NUMBER SYSTEMS

Malik Adnan Jaleel

Number Systems

Positional number systems

- Uses only a few symbols called digits
- These digits represents different values depending on the position they occupy in the number
- The value of each digit is determined by
  - the digit itself
  - the position of the digit in the number
  - the base of the number system
    - base is the total number of digits in the number system

Non-positional number systems

- Uses symbols such as I for 1, II for 2, III for 3 etc.
- Each symbol represents the same value regardless of its position in the number
- The symbols are simply added to find out the value of a particular number
- It is difficult to perform arithmetic with such a number system

Decimal Number System

- A positional number system which has 10 symbols or digits
- A total of 10 digits means base of decimal number system is 10

  0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- The maximum value of a single digit is 9 (one less than the value of the base)
- Each position of a digit represents a specific power of the base (10)
- The most popular and used by us in our day-to-day life

Example

$(2586)_{10}$

$(163)_{10}$

$(981)_{10}$

Binary Number System

A positional number system which has only two digits

- Base = 2, only two digits 0, 1
- The maximum value of a single digit is 1 (one less than the value of the base)
- Each position of a digit represents a specific power of the base (2)
- Mostly is used in the field of computer science

Example

$(10101)_2$

$(101111010)_2$

$(1110101)_2$

Octal Number System

- A positional number system which has eight digits
- Base = 8, total digits 0, 1, 2, 3, 4, 5, 6, 7
- The maximum value of a single digit is 7 (one less than the value of the base)
- Each position of a digit represents a specific power of the base (8)
- Not much used in the real-world mathematics

Example

$(2057)_8$

$(6605)_8$

$(321)_8$

Hexadecimal Number System

- A positional number system which has a total of 16 digits or symbols
- Base = 16, so total 16 symbols and digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- The symbols A, B, C, D, E, F represents decimal values 10, 11, 12, 13, 14 and 15
- The maximum value of a single digit is 15 (one less than the value of the base)
- Each position of a digit represents a specific power of the base (16)
- Not much used in the real-world mathematics

Example

$(1AF)_{16}$

$(9EA)_{16}$

$(664BD)_{16}$

Conversion between Number Systems

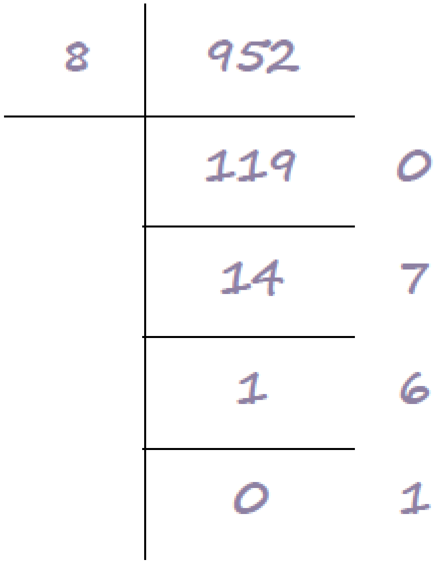
- **Converting a number of another base to decimal number**

Example:

$$
\begin{aligned}
(4706)_8 &= (4x8^3) + (7x8^2) + (0x8^1) + (6x8^0) \\
&= (4x512) + (7x64) + (0x8) + (6x1) \\
&= 2048 + 448 + 0 + 6 \\
&= (2502)_{10}
\end{aligned}
$$

Conversion between Number Systems

- □ **Converting a decimal number to a number of another base**
- □ **Example:**
- □ **$(952)_{10} = ?_8$**



8	952	
	119	0
	14	7
	1	6
	0	1

- □ **Hence, $(952)_{10} = (1670)_8$**

Data Types

- **Numeric data consists of only numbers, 0, 1, 2, …. , 9**
- **Alphabetic data consists of only letters A, B, C, …. , Z, in both uppercase and lowercase, and blank character**
- **Alphanumeric data is a string of symbols where a symbol may be one of the letters A, B, C, … , Z, in either uppercase or lowercase, or one of the digits 0, 1, 2, … , 9, or a special character such as + - * / (= etc.**

Bit, Byte, and Words

- In binary coding, every symbol that appears in the data is represented by a group of bits (Binary Digits)
- The group of bits used to represents a symbol is called a byte
- As most modern coding schemes use 8 bits to represents a symbol, the term byte is often used to mean a group of 8 bits
- To differentiate between bits and bytes, bit is written with a small ‘b’ while byte with a capital ‘B’

Bit, Byte, and Words

- A binary number represented using 8 bits is called a byte
- A less common, 4 bits representation is called nibble
- 16 bits is a word, 32 bits is a double word, and 64 bits is called a quadruple word
- Other quantifiers:

KILO	1K = 2^{10}
MEGA	1M = 2^{20}
GIGA	1G = 2^{30}
TERA	1T = 2^{40}
PETA	1P = 2^{50}

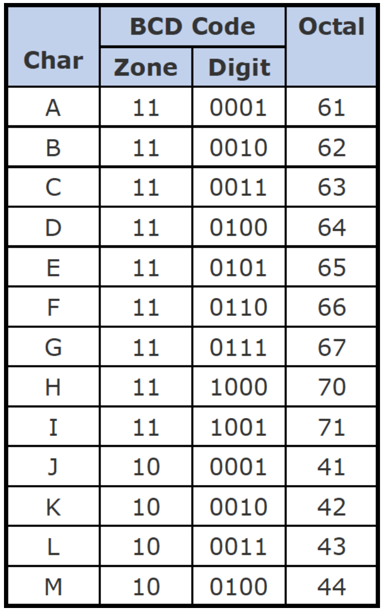
Computer Codes

- **Computer codes are used for internal representation of data in computers**
- **As computers use binary numbers for internal data representation, computers codes use binary coding schemes**
- **Commonly used computer codes are BCD, EBCDIC, ASCII and Unicode**

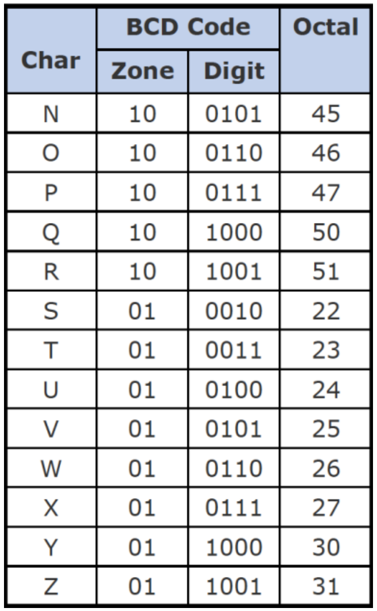
Computer Codes

- **BCD or Binary Coded Decimal**
- **It is one of the early computer codes**
- **It uses 6 bits to represents a symbol**
- **It can represents 64 (2^6) different characters**

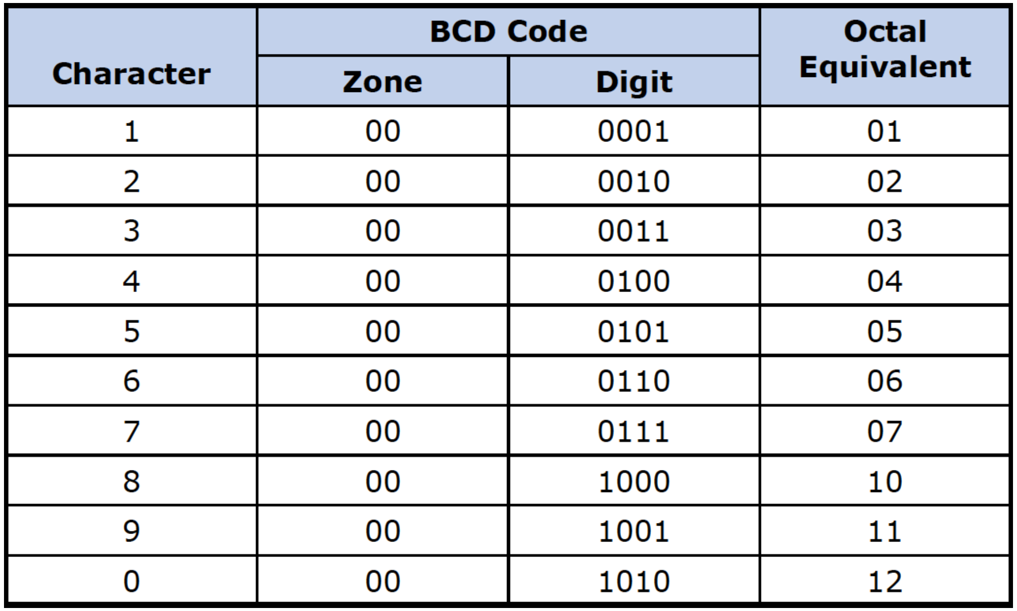
Computer Codes



Char	BCD Code		Octal
	Zone	Digit	
A	11	0001	61
B	11	0010	62
C	11	0011	63
D	11	0100	64
E	11	0101	65
F	11	0110	66
G	11	0111	67
H	11	1000	70
I	11	1001	71
J	10	0001	41
K	10	0010	42
L	10	0011	43
M	10	0100	44



Char	BCD Code		Octal
	Zone	Digit	
N	10	0101	45
O	10	0110	46
P	10	0111	47
Q	10	1000	50
R	10	1001	51
S	01	0010	22
T	01	0011	23
U	01	0100	24
V	01	0101	25
W	01	0110	26
X	01	0111	27
Y	01	1000	30
Z	01	1001	31



Character	BCD Code		Octal Equivalent
	Zone	Digit	
1	00	0001	01
2	00	0010	02
3	00	0011	03
4	00	0100	04
5	00	0101	05
6	00	0110	06
7	00	0111	07
8	00	1000	10
9	00	1001	11
0	00	1010	12

Computer Codes

Example:

- **The binary digits used to record the word BASE in BCD**

  B = 110010

  A = 110001

  S = 010010

  E = 110101

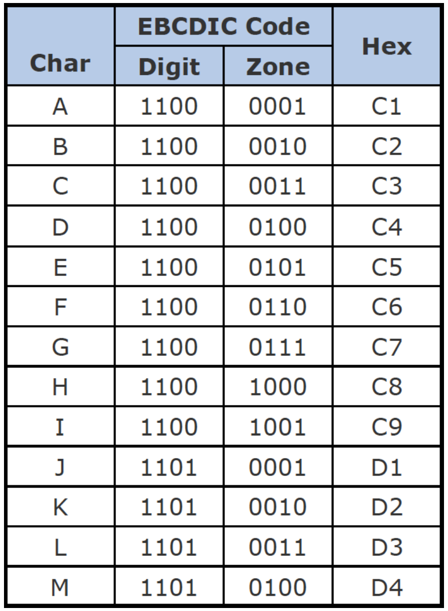
- **So, 110010 110001 010010 110101 will record BASE in BCD**

Computer Codes

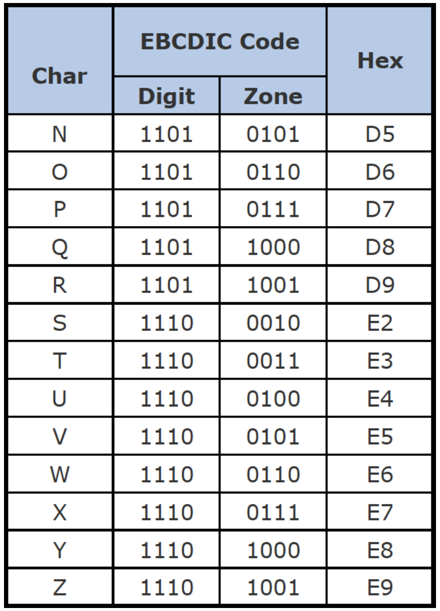
EBCDIC or Extended Binary Coded Decimal Interchange Code

- **It uses 8 bits to represents a symbol**
- **It can represents 256 (2^8) different characters**

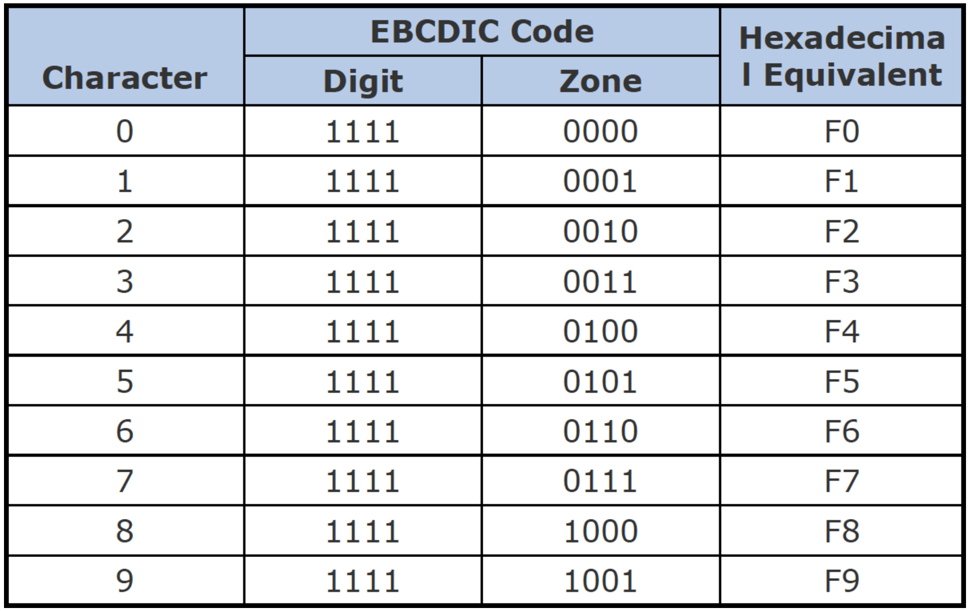
Computer Codes



Char	EBCDIC Code		Hex
	Digit	Zone	
A	1100	0001	C1
B	1100	0010	C2
C	1100	0011	C3
D	1100	0100	C4
E	1100	0101	C5
F	1100	0110	C6
G	1100	0111	C7
H	1100	1000	C8
I	1100	1001	C9
J	1101	0001	D1
K	1101	0010	D2
L	1101	0011	D3
M	1101	0100	D4



Char	EBCDIC Code		Hex
	Digit	Zone	
N	1101	0101	D5
O	1101	0110	D6
P	1101	0111	D7
Q	1101	1000	D8
R	1101	1001	D9
S	1110	0010	E2
T	1110	0011	E3
U	1110	0100	E4
V	1110	0101	E5
W	1110	0110	E6
X	1110	0111	E7
Y	1110	1000	E8
Z	1110	1001	E9



Character	EBCDIC Code		Hexadecimal Equivalent
	Digit	Zone	
0	1111	0000	F0
1	1111	0001	F1
2	1111	0010	F2
3	1111	0011	F3
4	1111	0100	F4
5	1111	0101	F5
6	1111	0110	F6
7	1111	0111	F7
8	1111	1000	F8
9	1111	1001	F9

Computer Codes

Example:

- **The binary digits used to record the work BIT in EBCDIC**

  B = 11000010

  I = 11001001

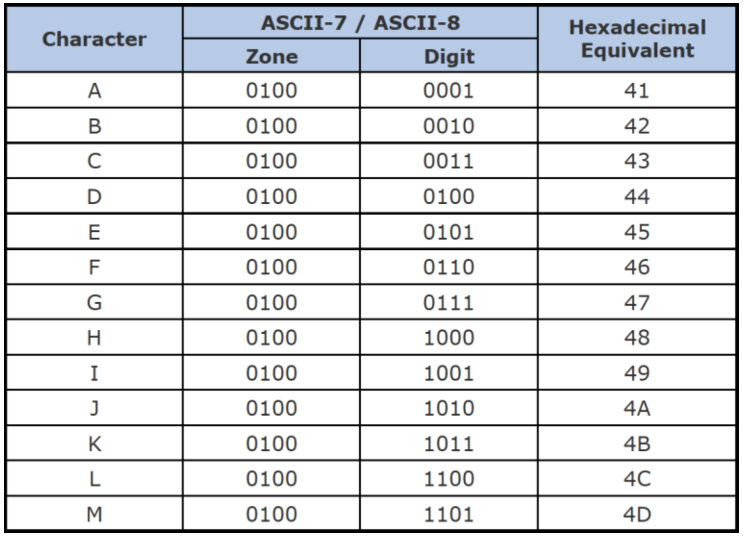
  T = 11100011

- **So, 11000010 11001001 11100011 will record BIT in EBCDIC**

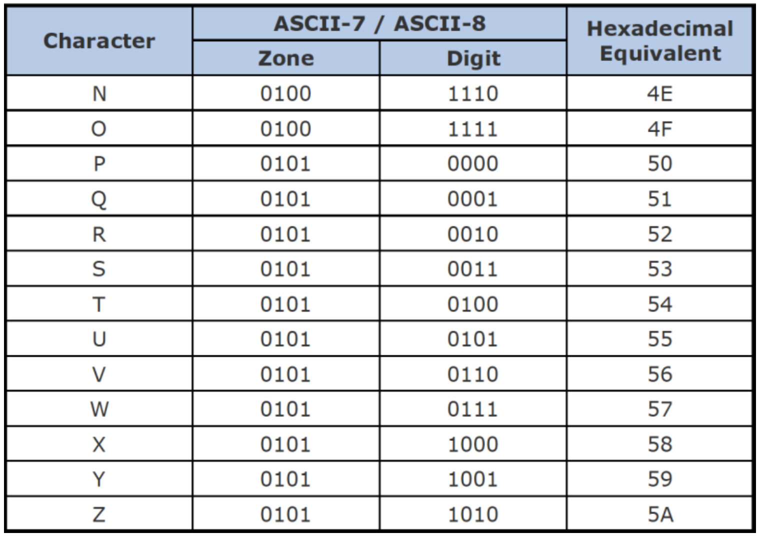
Computer Codes

- **ASCII or American Standard Code for Information Interchange**
- **It is of type types, ASCII-7 and ASCII-8**
- **ASCII-7 uses 7 bits (2^7) and ASCII-8 uses 8 bits (2^8) to represents a symbol**
- **First 128 characters in both ASCII-7 and ASCII-8 are same**

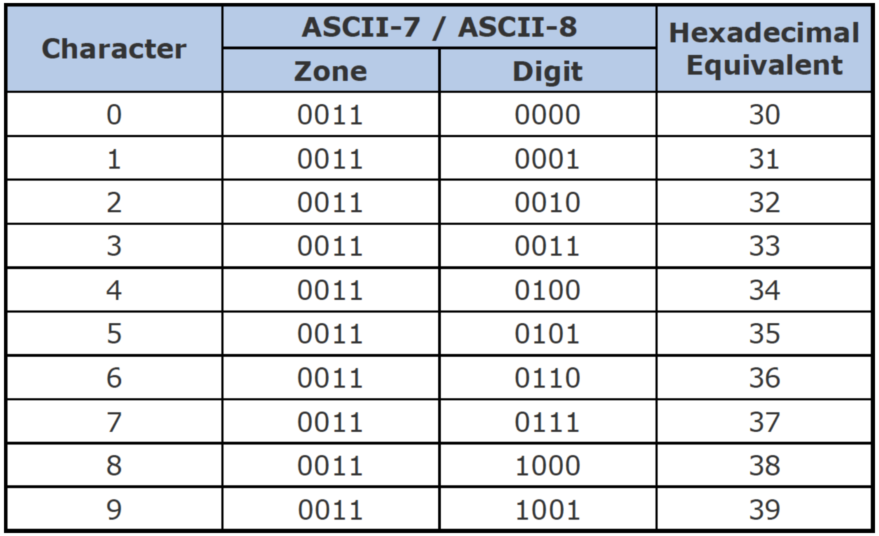
Computer Codes



Character	ASCII-7 / ASCII-8		Hexadecimal Equivalent
	Zone	Digit	
A	0100	0001	41
B	0100	0010	42
C	0100	0011	43
D	0100	0100	44
E	0100	0101	45
F	0100	0110	46
G	0100	0111	47
H	0100	1000	48
I	0100	1001	49
J	0100	1010	4A
K	0100	1011	4B
L	0100	1100	4C
M	0100	1101	4D



Character	ASCII-7 / ASCII-8		Hexadecimal Equivalent
	Zone	Digit	
N	0100	1110	4E
O	0100	1111	4F
P	0101	0000	50
Q	0101	0001	51
R	0101	0010	52
S	0101	0011	53
T	0101	0100	54
U	0101	0101	55
V	0101	0110	56
W	0101	0111	57
X	0101	1000	58
Y	0101	1001	59
Z	0101	1010	5A



Character	ASCII-7 / ASCII-8		Hexadecimal Equivalent
	Zone	Digit	
0	0011	0000	30
1	0011	0001	31
2	0011	0010	32
3	0011	0011	33
4	0011	0100	34
5	0011	0101	35
6	0011	0110	36
7	0011	0111	37
8	0011	1000	38
9	0011	1001	39

Computer Codes

Example:

- **The binary digits used to record the work BOY in ASCII-7**

  B = 1000010

  O = 1001111

  Y = 1011001

- **So, 100010 1001111 1011001 will record BOY in ASCII-7**

Computer Codes

Example:

- **The binary digits used to record the work SKY in ASCII-8**

  S = 01010011

  K = 01001011

  Y = 01011001

- **So, 01010011 01001011 01011001 will record SKY in ASCII-8**

Computer Codes

- Unicode or Universal Code, provides a consistent way to encoding multilingual plain text
- Defines codes for characters used in all major languages of the world
- Defines codes for special characters, mathematical symbols, technical symbols and diacritics
- Capacity to encode as many as million characters
- Assigns each character a unique numeric value and name
- Reserves a part of the code space for private use
- Affords simplicity and consistency for ASCII, even corresponding characters have same code
- Specifies an algorithm for the presentation of text with bi-directional behaviour
- Unicode Transformation Format - UTF-8, UTF-16 and UTF-32 variants
  - Read more about Unicode at https://unicode-table.com/en/