Digital Communication Systems ECS 452

Asst. Prof. Dr. Prapun Suksompong

prapun@siit.tu.ac.th

5. Channel Coding

Error control code/coding/strategy

Office Hours:

BKD, 6th floor of Sirindhralai building

Tuesday	14:20-15:20
Wednesday	14:20-15:20
Friday	9:15-10:15

Digital Communication Systems ECS 452

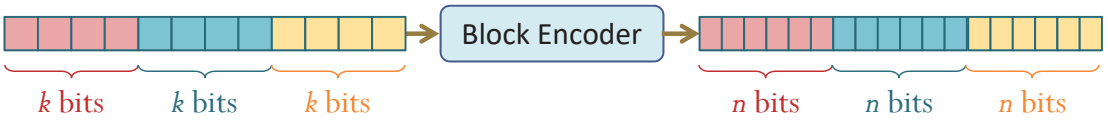
Asst. Prof. Dr. Prapun Suksompong

prapun@siit.tu.ac.th

5.1 Binary Linear Block Codes

Review: Block Encoding

- We mentioned the general form of channel coding over BSC.
- In particular, we looked at the general form of **block codes**.



Code length

"Dimension" of the code = # bits in a (block of) message

- **(n,k) codes**: n-bit blocks are used to conveys k-info-bit blocks
  - n-bit blocks → codewords
  - k-info-bit blocks → messages
- **Assume $n > k$**
- **Rate**: $R = \frac{k}{n} < 1$

[bpcu]

Max. achievable rate

Recall that the capacity of BSC is $C = 1 - H(p)$.
For $p \in (0,1)$, we also have $C \in (0,1)$.
Achievable rate is < 1.

crossover probability of BSC

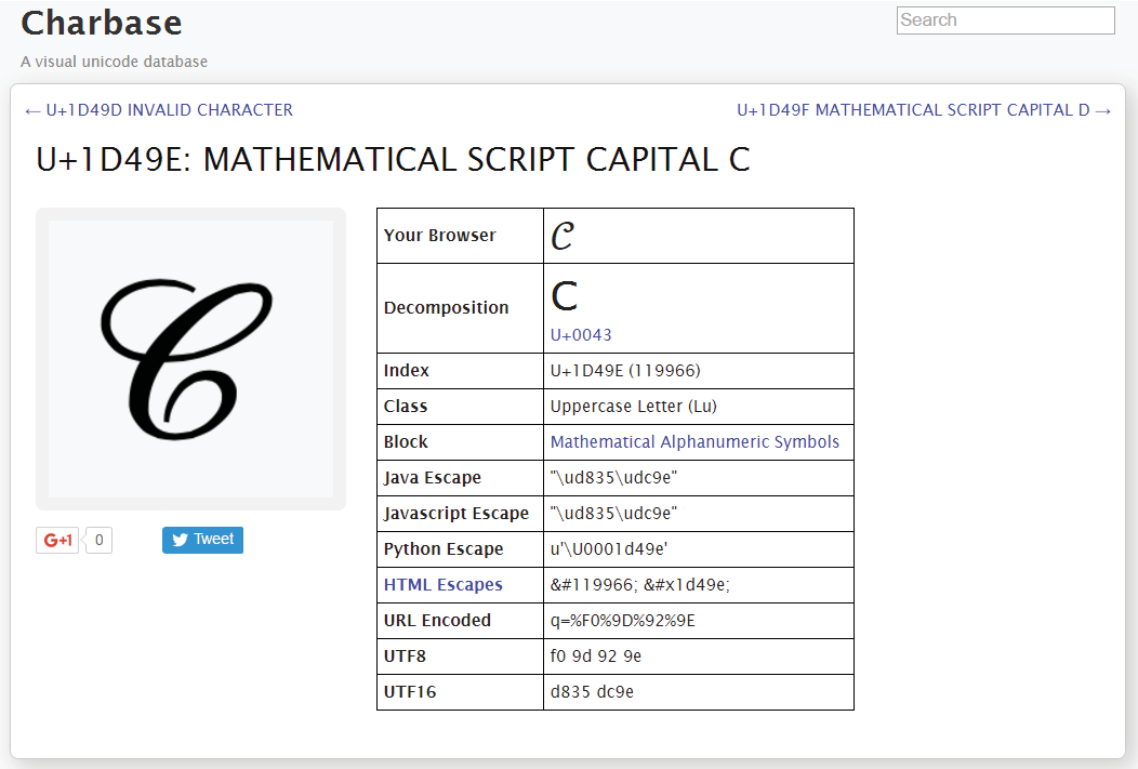
$\mathcal{C}$ ← New symbol

- $\mathcal{C}$ = the collection of all codewords for the code considered
- Each n-bit block is selected from $\mathcal{C}$.
- The message (data block) has k bits, so there are 2^k possibilities.
- A reasonable code would not assign the same codeword to different messages.
- Therefore, there are 2^k (distinct) codewords in $\mathcal{C}$.
- Ex. Repetition code with $n = 3$, $k = 1$

$\mathcal{C} = \{000, 111\}$

$R = \frac{k}{n} = \frac{1}{3}$

MATHEMATICAL SCRIPT CAPITAL C



Charbase

A visual unicode database

Search

← U+1D49D INVALID CHARACTER

U+1D49F MATHEMATICAL SCRIPT CAPITAL D →

U+1D49E: MATHEMATICAL SCRIPT CAPITAL C

Your Browser	𝒞
Decomposition	C U+0043
Index	U+1D49E (119966)
Class	Uppercase Letter (Lu)
Block	Mathematical Alphanumeric Symbols
Java Escape	"\ud835\udc9e"
Javascript Escape	"\ud835\udc9e"
Python Escape	u"\U0001d49e"
HTML Escapes	𝒞 𝒞
URL Encoded	q=%F0%9D%92%9E
UTF8	f0 9d 92 9e
UTF16	d835 dc9e

G+1 0 Tweet

[http://www.charbase.com/1d49e-unicode-mathematical-script-capital-c]

GF(2)

5 mod 2 = 1

↑

remainder of the division of 5 by 2

- The construction of the codes can be expressed in matrix form using the following definition of addition and multiplication of bits:

$\oplus$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

- These are **modulo-2** addition and **modulo-2** multiplication, respectively.
- The operations are the same as the **exclusive-or** (**XOR**) operation and the **AND** operation.
  - We will simply call them addition and multiplication so that we can use a matrix formalism to define the code.
- The two-element set {0, 1} together with this definition of addition and multiplication is a number system called a **finite field** or a **Galois field**, and is denoted by the label **GF(2)**.

GF(2)

- The construction of the codes can be expressed in matrix form using the following definition of addition and multiplication of bits:

$\oplus$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

- Note that $x \oplus 0 = x$

  $x \oplus 1 = \bar{x}$

  $x \oplus x = 0$

The above property implies $-x = x$

By definition, "-x" is something that, when added with x, gives 0.

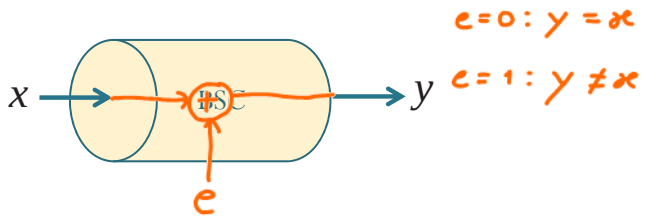
$1\times1+1\times0=1$

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

$1\times1+1\times1=2$

- Extension: For vector and matrix, apply the operations to the elements the same way that addition and multiplication would normally apply (except that the calculations are all in GF(2)).

BSC and the Error Pattern

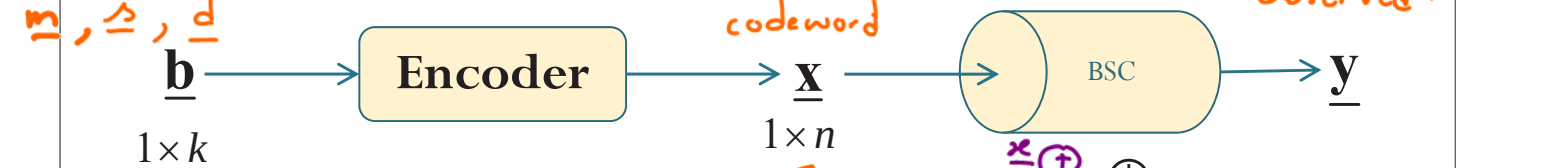


- Again, to transmit k information bits, the channel is used n times.

(uncoded data)
m, ≙, d

(coded data)
code vector
codeword

received
observed
} vector



message data } { vector block word

Ex. $\underline{x} = 0\,1\,0\,1\,0$

$\underline{y} = 0\,1\,1\,1\,1$

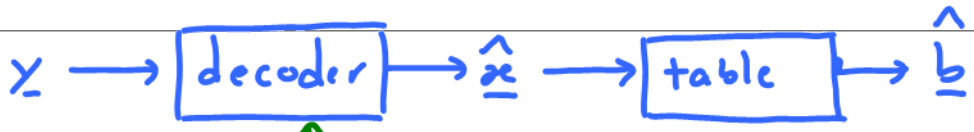
$\underline{e} = \underline{x} \oplus \underline{y} = 0\,0\,1\,0\,1$

$= 0\,0\,1\,0\,1$

$\underline{x} \oplus \underline{y} = \underline{x} \oplus \underline{e}$

error pattern

Its nonzero elements mark the positions of transmission error in $\underline{y}$



Review: Block Decoding

- In this chapter, we assume the use of **minimum distance decoder**.
  - $\hat{\underline{\mathbf{x}}}(\underline{\mathbf{y}}) = \arg\min_{\underline{\mathbf{x}}} d(\underline{\mathbf{x}}, \underline{\mathbf{y}})$
- Recall
  1. The **MAP decoder** is the optimal decoder.
  2. When the codewords are equally-likely, the **ML decoder** the same as the MAP decoder; hence it is also **optimal**.
  3. When the **crossover probability** of the BSC p is < 0.5, ML decoder is the same as the minimum distance decoder.
- Also, in this chapter, we will focus
  - less on probabilistic analysis,
  - but more on explicit codes.

equally-likely codewords

MAPD (optimal) ≡ MLD ≡ min dist. decoder

BSC with $p < 0.5$

Error pattern with less 1s are more likely to occur.

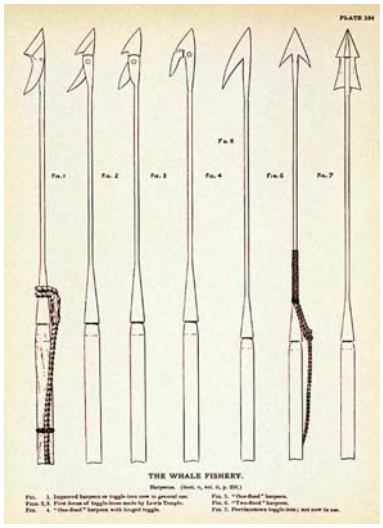
Ex. $\underline{e} = 000$ is more likely to occur than $\underline{e} = 010$

Vector Notation

- $\vec{\mathbf{v}}$: column vector $\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_i \\ \vdots \\ v_n \end{pmatrix}$
- $\underline{\mathbf{r}}$: row vector $(r_1, r_2, \ldots, r_i, \ldots r_n)$
- Subscripts represent element indices inside individual vectors.
  - v_i and r_i refer to the i^{th} elements inside the vectors $\vec{\mathbf{v}}$ and $\underline{\mathbf{r}}$, respectively.
- When we have a list of vectors, we use superscripts in parentheses as indices of vectors.
  - $\vec{\mathbf{v}}^{(1)}, \vec{\mathbf{v}}^{(2)}, \ldots, \vec{\mathbf{v}}^{(M)}$ is a list of M column vectors
  - $\underline{\mathbf{r}}^{(1)}, \underline{\mathbf{r}}^{(2)}, \ldots, \underline{\mathbf{r}}^{(M)}$ is a list of M row vectors
  - $\vec{\mathbf{v}}^{(i)}$ and $\underline{\mathbf{r}}^{(i)}$ refer to the i^{th} vectors in the corresponding lists.

Harpoon

- a long, heavy spear attached to a rope, used for killing large fish or whales



Linear Block Codes

- Definition: $\mathcal{C}$ is a **(binary) linear (block) code** if and only if $\mathcal{C}$ forms a vector (sub)space (over GF(2)).

In case you forgot about the concept of vector space,…

  - Equivalently, this is the same as requiring that

    if $\underline{\mathbf{x}}^{(1)}$ and $\underline{\mathbf{x}}^{(2)} \in \mathcal{C}$, then $\underline{\mathbf{x}}^{(1)} \oplus \underline{\mathbf{x}}^{(2)} \in \mathcal{C}$.

  - Note that any (non-empty) linear code $\mathcal{C}$ must contain $\underline{\mathbf{0}}$.

Take any $\underline{x} \in \mathcal{C}$ $\quad \underline{x} \oplus \underline{x}$ must be $\in \mathcal{C}$

$= \underline{0}$

- Ex. The code that we considered in HW4 is

$$\mathcal{C} = \{00000, 01000, 10001, 11111\}$$

Is it a linear code?

$\underline{x}^{(1)} = 01000 \in \mathcal{C}$

$\underline{x}^{(2)} = 10001 \in \mathcal{C}$

$\oplus$

$\hookrightarrow 11001 \notin \mathcal{C} \Rightarrow \mathcal{C}$ is not linear.

Linear Block Codes: Motivation (1)

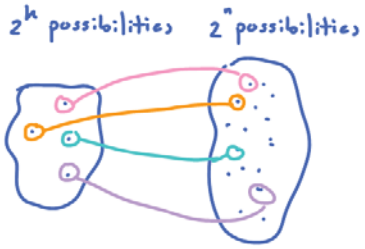
- Why linear block codes are popular?
- Recall: General block encoding
  - Characterized by its codebook.
    - The table that lists all the 2^k mapping from the k-bit info-block $\underline{\mathbf{s}}$ to the n-bit codeword $\underline{\mathbf{x}}$ is called the **codebook**.
    - The M info-blocks are denoted by $\underline{\mathbf{s}}^{(1)}, \underline{\mathbf{s}}^{(2)}, \ldots, \underline{\mathbf{s}}^{(M)}$. The corresponding M codewords are denoted by $\underline{\mathbf{x}}^{(1)}, \underline{\mathbf{x}}^{(2)}, \ldots, \underline{\mathbf{x}}^{(M)}$, respectively.

k bits n bits

index i	info-block $\underline{\mathbf{s}}$	codeword $\underline{\mathbf{x}}$
1	$\underline{\mathbf{s}}^{(1)} = 000\ldots0$	$\underline{\mathbf{x}}^{(1)} =$
2	$\underline{\mathbf{s}}^{(2)} = 000\ldots1$	$\underline{\mathbf{x}}^{(2)} =$
$\vdots$	$\vdots$	$\vdots$
M	$\underline{\mathbf{s}}^{(M)} = 111\ldots1$	$\underline{\mathbf{x}}^{(M)} =$

2^k rows

n positions



  - Can be realized by combinational/combinatorial circuit.
    - If lucky, can used K-map to simplify the circuit.

Linear Block Codes: Motivation (2)

- Why linear block codes are popular?
- Linear block encoding is the same as matrix multiplication.
  - See next slide.
  - The matrix replaces the table for the codebook.
  - The size of the matrix is only $k \times n$ bits.
    - Compare this against the table (codebook) of size $2^k \times (k+n)$ bits for general block encoding.
- Linearity $\Rightarrow$ easier implementation and analysis
- Performance of the class of linear block codes is similar to performance of the general class of block codes.
  - Can limit our study to the subclass of linear block codes without sacrificing system performance.

Linear Block Codes: Generator Matrix

The notation here is modified to be consistent with the one given on p.10.

For any linear code, there is a matrix $\mathbf{G} = \begin{bmatrix} \underline{g}^{(1)} \\ \underline{g}^{(2)} \\ \vdots \\ \underline{g}^{(2)} \end{bmatrix}_{k \times n}$

called the **generator matrix**
such that, for any codeword $\underline{\mathbf{x}}$, there is a message vector $\underline{\mathbf{b}}$ which produces $\underline{\mathbf{x}}$ by

$$\underline{\mathbf{x}} = \underline{\mathbf{b}}\mathbf{G} = \sum_{j=1}^{k} b_j \underline{g}^{(j)}$$

mod-2 summation

Note:

(G1) (1) Any codeword can be expressed as a linear combination of the rows of $\mathbf{G}$

(2) $C = \{\underline{\mathbf{b}}\mathbf{G} : \underline{\mathbf{b}} \in \{0,1\}^k\}$

Remark: Given a matrix G, the (block) code that is constructed by (2) is always linear.

Linear Block Codes: Examples

k=1

- **Repetition code**: $\underline{\mathbf{x}} = [\underbrace{b \quad b \quad \cdots \quad b}_{n \text{ times}}]$
  - $\mathbf{G} = [1 \quad 1 \quad \cdots \quad 1]$
  - $\underline{\mathbf{x}} = \underline{\mathbf{b}}\mathbf{G} = b\mathbf{G} = [b \quad b \quad \cdots \quad b]$
  - $R = \frac{k}{n} = \frac{1}{n}$

Ex. $n = 3$

$\underline{x} = [b \; b \; b]$

$= b[1 \; 1 \; 1]$ ← G

- **Single-parity-check code**: $\underline{\mathbf{x}} = \left[\underline{\mathbf{b}}; \underbrace{\sum_{j=1}^{k} b_j}_{\text{parity bit}}\right]$
  - $\mathbf{G} = [\mathbf{I}_{k \times k}; \underline{\mathbf{1}}^T]$
  - $R = \frac{k}{n} = \frac{k}{k+1}$

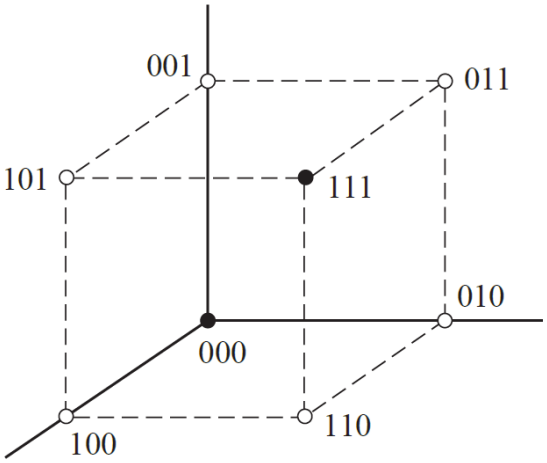
Ex. $k = 2$, $n = 3$

$\underline{b} = [b_1 \; b_2]$

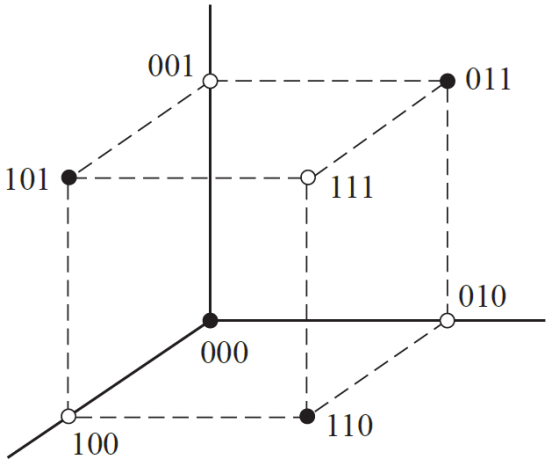
$[b_1 \; b_2] \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} = \underline{x} = [b_1 \quad b_2 \quad b_1 \oplus b_2]$

I

Vectors representing 3-bit codewords



Triple-repetition code



Parity-check code

Related Idea:

Even Parity vs. Odd Parity

- Parity bit checking is used occasionally for transmitting ASCII characters, which have 7 bits, leaving the 8th bit as a **parity bit**.
- Two options:
  - **Even Parity**: Added bit ensures an even number of 1s in each codeword.
    - A: 1000001**0**
  - **Odd Parity**: Added bit ensures an odd number of 1s in each codeword.
    - A: 1000001**1**

Even Parity vs. Odd Parity

- Even parity and odd parity are properties of a codeword (a vector), not a bit.
- Note: The generator matrix $\mathbf{G} = [\mathbf{I}_{k\times k}; \underline{\mathbf{1}}^T]$ previously considered produces even parity codeword

$$\underline{\mathbf{x}} = \left[\boxed{\underline{\mathbf{b}}} ; \sum_{j=1}^{k} b_j \right]$$

The sum of all elements inside $\underline{x}$: $\sum_{j=1}^{k} b_j \oplus \sum_{j=1}^{k} b_j = 0$

- Q: Consider a code that uses odd parity. Is it linear?

Odd parity is not a linear code.

Reason: $\underline{0}$ is not a member.

($\underline{0}$ always has even parity)

Error Control using Parity Bit

- If an odd number of bits (including the parity bit) are transmitted incorrectly, the parity bit will be incorrect, thus indicating that a parity error occurred in the transmission.
- Ex.
  - Suppose we use even parity.
  - Consider the codeword $\underline{\mathbf{x}} = 10000010$ ("A")

one bit error

$\underline{y} = 10000110 \Rightarrow$ odd parity $\Rightarrow$ sth. is wrong $\Rightarrow$ error detected $\Rightarrow$ may request for retransmission

Two bit error

- Suitable for *detecting* errors; *cannot correct* any errors

$\underline{y} = 10000100 \Rightarrow$ still even parity $\Rightarrow$ error undetected $\Rightarrow$ incorrectly interpret it as "B"

Error Detection

- Two types of **error control**:
  1. **error detection**
  2. **error correction**

1 **Error detection**: the determination of whether errors are present in a received word.

- An error pattern is **undetectable** if and only if it causes the received word to be a valid codeword other than that which was transmitted.
  - Ex: In single-parity-check code, error will be undetectable when the number of bits in error is even.

2 Error Correction

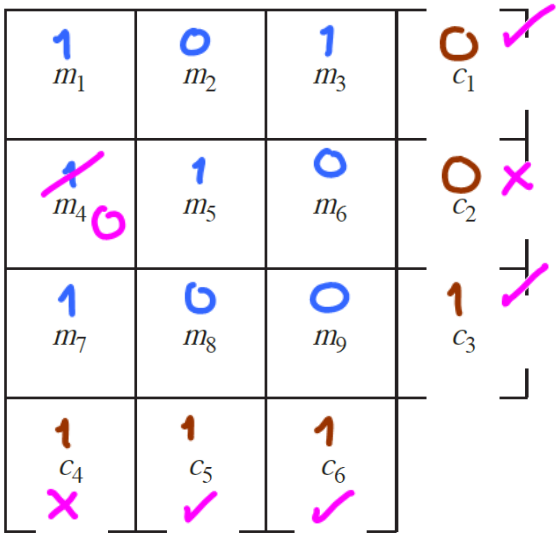
- In **FEC** (**forward error correction**) system, when the decoder detects error, the arithmetic or algebraic **structure** of the code is used to determine which of the valid codewords was transmitted.

- It is possible for a detectable error pattern to cause the decoder to select a codeword other than that which was actually transmitted. The decoder is then said to have committed a **decoding error**.

Square array for error correction by parity checking.

$\underline{m} = [m_1\ m_2\ \cdots\ m_4]$

- The codeword is formed by arranging k message bits in a square array whose rows *and* columns are checked by $2\sqrt{k}$ parity bits.
- A transmission error in one message bit causes a row and column parity failure with the error at the intersection, so single errors can be corrected.



$\underline{x} = [m_1 \cdots m_4\ c_1 \cdots c_6]$

[Carlson & Crilly, p 594]

Weight and Distance

- The **weight** of a codeword $\underline{\mathbf{x}}$ or an error pattern $\underline{\mathbf{e}}$ is the number of nonzero coordinates in the codeword or the error pattern.
  - The weight of a codeword $\underline{\mathbf{x}}$ is commonly written as $w(\underline{\mathbf{x}})$.
  - Ex. $w(010111) = 4$
- The **Hamming distance** between two n-bit blocks is the number of coordinates in which the two blocks differ.
  - Ex. $d(010111, 011011) = 2$

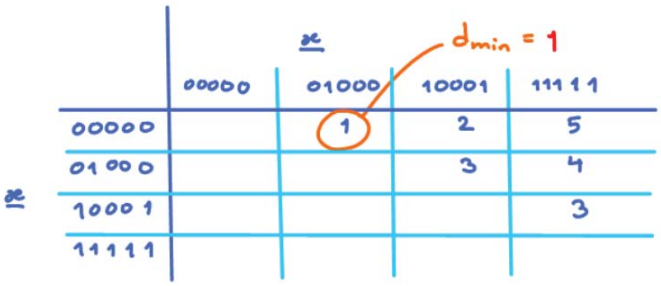
Note: $d(\underline{x}, \underline{y}) = w(\underline{x} \oplus \underline{y}) = w(\underline{e})$

Review: Minimum Distance (d_{min})

The **minimum distance** (d_{min}) of a block code is the minimum Hamming distance between all distinct pairs of codewords.

HW4 **Problem 2.** A channel encoder map blocks of two bits to five-bit (channel) codewords. The four possible codewords are 00000, 01000, 10001, and 11111. A codeword is transmitted over the BSC with crossover probability $p = 0.1$.

(a) What is the minimum (Hamming) distance d_{min} among the codewords?



d_{min}: two important facts

- For any linear block code, the **minimum distance** (d_{min}) can be found from minimum weight of its nonzero codewords.
  - So, instead of checking $\binom{2^k}{2}$ pairs, simply check the weight of the 2^k codewords.

- A code with minimum distance d_{min} can
  - detect all error patterns of weight w $\leq d_{min}-1$.
  - correct all error patterns of weight w $\leq \left\lfloor \frac{d_{min}-1}{2} \right\rfloor$.

the floor function