Unit 9

Codes

Outline of Unit 9

- 9.1 Parity-Check Codes
- 9.2 Generator and Parity Check Matrices
- 9.3 Hamming Codes

Example: Error Detection

Is this a valid HKID card number?

Weighted Average Mod 11

A=10, B=11, …, Z=35, Space = 36

HKID:		C	6	6	8	6	6	8	(?)
Weight:	9	8	7	6	5	4	3	2	1

$$36 \times 9 + 12 \times 8 + 6 \times 7 + 6 \times 6 + 8 \times 5 + 6 \times 4 + 6 \times 3 + 8 \times 2 + x \equiv 0 \pmod{11}$$

$$5 + 8 + 9 + 3 + 7 + 2 + 7 + 5 + x \equiv 0 \pmod{11}$$

$$x \equiv -2 \equiv 9 \pmod{11}$$

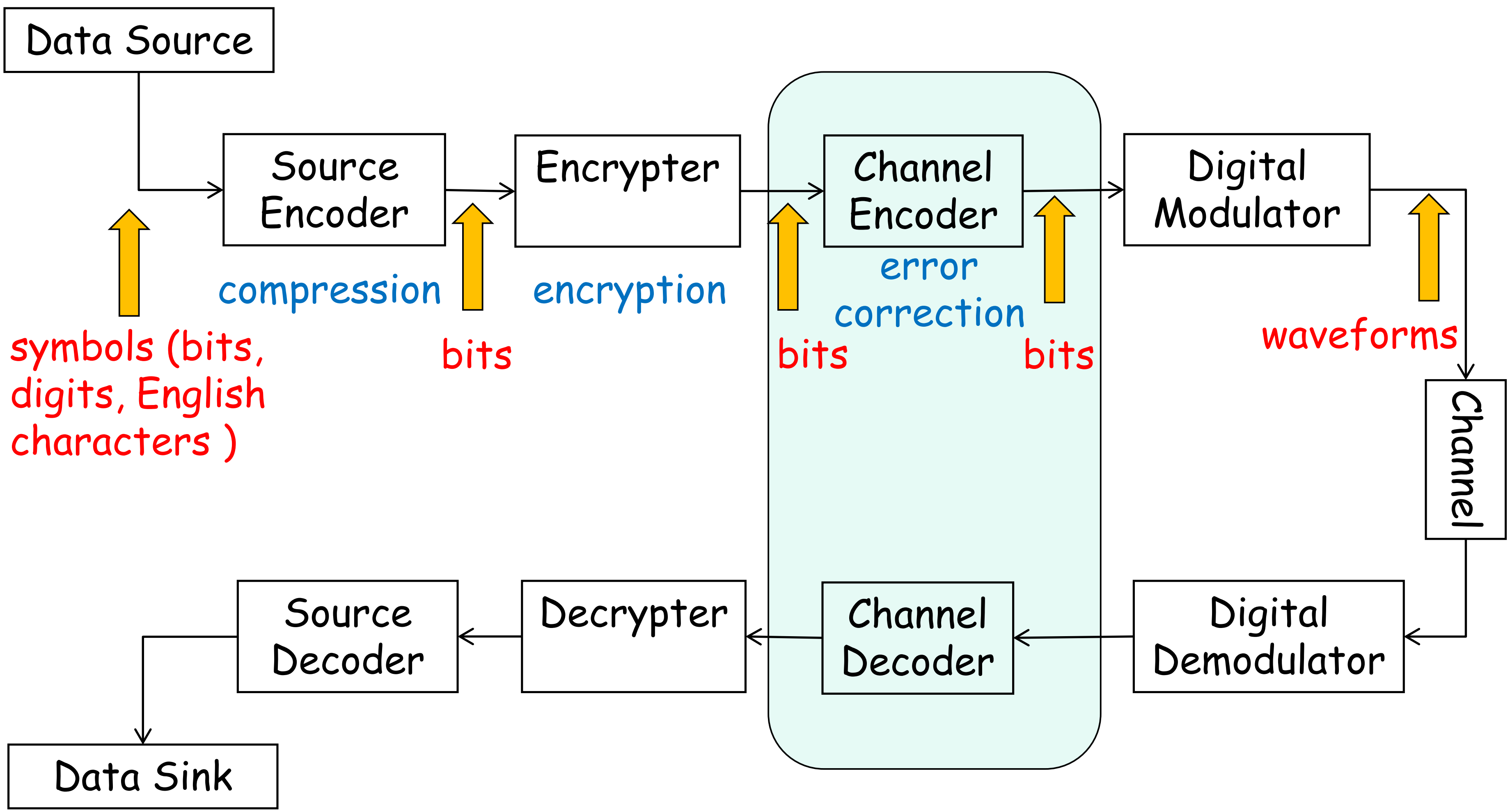
Example: Error Correction

Does it still work?

Unit 9.1

Parity-Check Codes

Digital Communication Systems



Bit Errors due to Noise

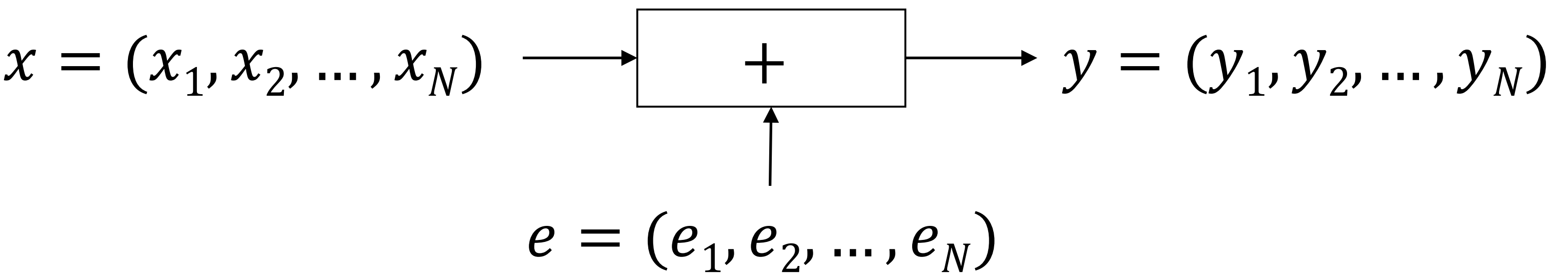
- Suppose N bits are transmitted.

$$x = (x_1, x_2, \dots, x_N) \Rightarrow \boxed{\text{Channel}} \Rightarrow y = (y_1, y_2, \dots, y_N)$$

- During the transmissions, bit errors may occur due to noise.
- The probability that a bit error occurs is called the *bit error rate*.

	Twisted Pair	Coaxial Cable	Optical Fiber
Data Rate in Mbps	10	100	1000
Bit Error Rate	10^{-5}	10^{-6}	10^{-9}
Bandwidth	250 kHz	350 MHz	1 GHz

Error Vector



- If the i-th bit is in error,

  then $e_i = 1$; else $e_i = 0$.

- Hence, for all i,

$$y_i = x_i + e_i,$$

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	0

where binary addition (+) means logical XOR.

How to Handle Bit Errors?

Two basic strategies:

1. Error Correction
   - Include enough redundant information along with a data packet to enable the receiver to identify which bits are in error and then *correct* them.
2. Error Detection
   - Include only enough redundant information to allow the receiver to detect that an error occurred.
   - If error is detected, the receiver may request for *retransmissions*.

Single Parity Check

1001010

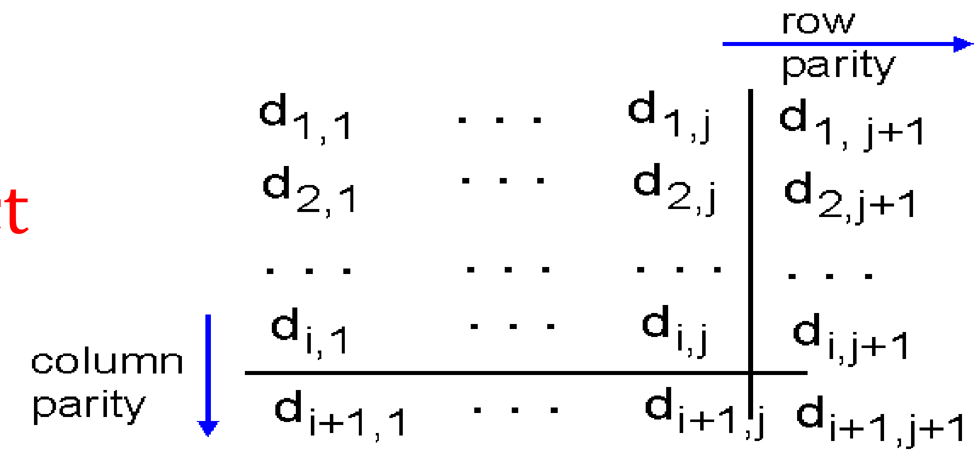
10010101 → Error-prone communication link → 10000101

Oops, something went wrong!

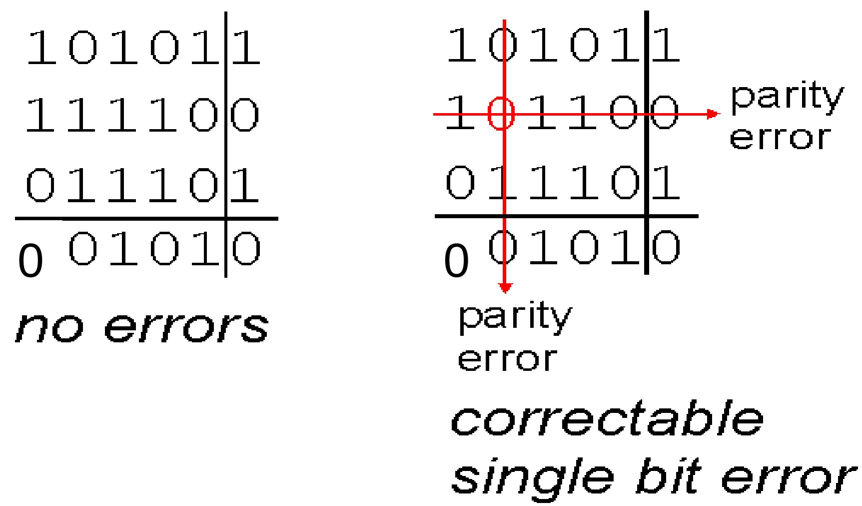
- Suppose there are *k* information bits.
- An extra bit (called parity bit) is added for detecting single-bit errors.
  - *Even parity*: add an extra bit so that the total number of '1's is even.
    - Odd parity can be defined similarly.
  - The receiver doesn't know which bit is in error.

Two-Dimensional Parity Check

- This scheme can detect and correct single-bit errors.



- Even parity is assumed in this example.



Parity-Check Codes

message u (a row vector)

k
information bits

Systematic encoding $\longrightarrow$

codeword c (a row vector)

k information bits	$r = n - k$ check bits

- (n, k) binary code with the following notation:
  - k information bits
  - r redundant bits
  - Codeword length:
    $$n = k + r$$
  - The code is a set of 2^k codewords.
- The encoding of a code is systematic if the information bits are embedded as part of the encoded output.
  - The check bits are *not* necessarily after the information bits.

Examples

(4,3) Even Parity

- 8 codewords

```
0 0 0 0
0 0 1 1
0 1 0 1
0 1 1 0
1 0 0 1
1 0 1 0
1 1 0 0
1 1 1 1
```

Parity bits

(5,1) Repetition

- 2 codewords

```
0 0 0 0 0
1 1 1 1 1
```

Parity bits

Code Rate

- The code rate of an (n, k) code is defined as
$$R_c = \frac{k}{n}.$$
  - Proportion of the data stream that is useful.
- If the raw data rate of a link is W bps, then the effective data rate (or information rate) is $R_c W$ bps.
- Example:
  - Suppose every 7 bits of data are encoded with single parity check.
  - The encoded output is then transmitted through a link with 1 Mbps.
  - What is the effective data rate?

Error Detection Failure

- (4, 3) Even Parity Check Code:

	0 0 0 0				Bits in error!	
x	**0 0 1 1**	→	Channel	→	**0 1 0 1**	y
	0 1 0 1					
	0 1 1 0					
	1 0 0 1					
	1 0 1 0					
	1 1 0 0					
	1 1 1 1					

- The decoder will fail to detect these bits errors. Why?
- Reason: There is an even number of “1”s.
- Deeper reason: If $y \neq x$ and y is a codeword, then the decoder will mistake that no bit errors have occurred.

Hamming Distance

- Hamming distance is a useful concept for analyzing the error correction/detection capability of a code.
- The Hamming distance $d(x, y)$ of two vectors, x and y, is defined as the number of bits that they are different.
- The Hamming weight $w(x)$ of a vector x is defined as the number of 1's in x.
- Example:
  - $x = (0, 0, 1, 1, 1), \quad w(x) = 3.$
  - $y = (0, 1, 1, 0, 1), \quad w(y) = 3.$
  - $d(x, y) = 2.$

Error Detection Failure

- If the Hamming distance between two codewords is d, then it will require d errors to convert one codeword into the other.
- Example:
  - Codeword 1: `1 0 0 0 1 0 0 1`
  - Codeword 2: `1 0 1 1 0 0 0 1`

  Hamming distance = 3

  - An *error detection failure* occurs if the above three bits are in error.

Error Detection Capability

Definition: The minimum distance, $d_{\min}$, of a code is the *smallest* Hamming distance between *all pairs* of distinct codewords in the code.

- Error detection capability of a code depends on its $d_{\min}$.
- It is guaranteed that error can be detected if number of bits in error is less than or equal to

$$s = d_{\min} - 1.$$

Examples (revisited)

(4,3) Even Parity

- Eight codewords:

```
0 0 0 0
0 0 1 1
0 1 0 1
0 1 1 0
1 0 0 1
1 0 1 0
1 1 0 0
1 1 1 1
```

(5,1) Repetition

- Two codewords:

```
0 0 0 0 0
1 1 1 1 1
```

What is $d_{\min}$ of each of these codes?

How many bit errors does each code guarantee to detect?

Decoding Rule for Error Correction

$x = (x_1, x_2, \dots, x_N) \Rightarrow \boxed{\text{Channel}} \Rightarrow y = (y_1, y_2, \dots, y_N)$

- *Nearest-Neighbor Decoding*: The decoder picks a codeword that is closest to y in terms of Hamming distance.
  - In other words, find $x \in C$ which minimizes $d(x, y)$, where C is the set of all codewords.
    - Tie is broken arbitrarily.
    - a.k.a minimum-distance decoding
- Example: (5, 1) Repetition Code
  - How many bit errors can the code correct?

Error Correction Capability

- Error correction capability of a code depends on its $d_{\min}$.
- Error can be corrected if no. of bits in error is less than or equal to

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor,$$

  - where $\lfloor x \rfloor$ is the floor operator which denotes the largest integer less than or equal to x.
- Example: $(6, 1)$ Repetition Code

$$d_{\min} = 6, \qquad t = \lfloor 2.5 \rfloor = 2.$$

Code Rate of (5, 1) Repetition

- $C = \{00000,\ 11111\}$.
- $d_{\min} = 5,\ t = 2.$ (correct all double-bit errors)
- Code rate $R = \frac{k}{n} = \frac{1}{5}$.
  - For each information bit, we need to transmit 5 bits.
  - For example, if transmission rate equals 1 Mbps, then we can transmit 200 kbps of useful information.
- What if we want to convey information faster?

Classwork (Repetition with an Extra Parity)

$u = (u_1, u_2)$ ➡ Encoder ➡ $c = (c_1, c_2, c_3, c_4, c_5)$

- $c_1 = c_3 = u_1$
- $c_2 = c_4 = u_2$
- $c_5 = u_1 + u_2$

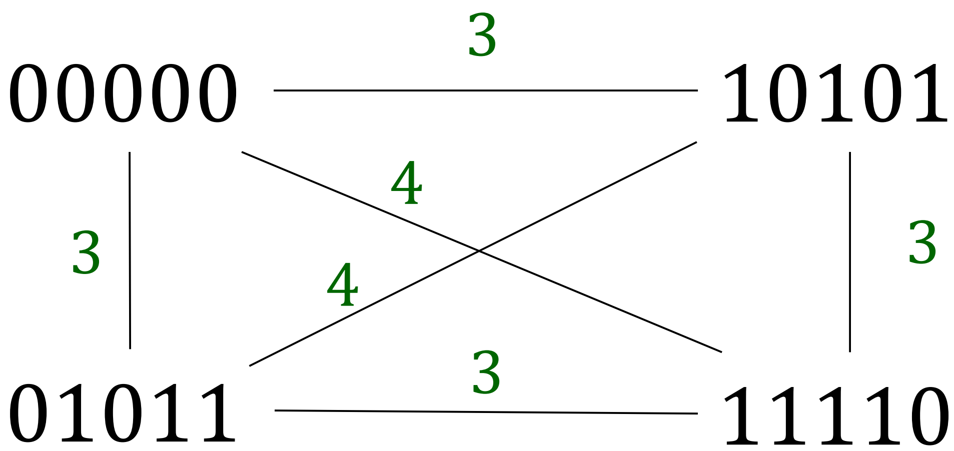
Message u	Codeword c
00	
01	
10	
11	

a) Complete the table.

b) Determine $d_{\min}$.

c) How many errors can it correct?

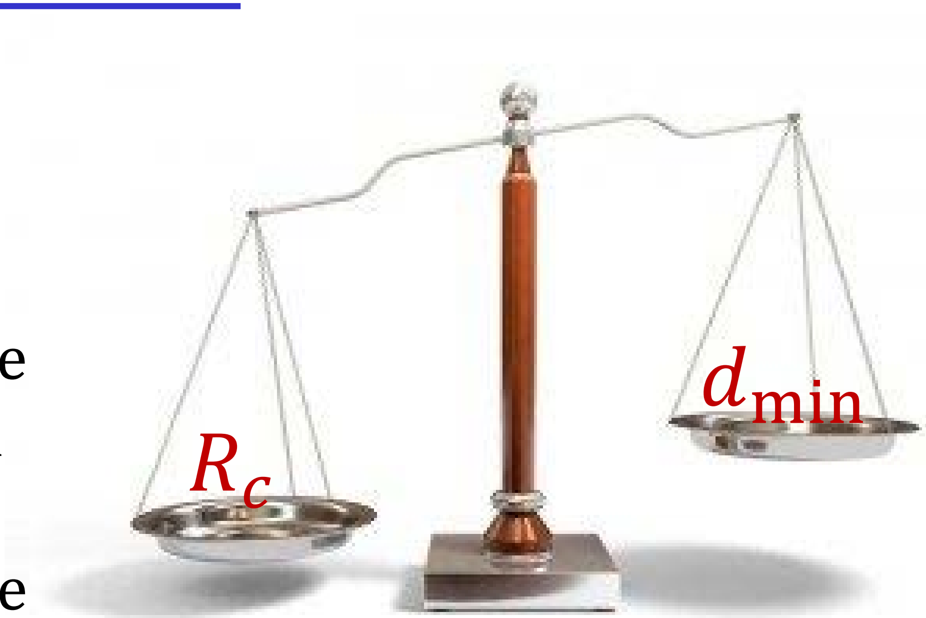
Solution



- $R = \frac{2}{5}, d_{\min} = 3,\ t = 1.$
  - For example, if transmission rate equals 1 Mbps, then we can transmit 400 kbps of useful information.
- Comparison with (5, 1) repetition code:
  - More efficient in communications (less redundancy)
  - Weaker error correction capability

Performance Measures

- **Code rate**
  - The higher the value of R_c, the more efficient the coding scheme is, which means the higher the effective data rate can be achieved.
- **Minimum distance**
  - The larger the value of $d_{\min}$, the higher the error detection/correction capability.



Unit 9.2

Generator and Parity-Check Matrices

Binary Linear Codes

- A *linear* code is defined by the *generator matrix*, G.
  - $k \times n$ matrix
  - Each entry is 0 or 1.
- Encoding is done by :

$$c = uG.$$

$$u \longrightarrow \boxed{\text{Encoder}} \longrightarrow c$$

(u and c are *row* vectors.)

- For systematic encoding,

$$G = [\, I_k \mid P \,].$$

  - I_k is the $k \times k$ identity matrix.
  - G is said to be in standard form.
  - In general, $\boldsymbol{G}$ need *not* contain the identity matrix, and the corresponding code is non-systematic.

Example: (5, 4) Even Parity

- Message $u = [u_1 \quad u_2 \quad u_3 \quad u_4]$.
- Add a parity c_5 so that there is an *even number* of 1's in every codeword.
- In matrix form, $c = uG$, where

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

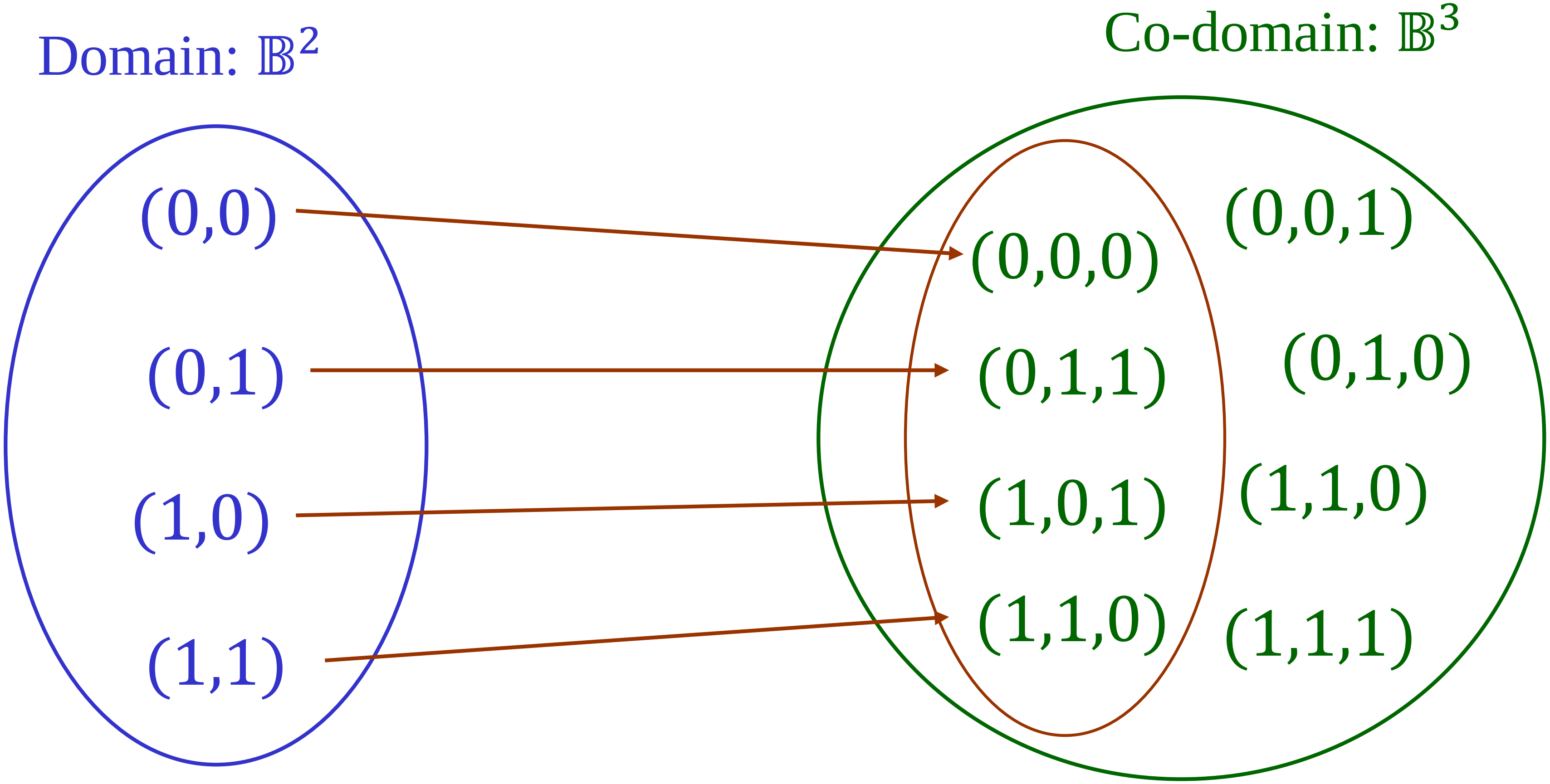
(G is the generator matrix.)

$$c_1 = u_1$$
$$c_2 = u_2$$
$$c_3 = u_3$$
$$c_4 = u_4$$
$$c_5 = u_1 + u_2 + u_3 + u_4$$

Encoding: An Injective, Linear Mapping

- Encoding of a linear code is a linear function:
  $$f: \mathbb{B}^k \to \mathbb{B}^n,$$
  where
  - $f(u) = uG$;
  - $\mathbb{B}^m$ is the set of all binary m-vectors.
- The mapping should be *injective*.
  - That means, no two inputs map to the same output.
    - Remark: G needs to be of full row rank, (i.e., the rows are linearly independent).
- Example:
  - $G = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$.
  - Consider two messages
    - $u_1 = [1 \quad 1 \quad 0]$
    - $u_2 = [0 \quad 0 \quad 1]$
  - Both of them map to the same codeword:
    - $u_1G = [1 \quad 0 \quad 1 \quad 1]$
    - $u_2G = [1 \quad 0 \quad 1 \quad 1]$
  - Ambiguity arises in decoding.

(3,2) Even Parity



The range (i.e., the code) is a vector subspace of $\mathbb{B}^3$.

Linear Code is a Subspace of $\mathbb{B}^n$

- (Closed under vector addition)
  - For any binary linear code, the *sum* (i.e., XOR) of any two codewords is a *codeword*.
    - Consider two codewords, c_u and c_v.
    - $c_u + c_v = uG + vG = (u + v)G$, which is a codeword.

- (Closed under scalar multiplication)
  - Any codeword c multiplied by a *scalar* (i.e., 0 or 1) is also a codeword.
    - $c \times 1 = c$ (a codeword)
    - $c \times 0 = \mathbf{0}$ (a codeword due to zero-in zero-out)
  - Note that binary multiplication is the same as logical AND.

Example: (5,4) Even Parity (cont'd)

- Clearly, any codeword satisfies

$$c_1 + c_2 + c_3 + c_4 + c_5 = 0.$$

- This is called the parity-check equation.

- Represented in matrix form,

$$c\,H^T = 0,$$

This equation can be used to check whether a given vector is a codeword or not.

where

$$H = [1\ 1\ 1\ 1\ 1]$$

is called the *parity-check matrix*.

Parity-Check Matrices

- Let G be a generator matrix of an (n, k) code.
  - It is a $k \times n$ matrix.
- A parity-check matrix H is an $r \times n$ matrix satisfying
  $$G\ H^T = 0.$$
  - It is not unique.
  - Recall $r = n - k$ is the number of redundant bits.
- For a systematic code, the generator matrix is of the form
  $$G = [\ I_k \mid A\ ].$$
- A parity-check matrix is given by
  $$H = [A^T \mid I_r\ ].$$
  - Caution: This applies only to binary codes.

Re-visit (Repetition with an Extra Parity)

- $c_1 = c_3 = u_1,\ c_2 = c_4 = u_2,\ c_5 = u_1 + u_2$
- This is a systematic matrix, which can be expressed as

$$c = uG = [u_1 \quad u_2]\begin{bmatrix} \boxed{\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}} & \begin{matrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{matrix} \end{bmatrix}.$$

$\boxed{G \text{ is } k \times n}$

- According to the previous slide, the parity-check matrix can be expressed as

$$H = \begin{bmatrix} \begin{matrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{matrix} & \boxed{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}} \end{bmatrix}.$$

$\boxed{H \text{ is } r \times n}$

- All codewords satisfy the parity-check equation:

$$cH^T = 0.$$

Re-visit (Repetition with an Extra Parity)

- $cH^T = [c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5] \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = 0.$
- Parity check equations:

1. $c_1 + c_3 = 0$ (repetition)
2. $c_2 + c_4 = 0$ (repetition)
3. $c_1 + c_2 + c_5 = 0.$ (c_5 is parity)

Error Detection by Checking r Parities

$$x = (x_1, x_2, \dots, x_N) \Rightarrow \boxed{\text{Channel}} \Rightarrow y = (y_1, y_2, \dots, y_N)$$

- The receiver computes $s = yH^T$ to check the parities.
  - s is an r-vector, which corresponds to r parity-check equations.
  - If $s_i \neq 0$, then the i-th parity-check equation does not hold.
- s is called the syndrome.

$$s \begin{cases} = 0 & \text{no error is detected.} \\ \neq 0 & \text{error is detected.} \end{cases}$$

Re-visit (Repetition with an Extra Parity)

- Suppose $y = (0, 1, 0, 0, 1)$ is received.
- Compute the syndrome:

$$s = yH^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}.$$

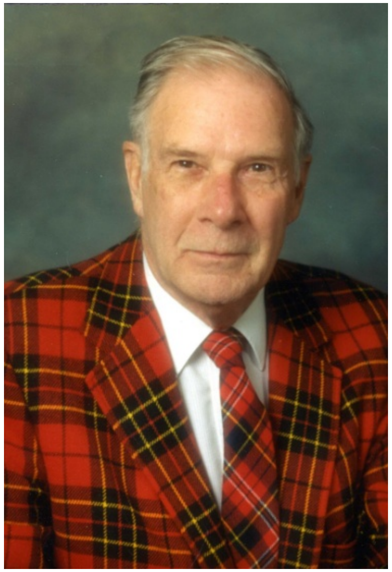
- Since the syndrome is non-zero, error is detected.
- Only the second parity-check equation does not hold, i.e.,
  1. $c_1 + c_3 = 0$
  2. $c_2 + c_4 \neq 0$ (i.e., either c_2 or c_4 is in error)
  3. $c_1 + c_2 + c_5 = 0$.
- If one bit is in error, then it must be c_4.
  - Otherwise, if c_2 is in error, the third equation cannot hold.

Unit 9.3

Hamming Codes

Hamming Codes

- Goal: *To correct single-bit errors*.
- A Hamming code has $r \geq 2$ parity bits, which yields $d_{min} = 3$.
  - Either detect up to two-bit errors or correct one-bit errors (but not both).
- Inventor:
  - Worked at Bell Labs in 1940s.
  - Frustrated with the error-prone punched card reader and invented the famous (7,4) Hamming code in 1950.

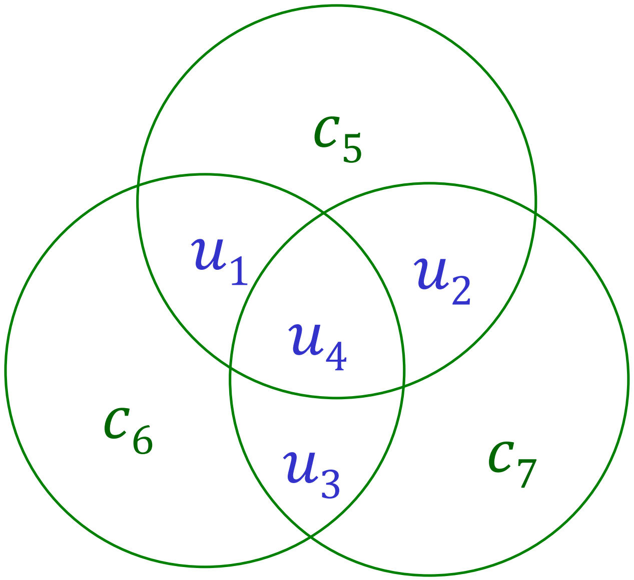


Richard Wesley Hamming (1915-1998). He won the Turing Award in 1968.

Example: (7, 4) Hamming Code

- The encoding equations are

$$
\begin{aligned}
c_1 &= u_1 \\
c_2 &= u_2 \\
c_3 &= u_3 \\
c_4 &= u_4 \\
c_5 &= u_1 + u_2 + u_4 \\
c_6 &= u_1 + u_3 + u_4 \\
c_7 &= u_2 + u_3 + u_4
\end{aligned}
$$



- Parity-check equations:

$$
\begin{aligned}
c_1 + c_2 + c_4 + c_5 &= 0 \\
c_1 + c_3 + c_4 + c_6 &= 0 \\
c_2 + c_3 + c_4 + c_7 &= 0
\end{aligned}
$$

$$
G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}_{k \times n}
\qquad
H = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}_{r \times n}
$$

Codewords of (7,4) Hamming code

- It can be checked that $d_{\min} = 3$.

Either detect all single-bit errors and double-bit errors, **or** correct all single-bit errors, **but not** both.

c_1	c_2	c_3	c_4	c_5	c_6	c_7
0	0	0	0	0	0	0
0	0	0	1	1	1	1
0	0	1	0	0	1	1
0	0	1	1	1	0	0
0	1	0	0	1	0	1
0	1	0	1	0	0	1
0	1	1	0	1	1	0
0	1	1	1	0	0	1
1	0	0	0	1	1	0
1	0	0	1	0	0	1
1	0	1	0	1	0	1
1	0	1	1	0	1	0
1	1	0	0	0	1	1
1	1	0	1	1	0	1
1	1	1	0	0	0	0
1	1	1	1	1	1	1

Fast Method to Determine $d_{\min}$

Theorem: Given any linear code,

$d_{\min}$ = min. weight of all non-zero codewords

Proof: Pick two distinct codewords x and y.

$d(x, y) = w(x + y)$ ($x_i + y_i = 1$ if the two bits are different)

$= w(z)$ for some $z \in C$. (property of linear code)

Note that z is non-zero since $x \neq y$. *Q.E.D.*

- Verify it using the (7,4) Hamming code in the previous slide!

Example: (7, 4) Hamming Code

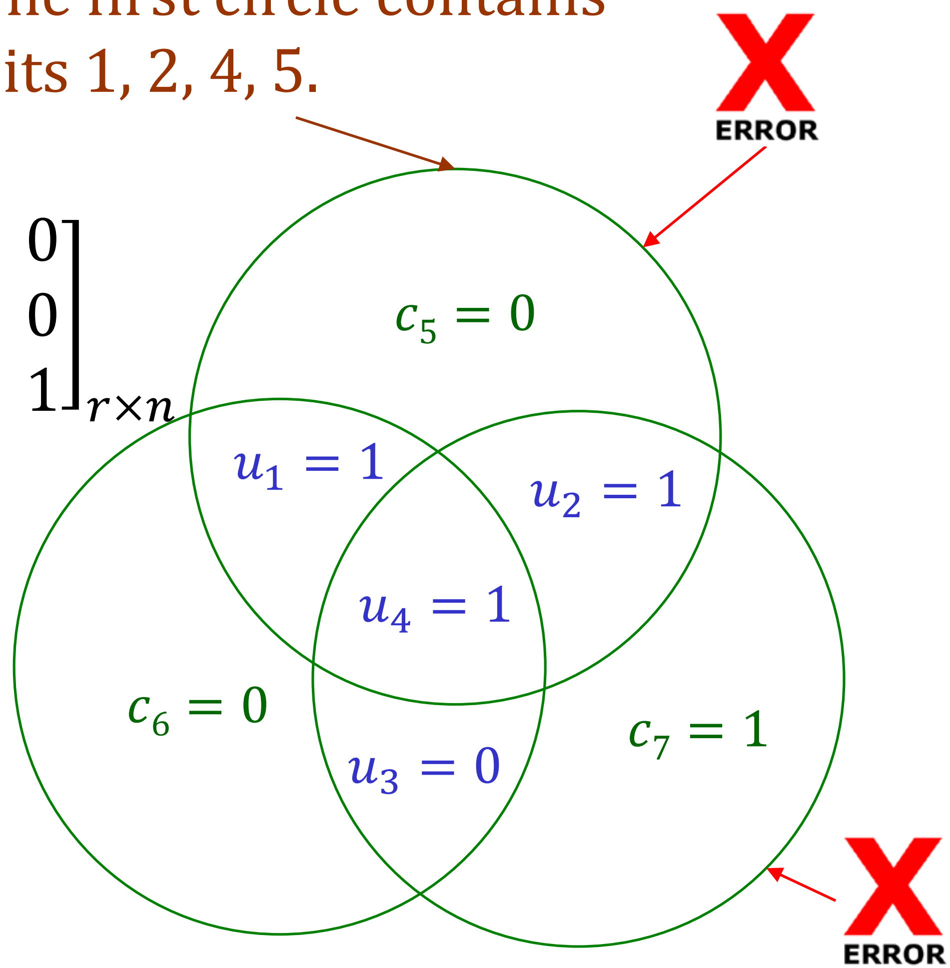
- $y = (1, 1, 0, 1, 0, 0, 1)$

The first circle contains bits 1, 2, 4, 5.

- $H = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}_{r \times n}$

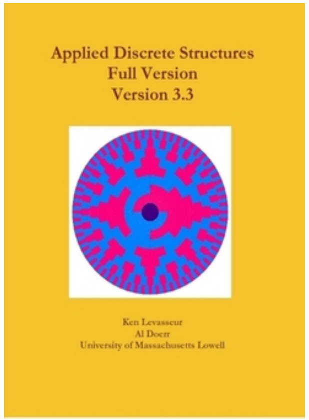
- $s = yH^T = [1 \quad 0 \quad 1]$

1st and 3rd circles are in error.

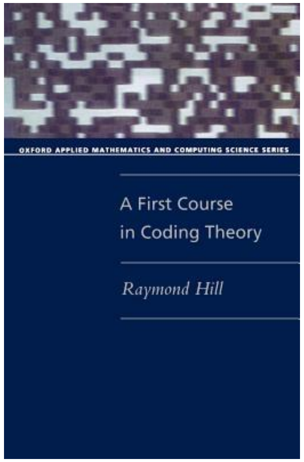


If one bit is in error, which one is it?

Recommended Reading



- Section 15.5, K. Levasseur and A. Doerr, *Applied Discrete Structures*, lulu.com, 2017.
  - Available online: http://faculty.uml.edu/klevasseur/ads/



- Chapters 5-7, R. Hill, *A First Course in Coding Theory*, Oxford 1986.