mirafra
TECHNOLOGIES

Digital Electronics

Dr. Ipsita Biswas Mahapatra
Member of Technical Staff

Goals & Objectives

- ➢ Introduction to number systems—decimal number, binary number, hexadecimal number and octal number
- ➢ Introduction and explanation of binary number system and number conversion
- ➢ Introduction and explanation of decimal number system and number conversion
- ➢ Conversion of octal number system—octal to binary and binary to octal conversion
- ➢ Conversion of hexadecimal number system—hexadecimal to binary and binary to hexadecimal
- ➢ Binary arithmetic: Addition & Subtraction
- ➢ Complements of number systems
- ➢ Boolean subtraction using 1’s and 2’s complement

Number system

- There are four number systems of arithmetic that are used in the digital systems:
  - ✓ Decimal
  - ✓ Binary
  - ✓ Hexa-decimal
  - ✓ Octal

Decimal number system

- The decimal number system has ten symbols, and any number of any magnitude can be expressed by using this system of positional weighting.
- Digits: 0 1 2 3 4 5 6 7 8 9
- Radix: $(..)_{10}$, $(..)_{D,}$ $(..)_{d}$
- For example, 6841 can be broken down as

6841 = 6000 + 800 + 40 + 1

=6 × 10^3 + 8 × 10^2 + 4 × 10^1 + 1 × 10^0

Binary number system

- In this, each position in a number can take only one of two values: 0 or 1.
- These positions are called bits, a contraction of the word's binary digits.
- From the table it is clear that, using 4 bits we can represent decimal numbers from 0 to 15.
- Like the decimal system, the binary is also positionally weighted. Each position represents a particular value of 2^n.

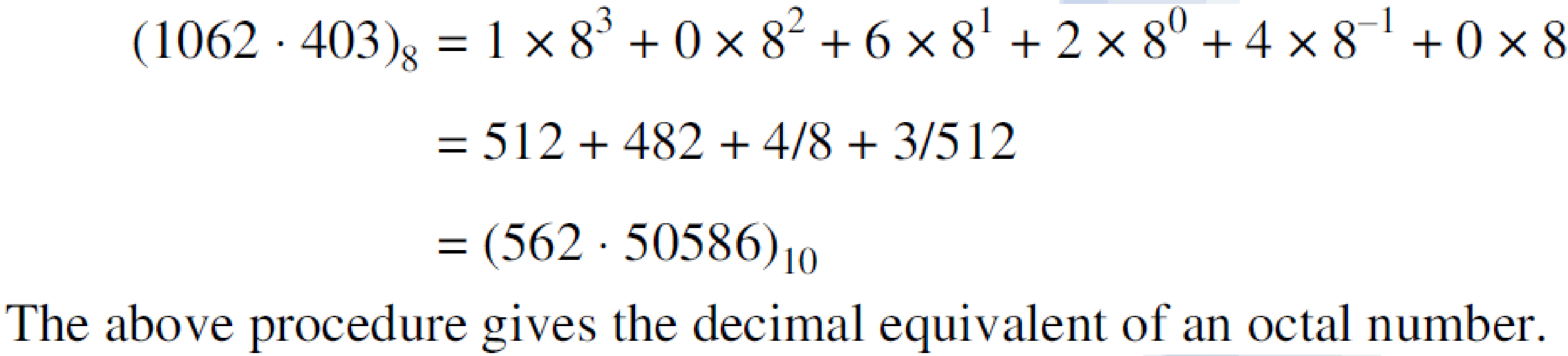
$$1_2 = 0 \times 2^0 = 1$$

$$1010_2 = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0 = 8 + 0 + 2 + 0 = 10$$

Binary	Decimal
0000	0
0001	1
0010	2
0011	3
0100	4
1101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Octal numbers

- Another popular number system is the octal number system. There are 8(2^3) combinations of 3-bit binary numbers.
- Therefore, sets of 3-bit binary numbers can be conveniently represented by octal numbers with base 8.
- These numbers are O, 1, 2, 3, 4, 5, 6 and 7. This also is a positional number system and has two parts, integer and fractional.
- For example, (1O62 0 4O3)$_8$ is an octal number and can be written as



$$(1062 \cdot 403)_8 = 1 \times 8^3 + 0 \times 8^2 + 6 \times 8^1 + 2 \times 8^0 + 4 \times 8^{-1} + 0 \times 8^{-2} + 3 \times 8^{-3}$$

$$= 512 + 482 + 4/8 + 3/512$$

$$= (562 \cdot 50586)_{10}$$

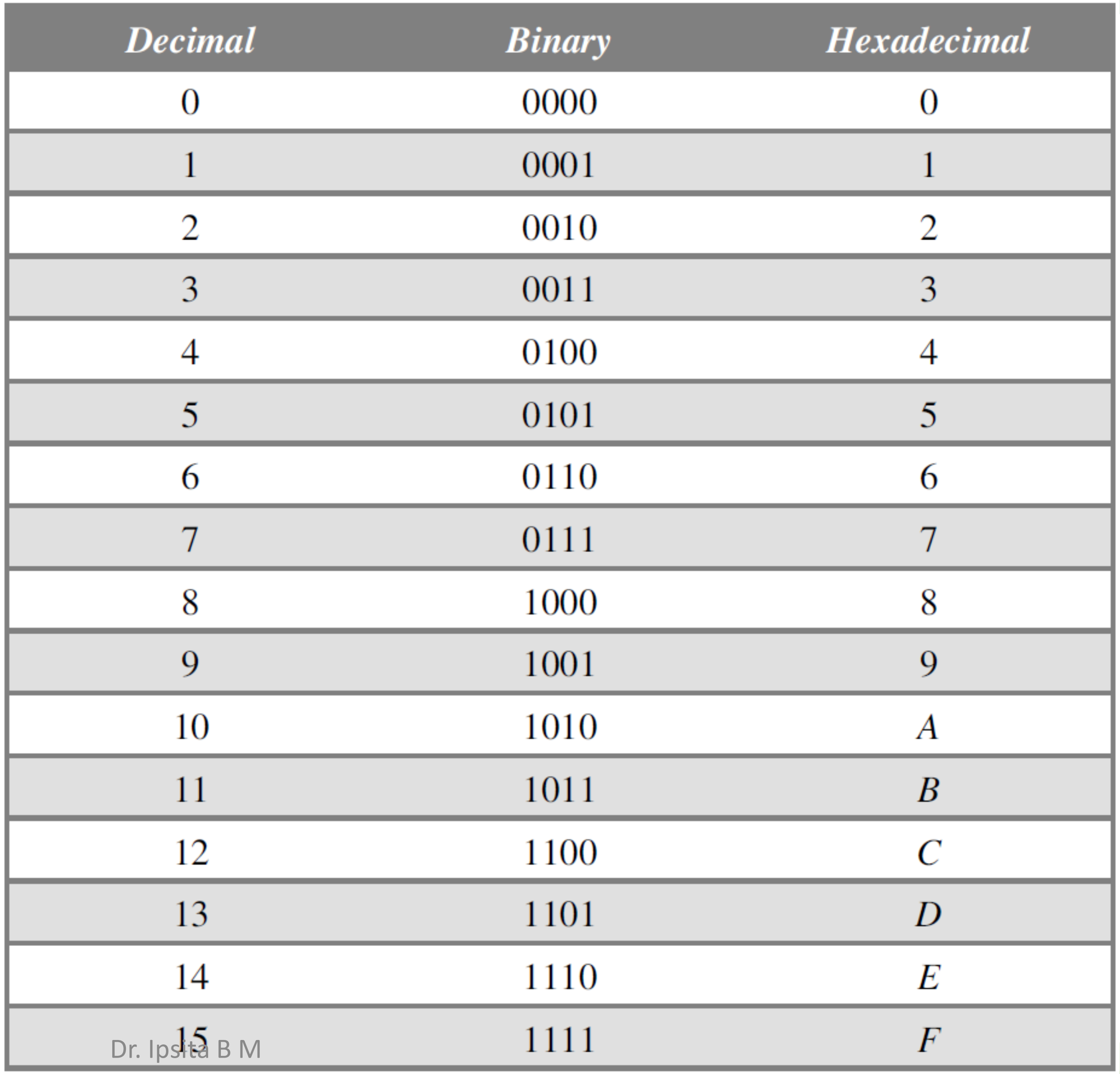
The above procedure gives the decimal equivalent of an octal number.

Hexadecimal number system

- Hexadecimal means 16.
- There are 16 combinations of 4-bit binary numbers and sets of 4-bit binary numbers can be entered in the microprocessor in the form of hexadecimal digits.
- The base (or radix) of a hexadecimal number is 16.
- This means that it uses 16 symbols to represent all numbers. These are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
- Since both numbers as well as alphabets are used to represent the digits in a hexadecimal number system, it is also called the alphanumeric number system.

Hexadecimal number system

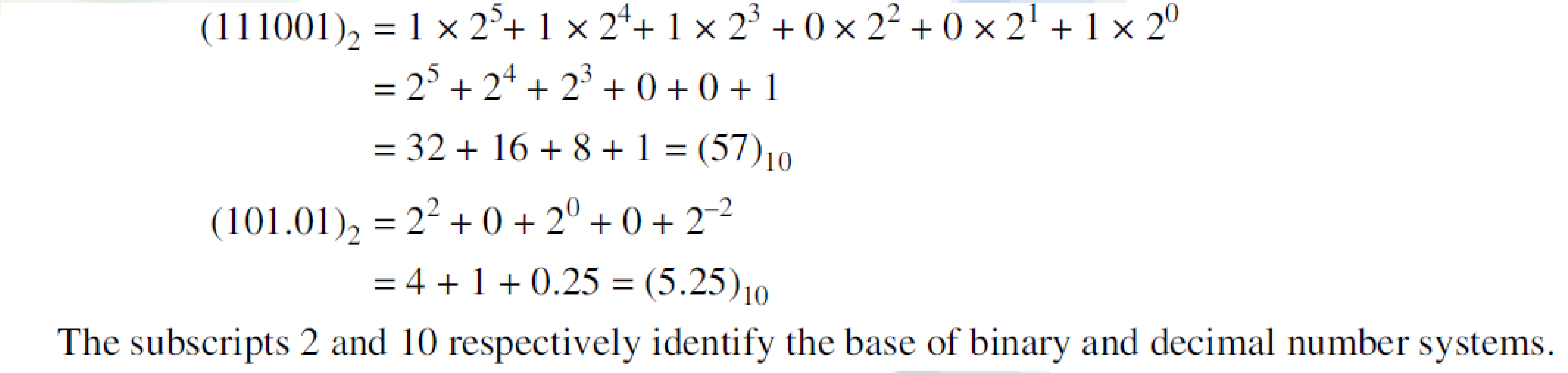
- Table shows the equivalences between hexadecimal, binary and decimal digits.



Decimal	*Binary*	*Hexadecimal*
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	*A*
11	1011	*B*
12	1100	*C*
13	1101	*D*
14	1110	*E*
15	1111	*F*

Binary to Decimal conversion

- To convert a binary number to its decimal equivalent, add the decimal equivalent of each position occupied by a 1. For example,



$$
\begin{aligned}
(111001)_2 &= 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
&= 2^5 + 2^4 + 2^3 + 0 + 0 + 1 \\
&= 32 + 16 + 8 + 1 = (57)_{10} \\
(101.01)_2 &= 2^2 + 0 + 2^0 + 0 + 2^{-2} \\
&= 4 + 1 + 0.25 = (5.25)_{10}
\end{aligned}
$$

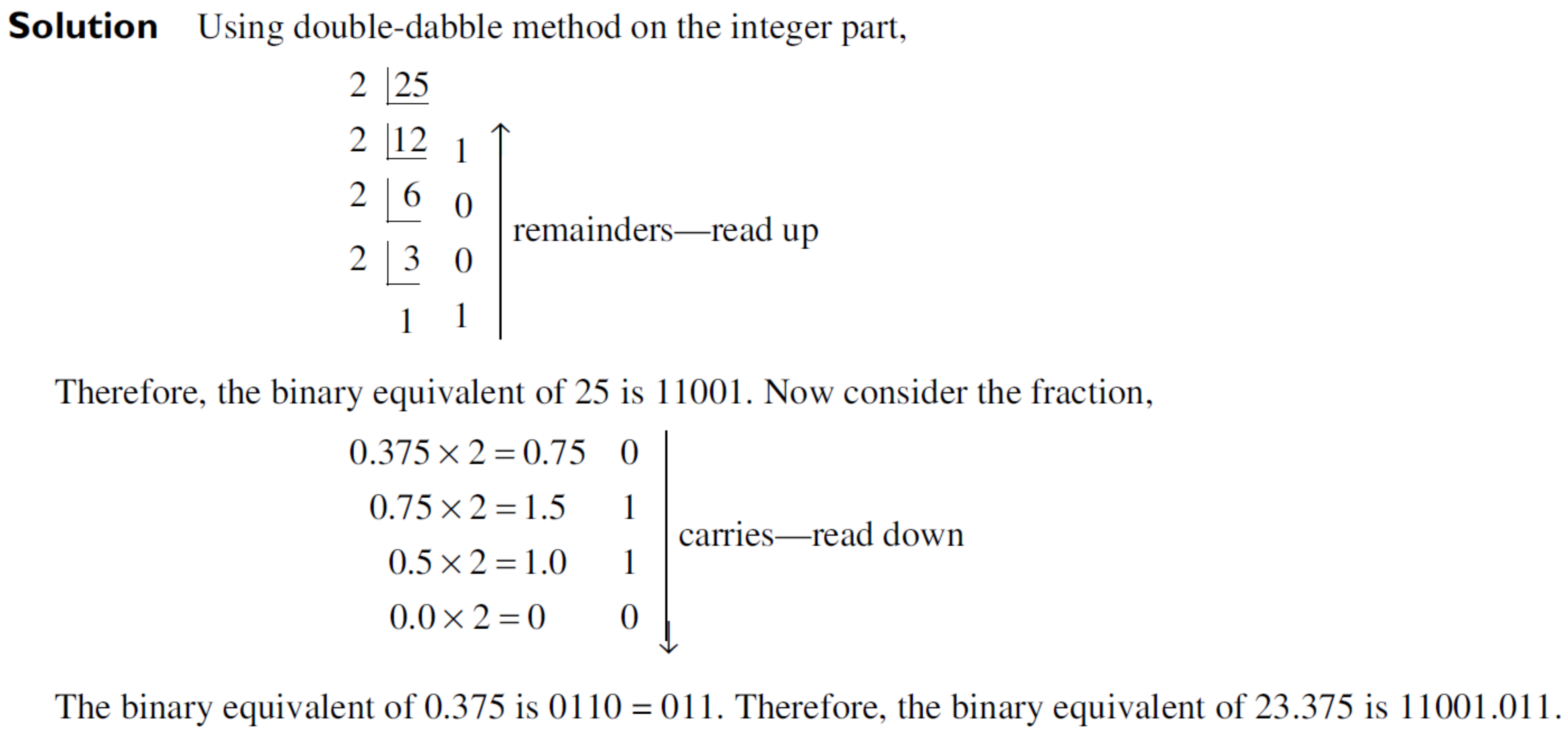
The subscripts 2 and 10 respectively identify the base of binary and decimal number systems.

Decimal to Binary conversion

- A decimal number can be converted to its equivalent binary form by the inverse process, i.e. by expressing the decimal number as the sum of powers of 2.
- The double-dabble is a very popular method for decimal-to- binary conversion, in which the integers and the decimals are handled separately.
- It can be summarised as follows:
    - To convert a decimal number to its binary equivalent, progressively divide the decimal number by 2, noting the remainders; the remainders taken in reverse order form the binary equivalent.
    - To convert a decimal fraction to its binary equivalent, progressively multiply the fraction by 2, removing and noting the carries; the carries taken in forward order form the binary equivalent.

Decimal to Binary conversion: Example

- Convert the decimal number 25.375 to its binary equivalent.



Solution Using double-dabble method on the integer part,

$$\begin{array}{r|l l} 2 & 25 & \\ 2 & 12 & 1 \\ 2 & 6 & 0 \\ 2 & 3 & 0 \\ & 1 & 1 \end{array} \Bigg\uparrow \text{remainders—read up}$$

Therefore, the binary equivalent of 25 is 11001. Now consider the fraction,

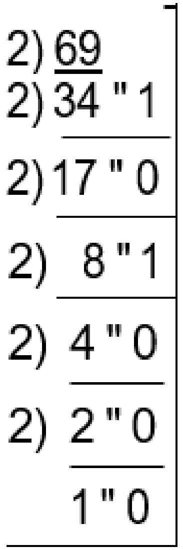
$$\begin{array}{r l} 0.375 \times 2 = 0.75 & 0 \\ 0.75 \times 2 = 1.5 & 1 \\ 0.5 \times 2 = 1.0 & 1 \\ 0.0 \times 2 = 0 & 0 \end{array} \Bigg\downarrow \text{carries—read down}$$

The binary equivalent of 0.375 is 0110 = 011. Therefore, the binary equivalent of 23.375 is 11001.011.

Decimal to Binary conversion: Example

Convert 69 to binary.

Solution



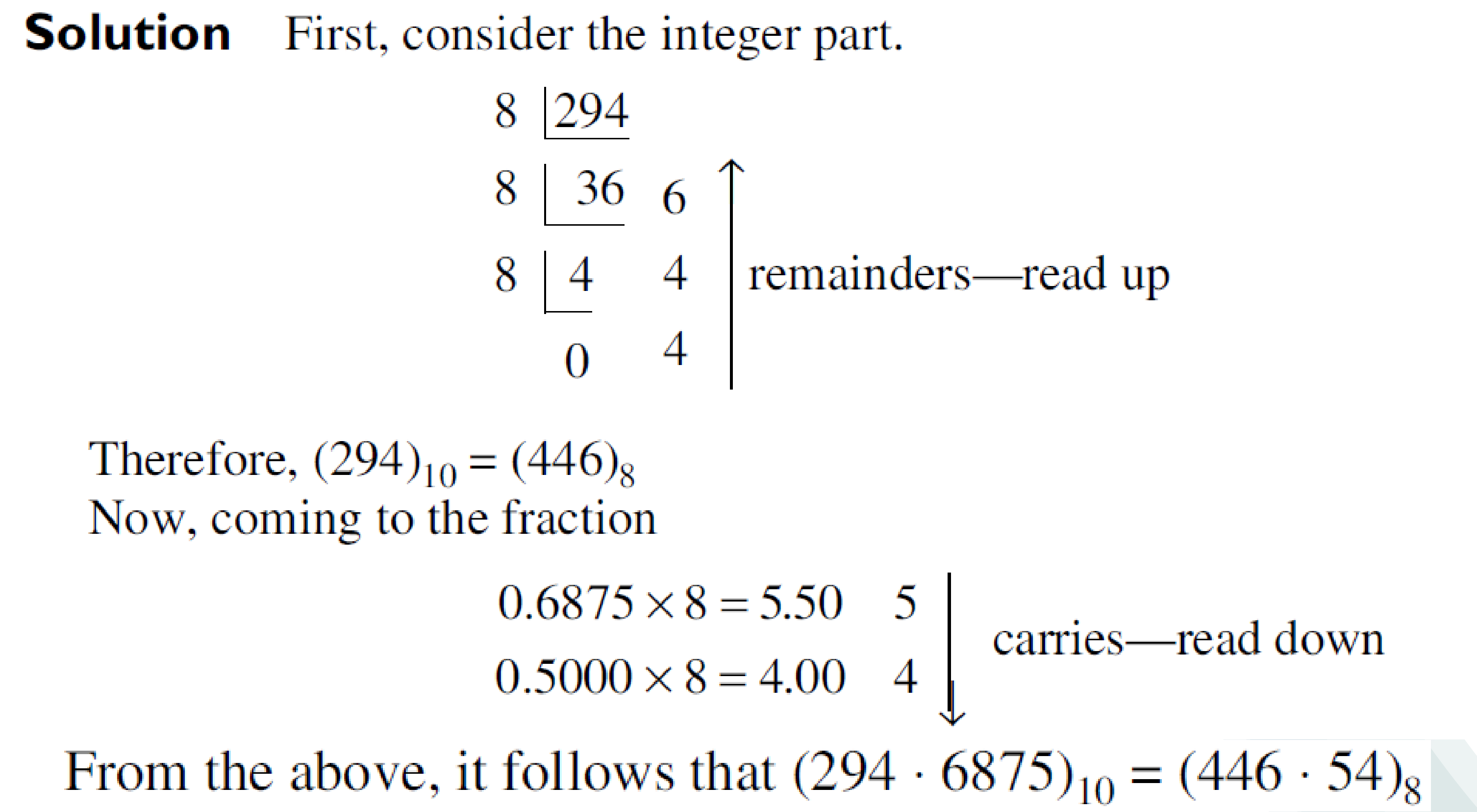
2) 69
2) 34 " 1
2) 17 " 0
2)   8 " 1
2) 4 " 0
2) 2 " 0
1 " 0

Read the remainders from bottom to top.

$69_{10} = 1000101_2$.

Decimal to Octal conversion

- Convert $(294.6875)_{10}$ into octal.



Solution First, consider the integer part.

$$\begin{array}{r|rc} 8 & 294 & \\ 8 & 36 & 6 \\ 8 & 4 & 4 \\ & 0 & 4 \end{array} \Bigg\uparrow \text{remainders—read up}$$

Therefore, $(294)_{10} = (446)_8$

Now, coming to the fraction

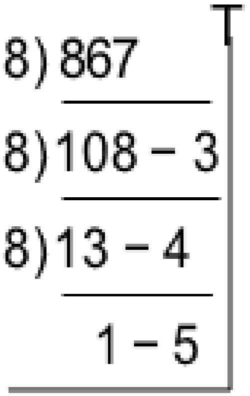
$$\left.\begin{array}{ll} 0.6875 \times 8 = 5.50 & 5 \\ 0.5000 \times 8 = 4.00 & 4 \end{array}\right\downarrow \text{carries—read down}$$

From the above, it follows that $(294 \cdot 6875)_{10} = (446 \cdot 54)_8$

Example

Convert 867_{10} to octal number. It is simply a successive division by 8.

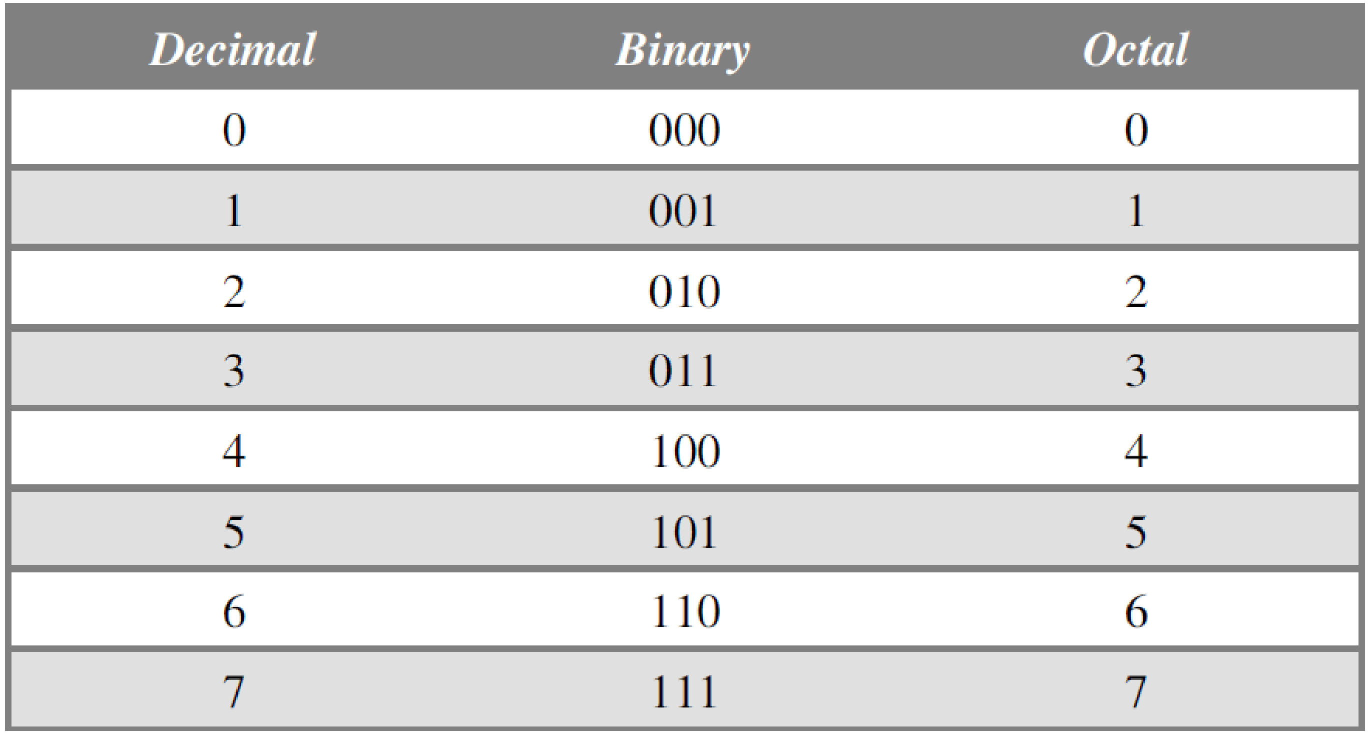
Solution



The result is $(1543)_8$

Octal to Binary & Binary to Octal conversion

- Octal numbers can be converted to equivalent binary numbers by replacing each digit by its 3-bit binary equivalent.
- Below table gives octal and binary equivalents for decimal numbers 0-7.



Decimal	*Binary*	*Octal*
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

Octal to Binary & Binary to Octal conversion

$$(642 \cdot 71)_8 = (110100010 \cdot 111001)_2$$

- Similarly, binary numbers can be converted into equivalent octal numbers by making groups of 3 bits starting from LSB and moving towards MSB for integer part. For example,

$$(101110011)_2 = 101110011 = (563)_8$$

- For fractional parts, we start grouping from the bit next to the binary point and move towards right. For example,

$$(0.101010110)_2 = 0.101010110 = (0.526)_8$$

- In forming the 3-bit groupings, sometimes, we may have to add 0's to complete the most significant digit group in the integer part and the least significant digit group in the fractional part.

Converting Hex to Decimal

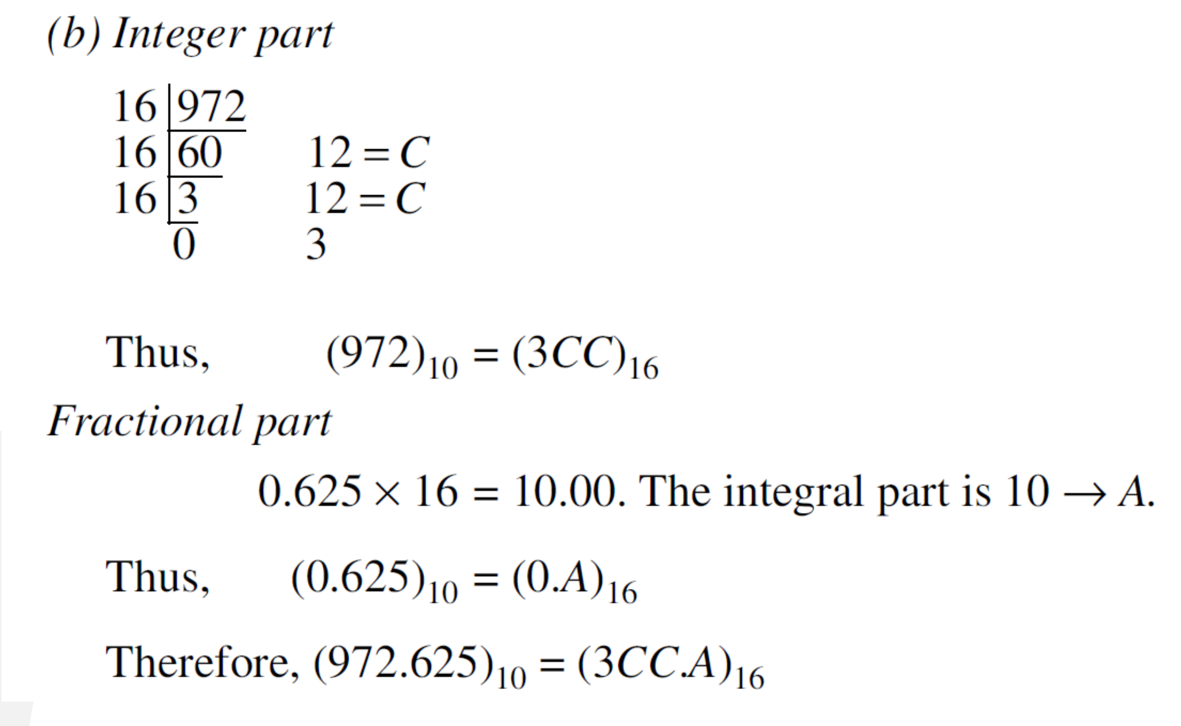
- Since 16 digits are used, the weights are in powers of 16.
- The decimal equivalent of a hexadecimal string equals the sum of all hexadecimal digits multiplied by their weights.

$$\begin{aligned}(F8E.28)_{16} &= F \times 16^{2} + 8 \times 16^{1} + E \times 16^{0} + 2 \times 16^{-1} + B \times 16^{-2} \\ &= 15 \times 16^{2} + 8 \times 16^{1} + 14 \times 16^{0} + 2 \times 16^{-1} + 11 \times 16^{-2} \\ &= 3840 + 128 + 14 + 2/16 + 11/256 \\ &= (3982.16796875)_{10}\end{aligned}$$

Decimal to Hex conversion

- Convert the following numbers to their hexadecimal equivalents.

  (a) $(49.5)_{10}$ (b) $(972.625)_{10}$



(b) Integer part

$$\begin{array}{r|l l} 16 & 972 & \\ 16 & 60 & 12 = C \\ 16 & 3 & 12 = C \\ & 0 & 3 \end{array}$$

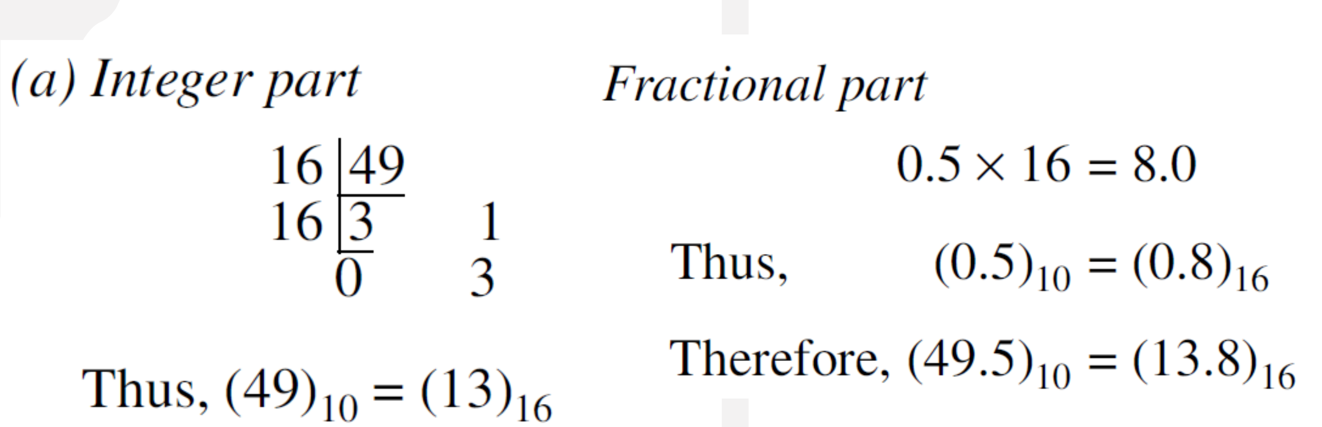
Thus, $(972)_{10} = (3CC)_{16}$

Fractional part

$0.625 \times 16 = 10.00$. The integral part is $10 \rightarrow A$.

Thus, $(0.625)_{10} = (0.A)_{16}$

Therefore, $(972.625)_{10} = (3CC.A)_{16}$



(a) Integer part

$$\begin{array}{r|l l} 16 & 49 & \\ 16 & 3 & 1 \\ & 0 & 3 \end{array}$$

Thus, $(49)_{10} = (13)_{16}$

Fractional part

$0.5 \times 16 = 8.0$

Thus, $(0.5)_{10} = (0.8)_{16}$

Therefore, $(49.5)_{10} = (13.8)_{16}$

Hex to Binary & Binary to Hex conversion

- Hexadecimal numbers can be converted into equivalent binary by replacing each hexadecimal digit by its equivalent 4-bit binary number. For example,

$$(20E.CA)_{16} = (0010\ 0000\ 1110.1100\ 1010)_2$$

$$= (001000001110.11001010)_2$$

- Similarly, binary numbers can be converted into hexadecimal numbers by making groups of four bits starting from LSB and moving towards MSB, for integers, and then replacing each group of four bits by its hexadecimal equivalents. Sometimes, in forming 4-bit groupings, 0's may be required to complete the most significant digit group in the integer part. For example,

$$(10100110111110)_2 = (0010\ 1001\ 1011\ 1110)_2$$

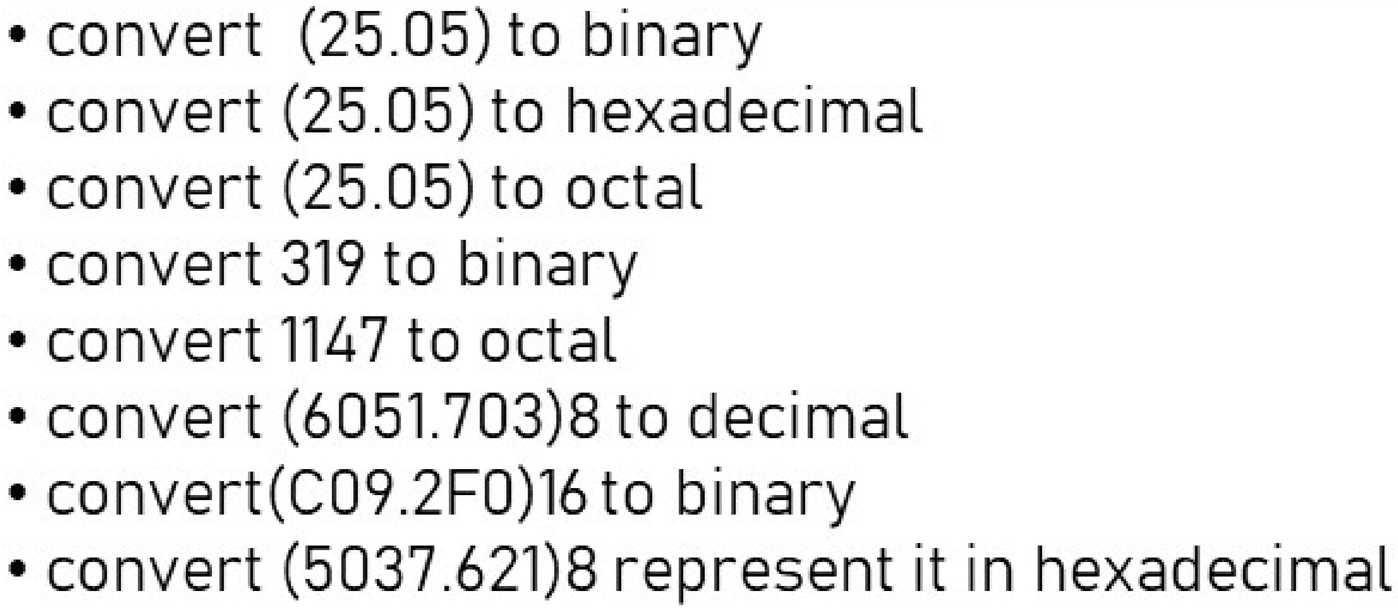
$$= (29BE)_{16}$$

- For the fractional part, the above procedure is repeated starting from the bit next to the hexadecimal point and moving towards the right. Here again, in forming 4-bit groupings, O's may be required to complete the least significant digit group. For example,

$$(0.00111110111101)_2 = (0.0011\ 1110\ 1111\ 0100)_2$$

$$= (0.3EF4)_{16}$$

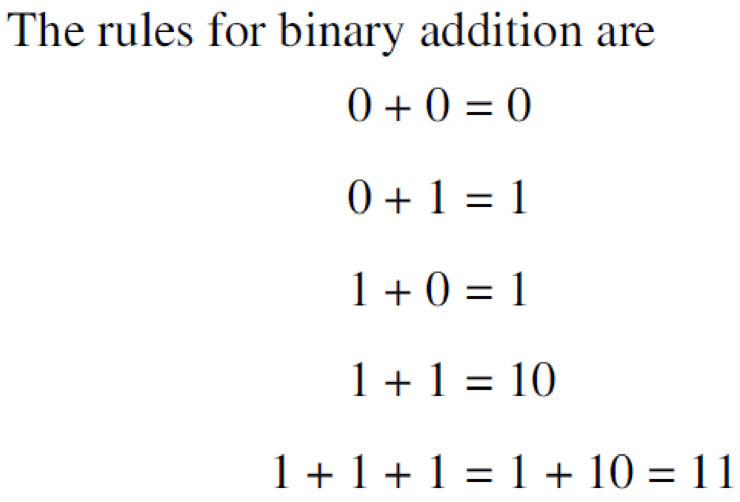
Practice problems



- convert (25.05) to binary
- convert (25.05) to hexadecimal
- convert (25.05) to octal
- convert 319 to binary
- convert 1147 to octal
- convert (6051.703)8 to decimal
- convert(C09.2F0)16 to binary
- convert (5037.621)8 represent it in hexadecimal

Binary arithmetic

- **Binary addition:**



The rules for binary addition are

$$0 + 0 = 0$$

$$0 + 1 = 1$$

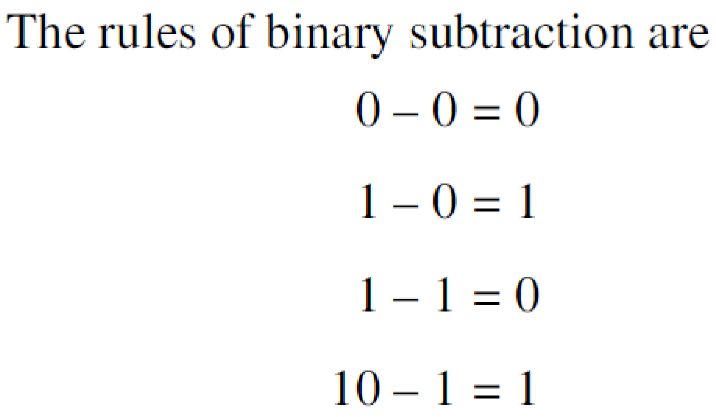
$$1 + 0 = 1$$

$$1 + 1 = 10$$

$$1 + 1 + 1 = 1 + 10 = 11$$

  - It is important to note that the sign "+" used here corresponds to arithmetic addition and not logical operation.
  - For large numbers, we add column by column, carrying where necessary into higher position columns.

- **Binary Subtraction:**



The rules of binary subtraction are

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$10 - 1 = 1$$

  - In subtraction, we subtract column by column, borrowing wherever necessary from higher position columns.
  - In subtracting a large number from a smaller one, we can subtract the smaller from the larger and change the sign just as we do with decimals.

Binary addition: Example

Add 101111_2 and 10111_2.

Solution

Number 1	1 0 1 1 1 1
Number 2	1 0 1 1 1
	1000110_2

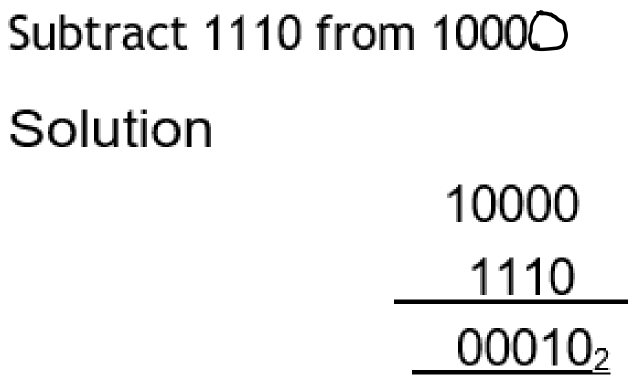
During the addition, 1 + 1 = 10 and 1 + 1 + carry (1) = 11, and so on.

Add 1111 and 1111.

Solution

Number 1:	1 1 1 1
Number 2:	1 1 1 1
Carry:	1 1 1
	11110_2

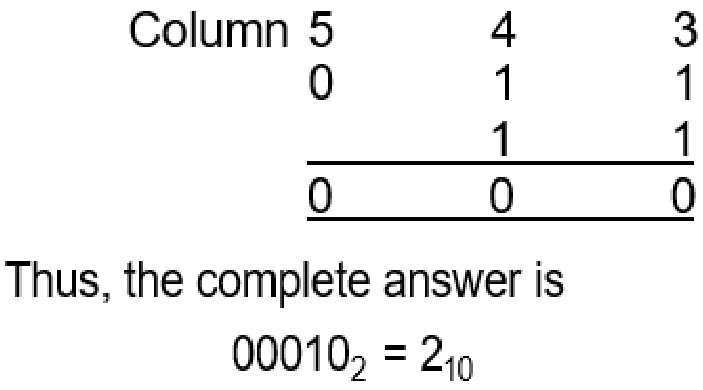
Binary subtraction



Subtract 1110 from 10000

Solution

$$\begin{array}{r} 10000 \\ 1110 \\ \hline 00010_2 \\ \hline \end{array}$$



Column 5	4	3
0	1	1
	1	1
0	0	0

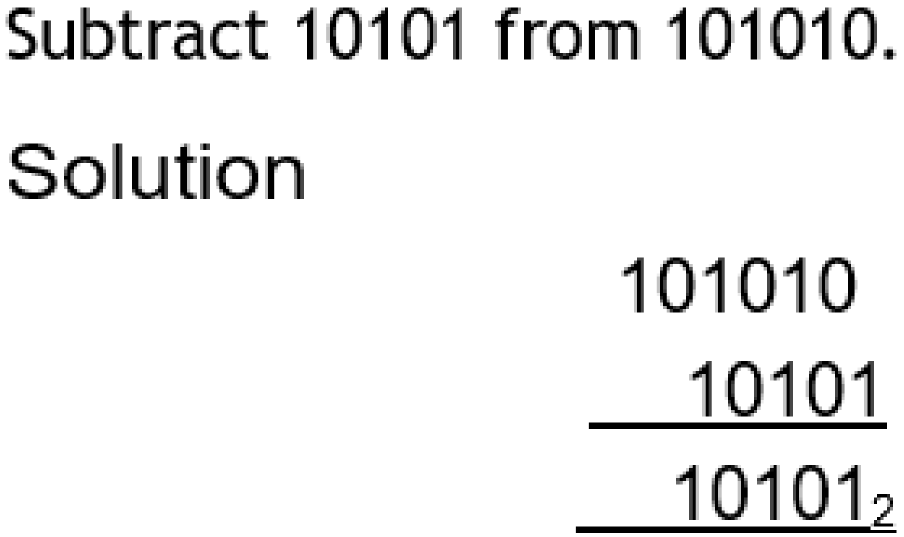
Thus, the complete answer is

$00010_2 = 2_{10}$

$$\begin{array}{r} 1+\ \ 1+\ \ 1+\ \ \ \\ \cancel{1}\ \ 0\ \ 0\ \ 0\ \ 0 \\ 1\ \ 1\ \ \cancel{1}\ \ 0 \\ \hline (0\ \ 0\ \ 0\ \ 1\ \ 0)_2 \end{array}$$

- We find that in the second column 1 cannot be subtracted from 0. So, a 1 must be borrowed from the third column but it is a 0.
- In this example, 1 is available in the fifth column. So, borrow this 1, leaving behind a 0.
- Then 1 is (1 + 1) in the fourth column. We borrow 1 leaving behind 1 in the fourth column.
- Finally, successive borrowing makes (1 + 1) in the second column from which we subtract 1, yielding 1 as answer in the second column.
- At this stage, we have the answers for zeroth and first column. The third, fourth and fifth columns are shown.

Binary subtraction: Example



Subtract 10101 from 101010.

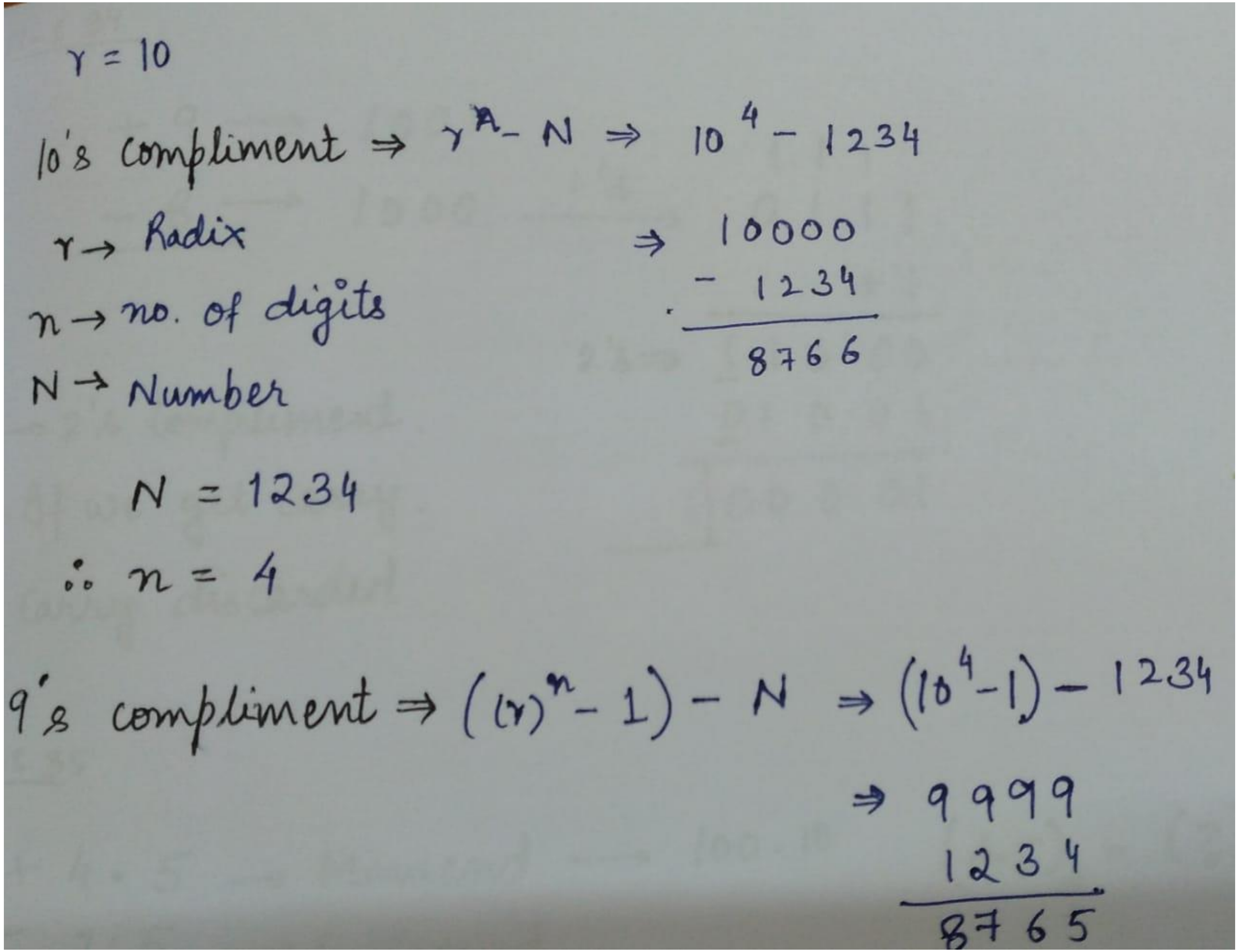
Solution

$$\begin{array}{r} 101010 \\ \underline{10101} \\ \underline{10101_2} \end{array}$$

Complement of a number – General representation

- Let's consider "r" number system, where r is the radix/base of the number system. So we will have two types of complement. They are:
  - r's complement
  - (r-1)'s complement
- When r=2 (binary NS): its complements are 2's & 1's complement.
- When r=10 (decimal NS): its complements are 10's & 9's complement.
- When r=8 (octal NS): its complements are 8's and 7's complement.
- When r=16 (hexadecimal NS): its complements are 16's and 15's complement.
- Rule: ***For any number system with radix as "r",***
  - ***r's complement is given by r^n-N***
  - ***(r-1)'s complement is given by ((r^n)-1)-N where,***
    - r – radix, n – no. of the digits of the given no. and N – given number

Complement of Decimal numbers (1234)



$r = 10$

10's compliment ⇒ $r^n - N$ ⇒ $10^4 - 1234$

$$\Rightarrow \begin{array}{r} 10000 \\ -\ 1234 \\ \hline 8766 \end{array}$$

r → Radix

n → no. of digits

N → Number

$N = 1234$

$\therefore n = 4$

9's compliment ⇒ $((r)^n - 1) - N$ ⇒ $(10^4 - 1) - 1234$

$$\Rightarrow \begin{array}{r} 9999 \\ 1234 \\ \hline 8765 \end{array}$$

Complement of Binary numbers

- The computer arithmetic process is done not only with positive numbers; it is also done with negative numbers. In such a condition, signed and unsigned numbers are dealt by the processors. The 1's and 2's complement operation is useful for this type of arithmetic process.
- 1's complement of a binary number is just an inversion of individual bits. For example, the 1's complement of 1010101 is found by inversion of each bit, i.e. 0 to 1 and 1 to zero conversion.
  - 1's complement of $(1010101)_2$ is

    $(0101010)_2$
- The 2's complement of a positive binary number whose integral part has p digits is also found by subtracting the number from 2^p.
- The 2's complement of a given binary number is also found by adding 1 to the least significant bit of the 1's complement of that binary number.

Complement of Binary numbers

- For instance, 2's complement of 1100 is
  - 1st method: 2^4 – 1100 = 10000 – 1100 = 0100
  - 2nd method: 1’s complement of 1100 = 0011; 0011 + 1 = 0100

- Find the 2's complement of the number $(10101011)_2$?

Complement of Binary numbers

- The 2's complement of the number (10101011)2 is found by two steps. First, convert the given number into its 1's complement by inversion.
  - 1010101 → 0101010 is the 1's complement of the number. Then add one to the least significant bit (LSB). So

0101010

1

0101011_2 – 2's complement of the number is $(1010101)_2$. 2's complement is used to represent negative numbers.

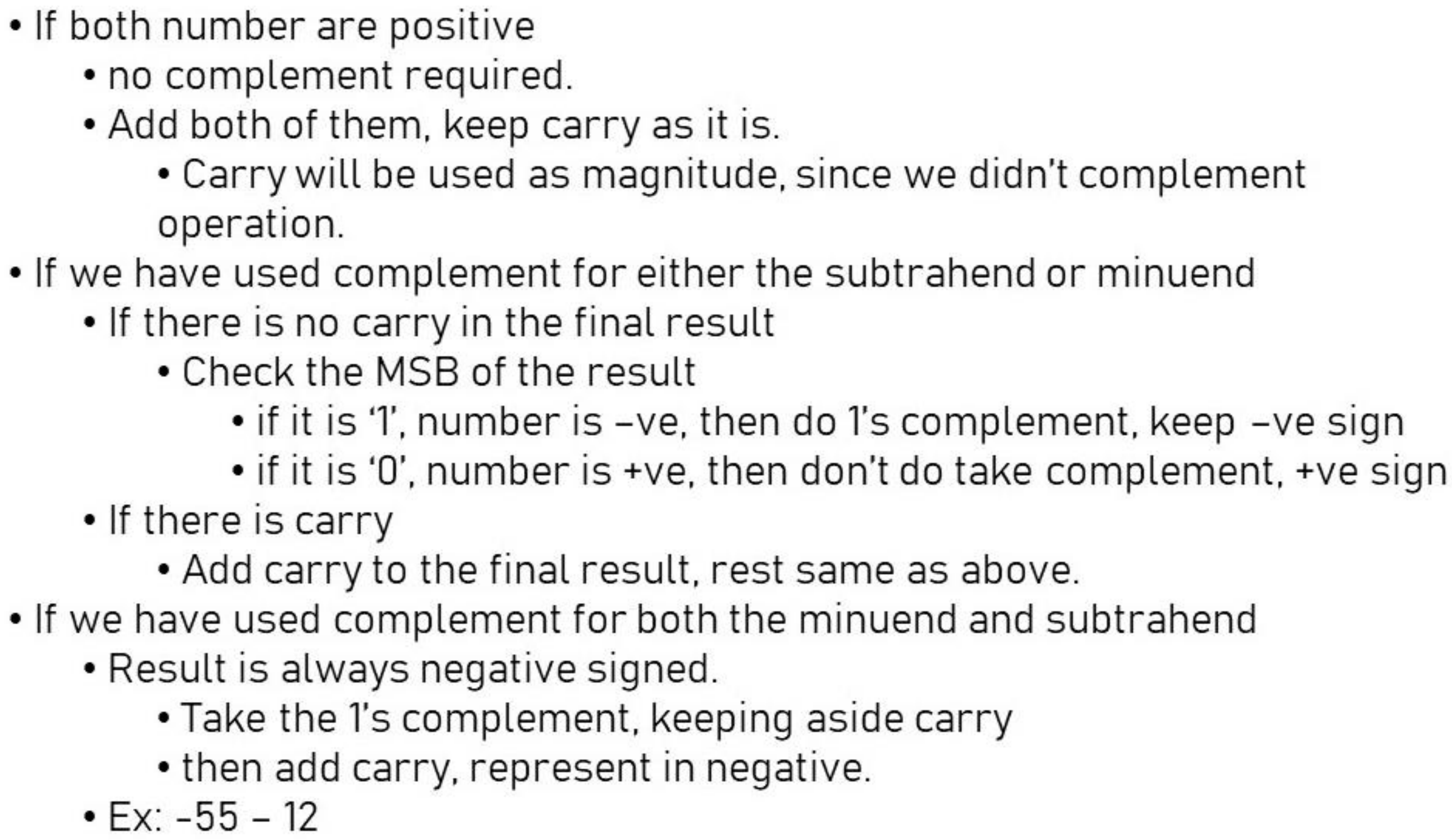
Complement of Binary numbers

- The previous procedure also applies for a decimal binary number. For example,

  - To find 2's complement of 0.1110100
    1's complement 0.0001011
    +1
    2's complement 0.0001100

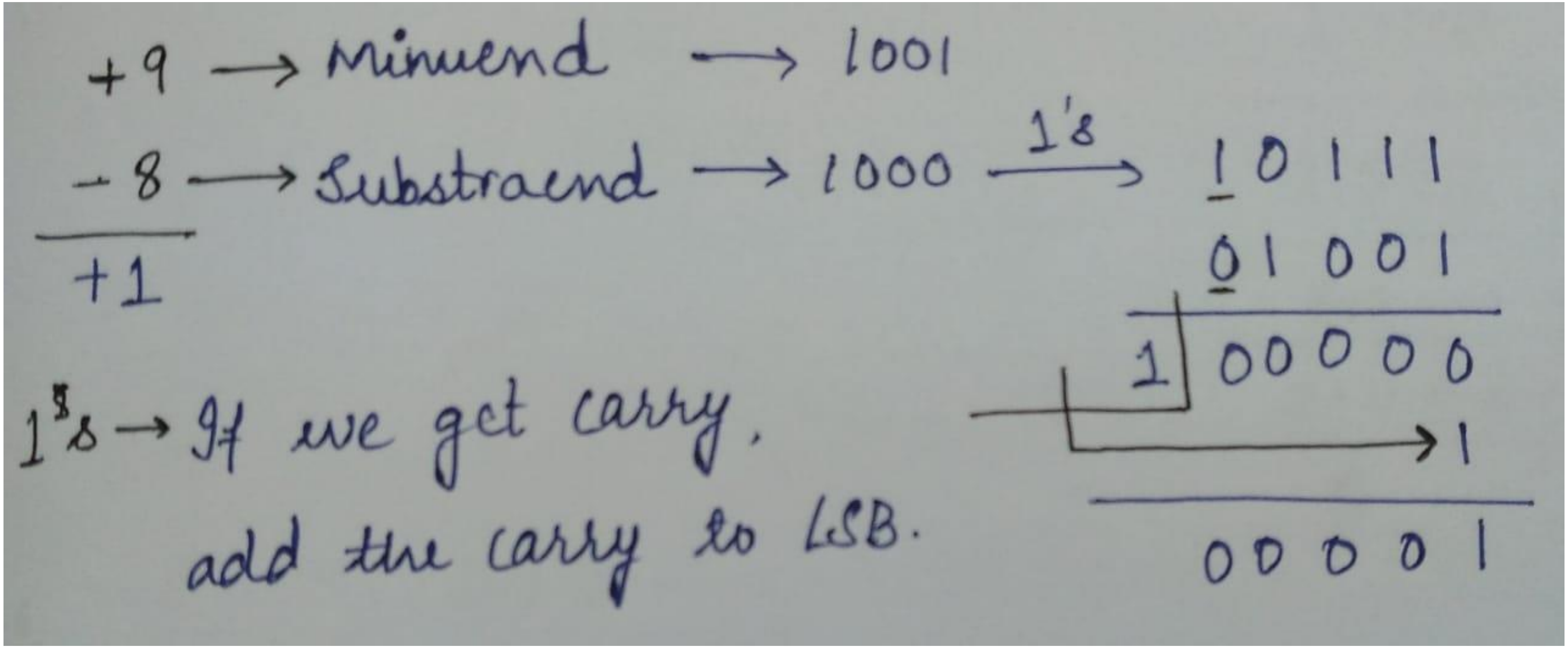
  - To find 2's complement of 1011.001
    1's complement 0100.110
    +1
    2's complement 0100.111

1’s complement subtraction



- If both number are positive
  - no complement required.
  - Add both of them, keep carry as it is.
    - Carry will be used as magnitude, since we didn’t complement operation.
- If we have used complement for either the subtrahend or minuend
  - If there is no carry in the final result
    - Check the MSB of the result
      - if it is ‘1’, number is –ve, then do 1’s complement, keep –ve sign
      - if it is ‘0’, number is +ve, then don’t do take complement, +ve sign
  - If there is carry
    - Add carry to the final result, rest same as above.
- If we have used complement for both the minuend and subtrahend
  - Result is always negative signed.
    - Take the 1’s complement, keeping aside carry
    - then add carry, represent in negative.
  - Ex: -55 – 12

1's complement subtraction



+9 → Minuend → 1001

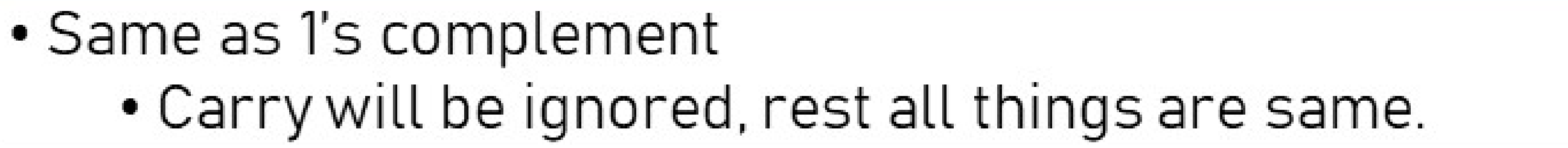
−8 → Substraend → 1000 $\xrightarrow{1's}$ 10111

+1

$$
\begin{array}{r}
10111 \\
01001 \\
\hline
1\,|\,00000 \\
\rightarrow 1 \\
\hline
00001
\end{array}
$$

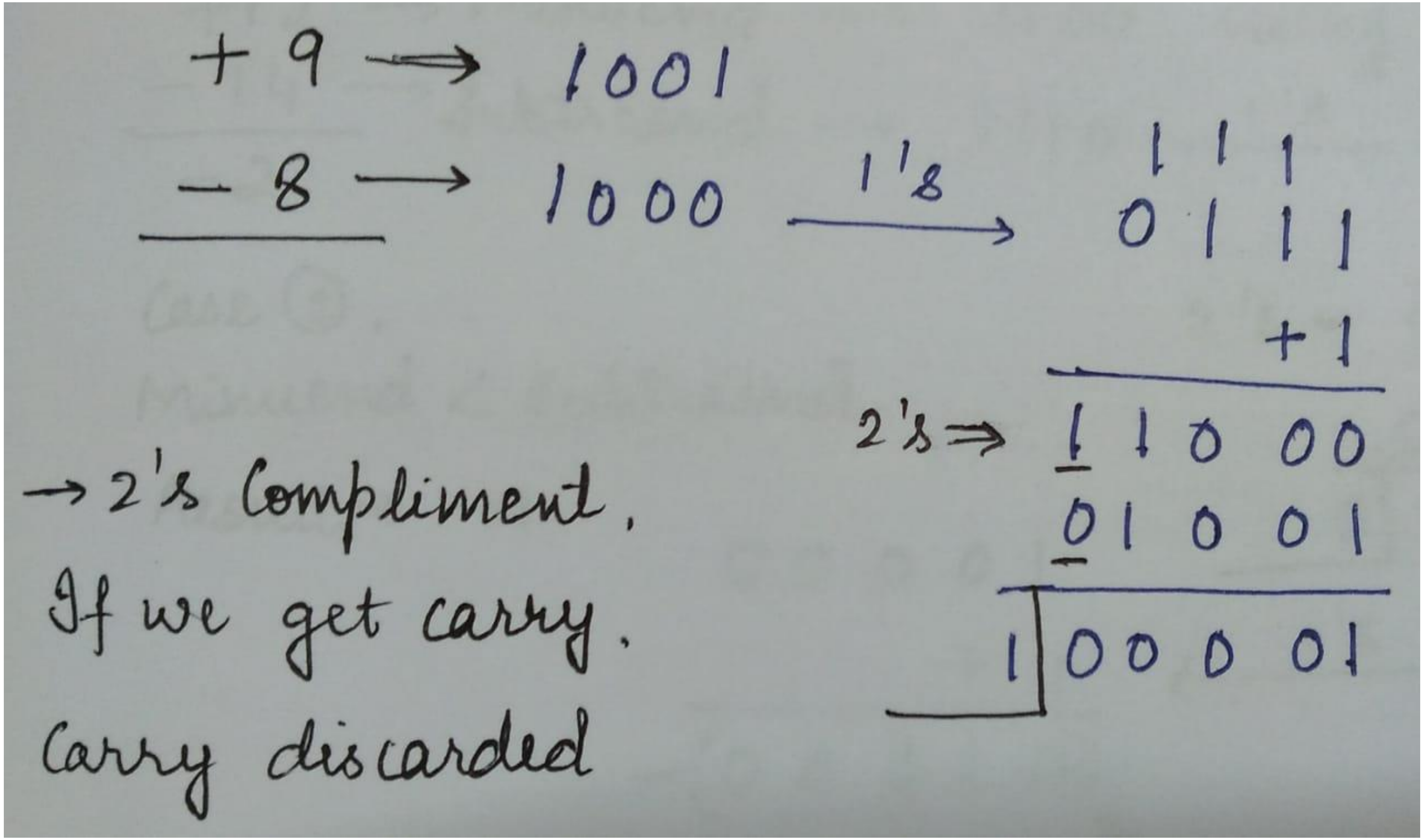
1's → If we get carry, add the carry to LSB.

2’s complement subtraction



- Same as 1’s complement
    - Carry will be ignored, rest all things are same.

2's complement subtraction



+9 ⟹ 1001

−8 → 1000 $\xrightarrow{1's}$ 0111

1 1 1

0 1 1 1

+1

2's ⟹ 1 1 0 0 0

0 1 0 0 1

1 | 0 0 0 0 1

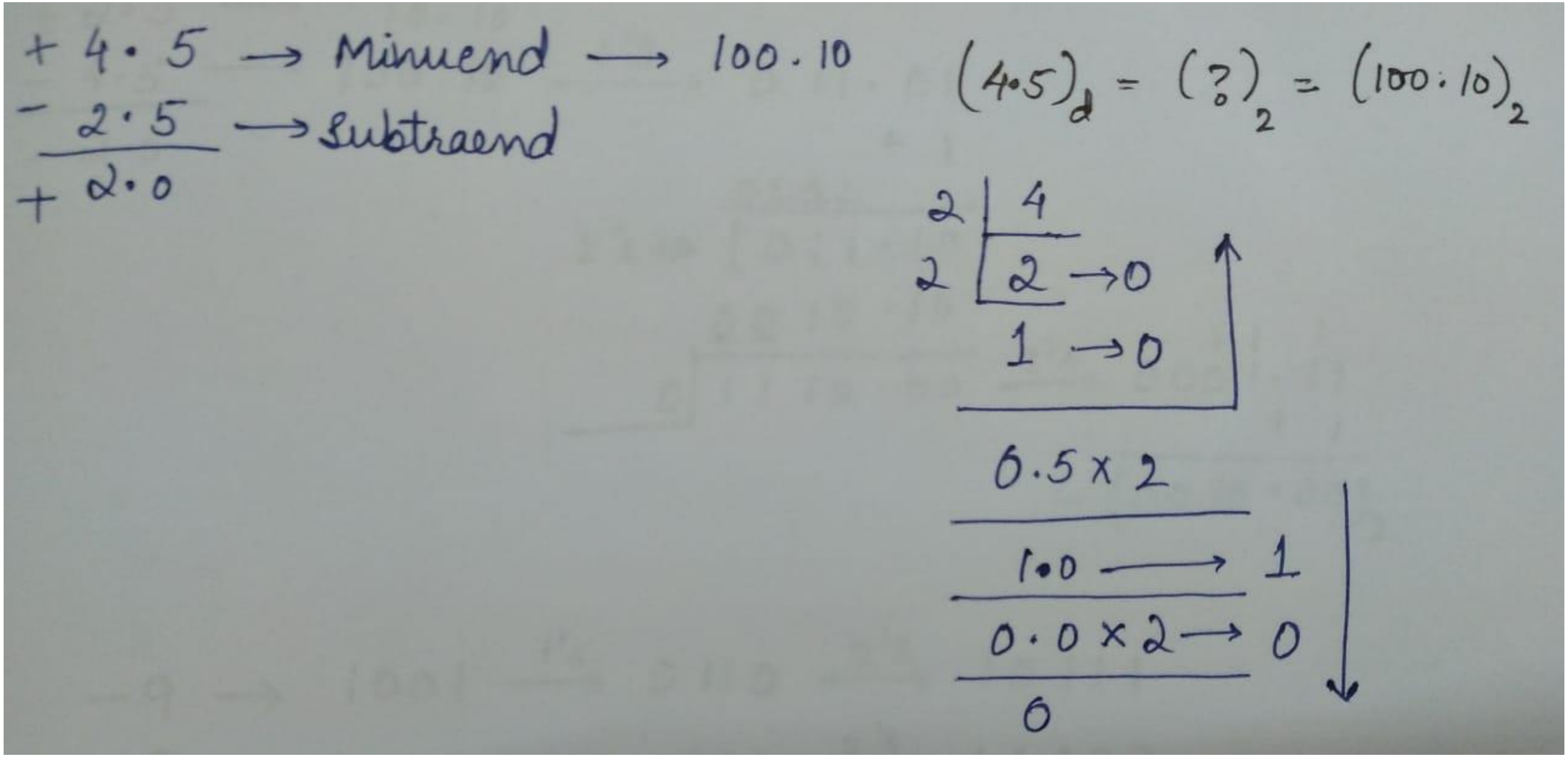
→ 2's Compliment.

If we get carry.

Carry discarded

2’s complement subtraction

- Case 1: Minuend > Subtrahend



$+4.5 \rightarrow$ Minuend $\longrightarrow 100.10$

$-\,2.5 \rightarrow$ Subtraend

$+\,2.0$

$(4.5)_d = (?)_2 = (100.10)_2$

$$\begin{array}{r|l} 2 & 4 \\ \hline 2 & 2 \rightarrow 0 \\ \hline & 1 \rightarrow 0 \end{array}$$

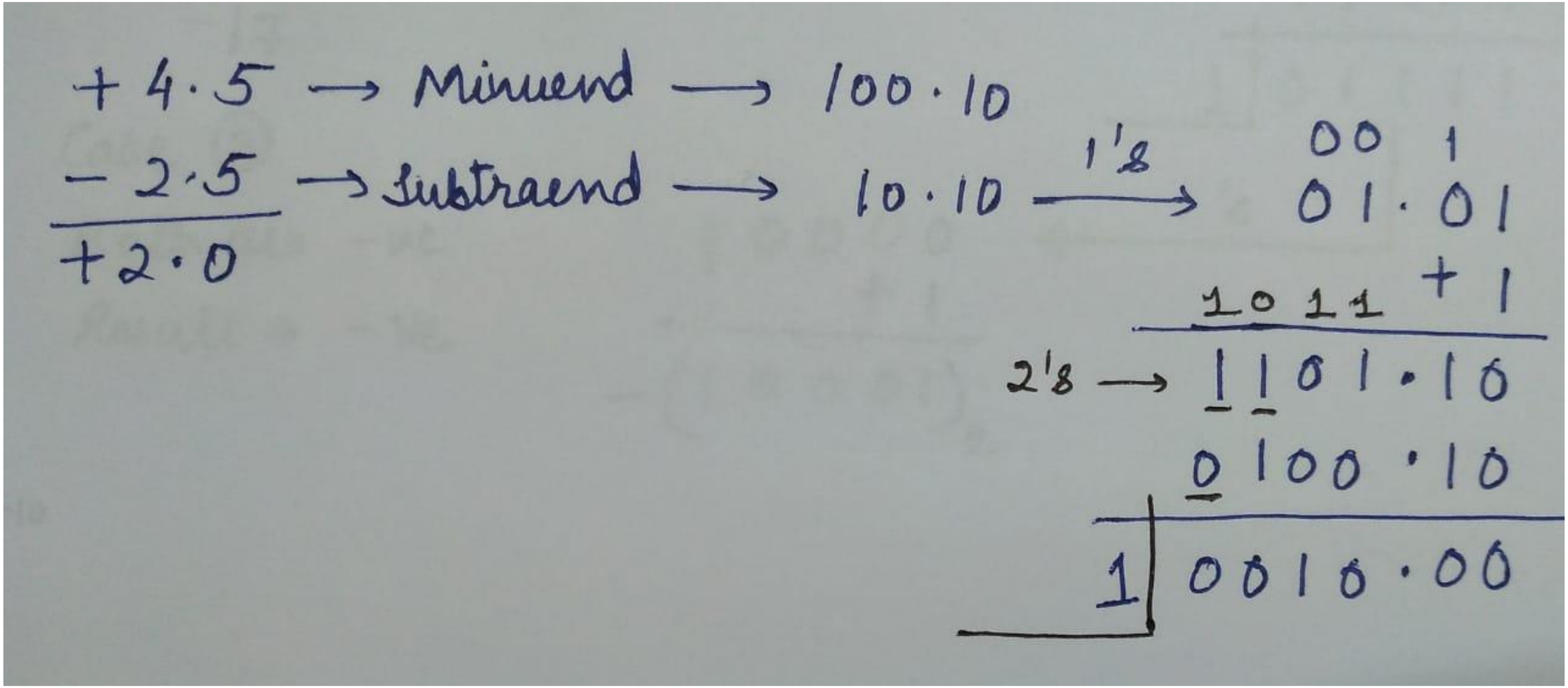
0.5×2

$1.0 \longrightarrow 1$

$0.0 \times 2 \longrightarrow 0$

0

2's complement subtraction



+4.5 → Minuend → 100.10

−2.5 → Subtraend → 10.10 $\xrightarrow{1's}$ 01.01

+2.0

00 1

+1

1 0 1 1

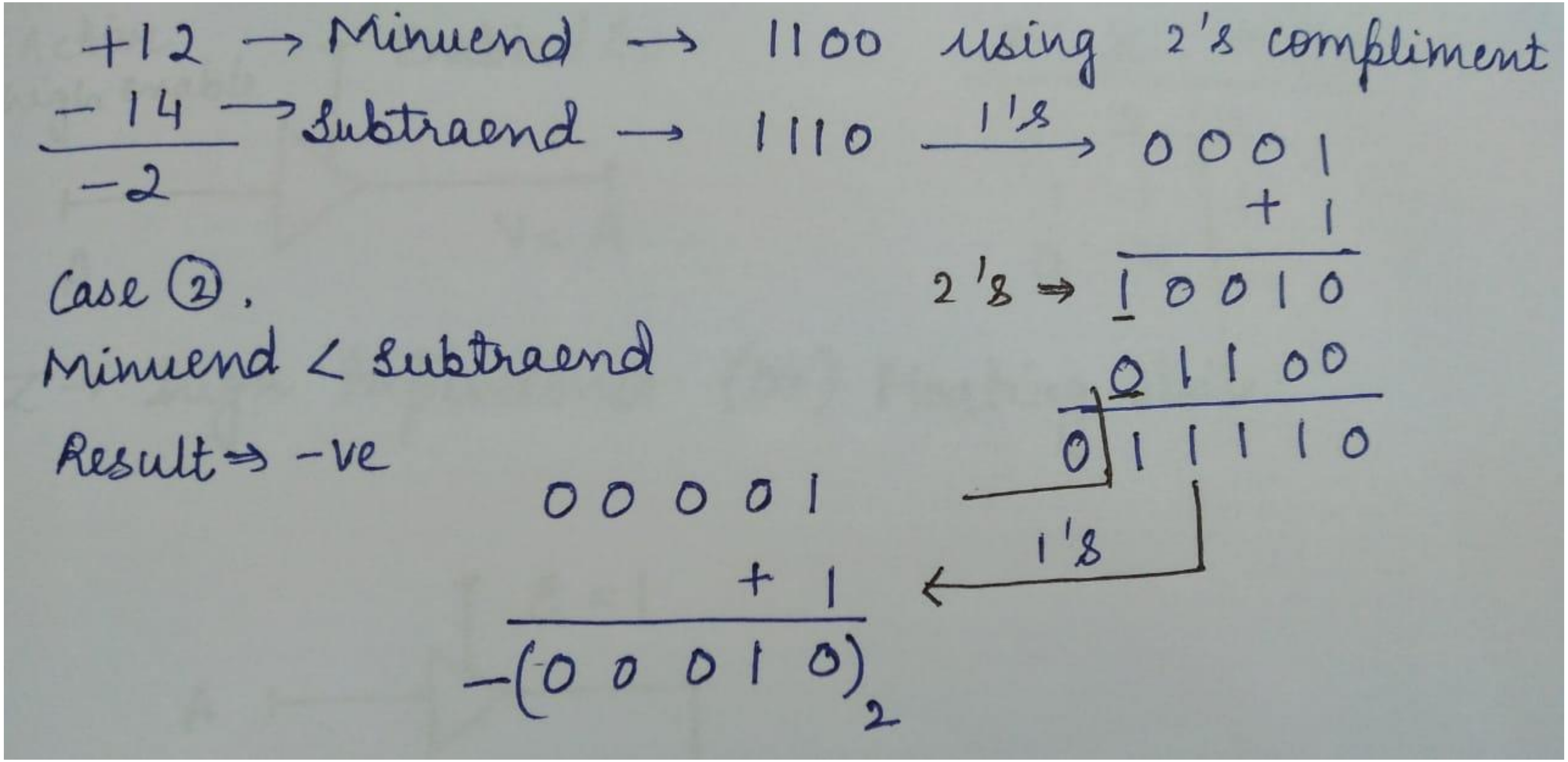
2's → 1101.10

0100.10

1 | 0010.00

2’s complement subtraction

- Case 2: Minuend < Subtrahend



+12 → Minuend → 1100 using 2's compliment
−14 → Subtraend → 1110 $\xrightarrow{1's}$ 0001
−2

$$\begin{array}{r} 0001 \\ +\;1 \\ \hline \underline{1}0010 \end{array}$$

2's ⇒ $\underline{1}0010$

$$\begin{array}{r} \underline{1}0010 \\ \underline{0}1100 \\ \hline 0|11110 \end{array}$$

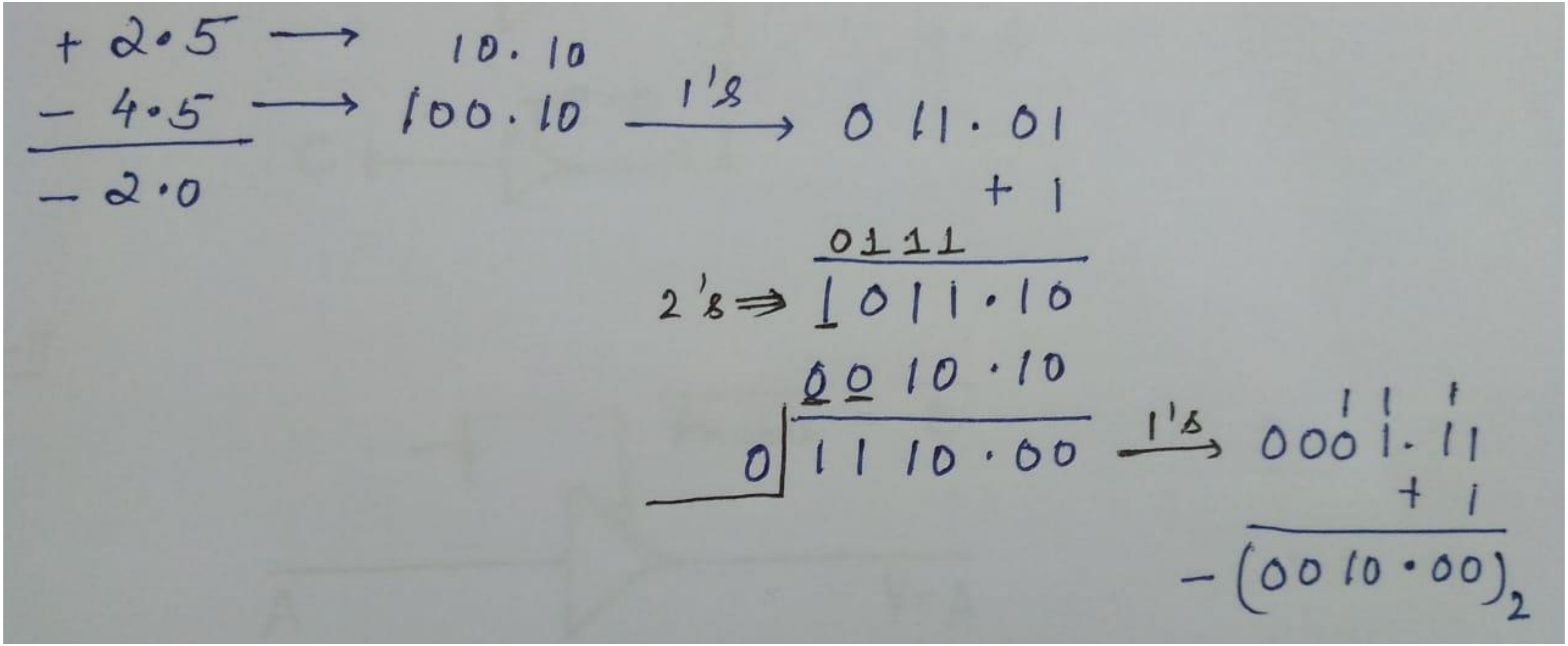
Case ②.
Minuend < Subtraend
Result ⇒ −ve

1's

$$\begin{array}{r} 00001 \\ +\;1 \\ \hline -(00010)_2 \end{array}$$

2’s complement subtraction

- 2.5 – 4.5 using 2’s complement?



$+2\cdot5 \longrightarrow 10.10$

$-4\cdot5 \longrightarrow 100.10 \xrightarrow{1's} 011.01$

$-2\cdot0$

$+1$

0111

$2's \Rightarrow 1011.10$

0010.10

$0 \; | \; 1110.00 \xrightarrow{1's} 0001.11$

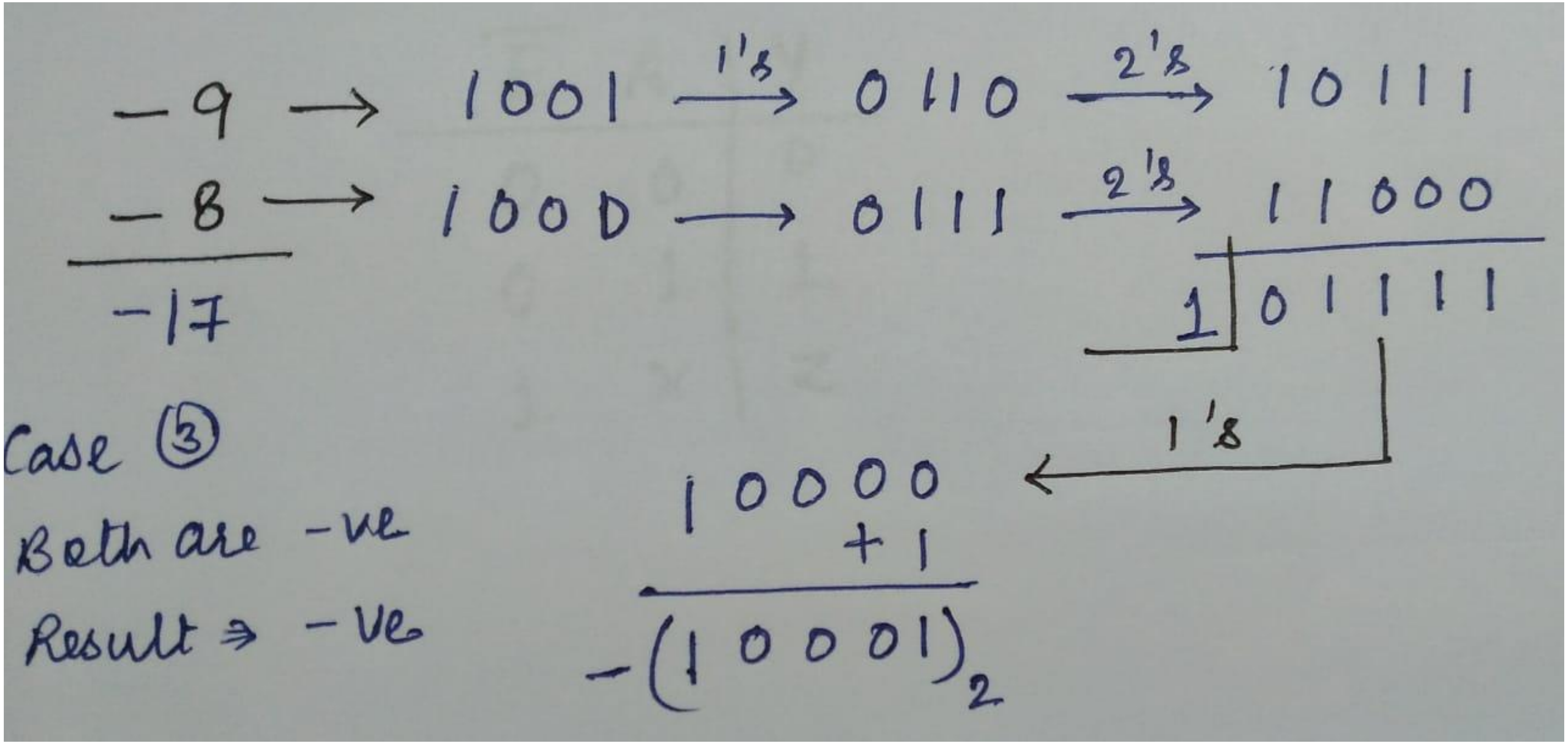
$1\;1\;\;1$

$+1$

$-(0010.00)_2$

2’s complement subtraction

- Case 3: Both Minuend & Subtrahend are negative



$$-9 \rightarrow 1001 \xrightarrow{1's} 0110 \xrightarrow{2's} 10111$$

$$-8 \rightarrow 1000 \longrightarrow 0111 \xrightarrow{2's} 11000$$

$$-17$$

$$\boxed{1}\,011111$$

$$\xrightarrow{1's}$$

$$\begin{array}{r} 10000 \\ +1 \\ \hline -(10001)_2 \end{array}$$

Case ③

Both are -ve

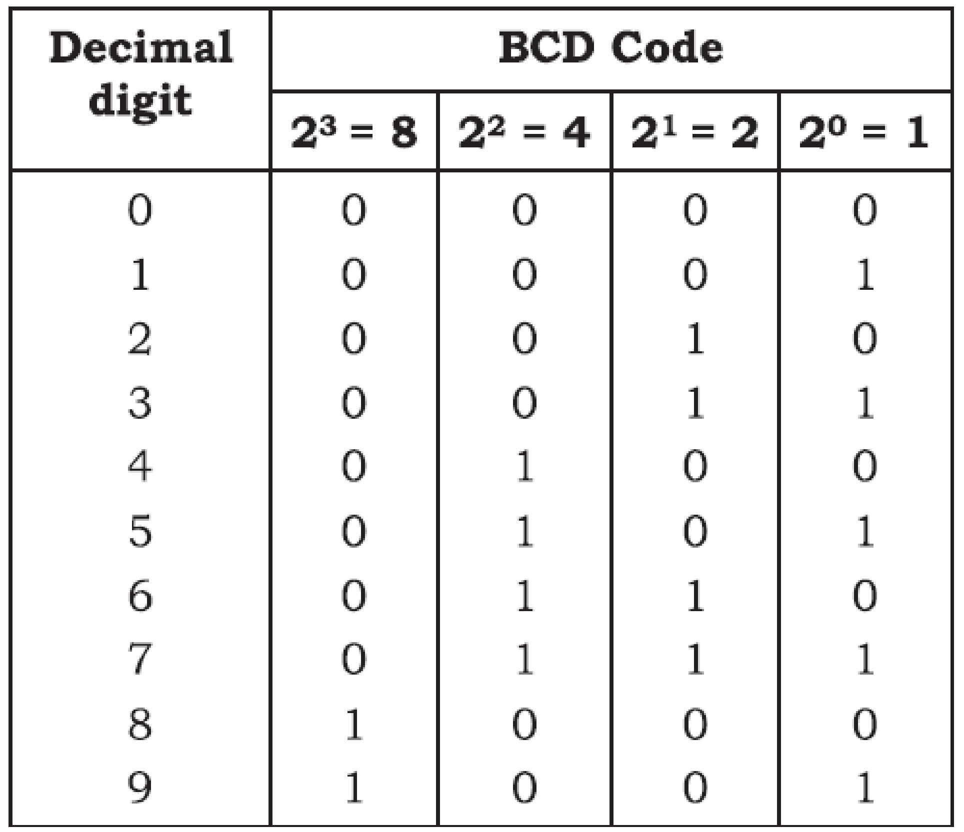
Result ⇒ -ve

Codes

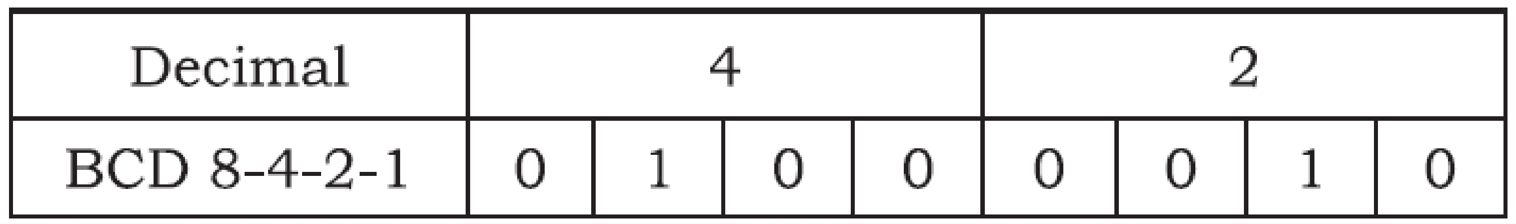
- They are just two binary digits which are 0 and 1. Types:
    - Numeric code – Digital code represented by numbers.
    - Alpha-numeric code – Digital code represented by alphabets, characters & numerals.
    - Weighted code – Binary codes which obey the positional weight principle. Ex: 8-4-2-1 code; 2-4-2-1 code
    - Non-weighted code – Binary codes which do not obey the positional weight principle. Ex: Excess-3 code; Gray code

Binary Coded Decimal (BCD) Codes

8-4-2-1 code

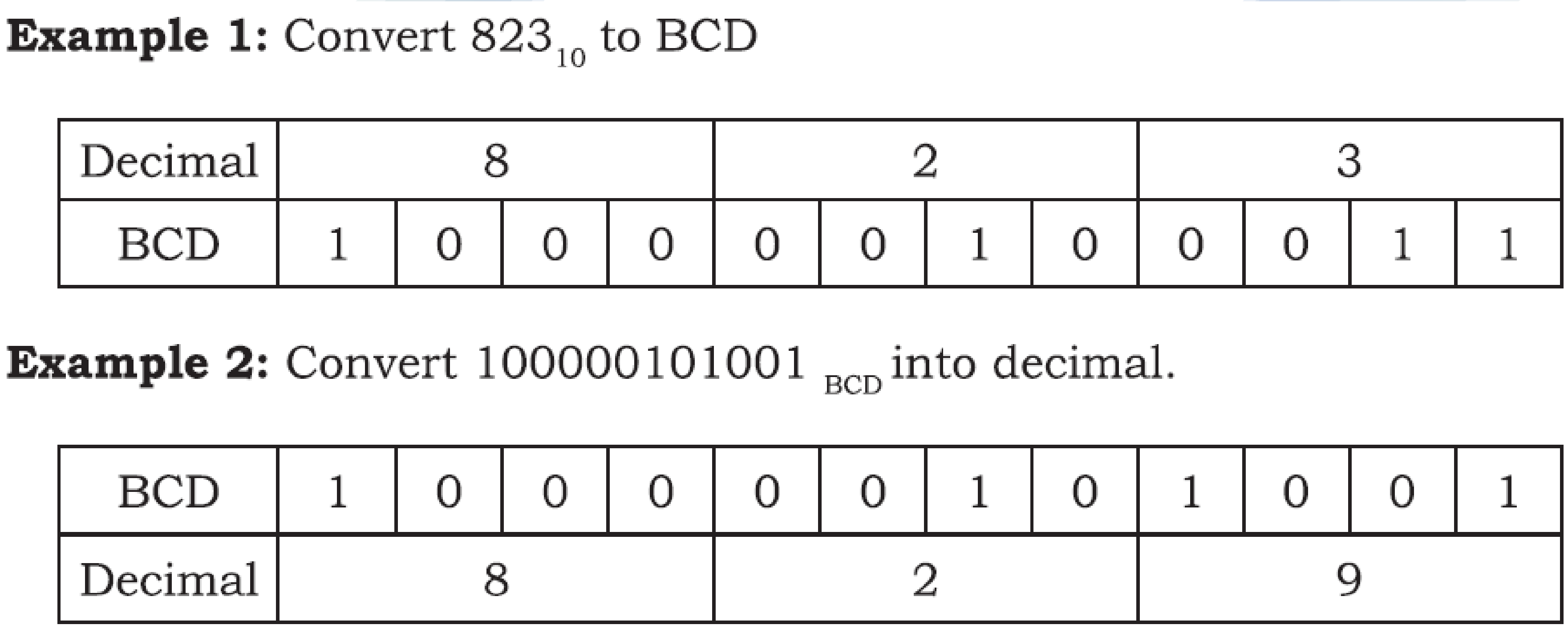


Decimal digit	BCD Code			
	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1



Decimal	4				2			
BCD 8-4-2-1	0	1	0	0	0	0	1	0

BCD Codes conversion: Few examples



Example 1: Convert 823_{10} to BCD

Decimal	8				2				3			
BCD	1	0	0	0	0	0	1	0	0	0	1	1

Example 2: Convert 100000101001_{BCD} into decimal.

BCD	1	0	0	0	0	0	1	0	1	0	0	1
Decimal	8				2				9			

BCD Codes conversion: Few examples

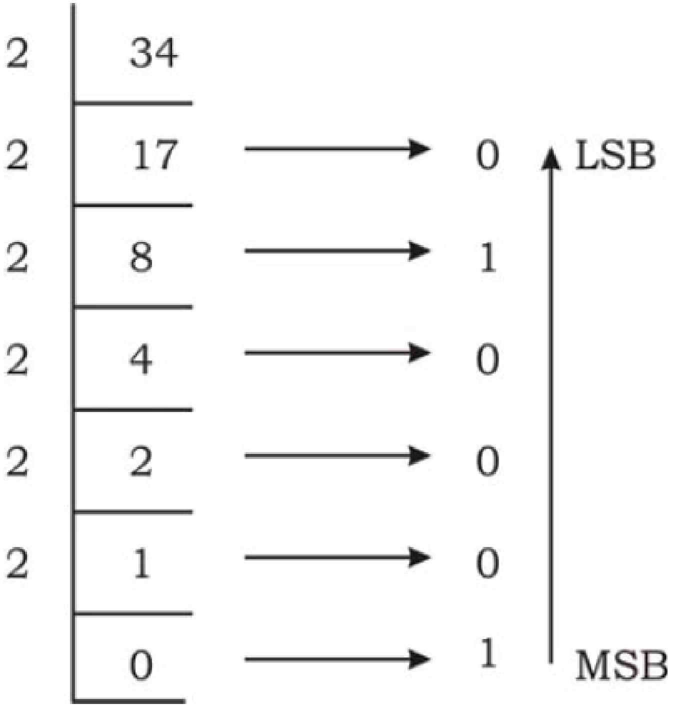
Example 3: Convert $(0011\ 0100)_{BCD}$ to binary

Solution:

Step 1: BCD to decimal conversion

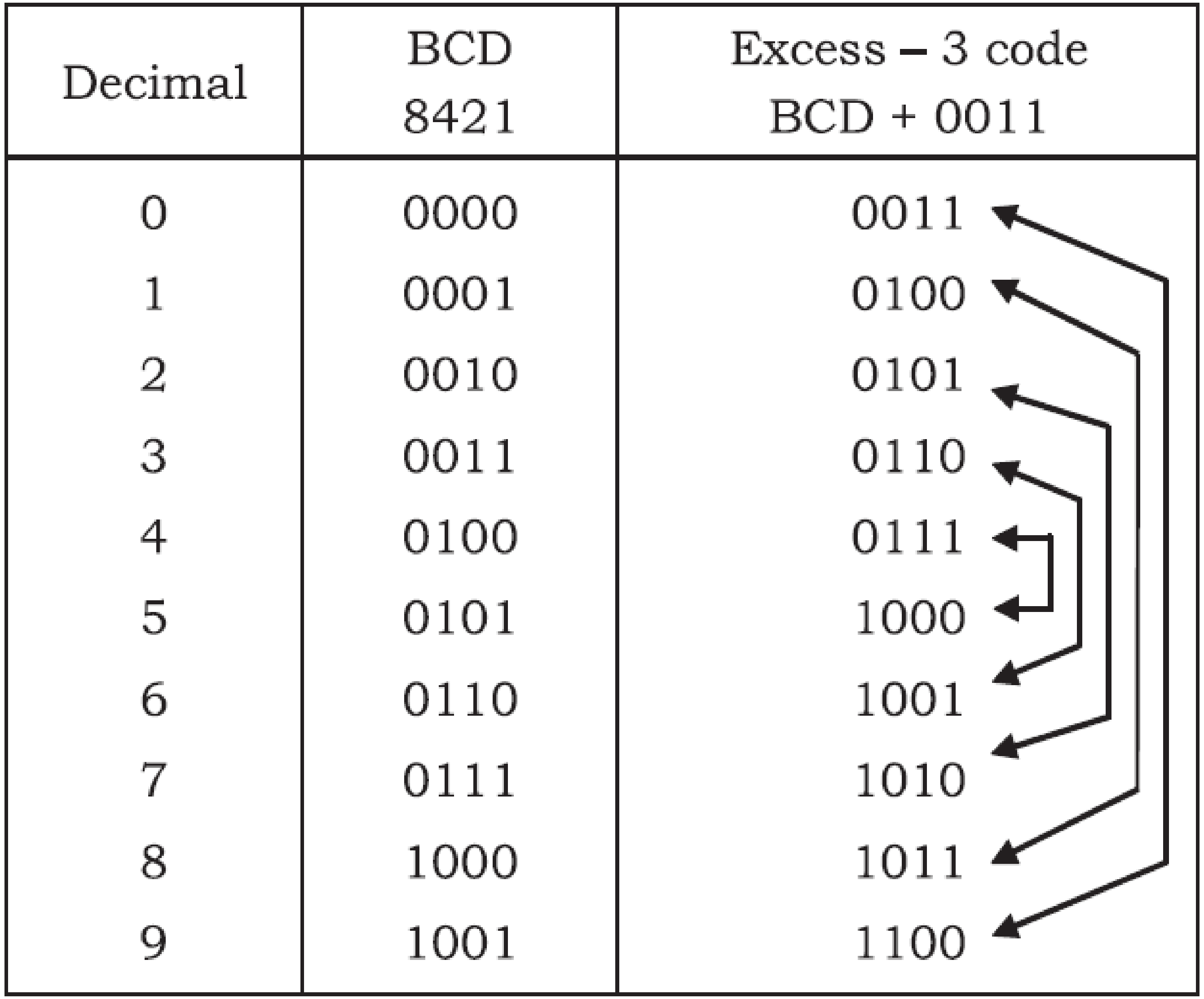
BCD	0	0	1	1	0	1	0	0
Decimal	3				4			

Step 2: Decimal to binary conversion



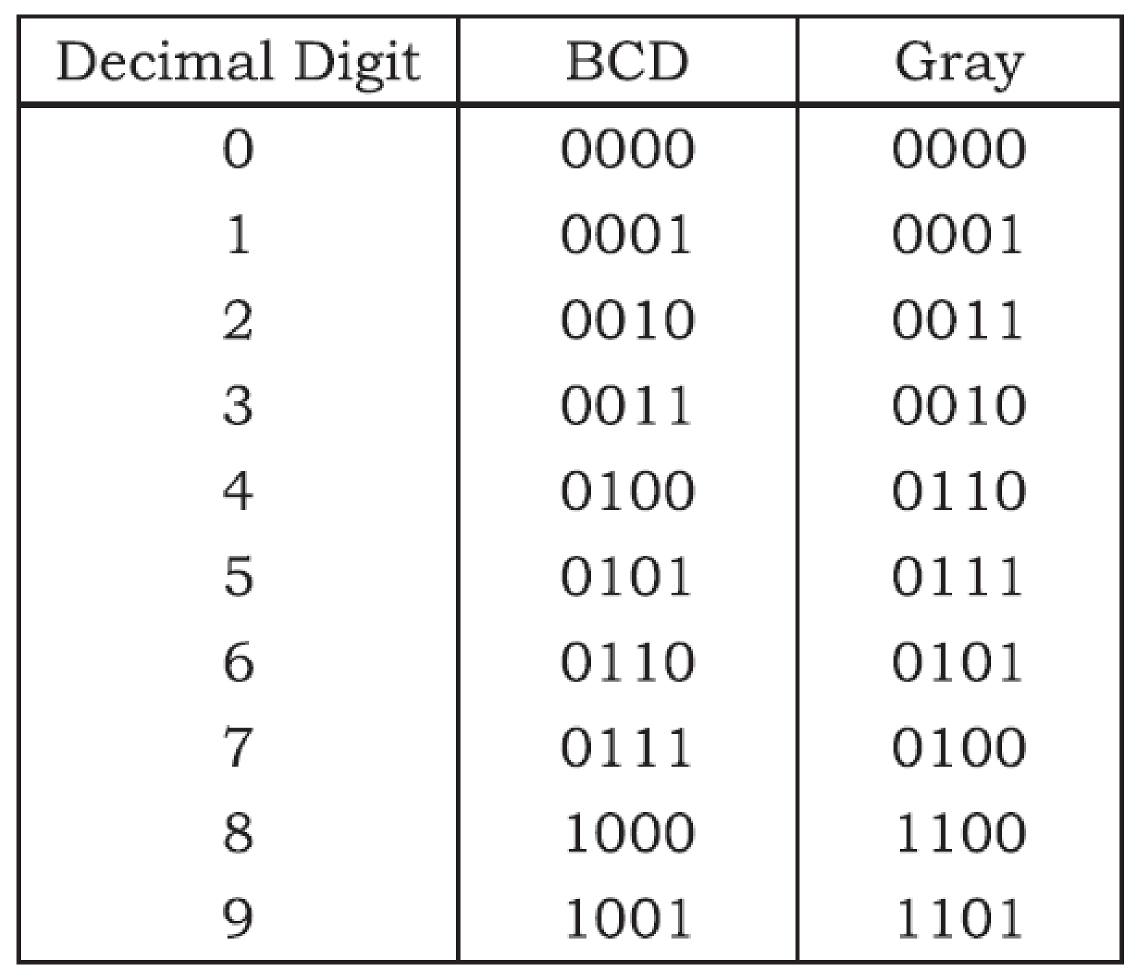
$$(0011\ 0100)_{BCD} = (34)_{10} = (100010)_2$$

Excess-3 Codes



Decimal	BCD 8421	Excess – 3 code BCD + 0011
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

Gray Codes



Decimal Digit	BCD	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101

Thank you

Codes

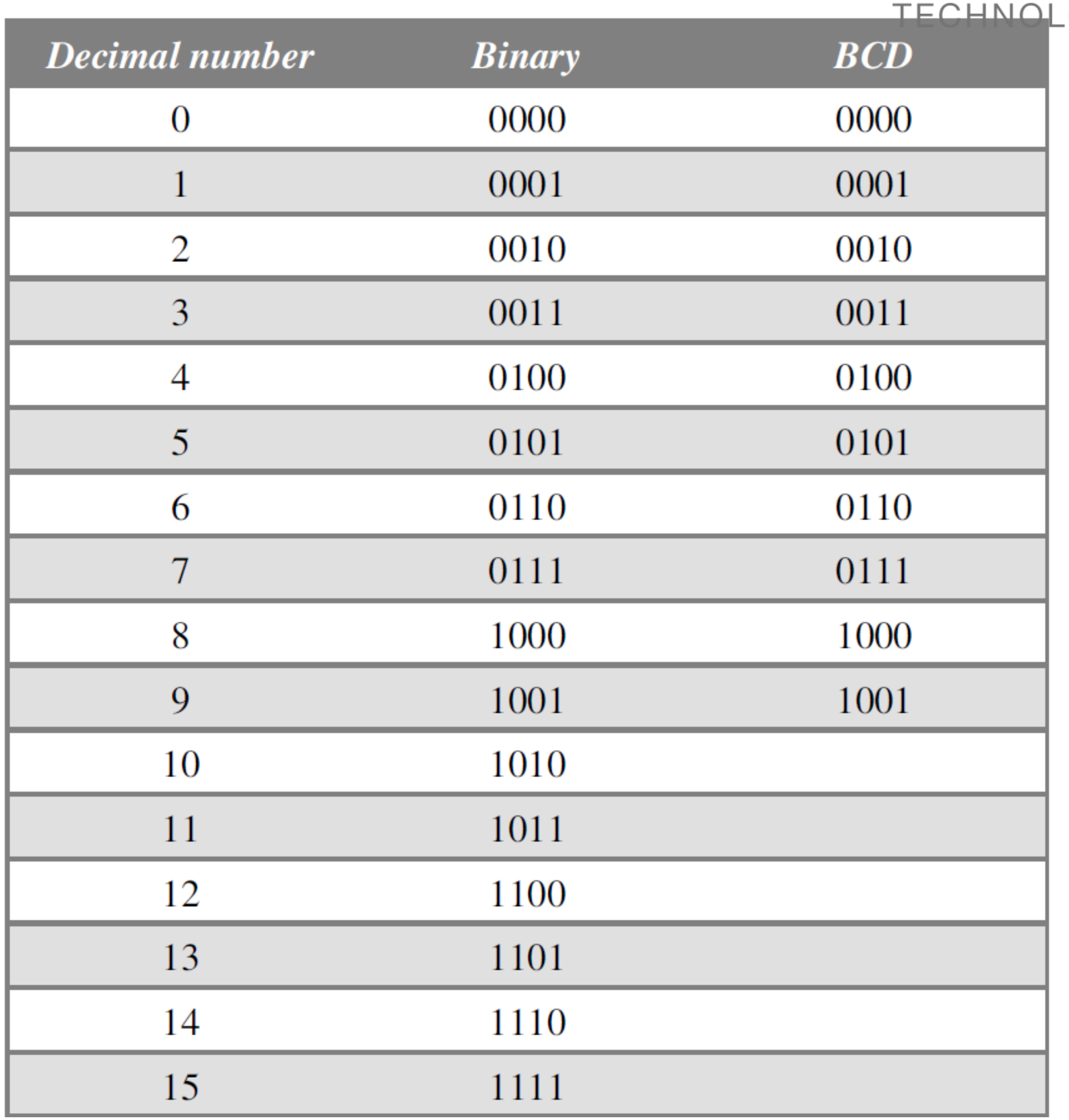
- Binary code
  - This is obtained by converting decimal numbers to their binary equivalents. The CPUs of computers process only binary numbers.
- BCD code
  - This is a binary code in which decimal digits 0 through 9 are represented by their binary equivalents using four bits. As the weights in the BCD code are 8, 4, 2, 1, it is also known as 8421 code. For instance, the decimal 257 converts to BCD as follows:

$$\begin{array}{ccc} 2 & 5 & 7 \\ \downarrow & \downarrow & \downarrow \\ 0010 & 0101 & 0111 \end{array}$$

  - The reverse conversion is similar. For instance,

$$1001\ 1000\ 0110 = (986)_{10}$$

Binary & BCD code



Decimal number	*Binary*	*BCD*
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	
11	1011	
12	1100	
13	1101	
14	1110	
15	1111	