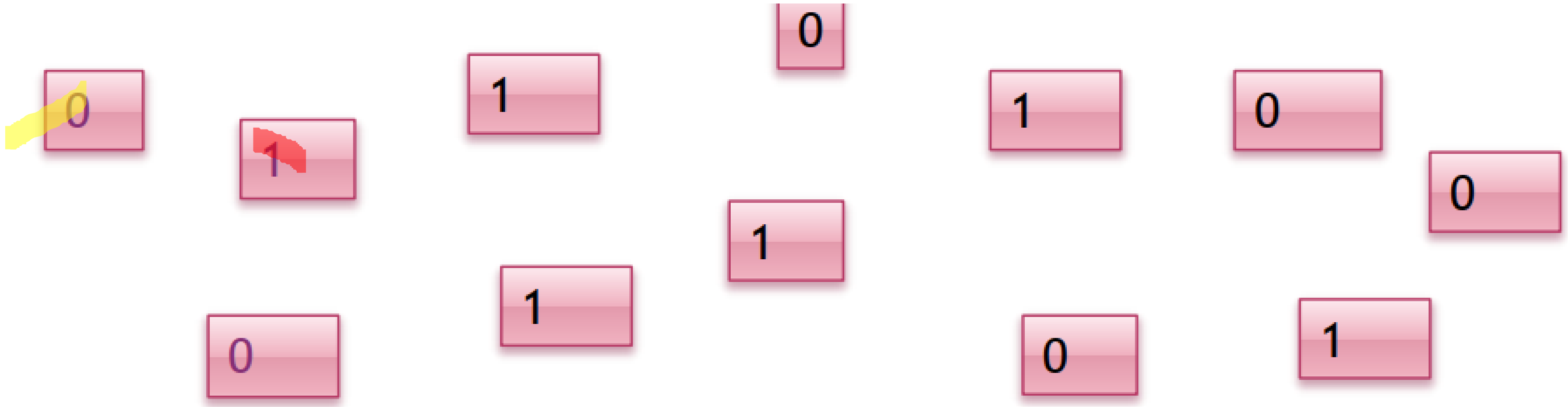
Digital Electronics-I

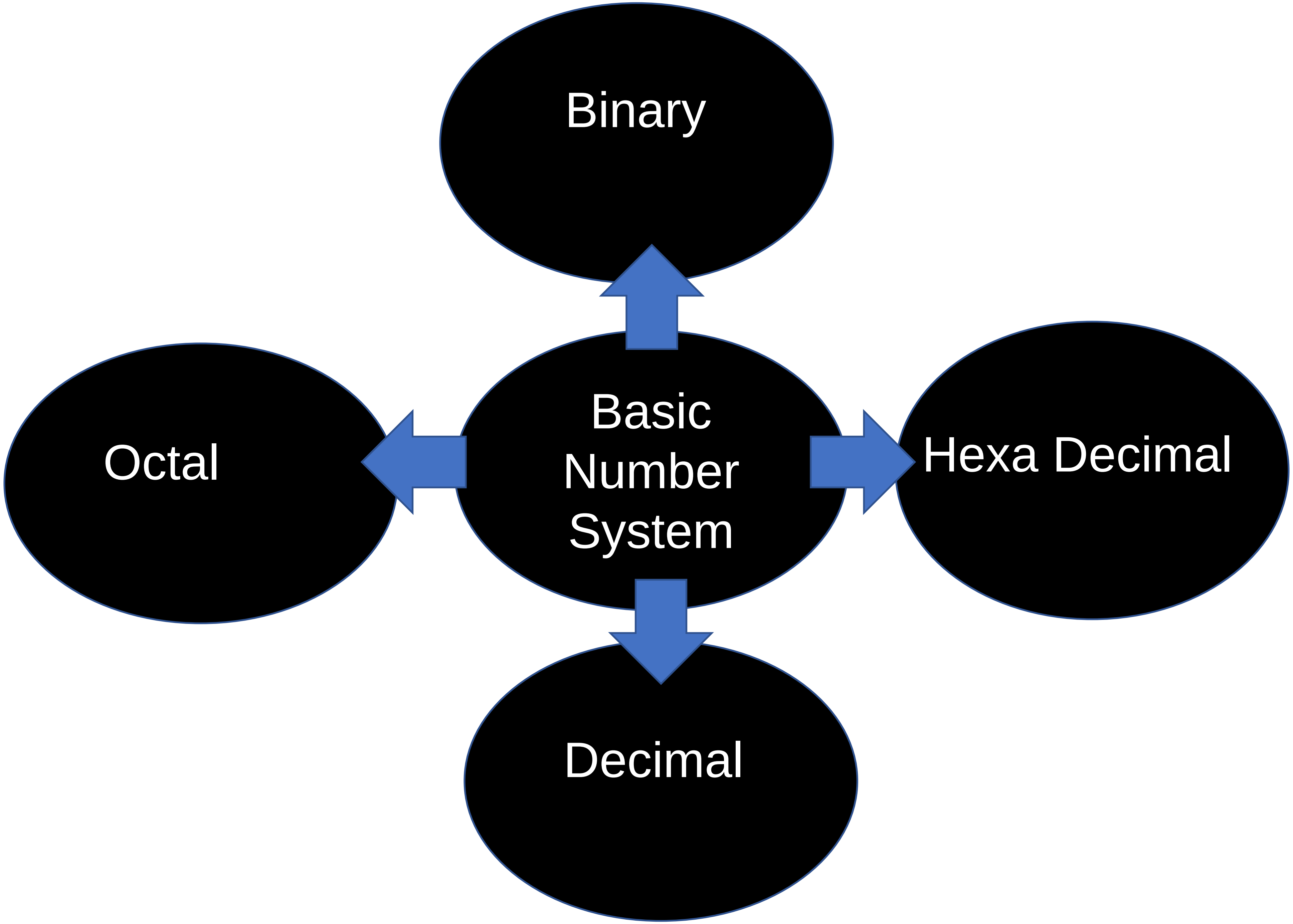
Ajith Kumar M M
Lecturer in electronics
MVGM, Government Polytechnic , vennikulam

How computers gets the answer?

128 + 43 = 171



- The computers understand information in terms of zeros and ones
- Number system using zeros and ones are called as binary number system
- Programmers uses decimal number system


Binary
Octal
Basic Number System
Hexa Decimal
Decimal

DECIMAL NUMBER SYSTEM

DIGITS	• 0,1,2,3,4,5,6,7,8,9.
BASE	• 10

10^n possible combinations with *n* digits

BINARY NUMBER SYSTEM

DIGITS	• 0, 1
BASE	• 2

2^n possible combinations with *n* digits

Each digit is called as “bit”

BINARY NUMBER SYSTEM

E.g. $(1011)_2$

Each digit is called as “bit”

2^n possible combinations with *n* digits

$(1011)_2 = (0001011)_2$

The bit in left side of a digital number is called as MSB & the bit in right side of a digital number is called as LSB

1011001

MSB LSB

HEXADECIMAL NUMBER SYSTEM

DIGITS	• 0,1,2,3,4,5,6,7, • 8,9,A,B,C,D,E,F.
BASE	• 16

16^n possible combinations with *n* digits

CONVERSIONS IN BASIC NUMBER SYSTEM

Decimal number system	Binary number system	Hexadecimal number system
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A

Decimal number system	Binary number system	Hexadecimal number system
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10
17	10001	11
18	10010	12
19	10011	13
20	10100	14
21	10101	15

BINARY TO DECIMAL

Convert binary number 10101 to decimal

Step-1: Find bit position start the LSB with position 0.

here

Binary number	1	0	1	0	1
Bit positions	4	3	2	1	0

Step-2: Multiply each bits with base

here

Binary number	1	0	1	0	1
Base	2^4	2^3	2^2	2^1	2^0

Step-3: Add to find the result

here $(1 \bullet 2^4)+(0 \bullet 2^3)+ (1 \bullet 2^2)+(0 \bullet 2^1)+ (1 \bullet 2^0)=$ **21**

HEXADECIMAL TO DECIMAL

Convert Hexadecimal number 5A9 to decimal

Step-1: Find bit position start the LSB with position 0.

here

Hexadecimal number	5	A	9
Bit positions	2	1	0

Step-2: Multiply each bits with base

here

Binary number	5	A	9
Base	16^2	16^1	16^0

Step-3: Add to find the result

here $(5\bullet 16^2)+(A\bullet 16^1)+(9\bullet 16^0)=$ **1449**

Decimal number system	Hexadecimal number system
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F

DECIMAL NUMBER SYSTEM

DIGITS	• 0,1,2,3,4,5,6,7,8,9.
BASE	• 10

BINARY NUMBER SYSTEM

DIGITS	• 0, 1
BASE	• 2

HEXADECIMAL NUMBER SYSTEM

DIGITS	• 0,1,2,3,4,5,6,7, • 8,9,A,B,C,D,E,F.
BASE	• 16

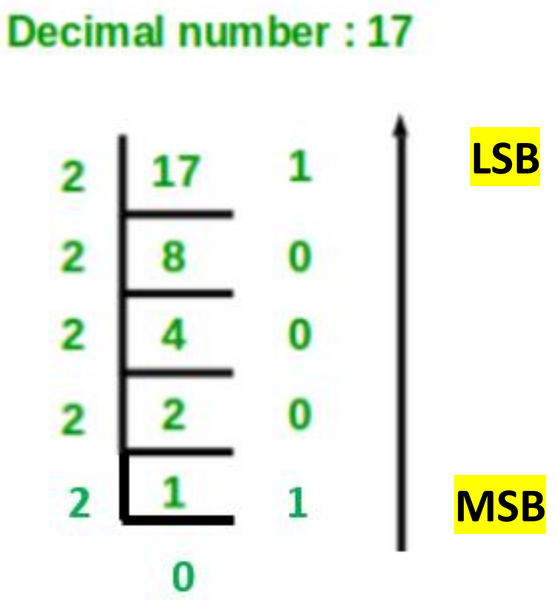
DECIMAL TO BINARY

Convert decimal number 17 to binary

Step-1: Divide the number though ought by 2 so that final output is 0

Step-2: Read the remainders from bottom to top

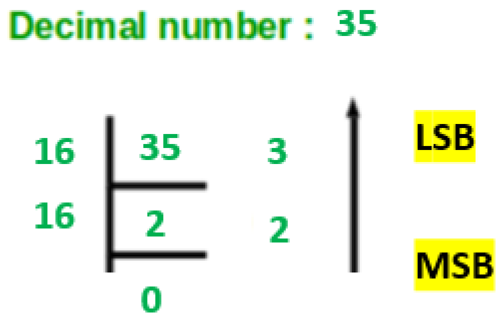
The binary number is 10001



DECIMAL TO HEX

Convert decimal number 35 to Hexadecimal

Step-1: Divide the number though ought by 16 so that final output is 0



Step-2: Read the remainders from bottom to top

The Hexadecimal number is 23

binary to HEX

Convert binary number 1001110 to Hexadecimal

Step-1: Form the group of 4 bits

here 100,**1110**

We can write it as 0100,**1110**

Step-2: Write hexadecimal equivalent of each groups

here **1110** → **E**

0100 → 4

Step-3: Now we can write the hexadecimal equivalent as

here 0100,**1110**

↓ ↓

4 E

The HEX number is 4E

Hexadecimal number system	Binary number system
0	**0000**
1	**0001**
2	**0010**
3	**0011**
4	**0100**
5	**0101**
6	**0110**
7	**0111**
8	**1000**
9	**1001**
A	**1010**
B	**1011**
C	**1100**
D	**1101**
E	**1110**
F	**1111**

HEX to binary

Convert Hexadecimal number 4E to binary number

Step-1: Find the 4 bit binary equivalent of each symbol in HEX number

here 4 → 0100
E → **1110**

Step-2: Now we can write the hexadecimal equivalent as

here 4 ↓ E ↓

0100,**1110**

The binary number is 01001110

Hexadecimal number system	Binary number system
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Binary arithmetic -addition

Case	A + B	Sum	Carry
1	0 + 0	0	0
2	0 + 1	1	0
3	1 + 0	1	0
4	1 + 1	0	1

Example − Addition

0011010 + 001100 = 00100110

$$
\begin{array}{rl}
1\,1 & \text{carry} \\
0\,0\,1\,1\,0\,1\,0 & = 26_{10} \\
+\,0\,0\,0\,1\,1\,0\,0 & = 12_{10} \\
\hline
0\,1\,0\,0\,1\,1\,0 & = 38_{10}
\end{array}
$$

Binary arithmetic -Subtraction

Case	A - B	Subtract	Borrow
1	0 - 0	0	0
2	1 - 0	1	0
3	1 - 1	0	0
4	0 - 1	0	1

Example − Subtraction

0011010 - 001100 = 00001110

$$
\begin{array}{rl}
1\ 1\ & \text{borrow} \\
0\ 0\ \cancel{1}\ \cancel{1}\ 0\ 1\ 0 & = 26_{10} \\
-\ 0\ 0\ 0\ 1\ 1\ 0\ 0 & = 12_{10} \\
\hline
0\ 0\ 0\ 1\ 1\ 1\ 0 & = 14_{10}
\end{array}
$$

Binary arithmetic -Multiplication

Case	A x B	Multiplication
1	0 x 0	0
2	0 x 1	0
3	1 x 0	0
4	1 x 1	1

Example − Multiplication

Example:

0011010 x 001100 = 100111000

```
      0 0 1 1 0 1 0    = 26₁₀
     x0 0 0 1 1 0 0    = 12₁₀
   ------------------
      0 0 0 0 0 0 0
     0 0 0 0 0 0 0
    0 0 1 1 0 1 0
   0 0 1 1 0 1 0
   ------------------
   0 1 0 0 1 1 1 0 0 0    = 312₁₀
```

Binary arithmetic -Multiplication

Case	A x B	Multiplication
1	0 x 0	0
2	0 x 1	0
3	1 x 0	0
4	1 x 1	1

Example − Multiplication

Example:

0011010 x 001100 = 100111000

```
     0011010    = 26₁₀
    x0001100    = 12₁₀
  ----------
     0000000
    0000000
   0011010
  0011010
  ----------
  0100111000    = 312₁₀
```

1's complement
&
2's complement

1's complement & 2's complement

- **We can find the 1's complement of the binary number by simply inverting the given number.**

- **For example, 1's complement of binary number 1011001 is 0100110.**

- **We can find the 2's complement of the binary number by adding 1 to the least significant bit of 1's complement of that binary number**

- **For example, 2's complement of binary number 1011001 is (0100110)+1=0100111.**

Binary subtraction (A-B) using 1's complement

- In the first step, find the 1's complement of the subtrahend (B).
- Next, add the complement number with the minuend (A).
- If got a carry, add the carry to its LSB.
- Else take 1's complement of the result which will be negative

21 – 7=14

(10101) → 21

(00111) → 7

(01110) → 14

Example 1: 10101 - 00111

1. We take 1's complement of subtrahend 00111, which is 11000.
2. Then find, 10101+11000 = 01101 with carry 1.
3. In the above result, we get the carry bit 1, so add this to the LSB of a given result, i.e., 01101+1=01110, which is the answer

Binary subtraction using 1's complement

21 – 23=-2

(10101) → 21

(10111) → 23

(00010) → 2

Example 2: 10101 - 10111

1. We take 1's complement of subtrahend 10111, which is 01000.
2. Then find, 10101+01000 = 11101 with carry 0.
3. In the above result, we get the carry bit 0, So calculate the 1's complement of the result, i.e., 00010, which is the negative number and the final answer.

Binary subtraction (A-B) using 2's complement

1. At first, 2’s complement of the subtrahend (B) is found.
2. Then it is added to the minuend (A).
3. If the final carry over of the sum is 1, it is dropped and the result is positive.
4. If there is no carry over, the two’s complement of the sum will be the result and it is negative

21 – 7=14

(10101) → 21

(00111) → 7

(01110) → 14

Example 1: 10101 - 00111

1. We take 2's complement of subtrahend 00111, which is 11001.
2. Then find, 10101+11001 = 01110 with carry 1.
3. In the above result, we get the carry bit 1, so we ignore the carry and the result is 01110, the result is positive

Binary subtraction using 2's complement

Example 2: 10101 - 10111

21 – 23=-2

(10101) $\rightarrow$ 21

(10111) $\rightarrow$ 23

(00010) $\rightarrow$ 2

1. We take 2's complement of subtrahend 10111, which is 01001.
2. Then find, 10101+01001 = 11110 with carry 0.
3. In the above result, we get the carry bit 0, So calculate the 2's complement of the result, i.e., 00010, which is the negative number and the final answer.

Binary codes

Binary codes

- In the coding, when numbers, letters or words are represented by a specific group of symbols, it is said that the number, letter or word is being encoded.
- The group of symbols is called as a code.
- The digital data is represented, stored and transmitted as group of binary bits.
- This group is also called as **binary code**.

Binary codes-Advantages

- Binary codes are suitable for the computer applications.
- Binary codes are suitable for the digital communications.
- Binary codes make the analysis and designing of digital circuits if we use the binary codes.
- Since only 0 & 1 are being used, implementation becomes easy

Binary codes-Binary Coded Decimal (BCD)

- In this code each decimal digit is represented by a 4-bit binary number.
- BCD is a way to express each of the decimal digits with a binary code.
- In the BCD, with four bits we can represent sixteen numbers (0000 to 1111).
- But in BCD code only first ten of these are used (0000 to 1001).
- The remaining six code combinations i.e. 1010 to 1111 are invalid in BCD.

Decimal	BCD
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

Binary codes-Binary Coded Decimal (BCD)

Advantages of BCD Codes

- It is very similar to decimal system.
- We need to remember binary equivalent of decimal numbers 0 to 9 only.

Disadvantages of BCD Codes

- The addition and subtraction of BCD have different rules.
- The BCD arithmetic is little more complicated.
- BCD needs more number of bits than binary to represent the decimal number. So BCD is less efficient than binary.

Binary codes-Binary Coded Decimal (BCD)

Example : convert 0110100000111001(BCD) to its decimal equivalent.

Solution:

Divide the BCD number into four-bit groups and convert each to decimal:

0110	1000	0011	1001
↓	↓	↓	↓
6	8	3	9

0110100000111001(BCD) = 6839_{10}

Binary codes-Binary Coded Decimal (BCD)

Example : Convert the following decimal and binary numbers to BCD.

a) 5648_{10}
b) 10001101_2

Solution:

a) $5648_{10} = 0101\ 0110\ 0100\ 1000$
b) $10001101_2 = 141_{10} = 0001\ 0100\ 0001$

Binary codes- Excess-3 code (XS-3)

- The Excess-3 code words are derived from the BCD code words adding $(0011)_2$ or (decimal value 3) to each code word in BCD.

Decimal	BCD 8 4 2 1	Excess-3 BCD + 0011
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

Binary codes- Gray Code

- It has a very special feature that, only one bit will change each time the decimal number is incremented.
- As only one bit changes at a time, the gray code is called as a unit distance code.
- Gray code cannot be used for arithmetic operation.

Decimal	BCD	Gray
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 1
3	0 0 1 1	0 0 1 0
4	0 1 0 0	0 1 1 0
5	0 1 0 1	0 1 1 1
6	0 1 1 0	0 1 0 1
7	0 1 1 1	0 1 0 0
8	1 0 0 0	1 1 0 0
9	1 0 0 1	1 1 0 1

Binary codes- Alphanumeric codes

- We need many symbols for communication between two computers
- These symbols are required to represent 26 alphabets with capital and small letters, numbers from 0 to 9, punctuation marks and other symbols.
- The alphanumeric codes are the codes that represent numbers and alphabetic characters.

- ASCII- American Standard Code for Information Interchange.
- EBCDIC- Extended Binary Coded Decimal Interchange Code.

- ASCII code is a 7-bit code
- EBCDIC is an 8-bit code.
- ASCII code is more commonly used worldwide
- EBCDIC is used primarily in large IBM computers.

Character	ASCII	EBCDIC	Character	ASCII	EBCDIC
Space	010 0000	0100 0000	A	100 0001	1100 0001
!	010 0001	0101 1010	B	100 0010	1100 0010
"	010 0010	0111 1111	C	100 0011	1100 0011
#	010 0011	0111 1011	D	100 0100	1100 0100
$	010 0100	0101 1011	E	100 0101	1100 0101
%	010 0101	0110 1100	F	100 0110	1100 0110
&	010 0110	0101 0000	G	100 0111	1100 0111
'	010 0111	0111 1101	H	100 1000	1100 1000
(	010 1000	0100 1101	I	100 1001	1100 1001
)	010 1001	0101 1101	J	100 1010	1101 0001
*	010 1010	0101 1100	K	100 1011	1101 0010
+	010 1011	0100 1110	L	100 1100	1101 0011
,	010 1100	0110 1011	M	100 1101	1101 0100
-	010 1101	0110 0000	N	100 1110	1101 0101
.	010 1110	0100 1011	O	100 1111	1101 0110
/	010 1111	0110 0001	P	101 0000	1101 0111
0	011 0000	1111 0000	Q	101 0001	1101 1000
1	011 0001	1111 0001	R	101 0010	1101 1001
2	011 0010	1111 0010	S	101 0011	1110 0010
3	011 0011	1111 0011	T	101 0100	1110 0011
4	011 0100	1111 0100	U	101 0101	1110 0100
5	011 0101	1111 0101	V	101 0110	1110 0101
6	011 0110	1111 0110	W	101 0111	1110 0110
7	011 0111	1111 0111	X	101 1000	1110 0111
8	011 1000	1111 1000	Y	101 1001	1110 1000
9	011 1001	1111 1001	Z	101 1010	1110 1001

Thank you!