• Binary-Coded Decimal Code

→ Although the binary number system is the most natural system for a computer because it is readily represented in today's electronic technology, most people are more accustomed to the decimal system.

→ One way to resolve this difference is to convert decimal numbers to binary, perform all arithmetic calculations in binary, and then convert the binary results back to decimal.

→ Since the computer can accept only binary values we must represent the decimal digits by means of a code that contains 1's and 0's

→ It is also possible to perform the arithmetic operations directly on decimal numbers when they are stored in the computer in coded form.

→ The 10 decimal digits form a set.

→ A binary code that distinguishes among 10 elements must contain at least four bits, but 6 out of the 16 possible combinations remain unassigned.

→ Different binary codes can be obtained by arranging four bits into 10 distinct combinations

→ The code most commonly used for decimal digits is the straight binary assignment. This scheme is called binary-coded decimal and is commonly referred to as BCD

→ A number with k decimal digits require 4k bits in BCD. Each group of 4 bits represent one decimal digit.

$$(185)_{10} = (0001\,1000\,0101)_{BCD} = (10111001)_2$$

Decimal Symbol	BCD Digit
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

→ A decimal number in BCD is same as its equivalent binary number only when the number is between 0 and 9.

→ A BCD number greater than 10 looks different from its equivalent binary number, even though both contain 1's and 0's.

→ The representation of a BCD number needs more bits than its equivalent binary value.

→ The only difference between a decimal number and BCD is that decimals are written with the symbols 0, 1, 2, ..., 9. and BCD numbers use the binary code 0000, 0001, 0010, ..., 1001

→ Decimal 10 is represented in BCD with eight bits as 0001 0000 and decimal 15 as 0001 0101. The corresponding binary values are 1010 and 1111 and have only four bits.

BCD Addition

→ Consider the addition of two decimal digits in BCD, together with a possible carry from a previous less significant pair of digits

→ Since each digit does not exceed 9, the sum cannot be greater than $9+9+1=19$, with the 1 being a previous carry.

→ If we add the BCD digits as binary numbers, then the binary sum will produce a result in the range from 0 to 19.

→ In binary, this range will be from 0000 to 10011, but in BCD, it is from 0000 to 11001, with the first (i.e. leftmost) 1 being a carry and the next four bits being the BCD sum.

→ When the binary sum is equal to less than 1001 (without a carry), the corresponding BCD digit is correct. However, when the binary sum is greater than or equal to 1010, the result is an invalid BCD digit.

→ The addition of $6 = (0110)_2$ to the binary sum converts it to the correct digit and also produces a carry as required. This is because a carry in the most significant bit position of the binary sum and a decimal carry differ by $16-10=6$

4	0100	4	0100	8	1000
+5	+0101	+8	+1000	+9	+1001
9	1001	12	1100	17	10001
			0110		+0110
			10010		10111

→ Consider the addition of $184+576=760$ in BCD

	184	0001	1000 0100
	+576	+0101	0111 0110
Binary Sum		0110	1111 1010
Add 6			0110 0110
BCD Sum 760		0111	0110 0000

Decimal Arithmetic

→ The representation of signed decimal numbers in BCD is similar to the representation of signed numbers in binary.

→ We can use either the signed-magnitude system or the signed-complement system.

→ The sign of a decimal number is usually represented with four bits to confirm the 4-bit code of the decimal digits.

→ It is customary to designate a plus with four 0's and a minus with the BCD equivalent of 9, which is 1001.

→ The signed-complement system can be either the 9's or the 10's complement, but the 10's complement is the one most often used.

→ Addition is done by summing all digits, including the sign digit, and discarding the end carry. This operation assumes that all negative numbers are in 10's-complement form.

→ consider the addition (+375)+(-240) = +135, done in the signed-complement system.

```
  0 375
+ 9 760
-------
 (*)0 135
```

→ Many computers have special hardware to perform arithmetic calculations directly with decimal numbers in BCD.