 - REDUNDANCY

4.3.1 - Principles of Redundancy

Redundancy is a measure of the proportion of a message that is unnecessary. In languages, it is a function of the statistical rules of the language such as letter probabilities, digrams, trigrams, etc. For example, if a message were received as follows:-

> 'An e ectr ni en neer must ac pt the f ct th t he needs
> to ke p in to ch wit n w d vel opm nts in a rap dl
> chan in subject"

it would not be difficult to complete the message in the position where no letter had been received, presumably due to noisy conditions.
The reader has at his disposal:-

(i) vocabulary

(ii) the rules of the language

(iii) the gist of the message

The majority of the letters missed out could therefore be considered redundant as they do not really help with the understanding of the message.

Shannon and others have investigated the effect of the statistical rules of a language and their relation to redundancy. For example, in the case of English, if all the letters of the alphabet are equiprobable, the average information per letter is $\log_2 26 = 4.7$ bits. If the relative frequency of the letters is taken into consideration, the average information is reduced to 4.065 bits. Finally, when all the rules of the language have been considered, the average information is reduced to 2.6 bits i.e., there is an information loss of 2.1 bits. Comparing this figure with the maximum information that could be obtained if all the letters were equiprobable, it is seen that only approximately 50% of the maximum information is actually obtained, the remaining 50% being redundant.

A definition of redundancy is:-

$$\text{Redundancy} = \frac{\text{Maximum information - actual information}}{\text{Maximum information}}$$

$$\frac{4.7 - 2.6}{4.7}$$

= 45% in the case of English.

4.3.2 - Minimum Redundancy Codes

A code can be derived which removes all the redundancy by obeying the following rules:-

a. all the combinations of the code are used from the shortest up,

b. the shortest combinations are allocated to the most commonly occurring letters,

c. no synchronisation or spacing is needed between each code group. This condition requires that no code group should commence with all or part of another code group.

Huffman (1952) proposed such a code and Shannon and Fano constructed the code for English. The principle of the construction can be seen from the following simple example.

	Symbol	Probability	Code
X_1	a	0.25	00
	b	0.25	01
	c	0.125	100
	d	0.125	101
X_2	e	0.0625	1100
	f	0.0625	1101
	g	0.0625	1110
	h	0.0625	1111

a, b, c, d, e, f, g and h are a sequence of symbols whose probabilities are known. The symbols are written down in descending order of probabilities and divided into subsets as shown. All codes in subset X_1 commence with 0, while those in subset X_2 commence with 1. A code tree can now be drawn which obeys the three rules.

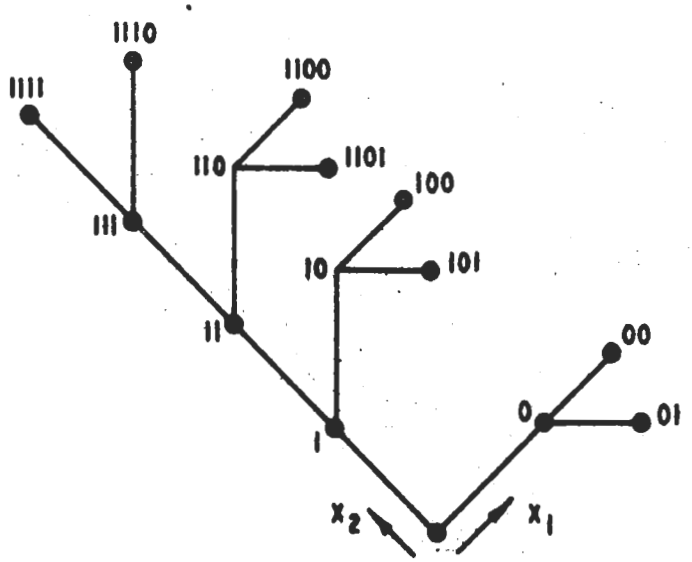

Fig. 4.3 - Minimum Redundant Code

Suppose the sequence of symbols 'dfah' were transmitted; this would appear as 1011101001111 at the output of the encoder. The receiver inspects the first digit and finds it to be a 1. It therefore moves up the left branch of the tree to position 1. The next digit is inspected and found to be 0. The point 10 has now been reached. Two alternatives are possible and on inspection of the third digit, the receiver moves finally to the position 101. It knows that there can be no further digits in the code group corresponding to the first symbol and it will print the symbol d. The difficulties arise when there is an error introduced. For example, if the received code were 10101101001111, i.e. an error in the fourth position, the receiver would go through the decoding process as before and the first symbol printed would be d. The next symbol code begins with a 0 followed by a 1; this corresponds to symbol b. This process continues and a completely garbled message is received as follows.

Transmitted	101	1101	00	1111	
	d	f	a	h	
Received	101	01	101	00	1111
	d	b	d	a	h

This shows that the introduction of a single error into the code will produce complete corruption.

The Shannon Fano code is shown below in full.

E	100	F	01011	K	11011000
T	001	C	01010	X	1101100111
A	1111	M	01001	J	1101100110
O	1110	U	01000	Q	1101100101
N	1100	G	00001	Z	1101100100
R	1011	Y	00000		
I	0111	P	110111		
S	0110	W	101001		
H	0001	B	101000		
D	11010	V	1101101		
L	10101				

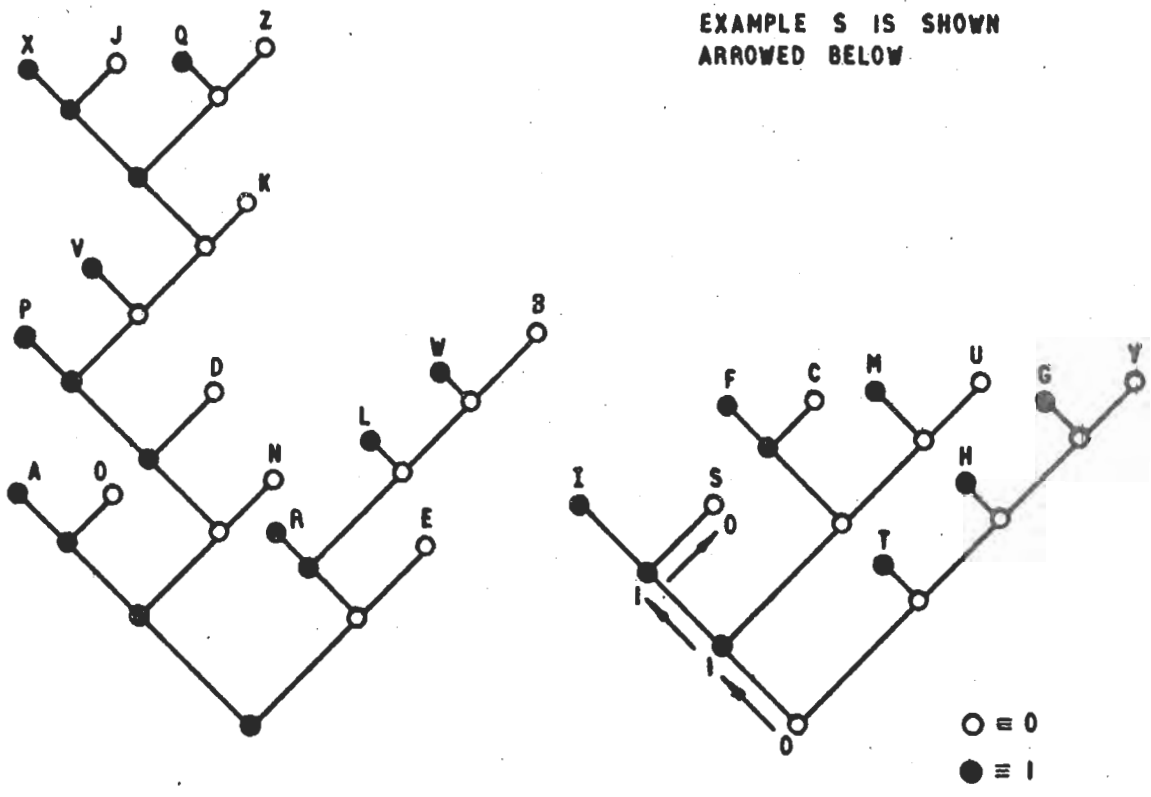

Fig. 4.4 - Shannon - Fano Code

4.4 - ERROR DETECTION AND CORRECTION CODES

4.4.1 - Principles of Error Detection and Correction

The codes so far referred to cannot be error checked, and if bits become erroneously changed, say because of circuit malfunction, there is no way to detect the error. This is because there are cases of two coded characteristics differing only in one bit position. For example, in the Murray Code, "NO" is transmitted as 00110 00011. If the received signal is 00111 00011, this is translated as MO, and there is no way of checking this error.

The distance between two coded characters is the number of bits that must change in one character so that the other character results. In the above example the distance between 00110 and 00011 is 2, since two bits must be changed to transform one character to the other. The minimum distance of a code is the minimum number of bits that must change in a coded character so that another valid character results. In the Murray Code the minimum distance is 1 since there is at least one case in which a coded character could be changed to another by changing only one bit. For example D (10010) becomes E (10000) by changing the fourth bit.

The minimum distance, the number of bits in error that can be detected, and the number of bits in error that can be corrected are connected by the formula

$$M - 1 = C + D$$

where M = minimum distance of the code
C = number of bits in error that can be corrected
D = number of bits in error that can be detected

C cannot be greater than D, since no error can be corrected without being detected. The table in figure 4.5 shows the relation between M, C and D for values of M up to 6.

M	C	D
1	0	0
2	0	1
3	0	2
	1	1
4	0	3
	1	2
5	0	4
	1	4
	2	3

Fig. 4.5 - Relation between M, C and D

An error detection code is defined according to the maximum error it will ALWAYS detect. Thus a code which detects all single, double and triple errors, and some quadruple errors is called a triple error detecting code.

Similarly an error correction code is defined in the same manner, that is according to the maximum error it will ALWAYS correct.

4.4.2 - The Seven Unit Code

For a code to have an error detecting and correcting facility it must have:-

a. redundancy

and b. code groups of equal length.

One such code that satisfies these conditions is the 7 unit or Van Duuren code. In this code, each letter is represented by seven digits of which 3 are always mark (1) and four are always space (0). There are a possible 128 combinations of seven digits but only 35, i.e. $^{7}C_{3}$ which satisfy the 3:4 mark to space ratio. Thus, there is a redundancy introduced.

$$\text{Redundancy} = \frac{\log_2 128 - \log_2 35}{\log_2 128}$$

$$= 28\% \text{ approximately.}$$

Note that the redundancy of the 7 unit code is less than the redundancy of English.

In the receiver, a simple device counts the number of marks (1) in each group. If the number is not three, an error is indicated. This indication can take one of many forms; it can cause a bell to be rung, completely inhibit the reception of any further groups until the error has been rectified, print an error symbol, or initiate a sequence of events designed to correct the message by asking for a repeat as in the ARQ system, see Section 4.4.4.

Letter Case	Figure Case	Code
A	-	0011010
B	?	0011001
C	:	1001100
D	Who are you?	0011100
E	3	0111000
F	%	0010011
G	@	1100001
H	£	1010010
I	8	1110000
J	Bell	0100011
K	(	0001011
L	)	1100010
M	.	1010001
N	,	1010100
O	9	1000110
P	0	1000011
Q	1	0001101
R	4	1100100
S	!	0101010
T	5	1000101
U	7	0110010
V	=	1001001
W	2	0100101
X	/	0010110
Y	6	0010101
Z	+	0110001

Carriage Return	1000011
Line Feed	1011000
Figure Shift	0100110
Letter Shift	0001110
Space	1101000
Error Symbol	0000111
Idle Beta (Continuous Stop)	0101100
Idle Alpha (Continuous Start)	0101001
RQ Signal	0110100

Fig. 4.6 - The Seven Unit Code

There is no requirement to introduce start and stop elements into the 7 unit code as it is a synchronous code.

Thus, the advantages of the 7 unit code over the 5 unit code are:-

a. redundancy is introduced thereby enabling error detection to be effected.

b. The 7 unit code is a synchronous code and requires no start or stop elements.

c. Three extra facilities are available over the 32 possible in the 5 unit code. These are Idle Alpha, Idle Beta and the RQ signal. In addition, provision can be made for an error symbol.

4.4.3 - Parity Checks

Another simple method of introducing an error detecting facility is the Parity Check. This involves the introduction of an extra digit onto the end of a 5 unit code which always ensures that the number of 1's in the complete code group is even. This is called an Even Parity Check. It is evident that a measure of redundancy has been introduced for the complete code group is now a 6 unit code.

$$\text{Redundancy} = \frac{6 - 5}{6}$$

$$= 17\% \text{ approximately.}$$

The redundancy can be reduced by increasing the number of digits in the code but this will eventually lead to an increase in the probability of a second error about which the simple parity check can do nothing.

If n = total number of digits in a code group and p is the probability of an error for any one digit, the probability of a single error in the group is

$$P(1) = {}^{n}C_{1}\, p(1 - p)^{n-1}$$ i.e. the Binomial Distribution.

4.4.4 - The ARQ System

Having detected an error, it is necessary to correct it. The simplest and cheapest way is to allow the decoder to produce the decoded error and then to use one's knowledge of the language to make some sense out of the message. Of course, this can only be effective when the error frequency is small, as will be appreciated by those who have had to make sense from A.G.Ms. over long distance communication links. More sophisticated methods of automatic correction are however available. These examine the code received and perform certain operations upon it to check for an error. One example is the A.R.Q. system used in the Shore Wireless Networks.

The A.R.Q. system works on the principle that the receiver inspects the incoming 7 unit code to check the mark to space ratio. If this ratio is incorrect, a request for a repeat is automatically sent to the transmitter which then repeats the last four characters prior to the RQ (request for repeat) signal being received.

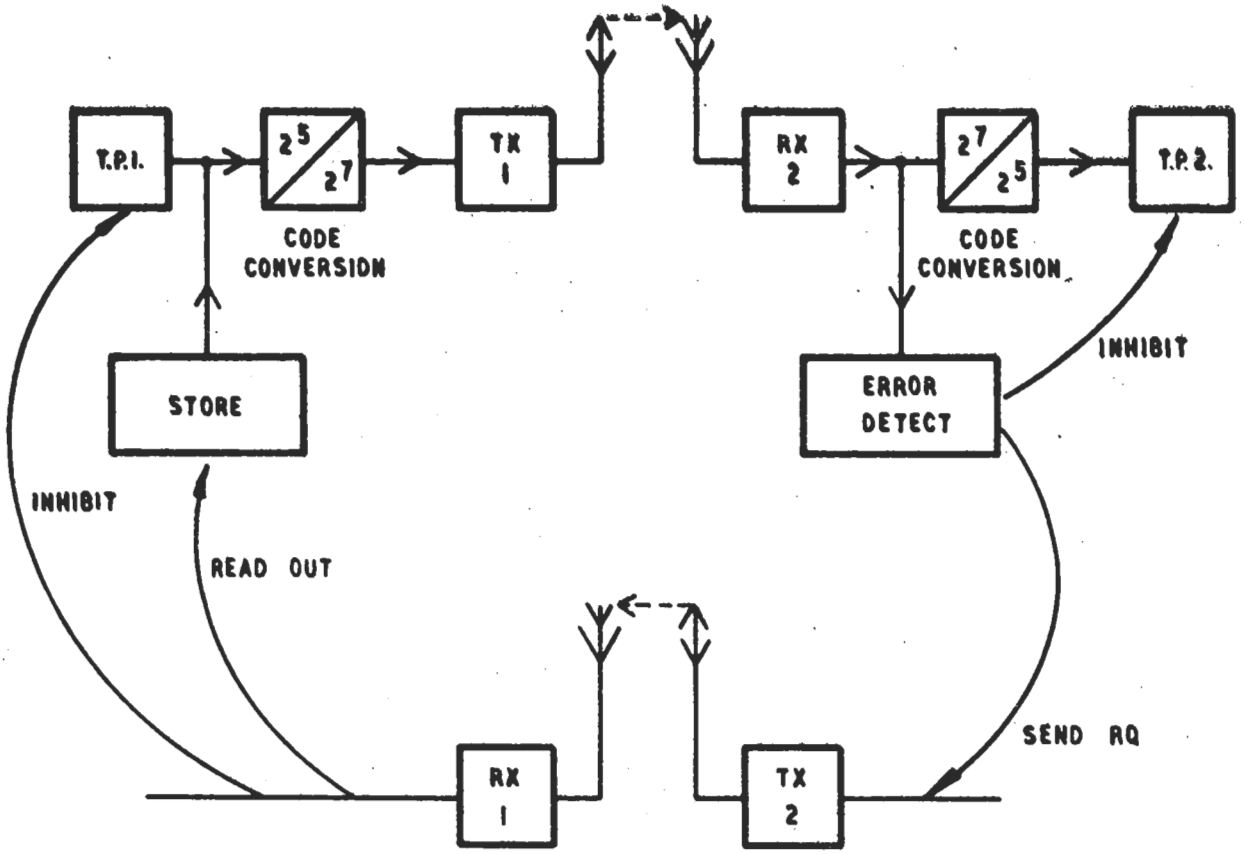

Fig. 4.7 - The A.R.Q. System

The block diagram in figure 4.7 shows a two way system which is passing messages in both directions simultaneously. If Receiver 2 receives an error, the error detection unit inhibits the Teleprinter 2 and causes Transmitter 2 to send the RQ signal. This is received by Receiver 1 which inhibits Teleprinter 1 and causes the store to be read out into Transmitter 1. Thus, there is a repeat of the last four characters which should in most cases be sufficient to correct any error.

Suppose that there is a path delay time of one element.

Tx 1	A	B	C	D	E	F	RQ	C	D	E	F	G	H
Rx 1		P	Q	R	S	RQ	P	Q	R	S	T	U	V
TP 1			P	Q	R	S					T	U	V
Tx 2	P	Q	R	S	RQ	P	Q	R	S	T	U	V	W
Rx 2		A	B	?	D	E	F	RQ	C	D	E	F	G
TP 2		A	B						C	D	E	F	G

There are two main drawbacks to this system. The first and most important is that in really bad conditions the system will cycle, i.e. it will be constantly asking for a repeat and no traffic will pass. The second one is that an error received on one channel will interrupt the other channel and hence reduce the flow of traffic.

It should be noted that the 5 unit code is used in the teleprinters, with a transfer from 5 to 7 unit code on the channel.

4.4.5 - Hamming Code

Another method of error correcting was devised by Hamming in 1950. This has been extended by Marconi in the form of the Autospec Code devised for the banks and is under consideration for RN use. Both codes have the same fundamental principles in that a form of multiple parity check is used.

To construct the Hamming Code, m of the n available positions are assigned as information positions. m is usually fixed and in the example that follows will be five. The remaining k positions are used as parity checks, the values (i.e. either 1 or 0) being determined by the encoding process.

It is found that there is a relationship between m and n such that

$$2^m \geq \frac{2^n}{n+1}$$ from which the following table can be made.

n	m	k
1	0	1
2	0	2
3	1	2
4	1	3
5	2	3
6	3	3
7	4	3
8	4	4
9	5	4
10	6	4
		etc.

Suppose a code symbol has been received. The k parity checks are applied in order and for each time the parity is satisfied a 0 is written down while for each time the parity is not satisfied, a 1 is written down. When written from right to left, the sequence of 0 and 1 obtained from checking may be regarded as a binary number and is called the checking number.

The positions over which each of the various parity checks are to be applied must now be determined. Since the checking number is to give the position of any error in a code symbol, any position which has a 1 on the right of its binary representation must cause the first check to fail.

Examining the binary form of various integers.

First check	Second check	Third check
1 = 1	2 = 10	4 = 100
3 = 11	3 = 11	5 = 101
5 = 101	6 = 110	6 = 110
7 = 111	7 = 111	7 = 111
9 = 1001	10 = 1010	12 = 1100
etc	etc	etc

The above tables show that the first check will check positions 1, 3, 5, 7, 9; the second check positions 2, 3, 6, 7, 10; the third check positions 4, 5, 6, 7, 12, 13, 14, 15; the fourth check positions 8, 9, 10, 11, 12, 13, 14, 15, 24 etc.

The following table shows the positions of the check numbers and the positions checked.

Check Number	Check Position	Positions Checked
1	1	1, 3, 5, 7, 9, 11, 13 .....
2	2	2, 3, 6, 7, 10, 11, 14 .....
3	4	4, 5, 6, 7, 12, 13, 14, 15 .....
4	8	8, 9, 10, 11, 12, 13, 14, 15, 24.

Having formulated the rules, apply them to a five unit code i.e. m = 5. From a previous table it will be seen that k = 4 giving a total code length of 9. The code is as follows and to illustrate its formation, consider the number 24.

Decimal	1 k	2 k	3 m	4 k	5 m	6 m	7 m	8 k	9 m	10 P
1	1	0	0	0	0	0	0	1	1	1
2	1	1	0	1	0	0	1	0	0	0
3	0	1	0	1	0	0	1	1	1	1
4	0	1	0	1	0	1	0	0	0	1
5	1	1	0	1	0	1	0	1	1	0
6	1	0	0	0	0	1	1	0	0	1
7	0	0	0	0	0	1	1	1	1	0
8	1	0	0	1	1	0	0	0	0	1
9	0	0	0	1	1	0	0	1	1	0
10	0	1	0	0	1	0	1	0	0	1
11	1	1	0	0	1	0	1	1	1	0
12	1	1	0	0	1	1	0	0	0	0
13	0	1	0	0	1	1	0	1	1	1
14	0	0	0	1	1	1	0	0	0	0
15	1	0	0	1	1	1	1	1	1	1
16	1	1	1	0	0	0	0	0	0	0
17	0	1	1	0	0	0	0	1	1	0
18	0	0	1	1	0	0	1	0	0	1
19	1	1	1	1	0	0	1	1	1	1
20	1	0	1	1	0	1	0	0	0	0
21	0	0	1	1	0	1	0	1	1	1
22	0	1	1	0	0	1	1	0	0	0
23	1	1	1	0	0	1	1	1	1	1
24	0	1	1	1	1	0	0	0	0	0
25	1	1	1	1	1	0	0	1	1	1
26	1	0	1	0	1	0	1	0	0	0
27	0	0	1	0	1	0	1	1	1	1
28	0	0	1	0	1	1	0	0	0	1
29	1	0	1	0	1	1	0	1	1	0
30	1	1	1	1	1	1	1	0	0	1
31	0	1	1	1	1	1	1	1	1	0

The binary representation of this number is 11000. Reading from left to right, these digits are allocated positions 3, 5, 6, 7 and 9.

First Parity Check

This examines positions 1, 3, 5, 7, 9. There are two 1's and the check is 0.

Second Parity Check

This examines positions 2, 3, 6, 7. There is only one and the check is 1.

Third Parity Check

This examines positions 4, 5, 6, 7. There is only one 1 and the check is 1.

Fourth Parity Check

This examines positions 8 and 9. There are no 1's and the check is 0.

After encoding in this manner, the coded number 24 becomes

0	1	1	1	1	0	0	0	0	0
									P

On receipt of this code group, the receiver performs the same process as the encoder

First check	0	
Second check	0	
Third check	0	Check number 0000
Fourth check	0	i.e. no error

Suppose that an error had occurred such that the received group was

0	1	1	1	1	1	0	0	0	0
									P

There is an error in the sixth position.

First check	0	
Second check	1	Check number 0110
Third check	1	i.e. an error in the
Fourth check	0	sixth position

In order to detect a double error, a further parity check P is made. If now the number 24 had been received as

0	0	1	1	1	1	0	0	0	0

which has errors in the second and sixth position, the receiver would first go through the process of checking and correcting

First check	0	
Second check	0	
Third check	1	Check number 0100
Fourth check	0	i.e. an error in the 4th position

After correction, the group now reads

0 0 1 0 1 1 0 0 0 0

But, the final parity check, the tenth position, will indicate that there is still an error and the receiver will reject the character.

Thus, it is evident that to have the facility for single error detecting and correcting as well as double error detecting, 5 check digits are required if the inform ation is contained in a five unit code. This gives a redundancy of 50%. Furthermore, if the number of information digits were increased to say 6, the check digits including parity would have to be 5. This is a reduction in redundancy but the more digits there are in a code group, the greater is the probability of a second error and in some instances, the probability of a third error is increased.

Consider now the Marconi ten unit code. This has the same facilities as the Hamming code and also has the same redundancy, for only five of the ten digits are used for information, the remaining five being used for checking.

The advantage of the Marconi system is that it is much simpler to construct than the Hamming code inasmuch as the rules are simpler. The rules are as follows:-

a. if the number of 1's in the information group is ODD, the check characters are a repeat of the information characters,

b. if the number of 1's in the information group is EVEN, the check characters are an inverted repeat of the information group.

e.g. 10011 10011

10010 01101

Suppose the following code group were transmitted and received

1001001101

The receiver examines the first five digits and counts the number of 1's in the group. In this case, it finds there are two 1's in the group and will invert the remaining check digits.

Information digits	10010	
Inverted check digits	10010	
	00000	on comparison.

Thus a series of 0's indicates there has been errorless reception. If there had been an error in the information digits and the following were received

1101001101

the receiver would go through the same process as before for it has no prior knowledge than an error has been received.

Information digits	11010	odd
Repeated check digits	01101	
	10111	on comparison.

Note - comparison is effected by binary subtraction without a carry. The comparison gives a 0 in the second place in a row of 1's. This indicates an error and its position.

If there had been an error in the check digits, for example

1001001001

the receiver would go through the same process.

Information digits	10010	even
Inverted check digits	10110	
	00100	

The comparison gives a 1 in a row of 0's indicating that there is an error in the check digits, the position of the 1 indicating the position of the error.

Finally, if a double error occurs

1101011101

Information digits	11010	odd
Repeated check digits	11101	
	00111	on comparison.

There are two 0's and three 1's indicating a double error. Any combination of 0's and 1's excluding the case of one 0 and four 1's or one 1 and four 0's will indicate a double error. The printer at the receiving end will then print an error symbol.