DIGITAL LOGIC DESIGN

Lecture 3

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**.
- The binary code is represented by the number as well as alphanumeric letter
- 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

- The codes are broadly categorized into following four categories.
- Weighted Codes
- Non-Weighted Codes
- Binary Coded Decimal Code
- Alphanumeric Codes
- Error Detecting Codes
- Error Correcting Codes

BINARY CODES

- Digital systems use signals that have two distinct values and circuit elements that have two stable states.
- There is a direct analogy among binary signals, binary circuit elements, and binary digits.
- A binary number of n digits, for example, may be represented by n binary circuit elements, each having an output signal equivalent to 0 or 1.
- Digital systems represent and manipulate not only binary numbers, but also many other discrete elements of information.
- Any discrete element of information that is distinct among a group of quantities can be represented with a binary code (i.e., a pattern of 0's and 1's).
- The codes must be in binary because, in today's technology, only circuits that represent and manipulate patterns of 0's and 1's can be manufactured economically for use in computers.

BINARY CODES

- However, it must be realized that binary codes merely change the symbols, not the meaning of the elements of information that they represent.
- If we inspect the bits of a computer at random, we will find that most of the time they represent some type of coded information rather than binary numbers.
- An *n*-bit binary code is a group of *n* bits that assumes up to 2^n distinct combinations of 1's and 0's, with each combination representing one element of the set that is being coded.
- A set of four elements can be coded with two bits, with each element assigned one of the following bit combinations: 00, 01, 10, 11.
- A set of eight elements requires a three-bit code and a set of 16 elements requires a four-bit code.

BINARY CODES

- The bit combination of an *n*-bit code is determined from the count in binary from 0 to $2^n - 1$.
- Each element must be assigned a unique binary bit combination, and no two elements can have the same value; otherwise, the code assignment will be ambiguous.
- Although the *minimum* number of bits required to code 2^n distinct quantities is *n*, there is no *maximum* number of bits that may be used for a binary code.
- For example, the 10 decimal digits can be coded with 10 bits, and each decimal digit can be assigned a bit combination of nine 0's and a 1.
- In this particular binary code, the digit 6 is assigned the bit combination 0001000000.

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.
- This method requires that we store decimal numbers in the computer so that they can be converted to binary.
- 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.

Binary-Coded Decimal Code

- A binary code will have some unassigned bit combinations if the number of elements in the set is not a multiple power of 2.
- The 10 decimal digits form such 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 the decimal digits is the straight binary assignment listed in Table 1.4 .
- This scheme is called *binary-coded decimal* and is commonly referred to as BCD.

Binary-Coded Decimal Code

Table 1.4
Binary-Coded Decimal (BCD)

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

Binary-Coded Decimal Code

- Table 1.4 gives the four-bit code for one decimal digit. A number with *k* decimal digits will require 4*k* bits in BCD.
- Decimal 396 is represented in BCD with 12 bits as 0011 1001 0110, with **each group of 4 bits representing one decimal digit.**
- A decimal number in BCD is the 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.
- Moreover, **the binary combinations 1010 through 1111 are not used and have no meaning in BCD.** Consider decimal 185 and its corresponding value in BCD and binary:
- $(185)_{10} = (0001\ 1000\ 0101)_{BCD} = (10111001)_2$

Binary-Coded Decimal Code

- The BCD value has 12 bits to encode the characters of the decimal value, but the equivalent binary number needs only 8 bits.
- It is obvious that the representation of a BCD number needs more bits than its equivalent binary value.
- However, there is an advantage in the use of decimal numbers, because computer input and output data are generated by people who use the decimal system.
- It is important to realize that BCD numbers are decimal numbers and not binary numbers, although they use bits in their representation.
- 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, c, 1001.

Binary-Coded Decimal Code (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.
- Suppose we add the BCD digits as if they were 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 1 1001, 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 or 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.

Binary-Coded Decimal Code (BCD) Addition

- 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

Binary-Coded Decimal Code (BCD) Addition

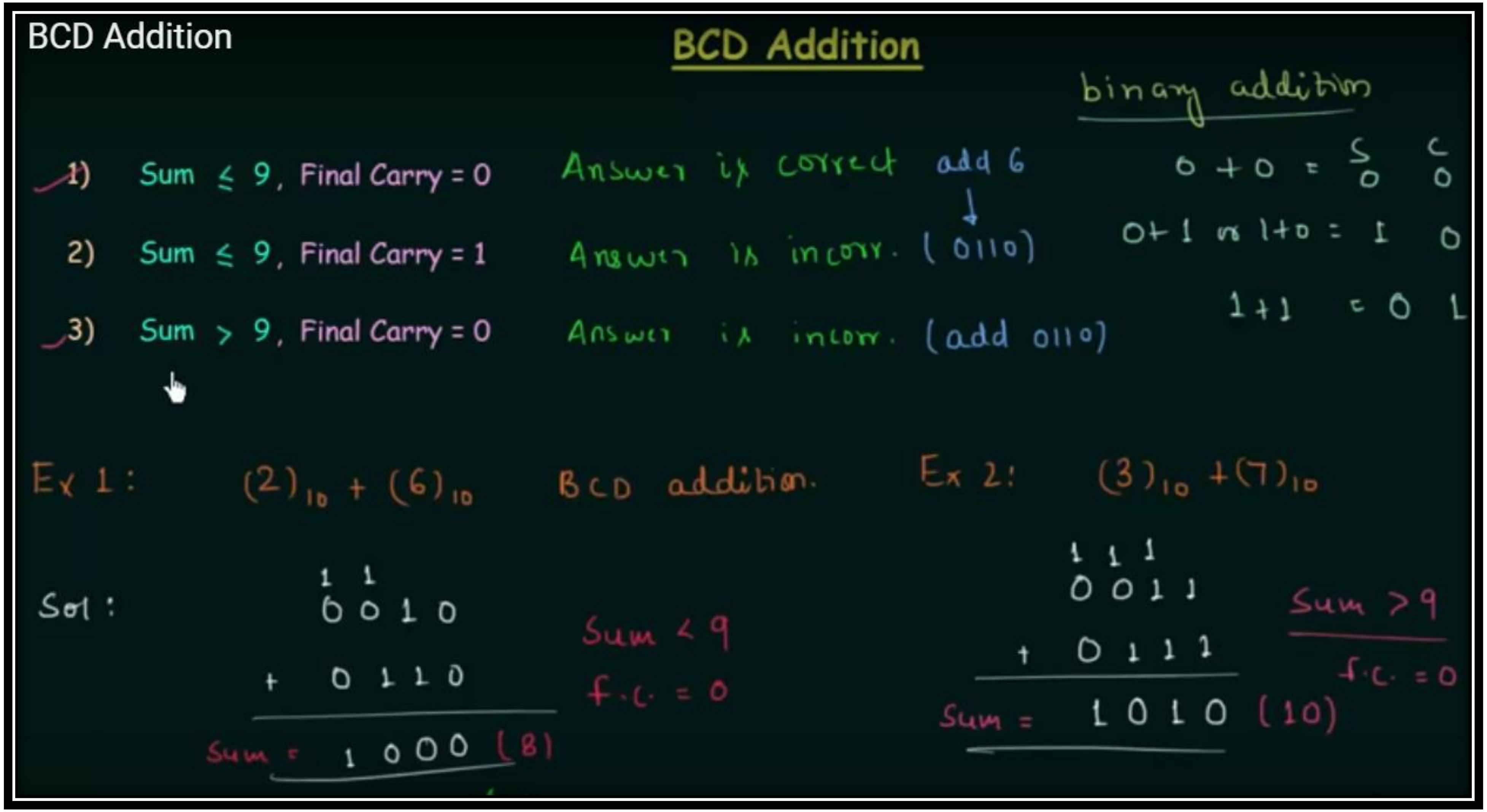
- In each case, the two BCD digits are added as if they were two binary numbers.
- If the binary sum is greater than or equal to 1010, we add 0110 to obtain the correct BCD sum and a carry.
- In the first example, the sum is equal to 9 and is the correct BCD sum.
- In the second example, the binary sum produces an invalid BCD digit.
- The addition of 0110 produces the correct BCD sum, 0010 (i.e., the number 2), and a carry.
- In the third example, the binary sum produces a carry. This condition occurs when the sum is greater than or equal to 16.
- Although the other four bits are less than 1001, the binary sum requires a correction because of the carry. Adding 0110, we obtain the required BCD sum 0111 (i.e., the number 7) and a BCD carry.

Binary-Coded Decimal Code (BCD) Addition

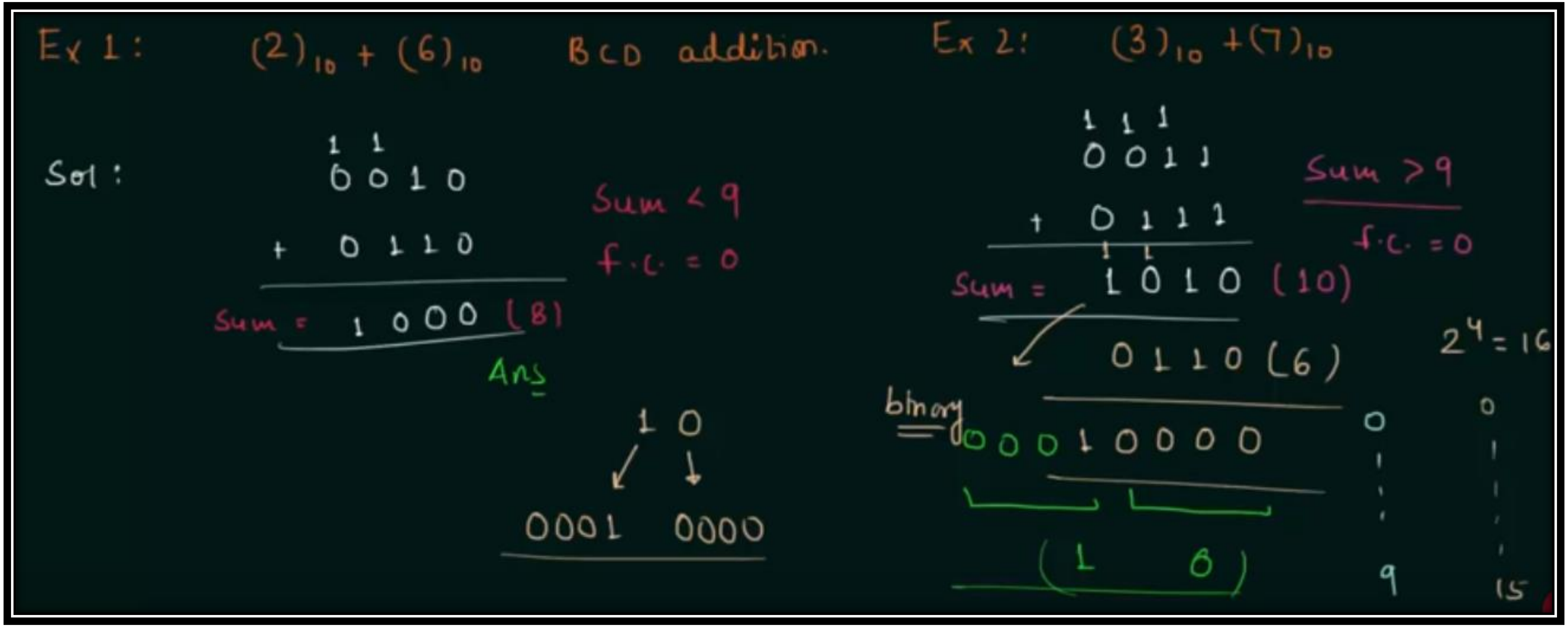
- The addition of two *n*-digit unsigned BCD numbers follows the same procedure.
- Consider the addition of 184 + 576 = 760 in BCD:

BCD	1	1		
	0001	1000	0100	184
	+0101	0111	0110	+576
Binary sum	0111	10000	1010	
Add 6		0110	0110	
BCD sum	0111	0110	0000	760

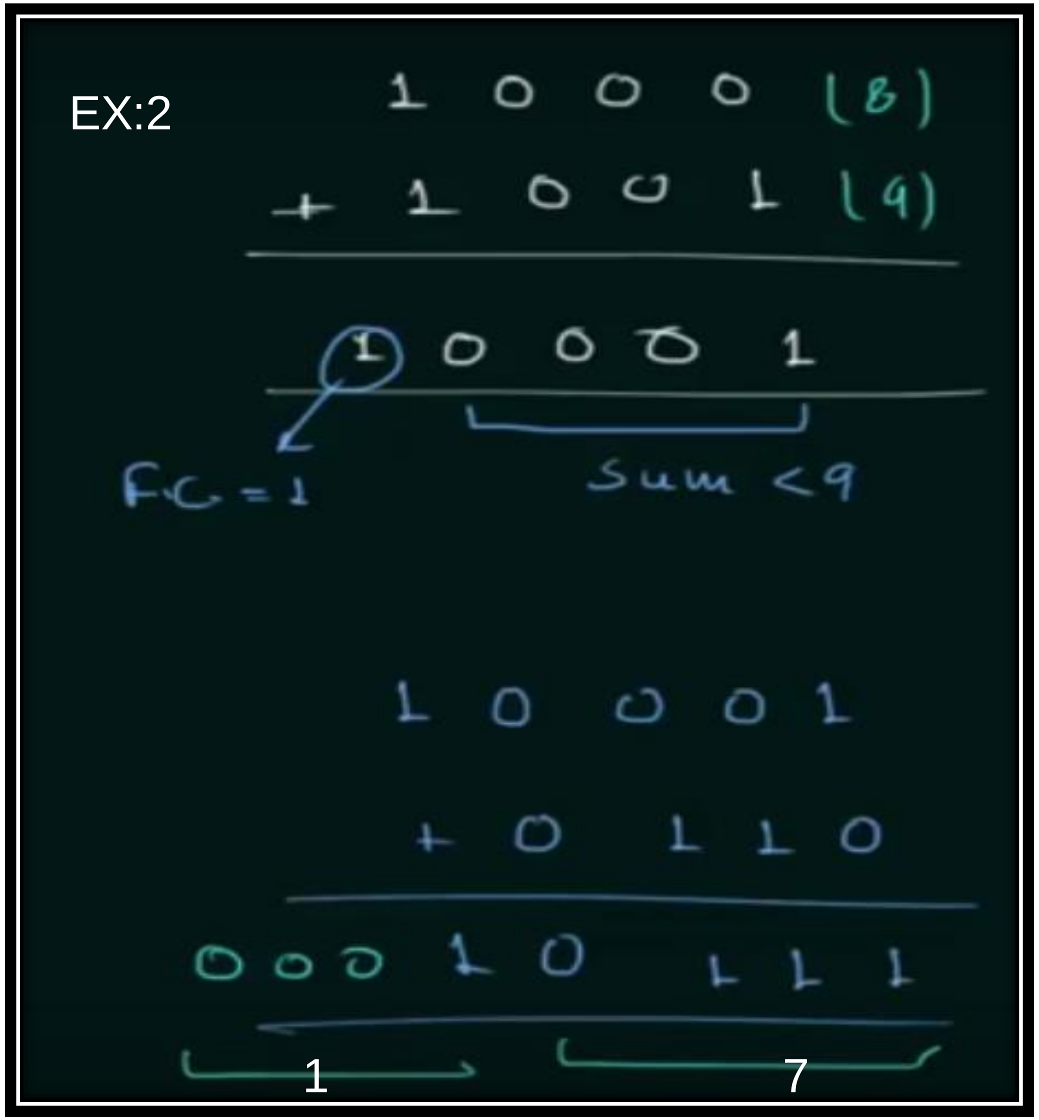
Binary-Coded Decimal Code (BCD) Addition



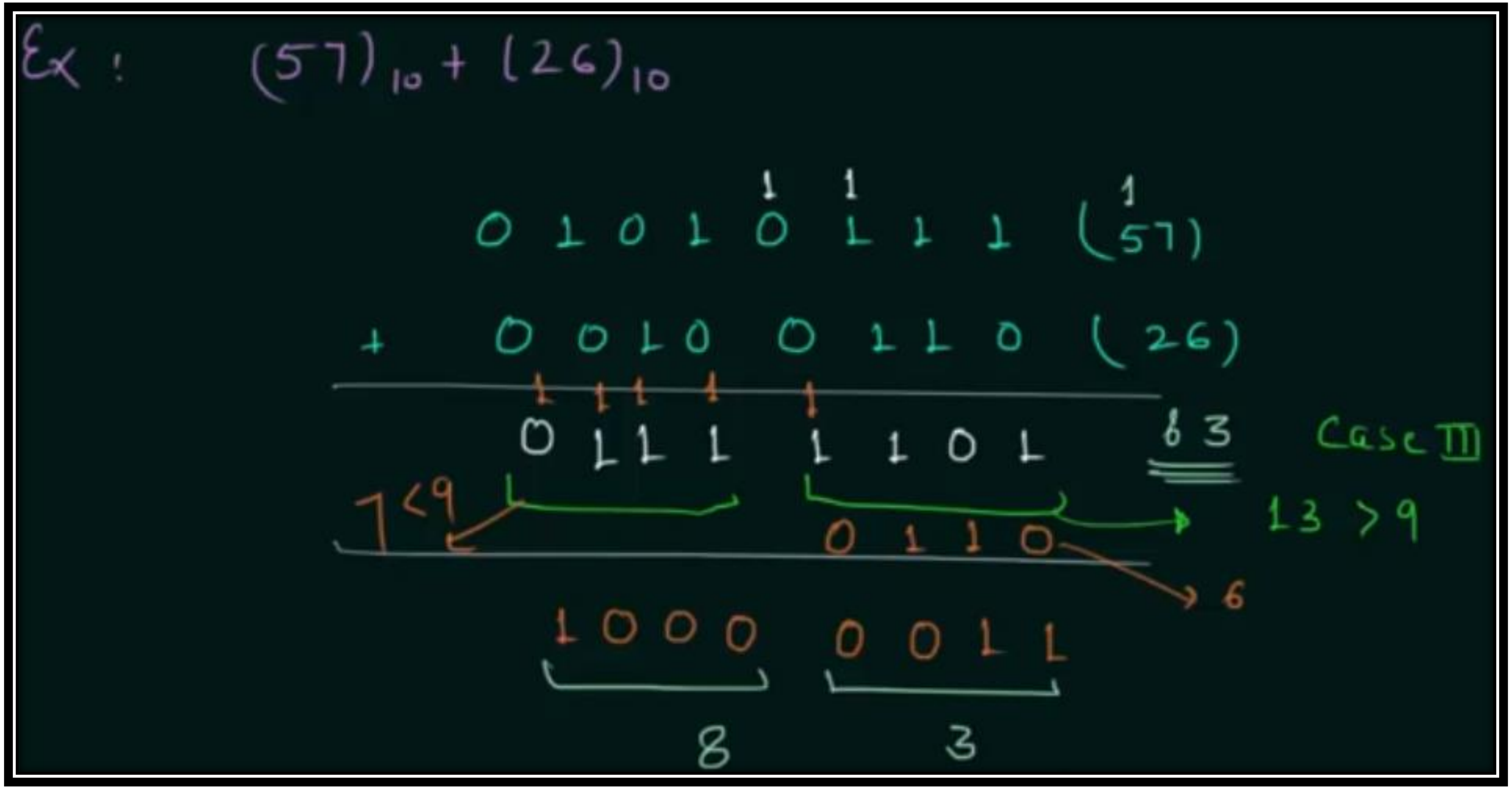
Binary-Coded Decimal Code (BCD) Addition



Binary-Coded Decimal Code (BCD) Addition



Binary-Coded Decimal Code (BCD) Addition



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 familiar signed-magnitude system or the signed-complement system.
- The sign of a decimal number is usually represented with four bits to conform to the four-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-magnitude system is seldom used in computers.
- 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.
- To obtain the 10’s complement of a BCD number, we first take the 9’s complement and then add 1 to the least significant digit.

Decimal Arithmetic

- The 9’s complement is calculated from the subtraction of each digit from 9.
- The procedures developed for the signed-2’s-complement system in the previous section also apply to the signed-10’s-complement system for decimal numbers.
- 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:

$$\begin{array}{rr} 0 & 375 \\ +9 & 760 \\ \hline 0 & 135 \end{array}$$

Decimal Arithmetic

- The 9 in the leftmost position of the second number represents a minus, and 9760 is the 10’s complement of 0240.
- The two numbers are added and the end carry is discarded to obtain +135.
- Of course, the decimal numbers inside the computer, including the sign digits, must be in BCD.
- The addition is done with BCD digits as described previously.
- The subtraction of decimal numbers, either unsigned or in the signed-10’s-complement system, is the same as in the binary case:
- Take the 10’s complement of the subtrahend and add it to the minuend.
- Many computers have special hardware to perform arithmetic calculations directly with decimal numbers in BCD.

Other Decimal Codes

- Binary codes for decimal digits require a minimum of four bits per digit.
- Many different codes can be formulated by arranging four bits into 10 distinct combinations.
- BCD and three other representative codes are shown in Table 1.5 .
- Each code uses only 10 out of a possible 16 bit combinations that can be arranged with four bits.
- The other six unused combinations have no meaning and should be avoided.
- BCD and the 2421 code are examples of weighted codes.
- In a weighted code, each bit position is assigned a weighting factor in such a way that each digit can be evaluated by adding the weights of all the 1's in the coded combination.
- The BCD code has weights of 8, 4, 2, and 1, which correspond to the power-of-two values of each bit.

Other Decimal Codes

Table 1.5
Four Different Binary Codes for the Decimal Digits

Decimal Digit	BCD 8421	2421	Excess-3	8, 4, −2, −1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	1011	1000	1011
6	0110	1100	1001	1010
7	0111	1101	1010	1001
8	1000	1110	1011	1000
9	1001	1111	1100	1111
	1010	0101	0000	0001
Unused	1011	0110	0001	0010
bit	1100	0111	0010	0011
combi-	1101	1000	1101	1100
nations	1110	1001	1110	1101
	1111	1010	1111	1110

Other Decimal Codes

- The bit assignment 0110, for example, is interpreted by the weights to represent decimal 6 because 8 * 0 + 4 * 1 + 2 * 1 + 1 * 0 = 6.
- The bit combination 1101, when weighted by the respective digits 2421, gives the decimal equivalent of 2 * 1 + 4 * 1 + 2 * 0 + 1 * 1 = 7.
- Note that some digits can be coded in two possible ways in the 2421 code.
- For instance, decimal 4 can be assigned to bit combination 0100 or 1010, since both combinations add up to a total weight of 4.

Other Decimal Codes

- BCD adders add BCD values directly, digit by digit, without converting the numbers to binary.
- However, it is necessary to add 6 to the result if it is greater than 9.
- BCD adders require significantly more hardware and no longer have a speed advantage of conventional binary adders.
- The 2421 and the excess-3 codes are examples of self-complementing codes.
- Such codes have the property that the 9’s complement of a decimal number is obtained directly by changing 1’s to 0’s and 0’s to 1’s.
- For example, decimal 395 is represented in the excess-3 code as 0110 1100 1000.
- The 9’s complement of 604 is represented as 1001 0011 0111, which is obtained simply by complementing each bit of the code.

Other Decimal Codes

- The excess-3 code has been used in some older computers because of its self-complementing property.
- **Excess-3 is an unweighted code in which each coded combination is obtained from the corresponding binary value plus 3.**
- The Excess-3 code is also called as XS-3 code. It is non-weighted code used to express decimal numbers. The Excess-3 code words are derived from the 8421 BCD code words adding (0011)2 or (3)10 to each code word in 8421.
- Note that the BCD code is not self-complementing.
- The 8, 4, -2, -1 code is an example of assigning both positive and negative weights to a decimal code.
- In this case, the bit combination 0110 is interpreted as decimal 2 and is calculated from 8 * 0 + 4 * 1 + (-2) * 1 + (-1) * 0 = 2.