# 20SK – Signals and Codes

## Lecture 10 – Error correcting codes (2018/12/03)

Topics discussed:

- Principles of (forward) error correction
- Information rate, minimum code distance, detection and correction capabilities
- Binary arithmetic modulo 2
- Linear codes: definition, basis vector, computation. General properties. Generator and parity-check matrix
- Systematic code
- Hard-decision decoding

The relevant literature is [1, chapter 3], [2, chapters 10 and 12] and [3, chapter 4].

### Resources


[1] Morelos-Zaragoza, R. H.: The Art of Error-Correcting Coding. 2nd edition, John Wiley & Sons, 2006, 263pp.

[2] Adámek, J: Foundations of Coding: Theory and Applications of Error-Correcting Codes with an Introduction to Cryptography and Information Theory. Wiley Interscience, 1991, 352 pp.

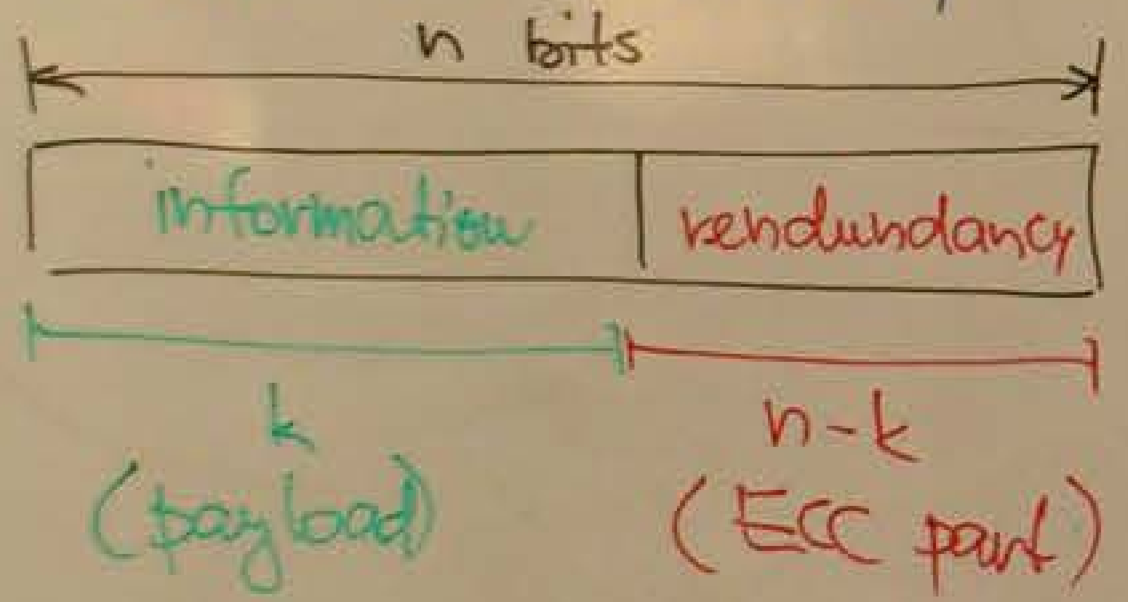
[3] Moon, T. K.: Error Correction Coding – Mathematical Methods and Algorithms. Wiley Interscience, 2005, 756 pp.

# ERROR CORRECTING CODES (ECC)

- variable length codes
- block codes

1. detecing an error
2. correct the error, if possible

→ add REDUNDANCY




(n, k) - code

**Information rate:** $\frac{k}{n}$

**Minimum code distance:**

$d(w_1, w_2)$ ... number of bits where $w_1$ and $w_2$ differ

$$\min_{\substack{w_1 \in K \\ w_2 \in K \\ w_1 \neq w_2}} d(w_1, w_2) = \boxed{d_{min}}$$

⇒ minimal number of bits to flip for obtaining another code word

**Detection capabilities:** code detects t-ary error iff $t < d_{min}$

**Example:**

parity (3,2)

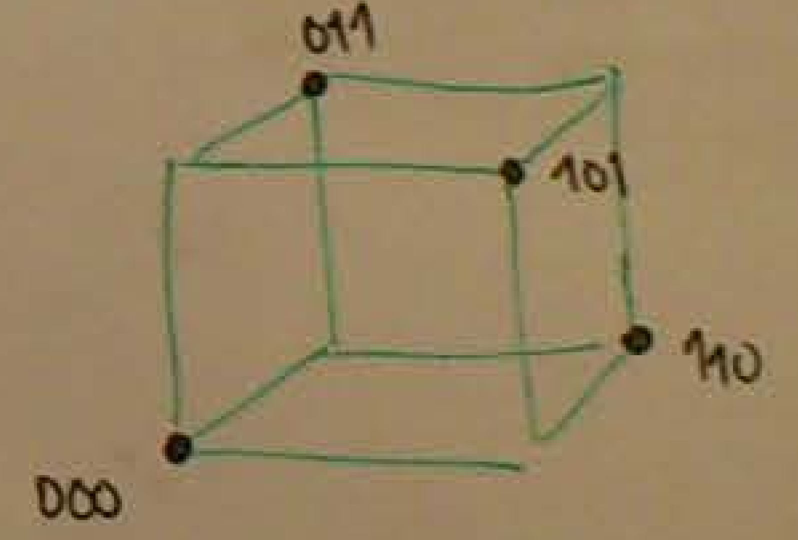
000
011
101
110

i.r. $= \frac{2}{3}$

$K \subset V = \{0,1\}^3$

$K$ ... 8 possible words ($2^3$)
... 4 codewords

$d(011, 110) = 2$
$d(000, 110) = 2$
$d_{min} = 2$




repetition code (3,1)

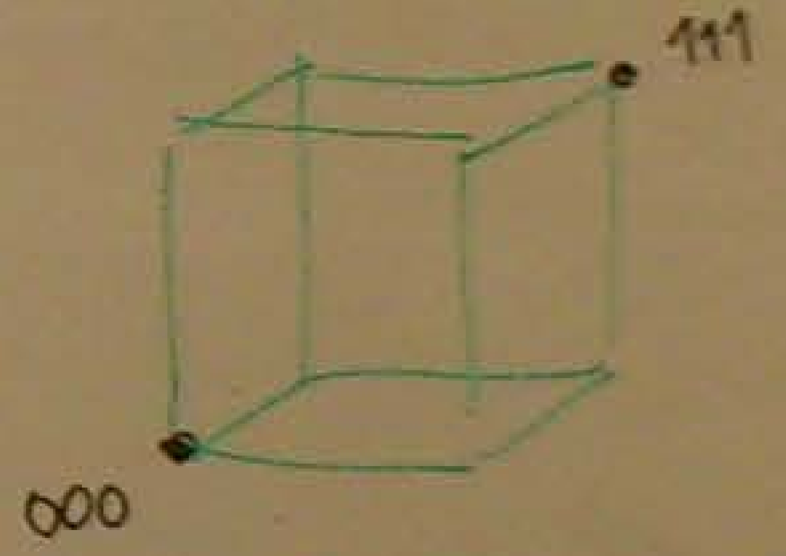
000
111

i.r. $= \frac{1}{3}$

2 codewords

$d(000, 111) = 3$

$d_{min} = 3$



# ERROR CORRECTING CODES (ECC)

Correction capabilities: code corrects t-ary error iff $\boxed{2t < d_{min}}$

↓

difficult to design by hand

↓

! use linear algebra !

→ $\boxed{\text{LINEAR CODES}}$

| ⊕ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

XOR

| · | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

AND

Information rate: $\frac{k}{n}$

Minimum code distance:

$d(w_1, w_2)$ ... number of bits where $w_1$ and $w_2$ differ

$$\min_{\substack{w_1 \in K \\ w_2 \in K \\ w_1 \neq w_2}} d(w_1, w_2) = \boxed{d_{min}} \qquad (2^k - 1)\, 2^{k-1}$$

⇒ minimal number of bits to flip for obtaining another code word

Detection capabilities: code detects t-ary error iff $\boxed{t < d_{min}}$

Example.

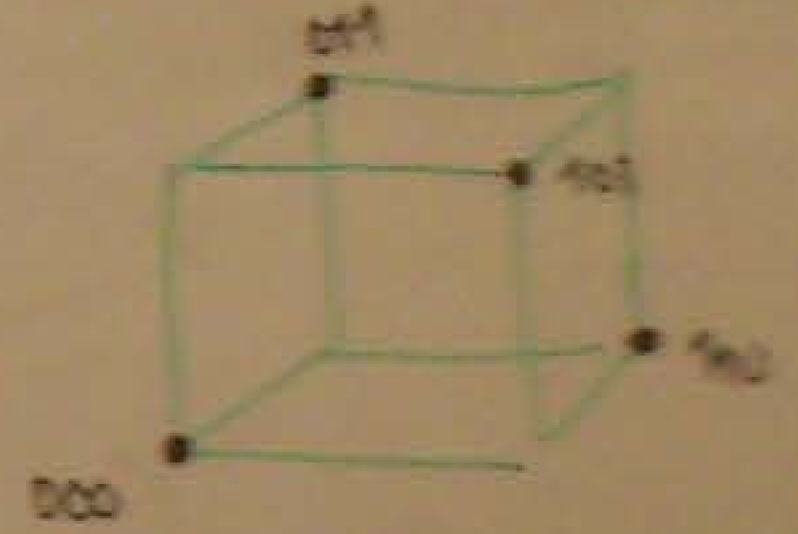
parity (3,2)

000   i.r. $= \frac{2}{3}$
011
101
110

$K \subset V = \{0,1\}^3$

$K$ ... 8 possible words $(2^3)$
... 4 codewords

$d(011, 110) = 2$
$d(000, 110) = 2$
$d_{min} = 2$



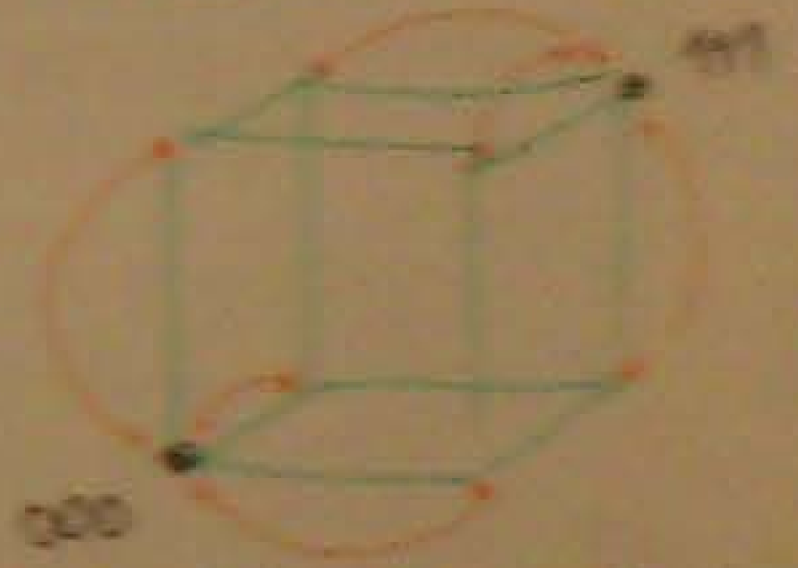

repetition code (3,1)

000   i.r. $= \frac{1}{3}$
111

2 codewords

$d(000, 111) = 3$

$d_{min} = 3$




take the closest code word

# LINEAR CODES

- all $(n,k)$-codes $\mathbb{K} \subset V_n = \{0,1\}^n$ (sub-space of dimension $\underline{k}$)

$\Rightarrow$ sub-space has to have $\underline{k}$ basis vectors $\{\vec{v}_0, \vec{v}_1, \dots, \vec{v}_{k-1}\}$

$\downarrow$

$2^k$ code words

$2^n$ possible words

code word $\vec{v} = (v_0, v_1, \dots, v_{n-1})$

plaintext $\vec{u} = (u_0, u_1, u_2, u_3, \dots, u_{k-1})$

$$\boxed{\vec{v} = u_0 \vec{v}_0 \oplus u_1 \vec{v}_1 \oplus \dots \oplus u_{k-1} \vec{v}_{k-1}}$$

$$\boxed{\vec{v} = \vec{u} \cdot G}$$

generator matrix

$$G = \begin{pmatrix} \vec{v}_0 \\ \vec{v}_1 \\ \vdots \\ \vec{v}_{k-1} \end{pmatrix}$$

$G_{k \times n}$

All codes can be described by a homogeneous system of equations for $v_0, v_1, \dots, v_{n-1}$

Verification of a code-word:

parity check matrix $H$

$\boxed{G \cdot H^T = \vec{0}}$ or $\boxed{\vec{v} \cdot H^T = \vec{0}}$ syndrome

$\vec{u} \cdot G \cdot H^T - 0$

## Example:

parity $(3,2)$

000
011
101
110

$\vec{u} = \{00, 01, 10, 11\}$

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

$$(000) = (00) \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

$$(011) = (01) \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

$\vdots$

$v_0 \oplus v_1 \oplus v_2 = 0$

repetition code $(3,1)$

000
111

$\vec{u} = \{0, 1\}$

$G = (1\ 1\ 1)$

$111 = 1 \cdot (111)$

$000 = 0 \cdot (111)$

$v_0 \oplus v_1 = 0$

$v_1 \oplus v_2 = 0$

$v_0 \oplus v_2 = 0$

# LINEAR CODES

- all $(n,k)$-codes $K \subset V_n = \{0,1\}^n$ (sub-space of dimension $\underline{k}$)

$\Rightarrow$ sub-space has to have $\underline{k}$ basis vectors $\{\vec{v}_0, \vec{v}_1, \dots, \vec{v}_{k-1}\}$

$\downarrow$

$2^k$ code words

$2^n$ possible words

code word $\vec{v} = (v_0, v_1, \dots, v_{n-1})$

plaintext $\vec{u} = (u_0, u_1, u_2, u_3, \dots, u_{k-1})$

| $u_0 u_1$ | $v_2 v_3$ |
|---|---|

$$\boxed{\vec{v} = u_0 \vec{v}_0 \oplus u_1 \vec{v}_1 \oplus \dots \oplus u_{k-1} \vec{v}_{k-1}}$$

$$\boxed{\vec{v} = \vec{u} \cdot G} \qquad G = \begin{pmatrix} \vec{v}_0 \\ \vec{v}_1 \\ \vdots \\ v_{k-1} \end{pmatrix}$$

generator matrix

$G_{k \times n}$

All codes can be described by a homogeneous system of equations for $v_0, v_1, \dots, v_{n-1}$

Verification of a code-word:

parity check matrix $H$

$$\boxed{G \cdot H^T = \vec{0}} \quad \text{or} \quad \boxed{\vec{v} \cdot H^T = \vec{0}}$$

$\vec{u} \cdot G \cdot H^T - 0$

syndrome

Example: $(4,2)$-code

$G = [\, I_{k \times k} \; P_{k \times n-k} \,]$

$H = [\, P^T_{n-k \times k} \; I_{n-k \times n-k} \,]$

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

$$G_{sys} = \left[\begin{array}{cc:cc} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right]$$

$I \qquad P$

systematic code (separate information and parity blocks)

$$P = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \rightarrow P^T = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

$$H_{sys} = \left[\begin{array}{cc:cc} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{array}\right]$$

$$\vec{v}' = (1110) \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$\vec{u} = (01)$

$$\vec{v} = (01) \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} = (0110)$$

$$\vec{v} \cdot H^T = (0110) \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$\hookrightarrow \vec{v}$ is a code word!

Ex: parity

00 0
01 1
10 1
11 0

$V = \{0,1\}^3$

$\downarrow$

8 words

K is only 4 of them

$n = 3$
$k = 2$

K ... (3,2) code

rate ... $\frac{2}{3}$

## Error correcting codes

- ~~variable length codes~~ ✓
- fixed length codes

$\downarrow$

block codes

n bits

| information | redundant |
|---|---|

k bits | n-k bits

Error detection & correction $\Rightarrow$ adding REDUNDANCY

$\Rightarrow$ for n-bit length of the code, we are using only $k < n$ for information

information rate $\frac{k}{n}$

$\Rightarrow$ $(n,k)$ code

Ex: repetition code

0 000
1 111

$V = \{0,1\}^3$

$\downarrow$

8 words

K ... only 2 of them

$n = 3$
$k = 1$

K ... (3,1) code

rate ... $\frac{1}{3}$

Ex. 1: Parity (3,2)-code

000 — $k=2 \Rightarrow$ 2 basis vectors

011 — $\vec{v} = u_0 \cdot \vec{v}_0 \oplus u_1 \cdot \vec{v}_1$

101 — $\vec{v} = u_0 \cdot (0,1,1) \oplus u_1 \cdot (1,0,1)$

110

---

$\vec{v} = (v_0, v_1, v_2)$

$v_0 \oplus v_1 \oplus v_2 = 0$

→ Correction of an t-ary error: $d_{min} > 2t$

# LINEAR CODES

XOR

| $\oplus$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | $\boxed{0}$ |

AND

| $\cdot$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

- all $(n,k)$ codes are subspaces of $V_n = \{0,1\}^n$ of dimension $k$

→ a subspace of dim. $k$ will have $k$ basis vectors $\{\vec{v}_0, \vec{v}_1, \dots, \vec{v}_{k-1}\}$

→ codeword: $\boxed{\vec{v} = u_0 \cdot \vec{v}_0 \oplus u_1 \cdot \vec{v}_1 \oplus \dots \oplus u_{k-1} \cdot \vec{v}_{k-1}}$ → information bits

$\vec{u} = \{u_0, u_1, \dots, u_{k-1}\}$ … „plaintext"

Ex. 2: Repetition code (3,1)-code

000 — $k=1 \Rightarrow$ 1 basis vector

111

$\vec{v} = u_0 \cdot \vec{v}_0$

$\vec{v} = u_0 \cdot (1,1,1)$

---

$\vec{v} = (v_0, v_1, v_2)$

$v_0 \oplus v_1 = 0$

$v_0 \oplus v_2 = 0$

Ex. 1: Parity (3,2)-code

000 $k=2 \Rightarrow$ 2 basis vectors

011 $\vec{v} = u_0 \cdot \vec{v_0} \oplus u_1 \cdot \vec{v_1}$

101 $\vec{v} = u_0 \cdot (0,1,1) \oplus u_1 \cdot (1,0,1)$

110

$\vec{v} = (v_0, v_1, v_2)$

$v_0 \oplus v_1 \oplus v_2 = 0$

$\vec{u} = (u_0, u_1)$

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

Observation: $\vec{v} = u_0 \vec{v_0} \oplus u_1 \vec{v_1} \oplus \dots \oplus u_{k-1} \vec{v}_{k-1}$ is equivalent to

$$\boxed{\vec{v} = \vec{u} \cdot G}$$ → this is just a system of lin. equations

↳ generator matrix of the code

where $G = \begin{pmatrix} \vec{v_0} \\ \vec{v_1} \\ \vdots \\ \vec{v}_{k-1} \end{pmatrix}$

$\vec{u}, \vec{v_0}, \vec{v_1}, \dots, \vec{v}_{k-1}$ are row vectors

Verification of a code word: For every $G$ we have a parity check matrix $H$ such that

$$\vec{v} \cdot H^T = 0 \quad \text{or} \quad G \cdot H^T = 0$$

Ex. 2: Repetition code (3,1)-code

000 $k=1 \Rightarrow$ 1 basis vector

111

$\vec{v} = u_0 \cdot \vec{v_0}$

$\vec{v} = u_0 \cdot (1,1,1)$

$\vec{v} = (v_0, v_1, v_2)$

$v_0 \oplus v_1 = 0$

$v_0 \oplus v_2 = 0$

$\vec{u} = (u_0)$

$$G = (1 \; 1 \; 1)$$

## Ex. 1: Parity (3,2)-code

$\boxed{000}$   $k=2 \Rightarrow$ 2 basis vectors

011   $\vec{v} = u_0 \cdot \vec{v}_0 \oplus u_1 \cdot \vec{v}_1$

101   $\vec{v} = u_0 \cdot (0,1,1) \oplus u_1 \cdot (1,0,1)$

110

---

$\vec{v} = (v_0, v_1, v_2)$

$v_0 \oplus v_1 \oplus v_2 = 0$

---

$\vec{u} = (u_0, u_1)$

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

## Definition: Linear code $\mathbb{K}$ is a code where

1. if $a \in \mathbb{K}$ and $b \in \mathbb{K}$, then $a \oplus b \in \mathbb{K}$
2. for $\lambda \in \{0,1\}$ and $a \in \mathbb{K}$, $\lambda \cdot a \in \mathbb{K}$

$\Rightarrow$ every lin. code contains a code-word $\{0\}^n$

## Hamming weight of a code-word:

$$w_H(\vec{v}) = \sum_{i=0}^{n-1} v_i = d_H(\vec{v}, 0)$$

in general, $d_{min}$ needs $2^{k-1}(2^k - 1)$ comparisons

For a linear code

$$d_{min}(\mathbb{K}) = \min_{\substack{\vec{v} \in \mathbb{K}, \\ \vec{v} \neq 0}} w_H(\vec{v})$$

$\Rightarrow 2^k$ comparisons in order to find $d_{min}$

## Ex. 2: Repetition code (3,1)-code

$\boxed{000}$   $k=1 \Rightarrow$ 1 basis vector

111

$\vec{v} = u_0 \cdot \vec{v}_0$

$\vec{v} = u_0 \cdot (1,1,1)$

---

$\vec{v} = (v_0, v_1, v_2)$

$v_0 \oplus v_1 = 0$

$v_0 \oplus v_2 = 0$

---

$\vec{u} = (u_0)$

$$G = (1\ 1\ 1)$$

$$\vec{v}\cdot H^T = 0$$

$$\Downarrow$$

$$H_{sys} = \left( P^T_{(n-k)\times k} \;\vdots\; I_{(n-k)\times(n-k)} \right)$$

→ it is easy to derive $H_{sys}$ from $G_{sys}$ or vice versa

## Example: (4,2)-code

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

$$G_{sys} = \left(\begin{array}{cc|cc} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right) \Rightarrow P = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

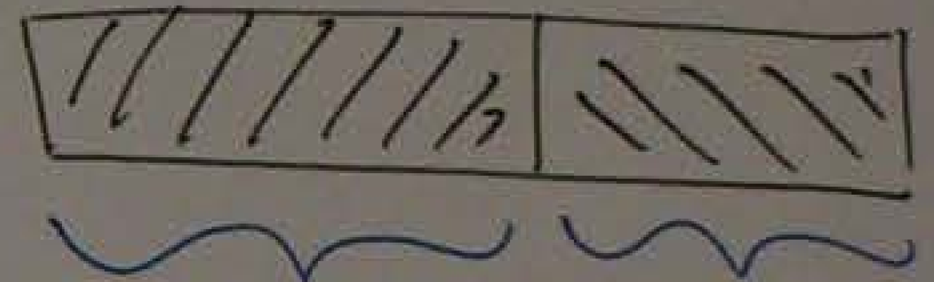
$$\begin{array}{cc} I_{2\times 2} & P_{2\times 2} \end{array}$$

$$\Rightarrow P^T = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \qquad \Rightarrow H_{sys} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

systematic code: $\vec{v} = (\overbrace{v_0, v_1, \ldots v_{k-1}}^{\text{inf. bits}} \ldots v_n)$ we have

$v_0 = u_0,\ v_1 = u_1, \ldots, v_{k-1} = u_{k-1}$

## Systematic coding




## Systematic linear code:

$$G \longrightarrow G_{sys}$$

$$G_{sys} = \left( I_{k\times k} \;\vdots\; P_{k\times(n-k)} \right)$$

## Corruption:

$\vec{v} = (0\,1\,1\,0)$ received as

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

$\vec{v}_{Rec} \cdot H_{sys}^{T} =$ syndrome

$$(1\,1\,1\,0)\begin{pmatrix}1&1\\1&0\\1&0\\0&1\end{pmatrix} = \boxed{(1\,1)}$$

this is not a code word

## Example: (4,2)-code

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

$$G_{sys} = \left(\underbrace{\begin{matrix}1&0\\0&1\end{matrix}}_{\mathbb{I}_{2\times2}}\;\middle|\;\underbrace{\begin{matrix}1&1\\1&0\end{matrix}}_{\mathbb{P}_{2\times2}}\right) \Rightarrow \mathbb{P} = \begin{pmatrix}1&1\\1&0\end{pmatrix}$$

$$\Rightarrow \mathbb{P}^{T} = \begin{pmatrix}1&1\\1&0\end{pmatrix} \quad \Rightarrow H_{sys} = \begin{pmatrix}1&1&1&0\\1&0&0&1\end{pmatrix}$$

systematic code: $\vec{v} = (\overbrace{v_0, v_1, \dots v_{k-1}}^{\text{inf. bits}} \dots v_n)$ we have

$v_0 = u_0,\ v_1 = u_1, \dots, v_{k-1} = u_{k-1}$

## How does it work?

$\vec{u} = (0,1)$

$$\vec{v} = \boxed{(0\;1)}\begin{pmatrix}1&0&1&1\\0&1&1&0\end{pmatrix} = (\boxed{0\;1}\;1\;0)$$

**Parity check:** $\vec{v} \cdot H_{sys}^{T}$

$$(0\,1\,1\,0)\begin{pmatrix}1&1\\1&0\\1&0\\0&1\end{pmatrix} = (0\,0)$$

(4,2)-code

n = 4

k = 2

$4 \leq 2^2 - 1 = 3$

→ does not correct single.

(7,4)-code

$7 \leq 2^3 - 1 = 7$

→ perfect code

(8,4)

$8 \leq 2^4 - 1 = 15$

→ corrects single err.
not perfect

# „PERFECT" CODES

- the shortest possible code for given detection & correction capabilities (it does not always exist for given n and k)

→ **Hamming bound:** For every single-error correction code it holds

$$n \leq 2^{n-k} - 1$$

and for a perfect code

$$n = 2^{n-k} - 1$$

added redundancy

Ex. (3,1)-code

n = 3
k = 1

$3 \leq 2^2 - 1 = 3$

⇒ perfect single-error-corr. code

(4,3)-code

n = 4
k = 3

$4 \leq 2 - 1$

⇒ does not correct any error

(4,2)-code

$n = 4$

$k = 2$

$4 \leq 2^2 - 1 = 3$

→ does not correct anyth.

(7,4)-code

$7 \leq 2^3 - 1 = 7$

→ perfect code

(8,4)

$8 \leq 2^4 - 1 = 15$

→ corrects single err.
not perfect

# „PERFECT" CODES

$$n = 2^m - 1$$

(3,1), (5,2), (6,3), (7,4),

(9,5), (10,6), (11,7), ... (15,11) → (16,12) does not correct a single err.

⇒ we need (17,12)

| m | n | k |
|---|---|---|
| 1 | 1 | ∅ |
| 2 | 3 | 1 |
| 3 | 7 | 4 |
| 4 | 15 | 11 |
| 5 | 31 | 26 |

and so on...

⇓

Hamming codes = perfect codes for correcting single errors for m bits of redundancy

$n = 2^m - 1$ $\quad d_{min} = 3$

$k = 2^m - m - 1$

Ex. (3,1)-code

$n = 3$

$k = 1$

$3 \leq 2^2 - 1 = 3$

⇒ perfect single-error-corr. code

(4,3)-code

$n = 4$

$k = 3$

$4 \leq 2 - 1$

⇒ does not correct any error