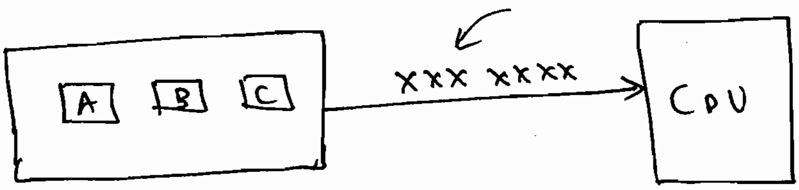
☆ Binary Codes:

7bit ASCII code in serial fashion.



zero → '0' ⇒ $30_H = \cancel{0}\boxed{0110000_2}$

'A' ⇒ $41_H = \cancel{0}\boxed{1000001_2}$

'a' ⇒ $61_H = \cancel{0}\boxed{1100001_2}$

(1) Alphanumberic Codes.

(2) Numeric Codes.

→

1) Alphanumeric codes →
- → ASCII Code (7 bits; $2^7 = 128$ Alphanumericals)
- → EBCDIC Code (8 bit = $2^8 = 256$ Alphanumericals)

→ used in IBM Computers.

2) Numeric Codes:

→ BCD (Binary coded Decimals) codes.

- Weighted
  - Positively
    - → 8 4 2 1 code (Natural BCD code)
    - → 5 4 2 1
    - → 3 3 2 1
    - → 2 4 2 1
  - negatively
    - → 8 4 −2 −1 code.
    - → 6 3 1 −1 Code.
- Non-weighted codes
  - Gray code (Unit distance code)
  - Excess −3

(Self Complementry Codes: sum of weights = 9)

Decimal Digit	8 4 2 1	Excess - 3
0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 1 0 0
2	0 0 1 0	0 1 0 1
3	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 1 1
5	0 1 0 1	1 0 0 0
6	0 1 1 0	1 0 0 1
7	0 1 1 1	1 0 1 0
8	1 0 0 0	1 0 1 1
9	1 0 0 1	1 1 0 0
Invalid BCD { 10	1 0 1 0	
:	: : : :	
16	1 0 1 1	

(1) Self Complementary.

	8421	Ex-3	
3_{10} =	0011	0110	→ 3_{10}
9's Comp	~~1's Comp~~		1's Comp.
6_{10} =	~~0000~~ 0110	1001	→ 6_{10}

(2) Sequential Codes:

	8421	Ex-3	
3_{10} →	0011	0110	← 3_{10}
	+ 1	+ 1	
4_{10} →	0100	0111	← 4_{10}
	+ 1	+ 1	
5_{10} →	0101	1000	← 5_{10}.

✓ ✓

→ The advantage of Ex-3 code is it is both sequential and self complementary.

→ During BCD addition output is invalid

(1) If the result is greater than 9

(2) If carry occurs during BCD addition.

E.g.

$$
\begin{array}{rcr}
 & & \text{111①} \rightarrow \text{carry} = 1 \rightarrow \text{Invalid} \\
68_{10} & \rightarrow & 0110\ 1000 \\
+58_{10} & & +\ 0101\ 1000 \\
\hline
126_{10} & & 1100\ 0000 \\
 & & +\ 0110\ 0110 \\
\hline
 & & 1\ 0010\ 0110
\end{array}
$$

>9 → invalid

Ans → 126_{10}

(2) Find the no. of BCD correction = ?

(a)

$$
\begin{array}{rcr}
174_{10} & \rightarrow & 0001\ 0111\ 0100 \\
+\ 826_{10} & & +\ 1000\ 0010\ 0110 \\
\hline
1000 & & 1001\ 1001\ 1010 \rightarrow >9 \times \\
 & & +\ 0110 \\
\hline
 & & 1001\ 1010\ 0000 \\
 & & +\ 0110\ 0000 \\
\hline
 & & 1010\ 0000\ 0000 \\
 & & +\ 0110\ 0000\ 0000 \\
\hline
 & & 1\ 0000\ 0000\ 0000
\end{array}
$$

So, 3 correction

1 0 0 0

$= 1000_{10}$

* Excess-3 Addition

① If carry occur → Add 3_{10}

② If carry doesn't occur → Subtract 3_{10}.

e.g.

$$
\begin{array}{rcll}
 & & \scriptstyle 1\,1\,1\,1\,1 & \scriptstyle 1 \\
38_{10} & \rightarrow & 0110 & 1011 \\
+\,46_{10} & \rightarrow & 0111 & 1001 \;\; \cancel{0110} \\
\hline
 & & 1110 & 0100 \\
 & & -\,0011 & +\,0011 \\
\hline
 & & 1011 & 0111 \\
 & & \downarrow & \downarrow \\
 & & 11-3 & 7-3 \\
 & & \downarrow & \downarrow \\
 & & 8 & 4
\end{array}
$$

$\Rightarrow 84_{10}$

* Grey Code (Reflected code) (unit distance code) (cyclic code).

1-bit Gray code $\xrightarrow{\text{Reflective property}}$ 2-bit gray code $\xrightarrow{RP}$ 3-bit gray code

1-bit	2-bit	3-bit	
0	00	000	(0)
1	01	001	(1)
	11	011	(2)
	10	010	(3)
		110	(4)
		111	(5)
		101	(6)
		100	(7)

dist. unit = 1

dist=1 (Cyclic code)

→ Use of gray code:

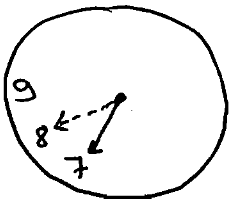
① K-map

② Shaft encoders

③ Error correction and detection

④ Genetic algorithm.

→ Shaft encoders



	8421
7 →	0111
8 →	1000

More chance to get error (or) all 4 bit should change

Gray code

0100
↓
1100

Less chance to get error as it is unit distance code.

Code Conversion

① Binary to Gray

$B_4 \oplus B_3 \oplus B_2 \oplus B_1$

Binary:	1	1	0	1
↓				
Gray:	1	0	1	1
	G_4	G_3	G_2	G_1

$G_4 = B_4$

$G_3 = B_4 \oplus B_3$

$G_2 = B_3 \oplus B_2$

$G_1 = B_2 \oplus B_1$

→ EX-OR → Modulo - 2 Addition

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 0.$

(2) Gray to Binary.

	G_4	G_3	G_2	G_1
Gray :	1	0	1	1
Binary :	1	1	0	1
	B_4	B_3	B_2	B_1

$B_4 = G_4$

$B_3 = B_4 \oplus G_3$

$B_2 = B_3 \oplus G_2$

$B_1 = B_2 \oplus G_1$

Ex-1 Represent $(743)_8$ in Gray code.

Ans: $(743)_8$

$\downarrow$

$(\underbrace{111}_{7}\,\underbrace{100}_{4}\,\underbrace{011}_{3})_2$

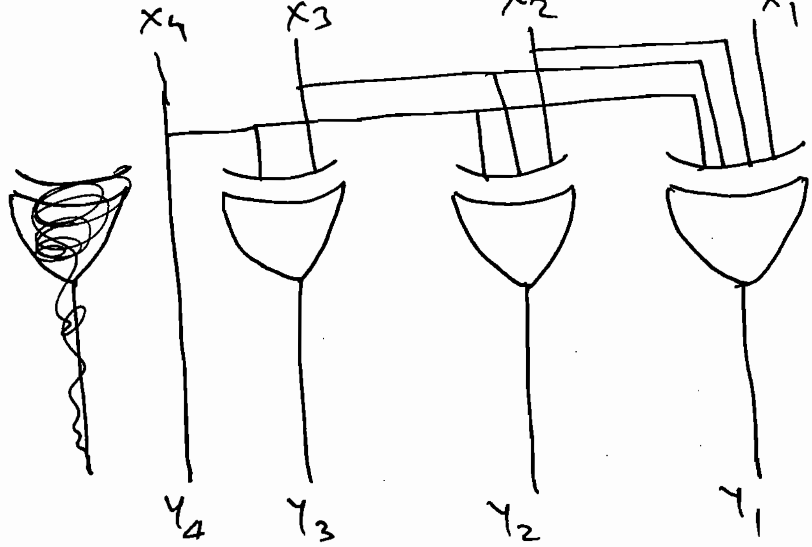
Now B : 1 1 1 1 0 0 0 1 1

$\downarrow$

G : $(100010010)_2$

$\therefore (743)_8 = 100010010 \leftarrow$ Gray code.

Ex-2 Identify the following code converter.



$Y_4 = X_4$

$Y_3 = X_4 \oplus X_3$

$Y_2 = X_4 \oplus X_3 \oplus X_2$

$Y_3 = X_4 \oplus X_3 \oplus X_2 \oplus X_1.$

& $G \rightarrow B$

Ex-3 $b_4 b_3 b_2 b_1$ is a 4-Bit binary no. What is the f^n of the following eq^{n}s.

(1) $b_{11}^* = b_1$.

$b_{22}^* = b_1 \oplus b_2$.

$b_{33}^* = (b_1 + b_2) \oplus b_3$.

$b_{44}^* = (b_1 + b_2 + b_3) \oplus b_4$.

b_1	b_2	b_3	b_4
0	1	0	1
0	1	1	0
~~0~~	~~1~~	~~1~~	~~0~~
~~0~~	~~0~~	~~0~~	~~1~~
1	0	1	0
0	1	1	0

So, Ans is 2's complement.

$(b_1+b_2+b) \oplus$ $\oplus (b_1+b_2)$

b_4	b_3	b_2	b_1
↓	↓	↓	↓
b_{44}	b_{33}	b_{22}	b_{11}

* Hamming Code (Single Error Correcting Code):

→ $\boxed{2^k \geq m + k + 1}$

m = no. of message bits.

k = no. of parity bits.

E.g. $\boxed{m = 4}$ ⟹ No. of Parity bits $\boxed{k = 3}$.

Let, m_1, m_2, m_3, m_4 Let, P_1, P_2, P_3.

001	010	011	100	101	110	111
$2^0 = 1$	$2^1 = 2$	3	$2^2 = 4$	5	6	7
P_1	P_2	m_1	P_3	m_2	m_3	m_4

✓ Choose 'P_1' such that $1, 3, 5, 7 = P_1, m_1, m_2, m_4$ has odd parity

✓ Choose 'P_2' such that $2, 3, 6, 7 = P_2, m_1, m_3, m_4$ " " "

o Choose 'P_3' such that $4, 5, 6, 7 = P_3, m_2, m_3, m_4$ has odd parity.

Ex 1 (7,4) Hamming Code with odd parity for the message 1001.

(a)

→ 7 = total no. of bits.

4 = no. of message bits.

1	2	3	4	5	6	7
P_1	P_2	1	P_3	0	0	1

Choose P_1 = P_1, 1, 0, 1 should have odd parity ~~⇒ cho~~
1, 3, 5, 7
⇒ Choose $\boxed{P=1}$

Choose P_2 2, 3, 6, 7 = P_2 1 0 1 " " " "
$\boxed{P_2 = 1}$

Choose P_3 4, 5, 6, 7 = P_3 0 0 1 " " "
$\boxed{P_3 = 0}$.

Corrected code: 1 1 1 0 0 0 1

(b) (7,4) code hamming code is received as

1 1 1 0 1 0 1

Ans: 4, 5, 6, 7 ⇒ 0 1 0 1 ⇒ even parity ⇒ $C_3 = 1$
(in error)

2, 3, 6, 7 ⇒ 1, 1, 0, 1 ⇒ odd parity ⇒ $C_2 = 0$

1, 3, 5, 7 ⇒ 1, 1, 1, 1 ⇒ even parity ⇒ $C_1 = 1$.
(in error)

i.e. error occured at $C_3 C_2 C_1$ = 101 = 5th position.

Received code = 1 1 1 0 1 0 1 transmitted code

Corrected code = 1 1 1 0 0 0 1 = 1001.

⇒ For Hamming Distance

a) For correcting
't' errors: $\boxed{\text{Hamming distance} \geq 2t+1.}$

b) For detecting
't' errors: $\boxed{\text{Hamming distance} \geq t+1.}$

☆ Boolean Algebra:

⇒ 'AND' Law | 'OR' Law

Identity element ⇒ '1' | '0'

$A \cdot 0 = 0$

$A \cdot \boxed{1} = A$

$A + \boxed{0} = A$

$A + 1 = 1.$

ⓔ Commutative Law:

Dual

$A + B = B + A.$

$A \cdot B = B \cdot A.$

NAND : $\uparrow$

$A \uparrow B = B \uparrow A$

$\therefore \overline{A \cdot B} = \overline{B \cdot A}$

Dual

$A \downarrow B = B \downarrow A$