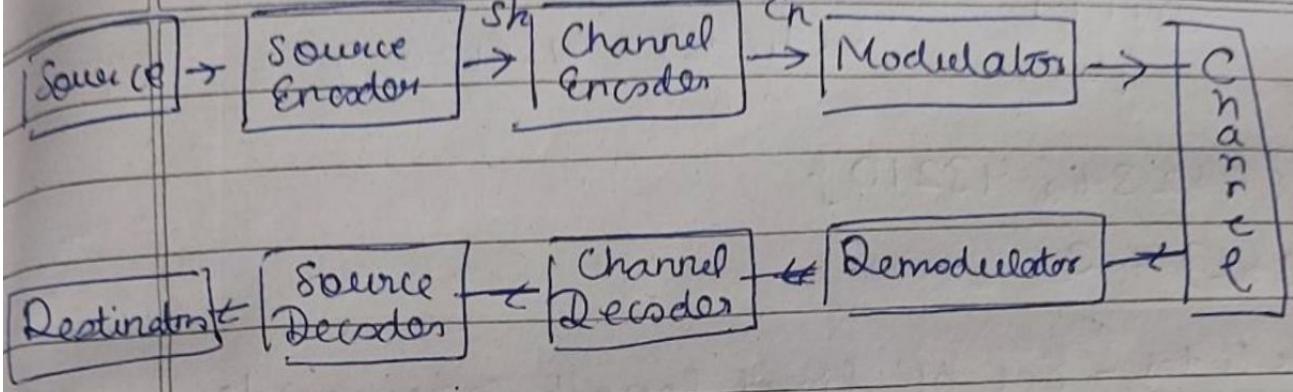
Error Correcting Codes :-

Date:
Page No.



Coding Theory is an intrsting subject

interesting subject

The objectives of coding techniques

Error correcting capacity → in terms of no. of errors can be detected & corrected by the code

~~First~~ Aim encoding of the message → Efficient encoding strategy

Fast & efficient decoding of message received

Maximum transfer of information per unit time

Word → sequence of symbol that is representing an information $w = \{101\}$

Code - Sequence of vector also known as code word.

$C = \{1101, 1111, 0011\}$

Hamming weight - No. of non-zero elements in codeword.

$w(c_1) = 3$; $w(c_2) = 4$; $w(c_3) = 2$

Hamming Distance → No. of places code word differs.

$d(c_1, c_2)$ = 1101 & 1111 differ at 3rd pos. from left

$\therefore d(c_1, c_2) = 1$ $\therefore d(c_1 c_3) = 3$ $\therefore c(c_2, c_3) = 2$

$C = \{0100, 1111\}$ $d(c_1, c_2) = w(c_1 - c_2)$

$w(c_1) = 1$; $w(c_2) = 4$; $d(c_1, c_2) = 3$

$c_1 - c_2 = 0100 \oplus 1111 = 1011$ $\therefore w(c_1 - c_2) = 3$

$\therefore C(d_1, d_2) = w(c_1 - c_2)$

$C = \{01234, 43210\}$

$d(c_1, c_2) = 3$

Block Codes - Set of fixed length codewords.

Block length - No of symbols in each codeword.

Block size - No. of codewords in the code.

Block code of size 'M' representing alphabets of 'q' symbols.

⇒ M q-ary code of each of length n

$M = q^k$ → no. of bits

↑ size ↑ no. of symbols representing code

→ (n, k) code

$C = \{00000, 10100, 11110, 11001\}$

$n = 5$ $q = 2$ $M = 4$

Now $M = 2^k$ ∴ $K = 2$

Unoded	Coded
00	00000
01	10100
10	11110
11	11001

K → no. of bits division for the input stream

10|01|01|00|11

→ 11110 10100 10100 00000 11001

(n, k) → $r = \frac{k}{n} \leq 1$ for $r = 1$ (No encoding)

To increase r ↑ we must decrease n ↓

∴ error correction capability decreases.

Minimum weight :- minimum hamming wt. out of all the codewords in a given code. $(w^*) = w(C_i)_{min}$

Page No.

Minimum distance :- Minimum distance between a pair of codewords out of all the possible codeword pairs in the given code $(d^*) = d(C_i, C_j)_{min}$ $i \neq j$

$d^* = w^*$

Linear Code - Linear Code / Linear block has these following properties :-

→ Sum of two codewords belongs to the code is also a codeword belonging to the code.

Ex :- $C = \{C_1, C_2, C_3\}$ then $C_1 + C_2 = C_3$ as $C_3 \in C$

$\therefore C_1 = C_3 - C_2$ and $C_2 = C_3 - C_1$

→ The all zero codeword is always one of the codewords

Ex :- $C = \{0000, 0101, 1111, 1010\}$

↑ belongs ∴ linear

→ Minimum hamming distance between 2 codewords of a linear code is equal to the min. of weight ~~of~~ any non-zero code word.

If all 3 properties above are satisfied by a code, then its called a linear code.

Field: Set of elements with two operations i.e. addition & multiplication satisfying some properties

→ F is closed over addition & multiplication

~~Linear Codes~~:- $F = \{a, b, c\}$ $(+)$ $(\cdot)$

~~$C = \{c_1, c_2\}$~~, $a+b$ and $a \cdot b$ must be included in F in $a, b \in F$.

Commutative Law

$a + b = b + a$ and $a \cdot b = b \cdot a$

Associative Law

$(a+b)+c = a+(b+c)$

$a \cdot (b \cdot c) = (a \cdot b) \cdot c$

Distributive Law

$a \cdot (b+c) = a \cdot b + a \cdot c$

Identity element '0' and '1' must exist in F satisfying :-

$a + 0 = a$ and $a \cdot 1 = a$

Additive Inverse

For any 'a' in F there must exist an additive inverse '-a' such that $a + (-a) = 0$

Multiplicative Inverse

For any 'a' in F there exists a multiplicative inverse 'a^{-1}' such that $a \cdot a^{-1} = 1$

A field with a finite no. of elements 'q' is known as Galois (Galva) field 'GF(q)'. If only the first '4' properties are satisfied, then set is called a 'ring'.

$q = p^n$ ← any integer

↑ prime no.

GF(2), GF(3), GF(5), GF(7)

GF(2) = {0, 1}

GF(3) = {0, 1, 2}

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

(Add table)

×	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

(Multiply Table)

$0 + 0 = 0$

$1 + 4 = 0$

$2 + 3 = 0$

$1 \times 1 = 1$

$2 \times 3 = 1$

$4 \times 4 = 1$

If 'q' is not a prime no, then module operation is not possible.

$GF(q)$ is a subset of $GF(q^n)$

$GF(2^2) = \{00, 01, 10, 11\}$

S is set of vectors defined over $GF(q)$

$GF(2) = \{0, 1\} \therefore S = \{001, 010, 110\}$

All linear combinations of vectors in S is called linear span of S, $\langle S \rangle = C$

C must contain all codes in S

$S = \{1100, 0100, 0011\}$

$1100 + 0100 = 1000$

$0100 + 0011 = 0111$

$1100 + 0011 = 1111$

$1100 + 0100 + 0011 = 1011$

$\langle S \rangle = \{1100, 0100, 0011, 1000, 1111, 0111, 1011\}$

Matrix representation of linear code:-

G → Generator Matrix

$S = \{101, 111, 010\}$

$$k \times n \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Rows are linearly independent.

Ik	D
00	00 010
01	
10	
11	

Choice of row is not unique for generator matrix, ∴ generator matrix is also not unique for a codeword.

Generator matrix converts codeword of length 'k' to a vector of length 'n' and any codeword 'C' can be represented as $C = IG$ where 'I' is information bit/word & 'G' is generator matrix.

$G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}_{k \times n}$ ∴ $k = 2$, $n = 3$

$C_1 = [00]\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

$= 000$

$C_2 = [01]\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

$= 010$

$C_3 = [10]\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

$= 101$

$C_4 = [11]\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

$= 111$

Advantages of Generator Matrix

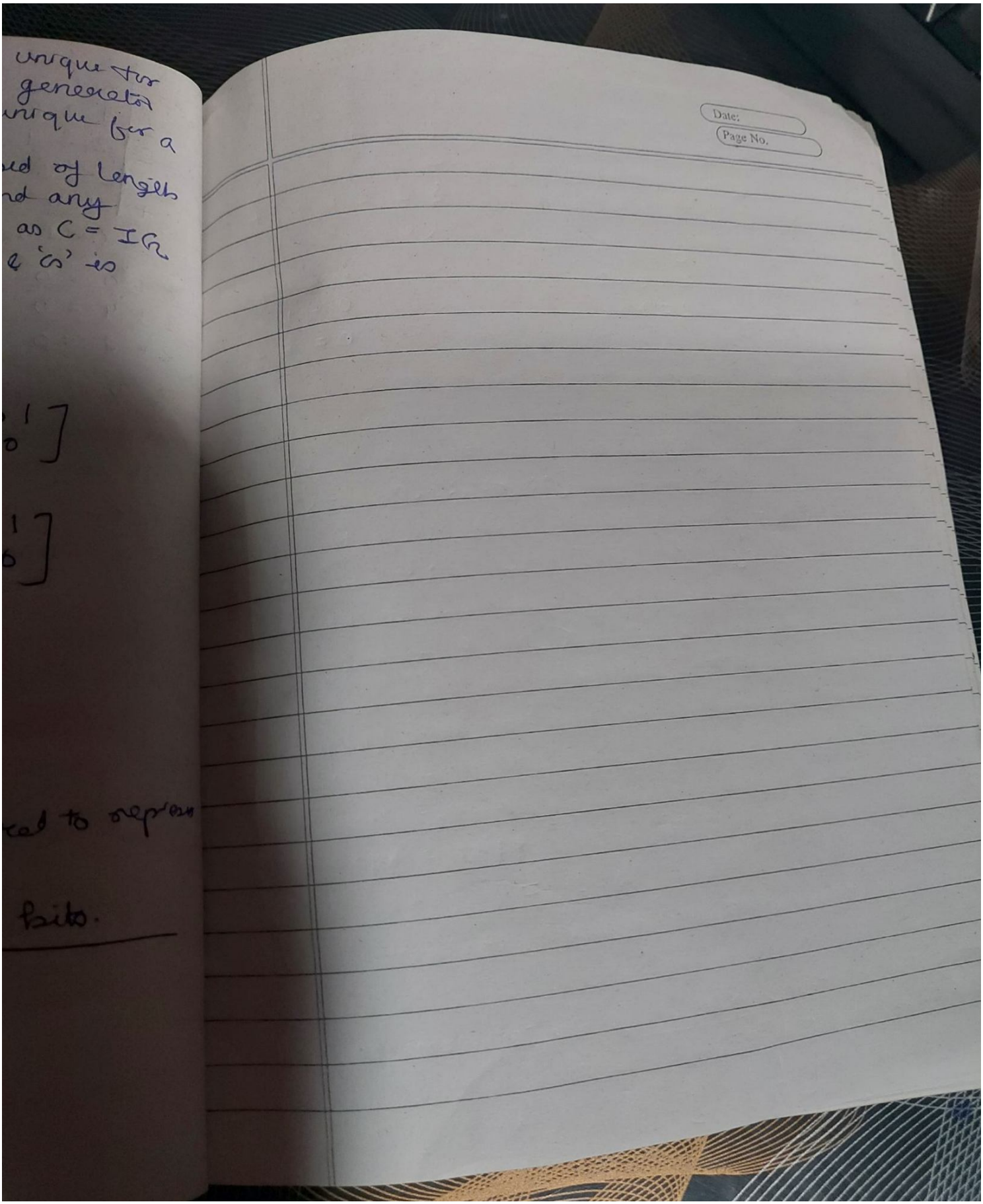
→ Memory Saver.

$G_{n \times k} \rightarrow q^k$ $(n, k) \rightarrow (\underline{46}, 24)$ code

$\rightarrow 2^{24} = 1777216$ codes available

$\therefore 2^{24} \times 46 = 771,751936$ bits required to represent all the codes.

whereas $n \times k$ matrix required = 1107 bits.


unique for
generator
unique for a
ed of length
nd any
as C = IG
& 'G' is
Date:
Page No.

Symmetric Matrix

$G = [I/P]$ (k×k, k×(n-k)) $GH^T = 0$ $H = [-P^T \uparrow I]$ ((n-k)×n)

GF(3) ← (n, k) → $[I_1, I_2, I_3 \cdots I_k \mid P_1\ P_2 \cdots P_{n-k}]$

SN	Information Symbol k=2	Codeword n=5
1	0 0	00 000
2	0 1	01 121
3	0 2	02 220
4	1 0	10 012
5	1 1	11 221
6	1 2	12 210
7	2 0	20 020
8	2 1	21 100
9	2 2	22 212

Singleton Bound:

$d^* \leq n-k+1$

Minimum distance (minimum weight) of (n, k) linear code is bounded by this relation where d^* is min distance of the code.

$$G = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{array}\right]$$

I : P (k×n)

Each row represents a valid codeword k=3 and n=4

First 'k' symbols are information bits and remaining 'n-k' symbols are parity bits.

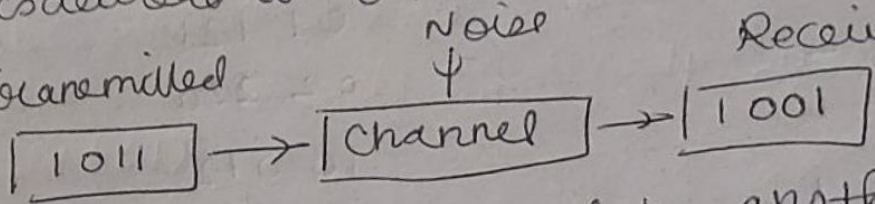
Information position contains only a single non zero element (1) while (n-k) parity bits may or not be zero.

∴ No. of non zero bits in a codeword
= 1 + (n-k) (Maximum when all 'n-k' parity bits)

No. row has more symbols '1' than above value
when $d^* = n-k-1$ → maximum distance code.

Decoding of Linear Block Code

Main objective is to detect & correct errors
a codeword is transmitted over a noisy channel



Noise converts a symbol to another.

$d(C_T, C_R) = 1$

For 't' errors → t no. of transmitted symbols are changed.

Linear Block Code (n, k) with d^* = min hamming dis
If the received codeword is also part of the LBC
(although erroneously transmitted) it cannot be error
detected / corrected.

An error will be detected, as long as one codeword
does not transforms into another valid codeword.

If the no. of bits transformed due to error is
than or equal to $d^* - 1$ then all non-zero error
patterns can be detected.

$C_1 = \{000, 111\}$

$d^* = 3$

For error detection
change $\leq d^* - 1 = 2$ → {001, 010, 011, 101, 100, 110}

$C_2 = \{001, 110, 101\}$ $d^* = 1$

error correction & detection → $d^* - 1 \leq 0$

Error Correction

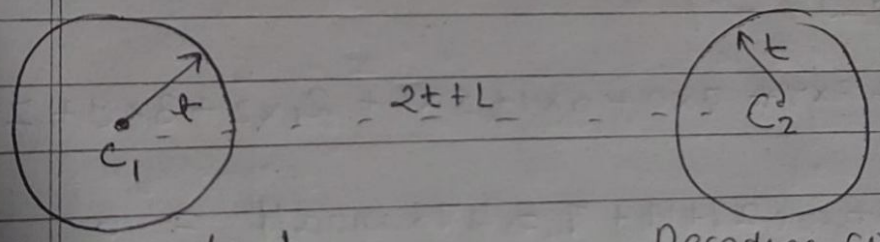
To make best possible guess regarding the originally transmitted codeword after observing the received word.

Nearest neighbour decoding:

Codewords = [101, 110, 111, 001] $v = [100]$

The received codeword is compared with all the codewords, and the least hamming distance one is selected to be the correct one.

In case two or more codeword have the same hamming distances then either the receiver asks for retransmission or arbitrarily selects one codeword to be the correct one.



Decoding circle-1 Decoding circle-2

For no overlap $dist(C_1, C_2) \geq 2t+1$

Number of errors that can be detected are fixed.

$C = \{00000, 01010, 10101, 11111\}$ $d^* = 2$

let $d^* = 2t+1 \Rightarrow 2 = 2t+1 \therefore t = 1/2$ but not possible as t can't be a fraction.

$C_7 = 11111$ $C_R = 11110$

$d(C_1, C_R) = 4$; $d(C_2, C_R) = 2$; $d(C_3, C_R) < 3$; $d(C_4, C_R) = 1$

$\therefore t = 1$ so $d^* \geq 2t+1$ $\boxed{\Rightarrow 2 \geq 3}$

Not fulfilled the condition but still sometimes possible to detect errors.

$C_T = [00000]$ $C_R = [01000]$

$d[C_1, C_R] = 1$; $d[C_2, C_R] = 1$, $d[C_3, C_R] = 4$ $d[C_4, C_R] = 3$

↳ Ambiguity, can't detect single error

Incomplete Decoder: No ambiguity → Detect
Ambiguity → Retransmission

Complete Decoder: Always detects, even in case of ambiguity → arbitrary selection or logic based correction

---

International Standard Book Number (ISBN)

→ Linear block code of length '10' defined over Galois field (11)

10 symbols - '0', '1', ..., '9', 'X'

$\sum_{i=0}^{9} (10-i) c_i = 0$ calculated over mod 11

c_i = ith symbol.

0-07-048297-7

⇓

$10\times0 + 9\times0 + 8\times7 + 7\times0 + 6\times4 + 5\times8 + 4\times2 + 3\times9 + 2\times7 + 1\times7$

$56 + 24 + 40 + 8 + 27 + 14 + 7 = 176 \bmod 11 = 0$

All zero code is a valid ISBN code.

1 00 000 000 1 → is also valid

$\Rightarrow 10\times1 + 1\times1 \Rightarrow 11 \bmod 11 \Rightarrow 0$ $\quad d^* = 2$

$d^* \geq 2t+1 \Rightarrow 2 \geq 2t+1 \Rightarrow 1 \geq 2t \Rightarrow t = 1/2$ not possible

only $d^* - 1 \therefore 2-1 = 1$ error can be detected

But no error can be corrected as $t = 0$

0-07048e97-7 ← error

$8\times7 + 6\times4 + 5\times8 + 3\times9 + 2\times7 + 1\times7 = (168 + 4e) = 0$

$\therefore (168 + 4e) \bmod 11 = 0$

$\therefore 4e \bmod 11 = -3$ $\quad \boxed{e = 2}$

Example:
0 volt 5 volt
↑ ↑
Binary pam → Either '0' or '1' transmitted
Noise margin: 1V (range of values that can be converted as '1' or '0')

∴ 4V–5V : 1 and 0–1V → 0 and 1–4V → Erasure

Practical channels have both erasure & noise. The decoder must have the capacity to compensate the erasure & detect + correct the errors by noise.

Hamming min distance becomes $(d^* - x)$
where t = no. of errors, and x = no. of erasures.

$\Rightarrow d^* - x \geq 2t+1$ or $d^* \geq 2t+x+1$

for $x=0$, $d^* \geq 2t+1$ for $t=0$; $d^* \geq 2x+0+1$

$x=2$

0 → 00000 ?? 000

1 → 11111 ?? 111

$d^* = 5$ $d^* = 3 = d^* - x = 5-2=3$

Coset:

$C = (n, k)$, $a \in (q)$

a = any vector of length n

$a + C = \{a+x \mid x \in C\}$

for $C = \{c_1, c_2, c_3, c_4\}$ $\Rightarrow C+a = \{c_1+a, c_2+a, c_3+a, c_4+a\}$

↳ Coset of a

a & b are two vectors of length n are said to be in same coset if $a-b \in C$

Every vector of length n is in some coset of C

C = {0000, 0101, 1010, 1111} ; a = 0110 ; b = 1000

a+c = {0110, 0011, 1100, 1001} b+c = {1000, 1101, 0010, 0111}

Each coset contain exactly q^k vectors
$C(n,k)$ GF(q) where $q=2$ and $k=2$ $\therefore q^k = 4$

Two cosets are either disjoint or coincide. Partial overlap is not possible.

If $a+c$ is a coset of C and $b \in (a+C)$
So $b+C = a+C$

Coset Leader: Vectors having min. distance is considered coset leader. For many choice, arbitrary selection may be done.

$C \rightarrow [n,k]$ $G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ $C = \{000, 101, 010, 111\}$
$[3,2]$

let $a = 000$ $\therefore a+C = \{000, 101, 010, 111\}$
let $b = 001$ $\therefore b+C = \{001, 100, 011, 110\}$
$2^3 = 8$
These two cosets cover all the possible combos (q^n)
If $a, c \Rightarrow C$ and $b \in a+c$ $\therefore a+C = b+C$

$a = 000$ and $a+C = \{000, 101, 010, 111\}$
Let's take $b = 101$
$\therefore b+c = \{101, 000, 111, 010\}$
— equal. used in standard array decoding.

Standard Array
$q^{n-k} \times q^k$ ← dimension of all vectors of GF(q^n)

The 1st row consists of original code with all zero codeword at the leftmost position.
The 2nd & other rows is the coset of a_i+C until all possible positions are covered by the array.

Formation:
1st row → all valid codewords (start with all zero)
Choose a vector a_1 not in 1st row
Write coset a_1+C as 2nd row such that a_1+X is written under X where X is an element of C

Next choose another vector a_2 not present in first two rows.

Date:

Write $a_2 + C$ as 3rd row such that ~~$a_2 + x$~~ is written below x where $x \in$ ~~2nd row~~ C.

Continue until all vectors of $GF(q^n)$ are listed.

$C = \{0000, 1011, 0101, 1110\}$

1st row = 0000 1011 0101 1110

choose $a_1 = 1000$

2nd row = 1000 0011 1101 0110

choose $a_2 = 0100$

3rd row = 0100 1111 0001 1010

choose $a_3 = 0010$

4th row = 0010 1001 0111 1100

↑ coset leaders

For ex received codeword = 0001

∴ error = coset leader of its row = 0100

⇒ Original codeword = 0001 − 0100 = 0101 ← top in 1st row

Cosets:

$C = (n, k)$ $a \to n$ same coset ← (a, b) n

$a + C = \{a + x, x \in C\}$ ∴ $a - b \in C$

$a + c \to C$ and $b \in a + c$ ⇒ $a + C = b + C$

$C = \{c_1, c_2, c_3, c_4\}$

$a + C = \{c_1 + a, c_2 + a, c_3 + a, c_4 + a\}$

Coset leaders $w^t = \{\min w(a + c)\}$

Standard Array →

$C, [n, k] \rightarrow [q^{n-k} \times q^k] \rightarrow GF(q^n)$

$q=2, n=4 \Rightarrow q^n = 16$

$C = \{0000, 1011, 0101, 1110\}$

1st ⇒	0000	1011	0101	1110
2nd ⇒	1000	0011	1101	0110
3rd ⇒	0100	1111	0001	1010
4th ⇒	0010	1001	0111	1100

coset leader column min wt. codeword.

pick an element not in 1st row & then add to all elements

rows $= q^{n-k}$
$= 2^{4-2} = 4$

columns $= q^k = M$
$= 2^2 = 4$

$V = 1101$ $\quad 0101 \xrightarrow{\text{channel}} 1101$

$0000 \xrightarrow{\text{channel}} 0101$

No error detected as both are valid codewords

1101 ← wrong keyword
+1000 ← coset leader
0101 ← valid keyword

Similarly for $V = 1001$; $DC = 0011$; $e = 0010$

$V = 1101$

$d(V, 0000) = 3$
$d(V, 1011) = 2$
$d(V, 0101) = 1$ → min dist ∴ considered original code.
$d(V, 1110) = 2$

$C[n, k] \rightarrow q^{n-k} \times q^k$ if n, k are large, then it's difficult to solve the large standard array, we'll use Syndrome Decoding we can reduce the size of the standard array.

Let $H \Rightarrow$ Parity matrix

∴ Syndrome of received word $(V) = VH^T$

Now if $V = 0$ ∴ $\boxed{CH^T = 0}$

If $V \neq C$ $\quad a = VH^T \rightarrow$ a vector $S(V)$ ← syndrome

$a + C \xrightarrow{\text{coset}} C$ We can say $x \& y$ in same coset C if they have same syndrome.

We know that for $x \in C$ & $y \in C$

$x + C = y + C$ and $x - y \in C$

$\Rightarrow (x-y)H^T \in CH^T \Rightarrow (x-y)H^T = 0$

$\therefore xH^T = yH^T \therefore S(x) = S(y)$

There's one to one correspondence b/w vectors & syndrom

					Syndrome
$C \Rightarrow$	0000	1011	0101	1110	00
	1000	0011	1101	0110	11
	0100	1111	0001	1010	01
	0010	1001	0111	1100	10

only coset leaders are considered.

First find syndrome of v; $V \to S(V) = VH^T$

Identify this syndrome from the table.

The corresponding coset leader gives the error pattern

Then original code $c = v - e$

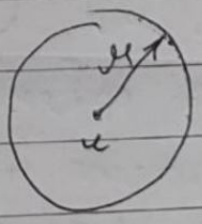
Perfect Code:

Any vector u in $GF(q^n)$

any integer $n \geq 0$

Sphere of radius r, centered u is

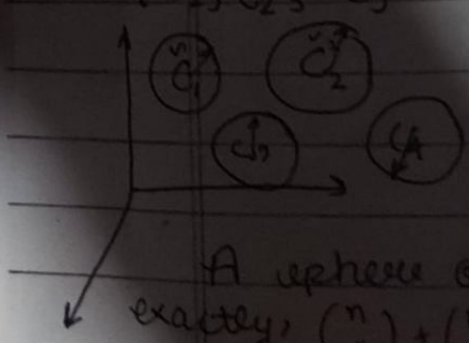
$S(u, r) \to \{V \in (GF(q^n)) \mid d(U, V) \leq r\}$



$C = \{C_1, C_2, C_3 \cdots C_n\}$ $\quad d > 2t+1$

If $d(C_1, V_1) \leq t$ $\quad V_1$ & V_2 lie in

If $d(C_2, V_2) \leq t$ spheres of C_1 & C_2



A sphere of radius 'r' $(0 \leq r \leq n)$ contains exactly, $\binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \cdots \binom{n}{r}(q-1)^r$

$u \to GF(q^n)$ $\quad d(u,v) = m$

$\binom{n}{m}$ → Choose n positions from m

Each of the 'm' places of difference between u & v can be replaced by $q-1$ symbols

The no. of vectors at exactly 'm' distance from u are given by

$$\binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-2)^2 + \cdots \binom{n}{m}(q-1)^m$$

For binary ($q=2$) and 4 bits code ($n=4$): find no. of vectors with distance 2 or less from any codeword.

$$\Rightarrow \binom{4}{0} + \binom{4}{1}(2-1) + \binom{4}{2}(2-1)^2$$

$$\Rightarrow 1 + 4 + 6 = 11$$

$v = 0000$

$d=2$; 0011, 1001, 1010, 1100, 0110, 0101 } Total = 11

$d=1$; 0001, 0010, 0100, 1000

$d=0$; 0000

q ary (n,k) M codewords $d^* = 2t+1$

$$M\underbrace{\left\{\binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \cdots \binom{n}{t}(q-1)^t\right\}}_{\text{Each sphere of radius 't' centered about the codeword contains the above mentioned no. of codewords.}} \leq q^n$$

For M no. of codewords multiply by M.

For a q ary Galva field with n bits codewords Total possible symbols = q^n : the boundation on above expression (known as Hamming Bound/ Sphere packing bound).

For binary code:

$$M\left[\binom{n}{0} + \binom{n}{1} + \binom{n}{2} \cdots \binom{n}{t}\right] \leq 2^n$$

Now, put $M = q^k$ and take log on both sides

$$2^k\left[\binom{n}{0}+\binom{n}{1}+\binom{n}{2}\cdots\binom{n}{t}\right] \le 2^n$$

$$\log_2\left[\binom{n}{0}+\binom{n}{1}+\binom{n}{2}\cdots\binom{n}{t}\right] \le n-k$$

Date:
Page No.

A Perfect Code is one that achieves the Hamming Bound Based on above expression we can conclude that for a perfect code there are equal radius disjoint spheres centered at the codeword that completely fill the space, thus a 't' error correcting perfect code utilize the entire space in the most efficient manner.

$C = \{ \underset{n \text{ bits } n \text{ is odd!}}{\underset{1111\cdots11}{0000\cdots00}} \}$ Block Length = n; Binary Repetion Code

Let M = 2

and $t = \frac{n-1}{2}$ $\quad d^* = 2t+1 = n \Rightarrow t = \frac{n-1}{2}$

Find out the expression value & prove Binary repetition code is a Perfect Code

$$\Rightarrow 2\left[\binom{n}{0}+\binom{n}{1}+\cdots\binom{n}{\frac{n-1}{2}}\right] = 2^n$$

$$\Rightarrow 2 \times 2^{n-1} = 2^n = 2^n$$

LHS = RHS ∴ Binary Repetition Code is Perfect Code

Parameters for Perfect Code: q, n, M, t

→ All should be integers & n should be odd.

Eg:

n	q	M	t
23	2	2^{12}	3
90	2	2^{78}	2
11	3	3^6	2

Perfect Codes → Combinations

Binary Hamming Code

$(n,k) = (2^m-1,\ 2^m-1-m)$ $\quad m$ = +ve integer

for $m=3$ $\quad (n,k) = (7,4)$

For a binary (n, k) Hamming Code $H_{(n-k)\times k}$ parity check matrix
the n columns consist of all possible binary vectors with $n-k = m$ elements except the all zero vector

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} = [-P^T \mid I]$$

$$G = [I \mid P]$$

$\Rightarrow n - m = k \Rightarrow 4 \; 7 \; 4 \; 5 \; 3$

$n = 2^3 - 1 = 7$

$n - k = m = 3$

For $m > 1$ it is possible to identify 3 columns of H that would add up to zero vector which implies that min. distance of $(n, k]$ hamming code $= 3$.

$\therefore d^* \geq 2t+2 \Rightarrow 3 \geq 2t+1 \Rightarrow t = 1$

Single error correcting code.

By adding an overall parity bit, an $m-k$ hamming code can be modified to $(n+1, k)$ code with $d^* = 4$

On the other hand, an (n, k) hamming code can be shortened to $(n-l, k-l)$ code by removing 'l' rows from G matrix or l no. of columns from parity check matrix; ~~or equivalently~~

Low Density Parity Check Code

(r, s) Gallager Code (Linear Code)

Each column $\rightarrow r$ ones
Each row $\rightarrow s$ ones
} H parity check matrix $(n-k)\times(n)$

If r & s are very small then it's known as Low Density Parity Check Code.

$r \leq \log_2 N$ where N = block length

(n, r, s) code

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$r = 2$
$l = 4$
$n = 10$
$k = 5$

→ Irregular

LDPC → Irregular (rRs – not fixed)
→ Regular (rRs – fixed)

Sum of last 3 columns = 0 ∴ $d^* = 3$

LDPC code can be generated by either Random Construction (Irregular Code) or Algebric Construction (Regular Code) or a combination of both.

Random Construction:

Step 1: Set $i = 1$

Step 2: Generate a random binary vector of n 0's/1's length with hamming weight 'r'.
→ ith column of H

Step 3: Hamming wt. of each row of $H \leq s$ and the scalar product of each pair of column ≤ 1
set $i = i + 1$

step 4: If $i = n$ stop the iteration else go to step 2

But this algorithm doesn't guarantee 's' no of 'ones' in each row.

Algebric Construction:

step 1: Choose p) $(n-1)/(s-1)$

step 2: PXP I matrix by cycling shifting each row one position to the right.

For ex: $\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$ —shift one right→ $\begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & & & \\ 1 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$ ≠ I^1

Parity check matrix can be formed from above matrix

$$H = \begin{bmatrix} I^0 & I^0 & \cdots & I^0 \\ I^0 & I^1 & \cdots & I^{c-1} \\ \vdots & & & \\ I^0 & I^{r-1} & \cdots & I^{(r-1)(c-1)} \end{bmatrix}$$

where $I_0 = I_p$
I^0 = no shift
I^1 = 1 shift I_p
I^n = n shift I_p

All types of linear codes & even LDPC can be represented by Tanner Code / Graph.

Every received codeword 'v' is checked by 'r' eqn and every 'r' eqn adds up 'c' code symbols.

Tanner graph has two kinds of nodes:
→ Check Nodes (check equation)
→ Symbol Nodes (bit / block length)

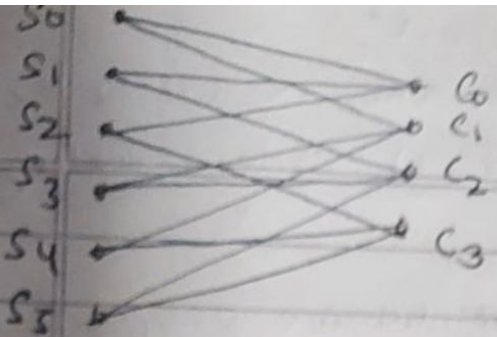
H(n-k, n)
→ Check node (connected to only symbol node) & vice versa.

$$H = \begin{array}{c} \\ c_0 \\ c_1 \\ c_2 \\ c_3 \end{array} \begin{array}{c} \begin{array}{cccccc} s_0 & s_1 & s_2 & s_3 & s_4 & s_5 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

$n = 6$ (columns)
$x = 2$ (ones per column)
$s = 3$ (ones per rows)

symbol (node) = 6 ; check node $(n-k) = 6 - 2 = 4$

$\frac{k}{n} = \frac{1}{3} \Rightarrow \frac{k}{6} = \frac{1}{3} \Rightarrow k = 2$



If $H_{ij} = 1$ then connect C_i & S_j

Step 1:- Make hard decision to decide in favour of '1' & '0'
Let received codeword 'v' = 001000

Step 2:- Calculate syndrome of v
$S = vH^T \Rightarrow S = 1001$
If S = all zeroes then no error, in given some errors are present.
By position of '1' we can determine errors at C_0 and C_3.

Step 3:- Consider S_3. Its connected with C_1 & C_2 where there is no error ∴ no error in transmission of S_3

Step 4:- For C_0 → symbols connected → S_0, S_1, S_2
Fail Check = 1

Step 5:- For C_3 → symbols connected → S_2, S_4, S_5
Fail Check = 1

Step 6:- For S_2 → Connected to C_0 & C_3
Fail Check = 2 (Both C_0 & C_3 are 1)
Maximum fail check for S_2
⇒ Flip the maximum erroneous symbol
⇒ [001000] → [000000]
Syndrome of new codeword = all zeroes
∴ No error remains.

Optimal Linear Code (n, k, d^*)

↳ Considered Optimal if there are no $(n-1, k, d^*)$, $(n+1, k+1, d^*)$ or $(n+1, k, d^*+1)$ codes existing.

Optimal linear codes gives best distance possible under the constraint of block length.
Most optimal codes are found by long computational process.
For a particular value of n, k & d^* there are multiple optimal linear codes available.

Maximum Distance Seperable Code (MDS):

We find max d^* for given redundancy 'r';
r = No. of bits added to input to get codeword
$= n - k$

$(n, \underbrace{n-r}_{k}, d^*) \Rightarrow d^* \leq r+1$

Singleton bound $d^* \leq n-k+1$

$\Downarrow$

$d^* \leq r+1 \quad (r = n-k)$

When $d^* = d^*_{max} = r+1$; then code becomes MDS represented by $(n, n-r, r+1)$

Cyclic Code: Cyclic shift on a code results in another code. This condition allows us for the simpler implication of cyclic code using shift-registers.
a code 'C' is cyclic codeword if)
→ C is a linear code
→ Any cyclic shift of the codeword is another codeword.

$C = a_0 a_1 a_2 \cdots a_{n-1}$

$C_1 = a_{n-1} a_0 a_1 a_2 \cdots a_{n-2}$ ← cyclic shift

$C_1 = \{0000, 0101, 1010, 1111\}$ → cyclic code

$C_2 = \{0000, 1001, 0110, 1111\}$

cyclic — Not cyclic — cyclic

If there is an interchange between 3rd & 4th element in C_2, we get 1001 → 1010 and 0110 → 0101 which makes it a cyclic code.

Polynomials:-

$f(x) = f_0 + f_1 x + f_2 x^2 + f_3 x^3 + \cdots + f_m x^m$

$f_0, f_1 \cdots f_m$ are defined over Galva field $GF(q)$

f_m → leading coefficient (coefficient associated with max power of x)

if $f_m \neq 0$ then m = degree of $f(x)$ → $\deg f(x) = m$

Monic:- ($f_m = 1$)

$f(x) = 3 + 7x + x^3 + 5x^4 + x^6$ ($q = 8$) → (0 to 7) (min value)

$F[x]$ - set of polynomials defined over $GF(q)$

different polynomials belong to $F[x]$ that can be added, subtracted or multiplied in usual manner.

$F[x]$ → represents algebraic structure known as 'ring'.

If degree $m > 0$ then for that particular polynomial, no multiplicative inverse exists.

$f(x), g(x) \in F[x]$

$\deg[f(x) \cdot g(x)] = \deg f(x) + \deg g(x)$

$\deg[f(x) + g(x)] \neq \deg f(x) + \deg g(x)$

$= \max\{\deg f(x), \deg g(x)\}$

$f(x), g(x) \longrightarrow GF(2)$

$f(x) = 1 + x^2 \qquad g(x) = 1 + x + x^2$

$\deg(f(x)g(x)) = \deg f(x) + \deg g(x)$

$= 2 + 2 = 4$

$f(x)g(x) = 1 + x + x^2 + x^2 + x^3 + x^4$

$= 1 + x + 2x^2 + x^3 + x^4$

$\deg(x) = 1$

$f(x) = 2 + x + x^2 + 2x^4$ $\Big\}$ $\longrightarrow GF(3)$

$g(x) = 1 + 2x^2 + 2x^4 + x^5$

$f(x) + g(x) = 3 + x + 3x^2 + 4x^4 + x^5 \qquad \deg = 5$

$f(x)g(x) = 2 + 4x^2 + 4x^4 + 2x^5$

$x + 2x^3 + 2x^5 + x^6$

$x^2 + 2x^4 + 2x^6 + x^7$

$2x^4 + 4x^6 + 4x^8 + 2x^9 \qquad \deg = 9$

$= 2 + x + 6x^2 + 2x^3 + 6x^4 + 4x^5 + 7x^6 + x^7 + 4x^8 + 2x^9$

$f(x) = 1 + x$ over $GF(2)$

$(f(x))^2 = 1 + x^2$

$f(x) = 1 + x$ over $GF(3)$

$(f(x))^2 = 1 + 2x + x^2$

Division Algorithm:

$a[x], b(x) \neq 0$ in $F[x]$

The division algorithm states that for every pair of polynomials $a(x)$ & $b(x) \neq 0$ in $F[x]$, there exists a unique pair of polynomial $q(x)$ → quotient and $r(x)$ → remainder such that $R(x) = q(x)b(x) + r(x)$ where $r(x)$ is $\deg r(x) < \deg b(x)$

Residue denoted by $R_{b(x)}[a(x)] = r(x)$

$R_{f(x)}[a(x)+b(x)]$

$= R_{f(x)}[a(x)] + R_{f(x)}[b(x)]$

$R_{f(x)}[a(x)-b(x)] = R_{f(x)}\{R_{f(x)}[a(x)], R_{f(x)}[b(x)]\}$

Let $a(x) = x^3+x+1$ and $b(x) = x^2+x+1$ GF(2)

find out $q(x)$ and $r(x)$

$x^3+x+1 = x[x^2+x+1] + [1-x^2]$

$q(x) = x$ and $r(x) = 1-x^2$

$x^2+x+1 \;|\; x^3+x+1$ — $x+1$

x^3+x^2+x

x^2+1

x^2+x+1

x

Congruent Modulo $f(x)$

$f(x) \rightarrow$ fixed polynomial in $F[x]$

$g(x), h(x)$ in $F[x]$

$g(x) = h(x)$ modulo $(f(x))$

if $g(x)-h(x)$ is divisible by $f(x)$

$g(x) = x^9+x^2+1$; $h(x) = x^5+x^2+1$; $f(x) = x^4+1$, GF(2)

$g(x)-h(x) = x^9+x^5$

x^5+x

$x^4+1 \;|\; x^9+x^4+x^2$

x^9+x^5

$x^5+x^4+x^2$

x^5+x

x^4+x^2+x

$g(x)-h(x) = x^5 f(x)$

↳ congruent modulo of $f(x)$.

We consider a new set of polynomials by $\frac{F[x]}{f(x)}$, where all polynomials belong in $F[x]$ where also the degree of all polynomials must be less than degree of $f(x)$

① $a(x)$ and $b(x)$ belongs to $F[x]/f(x)$ then $a(x) + b(x)$ in $\frac{F[x]}{f(x)}$ is same as $F[x]$

② The product $a(x).b(x)$ gives a unique polynomial of degree $< \deg(f(x))$ where $a(x).b(x)$ is congruent modulo $f(x)$

$\frac{F[x]}{f(x)}$ ring of polynomials over $F[x]$, modulo $f(x)$

$(x+1)^2$ in $\frac{F[x]}{x^2+x+1}$ defined over $GF(2)$

$(x+1)^2 = x^2+2x+1 = x^2+1$

$x^2+1 = x \pmod{x^2+x+1}$

$(x+1)^2$ in $\frac{F[x]}{x^2+1}$ defined over $GF(2)$

$(x+1)^2 = x^2+2x+1 = x^2+1 = 0 \pmod{(x^2+1)}$

$(x+1)^2$ in $\frac{F[x]}{x^2+x+1}$ $GF(3)$

$(x+1)^2 = x^2+2x+1$

$\Rightarrow x^2+2x+1 = x \pmod{x^2+x+1}$

If polynomial $f(x)$ has degree 'n', then the ring of polynomials represented by $\frac{F[x]}{f(x)}$ over $GF(q)$ having degree $\leq n-1$

Size of the ring $= q^n$

$\Rightarrow \frac{F[x]}{x^2+x+1}$ over $q(2) \Rightarrow q^n = 0, 1, x, x+1$

$= 2^2 = 4$

Additive Inverse Table

	0	1
$\frac{F[x]}{f(x)}$		

Multiplicative Inverse Table

	0
$\frac{F[x]}{f(x)}$	

$\frac{F[x]}{x^2+1}$ over

Additive Inv

Multiplicative Inverse Table

If $f(x)$ we can can rep are eli deg g term

Additive Inverse Table

	0	1	x	$x+1$
0	0	1	x	$x+1$
1	1	0	$x+1$	x
x	x	$x+1$	0	1
$x+1$	$x+1$	x	1	0

Multiplicative inverse for all non zero element
$1-1$ $x-x+1$ $x+1-x$

$\frac{F[x]}{f(x)}$ →

Multiplicative Inverse Table

	0	1	x	$x+1$
0	0	0	0	0
1	0	(1)	x	$x+1$
x	0	x	$x+1$	(1)
$x+1$	0	$x+1$	(1)	x

For any product going beyond a degree of 1, we get the remainder after dividing by x^2+x+1

$\frac{F[x]}{x^2+1}$ over GF(2) size $= 2^2 = 4$

Components = 0, 1, x & $x+1$

Additive Inverse Table is same as above.

Multiplicative Inverse Table

	0	1	x	$x+1$
0	0	0	0	0
1	0	(1)	x	$x+1$
x	0	x	(1)	$x+1$
$x+1$	0	$x+1$	$x+1$	0

Multiplicative Inverse for non zero elements:
$1-1$ and $x-x$ but none for $x+1$
→ not a field.

If $f(x)$ becomes irreducible (i.e. not factorisable) we consider field in $F[x]$ is said to be reducible if we can represent it in product of two polynomial $a(x)$ & $b(x)$, both of which are elements of $F[x]$ and $\deg f(x) \geq \deg(a(x))$ and $\deg f(x) > \deg g(x)$. If this condition is not fulfilled then $f(x)$ is termed as irreducible polynomial.

Prime polynomial :- Irreducible polynomial of degree atleast 'one'

1) $f(x)$ has a linear factor $(x-a)$ if and only if $f(a) = 0$ for $x = a$, where a is a field element [elements determined by q of GF(q)]
2) $f(x)$ is a polynomial in $F[x]$ of degree 2 or 3 over GF(q) is irreducible if $f(x) \neq 0$ for all a in GF(q)
3) over any field, $x^n - 1 = (x-1)(x^{n-1} + x^{n-2} \dots + 1)$

For $f(x) = x^3 - 1$ defined over GF(2)
$\rightarrow f(x) = (x-1)(x^2+x+1)$
$g(0) = 1$
$g(1) = 1$

Same question over GF(3)
$\rightarrow f(x) = (x-1)(x^2+x+1)$
$g(0) = 1 \; ; \; g(1) = 0 \; ; \; g(2) = 1 \quad \Rightarrow g(x) = (x-1)(x-1)$

If the no of elements changes then the equation may become irreducible.

$\left(\frac{F[x]}{f(x)}\right)_{\text{ring}} \rightarrow$ field $\quad f(x) \rightarrow$ prime polynomial

$S(x)$ belongs to ring $\left(\frac{F[x]}{f(x)}\right)$
$\deg S(x) < \deg f(x)$
$GCD\{f(x), s(x)\} = a(x) f(x) + b(x) s(x)$
$a(x), b(x)$ defined over GF(q)
If $f(x)$ is irreducible then $GCD = 1 = R_{f(x)}[1]$
$= R_{f(x)}[a(x) f(x) + b(x) s(x)]$
$= R_{f(x)}[a(x) f(x)] + R_{f(x)}[b(x) s(x)]$
$= 0 + R_{f(x)}[R_{f(x)}[b(x)] R_{f(x)}[s(x)]]$

$1 = R_{f(x)}[R_{f(x)}\{b(x)\} \cdot R_{f(x)}\{s(x)\}]$

$1 = R_{f(x)}[R_{f(x)}\{b(x)\} \cdot s(x)]$

$= R_{f(x)}\{b(x)\} \cdot s(x) = 1$

$\Rightarrow R_{f(x)}\{b(x)\} = s(x)^{-1}$

$f(x) \rightarrow GF(2)$

Prime polynomial of deg n

$GF[q^n] \rightarrow q^n$ polynomials

$f[x] = x^3 + x + 1 \quad GF(2)$

$n = 3 \quad q = 2 \rightarrow GF(q^n) = GF(2^3) - GF(8)$

$\hookrightarrow 0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1$

Cyclic Code

$F[x] = x^n - 1 \qquad R_n = \frac{F[x]}{f(x)}$

$a(x), b(x), g[x], h(x)$

$\hookrightarrow deg_g < deg_g(f(x))$

1) $x^n = 1 \pmod{x^n - 1}$

$x^{n+1} = x$

$x^{n+2} = x^2$

2) A codeword can be uniquely represented by a polyn

If block length $= n \rightarrow n$ elements

$c \rightarrow c_0 c_1 c_2 c_3 \ldots c_{n-1}$

$c(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 \ldots c_{n-1} x^{n-1}$

$c = 2077735$

$= 2 + 0x + 7x^2 + 7x^3 + 3x^4 + 5x^5$

3) Any polynomial $g(x)$ multiplied by x corresponds to

$g(x) \rightarrow C_0 C_1 C_2 \cdots C_{n-1} = C_0 + C_1 x + C_2 x^2 \cdots C_{n-1} x^{n-1}$

$xC(x) = C_0 x + C_1 x^2 + C_2 x^3 + \cdots C_{n-1} x^n$

$= C_{n-1} + C_0 x + C_1 x^2 + C_2 x^3 \cdots C_{n-2} x^{n-1}$ [$x^n - 1$, $C_{n-1} x^n$, C_{n-1}]

cyclic shift to the right.

A code $C \rightarrow R_n$ is a cyclic code if & only if C satisfies the following condition

a) $a(x), b(x) \in C$; $a(x) + b(x) \in C$

b) $a(x) \in C$, ~~$r(x) \in R_x$~~ $r(x) \in R_x$

then ~~$a(x) r(x) \in C$~~ $a(x) r(x) \in C$

$R_n = \frac{F[x]}{x^n - 1}$

$r(x) = r_0 + r_1 x + r_2 x^2 + r_3 x^3 + \cdots r_{n-1} x^{n-1}$

$a(x) = C(x)$; $x C(x) \in C$

Method of Generating Cyclic Code

Take any polynomial $g(x)$ in R_n

Obtain a set of polynomial by multiplying $g(x)$ by all possible polynomials in R_x

The resultant set of polynomials corresponds to the set of codeword. The block length is n.

$g(x) = 1 + x^2$ in R_3, GF(2)

$R_3 = \frac{F[x]}{x^3 - 1}$; $R(x) = r_0 + r_1 x + r_2 x^2$ Total combo $= 2^3 = 8$

$0, 1, x, x^2, 1+x, 1+x^2, x+x^2, 1+x+x^2$

Multiply $g(x)$ with above functions :-

$g(x) \times 0 = 0$ → 000 ; $g(x) \times (1+x) = 2 + x + x^2 = x + x^2$

$g(x) \times 1 = 1 + x^2$ → 101 ; $g(x) \times (1+x^2) = 1 + x + 2x^2 = 1 + x$

$g(x) \times x = x + x^3 \rightarrow 1 + x$ → 110 ; $g(x) \times (x + x^2) = 1 + 2x + x^2 = 1 + x^2$

$g(x) \times x^2 = x^2 + x^4 \rightarrow x^2 + x$ → 011 ; $g(x)(1 + x + x^2) = 2x + 2x^2 = 0$

Let C be a non zero (n, k) code (cyclic) in R_n

a) There exists a unique monic polynomial $g(x)$ in R_n of smallest degree.

b) The cyclic code C consists of all multiples of the generator polynomial $g(x)$ by polynomial of degree $k-1$ or less.

$c \rightarrow a(x)\, g(x)$

$\hookrightarrow \leq k-1 \quad \deg a(x) \leq k-1$

c) $g(x)$ is a factor of $x^n - 1$.

A cyclic code C may contain polynomials other than generator polynomials that also generates C

$\deg g(x) = n-k$

$n = 3$; $R_3 = F[x]/(x^3-1)$ over GF(2)

$x^3-1 = (x-1)(1+x+x^2)$ where $1+1=0 \Rightarrow -1=1$

$= (x+1)(1+x+x^2)$

$g(x)$	Code (Polynomial)	Code (Binary)
	$\{R_3\}$	[000, 001, 101, 111]
1	$\{0, x+1, x^2+x, x^2+1\}$	[000, 011, 110, 101]
$x+1$	$\{0, x^2+x+1\}$	{000, 111}
x^2+x+1	$\{0\}$	{000}

Another way of encoding cyclic code from generator code:

$c(x) = I(x)\, g(x)$

(where $I(x) \rightarrow$ Information Polynomial $\Rightarrow 0101 \Rightarrow x^3+1$ $V(x) = c(x)+e(x)$

$e(x)$ [Error Polynomial]

Syndrome Polynomial $s(x) = R_{g(x)}[v(x)]$

$= R_{g(x)}[c(x) + e(x)]$

$= R_{g(x)}[c(x)] + R_{g(x)}[e(x)]$

$= 0 + R_{g(x)}[e(x)]$

$g(x) = x^2+1$ over GF(3) $n = 4$ $R_n = \frac{F[x]}{x^4-1}$

deg of $g(x) = n-k = 2$

$= 4-k = 2 \Rightarrow k = 2$ $(n,k) = (4,2)$

No. of distinct codes $= q^k = 3^2 = 9$

i	i(x)	C(x) = i(x) g(x)	C (binary)
00	0	0	0000
01	1	x^2+1	0101
02	2	$2x^2+2$	0202
10	x	x^3+x	1010
11	$x+1$	x^3+x^2+x+1	1111
12	$x+2$	x^3+2x^2+x+2	1212
20	$2x$	$2x^3+2x$	2020
21	$2x+1$	$2x^3+2x^2+2x+1$	2121
22	$2x+2$	$2x^3+2x^2+2x+2$	2222

The cyclic shift of any codeword results in another valid codeword. Min hamming distance = 2 ∴ it can detect one error and correcting zero error.

The valid codeword polynomial is divisible by generator polynomial, so we can detect more no. of errors than suggested by the min distance of the code.

Matrix Description of Cyclic Codes

$R_n = \frac{F[x]}{x^n-1}$ $C(x) \in R_n$ $C(x) = i(x)g(x)$

$g(x)$ is a factor of x^n-1

deg $g(x) \le n-k$ and deg $i(n) \le (k-1)$

$C \in R_n$

$g(x) = g_0 + g_1 x + g_2 x^2 \cdots g_r x^r$

$\deg g(x) = r$ $\quad G \rightarrow k \times n$ matrix

Date:

Page No.

$$G = \begin{bmatrix} g_0 & g_1 & \cdots & g_r & \cdots & 0 & 0 & 0 \\ 0 & g_0 & g_1 & \cdots & g_{r-1} & g_r & 0 & 0 \\ 0 & 0 & g_0 & \cdots & g_{r-2} & g_{r-1} & g_r & 0 \\ \vdots & & & & & & & \\ 0 & 0 & 0 & \cdots & g_0 & \cdots & \cdots & g_r \end{bmatrix} \quad k = n - r$$

$g(x)$

$x\,g(x)$

$x^2 g(x)$

$x^{n-r-1} g(x)$

$q(x) \in R_n \quad c(x) = q(x) g(x) \quad \deg c(x) < n$

$\deg(q(x)) < n - r$

$q(x) = q_0 + q_1 x + q_2 x^2 + \cdots q_{n-r-1} x^{n-r-1}$

$c(x) = q(x) g(x) = q_0 g(x) + q_1 x g(x) + \cdots q_{n-r-1} x^{n-r-1} g(x)$

$G \rightarrow R \times n$ where $k = n - r \Rightarrow \boxed{r = n - k}$

Find the generator matrix of all ternary codes, where

$n = 4, \quad R_4 = \dfrac{F(x)}{x^4 - 1}$ $\quad GF(3)$

$(x^n - 1) = g(x) h(x) \Rightarrow x^4 - 1 = (x-1)(x^3 + x^2 + x + 1)$

$= (x-1)(x+1)(x^2+1)$

$g(x)$	(n, k)
1	$(4, 4)$
$(x-1)$	$(4, 3)$

G

$[I_4] \rightarrow 0 + 1 + 0x + 0x^2 +$

$$k+4 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \leftarrow I_4$$

you may replace -1 by 2 as $GF(3)$

$$\begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \quad x - 1 = -1 + 1 \times x + 0$$

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad x^2 + 1 = 1 +$$

$(x-1)(x^2+1)$ $(4,1)$ $[-1 \quad 1 \quad -1 \quad 1]$

$= x^2+x-x^2-1$

$(x+1)(x^2+1)$ $(4,4)$ $[1 \quad 1 \quad 1 \quad 1]$

x^2+x+x^2+1

$(x^4-1)=0$ $(4,1)$ $[0\,0\,0\,0]$

$CH^T = 0$ where $H \rightarrow$ parity checker

$C \in R_n \quad h(x)g(x) \rightarrow R_n = x^n-1 = 0$ $\quad (x^n-1) \bmod (x^n-1) = 0$

$C(x) = a(x)g(x) \quad a(x) \in R_n$

$C(x)h(x) = a(x)g(x)h(x) = a(x)\cdot 0 = 0$

$h(x) \rightarrow$ parity check polynomial.

$c(x) \longrightarrow v(x) \Rightarrow c(x)+e(x)$

$S(x) \rightarrow v(x)h(x) = c(x)h(x) + e(x)h(x) = e(x)h(x)$

$(x^7-1) \longrightarrow R_7 = \frac{F[x]}{x^7-1} \qquad g(x) = x^3+x+1$

Find out parity checker polynomial.

$(x^7-1) = (x-1)(x^3+x+1)(x^3+x^2+1)$

$(x^7-1) = g(x)h(x) \Rightarrow h(x) = (x-1)(x^3+x^2+1)$

$x^n-1 = g(x)h(x)$

$\downarrow \quad \downarrow \quad \downarrow$

deg n, deg n-k, deg k $\quad \therefore h(x) = h_0 + h_1x + h_2x^2 + \cdots h_{k-1}x^{k-1}$

$G \rightarrow k \times n \quad H \rightarrow n-k \times n$

$$H = \begin{matrix} \\ n-k \end{matrix} \begin{bmatrix} h_k & h_{k-1} & \cdots & h_0 & \cdots & 0 \\ 0 & h_k & h_{k-1} & h_1 & h_0 & \cdots & 0 \\ 0 & 0 & h_k & h_{k-1} & \cdots & h_1 & h_0 \\ 0 & 0 & & & h_k & \cdots & h_0 \end{bmatrix}$$

(n columns)

For above problem find out generator & parity check matrix.

$g(x) = 1 + x + 0x^2 + 0x^3 + 0x^4 + 0x^5 + 0x^6$

$h(x) = (x-1)(x^3+x^2+1)$

$= x^4 + x^3 + x$

$\quad - x^3 - x^2 - 1$

$= x^4 - x^2 + x - 1$

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}_{4\times 7}$$

$$\begin{bmatrix} -1 & 1 & -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}_{3\times 7}$$

Quasi Cyclic Code

An $[n, k]$ code becomes quasi-cyclic for some value m that is co-prime with n. The polynomial $x^m c(x)$ calculated over mod $(x^n - 1)$ is a valid codeword polynomial if $c(x)$ is a valid polynomial codeword.

$[12, 4]$ $G =$

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$m = 3$

shift of 2 units in each row.

multiply with x^3 we get valid codeword.

Any cyclic code can be punctured to form quasi-cyclic code by dropping every m^{th} symbol where m is a factor of n.

The quasi-cyclic code could be a shortened code if none of the dropped symbol are checked symbol.

All dropped symbol are data symbol which can be first set to zero.

check

Shortened Code:-

An [n, k] linear code is called a proper shortened cyclic code if it is obtained by deleting n consecutive places from an (n+m, k+m) cyclic code.
In case of shortened code, an unused symbol is reset to zero before transmission.

Burst Error Correction:

C → 10 kB/s 01000 111 0101000 0 1011 01

b = 00000 | 11 011 01111 | 000 0000
← 10 bits →

A cyclic burst of length 't' is a vector whose non zero components are among t successive components and of which 1st and last bit must be non-zero.

Burst error $e(x) = x^i b(x)$
b(x) → polynomial corresponding to cyclic burst
i → position the burst started (from ee → 5 above)

$S(x) = Rg(x)[e(x)]$

Cyclic Redundancy Code :- Only detect no correct

T → n no. of bits
D → k no. of information bit
F = (n-k) no of FCS bits (Frame Check Sequence)
P = (n-K+1) bits predetermined divisor

T = k + n-k → FCS
(k → information bits)

$\frac{I}{P}$ ~ 0 remainder for valid code
else non-zero for error.

$D = 1010001101$ (transmitted codeword)

$n-k = 5$

$P = 110101$

$2^{n-k}D$ → 2× Binary sequence → shift bits left

$2^{n-k}D = 2^5 D \rightarrow 10100011010\underline{00000}$ → appends 5 zeros

$\frac{2^{n-k}D}{P} = Q + R \rightarrow$ FCS

$R = \frac{2^5 D}{P} = 01110$

↑ append 0's left to keep same value

$T = \underbrace{1010001101}_{D} \underbrace{01110}_{FCS}$

Circuit Implementation:

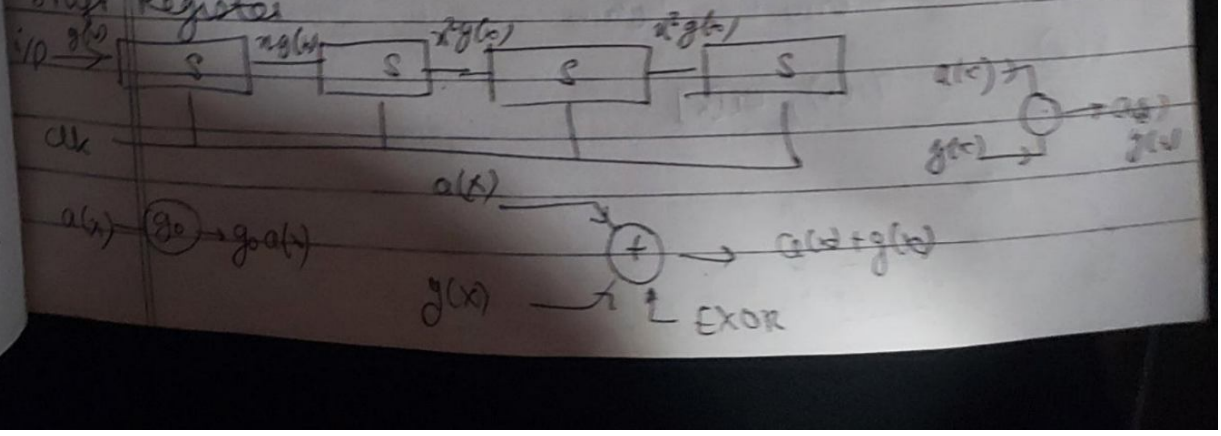
$c(x) = g(x)\,h(x)$

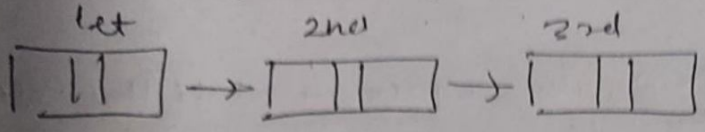
$(n-1)\quad(k-1)\quad(n-k)$

$a(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1}$

$g(x) = g_0 + g_1 x + g_2 x + \cdots$

$c(x) = a_0 g(x) + a_1 g(x) + \cdots$





min 3 clock cycles required to shift from one number to next.

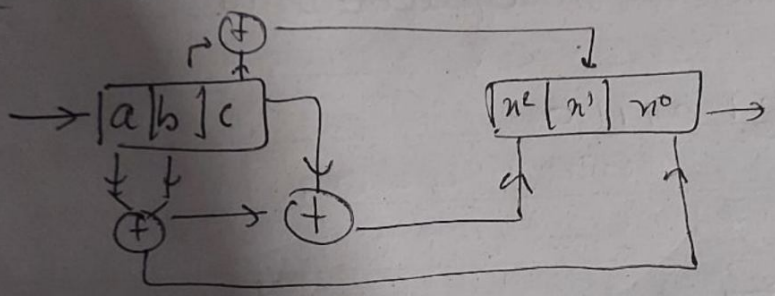
$P(x) = x^3 + x + 1$

$GF(2^3) \rightarrow 0, 1, x, 1+x, x^2, x^2+1, x^2+x, x^2+x+1$

$\hookrightarrow ax^2 + bx + c$

$(ax^2+bx+c)(x^2+x) \mod p(x)$

$= (a+b+c)x^2 + (b+c)x + (a+b) \mod p(x)$

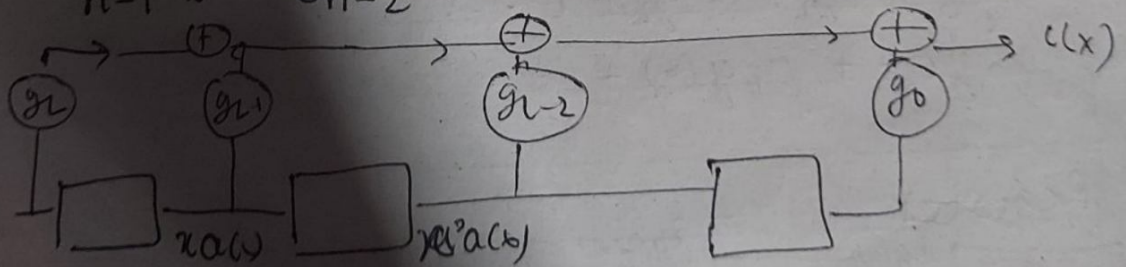


$c(x) = a(x)\,g(x)$

$a(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \dots + a_0$

$g(x) = g_L x^L + g_{L-1}x^{L-1} + \dots g_0$

$c(x) = c_{n-1}x^{n-1}\ c_{n-2}x^{n-2} + \dots c_0$



$\frac{a(x)}{b(x)} = q(x),\ r(x)$