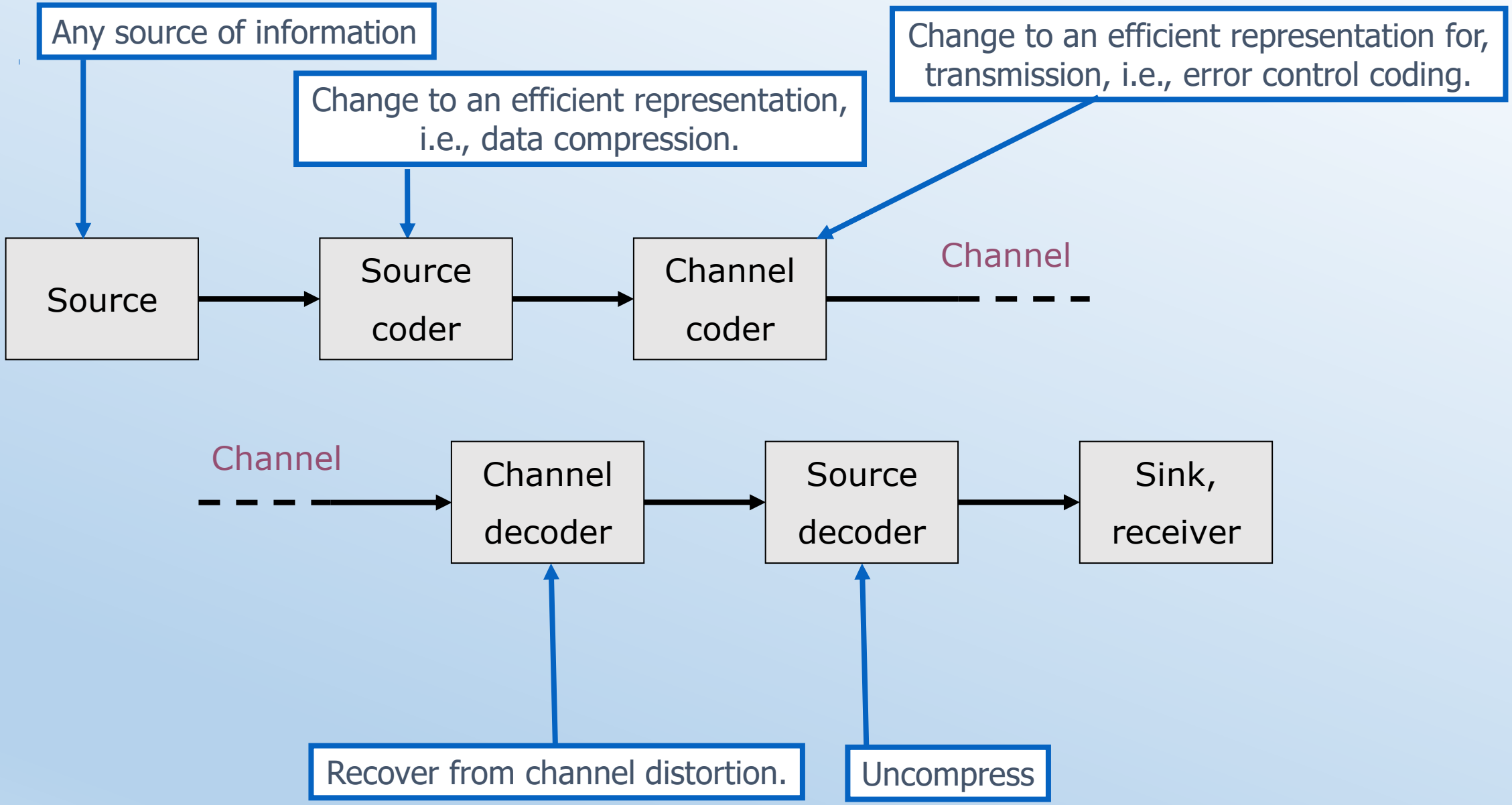
Linear Block Codes part1

Contents:

Communication block diagram
Types of codes,
linear and systematic
Encoding
Hamming Distance, Hamming weight
Syndrome calculation, decoding, etc
Standard array

Shivani M Deosthale



The channel is anything transmitting or storing information –
a radio link, a cable, a disk, a CD, a piece of paper, ...

Transmission through noisy channel

* Transmission errors can occur, 1's become 0's and 0's become 1's
* To correct the errors, some redundancy bits are added to the information sequence, at the receiver the correlation is exploited to locate transmission errors
* Here, only binary transmission is considered.
*Here we try to optimally add this redundancy to the information bits.
*Codes which allow only error detection are called error detecting codes and codes which allow error detection and correction are called error detecting and correcting codes

A parity bit is used for the purpose of detecting errors during transmission of binary information. A parity bit is an extra bit included with a binary message to make the number of 1s either odd or even.

The message including the parity bit is transmitted and then checked at the receiving end for errors.

- An error is detected if the checked parity does not correspond with the one transmitted.
- The circuit that generates the parity bit in the transmitter is called a parity generator and the circuit that checks the parity in the receiver is called a parity checker.
- In even parity the added parity bit will make the total number of 1s an even amount. In odd parity the added parity bit will make the total number of 1s an odd amount.
- As a general rule in the digital system where the transmission system is relatively short, it may be assumed that probability of a single-bit error is small and that of a double-bit error and higher order errors is extremely small.

Types of codes:

Block codes(n,k) , Convolutional codes

Convolutional codes(n,k,m): Here each encoded message block depends upon the corresponding k bit message block as well as on the previous m message blocks. Thus encoder has a memory order of m. This set of encoded sequences produced by a k-input, n-output encoder of memory of order m is known as (n,k,m) convolutional code.

Since it is memory based so is implemented with sequential logic circuit.

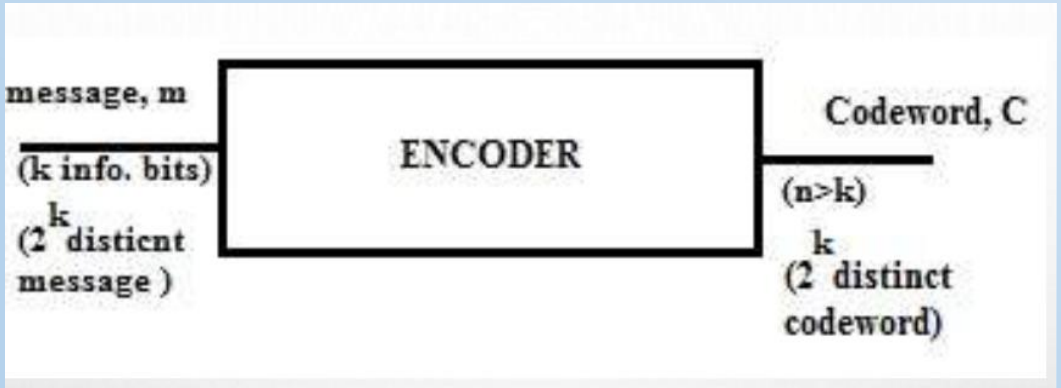
code rate = k/n , by keeping same code rate, redundancy can be added by increasing m

Block codes:

set of words having a well defined mathematical property, where each word is a sequence of a fixed number of bits.

K: message bits, n: number of coded bits , (n-k=r parity bits to be added to message bits to form coded bits)

Linear Block codes



The encoder generates a block of n coded bits from k message bits and we shall call this as (n,k) block code.

Linear block code: if a modulo 2 addition of two code words is also a valid code word then the block code is linear.

K bit message will have 2^k message words and n bit possible codewords will be 2^n.

Systematic codes:

*If in all the codewords we can find exactly the corresponding information sequence, the code is called systematic. It is convenient to group all these bits either at the end or at the beginning of the code word.

* In this case the generator matrix G can be divided into two sub matrices [P | I]. Or G matrix can be written as [I | P]

$[G]_{kxn} = [P_{kx(n-k)} \mid I_{kxk}]$ or $[G]_{kxn} = [I_{kxk} \mid P_{kx(n-k)}]$

Where P is parity submatrix and I is identity submatrix. Identity submatrix places message bits in coded word systematically.

Encoding steps:

Coded word $[C]_{1xn} = [D]_{1xk} . [G]_{kxn}$

[D] is a 1xk row vector of k bits message word.

As information is in binary, the operations are modulo 2, i.e, addition corresponds to EXOR operation whereas multiplication is AND operation.

Modulo 2 addition (EXOR)	Modulo 2 multiplication(AND)
0⊕0=0	0.0=0
0⊕1=1	0.1=0
1⊕0=1	1.0=0
1⊕1=0	1.1=1

e.g. $[1\ 1\ 0] . \begin{vmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{vmatrix}$

$= [1.1 \oplus 1.0 \oplus 0.1 \ , \ 1.1 \oplus 1.1 \oplus 0.0, \ 1.1 \oplus 1.1 \oplus 0.1]$

$= [1 \oplus 0 \oplus 0 \ , \ 1 \oplus 1 \oplus 0 \ , \ 1 \oplus 1 \oplus 0]$

$= [1 , 0, 0]$

Example 1: Find code words.

Coded word $[C]_{1xn} = [D]_{1xk} . [G]_{kxn}$

$$\begin{bmatrix} C \end{bmatrix} = \begin{bmatrix} D1 & D2 & D3 & D4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

[C1,C2,C3,C4,C5,C6,C7] = [D1, D2, D3, D4, D1⊕D2⊕D3, D1⊕D2⊕D4, D1⊕D3⊕D4]

$(_2^k) = {}_2^4$ Codewords
= 16 codewords

D1	D2	D3	D4	C1	C2	C3	C4	C5	C6	C7	W(C)
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	1	1	1	4
0	0	1	0	0	0	1	0	1	0	1	3
0	0	1	1	0	0	1	1	1	1	0	4
0	1	0	0	0	1	0	0	1	0	0	2
0	1	0	1	0	1	0	1	1	0	1	4
0	1	1	0	0	1	1	0	0	1	1	4
0	1	1	1	0	1	1	1	0	0	0	3
1	0	0	0	1	0	0	0	1	1	1	4
1	0	0	1	1	0	0	1	1	0	0	3
1	0	1	0	1	0	1	0	0	1	0	3
1	0	1	1	1	0	1	1	0	0	1	4
1	1	0	0	1	1	0	0	0	0	1	3
1	1	0	1	1	1	0	1	0	1	0	4
1	1	1	0	1	1	1	0	1	0	0	4
1	1	1	1	1	1	1	1	1	1	1	7

Associated with generator matrix G , there is another matrix with order ((n-k)xn) matrix , called as Hamming matrix denoted as H

$$[H]_{(n-k)xn} = [\ [P^T]_{(n-k)xk} \ \mid \ [I]_{(n-k)x(n-k)}\ ]$$

For [G] = $\begin{matrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{matrix}$

I(kxk) **P(kx(n-k)**

Find [H] = $\begin{matrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{matrix}$

$[P^T]_{(n-k)xk}$ **[I(n-k),n-k)]**

Important Terms:

Hamming weight: Hamming weight of a binary n bit code word is defined as the number of non zero components i.e. number of ones and is denoted as w(c).

e.g. The Hamming weight of a codeword “0110011” is 4

The previous table has last column for Hamming weight of all valid code words for the given generator matrix.

Hamming Distance: It is the distance between the two code words C1 andC2 and is defined as the number of places in which they differ and is denoted as d(C1,C2)

e.g. if C1 is “

1	0	0	0	1	1	1

“ and

C2 is “

1	1	0	1	0	1	0

“

Then Hamming distance between them is d(C1, C2)= 4

Hamming distance and Hamming weight

Property 1: $d(C1, C2) + d(C2, C3) \geq d(C1, C3)$ ……. (triangle inequality)

Property 2: $d(v, w) = w(v + w)$

Minimum distance of block code

$d_{min} = \min_{C1,C2} \{d(C1,C2) : C1,C2 \in C \text{ and } C1 \neq C2\}$

If the code is linear: $d_{min} = \min\{w(C1 + C2) : C1,C2 \in C \text{ and } C1 \neq C2\}$

$= \min\{w(C3) : C3 \in C \text{ and } C3 \neq 0\}$

$= W_{min}$

Theorem: The minimum distance of a linear block code is equal to the minimum weight of its non zero code word.

Example 2. For a (7,4) linear block code, the parity check matrix P is given by

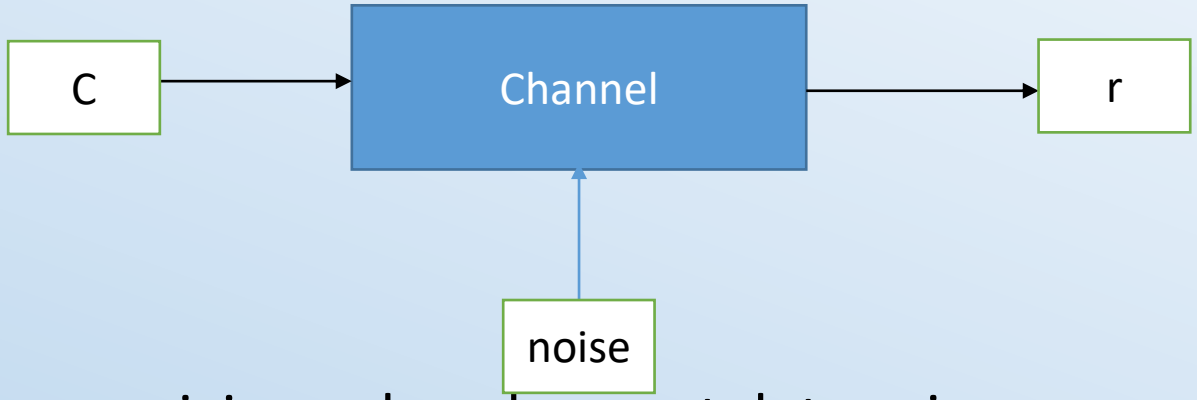
$$[P] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

a. Find G matrix, H matrix.
b. Find all code words.
c. Find Hamming weight for all code words.
d. Find Minimum Hamming weight for this set of code words.

Decoding

syndrome decoding: Consider (n,k) linear block code generated by generator matrix G and having corresponding Parity check matrix H. If C is the transmitted code word and r is the received code word at the received end of the noisy channel due to which r is different from C Transmitted.

C + e = r ; e is the error word or error vector.



Receiver is unaware of either C or e. Hence after receiving r, decoder must determine whether there is any error within r.

Calculation of Syndrome: S is called as Syndrome of r. if there is no error(i.e. e=0), then r=C (i.e. valid code word)and S=0. But in case of error , S≠0 as e ≠0

$$[S]_{(1x(n-k))} = [r]_{(1xn)}.\ [H]^T_{(nx(n-k))}$$
$$= [C+e].\ [H]^T$$
$$= C.\ [H]^T + e.\ [H]^T$$
$$= 0 + e.\ [H]^T = e.\ [H]^T$$

Example :3

Let C=(0101110) be the transmitted code and r=(0001110) be the received vector.

s=r. $H^T=(s_1, s_2, s_3)$

$$=(r_1, r_2, r_3, r_4, r_5, r_6, r_7)\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\\ 1 & 1 & 0\\ 0 & 1 & 1\\ 1 & 1 & 1\\ 1 & 0 & 1\end{bmatrix}$$

The syndrome digits are:

$$s_1 = r_1 + r_4 + r_6 + r_7 = 0$$
$$s_2 = r_2 + r_4 + r_5 + r_6 = 1$$
$$s_3 = r_3 + r_5 + r_6 + r_7 = 0$$

The error vector, $e=(e_1, e_2, e_3, e_4, e_5, e_6, e_7)=(0100000)$

$$C^* = r + e$$
$$= (0001110)+(0100000)$$
$$= (0101110)$$

where C^* is the actual transmitted code word

So The syndrome is not a function of the transmitted codeword but a function of error pattern.
So we can construct only a matrix of all possible error pattern with corresponding syndrome.

Syndrome Table Construction

There are 2^n possible received vectors.
There are 2^k valid code words.
There are 2^{n-k} possible syndromes
First we generate all the error patterns of length 1, and calculate the corresponding syndrome and put all of them into a matrix.
Then, we increase the error pattern weight and we do the same thing as before.
Each time if the new syndrome had already been saved, it will be thrown away and we continue with the next error pattern.
When all the 2k syndromes are found, the table is complete.
For the previous example (256,200) the size of the matrix is
256 = 7:2 1016, which is still too big.

Error-Detecting Capabilities of a Block Code

*If the minimum distance of a block code C is dmin, any two distinct code vector of C differ in at least dmin places.

*A block code with minimum distance dmin is capable of detecting all the error pattern of dmin – 1 or fewer errors.

*However, it cannot detect all the error pattern of dmin errors because there exists at least one pair of code vectors that differ in dmin places and there is an error pattern of dmin errors that will carry one into the other.

*The random-error-detecting capability of a block code with minimum distance dmin is dmin – 1 ;

Number of errors can be detected = dmin – 1;

Error-Correcting Capabilities of a Block Code

*An (n, k) linear code is capable of detecting $2^n - 2^k$ error patterns of length n.

*Among the $2^n - 1$ possible nonzero error patterns, there are $2^k - 1$ error patterns that are identical to the $2^k - 1$ nonzero code words.

- If any of these $2^k - 1$ error patterns occurs, it alters the transmitted code word v into another code word w, thus w will be received and its syndrome is zero.
- There are $2^k - 1$ undetectable error patterns.
- If an error pattern is not identical to a nonzero code word, the received vector r will not be a code word and the syndrome will not be zero.

- Number of errors can be corrected = (dmin -1)/2

- For Example 3, find how many errors can be detected and corrected?

The possible error combinations will be nCk = C(n,k)= n!(n−k)!×k!

Error patterns for (7,4) linear block code.

Since n= 7 , and 1 bit error , number of patterns = 7C1 = 7 ! / (1 ! x 6 !) = 7

N=7, 2 bit error, number of patterns = 7C2 = 7 !/(2 ! x 5 !) = (7 x 6)/2 =21 and ……so on……

Single and double error patterns shown below.

e1	e2	e3	e4	e5	e6	e7
1	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	1

e1	e2	e3	e4	e5	e6	e7
1	1	0	0	0	0	0
1	0	1	0	0	0	0
1	0	0	1	0	0	0
1	0	0	0	1	0	0
1	0	0	0	0	1	0
1	0	0	0	0	0	1
0	1	1	0	0	0	0
---- ---- ---- ----						
0	0	1	1	0	0	0
---- ----						
					1	1

Standard Array and Syndrome Decoding

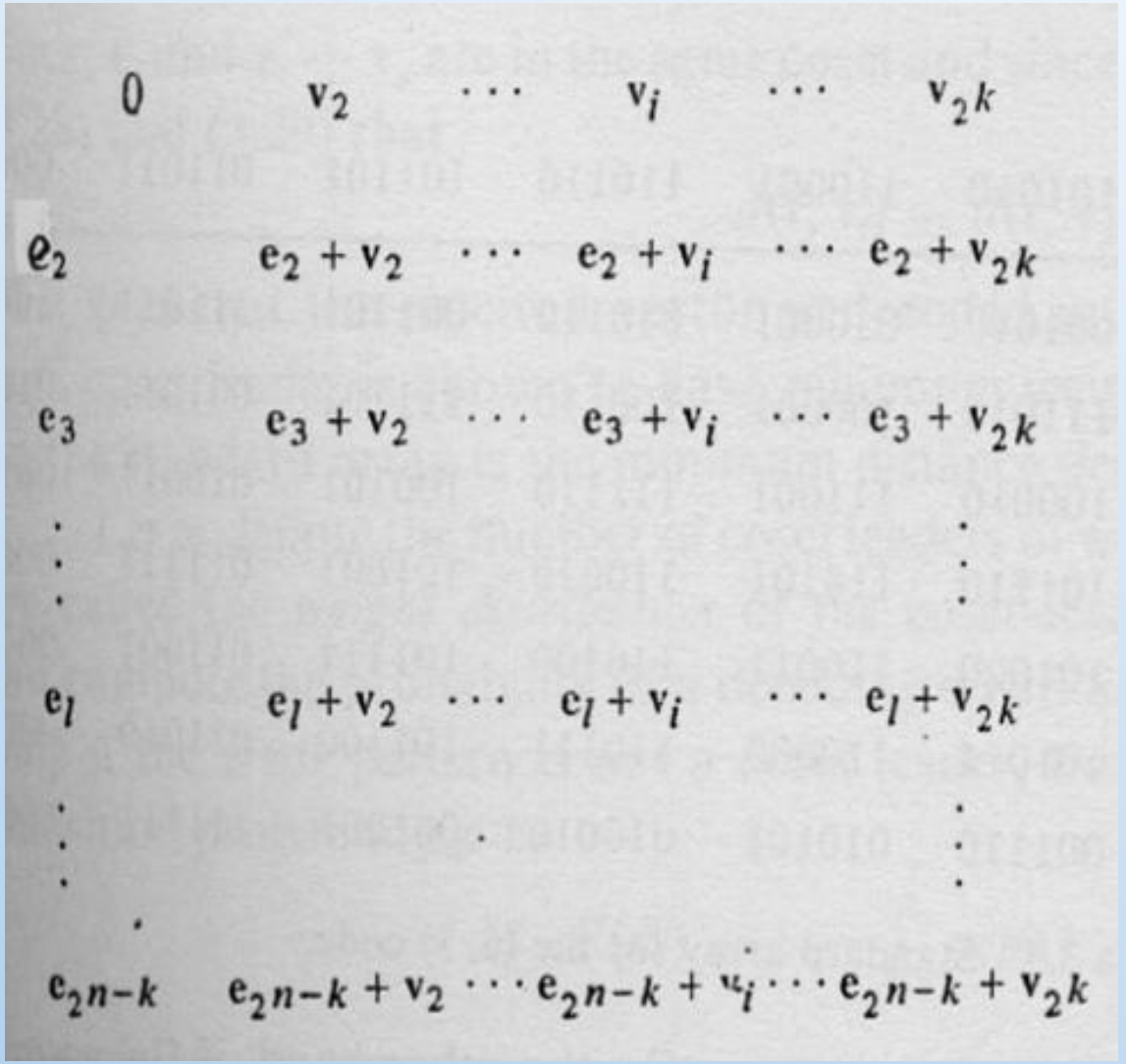
- Let $v_1, v_2, \ldots, v_2k$ be the code vector of C.
- Any decoding scheme used at the receiver is a rule to partition the 2^n possible received vectors into 2^k disjoint subsets $D_1, D_2, \ldots, D_2k$ such that the code vector v_i is contained in the subset Di for $1 \leq i \leq 2k$

- Each subset D_i is one-to-one correspondence to a code vector v_i
- If the received vector r is found in the subset D_i , r is decoded into v_i
- Correct decoding is made if and only if the received vector r is in the subset Di that corresponds to the actual code vector transmitted

- A method to partition the 2^n possible received vectors into 2^k disjoint subsets such that each subset contains one and only one code vector is described here

- First, the 2^k code vectors of C are placed in a row with the all-zero code vector v1 = (0, 0, …, 0) as the first (leftmost) element From the remaining $2^n - 2^k$ n-tuple, an n-tuple e2 is chosen and is placed under the zero vector v_1
- Now, we form a second row by adding e2 to each code vector v_i in the first row and placing the sum e2 + v_i under v_i
- An unused n-tuple e3 is chosen from the remaining n-tuples and is placed under e2.
- Then a third row is formed by adding e3 to each code vector v_i in the first row and placing e3 + v_i under v_i .
- we continue this process until all the n-tuples are used.

Standard Array



0	v_2	$\cdots$	v_i	$\cdots$	v_{2^k}
e_2	$e_2 + v_2$	$\cdots$	$e_2 + v_i$	$\cdots$	$e_2 + v_{2^k}$
e_3	$e_3 + v_2$	$\cdots$	$e_3 + v_i$	$\cdots$	$e_3 + v_{2^k}$
$\vdots$					$\vdots$
e_l	$e_l + v_2$	$\cdots$	$e_l + v_i$	$\cdots$	$e_l + v_{2^k}$
$\vdots$					$\vdots$
$e_{2^{n-k}}$	$e_{2^{n-k}} + v_2$	$\cdots$	$e_{2^{n-k}} + v_i$	$\cdots$	$e_{2^{n-k}} + v_{2^k}$

Example4 : Find code words. Find syndrome table, standard array. Error detection and correction capability of a code.

Coded word $[C]_{1xn} = [D]_{1xk} \cdot [G]_{kxn}$

$$[C] = [D1\ D2\ D3\ D4] \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

[C1,C2,C3,C4,C5,C6,C7] = [D1, D2, D3, D4, D1⊕D2⊕D3, D1⊕D2⊕D4, D1⊕D3⊕D4]

$\binom{k}{2}$ = 2^4 Codewords
= 16 codewords

D1	D2	D3	D4	C1	C2	C3	C4	C5	C6	C7	W(C)
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	1	1	1	4
0	0	1	0	0	0	1	0	1	0	1	3
0	0	1	1	0	0	1	1	1	1	0	4
0	1	0	0	0	1	0	0	1	0	0	2
0	1	0	1	0	1	0	1	1	0	1	4
0	1	1	0	0	1	1	0	0	1	1	4
0	1	1	1	0	1	1	1	0	0	0	3
1	0	0	0	1	0	0	0	1	1	1	4
1	0	0	1	1	0	0	1	1	0	0	3
1	0	1	0	1	0	1	0	0	1	0	3
1	0	1	1	1	0	1	1	0	0	1	4
1	1	0	0	1	1	0	0	0	0	1	3
1	1	0	1	1	1	0	1	0	1	0	4
1	1	1	0	1	1	1	0	1	0	0	4
1	1	1	1	1	1	1	1	1	1	1	7