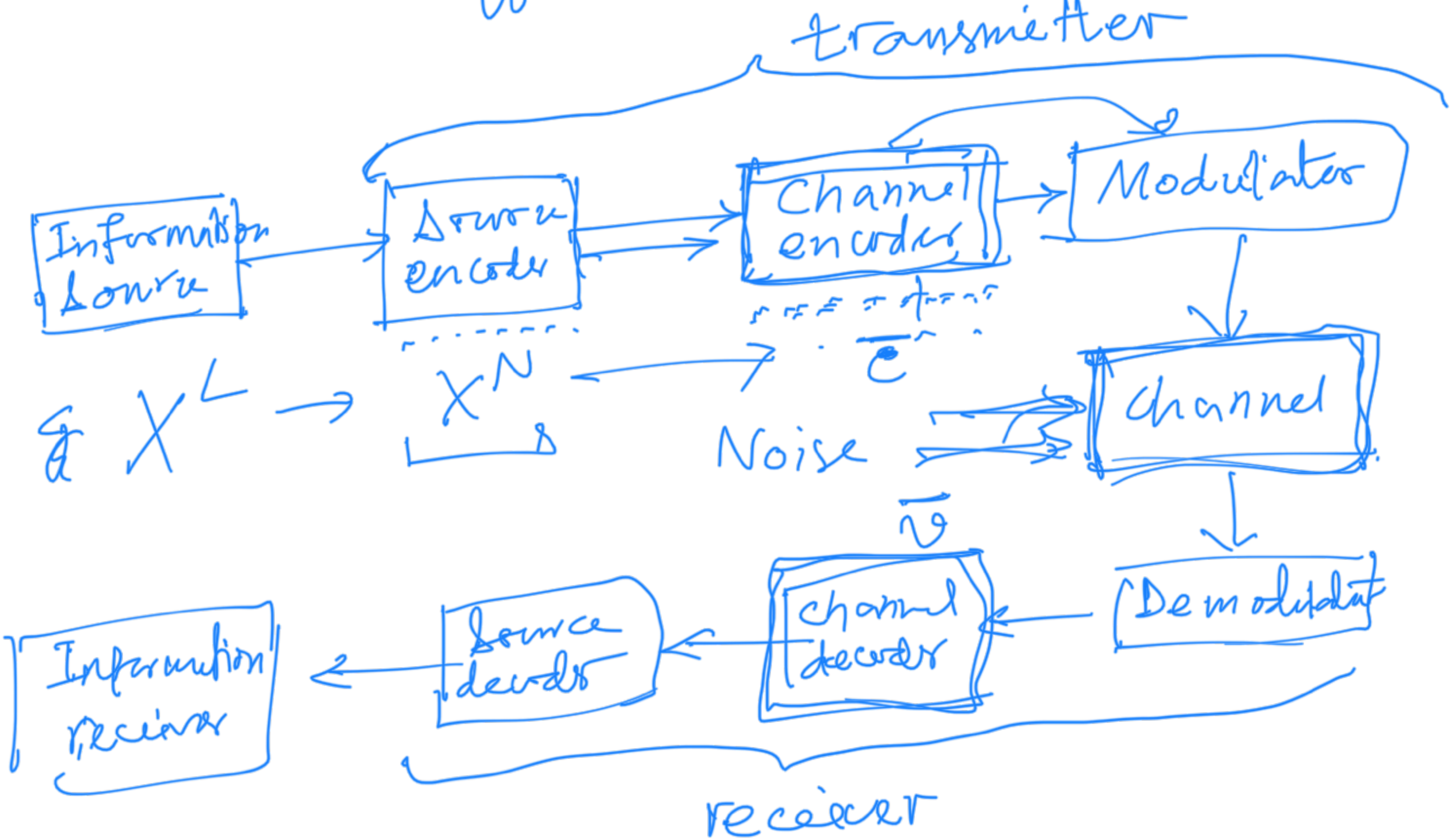
# Information and Coding theory.

**Lec-17.**

Friday, tutorial classes are converted to lecture classes.



101 → 111

Error control codes
- error - ~~correct~~ detecting code
- error - correcting code

Error control codes →
- block codes
- convolution codes.

Source seq.

[ 2 1 1 ] → fixed length codeword

information bits

[ . . . . . . . ] ← add certain adhoc bits

101 → [101|01]

codeword

"Block" Code:

a set of codewords/words with a well-defined structure (mathematical) $\mathbb{Z}_p$

and each codeword/word is a string of fixed/constant number of bits.

A codeword has two parts

- information bits
- parity-check bits/parity bits/check bits

For example, carries information     no information

| | |
|---|---|
| information bits | parity bits |

$\underbrace{\qquad\qquad\qquad}_{\text{information codewords}}$

The job of the channel encoder — to decide the parity bits & appends it to ~~another~~ the information word.

k-bit | r-bits
--- | ---
information | parity

$n = k + r$ — blocklength ($n$), information length ($k$)

the format in which information bits are altogether then we can easily identify it in the ~~latcor~~ block codeword is called 'systematic form'

— non-systematic.

otherwise it is called ...

$\boxed{(n, k) \text{ block code}}$

Example,

(5, 4) block code with $\boxed{\text{odd parity}}$ → # of 1s in each codeword is odd.

$n = 5$

$\boxed{k = 4}$

Code words.

1. (0 0 0 0 1)
2. (0 0 0 1 0)
3. (0 0 1 0 0)
4. (0 0 1 1 1)
5. (0 1 0 0 0)
6. (0 1 0 1 1)
7. (0 1 1 0 1)
8. (0 1 1 1 0)
9. (1 0 0 0 0)
10. (1 0 0 1 1)
(1 0 1 0 1)

Redundant ~~co~~ words.

(0 0 0 0 0)
(0 0 0 1 1)
(0 0 1 0 1)
(0 0 1 1 0)
(0 1 0 0 1)
⋮

$2^n = 2^5 = 32$

$2^k = 2^4 = 16$

11. ( 1 0 ... )
12. ( 1 0 1 1 0 )
13. ( 1 1 0 0 1 )
14. ( 1 1 0 1 0 )
15. ( 1 1 1 0 0 )
16 ( 1 1 1 1 1 ) .

(a) # of codewords in an (n, k) blockcode is $= 2^k$, with a desired mathematical structure.

(b) There are $2^n - 2^k$ redundant words which do not have the structure of the codewords.
help to detect errors.

Measure of redundancy for (n, k) block code.

Code rate: $R = \frac{k}{n} = \frac{\text{Information length}}{\text{block length}}$

$$\text{"}0\text{"} \leq R \leq 1.$$

Information word $\equiv$ $\bar{i} = (i_1, i_2, \ldots, i_k)$

$i_j \in \{0, 1\}, 1 \leq j \leq k$

Codeword $\equiv$ $\bar{c} = (i_1, i_2, \ldots, i_k, p_1, p_2, \ldots, p_r)$

$\equiv (c_1, c_2, \ldots, c_n)$

where, $c_j = \begin{cases} i_j & \text{if } 1 \leq j \leq k \\ p_{j-k} & \text{if } k < j \leq n \end{cases}$

$(1\,0\,0\,0\,1) \rightarrow (0\,1\,1\,1\,0)$

maximum # of number of errors is $= n$

$\bar{e} = (e_1, e_2, \ldots, e_n)$

error vector:

$$e_j = \begin{cases} 1 & \text{if there is an error at the } j\text{th position} \\ 0 & \text{if the position is error free} \end{cases}$$

represented by $\bar{e}$.

~~Then~~ If there is an error occurred then the received codeword must be

$$\boxed{\bar{v}} = \bar{c} + \boxed{\bar{e}},$$

'modulo 2 addition'

If we receive a codeword with error then $\bar{v} = (v_1, v_2, \ldots, v_n)$

$\bar{v} \neq \bar{c}$.

i.e. for some '$j$', $v_j \neq c_j$

i.e. $v_j = 1$ and $c_j = 0$

or $v_j = 0$ and $c_j = 1$

$$v_j + c_j = v_j - c_j = e_j$$

i.e. $v_j = c_j + e_j$.

i.e. $\boxed{\bar{c} = \bar{v} + \bar{e}}$

**Remark.** The decoder has no prior knowledge about $\bar{c}$.

The decoder only knows $\bar{v}$

received word/decoder input.

I

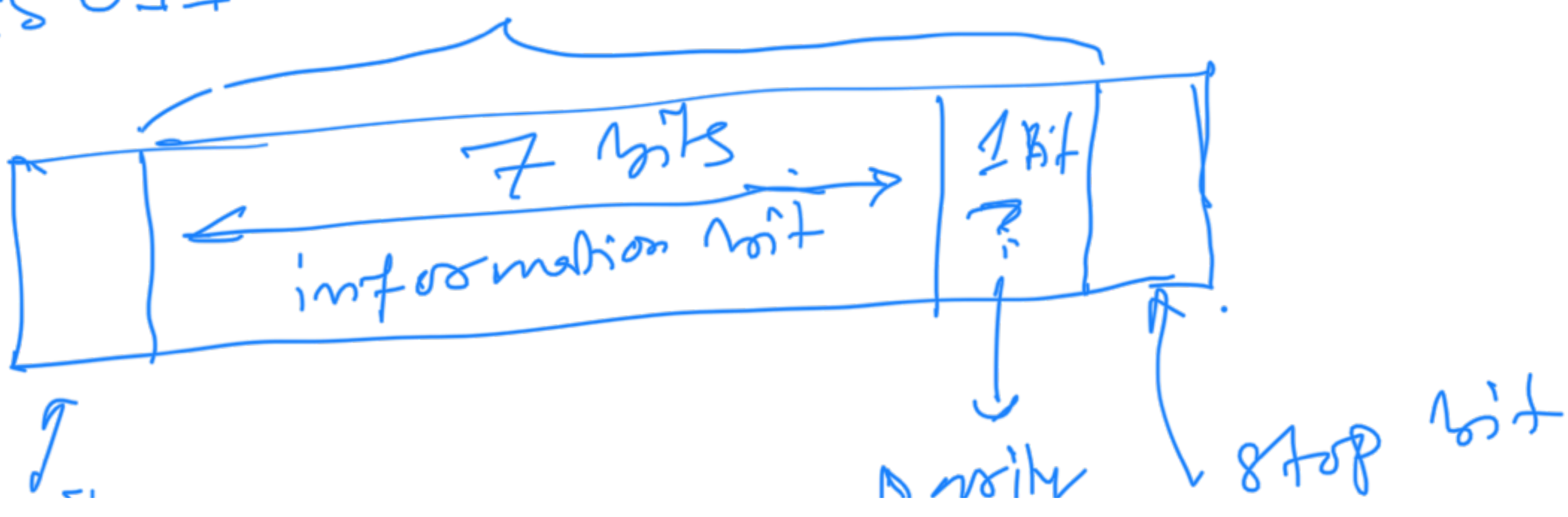
and the decoder can check if $\bar{v}$ has the mathematical structure of the codeword.

For Error-correcting code, the job of the decoder is to estimate the original codeword from the received ~~code~~ codeword $\bar{r}$.

## Single parity-check codes.

(n, k) block code, where $n = k + 1$.

ASCII : (8, 7) block code



Start bit                                   Parity bit

$\bar{i} = (i_1, i_2, \ldots, i_k)$

Then the parity bit corresponding to $\bar{i}$ is defined as

$$p = i_1 + i_2 + \cdots + i_k .$$

$$\boxed{\bar{c}} = (i_1, i_2, \ldots, i_k, \boxed{p})$$

For example, if $\bar{i} = (1\,0\,1\,1)$

then $\bar{c} = (1\,0\,1\,1\,1)$

Suppose the received word is

$\bar{v} = (v_1, v_2, \ldots, v_n)$, $n = k+1$

code word

To check whether $v$ is a codeword
the decoder determines

$$s = v_1 + v_2 + \dots + v_n$$

then $s = 0$ when $\overline{v}$ has even parity
$s = 1$ when $\overline{v}$ has odd parity.

If we have even-parity code:

Then the decoder it is enough to check
if $s = 0$ to decide $\overline{v}$ as a
codeword.

$\overline{c} = (1\,1\,0\,0\,0\,|\,0) \longrightarrow \overline{v} = (0\,0\,0\,0\,|\,0)$

... this model, the decoder is able
... single-bit error.

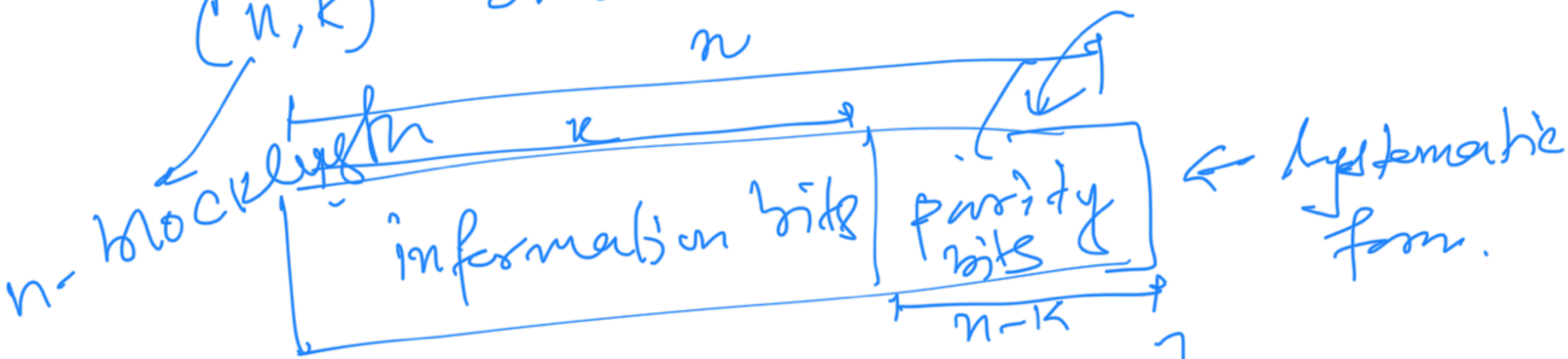
... any single

to detect ~~so~~ ... number of errors.

→ it can detect odd number of errors.

Lec-18

(n, k) block code




$[\quad\quad]$

$(a_1, \ldots, a_n) \quad 2^n, \quad a_i \in \{0,1\}$

parity-check sum — $\sum a_i \in \{0,1\}$

$\rightarrow (b_1, b_2, \ldots, b_m)$

Error detection & error correction.

$\{0,1\}^n \ni \text{'}\bar{c}\text{'} \rightarrow$ original codeword

$\{0,1\}^n \ni \text{'}\bar{v}\text{'} \rightarrow$ received codeword.

$\{0,1\}^n \ni \text{'}\bar{e}\text{'} \rightarrow$ error

$$\bar{v} = \bar{c} + \bar{e}$$

There are three possible outcomes

- correct decoding
- decoding error
- decoding ~~error~~ failure

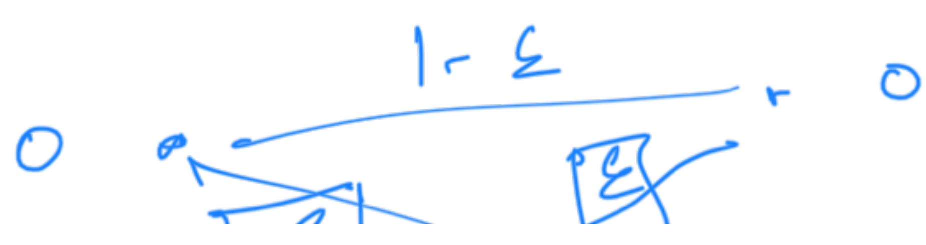
— decoding

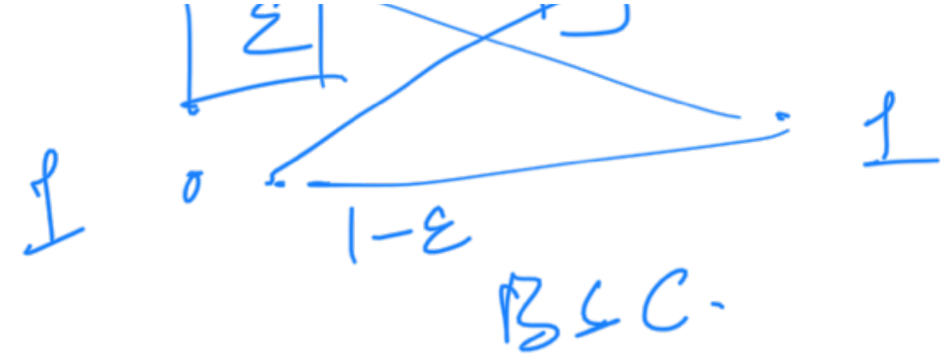
$P_c$ = the probability that the decoder gives the correct codeword $c$

$P_e$ = the prob. that the decoder gives incorrect codeword

$P_f$ = the prob. that the decoder fails to give a code.

$$P_c + P_e + P_f = 1.$$






'ε' = error probability

Then having 'j' bits in error out of n bits. — $^nC_j = \frac{n!}{(n-j)!\,j!}$

prob. of getting j errors in an n-bit codeword.

← $\boxed{P_j} = {}^nC_j\, p^j (1-p)^{n-j}$

$\bar{c} \longrightarrow \bar{v}$

$\boxed{P_c} = P_0 = {}^nC_0\, p^0 (1-p)^{n-0}$

$= {}^{n}C_{0} \, (1-\epsilon)^{n}$.

$$p_e = P_2 + P_4 + P_6 + \cdots + P_{n'}, \quad n' = n \text{ if } n \text{ is even}$$

$$= \sum_{\substack{j \le 2 \\ (\text{even})}}^{n'} {}^{n}C_j \, \epsilon^j (1-\epsilon)^{n-j} \qquad n' = n-1 \text{ if } n \text{ is odd}$$

$$p_f = P_1 + P_3 + P_5 + \cdots + P_{n'}, \quad n' = n \text{ if } n \text{ is odd},$$

$$= \sum_{j \ge 2, \text{ odd}}^{n'} {}^{n}C_j \, \epsilon^j (1-\epsilon)^{n-j} \qquad n' = n-1 \text{ otherwise}$$

$$\boxed{P_i + P_e + P_f = 1} \quad n.N.$$

---

## Hamming Code

$(n, k)$ block code

**Exp.** $(7, 4)$ - Hamming code

$(7, 4)$: $7 \to n$, $4 \to k$

| information | parity bits |
|---|---|
| $i$ | $P$ |

$\bar{i} = (i_1 \; i_2 \; i_3 \; i_4) \leftarrow$ information code

$n - k = 7 - 4 = 3 \rightarrow$ parity bits.

$p_1 = \boxed{i_1 + i_2 + i_3}$

$p_2 = i_2 + i_3 + i_4$

$p_2$

$p_3 = \underline{i_1 + i_2 + i_4}$

$\bar{c} = (i_1\ i_2\ i_3\ i_4\ p_1\ p_2\ p_3)$

# of code words $= 2^4 = 16$.

| Information words | Codewords |
|---|---|
| $0 \leftarrow$ $\bar{i}_0 = (0\ 0\ 0\ 0)$ | $\bar{c}_0 = (0\ 0\ 0\ 0\ 0\ 0\ 0)$ $\leftarrow$ |
| $1 \rightarrow$ $\bar{i}_1 = (0\ 0\ 0\ 1)$ | $\bar{c}_1 = (0\ 0\ 0\ 1\ 0\ 1\ 1)$ |
| $\bar{i}_2 = (0\ 0\ 1\ 0)$ | $\bar{c}_2 = (0\ 0\ 1\ 0\ 1\ 1\ 0)$ |
| $\vdots$ | $\vdots$ |

Decoding Scheme?

$$\bar{v} = (v_1\ v_2\ v_3\ v_4\ v_5\ v_6\ v_7)$$

$$v_1 = i_1,\ v_2 = i_2,\ v_3 = i_3,\ v_4 = i_4,\ v_5 = p_1,\ v_6 = p_2,\ v_7 = p_3$$

Compute 3 parity check ~~s~~ sums

$s_1 = (v_1 + v_2 + v_3) + v_5$

$s_2 = (v_2 + v_3 + v_4) + v_6$

$s_3 = (v_1 + v_2 + v_4) + v_7$

$\bar{s} = (s_1, s_2, s_3)$ — Error Syndrome

~~then~~ $\bar{s} = 0$ when there are no errors.

Exp. $\bar{c} = (1011000)$

Suppose $\bar{v} = (1011000)$

$s_1 = 0, \; s_2 = 0, \; s_3 = 0$

Note: Zero error syndrome does not mean there is

does not ...
no error.

Exp. $c = (0\ 1\ 0\ 1\ 1\ 0\ 0)$

$\boxed{e = (0\ 1\ 1\ 0\ 0\ 0\ 1)}$

$\bar{r} = \bar{c} + \bar{e}$

$= (0\ 0\ 1\ 1\ 1\ 0\ 1)$

$s_1 = (0+0+1)+1 = 0$

$s_2 = (0+1+1)+0 = 0$

$s_3 = (0+0+1)+1 = 0$

$\bar{s} = 0$

$\bar{s}$

Syndrome table for (7,4) Hamming code

| Error pattern $(e_1 e_2 \ldots, e_7)$ | Syndrome $(s_1, s_2, s_3)$ |
|---|---|
| (0,000000) | (000) |
| (0000001) | (001) |
| (0000010) | (010) |
| (0000100) | (100) |
| (0001000) | (011) |
| (0010000) | (110) |
| (0100000) | (111) |
| (1000000) | (101) |

Check H.W.

$c \leftarrow v + e$

## Decoding Algo.

1. Compute the syndrome $\bar{s}$

syndrome table

2. From the syndrome ~~creat~~ determine the error $\bar{e}$

3. $\bar{c} = \bar{v} + \bar{e}$

(7,4) Hamming Code is a single-error-correcting code.

2. $(n,k)$ - Hamming code

BCH code

$Z_2 = \{0,1\}$

$n = 2^r - 1$

$k = 2^r - 1 - r$

when $r \geq 3$, $r = n - k$.

Exp. $r = \text{"}4\text{"} \Rightarrow n = 15$

$k = 11$

$(15, 11)$ - Hamming code.

$$\begin{cases} p_1 = i_1 + i_2 + i_3 + i_4 + i_6 + i_8 + i_9 \\ p_2 = i_2 + i_3 + i_4 + i_5 + i_7 + i_9 + i_{10} \\ p_3 = i_3 + i_4 + i_5 + i_6 + i_8 + i_{10} + i_{11} \\ p_4 = i_1 + i_2 + i_3 + i_5 + i_7 + i_8 + i_{11} \end{cases}$$

$$c = (i_1\ i_2 \ldots i_{11}\ p_1\ p_2\ p_3\ p_4).$$

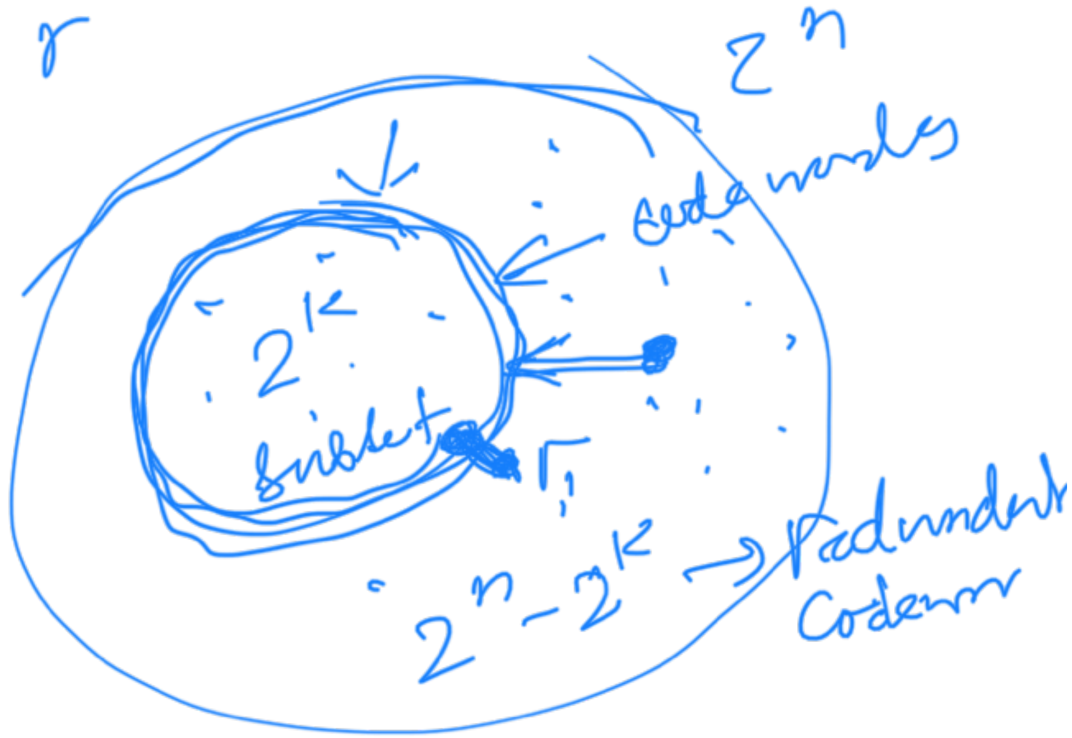
$(2^r - 1,\ 2^r - 1 - r)$ Hamming code.

$$n - k = r$$

H.W.

$$\boxed{\{0,1\}^n \text{ is a vector space over } \{0,1\}.}$$

$\dim = n$




Hamming weight of a codeword $\bar{v}$

$wt(\bar{v}) = w(\bar{v}) = \|v\|$ = the # of nonzero entries of $\bar{v}$.

Hamming distance betw. two codewords.

$d(\bar{v}_1, \bar{v}_2)$ = the # of place where

$\bar{v}_1$ and $\bar{v}_2$ differ.

$$\bar{v}_1 = (0\ 1\ 1\ 0\ 1\ 0)$$
$$\bar{v}_2 = (1\ 0\ 1\ 0\ 0\ 0)$$
$$d(\bar{v}_1, \bar{v}_2) = 3.$$

**Minimum distance of a block code:**

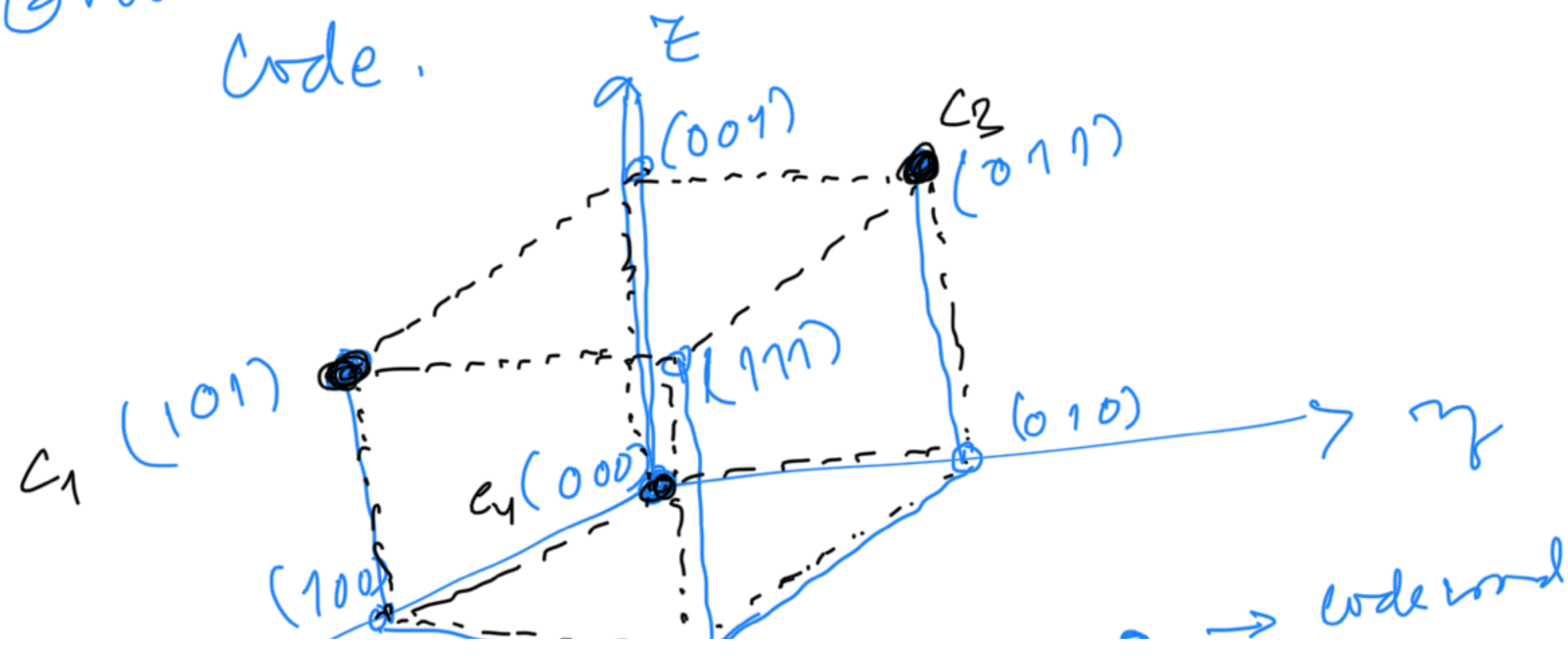
$d_{min}$ = the smallest distance betn all the codewords

$$\begin{array}{c|cccc} & c_1 & \cdots & & c_l \\ \hline c_1 & 0 & \cdot & \cdot & \cdot \\ \vdots & & & d(c_i, c_j) & \\ c_l & & & & \end{array}$$

Remark: finding $d_{min}$ is computationally challenging; however for linear codes it has a formula

Geometric view to look at code.

Consider $(3,2)$ $\boxed{\text{even}}$ parity code.



$c_2$

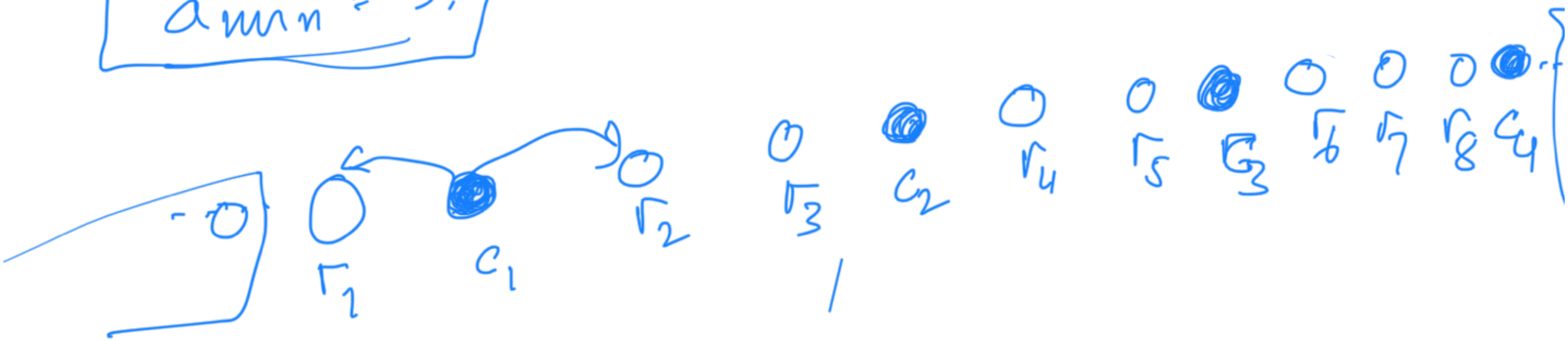
$(1\,1\,0)$ → redundant codeword.

$x$

$c = (c_x, c_y, c_z)$ $\boxed{d_{min} = 2}$

Minimum-distance decoding scheme.

$\boxed{d_{min} = 3.}$




$c_1$ incurs a single error.

$c_1$ incurs a double errors.