1, 2, 3,...

B1A

1100

D2

Number System and Representation

0

01101

500

11001010

2A

Number System

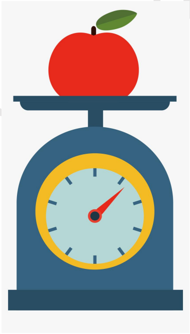
- Ordered set of basic symbols (digits)
- Radix/base of a number system

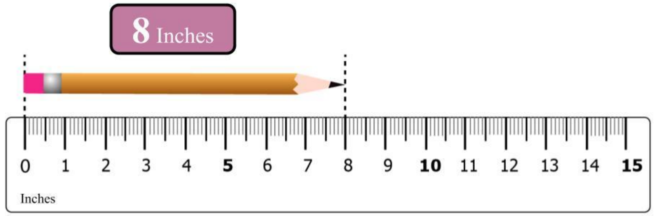
Total number of unique basic numerals/ symbols presents in the system

- E.g..
  - Decimal
  - Binary
  - Hexadecimal
  - Octal

Decimal Number System

- We use decimals every day while dealing with money, weight, length, etc
- Base/ Radix is 10 (0 - 9)





Decimal Number System

Decimal Number →

2	1	0
5	2	3

Base or Radix is 10

$$10^0 \times 3 = 3$$

$$10^1 \times 2 = 20$$

$$10^2 \times 5 = 500$$

$$\Sigma = 523$$

10^4	10^3	10^2	10^1	10^0
1	2	8	0	5

12805

How do we count using Decimal Numbers?

0
1
2
3
4
5
6
7
8
9

If you run out of digits
then
add **1** to the left
and
start from first digit

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

Binary Number System

- A Binary Number is made up of only 0s and 1s

110100

Example of a Binary Number

Binary Number System

- A Binary Number is made up of only 0s and 1s

A "bit" is a single binary digit. This number has 6 bits

110100

Example of a Binary Number

Binary Number System

- A Binary Number is made up of only 0s and 1s

A "bit" is a single binary digit. This number has 6 bits

110100

Example of a Binary Number

- There is no 2, 3, 4, 5, 6, 7, 8 or 9 in Binary!

How do we Count using Binary?

How do we count in Decimal? Vs How do we count in Binary?

0		We start at 0
1		Then 1
???		But then there is no symbol for 2 ... what do we do?

How do we Count using Binary?

How do we count in Decimal?

0	Start at 0
...	Count 1,2,3,4,5,6,7,8, and then...
9	This is the **last digit** in Decimal
10	So we start back at 0 again, but add 1 on the left

Vs

How do we count in Binary?

0		We start at 0
1		Then 1
???		But then there is no symbol for 2 ... what do we do?

How do we Count using Binary?

How do we count in Decimal? Vs How do we count in Binary?

0	Start at 0
...	Count 1,2,3,4,5,6,7,8, and then...
9	This is the **last digit** in Decimal
10	So we start back at 0 again, but add 1 on the left

0	We start at 0
1	Then 1
???	But then there is no symbol for 2 ... what do we do?

0	Start at 0
1	Then 1
10	Now start back at 0 again, but **add 1 on the left**
11	1 more
???	But NOW what ... ?

How do we Count using Binary?

How do we count in Decimal? Vs How do we count in Binary?

0	Start at 0
...	Count 1,2,3,4,5,6,7,8, and then...
9	This is the **last digit** in Decimal
10	So we start back at 0 again, but add 1 on the left

99	When we run out of digits, we ...
100	... start back at 0 again, but add 1 on the left

0		We start at 0
1		Then 1
???		But then there is no symbol for 2 ... what do we do?

0		Start at 0
1		Then 1
10		Now start back at 0 again, but **add 1 on the left**
11		1 more
???		But NOW what ... ?

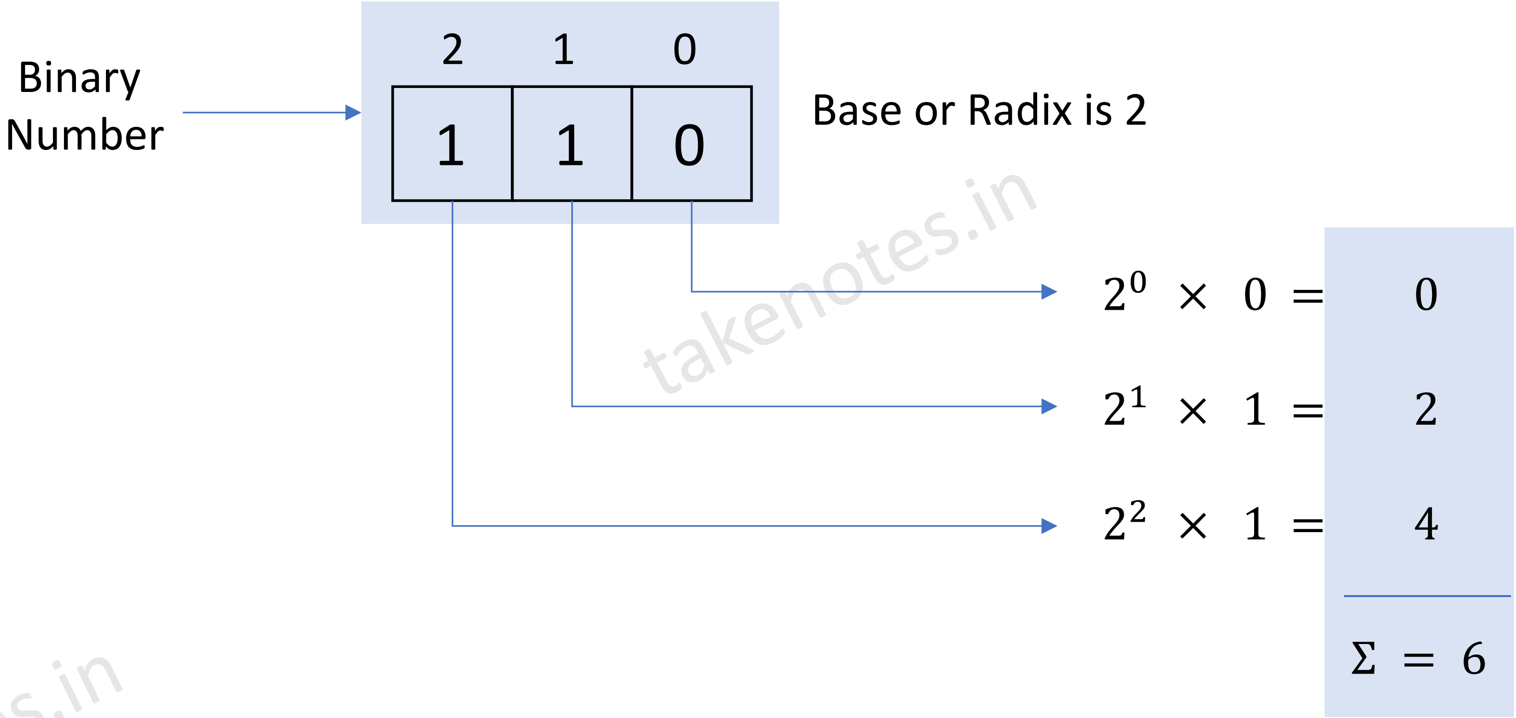
How do we Count using Binary?

How do we count in Decimal? Vs How do we count in Binary?

0	Start at 0
...	Count 1,2,3,4,5,6,7,8, and then...
9	This is the **last digit** in Decimal
10	So we start back at 0 again, but add 1 on the left
99	When we run out of digits, we ...
100	... start back at 0 again, but add 1 on the left

0	Start at 0
1	Then 1
10	Start back at 0 again, but add 1 on the left
11	
100	start back at 0 again, and add one to the number on the left... ... but that number is already at 1 so it also goes back to 0 ... ... and 1 is added to the *next position* on the left
101	
110	
111	
1000	Start back at 0 again (for all 3 digits), add 1 on the left
1001	And so on!

Binary Number System



110 represents 6 in binary number system

Binary Vs Decimal Numbers

			0	0
			1	1
		1	0	2
		1	1	3
	1	0	0	4

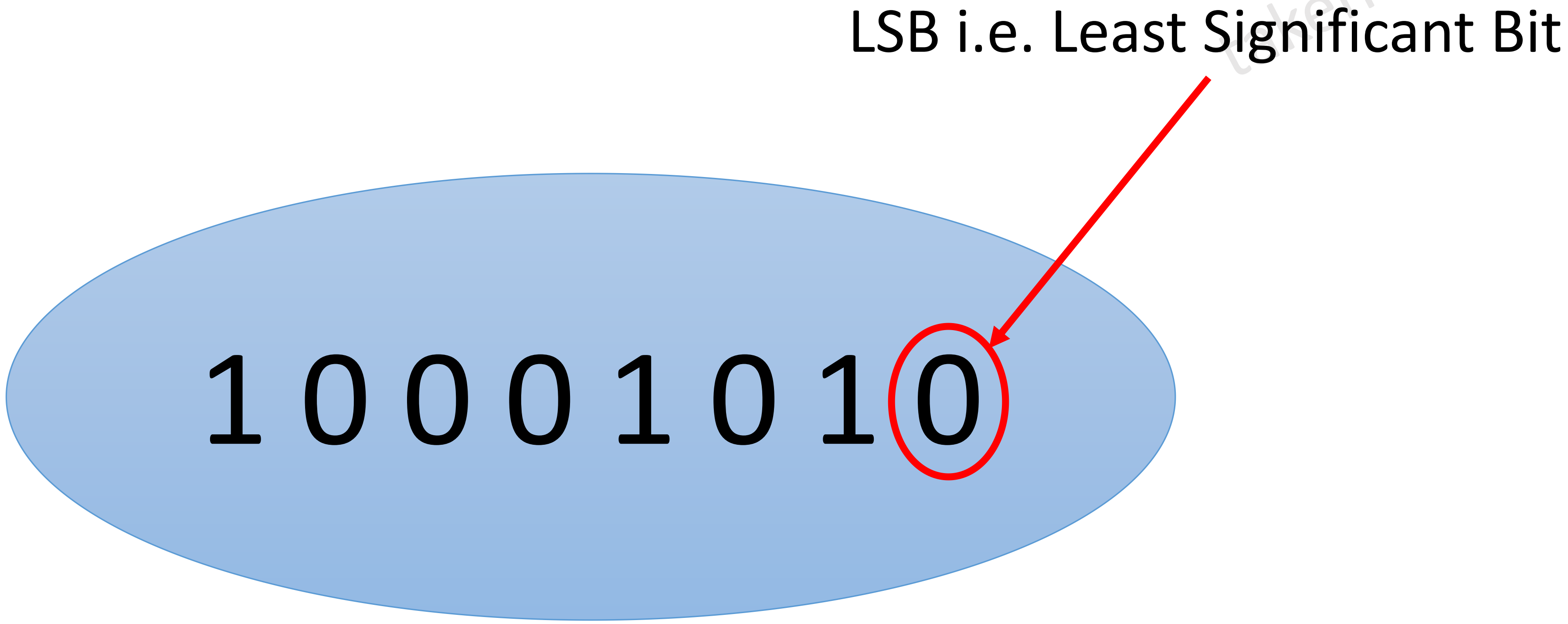
	1	0	1	5
	1	1	0	6
	1	1	1	7
1	0	0	0	8
1	0	0	1	9

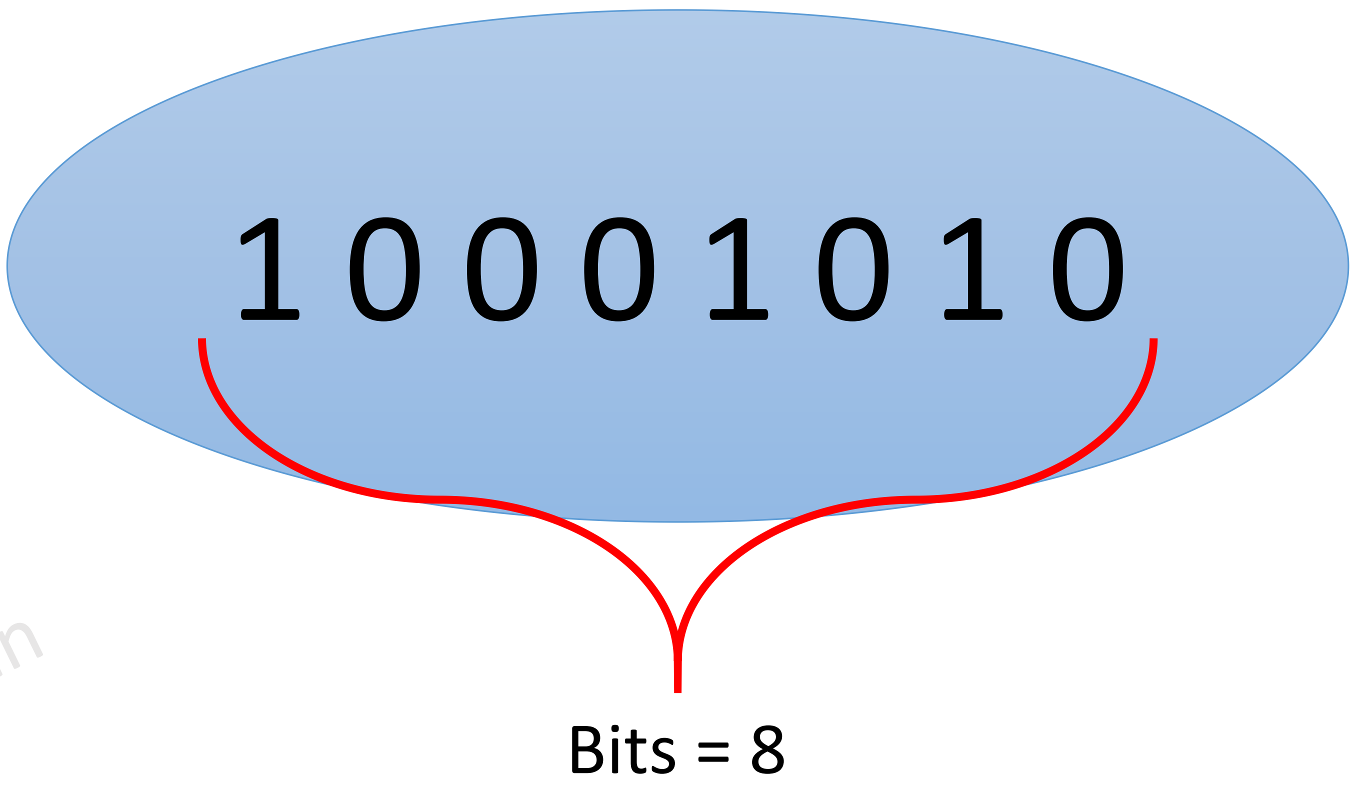
…

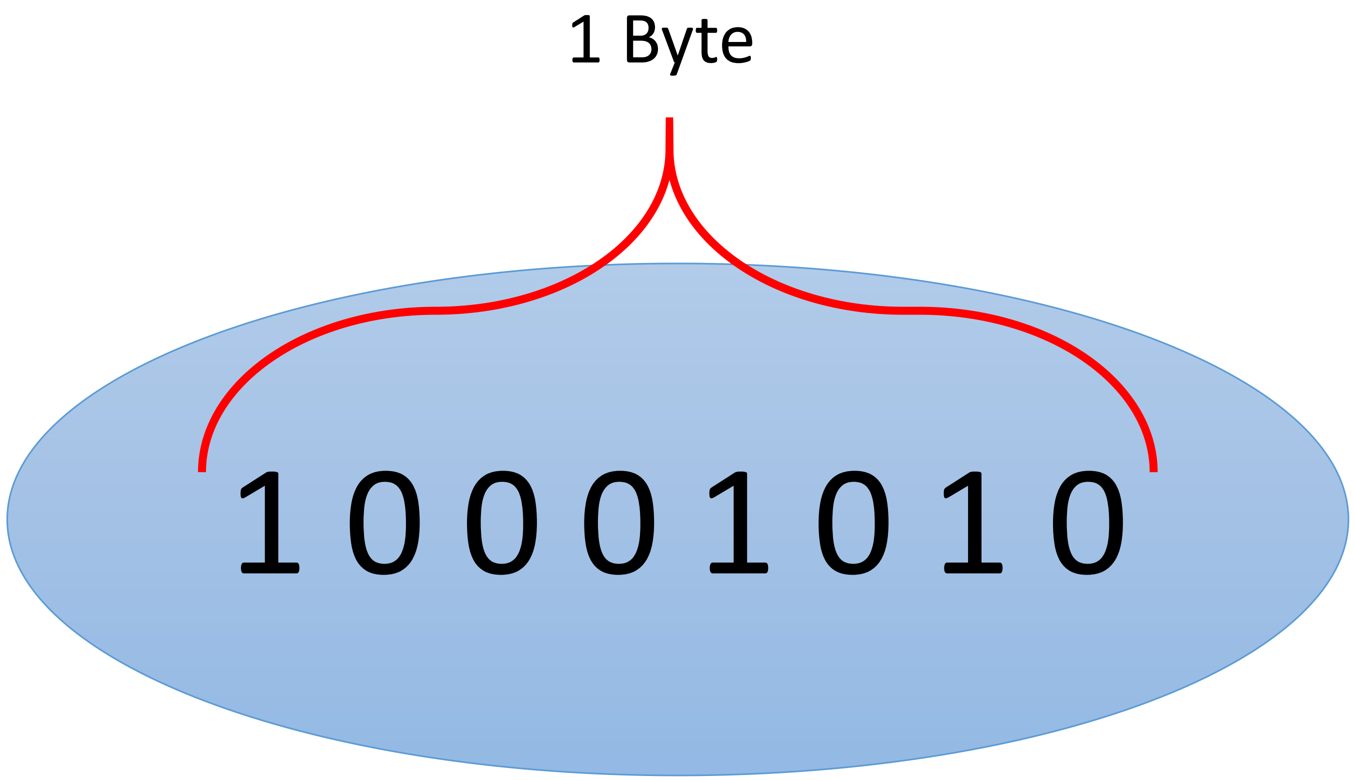
MSB i.e. Most Significant Bit

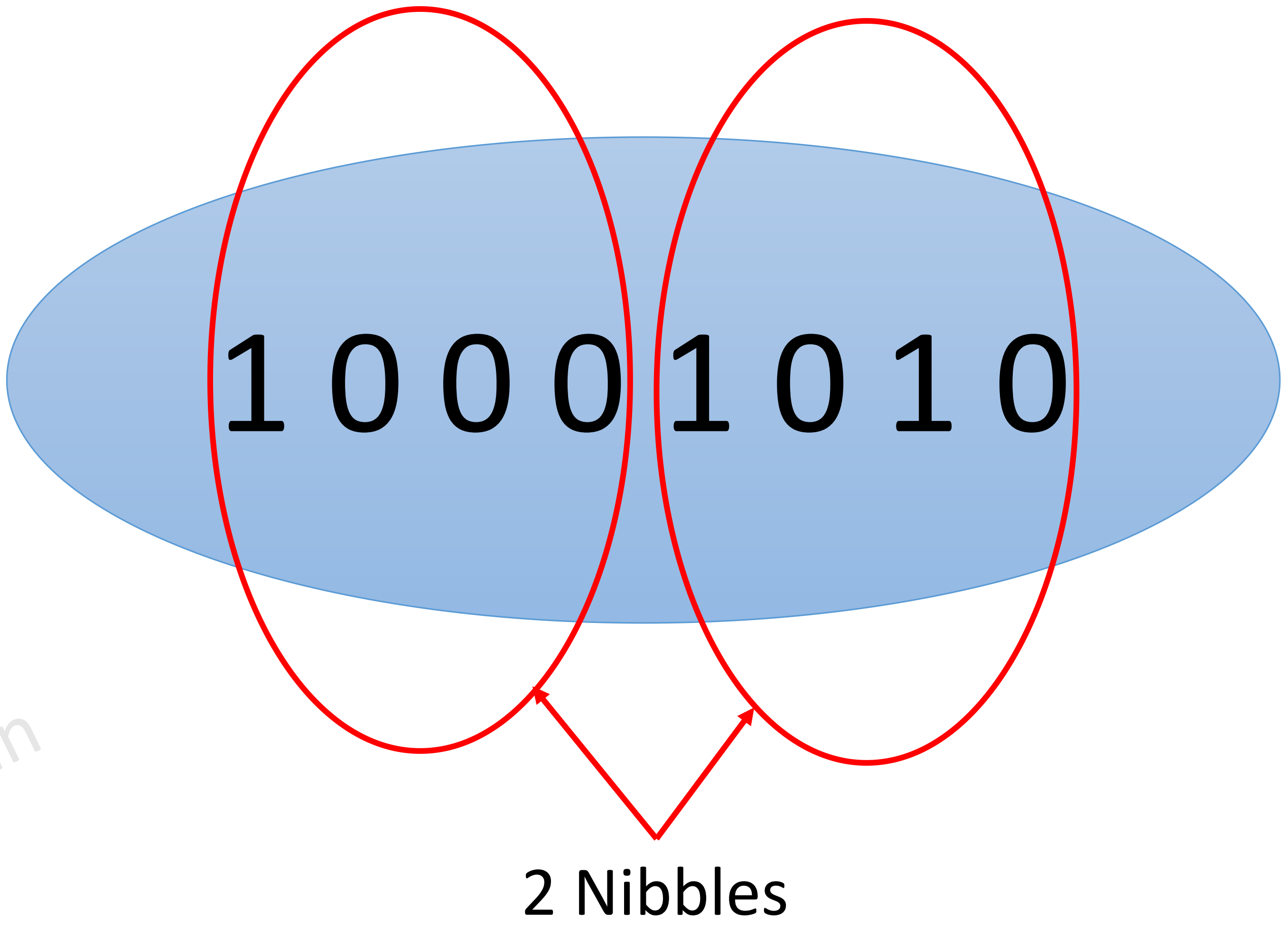
DON'T FORGET
A "bit" is a single binary digit

1 0 0 0 1 0 1 0


LSB i.e. Least Significant Bit
1 0 0 0 1 0 1 0


1 0 0 0 1 0 1 0
Bits = 8


1 Byte
1 0 0 0 1 0 1 0


1 0 0 0 1 0 1 0
2 Nibbles

Why different Number Systems?

Why can’t we use one Number system?

1, 2, 3,...

$$\frac{4}{5}$$

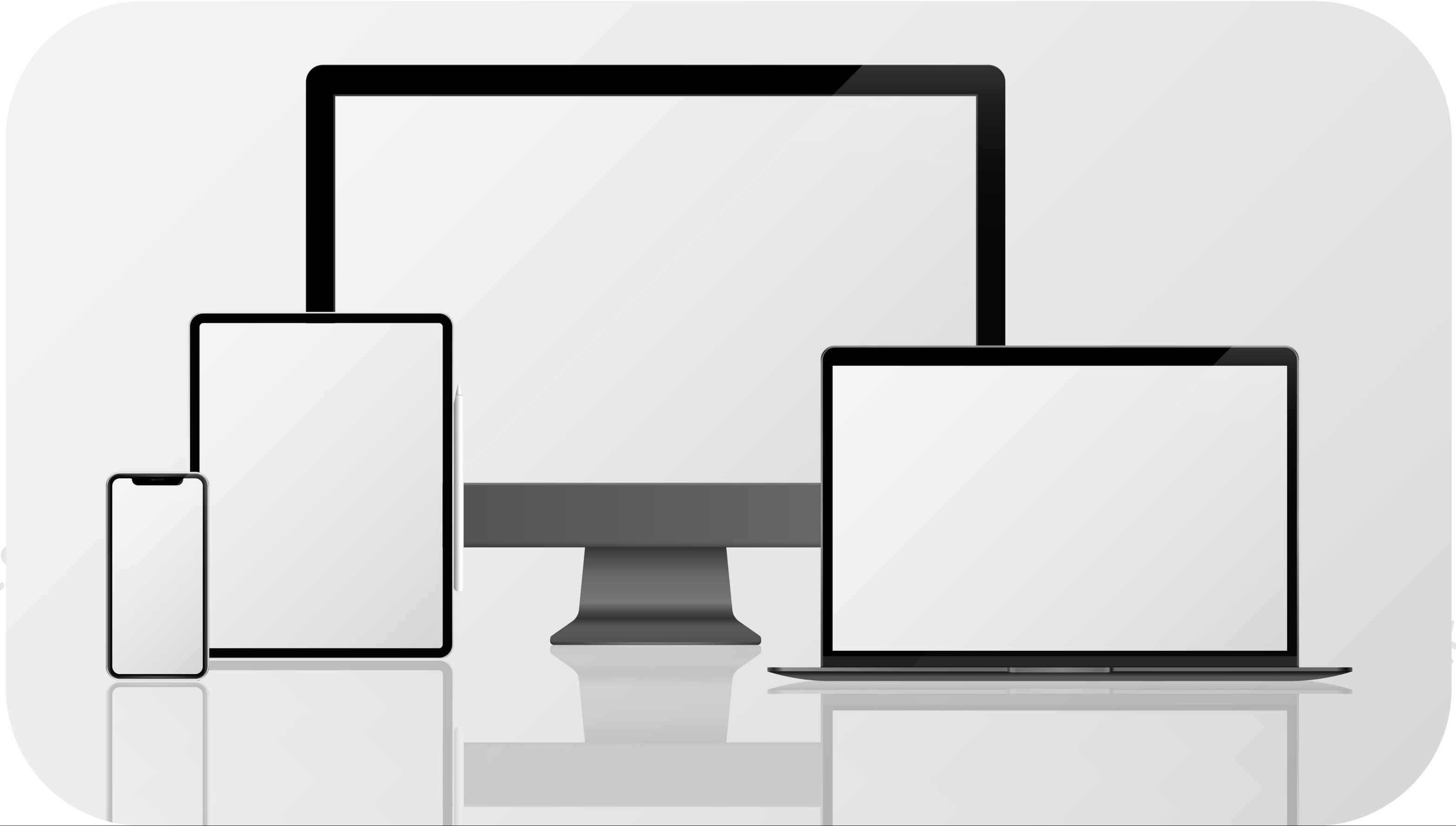
$$7^5$$

1.2746

01101

11001010

1100

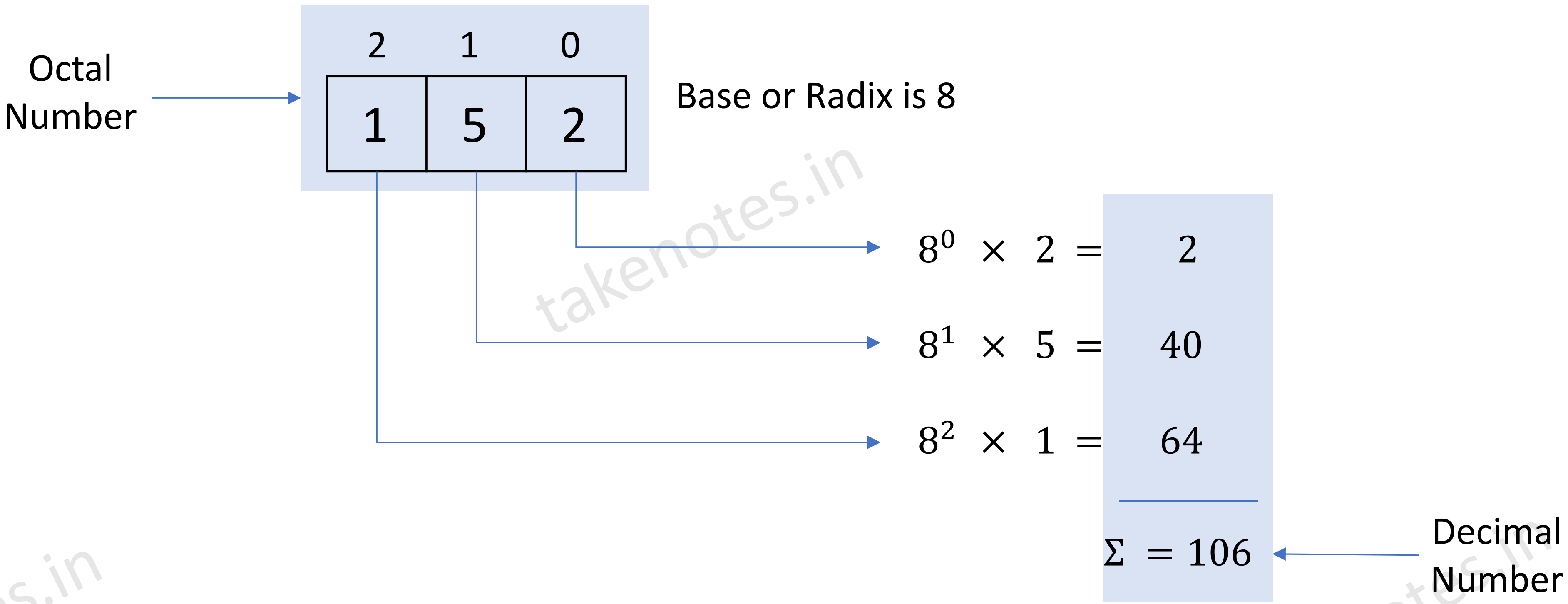


Octal Number System

Octal Number System

- The octal numeral system, or oct for short, is the base-8 number system, and uses the digits 0 to 7
- Octal and hexadecimal data types are integer types that are available in most computer languages. They provide a convenient notation for the construction of integer values in the binary number system.
- After CPUs converted to bit sizes of 32 and 64 bit octal was no longer utilised because, though both are still dividable by 8 they can also be divided by 16
  which is far more efficient.

Octal Number System



How do we count using Octal Number System?

Decimal Number	Octal Number
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

How do we count using Octal Number System?

Decimal Number	Octal Number
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

Decimal Number	Octal Number
8	8
8	10
9	11
10	12
11	13
12	14
13	15
14	16

How do we count using Octal Number System?

Decimal Number	Octal Number
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

Decimal Number	Octal Number
8	8
8	10
9	11
10	12
11	13
12	14
13	15
14	16

Decimal Number	Octal Number
15	17
16	18
16	20
17	21
18	22
19	23
20	24
21	25

How do we count using Octal Number System?

Decimal Number	Octal Number
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

Decimal Number	Octal Number
8	8
8	10
9	11
10	12
11	13
12	14
13	15
14	16

Decimal Number	Octal Number
15	17
16	18
16	20
17	21
18	22
19	23
20	24
21	25

• • •

How do we count using Octal Number System?

Decimal Number	Octal Number
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

Decimal Number	Octal Number
8	8
8	10
9	11
10	12
11	13
12	14
13	15
14	16

Decimal Number	Octal Number
15	17
16	18
16	20
17	21
18	22
19	23
20	24
21	25

• • •

Decimal Number	Octal Number
61	75
61	76
63	77
64	78
64	100
65	101
66	102
67	103

How do we count using Octal Number System?

Decimal Number	Octal Number
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

Decimal Number	Octal Number
8	8
8	10
9	11
10	12
11	13
12	14
13	15
14	16

Decimal Number	Octal Number
15	17
16	18
16	20
17	21
18	22
19	23
20	24
21	25

• • •

Decimal Number	Octal Number
61	75
61	76
63	77
64	78
64	100
65	101
66	102
67	103

Decimal Number	Octal Number
68	104
69	105
70	106
71	107
72	108
72	110
73	111
74	112

How do we count using Octal Number System?

Decimal Number	Octal Number
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

Decimal Number	Octal Number
8	8
8	10
9	11
10	12
11	13
12	14
13	15
14	16

Decimal Number	Octal Number
15	17
16	18
16	20
17	21
18	22
19	23
20	24
21	25

• • •

Decimal Number	Octal Number
61	75
61	76
63	77
64	78
64	100
65	101
66	102
67	103

Decimal Number	Octal Number
68	104
69	105
70	106
71	107
72	108
72	110
73	111
74	112

• • •

How do we count using Octal Number System?

Decimal Number	Octal Number
75	113
76	114
77	115
78	116
79	117
80	118
80	120
81	121

How do we count using Octal Number System?

Decimal Number	Octal Number
75	113
76	114
77	115
78	116
79	117
80	118
80	120
81	121

• • •

How do we count using Octal Number System?

Decimal Number	Octal Number
75	113
76	114
77	115
78	116
79	117
80	118
80	120
81	121

…

Decimal Number	Octal Number
123	173
124	174
125	175
126	176
127	177
128	178
128	200
129	201

How do we count using Octal Number System?

Decimal Number	Octal Number
75	113
76	114
77	115
78	116
79	117
80	118
80	120
81	121

• • •

Decimal Number	Octal Number
123	173
124	174
125	175
126	176
127	177
128	178
128	200
129	201

Decimal Number	Octal Number
130	202
131	203
132	204
133	205
134	206
135	207
136	208
136	210

How do we count using Octal Number System?

Decimal Number	Octal Number
75	113
76	114
77	115
78	116
79	117
80	118
80	120
81	121

• • •

Decimal Number	Octal Number
123	173
124	174
125	175
126	176
127	177
128	178
128	200
129	201

Decimal Number	Octal Number
130	202
131	203
132	204
133	205
134	206
135	207
136	208
136	210

• • •

How do we count using Octal Number System?

Decimal Number	Octal Number
75	113
76	114
77	115
78	116
79	117
80	118
80	120
81	121

…

Decimal Number	Octal Number
123	173
124	174
125	175
126	176
127	177
128	178
128	200
129	201

Decimal Number	Octal Number
130	202
131	203
132	204
133	205
134	206
135	207
136	208
136	210

…

Decimal Number	Octal Number
509	775
510	776
511	777
512	778
512	1000
513	1001
514	1002
515	1003

How do we count using Octal Number System?

Decimal Number	Octal Number
75	113
76	114
77	115
78	116
79	117
80	118
80	120
81	121

• • •

Decimal Number	Octal Number
123	173
124	174
125	175
126	176
127	177
128	178
128	200
129	201

Decimal Number	Octal Number
130	202
131	203
132	204
133	205
134	206
135	207
136	208
136	210

• • •

Decimal Number	Octal Number
509	775
510	776
511	777
512	778
512	1000
513	1001
514	1002
515	1003

• • •

Octal Number System

Decimal Number	3 bit binary Numer	Octal Number
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

Decimal Number	3 bit binary Numer	Octal Number
8	001 000	10
9	001 001	11
10	001 010	12
11	001 011	13
12	001 100	14
13	001 101	15
14	001 110	16
15	001 111	17

Hexadecimal Number System

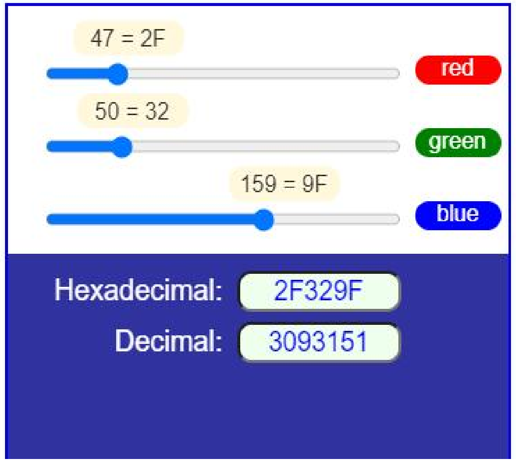
Hexadecimal Number System

- The hexadecimal numeral system, often shortened to "hex", is a numeral system made up of 16 symbols (base 16)
- The standard numeral system is called decimal (base 10) and uses ten symbols: 0,1,2,3,4,5,6,7,8,9.
- Hexadecimal uses the decimal numbers and six extra symbols
- Hexadecimal numerals are widely used by computer system designers and programmers because they provide a human-friendly representation of binary-coded values

The color is defined by its mix of **Red**, **Green** and **Blue**, each of which can be in the range:

0 to 255 (in decimal) , or

00 to **FF** (in hexadecimal)



How do we count using Hexadecimal System?

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

How do we count using Hexadecimal System?

Decimal Number	Hex Number	Decimal Number	Hex Number	Decimal Number	Hex Number
0	0	8	8	16	10
1	1	9	9	17	11
2	2	10	A	18	12
3	3	11	B	19	13
4	4	12	C	20	14
5	5	13	D	21	15
6	6	14	E	22	16
7	7	15	F	23	17

How do we count using Hexadecimal System?

Decimal Number	Hex Number	Decimal Number	Hex Number	Decimal Number	Hex Number	Decimal Number	Hex Number
0	0	8	8	16	10	24	18
1	1	9	9	17	11	25	19
2	2	10	A	18	12	26	1A
3	3	11	B	19	13	27	1B
4	4	12	C	20	14	28	1C
5	5	13	D	21	15	29	1D
6	6	14	E	22	16	30	1E
7	7	15	F	23	17	31	1F

How do we count using Hexadecimal System?

Decimal Number	Hex Number	Decimal Number	Hex Number	Decimal Number	Hex Number	Decimal Number	Hex Number	Decimal Number	Hex Number
0	0	8	8	16	10	24	18	32	20
1	1	9	9	17	11	25	19	33	21
2	2	10	A	18	12	26	1A	34	22
3	3	11	B	19	13	27	1B	35	23
4	4	12	C	20	14	28	1C	36	24
5	5	13	D	21	15	29	1D	37	25
6	6	14	E	22	16	30	1E	38	26
7	7	15	F	23	17	31	1F	39	27

How do we count using Hexadecimal System?

Decimal Number	Hex Number	Decimal Number	Hex Number	Decimal Number	Hex Number	Decimal Number	Hex Number	Decimal Number	Hex Number	Decimal Number	Hex Number
0	0	8	8	16	10	24	18	32	20	40	28
1	1	9	9	17	11	25	19	33	21	41	29
2	2	10	A	18	12	26	1A	34	22	42	2A
3	3	11	B	19	13	27	1B	35	23	43	2B
4	4	12	C	20	14	28	1C	36	24	44	2C
5	5	13	D	21	15	29	1D	37	25	45	2D
6	6	14	E	22	16	30	1E	38	26	46	2E
7	7	15	F	23	17	31	1F	39	27	47	2F

How do we count using Hexadecimal System?

Decimal Number	Hex Number
153	99
154	9A
155	9B
156	9C
157	9D
158	9E
159	9F
160	A0

How do we count using Hexadecimal System?

Decimal Number	Hex Number	Decimal Number	Hex Number
153	99	161	A1
154	9A	162	A2
155	9B	163	A3
156	9C	164	A4
157	9D	165	A5
158	9E	166	A6
159	9F	167	A7
160	A0	168	A8

How do we count using Hexadecimal System?

Decimal Number	Hex Number	Decimal Number	Hex Number	Decimal Number	Hex Number
153	99	161	A1	169	A9
154	9A	162	A2	170	AA
155	9B	163	A3	171	AB
156	9C	164	A4	172	AC
157	9D	165	A5	173	AD
158	9E	166	A6	174	AE
159	9F	167	A7	175	AF
160	A0	168	A8	176	B0

…

How do we count using Hexadecimal System?

Decimal Number	Hex Number	Decimal Number	Hex Number	Decimal Number	Hex Number	Decimal Number	Hex Number
153	99	161	A1	169	A9	249	F9
154	9A	162	A2	170	AA	250	FA
155	9B	163	A3	171	AB	251	FB
156	9C	164	A4	172	AC	252	FC
157	9D	165	A5	173	AD	253	FD
158	9E	166	A6	174	AE	254	FE
159	9F	167	A7	175	AF	255	FF
160	A0	168	A8	176	B0	256	100

How do we count using Hexadecimal System?

Decimal Number	Hex Number	Decimal Number	Hex Number	Decimal Number	Hex Number	Decimal Number	Hex Number	Decimal Number	Hex Number
153	99	161	A1	169	A9	249	F9		
154	9A	162	A2	170	AA	250	FA	266	10A
155	9B	163	A3	171	AB	251	FB	267	10B
156	9C	164	A4	172	AC	252	FC	268	10C
157	9D	165	A5	173	AD	253	FD	269	10D
158	9E	166	A6	174	AE	254	FE	270	10E
159	9F	167	A7	175	AF	255	FF	271	10F
160	A0	168	A8	176	B0	256	100	272	110

Hexadecimal Number System

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

A, B, C, D, E, F

Hexadecimal Number

1	0
3	F

Base or Radix is 16

$$16^0 \times 15 = 15$$

$$16^1 \times 3 = 48$$

$$\Sigma = 63$$

3F represents 63 in hexadecimal number system

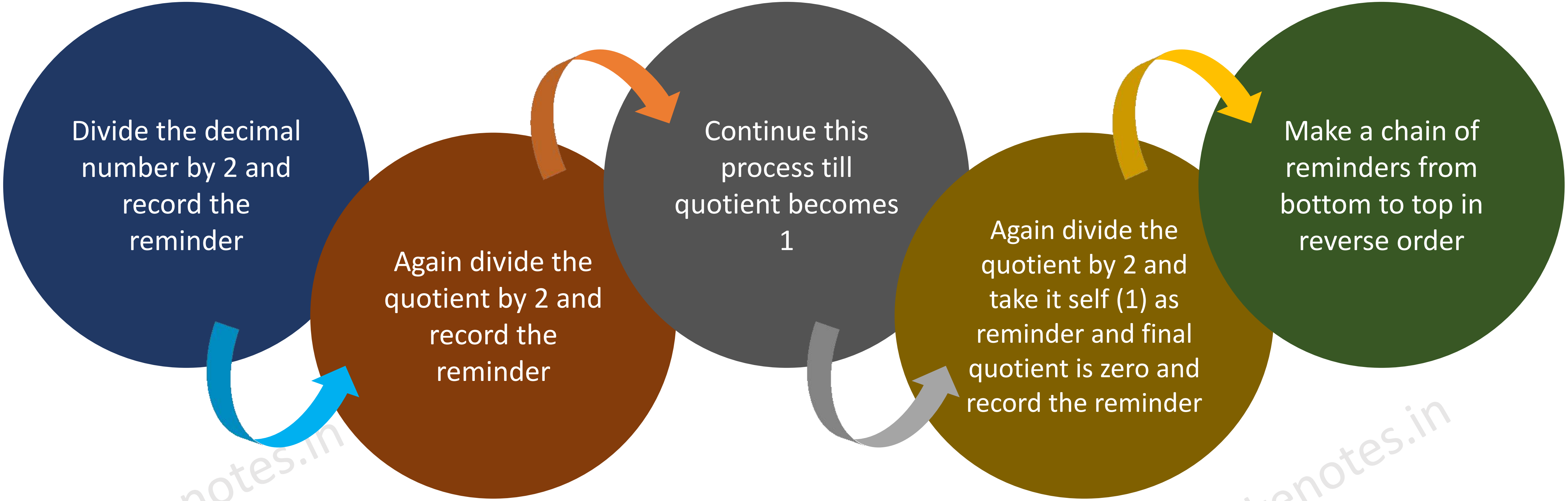
How to find given number belongs to which Number System????

$(123)_{10}$ $(123)_8$ $(1111011)_2$ $(123)_{16}$

$(1011)_{10}$ $(1011)_8$ $(1011)_2$ $(1011)_{16}$

Converting Decimal Number to Binary Number

Steps


Divide the decimal number by 2 and record the reminder
Again divide the quotient by 2 and record the reminder
Continue this process till quotient becomes 1
Again divide the quotient by 2 and take it self (1) as reminder and final quotient is zero and record the reminder
Make a chain of reminders from bottom to top in reverse order

$(12)_{10} = (?)_2$

2	12	0
	6	

$\frac{12}{2} = 6$ and Reminder=0

$(12)_{10} = (?)_2$

2	12	0
2	6	0
	3	

$\frac{6}{2} = 3$ and Reminder=0

$(12)_{10} = (?)_2$

2	12	0
2	6	0
2	3	1
	1	

$\frac{3}{2} = 1$ and Reminder=1

$(12)_{10} = (?)_2$

2	12	0
2	6	0
2	3	1
2	1	1
	0	

$\frac{1}{2} = 0$ and Reminder=1

$(12)_{10} = (?)_2$

2	12	0
2	6	0
2	3	1
2	1	1
	0	

$\boxed{1\,1\,0\,0}$

$(12)_{10} = (1\,1\,0\,0)_2$

Steps for fractional number

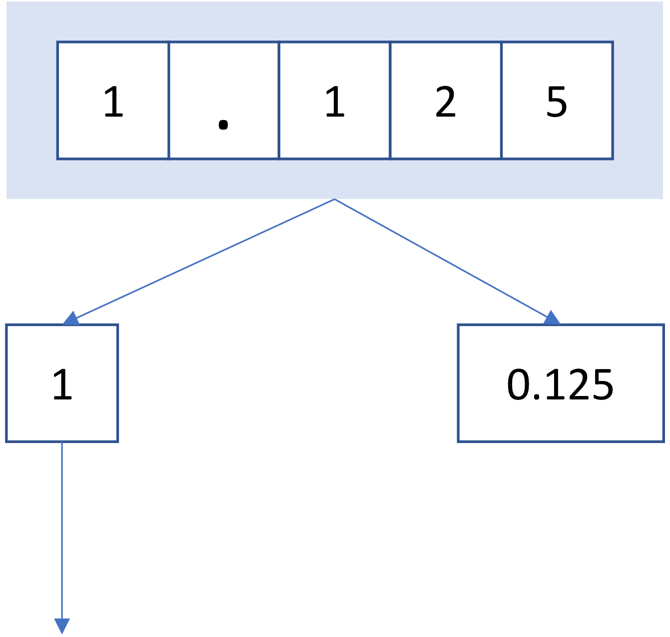
Multiply given fractional number by 2 and record the integer of the product

Again multiply the remaining fractional part by 2 and record the integer of the product

Continue this process till the fractional part becomes zero

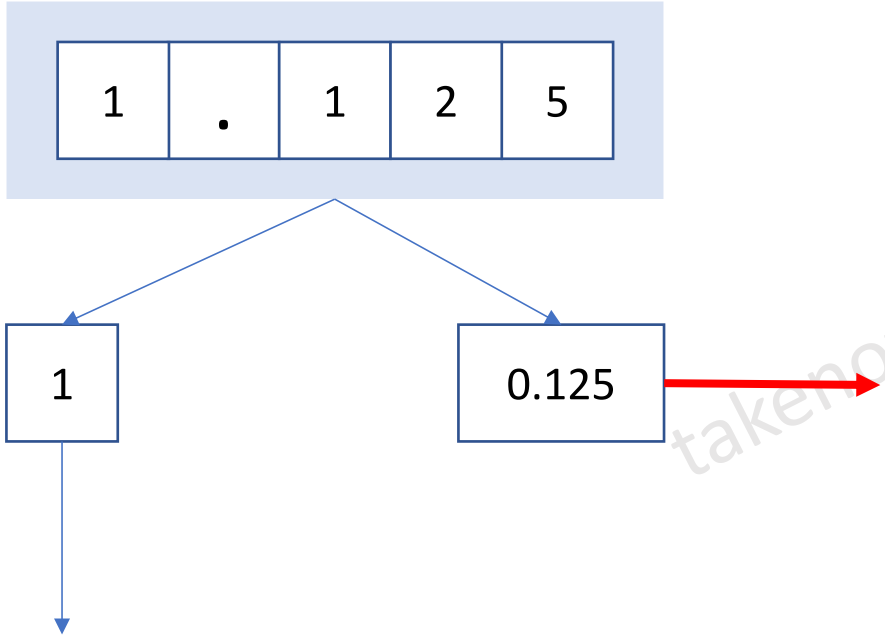
Make a chain of integers from top to bottom in forward order

$(1.125)_{10} = (?)_2$



$(1)_{10} = (1)_2$

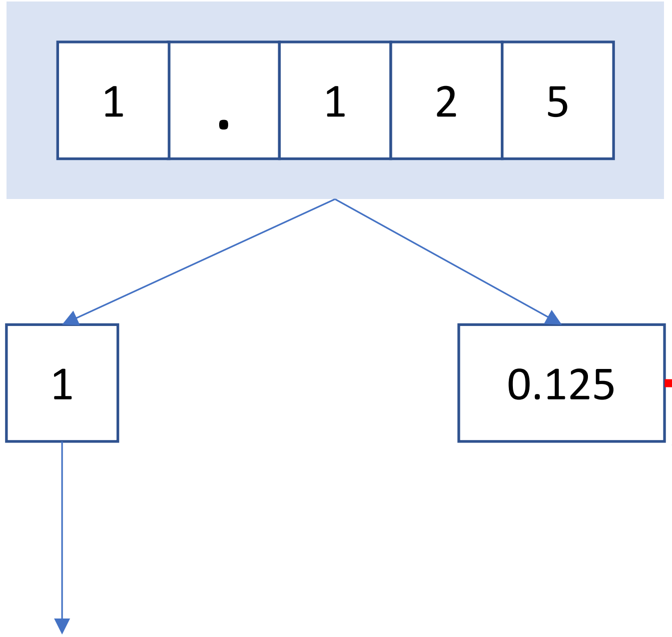
$(1.125)_{10} = (?)_2$



$(1)_{10} = (1)_2$

$2 \times 0.125 = 0.250 \quad 0$

$(1.125)_{10} = (?)_2$

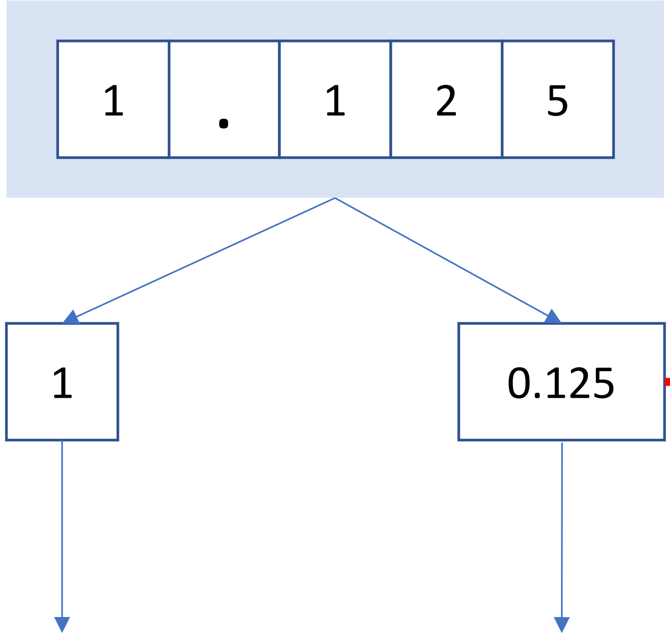


$(1)_{10} = (1)_2$

$2 \times 0.125 = 0.250 \quad 0$

$2 \times 0.250 = 0.50 \quad 0$

$(1.125)_{10} = (?)_2$



2	$\times$	0.125	=	0.250	0
2	$\times$	0.250	=	0.50	0
2	$\times$	0.50	=	1.0	1

$(1)_{10} = (1)_2$

$(0.125)_{10} = (001)_2$

$(1.125)_{10} = (1.001)_2$

Converting Binary Number to Decimal Number

$(1\ 0\ 1\ 0)_2 = (?)_{10}$

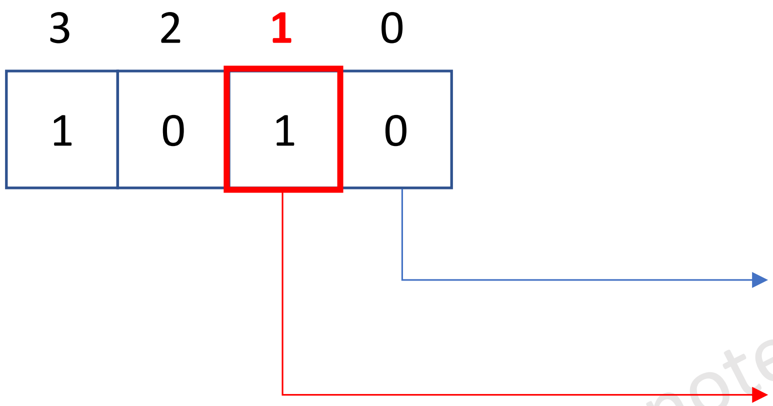
3	2	1	0
1	0	1	0

$(1\ 0\ 1\ 0)_2 = (?)_{10}$

3	2	1	**0**
1	0	1	0

$$2^0 \times 0 = \quad 0$$

$(1\ 0\ 1\ 0)_2 = (?)_{10}$



$2^0 \times 0 = 0$

$2^1 \times 1 = 2$

$(1\ 0\ 1\ 0)_2 = (?)_{10}$

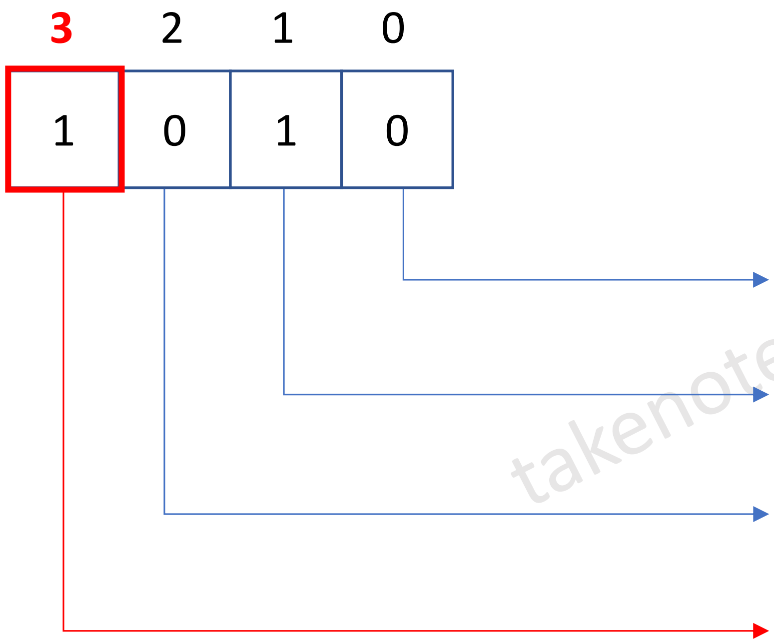
3	**2**	1	0
1	0	1	0

$2^0 \times 0 = 0$

$2^1 \times 1 = 2$

$2^2 \times 0 = 0$

$(1\ 0\ 1\ 0)_2 = (?)_{10}$



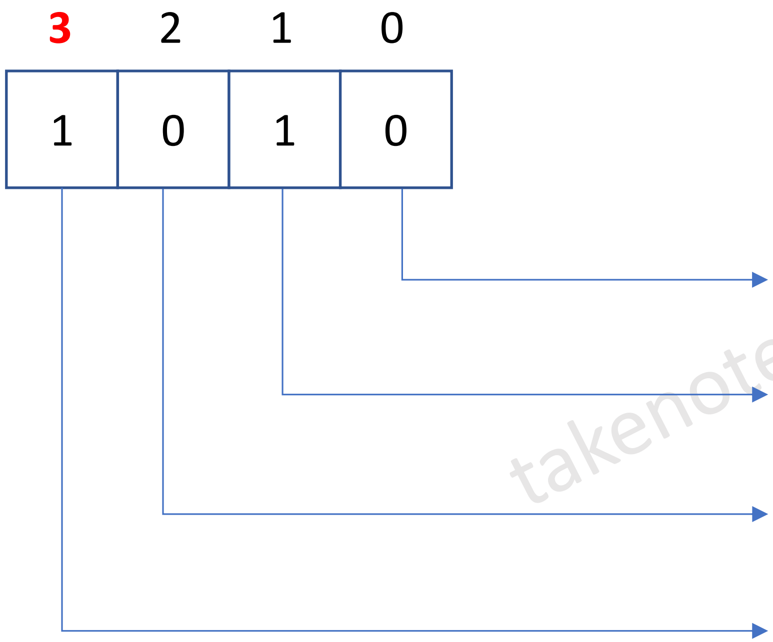
$2^0 \times 0 = 0$

$2^1 \times 1 = 2$

$2^2 \times 0 = 0$

$2^3 \times 1 = 8$

$(1\ 0\ 1\ 0)_2 = (?)_{10}$



$$2^0 \times 0 = 0$$
$$2^1 \times 1 = 2$$
$$2^2 \times 0 = 0$$
$$2^3 \times 1 = 8$$
$$\Sigma = 10$$

$(1\ 0\ 1\ 0)_2 = (10)_{10}$

Fractional Number

$(1\,0\,.\,1\,0)_2 = (?)_{10}$

1	0		-1	-2
1	0	.	1	0

$(1\,0\,.\,1\,0)_2 = (?)_{10}$

1	0		-1	**-2**
1	0	.	1	$\boxed{0}$

$2^{-2} \times 0 = 0$

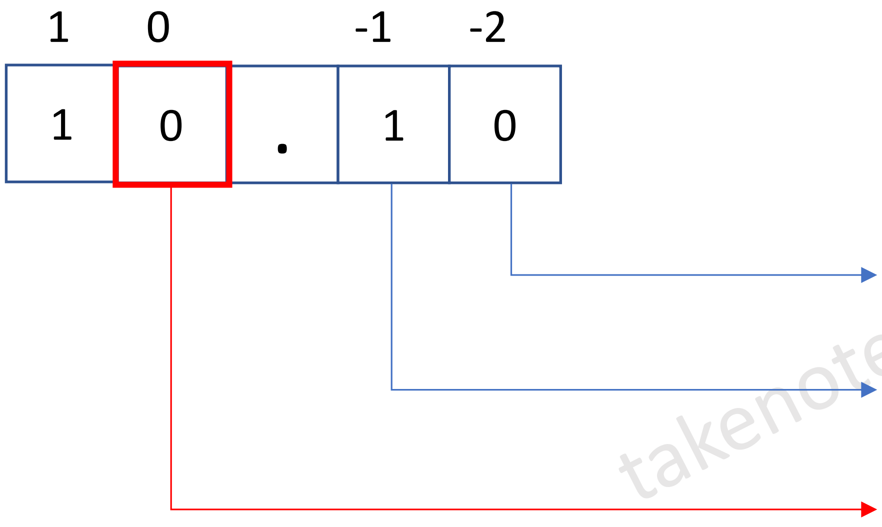
$(1\,0\,.\,1\,0)_2 = (?)_{10}$

1	0		-1	-2
1	0	.	1	0

$$2^{-2} \times 0 = 0$$

$$2^{-1} \times 1 = 0.5$$

$(1\,0\,.\,1\,0)_2 = (?)_{10}$

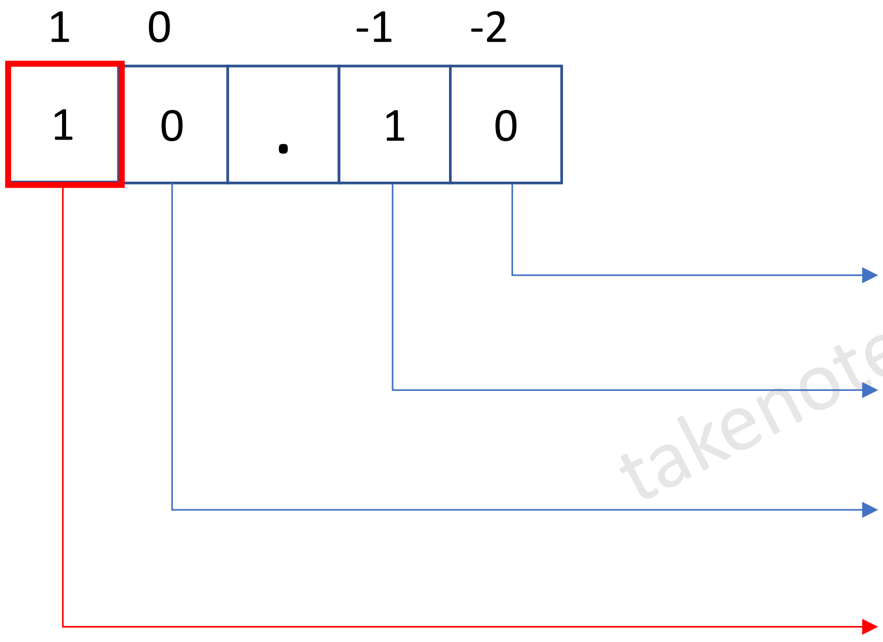


$2^{-2} \times 0 = 0$

$2^{-1} \times 1 = 0.5$

$2^{0} \times 0 = 0$

$(1\,0\,.\,1\,0)_2 = (?)_{10}$



$2^{-2} \times 0 = 0$

$2^{-1} \times 1 = 0.5$

$2^{0} \times 0 = 0$

$2^{1} \times 1 = 2$

$(1\,0\,.\,1\,0)_2 = (?)_{10}$

1	0		-1	-2
1	0	.	1	0

$2^{-2} \times 0 = 0$

$2^{-1} \times 1 = 0.5$

$2^{0} \times 0 = 0$

$2^{1} \times 1 = 2$

$\Sigma = 2.5$

$(1\,0.\,1\,0)_2 = (2.5)_{10}$

Converting Decimal Number to Octal Number

Decimal to Octal

Decimal Number	Octal Number
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17

8	8	0
	1	

$\frac{8}{8} = 1$ and Reminder=0

Decimal to Octal

Decimal Number	Octal Number
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17

8	8	0
8	1	1
	0	

$\frac{1}{8} = 0$ and Reminder=1

Decimal to Octal

Decimal Number	Octal Number
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17

8	8	0
8	1	1
	0	

1 0

$(8)_{10} = (10)_8$

Decimal to Octal

Decimal Number	Octal Number
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17

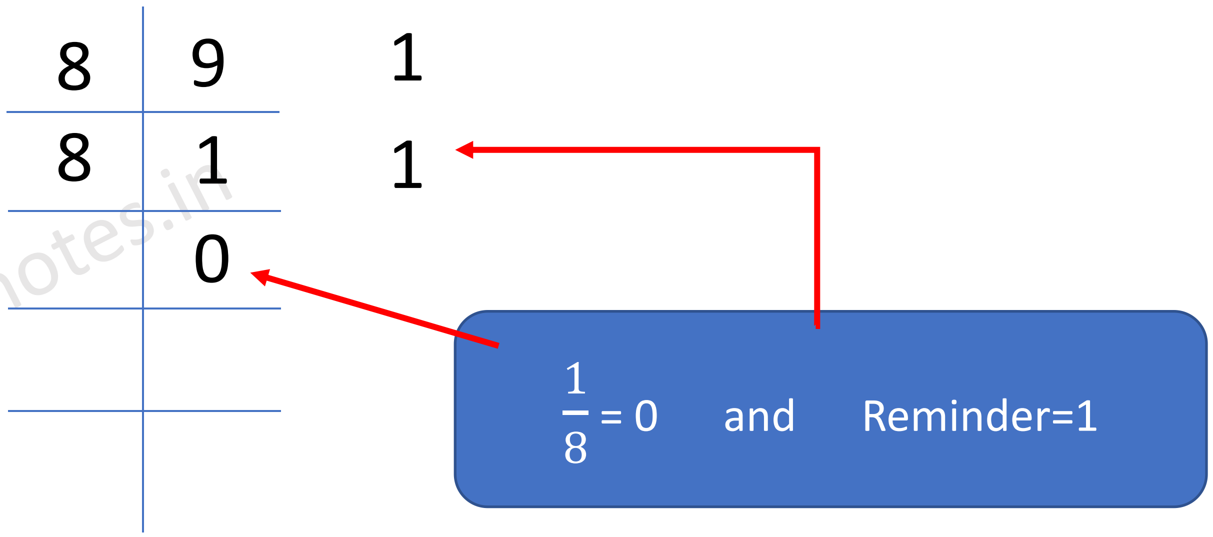
8 | 9   1

1

$\frac{9}{8} = 1$ and Reminder=1

Decimal to Octal

Decimal Number	Octal Number
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17



Decimal to Octal

Decimal Number	Octal Number
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17

Divisor	Dividend	Remainder
8	9	1
8	1	1
	0	

1 1

$(9)_{10} = (11)_8$

Decimal to Octal

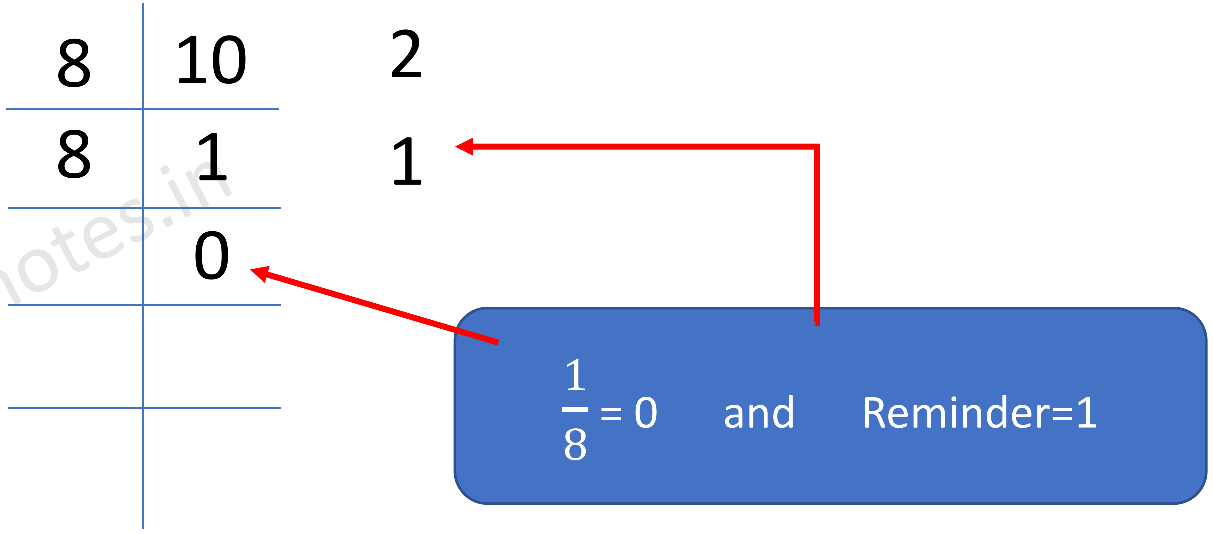
Decimal Number	Octal Number
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17

8	10	2
8	1	

$\frac{10}{8} = 1$ and Reminder=2

Decimal to Octal

Decimal Number	Octal Number
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17



Decimal to Octal

Decimal Number	Octal Number
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17

8	10	2
8	1	1
	0	

1 2

$(10)_{10} = (12)_8$

Decimal to Octal

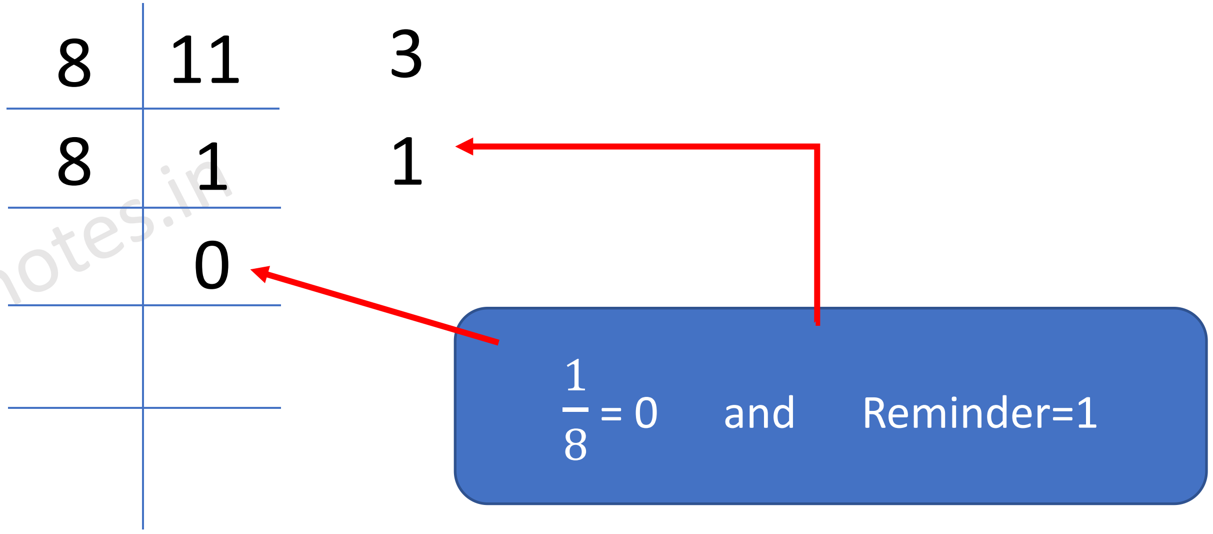
Decimal Number	Octal Number
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17

8	11	3
8	1	

$\frac{10}{8} = 1$ and Reminder=3

Decimal to Octal

Decimal Number	Octal Number
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17



Decimal to Octal

Decimal Number	Octal Number
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17

8	11	3
8	1	1
	0	

1 3

$(11)_{10} = (13)_8$

Converting Octal Number to Decimal Number

$(177)_8 = (?)_{10}$

2	1	0
1	7	7

$$8^0 \times 7 = 7$$

$(177)_8 = (?)_{10}$

2	1	0
1	7	7

$8^0 \times 7 = 7$

$8^1 \times 7 = 56$

$(177)_8 = (?)_{10}$

2	1	0
1	7	7

$$8^0 \times 7 = 7$$

$$8^1 \times 7 = 56$$

$$8^2 \times 1 = 64$$

$(177)_8 = (?)_{10}$

2	1	0
1	7	7

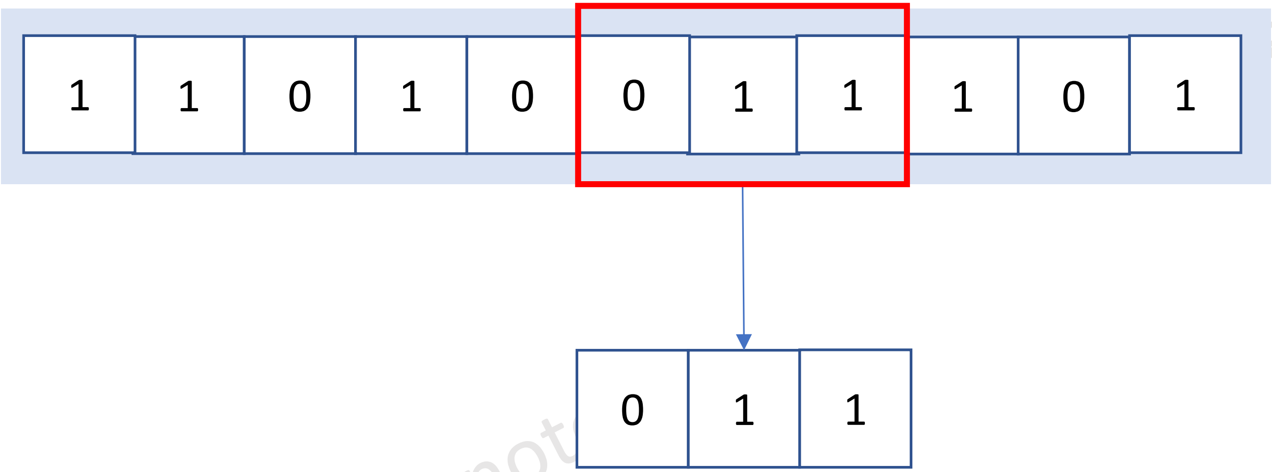
$$8^0 \times 7 = 7$$
$$8^1 \times 7 = 56$$
$$8^2 \times 1 = 64$$
$$\Sigma = 127$$

$(177)_8 = (127)_{10}$

Converting Binary Number to Octal Number

$(11010011101)_2 = (?)_8$

$(11010011101)_2 = (?)_8$



$(11010011101)_2 = (?)_8$

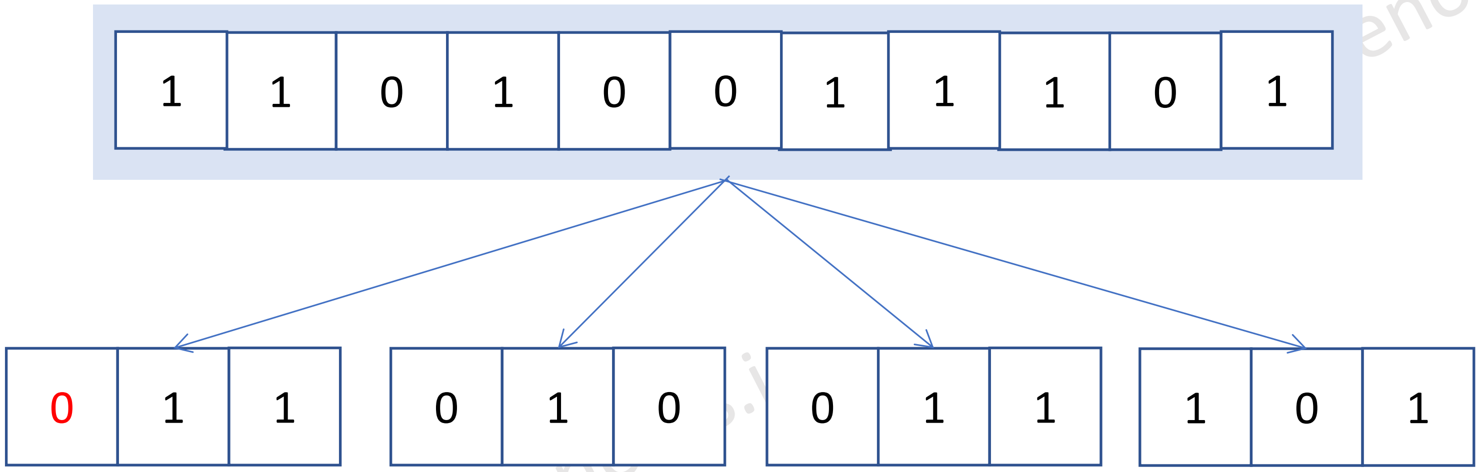
1	1	0	1	0	0	1	1	1	0	1

0	1	0

$(11010011101)_2 = (?)_8$

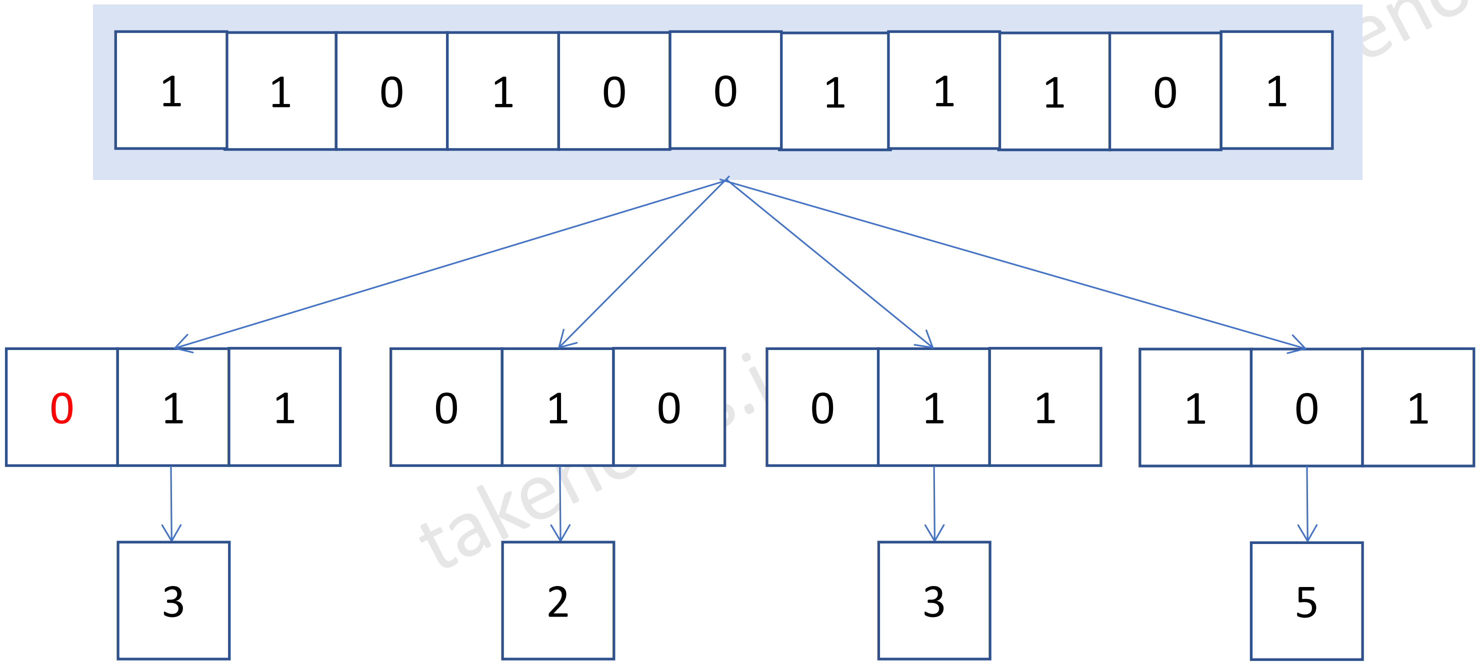
$(11010011101)_2 = (?)_8$

3 bit binary Numer	Octal Number
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7



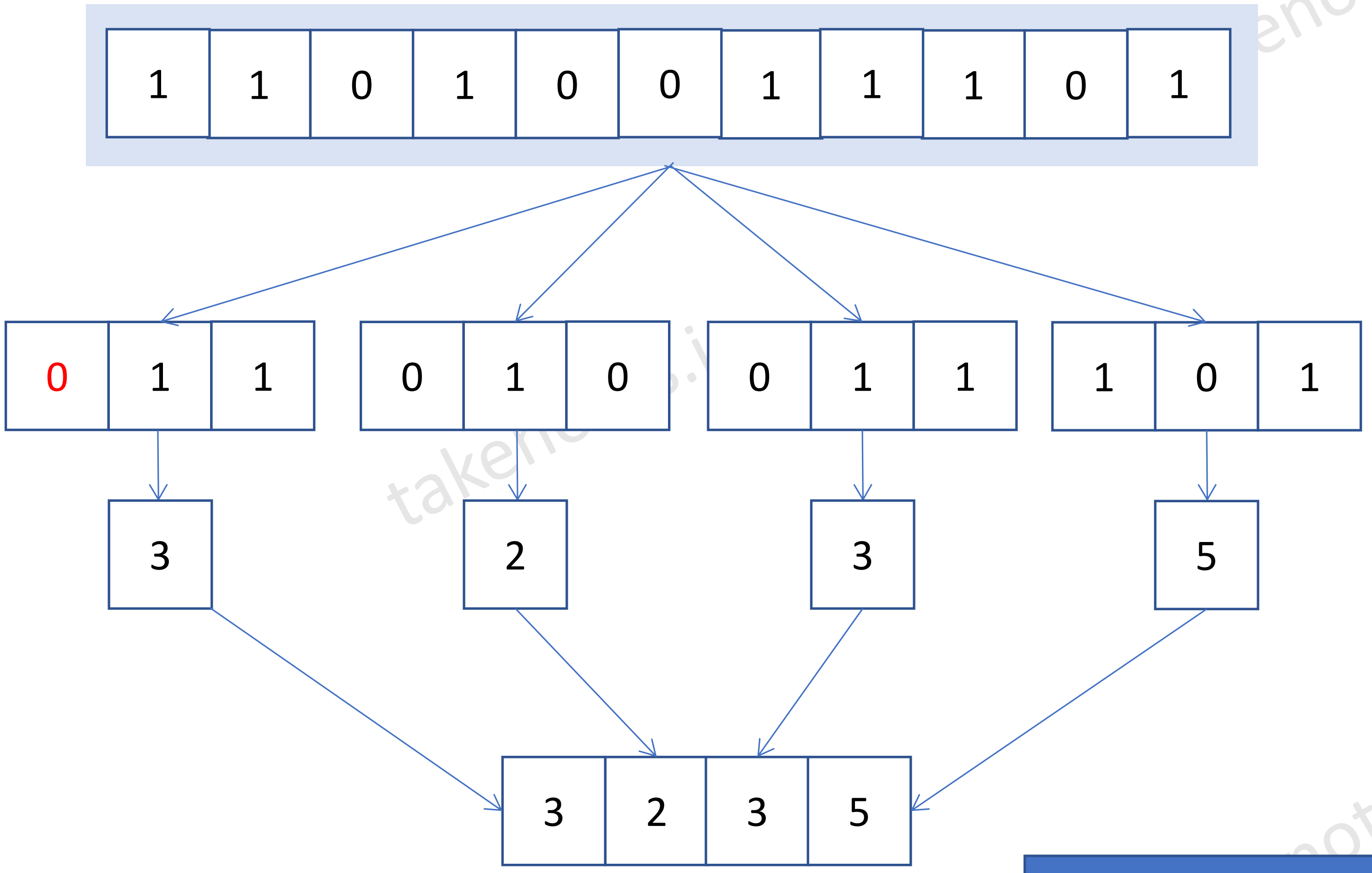
$(11010011101)_2 = (?)_8$

3 bit binary Numer	Octal Number
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7



$(11010011101)_2 = (?)_8$

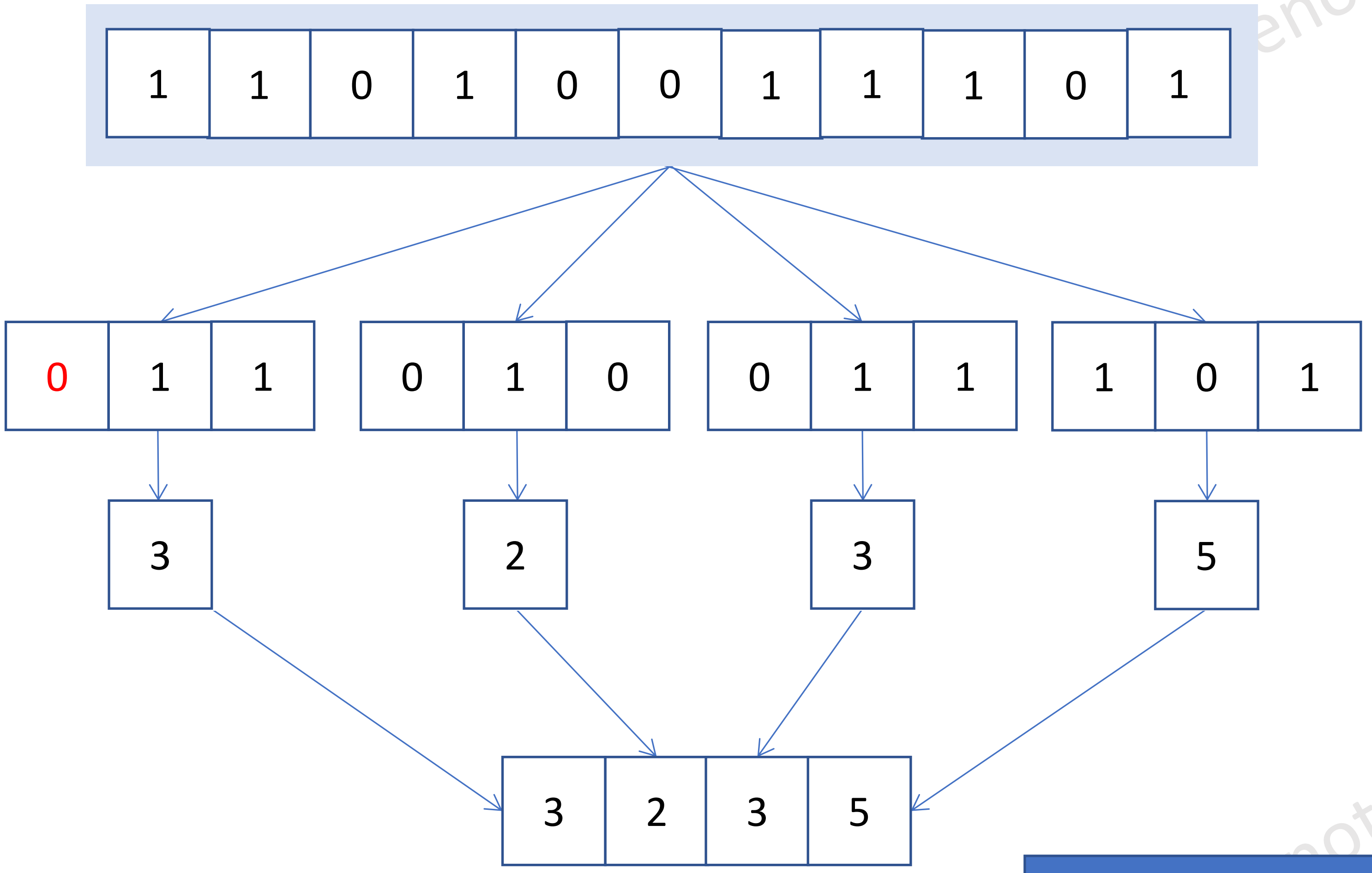
3 bit binary Numer	Octal Number
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7



$(11010011101)_2 = (3235)_8$

$(11010011101)_2 = (?)_8$

3 bit binary Numer	Octal Number
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

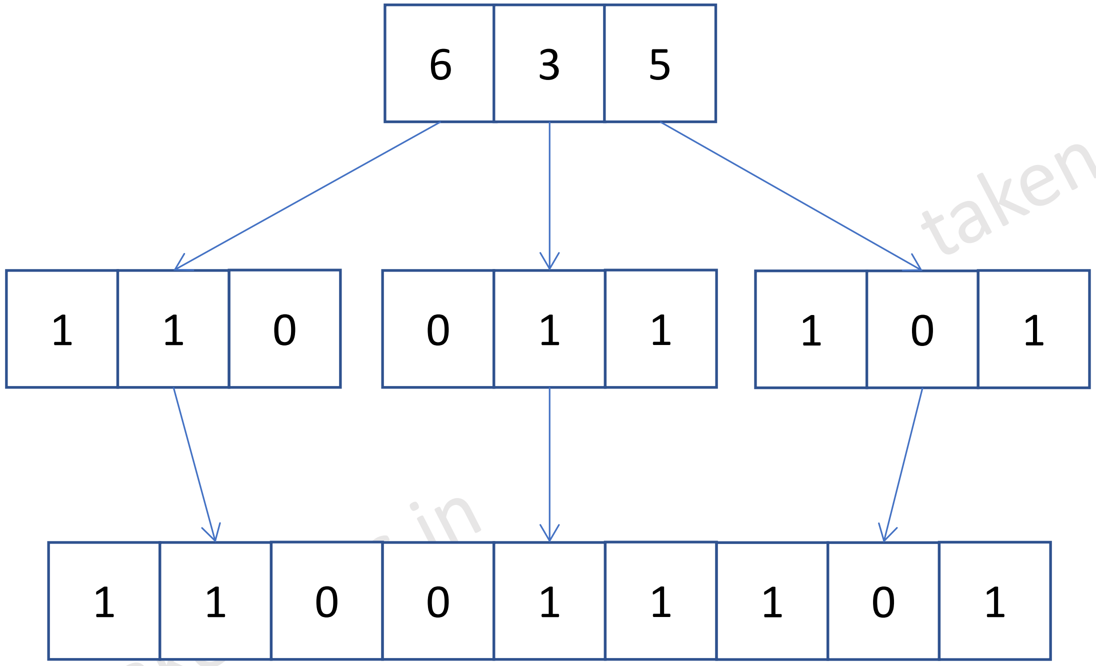


$(11010011101)_2 = (3235)_8$

Converting Octal Number to Binary Number

$(635)_8 = (?)_2$

3 bit binary Numer	Octal Number
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7



$(635)_8 = (110011101)_2$

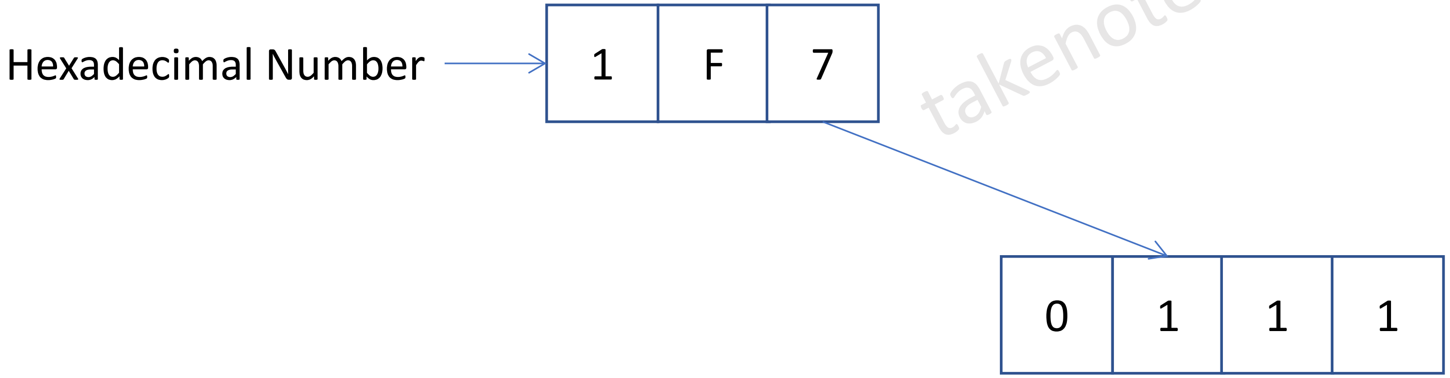
Converting Hexadecimal Number to Octal Number

Decimal Number	Binary Number	Hexadecimal Number	Octal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17

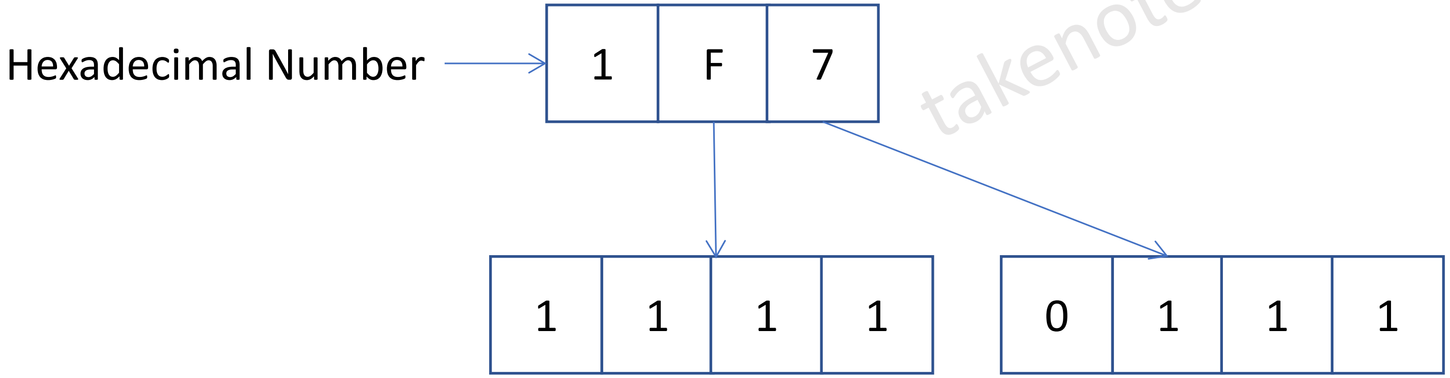
Hexadecimal Number →

1	F	7

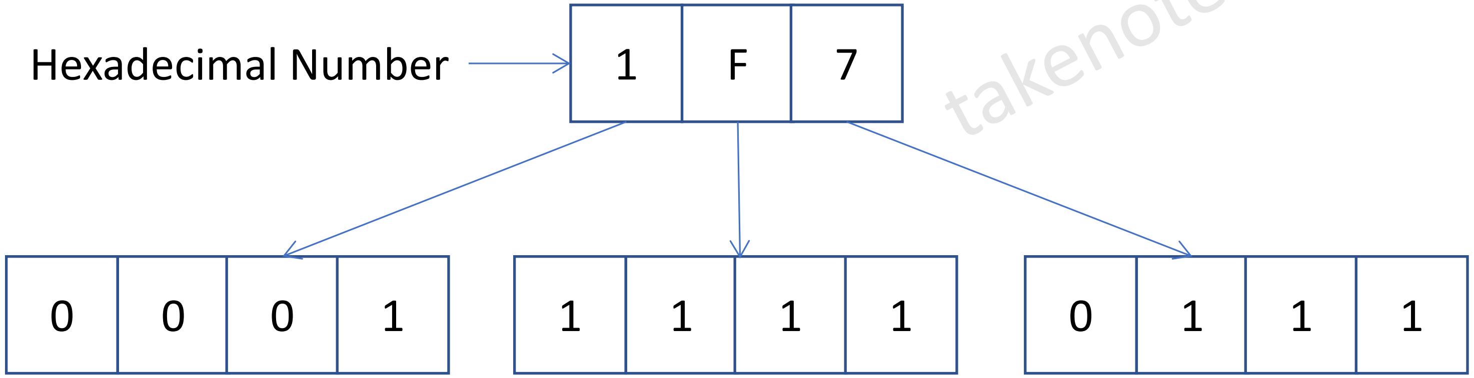
Decimal Number	Binary Number	Hexadecimal Number	Octal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17



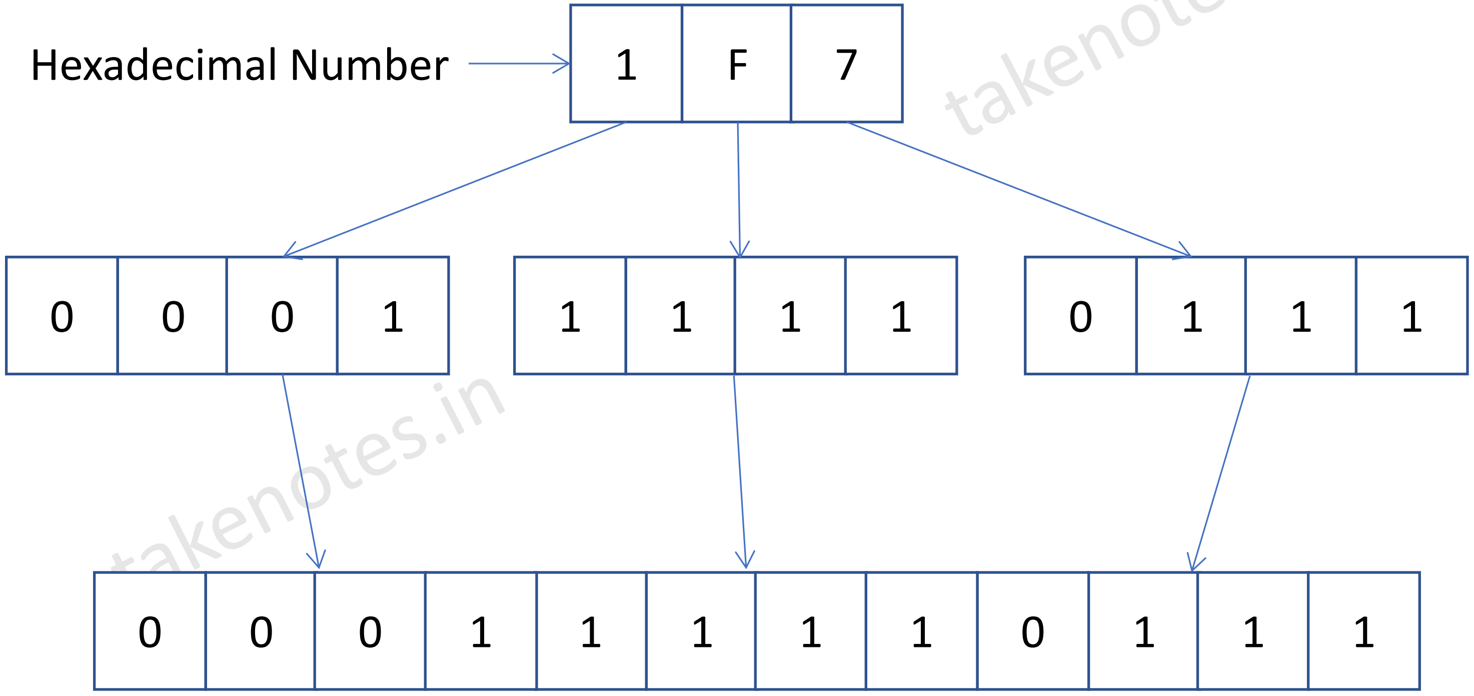
Decimal Number	Binary Number	Hexadecimal Number	Octal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17



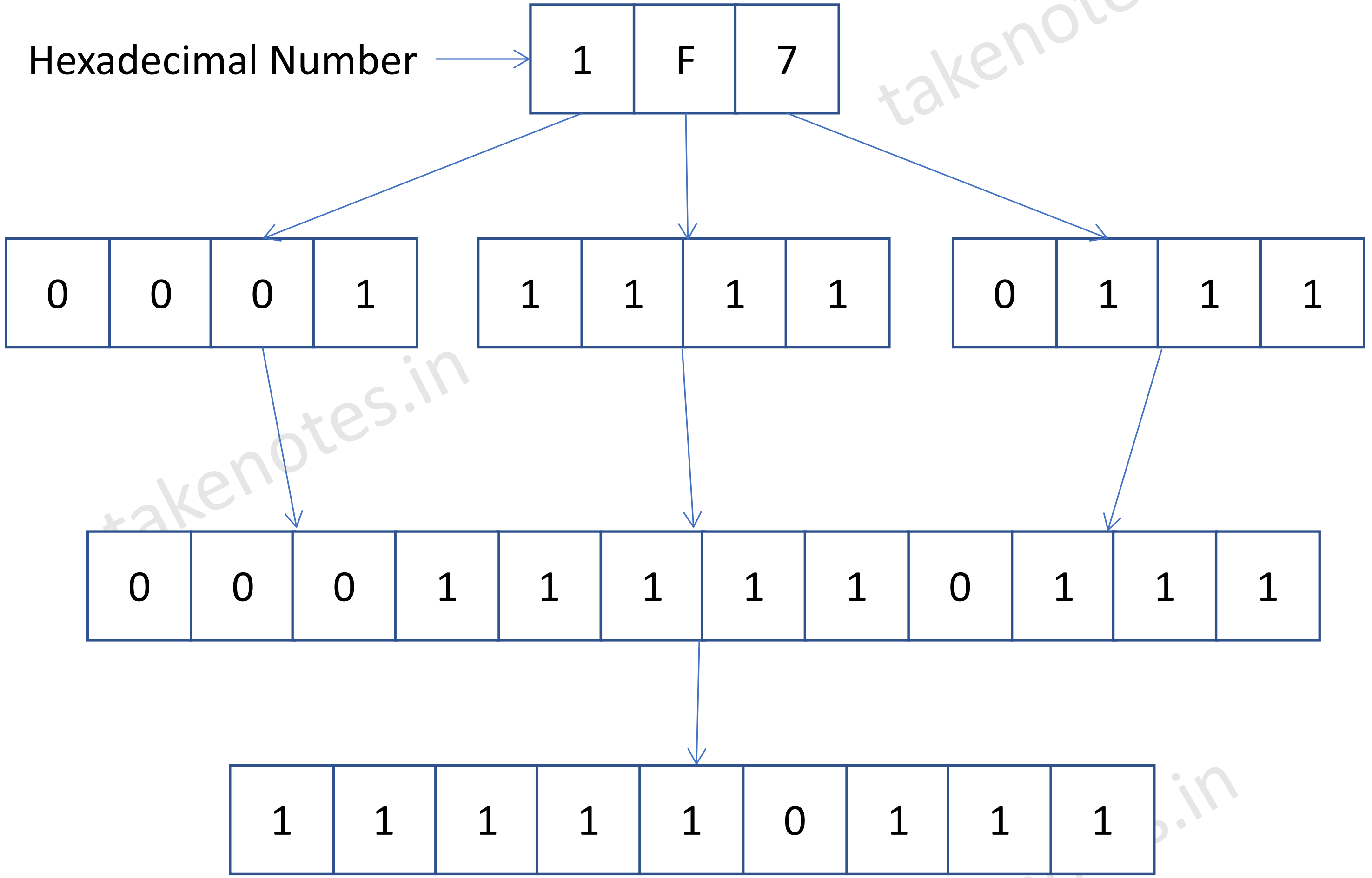
Decimal Number	Binary Number	Hexadecimal Number	Octal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17



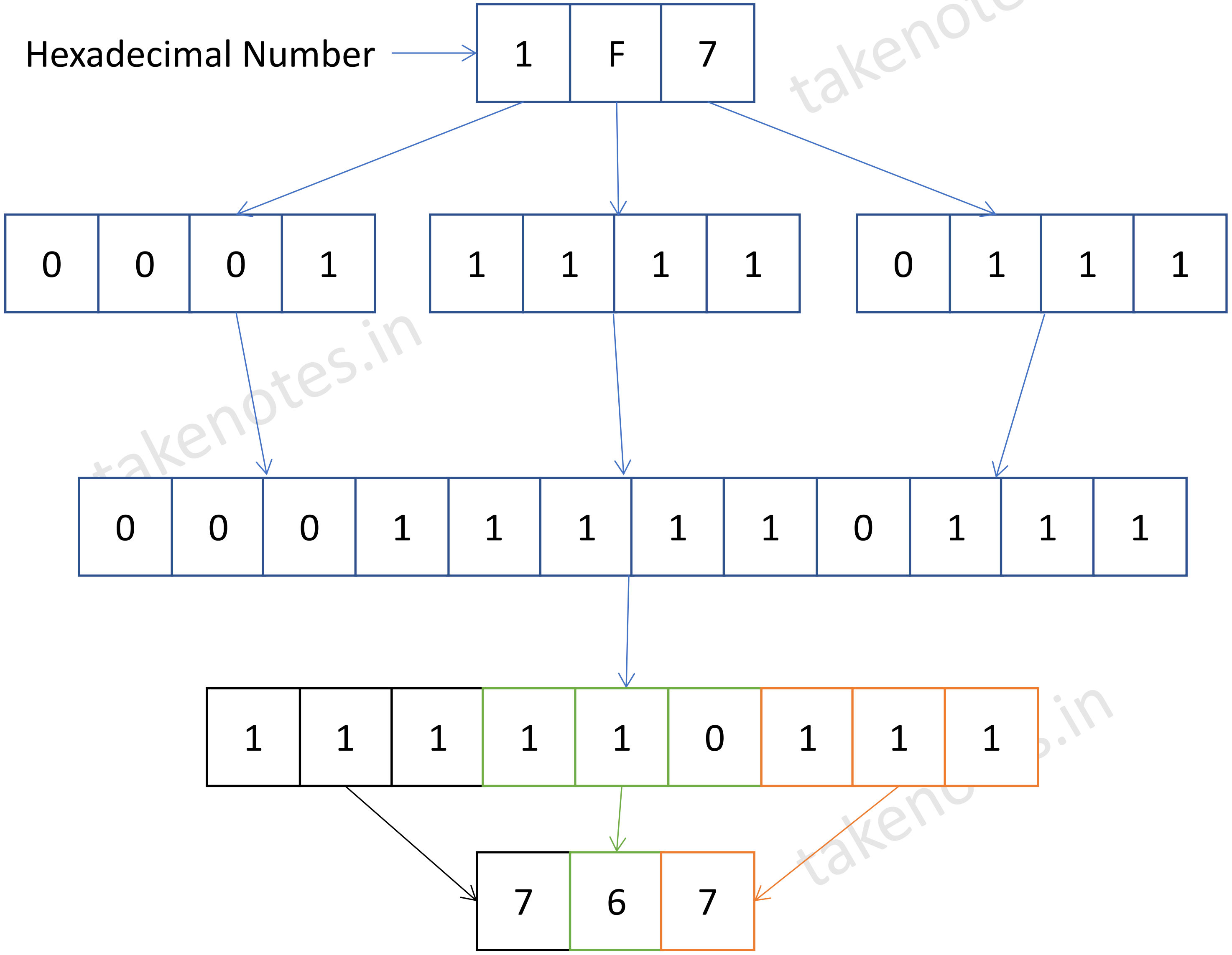
Decimal Number	Binary Number	Hexadecimal Number	Octal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17



Decimal Number	Binary Number	Hexadecimal Number	Octal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17



Decimal Number	Binary Number	Hexadecimal Number	Octal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17

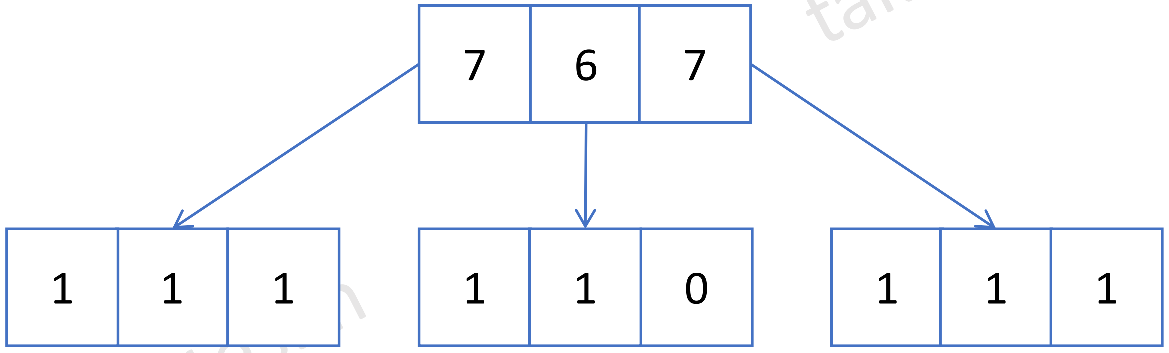


Converting Octal Number to Hexadecimal Number

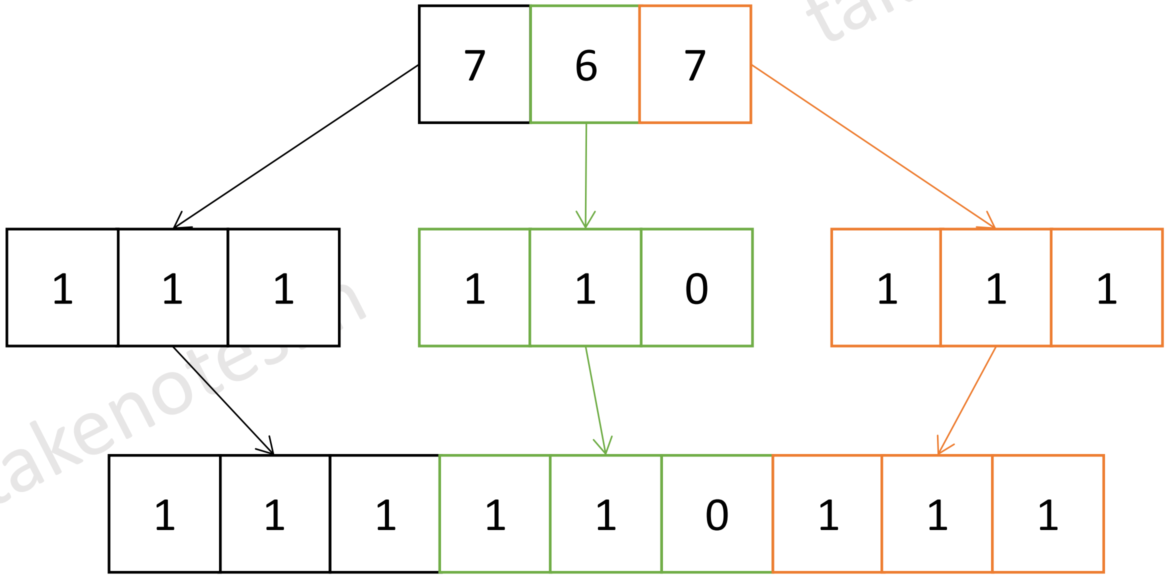
Decimal Number	Binary Number	Hexadecimal Number
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

7	6	7

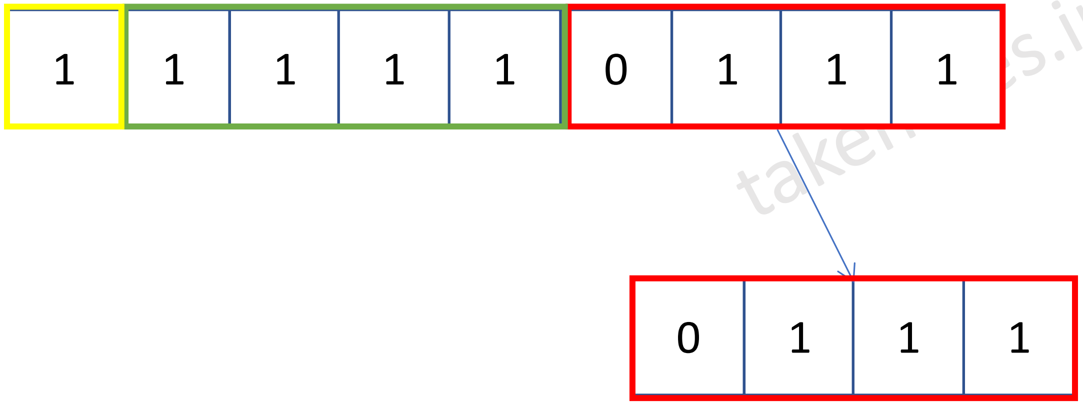
Decimal Number	Binary Number	Hexadecimal Number
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F



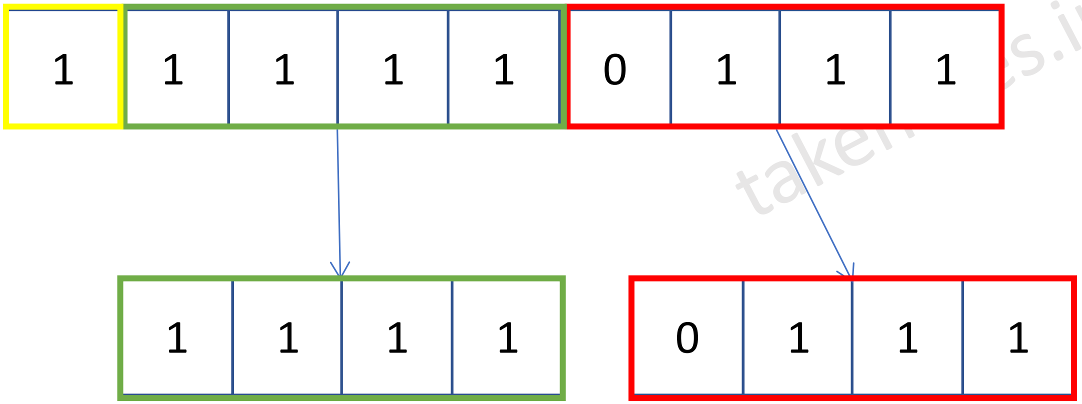
Decimal Number	Binary Number	Hexadecimal Number
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F



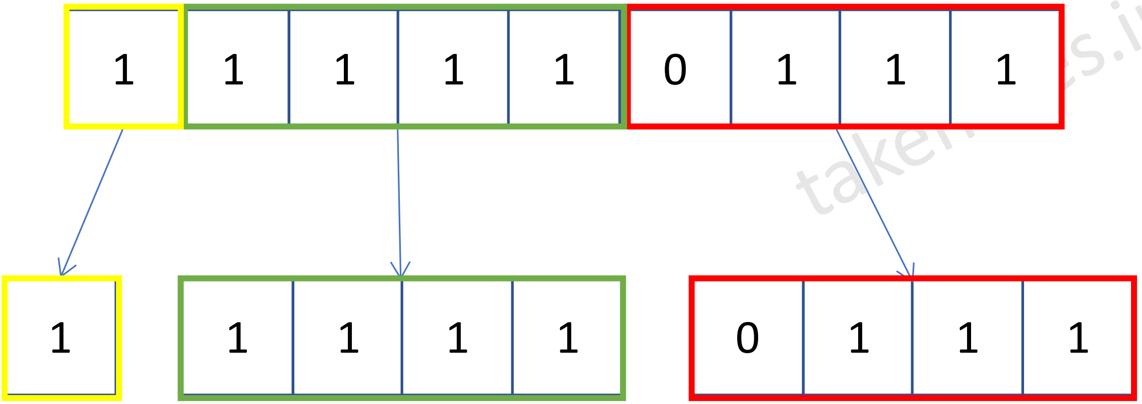
Decimal Number	Binary Number	Hexadecimal Number
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F



Decimal Number	Binary Number	Hexadecimal Number
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
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

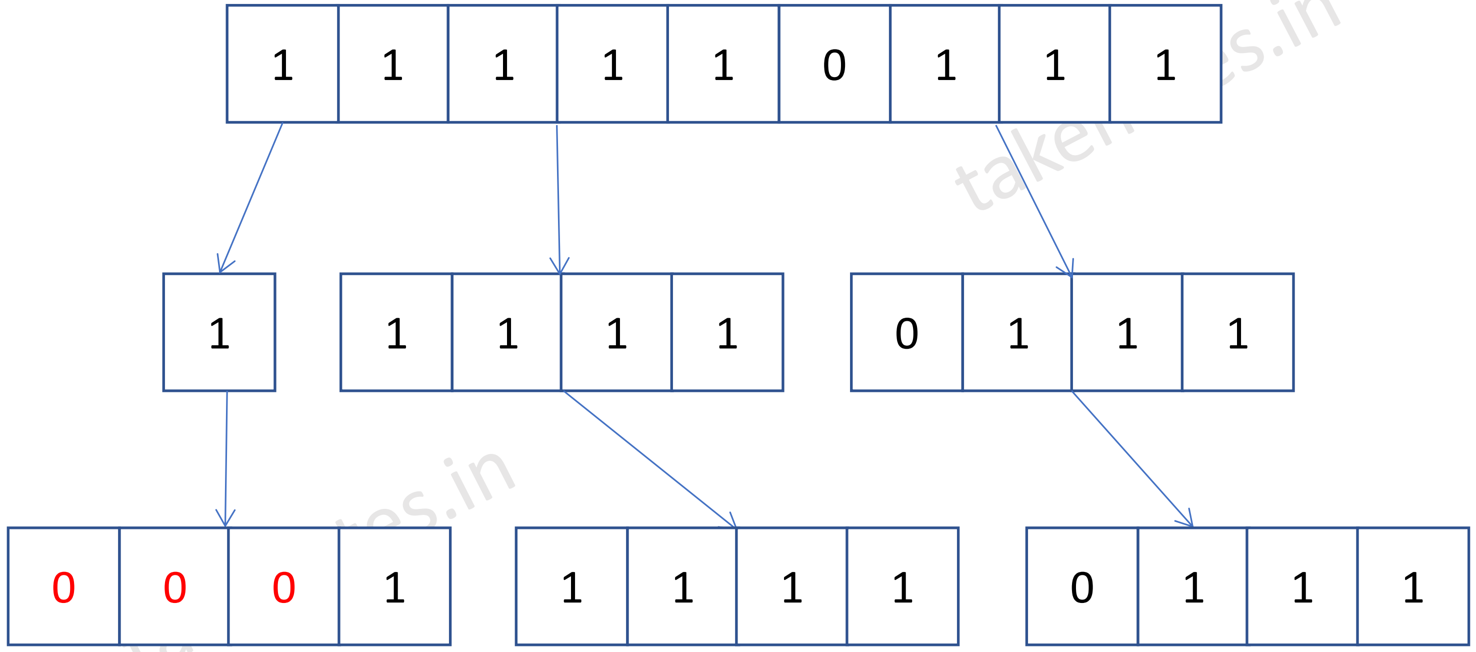


Decimal Number	Binary Number	Hexadecimal Number
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

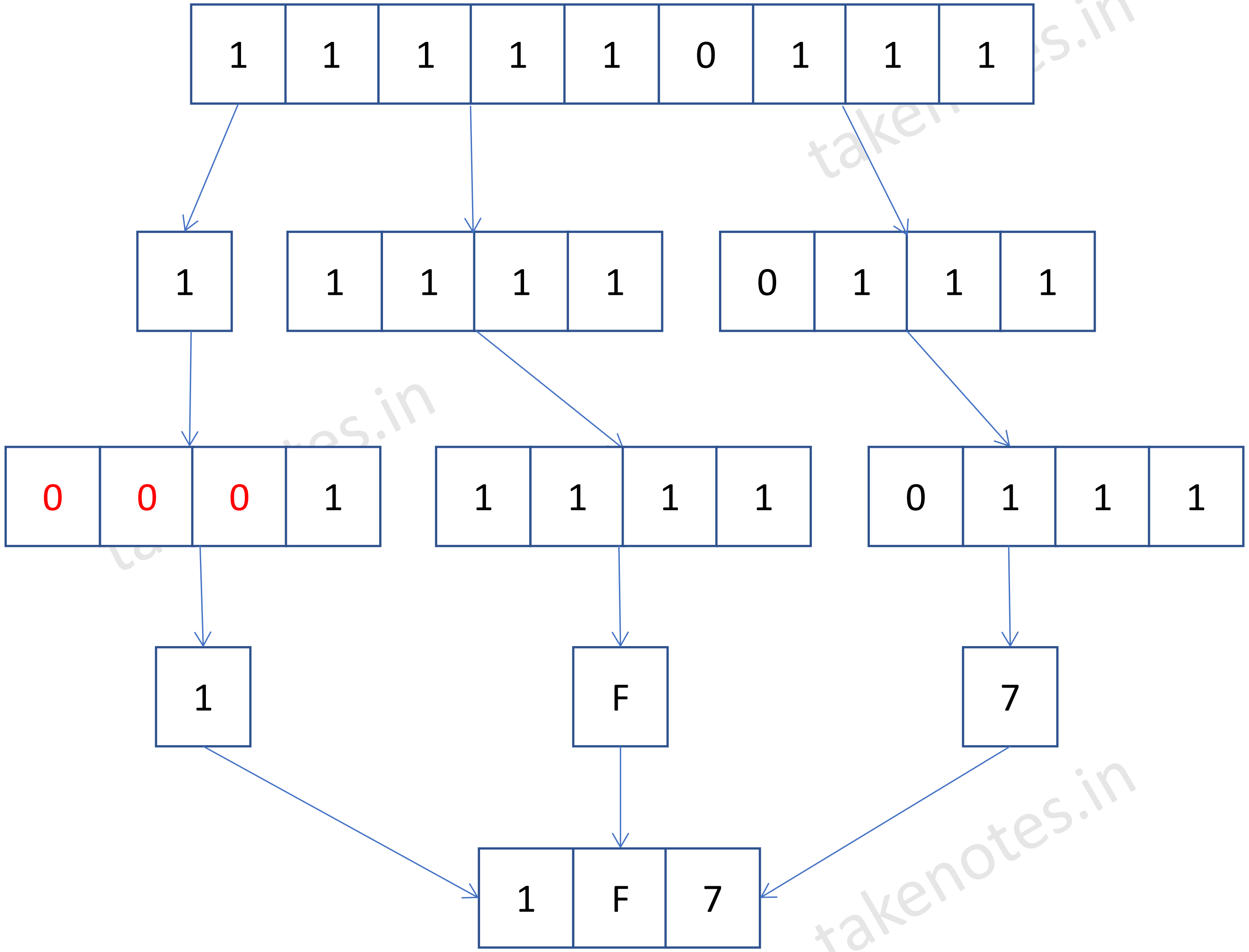


takenotes.in

Decimal Number	Binary Number	Hexadecimal Number
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
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

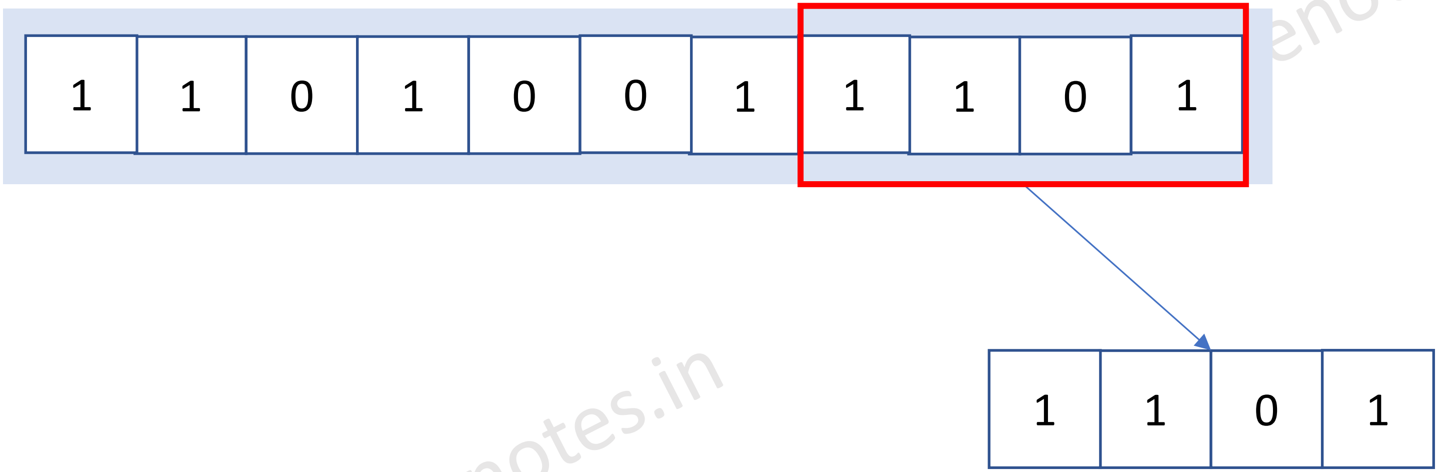


Decimal Number	Binary Number	Hexadecimal Number
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

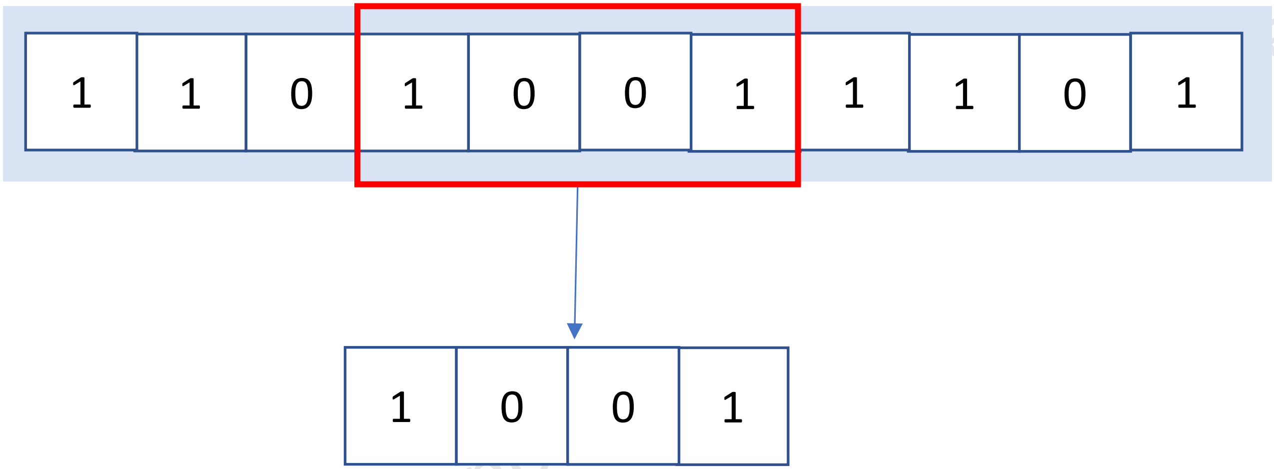


Converting Binary Number to Hexadecimal Number

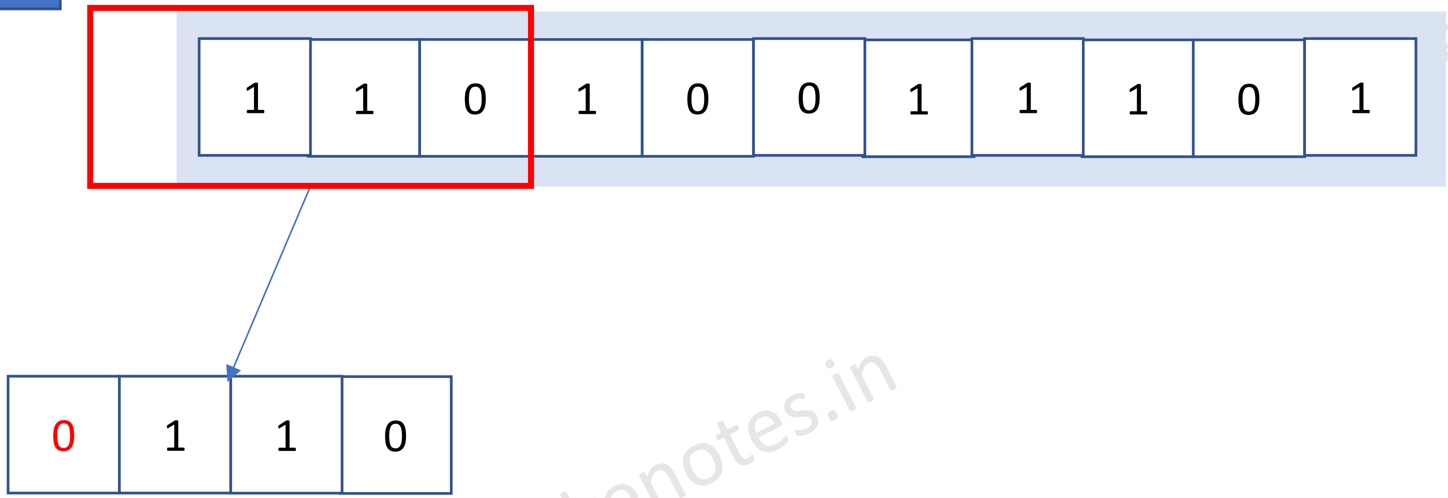
$(11010011101)_2 = (?)_{16}$



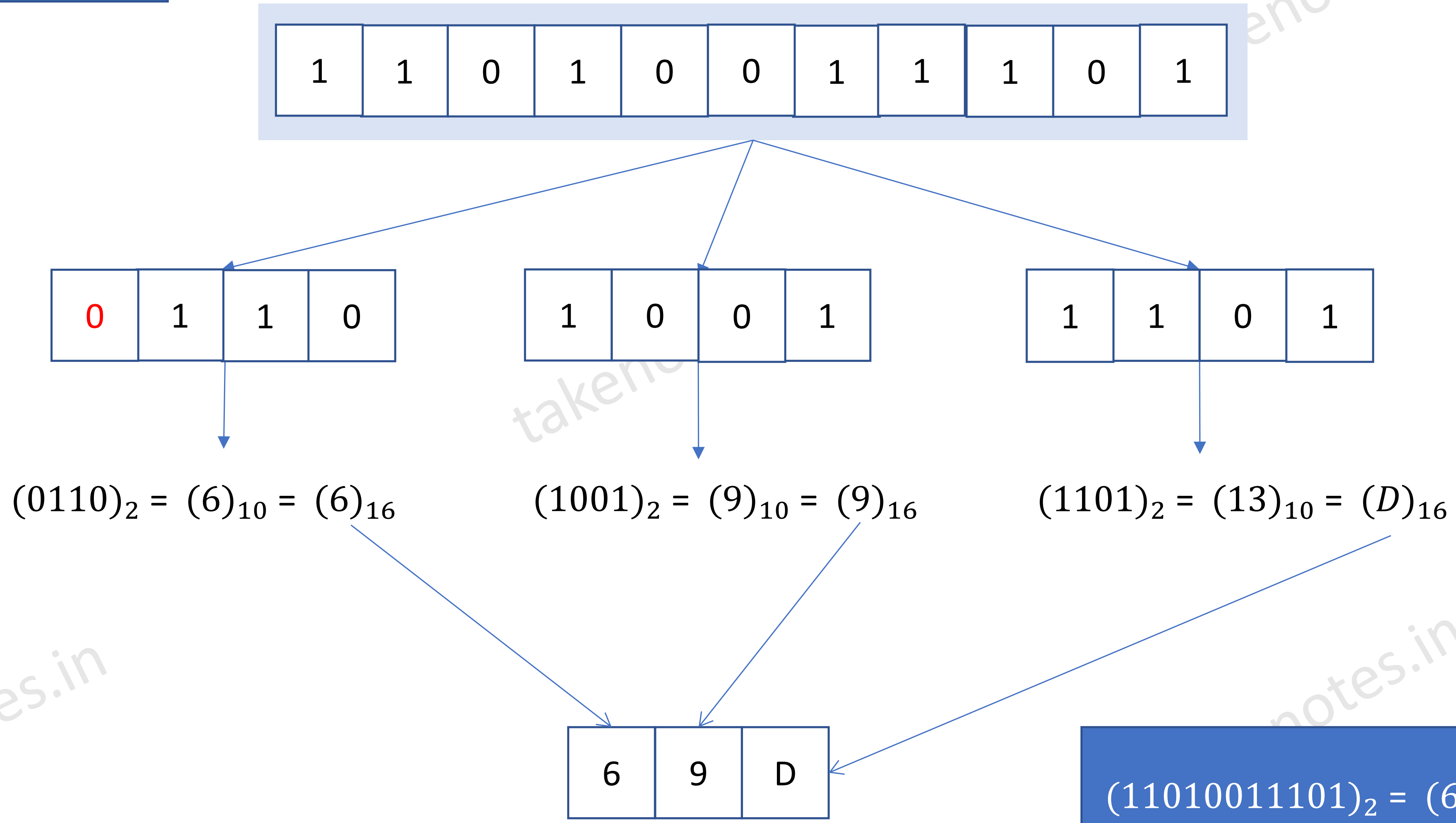
$(11010011101)_2 = (?)_{16}$



$(11010011101)_2 = (?)_{16}$

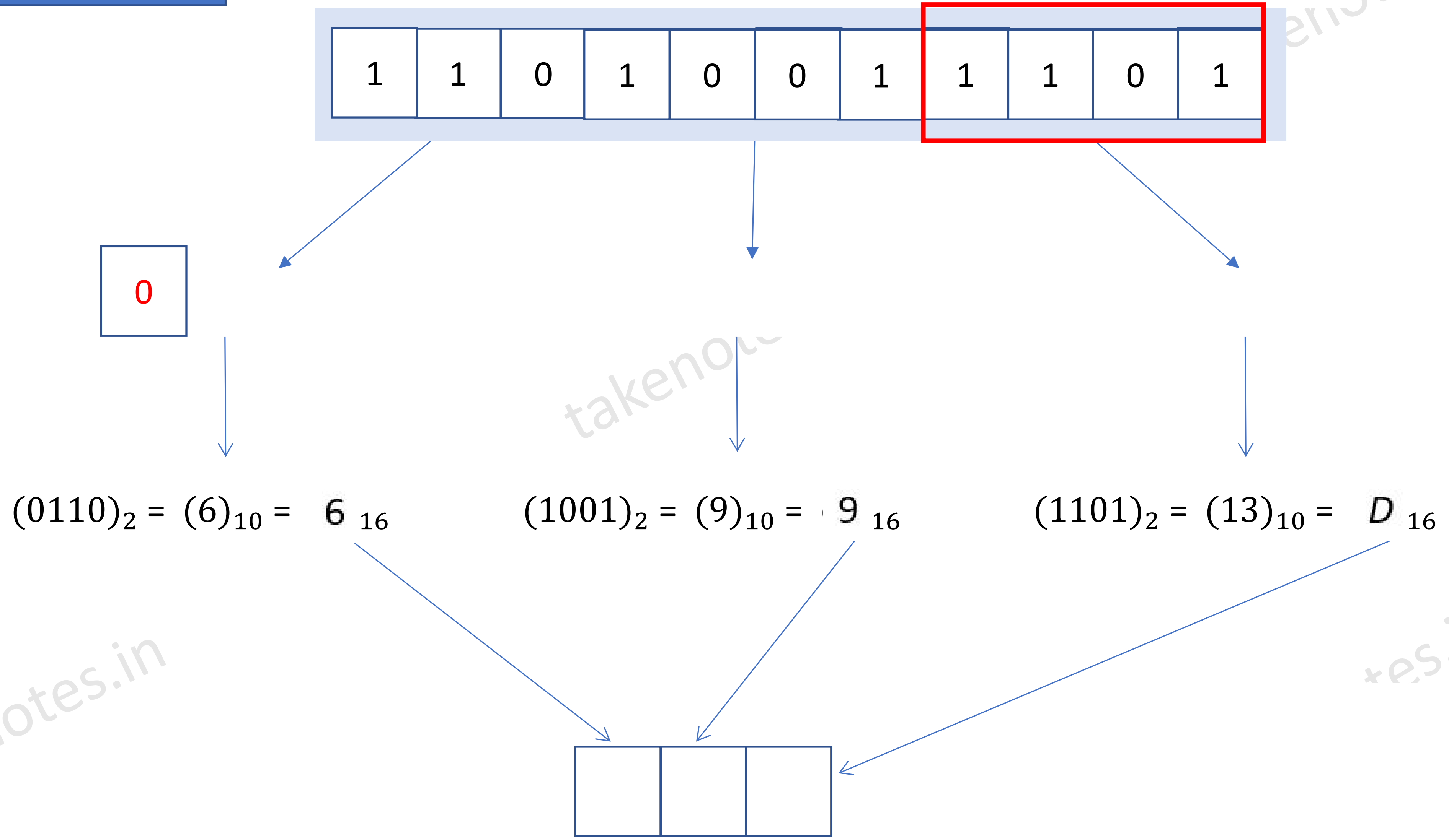


$(11010011101)_2 = (?)_{16}$



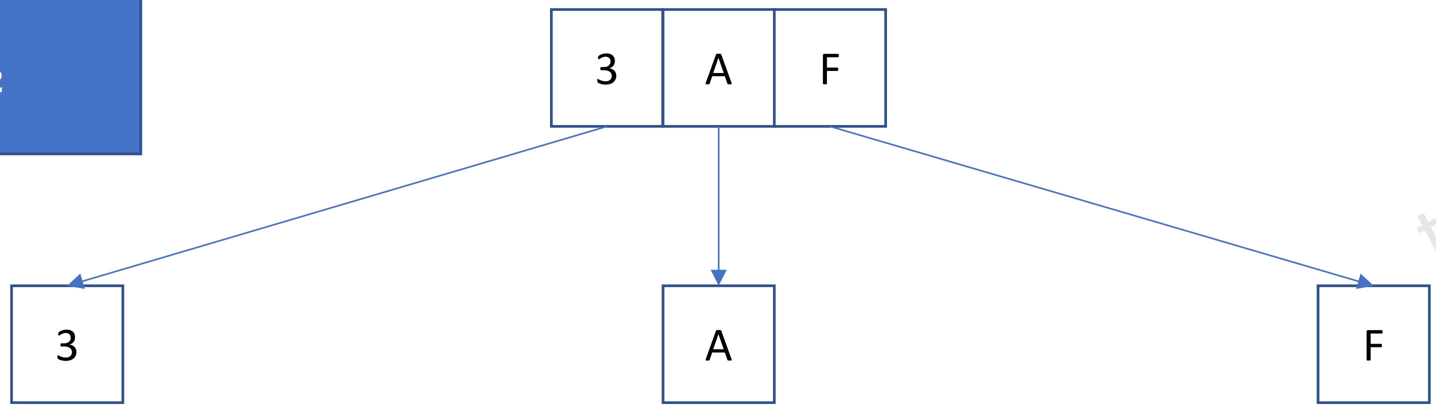
$(11010011101)_2 = (69D)_{16}$

$(11010011101)_2 = (?)_{16}$

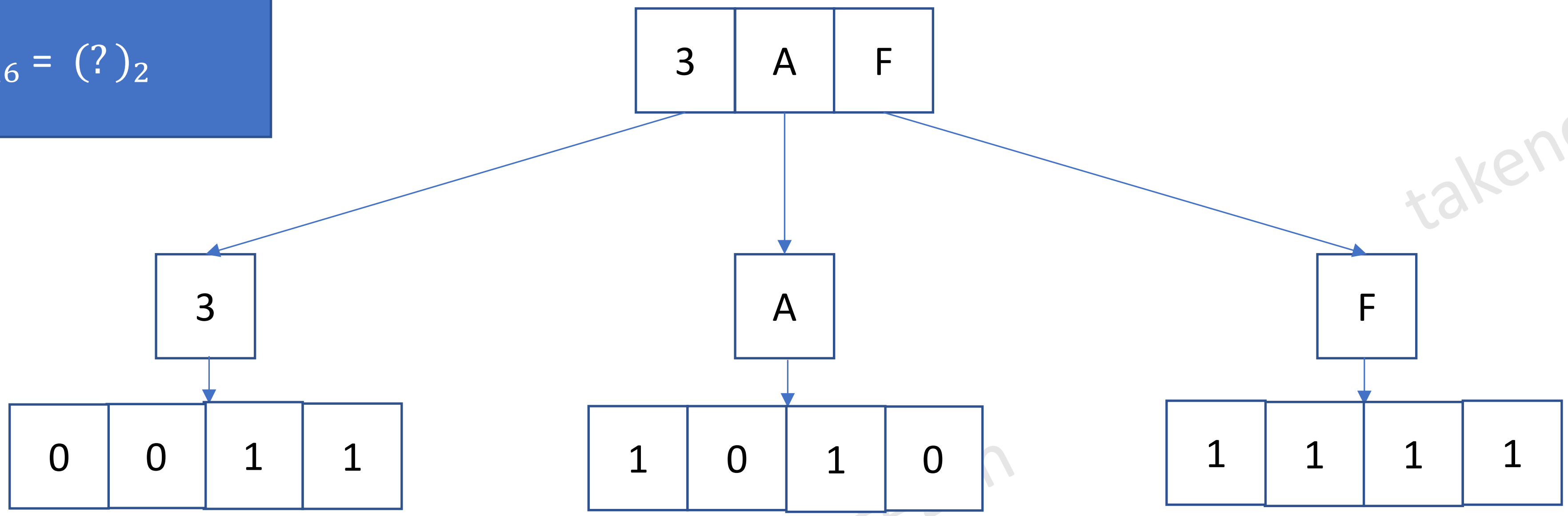


Converting Hexadecimal Number to Binary Number

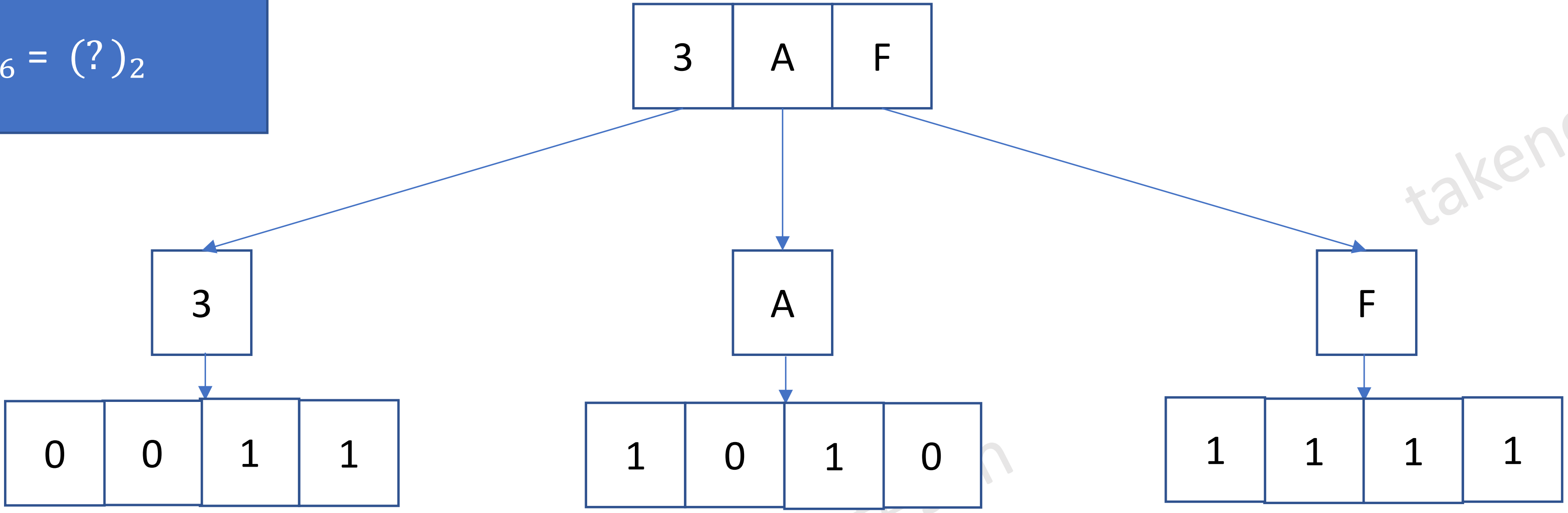
$(3AF)_{16} = (?)_2$



$(3AF)_{16} = (?)_2$

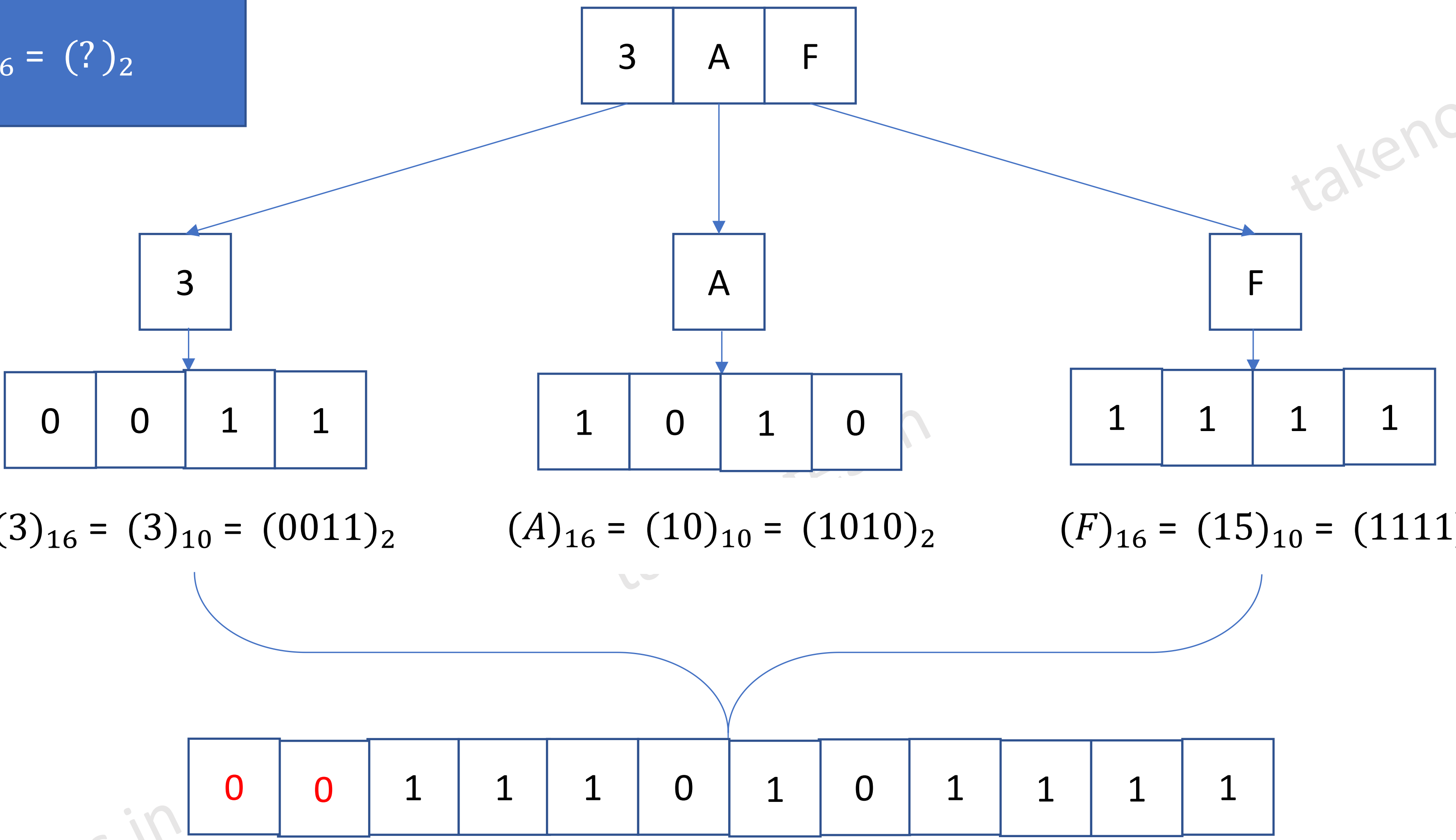


$(3AF)_{16} = (?)_2$



$(3)_{16} = (3)_{10} = (0011)_2$ $(A)_{16} = (10)_{10} = (1010)_2$ $(F)_{16} = (15)_{10} = (1111)_2$

$(3AF)_{16} = (?)_2$



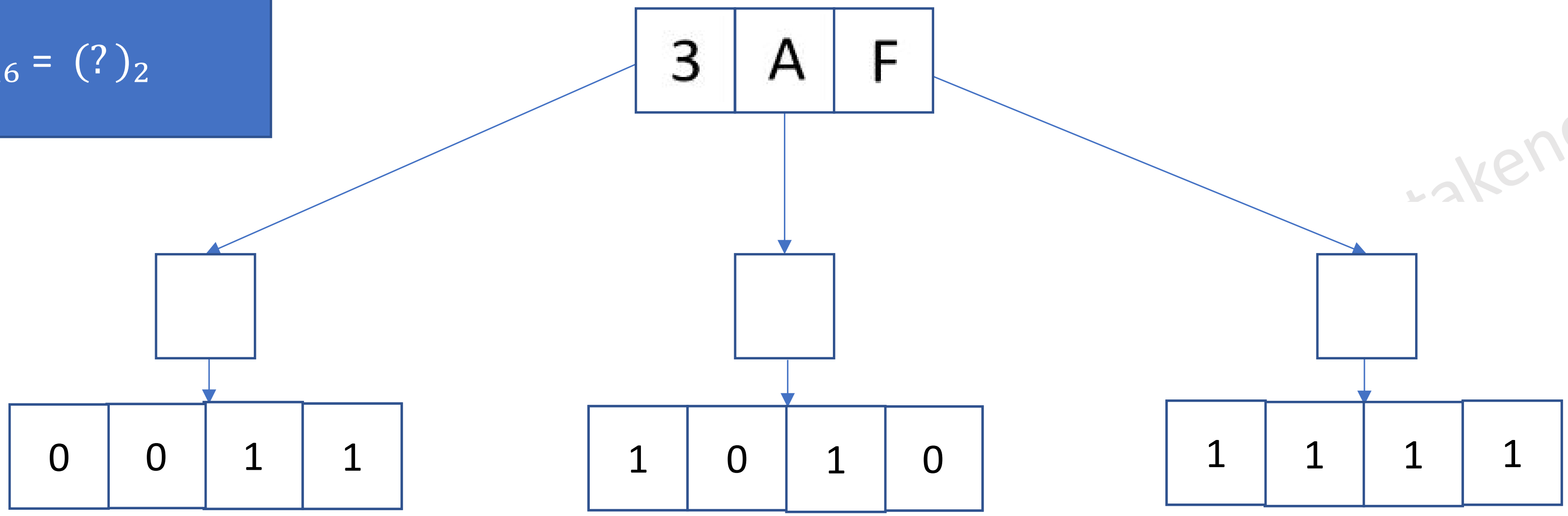
$(3)_{16} = (3)_{10} = (0011)_2$

$(A)_{16} = (10)_{10} = (1010)_2$

$(F)_{16} = (15)_{10} = (1111)_2$

$(3AF)_{16} = (1110101111)_2$

$(3AF)_{16} = (?)_2$



$(3)_{16} = (3)_{10} = (0011)_2$ $(A)_{16} = (10)_{10} = (1010)_2$ $(F)_{16} = (15)_{10} = (1111)_2$

Converting Hexadecimal Number to Decimal Number

$(3F)_{16} = (?)_{10}$

1	0
3	F

$(3F)_{16} = (?)_{10}$

1	0
3	F

$(F)_{16} = (15)_{10}$

$16^0 \times 15 = 15$

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

$(3F)_{16} = (?)_{10}$

1	0
3	F

$(F)_{16} = (15)_{10}$

$$16^0 \times 15 = 15$$

$(3)_{16} = (3)_{10}$

$$16^1 \times 3 = 48$$

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

$(3F)_{16} = (?)_{10}$

1	0
3	F

$(F)_{16} = (15)_{10}$

$(3)_{16} = (3)_{10}$

$$16^0 \times 15 = 15$$

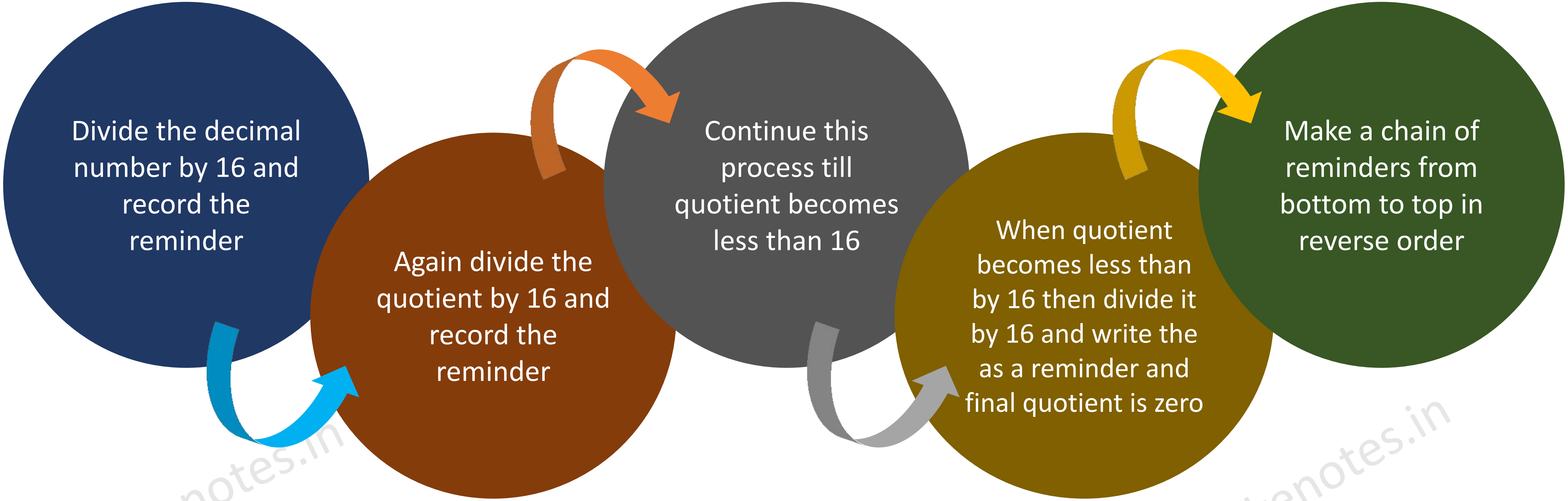
$$16^1 \times 3 = 48$$

$$\Sigma = 63$$

$(3F)_{16} = (63)_{10}$

Converting Decimal Number to Hexadecimal Number

Steps


Divide the decimal number by 16 and record the reminder
Again divide the quotient by 16 and record the reminder
Continue this process till quotient becomes less than 16
When quotient becomes less than by 16 then divide it by 16 and write the as a reminder and final quotient is zero
Make a chain of reminders from bottom to top in reverse order

$(251)_{10} = (?)_{16}$

$$\begin{array}{c|c l}
16 & 251 & B \\
\hline
 & 15 & \\
\hline
 & & \\
\hline
 & & \\
\hline
 & &
\end{array}$$

$\frac{251}{16} = 15$ and Reminder=11

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	$\boxed{\textbf{B}}$	**C**	**D**	**E**	**F**

$(251)_{10} = (?)_{16}$

16	251	B
16	15	F
	0	

$\frac{15}{16} = 0$ and Reminder=15

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

$(251)_{10} = (?)_{16}$

16	251	B
16	15	F
	0	

$(251)_{10} = (FB)_{16}$

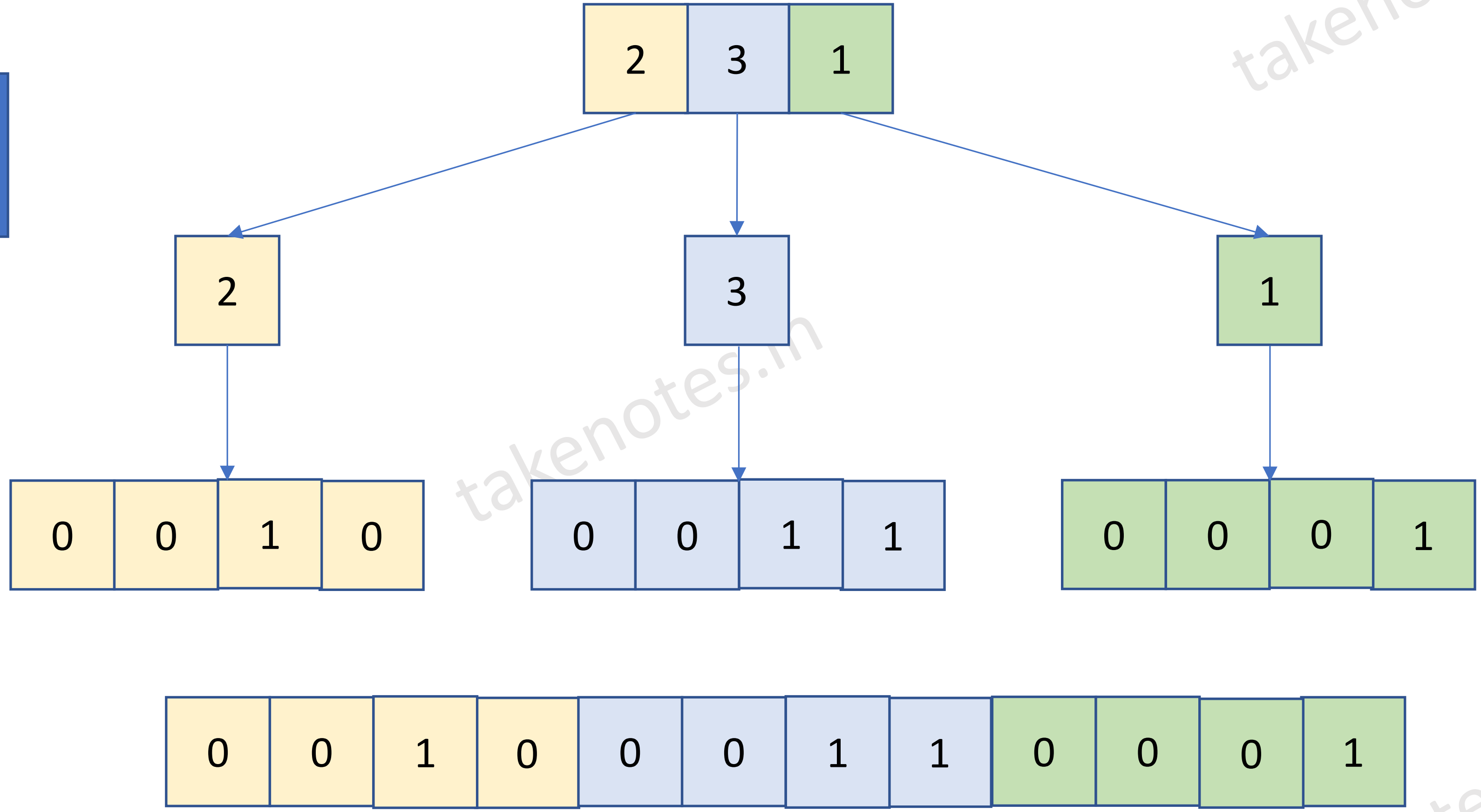
Binary Coded Decimal

Binary Coded Decimal

- We can write higher decimal numbers in the language of 1’s and 0’s
- We have to memories four bit binary equivalents of zero to nine decimal numbers

Decimal to BCD

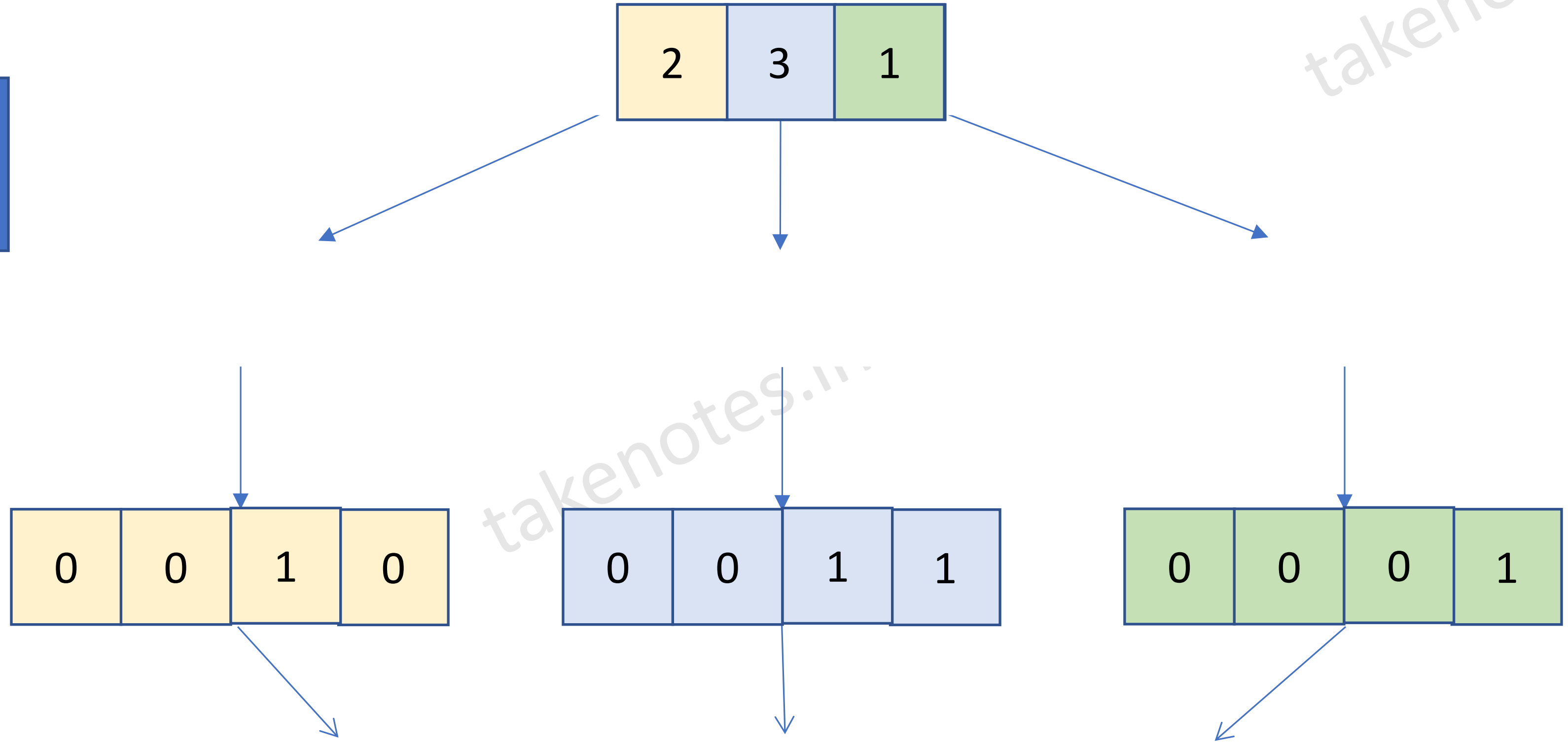
$(231)_{10} = (?)_{BCD}$



$(231)_{10} = (001000110001)_{BCD}$

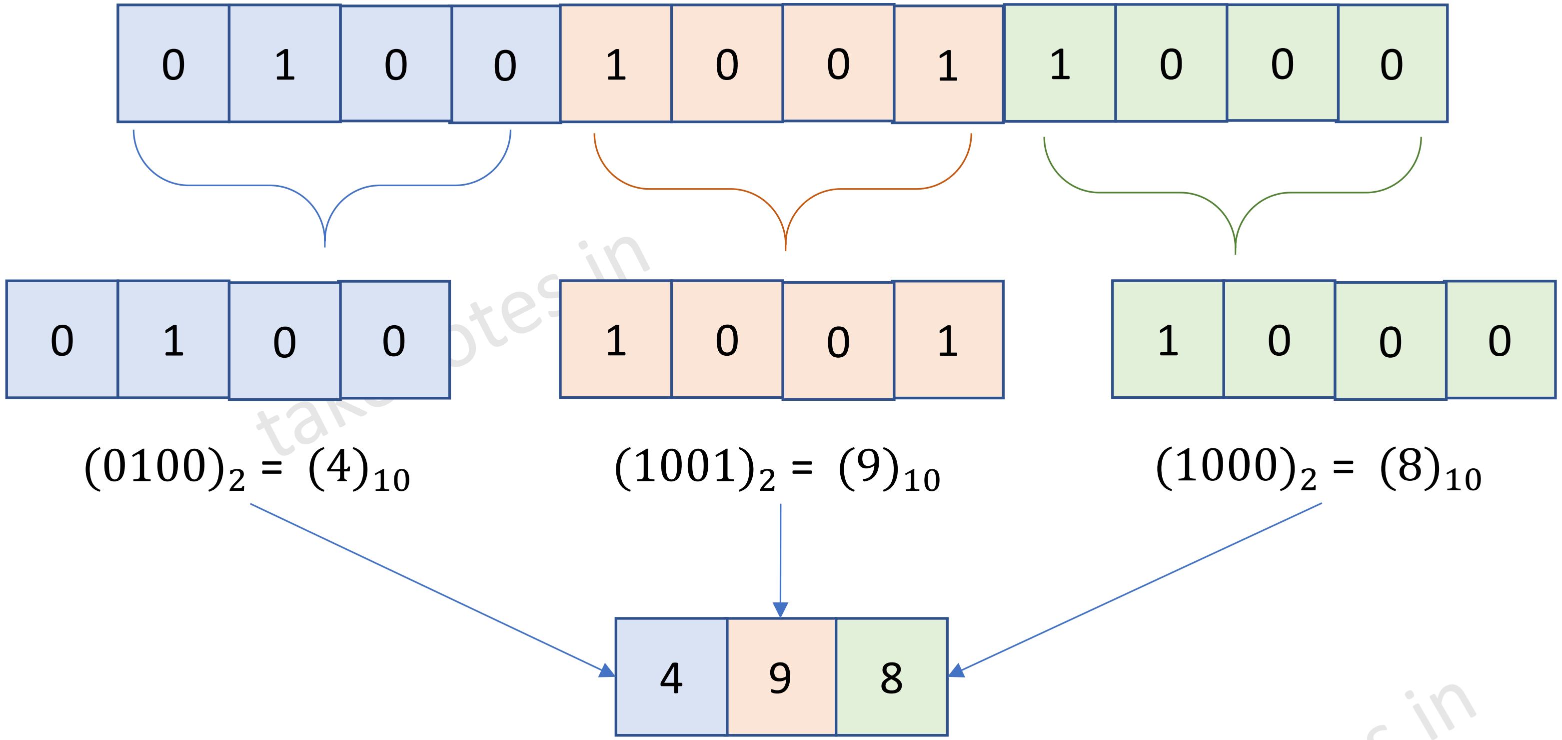
Decimal to BCD

$(231)_{10} = (?)_{BCD}$



BCD to Decimal

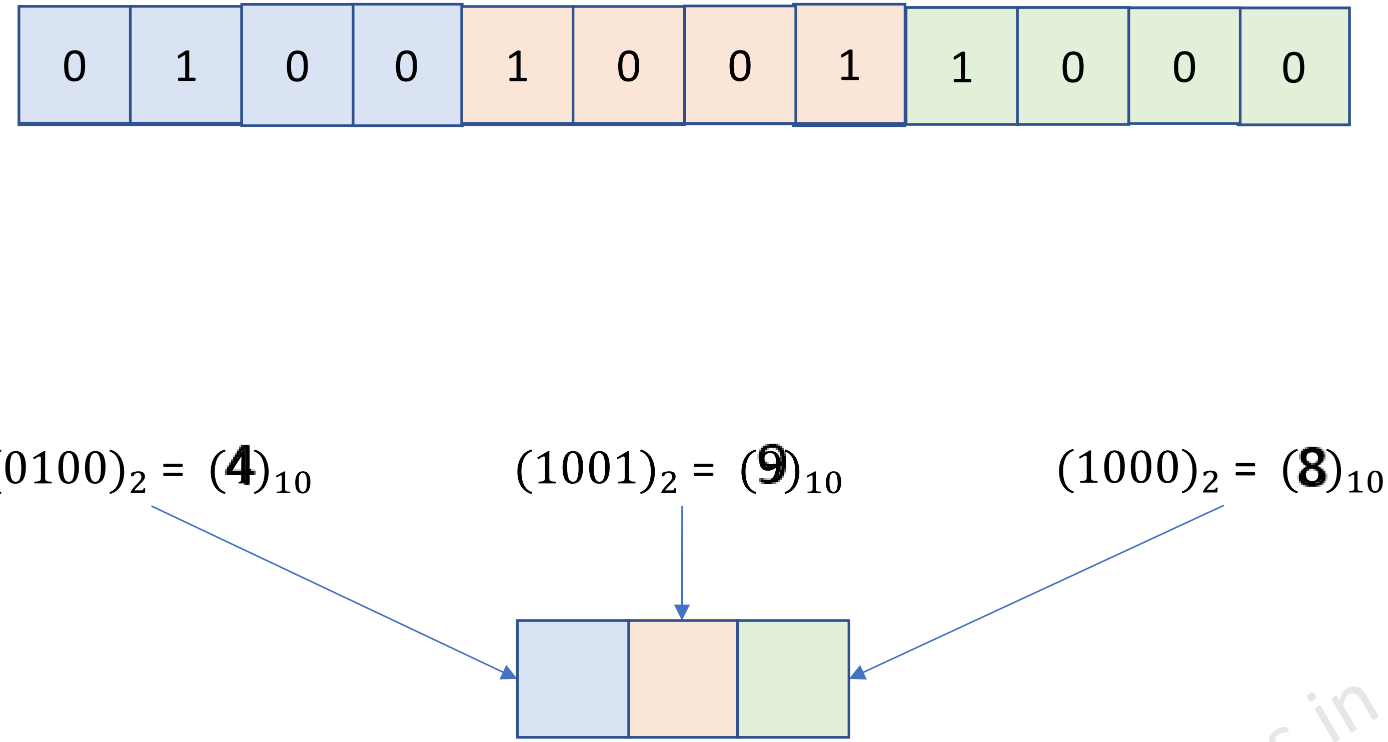
$$(010010011000)_{BCD} = (?)_{10}$$



$$(010010011000)_{BCD} = (498)_{10}$$

BCD to Decimal

$(010010011000)_{BCD} = (?)_{10}$



Binary Addition Rules

Binary Addition Rules

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 0
with carry 1 (10)

1 + 1 +1 = 1
with carry 1 (11)

Binary Addition Rules

$(1101)_2 + (1010)_2 = (?)_2$

$$
\begin{array}{rcccc}
 & 1 & 1 & 0 & 1 \\
+ & 1 & 0 & 1 & 0 \\
\hline
1 & 0 & 1 & 1 & 1
\end{array}
$$

carry → 1

1's Compliment and 2's Compliment

1's Compliment and 2's Compliment

1's Compliment	2's Compliment
Replace each 1 by 0 and each 0 by 1 $e.g.\ \ (1101)_2 \xrightarrow{\text{1's Compliment}} (0010)_2$	Find 1's compliment of given number and add 1 $e.g.\ \ (1101)_2 \xrightarrow{\text{1's Compliment}} (0010)_2$

1's Compliment and 2's Compliment

1's Compliment	**2's Compliment**
Replace each 1 by 0 and each 0 by 1 $e.g.\ \ (1101)_2 \xrightarrow{\text{1's Compliment}} (0010)_2$	Find 1's compliment of given number and add 1 $e.g.\ \ (1101)_2 \xrightarrow{\text{1's Compliment}} (0010)_2$ $\begin{array}{r} 0\ 0\ 1\ 0 \\ +\ \ \ \ \ \ \ 1 \\ \hline 0\ 0\ 1\ 1 \end{array}$

1's Compliment and 2's Compliment

1's Compliment	2's Compliment
Replace each 1 by 0 and each 0 by 1 $e.g.\ (1101)_2 \xrightarrow{\text{1's Compliment}} (0010)_2$	Find 1's compliment of given number and add 1 $e.g.\ (1101)_2 \xrightarrow{\text{1's Compliment}} (0010)_2$ $\begin{array}{r} 0\ 0\ 1\ 0 \\ +\quad\quad\ 1 \\ \hline 0\ 0\ 1\ 1 \end{array}$ $(1101)_2 \xrightarrow{\text{2's Compliment}} (0011)_2$

Binary Subtraction using 1’s Compliment

Binary Subtraction using 1's Compliment

$(1100)_2 - (1010)_2 = (?)_2$

$(1010)_2 \xrightarrow{\text{1's Compliment}} (0101)_2$

Binary Subtraction using 1's Compliment

$(1100)_2 - (1010)_2 = (?)_2$

$(1010)_2 \xrightarrow{\text{1's Compliment}} (0101)_2$

$$
\begin{array}{cccccc}
 & & 1 & & & \\
 & & 1 & 1 & 0 & 0 \\
+ & & 0 & 1 & 0 & 1 \\
\hline
\text{carry} \rightarrow & 1 & 0 & 0 & 0 & 1
\end{array}
$$

0 + 0 = 0

1 + 0 = 1

0 + 1 = 1

1 + 1 = 0
with carry 1 (10)

1 + 1 +1 = 1
with carry 1 (11)

Binary Subtraction using 1's Compliment

$(1100)_2 - (1010)_2 = (?)_2$

$(1010)_2 \xrightarrow{\text{1's Compliment}} (0101)_2$

$$
\begin{array}{rccccc}
 & & 1 & & & \\
 & & 1 & 1 & 0 & 0 \\
+ & & 0 & 1 & 0 & 1 \\
\hline
\text{carry} \rightarrow & 1 & 0 & 0 & 0 & 1
\end{array}
$$

$$
\begin{array}{rcccc}
 & & & 1 & \\
 & 0 & 0 & 0 & 1 \\
+ & & & & 1 \\
\hline
 & 0 & 0 & 1 & 0
\end{array}
$$

$(1100)_2 - (1010)_2 = (0010)_2$

Binary Subtraction using 1's Compliment

$(1001)_2 - (1100)_2 = (?)_2$

$$(1100)_2 \xrightarrow{\text{1's Compliment}} (0011)_2$$

Binary Subtraction using 1's Compliment

$(1001)_2 - (1100)_2 = (?)_2$

$(1100)_2 \xrightarrow{\text{1's Compliment}} (0011)_2$

$$
\begin{array}{cccccc}
 & & \color{red}{1} & \color{red}{1} & & \\
 & 1 & 0 & 0 & 1 \\
+ & 0 & 0 & 1 & 1 \\
\hline
 & 1 & 1 & 0 & 0
\end{array}
$$

Carry is absent
Answer is -ve

$0 + 0 = 0$

$1 + 0 = 1$

$0 + 1 = 1$

$1 + 1 = 0$ with carry 1 (10)

$1 + 1 + 1 = 1$ with carry 1 (11)

Binary Subtraction using 1's Compliment

$(1001)_2 - (1100)_2 = (?)_2$

$(1100)_2 \xrightarrow{\text{1's Compliment}} (0011)_2$

$$
\begin{array}{r}
\color{red}{1\ 1} \\
1\ 0\ 0\ 1 \\
+\ \ 0\ 0\ 1\ 1 \\
\hline
1\ 1\ 0\ 0
\end{array}
$$

Carry is absent
Answer is -ve

$(1100)_2 \xrightarrow{\text{1's Compliment}} (0011)_2$

$(1001)_2 - (1100)_2 = -(0011)_2$

Binary Subtraction using 2’s Compliment

Binary Subtraction using 2's Compliment

$(1100)_2 - (1010)_2 = (?)_2$

$(1010)_2 \xrightarrow{\text{1's Compliment}} (0101)_2$

$$
\begin{array}{r}
\color{red}{1} \\
0\;1\;0\;1 \\
+ \qquad\quad 1 \\
\hline
0\;1\;1\;0
\end{array}
$$

0 + 0 = 0

1 + 0 = 1

0 + 1 = 1

1 + 1 = 0
with carry 1 (10)

1 + 1 +1 = 1
with carry 1 (11)

Binary Subtraction using 2’s Compliment

$(1100)_2 - (1010)_2 = (?)_2$

$(1010)_2 \xrightarrow{\text{1's Compliment}} (0101)_2$

$$
\begin{array}{rcccc}
 & & & \color{red}{1} & \\
 & 0 & 1 & 0 & 1 \\
+ & & & & 1 \\
\hline
 & 0 & 1 & 1 & 0
\end{array}
$$

$$
\begin{array}{rccccc}
 & \color{red}{1} & & & & \\
 & & 1 & 1 & 0 & 0 \\
+ & & 0 & 1 & 1 & 0 \\
\hline
\text{carry} \rightarrow & 1 & 0 & 0 & 1 & 0
\end{array}
$$

Answer is +ve
Drop the carry

$(1100)_2 - (1010)_2 = (0010)_2$

Binary Subtraction using 2's Compliment

$(0111)_2 - (1000)_2 = (?)_2$

$(1000)_2 \xrightarrow{\text{1's Compliment}} (0111)_2$

$$
\begin{array}{r}
\color{red}{1\ 1\ 1\ \ } \\
0\ 1\ 1\ 1 \\
+ \qquad\quad 1 \\
\hline
1\ 0\ 0\ 0
\end{array}
$$

0 + 0 = 0

1 + 0 = 1

0 + 1 = 1

1 + 1 = 0
with carry 1 (10)

1 + 1 +1 = 1
with carry 1 (11)

Binary Subtraction using 2's Compliment

$(0111)_2 - (1000)_2 = (?)_2$

$(1000)_2 \xrightarrow{\text{1's Compliment}} (0111)_2$

$$\begin{array}{rcccc} & 1 & 1 & 1 & \\ & 0 & 1 & 1 & 1 \\ + & & & & 1 \\ \hline & 1 & 0 & 0 & 0 \end{array}$$

$$\begin{array}{rcccc} & 0 & 1 & 1 & 1 \\ + & 1 & 0 & 0 & 0 \\ \hline & 1 & 1 & 1 & 1 \end{array}$$

Carry is absent

Answer is -ve

Binary Subtraction using 2’s Compliment

$(0111)_2 - (1000)_2 = (?)_2$

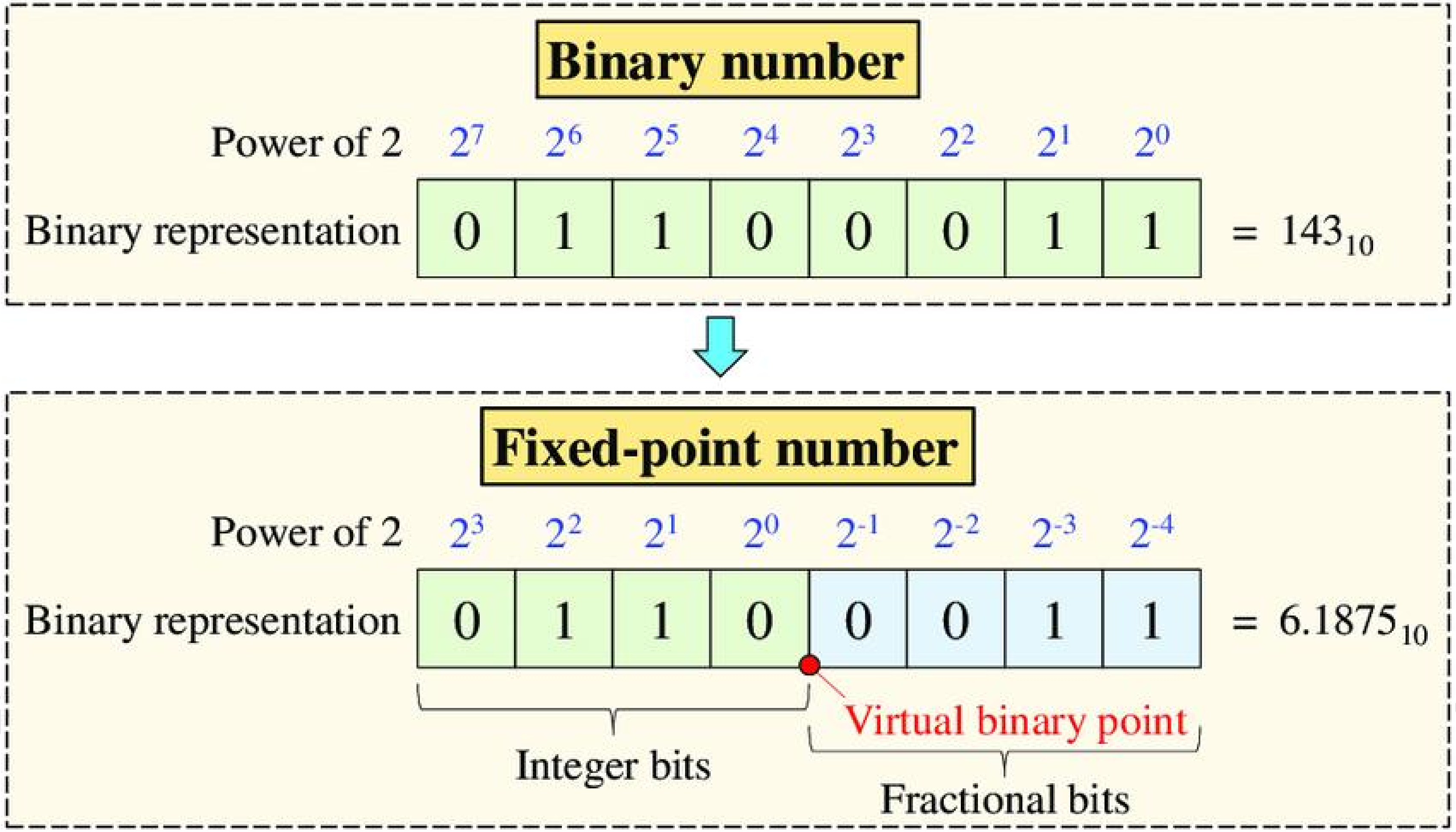
$(1111)_2 \xrightarrow{\text{1's Compliment}} (0000)_2$

$$
\begin{array}{r}
0\ 0\ 0\ 0 \\
+ \qquad\quad 1 \\
\hline
0\ 0\ 0\ 1
\end{array}
$$

$(0111)_2 - (1000)_2 = -(0001)_2$

Binary Fixed Point Representation.

Binary Fixed Point Representation.



Binary Fixed Point Representation.

Binary fixed-point representation is a method of encoding real numbers in a binary format where the position of the binary point (analogous to a decimal point) is fixed.

This representation allows for efficient arithmetic operations and is particularly useful in applications where performance is critical, such as digital signal processing and embedded systems.

Key Concepts

Structure:

- In fixed-point representation, a number is divided into two parts: an integer part and a fractional part.
- The total number of bits used to represent the number is defined by the word length (e.g., 16 bits, 32 bits).
- The position of the binary point determines how many bits are allocated to the integer part and how many to the fractional part.

Key Concepts

Representation:

- A fixed-point number can be represented as:

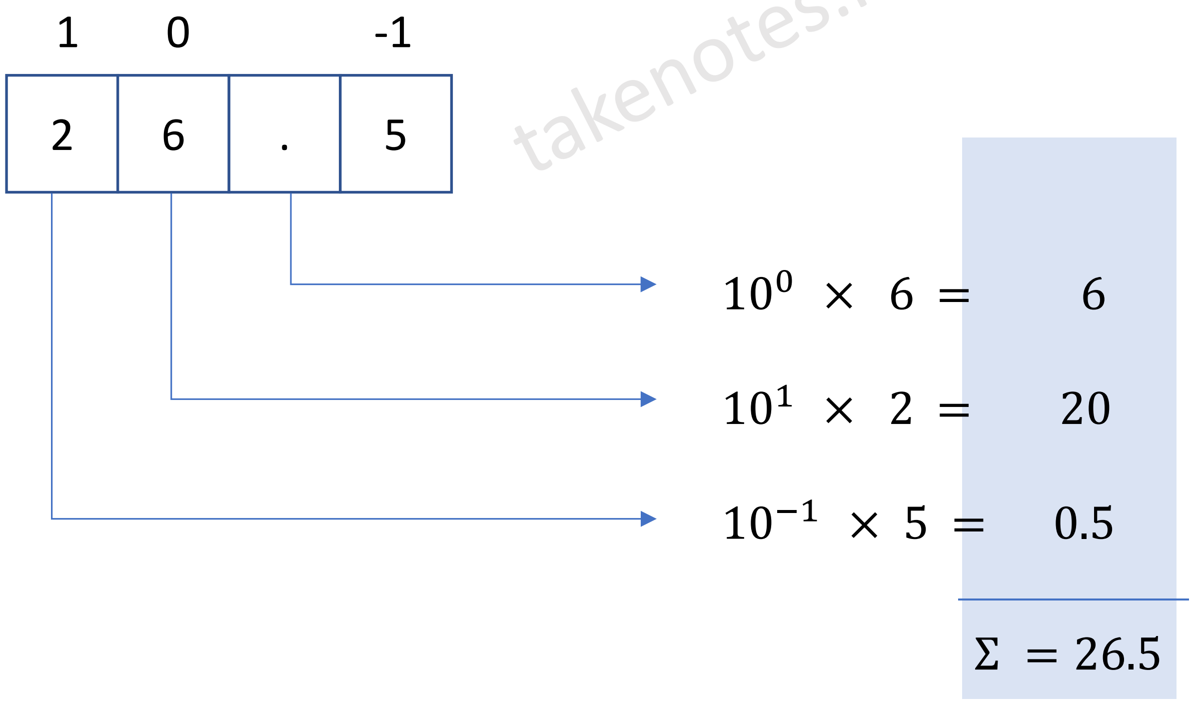
$$\text{Value} = \text{Integer Part} + \text{Fractional Part}$$

- For example, if a binary fixed-point number has 4 bits for the integer part and 4 bits for the fractional part (denoted as fixed<8,4>), the binary representation of 6.5 would be:
  - ➢ Integer part: 0110 (6 in binary)
  - ➢ Fractional part: 1000 (0.5 in binary)
  - ➢ Combined: 0110.1000

Key Concepts

Fractional values such as 26.5 are represented using the binary point concept. The decimal point in a decimal numeral system and a binary point are comparable.

It serves as a divider between a number’s integer and fractional parts.



1	0		-1
2	6	.	5

1	0
2	6

.

-1
5

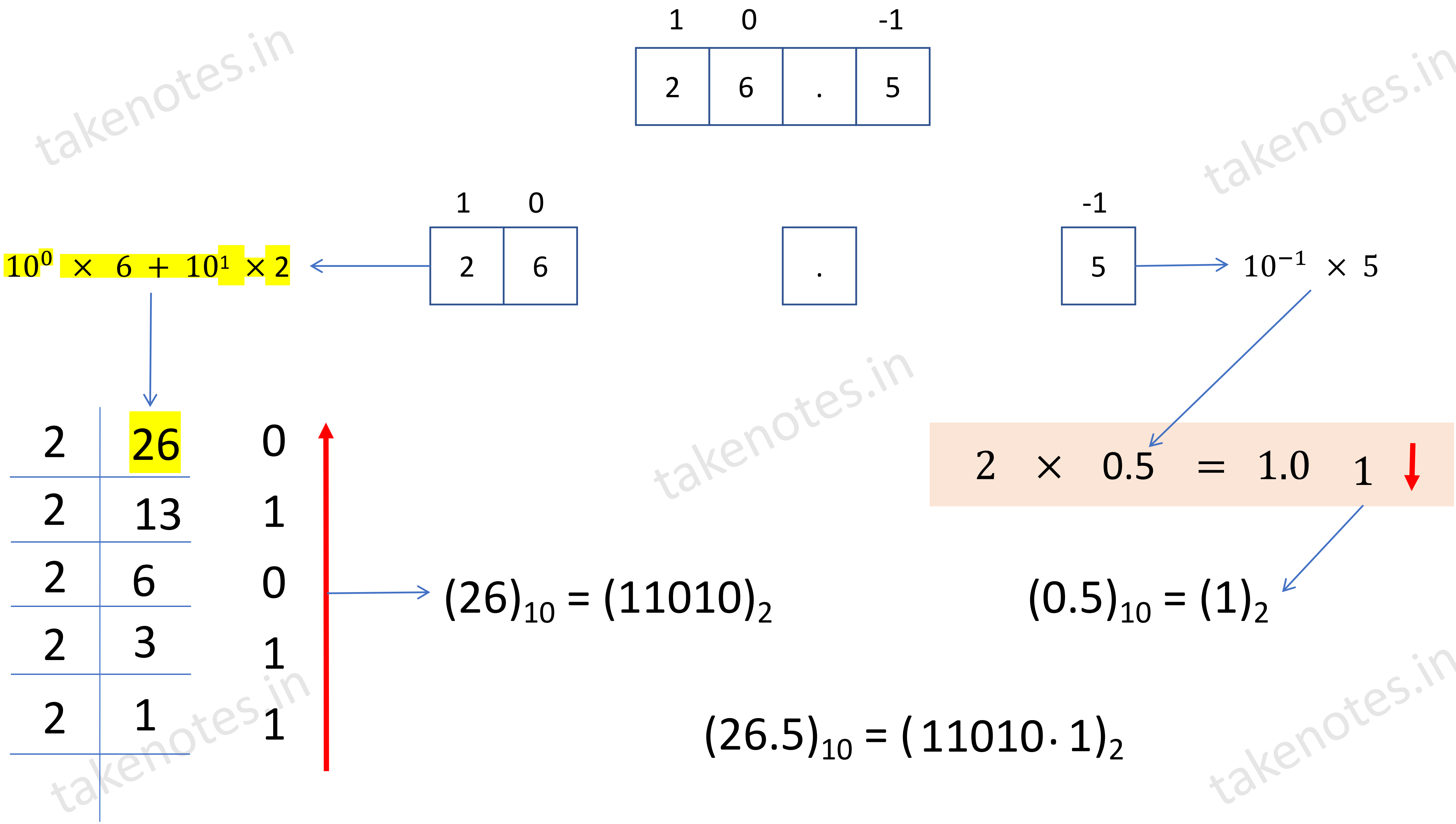
$10^0 \times 6 + 10^1 \times 2$

$10^{-1} \times 5$

$2 \times 0.5 = 1.0 \quad 1$

$(26)_{10} = (11010)_2$

$(0.5)_{10} = (1)_2$


1 0 -1
2 6 . 5
1 0
2 6
.
-1
5
$10^0 \times 6 + 10^1 \times 2$
$10^{-1} \times 5$
2 | 26 | 0
2 | 13 | 1
2 | 6 | 0
2 | 3 | 1
2 | 1 | 1
$(26)_{10} = (11010)_2$
$2 \times 0.5 = 1.0$ 1
$(0.5)_{10} = (1)_2$
$(26.5)_{10} = (11010 \cdot 1)_2$

How to Write Fixed Point Number?

Two arguments are all that are required to theoretically create a fixed point type:

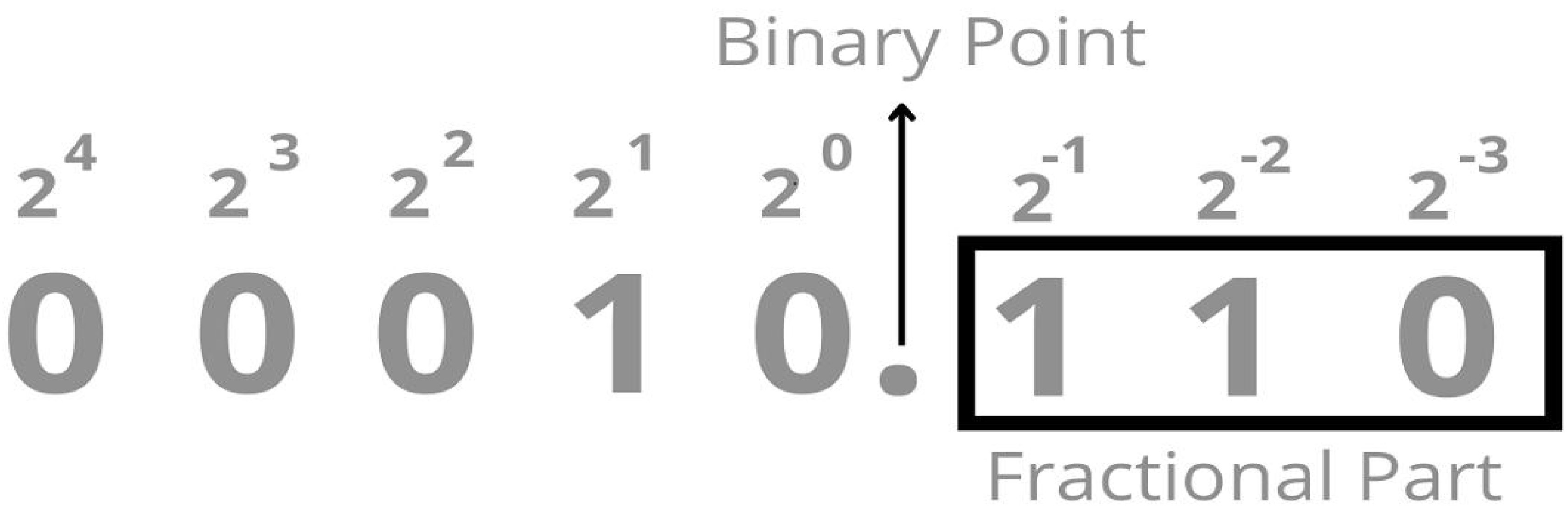
Width of the number representation.

Binary point position within the number.

the notation fixed<w, b>, where “w” stands for the overall amount of bits used (the width of a number) and “b” stands for the location of the binary point counting from the least significant bit (counting from 0).

Unsigned Representation

For example, fixed<8,3> signifies an 8-bit fixed-point number, the rightmost 3 bits of which are fractional.



Signed Representation

Signed binary numbers (+ve or -ve) can be represented in one of three ways:

1. Sign-Magnitude form
2. 1’s complement form
3. 2’s complement form

Sign-Magnitude form

The number's sign is represented by the MSB (Most Significant Bit also called as Leftmost Bit), while its magnitude is shown by the remaining bits

$$55_{10} = 00110111_2$$

$$-55_{10} = 100110111_2$$

1’s complement form

By complementing each bit in a signed binary integer, the 1’s complement of a number can be derived.

A result is a negative number when a positive number is complemented by 1. Similar to this, complementing a negative number by 1 results in a positive number.

$$55_{10} = 00110111_2$$

$$-55_{10} = 11001000_2$$

2’s complement form

By adding one to the signed binary number’s 1’s complement, a binary number can be converted to its 2’s complement.

Therefore, a positive number’s 2’s complement results in a negative number. The complement of a negative number by two yields a positive numbe

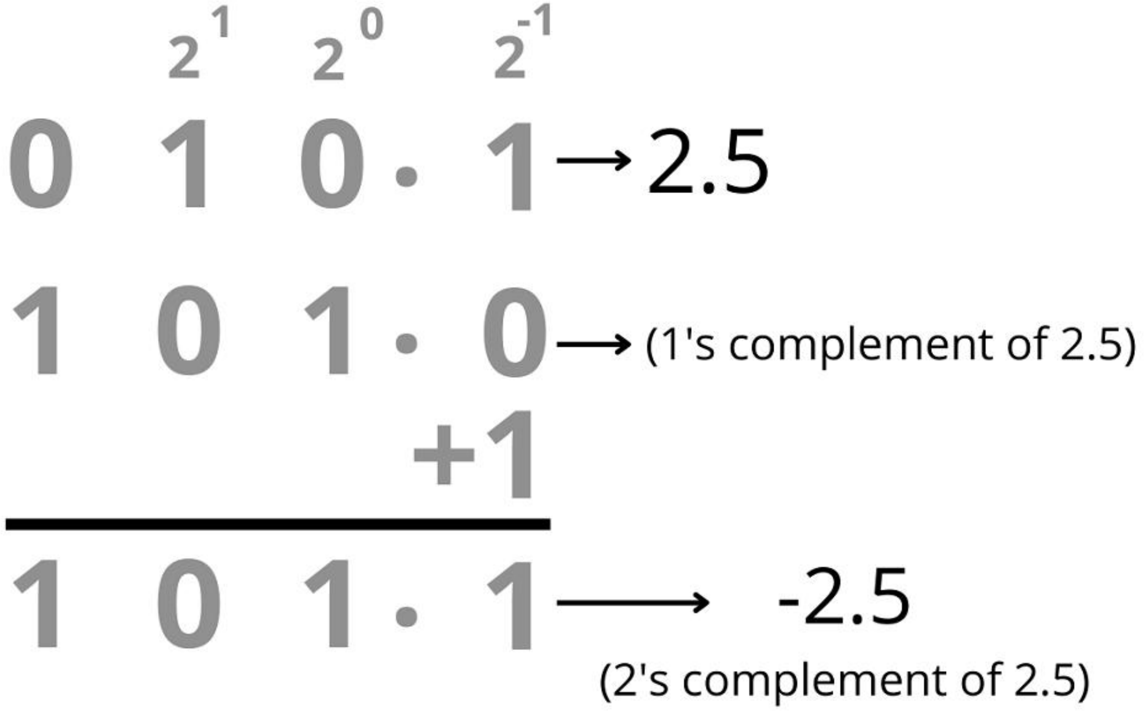
$$55_{10} = (11001000 + 1)_2$$

$$-55_{10} = (11001001)_2$$

Fixed Point Representation of Negative Number

Consider the number -2.5, fixed<w,b> width = 4 bit, binary point = 1 bit (assume the binary point is at position 1).

First, represent 2.5 in binary, then find its 2’s complement and you will get the binary fixed-point representation of -2.5.



1’s Complement Representation Range

- One bit is essentially used as a sign bit for 1’s complement numbers, leaving you with only 7 bits to store the actual number in an 8-bit number.
- Therefore, the biggest number is just 127 (anything greater would require 8 bits, making it appear to be a negative number).
- The least figure is likely to be -127 or -128 as well.

1’s Complement Representation Range

- The least figure is likely to be -127 or -128 as well.

1’s complement:

127 = 01111111 : 1s complement is 10000000
128 = 10000000 : 1s complement is 01111111

Storing -128 in 1’s complement is impossible (since the top bit is unset and it looks like a positive number)
The 1’s complement range is -127 to 127.

2’s Complement Representation Range

- Additionally, one bit in 2’s complement numbers is effectively used as a sign bit, leaving you with only 7 bits to store the actual number in an 8-bit integer.

2’s complement:

127 = 01111111 : 2s complement is 10000001

128 = 10000000 : 2s complement is 10000000

We can store -128 in 2s complement.

The 2s complement range is -128 to 127.

Advantages

Advantages:

- Performance: Fixed-point arithmetic is generally faster than floating-point arithmetic because it uses simpler hardware and fewer computational resources.
- Predictability: Fixed-point representation provides consistent precision and range, making it suitable for real-time applications.
- Integer representation and fixed point numbers are indeed close relatives.
- Because of this, fixed point numbers can also be calculated using all the arithmetic operations a computer can perform on integers.
- They are just as simple and effective as computer integer arithmetic.

Disadvantages

- Limited Range: Fixed-point numbers have a restricted range compared to floating-point numbers, making them less suitable for applications requiring high precision or large dynamic ranges.
- Precision Issues: When performing arithmetic operations, especially multiplication or division, scaling issues can arise that require careful management to avoid overflow or underflow.
- Loss in range and precision when compared to representations of floating point numbers.
- When performing arithmetic operations, particularly multiplication and division, scaling issues can arise, requiring careful handling to ensure correct results.
- Inflexibility

Applications

Binary fixed-point representation is widely used in various fields, including:

- Digital Signal Processing (DSP): Where performance is critical and operations on large volumes of data are common.
- Embedded Systems: Such as automotive control systems where computational efficiency is essential.
- Gaming Applications: Where speed is prioritized over the precision of calculations.

Applications

- Manages real-time tasks efficiently when processing large volumes of data.
- Applied in low-power or older hardware to achieve faster computations without floating-point units.
- Utilized in real-time system like automotive.
- lower computatinal const.
- efficient arithmetic operations.

Practice Questions on Number Systems

Section 1: Theory Questions

1. Explain the significance of the binary number system in computing. Why is it preferred over the decimal system?
2. Describe the process of converting a binary number to its decimal equivalent. Provide an example.

Practice Questions on Number Systems

Section 2: Numerical Problems

Convert the following numbers:

a) Decimal 255 to Binary. 11111111

b) Binary 10110101 to Decimal. 181

c) Decimal 64 to Hexadecimal. 40

Perform the following conversions:

a) Convert the binary number 11011100 to Octal. 334

b) Convert the hexadecimal number 4F to Decimal. 79

c) Convert the octal number 27 to Binary. 10111

Add the following binary numbers:

a) 101011+11001 100010

b) 1110+10101 100011

Subtract the following binary numbers:

a) 110110−10101 100001

b) 1000000−111 110001

Given the binary number 1100101

a) Calculate its 1's complement. 0011010

b) Calculate its 2's complement. 0011011

c) Convert it to Decimal. 101

Fixed Point Representation:

a) Represent the decimal number 6.75 in an 8-bit fixed-point binary format (4 bits for integer part and 4 bits for fractional part).

0110.1100

b) Convert your fixed-point representation back to decimal and verify your answer.

Given two numbers

A and B, where $A=1011_2$ and $B=1100_2$

a) Perform multiplication of A and B. 10000100

b) Convert the product back to decimal and verify the multiplication.

11*12=132

A number in hexadecimal is represented as $A3F_{16}$

2623

Convert this number into decimal and then into binary.

101000111111